A Synthesizable VHDL Model of the Serial Communication Interface and Synchronous Serial Interface of Motorola DSP56002

Master thesis performed in Eletronics Systems by

Swaroop Mattam

LiTH-ISY-EX--06/3860--SE

Linköping, 30 Aug 2006.

A Synthesizable VHDL Model of the Serial Communication Interface and Synchronous Serial Interface of Motorola DSP56002

Master thesis performed in Electronics System

at Linköping Institute of Technology by

Swaroop Mattam

LiTH-ISY-EX--06/3860--SE

Supervisor:

Examiner:

Thomas Johansson Kent Palmkvist

Linköping, 30 Aug 2006.

| Presentation Date 2006-08-30 Publishing Date (Electronic version) | Departmen Departmen | Department and Division Department of Electrical Engineering Linköping | | | | | | | | | |
|--|-------------------------|--|------------------------|--|--|--|--|--|--|--|--|
| Language Type of Pub | lication | ISBN (Licentiate thesis) | | | | | | | | | |
| X English | e thesis | ISRN: LiTH-ISY-EX06/386 | 60SE | | | | | | | | |
| Other (specify below) Degree th Thesis C X Thesis D- | level level level | Title of series (Licentiate thesis) | | | | | | | | | |
| Number of Pages Report 58 Other (sp | ecify below) | Series number/ISSN (Licenti | ate thesis) | | | | | | | | |
| URL, Electronic Version http://www.ep.liu.se | | | | | | | | | | | |
| Publication Title | | | | | | | | | | | |
| A Synthesizable VHDL Model of the Seri Motorola DSP56002 | al Communica | ation Interface and Synchronou | us Serial Interface of | | | | | | | | |
| Author(s) | | | | | | | | | | | |
| Swaroop Mattam | | | | | | | | | | | |
| | | | | | | | | | | | |
| Abstract | | | | | | | | | | | |
| The design and implementation of a synthesizable model of the Serial Communication Interface and Synchronous Serial Interface, which constitutes the Port C of Motorola DSP56002 is presented in this report. They serves as a full duplex serial interface to other DSPs, processors, codecs, digital-to-analog and analog-to-digital converters and other transducers. The SCI block is able to handle a data rate of 5Mbps in Synchronous mode and 625Kbps in asynchronous mode for a 40MHz clock. It supports five word formats including a multidrop mode for multiprocessor systems. SSI provides a data rate of 10Mbps for the same 40 MHz clock. | | | | | | | | | | | |
| The design includes a programmable on-chip or external baud rate generator/interrupt timer for the SCI and a clock generator and frame Sync generator for the SSI. The thesis focus on arriving at a full functional description of individual blocks included with Port C from the data sheets and product users manual. From this operational description a behavioural model was developed. The structure and implementation is based on the Motorola DSP56002 with additional support for a variable data-width. The model is written completely in behavioural VHDL with a top-down approach and the model was verified and validated. | | | | | | | | | | | |
| Number of pages: 58 | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |

Abstract

The design and implementation of a synthesizable model of the Serial Communication Interface and Synchronous Serial Interface, which constitutes the Port C of Motorola DSP56002 is presented in this report. They serves as a full duplex serial interface to other DSPs, processors, codecs, digitalto-analog and analog-to-digital converters and other transducers. The SCI block is able to handle a data rate of 5Mbps in Synchronous mode and 625Kbps in asynchronous mode for a 40MHz clock. It supports five word formats including a multidrop mode for multiprocessor systems. SSI provides a data rate of 10Mbps for the same 40 MHz clock. The design includes a programmable on-chip or external baud rate generator/interrupt timer for the SCI and a clock generator and frame Sync generator for the SSI.

The thesis focus on arriving at a full functional description of individual blocks included with Port C from the data sheets and product users manual. From this operational description a behavioural model was developed. The structure and implementation is based on the Motorola DSP56002 with additional support for a variable data-width. The model is written completely in behavioural VHDL with a top-down approach and the model was verified and validated.

Acknowledgments

This Master thesis has been written at the Department of Electrical Engineering (ISY), Linköping University as a final work of the International Master Program in SoCware.

I would like to thank my Examiner Prof. Kent Palmkvist, for such an interesting topic, which is a part of the research project in the Department of Elecronic Systems.

I would also like to thank my supervisor Thomas Johansson who is a Research Engineer in the Division of Electronic Systems, for assisting with helpful advices and for solving problems during the work.

My special thanks are to my opponent and friend Krishna Kumar Guda for his suggestions in different aspects of the work.

I wish to thank all the people who were involved in the work during all these weeks, for their help and support.

Table of Contents

| 1 Introduction | |
|---|----|
| 1.1 Background [3] | 1 |
| 1.2 Motivation | |
| 1.3 Structure & Applicability | 1 |
| 1.4 Overview of the Task | 1 |
| 1.5 Design of the system and Tools Used [2] | 2 |
| 1.6 About the document | 2 |
| 2 Port C I/O | 5 |
| 2.1 Introduction | |
| 2.2 Port C I/O Model. | 5 |
| 2.3 On chip Memory Map | |
| 2.4 Register Descriptions | 6 |
| 2.4.1 Port C Control Register | 7 |
| 2.4.2 Port C Data Direction Register | 7 |
| 2.4.3 Port C Data Register | |
| 2.5 Functional Description | 8 |
| 3 Serial Communication Interface | 9 |
| 3.1 SCI Overview | 9 |
| 3.2 SCI Features | 9 |
| 3.3 Pin Description | 9 |
| 3.3.1 Receive Data (RXD) | 9 |
| 3.3.2 Transmit Data (TXD) | |
| 3.3.3 SCI Serial Clock (SCLK) | 10 |
| 3.4 SCI Block Diagram | 10 |
| 3.5 SCI Memory Map | 11 |
| 3.6 SCI Register Descriptions | 11 |
| 3.6.1 SCI Control Register(SCR) | 11 |
| 3.6.1.1 Data Word Format (WDS0 – WDS2) | |
| 3.6.1.1.1 8-bit Synchronous Mode | |
| 3.6.1.1.2 10-bit Asynchronous Mode | |
| 3.6.1.1.3 11-bit Asynchronous mode with Even parity | |
| 3.6.1.1.4 11-bit Asynchronous mode with Odd parity | |
| 3.6.1.1.5 11-bit Multidrop Mode | |
| 3.6.1.2 Data Shift Direction (SSFTD) | 13 |
| 3.6.1.3 Send Break (SBK) | 13 |
| 3.6.1.4 Wakeup Mode(WAKE) | 13 |
| 3.6.1.4.1 Idle Line Wake-Up | |
| 3.6.1.4.2 Address bit Wake-Up | |
| 3.6.1.5 Receiver Wakeup Enable(RWU) | 14 |
| 3.6.1.6 Wired-OR Mode(WOMS) | 14 |

| 4.1 Introduction | 27 |
|---|------|
| 4 Synchronous Serial Interface | . 27 |
| 5.6.5 Address on wake up mode | 23 |
| 2.8.5 Address bit wake up mode | 23 |
| 3.8.3 Multidrop Mode. | 25 |
| 3.8.2 Asynchronous mode | 25 |
| 3.8.1 Synchronous Mode | 24 |
| 3.8 Modes of Operation: | 24 |
| 3.7.4.1 Character Reception | 24 |
| 3.7.4 Receiver. | 23 |
| 3.7.3.2 Preamble and Break Characters | 23 |
| 3.7.3.1 Character Transmission | 23 |
| 3.7.3 Transmitter | 22 |
| 3.7.2 Baud Rate Generator/Interrupt Timer | 20 |
| 3.7.1.3 Data format in the 11 bit Asynchronous Mode | 20 |
| 3.7.1.2 Data format in the 10 bit Asynchronous Mode | 19 |
| 3.7.1 Data Format | 19 |
| 3.7 SCI Functional Description | 18 |
| 3.6.4.2 SCI Transmit Data Registers | 18 |
| 3.6.4.1 SCI Receive Data Registers | 17 |
| 3.6.4 SCI Data Registers | 17 |
| 3.6.3.5 Clock Source(Transmitter) | 17 |
| 3.6.3.4 Clock Source(Receiver) | 17 |
| 3.6.3.3 Clock Prescalar(SCP) | 17 |
| 3.6.3.2 Clock Out Divider(COD) | 17 |
| 3.6.3.1 Clock Divider(CD) | 16 |
| 3.6.3 SCI Clock Control Register | 16 |
| 3.6.2.8 Received Bit 8(R8) | 16 |
| 3.6.2.7 Framing Error(FE) | 16 |
| 3.6.2.6 Parity Error(PE) | 16 |
| 3.6.2.5 Over Run Error(OR) | 16 |
| 3.6.2.4 Idle Line Flag(IDLE) | 16 |
| 3.6.2.3 Receive Data Register Full(RDRF) | 16 |
| 3.6.2.2 Transmit Data Register Empty(TDRE) | 15 |
| 3.6.2.1 Transmitter Empty(TRNE) | 15 |
| 3.6.2 SCI Status Register | 15 |
| 3.6.1.14 Clock Polarity(SCKP) | 15 |
| 3.6.1.13 Timer Interrupt Rate(STIR) | 15 |
| 3.6.1.12 Timer Interrupt Enable(TMIE) | 15 |
| 3.6.1.11 Transmit Interrupt Enable(TIE) | 15 |
| 3.6.1.10 Receive Interrupt Enable(RIE) | 14 |
| 3.6.1.9 Idle Line Interrupt Enable(ILIE) | 14 |
| 3.6.1.8 Transmit Enable(TE) | 14 |
| 3.6.1.7 Receiver Enable(RE) | 14 |
| | |

| 4.2 SCI Features | |
|---|----|
| 4.3 SSI Block Diagram | |
| 4.4 SSI Pin Description | |
| 4.4.1 Serial Transmit Data Pin (STD) | |
| 4.4.2 Serial Receive Data Pin (SRD) | |
| 4.4.3 Serial Clock (SCK) | |
| 4.4.4 Serial Control Pin (SC0) | |
| 4.4.5 Serial Control Pin (SC1) | |
| 4.4.6 Serial Control Pin (SC2) | |
| 4.5 SSI Memory Map | |
| 4.6 SSI Registers | 29 |
| 4.6.1 SSI Control Register A (CRA) | 29 |
| 4.6.1.1 Prescale Modulus Select (PM7–PM0) | |
| 4.6.1.2 Frame Rate Divider Control (DC4–DC0) | |
| 4.6.1.3 Word Length Control (WL0, WL1) | |
| 4.6.1.4 Prescaler Range (PSR) | |
| 4.6.2 SSI Control Register B (CRB) | |
| 4.6.2.1 Output Flag 0 (OF0) | |
| 4.6.2.2 Output Flag 1 (OF1) | |
| 4.6.2.3 Serial Control Direction 0 (SCD0) | |
| 4.6.2.4 Serial Control Direction 1 (SCD1) | |
| 4.6.2.5 Serial Control Direction 2 (SCD2) | |
| 4.6.2.6 Clock Source Direction (SCKD) | |
| 4.6.2.7 Shift Direction (SHFD). | |
| 4.6.2.8 Frame Sync Length (FSL0 and FSL1) | |
| 4.6.2.9 Sync/Async (SYN) | |
| 4.6.2.10 Gated Clock Control (GCK) | |
| 4.6.2.11 Mode Select (MOD) | |
| 4.6.2.12 Transmit Enable (TE) | |
| 4.6.2.13 SSI Receive Enable (RE) | |
| 4.6.2.14 SSI Transmit Interrupt Enable (TIE). | |
| 4 6 2 15 SSI Receive Interrupt Enable (RIE) | 32 |
| 4 6 3 SSI Status Register (SSISR) | 32 |
| 4.6.3.1 Serial Input Flag 0 (IF0). | |
| 4.6.3.2 Serial Input Flag 1 (IF1) | |
| 4.6.3.3 Transmit Frame Sync Flag (TFS). | |
| 4 6 3 4 Receive Frame Sync Flag (RFS) | 32 |
| 4 6 3 5 Transmitter Under run Error Flag (TUE) | 33 |
| 4 6 3 6 Receiver Overrun Error Flag (ROE) | 33 |
| 4 6 3 7 Transmit Data Register Empty (TDF) | 33 |
| 4 6 3 8 Receive Data Register Full (RDF) | |
| 4.6.4 SSI Receive Shift Register & SSI Receive Data Degister (PY) | |
| 4.6.5 SSI Transmit Shift Register & SSI Transmit Data Register (TX) | |
| 4.6.6 Time Slot Register (TSR) | |
| | |

| 4.7 Functional Description | 35 |
|---|----|
| 4.7.1 Clock Generator Unit | |
| 4.7.2 Frame Sync Generator | |
| 4.8 SSI Operation | 38 |
| 4.8.1 Frame Sync Selection | |
| 4.8.2 Clock Selection- Continuous/Gated Clock | |
| 4.8.3 Operating Mode Selection- Normal/Network | 40 |
| 4.8.3.1 Normal Mode | 40 |
| 4.8.3.2 Network mode | 41 |
| 5 System Design, Simulation and Results | 43 |
| 5.1 Port C Design | 43 |
| 5.2 Port C I/O | 44 |
| 5.2.1 Simulations of Port C I/O | 44 |
| 5.3 SCI Design | 44 |
| 5.3.1 Baud Rate Generator | 45 |
| 5.3.1.1 Simulation of Baud Rate Generator Block | 45 |
| 5.3.2 SCI Interface | |
| 5.3.2.1 Simulation of the SCI block | 46 |
| 5.4 SSI Design | |
| 5.4.1 Clock Generator | |
| 5.4.1.1 Simulation of the Clock Generator Block | 50 |
| 5.4.2 Frame Sync Generator | 50 |
| 5.4.2.1 Simulation of the Frame Sync Generator Block | 50 |
| 5.4.3 SSI Interface | |
| 5.4.3.1 Simulation of SSI block | |
| 6 Results & Further Enhancements | 53 |
| 6.1 Design Issues | |
| 6.2 Deviations From the Motorola DSP56000 and Possible Enhancements [1] | 53 |
| 6.3 Results | 54 |
| 6.4 Further Work [4] | 54 |
| References | 55 |
| Appendix A | |
| | |

List of Figures

| Figure 2-1 Port C Interface | |
|--|----|
| Figure 2-2 Port C Pin Control Logic | 6 |
| Figure 2-3 Port C Registers | 7 |
| Figure 2-4 Port C Control Register | 7 |
| Figure 2-5 Port C Data Direction Register | 7 |
| Figure 2-6 Port C Data Register | |
| Figure 2-7 Block level Description of Port C Programming Model | |
| Figure 3-1 SCI Block Diagram | |
| Figure 3-2 SCI Control Register Format | 11 |
| Figure 3-3 8-bit Synchronous Mode | |
| Figure 3-4 10-bit Asynchronous Mode | |
| Figure 3-5 Asynchronous mode with Even parity | 12 |
| Figure 3-6 Asynchronous mode with Odd parity | 13 |
| Figure 3-7 11-bit Asynchronous Multidrop Mode | |
| Figure 3-8 SCI Status Register Format | |
| Figure 3-9 SCI Clock Control Register Format | |
| Figure 3-10 Registers for Receiving SCI Data | |
| Figure 3-11 Registers for Transmitting Data | |
| Figure 3-12 SCI Functional Block Diagram | |
| Figure 3-13 Data Format in Synchronous Mode | 19 |
| Figure 3-14 Data Format in 10-bit Asynchronous Mode | 19 |
| Figure 3-15 Data Format in 11-bit Asynchronous Mode | |
| Figure 3-16 Block Diagram of Baud Rate Generator | |
| Figure 3-17 Block Diagram of SCI Transmitter | |
| Figure 3-18 SCI Receiver Block Diagram | |
| Figure 4-1 SSI Block Diagram | |
| Figure 4-2 SSI Control Register A Format | |
| Figure 4-3 SSI Control Register B Format | |
| Figure 4-4 SSI Status Register Format | |
| Figure 4-5 SSI receive Data Registers | |
| Figure 4-6 SSI Transmit Data Registers | |
| Figure 4-7 SSI Finctional Level Diagram | |
| Figure 4-8 Block Diagram of Clock Generator | |
| Figure 4-9 Block Diagram of SSI Frame Sync Generator | |
| Figure 4-10 Sample format for Mixed frame length and | |
| same frame length for TX and Rx Sections | |
| Figure 4-11 Clock diagram showing the difference between | |
| continuous and gated clock systems | 40 |
| Figure 4-12 Data transmission format in Normal mode | |
| Figure 4-13 Data transmission format in Network mode | 41 |

| Figure 5.1 | Port C Interface | 13 |
|--|--|----|
| $\frac{1}{2} = \frac{1}{2} = \frac{1}$ | | |
| Figure 5-2 | Port C Internal Structure | |
| Figure 5-4 | SCI Structural View | 45 |
| Figure 5-5 | Simulation of Baud Rate Generator Block with a Clock Divider of 2 | 46 |
| Figure 5-6 | SCI Transmitter in Synchronous 8-bit mode of transmission | 46 |
| Figure 5-7 | SCI Receiver in Synchronous 8-bit mode of transmission | 47 |
| Figure 5-8 | SCI Transmitter in asynchronous 11-bit Odd parity | |
| | mode of transmission | 47 |
| Figure 5-9 | SCI Receiver in asynchronous 11-bit Even parity | |
| | mode of transmission | 47 |
| Figure 5-10 | SSI Structural View | 49 |
| Figure 5-11 | Simulation Results of Clock generator | |
| | with a prescale modulo of 2 | 50 |
| Figure 5-12 | Simulation results of frame Sync with word length(8 bits) | |
| | Frame Sync for Rx and Tx | 50 |
| Figure 5-13 | Simulation Results Showing the Gated Clock | |
| | when GCK bit is set to 1 | 51 |
| Figure 5-14 | Simulation run for the receiver part of the SSI with a word length | |
| | of 8 bits and operation is in normal mode with gated clock | 51 |
| Figure 5-15 | Simulation Results showing the transmission of serial data | |
| | in normal mode of operation with a word length of 8 bits | 52 |
| Figure 5-15 | Simulation Results showing the transmission of serial data in normal mode of operation with a word length of 8 bits | 52 |

List of Tables

| 0 |
|------|
| 11 |
| 12 |
| . 22 |
| 29 |
| |

1 Introduction

1.1 Background [3]

The thesis is a part of the on-going implementation of a fixed-point 24 bit DSP processor with scalable architecture, carried out at Electronics Systems Division of Linkoping University. The project is carried out with a sequence of models and adequate refinements in each stage, to make a scalable behavioral model of the kernel that should be instruction compatible with the Motorola DSP56002. The scalability refers to variable data word length and addition or removal of memories and instructions. A top-down design methodology was adopted to design the system, synthesizing and validating at each levels of development. Each design level is further partitioned to sub blocks. The thesis work focus on the design of Port C, which is one of the functional peripheral blocks which describes the Port-C Serial communication Interface, Synchronous Serial interface , their registers and their control.

1.2 Motivation

The device should provide required communication with other DSPs, processors, various industrial standard Codecs, A/D and D/A converters and other peripheral devices like modems. The serial interface can be clocked at a faster rate which helps in attaining a higher data rate. Issues related with cross talk is of less validity and more space due to less cables. So a cost effective communication link as made possible with less number of pins.

1.3 Structure & Applicability

This design pattern is applicable to almost all serial devices which involves direct byte transfers to and from the device using program instructions. In such devices, serial transmission is implemented by the device interrupting the processor for data bytes. When data is received on the serial link, the device interrupts the processor to transfer data and the processor responds.

1.4 Overview of the Task

The purpose of the thesis is to make a VHDL synthesizable model of the Port C of Motorola DSP56002.

Main Tasks:

- Obtaining a complete functional description of the full system, based on the details given in the DSP56000 Users Manual and product data sheets.
- Analyzing the variations of the VHDL model from the actual DSP56000.
- Modeling the complete system in VHDL using the tool FPGA Advantage..
- Simulating the individual blocks and entire design with various test cases.

The guide for the design was the Motorola DSP56002 Users Manual. The implemented design has to be validated through testing. The behavior of the model was validated with the signals and timing details given with the manual and data sheets.

1.5 Design of the system and Tools Used [2]

The design was structured into different levels. Port C forms the top level and various sub-levels including SCI, SSI and Port C I/O follows. This level is again structured into various sub-levels. The designing job was made easy with the HDL designer. A graphical user interface was provided to design the various functional blocks and their descending level hierarchy, along with the signals(wires and buses) interconnecting the various blocks. So a great part of coding effort was reduced by the efficient usage of the tool, to code the interface parts and the whole structure of the system.

The various blocks designed in the HDL designer was simulated individually using the Modelsim, which is a part of FPGA advantage tool box. Macro files(*.do) are used to set various test patterns when simulating and also enforce commands to have a better analysis of the system. The synthesisability of the model was checked by the tool- Precision Synthesis, which is also an addendum of FPGA advantage.

1.6 About the document

Mostly the design was based on the user manual of Motorola DSP56002 and also from some papers which focus on describing design techniques and tools. In addition, used Internet databases to find some information related to programming and design techniques. The references section includes the links referenced in this report.

It must be noticed that the intended reader of this thesis is assumed to have good knowledge in digital system design.

- Chapter 2 provides a general outline of the system, its total structure and interfaces. More details of the Port C I/O comes in this chapter. A better functional description is described which can give a clear outlook on the functionality of the Port C I/O. In addition, more precise definition of the problem is

presented.

- Chapter 3 presents the SCI architecture along with the shared pins and registers involved. A full operational description of the system was derived including all the set and reset conditions of the various flags. The design of the baud rate generator and selection of various baud rates are highlighted.

-Chapter 4 details the SSI block of the Port C. SSI features, pin description and registers and their mapping strategy are outlined. A complete operational model which can be easily transferred in to VHDL code is described in this chapter. The details of the design of clock generator and frame synchronization block is given.

- Chapter 5 describes the system design details. Various levels in the design hierarchy is described in detail. The different sections of this chapter was first decided to be included with chapters 2, 3 and 4, but for the ease of understanding the design of the system as a whole, design should be detailed after the theoretical back ground and design specifications.

- Chapter 6 concludes with the implementation by simulating the VHDL models for various test conditions and also introducing the experimental results based on testing some design examples. This chapter also presents some deviations of the thesis work from the actual DSP model, some

conclusions and proposes suggestions for future work.

- References section gives the books or journals or Internet resources which were used while compiling the report and doing the project work, which can give a better insight into the work.

- Appendix A contains a glossary of terms used in this report.

2 Port C I/O

2.1 Introduction

Port C is one of the I/O ports of DSP56002 which works as a general purpose input-output or as a serial communication port. Of the total nine pins three pins are used as pins for serial communication interface and rest six pins are used for synchronous serial interface. While the pins are not configured as serial interfaces, it can be used as general purpose I/O. The general structure of Port C is given in figure 2-1.



Figure 2-1 Port C Interface [1]

2.2 Port C I/O Model

This is a high level description of the Port C module which gives insight into the configuration settings of an I/O pin of Port C. The switching strategy is made according to the bit patterns in the Port C registers. More-over control is also made from the peripherals area. But the thesis part concentrates on the configuration of Pin based on the Port C register settings. Figure 2-2 states the general description of the logic for control of a pin.



Figure 2-2 Port C Pin Control Logic [1]

2.3 On chip Memory Map

Port C is memory mapped and a standard MOVE instruction can be used to transfer the data between Port C and a register. The memory map for the Port C I/O is shown in the table 2-1.

| Reg. | Addr | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|-------|----------|------|------|------|------|------|------|------|------|
| PCD | X:\$FFE5 | PD7 | PD6 | PD5 | PD4 | PD3 | PD2 | PD1 | PD0 |
| PCDDR | X:\$FFE3 | CD7 | CD6 | CD5 | CD4 | CD3 | CD2 | CD1 | CD0 |
| PCC | X:\$FFE1 | CC7 | CC6 | CC5 | CC4 | CC3 | CC2 | CC1 | CC0 |

2.4 Register Descriptions

The I/O pins of the Port C is controlled by three registers Port C control register (PCC), Port C data direction register (PCDDR), and Port C data register (PCD). These registers are mapped to the memory. The register bit fields and their corresponding pins are represented in the figure 2-3.

| | | ENABLED BY BITS IN X:\$FFE1 | DIRECTION SELECTED BY X:\$FFE3 | INPUT/OUTPUT DATA REGISTER X:\$FFE5 |
|---------|---------|-----------------------------------|--------------------------------------|---|
| | PC0 < | | CD0 | PC0 |
| _ | PC1 🗲 | | CD1 | PC1 |
| P | PC2 < | → CC2 | CD2 | PC2 |
| 0 | PC3 🗲 | | CD3 | PC3 |
| к т | PC4 < | | CD4 | PC4 |
| · · | PC5 🗲 🗕 | → CC5 | CD5 | PC5 |
| 0 | PC6 < | | CD6 | PC6 |
| Ŭ | PC7 < | | CD7 | PC7 |
| | PC8 🔫 | | CD8 | PC8 |
| | | | | |

Figure 2-3 Port C Registers [1]

2.4.1 Port C Control Register

Nine least significant bits in the PCC will determine on the configuration of corresponding pin of the Port C as GPIO or Serial interface pin. If the bit is set to one for GPIO and zero for serial peripheral interface pin. The register is mapped to memory location X:\$FFE1.

| 23 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----|--------|----|----|----|----|----|----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | CC8 | CC7 | CC6 | CC5 | CC4 | CC3 | CC2 | CC1 | CC0 |

STD SRD SCK SC2 SC1 SC0 SCLK TXD RXD

Port C Pin Control CCx = 0 -> General Purpose I/O Pin 1 -> On-chip Peripheral Pin



2.4.2 Port C Data Direction Register

The bit corresponding to the Port C pin will determine the pin is to be configured as input pin, when bit value is zero, and output, when bit value is one. PCDDR bits are memory mapped as a 24 bit word to X:\$FFE3.

23 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

Port C Data Direction Control CDx = 0 -> Input Pin 1 -> Output Pin



2.4.3 Port C Data Register

PCD treats the data which is to be transmitted or received through the Port C. data register is mapped to X:\$FFE5

| 23 | 15 | 5 1 | 4 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 3 | 2 | 1 | 0 |
|--------|------------|--------|--------|--------|--------|--------|--------|---------|---------|---------|---------|---------|---------|---------|---------|-----|---|---|
| 0 * | 0 * | 0 * | 0 * | 0 * | 0 * | 0 * | 0 * | PD 8 | PD 7 | PD 6 | PD 5 | PD 4 | PD 3 | PD 2 | PD 1 | PD0 | | |

Figure 2-6 Port C Data Register

2.5 Functional Description

This section provides the operation of the Port C I/O based on which design is made. The pin control logic is given in the figure 2-7. The design was made simple using multiplexers for selecting between GPIO and Serial pins and selection between input and output positions for the pin. When configured as an input the tristate buffer will be in a high impedance state thereby disabling data bit transfer to the pin. But when configured as an output data from PCD is transferred to the pin.



Figure 2-7 Block level Description of Port C Programming Model

3 Serial Communication Interface

3.1 SCI Overview

The Serial Communications Interface (SCI) module is a relatively slow, full duplex communication port that is widely used to communicate with other processors, peripheral devices and embedded systems. SCI interface follows most of the serial communication standards, for example RS232C, RS422 standards. The SCI is more widely used to communicate over longer distances, ideal for off board communications. The SCI supports asynchronous and synchronous data transfer. It consists of separate receiver and transmitter sections whose operations are asynchronous with respect to each other. SCI makes use of three pins of the Port C – PC0-PC2. As an asynchronous interface it supports a maximum data rate of 625Kbps and while in synchronous mode it acts as a fast interface with data rate of 5Mbps. The asynchronous mode can support five different data word formats including the Multidrop mode in case of systems having more processors.

3.2 SCI Features

- Full Duplex Communication
- Standard non-return-to-zero format(asynchronous mode)
- On chip baud rate generator with 12-bit Baud rate selection & Programmable Interrupt timer.
- Asynchronous Interface(maximum data rate of 625Kbps)
- Synchronous Interface(maximum data rate of 5.0Mbps)
- Five Different Data Formats
 - 8-bit synchronous
 - 10-bit asynchronous
 - 11-bit asynchronous with odd parity
 - 11-bit asynchronous with even parity
 - 11-bit multidrop mode(idle line wake up and address bit wake up modes)
- Frame error, parity error and over run error detection.
- Interrupt Driven Operation.

3.3 Pin Description

The SCI physically consists of three pins; a Serial clock input-output pin(SCLK) and independent Receiver (RXD) and Transmitter(TXD) pins shared with Port C PC0-PC2. The three pins reserved for the SCI can be configured as GPIO or serial I/O pin depending upon the bits value in the PCC register. As SCI is having separate transmit and receive pins, while in serial transmit mode the receive pin can simultaneously act as a GPIO pin.

3.3.1 Receive Data (RXD)

The RXD pin is an input pin which receives the data in serial format which is to be transferred to the receive shift register as bytes. The clock polarity bit of the SCI control register will determine whether the incoming data is sampled on the negative edge of the SCLK or positive edge depending upon the clock polarity bit value , which is set as 1 or 0 respectively. While not in use as serial interface the pin function as the general purpose PC0 input-output pin.

3.3.2 Transmit Data (TXD)

The TXD pin is an output pin which transmits the data in serial format from the transmit shift register. The clock polarity bit of the SCI control register will determine whether the change in value of data to be transmitted is on the negative edge of the SCLK or positive edge depending upon the clock polarity bit value , which is set as 0 or 1 respectively. The pin will act as general purpose PC1 input-output pin when TXD function is not used.

3.3.3 SCI Serial Clock (SCLK)

The SCLK is an Input-output pin which determines the data transmission or reception of data in synchronous mode. In asynchronous mode the different baud rates are derived based on the scaling bits in SCI clock control register. SCLK may be programmed as a general-purpose I/O pin (PC2) when the SCI SCLK function is not being used. The SCLK pin can be used as GPIO when SCI is operating in asynchronous mode as data transfer is based on the derived baud rates.

3.4 SCI Block Diagram

This section briefs the different operational units in the SCI module and the data transfer in between the different blocks. The general block diagram for the SCI is given in the figure 3-1.



Figure 3-1 SCI Block Diagram

3.5 SCI Memory Map

The memory map for the full SCI module is listed in the table 3-1. When functioning as a receiver the data registers will be in read only mode and while transmitter is enabled, data registers will be write only.

| Memory Address | SCI Register | Bit Pattern |
|--------------------------|------------------------------|---|
| X:\$FFF0 (READ/WRITE) | SCR 24 - bit | The values in the bit positions 0-15 controls the SCI module which includes WDS(0-2), SSFTD, SBK, WAKE, RWU, WOMS, RE, TE, ILIE, RIE, TIE, TMIE, STIR, SCKP respectively and the positions 16-23 are masked with zeros. |
| X:\$FFF1 (READ ONLY) | SSR 24 - bit | The values in the bit positions 0-7 sets the status of SCI module which includes TRNE, TDRE, RDRF, IDLE, OR, PE, FE, R8 respectively and the positions 8-23 are masked with zeros. |
| X:\$FFF2 (READ/WRITE) | SCCR 24 - bit | The bit values in the locations 0-15 will decides clocking strategy of SCI which includes CD(0-11), COD, SCP, RCM, TCM respectively and the positions 16-23 are masked with zeros. |
| X:\$FFF3 (WRITE ONLY) | STXA 24 - bit | The lower byte will hold the transmit data address in the multidrop mode and the higher 2 bytes are filled with zeros. |
| X:\$FFF4 (READ/WRITE) | SRX/STX (LOW) 24 - bit | The lower byte will hold the transmit/receive data and the lower 2 bytes are filled with zeros. |
| X:\$FFF5 (READ/WRITE) | SRX/STX (MID) 24 - bit | The middle byte will hold the transmit/receive data and the higher and lower bytes are filled with zeros. |
| X:\$FFF6 (READ/WRITE) | SRX/STX (HIG) 24 - bit | The higher byte will hold the transmit/receive data and the lower 2 bytes are filled with zeros. |

Table 3-1 SCI Memory Mapping

3.6 SCI Register Descriptions

The SCI comprises of six registers: SCI control register (SCR), SCI status register (SSR), SCI clock control register (SCCR), SCI receive data registers (SRX), SCI transmit data registers (STX), and the SCI transmit data address register (STXA). These can be further grouped as three types : 1) control – SCR, SCCR 2) status – SSR and 3) data transfer – SRX, STX, and STXA. This section describes the registers in th address order. More details of each register in the bit and the function of each bit is described in the sub sections.

3.6.1 SCI Control Register(SCR)

The SCR is a 24-bit read/write register that controls the serial interface operation. Each bit is described in the following sections. The SCR is mapped to memory location X:\$FFF0.

| 23 | 3 | 15 | 14 | 13 | 12 | 11 | 10 |) | 9 | 8 7 | 7 6 | 5 | 4 | 3 | 2 1 | 0 |
|----|---|------|------|------|-----|-----|------|----|----|------|-----|------|-----|-------|------|--------|
| 0* | · | SCKP | STIR | TMIE | TIE | RIE | ILIE | TE | RE | WOMS | RWU | WAKE | SBK | SSFTD | WDS0 | - WDS2 |

Figure 3-2 SCI Control Register Format

3.6.1.1 Data Word Format (WDS0 – WDS2)

The three word-select bits (WDS0, WDS1, WDS2) select the format of the transmit and receive data. The formats include three asynchronous, one multidrop asynchronous mode, and an 8-bit synchronous (shift register) mode. The asynchronous modes are compatible with most UART-type serial devices and support standard RS232C communication links. Synchronous mode acts as a fast serial to parallel or parallel to serial shift register.

| Mode | WDS2 | WDS1 | WDS0 | Data Format |
|------|------|------|------|----------------------------------|
| 0 | 0 | 0 | 0 | 8-bit Synchronous |
| 2 | 0 | 1 | 0 | 10 bit Asynchronous |
| 4 | 1 | 0 | 0 | 11 bit Asynchronous(Even parity) |
| 5 | 1 | 0 | 1 | 11 bit Asynchronous(Odd parity) |
| 6 | 1 | 1 | 0 | 11 Bit Multidrop |

Table 3-2 Data Format Selection

3.6.1.1.1 8-bit Synchronous Mode

In this mode serial data is shifted in synchronous with the clock to the 8 bit shift register.

| TX (SSFTD = 0) | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 |
|-------------------|----|----|--------|--------|----------|--------|----|----|
| | - | | ONE BY | E FROM | SHIFT RE | GISTER | | |

Figure 3-3 8-bit Synchronous Mode

3.6.1.1.2 10-bit Asynchronous Mode

Data format includes a low start bit followed by one byte data and at the end appended with high stopped bit.



Figure 3-4 10-bit Asynchronous Mode

3.6.1.1.3 11-bit Asynchronous mode with Even parity

When even parity is selected, an even number must result from the calculation performed at both ends of the line or an error in transmission has occurred.

| TX (SSFTD = 0) | START BIT | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 OR DATA TYPE | EVEN PARITY | STOP BIT |
|-------------------|--------------|----|----|----|----|----|----|----|-----------------------|----------------|-------------|
|-------------------|--------------|----|----|----|----|----|----|----|-----------------------|----------------|-------------|

Figure 3-5 Asynchronous mode with Even parity

3.6.1.1.4 11-bit Asynchronous mode with Odd parity

When odd parity is selected, the transmitter will count the number of bits in the data word. If the total is not an odd number, the parity bit is made equal to one and thus produces an odd number. If the receiver counts an even number of ones, an error in transmission has occurred.



Figure 3-6 Asynchronous mode with Odd parity

3.6.1.1.5 11-bit Multidrop Mode

In multidrop mode 10th bit in the data format will indicate the received character is and address byte or data byte.



Figure 3-7 11-bit Asynchronous Multidrop Mode

3.6.1.2 Data Shift Direction (SSFTD)

The SSFTD bit in the control register decides on whether to shift the data in to the receive shift register or transmit from the transmit shift register with MSB first or LSB first. If SSFTD is set data is shifted LSB first and if SSFTD is cleared data is shifted in or out LSB first.

3.6.1.3 Send Break (SBK)

Send Break bit is used to signal an unusual condition or to pass a message like error in transmission word or format. The SBK bit is set and cleared to signal the condition and at that stage a frame of 10 or 11 zeros are transmitted depending upon the word format chosen (11- bit or 10- bit). IF SBK is set when transmitter is actively sending data bits, it first completes the the transmission of data and then comes back to state of transmission or idle state. The break bit frame doesn't have designated start or stop bits it will be a frame of zeros. As the start bit is a zero bit to make a valid start bit , after sending the break frame the transmitter will send a minimum of one high bit.

3.6.1.4 Wakeup Mode(WAKE)

3.6.1.4.1 Idle Line Wake-Up

Idle line wake up is selected when WAKE bit is cleared. An idle string of 10 or 11(based on the word format selected) consecutive 1's will be send between two messages. As in the asynchronous mode each data byte in a message consists of low start bit, its not possible to forward idle line frames within a message. So data bytes are separated by start bits and stop bits and messages are separated by a preamble of ones.

3.6.1.4.2 Address bit Wake-Up

The device will work in address bit wake up mode when WAKE bit is set. In the address bit wake up mode all the processors in the sleeping state has to compare the received address with its own and decide to accept or ignore the characters. The SCI receiver in the address bit wake up mode is re enabled when the 8th bit of the received data in 10-bit asynchronous format or 9th bit in the 11-bit data format is one.

3.6.1.5 Receiver Wakeup Enable(RWU)

Receiver wake up enable finds use in asynchronous multidrop mode where the decision has to be made on whether the message is intended for the particular device in the multidrop network. When this bit is cleared receiver is put to sleep and when bit is one receiver will be in wake up mode. When idle line wake up mode is selected the the receiver will respond only when idle line is detected and in address bit wake up mode the receiver will respond when the 10th data bit is detected as one.

3.6.1.6 Wired-OR Mode(WOMS)

WOMS bit finds application when SCI device is connected with other processors in master slave mode or in multidrop network(serial) mode. In this case the transmitter pin uses and active internal pullup when connected to another TXD pin if WOMS bit is zero. But when WOMS bit is set TXD pin must search for an external pull up resistor while connecting to other processors.

3.6.1.7 Receiver Enable(RE)

RE is the bit used to enable or disable the receiver part f the SCI. As the receiver and transmitter registers are mapped to the same location in the memory it is necessary to disable/enable the receiver part as desired. But once receiver starts receiving data and when RE bit is cleared the receiver finish receiving the data byte before going to the disabled state. When the RE bit is cleared the receiver register is disabled to receive further incoming data bits.

3.6.1.8 Transmit Enable(TE)

When TE is set, the transmitter is enabled. When TE is cleared, the transmitter will complete transmission of data in the SCI transmit data shift register; then the serial output is forced high (idle). Data present in the SCI transmit data register (STX) will not be transmitted. STX may be written and TDRE will be cleared, but the data will not be transferred into the shift register. TE does not inhibit TDRE or transmit interrupts.

3.6.1.9 Idle Line Interrupt Enable(ILIE)

When ILIE is set, the SCI interrupt occurs when IDLE is set. When ILIE is clear, the IDLE interrupt is disabled. When a start bit is detected an interrupt request will be generated. At that time both the IDLE nd ILIE bit should be set. The status of the interrupt is determined by the shift register Idle interrupt flag which is a flag outside the SCI, communicates with the interrupt service routine controller. The interrupt request will be cleared upon the reception of the acknowledgment.

3.6.1.10 Receive Interrupt Enable(RIE)

RIE bit along with RDRF bit will decide on receive interrupts. RIE bit will enable the SCI receive interrupt. When RIE is set and receive data register is full then an interrupt request fro data reception is send to interrupt controller. When RIE is zero, then all the receive interrupts are disabled.

3.6.1.11 Transmit Interrupt Enable(TIE)

TIE bit along with TDRE bit will decide on transmit interrupts. RIE bit will enable the SCI transmit data interrupt. When TIE is set and transmit data register is empty then an interrupt request for data reception is send to interrupt controller. When TIE is zero, then all the transmit interrupts are disabled.

3.6.1.12 Timer Interrupt Enable(TMIE)

When TMIE bit is set, then SCI timer interrupts will be send to the interrupt controller. The frequency of interrupt requests will depends on the value of the scaling bits in the clock control register. When the SCI device is not used as as dedicated serial interface the baud rate generator can be used as interrupt generator block. Same baud rate generator can be used as interrupt timer when interrupts has to be generated in the same frequency as of the baud clock.

3.6.1.13 Timer Interrupt Rate(STIR)

STIR controls the rate of generation of timer interrupts. When STIR bit value is one then SCI will not scale the time further for better resolution. But when the STIR bit is cleared then the interrupt rate is increased by 32 times. SO the clock id passed through a divide by 32 routine.

3.6.1.14 Clock Polarity(SCKP)

The polarity of the SCLK will be set to negative or positive dependent on the value of SCKP be 1 or zero respectively.

3.6.2 SCI Status Register

SSR is a 24 bit register which is read only. The lower byte holds the stats bits and the higher bytes are filled with zeros. Each bit in the status register determines the staus of various SCI signals.

| 23 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----|--------|----|----|----|----|----|----|----|----|----|----|----|------|------|------|------|
| 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | R8 | FE | PE | OR | IDLE | RDRF | TDRE | TRNE |

Figure 3-8 SCI Status Register Format

3.6.2.1 Transmitter Empty(TRNE)

TRNE flag indicates the presence of absence of data to be transmitted. The flag is set when both transmit data register and transmit shift register are empty. At this stage data written into the HI,MID or LOW location of the data register or transmit data address register can be transferred to the transmit shift register. The TRNE bit is cleared when data is already written into the transmit data register or to the transmit data address register or in case during the transmission of preamble of zeros or ones. This bit act as a suitable protocol for data transfer which indicates that the next data to be transmitted is on the way of being written in to the transmit register.

3.6.2.2 Transmit Data Register Empty(TDRE)

TDRE status states the transmit data register is filled with data to e transmitted or empty. When the status flag is set the data register will be waiting to receive valid data. At this time data to be transmitted can be written to transmit data register or transmit data address register. When the data register is written with data then status flag is cleared.

3.6.2.3 Receive Data Register Full(RDRF)

RDRF bit is cleared when the data written t the receive data register is being read. In the same way the status of the flag is set when a valid data byte is written completely into the receive data register.

3.6.2.4 Idle Line Flag(IDLE)

IDLE line shows the status of the idle state. While receiving the preamble or break signal of consecutive ones or zeros the IDLE flag will be set. When the start bit of the data is detected by a low bit after the preamble frame IDLE bit will be cleared.

3.6.2.5 Over Run Error(OR)

Over run condition arises when there is a byte to be transferred to the receive data register and at the same time receive data register is filled with data. The OR flag will be set at this time. The over run error has more priority than frame error and parity error. So OR flag automatically clears th frame error and parity error flags.

3.6.2.6 Parity Error(PE)

When data is received in the asynchronous mode with odd or even parity, a parity checker is activated which will check for the parity bit balances with the actual parity of the received data byte. Any deviation will be indicated as a parity error which is asserted by setting the PE flag.

3.6.2.7 Framing Error(FE)

While reception of data in the asynchronous format, if tit was not able to detect any stop bit then there is some error in the data frame received which is indicated by setting the FE flag. After the FE bit being set further transfer of data to the receive data register will be blocked till the bit is cleared. If the data format is synchronous or otherwise if over run error is detected frame error flag will be cleared.

3.6.2.8 Received Bit 8(R8)

R8 bit indicates the status of the 8th data bit in the 11-bit multidrop mode. When R8 is set the data type bit indicates the byte received represents the address and when R8 is cleared the character received will be data. R8 is not affected by reading the SRX or status register.

3.6.3 SCI Clock Control Register

The SCCR is a 24-bit read/write register, whose lower two bytes holds the clock control bits and the higher byte is filled with zeros, which controls the selection of different clock modes and baud rates for the transmit and receive sections of the SCI interface.

| 23 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----|---------|-----|-----|-----|----|----|----|----|----|----|----|----|----|----|----|----|
| 0* | TCM | RCM | SCP | COD | CD |

Figure 3-9 SCI Clock Control Register Format

3.6.3.1 Clock Divider(CD)

Clock divider consists of 12 bits which scales the oscillator clock by enabling a counter. The counter divides the clock by a factor denoted by the value of the clock divider set by 12 bits. The counter valu will be decremented by one each positive edge of the main clock, which truly divides the clock by 2 during each decrement. When the counter value reaches 0, then again the counter is

set with the actual counter value. So a lowest division rate, divide by 4096 is possible.

3.6.3.2 Clock Out Divider(COD)

The function of the clock divider bit depend on the modes of operation. If SCI is in synchronous mode then output divider is fixed to divide by 2. In asynchronous mode, when the COD bit is zero then bfore letting the clock to the SCLK as output the clock is divided by 16, else when COD bit is one clock is fed directly to the SCLK pin as output.

3.6.3.3 Clock Prescalar(SCP)

When the SCP bit is set the clock is prescaled by a factor of 8 and when SCP bit is cleared the clock is bypassed the prescaling stage.

3.6.3.4 Clock Source(Receiver)

RCM is set to one to select the external clock from the SCLK pin and when RCM is cleared receiver will select the internal clock fro data reception.

3.6.3.5 Clock Source(Transmitter)

TCM is set to one to select the external clock from the SCLK pin and when TCM is cleared transmitter will select the internal clock for data reception.

3.6.4 SCI Data Registers

The SCI data registers are divided into two groups: receive and transmit. There are two receive registers – a receive data register (SRX) and a serial-to-parallel receive shift register. There are also two transmit registers – a transmit data register (called either STX or STXA) and a parallel-to-serial transmit shift register. The STX/SRX data are mapped to memory locations X:\$FFF4, X:\$FFF5, X:\$FFF6 and TSXA is mapped to X:\$FFF3.

Data register is a 24 bit register, for the ease or packing and unpacking is segregated into three bytes- Hi(mapped to higher byte of X:\$FFF4), Mid(mapped to middle byte of X:\$FFF5) and Low(mapped to lower byte of X:\$FFF6).Location X:\$FFF3 should be written in the 11-bit asynchronous multidrop mode when the data is an address and it is desired that the ninth bit (the address bit) be set.

3.6.4.1 SCI Receive Data Registers

The data bits received in different formats through the RXD pin are shifted into the 8 bit receive shift register. Once the complete byte is received, it is transferred to SRX. So the serial data is converted to parallel form.



Figure 3-10 Registers for Receiving SCI Data [1]

3.6.4.2 SCI Transmit Data Registers

The data bits to be transmitted through the RXD pin are shifted into the 8 bit transmit shift register. Once the complete byte is shifted into the register, it is transferred serially through the TXD pin. So the parallel data is converted to serial form. In the 11-bit asynchronous multidrop mode when the address bit is set the address will be transferred to the STXA register.





3.7 SCI Functional Description

This section gives the complete functionality of the SCI module based on which the entire module is designed. The details of the operation of the design from the customer point of view is given in sub sections. The following figure 3-12 shows the structure of the SCI module. The SCI transmitter and receiver operates independently although they share the common clock from the baud rate generator ans also the same memory mapped data registers. The central processors keeps track of the SCI interface and writes into the memory the data to be transmitted and processes the received serial data.


Figure 3-12 SCI Functional Block Diagram

3.7.1 Data Format

SCI support different data formats in asynchronous and synchronous modes, which is determined by the WDS bits. The different data formats are given in the figures given below.

3.7.1.1 Data format in the 8-bit Synchronous Mode



Figure 3-13 Data Format in Synchronous Mode[1]

3.7.1.2 Data format in the 10 bit Asynchronous Mode



Figure 3-14 Data Format in 10-bit Asynchronous Mode [1]

3.7.1.3 Data format in the 11 bit Asynchronous Mode



Figure 3-15 Data Format in 11-bit Asynchronous Mode[1]

Synchronous mode is selected by the clearing the bits in the position WDS 0-2. In synchronous mode, data is transmitted/received as bytes, which is sampled on the rising edge of each synchronous clock. In the asynchronous mode each data character is contained in a frame that includes a start bit, the data byte and a stop bit(so a total of 10 bits). In the 11bit asynchronous format an additional bit is included in the frame before th stop bit which indicates the parity bit in the parity mode and address bit in the multidrop mode.

3.7.2 Baud Rate Generator/Interrupt Timer

The programmable baud rate generator provides both the receiver and the transmitter with the baud rate clock which is a bit-period clock. The baud rate generator is controlled by the SCCR. This block will generate the internal clock and also establish a periodic interrupt timer. The baud rate is derived from

the oscillator frequency after scaling with the time base from the clock divider bits in the SCCR.



Eq: Bit rate in Synchronous mode fo

BPS = $64 \times (7(SCP) + 1) \times CD + 1)$

Eq: Bit rate in Asynchronous mode

 $BPS = \otimes \overline{x (7(SCP) + 1) \times CD + 1)}$

where SCP = 0 or 1, CD = 0 to 4096(\$FFF)

The crystal oscillator frequency is first divided by 2 before division by the clock divider bits. The clock divider bits CD0-CD11 scales the frequency to a maximum of 4096 depending upon the total value of 12 bits in the SCCR register. The scaled frequency is further reduced by a factor of 1 or 8 depending upon the value of SCP bit is 0 or 1 respectively. This scaled clock can be used as interrupt timer or an internal clock. But while using as a interrupt timer the clock is again divided by 2 and then by 16 if STIR is set otherwise directly the clock act as timer interrupt without anymore scaling. The clock is further divided by a fixed factor 2 to form the internal clock. The same internal clock is used to derive the serial clock, SCLK. In the synchronous mode the internal clock is divided by 2 to get the SCLK, but in asynchronous mode if the COD bit is set the internal clock will be divided by 16 and if COD bit is cleared then internal clock will be act directly as the SCLK. Before setting the SCLK the polarity of the clock is set to positive if SCKP bit is 0 and negative if SCKP bit position is cleared.

| Synchronou | us Mode for a | a module clock of | 32.768 MHz | Asynchro | nous Mod | e for a module clo | ock of 40Mhz |
|--------------------|---------------|----------------------------|---------------------|-------------------|------------|----------------------------|----------------------------|
| Baud Rate (BPS) | SCP Bit | Divider Bits (CD0-CD11) | Baud Rate Error, | Bit Rate (BPS) | SCP Bit | Divider Bits (CD0–CD11) | Bit Rate Error, Percent |
| (5. 0) | | (020 0211) | Percent | 625.0K | 0 | \$000 | 0 |
| 4.096M | 0 | \$000 | 0 | 56.0K | 0 | \$00A | +1.46 |
| 128K | 0 | \$01F | 0 | 38.4K | 0 | \$00F | +1.72 |
| 64K | 0 | \$03F | 0 | 19.2K | 0 | \$020 | -1.36 |
| 56K | 0 | \$048 | -0.195 | 9600 | 0 | \$040 | +0.16 |
| 32K | 0 | \$07F | 0 | 8000 | 0 | \$04D | +0.15 |
| 16K | 0 | \$0FF | 0 | 4800 | 0 | \$081 | +0.15 |
| 8000 | 0 | \$1FF | 0 | 2400 | 1 | \$020 | -1.38 |
| 4000 | 0 | \$3FF | 0 | 1200 | 1 | \$040 | +0.08 |
| 2000 | 0 | \$7FF | 0 | 600 | 1 | \$081 | 0 |
| 1000 | 0 | SFFF | 0 | 300 | 1 | \$103 | 0 |

 Table 3-3 Sample Baud rates in Asynchronous and Synchronous Modes[1]

3.7.3 Transmitter

This section describe the design of the transmitter section in detail. The block diagram illustrates the total structure of the transmitter in the SCI Module.



Figure 3-17 Block Diagram of SCI Transmitter

3.7.3.1 Character Transmission

The SCI transmitter accommodates 8 bit data characters. The values of the WDS bits determines the format of data transmission. In order to transmit the data the data is first written into the data registers(STX/STXA). The data from the data registers is the transmitted into the transmit shift register. The data in the shift register is then shifted out through the TXD pin, after appending with a stop bit and preface with a start bit. The TDRE bit is set when the shift register is loaded with the data to be transmitted, from the SCI data register. This indicates that SCI register is now ready to be written with next data to be transmitted from the internal data bus. In order to prevent the gap between the words, while writing to data registers, TDRE goes high halfway into the transmission of bit 1 until the middle of bit 6. While initiating the transfer of message, to make the process easy, TRNE flag is set to indicate that there is no data in either data register of shift register. If the interrupt enable bit is also set, then TDRE generates an interrupt request.

3.7.3.2 Preamble and Break Characters

When the TE bit is toggled from 0 to 1, then transmit shift register is automatically loaded with a preamble of 10 or 11 ones, depending on the status of WDS bits. After the preamble shifts out, transmitter is ready to accept data from internal data bus. The hardware also generates parity bit and is appended in the parity bit position while parity mode is selected. When the SBK bit is set, the transmit shift register is loaded with 10 or 11zeros, based on the WDS bit's status. After the SBK bit is cleared, before transmitting the original data from the shift register, one high bit is transmitted to identify the start bit of the next frame.

3.7.4 Receiver

This section gives an independent idea of the functionality or the SCI receiver. The block diagram below describes the design of the receiver part.



Figure 3-18 SCI Receiver Block Diagram

3.7.4.1 Character Reception

The status of the WDS bits indicates the format in which data is received. During the SCI reception, the receive shift register is shifts in serial data bits through the RXD pin. After a complete frame shifts in to the shift register, the data portion of the frame transfers to the data register(SRX). At that instant the RDRF flag is set to indicate that, its time to read the character written into the data register. In the asynchronous multidrop mode, R8 bit will indicate that the received character is address or data. If the receive interrupt, RIE is set, the RDRF flag generates an RDRF interrupt request.

3.8 Modes of Operation:

3.8.1 Synchronous Mode

In synchronous mode SCI acts as a high speed shift register. So SCI acts as serial to parallel or parallel to serial converter. The clock is made common to the transmitter and receiver. The data shift out through the TXD pin from the transmit shift register at the rising edge of the clock and the TDRE bit will help in loading the data registers and shift registers with data, to prevent the gap between data words while transmission. There are no parity checking or error flags, to detect the plausible errors.

3.8.2 Asynchronous mode

In the asynchronous communication, the device puts a start bit to the head of the transferred data(8 bits or 9 bits) and it puts a stop bit at the end of the data. Recognition in the data block is done by it. The start bit is logic low and the stop bit is the signal of the logic high. The signaling speed through the interface is controlled by the programmable baud rate generator. In the 11-bit asynchronous mode the bit8 will indicate the parity bit(even or odd) or address bit in the multidrop mode.

3.8.3 Multidrop Mode

The purpose of the multidrop mode is to allow multiple receivers and transmitters to be connected on a single line network. The receiver will check the bit8 of the received data in 11-bit format and if an address match is found by the status of the bit, the message is received otherwise remaining message is discarded and switch itself to wakeup mode.

3.8.4 Idle line wakeup mode

In the IIDLE wake up mode all messages are separated by idle character frames. The RWU bit is cleared by the idle condition(full frame of ones) on the input and the SCI wakes up. The initial part of every frame contains the address information and all the receivers in the multidrop mode whose address doesn't match will set its RWU bit and moves to a stand by state and stays there till the reception of an idle character frame and the intended receiver starts receiving the data. While sending the data the TE bit must be toggled to generate the preamble of ones.

3.8.5 Address bit wake up mode

Address bit mode doesn't require an Idle condition in between messages. If the address bit(8th or 9th bit) is logic high, the RWU bit is cleared and SCI wakes up. The same procedure for idle line mode is carried out when a address is detected otherwise device will be in standby state. This way the dead time for idle character in between the messages can be eliminated.

4 Synchronous Serial Interface

4.1 Introduction

The Synchronous Serial Interface (SSI) is a synchronous (clocked) module used for high-speed communications over short distances with other devices that have an SSI interface. SSI is also used to communicate with a high-speed, high-resolution A/D converter,D/A converters and various industry standard CODECs for instance. Because of signal degradation and critical timing issues, the SSI is not suitable for long-distance communications. The SSI is used for interfacing to Motorola SPI interfaces and other DSP processors and shift registers.

4.2 SCI Features

- Full Duplex port for serial Communication.
- Data transfer rate of 10Mbps for 40MHz oscillator clock.
- User definable characteristics
 - Number of bits per word
 - Communication protocol
 - Clock selection
 - Tx/Rx Synchronization
- Clocks
 - Continuous clock
 - Gated clock
- Synchronization
 - Bit length
 - Word Length
- Modes of Operation
 - Normal
 - Network
 - On-Demand
- Word Length Selection
 - 8, 12, 16, 24 bits
- Frame Sync Generator
- Clock Generator
- under run and over run error detection.
- Interrupt Driven Operation.

4.3 SSI Block Diagram



Figure 4-1 SSI Block Diagram

4.4 SSI Pin Description

The full operation of SSI requires six pins- STD,SRD,SCK,SC0,SC1,SC2, depending upon the operating modes, but dedicated operation can be made possible with the three pins(TXD,RXD and SCK). SC0, SC1 and SC2 are the serial control pins which are used based on the operating modes.

4.4.1 Serial Transmit Data Pin (STD)

STD is the transmit pin(output) which is used for transmitting the data loaded in the transmit shift register. The change in data changes on the positive edge of the SCK. When the transmission of one word completes, the STD clocks to a high impedance state. While not in use as serial interface the pin function as the general purpose PC8 input-output pin.

4.4.2 Serial Receive Data Pin (SRD)

SRD transfers the data serially into the receive shift register. Data is changed on the negative edge of the bit clock. The pin will act as general purpose PC1 input-output pin when TXD function is not used.

4.4.3 Serial Clock (SCK)

SCK is an input/output pin which acts as a serial clock for the SSI interface. The bidirectional pin is used by both transmitter and receiver in synchronous mode and transmitter alone in asynchronous mode.

4.4.4 Serial Control Pin (SC0)

The function of this pin is based on the mode of operation as synchronous or asynchronous. For asynchronous mode, this pin will be used for the receive clock I/O. For synchronous mode, this pin is used for serial flag I/O.

4.4.5 Serial Control Pin (SC1)

The function of this pin is determined solely on the selection of either synchronous or asynchronous mode. In asynchronous mode the pin is the receiver frame sync I/O and in synchronous mode with continuous clock, this pin is serial flag SC1 and operates like the previously described SC0.

4.4.6 Serial Control Pin (SC2)

SC2 is the frame sync pin. In synchronous mode it act as frame sync for transmitter and receiver and in synchronous mode it act as a frame sync for transmitter alone.

4.5 SSI Memory Map

The memory map for the full SSI module is listed in the table 4-1. When functioning as a receiver the data registers will be in read only mode and while transmitter is enabled, data registers will be write only.

| Memory Address | SCI Register | Bit Pattern |
|--------------------------|---------------------|--|
| X:\$FFEC (READ/WRITE) | CRA 24 - bit | The values in the bit positions 0-15 controls the SSI module which includes $PM(0-7)$, $DC(0-4)$, $WL(0-1)$, PSR respectively and the positions 16-23 are masked with zeros. |
| X:\$FFEE (READ ONLY) | SSISR 24 - bit | The values in the bit positions 0-7 sets the status of SSI module which includes IF(0-1), TFS, RFS, TUE, ROE, TDE, RDF respectively and the positions 8-23 are masked with zeros. |
| X:\$FFED (READ/WRITE) | CRB 24 - bit | The bit values in the locations 0-15 will control the SCI which includes OF0,OF1,SCD(0-2), SCKD, SHFD, FSL(0-1), SYN, GCK, MOD, TE, RE, TIE, RIE respectively and the positions 16-23 are masked with zeros. |
| X:\$FFEF (READ/WRITE) | SRX/STX 24 - bit | Depending up on the selected word length(8, 12, 16, 24 bits) the MSB bits of the 24 bits will be filled with the data and lower bits will be filled with zeros |

Table 4-1 Table Showing SSI Register Memory Map

4.6 SSI Registers

This section describes in detail about SSI registers in address order. Each description is included with a standard register diagram with the description of each bit and each fields function.

4.6.1 SSI Control Register A (CRA)

The CRA is a 24-bit read/write register that controls the serial interface operation. Each bit is described in the following section. The CRA is mapped to memory location X:\$FFEC. Only the lower two bytes holds the control bits and the higher byte is filled with zeros.

| 23 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----|---------|-----|-----|-----|-------|-------|-------|-------|-------|-------|-----|-----|-----|-----|-----|-----|
| 0* | PSR | WL1 | WL0 | DC4 | DC3 | DC2 | DC1 | DC0 | PM7 | PM6 | PM5 | PM4 | PM3 | PM2 | PM1 | PM0 |
| | I. | | | Fig | ure 4 | -2 SS | I Cor | ntrol | Regis | ter A | For | nat | Į. | | Į. | |

4.6.1.1 Prescale Modulus Select (PM7–PM0)

Prescale modulus is the clock divider bit. It selects the modulo from 1 to 256, for dividing the clock. All the bit are set to get the maximum divide ratio. The prescale counter modulo is chosen depending upon the interface to the port.

4.6.1.2 Frame Rate Divider Control (DC4–DC0)

Frame Rate divider is used in generating clocks frame sync generators. It put a control on the frame clock by setting the five bits with values range from 1 to 32. The scaling range varies from network mode(2 to 32) to normal mode(1 to 32) by a factor of 2.

4.6.1.3 Word Length Control (WL0, WL1)

Word length bit determines the size of each data word. The two control bits supports the SSI with 4 word lengths- 8, 12, 16, or 24 bits. These bits controls the frame sync.

4.6.1.4 Prescaler Range (PSR)

Compared to the PM bits which is programmable, PSR is fixed. PSR will divide the clock by a factor of 8 if it is set. This bit can helps in getting a much higher time base for interfacing to some other codecs.

4.6.2 SSI Control Register B (CRB)

The CRB is a 24-bit read/write register that controls the serial interface operation. Each bit is described in the following section. The CRB is mapped to memory location X:\$FFED. Only the lower two bytes holds the control bits to control the SSI operation and the higher byte is filled with zeros.

| 23 | | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----|------|-----|-----|----|----|-----|-----|-----|------|------|------|------|------|------|------|-----|-----|
| 0* | •••• | RIE | TIE | RE | TE | MOD | GCK | SYN | FSL1 | FSL0 | SHFD | SCKD | SCD2 | SCD1 | SCD0 | OF1 | OF0 |

Figure 4-3 SSI Control Register B Format

4.6.2.1 Output Flag 0 (OF0)

When the serial control direction zero bit is set, then data present in OF0 will be written to SC0 at the beginning of the frame in normal mode or at the beginning of the next time slot in network mode. At this time SC0 will be configured as an output pin.

4.6.2.2 Output Flag 1 (OF1)

When the serial control direction zero bit is set, then data present in OF1 will be written to SC1 at the beginning of the frame in normal mode or at the beginning of the next time slot in network mode. At this time SC1 will be configured as an output pin.

4.6.2.3 Serial Control Direction 0 (SCD0)

SCD0 controls the direction of the SC0 I/O line. When SCD0 is cleared, SC0 is an input; when SCD0 is set, SC0 is an output .

4.6.2.4 Serial Control Direction 1 (SCD1)

SCD1 controls the direction of the SC1 I/O line. When SCD1 is cleared, SC1 is an input; when SCD1 is set, SC1 is an output.

4.6.2.5 Serial Control Direction 2 (SCD2)

SCD2 controls the direction of the SC2 I/O line. When SCD2 is cleared, SC2 is an input; when SCD2 is set, SC2 is an output

4.6.2.6 Clock Source Direction (SCKD)

SCKD selects the clock for the transmitter and receiver while operating in th synchronous mode and for transmitter alone in asynchronous mode. The data is shifted in and out according to the clock selected. If SCKD bit is one then clock source will be the internal clock generated. But when SCKD is cleared clock will be chosen externally. In both cases selected clock will be connected to the SCK pin.

4.6.2.7 Shift Direction (SHFD)

SHFD determines the shifting of data into the shift register. When the SHFD bit is set data is shifted LSB first and data will be shifted into transmit shift register or out of receive shift register with MSB first when SHFD is cleared.

4.6.2.8 Frame Sync Length (FSL0 and FSL1)

FSL0 and FSL1 are the bits used by the frame sync generator to determine the type of frame to be generated. Four type of frame lengths will be made possible by selecting different or same frame sync length for both transmitter and receiver.

4.6.2.9 Sync/Async (SYN)

SYN is the main control bit which determines whether the transmitter or receiver should work in asynchronous mode or synchronous mode. When SYN bit is set then transmitter and receiver will operate in synchronous mode and they will share a common clock and frame sync. When SYN is cleared asynchronous mode will be selected and then transmitter and receiver will work with separate clocks.

4.6.2.10 Gated Clock Control (GCK)

GCK bit selects between continuous clock and gated clock. When GCK is set a gated clock will be selected and when cleared SSI will work on the continuous clock. Gated clock is the clock which runs only when data is shifted.

4.6.2.11 Mode Select (MOD)

SSI will operate mainly in normal mode or in network mode. While operating in network mode data word is transferred in every time period but in normal mode data is transferred per frame sync. MOD bit will be set if the device is opted to work in network mode and MOD bit will be cleared when the device works in normal mode.

4.6.2.12 Transmit Enable (TE)

When TE is set data is transferred from the transmit data register to transmit shift register. When a frame sync is detected only transmission will be enabled completely. While transmitting, if the TE

is cleared the transmission is completed and after that only transmitter gets disabled.

4.6.2.13 SSI Receive Enable (RE)

When RE bit is set data latched in the SRD pin which is transferred to the receive shift register will be transferred to the receive data register. If the RE bit is cleared while data is received, the remaining bits of the word will be shifted to the shift register before disabling the receiver.

4.6.2.14 SSI Transmit Interrupt Enable (TIE)

When TIE is set an interrupt request will be sent to the interrupt controller. At this stage transmit data register empty flag will also be set. But the interrupt will be cleared when data to be transmitted is being written to transmit data register.

4.6.2.15 SSI Receive Interrupt Enable (RIE)

When TIE is set an interrupt request will be sent to the interrupt controller. At this stage receive data register full flag will also be set. But the interrupt will be cleared when data received is written to receive data register. Reading the receive data register will clear RDF, thus clearing the pending interrupt.

4.6.3 SSI Status Register (SSISR)

SSISR is a 24 bit register whose lower byte holds the status bits and the rest of the bit positions are filled with zeros. This register hold status and serial input flags. SSISR is mapped to X:\$FFEE.

| 23 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----|----|----|----|-------|----|----|---|----|-----|-----|-----|-----|-----|-----|-----|-----|
| 0* | | | | ••••• | | | | 0* | RDF | TDE | ROE | TUE | RFS | TFS | IF1 | IF0 |

Figure 4-4 SSI Status Register Format

4.6.3.1 Serial Input Flag 0 (IF0)

When SCD0 is cleared and device operate in synchronous mode(SYN=1) then data latched in the SC0 pin is transferred into the IF0 flag. At this time frame sync is detected and at this time reception of first bit has already started. IF0 reads as a zero when it is not enabled.

4.6.3.2 Serial Input Flag 1 (IF1)

The SSI latches data present on the SC1 pin during reception of the first received bit after frame sync is detected. The IF1 flag is updated with the data, which is latched with the SC1 pin, when the receiver shift register is transferred into the receive data register. The IF1 bit is enabled only when SCD1 is cleared and SYN is set. IF1 is cleared when it is not enabled.

4.6.3.3 Transmit Frame Sync Flag (TFS)

In the transmit time period when a frame sync is detected, the TFS flag is et to indicate the frame sync and all the other time TFS flag will be left cleared.

4.6.3.4 Receive Frame Sync Flag (RFS)

While in the receive time period when a receive frame sync is detected, the RFS flag is set to indicate the frame sync and all the other time RFS flag will be cleared.

4.6.3.5 Transmitter Under run Error Flag (TUE)

When a transmit time slot comes and at that time if serial transmit shift register is empty then TUE is set. This indicates that when an under run error occurs the previous data(that is the data already present in the data register) will be retransmitted.

4.6.3.6 Receiver Overrun Error Flag (ROE)

Overrun occurs when the SSI receive shift register wants to transfer the received data to the SSI receive data register and the data register is already full. ROE flag will be set if and overrun occurs otherwise it will be loaded with zero.

4.6.3.7 Transmit Data Register Empty (TDE)

TDE is set when the data to be transmitted is shifted to the transmit shift register . But the bit will be cleared when the processor writes to the data register the next data word to be transmitted. IF TDE is set and also when TIE is set then a transmit interrupt request will be issued.

4.6.3.8 Receive Data Register Full (RDF)

RDF will be set when the data is fully transferred from the receive shift register to the receive data register. While the processor retrieves the data obtained through the interface which is loaded into the receive data registers, the RDF bit will be cleared.

4.6.4 SSI Receive Shift Register & SSI Receive Data Register (RX)

Data latched in the SRD pin is shifted into the receive shift register. Receive shift register is 24 bit and so it supports different word sizes selected. Data is shifted into in accordance with the receive frame sync.

When the received data is fully shifted to the shift register and it becomes full then the data is transferred to the receive data register. So the receive data register can occupy 24 bit of data. The unused bits will be declared as zeros. Receive data register is a read only and most significant portion is read first.



Figure 4-5 SSI Receive Data Registers^[1]

4.6.5 SSI Transmit Shift Register & SSI Transmit Data Register (TX)

Data to be transmitted through the STD pin will be shifted shifted into the transmit shift register. Transmit shift register is 24 bit and it supports different word sizes selected based on the WL bits in CRA. Data is shifted into in accordance with the transmit frame sync. The data to be transmitted occupies the most significant portion of the shift register. The unused portion of the register is filled with zeros.

The TX is a 24 bit write only register which holds the data to be transmitted. The data from TX is shifted to the transmit shift register ans transmitted serially. The data will be stored in 8,12,16 or 24 bit formats. The data to be transmitted occupies the most significant portion of the TX register. The least significant portion is ignored.



Figure 4-6 SSI Transmit Data Registers^[1]

4.6.6 Time Slot Register (TSR)

TSR is a register which is used only when no data is required to be transmitted in tha given time slot. So it remains in high impedance state without transmitting in that slot, but have all the features of the transmit register.

4.7 Functional Description

The SSI module allows a full duplex operation between the processor and the peripheral devices. The SSSI register flags are polled by the software and the operations can be interrupt driven. This section gives a functional description of the design including its various modes of operation.



4.7.1 Clock Generator Unit



Figure 4-8 Block Diagram of Clock Generator [1]

SSI clock generator unit is used to derive the receiver and transmitter clocks and also used for generating the frame sync signals. The crystal oscillator generator clock is divided by a factor of 2 and depending upon the prescaler input PSR, the clock is further divided by 1 or 8. The clock is again scaled by a factor determined by the bit sequence PM0-PM7, which can be programmed to scale the clock by a factor ranging from 1-256. The clock obtained is again scaled by a factor of two and then fed to transmit and receive section after considering the bit values for SCD0, SCKD and SYN. The design is illustrated in the figure. Multiplexers are used to select between 2 values and tristate buffer is used for input output control. SCK is the clock input used by both transmitter and receiver in synchronous mode. In asynchronous mode SC0 will act as receive clock and in synchronous mode it is used for serial flag I/O.

4.7.2 Frame Sync Generator



Figure 4-9 Block Diagram of SSI Frame Sync Generator [1]

The frame sync generator receives the transmit clock and receive clock from the clock generator and the word length for the data transmission/reception is determined. The programmable frame rate divider will generate the frame clocks. This allows a scaling range from 1 - 32. Before generating the frame sync, the frame sync length for the transmitter an receiver will be determined. In asynchronous mode the SC1 acts as receiver frame sync I/O and is used for serial flag in synchronous mode. In synchronous mode SC2 will act as the frame sync I/O for receiver and the transmitter and transmitter only in asynchronous mode. The selection between the synchronous and asynchronous mode is determined by the SYN bit and SCD1 and SCD2 will determine the serial I/O control directions for the corresponding pins.

4.8 SSI Operation

The transmit and receive sections of the SSI interface can be configured as synchronous or asynchronous. In synchronous mode transmitter and receiver share the common clock and various synchronization signals. But in asynchronous mode device requires separate clocks and Synchronization signals for both transmitter and receiver. When the SYN bit is cleared, then transmitter and receiver clocks and sync signals sources are independent and when set, source for the clocks and sync are the same. The data register is a 24 bit register and the data is shifted on the SCK clock and the positions shifted depends on the selected word length. A write to a register puts the data into the transmit shift register, when the transfer is complete. When transfer is completed received data can be moved to the data register from the receive data shift register. Appropriated flags are set for for checking receiver full, transmitter empty and also for the over rune and under

run errors. Interrupt service routines are activated with timely interrupt request considering the values of TIE and RIE.

4.8.1 Frame Sync Selection

The selection of the frame sync is determined by the bits FSL1 and FSL0, which will determine the frame sync length. Both transmitter and receiver can have bit length or word length frame sync signal format. For interfacing with different types of devices, while receiving and transmitting simultaneously ability to mix different frame lengths may be useful. FSL0 determines whether frame lengths are equal or different and FSL1 will determine the required frame length.



Figure 4-10 Sample format for Mixed frame length and same frame length for TX and Rx Sections[1]

4.8.2 Clock Selection- Continuous/Gated Clock

The GCK bit will decide on selecting between the Continuous clock and Gated clock. Mostly SSI interface is configured in continuous clock mode where clock is used for more fields than just transfer of data, like in Codecs. But in some cases like interfacing with SPI the a gated clock is used to allow clock transitions while transfer of data only. In continuous clock systems a separate frame synchronization signal is required, while in gated clock system clock itself will act as a Synchronization signal.



4.8.3 Operating Mode Selection- Normal/Network

SSI has three operating modes-normal, network and on-demand operating modes. The on-demand mode is not implemented as a part of this thesis as more importance is given to normal mode and then to network mode. The selection and operation of device in these modes will be determined by the bits in the control registers- MOD, GCK, SYN, FSL1, FSL0 and SHFD. But MOD bit will select between normal and network mode.

4.8.3.1 Normal Mode

In normal mode only one data word is transmitted or received per frame. But the data transfer is periodic. This mode is used when a single device in attached to the interface.



Figure 4-12 Data transmission format in Normal mode [1]

4.8.3.2 Network mode

The network mode is selected when the MOD bit is set to one. But unlike normal mode, 2 to 32 data words can be transferred per frame. Here also data transfer is periodic. Usually network mode is used in time division multiplexed systems where more than one Codecs or DSPs are connected in a network to the interface with multiple words per frame.



Figure 4-13 Data transmission format in Network mode [1]

5 System Design, Simulation and Results

Port C is modeled as though it depicts so called an IP core. The system consists of one main interface which descends hierarchically into different sub levels. The entire design was setup in the HDL Designer.

5.1 Port C Design

Port C is the main level of the design. It interfaces to the outside world through the nine dedicated I/O pins. In addition to the I/O pins there are some more signal which is connected to the core of the processor. This includes memory read and write control and interrupt control signals for the SSI and SCI blocks. For data transfer to and fro the main memory, the block is connected to the internal memory through the shared data bus and address bus.

| \leq | NO_INTRPT | | |
|------------|---|-------------|------------|
| \leq | INTRPT_SCI_RCX | PCO | \diamond |
| \leq | INTRPT_SCI_RCX_W_EXC | | L. |
| \leq | INTRPT_SCI_TIMER | PC2 | p |
| \leq | INTRPT_SCI_IDLE | | |
| < | INTRPT_SCI_TRX | PC1 | p |
| _ | TUMPER OCT DOX | DCA | h |
| \geq | INTRPI_SSI_KCA TAMPDOM COT DOY M DYC | PC4 | Μ |
| \geq | INTRPI_SSI_KCA_M_BAC | DC 2 | h |
| \geq | INTRPI_SSI_IKK INTRPI SSI TRY W RYC | PCS | Μ |
| | ININE I_DDI_IKA_N_BAC | PC5 | b |
| | DCDE(), Tabardana Daubo lib | | ř |
| | Port_C | PC6 | \diamond |
| | | PC7 | \diamond |
| < | MEM ADDRESS · (15.0) | PC8 | 6 |
| | ····· <u>·</u> ······························ | | ľ |
| \diamond | MEM DATA : (23:0) | | |
| Ť | | | |
| | | | |

Figure 5-1 Port C Interface



Figure 5-2 Port C Internal Structure

5.2 Port C I/O

The logic for a single pin is designed and copied for the rest of the eight pins as different concurrent processes. The details of the model is given in section 2.5 of this report.

5.2.1 Simulations of Port C I/O

The model was simulated in Modelsim. The behavior was verified with the functionality of the pin as input or output and also switch status between serial I/O pin and GPIO. Registers PCD, PCC and PCDDR were assigned with signal values for various conditions and the results were found to be satisfying.

5.3 SCI Design

The SCI was designed fully based on the functional description given in section 3.7. The input

block loads the SCI data registers periodically with transmit and receive data. The SCI descends into one more sub-level which includes the SCI interface block and the Baud Generator block. Figure shows the internal structure of the SCI. In addition to data register values, the SCI register, SCR, SCCR and SSR are loaded with test value to be simulated and passed to the SCI block along with the data register values. There are no separate data registers for transmitter and receiver as both sections are mapped to the same memory location. The data registers values are loaded to or from the Hi, Low or Mid Bytes of the 24 bit data registers, as the incoming/outgoing data is received/transmitted as bytes.





5.3.1 Baud Rate Generator

The design is exactly made as per the details given in section 3.7.2. One modification is that instead of selecting the asynchronous and synchronous clock from the baud rate generator itself based on the WDS flags, the selection is carried out in the SCI block as it reduces the risk of multiple clock domains and their control. So Baud Generator output is a single clock, which is the internal clock.

5.3.1.1 Simulation of Baud Rate Generator Block

The system is simulated with an input clock of 40MHz and also various sets of input values for the scaling bits from the SCCR register. The results are checked with the sample values given in Table 3-3 for verifying the functionality of the block. The detailed simulation results are included in the figure 5-5

| ۰ | osc_clk | | | | | | | |
|---------------|----------|-------------|------------|------|-------------|--|--|--|
| ٠ | reset | | | | | | | |
| | seer : | 00000000000 | 1000000000 | 0010 | | | | |
| ٠ | divby2_0 | | | | ່າກກາກກາກກາ | | | |
| ٠ | divbycd | | | | | | | |
| ٠ | divbyscp | | | | | | | |
| ٠ | divby2_1 | | | | | | | |
| \rightarrow | selk | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |

Figure 5-5 Simulation of Baud Rate Generator Block with a Clock Divider of 2

5.3.2 SCI Interface

The SCI interface receives the SCI register values, SCR, SSR, SCCR and SX from the input block and also the internal clock from the baud rate generator block. The selection between the synchronous and asynchronous clock is done inside the SCI as a separate process. The rest of the design follows the functional description as per figure 3-12.

5.3.2.1 Simulation of the SCI block

Simulations are made excluding the baud rate generator, which means SCI interface block separately with a fixed clock which gives maximum data rate of 10Mbps. This clock will be further scaled inside th SCI interface block for obtaining the asynchronous or synchronous clock. There is a separate clock which controls and coordinates the activities of the transmitter and receiver. The receiver works as a state machine for shifting the received values from the pin to the receive shift register. But the transmitter shifts out the values from the shift register directly. The operations for five different supported data formats(refer section 3.7.1) are carried out separately in receiver and transmitter parts. The details of the simulations results are given in figure

| 🔶 reset 👘 | | | | | | | | | | |
|--------------|--------------------|---------------------|---|----------|-----------|-----|----------|---------------------------------------|----------|----|
| 🔷 sci_hi | <u>(000000000</u> | <u> 10000001110</u> | 0011 | | | | | | ' | |
| 🔷 scr | <u>,0000000000</u> | <u>10000100000</u> | <u>,0000 </u> | | | | | | | |
| 🔶 solk | | | | | | | | | | mm |
| 🔶 sync_scl | | יתתתי | ഹസ് | ידרררון | יררריו | | | | יתתתי | תת |
| ♦sig_te | | [' | (| | · · · · · | (/ | [' | · · · · · · · · · · · · · · · · · · · | [' | |
| 🐟 txdata 👘 | (11100011 | <u> </u> | | <u> </u> | <u> </u> | | <u> </u> | | <u> </u> | |
| 🔶 tmptrege 👘 | | | | | | | | | | |
| √txd | י' | <u> </u> | | | μ | | | · · · · · · · · · · · · · · · · · · · | | |

Figure 5-6 SCI Transmitter in Synchronous 8-bit mode of transmission.

| ۲ | /sci_block/txd | | | | | | | | | | |
|---|------------------------------------|----------|----------|---------------|----------------|-------|---------|---------|---------|----------------|--------------|
| ė | /sci_block/scr | 00000000 | 00000001 | 00000000 | | | | | | | |
| ė | /sci_block/reset | | | | | | | | | | |
| | /sci_block/sclk | | | תתת | | | | | | | |
| ۲ | /sei_block/sync_sel | | | | | | | | | | |
| ۲ | <pre>/sci_block/rxd</pre> | | | | | | | | | | |
| ٠ | <pre>/sci_block/rx_shift_reg</pre> | υυυυυι | 100 | <u>100000</u> | <u>UU (110</u> | UUUUU | (111000 | JUU (11 | 1110000 | <u>)(01111</u> | <u>UUU</u> ; |
| ۲ | <pre>/sci_block/rx_shift_e</pre> | | | | | | | | | | |
| | /sci_block/d_out_va | \vdash | | | | | | | | | |

Figure 5-7 SCI Receiver in Synchronous 8-bit mode of reception.

| -🔷 /sci_block/sci_hi 👘 | (0000000000 | 0000001100 | 0111 | | | | | |
|------------------------|-------------------|------------|-------------------|------------------|------------------|-------------|--------|------|
| /sci_block/reset | 1 | | | | | | | |
| 🔷 /sci_block/sclk | | | | | | | | |
| ♦ /sci_block/sig_te | J | | | | | | | |
| ♦ /sci_block/async_sc | | | | | | | | |
| | (11000111 | | | | | | | |
| | <u>(1100)(111</u> | 0)(11111) | <u>1111 (1111</u> | <u>X1111 X</u> 1 | 111 1 111 | X 11 | 000111 | (111 |
| √ /sci_block /txd | | | | | | | | |

Figure 5-8 SCI Transmitter in asynchronous 11-bit Odd parity mode of transmission.

| | /sci_block/scr | 00000000 | 00000001000 | 100100 | | | | | | | = |
|---|-------------------------|------------|--------------------|--------|-----|---------|------|--------------------|-------------|------------|-----------|
| | 7 SCEDIOCK7 SCI | | | | | | | | | | t, |
| | /sci_block/rxd | μυυυι | | | | | | | 1 0 0 0 0 0 | | 4 U |
| • | /sci_block/sdk | | | | | | ՄՈՐՈ | | | | LΓL |
| ۲ | /sci_block/async_sc | | | | | | | | | | \square |
| • | /sci_block/cs | b6 end 3 | (b7 start 3 | | (Ь7 | end 3 | | (b8 start | 3 | <u>(</u> β | e |
| 4 | /sci_block/ns | h7 start 3 | <u>)h7 ent 3</u> | | (h8 | start 3 | | (h8 erd (| 8 | (st | ٥n. |
| - | /sci_block/rx_shift_reg | 110111UU |) <u>)1110111U</u> | | | | | <u>)</u> 11111C111 | | | |
| • | /sci_block/rx_shift_e | | | | | | | | | | |

Figure 5-9 SCI Receiver in asynchronous 11-bit Even parity mode of reception.

5.4 SSI Design

The design is made completely based on the functional level description given in section 4.7 of this report. The clock generator and frame sync generator forms a major part of the design. The receive and transmit controls are activated separately in different processes. The method of receiving and transmitting follows almost same as in SCI block as the results obtained maintains high level of accuracy. Further the selection of different modes of operation and clocking styles are done accordingly with the status of the corresponding flags. Transmission and reception for different word lengths are carried out separately.



Figure 5-10 SSI Structural View

5.4.1 Clock Generator

Based on the selection of synchronous mode or asynchronous mode the clock generator will decide on separate clock for transmitter and receiver or same clock for the both. It also controls the SSI pins SC0 and SCK. The design is based on the block diagram given in section 4.7.1 and the description is also given in detail in the same section.

5.4.1.1 Simulation of the Clock Generator Block

The system is fed with the 40MHz oscillator clock and the clock is further scaled depending on the clock divider bits from the SSI registers. The simulation is run for different scaling factors. The simulation results are given in the figure

| /clock_gen_blk/reset | F | | | | | | | |
|----------------------------|----|-----------|----------|--------|--|--|--|--|
| 🐟 /clock_gen_blk/cra | 00 | 100000000 | 00000000 | 000010 | | | | |
| 🔶 /clock_gen_blk/osc_clk - | | | | | | | | |
| /clock_gen_blk/divby2_0 | | | | | | | | |
| 🔶 /clock_gen_blk/divbypsr | F | | | | | | | |
| /clock_gen_blk/divbypm | | | | | | | | |
| /clock_gen_blk/divby2_1 | | | | | | | | |

Figure 5-11 Simulation Results of Clock generator with a prescale modulo of 2

5.4.2 Frame Sync Generator

Frame sync generator receives the output from the clock generator and generates the frame sync for both transmitter and receiver separately or same for both, depending upon the mode selected. The design details are given in the section 4.7.2.

5.4.2.1 Simulation of the Frame Sync Generator Block

The frame sync generator is attached with the previously verified clock generator and the outputs are compared with the standard patterns given in the figure 4-10. The modes of operation is also considered while verifying the output. The input values for registers are directly fed from the input data block which is used for testing purposes. The detailed simulation results can be analyzed from the figure

| 🔶 /frame_sync_gen_blk/osc_clk | | | | | | | |
|-------------------------------|-------------|-------------|------|-----|-----|-----|--|
| 🔶 /frame_sync_gen_blk/cra 👘 | (0000000000 | 0000110000 | 0000 | | | | |
| 🔶 /frame_sync_gen_blk/crb 👘 | (0000000000 | 00000000001 | 1000 | | | | |
| /frame_sync_gen_blk/divbydc | | | | hun | Inn | תתת | |
| 🔷 /frame_sync_gen_blk/divbywl | | | | | | | |

Figure 5-12 Simulation results of frame Sync Generator with word length(8 bits) Frame Sync for Rx and Tx

5.4.3 SSI Interface

The data transmission and reception is carried out almost same way as in SCI, but the over all control strategy and speed of operation varies. The total interface is designed based on the functional description given in section 4.7. The details regarding the design of the part for data buffering using shift registers and data registers for receiver and transmitter sections are done as outlined in sections 4.6.4 and 4.6.5. Moreover the clock selection, word length selection and selection of the mode of operation are carried out in this part.

5.4.3.1 Simulation of SSI block

The transmitter and receiver sections are verified using the clock and sync outputs from the clock generator and frame sync generator respectively. The output can be verified with the mode of operation selected which is given in the 4-12 and figure 4-13. The clocking strategy is also compared with the pattern in figure 4-11. The error checking is also carried out and the status of the error flags are also monitored. The detailed simulation results are given in the figure 5-13, 5-14 and 5-15.

| | /trame_sync_gen_blk/osc_clk | | | | | |
|---|----------------------------------|-----|-----|-------|----|----|
| | /frame_sync_gen_blk/divbydc | UUU | ллл | linin | nn | ЛЛ |
| ۲ | /frame_sync_gen_blk/divbywl | | | | | |
| ٠ | /frame_sync_gen_blk/serial_clock | | | | | |

Figure 5-13 Simulation Results Showing the Gated Clock when GCK bit is set to 1

| /ssi_blk/reset | | | | | | | | | |
|-----------------------|------------|---------------------|------------|------------|----------|------------|------------|-----|-------------------|
| /ssi_blk/crb | 0000000000 | 1000000001 | 1000 | | | | | | |
| /ssi_blk/serial_clock | | | | | | | | | |
| /ssi_blk/divbywl | ٦ | | | | | | | | |
| /ssi_blk/srd | ותתתה | וחחחח | | | | | | | |
| /ssi_blk/rx_shift_re | 11100000 | (111 | 10000 | | <u> </u> | 1111000 | | | <u>)(1111111(</u> |
| /ssi_blk/rx_shift_reg | 1100000000 | <u> 100000 (111</u> | 0000000000 | 0000000000 | D 1(1 | 1110000000 | 0000000000 | 000 |)1111100 |

Figure 5-14 Simulation run for the receiver part of the SSI with a word length of 8 bits and operation is in normal mode with gated clock

| 🐟 /ssi_blk/reset | | | | | |
|-----------------------|---|--------------|--|--|--|
| /ssi_blk/serial_clock | | | | | |
| 🔶 /ssi_blk/cra | (000000000000000 | 0010000000 | | | |
| 🚸 /ssi_blk/crb | (00000000000000000000000000000000000000 | 00000011000 | | | |
| 🔶 /ssi_blk/sig_te | | | | | |
| 🚸 /ssi_blk/sx | (0000000000000 | 000011100011 | | | |
| 🚸 /ssi_blk/divbywl 👘 | | | | | |
| 🔶 /ssi_blk/stxd | | | | | |

Figure 5-15 Simulation Results showing the transmission of serial data in normal mode of operation with a word length of 8 bits.

6 Results & Further Enhancements

6.1 Design Issues

The thesis was carried out as a part of a large project. So it took some idea to understand the basic functions and definitions used for the project, so as to maintain the overall coherence. Issues pop up while synthesizing the blocks, as multiple clock domains(synchronous and asynchronous clocks) in the process act as bottle necks, which some times gave wrong results during the later simulations.

Another issue came up with managing drivers for the signals. As the number of processes running concurrently increased, it was difficult to track the drivers of the signals to trouble shoot the system. The issue became more complicated when the entire Port C is simulated as a whole. The priority given to different signals was also of consideration. Some situations where problems where solved using higher priority to reset pins and then clock signals and the other control signals. The idea of synchronize the whole system with the master clock led to alleviate most of the synchronization issues. Reset was always kept at asynchronous reset, which was also some times made the task tedious while simulating with a large group of signals, with activating and deactivating resets in between.

More time was spend on understanding the system from the manual and data sheets. The information delivered in these was so stipulated the to derive a functional description of the system Several things has to be studied while arriving at a good design which are outside the bounds of the manual. So more time was spend on designing the system as a whole.

The HDL designer has made the task easy, but some software bugs which shut down the software in between or error while saving the design was a problem. The idea of executing the simulation runs with macro files run well. The Modelsim allowed to trouble shoot the system with break points. But some cases, the simulation results were not found to be satisfactory to synthesize the whole block

6.2 Deviations From the Motorola DSP56000 and Possible Enhancements[1]

The Port C was almost completely designed as per the product users manual. But the thesis focused on implementing the essential features of the Port C, which can be considered as a basic operational model. So the advanced features of the Port C were kept out of focus of the thesis part. But provisions were made to upgrade the design by including all the missed parts, by providing all the signals which should be used as necessary.

In the SCI block Interrupt handling was not implemented, as the service routing lies out side the scope of Port C and so the intended functionality was not tested. As a result the function of the STMINT bit was not added with the design of the baud rate generator. Also, as the operation of WOMS bit cannot be tested, its function was not included in this model. The operation of the device in multidrop mode was not implemented as the function the processor is going to perform doesn't require its support. But provisions were made so that its easy to add the operational mode in the future. Then the memory mapping of STXA was not verified. There was no procedure called to initialize the SCI than the global reset pin.

In the SSI section, the priority was given to the synchronous mode of operation, although the device

supports asynchronous mode pf transmission also. The functions of the SC0 SC1 and SC2 pins were implemented, but not verified for functionality as it was less important in the thesis part. The design was tested in the normal operational mode but the network mode of operation was not tested for functionality although implemented. The On demand mode is yet to be implemented.

Another possibility is to check the code with the base defenitions, for example scalable length, and verify the functionality for an interface coherent with a scalable DSP core.

The model is made as a synthesizable VHDL behavioral model, but the true functionality of the system can be verified when it is connected with the whole processor. The memory mapping of registers to the device main memory and the interrupt routines has to be verified before moving on to the synthesis of the blocks.

6.3 Results

The thesis was a success in terms of obtaining a complete behavioral model of the Port C of DSP56k But the sythesisability of the model was verified with no errors and few warnings. The simulations were run with test patterns made with the macro files. The results of the simulations shown much accurate results, which confirms the functionality of the design features. As the device behave as intended in the respective simulation runs, the synthesis of the model could be done without much compensation after fixing the left out parts and verification with actual processor as its dedicated part.

6.4 Further Work [4]

The ongoing reasearch in the Electronic System division focus on building a model of the Dsp with a scalable architecture. So another enhancement of this this work will be to make a sclable architecture of the PorC interface with respect to the data word length. The problems arinisn out of the memory read/write delays have to be supressed to get a continous transmission. The deviation from the actual DSP as mentioned in the section 6.2 has to be implemented. The problem arising of the multiple clock domains als to be verifies inorder to get a correct transition of signal. Some parts of code has to be re written to elimuinate these issues.
References

[1] DSP56002UM/AD REV1 Digital Signal Processor User's Manual, Austin, Motorola Inc 1993.

[2] "VHDL for Programmable Logic", Kevin Skahill ISBN 0-201-89573-0, Menlo Park , Addison-Wesley Publishing Company Inc. 1996.

[3] "A Behavioural Model of a DSP Processor with Scalable Architecture", Master Thesis by Patrik Thalin, Ulrik Lindblad – LiTH-ISY-EX 3289-2002

[4] "Development and validation of a Scalable DSP Core", Paper SSoCC 2004, Thomas Johansson, Patrik Thalin, Ulrik Lindblad and Lars Wanhammar

Appendix A

Glossary of Terms & Abbreviations

| A/D | - Analog to Digital |
|-------|---------------------------------|
| BPS | -Bits Per Second |
| CD | -Clock Divider |
| COD | -Clock Output Divider |
| CRB | -Control Register B |
| CRA | -Control Register A |
| CODEC | -Coding-Decoding |
| DSP | -Digital Signal Processor |
| D/A | -Digital to Analog Converter |
| DC | -Frame Rte Divider Control |
| FE | -Frame Error |
| GPIO | -General Purpose Input Output |
| GCK | -Gated Clock Control |
| I/O | -Input-Out |
| ILIE | -Idle Line Interrupt Enable |
| IDLE | -Idle Line Flag |
| IF | -Input Flag |
| Kbps | -Kilo bits per second |
| LSB | -Least Significant Bit |
| Mbps | -Megabits per second |
| MUX | -Multiplexer |
| MSB | -Most Significant Bit |
| MOD | -ModeSelect |
| OR | -Over Run Error |
| OF | -Output Flag |
| Osc | -Oscillator |
| PCC | -Port C Control Register |
| PCD | -Port C Data Register |
| PCDDR | -Port C Data Direction Register |
| PE | -Parity Error |
| PM | -Prescale Modulus |
| PSR | -Prescale Range |
| RXD | -Receive Data |
| RWU | -Receiver Wake Up |
| RE | -Receiver Enable |
| RDF | -receive Data Full |
| RIE | -Receiver Interrupt Enable |
| RX | -receive Data Register |
| RDRF | -Receive Data register Full |
| R8 | -Receive 8 th Bit |
| RCM | -Receive Clock Source bit |
| RD | -Read |
| RFS | -Receive Frame Svnc |
| SCI | -Serial Communication Interface |
| | |

| SSI | -Synchronous Serial Interface |
|-------|--|
| SoC | - System-on-Chip |
| SCLK | -SCI Serial Clock |
| SCK | -SSI Serial Clock |
| SC | -Synchronous Control |
| STD | -Serial Transmit data |
| SRD | -Serial Receive Data |
| SSFTD | -Serial Shift Data |
| SBK | -Send Break |
| STIR | -SCI Timer Interrupt Enable |
| SCKP | -Serial Clock Polarity |
| SCCR | -SCI Clock Control Register |
| SSR | -SCI Status Register |
| SCR | -SCI Control Register |
| STX | -SCI Transmit Data Register |
| STXA | -SCI Transmit Data Address Register |
| SRX | -SCI Receive Data Register |
| SCP | -Clock Prescaler |
| SPI | -Serial Peripheral Interface |
| SYN | -Synchronous |
| SCD | -Serial Control Direction |
| SHFD | -Shift Direction |
| SSISR | -SSI Status Register |
| TXD | -Transmit Data |
| TE | -Transmitter Enable |
| TIE | -Transmit Interrupt Enable |
| TMIE | -Timer Interrupt Enable |
| Tx | -Transmit |
| TDRE | -Transmit Data Register Empty |
| TRNE | -Transmitter Empty |
| TCM | -Transmit Clock Source |
| TFS | -Transmit Frame Sync |
| TUE | -Transmit Under-run Error |
| TDE | -Transmit Data register Empty(SSI) |
| TSR | -Time Slot Register |
| UART | -Universal Asynchronous Receiver Transmitter |
| WDS | -Word Format Select |
| WOMS | -Wored-Or Mode Select |
| WR | -Write |
| WL | -Word Length |





På svenska

Detta dokument hålls tillgängligt på Internet – eller dess framtida ersättare – under en längre tid från publiceringsdatum under förutsättning att inga extra-ordinära omständigheter uppstår.

Tillgång till dokumentet innebär tillstånd för var och en att läsa, ladda ner, skriva ut enstaka kopior för enskilt bruk och att använda det oförändrat för ickekommersiell forskning och för undervisning. Överföring av upphovsrätten vid en senare tidpunkt kan inte upphäva detta tillstånd. All annan användning av dokumentet kräver upphovsmannens medgivande. För att garantera äktheten, säkerheten och tillgängligheten finns det lösningar av teknisk och administrativ art.

Upphovsmannens ideella rätt innefattar rätt att bli nämnd som upphovsman i den omfattning som god sed kräver vid användning av dokumentet på ovan beskrivna sätt samt skydd mot att dokumentet ändras eller presenteras i sådan form eller i sådant sammanhang som är kränkande för upphovsmannens litterära eller konstnärliga anseende eller egenart.

För ytterligare information om Linköping University Electronic Press se förlagets hemsida <u>http://www.ep.liu.se/</u>

In English

The publishers will keep this document online on the Internet - or its possible replacement - for a considerable time from the date of publication barring exceptional circumstances.

The online availability of the document implies a permanent permission for anyone to read, to download, to print out single copies for your own use and to use it unchanged for any non-commercial research and educational purpose. Subsequent transfers of copyright cannot revoke this permission. All other uses of the document are conditional on the consent of the copyright owner. The publisher has taken technical and administrative measures to assure authenticity, security and accessibility.

According to intellectual property law the author has the right to be mentioned when his/her work is accessed as described above and to be protected against infringement.

For additional information about the Linköping University Electronic Press and its procedures for publication and for assurance of document integrity, please refer to its WWW home page: http://www.ep.liu.se/

© Swaroop Mattam