

TC1100

Peripheral Units

32-Bit Single-Chip Microcontroller

32bit

Microcontrollers



Never stop thinking.

Edition 2004-07

**Published by Infineon Technologies AG,
St.-Martin-Strasse 53,
D-81541 München, Germany**

**© Infineon Technologies AG 2004.
All Rights Reserved.**

Attention please!

The information herein is given to describe certain components and shall not be considered as warranted characteristics.

Terms of delivery and rights to technical change reserved.

We hereby disclaim any and all warranties, including but not limited to warranties of non-infringement, regarding circuits, descriptions and charts stated herein.

Infineon Technologies is an approved CECC manufacturer.

Information

For further information on technology, delivery terms and conditions and prices please contact your nearest Infineon Technologies Office in Germany or our Infineon Technologies Representatives worldwide.

Warnings

Due to technical requirements components may contain dangerous substances. For information on the types in question please contact your nearest Infineon Technologies Office.

Infineon Technologies Components may only be used in life-support devices or systems with the express written approval of Infineon Technologies, if a failure of such components can reasonably be expected to cause the failure of that life-support device or system, or to affect the safety or effectiveness of that device or system. Life support devices or systems are intended to be implanted in the human body, or to support and/or maintain and sustain and/or protect human life. If they fail, it is reasonable to assume that the health of the user or other persons may be endangered.

TC1100

Peripheral Units

32-Bit Single-Chip Microcontroller

Microcontrollers



Never stop thinking.

We Listen to Your Comments

Any information within this document that you feel is wrong, unclear or missing?
Your feedback will help us to continuously improve the quality of this document.
Please send your proposal (including a reference to this document) to:

mcdocu.comments@infineon.com



Table of Contents	Page
1 Introduction	1-1
1.1 About This Document	1-1
1.1.1 Related Documentations	1-1
1.1.2 Textual Conventions	1-1
1.1.3 Reserved, Undefined, and Unimplemented Terminology	1-2
1.1.4 Register Access Modes	1-3
1.1.5 Acronyms	1-4
1.2 Peripheral Units of the TC1100	1-7
1.2.1 Serial Interfaces	1-8
1.2.1.1 Asynchronous/Synchronous Serial Interface (ASC)	1-8
1.2.1.2 High-Speed Synchronous Serial Interface (SSC)	1-11
1.2.1.3 Inter IC Serial Interface (IIC)	1-13
1.2.1.4 Micro Link Serial Bus Interface (MLI)	1-15
1.2.2 General Purpose Timer Unit	1-17
1.2.3 Capture/Compare Unit 6 (CCU6)	1-19
2 Asynchronous/Synchronous Serial Interface (ASC)	2-1
2.1 ASC Kernel Description	2-2
2.1.1 Overview	2-3
2.1.2 General Operation	2-4
2.1.3 Asynchronous Operation	2-5
2.1.3.1 Asynchronous Data Frames	2-6
2.1.3.2 Asynchronous Transmission	2-9
2.1.3.3 Transmit FIFO Operation	2-10
2.1.3.4 Asynchronous Reception	2-12
2.1.3.5 Receive FIFO Operation	2-12
2.1.3.6 FIFO Transparent Mode	2-14
2.1.3.7 IrDA Mode	2-15
2.1.3.8 RXD/TXD Data Path Selection in Asynchronous Modes	2-17
2.1.4 Synchronous Operation	2-18
2.1.4.1 Synchronous Transmission	2-19
2.1.4.2 Synchronous Reception	2-19
2.1.4.3 Synchronous Timing	2-19
2.1.5 Baud Rate Generation	2-20
2.1.5.1 Baud Rate in Asynchronous Mode	2-21
2.1.5.2 Baud Rate in Synchronous Mode	2-24
2.1.6 Hardware Error Detection Capabilities	2-25
2.1.7 Interrupts	2-27
2.2 ASC Kernel Registers	2-29
2.3 ASC0/ASC1 Module Implementation	2-45
2.3.1 Interfaces of the ASC Modules	2-45
2.3.2 ASC0/ASC1 Module Related External Registers	2-46
2.3.2.1 Clock Control Registers	2-46

Table of Contents	Page
2.3.2.2 Peripheral Input Select Register	2-48
2.3.2.3 Port Control	2-50
2.3.2.4 Interrupt Registers	2-58
2.3.3 DMA Requests	2-60
2.3.4 ASC0/ASC1 Register Address Ranges	2-60
3 Synchronous Serial Interface (SSC)	3-1
3.1 SSC Kernel Description	3-2
3.1.1 Overview	3-3
3.1.2 General Operation	3-4
3.1.2.1 Operating Mode Selection	3-6
3.1.2.2 Full-Duplex Operation	3-7
3.1.2.3 Half-Duplex Operation	3-10
3.1.2.4 Continuous Transfers	3-11
3.1.2.5 Port Control	3-12
3.1.2.6 Transmit FIFO Operation	3-13
3.1.2.7 Receive FIFO Operation	3-15
3.1.2.8 FIFO Transparent Mode	3-17
3.1.2.9 Baud Rate Generation	3-19
3.1.2.10 Slave Select Input Operation	3-21
3.1.2.11 Slave Select Output Generation Unit	3-23
3.1.2.12 Shift Clock Generation	3-25
3.1.2.13 Error Detection Mechanisms	3-26
3.2 SSC Kernel Registers	3-28
3.3 SSC0/SSC1 Module Implementation	3-45
3.3.1 Interfaces of the SSC Modules	3-45
3.3.2 SSC0/SSC1 Module Related External Registers	3-47
3.3.3 Clock Control	3-48
3.3.3.1 Port Input Select Register	3-52
3.3.3.2 Port Control	3-56
3.3.3.3 Interrupt Registers	3-72
3.3.4 DMA Requests	3-73
3.3.5 SSC0/SSC1 Register Address Ranges	3-73
4 IIC	4-1
4.1 IIC Kernel Description	4-2
4.1.1 Introduction	4-2
4.1.2 Operational Overview	4-3
4.1.3 Functional Overview	4-6
4.1.3.1 Operation in Master Mode	4-6
4.1.3.2 Operation in Multimaster Mode	4-6
4.1.3.3 Operation in Slave Mode	4-6
4.1.4 Baud Rate Selection	4-7

Table of Contents		Page
4.1.5	Interrupts	4-9
4.1.6	Synchronization	4-10
4.1.7	Programming	4-10
4.1.7.1	Initialization	4-10
4.1.7.2	Repeated Start Condition	4-10
4.1.7.3	Start Condition	4-10
4.1.7.4	Sending Data Bytes	4-10
4.1.7.5	Stop Condition	4-10
4.1.7.6	Receiving Data Bytes	4-11
4.2	IIC Kernel Registers	4-12
4.3	IIC Module Implementation	4-25
4.3.1	Interfaces of the IIC Module	4-25
4.3.2	IIC Module Related External Registers	4-26
4.3.2.1	Clock Control Register	4-27
4.3.2.2	Port Registers	4-28
4.3.2.3	Service Request Control Registers	4-31
4.3.3	DMA Requests	4-32
4.3.4	IIC Register Address Range	4-32
5	Micro Link Serial Bus Interface	5-1
5.1	MLI Kernel Description	5-2
5.1.1	MLI Applications	5-2
5.1.2	Overview	5-4
5.1.2.1	Naming Conventions	5-5
5.1.2.2	MLI Communication Principles	5-6
5.1.3	General Description	5-7
5.1.4	Handshake Description	5-13
5.1.5	Startup Procedure	5-15
5.1.6	MLI Kernel and MLI Interface Logical Connection	5-17
5.1.7	MLI Transmitter	5-18
5.1.7.1	MLI Transmitter Reset	5-18
5.1.7.2	MLI Transmitter Operation Modes	5-18
5.1.7.3	Internal Architecture and Interface Signals	5-19
5.1.7.4	Transmission Format	5-20
5.1.7.5	Transmission Modes	5-21
5.1.7.6	Transfer Mode Selection	5-27
5.1.7.7	Parity Generation	5-30
5.1.7.8	Error Detection and Handling	5-30
5.1.7.9	MLI Transmitter Input/Output Control	5-31
5.1.8	MLI Receiver	5-34
5.1.8.1	MLI Receiver Reset	5-34
5.1.8.2	MLI Receiver Operation Modes	5-34
5.1.8.3	Internal Architecture and Interface Signals	5-35

Table of Contents	Page
5.1.8.4 MLI Receiver Operation	5-36
5.1.8.5 Access Protection	5-41
5.1.8.6 Error Handling	5-42
5.1.8.7 MLI Receiver Input/Output Control	5-43
5.1.9 Reading Process Summary	5-45
5.1.10 MLI Interrupts	5-46
5.1.11 Clock Domains and Handshake Timing	5-48
5.1.12 Data Flow Description	5-53
5.1.12.1 Copy Base Address	5-53
5.1.12.2 Command Frame	5-54
5.1.12.3 Write Frame	5-55
5.1.12.4 Read Frame	5-56
5.1.12.5 Access to Remote Window	5-57
5.2 MLI Kernel Registers	5-58
5.2.1 MLI Transmitter Registers	5-61
5.2.2 MLI Receiver Registers	5-73
5.2.3 MLI Kernel Common Registers	5-79
5.2.4 MLI Interrupt Registers	5-85
5.2.5 Memory Protection Registers	5-94
5.3 MLI0 Module Implementation	5-96
5.3.1 Interface of the MLI Module	5-96
5.3.1.1 Port Connections of MLI0	5-96
5.3.2 Access Protection	5-98
5.3.3 MLI0 Module Related External Registers	5-103
5.3.3.1 DMA Requests	5-103
5.3.3.2 Interrupt Registers	5-103
5.3.3.3 Fractional Divider Registers	5-103
5.3.3.4 Port Control	5-105
5.3.4 MLI0 Register Address Ranges	5-114
6 General Purpose Timer Unit (GPTU)	6-1
6.1 GPTU Kernel Description	6-2
6.1.1 Operational Overview	6-3
6.1.2 Functional Overview	6-4
6.1.2.1 Timers T0 and T1	6-4
6.1.2.2 Input Selection	6-5
6.1.2.3 Reload Selection	6-7
6.1.2.4 Service Requests, Output Signals, and Trigger Signals	6-8
6.1.2.5 Timers T0 and T1 Configuration Limitations	6-9
6.1.2.6 Timer T2	6-10
6.1.2.7 Quadrature Counting Mode	6-17
6.1.3 Global GPTU Controls	6-18
6.1.3.1 Output Control	6-18

Table of Contents	Page
6.1.3.2 Service Request Control	6-20
6.2 GPTU Kernel Registers	6-22
6.2.1 Timer T0/T1 Registers	6-24
6.2.1.1 Timer T0/T1 Input & Reload Source Selection Register	6-24
6.2.1.2 Timer T0/T1 Output, Trigger, and Service Request Selection Register . 6-27	
6.2.1.3 Timer T0 and T1 Count and Reload Registers	6-29
6.2.2 Timer T2 Registers	6-33
6.2.2.1 Input Control Registers	6-33
6.2.2.2 Mode Control and Status Register	6-38
6.2.2.3 Timer T0/T1/T2 Run Control Register	6-41
6.2.2.4 T2 Reload/Capture Mode Control Register	6-43
6.2.2.5 Timer T2 Count and Reload/Capture Registers	6-45
6.2.3 Global Control Registers	6-47
6.3 GPTU Module Implementation	6-52
6.3.1 Interfaces of the GPTU Modules	6-52
6.3.2 GPTU Module Related External Registers	6-53
6.3.2.1 Clock Control Registers	6-54
6.3.2.2 Port Control	6-55
6.3.2.3 Interrupt Registers	6-61
6.3.3 GPTU Register Address Ranges	6-62
7 Capture/Compare Unit 6 (CCU6)	7-1
7.1 CCU6 Kernel Description	7-2
7.1.1 Overview	7-2
7.1.2 Timer T12	7-4
7.1.2.1 Overview	7-4
7.1.2.2 Counting Rules	7-5
7.1.2.3 Switching Rules	7-7
7.1.2.4 Duty Cycle of 0% and 100%	7-8
7.1.2.5 External Timer Start	7-8
7.1.2.6 Compare Mode of T12	7-9
7.1.2.7 Switching Examples in edge-aligned Mode	7-13
7.1.2.8 Switching Examples in center-aligned Mode	7-14
7.1.2.9 Dead-time Generation	7-15
7.1.2.10 Capture Mode	7-17
7.1.2.11 Single Shot Mode	7-18
7.1.2.12 Hysteresis-Like Control Mode	7-19
7.1.3 Timer T13	7-20
7.1.3.1 Overview	7-20
7.1.3.2 Compare Mode	7-21
7.1.3.3 Single Shot Mode	7-22
7.1.3.4 External Timer Start	7-22

Table of Contents	Page
7.1.3.5 Synchronization of T13 to T12	7-23
7.1.4 Modulation Control	7-24
7.1.5 Trap Handling	7-26
7.1.6 Multi-Channel Mode	7-28
7.1.7 Hall Sensor Mode	7-30
7.1.7.1 Introduction	7-30
7.1.7.2 Sampling of the Hall Pattern	7-30
7.1.7.3 Hall Events	7-31
7.1.7.4 Hall Compare Logic	7-32
7.1.7.5 Brushless-DC Control	7-33
7.1.8 Interrupt Generation	7-34
7.1.9 Suspend Mode	7-35
7.2 CCU6 Kernel Registers	7-36
7.2.1 CCU Control Registers	7-38
7.2.2 Timer12 - Related Registers	7-51
7.2.3 Timer13 - Related Registers	7-57
7.2.4 Modulation Control Registers	7-61
7.2.5 Interrupt Control Registers	7-77
7.3 CCU61 Module Implementation	7-91
7.3.1 Interface of the CCU6 Module	7-91
7.3.2 CCU61 Module Related External Registers	7-92
7.3.2.1 Clock Control	7-93
7.3.2.2 Clock Control Register	7-94
7.3.2.3 Fractional Divider Register	7-95
7.3.3 Port Control	7-96
7.3.3.1 Service Request Registers	7-102
7.3.4 DMA Requests	7-103
7.3.5 CCU61 Register Address Ranges	7-103
8 Index	8-1
8.1 Keyword Index	8-1
8.2 Register Index	8-5

1 Introduction

1.1 About This Document

This document is designed to be read primarily by design engineers and software engineers who need a detailed description of the interactions of the TC1100 functional units, registers, instructions, and exceptions.

1.1.1 Related Documentations

A complete description of the TriCore architecture is provided in the *TriCore Architecture Manual*. The architecture of the TC1100 is described separately because of the configurable nature of the TriCore architecture: different embodiments of the architecture may contain a different mix of systems components. The TriCore architecture, however, remains constant across all derivative designs in order to preserve compatibility.

Additionally to this *TC1100 Peripheral Units User's Manual*, there is also the *TC1100 System Units User's Manual*. These two User's Manuals and the *TriCore Architecture Manual* provide the complete description of the TC1100 microcontroller functionality.

Implementation-specific details such as electrical characteristics and timing parameters of the TC1100 can be found in the *TC1100 Data Sheet*.

1.1.2 Textual Conventions

This document uses the following textual conventions for named components of the TC1100:

- Functional units of the TC1100 are given in plain UPPER CASE. For example: "The EBU provides an interface to external peripherals."
- Pins using negative logic are indicated by an overbar. For example: "The $\overline{\text{BYPASS}}$ pin is latched with the rising edge of the $\overline{\text{PORST}}$ pin."
- Bit fields and bits in registers are generally referenced as "Register name.Bit field" or "Register name.Bit". For example: "The Current CPU Priority Number bit field ICR.CCPN is cleared.". Most of the register names contain a module name prefix, separated by an underscore character "_" from the real register name (for example, "ASC0_CON", where "ASC0" is the module name prefix, and "CON" is the real register name). In chapters describing peripheral modules, the real register name is referenced also as the kernel register name.
- Variables used to describe sets of processing units or registers appear in mixed-case type. For example, the register name "MSGCFGn" refers to multiple "MSGCFG" registers with the variable n. The bounds of the variables are always given where the register expression is first used (for example, "n = 31-0"), and is repeated as needed in the rest of the text.

- The default radix is decimal. Hexadecimal constants have a suffix with the subscript letter “H”, as in 100_H. Binary constants have a suffix with the subscript letter “B”, as in: 111_B.
- When the extent of register fields, groups of signals, or groups of pins are collectively named in the body of the document, they are given as “NAME[A:B]”, which defines a range for the named group from B to A. Individual bits, signals, or pins are given as “NAME[C]” where the range of the variable C is given in the text. For example: CLKSEL[2:0], and TOS[0].
- Units are abbreviated as follows:
 - **MHz** = Megahertz
 - **μs** = Microseconds
 - **kBaud, kBit** = 1000 characters/bits per second
 - **MBaud, MBit** = 1,000,000 characters per second
 - **KByte** = 1024 bytes of memory
 - **MByte** = 1,048,576 bytes of memory

In general, the *k* prefix scales a unit by 1000 whereas the *K* prefix scales a unit by 1024. Hence, the KByte unit scales the expression preceding it by 1024. The kBaud unit scales the expression preceding it by 1000. The M prefix scales by 1,000,000 or 1,048,576, and μ scales by 0.000001. For example, 1 KByte is 1024 bytes, 1 MByte is 1024 × 1024 bytes, 1 kBaud/kBit are 1000 characters/bits per second, 1 MBaud/MBit are 1,000,000 characters/bits per second, and 1 MHz is 1,000,000 Hz.
- Data format quantities are defined as follows:
 - **Byte** = 8-bit quantity
 - **Half-word** = 16-bit quantity
 - **Word** = 32-bit quantity
 - **Double-word** = 64-bit quantity

1.1.3 Reserved, Undefined, and Unimplemented Terminology

In tables where register bit fields are defined, the following conventions are used to indicate undefined and unimplemented function. Further, types of bits and bit fields are defined using the abbreviations as shown in [Table 1-1](#).

Table 1-1 Bit Function Terminology

Function of Bits	Description
Unimplemented	Register bit fields named 0 indicate unimplemented functions with the following behavior. - Reading these bit fields returns 0. - Writing these bit fields has no effect. These bit fields are reserved. When writing, software should always set such bit fields to 0 in order to preserve compatibility with future products.

Table 1-1 Bit Function Terminology (cont'd)

Function of Bits	Description
Undefined	Certain bit combinations in a bit field can be labeled “Reserved”, indicating that the behavior of the TC1100 is undefined for that combination of bits. Setting the register to undefined bit combinations may lead to unpredictable results. Such bit combinations are reserved. When writing, software must always set such bit fields to legal values as given in the tables.
rw	The bit or bit field can be read and written.
r	The bit or bit field can only be read (read-only).
w	The bit or bit field can only be written (write-only).
h	The bit or bit field can also be modified by hardware (such as a status bit). This symbol can be combined with ‘rw’ or ‘r’ bits to ‘rwh’ and ‘rh’ bits.

1.1.4 Register Access Modes

Read and write access to registers and memory locations are sometimes restricted. The following terms are used in memory and register access tables, as shown in [Table 1-2](#).

Table 1-2 Access Terms

Symbol	Description
U	Access permitted in User Mode 0 or 1
SV	Access permitted in Supervisor Mode
R	Read-only register
32	Only 32-bit word accesses are permitted to that register/address range
E	ENDINIT protected register/address
PW	Password protected register/address
NC	No change, indicated register is not changed
BE	Indicates that an access to this address range generates a Bus Error
nBE	Indicates that no Bus Error is generated when accessing this address range, even though it is either an access to an undefined address or the access does not follow the given rules
nE	Indicates that no Error is generated when accessing this address or address range, even though the access is to an undefined address or address range. True for CPU accesses (MTCR/MFCR) to undefined addresses in the CSFR range
X	Undefined value or bit

1.1.5 Acronyms

Table 1-3 lists the acronyms used in this document:

Table 1-3 Acronyms

AGPR	Address General Purpose Register
ALE	Address Latch Enable
ALU	Arithmetic and Logic Unit
ASC	Asynchronous/Synchronous Serial Controller
ASI	Address Space Identifier
BCU	Bus Control Unit
BIV	Base of Interrupt Vector
CCU6	Capture/Compare Unit 6
CISC	Complex Instruction Set Computing
CPS	CPU Slave (Interface Register)
CPU	Central Processing Unit
CSA	Context Save Area
CSFR	Core Special Function Register
DCACHE	Data Cache
DGPR	Data General Purpose Register
DMA	Direct Memory Access
DMI	Direct Memory Interface
DMU	Data Memory Unit
DRAM	Dynamic Random Access Memory
DSP	Digital Signal Processor
EBU	External Bus Unit
EMI	Electromagnetic Interference
FIFO	First-In First-Out
FPI	Flexible Peripheral Interconnect (Bus)
FPU	Floating-Point Unit
GPIO	General Purpose I/O
GPR	General Purpose Register
GPTU	General Purpose Timer Unit
I/O	Input/Output

Table 1-3 Acronyms (cont'd)

ICACHE	Instruction Cache
ICE	In-Circuit Emulation
ICR	Interrupt Control Register
ICU	Interrupt Control Unit
IIC	Inter IC
ISA	Instruction Set Architecture
ISR	Interrupt Service Routine
JTAG	Joint Test Action Group
LFI	LMB to FPI Interface
LMB	Local Memory Bus
MFCR	Move From Core Register
MLI	Micro Link Serial Interface (Bus)
MMCI	MultiMediaCard Interface
MMU	Memory Management Unit
MTCR	Move To Core Register
NMI	Non-Maskable Interrupt
OCDS	On-Chip Debug Support
PC	Program Counter
PCODE	PCP Code Memory
PCP	Peripheral Control Processor
PLL	Phase Locked Loop
PMI	Program Memory Interface
PMSM	Power Management State Machine
PMU	Program Memory Unit
PPN	Physical Page Number
PRAM	PCP Parameter RAM
PRS	Protection Register Set
PSW	Processor Status Word
PTE	Page Table Entry
PWM	Pulse Width Modulation
RAM	Random Access Memory

Table 1-3 Acronyms (cont'd)

RFE	Return From Exception
RISC	Reduced Instruction Set Computing
SCU	System Control Unit
SFR	Special Function Register
SIMD	Single Instruction Multiple Data
SMT	Software Managed Task
SPRAM	Scratch-Pad Random Access Memory (Code)
SRAM	Static Random Access Memory (Data)
SRN	Service Request Node
SRPN	Service Request Priority Number
SSC	Synchronous Serial Controller
STM	System Timer
TIN	Trap Identification Number
TLB	Translation Lookaside Buffer
TSR	Trap Service Routine
TTE	TLB Table Entry
VPN	Virtual Page Number
WDT	Watchdog Timer

1.2 Peripheral Units of the TC1100

The TC1100 microcontroller offers several versatile on-chip peripheral units such as serial controllers and timer units. Within the TC1100, all these peripheral units are connected to the TriCore CPU/system via the FPI (Flexible Peripheral Interconnect) Buses. Several I/O lines on the TC1100 ports are reserved for these peripheral units to communicate with the external world.

The following peripherals are described in detail in this *TC1100 Peripheral Units User's Manual*:

Peripheral Units of the TC1100:

- Two Asynchronous/Synchronous Serial Channels (ASC0/1) with baud rate generator, parity, framing and overrun error detection, and IrDA data transmission. An 8-byte data buffer (FIFO with depth of 8) for each ASC.
- Two High Speed Synchronous Serial Channels (SSC0/1) with programmable data length and shift direction. A 4-byte data buffer (FIFO with depth of 4) for each SSC.
- One Inter IC Serial Module with two channels.
- One high speed Micro Link Interfaces (MLI0) for controller communication and emulation.
- One Multifunctional General Purpose Timer Unit (GPTU) with three 32-bit timer/counters.
- One Capture and Compare Unit 6 (CCU6) for PWM signal generation.

The remaining sections in this chapter provide an overview of these peripheral units.

1.2.1 Serial Interfaces

The TC1100 includes four serial peripheral interface units:

- Asynchronous/Synchronous Serial Interface (ASC)
- High-Speed Synchronous Serial Interface (SSC)
- Inter IC Serial Interface (IIC)
- Micro Link Serial Bus Interface (MLI)

1.2.1.1 Asynchronous/Synchronous Serial Interface (ASC)

Figure 1-1 shows the functional blocks of the two Asynchronous/Synchronous Serial interfaces (ASC0 and ASC1).

Each ASC Module (ASC0/ASC1) communicates with the external world via one pair of I/O lines. The RXD line is the receive data input signal (also the output signal in Synchronous Mode). TXD is the transmit output signal. Clock control, address decoding, and interrupt service request control are managed outside the ASC Module kernel.

The Asynchronous/Synchronous Serial Interfaces provide serial communication between the TC1100 and other microcontrollers, microprocessors or external peripherals.

Each ASC supports full-duplex asynchronous communication and half-duplex synchronous communication. In Synchronous Mode, data is transmitted or received synchronous to a shift clock that is generated by the ASC internally. In Asynchronous Mode, 8-bit or 9-bit data transfer, parity generation, and the number of stop bits can be selected. Parity, framing, and overrun error detection are provided to increase the reliability of data transfers. Transmission and reception of data are double-buffered. For multiprocessor communication, a mechanism is included to distinguish address bytes from data bytes. Testing is supported by a loop-back option. A 13-bit baud rate generator provides the ASC with a separate serial clock signal that can be accurately adjusted by a prescaler implemented as a fractional divider.

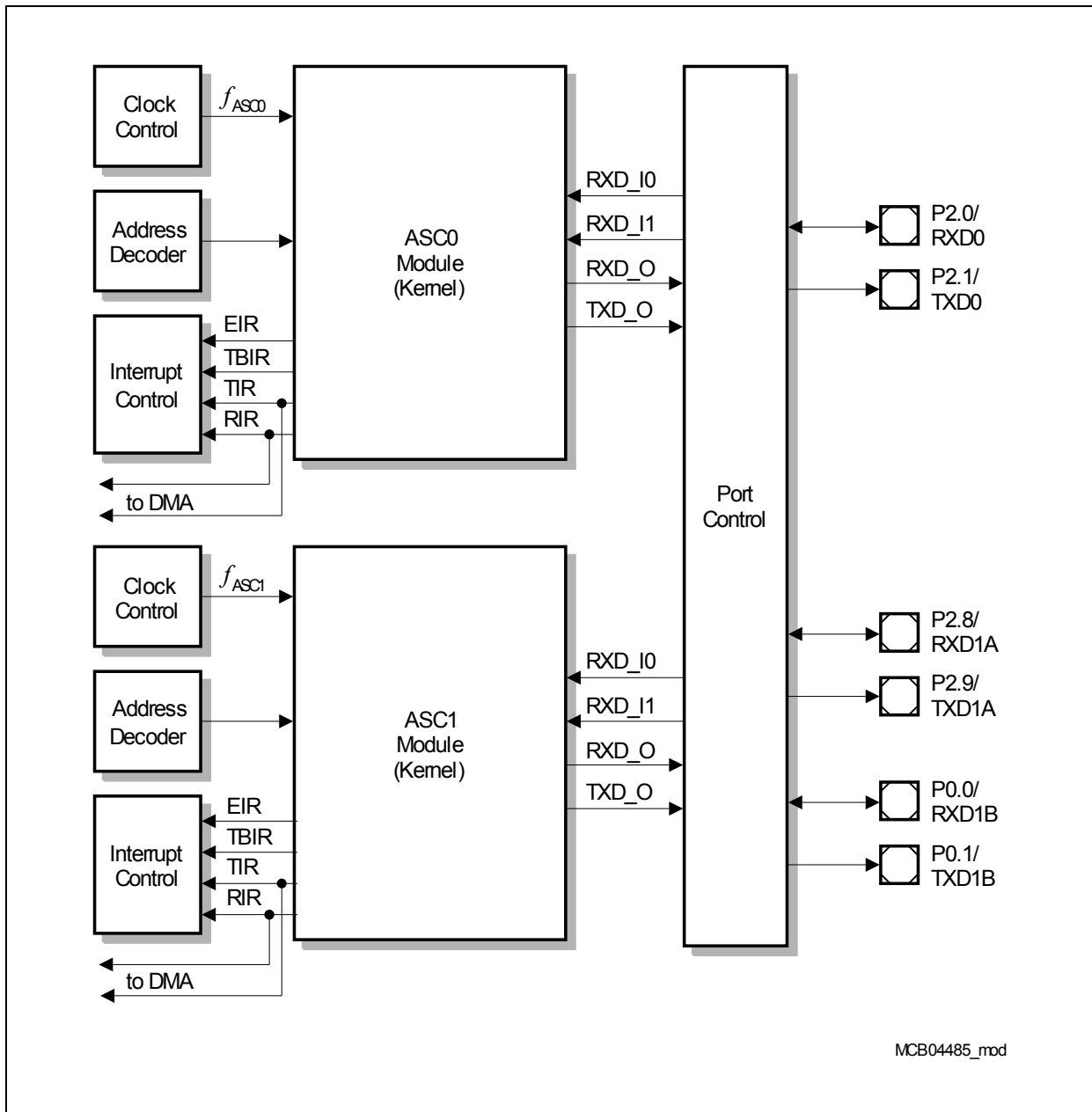


Figure 1-1 General Block Diagram of the ASC Interfaces

Features:

- Full-duplex asynchronous operating modes
 - 8-bit or 9-bit data frames, LSB first
 - Parity bit generation/checking
 - One or two stop bits
 - Baud rate from 4.6875 MBaud to 1.1 Baud (@ 75 MHz clock)
- Multiprocessor Mode for automatic address/data byte detection

- Loop-back capability
- Half-duplex 8-bit synchronous operating mode
 - Baud rate from 9.375 MBaud to 762.9 Baud (@ 75 MHz clock)
- Support for IrDA data transmission up to 115.2 KBaud maximum.
- Double buffered transmitter/receiver
- Interrupt generation
 - On a transmitter buffer empty condition
 - On a transmit last bit of a frame condition
 - On a receiver buffer full condition
 - On an error condition (frame, parity, overrun error)
- FIFO
 - 8-stage receive FIFO (RXFIFO)
 - 8-stage transmit FIFO (TXFIFO)
 - Independent control of RXFIFO and TXFIFO
 - 9-bit FIFO data width
 - Programmable Receive/Transmit Interrupt Trigger Level
 - Receive and Transmit FIFO filling level indication
 - Overrun error generation
 - Underflow error generation

1.2.1.2 High-Speed Synchronous Serial Interface (SSC)

Figure 1-2 shows the functional blocks of two High-Speed Synchronous Serial interfaces (SSC0 and SSC1).

Each SSC supports full-duplex and half-duplex serial synchronous communication up to 37.5 MBaud (@ 75 MHz module clock) with receive and transmit FIFO support. The serial clock signal can be generated by the SSC itself (master mode) or can be received from an external master (slave mode). Data width, shift direction, clock polarity and phase are programmable. This allows communication with SPI-compatible devices. Transmission and reception of data is double-buffered. A shift clock generator provides the SSC with a separate serial clock signal. Eight slave select inputs are available for slave mode operation. Eight programmable slave select outputs (chip selects) are supported in master mode.

Features:

- Master and slave mode operation
 - Full-duplex or half-duplex operation
 - Automatic pad control possible
- Flexible data format
 - Programmable number of data bits: 2 to 16 bits
 - Programmable shift direction: LSB or MSB shift first
 - Programmable clock polarity: idle low or high state for the shift clock
 - Programmable clock/data phase: data shift with leading or trailing edge of the shift clock
- Baud rate generation minimum at 572.2 Baud (@ 75 MHz module clock)
- Interrupt generation
 - On a transmitter empty condition
 - On a receiver full condition
 - On an error condition (receive, phase, baud rate, transmit error)
- Four-pin interface
- Flexible SSC pin configuration
- Up to eight slave select inputs in slave mode
- Up to eight programmable slave select outputs SLSO in master mode
 - Automatic SLSO generation with programmable timing
 - Programmable active level and enable control
- 4-stage receive FIFO (RXFIFO) and 4-stage transmit FIFO (TXFIFO)
 - Independent control of RXFIFO and TXFIFO
 - 2 to 16 bit FIFO data width
 - Programmable receive/transmit interrupt trigger level
 - Receive and transmit FIFO filling level indication
 - Overrun error generation
 - Underflow error generation

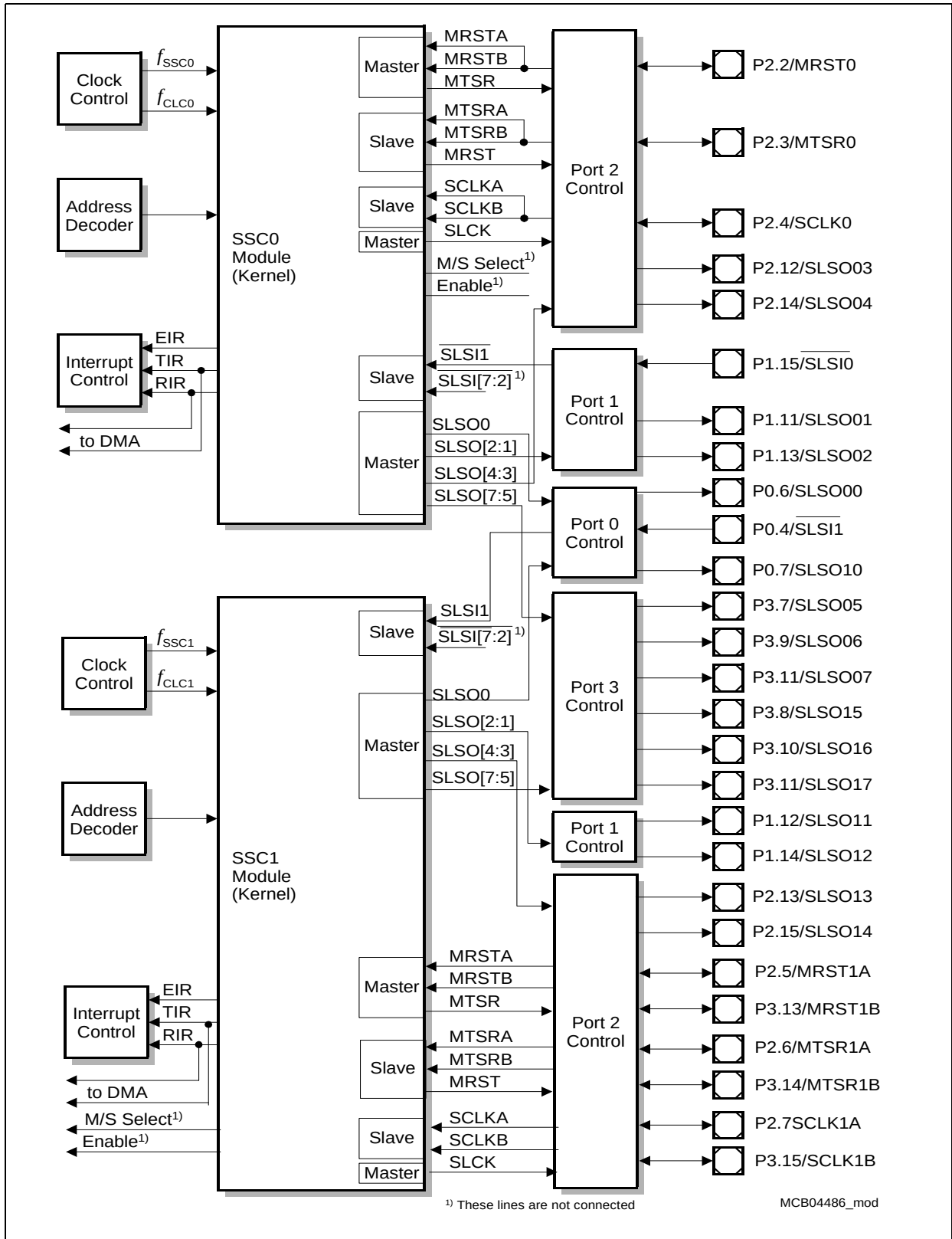


Figure 1-2 General Block Diagram of the SSC Interfaces

1.2.1.3 Inter IC Serial Interface (IIC)

Figure 1-3 shows the functional blocks of the Inter IC Serial Interface (IIC).

The IIC module has four I/O lines, located at Port 2. The IIC module is also supplied by a clock control, interrupt control, and address decoding logic. One DMA request can be generated by IIC module.

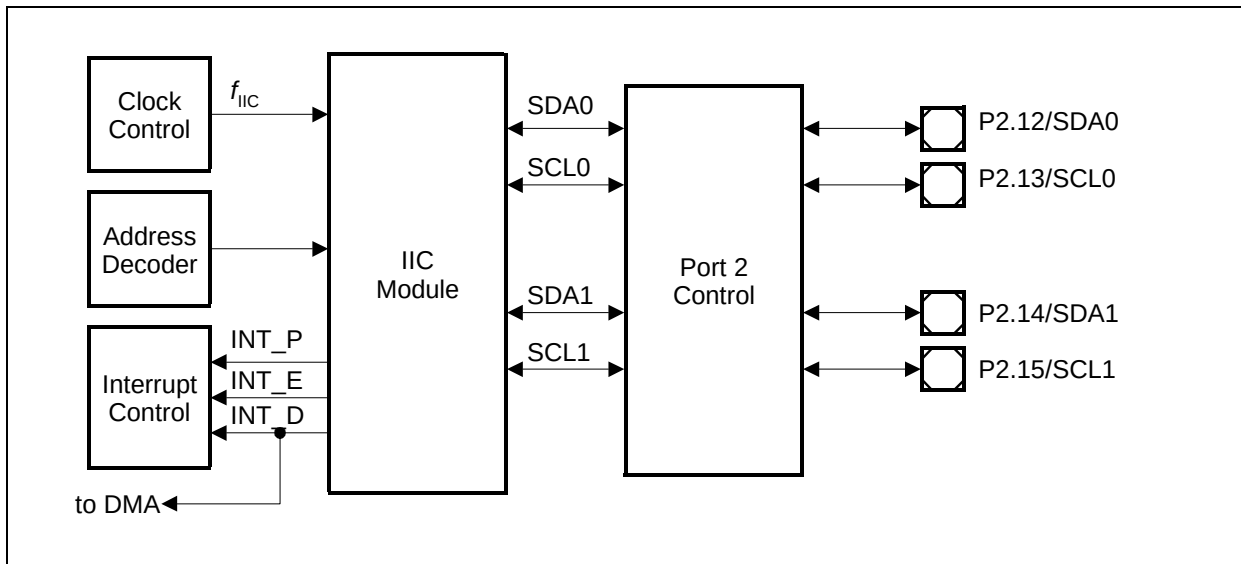


Figure 1-3 General Block Diagram of the IIC Interface

The on-chip IIC Bus module connects the platform buses to other external controllers and/or peripherals via the two-line serial IIC interface. One line is responsible for clock transfer and synchronization (SCL), the other is responsible for the data transfer (SDA). The IIC Bus module provides communication at data rates of up to 400 Kbit/s and features 7-bit addressing as well as 10-bit addressing. This module is fully compatible with the IIC bus protocol.

The module can operate in three different modes:

Master Mode, where the IIC controls the bus transactions and provides the clock signal.

Slave Mode, where an external master controls the bus transactions and provides the clock signal.

Multimaster Mode, where several masters can be connected to the bus, i.e. the IIC can be master or slave.

The on-chip IIC bus module allows efficient communication via the common IIC bus. The module unloads low level tasks from the CPU such as:

- (De)Serialization of bus data.
- Generation of start and stop conditions.
- Monitoring the bus lines in slave mode.
- Evaluation of the device address in slave mode.

- Bus access arbitration in multimaster mode.

Features

- Extended buffer allows up to 4 send/receive data bytes to be stored.
- Selectable baud rate generation.
- Support of standard 100 KBaud and extended 400 KBaud data rates.
- Operation in 7-bit addressing mode or 10-bit addressing mode.
- Flexible control via interrupt service routines or by polling.
- Dynamic access to up to 2 physical IIC buses.

1.2.1.4 Micro Link Serial Bus Interface (MLI)

Figure 1-4 shows the functional blocks of the Micro Link Serial Bus Interface (MLI0).

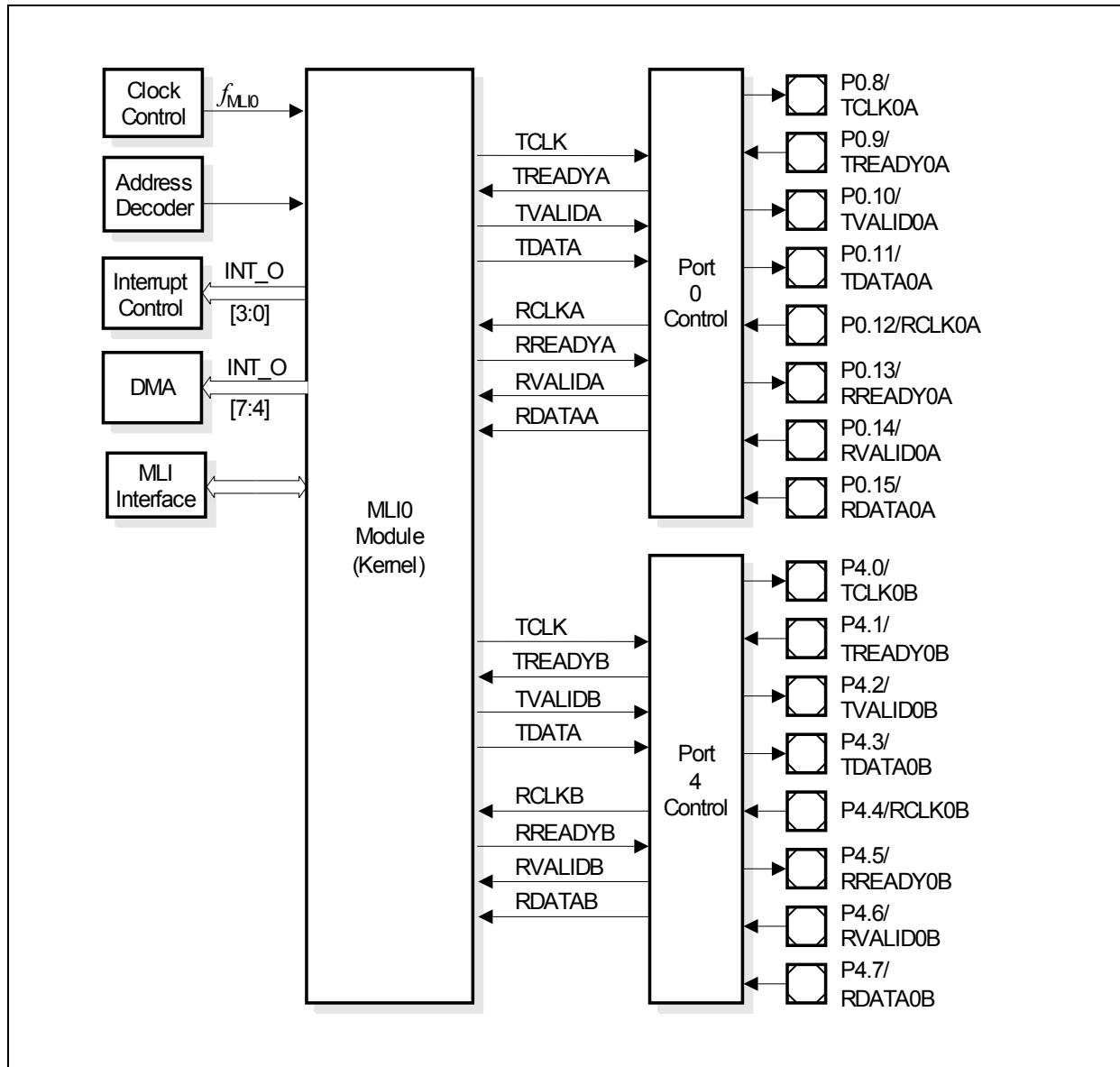


Figure 1-4 General Block Diagram of the MLI0

The Micro Link Serial Bus Interface is dedicated for the serial communication between controllers of the AUDO - NG family. The communication is intended to be fast and intelligent due to an address translation system, and it is not necessary to have any special program in the second controller.

Features:

- Serial communication from the MLI transmitter to the MLI receiver of another controller
- The Module supports connection of the MLI with up to four MLI from other controllers (see implementation sub-chapter for details of this product)
- Fully transparent read/write access supported (= remote programming)
- Complete address range of target controller available
- Special protocol to transfer data, address offset, or address offset and data
- Error control using a parity bit
- 8-bit, 16-bit, and 32-bit data transfers
- Address offset width: from 1 to 16 bits
- Baud rate: $f_{\text{MLI}} / 2$ (symmetric shift clock approach),
baud rate defined by the corresponding fractional divider

1.2.2 General Purpose Timer Unit

Figure 1-5 shows all the functional blocks of the General Purpose Timer Unit (GPTU).

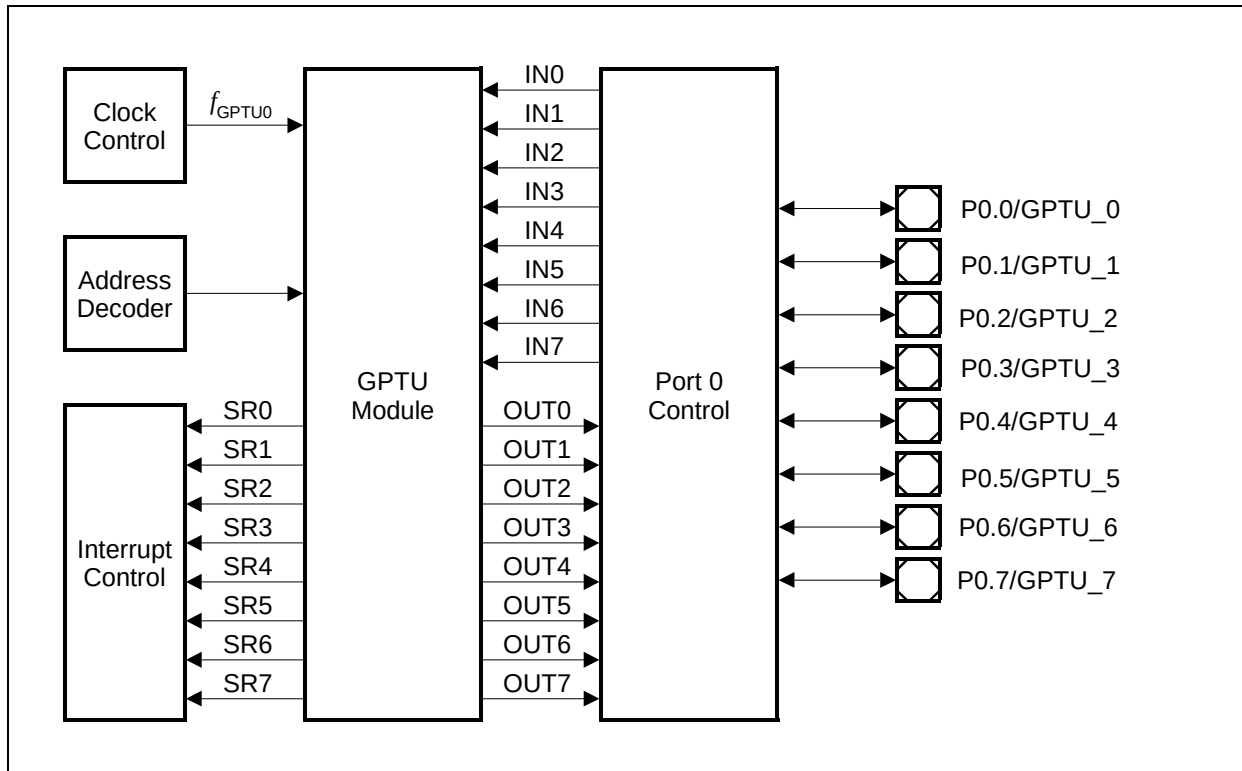


Figure 1-5 General Block Diagram of the GPTU Interface

The GPTU consists of three 32-bit timers designed to solve such application tasks as event timing, event counting, and event recording. The GPTU communicates with the external world via eight I/O lines located at Port 0.

The three timers of GPTU Module T0, T1, and T2, can operate independently from each other or can be combined:

General Features:

- All timers are 32-bit precision timers with a maximum input frequency of f_{GPTU} .
- Events generated in T0 or T1 can be used to trigger actions in T2
- Timer overflow or underflow in T2 can be used to clock either T0 or T1
- T0 and T1 can be concatenated to form one 64-bit timer

Features of T0 and T1:

- Each timer has a dedicated 32-bit reload register with automatic reload on overflow
- Timers can be split into individual 8-, 16-, or 24-bit timers with individual reload registers

- Overflow signals can be selected to generate service requests, pin output signals, and T2 trigger events
- Two input pins can define a count option

Features of T2:

- Count up or down is selectable
- Operating modes:
 - Timer
 - Counter
 - Quadrature counter (incremental/phase encoded counter interface)
- Options:
 - External start/stop, one-shot operation, timer clear on external event
 - Count direction control through software or an external event
 - Two 32-bit reload/capture registers
- Reload modes:
 - Reload on overflow or underflow
 - Reload on external event: positive transition, negative transition, or both transitions
- Capture modes:
 - Capture on external event: positive transition, negative transition, or both transitions
 - Capture and clear timer on external event: positive transition, negative transition, or both transitions
- Can be split into two 16-bit counter/timers
- Timer count, reload, capture, and trigger functions can be assigned to input pins. T0 and T1 overflow events can also be assigned to these functions.
- Overflow and underflow signals can be used to trigger T0 and/or T1 and to toggle output pins
- T2 events are freely assignable to the service request nodes.

1.2.3 Capture/Compare Unit 6 (CCU6)

Figure 1-6 shows all the functional blocks of the Capture/Compare Units (CCU6).

The CCU6 module is further supplied by clock control, interrupt control, address decoding, and port control logic. One DMA request can be generated by the CCU6 module.

The CCU6 provides two independent timers (T12, T13) that can be used for PWM generation, especially for AC-motor control. Additionally, special control modes for block commutation and multi-phase machines are supported.

Timer 12 Features

- Three capture/compare channels, each channel can be used as either a capture or as compare channel.
- Generation of a three-phase PWM supported (six outputs, individual signals for highside and lowside switches)
- 16-bit resolution, maximum count frequency = peripheral clock
- Dead-time control for each channel to avoid short-circuits in the power stage
- Concurrent update of the required T12/13 registers
- Center-aligned and edge-aligned PWM can be generated
- Single-shot mode supported
- Many interrupt request sources
- Hysteresis-like control mode

Timer 13 Features

- One independent compare channel with one output
- 16-bit resolution, maximum count frequency = peripheral clock
- Can be synchronized to T12
- Interrupt generation at period-match and compare-match
- Single-shot mode supported

Additional Features

- Block commutation for Brushless DC-drives implemented
- Position detection via Hall-sensor pattern
- Automatic rotational speed measurement for block commutation
- Integrated error handling
- Fast emergency stop without CPU load via external signal ($\overline{\text{CTRAP}}$)
- Control modes for multi-channel AC-drives
- Output levels can be selected and adapted to the power stage

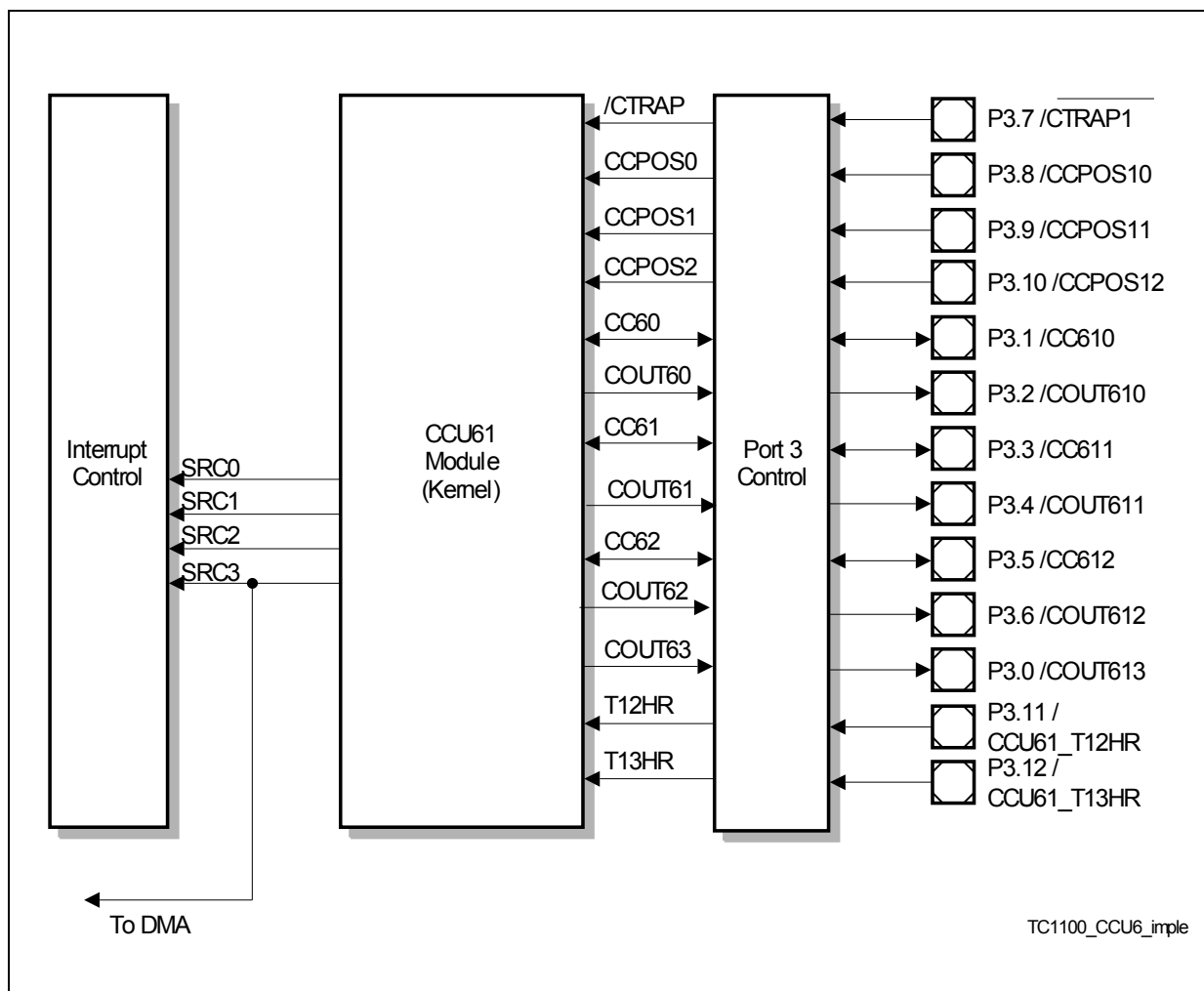


Figure 1-6 General Block Diagram of the CCU6

2 Asynchronous/Synchronous Serial Interface (ASC)

This chapter describes the two ASC asynchronous/synchronous serial interfaces ASC0 and ASC1 of the TC1100. It contains the following sections:

- Functional description of the ASC Kernel, valid for ASC0 and ASC1 (see [Section 2.1](#))
- Register descriptions of all ASC Kernel specific registers (see [Section 2.2](#))
- TC1100 implementation specific details and registers of the ASC0/ASC1 modules (port connections and control, interrupt control, address decoding, clock control, see [Section 2.3](#)).

Note: The ASC kernel register names described in [Section 2.2](#) will be referenced in the TC1100 User's Manual by the module name prefix "ASC0_" for the ASC0 interface and by "ASC1_" for the ASC1 interface.

Asynchronous/Synchronous Serial Interface (ASC)

2.1 ASC Kernel Description

Figure 2-1 shows a global view of all functional blocks of the ASC interface.

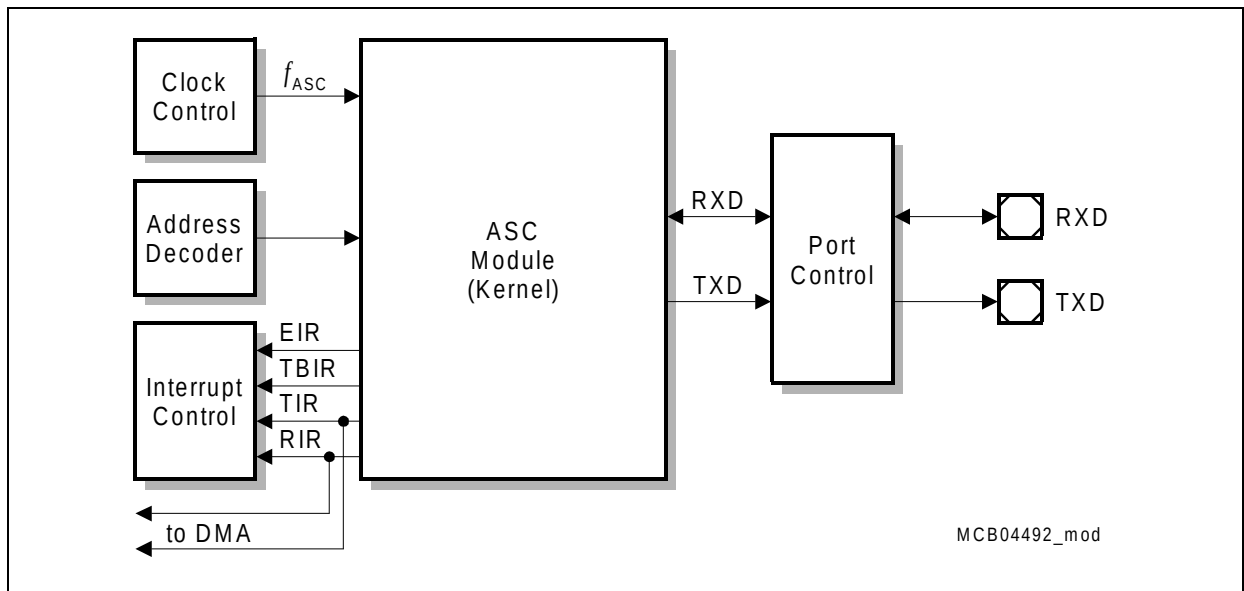


Figure 2-1 General Block Diagram of the ASC Interface

The ASC module communicates with the external world via two I/O lines. The RXD line is the receive data input signal (in synchronous mode also output), and TXD is the transmit output signal.

Clock control, address decoding, and interrupt service request control are managed outside the ASC Module kernel.

Asynchronous/Synchronous Serial Interface (ASC)**2.1.1 Overview**

The Asynchronous/Synchronous Serial Interfaces provide serial communication between the TC1100 and other microcontrollers, microprocessors or external peripherals.

The ASC supports full-duplex asynchronous communication and half-duplex synchronous communication. In Synchronous Mode, data is transmitted or received synchronous to a shift clock generated by the ASC internally. In Asynchronous Mode, 8-bit or 9-bit data transfer, parity generation and the number of stop bits can be selected. Parity, framing and overrun error detection are provided to increase the reliability of data transfers. Transmission and reception of data are double-buffered. For multiprocessor communication, a mechanism is included to distinguish address bytes from data bytes. Testing is supported by a loop-back option. A 13-bit baud rate generator provides the ASC with a separate serial clock signal, which can be very accurately adjusted by a prescaler implemented as fractional divider.

Features:

- Full-duplex asynchronous operating modes
 - 8-bit or 9-bit data frames, LSB first
 - Parity bit generation/checking
 - One or two stop bits
 - Baud rate from 4.6875 MBaud to 1.12 Baud (@ 75 MHz clock)
- Multiprocessor mode for automatic address/data byte detection
- Loop-back capability
- Half-duplex 8-bit synchronous operating mode
 - Baud rate from 9.375 MBaud to 762.9 Baud (@ 75 MHz clock)
- Support for IrDA data transmission up to 115.2 KBaud maximum.
- Double buffered transmitter/receiver
- Interrupt generation
 - On a transmitter buffer empty condition
 - On a transmit last bit of a frame condition
 - On a receiver buffer full condition
 - On an error condition (frame, parity, overrun error)
- FIFO
 - 8-stage receive FIFO (RXFIFO)
 - 8-stage transmit FIFO (TXFIFO)
 - Independent control of RXFIFO and TXFIFO
 - 9-bit FIFO data width
 - Programmable Receive/Transmit Interrupt Trigger Level
 - Receive and Transmit FIFO filling level indication
 - Overrun error generation
 - Underflow error generation

Asynchronous/Synchronous Serial Interface (ASC)**2.1.2 General Operation**

The ASC supports full-duplex asynchronous communication up to 4.6875 MBaud and half-duplex synchronous communication up to 9.375 MBaud (@ 75 MHz module clock). In Synchronous Mode, data are transmitted or received synchronous to a shift clock generated by the microcontroller. In Asynchronous Mode, 8-bit or 9-bit data transfer, parity generation, and the number of stop bits can be selected. Parity, framing, and overrun error detection are provided to increase the reliability of data transfers. Transmission and reception of data are double-buffered. For multiprocessor communication, a mechanism is included to distinguish address bytes from data bytes. Testing is supported by a loop-back option. A 13-bit baud rate timer with a versatile input clock divider circuitry provides the ASC with the serial clock signal. In a special asynchronous mode, the ASC supports IrDA data transmission up to 115.2 KBaud with fixed or programmable IrDA pulse width.

A transmission is started by writing to the Transmit Buffer register TBUF. Only the number of data bits determined by the selected operating mode will actually be transmitted, that is, bits written to positions 9 through 15 of register TBUF are always insignificant.

Data transmission is double-buffered, so a new character may be written to the transmit buffer register, before the transmission of the previous character is complete. This allows back-to-back transmission of characters without gaps.

Data reception is enabled by the Receiver Enable Bit CON.REN. After reception of a character has been completed, the received data and, if provided by the selected operating mode, the received parity bit can be read from the (read-only) Receive Buffer register RBUF. Bits in the upper half of RBUF that are not valid in the selected operating mode will be read as zeroes.

Data reception is double-buffered, so that reception of a second character may begin before the previously received character has been read out of the receive buffer register. In all modes, receive buffer overrun error detection can be selected through bit CON.OEN. When enabled, the overrun error status flag CON.OE and the error interrupt request line EIR will be activated when the receive buffer register has not been read by the time reception of a second character is complete. The previously received character in the receive buffer is overwritten.

The Loop-Back option (selected by bit CON.LB) allows the data currently being transmitted to be received simultaneously in the receive buffer. This may be used to test serial communication routines at an early stage without having to provide an external network. In Loop-Back Mode, the alternate input/output function of port pins is not required.

Asynchronous/Synchronous Serial Interface (ASC)

2.1.3 Asynchronous Operation

Asynchronous mode supports full-duplex communication, where both transmitter and receiver use the same data frame format and have the same baud rate. Data is transmitted on pin TXD and received on pin RXD. IrDA data transmission/reception is supported up to 115.2 KBit/s. **Figure 2-2** shows the block diagram of the ASC when operating in Asynchronous Mode.

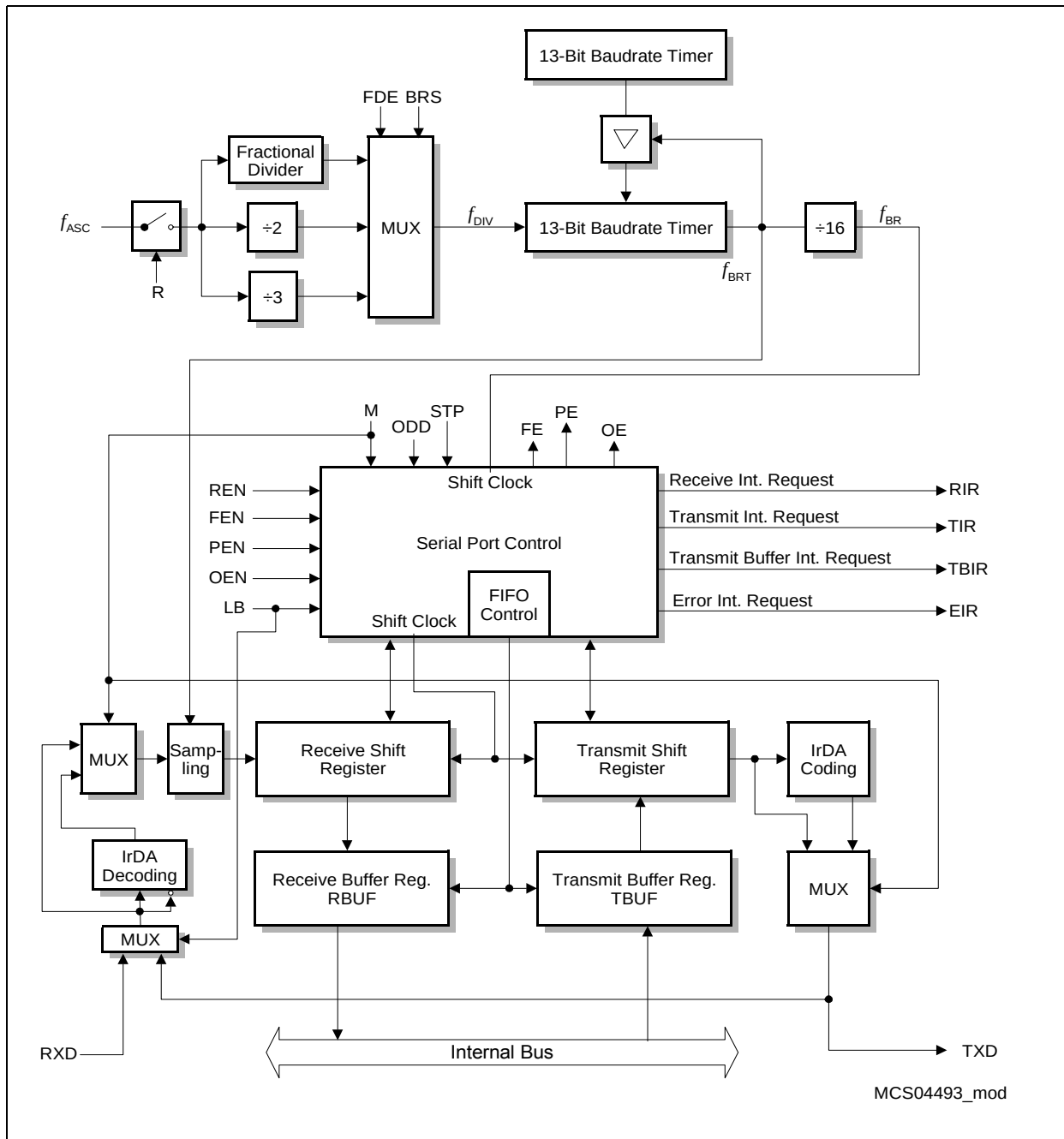


Figure 2-2 Asynchronous Mode of the ASC

Asynchronous/Synchronous Serial Interface (ASC)

2.1.3.1 Asynchronous Data Frames

8-Bit Data Frames

8-bit data frames consist of either eight data bits D7 ... D0 (CON.M = 001_B), or of seven data bits D6 ... D0 plus an automatically generated parity bit (CON.M = 011_B). Parity may be odd or even, depending on bit CON.ODD. An even parity bit will be set if the modulo-2-sum of the seven data bits is 1. An odd parity bit will be cleared in this case. Parity checking is enabled via bit CON.PEN (always OFF in 8-bit data mode). The parity error flag CON.PE will be set, along with the error interrupt request flag, if a wrong parity bit is received. The parity bit itself will be stored in bit RBUF.7.

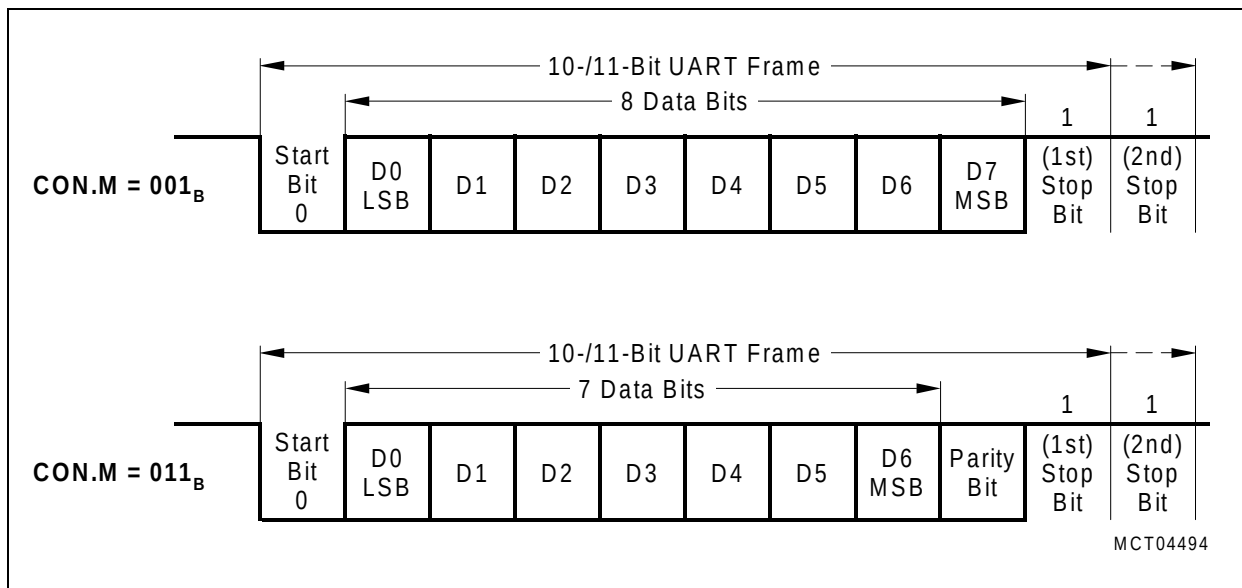


Figure 2-3 Asynchronous 8-Bit Frames

Asynchronous/Synchronous Serial Interface (ASC)

9-Bit Data Frames

9-bit data frames consist of nine data bits D8 ... D0 (CON.M = 100_B), or eight data bits D7 ... D0 plus an automatically generated parity bit (CON.M = 111_B) or eight data bits D7 ... D0 plus wake-up bit (CON.M = 101_B). Parity may be odd or even, depending on bit CON.ODD. An even parity bit will be set if the modulo-2-sum of the eight data bits is 1. An odd parity bit will be cleared in this case. Parity checking is enabled via bit CON.PEN (always OFF in 9-bit data and wake-up mode). The parity error flag CON.PE will be set along with the error interrupt request flag, if a wrong parity bit is received. The parity bit itself will be stored in bit RBUF.8.

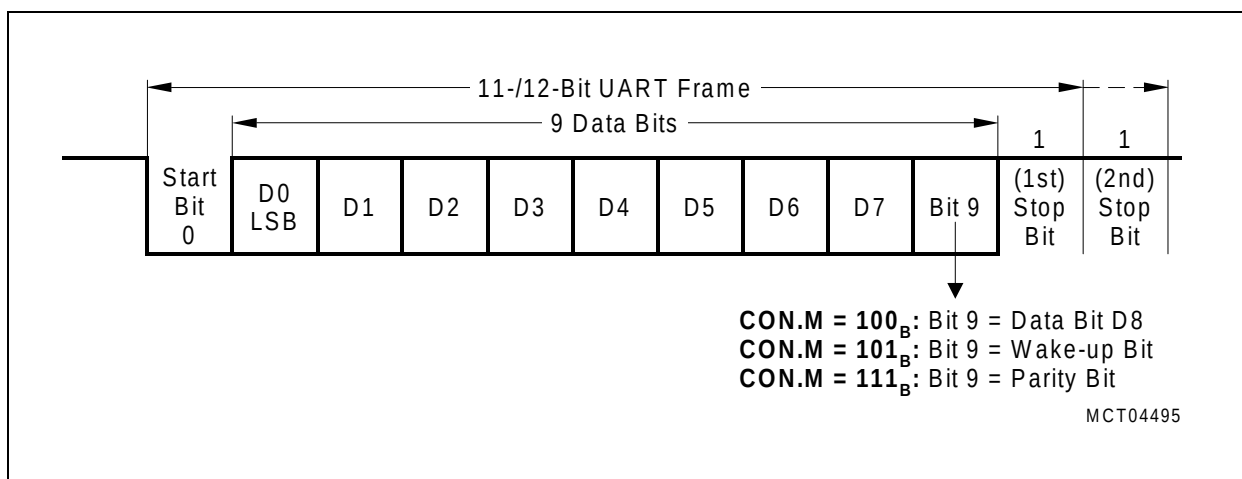


Figure 2-4 Asynchronous 9-Bit Frames

In Wake-up Mode, received frames are transferred to the receive buffer register only if the 9th bit (the wake-up bit) is 1. If this bit is 0, no receive interrupt request will be activated and no data will be transferred.

This feature may be used to control communication in multi-processor systems:

When the master processor wants to transmit a block of data to one of several slaves, it first sends out an address byte that identifies the target slave. An address byte differs from a data byte in that the additional 9th bit is a 1 for an address byte, but is a 0 for a data byte, so, no slave will be interrupted by a data 'byte'. An address 'byte' will interrupt all slaves (operating in 8-bit data + wake-up bit mode), so each slave can examine the eight LSBs of the received character (the address). The addressed slave will switch to 9-bit data mode (for example, by clearing bit CON.M[0]), to enable it to also receive the data bytes that will be coming (having the wake-up bit cleared). The slaves not being addressed remain in 8-bit data + wake-up bit mode, ignoring the following data bytes.

Asynchronous/Synchronous Serial Interface (ASC)

IrDA Frames

The modulation schemes of IrDA are based on standard asynchronous data transmission frames. The asynchronous data format in IrDA mode ($CON.M=010_B$) is defined as follows:

- 1 start bit / 8 data bits / 1 stop bit

The coding/decoding of/to the asynchronous data frames is shown in **Figure 2-5**. In general, during IrDA transmissions, UART frames are encoded into IR frames and vice versa. A low level on the IR frame indicates an “LED off” state. A high level on the IR frame indicates a “LED on” state.

For a “0” bit in the UART frame, a high pulse is generated. For a “1” bit in the UART frame, no pulse is generated. The high pulse starts in the middle of a bit cell and has a fixed width of $3/16$ of the bit time. The ASC also makes it possible to program the length of the IrDA high pulse. Further, the polarity of the received IrDA pulse can be inverted in IrDA mode. **Figure 2-5** shows the non-inverted IrDA pulse scheme.

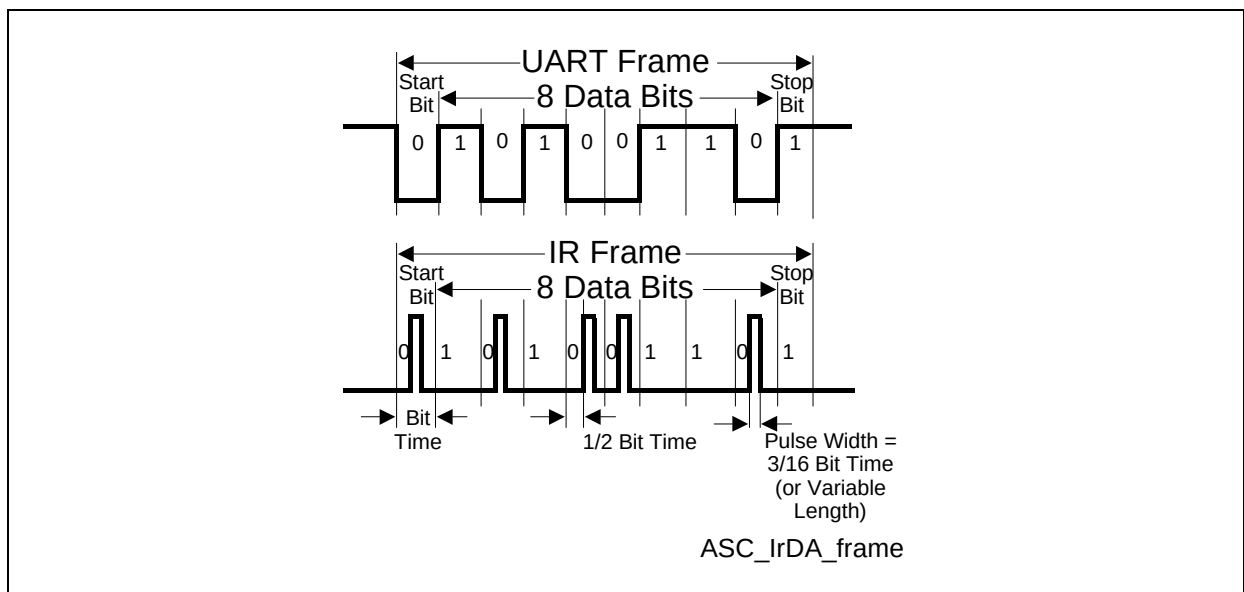


Figure 2-5 IrDA Frame Encoding/Decoding

The ASC IrDA pulse mode/width register PMW contains the 8-bit IrDA pulse width value and the IrDA pulse width mode select bit. This register is required in the IrDA operating mode only.

Asynchronous/Synchronous Serial Interface (ASC)**2.1.3.2 Asynchronous Transmission**

Asynchronous transmission begins at the next overflow of the divide-by-16 baud rate timer (transition of the baud rate clock f_{BR}), if bit CON.R must be set and data has been loaded into TBUF. The transmitted data frame consists of three basic elements:

- Start bit
- Data field (eight or nine bits, LSB first, including a parity bit, if selected)
- Delimiter (one or two stop bits)

Data transmission is double-buffered. When the transmitter is idle, the transmit data loaded into the transmit buffer register is immediately moved to the transmit shift register, thus freeing the transmit buffer for the next data to be sent. This is indicated by the transmit buffer interrupt request line TBIR being activated. TBUF may now be loaded with the next data, while transmission of the previous data continues.

The transmit interrupt request line TIR will be activated before the last bit of a frame is transmitted, that is, before the first or the second stop bit is shifted out of the transmit shift register.

Note: The transmitter output pin TXD must be configured for alternate data output.

Asynchronous/Synchronous Serial Interface (ASC)

2.1.3.3 Transmit FIFO Operation

The transmit FIFO (TXFIFO) provides the following functionality:

- Enable/disable control
- Programmable filling level for transmit interrupt generation
- Filling level indication
- FIFO clear (flush) operation
- FIFO overflow error generation

The 8-stage transmit FIFO is controlled by the TXFCON control register. When bit TXFCON.TXFEN is set, the transmit FIFO is enabled. The interrupt trigger level defined by TXFCON.TXFITL defines the filling level of the TXFIFO at which a transmit buffer interrupt TBIR or a transmit interrupt TIR is generated. These interrupts are always generated when the filling level of the transmit FIFO is equal to or less than the value stored in TXFCON.TXFITL.

Bit field FSTAT.TXFLL in the FIFO status register FSTAT indicates the number of entries that are actually written (valid) in the TXFIFO. Therefore, the software can check, in the interrupt service routine, for instance, how many bytes can be still written into the transmit FIFO via register TBUF without getting an overrun error.

The transmit FIFO cannot be accessed directly. All data write operations into the TXFIFO are executed by writing into the TBUF register.

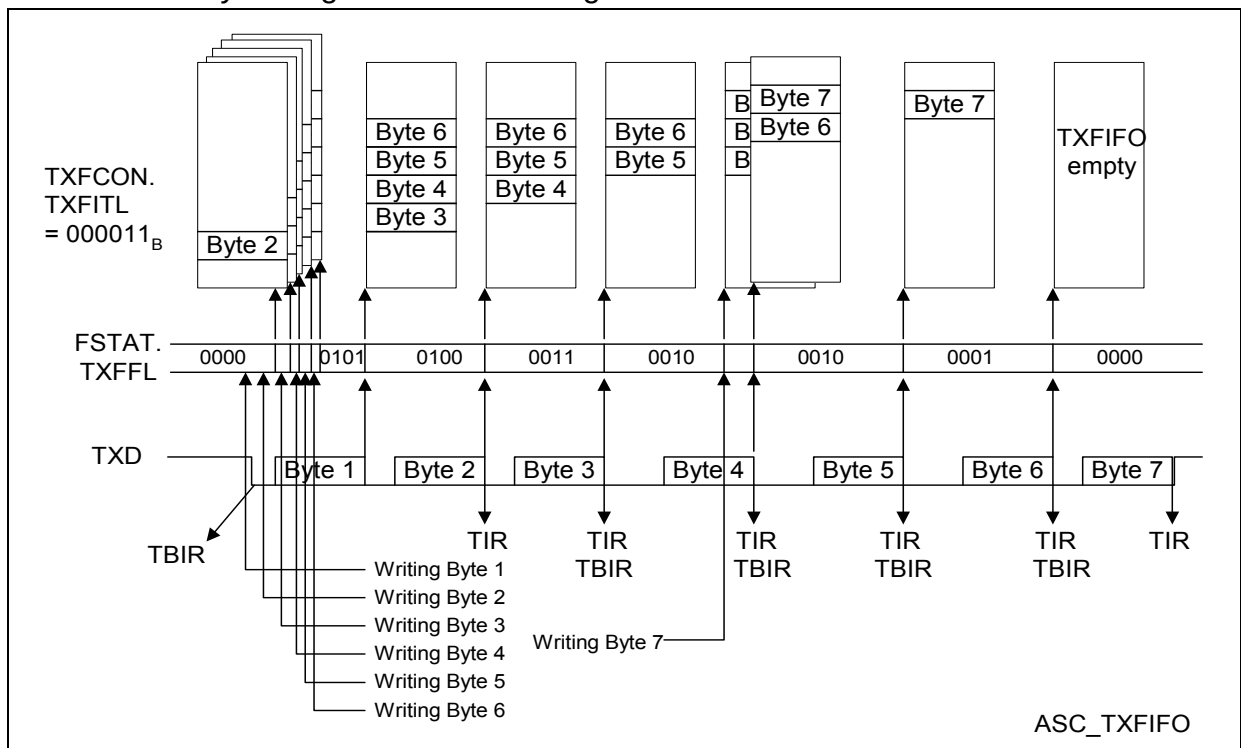


Figure 2-6 Transmit FIFO Operation Example

Asynchronous/Synchronous Serial Interface (ASC)

The example in [Figure 2-6](#) shows a typical 8-stage transmit FIFO operation. In this example, seven bytes are transmitted via the TXD output line. The transmit FIFO interrupt trigger level TXFCON.TXFITL is set to 000011_B. The first byte written into the empty TXFIFO via TBUF is directly transferred into the transmit shift register and is not written into the FIFO. A transmit buffer interrupt will be generated in this case. After byte 1, bytes 2 to 6 are written into the transmit FIFO.

After the transfer of byte 3 from the TXFIFO into the transmit shift register of the ASC, 3 bytes remain in the TXFIFO. Therefore, the value of TXFCON.TXFITL is reached and a transmit buffer interrupt will be generated at the beginning and a transmit interrupt at the end of the byte 3 serial transmission. During the serial transmission of byte 4, another byte (byte 7) is written into the TXFIFO (TBUF write operation). Finally, after the start of the serial transmission of byte 7, the TXFIFO is again empty.

If the TXFIFO is full and additional bytes are written into TBUF, the error interrupt will be generated with bit CON.OE set. In this case, the data byte that was last written into the transmit FIFO is overwritten and the transmit FIFO filling level FSTAT.TXFFL is set to maximum.

The TXFIFO can be flushed or cleared by setting bit TXFCON.TXFFLU in register TXFCON. After this TXFIFO flush operation, the TXFIFO is empty and the transmit FIFO filling level FSTAT.TXFFL is set to 000000_B. A running serial transmission is not aborted by a receive FIFO flush operation.

Note: The TXFIFO is flushed automatically with a reset operation of the ASC module and if the TXFIFO becomes disabled (resetting bit TXFCON.TXFEN) after it was previously enabled.

Asynchronous/Synchronous Serial Interface (ASC)

2.1.3.4 Asynchronous Reception

Asynchronous reception is initiated by a falling edge (1-to-0 transition) on pin RXD, provided that bits CON.R and CON.REN are set. The receive data input pin RXD is sampled at 16 times the rate of the selected baud rate. A majority decision of the 7th, 8th and 9th sample determines the effective bit value. This avoids erroneous results that may be caused by noise.

If the detected value is not a 0 when the start bit is sampled, the receive circuit is reset and waits for the next 1-to-0 transition at pin RXD. If the start bit proves valid, the receive circuit continues sampling and shifts the incoming data frame into the receive shift register.

When the last stop bit has been received, the contents of the receive shift register are transferred to the receive data buffer register RBUF. Simultaneously, the receive interrupt request line RIR is activated after the 9th sample in the last stop bit timeslot (as programmed), regardless whether valid stop bits have been received or not. The receive circuit then waits for the next start bit (1-to-0 transition) at the receive data input line.

Note: The receive input pin RXD must be configured as input mode.

Asynchronous reception is stopped by clearing bit CON.REN. A currently received frame is completed including generation of the receive interrupt request and an error interrupt request, if appropriate. Start bits that follow this frame will not be recognized.

Note: In wake-up mode received frames are transferred to the receive buffer register only if the 9th bit (the wake-up bit) is 1. If this bit is 0, no receive interrupt request will be activated and no data will be transferred.

2.1.3.5 Receive FIFO Operation

The receive FIFO (RXFIFO) provides the following functionality:

- Enable/disable control
- Programmable filling level for receive interrupt generation
- Filling level indication
- FIFO clear (flush) operation
- FIFO overflow error generation

The 8-stage receive FIFO is controlled by the RXFCON control register. When bit RXFCON.RXFEN is set, the receive FIFO is enabled. The interrupt trigger level defined by RXFCON.RXFITL defines the filling level of RXFIFO at which a receive interrupt RIR is generated. RIR is always generated when the filling level of the receive FIFO is equal to or greater than the value stored in RXFCON.RXFITL.

Bit field FSTAT.RXFLL in the FIFO status register FSTAT indicates the number of bytes that have been actually written into the FIFO and can be read out of the FIFO by a user program.

Asynchronous/Synchronous Serial Interface (ASC)

The receive FIFO cannot be accessed directly. All data read operations from the RXFIFO are executed by reading the RBUF register.

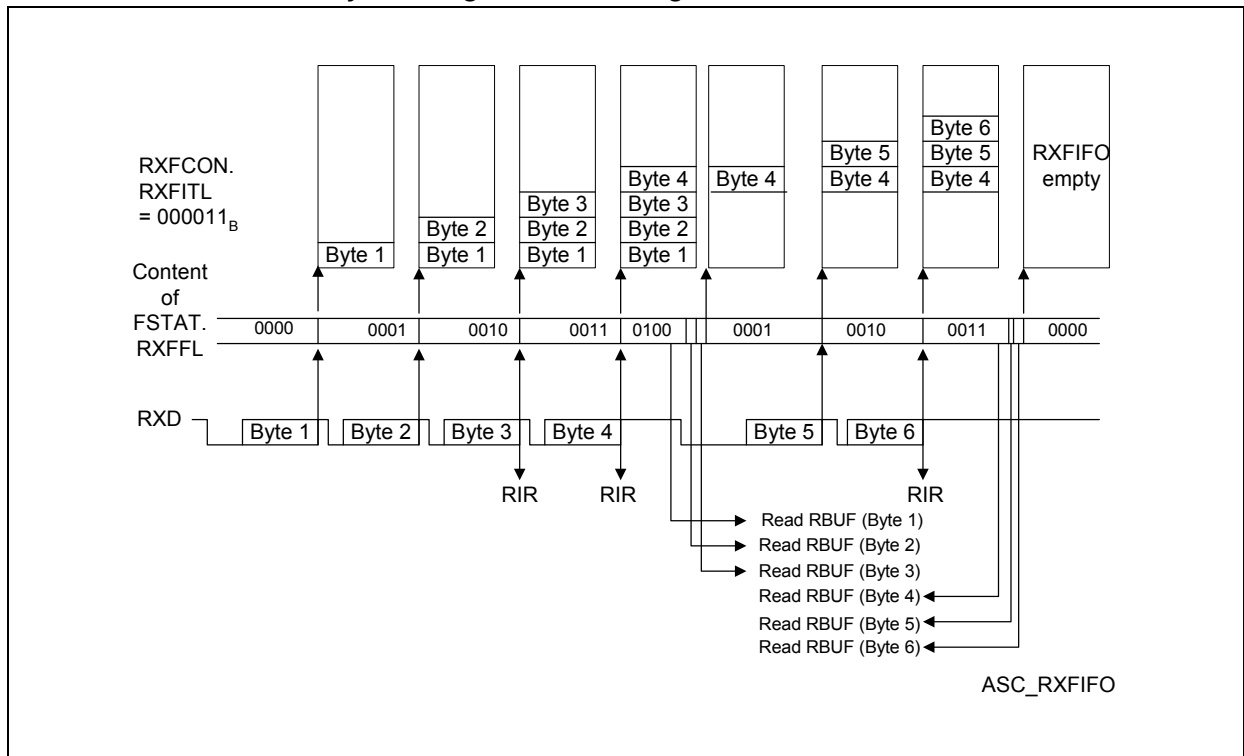


Figure 2-7 Receive FIFO Operation Example

The example in [Figure 2-7](#) shows a typical 8-stage receive FIFO operation. In this example, six bytes are received via the RXD input line. The receive FIFO interrupt trigger level RXFCON.RXFITL is set to 000011_B. Therefore, the first receive interrupt RIR is generated after the reception of byte 3 (RXFIFO is filled with three bytes).

After the reception of byte 4, three bytes are read out of the receive FIFO. After this read operation, the RXFIFO still contains one byte. RIR becomes again active after two more bytes (byte 5 and 6) have been received (RXFIFO filled again with 3 bytes). Finally, the FIFO is cleared after three read operation.

If the RXFIFO is full and additional bytes are received, the receive interrupt RIR and the error interrupt EIR will be generated with bit CON.OE set. In this case, the data byte last written into the receive FIFO is overwritten. With the overrun condition, the receive FIFO filling level FSTAT.RXFFL is set to maximum. If a RBUF read operation is executed with the RXFIFO enabled but empty, an error interrupt EIR will be generated as well with bit CON.OE set. In this case, the receive FIFO filling level FSTAT.RXFFL is set to 000000_B.

If the RXFIFO is available but disabled (RXFCON.RXFEN=0) and the receive operation is enabled (CON.REN=1), the asynchronous receive operation is functionally equivalent to the asynchronous receive operation of the ASC module.

Asynchronous/Synchronous Serial Interface (ASC)

The RXFIFO can be flushed or cleared by setting bit RXFCON.RXFFLU in register RXFCON. After this RXFIFO flush operation, the RXFIFO is empty and the receive FIFO filling level FSTAT.RXFFL is set to 000000_B.

The RXFIFO is flushed automatically with a reset operation of the ASC module and if the RXFIFO becomes disabled (resetting bit RXFCON.RXFEN) after it was previously enabled. Resetting bit CON.REN without resetting RXFCON.RXFEN does not affect (reset) the RXFIFO state. This means that the receive operation of the ASC is stopped, in this case, without changing the content of the RXFIFO. After setting CON.REN again, the RXFIFO with its content is again available.

2.1.3.6 FIFO Transparent Mode

In Transparent Mode, a specific interrupt generation mechanism is used for receive and transmit buffer interrupts. In general, in Transparent Mode, receive interrupts are always generated if data bytes are available in the RXFIFO. Transmit buffer interrupts are always generated if the TXFIFO is not full. The relevant conditions for interrupt generation in Transparent Mode are:

- FIFO filling levels
- Read/write operations on the RBUF/TBUF data register

Interrupt generation for the receive FIFO depends on the RXFIFO filling level and the execution of read operations of register RBUF (see [Figure 2-8](#)). Transparent Mode for the RXFIFO is enabled when bits RXFCON.RXTMEN and RXFCON.RXFEN in register RXFCON are set.

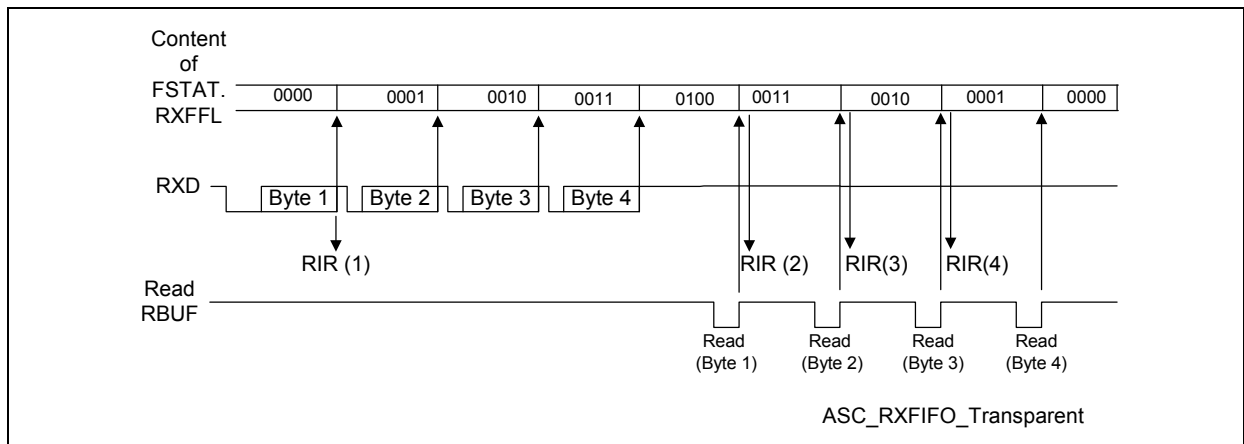


Figure 2-8 Transparent Mode Receive FIFO Operation

If the RXFIFO is empty, a receive interrupt RIR is always generated when the first byte is written into an empty RXFIFO (FSTAT.RXFFL changes from 000000_B to 000001_B). If the RXFIFO is filled with at least one byte, the occurrence of further receive interrupts depends on the read operations of register RBUF. The receive interrupt RIR will always be activated after a RBUF read operation if the RXFIFO still contains data.

Asynchronous/Synchronous Serial Interface (ASC)

(FSTAT.RXFFL is not equal to 0000_B). If the RXFIFO is empty after a RBUF read operation, no further receive interrupt will be generated.

If the RXFIFO is full (FSTAT.RXFFL=1000_B) and additional bytes are received, an error interrupt EIR will be generated with bit CON.OE set. In this case, the data byte last written into the receive FIFO is overwritten. If a RBUF read operation is executed with the RXFIFO enabled but empty (underflow condition), an error interrupt EIR will be generated as well, with bit CON.OE set.

If the RXFIFO is flushed in Transparent Mode, the software must take care that a previous pending receive interrupt is ignored.

Note: The Receive FIFO Interrupt Trigger Level bit field RXFCON.RXFITL is not applicable in Transparent Mode.

Interrupt generation for the transmit FIFO depends on the TXFIFO filling level and the execution of write operations to the register TBUF. Transparent Mode for the TXFIFO is enabled when bits TXFCON.TXTMEN and TXFCON.TXFEN are set.

A transmit buffer interrupt TBIR is always generated when the TXFIFO is not full (FSTAT.TXFFL not equal to 1000_B) after a byte has been written into register TBUF. TBIR is also activated after a TXFIFO flush operation or when the TXFIFO becomes enabled (TXFCON.TXTMEN and TXFCON.TXFEN set) when it was previously disabled. In these cases, the TXFIFO is empty and ready to be filled with data.

If the TXFIFO is full (FSTAT.TXFFL=maximum) and an additional byte is written into TBUF, no further transmit buffer interrupt will be generated after the TBUF write operation. In this case the data byte last written into the transmit FIFO is overwritten and an overrun error interrupt (EIR) will be generated with bit CON.OE set.

Note: The Transmit FIFO Interrupt Trigger Level bit field TXFCON.TXFITL is not applicable in Transparent Mode.

2.1.3.7 IrDA Mode

The duration of the IrDA pulse is normally 3/16 of a bit period. The IrDA standard also allows the pulse duration to be independent of the baud rate or bit period. In this case, the transmitted pulse always has the width corresponding to the 3/16 pulse width at 115.2 kBaud, which is 1.627 μs. Either bit period depended or fixed IrDA pulse width generation can be selected. The IrDA pulse width mode is selected by bit PMW.IRPW.

In case of fixed IrDA pulse width generation, the lower eight bits in register PMW are used to adapt the IrDA pulse width to a fixed value such as 1.627 μs. The fixed IrDA pulse width is generated by a programmable timer as shown in [Figure 2-9](#).

Asynchronous/Synchronous Serial Interface (ASC)

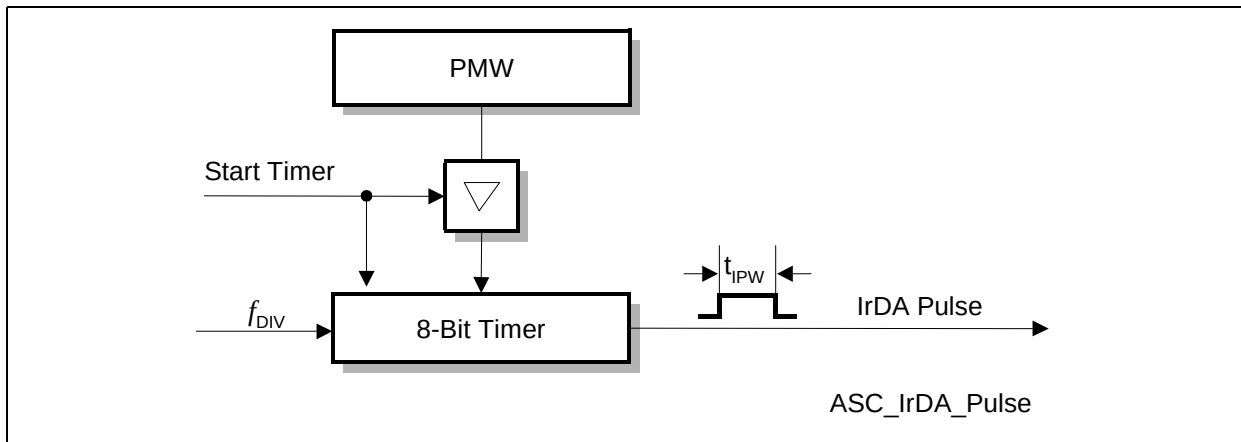


Figure 2-9 Fixed IrDA Pulse Generation

The IrDA pulse width can be calculated according to the formulas given in [Table 2-1](#).

Table 2-1 Formulas for the IrDA Pulse Width Calculation

PMW	PMW.IRPW	Formulas	
1 ... 255	0	$t_{IPW} = \frac{3}{16 \times \text{Baud rate}}$	$t_{IPW \min} = \frac{(\text{PMW} \gg 1)}{f_{ASC}}$
	1	$t_{IPW} = \frac{\text{PMW}}{f_{ASC}}$	

Note: The name PMW in the formulas of [Table 2-1](#) represents the content of the reload register PMW (PW_VALUE), taken as unsigned 8-bit integer.

The contents of PMW.PW_VALUE further define the minimum IrDA pulse width ($t_{IPW \min}$) that is still recognized as a valid IrDA pulse during a receive operation. This function is independent of the selected IrDA pulse width mode (fixed or variable) which is defined by bit PMW.IRPW. The minimum IrDA pulse width is calculated by a shift right operation of PMW bit 7-0 by one bit divided by the module clock f_{ASC} .

Note: If PMW.IRPW=0 (fixed IrDA pulse width), PMW.PW_VALUE must be a value which assures that $t_{IPW} > t_{IPW \min}$.

Note: PMW value has to be carefully selected to guarantee the IrDA pulse is valid. Especially, in some transmission baud rates, the pulse width cannot be too small to be recognized.

[Table 2-2](#) gives some examples for typical frequencies of f_{ASC} .

Asynchronous/Synchronous Serial Interface (ASC)

Table 2-2 IrDA Pulse Width Adaptation to 1.627 μ s @115.2K Baud Rate

f_{ASC}	PMW	t_{IPW}	Error	$t_{IPW \min}$
25 MHz	41 _D	1.64 μ s	+ 0.8 %	0.82 μ s
40 MHz	65 _D	1.625 μ s	- 0.12 %	0.81 μ s
48 MHz	78 _D	1.625 μ s	- 0.12 %	0.81 μ s
50 MHz	81 _D	1.62 μ s	- 0.4 %	0.81 μ s
75 MHz	122 _D	1.6267 μ s	- 0.02 %	0.813 μ s

2.1.3.8 RXD/TXD Data Path Selection in Asynchronous Modes

The data paths for the serial input and output data of the Asynchronous mode are affected by control bits in the register CON as shown in [Figure 2-10](#). The synchronous mode operation is not affected by these data path selection capabilities.

Two multiplexers are in the RXD input signal path for providing the loopback mode capability (controlled by bit CON.LB) and the IrDA receive pulse inversion capability (controlled by bit CON.RXDI). Depending on the asynchronous operating mode (controlled by bit field CON.M), the ASC output signal or the IrDA coded ASC output signal is switched to the TXD output via a multiplexer.

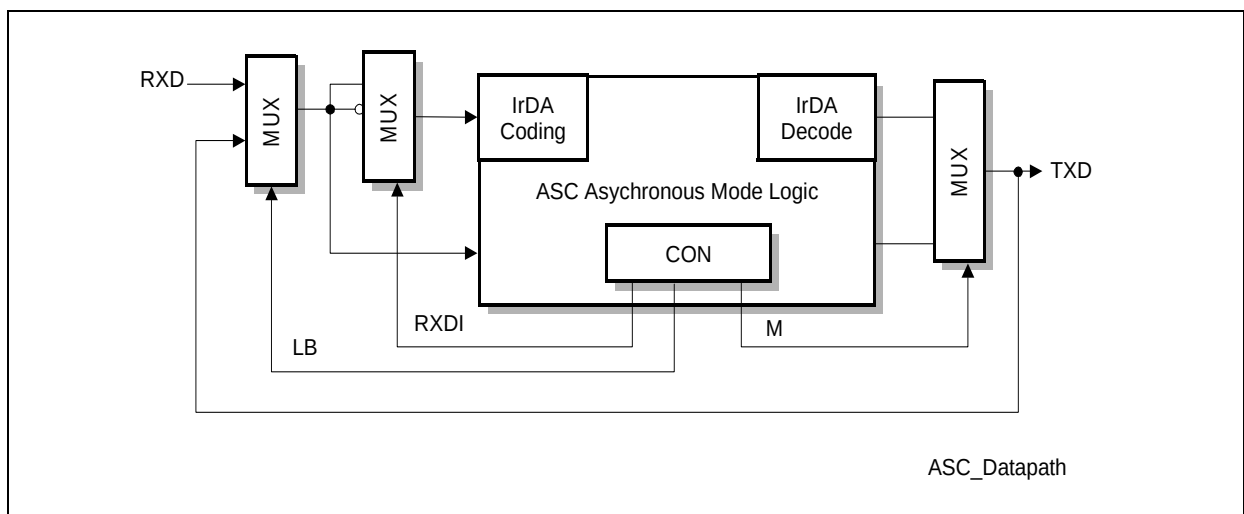


Figure 2-10 RXD/TXD Data Path in Asynchronous Modes

Asynchronous/Synchronous Serial Interface (ASC)

2.1.4 Synchronous Operation

Synchronous Mode supports half-duplex communication, basically for simple I/O expansion via shift registers. Data is transmitted and received via pin RXD while line TXD outputs the shift clock. Synchronous mode is selected with $CON.M = 000_B$.

Eight data bits are transmitted or received synchronous to a shift clock generated by the internal baud rate generator. The shift clock is active only as long as data bits are transmitted or received.

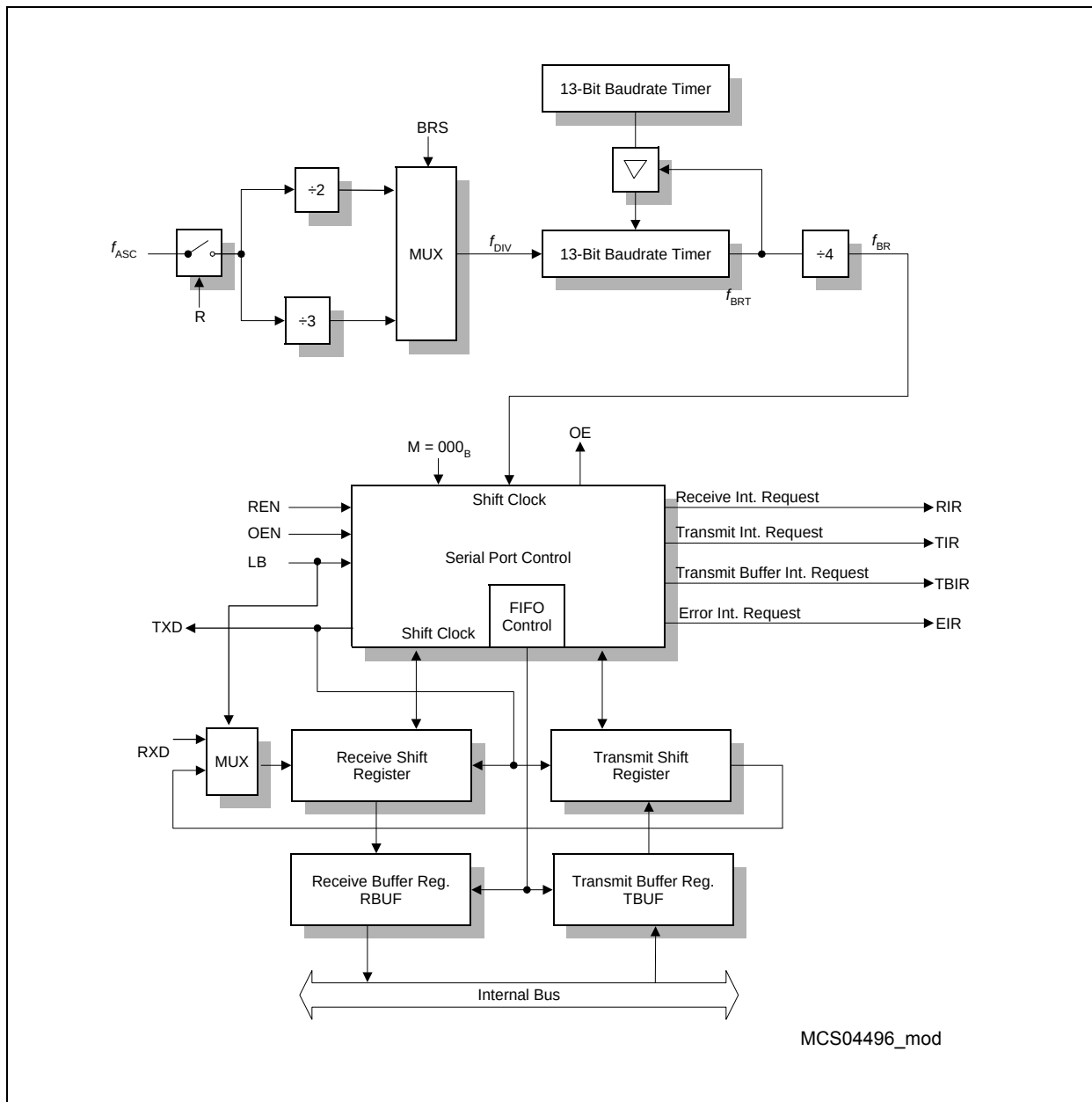


Figure 2-11 Synchronous Mode of Serial Channel ASC

Asynchronous/Synchronous Serial Interface (ASC)

2.1.4.1 Synchronous Transmission

Synchronous transmission begins within four state times after data has been loaded into TBUF, provided that CON.R is set and CON.REN = 0 (half-duplex, no reception). Exception: in Loop-back Mode (bit CON.LB set), CON.REN must be set for reception of the transmitted byte. Data transmission is double buffered. When the transmitter is idle, the transmit data loaded into TBUF is immediately moved to the transmit shift register thus freeing TBUF for more data. This is indicated by the transmit buffer interrupt request line TBIR being activated. TBUF may now be loaded with the next data, while transmission of the previous one continues. The data bits are transmitted synchronous with the shift clock. After the bit time for the 8th data bit, both TXD and RXD will go high, the transmit interrupt request line TIR is activated, and serial data transmission stops.

Pin TXD must be configured for alternate data output in order to provide the shift clock. Pin RXD must also be configured for output during transmission.

2.1.4.2 Synchronous Reception

Synchronous reception is initiated by setting bit CON.REN = 1. If bit CON.R = 1, the data applied at RXD is clocked into the receive shift register synchronous to the clock that is output at pin TXD. After the 8th bit has been shifted in, the contents of the receive shift register are transferred to the receive data buffer RBUF, the receive interrupt request line RIR is activated, the receiver enable bit CON.REN is reset, and serial data reception stops.

Pin TXD must be configured for alternate data output in order to provide the shift clock. Pin RXD must be configured as alternate data input.

Synchronous reception is stopped by clearing bit CON.REN. A currently received byte is completed, including the generation of the receive interrupt request and an error interrupt request, if appropriate. Writing to the transmit buffer register while a reception is in progress has no effect on reception and will not start a transmission.

If a previously received byte has not been read out of the receive buffer register by the time the reception of the next byte is complete, both the error interrupt request line EIR and the overrun error status flag CON.OE will be activated/set, provided that the overrun check has been enabled by bit CON.OEN.

2.1.4.3 Synchronous Timing

Figure 2-12 shows timing diagrams of the ASC Synchronous Mode data reception and data transmission. In Idle State the shift clock is at high level. With the beginning of a synchronous transmission of a data byte, the data is shifted out at RXD with the falling edge of the shift clock. If a data byte is received through RXD, data is latched with the rising edge of the shift clock.

One shift clock cycle (f_{BR}) delay is inserted between two consecutive receive or transmit data bytes.

Asynchronous/Synchronous Serial Interface (ASC)

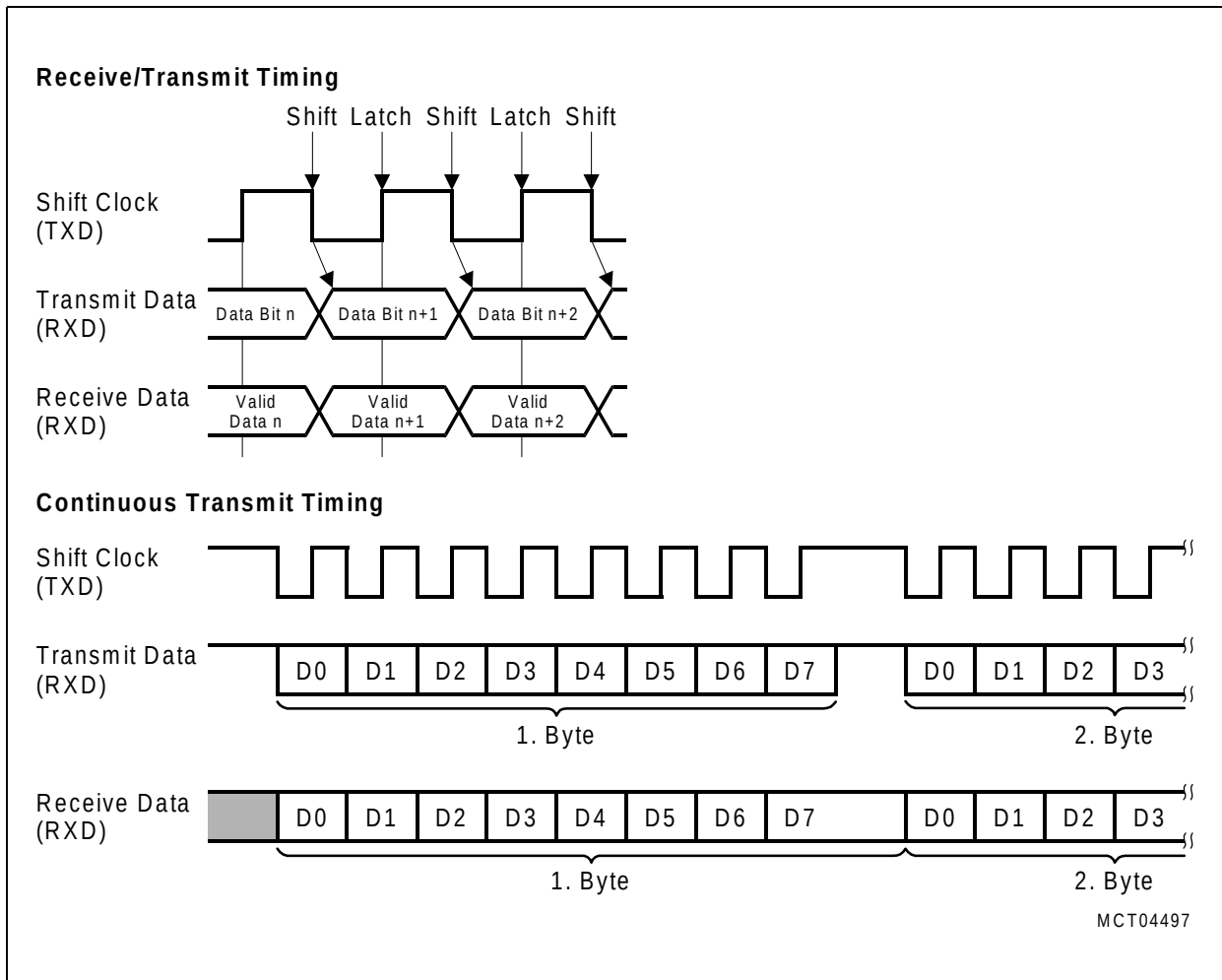


Figure 2-12 ASC Synchronous Mode Waveforms

2.1.5 Baud Rate Generation

The serial channel ASC has its own dedicated 13-bit baud rate generator with reload capability, allowing baud rate generation independent of other timers.

The baud rate generator is clocked with a clock (f_{DIV}) derived via a prescaler from the ASC input clock f_{ASC} . The baud rate timer is counting downwards and can be started or stopped through the baud rate generator run bit CON.R. Each underflow of the timer provides one clock pulse to the serial channel. The timer is reloaded with the value stored in its 13-bit reload register each time it underflows. The resulting clock f_{BRT} is again divided by a factor for the baud rate clock (± 16 in asynchronous modes and ± 4 in synchronous mode). The prescaler is selected by the bits CON.BRS and CON.FDE. In addition to the two fixed dividers, a fractional divider prescaler unit is available in the Asynchronous Modes that allows selection of prescaler divider ratios of $n/512$ with $n = 0-511$. Therefore, the baud rate of ASC is determined by the module clock, the content of FDV, the reload value of BG, and the operating mode (asynchronous or synchronous).

Asynchronous/Synchronous Serial Interface (ASC)

Register BG is the dual-function Baud Rate Generator/Reload register. Reading BG returns the contents of the timer BR_VALUE (bits 15 ... 13 return zero), while writing to BG always updates the reload register (bits 15 ... 13 are insignificant).

An auto-reload of the timer with the contents of the reload register is performed each time BG is written to. However, if CON.R = 0 at the time the write operation to BG is performed, the timer will not be reloaded until the first instruction cycle after CON.R = 1. For a clean baud rate initialization BG should only be written if CON.R = 0. If BG is written with CON.R = 1, an unpredicted behavior of the ASC may occur during running transmit or receive operations.

2.1.5.1 Baud Rate in Asynchronous Mode

For asynchronous operation, the baud rate generator provides a clock f_{BRT} with sixteen times the rate of the established baud rate. Every received bit is sampled at the 7th, 8th and 9th cycle of this clock. The clock divider circuitry, which generates the input clock for the 13-bit baud rate timer, is extended by a fractional divider circuitry that allows the adjustment of more accurate baud rates and the extension of the baud rate range.

The baud rate of the baud rate generator depends on the settings of the following bits and register values:

- Input clock f_{ASC}
- Selection of the baud rate timer input clock f_{DIV} by bits CON.FDE and CON.BRS
- If bit CON.FDE = 1 (fractional divider): value of register FDV
- Value of the 13-bit reload register BG

The output clock of the baud rate timer with the reload register is the sample clock in the asynchronous modes of the ASC. For baud rate calculations, this baud rate clock f_{BR} is derived from the sample clock f_{BRT} by a division by sixteen.

Asynchronous/Synchronous Serial Interface (ASC)

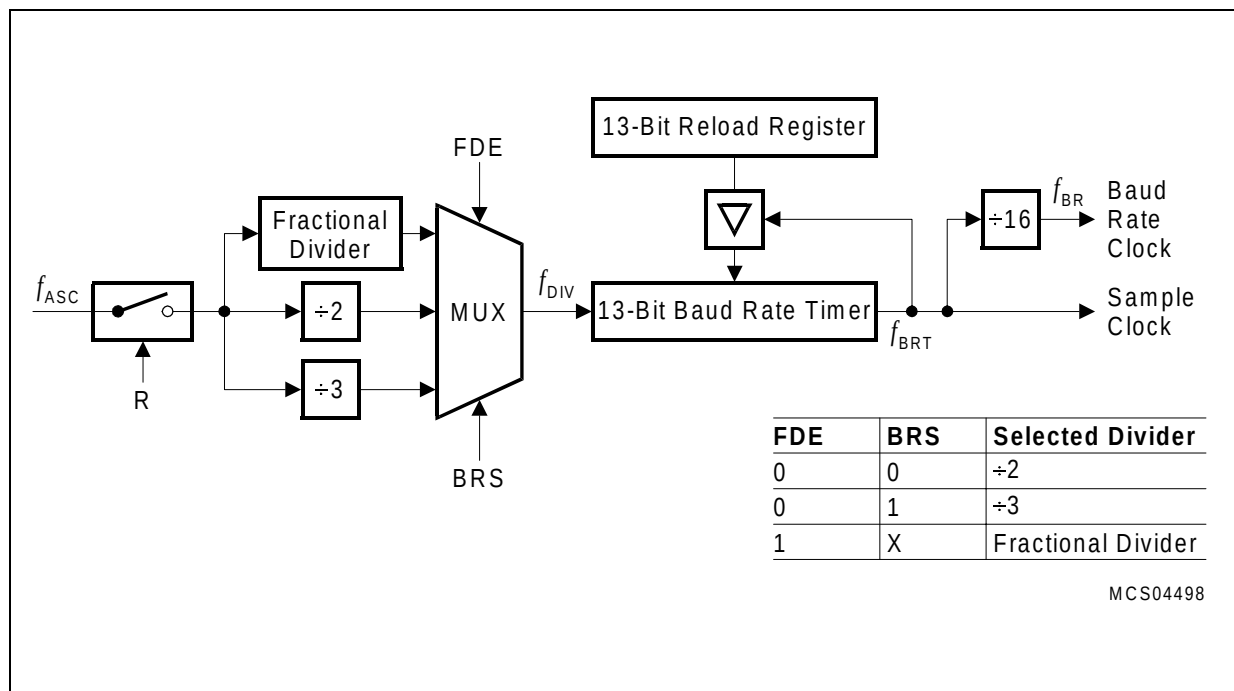


Figure 2-13 ASC Baud Rate Generator Circuitry in Asynchronous Modes

Using the Fixed Input Clock Divider

The baud rate for asynchronous operation of the serial channel ASC when using the fixed input clock divider ratios (CON.FDE = 0) and the required reload value for a given baud rate can be determined by the following formulas:

Table 2-3 Asynchronous Baud Rate Formulas using the Fixed Input Clock Dividers

FDE	BRS	BG	Formula
0	0	0 ... 8191	$\text{Baud rate} = \frac{f_{\text{ASC}}}{32 \times (\text{BG} + 1)}$ $\text{BG} = \frac{f_{\text{ASC}}}{32 \times \text{Baud rate}} - 1$
	1		$\text{Baud rate} = \frac{f_{\text{ASC}}}{48 \times (\text{BG} + 1)}$ $\text{BG} = \frac{f_{\text{ASC}}}{48 \times \text{Baud rate}} - 1$

BG represents the content of the reload register BG (BR_VALUE), taken as unsigned 13-bit integer.

Asynchronous/Synchronous Serial Interface (ASC)

The maximum baud rate that can be achieved for the asynchronous modes when using the two fixed clock dividers and a module clock of 75 MHz is 2.34375 MBaud. The table below lists various commonly used baud rates together with the required reload values and the deviation errors compared to the intended baud rate.

Table 2-4 Typical Asynchronous Baud Rates using Fixed Input Clock Dividers

Baud Rate	BRS = 0, $f_{ASC} = 75 \text{ MHz}$		BRS = 1, $f_{ASC} = 75 \text{ MHz}$	
	Deviation Error	Reload Value	Deviation Error	Reload Value
2.34375 MBaud	–	0000 _H	NA	NA
1.5625 MBaud	–	–	NA	0000 _H
19.2 KBaud	+0.0%	0079 _H	+0.5% / -0.7%	0050 _H / 0051 _H
9600 Baud	+0.0% / -0.3%	00F3 _H / 00F4 _H	+0.4% / -0.1%	00A1 _H / 00A2 _H
4800 Baud	+0.0% / -0.1%	01E7 _H / 01E8 _H	+0.1% / -0.1%	0144 _H / 0145 _H
2400 Baud	+0.0% / -0.0%	03CF _H / 03D0 _H	+0.0% / -0.1%	028B _H / 028C _H
1200 Baud	+0.0% / -0.5%	07A0 _H / 07A1 _H	+0.0% / -0.0%	0515 _H / 0516 _H

Note: CON.FDE must be 0 to achieve the baud rates in the table above. The deviation errors given in the table above are rounded. Using a baud rate crystal will provide correct baud rates without deviation errors.

Using the Fractional Divider

When the fractional divider is selected, the input clock f_{DIV} for the baud rate timer is derived from the module clock f_{ASC} by a programmable divider. If CON.FDE = 1, the fractional divider is activated. It divides f_{ASC} by a fraction of $n/512$ for any value of n from 0 to 511. If $n = 0$, the divider ratio is 1, which means that $f_{DIV} = f_{ASC}$. In general, the fractional divider allows the baud rate to be programmed with a much better accuracy than with the two fixed prescaler divider stages.

Table 2-5 Asynchronous Baud Rate Formulas using the Fractional Input Clock Divider

FDE	BRS	BG	FDV	Formula
1	–	0 ... 8191	1 ... 511	Baud rate = $\frac{FDV}{512} \times \frac{f_{ASC}}{16 \times (BG + 1)}$
			0	Baud rate = $\frac{f_{ASC}}{16 \times (BG + 1)}$

Asynchronous/Synchronous Serial Interface (ASC)

Note: BG represents the contents of the reload register BG (BR_VALUE), taken as an unsigned 13-bit integer. FDV represents the contents of the fractional divider register (FD_VALUE) taken as an unsigned 9-bit integer.

Table 2-6 Typical Asynchronous Baud Rates using the Fractional Input Clock Divider

f_{ASC}	Desired Baud Rate	BG	FDV	Resulting Baud Rate	Deviation
25 MHz	115.2 kBaud	7	302	115.204 kBaud	< 0.01%
	57.6 kBaud	15	302	57.602 kBaud	< 0.01%
	38.4 kBaud	23	302	39.401 kBaud	< 0.01%
	19.2 kBaud	47	302	19.201 kBaud	< 0.01%
50 MHz	115.2 kBaud	15	302	115.203 kBaud	0.00%
	57.6 kBaud	15	151	57.601 kBaud	0.00%
	38.4 kBaud	23	151	38.401 kBaud	0.00%
	19.2 kBaud	47	151	19.200 kBaud	0.00%
75 MHz	115.2 kBaud	23	302	115.204 kBaud	< 0.01%
	57.6 kBaud	47	302	57.602 kBaud	< 0.01%
	38.4 kBaud	71	302	39.401 kBaud	< 0.01%
	19.2 kBaud	95	302	19.201 kBaud	< 0.01%

2.1.5.2 Baud Rate in Synchronous Mode

For synchronous operation, the baud rate generator provides a clock with four times the rate of the established baud rate (see [Figure 2-14](#)).

Asynchronous/Synchronous Serial Interface (ASC)

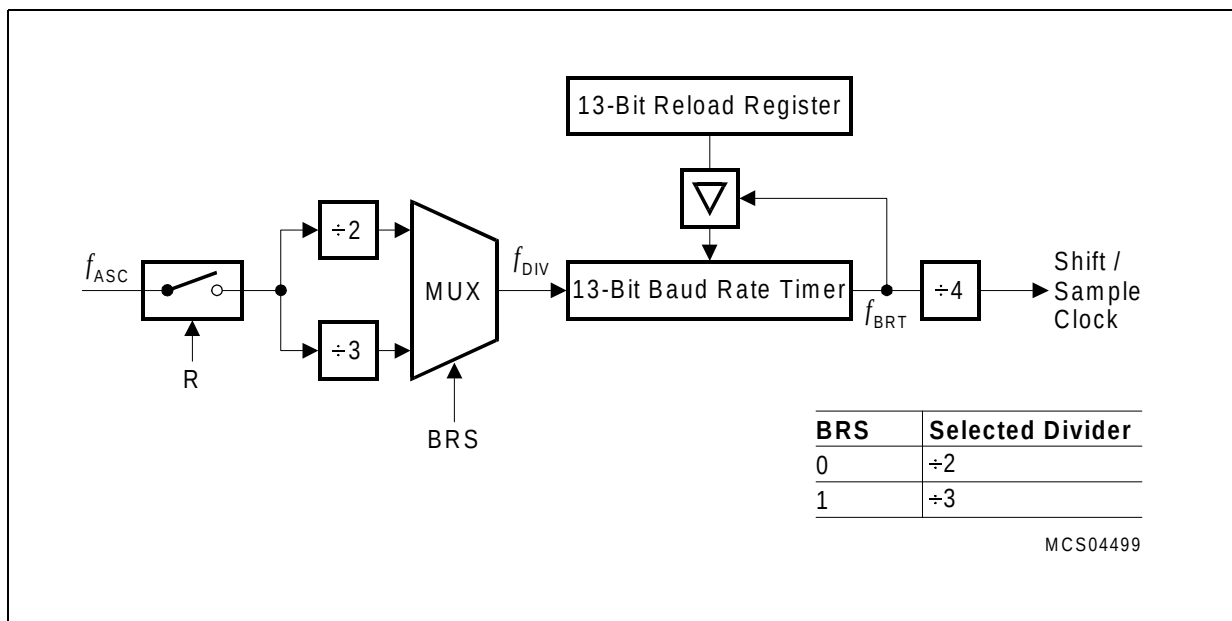


Figure 2-14 ASC Baud Rate Generator Circuitry in Synchronous Mode

The baud rate for synchronous operation of serial channel ASC can be determined by the formulas as shown in [Table 2-7](#).

Table 2-7 Synchronous Baud Rate Formulas

BRS	BG	Formula
0	0 ... 8191	$\text{Baud rate} = \frac{f_{\text{ASC}}}{8 \times (\text{BG} + 1)} \quad \text{BG} = \frac{f_{\text{ASC}}}{8 \times \text{Baud rate}} - 1$
1		$\text{Baud rate} = \frac{f_{\text{ASC}}}{12 \times (\text{BG} + 1)} \quad \text{BG} = \frac{f_{\text{ASC}}}{12 \times \text{Baud rate}} - 1$

Note: BG represents the contents of the reload register (BR_VALUE), taken as unsigned 13-bit integers.

The maximum baud rate that can be achieved in Synchronous Mode when using a module clock of 75 MHz is 9.375 MBaud.

2.1.6 Hardware Error Detection Capabilities

To improve the safety of serial data exchange, the serial channel ASC provides an error interrupt request flag that indicates the presence of an error and three (selectable) error status flags in register CON that indicate which error has been detected during reception. Upon completion of a reception, the error interrupt request line EIR will be activated

Asynchronous/Synchronous Serial Interface (ASC)

simultaneously with the receive interrupt request line RIR, if one or more of the following conditions are met:

- If the framing error detection enable bit CON.FEN is set and any of the expected stop bits is not high, the framing error flag CON.FE is set, indicating that the error interrupt request is due to a framing error (Asynchronous Mode only).
- If the parity error detection enable bit CON.PEN is set in the modes where a parity bit is received and the parity check on the received data bits proves false, the parity error flag CON.PE is set, indicating that the error interrupt request is due to a parity error (Asynchronous Mode only).
- If the overrun error detection enable bit CON.OEN is set and the last character received was not read out of the receive buffer by software or DMA transfer at the time the reception of a new frame is complete, the overrun error flag CON.OE is set indicating that the error interrupt request is due to an overrun error (asynchronous and synchronous mode).

Asynchronous/Synchronous Serial Interface (ASC)

2.1.7 Interrupts

Four interrupt sources are provided for serial channel ASC. Line TIR indicates a transmit interrupt, TBIR indicates a transmit buffer interrupt, RIR indicates a receive interrupt, and EIR indicates an error interrupt of the serial channel. The interrupt output lines TBIR, TIR, RIR, and EIR are activated (active state) for two periods of the module clock f_{ASC} . The interrupt control unit provides interrupt request flags that are set when these interrupt output lines are activated.

The cause of an error interrupt request EIR (framing, parity, overrun error) can be identified by the error status flags FE, PE, and OE located in control register CON.

Note: In contrary to the error interrupt request line EIR, the error status flags FE/PE/OE are not reset automatically but must be cleared by software.

For normal operation (that is, other than error interrupt) the ASC provides three interrupt requests to control data exchange via this serial channel:

- TBIR is activated when data is moved from TBUF to the transmit shift register.
- TIR is activated before the last bit of an asynchronous frame is transmitted, or after the last bit of a synchronous frame has been transmitted.
- RIR is activated when the received frame is moved to RBUF.

Note: While the receive task is handled by a single interrupt handler, the transmitter is serviced by two interrupt handlers. This provides advantages for the servicing software.

For single transfers it is sufficient to use the transmitter interrupt (TIR), which indicates that the previously loaded data has been transmitted, except for the last bit of an asynchronous frame.

For multiple back-to-back transfers it is necessary to load the following piece of data at last until the time the last bit of the previous frame has been transmitted. In Asynchronous Mode, this leaves just one bit-time for the handler to respond to the transmitter interrupt request; in Synchronous Mode, it is entirely impossible.

Using the Transmit Buffer Interrupt (TBIR) to reload transmit data gives the time to transmit a complete frame for the service routine, as TBUF may be reloaded while the previous data is still being transmitted.

Asynchronous/Synchronous Serial Interface (ASC)

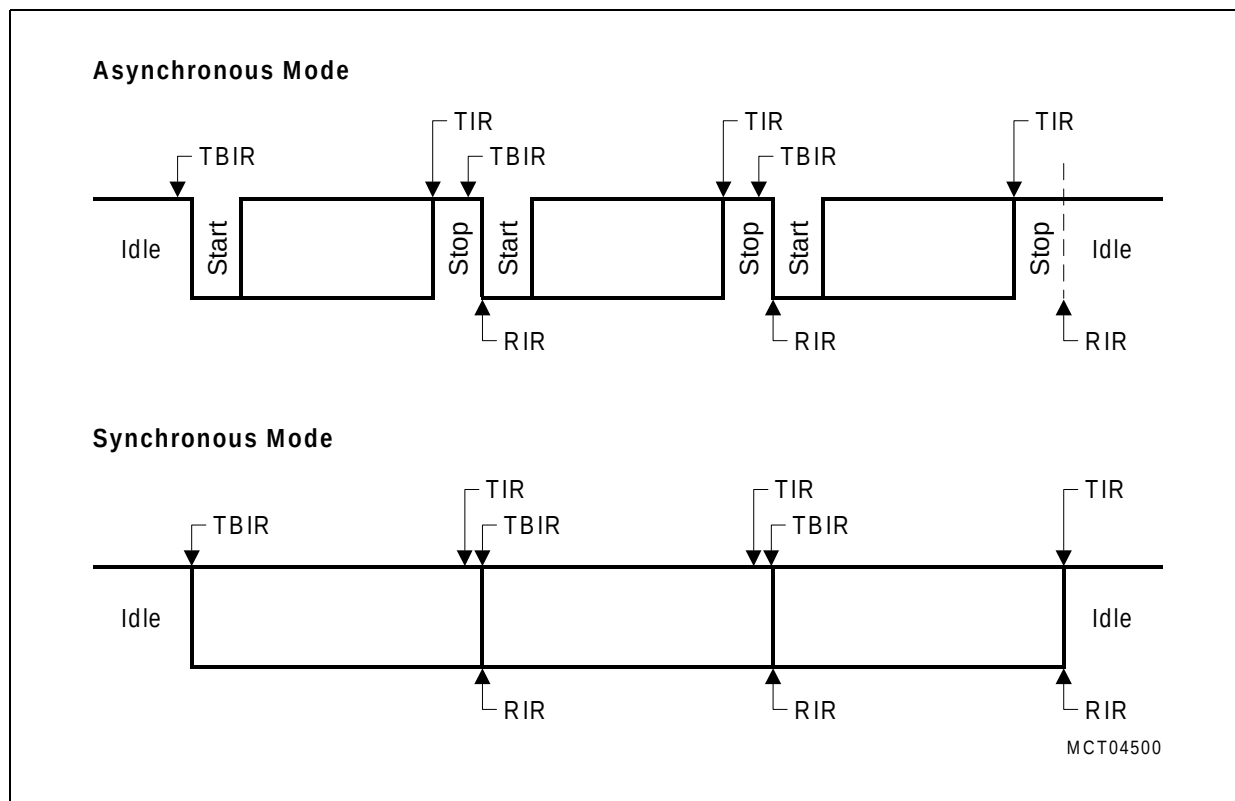


Figure 2-15 ASC Interrupt Generation

As shown in [Figure 2-15](#) above, TBIR is an early trigger for the reload routine, while TIR indicates the completed transmission. Therefore, software using handshake should rely on TIR at the end of a data block to ensure that all data has actually been transmitted.

Asynchronous/Synchronous Serial Interface (ASC)

2.2 ASC Kernel Registers

Figure 2-16 and Table 2-8 show all registers associated with the ASC Kernel.

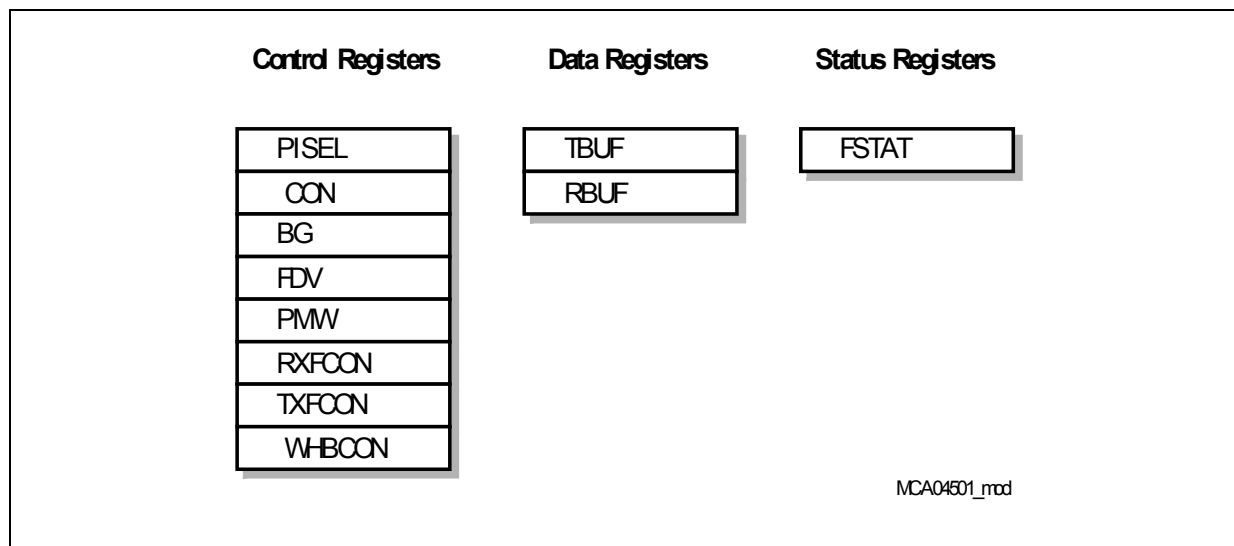


Figure 2-16 ASC Kernel Registers

Table 2-8 ASC Kernel Registers

Register Short Name	Register Long Name	Offset Address	Description see
PISEL	Peripheral Input Select Register	0004 _H	Page 2-30
CON	Control Register	0010 _H	Page 2-30
BG	Baud Rate Timer Reload Register	0014 _H	Page 2-34
FDV	Fractional Divider Register	0018 _H	Page 2-35
PMW	IrDA Pulse Mode and Width Register	001C _H	Page 2-36
TBUF	Transmit Buffer Register	0020 _H	Page 2-36
RBUF	Receive Buffer Register	0024 _H	Page 2-37
RXFCON	Receive FIFO Control Register	0040 _H	Page 2-39
TXFCON	Transmit FIFO Control Register	0044 _H	Page 2-41
FSTAT	FIFO Status Register	0048 _H	Page 2-43
WHBCON	Write Hardware Bits Control Register	0050 _H	Page 2-33

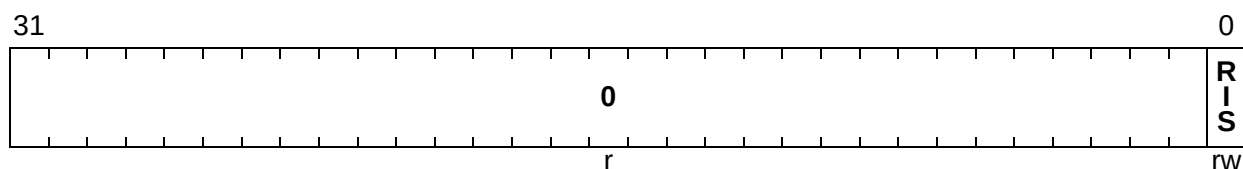
Asynchronous/Synchronous Serial Interface (ASC)

The PISEL register controls the input signal selection of the ASC module. Each input of the module kernel receiver signals has associated two input lines

PISEL

Peripheral Input Select Register

Reset Value: 0000 0000_H



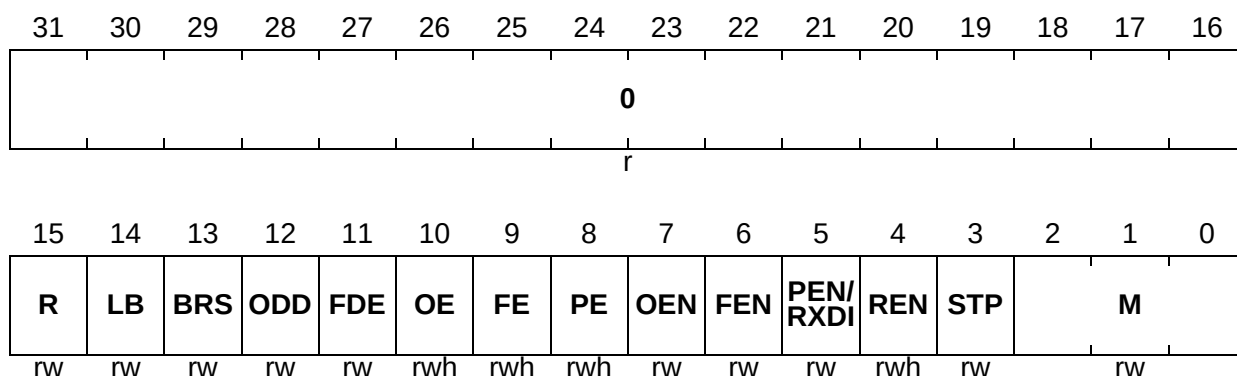
Field	Bits	Type	Description
RIS	0	rw	Receive Input Select 0 Default input port is selected for receiver input 1 Alternate input port is selected for receiver input
0	[31:1]	0	Reserved ; read as 0; should be written with 0.

The serial operating modes of the ASC module are controlled by its control register CON. This register contains control bits for mode and error check selection, and status flags for error identification.

CON

Control Register

Reset Value: 0000 0000_H



Asynchronous/Synchronous Serial Interface (ASC)

Field	Bits	Type	Description
M	[2:0]	rw	Mode Selection 000 8-bit data Synchronous Mode 001 8-bit data Asynchronous Mode 010 IrDA mode, 8-bit data Asynchronous Mode 011 7-bit data + parity Asynchronous Mode 100 9-bit data Asynchronous Mode 101 8-bit data + wake up bit Asynchronous Mode 110 Reserved. Do not use this combination! 111 8-bit data + parity Asynchronous Mode
STP	3	rw	Number of Stop Bit Selection 0 One stop bit 1 Two stop bits
REN	4	rwh	Receiver Enable Control 0 Receiver disabled 1 Receiver enabled <i>Note: Bit is reset by hardware after reception of byte in Synchronous Mode.</i>
PEN/ RXDI	5	rw	Parity Check Enable / RXD Input Inverter Enable in IrDA Mode All Asynchronous modes except IrDA mode 0 Ignore parity 1 Check parity Only IrDA mode (M = 010 _B) 0 RXD input is not inverted 1 RXD input is inverted
FEN	6	rw	Framing Check Enable (asynchronous modes only) 0 Ignore framing errors 1 Check framing errors
OEN	7	rw	Overrun Check Enable 0 Ignore overrun errors 1 Check overrun errors
PE	8	rwh	Parity Error Flag Set by hardware on a parity error (PEN = 1). Must be reset by software.
FE	9	rwh	Framing Error Flag Set by hardware on a framing error (FEN = 1). Must be reset by software.

Asynchronous/Synchronous Serial Interface (ASC)

Field	Bits	Type	Description
OE	10	rwh	Overrun Error Flag Set by hardware on an overrun/underflow error (OEN = 1). Must be reset by software.
FDE	11	rw	Fractional Divider Enable 0 Fractional divider disabled 1 Fractional divider is enabled and used as prescaler for baud rate timer (bit BRS is don't care)
ODD	12	rw	Parity Selection 0 Even parity selected (parity bit set on odd number of 1s in data) 1 Odd parity selected (parity bit set on even number of 1s in data)
BRS	13	rw	Baud Rate Selection 0 Baud rate timer prescaler divide-by-2 selected 1 Baud rate timer prescaler divide-by-3 selected <i>Note: BRS is don't care if FDE = 1 (fractional divider enabled)</i>
LB	14	rw	Loopback Mode Enable 0 Loop-Back mode disabled 1 Loop-Back mode enabled
R	15	rw	Baud Rate Generator Run Control 0 Baud rate generator disabled (ASC inactive) 1 Baud rate generator enabled <i>Note: BR_VALUE should only be written if R = 0.</i>
0	[31:16]	r	Reserved ; read as 0; should be written with 0.

Note: Serial data transmission or reception is possible only when the run bit CON.R is set to 1. Otherwise, the serial interface is idle. Do not program the mode control field CON.M to the reserved combination to avoid unpredictable behavior of the serial interface.

Critical “rwh” Bits

Register CON contains three error flags: PE, FE and OE. If the software wants to modify only one of these error flags, it uses typically a Read-Modify-Write (RMW) instruction. When one of the other error flags, which is not intended to be modified by the RWM instruction, is changed by hardware after the read access but before write back access of the RMW instruction, it is overwritten with the old bit value and the hardware change

Asynchronous/Synchronous Serial Interface (ASC)

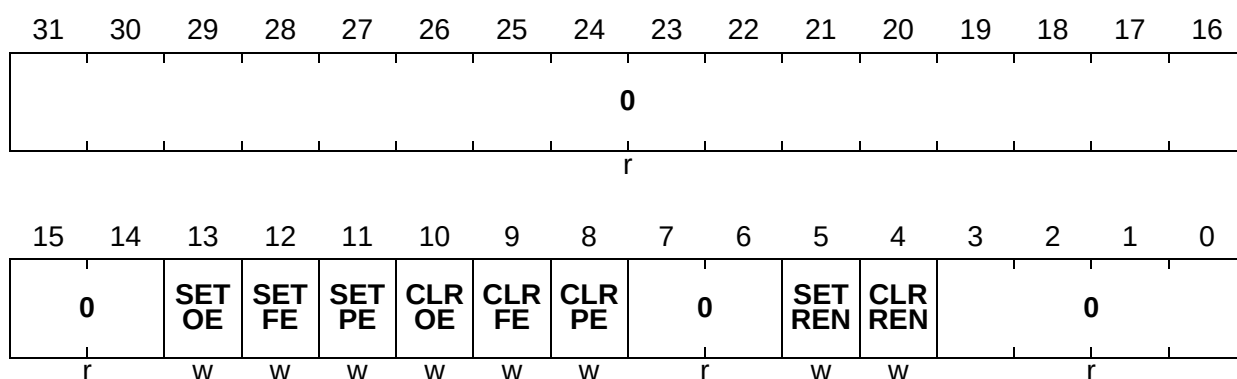
of the bit gets lost. This problem does not affect the bits which are intended to be modified by the RMW instruction. It only affects bits which are not intended to be changed with the RMW instruction.

The three error flags in register CON and the REN bit can be additionally set or reset by software via register WHBCON. This capability avoids the problem with the CON register RMW instruction access to the error flags. WHBCON is a write-only register. Reading WHBCON always returns 0000 0000_H.

WHBCON

Write Hardware Bits Control Register

Reset Value: 0000 0000_H



Field	Bits	Type	Description
CLRREN	4	w	Clear Receiver Enable Bit 0 No effect. 1 Bit CON.REN is cleared. Bit is always read as 0.
SETREN	5	w	Set Receiver Enable Bit 0 No effect. 1 Bit CON.REN is set. Bit is always read as 0.
CLRPE	8	w	Clear Parity Error Flag 0 No effect. 1 Bit CON.PE is cleared. Bit is always read as 0.
CLRFE	9	w	Clear Framing Error Flag 0 No effect. 1 Bit CON.FE is cleared. Bit is always read as 0.

Asynchronous/Synchronous Serial Interface (ASC)

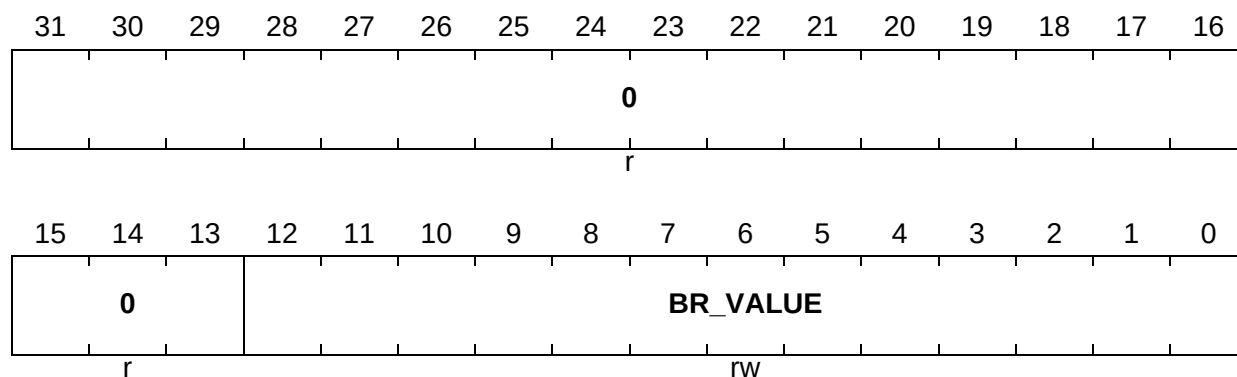
Field	Bits	Type	Description
CLROE	10	w	Clear Overrun Error Flag 0 No effect. 1 Bit CON.OE is cleared. Bit is always read as 0.
SETPE	11	w	Set Parity Error Flag 0 No effect. 1 Bit CON.PE is set. Bit is always read as 0.
SETFE	12	w	Set Framing Error Flag 0 No effect. 1 Bit CON.FE is set. Bit is always read as 0.
SETOE	13	w	Set Overrun Error Flag 0 No effect. 1 Bit CON.OE is set. Bit is always read as 0.
0	[3:0], [7:6], [31:14]	r	Reserved ; read as 0; should be written with 0.

The baud rate timer reload register BG of the ASC module contains the 13-bit reload value for the baud rate timer in Asynchronous and Synchronous Mode.

BG

Baud Rate Timer/Reload Register

Reset Value: 0000 0000_H



Asynchronous/Synchronous Serial Interface (ASC)

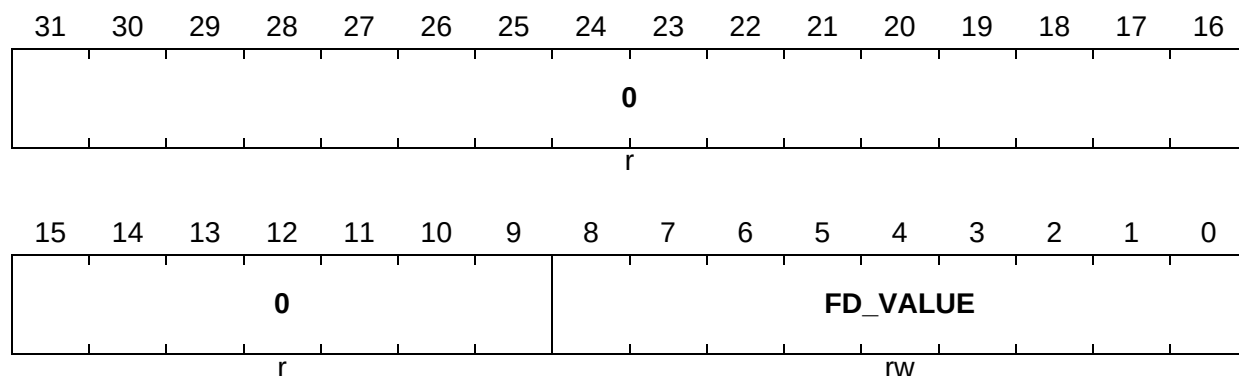
Field	Bits	Type	Description
BR_VALUE	[12:0]	rw	Baud Rate Timer/Reload Value Reading returns the 13-bit content of the baud rate timer; writing loads the baud rate timer/reload value. <i>Note: BG should only be written if R=0.</i>
0	[31:13]	r	Reserved for future use; reading returns 0; writing to these bit positions has no effect.

The fractional divider register FDV of the ASC module contains the 9-bit divider value for the fractional divider (asynchronous mode only).

FDV

Fractional Divider Register

Reset Value: 0000 0000_H



Field	Bits	Type	Description
FD_VALUE	[8:0]	rw	Fractional Divider Register Value FD_VALUE contains the 9-bit value of the fractional divider which defines the fractional divider ratio $n/512$ ($n=0-511$). With $n=0$, the fractional divider is switched off (input = output frequency)
0	[31:9]	r	Reserved for future use; reading returns 0; writing to these bit positions has no effect.

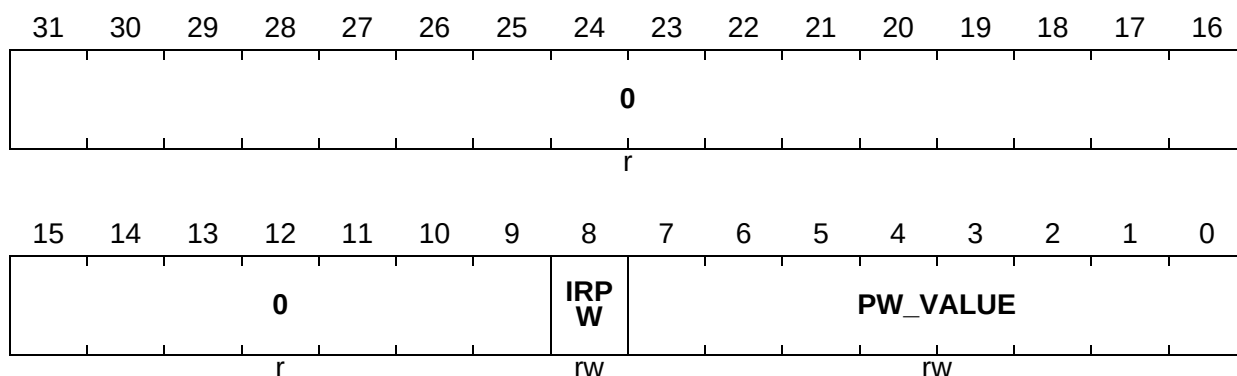
The IrDA pulse mode and width register PMW of the ASC module contains the 8-bit IrDA pulse width value and the IrDA pulse width mode select bit. This register is only required in the IrDA operating mode.

Asynchronous/Synchronous Serial Interface (ASC)

PMW

IrDA Pulse Mode/Width Register

Reset Value: 0000 0000_H



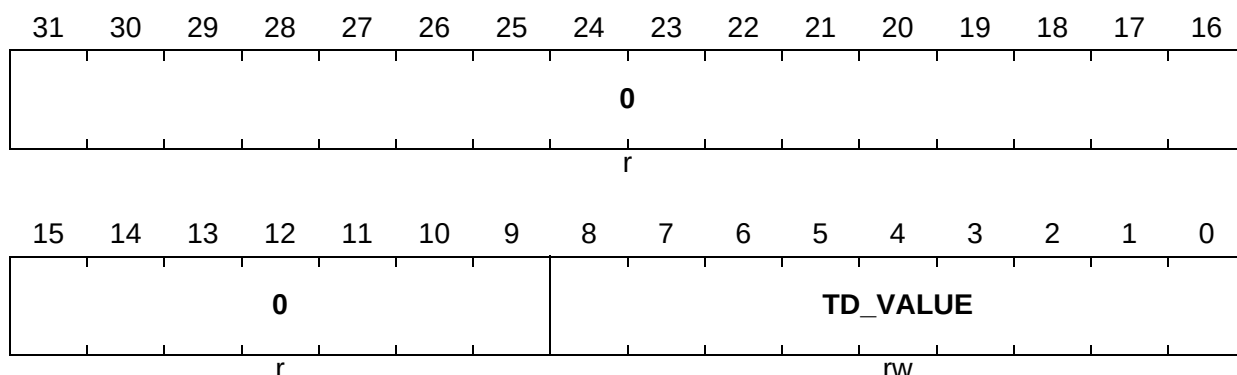
Field	Bits	Type	Description
PW_VALUE	[7:0]	rw	IrDA Pulse Width Value PW_VALUE is the 8-bit value n, which defines the variable pulse width of an IrDA pulse. Depending on the ASC input frequency f_{ASC} , this value can be used to adjust the IrDA pulse width to value which is not equal 3/16 bit time (e.g. 1.6 μ s).
IRPW	8	rw	IrDA Pulse Width Mode Control 0 IrDA pulse width is 3/16 of the bit time 1 IrDA pulse width is defined by PW_VALUE
0	[31:9]	r	Reserved ; read as 0; should be written with 0.

The transmitter buffer register TBUF of the ASC module contains the transmit data value in Asynchronous and Synchronous Modes.

TBUF

Transmit Buffer Register

Reset Value: 0000 0000_H



Asynchronous/Synchronous Serial Interface (ASC)

Field	Bits	Type	Description
TD_VALUE	[8:0]	rw	Transmit Data Register Value TBUF contains the data to be transmitted in asynchronous and synchronous operating mode of the ASC. Data transmission is double buffered; therefore, a new value can be written to TBUF before the transmission of the previous value is complete.
0	[31:9]	r	Reserved for future use; reading returns 0; writing to these bit positions has no effect.

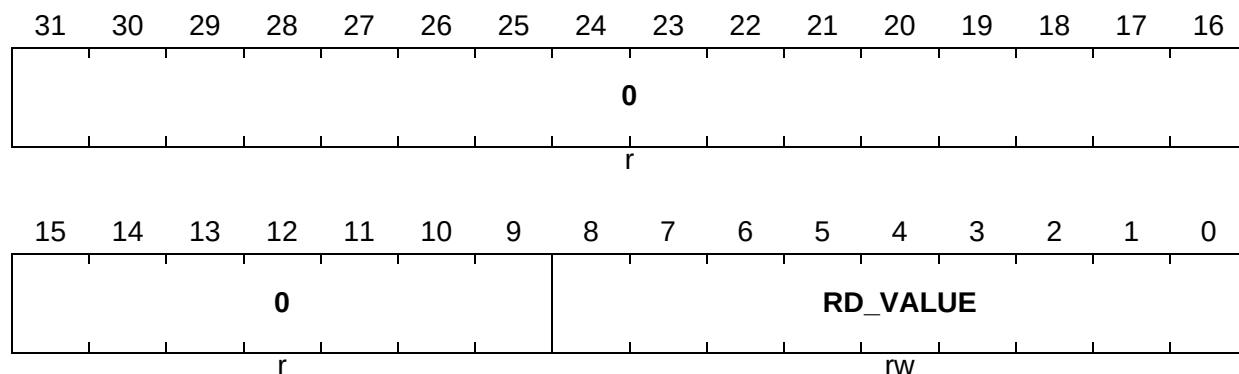
Note: In IrDA Mode, after CON register is set, some delay should be inserted before writing the transmitted data to TBUF register. This is because it takes time to generate the baud rate to transmit the IrDA frame.

The receiver buffer register RBUF of the ASC module contains the receive data value in Asynchronous and Synchronous Modes.

RBUF

Receive Buffer Register

Reset Value: 0000 0000_H



Asynchronous/Synchronous Serial Interface (ASC)

Field	Bits	Type	Description
RD_VALUE	[8:0]	rw	Receive Data Register Value RBUF contains the received data bits and, depending on the selected mode, the parity bit in the asynchronous and synchronous operating modes of the ASC. In Asynchronous Mode, with CON.M = 011 _B (7-bit data + parity), the received parity bit is written into RBUF.7. In Asynchronous Mode with CON.M = 111 _B (8-bit data + parity), the received parity bit is written into RBUF.8.
0	[31:9]	r	Reserved for future use; reading returns 0; writing to these bit positions has no effect.

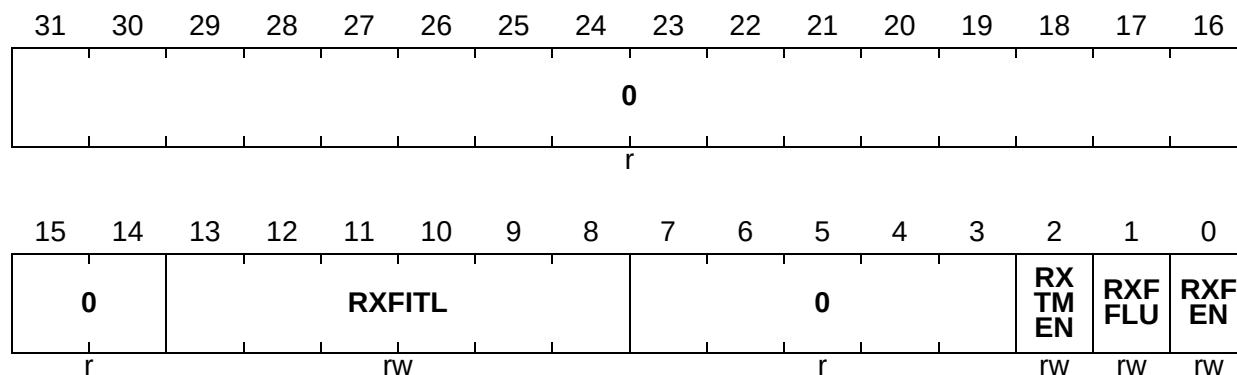
Asynchronous/Synchronous Serial Interface (ASC)

The receive FIFO control register RXFIFO contains control bits and bit fields that define the operating mode of the receive FIFO.

RXFCON

Receive FIFO Control Register

Reset Value: 0000 0100_H



Field	Bits	Type	Description
RXFEN	0	rw	Receive FIFO Enable 0 Receive FIFO is disabled 1 Receive FIFO is enabled <i>Note: Resetting RXFEN automatically flushes the receive FIFO.</i>
RXFFLU	1	rw	Receive FIFO Flush 0 No operation 1 Receive FIFO is flushed <i>Note: Setting RXFFLU clears bit field FSTAT. RXFFL. Bit RXFFLU is always read as 0.</i>
RXTMEN	2	rw	Receive FIFO Transparent Mode Enable 0 Receive FIFO Transparent Mode is disabled 1 Receive FIFO Transparent Mode is enabled <i>Note: This bit is not applicable if the receive FIFO is disabled (RXFEN = 0).</i>

Asynchronous/Synchronous Serial Interface (ASC)

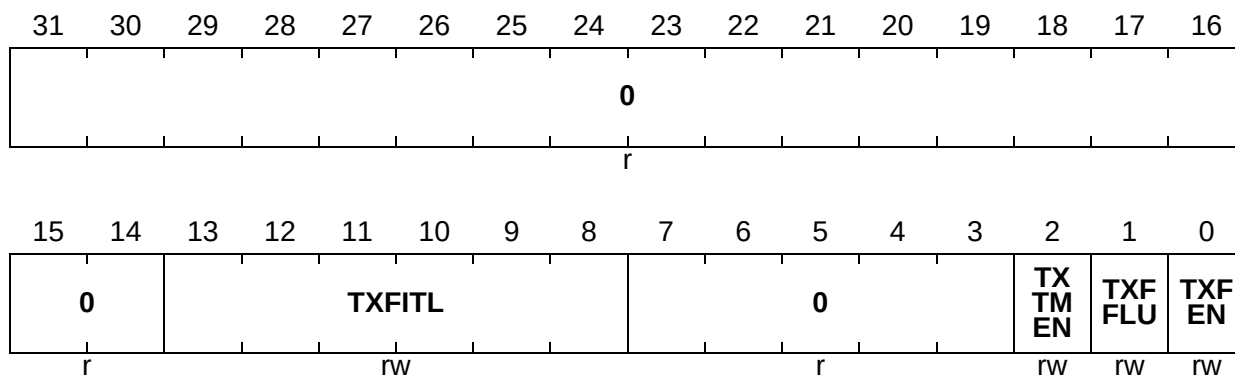
Field	Bits	Type	Description
RXFITL	[13:8]	rw	Receive FIFO Interrupt Trigger Level Defines a receive FIFO interrupt trigger level. A receive interrupt request (RIR) is always generated after the reception of a byte when the filling level of the receive FIFO is equal to or greater than RXFITL 000000 Reserved. 000001 Interrupt trigger level is set to one 000010 Interrupt trigger level is set to two ... 001000 Interrupt trigger level is set to eight Others: reserved <i>Note: In Transparent Mode, this bit field is not applicable.</i> <i>Note: Combinations defining an interrupt trigger level greater than the configured FIFO size should not be used</i>
0	[7:3], [31:14]	r	Reserved for future use; reading returns 0; writing to these bit positions has no effect.

Asynchronous/Synchronous Serial Interface (ASC)

The transmit FIFO control register TXFIFO contains control bits and bit fields that define the operating mode of the transmit FIFO.

TXFCON

Transmit FIFO Control Register Reset Value: 0000 0100_H



Field	Bits	Type	Description
TXFEN	0	rw	Transmit FIFO Enable 0 Transmit FIFO is disabled 1 Transmit FIFO is enabled <i>Note: Resetting TXFEN automatically flushes the transmit FIFO.</i>
TXFFLU	1	rw	Transmit FIFO Flush 0 No operation 1 Transmit FIFO is flushed <i>Note: Setting TXFFLU clears bit field FSTAT.TXFFL. Bit TXFFLU is always read as 0.</i>
TXTMEN	2	rw	Transmit FIFO Transparent Mode Enable 0 Transmit FIFO Transparent Mode is disabled 1 Transmit FIFO Transparent Mode is enabled <i>Note: This bit is not applicable if the transmit FIFO is disabled (TXFEN = 0).</i>

Asynchronous/Synchronous Serial Interface (ASC)

Field	Bits	Type	Description
TXFITL	[13:8]	rw	Transmit FIFO Interrupt Trigger Level Defines a transmit FIFO interrupt trigger level. A transmit interrupt request (TIR) is always generated after the transfer of a byte when the filling level of the transmit FIFO is equal to or lower than TXFITL 000000 Reserved. 000001 Interrupt trigger level is set to one 000010 Interrupt trigger level is set to two ... 001000 Interrupt trigger level is set to eight Others: reserved <i>Note: In Transparent Mode, this bit field is not applicable.</i> <i>Note: Combinations defining an interrupt trigger level greater than the configured FIFO size should not be used.</i>
0	[7:3], [31:14]	r	Reserved for future use; reading returns 0; writing to these bit positions has no effect.

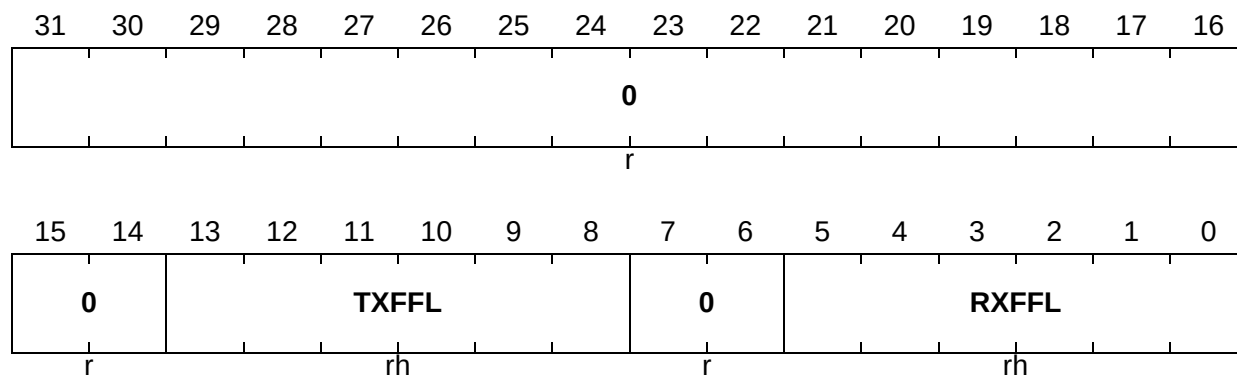
Asynchronous/Synchronous Serial Interface (ASC)

The FIFO status register FSTAT indicates the filling levels of the receive and transmit FIFOs.

FSTAT

FIFO Status Register

Reset Value: 0000 0000_H



Asynchronous/Synchronous Serial Interface (ASC)

Field	Bits	Type	Description
RXFFL	[5:0]	rh	Receive FIFO Filling Level 000000Receive FIFO is filled with zero byte 000001Receive FIFO is filled with one byte 000010Receive FIFO is filled with two bytes ... 001000Receive FIFO is filled with eight bytes Others: reserved <i>Note: RXFFL is cleared after a receive FIFO flush operation.</i> <i>Note: Combinations defining an interrupt trigger level greater than the configured FIFO size should not be used</i>
TXFFL	[13:8]	rh	Transmit FIFO Filling Level 000000Transmit FIFO is filled with zero byte 000001Transmit FIFO is filled with one byte 000010Transmit FIFO is filled with two bytes ... 001000Transmit FIFO is filled with eight bytes Others: reserved <i>Note: TXFFL is cleared after a transmit FIFO flush operation.</i> <i>Note: Combinations defining an interrupt trigger level greater than the configured FIFO size should not be used</i>
0	[7:6], [31:14]	r	Reserved for future use; reading returns 0; writing to these bit positions has no effect.

Asynchronous/Synchronous Serial Interface (ASC)

2.3 ASC0/ASC1 Module Implementation

This section describes ASC0/ASC1 module interfaces with the clock control, port connections, interrupt control, and address decoding.

2.3.1 Interfaces of the ASC Modules

Figure 2-17 shows the TC1100 specific implementation details and interconnections of the ASC0/ASC1 modules. The serial I/O lines of these modules can be connected either to Port 0 or to Port 2. Each of the ASC modules is further supplied by clock control, interrupt control, address decoding, and port control logic. Two DMA requests can be generated by each ASC module.

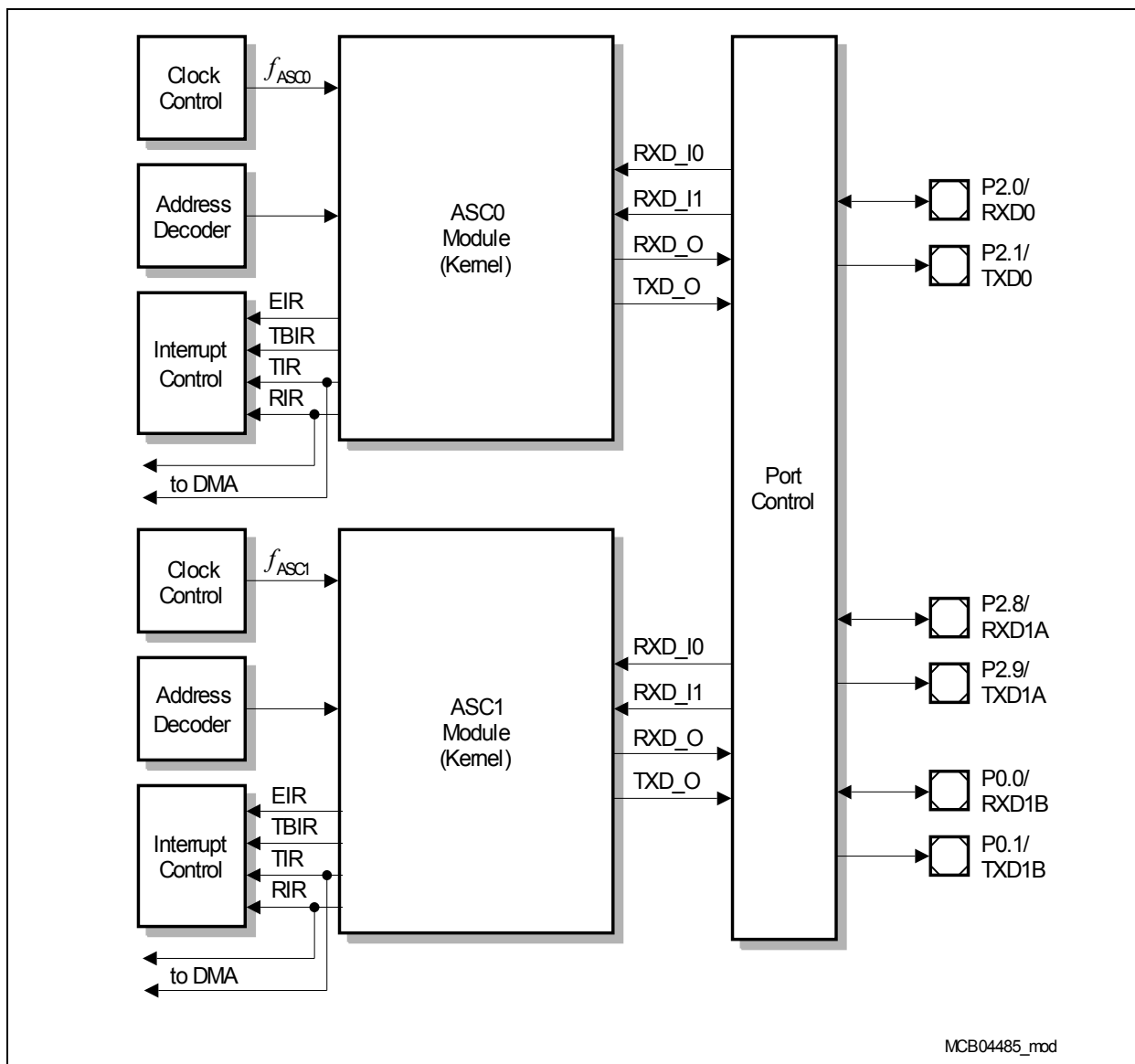


Figure 2-17 ASC0/ASC1 Module Implementation and Interconnections

Asynchronous/Synchronous Serial Interface (ASC)

2.3.2 ASC0/ASC1 Module Related External Registers

Figure 2-18 summarizes the module related external registers which are required for ASC0/ASC1 programming (see also **Figure 2-16** for the module kernel specific registers).

Control Registers	Port Registers	Interrupt Registers
ASC0_CLC	P0_DIR	ASC0_TSRC
ASC1_CLC	P0_ALTSEL0	ASC0_RSRC
ASC1_PSEL	P0_ALTSEL1	ASC0_ESRC
	P0_PUDEL	ASC0_TBSRC
	P0_PUDEN	ASC1_TSRC
	P0_OD	ASC1_RSRC
	P2_DIR	ASC1_ESRC
	P2_ALTSEL0	ASC1_TBSRC
	P2_ALTSEL1	
	P2_PUDEL	
	P2_PUDEN	
	P2_OD	

Figure 2-18 ASC0/ASC1 Implementation Specific Special Function Registers

2.3.2.1 Clock Control Registers

The clock control register allows the programmer to adapt the functionality and power consumption of an ASC module to the requirements of the application. The table below shows the clock control register functionality which is implemented for the ASC modules. ASC0_CLC is controlling the f_{ASC0} clock signal and ASC1_CLC is controlling the f_{ASC1} clock signal.

Asynchronous/Synchronous Serial Interface (ASC)

ASC0_CLC

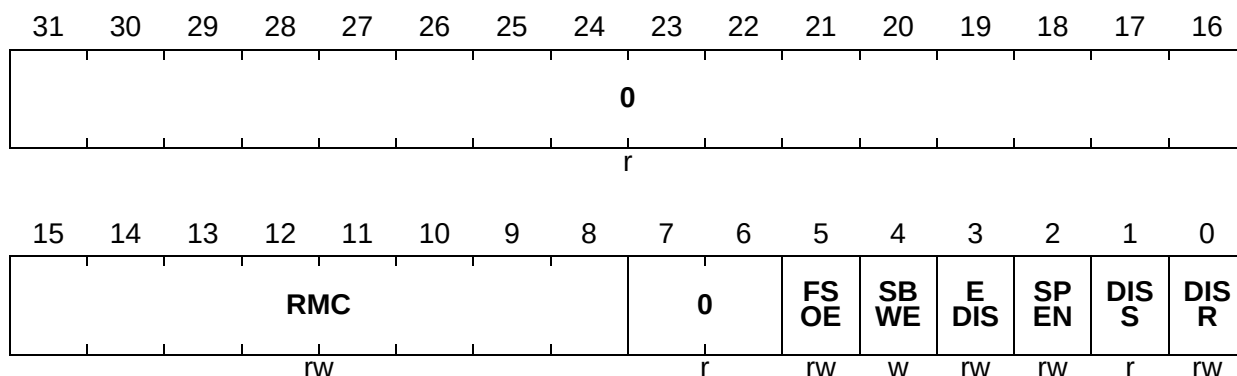
ASC0 Clock Control Register

Reset Value: 0000 0002_H

ASC1_CLC

ASC1 Clock Control Register

Reset Value: 0000 0002_H



Field	Bits	Type	Description
DISR	0	rw	Module Disable Request Bit Used for enable/disable control of the module.
DISS	1	r	Module Disable Status Bit Bit indicates the current status of the module.
SPEN	2	rw	Module Suspend Enable for OCDS Used for enabling the suspend mode.
EDIS	3	rw	External Request Disable Used for controlling the external clock disable request.
SBWE	4	w	Module Suspend Bit Write Enable for OCDS Defines whether SPEN and FSOE are write protected.
FSOE	5	rw	Fast Switch Off Enable Used for fast clock switch off in OCDS suspend mode.
RMC	[15:8]	rw	8-Bit Clock Divider Value in RUN Mode
0	[7:6], [31:16]	r	Reserved ; read as 0; should be written with 0.

Note: After a hardware reset operation, the ASC modules are disabled.

Note: As described in the “System Control Unit” chapter under “Module Clock Divider Control” (“destructive read” access), it must be considered that using the RMC bit field in ASC0_CLC and ASC1_CLC may result in a longer read cycle access time on the FPI Buses for the ASC.

Asynchronous/Synchronous Serial Interface (ASC)

2.3.2.2 Peripheral Input Select Register

The ASC1 module provides a Peripheral Input Select Register that is used to switch the RXD input lines of the ASC1 module kernel to either Port 0 or Port 2 as shown in **Figure 2-19**.

*Note: As shown in **Figure 2-19**, the RXD lines of the ASC module can also be output lines in Synchronous Mode. Port line input/output switching is controlled by the input/output control registers DIR.*

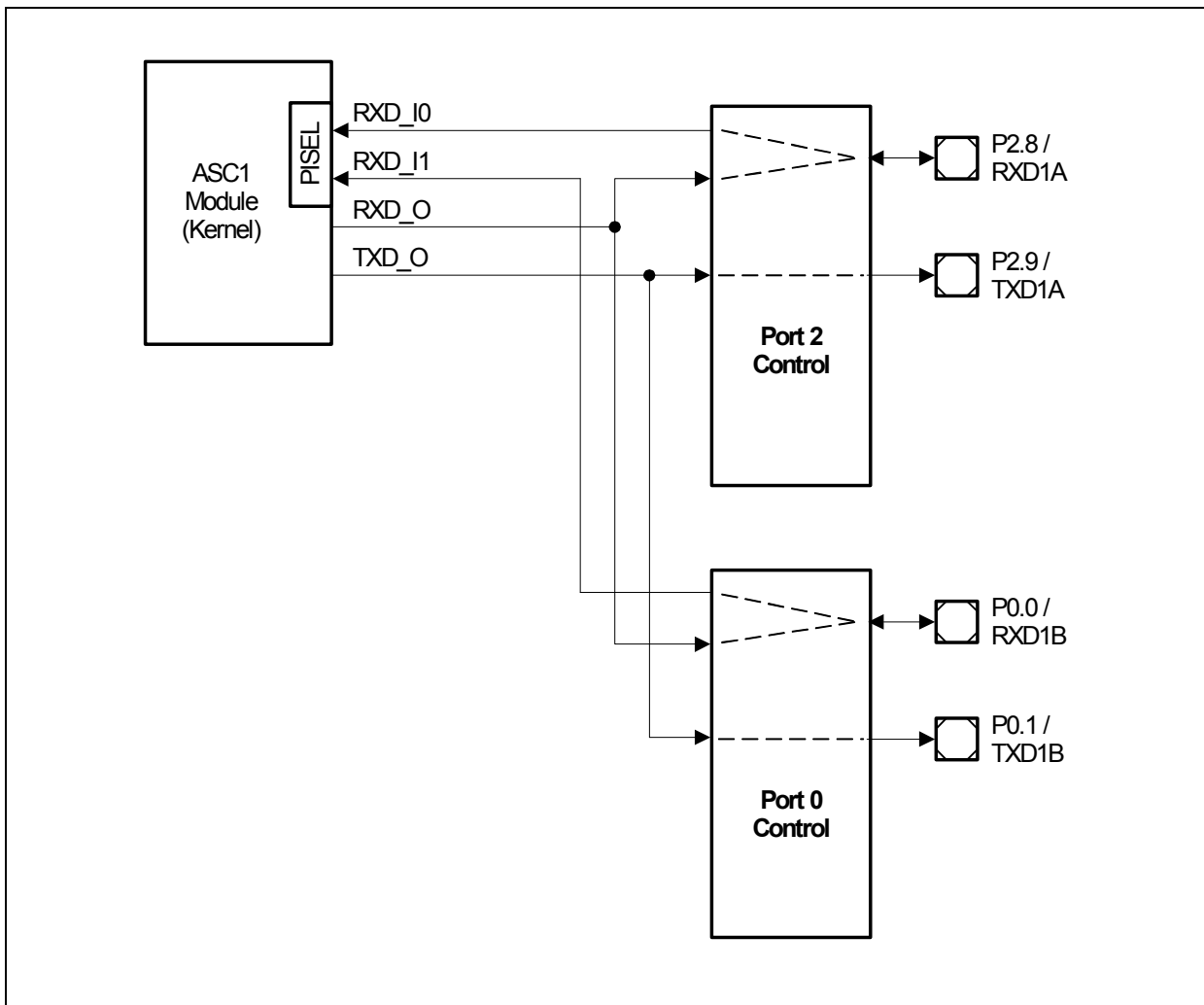


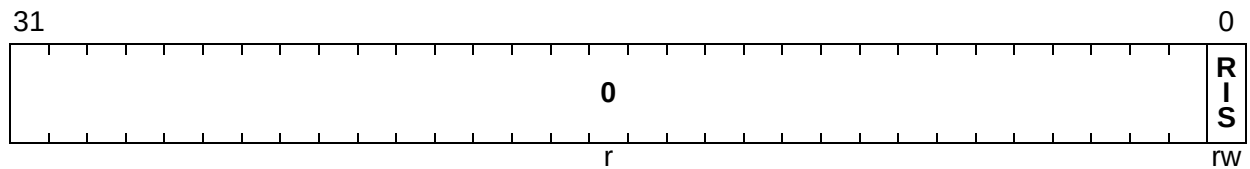
Figure 2-19 RXD Input Line Selection of the ASC Module

Asynchronous/Synchronous Serial Interface (ASC)

ASC1_PISEL

ASC1 Peripheral Input Select Register

Reset Value: 0000 0000_H



Field	Bits	Type	Description
RIS	0	rw	Receive Input Select 0 ASC1 receiver input RXD1A selected 1 ASC1 receiver input RXD1B selected
0	[31:1]	0	Reserved ; read as 0; should be written with 0.

Asynchronous/Synchronous Serial Interface (ASC)

2.3.2.3 Port Control

The interconnections between the ASC modules and the port I/O lines are controlled in the port logics. The following port control operation selections must be executed (additionally to the PISEL programming):

- Input/output direction selection (DIR registers)
- Alternate function selection (ALTSEL0 and ALTSEL1 registers)
- Input/Output driver characteristic control (PUDSEL, PUDEN and OD registers)

Input/Output Function Selection

The port input/output control registers contain the bit fields that select the digital output and input driver characteristics such as pull-up/down devices, port direction (input/output), open-drain, and alternate output selections. The I/O lines for the ASC modules are controlled by the port input/output control registers of Port 0 and Port2.

Table 2-9 shows how bits and bit fields must be programmed for the required I/O functionality of the ASC I/O lines. This table also shows the values of the peripheral input select registers.

Table 2-9 ASC0/ASC1 I/O Control Selection and Setup

Module	Port Lines	PISEL Register	Input/Output Control Register Bits	I/O
ASC0	P2.0/RXD0	–	P2_DIR.P0 = 0 _B	Input
		–	P2_DIR.P0 = 1 _B	Output
			P2_ALTSEL0.P0 = 1 _B	
			P2_ALTSEL1.P0 = 0 _B	
	P2.1/TXD0	–	P2_DIR.P1 = 1 _B	Output
			P2_ALTSEL0.P1 = 1 _B	
			P2_ALTSEL1.P1 = 0 _B	

Asynchronous/Synchronous Serial Interface (ASC)

Table 2-9 ASC0/ASC1 I/O Control Selection and Setup (cont'd)

Module	Port Lines	PISEL Register	Input/Output Control Register Bits	I/O
ASC1	P2.8/RXD1A	ASC1_PISEL.RIS = 0	P2_DIR.P8 = 0 _B	Input
		–	P2_DIR.P8 = 1 _B	Output
			P2_ALTSEL0.P8 = 1 _B	
			P2_ALTSEL1.P8 = 0 _B	
	P0.0/RXD1B	ASC1_PISEL.RIS = 1	P0_DIR.P0 = 0 _B	Input
		–	P0_DIR.P0 = 1 _B	Output
			P0_ALTSEL0.P0 = 0 _B	
			P0_ALTSEL1.P0 = 1 _B	
	P2.9/TXD1A	–	P2_DIR.P9 = 1 _B	Output
			P2_ALTSEL0.P9 = 1 _B	
			P2_ALTSEL1.P9 = 0 _B	
	P0.1/TXD1B	–	P0_DIR.P1 = 1 _B	Output
			P0_ALTSEL0.P1 = 0 _B	
			P0_ALTSEL1.P1 = 1 _B	

Note: In synchronous operating mode of the ASC, the type of the selected RXD port pin (input or output) is not automatically controlled by the ASC but must be defined by a user program by writing the appropriate bit field in the DIR registers.

P0_DIR

Port 0 Direction Register

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
P15	P14	P13	P12	P11	P10	P9	P8	P7	P6	P5	P4	P3	P2	P1	P0
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

Asynchronous/Synchronous Serial Interface (ASC)

Field	Bits	Type	Description
Pn (n = 0-1)	n	rw	Port 0 Pin 0-1 Direction Control ¹⁾ 0 Direction is set to input (default after reset) 1 Direction is set to output
0	[31:16]	r	Reserved ; read as 0; should be written with 0.

¹⁾ Shaded bits and bit field are don't care for ASC I/O port control

P2_DIR

Port 2 Direction Register

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
P15	P14	P13	P12	P11	P10	P9	P8	P7	P6	P5	P4	P3	P2	P1	P0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Field	Bits	Type	Description
Pn (n = 0-1, 8 -9)	n	rw	Port 2 Pin 0, 1, 8-9 Direction Control ¹⁾ 0 Direction is set to input (default after reset) 1 Direction is set to output
0	[31:16]	r	Reserved ; read as 0; should be written with 0.

¹⁾ Shaded bits and bit field are don't care for ASC I/O port control

P0_ALTSELn (n = 1, 0)

Port 0 Alternate Select Register

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
P15	P14	P13	P12	P11	P10	P9	P8	P7	P6	P5	P4	P3	P2	P1	P0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Asynchronous/Synchronous Serial Interface (ASC)

Table 2-10 Function of the Bits P0_ALTSEL0.Pn and P0_ALTSEL1.Pn (n = 0-1)¹⁾

P0_ALTSEL0.Pn	P0_ALTSEL1.Pn	Function
0	1	Alternate Select 2

¹⁾ Shaded bits and bit field are don't care for ASC I/O port control

P2_ALTSELn (n = 1, 0)

Port 2 Alternate Select Register

Reset Value: 0000 0000_H

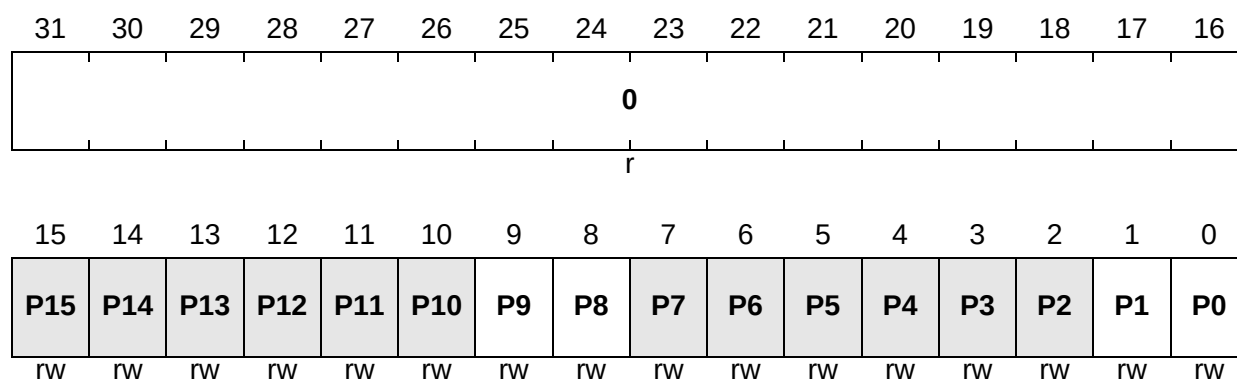


Table 2-11 Function of the Bits P2_ALTSEL0.Pn and P2_ALTSEL1.Pn (n = 0-1, 8-9)¹⁾

P2_ALTSEL0.Pn	P2_ALTSEL1.Pn	Function
1	0	Alternate Select1

¹⁾ Shaded bits and bit field are don't care for ASC I/O port control

The ASC ports also offer the possibility to configure the following output characteristics:

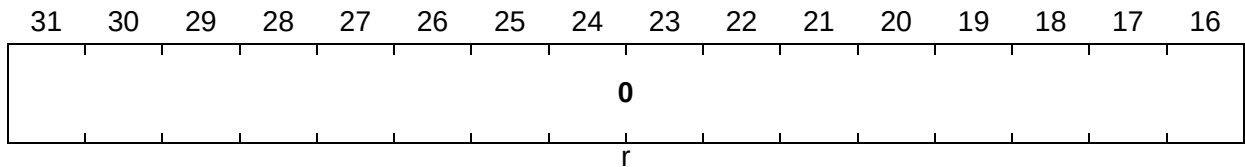
- push/pull (optional pull-up/pull-down)
- open drain with internal pull-up
- open drain with external pull-up

Asynchronous/Synchronous Serial Interface (ASC)

P0_PUDSEL

Port 0 Pull-Up/Pull-Down Select Register

Reset Value: 0000 FFFF_H



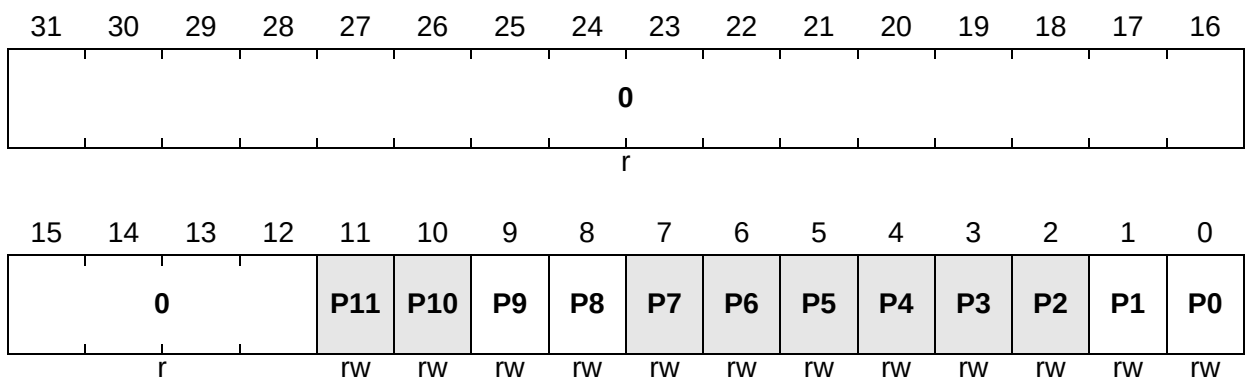
Field	Bits	Type	Description
Pn (n = 0-1)	n	rw	Pull-Up/Pull-Down Select Port 0 Bit n¹⁾ 0 Pull-down device is selected 1 Pull-up device is selected
0	[31:16]	r	Reserved ; read as 0; should be written with 0.

¹⁾ Shaded bits and bit field are don't care for ASC I/O port control

P2_PUDSEL

Port 2 Pull-Up/Pull-Down Select Register

Reset Value: 0000 0FFF_H



Field	Bits	Type	Description
Pn (n = 0-1, 8-9)	n	rw	Pull-Up/Pull-Down Select Port 2 Bit n¹⁾ 0 Pull-down device is selected 1 Pull-up device is selected
0	[31:12]	r	Reserved ; read as 0; should be written with 0.

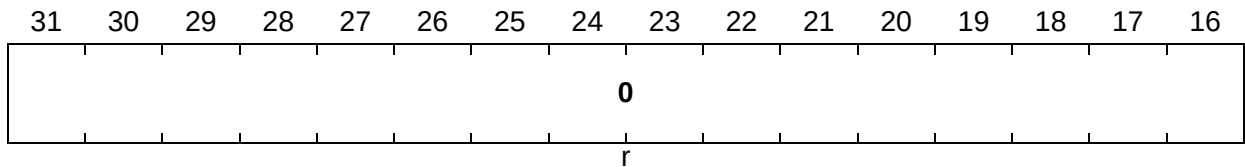
¹⁾ Shaded bits and bit field are don't care for ASC I/O port control

Asynchronous/Synchronous Serial Interface (ASC)

P0_PUDEN

Port 0 Pull-Up/Pull-Down Enable Register

Reset Value: 0000 FFFF_H



15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
P15	P14	P13	P12	P11	P10	P9	P8	P7	P6	P5	P4	P3	P2	P1	P0
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

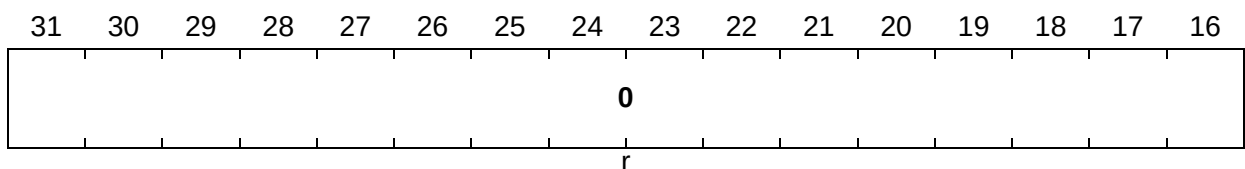
Field	Bits	Type	Description
Pn (n = 0-1)	n	rW	Pull-Up/Pull-Down Enable at Port 0 Bit n¹⁾ 0 Pull-up or Pull-down device is disabled 1 Pull-up or Pull-down device is enabled
0	[31:16]	r	Reserved ; read as 0; should be written with 0.

¹⁾ Shaded bits and bit field are don't care for ASC I/O port control

P2_PUDEN

Port 2 Pull-Up/Pull-Down Enable Register

Reset Value: 0000 0FFF_H



15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0				P11	P10	P9	P8	P7	P6	P5	P4	P3	P2	P1	P0
r				rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

Field	Bits	Type	Description
Pn (n = 0-1, 8-9)	n	rW	Pull-Up/Pull-Down Enable at Port 2 Bit n¹⁾ 0 Pull-up or Pull-down device is disabled 1 Pull-up or Pull-down device is enabled
0	[31:12]	r	Reserved ; read as 0; should be written with 0.

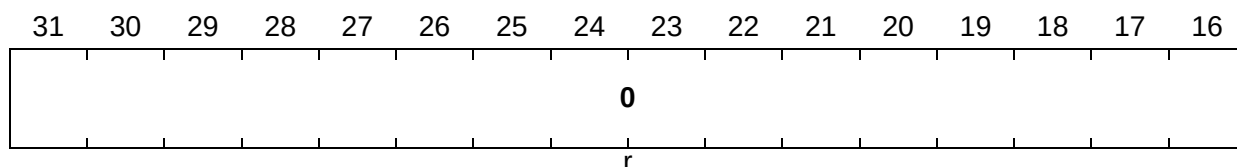
¹⁾ Shaded bits and bit field are don't care for ASC I/O port control

Asynchronous/Synchronous Serial Interface (ASC)

P0_OD

Port 0 Open Drain Control Register

Reset Value: 0000 0000_H



15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
P15	P14	P13	P12	P11	P10	P9	P8	P7	P6	P5	P4	P3	P2	P1	P0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

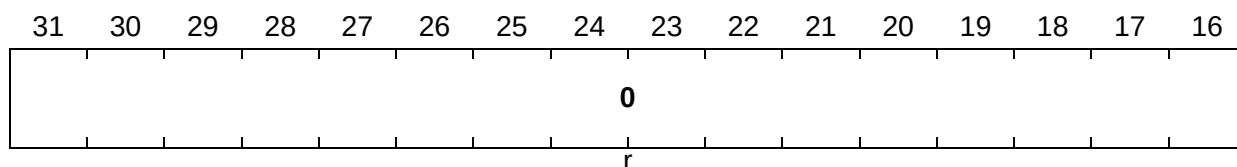
Field	Bits	Type	Description
Pn (n = 0-1)	n	rw	Port 0 Pin n Open Drain Mode¹⁾ 0 Normal Mode, output is actively driven for 0 and 1 state 1 Open Drain Mode, output is actively driven only for 0 state
0	[31:16]	r	Reserved ; read as 0; should be written with 0.

¹⁾ Shaded bits and bit field are don't care for ASC I/O port control

P2_OD

Port 2 Open Drain Control Register

Reset Value: 0000 F000_H



15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
P15	P14	P13	P12	P11	P10	P9	P8	P7	P6	P5	P4	P3	P2	P1	P0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Asynchronous/Synchronous Serial Interface (ASC)

Field	Bits	Type	Description
Pn (n = 0-1, 8-9)	n	rw	Port 2 Pin n Open Drain Mode¹⁾ 0 Normal Mode, output is actively driven for 0 and 1 state 1 Open Drain Mode, output is actively driven only for 0 state
0	[31:16]	r	Reserved ; read as 0; should be written with 0.

¹⁾ Shaded bits and bit field are don't care for ASC I/O port control

Asynchronous/Synchronous Serial Interface (ASC)

2.3.2.4 Interrupt Registers

The eight interrupts of the ASC0 and ASC1 modules are controlled by the following service request control registers:

- ASC0_TSRC, ASC1_TSRC controls the transmit interrupts
- ASC0_RSRC, ASC1_RSRC controls the receive interrupts
- ASC0_ESRC, ASC1_ESRC controls the error interrupts
- ASC0_TBSRC, ASC1_TBSRC controls the transmit buffer empty interrupts

ASC0_TSRC

ASC0 Transmit Interrupt Service Request Control Register

ASC0_RSRC

ASC0 Receive Interrupt Service Request Control Register

ASC0_ESRC

ASC0 Error Interrupt Service Request Control Register

ASC0_TBSRC

ASC0 Transmit Buffer Interrupt Service Request Control Register

ASC1_TSRC

ASC1 Transmit Interrupt Service Request Control Register

ASC1_RSRC

ASC1 Receive Interrupt Service Request Control Register

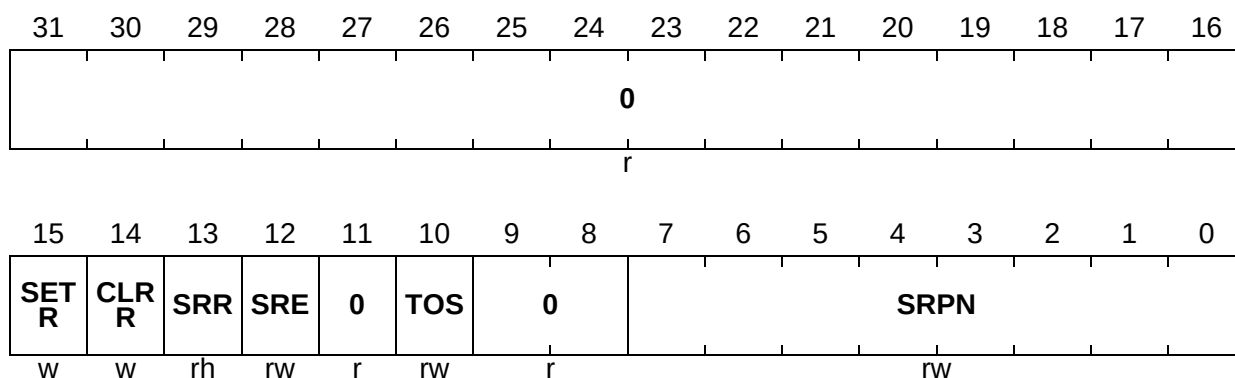
ASC1_ESRC

ASC1 Error Interrupt Service Request Control Register

ASC1_TBSRC

ASC1 Transmit Buffer Interrupt Service Request Control Register

Reset Values: 0000 0000_H



Field	Bits	Type	Description
SRPN	[7:0]	rw	Service Request Priority Number
TOS	10	rw	Type of Service Control
SRE	12	rw	Service Request Enable

Asynchronous/Synchronous Serial Interface (ASC)

Field	Bits	Type	Description
SRR	13	rh	Service Request Flag
CLRR	14	w	Request Clear Bit
SETR	15	w	Request Set Bit
0	[9:8], 11, [31:16]	r	Reserved ; read as 0; should be written with 0.

Note: Further details on interrupt handling and processing are described in the chapter "Interrupt System" of the TC1100 System Units User's Manual.

Asynchronous/Synchronous Serial Interface (ASC)

2.3.3 DMA Requests

The DMA request output lines of the ASC0/ASC1 modules become active whenever its related interrupt line is activated. The DMA request lines are connected to the DMA controller as shown in [Table 2-12](#).

Table 2-12 DMA Request Lines of ASC0/ASC1

Module	Related ASC Interrupt	DMA Request Line	Description
ASC0	RIR	ASC0_RDR	ASC0 Receive DMA Request
	TIR	ASC0_TDR	ASC0 Transmit DMA Request
ASC1	RIR	ASC1_RDR	ASC1 Receive DMA Request
	TIR	ASC1_TDR	ASC1 Transmit DMA Request

2.3.4 ASC0/ASC1 Register Address Ranges

In the TC1100, the registers of the two ASC modules are located in the following address ranges:

- ASC0 module: Module Base Address = F010 0300_H
 Module End Address = F010 03FF_H
- ASC1 module: Module Base Address = F010 0400_H
 Module End Address = F010 04FF_H
- Absolute Register Address = Module Base Address + Offset Address
 (offset addresses see [Table 2-8](#))

Note: The complete and detailed address map of the ASC0/ASC1 modules is described in the chapter “Register Overview” of the TC1100 System Units User’s Manual.

3 Synchronous Serial Interface (SSC)

This chapter describes the two SSC synchronous serial interfaces SSC0 and SSC1 of the TC1100. It contains the following sections:

- Functional description of the SSC Kernel, valid for SSC0 and SSC1 (see [Section 3.1](#))
- Register descriptions of all SSC Kernel specific registers (see [Section 3.2](#))
- TC1100 implementation specific details and registers of the SSC0/SSC1 modules (port connections and control, interrupt control, address decoding, clock control, see [Section 3.3](#)).

Note: The SSC kernel register names described in [Section 3.2](#) will be referenced in the TC1100 User's Manual by the module name prefix "SSC0_" for the SSC0 interface and by "SSC1_" for the SSC1 interface.

3.1 SSC Kernel Description

Figure 3-1 shows a global view of all functional blocks of the SSC interface.

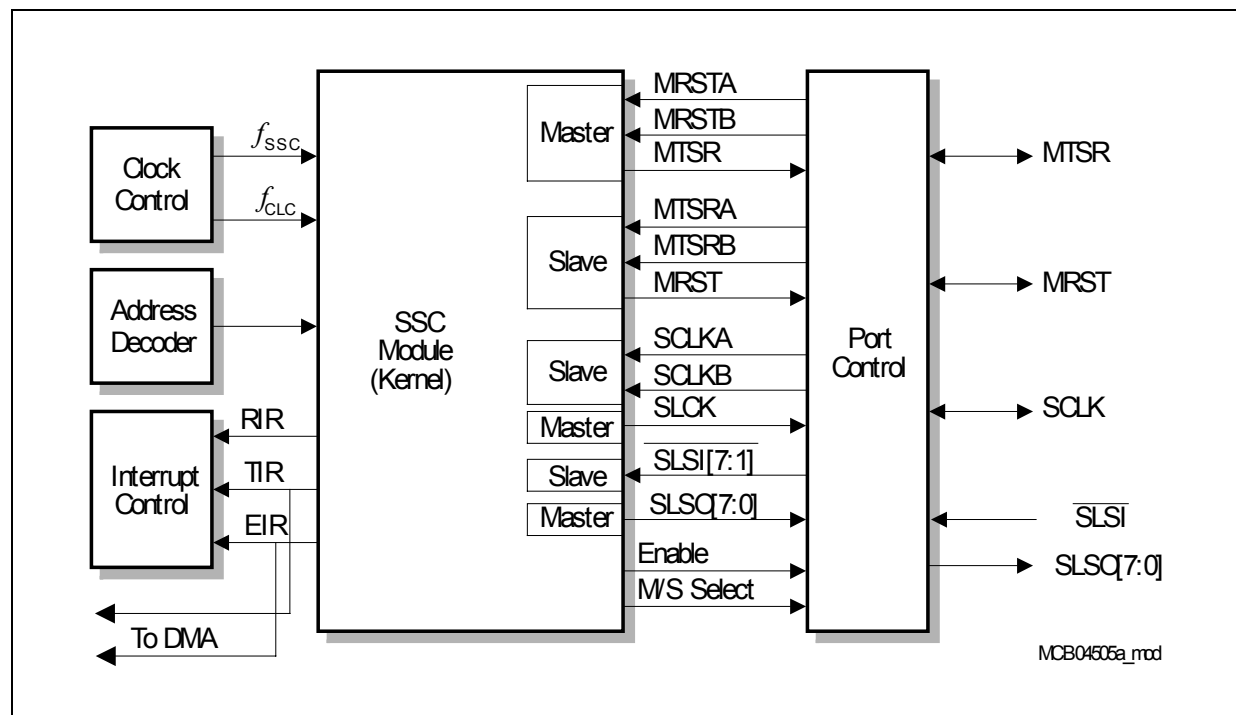


Figure 3-1 General Block Diagram of the SSC Interface

3.1.1 Overview

The SSC supports full-duplex and half-duplex serial synchronous communication up to 37.5 MBaud (@ 75 MHz module clock) with receive and transmit FIFO support. The serial clock signal can be generated by the SSC itself (master mode) or can be received from an external master (slave mode). Data width, shift direction, clock polarity and phase are programmable. This allows communication with SPI-compatible devices. Transmission and reception of data is double-buffered. A shift clock generator provides the SSC with a separate serial clock signal. Eight slave select inputs are available for slave mode operation. Eight programmable slave select outputs (chip selects) are supported in master mode.

Features:

- Master and slave mode operation
 - Full-duplex or half-duplex operation
 - Automatic pad control possible
- Flexible data format
 - Programmable number of data bits: 2 to 16 bit
 - Programmable shift direction: LSB or MSB shift first
 - Programmable clock polarity: idle low or high state for the shift clock
 - Programmable clock/data phase: data shift with leading or trailing edge of the shift clock
- Baud rate generation from 37.5MBaud to 572.2 Baud (@ 75 MHz module clock)
- Interrupt generation
 - On a transmitter empty condition
 - On a receiver full condition
 - On an error condition (receive, phase, baud rate, transmit error)
- Flexible SSC pin configuration
- One slave select inputs $\overline{\text{SLSI}}$ in slave mode
- Eight programmable slave select outputs SLSO in master mode
 - Automatic SLSO generation with programmable timing
 - Programmable active level and enable control
- 4-stage receive FIFO (RXFIFO) and 4-stage transmit FIFO (TXFIFO)
 - Independent control of RXFIFO and TXFIFO
 - 2 to 16 bit FIFO data width
 - Programmable receive/transmit interrupt trigger level
 - Receive and transmit FIFO filling level indication
 - Overrun error generation
 - Underflow error generation

3.1.2 General Operation

The SSC supports full-duplex and half-duplex synchronous communication up to 37.5 MBaud (@ 75 MHz module clock). The serial clock signal can be generated by the SSC itself (master mode) or be received from an external master (slave mode). Data width, shift direction, clock polarity and phase are programmable. This allows communication with SPI-compatible devices. Transmission and reception of data are double-buffered. A shift clock generator provides the SSC with a separate serial clock signal.

Configuration of the high-speed synchronous serial interface is very flexible, so it can work with other synchronous serial interfaces, can serve for master/slave or multimaster interconnections, or can operate compatibly with the popular SPI interface. It can be used to communicate with shift registers (I/O expansion), peripherals (e.g. EEPROMs etc.), or other controllers (networking). The SSC supports half-duplex and full-duplex communication. Data is transmitted or received on pins MTSR (Master Transmit/Slave Receive) and MRST (Master Receive/Slave Transmit). The clock signal is output or input via pin SCLK (Serial Clock). These three pins are typically alternate output functions of port pins. If they are implemented as dedicated bi-directional pins they can be directly controlled by the SSC. In slave mode, the SSC can be selected from a master via dedicated slave select input lines (SLSI). In master mode, automatic generation of slave select output lines (SLSO) is supported.

Synchronous Serial Interface (SSC)

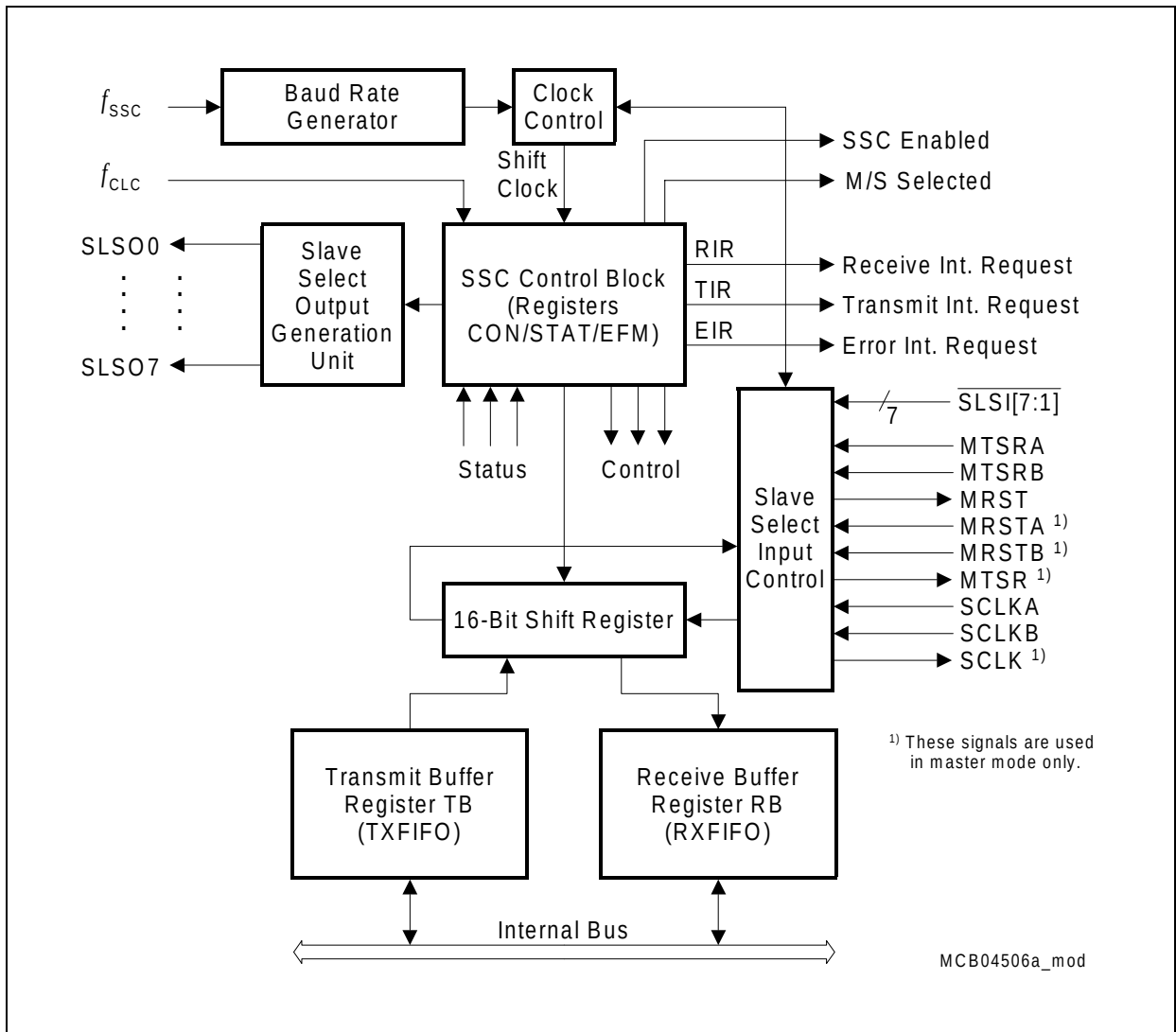


Figure 3-2 Synchronous Serial Channel SSC Block Diagram

3.1.2.1 Operating Mode Selection

The operating mode of the serial channel SSC is controlled by its control register, CON. Status information is contained in its status register, STAT.

The shift register of the SSC is connected to both the transmit pin and the receive pin via the pin control logic (see block diagram in [Figure 3-2](#)). Transmission and reception of serial data are synchronized and take place at the same time, that is, the same number of transmitted bits is also received. Transmit data is written into the Transmit Buffer TB. It is moved to the shift register as soon as this is empty. An SSC master (CON.MS = 1) immediately begins transmitting, while an SSC slave (CON.MS = 0) will wait for an active shift clock. When the transfer starts, the busy flag STAT.BSY is set and the Transmit Interrupt Request line (TIR) will be activated to indicate that register Transmit Buffer (TB) may be reloaded. When the number of bits (2 to 16, as programmed) have been transferred, the contents of the shift register are moved to the Receive Buffer (RB) and the Receive Interrupt Request line (RIR) will be activated. If no further transfer is to take place (TB is empty), STAT.BSY will be cleared at the same time. Software should not modify STAT.BSY, as this flag is hardware controlled.

Note: Only one SSC (etc.) can be master at a given time.

The transfer of serial data bits can be programmed in many respects:

- The data width can be selected from 2 bits to 16 bits
- A transfer may start with the LSB or the MSB
- The shift clock may be idle low or idle high
- The data bits may be shifted with the leading or trailing edge of the clock signal
- The baud rate (shift clock) can be set from 572.2 Baud up to 37.5 MBaud (@ 75 MHz module clock)
- The shift clock can be generated (master) or received (slave)

These features allow the SSC to be adapted to a wide range of applications that require serial data transfer.

The Data Width Selection supports the transfer of frames of any data length from 2-bit “characters” up to 16-bit “characters”. Starting with the LSB (CON.HB = 0) allows communication with such devices as an SSC device in synchronous mode or 8051-like serial interfaces. Starting with the MSB (CON.HB = 1) allows operation compatible with the SPI interface.

Regardless of the data width selected and whether the MSB or the LSB is transmitted first, the transfer data is always right-aligned in registers TB and RB, with the LSB of the transfer data in bit 0 of these registers. The data bits are rearranged for transfer by the internal shift register logic. The unselected bits of TB are ignored; the unselected bits of RB will not be valid and should be ignored by the receiver service routine.

The Clock Control allows the adaptation of transmit and receive behavior of the SSC to a variety of serial interfaces. A specific clock edge (rising or falling) is used to shift out transmit data, while the other clock edge is used to latch in receive data. Bit CON.PH

Synchronous Serial Interface (SSC)

selects the leading edge or the trailing edge for each function. Bit CON.PO selects the level of the clock line in the idle state. So for an idle-high clock, the leading edge is a falling one, a 1-to-0 transition (see [Figure 3-3](#)).

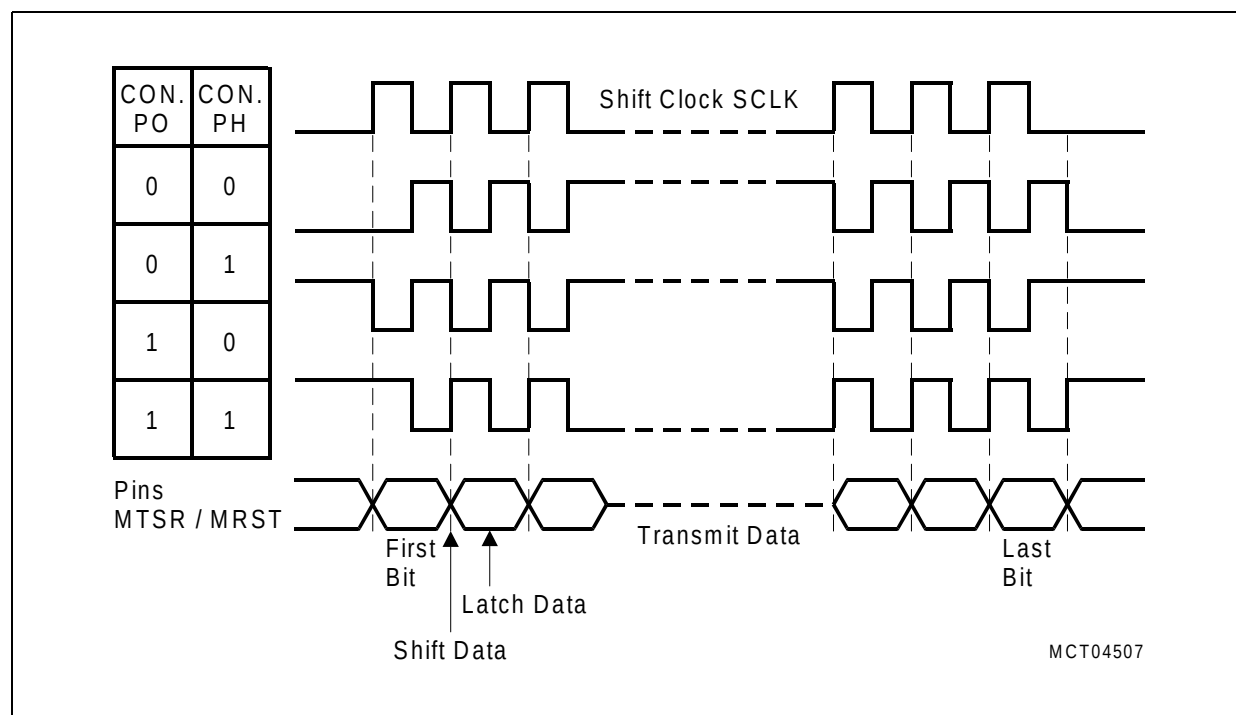


Figure 3-3 Serial Clock Phase and Polarity Options

3.1.2.2 Full-Duplex Operation

Note: The description in this section assumes that the SSC is used with software controlled bi-directional GPIO port lines that provide open-drain capability (see also [Section 3.1.2.5](#)).

The various devices are connected through three lines. The definition of these lines is always determined by the master. The line connected to the master's data output pin MTSR is the transmit line, the receive line is connected to its data input line MRST, and the clock line is connected to pin SCLK. Only the device selected for master operation generates and outputs the serial clock on pin SCLK. All slaves receive this clock, so their pin SCLK must be switched to input mode. The output of the master's shift register is connected to the external transmit line, which in turn is connected to the slaves' shift register input. The output of the slaves' shift register is connected to the external receive line in order to enable the master to receive the data shifted out of the slave. The external connections are hard-wired, with the function and direction of these pins determined by the master or slave operation of the individual device.

Note: The shift direction shown in [Figure 3-4](#) applies to both MSB-first and LSB-first operation.

Synchronous Serial Interface (SSC)

for transmission to the master send only 1s. Since this high level is not actively driven onto the line, but is only held through the pull-up device, the selected slave can pull this line actively to a low level when transmitting a zero bit. The master selects the slave device from which it expects data either by separate select lines, or by sending a special command to this slave.

After performing all necessary initializations of the SSC, the serial interfaces can be enabled. For a master device, the alternate clock line will now go to its programmed polarity. The alternate data line will go to either 0 or 1, until the first transfer will start. After a transfer, the alternate data line will always remain at the logic level of the last transmitted data bit.

When the serial interfaces are enabled, the master device can initiate the first data transfer by writing the transmit data into register TB. This value is copied into the shift register (assumed to be empty at this time), and the selected first bit of the transmit data will be placed onto the MTSR line on the next clock from the shift clock generator (transmission only starts, if CON.EN = 1). Depending on the selected clock phase, also a clock pulse will be generated on the SCLK line. With the opposite clock edge, the master simultaneously latches and shifts in the data detected at its input line MRST. This “exchanges” the transmit data with the receive data. Because the clock line is connected to all slaves, their shift registers will be shifted synchronously with the master’s shift register, shifting out the data contained in the registers, and shifting in the data detected at the input line. After the pre-programmed number of clock pulses (via the data width selection), the data transmitted by the master is contained in all slaves’ shift registers, while the master’s shift register holds the data of the selected slave. In the master and all slaves, the contents of the shift register are copied into the Receive Buffer (RB) and the Receive Interrupt Line (RIR) is activated.

A slave device will immediately output the selected first bit (MSB or LSB of the transfer data) at pin MRST when the contents of the transmit buffer are copied into the slave’s shift register. Bit STAT.BSY is not set until the first clock edge at SCLK appears. The slave device will not wait for the next clock from the shift clock generator — as the master does — because the first clock edge generated by the master may be already used to clock in the first data bit, depending on the selected clock phase. So the slave’s first data bit must already be valid at this time.

Note: On the SSC a transmission and a reception always takes place at the same time, regardless whether valid data has been transmitted or received.

3.1.2.3 Half-Duplex Operation

Note: The description in this section assumes that the SSC is used with software controlled bi-directional GPIO port lines that provide open-drain capability (see also [Section 3.1.2.5](#)).

In a half-duplex configuration, only one data line is necessary for both receiving and transmitting data. The data exchange line is connected to both pins MTSR and MRST of each device, the clock line is connected to the SCLK pin.

The master device controls the data transfer by generating the shift clock, while the slave devices receive it. Due to the fact that all transmit and receive pins are connected to the one data exchange line, serial data may be moved between arbitrary stations.

Similar to full-duplex mode there are two ways to avoid collisions on the data exchange line:

- Only the transmitting device may enable its transmit pin driver
- The non-transmitting devices use open drain output and only send 1s

Because the data inputs and outputs are connected together, a transmitting device will clock in its own data at the input pin (MRST for a master device, MTSR for a slave). In this way, any corruption is detected on the common data exchange line where the received data is not equal to the transmitted data.

Synchronous Serial Interface (SSC)

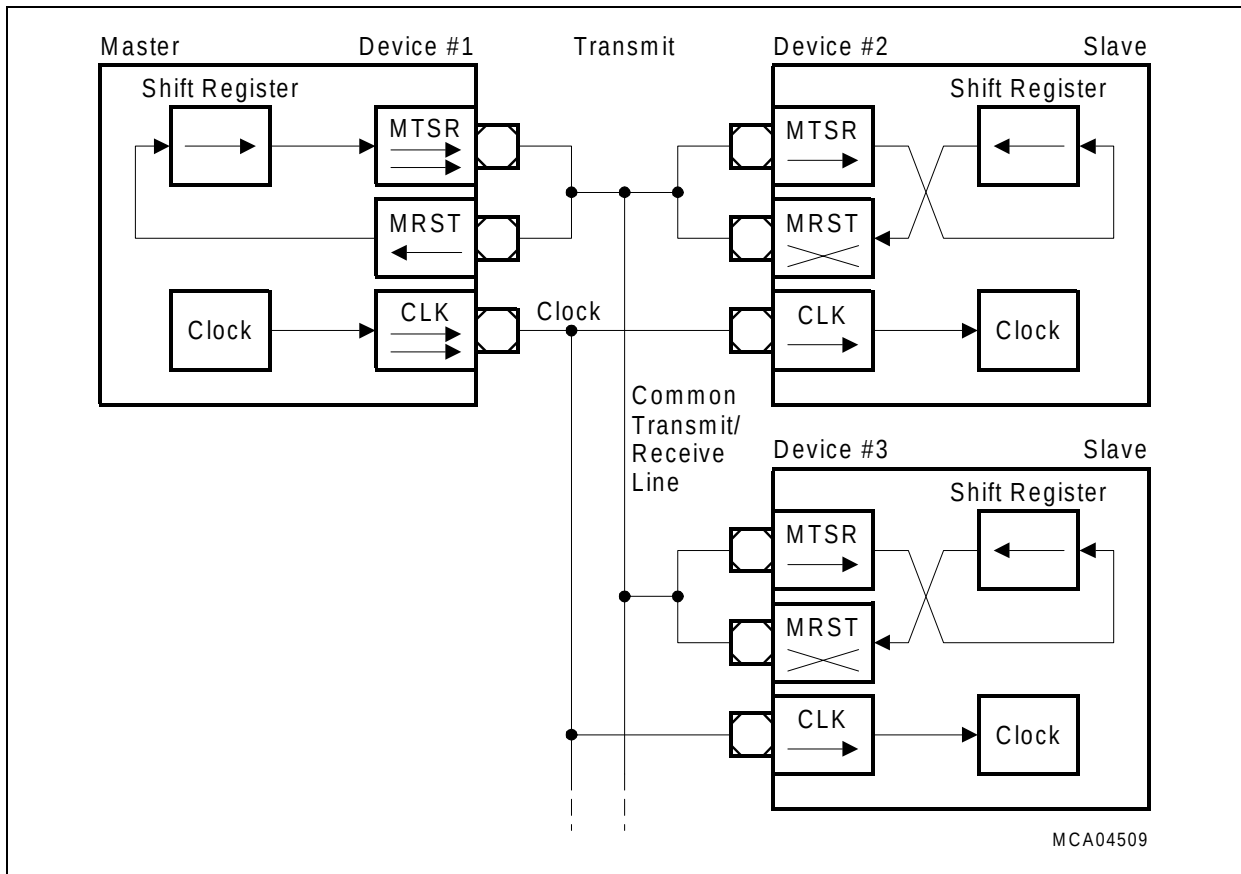


Figure 3-5 SSC Half-Duplex Configuration

3.1.2.4 Continuous Transfers

When the transmit interrupt request flag is set, it indicates that the Transmit Buffer (TB) is empty and is ready to be loaded with the next transmit data. If TB has been reloaded by the time the current transmission is finished, the data is immediately transferred to the shift register and the next transmission can start without any additional delay (according to the selected SLSO timings). On the data line there is no gap between the two successive frames if no delays are selected. For example, two byte transfers would look the same as one word transfer. This feature can be used to interface with devices that can operate with or require more than 16 data bits per transfer. It is just a matter for software how long a total data frame length can be. This option can also be used e.g. to interface to byte-wide and word-wide devices on the same serial bus.

Note: This can only happen in multiples of the selected basic data width, because it would require disabling/enabling of the SSC to reprogram the basic data width on-the-fly.

3.1.2.5 Port Control

The SSC uses three lines to communicate with the external world. Pin SCLK serves as the clock line, while pins MRST (Master Receive/Slave Transmit) and MTSR (Master Transmit/Slave Receive) serve as the serial data input/output lines. As shown in **Figure 3-1** these three lines (SCLK as input, Master Receive, Slave Receive) have all two inputs at the SSC Module kernel. Three bits in register PISEL define which of the two kernel inputs (A or B) are connected. This feature allows for each of the three SSC communication lines to be connected to two inputs coming from different port pins.

Operation of the SSC I/O lines depends on the selected operating mode (master or slave). The direction of the port lines depends on the operating mode. The SSC will automatically use the correct kernel output or kernel input line of the ports when switching modes. Port pins assigned as SSC I/O lines can be controlled in two ways:

- by hardware
- by software

When the SSC I/O lines are connected with dedicated pins typically hardware I/O control should be used. In this case, two output signals reflect directly the state of the CON.EN and CON.MS bits (the M/S select line is inverted to the CON.MS bit definition).

When the SSC I/O lines are connected with bi-directional lines of general purpose I/O ports typically software I/O control should be used. In this case port registers must be programmed for alternate output and input selection. When switching between master and slave mode port registers must be reprogrammed.

Using the open-drain output feature of port lines helps avoid bus contention problems and reduces the need for hard-wired hand-shaking or slave select lines. In this case, it is not always necessary to switch the direction of a port pin. Note that in hardware controlled I/O mode the availability of open-drain outputs depends on the type of the used dedicated output pins. The SSC module itself does not provide any control capability for open drain control.

Note: Details on SSC port connections and configuration see [Section 3.3.1](#).

3.1.2.6 Transmit FIFO Operation

The transmit FIFO (TXFIFO) provides the following functionality:

- Enable/disable control
- Programmable filling level for transmit interrupt generation
- Filling level indication
- FIFO clear (flush) operation
- FIFO overflow error generation
- 2 to 16 bit TXFIFO data width

The transmit FIFO is controlled by the TXFCON control register. When bit TXFCON.TXFEN is set, the transmit FIFO is enabled. The interrupt trigger level defined by TXFCON.TXFITL defines the filling level of the TXFIFO at which a transmit interrupt TIR is generated. This interrupt is always generated when the filling level of the transmit FIFO is equal to or less than the value stored in TXFCON.TXFITL.

Bit field TXFFL in the FIFO status register FSTAT indicates the number of entries that are actually written (valid) in the TXFIFO. Therefore, the software can verify, in the interrupt service routine, for instance, how many bytes can be still written into the transmit FIFO via register TB without getting an overrun error.

The transmit FIFO cannot be accessed directly. All data write operations into the TXFIFO are executed by writing into the TB register. The data width of one TXFIFO stage can be from 2 to 16 bits (as programmed in CON.BM).

The example in [Figure 3-6](#) shows an example of a transmit FIFO operation with a typical data width of 8 bits, representing a byte. In this example, four bytes are transmitted via the transmit output line. The transmit FIFO interrupt trigger level TXFCON.TXFITL is set to 0010_B. The first byte written into the empty TXFIFO via TB is directly transferred into the transmit shift register and is not written into the FIFO. After Byte 1, Bytes 2 to 4 are written into the transmit FIFO.

After the transfer of Byte 2 from the TXFIFO into the transmit shift register of the SSC, 2 bytes remain in the TXFIFO. Therefore, the value of TXFCON.TXFITL is reached and a transmit buffer interrupt will be generated at the beginning and a transmit interrupt at the end of the Byte 2 serial transmission. Finally, after the start of the serial transmission of Byte 4, the TXFIFO is again empty.

If the TXFIFO is full and additional bytes are written into TB, the transmit interrupt will be generated with bit CON.TE set if bit CON.TEN was set. In this case, the data that was last written into the transmit FIFO is overwritten and the transmit FIFO filling level FSTAT.TXFFL is set to maximum.

The TXFIFO can be flushed or cleared by setting bit TXFCON.TXFFLU. After this TXFIFO flush operation, the TXFIFO is empty and the transmit FIFO filling level FSTAT.TXFFL is set to 0000_B. A running serial transmission is not aborted by a receive FIFO flush operation.

Synchronous Serial Interface (SSC)

Note: The TXFIFO is flushed automatically with a reset operation of the SSC module and if the TXFIFO becomes disabled (resetting bit TXFCON.TXFEN) after it was previously enabled.

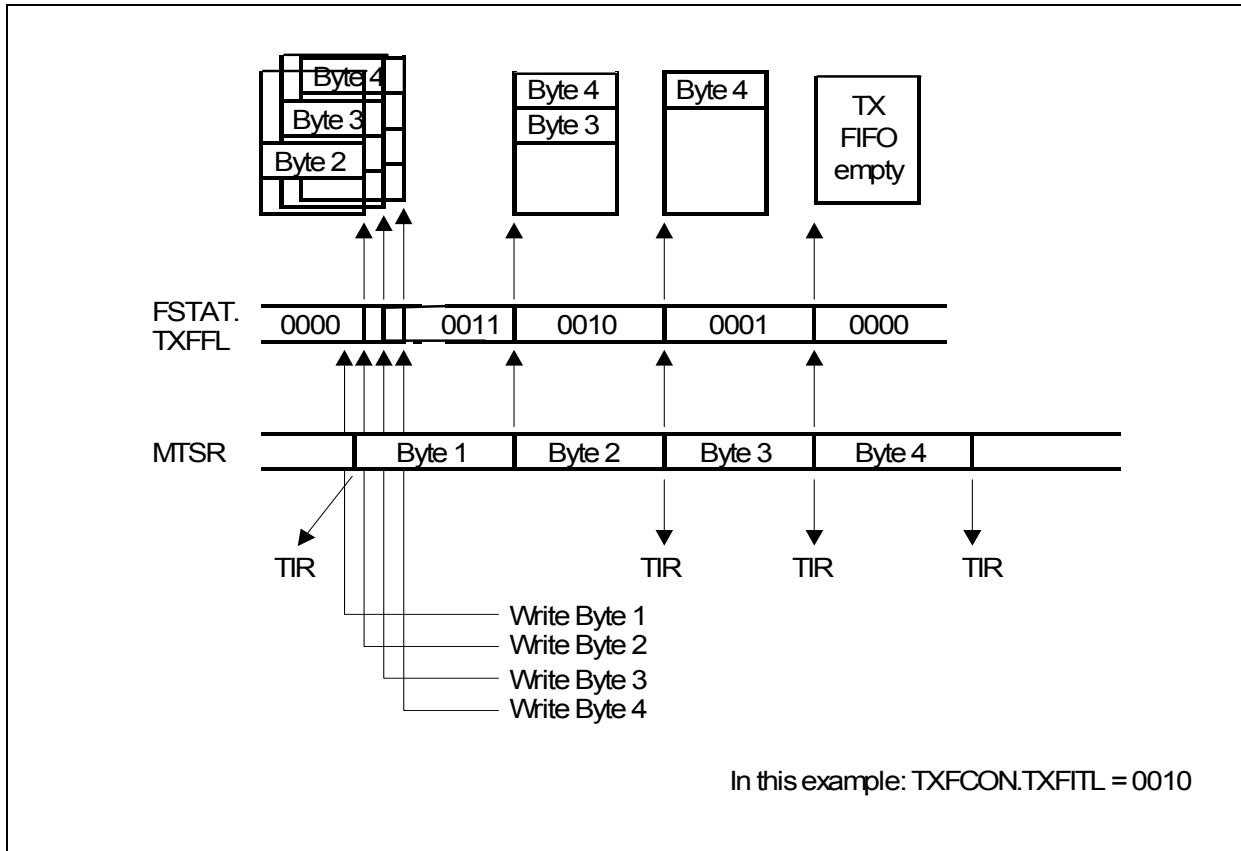


Figure 3-6 Transmit FIFO Operation Example

3.1.2.7 Receive FIFO Operation

The receive FIFO (RXFIFO) provides the following functionality:

- Enable/disable control
- Programmable filling level for receive interrupt generation
- Filling level indication
- FIFO clear (flush) operation
- FIFO overflow error generation
- 2 to 16 bit RXFIFO data width

The receive FIFO is controlled by the RXFCON control register. When bit RXFCON.RXFEN is set, the receive FIFO is enabled. The interrupt trigger level defined by RXFCON.RXFITL defines the filling level of RXFIFO at which a receive interrupt RIR is generated. RIR is always generated when the filling level of the receive FIFO is equal to or greater than the value stored in RXFCON.RXFITL.

Bit field RXFFL in the FIFO status register FSTAT indicates the number of bytes that have been actually written into the FIFO and can be read out of the FIFO by a user program.

The receive FIFO cannot be accessed directly. All data read operations from the RXFIFO are executed by reading the RB register. The data width of one RXFIFO stage can be from 2 to 16 bits (as programmed in CON.BM).

The example in [Figure 3-7](#) shows an example of a receive FIFO operation with a typical data width of 8 bits, representing a byte. In this example, six bytes are received via the receive input line. The receive FIFO interrupt trigger level RXFCON.RXFITL is set to 0011_B. Therefore, the first receive interrupt RIR is generated after the reception of Byte 3 (RXFIFO is filled with three messages).

After the reception of Byte 4, three bytes are read out of the receive FIFO. After this read operation, the RXFIFO still contains one message. Finally, the FIFO is cleared after reading the last message.

If the RXFIFO is full and additional data are received, the receive interrupt RIR will be generated and bit CON.RE is set, if CON.REN is not cleared. In this case, the data byte last written into the receive FIFO is overwritten. With the overrun condition, the receive FIFO filling level FSTAT.RXFFL is set to maximum. If a RB read operation is executed with the RXFIFO enabled but empty, a receive interrupt RIR will be generated. In this case, the receive FIFO filling level FSTAT.RXFFL is set to 0000_B.

If the RXFIFO is available but disabled (RXFCON.RXFEN = 0) the receive operation is functionally equivalent to the receive operation of the SSC module without FIFO.

The RXFIFO can be flushed or cleared by setting bit RXFCON.RXFFLU in register RXFCON. After this RXFIFO flush operation, the RXFIFO is empty and the receive FIFO filling level FSTAT.RXFFL is set to 0000_B.

The RXFIFO is flushed automatically with a reset operation of the SSC module and if the RXFIFO becomes disabled (resetting bit RXFCON.RXFEN) after it was previously

Synchronous Serial Interface (SSC)

enabled. Resetting bit CON.REN without resetting RXFCON.RXFEN does not affect (reset) the RXFIFO state. This means that the receive operation of the SSC is stopped, in this case, without changing the content of the RXFIFO. After setting CON.REN again, the RXFIFO with its content is again available.

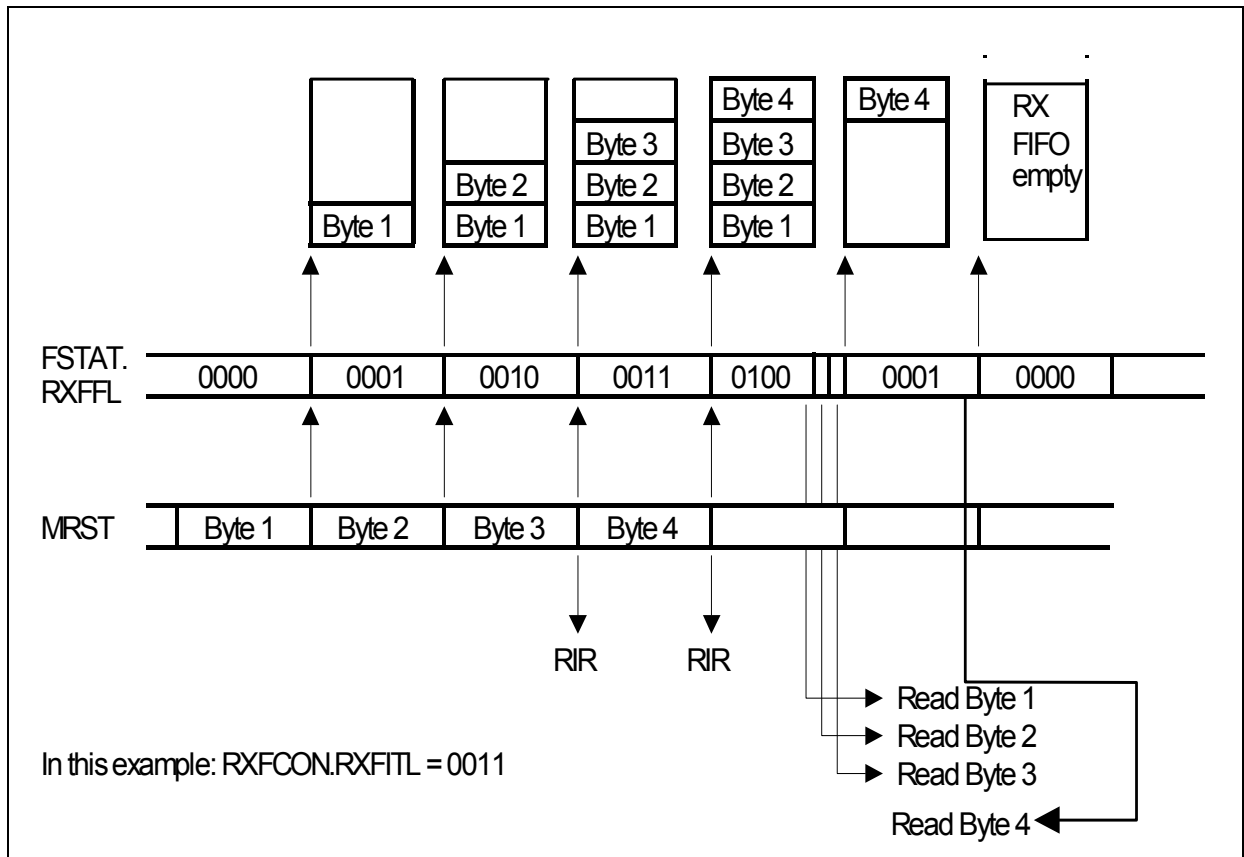


Figure 3-7 Receive FIFO Operation Example

3.1.2.8 FIFO Transparent Mode

In Transparent Mode, a specific interrupt generation mechanism is used for receive and transmit interrupts. In Transparent Mode, receive interrupts are always generated if data bytes are available in the RXFIFO. The relevant conditions for interrupt generation in Transparent Mode are:

- FIFO filling level
- Read/Write operations from/to the RB/TB data registers

Receive Operation

The interrupt generation for the receive FIFO depends on the RXFIFO filling level and the execution of read operations of register RB (see [Figure 3-8](#)). Transparent Mode for the RXFIFO is enabled when bits RXFCON.RXTMEN and RXFCON.RXFEN in register RXFCON are set.

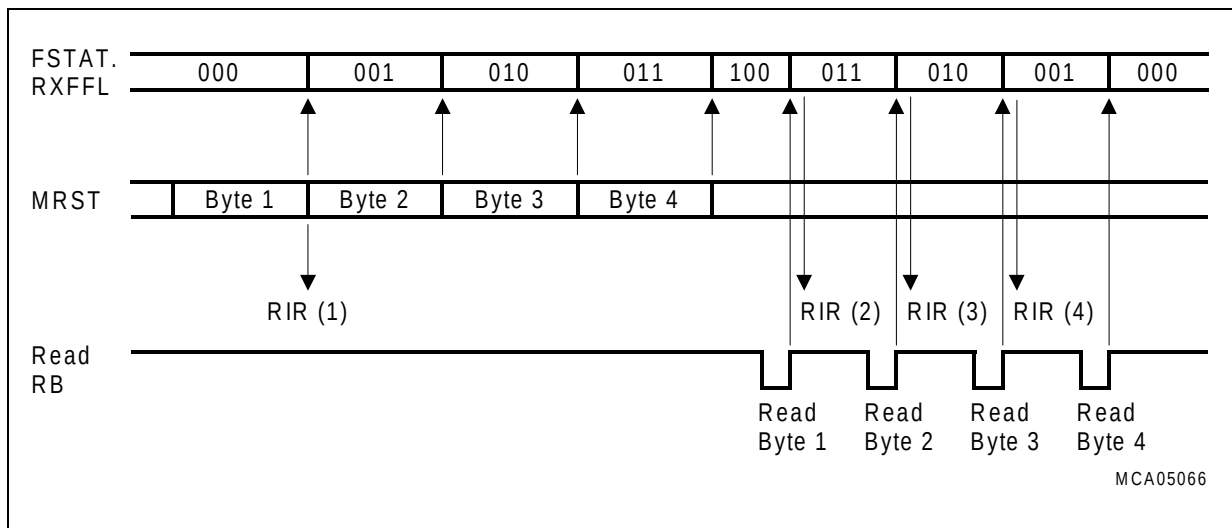


Figure 3-8 Transparent Mode Receive FIFO Operation

If the RXFIFO is empty, a receive interrupt RIR is always generated when the first message is written into an empty RXFIFO (FSTAT.RXFFL changes from 0000_B to 0001_B). If the RXFIFO is filled with at least one message, the occurrence of further receive interrupts depends on the read operations of register RB. The receive interrupt RIR will always be activated after a RB read operation if the RXFIFO still contains data (FSTAT.RXFFL is not equal to 0000_B). If the RXFIFO is empty after a RB read operation, no further receive interrupt will be generated.

If the RXFIFO is full (FSTAT.RXFFL = 1000_B) and additional messages are received, a receive interrupt RIR will be generated. In this case, the message last written into the receive FIFO is overwritten. If a RB read operation is executed with the RXFIFO enabled but empty (underflow condition), a receive interrupt RIR will be generated as well, with bit CON.RE set.

Synchronous Serial Interface (SSC)

If the RXFIFO is flushed in Transparent Mode, the software must take care that a previous pending receive interrupt is ignored.

Note: The Receive FIFO Interrupt Trigger Level bit field RXFCON.RXFITL is not applicable in Transparent Mode.

Transmit Operation

Interrupt generation for the transmit FIFO depends on the TXFIFO filling level and the execution of write operations to the register TB. Transparent Mode for the TXFIFO is enabled when bits TXFCON.TXTMEN and TXFCON.TXFEN are set.

TIR is also activated after a TXFIFO flush operation or when the TXFIFO becomes enabled (TXFCON.TXTMEN and TXFCON.TXFEN set) when it was previously disabled. In these cases, the TXFIFO is empty and ready to be filled with data.

If the TXFIFO is full ($FSTAT.TXFFL = 1000_B$) and an additional message is written into TB, a transmit interrupt will be generated after the TB write operation. In this case the data byte last written into the transmit FIFO is overwritten and a transmit interrupt (TIR) will be generated with bit CON.TE set.

Note: The Transmit FIFO Interrupt Trigger Level bit field TXFCON.TXFITL is not applicable in Transparent Mode.

3.1.2.9 Baud Rate Generation

The serial channel SSC has its own dedicated 16-bit baud rate generator with 16-bit reload capability, allowing baud rate generation independent from the timers. In addition to [Figure 3-2](#), [Figure 3-9](#) shows the baud rate generator of the SSC in more detail.

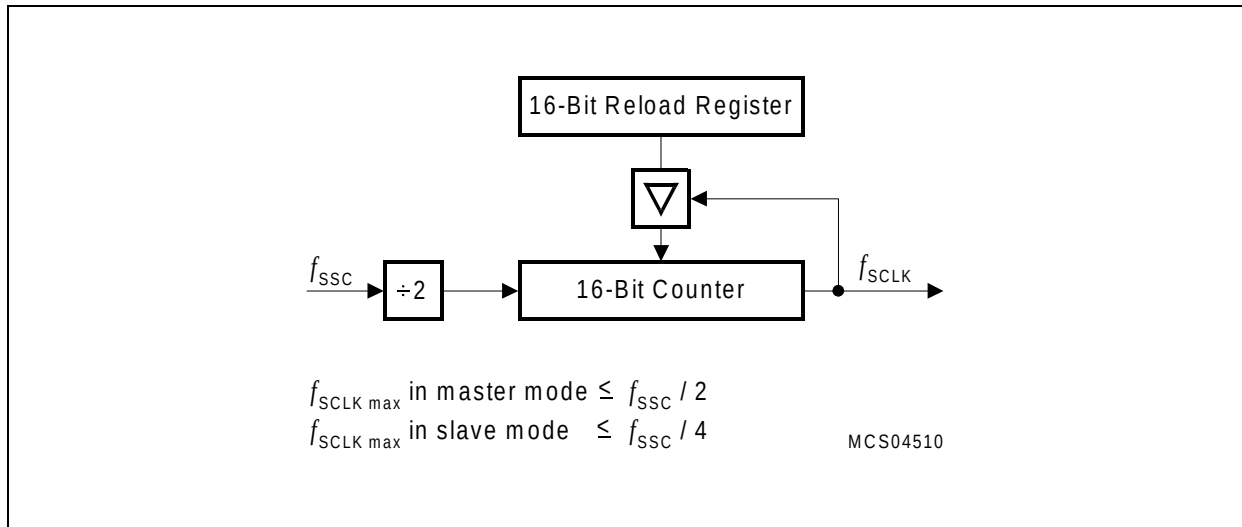


Figure 3-9 SSC Baud Rate Generator

The baud rate generator is clocked with the module clock f_{SSC} . The timer counts downwards. Register BR is the dual-function Baud Rate Generator/Reload register. Reading BR, while the SSC is enabled, returns the contents of the timer. Reading BR, while the SSC is disabled, returns the programmed reload value. In this mode, the desired reload value can be written to BR.

Note: Never write to BR while the SSC is enabled.

The formulas below calculate either the resulting baud rate for a given reload value, or the required reload value for a given baud rate:

$$\text{Baud rate}_{\text{SSC}} = \frac{f_{\text{SSC}}}{2 \times (\text{BR\_VALUE} + 1)} \quad \text{BR\_VALUE} = \frac{f_{\text{SSC}}}{2 \times \text{Baud rate}_{\text{SSC}}} - 1$$

BR_VALUE represents the content of the reload register, taken as unsigned 16-bit integer while Baud rate_{SSC} is equal to f_{SCLK} as shown in [Figure 3-9](#).

The maximum baud rate that can be achieved when using a module clock of 75 MHz is 37.5 MBaud in master mode (with
 = 0000_H) and 18.75 MBaud in slave mode (with
 = 0001_H).

[Table 3-1](#) lists some possible baud rates together with the required reload values and the resulting bit times, assuming a module clock of 75 MHz.

Synchronous Serial Interface (SSC)

Table 3-1 Typical Baud Rates of the SSC ($f_{SSC} = 75 \text{ MHz}$)

Reload Value	Baud Rate ($= f_{SCLK}$)	Deviation
0000 _H	37.5 MBaud (only in master mode)	0.0%
0001 _H	18.75 MBaud	0.0%
0025 _H	1 MBaud	-1.3%
0176 _H	100 kBaud	0.0%
0EA5 _H	10 kBaud	0.0%
9276 _H	1 kBaud	0.0%
FFFF _H	572.2 Baud	0.0%

3.1.2.10 Slave Select Input Operation

For systems with multiple slaves, the SSC module provides $\overline{\text{SLSI}}$ slave select input lines, that allow the enabling/disabling of the SCLK, MTSR, and MRST signals in slave mode. Slave mode is selected by $\text{CON.MS} = 0$. The $\overline{\text{SLSI}}$ input logic shown in [Figure 3-10](#) is controlled by register PISEL and CON.

With $\text{PISEL.SLSIS} = 0$ and slave mode selected, the $\overline{\text{SLSI}}[7:1]$ lines do not control the SSC I/O lines. The slave receive input signal at pins MTSRA or MTSRB and the slave clock signal at pin SCLKA or SCLKB are passed further to MTSRI and SCLKI. The slave transmit signal MRSTI is passed directly to MRST.

With $\text{PISEL.SLSIS} = 1$, slave select mode is enabled and input signals $\overline{\text{SLSI}}[7:1]$ control the operation of the SSC I/O lines as follows:

- $\overline{\text{SLSI}}_x = 1$: SSC slave is not selected.
 - MTSRI is connected with the slave receive input signals MTSRA or MTSRB, depending on PISEL.SRIS (slave mode receive input select).
 - MRST is driven with the logic level of bit PISEL.STIP (slave transmit idle state).
 - SCLKI is driven with the logic level of CON.PO (clock polarity control).
- $\overline{\text{SLSI}}_x = 0$: SSC is selected as slave.
 - MTSRI is connected with the slave receive input signals MTSRA or MTSRB, depending on PISEL.SRIS (slave mode receive input select).
 - MRST is directly driven with the slave transmit output signal MRSTI.
 - SCLKI is connected with the slave clock input signals SCLKA or SCLKB, depending on PISEL.SCIS (slave mode clock input select).

Synchronous Serial Interface (SSC)

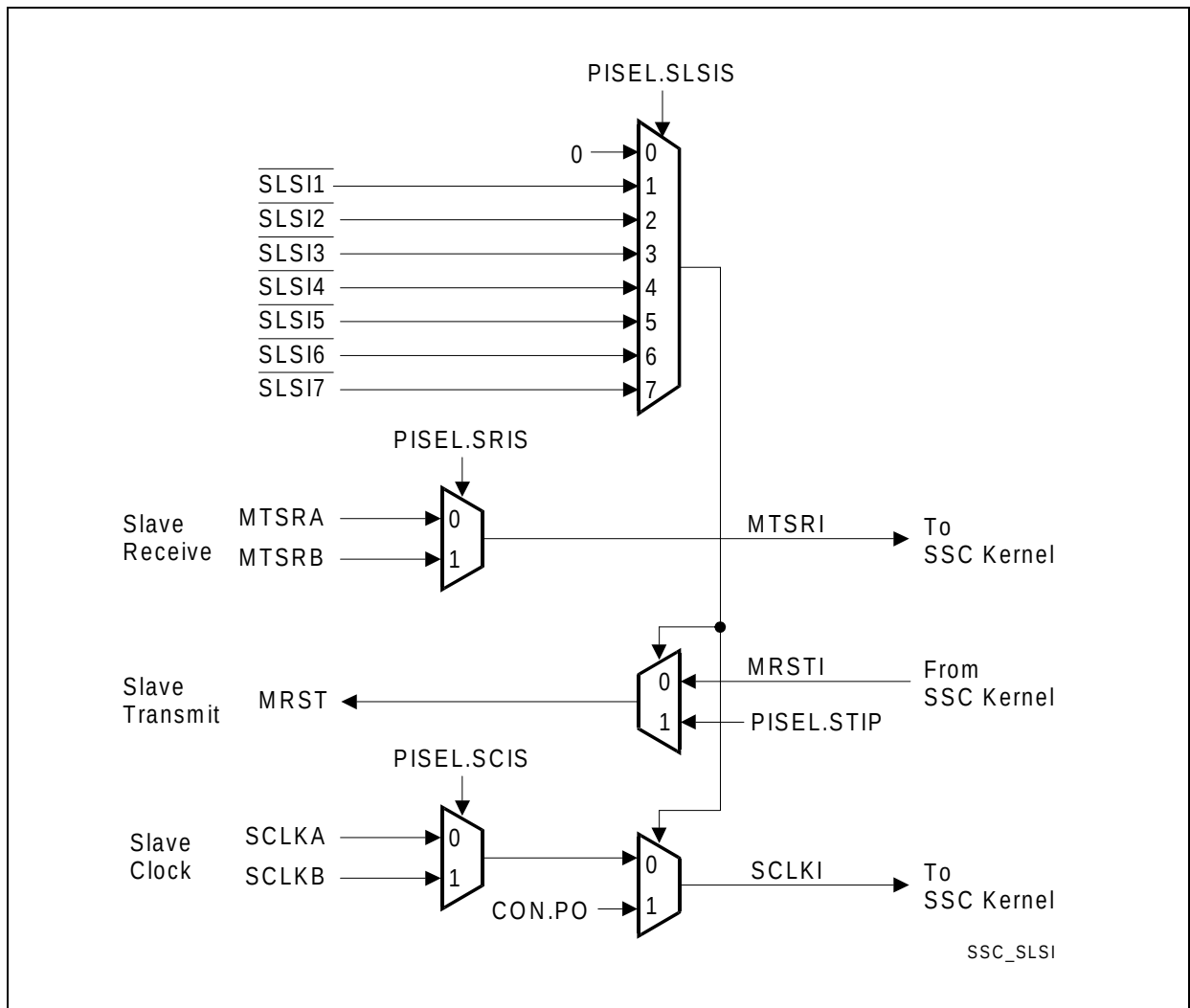


Figure 3-10 Slave Select Input Logic

3.1.2.11 Slave Select Output Generation Unit

In master mode the slave select output generation unit of the SSC provides an automatic generation of up to eight slave select output lines for serial transmit operations. The slave select output generation unit also makes it possible to adjust the chip select timing parameters. The active/inactive state of a slave select output as well as the enable/disable state can be controlled individually for each slave select output (see [Figure 3-12](#)). The basic slave select output timing is shown in [Figure 3-11](#), assuming a low active level of the SLSON lines.

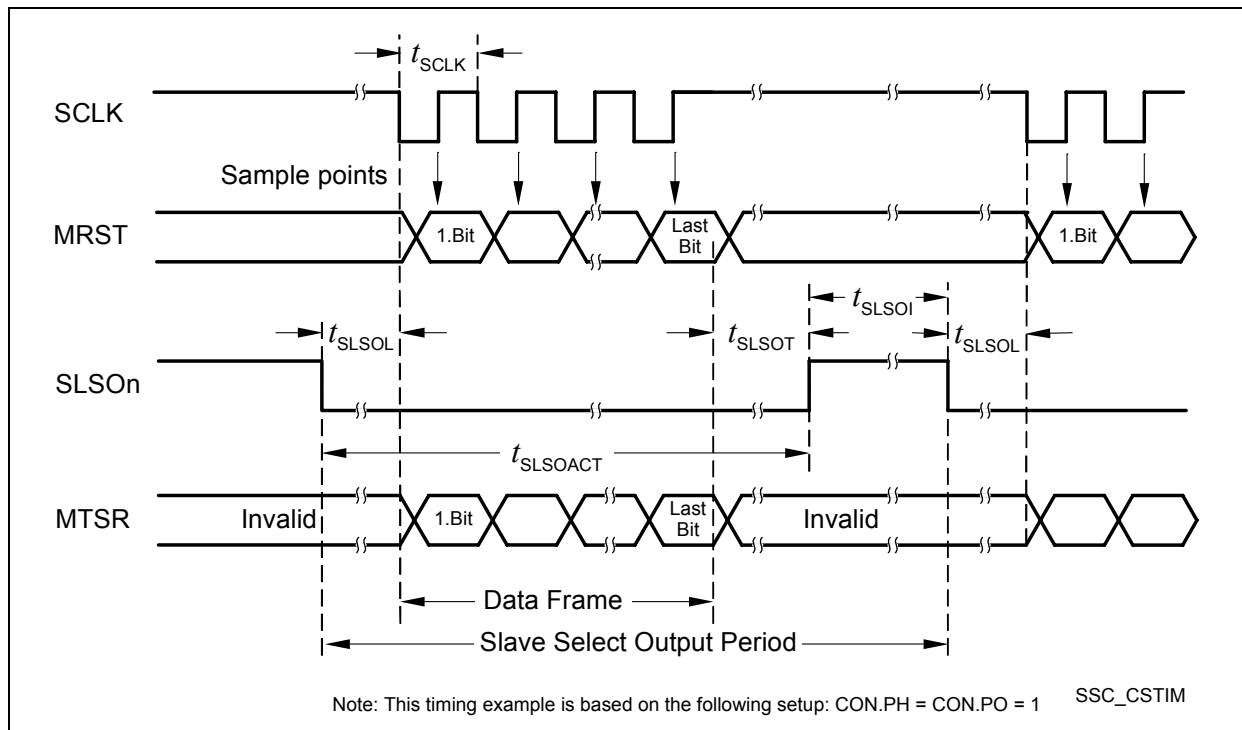


Figure 3-11 SSC Slave Select Output Timing

A slave select output period always starts after a serial write operation to register TB. Afterwards SLSON becomes active (low) for a number of SCLK cycles (leading delay cycles) before the first bit of the serial data stream occurs at MTSR. After the transmission of the data frame SLSON remains active (low) for a number of SCLK cycles (trailing delay cycles) before it becomes again inactive. This inactive state of SLSON is valid at least for a number of SCLK cycles (inactive delay cycles) before a new chip select period can be started.

The three parameters of a chip select period are controlled by bit fields in the slave select output timing control register SSOTC. Each of these bit fields can contain a value from 0 to 3 defining delay cycles of 0 to 3 multiples of the t_{SCLK} shift clock period. The three parameters are:

- Number of leading delay cycles ($t_{SLSOL} = SSOTC.LEAD \times t_{SCLK}$)
- Number of trailing delay cycles ($t_{SLSOT} = SSOTC.TRAIL \times t_{SCLK}$)

Synchronous Serial Interface (SSC)

- Number of inactive delay cycles ($t_{\text{SLSOI}} = \text{SSSOTC.INACT} \times t_{\text{SCLK}}$)

If $\text{SSOTC.INACT} = 00_{\text{B}}$ and register TB has already been loaded with the data for the next data frame, the next chip select period is started with its leading delay phase without SLSON going inactive. If, in this case, TB has not been loaded in time with the data for the next data frame, SLSOx becomes inactive again.

Slave Select Output Control

Each slave select output SLSON can be enabled individually. When $\text{SSOC.OENn} = 0$, SLSON is enabled. Further, active and inactive levels of the SLSON outputs are programmable. Bit SSOC.AOLn defines the state of the active level of SLSON .

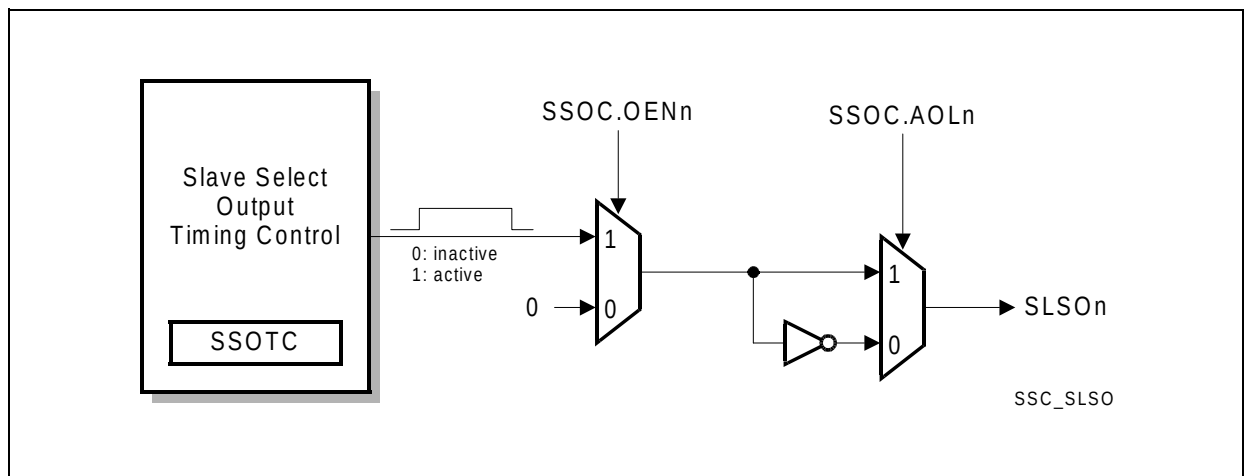


Figure 3-12 Slave Select Output Control Logic

Slave Select Output 7 Delayed Mode

In the SLSO7 delayed mode ($\text{SSOTC.SLSO7MOD} = 1$), the timing of the slave select output SLSO7 as programmed by the three parameters in SSOTC (number of trailing, leading, and inactive delay clock cycles) is delayed by one shift clock period for the inactive to active edge. The active to inactive edge is not delayed. The timing of SLSO7 in the delayed mode is shown in [Figure 3-13](#). The bold lines show the timing of SLSO7 in normal operating mode and the dotted lines show the timing of SLSO7 in delayed mode.

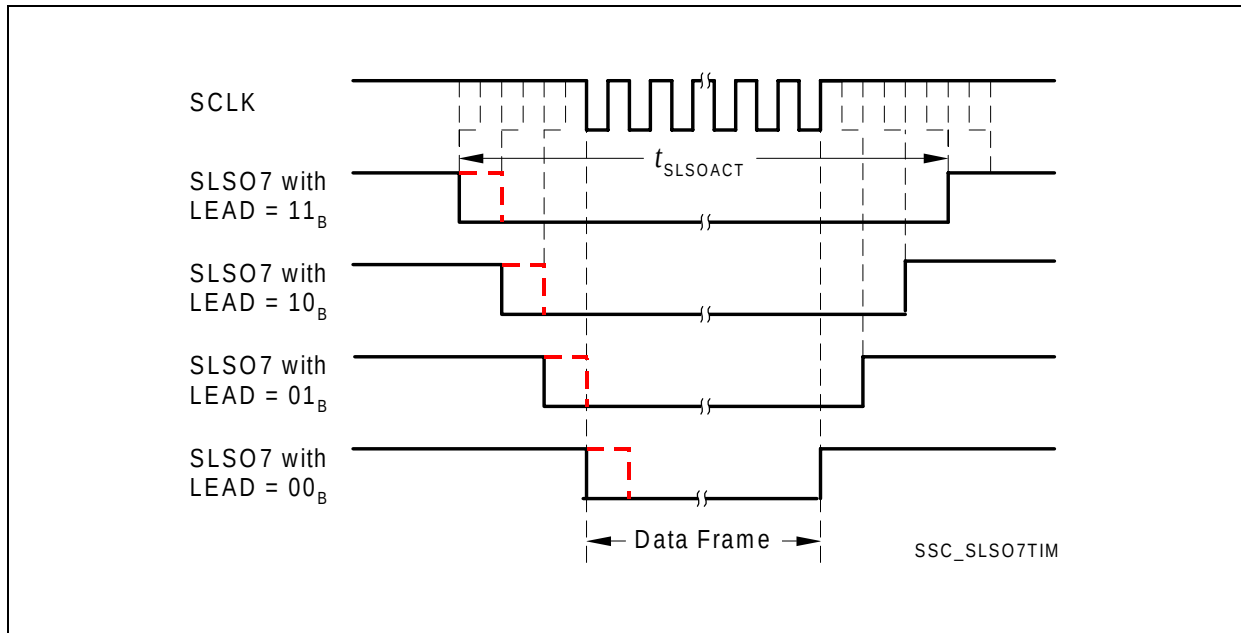


Figure 3-13 SLSO7 Delayed Mode

Slave Select Register Update

The bits in the registers SSOC and SSOTC are buffered while a transfer is in progress. The buffer samples the values written to these registers in the following case:

- Start of the internal transfer sequence

So it is always guaranteed that the data of one SSC transfer is transmitted with one constant slave select configuration and a configuration change is only valid with the start of the next new SSC transfer.

3.1.2.12 Shift Clock Generation

The serial channel SSC operates with its own shift clock f_{SSC} . The shift clock is generated outside the SSC module kernel in the clock control block using a fractional divider (see [Section 3.3.3](#)).

3.1.2.13 Error Detection Mechanisms

The SSC is able to detect four different error conditions. Receive Error and Phase Error are detected in all modes, while Transmit Error and Baud Rate Error apply to slave mode only. When an error is detected, the respective error flag is set and an error interrupt request will be generated by activating the EIR line (see [Figure 3-14](#)). The error interrupt handler may then check the error flags to determine the cause of the error interrupt. The error flags are not reset automatically, but must be cleared via register EFM after servicing. This allows servicing of some error conditions via interrupt, while others may be polled by software. The error status flags can be set and reset by software via the error flag modification register EFM.

Note: The error interrupt handler must clear the associated (enabled) error flag(s) to prevent repeated interrupt requests.

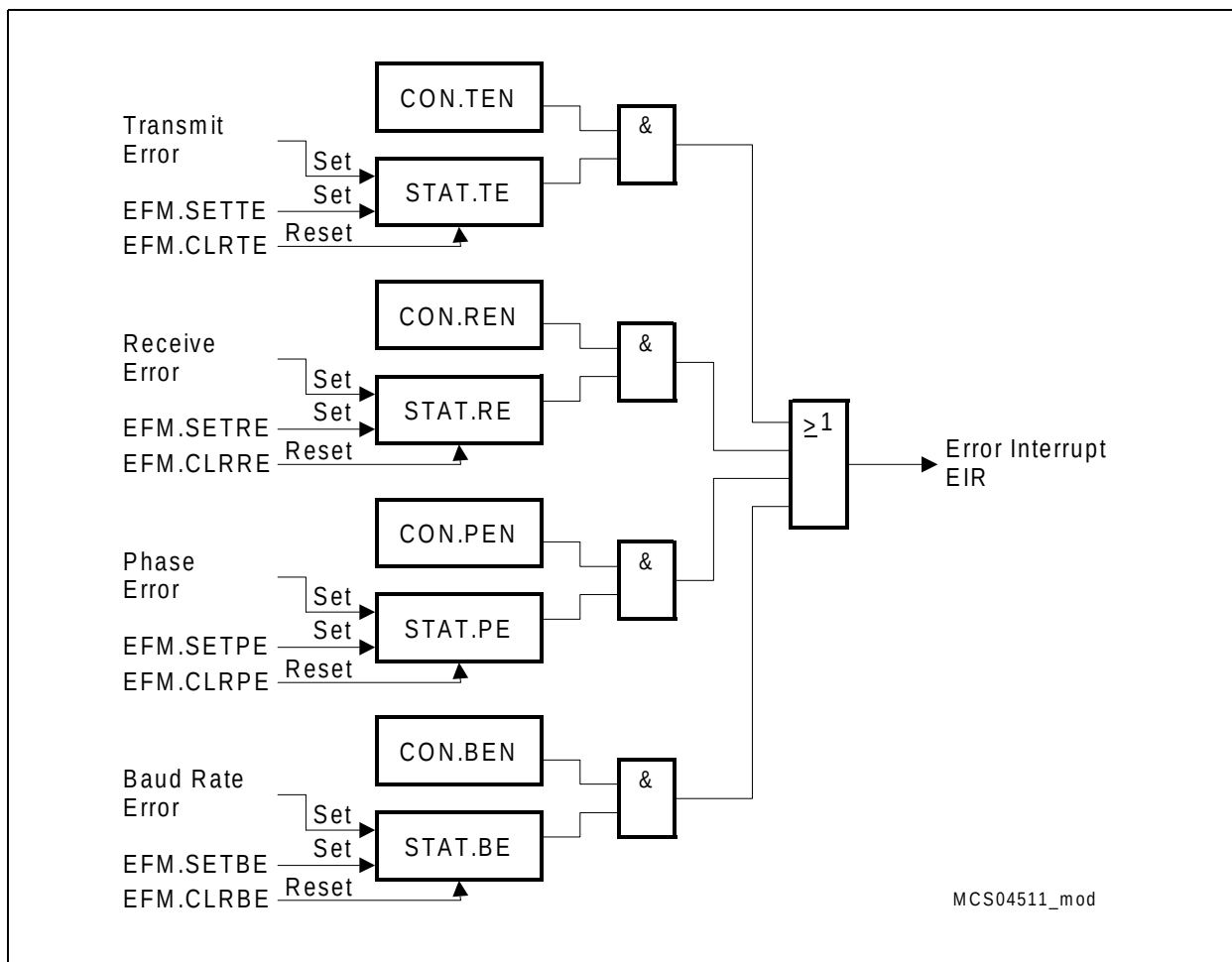


Figure 3-14 SSC Error Interrupt Control

A **Receive Error** (Master or Slave mode) is detected when a new data frame is completely received, but the previous data was not read out of the receive buffer register RB. This condition sets the error flag STAT.RE, when enabled via CON.REN, the error

Synchronous Serial Interface (SSC)

interrupt request activates the EIR line. The old data in the receive buffer RB will be overwritten with the new value and is unretrievably lost.

A **Phase Error** (Master or Slave mode) is detected when the incoming data at pin MRST (master mode) or MTSR (slave mode), sampled with the same frequency as the module clock, changes between one cycle before and two cycles after the latching edge of the shift clock signal SCLK. This condition sets the error status flag STAT.PE, when enabled via CON.PEN, the error interrupt request activates the EIR line.

A **Baud Rate Error** (Slave mode) is detected when the incoming clock signal deviates from the programmed baud rate (shift clock) by more than 100%, meaning it is either more than double or less than half the expected baud rate. This condition sets the error status flag STAT.BE, when enabled via CON.BEN, the error interrupt request activates the EIR line. Using this error detection capability requires that the slave's shift clock generator is programmed to the same baud rate as the master device. This feature detects false additional pulses or missing pulses on the clock line (within a certain frame).

Note: If this error condition occurs and bit CON.REN = 1, an automatic reset of the SSC will be performed in case of this error. This is done to re-initialize the SSC, if too few or too many clock pulses have been detected.

A **Transmit Error** (Slave mode) is detected when a transfer was initiated by the master (shift clock gets active), but the Transmit Buffer (TB) of the slave was not updated since the last transfer. This condition sets the error status flag STAT.TE, when enabled via CON.TEN, the error interrupt request activates the EIR line. If a transfer starts while the transmit buffer is not updated, the slave will shift out the 'old' contents of the shift register, which is normally the data received during the last transfer. This may lead to the corruption of the data on the transmit/receive line in Half-duplex Mode (open drain configuration) if this slave is not selected for transmission. This mode requires that slaves not selected for transmission only shift out ones, thus, their transmit buffers must be loaded with FFFF_H prior to any transfer.

Note: A slave with push/pull output drivers not selected for transmission, will normally have its output drivers switched. However, to avoid possible conflicts or misinterpretations, it is recommended to always load the slave's transmit buffer prior to any transfer.

The cause of an error interrupt request (receive, phase, baud rate, transmit error) can be identified by the error status flags in control register CON.

Note: In contrast to the error interrupt request line EIR, the error status flags STAT.TE, STAT.RE, STAT.PE, and STAT.BE, are not reset automatically upon entry into the error interrupt service routine, but must be cleared by software.

3.2 SSC Kernel Registers

Figure 3-15 and Table 3-2 shows all registers associated with the SSC Kernel.

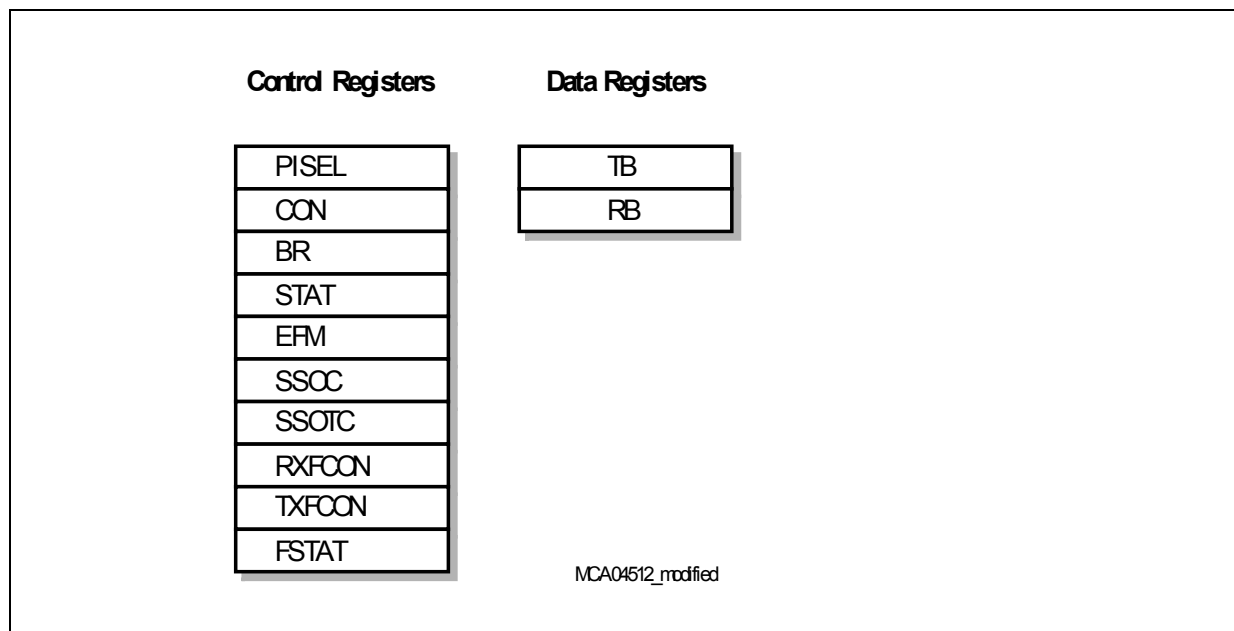


Figure 3-15 SSC Kernel Registers

Table 3-2 SSC Kernel Registers

Register Short Name	Register Long Name	Offset Address	Description see
PISEL	Port Input Select Register	0004 _H	Page 3-29
CON	Control Register	0010 _H	Page 3-31
BR	Baud Rate Timer Reload Register	0014 _H	Page 3-38
STAT	Status Register	0028 _H	Page 3-33
EFM	Error Flag Modification Register	002C _H	Page 3-34
SSOC	Slave Select Output Control Register	0018 _H	Page 3-36
SSOTC	Slave Select Output Timing Control Register	001C _H	Page 3-37
TB	Transmit Buffer Register	0020 _H	Page 3-39
RB	Receive Buffer Register	0024 _H	Page 3-39
RXFCON	Receive FIFO Control Register	0030 _H	Page 3-40
TXFCON	Transmit FIFO Control Register	0034 _H	Page 3-42
FSTAT	FIFO Status Register	0038 _H	Page 3-44

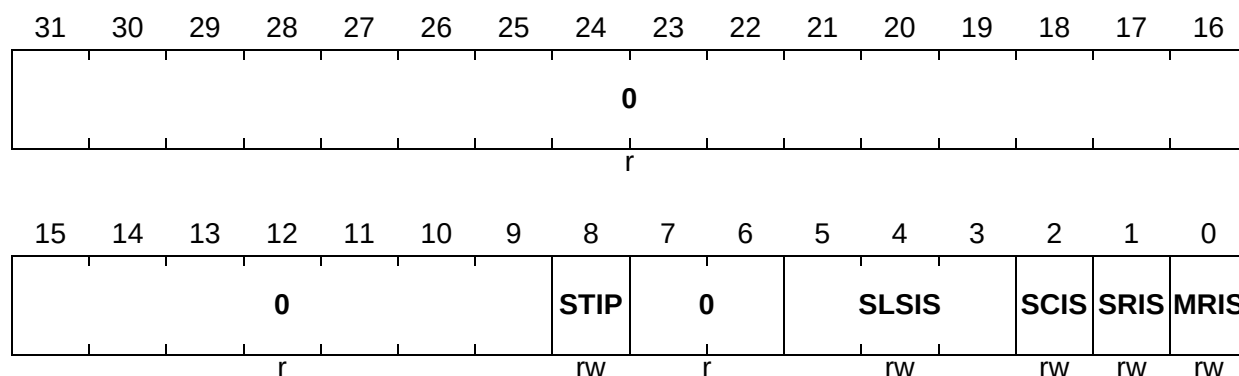
Synchronous Serial Interface (SSC)

The PISEL register controls the input signal selection of the SSC module. Each input of the module kernel receive, transmit and clock signals has associated two input lines (port A and port B).

PISEL

Port Input Select Register

Reset Value: 0000 0000_H



Field	Bits	Type	Description
MRIS	0	rw	Master Mode Receive Input Select MRIS selects the receive input line in master mode. 0 Receive input line MRSTA is selected 1 Receive input line MRSTB is selected
SRIS	1	rw	Slave Mode Receive Input Select SRIS selects receive input line that in slave mode. 0 Receive input line MTSRA is selected 1 Receive input line MTSRB is selected
SCIS	2	rw	Slave Mode Clock Input Select SCIS selects the module kernel SCLK input line that is used as clock input line in slave mode. 0 Slave mode clock input line SCLKA is selected 1 Slave mode clock input line SCLKB is selected

Synchronous Serial Interface (SSC)

Field	Bits	Type	Description
SLSIS	[5:3]	rw	Slave Mode Slave Select Input Selection 000 _B Slave select input lines are deselected; SSC is operating without slave select input functionality. 001 _B <u>SLSI1</u> input line is selected for operation 010 _B <u>SLSI2</u> input line is selected for operation 011 _B <u>SLSI3</u> input line is selected for operation 100 _B <u>SLSI4</u> input line is selected for operation 101 _B <u>SLSI5</u> input line is selected for operation 110 _B <u>SLSI6</u> input line is selected for operation 111 _B <u>SLSI7</u> input line is selected for operation
STIP	8	rw	Slave Transmit Idle State Polarity This bit defines the logic level of the slave mode transmit signal MRST when the SSC is deselected (PISEL.SLSIS = 0). 0 MRST = 0 when SSC is deselected in slave mode. 1 MRST = 1 when SSC is deselected in slave mode.
0	[7:6], [31:9]	r	Reserved ; read as 0; should be written with 0.

Synchronous Serial Interface (SSC)

The operating modes of the SSC are controlled by the control register CON. This register contains control bits for mode and error check selection.

CON

Control Register

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EN	MS	0	A REN	BEN	PEN	REN	TEN	LB	PO	PH	HB	BM			
rw	rw	r	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw			

Field	Bits	Type	Description
BM	[3:0]	rw	Data Width Selection BM defines the number of data bits of the serial frame. 0000 Reserved; do not use this combination. 0001 to 1111 Transfer Data Width is 2 ... 16 bit (<BM> + 1)
HB	4	rw	Heading Control 0 Transmit/Receive LSB First 1 Transmit/Receive MSB First
PH	5	rw	Clock Phase Control 0 Shift transmit data on the leading clock edge, latch on trailing edge 1 Latch receive data on leading clock edge, shift on trailing edge
PO	6	rw	Clock Polarity Control 0 Idle clock line is low, the leading clock edge is low-to-high transition 1 Idle clock line is high, the leading clock edge is high-to-low transition
LB	7	rw	Loop Back Control 0 Normal output 1 Receive input is connected with transmit output (Half-duplex Mode)

Synchronous Serial Interface (SSC)

Field	Bits	Type	Description
TEN	8	rw	Transmit Error Enable 0 Ignore transmit errors 1 Check transmit errors
REN	9	rw	Receive Error Enable 0 Ignore receive errors 1 Check receive errors
PEN	10	rw	Phase Error Enable 0 Ignore phase errors 1 Check phase errors
BEN	11	rw	Baud Rate Error Enable 0 Ignore baud rate errors 1 Check baud rate errors
AREN	12	rw	Automatic Reset Enable 0 No additional action upon a baud rate error 1 SSC is automatically reset on a baud rate error
MS	14	rw	Master Select 0 Slave Mode. Operate on shift clock received via SCLK 1 Master Mode. Generate shift clock and output it via SCLK The inverted state of this bit is available at the M/S select output line.
EN	15	rw	Enable Bit 0 Transmission and reception is disabled. 1 Transmission and reception is enabled. This bit is available at the enable output line.
0	13, [31:16]	r	Reserved ; read as 0; should be written with 0.

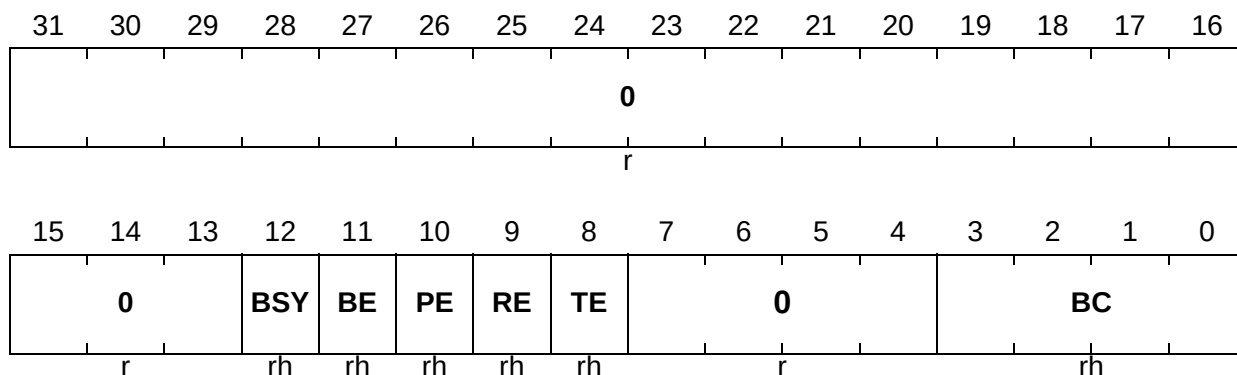
Synchronous Serial Interface (SSC)

The status register STAT contains status flags for error identification, the busy flag, and a bit field that indicates the current shift counter status.

STAT

Status Register

Reset Value: 0000 0000_H



Field	Bits	Type	Description
BC	[3:0]	rh	Bit Count Status BC indicates the current status of the shift counter. The shift counter is updated with every shifted bit.
TE	8	rh	Transmit Error Flag 0 No error 1 Transfer starts with the slave's transmit buffer not being updated
RE	9	rh	Receive Error Flag 0 No error 1 Reception completed before the receive buffer was read
PE	10	rh	Phase Error Flag 0 No error 1 Received data changes around the sampling clock edge
BE	11	rh	Baud Rate Error Flag 0 No error 1 More than factor 2 or 0.5 between slave's actual and expected baud rate
BSY	12	rh	Busy Flag BSY is set while a transfer is in progress.
0	[7:4], [31:13]	r	Reserved ; read as 0; should be written with 0.

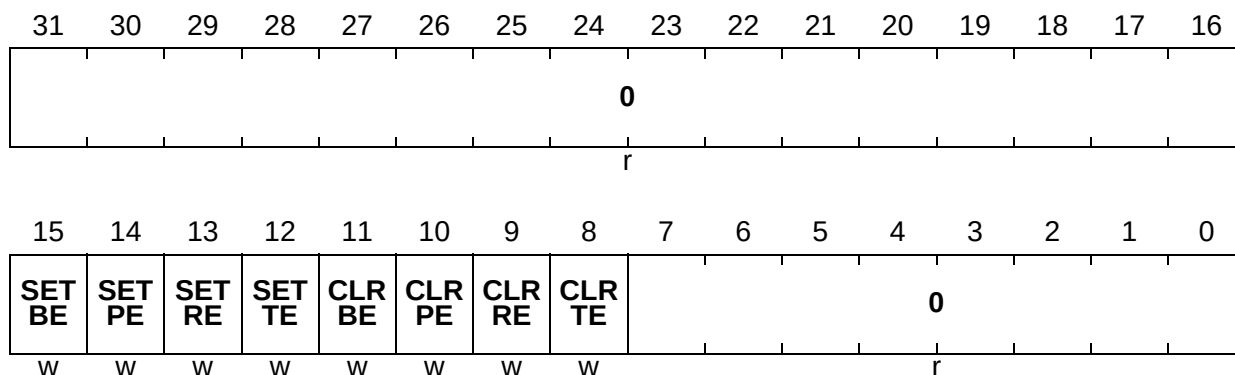
Synchronous Serial Interface (SSC)

The error flag modification register EFM is required for resetting or setting the four error flags which are located in register CON.

EFM

Error Flag Modification Register

Reset Value: 0000 0000_H



Field	Bits	Type	Description
CLRTE	8	w	Clear Transmit Error Flag Bit 0 No effect. 1 Bit CON.TE is cleared. Bit is always read as 0.
CLRRE	9	w	Clear Receive Error Flag Bit 0 No effect. 1 Bit CON.RE is cleared. Bit is always read as 0.
CLRPE	10	w	Clear Phase Error Flag Bit 0 No effect. 1 Bit CON.PE is cleared. Bit is always read as 0.
CLRBE	11	w	Clear Baud Rate Error Flag Bit 0 No effect. 1 Bit CON.BE is cleared. Bit is always read as 0.
SETTE	12	w	Set Transmit Error Flag Bit 0 No effect. 1 Bit CON.TE is set. Bit is always read as 0.
SETRE	13	w	Set Receive Error Flag Bit 0 No effect. 1 Bit CON.RE is set. Bit is always read as 0.

Synchronous Serial Interface (SSC)

Field	Bits	Type	Description
SETPE	14	w	Set Phase Error Flag Bit 0 No effect. 1 Bit CON.PE is set. Bit is always read as 0.
SETBE	15	w	Set Baud Rate Error Flag Bit 0 No effect. 1 Bit CON.BE is set. Bit is always read as 0.
0	[7:0], [31:16]	r	Reserved ; read as 0; should be written with 0.

Note: When the set and clear bit for an error flag is set at the same time during an EFM write operation (e.g SETPE = CLRPE = 1), the error flag in STAT is not affected. The chip select control register controls the operation of the chip select generation unit.

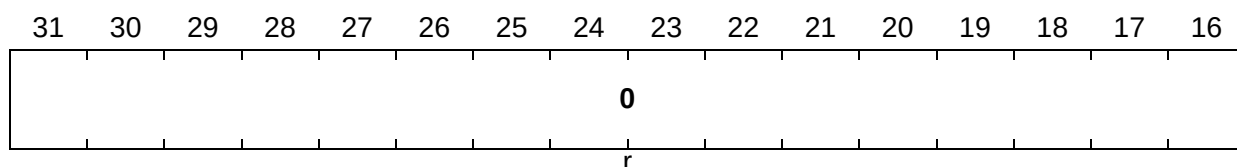
Synchronous Serial Interface (SSC)

The chip select control register controls the operation of the chip select generation unit.

SSOC

Slave Select Output Control Register

Reset Value: 0000 0000_H



15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OEN 7	OEN 6	OEN 5	OEN 4	OEN 3	OEN 2	OEN 1	OEN 0	AOL 7	AOL 6	AOL 5	AOL 4	AOL 3	AOL 2	AOL 1	AOL 0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Field	Bits	Type	Description
AOLn (n = 0-7)	n	rw	Active Output Level 0 SLSON is at low level during the chip select active time $t_{SLSOACT}$. The high level is the inactive level of SLSON. 1 SLSON line n is at high level during the chip select active time $t_{SLSOACT}$. The low level is the inactive level of SLSON.
OENn (n = 0-7)	8 + n	rw	Output n Enable Control 0 SLSON output is disabled; SLSON is always at inactive level as defined by AOLn. 1 SLSON output is enabled
0	[31:16]	r	Reserved ; read as 0; should be written with 0.

Note: This register is buffered during a transfer.

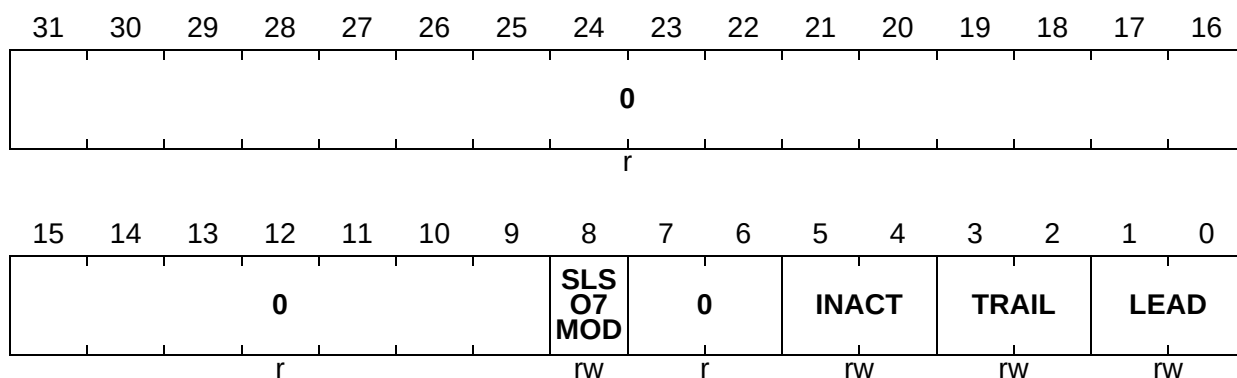
Synchronous Serial Interface (SSC)

The chip select control register controls the operation of the chip select generation unit.

SSOTC

Slave Select Output Timing Control Register

Reset Value: 0000 0000_H



Field	Bits	Type	Description
LEAD	[1:0]	rw	Slave Output Select Leading Delay This bit field defines the number of leading 00 Zero leading delay clock cycles selected 01 One leading delay clock cycle selected 10 Two leading delay clock cycles selected 11 Three leading delay clock cycles selected A leading delay clock cycle is always a multiple of an SCLK shift clock period.
TRAIL	[3:2]	rw	Slave Output Select Trailing Delay 00 Zero trailing delay clock cycles selected 01 One trailing delay clock cycle selected 10 Two trailing delay clock cycles selected 11 Three trailing delay clock cycles selected A trailing delay clock cycle is always a multiple of an SCLK shift clock period.
INACT	[5:4]	rw	Slave Output Select Inactive Delay 00 Zero inactive delay clock cycles selected 01 One inactive delay clock cycle selected 10 Two inactive delay clock cycles selected 11 Three inactive delay clock cycles selected A inactive delay clock cycle is always a multiple of an SCLK shift clock period.

Synchronous Serial Interface (SSC)

Field	Bits	Type	Description
SLSO7MOD	8	rw	SLSO7 Delayed Mode Selection This bit selects the delayed mode for the SLSO7 slave select output. 0 Normal mode selected for SLSO7 1 Delayed mode selected for SLSO7
0	[7:6], [31:9]	r	Reserved ; read as 0; should be written with 0.

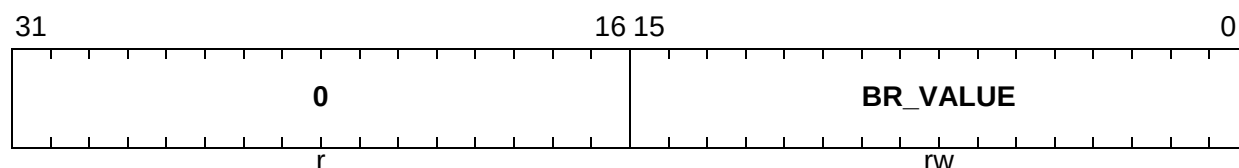
Note: This register is buffered during a transfer.

The SSC baud rate timer reload register BR contains the 16-bit reload value for the baud rate timer.

BR

Baud Rate Timer Reload Register

Reset Value: 0000 0000_H



Field	Bits	Type	Description
BR_VALUE	[15:0]	rw	Baud Rate Timer/Reload Register Value Reading BR returns the 16-bit content of the baud rate timer. Writing BR loads the baud rate timer reload register with BR_VALUE.
0	[31:16]	r	Reserved ; read as 0; should be written with 0.

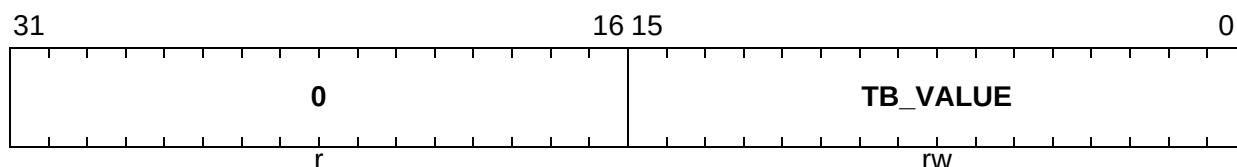
Synchronous Serial Interface (SSC)

The SSC transmit buffer register TB contains the transmit data value.

TB

Transmit Buffer Register

Reset Value: 0000 0000_H



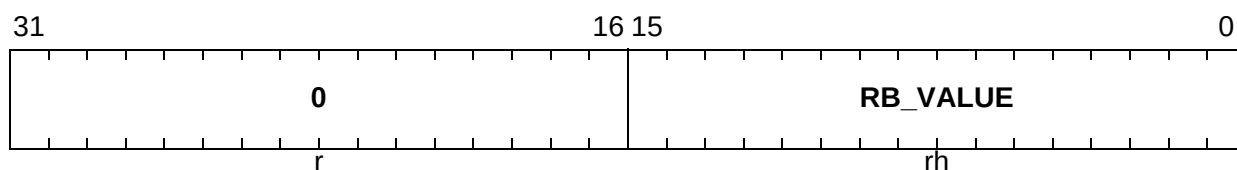
Field	Bits	Type	Description
TB_VALUE	[15:0]	rw	Transmit Data Register Value TB_VALUE is the data value to be transmitted. Unselected bits of TB are ignored during transmission.
0	[31:16]	r	Reserved ; read as 0; should be written with 0.

The SSC receive buffer register RB contains the receive data value.

RB

Receive Buffer Register

Reset Value: 0000 0000_H



Field	Bits	Type	Description
RB_VALUE	[15:0]	rh	Receive Data Register Value RB contains the received data value RB_VALUE. Unselected bits of RB will be not valid and should be ignored.
0	[31:16]	r	Reserved ; read as 0; should be written with 0.

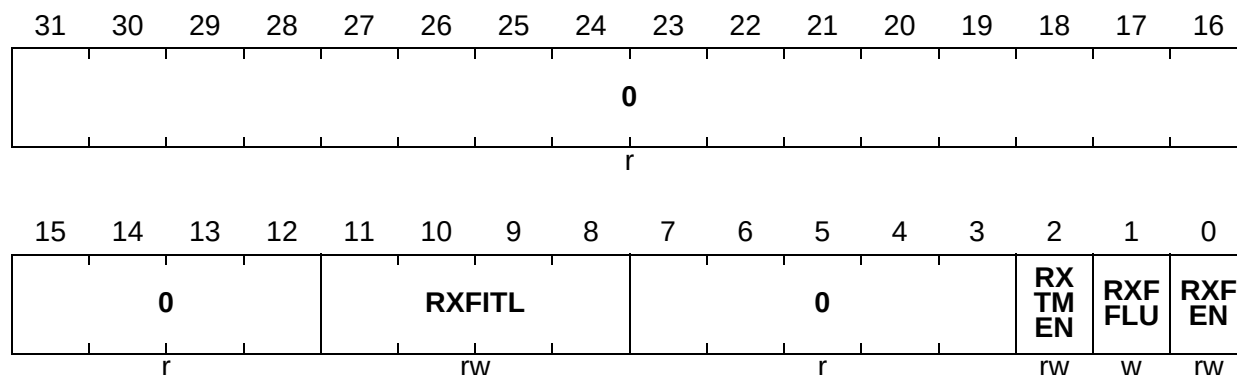
Synchronous Serial Interface (SSC)

The receive FIFO control register RXFIFO contains control bits and bit fields that define the operating mode of the receive FIFO.

RXFCON

Receive FIFO Control Register

Reset Value: 0000 0100_H



Field	Bits	Type	Description
RXFEN	0	rw	Receive FIFO Enable 0 Receive FIFO is disabled 1 Receive FIFO is enabled <i>Note: Resetting RXFEN automatically flushes the receive FIFO.</i>
RXFFLU	1	w	Receive FIFO Flush 0 No operation 1 Receive FIFO is flushed <i>Note: Setting RXFFLU clears bit field FSTAT.RXFFL. Bit RXFFLU is always read as 0.</i>
RXTMEN	2	rw	Receive FIFO Transparent Mode Enable 0 Receive FIFO Transparent Mode is disabled 1 Receive FIFO Transparent Mode is enabled <i>Note: This bit is not applicable if the receive FIFO is disabled (RXFEN = 0).</i>

Synchronous Serial Interface (SSC)

Field	Bits	Type	Description
RXFITL	[11:8]	rw	Receive FIFO Interrupt Trigger Level Defines the receive FIFO interrupt trigger level. A receive interrupt request (RIR) is always generated after the reception of a byte when the filling level of the receive FIFO is equal to or greater than RXFITL. 0000 _B Reserved. 0001 _B Interrupt trigger level is set to one 0010 _B Interrupt trigger level is set to two 0011 _B Interrupt trigger level is set to three 0100 _B Interrupt trigger level is set to four Others: reserved <i>Note: In Transparent Mode, this bit field is not applicable.</i>
0	[7:3], [31:12]	r	Reserved ; read as 0; should be written with 0.

Synchronous Serial Interface (SSC)

The transmit FIFO control register TXFIFO contains control bits and bit fields that define the operating mode of the transmit FIFO.

TXFCON

Transmit FIFO Control Register

Reset Value: 0000 0100_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0				TXFITL				0				TX TM EN		TXF FLU	TXF EN
r				rw				r				rw		w	rw

Field	Bits	Type	Description
TXFEN	0	rw	Transmit FIFO Enable 0 Transmit FIFO is disabled 1 Transmit FIFO is enabled <i>Note: Resetting TXFEN automatically flushes the transmit FIFO.</i>
TXFFLU	1	w	Transmit FIFO Flush 0 No operation 1 Transmit FIFO is flushed <i>Note: Setting TXFFLU clears bit field FSTAT.TXFFL. Bit TXFFLU is always read as 0.</i>
TXTMEN	2	rw	Transmit FIFO Transparent Mode Enable 0 Transmit FIFO Transparent Mode is disabled 1 Transmit FIFO Transparent Mode is enabled <i>Note: This bit is not applicable if the transmit FIFO is disabled (TXFEN = 0).</i>

Synchronous Serial Interface (SSC)

Field	Bits	Type	Description
TXFITL	[11:8]	rw	Transmit FIFO Interrupt Trigger Level Defines a transmit FIFO interrupt trigger level. A transmit interrupt request (TIR) is always generated after the transfer of a byte when the filling level of the transmit FIFO is equal to or lower than TXFITL. 0000 _B Reserved. 0001 _B Interrupt trigger level is set to one 0010 _B Interrupt trigger level is set to two 0011 _B Interrupt trigger level is set to three 0100 _B Interrupt trigger level is set to four Others: reserved <i>Note: In Transparent Mode, this bit field is not applicable.</i>
0	[7:3], [31:12]	r	Reserved ; read as 0; should be written with 0.

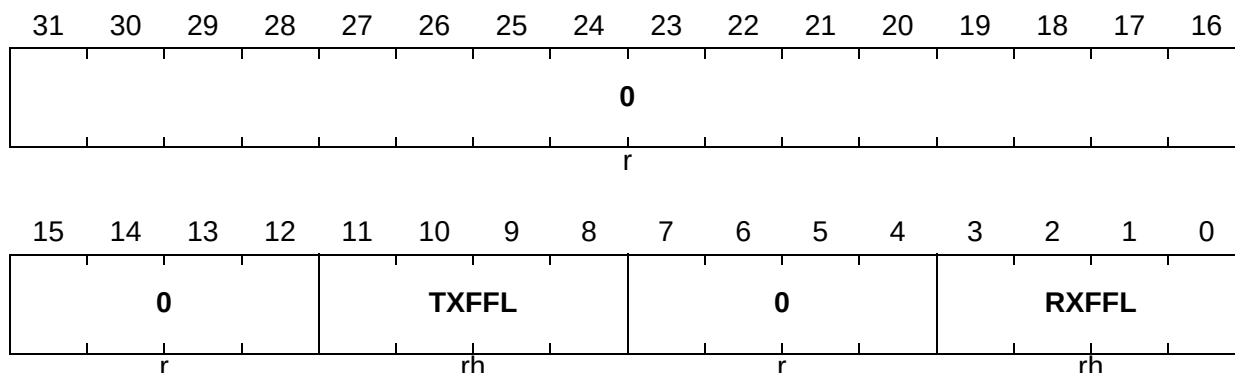
Synchronous Serial Interface (SSC)

The FIFO status register FSTAT indicates the filling levels of the receive and transmit FIFOs.

FSTAT

FIFO Status Register

Reset Value: 0000 0000_H



Field	Bits	Type	Description
RXFFL	[3:0]	rh	Receive FIFO Filling Level¹⁾ This bit field indicates the filling level of the RXFIFO. 0000 _B Receive FIFO is filled with zero byte 0001 _B Receive FIFO is filled with one byte 0010 _B Receive FIFO is filled with two bytes 0011 _B Receive FIFO is filled with three bytes 0100 _B Receive FIFO is filled with four bytes Others: reserved <i>Note: RXFFL is cleared after a receive FIFO flush operation.</i>
TXFFL	[11:8]	rh	Transmit FIFO Filling Level¹⁾ This bit field indicates the filling level of the TXFIFO. 0000 _B Transmit FIFO is filled with zero byte 0001 _B Transmit FIFO is filled with one byte 0010 _B Transmit FIFO is filled with two bytes 0011 _B Transmit FIFO is filled with three bytes 0100 _B Transmit FIFO is filled with four bytes Others: reserved <i>Note: TXFFL is cleared after a transmit FIFO flush operation.</i>
0	[7:4], [31:12]	r	Reserved ; read as 0; should be written with 0.

¹⁾ The data width of a RXFIFO and TXFIFO stage can be programmed from 2 to 15 bits. The data width “byte” mentioned in this description represents a data width of 8 bits.

3.3 SSC0/SSC1 Module Implementation

This section describes SSC0/SSC1 module interfaces with the clock control, port connections, interrupt control, and address decoding.

3.3.1 Interfaces of the SSC Modules

Figure 3-16 shows the TC1100 specific implementation details and interconnections of the SSC0/SSC1 modules. Each of the SSC modules is further supplied by clock control, interrupt control, address decoding, and port control logic. Two DMA requests can be generated by each SSC module.

Synchronous Serial Interface (SSC)

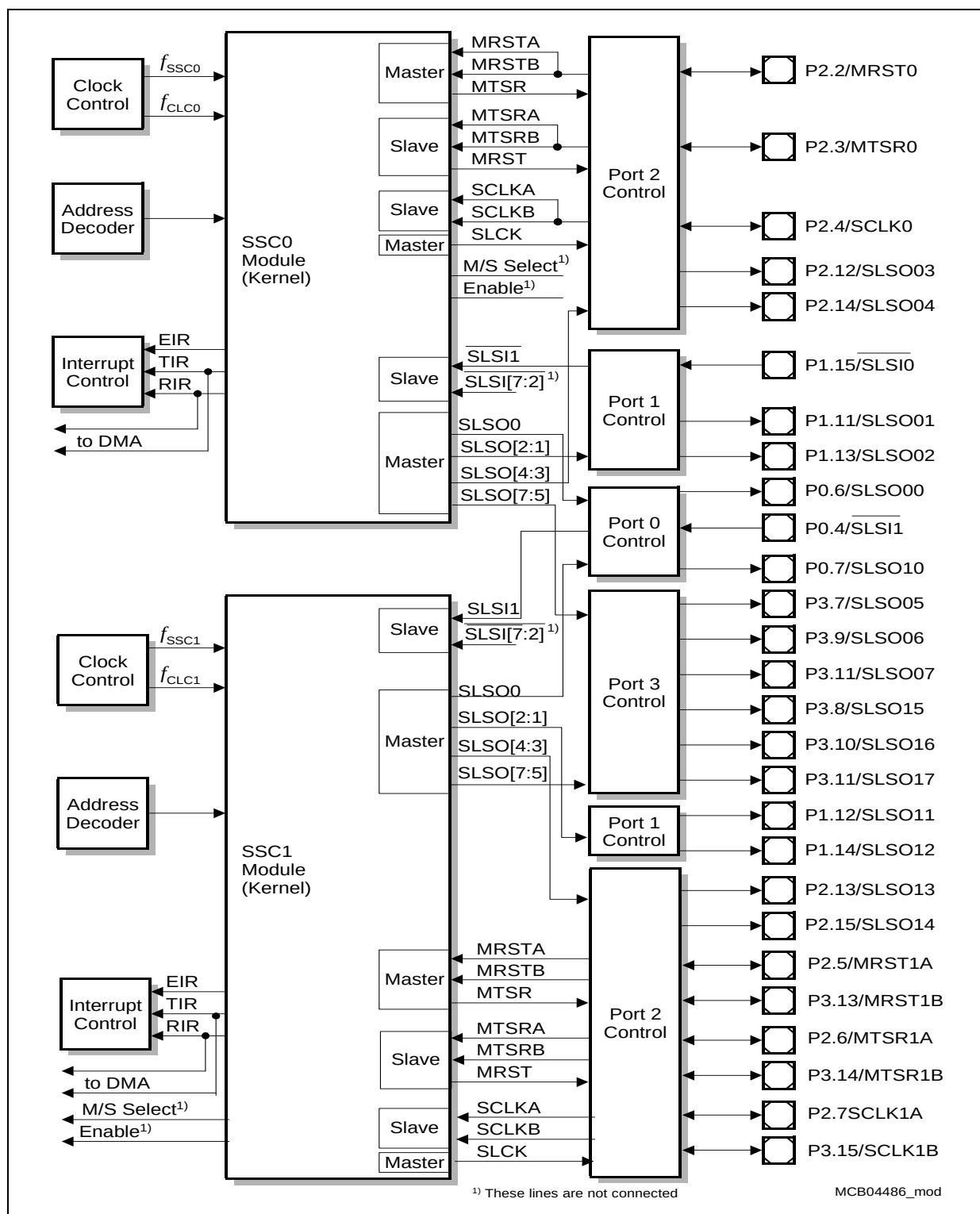


Figure 3-16 SSC0/SSC1 Module Implementation and Interconnections

3.3.2 SSC0/SSC1 Module Related External Registers

Figure 3-17 summarizes the module related external registers which are required for SSC0/SSC1 programming (see also **Figure 3-15** for the module kernel specific registers).

Clock Control Registers	Port Registers	Interrupt Registers
SSC0_CLC	P0_DIR	SSC0_TSRC
SSC1_CLC	P0_ALTSEL0	SSC0_RSRC
SSC0_FDR	P0_ALTSEL1	SSC0_ESRC
SSC1_FDR	P0_PUDSEL	SSC1_TSRC
	P0_PUDEN	SSC1_RSRC
	P0_OD	SSC1_ESRC
	P1_DIR	
	P1_ALTSEL0	
	P1_ALTSEL1	
	P1_PUDSEL	
	P1_PUDEN	
	P1_OD	
	P2_DIR	
	P2_ALTSEL0	
	P2_ALTSEL1	
	P2_PUDSEL	
	P2_PUDEN	
	P2_OD	
	P3_DIR	
	P3_ALTSEL0	
	P3_ALTSEL1	
	P3_PUDSEL	
	P3_PUDEN	
	P3_OD	

MCA04514_mod

Figure 3-17 SSC0/SSC1 Implementation Specific Special Function Registers

3.3.3 Clock Control

The SSC modules are provided each with two clock signals:

- f_{CLC0} and f_{CLC1}
This is the module clock that is used inside the SSC kernel for control purposes such as e.g. for clocking of control logic and register operations. The frequency of f_{CLC0} and f_{CLC1} is always identical to the system clock frequency f_{SYS} . The clock control registers SSC0_CLC and SSC1_CLC allow the enabling/disabling of f_{CLC0} and f_{CLC1} under certain conditions.
- f_{SSC0} and f_{SSC1}
This clock is the module clock that is used in the SSC as input clock of the baud rate generator which finally defines the baud rate of the serial data. The fractional divider registers SSC0_FDR and SSC1_FDR control the frequency of f_{SSC0} and f_{SSC1} and allow them to be enabled/disabled independently of f_{CLC0} and f_{CLC1} . The baud rate timer reload register SSC0_BR and SSC1_BR define serial data baud rate dependent from the frequency of f_{SSC0} and f_{SSC1} .

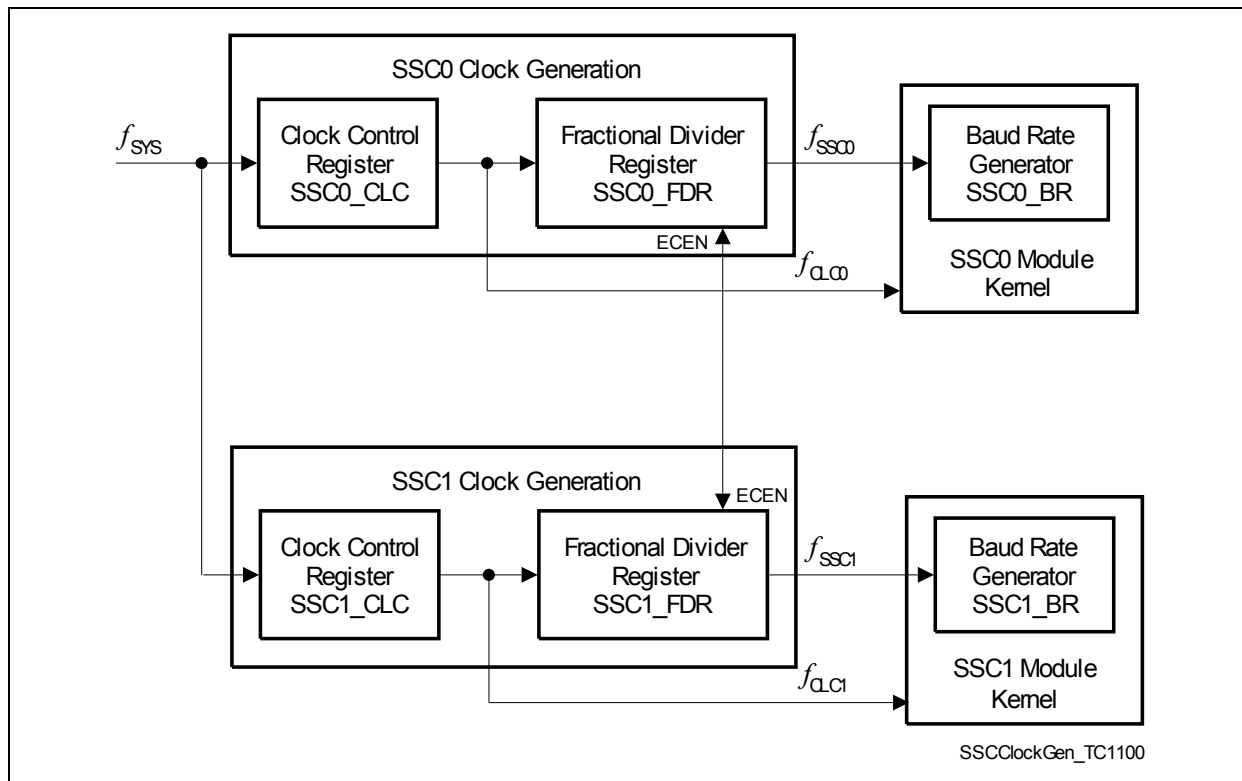


Figure 3-18 SSC Clock Generation

Synchronous Serial Interface (SSC)

The following formulas define the frequency of f_{SSC0} or f_{SSC1} :

$$f_{SSCx} = f_{SYS} \times \frac{1}{n} \quad \text{with } n = 1024 - \text{FDR.STEP}$$

$$\text{or } f_{SSCx} = f_{SYS} \times \frac{n}{1024} \quad \text{with } n = 0-1023$$

Note: In SSC master mode, the maximum shift clock frequency is $f_{SSCx}/2$. In SSC slave mode the maximum shift clock frequency is $f_{SSCx}/4$.

Combined with the formulas of the baud rate generator (see [Page 3-19](#)) and the fractional divider (see chapter “System Control Unit” of the *TC1100 System Units User’s Manual*), the resulting serial data baud rate is defined by:

$$\text{Baud rate}_{SSC} = \frac{f_{SYS}}{2 \times (\text{BR.BR\_VALUE} + 1) \times (1024 - \text{FDR.STEP})}$$

$$\text{Baud rate}_{SSC} = \frac{f_{SYS} \times \text{FDR.STEP}}{2 \times (\text{BR.BR\_VALUE} + 1) \times 1024}$$

with $\text{FDR.STEP} = 0-1023$

Note: The upper formula applies to normal divider mode of the fractional divider ($\text{FDR.DM} = 01_B$). The lower formula applies to fractional divider mode ($\text{FDR.DM} = 10_B$).

Synchronous Serial Interface (SSC)

The clock control registers allow the control (enable/disable) of the clock signals f_{CLC0} and f_{CLC1} under certain conditions. Each SSC has its own clock control register.

SSC0_CLC

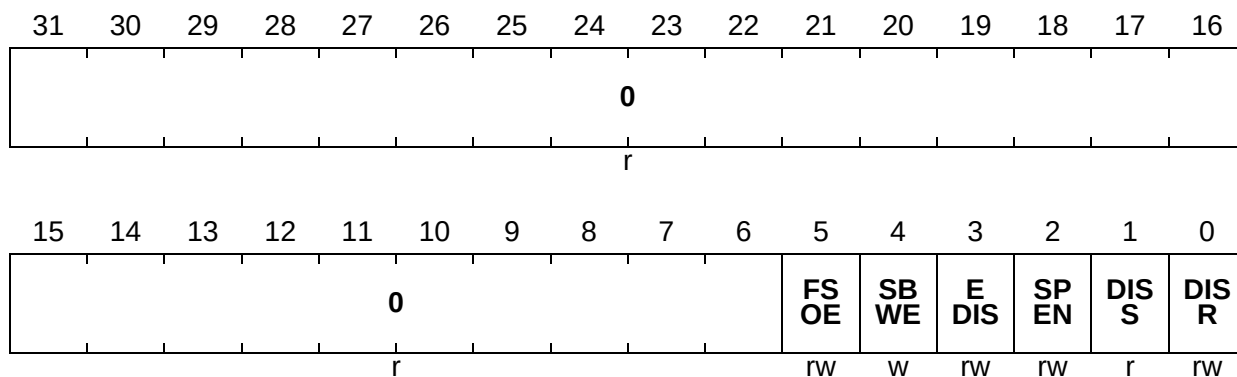
SSC0 Clock Control Register

Reset Value: 0000 0003_H

SSC1_CLC

SSC1 Clock Control Register

Reset Value: 0000 0003_H



Field	Bits	Type	Description
DISR	0	rw	Module Disable Request Bit Used for enable/disable control of the module.
DISS	1	r	Module Disable Status Bit Bit indicates the current status of the module.
SPEN	2	rw	Module Suspend Enable for OCDS Used for enabling the suspend mode.
EDIS	3	rw	External Request Disable Used for controlling the external clock disable request.
SBWE	4	w	Module Suspend Bit Write Enable for OCDS Defines whether SPEN and FSOE are write protected.
FSOE	5	rw	Fast Switch Off Enable Used for fast clock switch off in OCDS suspend mode.
0	[31:6]	r	Reserved ; read as 0; should be written with 0.

Synchronous Serial Interface (SSC)

The fractional divider register controls the clock rate of the shift clock f_{SSC0} and f_{SSC1} . Each SSC has its own fractional divider register.

SSC0_FDR

SSC0 Fractional Divider Register

Reset Value: 0000 0000_H

SSC1_FDR

SSC1 Fractional Divider Register

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DIS CLK	EN HW	SUS REQ	SUS ACK	0		RESULT									
rwh	rw	rh	rh	r		rh									
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DM		SC		SM	0	STEP									
rw		rw		rw	r	rw									

Field	Bits	Type	Description
STEP	[9:0]	rw	Step Value Reload or addition value for RESULT.
SM	11	rw	Suspend Mode 0 Granted suspend mode. 1 Immediate suspend mode.
SC	[13:12]	rw	Suspend Control This bit field defines the behavior of the fractional divider in suspend mode.
DM	[15:14]	rw	Divider Mode This bit field selects normal divider mode, fractional divider mode, and off-state.
RESULT	[25:16]	rh	Result Value Bit field for the addition result.
SUSACK	28	rh	Suspend Mode Acknowledge Indicates state of SPNDACK signal.
SUSREQ	29	rh	Suspend Mode Request Indicates state of SPND signal.
ENHW	30	rw	Enable Hardware Clock Control Controls operation of ECEN input and DISCLK bit.
DISCLK	31	rwh	Disable Clock Hardware controlled disable for f_{OUT} signal.

Field	Bits	Type	Description
0	10, [27:26]	rw	Reserved ; read as 0; should be written with 0

3.3.3.1 Port Input Select Register

The SSC1 module provides a Peripheral Input Select Register that is used to switch the MRST, MTSR, SCLK input lines of the SSC1 module kernel to either Port 2 or Port 3 as shown in [Figure 3-19](#).

Note: As shown in [Figure 3-19](#), the MRST, MTSR and SCLK lines of the SSC1 module can also be output lines. Port line input/output switching is controlled by the input/output control registers DIR.

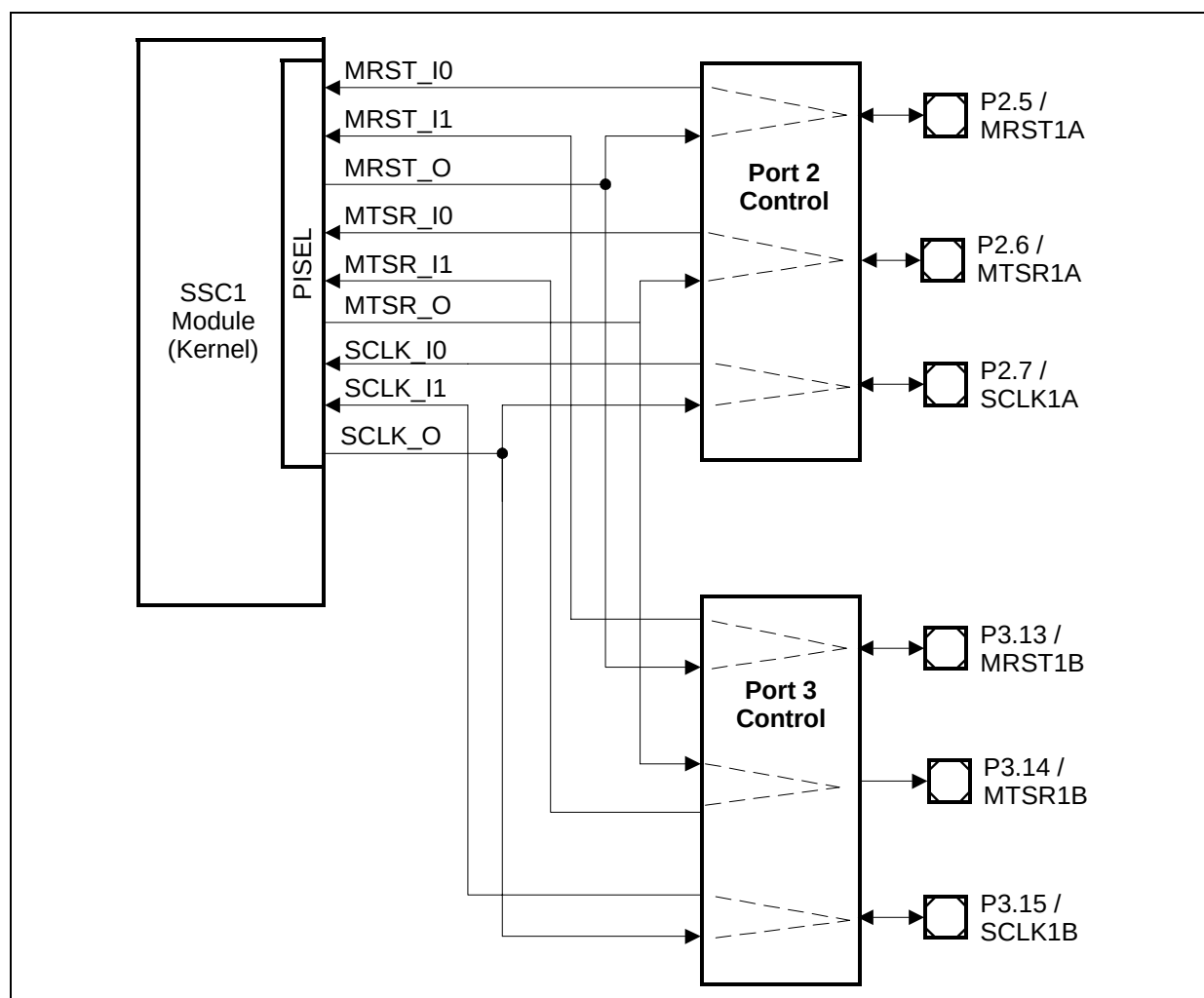


Figure 3-19 Input Line Selection of the SSC1 Module

The functionality of the SSC0/SSC1 port input select registers as shown on [Page 3-29](#) is reduced according the diagram described below:

Synchronous Serial Interface (SSC)

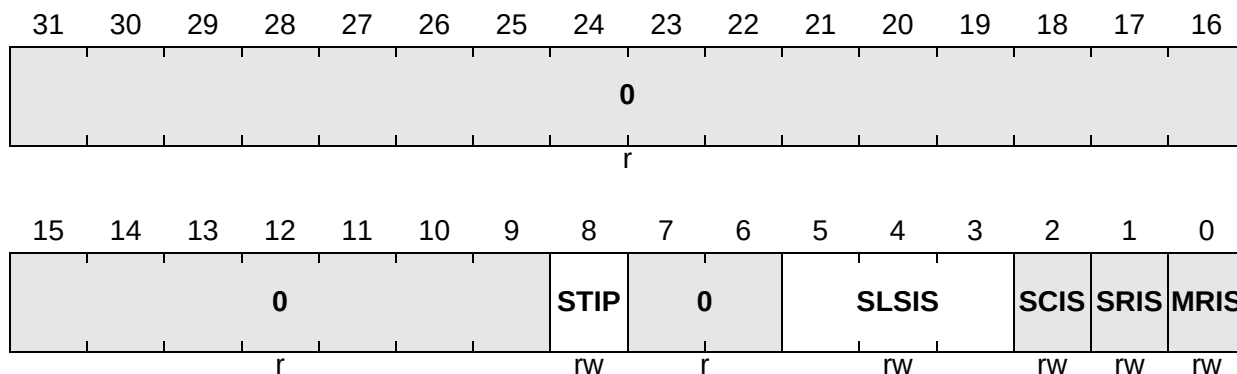
- No alternate input lines for SSC0 available
- Only one slave select input for each SSC0/SSC1 module used

Therefore, in the port input select registers the corresponding bit fields must be set as required in the application.

SSC0_PISEL

Port Input Select Register

Reset Value: 0000 0000_H



Field	Bits	Type	Description
SLSIS	[5:3]	rw	Slave Mode Slave Select Input Selection 000 _B Slave select input lines are deselected; SSC is operating without slave select input functionality. 001 _B SLSI1 input line is selected for operation. others Reserved; don't use these combinations.
STIP	8	rw	Slave Transmit Idle State Polarity This bit defines the logic level of the slave mode transmit signal MRST when the SSC0 is deselected (PISEL.SLSIS = 0). 0 MRST = 0 when SSC0 is deselected in slave mode. 1 MRST = 1 when SSC0 is deselected in slave mode.
0	[7:6], [31:9]	r	Reserved ; read as 0; should be written with 0.

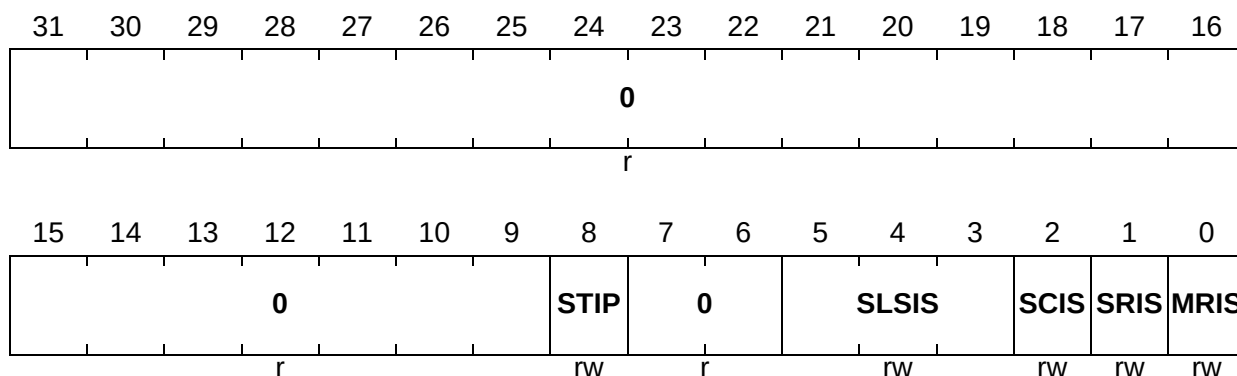
Note: Shaded bits and bit fields are don't care for SSC0 input control.

Synchronous Serial Interface (SSC)

SSC1_PISEL

Port Input Select Register

Reset Value: 0000 0000_H



Field	Bits	Type	Description
MRIS	0	rw	Master Mode Receive Input Select MRIS selects the receive input line in master mode. 0 Receive input line MRSTA is selected 1 Receive input line MRSTB is selected
SRIS	1	rw	Slave Mode Receive Input Select SRIS selects receive input line that in slave mode. 0 Receive input line MTSRA is selected 1 Receive input line MTSRB is selected
SCIS	2	rw	Slave Mode Clock Input Select SCIS selects the module kernel SCLK input line that is used as clock input line in slave mode. 0 Slave mode clock input line SCLKA is selected 1 Slave mode clock input line SCLKB is selected
SLSIS	[5:3]	rw	Slave Mode Slave Select Input Selection 000 _B Slave select input lines are deselected; SSC is operating without slave select input functionality. 001 _B <u>SLSI1</u> input line is selected for operation others Reserved; don't use these combinations.

Synchronous Serial Interface (SSC)

Field	Bits	Type	Description
STIP	8	rw	Slave Transmit Idle State Polarity This bit defines the logic level of the slave mode transmit signal MRST when the SSC1 is deselected (PISEL.SLSIS = 0). 0 MRST = 0 when SSC1 is deselected in slave mode. 1 MRST = 1 when SSC1 is deselected in slave mode.
0	[7:6], [31:9]	r	Reserved ; read as 0; should be written with 0.

3.3.3.2 Port Control

The interconnections between the SSC modules and the port I/O lines are controlled in the port logics. The following port control operation selections must be executed (additionally to the PISEL programming):

- Input/output direction selection (DIR registers)
- Alternate function selection (ALTSEL0 and ALTSEL1 registers)
- Input/Output driver characteristic control (PUDSEL, PUDEN and OD registers)

The port input/output control registers contain the bit fields that select the digital output and input driver characteristics such as pull-up/down devices, port direction (input/output), open-drain, and alternate output selections. The I/O lines for the SSC modules are controlled by the port input/output control registers of Port 0, Port 1, Port 2 and Port 3.

Table 3-3 shows how bits and bit fields must be programmed for the required I/O functionality of the SSC I/O lines.

Synchronous Serial Interface (SSC)

Table 3-3 SSC0 and SSC1 I/O Line Selection and Setup

Module	Port Lines	Input/Output Control Register Bits	I/O
SSC0	P2.2/MRST0	P2_DIR.P2 = 0 _B	Input
		P2_DIR.P2 = 1 _B	Output
		P2_ALTSEL0.P2 = 1 _B	
		P2_ALTSEL1.P2 = 0 _B	
	P2.3/MTSR0	P2_DIR.P3 = 0 _B	Input
		P2_DIR.P3 = 1 _B	Output
		P2_ALTSEL0.P3 = 1 _B	
		P2_ALTSEL1.P3 = 0 _B	
	P2.4/SCLK0	P2_DIR.P4 = 0 _B	Input
		P2_DIR.P4 = 1 _B	Output
		P2_ALTSEL0.P4 = 1 _B	
		P2_ALTSEL1.P4 = 0 _B	
	P1.15/SLSI0	P1_DIR.P15 = 0 _B	Input

Synchronous Serial Interface (SSC)

Table 3-3 SSC0 and SSC1 I/O Line Selection and Setup (cont'd)

Module	Port Lines	Input/Output Control Register Bits	I/O
SSC1	P2.5/MRST1A	P2_DIR.P5 = 0 _B	Input
		P2_DIR.P5 = 1 _B	Output
		P2_ALTSEL0.P5 = 1 _B	
		P2_ALTSEL1.P5 = 0 _B	
	P3.13/MRST1B	P3_DIR.P13 = 0 _B	Input
		P3_DIR.P13 = 1 _B	Output
		P3_ALTSEL0.P13 = 1 _B	
		P3_ALTSEL1.P13 = 0 _B	
	P2.6/MTSR1A	P2_DIR.P6 = 0 _B	Input
		P2_DIR.P6 = 1 _B	Output
		P2_ALTSEL0.P6 = 1 _B	
		P2_ALTSEL1.P6 = 0 _B	
	P3.14/MTSR1B	P3_DIR.P14 = 0 _B	Input
		P3_DIR.P14 = 1 _B	Output
		P3_ALTSEL0.P14 = 1 _B	
		P3_ALTSEL1.P14 = 0 _B	
	P2.7/SCLK1A	P2_DIR.P7 = 0 _B	Input
		P2_DIR.P7 = 1 _B	Output
		P2_ALTSEL0.P7 = 1 _B	
		P2_ALTSEL1.P7 = 0 _B	
	P3.15/SCLK1B	P3_DIR.P15 = 0 _B	Input
		P3_DIR.P15 = 1 _B	Output
		P3_ALTSEL0.P15 = 1 _B	
		P3_ALTSEL1.P15 = 0 _B	
	P0.4/SLSI1	P0_DIR.P4 = 0 _B	Input
Slave Select Outputs			

Synchronous Serial Interface (SSC)

Table 3-3 SSC0 and SSC1 I/O Line Selection and Setup (cont'd)

Module	Port Lines	Input/Output Control Register Bits	I/O
SSC0	P0.6/SLSO00	P0_DIR.P6 = 1 _B	Output
		P0_ALTSEL0.P6 = 1 _B	
		P0_ALTSEL1.P6 = 1 _B	
	P1.11/SLSO01	P1_DIR.P11 = 1 _B	Output
		P1_ALTSEL0.P11 = 0 _B	
		P1_ALTSEL1.P11 = 1 _B	
	P1.13/SLSO02	P1_DIR.P13 = 1 _B	Output
		P1_ALTSEL0.P13 = 0 _B	
		P1_ALTSEL1.P13 = 1 _B	
	P2.12/SLSO03	P2_DIR.P12 = 1 _B	Output
		P2_ALTSEL0.P12 = 0 _B	
		P2_ALTSEL1.P12 = 1 _B	
	P2.14/SLSO04	P2_DIR.P14 = 1 _B	Output
		P2_ALTSEL0.P14 = 0 _B	
		P2_ALTSEL1.P14 = 1 _B	
	P3.7/SLSO05	P3_DIR.P7 = 1 _B	Output
		P3_ALTSEL0.P7 = 1 _B	
		P3_ALTSEL1.P7 = 0 _B	
	P3.9/SLSO06	P3_DIR.P9 = 1 _B	Output
		P3_ALTSEL0.P9 = 1 _B	
		P3_ALTSEL1.P9 = 0 _B	
	P3.11/SLSO07	P3_DIR.P11 = 1 _B	Output
		P3_ALTSEL0.P11 = 1 _B	
		P3_ALTSEL1.P11 = 0 _B	

Synchronous Serial Interface (SSC)

Table 3-3 SSC0 and SSC1 I/O Line Selection and Setup (cont'd)

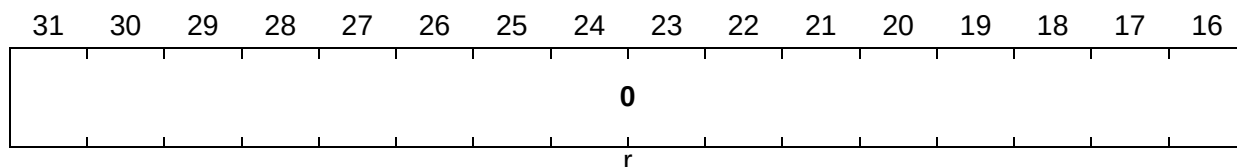
Module	Port Lines	Input/Output Control Register Bits	I/O
SSC1	P0.7/SLSO10	P0_DIR.P7 = 1 _B	Output
		P0_ALTSEL0.P7 = 1 _B	
		P0_ALTSEL1.P7 = 1 _B	
	P1.12/SLSO11	P1_DIR.P12 = 1 _B	Output
		P1_ALTSEL0.P12 = 0 _B	
		P1_ALTSEL1.P12 = 1 _B	
	P1.14/SLSO12	P1_DIR.P14 = 1 _B	Output
		P1_ALTSEL0.P14 = 0 _B	
		P1_ALTSEL1.P14 = 1 _B	
	P2.13/SLSO13	P2_DIR.P13 = 1 _B	Output
		P2_ALTSEL0.P13 = 0 _B	
		P2_ALTSEL1.P13 = 1 _B	
	P2.15/SLSO14	P2_DIR.P15 = 1 _B	Output
		P2_ALTSEL0.P15 = 0 _B	
		P2_ALTSEL1.P15 = 1 _B	
	P3.8/SLSO15	P3_DIR.P8 = 1 _B	Output
		P3_ALTSEL0.P8 = 1 _B	
		P3_ALTSEL1.P8 = 0 _B	
	P3.10/SLSO16	P3_DIR.P10 = 1 _B	Output
		P3_ALTSEL0.P10 = 1 _B	
		P3_ALTSEL1.P10 = 0 _B	
	P3.12/SLSO17	P3_DIR.P12 = 1 _B	Output
		P3_ALTSEL0.P12 = 1 _B	
		P3_ALTSEL1.P12 = 0 _B	

Synchronous Serial Interface (SSC)

P0_DIR

Port 0 Direction Register

Reset Value: 0000 0000_H



15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
P15	P14	P13	P12	P11	P10	P9	P8	P7	P6	P5	P4	P3	P2	P1	P0
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

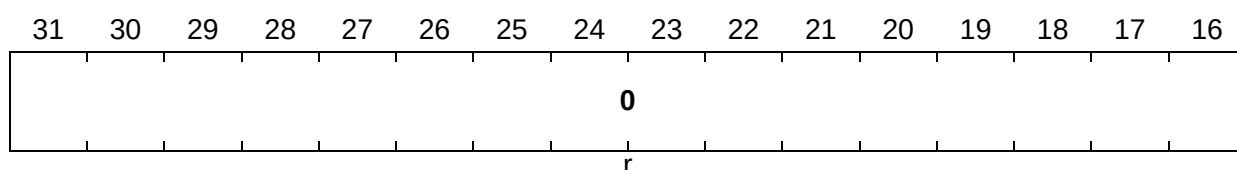
Field	Bits	Type	Description
Pn (n = 4, 6, 7)	n	rW	Port 0 Pin 4, 6, 7 Direction Control ¹⁾ 0 Direction is set to input (default after reset) 1 Direction is set to output
0	[31:16]	r	Reserved ; read as 0; should be written with 0.

¹⁾ Shaded bits and bit field are not taken into account for SSC I/O port control

P1_DIR

Port 1 Direction Register

Reset Value: 0000 0000_H



15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
P15	P14	P13	P12	P11	P10	P9	P8	P7	P6	P5	P4	P3	P2	P1	P0
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

Field	Bits	Type	Description
Pn (n = 11-15)	n	rW	Port 1 Pin 11 - 15 Direction Control ¹⁾ 0 Direction is set to input (default after reset) 1 Direction is set to output
0	[31:16]	r	Reserved ; read as 0; should be written with 0.

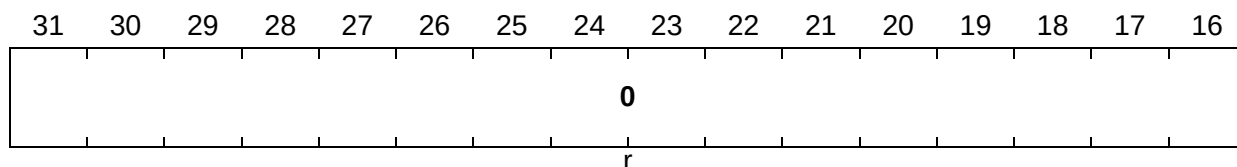
¹⁾ Shaded bits and bit field are don't care for SSC I/O port control

Synchronous Serial Interface (SSC)

P2_DIR

Port 2 Direction Register

Reset Value: 0000 0000_H



15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
P15	P14	P13	P12	P11	P10	P9	P8	P7	P6	P5	P4	P3	P2	P1	P0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

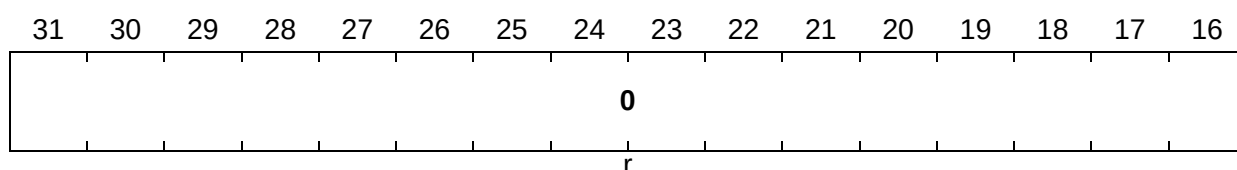
Field	Bits	Type	Description
Pn (n = 2-7, 12-15)	n	rw	Port 2 Pin 2-7, 12-15 Direction Control ¹⁾ 0 Direction is set to input (default after reset) 1 Direction is set to output
0	[31:16]	r	Reserved ; read as 0; should be written with 0.

¹⁾ Shaded bits and bit field are don't care for SSC I/O port control

P3_DIR

Port 3 Direction Register

Reset Value: 0000 0000_H



15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
P15	P14	P13	P12	P11	P10	P9	P8	P7	P6	P5	P4	P3	P2	P1	P0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Field	Bits	Type	Description
Pn (n = 7-15)	n	rw	Port 3 Pin 7-15 Direction Control ¹⁾ 0 Direction is set to input (default after reset) 1 Direction is set to output
0	[31:16]	r	Reserved ; read as 0; should be written with 0.

¹⁾ Shaded bits and bit field are don't care for SSC I/O port control

Synchronous Serial Interface (SSC)

P0_ALTSELn (n = 1, 0)

Port 0 Alternate Select Register

Reset Value: 0000 0000_H

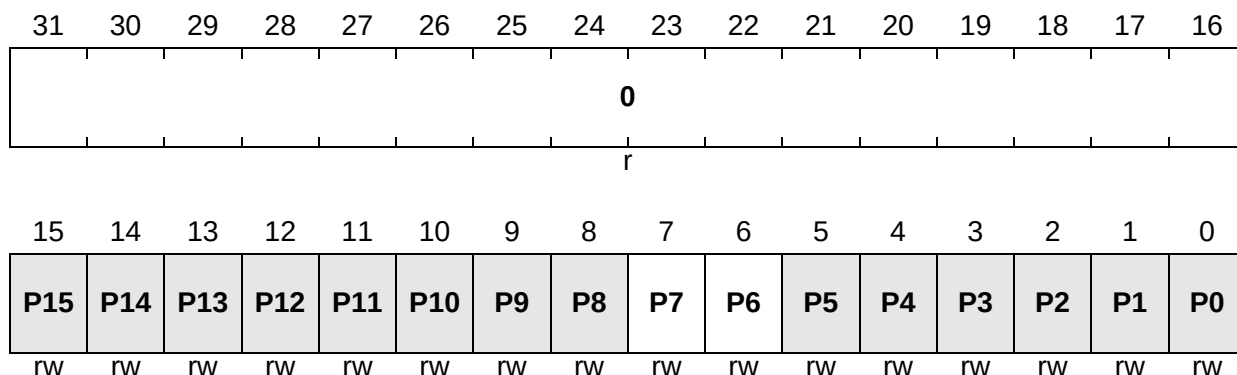


Table 3-4 Function of the Bits P0_ALTSEL0.Pn and P0_ALTSEL1.Pn (n = 6,7)¹⁾

P0_ALTSEL0.Pn	P0_ALTSEL1.Pn	Function
1	1	Alternate Select 3

¹⁾ Shaded bits and bit field are don't care for SSC I/O port control

P1_ALTSELn (n = 1, 0)

Port 1 Alternate Select Register

Reset Value: 0000 0000_H

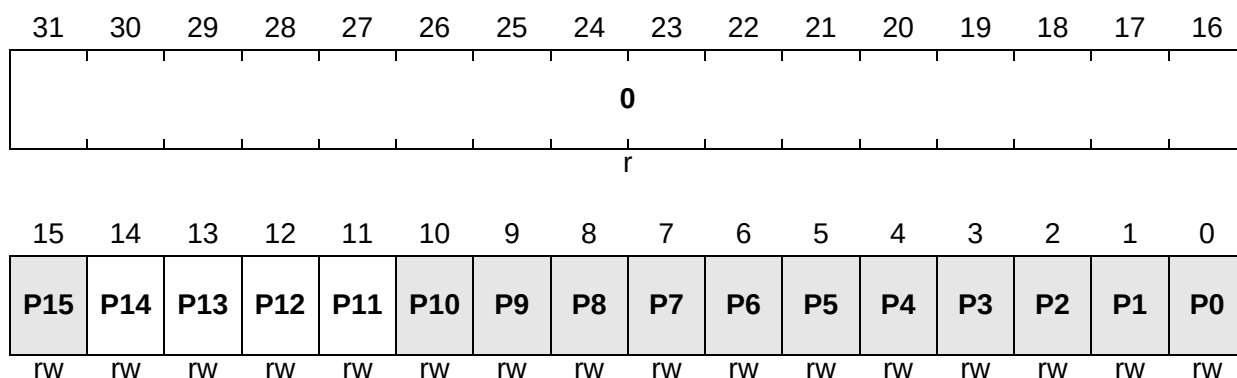


Table 3-5 Function of the Bits P1_ALTSEL0.Pn and P1_ALTSEL1.Pn (n = 11-14)¹⁾

P1_ALTSEL0.Pn	P1_ALTSEL1.Pn	Function
0	1	Alternate Select 2

¹⁾ Shaded bits and bit field are don't care for SSC I/O port control

Synchronous Serial Interface (SSC)

P2_ALTSELn (n = 1, 0)

Port 2 Alternate Select Register

Reset Value: 0000 0000_H

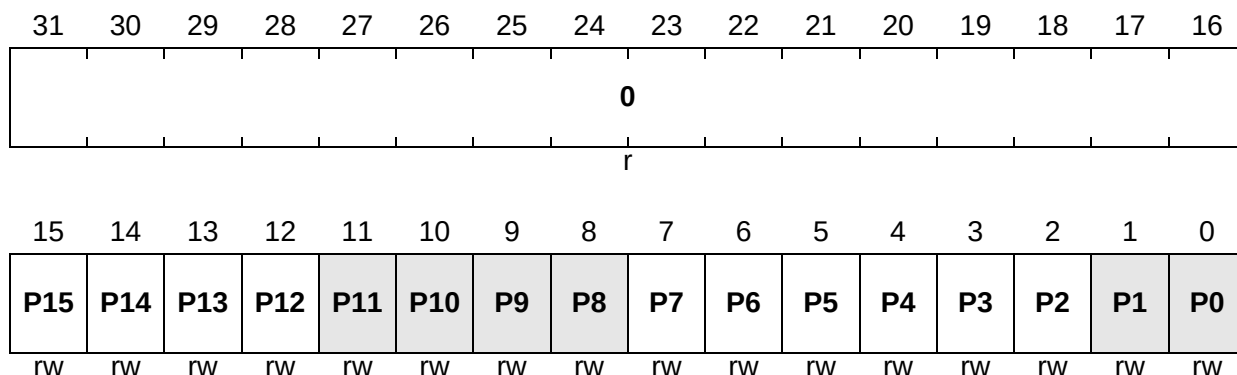


Table 3-6 Function of the Bits P2_ALTSEL0.Pn and P2_ALTSEL1.Pn (n = 2-7, 12-15)¹⁾

P2_ALTSEL0.Pn	P2_ALTSEL1.Pn	Function
1	0	Alternate Select 1
0	1	Alternate Select 2

¹⁾ Shaded bits and bit field are don't care for SSC I/O port control

P3_ALTSELn (n = 1, 0)

Port 3 Alternate Select Register

Reset Value: 0000 0000_H

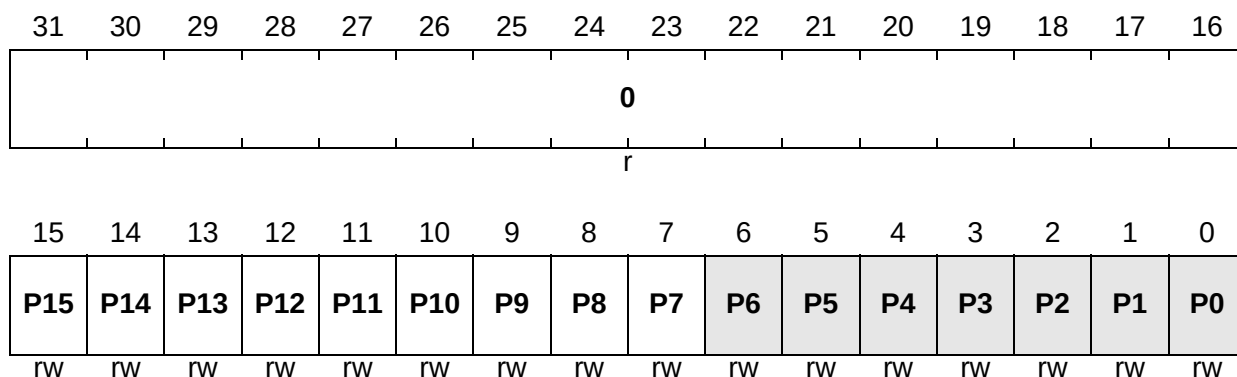


Table 3-7 Function of the Bits P3_ALTSEL0.Pn and P3_ALTSEL1.Pn (n = 7-15)¹⁾

P3_ALTSEL0.Pn	P3_ALTSEL1.Pn	Function
1	0	Alternate Select 1

¹⁾ Shaded bits and bit field are don't care for SSC I/O port control

Synchronous Serial Interface (SSC)

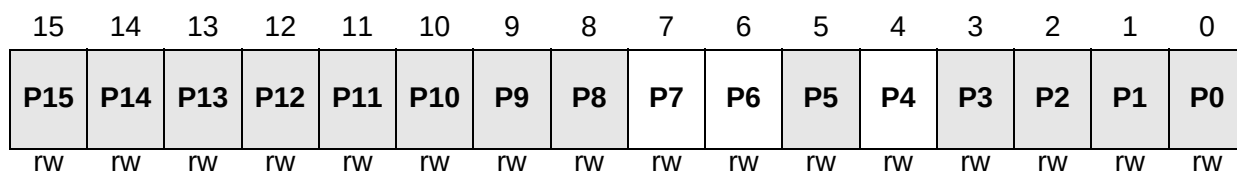
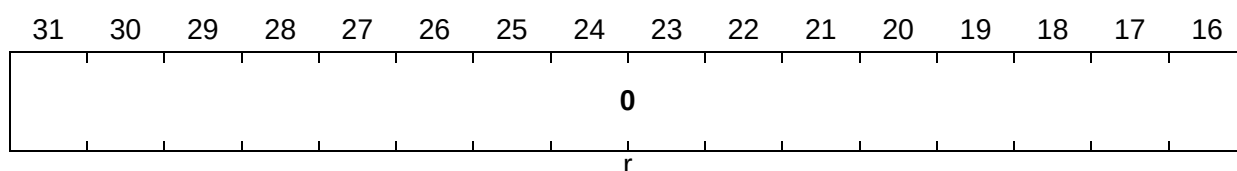
The SSC0/SSC1 ports also offer the possibility to configure the following output characteristics:

- push/pull (optional pull-up/pull-down)
- open drain with internal pull-up
- open drain with external pull-up

P0_PUDSEL

Port 0 Pull-Up/Pull-Down Select Register

Reset Value: 0000 FFFF_H



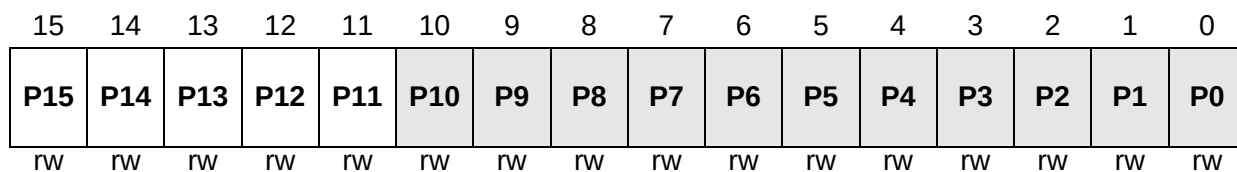
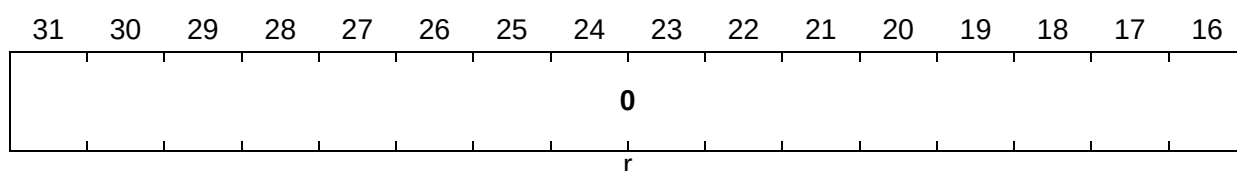
Field	Bits	Type	Description
Pn (n = 4,6,7)	n	rw	Pull-Up/Pull-Down Select Port 0 Bit n¹⁾ 0 Pull-down device is selected 1 Pull-up device is selected
0	[31:16]	r	Reserved ; read as 0; should be written with 0.

¹⁾ Shaded bits and bit field are don't care for SSC I/O port control

P1_PUDSEL

Port 1 Pull-Up/Pull-Down Select Register

Reset Value: 0000 FFFF_H



Synchronous Serial Interface (SSC)

Field	Bits	Type	Description
Pn (n = 11-15)	n	rw	Pull-Up/Pull-Down Select Port 1 Bit n¹⁾ 0 Pull-down device is selected 1 Pull-up device is selected
0	[31:16]	r	Reserved ; read as 0; should be written with 0.

¹⁾ Shaded bits and bit field are don't care for SSC I/O port control

P2_PUDSEL

Port 2 Pull-Up/Pull-Down Select Register

Reset Value: 0000 FFFF_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0				P11	P10	P9	P8	P7	P6	P5	P4	P3	P2	P1	P0
r				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Field	Bits	Type	Description
Pn (n = 2-7)	n	rw	Pull-Up/Pull-Down Select Port 2 Bit n¹⁾ 0 Pull-down device is selected 1 Pull-up device is selected
0	[31:12]	r	Reserved ; read as 0; should be written with 0.

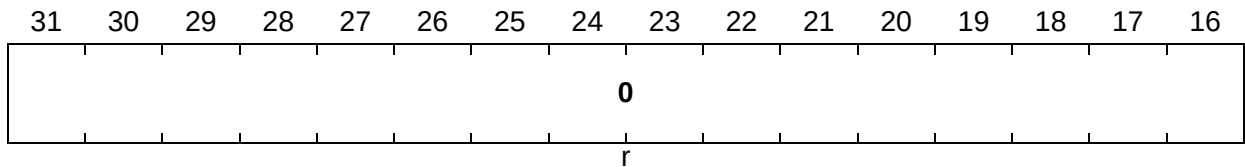
¹⁾ Shaded bits and bit field are don't care for SSC I/O port control

Synchronous Serial Interface (SSC)

P3_PUDSEL

Port 3 Pull-Up/Pull-Down Select Register

Reset Value: 0000 FFFF_H



15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
P15	P14	P13	P12	P11	P10	P9	P8	P7	P6	P5	P4	P3	P2	P1	P0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

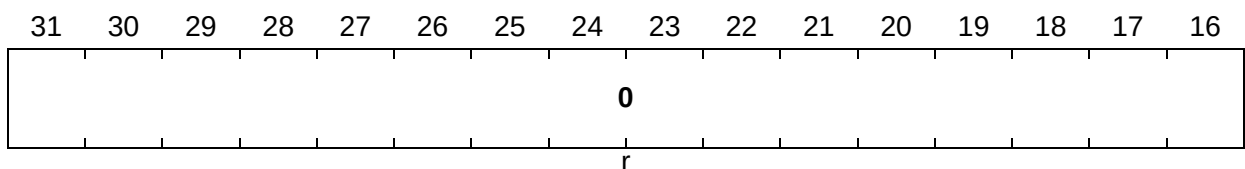
Field	Bits	Type	Description
Pn (n = 7-15)	n	rw	Pull-Up/Pull-Down Select Port 3 Bit n¹⁾ 0 Pull-down device is selected 1 Pull-up device is selected
0	[31:16]	r	Reserved ; read as 0; should be written with 0.

¹⁾ Shaded bits and bit field are don't care for SSC I/O port control

P0_PUDEN

Port 0 Pull-Up/Pull-Down Enable Register

Reset Value: 0000 FFFF_H



15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
P15	P14	P13	P12	P11	P10	P9	P8	P7	P6	P5	P4	P3	P2	P1	P0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Field	Bits	Type	Description
Pn (n = 4,6,7)	n	rw	Pull-Up/Pull-Down Enable at Port 0 Bit n¹⁾ 0 Pull-up or Pull-down device is disabled 1 Pull-up or Pull-down device is enabled
0	[31:16]	r	Reserved ; read as 0; should be written with 0.

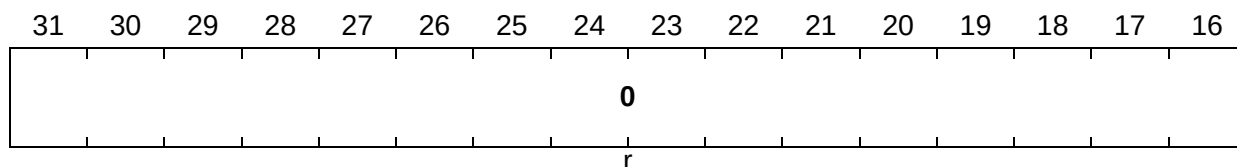
¹⁾ Shaded bits and bit field are don't care for SSC I/O port control

Synchronous Serial Interface (SSC)

P1_PUDEN

Port 1 Pull-Up/Pull-Down Enable Register

Reset Value: 0000 FFFF_H



15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
P15	P14	P13	P12	P11	P10	P9	P8	P7	P6	P5	P4	P3	P2	P1	P0
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

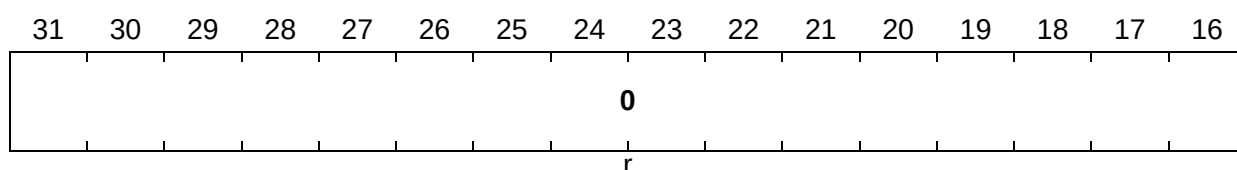
Field	Bits	Type	Description
Pn (n = 11-15)	n	rw	Pull-Up/Pull-Down Enable at Port 1 Bit n¹⁾ 0 Pull-up or Pull-down device is disabled 1 Pull-up or Pull-down device is enabled
0	[31:16]	r	Reserved ; read as 0; should be written with 0.

¹⁾ Shaded bits and bit field are don't care for SSC I/O port control

P2_PUDEN

Port 2 Pull-Up/Pull-Down Enable Register

Reset Value: 0000 0FFF_H



15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0				P11	P10	P9	P8	P7	P6	P5	P4	P3	P2	P1	P0
r				r	r	r	r	r	r	r	r	r	r	r	r

Field	Bits	Type	Description
Pn (n = 2-7)	n	rw	Pull-Up/Pull-Down Enable at Port 2 Bit n¹⁾ 0 Pull-up or Pull-down device is disabled 1 Pull-up or Pull-down device is enabled
0	[31:12]	r	Reserved ; read as 0; should be written with 0.

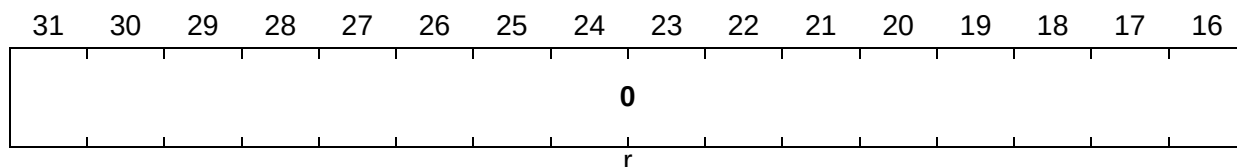
¹⁾ Shaded bits and bit field are don't care for SSC I/O port control

Synchronous Serial Interface (SSC)

P3_PUDEN

Port 3 Pull-Up/Pull-Down Enable Register

Reset Value: 0000 FFFF_H



15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
P15	P14	P13	P12	P11	P10	P9	P8	P7	P6	P5	P4	P3	P2	P1	P0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

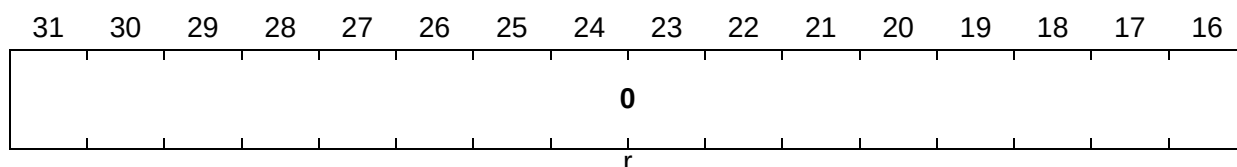
Field	Bits	Type	Description
Pn (n = 7-15)	n	rw	Pull-Up/Pull-Down Enable at Port 3 Bit n¹⁾ 0 Pull-up or Pull-down device is disabled 1 Pull-up or Pull-down device is enabled
0	[31:16]	r	Reserved ; read as 0; should be written with 0.

¹⁾ Shaded bits and bit field are don't care for SSC I/O port control

P0_OD

Port 0 Open Drain Control Register

Reset Value: 0000 0000_H



15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
P15	P14	P13	P12	P11	P10	P9	P8	P7	P6	P5	P4	P3	P2	P1	P0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Field	Bits	Type	Description
Pn (n = 6,7)	n	rw	Port 0 Pin n Open Drain Mode¹⁾ 0 Normal Mode, output is actively driven for 0 and 1 state 1 Open Drain Mode, output is actively driven only for 0 state

Synchronous Serial Interface (SSC)

Field	Bits	Type	Description
0	[31:16]	r	Reserved ; read as 0; should be written with 0.

¹⁾ Shaded bits and bit field are don't care for SSC I/O port control

P1_OD

Port 1 Open Drain Control Register

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
P15	P14	P13	P12	P11	P10	P9	P8	P7	P6	P5	P4	P3	P2	P1	P0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Field	Bits	Type	Description
Pn (n = 11-14)	n	rw	Port 1 Pin n Open Drain Mode¹⁾ 0 Normal Mode, output is actively driven for 0 and 1 state 1 Open Drain Mode, output is actively driven only for 0 state
0	[31:16]	r	Reserved ; read as 0; should be written with 0.

¹⁾ Shaded bits and bit field are don't care for SSC I/O port control

P2_OD

Port 2 Open Drain Control Register

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
P15	P14	P13	P12	P11	P10	P9	P8	P7	P6	P5	P4	P3	P2	P1	P0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Synchronous Serial Interface (SSC)

Field	Bits	Type	Description
Pn (n = 2-7, 12-15)	n	rw	Port 2 Pin n Open Drain Mode¹⁾ 0 Normal Mode, output is actively driven for 0 and 1 state 1 Open Drain Mode, output is actively driven only for 0 state
0	[31:16]	r	Reserved ; read as 0; should be written with 0.

¹⁾ Shaded bits and bit field are don't care for SSC I/O port control

P3_OD

Port 3 Open Drain Control Register

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
P15	P14	P13	P12	P11	P10	P9	P8	P7	P6	P5	P4	P3	P2	P1	P0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Field	Bits	Type	Description
Pn (n = 7-15)	n	rw	Port 3 Pin n Open Drain Mode¹⁾ 0 Normal Mode, output is actively driven for 0 and 1 state 1 Open Drain Mode, output is actively driven only for 0 state
0	[31:16]	r	Reserved ; read as 0; should be written with 0.

¹⁾ Shaded bits and bit field are don't care for SSC I/O port control

3.3.3.3 Interrupt Registers

The 2×3 interrupts of the SSC0 and SSC1 module are controlled by the following service request control registers:

- SSC0_TSRC, SSC1_TSRC controls the transmit interrupts
- SSC0_RSRC, SSC1_RSRC controls the receive interrupts
- SSC0_ESRC, SSC1_ESRC controls the error interrupts

SSC0_TSRC

SSC0 Transmit Interrupt Service Request Control Register

SSC0_RSRC

SSC0 Receive Interrupt Service Request Control Register

SSC0_ESRC

SSC0 Error Interrupt Service Request Control Register

SSC1_TSRC

SSC1 Transmit Interrupt Service Request Control Register

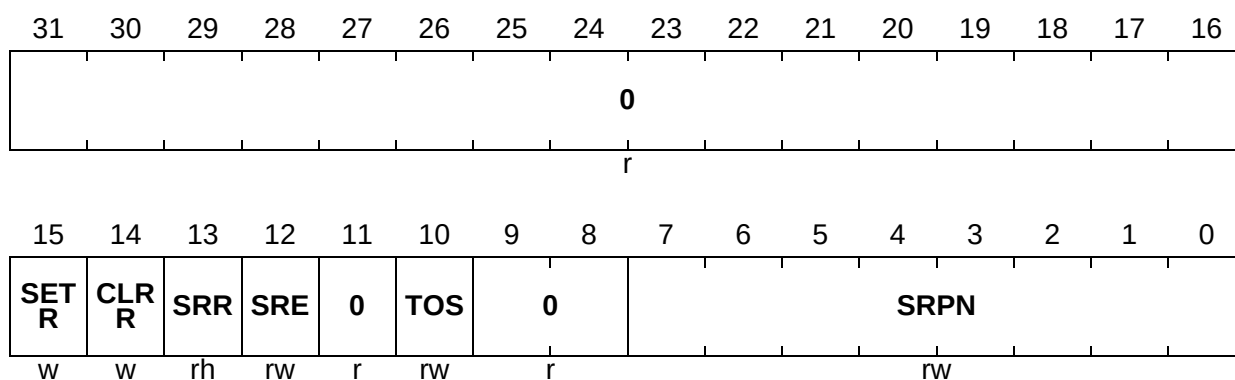
SSC1_RSRC

SSC1 Receive Interrupt Service Request Control Register

SSC1_ESRC

SSC1 Error Interrupt Service Request Control Register

Reset Values: 0000 0000_H



Field	Bits	Type	Description
SRPN	[7:0]	rw	Service Request Priority Number
TOS	10	rw	Type of Service Control
SRE	12	rw	Service Request Enable
SRR	13	rh	Service Request Flag
CLRR	14	w	Request Clear Bit
SETR	15	w	Request Set Bit

Synchronous Serial Interface (SSC)

Field	Bits	Type	Description
0	[9:8], 11, [31:16]	r	Reserved ; read as 0; should be written with 0.

Note: Further details on interrupt handling and processing are described in chapter "Interrupt System" of the TC1100 System Units User's Manual.

3.3.4 DMA Requests

The DMA request lines of the SSC0/SSC1 modules become active whenever its related interrupt line is activated. The DMA request lines are connected to the DMA controller as shown in [Table 3-8](#).

Table 3-8 DMA Request Lines of SSC0/SSC1

Module	Related SSC Interrupt	DMA Request Line	Description
SSC0	RIR	SSC0_RDR	SSC0 Receive DMA Request
	TIR	SSC0_TDR	SSC0 Transmit DMA Request
SSC1	RIR	SSC1_RDR	SSC1 Receive DMA Request
	TIR	SSC1_TDR	SSC1 Transmit DMA Request

3.3.5 SSC0/SSC1 Register Address Ranges

In the TC1100, the registers of the two SSC modules are located in the following address ranges:

- SSC0 module: Module Base Address = F010 0100_H
 Module End Address = F010 01FF_H
- SSC1 module: Module Base Address = F010 0200_H
 Module End Address = F010 02FF_H
- Absolute Register Address = Module Base Address + Offset Address
(offset addresses see [Table 3-2](#))

Note: The complete and detailed address map of the SSC0/SSC1 modules is described in the chapter "Register Overview" of the TC1100 System Units User's Manual.

4 IIC

This chapter describes the IIC Module of the TC1100. It contains the following sections:

- Functional description of the IIC Kernel (see [Section 4.1](#))
- Register descriptions of all IIC Kernel specific registers (see [Section 4.2](#))
- TC1100 implementation specific details and registers of the IIC (port connections and control, interrupt control, address decoding, clock control, see [Section 4.3](#))

Note: The IIC kernel register names described in [Section 4.2](#) will be referenced in other parts of the TC1100 User's Manual with the module name prefix "IIC_".

4.1 IIC Kernel Description

IIC supports a certain protocol to allow devices to communicate directly with each other via two wires. One line is responsible for clock transfer and synchronization (SCL), the other is responsible for the data transfer (SDA).

4.1.1 Introduction

The on-chip IIC Bus module connects the platform buses to other external controllers and/or peripherals via the two-line serial IIC interface. The IIC Bus module provides communication at data rates of up to 400 Kbit/s and features 7-bit addressing as well as 10-bit addressing. This module is fully compatible to the IIC bus protocol.

The module can operate in three different modes:

Master mode, where the IIC controls the bus transactions and provides the clock signal.

Slave mode, where an external master controls the bus transactions and provides the clock signal.

Multimaster mode, where several masters can be connected to the bus, i.e. the IIC can be master or slave.

The on-chip IIC bus module allows efficient communication via the common IIC bus. The module unloads the CPU of low level tasks such as:

- (De)Serialization of bus data.
- Generation of start and stop conditions.
- Monitoring the bus lines in slave mode.
- Evaluation of the device address in slave mode.
- Bus access arbitration in multimaster mode.

Features

- Software compatible to V1.0 of C161RI.
- Extended buffer allows up to 4 send/receive data bytes to be stored.
- Selectable baud rate generation.
- Support of standard 100 KBAud and extended 400 KBAud data rates.
- Operation in 7-bit addressing mode or 10-bit addressing mode.
- Flexible control via interrupt service routines or by polling.
- Dynamic access to up to 2 physical IIC buses.

Applications

- EEPROMs
- 7-Segment Displays
- Keyboard Controllers
- On-Screen Display
- Audio Processors

4.1.2 Operational Overview

Data is transferred by the 2-line IIC bus (SDA, SCL) using a protocol that ensures reliable and efficient transfers. This protocol clearly distinguishes regular data transfers from defined control signals which control the data transfers.

The following bus conditions are defined:

Bus Idle: SDA and SCL remain high. The IIC bus is currently not used.

Data Valid: SDA stable during the high phase of SCL. SDA then represents the transferred bit. There is one clock pulse for each transferred bit of data. During data transfers SDA may only change while SCL is low (see below)!

Start Transfer: A falling edge on SDA ($\searrow$) while SCL is high indicates a start condition. This start condition initiates a data transfer over the IIC bus.

Stop Transfer: A rising edge on SDA ($\nearrow$) while SCL is high indicates a stop condition. This stop condition terminates a data transfer. Between a start condition and a stop condition an arbitrary number of bytes may be transferred.

The figure below gives examples for these bus conditions.

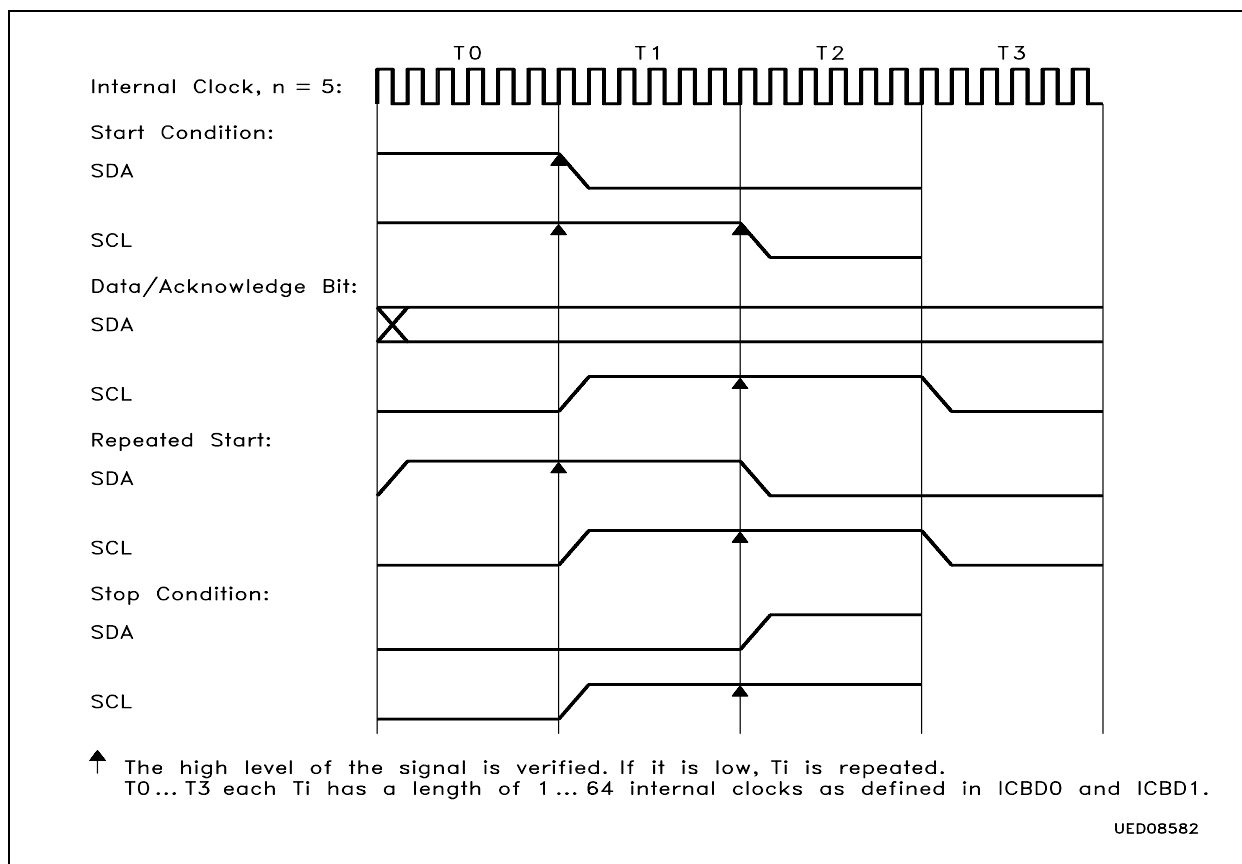


Figure 4-1 Bus Conditions

The Physical IIC-Bus Interface

Communication via the IIC Bus uses two bi-directional lines, the serial data line SDA and the serial clock line SCL. These two generic interface lines can each be connected to a number of IO port lines. These connections can be established and released under software control.

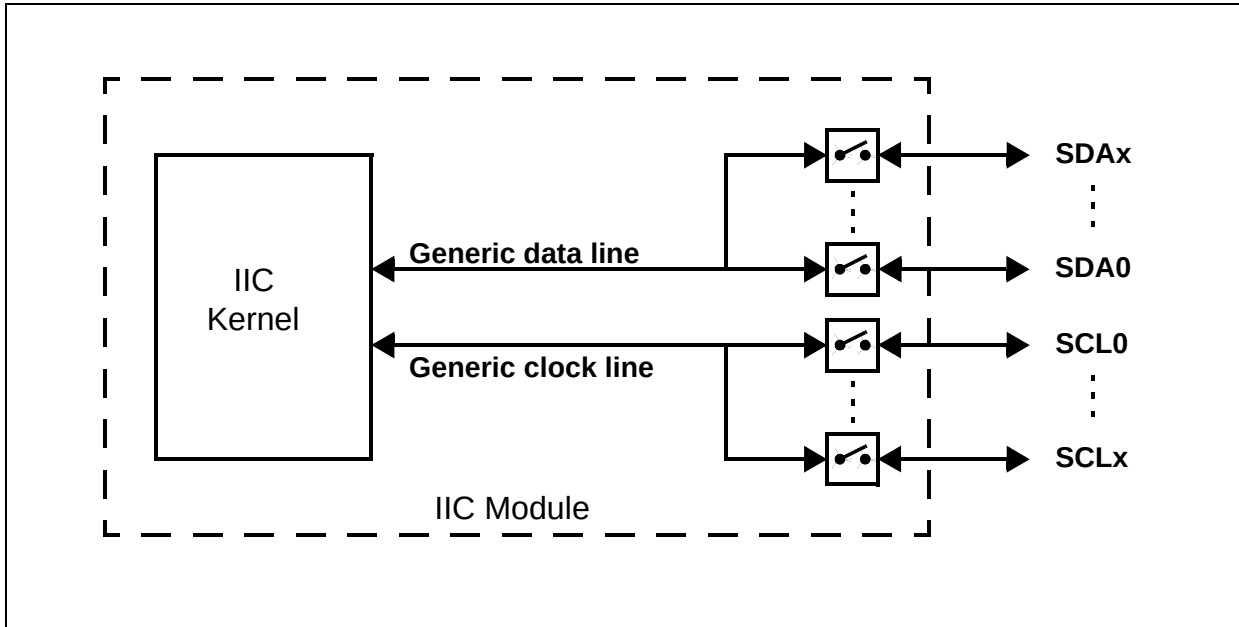


Figure 4-2 IIC Bus Line Connections

This mechanism allows a number of configurations of the physical IIC Bus interface:

Channel switching: The IIC module can be connected to a specific pair of pins (e.g. SDA0 and SCL0) which then forms a separate IIC channel to the external system. The channel can be dynamically switched by connecting the module to another pair of pins (e.g. SDA1 and SCL1). This establishes physically separate interface channels.

Broadcasting: Connecting the module to more than one pair of pins (e.g. SDA0/1 and SCL0/1) allows the transmission of messages over multiple physical channels at the same time. Please note that this configuration is critical when the IIC is a slave. In master mode it cannot be guaranteed that all selected slaves have reached the message.

Register BUSCON selects the bus baud rate as well the activation of SDA and SCL lines. So an external IIC channel can be established (baud rate and physical lines) with one single register access.

Physical channels can be selected, so the IIC module can use electrically separated channels or increase the addressing range by using more data lines.

Note: Baud rate and physical channels should never be changed (via BUSCON) during a transfer.

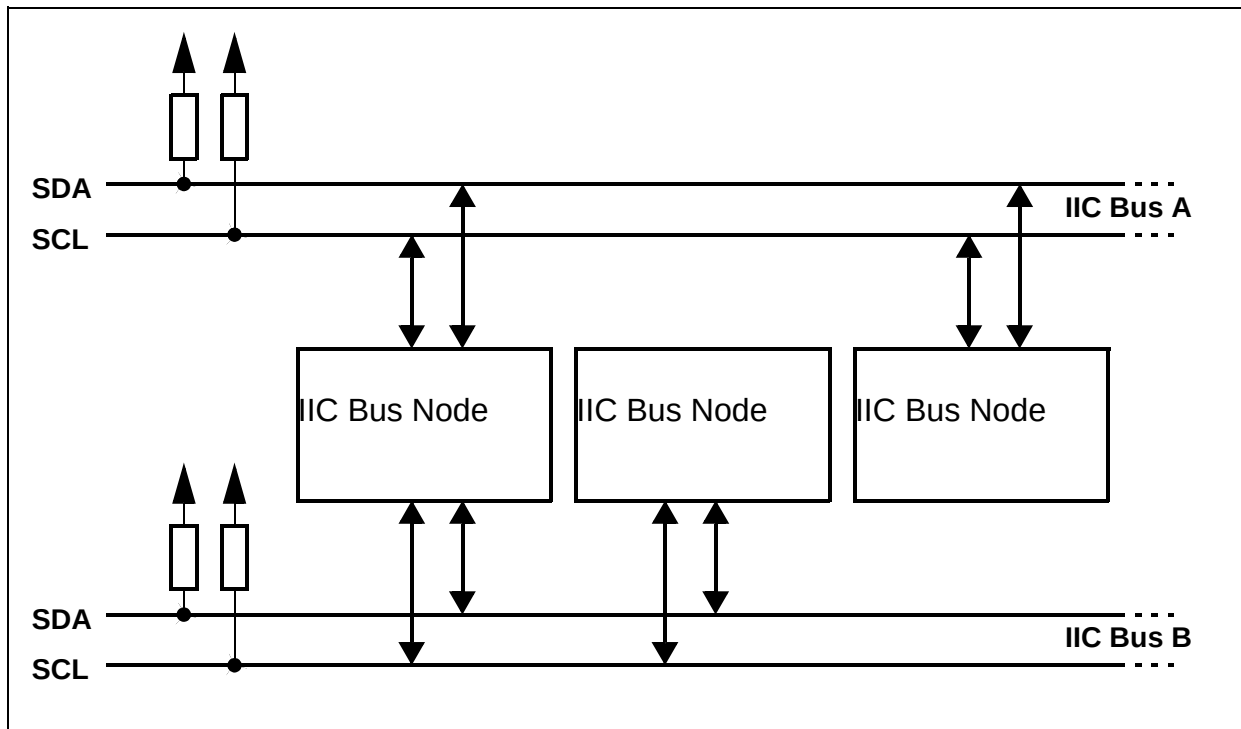


Figure 4-3 Physical Bus Configuration Example

Output Pin Configuration

The pin drivers that are assigned to the IIC channel(s) provide open drain outputs (i.e. no upper transistor). This ensures that the IIC module does not put any load on the IIC bus lines while the IIC is not powered. The IIC bus lines therefore require external pull-up resistors (approx. 10 K Ω for operation at 100 KBaud, 2 K Ω for operation at 400 KBaud).

All pins of the IIC that are to be used for IIC bus communication must be switched to output and their alternate function must be enabled (by setting the respective port output latch to 1), before any communication can be established.

If not driven by the IIC module (i.e. the corresponding enable bit in register BUSCON is 0) they then switch off their drivers (i.e. driving 1 to an open drain output). Due to the external pull-up devices the respective bus levels will then be 1 which is idle.

The IIC module features digital input filters in order to improve the rejection of noise from the external bus lines.

4.1.3 Functional Overview

4.1.3.1 Operation in Master Mode

If the on-chip IIC module shall control the IIC bus (i.e. be bus master) master mode must be selected via bit field MOD in register SYSCON. The physical channel is configured by a control word written to register BUSCON, defining the active interface pins and the used baud rate. More than one SDA and/or SCL line may be active at a time. The address of the remote slave that is to be accessed is written to RTB0..3. The bus is claimed by setting bit BUM in register SYSCON. This generates a start condition on the bus and automatically starts the transmission of the address in RTB0. Bit TRX in register SYSCON defines the transfer direction (TRX= 1, i.e. transmit, for the slave address). A repeated start condition is generated by setting bit RSC in register SYSCON, which automatically starts the transmission of the address previously written to RTB0. This may be used to change the transfer direction. RSC is cleared automatically after the repeated start condition has been generated.

The bus is released by clearing bit BUM in register SYSCON. This generates a stop condition on the bus.

In receive mode, if a data transfer is stopped by setting the STP bit in SYSCON register, no acknowledge is issued when the last byte is shifted in. But if ACKDIS is supposed to disable the acknowledge, the ACKDIS bit has to be set before TRX is clear to 0.

4.1.3.2 Operation in Multimaster Mode

If multimaster mode is selected via bit field MOD in register SYSCON the on-chip IIC module can operate concurrently as a bus master or as a slave. The descriptions of these modes apply accordingly.

Multimaster mode implies that several masters are connected to the same bus. As more than one master may try to claim the bus at a given time an arbitration is done on the SDA line. When a master device detects a mismatch between the data bit to be sent and the actual level on the SDA (bus) line it loses the arbitration and automatically switches to slave mode (leaving the other device as the remaining master). This loss of arbitration is indicated by bit AL in register SYSCON which must be checked by the driver software when operating in multimaster mode. Lost arbitration is also indicated when the software tries to claim the bus (by setting bit BUM) while the IIC bus is active (indicated by bit BB = 1). Bit AL must be cleared via software.

4.1.3.3 Operation in Slave Mode

If the on-chip IIC module shall be controlled via the IIC bus by a remote master (i.e. be a bus slave) slave mode must be selected via bit field MOD in register SYSCON. The physical channel is configured by a control word written to register SYSCON, defining the active interface pins and the used baud rate. It is recommended to have only one

SDA and SCL line active at a time when operating in slave mode. The address by which the slave module can be selected is written to register BUSCON.

The IIC module is selected by another master when it receives (after a start condition) either its own device address (stored in BUSCON) or the general call address (00_H). In this case an interrupt is generated and bit SLA in register SYSCON is set indicating the valid selection. The desired transfer mode is then selected via bit TRX (TRX= 0 for reception, TRX= 1 for transmission).

For a transmission the respective data byte is placed into the buffer RTB0..3 (which automatically sets bit TRX) and the acknowledge behavior is selected via bit ACKDIS.

For a reception the respective data byte is fetched from the buffer RTB0..3 after IRQD has been activated.

In both cases the data transfer itself is enabled by clearing bits IRQD, IRQP and IRQE which releases the SCL line.

When a stop condition is detected bit SLA is cleared.

The IIC control register CR selects the bus baud rate as well as the activation of SDA and SCL lines. So an external IIC channel can be established (baud rate and physical lines) with one single register access.

Systems that utilize several IIC channels can prepare a set of control words which configure the respective channels. By writing one of these control words to BUSCON the respective channel is selected. Different channels may use different baud rates. Also different operating modes can be selected, e.g. enabling all physical interfaces for a broadcast transmission.

4.1.4 Baud Rate Selection

In order to give the user high flexibility in selection of CPU frequency and baud rate without constraints to baud rate accuracy a flexible baud rate generator has been implemented. It uses two different modes and an additional pre divider. Low baud rates may be configured at high precision in mode 0 which is compatible with older versions. High baud rates may be configured precisely in mode 1.

Mode 0: Reciprocal Divider

The resulting baud rate is:

$$B_{IIC} = \frac{f_{sys}}{4 \cdot (BRP + 1)} \qquad BRP = \frac{f_{sys}}{4 \cdot B_{IIC}} - 1$$

Mode 1: Fractional Divider

The resulting baud rate is

$$B_{IIC} = \frac{f_{sys} \cdot BRP}{1024}$$

$$BRP = \frac{1024 \cdot B_{IIC}}{f_{sys}}$$

Table 4-1 IIC-Bus Baud Rate Selection

BRPMOD = 0	BRP @ 100 kBaud		BRP @ 400 kBaud	
f_{sys} [MHz]	PREDIV=00 _B	PREDIV=01 _B	PREDIV=00 _B	PREDIV=01 _B
100	F9 _H	1E _H	3E _H	07 _H
50	7C _H	0F _H	1E _H	03 _H
24	3B _H	06 _H	0E _H	-
20	31 _H	05 _H	0C _H	-
16	27 _H	04 _H	09 _H	-
12	1D _H	-	06 _H	-
8	13 _H	-	04 _H	-
5	0C _H	-	02 _H	-

BRPMOD = 1	BRP @ 100 kBaud			BRP @ 400kBaud		
f_{sys} [MHz]	PREDIV=00 _B	PREDIV=01 _B	PREDIV=10 _B	PREDIV=00 _B	PREDIV=01 _B	PREDIV=10 _B
100	-	-	42 _H	-	21 _H	-
50	-	-	83 _H	-	42 _H	-
24	-	22 _H	-	-	89 _H	-
20	-	29 _H	-	-	A4 _H	-
16	-	33 _H	-	-	CD _H	-
12	-	44 _H	-	22 _H	-	-
8	-	66 _H	-	33 _H	-	-
5	-	A4 _H	-	52 _H	-	-

Note: Correct functionality can only be guaranteed for 16 MHz < f_{sys} < 100 MHz.

4.1.5 Interrupts

Table 4-2 Interrupt Sources

Interrupt	Signal	SRC Register	Description
Data	IC_INT_D_O	XP0SRC	Interrupt is requested after the acknowledge bit of the last byte has been received or transmitted.
Data Error	IC_INT_D_O	XP0SRC	Interrupt is requested if a multi byte write could not be finished in slave mode because of missing acknowledge, then the data interrupt is followed by an end of transmission interrupt.
Protocol: Arbitration Lost	IIC_INT_P_O	XP1SRC	Interrupt is requested when the IIC module has tried to become master on the bus but has lost the arbitration.
Protocol: Slave Mode after Lost Arbitration	IIC_INT_P_O	XP1SRC	Interrupt is requested if multimaster mode is selected and the IIC module temporarily switches to slave mode after a lost arbitration.
Protocol: Slave Mode after Device Address	IIC_INT_P_O	XP1SRC	Interrupt is requested if multimaster mode is selected and the IIC module temporarily switches to slave mode after a lost arbitration.
Data Transmission End after Stop Condition	IIC_INT_E_O	XP2SRC	Interrupt is requested after transmission is finished by a stop condition.
Data Transmission End after RSC Condition	IIC_INT_E_O	XP2SRC	Interrupt is requested after transmission is finished by a repeated start condition (RSC).
Data Transmission End after missing Acknowledge	IIC_INT_E_O	XP2SRC	Interrupt is also requested if a transmission is stopped by a missing acknowledge.

4.1.6 Synchronization

In Mastermode, the SCL line is controlled by the IIC Module. Sent and received data is only valid if SCL is high. With SCL going down, all modules are starting to count down their low period. During the low period all connected modules are allowed to hold SCL low. As the physical bus connection is wired-AND, SCL will remain low until the device with the longest low period enters high state. Then the device with the shortest high period will pull SCL low again.

4.1.7 Programming

It is strictly recommended not to write to the IIC registers except for interrupt handling, when the IIC is working. This is indicated by the BUM bit (in master mode) and the interrupt flags. In initial mode all registers can be written. In master mode the IIC is working as long as the BUM bit is set, in slave mode the IIC is working from receiving a start condition until receiving the next stop condition. Change of transmit direction is possible only after a protocol interrupt (IRQP) or in initialization mode (MOD = 00_b).

4.1.7.1 Initialization

Before data can be sent or received data buffer size must be set in the bit field CI (only necessary if buffer greater than one byte is available). To decide if slave/master or multimaster mode is required, the MOD bits must be programmed.

4.1.7.2 Repeated Start Condition

The RSC bit must be set to one.

4.1.7.3 Start Condition

To generate a start condition the IIC must be in master mode. If the BUM bit is set, a start condition is sent and the transmission is started. The slave returns the acknowledge bit, which is indicated by the LRB bit.

4.1.7.4 Sending Data Bytes

To send bytes it is only necessary to write data bytes to the transmit buffer every time a data interrupt (IRQD) occurs.

4.1.7.5 Stop Condition

The BUM bit must be set to zero, or the STP bit must be set to one.

4.1.7.6 Receiving Data Bytes

To receive bytes it is necessary to set the TRX bit to zero. The bytes can be read after every data interrupt (IRQD). After a stop condition (protocol interrupt IRQE), the count bit field CO must be read in case of buffer size (defined in CI) is greater than one byte to decide which bytes in the receive buffer were received in the last transmission cycle.

4.2 IIC Kernel Registers

Figure 4-4 and Table 4-3 shows all registers associated with the IIC Kernel.

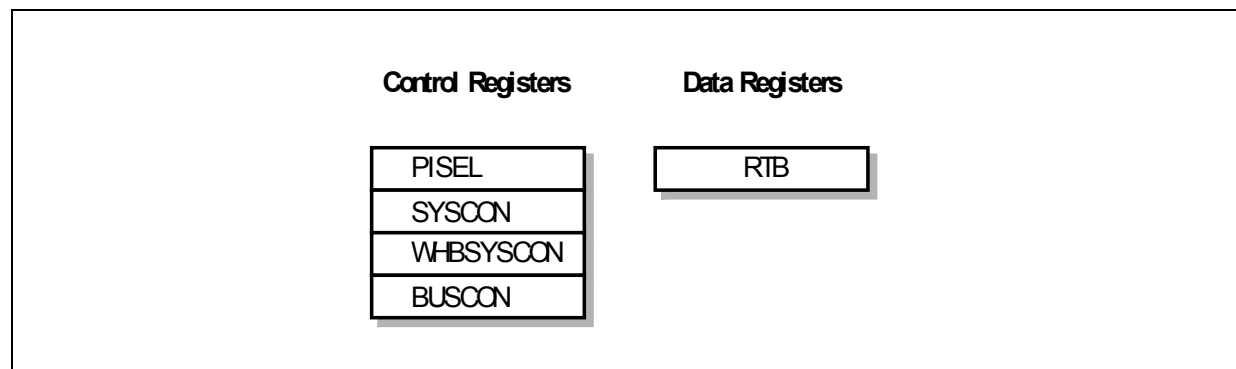


Figure 4-4 IIC Kernel Registers

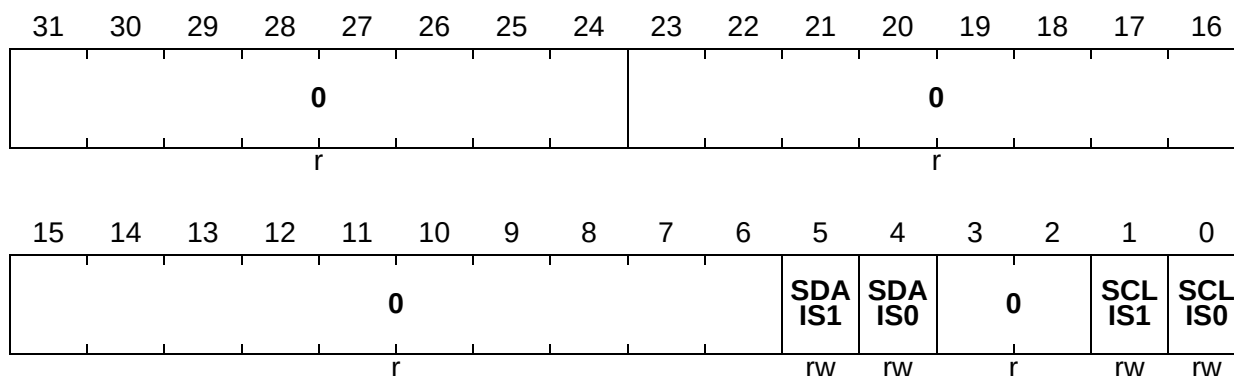
Table 4-3 IIC Kernel Registers

Register Short Name	Register Long Name	Offset Address	Description see
PISEL	Port Input Select Register	0004 _H	Page 4-13
SYSCON	System Control Register	0010 _H	Page 4-14
WHBSYSCON	Write Hardware Bits Control Register	0020 _H	Page 4-20
BUSCON	Bus Control Register	0014 _H	Page 4-23
RTB	Receive Transmit Buffer Register	0018 _H	Page 4-24

For switching between different port input sources the IIC module provides input multiplexer, allowing selection between different input sources. This multiplexer is controlled by the PISEL register.

PISEL

Port Input Select Register



Field	Bits	Type	Description
SCLISx (x = 1..0)	[1:0]	rw	Select Input for Clock Signal x 0 Port A, Default input port 1 Port B, Alternate input port
SDAISx (x = 1..0)	[5:4]	rw	Select Input for Data Signal x 0 Port A, Default input port 1 Port B, Alternate input port
0	[3:2] [31:6]	r	Reserved for future use; reading returns 0; writing to these bit positions has no effect.

The operating mode of the IIC is controlled by the system control register SYSCON. This register contains control bits for mode and error check selection, and status flags for error identification. Depending on bits WMEN and RMEN, either write mirror or receive mirror is enabled. The setting and clearing of the bits SYSCON.AL, SYSCON.IRQD, SYSCON.IDQP, SYSCON.IDQe, SYSCON.RMEN, SYSCON.RSC, SYSCON.BUM, SYSCON.ACKDIS, SYSCON.TRX, SYSCON.STP and SYSCON.WMEN has to be done via the special SYSCON register WHBSYSCON for the setting/clearing of those single bits.

- ¹⁾ While either IRQD, IRQP or IRQE is set and the IIC module is in master mode or has been selected as a slave, the IIC clock line is held low which prevents further transfers on the IIC bus.
 The clock line of the IIC bus is released when IRQD, IRQE and IRQP are cleared. Only in this case the next IIC bus action can take place.
 Interrupt request bits may be set or cleared via software, e.g. to control the IIC bus.

SYSICON

System Control Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WM EN	0	0	0	RM/ CI	STP	IGE	TRX	INT	ACK DIS	BUM	MOD	RSC	M10		
rw	r	r	r	rw	rwh	rw	rwh	rw	rwh	rwh	rw	rwh	rw		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RM EN	0	0	0	0	WM/ CO	IRQE	IRQP	IRQ D	BB	LRB	SLA	AL	ADR		
rwh	r	r	r	r	rh	rwh	rwh	rwh	rh	rh	rh	rwh	rh		

Field	Bits	Type	Description
ADR	0	rh	Address Bit ADR is set after a start condition in slave mode until the address has been received (1 byte in 7-bit address mode, 2 bytes in 10-bit address mode).
AL	1	rwh	Arbitration Lost Bit AL is set when the IIC module has tried to become master on the bus but has lost the arbitration. Operation is continued until the 9th clock pulse. If multimaster mode is selected the IIC module temporarily switches to slave mode after a lost arbitration. Bit IRQP is set along with bit AL. AL must be cleared via software.
SLA	2	rh	Slave 0 The IIC module is not selected as a slave, or the module is in master mode. 1 The IIC module has been selected as a slave (device address received).
LRB	3	rh	Last Received Bit Bit LRB represents the last bit (i.e. the acknowledge bit) of the last transferred frame. It is automatically set to zero by a write or read access to the buffer RTB0..3. <i>Note: If LRB is high (no acknowledge) in slave mode, TRX bit is set automatically.</i>

Field	Bits	Type	Description
BB	4	rh	Bus Busy 0 The IIC bus is idle, i.e. a stop condition has occurred. 1 The IIC bus is active, i.e. a start condition has occurred. <i>Note: Bit BB is always 0 while the IIC module is disabled.</i>
IRQD	5	rwh	IIC Interrupt Request Bit for Data Transfer Events ¹⁾ 0 No interrupt request pending. 1 A data transfer event interrupt request is pending. IRQD is set after the acknowledge bit of the last byte has been received or transmitted, and is cleared automatically upon a complete read or write access to the buffer(s) RTB0..3. New data transfers will start immediately after clearing IRQD. Do not access any register until next interrupt. If a multi byte write could not be finished in slave mode because of missing acknowledge, then the data interrupt is followed by an end of transmission interrupt. The number of bytes sent can be read from CO. The data interrupt must have higher priority than IRQE.
IRQP	6	rwh	IIC Interrupt Request Bit for Protocol Events ¹⁾ 0 No interrupt request pending. 1 A protocol event interrupt request is pending. IRQP is set when bit SLA or bit AL is set (✓), and must be cleared via software. If the IIC has been selected by another master, the software must look up the required transmission direction by reading the received address and direction bit, stored in RTB0. The TRX-bit must be set by software correspondingly.

Field	Bits	Type	Description
IRQE	7	rwh	IIC Interrupt Request Bit for Data Transmission End ¹⁾ 0 No interrupt request pending. 1 A receive end event interrupt request is pending (a stop is detected). IRQE is automatically cleared upon a start condition. IRQE is not activated in init-mode. IRQE must always be deleted to continue transmission. <i>Note: In slave mode IRQE is set after the transmission is finished. This can also be after a stop or RSC condition. In this case the slave is not selected any more. This bit is also set, if a transmission is stopped by a missing acknowledge. In this case the bit must be cleared by software.</i>
CO	[10:8]	rh	Counter of Transmitted Bytes Since Last Data Interrupt. If a multi byte transmission could not be finished because of missing acknowledge, the number of correctly transferred bytes can be read from CO. It is automatically set to zero by the correct number (defined by CI) of write/read accesses to the buffers ICRTB0..3. 000 No Byte 001 1 Byte 010 2 Bytes 011 3 Bytes 100 4 Bytes The number of legal bytes depends on the data buffer size (CI). Writing to this bit field does not affect its content. If WMEN is set, WM is mirrored here.
0	[14:11]	r	Reserved , do not use, read/write zero.

Field	Bits	Type	Description
RMEN	15	rwh	Read Mirror Enable 0 Read mirror is not active 1 Read mirror is active <i>Note: If WMEN is set RMEN cannot be set and will remain zero. If RMEN and WMEN are set simultaneously to 1, both will remain what they are. So only one of both can be set to 1.</i>
WM	[15:8]	wh	Write Mirror If WMEN is set, RTB0 may be written here. Reading WM will result zero.
M10	16	rw	Address Mode 0 7-bit addressing using ICA7..1. 1 10-bit addressing using ICA9..0.
RSC	17	rwh	Repeated Start Condition 0 No operation. 1 Generate a repeated start condition in (multi) master mode. RSC cannot be set in slave mode. <i>Note: RSC is cleared automatically after the repeated start condition has been sent.</i>
MOD	[19:18]	rw	Basic Operating Mode 00 IIC module is disabled and initialized (Init-Mode). Transmissions under execution will be aborted. 01 Slave mode. 10 Master mode. 11 Multi-Master mode.
BUM	20	rwh	Busy Master 0 Clearing bit BUM ($\bar{B}$) generates a stop condition immediately. 1 Setting bit BUM (B) generates a start condition in (multi)master mode. <i>Note: Setting bit BUM (B) while BB = 1 generates an arbitration lost situation. In this case BUM is cleared and bit AL is set. BUM cannot be set in slave mode.</i>

Field	Bits	Type	Description
ACKDIS	21	rwh	Acknowledge Pulse Disable 0 An acknowledge pulse is generated for each received frame. 1 No acknowledge pulse is generated. <i>Note: ACKDIS is automatically cleared by a stop condition.</i>
INT	22	rw	Interrupt Delete Select 0 Interrupt flag IRQD is deleted by read/write to RTB0..3. 1 Interrupt flag IRQD is not deleted by read/write to RTB0..3.
TRX	23	rwh	Transmit Select 0 No data is transmitted to the IIC bus. 1 Data is transmitted to the IIC bus. <i>Note: TRX is set automatically when writing to the transmit buffer. It is not allowed to delete this bit in the same buscycle. It is automatically cleared after last byte as slave transmitter.</i>
IGE	24	rw	Ignore IRQE Ignore IRQE (End of transmission) interrupt. 0 The IIC is stopped at IRQE interrupt. 1 The IIC ignores the IRQE interrupt. <i>Note: If RMEN is set, RM is mirrored here.</i>
STP	25	rwh	Stop Master 0 Clearing bit STP generates no stop condition. 1 Setting bit STP generates a stop condition after next transmission. BUM is set to zero. ACKDIS is set to one. <i>Note: STP is automatically cleared by a stop condition. If RMEN is set, RM is mirrored here.</i>
CI	[27:26]	rw	Length of the Receive/Transmit Buffer 00 1 Byte 01 2 Bytes 10 3 Bytes 11 4 Bytes <i>Note: If RMEN is set, RM is mirrored here.</i>

Field	Bits	Type	Description
0	[30:28]	r	Reserved , do not use, read/write zero.
WMEN	31	rwh	Write Mirror Enable 0 write mirror is not active 1 write mirror is active <i>Note: If RMEN is set WMEN cannot be set and will remain zero. If WMEN and RMEN are set simultaneously to 1, both will remain what they are. So only one of both can be set to 1.</i>
RM	[31:24]	rh	Read Mirror If RMEN is set, RTB0 may be read here. Writing to RM has no effect in this mode.

*Note: This register contains critical rwh bits. In case of 32-bit bus systems, these bits may not be modified directly. Please use the associated **WHBSYSCON** register instead.*

WHBSYSCON

Write Hardware Bits Control Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SET WM EN	CLR WM EN		0		SET STP	CLR STP	SET TRX	CLR TRX	SET ACK DIS	CLR ACK DIS	SET BUM	CLR BUM	0	SET RSC	CLR RSC
w	w		r		w	w	w	w	w	w	w	w	r	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SET RM EN	CLR RM EN		0		SET IRQ E	SET IRQ P	SET IRQ D	CLR IRQ E	CLR IRQ P	CLR IRQ D		0	SET AL	CLR AL	0
w	w		r		w	w	w	w	w	w		r	w	w	r

Field	Bits	Type	Description
CLRAL	1	w	Clear Arbitration Lost Bit Writing 1 to this bit clears bit SYSCON.AL. Writing 0 has no effect. Reading returns 0 always.
SETAL	2	w	Set Arbitration Lost Bit Writing 1 to this bit sets bit SYSCON.AL. Writing 0 has no effect. Reading returns 0 always.
CLRIRQD	5	w	Clear IIC Interrupt Request Bit for Data Transfer Events Bit Writing 1 to this bit clears bit SYSCON.IRQD. Writing 0 has no effect. Reading returns 0 always.
CLRIRQP	6	w	Clear IIC Interrupt Request Bit for Protocol Events Bit Writing 1 to this bit clears bit SYSCON.IRQP. Writing 0 has no effect. Reading returns 0 always.
CLRIRQE	7	w	Clear IIC Interrupt Request Bit for Data Transmission End Bit Writing 1 to this bit clears bit SYSCON.IRQE. Writing 0 has no effect. Reading returns 0 always.

Field	Bits	Type	Description
SETIRQD	8	w	Set IIC Interrupt Request Bit for Data Transfer Events Bit Writing 1 to this bit sets bit SYSCON.IRQD. Writing 0 has no effect. Reading returns 0 always.
SETIRQP	9	w	Set IIC Interrupt Request Bit for Protocol Events Bit Writing 1 to this bit sets bit SYSCON.IRQP. Writing 0 has no effect. Reading returns 0 always.
SETIRQE	10	w	Set IIC Interrupt Request Bit for Data Transmission End Bit Writing 1 to this bit sets bit SYSCON.IRQE. Writing 0 has no effect. Reading returns 0 always.
CLRRMEN	14	w	Clear Read Mirror Enable Bit Writing 1 to this bit clears bit SYSCON.RMEN. Writing 0 has no effect. Reading returns 0 always.
SETRMEN	15	w	Set Read Mirror Enable Bit Writing 1 to this bit sets bit SYSCON.RMEN. Writing 0 has no effect. Reading returns 0 always.
CLRRSC	16	w	Clear Repeated Start Condition Bit Writing 1 to this bit clears bit SYSCON.RSC. Writing 0 has no effect. Reading returns 0 always.
SETRSC	17	w	Set Repeated Start Condition Bit Writing 1 to this bit sets bit SYSCON.RSC. Writing 0 has no effect. Reading returns 0 always.
CLRBUM	19	w	Clear Busy Master Bit Writing 1 to this bit clears bit SYSCON.BUM. Writing 0 has no effect. Reading returns 0 always.

Field	Bits	Type	Description
SETBUM	20	w	Set Busy Master Bit Writing 1 to this bit sets bit SYSCON.BUM. Writing 0 has no effect. Reading returns 0 always.
CLRACKDIS	21	w	Clear Acknowledge Pulse Disable Bit Writing 1 to this bit clears bit SYSCON.ACKDIS. Writing 0 has no effect. Reading returns 0 always.
SETACKDIS	22	w	Set Acknowledge Pulse Disable Bit Writing 1 to this bit sets bit SYSCON.ACKDIS. Writing 0 has no effect. Reading returns 0 always.
CLRTRX	23	w	Clear Transmit Select Bit Writing 1 to this bit clears bit SYSCON.TRX. Writing 0 has no effect. Reading returns 0 always.
SETTRX	24	w	Set Transmit Select Bit Writing 1 to this bit sets bit SYSCON.TRX. Writing 0 has no effect. Reading returns 0 always.
CLRSTP	25	w	Clear Stop Master Bit Writing 1 to this bit clears bit SYSCON.STP. Writing 0 has no effect. Reading returns 0 always.
CLRWMEN	30	w	Set Write Mirror Enable Bit Writing 1 to this bit sets bit SYSCON.WMEN. Writing 0 has no effect. Reading returns 0 always.
SETWMEN	31	w	Clear Write Mirror Enable Bit Writing 1 to this bit clears bit SYSCON.WMEN. Writing 0 has no effect. Reading returns 0 always.
0	0, [4:3] 18, [13:11] [29:27]	r	Reserved ; read as 0; should be written with 0.

BUSCON

Bus Control Register

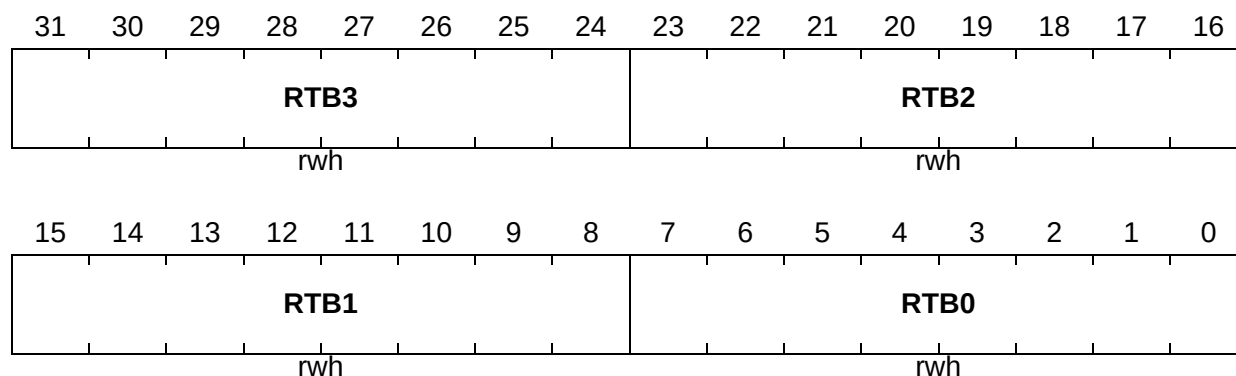
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BRP MOD	PREDIV		0			ICA9 /0	ICA8 /0	ICA7..1							ICA0 /0
rw	rw		r			rw	rw	rw							rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BRP								0	SCL EN1	SCL EN0	0		SDA EN1	SDA EN0	
rw								r	rw	rw	r		rw	rw	

Field	Bits	Type	Description
SDAENx (x = 1..0)	[1:0]	rw	Enable Input for Data Pin x These bits determine to which pins the IIC data line is connected. 0 SDA pin x is disconnected. 1 SDA pin x is connected with IIC data line.
SCLENx (x = 1..0)	[5:4]	rw	Enable Input for Clock Pin x These bits determine to which pins the IIC clock line is connected. 0 SCL pin x is disconnected. 1 SCL pin x is connected with IIC clock line.
BRP	[15:8]	rw	Baud Rate Prescaler Determines the baud rate for the active IIC channel(s). The prescaler may operate in two modes. Bit BRPMOD selects the actual mode. Bit field PREDIV selects an additional pre divider. <i>Note: See Table 4-1 below.</i>
ICA0	16	rw	Node Address Bit 0 in 10-Bit Mode (See SYSCON bit M10) <i>Note: Access is only possible in 10-bit mode.</i>
0	[25:24], 16	r	Reserved read/write 0 if in 7-bit mode.
ICA7..1	[23:17]	rw	Node Address in 7-Bit Mode (ICA9, ICA8 and ICA0 disregarded)
ICA9..0	[25:16]	rw	Node Address in 10-Bit Mode (all bits used). <i>Note: Access is only possible in 10-bit mode.</i>

Field	Bits	Type	Description
0	[3:2], [7:6], [28:26]	r	Reserved ; read as 0; should be written with 0.
PREDIV	[30:29]	rw	Pre Divider for Baud Rate Generation 00 pre divider is disabled 01 pre divider factor 8 is enabled 10 pre divider factor 64 is enabled 11 reserved, do not use <i>Note: See Table 4-1.</i>
BRPMOD	31	rw	Baud Rate Prescaler Mode 0 Mode 0 is enabled (by default) 1 Mode 1 is enabled. <i>Note: See Table 4-1.</i>

RTB

Receive Transmit Buffer



Field	Bits	Type	Description
RTBx (x= 3..0)	[31:0]	rwh	Receive/Transmit Buffer ¹⁾ The buffers contain the data to be sent/received. The buffer size can be set in bit field CI (from 1 up to 4 bytes). RTB0 is sent/received first.

¹⁾) If bit INT is set to zero and all bytes (specified in CI) of RTB0..3 are read/written (dependent on bit TRX) IRQD will be cleared by hardware after completion of this access.

4.3 IIC Module Implementation

This section describes the IIC module interfaces with port connections, interrupt control, and address decoding and the requirement in TC1100.

4.3.1 Interfaces of the IIC Module

In TC1100, only two physical IIC Buses are implemented. **Figure 4-5** shows the TC1100 specific implementation details and interconnections of the IIC module. The IIC module has four I/O lines, located at Port 2. The IIC module is further supplied by a clock control, interrupt control, and address decoding logic. One DMA request can be generated by IIC module.

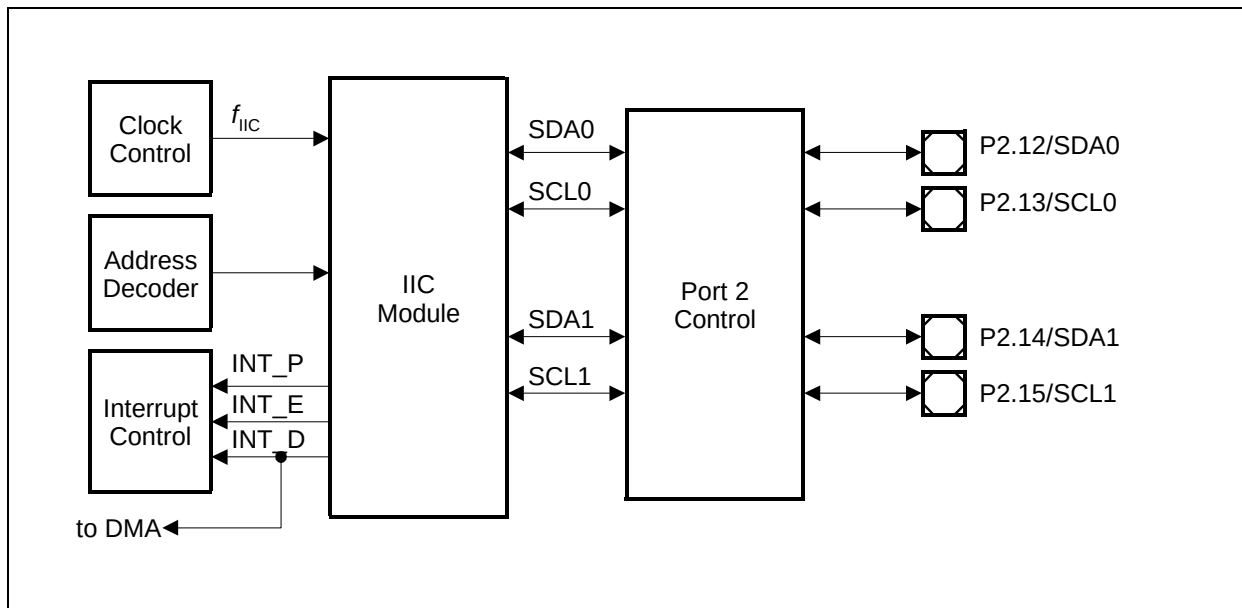


Figure 4-5 IIC Module Implementation and Interconnections

4.3.2 IIC Module Related External Registers

Figure 4-6 summarizes the module related external registers which are required for IIC programming (see also **Figure 4-4** for the module kernel specific registers).

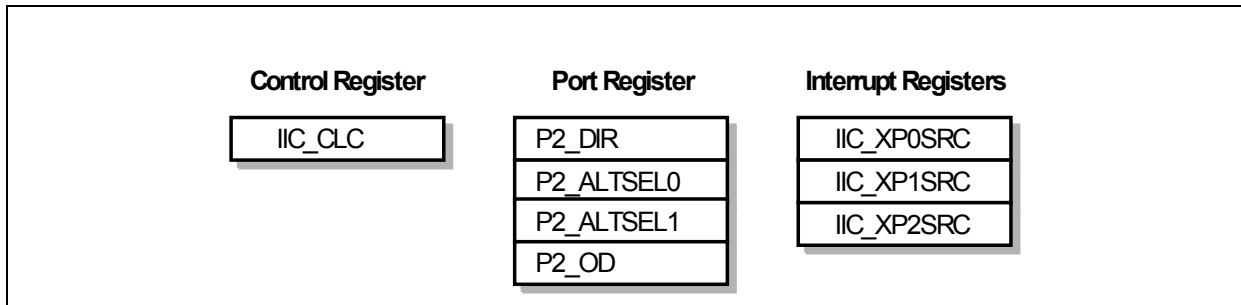


Figure 4-6 IIC Implementation Specific Special Function Registers

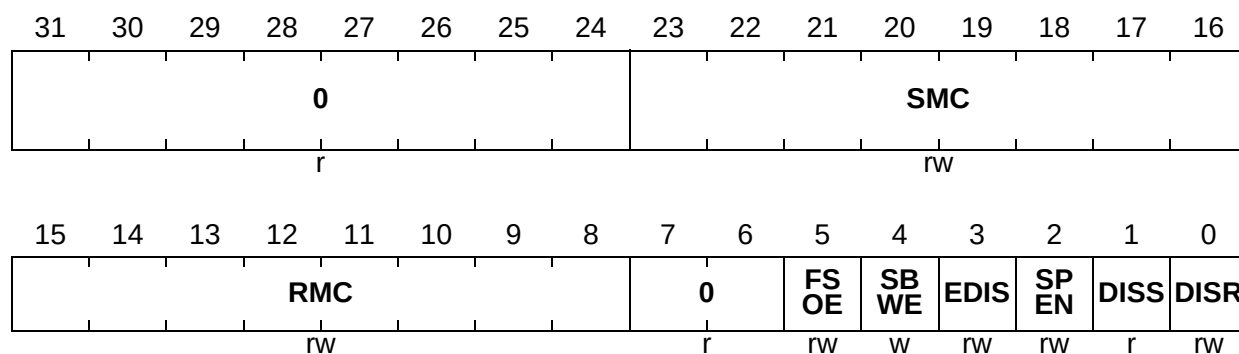
4.3.2.1 Clock Control Register

IIC_CLC is used to control the f_{IIC} clock signal.

IIC_CLC

IIC Clock Control Register

Reset Value: 0000 0002_H



Field	Bits	Type	Description
DISR	0	rw	Module Disable Request Bit Used for enable/disable control of the module.
DISS	1	r	Module Disable Status Bit Bit indicates the current status of the module.
SPEN	2	rw	Module Suspend Enable for OCDS Used for enabling the suspend mode.
EDIS	3	rw	External Request Disable Used for controlling the external clock disable request.
SBWE	4	w	Module Suspend Bit Write Enable for OCDS Defines whether SPEN and FSOE are write protected.
FSOE	5	rw	Fast Switch Off Enable Used for fast clock switch off in OCDS suspend mode.
RMC	[15:8]	rw	8-Bit Clock Divider Value in RUN Mode
SMC	[23:16]	rw	Clock Divider Addition for Sleep Mode Max. 8-bit adding value. <i>Note: FPI Bus only.</i>
0	[31:24]	r	Reserved ; read as 0; should be written with 0.

4.3.2.2 Port Registers

The interconnections between the IIC module and the port I/O lines are controlled in the port logics. The following port control operations selections must be executed:

- Input/output function selection (DIR register)
- Alternate function selection (ALTSEL0 and ALTSEL1 registers)
- Pad driver characteristics selection for the outputs (OD register)

Input/Output Function Selection

The port input/output control registers contain the bit fields that select the digital output and input driver characteristics such as port direction (input/output), open-drain, and alternate output selections. The I/O lines for the IIC modules are controlled by the port input/output control registers of Port 2.

Table 4-4 shows how bits and bit fields must be programmed for the required I/O functionality of the IIC I/O lines.

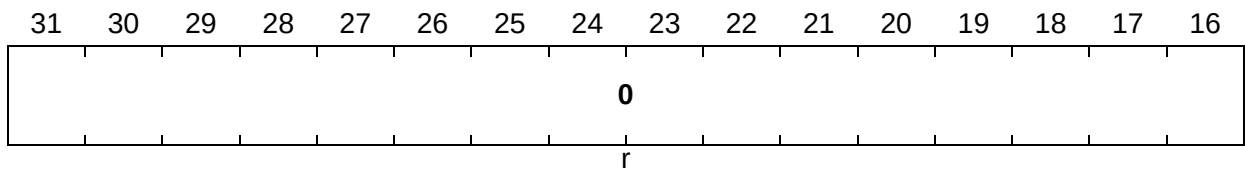
Table 4-4 IIC I/O Control Selection and Setup

Module	Port Lines	Input/Output Control Register Bits	I/O
IIC	P2.12/SDA0	P2_DIR.P12 = 0 _B	Input
		P2_DIR.P12 = 1 _B	Output
		P2_ALTSEL0.P12 = 1 _B	
		P2_ALTSEL1.P12 = 0 _B	
	P2.13/SCL0	P2_DIR.P13 = 0 _B	Input
		P2_DIR.P13 = 1 _B	Output
		P2_ALTSEL0.P13 = 1 _B	
		P2_ALTSEL1.P13 = 0 _B	
	P2.14/SDA1	P2_DIR.P14 = 0 _B	Input
		P2_DIR.P14 = 1 _B	Output
		P2_ALTSEL0.P14 = 1 _B	
		P2_ALTSEL1.P14 = 0 _B	
	P2.15/SCL1	P2_DIR.P15 = 0 _B	Input
		P2_DIR.P15 = 1 _B	Output
		P2_ALTSEL0.P15 = 1 _B	
		P2_ALTSEL1.P15 = 0 _B	

P2_DIR

Port 2 Direction Register

Reset Value: 0000 0000_H



15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
P15	P14	P13	P12	P11	P10	P9	P8	P7	P6	P5	P4	P3	P2	P1	P0
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

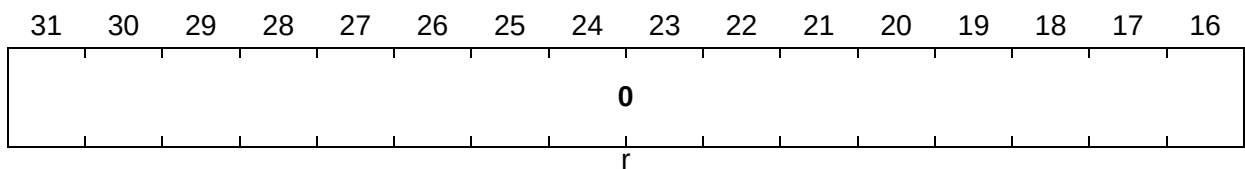
Field	Bits	Type	Description
Pn (n = 12 -15)	n	rW	Port 2 Pin 12-15 Direction Control ¹⁾ 0 Direction is set to input (default after reset) 1 Direction is set to output
0	[31:16]	r	Reserved ; read as 0; should be written with 0.

¹⁾ Shaded bits and bit field are don't care for IIC I/O port control

P2_ALTSELn (n = 1, 0)

Port 2 Alternate Select Register

Reset Value: 0000 0000_H



15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
P15	P14	P13	P12	P11	P10	P9	P8	P7	P6	P5	P4	P3	P2	P1	P0
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

Table 4-5 Function of the Bits P2_ALTSEL0.Pn and P2_ALTSEL1.Pn (n = 12-15)¹⁾

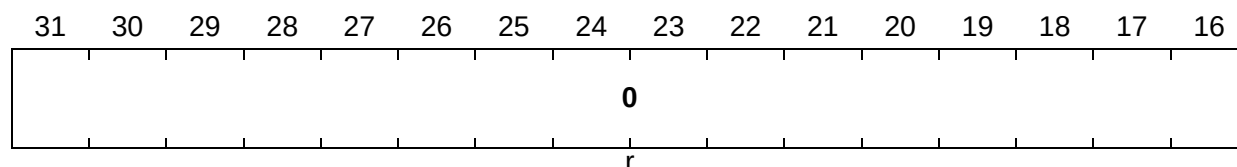
P2_ALTSEL0.Pn	P2_ALTSEL1.Pn	Function
1	0	Alternate Select1

¹⁾ Shaded bits and bit field are don't care for IIC I/O port control

P2_OD

Port 2 Open Drain Control Register

Reset Value: 0000 F000_H



15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
P15	P14	P13	P12	P11	P10	P9	P8	P7	P6	P5	P4	P3	P2	P1	P0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Field	Bits	Type	Description
Pn (n = 12-15)	n	rw	Port 2 Pin n Open Drain Mode¹⁾ 0 Normal Mode, output is actively driven for 0 and 1 state 1 Open Drain Mode, output is actively driven only for 0 state
0	[31:16]	r	Reserved ; read as 0; should be written with 0.

¹⁾ Shaded bits and bit field are don't care for IIC I/O port control

4.3.2.3 Service Request Control Registers

Each of the eight interrupts of the IIC module are controlled by its own service request control registers.

IIC_XP0SRC

IIC Service Request Control Register 0

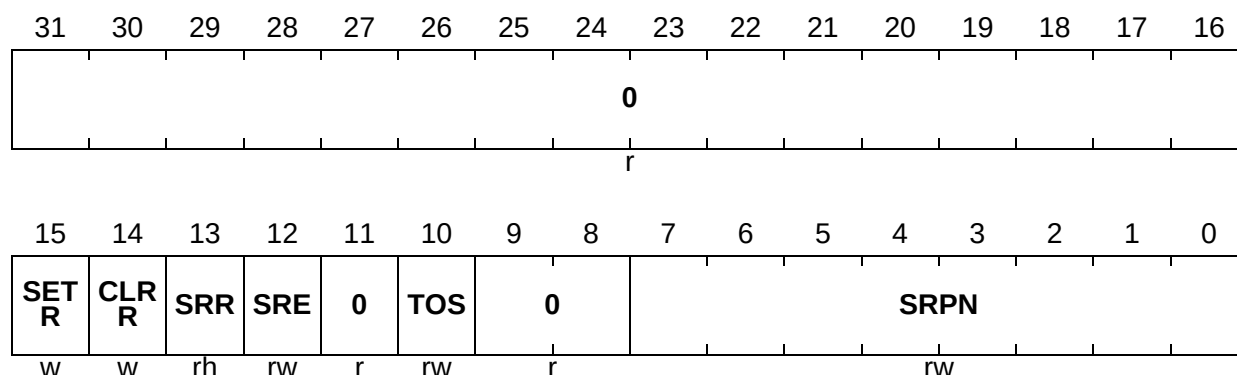
IIC_XP1SRC

IIC Service Request Control Register 1

IIC_XP2SRC

IIC Service Request Control Register 2

Reset Values: 0000 0000_H



Field	Bits	Type	Description
SRPN	[7:0]	rw	Service Request Priority Number
TOS	10	rw	Type of Service Control
SRE	12	rw	Service Request Enable
SRR	13	rh	Service Request Flag
CLRR	14	w	Request Clear Bit
SETR	15	w	Request Set Bit
0	[9:8], 11 [31:16]	r	Reserved; read as 0; should be written with 0.

For proper operation of an IIC function controlled by an interrupt service routine, the following conditions should be checked:

- An interrupt request can only be serviced if the respective Service Request Enable Bit (IIC_XPxSRC.SRE) is set to 1.
- The exact source of an interrupt request should be identified by analyzing the system Control register SYSCON and WHBSYSCON.

- The Service Request Priority Number bit field SRPN defines the sequence for the CPU arbitration in case of simultaneously set Interrupt Service Request Flags. That requires careful estimation of the IIC service request priorities depending on the real time characteristic of higher prioritized interrupt sources, the CPU load, and the timing constraints to be matched by an IIC interrupt service routine.

Note: Further details on interrupt handling and processing are described in the “Interrupt System” chapter of the TC1100 System Units User’s Manual.

4.3.3 DMA Requests

The DMA request lines of the IIC modules become active whenever its related interrupt line is activated. The DMA request lines are connected to the DMA controller as shown in [Table 4-6](#).

Table 4-6 DMA Request Lines of IIC

Module	Related IIC Interrupt	DMA Request Line	Description
IIC	INT_D	IIC_INTD	IIC Data Interrupt DMA Request

4.3.4 IIC Register Address Range

In the TC1100, the registers of the IIC module are located within the following address range:

- Module Base Address = F010 0600_H
Module End Address = F010 06FF_H
- Absolute Register Address = Module Base Address + Offset Address
(offset addresses see [Table 4-3](#))

Note: The complete and detailed address map of the IIC module is described in the chapter “Register Overview” of the TC1100 System Units User’s Manual.

5 Micro Link Serial Bus Interface

This chapter describes the micro link serial bus interface MLI0 of the TC1100. It contains the following sections:

- Functional description of the MLI kernel, valid for MLI0 (see [Section 5.1](#))
- MLI kernel register descriptions of all MLI kernel specific registers (see [Section 5.2](#))
- TC1100 implementation specific details and registers of the MLI0 module (port connections and control, interrupt control, address decoding and clock control, see [Section 5.3](#))

Note: All MLI kernel register names described in this section will be referenced in other parts of the TC1100 User's Manual with the module name prefix "MLI0_" for the MLI0 interface.

5.1 MLI Kernel Description

5.1.1 MLI Applications

- Data and program exchanging without intervention of CPU or PCP between microcontrollers of the AUDDO - NG family.
- The internal architecture of the block allows the communication between controllers in different clock domains.
- Compatibility with the SSC interface.
- The read mode also makes it possible to ask the other controller for the desired data.
- Resources sharing. It is possible to use resources not available in the controller but present in another one.
- Capability of programming up to four different interrupts in the second controller by sending a command.

Figure 5-1 shows a general overview of the MLI location in the controller and its connection within another MLI.

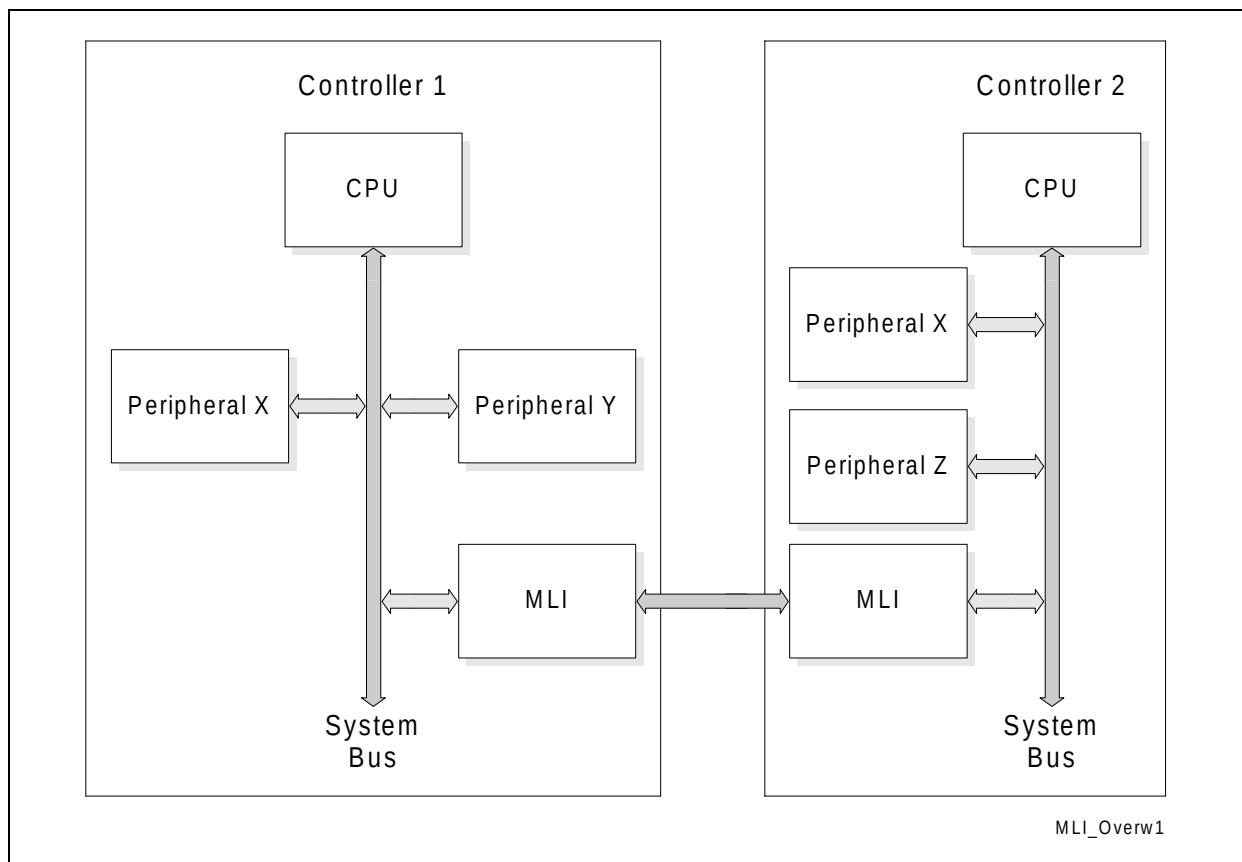


Figure 5-1 Location of MLI in the Controller and Connection

Figure 5-2 shows a global view of all functional blocks of the MLI Module.

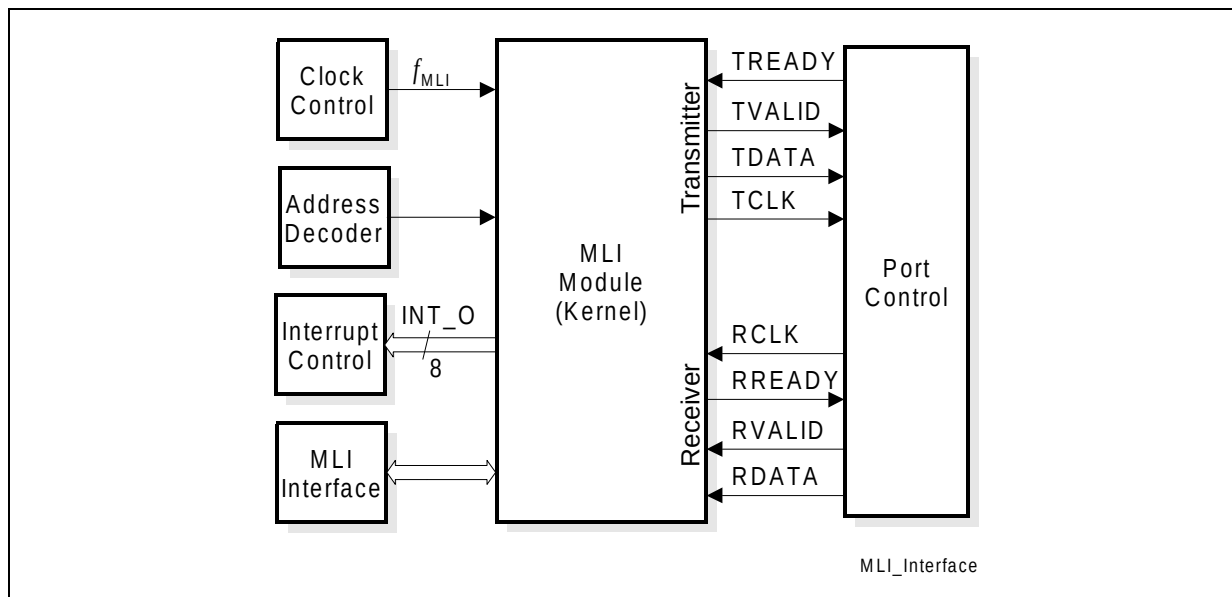


Figure 5-2 General Block Diagram of the MLI Module

5.1.2 Overview

The Micro Link Serial Bus Interface, referenced as MLI in the whole chapter, is dedicated for the serial communication between controllers of the AUDO - NG family. The communication is intended to be fast and intelligent due to an address translation system, and it is not necessary to have any special program in the second controller.

An overview of the MLI kernel is shown in [Figure 5-3](#).

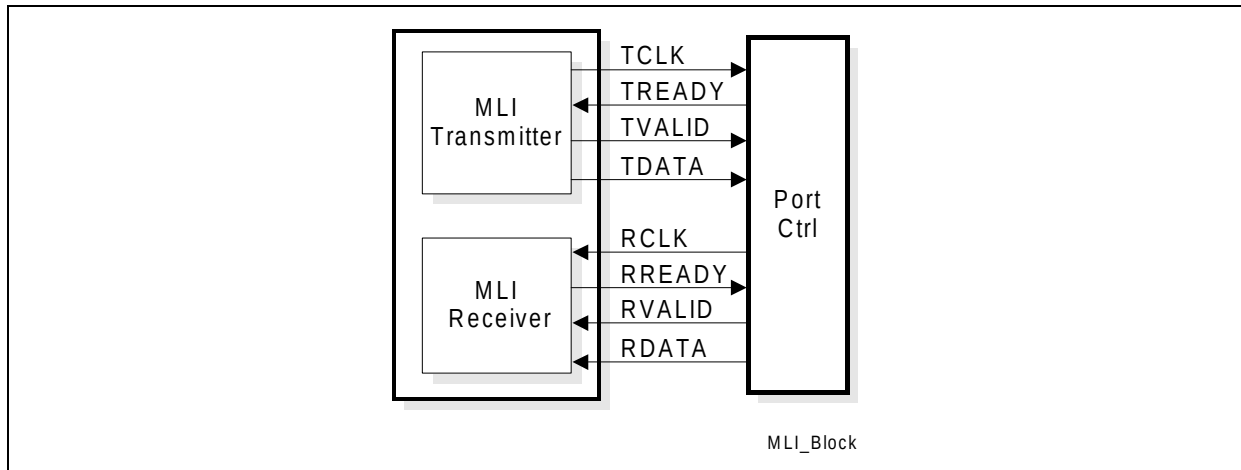


Figure 5-3 MLI Overview

Note: The prefixes “T” and “R” indicate if the corresponding signals belong to the MLI transmitter or to the MLI receiver.

Features:

- Serial communication from the MLI transmitter to MLI receiver of another controller
- Module supports connection of each MLI with up to four MLI from other controllers
- Fully transparent read/write access supported (= remote programming)
- Complete address range of target controller available
- Special protocol to transfer data, address offset, or address offset and data
- Error control using a parity bit
- 32-bits, 16-bits, and 8-bits data transfers
- Address offset width: from 1- to 16-bits
- Baud rate: $f_{MLI} / 2$ (symmetric shift clock approach),
baud rate definition by the corresponding fractional divider

5.1.2.1 Naming Conventions

Local and Remote Controller

The names “local” and “remote” controller (device) are assigned to the two partners (microcontrollers with MLI modules) of a serial MLI connection. The controller with an MLI module that operates as a master of the serial MLI connection is defined as local controller. A local controller handles data operations with transfer windows and further initiates all control tasks (control, address, and data transmissions) that are required for the data transfer/request between local and remote controller.

The controller with an MLI module that operates as a slave of the serial MLI connection is defined as remote controller. A remote controller handles data operations with remote windows and executes the tasks that have been assigned/requested by the local controller.

Due to the full duplex operation capability of an MLI module, two serial MLI connections can be installed simultaneously (both transmitters can send a frame to their receivers). This means, each microcontroller with an MLI module is able to operate as local controller as well as remote controller at the same time.

Transfer Window

A transfer window is an address space in the address map of the local controller that is typically not assigned to memories or peripheral units. Transfer windows are always assigned to fixed address space (base address and size) in a specific microcontroller. Each MLI module supports up to four transfer windows with two different window sizes: four small transfer windows with 8 KByte and four large transfer windows with 64 KByte. Address and data information that has been written or read to/from transfer windows can be detected and handled by the MLI module of the local controller.

Remote Window

A remote window is an area in the address space of the remote controller. Remote window parameters (base address and window size) are defined and controlled by the local microcontroller. The size of a remote window is defined in a 4-bit coded buffer size parameter that defines the number of variable address bits of a remote window. Each MLI module supports up to four remote windows.

Pipe

A pipe defines the logical connection between an MLI module in the local controller and an MLI module in the remote controller. The logical connection of a pipe maps the transfer window in the local controller to its corresponding remote window in the remote controller. The MLI module supports four pipes.

5.1.2.2 MLI Communication Principles

The communication principle of the MLI module allows that data is transferred between a local and a remote controller without intervention of a CPU. Data transfers are always triggered in the local controller by read or write operations to memory locations that are located in a transfer window of the local controller. All control tasks (control, address, and data transmissions) that are required for the data read/write accesses between local and remote controller are handled autonomously by the MLI module.

A **write access to a location within a transfer window** of the local controller is detected by the MLI transmitter. This detection initiates a transfer of the data that has been written to the transfer window from the local microcontroller to the MLI receiver of the remote controller which places the data at an address location in a remote window of the remote controller.

A **read access from a location of a transfer window** in the local controller returns dummy data and initiates the MLI connection to request data from the remote controller. Data is read in the remote controller by the MLI module from an address location within the remote window and transferred to the local controller where it is stored in its MLI receiver registers. Afterwards, the CPU in the local controller is informed by an interrupt that the requested data is now available and can be read from a register.

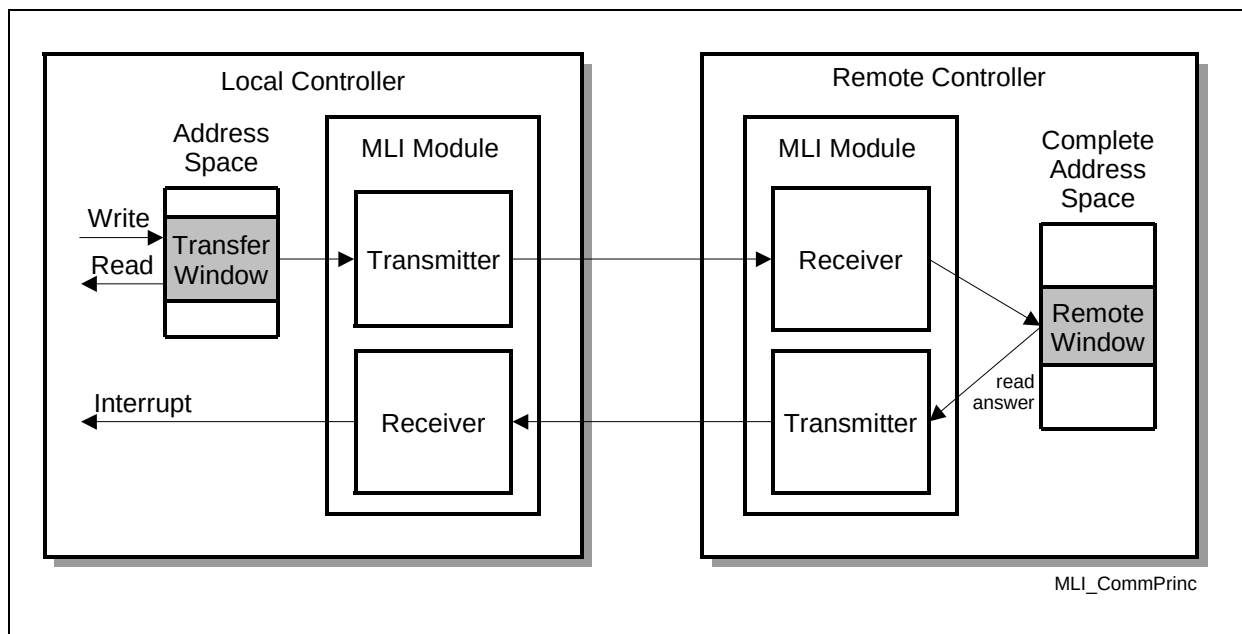


Figure 5-4 MLI Communication Principles

5.1.3 General Description

The communication between both controllers is based in an address translation table that allows the MLI transmitter from the first controller just sending an offset relative to these addresses instead of the full 32-bits address. The consistency for these addresses is guaranteed because the first controller sends all of them to the second controller and this one stores them as explained in [Section 5.1.7.5](#) and in [Section 5.1.7.6](#). Each of the addresses defines a pipe and together with the Buffer Size parameter will define a buffer in the second controller address map. For each MLI transmitter there will be up to four pipes. A pipe may be seen as a logical connection between two controllers.

Figure 5-7 shows the organization in memory of the MLI transfer windows and their possible correspondence in the second controller address map.

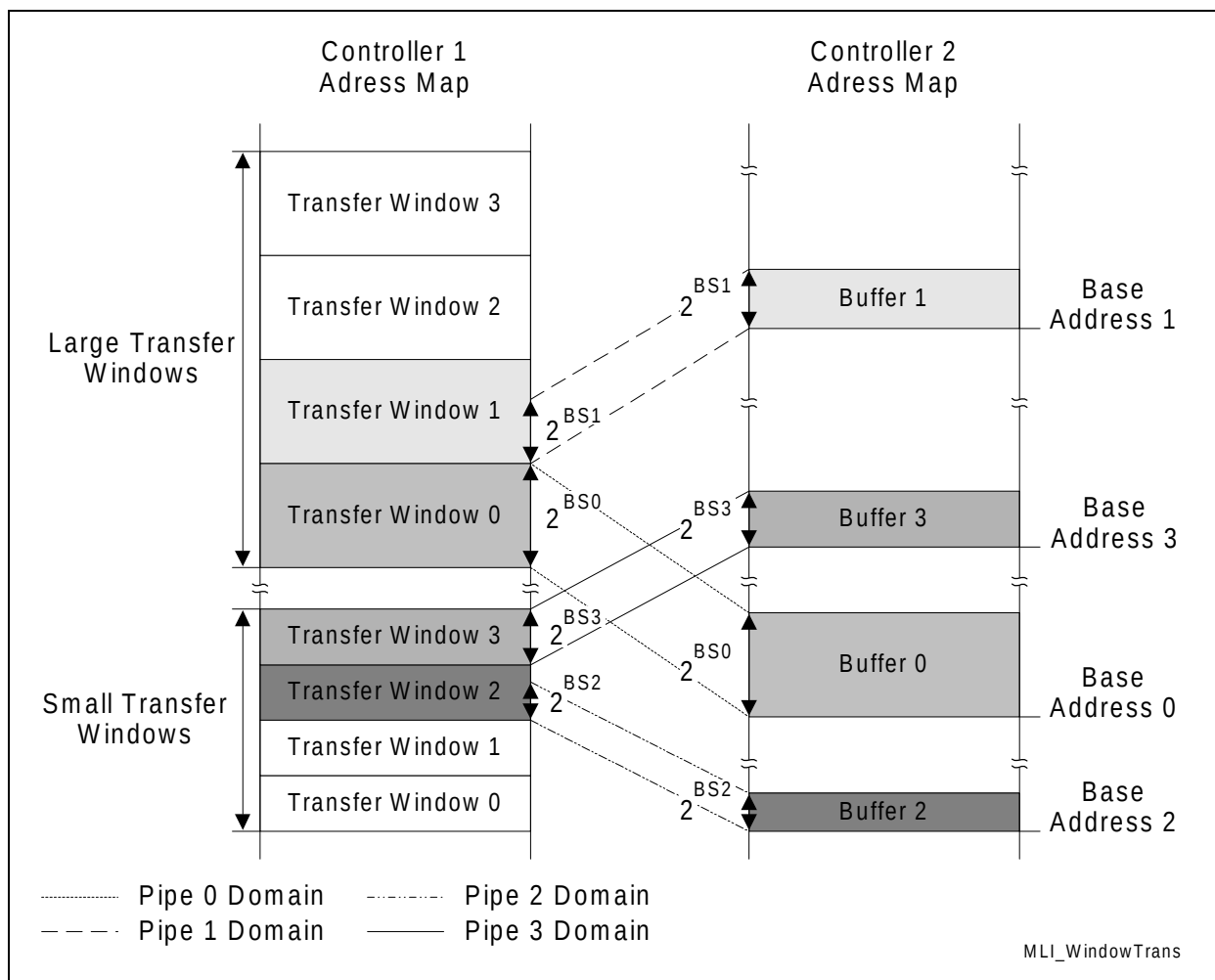


Figure 5-5 Transfer Window Base Address Copy

Note: BSx is the buffer size of each of the different pipes (where x = 0, 1, 2, 3).

Figure 5-6 illustrates a general overview of a two MLI connected, in which the transfer windows, TW, and the move engine of the receiver have been detailed.

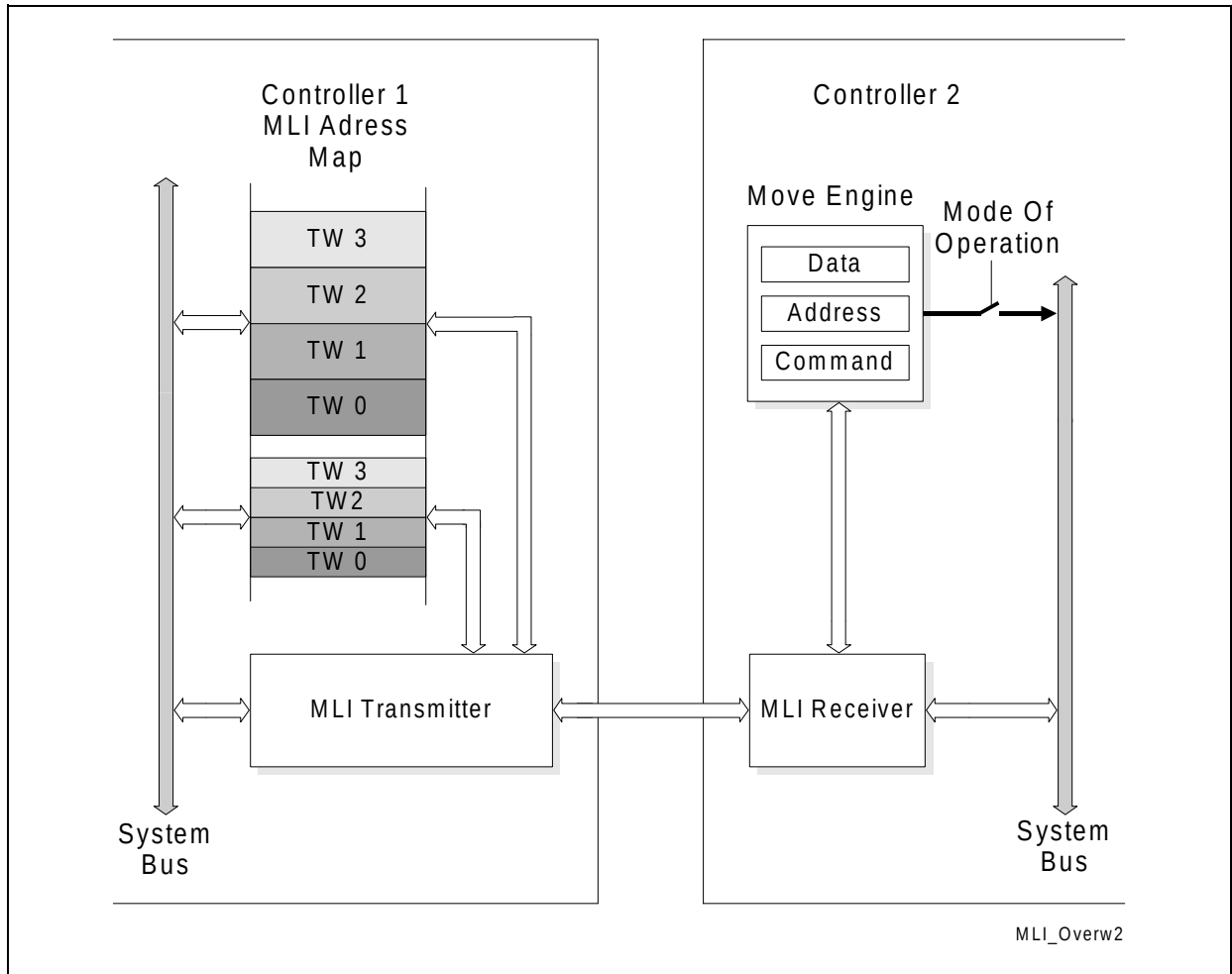


Figure 5-6 MLI Connection Overview

A **write access to a transfer window** in controller 1 leads to a transfer from the MLI transmitter to the MLI receiver on controller 2. The received information (incl. data and address or the command) is stored in the MLI receiver. There it is available for the CPU of controller 2 or the move can be executed autonomously by the MLI move engine (according to the selected access protection).

A **read access to a transfer window** delivers a dummy value on the system bus of controller 1. The read request is transferred to the MLI receiver on controller 2. If enabled, the MLI move engine executes the read operation autonomously and the requested data will be sent back to the MLI on controller 1 (by the MLI transmitter on controller 2 to the MLI receiver of controller 1). When this information is available in the MLI module of controller 1, an interrupt can be generated and the CPU (or a DMA, etc.) of controller 1 can read the requested data.

Figure 5-7 shows the process of transmission of the base address for each transfer window.

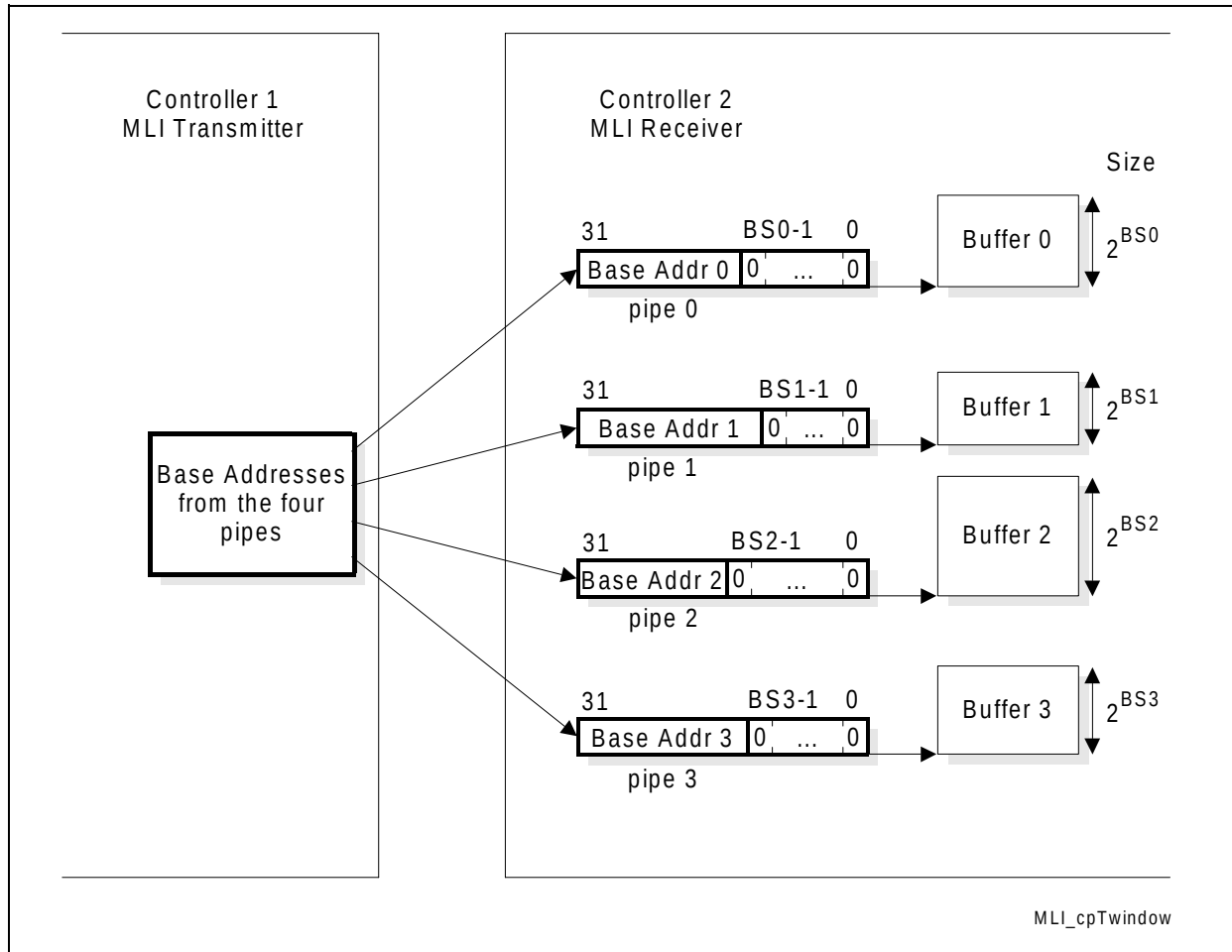


Figure 5-7 Transfer Window Base Address Copy

Note: BS_x is the buffer size of each of the different pipes (where $x = 0, 1, 2, 3$). The selected buffer size must not exceed the size of the targeted transfer window (8 KByte for small transfer windows)

Within the offset (its width in bits is the same as indicated in the buffer size), the MLI transmitter from the first controller sends a reference to the pipe in use. When the MLI receiver obtained this data it will know what is the absolute address by simply concatenating the offset to the base address of the pipe.

Note: A pipe should always be accessed by either its small transfer window or the corresponding large transfer window. Mixing accesses via both window types to the same pipe is possible but not recommended (optimized frames require a single transfer window type). The used transfer window type of a pipe can be different from those of the other pipes.

Figure 5-8 illustrates the address translation process from the first controller to the second one.

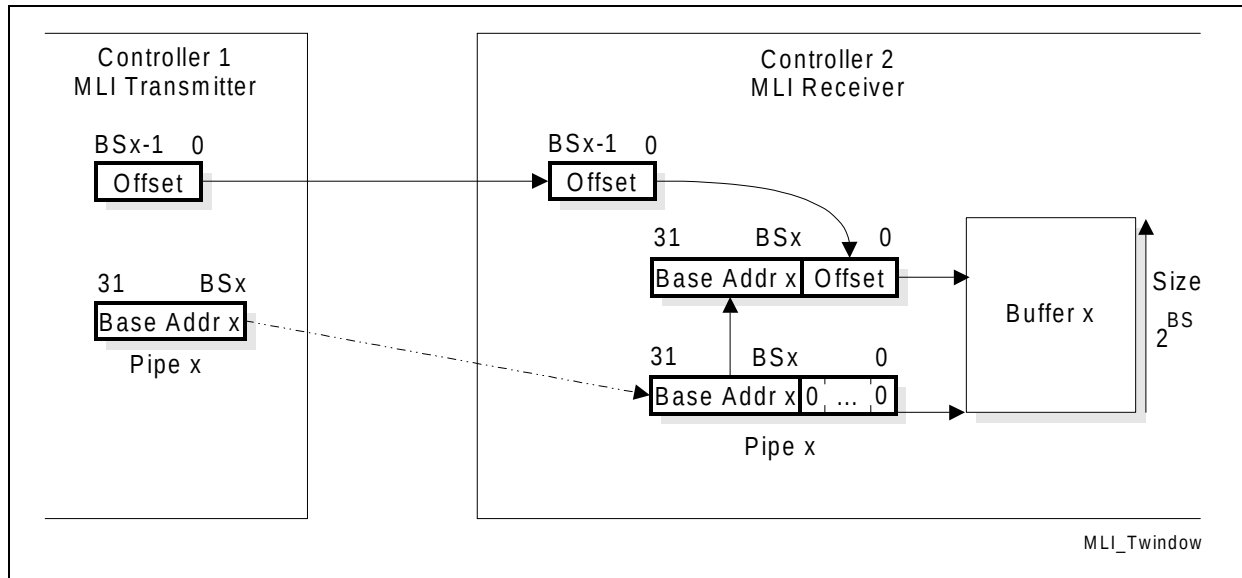


Figure 5-8 Address translation process

The kernel MLI includes an optimized mode to transfer data blocks. Whenever the MLI transmitter detects that the new address and the previous one follow a predictable scheme, it will send just the data reducing this way the bits to be transferred. This mode is based in a prediction method of the new address in the MLI receiver of the second controller. In fact the MLI receiver will automatically update the address, as explained in the next sections.

From the point of view of the connection, each MLI transmitter will have four READY, four VALID, one CLK and one DATA possible connections with the external world. This will provide the possibility of connecting each MLI transmitter with up to four MLI receivers in other controllers although the MLI transmitter will not have the possibility of transferring data to two or more different MLI receivers at the same time.

Each MLI receiver will have one READY, four VALID, four CLK and four DATA possible connections with the external world. With this scheme, each MLI receiver could be connected with up to four MLI transmitters in other controllers although the MLI receiver will not have the be possibility of receiving data from two or more different MLI transmitters at the same time.

Figure 5-9 illustrates an example of connection between two MLI.

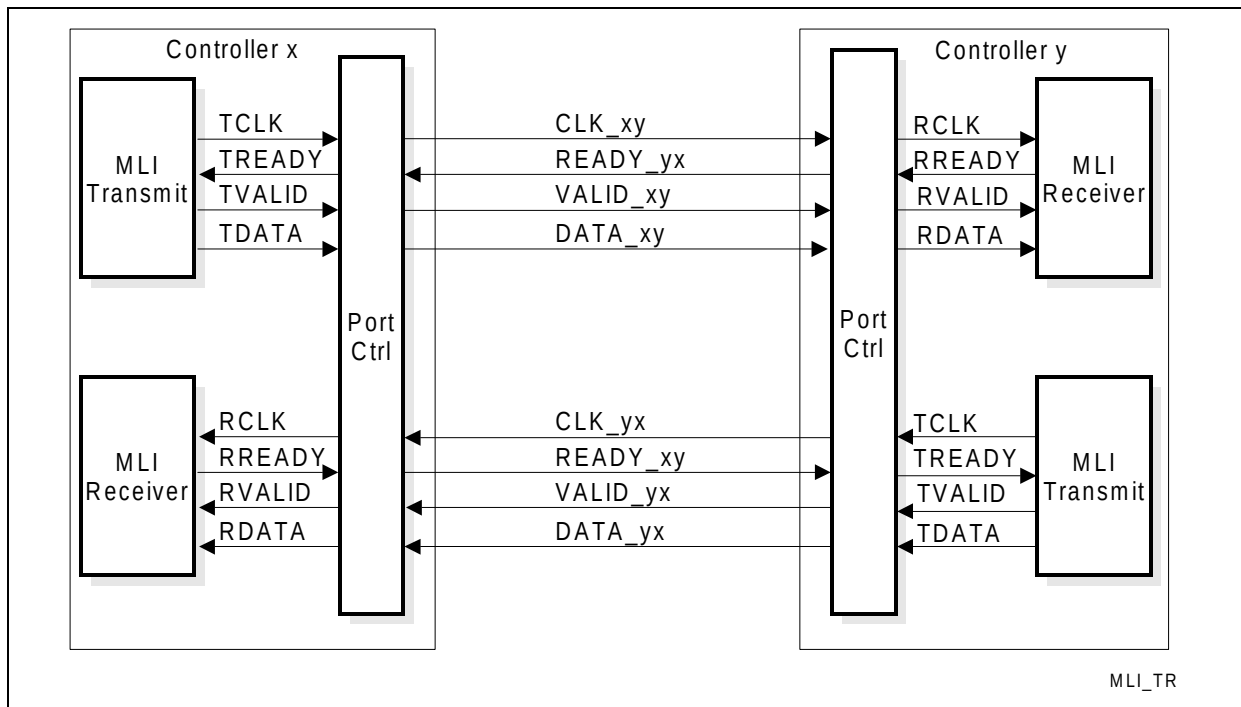


Figure 5-9 MLI Transmitter - Receiver Connection

Note: The suffixes x and y indicate the source and destination of the signals.

Figure 5-10 illustrates an example of connection between three different MLI.

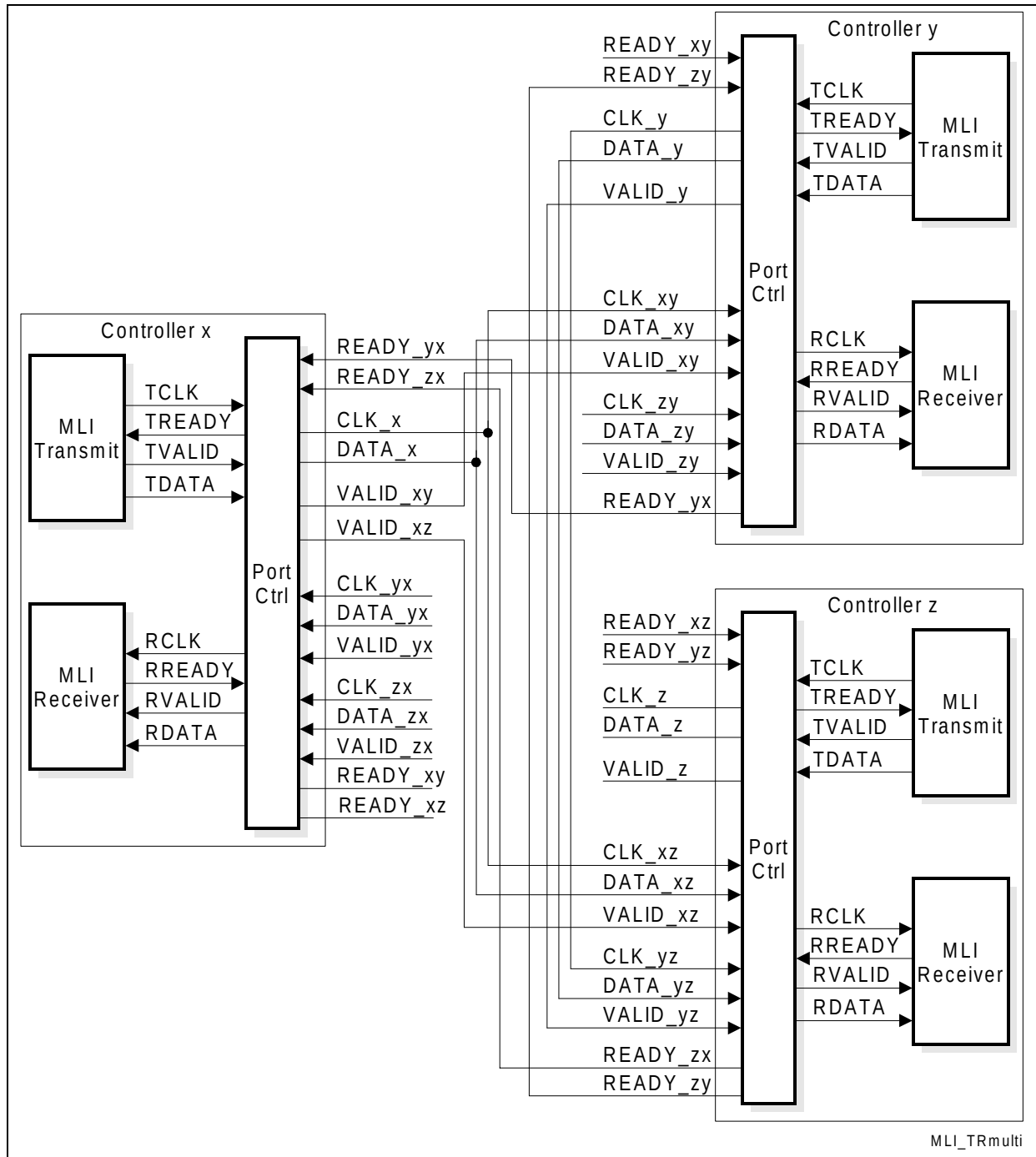


Figure 5-10 MLI Transmitter - Receiver Connection

Note: The suffixes x, y and z indicate the source and destination of the signals. For instance CLK_xz connects the clock signal CLK_x in the MLI transmitter x and finishes in the MLI receiver z. The signals selection is made by programming the OICR register.

5.1.4 Handshake Description

The transmission may start whenever the MLI receiver in the other controller is ready to receive data, i.e. READY signal is high. When the MLI transmitter from the first controller wants to start the transmission, it has to set the VALID signal high and it will hold it as long as it sends the data. When the VALID rising edge is detected by the MLI receiver in the other controller, it will set the READY signal to low level again.

When the transmission is finished, the MLI transmitter resets the VALID signal to zero and looks if the READY signal is low level again. This will indicate the MLI transmitter in the first controller that the MLI receiver has acknowledged the transmission. It will also reset the ready delay counter (TSTATR.RDC) and it will start counting TCLK clock periods. The counter will be stopped when the MLI transmitter detected that the READY signal is high level again or in the counting overflow.

If the MLI receiver sets the READY signal high again in a number of clock periods less than a programmed number (maximum delay for parity, TCR.MDP), this will indicate the MLI transmitter in the first controller that the MLI receiver is prepared for a new transmission and the previous one was received without parity error.

A detailed explanation about the handshake timing and clock domains in the MLI transmitter and the receiver may be found in [Section 5.1.11](#).

The handshaking in a situation without error is shown in [Figure 5-11](#).

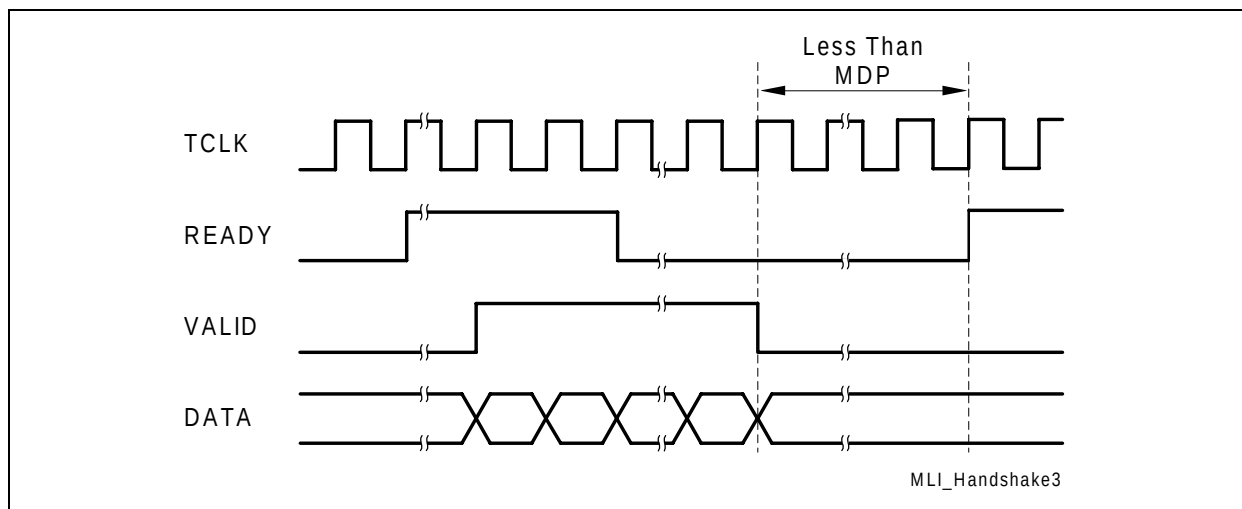


Figure 5-11 MLI Transmitter - Receiver Handshake in Transfer Without Error

Note: The signals are seen from the MLI transmitter in the first controller. When VALID is not asserted, the DATA line will have only a value (one or zero) depending on the programmed value in TCR.DNT and its chosen polarity.

The delay between the falling edge of VALID and the rising edge of READY is measured by the ready delay counter TSTATR.RDC. This value is compared to TCR.MDP on the transmitter side in order to detect when the receiver has signaled a parity error.

Figure 5-12 illustrates a situation of non-acknowledge. The Ready signal remains high when the VALID is set low level again.

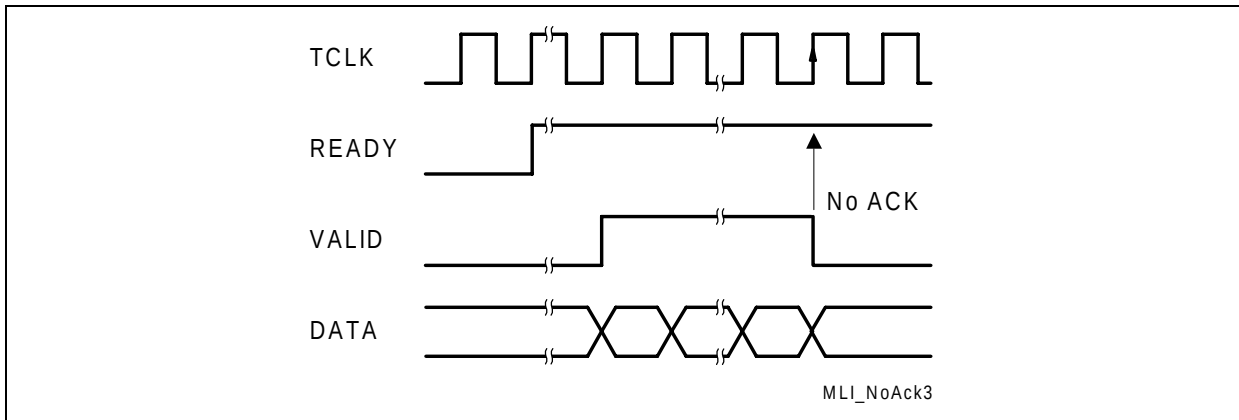


Figure 5-12 Non-Acknowledge Situation

Note: The signals are seen from the MLI transmitter in the first controller.

If the READY signal raises after a number of TCLK clock periods (measured by TSTATR.RDC) is greater than the maximum delay for parity programmed value (TCR.MDP), then this situation will be interpreted as parity error, as shown in **Figure 5-13**.

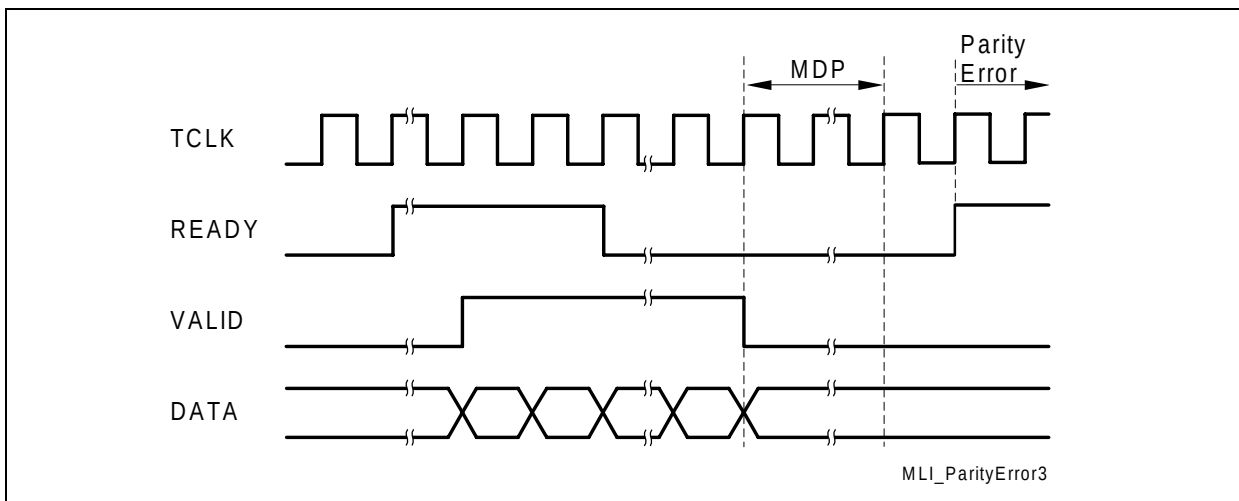


Figure 5-13 Parity Error Situation

Note: The signals are seen from the MLI transmitter in the first controller.

If a non-acknowledge persists, the MLI transmitter will keep on counting the number of non-acknowledge errors and if a maximum number is reached then a time-out error is produced. **Figure 5-14** illustrates this situation.

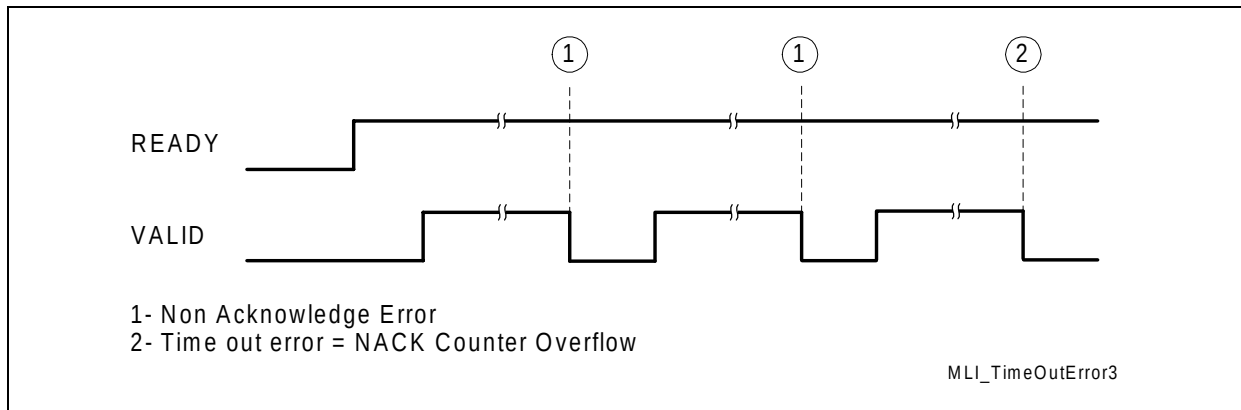


Figure 5-14 Time-out Error Situation

Note: The signals are seen from the MLI transmitter in the first controller.

All the error situations and the actions taken by both parts are explained in [Section 5.1.7.8](#) and [Section 5.1.8.6](#).

5.1.5 Startup Procedure

During the startup procedure of the MLI, an appropriate value for the maximum delay for parity signaling must be set up in the receiver and in the transmitter. Therefore, the following actions must be taken:

- The overall loop delay (total propagation delay between the transmitter and the receiver, including the output and the input driver delay, the line propagation and the synchronization time) must be measured.
- The appropriate MDP value must be programmed in the transmitter and then transferred to the receiver (automatic receiver setup by a command frame on pipe 1).
- Dummy frames with parity error and without parity error must be sent in order to check for a correct reply of the receiver.
- If the parity error signaling is working correctly, the setup is finished and normal frame traffic can be started.

For the measurement of the overall loop delay a dummy frame is sent to the receiver and the time is measured between the falling edge of the VALID signal and the rising edge of the READY signal (if the READY signal does not rise after a certain while, the receiver might be defect, not correctly connected or not powered).

The signaling of the READY signal on the receiver side takes place in the clock domain of the transmit clock. The reset value of the TCR.MDP is 0 in the transmitter and in the receiver and as a result, the READY signal will be set high immediately after receiving the parity bit by the receiver. In this case (MDP = 0), the signaling of READY is identical

for a parity error and for a correct frame. Therefore, the number of transmit clock cycles between VALID becoming 0 until READY becoming 1 represents the overall loop delay. It is indicated by the counter bit field TSTATR.RDC (ready delay count). This bit field is a counter starting from 0 each time VALID becomes low (1 to 0 transition) and that stops when READY becomes 1 (0 to 1 transition). It holds the value until the next 1 to 0 transition of VALID is reached. This value can be read out to determine an appropriate MDP value.

The desired value for MDP has to be transferred to the receiver. Therefore, the command pipe 1 can be used. A frame of command pipe 1 transfers the value and the receiver stores it automatically as its delay for parity error (RCR.DPE). In order to verify the correct setting of DPE of the receiver, dummy frames are transferred with and without parity error. The parity generation of the receiver always starts with 0 and this value is toggled each time a 1 is received. This result is compared to the received parity bit (the received parity bit does not modify the receivers parity check bit). The parity start bit of the transmitter can be programmed. Assuming a correct transfer, the start value of 1 in the transmitter, will lead to a parity error detection on receiver side. This event is signaled by the receiver by setting READY to 1 after the time indicated by RCR.DPE (in the receiver) has elapsed. By reading the value of TSTATR.RDC, the transmitter software can detect if the receiver's DPE has been set up correctly. In case of an error, the transfer of DPE by the command pipe 1 has to be started again, until the results are correct.

All these setup actions should take place while the MLI move engine of the receiver is switched off (automatic mode disabled, receiver in listen mode). The complete setup can be done under the control of the transmitter. A special software on receiver side is not required. If the required values are known on both sides (transmitting and receiving controller), the normal transfers can start.

Note: A dummy frame can be any frame that does not lead to a hardware action in the receiver. In listen mode, the CPU on receiver side should ignore the reception of dummy frames. In order to start normal operation, the transmitter can switch on the automatic mode of the receiver's move engine or send a command via pipe3 that is then taken into account by the receiver's CPU.

5.1.6 MLI Kernel and MLI Interface Logical Connection

The MLI transmitter will have the possibility of getting 32 address bits, 32 bits of data, a selection for read or write operation, and a selection of the transmission window. The MLI receiver may provide the MLI interface 32 bits of address, 32 bits of data and the type of the transaction (read or write). In addition there will be a signal to request the DMA from the MLI receiver.

Figure 5-15 shows a graphical representation of this interface.

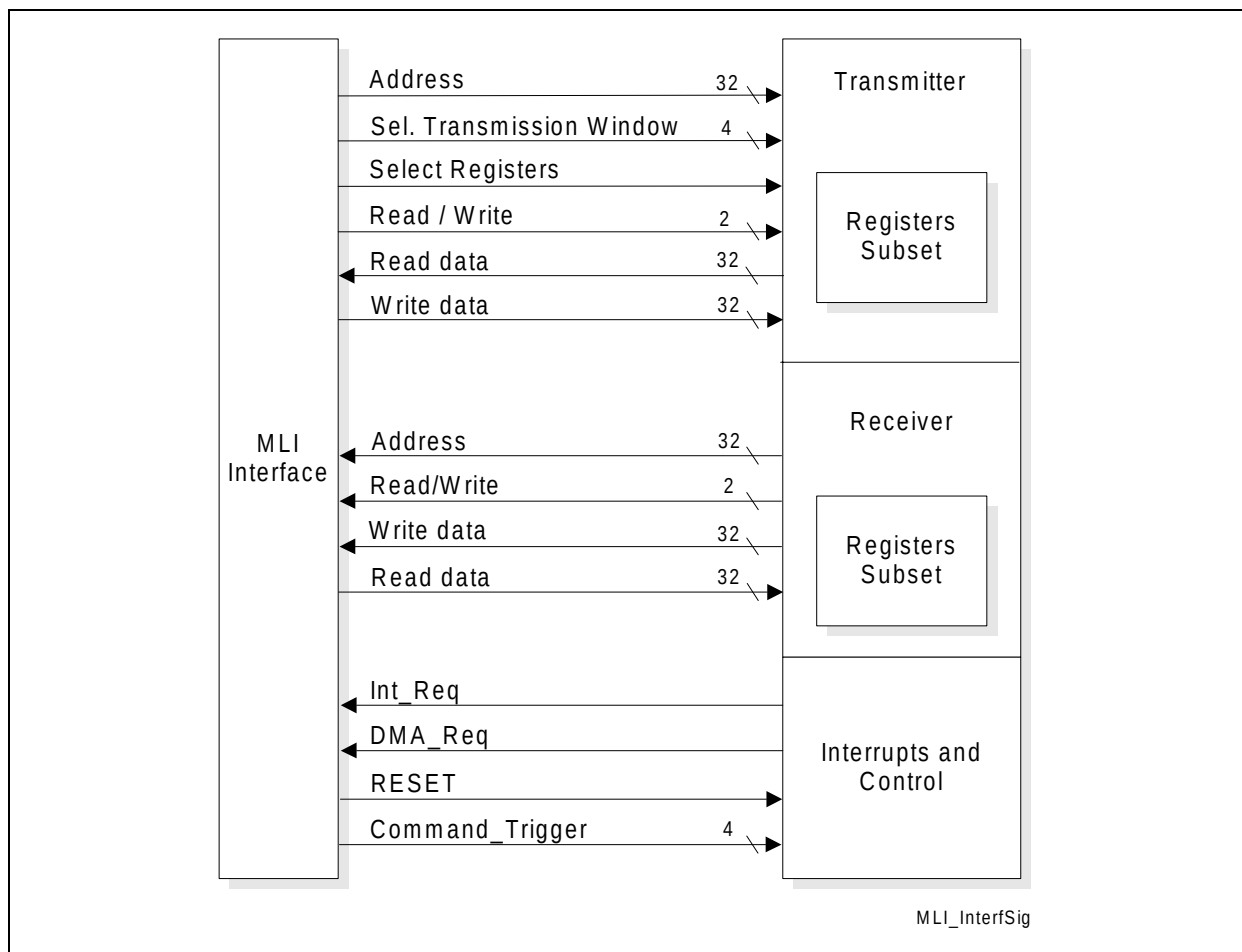


Figure 5-15 Signals Between MLI Interface and MLI Kernel

5.1.7 MLI Transmitter

5.1.7.1 MLI Transmitter Reset

After the hardware reset the MLI transmitter will be in transmitter off mode as it will be explained in [Section 5.1.7.2](#). In this state the transmitter will not raise its VALID signal and therefore no transaction will be performed.

5.1.7.2 MLI Transmitter Operation Modes

By programming the MLI transmitter control register it is possible to set its operation mode. In [Table 5-1](#) are shown all the possible modes of the MLI transmitter, depending on the values of the mode of operation parameter (TCR.MOD). The characteristics of the different modes are explained below.

Table 5-1 MLI Transmitter Operation Modes

TCR.MOD	Mode of Operation
0	MLI transmitter off
1	MLI transmitter on

Transmitter Off

This is the mode in which the MLI transmitter is after the hardware reset. It has the following characteristics:

- The VALID signal is not activated. As a consequence, no transfer is supported.

Transmitter On

Its characteristics are the following ones:

- The whole functionality of the MLI transmitter is available.

Note: The next section will explain the whole MLI transmitter functionality and its interfaces arbitration for the transmitter on operation mode. For each case, it must be taken into account the operation mode in which the MLI transmitter is programmed and consider the limitations introduced by each of the explained modes.

5.1.7.3 Internal Architecture and Interface Signals

Figure 5-16 illustrates the internal architecture of the MLI transmitter:

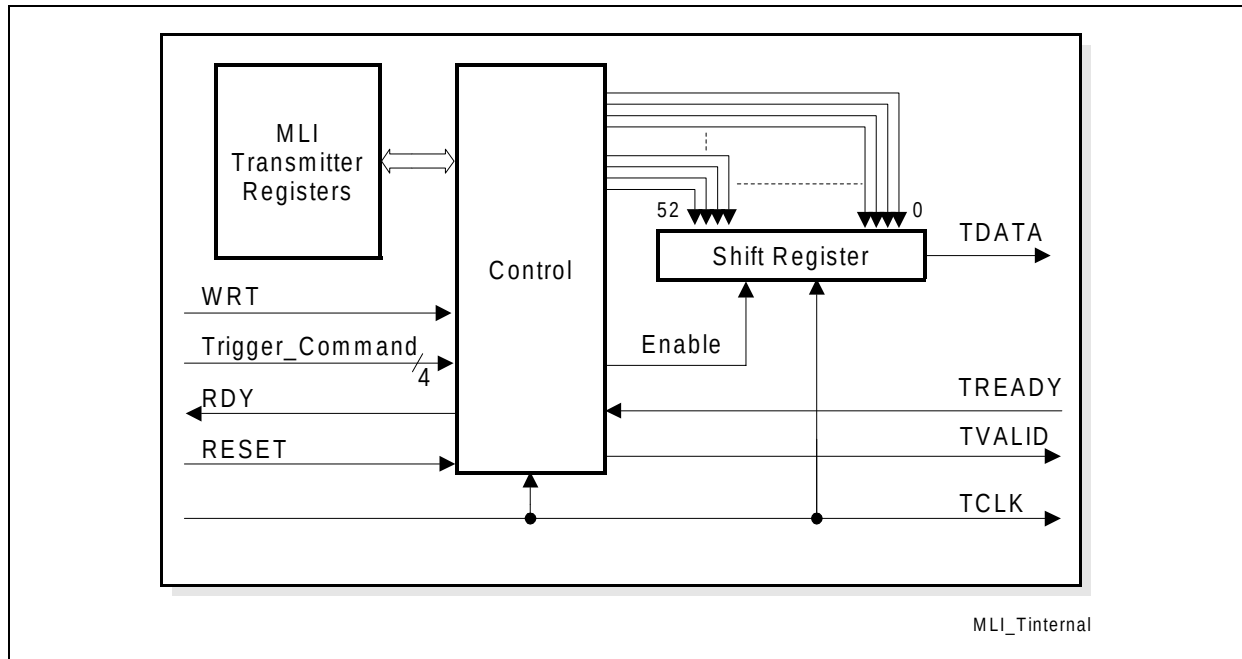


Figure 5-16 MLI Transmitter Internal Architecture

Note: The letter “x” indicates the pipe number, from 0 to 3.

The signal RDY is used to notify the controller that the MLI transmitter is prepared to receive new information in the data and address offset registers (TPxDATAR and TPxAOFR). The ‘active’ WRT signal indicates that a write access is performed in the MLI registers. The enable signal will control the shift register.

The four trigger command lines will be programmed via hardware the MLI transmitter to send up to four different commands.

The signals TDATA, TREADY, TVALID and TCLK will follow the scheme explained in [Section 5.1.4](#).

The MLI transmitter will operate as an information moving agent between two controllers. It will receive through its controller interface side, address offset and data, only address offset or only data. These values will be written in TPxDATAR and TPxAOFR registers by software via the DMA switch, or by the DMA itself.

The MLI transmitter will keep track of the next information:

- Current address offset in the pipe (TPxAOFR, where x denotes the pipe number)
- Width of the current address offset in each of the pipe's address offset registers (TPxSTATR.BS, where x denotes the pipe number)
- Current data in the pipe (TPxDATAR, where x denotes the pipe number)

- Width of the current data received in each of the pipe's data registers (TPxSTATR.DW, where x denotes the pipe number)

A complete list of the MLI transmitter registers may be found in [Section 5.2.1](#).

5.1.7.4 Transmission Format

The MLI transmitter first transmits four bits as header (denoted as H) that contains information about the mode of transfer and which is the current pipe in use.

The first two bits in the header will determine what type of transmission it is and it will be referenced as frame code (denoted as FC in the whole chapter). These two bits together with the number of bits of the frame will determine the type of transmission. [Table 5-2](#) shows its encoding.

Table 5-2 First Two Bits Header Encoding. Frame Code

Header bits 0 and 1 (FC)	Type of Frame
00 _B	Copy base address frame
01 _B	Write in offset and data frame or Discrete read frame
10 _B	Command frame or Answer frame
11 _B	Optimized write frame or Optimized read frame

The second two bits field, indicate which is the current pipe in use. These two bits will be referenced as pipe number (PN) in the whole chapter. [Table 5-3](#) shows its encoding.

Table 5-3 Second Two Bits Header Encoding. Pipe Number

Header bits 2 and 3, (PN)	Pipe in Use
00 _B	Pipe 0
01 _B	Pipe 1
10 _B	Pipe 2
11 _B	Pipe 3

After this header the MLI transmitter sends the information as it will be explained in [Section 5.1.7.5](#). In every case a parity bit will be transmitted in the last position of the frame (p). [Figure 5-17](#) illustrates parts of the transmission frame.

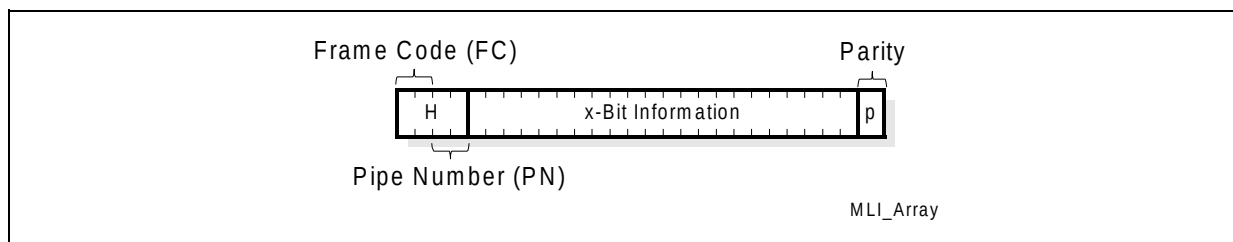


Figure 5-17 Parts of the Transmission Frame

5.1.7.5 Transmission Modes

In this section are explained the different modes in transmission, indicating for each one:

- the frame composition
- where are the values of each field of the frame taken from
- the number of bits that each frame has

Section 5.1.7.6 will explain how the MLI transmitter chooses between these different options of transmission.

Copy Base Address Frame

Its frame code is 00_B. This mode will allow the MLI receiver in the other controller to know the base addresses and the buffer size of each transfer window. **Figure 5-18** illustrates the frame sent by the MLI transmitter.

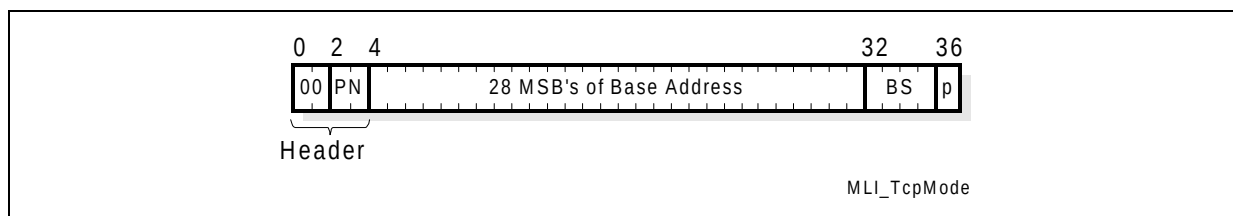


Figure 5-18 Copy Base Address Frame

The frame contains the 28 more significant bits of the base address and the buffer size of this transfer window (the 4 bits denoted as BS). **Table 5-4** illustrates where each of the fields of the frame are taken from.

Table 5-4 Storage of the Values Used in the Frame

Field	Value Taken From
PN	TRSTATR.PN
Base Address	28 MSBs of TCBAR
Buffer Size	TPxSTATR.BS

Note: x indicates the pipe number; x = 0, 1, 2, 3.

The number of bits transmitted in this mode is shown in [Table 5-5](#).

Table 5-5 Number of Bits In Copy Base Address Frame

Header	Base Address	Parity	Total
4 bits	32 bits	1 bit	37 bits

Command Frame

Its frame code is 10_B. The MLI transmitter sends the following frame:

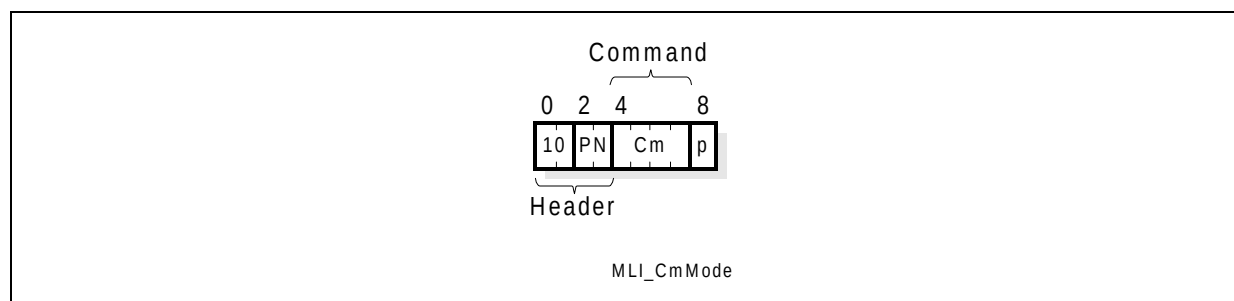


Figure 5-19 Command Frame

The bits denoted as “Cm” in [Figure 5-19](#) represent the command. Its value is obtained from the command bit field contained in the MLI transmitter command register (TCMDR.CMDPx, where x denotes the pipe number). The meaning of the command depends on the pipe used.

The command is a subset of four bits, which encoding is shown in [Table 5-22](#). The actions indicated are performed by the MLI receiver of the second controller. In this table, DPE stands for delay for parity error (stored in RCR.DPE). It indicates the number of TCLK clock periods that the MLI receiver must wait without raising again the READY signal to inform that a parity error was detected.

[Table 5-6](#) illustrates where each of the fields of the frame are taken from.

Table 5-6 Storage of the Values Used in the Frame

Field	Value Taken From
PN	The correspondent to the accessed part of the command register
CM	TCMDR.CMDPx

Note: x indicates the pipe number; x = 0, 1, 2, 3.

The number of bits transmitted is described in [Table 5-7](#).

Table 5-7 Bits Transmitted in Command Frame

Header	Command	Parity	Total
4 bits	4 bits	1 bit	9 bits

Write Access To Transfer Window:

- Write in Offset and Data Frame: the address offset cannot be predicted.

Its frame code is 01_B. The MLI transmitter sends the offset and data frame. **Figure 5-20** illustrates this transfer. The number of bits of the address offset (m) and the bits of data (n) are known and specified in the transmitter status register of the current pipe.

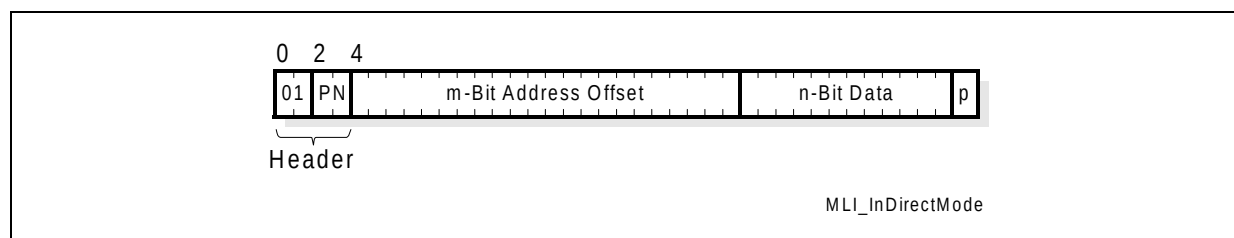


Figure 5-20 Write Access in Offset and Data Frame

Table 5-8 illustrates where each of the fields of the frame are taken from.

Table 5-8 Storage of the Values Used in the Frame

Field	Value Taken From
PN	The correspondent to the accessed registers
Address Offset	TPxAOFR
Data	TPxDATAR

Note: x indicates the pipe number; x = 0, 1, 2, 3.

The number of bits transmitted is shown in **Table 5-9**.

Table 5-9 Number of Bits In Offset And Data Frame

Data Width	Header	Offset	Data	Parity	Total
8 bits	4 bits	m bits	8 bits	1 bit	13+m bits
16 bits	4 bits	m bits	16 bits	1 bit	21+m bits
32 bits	4 bits	m bits	32 bits	1 bit	37+m bits

- Optimized Write Frame: the new address offset can be predicted.

Its frame code is 11_B. **Figure 5-21** shows this transmission mode.

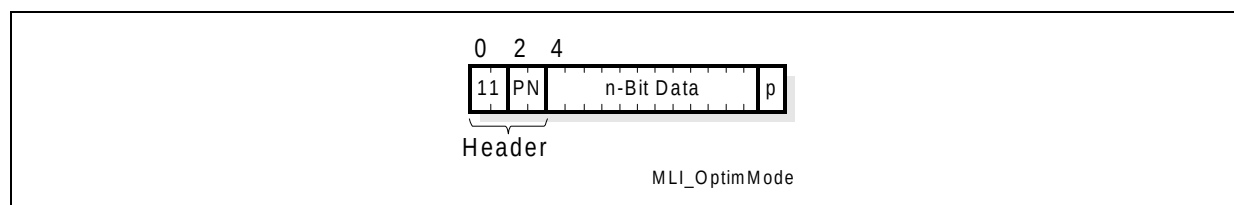


Figure 5-21 Optimized Write Frame

The offset will be deducted in the MLI receiver of the second controller depending on the address prediction factor associated to the current pipe, TPxSTATR.AP in the transmitter side and RPxSTATR.AP in the receiver side.

Table 5-10 illustrates where each of the fields of the frame are taken from.

Table 5-10 Storage of the Values Used in the Frame

Field	Value Taken From
PN	The correspondent to the accessed registers
Data	TPxDATAR

Note: x indicates the pipe number; x = 0, 1, 2, 3.

The number of bits transmitted is shown in **Table 5-11**.

Table 5-11 Number of Bits In Optimized Write Frame

Data Width	Header	Data	Parity	Total
8 bits	4 bits	8 bits	1 bit	13 bits
16 bits	4 bits	16 bits	1 bit	21 bits
32 bits	4 bits	32 bits	1 bit	37 bits

Read Access To Transfer Window

- Discrete Read Frame: the new offset address cannot be predicted.

Its frame code is 01_B. The MLI transmitter sends the offset it wants to read from, indicating the width of the data. The frame in this case is shown in **Figure 5-22**.

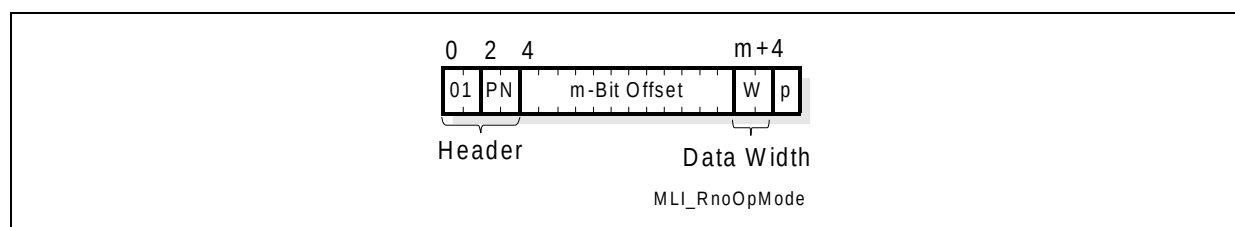


Figure 5-22 Discrete Read Frame

The field referenced as “W” (data width) in **Figure 5-22** indicates the size of data that the first controller wants to read.

Table 5-12 shows the encoding of this field.

Table 5-12 Data Width Encoding

W	Data Width
00 _B	8 bits
01 _B	16 bits

Table 5-12 Data Width Encoding (cont'd)

W	Data Width
10 _B	32 bits
11 _B	Reserved

Table 5-13 illustrates where each of the fields of the frame are taken from.

Table 5-13 Storage of the Values Used in the Frame

Field	Value Taken From
PN	The correspondent to the accessed registers
W	TPxSTATR.DW
Address Offset	TPxAOFR

Note: x indicates the pipe number; x = 0, 1, 2, 3.

The number of bits transmitted is shown in **Table 5-14**.

Table 5-14 Number of Bits In Discrete Read Frame

Header	Data Width	Offset	Parity	Total
4 bits	2 bits	m bits	1 bit	7+m bits

Note: This case is perfectly distinguishable from the write in offset and data frame because the value of the buffer size (TPxSTATR.BS = m, where x indicates the current pipe) is fixed all the time for each pipe.

- Optimized Read Frame: the new address offset can be predicted.

Its frame code is 11_B. The frame in this case is shown in **Figure 5-23**.

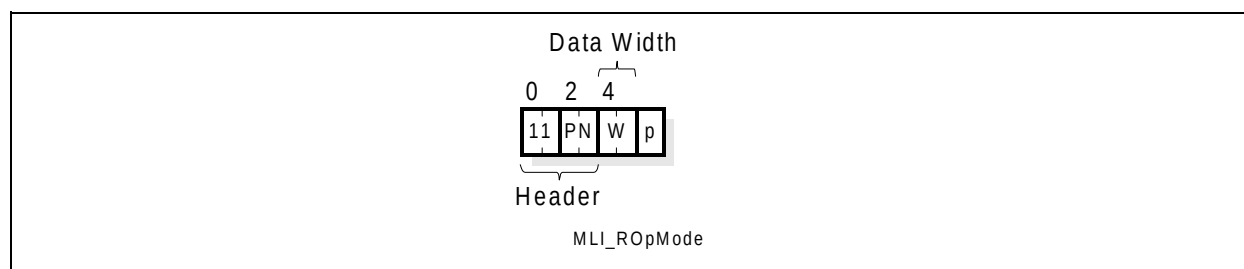


Figure 5-23 Optimized Read Frame

The bit field data width (W) has the same meaning as explained in **Table 5-12**.

Table 5-15 illustrates where each of the fields of the frame are taken from.

Table 5-15 Storage of the Values Used in the Frame

Field	Value Taken From
PN	The correspondent to the accessed registers
W	TPxSTATR.DW

Note: x indicates the pipe number; x = 0, 1, 2, 3.

The number of bits transmitted is shown in **Table 5-16**.

Table 5-16 Number of Bits In Optimized Read Frame

Header	Data Width	Parity	Total
4 bits	2 bits	1 bit	7 bits

Answer Frame

Its frame code is 10_B. **Figure 5-24** illustrates the frame sent by the MLI transmitter.

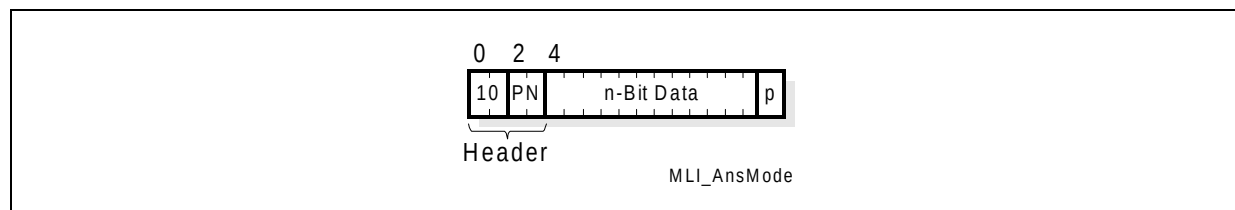


Figure 5-24 Answer Frame

The frame contains a data bit field that is the answer to a read operation made before. **Table 5-17** illustrates where each of the fields of the frame are taken from.

Table 5-17 Storage of the Values Used in the Frame

Field	Value Taken From
PN	TSTATR.APN
Data	TDRAR

The answer frame may be sent through any pipe because in any moment there will be only one read operation in course.

The number of bits transmitted is shown in [Table 5-18](#).

Table 5-18 Number of Bits In Answer Frame

Data	Header	Parity	Total
8 bits	4 bits	1 bit	13 bits
16 bits	4 bits	1 bit	21 bits
32 bits	4 bits	1 bit	37 bits

5.1.7.6 Transfer Mode Selection

Each time a write access to the MLI registers is made, a related flag is set in the transmitter status register (TRSTATR), indicating that the data stored in them has not been sent (transfer pending) and no more writing accesses may be made on these registers. These flags are reset again when the registers may be written again.

[Table 5-19](#) shows the flags that are set when accessing the different registers and when they are reset again.

Table 5-19 Valid Flags

Flag	Set When Access to	Reset When
TRSTATR.DVx	TPxAOFR, TPxDATAR for read or write operation	The write or read frame has been sent correctly through the correspondent pipe
TRSTATR.RPx	TPxAOFR, TPxDATAR for read operation	The answer frame has been received correctly through the correspondent pipe
TRSTATR.CIVx	The MLI transmitter detects a rising edge in the trigger_commandx line	The correspondent command frame has been sent correctly through the pipe 0
TRSTATR.CVx	TCMDR or CVx is set via software	The command frame has been sent correctly through the correspondent pipe
TRSTATR.BAV	TCBAR	The copy base address frame has been sent correctly through the correspondent pipe
TRSTATR.AV	TDRAR	The answer frame has been sent correctly through the correspondent pipe

Note: x indicates the pipe number; x = 0, 1, 2, 3.

When a rising edge is detected in the trigger_commandx line, the MLI transmitter will set the correspondent CIVx bit, and it will send a command frame through pipe 0 as explained in [Table 5-20](#).

Table 5-20 Hardware Triggered Command

Rising Edge	Command to Send Through Pipe 0
Trigger_command0	0001 _B
Trigger_command1	0010 _B
Trigger_command2	0011 _B
Trigger_command3	0100 _B

When more than one transfer is pending, the criteria to choose between the different frames to send will be based on a priority scheme as follows:

- Read answer frame, command frame triggered by hardware, command frame triggered by software, read frame, write frame and copy base address frame.
- If two or more pipes operations are pending with the same priority, then the one with the highest priority will be chosen. The pipe number 0 has the highest priority, and then pipe 1, 2 and 3 by this order.

The following paragraphs illustrate the different actuations taken depending on the type of frame chosen to be sent:

Copy Base Address Frame

The TRSTATR.BAV flag is set to one, and there is no other type of frame pending in this pipe. This transfer means that the controller wants to initialize the corresponding pipe base address of the MLI receiver in the other controller and its correspondent buffer size. The MLI transmitter will send the copy of base address frame with the base address stored in the TCBAR register and the buffer size stored in the correspondent TPxBAR register.

Write and Read Frame

- If only the TRSTATR.DVx flag is set, it indicates a write operation.
- If TRSTATR.DVx and TRSTATR.RPx flags are set, they indicate a read operation.

Depending on the operation code this will mean that the first controller wants to write the data (stored in TPxDATAR) in the relative address indicated by the address offset (that will be stored in TPxAOFR) of the transfer window defined by the selected pipe or it wants to read data from that position.

If the address prediction method is allowed (TCR.NO = 0, where x indicates the pipe) then the MLI transmitter will compare the new address offset written in the bus with the old one (TPxAOFR in the moment of accessing). The difference between these two

addresses is stored in the correspondent pipe status register (TPxSTATR.AP) if it is not greater than 10 bits (in two's complement). If the difference is the same in two consecutive transfers then the MLI transmitter will set the optimized flag in the status register of the correspondent pipe (TPxSTATR.OP). This will indicate that the optimized mode will be used to send this frame when it was chosen to be sent. The optimized mode will be used for the rest of the frames, as far as TPxSTATR.OP = 1, until the difference was not the same as the one stored in the status register, (then the MLI resets TPxSTATR.OP again).

When the conditions explained above are not met, then the transfer is performed using a normal frame for writing or reading (discrete read frame or write in address offset and data frame).

When the necessary comparisons to infer the address prediction factor, the new address offset is finally stored in the address offset register of the pipe (TPxAOFR).

Next pseudocode illustrates the process:

```

if (TPxSTATR.NO = 0) then                                     -- Optimized mode possible
    delta = new_address_offset - TPxAOFR
    if (delta = TPxSTATR.AP) then
        if (Coincidence = TRUE) then
            TPxSTAT.OP = 1
        else
            Coincidence = TRUE
            TPxSTAT.OP = 0
        end if
    else if (if difference delta is not bigger than 9 bits) then
        TPxSTATR.AP = delta
        Coincidence = FALSE
        TPxSTAT.OP = 0
    else
        Coincidence = FALSE
        TPxSTAT.OP = 0
    end if
else                                                         -- Optimized mode disabled
    TPxSTAT.OP = 0
end if

```

Note: TPxAOFR contains the last address offset used in the pipe. TPxSTATR.AP is the address prediction factor. The boolean variable Coincident, expresses the condition of two consecutive coincidences in the address offsets.

When the data of the pipe was chosen to be sent, using the priority method explained before, the MLI will know whether the frame should or not be sent using the optimized mode.

When the write or read frame is correctly received by the other controller, the MLI resets the correspondent TRSTATR.DVx flag. In the case of a read operation, the flag

TRSTATR.RPx will be reset when the answer is received and read. Until this read pending flag is reset again, the registers TPxDATAR and TPxAOFR of the pipe may be written again.

Note: The users who want to use the optimized read/write frames must not use the small and large MLI transfer windows in a single pipe at the same time.

Answer Data Frame

This means that the written data in the TDRAR register is the answer to a read operation, therefore the transmission will be made in answer frame.

When the writing frame is correctly received by the other controller, the MLI resets the TRSTATR.AV flag.

Command Frame

If CIVx is set, the MLI transmitter will send the command frame correspondent to the line through the pipe 0 as explain in [Table 5-20](#). When the command frame is correctly received by the other controller, the MLI resets the correspondent TRSTATR.CIVx flag.

If CVx is set, the MLI transmitter will check the value stored in the command bit field correspondent to the pipe of the TCMDR register and it will send it using the command frame. When the command frame is correctly received by the other controller, the MLI resets the correspondent TRSTATR.CVx flag.

5.1.7.7 Parity Generation

The type of parity used is even or odd parity, depending on the programmed value in TCR.TP. The parity bit is calculated by toggling a bit each time that a one is sent in the frame. The starting value for the toggling bit is zero if the parity is even and is one if the parity is odd. Assuming a correct transmission, a starting value of 1 in the toggling bit will lead into a parity error situation as explained in [Section 5.1.5](#).

5.1.7.8 Error Detection and Handling

The MLI transmitter will be able to recognize the following error situations in the transmission:

- a non-acknowledged transfer
- a parity error

Non-Acknowledge Error Detection

This situation is explained in [Figure 5-12](#) and in [Figure 5-40](#). When this error is detected, the MLI transmitter will set the non-acknowledge error flag (TSTATR.NAE) and decreases the counter of non-acknowledge errors (TCR.MNAE). When this counter reaches the value zero, a time-out interrupt is generated if enabled.

Parity Error Detection

The Parity Error situation is explained in [Figure 5-13](#) and in [Figure 5-39](#). When this error is detected, the MLI transmitter will set the parity error flag (TSTATR.PE) and decreases the counter of parity errors (TCR.MPE). When this counter reaches the value zero, a parity error interrupt is generated if enabled.

Whenever the MLI transmitter sends correctly a new frame, it will produce an interrupt. The interrupt is enabled depending on the type of frame (command frame interrupt or normal frame interrupt). Please refer to [Section 5.1.10](#).

5.1.7.9 MLI Transmitter Input/Output Control

[Figure 5-25](#) shows the control structure for the transmitter output signals VALID, CLK and DATA. The VALID signal can be distributed to up to four output lines (TVALIDA to TVALIDD). It is possible to individually enable/disable each line (except DATA) and to select its polarity.

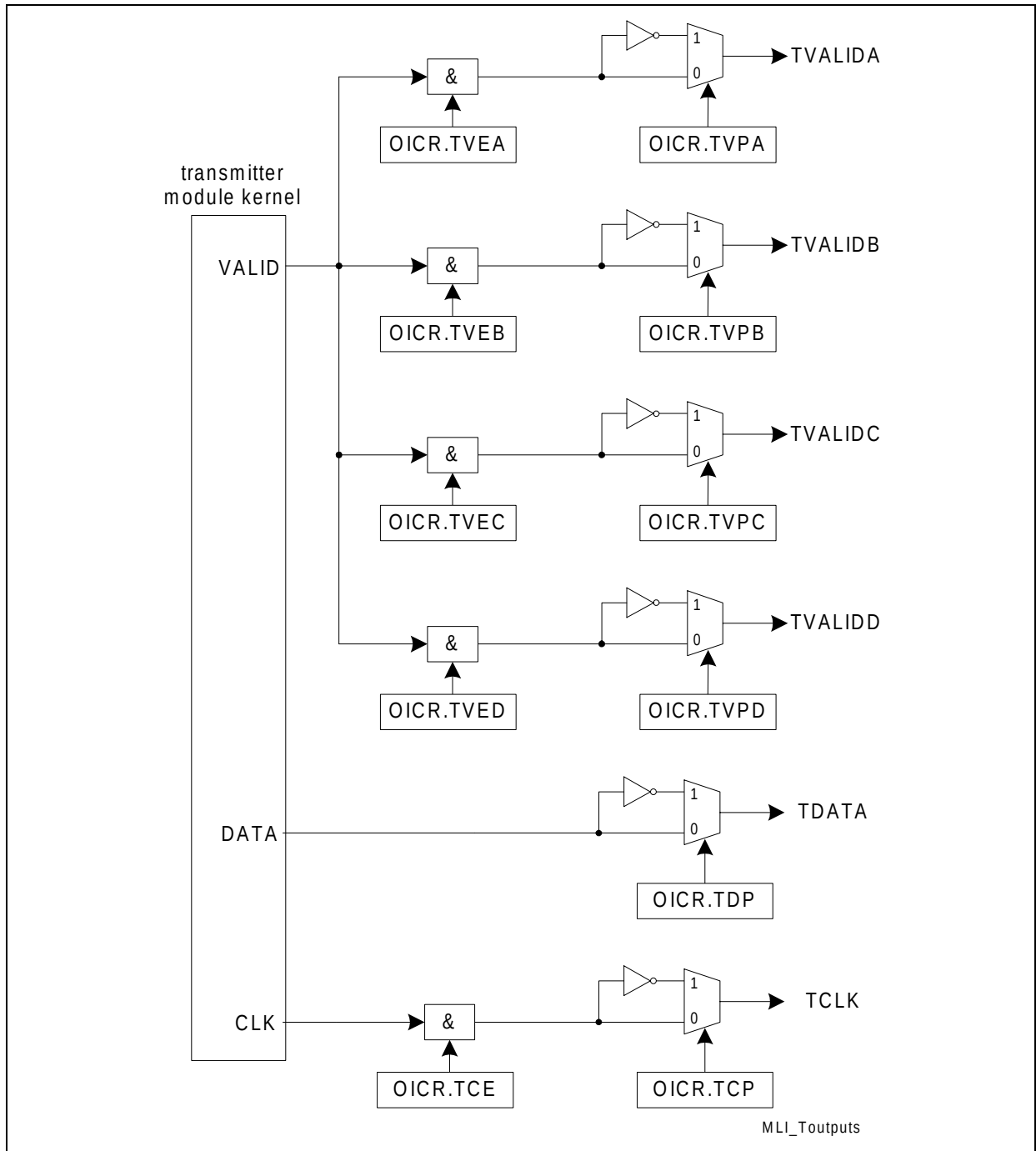


Figure 5-25 Control of the Transmitter Output Signals

The transmitter output shift clock signal CLK can be enabled by the bit OICR.TCE and its output polarity can be selected by OICR.TCP. For the data signal DATA it is possible to select the polarity by programming bit OICR.TDP.

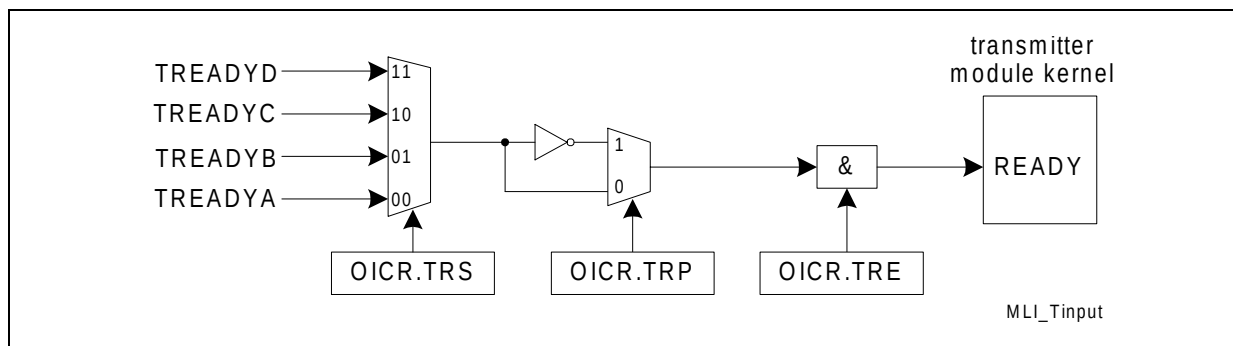


Figure 5-26 Control of the Transmitter Input Signal

The transmitter input signal READY can be selected from four possible input signals (TREADYA to TREADYD). The selected input signal can be enabled by bit OICR.TRE and its polarity can be chosen by OICR.TRP.

Note: The first letter "T" of the signal names indicates that these signals belong to the transceiver part of an MLI module. The last letter "A" to "D" of a signal, belonging to a set of lines, indicates that the signal can be selected from (input) or can be distributed to (output) up to 4 lines.

5.1.8 MLI Receiver

5.1.8.1 MLI Receiver Reset

After the hardware reset, the MLI receiver will be in receiver off mode as explained in [Section 5.1.8.2](#).

5.1.8.2 MLI Receiver Operation Modes

By programming the MLI receiver control register it is possible to set its operation mode. [Table 5-21](#) shows all the possible modes of the MLI receiver, depending on the values of the mode parameter (RCR.MOD). The characteristics of the different modes are explained below.

Table 5-21 MLI Receiver Operation Modes

RCR.MOD	Mode of Operation
0	MLI move engine off, receiver off
1	MLI move engine in automatic mode

Move Engine Off

- The Ready signal is activated and the acknowledge and parity error conditions are accomplished.
- All the write or read transfers have no active effect on the FPI bus in the MLI receiver side.
- Modification of the control parameters or the base addresses of the transmission pipes are allowed.
- Commands are taken into account. Possibility of interrupts programming.

Move Engine in Automatic Mode

- The Ready signal is activated and the acknowledge and parity error conditions are accomplished.
- All the data transfers are taken into account.
- Modification of the control parameters or the base addresses of the transmission pipes are allowed.

This bit may be modified by the MLI receiver whenever it receives the proper command.

Note: The next section will explain the whole MLI receiver functionality and its interfaces arbitration for the receiver on operation mode. For each case, it must be taken into account the operation mode in which the MLI receiver is programmed and consider the limitations introduced by each of the explained modes.

5.1.8.3 Internal Architecture and Interface Signals

Figure 5-27 shows the internal architecture of the MLI receiver:

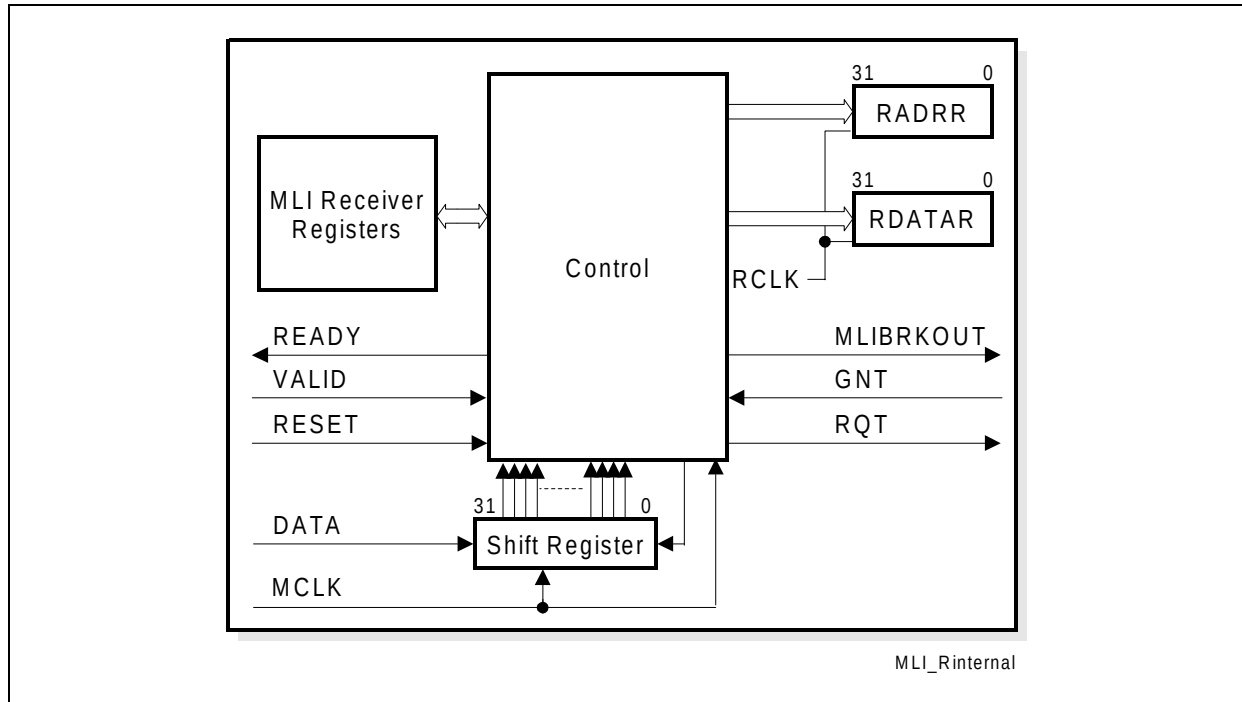


Figure 5-27 MLI Receiver Internal Architecture

Note: The letter “x” indicates the pipe number; from 0 to 3.

The signal RQT is used to notify the DMA that the MLI receiver has new data in its registers RADRR and RDATAR. When the DMA reads the registers RADRR and RDATAR, the MLI interface sets the signal GNT high. From the MLI interface point of view, the signals are explained in [Section 5.1.4](#).

Each register RPxBAR (where x indicates the pipe) contains a default value for the base address of each of the four pipes. Each time the MLI receiver obtains a new frame, it will recognize the kind of transmission and the pipe from the header and from the number of received bits. The MLI receiver will keep track of the next parameters:

- Width of the current data received in the pipe (RPxSTATR.DW, where x denotes the pipe number)
- Width of each transfer window (RPxSTATR.BS, where x denotes the pipe number)

A complete list of the MLI receiver registers may be found in [Section 5.2.2](#).

5.1.8.4 MLI Receiver Operation

The MLI receiver will obtain the information sent from the MLI transmitter of the other controller. It will operate on the information received in order to extract each of the bit fields. Finally the data and address registers (RDATAR, RADRR) are updated with a new value. Due to the fact that all the data is received synchronized with the clock of the MLI transmitter from the other controller, the MLI receiver must synchronize the address and the data values with its internal clock (RCLK) and the registers RDATAR and RADRR contain their values.

Its operation depends on the type of frame received. A complete reference of different modes is described in [Section 5.1.7.5](#).

In order to accomplish the information split, the MLI receiver will take into account the number of bits of the received frame, the transmission mode and the buffer size for the current pipe.

Copy Base Address Frame

Its description and the number of bits of this frame are shown in [Figure 5-18](#) and in [Table 5-5](#) respectively.

After the header, the MLI receiver obtains the 28 MSBs of the current pipe base address, which is stored in the 28 MSBs of its corresponding pipe base address register (RPxBAR). The buffer size is stored in the status register (RPxSTATR.BS) and the type of frame bit field is updated (RCR.TF = 00_B). A normal frame received interrupt is produced if it is enabled by RIER.NFRIE.

Command Frame

Its description and the number of bits of this frame are shown in [Figure 5-19](#) and in [Table 5-7](#) respectively. After the complete reception of this frame, the MLI receiver splits the information in its different bit fields.

[Table 5-22](#) illustrates the different values that may be received in the command frame and the action taken for each one.

Table 5-22 Command Frame Encoding

PN	Cm	Action
00 _B	0001 _B	Generate interrupt 0, if enabled by RIER.ICE
	0010 _B	Generate interrupt 1, if enabled by RIER.ICE
	0011 _B	Generate interrupt 2, if enabled by RIER.ICE
	0100 _B	Generate interrupt 3, if enabled by RIER.ICE
	Others	No effect

Table 5-22 Command Frame Encoding (cont'd)

PN	Cm	Action
01 _B	0000 _B	Set RCR.DPE = 0000 _B
	0001 _B	Set RCR.DPE = 0001 _B
	0010 _B	Set RCR.DPE = 0010 _B
	0011 _B	Set RCR.DPE = 0011 _B
	...	...
	1111 _B	Set RCR.DPE = 1111 _B
10 _B	0001 _B	Set RCR.MOD = 1, enable automatic mode
	0010 _B	Set RCR.MOD = 0, disable automatic mode (listen mode)
	0100 _B	Reset TRSTATR.RP0
	0101 _B	Reset TRSTATR.RP1
	0110 _B	Reset TRSTATR.RP2
	0111 _B	Reset TRSTATR.RP3
	1111 _B	Generate a pulse on line $\overline{\text{MLIBRKOUT}}$ (if enabled by RCR.BEN)
	others	No effect
11 _B	Any	Command meaning interpreted by software

When the MLI receiver gets the command 1111_B in pipe 2 (PN = 10_B), then it asserts the signal MLIBRKOUT, which is active in low level if enabled by RCR.BEN bit. **Figure 5-28** illustrates this procedure.

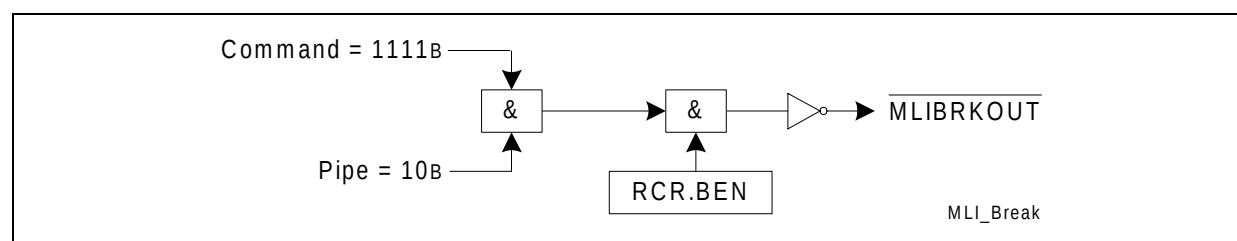


Figure 5-28 Assertion of MLIBRKOUT Signal

Write Access In Offset and Data Frame

Its description and the number of bits of this frame are shown in [Figure 5-20](#) and in [Table 5-9](#) respectively.

After the header, the MLI receiver obtains the m bits corresponding to the offset. The MLI receiver knows how many bits corresponding to the address offset because it is the same as the buffer size of the current pipe (RPxSTATR.BS).

In order to follow the same address prediction method that is carried out by the MLI transmitter, the MLI receiver will compare the address offset of the currently received frame with the address offset previously received. The last address offset is in the receiver base address register of the pipe (RPxSTATR.BS/ LSB's of RPxBAR, where $x = 0, 1, 2, 3$ indicates the pipe). If the difference between both addresses is less than 9 bits, the MLI transmitter will store it in the address prediction factor bit field (RPxSTATR.AP), and this value will be used to obtain the address whenever an optimized frame was received. After this comparison, the newly received address offset is stored in the lowest part of RPxBAR.

[Figure 5-29](#) illustrates this process.

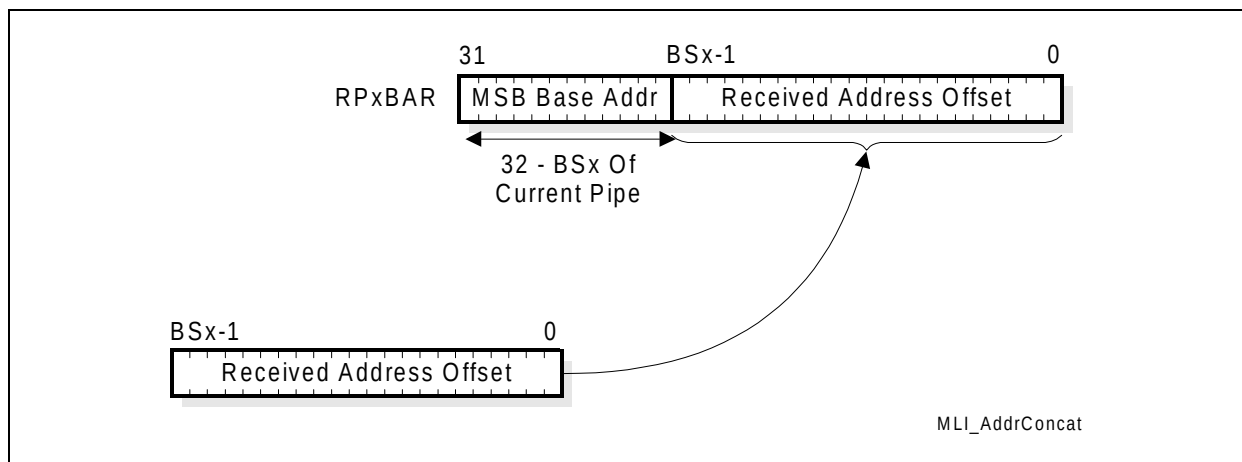


Figure 5-29 Absolute Address Obtaining

Note: BS stands for buffer size (in bits) of the current pipe (x).

After the concatenation, this absolute address will be stored in the MLI receiver address register RADRR. The next subset of bits corresponds to the data and the parity bit. The data will be stored in the RDATAR. The RCR.DW and RCR.TF bit fields are updated, RCR.TF = 10. A normal frame received interrupt is produced if enabled by RIER.NFRIE.

Table 5-23 shows the place of storage of the information obtained from the frame.

Table 5-23 Place of Storage for Address Offset and Data Bits

m - Bit Address Offset	n-Bits Data
BS least-significant bits of RPxBAR. After that, copy 32 bits of the register RPxBAR in RADRR	RDATAR

Optimized Write Frame

Its description and the number of bits of this frame are shown in **Figure 5-21** and in **Table 5-11** respectively.

After the header, the MLI receiver obtains n bits of data that will be stored in the MLI receiver data register (RDATAR).

The address offset must be calculated by adding to the last address offset used (the content of RPxBAR register) the address prediction factor (RPxSTATR.AP, 10 bits with one sign bit).

Figure 5-30 illustrates how to obtain the address offset.

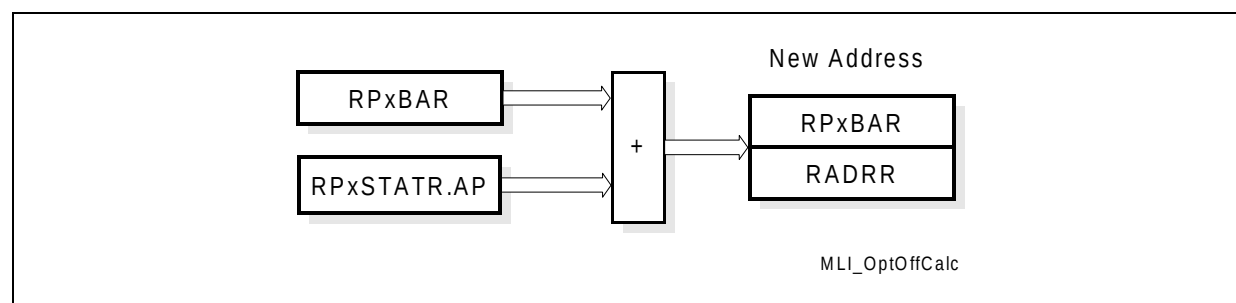


Figure 5-30 Address Offset Prediction for Optimized Write Frame

If the increment or decrement of the address offset results in overflow or underflow, then the wraparound method is used to fit the new address to the transfer window.

The newly obtained 32-bit address is stored in RPxBAR and RADRR registers. The RCR.DW and RCR.TF bit fields are updated, RCR.TF = 10. A normal frame received interrupt is produced if it is enabled by RIER.NFRIE.

Discrete Read Frame

Its description and the number of bits of this frame are shown in [Figure 5-22](#) and in [Table 5-14](#) respectively.

After the header, the MLI receiver obtains two bits, which indicate the width of the data that must be read. The next m bits (as $RPxSTATR.BS$) represent the offset from which this data will be read from.

In order to follow the same address prediction method that is carried out by the MLI transmitter, the MLI receiver will compare the address offset of the currently received frame with the address offset previously received. This last address offset is in the receiver base address register of the pipe ($RPxSTATR.BS$ / LSB's of $RPxBAR$, where $x = 0, 1, 2, 3$ indicates the pipe). If the difference between both addresses is less than 9 bits, the MLI transmitter will store it in the address prediction bit field ($RPxSTATR.AP$), and this value will be used whenever an optimized frame is received.

The absolute address is calculated by concatenating the obtained address offset with the base address of the current pipe as explained in [Figure 5-29](#) and then it is stored in the register $RADRR$. The $RCR.DW$ and $RCR.TF$ bit fields are updated, $RCR.TF = 01$. A normal frame received interrupt is produced if it is enabled by $RIER.NFRIE$.

[Table 5-24](#) shows where each of the received bit fields is stored.

Table 5-24 Place of Storage

Data Width	$RPxSTATR.BS$ bits of Address Offset
$RPxSTATR.DW$	Concatenate with the $(32 - RPxSTATR.BS)$ more significant bits of $RPxBAR$. Copy this new value of $RPxBAR$ in $RADRR$.

Optimized Read Frame

Its description and the number of bits of this frame are shown in [Figure 5-23](#) and in [Table 5-16](#) respectively.

After the header, the MLI receiver obtains two bits, which indicate the width of the data that must be read.

The absolute address must be calculated by adding to the last address used ($RPxBAR$) the address prediction factor ($RPxSTATR.AP$, 10 bits with one sign bit) as explained in [Figure 5-30](#). If the increment or decrement of the address offset results in overflow or underflow, then the wraparound method is used to fit the new address to the transfer window.

The 32-bit resulting absolute address will be stored in the $RADRR$ and $RPxBAR$ registers. The $RCR.DW$ and $RCR.TF$ bit fields are updated, $RCR.TF = 10$. A normal frame received interrupt is produced if it is enabled by $RIER.NFRIE$.

Answer Frame

Its description and the number of bits of this frame are shown in [Figure 5-24](#) and in [Table 5-18](#) respectively. If an answer frame is received and the next condition is not met:

- $TRSTATR.RPx = 1$ and $TRSTATR.DVx = 0$, where x is the pipe from which the frame is received,

then the frame is discarded and a discarded read answer interrupt is produced if it is enabled by $RIER.DRAIE$.

After the header, the MLI receiver obtains n bits of data that will be stored in the $RDATAR$ register. The data is the answer to a read operation started by the MLI transmitter of the same MLI. The $RCR.DW$ and $RCR.TF$ bit fields are updated, $RCR.TF = 11$.

The MLI receiver will then produce a normal frame received interrupt controlled by $RIER.NFRIE$. When the register $RDATAR$ is read, the MLI resets the read pending flag ($TRSTATR.RPx$) again.

5.1.8.5 Access Protection

An access protection is implemented in the MLI receiver. It prevents undesirable read or write access to parts of the memory map in the receiving controller. The register AER will enable the read and write rights to parts of the memory map, and the register ARR will define the range of memory for parts of the memory.

5.1.8.6 Error Handling

The parity bit is checked using the same method as explained in [Section 5.1.7.7](#). If the MLI receiver detects a parity error in the transmission it will raise the READY signal following the next conditions:

- if the MLI receiver is prepared to receive a new frame before the delay that indicates the parity error programmable value (RCR.DPE) clock periods, then it will wait until RCR.DPE clock periods had passed. This DPE value will be received with a command frame from the MLI transmitter in the other controller.
- if the MLI is not prepared before RCR.DPE clock periods had passed, then it will raise the READY signal as soon as it is prepared to receive a new frame.

This will inform the transmitter in the other controller that a parity error is detected. If the parity error is signaled, the MLI receiver sets the parity error flag (RCR.PE) and decreases the counter of parity errors (RCR.MPE). When this counter reaches the value zero, a parity error interrupt is generated if it is enabled.

Figure 5-31 illustrates how the MLI receiver informs the MLI transmitter a parity error.

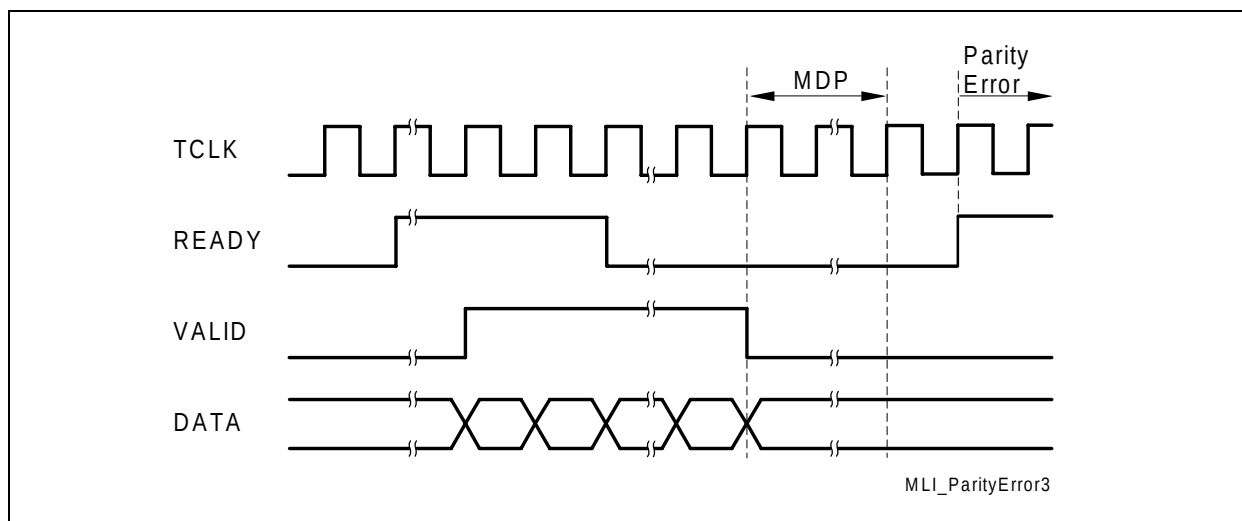


Figure 5-31 Parity Error

Note: The signals are seen from the MLI transmitter side. When VALID is not asserted the DATA line will have only a value (one or zero) depending on the programmed value in TCR.DNT and its chosen polarity.

5.1.8.7 MLI Receiver Input/Output Control

Figure 5-32 shows the control structure for the receiver input signals VALID, CLK and DATA. Each input signal to the receiver module kernel can be selected from up to four input lines. It is possible to individually enable/disable each line (except DATA) and to select its polarity.

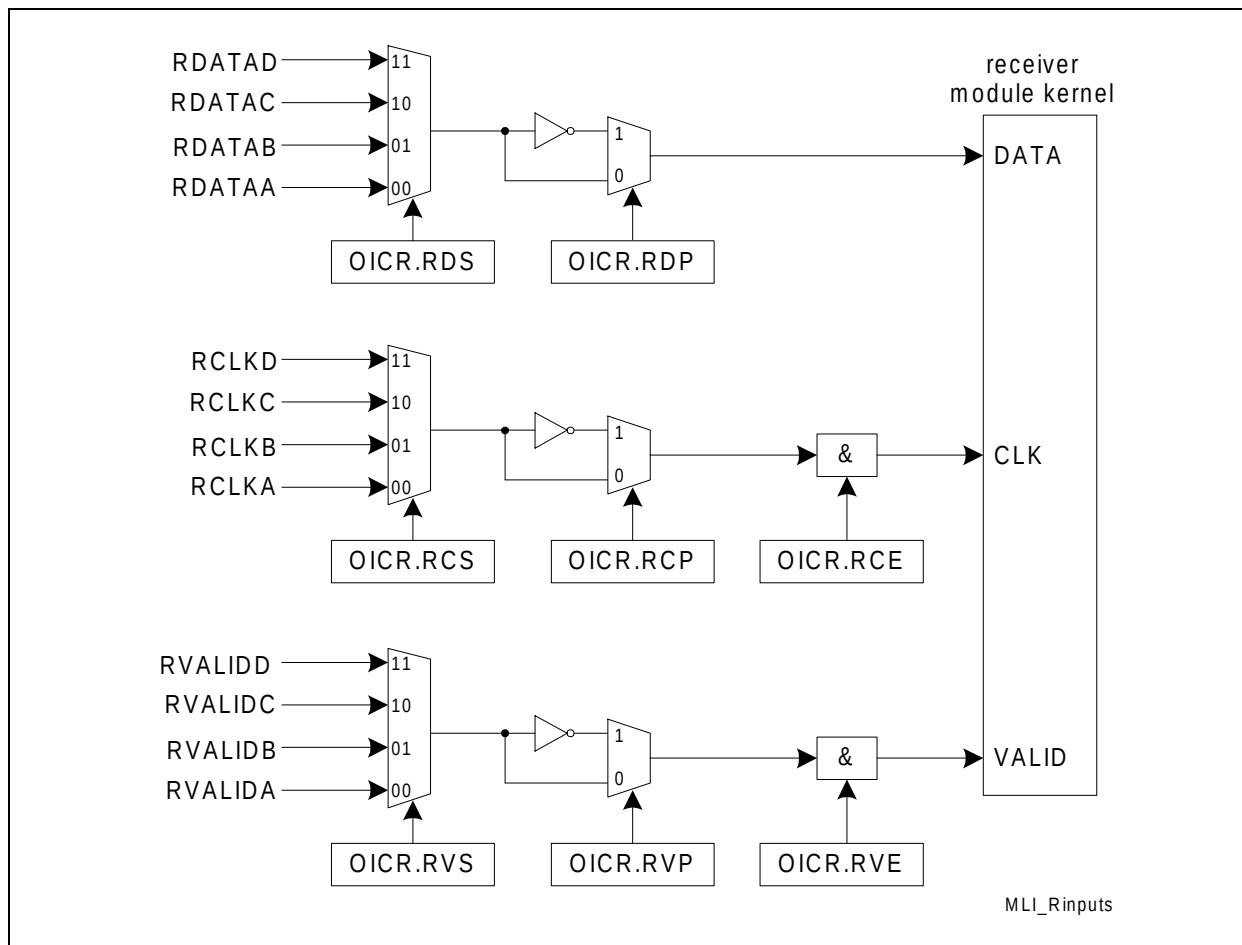


Figure 5-32 Control of Receiver Input Signals

Note: The first letter “R” of the signal names indicates that these signals belong to the receiver part of an MLI module. The last letter “A” to “D” of a signal, belonging to a set of lines, indicates that the signal can be selected from (input) or can be distributed to (output) up to 4 lines.

The receiver output signal READY can be distributed to up to four output lines (RREADYA to READYD). It is possible to individually enable/disable each line and to select its polarity. See [Figure 5-33](#).

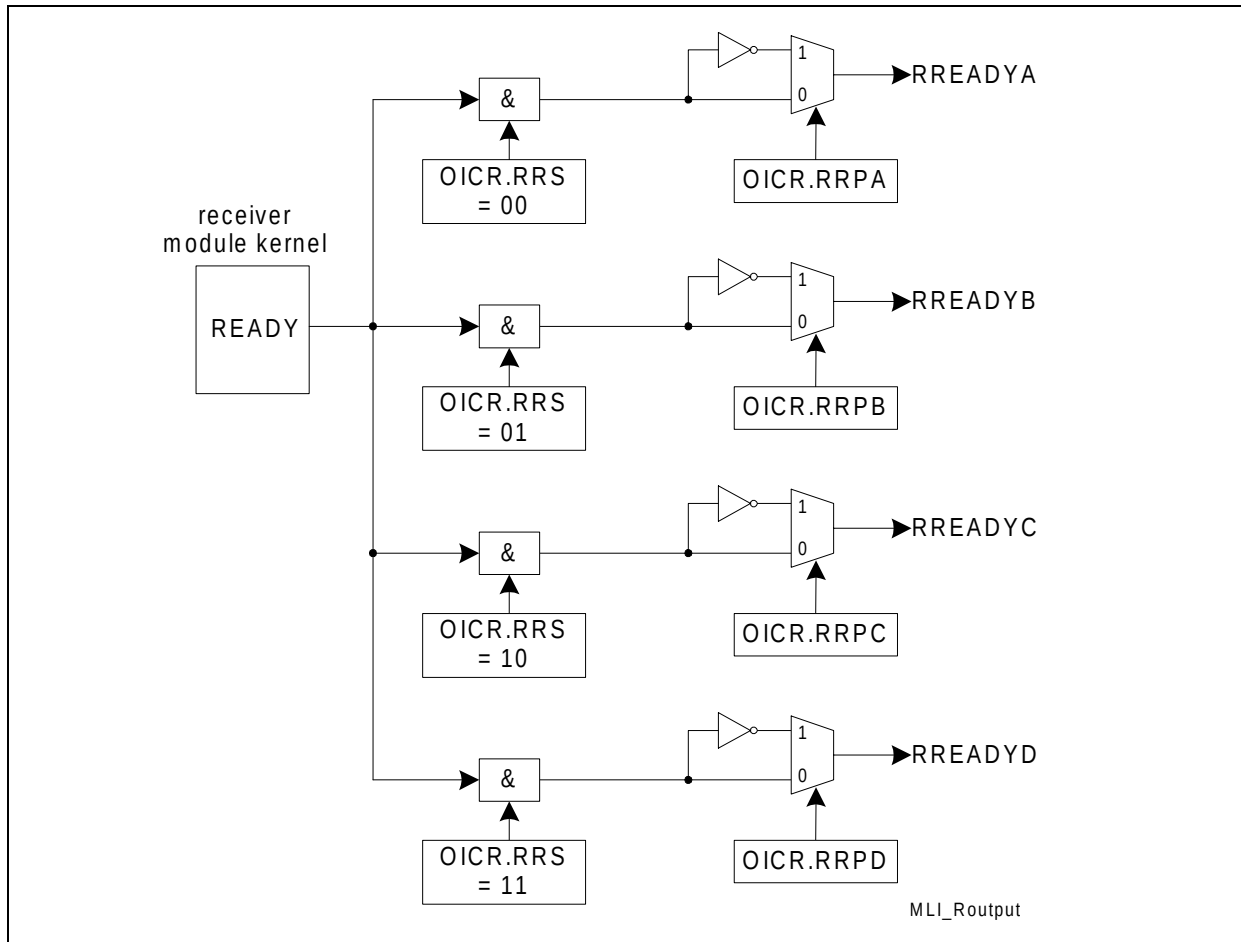


Figure 5-33 Control of Receiver Output Signal

For the CLK and DATA lines, it is possible to choose which of them will be active with the OICR.RCS and OICR.RDS bit fields. Their polarities can be set by programming the bits OICR.RCP and OICR.RDP.

5.1.9 Reading Process Summary

Figure 5-34 illustrates the whole reading process, since it is originated by the first controller, until it gets the answer.

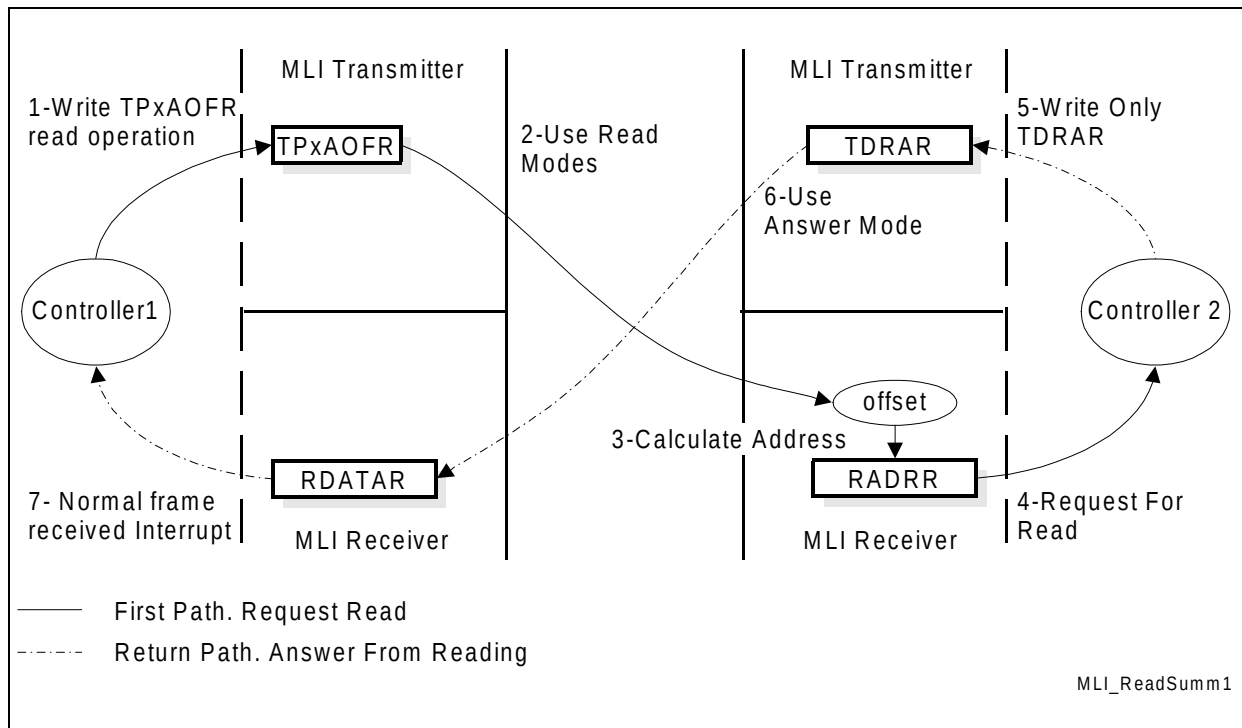


Figure 5-34 Full Read Operation Process

In step 2, the MLI transmitter of controller 1 sets TRSTATR.RPx and TRSTATR.DVx equal to one. Then it sends the read frame (optimized or not). When this frame is properly received, the MLI transmitter resets again the TRSTATR.DVx flag.

In step 3, the MLI receiver of controller 2 calculates the address and writes TSTATR.APN with the value received in the read frame. In step 6, the MLI transmitter of controller 2 sends an answer frame using the pipe number indicated in TSTATR.APN.

In step 7, the MLI receiver of the controller 1 gets an answer frame. If the read pending flag of the pipe is reset from which the answer is received or its DVx is not 0, then the frame is discarded and the MLI receiver produces an interrupt if it is enabled by CIR.DRAIE. If the RPx is set and DVx is reset, the MLI receiver produces a normal frame received interrupt (if enabled by CIR.NFRIE) to inform its CPU that it has the answer.

The software may read the data width and the address offset from the TPxSTATR TPxAOFR registers (the pipe is indicated by RCR.PN). When the register RDATAR is finally read, the MLI of controller 1 resets again the TRSTATR.RPx flag.

5.1.10 MLI Interrupts

The general interrupt structure is shown in [Figure 5-35](#). The interrupt event can trigger the interrupt generation and sets the corresponding bit in the status register. The interrupt pulse is generated independently from the interrupt flag in the interrupt status register. The interrupt flag can be reset by software.

If the interrupt is enabled by the related interrupt enable bit in the interrupt enable register, an interrupt pulse can be generated at one of the interrupt output lines INT_Ox of the module. If more than one interrupt source are connected to the same interrupt node pointer (in the interrupt node pointer register), the requests are combined to one common line.

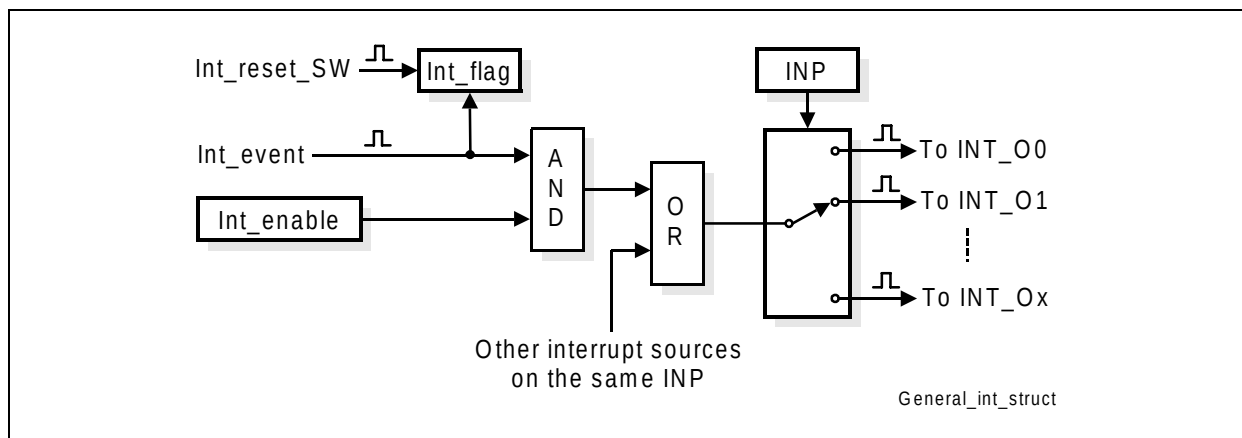


Figure 5-35 General Interrupt Structure

The request compressor condenses the MLI interrupt request sources to 8 interrupt outputs, reporting the interrupt requests of the MLI module to the interrupt controller. Each request source is provided with an interrupt output pointer, selecting the interrupt output to start the associated service routine to increase flexibility in interrupt processing. Each of the 8 interrupt outputs can trigger an independent routine with its own interrupt vector and its own priority.

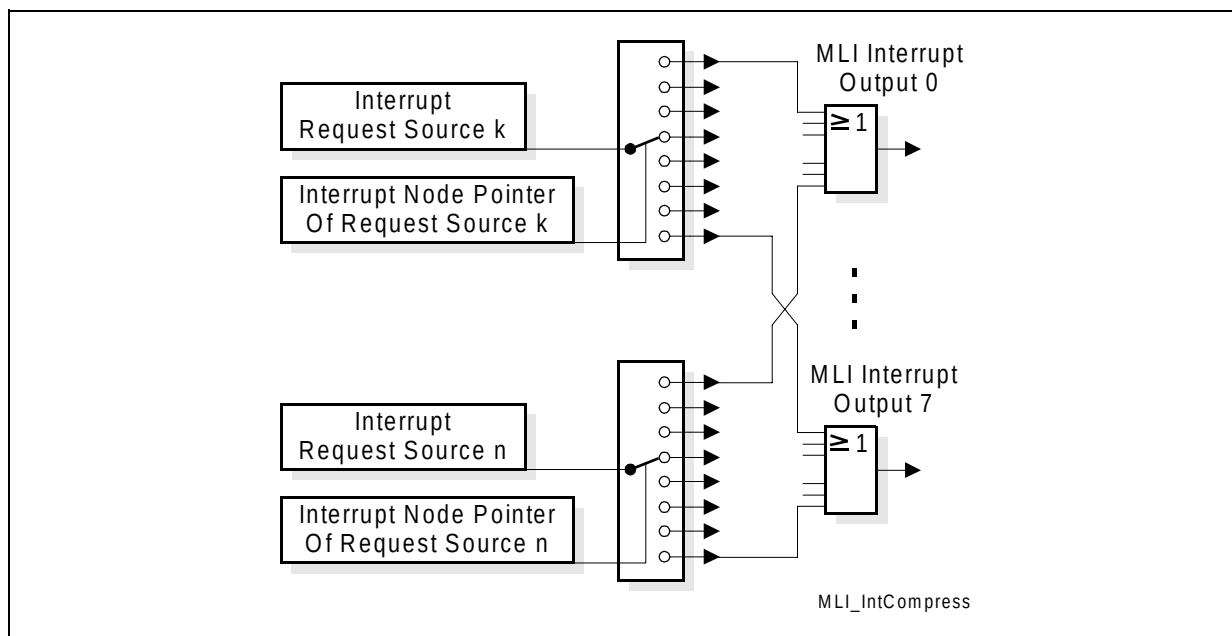


Figure 5-36 Interrupt Output Pointer and Interrupt Request Compressor

Table 5-25 shows the MLI interrupts.

Table 5-25 MLI Interrupts

Description and Condition	Agent that Generates it	Controlled by
Parity error interrupt. Produced when a programmable maximum number of parity errors is reached.	MLI transmitter	TINPR.PTEIP TIER.PEIE TIER.PEIR
Time-out error interrupt. Generated when a programmable maximum number of non-acknowledge errors is reached.	MLI transmitter	TINPR.PTEIP TIER.TEIE TIER.TEIR
Normal frame sent interrupt. MLI transmitter has sent a normal frame (not command) through pipe number x.	MLI transmitter	TINPR.NFSIPx TIER.NFSIEx TIER.NFSIRx
Command frame sent interrupt. MLI transmitter has sent a command frame through pipe number x, (only pipes 0, 1, 2 and 3).	MLI transmitter	TINPR.CFSIP TIER.CFSIEx TIER.CFSIRx
Discarded read answer received interrupt. Produced whenever an answer frame is received and the RPx flag of its correspondent pipe is 0, or RPx and DVx are 1.	MLI receiver	RINPR.DRAIP RIER.DRAIE RIER.DRAIR

Table 5-25 MLI Interrupts (cont'd)

Description and Condition	Agent that Generates it	Controlled by
Memory protection or parity error interrupt. Produced when detected a non-allowed read or write access or when a programmable maximum number of parity errors is reached.	MLI receiver	RINPR.MPPEIP RIER.MPEIE RIER.PEIE RIER.MPEIR RIER.PEIR
Normal frame received interrupt. The MLI receiver has obtained a normal frame (not command).	MLI receiver	RINPR.NFRIP RIER.NFRIE RIER.NFRIR
Command frame received interrupt. The MLI receiver has obtained a command frame through pipe number x.	MLI receiver	RINPR.CFRIP RIER.CFRIEx RIER.CFRIRx

5.1.11 Clock Domains and Handshake Timing

Figure 5-37 illustrates how the signals are synchronized in the MLI transmitter and in the receiver. In the figure, the suffixes T1 and R2 indicate from which side the signals are seen (from the MLI1 transmitter or from the MLI2 receiver).

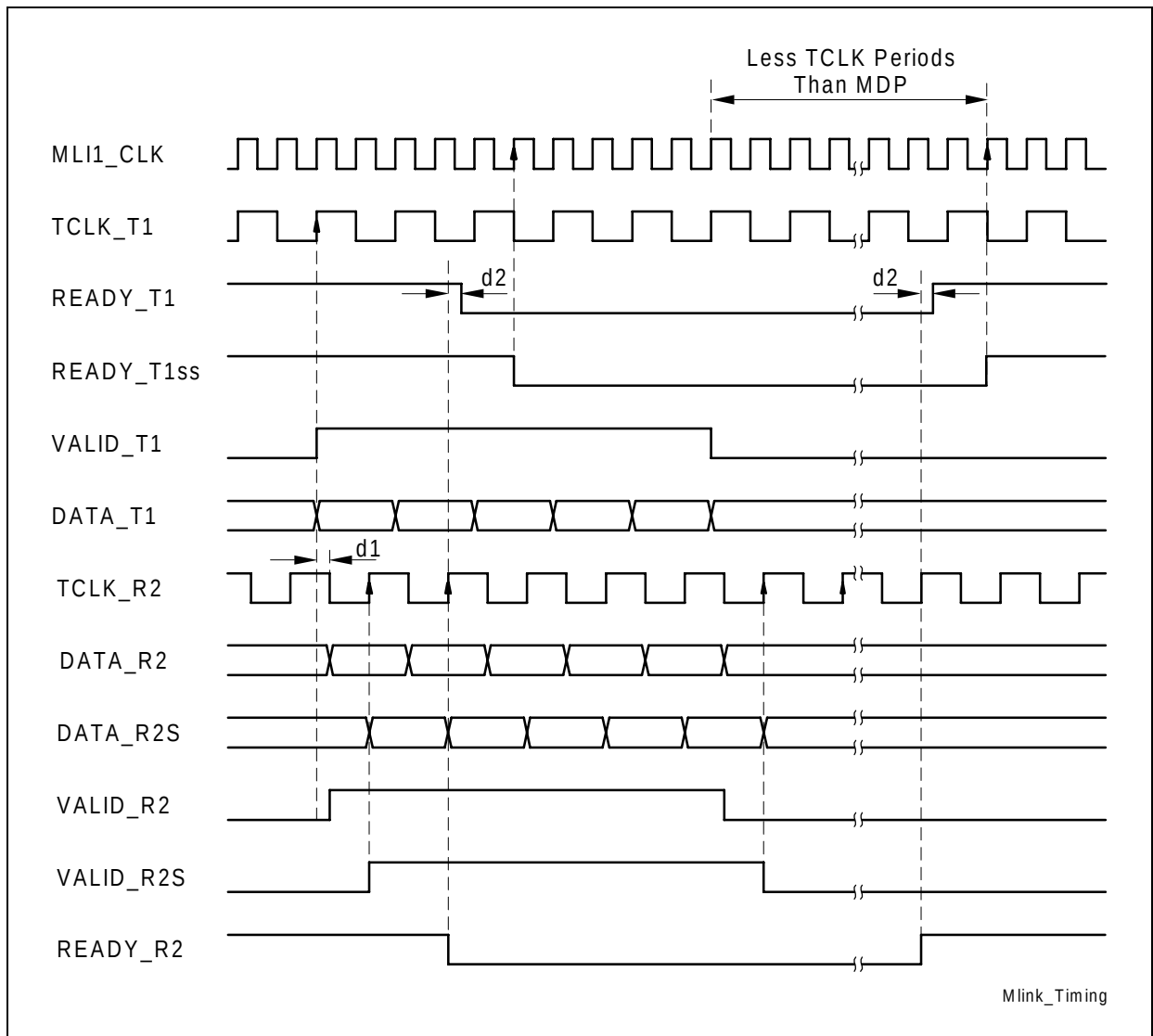


Figure 5-38 Detailed Handshake in a Correct Transfer

Figure 5-39 illustrates how the MLI receiver informs the transmitter that it has received a frame with a parity error.

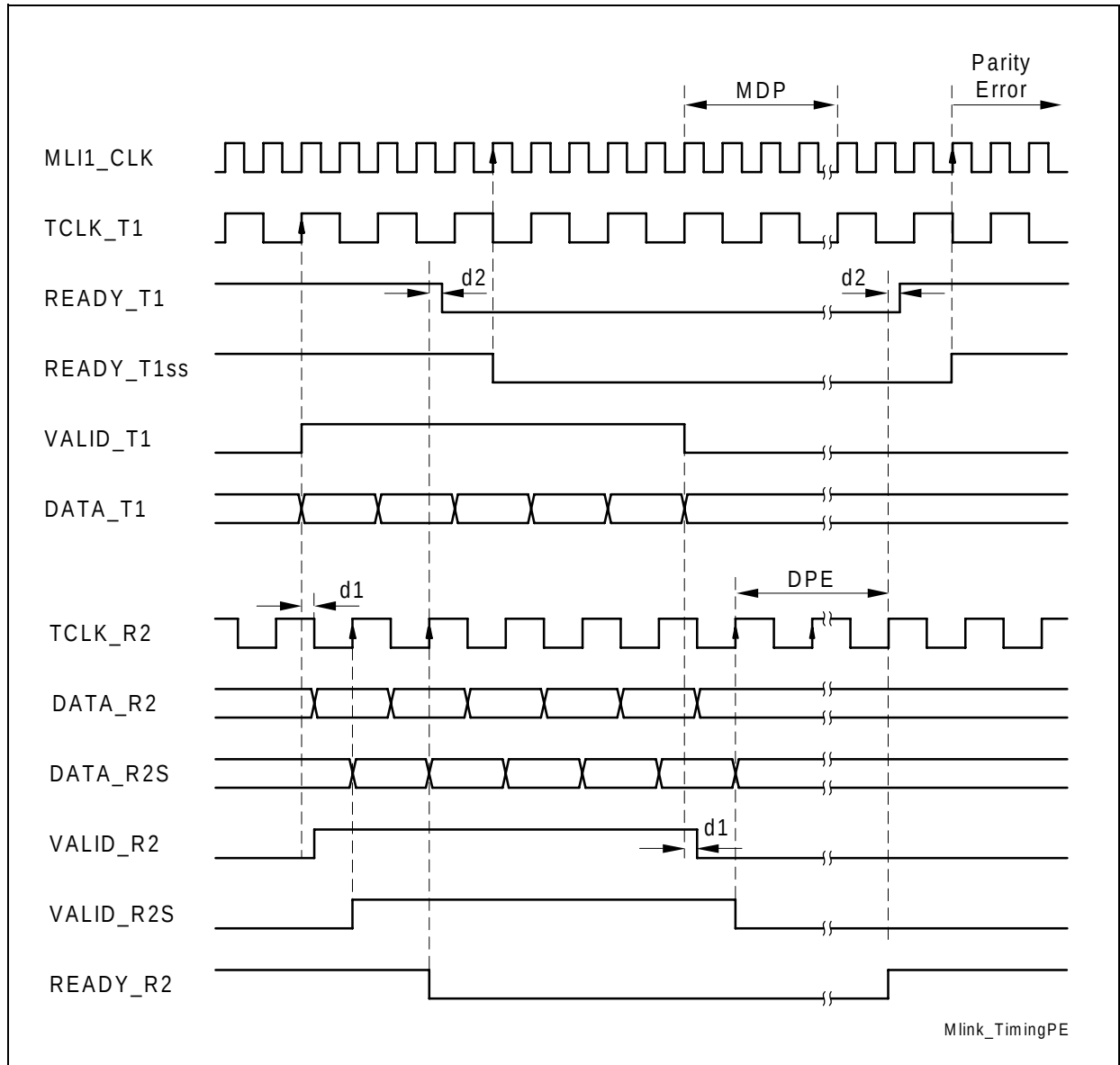


Figure 5-39 Detailed Handshake in a Transfer with Parity Error

Figure 5-40 illustrates the situation in which the MLI receiver has not acknowledged a transmission.

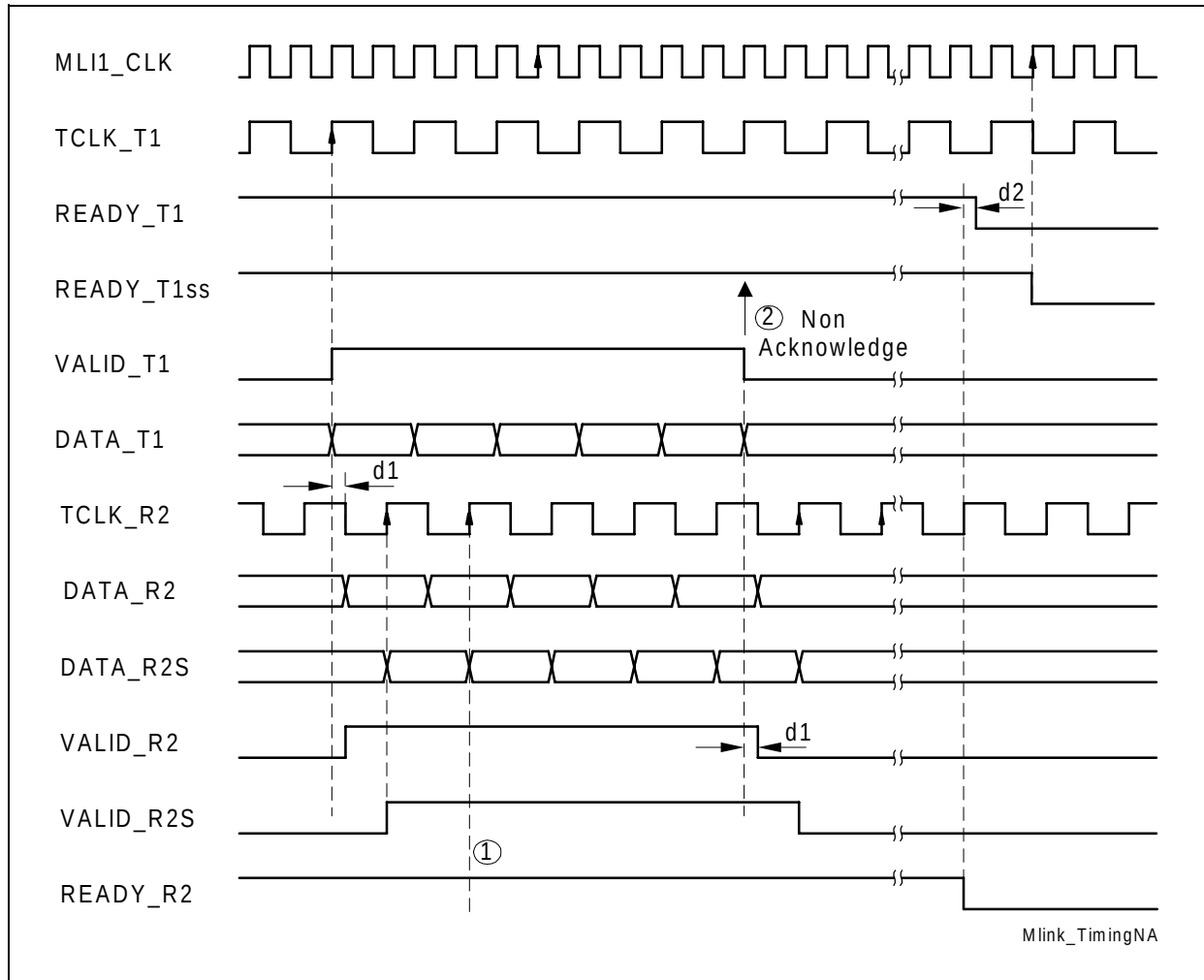


Figure 5-40 Detailed Handshake in a Transfer with Non-Acknowledge Error

In **Figure 5-40**, number 1 represents the moment in which the MLI receiver should set its READY signal. Number 2 represents the instant in which the MLI transmitter checks the READY signal status.

A non-acknowledge situation may lead into a time-out if the MLI receiver does not raise again the READY signal before the counter of non-acknowledge errors overflow.

5.1.12 Data Flow Description

5.1.12.1 Copy Base Address

The copy base address frame is used to transmit the two parameters of a remote window for pipe x, the 28 most-significant base address bits and the 4-bit coded buffer size, from the local microcontroller to the remote microcontroller.

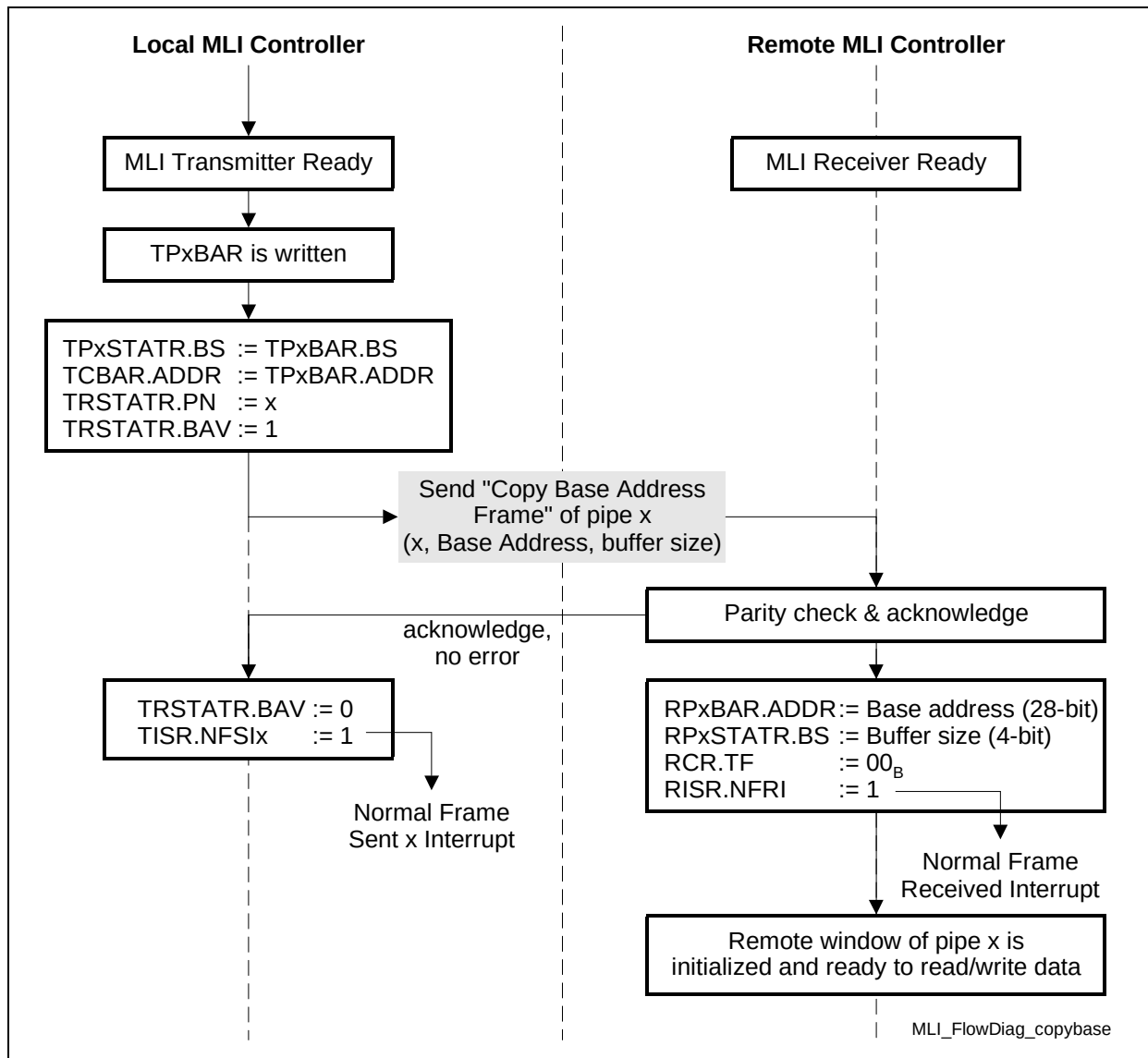


Figure 5-41 Copy Base Address Frame Flow

The transmission of a copy base address frame is initiated by writing the two parameters for a pipe remote window, the 28 most-significant base address bits and the 4-bit coded remote window (buffer) size, into register TPxBAR.

5.1.12.2 Command Frame

The transmission of a command frame is initiated by writing one of the four pipe x related command code bit fields in register TCMDR. Depending on the pipe x related command code that is transmitted, different actions are triggered in the remote controller.

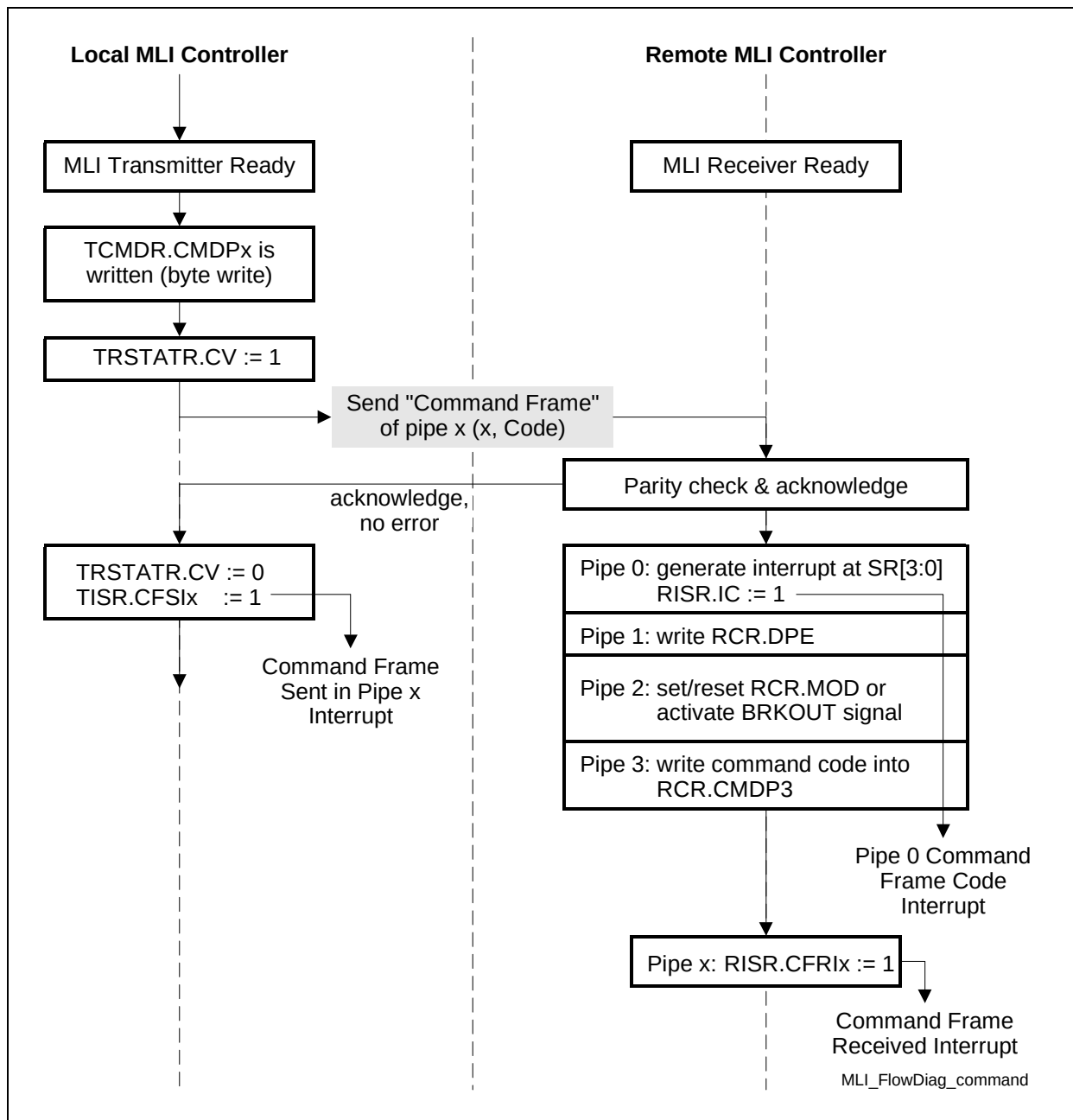


Figure 5-42 Command Frame Flow

5.1.12.3 Write Frame

The transmission of a write frame is initiated in the local controller by a write access of a bus master (e.g. the CPU) to one of the four transfer windows.

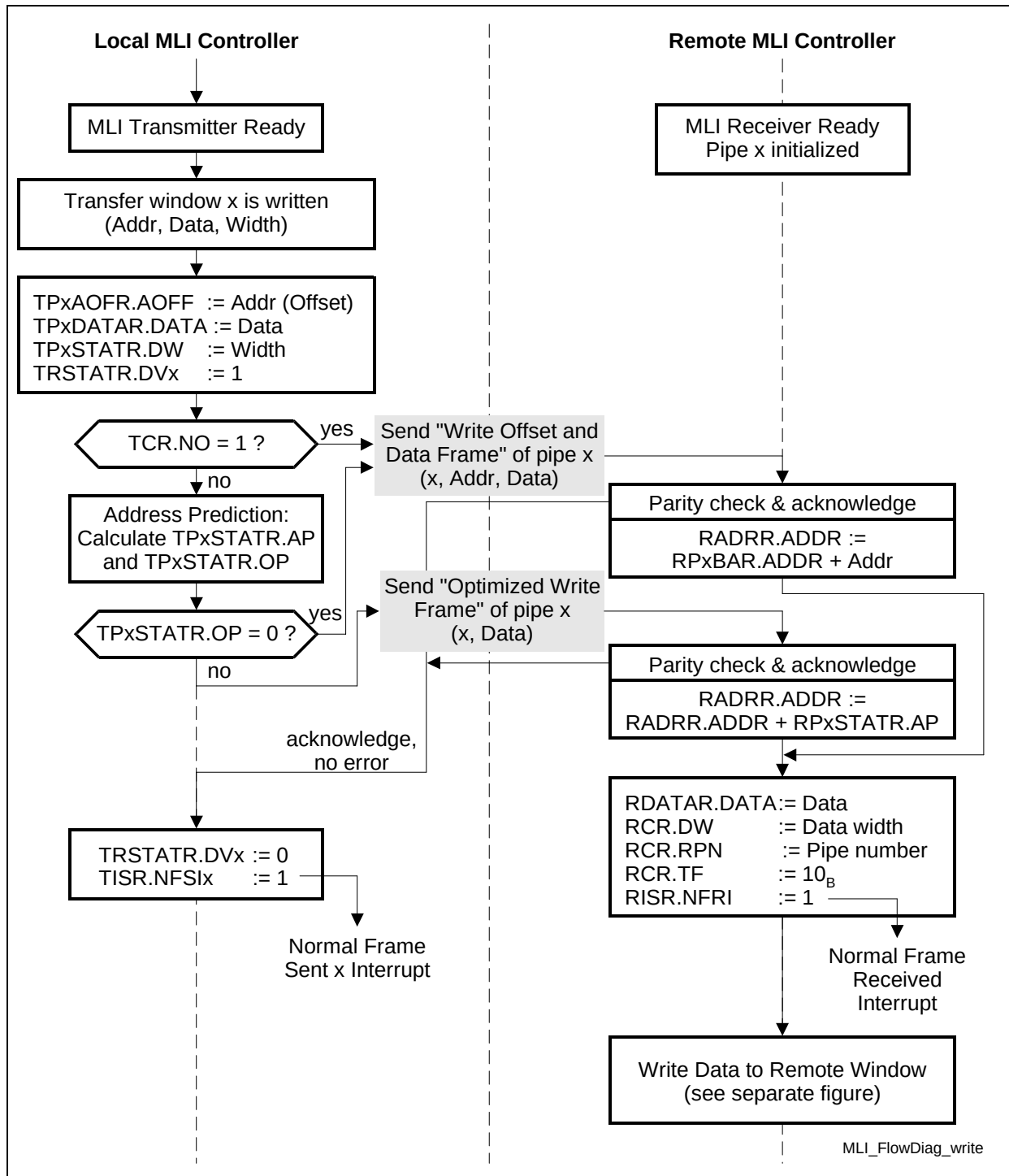


Figure 5-43 Write Frame Flow

5.1.12.4 Read Frame

The transmission of a read frame (see [Figure 5-44](#)) is initiated in the local controller by a read access of a bus master (e.g. the CPU) to one of the four transfer windows.

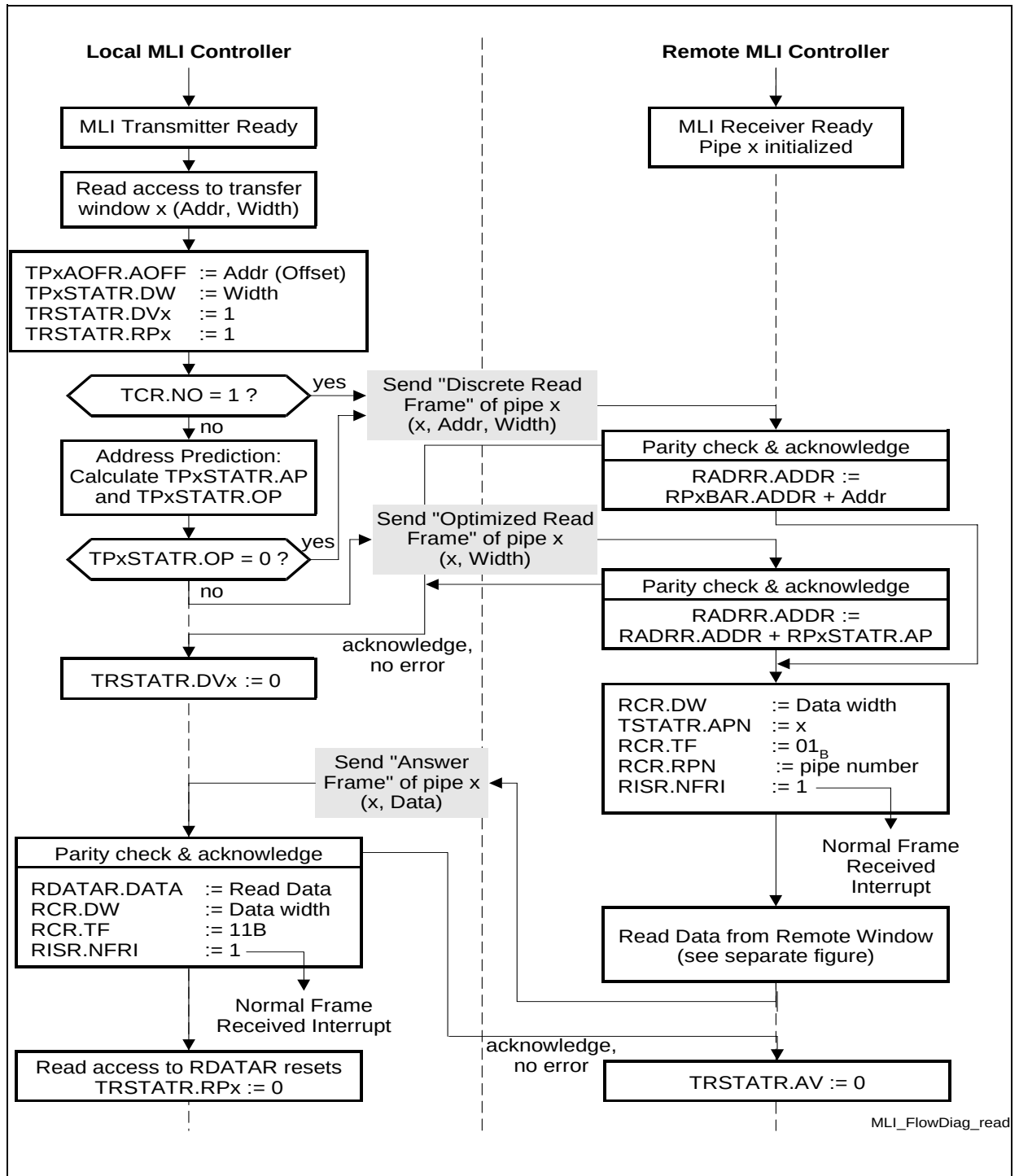


Figure 5-44 Read Frame Flow

5.1.12.5 Access to Remote Window

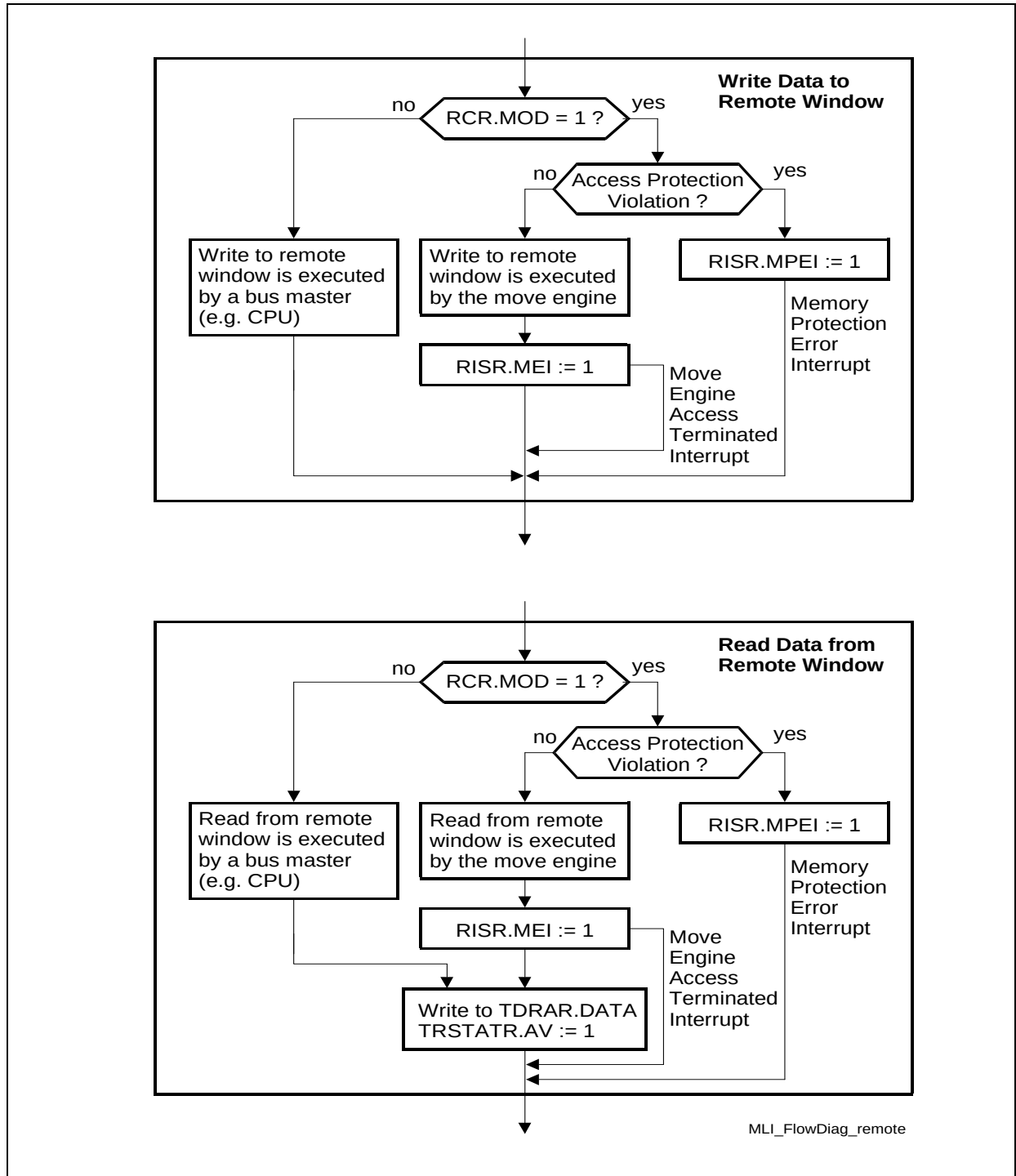


Figure 5-45 Access to Remote Window

5.2 MLI Kernel Registers

Figure 5-46 and Table 5-26 show all registers associated with the MLI Kernel.

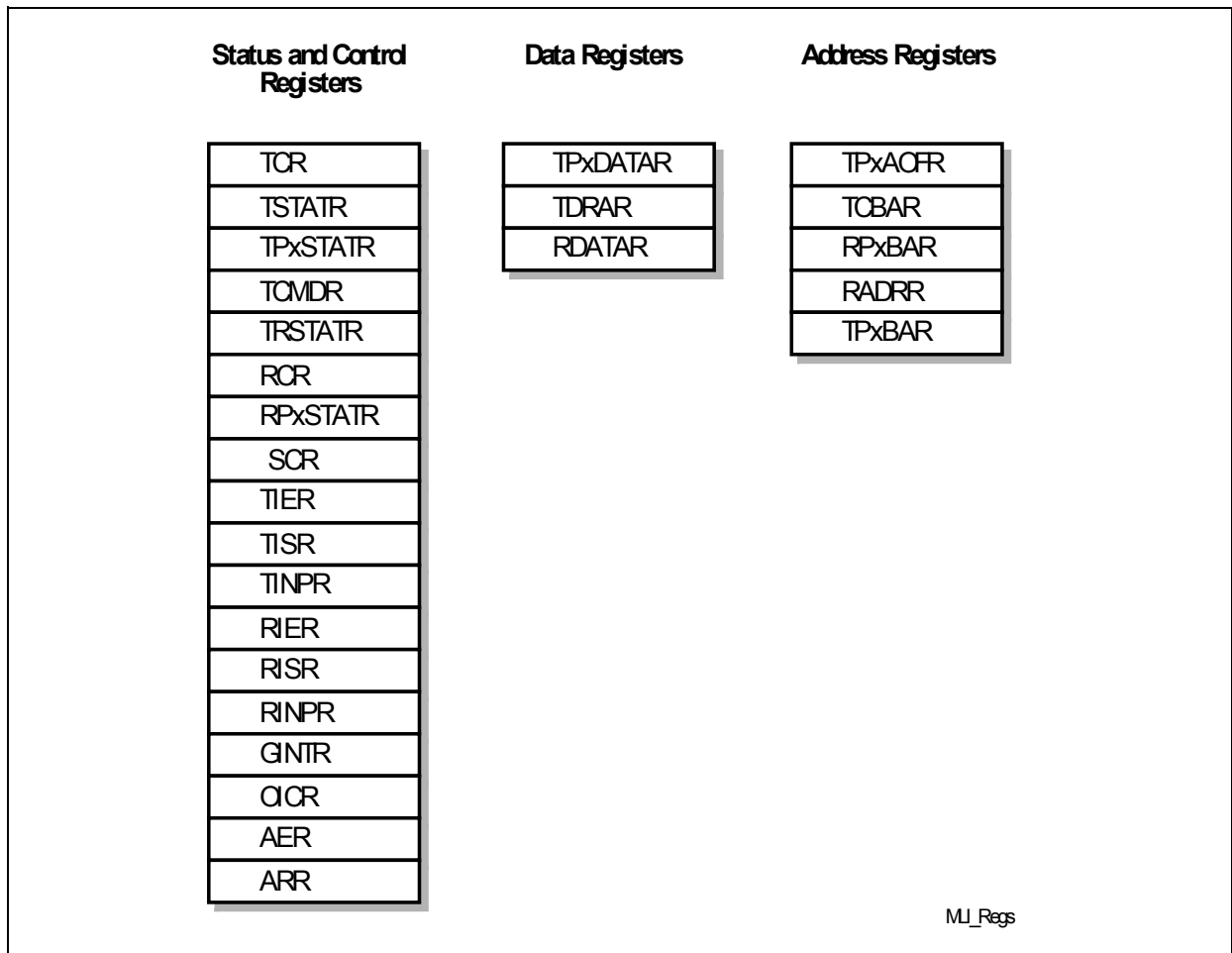


Figure 5-46 MLI Kernel Registers

Note: The letter "x" indicates the number of pipe (pipes 0, 1, 2 and 3).

Note: All bits marked 'w' return 0 when read.

Table 5-26 MLI Kernel Registers

Register Short Name	Register Long Name	Offset Address	Description see
TCR	Transmitter Control Register	0010 _H	Page 5-61
TSTATR	Transmitter Status Register	0014 _H	Page 5-63
TP0STATR	Transmitter Pipe 0 Status Register	0018 _H	Page 5-65
TP1STATR	Transmitter Pipe 1 Status Register	001C _H	Page 5-65
TP2STATR	Transmitter Pipe 2 Status Register	0020 _H	Page 5-65
TP3STATR	Transmitter Pipe 3 Status Register	0024 _H	Page 5-65
TCMDR	Transmitter Command Register	0028 _H	Page 5-66
TRSTATR	Transmitter Registers Status Register	002C _H	Page 5-68
TP0AOFR	Transmitter Pipe 0 Address Offset Register	0030 _H	Page 5-70
TP1AOFR	Transmitter Pipe 1 Address Offset Register	0034 _H	Page 5-70
TP2AOFR	Transmitter Pipe 2 Address Offset Register	0038 _H	Page 5-70
TP3AOFR	Transmitter Pipe 3 Address Offset Register	003C _H	Page 5-70
TP0DATAR	Transmitter Pipe 0 Data Register	0040 _H	Page 5-70
TP1DATAR	Transmitter Pipe 1 Data Register	0044 _H	Page 5-70
TP2DATAR	Transmitter Pipe 2 Data Register	0048 _H	Page 5-70
TP3DATAR	Transmitter Pipe 3 Data Register	004C _H	Page 5-70
TDRAR	Transmitter Data Read Answer Register	0050 _H	Page 5-71
TP0BAR	Transmitter Pipe 0 Base Address Register	0054 _H	Page 5-71
TP1BAR	Transmitter Pipe 1 Base Address Register	0058 _H	Page 5-71
TP2BAR	Transmitter Pipe 2 Base Address Register	005C _H	Page 5-71
TP3BAR	Transmitter Pipe 3 Base Address Register	0060 _H	Page 5-71
TCBAR	Transmitter Copy Base Address Register	0064 _H	Page 5-72
RCR	Receiver Control Register	0068 _H	Page 5-73
RP0BAR	Receiver Pipe 0 Base Address Register	006C _H	Page 5-76
RP1BAR	Receiver Pipe 1 Base Address Register	0070 _H	Page 5-76
RP2BAR	Receiver Pipe 2 Base Address Register	0074 _H	Page 5-76
RP3BAR	Receiver Pipe 3 Base Address Register	0078 _H	Page 5-76
RP0STATR	Receiver Pipe 0 Status Register	007C _H	Page 5-77
RP1STATR	Receiver Pipe 1 Status Register	0080 _H	Page 5-77

Table 5-26 MLI Kernel Registers (cont'd)

Register Short Name	Register Long Name	Offset Address	Description see
RP2STATR	Receiver Pipe 2 Status Register	0084 _H	Page 5-77
RP3STATR	Receiver Pipe 3 Status Register	0088 _H	Page 5-77
RADRR	Receiver Address Register	008C _H	Page 5-78
RDATAR	Receiver Data Register	0090 _H	Page 5-78
SCR	Set Clear Register	0094 _H	Page 5-79
TIER	Transmitter Interrupt Enable Register	0098 _H	Page 5-85
TISR	Transmitter Interrupt Status Register	009C _H	Page 5-86
TINPR	Transmitter Interrupt Node Pointer Register	00A0 _H	Page 5-87
RIER	Receiver Interrupt Enable Register	00A4 _H	Page 5-89
RISR	Receiver Interrupt Status Register	00A8 _H	Page 5-91
RINPR	Receiver Interrupt Node Pointer Register	00AC _H	Page 5-92
GINTR	Global Interrupt Set Register	00B0 _H	Page 5-93
OICR	Output Input Control Register	00B4 _H	Page 5-80
AER	Access Enable Register	00B8 _H	Page 5-94
ARR	Access Range Register	00BC _H	Page 5-95

5.2.1 MLI Transmitter Registers

TCR

Transmitter Control Register

Reset Value: 0000 0110_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TP	NO	MDP				MNAE		MPE				0	RTY	DNT	MOD
rw	rw	rw				rwh		rwh				r	rw	rw	rw

Field	Bits	Type	Description
MOD	0	rw	Mode of Operation This bit establishes the operation mode of the MLI transmitter. Its encoding is as follows: 0 MLI transmitter off 1 MLI transmitter on
DNT	1	rw	Data in Not Transmission This bit will determine the level of the data line when no transmission is in progress. If set to one, the data line when the transmission is finished will have the value 1. If set to zero the DATA line level will be 0.
RTY	2	rw	Retry This bit enables the retry mechanism for the transfer windows. 0 The retry mechanism is disabled. Any access while the transmitter is busy is discarded without additional action. 1 The retry mechanism is enabled. Any access while the transmitter is busy is acknowledged with a retry. In this case, the requesting bus master sends the requested access again until the request is accepted.

Field	Bits	Type	Description
MPE	[7:4]	rwh	Maximum Parity Errors This bit field indicates the value of parity errors for the parity interrupt generation. It is set to its maximum value by software and its number will be decreased by the MLI, each time it detects a parity error. Its encoding is as follows: 0000 _B Generate parity error interrupt 0001 _B 1 parity error for interrupt generation 0010 _B 2 parity errors for interrupt generation ... 1111 _B 15 parity errors for interrupt generation
MNAE	[9:8]	rwh	Maximum Non-Acknowledge Errors This bit field indicates the number of acknowledge errors for the time-out interrupt generation. It is set to its maximum value by software and its number will be decreased by the MLI, each time it detects a non-acknowledge error. Its encoding is as follows: 00 _B Generate non-acknowledge error interrupt 01 _B One error for the interrupt generation 10 _B Two errors for the interrupt generation 11 _B Three errors for the interrupt generation If the non-acknowledge error does not appear again before the counter reaches zero, the MLI resets again this field to its initial value.
MDP	[13:10]	rw	Maximum Delay for Parity Error This bit field is written by software and it defines the number of clock periods (from the clock used in the transmission) above which if the READY signal remains low it will be considered parity error condition. These bits will be interpreted as follows: 0000 _B Zero clock periods 0001 _B One clock period ... 1110 _B Fourteen clock periods 1111 _B Fifteen clock periods

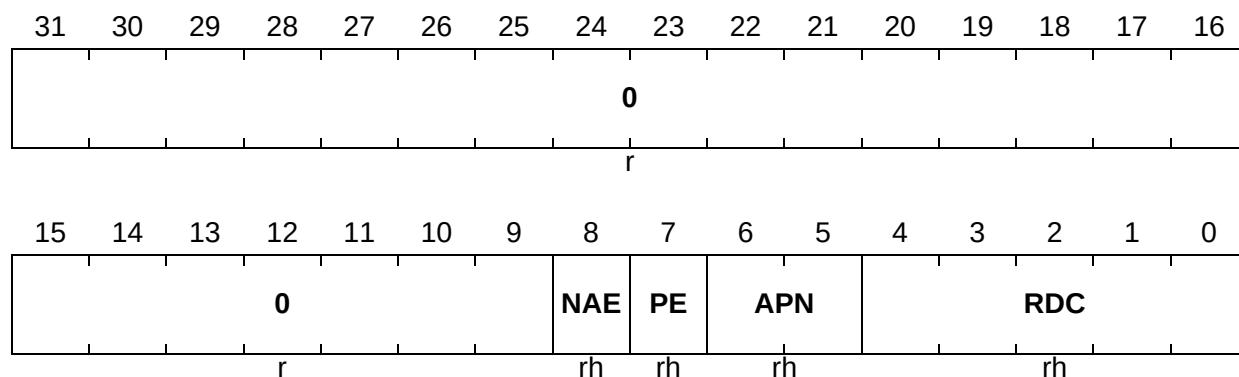
Micro Link Serial Bus Interface

Field	Bits	Type	Description
NO	14	rw	No Optimized Method This bit field indicates if the optimized method for address prediction is enabled or not. Its encoding is as follows: 0 Optimized method enabled 1 Optimized method not enabled
TP	15	rw	Type of Parity This bit will determine the type of parity used in the transmission. This value will be the one initially used for the toggle bit that produces the parity bit. Assuming a correct transmission, when set to one it will force the MLI receiver to produce a parity error condition.
0	3, [31:16]	r	Reserved ; read as 0; should be written with 0.

TSTATR

Transmitter Status Register

Reset Value: 0000 0000_H



Micro Link Serial Bus Interface

Field	Bits	Type	Description
RDC	[4:0]	rh	Ready Delay Counter This counter is reset to zero when the VALID signal goes low level after a transmission, and it will count TCLK periods until the MLI transmitter detects that the READY signal is high level again, or when the counter reaches its maximum value.
APN	[6:5]	rh	Answer Pipe Number This bit field is written by the MLI receiver whenever it gets a read frame. This value will be coincident with the pipe number of the received read frame and it will be used to send the answer frame. Its encoding is as follows: 00 _B Send the answer frame through pipe 0 01 _B Send the answer frame through pipe 1 10 _B Send the answer frame through pipe 2 10 _B Send the answer frame through pipe 3
PE	7	rh	Parity Error Flag Set to one when the MLI transmitter detects a parity error in the transmission. It is reset again when the MLI makes a transfer without parity error or when set the SCR.CTPE bit.
NAE	8	rh	Non-Acknowledge Error Flag Set to one when the MLI transmitter detects a non-acknowledge error in the transmission. It is reset again when the MLI makes a transfer without error or when set the SCR.CNAE bit.
0	[31:9]	r	Reserved ; read as 0; should be written with 0.

TP0STATR

Transmitter Pipe 0 Status Register

Reset Value: 0000 0000_H

TP1STATR

Transmitter Pipe 1 Status Register

Reset Value: 0000 0000_H

TP2STATR

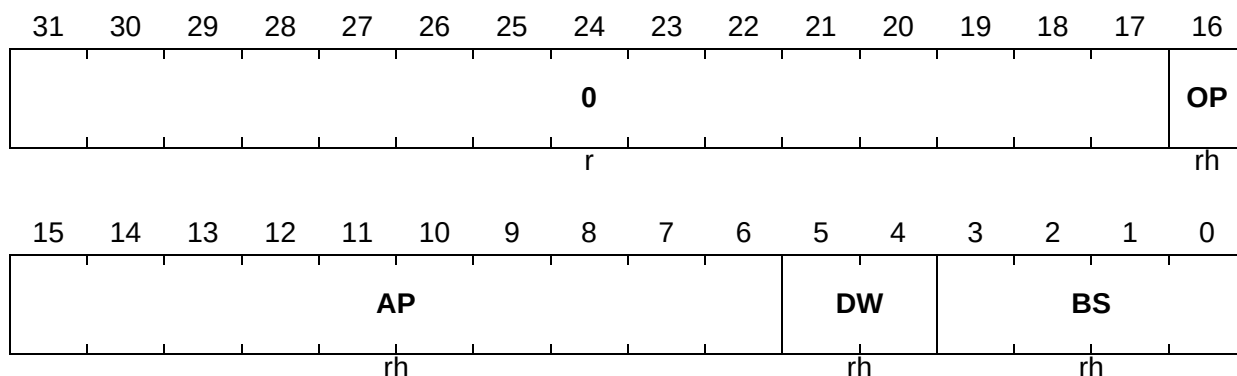
Transmitter Pipe 2 Status Register

Reset Value: 0000 0000_H

TP3STATR

Transmitter Pipe 3 Status Register

Reset Value: 0000 0000_H



Field	Bits	Type	Description
BS	[3:0]	rh	Buffer Size It gives the size of the transfer window in the second controller. The offset width will be coincident with this buffer size field. This field is updated by the MLI transmitter when the correspondent TPxBAR register is written. 0000 _B One bit offset 0001 _B Two bits offset ... 1111 _B Sixteen bits offset
DW	[5:4]	rh	Data Width This bit field defines the width of the data written in the TPxDATAR register. It is written by the MLI transmitter each time a new data is received in the TPxDATAR register. 00 _B Data width of 8 bits selected 01 _B Data width of 16 bits selected 10 _B Data width of 32 bits selected 11 _B Reserved

Field	Bits	Type	Description
AP	[15:6]	rh	Address Prediction Factor It is written by the MLI transmitter. It is used to keep track of the address prediction method. It represents a number of ten bits with sign in two's complement.
OP	16	rh	Use Optimized Frame This bit field is written by the MLI transmitter each time it performs the necessary operations to know if the address offset follows the address prediction scheme. Its meaning is as follows: 0 Do not use optimized mode to send the frame 1 Use optimized mode to send the frame It is used by the MLI to know if the optimized method should or not be used when a frame is selected to be sent.
0	[31:17]	r	Reserved ; read as 0; should be written with 0.

The TCMDR register must be written only bite-wisely. Write accesses with a larger data width than a byte are forbidden.

TCMDR

Transmitter Command Register

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0				CMDP3				0				CMDP2			
r				rw				r				rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0				CMDP1				0				CMDP0			
r				rw				r				rw			

Field	Bits	Type	Description
CMDP0	[3:0]	rw	Command in pipe 0 This bit field is written by software and it defines the command to be sent through pipe 0. These bits will be interpreted as follows: 0000 _B Program MLI receiver to produce interrupt 0 0001 _B Program MLI receiver to produce interrupt 1 0010 _B Program MLI receiver to produce interrupt 2 0011 _B Program MLI receiver to produce interrupt 3 others No effect
CMDP1	[11:8]	rw	Command in pipe 1 This bit field is written by software and it defines the command to be sent through pipe 1. These bits will be interpreted as follows: 0000 _B Make RCR.DPE = 0000 _B 0001 _B Make RCR.DPE = 0001 _B ... 1111 _B Make RCR.DPE = 1111 _B
CMDP2	[19:16]	rw	Command in pipe 2 This bit field is written by software and it defines the command to be sent through pipe 2, see Table 5-22 .
CMDP3	[27:24]	rw	Command in pipe 3 This bit field is written by software and it defines the command to be sent through pipe 3. The commands will be software interpreted.
0	[7:4] [15:12] [23:20] [31:28]	r	Reserved ; read as 0; should be written with 0.

TRSTATR

Transmitter Registers Status Register

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0						PN	RP3	RP2	RP1	RP0	DV3	DV2	DV1	DV0	
r						rh	rh	rh	rh	rh	rh	rh	rh	rh	rh
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0						BAV	AV	CV3	CV2	CV1	CV0	CIV3	CIV2	CIV1	CIV0
r						rh	rh	rh	rh	rh	rh	rh	rh	rh	rh

Field	Bits	Type	Description
CIVx, x = 0, 1, 2, 3	[3:0]	rh	Command Interrupt Valid Set to one by the MLI transmitter whenever it detects a rising edge in the correspondent trigger_commandx line. It is reset again when the correspondent command frame to the line is correctly sent through pipe 0, or when set SCR.CCIVx.
CVx, x = 0, 1, 2, 3	[7:4]	rh	Command Valid Set to one by the MLI transmitter when the TCMDR.CMDPx bit field is written, or when set SCR.SCVx. It is reset again when the command frame is correctly sent, or when set SCR.CCVx.
AV	8	rh	Answer Valid Set to one by the MLI transmitter when the TDRAR register is written. It is reset again when the answer frame is correctly sent, or when set SCR.CAV.
BAV	9	rh	Base Address Valid Set to one by the MLI transmitter when the TCBAR register is written. It is reset again when the copy base address frame is correctly sent, or when set SCR.CBAV.
DVx, x = 0, 1, 2, 3	[19:16]	rh	Data Valid Set to one by the MLI transmitter when the TPxDATAR and/or the TPxAOFR registers are written. It is reset again when the read or write frame is correctly sent, or when set SCR.CDVx.

Micro Link Serial Bus Interface

Field	Bits	Type	Description
RPx, x = 0, 1, 2, 3	[23:20]	rh	Read Pending Set to one by the MLI transmitter when the TPxAOFR register is written for a read operation. It is reset again when MLI receiver gets an answer frame and the answer is read out, i.e. RDATA is read, or when set SCR.CRPx.
PN	[25:24]	rh	Pipe Number This bit field will indicate to which pipe number corresponds the base address that has been stored in the TCBAR register, and it is written by the MLI transmitter each time it writes TCBAR with the data from the correspondent TPxBAR register. Its encoding is as follows: 00 _B Pipe number 0 01 _B Pipe number 1 01 _B Pipe number 2 11 _B Pipe number 3
0	[15:10] [31:26]	r	Reserved ; read as 0; should be written with 0.

TP0AOFR

Transmitter Pipe 0 Address Offset Register

Reset Value: 0000 0000_H

TP1AOFR

Transmitter Pipe 1 Address Offset Register

Reset Value: 0000 0000_H

TP2AOFR

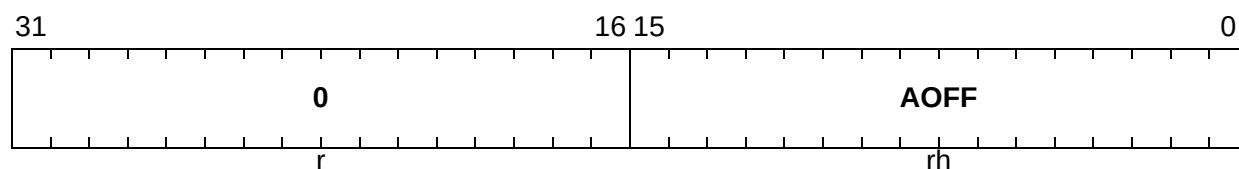
Transmitter Pipe 2 Address Offset Register

Reset Value: 0000 0000_H

TP3AOFR

Transmitter Pipe 3 Address Offset Register

Reset Value: 0000 0000_H



Field	Bits	Type	Description
AOFF	[15:0]	rh	Address Offset This address offset together with the base address of the pipe will point to an address position in the transfer window of the other controller in which the controller wants to write or read.
0	[31:16]	r	Reserved ; read as 0; should be written with 0.

TP0DATAR

Transmitter Pipe 0 Data Register

Reset Value: 0000 0000_H

TP1DATAR

Transmitter Pipe 1 Data Register

Reset Value: 0000 0000_H

TP2DATAR

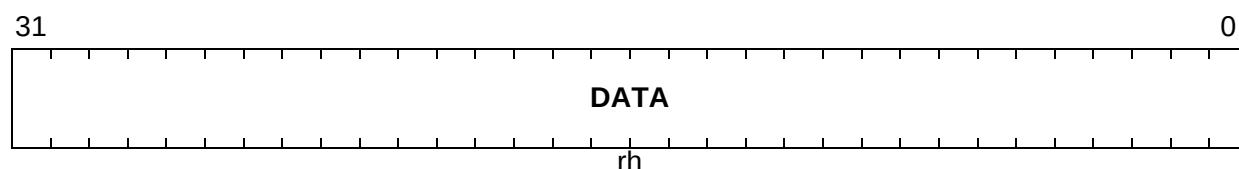
Transmitter Pipe 2 Data Register

Reset Value: 0000 0000_H

TP3DATAR

Transmitter Pipe 3 Data Register

Reset Value: 0000 0000_H



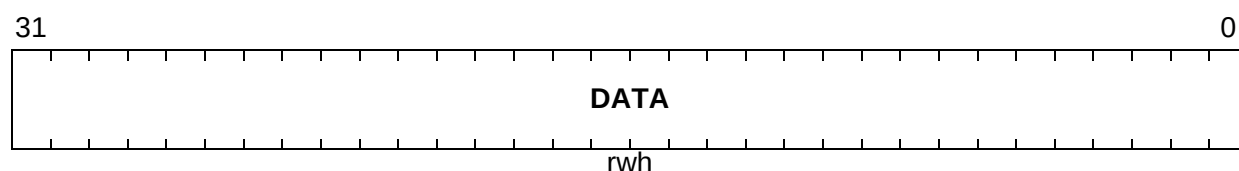
Micro Link Serial Bus Interface

Field	Bits	Type	Description
DATA	[31:0]	rh	Data Contains the data that will be sent to the MLI receiver in the other controller through the correspondent pipe. It will be the data to be written and to send it the MLI will use a write frame (optimized or not).

TDRAR

Transmitter Data Read Answer Register

Reset Value: 0000 0000_H



Field	Bits	Type	Description
DATA	[31:0]	rwh	Data Contains the data that proceeds from a read operation and it will be sent to the MLI receiver in the other controller. It will be the data that is sent in answer frame.

When the LSB of TPxBAR register is updated, the 28 MSBs are directly copied in the register TCBAR, and the BS bit field is copied in the correspondent TPxSTATR register.

TP0BAR

Transmitter Pipe 0 Base Address Register

Reset Value: 0000 0000_H

TP1BAR

Transmitter Pipe 1 Base Address Register

Reset Value: 0000 0000_H

TP2BAR

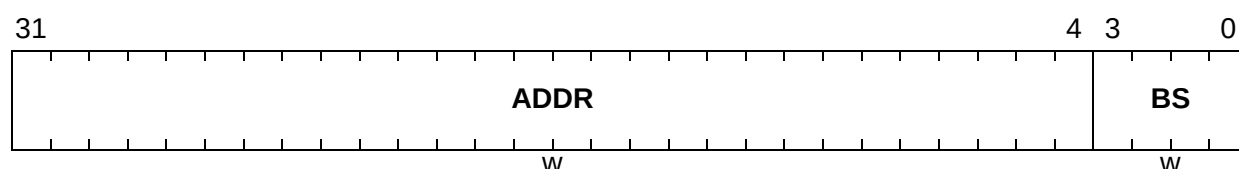
Transmitter Pipe 2 Base Address Register

Reset Value: 0000 0000_H

TP3BAR

Transmitter Pipe3 Base Address Register

Reset Value: 0000 0000_H



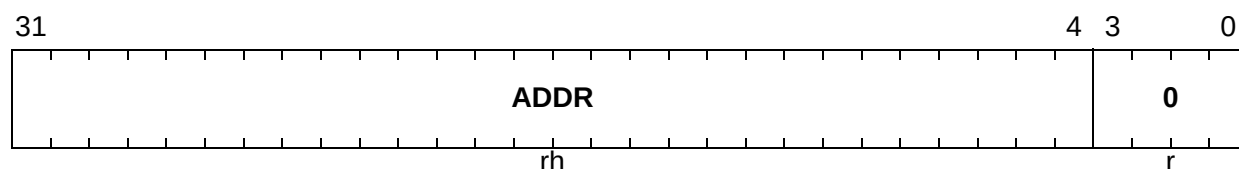
Field	Bits	Type	Description
BS	[3:0]	w	Buffer Size It gives the used size of the remote window in the remote controller. The offset width will be coincident with this buffer size field. 0000 _B One bit offset 0001 _B Two bits offset 0010 _B Three bits offset ... 1110 _B Fourteen bits offset 1111 _B Fifteen bits offset
ADDR	[31:4]	w	Address This bit field contains the base address 28 MSBs of the correspondent pipe. It is written each time the controller wanted to initialize a pipe in the other controller.

The TCBAR register is updated with the 28 MSBs contained in the latest accessed TPxBAR register. The trigger condition to make the copy is that the LSB of any TPxBAR registers had been written.

TCBAR

Transmitter Copy Base Address Register

Reset Value: 0000 0000_H



Field	Bits	Type	Description
ADDR	[31:4]	rh	Address This bit field contains the base address 28 MSBs of any of the four pipes.
0	[3:0]	r	Reserved ; read as 0; should be written with 0.

Note: The TRSTATR.BAVx flag indicates if the TCBAR register contains a base address that has been sent or not. The flag is reset again when the copy base address frame has been correctly sent.

5.2.2 MLI Receiver Registers

RCR

Receiver Control Register

Reset Value: 0100 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0							RCV RST	0			BEN	MPE			
r							rw	r			rw	rwh			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RPN		PE		TF		DW		MOD	CMDP3			DPE			
rh		rh		rh		rh		rh	rh			rh			

Field	Bits	Type	Description
DPE	[3:0]	rh	Delay for Parity Error This bit field is written by the MLI when it receives the proper command to program it. It defines the number of clock periods (from the clock used in the transmission) that as minimum the MLI receiver has to wait to raise again the READY signal when it has detected a parity error. 0000 _B Wait zero clock periods 0001 _B Wait one clock period 0010 _B Wait two clock periods ... 1110 _B Wait fourteen clock periods 1111 _B Wait fifteen clock periods
CMDP3	[7:4]	rh	Command From Pipe 3 Whenever the MLI receiver gets a command frame trough the pipe 3, it stores its value in this bit field. The command will be interpreted by software.

Micro Link Serial Bus Interface

Field	Bits	Type	Description
MOD	8	rh	Mode of Operation This bit field defines the mode of operation of the MLI receiver. The MLI receiver may write this bit field when it receives the proper command from other controller. This bit is also set and reset when its correspondent bits of the clear and set register are set, SCR.CMOD and SCR.SMOD. 0 The automatic read/write handling is disabled. The request is stored in the registers and can be served by the CPU (listening mode). 1 The automatic read/write handling is enabled.
DW	[10:9]	rh	Data Width This bit field is updated by the MLI receiver whenever it writes new data in the RDATA register. Bit field DW is used by the MLI receiver when it delivered the data or by the software whenever it had to fetch data from the MLI receiver RDATA register. It is updated by read frames, write frames or answer frames. 00_B Data width of 8 bits selected 01_B Data width of 16 bits selected 10_B Data width of 32 bits selected 11_B Reserved
TF	[12:11]	rh	Type of Frame Set by the MLI receiver when it writes the RDATA, RADRR and RPxBAR registers. Its value depends on the kind of frame in which the data was received. This bit field will be used by the software in order to know where it must take the data from and how to deliver it. It is updated by copy base address frames, read frames, write frames or answer frames. 00 Copy base address frame 01 Read frame, optimized or not 10 Write frame, optimized or not 11 Answer frame
PE	13	rh	Parity Error Set to one when the MLI receiver detects a parity error in the transmission. It is reset again when the MLI receives a transfer without parity error, or when set SCR.CRPE.

Micro Link Serial Bus Interface

Field	Bits	Type	Description
RPN	[15:14]	rh	Received Pipe Number This bit field contains the pipe number that was indicated by the pipe number bit field of the latest received frame. It is updated by copy base address frames, read frames, write frames or answer frames.
MPE	[19:16]	rwh	Maximum Parity Errors This bit field indicates after how many parity errors the parity error interrupt will be generated. It is set to a desired value by software and it is decremented automatically by the MLI, each time it detects a parity error. If 0, each parity error will generate an interrupt and MPE stays 0. 0000 Each parity error will generate the interrupt. 0001 After 2 parity errors the interrupt is generated. 0010 After 3 parity errors the interrupt is generated. ... 1111 After 16 parity errors the interrupt is generated.
BEN	20	rw	Break Out Enable If set to one the MLI receiver will produce a pulse in its BREAKOUT line when received the corresponding command frame.
RCVRST	24	rw	Receiver Reset This bit forces the receiver to reset in order to be able to change OICR settings without influencing the receiver register. During chip reset, this bit is 1 and in the second clock cycle after the chip reset, the specified reset value is applied. This ensures that no spikes/pulses are received during chip reset (OICR might change). 0 The receiver is not held in reset (operating mode). 1 The receiver is held in reset.
0	[23:21], [31:25]	r	Reserved ; read as 0; should be written with 0.

Micro Link Serial Bus Interface

RP0BAR

Receiver Pipe 0 Base Address Register

Reset Value: 0000 0000_H

RP1BAR

Receiver Pipe 1 Base Address Register

Reset Value: 0000 0000_H

RP2BAR

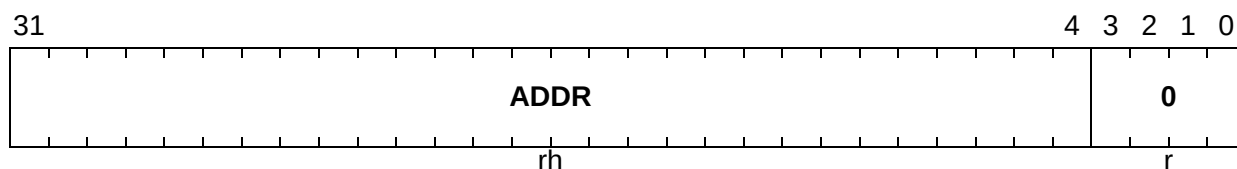
Receiver Pipe 2 Base Address Register

Reset Value: 0000 0000_H

RP3BAR

Receiver Pipe 3 Base Address Register

Reset Value: 0000 0000_H



Field	Bits	Type	Description
ADDR	[31:4]	rh	It will contain the base address of the corresponding transfer window in its 28 MSBs after receiving the copy base address frame. Each time the MLI receiver got a new address offset for a read or write operation in the pipe, it will be concatenated with this base address and it will represent the current absolute base address.
0	[3:0]	r	Reserved ; read as 0; should be written with 0.

RP0STATR

Receiver Pipe 0 Status Register

Reset Value: 0000 0000_H

RP1STATR

Receiver Pipe 1 Status Register

Reset Value: 0000 0000_H

RP2STATR

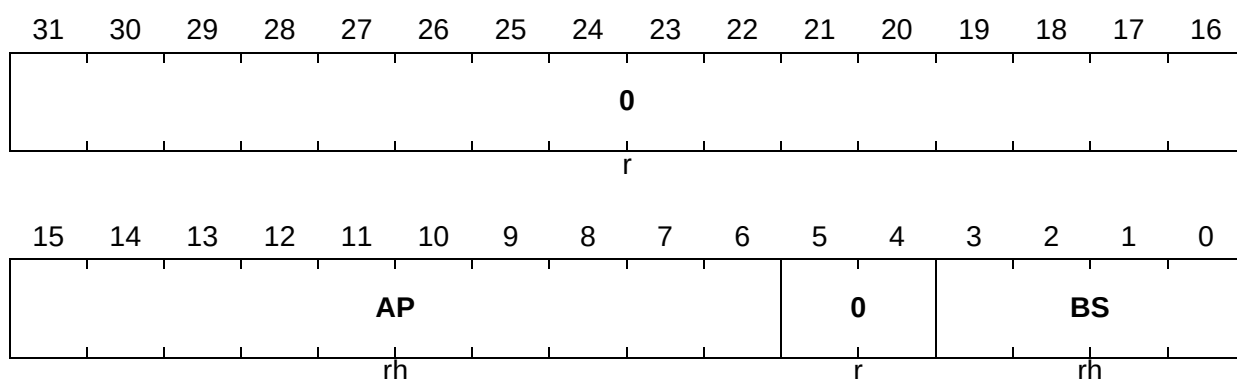
Receiver Pipe 2 Status Register

Reset Value: 0000 0000_H

RP3STATR

Receiver Pipe 3 Status Register

Reset Value: 0000 0000_H

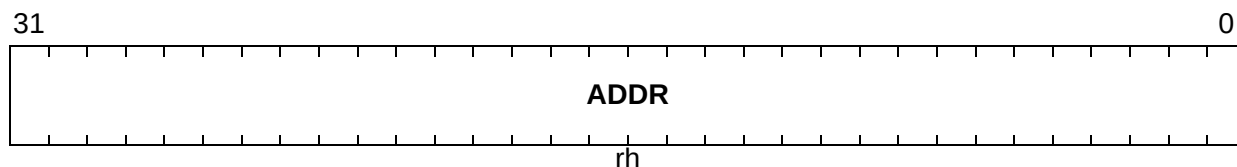


Field	Bits	Type	Description
BS	[3:0]	rh	Buffer Size It is written by the MLI receiver each time it receives a copy base address frame. The offset width of the offset field received in the frames will be coincident with this buffer size field. 0000 _B One bit offset 0001 _B Two bits offset 0010 _B Three bits offset ... 1110 _B Fourteen bits offset 1111 _B Sixteen bits offset
AP	[15:6]	rh	Address Prediction Factor It is written by the MLI receiver. It is used to keep track of the address prediction method. It represents a number of ten bits with sign in two's complement.
0	[5:4] [31:16]	r	Reserved ; read as 0; should be written with 0.

RADRR

Receiver Address Register

Reset Value: 0000 0000_H

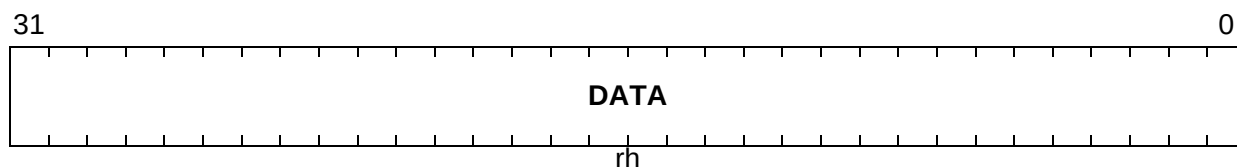


Field	Bits	Type	Description
ADDR	[31:0]	rh	Address It contains the destination absolute address in which the data will be written or from which the data will be read. Its value is totally synchronous to the MLI receiver internal clock.

RDATAR

Receiver Data Register

Reset Value: 0000 0000_H



Field	Bits	Type	Description
DATA	[31:0]	rh	Data It contains the data that will be written in the controller. It can be data of a write operation or the result from a read operation. Its value is totally synchronous to the MLI receiver internal clock.

5.2.3 MLI Kernel Common Registers

SCR

Set Clear Register

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCIV 3	CCIV 2	CCIV 1	CCIV 0	CNA E	CTP E	CRP E	CAV				0			CBA V	CMO D
w	w	w	w	w	w	w	w				r			w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCV 3	CCV 2	CCV 1	CCV 0	CDV 3	CDV 2	CDV 1	CDV 0			0	SMO D	SCV 3	SCV 2	SCV 1	SCV 0
w	w	w	w	w	w	w	w			r	w	w	w	w	w

Field	Bits	Type	Description
SCVx x = 0, 1, 2, 3	[3:0]	w	Set Command Valid 0 No effect. 1 Bit TRSTATR.CVx is set.
SMOD	4	w	Set MOD Flag 0 No effect. 1 Bit RCR.MOD is set.
CDVx x = 0, 1, 2, 3	[11:8]	w	Clear Data Valid 0 Flag 0 No effect. 1 Bits TRSTATR.DVx and TRSTATR.RPx are cleared.
CCVx x = 0, 1, 2, 3	[15:12]	w	Clear Command Valid 0 Flag 0 No effect. 1 Bit TRSTATR.CVx is cleared.
CMOD	16	w	Clear MOD Flag 0 No effect. 1 Bit RCR.MOD is cleared.
CBAV	17	w	Clear BAV Flag 0 No effect. 1 Bit TRSTATR.BAV is cleared.
CAV	24	w	Clear AV Flag 0 No effect. 1 Bit TRSTATR.AV is cleared.

Field	Bits	Type	Description
CRPE	25	w	Clear Receiver PE Flag 0 No effect. 1 Bit RCR.PE is cleared.
CTPE	26	w	Clear Transmitter PE Flag 0 No effect. 1 Bit TSTATR.PE is cleared.
CNAE	27	w	Clear NAE Flag 0 No effect. 1 Bit TSTATR.NAE is cleared.
CCIVx x = 0, 1, 2, 3	[31:28]	w	Clear Command Interrupt Valid x Flag 0 No effect. 1 Bit TRSTATR.CIVx is cleared.
0	[7:5], [23:18]	r	Reserved ; read as 0; should be written with 0.

Note: The implementation of this register does not involve any flip-flop.

OICR

Output Input Control Register

Reset Value: 1000 8000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RDP	RDS	RCE	RCP	RCS	RVP	RVS	RRP D	RRP C	RRP B	RRP A	RRS				
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RVE	TDP	TCP	TCE	TRE	TRP	TRS	TVP D	TVP C	TVP B	TVP A	TVE D	TVE C	TVE B	TVE A	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Micro Link Serial Bus Interface

Field	Bits	Type	Description
TVEA, TVEB, TVEC, TVED	0 1 2 3	rw	Transmitter Valid Enable These bits field enable the transmitter output signals TVALIDx that are driven outside the module. 0 The transmitter output signal TVALIDx is considered as passive. 1 The transmitter output signal TVALIDx reflects the status of the current transmitter kernel signal VALID.
TVPA, TVPB, TVPC, TVPD	4 5 6 7	rw	Transmitter Valid Polarity These bits define the polarity of each of the transmitter output signals TVALIDx. 0 An active TVALIDx line is driving a "1", a passive level is "0" (not inverted). 1 An active TVALIDx line is driving a "0", a passive level is "1" (inverted).
TRS	[9:8]	rw	Transmitter Ready Selector This bit field defines the transmitter input line that is used for the transmitter kernel signal READY. 00 TREADYA is selected. 01 TREADYB is selected. 10 TREADYC is selected. 11 TREADYD is selected.
TRP	10	rw	Transmitter Ready Polarity This bit defines the polarity of the selected transmitter input signal TREADYx. 0 An active TREADYx level is "1", a passive level is "0" (not inverted). 1 An active TREADYx level is "0", a passive level is "1" (inverted).
TRE	11	rw	Transmitter Ready Enable This bit enables the input of the transmitter kernel signal READY. 0 The READY signal is considered as passive (internal = 0). 1 The TREADYx line according to the bit fields TRS and TRP is taken into account.

Field	Bits	Type	Description
TCE	12	rw	Transmitter Clock Enable This bit enables the transmitter kernel signal CLK to be driven outside the module. 0 The transmitter clock signal is considered as passive (internal = 0). 1 The TCLK line reflects the status of the current transmitter kernel signal CLK according to bit TCP.
TCP	13	rw	Transmitter Clock Polarity This bit defines the polarity of the transmitter output signal TCLK. 0 A passive TCLK line is driving a "0" (not inverted). 1 A passive TCLK line is driving a "1" (inverted).
TDP	14	rw	Transmitter Data Polarity This bit defines the polarity of the transmitter output signal TCLK. 0 The transmitter kernel signal DATA drives directly the transmitter output line TDATA (not inverted). 1 The transmitter kernel signal DATA is inverted before driving the transmitter output line TDATA.
RVE	15	rw	Receiver Valid Enable This bit enables the receiver kernel input signal VALID. 0 The VALID signal is considered as passive (internal = 0). 1 The RVALIDx line according to the bit fields RVS and RVP is taken into account.
RRS	[17:16]	rw	Receiver Ready Selector This bit field defines the receiver output signal RREADYx that is driven outside the module by the receiver kernel signal READY. An RREADYx output signal that is not selected is considered as passive and drives a level according to its corresponding bit RRPx. 00 Select RREADYA. 01 Select RREADYB. 10 Select RREADYC. 11 Select RREADYD.

Micro Link Serial Bus Interface

Field	Bits	Type	Description
RRPA, RRPB, RRPC, RRPD	18 19 20 21	rw	Receiver Ready Polarity These bits define the polarity of the receiver output signals RREADYx. 0 An active RREADYx line level is "1", a passive level is "0" (not inverted). 1 An active RREADYx line level is "0", a passive level is "1" (inverted).
RVS	[23:22]	rw	Receiver Valid Selector This bit defines which one of the receiver input signals RVALIDx that is taken as input for the receiver kernel signal VALID. 00 Select RVALIDA. 01 Select RVALIDB. 10 Select RVALIDC. 11 Select RVALIDD.
RVP	24	rw	Receiver Valid Polarity This bit defines the polarity of the selected receiver input signal RVALIDx. 0 An active RVALIDx level is "1", a passive level is "0" (not inverted). 1 An active RVALIDx level is "0", a passive level is "1" (inverted).
RCS	[26:25]	rw	Receiver Clock Selector This bit defines which one of the receiver input signals RCLKx that is taken as input for the receiver kernel signal CLK. 00 Select RCLKA. 01 Select RCLKB. 10 Select RCLKC. 11 Select RCLKD.
RCP	27	rw	Receiver Clock Polarity This bit defines the polarity of the selected receiver input signal RCLKx. 0 An active RCLKx level is "1", a passive level is "0" (not inverted). 1 An active RCLKx level is "0", a passive level is "1" (inverted).

Micro Link Serial Bus Interface

Field	Bits	Type	Description
RCE	28	rw	Receiver Clock Enable This bit enables the receiver kernel input signal CLK. 0 The CLK signal is considered as passive (internal = 0). 1 The RCLKx line according to the bit fields RCS and RCP is taken into account.
RDS	[30:29]	rw	Receiver Data Selector This bit defines which one of the receiver input signals RDATAx that is taken as input for the receiver kernel signal DATA. 00 Select RDATAA. 01 Select RDATAB. 10 Select RDATA C. 11 Select RDATA D.
RDP	31	rw	Receiver Data Polarity This bit defines the polarity of the selected receiver input signal RDATAx. 0 The receiver kernel signal DATA is directly driven by the selected receiver input line RDATAx (not inverted). 1 The receiver kernel signal DATA is driven by the inverted signal from the selected receiver input line RDATAx.

5.2.4 MLI Interrupt Registers

TIER

Transmitter Interrupt Enable Register

Reset Value: 0000 0000_H

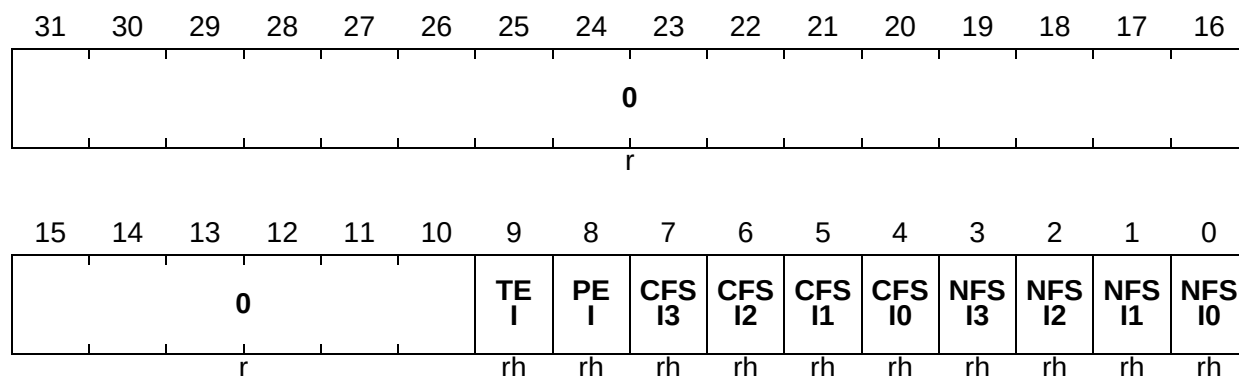
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0						TE IR	PE IR	CFS IR3	CFS IR2	CFS IR1	CFS IR0	NFS IR3	NFS IR2	NFS IR1	NFS IR0
r						w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0						TE IE	PE IE	CFS IE3	CFS IE2	CFS IE1	CFS IE0	NFS IE3	NFS IE2	NFS IE1	NFS IE0
r						rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Field	Bits	Type	Description
NFSIE _x , x = 0, 1, 2, 3	[3:0]	rw	Normal Frame Sent in Pipe x Interrupt Enable If set to one the interrupt NFSI _x can be generated.
CFSIE _x , x = 0, 1, 2, 3	[7:4]	rw	Command Frame Sent in Pipe x Interrupt Enable If set to one the interrupt CFSI _x can be generated.
PEIE	8	rw	Parity Error Interrupt Enable If set to one the interrupt PEI can be generated.
TEIE	9	rw	Time-out Error Interrupt Enable If set to one the interrupt TEI can be generated.
NFSIR _x , x = 0, 1, 2, 3	[19:16]	w	Normal Frame Sent in Pipe x Flag Reset Writing this bit with 1 resets bit TISR.NFSI _x . Writing a 0 has no effect. A read action always delivers 0.
CFSIR _x , x = 0, 1, 2, 3	[23:20]	w	Command Frame Sent in Pipe x Flag Reset Writing this bit with 1 resets bit TISR.CFSI _x . Writing a 0 has no effect. A read action always delivers 0.
PEIR	24	w	Parity Error or Time-out Error Flag Reset Writing this bit with 1 resets bit TISR.PEI. Writing a 0 has no effect. A read action always delivers 0.
TEIR	25	w	Time-out Error Flag Reset Writing this bit with 1 resets bit TISR.TEI. Writing a 0 has no effect. A read action always delivers 0.
0	[15:10], [31:26]	r	Reserved ; read as 0; should be written with 0.

TISR

Transmitter Interrupt Status Register

Reset Value: 0000 0000_H



Field	Bits	Type	Description
NFSIx	[3:0]	rh	Normal Frame Sent in Pipe x Flag It is set to one when a normal frame has been sent out correctly on pipe x.
CFSIx	[7:4]	rh	Command Frame Sent in Pipe x Flag It is set to one when a command frame has been sent out correctly on pipe x.
PEI	8	rh	Parity Error Flag It is set to one when the parity error counter on transmitter side has reached 0.
TEI	9	rh	Time-out Error Flag It is set to one when the non-acknowledge counter has reached 0 (it is counting down by 1 with each retry due to READY = 1 when VALID becomes 0).
0	[31:10]	r	Reserved ; read as 0; should be written with 0.

TINPR

Transmitter Interrupt Node Pointer Register

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0									PTEIP			0	CFSIP		
r									rw			r	rw		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	NFSIP3			0	NFSIP2			0	NFSIP1			0	NFSIP0		
r	rw			r	rw			r	rw			r	rw		

Field	Bits	Type	Description
NFSIP0	[2:0]	rw	Normal Frame Sent in Pipe 0 Interrupt Pointer Number of the interrupt output reporting the sending of a normal frame through pipe 0 if enabled by NFSIE0 = 1. 000 _B MLI interrupt output 0 is selected. ... 111 _B MLI interrupt output 7 is selected.
NFSIP1	[6:4]	rw	Normal Frame Sent in Pipe 1 Interrupt Pointer Number of the interrupt output reporting the sending of a normal frame through pipe 1 if enabled by NFSIE1 = 1. 000 _B MLI interrupt output 0 is selected. ... 111 _B MLI interrupt output 7 is selected.
NFSIP2	[10:8]	rw	Normal Frame Sent in Pipe 2 Interrupt Pointer Number of the interrupt output reporting the sending of a normal frame through pipe 2 if enabled by NFSIE2 = 1. 000 _B MLI interrupt output 0 is selected. ... 111 _B MLI interrupt output 7 is selected.

Field	Bits	Type	Description
NFSIP3	[14:12]	rw	Normal Frame Sent in Pipe 3 Interrupt Pointer Number of the interrupt output reporting the sending of a normal frame through pipe 3 if enabled by NFSIE3 = 1. 000 _B MLI interrupt output 0 is selected. ... 111 _B MLI interrupt output 7 is selected.
CFSIP	[18:16]	rw	Command Frame Sent Interrupt Pointer Number of the common interrupt output reporting the sending of a command frame for: pipe 0 if enabled by CFSIE0 = 1 or pipe 1 if enabled by CFSIE1 = 1 or pipe 2 if enabled by CFSIE2 = 1 or pipe 3 if enabled by CFSIE3 = 1. 000 _B MLI interrupt output 0 is selected. ... 111 _B MLI interrupt output 7 is selected.
PTEIP	[22:20]	rw	Parity or Time-out Interrupt Pointer Number of the common interrupt output reporting: the parity error if enabled by PEIE = 1 or the time-out interrupt if enabled by TEIE = 1. 000 _B MLI interrupt output 0 is selected. ... 111 _B MLI interrupt output 7 is selected.
0	3, 7, 11, 15, 19, [31:23]	r	Reserved ; read as 0; should be written with 0.

RIER

Receiver Interrupt Enable Register

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0						DRA IR	MPE IR	PE IR	ICE R	CFR IR3	CFR IR2	CFR IR1	CFR IR0	ME IR	NFR IR
r						W	W	W	W	W	W	W	W	W	W
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0						DRA IE	MPE IE	PE IE	ICE	CFR IE3	CFR IE2	CFR IE1	CFR IE0	NFR IE	
r						rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Field	Bits	Type	Description
NFRIE	[1:0]	rw	Normal Frame Received Interrupt Enable This bit field defines if an interrupt is generated when a normal frame is correctly received. 00 _B The interrupt generation is disabled. 01 _B The interrupt is generated each time a normal frame is correctly received. 10 _B The interrupt is generated each time a normal frame is correctly received that is not automatically handled by the MLI move engine. 11 _B reserved
CFRIEx, x = 0, 1, 2, 3	[5:2]	rw	Command Received through Pipe x Interrupt Enable This bit defines if an interrupt is generated when a command frame is correctly received through pipe x. 0 The interrupt generation is disabled. 1 The interrupt generated is enabled.
ICE	6	rw	Interrupt Command Enable This bit defines if an interrupt is generated when a command on pipe 0 requests the interrupt generation. The activated interrupt output line is defined by the command. 0 The interrupt generation is disabled. 1 The interrupt generated is enabled.

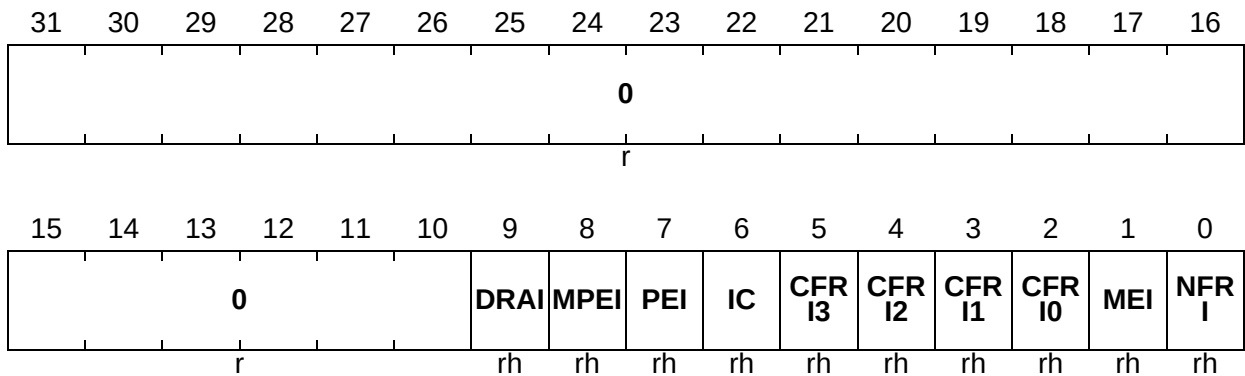
Micro Link Serial Bus Interface

Field	Bits	Type	Description
PEIE	7	rw	Parity Error Interrupt Enable This bit enables the interrupt is generated when a parity error is detected and the parity error counter is 0. 0 The interrupt generation is disabled. 1 The interrupt generation is enabled.
MPEIE	8	rw	Memory Protection Interrupt Enable This bit enables the interrupt is generated when a memory protection error is detected. 0 The interrupt generation is disabled. 1 The interrupt generation is enabled.
DRAIE	9	rw	Discarded Read Answer Interrupt Enable If set to one the interrupt DRAI can be generated.
NFRIR	16	w	Normal Frame Received Interrupt Flag Reset Writing this bit with 1 resets bit RISR.NFRI. Writing a 0 has no effect. A read action always delivers 0.
MEIR	17	w	MLI Move Engine Interrupt Flag Reset Writing this bit with 1 resets bit RISR.MEI. Writing a 0 has no effect. A read action always delivers 0.
CFRIRx, x = 0, 1, 2, 3	[21:18]	w	Command Frame Received through pipe x Interrupt Flag Reset Writing this bit with 1 resets bit RISR.CFRIx. Writing a 0 has no effect. A read action always delivers 0.
ICER	22	w	Interrupt Command Flag Reset Writing this bit with 1 resets bit RISR.ICE. Writing a 0 has no effect. A read action always delivers 0.
PEIR	23	w	Parity Error Interrupt Flag Reset Writing this bit with 1 resets bit RISR.PEI. Writing a 0 has no effect. A read action always delivers 0.
MPEIR	24	w	Memory Protection Error Interrupt Flag Reset Writing this bit with 1 resets bit RISR.MPEI. Writing a 0 has no effect. A read action always delivers 0.
DRAIR	25	w	Discarded Read Answer Interrupt Flag Reset Writing this bit with 1 resets bit RISR.DRAI. Writing a 0 has no effect. A read action always delivers 0.
0	[15:10] [31:26]	r	Reserved ; read as 0; should be written with 0.

RISR

Receiver Interrupt Status Register

Reset Value: 0000 0000_H

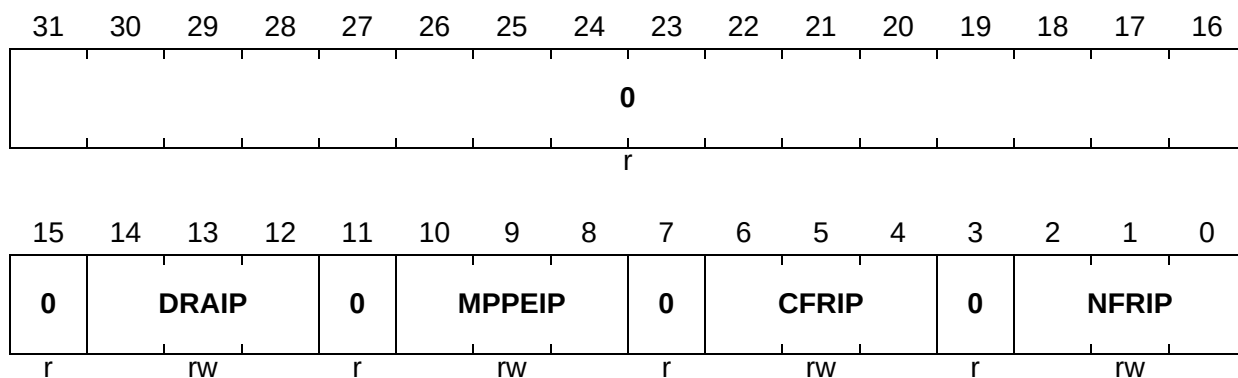


Field	Bits	Type	Description
NFRI	0	rh	Normal Frame Received Interrupt Flag It is set to 1 when received a normal frame.
MEI	1	rh	MLI Move Engine Interrupt Flag It is set when the MLI move engine has done the requested transfer.
CFRI _x , x = 0, 1, 2, 3	[5:2]	rh	Command Frame Received through pipe x Interrupt Flag It is set to 1 when a command frame has been received correctly on pipe x.
IC	6	rh	Interrupt Command Flag It is set to 1 when an interrupt command on pipe 0 requests an interrupt.
PEI	7	rh	Parity Error Interrupt Flag It is set to one when the parity error counter on receiver side has reached 0.
MPEI	8	rh	Memory Protection Error Interrupt Flag It is set to one when detected a not allowed memory access.
DRAI	9	rh	Discarded Read Answer Interrupt Flag It is set to one when the answer to a read command has been discarded, because no read pending flag was set.
0	[31:10]	r	Reserved ; read as 0; should be written with 0.

RINPR

Receiver Interrupt Node Pointer Register

Reset Value: 0000 0000_H



Field	Bits	Type	Description
NFRIP	[2:0]	rw	Normal Frame Received Interrupt Pointer Number of the interrupt output reporting the reception of a normal frame if enabled by NFRIE. 000 _B MLI interrupt output 0 is selected. ... 111 _B MLI interrupt output 7 is selected.
CFRIP	[6:4]	rw	Command Frame Received Interrupt Pointer Number of the common interrupt output reporting the reception of a command frame through: pipe 0 if enabled by CFRIE0 = 1 or pipe 1 if enabled by CFRIE1 = 1 or pipe 2 if enabled by CFRIE2 = 1 or pipe 3 if enabled by CFRIE3 = 1. 000 _B MLI interrupt output 0 is selected. ... 111 _B MLI interrupt output 7 is selected.
MPPEIP	[10:8]	rw	Memory Protection or Parity Error Interrupt Pointer Number of the interrupt output reporting a parity error or a memory protection error on receiver side if enabled by MPEIE = 1 or PEIE=1. 000 _B MLI interrupt output 0 is selected. ... 111 _B MLI interrupt output 7 is selected.

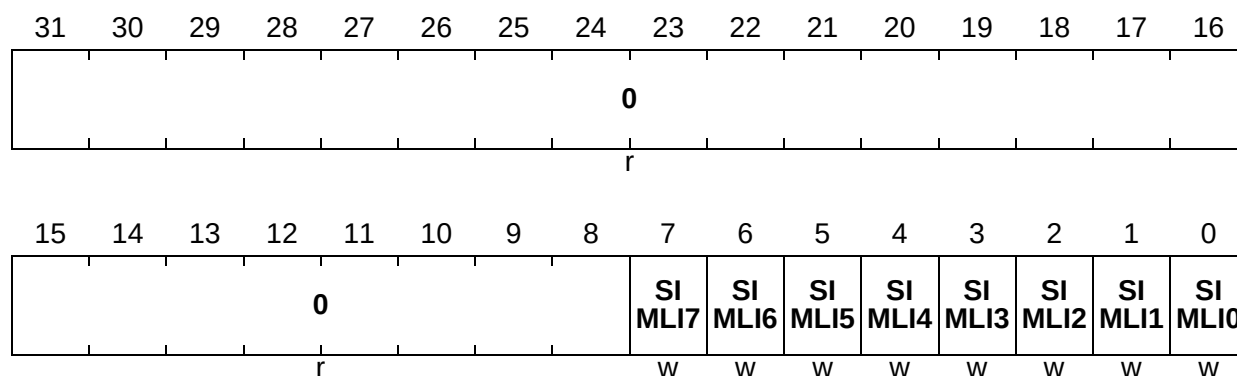
Field	Bits	Type	Description
DRAIP	[14:12]	rw	Discarded Read Answer Interrupt Pointer Number of the interrupt output reporting that a read answer has been discarded if enabled by DRAIE = 1. 000 _B MLI interrupt output 0 is selected. ... 111 _B MLI interrupt output 7 is selected.
0	3, 7, 11, [31:15]	r	Reserved ; read as 0; should be written with 0.

The GINTR can activate the interrupt output lines of the MLI module. The implementation of this register does not involve any flip-flop.

GINTR

Global Interrupt Set Register

Reset Value: 0000 0000_H



Field	Bits	Type	Description
SIMLix (x = 7 - 0)	x	w	Set MLI Interrupt Output Line x 0 No action 1 The MLI interrupt output line x will be activated.
0	[31:8]	r	Reserved ; read as 0; should be written with 0.

5.2.5 Memory Protection Registers

The AER register enables write and read operations in the corresponding address ranges ($x = 0$ to 31). Each address range can be individually enabled.

AER

Access Enable Register

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AEN 31	AEN 30	AEN 29	AEN 28	AEN 27	AEN 26	AEN 25	AEN 24	AEN 23	AEN 22	AEN 21	AEN 20	AEN 19	AEN 18	AEN 17	AEN 16
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AEN 15	AEN 14	AEN 13	AEN 12	AEN 11	AEN 10	AEN 9	AEN 8	AEN 7	AEN 6	AEN 5	AEN 4	AEN 3	AEN 2	AEN 1	AEN 0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Field	Bits	Type	Description
AENx ($x = 31 - 0$)	x	rw	Access Enable This bit enables the read and write capability of the MLI from the address range x. ($x = 31 - 0$). 0 The MLI read and write action to this address range is disabled. 1 The MLI read and write action to this address range is enabled.

Note: The address ranges related to these bits are described in the implementation section for this module. These registers are ENDINIT-protected.

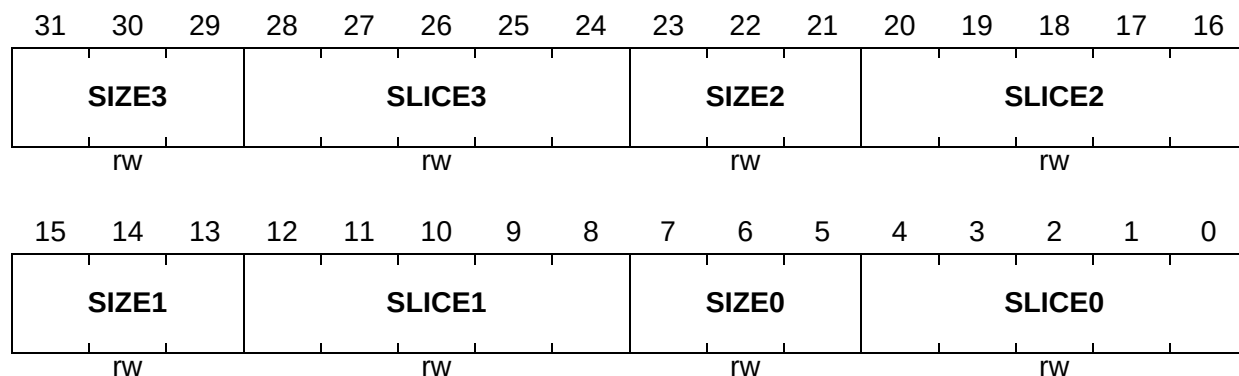
Micro Link Serial Bus Interface

The ARR register selects the address range (only for internal memories, not for modules) that can be accessed by the MLI if the corresponding address range is enabled.

ARR

Access Range Register

Reset Value: 0000 0000_H



Field	Bits	Type	Description
SLICE0, SLICE1, SLICE2, SLICE3	[4:0], [12:8], [20:16], [28:24]	rw	Address Slice x (x = 0, 1, 2, 3) This bit field defines which part of the memory address range x can be accessed by the MLI, if enabled.
SIZE0, SIZE1, SIZE2, SIZE3	[7:5], [15:13], [23:21], [31:29]	rw	Address Size x (x = 0, 1, 2, 3) This bit field defines which size of the memory address range x can be accessed by the MLI, if enabled.

Note: The address ranges related to these bits are described in the implementation section for this module. These registers are ENDINIT-protected.

5.3 MLI0 Module Implementation

This section describes the MLI0 Module interface with the clock control, port connections, interrupt control, and address decoding.

5.3.1 Interface of the MLI Module

5.3.1.1 Port Connections of MLI0

Figure 5-47 shows the TC1100 specific implementation details and interconnections of the MLI0 module. It supplied by a separate clock control, interrupt control, address decoding, and port control logic.

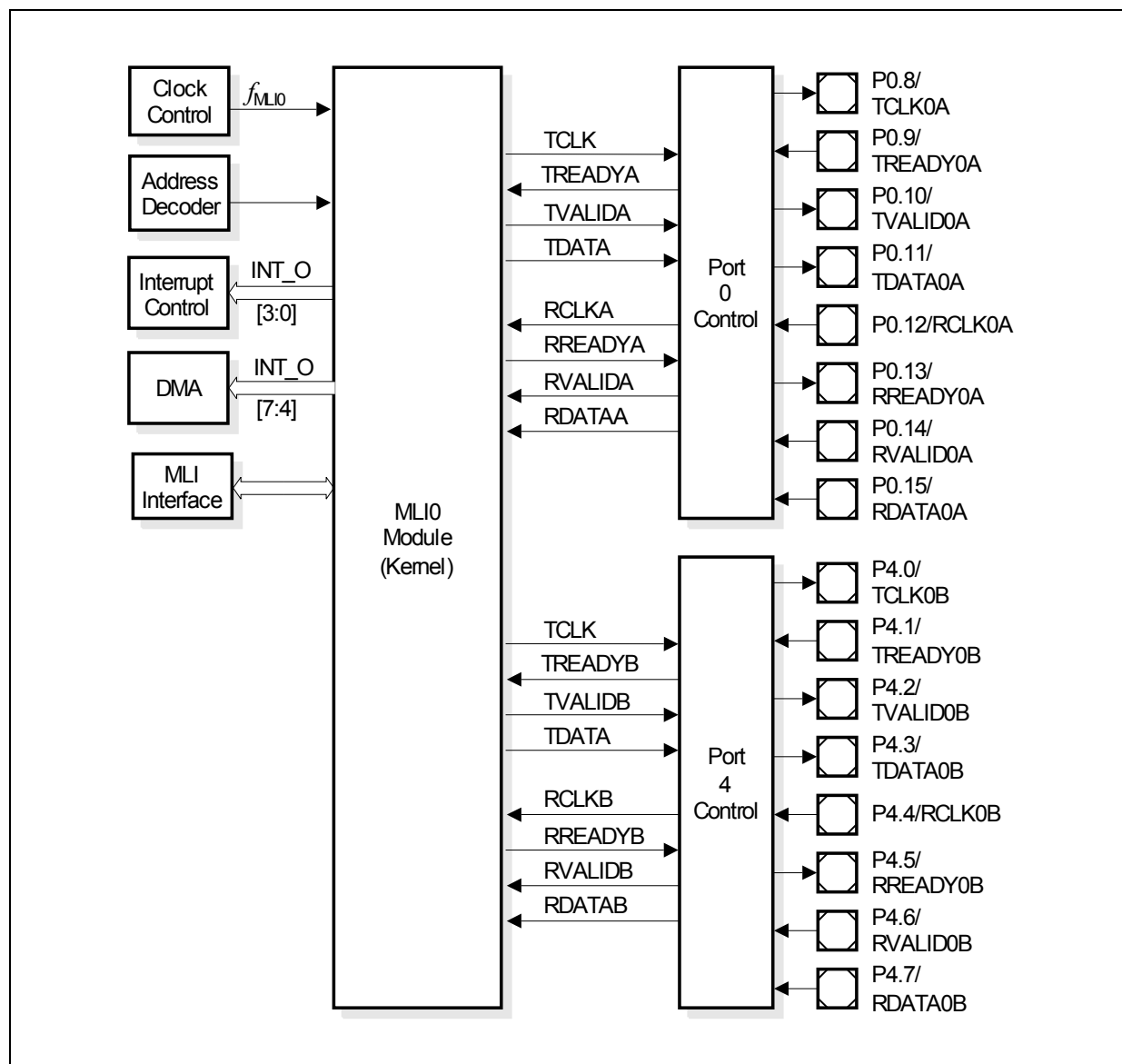


Figure 5-47 MLI0 Module Implementation and Interconnections

Table 5-27 MLI0 Signal Connections

MLI0 Signals	Input/ Output	Connected to
TCLK	O	port P0.8 (named TCLK0A) port P4.0 (named TCLK0B) RCLKD of MLI0 (loop back mode)
TREADYA	I	port P0.9 (named TREADY0A)
TREADYB	I	port P4.1 (named TREADY0B)
TREADYC	I	0
TREADYD	I	RREADYD of MLI0 (loop back mode)
TVALIDA	O	port P0.10 (named TVALID0A)
TVALIDB	O	port P4.2 (named TVALID0B)
TVALIDC	O	left open
TVALIDD	O	RVALIDD of MLI0 (loop back mode)
TDATA	O	port P0.11 (named TDATA0A) port P4.3 (named TDATA0B) RDATA0 of MLI0 (loop back mode)
RCLKA	I	port P0.12 (named RCLK0A)
RCLKB	I	port P4.4 (named RCLK0B)
RCLKC	I	0
RCLKD	I	TCLK of MLI0 (loop back mode)
RREADYA	O	port P0.13 (named RREADY0A)
RREADYB	O	port P4.5 (named RREADY0B)
RREADYC	O	left open
RREADYD	O	TREADYD of MLI0 (loop back mode)
RVALIDA	I	port P0.14 (named RVALID0A)
RVALIDB	I	port P4.6 (named RVALID0B)
RVALIDC	I	0
RVALIDD	I	TVALIDD of MLI0 (loop back mode)
RDATAA	I	port P0.15 (named RDATA0A)
RDATAB	I	port P4.7 (named RDATA0B)
RDATAC	I	0
RDATAD	I	TDATA of MLI0 (loop back mode)

5.3.2 Access Protection

The table below shows the address ranges covered by the access protection (for read and write) of the MLI module. The access enable bits are located in the register AER. The bit at the bit position x in this register is related to the address range x in [Table 5-28](#). Some bits can cover several address ranges (cluster of modules). In the case that a read or a write access has been requested with the corresponding enable bit =0, an error interrupt is generated.

Table 5-28 DMA Access Protection Address Ranges

Range Number	Related Enable Bits	Covered Address Range	Corresponding to Modules
x = 0	ME0AENR.AEN0	F000 0000 _H to F000 00FF _H F010 C200 _H to F010 C2FF _H	SCU, incl. WDT, MEMCHK
x = 1	ME0AENR.AEN1	F000 0100 _H to F000 01FF _H	SBCU
x = 2	ME0AENR.AEN2	F000 0200 _H to F000 02FF _H	STM
x = 3	ME0AENR.AEN3	F000 0300 _H to F000 03FF _H	OCDS
x = 4	ME0AENR.AEN4	F000 0600 _H to F000 06FF _H	GPTU
x = 5	ME0AENR.AEN5	F000 0C00 _H to F000 10FF _H	P0, P1, P2, P3,P4
x = 6	ME0AENR.AEN6	F000 3C00 _H to F000 3EFF _H	DMA
x = 7	ME0AENR.AEN7	-	-
x = 8	ME0AENR.AEN8	F010 0600 _H to F010 06FF _H	IIC
x = 9	ME0AENR.AEN9	-	-
x = 10	ME0AENR.AEN10	F000 2100 _H to F000 21FF _H	CCU1
x = 11	ME0AENR.AEN11	-	-
x = 12	ME0AENR.AEN12	-	-
x = 13	ME0AENR.AEN13	F010 0100 _H to F010 01FF _H	SSC0
x = 14	ME0AENR.AEN14	F010 0200 _H to F010 02FF _H	SSC1
x = 15	ME0AENR.AEN15	F010 0300 _H to F010 03FF _H	ASC0
x = 16	ME0AENR.AEN16	F010 0200 _H to F010 02FF _H	ASC1
x = 17	ME0AENR.AEN17	-	-
x = 18	ME0AENR.AEN18	-	-
x = 19	ME0AENR.AEN19	F010 C000 _H to F010 C0FF _H F01E 0000 _H to F01E 7FFF _H F020 0000 _H to F023 FFFF _H	MLIO Module, MLIO Small TWs, MLIO Large TWs
x = 20	ME0AENR.AEN20	-	-

Table 5-28 DMA Access Protection Address Ranges (cont'd)

Range Number	Related Enable Bits	Covered Address Range	Corresponding to Modules
x = 21	ME0AENR.AEN21	F7E0 FF00 _H to F7E0 FFFF _H F7E1 0000 _H to F7E1 FFFF _H F800 0400 _H to F87F FFFF _H	CPS, MMU, CPU SFRs & GPRs, DMU, DMI, PMI, LBCU, LFI
x = 22	ME0AENR.AEN22	F800 000 _H to F800 03FF _H	EBU
x = 23	ME0AENR.AEN23	-	-
x = 24	ME0AENR.AEN24	8000 0000 _H to 8FFF FFFF _H A000 0000 _H to AFFF FFFF _H	Ext. EBU Space
x = 25	ME0AENR.AEN25	-	-
x = 26	ME0AENR.AEN26	-	-
x = 27	ME0AENR.AEN27	D800 0000 _H to DEFF FFFF _H E000 0000 _H to E7FF FFFF _H	External Peripherals & External Emulator
x = 28	ME0AENR.AEN28	DFFF C000 _H to DFFFFFFF _H	Boot ROM
x = 29	ME0AENR.AEN29	E800 0000 _H to E83F FFFF _H	DMU Image (E80x translated to C00x)
x = 30	ME0AENR.AEN30	E840 0000 _H to E84F FFFF _H	DMI Image (E84x translated to D00x)
x = 31	ME0AENR.AEN31	E850 0000 _H to E85F FFFF _H	PMI Image (E85x translated to D40x)

The internal memory blocks are protected by an address range verification in addition to the access enable bits. The address range verification is based on the bit fields SIZE_x and SLICE_x, which are located in the registers ARR. Only accesses targeting the enabled (by the corresponding AEN bit) and selected memory area (value given by SIZE_x, SLICE_x) can be handled automatically. Automatic accesses to other locations are not supported.

In order to keep a similar structure for the access protection structure, the assignment is kept identical, although not always the complete range will be available (depending on the memory implementation of different devices).

The address ranges described by SLICE_x and SIZE_x are defined as follows:

- SIZE₀, SLICE₀:
8 KBytes address range from F010 A000_H to F010 BFFF_H,
this range is unused in TC1100

- **SIZE1, SLICE1:**
64 KBytes address range from E800 0000_H to E800 FFFF_H,
with address translation from E800 to C000, covering the internal SRAM area
- **SIZE2, SLICE2:**
64 KBytes address range from E840 0000_H to E840 FFFF_H,
with address translation from E800 to D000, covering the DMI RAM area
- **SIZE3, SLICE3:**
address range reserved for future extensions,
this range is unused in TC1100

For the internal SRAM (with address translation from E800 to C000 in the LFI):

Table 5-29 SRAM Read/Write Address Range Verification

Bit Field SIZE1	Size of the Available Address Slice	Bit Field SLICE1	Available Address Range
000	512 Bytes	00000 00001 ... 11111	E800 0000 _H to E800 01FF _H E800 0200 _H to E800 03FF _H E800 3E00 _H to E800 3FFF _H
001	1 KByte	00000 00001 ... 11111	E800 0000 _H to E800 03FF _H E800 0400 _H to E800 07FF _H E800 7C00 _H to E800 7FFF _H
010	2 KBytes	00000 00001 ... 11111	E800 0000 _H to E800 07FF _H E800 0800 _H to E800 0FFF _H E800 F800 _H to E800 FFFF _H
011	4 KBytes	X0000 X0001 ... X1111	E800 0000 _H to E800 0FFF _H E800 1000 _H to E800 1FFF _H E800 F000 _H to E800 FFFF _H
100	8 KBytes	XX000 XX001 ... XX111	E800 0000 _H to E800 1FFF _H E800 2000 _H to E800 3FFF _H E800 E000 _H to E800 FFFF _H
101	16 KBytes	XXX00 XXX01 XXX10 XXX11	E800 0000 _H to E800 3FFF _H E800 4000 _H to E800 7FFF _H E800 8000 _H to E800 BFFF _H E800 C000 _H to E800 FFFF _H

Table 5-29 SRAM Read/Write Address Range Verification (cont'd)

Bit Field SIZE1	Size of the Available Address Slice	Bit Field SLICE1	Available Address Range
110	32 KBytes	XXXX0 XXXX1	E800 0000 _H to E800 7FFF _H E800 8000 _H to E800 FFFF _H
111	64 KBytes	XXXXX	E800 0000 _H to E800 FFFF _H

For the internal DMI SPRAM (with address translation from E840 to D000 in the LFI):

Table 5-30 DMI RAM Read/Write Address Range Verification

Bit Field SIZE2	Size of the Available Address Slice	Bit Field SLICE2	Available Address Range
000	512 Bytes	00000 00001 ... 11111	E840 0000 _H to E840 01FF _H E840 0200 _H to E840 03FF _H E840 3E00 _H to E840 3FFF _H
001	1 KByte	00000 00001 ... 11111	E840 0000 _H to E840 03FF _H E840 0400 _H to E840 07FF _H E840 7C00 _H to E840 7FFF _H
010	2 KBytes	00000 00001 ... 01111 1XXXX	E840 0000 _H to E840 07FF _H E840 0800 _H to E840 0FFF _H E840 7C00 _H to E840 7FFF _H Not Valid
011	4 KBytes	X0000 X0001 ... X0111 X1XXX	E840 0000 _H to E840 0FFF _H E840 1000 _H to E840 1FFF _H E840 7000 _H to E840 7FFF _H Not Valid
100	8 KBytes	XX000 XX001 XX010 XX011 XX1XX	E840 0000 _H to E840 1FFF _H E840 2000 _H to E840 3FFF _H E840 4000 _H to E840 5FFF _H E840 6000 _H to E840 7FFF _H Not Valid

Table 5-30 DMI RAM Read/Write Address Range Verification (cont'd)

Bit Field SIZE2	Size of the Available Address Slice	Bit Field SLICE2	Available Address Range
101	16 KBytes	XXX00 XXX01 XXX1X	E840 0000 _H to E840 3FFF _H E840 4000 _H to E840 7FFF _H Not Valid
110	32 KBytes	XXXX0 XXXX1	E840 0000 _H to E840 7FFF _H Not Valid
111	64 KBytes	XXXXX	E840 0000 _H to E840 7FFF _H E840 8000 _H to E840 FFFF _H Not Valid

5.3.3 MLI0 Module Related External Registers

Figure 5-48 summarizes the module related external registers which are required for MLI0 programming (see also **Figure 5-46** for the module kernel specific registers).

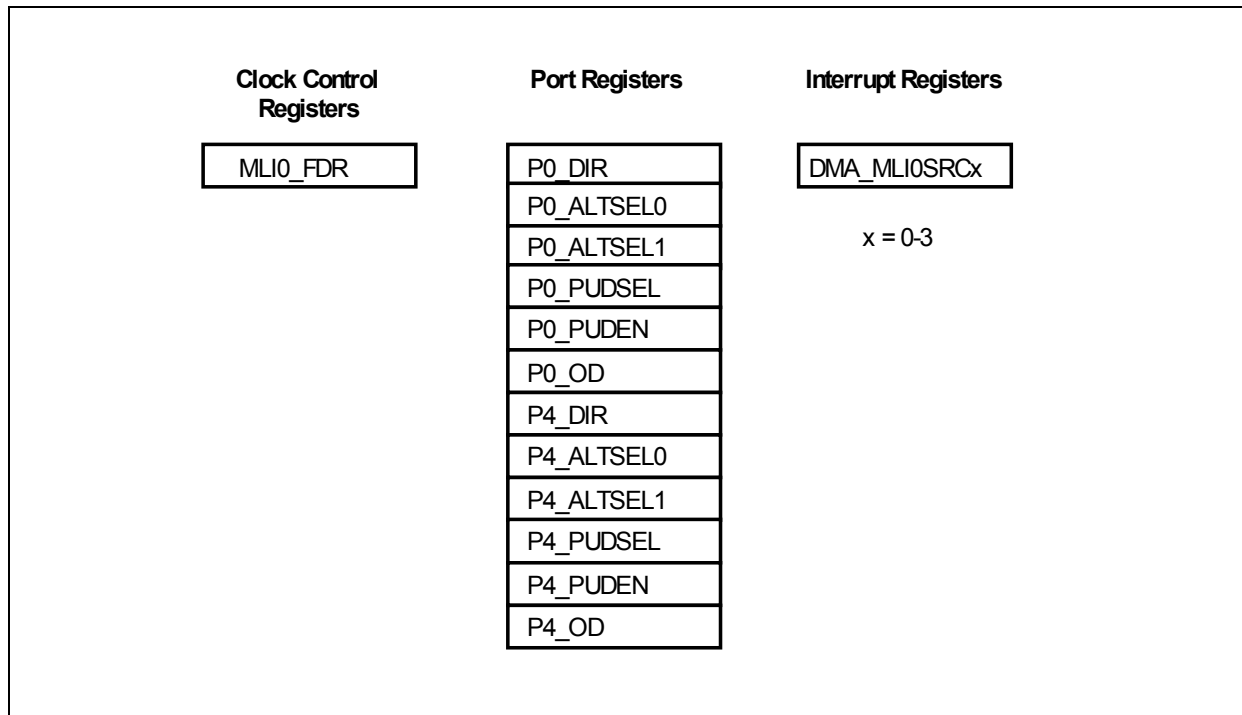


Figure 5-48 MLI0 Implementation Specific Special Function Registers

Note: In the current version the four lines trigger_command and their related registers are not implemented. Further, the interrupt registers are described in the chapter “Direct Memory Access Controller” of the TC1100 System Units User’s Manual.

5.3.3.1 DMA Requests

The MLI interface provides DMA request output lines to the DMA controller. These are the interrupt output lines INT_O[7:4] for MLI0.

5.3.3.2 Interrupt Registers

The MLI module has eight interrupt request outputs. For MLI0, the lower four lines (INT_O[3:0]) are connected to service request nodes. These registers are described in the chapter “Direct Memory Access Controller” of the *TC1100 System Units User’s Manual*.

5.3.3.3 Fractional Divider Registers

The fractional divider register allows the programmer to control the clock rate and period of the 50% duty cycle shift clock f_{MLI} . The period of f_{MLI} can be either 1/STEP or a

Micro Link Serial Bus Interface

fraction of STEP/1024 (for any value of STEP from 0 to 1023) of clock f_{MLI} . The MLI has its own fractional divider.

MLI0_FDR

MLI0 Fractional Divider Register

Reset Value: 03FF 43FF_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DIS CLK	EN HW	SUS REQ	SUS ACK	0	RESULT										
rwh	rw	rh	rh	r	rh										
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DM		SC		SM	0	STEP									
rw		rw		rw	r	rw									

Field	Bits	Type	Description
STEP	[9:0]	rw	Step Value Reload or addition value for RESULT.
SM	11	rw	Suspend Mode 0 Granted suspend mode. 1 Immediate suspend mode.
SC	[13:12]	rw	Divider Mode This bit field selects normal divider mode or fractional divider mode.
DM	[15:14]	rw	Suspend Control This bit field defines the behavior of the fractional divider in suspend mode.
RESULT	[25:16]	rh	Result Value Bit fields for the addition result.
SUSACK	28	rh	Suspend Mode Acknowledge Indicates state of SPNDACK signal.
SUSREQ	29	rh	Suspend Mode Request Indicates state of SPND signal.
ENHW	30	rw	Enable Hardware Clock Control Controls operation of ECEN input and DISCLK bit.
DISCLK	31	rwh	Disable Clock Hardware controlled disable for f_{OUT} signal.
0	10, [27:26]	r	Reserved ; read as 0; should be written with 0

5.3.3.4 Port Control

The interconnections between the MLI module and the port I/O lines are controlled in the port logics. The following port control operation selections must be executed (additionally to the PISEL programming):

- Input/output direction selection (DIR registers)
- Alternate function selection (ALTSEL0 and ALTSEL1 registers)
- Input/Output driver characteristic control (PUDSEL, PUDEN and OD registers)

The port input/output control registers contain the bit fields that select the digital output and input driver characteristics such as pull-up/down devices, port direction (input/output), open-drain, and alternate output selections. The I/O lines for the MLI module are controlled by the port input/output control registers of Port 0, Port 3 and Port 4.

Table 5-31 shows how bits and bit fields must be programmed for the required I/O functionality of the MLI I/O lines.

Table 5-31 MLI0 I/O Line Selection and Setup

Module	Port Lines	Input/Output Control Register Bits	I/O
MLI0	P0.8/TCLK0A	P0_DIR.P8 = 1 _B	Output
		P0_ALTSEL0.P8 = 0 _B	
		P0_ALTSEL1.P8 = 1 _B	
	P0.9/TREADY0A	P0_DIR.P9 = 0 _B	Input
	P0.10/TVALID0A	P0_DIR.P10 = 1 _B	Output
		P0_ALTSEL0.P10 = 0 _B	
		P0_ALTSEL1.P10 = 1 _B	
	P0.11/TDATA0A	P0_DIR.P11 = 1 _B	Output
		P0_ALTSEL0.P11 = 0 _B	
		P0_ALTSEL1.P11 = 1 _B	
	P0.12/RCLK0A	P0_DIR.P12 = 0 _B	Input
	P0.13/RREADY0A	P0_DIR.P13 = 1 _B	Output
		P0_ALTSEL0.P13 = 0 _B	
		P0_ALTSEL1.P13 = 1 _B	
	P0.14/RVALID0A	P0_DIR.P14 = 0 _B	Input
	P0.15/RDATA0A	P0_DIR.P15 = 0 _B	Input
	P4.0/TCLK0B	P4_DIR.P0 = 1 _B	Output
		P4_ALTSEL0.P0 = 0 _B	
		P4_ALTSEL1.P0 = 1 _B	
	P4.1/TREADY0B	P4_DIR.P1 = 0 _B	Input
	P4.2/TVALID0B	P4_DIR.P2 = 1 _B	Output
		P4_ALTSEL0.P2 = 0 _B	
		P4_ALTSEL1.P2 = 1 _B	
	P4.3/TDATA0B	P4_DIR.P3 = 1 _B	Output
		P4_ALTSEL0.P3 = 0 _B	
		P4_ALTSEL1.P3 = 1 _B	
	P4.4/RCLK0B	P4_DIR.P4 = 0 _B	Input

Table 5-31 MLI0 I/O Line Selection and Setup (cont'd)

Module	Port Lines	Input/Output Control Register Bits	I/O
	P4.5/RREADY0B	P4_DIR.P5 = 1 _B	Output
		P4_ALTSEL0.P5 = 0 _B	
		P4_ALTSEL1.P5 = 1 _B	
	P4.6/RVALID0B	P4_DIR.P6 = 0 _B	Input
	P4.7/RDATA0B	P4_DIR.P7 = 0 _B	Input

P0_DIR

Port 0 Direction Register

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
P15	P14	P13	P12	P11	P10	P9	P8	P7	P6	P5	P4	P3	P2	P1	P0
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

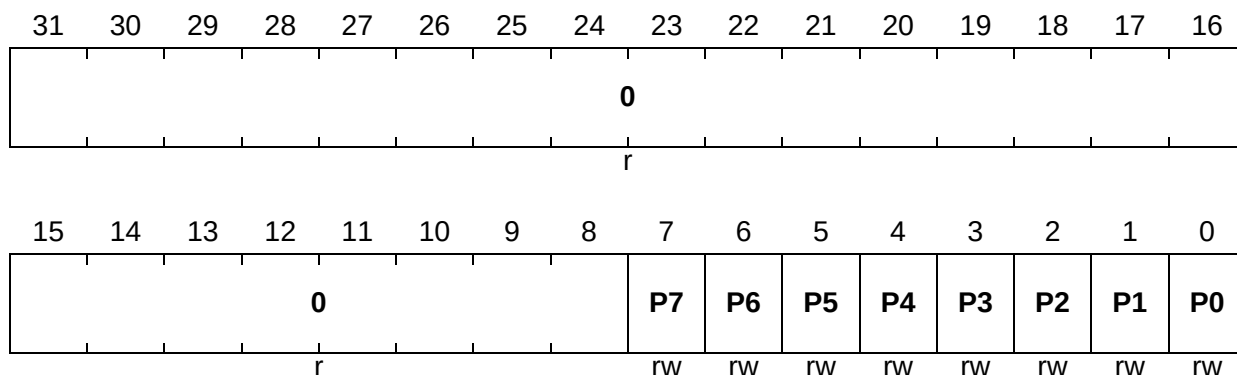
Field	Bits	Type	Description
Pn (n = 8-15)	n	rW	Port 0 Pin 8 - 15 Direction Control ¹⁾ 0 Direction is set to input (default after reset) 1 Direction is set to output
0	[31:16]	r	Reserved ; read as 0; should be written with 0.

¹⁾ Shaded bits and bit field are don't care for MLI I/O port control

P4_DIR

Port 4 Direction Register

Reset Value: 0000 0000_H



Field	Bits	Type	Description
Pn (n = 0-7)	n	rw	Port 0 Pin 0 - 7 Direction Control 0 Direction is set to input (default after reset) 1 Direction is set to output
0	[31:8]	r	Reserved ; read as 0; should be written with 0.

P0_ALTSELn (n = 1, 0)

Port 0 Alternate Select Register

Reset Value: 0000 0000_H

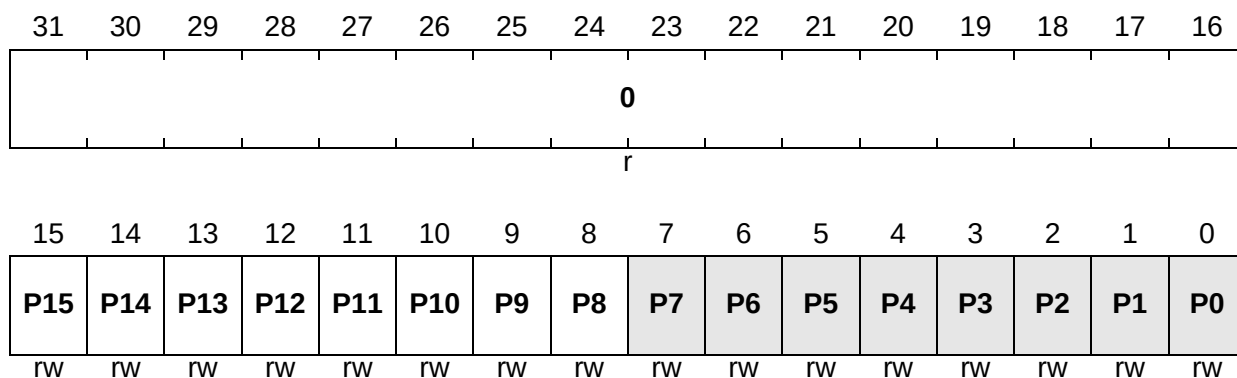


Table 5-32 Function of the Bits P0_ALTSEL0.Pn and P0_ALTSEL1.Pn (n=8-15)¹⁾

P0_ALTSEL0.Pn	P0_ALTSEL1.Pn	Function
0	1	Alternate Select2

¹⁾ Shaded bits and bit field are don't care for MLI I/O port control

P4_ALTSELn (n = 1, 0)

Port 4 Alternate Select Register

Reset Value: 0000 0000_H

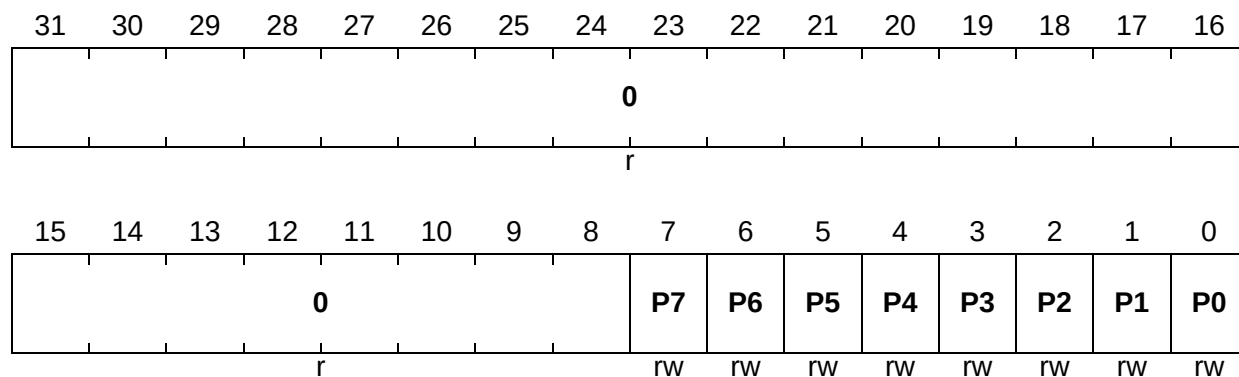


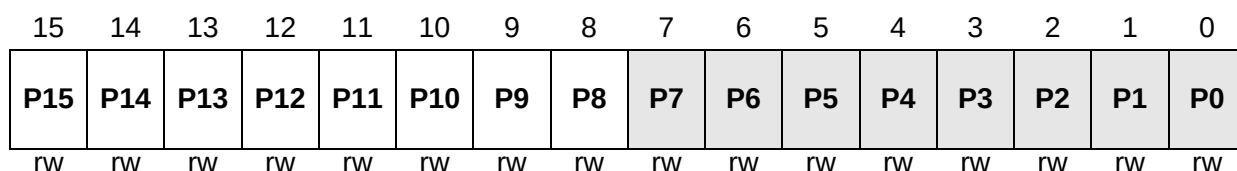
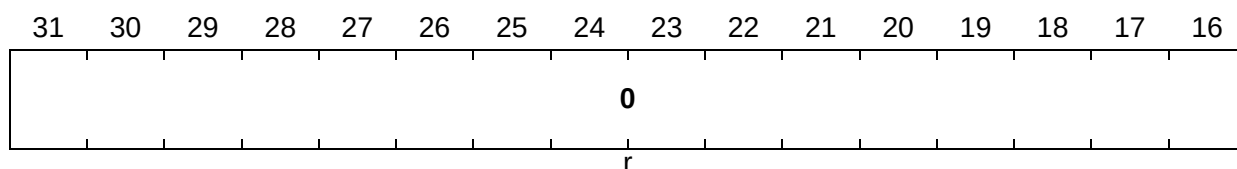
Table 5-33 Function of the Bits P4_ALTSEL0.Pn and P4_ALTSEL1.Pn (n=0-7)

P4_ALTSEL0.Pn	P4_ALTSEL1.Pn	Function
0	1	Alternate Select2

P0_OD

Port 0 Open Drain Control Register

Reset Value: 0000 0000_H



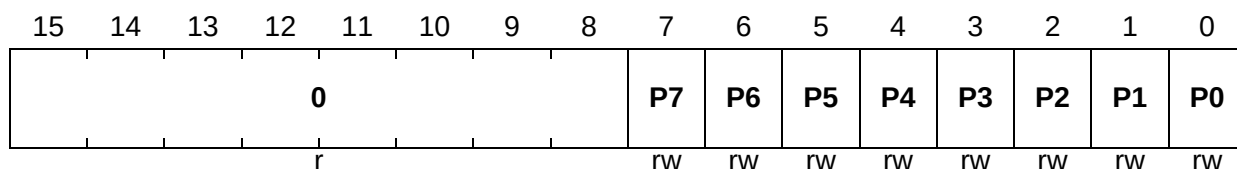
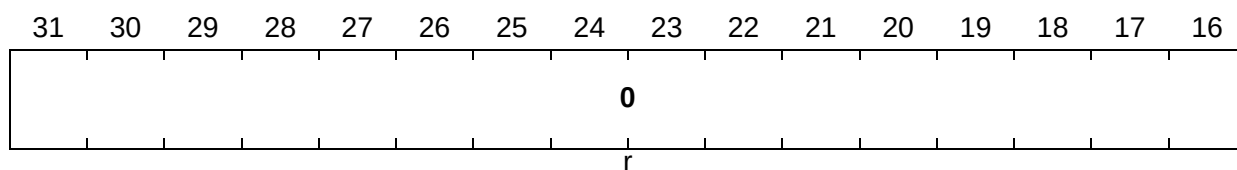
Field	Bits	Type	Description
Pn (n = 8-15)	n	r	Port 0 Pin n Open Drain Mode¹⁾ 0 Normal Mode, output is actively driven for 0 and 1 state 1 Open Drain Mode, output is actively driven only for 0 state
0	[31:16]	r	Reserved ; read as 0; should be written with 0.

¹⁾ Shaded bits and bit field are don't care for MLI I/O port control

P4_OD

Port 4 Open Drain Control Register

Reset Value: 0000 0000_H



Field	Bits	Type	Description
Pn (n = 0-7)	n	rw	Port 4 Pin n Open Drain Mode 0 Normal Mode, output is actively driven for 0 and 1 state 1 Open Drain Mode, output is actively driven only for 0 state
0	[31:16]	r	Reserved ; read as 0; should be written with 0.

P0_PUDSEL

Port 0 Pull-Up/Pull-Down Select Register

Reset Value: 0000 FFFF_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
P15	P14	P13	P12	P11	P10	P9	P8	P7	P6	P5	P4	P3	P2	P1	P0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Field	Bits	Type	Description
Pn (n = 8-15)	n	rw	Pull-Up/Pull-Down Select Port 0 Bit n¹⁾ 0 Pull-down device is selected 1 Pull-up device is selected
0	[31:16]	r	Reserved ; read as 0; should be written with 0.

¹⁾ Shaded bits and bit field are don't care for MLI I/O port control

P4_PUDSEL

Port 4 Pull-Up/Pull-Down Select Register

Reset Value: 0000 00FF_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
r															

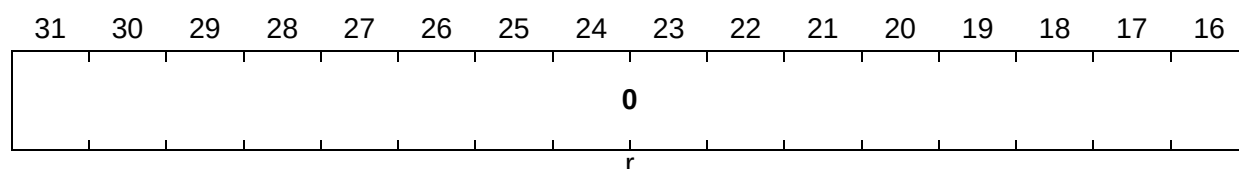
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0								P7	P6	P5	P4	P3	P2	P1	P0
r								rw	rw	rw	rw	rw	rw	rw	rw

Field	Bits	Type	Description
Pn (n = 0-7)	n	rw	Pull-Up/Pull-Down Select Port 4 Bit n 0 Pull-down device is selected 1 Pull-up device is selected
0	[31:8]	r	Reserved ; read as 0; should be written with 0.

P0_PUDEN

Port 0 Pull-Up/Pull-Down Enable Register

Reset Value: 0000 FFFF_H



15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
P15	P14	P13	P12	P11	P10	P9	P8	P7	P6	P5	P4	P3	P2	P1	P0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

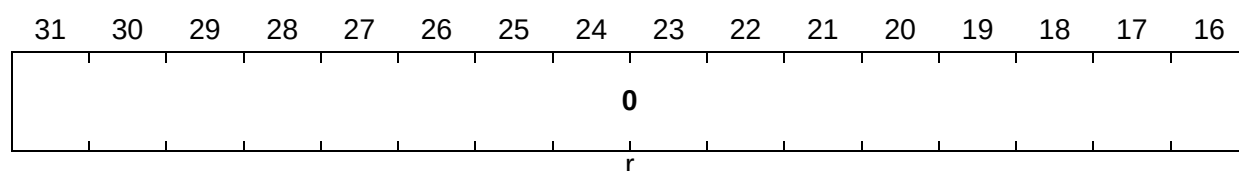
Field	Bits	Type	Description
Pn (n = 8-15)	n	rw	Pull-Up/Pull-Down Enable at Port 0 Bit n¹⁾ 0 Pull-up or Pull-down device is disabled 1 Pull-up or Pull-down device is enabled
0	[31:16]	r	Reserved ; read as 0; should be written with 0.

¹⁾ Shaded bits and bit field are don't care for ASC I/O port control

P4_PUDEN

Port 4 Pull-Up/Pull-Down Enable Register

Reset Value: 0000 00FF_H



15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0								P7	P6	P5	P4	P3	P2	P1	P0
r								rw	rw	rw	rw	rw	rw	rw	rw

Micro Link Serial Bus Interface

Field	Bits	Type	Description
Pn (n = 0-7)	n	rw	Pull-Up/Pull-Down Enable at Port 4 Bit n 0 Pull-up or Pull-down device is disabled 1 Pull-up or Pull-down device is enabled
0	[31:8]	r	Reserved ; read as 0; should be written with 0.

5.3.4 MLI0 Register Address Ranges

In the TC1100, the registers of the MLI module are located in the following address ranges:

- MLI0 module:

Module Base Address	= F010 C000 _H
Module End Address	= F010 C0FF _H
Small Transfer Windows (8 KBytes max.):	
Pipe 0	= F01E 0000 _H - F01E 1FFF _H
Pipe 1	= F01E 2000 _H - F01E 3FFF _H
Pipe 2	= F01E 4000 _H - F01E 5FFF _H
Pipe 3	= F01E 6000 _H - F01E 7FFF _H
Large Transfer Windows (64 KBytes max.):	
Pipe 0	= F020 0000 _H - F020 FFFF _H
Pipe 1	= F021 0000 _H - F021 FFFF _H
Pipe 2	= F022 0000 _H - F022 FFFF _H
Pipe 3	= F023 0000 _H - F023 FFFF _H

- Absolute Register Address = Module Base Address + Offset Address
(offset addresses see [Table 5-26](#))

Note: The complete and detailed address map of the MLI0 module is described in the chapter “Register Overview” of the TC1100 System Units User’s Manual.

6 General Purpose Timer Unit (GPTU)

This chapter describes the General Purpose Timer Units GPTU of the TC1100. The information is presented in the following sections:

- Functional description of the GPTU Kernel (see [Section 6.1](#))
- Register descriptions of all GPTU Kernel specific registers (see [Section 6.2](#))
- TC1100 implementation specific details and registers of the GPTU (port connections and control, interrupt control, address decoding, clock control, see [Section 6.3](#)).

Note: The GPTU kernel register names described in [Section 6.2](#) will be referenced in other parts of the TC1100 User's Manual with the module name prefix "GPTU_" for the GPTU module.

6.1 GPTU Kernel Description

Figure 6-1 shows a global view of all functional blocks of the GPTU kernel and its interfaces.

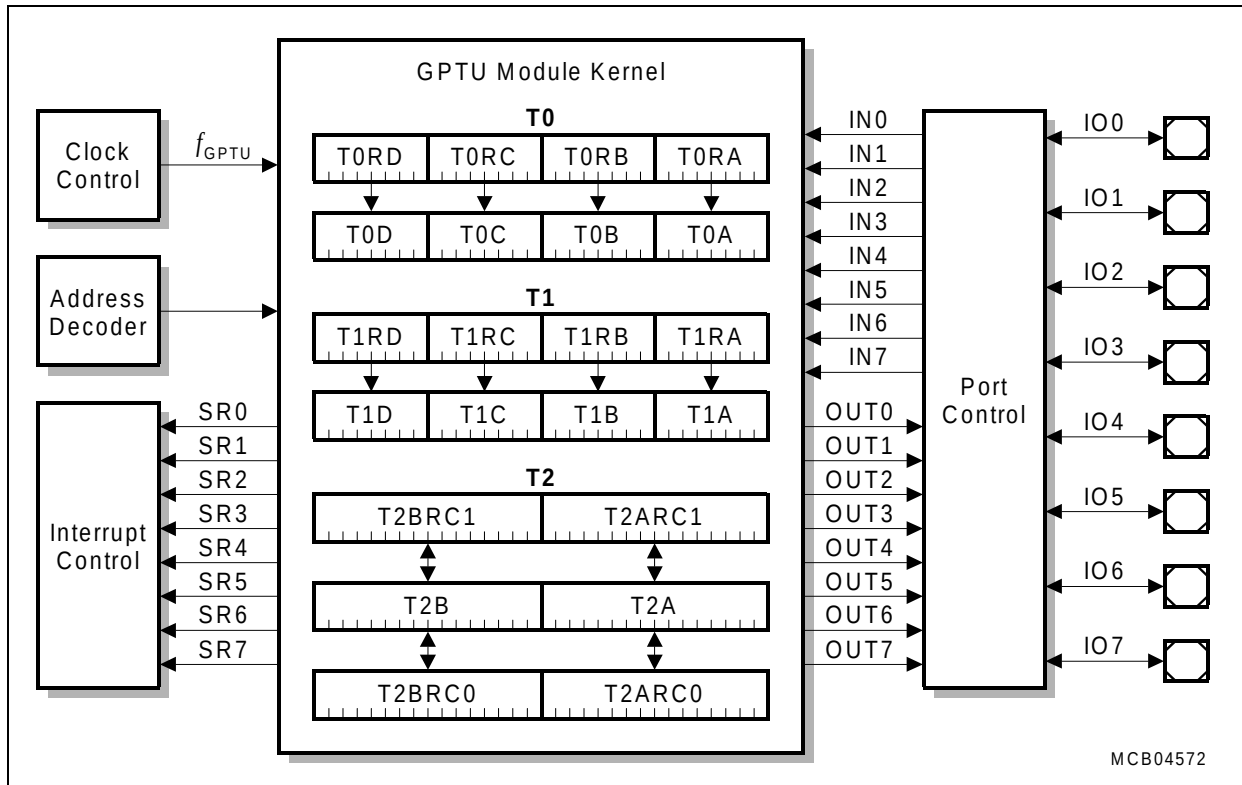


Figure 6-1 General Block Diagram of the GPTU Interface

The GPTU consists of three 32-bit timers designed to solve such application tasks as event timing, event counting, and event recording. The GPTU communicates with the external world via eight inputs and eight outputs concatenated in the port control logic to eight I/O pins **IO[7:0]**. The input signals coming from the port logic are named **IN[7:0]**, and the output signals going to the port logic are named **OUT[7:0]**. These signals are used in the further descriptions of the timers. Further, the GPTU can generate eight service requests **SR[7:0]** within the TC1100.

Clock control, address decoding, and interrupt service request control are managed outside the GPTU Module kernel.

6.1.1 Operational Overview

The GPTU consists of the timers T0, T1, and T2. The functionality of the timers T0 and T1 differs from that of T2.

Additional features of timers T0 and T1 include:

- Each timer has a dedicated 32-bit reload register with automatic reload on overflow
- Timers can be split into individual 8-, 16-, or 24-bit timers with individual reload registers
- T0 and T1 can be concatenated to form one 64-bit timer
- Events generated in T0 or T1 can be used to trigger actions in T2
- Overflow signals can be selected to generate service requests, pin output signals, and T2 trigger events
- Two input pins can define a count option

Additional features of Timer T2 include:

- Up or down Count
- Operating modes:
 - Timer
 - Counter
 - Quadrature counter (incremental/phase encoded counter interface)
- Options:
 - External start/stop, one-shot operation, timer clear on external event
 - Count direction control through software or an external event
 - Two 32-bit reload/capture registers
- Reload modes:
 - Reload on overflow or underflow
 - Reload on external event: positive transition, negative transition, or both transitions
- Capture modes:
 - Capture on external event: positive transition, negative transition, or both transitions
 - Capture and clear timer on external event: positive transition, negative transition, or both transitions
- Can be split into two 16-bit counter/timers
- Timer count, reload, capture, and trigger functions can be assigned to input pins. T0 and T1 overflow events can also be assigned to these functions
- Overflow and underflow signals can be used to trigger T0 and/or T1 and to toggle output pins
- T2 events are freely assignable to the service request nodes

6.1.2 Functional Overview

6.1.2.1 Timers T0 and T1

Figure 6-2 and **Figure 6-3** show detailed block diagrams of Timers T0 and T1. Both, T0 and T1, consist of four 8-bit timer blocks named TxA, TxB, TxC, and TxD ($x = 0, 1$). Each eight-bit timer block contains a count register and a reload register. These blocks can be configured to run independently as 8-bit timers, or can be concatenated to form wider timers (16-bit, 24-bit, or 32-bit). A cross-connection between T0 and T1 extends these options to permit creation of a 64-bit timer.

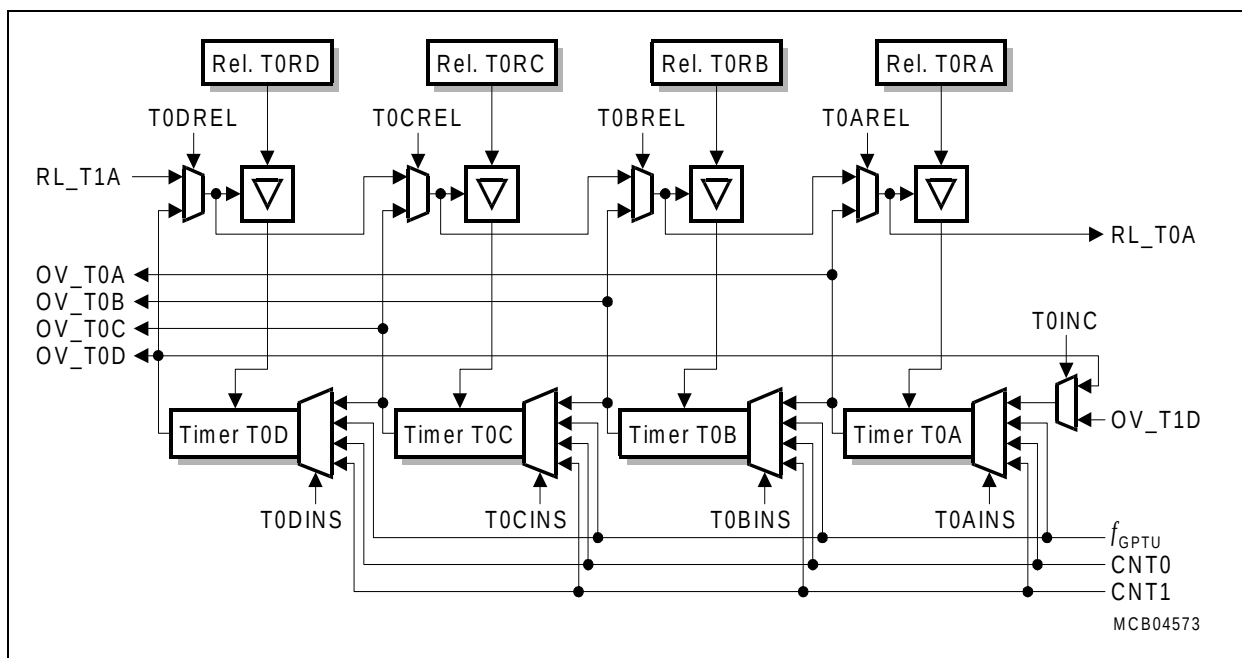


Figure 6-2 Detailed Block Diagram of T0

General Purpose Timer Unit (GPTU)

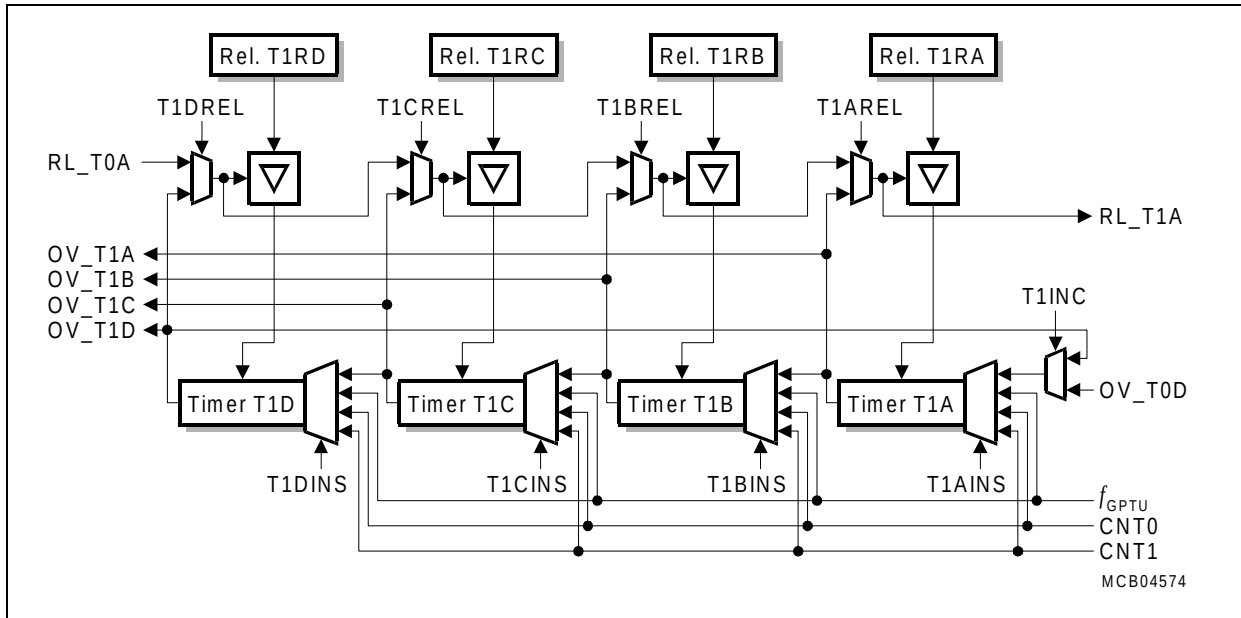


Figure 6-3 Detailed Block Diagram of T1

6.1.2.2 Input Selection

Each 8-bit timer block can select one of three possible inputs:

- The overflow of the previous timer (handled specially for T0A and T1A)
- An input frequency f_{GPTU} derived from the system clock
- One of two count inputs (CNT0, CNT1)

As shown in [Figure 6-2](#) and [Figure 6-3](#), each of the four 8-bit timer blocks within T0 and T1 receives an overflow from the previous 8-bit timer block. Additionally, the A blocks of both timers can be separately configured to receive overflow either from its own D block, or the other's D block (by way of T0INC and T1INC). The two selectable configurations are:

1. The A blocks receive the overflow of their own D-block timer (T0A input is T0D overflow, and T1A input is T1D overflow).
2. The A blocks receive the overflow of the other's D-block timer (T0A input is T1D overflow, and T1A input is T0D overflow).

When configuration 1 is selected, T0 and T1 operate independently. Both timers can be set up individually as 8-bit, 16-bit, 24-bit, or 32-bit timers.

When configuration 2 is selected T0 and T1 inter-operate, and can be concatenated to form wider timers. For 40-bit, 48-bit, 56-bit or 64-bit operation, the timer not receiving overflow from the other timer must be driven by the module clock, CNT0, or CNT1. Additionally, the overflow selection of the other 8-bit timers within T0 and T1 must all be configured appropriately to source overflow from its previous timer.

General Purpose Timer Unit (GPTU)

The source for the two count inputs (CNT0 or CNT1) can be either an external input or a trigger signal from T2 (by way of T2 overflow signals, OUV_T2A and OUV_T2B). **Figure 6-4** shows these options.

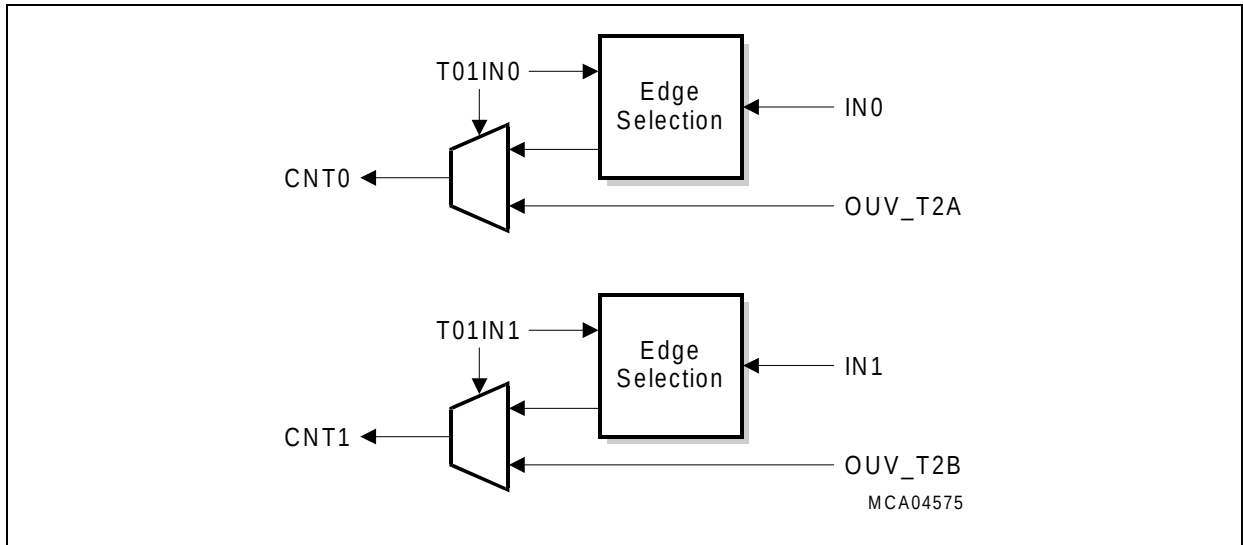


Figure 6-4 Timer T0 and T1 Global Input Control

Access to Timer T0 and T1 Count and Reload Registers

Two address locations are provided for each of the count and reload registers, which enable access to the appropriate registers even for a 24-bit timer configuration. The first address location provides all four bytes of a timer count/reload register. The symbolic name for this address indicates that all four parts, D..A, are accessible. Registers TxDCBA provide access to the count registers and registers TxRDCBA provide access to the reload registers. Individual access to the single bytes, combined 16-bit half-word aligned combination, or full 32-bit combination, is possible in this way. The second address location provides the lower three bytes of a timer count/reload register; the most-significant byte is not connected. The second address location enables access to a timer count/reload register in a 24-bit combination without corrupting the upper byte of the timer count/reload register. The symbolic name for this second address location is TxCBA (for the count registers) and TxRCBA (for the reload registers). These locations provide access only to the lower three parts, C..A, of the timer count and reload registers. **Table 6-1** gives an overview on the different access options to the individual combinations of T0 and T1.

General Purpose Timer Unit (GPTU)

Table 6-1 Access Options to T0/T1

Register	Access Width	Least-Significant Address Bits	TxD TxRD	TxC TxRC	TxB TxRB	TxA TxRA
TxDCBA TxRDCBA	Byte	000				x
	Byte	001			x	
	Byte	010		x		
	Byte	011	x			
	Half-word	000			x	
	Half-word	010	x			
	Word	000	x			
TxCBA TxRCBA	Byte	100				x
	Byte	101			x	
	Byte	110		x		
	Byte	111	0			
	Half-word	100			x	
	Half-word	110	0	x		
	Word	100	0	x		

Reading and writing to the individual byte or half-word parts of a timer is performed on the first address location using byte or half-word load/store operations. The entire 32-bit timer is accessed with word load/store operations. Reading from the second address location with a word load operation provides the contents of the lower three bytes of the timer count/reload register, with the most-significant byte returning 0. Writing to it with a word store operation affects only the lower three bytes. The value of the most-significant byte is not stored. It is recommended that software always writes 0 to the most-significant byte. The second address location can also be accessed with byte or half-word load/store operations.

Note: Access to a 16-bit half-word that crosses a half-word boundary (for example, the combination of T0C and T0B as one 16-bit timer) and access to a 24-bit combination using the upper three bytes (for example, T0D, T0C, and T0B) are not provided. Because it is always possible to align 16-bit timers on half-word boundaries, and right-align a 24-bit timer, these combinations are not required.

6.1.2.3 Reload Selection

As shown in [Figure 6-2](#) and [Figure 6-3](#), the reload trigger signals for the reload registers are controlled independently from timer concatenation. The independent control

General Purpose Timer Unit (GPTU)

provides the option of concatenating timers while giving each timer its own reload period. Reload selection is controlled by T0xREL and T1xREL so that each eight-bit timer can be triggered by either:

- The overflow of its own counter
- The reload event of one of the higher-order timer(s)

6.1.2.4 Service Requests, Output Signals, and Trigger Signals

Overflow signals from T0 and T1 can be used to generate service requests, output signals, or trigger signals for T2.

The four overflow signals from each 8-bit timer in T0 and T1 can trigger two service requests, two output signals, and two trigger signals. These options are shown in [Figure 6-5](#) for T0 and [Figure 6-6](#) for T1.

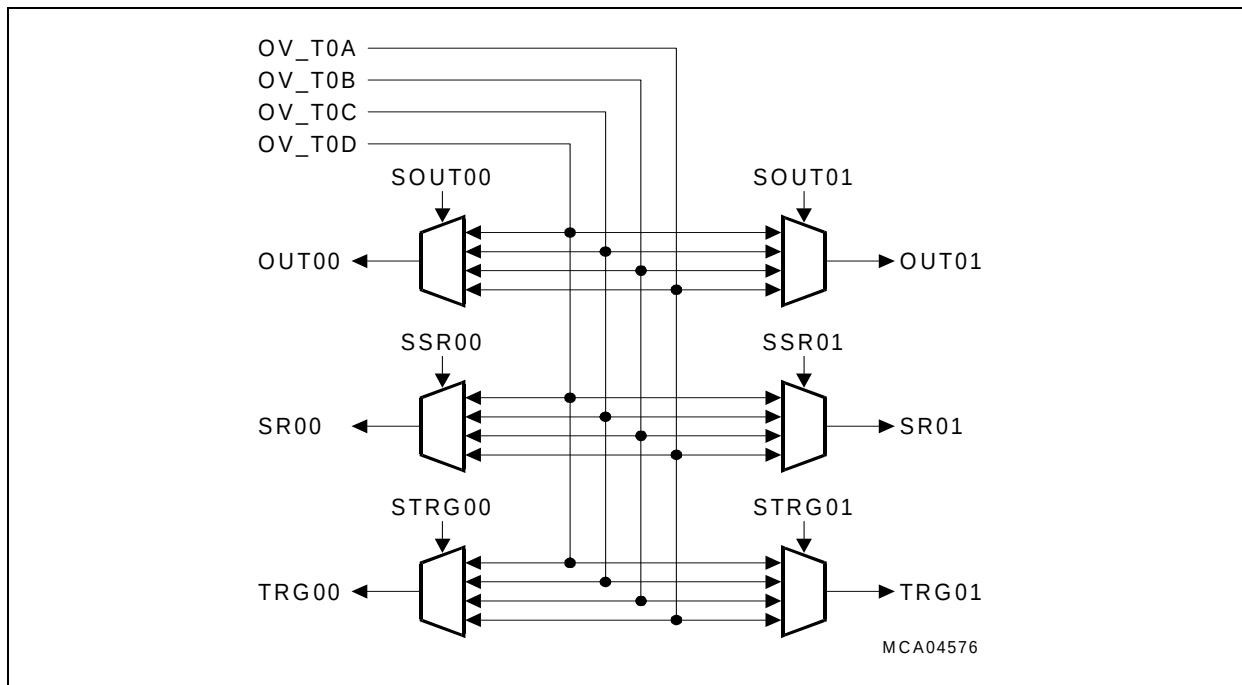


Figure 6-5 Timer T0 Output, Trigger, and Service Request Selection Control

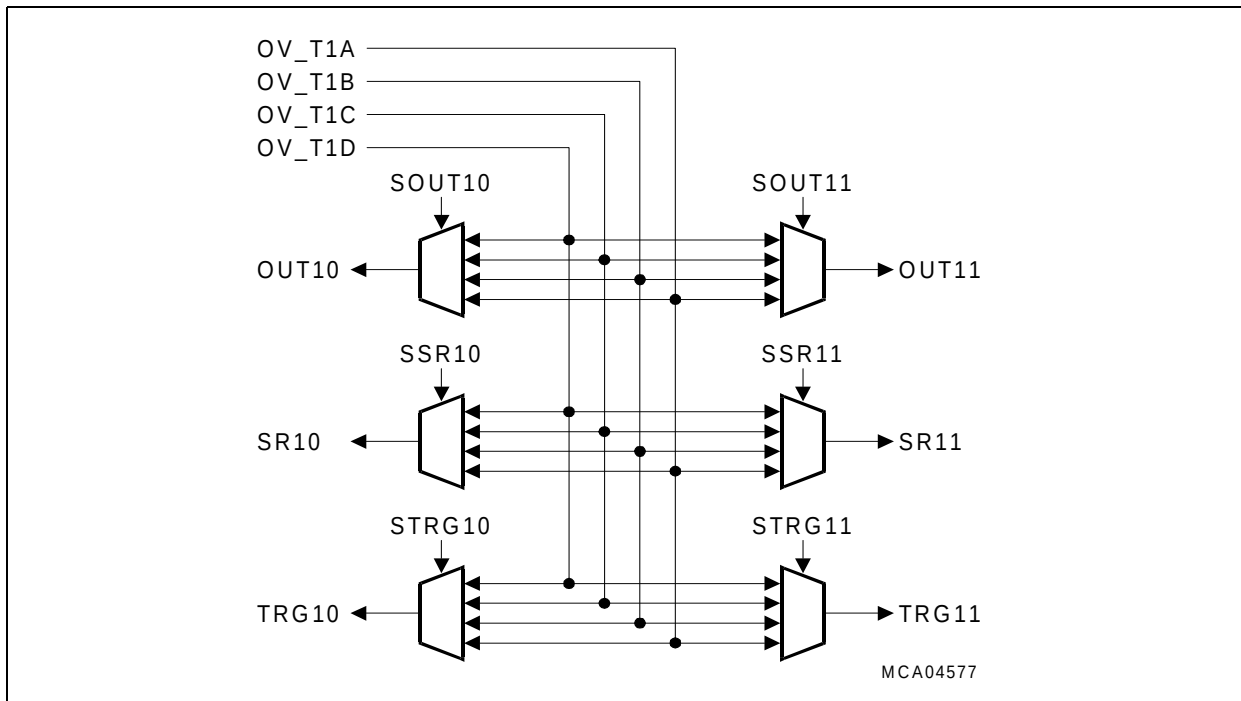


Figure 6-6 Timer T1 Output, Trigger, and Service Request Selection Control

6.1.2.5 Timers T0 and T1 Configuration Limitations

Due to timing delays of the internal circuitry, there are certain special cases and restrictions associated with the configuration possibilities of Timers T0 and T1.

In the following cases, one additional GPTU clock pulse is inserted into the count or reload signal:

- Overflow of T0D is used as count input to T0A
- Overflow of T1D is used as count input to T1A
- Overflow of T1D is used as count input to T0A
- Reload trigger of T0A/T0RA is used as reload trigger for T1D/T1RD

These combinations should either be avoided or the additional clock pulse needs to be taken into account. In the first three cases, if the timer producing the overflow is used as a prescaler for the following timer, the effect of the additional clock pulse is usually irrelevant. The prescaler just needs to be started such that the timer contents are one count higher than the reload value. This avoids a longer initial period due to the pulse delay.

It is recommended that the fourth case is always avoided. This case would occur if T0 and T1 (or parts of them) are concatenated such that T1D is the less significant and T0A is the more significant part of this timer combination. The overflow of T1D would be used as count input to T0A would experience a clock delay. The reload trigger line from T0A back to T1D would experience another clock delay, resulting in a total delay of two GPTU clocks from T1D overflow to its reload event. Because T1D continues counting after its

General Purpose Timer Unit (GPTU)

overflow, its contents will be overwritten by the reload two clock cycles later, resulting in the loss of two counts.

Concatenating T0 and T1 such that T0 contains the less significant part of the combined timer does not present a problem. The overflow of T0D to T1A and the reload trigger signal from T1A back to T0D do not have this extra delay.

Due to the high flexibility of the configuration options for Timers T0 and T1, it is almost never required to use one of the cases described above.

6.1.2.6 Timer T2

Timer T2 consists of two 16-bit timer blocks, T2A and T2B. Each 16-bit timer block contains a count register and two reload/capture registers. These blocks can be configured to form one 32-bit timer as shown in [Figure 6-7](#), or to run independently as two 16-bit timers as shown in [Figure 6-8](#). This basic configuration of Timer T2 is controlled by the T2CON.T2SPLIT control bit.

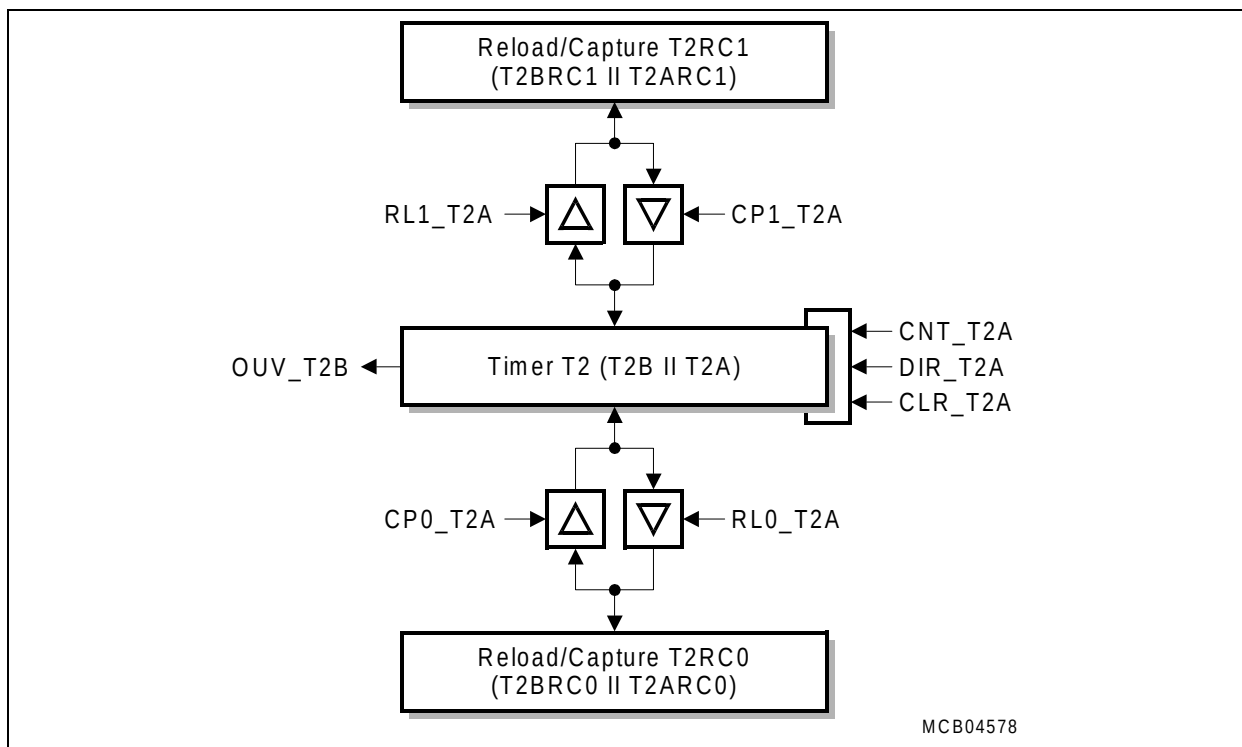


Figure 6-7 Block Diagram of Timer 2 in 32-Bit Mode

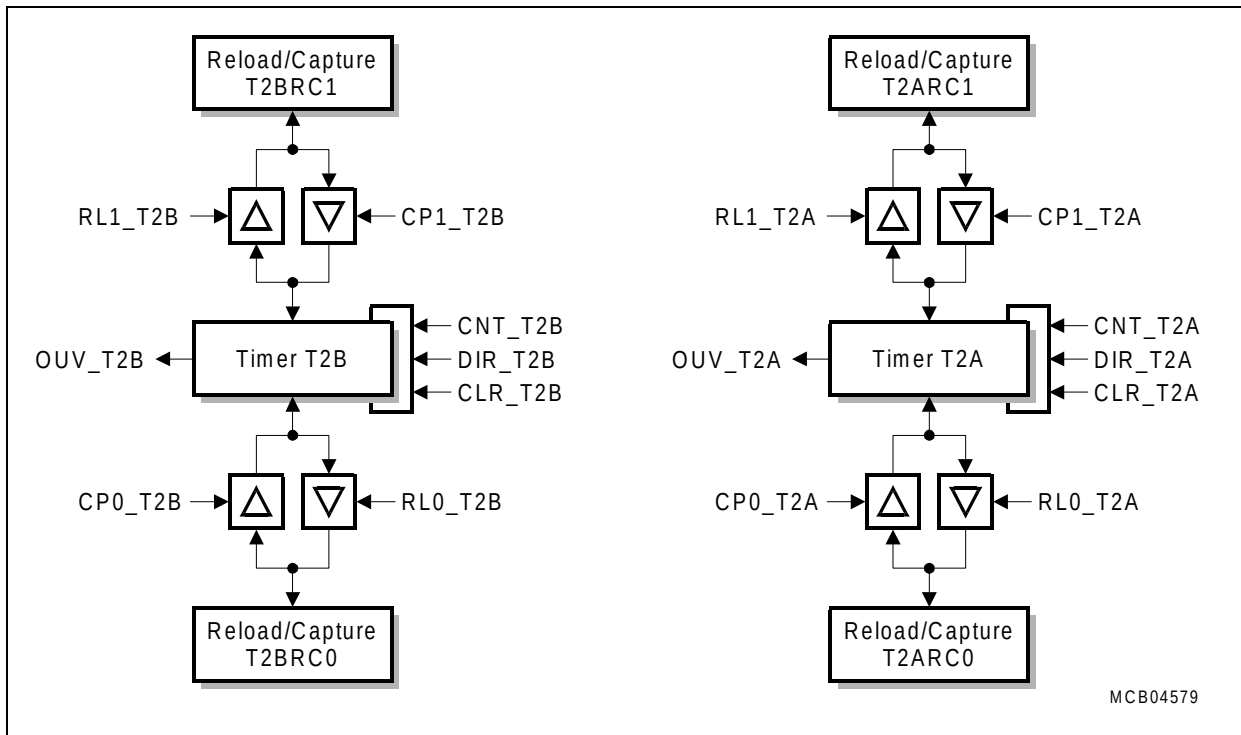


Figure 6-8 Block Diagram of Timer 2 in Split Mode

As shown in [Figure 6-9](#), any of the eight GPTU input lines can be assigned to trigger any of the functions performed by T2, including count, start, stop, change direction, clear, reload/capture, and service request. Each of these functions can be selectively triggered on a positive edge, a negative edge, or both edges of the input signal. In addition to these external inputs, signals from Timers T0 and T1 can be used to trigger functions in T2.

All external inputs can be assigned to any of the input functions of T2A and T2B, whether they are split or concatenated. When concatenated, all functions in T2A and T2B are controlled by the T2A mode control block. When split, T2A and T2B are controlled by their individual mode control blocks.

Three registers select the input line and the triggering edge for a specific function. The first register, T2AIS, selects the inputs for either T2 in 32-bit mode or T2A in Split Mode. Register T2BIS does the same for T2B in Split Mode. The third register, T2ES, provides the means to select which edge of the selected external signal causes a trigger of the associated function.

Most of these input signals can be used to generate a service request, independent of whether they are used to trigger Timer T2 functions or not.

Two registers control the mode of operation for the timer and the reload/capture registers. They also provide status information. Register T2CON controls the operation of the timer itself and holds the status information. Register T2RCCON controls the operation of the two reload/capture registers.

General Purpose Timer Unit (GPTU)

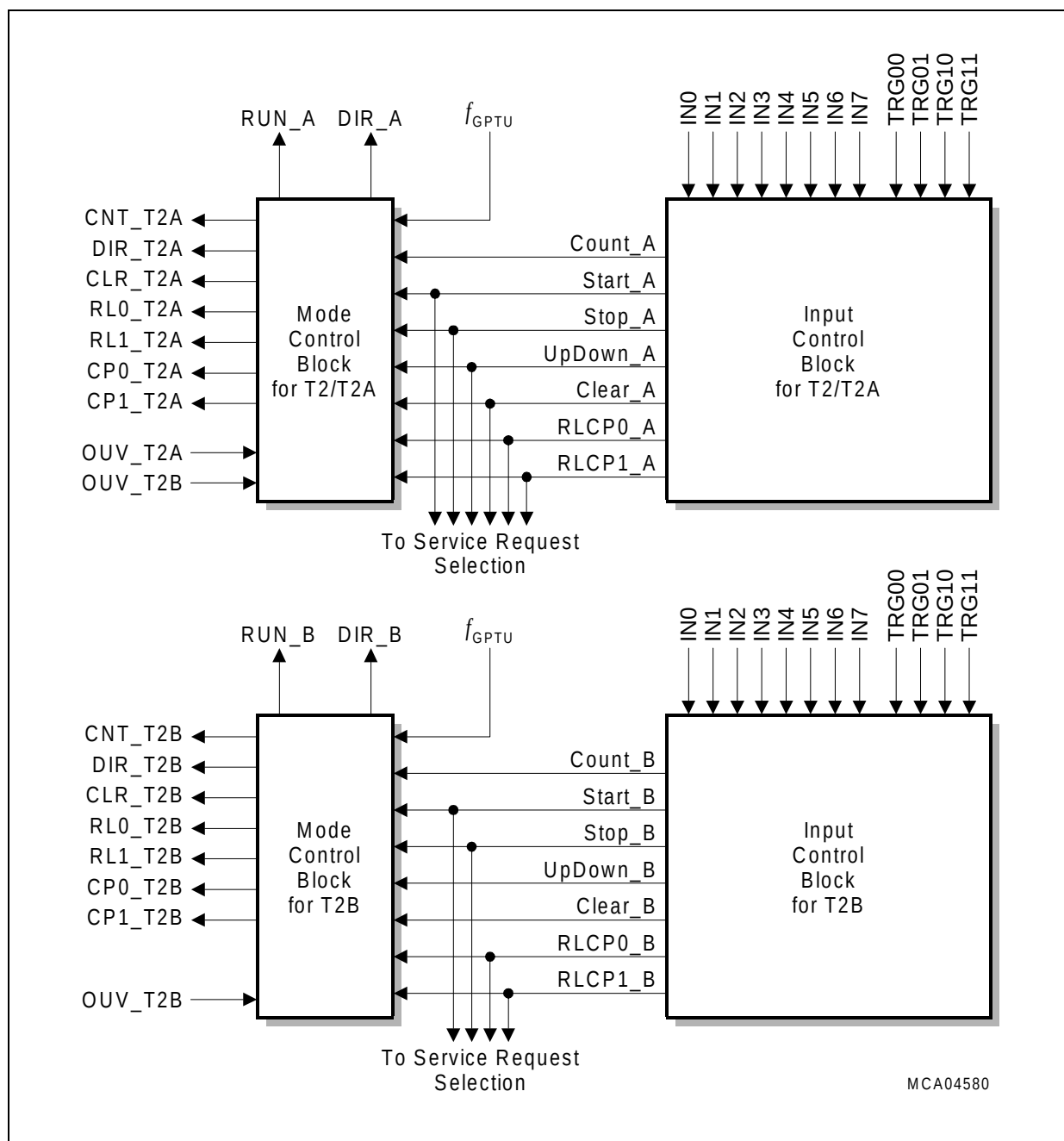


Figure 6-9 Timer 2 Input and Mode Control Blocks

General Purpose Timer Unit (GPTU)

Figure 6-10 and **Figure 6-11** show how T2 control signals are determined. This information is summarized as follows.

- Count control CNT_T2x
 - Clock Source Control, T2CON.T2xCSRC, determines the clocking trigger. Input can be the module clock (f_{GPTU}), or an external trigger source, Count_x. In Quadrature Counter Mode, count input sources are the two inputs, Count_x and UpDown_x.
 - External clocking trigger, Count_x, is determined by T2xIS.T2xICNT. Trigger source can be either an external input, INy, or a trigger signal, TRGxx, from Timer T0 or Timer T1. Bit T2ES.T2xECNT determines the active clock edge.
 - Starting and stopping of the timer can be controlled either by software via setting or clearing the run bit T012RUN.T2xRUN (software modifications of this bit are performed through the run bit set and clear bits, T012RUN.T2xSETR and T012RUN.T2xCLRR, respectively), or through the signals Start_x and Stop_x, selected by T2xIS.T2xISTR and T2xIS.T2xISTP, respectively. Any external input INy can be selected for this purpose. T2ES.T2xESTR and T2ES.T2xESTP determine the active clock edges for these sources, respectively. Additionally, in one-shot mode, the timer is stopped in response to its own overflow, OUV_T2x.
 - The running/stopped status of T2A and T2B can be examined via the T012RUN.T2xRUN status bits.
- Count direction control DIR_T2x
 - Input source control, T2CON.T2xCDIR, selects whether the count direction is up or down, or whether it is determined from an external input.
 - External input selection is controlled by T2xIS.T2xIUD, which selects any of the INy input signals. T2ES.T2xEUD determines the active clock edge. In Quadrature Counter Mode, up/down count information is derived from the two input sources, Count_x and UpDown_x.
- Clear control CLR_T2x
 - T2CON.T2xCCLR selects whether to clear the timer to 0 on an external event (Clear_x), or to clear the timer on capture 0 event (CP0_T2x), or to clear timer on capture 1 event (CP1_T2x).
 - Selection of the external trigger is determined by T2xIS.T2xICLR, which selects any of the INy input signals. T2ES.T2xECLR determines the active clock edge.
- Reload/capture RL0_T2x, RL1_T2x, and CP0_T2x, CP1_T2x
 - There are two reload/capture registers each in T2A and T2B which can be programmed independently.
 - Controls T2RCCON.T2xMRC0 and T2RCCON.T2xMRC1 determine reload/capture modes. Modes include disabled, capture on external event, reload on overflow or underflow, reload on external event, reload on overflow only, reload on underflow only, reload on external event if count direction is up (if T2CON.T2xDIR = 0), reload on external event if count direction is down (T2CON.T2xDIR = 1).

General Purpose Timer Unit (GPTU)

- Selection of external trigger source for RLCP0_x and RLCP1_x is determined by T2xIS.T2xIRC0 and T2xIS.T2xIRC1. Trigger source can be either an external input, GPTUx_INy, or a trigger signal, TRGxx, from T0 or T1. T2ES.T2xERC0 and T2ES.T2xERC1 determine the active edge of the trigger signal.

General Purpose Timer Unit (GPTU)

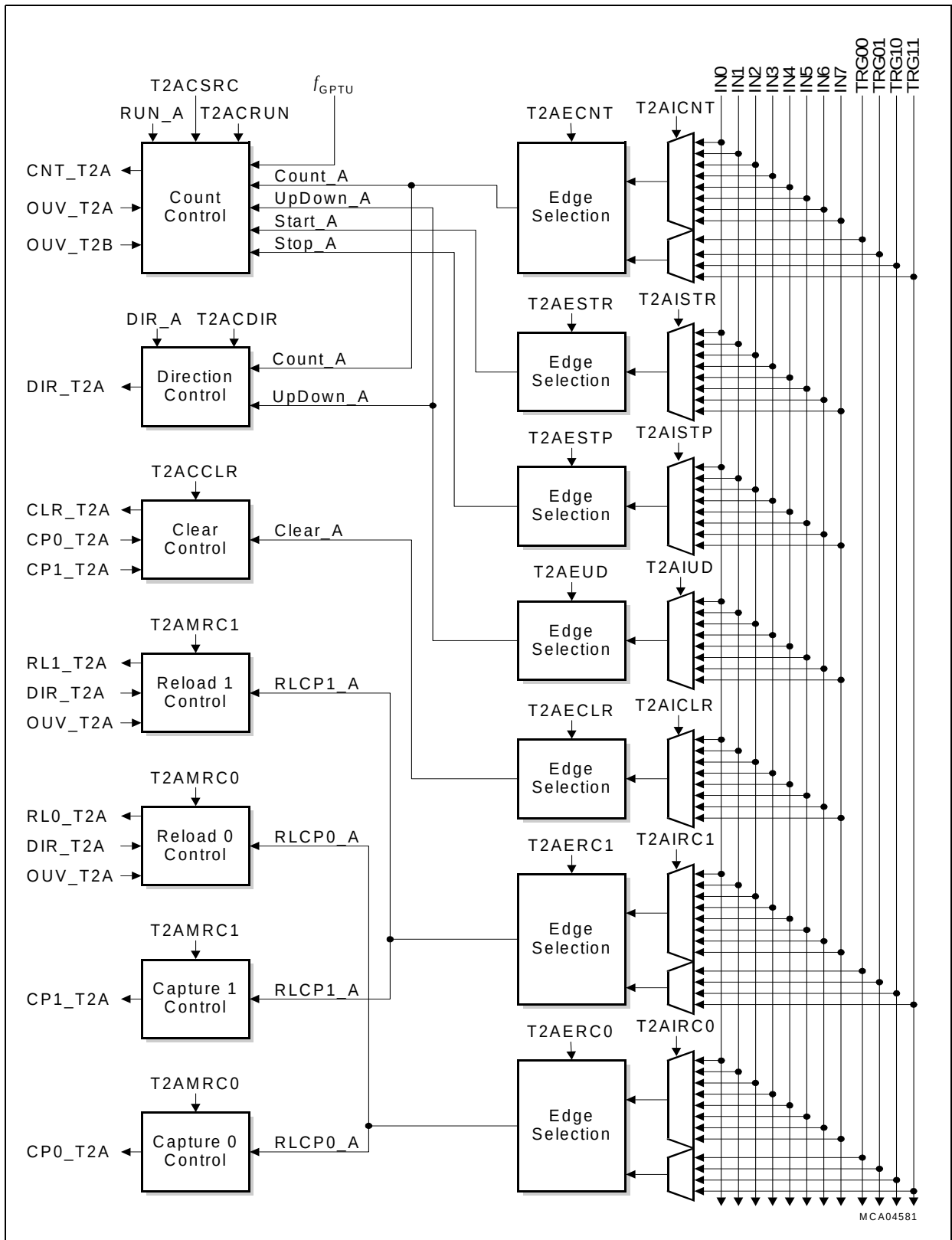


Figure 6-10 Timer T2/T2A Input and Mode Control Details

General Purpose Timer Unit (GPTU)

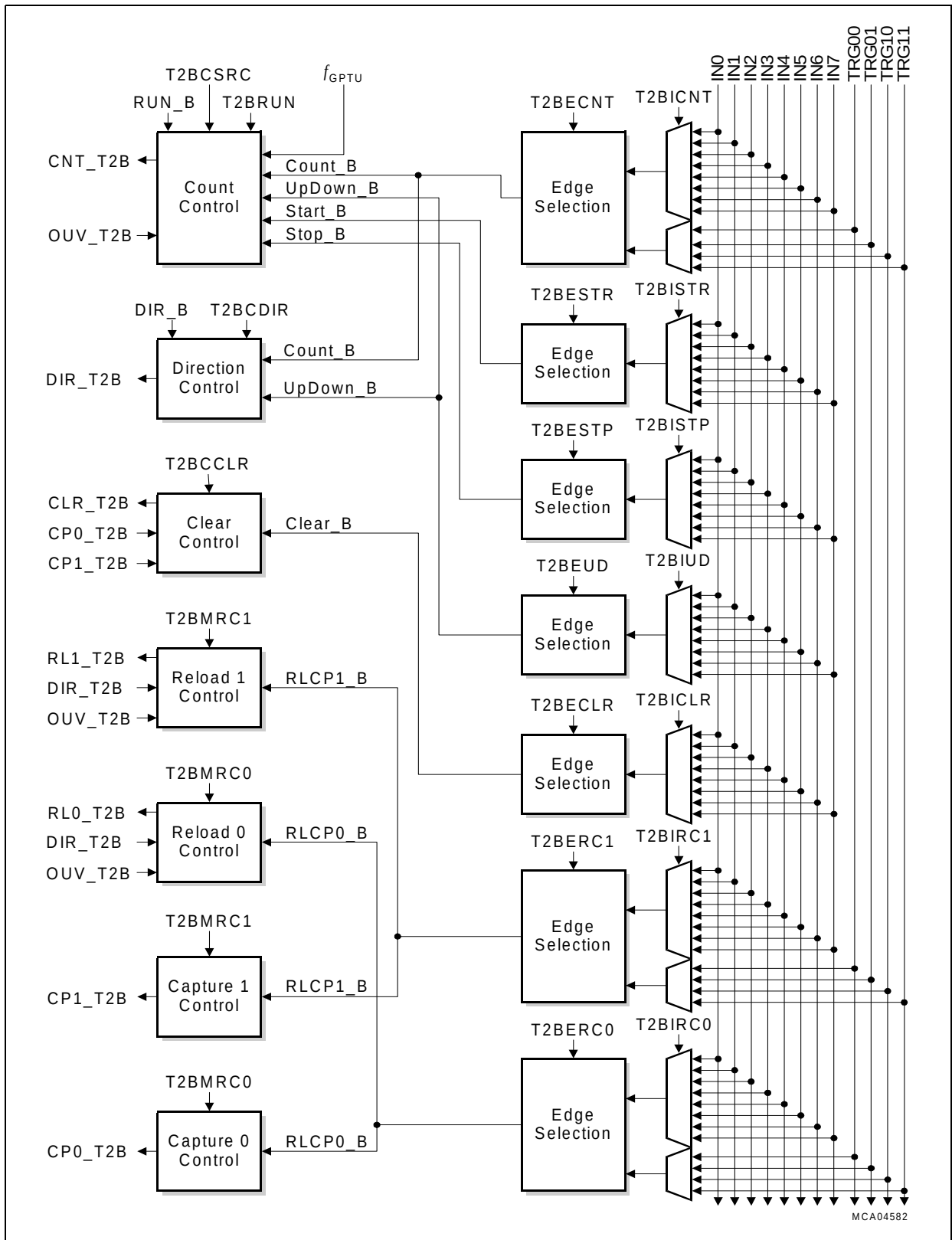


Figure 6-11 Timer T2B Input and Mode Control Details

6.1.2.7 Quadrature Counting Mode

Position tracking can be performed with Timer T2 in Quadrature Counting Mode, sometimes referred to as incremental or phase encoded interface. The standard way of tracking positions is to use two phase-shifted input signals. These provide the counting and direction information necessary for this task. As shown in [Figure 6-12](#), the edges of the signals provide the count signal, while the phase relation between the two signals provides the direction information.

To operate Timer T2 in this mode, the two signals are connected such that they trigger the Count_A/Count_B and the UpDown_A/UpDown_B inputs of the timer block.

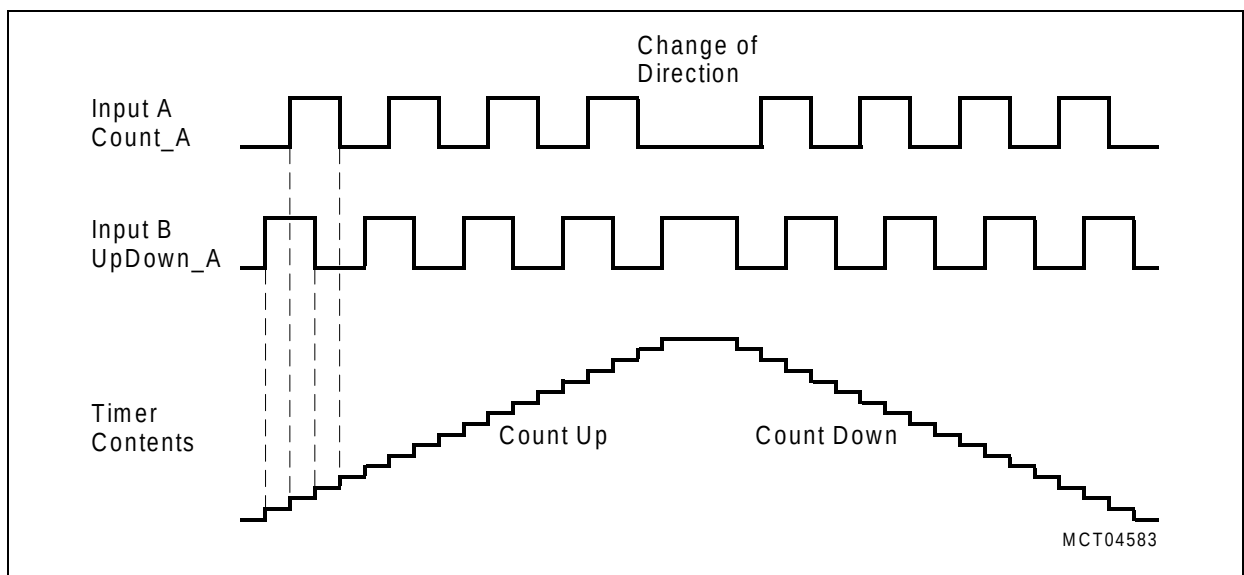


Figure 6-12 Quadrature Counting Operation

6.1.3 Global GPTU Controls

This section describes global control of the GPTU. Global controls are provided for the outputs and interrupt service requests.

6.1.3.1 Output Control

The register OUT has eight bits OUT_x (x = 0-7) which store the output signals from the GPTU. The bits in register OUT can also be set or cleared via software. The connection of timer signals to these output bits is determined by eight bit fields in register OSEL, named SO_x (x = 0-7). Each output bit in register OUT is connected to a GPTU output line, which connects to the Parallel Ports.

Six signals from Timers T0, T1, and T2 can be selected to generate outputs from the GPTU timers to the Parallel Ports. For each of the eight GPTU output signals, OUT[7:0], the user can select which of the timer signals, OUT00, OUT01, OUT10, OUT11, OUV_T2A, or OUV_T2B, activates the selected output line.

OUT00 and OUT01 can be any T0 timer overflow, OUT10, OUT11 can be any T1 timer overflow. OUV_T2A and OUV_T2B are the timer overflows of T2A and T2B.

Figure 6-13 provides an overview of the output options.

General Purpose Timer Unit (GPTU)

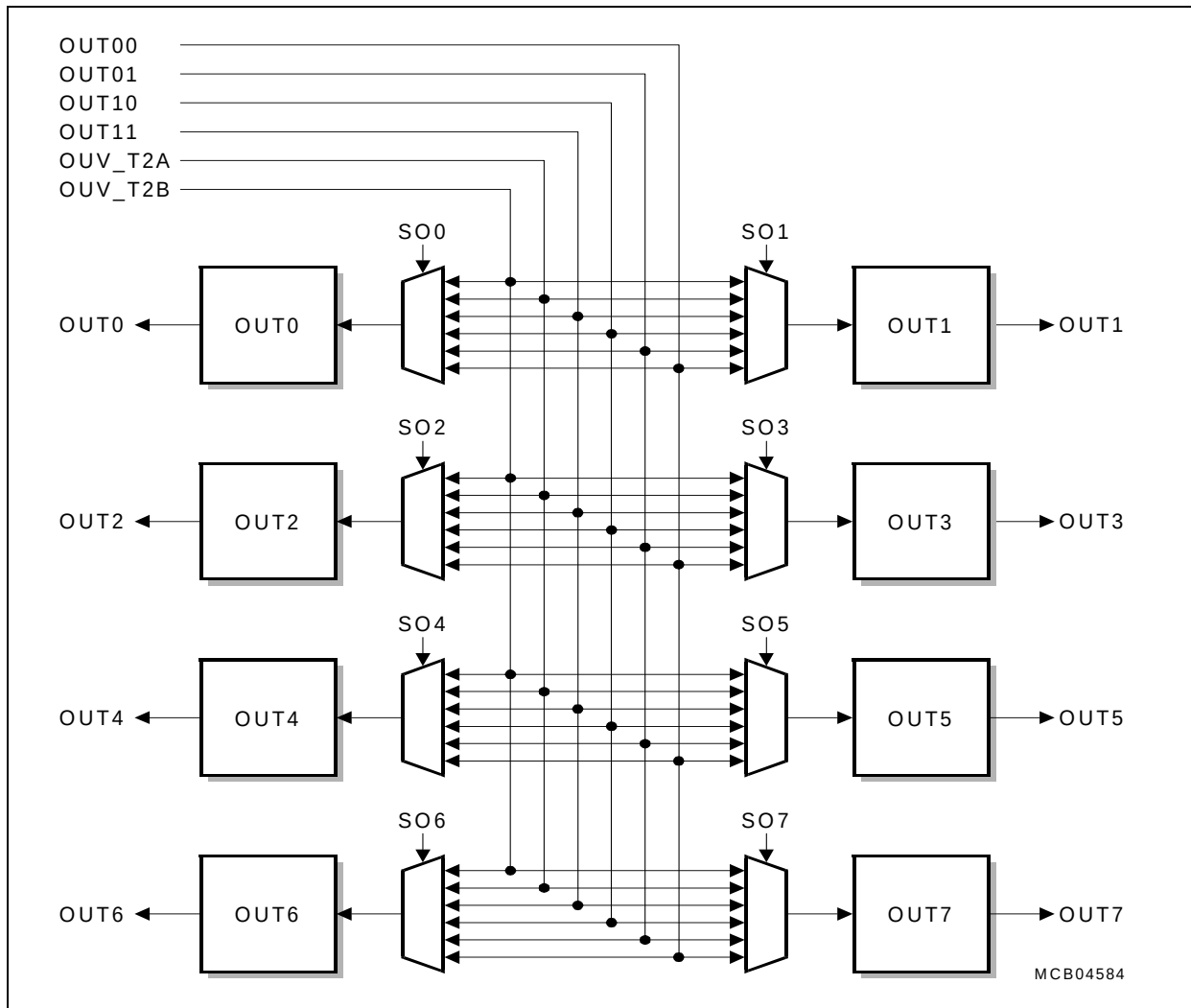


Figure 6-13 Output Control Block Diagram

6.1.3.2 Service Request Control

Sixteen events in T0, T1, and T2 can be selected to generate a service request to the CPU. Eight service request outputs (nodes), SR[7:0], are provided for the GPTU; they can be freely assigned to any of the GPTU events.

Timer T2 events which can be selected include Start_x, Stop_x, UpDown_A, Clear_A (signals UpDown_B and Clear_B are not available for service request generation), RLCP0_x, RLCP1_x, OUV_T2x. Timer T0 overflow events (SR00, SR01) and Timer T1 overflow events (SR10, and SR11) can also be selected. **Figure 6-14** shows these options. Please note that the signals Start_x, Stop_x, UpDown_A, Clear_A, RLCP0_x, and RLCP1_x are the signals coming out of the input selection block, before these lines go into the Timer T2 control logic (see **Figure 6-9**). This has the advantage, that an input line can be used to generate a service request only; it may or may not be used to also trigger a T2 function. In this way, all of the GPTU input lines connected to parallel port pins can be configured as external interrupt inputs.

Because Timers T0 and T1 can generate triggers for Timer T2 signals (such as Count_x, RLCP0_x, and RLCP1_x), it is possible to use these signals for service request generation (whether or not they are also used to trigger functions of T2). This gives additional service requests to Timers T0 and T1.

Because of the flexibility in selecting service requests, more than one service request can be generated by the same event.

General Purpose Timer Unit (GPTU)

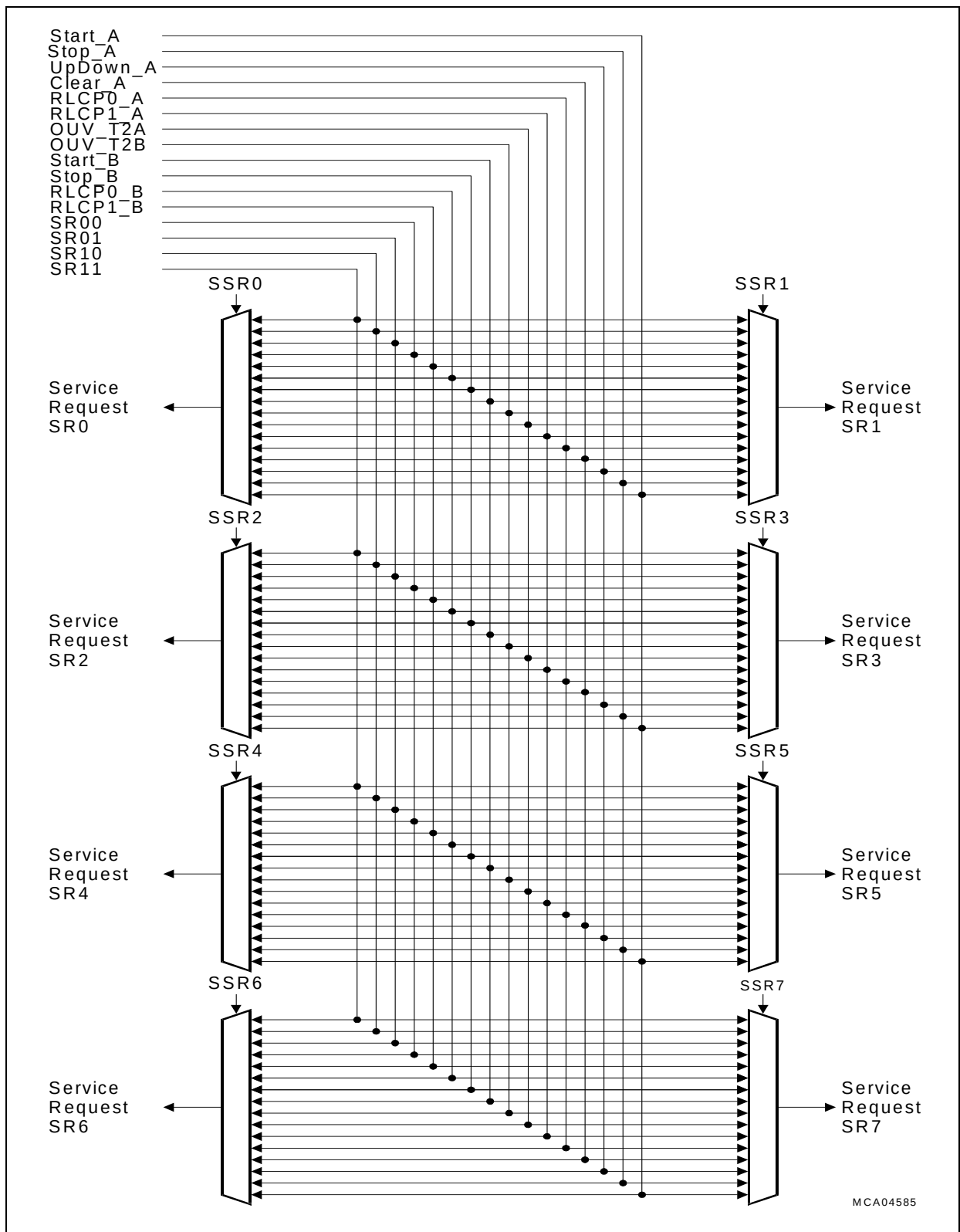


Figure 6-14 Service Request Selection

6.2 GPTU Kernel Registers

Figure 6-15 and Table 6-2 show all registers associated with the GPTU Kernel.

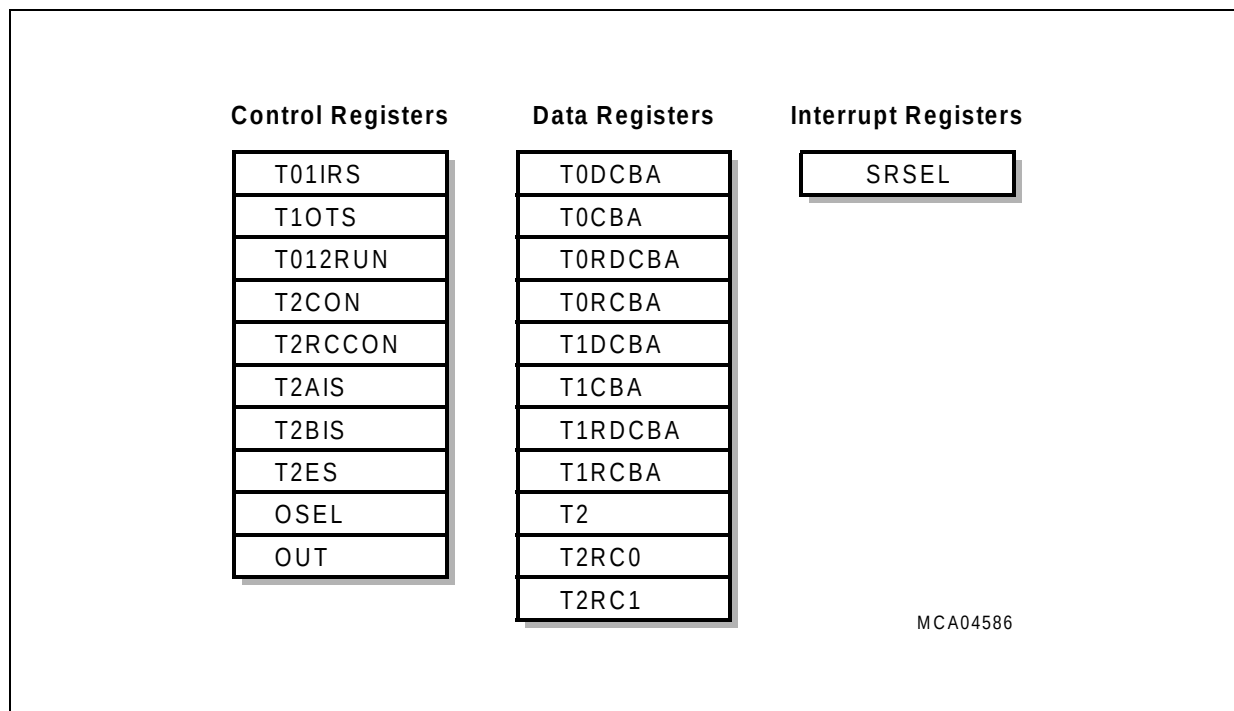


Figure 6-15 GPTU Kernel Registers

Table 6-2 GPTU Kernel Registers

Register Short Name	Register Long Name	Offset Address	Description see
T01IRS	Timer T0 and T1 Input and Reload Source Selection Register	0010 _H	Page 6-24
T01OTS	Timer T0 and T1 Output, Trigger and Service Request Register	0014 _H	Page 6-27
T2CON	Timer T2 Control Register	0018 _H	Page 6-38
T2RCCON	Timer T2 Reload/Capture Control Register	001C _H	Page 6-43
T2AIS	Timer T2/T2A Ext. Input Selection Register	0020 _H	Page 6-33
T2BIS	Timer T2B External Input Selection Register	0024 _H	Page 6-35
T2ES	Timer T2 External Input Edge Selection Reg.	0028 _H	Page 6-36
OSEL	Output Source Selection Register	002C _H	Page 6-47
OUT	Output Register	0030 _H	Page 6-48

General Purpose Timer Unit (GPTU)

Table 6-2 GPTU Kernel Registers (cont'd)

Register Short Name	Register Long Name	Offset Address	Description see
T0DCBA	Timer T0 Count Register (T0D, T0C, T0B, T0A)	0034 _H	Page 6-29
T0CBA	Timer T0 Count Register (T0C, T0B, T0A)	0038 _H	Page 6-29
T0RDCBA	Timer T0 Reload Register (T0RD, T0RC, T0RB, T0RA)	003C _H	Page 6-30
T0RCBA	Timer T0 Reload Register (T0RC, T0RB, T0RA)	0040 _H	Page 6-30
T1DCBA	Timer T1 Count Register (T1D, T1C, T1B, T1A)	0044 _H	Page 6-31
T1CBA	Timer T1 Count Register (T1C, T1B, T1A)	0048 _H	Page 6-31
T1RDCBA	Timer T1 Reload Register (T1RD, T1RC, T1RB, T1RA)	004C _H	Page 6-31
T1RCBA	Timer T1 Reload Register (T1RC, T1RB, T1RA)	0050 _H	Page 6-32
T2	Timer T2 Count Register	0054 _H	Page 6-45
T2RC0	Timer T2 Reload/Capture Register 0	0058 _H	Page 6-46
T2RC1	Timer T2 Reload/Capture Register 1	005C _H	Page 6-46
T012RUN	Timers T0, T1, T2 Run Control Register	0060 _H	Page 6-41
SRSEL	Service Request Source Select Reg.	00DC _H	Page 6-49

6.2.1 Timer T0/T1 Registers

This section describes the registers related to Timers T0 and T1. Note that register T012RUN is shared between all three timers and is described in [Section 6.2.2.3](#).

6.2.1.1 Timer T0/T1 Input & Reload Source Selection Register

The T01IRS register contains the individual controls for the count input and the reload trigger selections for the individual parts of T0 and T1. This register also contains the control for the global input signals CNT0 and CNT1.

T01IRS

Timer T0 and T1 Input and Reload Source Selection Register

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
T01 IN1	T01 IN0	0		T1 INC	T0 INC	T1D REL	T1C REL	T1B REL	T1A REL	T0D REL	T0C REL	T0B REL	T0A REL		
rw	rw	r		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
T1D INS	T1C INS	T1B INS		T1A INS	T0D INS	T0C INS		T0B INS	T0A INS						
rw	rw	rw		rw	rw	rw		rw	rw						

Field	Bits	Type	Description
T0AINS	[1:0]	rw	T0A Input Selection 00 Clock input f_{GPTU} 01 Global input CNT0 10 Global input CNT1 11 Carry input (concatenation)
T0BINS	[3:2]	rw	T0B Input Selection ; coding as T0AINS
T0CINS	[5:4]	rw	T0C Input Selection ; coding as T0AINS
T0DINS	[7:6]	rw	T0D Input Selection ; coding as T0AINS
T1AINS	[9:8]	rw	T1A Input Selection ; coding as T0AINS
T1BINS	[11:10]	rw	T1B Input Selection ; coding as T0AINS
T1CINS	[13:12]	rw	T1C Input Selection ; coding as T0AINS
T1DINS	[15:14]	rw	T1D Input Selection ; coding as T0AINS

General Purpose Timer Unit (GPTU)

Field	Bits	Type	Description
T0AREL	16	rw	T0A Reload Source Selection 0 Reload on overflow of timer T0A 1 Concatenation with T0RB
T0BREL	17	rw	T0B Reload Source Selection 0 Reload on overflow of timer T0B 1 Concatenation with T0RC
T0CREL	18	rw	T0C Reload Source Selection 0 Reload on overflow of timer T0C 1 Concatenation with T0RD
T0DREL	19	rw	T0D Reload Source Selection 0 Reload on overflow of timer T0D 1 Reload on signal T1RA
T1AREL	20	rw	T1A Reload Source Selection 0 Reload on overflow of timer T1A 1 Concatenation with T1RB
T1BREL	21	rw	T1B Reload Source Selection 0 Reload on overflow of timer T1B 1 Concatenation with T1RC
T1CREL	22	rw	T1C Reload Source Selection 0 Reload on overflow of timer T1C 1 Concatenation with T1RD
T1DREL	23	rw	T1D Reload Source Selection 0 Reload on overflow of timer T1D 1 Concatenation with T0RA
T0INC	24	rw	T0 Carry Input Selection 0 T0A carry in is T0D carry out 1 T0A carry in is T1D carry out
T1INC	25	rw	T1 Carry Input Selection 0 T1A carry in is T1D carry out 1 T1A carry in is T0D carry out
0	[27:26]	r	Reserved ; read as 0; writing to these bit positions has no effect.

General Purpose Timer Unit (GPTU)

Field	Bits	Type	Description
T01IN0	[29:28]	rw	T0 and T1 Global Input CNT0 Selection 00 Timer T2A overflow/underflow OUV_T2A 01 Positive edge of IN0 10 Negative edge of IN0 11 Both edges of IN0
T01IN1	[31:30]	rw	T0 and T1 Global Input CNT1 Selection 00 Timer T2A overflow/underflow OUV_T2B 01 Positive edge of IN1 10 Negative edge of IN1 11 Both edges of IN1

General Purpose Timer Unit (GPTU)

6.2.1.2 Timer T0/T1 Output, Trigger, and Service Request Selection Register

The T01OTS register performs the selections for the output, service request, and trigger signals of the individual parts of both Timers T0 and T1.

T01OTS

Timer T0 and T1 Output, Trigger and Service Request Selection Register

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0				SSR11		SSR10		STRG11		STRG10		SOUT11		SOUT10	
r				rw		rw		rw		rw		rw		rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0				SSR01		SSR00		STRG01		STRG00		SOUT01		SOUT00	
r				rw		rw		rw		rw		rw		rw	

Field	Bits	Type	Description
SOUT00	[1:0]	rw	T0 Output 0 Source Selection encoding see Table 6-3
SOUT01	[3:2]	rw	T0 Output 1 Source Selection encoding see Table 6-3
STRG00	[5:4]	rw	T0 Trigger Output 0 Source Selection encoding see Table 6-3
STRG01	[7:6]	rw	T0 Trigger Output 1 Source Selection encoding see Table 6-3
SSR00	[9:8]	rw	T0 Service Request 0 Source Selection encoding see Table 6-3
SSR01	[11:10]	rw	T0 Service Request 1 Source Selection encoding see Table 6-3
SOUT10	[17:16]	rw	T1 Output 0 Source Selection encoding see Table 6-3
SOUT11	[19:18]	rw	T1 Output 1 Source Selection encoding see Table 6-3
STRG10	[21:20]	rw	T1 Trigger Output 0 Source Selection encoding see Table 6-3

General Purpose Timer Unit (GPTU)

Field	Bits	Type	Description
STRG11	[23:22]	rw	T1 Trigger Output 1 Source Selection encoding see Table 6-3
SSR10	[25:24]	rw	T1 Service Request 0 Source Selection encoding see Table 6-3
SSR11	[27:26]	rw	T1 Service Request 1 Source Selection encoding see Table 6-3
0	[15:12] [31:28]	r	Reserved ; read as 0; writing to these bit positions has no effect.

Table 6-3 T0/T1 Overflow Source Selection (x, y = 0, 1)

Service Request Selection SSRxy	Trigger Output Selection STRGxy	Output Source Selection SOUTxy	Selected Overflow Signal
00	00	00	TxA overflow
01	01	01	TxB overflow
10	10	10	TxC overflow
11	11	11	TxD overflow

6.2.1.3 Timer T0 and T1 Count and Reload Registers

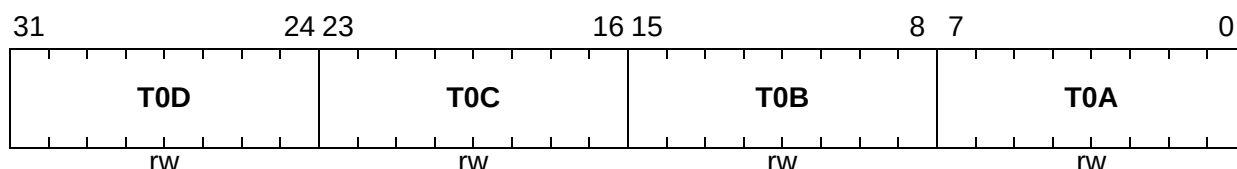
Timer T0 Count Register T0DCBA (T0D, T0C, T0B, T0A)

This register provides read/write access to all four parts of Timer T0.

T0DCBA

Timer T0 Count Register (T0D, T0C, T0B, T0A)

Reset Value: 0000 0000_H



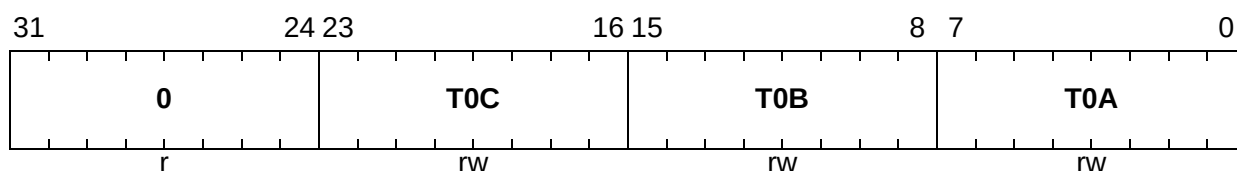
Timer T0 Count Register T0CBA (T0C, T0B, T0A)

This register provides read/write access to the lower three parts of Timer T0. The upper byte is always read as 0; writes to it have no effect and are not stored. This register needs to be used if parts A, B, and C of Timer T0 are configured as a 24-bit timer. Part D of Timer T0 will not be affected when writing to this register.

T0CBA

Timer T0 Count Register (T0C, T0B, T0A)

Reset Value: 0000 0000_H



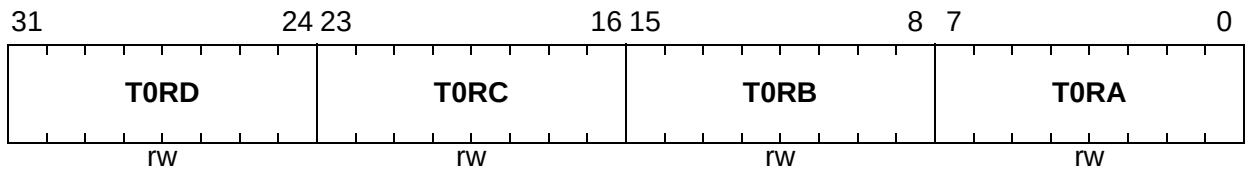
General Purpose Timer Unit (GPTU)

Timer T0 Reload Register T0RDCBA (T0RD, T0RC, T0RB, T0RA)

This register provides read/write access to all four parts of the reload register of Timer T0.

T0RDCBA

Timer T0 Reload Register (T0RD, T0RC, T0RB, T0RA) Reset Value: 0000 0000_H

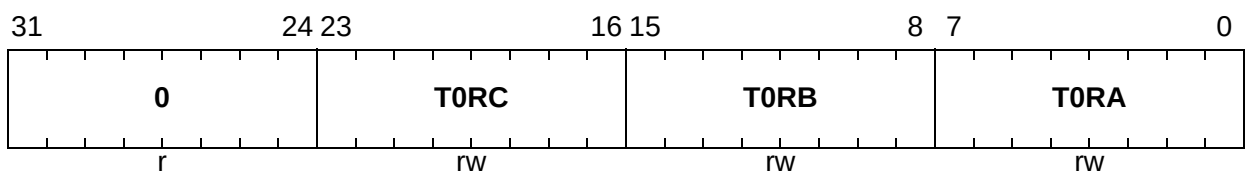


T0RCBA, Timer T0 Reload Register (T0RC, T0RB, T0RA)

This register provides read/write access to the lower three parts of the reload register of Timer T0. The upper byte is always read as 0; writes to it have no effect and are not stored. This reload register needs to be used if parts A, B, and C of Timer T0 are configured as a 24-bit timer. Part D of the reload register will not be affected when writing to this register.

T0RCBA

Timer T0 Reload Register (T0RC, T0RB, T0RA) Reset Value: 0000 0000_H



General Purpose Timer Unit (GPTU)

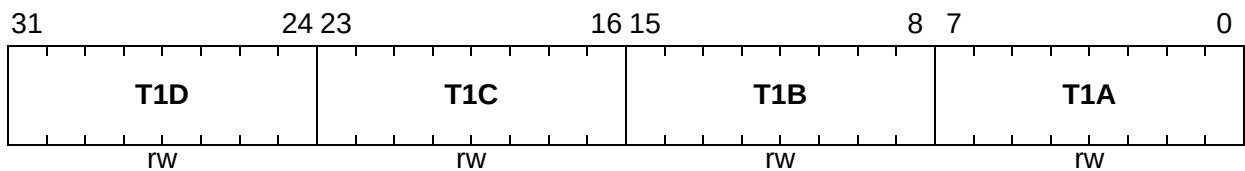
Timer T1 Count Register T1DCBA (T1D, T1C, T1B, T1A)

This register provides read/write access to all four parts of Timer T1.

T1DCBA

Timer T1 Count Register (T1D, T1C, T1B, T1A)

Reset Value: 0000 0000_H



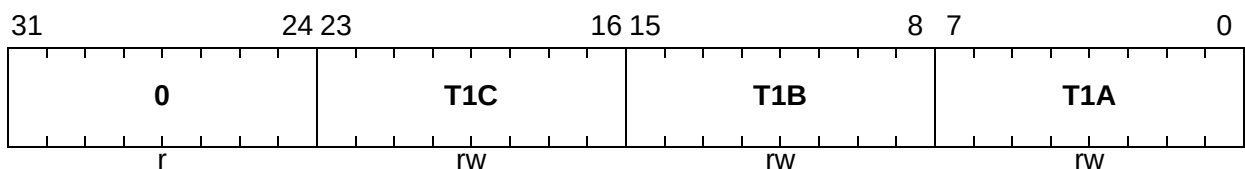
Timer T1 Count Register T1CBA (T1C, T1B, T1A)

This register provides read/write access to the lower three parts of Timer T1. The upper byte is always read as 0; writes to it have no effect and are not stored. This register needs to be used if parts A, B, and C of Timer T1 are configured as a 24-bit timer. Part D of Timer T1 will not be affected when writing to this register.

T1CBA

Timer T1 Count Register (T1C, T1B, T1A)

Reset Value: 0000 0000_H



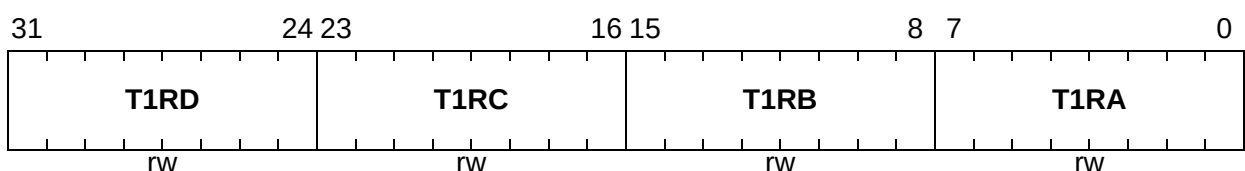
Timer T1 Reload Register T1RDCBA (T1RD, T1RC, T1RB, T1RA)

This register provides read/write access to all four parts of the reload register of Timer T1.

T1RDCBA

Timer T1 Reload Register (T1RD, T1RC, T1RB, T1RA)

Reset Value: 0000 0000_H



General Purpose Timer Unit (GPTU)

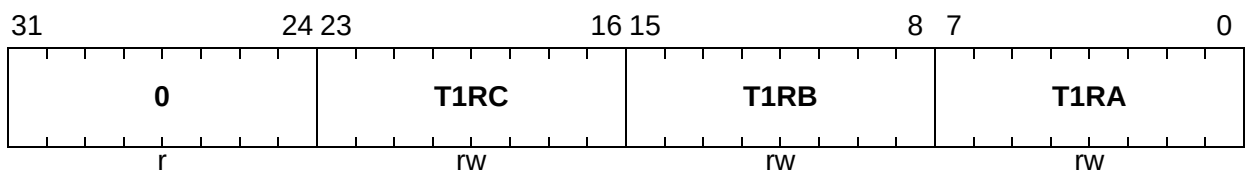
Timer T1 Reload Register T1RCBA (T1RC, T1RB, T1RA)

This register provides read/write access to the lower three parts of the reload register of Timer T1. The upper byte is always read as 0; writes to it have no effect and are not stored. This reload register needs to be used if parts A, B, and C of Timer T1 are configured as a 24-bit timer. Part D of the reload register will not be affected when writing to this register.

T1RCBA

Timer T1 Reload Register (T1RC, T1RB, T1RA)

Reset Value: 0000 0000_H



6.2.2 Timer T2 Registers

This section describes the Timer T2 registers.

6.2.2.1 Input Control Registers

Three registers select the input line and the triggering edge for a specific function. The first register, T2AIS, selects the inputs for either Timer T2 in 32-bit mode or Timer T2A in Split Mode. Register T2BIS does the same for Timer T2B in Split Mode. The third register, T2ES, provides the means to select which edge of the selected external signal causes a trigger of the associated function.

Most of these input signals can be used to generate a service request, independent of whether they are used to trigger Timer T2 functions or not.

Timer T2/T2A External Input Selection Register T2AIS

The T2AIS register selects which of the eight external inputs or trigger events from Timer T0/T1 is to be used for the various input functions for Timer T2A. It controls the input selection for Timer T2A in Split Mode and for the entire Timer T2 in 32-bit mode.

T2AIS

Timer T2/T2A External Input Selection Register

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0					T2AIRC1				0	T2AIRC0			0	T2AICLR	
r					rw				r	rw			r	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	T2AIUD			0	T2AISTP			0	T2AISTR			0	T2AICNT		
r	rw			r	rw			r	rw			r	rw		

Field	Bits	Type	Description
T2AICNT	[2:0]	rw	Timer T2A External Count Input Selection encoding see Table 6-4
T2AISTR	[6:4]	rw	Timer T2A External Start Input Selection encoding see Table 6-4
T2AISTP	[10:8]	rw	Timer T2A External Stop Input Selection encoding see Table 6-4
T2AIUD	[14:12]	rw	Timer T2A External Up/Down Input Selection encoding see Table 6-4

General Purpose Timer Unit (GPTU)

Field	Bits	Type	Description
T2AICLR	[18:16]	rw	Timer T2A External Clear Input Selection encoding see Table 6-4
T2AIRC0	[22:20]	rw	Timer T2A External Reload/Capture 0 Input encoding see Table 6-4
T2AIRC1	[26:24]	rw	Timer T2A External Reload/Capture 1 Input encoding see Table 6-4
0	3, 7, 11,15, 19, 23, [31:27]	r	Reserved ; read as 0; writing to these bit positions has no effect.

Table 6-4 T2 Input Source Selection (x, y = 0, 1)

Value	Selected External Input	In Parallel Selected Input for T2AICNT, T2AIRC1, and T2AIRC0; and T2BICNT, T2BIRC1, and T2BIRC0
000	Input IN0	T0/T1 Trigger Input Signal TRG00
001	Input IN1	T0/T1 Trigger Input Signal TRG01
010	Input IN2	T0/T1 Trigger Input Signal TRG10
011	Input IN3	T0/T1 Trigger Input Signal TRG11
100	Input IN4	T0/T1 Trigger Input Signal TRG00
101	Input IN5	T0/T1 Trigger Input Signal TRG01
110	Input IN6	T0/T1 Trigger Input Signal TRG10
111	Input IN7	T0/T1 Trigger Input Signal TRG11

Note: Selection between the input lines and TRGxy is done via the edge selection control (register T2ES, encoding see [Table 6-5](#)).

General Purpose Timer Unit (GPTU)

Timer T2B External Input Selection Register T2BIS

The T2BIS register selects which of the external pins or trigger events from Timer T0/T1 is to be used for the various input functions for Timer T2B. This register is used only to select the inputs for Timer T2B in Split Mode; it is inactive in 32-bit mode. The selection is the same as for Timer T2A.

T2BIS

Timer T2B External Input Selection Register

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0					T2BIRC1			0	T2BIRC0			0	T2BICLR		
r					rw			r	rw			r	rw		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	T2BIUD			0	T2BISTP			0	T2BISTR			0	T2BICNT		
r	rw			r	rw			r	rw			r	rw		

Field	Bits	Type	Description
T2BICNT	[2:0]	rw	Timer T2B External Count Input Selection encoding see Table 6-4
T2BISTR	[6:4]	rw	Timer T2B External Start Input Selection encoding see Table 6-4
T2BISTP	[10:8]	rw	Timer T2B External Stop Input Selection encoding see Table 6-4
T2BIUD	[14:12]	rw	Timer T2B External Up/Down Input Selection encoding see Table 6-4
T2BICLR	[18:16]	rw	Timer T2B External Clear Input Selection encoding see Table 6-4
T2BIRC0	[22:20]	rw	Timer T2B External Reload/Capture 0 Input encoding see Table 6-4
T2BIRC1	[26:24]	rw	Timer T2B External Reload/Capture 1 Input encoding see Table 6-4
0	3, 7, 11,15, 19, 23, [31:27]	r	Reserved ; read as 0; writing to these bit positions has no effect.

General Purpose Timer Unit (GPTU)

Timer T2 External Input Edge Selection Register T2ES

This register selects the active edge of the external pin input for both Timer T2A and Timer T2B. [Table 6-5](#) lists the truth table for the edge selection bit fields.

T2ES

Timer 2 External Input Edge Selection Register

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	T2BERC1	T2BERC0	T2BECLR	T2BEUD	T2BESTP	T2BESTR	T2BECNT								
r	rw	rw	rw	rw	rw	rw	rw								

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	T2AERC1	T2AERC0	T2AECLR	T2AEUD	T2AESTP	T2AESTR	T2AECNT								
r	rw	rw	rw	rw	rw	rw	rw								

Field	Bits	Type	Description
T2AECNT	[1:0]	rw	Timer T2A External Count Input Active Edge Selection (encoding see Table 6-5)
T2AESTR	[3:2]	rw	Timer T2A External Start Input Active Edge Selection (encoding see Table 6-5)
T2AESTP	[5:4]	rw	Timer T2A External Stop Input Active Edge Selection (encoding see Table 6-5)
T2AEUD	[7:6]	rw	Timer T2A External Up/Down Input Active Edge Selection (encoding see Table 6-5)
T2AECLR	[9:8]	rw	Timer T2A External Clear Input Active Edge Selection (encoding see Table 6-5)
T2AERC0	[11:10]	rw	Timer T2A External Reload/Capture 0 Input Active Edge Selection (encoding see Table 6-5)
T2AERC1	[13:12]	rw	Timer T2A External Reload/Capture 1 Input Active Edge Selection (encoding see Table 6-5)
T2BECNT	[17:16]	rw	Timer T2B External Count Input Active Edge Selection (encoding see Table 6-5)
T2BESTR	[19:18]	rw	Timer T2B External Start Input Active Edge Selection (encoding see Table 6-5)
T2BESTP	[21:20]	rw	Timer T2B External Stop Input Active Edge Selection (encoding see Table 6-5)

General Purpose Timer Unit (GPTU)

Field	Bits	Type	Description
T2BEUD	[23:22]	rw	Timer T2B External Up/Down Input Active Edge Selection (encoding see Table 6-5)
T2BECLR	[25:24]	rw	Timer T2B External Clear Input Active Edge Selection (encoding see Table 6-5)
T2BERC0	[27:26]	rw	Timer T2B External Reload/Capture 0 Input Active Edge Selection (encoding see Table 6-5)
T2BERC1	[29:28]	rw	Timer T2B External Reload/Capture 1 Input Active Edge Selection (encoding see Table 6-5)
0	[15:14], [31:30]	r	Reserved ; read as 0; writing to these bit positions has no effect.

Table 6-5 T2 Input Source Active Edge Selection

Value	Selected Active Edge	Selected Active Input for T2AECNT, T2AERC1, T2AIRC0; and T2BECNT, T2BERC1, and T2BERC0
00	None.	Input is connected to T0/T1 Trigger Input Signal TRGxy as selected in T2xIS
01	Positive edge	—
10	Negative edge	—
11	Both edges	—

General Purpose Timer Unit (GPTU)

6.2.2.2 Mode Control and Status Register

Two registers control the mode of operation for the timer and the reload/capture registers. They also provide status information. The first register, T2CON, controls the operation of the timer itself and holds the status information, while the second register, T2RCCON, controls the operation of the two reload/capture registers.

The T2CON register controls the operating mode of Timer T2. The control bits and functions are the same for Timer T2A and Timer T2B.

T2CON

Timer 2 Mode Control and Status Register

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0			T2B DIR	0			T2B COS	T2BCOV	T2BCCLR		T2BCDIR		T2BCSRC		
r			rw	r			rw	rw	rw		rw		rw		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
T2S PLIT	0		T2A DIR	0			T2A COS	T2ACOV	T2ACCLR		T2ACDIR		T2ACSRC		
rw	r		rw	r			rw	rw	rw		rw		rw		

Field	Bits	Type	Description
T2ACSRC	[1:0]	rw	Timer T2A Count Input Source Control encoding see Table 6-9
T2ACDIR	[3:2]	rw	Timer T2A Direction Control encoding see Table 6-8
T2ACCLR	[5:4]	rw	Timer T2A Clear Control encoding see Table 6-7
T2ACOV	[7:6]	rw	Timer T2A Overflow/Underflow Generation Control (encoding see Table 6-6)
T2ACOS	8	rw	Timer T2A One-Shot Control 0 T2A continues to run after overflow or underflow 1 T2A stops after the first overflow or underflow
T2ADIR	12	rw	Timer T2A Direction Status Flag 0 T2A Direction is up-counting 1 T2A Direction is down-counting

General Purpose Timer Unit (GPTU)

Field	Bits	Type	Description
T2SPLIT	15	rw	Timer T2 Split Control 0 Timer T2 operates as one 32-bit timer, controlled via T2A controls 1 Timer T2 operates as two independent 16-bit timers T2A and T2B
T2BCSRC	[17:16]	rw	Timer T2B Count Input Source Control encoding see Table 6-9
T2BCDIR	[19:18]	rw	Timer T2B Direction Control encoding see Table 6-8
T2BCCLR	[21:20]	rw	Timer T2B Clear Control encoding see Table 6-7
T2BCOV	[23:22]	rw	Timer T2B Overflow/Underflow Generation Control (encoding see Table 6-6)
T2BCOS	24	rw	Timer T2B One-Shot Control 0 T2B continues to run after overflow or underflow 1 T2B stops after the first overflow or underflow
T2BDIR	28	rw	Timer T2B Direction Status Flag 0 T2B direction is up-counting 1 T2B direction is down-counting
0	[11:9], [14:13], [27:25], [31:29]	r	Reserved ; read as 0; writing to these bit positions has no effect.

Table 6-6 T2 Overflow/Underflow Generation Control

T2BCOV T2ACOV	Selected Function
00	Overflow is generated for FF..FF _H -> 00..00 _H ; Underflow is generated for 00..00 _H -> FF..FF _H
01	Overflow is generated for FF..FE _H -> FF..FF _H ;; underflow is generated for 00..00 _H -> FF..FF _H
10	Overflow is generated for FF..FF _H -> 00..00 _H ; underflow is generated for 00..01 _H -> 00..00 _H
11	Overflow is generated for FF..FE _H -> FF..FF _H ;; Underflow is generated for 00..01 _H -> 00..00 _H

General Purpose Timer Unit (GPTU)

Table 6-7 T2 Clear Control

T2BCCLR T2ACCLR¹⁾	Selected Function
00	Clear timer to 00..00 _H on external event (Clear_B/Clear_A)
01	Clear timer on capture 0 event (CP0_T2B/CP0_T2A)
10	Clear timer on capture 1 event (CP1_T2B/CP1_T2A)
11	Reserved. Do not use this combination.

¹⁾ In Clear-on-Capture mode, the timer contents are first captured, then the timer is cleared.

Table 6-8 T2 Direction Control

T2BCDIR T2ACDIR	Selected Function_B¹⁾
00	Count direction is count up (software controlled).
01	Count direction is count down (software controlled).
10²⁾	Count direction controlled through external signal (UpDown_B / UpDown_A). Count up if external signal is 1, else count down.
11²⁾	Count direction controlled through external signal (UpDown_B / UpDown_A). Count down if external signal is 1, else count up.

¹⁾ If Quadrature Counting is selected, the count direction is controlled through the relation of the two signals Count_A/B and Up/Down A/B; the bit fields T2ACDIR/T2BCDIR have no effect in this case.

²⁾ The last two options have an extra line going from the input selection to the direction control representing the state of the input (not shown in the diagrams). The edge selection has no effect on the direction control; however, it can be used to generate a service request (UpDown_A only).

Table 6-9 T2 Count Input Source Control

T2BCSRC T2ACSRC	Selected Function
00	Count input source is the module clock f_{GPTU} .
01	Count input source is external count input Count_x.
10	Quadrature Counter Mode. Count input sources are the two inputs Count_x and UpDown_x.
11	Reserved. Do not use this combination.

6.2.2.3 Timer T0/T1/T2 Run Control Register

The run control bits of the individual parts of timers T0, T1, and T2 are all contained in register T012RUN. This register allows synchronous starting or stopping of several or all timers with one instruction.

T012RUN

Timer T0, T1, and T2 Run Control Register

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	T2B CLR R	T2B SET R	T2B RUN	0	T2A CLR R	T2A SET R	T2A RUN	T1D RUN	T1C RUN	T1B RUN	T1A RUN	T0D RUN	T0C RUN	T0B RUN	T0A RUN
r	w	w	rh	r	w	w	rh	rw	rw	rw	rw	rw	rw	rw	rw

Field	Bits	Type	Description
T0ARUN	0	rw	Timer T0A Run Control 0 Stop T0A 1 Start T0A
T0BRUN	1	rw	Timer T0B Run Control 0 Stop T0B 1 Start T0B
T0CRUN	2	rw	Timer T0C Run Control 0 Stop T0C 1 Start T0C
T0DRUN	3	rw	Timer T0D Run Control 0 Stop T0D 1 Start T0D
T1ARUN	4	rw	Timer T1A Run Control 0 Stop T1A 1 Start T1A
T1BRUN	5	rw	Timer T1B Run Control 0 Stop T1B 1 Start T1B

General Purpose Timer Unit (GPTU)

Field	Bits	Type	Description
T1CRUN	6	rw	Timer T1C Run Control 0 Stop T1C 1 Start T1C
T1DRUN	7	rw	Timer T1D Run Control 0 Stop T1D 1 Start T1D
T2ARUN	8	rh	Timer T2A Run Status Flag 0 T2A is stopped 1 T2A is running This bit indicates the running/stopped status of Timer T2A. This status bit can be directly set or reset by hardware depending on the selections and external events causing a start or a stop of the timer. It can only be affected by software through the set and clear bits T2ASET and T2ACLRR, respectively. Writing directly to this bit via software has no effect.
T2ASET	9	w	Timer T2A Run Set Bit Writing a 1 to this bit causes the run bit T2ARUN to be set to 1, thus starting Timer T2A. Possible hardware modifications of T2ARUN that occurred during read-modify-write instructions (for example, bit set, bit clear instructions) are lost; the software modification has priority. The value written to T2ASET is not stored. Writing a 0 to this bit has no effect. This bit always returns 0 when read. If both T2ASET and T2ACLRR are set, T2ARUN is not affected.
T2ACLRR	10	w	Timer T2A Run Clear Bit Writing a 1 to this bit causes the run bit T2ARUN to be cleared, thus stopping timer T2A. Possible hardware modifications of T2ARUN that occurred during read-modify-write instructions (for example, bit set, bit clear instructions) are lost; the software modification has priority. The value written to T2ACLRR is not stored. Writing a 0 to this bit has no effect. This bit always returns 0 when read. If both T2ASET and T2ACLRR are set, T2ARUN is not affected.

General Purpose Timer Unit (GPTU)

Field	Bits	Type	Description
T2BRUN	12	rh	Timer T2B Run Status Flag 0 T2B is stopped 1 T2B is running More details see description for T2ARUN.
T2BSETR	13	w	Timer T2B Run Set Bit More details see description for T2ASETR.
T2BCLRR	14	w	Timer T2B Run Clear Bit More details see description for T2ACLRR.
0	11, [31:15]	r	Reserved ; read as 0; writing to these bit positions has no effect.

6.2.2.4 T2 Reload/Capture Mode Control Register

This register selects the reload/capture mode operation for the reload/capture registers T2ARC0, T2ARC1, T2BRC0, and T2BRC1.

T2RCCON

Timer 2 Reload/Capture Mode Control Register

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0									T2BMRC1			0	T2BMRC0		
r									rw			r	rw		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0									T2AMRC1			0	T2AMRC0		
r									rw			r	rw		

Field	Bits	Type	Description
T2AMRC0	[2:0]	rw	Timer T2A Reload/Capture 0 Mode Control encoding see Table 6-10
T2AMRC1	[6:4]	rw	Timer T2A Reload/Capture 1 Mode Control encoding see Table 6-10

General Purpose Timer Unit (GPTU)

Field	Bits	Type	Description
T2BMRC0	[18:16]	rw	Timer T2B Reload/Capture 0 Mode Control encoding see Table 6-10
T2BMRC1	[22:20]	rw	Timer T2B Reload/Capture 1 Mode Control encoding see Table 6-10
0	3, 19, [15:7], [31:23]	r	Reserved ; read as 0; writing to these bit positions has no effect.

Table 6-10 T2 Capture/Reload Mode Selection

T2AMRCx T2BMRCx	Selected Operation for T2ARC0/T2BRC0	Selected Operation for T2ARC1/T2BRC1
000	Disabled	
001	Reserved. Do not use this combination.	
010	Reserved. Do not use this combination.	
011	Capture on external event	
100	Reload on overflow or underflow	
101	Reload on external event	
110	Reload on overflow only	Reload on underflow only
111	Reload on external event if count direction is up (if T2ADIR/T2BDIR = 0)	Reload on external event if count direction is down (T2ADIR/T2BDIR = 1)

Note: If a capture event for one register and a reload event for the other register occur at the same time, the timer contents are captured first; then, the timer is reloaded. If both reload/capture registers are set up for reload and the trigger events occur at the same time for both, only the reload from the higher numbered register (T2ARC1/T2BRC1) is performed.

6.2.2.5 Timer T2 Count and Reload/Capture Registers

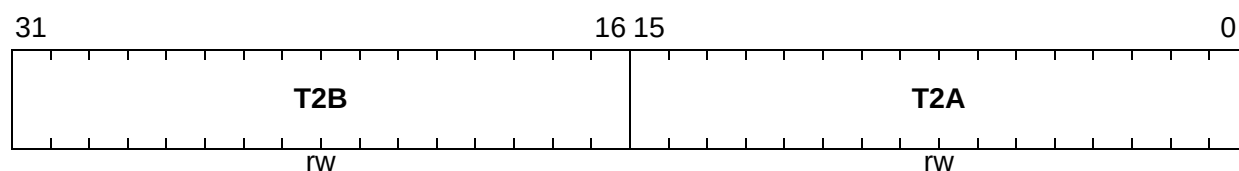
Timer T2 Count Register

Register T2 holds the actual count value of Timer T2. In Split Mode, the lower half-word of this register represents the contents of Timer T2A, while the upper half-word represents the contents of Timer T2B. Proper load/store instructions must be used depending on whether the timer is operated in full 32-bit or in Split Mode.

T2

Timer T2 Count Register

Reset Value: 0000 0000_H



Field	Bits	Type	Description
T2A	[15:0]	rw	T2A Contents (in Split Mode)
T2B	[31:16]	rw	T2B Contents (in Split Mode)

General Purpose Timer Unit (GPTU)

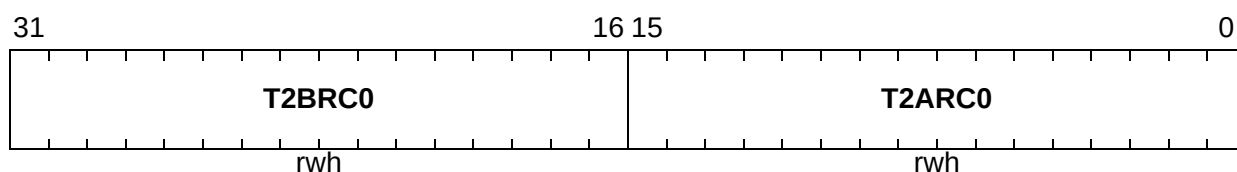
Timer T2 Reload/Capture Registers

The two reload/capture values for Timer T2 are held in registers T2RC0 and T2RC1, respectively. In Split Mode, the lower half-word of these registers represent the respective Timer T2A reload/capture values (T2ARC0, T2ARC1), while the upper half-word is used for the Timer T2B reload/capture values (T2BRC0, T2BRC1). The same access mechanisms apply here as for the timer count register Timer T2.

T2RC0

Timer T2 Reload/Capture Register 0

Reset Value: 0000 0000_H

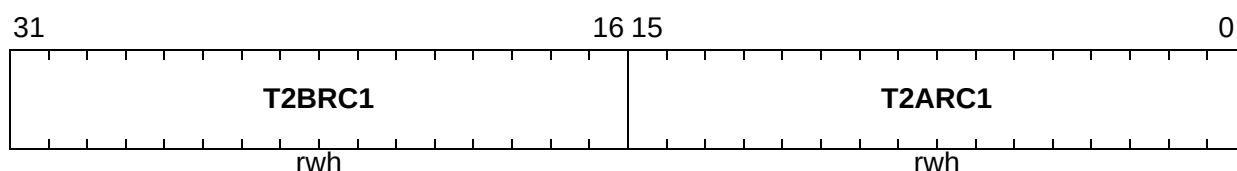


Field	Bits	Type	Description
T2ARC0	[15:0]	rwh	T2A Reload/Capture Value (in Split Mode) In Capture Mode, the register contents are also affected by hardware.
T2BRC0	[31:16]	rwh	T2B Reload/Capture Value (in Split Mode) In Capture Mode, the register contents are also affected by hardware.

T2RC1

Timer T2 Reload/Capture Register 1

Reset Value: 0000 0000_H



Field	Bits	Type	Description
T2ARC1	[15:0]	rwh	T2A Reload/Capture Value (in Split Mode) In Capture Mode, the register contents are also affected by hardware.
T2BRC1	[31:16]	rwh	T2B Reload/Capture Value (in Split Mode) In Capture Mode, the register contents are also affected by hardware.

6.2.3 Global Control Registers

The OSEL register selects the output source function for the output state bits.

OSEL

Output Source Selection Register

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	SO7			0	SO6			0	SO5			0	SO4		
r	rw			r	rw			r	rw			r	rw		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	SO3			0	SO2			0	SO1			0	SO0		
r	rw			r	rw			r	rw			r	rw		

Field	Bits	Type	Description
SO0	[2:0]	rw	Output 0 Source Selection see Table 6-11 for encoding
SO1	[6:4]	rw	Output 1 Source Selection encoding see Table 6-11
SO2	[10:8]	rw	Output 2 Source Selection encoding see Table 6-11
SO3	[14:12]	rw	Output 3 Source Selection encoding see Table 6-11
SO4	[18:16]	rw	Output 4 Source Selection encoding see Table 6-11
SO5	[22:20]	rw	Output 5 Source Selection encoding see Table 6-11
SO6	[26:24]	rw	Output 6 Source Selection encoding see Table 6-11
SO7	[30:28]	rw	Output 7 Source Selection encoding see Table 6-11
0	3, 7, 11, 15, 19, 23, 27, 31	r	Reserved ; read as 0; writing to these bit positions has no effect.

General Purpose Timer Unit (GPTU)

Table 6-11 T2 Output Signal Source Selection

Value	Selected Source
000	OUT00
001	OUT01
010	OUT10
011	OUT11
100	OUV_T2A
101	OUV_T2B
110	Reserved. Do not use these combinations.
111	

Each output has an output state bit, OUTx. These bits toggle each time a trigger signal occurs. The state of these bits can be made available at the respective output pins through the alternate function selections at these pins. The output state bits and the enable bits are contained in the output control register OUT.

The output state bits can also be modified by software. Individual set and clear bits are provided for each of the output state bits. Software can update a state bit via these separate bits only.

OUT

Output Register

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0								SET 07	SET 06	SET 05	SET 04	SET 03	SET 02	SET 01	SET 00
r								w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CLR 07	CLR 06	CLR 05	CLR 04	CLR 03	CLR 02	CLR 01	CLR 00	OUT 7	OUT 6	OUT 5	OUT 4	OUT 3	OUT 2	OUT 1	OUT 0
w	w	w	w	w	w	w	w	rh	rh	rh	rh	rh	rh	rh	rh

This SRSEL register selects which of the various events in the Timer T0, T1, and T2 blocks generate one of the eight service requests.

General Purpose Timer Unit (GPTU)

Field	Bits	Type	Description
OUTx (x = 7-0)	[7:0]	rh	Output x Status Bit This status bit can be directly set or reset by the associated trigger event. It can be set or reset only by software via writing a 1 to either bit SETOx or bit CLROx, respectively. Writing directly to this bit via software has no effect.
CLROx (x = 7-0)	[15:8]	w	Output x Clear Bit Writing a 1 to this bit causes the output bit OUTx to be cleared. Possible hardware modifications of OUTx that occurred during read-modify-write instructions (for example, bit set, bit clear instructions) are lost; the software modification has priority. The value written to CLROx is not stored. Writing a 0 to this bit has no effect. This bit always returns 0 when read. If both SETOx and CLROx are set, OUTx is not affected.
SETOx (x = 7-0)	[23:16]	w	Output x Set Bit Writing a 1 to this bit causes the output bit OUTx to be set to 1. Possible hardware modifications of OUTx that occurred during read-modify-write instructions (for example, bit set, bit clear instructions) are lost; the software modification has priority. The value written to SETOx is not stored. Writing a 0 to this bit has no effect. This bit always returns 0 when read. If both SETOx and CLROx are set, OUTx is not affected.
0	[31:24]	r	Reserved ; read as 0; writing to these bit positions has no effect.

SRSEL

Service Request Source Selection Register

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SSR0				SSR1				SSR2				SSR3			
rw				rw				rw				rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSR4				SSR5				SSR6				SSR7			
rw				rw				rw				rw			

General Purpose Timer Unit (GPTU)

Field	Bits	Type	Description
SSR7	[3:0]	rw	Service Request Node 7 Source Selection encoding see Table 6-12
SSR6	[7:4]	rw	Service Request Node 6 Source Selection encoding see Table 6-12
SSR5	[11:8]	rw	Service Request Node 5 Source Selection encoding see Table 6-12
SSR4	[15:12]	rw	Service Request Node 4 Source Selection encoding see Table 6-12
SSR3	[19:16]	rw	Service Request Node 3 Source Selection encoding see Table 6-12
SSR2	[23:20]	rw	Service Request Node 2 Source Selection encoding see Table 6-12
SSR1	[27:24]	rw	Service Request Node 1 Source Selection encoding see Table 6-12
SSR0	[31:28]	rw	Service Request Node 0 Source Selection encoding see Table 6-12

General Purpose Timer Unit (GPTU)

Table 6-12 T2 Service Request Source Selection

Value	Selected Source
0000	Start_A
0001	Stop_A
0010	UpDown_A
0011	Clear_A
0100	RLCP0_A
0101	RLCP1_A
0110	OUV_T2A
0111	OUV_T2B
1000	Start_B
1001	Stop_B
1010	RLCP0_B
1011	RLCP1_B
1100	SR00
1101	SR01
1110	SR10
1111	SR11

6.3 GPTU Module Implementation

This section describes the GPTU Module interfaces with the clock control, port connections, interrupt control, and address decoding.

6.3.1 Interfaces of the GPTU Modules

Figure 6-16 shows the TC1100 specific implementation details and interconnections of the GPTU Module. The GPTU Module has eight I/O lines located at Port 0. Further, the GPTU Module is supplied by a separate clock control, interrupt control and address decoding logic.

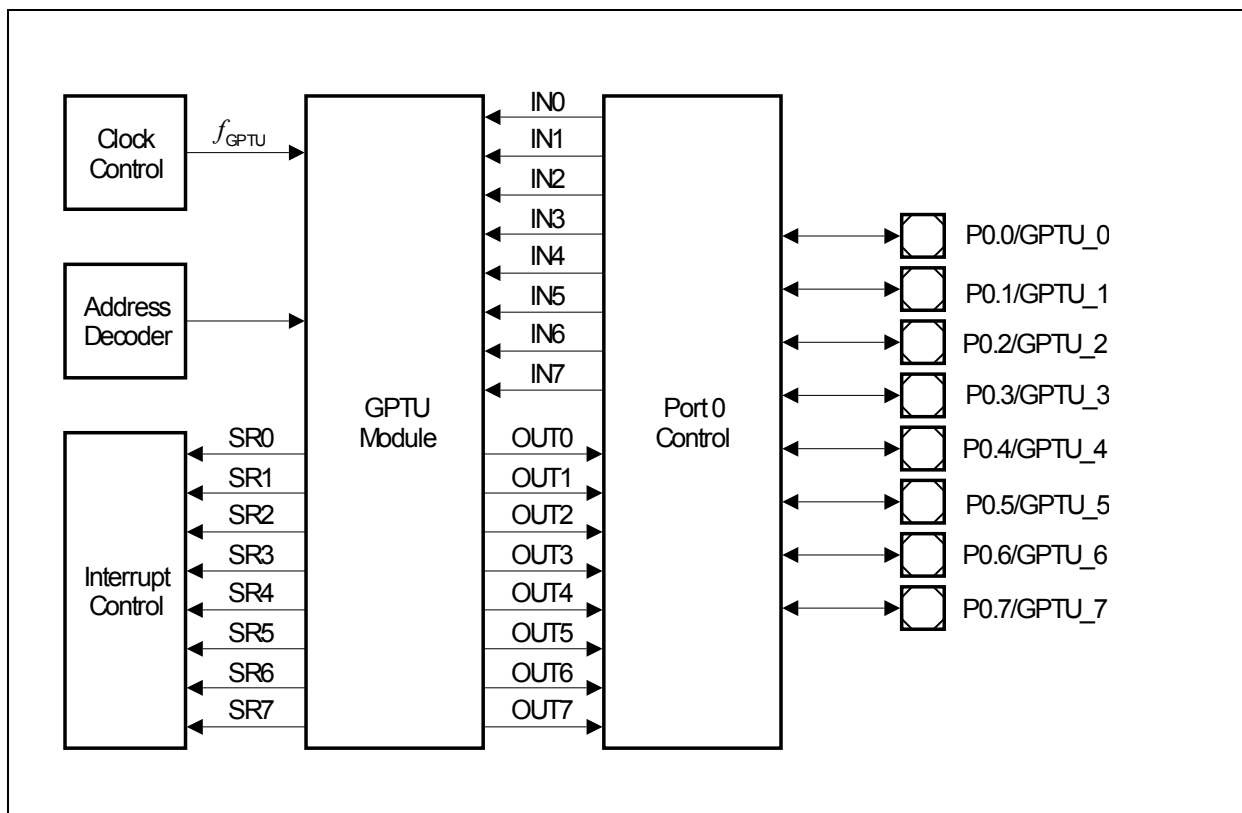


Figure 6-16 GPTU Module Implementation and Interconnections

6.3.2 GPTU Module Related External Registers

Figure 6-17 summarizes the module related external registers which are required for GPTU programming (see also **Figure 6-15** for the module kernel specific registers).

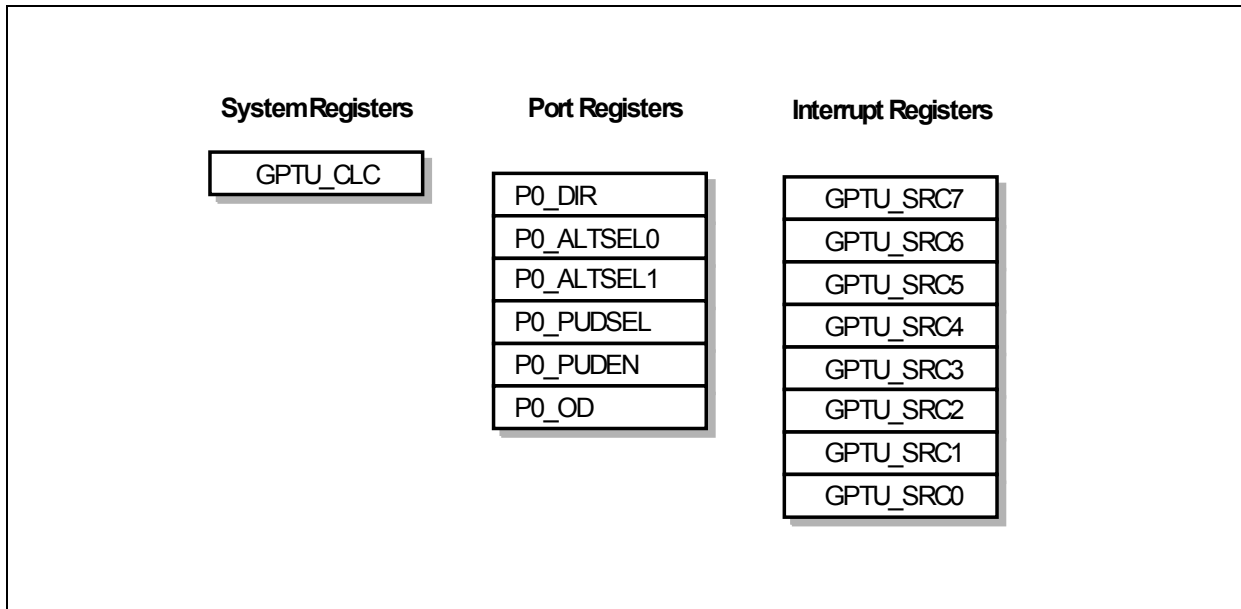


Figure 6-17 GPTU Implementation Specific Special Function Registers

General Purpose Timer Unit (GPTU)

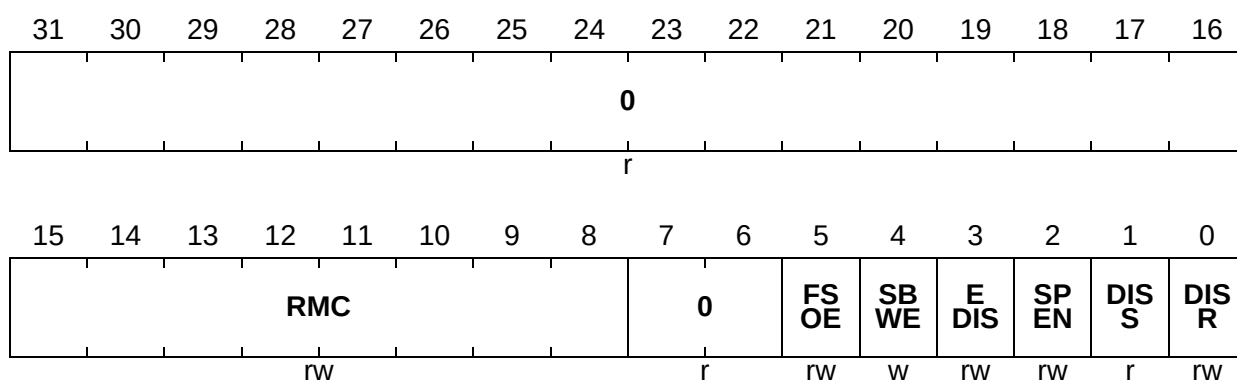
6.3.2.1 Clock Control Registers

The clock control register allows the programmer to adapt the functionality and power consumption of a GPTU Module to the requirements of the application. The diagram below shows the clock control register functionality as implemented for the GPTU Module.

GPTU_CLC

GPTU Clock Control Register

Reset Value: 0000 0002_H



Field	Bits	Type	Description
DISR	0	rw	Module Disable Request Bit Used for enable/disable control of the module.
DISS	1	r	Module Disable Status Bit Bit indicates the current status of the module.
SPEN	2	rw	Module Suspend Enable for OCDS Used for enabling the suspend mode.
EDIS	3	rw	External Request Disable Used for controlling the external clock disable request.
SBWE	4	w	Module Suspend Bit Write Enable for OCDS Defines whether SPEN and FSOE are write protected.
FSOE	5	rw	Fast Switch Off Enable Used for fast clock switch off in OCDS suspend mode.
RMC	[15:8]	rw	8-Bit Clock Divider Value in RUN Mode
0	[7:6], [31:16]	r	Reserved ; read as 0; should be written with 0.

6.3.2.2 Port Control

The interconnections between the GPTU modules and the port I/O lines are controlled in the port logics. The following port control operations selections must be executed:

- Input/output direction selection (DIR registers)
- Alternate function selection (ALTSEL0 and ALTSEL1 registers)
- Input/Output driver characteristic control (PUDSEL, PUDEN and OD registers)

Input/Output Function Selection

The port input/output control registers contain the bit fields that select the digital output and input driver characteristics such as pull-up/down devices, port direction (input/output), open-drain, and alternate output selections. The I/O lines for the GPTU modules are controlled by the port input/output control registers of Port0.

Table 6-13 shows how bits and bit fields must be programmed for the required I/O functionality of the GPTU I/O lines.

General Purpose Timer Unit (GPTU)
Table 6-13 GPTU I/O Line Selection and Setup

Module	Port Lines	Input/Output Control Register Bits	I/O
GPTU	P0.0/GPTU_0	P0_DIR.P0 = 0 _B	Input
		P0_DIR.P0 = 1 _B	Output
		P0_ALTSEL0.P0 = 1 _B	
		P0_ALTSEL1.P0 = 0 _B	
	P0.1/GPTU_1	P0_DIR.P1 = 0 _B	Input
		P0_DIR.P1 = 1 _B	Output
		P0_ALTSEL0.P1 = 1 _B	
		P0_ALTSEL1.P1 = 0 _B	
	P0.2/GPTU_2	P0_DIR.P2 = 0 _B	Input
		P0_DIR.P2 = 1 _B	Output
		P0_ALTSEL0.P2 = 1 _B	
		P0_ALTSEL1.P2 = 0 _B	
	P0.3/GPTU_3	P0_DIR.P3 = 0 _B	Input
		P0_DIR.P3 = 1 _B	Output
		P0_ALTSEL0.P3 = 1 _B	
		P0_ALTSEL1.P3 = 0 _B	
	P0.4/GPTU_4	P0_DIR.P4 = 0 _B	Input
		P0_DIR.P4 = 1 _B	Output
		P0_ALTSEL0.P4 = 1 _B	
		P0_ALTSEL1.P4 = 0 _B	
	P0.5/GPTU_5	P0_DIR.P5 = 0 _B	Input
		P0_DIR.P5 = 1 _B	Output
		P0_ALTSEL0.P5 = 1 _B	
		P0_ALTSEL1.P5 = 0 _B	
	P0.6/GPTU_6	P0_DIR.P6 = 0 _B	Input
		P0_DIR.P6 = 1 _B	Output
		P0_ALTSEL0.P6 = 1 _B	
		P0_ALTSEL1.P6 = 0 _B	

General Purpose Timer Unit (GPTU)

Table 6-13 GPTU I/O Line Selection and Setup (cont'd)

Module	Port Lines	Input/Output Control Register Bits	I/O
	P0.7/GPTU_7	P0_DIR.P7 = 0 _B	Input
		P0_DIR.P7 = 1 _B	Output
		P0_ALTSEL0.P7 = 1 _B	
		P0_ALTSEL1.P7 = 0 _B	

P0_DIR

Port 0 Direction Register

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
P15	P14	P13	P12	P11	P10	P9	P8	P7	P6	P5	P4	P3	P2	P1	P0
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

Field	Bits	Type	Description
Pn (n = 0-7)	n	rW	Port 0 Pin 0 - 7 Direction Control ¹⁾ 0 Direction is set to input (default after reset) 1 Direction is set to output
0	[31:16]	r	Reserved ; read as 0; should be written with 0.

¹⁾ Shaded bits and bit field are don't care for GPTU I/O port control

General Purpose Timer Unit (GPTU)

P0_ALTSELn (n = 1, 0)

Port 0 Alternate Select Register

Reset Value: 0000 0000_H

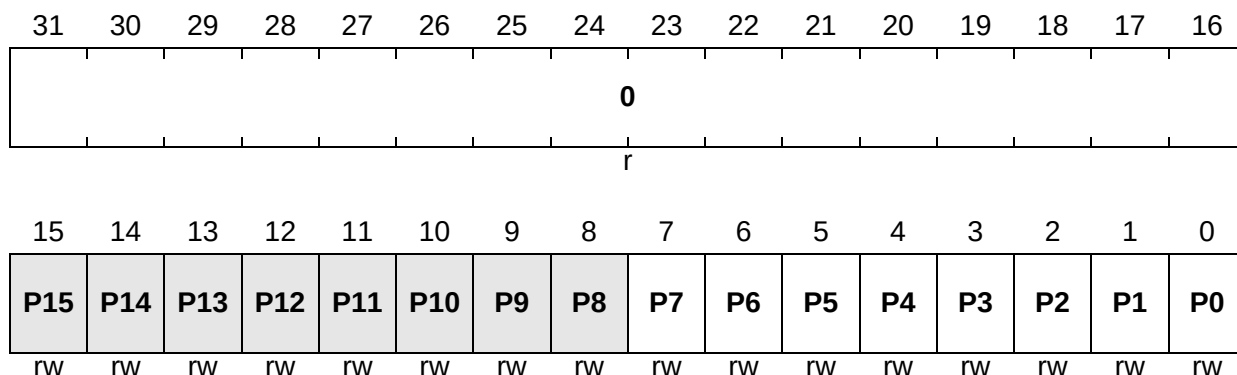


Table 6-14 Function of the Bits P0_ALTSEL0.Pn and P0_ALTSEL1.Pn (n = 0-7)¹⁾

P0_ALTSEL0.Pn	P0_ALTSEL1.Pn	Function
1	0	Alternate Select 1

¹⁾ Shaded bits and bit field are don't care for GPTU I/O port control

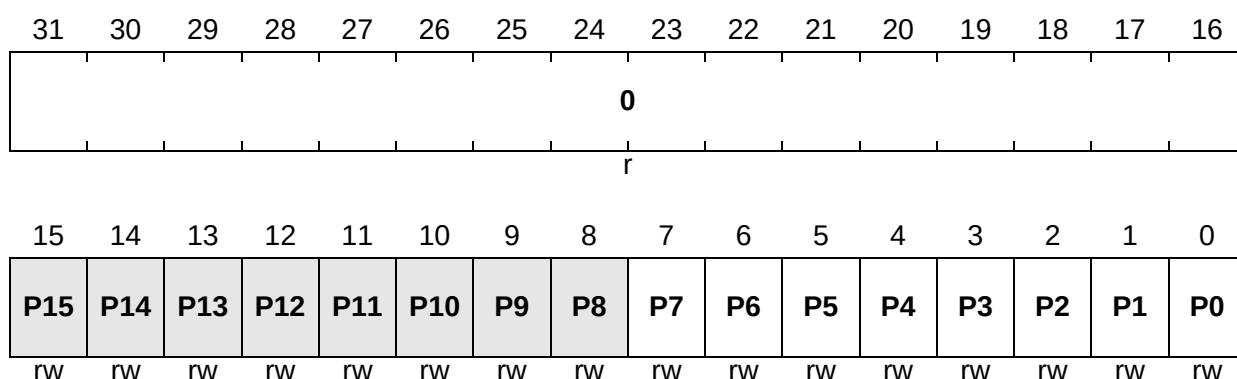
The GPTU ports also offer the possibility to configure the following output characteristics:

- push/pull (optional pull-up/pull-down)
- open drain with internal pull-up
- open drain with external pull-up

P0_PUDSEL

Port 0 Pull-Up/Pull-Down Select Register

Reset Value: 0000 FFFF_H



General Purpose Timer Unit (GPTU)

Field	Bits	Type	Description
Pn (n = 0-7)	n	rw	Pull-Up/Pull-Down Select Port 0 Bit n¹⁾ 0 Pull-down device is selected 1 Pull-up device is selected
0	[31:16]	r	Reserved ; read as 0; should be written with 0.

¹⁾ Shaded bits and bit field are don't care for GPTU I/O port control

P0_PUDEN

Port 0 Pull-Up/Pull-Down Enable Register

Reset Value: 0000 FFFF_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
P15	P14	P13	P12	P11	P10	P9	P8	P7	P6	P5	P4	P3	P2	P1	P0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Field	Bits	Type	Description
Pn (n = 0-7)	n	rw	Pull-Up/Pull-Down Enable at Port 0 Bit n¹⁾ 0 Pull-up or Pull-down device is disabled 1 Pull-up or Pull-down device is enabled
0	[31:16]	r	Reserved ; read as 0; should be written with 0.

¹⁾ Shaded bits and bit field are don't care for GPTU I/O port control

P0_OD

Port 0 Open Drain Control Register

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
P15	P14	P13	P12	P11	P10	P9	P8	P7	P6	P5	P4	P3	P2	P1	P0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

General Purpose Timer Unit (GPTU)

Field	Bits	Type	Description
Pn (n = 0-7)	n	rw	Port 0 Pin n Open Drain Mode¹⁾ 0 Normal Mode, output is actively driven for 0 and 1 state 1 Open Drain Mode, output is actively driven only for 0 state
0	[31:16]	r	Reserved ; read as 0; should be written with 0.

¹⁾ Shaded bits and bit field are don't care for GPTU I/O port control

6.3.2.3 Interrupt Registers

The eight interrupt outputs SR7 - SR0 of each GPTU Module are controlled by the service request control registers GPTU_SRC7 to GPTU_SRC0.

GPTU_SRC0

GPTU Interrupt Service Request Control Register 0

GPTU_SRC1

GPTU Interrupt Service Request Control Register 1

GPTU_SRC2

GPTU Interrupt Service Request Control Register 2

GPTU_SRC3

GPTU Interrupt Service Request Control Register 3

GPTU_SRC4

GPTU Interrupt Service Request Control Register 4

GPTU_SRC5

GPTU Interrupt Service Request Control Register 5

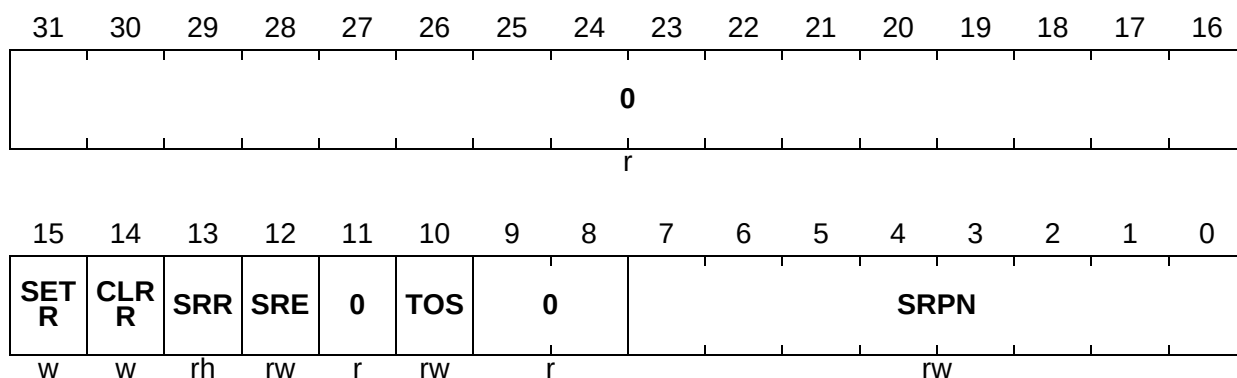
GPTU_SRC6

GPTU Interrupt Service Request Control Register 6

GPTU_SRC7

GPTU Interrupt Service Request Control Register 7

Reset Values: 0000 0000_H



Field	Bits	Type	Description
SRPN	[7:0]	rw	Service Request Priority Number
TOS	10	rw	Type of Service Control
SRE	12	rw	Service Request Enable
SRR	13	rh	Service Request Flag
CLRR	14	w	Request Clear Bit
SETR	15	w	Request Set Bit

General Purpose Timer Unit (GPTU)

Field	Bits	Type	Description
0	[9:8], 11 [31:16]	r	Reserved ; read as 0; should be written with 0.

Note: Further details on interrupt handling and processing are described in the “Interrupt System” chapter of the TC1100 System Units User’s Manual.

6.3.3 GPTU Register Address Ranges

In the TC1100, the registers of the GPTU Module is located in the following address ranges:

- Module Base Address = F000 0600_H
- Module End Address = F000 06FF_H
- Absolute Register Address = Module Base Address + Offset Address
(offset addresses see [Table 6-2](#))

Note: The complete and detailed address map of the GPTU module is described in the chapter “Register Overview” of the TC1100 System Units User’s Manual.

7 Capture/Compare Unit 6 (CCU6)

This chapter describes the Capture/Compare Unit 6 Module (CCU61) of the TC1100. It contains the following sections:

- Functional description of a CCU6 kernel (see [Section 7.1](#))
- CCU6 kernel register descriptions (see [Section 7.2](#))
- TC1100 implementation specific details and registers of the CCU6 module (port connections and control, interrupt control, address decoding, and clock control, see [Section 7.3](#))

Note: The CCU6 kernel register names described in [Section 7.2](#) will be referenced in the TC1100 User's Manual by the module name prefix "CCU61_" for the CCU61 interface.

7.1 CCU6 Kernel Description

7.1.1 Overview

The CCU6 provides two independent timers (T12, T13), which can be used for PWM generation, especially for AC-motor control. Additionally, special control modes for block commutation and multi-phase machines are supported.

The timer T12 can work in capture and/or compare mode for its three channels. The modes can also be combined. The timer T13 can work in compare mode only. The multi-channel control unit generates output patterns which can be modulated by T12 and/or T13. The modulation sources can be selected and combined for the signal modulation.

Timer 12 Features

- Three capture/compare channels, each channel can be used either as capture or as compare channel.
- Generation of a three-phase PWM supported (six outputs, individual signals for highside and lowside switches)
- 16 bit resolution, maximum count frequency = peripheral clock
- Dead-time control for each channel to avoid short-circuits in the power stage
- Concurrent update of the required T12/T13 registers
- Center-aligned and edge-aligned PWM can be generated
- Single-shot mode supported
- Many interrupt request sources
- Hysteresis-like control mode

Timer 13 Features

- One independent compare channel with one output
- 16 bit resolution, maximum count frequency = peripheral clock
- Can be synchronized to T12
- Interrupt generation at period-match and compare-match
- Single-shot mode supported

Additional Features

- Block commutation for Brushless DC-drives implemented
- Position detection via Hall-sensor pattern
- Automatic rotational speed measurement for block commutation
- Integrated error handling
- Fast emergency stop without CPU load via external signal ($\overline{\text{CTRAP}}$)
- Control modes for multi-channel AC-drives
- Output levels can be selected and adapted to the power stage

Capture/Compare Unit 6 (CCU6)

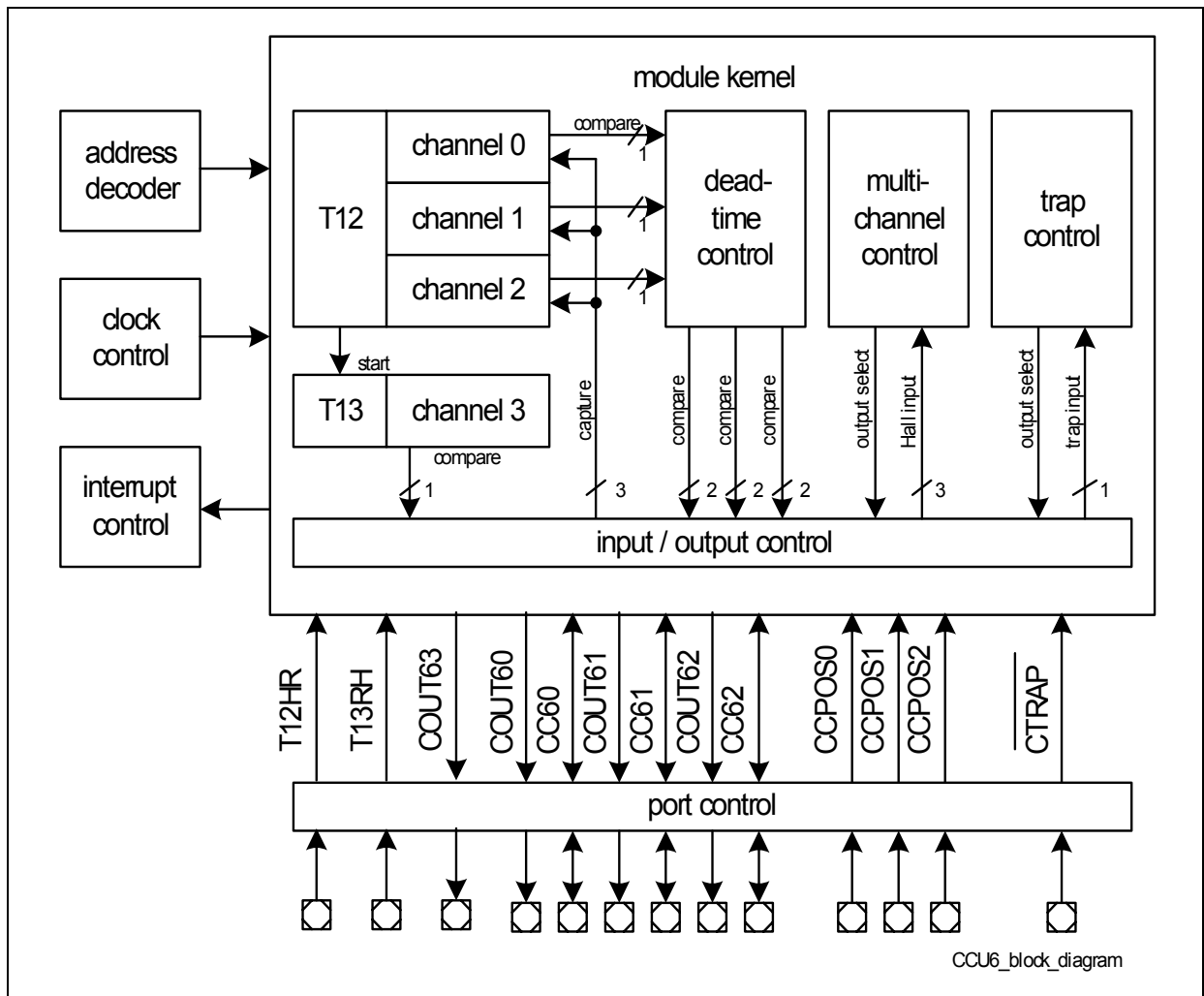


Figure 7-1 CCU6 Block Diagram

7.1.2 Timer T12

7.1.2.1 Overview

The timer T12 is used for capture/compare purposes with three independent channels. The timer T12 is a 16-bit wide counter. Three channel registers (CC60R, CC61R, CC62R), which are built with shadow registers (CC60SR, CC61SR, CC62SR), contain the compare value or the captured timer value. In compare mode, the software writes to the shadow registers and their contents are transferred simultaneously to the actual compare registers during the T12 shadow transfer. In capture mode, the captured value of T12 can be read from the channel registers. The period of the timer T12 is fixed by the period register T12PR, which is also built with a shadow register.

The write access from the CPU targets the corresponding shadow registers, whereas the read access targets the registers actually used (except for the three compare channels, where the actual and the shadow registers can be read).

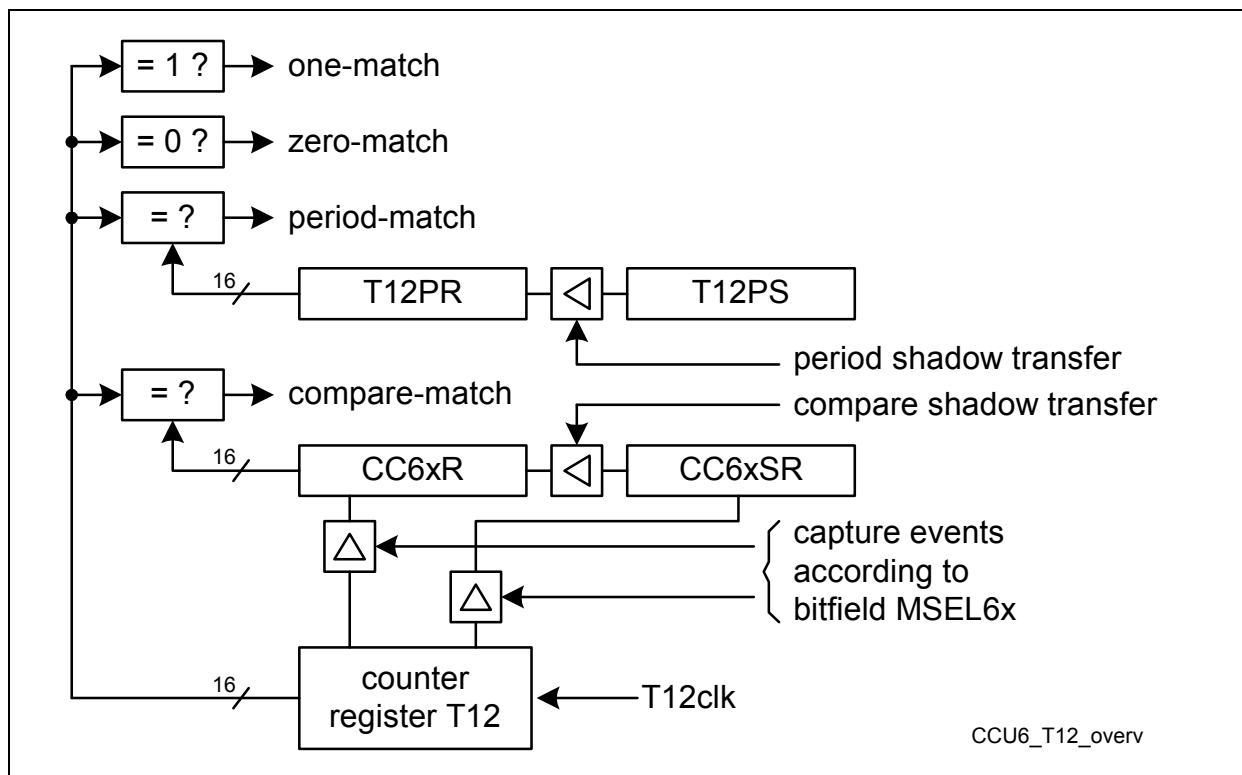


Figure 7-2 T12 Overview

While timer T12 is running, write accesses to register T12 are not taken into account. If the timer T12 is stopped and the dead-time counters are 0, write actions to register T12 are immediately taken into account.

7.1.2.2 Counting Rules

Referring to T12 input clock the counting sequence is defined by the following counting rules:

T12 in edge-aligned mode:

- The counter is reset to zero and (if desired) the T12 shadow transfer takes place if the period-match is detected. The counting direction is always upwards.

T12 in center-aligned mode:

- The count direction is set to counting up (CDIR = 0) if the one-match is detected while counting down.
- The count direction is set to counting down (CDIR = 1) if the period-match is detected while counting up.
- The counter counts up while CDIR = 0 and it counts down while CDIR = 1.
- If enabled shadow transfer takes place:
 - if the period-match is detected while counting up
 - if the one-match is detected while counting down

The timer T12 prescaler is reset while T12 is not running to ensure reproducible timings and delays.

The counting rules lead to the following sequences:

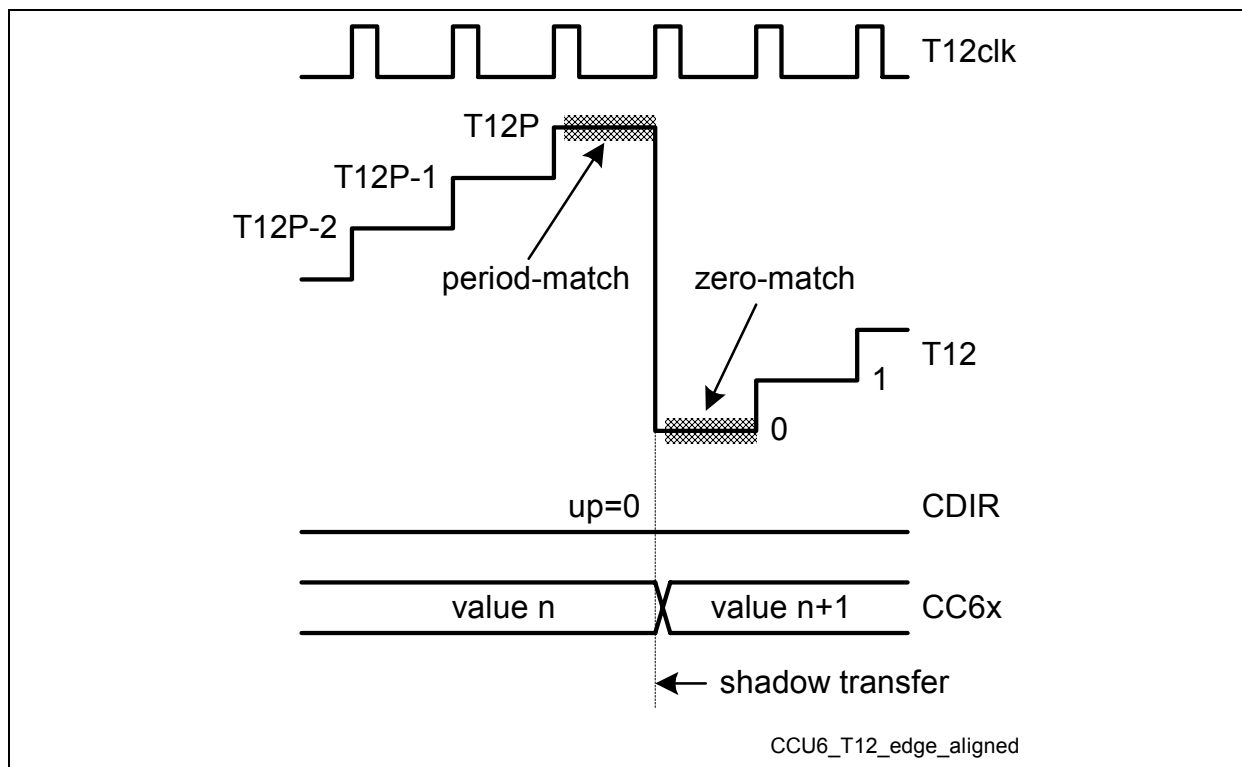


Figure 7-3 T12 in edge-aligned mode

Capture/Compare Unit 6 (CCU6)

In the center-aligned mode (T12 counts up and down), the counting rules lead to the following behavior:

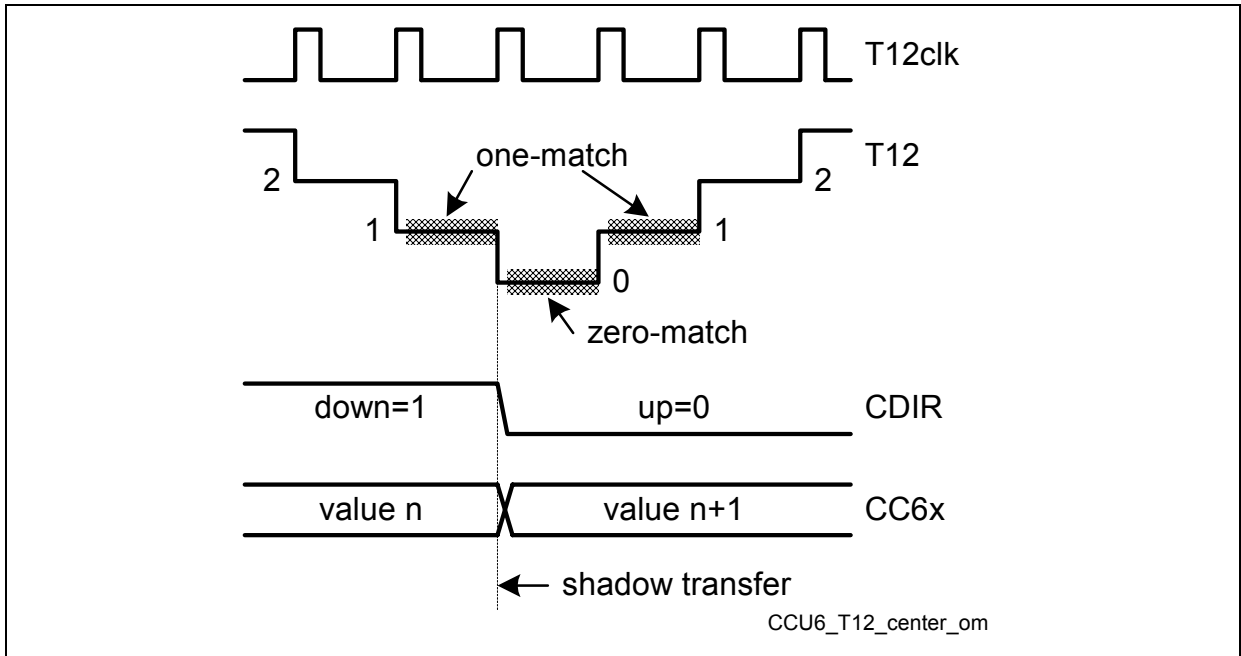


Figure 7-4 T12 in center-aligned mode, one-match detected

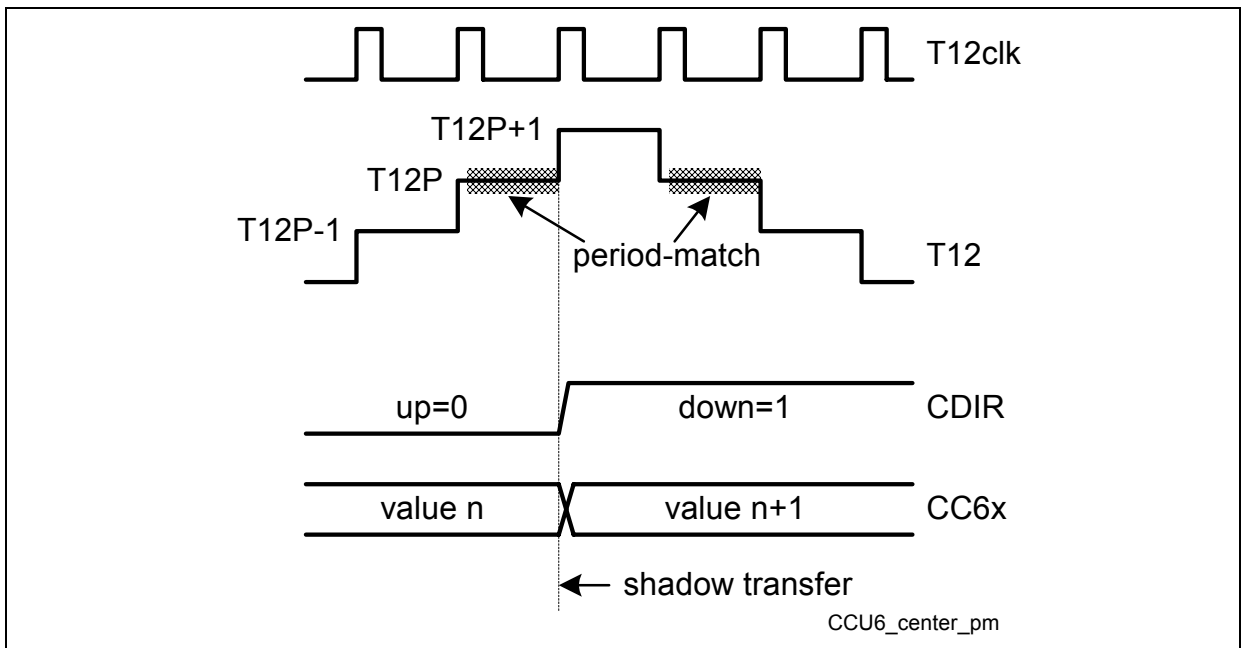


Figure 7-5 T12 in center-aligned mode, period-match detected

7.1.2.3 Switching Rules

The compare actions take place in parallel for the three compare channels. Depending on the count direction, the compare matches have different meanings. In order to get the PWM information independent from the output levels, two different states have been introduced for the compare actions: The active state and the passive state, which are used to generate the desired PWM as a combination of the states delivered by T13, the trap control unit and the multi-channel control unit. If the active state is interpreted as a 1 and the passive state as a 0, the state information is combined with a logical AND function.

- active AND active = active
- active AND passive = passive
- passive AND passive = passive

The compare states change with the detected compare-matches and are indicated by the CC6xST bits. The compare states of T12 are defined as follows:

- passive if the counter value is below the compare value
- active if the counter value is above the compare value

This leads to the following switching rules for the compare states:

- set to the active state when the counter value reaches the compare value while counting up
- reset to the passive state when the counter value reaches the compare value while counting down
- reset to the passive state in case of a zero-match without compare-match while counting up
- set to the active state in case of a zero-match with a parallel compare-match while counting up

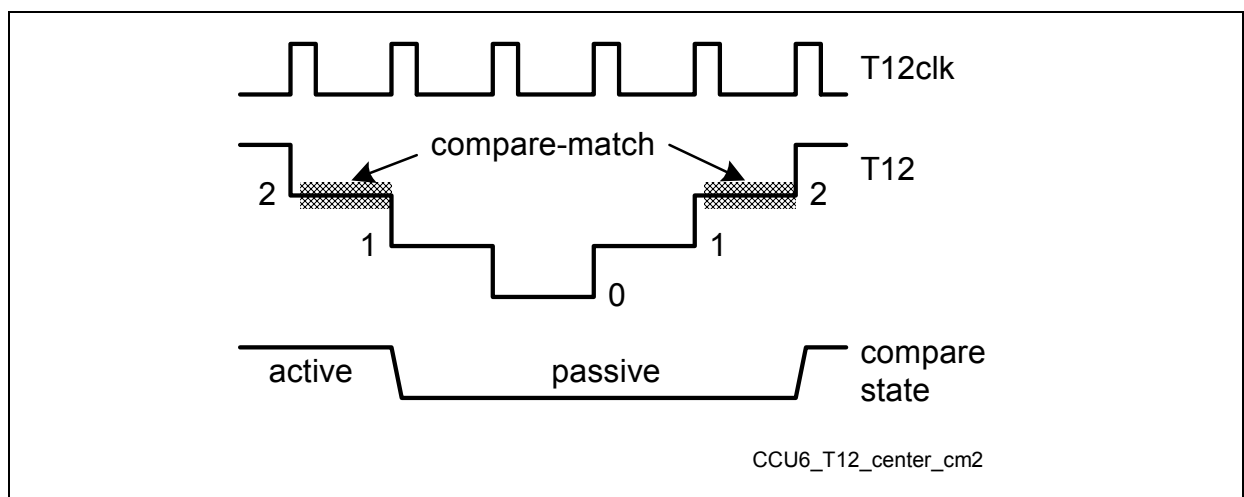


Figure 7-6 Compare States for Compare Value = 2

The switching rules are only taken into account while the timer is running. As a result, write actions to the timer registers while the timer is stopped do not lead to compare actions.

7.1.2.4 Duty Cycle of 0% and 100%

These counting and switching rules ensure a PWM functionality in the full range between 0% and 100% duty cycle (duty cycle = active time / total PWM period). In order to obtain a duty cycle of 0% (compare state never active), a compare value of T12P+1 has to be programmed (for both compare modes). A compare value of 0 will lead to a duty cycle of 100% (compare state always active).

7.1.2.5 External Timer Start

The timer run bit T12R can be also set automatically if an event is detected on the external signal T12HR. The event can be either a rising edge, a falling edge or any edge, see [Figure 7-7](#). If bit field T12RSEL = 00, the external timer start feature is disabled and the timer run bit can be only controlled by software.

The set and reset conditions of the timer run bit T12R is kept unchanged due to T12 functionality.

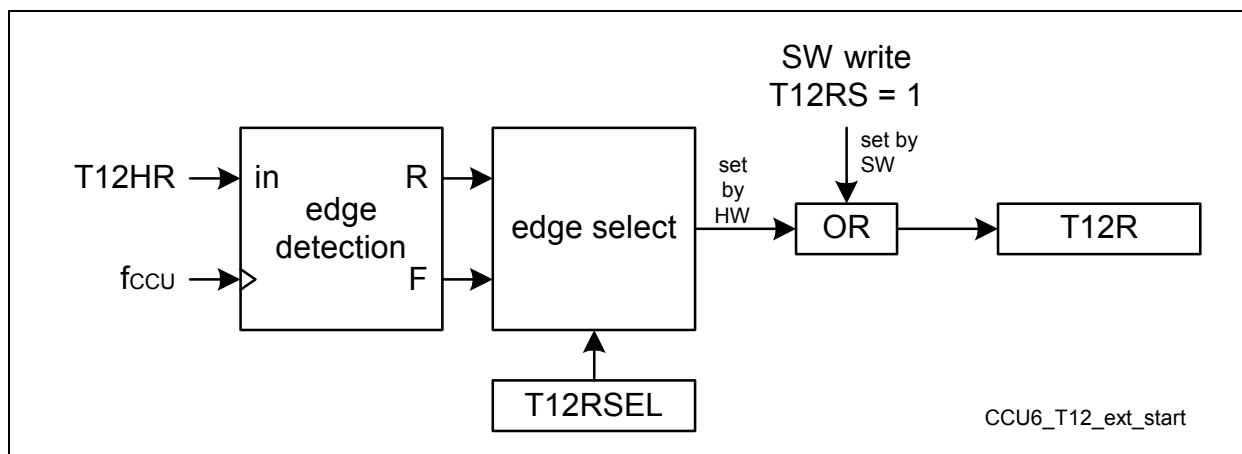


Figure 7-7 External Timer Start

7.1.2.6 Compare Mode of T12

The following figure shows the setting and resetting of the compare state bit CC6xST. In order to simplify the description, only one out of the three parallel channels is described. The letter “x” in the simplified bit names and signal names indicates that there is more than one channel. The CC6xST bit is the compare state bit in register CMPSTAT, the bit CC6xPS represents passive state select bit.

The timer T12 generates pulses indicating events like compare-matches, period-matches and zero-matches, which are used to set (signal T12_xST_se) and to reset (signal T12_xST_re) the corresponding compare state bit (CC6xST) according to the counting direction.

The timer T12 modulation output lines T12xO (two for each channel) can be selected to be in the active state while the corresponding compare state is 0 (with CC6xPS= 0) or while the corresponding compare state is 1 (with CC6xPS= 1). The bit COUT6xPS has the same effect for the second output of the channel. The example is shown without dead-time.

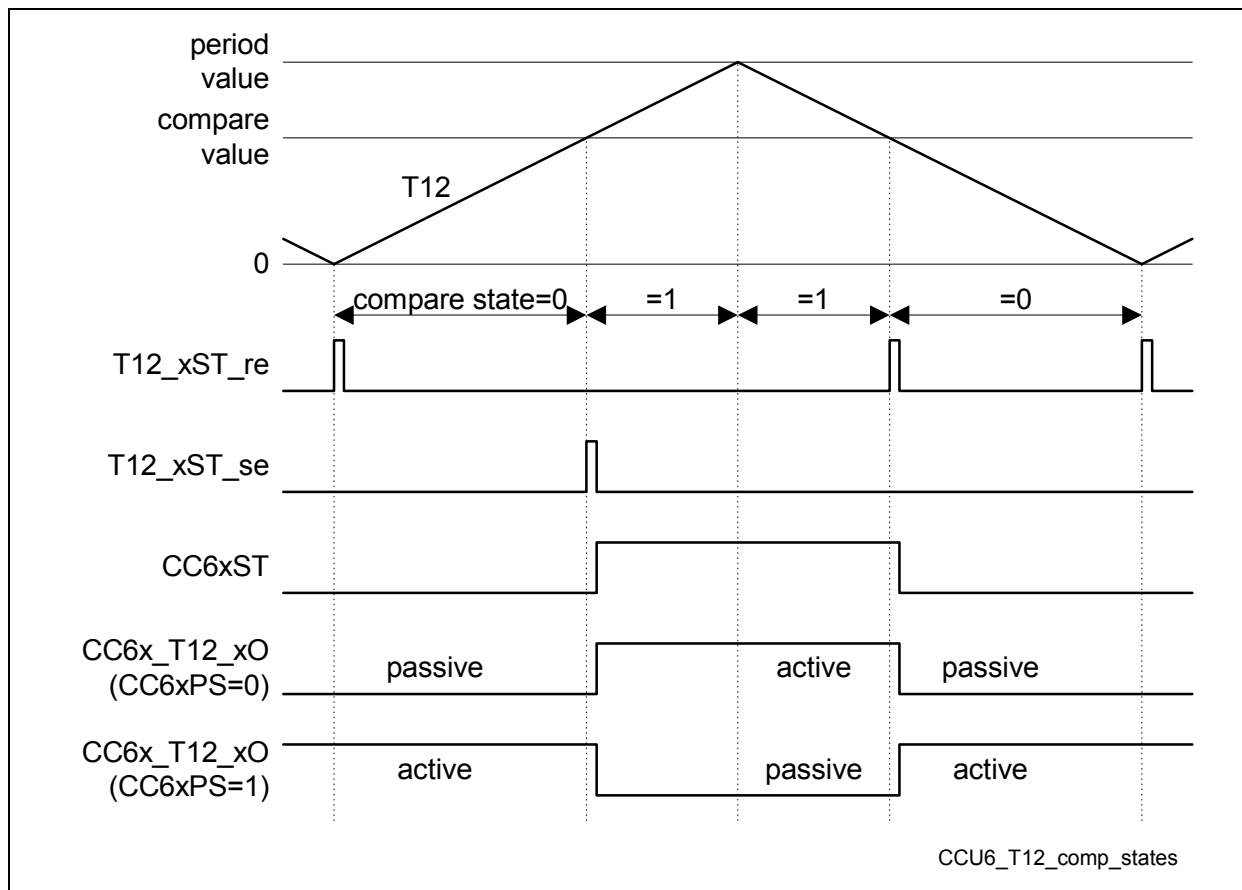


Figure 7-8 Compare States of Timer T12

Capture/Compare Unit 6 (CCU6)

edge (ICC6xR), whereas the signal T12_xST_ro indicates the falling edge event (ICC6xF) in compare mode.

The compare state bits indicate the occurrence of a capture or compare event of the corresponding channel. It can be set (if it is 0) by the following events:

- upon a software set (CC6xS)
- upon a compare set event (see switching rules) if the T12 runs and if the T12 set event is enabled
- upon a capture set event

The bit CC6xST can be reset (if it is 1) by the following events:

- upon a software reset (CC6xR)
- upon a compare reset event (see switching rules) if the T12 runs and if the T12 reset event is enabled (including in single shot mode the end of the T12 period)
- upon a reset event in the hysteresis-like control mode

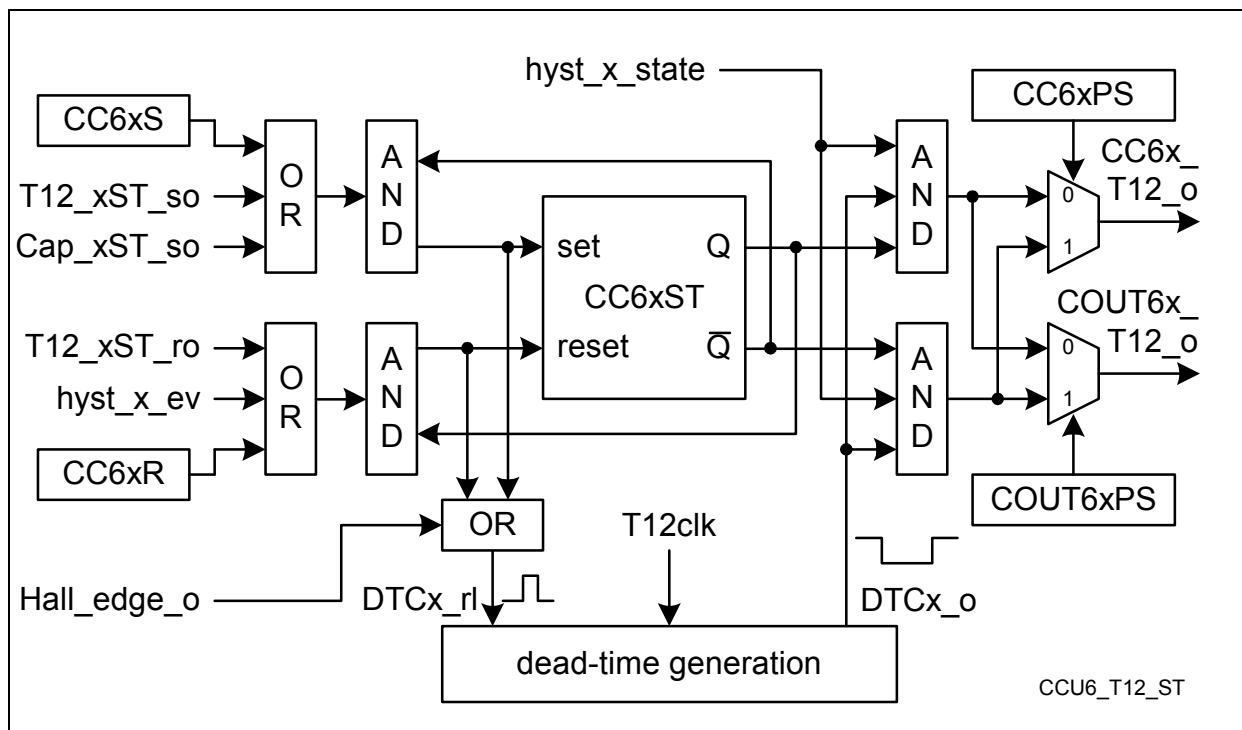


Figure 7-10 T12 Logic for CC6xST Control

The events triggering the set and reset action of the CC6xST bits have to be combined, see [Figure 7-10](#). The occurrence of the selected capture event (signal Cap_xST_so) or the setting of CC6xS in register CMPMODIF also leads to a set action of bit CC6xST, whereas the negative edge at pin CCPOSx (in hysteresis-like mode, signal hyst_x_ev) or the setting of bit CC6xR leads to reset action.

The set signal is only generated while bit CC6xST is reset, a reset can only take place while the bit is set. This permits the OR-combination of the resulting set and reset signals

Capture/Compare Unit 6 (CCU6)

to one common signal (DTCx_rl) triggering the reload of the dead-time counter. It is only triggered if bit CC6xST is changed, permitting a correct PWM generation with dead-time and the complete duty cycle range of 0% to 100% in edge-aligned and in center-aligned mode.

In the case that the dead-time generation is enabled, the change of bit CC6xST triggers the dead-time unit and a signal DTCx_o is generated. The length of the 0 level of this signal corresponds to the desired dead-time, which is used to delay the rising edge (passive to active edge) of the output signal.

In order to generate independent PWM patterns for the highside and the lowside switches of the power inverter, the interval when a PWM signal should be active can be selected by the bits CC6xPS. They select if the PWM signal is active while the compare state bit is 0 (T12 counter value below the compare value) or while it is 1 (T12 counter value above the compare value).

In **Figure 7-10**, the signals CC6x_T12_o and COUT6x_T12_o are inputs to the modulation control block, where they can be combined with other PWM signals.

7.1.2.7 Switching Examples in edge-aligned Mode

The following figure shows two switching examples in edge-aligned mode with duty cycles near to 0% and near to 100%. The compare-, period- or zero-matches lead to modifications of the compare state and the shadow transfer (if requested by STE12 = 1) in the next clock cycle.

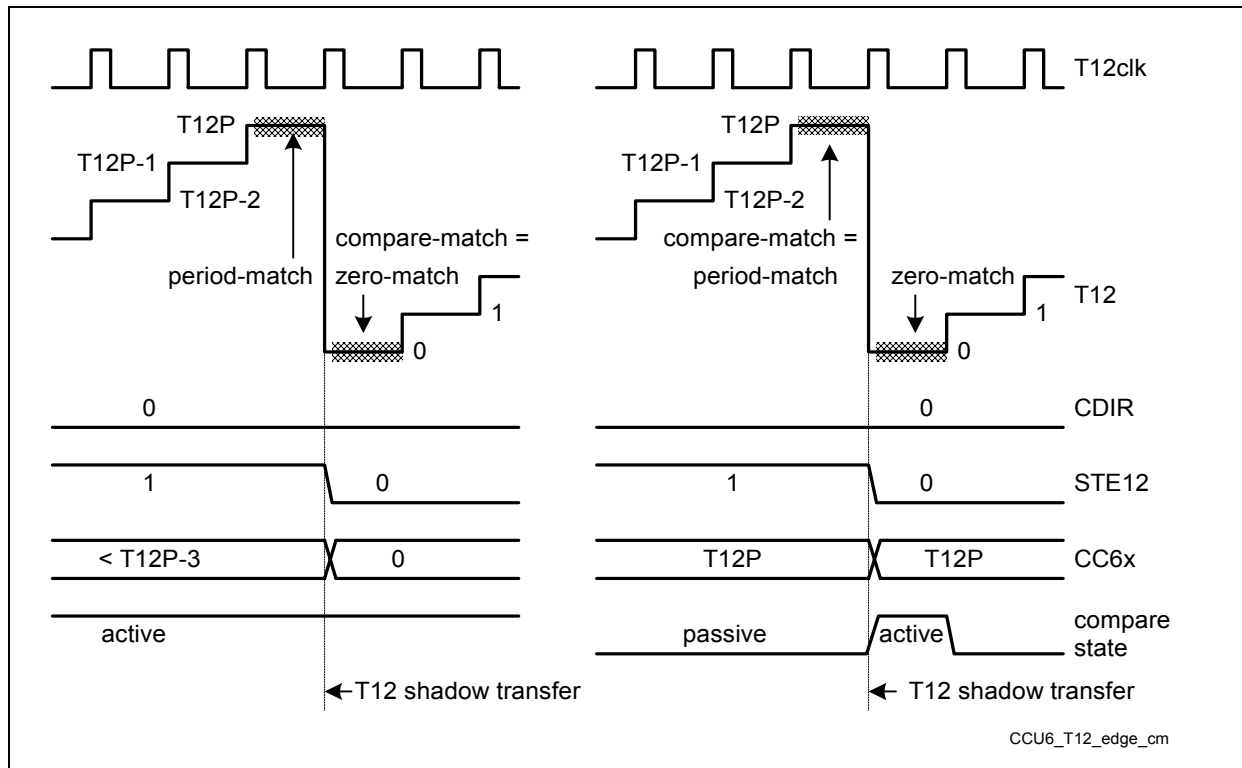


Figure 7-11 Switching Examples in edge-aligned Mode

7.1.2.8 Switching Examples in center-aligned Mode

The following figures show examples of the switching of the compare state and the T12 shadow transfer according to the programmed compare values.

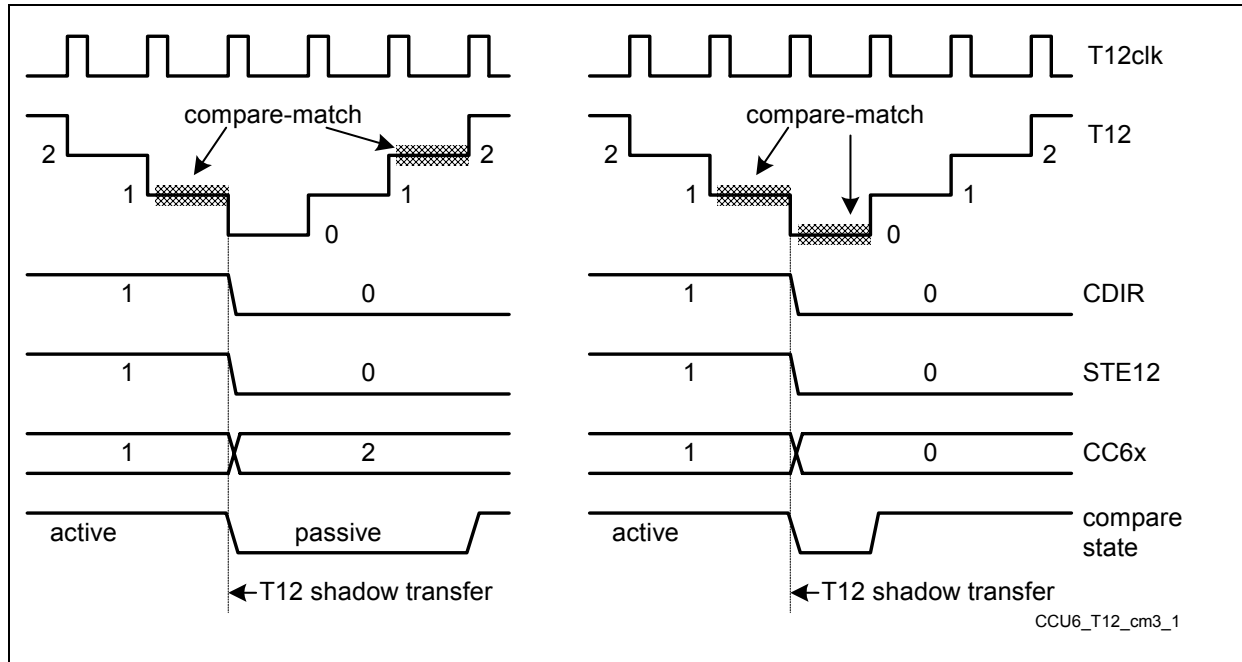


Figure 7-12 Switching Example for Duty Cycles near to 100%

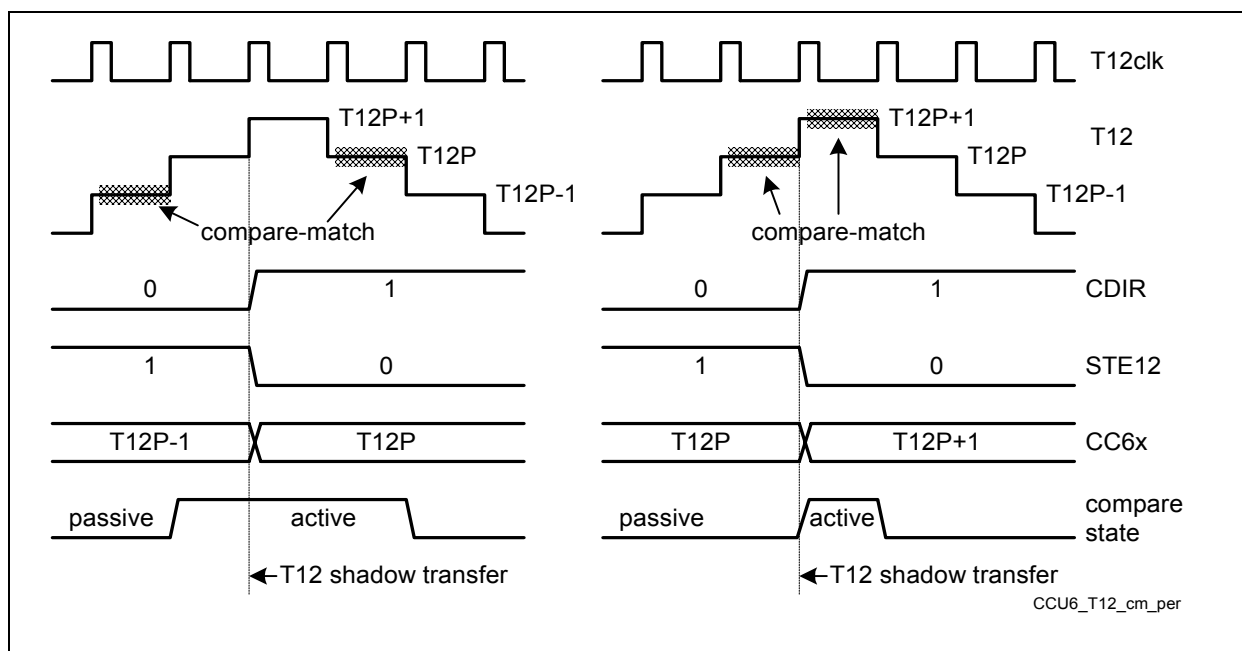


Figure 7-13 Switching Example for Duty Cycles near to 0%

7.1.2.9 Dead-time Generation

The generation of (complementary) signals for the highside and the lowside switches of one power inverter phase is based on the same compare channel. For example, if the highside switch should be active while the T12 counter value is above the compare value (compare state = 1), then the lowside switch should be active while the counter value is below (compare state = 0). The compare state, which may lead to an active output (respecting other modulation sources and the trap functionality) can be selected by the CC6xPS bits.

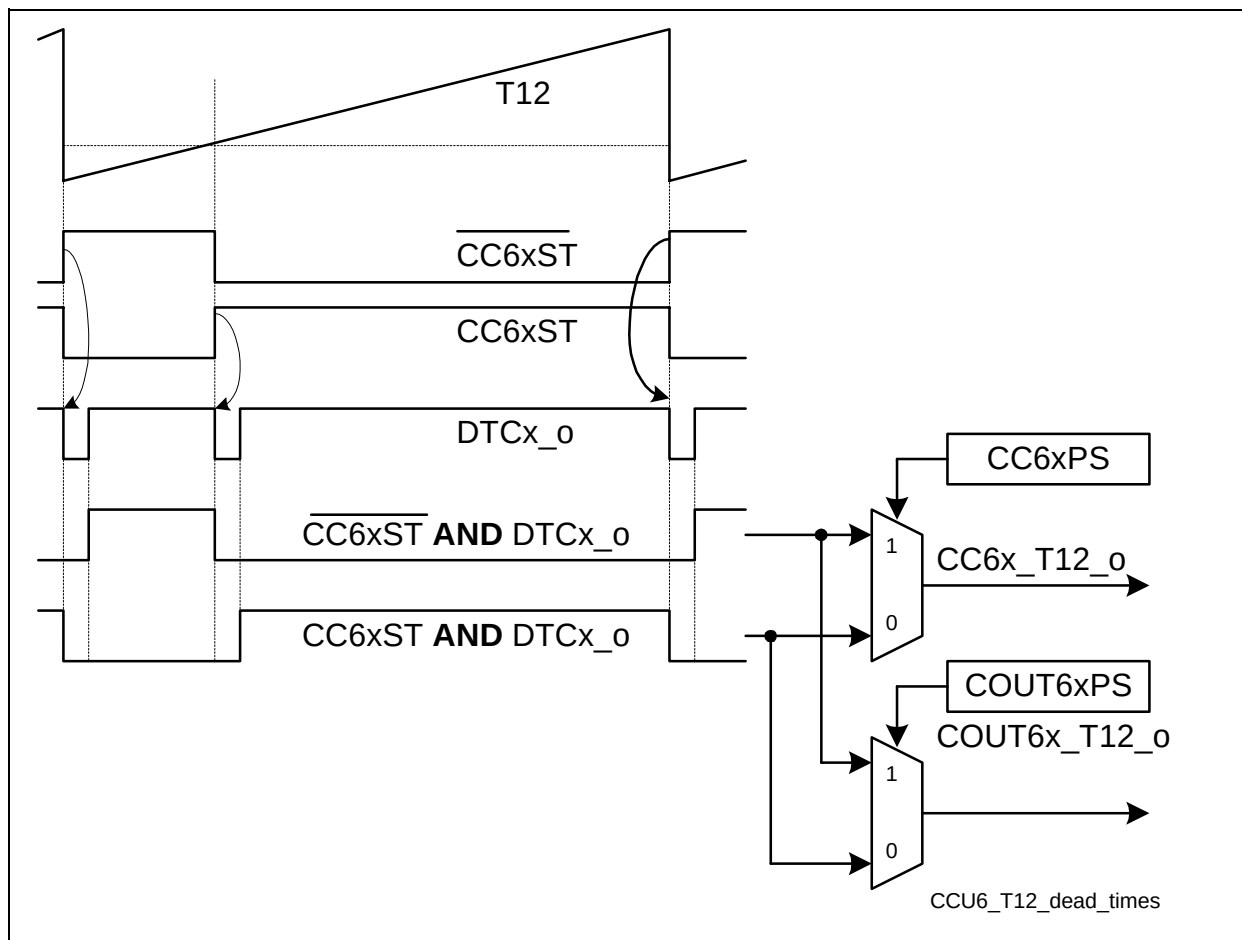


Figure 7-14 PWM-signals with Dead-time Generation

In most cases, the switching behavior of the connected power switches is not symmetrical concerning the times needed to switch on and to switch off. A general problem arises if the time to switch on is smaller than the time to switch off the power device. In this case, a short-circuit in the inverter bridge leg occurs, which may damage the complete system. In order to solve this problem by hardware, this capture/compare unit contains a programmable dead-time counter, which delays the passive to active

edge of the switching signals (the active to passive edge is not delayed), see [Figure 7-14](#).

The dead-time generation logic (see [Figure 7-15](#)) is built in a similar way for all three channels of T12. Each change of the CC6xST bits triggers the corresponding dead-time counter (6 bit down counter, clocked with T12clk). The trigger pulse (DTCx_rl) leads to a reload of the dead-time counter with the value, which has been programmed in bit field DTM. This reload can only take place if the dead-time feature is enabled by bit DTE_x and while the counter is zero.

While counting down (zero is not yet reached), the output line DTCx_o becomes 0. This output line is combined with the T12 modulation signals, leading to a delay of the passive to active edge of the resulting signal, which is shown in [Figure 7-14](#). When reaching the counter value zero, the dead-time counter stops counting and the signal DTCx_o becomes 1. The dead-time counter cannot be reloaded while it is counting.

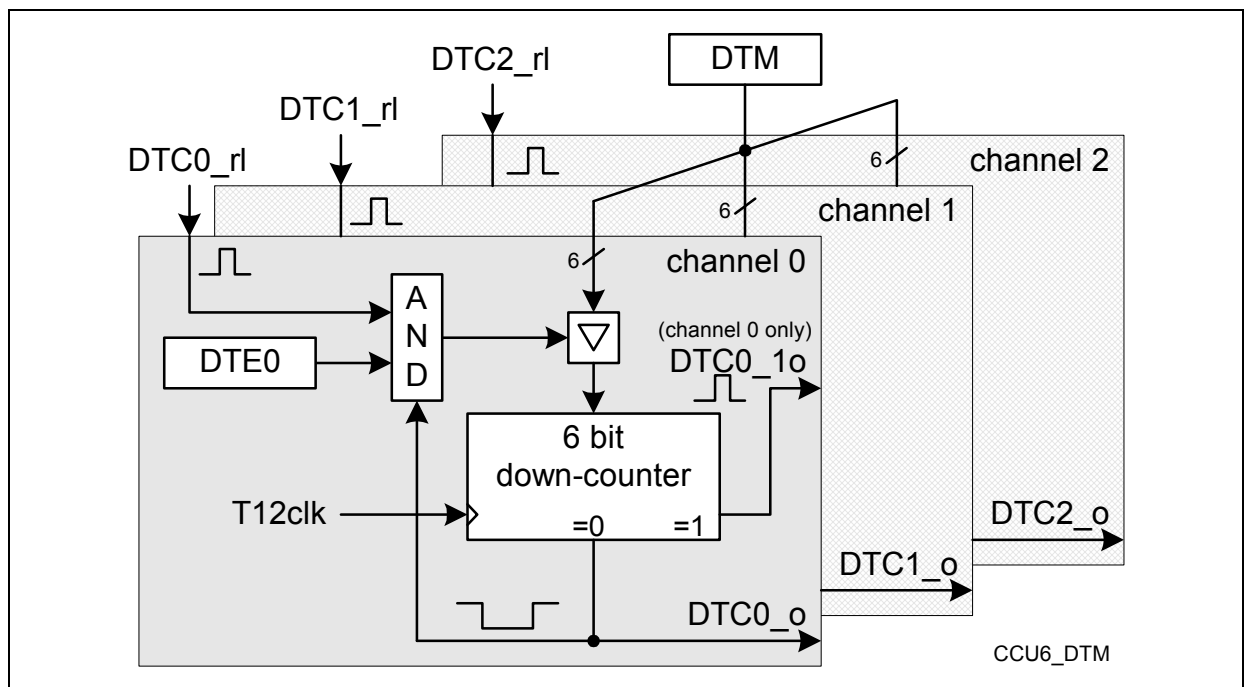


Figure 7-15 Dead-time Counter

Each of the three channels works independently with its own dead-time counter and the trigger and enable signals. The value of bit field DTM is valid for all of the three channels.

In the Hall sensor mode, timer T12 is used to measure the rotational speed of the motor (channel 0 in capture mode) and to control the phase delay before switching to the next state (channel 1 in compare mode). Furthermore, channel 2 can be used to generate a time-out signal (in compare mode). As a result, T12 cannot be used for modulation and, due to the block commutation patterns, a dead-time generation is not required. In order to build an efficient noise filter for the Hall signals, channel 0 of the dead-time unit is triggered (reloaded) with each detected edge of the Hall signals, see signal Hall_edge_o

in **Figure 7-10**. For this feature, channel 0 also generates a pulse if its counter value is one.

7.1.2.10 Capture Mode

In capture mode the bits CC6xST indicate the occurrence of the selected capture event according to the bit fields MSEL6x. A rising and/or a falling edge on the pins CC6x can be selected as capture event that is used to transfer the contents of timer T12 to the CC6xR and CC6xSR registers. In order to work in capture mode, the capture pins have to be configured as inputs.

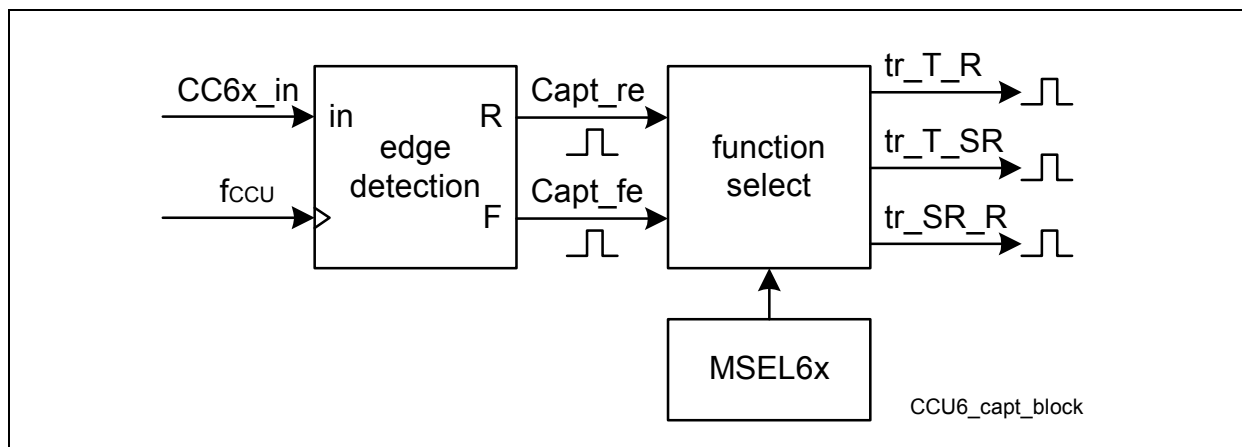


Figure 7-16 Capture Logic

The block diagram of the capture logic for one channel is shown in **Figure 7-16**. This logic is identical for all three independent channels of timer T12. The input signal (CC6x_in) from the input pin CC6x is connected to an edge detection logic delivering two output signals, one for the rising edge (Capt_re) and one for the falling edge (Capt_fe). These signals are also used as trigger sources for the channel interrupts if capture mode is selected.

There are several possibilities to store the captured values in the registers. In double register capture mode the timer value is stored in the channel shadow register CC6xSR. The value formerly stored in this register is simultaneously copied to the channel register CC6xR. This mode can be used if two capture events occur with very little time between them. The software can then check the new captured value and has still the possibility to read the value captured before.

The selection of the capture mode is done by bit field MSEL6x. According to the selected mode and the detected capture event, the signals tr_T_R (transfer T12 contents to register CC6xR), tr_T_SR (transfer T12 contents to register CC6xSR) or tr_SR_R (transfer contents of CC6xSR to register CC6xR) are activated.

Note: In capture mode, a shadow transfer can be requested according to the shadow transfer rules, except for the capture / compare registers, that are left unchanged.

7.1.2.11 Single Shot Mode

In single shot mode, the timer T12 stops automatically at the end of its counting period. **Figure 7-17** shows the functionality at the end of the timer period in edge-aligned and in center-aligned mode. If the end of period event is detected while bit T12SSC is set, the bits T12R and all CC6xST bits are reset.

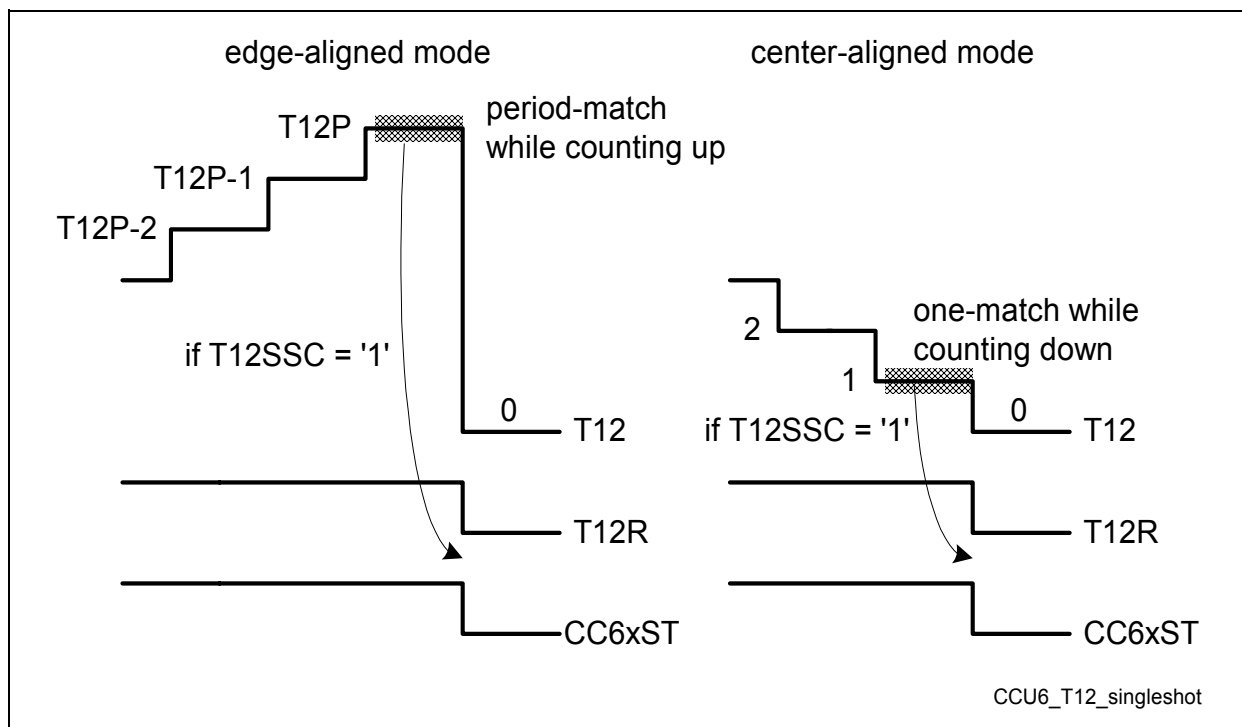


Figure 7-17 End of Single Shot Mode of T12

7.1.2.12 Hysteresis-Like Control Mode

The hysteresis-like control mode ($MSEL6x = 1001$) offers the possibility to switch off the PWM output if the input $CCPOSx$ becomes 0 by resetting bit $CC6xST$. This can be used as a simple motor control feature by using a comparator indicating e.g. over current. While $CCPOSx = 0$, the PWM outputs of the corresponding channel are driving their passive levels. The setting of bit $CC6xST$ is only possible while $CCPOSx = 1$.

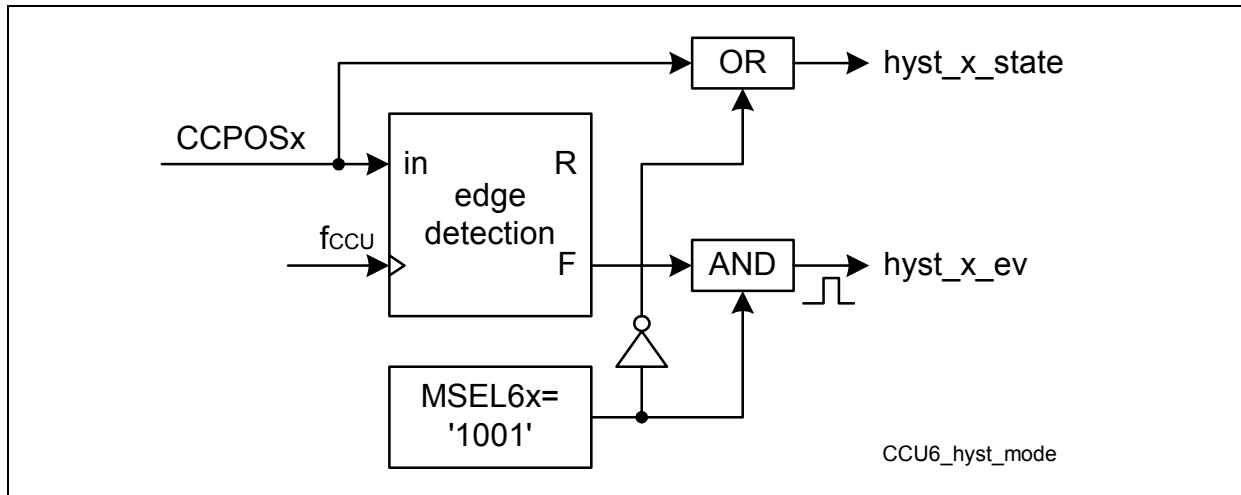


Figure 7-18 Hysteresis-Like Control Mode Logic

This mode can be used to introduce a timing-related behavior to a hysteresis controller. A standard hysteresis controller detects if a value exceeds a limit and switches its output according to the compare result. Depending on the operating conditions, the switching frequency and the duty cycle are not fixed, but change permanently.

If (outer) control loops based on a hysteresis controller (inner loop) should be implemented, the outer loops show a better behavior if they are synchronized to the inner loops. Therefore, the hysteresis-like mode can be used, which combines timer-related switching with a hysteresis controller behavior. For example, in this mode an output can be switched on a fixed time base, but it is switched off as soon as a rising edge is detected at input $CCPOSx$.

7.1.3 Timer T13

7.1.3.1 Overview

The timer T13 is built similar to T12, but only with one channel in compare mode. The counter can only count up (similar to the edge-aligned mode of T12). The T13 shadow transfer in case of a period-match is enabled by bit STE13 in register TCTR0. During the T13 shadow transfer, the contents of register CC63SR is transferred to register CC63R. Both registers can be read by software, whereas only the shadow register can be written by software.

The bits CC63PS, T13IM and PSL63 have shadow bits. The content of the shadow bits is transferred to the actually used bits during the T13 shadow transfer. Write actions target the shadow bits, read actions deliver the value of the actually used bit.

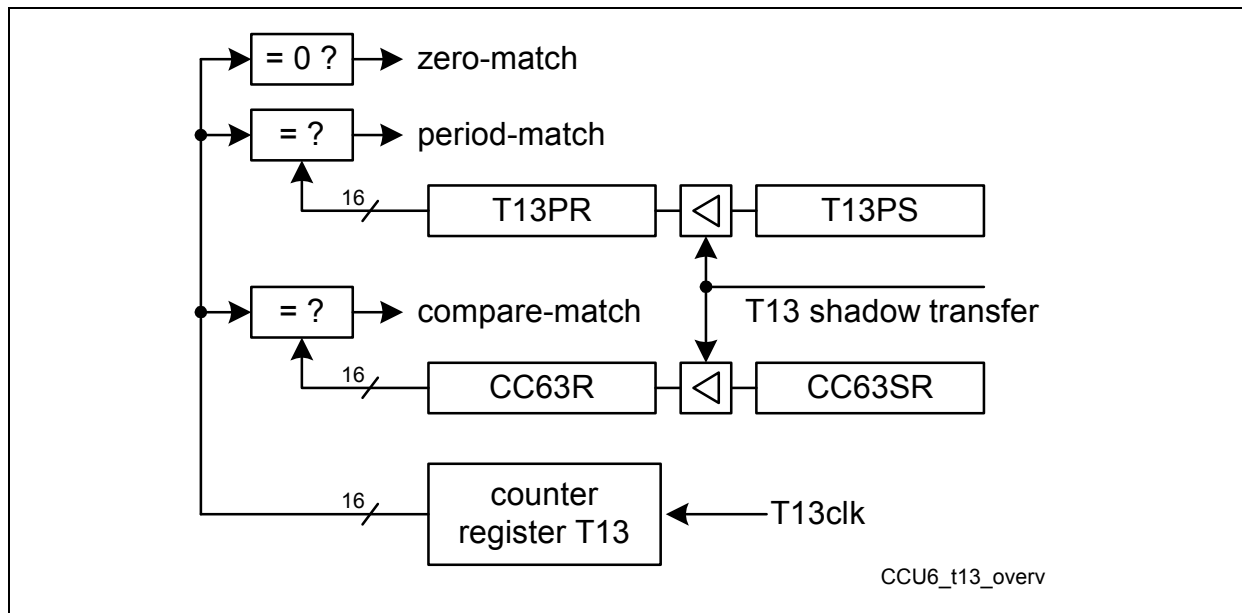


Figure 7-19 T13 Overview

Timer T13 counts according to the same counting and switching rules as timer T12 in edge-aligned mode.

7.1.3.2 Compare Mode

The compare structure of T13 is based on the compare signals T13_ST_se (compare - match detected) and T13_ST_re (zero-match detected without compare-match). These compare signals may modify bit CC63ST only while the timer is running (T13R = 1).

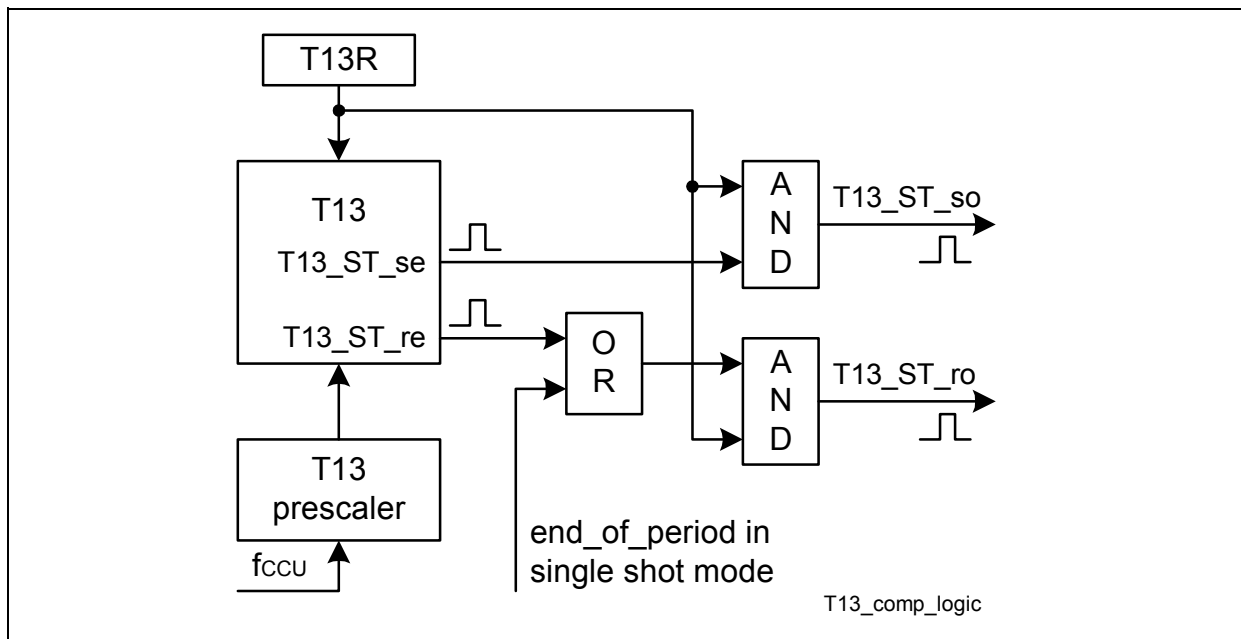


Figure 7-20 T13 Compare Logic

Similar to T12, bit CMPSTAT.CC63ST can be modified by software by CC63SR and CC63R register. The output line COUT63_T13_o can generate a T13 PWM at the output pin COUT63. The signal MOD_T13_o can be used to modulate the other output signals with a T13 PWM. In order to decouple COUT63 from the internal modulation, the compare state leading to an active signal can be selected independently by bits CMPSTAT.T13IM and CMPSTAT.COUT63PS.

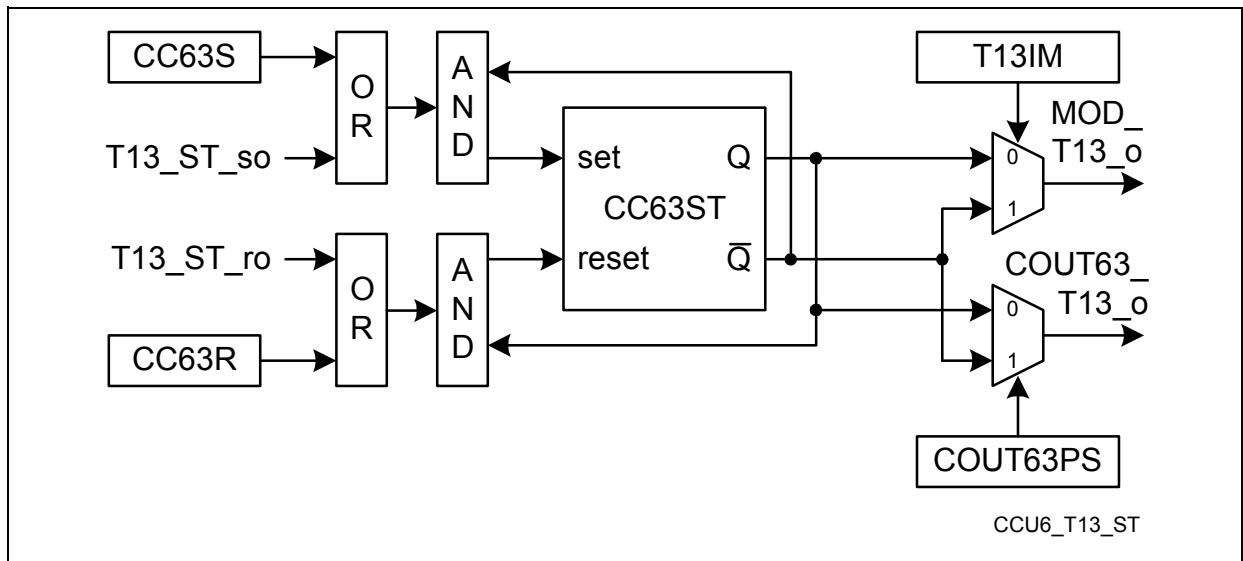


Figure 7-21 T13 Logic for CC6xST Control

7.1.3.3 Single Shot Mode

The single shot mode of T13 is similar to the single shot mode of T12 in edge-aligned mode.

7.1.3.4 External Timer Start

The external timer start feature of T13 is similar to the one of T12.

7.1.3.5 Synchronization of T13 to T12

The timer T13 can be synchronized on a T12 event. The bit fields TCTR2.T13TEC and TCTR2.T13TED select the event, which is used to start timer T13. This event sets bit T13R per hardware and T13 starts counting. Combined with the single shot mode, this feature can be used to generate a programmable delay after a T12 event.

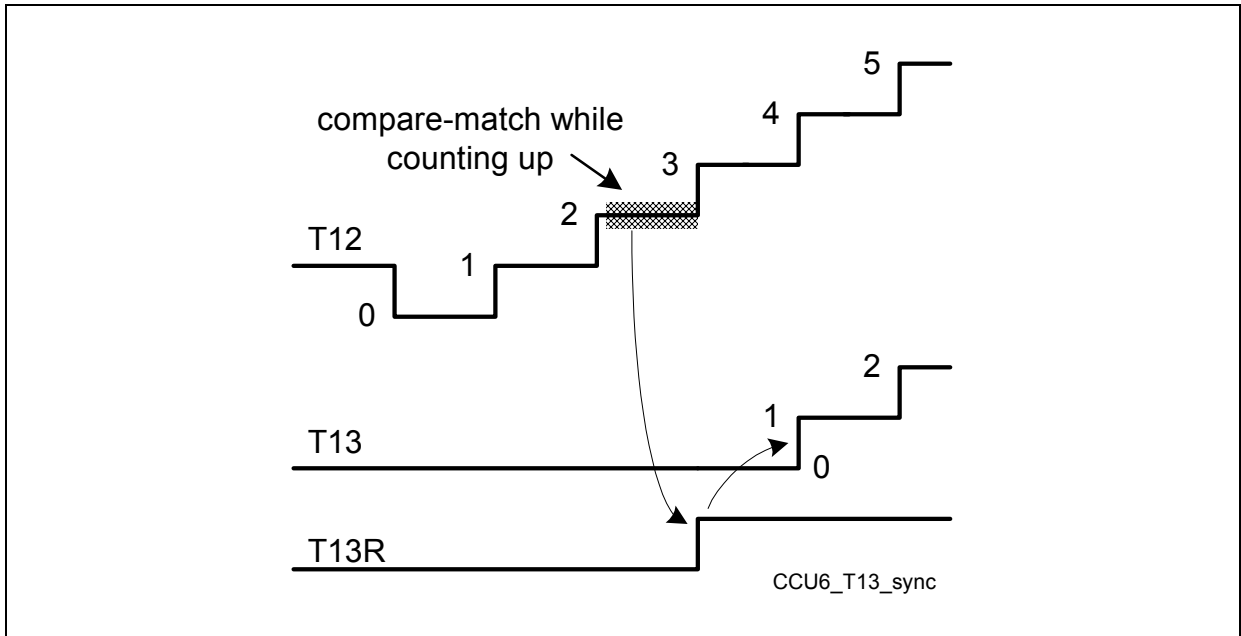


Figure 7-22 Synchronization of T13 to T12

This figure shows the synchronization of T13 to a T12 event. The selected event in this example is a compare-match (compare value = 2) while counting up. The clocks of T12 and T13 can be different (other prescaler factor), but for reasons of simplicity, this example shows the case for T12clk equal to T13clk.

7.1.4 Modulation Control

The modulation control part combines the different modulation sources (CC6x_T12_o, COUT6x_T12_o = six T12-related signals from the three compare channels), the T13-related signal (MOD_T13_o) and the multi-channel modulation signals (MCMP bits). Each modulation source can be individually enabled for each output line. Furthermore, the trap functionality is taken into account to disable the modulation of the corresponding output line during the trap state (if enabled).

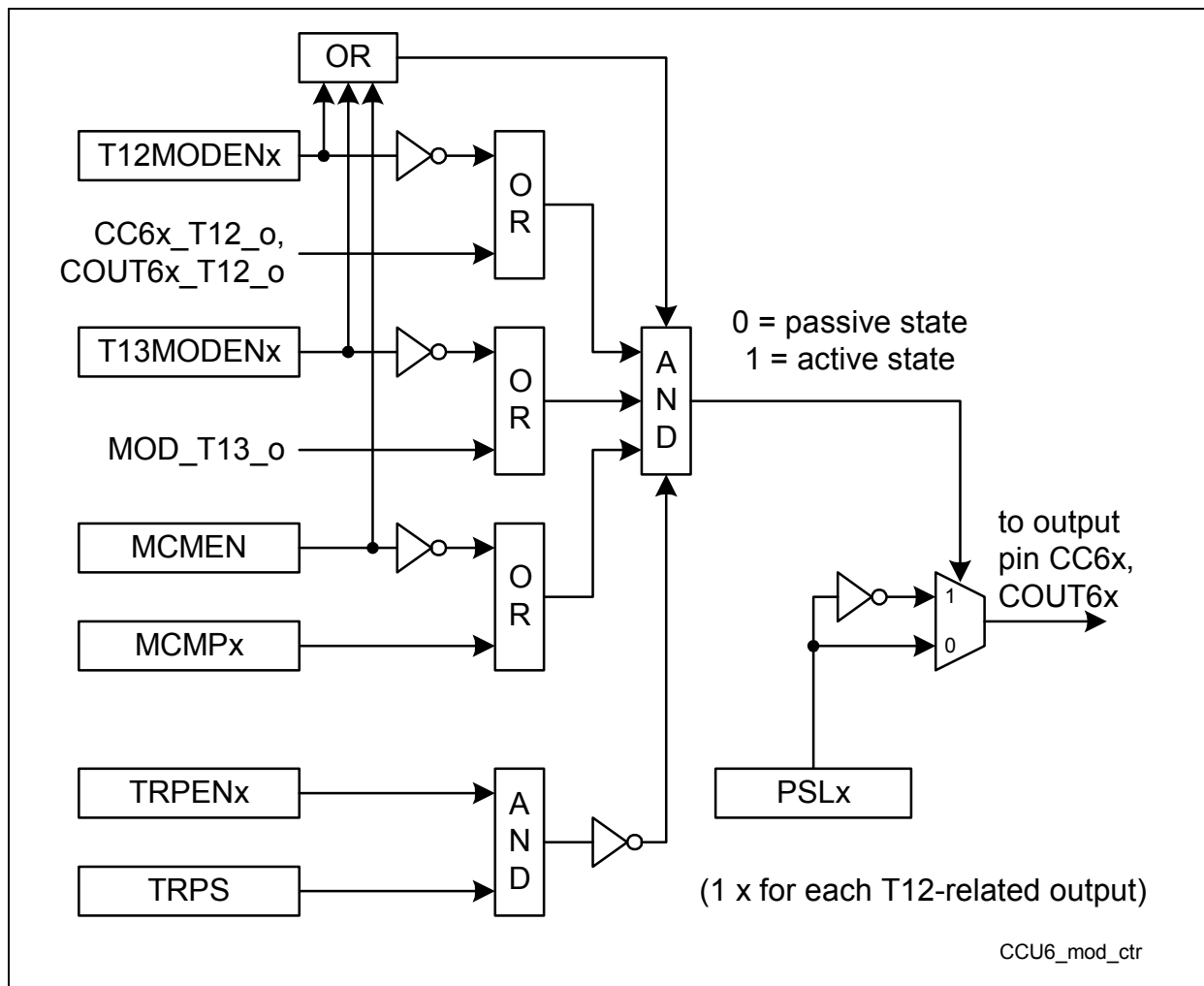


Figure 7-23 Modulation Control of T12-related Outputs

The logic shown in **Figure 7-23** has to be built separately for each of the six T12-related output lines, referring to the index “x” in the figure above.

The output level that is driven while the output is in the passive state is defined by the corresponding bit PSLR.PSL. If the resulting modulation signal is active, the inverted level of the PLSx bit is driven by the output stage.

7.1.5 Trap Handling

The trap functionality permits the PWM outputs to react on the state of the input pin $\overline{\text{CTRAP}}$. This functionality can be used to switch off the power devices if the trap input becomes active (e.g. as emergency stop).

During the trap state, the selected outputs are forced to the passive state and no active modulation is possible. The trap state is entered immediately by hardware if the $\overline{\text{CTRAP}}$ input signal becomes active and the trap function is enabled by bit TRPCTR.TRPPEN. It can also be entered by software by setting bit IS.TRPF (trap input flag), leading to IS.TRPS = 1 (trap state indication flag). The trap state can be left when the input is inactive, by software control and synchronized to the following events:

- TRPF is automatically reset after $\overline{\text{CTRAP}}$ becomes inactive (if TRPCTR.TRPM2 = 0)
- TRPF has to be reset by software after $\overline{\text{CTRAP}}$ becomes inactive (if TRPCTR.TRPM2 = 1)
- synchronized to T12 PWM after TRPF is reset (T12 period-match in edge-aligned mode or one-match while counting down in center-aligned mode)
- synchronized to T13 PWM after TRPF is reset (T13 period-match)
- no synchronization to T12 or T13

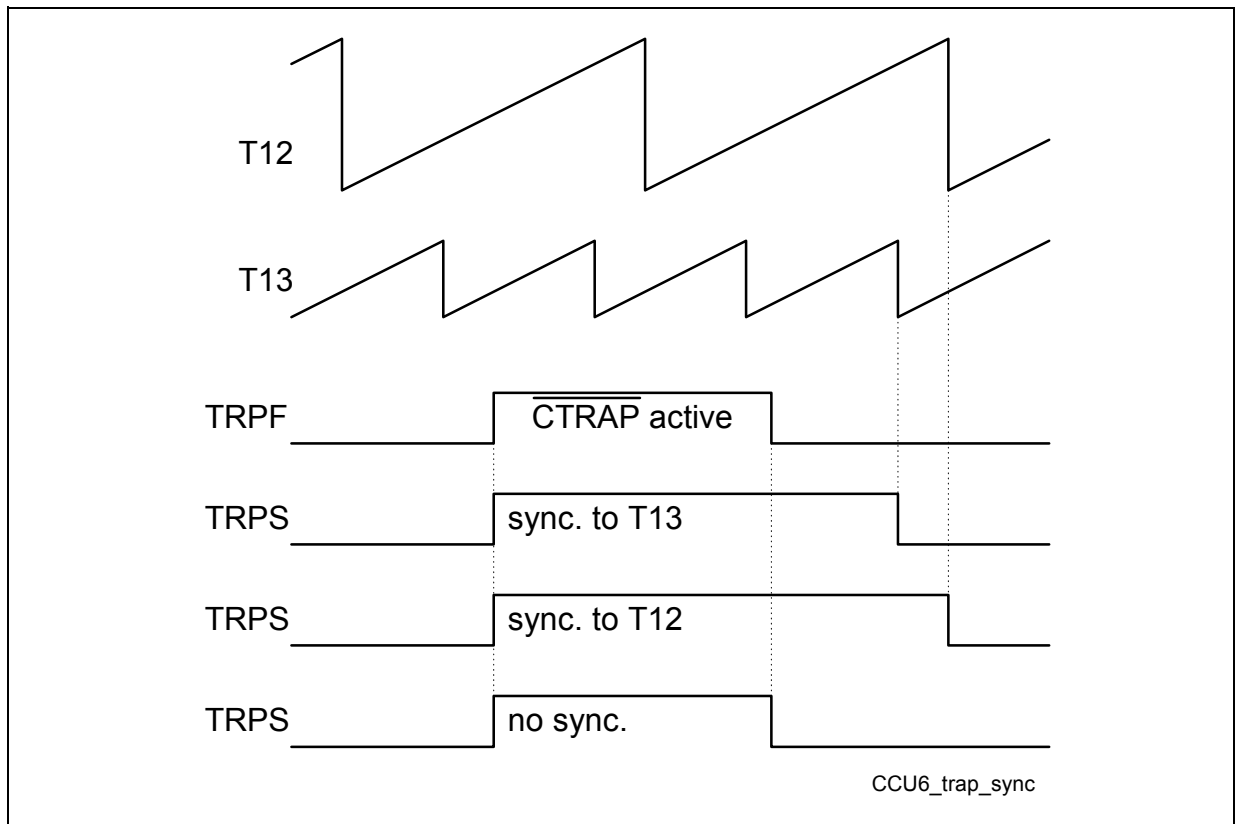


Figure 7-25 Trap State Synchronization (with TRM2 = 0)

7.1.6 Multi-Channel Mode

The multi-channel mode offers a possibility to modulate all six T12-related output signals within one instruction. The bits in bit field MCMP are used to select the outputs that may become active. If the multi-channel mode is enabled (bit MCMEN = 1), only those outputs may become active, which have a 1 at the corresponding bit position in bit field MCMP. This bit field has its own shadow bit field MCMPS, which can be written by software. The transfer of the new value in MCMPS to the bit field MCMP can be triggered by and synchronized to T12 or T13 events. This structure permits the software to write the new value, which is then taken into account by the hardware at a well-defined moment and synchronized to a PWM period. This avoids unintended pulses due to unsynchronized modulation sources (T12, T13, software).

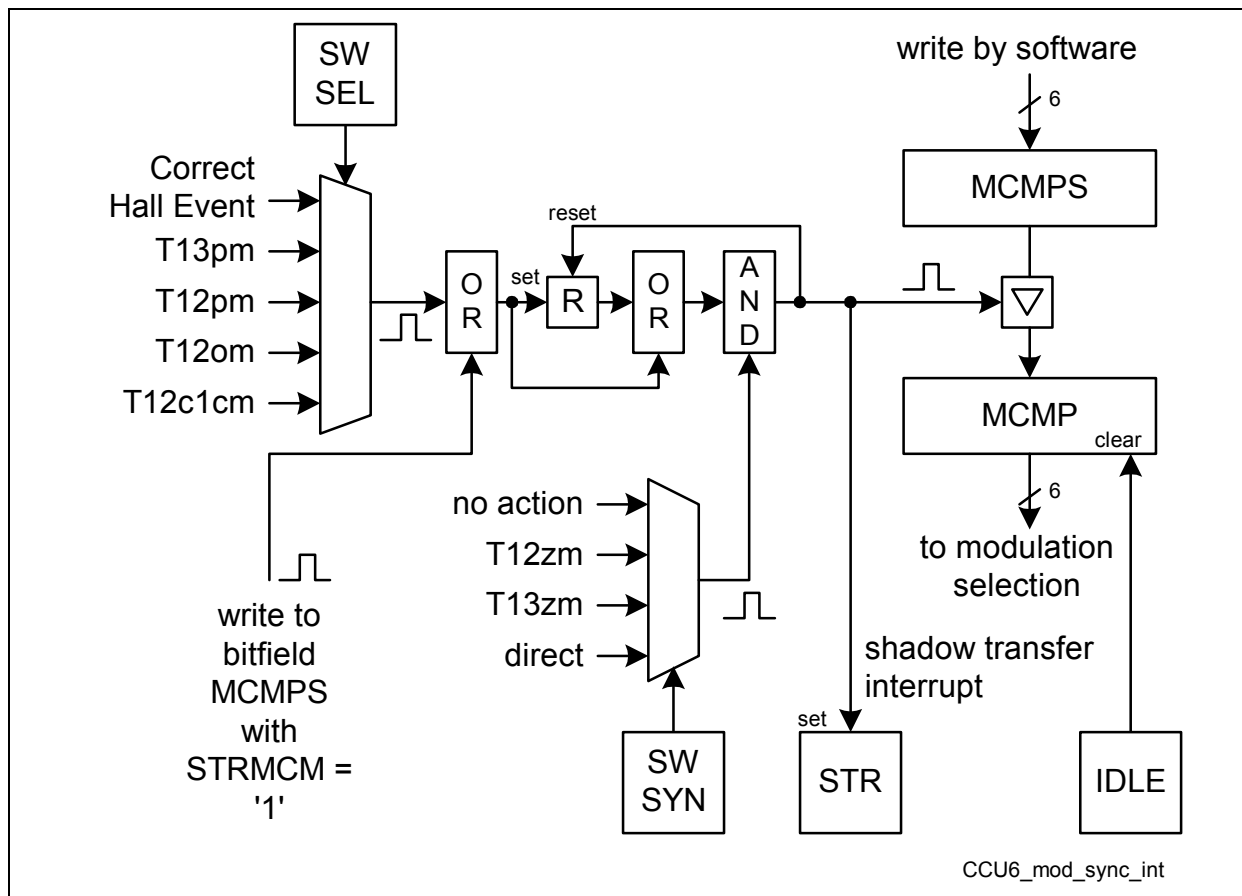


Figure 7-26 Modulation Selection and Synchronization

Figure 7-26 shows the modulation selection for the multi-channel mode. The event that triggers the update of bit field MCMP is chosen by SWSEL. If the selected switching event occurs, the reminder flag R is set. This flag monitors the update request and it is automatically reset when the update takes place. In order to synchronize the update of MCMP to a PWM generated by T12 or T13, bit field SWSYN allows the selection of the

synchronization event, which leads to the transfer from MCMPS to MCMP. Due to this structure, an update takes place with a new PWM period.

If it is explicitly desired, the update takes place immediately with the setting of flag R when the direct synchronization mode is selected. The update can also be requested by software by writing to bit field MCMPS with the shadow transfer request bit STRMCM set. If this bit is set during the write action to the register, the flag R is automatically set. By using the direct mode and bit STRMCM, the update takes place completely under software control.

A shadow transfer interrupt can be generated when the shadow transfer takes place.

The possible hardware request events are:

- a T12 period-match while counting up (T12pm)
- a T12 one-match while counting down (T12om)
- a T13 period-match (T13pm)
- a T12 compare-match of channel 1 (T12c1cm)
- a correct Hall event

The possible hardware synchronization events are:

- a T12 zero-match while counting up (T12zm)
- a T13 zero-match (T13zm)

7.1.7 Hall Sensor Mode

7.1.7.1 Introduction

In **Brushless-DC motors** the next multi-channel state values depend on the pattern of the Hall inputs. There is a strong correlation between the **Hall pattern** (CURH) and the **modulation pattern** (MCMP). Because of different machine types the modulation pattern for driving the motor can be different. Therefore it is wishful to have a wide flexibility in defining the correlation between the Hall pattern and the corresponding modulation pattern. The CCU6 offers this by having a register that contains the actual Hall pattern (CURHS), the next expected Hall pattern (EXPHS) and its output pattern (MCMPS). At every correct Hall event (CHE, see [Figure 7-28](#)) a new Hall pattern with its corresponding output pattern can be loaded (from a predefined table) by software into the register MCMOUTS. Loading this shadow register can also be done by a write action on MCMOUTS with bit STRHP = 1. In case of a phase delay (generated by T12 channel 1) a new pattern can be loaded when the multi-channel mode shadow transfer (indicated by bit STR) occurs.

7.1.7.2 Sampling of the Hall Pattern

The **sampling** of the Hall pattern (on CCPOSx) is with the module clock f_{CCU} . By using the dead-time counter DTC0 (mode MSEL6x = 1000) a hardware **noise filter** can be implemented to suppress spikes on the Hall inputs due to high di/dt in rugged inverter environment. In case of a Hall event the DTC0 is reloaded, starts counting and generates a delay between the detected event and the sampling point. After the counter value of one is reached, the CCPOSx inputs are sampled (without noise and spikes) and are compared to the current Hall pattern (CURH) and to the expected Hall pattern (EXPH). If the sampled pattern equals to the current pattern the edge on CCPOSx was due to a noise spike and no action will be triggered (implicit noise filter by delay). If the sampled pattern equals to the next expected pattern the edge on CCPOSx was a correct Hall event, the bit CHE is set which causes an interrupt.

Additionally, if enabled by all MSEL6x bit fields, timer T12 can be controlled for Hall mode specific actions when a correct Hall event is detected. These specific actions are the capturing of the current content of T12 in register CC60R and a reset of T12 (for new a speed measurement). Furthermore, a shadow transfer takes place for the compare channels CC61 and CC62 (for phase delay generation and time-out criteria).

In the case that the multi-channel mode and the Hall pattern comparison should work independently from timer T12, the delay generation by DTC0 can be bypassed. In this case, timer T12 can be used for other purposes. In order to increase flexibility, the signal to start a Hall pattern comparison (hcrdy) can be selected among several sources, see [Figure 7-27](#).

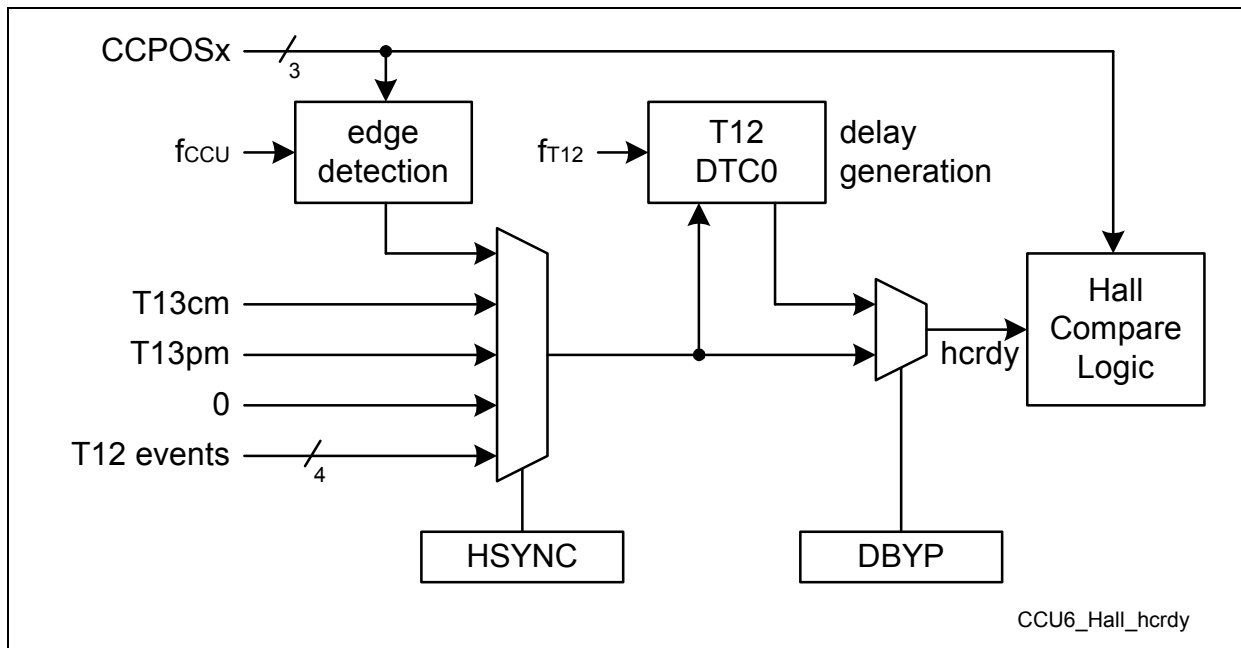


Figure 7-27 Trigger for Hall Compare

7.1.7.3 Hall Events

This correct Hall event can be used as a transfer request event for register MCMOUTS. The transfer from MCMOUTS to MCMOUT transfers the new CURH-pattern as well as the next EXPH-pattern. In case of the sampled Hall inputs were neither the current nor the expected Hall pattern, the bit WHE (wrong Hall event) is set which also can cause an interrupt and sets the IDLE mode clearing MCMP (modulation outputs are inactive). To restart from IDLE the transfer request of MCMOUTS have to be initiated by software (bit STRHP and bit fields SWSEL/SWSYN).

7.1.7.4 Hall Compare Logic

The logic for the Hall compare action (correct Hall event, wrong Hall event) is shown in **Figure 7-28**. It shows the functional dependencies when the Hall compare action is triggered (by signal hcrdy = Hall compare ready).

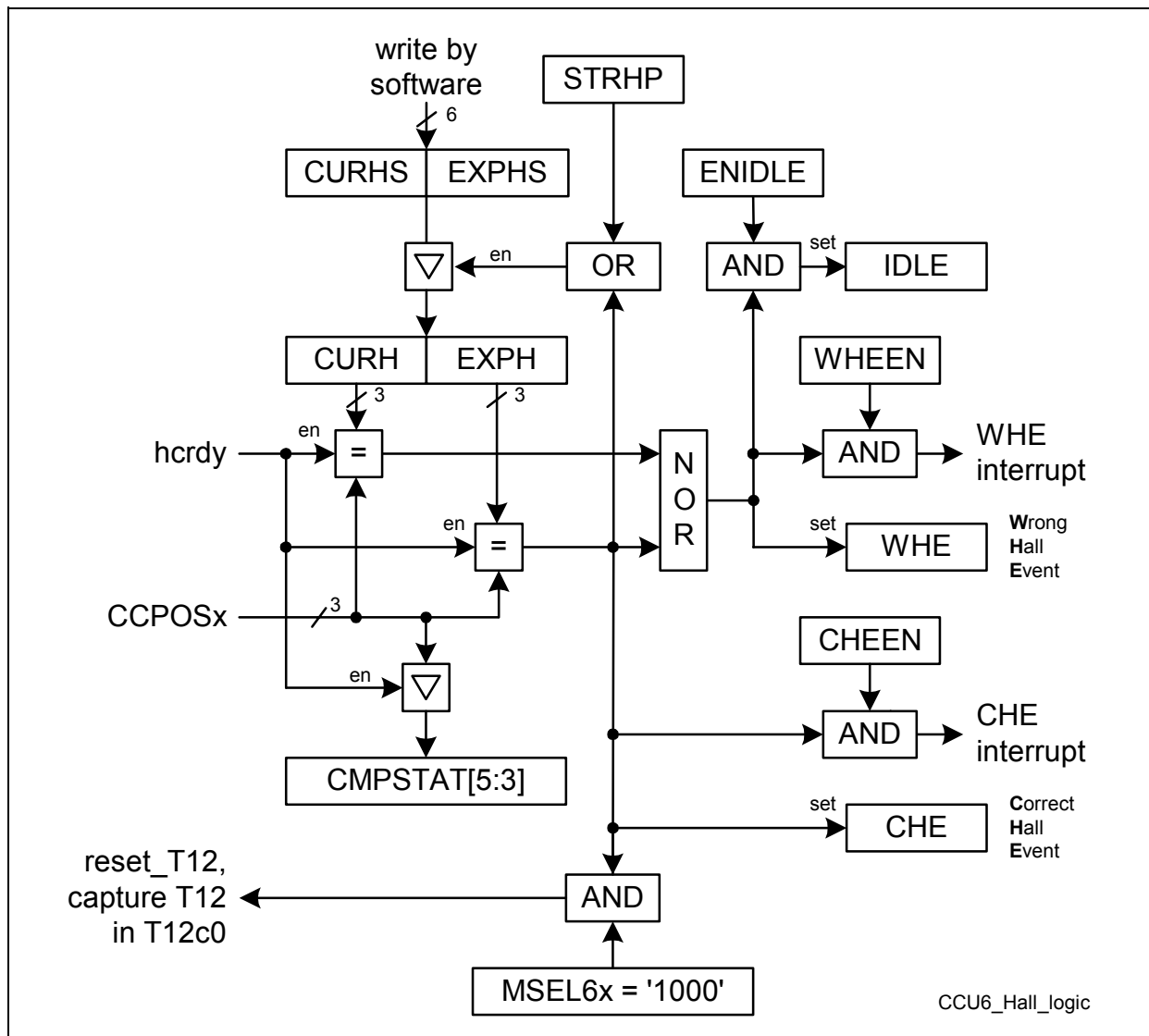


Figure 7-28 Hall Compare Logic

7.1.7.5 Brushless-DC Control

For **Brushless-DC** motors there is a special mode ($MSEL6x = 1000b$) which is triggered by a change of the Hall-inputs (CCPOSx). This mode shows the capabilities of the CCU6 (see [Figure 7-23](#), [Figure 7-26](#), [Figure 7-28](#) and [Figure 7-29](#)). Here, the T12's channel 0 acts in capture function, channel 1 and 2 in compare function (without output modulation) and the multi-channel-block is used to trigger the output switching together with a possible modulation of T13.

After the detection of a valid Hall edge the T12 count value is captured to channel 0 (representing the actual motor speed) and resets the T12. When the timer reaches the compare value in channel 1, the next multi-channel state is switched by triggering the shadow transfer of bit field MCMP (if enabled in bit field SWEN). This trigger event can be combined with several conditions which are necessary to implement a noise filtering (correct Hall event) and to synchronize the next multi-channel state to the modulation sources (avoiding spikes on the output lines). This compare function of channel 1 can be used as a phase delay for the position input to the output switching which is necessary if a sensorless back-EMF technique is used instead of Hall sensors. The compare value in channel 2 can be used as a time-out trigger (interrupt) indicating that the motors destination speed is far below the desired value which can be caused by a abnormal load change. In this mode the modulation of T12 has to be disabled ($T12MODENx = 0$).

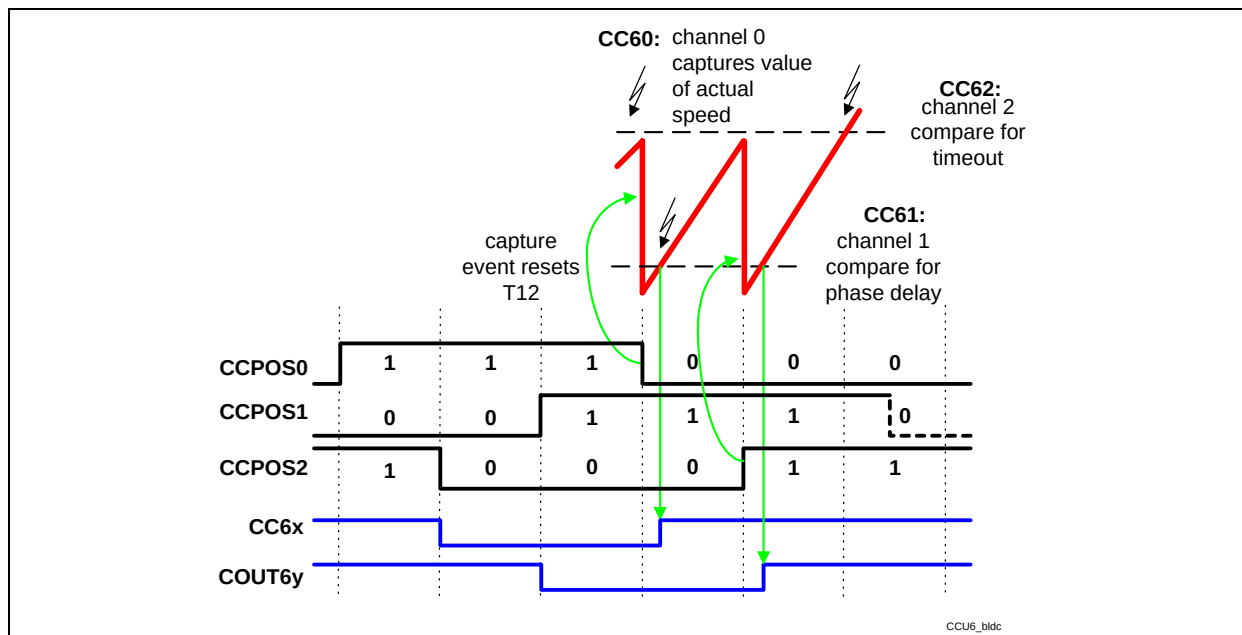


Figure 7-29 Timer T12 Brushless-DC Mode (all $MSEL6x = 1000_H$)

The capturing of the timer value in register CC60R, the shadow transfer from registers CC61SR to CC61R, from CC62SR to CC62R and for the T12 period value is done together with the reset event for T12 (respecting the corresponding STE bit).

7.1.8 Interrupt Generation

The interrupt structure is shown in [Figure 7-30](#). The interrupt event or the corresponding interrupt set bit (in register ISS) can trigger the interrupt generation. The interrupt pulse is generated independently from the interrupt flag in register IS. The interrupt flag can be reset by software by writing to the corresponding bit in register ISR.

If enabled by the related interrupt enable bit in register IEN, an interrupt pulse can be generated at one of the four interrupt output lines of the module.

If more than one interrupt source is connected to the same interrupt node pointer (in register INP), the requests are combined to one common line.

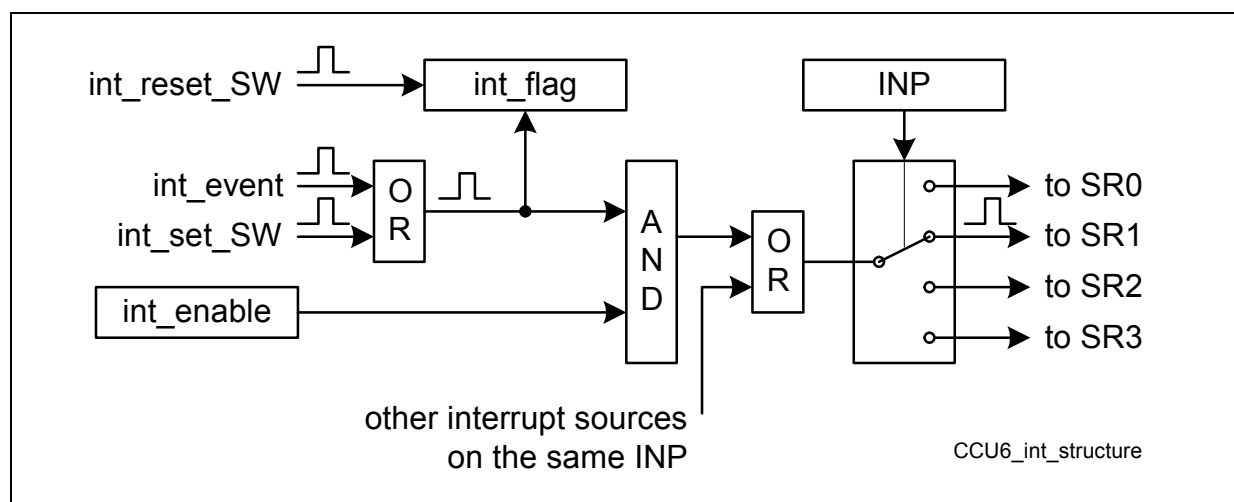


Figure 7-30 Interrupt Generation

The interrupt sources of the CCU6 module can be mapped to four interrupt output lines by programming the interrupt node pointer register INP. The default assignment of the interrupt sources to the output lines and their corresponding control registers is listed in the following table:

Table 7-1 CCU6 Default Interrupt Node Assignment

Source of Interrupt	Interrupt Output Line	Service Request Control Register
Channel 0 Interrupts	SRC0	CCU6_SRC0
Channel 1 Interrupts	SRC0	CCU6_SRC0
Channel 2 Interrupts	SRC0	CCU6_SRC0
Correct Hall Pattern Interrupts	SRC1	CCU6_SRC1
Emergency Interrupts	SRC1	CCU6_SRC1
Timer T12 Interrupts	SRC2	CCU6_SRC2
Timer T13 Interrupts	SRC3	CCU6_SRC3

7.1.9 Suspend Mode

In suspend mode, the functional clock f_{CCU} of the module kernel is stopped. The registers can still be accessed by the CPU (read and write). This mode is useful for debugging purposes, e.g. where the current device status should be 'frozen' in order to get a snapshot of the internal values.

The suspend mode can be entered when the suspend mode is requested, the suspend mode is enabled and the module has reached a safe, deterministic state (equal to the timer stop conditions in single shot mode). This behavior avoids critical situations if a power inverter is connected to the module's outputs.

The suspend mode is non-intrusive concerning the register bits. Register bits must not be changed by hardware when entering or leaving the suspend mode. In suspend mode, all registers can be accessed by write or read instructions for debugging purposes.

In suspend mode, the timers T12 and T13 are not running. Other module functions are still available. The suspend request can lead to a behavior of the output signals equivalent to the trap case.

7.2 CCU6 Kernel Registers

Figure 7-31 and Table 7-2 show all registers associated with the CCU6 Kernel.

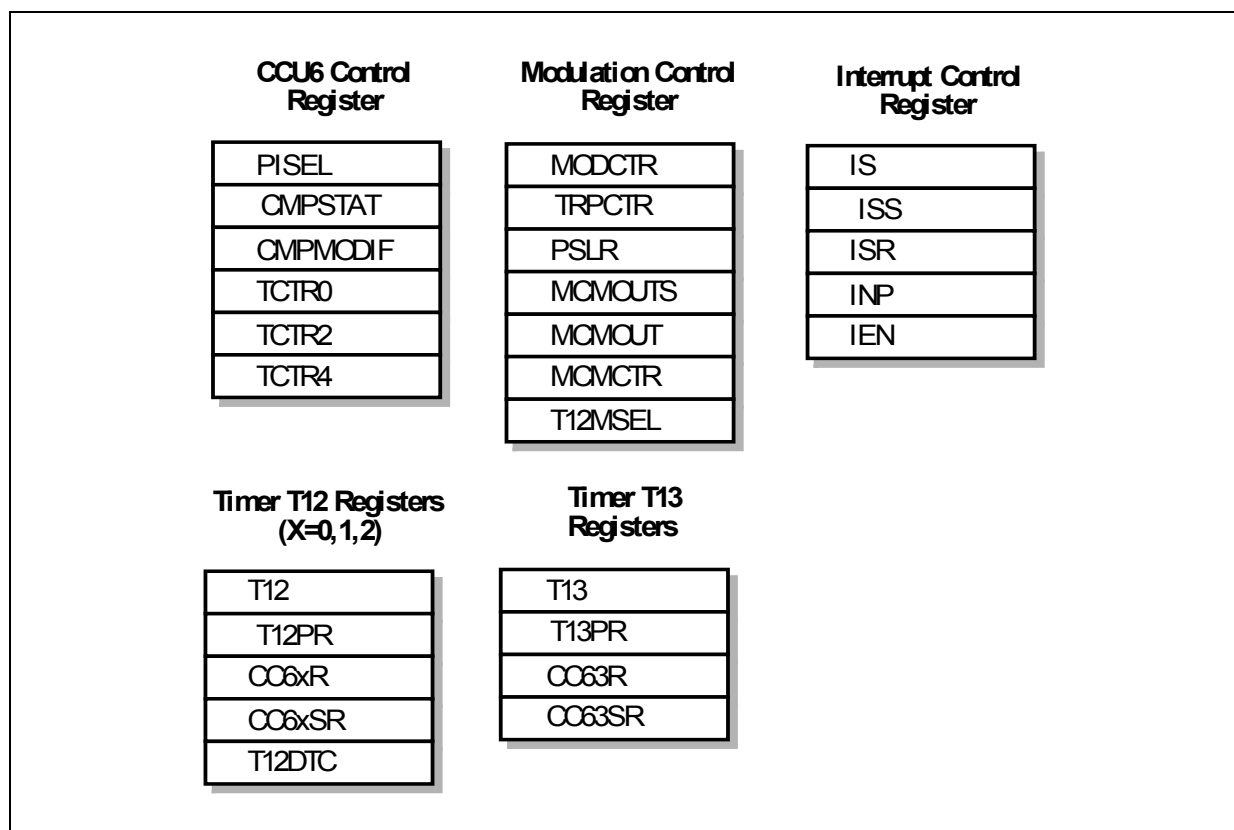


Figure 7-31 CCU6 Register Overview

Note: If a hardware and a software request to modify a bit occur simultaneously, the software wins.

Table 7-2 CCU Kernel Registers

Register Short Name	Register Long Name	Offset Address	Description see
PISEL0	Port Input Select Register 0	0010 _H	Page 7-38
PISEL2	Port Input Select Register 2	0014 _H	Page 7-38
T12	Timer T12 Counter Register	0020 _H	Page 7-51
T12PR	Timer T12 Period Register	0024 _H	Page 7-52
T12DTC	Timer T12 Dead-Time Control Register	0028 _H	Page 7-55
CC60R	Capture/Compare Register T12 Channel 0	0030 _H	Page 7-53

Capture/Compare Unit 6 (CCU6)

Table 7-2 CCU Kernel Registers (cont'd)

Register Short Name	Register Long Name	Offset Address	Description see
CC61R	Capture/Compare Register T12 Channel 1	0034 _H	Page 7-53
CC62R	Capture/Compare Register T12 Channel 2	0038 _H	Page 7-53
CC60SR	Compare Shadow Register T12 Channel 0	0040 _H	Page 7-54
CC61SR	Compare Shadow Register T12 Channel 1	0044 _H	Page 7-54
CC62SR	Compare Shadow Register T12 Channel 2	0048 _H	Page 7-54
T13	Timer T13 Counter Register	0050 _H	Page 7-57
T13PR	Timer T13 Period Register	0054 _H	Page 7-58
CC63R	Compare Register T13	0058 _H	Page 7-59
CC63SR	Compare Shadow Register T13	005C _H	Page 7-60
CMPSTAT	Compare State Register	0060 _H	Page 7-40
CMPMODIF	Compare State Modification Register	0064 _H	Page 7-42
TCTR0	Timer Control Register 0	0068 _H	Page 7-43
TCTR2	Timer Control Register 2	006C _H	Page 7-46
TCTR4	Timer Control Register 4	003C _H	Page 7-49
MODCTR	Modulation Control Register	0070 _H	Page 7-61
TRPCTR	Trap Control Register	0074 _H	Page 7-63
PSLR	Passive State Level Register	0078 _H	Page 7-65
T12MSEL	T12 Mode Select Register	007C _H	Page 7-73
MCMOUTS	Multi_Channel Mode Output Shadow Register	0080 _H	Page 7-67
MCMOUT	Multi_Channel Mode Output Register	0084 _H	Page 7-68
MCMCTR	Multi_Channel Mode Control Register	0088 _H	Page 7-71
IS	Interrupt Status Register	0090 _H	Page 7-77
ISS	Interrupt Status Set Register	0094 _H	Page 7-80
ISR	Interrupt Status Reset Register	0098 _H	Page 7-82
IEN	Interrupt Enable Register	009C _H	Page 7-84
INP	Interrupt Node Pointer Register	00A0 _H	Page 7-88

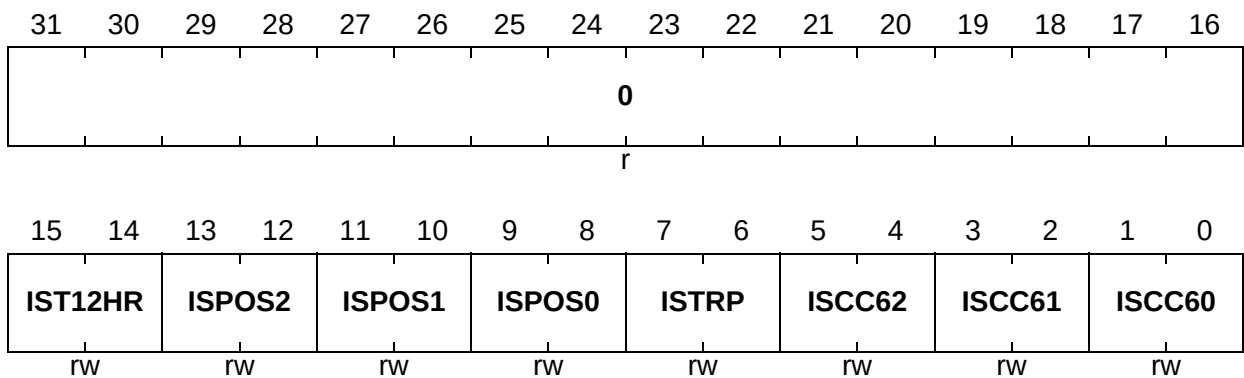
7.2.1 CCU Control Registers

Registers PISEL0 and PISEL2 contain bit fields selecting the actual input signal for the module inputs. This permits to adapt the pin functionality of the device to the application's requirements. The output pins are chosen according to the registers in the ports.

PISEL0

Port Input Select Register 0

Reset Value: 0000 0000_H



Field	Bits	Type	Description
ISCC60	[1:0]	rw	Input Select for CC60 This bit field defines the port pin that is used for the CC60 capture input signal. 00 The input pin for CC60_I0. 01 The input pin for CC60_I1. 10 The input pin for CC60_I2. 11 The input pin for CC60_I3.
ISCC61	[3:2]	rw	Input Select for CC61 This bit field defines the port pin that is used for the CC61 capture input signal. 00 The input pin for CC61_I0. 01 The input pin for CC61_I1. 10 The input pin for CC61_I2. 11 The input pin for CC61_I3.
ISCC62	[5:4]	rw	Input Select for CC62 This bit field defines the port pin that is used for the CC62 capture input signal. 00 The input pin for CC62_I0. 01 The input pin for CC62_I1. 10 The input pin for CC62_I2. 11 The input pin for CC62_I3.

Capture/Compare Unit 6 (CCU6)

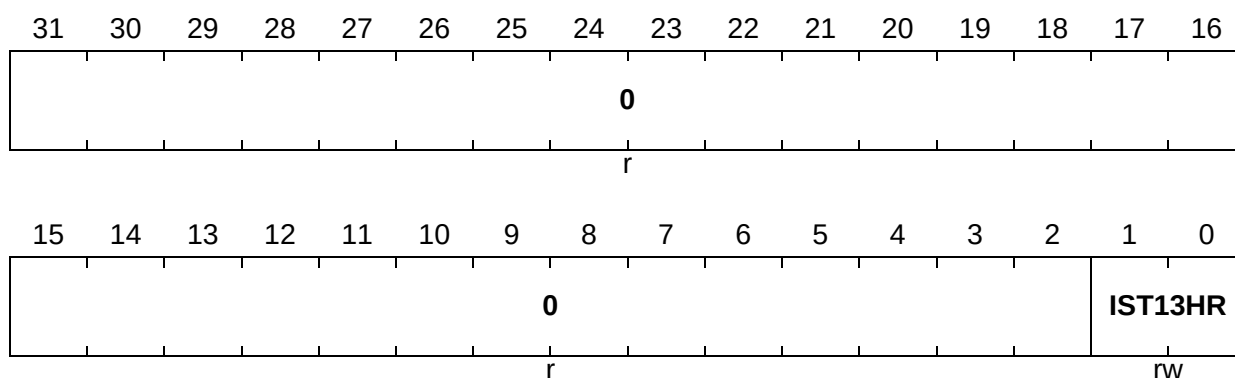
Field	Bits	Type	Description
ISTRP	[7:6]	rw	Input Select for CTRAP This bit field defines the port pin that is used for the CTRAP input signal. 00 The input pin for CTRAP_I0. 01 The input pin for CTRAP_I1. 10 The input pin for CTRAP_I2. 11 The input pin for CTRAP_I3.
ISPOS0	[9:8]	rw	Input Select for CCPOS0 This bit field defines the port pin that is used for the CCPOS0 input signal. 00 The input pin for CCPOS0_I0. 01 The input pin for CCPOS0_I1. 10 The input pin for CCPOS0_I2. 11 The input pin for CCPOS0_I3.
ISPOS1	[11:10]	rw	Input Select for CCPOS1 This bit field defines the port pin that is used for the CCPOS1 input signal. 00 The input pin for CCPOS1_I0. 01 The input pin for CCPOS1_I1. 10 The input pin for CCPOS1_I2. 11 The input pin for CCPOS1_I3.
ISPOS2	[13:12]	rw	Input Select for CCPOS2 This bit field defines the port pin that is used for the CCPOS2 input signal. 00 The input pin for CCPOS2_I0. 01 The input pin for CCPOS2_I1. 10 The input pin for CCPOS2_I2. 11 The input pin for CCPOS2_I3.
IST12HR	[15:14]	rw	Input Select for T12HR This bit field defines the port pin that is used for the T12HR input signal. 00 The input pin for T12HR_I0. 01 The input pin for T12HR_I1. 10 The input pin for T12HR_I2. 11 The input pin for T12HR_I3.
0	[31:16]	r	Reserved ; read as 0; should be written with 0.

Capture/Compare Unit 6 (CCU6)

PISEL2

Port Input Select Register 2

Reset Value: 0000 0000_H



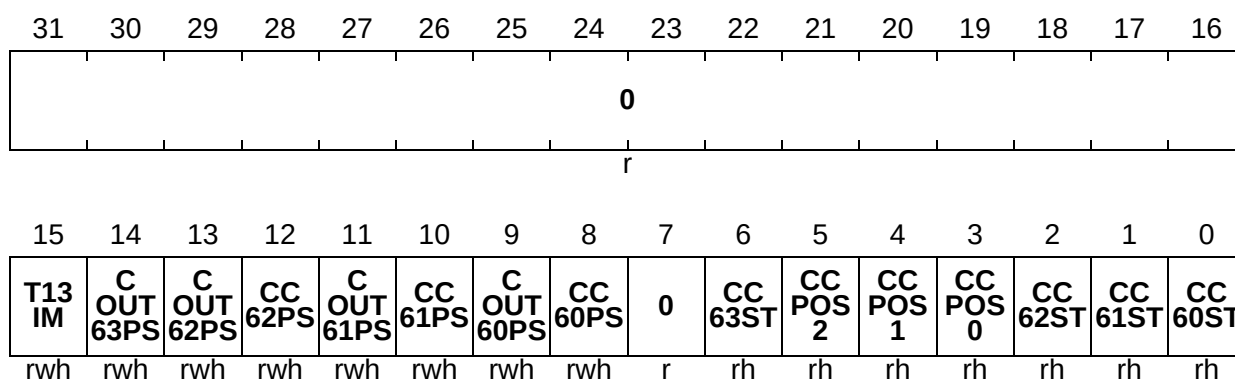
Field	Bits	Type	Description
IST13HR	[1:0]	rw	Input Select for T13HR This bit field defines the port pin that is used for the T13HR input signal. 00 The input pin for T13HR_I0. 01 The input pin for T13HR_I1. 10 The input pin for T13HR_I2. 11 The input pin for T13HR_I3.
0	[31:2]	r	Reserved ; read as 0; should be written with 0.

Register CMPSTAT contains status bits monitoring the current capture and compare state and control bits defining the active/passive state of the compare channels.

CMPSTAT

Compare State Register

Reset Value: 0000 0000_H



Capture/Compare Unit 6 (CCU6)

Field	Bits	Type	Description
CC60ST CC61ST CC62ST CC63ST ¹⁾	0 1 2 6	rh	Capture/Compare State Bits Bits CC6xST monitor the state of the capture/compare channels. Bits CC6xST (x=0, 1, 2) are related to T12, bit CC63ST is related to T13. 0 In compare mode, the timer count is less than the compare value. In capture mode, the selected edge has not yet been detected since the bit has been reset by software the last time. 1 In compare mode, the counter value is greater than or equal to the compare value. In capture mode, the selected edge has been detected.
CCPOS0 CCPOS1 CCPOS2	3 4 5	rh	Sampled Hall Pattern Bits Bits CCPOSx (x=0, 1, 2) are indicating the value of the input Hall pattern that has been compared to the current and expected value. The value is sampled when the event hcrdy (Hall compare ready) occurs. 0 The input CCPOSx has been sampled as 0. 1 The input CCPOSx has been sampled as 1.
CC60PS CC61PS CC62PS COUT60PS COUT61PS COUT62PS COUT63PS ²⁾	8 10 12 9 11 13 14	rwh	Passive State Select for Compare Outputs Bits CC6xPS, COUT6xPS select the state of the corresponding compare channel, which is considered to be the passive state. During the passive state, the passive level (defined in register PSLR) is driven by the output pin. Bits CC6xPS, COUT6xPS (x=0, 1, 2) are related to T12, bit CC63PS is related to T13. 0 The corresponding compare output drives passive level while CC6xST is 0. 1 The corresponding compare output drives passive level while CC6xST is 1. In capture mode, these bits are not used.
T13IM³⁾	15	rwh	T13 Inverted Modulation Bit T13IM inverts the T13 signal for the modulation of the CC6x and COUT6x (x = 0, 1, 2) signals. 0 T13 output is not inverted. 1 T13 output is inverted for further modulation.
0	7, [31:16]	r	Reserved ; read as 0; should be written with 0.

¹⁾ These bits are set and reset according to the T12, T13 switching rules

Capture/Compare Unit 6 (CCU6)

- 2) These bits have shadow bits and are updated in parallel to the capture/compare registers of T12, T13 respectively. A read action targets the actually used values, whereas a write action targets the shadow bits.
- 3) This bit has a shadow bit and is updated in parallel to the compare and period registers of T13. A read action targets the actually used values, whereas a write action targets the shadow bit.

Register CMPMODIF contains control bits allowing for modification by software of the Capture/Compare state bits.

CMPMODIF

Compare State Modification Register

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	MCC 63R		0		MCC 62R	MCC 61R	MCC 60R	0	MCC 63S		0		MCC 62S	MCC 61S	MCC 60S
r	w		r		w	w	w	r	w		r		w	w	w

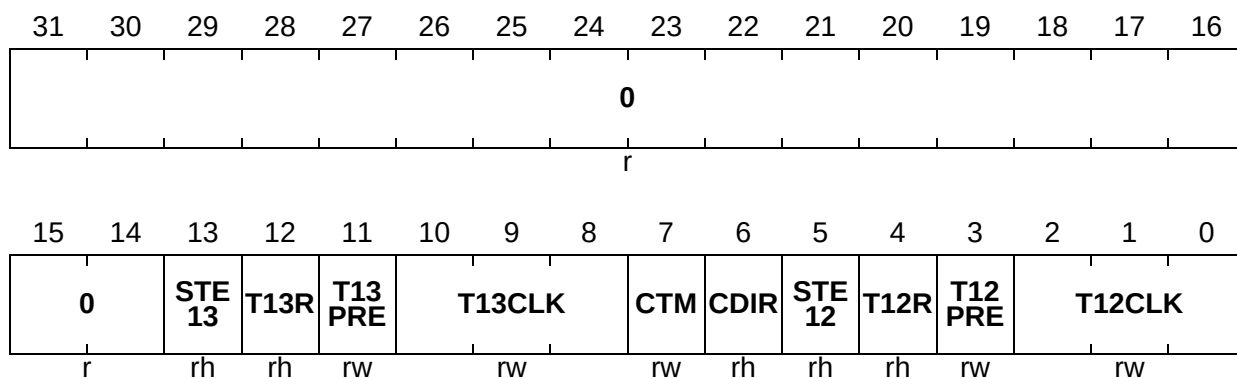
Field	Bits	Type	Description
MCC60S	0	w	Capture/Compare Status Modification Bits These bits are used to bits to set (MCC6xS) or to reset (MCC6xR) the corresponding bits CC6xST by software. This feature allows the user to individually change the status of the output lines by software, e.g. when the corresponding compare timer is stopped. This allows a bit manipulation of CC6xST-bits by a single data write action. The following functionality of a write access to bits concerning the same capture/compare state bit is provided: MCC6xR, MCC6xS = 0,0 Bit CC6xST is not changed. 0,1 Bit CC6xST is set. 1,0 Bit CC6xST is reset. 1,1 reserved (toggle)
MCC61S	1		
MCC62S	2		
MCC63S	6		
MCC60R	8		
MCC61R	9		
MCC62R	10		
MCC63R	14		
0	[5:3], 7, [13:11], [31:15]	r	Reserved ; read as 0; should be written with 0.

Register TCTR0 controls the basic functionality of both timers T12 and T13.

TCTR0

Timer Control Register 0

Reset Value: 0000 0000_H



Field	Bits	Type	Description
T12CLK	[2:0]	rw	Timer T12 Input Clock Select Selects the input clock for timer T12 which is derived from the peripheral clock according to the equation $f_{T12} = f_{CCU} / 2^{<T12CLK>}$. 000 $f_{T12} = f_{CCU}$ 001 $f_{T12} = f_{CCU} / 2$ 010 $f_{T12} = f_{CCU} / 4$ 011 $f_{T12} = f_{CCU} / 8$ 100 $f_{T12} = f_{CCU} / 16$ 101 $f_{T12} = f_{CCU} / 32$ 110 $f_{T12} = f_{CCU} / 64$ 111 $f_{T12} = f_{CCU} / 128$
T12PRE ¹⁾	3	rw	Timer T12 Prescaler Bit In order to support higher clock frequencies, an additional prescaler factor of 1/256 can be enabled for the prescaler for T12. 0 The additional prescaler for T12 is disabled. 1 The additional prescaler for T12 is enabled.
T12R ²⁾	4	rh	Timer T12 Run Bit T12R starts and stops timer T12. It is set/reset by software by setting bits T12RR or T12RS or it is reset by hardware according to the function defined by bit field T12SSC. 0 Timer T12 is stopped. 1 Timer T12 is running.

Capture/Compare Unit 6 (CCU6)

Field	Bits	Type	Description
STE12	5	rh	Timer T12 Shadow Transfer Enable Bit STE12 enables or disables the shadow transfer of the T12 period value, the compare values and passive state select bits and levels from their shadow registers to the actual registers if a T12 shadow transfer event is detected. Bit STE12 is cleared by hardware after the shadow transfer. A T12 shadow transfer event is a period-match while counting up or a one-match while counting down. 0 The shadow register transfer is disabled. 1 The shadow register transfer is enabled.
CDIR	6	rh	Count Direction of Timer T12 This bit is set/reset according to the counting rules of T12. 0 T12 counts up. 1 T12 counts down.
CTM	7	rw	T12 Operating Mode 0 Edge-aligned Mode: T12 always counts up and continues counting from zero after reaching the period value. 1 Center-aligned Mode: T12 counts down after detecting a period-match and counts up after detecting a one-match.
T13CLK	[10:8]	rw	Timer T13 Input Clock Select Selects the input clock for timer T13 which is derived from the peripheral clock according to the equation $f_{T13} = f_{CCU} / 2^{<T13CLK>}$. 000 $f_{T13} = f_{CCU}$ 001 $f_{T13} = f_{CCU} / 2$ 010 $f_{T13} = f_{CCU} / 4$ 011 $f_{T13} = f_{CCU} / 8$ 100 $f_{T13} = f_{CCU} / 16$ 101 $f_{T13} = f_{CCU} / 32$ 110 $f_{T13} = f_{CCU} / 64$ 111 $f_{T13} = f_{CCU} / 128$

Capture/Compare Unit 6 (CCU6)

Field	Bits	Type	Description
T13PRE³⁾	11	rw	Timer T13 Prescaler Bit In order to support higher clock frequencies, an additional prescaler factor of 1/256 can be enabled for the prescaler for T13. 0 The additional prescaler for T13 is disabled. 1 The additional prescaler for T13 is enabled.
T13R⁴⁾	12	rh	Timer T13 Run Bit T13R starts and stops timer T13. It is set/reset by software by setting bits T13RR or T13RS or it is set/reset by hardware according to the function defined by bit fields T13SSC, T13TEC and T13TED. 0 Timer T13 is stopped. 1 Timer T13 is running.
STE13	13	rh	Timer T13 Shadow Transfer Enable Bit STE13 enables or disables the shadow transfer of the T13 period value, the compare value and passive state select bit and level from their shadow registers to the actual registers if a T13 shadow transfer event is detected. Bit STE13 is cleared by hardware after the shadow transfer. A T13 shadow transfer event is a period-match. 0 The shadow register transfer is disabled. 1 The shadow register transfer is enabled.
0	[31:14]	r	Reserved ; read as 0; should be written with 0.

¹⁾ not available in first silicon

²⁾ A concurrent set/reset action on T12R (from T12SSC, T12RR or T12RS) will have no effect. The bit T12R will remain unchanged.

³⁾ not available in first silicon

⁴⁾ A concurrent set/reset action on T13R (from T13SSC, T13TEC, T13RR or T13RS) will have no effect. The bit T12R will remain unchanged.

Note: A write action to the bit fields T12CLK or T12PRE is only taken into account while the timer T12 is not running (T12R=0). A write action to the bit fields T13CLK or T13PRE is only taken into account while the timer T13 is not running (T13R=0).

Capture/Compare Unit 6 (CCU6)

Register TCTR2 controls the single-shot and the synchronization functionality of both timers T12 and T13. Both timers can run in single-shot mode. In this mode they stop their counting sequence automatically after one counting period with a count value of zero. The single-shot mode and the synchronization feature of T13 to T12 allow the generation of events with a programmable delay after well-defined PWM actions of T12. For example, this feature can be used to trigger AD conversions after a specified delay (to avoid problems due to switching noise) synchronously to a PWM event.

TCTR2

Timer Control Register 2

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0				T13 RSEL		T12 RSEL		SUS CFG	T13 TED		T13 TEC		T13 SSC	T12 SSC	
r				rw		rw		rw	rw		rw		rw	rw	

Field	Bits	Type	Description
T12SSC	0	rw	Timer T12 Single Shot Control This bit controls the single shot-mode of T12. 0 The single-shot mode is disabled, no hardware action on T12R. 1 The single shot mode is enabled, the bit T12R is reset by hardware if - T12 reaches its period value in edge-aligned mode - T12 reaches the value 1 while down counting in center-aligned mode. In parallel to the reset action of bit T12R, the bits CC6xST (x=0, 1, 2) are reset.
T13SSC	1	rw	Timer T13 Single Shot Control This bit controls the single shot-mode of T13. 0 No hardware action on T13R 1 The single-shot mode is enabled, the bit T13R is reset by hardware if T13 reaches its period value. In parallel to the reset action of bit T13R, the bit CC63ST is reset.

Capture/Compare Unit 6 (CCU6)

Field	Bits	Type	Description
T13TEC	[4:2]	rw	T13 Trigger Event Control bit field T13TEC selects the trigger event to start T13 (automatic set of T13R for synchronization to T12 compare signals) according to following combinations: 000 no action 001 set T13R on a T12 compare event on channel 0 010 set T13R on a T12 compare event on channel 1 011 set T13R on a T12 compare event on channel 2 100 set T13R on any T12 compare event on the channels 0, 1, or 2 101 set T13R upon a period-match of T12 110 set T13R upon a zero-match of T12 (while counting up) 111 set T13R on any edge of inputs CCPOSx
T13TED¹⁾	[6:5]	rw	Timer T13 Trigger Event Direction Bit field T13TED delivers additional information to control the automatic set of bit T13R in the case that the trigger action defined by T13TEC is detected. 00 reserved, no action 01 while T12 is counting up 10 while T12 is counting down 11 independent on the count direction of T12
SUSCFG	7	rw	Suspend Configuration Bit SUSCFG defines the behavior of the module while the suspend request is active (independent of the status of the acknowledge signal). In any case, the timers T12 and T13 are stopped when reaching the end of their period. 0 No additional action. 1 The outputs enabled for trap functionality are set to their passive values; see Figure 7-23 and Figure 7-24 .

Field	Bits	Type	Description
T12RSEL	[9:8]	rw	Timer T12 External Run Selection Bit field T12RSEL defines the event of signal T12HR that can set the run bit T12R by hardware. 00 The external setting of T12R is disabled. 01 Bit T12R is set if a rising edge of signal T12HR is detected. 10 Bit T12R is set if a falling edge of signal T12HR is detected. 11 Bit T12R is set if an edge of signal T12HR is detected.
T13RSEL	[11:10]	rw	Timer T13 External Run Selection Bit field T13RSEL defines the event of signal T13HR that can set the run bit T13R by hardware. 00 The external setting of T13R is disabled. 01 Bit T13R is set if a rising edge of signal T13HR is detected. 10 Bit T13R is set if a falling edge of signal T13HR is detected. 11 Bit T13R is set if an edge of signal T13HR is detected.
0	[31:12]	r	Reserved ; read as 0; should be written with 0.

¹⁾ Example:

If the timer T13 is intended to start at any compare event on T12 (T13TEC = 100) the trigger event direction can be programmed to

- counting up >> a T12 channel 0, 1, 2 compare match triggers T13R only while T12 is counting up
- counting down >> a T12 channel 0, 1, 2 compare match triggers T13R only while T12 is counting down
- independent from bit CDIR >> each T12 channel 0, 1, 2 compare match triggers T13R

The timer count direction is taken from the value of bit CDIR. As a result, if T12 is running in edge-aligned mode (counting up only), T13 can only be started automatically if bit field T13TED = 01 or 11.

Capture/Compare Unit 6 (CCU6)

Register TCTR4 allows the software control of the run bits T12R and T13R by independent set and reset conditions. Furthermore, the timers can be reset (while running) and the bits STE12 and STE13 can be controlled by software.

TCTR4

Timer Control Register 4

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
T13 STD	T13 STR	0			T13 RES	T13 RS	T13 RR	T12 STD	T12 STR	0		DT RES	T12 RES	T12 RS	T12 RR
w	w	r			w	w	w	w	w	r		w	w	w	w

Field	Bits	Type	Description
T12RR	0	w	Timer T12 Run Reset Setting this bit resets the T12R bit. 0 T12R is not influenced. 1 T12R is cleared, T12 stops counting.
T12RS	1	w	Timer T12 Run Set Setting this bit sets the T12R bit. 0 T12R is not influenced. 1 T12R is set, T12 counts.
T12RES	2	w	Timer T12 Reset 0 No effect on T12. 1 The T12 counter register is reset to zero. The switching of the output signals is according to the switching rules. Setting of T12RES has no impact on bit T12R.
DTRES	3	w	Dead-Time Counter Reset 0 No effect on the dead-time counters. 1 The three dead-time counter channels are reset to zero.
T12STR	6	w	Timer T12 Shadow Transfer Request 0 No action 1 STE12 is set, enabling the shadow transfer.

Capture/Compare Unit 6 (CCU6)

Field	Bits	Type	Description
T12STD	7	w	Timer T12 Shadow Transfer Disable 0 No action 1 STE12 is reset without triggering the shadow transfer.
T13RR	8	w	Timer T13 Run Reset Setting this bit resets the T13R bit. 0 T13R is not influenced. 1 T13R is cleared, T13 stops counting.
T13RS	9	w	Timer T13 Run Set Setting this bit sets the T13R bit. 0 T13R is not influenced. 1 T13R is set, T13 counts.
T13RES	10	w	Timer T13 Reset 0 No effect on T13. 1 The T13 counter register is reset to zero. The switching of the output signals is according to the switching rules. Setting of T13RES has no impact on bit T13R.
T13STR	14	w	Timer T13 Shadow Transfer Request 0 No action 1 STE13 is set, enabling the shadow transfer.
T13STD	15	w	Timer T13 Shadow Transfer Disable 0 No action 1 STE13 is reset without triggering the shadow transfer.
0	[5:4], [13:11], [31:16]	r	Reserved ; read as 0; should be written with 0.

Note: A simultaneous write of a 1 to bits which set and reset the same bit will trigger no action. The corresponding bit will remain unchanged.

7.2.2 Timer12 - Related Registers

The generation of the patterns for a 3-channel pulse width modulation (PWM) is based on timer T12. The registers related to timer T12 can be concurrently updated (with well-defined conditions) in order to ensure consistency of the three PWM channels.

Timer T12 supports capture and compare modes, which can be independently selected for the three channels CC60, CC61 and CC62.

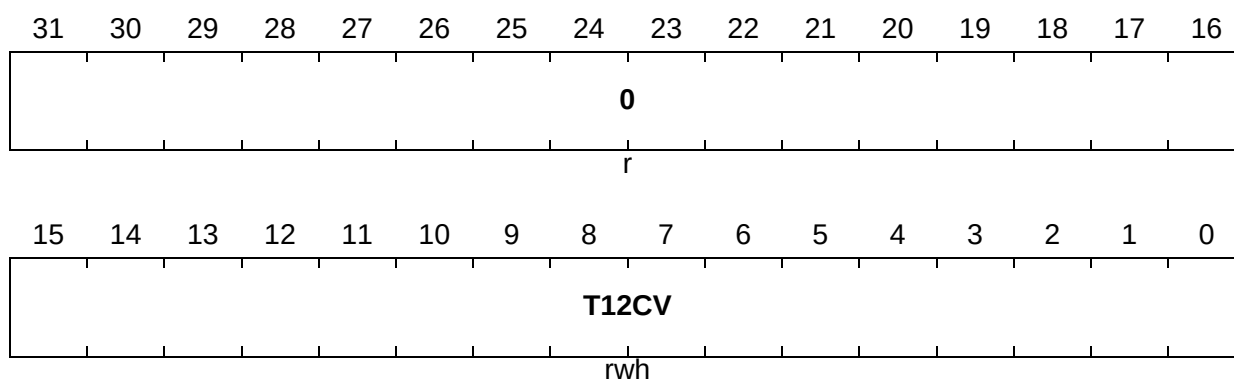
Register T12 represents the counting value of timer T12. It can only be written while the timer T12 is stopped. Write actions while T12 is running are not taken into account. Register T12 can always be read by software.

In edge-aligned mode, T12 only counts up, whereas in center-aligned mode, T12 can count up and down.

T12

Timer T12 Counter Register

Reset Value: 0000 0000_H



Field	Bits	Type	Description
T12CV	[15:0]	rwh	Timer 12 Counter Value This register represents the 16-bit counter value of Timer12.
0	[31:16]	r	Reserved ; read as 0; should be written with 0.

Note: While timer T12 is stopped, the internal clock divider is reset in order to ensure reproducible timings and delays.

Note: The timer period, compare values, passive state selects bits and passive levels bits for both timers are written to shadow registers and not directly to the actual registers. Thus the values for a new output signal can be programmed without disturbing the currently generated signal(s). The transfer from the shadow registers to the actual registers is enabled by setting the respective shadow transfer enable bit STEx.

If the transfer is enabled the shadow registers are copied to the respective registers as soon as the associated timer reaches the value zero the next time

Capture/Compare Unit 6 (CCU6)

(being cleared in edge aligned mode or counting down from 1 in center aligned mode). When timer T12 is operating in center aligned mode, it will also copy the registers (if enabled by STE12) if it reaches the currently programmed period value (counting up).

When a timer is stopped (TxR=0, the shadow transfer takes place immediately if the corresponding bit STE_x is set.

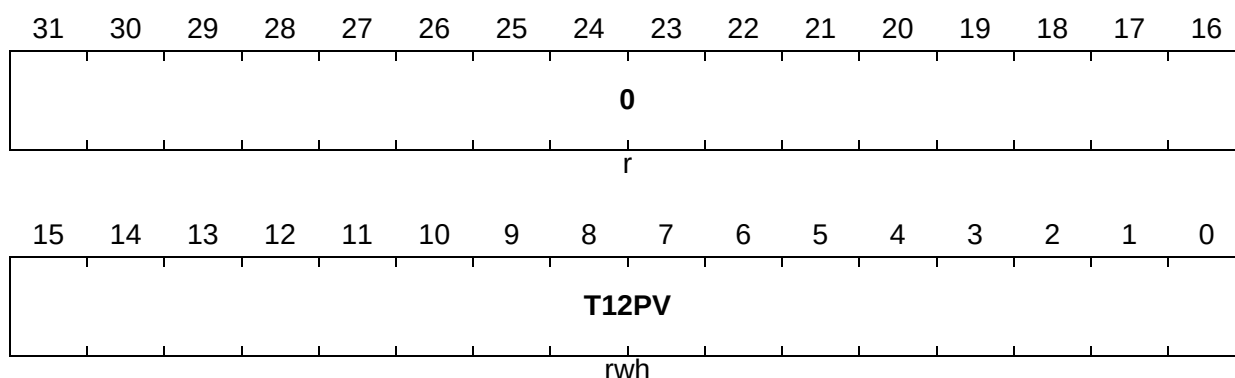
After the transfer the respective bit STE_x is cleared automatically.

Register T12PR contains the period value for timer T12. The period value is compared to the actual counter value of T12 and the resulting counter actions depend on the defined counting rules. This register has a shadow register and the shadow transfer is controlled by bit STE12. A read action by software delivers the value which is currently used for the compare action, whereas the write action targets a shadow register. The shadow register structure allows a concurrent update of all T12-related values.

T12PR

Timer T12 Period Register

Reset Value: 0000 0000_H



Field	Bits	Type	Description
T12PV	[15:0]	rwh	T12 Period Value The value T12PV defines the counter value for T12, which leads to a period-match. When reaching this value, the timerT12 is set to zero (edge-aligned mode) or changes its count direction to down counting (center-aligned mode).
0	[31:16]	r	Reserved ; read as 0; should be written with 0.

Capture/Compare Unit 6 (CCU6)

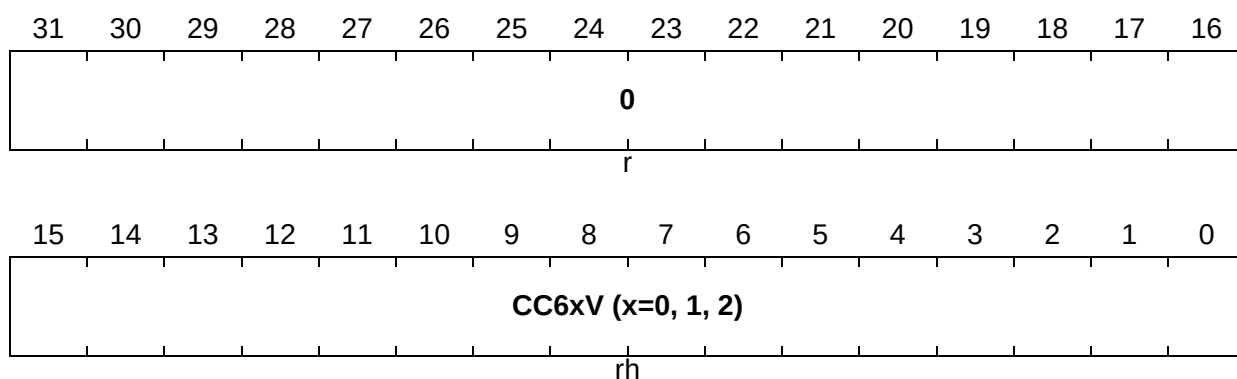
In compare mode, the registers CC6xR (x=0,1,2) are the actual compare registers for T12. The values stored in CC6xR are compared (all three channels in parallel) to the counter value of T12. In capture mode, the current value of the T12 counter register is captured by registers CC6xR if the corresponding capture event is detected.

The registers CC6xR can only be read by software, the modification of the value is done by a shadow register transfer from register CC6xSR. The corresponding shadow registers CC6xSR can be read and written by software. In capture mode, the value of the T12 counter register can also be captured by registers CC6xSR if the selected capture event is detected (depending on the selected mode).

CC6xR (x=0, 1, 2)

Capture/Compare Register for Channel CC6x

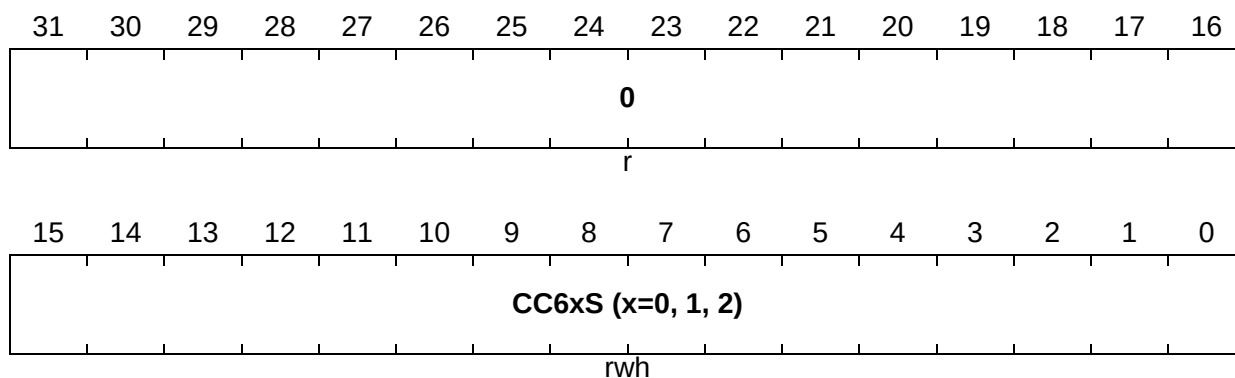
Reset Value: 0000 0000_H



Field	Bits	Type	Description
CC6xV (x=0, 1, 2)	[15:0]	rh	Channel x Capture/Compare Value In compare mode, the bit fields CC6xV contain the values, that are compared to the T12 counter value. In capture mode, the captured value of T12 can be read from these registers.
0	[31:16]	r	Reserved ; read as 0; should be written with 0.

CC6xSR (x=0, 1, 2)

Capture/Compare Shadow Register for Channel CC6x Reset Value: 0000 0000_H



Field	Bits	Type	Description
CC6xS (x=0, 1, 2)	[15:0]	rwh	Shadow Register for Channel x Capture/Compare Value In compare mode, the bit fields contents of CC6xS are transferred to the bit fields CC6xV during a shadow transfer. In capture mode, the captured value of T12 can be read from these registers.
0	[31:16]	r	Reserved ; read as 0; should be written with 0.

Capture/Compare Unit 6 (CCU6)

Register T12DTC controls the dead-time generation for the timer T12 compare channels. Each channel can be independently enabled/disabled for dead-time generation. If enabled, the transition from passive state to active state is delayed by the value defined by bit field DTM. The dead-time counter can only be reloaded while it is zero.

T12DTC

Dead-Time Control Register for Timer12

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	DTR 2	DTR 1	DTR 0	0	DTE 2	DTE 1	DTE 0	DTM							
r	rh	rh	rh	r	rw	rw	rw	rw							

Field	Bits	Type	Description
DTM	[7:0]	rw	Dead-Time Bit field DTM determines the programmable delay between switching from the passive state to the active state of the selected outputs. The switching from the active state to the passive state is not delayed.
DTE2 DTE1 DTE0	10 9 8	rw	Dead Time Enable Bits Bits DTE0..DTE2 enable and disable the dead time generation for each compare channel (0, 1, 2) of timer T12. 0 Dead time generation is disabled. The corresponding outputs switch from the passive state to the active state (according to the actual compare status) without any delay. 1 Dead time generation is enabled. The corresponding outputs switch from the passive state to the active state (according to the compare status) with the delay programmed in bit field DTM.

Capture/Compare Unit 6 (CCU6)

Field	Bits	Type	Description
DTR2	14	rh	Dead Time Run Indication Bits Bits DTR0..DTR2 indicate the status of the dead time generation for each compare channel (0, 1, 2) of timer T12. 0 The value of the corresponding dead time counter channel is 0. 1 The value of the corresponding dead time counter channel is not 0.
DTR1	13		
DTR0	12		
0	11, [31:15]	r	Reserved ; read as 0; should be written with 0.

*Note: The dead time counters are clocked with the same frequency as T12.
 This structure allows symmetrical dead time generation in center-aligned and in edge-aligned PWM mode. A duty cycle of 50% leads to CC6x, COUT6x switched on for: $0.5 \cdot \text{period} - \text{dead time}$.*

Note: The dead-time counters are not reset by bit T12RES, but by bit DTRES.

7.2.3 Timer13 - Related Registers

The generation of the patterns for a single channel pulse width modulation (PWM) is based on timer T13. The registers related to timer T13 can be concurrently updated (with well-defined conditions) in order to ensure consistency of the PWM signal. T13 can be synchronized to several timer T12 events.

Timer T13 only supports compare mode on its compare channel CC63.

Register T13 represents the counting value of timer T13. It can only be written while the timer T13 is stopped. Write actions while T13 is running are not taken into account.

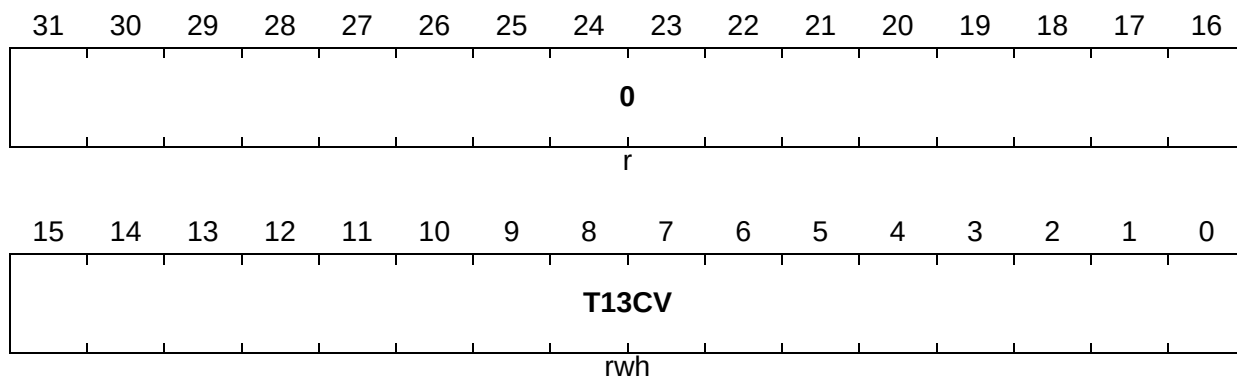
Register T13 can always be read by software.

Timer T13 only supports edge-aligned mode (counting up).

T13

Timer T13 Register

Reset Value: 0000 0000_H



Field	Bits	Type	Description
T13CV	[15:0]	rwh	Timer 13 Counter Value This register represents the 16-bit counter value of Timer13.
0	[31:16]	r	Reserved ; read as 0; should be written with 0.

Note: While timer T13 is stopped, the internal clock divider is reset in order to ensure reproducible timings and delays.

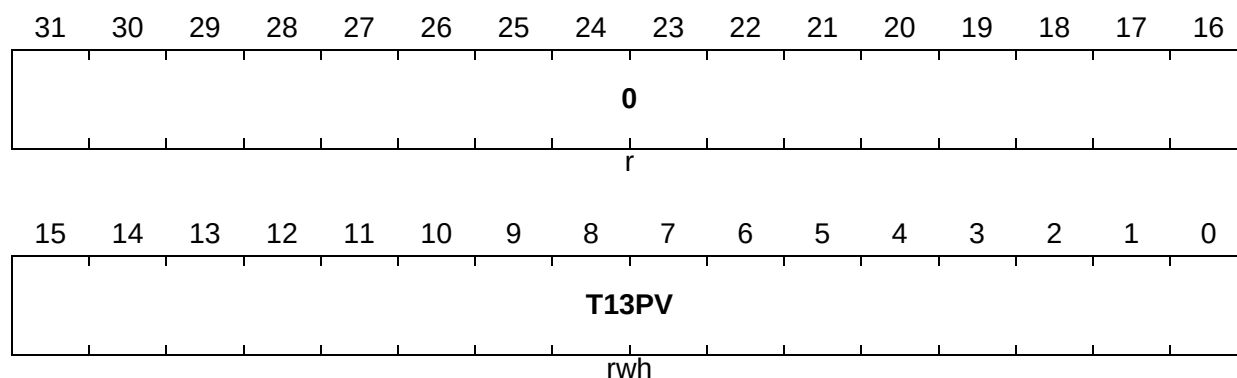
Capture/Compare Unit 6 (CCU6)

Register T13PR contains the period value for timer T13. The period value is compared to the actual counter value of T13 and the resulting counter actions depend on the defined counting rules. This register has a shadow register and the shadow transfer is controlled by bit STE13. A read action by software delivers the value which is currently used for the compare action, whereas the write action targets a shadow register. The shadow register structure allows a concurrent update of all T13-related values.

T13PR

Timer T13 Period Register

Reset Value: 0000 0000_H



Field	Bits	Type	Description
T13PV	[15:0]	rwh	T13 Period Value The value T13PV defines the counter value for T13, which leads to a period-match. When reaching this value, the timer T13 is set to zero.
0	[31:16]	r	Reserved ; read as 0; should be written with 0.

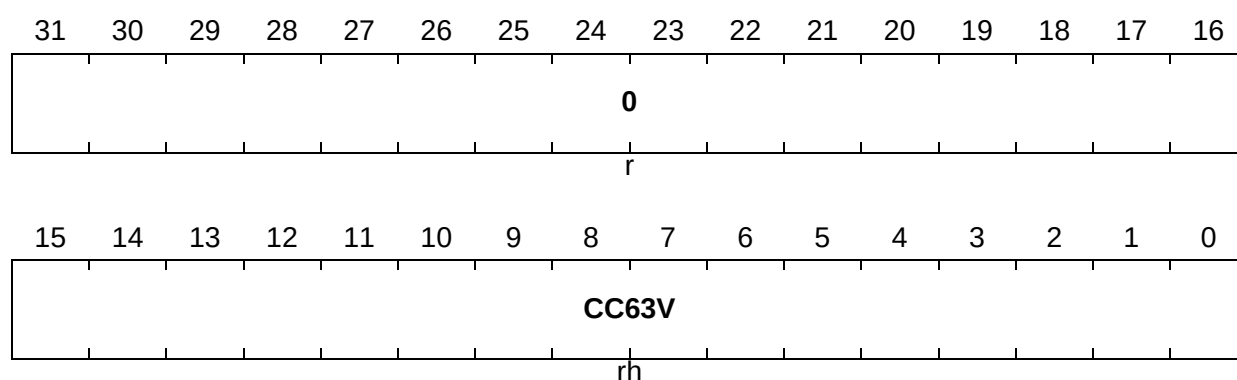
Capture/Compare Unit 6 (CCU6)

Register CC63R is the actual compare register for T13. The values stored in CC63R is compared to the counter value of T13. The register CC63R can only be read by software, the modification of the value is done by a shadow register transfer from register CC63SR. The corresponding shadow register CC63SR can be read and written by software.

CC63R

Compare Register for Channel CC63

Reset Value: 0000 0000_H

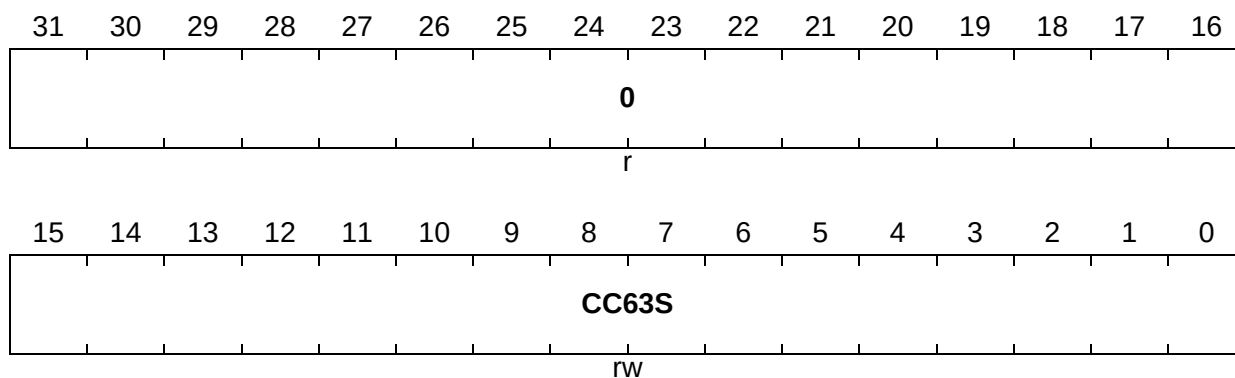


Field	Bits	Type	Description
CC63V	[15:0]	rh	Channel CC63 Compare Value The bit field CC63V contains the value that is compared to the T13 counter value.
0	[31:16]	r	Reserved ; read as 0; should be written with 0.

CC63SR

Compare Shadow Register for Channel CC63

Reset Value: 0000 0000_H



Field	Bits	Type	Description
CC63S	[15:0]	rw	Shadow Register for Channel CC63 Compare Value The bit field contents of CC63S is transferred to the bit field CC63V during a shadow transfer.
0	[31:16]	r	Reserved ; read as 0; should be written with 0.

7.2.4 Modulation Control Registers

Register MODCTR contains control bits enabling the modulation of the corresponding output signal by PWM pattern generated by the timers T12 and T13. Furthermore, the multi-channel mode can be enabled as additional modulation source for the output signals.

MODCTR

Modulation Control Register

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ECT 130	0	T13MODEN						MCM EN	0	T12MODEN					
rw	r	rw						rw	r	rw					

Field	Bits	Type	Description
T12MODEN	[5:0]	rw	T12 Modulation Enable Setting these bits enables the modulation of the corresponding compare channel by a PWM pattern generated by timer T12. The bit positions are corresponding to the following output signals: bit 0 modulation of CC60 bit 1 modulation of COUT60 bit 2 modulation of CC61 bit 3 modulation of COUT61 bit 4 modulation of CC62 bit 5 modulation of COUT62 The enable feature of the modulation is defined as follows: 0 The modulation of the corresponding output signal by a T12 PWM pattern is disabled. 1 The modulation of the corresponding output signal by a T12 PWM pattern is enabled.

Capture/Compare Unit 6 (CCU6)

Field	Bits	Type	Description
MCMEN	7	rw	Multi-Channel Mode Enable 0 The modulation of the corresponding output signal by a multi-channel pattern according to bit field MCMOUT is disabled. 1 The modulation of the corresponding output signal by a multi-channel pattern according to bit field MCMOUT is enabled.
T13MODEN	[13:8]	rw	T13 Modulation Enable Setting these bits enables the modulation of the corresponding compare channel by a PWM pattern generated by timer T13. The bit positions are corresponding to the following output signals: bit 8 modulation of CC60 bit 9 modulation of COUT60 bit 10 modulation of CC61 bit 11 modulation of COUT61 bit 12 modulation of CC62 bit 13 modulation of COUT62 The enable feature of the modulation is defined as follows: 0 The modulation of the corresponding output signal by a T13 PWM pattern is disabled. 1 The modulation of the corresponding output signal by a T13 PWM pattern is enabled.
ECT130	15	rw	Enable Compare Timer T13 Output 0 The alternate output function COUT63 is disabled. 1 The alternate output function COUT63 is enabled for the PWM signal generated by T13.
0	6, 14, [31:16]	r	Reserved ; read as 0; should be written with 0.

Capture/Compare Unit 6 (CCU6)

Register TRPCTR controls the trap functionality. It contains independent enable bits for each output signal and control bits to select the behavior in case of a trap condition. The trap condition is a low level on the CTRAP input pin, which is monitored (inverted level) by bit TRPF (in register IS). While TRPF = 1 (trap input active), the trap state bit TRPS (in register IS) is set to 1.

TRPCTR

Trap Control Register

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRP PEN	TRP EN 13	TRPEN						0					TRP M2	TRP M1	TRP M0
rw	rw	rw						r					rw	rw	rw

Field	Bits	Type	Description
TRPM1, TRPM0	[1:0]	rw	Trap Mode Control Bits 1, 0 These two bits define the behavior of the selected outputs when leaving the trap state after the trap condition has become inactive again. A synchronization to the timer driving the PWM pattern permits to avoid unintended short pulses when leaving the trap state. The combination (TRPM1, TRPM0) leads to: 00 The trap state is left (return to normal operation according to TRPM2) when a zero-match of T12 (while counting up) is detected (synchronization to T12). 01 The trap state is left (return to normal operation according to TRPM2) when a zero-match of T13 is detected (synchronization to T13). 10 reserved 11 The trap state is left (return to normal operation according to TRPM2) immediately without any synchronization to T12 or T13.

Field	Bits	Type	Description
TRPM2	2	rw	Trap Mode Control Bit 2 0 The trap state can be left (return to normal operation = bit TRPS = 0) as soon as the input $\overline{\text{CTRAP}}$ becomes inactive. Bit TRPF is automatically cleared by hardware if the input pin $\overline{\text{CTRAP}}$ becomes 1. Bit TRPS is automatically cleared by hardware if bit TRPF is 0 and if the synchronization condition (according to TRPM0,1) is detected. 1 The trap state can be left (return to normal operation = bit TRPS= 0) as soon as bit TRPF is reset by software after the input $\overline{\text{CTRAP}}$ becomes inactive (TRPF is not cleared by hardware). Bit TRPS is automatically cleared by hardware if bit TRPF = 0 and if the synchronization condition (according to TRPM0,1) is detected.
TRPEN	[13:8]	rw	Trap Enable Control Setting these bits enables the trap functionality for the following corresponding output signals: bit 8 trap functionality of CC60 bit 9 trap functionality of COUT60 bit 10 trap functionality of CC61 bit 11 trap functionality of COUT61 bit 12 trap functionality of CC62 bit 13 trap functionality of COUT62 The enable feature of the trap functionality is defined as follows: 0 The trap functionality of the corresponding output signal is disabled. The output state is independent from bit TRPS. 1 The trap functionality of the corresponding output signal is enabled. The output is set to the passive state while TRPS= 1.

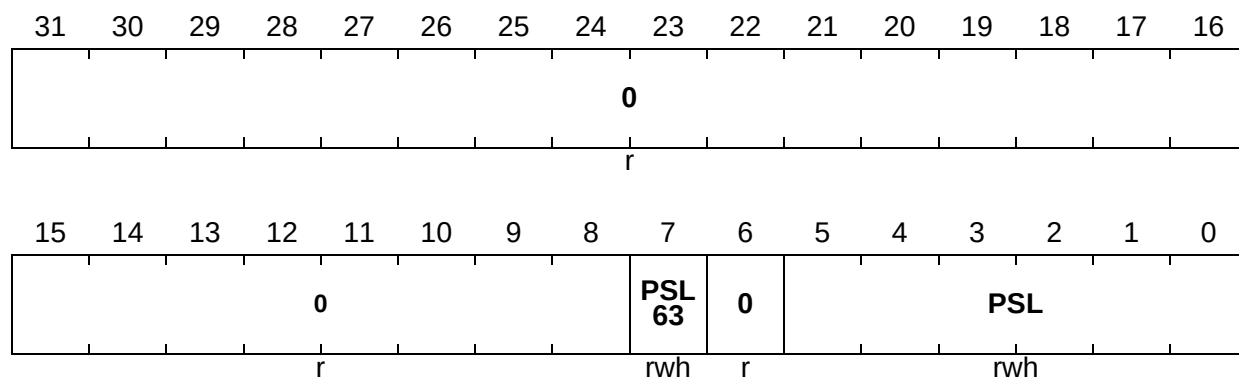
Field	Bits	Type	Description
TRPEN13	14	rw	Trap Enable Control for Timer T13 0 The trap functionality for T13 is disabled. Timer T13 (if selected and enabled) provides PWM functionality even while TRPS = 1. 1 The trap functionality for T13 is enabled. The timer T13 PWM output signal is set to the passive state while TRPS= 1.
TRPPEN	15	rw	Trap Pin Enable 0 The trap functionality based on the input pin $\overline{\text{CTRAP}}$ is disabled. A trap can only be generated by software by setting bit TRPF. 1 The trap functionality based on the input pin $\overline{\text{CTRAP}}$ is enabled. A trap can be generated by software by setting bit TRPF or by $\overline{\text{CTRAP}} = 0$.
0	[7:3], [31:16]	r	Reserved ; read as 0; should be written with 0.

Register PSLR defines the passive state level driven by the output pins of the module. The passive state level is the value that is driven by the port pin during the passive state of the output. During the active state, the corresponding output pin drives the active state level, which is the inverted passive state level. The passive state level permits to adapt the driven output levels to the driver polarity (inverted, not inverted) of the connected power stage.

PSLR

Passive State Level Register

Reset Value: 0000 0000_H



Register MCMOUTS contains bits controlling the output states for multi-channel mode. Furthermore, the appropriate signals for the block commutation by Hall sensors can be

Capture/Compare Unit 6 (CCU6)

Field	Bits	Type	Description
PSL ¹⁾	[5:0]	rwh	Compare Outputs Passive State Level The bits of this bit field define the passive level driven by the module outputs during the passive state. The bit positions are: bit 0 passive level for output CC60 bit 1 passive level for output COUT60 bit 2 passive level for output CC61 bit 3 passive level for output COUT61 bit 4 passive level for output CC62 bit 5 passive level for output COUT62 The value of each bit position is defined as: 0 The passive level is 0. 1 The passive level is 1.
PSL63 ²⁾	7	rwh	Passive State Level of Output COUT63 This bit field defines the passive level of the output pin COUT63. 0 The passive level is 0. 1 The passive level is 1.
0	6, [31:8]	r	Reserved ; read as 0; should be written with 0.

¹⁾ Bit field PSL has a shadow register to allow for updates without undesired pulses on the output lines. The bits are updated with the T12 shadow transfer. A read action targets the actually used values, whereas a write action targets the shadow bits.

²⁾ Bit PSL63 has a shadow register to allow for updates without undesired pulses on the output line. The bit is updated with the T13 shadow transfer. A read action targets the actually used values, whereas a write action targets the shadow bits.

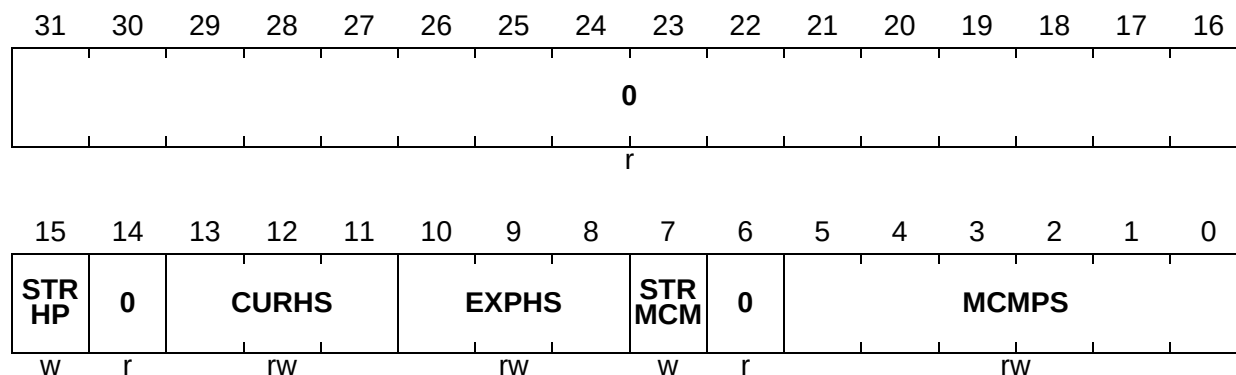
Capture/Compare Unit 6 (CCU6)

selected. This register is a shadow register (that can be written) for register MCMOUT, which indicates the currently active signals.

MCMOUTS

Multi-Channel Mode Output Shadow Register

Reset Value: 0000 0000_H



Field	Bits	Type	Description
MCMP5	[5:0]	rw	Multi-Channel PWM Pattern Shadow bit field MCMP5 is the shadow bit field for bit field MCMP. The multi-channel shadow transfer is triggered according to the transfer conditions defined by register MCMCTR.
STRMCM	7	w	Shadow Transfer Request for MCMP5 Setting this bits during a write action leads to an immediate update of bit field MCMP by the value written to bit field MCMP5. This functionality permits an update triggered by software. When read, this bit always delivers 0. 0 bit field MCMP is updated according to the defined hardware action. The write access to bit field MCMP5 does not modify bit field MCMP. 1 bit field MCMP is updated by the value written to bit field MCMP5.
EXPHS	[10:8]	rw	Expected Hall Pattern Shadow bit field EXPHS is the shadow bit field for bit field EXPH. The bit field is transferred to bit field EXPH if an edge on the hall input pins CCPOSx (x=0, 1, 2) is detected.

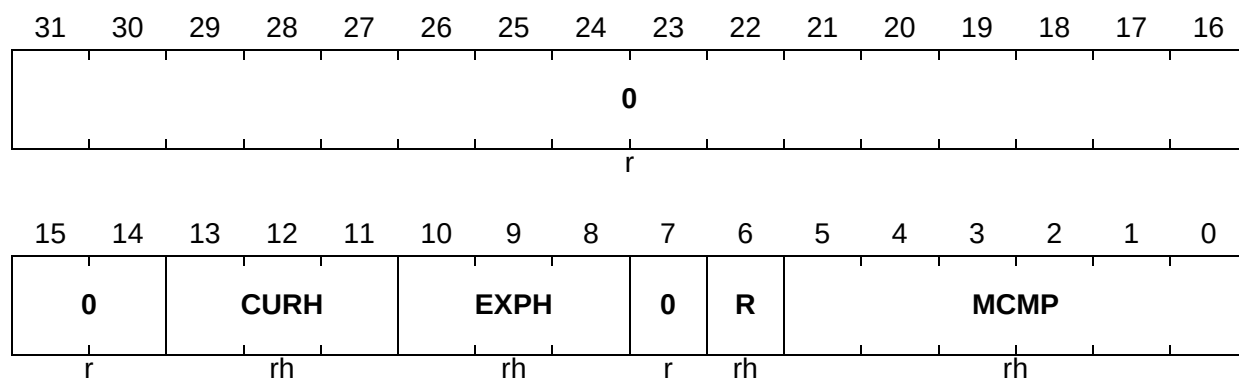
Field	Bits	Type	Description
CURHS	[11:13]	rw	Current Hall Pattern Shadow Bit field CURHS is the shadow bit field for bit field CURH. The bit field is transferred to bit field CURH if an edge on the hall input pins CCPOSx (x=0, 1, 2) is detected.
STRHP	15	w	Shadow Transfer Request for the Hall Pattern Setting this bits during a write action leads to an immediate update of bit fields CURH and EXPH by the value written to bit fields CURHS and EXPH. This functionality permits an update triggered by software. When read, this bit always delivers 0. 0 The bit fields CURH and EXPH are updated according to the defined hardware action. The write access to bit fields CURHS and EXPH does not modify the bit fields CURH and EXPH. 1 The bit fields CURH and EXPH are updated by the value written to the bit fields CURHS and EXPHS.
0	6, 14, [31:16]	r	Reserved ; read as 0; should be written with 0.

Register MCMOUT shows the multi-channel control bits, that are currently used. Register MCMOUT is defined as follows:

MCMOUT

Multi-Channel Mode Output Register

Reset Value: 0000 0000_H



Capture/Compare Unit 6 (CCU6)

Field	Bits	Type	Description
MCMP¹⁾	[5:0]	rh	Multi-Channel PWM Pattern Bit field MCMP is written by a shadow transfer from bit field MCMPS. It contains the output pattern for the multi-channel mode. If this mode is enabled by bit MCMEN in register MODCTR, the output state of the following output signal can be modified: bit 0 multi-channel state for output CC60 bit 1 multi-channel state for output COUT60 bit 2 multi-channel state for output CC61 bit 3 multi-channel state for output COUT61 bit 4 multi-channel state for output CC62 bit 5 multi-channel state for output COUT62 The multi-channel patterns can set the related output to the passive state. 0 The output is set to the passive state. The PWM generated by T12 or T13 are not taken into account. 1 The output can deliver the PWM generated by T12 or T13 (according to register MODCTR).
R	6	rh	Reminder Flag This reminder flag indicates that the shadow transfer from bit field MCMPS to MCMP has been requested by the selected trigger source. This bit is cleared when the shadow transfer takes place and while MCMEN = 0. 0 Currently, no shadow transfer from MCMPS to MCMP is requested. 1 A shadow transfer from MCMPS to MCMP has been requested by the selected trigger source, but it has not yet been executed, because the selected synchronization condition has not yet occurred.

Capture/Compare Unit 6 (CCU6)

Field	Bits	Type	Description
EXPH ²⁾	[10:8]	rh	Expected Hall Pattern Bit field EXPH is written by a shadow transfer from bit field EXPHS. The contents is compared after every detected edge at the hall input pins with the pattern at the hall input pins in order to detect the occurrence of the next desired (=expected) hall pattern or a wrong pattern. If the current hall pattern at the hall input pins is equal to the bit field EXPH, bit CHE (correct hall event) is set and an interrupt request is generated (if enabled by bit ENCHE). If the current hall pattern at the hall input pins is not equal to the bit fields CURH or EXPH, bit WHE (wrong hall event) is set and an interrupt request is generated (if enabled by bit ENWHE).
CURH	[13:11]	rh	Current Hall Pattern Bit field CURH is written by a shadow transfer from bit field CURHS. The contents is compared after every detected edge at the hall input pins with the pattern at the hall input pins in order to detect the occurrence of the next desired (=expected) hall pattern or a wrong pattern. If the current hall input pattern is equal to bit field CURH, the detected edge at the hall input pins has been an invalid transition (e.g. a spike).
0	7, [31:14]	r	Reserved ; read as 0; should be written with 0.

¹⁾ While IDLE = 1, bit field MCMP is cleared.

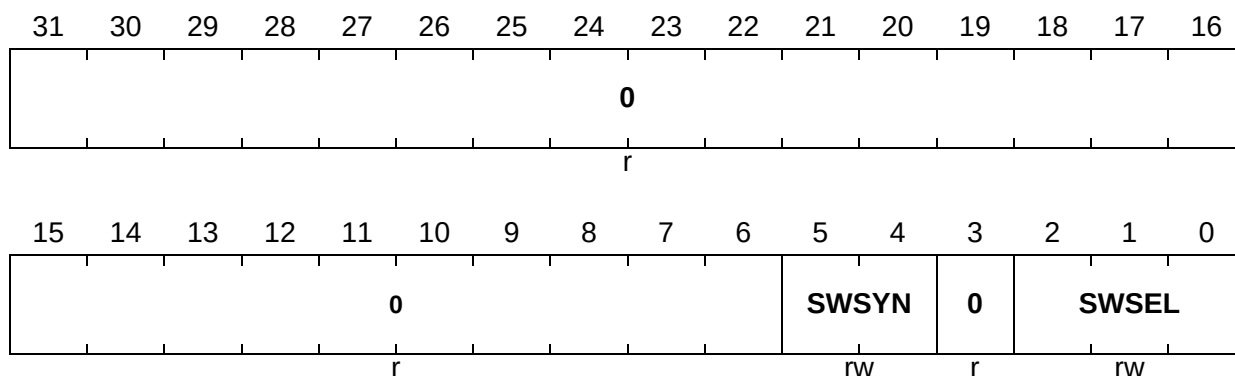
²⁾ The bits in the bit fields EXPH and CURH correspond to the hall patterns at the input pins CCPOSx (x=0, 1, 2) in the order (EXPH.2, EXPH.1, EXPH.0), (CURH.2, CURH.1, CURH.0), (CCPOS2, CCPOS.1, CCPOS0).

Register MCMCTR contains control bits for the multi-channel functionality.

MCMCTR

Multi-Channel Mode Control Register

Reset Value: 0000 0000_H



Field	Bits	Type	Description
SWSEL	[2:0]	rw	Switching Selection Bit field SWSEL selects one of the following trigger request sources (next multi-channel event) for the shadow transfer from MCMPS to MCMP. The trigger request is stored in the reminder flag R until the shadow transfer is done and flag R is cleared automatically with the shadow transfer. The shadow transfer takes place synchronously with an event selected in bit field SWSYN. 000 no trigger request will be generated 001 correct hall pattern on CCPOSx detected 010 T13 period-match detected (while counting up) 011 T12 one-match (while counting down) 100 T12 channel 1 compare-match detected (phase delay function) 101 T12 period match detected (while counting up) else reserved, no trigger request will be generated

Capture/Compare Unit 6 (CCU6)

Field	Bits	Type	Description
SWSYN	[5:4]	rw	Switching Synchronization Bit field SWSYN triggers the shadow transfer between MCMPS and MCMP if it has been requested before (flag R set by an event selected by SWSEL). This feature permits the synchronization of the outputs to the PWM source that is used for modulation (T12 or T13). 00 direct; the trigger event directly causes the shadow transfer 01 T13 zero-match triggers the shadow transfer 10 a T12 zero-match (while counting up) triggers the shadow transfer 11 reserved; no action
0	3, [31:6]	r	Reserved ; read as 0; should be written with 0.

Note: The generation of the shadow transfer request by hardware is only enabled if bit MCMEN = 1.

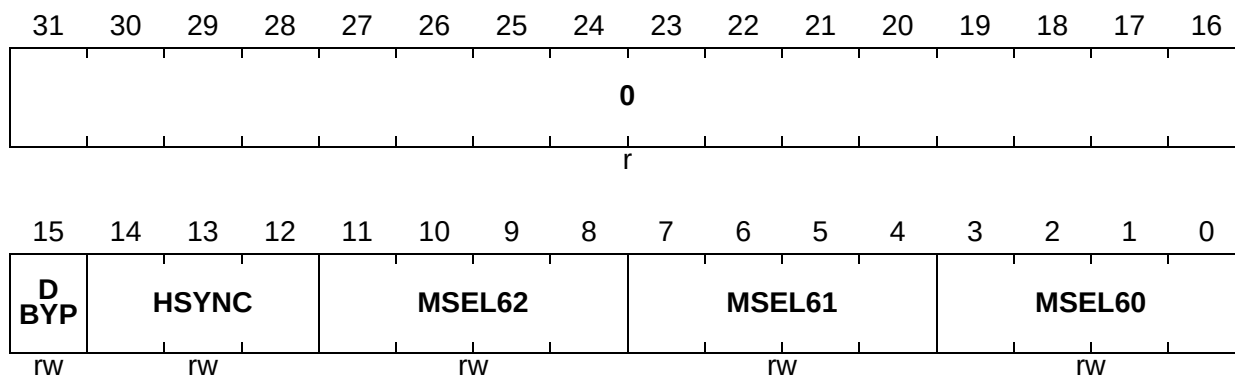
Capture/Compare Unit 6 (CCU6)

Register T12MSEL contains control bits to select the capture/compare functionality of the three channels of timer T12.

T12MSEL

T12 Capture/Compare Mode Select Register

Reset Value: 0000 0000_H



Field	Bits	Type	Description
MSEL60, MSEL61, MSEL62	[3:0], [7:4], [11:8]	rw	Capture/Compare Mode Selection These bit fields select the operating mode of the three timer T12 capture/compare channels. Each channel (n=0, 1, 2) can be programmed individually either for compare or capture operation according to: 0000 Compare outputs disabled, pins CC6n and COUT6n can be used for IO. No capture action. 0001 Compare output on pin CC6n, pin COUT6n can be used for IO. No capture action. 0010 Compare output on pin COUT6n, pin CC6n can be used for IO. No capture action. 0011 Compare output on pins COUT6n and CC6n. 01XX Double-Register Capture modes, see Table 7-3 . 1000 Hall Sensor mode, see Table 7-4 . In order to enable the hall edge detection, all three MSEL6x have to be programmed to Hall Sensor mode. 1001 Hysteresis-like mode, see Table 7-4 . 101X Multi-Input Capture modes, see Table 7-5 . 11XX Multi-Input Capture modes, see Table 7-5 .

Capture/Compare Unit 6 (CCU6)

Field	Bits	Type	Description
HSYNC	[14:12]	rw	Hall Synchronization Bit field HSYNC defines the source for the sampling of the Hall input pattern and the comparison to the current and the expected Hall pattern bit fields. In all modes, a trigger by software by writing a 1 to bit SWHC is possible. 000 Any edge at one of the inputs CCPOSx (x = 0, 1, 2) triggers the sampling. 001 A T13 compare-match triggers the sampling. 010 A T13 period-match triggers the sampling. 011 The Hall sampling triggered by hardware sources is switched off. 100 A T12 period-match (while counting up) triggers the sampling. 101 A T12 one-match (while counting down) triggers the sampling. 110 A T12 compare-match of channel 0 (while counting up) triggers the sampling. 111 A T12 compare-match of channel 0 (while counting down) triggers the sampling.
DBYP	15	rw	Delay Bypass Bit DBYP defines if the source signal for the sampling of the Hall input pattern (selected by HSYNC) uses the dead-time counter DTC0 of timer T12 as additional delay or if the delay is bypassed. 0 The delay bypass is not active. The dead-time counter DTC0 is generating a delay after the source signal becomes active. 1 The delay bypass is active. The dead-time counter DTC0 is not used by the sampling of the Hall pattern.
0	[31:16]	r	Reserved ; read as 0; should be written with 0.

Note: In the capture modes, all edges at the CC6x inputs are leading to the setting of the corresponding interrupt status flags in register IS. In order to monitor the selected capture events at the CCPOSx inputs in the multi-input capture modes, the CC6xST bits of the corresponding channel are set when detecting the selected event. The interrupt status bits and the CC6xST bits have to be reset by software.

Table 7-3 Double-Register Compare Modes

Description	
Double-Register Capture modes	
0100	The contents of T12 is stored in CC6nR after a rising edge and in CC6nSR after a falling edge on the input pin CC6n.
0101	The value stored in CC6nSR is copied to CC6nR after a rising edge on the input pin CC6n. The actual timer value of T12 is simultaneously stored in the shadow register CC6nSR. This feature is useful for time measurements between consecutive rising edges on pins CC6n. COUT6n is IO.
0110	The value stored in CC6nSR is copied to CC6nR after a falling edge on the input pin CC6n. The actual timer value of T12 is simultaneously stored in the shadow register CC6nSR. This feature is useful for time measurements between consecutive falling edges on pins CC6n. COUT6n is IO.
0111	The value stored in CC6nSR is copied to CC6nR after any edge on the input pin CC6n. The actual timer value of T12 is simultaneously stored in the shadow register CC6nSR. This feature is useful for time measurements between consecutive edges on pins CC6n. COUT6n is IO.

Table 7-4 Combined T12 Modes

Description	
Combined T12 Modes	
1000	Hall Sensor mode: Capture mode for channel 0, compare mode for channels 1 and 2. The contents of T12 is captured into CC60 at a valid hall event (which is a reference to the actual speed). CC61 can be used for a phase delay function between hall event and output switching. CC62 can act as a time-out trigger if the expected hall event comes too late. The value 1000 has to be programmed to MSEL0, MSEL1 and MSEL2 if the hall signals are used. In this mode, the contents of timer T12 is captured in CC60 and T12 is reset after the detection of a valid hall event. In order to avoid noise effects, the dead-time counter channel 0 is started after an edge has been detected at the hall inputs. When reaching the value of 000001, the hall inputs are sampled and the pattern comparison is done.
1001	Hysteresis-like control mode with dead time generation: The negative edge of the CCPOSx input signal is used to reset bit CC6nST. As a result, the output signals can be switched to passive state immediately and switch back to active state (with dead time) if the CCPOSx is high and the bit CC6nST is set by a compare event.

Table 7-5 Multi-Input Capture Modes

Description	
Multi-Input Capture Modes	
1010	The timer value of T12 is stored in CC6nR after a rising edge at the input pin CC6n. The timer value of T12 is stored in CC6nSR after a falling edge at the input pin CCPOSx.
1011	The timer value of T12 is stored in CC6nR after a falling edge at the input pin CC6n. The timer value of T12 is stored in CC6nSR after a rising edge at the input pin CCPOSx.
1100	The timer value of T12 is stored in CC6nR after a rising edge at the input pin CC6n. The timer value of T12 is stored in CC6nSR after a rising edge at the input pin CCPOSx.
1101	The timer value of T12 is stored in CC6nR after a falling edge at the input pin CC6n. The timer value of T12 is stored in CC6nSR after a falling edge at the input pin CCPOSx.
1110	The timer value of T12 is stored in CC6nR after any edge at the input pin CC6n. The timer value of T12 is stored in CC6nSR after any edge at the input pin CCPOSx.
1111	reserved (no capture or compare action)

7.2.5 Interrupt Control Registers

Register IS contains the individual interrupt request bits. This register can only be read, write actions have no impact on the contents of this register. The software can set or reset the bits individually by writing to the registers ISS (to set the bits) or to register ISR (to reset the bits).

IS

Capture/Compare Interrupt Status Register

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STR	IDLE	WHE	CHE	TRP S	TRP F	T13 PM	T13 CM	T12 PM	T12 OM	ICC 62F	ICC 62R	ICC 61F	ICC 61R	ICC 60F	ICC 60R
rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh

Field	Bits	Type	Description
ICC60R, ICC61R, ICC62R	0, 2, 4	rh	Capture, Compare-Match Rising Edge Flag In compare mode, a compare-match has been detected while T12 was counting up. In capture mode, a rising edge has been detected at the input CC6x (x=0, 1, 2). 0 The event has not yet occurred since this bit has been reset for the last time. 1 The event described above has been detected.
ICC60F, ICC61F, ICC62F	1, 3, 5	rh	Capture, Compare-Match Falling Edge Flag In compare mode, a compare-match has been detected while T12 was counting down. In capture mode, a falling edge has been detected at the input CC6x (x=0, 1, 2). 0 The event has not yet occurred since this bit has been reset for the last time. 1 The event described above has been detected.

Capture/Compare Unit 6 (CCU6)

Field	Bits	Type	Description
T12OM	6	rh	Timer T12 One-Match Flag 0 A timer T12 one-match (while counting down) has not yet been detected since this bit has been reset for the last time. 1 A timer T12 one-match (while counting down) has been detected.
T12PM	7	rh	Timer T12 Period-Match Flag 0 A timer T12 period-match (while counting up) has not yet been detected since this bit has been reset for the last time. 1 A timer T12 period-match (while counting up) has been detected.
T13CM	8	rh	Timer T13 Compare-Match Flag 0 A timer T13 compare-match has not yet been detected since this bit has been reset for the last time. 1 A timer T13 compare-match has been detected.
T13PM	9	rh	Timer T13 Period-Match Flag 0 A timer T13 period-match has not yet been detected since this bit has been reset for the last time. 1 A timer T13 period-match has been detected.
TRPF	10	rh	Trap Flag The trap flag TRPF will be set by hardware if $TRPPEN = 1$ and $\overline{CTRAP} = 0$ or by software. If $TRPM2 = 0$, bit TRPF is reset by hardware if the input $\overline{CTRAP}$ becomes inactive ($TRPPEN = 1$). If $TRPM2 = 1$, bit TRPF has to be reset by software in order to leave the trap state. 0 The trap condition has not been detected. 1 The trap condition has been detected (input $\overline{CTRAP}$ has been 0 or by software).
TRPS¹⁾	11	rh	Trap State 0 The trap state is not active. 1 The trap state is active. Bit TRPS is set while bit $TRPF = 1$. It is reset according to the mode selected in register TRPCTR.

Field	Bits	Type	Description
CHE²⁾	12	rh	Correct Hall Event 0 A transition to a correct (=expected) hall event has not yet been detected since this bit has been reset for the last time. 1 A transition to a correct (=expected) hall event has not yet been detected.
WHE³⁾	13	rh	Wrong Hall Event 0 A transition to a wrong hall event (not the expected one) has not yet been detected since this bit has been reset for the last time. 1 A transition to a wrong hall event (not the expected one) has been detected.
IDLE⁴⁾	14	rh	IDLE State This bit is set together with bit WHE (wrong hall event) and it has to be reset by software. 0 No action. 1 Bit field MCMP is cleared, the selected outputs are set to passive state.
STR	15	rh	Multi-Channel Mode Shadow Transfer Request This bit is set when a shadow transfer from MCMOUTS to MCMOUT takes places in multi-channel mode. 0 The shadow transfer has not yet taken place. 1 The shadow transfer has taken place.
0	[31:16]	r	Reserved ; read as 0; should be written with 0.

- ¹⁾ During the trap state, the selected outputs are set to the passive state. The logic level driven during the passive state is defined by the corresponding bit in register PSLR. Bit TRPS = 1 and TRPF = 0 can occur if the trap condition is no longer active but the selected synchronization has not yet taken place.
- ²⁾ On every valid hall edge the contents of CURH is compared with the pattern on pin CCPOSx and if equal bit CHE is set.
- ³⁾ On every valid hall edge the contents of EXPH is compared with the pattern on pin CCPOSx. If both compares (CURH and EXPH with CCPOSx) are not true, bit WHE (wrong hall event) is set.
- ⁴⁾ Bit field MCMP is hold to 0 by hardware as long as IDLE = 1.

Note: Not all bits in register IS can generate an interrupt. Other status bits have been added, which have a similar structure for their set and reset actions.

Note: The interrupt generation is independent from the value of the bits in register IS, e.g. the interrupt will be generated (if enabled) even if the corresponding bit is already set. The trigger for an interrupt generation is the detection of a set condition (by hardware or software) for the corresponding bit in register IS.

Capture/Compare Unit 6 (CCU6)

Note: In compare mode (and hall mode), the timer-related interrupts are only generated while the timer is running (TxR=1). In capture mode, the capture interrupts are also generated while the timer T12 is stopped.

Register ISS contains the individual interrupt request set bits to generate a CCU6 interrupt request by software.

ISS

Capture/Compare Interrupt Status Set Register

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
S STR	S IDLE	S WHE	S CHE	S WHC	S TRP F	S T13 PM	S T13 CM	S T12 PM	S T12 OM	S CC 62F	S CC 62R	S CC 61F	S CC 61R	S CC 60F	S CC 60R
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Field	Bits	Type	Description
SCC60R	0	w	Set Capture, Compare-Match Rising Edge Flag 0 No action 1 Bit CC60R in register IS will be set.
SCC60F	1	w	Set Capture, Compare-Match Falling Edge Flag 0 No action 1 Bit CC60F in register IS will be set.
SCC61R	2	w	Set Capture, Compare-Match Rising Edge Flag 0 No action 1 Bit CC61R in register IS will be set.
SCC61F	3	w	Set Capture, Compare-Match Falling Edge Flag 0 No action 1 Bit CC61F in register IS will be set.
SCC62R	4	w	Set Capture, Compare-Match Rising Edge Flag 0 No action 1 Bit CC62R in register IS will be set.
SCC62F	5	w	Set Capture, Compare-Match Falling Edge Flag 0 No action 1 Bit CC62F in register IS will be set.

Capture/Compare Unit 6 (CCU6)

Field	Bits	Type	Description
ST12OM	6	w	Set Timer T12 One-Match Flag 0 No action 1 Bit T12OM in register IS will be set.
ST12PM	7	w	Set Timer T12 Period-Match Flag 0 No action 1 Bit T12PM in register IS will be set.
ST13CM	8	w	Set Timer T13 Compare-Match Flag 0 No action 1 Bit T13CM in register IS will be set.
ST13PM	9	w	Set Timer T13 Period-Match Flag 0 No action 1 Bit T13PM in register IS will be set.
STRPF	10	w	Set Trap Flag 0 No action 1 Bits TRPF and TRPS in register IS will be set.
SWHC	11	w	Software Hall Compare 0 No action 1 The Hall compare action is triggered.
SCHE	12	w	Set Correct Hall Event Flag 0 No action 1 Bit CHE in register IS will be set.
SWHE	13	w	Set Wrong Hall Event Flag 0 No action 1 Bit WHE in register IS will be set.
SIDLE	14	w	Set IDLE Flag 0 No action 1 Bit IDLE in register IS will be set.
SSTR	15	w	Set STR Flag 0 No action 1 Bit STR in register IS will be set.
0	[31:16]	r	Reserved ; read as 0; should be written with 0.

Note: If the setting by hardware of the corresponding flags can lead to an interrupt, the setting by software has the same effect.

Capture/Compare Unit 6 (CCU6)

Register ISR contains the individual interrupt request reset the corresponding flags by software.

ISR

Capture/Compare Interrupt Status Reset Register

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R STR	R IDLE	R WHE	R CHE	0	R TRP F	R T13 PM	R T13 CM	R T12 PM	R T12 OM	R CC 62F	R CC 62R	R CC 61F	R CC 61R	R CC 60F	R CC 60R
w	w	w	w	r	w	w	w	w	w	w	w	w	w	w	w

Field	Bits	Type	Description
RCC60R	0	w	Reset Capture, Compare-Match Rising Edge Flag 0 No action 1 Bit CC60R in register IS will be reset.
RCC60F	1	w	Reset Capture, Compare-Match Falling Edge Flag 0 No action 1 Bit CC60F in register IS will be reset.
RCC61R	2	w	Reset Capture, Compare-Match Rising Edge Flag 0 No action 1 Bit CC61R in register IS will be reset.
RCC61F	3	w	Reset Capture, Compare-Match Falling Edge Flag 0 No action 1 Bit CC61F in register IS will be reset.
RCC62R	4	w	Reset Capture, Compare-Match Rising Edge Flag 0 No action 1 Bit CC62R in register IS will be reset.
RCC62F	5	w	Reset Capture, Compare-Match Falling Edge Flag 0 No action 1 Bit CC62F in register IS will be reset.
RT12OM	6	w	Reset Timer T12 One-Match Flag 0 No action 1 Bit T12OM in register IS will be reset.

Capture/Compare Unit 6 (CCU6)

Field	Bits	Type	Description
RT12PM	7	w	Reset Timer T12 Period-Match Flag 0 No action 1 Bit T12PM in register IS will be reset.
RT13CM	8	w	Reset Timer T13 Compare-Match Flag 0 No action 1 Bit T13CM in register IS will be reset.
RT13PM	9	w	Reset Timer T13 Period-Match Flag 0 No action 1 Bit T13PM in register IS will be reset.
RTRPF	10	w	Reset Trap Flag 0 No action 1 Bit TRPF in register IS will be reset (not taken into account while input $\overline{\text{CTRAP}}=0$ and $\text{TRPEN} = 1$).
RCHE	12	w	Reset Correct Hall Event Flag 0 No action 1 Bit CHE in register IS will be reset.
RWHE	13	w	Reset Wrong Hall Event Flag 0 No action 1 Bit WHE in register IS will be reset.
RIDLE	14	w	Reset IDLE Flag 0 No action 1 Bit IDLE in register IS will be reset.
RSTR	15	w	Reset STR Flag 0 No action 1 Bit STR in register IS will be reset.
0	11, [31:16]	r	Reserved ; read as 0; should be written with 0.

Capture/Compare Unit 6 (CCU6)

Register IEN contains the interrupt enable bits and a control bit to enable the automatic idle function in the case of a wrong hall pattern.

IEN

Capture/Compare Interrupt Enable Register

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EN STR	EN IDLE	EN WHE	EN CHE	0	EN TRP F	EN T13 PM	EN T13 CM	EN T12 PM	EN T12 OM	EN CC 62F	EN CC 62R	EN CC 61F	EN CC 61R	EN CC 60F	EN CC 60R
rw	rw	rw	rw	r	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Field	Bits	Type	Description
ENCC60R	0	rw	Capture, Compare-Match Rising Edge Interrupt Enable for Channel 0 0 No interrupt will be generated if the set condition for bit CC60R in register IS occurs. 1 An interrupt will be generated if the set condition for bit CC60R in register IS occurs. The interrupt line, which will be activated, is selected by bit field INPCC60.
ENCC60F	1	rw	Capture, Compare-Match Falling Edge Interrupt Enable for Channel 0 0 No interrupt will be generated if the set condition for bit CC60F in register IS occurs. 1 An interrupt will be generated if the set condition for bit CC60F in register IS occurs. The interrupt line, which will be activated, is selected by bit field INPCC60.
ENCC61R	2	rw	Capture, Compare-Match Rising Edge Interrupt Enable for Channel 1 0 No interrupt will be generated if the set condition for bit CC61R in register IS occurs. 1 An interrupt will be generated if the set condition for bit CC61R in register IS occurs. The interrupt line, which will be activated, is selected by bit field INPCC61.

Capture/Compare Unit 6 (CCU6)

Field	Bits	Type	Description
ENCC61F	3	rw	Capture, Compare-Match Falling Edge Interrupt Enable for Channel 1 0 No interrupt will be generated if the set condition for bit CC61F in register IS occurs. 1 An interrupt will be generated if the set condition for bit CC61F in register IS occurs. The interrupt line, which will be activated, is selected by bit field INPCC61.
ENCC62R	4	rw	Capture, Compare-Match Rising Edge Interrupt Enable for Channel 2 0 No interrupt will be generated if the set condition for bit CC62R in register IS occurs. 1 An interrupt will be generated if the set condition for bit CC62R in register IS occurs. The interrupt line, which will be activated, is selected by bit field INPCC62.
ENCC62F	5	rw	Capture, Compare-Match Falling Edge Interrupt Enable for Channel 2 0 No interrupt will be generated if the set condition for bit CC62F in register IS occurs. 1 An interrupt will be generated if the set condition for bit CC62F in register IS occurs. The interrupt line, which will be activated, is selected by bit field INPCC62.
ENT12OM	6	rw	Enable Interrupt for T12 One-Match 0 No interrupt will be generated if the set condition for bit T12OM in register IS occurs. 1 An interrupt will be generated if the set condition for bit T12OM in register IS occurs. The interrupt line, which will be activated, is selected by bit field INPT12.
ENT12PM	7	rw	Enable Interrupt for T12 Period-Match 0 No interrupt will be generated if the set condition for bit T12PM in register IS occurs. 1 An interrupt will be generated if the set condition for bit T12PM in register IS occurs. The interrupt line, which will be activated, is selected by bit field INPT12.

Capture/Compare Unit 6 (CCU6)

Field	Bits	Type	Description
ENT13CM	8	rw	Enable Interrupt for T13 Compare-Match 0 No interrupt will be generated if the set condition for bit T13CM in register IS occurs. 1 An interrupt will be generated if the set condition for bit T13CM in register IS occurs. The interrupt line, which will be activated, is selected by bit field INPT13.
ENT13PM	9	rw	Enable Interrupt for T13 Period-Match 0 No interrupt will be generated if the set condition for bit T13PM in register IS occurs. 1 An interrupt will be generated if the set condition for bit T13PM in register IS occurs. The interrupt line, which will be activated, is selected by bit field INPT13.
ENTRPF	10	rw	Enable Interrupt for Trap Flag 0 No interrupt will be generated if the set condition for bit TRPF in register IS occurs. 1 An interrupt will be generated if the set condition for bit TRPF in register IS occurs. The interrupt line, which will be activated, is selected by bit field INPERR.
ENCHE	12	rw	Enable Interrupt for Correct Hall Event 0 No interrupt will be generated if the set condition for bit CHE in register IS occurs. 1 An interrupt will be generated if the set condition for bit CHE in register IS occurs. The interrupt line, which will be activated, is selected by bit field INPCHE.
ENWHE	13	rw	Enable Interrupt for Wrong Hall Event 0 No interrupt will be generated if the set condition for bit WHE in register IS occurs. 1 An interrupt will be generated if the set condition for bit WHE in register IS occurs. The interrupt line, which will be activated, is selected by bit field INPERR.

Capture/Compare Unit 6 (CCU6)

Field	Bits	Type	Description
ENIDLE	14	rw	Enable Idle This bit enables the automatic entering of the idle state (bit IDLE will be set) after a wrong hall event has been detected (bit WHE is set). During the idle state, the bit field MCMP is automatically cleared. 0 The bit IDLE is not automatically set when a wrong hall event is detected. 1 The bit IDLE is automatically set when a wrong hall event is detected.
ENSTR	15	rw	Enable Multi-Channel Mode Shadow Transfer Interrupt 0 No interrupt will be generated if the set condition for bit STR in register IS occurs. 1 An interrupt will be generated if the set condition for bit STR in register IS occurs. The interrupt line, which will be activated, is selected by bit field INPCHE.
0	11, [31:16]	r	Reserved ; read as 0; should be written with 0.

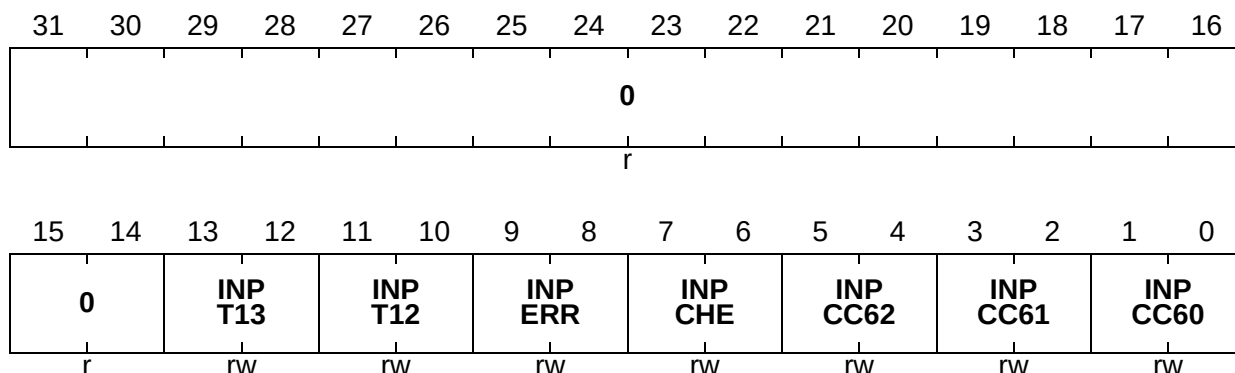
Capture/Compare Unit 6 (CCU6)

Register INP contains the interrupt node pointer bits, allowing for flexible interrupt handling.

INP

Capture/Compare Interrupt Node Pointer Register

Reset Value: 0000 3940_H



Field	Bits	Type	Description
INPCC60	[1:0]	rw	Interrupt Node Pointer for Channel 0 Interrupts This bit field defines the interrupt output line, which is activated due to a set condition for bit CC60R (if enabled by bit ENCC60R) or for bit CC60F (if enabled by bit ENCC60F). 00 Interrupt output line SR0 is selected. 01 Interrupt output line SR1 is selected. 10 Interrupt output line SR2 is selected. 11 Interrupt output line SR3 is selected.
INPCC61	[3:2]	rw	Interrupt Node Pointer for Channel 1 Interrupts This bit field defines the interrupt output line, which is activated due to a set condition for bit CC61R (if enabled by bit ENCC61R) or for bit CC61F (if enabled by bit ENCC61F). 00 Interrupt output line SR0 is selected. 01 Interrupt output line SR1 is selected. 10 Interrupt output line SR2 is selected. 11 Interrupt output line SR3 is selected.

Capture/Compare Unit 6 (CCU6)

Field	Bits	Type	Description
INPCC62	[5:4]	rw	Interrupt Node Pointer for Channel 2 Interrupts This bit field defines the interrupt output line, which is activated due to a set condition for bit CC62R (if enabled by bit ENCC62R) or for bit CC62F (if enabled by bit ENCC62F). 00 Interrupt output line SR0 is selected. 01 Interrupt output line SR1 is selected. 10 Interrupt output line SR2 is selected. 11 Interrupt output line SR3 is selected.
INPCHE	[7:6]	rw	Interrupt Node Pointer for the CHE Interrupt This bit field defines the interrupt output line, which is activated due to a set condition for bit CHE (if enabled by bit ENCHE) or for bit STR (if enabled by bit ENSTR). 00 Interrupt output line SR0 is selected. 01 Interrupt output line SR1 is selected. 10 Interrupt output line SR2 is selected. 11 Interrupt output line SR3 is selected.
INPERR	[9:8]	rw	Interrupt Node Pointer for Error Interrupts This bit field defines the interrupt output line, which is activated due to a set condition for bit TRPF (if enabled by bit ENTRPF) or for bit WHE (if enabled by bit ENWHE). 00 Interrupt output line SR0 is selected. 01 Interrupt output line SR1 is selected. 10 Interrupt output line SR2 is selected. 11 Interrupt output line SR3 is selected.
INPT12	[11:10]	rw	Interrupt Node Pointer for Timer12 Interrupts This bit field defines the interrupt output line, which is activated due to a set condition for bit T12OM (if enabled by bit ENT12OM) or for bit T12PM (if enabled by bit ENT12PM). 00 Interrupt output line SR0 is selected. 01 Interrupt output line SR1 is selected. 10 Interrupt output line SR2 is selected. 11 Interrupt output line SR3 is selected.

Capture/Compare Unit 6 (CCU6)

Field	Bits	Type	Description
INPT13	[13:12]	rw	Interrupt Node Pointer for Timer13 Interrupt This bit field defines the interrupt output line, which is activated due to a set condition for bit T13CM (if enabled by bit ENT13CM) or for bit T13PM (if enabled by bit ENT13PM). 00 Interrupt output line SR0 is selected. 01 Interrupt output line SR1 is selected. 10 Interrupt output line SR2 is selected. 11 Interrupt output line SR3 is selected.
0	[31:14]	r	Reserved ; read as 0; should be written with 0.

7.3 CCU61 Module Implementation

This section describes CCU61 module interface with the clock control, port connections, interrupt control, and address decoding.

7.3.1 Interface of the CCU6 Module

Figure 7-32 shows the TC1100 specific implementation details and interconnections of the CCU61 module. The CCU6 module is further supplied by clock control, interrupt control, address decoding, and port control logic. One DMA request can be generated by the CCU6 module.

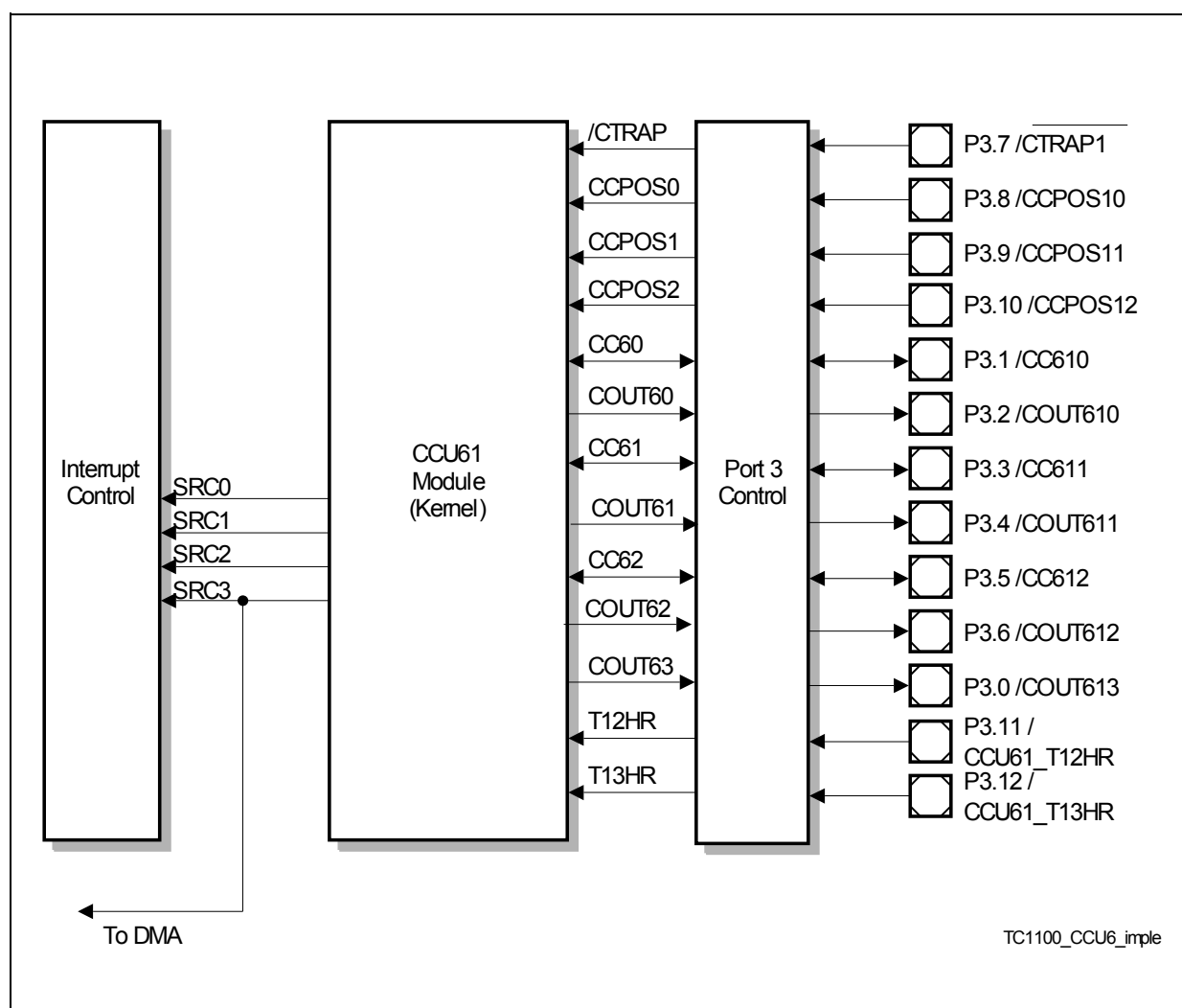


Figure 7-32 CCU6 Module Implementation and Interconnections

7.3.2 CCU61 Module Related External Registers

Figure 7-33 summarizes the module related external registers which are required for CCU61 programming (see also **Figure 7-31** for the module kernel specific registers).

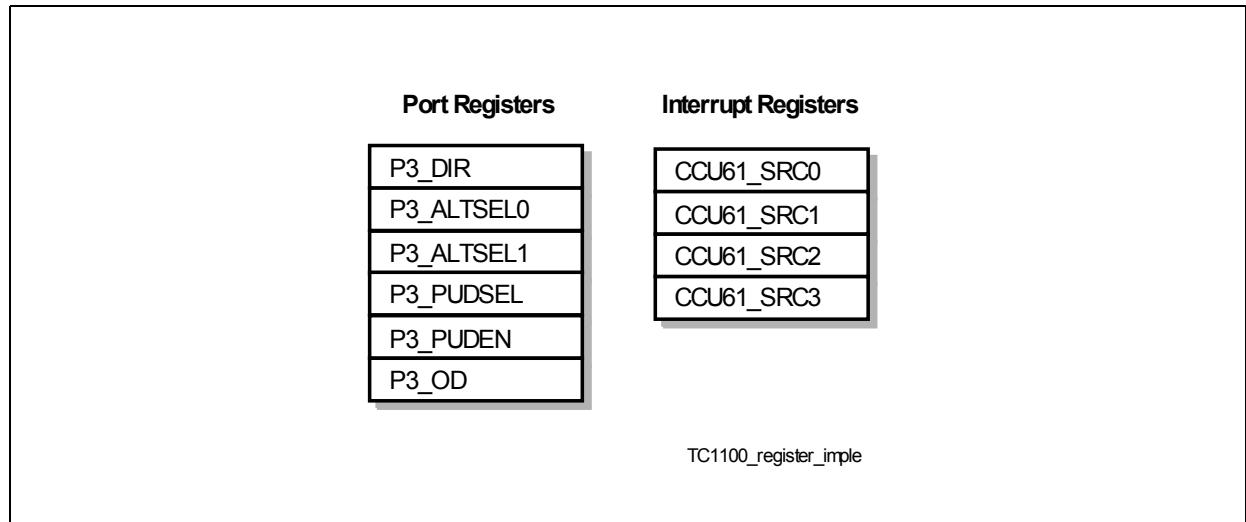


Figure 7-33 CCU61 Implementation Specific Special Function Registers

7.3.2.1 Clock Control

The CCU6 module is provided with two clock signals:

- f_{CLC}
This is the module clock that is used inside the CCU6 kernel for control purposes such as e.g. for clocking of control logic and register operations. The frequency of f_{CLC} is always identical to the system clock frequency f_{SYS} . The clock control register CCU60_CLC makes it possible to enable/disable f_{CLC} under certain conditions.
- f_{CCU}
This is the module clock that is used in the CCU6 module as input of the prescalers for T12 and T13. The fractional divider register CCU60_FDR controls the frequency of f_{CCU} and allows it to be enabled/disabled independently of f_{CLC} .

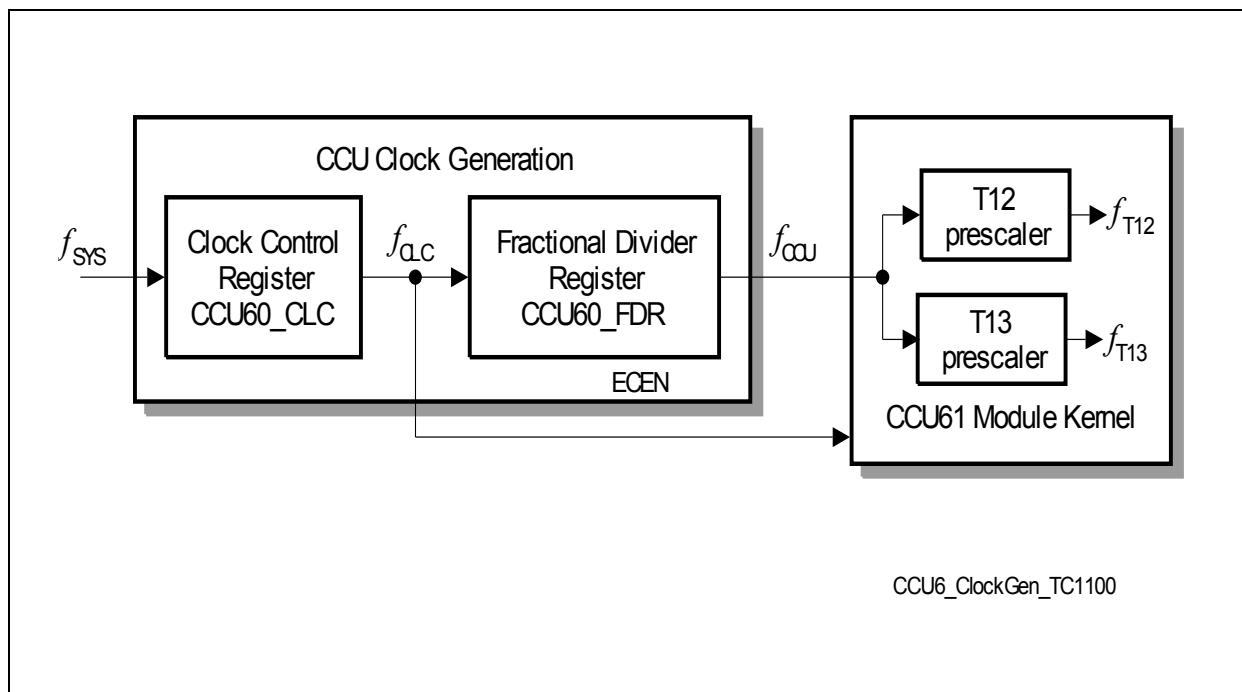


Figure 7-34 CCU6 Clock Generation

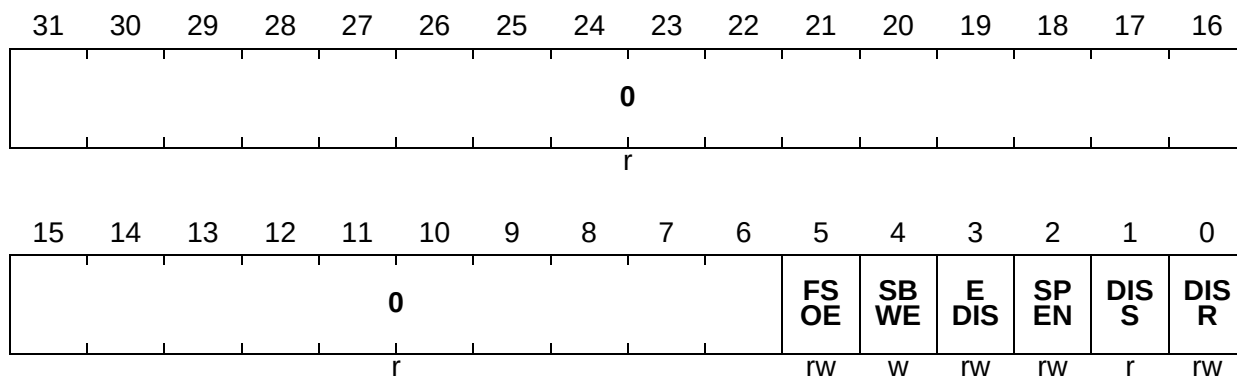
7.3.2.2 Clock Control Register

The common clock control register allows the programmer to control (enable/disable) the clock signals to the CCU6 module under certain conditions.

CCU60_CLC

CCU60 Clock Control Register

Reset Value: 0000 0003_H



Field	Bits	Type	Description
DISR	0	rw	Module Disable Request Bit Used for enable/disable control of the module.
DISS	1	r	Module Disable Status Bit Bit indicates the current status of the module.
SPEN	2	rw	Module Suspend Enable for OCDS Used for enabling the suspend mode.
EDIS	3	rw	External Request Disable Used for controlling the external clock disable request.
SBWE	4	w	Module Suspend Bit Write Enable for OCDS Defines whether SPEN and FSOE are write protected.
FSOE	5	rw	Fast Switch Off Enable Used for fast clock switch off in OCDS suspend mode.
0	[31:6]	r	Reserved ; read as 0; should be written with 0.

7.3.2.3 Fractional Divider Register

The common fractional divider register controls the clock rate of the shift clock f_{CCU} .

CCU60_FDR

CCU60 Fractional Divider Register

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DIS CLK	EN HW	SUS REQ	SUS ACK	0	RESULT										
rwh	rw	rh	rh	r	rh										
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DM		SC		0	STEP										
rw		rw		r	rw										

Field	Bits	Type	Description
STEP	[9:0]	rw	Step Value Reload or addition value for RESULT.
SC	[13:12]	rw	Suspend Control This bit field defines the behavior of the fractional divider in suspend mode.
DM	[15:14]	rw	Divider Mode This bit field selects normal divider mode or fractional divider mode.
RESULT	[25:16]	rh	Result Value Bit fields for the addition result.
SUSACK	28	rh	Suspend Mode Acknowledge Indicates state of SPNDACK signal.
SUSREQ	29	rh	Suspend Mode Request Indicates state of SPND signal.
ENHW	30	rw	Enable Hardware Clock Control Controls operation of ECEN input and DISCLK bit.
DISCLK	31	rwh	Disable Clock Hardware controlled disable for f_{OUT} signal.
0	[11:10], [27:26]	rw	Reserved ; read as 0; should be written with 0

7.3.3 Port Control

The interconnections between the CCU6 module and the I/O lines/pins are controlled by software in the port logics. The CCU61 I/O functionality must be selected by the following port control operations (additionally to the PISEL programming):

- Input/output function selection (DIR registers)
- Alternate function selection (ALTSEL0 and ALTSEL1 registers)
- Input/Output driver characteristic control (PUDSEL, PUDEN and OD registers)

The CCU61 port input/output control registers contain the bit fields that select the digital output and input driver characteristics such as pull-up/down devices, port direction (input/output), open-drain, and alternate output selections. The I/O lines for the CCU61 module are controlled by the port input/output control registers of Port 0, Port 2 and Port 3.

Table 7-6 shows how bits and bit fields must be programmed for the required I/O functionality of the CCU6 I/O lines.

Table 7-6 CCU61 I/O Line Selection and Setup

Module	Port Lines	Input/Output Control Register Bits	I/O
CCU61	P3.0/COOUT613	P3_DIR.P0 = 1 _B	Output
		P3_ALTSEL0.P0 = 1 _B	
		P3_ALTSEL1.P0 = 0 _B	
	P3.1/CC610	P3_DIR.P1 = 0 _B	Input
		P3_DIR.P1 = 1 _B	Output
		P3_ALTSEL0.P1 = 1 _B	
		P3_ALTSEL1.P1 = 0 _B	
	P3.2/COOUT610	P3_DIR.P2 = 1 _B	Output
		P3_ALTSEL0.P2 = 1 _B	
		P3_ALTSEL1.P2 = 0 _B	
	P3.3/CC611	P3_DIR.P3 = 0 _B	Input
		P3_DIR.P3 = 1 _B	Output
		P3_ALTSEL0.P3 = 1 _B	
		P3_ALTSEL1.P3 = 0 _B	
	P3.4/COOUT611	P3_DIR.P4 = 1 _B	Output
		P3_ALTSEL0.P4 = 1 _B	
		P3_ALTSEL1.P4 = 0 _B	
	P3.5/CC612	P3_DIR.P5 = 0 _B	Input
		P3_DIR.P5 = 1 _B	Output
		P3_ALTSEL0.P5 = 1 _B	
		P3_ALTSEL1.P5 = 0 _B	
	P3.6/COOUT612	P3_DIR.P6 = 1 _B	Output
		P3_ALTSEL0.P6 = 1 _B	
		P3_ALTSEL1.P6 = 0 _B	
	P3.7/CTRAP1	P3_DIR.P7 = 0 _B	Input
	P3.8/CCPOS10	P3_DIR.P8 = 0 _B	Input
	P3.9/CCPOS11	P3_DIR.P9 = 0 _B	Input
	P3.10/CCPOS12	P3_DIR.P10 = 0 _B	Input

Table 7-6 CCU61 I/O Line Selection and Setup (cont'd)

Module	Port Lines	Input/Output Control Register Bits	I/O
	P3.11/ CC61_T12HR	P3_DIR.P11 = 0 _B	Input
	P3.12/ CC62_T13HR	P3_DIR.P12 = 0 _B	Input

P3_DIR

Port 3 Direction Register

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
P15	P14	P13	P12	P11	P10	P9	P8	P7	P6	P5	P4	P3	P2	P1	P0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Field	Bits	Type	Description
Pn (n = 0-12)	n	rw	Port 3 Pin 0-12 Direction Control ¹⁾ 0 Direction is set to input (default after reset) 1 Direction is set to output
0	[31:16]	r	Reserved ; read as 0; should be written with 0.

¹⁾ Shaded bits and bit field are don't care for CCU61 I/O port control

P3_ALTSELn (n = 1, 0)

Port 3 Alternate Select Register

Reset Value: 0000 0000_H

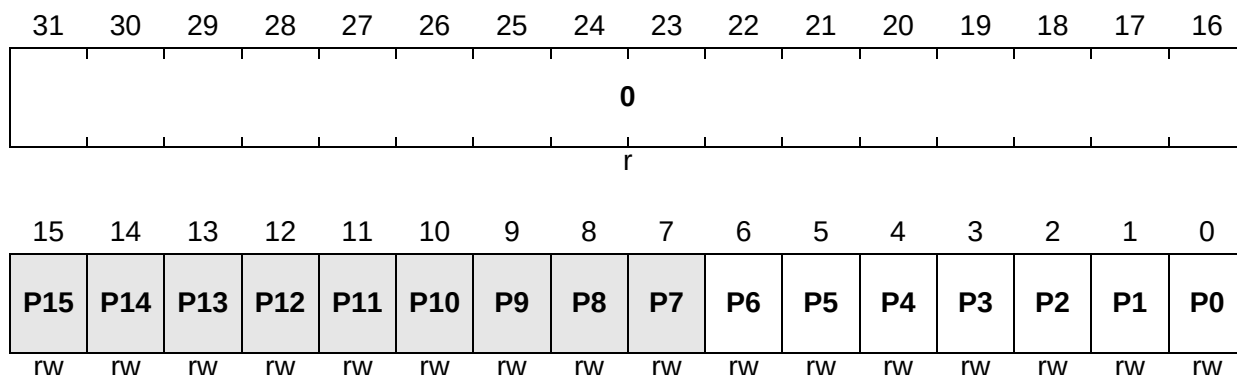


Table 7-7 Function of the Bits P3_ALTSEL0.Pn and P3_ALTSEL1.Pn (n = 0-6)¹⁾

P3_ALTSEL0.Pn	P3_ALTSEL1.Pn	Function
1	0	Alternate Select 1

¹⁾ Shaded bits and bit field are don't care for CCU61 I/O port control

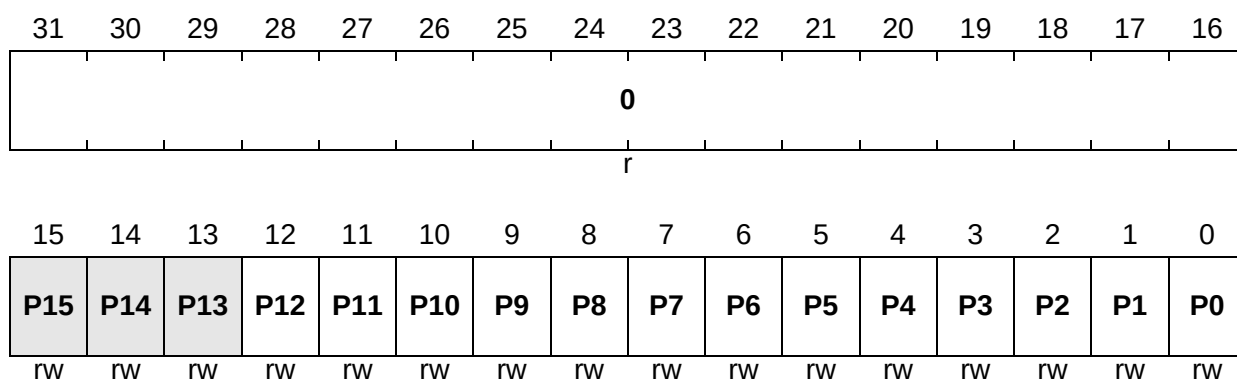
The CCU61 port also offer the possibility to configure the following output characteristics:

- push/pull (optional pull-up/pull-down)
- open drain with internal pull-up
- open drain with external pull-up

P3_PUDSEL

Port 3 Pull-Up/Pull-Down Select Register

Reset Value: 0000 FFFF_H



Capture/Compare Unit 6 (CCU6)

Field	Bits	Type	Description
Pn (n = 0-12)	n	rw	Pull-Up/Pull-Down Select Port 3 Bit n¹⁾ 0 Pull-down device is selected 1 Pull-up device is selected
0	[31:16]	r	Reserved ; read as 0; should be written with 0.

¹⁾ Shaded bits and bit field are don't care for CCU61 I/O port control

P3_PUDEN

Port 3 Pull-Up/Pull-Down Enable Register

Reset Value: 0000 FFFF_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
P15	P14	P13	P12	P11	P10	P9	P8	P7	P6	P5	P4	P3	P2	P1	P0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Field	Bits	Type	Description
Pn (n = 0-12)	n	rw	Pull-Up/Pull-Down Enable at Port 3 Bit n¹⁾ 0 Pull-up or Pull-down device is disabled 1 Pull-up or Pull-down device is enabled
0	[31:16]	r	Reserved ; read as 0; should be written with 0.

¹⁾ Shaded bits and bit field are don't care for CCU61 I/O port control

P3_OD

Port 3 Open Drain Control Register

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
P15	P14	P13	P12	P11	P10	P9	P8	P7	P6	P5	P4	P3	P2	P1	P0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Capture/Compare Unit 6 (CCU6)

Field	Bits	Type	Description
Pn (n = 0-6)	n	rw	Port 3 Pin n Open Drain Mode¹⁾ 0 Normal Mode, output is actively driven for 0 and 1 state 1 Open Drain Mode, output is actively driven only for 0 state
0	[31:16]	r	Reserved ; read as 0; should be written with 0.

¹⁾ Shaded bits and bit field are don't care for CCU6 I/O port control

7.3.3.1 Service Request Registers

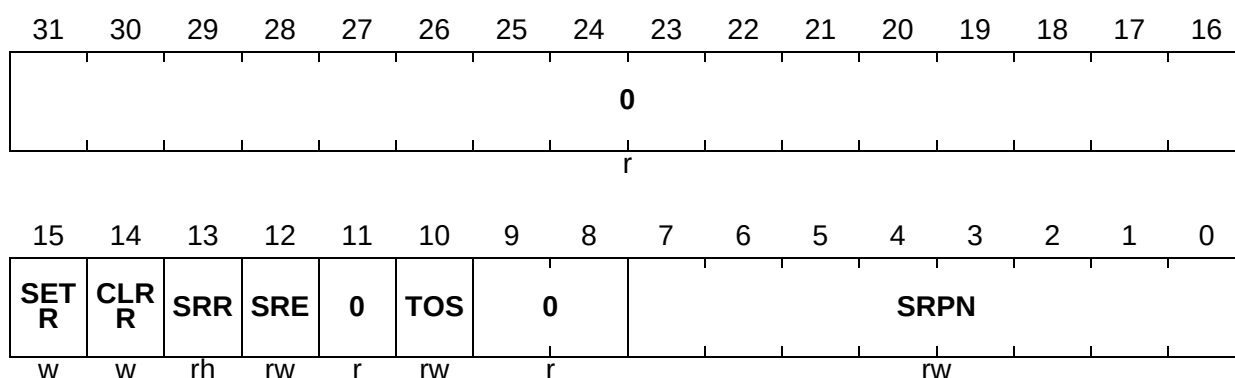
The CCU6 module has four service request outputs. The interrupt output lines SRC[3:0] of the CCU6 module are connected to service request nodes. The service request control registers are described as below.

The interrupt output lines SRC[3] of the CCU6 module are also connected to the DMA module. The request assignment is described in the DMA implementation chapter.

CCU61_SRCx (x = 0-3)

CCU61 Service Request Control Register x

Reset Value: 0000 0000_H



Field	Bits	Type	Description
SRPN	[7:0]	rw	Service Request Priority Number
TOS	10	rw	Type of Service Control
SRE	12	rw	Service Request Enable
SRR	13	rh	Service Request Flag
CLRR	14	w	Request Clear Bit
SETR	15	w	Request Set Bit
0	[9:8], 11 [31:16]	r	Reserved; read as 0; should be written with 0.

Note: Further details on interrupt handling and processing are described in chapter "Interrupt System" of the TC1100 System Units User's Manual. The detailed DMA request connections are defined in chapter "Direct Memory Access Controller" of the TC1100 System Units User's Manual.

7.3.4 DMA Requests

The DMA request lines of the CCU61 module become active whenever its related interrupt line is activated. The DMA request lines are connected to the DMA controller as shown in [Table 7-8](#).

Table 7-8 DMA Request Lines of CCU61

Module	Related SSC Interrupt	DMA Request Line	Description
CCU61	SRC3	CCU61_SRC3	CCU61 DMA Request

7.3.5 CCU61 Register Address Ranges

In the TC1100, the registers of the CCU6 module are located in the following address ranges:

- CCU61 module: Module Base Address = F000 2100_H
 Module End Address = F000 21FF_H
- Absolute Register Address = Module Base Address + Offset Address
 (offset addresses see [Table 7-2](#))

Note: The complete and detailed address map of the CCU61 module is described in the chapter “Register Overview” of the TC1100 System Units User’s Manual.

8 Index

8.1 Keyword Index

This section lists a number of keywords which refer to specific details of the TC1100 in terms of its architecture, its functional units, or functions. Bold page number entries identify the main definition material for a topic.

A

ASC

- Address ranges 2-60
- Asynchronous mode 2-5–2-17
 - Data frames 2-6–2-7
 - Data path selection 2-17
- Baud rate generation 2-20–2-25
 - Asynchronous modes 2-21
 - Synchronous mode 2-24
- Block diagram
 - Asynchronous modes 2-5
 - Synchronous mode 2-18
- DMA request outputs 2-60
- Error detection 2-25
- Features 2-3
- Interrupt generation 2-27
- IrDA Mode
 - Function 2-15
 - IrDA frames 2-8
- Module implementation 2-45–2-60
 - DMA request outputs 2-60
 - Input/output function selection 2-50
 - Interrupt registers 2-58
 - Module clock control 2-46
 - Peripheral input select 2-48
- Registers 2-29–2-43
 - Address ranges 2-60
 - BG **2-34**
 - CON **2-30**
 - FDV **2-35**
 - FSTAT **2-43**
 - Offset addresses 2-29

Overview 2-29

PISEL **2-30**

PMW **2-36**

RBUF **2-37**

RXFCON **2-39**

TBUF **2-36**

TXFCON **2-41**

WHBCON **2-33**

Synchronous mode 2-18–2-19

Timings 2-20

C

Capture/Compare Unit 6 1-19

CCU6

- DMA request outputs 7-103
- Module implementation
 - DMA request outputs 7-103
 - Module clock control 7-93
- Registers
 - Address ranges 7-103
 - CC63R **7-59**
 - CC63SR **7-60**
 - CC6xR **7-53**
 - CC6xSR **7-54**
 - CMPMODIF **7-42**
 - CMPSTAT **7-40**
 - IEN **7-84**
 - INP **7-88**
 - IS **7-77**
 - ISR **7-82**
 - ISS **7-80**
 - MCMCTR **7-71**
 - MCMOUT **7-68**

MCMOUTS **7-67**
MODCTR **7-61**
PISEL **7-38, 7-40**
PSLR **7-65**
T12 **7-51**
T12DTC **7-55**
T12MSEL **7-73**
T12PR **7-52**
T13 **7-57**
T13PR **7-58**
TCTR0 **7-43**
TCTR2 **7-46**
TCTR4 **7-49**
TRPCTR **7-63**

D

Document

Structure 1-1
Terminology 1-2
Textual conventions 1-1

G

GPTU

Block diagram 6-2
Features 6-3
Interrupt generation 6-20
Module implementation 6-52–6-62
 Input/output function selection 6-55
Output control 6-18
Registers 6-22–6-50
 Address range 6-62
 Offset addresses 6-22
 OSEL **6-47**
 OUT **6-48**
 Overview 6-22
 SRSEL **6-49**
 T012RUN **6-41**
 T01IRS **6-24**
 T01OTS **6-27**
 T0CBA **6-29**
 T0DCBA **6-29**
 T0RCBA **6-30**
 T0RDCBA **6-30**

T1CBA **6-31**
T1DCBA **6-31**
T1RCBA **6-32**
T1RDCBA **6-31**
T2 **6-45**
T2AIS **6-33**
T2BIS **6-35**
T2CON **6-38**
T2ES **6-36**
T2RC0 **6-46**
T2RC1 **6-46**
T2RCCON **6-43**

I

IIC

Address range 4-32
DMA request outputs 4-32
Module implementation 4-25–4-32
 DMA request outputs 4-32
 Input/output function selection 4-28
 Module clock control 4-27

Registers

Address range 4-32
BUSCON **4-23**
CLC **4-27**
Offset addresses 4-12
Overview 4-12
PISEL **4-13**
RTB **4-24**
SYSCON **4-14**
WHBSYSCON **4-20**

M

MLI

Applications 5-2
Communication principles 5-6
DMA requests 5-103
General description 5-7
Handshake timing 5-13
Interrupts 5-46
Naming conventions 5-5
Reading process 5-45
Receiver 5-34

Error handling 5-42
I/O control 5-43
Operation modes 5-34
Registers
 Address ranges 5-114
 AER 5-94
 ARR 5-95
 GINTR 5-93
 Offset addresses 5-59
 OICR 5-80
 Overview 5-58
 RADDR 5-78
 RCR 5-73
 RDATAR 5-78
 RIER 5-89
 RINPR 5-92
 RISR 5-91
 RPxBAR 5-76
 RPxSTATR 5-77
 SCR 5-79
 TCBAR 5-72
 TCMDR 5-66
 TCR 5-61
 TDRAR 5-71
 TIER 5-85
 TINPR 5-87
 TISR 5-86
 TPxAOFR 5-70
 TPxBAR 5-71
 TPxDATAR 5-70
 TPxSTATR 5-65
 TRSTATR 5-68
 TSTATR 5-63
Startup 5-15
Timings 5-48
Transaction flow diagrams
 Copy base address 5-53
Transmitter 5-18
 Errors 5-30
 I/O control 5-31
 Operation modes 5-18
 Parity 5-30
 Transfer mode selection 5-27

Transmission format 5-20
Transmission modes 5-21

S

Serial interfaces 1-8
 Async./sync. serial interface 1-8
 High-speed sync. serial interface 1-11
 IIC 1-13
 MLI 1-15
SSC
 Address ranges 3-73
 Baud rate generation 3-19, 3-25
 Block diagram 3-5
 Chip select generation 3-23
 DMA request outputs 3-73
 Error detection 3-26
 FIFO operation
 Receive FIFO 3-15
 Transmit FIFO 3-13
 Transparent Mode 3-17
 Full-duplex operation 3-7
 Half-duplex operation 3-10
 Interrupts 3-26
 Module implementation 3-45–3-73
 DMA request outputs 3-73
 Interrupt registers 3-72
 Module clock control 3-48
 Port input select 3-52
 Registers 3-28–3-39
 Address ranges 3-73
 BR 3-38
 CON 3-31
 EFM 3-34
 FSTAT 3-44
 Offset addresses 3-28
 Overview 3-28
 PISEL 3-29, 3-54
 RB 3-39
 RXFCON 3-40
 SSOC 3-36
 SSOTC 3-37
 STAT 3-33
 TB 3-39

TXFCON **3-42**
Slave select input operation 3-21

T

Timer units
General purpose timer unit 1-17

8.2 Register Index

This section lists the references to the Special Function Registers of the TC1100.

A

ASC module registers 2-29

ASC0_BG 2-34

ASC0_CLC 2-47

ASC0_CON 2-30

ASC0_ESRC 2-58

ASC0_FDV 2-35

ASC0_FSTAT 2-43

ASC0_PISEL 2-30

ASC0_PMW 2-36

ASC0_RBUF 2-37

ASC0_RSRC 2-58

ASC0_RXFCON 2-39

ASC0_TBSRC 2-58

ASC0_TBUF 2-36

ASC0_TSRC 2-58

ASC0_TXFCON 2-41

ASC0_WHBCON 2-33

ASC1_BG 2-34

ASC1_CLC 2-47

ASC1_CON 2-30

ASC1_ESRC 2-58

ASC1_FDV 2-35

ASC1_FSTAT 2-43

ASC1_PISEL 2-30, 2-49

ASC1_PMW 2-36

ASC1_RBUF 2-37

ASC1_RSRC 2-58

ASC1_RXFCON 2-39

ASC1_TBSRC 2-58

ASC1_TBUF 2-36

ASC1_TSRC 2-58

ASC1_TXFCON 2-41

ASC1_WHBCON 2-33

C

CCU6 module registers 7-36

CCU60_CLC 7-94

CCU60_FDR 7-95

CCU61_CC63R 7-59

CCU61_CC63SR 7-60

CCU61_CC6xR 7-53

CCU61_CC6xSR 7-54

CCU61_CMPMODIF 7-42

CCU61_CMPSTAT 7-40

CCU61_IEN 7-84

CCU61_INP 7-88

CCU61_IS 7-77

CCU61_ISR 7-82

CCU61_ISS 7-80

CCU61_MCMCTR 7-71

CCU61_MCMOUT 7-68

CCU61_MCMOUTS 7-67

CCU61_MODCTR 7-61

CCU61_PISEL0 7-38

CCU61_PISEL2 7-40

CCU61_PSLR 7-65

CCU61_SRCx 7-102

CCU61_T12 7-51

CCU61_T12DTC 7-55

CCU61_T12MSEL 7-73

CCU61_T12PR 7-52

CCU61_T13 7-57

CCU61_T13PR 7-58

CCU61_TCTR0 7-43

CCU61_TCTR2 7-46

CCU61_TCTR4 7-49

CCU61_TRPCTR 7-63

G

GPTU module registers 6-22

GPTU_CLC 6-54

GPTU_OSEL 6-47

GPTU_OUT 6-48

GPTU_SRC0 6-61

GPTU_SRC1 6-61

GPTU_SRC2 6-61
GPTU_SRC3 6-61
GPTU_SRC4 6-61
GPTU_SRC5 6-61
GPTU_SRC6 6-61
GPTU_SRC7 6-61
GPTU_SRSEL 6-49
GPTU_T012RUN 6-41
GPTU_T01IRST 6-24
GPTU_T01OTS 6-27
GPTU_T0CBA 6-29
GPTU_T0DCBA 6-29
GPTU_T0RCBA 6-30
GPTU_T0RDCBA 6-30
GPTU_T1CBA 6-31
GPTU_T1DCBA 6-31
GPTU_T1RCBA 6-32
GPTU_T1RDCBA 6-31
GPTU_T2 6-45
GPTU_T2AIS 6-33
GPTU_T2BIS 6-35
GPTU_T2CON 6-38
GPTU_T2ES 6-36
GPTU_T2RC0 6-46
GPTU_T2RC1 6-46
GPTU_T2RCCON 6-43

I

IIC module registers 4-12
IIC_BUSCON 4-23
IIC_CLC 4-27
IIC_PISEL 4-13
IIC_RTb 4-24
IIC_SYSCON 4-14
IIC_WHBSYSCON 4-20
IIC_XP0SRC 4-31
IIC_XP1SRC 4-31
IIC_XP2SRC 4-31

M

MLI module registers 5-58
MLIO_AER 5-94
MLIO_ARR 5-95

MLIO_FDR 5-104
MLIO_GINTR 5-93
MLIO_OICR 5-80
MLIO_RADDR 5-78
MLIO_RCR 5-73
MLIO_RDATAR 5-78
MLIO_RIER 5-89
MLIO_RINPR 5-92
MLIO_RISR 5-91
MLIO_RPxBAR 5-76
MLIO_RPxSTATR 5-77
MLIO_SCR 5-79
MLIO_TCBAR 5-72
MLIO_TCMDR 5-66
MLIO_TCR 5-61
MLIO_TDRAR 5-71
MLIO_TIER 5-85
MLIO_TINPR 5-87
MLIO_TISR 5-86
MLIO_TPxAOFR 5-70
MLIO_TPxBAR 5-71
MLIO_TPxDATAR 5-70
MLIO_TPxSTATR 5-65
MLIO_TRSTATR 5-68
MLIO_TSTATR 5-63

S

SSC module registers 3-28
SSC0_BR 3-38
SSC0_CLC 3-50
SSC0_CON 3-31
SSC0_EFM 3-34
SSC0_ESRC 3-72
SSC0_FDR 3-51
SSC0_FSTAT 3-44
SSC0_PISEL 3-29, 3-53
SSC0_RB 3-39
SSC0_RSRC 3-72
SSC0_RXFCON 3-40
SSC0_SSOC 3-36
SSC0_SSOTC 3-37
SSC0_STAT 3-33
SSC0_TB 3-39

SSC0_TSRC 3-72
SSC0_TXFCON 3-42
SSC1_BR 3-38
SSC1_CLC 3-50
SSC1_CON 3-31
SSC1_EFM 3-34
SSC1_ESRC 3-72
SSC1_FDR 3-51
SSC1_FSTAT 3-44
SSC1_PISEL 3-29, 3-54
SSC1_RB 3-39
SSC1_RSRC 3-72
SSC1_RXFCON 3-40
SSC1_SSOC 3-36
SSC1_SSOTC 3-37
SSC1_STAT 3-33
SSC1_TB 3-39
SSC1_TSRC 3-72
SSC1_TXFCON 3-42

www.infineon.com

Published by Infineon Technologies AG