MPI12x

User Manual

Revision 3.00

profichip®
automation in silicon

Liability Exclusion

We have tested the contents of this document regarding agreement with the hardware and software described. Nevertheless, there may be deviations and we do not guarantee complete agreement. The data in the document is tested periodically, however. Required corrections are included in subsequent versions. We gratefully accept suggestions for improvements.

Copyright

This document is subject to technical changes.

Table of Contents

Part A General Chip Description

A-1 Introduction

This manual is divided into four major parts:

- Part A: General Chip Description
- Part B: MPI Operation Mode
- Part C: USART Operation Mode (not available in this chip release)
- Part D: PROFIBUS Slave (VPC3+) Operation Mode

Please make sure you have read the general part as well as those parts describing the operating modes which are intended to be used.

Profichip's **MPI12x** is a communication chip with configurable processor interface mainly targeted at low-cost MPI applications. The MPI12x handles the message and address identification, the data security sequences and the protocol processing for PROFIBUS, which is the underlying transport protocol. Token handling including fault detection and all low level protocol functions are done by the MPI12x automatically relieving the host processor from all timing critical tasks.

Support of data transmissions rates up to 12 Mbit/s, the hardware integration of the PROFIBUS protocol, 4 kBytes communication RAM and the configurable processor interface are features to create high-performance and cost-effective MPI applications. The device is to be operated with 3.3V single supply voltage and offers 5V tolerant inputs.

The chip also incorporates profichip's well-known PROFIBUS slave core, **VPC3+**, which can be used alternatively to the MPI functions. Therefore the MPI12x is also well suited for applications which require selective MPI or PROFIBUS slave functionality sharing the same hardware interface. However, due to sharing internal resources MPI and PROFIBUS slave functionality cannot be used simultaneously but have to be selected exclusively by software right after reset.

As a third option the MPI12x can be used as serial interface device supporting standard **16550 UART** functions as well as special **high-speed USART** modes. This feature allows the implementation of standard or proprietary synchronous and asynchronous serial protocols with transmission rates up to 12 Mbit/s. A special SRAM mode offers the opportunity to send and receive up to 2 kB of user data within a single transmission. Again this feature cannot be used simultaneously with MPI or PROFIBUS slave operation mode due to internal resource sharing.

Note:
The USART operation mode is not released in the current version of the MPI12x chip. However, it will be available in a future release.

Further information about profichip products or current and future projects is available on our web page: http://www.profichip.com.

Notes:

A-2 Overview

Thanks to its configurable 8-bit **Bus Interface Unit** the MPI12x supports a broad range of common microprocessor families including but not limited to

Intel:	80C31, 80X86
Siemens:	80C166/165/167
Motorola:	HC11-, HC16- and HC916 types

From the processor's point of view the MPI12x is a 4kB external static memory with optional ready logic. Therefore it is recommended to use a processor with external data and address bus (either multiplexed or non-multiplexed) and standard control signals. The bus mode (synchronous or asynchronous) as well as the data format (Intel or Motorola) can be selected by configuration pins. For compatibility with a certain type of processor please check the signal description and the related timing diagrams of the MPI12x as well as the electrical characteristics of the chip.

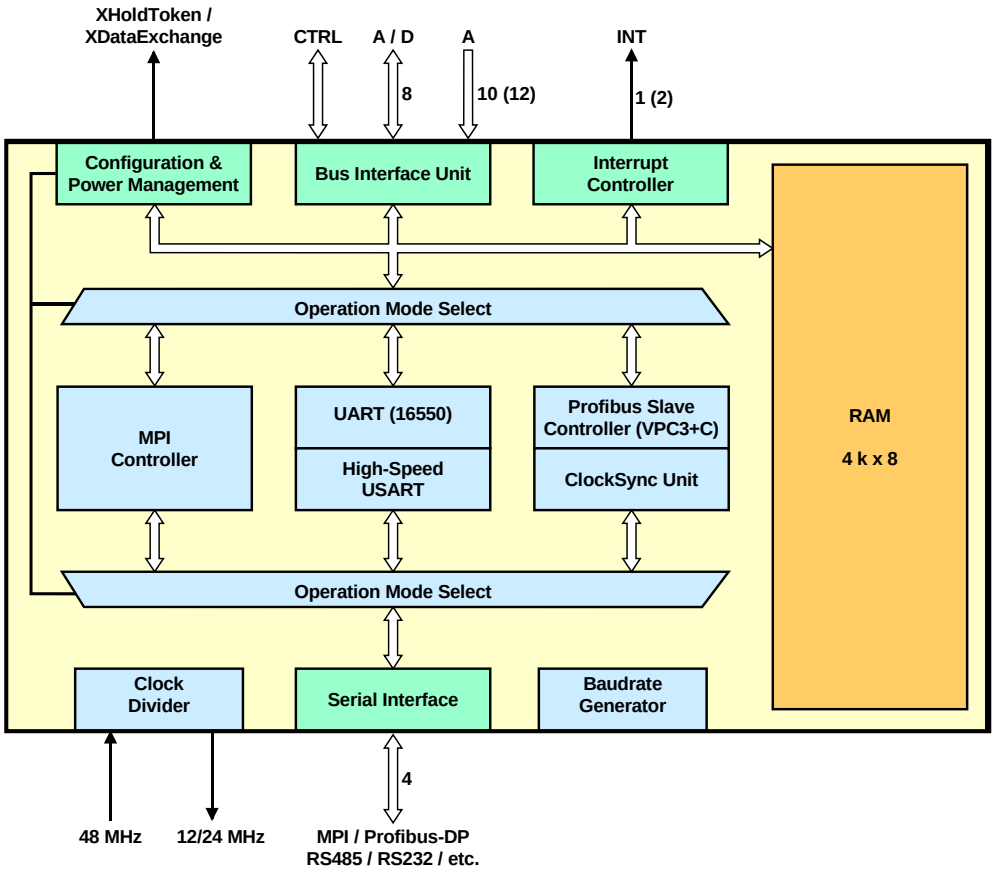


Figure A-1: Block Diagram of MPI12x

The MPI12x offers three **Operation Modes**:

- MPI Mode
- PROFIBUS Slave (VPC3+) Mode
- High-Speed USART Mode (not yet released)

The operation mode needs to be selected by software after reset. The operation modes are mutually exclusive and cannot be used simultaneously. Switching from one operation mode to another is only possible by resetting the chip and selecting a different operation mode thereafter. The **Power Management Unit** only enables those parts of the chip, required for the selected operation mode. For details about the different operation modes please see the corresponding sections of this manual.

The **4kB integrated RAM** serves as the communication memory for all three operation modes. Depending on the operation mode it is used for different data structures, communication buffers and task queues. After reset the RAM contents is undefined. Even if parts of the RAM are not used it is recommended to initialize the entire memory to a known value each time after selecting the desired operation mode.

The **Interrupt Controller** notifies the processor about pre-defined events occurring during protocol processing or data transmission. Which events are asserted depends on the Operation Mode of the chip and whether the corresponding interrupt sources are enabled by software.

The MPI12x needs to be operated with an external 48 MHz crystal oscillator. All internal clocks are derived from this master clock. Depending on the status of a configuration pin the integrated **Clock Divider** provides an output clock of 12 MHz or 24 MHz.

The 4-pin **Serial Interface** is shared between all operation modes. Galvanic isolation as well as the analog bus line drivers are not included in the chip but have to be added externally. The **Baudrate Generator** provides all the necessary send and receive clocks.

A-3 Signal Description

A-3.1 Pin Assignment

The MPI12x is available in a 44-pin Plastic Quad Flat Pack (PQFP). Figure A-2 shows the pin assignments of the device.

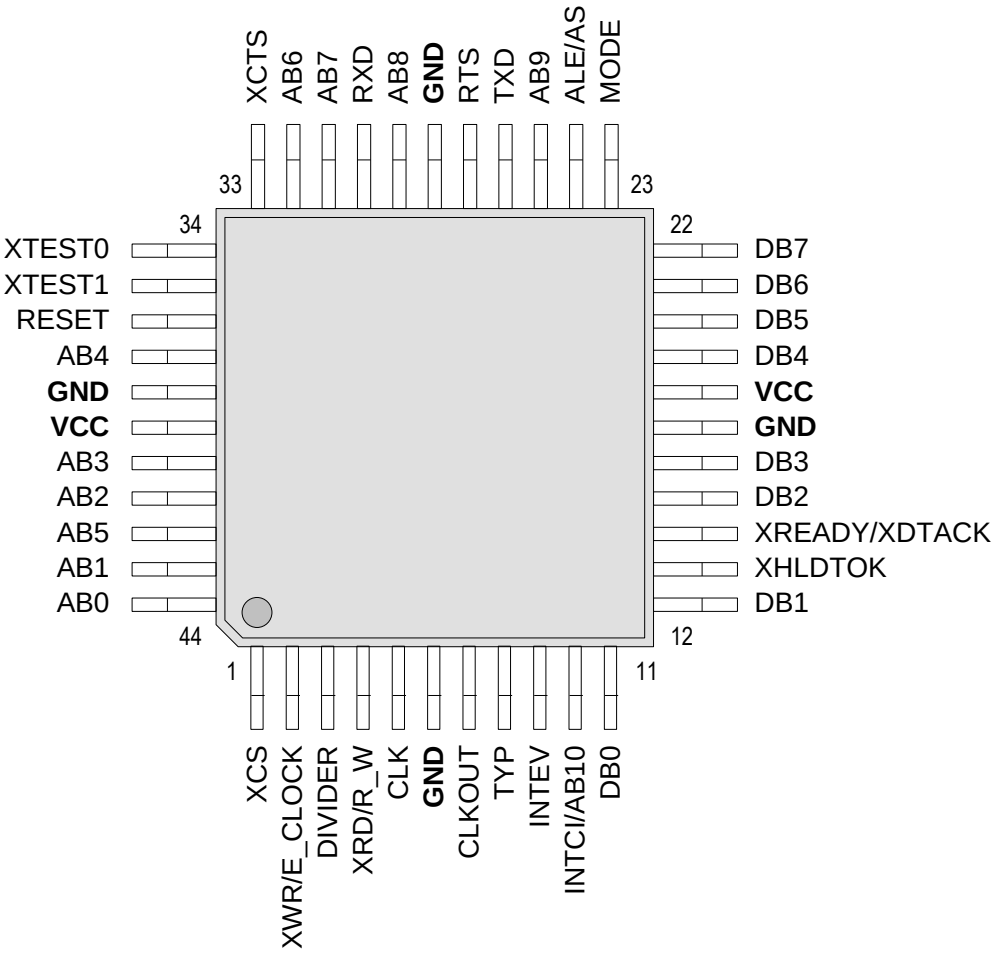


Figure A-2: PQFP44 Pin Assignments

A-3.2 Pin Table

Pin	Signal Name	In/Out	Description		Source / Destination
1	XCS	I(C)		C32 Mode: connect to VDD C165 Mode: CS-Signal	CPU
	AB11		Address Bus 11 (VPC3+; C32-Mode; 4 kB RAM)		
2	XWR / E_CLOCK	I(C)	Write Signal / E_Clock for Motorola		CPU
3	DIVIDER	I(C)	Setting the divider for CLKOUT	<log>0 = CLK divided by 4 <log>1 = CLK divided by 2	Configuration Pin
4	XRD / R_W	I(C)	Read Signal / Read _Write for Motorola		CPU
5	CLK	I(TS)	System Clock Input, 48 MHz		System
6	**GND**				
7	CLKOUT	O	Clock Output (System Clock divided by 2 or 4)		System, CPU
8	TYP	I(C)	<log>0 = Intel Interface <log>1 = Motorola Interface		Configuration Pin
9	INTEV	O	Interrupt		CPU; Interrupt-Controller
10	INTCI	O	Interrupt (MPI-Mode)		CPU; Interrupt-Controller
	AB10	I(CPD)	Address Bus (VPC3-Mode)	C32 Mode: <log>0 C165 Mode: Address Bus	CPU, Memory
11	DB0	I(C)/O	Data Bus	C32 Mode: Data/Address Bus multiplexed C165 Mode: Data/Address Bus separated	CPU, Memory
12	DB1	I(C)/O			
13	XHLDTOK	O	Indicates Hold Token State (MPI)		LED
	XDATAEXCH		Indicates Data-Exchange State for Profibus-DP (VPC3+)		LED
	SYNC		Synchronization Signal for Isochron Mode (VPC3+)		CPU
14	XREADY/XDTACK	O	Ready for external CPU		System, CPU
15	DB2	I(C)/O	Data Bus	C32 mode: Data /Address Bus multiplexed C165 mode: Data/Address Bus separate	CPU, Memory
16	DB3	I(C)/O			
17	**GND**				
18	**VCC**				
19	DB4	I(C)/O	Data Bus	C32 mode: Data/Address Bus multiplexed C165 mode: Data/Address Bus separate	CPU, Memory
20	DB5	I(C)/O			
21	DB6	I(C)/O			
22	DB7	I(C)/O			
23	MODE	I	<log>0 = 80C166 Data/Address Bus separated; Ready Signal <log>1 = 80C32 Data/Address Bus multiplexed, fixed Timing		Configuration Pin
24	ALE / AS	I(C)	Address Latch Enable	C32 mode: ALE C165 mode: <log>0	CPU
	AB11		Address Bus (Asyn. Intel and Syn. Motorola Mode; 4 kB RAM)		
25	AB9	I(C)	Address Bus	C32 Mode: <log>0 C165 Mode: Address Bus	CPU, Memory
26	TXD	O	Serial Transmit Port		Serial Interface
27	RTS	O	Request to Send		Serial Interface
28	**GND**				
29	AB8	I(C)	Address Bus	C32 mode: <log>0 C165 mode: address bus	CPU, Memory

30	RXD	I(C)	Serial Receive Port	Serial Interface
31	AB7	I(C)	Address Bus	CPU, Memory
32	AB6	I(C)		
33	XCTS	I(C)	Clear to Send: <log>0 = send enable	FSK Modem
34	XTEST0	I(C)	Pin must be connected to VCC.	
35	XTEST1	I(C)	Pin must be connected to VCC.	
36	RESET	I(CS)	Connect Reset Input with CPU's port pin.	
37	AB4	I(C)	Address Bus	CPU, Memory
38	**GND**			
39	**VCC**			
40	AB3	I(C)	Address Bus	CPU, Memory
41	AB2	I(C)		
42	AB5	I(C)		
43	AB1	I(C)		
44	AB0	I(C)		
Notes:			All signals beginning with X are LOW active. C32-Mode means 'Synchronous Intel Mode' and C165-Mode means 'Asynchronous Intel Mode'. MPI means 'MPI Op♣eration Mode' VPC3+ means 'VPC3+ Operation Mode'	
Input Levels:			I (C) : CMOS I (CS) : CMOS, Schmitt-Trigger I (CPD) : CMOS, pulldown resistor I (TS) : TTL, Schmitt-Trigger	

Figure A-3: Pin Table of MPI12x

A-3.3 Shared Pins

Depending on the selected processor interface mode, the operation mode and the internal chip configuration some pins of the MPI12x serve two or more functions, as described in the following paragraphs.

A-3.3.1 Pin 10 (INTCI / AB10):

After reset (before any operation mode has been selected) as well as in VPC3+ operation mode this pin serves as address input AB10 with integrated pull-down resistor. If synchronous Intel bus mode is selected this pin is part of those address lines which are used to decode the internal chip select signal. In all other bus modes this pin is used as 'standard' address input AB10 (without any internal chip select functionality).

After switching to MPI operation mode pin 10 is a push/pull output which can be used as a second interrupt line (INTCI) to the processor.

Note:
In MPI operation mode pin10 (INTCI) is used as a push/pull output. Please make sure that it is not driven externally in this operation mode.

A-3.3.2 Pin 13 (XHLDTOK / XDATAEXCH / SYNC)

After reset (before any operation mode has been selected) this pin is in-active (<log>1).

In MPI operation mode this pin indicates that the chip is in state 'Hold Token' (<log>0) and it is de-asserted (<log>1) otherwise.

In VPC3+ operation mode (non-isochronous Profibus mode) this pin indicates (<log>0) that the chip is in state 'Data Exchange'. When the isochronous Profibus mode is activated this pin is used to emit the Profibus synchronization pulses.

A-3.3.3 Address Pin AB11

In MPI operation mode the chip uses a segment and offset address mechanism in order to access the entire 4k internal memory. Therefore only 10 address lines AB[9:0] are necessary in this operation mode.

In VPC3+ operation mode linear addressing is used and a total of 11 address lines (2kB RAM mode) or 12 address lines (4kB RAM mode) are necessary to access the entire internal memory. AB10 is shared with INTCI at pin 10 as described earlier. If 4kB RAM mode is activated in VPC3+ operation mode hence another address line AB11 is required. Which pin is assigned to A11 depends on the Processor Interface Mode used.

Processor Interface Mode	Pin	Signal Name
Synchronous Intel Mode	1	XCS
Asynchronous Intel Mode	24	ALE/AS
Asynchronous Motorola Mode	2	XWR/E_CLOCK
Synchronous Motorola Mode	24	ALE/AS

Figure A-4: Pin Assignments for AB11 in VPC3+ Operation Mode

A-3.4 Test Pins

All output pins and I/O pins can be switched to the high-resistance state via the XTEST0 test pin.

Pin	Name	Value	Function
34	XTEST0	GND	All outputs high-resistance
		VCC	Normal function
35	XTEST1	GND	Various test modes
		VCC	Normal function

Figure A-5: Test Ports

An additional XTEST1 input is provided to test the chip on Automatic Test Equipment (ATE) but not in the target hardware!

Note:
The ATE test modes are for production tests only and not user accessible.

Notes:

A-4 Operation Mode Selection

The operation mode of the MPI12x can only be changed after resetting the chip. Only in this state the Operation Mode Select Register is writeable at address 000_H. The reset value of this register is 00_H (Operation Mode is 'NONE'). Selecting a certain operation mode is accomplished by writing a value different from 00_H to this register.

Address Intel	Address Mot.	Name Bit No.	Operation Mode
000_H	000_H	Operation Mode Select Reg. 7..0	00_H: NONE 01_H: MPI Operation Mode 02_H: USART Mode (not released) 03_H: VPC3+ Mode (Profibus Slave) Other values are not allowed and will hold the chip in mode 'NONE'.

Figure A-6: Operation Mode Select Register

Note:
Write access to the Operation Mode Select Register needs to be carried out according to the configured processor interface mode. AB10 (pin 10) needs to be held low or can be left undriven due to internal pull-down resistor. If MPI mode is selected pin 10 will be switched to a push/pull output immediately and may not be driven externally.

Reading from the MPI12x as long as no operation mode is selected will result in undefined read data. However, the chip will generate the READY signal according to the configured processor bus mode in order to prevent the processor from hanging-up.

After a certain operation mode has been selected the Operation Select Register is no longer accessible. It takes 10 system clock cycles (about 200 ns) for the Power Management Unit to enable the required internal function blocks. Thereafter the memory structure for the selected operation mode is visible.

Note:
After writing the Operation Select Register do not access the MPI12x for 200 ns due to internal startup procedure.

Notes:

A-5 Mixed Applications (MPI / VPC3+)

The MPI12x can be used easily in mixed applications which require switching between MPI and PROFIBUS-DP slave functionality on the same hardware. However, care needs to be taken with respect to the shared pin 10 which is an input pin (AB10) after reset and in VPC3+ operation mode while it is a push/pull output (INTCI) in MPI operation mode. Therefore connecting this pin directly to the processor's address bus will lead to a possible driver conflict when switching to MPI operation mode.

In Synchronous Intel Processor Mode it is recommended not to connect pin 10 to the processor's address bus but to use an external pull-down resistor. Optionally pin 10 could be connected to a second interrupt input of the processor for MPI operation mode with two separate interrupt sources.

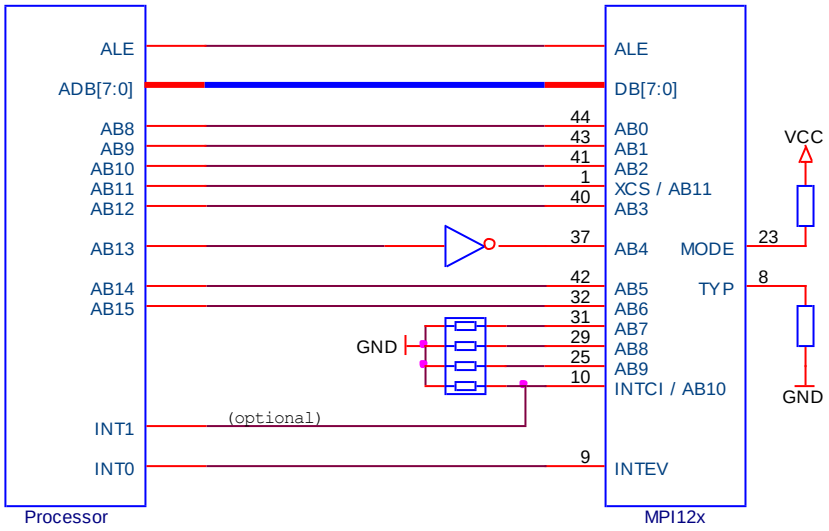


Figure A-7: Mixed Application (Principle), Synchronous Intel Mode

If using a processor mode other than Synchronous Intel for mixed applications it is recommended to use an external tri-state driver in order avoid driver conflicts on the address bus when switching from VPC3+ to MPI operation mode. The driver needs to be enabled in VPC3+ operation mode while it has to be disabled in MPI operation mode.

Note:
The external tri-state driver is only needed for mixed applications (where switching between MPI and VPC3+ operation mode is required) which do not use Synchronous Intel Processor Mode.
The tri-state driver is not required if MPI operation mode and VPC3+ operation mode are used mutually exclusive.

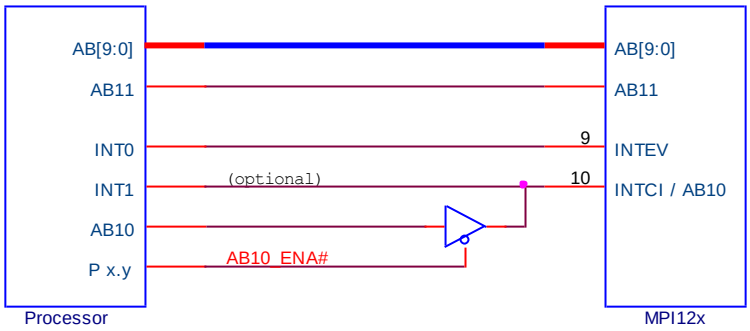


Figure A-8: Mixed Application (Principle), not Synchronous Intel Mode

For applications which use MPI operation mode exclusively pin 10 can be left unconnected due to internal pull-down resistor. Optionally it can be connected to a second interrupt input of the processor if two separate interrupts are to be used in MPI operation mode.

For applications which use VPC3+ operation mode exclusively pin 10 is to be connected to the address bus of the processor or can be left unconnected in Synchronous Intel Processor Mode due to internal pull-down resistor.

CAUTION:
If pin 10 is directly connected to the address bus of the processor (or any other output driver), please make sure that the MPI12x will not be switched to MPI operation mode. This would result in a possible driver conflict causing permanent damage to the MPI12x and/or the other device.

A-6 Operational Specification

A-6.1 Absolute Maximum Ratings

Parameter	Symbol	Limits	Unit
DC supply voltage	V_{CC}	-0.3 to 3.9	V
Input voltage	V_I	-0.3 to +6.3	V
Output voltage	V_O	-0.3 to V_{CC} +0.3	V
DC output current	I_O	See Figure A-12	mA
Storage temperature	T_{store}	-40 to +150	°C

Figure A-9: Absolute Maximum Ratings

Note:
Extended operation at the maximum ratings can adversely affect device reliability.

A-6.2 Recommended Operating Conditions

Parameter	Symbol	MIN	TYP	MAX	Unit
DC supply voltage	V_{CC}	3.00	3.30	3.60	V
Circuit ground	GND	0	0	0	V
Input voltage	V_I	0	3.30	5.50	V
Current consumption	I_A		45		mA
Static supply current	I_{DDQ}			100[1)]	µA
Ambient temperature	T_A	-40	+25	+85	°C

[1)] : Static I_{DDQ} current is exclusively of input/output drive requirements and is measured with the clock stopped and all inputs tied to V_{CC} or GND.

Figure A-10: Recommended Operating Conditions

A-6.3 Leakage Current and Capacitance

Parameter	Symbol	MIN	TYP	MAX	Unit
Input leakage current	I_{IN}	-10	±1	+10	µA
Tri-state leakage current	I_{OZ}	-10	±1	+10	µA
Input capacitance	C_{IN}		4		pF
Output capacitance	C_{OUT}		4.4		pF
Bi-directional buffer capacitance	C_{BID}		4.4		pF

Figure A-11: General DC Characteristics

A-6.4 Ratings of the Output Drivers

Signal	Direction	Driver Type	Driver Strength	Max. Cap. Load
DB 0-7	I/O	Tristate	8mA	100pF
RTS	O	Push / Pull	8mA	50pF
TXD	O	Tristate	8mA	50pF
INTEV	O	Push / Pull	4mA	50pF
INTCI	O	Push / Pull	4mA	50pF
XREADY/XDTACK	O	Tristate	4mA	50pF
XHLDTOK	O	Push / Pull	8mA	50pF
CLKOUT	O	Push / Pull	8mA	100pF

Figure A-12: Ratings for the Output Drivers

A-6.5 DC Characteristics of I/O Cells

Parameter	Symbol	MIN	TYP	MAX	Unit
DC supply voltage	V_{CC}	3.00	3.30	3.60	V
CMOS input voltage LOW level	V_{ILC}	0		0.3 V_{CC}	V
CMOS input voltage HIGH level	V_{IHC}	0.7 V_{CC}		5.50	V
CMOS Schmitt Trigger negative going threshold voltage	V_{T-}	0.9	1.2		V
CMOS Schmitt Trigger positive going threshold voltage	V_{T+}		2.1	2.5	V
TTL Schmitt Trigger negative going threshold voltage	V_{T-}	0.9	1.1		V
TTL Schmitt Trigger positive going threshold voltage	V_{T+}		1.6	1.9	V
Output voltage LOW level	V_{OL}			0.4	V
Output voltage HIGH level	V_{OH}	2.4			V

Figure A-13: DC Specification of I/O Cells

A-6.6 Timing Characteristics

A-6.6.1 System Interface

Clock

Clock frequency is 48 MHz. Distortion of the clock signal is permissible up to a ratio of 30:70 at the threshold levels 0.9 V and 1.9 V.

Parameter	Symbol	MIN	MAX	Unit
Clock period	T	20.83	20.83	
Clock high time	T_{CH}	6.25	14.6	ns
Clock low time	T_{CL}	6.25	14.6	ns
Clock rise time	T_{CR}		4	ns
Clock fall time	T_{CF}		4	ns

Figure A-14: Clock Timing

Interrupt:

After acknowledging an interrupt with EOI, the interrupt output of the MPI12x is deactivated for at least 1 us or 1 ms depending on the bit ‘EOI_Timebase’ in Mode Register 0 (VPC3+ Operation Mode) or EOI_Time in Mode Register 2 (MPI Operation Mode).

Parameter	MIN	MAX	Unit
Interrupt inactive time EOI_Timebase = ‘0’	1	1	µs
Interrupt inactive time EOI_Timebase = ‘1’	1	1	ms

Figure A-15: End-of-Interrupt Timing

Reset:

MPI12x requires a minimum reset phase of 100 ns.

A-6.6.2 Timing in the Synchronous Intel Mode

In the synchronous Intel mode, the MPI12x latches the least significant addresses with the falling edge of ALE. At the same time, the MPI12x expects the most significant address bits on the address bus. An internal chipselect signal is generated from the most significant address bits. The request for an access to the MPI12x is generated from the falling edge of the read signal (XRD) and from the rising edge of the write signal (XWR).

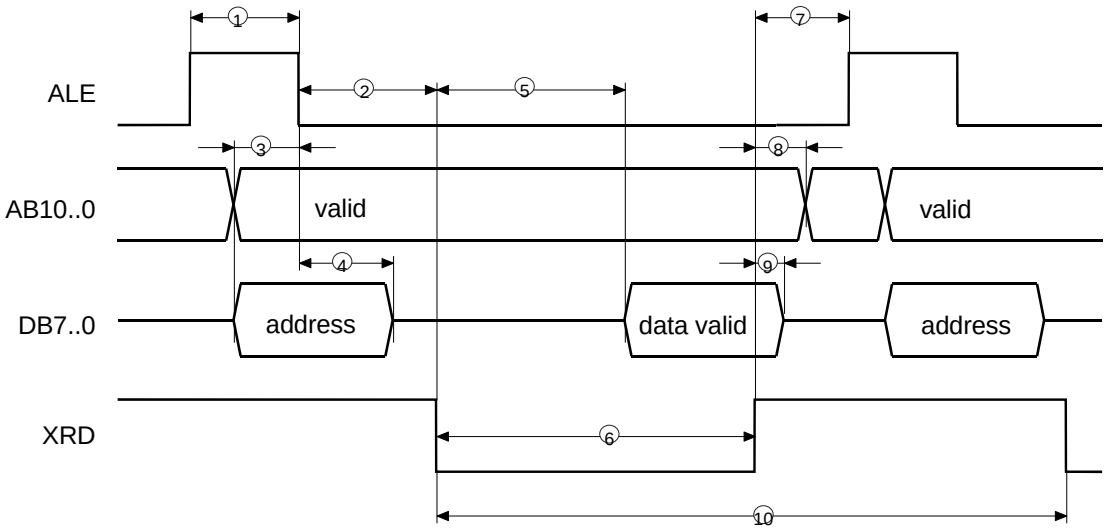


Figure A-16: Synchronous Intel Mode, READ (XWR = 1)

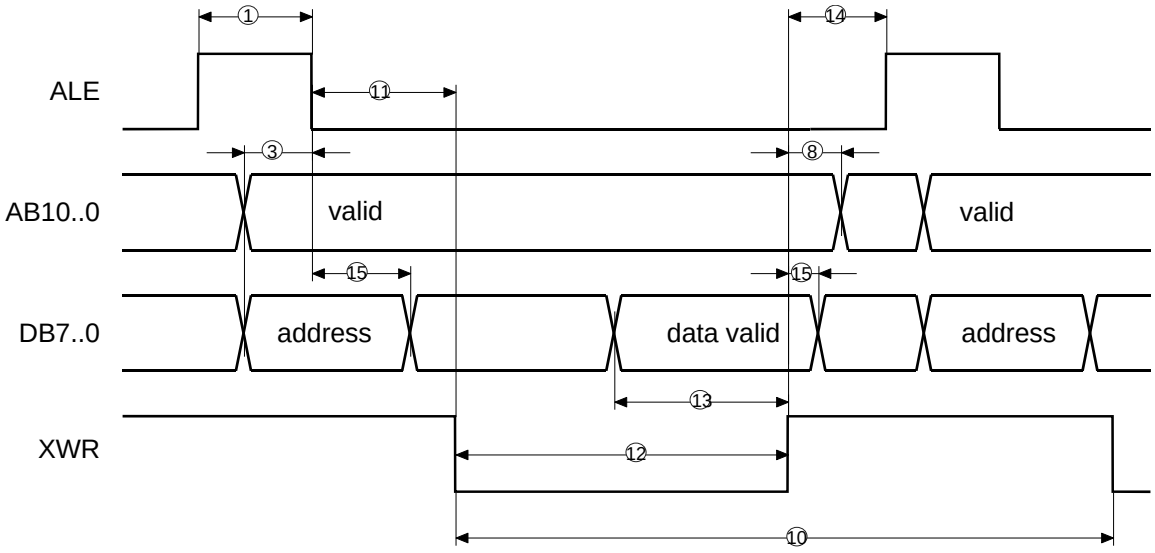


Figure A-17: Synchronous Intel Mode, WRITE (XRD = 1)

No.	Parameter	MIN	MAX	Unit
1	ALE pulse width	10		ns
2	ALE ↓ to XRD ↓	20		ns
3	Address to ALE ↓ setup time	10		ns
4	Address hold time after ALE ↓	10		ns
5	XRD ↓ to data valid		85	ns
6	XRD pulse width	115		ns
7	XRD ↑ to ALE ↑	10		ns
8	Address (AB7..0) hold time after XRD/XWR ↑	5		ns
9	Data hold time after XRD ↑	3	13	ns
10	XRD / XWR cycle time	155		ns
11	ALE ↓ to XWR ↓	20		ns
12	XWR pulse width	83		ns
13	Data setup time to XWR ↑	10		ns
14	XWR ↑ to ALE ↑	10		ns
15	Data hold time after XWR ↑	10		ns

Figure A-18: Timing, Synchronous Intel Mode

A-6.6.3 Timing in the Asynchronous Intel Mode

In the asynchronous Intel mode, the MPI12x acts like a memory with ready logic. The access time depends on the type of access. The request for an access to the MPI12x is generated from the falling edge of the read signal (XRD) or the rising edge of the write signal (XWR).

The MPI12x generates the Ready signal synchronously to the system clock. The Ready signal gets inactive when the read or the write signal is deactivated. The data bus is switched to Tristate with XRD = '1'.

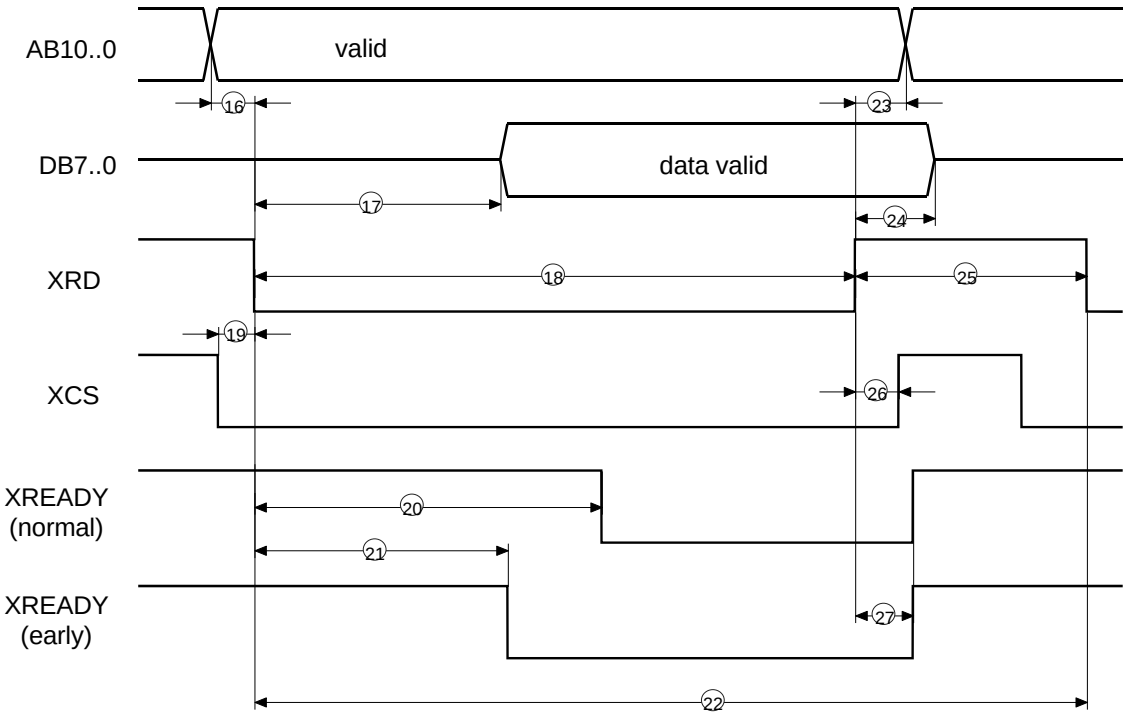


Figure A-19: Asynchronous Intel Mode, READ (XWR = 1)

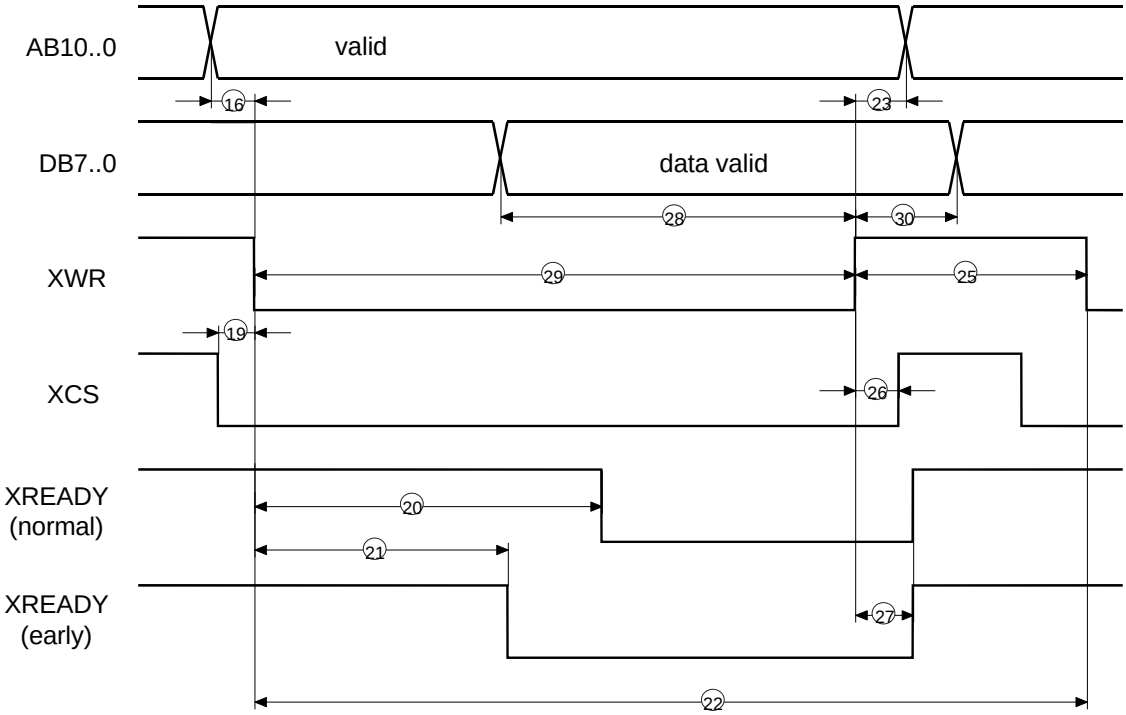


Figure A-20: Asynchronous Intel Mode, WRITE (XRD = 1)

No.	Parameter	MIN	MAX	Unit
16	address-setup time to XRD / XWR ↓	0		ns
17	XRD ↓ to data valid		85	ns
18	XRD pulse width	115		ns
19	XCS ↓ setup time to XRD / XWR ↓	0		ns
20	XRD ↓ to XREADY ↓ (Normal-Ready)		122	ns
21	XRD ↓ to XREADY ↓ (Early-Ready)		100	ns
22	XRD / XWR cycle time	125		ns
23	address hold time after XRD / XWR ↑	0		ns
24	data hold time after XRD ↑	3	13	ns
25	read/write inactive time	10		ns
26	XCS hold time after XRD / XWR ↑	0		ns
27	XREADY hold time after XRD / XWR	3	13	ns
28	data setup time to XWR ↑	10		ns
29	XWR pulse width	83		ns
30	data hold time after XWR ↑	10		ns

Figure A-21: Timing, Asynchronous Intel Mode

A-6.6.4 Timing in the Synchronous Motorola Mode

If the CPU is clocked by the MPI12x, the output clock pulse (CLKOUT) must be 4 times larger than the E_Clock. That is, a clock pulse signal must be present at the CLK input that is at least 10 times larger than the desired system clock pulse (E_Clock). The Divider-Pin must be connected to <log.0> (divider 4). This results in an E_Clock of 3 MHz.

The request for a read access to the MPI12x is derived from the rising edge of the E_Clock (in addition: XCS = 0, R_W = 1). The request for a write access is derived from the falling edge of the E_Clock (in addition: XCS = 0, R_W = 0).

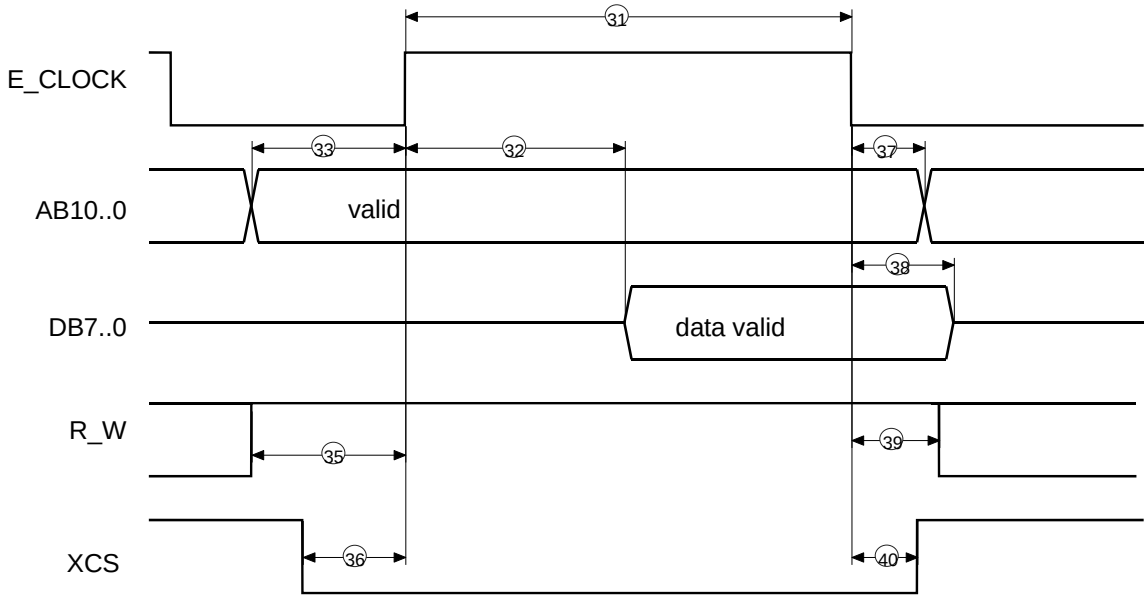


Figure A-22: Synchronous Motorola-Mode, READ (AS = 1)

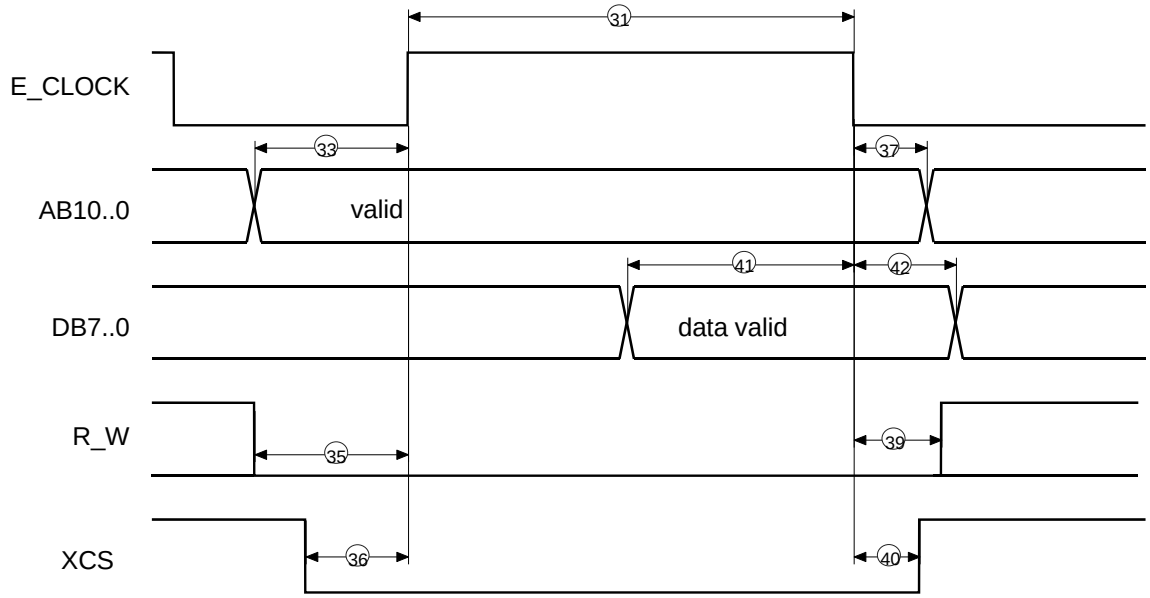


Figure A-23: Synchronous Motorola-Mode, WRITE (AS = 1)

No.	Parameter	MIN	MAX	Unit
31	E_Clock pulse width	136.7		ns
33	Address setup time (A10..0) to E_Clock ↑	10		ns
37	Address hold time after E_Clock ↓	5		ns
32	E_Clock ↑ to Data valid		85	ns
38	Data hold time after E_Clock ↓	3	13	ns
35	R_W setup time to E_Clock ↑	10		ns
39	R_W hold time after E_Clock ↓	5		ns
36	XCS setup time to E_Clock ↑	0		ns
40	XCS hold time after E_Clock ↓	0		ns
41	Data setup time to E_Clock ↓	10		ns
42	Data hold time after E_Clock ↓	10		ns

Figure A-24: Timing, Synchronous Motorola Mode

A-6.6.5 Timing in the Asynchronous Motorola Mode

In the asynchronous Motorola mode, the MPI12x acts like a memory with Ready logic, whereby the access times depend on the type of access.

The request for an access of the MPI12x is generated from the falling edge of the AS signal (in addition: XCS = '0', R_W = '1'). The request for a write access is generated from the rising edge of the AS signal (in addition: XCS = '0', R_W = '0').

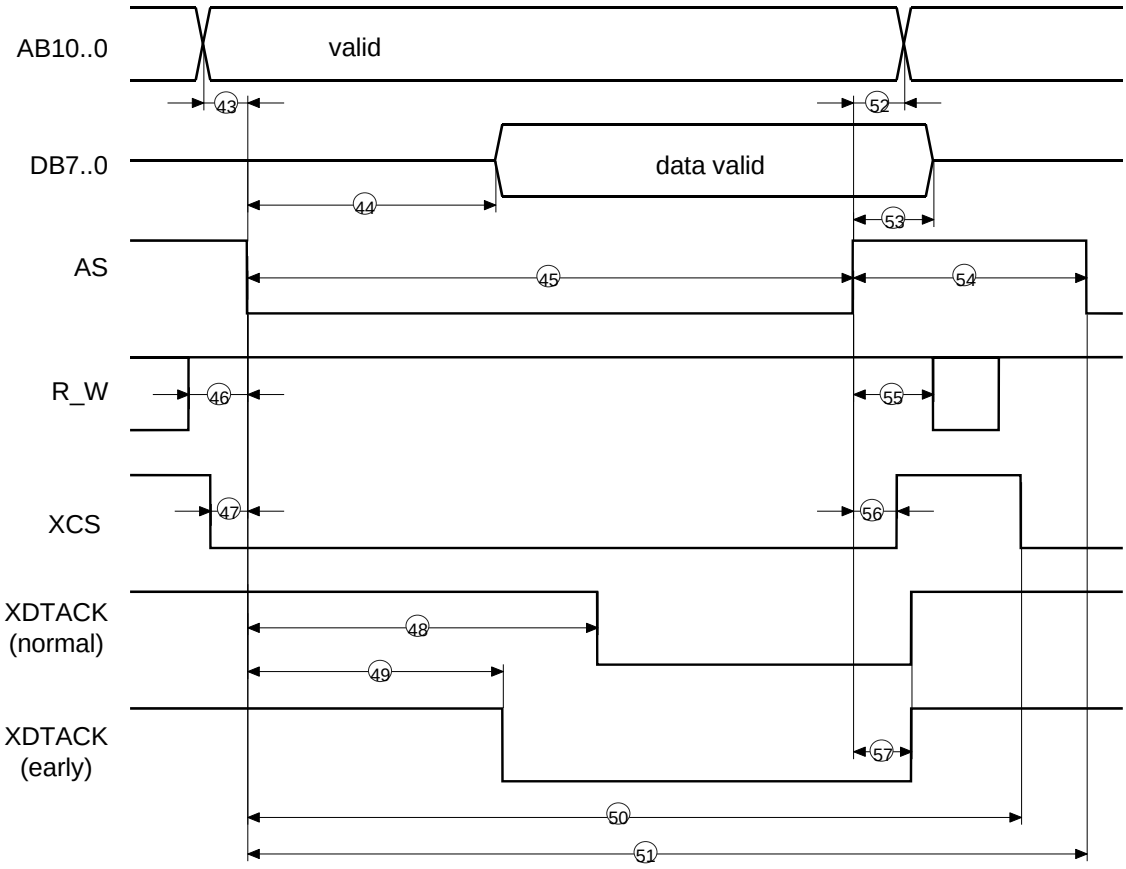


Figure A-25: Asynchronous Motorola Mode, READ (E_CLOCK = 0)

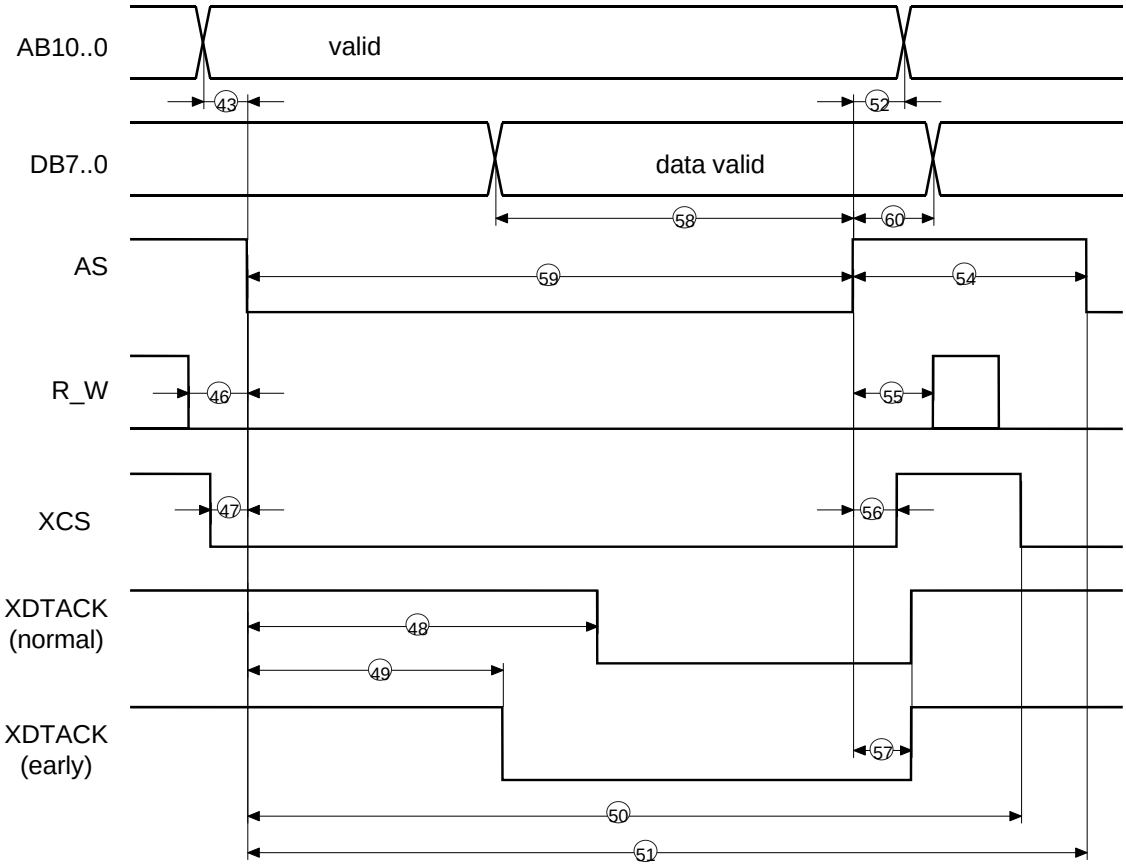


Figure A-26: Asynchronous Motorola Mode (WRITE)

No.	Parameter	MIN	MAX	Unit
43	Address setup time to AS ↓	0		ns
44	AS ↓ to data valid		85	ns
45	AS pulse width (read access)	115		ns
46	R_W ↓ setup time to AS ↓	10		ns
47	XCS ↓ setup time to AS ↓	5		ns
48	AS ↓ to XDTACK ↓ (Normal-Ready)		122	ns
49	AS ↓ to XDTACK ↓ (Early-Ready)		100	ns
50	Last AS ↓ to XCS ↓	93		ns
51	AS cycle time	125		ns
52	Address hold time after AS ↑	10		ns
53	Data hold time after AS ↑	3	13	ns
54	AS inactive time	10		ns
55	R_W hold time after AS ↑	10		ns
56	XCS hold time after AS ↑	0		ns
57	XDTACK hold time after AS ↑	3	13	ns
58	Data setup time to AS ↑	10		ns
59	AS pulse width (write access)	83		ns
60	Data hold time after AS ↑	10		ns

Figure A-27: Timing, Asynchronous Motorola Mode

A-6.7 Package

The MPI12x comes in a 44-pin PQFP package. It is compliant to the “Reduction of Hazardous Substances (RoHS) Directive” of the European Parliament. Please see the following figures for outlines and dimensions.

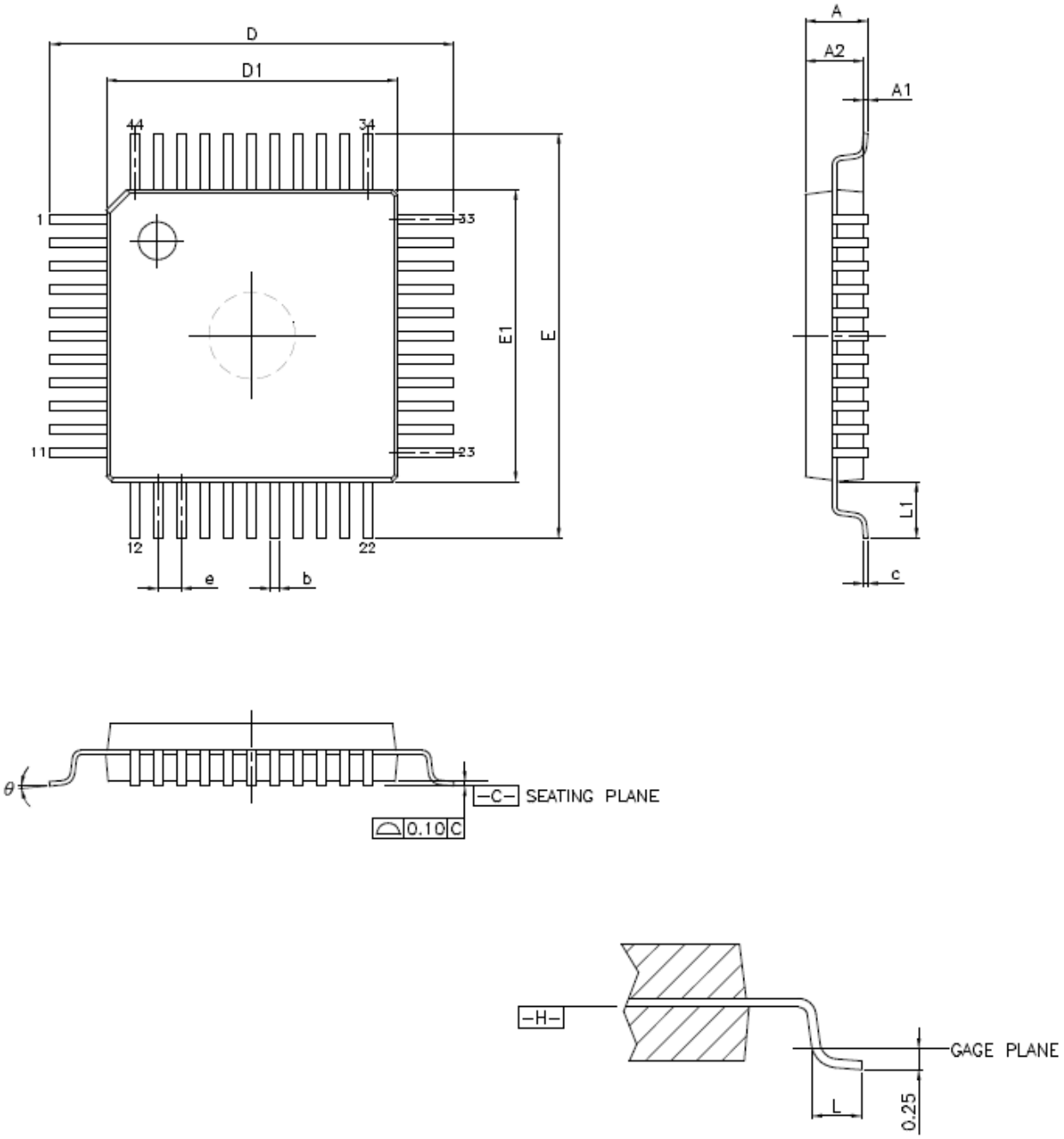


Figure A-28: Package Drawing

SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	——	——	2.70
A1	0.25	——	0.35
A2	1.80	2.00	2.20
b	0.22	0.30	0.38
c	0.10	0.15	0.20
E	0.80 BSC.		
D	13.70	13.90	14.10
D1	9.90	10.00	10.10
E	13.70	13.90	14.10
E1	9.90	10.00	10.10
L	0.73	0.88	0.93
L1	1.95 REF		
Θ	0°	——	7°

Figure A-29 : Package Dimensions and Tolerances

Notes:

1. JEDEC outline: n/a
2. Datum plane ⊢H⊣. is located at the bottom of the mold parting line coincident with where the lead exits the body.
3. Dimensions D1 and E1 do not include mold protrusion. Allowable protrusion is 0.25 mm per side. Dimensions D1 and E1 do not include mold mismatch and are determined at datum plane ⊢H⊣.
4. Dimension b does not include dambar protrusion.

A-6.8 Processing Instructions

Internal circuitry protects the inputs against damage caused by high static voltages or electric fields; however, normal precautions are necessary to avoid application of any voltage higher than maximum-rated voltages to this circuit.

The MPI12x is a cracking-endangered component that must be handled properly.

Profichip products are tested and classified for moisture sensitivity according to the procedures outlined by JEDEC. The MPI12x is classified as moisture sensitivity level (MSL) 3.

Note:
In order to minimize any potential risk caused by moisture trapped inside non-hermetic packages it is a general recommendation to perform a drying process before soldering.

A-6.9 Ordering Information

Version	Order Code	Package	Temperature Range	Notes
MPI12x	PALF2060	PQFP44	Industrial (-40°C to +85°C)	

Notes:

B-1 Overview

The MPI12x handles the physical layer 1 and the data link layer 2 of the ISO/OSI-reference-model excluding the analog RS485 drivers.

The following figure shows the block diagram of the MPI12x in MPI operation mode.

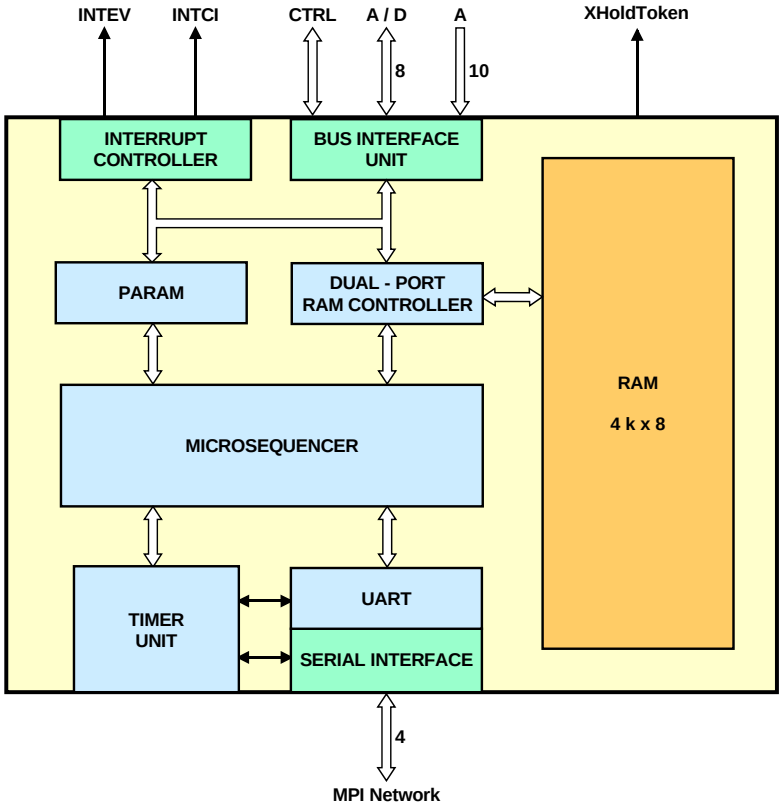


Figure B-1: Block Diagram MPI Operation Mode

The FLC is connected to the **Bus Interface Unit**. It has an 8 bit data bus and a separate 10 bit address bus. The bus mode (synchronous or asynchronous) as well as the microcontroller format (Intel or Motorola) can be selected by configuration pins.

All registers which parameterize the chip are located in the module **Param** (e.g. baud rate or slot time). Also the status register, which delivers information about the chip, is located there.

Events (e.g. indication and confirmation) are stored in the **Interrupt Controller**. The events which lead to an interrupt can be locked and unlocked with the interrupt mask. There is a separate register for acknowledging an interrupt. The chip has two separate interrupt lines, one only for confirmation and indication and the other for error events. It is also possible to put all events to only one interrupt line.

The **Microsequencer** represents the media access state machine. It is responsible for the token handling, processes the request telegrams in state hold-token and the indications in state not-hold-token. Indications to other stations are filtered. In state hold-token the remaining token hold time is calculated, both request lists are checked for queued requests. If a request is queued the data is moved from the RAM to the UART and the response (if available) is stored into the supplied area. In state not-hold-token the chip checks the incoming indications for plausibility and performs the SAP verification. Only if both conditions are met the chip indicates the telegram to the FLC and sends an answer if available.

The internal 4 KB SRAM is controlled by a **Dual Port RAM Controller**. There are circular buffers for Request/Response and Indication/ Confirmation handling. Additionally parameter cells, the LAS and the SAP list are located there. The integrated address calculation unit calculates the access of the queues. In the LAS area there is a list of all active stations deposited. This list is necessary to guarantee an accurate token traffic.

The different timers for the system are located in the **Timer Unit**. The *idle timer* controls the idle periods of the serial bus. In a passive station or in state not-hold-token it counts the T_{SYN} respectively the T_{RDY} time. In an active station holding the token it counts T_{ID1} or T_{ID2} depending on the last request. The *syni timer* controls the communication media. If there is no receiver synchronization within T_{SYNI} there is a permanent disturbance. In this case the MAC will go into state not-hold-token and generate an error interrupt. In state hold-token the *slot timer* is used to check if there is an activity from the remote station within T_{SLOT}. If the timer expires the token or request will be repeated. In state not-hold-token or passive stations the slot time is used for generating an event for the timeout timer. The *timeout timer* controls the bus activity. The timeout time is a multiple of the slot time and depends of the stations address. If there is no activity within this time there has been a loss of the token. This will lead to a re-initialization of all stations. The *target rotation timer* controls the token turnaround time. The target rotation time T_{TR} of the token is a multiple of 256 T_{BIT}.

The **UART** converts the internal parallel data to a serial data stream and vice versa. The transmission technique is asynchronous with start-, 8 data-, parity- and stop bit. The baud rate can be adjusted between 9.6 KBit/s and 12MBit/s.

B-2 Memory Organization

B-2.1 Overview

The MPI12x contains different memory areas for Common Parameters, LAS, SAP-List, Queues and Control Registers.

The whole area is addressed with a 10 bit address bus. Some areas could be read and written directly others only with segment and offset addressing. Figure B-2 shows the address map of the MPI12x and the table explains how addressing is done.

A9	A8	address window
0	0	Parameter (physical 000 H to 0FF H)
0	1	Parameter (physical 100 H to 1FF H)
1	0	256 byte within 4 Kbyte RAM (Base-Ptr & Offset)
1	1	Control Registers

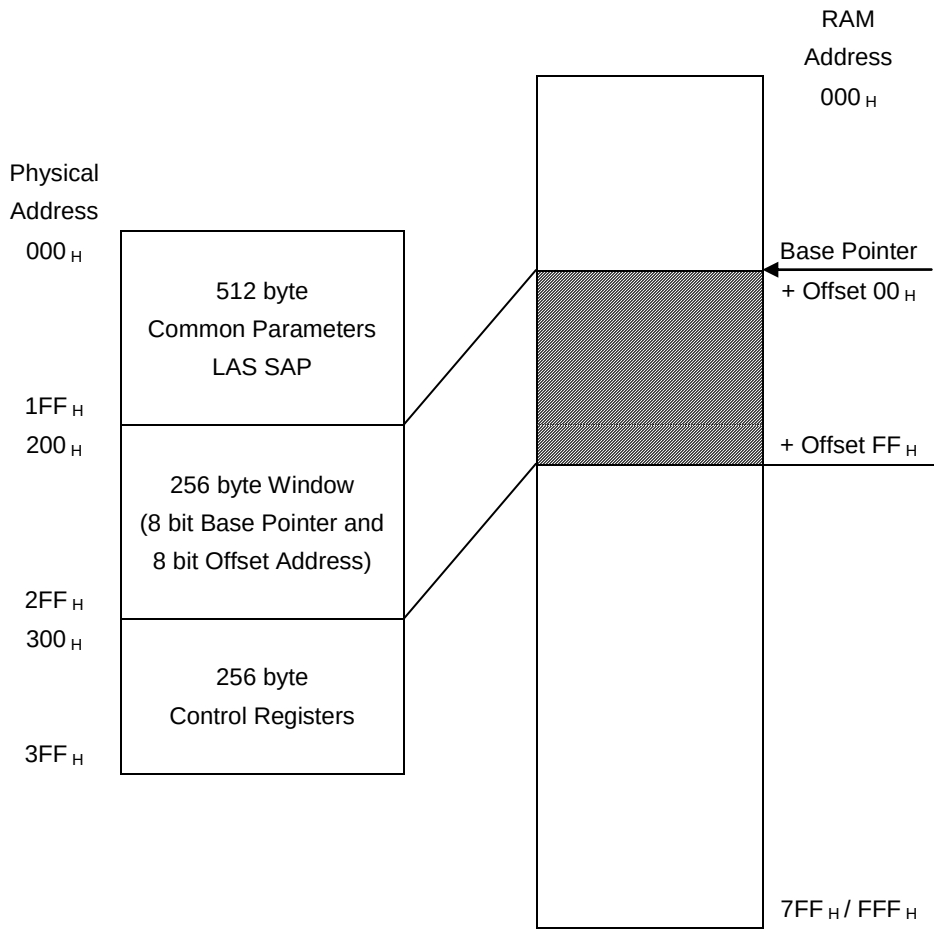


Figure B-2: Memory Access

The RAM area is split into separate areas: Control Parameters, LAS, SAP List and Queues. The RAM can only be accessed by using the 256 byte address window defined by base pointer and offset address. Figure 4-2 shows the memory map of the MPI12x RAM.

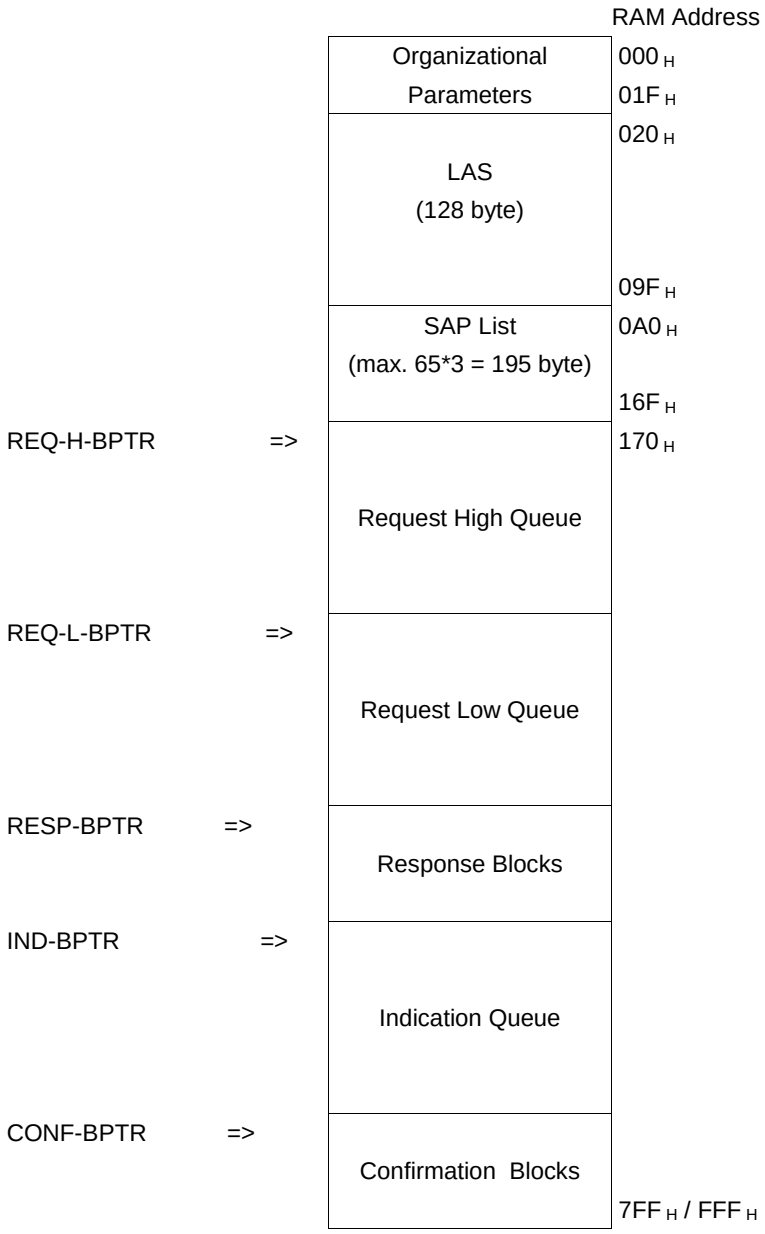
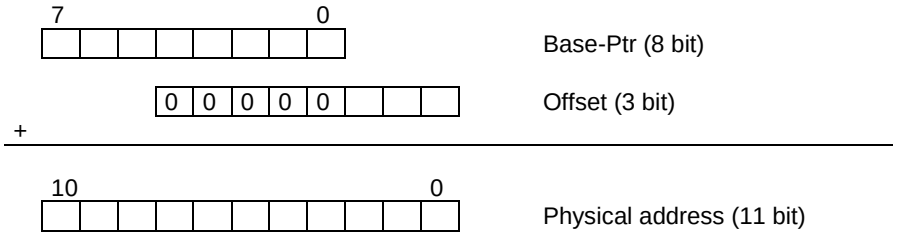


Figure B-3: Memory Table

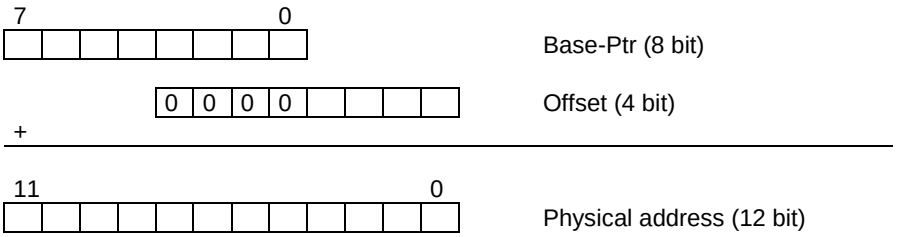
The internal RAM of the MPI12x can be operated in two different modes, the 2 kByte mode and the 4 kByte mode. The RAM operation mode can be selected by bit 0 in Mode Register 2.

In 2 kB mode the RAM has a granularity of 8 bytes whereas in the 4 kB mode the RAM granularity is 16 bytes. The following tables are describing how the physical RAM address is built when reading or writing addresses from 200_H to $2FF_H$.

2 kByte Mode:



4 kByte Mode:



As the RAM is addressed with segment and offset it would be necessary to recalculate the segment address if there is a wrap-around when writing to a queue/block. Since this is time consuming the wrap-around is done by the chip automatically. Therefore the FLC can read/write 256 byte consecutively without recalculating and repositioning the segment pointer.

Note:
The automatic wrap-around is only done once! If the FLC reads or writes beyond a queue or block border again, no automatic wrap-around will be generated (no modulo behavior).

B-2.2 Control Registers

The Control Registers are located in the register area starting with address 300_H. The registers are read- and writable, but most of them could only be written in state 'Offline' (except interrupt register and mode register 1).

Physical Address Intel	Physical Address Mot.	Name	Bit No.	Description
300_H	301_H	Int-Mask-Reg	7..0	Interrupt Controller
301_H	300_H	Int-Mask-Reg	15..8	
302_H	303_H	Int-Ack-Reg	7..0	
303_H	302_H	Int-Ack-Reg	15..8	
304_H		Mode-Reg 0*	7..0	Mode Register
305_H		Mode-Reg 1	7..0	
306_H	307_H	Tslot-Reg*	7..0	Slot Time
307_H	306_H	Tslot-Reg*	15..8	
308_H	309_H	Tid1-Reg*	7..0	Idle Time 1
309_H	308_H	Tid1-Reg*	15..8	
$30A_H$	$30B_H$	Tid2-Reg*	7..0	Idle Time 2
$30B_H$	$30A_H$	Tid2-Reg*	15..8	
$30C_H$	$30D_H$	BR-Reg*	7..0	Baudrate
$30D_H$	$30C_H$	BR-Reg*	15..8	
$30E_H$		Trdy-Reg*	7..0	Ready Time
$30F_H$		Tqui-Reg*	7..0	Quiet Time
310_H		Base-Ptr*	7..0	FLC Basepointer
311_H		Req-H-BPtr*	7..0	Basepointer
312_H		Req-L-BPtr*	7..0	
313_H		Resp-BPtr*	7..0	
314_H		Ind-BPtr*	7..0	
315_H		Conf-BPtr*	7..0	
316_H	317_H	Ttr-Reg*	7..0	Target Rotation Time
317_H	316_H	Ttr-Reg*	15..8	
318_H		Mem-Lock	7..0	Memory Lock
319_H		Mode-Reg 2*	7..0	Mode Register

$31A_H$	$31B_H$	Int-Req-Reg	7..0	Interrupt Controller
$31B_H$	$31A_H$	Int-Req-Reg	15..8	
$31C_H$	$31D_H$	Int-Reg	7..0	
$31D_H$	$31C_H$	Int-Reg	15..8	
$31E_H$	$31F_H$	Status-Reg	7..0	Status Register
$31F_H$	$31E_H$	Status-Reg	15..8	

Figure B-4: Assignment of the Internal Control Registers

* Only writable in state ‘Offline‘.
Registers shaded in grey are read only.

B-2.3 Organizational Parameters

The Organizational Parameters are located in the RAM area starting with RAM address 000_H. The parameters are read- and writeable, but may only be changed in state ‘Offline’, except for the pointers.

RAM Address Intel	RAM Address Mot.	Name	Bit No.	Description
000_H		REQ-H-WP	7..0	Request High Read/Write Pointer
001_H		REQ-H-RP	7..0	
002_H		REQ-L-WP	7..0	Request Low Read/Write Pointer
003_H		REQ-L-RP	7..0	
004_H		IND-WP	7..0	Indication Read/Write Pointer
005_H		IND-RP	7..0	
006_H		Retry	3..0	Request Retries
007_H		TS Addr	6..0	Station Address
008_H		GUD factor	7..0	Gap Update Factor
009_H		Token-Error-Limit	7..0	Token Error Limit

Figure B-5: Assignment of the Organizational Parameters

B-2.4 Accessing the Dual Port RAM

In the Dual Port RAM there are some areas which are used by the FLC as well as from the MS (LAS and SAP list). To avoid data inconsistency doing a write or a read-modify-write access there is a lock mechanism implemented in the chip. If the FLC wants to access such a memory area, it has to check first if the memory is locked by the MS. Therefore there is a symbolic cell in the register space called 'MEM-LOCK' (address 318_H). The value of this cell can be accessed by reading the register; it is put on data-bit(0).

If the value is logical '0', the memory is not locked by the microsequencer and the FLC can access the RAM. At the same time reading the register an internal lock flag is set preventing the MS from accessing the RAM. This lock state is active as long as the FLC is resetting MEM_LOCK. This is done by writing to it, whereas the data is 'don't care'.

If the value is logical '1' the MS accesses the memory under lock condition. In this case the FLC has to poll the MEM-LOCK register until there is a logical '0'. Normally this condition is already complied with the next poll, because the microsequencer is much faster then the external MCU.

The MEM-LOCK bit is also part of the status register.

Under a FLC lock condition the microsequencer is not able to access the RAM so the maximum lock time is limited to two consecutive memory accesses (corresponds to 14 µs @12MBit/s). If this time is exceeded the response time of the MPI12x may be drastically longer (possibly it can happen that the chip cannot recover a telegram).

B-2.5 List of Active Stations

The List of Active Stations (LAS) contains a list of all active stations. For every active station on the serial bus there is a byte reserved within a 128 byte large memory. The last address (127) is not used though. The LAS contains two bits for frame count (FCB) and frame validity (FCV) for every station on the serial bus. Additionally there are two bits for maintenance of the LAS called *change* and *state*. A signal change indicates that the corresponding active station has changed its state on the serial bus i.e. it has entered or left the logical token ring. In addition the state of the active station is signaled with the state bit. If the station is not present or has left the ring, state = 0, whereas if the station is in the logical token ring state = 1. All other bits are set to zero. The organization of the LAS is shown in Figure B-6. If the chip changes the state of a station, an interrupt is generated. The FLC has to look in the LAS which entry has changed and clear the changed bit.

The LAS can be addressed via base pointer and offset or with a physical address. To decide whether the state of a station has changed, the FLC has to scan for a set changed bit and clear this bit if necessary. **Clearing the changed bit has to be done under LOCK condition for data consistency**. Because of scanning the LAS for a changed state bit takes a long time, the maximum bus lock time must not be exceeded, in order not to disable the microsequencer for too long time. It is recommended to scan the LAS without setting a LOCK condition and to clear the changed bit with a read-modify-write access under LOCK condition afterwards.

RAM Address	Bit Position								Station Number
	7	6	5	4	3	2	1	0	
020_H	0	0	FCB	FCV	change	state	0	0	00_H
	...								
$09E_H$	0	0	FCB	FCV	change	state	0	0	$7E_H$

Change Bit: 0 the state of the station has not changed.
1 the state of the station has changed.

State Bit: 0 station is not in the logical token ring / has left the logical token ring.
1 station is in logical token ring.

Figure B-6: Structure of LAS

B-2.6 Service Access Point List

The data transfer service is handled with Service Access Points (SAP) within the chip. Every bus member can have up to 64 SAPs (0 to 63) at the same time. The communication between bus members is handled exclusively via SAPs.

For efficiency reasons it is possible to transfer data without SAPs. The data transport services are handled via the default SAP then.

Every service access point (even the default SAP) has special entries in the SAP list. If the MPI12x receives a telegram to a non existing SAP the answer will be *no service activated* (SD1 response).

Default SAP	Control Byte	8 bit
	Request-SA	8 bit
	Confirmation-PTR	8 bit
SAP[0]	Control Byte	8 bit
	Request-SA	8 bit
	Confirmation-PTR	8 bit
...		
SAP[63]	Control Byte	8 bit
	Request-SA	8 bit
	Confirmation-PTR	8 bit

Figure B-7: Structure of SAP-List

Organization of the Control byte:

Bit Position							
7	6	5	4	3	2	1	0
SAP LOCKED	RS	UE	RR	IN USE	BUFFER AVAILABLE		

SAP-LOCKED	At the moment the SAP does not accept data. If the MPI12x receives data for this SAP it sets the event flag 'user error' (UE) and answers with SD1-User-Error.
RS	Event flag for 'no service activated'. The MPI12x sets this flag if SAP-MAX is exceeded or the check of Request-SA was negative (Request-SA = $7F_H$ => the SAP is/was inactive or Request-SA differs from received SA). In either case the MPI12x answers with SD1-No-Service-Activated
UE	Event flag for 'user error'. This bit is set if the SAP is/was locked. The MPI12x answers with SD1-User-Error.
RR	Event flag for 'no resource'. This bit is set when the telegram header is received and buffer available is zero, which means that there is no resource for the telegram or the queue is full. In either case the MPI12x answers with SD1-No-Resource.
IN-USE	This bit is set by the MPI12x if the header of a telegram was received without errors. It is cleared if an indication is executed (valid or invalid). If the FLC wants to put a new reply-block into the list the FLC has to wait until this bit is cleared. Only then the FLC is allowed to change the reply update pointer (under LOCK condition).
BUFFER AVAILABLE	This three bit value specifies how many buffers are available for this SAP. The FLC increments this value after making available a resource for this SAP whereas the chip decrements this value after having received an indication. When receiving an indication the chip checks if there is a resource available. If there is no more resource available the event-flag „No Resource„ (RR) is set and the chip responds with a negative acknowledge message (No Resource, SD1)

Figure B-8: Organization of the Control Byte

Generally the control byte **must be written under LOCK condition**, even if only the value of buffer_available should be incremented. Since that is a time consuming process the increment of that value could done by the chip itself. Therefore the FLC has to write the value FF_H to the control byte and the value of buffer_available will be incremented by one.

Every indication with a supported service generates an interrupt and the write-pointer is set to the next free segment. An unsupported service does not generate an interrupt, but a correct response for the polling master is generated in order to keep the system in a defined state.

B-2.7 Request- and Indication Queues

In the request queues the FLC puts data transfer requests. Thereby there are two priorities of the queues. High priority requests are put into the request high queue whereas low priority requests are put into the request low queue. The MPI12x always scans the request high queue first for any requests. If there are requests in both lists the requests in the request high queue are transmitted first and afterwards the low priority requests are sent. Independent of the target rotation time there will be at least one high priority request per token rotation (if one is present).

If the FLC has put one or more requests in a request queue, the MPI12x processes one after the other, whereas the data is fetched from the request blocks of the queue, converted and transferred in the appropriate format for the serial bus. If response data is expected the FLC has to take care that there is a resource for the response data (a response buffer in the response block area).

After the MPI12x has **sent** a SRD request (it expects an answer) it fetches the response buffer pointer from the response header in the request block. If the response telegram was free of errors it writes the data in the response buffer. Additionally the response state is written to the response header, even if there was only a positive or negative acknowledge. If only a short acknowledge was received, the status is only written in the request header. If there is no response buffer available the status is also only written into the request header. If the request is completely processed the MPI12x confirms the request by setting the read pointer to the next request block and generating and interrupt, except the collect bit in the FC byte is set.

If the remote station does not answer or the answer has errors the MPI12x generates a repetition. If even the last repetition was ineffective, the request is confirmed with NA (no access), that is NA is written to the response status. Additionally an interrupt is generated for the executed confirmation.

If the FLC wants to cancel a request from the queue it can lock the request lists (‘Set_Lock_Request = 1‘in Mode Register 1). The MPI12x confirms this directly with an interrupt (‘Request Inhibit‘) if it is in state Not-Hold-Token. If it is in state Hold-Token the 'lock request list' is checked before a request is processed and confirmed then. So a current request is processed until passed before confirming the 'lock request list'. If the bit is set the MS will not process any request and pass the token to the next station of the bus. A request will only be processed again, after resetting this bit.

If the MPI12x **receives** a telegram the telegram header is written into the indication queue (this is possible because one segment has to be free). If there is at least one additional segment empty, the reception is continued and data is written to the next segment of the queue. This procedure is

done for every additional segment needed to store the telegram data. If the queue is full, the MPI12x receives the data and checks for errors and HD4 security but does not write any data to the memory (with this behavior it is ensured that a queued indication with toggled FCB/FCV is executed in any case).

Data for the confirmation telegram has to be made available by the FLC within the confirmation blocks. If confirmation data is requested the MPI12x fetches the pointer for the confirmation from the corresponding SAP and transmits the data from the provided confirmation buffer. If the indication is completely processed, the MPI12x confirms this by writing the status to the response header (valid indication), setting the write pointer to the next empty segment and generating an interrupt.

How often the provided data in the reply buffer is sent can be specified by the FLC with a bit in the conf-status. If 'resp status' (bit 5) is set the confirmation buffer is only sent once (single update reply), if it is cleared the confirmation buffer is sent every indication again (multiple update buffer).

Request Block	
administrative data (2 byte)	user data (not changed by chip)
response header (2 byte)	resp-buf-ptr resp-status
request header (6 byte)	req-data-length rem-adr loc-adr fc-code rem-sap loc-sap
request buffer (max. 246 byte)	request data (net)

Indication Block	
confirmation header (2 byte)	conf-buf-ptr ind-status
indication header (6 byte)	ind-data-length rem-adr loc-adr fc-code rem-sap loc-sap
indication buffer (max. 246 byte)	indication data (net)

Response Buffer	
byte 0	resp-buf/data-length
byte 1	resp-status
byte 2 ...	data

Confirmation Buffer	
byte 0	conf-data-length
byte 1	conf-status (08_H low priority $0A_H$ high priority)
byte 2 ...	

Figure B-9: Organization of the Blocks & Buffers

If a request without SAPs is added to a request queue the corresponding entries are dummy entries. The minimum length of a request header is always 10 byte, so the write pointer must be placed at least 16 byte further. The same rule counts for the indication queue. If the indication telegram contains no SAP entries, the MPI12x writes FF_H to the SAP entries.

B-2.7.1 Request Block

Figure B-10 shows the organization of a request block. It consists of administrative data, response header and request header with the subsequent request buffer.

Administrative Data	Administrative data are not changed by the chip.

Response Header	In the response header there are entries for the response buffer pointer as well as the response status.
resp-buf-ptr	A pointer to the response buffer. The buffer has to be provided by the FLC. If the value is 00h, no buffer is available.
resp-status	During the confirmation the chip writes the response status into this cell. The allowed codes are shown below.

function	code
acknowledge positive	$00xx0000_B$
acknowledge negative (UE – SAP locked)	$00xx0001_B$
acknowledge negative (RR – no resource)	$00xx0010_B$
acknowledge negative (RS – no service activated)	$00xx0011_B$
response FDL/FMA 1/2 data low (& send data ok)	$00xx1000_B$
response FDL/FMA 1/2 data high (& send data ok)	$00xx1010_B$
short acknowledge	10000000_B
NA – no access	10011111_B
NADT – double token	10101111_B
NAB – response buffer too small	10111111_B

with 'xx' as followed:

type of station	Code
passive station	00_B
active station, ready to enter logical token ring	10_B
active station in logical token ring	11_B

Response Buffer	The response buffer is in the area for the reply-on-request blocks and contains the response buffer length, the response status and the data (net) of the response telegram.
resp-buf/data-length	The FLC writes here the length of the response buffer. If the MPI12x receives a response and the response buffer is too small or is not existent, the confirmation will be NAB. After receiving the telegram the MPI12x overwrites the value with the real length of the received data (net).
resp-status	The response status is taken from the FC byte of the telegram (see above). The difference here is, that MPI12x generated codes are not written here (they are written to the response header).

Request Header	The request header contains all important information for the generation of the request telegram.
req-data-length	This value specifies the length of the data in the request buffer (net, 0 to 244 bytes with SAPs, 0 to 246 bytes without SAPs).
rem-adr	This is the address of the remote station. There are 127 possible stations (address 0 to 126). The address 127 is reserved for broadcast and multicast where all or a group of stations are addressed. When using a SAP extension the MSB of this field has to be set to '1'.
loc-adr	This is the address of the local station. This value need not to be specified, because it is generated from the TS-ADDR-REG parameter. If a SAP extension is used, the MSB of this field has to be set to '1'.
fc-code	This value specifies the function code of the request telegram. the following codes are possible:

function	code
request FDL-status with reply	x9 H
send data with no acknowledge low	x4 H
send data with no acknowledge high	x6 H
send data with acknowledge low	x3 H
send data with acknowledge high	x5 H
send and request data low	xC H
send and request data high	xD H

In the upper bits of the function code there are additionally functions defined:

bit 7 : collect	If this bit is set no interrupt is generated when confirming the request (in this way it is possible to send more requests as 'packet').
bit 6 : force-pass-token	If this bit is set the MPI12x passes the token immediately after confirming the request independent of the remaining token hold time.

rem-sap	This is the service access point of the remote station. There are 64 SAPs possible at all. This field is only valid if the extension bit in 'rem-adr' is set (the upper two bits of rem-sap have to be set to '0').
loc-sap	Specifies the service access point of the local station. This field is only valid if the extension bit in 'loc-adr' is set (the upper two bit of loc-sap have to be set to '0').

Figure B-10: Structure of the Request Block

B-2.7.2 Indication Block

Request and indication blocks are organized principally in the same way, but the administrative data is missing. The only difference is the status byte in the response header. In the indication block there is the indication status, whereas in the request buffer there is the responder status. If there is a reply update buffer, the MPI12x fetches the reply update pointer from the SAP list and writes it to the response header as resp-buf-ptr.

During an indication telegram the MPI12x checks the telegram data with the parameterized values form the SAP list.

Confirmation Header	In the response header there are entries for the response buffer pointer as well as the response status.
conf-buf-ptr	This pointer points to the separate confirmation buffer (in the confirmation block region).
ind-status	The MPI12x writes 00_H for a valid indication and CF_H for an invalid indication.

Confirmation Buffer	The confirmation buffer is placed in the area for reply on indication blocks. It contains the response buffer length, the response status and the net data length of the response telegram.
conf-data-length	The FLC puts the length of the confirmation buffer here.
conf-status	The FLC puts the status for the confirmation here. The following codes are allowed:

function	code
response FDL/FMA 1/2 data low (& send data ok)	$00x01000_B$
response FDL/FMA 1/2 data high (& send data ok)	$00x01010_B$

with the special encoding for bit 5:

single confirmation	If this bit is set, a provided confirmation telegram is sent once only. Otherwise the MPI12x sends the confirmation telegram every time it receives an indication telegram.

Indication Header	The indication header contains the information about the received indication telegram.
ind-data-length	This value specifies the length of the received data in the indication buffer (net, 0 to 244 bytes with SAPs, 0 to 246 bytes without SAPs).
rem-adr	The received source address (SA) is stored here.
loc-adr	The received destination address (DA) is stored here.
fc-code	This value specifies the function code (FC) of the indication telegram. Only the lower 4 bits are stored (the values can be seen in the description for the request block).

rem-sap	The received service access point of the remote station is stored here (DSAP). This field is only valid, if the extension bit (bit 8) in rem-adr is set.
loc-sap	The received service access point for the local station is stored here (SSAP). This field is only valid, if the extension bit (bit 8) in loc-adr is set.

Figure B-11: Structure of the Indication Block

B-3 ASIC Interface

B-3.1 Mode Registers

In the MPI12x parameter bits that access the controller directly or which the controller directly sets are combined in three mode registers (0, 1 and 2).

B-3.1.1 Mode Register 0

Setting parameters for Mode Register 0 may take place in the Offline state only (for example, after power-on). The MPI12x must not exit the off-line state until Mode Register 0, Mode Register 2, all Control and Organizational Parameters are loaded (Go_Online = 1 in Mode-Register 1).

Address	Bit Position								Description
	7	6	5	4	3	2	1	0	
004_H (Intel)	Monitor	Early_Rdy	INT_Pol	Sep_INT	HSA		Pas/Act	No_CTRL	Mode Reg 0 7 .. 0 See below for coding
					1	0			

	Mode Register 0, physical Address 004_H (Intel):
bit 7	**Monitor:** 0 = Normal operation mode. 1 = Monitor mode.
bit 6	**Early_Rdy**: Early Ready 0 = Normal Ready: Ready is generated when data is valid (write) or when data has been accepted (read). 1 = Ready is generated one clock pulse earlier
bit 5	**INT_Pol:** Interrupt Polarity 0 = The interrupt output is low-active. 1 = The interrupt output is high-active.
bit 4	**Sep_INT:** Separate Interrupts 0 = All interrupts are on pin X/INTEV. 1 = There are two interrupt pins for different events: X/INTCI ➔ Confirmations / Indications. X/INTEV ➔ All other interrups.
bit 3,2	**HSA:** Highest Station Address. 00 = Highest Station Address is $0F_H$. 01 = Highest Station Address is $1F_H$. 10 = Highest Station Address is $3F_H$. 11 = Highest Station Address is $7F_H$.

bit 1	**Act/Pas:** Active / Passive Station 0 = Passive Station. 1 = Active Station.
bit 0	**No_CTRL:** No Start- / Stop bit Control 0 = Start and stop bit is checked by the UART. 1 = Start and stop bit are not checked.

Figure B-12: Coding of Mode Register 0

B-3.1.2 Mode Register 1

Some control bits must be changed during operation. These control bits are combined in Mode-Register 1. There are bits for setting and resetting an action. A logical ‘1’ must be written to the bit position to be set or reset.

For example, to set Lock_Request write a '1' to bit 5, in order to reset this bit, write a '1' to bit 7.

Address	Bit Position								Description
	7	6	5	4	3	2	1	0	
006_H	Clr_Lock_ Request	Clr_SD4_ Filter	Operation_ Mode	Set_Lock_ Request	Go_Offline	Set_SD4_ Filter	EOI	Go_Online	Mode Reg 1 7..0

	Mode Register 1, Set, physical Address 006_H:
bit 7	**Clr_Lock Request:** Clear FLC Lock Request 1 = The FLC resets the lock request for the request queues.
bit 6	**Clr_SD4_Filter:** Clear FLC Lock Request 1 = SD4 telegrams will be indicated (monitor mode only).
bit 5	**Operation_Mode:** Operation Mode 0 = Chip operates in MPI mode. 1 = Chip operates in HSUART mode.
Bit 4	**Set_Lock_Request:** Set FLC Lock Request 1 = The FLC sets a request to lock the request queues. The MPI12x acknowledges this with the ‘Request_Inhibit’ interrupt. The micro-sequencer will not access the request lists until the lock request is cleared.

bit 3	**Go_Offline:** Going into the offline state 1 = After the current request ends, MPI12x goes to the offline state and sets Go_Offline to log.'0' again.
bit 2	**Set_SD4_Filter:** Clear FLC Lock Request 1 = SD4 telegrams will be indicated (monitor mode only).
bit 1	**EOI:** End-of-Interrupt 1 = MPI12x disables the interrupt output and sets EOI to log.'0' again.
bit 0	**Go_Online:** Leave state Offline and go Online 1 = MPI12x exits offline and goes to passive-idle In addition timers are started and 'Go_Offline = 0' is set

Figure B-13: Coding of Mode Register 1

B-3.1.3 Mode Register 2

Setting parameters for Mode Register 2 may take place in the Offline State only, except for bits [7:6].

Address	Bit Position								Description
	7	6	5	4	3	2	1	0	
019 H	SD4_Filter	FDL_Filter	Dont_Indic_SC	En_DXB_Sub	En_DXB_Pub	TX_Off	EOI_Time	4kB_Mode	Mode Reg 2 7 .. 0

	Mode Register 2, physical Address 019 H:
bit 7	**SD4_Filter:** Token Telegram Filter 0 = SD4 telegrams are indicated (monitor mode only). 1 = SD4 telegrams are not indicated (monitor mode only).
bit 6	**FDL_Filter:** Request FDL Status Filter 0 = FDL telegrams are indicated (monitor mode only). 1 = FDL telegrams are not indicated (monitor mode only).
bit 5	**Dont_Indic_SC:** Do Not Indicate Short Acknowledge (SC) 0 = SC is indicated if Filter 1 is set (monitor mode only). 1 = SC is not indicated if Filter 1 is set (monitor mode only).

bit 4	**En_DXB_Sub:** Enable DXB Subscriber 0 = DXB subscriber support is disabled. 1 = DXB subscriber support is enabled. NOTE: There is no built-in DXB filter table, the publisher has to be checked by the FLC.
bit 3	**En_DXB_Pub:** Enable DXB Publisher 0 = DXB publisher support is disabled. 1 = DXB publisher support is enabled.
bit 2	**TX_Off:** Transmitter Off 0 = Transmitter is controlled by MAC. 1 = Transmitter is disabled.
bit 1	**EOI_Time:** End of Interrupt Time 0 = End of interrupt time is 1 µs. 1 = End of interrupt time is 1 ms.
bit 0	**4KB_Mode:** Size of internal RAM 0 = 2 KB RAM (default). 1 = 4 KB RAM

Figure B-14: Coding of Mode Register 2

B-3.2 Status Register

The status register shows the current MPI12x status and can be read only.

Address	Bit Position								Description
	7	6	5	4	3	2	1	0	
01E H (Intel)	Early_Ready	Recv_State	Idle-Mux		Hold_ Token	Not_Hold_ Token	Listen_ Token	Offline	Status-Reg 7..0 See below for coding
			1	0					

Status Register, Low-Byte, physical Address 01E H (Intel):	
bit 7	**Early_Ready:** Early Ready Generation 0 = Ready signal is generated when data is valid. 1 = Ready signal is generated on clock cycle before data is valid.
bit 6	**Recv_State:** State of the Receiver 0 = Receiver is disabled. 1 = Receiver is enabled.
bit 5,4	**Idle-Mux:** State of the Idle Multiplexer 00 = T_{SYN} selected. 01 = T_{RDY} selected. 10 = T_{ID1} selected. 11 = T_{ID2} selected.
bit 3	**Hold_Token:** Hold Token State 0 = MAC is not in 'Hold Token' state. 1 = MAC is in 'Hold Token' state.
bit 2	**Not_Hold_Token:** Not Hold Token State 0 = MAC is not in 'Not Hold Token' state. 1 = MAC is in 'Not Hold Token' state.
bit 1	**Listen_Token:** Listen Token State 0 = MAC is not in 'Listen Token' state. 1 = MAC is in 'Listen Token' state.
bit 0	**Offline:** Offline state 0 = MAC is not in 'Offline' state. 1 = MAC is in 'Offline' state.

Figure B-15: Status Register, Low-Byte

Address	Bit Position								Description
	15	14	13	12	11	10	9	8	
01F H (Intel)	Stn_Typ 1	Stn_Typ 0	Conf_Stored	Ind_Stored	Curr_Prio	Syni/Slot	GAP_Stored	Mem_Lock	Status-Reg 15..8 See below for coding

Status Register, High-Byte, physical Address 01F H (Intel):	
bit 15,14	**Stn_Type:** Station Type 00 = Passive station. 01 = Active station, not ready to enter token ring. 10 = Active station, ready to enter token ring. 11 = Active station, in ring.
bit 13	**Conf_Stored:** Confirmation Stored 0 = No confirmation latched. 1 = A confirmation is latched.
bit 12	**Ind_Stored:** Indication Stored 0 = No indication latched. 1 = An indication is latched.
bit 11	**Curr_Prio:** Current Request Priority 0 = The current processed request has low priority. 1 = The current processed request has high priority.
bit 10	**Syni/Slot:** Syni / Slot Timer State 0 = T_{SLOT} selected. 1 = T_{SYNI} selected.
bit 9	**GAP_Stored:** GAP Request Stored 0 = No GAP request latched. 1 = A GAP request has been latched.
bit 8	**Mem_Lock:** Memory Lock 0 = No memory lock set. 1 = The FLC has set a memory lock.

Figure B-16: Status Register, High-Byte

B-3.3 Interrupt Controller

The processor is informed about indication messages and various error events via the interrupt controller. Up to a total of 16 events are stored in the interrupt controller. The events are summed up to a common interrupt output (INTEV). Optionally a second interrupt output INTCI can be enabled in order to assert indications and confirmations separate from other events. The interrupt controller does not have a prioritization level and does not provide an interrupt vector (not 8259A compatible!).

The controller consists of an Interrupt Request Register (IRR), an Interrupt Mask Register (IMR), an Interrupt Register (IR) and an Interrupt Acknowledge Register (IAR).

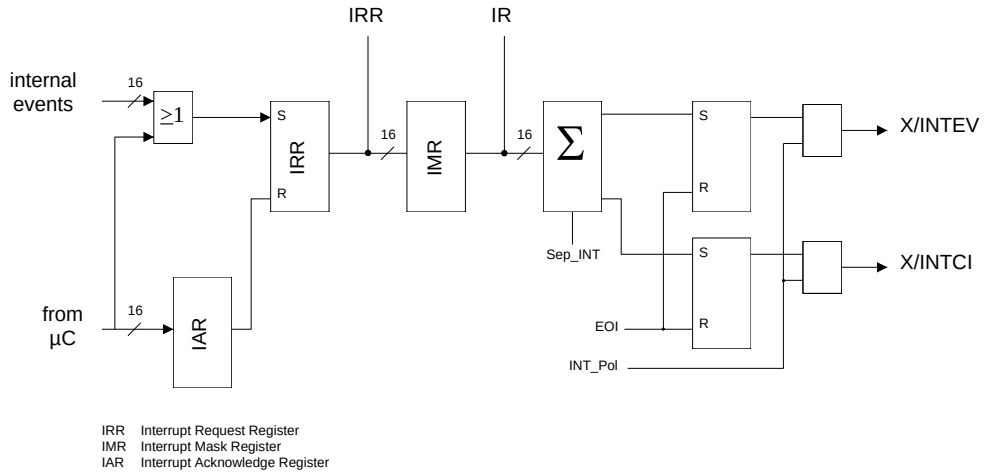


Figure B-17: Block Diagram of Interrupt Controller

Each event is stored in the IRR. Individual events can be suppressed via the IMR. The input in the IRR is independent of the interrupt masks. Events that are not masked in the IMR set the corresponding IR bit and generate an interrupt. Dependent on bit ‘Sep_INT’ in mode register 0 all interrupts are collected at pin X/INTEV or the additional interrupt pin X/INTCI is used. For debugging purposes the user can set each event in the IRR.

Each interrupt event that was processed by the microcontroller must be deleted via the IAR. A logical ‘1’ must be written on the specific bit position. If a new event and an acknowledge from the previous event are present at the IRR at the same time, the event remains stored. If the microcontroller enables a mask subsequently, it must be ensured that no prior IRR input is present. To be on the safe side, the position in the IRR must be deleted prior to the enabling of the mask.

Before leaving the interrupt routine, the microprocessor should set the ‘end of interrupt bit' (EOI = 1) in mode register 1. The interrupt output is switched to inactive with this edge change. If another event occurs, the interrupt output is not activated again until the interrupt inactive time (1 µs / 1 ms) ex-

pired. This interrupt inactive time can be configured via 'EOI_Time' in mode register 2. This allows entering the interrupt routine again when an edge-triggered interrupt input is used.
The polarity of the interrupt output is parameterized via the INT_POL mode bit in mode register 0. After hardware reset, the output is low-active.

B-3.3.1 Interrupt Request Register

Address	Bit Position								Description
	7	6	5	4	3	2	1	0	
01A$_H$ (Intel)	Rec_Frame_ Overflow	Request_ Inhibit	HSA_ Error	Syni_Error	Double_ Token	Timeout	Response_ Error	Go-Hold/ Not-Hold-Token	Int-Req-Reg 7 .. 0 See below for coding

	Interrupt-Request-Register, Low-Byte, physical Address 01A$_H$ (Intel):
bit 7	**Rec_Frame_Overflow:** The receiver FIFO had an overrun, which may be caused by a too long lock periode.
bit 6	**Request_Inhibit:** The FLC has set the 'Set_Lock_Request' bit in mode register 1. The MAC will not access the request lists.
bit 5	**HSA_Error:** The MAC has detected an active station outside the parameterized HSA. The MAC enters state 'Offline' If this happens two times successively.
bit 4	**Syni_Error:** The syni timer has expired. If the MAC is in state 'Listen_Token' it remains there. If it is in state 'Hold_Token' it will enter state 'Not_Hold_Token'.
bit 3	**Double_Token / Mon_Trig:** *Normal Mode*: The MAC has entered state 'Not_Hold_Token', because another station is sending. *Monitor Mode*: A trigger event has occurred.
bit 2	**Timeout / Timer Overflow:** *Normal Mode*: The timeout timer has expired. In passive stations this leads to no action. An active station will reinitialize the token ring. *Monitor Mode*: The 16 bit counter for the generating the timestamps had an overflow.

bit 1	**Response_Error / Mon_Full:** *Normal Mode*: The response was faulty during state 'Hold_Token' . The MPI12x will repeat the request if permissible. *Monitor Mode*: There is no free memory available; MPI12x has gone into state 'Offline'.
bit 0	**Go-Hold/Not-Hold-Token:** The MAC has entered / left state 'Hold_Token'. This event is generated every time the MAC enters / leaves state 'Hold_Token'

Figure B-18: Interrupt-Request-Register, Low-Byte

Address	Bit Position								Description
	15	14	13	12	11	10	9	8	
$01B_H$ (Intel)	IND	CONF	Mem_ Overflow	LAS_ Changed	MAC_Reset	Pass_ Token_Error	TS_ Addr_Error	LAS_ Useless	Int-Req-Reg 15 .. 8 See below for coding

	Interrupt Request Register, High-Byte, physical Address $01B_H$ (Intel):
bit 15	**IND / :** The MPI12x has processed an indication.
bit 14	**CONF:** The MPI12x has processed a confirmation.
bit 13	**Mem_Overflow:** The FLC addressed the memory beyond the physical limit. The write has not been processed in order not to overwrite the parameter area.
bit 12	**LAS_Changed:** The LAS has been changed. MPI12x has added/removed a station.
bit 11	**MAC_Reset:** The MAC has entered state 'Offline'.
bit 10	**Pass_Token_Error:** The token telegram was not read back correctly two times, while passing the token. The MAC enters state 'Offline' because this is a severe error (transmitter / receiver defective).

bit 9	**TS_Addr_Error:** The own address is already in the token ring.
bit 8	**LAS_Useless:** The token error counter has exceeded. The MAC enters state ‘Listen_Token’ and rebuilds the LAS.

Figure B-19: Interrupt Request Register, High-Byte

B-3.3.2 Interrupt Acknowledge / Mask Register

The other interrupt controller registers are assigned in the bit positions like the Interrupt Request Register.

Physical Address	Register		Reset state	Assignment
$01C_H$ / $01D_H$	Interrupt Register (IR)	Readable only	All bits cleared	
000_H / 001_H	Interrupt Mask Register (IMR)	Writeable, can be changed during operation	All bits set	1 = Mask is set and the interrupt is disabled 0 = Mask is cleared and the interrupt is enabled
002_H / 003_H	Interrupt Acknowledge Register (IAR)	Writeable, can be changed during operation	All bits cleared	1 = Interrupt is acknowledged and the IRR bit is cleared 0 = IRR bit remains unchanged

Figure B-20: Interrupt Acknowledge / Mask Register

B-3.4 Control Registers

Target Rotation Time (Physical Address 316_H, word)

$T_{TR}7$	$T_{TR}6$	$T_{TR}5$	$T_{TR}4$	$T_{TR}3$	$T_{TR}2$	$T_{TR}1$	$T_{TR}0$
$T_{TR}15$	$T_{TR}14$	$T_{TR}13$	$T_{TR}12$	$T_{TR}11$	$T_{TR}10$	$T_{TR}9$	$T_{TR}8$

This time defines the rotation time for the token. One unit corresponds to 256 T_{BIT}.

Slot Time (Physical Address 306_H, word)

$T_{SL}7$	$T_{SL}6$	$T_{SL}5$	$T_{SL}4$	$T_{SL}3$	$T_{SL}2$	$T_{SL}1$	$T_{SL}0$
-	-	$T_{SL}13$	$T_{SL}12$	$T_{SL}11$	$T_{SL}10$	$T_{SL}9$	$T_{SL}8$

The slot time defines the maximum time the chip waits for an answer. The unit for the slot time is T_{BIT}. Because of there are two slot times, the slot time has to be specified by the following rule:

$$T_{SL} = max\ (T_{SL1}, T_{SL2}).$$

Idle Time 1 (Physical Address 308_H, word)

$T_{ID1}7$	$T_{ID1}6$	$T_{ID1}5$	$T_{ID1}4$	$T_{ID1}3$	$T_{ID1}2$	$T_{ID1}1$	$T_{ID1}0$
-	-	-	-	-	-	$T_{ID1}9$	$T_{ID1}8$

The chip waits for idle time 1 after a request which has to be acknowledged. The unit for the idle time 1 is T_{BIT}.

Idle Time 2 (Physical Address $30A_H$, word)

$T_{ID2}7$	$T_{ID2}6$	$T_{ID2}5$	$T_{ID2}4$	$T_{ID2}3$	$T_{ID2}2$	$T_{ID2}1$	$T_{ID2}0$
-	-	-	-	-	-	$T_{ID2}9$	$T_{ID2}8$

The chip waits for idle time 2 after a request without acknowledge (SDN). The unit for the idle time 2 is T_{BIT}.

Ready Time (Physical Address $30E_H$, byte)

$T_{RDY}7$	$T_{RDY}6$	$T_{RDY}5$	$T_{RDY}4$	$T_{RDY}3$	$T_{RDY}2$	$T_{RDY}1$	$T_{RDY}0$

The ready time defines the time before a response is sent. The unit for the ready time is T_{BIT}.

Quiet Time (Physical Address $30F_H$, byte)

$T_{QUI}7$	$T_{QUI}6$	$T_{QUI}5$	$T_{QUI}4$	$T_{QUI}3$	$T_{QUI}2$	$T_{QUI}1$	$T_{QUI}0$

The receiver is disabled for the quiet time after the last byte of a telegram has been received. The unit for the ready time is T_{BIT}.

Baud Rate (Physical Address $30C_H$, word)

BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0
-	-	-	-	-	BR10	BR9	BR8

The value defines the divider ratio for the desired baud rate. It is calculated as followed:

$$BR[10:0] = \frac{48000000}{4 * baudrate} - 1$$

The following table shows a list of BR values for common baud rates:

baud rate	BR value
12 MBaud	0
6 MBaud	1
3 MBaud	3
1.5 MBaud	7
187.5 KBaud	63
19.2 KBaud	623

Pointer (Physical Addresses 310_H to 315_H, bytes)

Base Pointer	The base pointer represents the segment address of the FLC when accessing the RAM between 200_H to $2FF_H$ (segment & offset addressing mode).
Req-H-BPtr Req-L-BPtr Resp-BPtr Ind-BPtr Conf-BPtr	These pointers define the start segment of the corresponding list. The start of the next queue is the end of the current queue. The order of the lists can not be changed.

B-3.5 Organizational Parameters

Retries (RAM Address 006_H, byte)

-	-	-	-	RET3	RET2	RET1	RET0

The parameter defines the number of retries for an incorrect request.

This Station Address (RAM Address 007_H, byte)

-	TS6	TS5	TS4	TS3	TS2	TS1	TS0

The parameter defines the own station address. Permitted values are between 0 and 126.

GAP Update Time (RAM Address 008_H, byte)

$T_{GUD}7$	$T_{GUD}6$	$T_{GUD}5$	$T_{GUD}4$	$T_{GUD}3$	$T_{GUD}2$	$T_{GUD}1$	$T_{GUD}0$

The parameter defines the spare time between a complete GAP cycle has been processed and the beginning of the new GAP cycle. The unit for the gap update time is token cycles.

Token Error Limit (RAM Address 009_H, byte)

TEL7	TEL6	TEL5	TEL4	TEL3	TEL2	TEL1	TEL0

The parameter specifies the maximum amount of token errors per 256 consecutive token cycles. When the limit is reached, the station enters state LISTEN_TOKEN and rebuilds the LAS (only active stations).

Pointer (RAM Address 000_H to 005_H, bytes)

Req-H-WP	Write pointer for the request high queue. This pointer is managed by the FLC.
Req-H-RP	Read pointer for the request high queue. This pointer is managed by the chip.
Req-L-WP	Write pointer for the request low queue. This pointer is managed by the FLC.
Req-L-RP	Read pointer for the request low queue. This pointer is managed by the chip.
Ind-WP	Write pointer for the indication queue. This parameter is managed by the chip.
Ind-RP	Read pointer for the indication queue. This parameter is managed by the FLC.

A request will be processed, if WP <> RP. After the FLC has linked a new request to a request list, it has to reposition the pointer to the segment behind the linked request. More requests are linked one after the other, but a request block must always start at a segment border. After receiving the token the chip will check the read and write pointer of the two request lists (if bit Set_Req_Lock = 0, Mode Register 1). If RP <> WP than the chip starts sending the request(s), until there is no more request (RP = WP) or there is no more token hold time left.

For indications handling is a bit different. WP <> RP has the meaning that there is at least one segment for storing an indication. If there is no free segment anymore (WP = RP) the telegram will be received and checked but not stored to RAM. The chip will generate a negative acknowledge if required (RR, No Resource).

Note:
The response and indication buffers are not organized as queues. These buffers have to be managed by the FLC.

B-4 Supported Services

B-4.1 Request FDL Status

This service is used for getting information whether a remote station is present (and in what state it is) or if it is absent. If the FLC polls every possible station (DA = 0 to 126) it is possible to build a temporary life list of all accessible stations.
This service does not support a repetition of the telegram. If there is no (or a faulty) answer the chip writes „NA„ (no access) into the responder status field. If the station is present it answers with its station type and this value is written into the responder status field. The chip confirms the response by generating an interrupt (confirmation).

Request

SD1	DA	SA	FC 49_H	FCS	ED

Response

SD1	DA	SA	FC $x0_H$	FCS	ED

x 0000_B passive
0010_B active, ready to enter ring
0011_B active, in ring

B-4.2 Send Data with No Acknowledge (SDN)

This service allows sending data to more than one station (multicast) or to all stations (broadcast) at the same time. An interrupt is generated after the telegram was sent, but no confirmation if the telegram was received correctly at the remote stations.
The remote station indicates a correct received indication by generating an interrupt (indication). If the remote FLC

Request

SD1	DA	SA	FC $44/46_H$	FCS	ED

SD2	LE	LEr	SD2	DA	SA	FC $44/46_H$	DU	FCS	ED

SD3	DA	SA	FC $44/46_H$	DATA 0..7	FCS	ED

Note:
The MPI12x can not generate SD3 telegrams, but is able to receive them.

B-4.3 Send Data with Acknowledge (SDA)

This service allows the FLC sending data to a single remote station. The remote station acknowledges the reception with a short acknowledge. A negative acknowledge is generated if the remote station has no resource for that indication or the SAP is locked or not activated. The request will be repeated if the remote station does not respond.

The FLC of the local station will see an interrupt (confirmation) after the response has been received or the last repetition was sent without receiving a response.

The remote station will generate an interrupt (indication) after a new request or after the next token telegram.

Request

SD1	DA	SA	FC y3/y5 $_H$	FCS	ED

SD2	LE	LEr	SD2	DA	SA	FC y3/y5 $_H$	DU	FCS	ED

SD3	DA	SA	FC y3/y5 $_H$	DATA 0..7	FCS	ED

y 0110_B new synchronization
0101_B new request or repetition
0111_B

Response

SC

SD1	DA	SA	FC	FCS	ED

FC $x1_H$ user error
$x2_H$ no resource
$x3_H$ no service activated

B-4.4 Send and Request Data with Reply (SRD)

This service allows the FLC sending data to a single remote station and to receive data from the remote station immediately.

The local station will receive either the requested data or a negative acknowledge indicating the problem. If there are no data to send in the remote station, the remote station will send a short acknowledge. The request will be repeated if a transmission error occurred. The FLC will see an interrupt (confirmation) after the response has been received or the last repetition was sent without receiving a response.

The remote station will generate an interrupt (indication) after a new request or after the next token telegram.

Request

SD1	DA	SA	FC y3/y5 $_H$	FCS	ED

SD2	LE	LEr	SD2	DA	SA	FC y3/y5 $_H$	DU	FCS	ED

SD3	DA	SA	FC y3/y5 $_H$	DATA 0..7	FCS	ED

y 0110_B new synchronization
0101_B new request or repetition
0111_B

Response

SC

SD1	DA	SA	FC	FCS	ED

FC $x1_H$ user error
$x2_H$ no resource
$x3_H$ no service activated

SD2	LE	LEr	SD2	DA	SA	FC x8/xA $_H$	DU	FCS	ED

SD3	DA	SA	FC x8/xA $_H$	DATA 0..7	FCS	ED

Notes:

B-5 Monitor Mode

B-5.1 Overview

The Monitor Mode can be used for debug purposes but cannot replace a bus analyzer, because error states are not sufficient recovered (e.g. spike detection). It is intended for diagnosis of the data transfer so that any error on the protocol level can be recovered.

Because the monitor is completely different to the normal mode, the memory table is different form the one in normal mode, too. The memory table can be seen in the figure below.

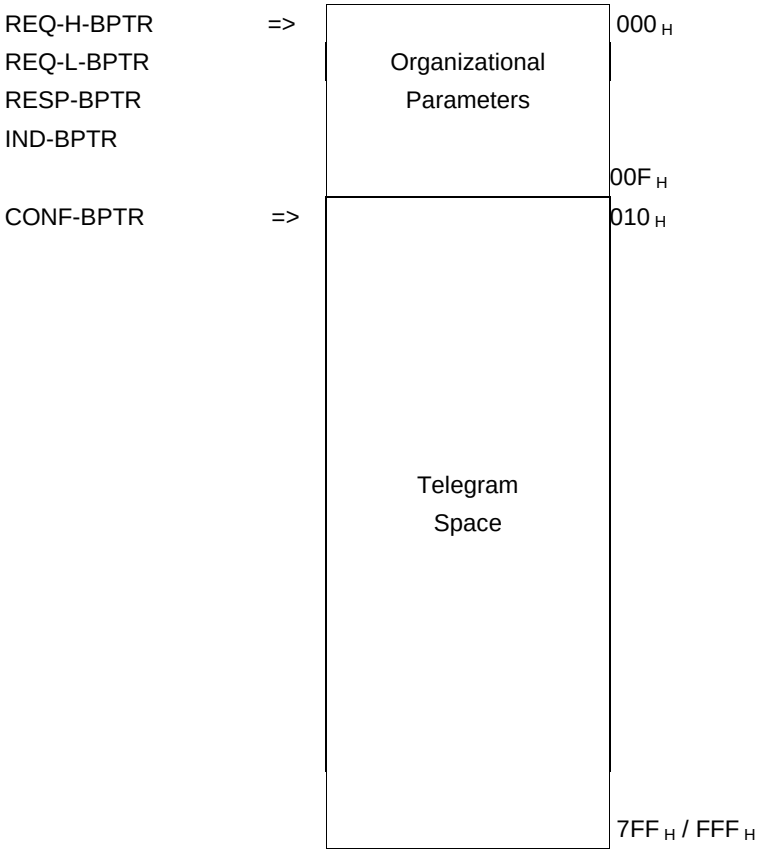


Figure B-21: Memory Table

B-5.2 Organizational Parameters

The Organizational Parameters are located in the RAM area starting with address 000_H. The parameters can be read and written, but must only be changed in state 'Offline', except for the pointers.

RAM Address Intel	Mot.	Name	Bit No.	Description
000_H		Write Pointer	7..0	Read/Write Pointer
001_H		Read Pointer	7..0	
002_H		Filter 1	7..0	Filter
003_H		Filter 2	7..0	
004_H		Trigger 1 Offset	7..0	Trigger 1
005_H		Trigger 1 Mask	7..0	
006_H		Trigger 1 Value	7..0	
007_H		Trigger 2 Offset	7..0	Trigger 2
008_H		Trigger 2 Mask	7..0	
009_H		Trigger 2 Value	7..0	
$00A_H$		Trigger Type	1..0	Trigger Type 0 start condition bit0: trigger 1 1 stop condition bit1: trigger 2

Figure B-22: Assignment of the Organizational Parameters in Monitor Mode

Note:
The Control Registers (register area) must also be initialized.

B-5.3 Monitor Block

Every telegram is stored in a separate Monitor Block. The table below shows in which order the data is stored.

$+00_H$	Timestamp	delay time
$+02_H$	Block Length	block_length_real
$+03_H$	Indicator	status
$+04_H$	Frame Buffer	block_length expected
$+05_H$		da
$+06_H$		sa
$+07_H$		fc
$+08_H$		dsap
$+09_H$		ssap
$+0A_H$		data [0]
$+0B_H$		data [1]
...		...
$+FF_H$		data [245]

delay_time Timestamp from the 16 bit delay timer.

block_length_real Real length of the frame buffer. If a telegram was received correctly, block_length_real and block_length_expected have the same size.

Status Status of the received telegram

00H ok
80H short acknowledge
90H parity error
91H protocol fault
92H FCS/ED fault

Any other not possible.

block_length_expected Contains the expected length of the frame buffer including block_lentgh_expected. The telegram is only stored to the point at which an error occurred. DSAP and SSAP always count to the block_length (real & expected) even the default SAP is used for data transfer. The following values for block_length_expected can occur:

telegram frame	block_length_expected
wrong start delimiter, telegram invalid	0
short acknowledge (SC)	1
token telegram (SD4)	3
SD1 frame	4
SD2 frame	6 to 252
SD3 frame	12 to 14
uart error, fcs/ed error	up to the error

Da Contains the destination address.

Sa Contains the source address.

Fc Contains the function code.

Dsap Contains the destination service access point.

Ssap Contains the source service access point.

data[0..n] Contains the received data.

Figure B-23: Structure of the Monitor Block

B-5.4 Filter Conditions

Up to two address filters can be parameterized, if telegrams with a defined address relationship should be recorded (Filter 1, Filter 2). If filter 1 is set, every telegram **from** and **to** that station is recorded, if both filter are set every telegram **between** the two specified stations is recorded. Thereby the address extension bit is not checked and broadcasts will not be recorded. In order to record also broadcast telegrams, the MSB has to be set.

Filter Value	Effect
00_H .. $7E_H$	Telegrams with/without SAPs, no broadcasts
$7F_H$	Broadcasts only
80_H .. FE_H	Telegrams with/without SAPs and broadcasts
FF_H	Filter off

Figure B-24: Filter Conditions

Note:
Filter 2 can not be set without setting filter 1, otherwise it is ignored.

B-5.5 Trigger Conditions

Up to two trigger conditions can be defined, which compare a given value with an entry in the frame buffer of a Monitor-Block. Every trigger condition consists of three entries (offset, mask and value) in the organizational parameter area, furthermore the type of the trigger (start resp. stop condition) has to specified. Both trigger can compare any byte within the Frame Buffer. The offset is added to the current Monitor Block and points to the byte which should be compared. The mask selects the bits for the comparison. The so generated byte is compared with the compare value. A trigger event has occurred, if the following condition has been met:

byte (offset) AND mask = value

A MON_TRIG interrupt occurs when at least one of the two trigger conditions is met. A trigger condition is enabled if the offset points to a segment unequal zero.

B-6 Hardware Interface

B-6.1 Processor Bus Interface

B-6.1.1 Overview

The MPI12x has a parallel 8-bit data interface with a 10-bit address bus. The MPI12x supports all 8-bit processors and microcontrollers based on the 80C51/52 (80C32) from Intel, the Motorola HC11 family, as well as 8- and 16-bit processors or microcontrollers from the Siemens 80C166 family, X86 from Intel and the HC16 and HC916 family from Motorola.
The Bus Interface Unit (BIU) and the Dual Port RAM Controller (DPC) that controls accesses to the internal RAM belong to the processor interface of the MPI12x.

The MPI12x is to be operated with an external 48 MHz crystal oscillator. The integrated clock divider provides a slower clock (divided by 2 or 4) at pin CLKOUT. This allows the direct connection of a slower controller without additional expenditures in a low-cost application.

B-6.1.2 Bus Interface Unit

The Bus Interface Unit (BIU) is the interface to the connected processor/microcontroller. This is a synchronous or asynchronous 8-bit data interface with a 10-bit address bus. The interface is configurable via 2 pins (TYP, MODE). The connected processor family (bus control signals such as XWR, XRD, or R_W and the data format) is specified with the TYP pin. Synchronous or asynchronous bus timing is specified with the MODE pin.

TYP	MODE	Processor Interface Mode
0	1	Synchronous Intel mode
0	0	Asynchronous Intel mode
1	0	Asynchronous Motorola mode
1	1	Synchronous Motorola mode

Figure B-25: Configuration of the Processor Interface

Examples of various Intel system configurations are given in subsequent sections. The internal address latch and the integrated decoder must be used in the synchronous Intel mode. One figure shows the minimum configuration of a system with the MPI12x, where the chip is connected to an EPROM version of the controller. Only a clock generator is necessary as an additional device in this configuration. If a controller is to be used without an integrated program memory, the addresses must be latched for the external memory.

Note:
If the MPI12x is connected to an 80286 or similar processor, it must be taken into consideration that the processor carries out word accesses. That is, either a ‘swapper’ is necessary that switches the characters out of the MPI12x at the correct byte position of the 16-bit data bus during reading or the least significant address bit is not connected and the 80286 must read word accesses and evaluate only the lower byte.

Name	Input/ Output	Type	Comments
DB(7..0)	I/O	Tristate	High-resistance during RESET
AB(9..0)	I		Address Bus
MODE	I		Configuration: sync / async Interface
XWR E_CLOCK	I		Intel: Write Sync. Motorola: E-Clock
XRD R_W	I		Intel: Read Motorola: Read/Write
XCS	I		Chip Select
ALE AS	I		Intel: Address Latch Enable Motorola: Address Strobe
DIVIDER	I		Scaling factor 2/4 for CLKOUT
INTEV	O	Push / Pull	Polarity programmable
INTCI	O	Push / Pull	Polarity programmable
XRDY XDTACK	O	Tristate	Intel: READY-Signal Motorola: DTACK-Signal
CLK	I	Schmitt-Trigger	48 MHz
TYP	I		Configuration: Intel/Motorola format
CLKOUT	O	Push / Pull	24/12 MHz
RESET	I	Schmitt-Trigger	Minimum of 4 clock cycles

Figure B-26: Microprocessor Bus Signals in MPI Operation Mode

Synchronous Intel Mode

In this mode Intel CPUs like 80C51/52/32 and compatible processor series from several manufacturers can be used.

- Synchronous bus timing without evaluation of the XREADY signal
- 8-bit multiplexed bus: ADB7..0
- The lower address bits AB7..0 are stored with the ALE signal in an internal address latch.
- The internal CS decoder is activated. MPI12x generates its own CS signal from the address lines AB9..2. The MPI12x selects the relevant address window from the AB1..0 signals.

Asynchronous Intel Mode

In this mode various 16-/8-bit microcontroller series like Intel's x86, Siemens 80C16x or compatible series from other manufacturers can be used.

- Asynchronous bus timing with evaluation of the XREADY signal
- 8-bit non-multiplexed bus: DB7..0, AB9..0
- The internal MPI12x address decoder is disabled; the XCS input is used instead.
- External address decoding is always necessary.
- External chip select logic is necessary if not present in the microcontroller

Asynchronous Motorola Mode

Motorola microcontrollers like the HC16 and HC916 can be used in this mode. When using HC11 types with a multiplexed bus the address signals AB7..0 must be generated from the DB7..0 signals externally.

- Asynchronous bus timing with evaluation of the XREADY signal
- 8-bit non-multiplexed bus: DB7..0, AB9..0
- The internal MPI12x address decoder is disabled; the XCS input is used instead.
- Chip select logic is available and programmable in all microcontrollers mentioned above.

Synchronous Motorola Mode

Motorola microcontrollers like the HC11 types K, N, M, F1 or the HC16- and HC916 types with programmable E_Clock timing can be used in this mode. When using HC11 types with a multiplexed bus the address signals AB7..0 must be generated from the DB7..0 signals externally.

- Synchronous bus timing without evaluation of the XREADY signal
- 8-bit non-multiplexed bus: DB7..0, AB9..0
- The internal MPI12x address decoder is disabled; the XCS input is used instead.
- For microcontrollers with chip select logic (K, F1, HC16 and HC916), the chip select signals are programmable regarding address range, priority, polarity and window width in the write cycle or read cycle.
- For microcontrollers without chip select logic (N and M) and others, an external chip select logic is required. This means additional hardware and a fixed assignment.
- If the CPU is clocked by the MPI12x, the output clock pulse (CLKOUT) must be 4 times larger than the E_Clock. That is, a clock pulse signal must be present at the CLK input that is at least 10 times larger than the desired system clock pulse (E_Clock). The Divider-Pin must be connected to <log>0 (divider 4). This results in an E_Clock of 3 MHz.

B-6.1.3 Application Examples (Principles)

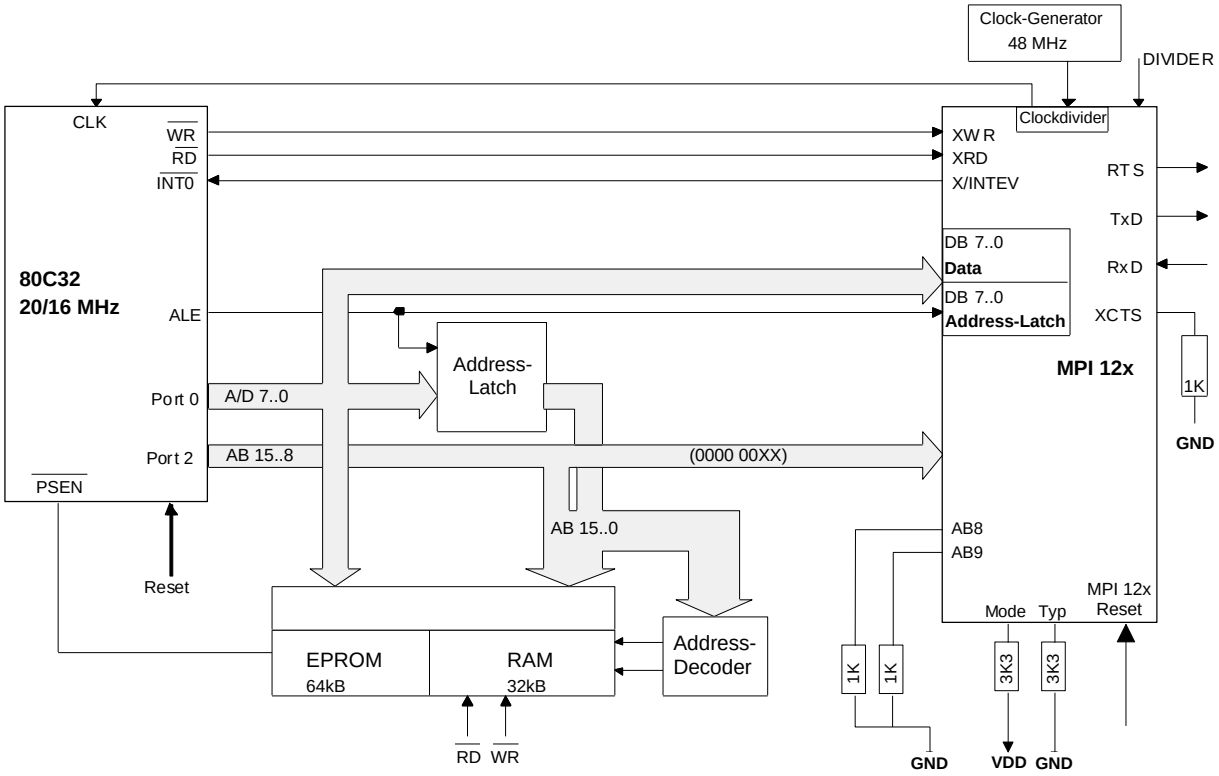


Figure B-27: 80C32 System with External Memory

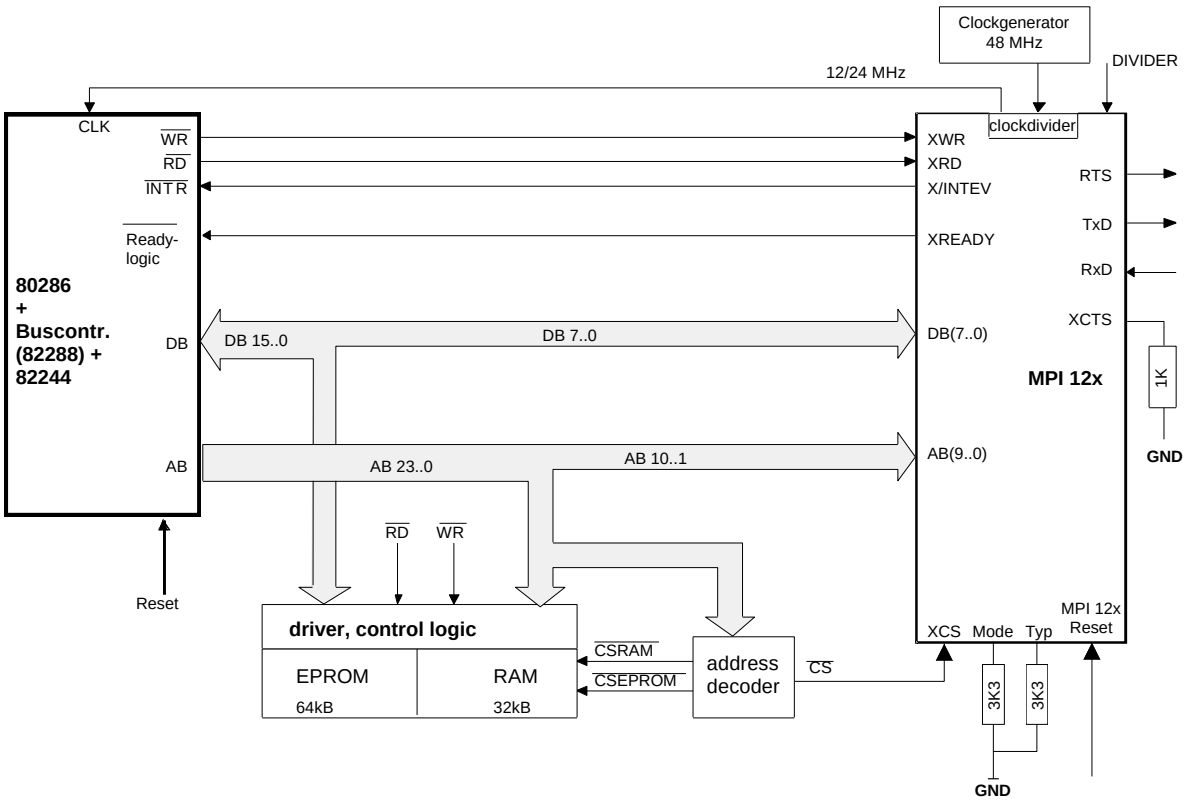


Figure B-28: 80286 System (X86 Mode)

B-6.1.4 Application with 80C32

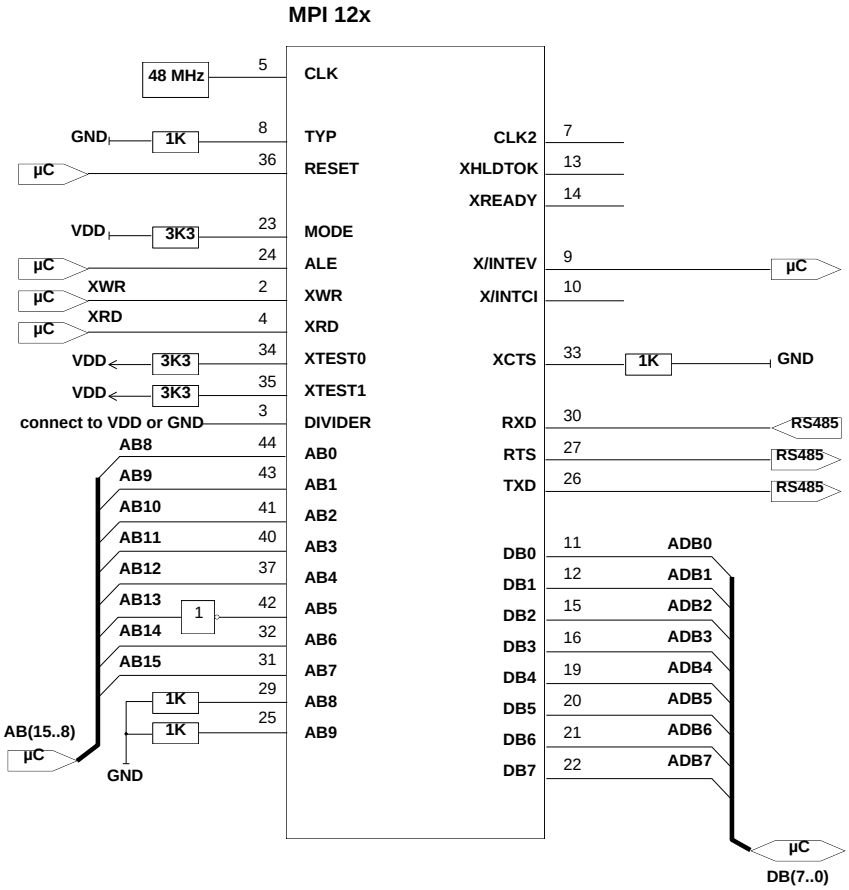


Figure B-29: 80C32 Application

The internal chipselect is activated when the address inputs AB[9..2] of the MPI12x are set to <log>0.
In the example above the start address of the MPI12x is set to 0x2000.

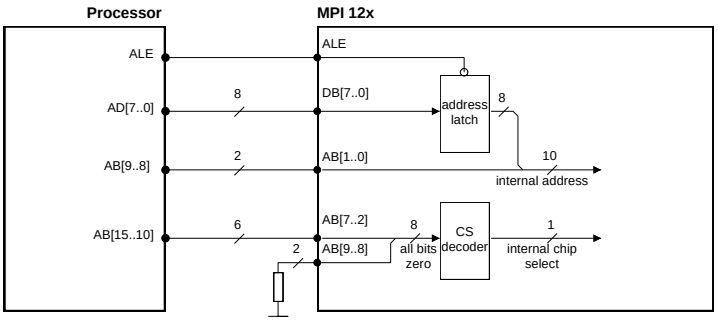


Figure B-30: Internal Chipselect Generation in Synchronous Intel Mode

B-6.1.5 Application with 80C165

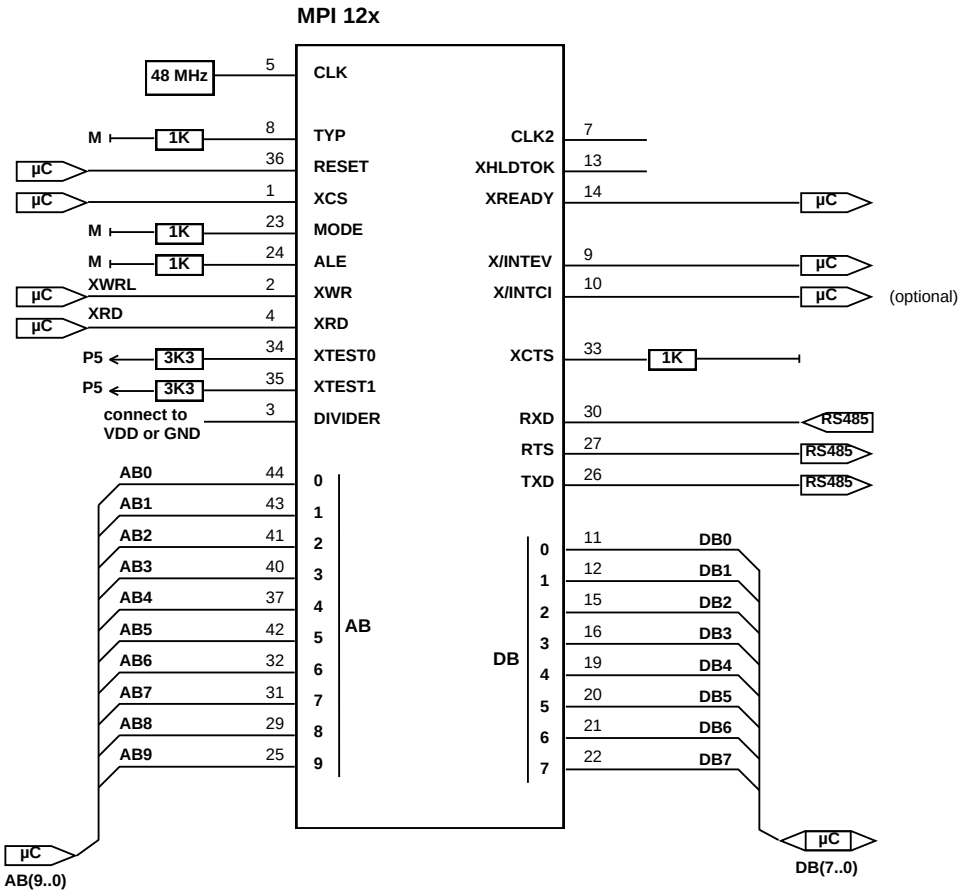


Figure B-31: 80C165 Application

B-6.2 Dual Port RAM Controller

The internal 4 KB RAM of the MPI12x is a single-port RAM. An integrated Dual-Port RAM controller, however, permits an almost simultaneous access of both ports (bus interface and microsequencer interface). When there is a simultaneous access of both ports, the bus interface has higher priority. This guarantees the shortest possible access time. If the MPI12x is connected to a microcontroller with an asynchronous interface, the controller can evaluate the Ready signal.

B-6.3 UART

The transmitter converts the parallel data structure into a serial data flow. Signal Request-to-Send (RTS) is generated before the first character. The XCTS input is available for connecting a modem. After RTS active, the transmitter must hold back the first telegram character until the modem activates XCTS. XCTS is checked again after each character.

The receiver converts the serial data flow into the parallel data structure and scans the serial data flow with the four-fold transmission speed. Stop bit testing can be switched off for test purposes (‘No_Ctrl = 1' in mode register 0). One requirement of the PROFIBUS protocol is that no slip is permitted between the telegram characters. The MPI12x transmitter ensures that this specification is maintained.

The synchronization of the receiver starts with the falling edge of the start bit. The start bit is checked again in the middle of the bit-time for low level. The data bits, the parity and the stop bit are also scanned in the middle of the bit-time. To compensate for the synchronization error, a repeater generates a ±25% distortion of the stop bit at a four-fold scan rate. In this case the MPI12x should be parameterized with 'No_Ctrl = 1' (in mode register 0) in order to increase the permissible distortion of the stop bit.

B-6.4 PROFIBUS Pin Assignment

The data transmission is performed in RS485 operating mode (i.e., physical RS485). The MPI12x is connected via the following signals to the galvanically isolated interface drivers.

Signal Name	Input/Output	Function
RTS	Output	Request to send
TXD	Output	Sending data
RXD	Input	Receiving data

Figure B-32: PROFIBUS Signals

The PROFIBUS interface is a 9-way, sub D, plug connector with the following pin assignment.

Pin 1 - Free
Pin 2 - Free
Pin 3 - B line
Pin 4 - Request to send (RTS)
Pin 5 - Ground 5V (M 5)
Pin 6 - Potential 5V (floating P5)
Pin 7 - Free
Pin 8 - A line
Pin 9 - Free

The cable shielding must be connected to the plug connector housing.
The free pins are described as optional in EN 50170 Vol. 2. If used, they should conform to the specifications in DIN 19245 Part 3.

CAUTION:
The pin names A and B on the plug connector refer to the signal names in the RS485 standard and not the pin names of driver ICs.

Keep the wires from driver to connector as short as possible.

B-6.5 Example of RS485 Interface

To minimize the capacity of the bus lines the user should avoid additional capacities. The typical capacity of a bus station should be 15...25 pF.

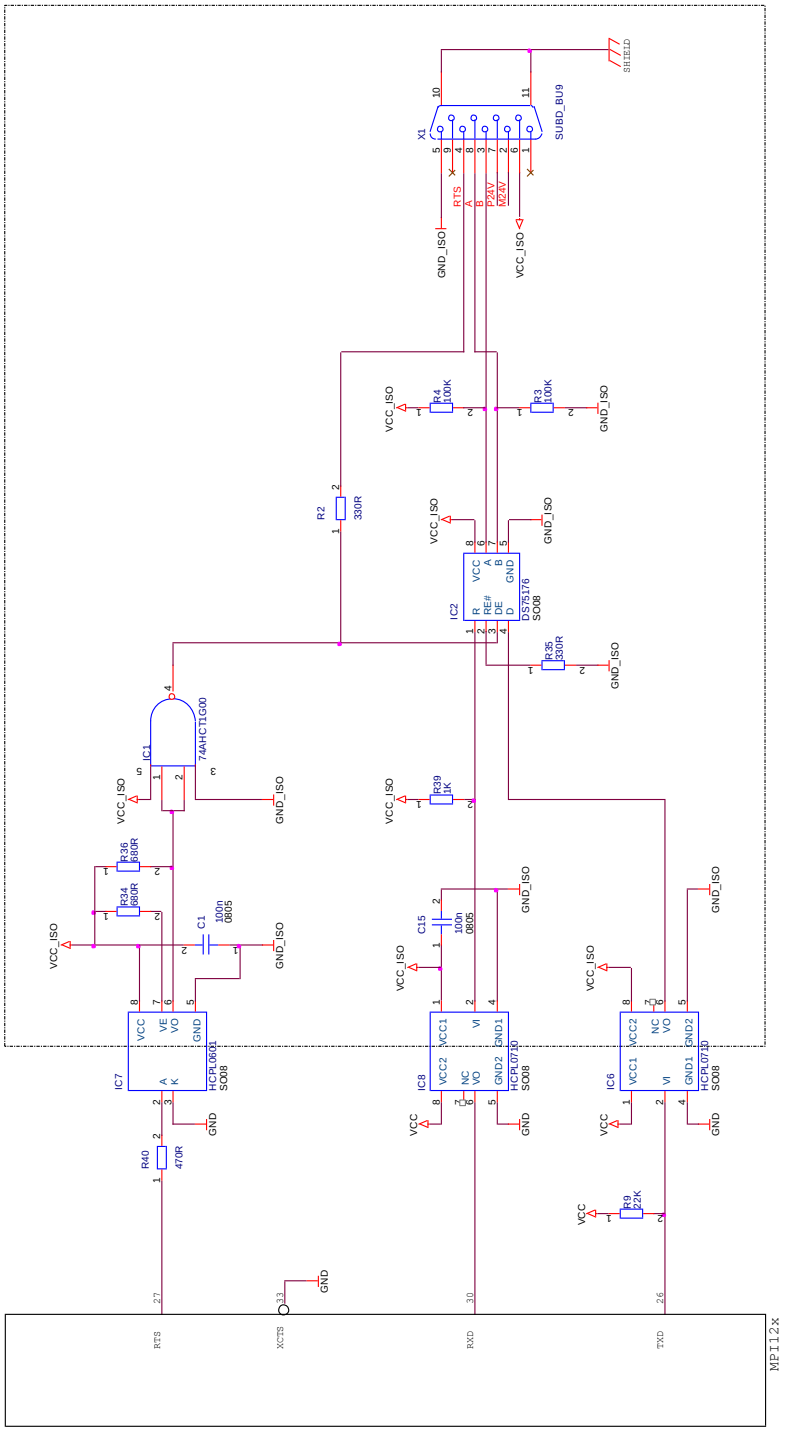


Figure B-33: Example of RS485 Interface

Notes:

Part C USART Operation Mode

The USART Operation Mode is not released but will be available in a later version of the MPI12x chip.

Notes:

Part D VPC3+ Operation Mode

D-1 Overview

The MPI12x incorporates the VPC3+ PROFIBUS-DP slave core. The VPC3+ handles the physical layer 1 and the data link layer 2 of the ISO/OSI-reference-model excluding the analog RS485 drivers. It supports DP-V0, DP-V1 and DP-V2 slave functionality.

The **integrated 4K Byte Dual-Port-RAM** serves as an interface between the VPC3+ and the software/application. In case of using 2K Byte the entire memory is divided into 256 segments, with 8 bytes each. Otherwise in the 4K Byte mode the segment base addresses starts at multiple of 16. Addressing by the user is done directly however, the internal Micro Sequencer (MS) addresses the RAM by means of the so-called base-pointer. The base-pointer can be positioned at the beginning of a segment in the memory. Therefore, all buffers must be located at the beginning of a segment.

If the VPC3+ carries out a DP communication it automatically sets up all DP-SAPs. The telegram information is made available to the user in separate data buffers (for example, parameter and configuration data). Three buffers are provided for data communication (three for output data and three for input data). One buffer is always available for communication therefore no resource problems can occur. For optimal diagnosis support, the VPC3+ offers two Diagnosis-Buffers. The user enters the updated diagnosis data into these buffers. One Diagnosis-Buffer is always assigned to the VPC3+.

The **Bus Interface Unit** is a configurable synchronous/asynchronous 8- bit interface for various Intel and Motorola microcontrollers/processors. The user can directly access the internal 2K/4K Byte RAM or the parameter and control registers via the 11/12-bit address bus.

Procedure-specific parameters (Station_Address, control bits, etc.) must be written into the **Parameter Registers** and the **Mode Registers** after reset.

The MAC status can be observed at any time in the **Status Register**.

Various events (e.g. indications, error events, etc.) are asserted via the **Interrupt Controller**. These events can be individually enabled by the mask register. Interrupt acknowledgement takes place by means of the acknowledge register. The VPC3+ has a single interrupt output.

The integrated **Watchdog Timer** is operated in three different states: BAUD_SEARCH, BAUD_CONTROL and DP_CONTROL.

The **Micro Sequencer** (MS) controls the entire process. It contains the DP-Slave state machine (DP_SM).

The integrated **4K Byte RAM** which operates as a Dual-Port-RAM contains procedure-specific parameters (buffer pointers, buffer lengths, Station-Address, etc.) and the data buffers.

In the **UART**, the parallel data flow is converted into the serial data flow and vice-versa. The VPC3+ is capable of automatically identifying all PROFIBUS-DP baud rates (9.6 Kbit/s - 12 Mbit/s).

The **Idle Timer** directly observes the bus timing on the serial bus line.

D-2 Memory Organization

D-2.1 Overview

The internal Control Parameters are located in the first 21 addresses. The latches/registers either come from the internal controller or influence the controller. Certain cells are read- or write-only. The internal working cells, which are not accessible by the user, are located in RAM at the same address locations.

The Organizational Parameters are located in RAM beginning with address 16H. The entire buffer structure (for the DP-SAPs) is based on these parameters. In addition, general parameter data (Station_Address, Ident_Number, etc.) and status information (Global_Control command, etc.) are also stored in these cells.
Corresponding to the parameter setting of the Organizational Parameters, the user-generated buffers are located beginning with address 40_H. All buffers or lists must begin at segment addresses (8 bytes segmentation for 2K Byte mode, 16 bytes segmentation for 4K Byte mode).

<table>
<tr><th>Address</th><th colspan="2">Function</th></tr>
<tr><td>000_H
:
015_H</td><td>Control Parameters
(latches/registers) (21 bytes)</td><td>Internal working cells</td></tr>
<tr><td>016_H
:
$03F_H$</td><td colspan="2">Organizational Parameters (42 bytes)</td></tr>
<tr><td>040_H

:
:
:
:

$7FF_H$ (FFF_H)</td><td>DP-buffers:

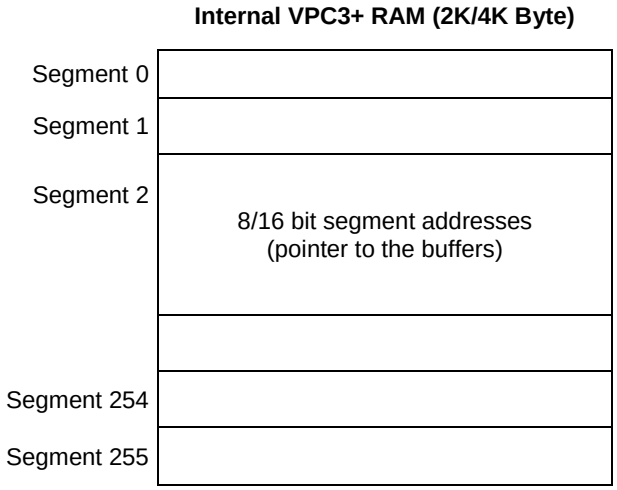
DP-V1-buffers:

DP-V2-buffers:

</td><td>Data in (3)*
Data out (3)**
Diagnosis data(2)
Parameter data (1)
Configuration data (2)
Auxiliary buffers (2)
SSA-buffer (1)
SAP-List (1)
Indication / Response buffers ***
DXB out (3)****
DXB-buffers (2)
CS-buffer (1)</td></tr>
</table>

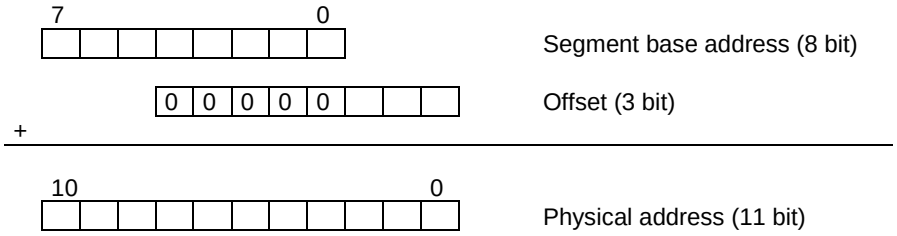
Figure D-1: Memory Table

* Data in means input data from DP-Slave to DP-Master
** Data out means output data from DP-Master to DP-Slave
*** number of buffers depends on the entries in the SAP-List
**** DXB out means input data from another DP-Slave (slave-to-slave communication)

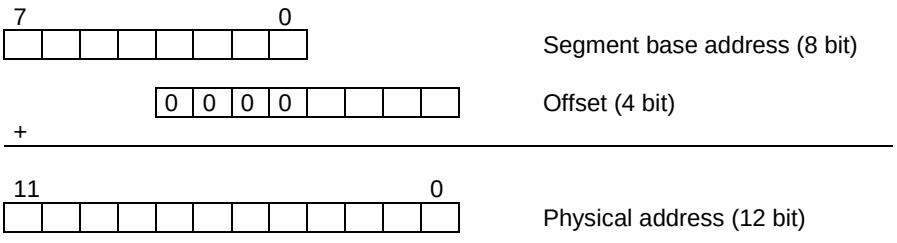


Building of the physical buffer address:

2K Byte mode:



4K Byte mode:



D-2.2 Control Parameters (Latches/Registers)

These cells can be either read-only or write-only. In the Motorola Mode the VPC3+ carries out 'address swapping' for an access to the address locations 00_H – 07_H (word registers). That is, the VPC3+ internally generates an even address from an odd address and vice-versa.

Address Intel	Address Mot.	Name	Bit No.	Description (Read Access!)
00_H	01_H	Int-Req-Reg	7..0	Interrupt Controller Register
01_H	00_H	Int-Req-Reg	15..8	
02_H	03_H	Int–Reg	7..0	
03_H	02_H	Int–Reg	15..8	
04_H	05_H	Status-Reg	7..0	Status Register
05_H	04_H	Status-Reg	15..8	
06_H	07_H	Mode-Reg 0	7..0	Mode Register 0
07_H	06_H	Mode-Reg 0	15..8	
08_H		Din_Buffer_SM	7..0	Buffer assignment of the DP_Din_Buffer_State_Machine
09_H		New_Din_Buffer_Cmd	1..0	The user makes a new DP Din_Buf available in the N state.
$0A_H$		Dout_Buffer_SM	7..0	Buffer assignment of the DP_Dout_Buffer_State_Machine
$0B_H$		Next_Dout_Buffer_Cmd	3..0	The user fetches the last DP Dout_Buf from the N state
$0C_H$		Diag_Buffer_SM	3..0	Buffer assignment for the DP_Diag_Buffer_State_Machine
$0D_H$		New_Diag_Buffer_Cmd	1..0	The user makes a new DP Diag_Buf available to the VPC3+.
$0E_H$		User_Prm_Data_Okay	1..0	The user positively acknowledges the user parameter setting data of a Set_(Ext_)Prm telegram.
$0F_H$		User_Prm_Data_Not_Okay	1..0	The user negatively acknowledges the user parameter setting data of a Set_(Ext_)Prm telegram.
10_H		User_Cfg_Data_Okay	1..0	The user positively acknowledges the configuration data of a Chk_Cfg telegram.
11_H		User_Cfg_Data_Not_Okay	1..0	The user negatively acknowledges the configuration data of a Chk_Cfg telegram.
12_H		DXBout_Buffer_SM	7..0	Buffer assignment of the DXBout_Buffer_State_Machine
13_H		Next_DXBout_Buffer_Cmd	2..0	The user fetches the last DXBout_Buf from the N state
14_H		SSA_Buffer_Free_Cmd		The user has fetched the data from the SSA_Buf and enables the buffer again.
15_H		Mode-Reg 1	7..0	

Figure D-2: Assignment of the Internal Parameter-Latches for READ

Address Intel	Address Mot.	Name	Bit No.	Description (Write Access!)
00_H	01_H	Int-Req-Reg	7..0	Interrupt-Controller-Register
01_H	00_H	Int-Req_Reg	15..8	
02_H	03_H	Int-Ack-Reg	7..0	
03_H	02_H	Int-Ack-Reg	15..8	
04_H	05_H	Int–Mask-Reg	7..0	
05_H	04_H	Int–Mask-Reg	15..8	
06_H	07_H	Mode-Reg0	7..0	Setting parameters for individual bits
07_H	06_H	Mode-Reg0	15..8	
08_H		Mode-Reg1-S	7..0	
09_H		Mode-Reg1-R	7..0	
$0A_H$		WD_BAUD_CONTROL_Val	7..0	Square-root value for baud rate monitoring
$0B_H$		$minT_{SDR}$_Val	7..0	$minT_{SDR}$ time
$0C_H$		Mode-Reg2	7..0	Mode Register 2
$0D_H$		Sync_PW_Reg	7..0	Sync Pulse Width Register
$0E_H$		Control_Command_Reg	7..0	Control_Command value for comparison with SYNCH telegram
$0F_H$		Group_Select_Reg	7..0	Group_Select value for comparison with SYNCH telegram
10_H		Reserved		
11_H				
12_H				
13_H				
14_H				
15_H				

Figure D-3: Assignment of the Internal Parameter-Latches for WRITE

D-2.3 Organizational Parameters (RAM)

The user stores the organizational parameters in the RAM under the specified addresses. These parameters are read- and writable.

Address Intel	Address Mot.	Name	Bit No.	Description
16_H		R_TS_Adr		Station_Address of the VPC3+
17_H		SAP_List_Ptr		Pointer to a RAM address which is preset with FFh or to SAP-List
18_H	19_H	R_User_WD_Value	7..0	In DP_Mode an internal 16-bit watchdog timer monitors the user.
19_H	18_H	R_User_WD_Value	15..8	
$1A_H$		R_Len_Dout_Buf		Length of the 3 Dout_Buf
$1B_H$		R_Dout_Buf_Ptr1		Segment base address of Dout_Buf 1
$1C_H$		R_Dout_Buf_Ptr2		Segment base address of Dout_Buf 2
$1D_H$		R_Dout_Buf_Ptr3		Segment base address of Dout_Buf 3
$1E_H$		R_Len_Din_Buf		Length of the 3 Din_Buf
$1F_H$		R_Din_Buf_Ptr1		Segment base address of Din_Buf 1
20_H		R_Din_Buf_Ptr2		Segment base address of Din_Buf 2
21_H		R_Din_Buf_Ptr3		Segment base address of Din_Buf 3
22_H		R_Len_DXBout_Buf		Length of the 3 DXBout_Buf
23_H		R_DXBout_Buf_Ptr1		Segment base address of DXBout_Buf 1
24_H		R_Len Diag_Buf1		Length of Diag_Buf 1
25_H		R_Len Diag_Buf2		Length of Diag_Buf 2
26_H		R_Diag_Buf_Ptr1		Segment base address of Diag_Buf 1
27_H		R_Diag_Buf_Ptr2		Segment base address of Diag_Buf 2
28_H		R_Len_Cntrl_Buf1		Length of Aux_Buf 1 and the corresponding control buffer, for example SSA_Buf, Prm_Buf, Cfg_Buf, Read_Cfg_Buf
29_H		R_Len_Cntrl_Buf2		Length of Aux_Buf 2 and the corresponding control buffer, for example SSA_Buf, Prm_Buf, Cfg_Buf, Read_Cfg_Buf
$2A_H$		R_Aux_Buf_Sel		Bit array; defines the assignment of the Aux_Buf 1 and 2 to the control buffers SSA_Buf, Prm_Buf, Cfg_Buf
$2B_H$		R_Aux_Buf_Ptr1		Segment base address of Aux_Buf 1
$2C_H$		R_Aux_Buf_Ptr2		Segment base address of Aux_Buf 2
$2D_H$		R_Len_SSA_Data		Length of the input data in the Set_Slave_Add_Buf
$2E_H$		R_SSA_Buf_Ptr		Segment base address of the Set_Slave_Add_Buf
$2F_H$		R_Len_Prm_Data		Length of the input data in the Prm_Buf

Address Intel	Address Mot.	Name	Bit No.	Significance
30_H		R_Prm_Buf_Ptr		Segment base address of the Prm_Buf
31_H		R_Len_Cfg_Data		Length of the input data in the Cfg_Buf
32_H		R_Cfg_Buf_Ptr		Segment base address of the Cfg_Buf
33_H		R_Len_Read_Cfg_Data		Length of the input data in the Read_Cfg_Buf
34_H		R_Read_Cfg_Buf_Ptr		Segment base address of the Read_Cfg_Buf
35_H		R_Len_DXB_Link_Buf		Length of the DXB_Linktable
36_H		R_DXB_Link_Buf_Ptr		Segment base address of the DXB_Link_Buf
37_H		R_Len_DXB_Status_Buf		Length of the DXB_Status
38_H		R_DXB_Status_Buf_Ptr		Segment base address of the DXB_Status_Buf
39_H		R_Real_No_Add_Change		This parameter specifies whether the Station_Address may be changed again later.
$3A_H$		R_Ident_Low		The user sets the parameters for the Ident_Number_Low value.
$3B_H$		R_Ident_High		The user sets the parameters for the Ident_Number_High value.
$3C_H$		R_GC_Command		The Control_Command of Global_Control last received
$3D_H$		R_Len_Spec_Prm_Buf		If parameters are set for the Spec_Prm_Buffer_Mode (see Mode Register 0), this cell defines the length of the Prm_Buf.
$3E_H$		R_DXBout_Buf_Ptr2		Segment base address of DXBout_Buf 2
$3F_H$		R_DXBout_Buf_Ptr3		Segment base address of DXBout_Buf 3

Figure D-4: Assignment of the Organizational Parameters

D-3 ASIC Interface

D-3.1 Mode Registers

In the VPC3+ parameter bits that access the controller directly or which the controller directly sets are combined in three mode registers (0, 1 and 2).

D-3.1.1 Mode Register 0

Setting parameters for Mode Register 0 may take place in the Offline state only (for example, after power-on). The VPC3+ may not exit the Offline state until Mode Register 0, all Control and Organizational Parameters are loaded (START_VPC3 = 1 in Mode Register 1).

Address	Bit Position								Description
	7	6	5	4	3	2	1	0	
06_H (Intel)	Freeze_ Supported	Sync_ Supported	Early_Rdy	Int_Pol	CS_ Supported	WD_Base	Dis_Stop_ Control	Dis_Start_ Control	Mode Reg 0 7 .. 0 See below for coding

Address	Bit Position								Description
	15	14	13	12	11	10	9	8	
07_H (Intel)	Reserved	PrmCmd_ Supported	Spec_Clear_ Mode *	Spec_Prm_ Buf_Mode **	Set_Ext_Prm _Supported	User_Time_ Base	EOI_Time_ Base	DP_Mode	Mode Reg 0 15 .. 8 See below for coding

* If Spec_Clear_Mode = 1 (Fail Safe Mode) the VPC3+ will accept Data_Exchange telegrams without any output data (data unit length = 0) in the state DATA-EXCH. The reaction to the outputs can be parameterized in the parameterization telegram.

** When a large number of parameters have to be transmitted from the DP-Master to the DP-Slave, the Aux-Buffer 1/2 must have the same length as the Parameter-Buffer. Sometimes this could reach the limit of the available memory in the VPC3+. When Spec_Prm_Buf_Mode = 1 the parameterization data are processed directly in this special buffer and the Aux-Buffers can be held compact.

	Mode Register 0, Low-Byte, Address 06_H (Intel):
bit 7	**Freeze_Supported**: Freeze_Mode support 0 = Freeze_Mode is not supported 1 = Freeze_Mode is supported
bit 6	**Sync_Supported:** Sync_Mode support 0 = Sync_Mode is not supported 1 = Sync_Mode is supported
bit 5	**Early_Rdy**: Early Ready 0 = Normal Ready: Ready is generated when data is valid (write) or when data has been accepted (read) 1 = Ready is generated one clock pulse earlier
bit 4	**Int_Pol:** Interrupt Polarity 0 = the interrupt output is low-active 1 = the interrupt output is high-active
bit 3	**CS_Supported:** Enable Clock Synchronization 0 = Clock Synchronization is disabled (default) 1 = Clock Synchronization is enabled
bit 2	**WD_Base:** Watchdog Time Base 0 = Watchdog time base is 10 ms (default state) 1 = Watchdog time base is 1 ms
bit 1	**Dis_Stop_Control:** Disable Stop bit Control 0 = Stop bit monitoring is enabled 1 = Stop bit monitoring is switched off Set_Prm telegram overwrites this memory cell in the DP_Mode. (Refer to the user specific data.)
bit 0	**Dis_Start_Control:** Disable Start bit Control 0 = monitoring the following Start bit is enabled 1 = monitoring the following Start bit is switched off Set_Prm telegram overwrites this memory cell in the DP_Mode. (Refer to the user specific data.)

Figure D-5: Coding of Mode Register 0, Low-Byte

	Mode Register 0, High-Byte, Address 07_H (Intel):
bit 15	**Reserved**
bit 14	**PrmCmd_Supported**: PrmCmd support for redundancy 0 = PrmCmd is not supported 1 = PrmCmd is supported
bit 13	**Spec_Clear_Mode:** Special Clear Mode (Fail Safe Mode) 0 = no Special Clear Mode 1 = Special Clear Mode; VPC3+ will accept data telegrams with data unit = 0
bit 12	**Spec_Prm_Buf_Mode:** Special-Parameter-Buffer Mode 0 = No Special-Parameter-Buffer 1 = Special-Parameter-Buffer Mode; Parameterization data will be stored directly in the Special-Parameter-Buffer
bit 11	**Set_Ext_Prm_Supported:** Set_Ext_Prm telegram support 0 = SAP 53 is deactivated 1 = SAP 53 is activated
bit 10	**User_Time_Base:** Time Base of the cyclical User_Time_Clock interrupt 0 = the User_Time_Clock interrupt occurs every 1 ms 1 = the User_Time_Clock interrupt occurs every 10 ms
bit 9	**EOI_Time_Base:** End-of-Interrupt Time Base 0 = the interrupt inactive time is at least 1 µs long 1 = the interrupt inactive time is at least 1 ms long
bit 8	**DP_Mode:** DP_Mode enable 0 = DP_Mode is disabled 1 = DP_Mode is enabled; VPC3+ sets up all DP_SAPs (default configuration!)

Figure D-6: Coding of Mode Register 0, High-Byte

D-3.1.2 Mode Register 1

Some control bits must be changed during operation. These control bits are combined in Mode Register 1 and can be set independently of each other (Mode-Reg_1_S) or can be reset independently of each other (Mode-Reg_1_R). Separate addresses are used for setting and resetting. A ‘1’ must be written to the bit position to be set or reset.

For example, to set START_VPC3 write a '1' to address 08_H, in order to reset this bit, write a '1' to address 09_H.

Address	Bit Position								Description
	7	6	5	4	3	2	1	0	
08_H	Reserved	Reserved	Res_ User_WD	En_Change_ Cfg_Buffer	User_LEAVE-MASTER	Go_Offline	EOI	START_ VPC3	Mode-Reg_1_S 7..0
09_H	Reserved	Reserved	Res_ User_WD	En_Change_ Cfg_Buffer	User_LEAVE-MASTER	Go_Offline	EOI	START_ VPC3	Mode-Reg_1_R 7..0 See below for coding

	Mode Register 1, Set, Address 08_H:
bit 7	**Reserved**
bit 6	**Reserved**
bit 5	**Res_User_WD:** Resetting the User_WD_Timer 1 = VPC3+ sets the User_WD_Timer to the parameterized value User_WD_Value. After this action, VPC3+ sets Res_User_WD to '0'.
bit 4	**En_Change_Cfg_Buffer:** Enabling buffer exchange (Config-Buffer for Read_Config-Buffer) 0 = With User_Cfg_Data_Okay_Cmd, the Config-Buffer may not be exchanged for the Read_Config-Buffer. 1 = With User_Cfg_Data_Okay_Cmd, the Config-Buffer must be exchanged for the Read_Config-Buffer.
bit 3	**User_LEAVE-MASTER** Request to the DP_SM to go to WAIT-PRM. 1 = The user causes the DP_SM to go to WAIT-PRM. After this action, VPC3+ sets User_LEAVE-MASTER to '0' again.
bit 2	**Go_Offline:** Going into the Offline state 1 = After the current request ends, VPC3+ goes to the Offline state and sets Go_Offline to '0' again.
bit 1	**EOI:** End-of-Interrupt 1 = VPC3+ disables the interrupt output and sets EOI to '0' again.
bit 0	**Start_VPC3:** Exiting the Offline state 1 = VPC3+ exits Offline and goes to Passive_Idle In addition the Idle Timer and Watchdog Timer are started and 'Go_Offline = 0' is set.

Figure D-7: Coding of Mode Register 1

D-3.1.3 Mode Register 2

Setting parameters for Mode Register 2 may take place in the Offline State only (like Mode Register 0).

Address	Bit Position								Description
	7	6	5	4	3	2	1	0	
	0	0	0	0	0	0	0	1	Reset Value
$0C_H$	4kB_Mode	No_Check_ Prm_Reserved	SYNC_Pol	SYNC_Ena	DX_Int_Port	DX_Int_Mode	No_Check_ GC_Reserved	New_GC_ Int_Mode	Mode Reg 2 7 .. 0

	Mode Register 2, Address $0C_H$:
bit 7	**4KB_Mode:** size of internal RAM 0 = 2K Byte RAM (default) 1 = 4K Byte RAM
bit 6	**No_Check_Prm_Reserved:** disables checking of the reserved bits in DPV1_Status_2/3 of Set_Prm telegram 0 = reserved bits of a Set_Prm telegram are checked (default) 1 = reserved bits of a Set_Prm telegram are not checked
bit 5	**SYNC_Pol:** polarity of SYNC pulse (for Isochron Mode only) 0 = negative polarity of SYNC pulse (default) 1 = positive polarity of SYNC pulse
bit 4	**SYNC_Ena:** enables generation of SYNC pulse (for Isochron Mode only) 0 = SYNC pulse generation is disabled (default) 1 = SYNC pulse generation is enabled
bit 3	**DX_Int_Port:** Port Mode for DX_Out interrupt (ignored if SYNC_Ena set) 0 = DX_Out interrupt is not assigned to port DATAEXCH (default) 1 = DX_Out interrupt (synchronized to SYNCH telegram) is assigned to port DATAEXCH.
bit 2	**DX_Int_Mode:** Mode of DX_Out interrupt 0 = DX_Out interrupt is only generated, if Len_Dout_Buf is unequal 0 (default) 1 = DX_Out interrupt is generated after every Data_Exchange telegram
bit 1	**No_Check_GC_Reserved:** disables checking of the reserved bits in Global_Control telegram 0 = reserved bits of a Global_Control telegram are checked (default) 1 = reserved bits of a Global_Control telegram are not checked
bit 0	**GC_Int_Mode:** controls generation of New_GC_Command interrupt 0 = New_GC_Command interrupt is only generated, if a changed Global_Control telegram is received 1 = New_GC_Command interrupt is generated after every Global_Control telegram (default)

Figure D-8: Coding of Mode Register 2

D-3.2 Status Register

The Status Register shows the current VPC3+ status and is read-only.

Address	Bit Position								Description
	7	6	5	4	3	2	1	0	
04_H (Intel)	WD_State		DP_State		Reserved	Diag_Flag	Reserved	Offline / Passive_Idle	Status-Reg 7..0 See below for coding
	1	0	1	0					

Address	Bit Position								Description
	15	14	13	12	11	10	9	8	
05_H (Intel)	VPC3+ Release				Baud Rate				Status-Reg 15..8 See below for coding
	3	2	1	0	3	2	1	0	

	Status Register,Low-Byte, Address 04_H (Intel):
bit 7,6	**WD_State 1..0:** State of the Watchdog State Machine 00 = BAUD_SEARCH state 01 = BAUD_CONTROL state 10 = DP_CONTROL state 11 = Not possible
bit 5,4	**DP_State 1..0:** State of the DP State Machine 00 = WAIT-PRM state 01 = WAIT-CFG state 10 = DATA-EXCH state 11 = Not possible
bit 3	**Reserved**
bit 2	**Diag_Flag:** Status of the Diagnosis-Buffer 0 = The Diagnosis-Buffer had been fetched by the DP-Master. 1 = The Diagnosis-Buffer had not been fetched by the DP-Master yet.
bit 1	**Reserved**
bit 0	**Offline / Passive_Idle:** Offline/Passive_Idle state 0 = VPC3+ is in Offline. 1 = VPC3+ is in Passive_Idle.

Figure D-9: Status Register, Low-Byte

	Status Register, High-Byte, Address 05_H (Intel):
bit 15-12	**VPC3+-Release 3..0 :** Release number for VPC3+ 0000 = Step A 1011 = Step B 1100 = Step C 1101 = Step D (this chip release) Rest = Not possible
bit 11-8	**Baud Rate 3..0 :** The baud rate found by VPC3+ 0000 = 12,00 Mbit/s 0001 = 6,00 Mbit/s 0010 = 3,00 Mbit/s 0011 = 1,50 Mbit/s 0100 = 500,00 Kbit/s 0101 = 187,50 Kbit/s 0110 = 93,75 Kbit/s 0111 = 45,45 Kbit/s 1000 = 19,20 Kbit/s 1001 = 9,60 Kbit/s 1111 = after reset and during baud rate search Rest = not possible

Figure D-10: Status Register, High-Byte

D-3.3 Interrupt Controller

The processor is informed about indication messages and various error events via the interrupt controller. Up to a total of 16 events are stored in the interrupt controller. The events are summed up to a common interrupt output. The controller does not have a prioritization level and does not provide an interrupt vector (not 8259A compatible!).

The controller consists of an Interrupt Request Register (IRR), an Interrupt Mask Register (IMR), an Interrupt Register (IR) and an Interrupt Acknowledge Register (IAR).

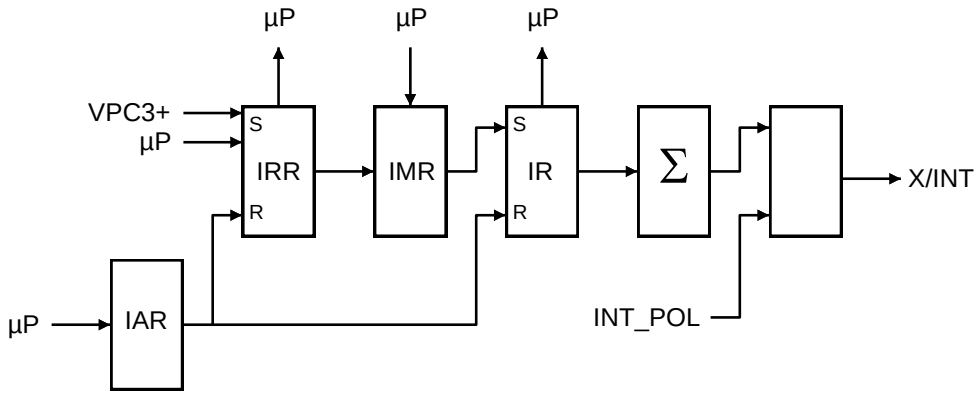


Figure D-11: Block Diagram of Interrupt Controller

Each event is stored in the IRR. Individual events can be suppressed via the IMR. The input in the IRR is independent of the interrupt masks. Events that are not masked in the IMR set the corresponding IR bit and generate the X/INT interrupt via a sum network. The user can set each event in the IRR for debugging.

Each interrupt event that was processed by the microcontroller must be deleted via the IAR (except for New_(Ext_)Prm_Data and New_Cfg_Data). A ‘1’ must be written on the specific bit position. If a new event and an acknowledge from the previous event are present at the IRR at the same time, the event remains stored. If the microcontroller enables a mask subsequently, it must be ensured that no prior IRR input is present. To be on the safe side, the position in the IRR must be deleted prior to the enabling of the mask.

Before leaving the interrupt routine, the microprocessor must set the ‘end of interrupt bit' (EOI = 1) in Mode Register 1. The interrupt output is switched to inactive with this edge change. If another event occurs, the interrupt output is not activated again until the interrupt inactive time of at least 1 µs or 1 ms expires. This interrupt inactive time can be set via EOI_Time_Base in Mode Register 0. This makes it possible to enter the interrupt routine again when an edge-triggered interrupt input is used.

The polarity of the interrupt output is parameterized via the Int_Pol bit in Mode Register 0. After hardware reset, the output is low-active.

D-3.3.1 Interrupt Request Register

Address	Bit Position								Description
	7	6	5	4	3	2	1	0	
00_H (Intel)	DXB_Out	New_Ext_ Prm_Data	DXB_Link_ Error	User_Timer_ Clock	WD_DP_ CONTROL_Timeout	Baud_Rate_ Detect	Go/Leave_ DATA-EXCH	MAC_Reset / Clock_Sync	Int-Req-Reg 7 .. 0 See below for coding

Address	Bit Position								Description
	15	14	13	12	11	10	9	8	
01_H (Intel)	FDL_Ind	Poll_End_Ind	DX_Out	Diag_Buffer_ Changed	New_Prm_ Data	New_Cfg_ Data	New_SSA_ Data	New_GC Command	Int-Req-Reg 15 .. 8 See below for coding

	Interrupt-Request-Register, Low-Byte, Address 00_H (Intel):
bit 7	**DXB_Out:** The VPC3+ has received a DXB telegram and made the new output data available in the 'N' buffer.
bit 6	**New_Ext_Prm_Data:** The VPC3+ has received a Set_Ext_Prm telegram and made the data available in the Parameter-Buffer.
bit 5	**DXB_Link_Error:** The Watchdog cycle is elapsed and at least one Publisher-Subscriber connection breaks down.
bit 4	**User_Timer_Clock:** The time base for the User_Timer_Clocks is run out (1 / 10ms).
bit 3	**WD_DP_CONTROL_Timeout:** The watchdog timer expired in the DP_CONTROL state.
bit 2	**Baud_Rate_Detect:** The VPC3+ has left the BAUD_SEARCH state and found a baud rate.
bit 1	**Go/Leave_DATA-EXCH:** The DP_SM has entered or exited the DATA-EXCH state.
bit 0	**MAC_Reset (used if CS_Supported=0)**: After processing the current request, the VPC3+ has entered the Offline state (by setting the Go_Offline bit). **Clock_Sync (used if CS_Supported=1)**: The VPC3+ has received a Clock_Value telegram or an error occurs. Further differentiation is made in the Clock_Sync-Buffer.

Figure D-12: Interrupt-Request-Register, Low-Byte

	Interrupt Request Register 0, High-Byte, Address 01_H (Intel):
bit 15	**FDL_Ind:** The VPC3+ has received an acyclic service request and made the data available in an Indication-Buffer.
bit 14	**Poll_End_Ind:** The VPC3+ has sent the response to an acyclic service.
bit 13	**DX_Out:** The VPC3+ has received a Data_Exchange telegram and made the new output data available in the 'N' buffer.
bit 12	**Diag_Buffer_Changed:** Due to the request made by New_Diag_Cmd, the VPC3+ exchanged the Diagnosis-Buffers and made the old buffer available to the user again.
bit 11	**New_Prm_Data:** The VPC3+ has received a Set_Prm telegram and made the data available in the Parameter-Buffer.
bit 10	**New_Cfg_Data:** The VPC3+ has received a Chk_Cfg telegram and made the data available in the Config-Buffer.
bit 9	**New_SSA_Data:** The VPC3+ have received a Set_Slave_Add telegram and made the data available in the Set_Slave_Add-Buffer.
bit 8	**New_GC_Command:** The VPC3+ has received a Global_Control telegram and stored the Control_Command in the R_GC_Command RAM cell.

Figure D-13: Interrupt Request Register, High-Byte

D-3.3.2 Interrupt Acknowledge / Mask Register

The other interrupt controller registers are assigned in the bit positions like the Interrupt Request Register.

Address	Register		Reset state	Assignment
02_H / 03_H	Interrupt Register (IR)	Readable only	All bits cleared	
04_H / 05_H	Interrupt Mask Register (IMR)	Writeable, can be changed during operation	All bits set	1 = Mask is set and the interrupt is disabled 0 = Mask is cleared and the interrupt is enabled
02_H / 03_H	Interrupt Acknowledge Register (IAR)	Writeable, can be changed during operation	All bits cleared	1 = Interrupt is acknowledged and the IRR bit is cleared 0 = IRR bit remains unchanged

Figure D-14: Interrupt Acknowledge / Mask Register

The New_(Ext_)Prm_Data and New_Cfg_Data interrupts cannot be acknowledged via the Interrupt Acknowledge Register. The relevant state machines clear these interrupts through the user acknowledgements (for example, User_Prm_Data_Okay etc.).

D-3.4 Watchdog Timer

The VPC3+ is able to identify the baud rate automatically. The state machine is in the BAUD_SEARCH state after each RESET and also after the Watchdog (WD) Timer has expired in the BAUD_CONTROL state.

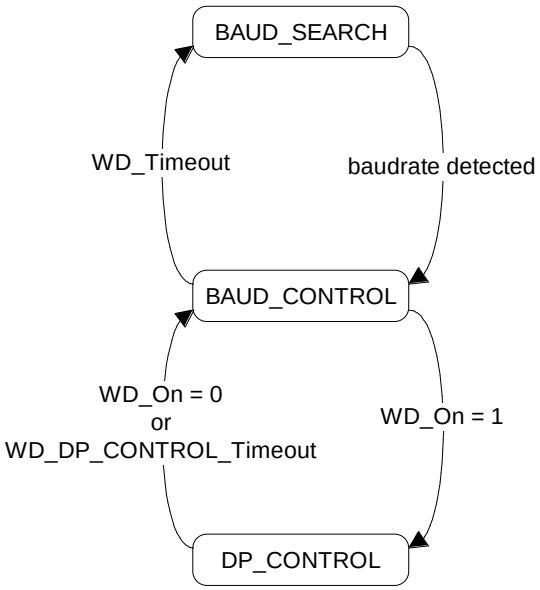


Figure D-15: Watchdog State Machine (WD_SM)

D-3.4.1 Automatic Baud Rate Identification

The VPC3+ starts searching for the transmission rate using the highest baud rate. If no SD1 telegram, SD2 telegram, or SD3 telegram was received completely and without errors during the monitoring time, the search continues using the next lower baud rate.

After identifying the correct baud rate, the VPC3+ switches to the BAUD_CONTROL state and observes the baud rate. The monitoring time can be parameterized (WD_BAUD_CONTROL_Val). The watchdog uses a clock of 100 Hz (10 ms). Each telegram to its own Station_Address received with no errors resets the Watchdog. If the timer expires, the VPC3+ switches to the BAUD_SEARCH state again.

D-3.4.2 Baud Rate Monitoring

The detected baud rate is permanently monitored in BAUD_CONTROL. The Watchdog is triggered by each error-free telegram to its own Station_Address. The monitoring time results from multiplying twice WD_BAUD_CONTROL_Val (user sets this parameter) by the time base (10 ms). If the timer expires, WD_SM again goes to BAUD_SEARCH. If the user uses the DP protocol (DP_Mode = 1, see Mode Register 0), the watchdog is used for the DP_CONTROL state, after a Set_Prm telegram was received with an enabled response time monitoring (WD_On = 1). The watchdog timer remains in the baud rate monitoring state when the master monitoring is disabled (WD_On = 0). The DP_SM is not reset when the timer expires in the state BAUD_CONTROL. That is, the DP-Slave remains in the DATA-EXCH state, for example.

D-3.4.3 Response Time Monitoring

The DP_CONTROL state serves as the response time monitoring of the DP-Master (Diag_Master_Add). The used monitoring time results from multiplying both watchdog factors and then multiplying this result with the time base (1 ms or 10 ms):

T_{WD} = WD_Base * WD_Fact_1 * WD_Fact_2
(See byte 7 of the Set_Prm telegram.)

The user can load the two watchdog factors (WD_Fact_1 and WD_Fact_2) and the time base that represents a measurement for the monitoring time via the Set_Prm telegram with any value between 1 and 255.

EXCEPTION:
The WD_Fact_1 = WD_Fact_2 = 1 setting is not allowed.
The device does not check this setting.

A monitoring time between 2 ms and 650 s - independent of the baud rate - can be implemented with the allowed watchdog factors.

If the monitoring time expires, the VPC3+ goes to BAUD_CONTROL state again and generates the WD_DP_CONTROL_Timeout interrupt. In addition, the DP State Machine is reset, that is, it generates the reset states of the buffer management. This operation mode is recommended for the most applications.
If another DP-Master takes over the VPC3+, the Watchdog State Machine either branches to BAUD_CONTROL (WD_On = 0) or to DP_CONTROL (WD_On = 1).

Notes:

D-4 PROFIBUS DP Interface

D-4.1 DP Buffer Structure

The DP_Mode is enabled in the VPC3+ with ‘DP_Mode = 1’ (see Mode Register 0). In this mode, the following SAPs are permanently reserved:

Default SAP:	Write and Read data (Data_Exchange)
SAP 53:	Sending extended parameter setting data (Set_Ext_Prm)
SAP 55:	Changing the Station_Address (Set_Slave_Add)
SAP 56:	Reading the inputs (RD_Input)
SAP 57:	Reading the outputs (RD_Output)
SAP 58:	Control commands to the DP-Slave (Global_Control)
SAP 59:	Reading configuration data (Get_Cfg)
SAP 60:	Reading diagnosis information (Slave_Diag)
SAP 61:	Sending parameter setting data (Set_Prm)
SAP 62:	Checking configuration data (Chk_Cfg)

The DP-Slave protocol is completely integrated in the VPC3+ and is handled independently. The user must correspondingly parameterize the ASIC and process and acknowledge received messages. All SAPs are always enabled except the Default SAP, SAP 56, SAP 57 and SAP 58. The remaining SAPs are not enabled until the DP_SM goes into the DATA-EXCH state. The user can disable SAP 55 to not permit changing the Station_Address. The corresponding buffer pointer R_SSA_Buf_Ptr must be set to ‘00H’ for this purpose.

The DP_SAP Buffer Structure is shown in Figure D-16. The user configures all buffers (length and buffer start) in the Offline state. During operation, the buffer configuration must not be changed, except for the length of the Dout-/Din-Buffers.

The user may still adapt these buffers in the WAIT-CFG state after the configuration telegram (Chk_Cfg). Only the same configuration may be accepted in the DATA-EXCH state.

The buffer structure is divided into the data buffers, Diagnosis-Buffers and the control buffers. Both the output data and the input data have three buffers available with the same length. These buffers are working as changing buffers. One buffer is assigned to the ‘D’ data transfer and one buffer is assigned to the ‘U’ user. The third buffer is either in a next state 'N' or a free state ‘F’. One of the two states is always unoccupied.

For diagnosis two Diagnosis-Buffers, that can have different lengths, are available. One Diagnosis-Buffer (D) is always assigned to the VPC3+ for sending. The other Diagnosis-Buffer (U) belongs to the user for preprocessing new diagnosis data.

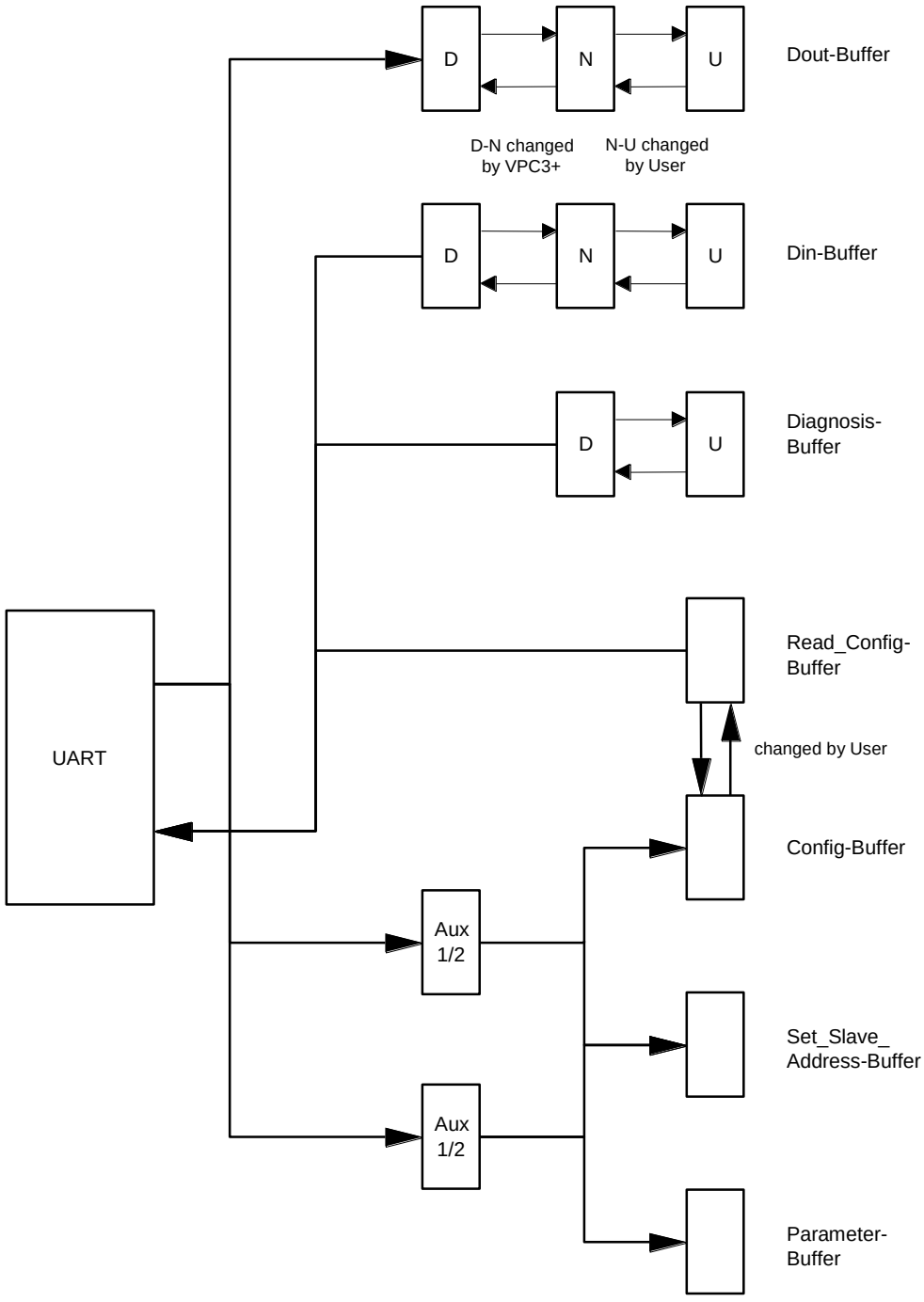


Figure D-16: DP_SAP Buffer Structure

The VPC3+ first stores the parameter telegrams (Set_Slave_Add and Set_(Ext_)Prm) and the configuration telegram (Chk_Cfg) in Aux-Buffer 1 or Aux-Buffer 2. If the telegrams are error-free, data is exchanged with the corresponding target buffer (Set_Slave_Add-Buffer, Parameter-Buffer and Config-Buffer). Each of the buffers to be exchanged must have the same length. In the R_Aux_Buf_Sel parameter cell the user defines which Aux-Buffers are to be used for the telegrams mentioned above. The Aux-Buffer

1 must always be available, Aux-Buffer 2 is optional. If the data profiles of these DP telegrams are very different (for example the length of the Set_Prm telegram is significantly larger than the length of the other telegrams) it is suggested to make an Aux-Buffer 2 available (R_Aux_Buf_Sel: Set_Prm = 1) for this telegram. The other telegrams are then read via Aux-Buffer 1 (R_Aux_Buf_Sel: Set_Slave_Add = 0, Chk_Cfg = 0). If the buffers are too small, the VPC3+ responds with “no resource” (RR)!

Address	Bit Position								Description
	7	6	5	4	3	2	1	0	
$2A_H$	0	0	0	0	0	Set_ Slave_Add	Chk_Cfg	Set_Prm	R_Aux_Buf_Sel See below for coding

	R_Aux_Buf_Sel, Address $2A_H$:
bit 7-3	**Don’t Care:** Read as ‘0’
bit 2	**Set_Slave_Add:** Set Slave Address 0 = Aux-Buffer 1 1 = Aux-Buffer 2
bit 1	**Chk_Cfg:** Check Configuration 0 = Aux-Buffer 1 1 = Aux-Buffer 2
bit 0	**Set_Prm:** Set (Extended) Parameter 0 = Aux-Buffer 1 1 = Aux-Buffer 2

Figure D-17: Aux-Buffer Management

The user makes the configuration data (Get_Cfg) available in the Read_Config-Buffer for reading. The Read_Config-Buffer must have the same length as the Config-Buffer.

The RD_Input telegram is serviced from the Din-Buffer in the ‘D’ state and the RD_Output telegram is serviced from the Dout-Buffer in the ‘U’ state.

All buffer pointers are 8-bit segment addresses, because the VPC3+ have only 8-bit address registers internally. For a RAM access, VPC3+ adds an 8-bit offset address to the segment address shifted by 4 bits (result: 12-bit physical address) in case of 4K Byte RAM or shifted by 3 bits (result: 11- bit physical address) in case of 2K Byte RAM. With regard to the buffer start addresses, this specification results either in a 16-byte or in an 8-byte granularity.

D-4.2 PROFIBUS DP Services

D-4.2.1 Set_Slave_Add (SAP 55)

Sequence for the Set_Slave_Add service

The user can disable this service by setting 'R_SSA_Puf_Ptr = 00H'. The Station_Address must then be determined, for example, by reading a DIP-switch or an EEPROM and writing the address in the RAM cell R_TS_Adr.

There must be a non-volatile memory available (for example an external EEPROM) to support this service. It must be possible to store the Station_Address and the Real_No_Add_Change ('True' = FFH) parameter in this EEPROM. After each restart caused by a power failure, the user must read these values from the EEPROM again and write them to the R_TS_Adr und R_Real_No_Add_Change RAM registers.

If SAP55 is enabled and the Set_Slave_Add telegram is received correctly, the VPC3+ enters the pure data in the Aux-Buffer 1/2, exchanges the Aux-Buffer 1/2 for the Set_Slave_Add-Buffer, stores the entered data length in R_Len_SSA_Data, generates the New_SSA_Data interrupt and internally stores the New_Slave_Add as Station_Address and the No_Add_Chg as Real_No_Add_Chg. The user does not need to transfer this changed parameter to the VPC3+ again. After reading the buffer, the user generates the SSA_Buffer_Free_Cmd (read operation on address 14H). This makes the VPC3+ ready again to receive another Set_Slave_Add telegram (for example, from a different DP-Master).

The VPC3+ reacts automatically to errors.

Address	Bit Position								Description
	7	6	5	4	3	2	1	0	
14_H	0	0	0	0	0	0	0	0	SSA_Buf_ Free_Cmd

	SSA_Buf_Free_Cmd, Address 14_H:
bit 7-0	**Don't care:** Read as '0'

Figure D-18: Coding of SSA_Buffer_Free_Command

Structure of the Set_Slave_Add Telegram

The net data are stored as follows in the SSA buffer:

Byte	Bit Position								Description
	7	6	5	4	3	2	1	0	
0									New_Slave_Add
1									Ident_Number_High
2									Ident_Number_Low
3									No_Add_Chg
4 : 243									Rem_Slave_Data additional application specific data

Figure D-19: Structure of the Set_Slave_Add Telegram

D-4.2.2 Set _Prm (SAP 61)

Parameter Data Structure

The VPC3+ evaluates the first seven data bytes (without User_Prm_Data), or the first eight data bytes (with User_Prm_Data). The first seven bytes are specified according to the standard. The eighth byte is used for VPC3+ specific characteristics. The additional bytes are available to the application.

If a PROFIBUS DP extension shall be used, the bytes 7-9 are called DPV1_Status and must be coded as described in section "PROFIBUS DP Extensions". Generally it is recommended to start the User_Prm_Data first with byte 10.

Byte	Bit Position								Description
	7	6	5	4	3	2	1	0	
0	Lock_ Req	Unlock_ Req	Sync_ Req	Freeze_ Req	WD_On	Reserved	Reserved	Reserved	Station Status
1									WD_Fact_1
2									WD_Fact_2
3									$minT_{SDR}$
4									Ident_Number_High
5									Ident_Number_Low
6									Group_Ident
7	DPV1_ Enable	Fail_Safe	Publisher_ Enable	0	0	WD_Base	Dis_Stop_ Control	Dis_Start_ Control	Spec_User_Prm_Byte /DPV1_Status_1
8									DPV1_Status_2
9									DPV1_Status_3
10 : 243									User_Prm_Data

Figure D-20: Format of the Set_Prm Telegram

	Spec_User_Prm_Byte / DPV1_Status_1:
bit 7	**DPV1_Enable:** 0 = DP-V1 extensions disabled (default) 1 = DP-V1 extensions enabled
bit 6	**Fail_Safe:** 0 = Fail Safe mode disabled (default) 1 = Fail Safe mode enabled
bit 5	**Publisher_Enable:** 0 = Publisher function disabled (default) 1 = Publisher function enabled
bit 4-3	**Reserved:** To be parameterized with '0'
bit 2	**WD_Base:** Watchdog Time Base 0 = Watchdog time base is 10 ms (default) 1 = Watchdog time base is 1 ms
bit 1	**Dis_Stop_Control:** Disable Stop bit Control 0 = Stop bit monitoring in the receiver is enabled (default) 1 = Stop bit monitoring in the receiver is disabled
bit 0	**Dis_Start_Control:** Disable Start bit Control 0 = Start bit monitoring in the receiver is enabled (default) 1 = Start bit monitoring in the receiver is disabled

Figure D-21: Spec_User_Prm_Byte / DPV1_Status_1

It is recommended <u>not</u> to use the DPV1_Status bytes (bytes 7-9) for user parameter data.

Parameter Data Processing Sequence

In the case of a positive validation of more than seven data bytes, the VPC3+ carries out the following reaction:

The VPC3+ exchanges Aux-Buffer 1/2 (all data bytes are entered here) for the Parameter-Buffer, stores the input data length in R_Len_Prm_Data and triggers the New_Prm_Data interrupt. The user must then check the User_Prm_Data and either reply with User_Prm_Data_Okay_Cmd or with User_Prm_Data_Not_Okay_Cmd. The entire telegram is entered in this buffer. The user parameter data are stored beginning with data byte 8, or with byte 10 if DPV1_Status bytes used.

The user response (User_Prm_Data_Okay_Cmd or User_Prm_Data_-Not_Okay_Cmd) clears the New_Prm_Data interrupt. The user cannot acknowledge the New_Prm_Data interrupt in the IAR register.

With the User_Prm_Data_Not_Okay_Cmd message, relevant diagnosis bits are set and the DP_SM branches to WAIT-PRM.

The User_Prm_Data_Okay and User_Prm_Data_Not_Okay acknowledgements are read accesses to defined registers with the relevant signals:

- User_Prm_Finished: No additional parameter telegram is present.
- Prm_Conflict: An additional parameter telegram is present, processing again
- Not_Allowed: Access not permitted in the current bus state

Address	Bit Position								Description
	7	6	5	4	3	2	1	0	
$0E_H$	0	0	0	0	0	0	⇓	⇓	User_Prm_ Data_Okay
							0	0	User_Prm_Finished
							0	1	Prm_Conflict
							1	1	Not_Allowed

Address	Bit Position								Description
	7	6	5	4	3	2	1	0	
$0F_H$	0	0	0	0	0	0	⇓	⇓	User_Prm_ Data_Not_Okay
							0	0	User_Prm_Finished
							0	1	Prm_Conflict
							1	1	Not_Allowed

Figure D-22: Coding of User_Prm_(Not_)Okay_Cmd

If another Set_Prm telegram is supposed to be received in the meantime, the signal Prm_Conflict is returned for the positive or negative acknowledgement of the first Set_Prm telegram. Then the user must repeat the validation because the VPC3+ has made a new Parameter-Buffer available.

D-4.2.3 Chk_Cfg (SAP 62)

The user checks the correctness of the configuration data. After receiving an error-free Chk_Cfg telegram, the VPC3+ exchanges the Aux-Buffer 1/2 (all data bytes are entered here) for the Config-Buffer, stores the input data length in R_Len_Cfg_Data and generates the New_Cfg_Data interrupt.

Then the user has to check the User_Config_Data and either respond with User_Cfg_Data_Okay_Cmd or with User_Cfg_Data_Not_Okay_Cmd. The pure data is entered in the buffer in the format of the standard.

The user response (User_Cfg_Data_Okay_Cmd or the User_Cfg_Data_Not_Okay_Cmd response) clears the New_Cfg_Data interrupt. The user cannot acknowledge the New_Cfg_Data in the IAR register.

If an incorrect configuration is reported, several diagnosis bits are changed and the VPC3+ branches to state WAIT-PRM.

For a correct configuration, the transition to DATA-EXCH takes place immediately, if trigger counters for the parameter telegrams and configuration telegrams are at 0. When entering into DATA-EXCH, the VPC3+ also generates the Go/Leave_DATA-EXCH interrupt.

If the received configuration data from the Config-Buffer is supposed to result in a change to the Read_Config-Buffer (contains the data for the Get_Cfg telegram), the user have to make the new Read_Config data available in the Read_Config-Buffer before the User_Cfg_Data_Okay_Cmd acknowledgement, that is the user has to copy the new configuration data into the Read_Config-Buffer.

During acknowledgement, the user receives information about whether there is a conflict or not. If another Chk_Cfg telegram was supposed to be received in the meantime, the user receives the Cfg_Conflict signal during the positive or negative acknowledgement of the first Chk_Cfg telegram. Then the user must repeat the validation, because the VPC3+ have made a new Config-Buffer available.

The User_Cfg_Data_Okay_Cmd and User_Cfg_Data_Not_Okay_Cmd acknowledgements are read accesses to defined memory cells with the relevant Not_Allowed, User_Cfg_Finished or Cfg_Conflict signals.

If the New_Prm_Data and New_Cfg_Data are supposed to be present simultaneously during start-up, the user must maintain the Set_Prm and then the Chk_Cfg acknowledgement sequence.

Address	Bit Position								Description
	7	6	5	4	3	2	1	0	
10_H	0	0	0	0	0	0	⇓	⇓	User_Cfg_ Data_Okay
							0	0	User_Cfg_Finished
							0	1	Cfg_Conflict
							1	1	Not_Allowed

Address	Bit Position								Description
	7	6	5	4	3	2	1	0	
11_H	0	0	0	0	0	0	⇓	⇓	User_Cfg_ Data_Not_Okay
							0	0	User_Cfg_Finished
							0	1	Cfg_Conflict
							1	1	Not_Allowed

Figure D-23: Coding of User_Cfg_(Not_)Okay_Cmd

D-4.2.4 Slave_Diag (SAP 60)

Diagnosis Processing Sequence

Two buffers are available for diagnosis. These two buffers can have different lengths. One Diagnosis-Buffer, which is sent on a diagnosis request, is always assigned to the VPC3+. The user can pre-process new diagnosis data in the other buffer parallel. If the new diagnosis data are to be sent, the user issues the New_Diag_Cmd to make the request to exchange the Diagnosis-Buffers. The user receives confirmation of the buffer exchange with the Diag_Buffer_Changed interrupt.

When the buffers are exchanged, the internal Diag_Flag is also set. For an activated Diag_Flag, the VPC3+ responds during the next Data_Exchange with high-priority response data. That signals the DP-Master that new diagnosis data are present at the DP-Slave. The DP-Master then fetches the new diagnosis data with a Slave_Diag telegram. Then the Diag_Flag is cleared again. However, if the user signals ‘Diag.Stat_Diag = 1’ (that is static diagnosis, see the structure of the Diagnosis-Buffer), the Diag_Flag still remains activated after the relevant DP-Master has fetched the diagnosis. The user can poll the Diag_Flag in the Status Register to find out whether the DP-Master has already fetched the diagnosis data before the old data is exchanged for the new data.

According to IEC 61158, Static Diagnosis should only be used during start-up.

Status coding for the Diagnosis-Buffers is stored in the Diag_Buffer_SM control parameter. The user can read this cell with the possible codings for both buffers: User, VPC3+, or VPC3+_Send_Mode.

Address	Bit Position								Description
	7	6	5	4	3	2	1	0	
$0C_H$	0	0	0	0	Diag_Buf2		Diag_Buf1		Diag_Buffer_SM

	Diag_Buffer_SM, Address 0CH:
bit 7-4	**Don't care:** Read as '0'
bit 3-2	**Diag_Buf2:** Assignment of Diagnosis-Buffer 2 00 = Nil 01 = User 10 = VPC3+ 11 = VPC3_Send_Mode
bit 1-0	**Diag_Buf1:** Assignment of Diagnosis-Buffer 1 00 = Nil 01 = User 10 = VPC3+ 11 = VPC3_Send_Mode

Figure D-24: Diagnosis-Buffer Assignment

The New_Diag_Cmd is also a read access to a defined control parameter indicating which Diagnosis-Buffer belongs to the user after the exchange or whether both buffers are currently assigned to the VPC3+ (No_Buffer, Diag_Buf1, Diag_Buf2).

Address	Bit Position								Description
	7	6	5	4	3	2	1	0	
$0D_H$	0	0	0	0	0	0	⇓	⇓	New_Diag_ Buffer_Cmd
							0	0	No_Buffer
							0	1	Diag_Buf1
							1	0	Diag_Buf2

Figure D-25: Coding of New_Diag_Cmd

Byte	Bit Position								Description
	7	6	5	4	3	2	1	0	
0						Ext_Diag_ Overflow	Stat_Diag	Ext_Diag	
1									
2									
3									
4									
5									
6 : n	user input								Ext_Diag_Data (n = max. 243)

Figure D-26: Format of the Diagnosis-Buffer

The Ext_Diag_Data must be entered into the buffers after the VPC3+ internal diagnosis data. Three different formats are possible here: device-related, ID-related and port-related. If PROFIBUS DP extensions shall be used, the device-related diagnosis is substituted by alarm and status messages. In addition to the Ext_Diag_Data, the buffer length also includes the VPC3+ diagnosis bytes (R_Len_Diag_Buf 1, R_Len_Diag_Buf 2).

D-4.2.5 Write_Read_Data / Data_Exchange (Default_SAP)

Writing Outputs

The VPC3+ writes the received output data in the 'D' buffer. After an error-free receipt, the VPC3+ shifts the newly filled buffer from ‘D’ to ‘N'. In addition, the DX_Out interrupt is generated. The user now fetches the current output data from ‘N’. The buffer changes from ‘N’ to ‘U’ with the Next_Dout_Buffer_Cmd, so that the current data can be transmitted to the application by a RD_Output request from a DP-Master.

If the user’s evaluation cycle time is shorter than the bus cycle time, the user does not find any new buffers with the next Next_Dout_Buffer_Cmd in ‘N'. Therefore, the buffer exchange is omitted. At a 12 Mbit/s baud rate, it is more likely, however, that the user’s evaluation cycle time is larger than the bus cycle time. This makes new output data available in ‘N’ several times before the user fetches the next buffer. It is guaranteed, however, that the user receives the data last received.

For power-on, LEAVE-MASTER and the Global_Control telegram with ‘Clear_Data = 1’, the VPC3+ deletes the ‘D’ buffer and then shifts it to ‘N'. This also takes place during power-up (entering the WAIT-PRM state). If the user fetches this buffer, he receives U_Buffer_Cleared during the

Next_Dout_Buffer_Cmd. If the user is supposed to enlarge the output data buffer after the Chk_Cfg telegram, the user must delete this deviation in the 'N' buffer himself (possible only during the start-up phase in the WAIT-CFG state).

If ‘Diag.Sync_Mode = 1’, the ‘D’ buffer is filled but not exchanged with the Data_Exchange telegram. It is exchanged at the next Sync or Unsync command sent by Global_Control telegram.

Address	Bit Position								Description
	7	6	5	4	3	2	1	0	
$0A_H$	F		U		N		D		Dout_Buffer_SM

	Dout_Buffer_SM, Address $0A_H$:
bit 7-6	**F:** Assignment of the F-Buffer 00 = Nil 01 = Dout_Buf_Ptr1 10 = Dout_Buf_Ptr2 11 = Dout_Buf_Ptr3
bit 5-4	**U:** Assignment of the U-Buffer 00 = Nil 01 = Dout_Buf_Ptr1 10 = Dout_Buf_Ptr2 11 = Dout_Buf_Ptr3
bit 3-2	**N:** Assignment of the N-Buffer 00 = Nil 01 = Dout_Buf_Ptr1 10 = Dout_Buf_Ptr2 11 = Dout_Buf_Ptr3
bit 1-0	**D:** Assignment of the D-Buffer 00 = Nil 01 = Dout_Buf_Ptr1 10 = Dout_Buf_Ptr2 11 = Dout_Buf_Ptr3

Figure D-27: Dout-Buffer Management

When reading the Next_Dout_Buffer_Cmd the user gets the information which buffer (‘U’ buffer) belongs to the user after the change, or whether a change has taken place at all.

Address	Bit Position								Description
	7	6	5	4	3	2	1	0	
$0B_H$	0	0	0	0	U_Buffer_Cleared	State_U_Buffer	Ind_U_Buffer		Next_Dout_Buf_Cmd See coding below

	Next_Dout_Buf_Cmd, Address $0B_H$:
bit 7-4	**Don't care:** Read as '0'
bit 3	**U_Buffer_Cleared:** User-Buffer-Cleared Flag 0 = U buffer contains data 1 = U buffer is cleared
bit 2	**State_U_Buffer:** State of the User-Buffer 0 = no new U buffer 1 = new U buffer
bit 1-0	**Ind_U_Buffer:** Indicated User-Buffer 01 = Dout_Buf_Ptr1 10 = Dout_Buf_Ptr2 11 = Dout_Buf_Ptr3

Figure D-28: Coding of Next_Dout_Buf_Cmd

The user must clear the 'U' buffer during initialization so that defined (cleared) data can be sent for a RD_Output telegram before the first data cycle.

Reading Inputs

The VPC3+ sends the input data from the 'D' buffer. Prior to sending, the VPC3+ fetches the Din-Buffer from 'N' to 'D'. If no new buffer is present in 'N', there is no change.

The user makes the new data available in 'U'. With the New_Din_-Buffer_Cmd, the buffer changes from 'U' to 'N'. If the user's preparation cycle time is shorter than the bus cycle time, not all new input data are sent, but just the most current. At a 12 Mbit/s baud rate, it is more likely, however, that the user's preparation cycle time is larger than the bus cycle time. Then the VPC3+ sends the same data several times in succession.

During start-up, the VPC3+ does not go to DATA-EXCH before all parameter telegrams and configuration telegrams have been acknowledged.

If 'Diag.Freeze_Mode = 1', there is no buffer change prior to sending.

The user can read the status of the state machine cell with the following codings for the four states: Nil, Dout_Buf_Ptr1, Dout_Buf_Ptr2 and Dout_Buf_Ptr3. The pointer for the current data is in the 'N' state.

Address	Bit Position								Description
	7	6	5	4	3	2	1	0	
08_H	F		U		N		D		Din_Buffer_SM

	Din_Buffer_SM, Address 08_H:
bit 7-6	**F:** Assignment of the F-Buffer 00 = Nil 01 = Din_Buf_Ptr1 10 = Din_Buf_Ptr2 11 = Din_Buf_Ptr3
bit 5-4	**U:** Assignment of the U-Buffer 00 = Nil 01 = Din_Buf_Ptr1 10 = Din_Buf_Ptr2 11 = Din_Buf_Ptr3
bit 3-2	**N:** Assignment of the N-Buffer 00 = Nil 01 = Din_Buf_Ptr1 10 = Din_Buf_Ptr2 11 = Din_Buf_Ptr3
bit 1-0	**D:** Assignment of the D-Buffer 00 = Nil 01 = Din_Buf_Ptr1 10 = Din_Buf_Ptr2 11 = Din_Buf_Ptr3

Figure D-29: Din-Buffer Management

Address	Bit Position								Description
	7	6	5	4	3	2	1	0	
09_H	0	0	0	0	0	0	⇓	⇓	New_Din_Buf_Cmd
							0	1	Din_Buf_Ptr1
							1	0	Din_Buf_Ptr2
							1	1	Din_Buf_Ptr3

Figure D-30: Coding of New_Din_Buf_Cmd

User_Watchdog_Timer

After start-up (DATA-EXCH state), it is possible that the VPC3+ continually answers Data_Exchange telegrams without the user fetching the received Dout-Buffers or making new Din-Buffers available. If the user processor ‘hangs up' the DP-Master would not receive this information. Therefore, a User_Watchdog_Timer is implemented in the VPC3+.

This User_WD_Timer is an internal 16-bit RAM cell that is started from a user parameterized value R_User_WD_Value and is decremented by the VPC3+ with each received Data_Exchange telegram. If the timer reaches the value 0000H, the VPC3+ goes to the WAIT-PRM state and the DP_SM carries out a LEAVE-MASTER. The user must cyclically set this timer to its start value. Therefore, ‘Res_User_WD = 1’ must be set in Mode Register 1. Upon receipt of the next Data_Exchange telegram, the VPC3+ again loads the User_WD_Timer to the parameterized value R_User_WD_Value and sets ‘Res_User_WD = 0’ (Mode Register 1). During power-up, the user must also set ‘Res_User_WD = 1’, so that the User_WD_Timer is set to its parameterized value.

D-4.2.6 Global_Control (SAP 58)

The VPC3+ processes the Global_Control telegrams like already described.

The first byte of a valid Global_Control is stored in the R_GC_Command RAM cell. The second telegram byte (Group_Select) is processed internally.

The interrupt behavior regarding to the reception of a Global_Control telegram can be configured via bit 8 of Mode Register 2. The VPC3+ either generates the New_GC_Control interrupt after each receipt of a Global_Control telegram (default) or just in case if the last Global_Control differs from the previous one.

The R_GC_Command RAM cell is not initialized by the VPC3+. Therefore the cell has to be preset with 00H during power-up. The user can read and evaluate this cell.

In order to use Sync and Freeze, these functions must be enabled in the Mode Register 0.

Address	Bit Position								Description
	7	6	5	4	3	2	1	0	
$3C_H$	Reserved	Reserved	Sync	Unsync	Freeze	Unfreeze	Clear_Data	Reserved	R_GC_ Command See below for coding

	R_GC_Command, Address $3C_H$:
bit 7-6	**Reserved**
bit 5	**Sync:** The output data transferred with a Data_Exchange telegram is changed from ‘D’ to ‘N’. The following transferred output data is kept in ‘D’ until the next Sync command is given.
bit 4	**Unsync:** The Unsync command cancels the Sync command.
bit 3	**Freeze:** The input data is fetched from ‘N’ to ‘D’ and „frozen“. New input data is not fetched again until the DP-Master sends the next Freeze command.
bit 2	**Unfreeze:** The Unfreeze command cancels the Freeze command.
bit 1	**Clear_Data:** With this command, the output data is deleted in ‘D’ and is changed to ‘N’.
bit 0	**Reserved**

Figure D-31: Format of the Global_Control Telegram

D-4.2.7 RD_Input (SAP 56)

The VPC3+ fetches the input data like it does for the Data_Exchange telegram. Prior to sending, ‘N’ is shifted to ‘D', if new input data are available in ‘N'. For ‘Diag.Freeze_Mode = 1', there is no buffer change.

D-4.2.8 RD_Output (SAP 57)

The VPC3+ fetches the output data from the Dout-Buffer in ‘U’. The user must preset the output data with ‘0’ during start-up so that no invalid data can be sent here. If there is a buffer change from ‘N’ to ‘U’ (through the Next_Dout_Buffer_Cmd) between the first call-up and the repetition, the new output data is sent during the repetition.

D-4.2.9 Get_Cfg (SAP 59)

The user makes the configuration data available in the Read_Config-Buffer. For a change in the configuration after the Chk_Cfg telegram, the user writes the changed data in the Config-Buffer, sets 'En_Change_Cfg_-Buffer = 1' (see Mode Register 1) and the VPC3+ then exchanges the Config-Buffer for the Read_Config-Buffer. If there is a change in the configuration data during operation (for example, for a modular DP systems), the user must return with Go_Offline command (see Mode Register 1) to WAIT-PRM.

D-5 PROFIBUS DP Extensions

D-5.1 Set_(Ext_)Prm (SAP 53 / SAP 61)

The PROFIBUS DP extensions require three bytes to implement the new parameterization function. The bits of the Spec_User_Prm_Byte are included.

Byte	Bit Position								Description
	7	6	5	4	3	2	1	0	
0 : 6									
7	DPV1_ Enable	Fail_Safe	Publisher_ Enable	Reserved	Reserved	WD_Base	Dis_Stop_ Control	Dis_Start_ Control	DPV1_Status_1
8	Enable_ Pull_Plug_Alarm	Enable_ Process_Alarm	Enable_ Diagnostic_Alarm	Enable_ Manufacturer_ Specific_Alarm	Enable_ Status_Alarm	Enable_ Update_Alarm	0	Chk_Cfg_Mode	DPV1_Status_2
9	PrmCmd	0	0	IsoM_Req	Prm_ Structure	Alarm_Mode			DPV1_Status_3
10 : 243									User_Prm_Data

Figure D-32: Set_Prm with DPV1_Status bytes

If the extensions are used, the bit Spec_Clear_Mode in Mode Register 0 serves as Fail_Safe_required. Therefore it is used for a comparison with the bit Fail_Safe in parameter telegram. Whether the DP-Master support the Fail_Safe mode or not is indicated by the telegram bit. If the DP-Slave requires Fail_Safe but the DP-Master doesn't the Prm_Fault bit is set.

If the VPC3+ should be used for DXB, IsoM, CS or redundancy mode, the parameterization data must be packed in a Structured_Prm_Data block to distinguish between the User_Prm_Data. The bit Prm_Structure indicates this.

If redundancy should be supported, the PrmCmd_Supported bit in Mode Register 0 must be set.

Byte	Bit Position								Description
	7	6	5	4	3	2	1	0	
0									Structured_Length
1									Structure_Typ
2									Slot_Number
3									Reserved
4 : 243									User_Prm_Data

Figure D-33 : Format of the Structured_Prm_Data block

Additional to the Set_Prm telegram (SAP 61) a Set_Ext_Prm (SAP 53) telegram can be used for parameterization. This service is only available in state WAIT-CFG after the reception of a Set_Prm telegram and before the reception of a Chk_Cfg telegram. The new Set_Ext_Prm telegram simply consists of Structured_Prm_Data blocks.

The new service uses the same buffer handling as described by Set_Prm. By means of the New_Ext_Prm_Data interrupt the user can recognize which kind of telegram is entered in the Parameter-Buffer. Additional the SAP 53 must be activated by Set_Ext_Prm_Supported bit in Mode Register 0.

The Aux-Buffer for the Set_Ext_Prm is the same as the one for Set_Prm and have to be different from the Chk_Cfg Aux-Buffer. Furthermore the Spec_Prm_Buf_Mode in Mode Register 0 must not be used together with SAP 53.

D-5.2 PROFIBUS DP-V1

D-5.2.1 Acyclic Communication Relationships

The VPC3+ supports acyclic communication as described in the DP-V1 specification. Therefore a memory area is required which contains all SAPs needed for the communication. The user must do the initialization of this area (SAP-List) in Offline state. Each entry in the SAP-List consists of 7 bytes. The pointer at address 17_H contains the segment base address of the first element of the SAP-List. The last element in the list is always indicated with FF_H. If the SAP-List shall not be used, the first entry must be FF_H, so the pointer at address 17_H must point to a segment base address location that contains FF_H.

The new communication features are enabled with DPV1_Enable in the Set_Prm telegram.

Byte	Bit Position								Description
	7	6	5	4	3	2	1	0	
0	Response_Sent	SAP_Number							SAP_Number
1									Request_SA
2									Request_SSAP
3									Service_Supported
4									Ind_Buf_Ptr[0]
5									Ind_Buf_Ptr[1]
6									Resp_Buf_Ptr

	SAP-List entry:
Byte 0	**Response_Sent:** Response-Buffer sent 0 = no Response sent 1 = Response sent **SAP_Number:** 0 – 51
Byte 1	**Request_SA:** The source address of a request is compared with this value. At differences, the VPC3+ response with “no service activated” (RS). The default value for this entry is 7FH.
Byte 2	**Request_SSAP:** The source SAP of a request is compared with this value. At differences, the VPC3+ response with “no service activated” (RS). The default value for this entry is 7FH.
Byte 3	**Service_Supported:** Indicates the permitted FDL service. 00 = all FDL services allowed
Byte 4	**Ind_Buf_Ptr[0]:** pointer to Indication-Buffer 0
Byte 5	**Ind_Buf_Ptr[1]:** pointer to Indication-Buffer 1
Byte 6	**Resp_Buf_Ptr:** pointer to Response-Buffer

Figure D-34: SAP-List entry

In addition an Indication- and Response-Buffer are needed. Each buffer consists of a 4-byte header for the buffer management and a data block of configurable length.

Byte	Bit Position								Description
	7	6	5	4	3	2	1	0	
0	USER	IND	RESP	INUSE					Control
1									Max_Length
2									Length
3									Function Code

	SAP-List entry:
Byte 0	**Control:** bits for buffer management USER buffer assigned to user IND indication data included in buffer RESP response data included in buffer INUSE buffer assigned to VPC3+
Byte 1	**Max_Length:** length of buffer
Byte 2	**Length:** length of data included in buffer
Byte 3	**Function Code:** function code of the telegram

Figure D-35: Buffer Header

Processing Sequence

A received telegram is compared with the values in the SAP-List. If this check is positive, the telegram is stored in an Indication-Buffer with the INUSE bit set. In case of any deviations the VPC3+ responses with “no service activated” (RS) or if no free buffer is available with “no resource” (RR). After finishing the processing of the incoming telegram, the INUSE bit is reset and the bits USER and IND are set by VPC3+. Now the FDL_Ind interrupt is generated. Polling telegrams do not produce interrupts. The RESP bit indicates response data, provided by the user in the Response-Buffer. The Poll_End_Ind interrupt is set after the Response-Buffer is sent. Also bits RESP and USER are cleared.

DP-Master	PROFIBUS	DP-Slave
	Request to acyclic SAP →	fill Indication-Buffer
	← short acknowledgement (SC)	
	Polling telegram to acyclic SAP →	process data
	← short acknowledgement (SC)	
	:	
	:	
	:	update Response-Buffer
	Polling telegram to acyclic SAP →	
	← Response from acyclic	

Figure D-36: acyclic communication sequence

VPC3+	Firmware
set Request_SA / Request_SSA	
set INUSE in Control of Ind_Buf	
write data in Ind_Buf	
clear INUSE and set USER and IND in Control of Ind_Buf	
set FDL_Ind interrupt	
	clear FDL_Ind interrupt
	search for Ind_Buf with IND = 1
	read Ind_Buf
	clear IND in Control of Ind_Buf
	write Response in Resp_Buf
	set RESP in Control of Resp_Buf
check on RESP = 1	
read Resp_Buf	
clear RESP and USER in Control of Resp_Buf	
set Response_Sent	
set Poll_End_Ind interrupt	
	clear Poll_End_Ind interrupt
	search for SAP with Response_Sent = 1
	clear Response_Sent

Figure D-37: FDL-interface of VPC3+ (e.g. same Buffer for Indication and Response)

D-5.2.2 Diagnosis Model

The format of the device related diagnosis data depends on the GSD keyword DPV1_Slave in the GSD. If 'DPV1_Slave = 1', alarm and status messages are used in diagnosis telegrams. Status messages are required by the Data eXchange Broadcast service, for example. Alarm_Ack is used as the other acyclic services.

D-5.3 PROFIBUS DP-V2

D-5.3.1 Data eXchange Broadcast (DXB)

The DXB mechanism enables a fast slave-to-slave communication. A DP-Slave that holds input data significant for other DP-Slaves, works as a Publisher. The Publisher can handle a special kind of Data_Exchange request from the DP-Master and sends its answer as a broadcast telegram. Other DP-Slaves that are parameterized as Subscribers can monitor this telegram. A link is opened to the Publisher if the address of the Publisher is registered in the linktable of the Subscriber. If the link has been established correctly, the Subscriber can fetch the input data from the Publisher.

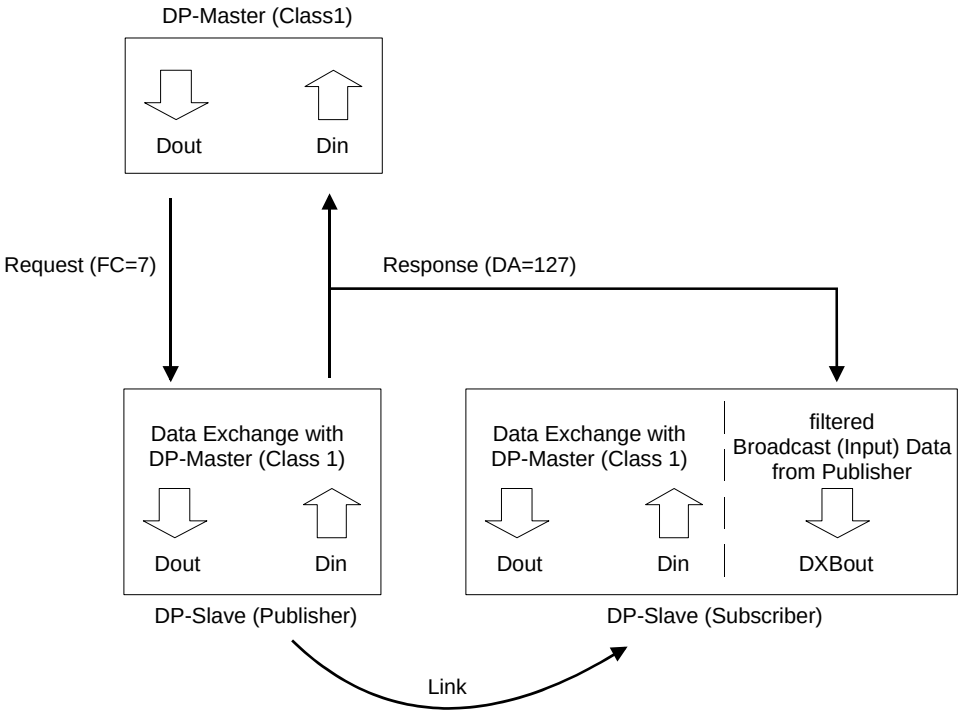


Figure D-38 : Overview DXB

The VPC3+ can handle a maximum of 29 links simultaneously.

Publisher

A Publisher is activated with 'Publisher_Enable = 1' in DPV1_Status_1. The time minT_{SDR} must be set to 'T_{ID1} = 37 t_{bit} + 2 T_{SET} + T_{QUI}'.

All Data_Exchange telegrams containing the function code 7 (Send and Request Data Multicast) are responded with destination address 127. If Publisher mode is not enabled, these requests are ignored.

Subscriber

A Subscriber requires information about the links to its Publishers. These settings are contained in a DXB Linktable or DXB Subscribertable and transferred via the Structured_Prm_Data in a Set_Prm or Set_Ext_Prm telegram. Each Structured_Prm_Data is treated like the User_Prm_Data and therefore evaluated by the user. From the received data the user must generate DXB_Link_Buf and DXB_Status_Buf entries. The watchdog must be enabled to make use of the monitoring mechanism. The user must check this.

Byte	Bit Position								Description
	7	6	5	4	3	2	1	0	
0									Structured_Length
1	0	0	0	0	0	0	1	1	Structure_Type
2	0	0	0	0	0	0	0	0	Slot_Number
3	0	0	0	0	0	0	0	0	Reserved
4	0	0	0	0	0	0	0	1	Version
5									Publisher_Addr
6									Publisher_Length
7									Sample_Offset
8									Sample_Length
9 : 120									further link entries

Figure D-39: Format of the Structured_Prm_Data with DXB Linktable (specific link is grey scaled)

Byte	Bit Position								Description
	7	6	5	4	3	2	1	0	
0									Structured_Length
1	0	0	0	0	0	1	1	1	Structure_Type
2	0	0	0	0	0	0	0	0	Slot_Number
3	0	0	0	0	0	0	0	0	Reserved
4	0	0	0	0	0	0	0	1	Version
5									Publisher_Addr
6									Publisher_Length
7									Sample_Offset
8									Dest_Slot_Number
9									Offset_Data_Area
10									Sample_Length
11 : 120									further link entries

Figure D-40: Format of the Structured_Prm_Data with DXB Subscribertable (specific link is grey scaled)

The user must copy the link entries of DXB Linktable or DXB Subscriber-table, without Dest_Slot_Number and Offset_Data_Area, in the DXB_Link_Buf and set R_Len_DXB_Link_Buf. Also the user must enter the default status message in DXB_Status_Buf with the received links and write the appropriate values to R_Len_DXB_Status_Buf. After that, the parameterization interrupt can be acknowledged.

Byte	Bit Position								Description
	7	6	5	4	3	2	1	0	
0	0	0	Block_Length						Header_Byte
1	1	0	0	0	0	0	1	1	Status_Type
2	0	0	0	0	0	0	0	0	Slot_Number
3	0	0	0	0	0	0	0	0	Status_Specifier
4									Publisher_Addr
5	Link_ Status	Link_ Error	0	0	0	0	0	Data_ Exist	Link_Status
6 : 61									further link entries

	Link_Status:
bit 7	**Link_Status:** 1 = active, valid data receipt during last monitoring period 0 = not active, no valid data receipt during last monitoring period (DEFAULT)
bit 6	**Link_Error:** 0 = no faulty Broadcast data receipt (DEFAULT) 1 = wrong length, error occurred by reception
bit 0	**Data_Exist:** 0 = no correct Broadcast data receipt during current monitoring period (DEFAULT) 1 = error free reception of Broadcast data during current monitoring period

Figure D-41: DXB_Link_Status_Buf (specific link is grey scaled)

Processing Sequence

The VPC3+ processes DXBout-Buffers like the Dout-Buffers. The only difference is that the DXBout-Buffers are not cleared by the VPC3+.

The VPC3+ writes the received and filtered broadcast data in the 'D' buffer. The buffer contains also the Publisher_Address and the Sample_Length. After error-free receipt, the VPC3+ shifts the newly filled buffer from 'D' to 'N'. In addition, the DXBout interrupt is generated. The user now fetches the current output data from 'N'. The buffer changes from 'N' to 'U' with the Next_DXBout_Buffer_Cmd.

Byte	Bit Position								Description
	7	6	5	4	3	2	1	0	
0									Publisher_Addr
1									Sample_Length
2 : 246									Sample_Data

Figure D-42: DXBout-Buffer

When reading the Next_DXBout_buffer_Cmd the user gets the information which buffer ('U' buffer) is assigned to the user after the change, or whether a change has taken place at all.

Address	Bit Position								Description
	7	6	5	4	3	2	1	0	
12H	F		U		N		D		DXBout_Buffer_SM

	DXBout_Buffer_SM, Address 0AH:
bit 7-6	**F:** Assignment of the F-Buffer 00 = Nil 01 = DXBout_Buf_Ptr1 10 = DXBout_Buf_Ptr2 11 = DXBout_Buf_Ptr3
bit 5-4	**U:** Assignment of the U-Buffer 00 = Nil 01 = DXBout_Buf_Ptr1 10 = DXBout_Buf_Ptr2 11 = DXBout_Buf_Ptr3
bit 3-2	**N:** Assignment of the N-Buffer 00 = Nil 01 = DXBout_Buf_Ptr1 10 = DXBout_Buf_Ptr2 11 = DXBout_Buf_Ptr3
bit 1-0	**D:** Assignment of the D-Buffer 00 = Nil 01 = DXBout_Buf_Ptr1 10 = DXBout_Buf_Ptr2 11 = DXBout_Buf_Ptr3

Figure D-43: DXBout-Buffer Management

Address	Bit Position								Description
	7	6	5	4	3	2	1	0	
13_H	0	0	0	0	0	State_U_Buffer	Ind_U_Buffer		Next_DXBout_Buf_Cmd See coding below

	Next_Dout_Buf_Cmd, Address 13_H:
bit 7-3	**Don't care:** Read as '0'
bit 2	**State_U_Buffer:** State of the User-Buffer 0 = no new U buffer 1 = new U buffer
bit 1-0	**Ind_U_Buffer:** Indicated User-Buffer 01 = DXBout_Buf_Ptr1 10 = DXBout_Buf_Ptr2 11 = DXBout_Buf_Ptr3

Figure D-44: Coding of Next_DXBout_Buf_Cmd

Monitoring

After receiving the DXB data the Link_Status in DXB_Status_Buf of the concerning Publisher is updated. In case of an error the bit Link_Error is set. If the processing is finished without errors, the bit Data_Exist is set.

In state DATA-EXCH the links are monitored in intervals defined by the parameterized watchdog time. After the monitoring time runs out, the VPC3+ evaluates the Link_Status of each Publisher and updates the bit Link_Status. The timer restarts again automatically.

Event	Link_ Status	Link_ Error	Data_ Exist
valid DXB data receipt		0	1
faulty DXB data receipt	0	1	0
WD_Time elapsed AND Data_Exist = 1	1	0	0
WD_Time elapsed AND Link_Error = 1	0	0	0

Figure D-45: Link_Status handling

To enable the monitoring of Publisher-Subscriber links the watchdog timer must be enabled in the Set_Prm telegram. The user must check this.

D-5.3.2 Isochron Mode (IsoM)

The IsoM synchronizes DP-Master, DP-Slave and DP-Cycle. The isochron cycle time starts with the transmission of the SYNCH telegram by the IsoM master. If the VPC3+ is configured to support IsoM, a synchronization pulse is generated at Pin 13 (XDATAEXCH/SYNC) with each reception of a SYNCH telegram.

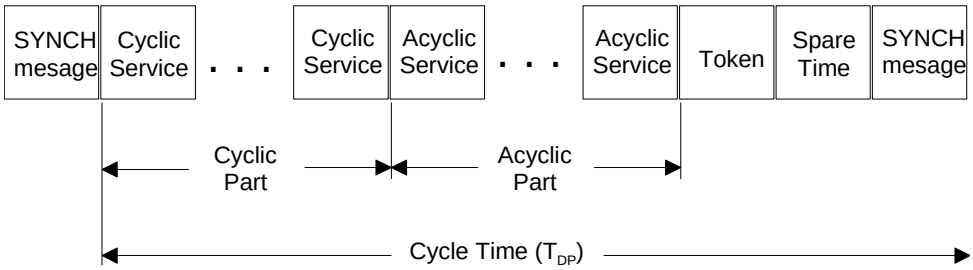


Figure D-46: Telegram sequences in IsoM with one DP-Master (Class 1)

Two operation modes for cyclic synchronization are available in the VPC3+:

1. Isochron Mode: Each SYNCH telegram causes an impulse on the SYNC output and a New_GC_Command interrupt.

2. Simple Sync Mode: A Data_Exchange telegram no longer causes a DX_Out interrupt immediately, rather the event is stored in a flag. By a following SYNCH message reception, the DX_Out interrupt and a synchronization signal are generated at the same time. Additionally a New_GC_Command interrupt is produced, as the SYNCH telegram behaves like a regular Global_Control telegram to the DP state machine. If no Data_Exchange telegram precedes the SYNCH telegram, only the New_GC_Command interrupt is generated.

Byte	Bit Position								Description
	7	6	5	4	3	2	1	0	
0	0	0						0	Control_Command
1	Group_8 = 1								Group_Select

Figure D-47: IsoM SYNCH telegram

The compare values for SYNCH telegram can be adjusted in Control_Command_Reg (0Eh) and Group_Select_Reg (0Fh).

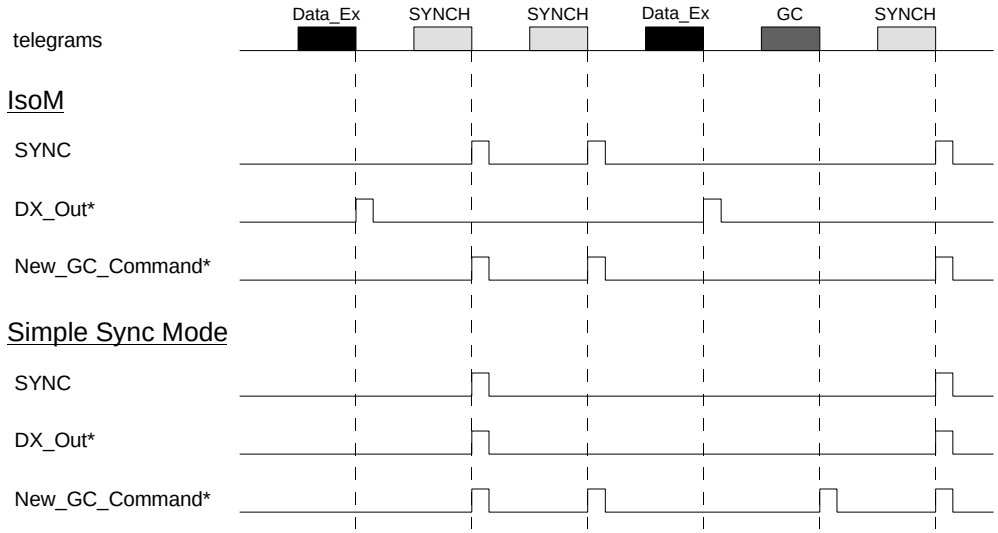


Figure D-48: SYNC-signal and interrupts for synchronization modes

Isochron Mode

To enable the Isochron Mode in the VPC3+, bit SYNC_Ena in Mode Register 2 must be set. Additionally the Spec_Clear_Mode in Mode Register 0 must be set. The polarity of the SYNC signal can be adjusted with the SYNC_Pol bit. The register Sync_PW contains a multiplicator with the base of 1/12 µs to adjust the SYNC pulse width. Settings in the Set_Prm telegram are shown below.

The Structured_Prm_Data block IsoM (Structure_Type = 4) is also required for the application. If it is sent by Set_Prm telegram the bit Prm_Structure must be set.

Byte	Bit Position								Description
	7	6	5	4	3	2	1	0	
0			Sync_Req = 0	Freeze_Req = 0					Station_Status
1									WD_Fact_1
2									WD_Fact_2
3									$minT_{SDR}$
4									Ident_Number_High
5									Ident_Number_Low
6	Group_8 = 0								Group_Ident
7		Fail_Safe = 1							DPV1_Status_1
8									DPV1_Status_2
9				IsoM_Req = 1					DPV1_Status_3
10 : 246									User_Prm_Data

Figure D-49: Format of Set_Prm telegram for IsoM

DP-Slave in a IsoM network

To enable cyclic synchronization via the 'Simple Sync Mode', the bit DX_-Int_Port in Mode Register 2 have to be set. Bit SYNC_Ena must not be set. The settings of the pulse polarity are adjusted like described in the IsoM.

For the parameterization telegram the DP format is used. Though the DPV1_Status bytes 1-3 could be used for User_Prm_Data, it is generally recommended starting the User_Prm_Data at byte 10.

Byte	Bit Position								Description
	7	6	5	4	3	2	1	0	
0			Sync_Req = depends on SYNCH-format	Freeze_Req = depends on SYNCH-format					Station_Status
1									WD_Fact_1
2									WD_Fact_2
3									$minT_{SDR}$
4									Ident_Number_High
5									Ident_Number_Low
6	Group_8 = 1								Group_Ident
7									DPV1_Status_1
8									DPV1_Status_2
9									DPV1_Status_3
10 : 246									User_Prm_Data

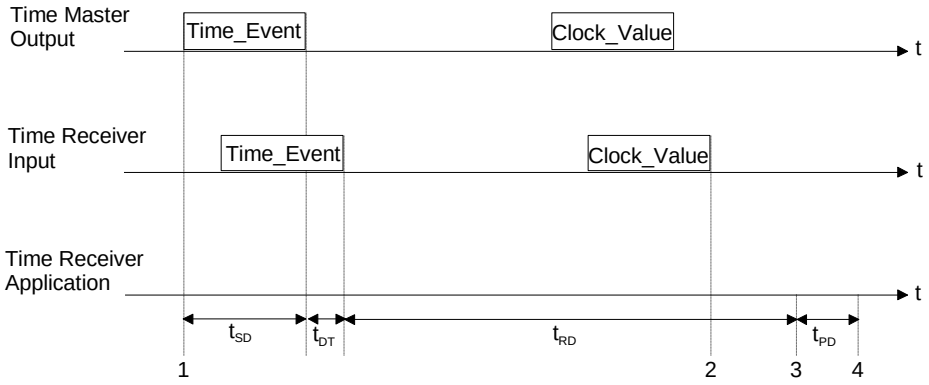
Figure D-50: Format of Set_Prm for DP-Slave using isochrones cycles

In opposite to IsoM the DX_Out interrupt is generated first after the receipt of a SYNCH telegram. If no Data_Exchange telegram had been received before a SYNCH occurred, no synchronization signal is generated.

For this mechanism the interrupt controller is used. Hence no signal will be generated, if the mask for DX_Out in the IMR is set. Since the synchronization signal is now the DX_Out interrupt, it remains active until the interrupt acknowledge.

D-5.3.3 Clock Synchronization (CS)

Clock Synchronization mechanism synchronizes the clocks of all devices on a PROFIBUS segment. A time master is a DP-Master. The scheme used is a “backwards time based correction”. The knowledge of when a special timer event message was broadcasted is subsequently used to calculate appropriate clock adjustments.
The synchronized clocks can be used for time stamp mechanism.



1: Time Event
2: Clock_Sync Interrupt
3: read access Receive_Delay_Time
4: update system timer

Figure D-51: clock synchronization mechanism

The clock synchronization sequence consists of two messages broadcasted by the time master. When the first message, called Time_Event, is received the VPC3+ starts the receive delay timer (t_{RD}). The time master then sends a second message, called Clock_Value, which contains the actual time when the Time_Event was sent plus the send delay time (t_{SD}). By receiption of the second message the Clock_Sync interrupt will be generated. To achieve high accuracy the receive delay timer runs until the user reads the Clock_Sync-Buffer.
The VPC3+ only synchronizes the received telegrams, the system clock management is done by the user. The user has to consider the time after the receive delay timer has been read till the update of the system clock is actually done (t_{PD}: process delay time).

The time for transmission delay (t_{DT}: CS_Delay_Time) and the Clock_Sync_Interval are transferred to the VPC3+ within a Structured_Prm_Data block. The CS_Delay_Time is used by the user to calculate the system time: t_S = Clock_Value_Time_Event + t_{DT} + t_{RD} + t_{PD}

Byte	Bit Position								Description
	7	6	5	4	3	2	1	0	
0									Structured_Length
1	0	0	0	0	1	0	0	0	Structure_Type
2	0	0	0	0	0	0	0	0	Slot_Number
3	0	0	0	0	0	0	0	0	Reserved
4 : 5									Clock_Sync_Interval Time Base 10 ms
6 : 13	Seconds (2^{31}..0) Fraction Part of Seconds (2^{31}..0) Base is 1/(2^{32}) Seconds								CS Delay Time can be omitted

Figure D-52: Format of Structured_Prm_Data with Time AR

Processing Sequence

The Clock_Sync_Interval is a time for monitoring and has to be written into the Clock_Sync-Buffer by the user. The Time Receiver state machine in the VPC3+ ist started after this write access. The value for Clock_Sync_Interval is locked until the next LEAVE-MASTER or a new parameterization occurs. Also it can be unlocked if the user set the Stop_Clock_Sync in Command byte.

Following to a clock synchronization sequence the Clock_Sync interrupt will be set. Further informations are contained in the Status byte. If an overflow of the Receive_Delay_Timer occurs the Status byte will be cleared. The VPC3+ can't write new data to the Clock_Sync-Buffer until the user acknowledge the Clock_Sync interrupt. Hence to ensure no new data overwrites the buffer, the user should read out the buffer prior to interrupt acknowledge.

The base address of the Clock_Sync-Buffer depends on memory mode:
2K Byte mode: $7E0_H$
4K Byte mode: $FE0_H$

Byte	Bit Position								Description
	7	6	5	4	3	2	1	0	
0	reserved						Clock_Sync_ Violation	Set_Time	Status
1	reserved					Clock_Value_ Check_Ena	Ignore_Cyclic_ State_Machine	Stop_ Clock_Sync	Command
2	C	CV					reserved		Clock_Value_Status1
3	ANH	SWT	reserved	CR		reserved		SYF	Clock_Value_Status2
4 : 11	Seconds (2^{31}..0) since 1.1.1900 0:00,00 or since 7.2.2036 6:28:16 if value < 9DFF4400H								Clock_Value_ Time_Event
	Fraction Part of Seconds (2^{31}..0) Base is 1/(2^{32}) Seconds								
12 : 15									Receive_Delay_Time Time Base 1 µs
16 : 23	Seconds (2^{31}..0) since 1.1.1900 0:00,00 or since 7.2.2036 6:28:16 if value < 9DFF4400H								Clock_Value_ previous_TE
	Fraction Part of Seconds (2^{31}..0) Base is 1/(2^{32}) Seconds								
24 : 25									Clock_Sync_Interval Time Base 10 ms

	Clock_Sync-Buffer
Status bit 1	**Clock_Sync_Violation**: Wrong telegram or Time period of 2*T_{CSI} expired after reception of Time_Event.
Status bit 0	**Set_Time**: The VPC3+ has received a valid 'Clock_Value telegram' and made the data available in the Clock_Sync-Buffer.
Command bit 2	**Clock_Value_Check_Ena**: 0 = don't evaluate Clock_Value_previous_TE 1 = check Clock_Value_previous_TE with local variable Time_Last_Rcvd
Command bit 1	**Ignore_Cyclic_State_Machine**: 0 = Clock Synchronization stops after the receiption of a new Set_Prm or a LEAVE-MASTER 1 = Clock Synchronization continues until the user set Stop_Clock_Sync
Command bit 0	**Stop_Clock_Sync**: Stop the Clock Synchronization, in order to write a new T_{CSI} without a previous Set_Prm or LEAVE-MASTER. The Bit is cleared by the Time_Receiver State Machine.
Clock_Value_ Status1 bit 7	**Sign of CV**: 0 = add correction value to Time 1 = substract correction value to Time
Clock_Value_ Status1 bit 6-2	**Correction Value**: 0 = 0 min 1..31 = 30..930 min
Clock_Value_ Status2 bit 7	**Announcment Hour**: 0 = no change planned within the next hour 1 = a change of SWT will occur within the next hour
Clock_Value_ Status2 bit 6	**Summertime**: 0 = Winter Time 1 = Summer Time
Clock_Value_ Status2 bit 4-3	**Accuracy**: 0 = 1 ms 1 = 10 ms 2 = 100 ms 3 = 1 s
Clock_Value_ Status2 bit 0	**Synchronisation Active**: 0 = Clock_Value_Time_Event is synchronized 1 = Clock_Value_Time_Event is not synchronized

Figure D-53: Format of the Clock_Sync-Buffer

Notes:

D-6 Hardware Interface

D-6.1 Processor Bus Interface

D-6.1.1 Bus Interface Unit

The Bus Interface Unit (BIU) is the interface to the connected processor/microcontroller. This is a synchronous or asynchronous 8-bit interface with an 11-bit (12-bit in 4K Byte mode) address bus. The interface is configurable via 2 pins (TYP, MODE). The connected processor family (bus control signals such as XWR, XRD, or R_W and the data format) is specified with the TYP pin. Synchronous or asynchronous bus timing is specified with the MODE pin.

TYP	MODE	Processor Interface Mode
0	1	Synchronous Intel mode
0	0	Asynchronous Intel mode
1	0	Asynchronous Motorola mode
1	1	Synchronous Motorola mode

Figure D-54: Configuration of the Processor Interface

Examples of various Intel system configurations are given in subsequent sections. The internal address latch and the integrated decoder must be used in the synchronous Intel mode. One figure shows the minimum configuration of a system with the VPC3+, where the chip is connected to an EPROM version of the controller. Only a clock generator is necessary as an additional device in this configuration. If a controller is to be used without an integrated program memory, the addresses must be latched for the external memory.

Note:
If the VPC3+ is connected to an 80286 or similar processor, it must be taken into consideration that the processor carries out word accesses. That is, either a ‘swapper’ is necessary that switches the characters out of the VPC3+ at the correct byte position of the 16-bit data bus during reading or the least significant address bit is not connected and the 80286 must read word accesses and evaluate only the lower byte.

Name	Input/ Output	Type	Comments
DB(7..0)	I/O	Tristate	High-resistance during RESET
AB(10..0)	I		AB(10) has a pull down resistor.
MODE	I		Configuration Sync interface: <log>1 Async interface: <log>0
XWR E_CLOCK AB11	I		Intel: Write Sync. Motorola: E-Clk Async. Motorola: AB11
XRD/R_W	I		Intel: Read Motorola: Read/Write
XCS AB11	I		Others: Chip Select Sync. Intel: AB11
ALE AS AB11	I		Sync. Intel: Address Latch Enable Async. Motorola: Address Strobe Aync. Intel: AB11 Sync. Motorola: AB11
DIVIDER	I		Scaling factor for CLKOUT Divide by 2: <log>1 Divide by 4: <log>0
INTEV	O	Push/Pull	Polarity programmable
XRDY XDTACK	O	Tristate	Asyc. Intel: Ready-Signal Async. Motorola: DTACK-Signal
CLK	I	Schmitt-Trigger	48 MHz
TYP	I		Processor Interface Format Intel: <log>0 Motorola: <log>1
CLKOUT	O	Push/Pull	24/12 MHz
RESET	I	Schmitt-Trigger	Minimum of 4 clock cycles

Figure D-55: Microprocessor Bus Signals in VPC3+ Operation Mode

Synchronous Intel Mode

In this mode Intel CPUs like 80C51/52/32 and compatible processor series from several manufacturers can be used.

- Synchronous bus timing without evaluation of the XREADY signal
- 8-bit multiplexed data/address bus: DB7..0
- The lower address bits AB7..0 are stored with the ALE signal in an internal address latch.
- The internal CS decoder is activated. VPC3+ generates its own CS signal from the address lines AB10..3. The VPC3+ selects the relevant address window from the AB2..0 signals.
- A11 from the microcontroller must be connected to XCS (pin 1) in 4K Byte mode as this is the additional address bus signal in this mode. In 2K Byte mode this pin is not used and should be pulled to VCC.

Asynchronous Intel Mode

In this mode various 16-/8-bit microcontroller series like Intel's x86, Siemens 80C16x or compatible series from other manufacturers can be used.

- Asynchronous bus timing with evaluation of the XREADY signal
- 8-bit non-multiplexed bus: DB7..0, AB10..0 (AB11..0 in 4K Byte mode)
- The internal VPC3+ address decoder is disabled, the XCS input is used instead.
- External address decoding is always necessary.
- External chip select logic is necessary if not present in the processor
- A11 from the microcontroller must be connected to ALE/AS (pin 24) in 4K Byte mode as this is the additional address bus signal in this mode. In 2K Byte mode this pin is not used and should be pulled to GND.

Asynchronous Motorola Mode

Motorola microcontrollers like the HC16 and HC916 can be used in this mode. When using HC11 types with a multiplexed bus the address signals AB7..0 must be generated from the DB7..0 signals externally.

- Asynchronous bus timing with evaluation of the XREADY signal
- 8-bit non-multiplexed bus: DB7..0, (AB11..0 in 4K Byte mode)
- The internal VPC3+ address decoder is disabled, the XCS input is used instead.
- Chip select logic is available and programmable in all microcontrollers mentioned above.
- AB11 must be connected to XWR/E_CLOCK (pin 2) in 4K Byte mode as this is the additional address bus signal in this mode. In 2K Byte mode this pin is not used and should be pulled to GND.

Synchronous Motorola Mode

Motorola microcontrollers like the HC11 types K, N, M, F1 or the HC16- and HC916 types with programmable E_Clock timing can be used in this mode. When using HC11 types with a multiplexed bus the address signals AB7..0 must be generated from the DB7..0 signals externally.

- Synchronous bus timing without evaluation of the XREADY signal
- 8-bit non-multiplexed bus: DB7..0, AB10..0 (AB11..0 in 4K Byte mode)
- The internal VPC3+ address decoder is disabled, the XCS input is used instead.
- For microcontrollers with chip select logic (K, F1, HC16 and HC916), the chip select signals are programmable regarding address range, priority, polarity and window width in the write cycle or read cycle.
- For microcontrollers without chip select logic (N and M) and others, an external chip select logic is required. This means additional hardware and a fixed assignment.
- If the CPU is clocked by the VPC3+, the output clock pulse (CLKOUT 2/4) must be 4 times larger than the E_Clock. That is, a clock pulse signal must be present at the CLK input that is at least 10 times larger than the desired system clock pulse (E_Clock). The Divider-Pin must be connected to ‘0’ (divider 4). This results in an E_Clock of 3 MHz.
- AB11 must be connected to ALE/AS (pin 24) in 4K Byte mode as this is the additional address bus signal in this mode. In 2K Byte mode this pin is not used and should be pulled to GND.

D-6.1.2 Application Examples (Principles)

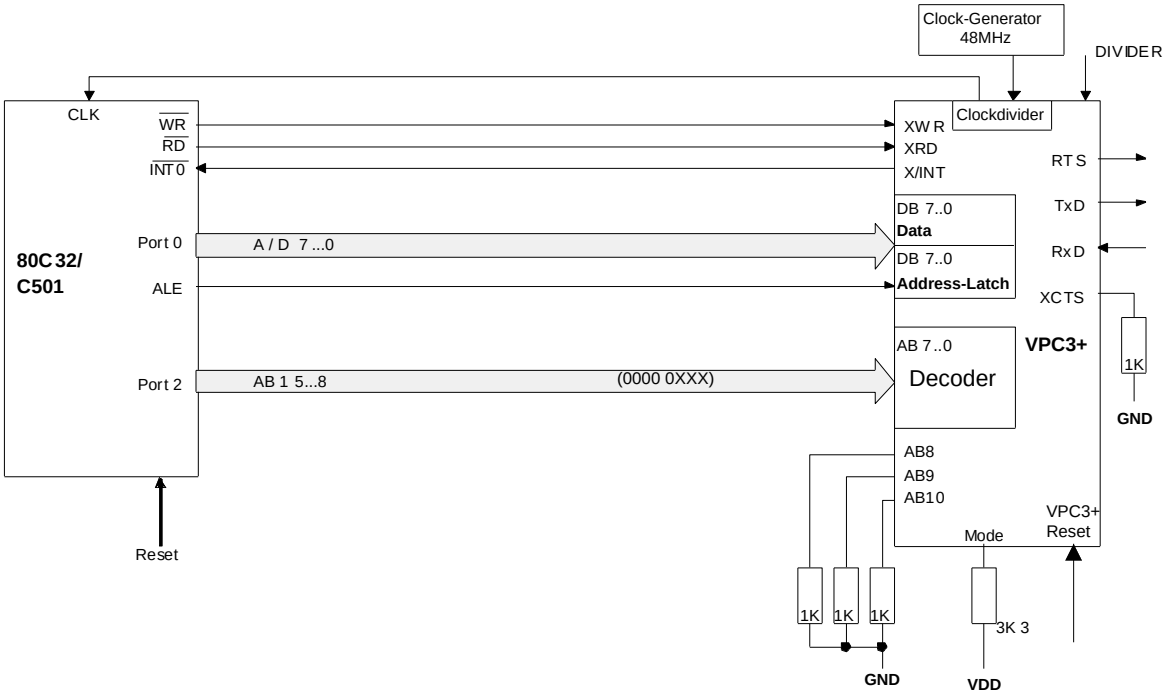


Figure D-56: Low Cost System with 80C32

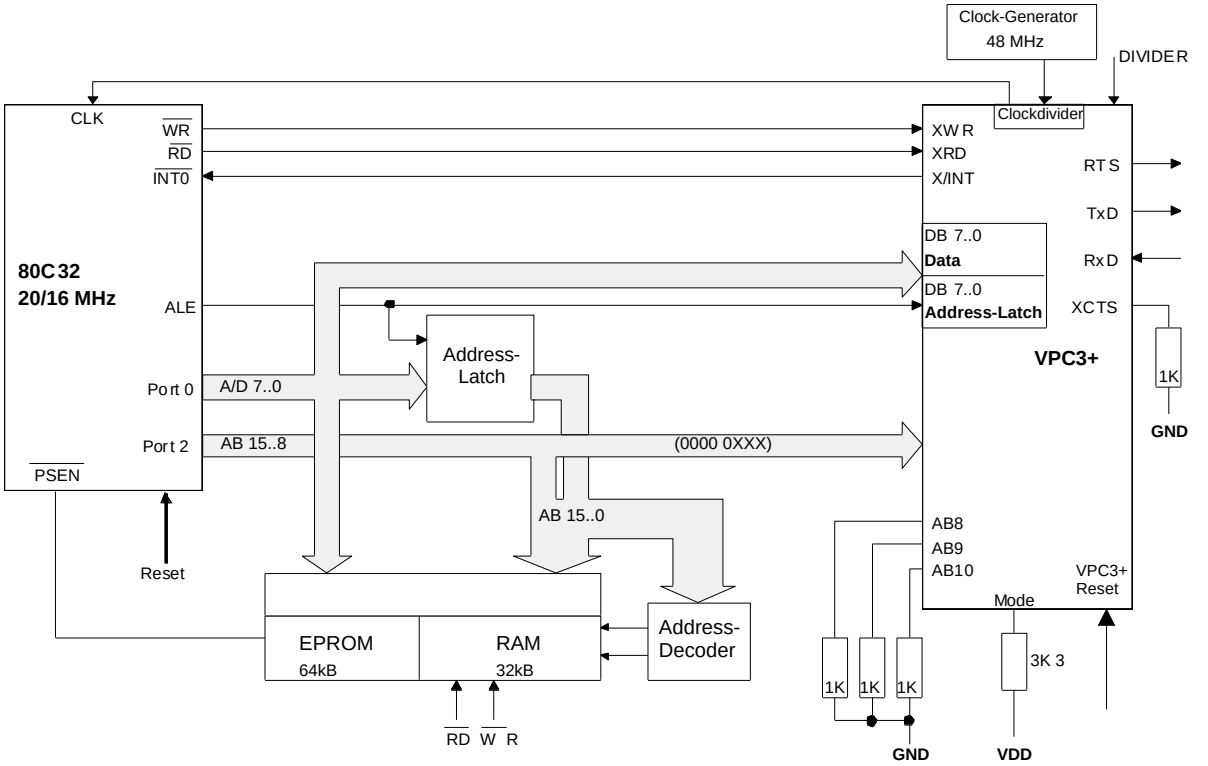


Figure D-57: 80C32 System with External Memory

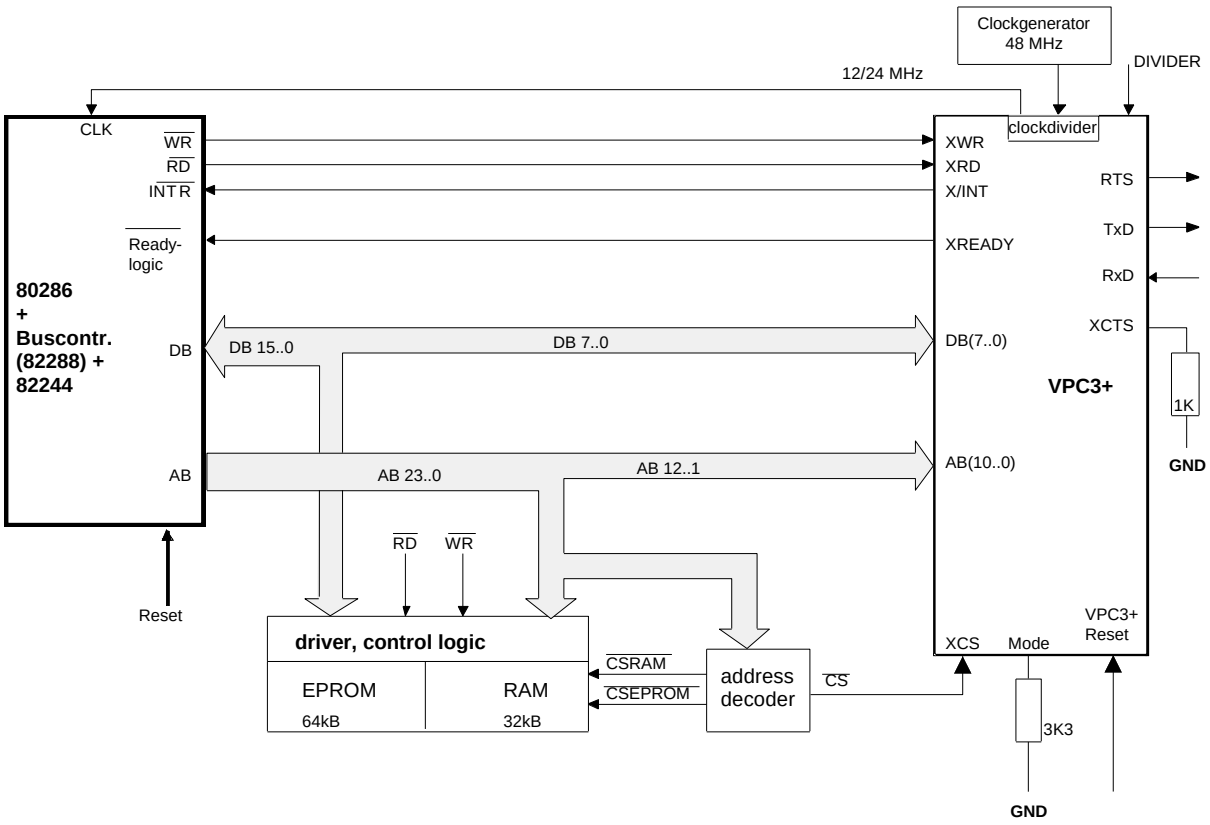


Figure D-58: 80286 System (X86 Mode)

D-6.1.3 Application with 80C32 (2K Byte RAM)

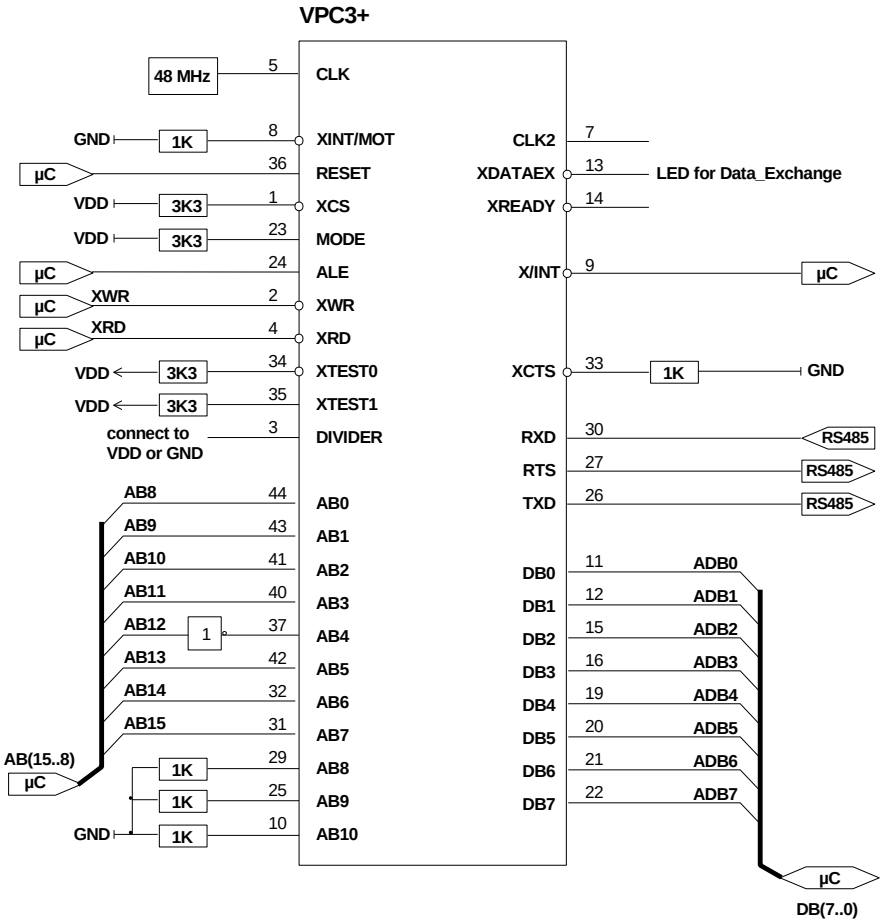


Figure D-59: 80C32 Application in 2K Byte mode

The internal chipselect is activated when the address inputs AB[10..3] of the VPC3+ are set to '0'.
In the example above the start address of the VPC3+ is set to 1000H.

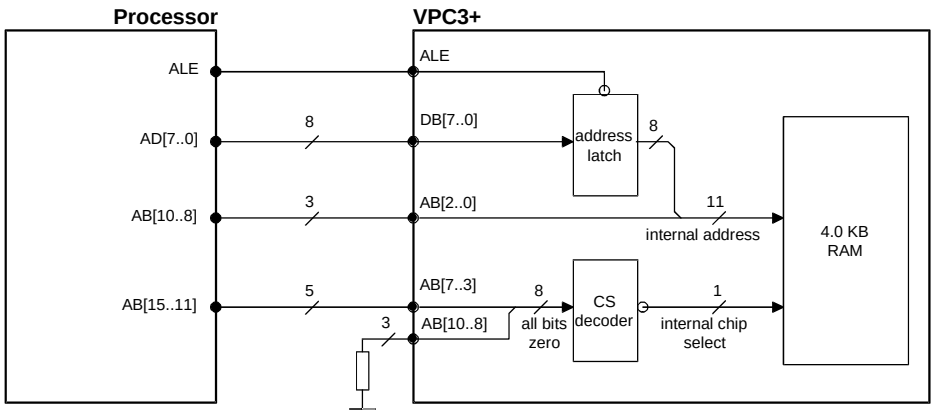


Figure D-60: Internal Chipselect Generation in Synchronous Intel Mode, 2K Byte mode

D-6.1.4 Application with 80C32 (4K Byte RAM)

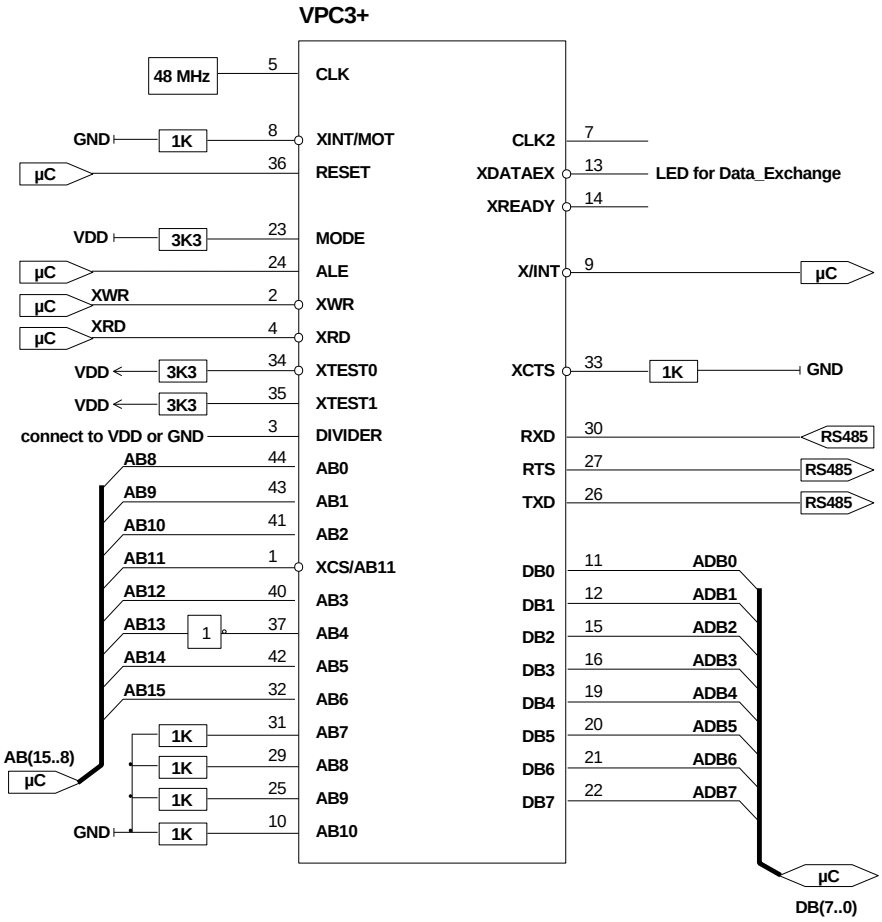


Figure D-61: 80C32 Application in 4K Byte mode

The internal chipselect is activated when the address inputs AB[10..3] of the VPC3+ are set to '0'.
In the example above the start address of the VPC3+ is set to 2000_H.

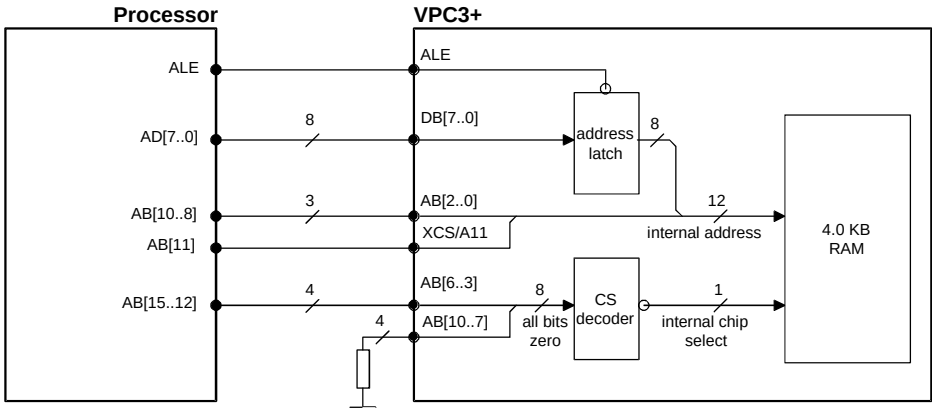


Figure D-62: Internal Chipselect Generation in Synchronous Intel Mode, 4K Byte mode

D-6.1.5 Application with 80C165

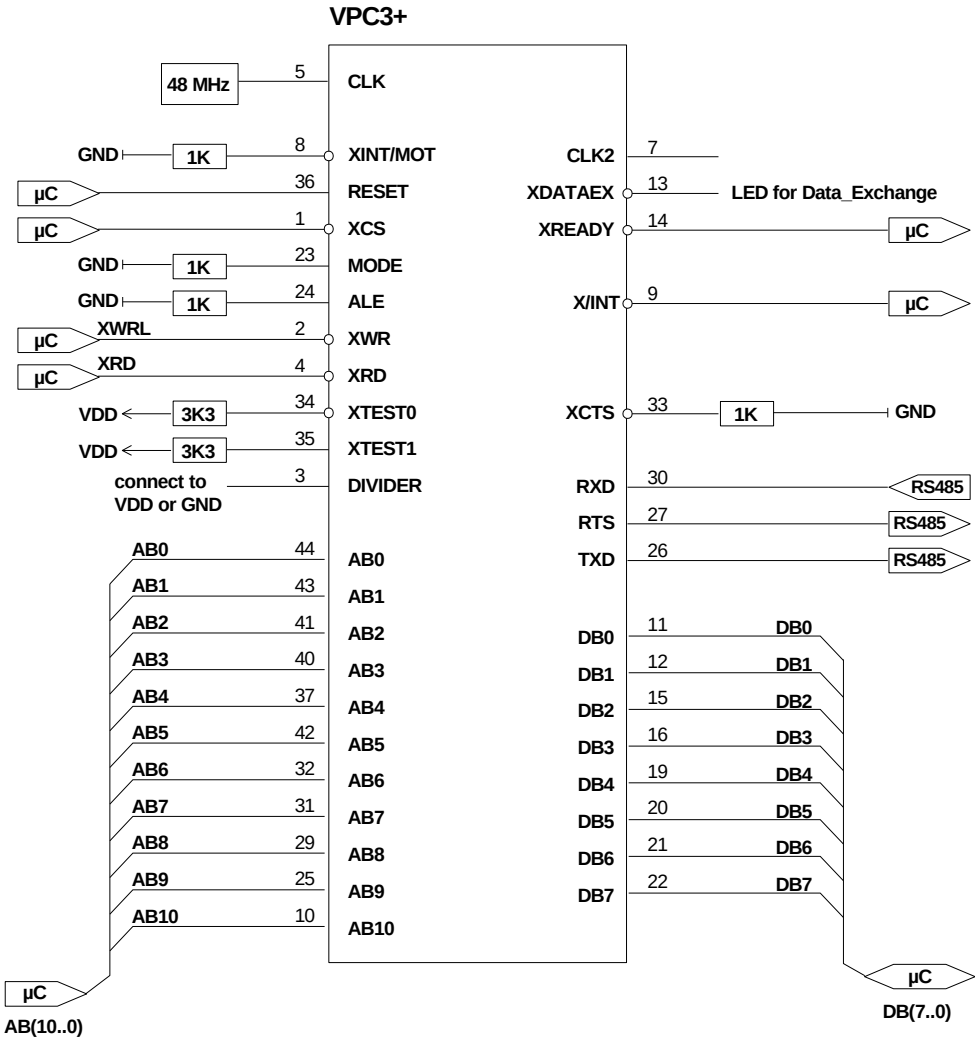


Figure D-63: 80C165 Application

D-6.2 Dual Port RAM Controller

The internal 4K Byte RAM of the VPC3+ is a single-port RAM. However, the integrated Dual-Port RAM controller, permits an almost simultaneous access of both ports (bus interface and microsequencer interface). When there is a simultaneous access from both ports, the bus interface has higher priority. This guarantees the shortest possible access time. If the VPC3+ is connected to a microcontroller with an asynchronous interface, the controller can evaluate the Ready signal.

D-6.3 UART

The transmitter converts the parallel data structure into a serial data flow. Signal Request-to-Send (RTS) is generated before the first character. The XCTS input is available for connecting a modem. After RTS active, the transmitter must hold back the first telegram character until the modem activates XCTS. XCTS is checked again after each character.

The receiver converts the serial data flow into the parallel data structure and scans the serial data flow with the four-fold transmission speed. Stop bit testing can be switched off for test purposes ('Dis_Stop_Control = 1' in Mode Register 0 or Set_Prm telegram for DP). One requirement of the PROFIBUS protocol is that no slip is permitted between the telegram characters. The VPC3+ transmitter ensures that this specification is maintained.

The synchronization of the receiver starts with the falling edge of the Start bit. The Start bit is checked again in the middle of the bit-time for low level. The data bits, the Parity and the Stop bit are also scanned in the middle of the bit-time. To compensate for the synchronization error, a repeater generates a ±25% distortion of the stop bit at a four-fold scan rate. In this case the VPC3+ should be parameterized with 'Dis_Start_Control = 1' (in mode register 0 or ‘Set_Prm telegram’ for DP) in order to increase the permissible distortion of the stop bit.

D-6.4 PROFIBUS Pin Assignment

The data transmission is performed in RS485 operating mode (i.e., physical RS485). The MPI12x is connected via the following signals to the galvanically isolated interface drivers.

Signal Name	Input/Output	Function
RTS	Output	Request to send
TXD	Output	Sending data
RXD	Input	Receiving data

Figure D-64: PROFIBUS Signals

The PROFIBUS interface is a 9-way, sub D, plug connector with the following pin assignment.

Pin 1 - Free
Pin 2 - Free (optional M24V Ground)
Pin 3 - B line
Pin 4 - Request to send (RTS)
Pin 5 - Ground 5V (M 5)
Pin 6 - Potential 5V (floating P5)
Pin 7 - Free (optional P24V Power Supply)
Pin 8 - A line
Pin 9 - Free

The cable shielding must be connected to the plug connector housing.
The free pins are described as optional in EN 50170 Vol. 2. If used, they should conform to the specifications in DIN 19245 Part 3.

CAUTION:
The pin names A and B on the plug connector refer to the signal names in the RS485 standard and not the pin names of driver ICs.

Keep the wires from driver to connector as short as possible.

D-6.5 Example of RS485 Interface

To minimize the capacity of the bus lines the user should avoid additional capacities. The typical capacity of a bus station should be 15...25 pF.

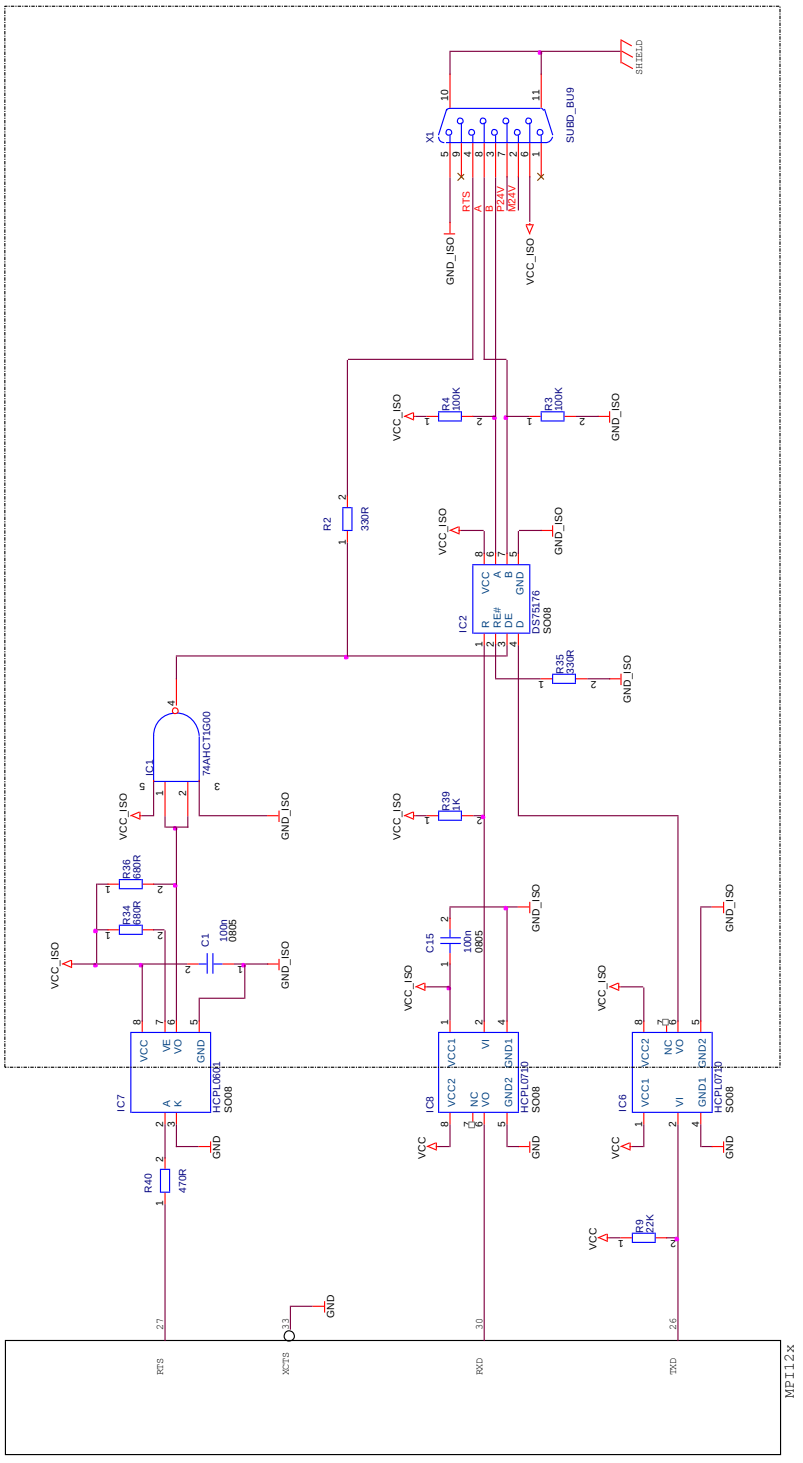


Figure D-65: Example of RS485 Interface

Notes:

Notes:

Revision History

Version	Date	Changes
V1.00	09.07.2004	First release
V2.00	10.08.2005	Document split into several parts Description of VPC3+ Operation Mode and General Chip Description added Several changes in Operational Specification
V2.01	21.12.2005	p 44: SAP_MAX removed p 22: Time (2) replaced by (11) in Figure A-17 p 85: Figure B-33 updated p 162: Figure D-65 updated
V3.00	31.10.2012	p 54: typo in Figure B-12 (Monitor Mode) fixed A-6.7: package data updated A-6.8: processing instructions updated A-6.9: ordering information added

profichip GmbH
Einsteinstrasse 6
91074 Herzogenaurach
Germany

Phone : +49.9132.744-200
Fax: +49.9132.744-2064

www.profichip.com

profichip®
The Clever Alternative