



**PowerPC
Set-Top Box Peripheral Chip
User's Manual**

Preliminary

Preliminary Version, October 1997

This edition of the *IBM PowerPC Set-Top Box Peripheral Chip User's Manual* applies to the initial version of the STBP chip unless otherwise indicated in new versions or technical newsletters.

The following paragraph does not apply to the United Kingdom or any country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS MANUAL "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions; therefore, this statement may not apply to you.

IBM does not warrant that the products in this publication, whether individually or as one or more groups, will meet your requirements or that the publication or the accompanying product descriptions are error-free.

This publication could contain technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or program(s) described in this publication at any time.

It is possible that this publication may contain references to, or information about, IBM products (machines and programs), programming, or services that are not announced in your country. Such references or information must not be construed to mean that IBM intends to announce such IBM products, programming, or services in your country. Any reference to an IBM licensed program in this publication is not intended to state or imply that you can use only IBM's licensed program. You can use any functionally equivalent program instead.

No part of this publication may be reproduced or distributed in any form or by any means, or stored in a data base or retrieval system, without the written permission of IBM.

Requests for copies of this publication and for technical information about IBM products should be made to your IBM Authorized Dealer or your IBM Marketing Representative.

Address comments about this publication to:

IBM Corporation
Department YM5A
P.O. Box 12195
Research Triangle Park, NC 27709

IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

©Copyright International Business Machines Corporation 1997. All rights reserved.

Printed in the United States of America.

4 3 2 1

Notice to U.S. Government Users—Documentation Related to Restricted Rights—Use, duplication, or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corporation.

Patents and Trademarks

IBM may have patents or pending patent applications covering the subject matter in this publication. The furnishing of this publication does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, 208 Harbor Drive, Stamford, CT 06904, United States of America.

The following terms are trademarks of IBM Corporation:

IBM
PowerPC
PowerPC Architecture
PowerPC Embedded Controllers
RISCWatch
RISCTrace
OS Open

The following terms are trademarks of other companies:

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

Windows is a trademark of Microsoft Corporation.

Other terms which are trademarks are the property of their respective owners.

Contents

Contents	v
Figures	ix
Tables	xi
About This Book	xiii
Who Should Use This Book	xiii
How to Use This Book	xiii
Related Publications	xiv
Document Conventions	xiv
Chapter 1. STBP Overview	1-1
1.1 STBP Features	1-1
1.1.2 403 Interface Macro	1-1
1.3 Interface Connections	1-3
1.4 PPC403GB Address and Data Connections	1-3
1.1.5 DMA Channels	1-4
1.1.6 Reset	1-4
1.7 Peripheral Function Addressing	1-4
1.1.8 Peripheral Function Programming	1-5
1.9 Clocks	1-5
1.10 Interrupt Programming	1-6
1.1.11 Interrupt Implementation	1-9
Chapter 2. Inter-Integrated Circuit Port	2-1
2.1 Interface to OPB	2-1
2.1.1 Timing	2-1
2.1.2 Reset	2-1
2.1.3 Timeout and Error Recovery	2-2
2.2 Programming Interface	2-2
2.2.1 Buffering	2-3
2.2.2 Master Data Buffer 0 and 1	2-3
2.2.3 Low Master Address	2-4
2.2.4 High Master Address	2-4
2.2.5 Control Register	2-4
2.2.6 Mode Control Register	2-6
2.2.7 Status Register	2-6
2.2.8 Extended Status Register	2-9
2.2.9 Low Slave Address	2-10

2.2.10 High Slave Address	2-11
2.2.11 Clock Divide Register	2-11
2.2.12 Interrupt Mask Register	2-12
2.2.13 Master Transfer Count Register	2-13
Chapter 3. Serial Communications Controller	3-1
3.1 Features	3-1
3.2 External Interfaces	3-2
3.3 Internal Interfaces	3-2
3.4 Serial Port and Modem (UARTs 1 and 2)	3-4
3.4.1 Handshaking	3-4
3.5 DMA Operation	3-5
3.6 UART Signals	3-5
3.6.1 Transmitter/Receiver Ready	3-8
3.7 UART Addressing	3-9
3.8 UART and Smart Card Speed Programming	3-9
3.8.1 Standard UART xtals	3-10
3.8.2 Formula	3-10
3.8.3 Required Rates	3-10
3.9 Registers	3-12
3.9.1 Line Control Register	3-12
3.9.2 Line Status Register	3-13
3.9.3 FIFO Control Register	3-15
3.9.4 Interrupt Identification Register	3-16
3.9.5 Interrupt Enable Register	3-17
3.9.6 Modem Control Register	3-18
3.9.7 Modem Status Register	3-19
3.9.8 Scratchpad Register	3-19
3.10 FIFO Operation	3-21
3.10.1 Receiver	3-21
3.10.2 Transmitter	3-21
3.10.3 Polled Mode	3-22
Chapter 4. Smart Card Interface	4-1
4.1 Features	4-1
4.2 Operation	4-3
4.3 Protocols	4-3
4.3.1 Transmission Error Detection Procedure (T0)	4-4
4.3.2 Reception Error Detection Procedure (T0)	4-4
4.4 Programming Interface	4-4
4.4.1 Clocking	4-4
4.5 Registers	4-5
4.5.1 Line Control Register	4-5
4.5.2 Line Status Register	4-6
4.5.3 FIFO Control Register	4-9

4.5.4 Interrupt Identification Register	4-9
4.5.5 Interrupt Enable Register	4-11
4.5.6 Modem Control Register	4-12
4.5.7 Modem Status Register	4-13
4.5.8 Scratchpad Register	4-13
Chapter 5. Serial Communications Protocol Interface	5-1
5.1 Programming Information	5-1
5.2 DMA Operation	5-3
5.3 SCP Interrupt Operation	5-3
5.3.1 SCP Loopback Operation	5-3
5.4 SCP Timing	5-3
Chapter 6. 1284 Parallel Port	6-1
6.1 Intercharacter Timer	6-2
6.2 IEEE 1284 Operating Modes	6-2
6.2.1 Compatibility Mode	6-2
6.2.2 Nibble Mode	6-3
6.2.3 Byte Mode	6-4
6.2.4 ECP Mode	6-5
6.3 Programming Information	6-7
6.3.1 Definition of IEEE 1284 Registers	6-8
6.3.2 IEEE 1284 Interface Signals	6-9
6.4 Programming Issues	6-12
6.5 Register Descriptions	6-14
6.5.1 RDR	6-14
6.5.2 RDF	6-14
6.5.3 TDR	6-15
6.5.4 TDF	6-15
6.5.5 DCR	6-16
6.5.6 DSR	6-18
6.5.7 IER	6-19
6.5.8 ISR	6-20
6.5.9 PCR	6-21
6.5.10 RCR	6-22
6.5.11 SCR	6-23
6.5.12 RTR	6-23
6.5.13 STR	6-23
6.5.14 AAT	6-24
6.5.15 CVR	6-24
6.5.16 EDR	6-25
6.6 External Interface	6-26
6.7 Standard Compatible Pseudo-Code	6-29
6.8 IEEE Std. 1284 Pseudo-Code	6-35
Chapter 7. General Purpose Input/Output Controller	7-1

7.1 GPIO Logic	7-1
7.2 Operation	7-1
7.2.1 Addressing	7-1
7.2.2 Functional Equations	7-2
Chapter 8. Pulse Width Modulation	8-1
8.1 Operation	8-1
8.1.1 IC Timers	8-2
8.1.2 Timebase Generator	8-2
8.2 Programming Information	8-2
8.2.1 Resetting:	8-3
Appendix A. Hardware Specifications A-1	
A.1 Pin Functional Descriptions	A-2
A.2 Package Thermal Specifications	A-14
A.3 Electrical Specifications	A-14
A.3.1 Absolute Maximum Ratings	A-14
A.3.2 Operating Conditions	A-14
A.3.3 Recommended Connections	A-15
A.3.4 DC Specifications	A-15
A.4 AC Specifications	A-15
Appendix B. Errata B-1	
Summary 1.	B-1
Summary 2.	B-1
Summary 3.	B-1
Summary 4.	B-2
Summary 5.	B-2
Summary 6.	B-2
Summary 7.	B-2
Solution 1.	B-2
Solution 2.	B-3
Solution 3.	B-4
Solution 4.	B-4
Solution5.	B-5
Solution 6.	B-5
Solution 7.	B-6
Appendix C. Acronyms and Abbreviations C-1	
Index	X-1

Figures

Figure 1-1.	OPB Bus Diagram for the STBP	1-3
Figure 1-2.	Sample Data Transfers Between PPC403 and STBP	1-6
Figure 3-1.	Serial Communications Controller — UARTs 1 and 2	3-3
Figure 4-1.	Smart Card Interface Controller	4-2
Figure 5-1.	SCP Timing (With Clock Inversion and revdat Set)	5-4
Figure 6-1.	1284 I/O Signals	6-1
Figure 6-2.	Compatibility Mode Data Transfer Cycle	6-3
Figure 6-3.	Nibble Mode Data Transfer Cycle	6-4
Figure 6-4.	Byte Mode Data Transfer Cycle	6-5
Figure 6-5.	Forward ECP Mode Data Transfer Cycle	6-6
Figure 6-6.	Reverse ECP Mode Data Transfer Cycle	6-7
Figure 6-7.	External Interface and Filters	6-28
Figure 6-8.	Buffer Manager (Standard Driver)	6-29
Figure 6-9.	Buffer Manager (IEEE Std. 1284 Parallel Driver)	6-36
Figure 8-1.	Output Waveform for PWM Unit	8-1
Figure A-1.	STBP Mechanical Drawing (Top View)	A-1
Figure A-2.	STBP Logic Symbol	A-2

Tables

Table 1-1.	Address Logic Truth Table	1-2
Table 1-2.	STBP Function Address	1-4
Table 1-3.	Interrupt Addressing	1-6
Table 1-4.	Functional Block Connection	1-7
Table 1-5.	Level/edge Bit Function	1-7
Table 1-6.	Mask Functions	1-8
Table 1-7.	Interrupt Level Functions	1-9
Table 2-1.	IIC Addressing	2-2
Table 2-2.	Control Register (Address 0x46)	2-5
Table 2-3.	Mode Control Register	2-6
Table 2-4.	Status Register	2-8
Table 2-5.	Decoding Pending Transfer/Error Bits	2-8
Table 2-6.	Bus Control State Bits	2-9
Table 2-7.	Extended Status Register	2-11
Table 2-8.	Clock Devide Register	2-12
Table 2-9.	Interrupt Mask Register	2-12
Table 2-10.	Master Transfer Count Register (Bits 5:7)	2-13
Table 2-11.	Master Transfer Count Register (0x4E)	2-13
Table 3-1.	UART IO Signals	3-5
Table 3-2.	UART Busmaode Register	3-9
Table 3-3.	Clock Selection	3-9
Table 3-4.	Divide Values for Input Clocks - Cases 1, 2, and 3	3-11
Table 3-5.	Divide Values for Input Clocks - Case 4	3-11
Table 3-6.	LCR	3-12
Table 3-7.	LSR	3-13
Table 3-8.	FCR	3-15
Table 3-9.	IIR	3-16
Table 3-10.	IER	3-17
Table 3-11.	MCR	3-18
Table 3-12.	MSR	3-19
Table 3-13.	Register Summary	3-20
Table 4-1.	Smart Card Busmode Register	4-4
Table 4-2.	Smartcard Clock Selection	4-5

Table 4-3.	LCR	4-5
Table 4-4.	LSR	4-6
Table 4-5.	FCR	4-9
Table 4-6.	IIR	4-10
Table 4-7.	IER	4-11
Table 4-8.	MCR	4-12
Table 4-9.	MSR	4-13
Table 4-10.	Register Summary	4-14
Table 5-1.	SCP Addressing	5-1
Table 5-2.	SPCOM Register Definitions	5-2
Table 5-3.	SPMODE Register Definitions	5-2
Table 5-4.	BUSMODE Register Definitions	5-2
Table 5-5.	SPRATE s	5-2
Table 6-1.	IEEE 1284 Address Bit 4	6-8
Table 6-2.	Bus Mode Register Definitions	6-8
Table 6-3.	IEEE 1284 Interface Signals	6-10
Table 6-4.	IEEE 1284 Signal Transition Event Summary	6-12
Table 6-5.	Negotiation Mode Events using EDR	6-25
Table 7-1	GPIO Address	7-1
Table 8-1.	PWM and Timers Register Addressing	8-2
Table 8-2.	PWM Control Registers	8-3
Table 8-3.	IC Interrupt Status Register	8-3
Table A-1.	Signal Descriptions	A-3
Table A-2	Pin Number-Signal Name	A-13
Table A-3.	STBP Maximum Ratings	A-14
Table A-4.	STBP DC Characteristics	A-15
Table A-5.	STBP I/O Capacitance	A-15

About This Book

This book provides the device overview, programming information, and detailed descriptions of the multiple peripheral processing interfaces that are integrated into the IBM Set-Top Box Peripheral chip (STBP).

Who Should Use This Book

This book is for hardware, software, and application developers who need to understand the STBP. The audience should understand embedded system design, real-time operating systems, and set-top box application development.

How to Use This Book

This book describes the STBP including various features, external and internal interfaces, and specific device functions.

This book contains the following chapters:

- Chapter Chapter 1., “STBP Overview” describes the device functions and peripheral interfaces, clocks, and interrupts.
- Chapter Chapter 2., “Inter-Integrated Circuit Port” contains a functional description of the inter-integrated circuit (IIC) and programming information, including register data.
- Chapter Chapter 3., “Serial Communications Controller” contains functional descriptions of universal asynchronous receiver/transmitter (UART) 1 and 2.
- Chapter Chapter 4., “Smart Card Interface” contains a functional description of the smart card interface (SCI).
- Chapter Chapter 5., “Serial Communications Protocol Interface” describes operation of the serial communications protocol (SCP) interface.
- Chapter Chapter 6., “1284 Parallel Port” contains a functional description of the high-speed IEEE 1284 parallel interface.
- Chapter Chapter 7., “General Purpose Input/Output Controller” contains a functional description of the general purpose input/output controller (GPIO).
- Chapter Chapter 8., “Pulse Width Modulation” describes operation of the pulse width modulation (PWM) macro.
- Appendix Appendix A., “Hardware Specifications,” contains physical data, pinout data and operating specifications.
- Appendix Appendix B., “Errata,” contains descriptions and work-arounds for errors.
- Appendix Appendix C., “Acronyms and Abbreviations,”

Related Publications

The following publications provide related information:

PowerPC Architecture: A Specification for a New Family of RISC Processors

403GB Core User's Manual

403GC Core User's Manual

Set-Top Box Reference Design Kit User's Manual

IEEE 1284 - 1994, Standard Signaling Method for a Bidirectional Parallel Interface for Personal Computer (ANSI) (referred to as IEEE 1284)

ISO/IES DIS 7816-3, Information Technology - Identification Cards - Integrated Circuit(s) Cards With Contacts - Part 3 Electronic Signals and Transmission Protocols

Document Conventions

The following is a list of notational conventions frequently used in this book.

Active_Low An overbar indicates an active-low signal.

0x1f Hexadecimal numbers

0b1001 Binary numbers

Chapter 1. STBP Overview

The STBP provides an integrated peripheral set for set-top applications in a single package. The STBP attaches to a PPC403 controller as a memory-mapped I/O (MMIO) peripheral device and uses two DMA channels on the PPC403GB to improve throughput to or from higher bandwidth devices.

The STBP operates with a 27MHz system clock input provided by the system. Additional clock inputs can provide the serial devices with base clock inputs for more accurate baud rate generation at higher operating frequencies.

The STBP has 101 control or data signals in a 144-pin plastic flat pack package. Eleven pins are reserved for manufacturing test. This device is fabricated in a 3.3V technology with TTL compatible I/Os and 5V-tolerant I/Os. External pull-up resistors should be used on open drain outputs.

1.1 STBP Features

- Eight peripheral units on a single chip:
 - General purpose input/output controller (GPIO)
 - Serial communications controller (SCC)
 - UART1 (Serial port)
 - UART2 (Modem interface)
 - Smart card interface (SCI)
 - Inter-integrated circuit interface (IIC)
 - Serial communications protocol interface (SCP)
 - IEEE 1284 interface
 - Pulse width modulation interface
- Simple interconnect to the IBM PPC403GB controller
- High DMA transfer rates using two-channel PPC403GB internal DMA controller
- PPC403GB SysClk signal generated by the STBP internal clock unit

1.1.2 403 Interface Macro

The interface macro interfaces with a single SRAM chip select signal from the PPC403GB along with the control signals of two DMA Channels and one interrupt line from the PPC403GB. The 403GB interface is responsible for converting these external signals into the subset of the OPB bus signals used on this chip. The bank register corresponding to this chip select is set for one wait state with the Ready signal used to provide additional wait

states when necessary. The Ready signal shall be implemented as a three-state driver which is enabled by $\overline{\text{CS_ASIC}}$ being active (low) and ADDR_x being high.

This PPC403GB interface macro generates a Chip Select Out ($\overline{\text{CSOUT}}$) signal based on the decode of the chip select signal being active and a single address line being inactive.

This PPC403GB interface macro latches all incoming and outgoing PowerPC interface signals. This reduces the risk of system timing problems.

To avoid glitches on the output chip select signals, the corresponding bank register should be programmed with $\overline{\text{CSon}}=1$. In this way, the signal, ADDR_x, will be stable when $\overline{\text{CS_ASIC}}$ is active. It is also necessary to program Hold=1 in this bank register.

Table 1-1. Address Logic Truth Table

CS_ASIC	ADDR_x	Int. CS	CS_Out	Function
0	0	1	0	External chip selected
0	1	0	1	PPC403GB selected
1	0	1	1	Nothing selected
1	1	1	1	Nothing selected

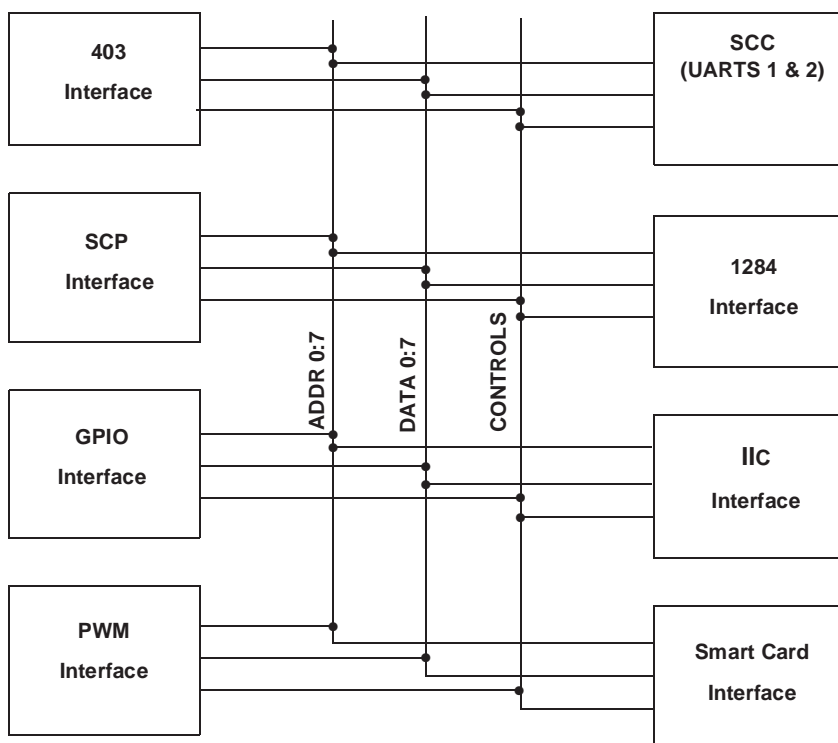


Figure 1-1. OPB Bus Diagram for the STBP

1.3 Interface Connections

The internal on-chip peripheral bus (OPB) connects the functional blocks on this chip. The OPB and attached peripherals are accessible to the PPC403GB controller.

1.4 PPC403GB Address and Data Connections

The STBP has an 8-bit address and data interface to the PPC403GB. The PPC403GB communicates with STBP over this 1-byte subset of its 32-bit interface. Data bit 0 of STBP is connected to Data bit 0 of the PPC403GB. Address bit 0 of STBP is connected to address bit 0 of the PPC403GB.

1.1.5 DMA Channels

The PPC403GB DMA channel assignment specifies DMA channels 0 and 1. Other systems in which STBP is installed can assign DMA channels 2 and 3 to STBP. This is assigned by card wiring connection of DmaReq and DmaAck signals on the STBP.

The DMA support logic is also implemented in this block. Both DMA channels transmit or receive data in one direction at a time. Both DMA channels can be used to provide full bidirectional data flow to one port. If DMA is not enabled (set by a bus-mode register in each component), MMIO is used for communications between the STBP and the PPC403GB. Software allocates the available DMA channels as needed. Unpredictable results can occur if two devices are programmed to use a single DMA channel simultaneously.

1.1.6 Reset

The 403GB Interface block receives the \overline{RST} (reset) signal from a system reset and distributes this signal to the rest of the chip as necessary. A negative active level on the \overline{RST} pin resets the STBP. This signal must be held low at least 8 clock cycles long. The system clock must be running during reset.

1.7 Peripheral Function Addressing

The STBP interfaces on the OPB and associated interrupts are under program control. Address bits 0:3 address the macros of the STBP chip. Bit 0 is msb. Address bits 4:7 are used internally by each macro.

Table 1-2. STBP Function Address

Address 0:3	Block
0000	GPIO
0001	UART 1
0010	UART 2
0011	Smart card interface
0100	IIC interface
0101	SCP
0110	IEEE 1284 interface
0111	Interrupt
1000	PWM and timers

1.1.8 Peripheral Function Programming

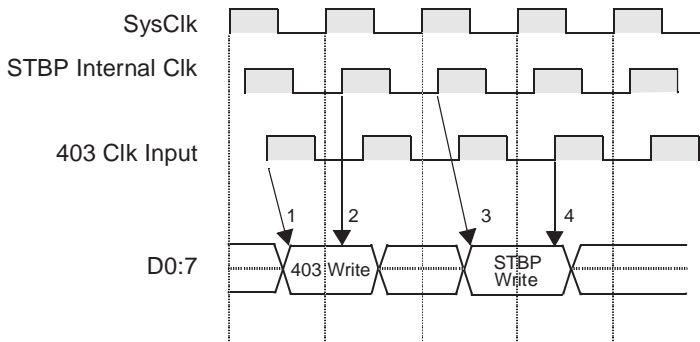
Refer to the individual chapters for operating modes, register descriptions and other programming information.

- Chapter Chapter 2., “Inter-Integrated Circuit Port,”
- Chapter Chapter 3. , “Serial Communications Controller,”
- Chapter Chapter 4., “Smart Card Interface,”
- Chapter Chapter 5., “Serial Communications Protocol Interface,”
- Chapter Chapter 6., “1284 Parallel Port,”
- Chapter Chapter 7., “General Purpose Input/Output Controller,”
- Chapter Chapter 8., “Pulse Width Modulation,”

1.9 Clocks

To achieve proper latching of data between STBP and the PPC403GB, the SysClkOut output of the STBP chip drives the PPC403GB SysClk input. This allows the PPC403GB clocks to be synchronized with the STBP clocks. The PPC403GB uses an internal phase-locked loop (PLL), which causes its internal clocks, at the latch inputs, to rise at the same time as the clock input. STBP does not have a PLL, so there is a significant internal delay between the rise of its clock input and the rise of a clock at an internal latch.

The SysClkOut signal from STBP is designed so that the rising edge of the clock occurs slightly before the change of data on the STBP outputs. This will assure correct latching of STBP data by the PPC403GB. Additionally, the SysClkOut signal occurs slightly after the internal clock signals, therefore assuring correct latching of PPC403GB data into STBP.



Notes:

1. PPC403 output data valid
2. STBP input data valid required
3. STBP output data valid
4. PPC403 input data valid required

Figure 1-2. Sample Data Transfers Between PPC403 and STBP

1.10 Interrupt Programming

A common interrupt signal ($\overline{403_INT}$) is generated by the STBP, which combines interrupts from all STBP functional blocks. Each interrupt source on this chip is wired to an interrupt summary register similar to the EXISR register on the PPC403GB. They are then gated via an enable register similar to the EXIER register on the PPC403GB, logically ORed together, and then presented to the PPC403GB as a single interrupt signal.

After the PPC403GB receives the interrupt, it must interrogate the interrupt status registers to determine the source of the interrupt.

Table 1-3. Interrupt Addressing

ADDRESS 4:7	Function
0000	Interrupt Status bits 0:7, connected to GPIO bits 0: 7
0001	Interrupt Status bits 8:15, connected to GPIO bits 8:15
0010	Interrupt Status bits 16:23, connected to GPIO bits 16:23
0011	Interrupt Status bits 24:31, connected to functional blocks.
0100	Level/edge bits 0:7, connected to GPIO bits 0:7
0101	Level/edge bits 8:15, connected to GPIO bits 8:7
0110	Level/edge bits 16:23, connected to GPIO bits 16:2

Table 1-3. Interrupt Addressing

ADDRESS 4:7	Function
0111	Level/edge bits 24:31, connected to functional blocks
1000	Interrupt Mask bits 0:7, connected to GPIO bits 0:7.
1001	Interrupt Mask bits 8:15, connected to GPIO bits 8:15.
1010	Interrupt Mask bits 16:23, connected to GPIO bits 16:23.
1011	Interrupt Mask bits 24:31, connected to functional blocks.
1100	Interrupt Polarity bits 0:7, connected to GPIO bits 0:7
1101	Interrupt Polarity bits 8:15, connected to GPIO bits 8:15
1110	Interrupt Polarity bits 16:23, connected to GPIO bits 16:23
1111	Interrupt Polarity bits 24:31, connected to functional blocks

Table 1-4. Functional Block Connection

Interrupt Status Bits, Mask Bits, Interrupt Polarity Bits, and Level Edge Bits 24:31	Function
24	Intercharacter timer expired (program as level sensitive)
25	UART 1
26	UART 2
27	Smart card Interface
28	IIC Interface
29	SCP interface
30	1284 Parallel Port
31	Reserved

Table 1-5. Level/edge Bit Function

Bit	Function:
0	Interrupt is level-sensitive.

Table 1-5. Level/edge Bit Function

Bit	Function:
1	Interrupt is edge-sensitive

Table 1-6. Mask Functions

Bit	
0	Allow interrupt status register bit to be ORed into the Int_403 signal STBP to PowerPC.
1	Keeps a 1 in the interrupt status register from causing an interrupt to PowerPC.

Equations:

$\text{Int_403} = ((\text{Status bit 0 and not Mask bit 0})$
 $\text{or } (\text{Status bit 1 and not Mask bit 1})$

$\text{or } (\text{Status bit 31 and not Mask bit 31}))$

This register will reset in the 1 or Masked state.

Table 1-7. Interrupt Level Functions

bit	Functions
0	Interrupt is positive level sensitive, or rising edge sensitive, depending on setting of level/edge
1	Interrupt is negative level sensitive, or falling edge sensitive, depending on setting of level/edge.

1.1.11 Interrupt Implementation

Individual interrupt Status bits are located in the respective functional blocks. The interrupt status register gathers all of the individual interrupt lines into a single location for software to read.

The interrupt status bits are reset by writing ones to the source register bit locations in the respective functional blocks.

In the case where the interrupt status bit has been programmed to be edge-sensitive, resetting the interrupt requires writing a one to the interrupt status register bit itself, not the original source of the interrupt.

In the case where a GPIO input is programmed to be an edge-sensitive interrupt source, the GPIO input signal may not be asynchronous. The GPIO signal is not allowed to change state within 5ns before or after the rising edge of the system clock.

The interrupt mask register resets to the mask state to prevent unwanted interrupts after reset. The other interrupt registers must be reconfigured by software after a reset.

Chapter 2. Inter-Integrated Circuit Port

The IIC macro supports the following features:

- 100 and 400 KHz operation
- 8-bit data transfers
- 7-bit and 10-bit address decode/generation
- Slave transmitter and receiver
- Master transmitter and receiver
- Multiple bus masters

Both the serial clock (SCL) and the serial data (SDA) lines are bidirectional to support multiple bus masters and fast and slow devices on the same bus.

2.1 Interface to OPB

The IIC macro supports 10-bit external addressing, and it provides a 16- or 8-bit interface internally by way of the OPB. Selection of the data width is controlled by signals on the OPB. The 8-bit interface connects to high-order data bits of the OPB. To connect to a 32-bit internal bus, connect IIC to the high-order half of the bits on the OPB. IIC can be dynamically switched between 10-bit and 7-bit addressing on the IIC interface by writing the appropriate bit in one of the control registers. All registers are readable and writable for test and diagnostic purposes, even if they are read- or write-only in normal operation.

2.1.1 Timing

After a byte is written by the OPB interface into the master or slave buffer, four clock cycles must pass before the OPB can read the data from the buffer.

2.1.2 Reset

IIC_RST must be active for at least eight clock periods for the reset to initialize IIC. Since this is a clocked reset, IIC_SYS_CLK must oscillate while reset is active.

IIC_RST must be inactive for at least eight clock cycles before IIC is programmed over the OPB interface.

After a reset, IIC enters the unknown IIC bus state. This state is exited when activity is seen on the bus, or when the 6.5 ms timeout expires.

2.1.3 Timeout and Error Recovery

The reset timeout is the only timeout function implemented in IIC. There is no timeout during data transfers. If such error recovery is needed, it must be implemented in software. The IIC halt operation and reset can implement any desired error recovery.

2.2 Programming Interface

The addresses in Table 2-1 refer to byte addresses. Double-byte (halfword) transfers are accomplished by setting OPB_hwXfer to 1 and accessing an even numbered address. Thus, address 0x40 with halfword size will access Master Data Buffer 0 and 1, address 0x42 will access Slave Data Buffer 0 and 1, address 0x44 will access Lo Master Address and Hi Master Address, and so on.

Address bits 0:3 address the macros of the STBP chip. Bit 0 is msb. Address bits 4:7 are used internally by each macro.

Table 2-1. IIC Addressing

Facility Name	Address	Type	Affected by Reset	Size (bits)
	0x40	R/W	Yes- cleared	8
Master Data Buffer 1	0x41	R/W	Yes- cleared	8
Slave Data Buffer 0	0x42	R/W	Yes- cleared	8
Slave Data Buffer 1	0x43	R/W	Yes- cleared	8
"Low Master Address," on p. 2-4	0x44	R/W	No	8
"High Master Address," on p. 2-4	0x45	R/W	No	8
"Control Register," on p. 2-4	0x46	R/W	Yes- cleared	8
"Mode Control Register," on p. 2-6	0x47	R/W	Yes- cleared	8
"Status Register," on p. 2-6	0x48	R/W	Yes- cleared	8
"Extended Status Register," on p. 2-9	0x49	R/W	Yes- cleared	8
"Low Slave Address," on p. 2-10	0x4A	R/W	No	8
"High Slave Address," on p. 2-11	0x4B	R/W	No	8
"Clock Divide Register," on p. 2-11	0x4C	R/W	Yes- cleared	8
"Interrupt Mask Register," on p. 2-12	0x4D	R/W	Yes- cleared	8
"Master Transfer Count Register," on p. 2-13	0x4E	R/W	Yes- cleared	8

Table 2-1. IIC Addressing

Facility Name	Address	Type	Affected by Reset	Size (bits)
Reserved	0x4F			

2.2.1 Buffering

The master and slave buffers are 4 by 1 byte-wide FIFOs. Thus the last byte written into a buffer is not the first one read out. Care must be exercised when these buffers are written or read.

For example, consider the case where a user program sends a byte of data over the IIC interface into the Buffer. If the program then checks that the byte is actually inside the buffer, it might be assumed that the program could then perform a read of the buffer. If this is done, the program is actually removing the byte that it wrote into the buffer. If the program then executes, invalid data is sent out over the IIC interface because the buffer no longer contains the expected byte.

2.2.2 Master Data Buffer 0 and 1

Master data buffer 0 and master data buffer 1 (and slave data buffer 0 and slave data buffer 1) can be written in one access if the address is set to 0x40 (or 0x42) and IIC_SIZE_EQ_16 is a logic one. Double-byte write or read operations using slave data buffer 0 and slave data buffer 1 is identical to master data buffer 0 and master data buffer 1.

The master data buffer is a FIFO, four bytes deep by one byte wide. To facilitate 16-bit transfers, a second address space, MDB 1 is provided. When MDB is written with a byte, the byte is put into the FIFO and the hardware pushes the byte down to the lowest unoccupied location. The byte advances one FIFO stage per clock. If the FIFO is empty, a total of four clocks are needed, one to write the byte and three for the byte to pop to the first stage of the FIFO. This timing is important when the program reads the FIFO immediately after data was written. When a master transfer is requested, IIC handles this latency.

When the buffer is written with two bytes, byte 1 is put into the first stage and byte 0 is put into the second stage. In this case, bits 8:15, which is byte 1, are sent out to the IIC first and bits 0:7, which is byte 0, are sent out second. Bytes written when the FIFO is full are discarded.

The MDB receives data when the requested master transfer is a read. The first byte received goes into the first stage, the second byte received goes into the second stage and so on. As data is read from the FIFO, any data in stage 2 is moved by the hardware into stage 1, data in stage 3 moves into 2 and so on. If 16 bits are read, IIC_SIZE_EQ_16 is a logic one. In this case, stages 3 and 4 pop up to stages 1 and 2. When an empty FIFO is read, data that was last stored in stage 1 (stages 1 and 2 for a 16-bit operation) is returned to the output data bus.

Care must be taken not to start a requested master operation when there is still data in the master data buffer. If, for example, a master read transfer was requested with stale data in the buffer, then this data would be presented to the program as data read by the requested operation.

The SDB0 and SDB 1 work in exactly the same way as the MDB0 and MDB1, except that they store data sent or received in a slave transfer on the IIC bus. This allows the overlapping of slave and master transfer operations.

The buffers are flushed (set to empty) whenever \overline{RST} is active or when one of the flush bits is set to 1 in the mode control register.

2.2.3 Low Master Address

When 7-bit addressing is used only the low master address register (LMAR) needs to be written. The high master address register (HMAR) is not used. For 7-bit addressing, bits 0:6 form the address that is transmitted on the IIC interface; bit 7 is don't care (unused). When 10-bit addressing is used, bits 0:7 form the second byte address transmitted on the IIC interface.

2.2.4 High Master Address

HMAR is not used for 7-bit addressing. For 10-bit addressing, this register must be programmed to 0b1110yyx, where:

- yy is the high-order 2 bits of the 10-bit address
- X is don't care

Thus, in 10-bit address mode bits 0:6 of both LMAR and HMAR form the address that is transmitted on the IIC interface; bit 0 is don't care in both registers.

2.2.5 Control Register

The control register **must** be the last register programmed. After the pending transfer bit is set, IIC attempts to perform the requested transfer using the values in the other registers. Note that not all of the registers in this block need to be programmed every time a transfer is performed. Only the pending transfer bit is cleared to a zero when the requested transfer has been completed. The remaining bits are left unaffected.

Count bits are defined:

Count Bits 2:3	Bytes Transferred
00	1
01	2
10	3

Count Bits 2:3	Bytes Transferred
11	4

To determine the state of the IIC bus, software can read the status register and extended status register.

Table 2-2. Control Register (Address 0x46)

Bit	Register '0x46' - Description
0	Halt Master Transfer. When set to 1 the Halt function is performed. If IIC was performing a requested master transfer it will issue the STOP signal at the earliest possible point in time on the IIC bus. If no transfer was being done then no action is taken on the bus. In either case, an interrupt is activated if so enabled in the Control registers. To halt a transfer the Pending Transfer bit must also be set to 1.
1	10/7 Bit Addressing. When set to 1 10-bit addressing is activated on the IIC bus. Slave 10-bit transfers are not controlled by this bit. Refer to the Hi Slave Address Register. On a requested master transfer 10-bit addressing will be used to form the address on the IIC bus. A zero will cause 7 bit addressing to be used.
2	Count bit 1 (MSB)
3	Count bit 0 (LSB). The count bits indicate how many bytes are in the requested master transfer. When all bytes have been successfully transferred the requested transfer is considered complete. The precise action taken is dependent upon the programming in this register and in the Mode Control register.
4	Repeated Start. When set to 1 the requested master transfer will be started using a repeated START function on the IIC bus
5	Chain. When set to 1, the requested master transfer is one in a sequence of transfers. Completion of the requested transfer only indicates that this piece of the transfer is complete. The next action taken on the IIC bus is under the control of the software. When set to 0, the requested master transfer is either the only transfer to be performed at this time or it is the last transfer in a sequence of transfers. In either case completion of the requested transfer causes a STOP condition to be sent on the IIC bus.
6	R/ \overline{W} . When set to 1, the requested master transfer is a read. When set to 0, a write is performed.
7	Pending Transfer. When set to 1, a master transfer is started if the IIC bus is free. {The type of requested transfer is specified by the other bits in this register.} If the IIC bus is busy, IIC waits for the bus to become free and then starts the requested transfer. This bit will be cleared when the requested transfer finishes. When the transfer is complete, an interrupt, if enabled in the Mode Control register, is activated.

2.2.6 Mode Control Register

In typical applications, the mode control register (MCR) is programmed only in the program initialization routine. Applications that include complex error handling can reprogram this register.

The clock divide register must be programmed before the MCR is programmed. The LMAR and HMAR should be programmed before the MCR is programmed. IIC does not implement any time-out functions for communications on the IIC bus. Any such function, if needed, must be implemented using the reset signal or halt function for this macro function.

Table 2-3. Mode Control Register

Bit	Register '0x47' - Description
0	Flush Slave Data Buffer. When set to 1 the Slave Data Buffer will be set to empty. This bit will be cleared once the Buffer has been emptied.
1	Flush Master Data Buffer. When set to 1 the Master Data Buffer will be set to empty. This bit will be cleared once the Buffer has been emptied.
2	Enable General Call. When set to 1 IIC will respond to a general call on the IIC bus. Note that Enable Slave Mode overrides this bit. If Enable slave mode is zero a general call will be ignored.
3	Fast/Standard Mode. When set to 1 IIC will transfer data at a rate of 400kHz (Fast Mode). When set to 0 IIC will transfer data at a rate of 100kHz (Standard Mode).
4	Enable Slave Mode. When set to 1 slave transfers are enabled. When set to 0 IIC will not respond to any slave transfers in the IIC bus. The slave address registers should be programmed before this bit is set to 1.
5	Enable Interrupt. When set to 1 allows an interrupt to be generated when an event occurs and the event is enabled in the Interrupt Mask register. Refer to the Interrupt Mask register for a list of the events that can generate an interrupt. When set to 0 the interrupt is disabled. This bit overrides any settings in the Interrupt Mask register.
6	Exit Unknown IIC Bus State. When set to 1 the Bus Control State machine will exit the unknown bus state if IIC is currently in this state. If IIC is not in the unknown bus state then setting this bit to a one has no effect.
7	reserved

2.2.7 Status Register

The status register (SR) should be the first register read by an interrupt service or error handling routine. The SR summarizes the state of IIC and the previously requested master transfer.

The SR can be read in a polling loop if software is not going to use the interrupt signal from IIC.

Table 2-4. Status Register

Bit	Register '0x48' - Description
0	Slave Data Buffer has Data. When set to 1, the Slave Data Buffer has data in it. When set to 0, the buffer is empty. This is a read-only bit.
1	Slave Data Buffer Full. When set to 1, the Slave Data Buffer is full. When set to 0, the Slave Data Buffer is not full. This is a read-only bit.
2	Master Data Buffer has Data. When set to 1, the Master Data Buffer has data in it. When set to 0, the buffer is empty. This is a read-only bit.
3	Master Data Buffer Full. When set to 1, the Master Data Buffer is full. When set to 0, the Master Data Buffer is not full. This is a read-only bit.).
4	Halted or Stopped. Set to 1 when a request to halt operations on the IIC bus by the program was completed. This bit is also set whenever the current requested master transfer has ended with the issuance of a STOP signal on the IIC bus.
5	Error. This bit is set when bits 1 and 0 are set in the Extended Status register. It is provided as a summary of the success of failure of the requested master transfer. This is a read-only bit.
6	IRQ Active. When set to 1 the IIC_IRQ signal is active. The Extended Status register can be read to see why the interrupt was set. The interrupt is cleared by writing a one to this bit. If interrupts are disabled, bit 2 in the Mode Control register is zero, then this bit, IRQ Active, will not be set.
7	Pending Transfer. This bit is also readable here to help the program, via one read operation, determine the state of a requested transfer. This is a read-only bit.

The pending transfer and Error bits are the main indicators of the success or failure of the requested transfer. They can be decoded as follows:

Table 2-5. Decoding Pending Transfer/Error Bits

Pending Transfer Bit - Error Bit	Definition
0 0	Requested transfer is complete, all data has been sent to or received from the slave.
0 1	Requested transfer is complete, but not all data was transferred.
1 0	Requested transfer is incomplete, no errors detected.

Table 2-5. Decoding Pending Transfer/Error Bits

Pending Transfer Bit - Error Bit	Definition
1 1	Requested transfer is incomplete, an error was detected.

The program should wait until the pending transfer bit is set to 0 before any action is taken. If the error required IIC to send a STOP, bit 3 is also set. The program must clear this register before requesting another transfer except for bits 6 and 7. Because these bits involve a slave transfer, they can remain set to 1. Bits 0:3, 5, and 7 are read only.

2.2.8 Extended Status Register

Bits 5:7 are cleared when the pending transfer bit is set in the control register. Bits 6:4 are read-only, and change as the bus control state machine sequences. Bit 7 is cleared when the current interrupt is cleared and the pending interrupt request (IRQ) becomes the current IRQ.

The bus control state bits are a direct encode of the main IIC state machine. Their defined values are:

Table 2-6. Bus Control State Bits

Bits	Definition
000	unused. If this value is read, there is a major malfunction in this macro function.
001	Slave selected state. This block enters this state after it detects and decodes a slave operation on the IIC bus.
010	Slave transfer state. This block enters this state after it has detected, but not yet decoded, a slave operation on the IIC bus.
011	Master transfer state. IIC enters this state after it was programmed with a requested master transfer and it has started a master operation on the IIC bus.
100	Free IIC bus state. IIC is in this state whenever the bus is free and no transfer is pending.
101	Busy IIC bus state. IIC is in this state whenever the bus is busy.
110	Unknown IIC bus state. The IIC enters this state after its RESET signal is activated
111	unused. If this value is read, there is a major malfunction in this macro function.

2.2.9 Low Slave Address

When 7-bit addressing is used the low slave address register is written with the low slave address. The high slave address register is written with all zeros. For 7-bit addressing, bits 0:6 decode the address transmitted on the IIC interface; bit 7 is don't care. When 10-bit addressing is used, bits 0:7 decode the second address byte transmitted on the IIC interface.

Table 2-7. Extended Status Register

Bit	Register '0x49' - Description
0	IRQ Pending. Set when IRQ is still active and another interrupting condition was generated. When IRQ Active is set to 0 the pending IRQ will cause IRQ Active to be set to 1 and IRQ Pending will be cleared to a zero. The pending interrupt can be cleared by writing a one to this bit. If interrupts are disabled, bit 2 in the Mode Control register is zero, then this bit, IRQ Pending, will not be set.
1	Bus Control State bit 2 (MSB). Read only.
2	Bus Control State bit 1. Read only.
3	Bus Control State bit 0 (LSB). Read only.
4	Lost Bus Arbitration. Set to 1 when an attempt a Bus Arbitration was NOT successful. IIC will retry the requested master transfer until arbitration is won.
5	Won Bus Arbitration. Set to 1 when an attempt a Bus Arbitration was successful.
6	Incomplete Transfer. When set to 1, some bytes in the requested master transfer were not sent or received from the slave. The Master Transfer Count register can be read to see how many bytes were transferred.
7	Transfer Aborted. When set to 1, a requested master transfer was aborted by a NOT Acknowledge after the transfer of the address byte.

2.2.10 High Slave Address

This register is not used for 7-bit addressing. To disable 10-bit addressing, this register must be set to zero. To enable 10-bit addressing, this register must be set to 0b11110yyx where:

- yy = The hi-order 2 bits of the 10-bit address
- x = Don't care

Thus, in 10-bit address mode bits 0:6 decode the first address byte transmitted on the IIC interface; bit 7 is don't care. Note that bit 1, 10/7bit Addressing, in the control register is not used to control the decode of 10-bit slave addresses.

2.2.11 Clock Divide Register

This register divides the IIC_SYS_CLK signal to form the base clock that is used for interfacing to the IIC bus. This register *must* be programmed before the MCR. The IIC state machines will not be activated until this register is programmed. Locking the state machines in this manner prevents this block from misinterpreting activity on the IIC bus. The value programmed into this register is governed by the equation:

$$\text{programmed value} = n - 1$$

Because the base clock is used as the basis for all setup and hold timings on the IIC bus, the correct value *must* be used. If the wrong value is used, IIC will violate most if not all of the IIC timing specifications. Note that it is possible to run at slower data rates by simply choosing larger values of *n*.

Table 2-8 lists the values of *n* and the possible frequencies of the IIC_SYS_CLK signal. The value of *n* shown is the same regardless of whether standard or fast mode is used.

Table 2-8. Clock Devide Register

<i>n</i>	Frequency Range (MHz)	<i>n</i>	Frequency Range (MHz)
0x2	20	0x9	80-90
0x3	20-30	0xA	90-100
0x4	30-40	0xB	100-110
0x5	40-50	0xC	110-120
0x6	50-60	0xD	120-130
0x7	60-70	0xE	130-140
0x8	70-80	0xF	140-150

2.2.12 Interrupt Mask Register

This register should be set before interrupts are enabled in the mode control register.

Table 2-9. Interrupt Mask Register

Bit	Register '0x4D' - DescriptionI
0	reserved.
1	reserved.
2	Enable IRQ on Halt Executed.
3	Enable IRQ on Incomplete Transfer.
4	Enable IRQ on Transfer Aborted.
5	Enable IRQ on Data Needed for a Slave Read. The interrupt is activated upon receipt of a STOP on the IIC bus or upon Slave Buffer Full.
6	Enable IRQ on Data Received in a Slave Write. The interrupt is activated upon receipt of a STOP on the IIC bus or upon Slave Buffer Empty.
7	Enable IRQ on Requested Master Transfer Complete.

2.2.13 Master Transfer Count Register

Count bits(5:7) are defined as follows:

Table 2-10. Master Transfer Count Register (Bits 5:7)

Count Bits 5:7	Bytes Transferred
000	0
001	1
010	2
011	3
100	4
101	Reserved
110	Reserved
111	Reserved

Table 2-11. Master Transfer Count Register (0x4E)

Bit	Register '0x4E' - Description
0	reserved.
1	reserved.
2	reserved.
3	reserved.
4	reserved.
5	Count bit 2 (msb)
6	Count bit 1
7	Count bit 0 (lsb)

Chapter 3. Serial Communications Controller

The Serial Communications Controller (SCC) is a universal asynchronous receiver/transmitter (UART) with FIFOs. The UART performs serial-to-parallel conversion on data characters received from a peripheral device or a modem, and parallel-to-serial conversion on data characters received from the PPC403GB.

The PPC403GB can read the UART status. Status information includes:

- The type and condition of the transfer operations being performed by the UART
- Any error conditions, such as parity, overrun, framing and break interrupt.

The SCC is in character mode on power-up. To relieve the PPC403GB of excessive software overhead, the SCC unit can be put into FIFO mode. In FIFO mode, internal FIFOs are activated, storing 16 bytes (plus three bits per byte of error data in the RCVR FIFO) to be stored in both receive and transmit modes.

The SCC has separate operational (SYS_CLK) and serial data (XTAL) clocks. The SCC includes a programmable baud rate generator that can divide the timing reference clock XTAL input by a divisor of 1 to $(2 \times 16-1)$, and it produces a 16X clock for driving the internal transmitter logic. This 16X clock can also drive the receiver logic.

The UART has complete modem-control capability and a processor interrupt system. Interrupts can be programmed to the user's requirements, minimizing the computing required to handle the communications link.

3.1 Features

The SCC has the following features:

- Can run existing NS16550A and NS16450 software
- After reset, all registers are identical to the NS16450 register set
- 16-byte transmitter FIFO
- 16-byte receiver FIFO
- Precise synchronization between PPC403 and serial data id not required
- Independently controlled transmit, receive, line status and data set ready
- Programmable baud rate generator
 - Serial clock frequency up to $0.5 \times$ system clock frequency
 - Baud rate = serial Clock/[16 x (value in divisor latch)]
- Modem control functions

- Fully programmable serial interface characteristics:
 - Character length (5-8 bits)
 - Number of stop bits (1-2 bits)
 - Even, odd, or no parity
 - False start bit detection
- Complete status reporting capability
- Line break generation and detection
- Internal diagnostic capability (loopback mode)
- Full prioritized interrupt system controls interrupt output
- System clock frequency up to 50MHz

3.2 External Interfaces

The external UART interface implements the full RS-232 protocol including, ring indicator (RI), data carrier detect (DCD), data set ready (DSR), data terminal ready (DTR), request to send (RTS), clear to send (CTS), and serial clock (SC). Figure 3-1 describes the SCC interfaces.

3.3 Internal Interfaces

The SCC has the following internal interfaces:

- Two general purpose control outputs are provided as OUT1 and OUT2.
- An interrupt output is provided for communication to the PPC403.
- A manufacturing test interface is provided to insure high fault coverage and testability.
- Reset is provided to initialize the soft core.
- An 8-bit STBP interface is provided for register control and communication to the transmit and receive FIFOs.
- A Class II clock and power management interface is provided to reduce power dissipation.

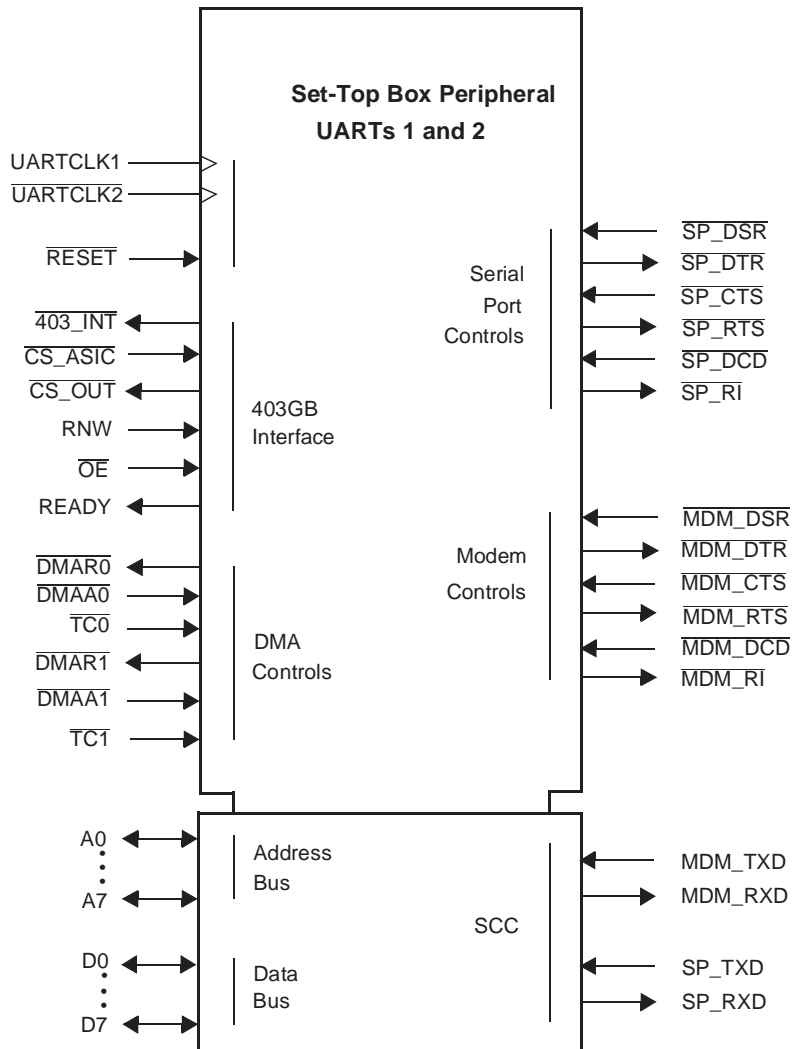


Figure 3-1. Serial Communications Controller — UARTs 1 and 2

3.4 Serial Port and Modem (UARTs 1 and 2)

UART1 and UART2 provide a standard full-duplex serial port that supports baud ranges from 1200 to 1152000 baud. The UARTs support 7-bit or 8-bit transfers with even, odd, or no parity, and 1 or 2 stop bits. Either channel of each serial port can be connected independently to the two DMA channels provided on the STBP or accessed using memory-mapped I/O (MMIO).

Both UARTs have a 16-bit receive and transmit FIFO. TC/EOT is bidirectional for 403, but only used as an input for STBP. TC is the function from 403 --> STBP. When TC asserted by 403, turn off RTS in UART.

The SelAddr input uniquely addresses both UARTs in hardware. SelAddr is an input to the UART inside the STBP chip. It is tied to 0 for the instantiation of UART 1 and tied to a 1 for the instantiation of UART 2.

There are two UART clock inputs to STBP. The clock inputs are multiplexed into the internally-derived UART clock under microcode control. The internally-derived uartclk is Sysclk/2 or 13.5MHz for STBP. Table 3-3 and Table 3-4 provide divide values for various input clock values.

The following outputs, which are described in the RS-232C standard, are provided at TTL compatible signaling levels: RXD, TXD, RTS, CTS, DTR and DSR.

3.4.1 Handshaking

The following handshaking is provided by these UARTS:

- Transmission ceases when the CTS signal is deactivated.
- The UART deactivates the RTS signal when DMA reaches TC or when the receiver buffer is full. RTS can also be driven via software.
- An interrupt occurs on change of state of the CTS and DSR signals.
- An intercharacter timer (Section 8.1.1, "IC Timers") provides the following characteristics:
 - Programmable reloadable down counter, 8 bit
 - Clocked by 9.7 ms ticks
 - Register programmed by host to COUNT
 - Counter loaded with COUNT whenever idle condition (high level) is detected on the RXD line and count started. Counter counts down as long as RXD is idle.
 - Counter stopped whenever line RXD goes active (low). Counter is reloaded with COUNT when next idle condition occurs.
 - If the counter reaches zero, count shall stop and interrupt sent to host.
 - 9.7 ms to 2.47 s, in steps of 9.7 ms

The iintercharacter timer is implemented in the Pwm/Timers block. The Count signal feeds the timer block from the UART block. The counter counts when the count signal is active, and resets when the count signal becomes inactive. The timer block causes an interrupt when a timer expires.

- The programmable time-out in SEND mode of 100ms to 25.6s shall be implemented in software.

3.5 DMA Operation

In DMA mode, read data available interrupts are still presented to the IntCntl logic. Software must mask such interrupts using the interrupt control register in the UART, not the interrupt mask register in the IntCntl logic.

Because read/write cycle durations on the SCC require four cycles, the PPC403GB should be programmed for 7 wait states from DmaAck active-to-inactive. An inactive DmaReq follows an inactive DmaAck by two cycles, requiring the PPC403GB be set for 4 cycles hold time.

Depending on the UART FIFO control register setting, the UART can generate a DmaReq when the transmit FIFO is not full.

3.6 UART Signals

Table 3-1 shows both internal and external signals

Table 3-1. UART IO Signals

Signal	Type	Description
LTADR(0:2)	I	Address (0:2) The PPC403 uses this address to select the SCC internal registers. It must be valid while the I/O write or I/O read strobe is active. Bit2 is the MSB. The Register addresses are provided in the next section describing the registers.
XDI(0:7)	I	Data Input (0:7) The PPC403 data to be written to the SCC internal registers. It must be valid while the I/O write strobe is active. Bit7 is the MSB.
CS_N	I	Chip Select The active low chip select signal. It must be valid while the I/O write or read strobe is in an active state.
IOW_N	I	I/O Write Strobe The I/O write strobe for internal registers.

Table 3-1. UART IO Signals

Signal	Type	Description
IOR_N	I	I/O Read Strobe The I/O read strobe for internal registers.
MRESET_N	I	Master Reset When low, clears the control logic of the SCC and all registers (except the Receiver Buffer, Transmitter Holding and Divisor Latches.)
XTAL	I	Xtal In The basic clock for the baud rate generation. This frequency is related to the baud rate as follows: Baud Rate x 16 x value in Divisor Latches = XTAL frequency This clock is sampled by SYS_CLK which must be > 2X this frequency.
SIN	I	Serial Input Serial data input from the communications link (peripheral device, modem, or data set).
DCD_N	I	Data Carrier Detect When low, indicates that the data carrier has been detected by the modem or data set. The DCD_N signal is a modem status input whose condition can be tested by the PPC403 reading bit 7 (DCD) of the modem status register (MSR). Bit 7 is the complement of the DCD_N signal. Bit 3 (DDCD) of the MSR indicates whether the DCD_N input has changed state since the previous reading of the MSR. DCD_N has no effect on the receiver.
SP_DSR	I	Data Set Ready When low, indicates that the modem or data set is ready to communicate with the UART. The SP_DSR signal is a modem status input whose condition can be tested by the PPC403 reading MSR[DSR], which is the complement of the SP_DSR signal. MSR[DDSR] indicates whether the SP_DSR input has changed state since the previous reading of the MSR.
SP_CTS	I	Clear To Send When low, indicates that the modem or data set is ready to exchange data. SP_CTS signal is a modem status input whose condition can be tested by the PPC403 reading MSR[CTS], which is the complement of the SP_CTS signal. Bit 0 (DCTS) of the MSR[DCTS] indicates whether the SP_CTS input has changed state since the previous reading of the MSR. SP_CTS has no effect on the Transmitter. When SP_CTS is deactivated, the UART does not transmit data.

Table 3-1. UART IO Signals

Signal	Type	Description
$\overline{\text{RI_N}}$	I	Ring Indicator When low, indicates that a telephone ringing signal has been received by the modem or data set. The $\overline{\text{RI_N}}$ signal is a modem status input whose condition can be tested by the PPC403 reading MSR[RI], which is the complement of the RI_N signal. MSR[TERI] indicates whether the $\overline{\text{RI_N}}$ input signal has changed from a low to a high state since the previous reading of the MSR. Whenever the RI bit of the MSR changes from a high to a low state, an interrupt is generated if the modem status interrupt is enabled.
XDO(0:7)	O	Data Output(0:7) Data output to the PPC403.
OUT1_N	O	Output 1 This is a user-designated output that can be set to an active low by programming bit2(OUT1) of the MCR to a high level. A master reset operation sets this signal to its inactive (high) state. Loop mode operation holds this signal in its inactive state.
OUT2_N	O	Output 2 This is a user-designated output that can be set to an active low by programming bit3(OUT2) of the MCR to a high level. A master reset operation sets this signal to its inactive (high) state. Loop mode operation holds this signal in its inactive state.
INTR	O	Interrupt Goes high whenever any one of the following interrupt types have an active high condition and is enabled via the IER: receiver error flag; received data available: timeout (in FIFO mode only); transmitter holding register empty; and modem status. The INTR signal is reset low upon the appropriate interrupt service or a master reset operation.
NT_BAUDOUT	O	Divided XTAL clock. This is the 16X baud clock generated by dividing the XTAL frequency by the decimal divisor.
SOUT	O	Serial Output Composite serial data output to the communications link (peripheral, modem or data set). The SOUT signal is set to 1 upon a master reset operation.

Table 3-1. UART IO Signals

Signal	Type	Description
SP_DTR	O	Data Terminal Ready When low, informs the modem or data set that the UART is ready to communicate. The $\overline{\text{SP_DTR}}$ output signal can be set to an active low by programming the MCR[DTR] to a high level. A master reset operation sets this signal to its inactive (high) state. Loop mode operation holds this signal in its inactive state.
SP_RTS	O	- Request to Send When low, informs the modem or data set that the UART is ready to exchange data. The $\overline{\text{SP_RTS}}$ output signal can be set to an active low by programming the MCR[RTS]. A master reset operation sets this signal to its inactive (high) state. Loop mode operation holds this signal in its inactive state.
TXRDY_N, RXRDY_N	O	- Transmitter/Receiver Ready See Section 3.6.1.

3.6.1 Transmitter/Receiver Ready

Transmitter and Receiver DMA signalling is available through two pins. When operating in the FIFO mode, one of two types of DMA signalling per pin can be selected via FCR3. When operating as in the NS16450 Mode, only DMA mode 0 is allowed. Mode 0 supports single transfer DMA where a transfer is made between PPC403 bus cycles. Mode 1 supports multiple transfer DMA where multiple transfers are made continuously until the RCVR FIFO has been emptied or the XMIT FIFO has been filled.

RXRDY Mode 0

When in the NS16450 Mode(FCR0=0) or in the FIFO Mode(FCR0=1,FCR3=0) and there is at least 1 character in the RCVR FIFO or RCVR holding register, the RXRDY_N pin will be low active. Once it is activated the RXRDY_N pin will go inactive when there are no more characters in the FIFO or holding register.

RXRDY Mode 1

In the FIFO Mode (FCR0=1) when the FCR3=1 and the trigger level or the timeout has been reached, the RXRDY_N pin will go low active. Once it is activated it will go inactive when there are no more characters in the FIFO or holding register.

TXRDY Mode 0

When in the NS16450 Mode(FCR0=0) or in the FIFO Mode (FCR0=1,FCR3=0) and there are no characters in the XMIT FIFO or XMIT holding register, the TXRDY_N pin will be low active. Once it is activated the TXRDY_N pin will go inactive after the first character is loaded into the XMIT FIFO or holding register.

TXRDY Mode 1

In the FIFO Mode(FCR0=1) when the FCR3=1 and there is at least one unfilled position in the XMIT FIFO, it will go low active. This pin will become inactive when the XMIT FIFO is completely full.

3.7 UART Addressing

Address bits 0:3 address the functional blocks of the STBP chip. Bit 0 is msb. Address bits 4:7 are used internally by each functional block.

Table 3-2. UART Busmaode Register

Name	Mode	Address 4:7	Definition
busmode	R/W	1000	bit 0: uart clk sel bit 0
			bit 1: uart clk sel bit 1
			bit 2: rcvdma1
			bit 3: rcvdma0
			bit 4: rcvmmio
			bit 5: txdma1
			bit 6: txdma0
			bit 7: txmmio

3.8 UART and Smart Card Speed Programming

There is a UART Clk input to STBP. This input is multiplexed into the internally-derived uart clock under microcode control. The internally-derived uartclk is Sysclk/2 (13.5 MHz for STBP).

Table 3-3. Clock Selection

Sel 0,1	Clock Used
00	Sysclk/2
01	UART Clk1
10	UART Clk2
11	Undefined

Tables of divide values for various input clock values are provided in Table 3-4 and Table 3-5. Fractional divide values are not supported.

It is recommended to use the internal sysclk/2 for rates up to 115,200 and an external uartclk of 4.9152mhz for the rate of 307,200 for the smart card. If the tolerance on the baud rates using sysclk/2 is not sufficient, then a second option is to provide 2 uartclk external inputs for two separate oscillators.

3.8.1 Standard UART xtals

1.8432MHz

3.0720MHz

3.8.2 Formula

Baud = $\text{clk}/(16)(\text{divide reg})$

3.8.3 Required Rates

1200

2400

4800

9600

19200

38400

57600

115200

307200

Table 3-4. Divide Values for Input Clocks - Cases 1, 2, and 3

Case 1 With 1.8432 MHz Clock and Formula: Divide reg = clk/(16)(baud).		Case 2 With 3.0720 MHz Clock and Formula: Divide reg = clk/(16)(baud).		Case 3 With 4.915 MHz Clock and Formula: Divide reg = clk/(16)(baud).	
Baud Rate	Divide	Baud Rate	Divide	Baud Rate	Divide
1200	96	1200	160	1200	256
2400	48	2400	80	2400	128
4800	24	4800	40	4800	64
9600	12	9600	20	9600	32
19200	6	19200	10	19200	16
38400	3	38400	5	38400	8
57600	2	57600	3.33	57600	5.33
115200	1	115200	1.67	115200	2.67
307200	0.375	307200	0.625	307200	1

Table 3-5. Divide Values for Input Clocks - Case 4

Case 4 With 13.5 MHz Clock and Formula: Divide reg = clk/(16)(baud).			
Desired Baud	Actual Baud	Tolerance	Divide
1200	1200	0%	703
2400	2397	0.1%	352
4800	4794	0.1%	176
9600	9588	0.1%	88
19200	19176	0.1%	44
38400	38352	0.1%	22
57600	56250	2.3%	15
115200	120536	4.6%	7
307200	281250	8.4%	3

3.9 Registers

The PPC403GB accesses the UART registers. These registers control the UART operations including transmitting and receiving data. The bit number is implemented with bit 0 as msb, matching the PPC403GB numbering convention.

3.9.1 Line Control Register

The Line Control Register (LCR) specifies the format of asynchronous data communications and sets the Divisor Latch Access Bit (DLAB).

The PPC403GB can read the contents of the LCR. The read capability simplifies system programming and eliminates the need to store line characteristics in system memory.

Table 3-6. LCR

Bit	Value	Description
0	0	Divisor Latch Access Bit. Required to address RBR, THR and IER with LTADR5:7 for read or write operation.
	1	Divisor Latch Access Bit. Required to address Divisor Latches with LTADR5:7 for read or write operation.
1	0	Break Control Bit. The break is disabled.
	1	The break is enabled; causes a break condition to be transmitted to the UART when the UART is receiving. SOUT is forced to the spacing state (logic 0). This bit acts only on SOUT and has no effect on the transmitter logic.
2	0	Stick parity is disabled.
	1	Stick parity is enabled. When bits 3 & 4 are logic 1 the parity bit is transmitted and checked as logic 0. If bit 4 is logic 1 and bit 3 is a logic 0, then the parity bit is transmitted and checked as logic 1.
3	0	Even parity select bit. When bit 4 is a logic 0, an odd number of logic 1s is transmitted or checked in the data word and the parity bits.
	1	Even parity select bit. When bit 4 is a logic 1, an even number of logic 1s is transmitted or checked in the data word and the parity bits.
4	0	Parity checking is disabled.
	1	Parity Enabled; a Parity bit is generated during transmission of data (or checked during the reception of data) between the last data word bit and the stop bit of the serial data. The parity bit is used to produce an odd or even number of ones when the data word bits and the parity bit are summed.
5	0	Specifies one stop bit transmitted and received in each serial character. The receiver checks the first stop bit only, regardless of how many stop bits are selected.
	1	If 5 bit word length is selected via 6:7, one and one half stop bits are generated. If any other character length is selected, two stop bits are generated. The receiver checks the first stop bit only, regardless of how many stop bits are selected.
6:7	00	Specifies 5-bit character length.
	01	Specifies 6-bit character length..
	10	Specifies 7-bit character length.
	11	Specifies 8-bit character length.

3.9.2 Line Status Register

This register holds information about data transfer for the PPC403. Bits 3:6 are conditions that produce a receiver line status interrupt whenever the condition corresponding to the active bit is detected and the interrupt is enabled. This register is intended for read operations only and writing is not recommended.

Table 3-7. LSR

Bit	Value	Description
0	0	Receiver FIFO Error indicator. Always 0 in 16450 mode. In FIFO mode, it is reset to 0 whenever PPC403 reads LSR, provided there are no subsequent errors in the FIFO.
	1	Receiver FIFO Error indicator. Set to 1 one there are one or more instances of parity error, framing error or break indication in the FIFO.
1	0	Transmitter Empty (TEMT) indicator. Reset to logic 0 whenever the THR or the transmitter shift register (TSR) contain a character. In FIFO mode, it is reset to logic 0 whenever the transmitter FIFO or the TSR contain a character.
	1	Transmitter Empty(TEMT) indicator. Set to logic 1 when the THR and the TSR are both empty. In FIFO mode, it is set to logic 1 when the transmitter FIFO and the TSR are both empty.
2	0	Transmitter Holding Register Empty (THRE) indicator. Concurrent reset to 0 with the loading of the THR by the PPC403. In FIFO mode it is reset to 0 when at least 1 byte is written to the transmitter FIFO.
	1	Transmitter Holding Register Empty (THRE) indicator. Logic 1 when the UART is ready to accept a new character for transmission. When the THRE enable (bit 1 in the IER) is set to logic 1, the UART will issue an interrupt to the PPC403. This bit is set to logic 1 when a character is transferred from the THR to the Transmitter Shift Register. In FIFO mode this bit is set when the transmitter FIFO is empty.
3	0	Break Interrupt(BI) indicator. Reset to 0 whenever PPC403 reads LSR.
	1	Break Interrupt(BI) indicator. Set to logic 1 whenever the received data input is held at the spacing level (logic 0) for longer than a full word transmission time. The full word transmission time is the time required for the start bit, data bits (can be 5-8 bits), parity and stop bits. In FIFO mode, this error is revealed to the PPC403 when the character this error is associated with is at the top of the FIFO. Only one zero character is loaded into the receiver FIFO when a break occurs. After SIN receives the next valid start bit, and has gone into the marking state, the next character transfer is enabled.
4	0	Framing Error(FE) indicator. Reset to 0 whenever PPC403 reads LSR.
	1	Framing Error(FE) indicator. Indicates that a valid stop bit was not found in the received character. Set to logic 1 whenever Stop bit following the last data bit or parity bit is detected as logic 0 (Spacing level). In FIFO mode, this error is revealed to the PPC403 when the character this error is associated with is at the top of the FIFO. To resynchronize after a framing error, the UART will assume that the framing error was due to the next start bit, so it will sample this "start bit" twice, then take in the data.

Table 3-7. LSR

Bit	Value	Description
5	0	Parity Error(PE) indicator. Reset to 0 whenever PPC403 reads LSR.
	1	Parity Error(PE) indicator. Indicates that the received data character does not have the correct parity as determined by the even parity select bit of the LCR. Set to logic 1 upon detection of a parity error. In FIFO mode, this error is revealed to the PPC403 when the character this error is associated with is at the top of the FIFO.
6	0	Overrun Error(OE) indicator. Reset to 0 whenever PPC403 reads LSR.
	1	Overrun Error(OE) indicator. Data in the RBR was read by the PPC403 before the next character was transferred into the RBR, hence the original data was lost. OE detected when this bit is 1. In FIFO mode, if the incoming data continues to fill the FIFO beyond the trigger level, an OE will occur only after the FIFO is completely full and the entire next character has been received in the receiver shift register. The PPC403 is informed of the OE immediately upon occurrence. The character in the shift register will be overwritten and will not be transferred to the FIFO.
7	0	Receiver Data Ready(DR) indicator. Reset to 0 when all data has been read from the receiver FIFO or the receiver buffer register.
	1	Receiver Data Ready(DR) indicator. An entire incoming character has been received into the RBR or receiver FIFO.

3.9.3 FIFO Control Register

The FIFO control register has the same address as the IIR and is a write-only register. This register is used perform FIFO control operations such as selecting the type of DMA signaling, setting the receiver FIFO trigger levels, clearing the FIFOs, and enabling the FIFOs.

Table 3-8. FCR

Bit	Value	Description
0:1	00	Receiver FIFO level trigger level set to 01 bytes.
	01	Receiver FIFO level trigger level set to 04 bytes.
	10	Receiver FIFO level trigger level set to 08 bytes.
	11	Receiver FIFO level trigger level set to 14 bytes.
2	-	Reserved
3	-	Reserved
4	0	DMA mode select. If bit 0 of this register is 1 (FIFOs are enabled), this bit set to logic 0 will cause RXRDY and TXRDY outputs to change from DMA mode 1 to DMA mode 0 (single transfer DMA).
	1	DMA mode select. If bit 0 of this register is 1 (FIFOs are enabled), this bit set to logic 1 will cause RXRDY and TXRDY outputs to change from DMA mode 0 to DMA mode 1 (multiple transfer DMA).
5	0	Transmitter FIFO reset. The 1 that is written into this position is self-clearing.
	1	Transmitter FIFO reset. A logic 1 written here will clear all bytes in the transmitter FIFO and reset all of its counter logic to 0. The transmitter shift register is not cleared by this bit.
6	0	Receiver FIFO reset. The 1 that is written into this position is self-clearing.
	1	Receiver FIFO reset. A logic 1 written here will clear all bytes in the receiver FIFO and reset all of its counter logic to 0. The receiver shift register is not cleared by this bit.
7	0	FIFO disabled. Resets both receiver and transmitter FIFOs. Data is automatically cleared from both FIFOs when changing to and from FIFO and 16450 modes.
	1	FIFO Enable. Writing a logic 1 here enables both the receiver and transmitter FIFOs.

3.9.4 Interrupt Identification Register

The UART prioritizes interrupts into four levels that are recorded in the interrupt identification register (IIR). The levels of interrupt in the order of their priority are receiver line status, received data ready, transmitter holding register empty, and modem status. When the PPC403GB accesses the IIR, the UART records new interrupts, but does not change its current contents until the access by the PPC403GB is complete. The UART indicates the highest priority interrupt pending to the PPC403GB via the IIR.

Table 3-9. IIR

Bit	Value	Description			
0	0	FIFOs disabled (bit 0 of FCR = 0)			
	1	FIFOs enabled (bit 0 of FCR = 1)			
1	0	FIFOs disabled (bit 0 of FCR = 0)			
	1	FIFOs enabled (bit 0 of FCR = 1)			
2	0	Always 0			
3	0	Always 0			
4:6		Bits 6 & 5 are used to indicate the interrupt priority as shown below. Bit 4 is always 0 in 16450 mode. In FIFO mode, when a timeout interrupt is pending, bits 4 & 5 are set to logic 1.			
		Priority Level	Interrupt Type	Interrupt Source	Interrupt Reset Control
	011	1st	Receiver Line Status	Overrun, Parity or Framing Error, or Break Interrupt	Read LSR
	010	2nd	Received Data Available	Receiver data available or trigger level reached.	Read RBR, or FIFO drops below trigger level.
	110	2nd	Character Timeout Indication	No characters have been removed from or input to the receiver FIFO during the last 4 char. times and it contains at least 1 char. during this time.	Read RBR
	001	3rd	Transmitter Holding Register Empty	Transmitter Holding Register Empty	Read IIR (if source of interrupt) or write THR
	000	4th	Modem Status	Clear to Send, Data Set Ready, Ring Indicator or Data Carrier Detect	Read MSR
7	0	Interrupt is pending. IIR contents may be used as a pointer to the appropriate interrupt service routine.			
	1	No interrupt is pending.			

3.9.5 Interrupt Enable Register

Five types of interrupts are enabled by the interrupt enable register (IER). The INTR output signal can be activated by any of the five interrupt types. Resetting bits 4:7 of the IER disables the interrupt system. Each interrupt type is enabled by setting one of bits 4:7. To disable an interrupt is to prevent it from activating the INTR output signal, and to prevent it from being shown as active in the IIR. All other system functions, LSR, MSR, continue to operate in their normal manner.

Table 3-10. IER

Bit	Value	Description
0	0	Always 0
1	0	Always 0
2	0	Always 0
3	0	Always 0
4	0 1	Modem Status Interrupt disabled Modem Status Interrupt enabled
5	0 1	Receiver Line Status Interrupt disabled Receiver Line Status Interrupt enabled
6	0 1	Transmitter Holding Register Empty Interrupt disabled Transmitter Holding Register Empty Interrupt enabled
7	0 1	Received Data Available Interrupt (and timeout interrupts in FIFO mode) disabled. Received Data Available Interrupt (and timeout interrupts in FIFO mode) enabled.

3.9.6 Modem Control Register

The interface between the modem, data set, or peripheral device emulating a modem, and the UART, is controlled by the MCR.

Table 3-11. MCR

Bit	Value	Description
0	0	Always 0
1	0	Always 0
2	0	Always 0
3	0 1	Loopback mode deactivated. Loopback mode activated. Provides a local loopback feature for diagnostic testing of the UART. The following occurs SOUT is set to the marking state (logic 1) SIN is disconnected The output of the transmitter shift register feeds the input of the receiver shift register ("this provides the loopback") The 4 modem control inputs $\overline{SP_DSR}$, $\overline{SP_CTS}$, RI_N, and DCD_N are disconnected The 4 modem control outputs $\overline{SP_DTR}$, $\overline{SP_RTS}$, OUT1_N, and OUT2_N are set to a logic 1 (their inactive state) The 4 modem control outputs are connected internally to the 4 modem control inputs Hence, the data that is transmitted is immediately received This allows the verification of the UART transmit and received data paths. The receiver and transmitter interrupts are fully operational. Their sources are external to the UART core. Also operational are the modem control interrupts, but their source is now the lower 4 bits of the MCR instead of the modem control inputs to the UART. The IER still controls the interrupts.
4	0 1	OUT2_N (auxiliary user designated output) set to 1. OUT2_N (auxiliary user designated output) set to 0.
5	0 1	OUT1_N (auxiliary user designated output) set to 1. OUT1_N (auxiliary user designated output) set to 0.
6	0 1	$\overline{SP_RTS}$ (Request To Send, active low) output set to 1. $\overline{SP_RTS}$ (Request To Send, active low) output set to 0.
7	0 1	$\overline{SP_DTR}$ (Data Terminal Ready, active low) output set to 1. $\overline{SP_DTR}$ (Data Terminal Ready, active low) output set to 0.

3.9.7 Modem Status Register

The PPC403GB monitors the present state of the modem (or peripheral device) control lines by reading the MSR. In addition, the MSR has 4 bits to indicate whether any of the modem (or peripheral device) control lines have changed state.

Table 3-12. MSR

Bit	Value	Description
0	-	Complement of Data Carrier Detect (DCD_N) input to the UART. In loopback mode (bit 3 of MCR is 1), it is equivalent to OUT 2 (bit 4) in the MCR.
1	-	Complement of Ring Indicator(RI_N) input to the UART. In loopback mode (bit 3 of MCR is 1), it is equivalent to OUT 1 (bit 5) in the MCR.
2	-	Complement of Data Set Ready(SP_DSR) input to the UART. In loopback mode (bit 3 of MCR is 1), it is equivalent to DTR (bit 7) in the MCR.
3	-	Complement of Clear to Send($\overline{\text{SP_CTS}}$) input to the UART. In loopback mode (bit 3 of MCR is 1), it is equivalent to RTS (bit 6) in the MCR.
4	0 1	Reset to 0 whiner the PPC403 reads the MSR. Delta Data Carrier Detect(DDCD). Indicates that the Data Carrier Detect(DCD_N) input to the UART has changed from 0 to 1 since the PPC403 last read the MSR. A MODEM Status Interrupt is generated.
5	0 1	Reset to 0 whiner the PPC403 reads the MSR. Trailing edge of Ring Indicator (TERI) detector. Indicates that the Ring Indicator (RI_N) input to the UART has changed from 0 to 1 since the PPC403 last read the MSR. A MODEM Status Interrupt is generated.
6	0 1	Reset to 0 whenever the PPC403 reads the MSR. Delta Data Set Ready (DDSR). Indicates that the Data Set Ready (SP_DSR) input to the UART has changed state since the PPC403 last read the MSR. A MODEM Status Interrupt is generated.
7	0 1	Reset to 0 whenever the PPC403 reads the MSR. Delta Clear to Send (DCTS). Indicates that the Clear to Send ($\overline{\text{SP_CTS}}$) input to the UART has changed state since the PPC403 last read the MSR. A MODEM Status Interrupt is generated.

3.9.8 Scratchpad Register

A scratchpad register intended for use by the programmer as a temporary data holder is provided in this UART. It does not control the UART operation in any way.

Table 3-13. Register Summary

Name	Address	Description							
		Data Bit 0	Data Bit 1	Data Bit 2	Data Bit 3	Data Bit 4	Data Bit 5	Data Bit 6	Data Bit 7
RBR	0000	Data Bit 0	Data Bit 1	Data Bit 2	Data Bit 3	Data Bit 4	Data Bit 5	Data Bit 6	Data Bit 7
THR	0000	Data Bit 0	Data Bit 1	Data Bit 2	Data Bit 3	Data Bit 4	Data Bit 5	Data Bit 6	Data Bit 7
IER	0001	0	0	0	0	Enable MODEM Status Interrupt (EDSSI)	Enable Receiver Line Status Interrupt (ELSI)	Enable Transmitter Holding Register Empty Interrupt (ETBEI)	Enable Received Data Available Interrupt (ERBFI)
IIR	X010	FIFOs Enabled	FIFOs Enabled	0	0	Interrupt ID Bit(2)	Interrupt ID Bit(1)	Interrupt ID Bit(0)	"0" if Interrupt Pending
FCR	X010	RCVR Trigger (MSB)	RCVR Trigger (LSB)	Reserved	Reserved	DMA Mode Select	XMIT FIFO Reset	RCVR FIFO Reset	FIFO Enable
LCR	X011	Divisor Latch Access Bit (DLAB)	Set Break	Stick Parity	Even Parity Select (EPS)	Parity Enable (PEN)	Number of Stop Bits (STB)	Word Length Select Bit 1 (WLS1)	Word Length Select Bit 0 (WLS0)
MCR	X100	0	0	0	Loop	Out 2	Out 1	Request To Send (RTS)	Data Terminal Ready (DTR)
LSR	X101	Error in Receiver FIFO	Transmitter Empty (TEMT)	Transmitter Holding Register (THRE)	Break Interrupt (BI)	Framing Error (FE)	Parity Error (PE)	Overrun Error (OE)	Data Ready (DR)
McR	X110	Data Carrier Detect (DCD)	Ring Indicator (RI)	Data Set Ready (DSR)	Clear To Send (CTS)	Delta Data Carrier Detect (DDCD)	Trailing Edge Ring Indicator (TERI)	Delta Data Set Ready (DDSR)	Delta Clear To Send (DCTS)
LSR	X111	Data Bit 7	Data Bit 6	Data Bit 5	Data Bit 4	Data Bit 3	Data Bit 2	Data Bit 1	Data Bit 0
MSR	1000	Data Bit 7	Data Bit 6	Data Bit 5	Data Bit 4	Data Bit 3	Data Bit 2	Data Bit 1	Data Bit 0
SCR	1001	Data Bit 15	Data Bit 14	Data Bit 13	Data Bit 12	Data Bit 11	Data Bit 10	Data Bit 9	Data Bit 8

Note: The D address in Table 3-13 is represented by the DLAB bit, bit0 in the LCR.

3.10 FIFO Operation

3.10.1 Receiver

Receiver interrupts occur when the receiver FIFO and receiver interrupts are enabled by setting FCR bit 7 and IER bit 7 to logic 1.

The received data available interrupt is issued when the number of characters in the FIFO reach the trigger level programmed into the FCR. This interrupt is reset to 0 when the FIFO character count drops below this trigger level.

The IIR received data available indicator is issued when the number of characters in the FIFO reach the trigger level programmed into the FCR. This indicator is reset to 0 when the FIFO character count drops below this trigger level.

The receiver line status interrupt (IIR=C6) is a top priority interrupt; the received data available interrupt (IIR=C4) is a second priority interrupt.

The data ready bit (bit 7 of LSR) is set as soon as a character is transferred from the shift register to the receiver FIFO. This bit is reset when the FIFO is empty.

Receiver timeout interrupts occur as described below when the receiver FIFO and receiver interrupts are enabled by setting FCR bit 7 and IER bit 7 to logic 1.

A FIFO timeout will occur when:

- At least one character is in the receiver FIFO
- No serial characters have been received for 4 serial character time periods.
- The PPC403 has not read the FIFO for 4 serial character time periods. A serial character time period is

$$1/(\text{baud rate}) * (\text{start bits} + \text{word length} + \text{parity bits} + \text{stop bits})$$

For example, the serial character time period for an 8 bit word with 1 parity bit, 2 stop bits at 56Kbaud is $1/(56000) * (1 + 8 + 1 + 2) = 214.3\mu\text{sec}$, so the timeout word occur after 857.1 μsec if the above conditions hold.

When a timeout interrupt occurs, it is cleared and its timer reset when the PPC403 reads one character from the receiver FIFO.

When there is no timeout interrupt, the timer reset after a new serial character is received or the PPC403 reads the receiver FIFO.

3.10.2 Transmitter

Transmitter interrupts occur when the transmitter FIFO and transmitter interrupts are enabled by setting FCR bit 7 and IER bit 6 to 1.

The transmitter holding register interrupt (IIR=C2) occurs when the transmitter holding register is empty, and is cleared as soon as the transmitter holding register is written to or

the IIR is read. 1-16 characters may be written to the transmitter FIFO while servicing this interrupt.

The transmitter FIFO empty indications are delayed by 1 character time minus the last stop bit time whenever the following event occurs:

- THRE=1 and there are less than 2 bytes simultaneously present in the transmit FIFO since the last THRE=1. If bit 0 of the FCR is 1 (FIFOs enabled), the first transmitter interrupt after changing this bit (bit 0 of the FCR) is immediate.

RCVR FIFO trigger level interrupts, received data available interrupts, and character timeouts all have equivalent second interrupt priority. Current transmitter holding register empty interrupt and XMIT FIFO empty have equivalent third interrupt priority.

3.10.3 Polled Mode

When bit 7 of the FCR=1 (FIFOs enabled) and bits 4:7 of the IER are set to 0 (interrupts disabled), the UART is in FIFO polled mode of operation. The receiver and transmitter are controlled separately, so either can be in polled mode of operation. In polled mode, software checks the LSR to see the status of the receiver and/or transmitter.

Bits 3:6 of the LSR specify which errors (if any) have occurred. Character status errors are handled in the same way as in interrupt mode. Since bit 5 of the IER = 0, the IIR is not affected. Bit 7 of the LSR is set as long as there is at least one character in the receiver FIFO. If the transmitter FIFO is empty, bit 2 of the LSR is set to 1. If the transmitter FIFO and the transmitter shift register are empty, bit 1 of the LSR is set to 1. If any errors in the receiver FIFO occur, bit 0 of the LSR is set to 1.

In FIFO polled mode, there is no character timeout or trigger levels; however, the FIFOs still hold characters.

Chapter 4. Smart Card Interface

The smart card interface (SCI) provides the communication interface between a smart card and the PPC403GB. When the SCI detects a card in the slot of a smart card receptacle (not part of this chip), the SCI initiates communication with the smart card. Software parses the information from the answer-to-reset, then initializes the IFD registers to communicate at a particular protocol.

4.1 Features

The SCI provides the following features:

- Compatibility with ISO/IEC 7816-3
- Support for T0 and T1 protocols
- Uses system clock or externally supplied oscillator
- A configurable 16-byte FIFO supports:
 - 8-byte read FIFO and 8-byte write FIFO
 - 16-byte read-only FIFO
 - 16-byte write-only FIFO
- Hardware error checking
- 8-bit memory mapped registers
- 32-bit OPB bus interface supports 8-, 16- or 32-bit accesses to FIFO

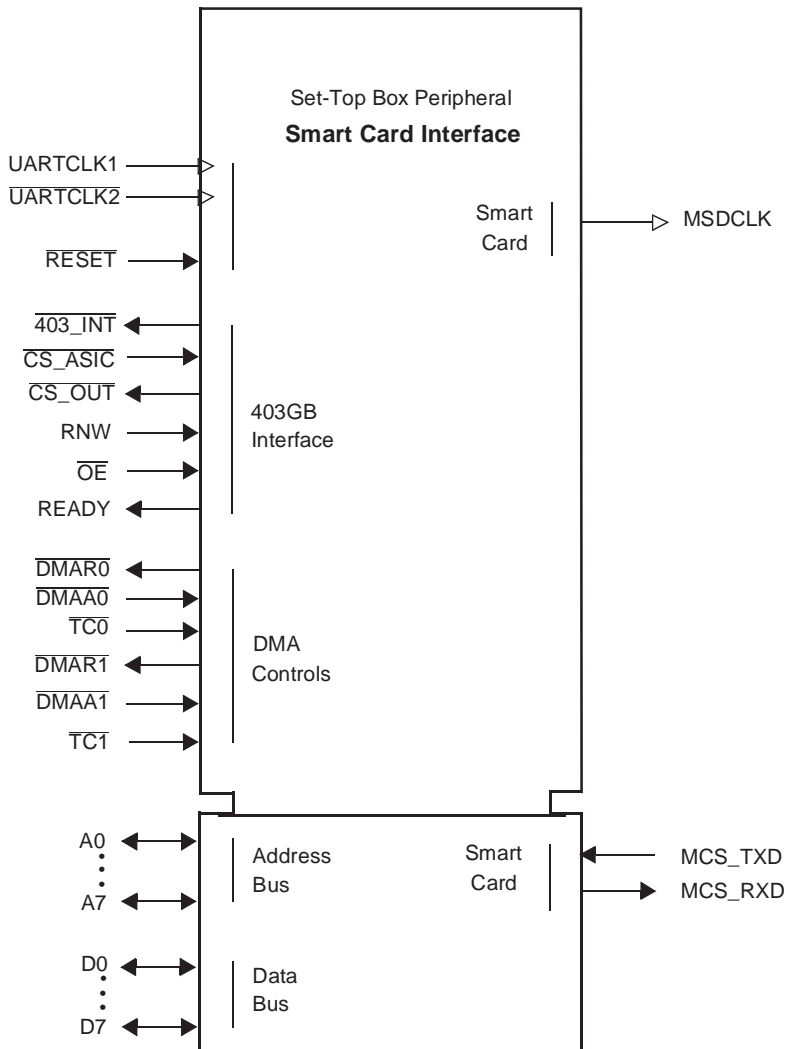


Figure 4-1. Smart Card Interface Controller

4.2 Operation

The SCI supports two smart cards with external multiplexing (controlled by the GPIO macro). With the appropriate software, the SCI can support both T0 and T1 versions of the protocol (in asynchronous mode only). SCI is based on a standard UART, with the addition of special hardware for parity error signaling back to the smart card. Special timing hardware is not required. All timing parameters specified by ISO/IEC 7816-3 are measured using PPC403GB programmable internal time (PIT) or using software timing functions.

To the PPC403GB, communication with the SCI is very much like communication with a serial port (Refer to Section 4.5, “Registers,” on p. 4-5, .). An interrupt from the SCI signals that data should be read to, or read from the SCI. Data is written as 8-, 16- or 32-bit data transfers to the transmit buffer for transmission to the card, or a byte or word is read from the receive buffer. T1 protocol is implemented in this fashion or can use a DMA channel to unburden the host. The SCI performs redundancy checks on the data and monitors for timing errors. When an error occurs, the host is interrupted in order for software to handle the recovery.

The RX and TX signals go to separate pins and are tied together externally. Because they are tied together externally, the receiver portion is disabled during transmission and the interface does not read back the characters being transmitted. This feature is enabled with a microcode bit under software control. The microcode bit, enabled by reset, must be disabled by the software if this functionality is not desired.

The TX pin is an open drain signal pulled up externally to support the parity error detection protocol provided in ISO/IEC 7816-3 for the T0 protocol as specified in section X.Y. According to ISO/IEC 7816-3, this parity error detection protocol is only active during answer-to-reset. A microcode-controlled bit allows the software to turn off parity error handling. When a parity error is detected, an interrupt occurs and the character is made available (in read buffer) to the PPC403GB for reading. The smart card, which is also informed of the parity error, retransmits the character.

4.3 Protocols

Memory-mapped I/O is used with the T0 protocol, providing byte-level error detection. When the SCI detects errors, it notifies the PPC403GB. Retry of the error must be handled entirely by software unless the parity error retry bit is enabled. The software notes the parity error and reacts appropriately (reads and discards the bad character). The software detects the parity error on a transmission (LSR_2) and performs a retry. There is no hardware support for automatic retransmission of data, with a parity error, that was received by the smart card.

The T1 protocol supports both MMI/O or DMA transfer modes. Error detection and retry in this mode are handled entirely by software using minimal assist logic provided by the SCI. The SCI will still be able to detect a parity error if enabled to do so.

Note: There is no special hardware in the SCI to really determine the difference between the T0 and T1 protocol. The protocol setup is really controlled by software.

4.3.1 Transmission Error Detection Procedure (T0)

1. The SCI device driver loads the STBP with character to send.
2. The STBP sends out data and analyzes the error signal interval.
3. If an error is detected, the STBP sends interrupt to indicate this to the SCI device driver.
The software must then wait the appropriate time (including extra guard time) before retransmitting the character.

4.3.2 Reception Error Detection Procedure (T0)

1. The STBP detects parity error in received character.
2. The STBP asserts the error signal condition and sends an interrupt to the software driver, indicating character has been received with a parity error.
3. The SCI device driver reads or discards the bad character, and re-arms the PIT to measure the work waiting time.

4.4 Programming Interface

Address bits 0:3 address the macros of the STBP chip. Bit 0 is msb. Address bits 4:7 are used internally by each macro.

Table 4-1. Smart Card Busmode Register

Name	Access	Address 4:7	Definition
Busmode	R/W	0x1000	Bit 0: smartcard clk sel bit 0
			Bit 1: smartcard clk sel bit 1
			Bit 2: rcvdma1
			Bit 3: rcvdma0
			Bit 4: rcvmio
			Bit 5: txdma1
			Bit 6: txdma0
			Bit 7: txmio

4.4.1 Clocking

All data rates supported by the ISO/IEC 7816-3 up to 307200 Kbps are supported.

There is a UARTCLK input to the STBP chip. This input is muxed into the internally derived UART clock under microcode control. The internally-derived UARTCLK is Sysclk/2 or 13.5mhz for STBP. Tables of divide values for various input clock values are provided in Section 3.8, "UART and Smart Card Speed Programming," on p. 3-9, .

Table 4-2. Smartcard Clock Selection

clk sel 0,1	Clock Used
00	sysclk/2
01	UartClk1
10	UartClk2
11	Undefined

4.5 Registers

The system programmer accesses any of the registers summarized in the tables below, through the PPC403GB. These registers control the SCI operations including transmission and reception of data. Each register bit in the table has name and reset state shown. The bit number is implemented with bit 0 as msb, matching the PPC403GB numbering convention.

4.5.1 .Line Control Register

The Line Control Register(LCR) specifies the format of the asynchronous data communications exchange and sets the Divisor Latch Access Bit.

The PPC403GB can read the contents of the LCR. The read capability simplifies system programming, and eliminates the need to store line characteristics in system memory.

Table 4-3. LCR

Bit	Value	Description
0	0	Divisor Latch Access Bit. Required to address RBR, THR and IER with LTADR5:7 for read or write operation.
	1	Divisor Latch Access Bit. Required to address Divisor Latches with LTADR5:7 for read or write operation.
1	0	Break Control Bit. The break is disabled.
	1	The break is enabled; causes a break condition to be transmitted to the SCI when the SCI is receiving. SOUT is forced to the spacing state (logic 0). This bit acts only on SOUT and has no effect on the transmitter logic.

Table 4-3. LCR

Bit	Value	Description
2	0	Stick parity is disabled.
	1	Stick parity is enabled. When bits 3 & 4 are logic 1 the parity bit is transmitted and checked as logic 0. If bit 4 is logic 1 and bit 3 is a logic 0, then the parity bit is transmitted and checked as logic 1.
3	0	Even parity select bit. When bit 4 is a logic 0, an odd number of logic 1s is transmitted or checked in the data word and the parity bits.
	1	Even parity select bit. When bit 4 is a logic 1, an even number of logic 1s is transmitted or checked in the data word and the parity bits.
4	0	Parity checking is disabled.
	1	Parity Enabled; a Parity bit is generated during transmission of data (or checked during the reception of data) between the last data word bit and the stop bit of the serial data. The parity bit is used to produce an odd or even number of ones when the data word bits and the parity bit are summed.
5	0	Specifies one stop bit transmitted and received in each serial character. The receiver checks the first stop bit only, regardless of how many stop bits are selected.
	1	If 5 bit word length is selected via 6:7, one and one half stop bits are generated. If any other character length is selected, two stop bits are generated. The receiver checks the first stop bit only, regardless of how many stop bits are selected.
6:7	00	Specifies 5-bit character length.
	01	Specifies 6-bit character length..
	10	Specifies 7-bit character length.
	11	Specifies 8-bit character length.

4.5.2 Line Status Register

Information concerning the data transfer is held for the PPC403 in this register. Bits 3:6 are conditions that produce a receiver line status interrupt whenever the condition corresponding to the active bit is detected and the interrupt is enabled. This register is intended for read operations only and writing is not recommended.

Table 4-4. LSR

Bit	Value	Description
0	0	Receiver FIFO Error indicator. Always 0 in 16450 mode. In FIFO mode, it is reset to 0 whenever PPC403 reads LSR, provided there are no subsequent errors in the FIFO.
	1	Receiver FIFO Error indicator. Set to 1 one there are one or more instances of parity error, framing error or break indication in the FIFO.

Table 4-4. LSR

Bit	Value	Description
1	0	Transmitter Empty (TEMT) indicator. Reset to logic 0 whenever the THR or the transmitter shift register (TSR) contain a character. In FIFO mode, it is reset to logic 0 whenever the transmitter FIFO or the TSR contain a character.
	1	Transmitter Empty (TEMT) indicator. Set to logic 1 when the THR and the TSR are both empty. In FIFO mode, it is set to logic 1 when the transmitter FIFO and the TSR are both empty.
2	0	Transmitter Holding Register Empty (THRE) indicator. Concurrent reset to 0 with the loading of the THR by the PPC403. In FIFO mode it is reset to 0 when at least 1 byte is written to the transmitter FIFO.
	1	Transmitter Holding Register Empty (THRE) indicator. Logic 1 when the SCI is ready to accept a new character for transmission. When the THRE enable (bit 1 in the IER) is set to logic 1, the SCI will issue an interrupt to the PPC403. This bit is set to logic 1 when a character is transferred from the THR to the TSR. In FIFO mode this bit is set when the transmitter FIFO is empty.
3	0	Break Interrupt (BI) indicator. Reset to 0 whenever PPC403 reads LSR.
	1	Break Interrupt(BI) indicator. Set to logic 1 whenever the received data input is held at the spacing level (logic 0) for longer than a full word transmission time. The full word transmission time is the time required for the start bit, data bits (can be 5-8 bits), parity and stop bits. In FIFO mode, this error is revealed to the PPC403 when the character this error is associated with is at the top of the FIFO. Only one zero character is loaded into the receiver FIFO when a break occurs. After SIN receives the next valid start bit, and has gone into the marking state, the next character transfer is enabled.
4	0	Framing Error(FE) indicator. Reset to 0 whenever PPC403 reads LSR.
	1	Framing Error(FE) indicator. Indicates that a valid stop bit was not found in the received character. Set to logic 1 whenever Stop bit following the last data bit or parity bit is detected as logic 0 (Spacing level). In FIFO mode, this error is revealed to the PPC403 when the character this error is associated with is at the top of the FIFO. To resynchronize after a framing error, the SCI will assume that the framing error was due to the next start bit, so it will sample this "start bit" twice, then take in the data.
5	0	Parity Error(PE) indicator. Reset to 0 whenever PPC403 reads LSR.
	1	Parity Error(PE) indicator. Indicates that the received data character does not have the correct parity as determined by the even parity select bit of the LCR. Set to logic 1 upon detection of a parity error. In FIFO mode, this error is revealed to the PPC403 when the character this error is associated with is at the top of the FIFO.
6	0	Overrun Error(OE) indicator. Reset to 0 whenever PPC403 reads LSR.
	1	Overrun Error(OE) indicator. Data in the RBR was read by the PPC403 before the next character was transferred into the RBR, hence the original data was lost. OE detected when this bit is 1. In FIFO mode, if the incoming data continues to fill the FIFO beyond the trigger level, an OE will occur only after the FIFO is completely full and the entire next character has been received in the receiver shift register. The PPC403 is informed of the OE immediately upon occurrence. The character in the shift register will be overwritten and will not be transferred to the FIFO.

Table 4-4. LSR

Bit	Value	Description
7	0	Receiver Data Ready(DR) indicator. Reset to 0 when all data has been read from the receiver FIFO or the receiver buffer register.
	1	Receiver Data Ready(DR) indicator. An entire incoming character has been received into the RBR or receiver FIFO.

4.5.3 FIFO Control Register

The FIFO control register has the same address as the IIR and is a write-only register. This register is used perform FIFO control operations such as selecting the type of DMA signaling, setting the receiver FIFO trigger levels, clearing the FIFOs, and enabling the FIFOs.

Table 4-5. FCR

Bit	Value	Description
0:1	00	Receiver FIFO level trigger level set to 01 bytes.
	01	Receiver FIFO level trigger level set to 04 bytes.
	10	Receiver FIFO level trigger level set to 08 bytes.
	11	Receiver FIFO level trigger level set to 14 bytes.
2	-	Reserved
3	-	Reserved
4	0	DMA mode select. If bit 0 of this register is 1 (FIFOs are enabled), this bit set to logic 0 will cause RXRDY and TXRDY outputs to change from DMA mode 1 to DMA mode 0 (single transfer DMA).
	1	DMA mode select. If bit 0 of this register is 1 (FIFOs are enabled), this bit set to logic 1 will cause RXRDY and TXRDY outputs to change from DMA mode 0 to DMA mode 1 (multiple transfer DMA).
5	0	Transmitter FIFO reset. The 1 that is written into this position is self-clearing.
	1	Transmitter FIFO reset. A logic 1 written here will clear all bytes in the transmitter FIFO and reset all of its counter logic to 0. The transmitter shift register is not cleared by this bit.
6	0	Receiver FIFO reset. The 1 that is written into this position is self-clearing.
	1	Receiver FIFO reset. A logic 1 written here will clear all bytes in the receiver FIFO and reset all of its counter logic to 0. The receiver shift register is not cleared by this bit.
7	0	FIFO disabled. Resets both receiver and transmitter FIFOs. Data is automatically cleared from both FIFOs when changing to and from FIFO and 16450 modes.
	1	FIFO Enable. Writing a logic 1 here enables both the receiver and transmitter FIFOs.

4.5.4 Interrupt Identification Register

The SCI prioritizes interrupts into four levels which are recorded in the Interrupt Identification Register (IIR). The levels of interrupt in the order of their priority are receiver line status, received data ready, transmitter holding register empty, and modem status. When the PPC403GB accesses the IIR, the SCI records new interrupts, but does not change its current contents until the access by the PPC403GB is complete. The SCI indicates the highest priority interrupt pending to the PPC403GB via the IIR.

Table 4-6. IIR

Bit	Value	Description			
0	0	FIFOs disabled (bit 0 of FCR = 0)			
	1	FIFOs enabled (bit 0 of FCR = 1)			
1	0	FIFOs disabled (bit 0 of FCR = 0)			
	1	FIFOs enabled (bit 0 of FCR = 1)			
2	0	Always 0			
3	0	Always 0			
4:6		Bits 6 & 5 are used to indicate the interrupt priority as shown below. Bit 4 is always 0 in 16450 mode. In FIFO mode, when a timeout interrupt is pending, bits 4 & 5 are set to logic 1.			
		Priority Level	Interrupt Type	Interrupt Source	Interrupt Reset Control
	011	1st	Receiver Line Status	Overrun, Parity or Framing Error, or Break Interrupt	Read LSR
	010	2nd	Received Data Available	Receiver data available or trigger level reached.	Read RBR, or FIFO drops below trigger level.
	110	2nd	Character Timeout Indication	No characters have been removed from or input to the receiver FIFO during the last 4 char. times and it contains at least 1 char. during this time.	Read RBR
	001	3rd	Transmitter Holding Register Empty	Transmitter Holding Register Empty	Read IIR (if source of interrupt) or write THR
	000	4th	Modem Status	Clear to Send, Data Set Ready, Ring Indicator or Data Carrier Detect	Read MSR
7	0	Interrupt is pending. IIR contents may be used as a pointer to the appropriate interrupt service routine.			
	1	No interrupt is pending			

4.5.5 Interrupt Enable Register

Five types of interrupts are enabled via the IER. The INTR (interrupt) output signal can be activated by any of the five interrupt types. Resetting bits 4:7 of the IER totally disables the interrupt system. Each interrupt type can be enabled by setting one of these bits. To disable an interrupt is to prevent it from activating the INTR output signal, and to prevent it from being shown as active in the IIR. All other system functions, LSR, MSR, continue to operate in their normal manner.

Table 4-7. IER

Bit	Value	Description
0	0	Always 0
1	0	Always 0
2	0	Always 0
3	0	Always 0
4	0	Modem Status Interrupt disabled
	1	Modem Status Interrupt enabled
5	0	Receiver Line Status Interrupt disabled
	1	Receiver Line Status Interrupt enabled
6	0	Transmitter Holding Register Empty Interrupt disabled
	1	Transmitter Holding Register Empty Interrupt enabled
7	0	Received Data Available Interrupt (and timeout interrupts in FIFO mode) disabled.
	1	Received Data Available Interrupt (and timeout interrupts in FIFO mode) enabled.

4.5.6 Modem Control Register

The interface between the modem, data set, or peripheral device emulating a modem, and the SCI, is controlled by the MCR.

Table 4-8. MCR

Bit	Value	Description
0	0	Always 0
1	0	Always 0
2	0	Always 0
3	0 1	Loopback mode deactivated. Loopback mode activated. Provides a local loopback feature for diagnostic testing of the SCI. The following occurs SOUT is set to the marking state (logic 1) SIN is disconnected The output of the transmitter shift register feeds the input of the receiver shift register ("this provides the loopback") The 4 modem control inputs $\overline{SP_DSR}$, $\overline{SP_CTS}$, RI_N , and DCD_N are disconnected The 4 modem control outputs $\overline{SP_DTR}$, $\overline{SP_RTS}$, $OUT1_N$, and $OUT2_N$ are set to a logic 1 (their inactive state) The 4 modem control outputs are connected internally to the 4 modem control inputs Hence, the data that is transmitted is immediately received This allows the verification of the SCI transmit and received data paths. The receiver and transmitter interrupts are fully operational. Their sources are external to the SCI core. Also operational are the modem control interrupts, but their source is now the lower 4 bits of the MCR instead of the modem control inputs to the SCI. The IER still controls the interrupts.
4	0 1	$OUT2_N$ (auxiliary user designated output) set to 1. $OUT2_N$ (auxiliary user designated output) set to 0.
5	0 1	$OUT1_N$ (auxiliary user designated output) set to 1. $OUT1_N$ (auxiliary user designated output) set to 0.
6	0 1	$\overline{SP_RTS}$ (Request To Send, active low) output set to 1. $\overline{SP_RTS}$ (Request To Send, active low) output set to 0.
7	0 1	$\overline{SP_DTR}$ (Data Terminal Ready, active low) output set to 1. $\overline{SP_DTR}$ (Data Terminal Ready, active low) output set to 0.

4.5.7 Modem Status Register

The PPC403GB can monitor the present state of the modem (or peripheral device) control lines by reading the MSR. In addition, the MSR has 4 bits to indicate whether any of the modem (or peripheral device) control lines have changed state.

Table 4-9. MSR

Bit	Value	Description
0	-	Complement of Data Carrier Detect (DCD_N) input to the SCI. In loopback mode (bit 3 of MCR is 1), it is equivalent to OUT 2 (bit 4) in the MCR.
1	-	Complement of Ring Indicator(RI_N) input to the SCI. In loopback mode (bit 3 of MCR is 1), it is equivalent to OUT 1 (bit 5) in the MCR.
2	-	Complement of Data Set Ready(SP_DSR) input to the SCI. In loopback mode (bit 3 of MCR is 1), it is equivalent to DTR (bit 7) in the MCR.
3	-	Complement of Clear to Send($\overline{\text{SP_CTS}}$) input to the SCI. In loopback mode (bit 3 of MCR is 1), it is equivalent to RTS (bit 6) in the MCR.
4	0 1	Reset to 0 whiner the PPC403 reads the MSR. Delta Data Carrier Detect(DDCD). Indicates that the Data Carrier Detect(DCD_N) input to the SCI has changed from 0 to 1 since the PPC403 last read the MSR. A MODEM Status Interrupt is generated.
5	0 1	Reset to 0 whiner the PPC403 reads the MSR. Trailing edge of Ring Indicator (TERI) detector. Indicates that the Ring Indicator (RI_N) input to the SCI has changed from 0 to 1 since the PPC403 last read the MSR. A MODEM Status Interrupt is generated.
6	0 1	Reset to 0 whenever the PPC403 reads the MSR. Delta Data Set Ready (DDSR). Indicates that the Data Set Ready (SP_DSR) input to the SCI has changed state since the PPC403 last read the MSR. A MODEM Status Interrupt is generated.
7	0 1	Reset to 0 whenever the PPC403 reads the MSR. Delta Clear to Send (DCTS). Indicates that the Clear to Send ($\overline{\text{SP_CTS}}$) input to the SCI has changed state since the PPC403 last read the MSR. A MODEM Status Interrupt is generated.

4.5.8 Scratchpad Register

A scratchpad register intended for use by the programmer as a temporary data holder is provided in this SCI. It does not control the SCI operation in any way.

Table 4-10. Register Summary

Name	Address	Description							
RBR	0000	Data Bit 0	Data Bit 1	Data Bit 2	Data Bit 3	Data Bit 4	Data Bit 5	Data Bit 6	Data Bit 7
THR	0000	Data Bit 0	Data Bit 1	Data Bit 2	Data Bit 3	Data Bit 4	Data Bit 5	Data Bit 6	Data Bit 7
IER	0001	0	0	0	0	Enable MODEM Status Interrupt (EDSSI)	Enable Receiver Line Status Interrupt (ELSI)	Enable Transmitter Holding Register Empty Interrupt (ETBEI)	Enable Received Data Available Interrupt (ERBFI)
IIR	X010	FIFOs Enabled	FIFOs Enabled	0	0	Interrupt ID Bit(2)	Interrupt ID Bit(1)	Interrupt ID Bit(0)	"0" if Interrupt Pending
FCR	X010	RCVR Trigger (MSB)	RCVR Trigger (LSB)	Reserved	Reserved	DMA Mode Select	XMIT FIFO Reset	RCVR FIFO Reset	FIFO Enable
LCR	X011	Divisor Latch Access Bit (DLAB)	Set Break	Stick Parity	Even Parity Select (EPS)	Parity Enable (PEN)	Number of Stop Bits (STB)	Word Length Select Bit 1 (WLS1)	Word Length Select Bit 0 (WLS0)
MCR	X100	0	Error Detection / Correction	Ignore RX Input When Transmit	Loop	Out 2	Out 1	Request To Send (RTS)	Data Terminal Ready (DTR)
LSR	X101	Error in Receiver FIFO	Transmitter Empty (TEMT)	Transmitter Holding Register (THRE)	Break Interrupt (BI)	Framing Error (FE)	Parity Error (PE)	Overrun Error (OE)	Data Ready (DR)
McR	X110	Data Carrier Detect (DCD)	Ring Indicator (RI)	Data Set Ready (DSR)	Clear To Send (CTS)	Delta Data Carrier Detect (DDCD)	Trailing Edge Ring Indicator (TERI)	Delta Data Set Ready (DDSR)	Delta Clear To Send (DCTS)
LSR	X111	Data Bit 7	Data Bit 6	Data Bit 5	Data Bit 4	Data Bit 3	Data Bit 2	Data Bit 1	Data Bit 0
MSR	1000	Data Bit 7	Data Bit 6	Data Bit 5	Data Bit 4	Data Bit 3	Data Bit 2	Data Bit 1	Data Bit 0
SCR	1001	Data Bit 15	Data Bit 14	Data Bit 13	Data Bit 12	Data Bit 11	Data Bit 10	Data Bit 9	Data Bit 8

Chapter 5. Serial Communications Protocol Interface

Serial Communications Protocol (SCP) is an inter-chip digital link that performs data exchanges via a full-duplex, synchronous, character-oriented serial communications link. The three-pin interface is receive, transmit and clock. The same clock is used for transmit and receive functions. Clock rates from 31.25KHz to 1MHz are supported. In master mode, SCP provides for clock inversion and reverse data mode.

In DMA mode, Read Data Available interrupts are not presented to the IntCntl logic.

5.1 Programming Information

The following tables contain addressing and register definitions. Address bits 0:3 address the macros of the STBP chip. Bit 0 is msb. Address bits 4:7 are used internally by each macro.

Table 5-1. SCP Addressing

Name	Mode	Address 4:7	Definition
spmode	r/w	x000	Bit 0: loop Bit 1: ci Bit 2: revdat Bit 3: en
rxdata	r	x001	Bit 0 - bit 7: received byte
txdata	r/w	x010	Bit 0 - bit 7: byte to be transmitted
spcom	w	x011	Bit 0: str
status	r/w	x100	Bit 0: received byte ready Bit 1: receive buffer overflow
busmode	r/w	x101	Bi 0: Bit 1: dma0 Bit 2: dma1
scprate	r/w	x110	Initializes to "000".

Table 5-2. SPCOM Register Definitions

Bit	Function
0	In slave mode, writing 1 loads the transmit shift register with a new value (when idle). In master mode, the same occurs and a transmit/receive cycle begins. This is cleared by the SCP when the next cycle begins.

Receive buffer overflow is set if a byte is received and the buffer is full. It is reset when written with 1.

Received byte ready is set at the end of a transmit/receive cycle. It is reset when the buffer is read.

Table 5-3. SPMODE Register Definitions

Bit	Function
0	Set to 1 for internal loopback of transmitter to receiver.
1	Set to 1 for clock inversion.
2	Set to 1 to transmit & receive LSB rather than MSB first.
3	Set to 1 to enable SCP operation.

Table 5-4. BUSMODE Register Definitions

Bit	Function
0	Set to 1 to operate in normal mode
1	Set to 1 to generate SI_DMAREq0 on received byte ready
2	Set to 1 to generate SI_DMAREq1 on received byte ready

Reset initializes this register to 0b199x_xxxx.

Table 5-5. SPRATE s

Bits 5:7	Selects
000	31.25KHz
001	62.5KHz
010	125KHz
011	250KHz

Table 5-5. SPRATE s

100	500KHz
101	1MHz
110	1MHz
111	1MHz

5.2 DMA Operation

In dma0 or dma1 modes, no interrupts are generated for received byte ready.

Note: In SCP, “received byte ready” also indicates “transmit byte sent”.

5.3 SCP Interrupt Operation

Each bit of the status register is also a source of interrupt. A single interrupt output is the logical OR of these two conditions:

1. One byte received (equivalent to one byte transmitted, or cycle completion)
2. Receive buffer overflow (a byte is received when the buffer is full)

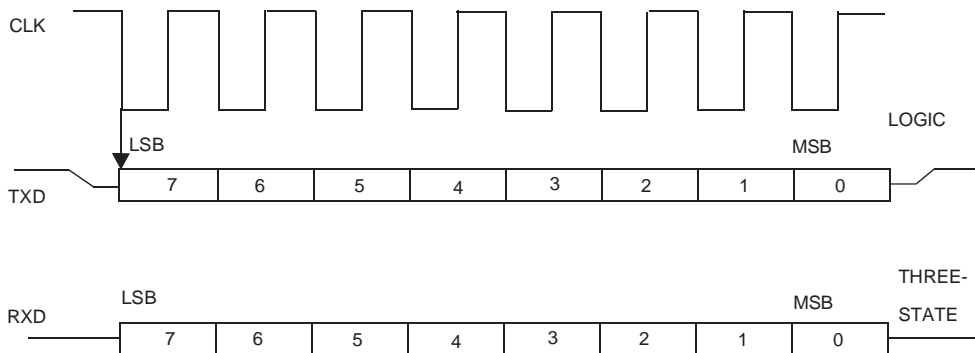
Condition 1 generates an interrupt only in normal mode. It is cleared when the receive byte is read. Condition 2 is cleared by writing ‘1’ to status register bit 6.

5.3.1 SCP Loopback Operation

When SCP is in loopback mode, the received data is inverted from the sent data.

5.4 SCP Timing

Figure 5-1 shows the timing with clock inversion and data reversal.



NOTE: By design, in this block, 7 is lsb on the serial line. By setting 'revdat' this block outputs and receives bit 7 lsb first.

Figure 5-1. SCP Timing (With Clock Inversion and revdat Set)

Chapter 6. 1284 Parallel Port

This port is an integrated IEEE 1284 high speed bidirectional parallel interface.

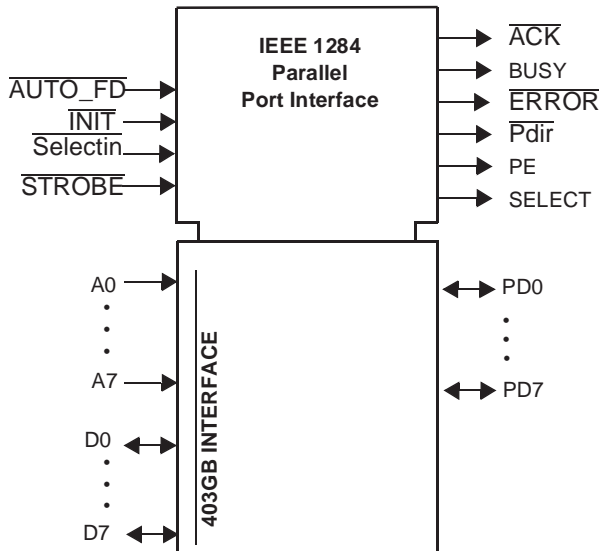


Figure 6-1. 1284 I/O Signals

This macro meets the IEEE 1284 parallel interface specification with the following features:

- **Compatibility Mode**
Refers to an asynchronous, byte-wide forward (PPC403GB-to-peripheral) channel.
- **Nibble Mode**
Refers to an asynchronous, reverse (peripheral-to-PPC403GB) channel, under the control of the PPC403GB.
- **Byte Mode**
Refers to an asynchronous, byte-wide reverse (peripheral-to-PPC403GB) channel, under the control of the PPC403GB. BYTE Mode is used with COMPATIBLE Mode to implement a bidirectional channel.
- **ECP mode**
Refers to the extended capability port. An asynchronous, byte-wide, bidirectional channel.

Note: In ECP mode, identification of external function (device ID string), and RLE (run length encoding) are not required by IEEE 1284.

The FIFO size provided on the STBP chip is eight bytes for transmit and eight bytes for receive.

6.1 Intercharacter Timer

The IEEE 1284 parallel port uses an intercharacter timer with the following characteristics:

- Timer times space between characters in Receive mode.
- Programmable reloadable down counter, 8 bit.
- Clocked by 1.2 ms ticks.
- 1.2 ms to 301 ms in steps of 1.2 ms.
- Register programmed by host to COUNT.
- Counter loaded with COUNT whenever idle condition (high level) is detected on the Strobe line and count started. Counter counts down as long as Strobe is idle.
- Counter stopped whenever line Strobe goes active (low). Counter is reloaded with COUNT when next idle condition occurs.
- If counter reaches zero, count shall stop and interrupt sent to host.

This timer is implemented in the STBP PWM/Timers macro. See Section 8.1.1, “IC Timers,” on p. 8-2 for more information. One signal (count) feeds the timer macro from the UART macro. The timer counts when the count signal is active. The timer is reset if the count signal drops. The timer macro causes an interrupt if a timer expires.

6.2 IEEE 1284 Operating Modes

The IEEE 1284 macro can be configured to operate in Compatibility, Nibble, Byte, and ECP modes to transfer data. All four of these modes are compliant with IEEE Std. 1284. The default mode is Compatibility mode.

The basic data transfers of these IEEE 1284 modes are given in the following sections. More in-depth timing details can be found in IEEE Std. 1284-1994.

6.2.1 Compatibility Mode

The Compatibility mode is a forward (host to peripheral) direction mode only, allowing the host to transfer a byte of data at a time to the peripheral. This mode is used to communicate with the older dot matrix and laser printers. Figure 6-2 shows the data transfer cycle.

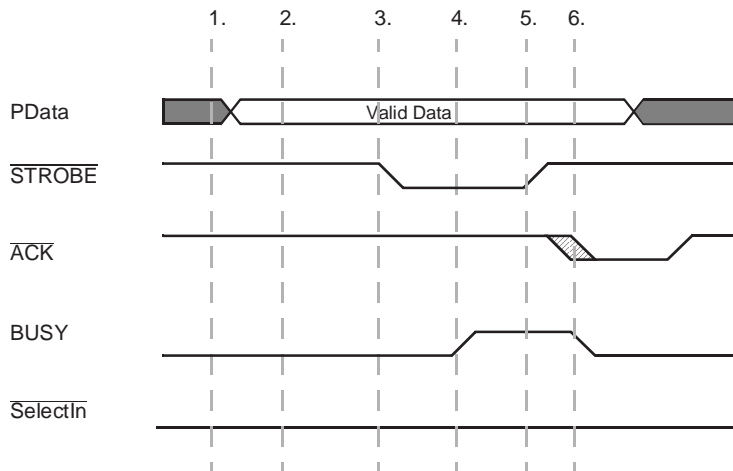


Figure 6-2. Compatibility Mode Data Transfer Cycle

Compatibility Mode Handshake

1. The data is written across the parallel port data bus.
2. Host reads to see that BUSY is low.
3. If BUSY is low, the host asserts $\overline{\text{STROBE}}$ low.
4. The macro acknowledges host by asserting the BUSY high.
5. The host de-asserts the $\overline{\text{STROBE}}$ line, latching data into the macro's registers.
6. The macro acknowledges the last byte was received with a $\overline{\text{ACK}}$ low pulse, and sets BUSY low when ready for another byte.

6.2.2 Nibble Mode

Nibble mode is a reverse (peripheral to host) direction mode only. This can be implemented with the Compatibility mode to form a bidirectional link. the peripheral can transfer four bits at a time using the status lines for data. Figure 6-3 shows the data transfer cycle.

IEEE 1284 Nibble Mode Phase transitions

1. Host indicates ready to take data by setting HostBusy ($\overline{\text{AUTOFEED}}$) low.
2. The macro places data nibble on the $\overline{\text{Error}}$, Select, PE, and BUSY.
3. The macro indicates valid nibble by asserting a low signal on the PtrClk($\overline{\text{Ack}}$).
4. The Host indicates receiving the nibble by asserting HostBusy high.
5. The macro acknowledges the host by setting PtrClk high.
6. States 1-5 are repeated for the second nibble.

Note: The host asserts IEEE 1284 Active ($\overline{\text{SelectIn}}$) high during Negotiation mode and keeps the IEEE 1284 Active ($\overline{\text{SelectIn}}$) high to show that it is in a IEEE 1284 transfer mode. The host drives the IEEE 1284 Active signal low to terminate the nibble mode and returns to compatibility mode.

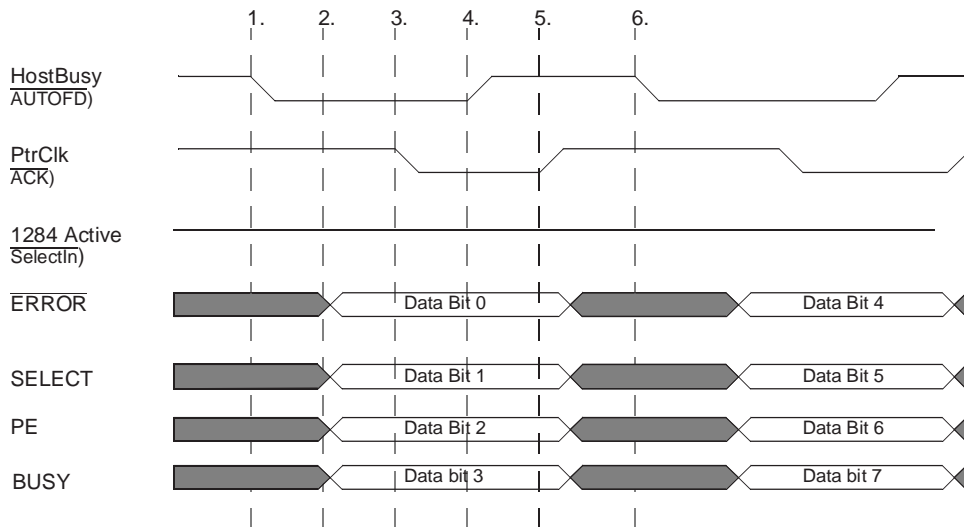


Figure 6-3. Nibble Mode Data Transfer Cycle

6.2.3 Byte Mode

Byte mode is a reverse (peripheral to host) direction mode only. Like Nibble mode, Byte mode can be implemented with the Compatibility mode to form a bidirectional link. This allows the peripheral to transfer eight bits at a time. Figure 6-4 shows the data transfer cycle.

Byte Mode signal transitions

1. The host asserts HostBusy ($\overline{\text{AUTOFD}}$) low to indicate ready to receive data.
2. The macro places data across the parallel port data bus.
3. The macro asserts PtrClk ($\overline{\text{ACK}}$) low to indicate valid byte on parallel port data bus.
4. The host asserts HostBusy high to indicate it has received the data and is not ready for another byte yet.
5. The macro asserts PtrClk high to acknowledge the host. The host pulses HostClk (STROBE) low as an acknowledgment for the macro.
6. States 1-5 are repeated for additional bytes.

Note: As in Nibble mode, IEEE 1284 Active ($\overline{\text{SelectIn}}$) is set high during Negotiation mode. It will remain high until the host decides to terminate the Byte Mode and return to compatibility mode. The macro asserts DataAvail ($\overline{\text{Error}}$) low to indicate there is reverse data available.

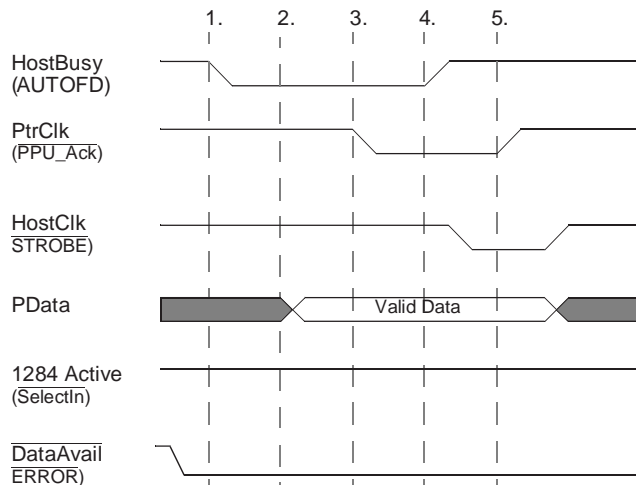


Figure 6-4. Byte Mode Data Transfer Cycle

6.2.4 ECP Mode

The ECP mode provides a high performance bidirectional communication path between the host and a peripheral. The ECP mode provides data and command cycle types in both the forward and reverse direction. A basic example of this mode in both forward and reverse directions follows.

ECP Forward Mode signal transitions

1. Host places data on to parallel port data lines. HostAck ($\overline{\text{AUTOFD}}$) indicates if the byte is data (HostAck = 1) or if it is a STROBE low to indicate valid data on parallel port data lines.
2. macro acknowledges the host by asserting PeriphAck (BUSY) high.
3. Host asserts HostClk high to indicate transfer done. This rising edge will latch data into macro's buffers.
4. macro acknowledges hosts completion of transfer and asserts PeriphAck low when ready for another byte.
5. States 1-5 are repeated for another byte transfer, this time the byte is a command instead of data.

Note: As in Nibble mode, IEEE 1284 Active ($\overline{\text{SelectIn}}$) is set high during Negotiation mode. It will remain high until the host decides to terminate the ECP Mode and return to compatibility mode.

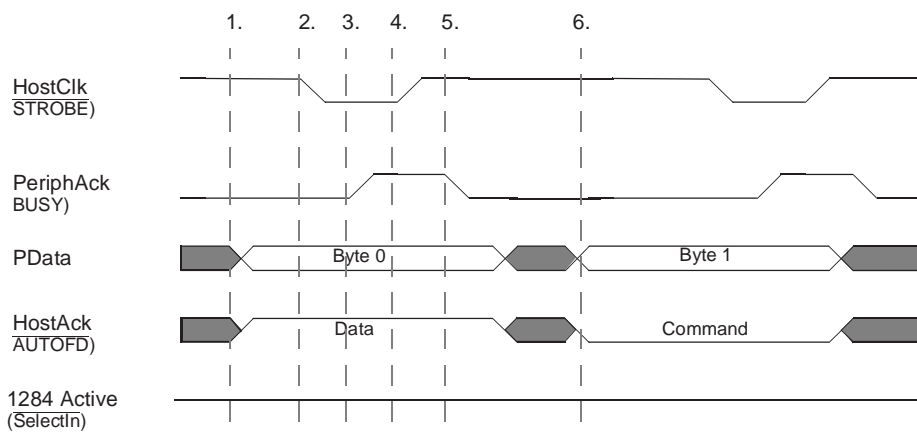


Figure 6-5. Forward ECP Mode Data Transfer Cycle

ECP Reverse Mode signal transitions

1. The Host requests a reverse channel transfer by asserting $\overline{\text{ReverseRequest}}$ ($\overline{\text{INIT}}$) low
2. The macro responds to the host by asserting $\overline{\text{AckReverse}}$ ($\overline{\text{PE}}$) low
3. The macro places data on the parallel port data lines and indicates if the byte is a command ($\text{PeriphAck (BUSY)} = 0$) or if the byte is data ($\text{PeriphAck (BUSY)} = 1$)
4. The macro asserts PeriphClk ($\overline{\text{ACK}}$) low indicating valid data on the parallel port bus.
5. The host acknowledges by setting HostAck ($\overline{\text{AUTO_FD}}$) high.
6. The macro sets the PeriphClk high. The rising edge of PeriphClk should latch the data into the host.
7. The host sets HostAck low when it is ready for another byte.
8. States 1-7 repeat for another transfer. However this time it is a command because PeriphAck is low.

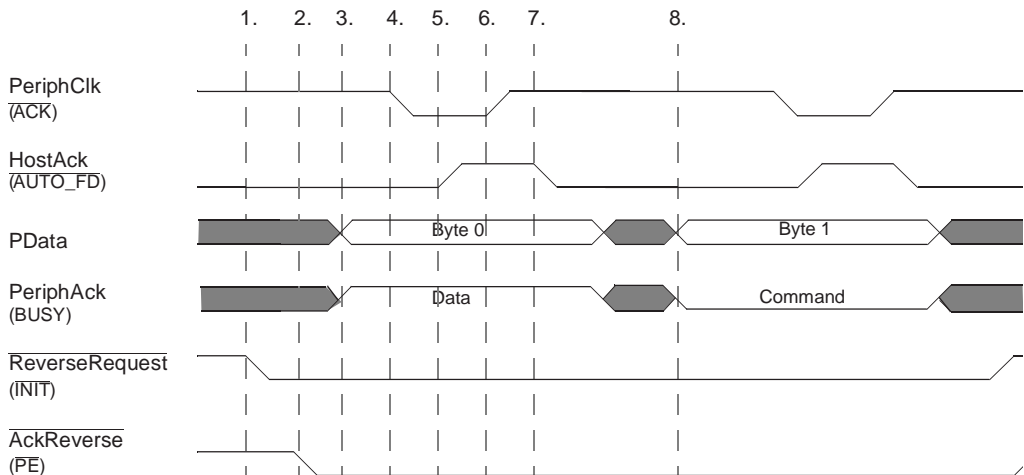


Figure 6-6. Reverse ECP Mode Data Transfer Cycle

6.3 Programming Information

The DMA requests, if enabled by the bus mode register, is split into two signals (one for send FIFO, and one for receive FIFO). Response to a DMA request can be in the form of a normal MMIO transfer, or using the corresponding DMA acknowledge signal. DMA operation is not supported when operating the IEEE 1284 in nibble mode.

Table 6-1. IEEE 1284 Address Bit 4

AUX	Address 4:7	Register
X	1000	Bus Mode Register

Table 6-2. Bus Mode Register Definitions

Name	Access	Address 4:7	Description
BMR	R/W	1000	bit7: s_dma0: 1 = send data on dma channel 0
			bit6: s_dma1: 1 = send data on dma channel 1
			bit5: r_dma0: 1 = receive data on dma channel 0
			bit4: r_dma1: 1 = receive data on dma channel 1

All four bits will be set = 0 by reset.

The DMA requests, if enabled via the bmr, is split into two signals (one for send fifo, and one for receive fifo). Response to a DMA request can be in the form of a normal memory-mapped I/O transfer, or via the corresponding DMA acknowledge signal. DMA operation is not supported when operating the IEEE 1284 in nibble mode.

6.3.1 Definition of IEEE 1284 Registers

Note: that the “A” address below is represented by the Aux bit, bit 0 in the PCR.

Address 4A567	Register	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
00000	RDR	Data 0 (msb)	Data 1	Data 2	Data 3	Data 4	Data 5	Data 6	Data 7 (lsb)
00000	RDF	Data 0 (msb)	Data 1	Data 2	Data 3	Data 4	Data 5	Data 6	Data 7 (lsb)
00000	TDR	Data 0 (msb)	Data 1	Data 2	Data 3	Data 4	Data 5	Data 6	Data 7 (lsb)
00000	TDF	Data 0 (msb)	Data 1	Data 2	Data 3	Data 4	Data 5	Data 6	Data 7 (lsb)
00000	DCR	+ busy	- ack	+ pe	+ slct	- Err	Dir	Reset	Aack
00010	DSR	BSYP	ACKP	DIRP	ECP CMD	SLCT IN	INIT	AUFD	STRB

Address 4A567	Register	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
00011	IER	Enable Rx Full	Enable Rx Empty	Enable Tx Full	Enable Tx Empty	Enable SelctIn	Enable Init	Enable AutoFd	Enable Strobe
00100	ISR	Rx Full IRQ	Rx Empty IRQ	Tx Full IRQ	Tx Empty IRQ	SelctIn IRQ	Init IRQ	AutoFd IRQ	Strobe IRQ
0X101	PCR	Aux	Conv	Rx DMA	Tx DMA	Enable Rx FIFO	Enable Tx FIFO	Clear Rx FIFO	Clear Tx FIFO
00110	RCR	Count 0 (msb)	Count 1	Count 2	Count 3	Count 4	Count 5	Count 6	Count 7 (lsb)
00111	SCR	Count 0 (msb)	Count 1	Count 2	Count 3	Count 4	Count 5	Count 6	Count 7 (lsb)
01001	RTR	Count 0 (msb)	Count 1	Count 2	Count 3	Count 4	Count 5	Count 6	Count 7 (lsb)
01010	STR	Count 0 (msb)	Count 1	Count 2	Count 3	Count 4	Count 5	Count 6	Count 7 (lsb)
01011	AAT	Count 0 (msb)	Count 1	Count 2	Count 3	Count 4	Count 5	Count 6	Count 7 (lsb)
01100	CVR	Prot Err	Nego	ECP	Byte	Nibl	Comp	100ms	1ms
01110	EDR	Err	Err	Err	Err	Astrb	Ainit	AAUFD	ASELI

6.3.2 IEEE 1284 Interface Signals

STBP chip names are used, with IEEE Std.1284 mode signal names in parentheses. “Pin” refers to the STBP pin number.

Pin function is described in the traditional (COMPATIBLE) mode of the parallel port, and a summary is given of its use in IEEE Std. 1284 mode operations where this differs from mode use. The term “reverse transfer mode” refers to the state where the peripheral sends data to the host.

Table 6-3. IEEE 1284 Interface Signals

Symbol	Pin	Name and Function
\overline{ACK} (PtrCik/ Periphclk)	79	<p>Compatible Mode - This signal is driven active low by the peripheral to acknowledge that the data on the bus has been received and it is ready to accept the next byte of data.</p> <p>IEEE Std. 1284 Mode - When the direction of the data transfer is from the peripheral to the system, this signal functions as a \overline{STROBE} and is driven active low by the peripheral to indicate that the data on PDATA(0-7) is valid. The HOST uses this signal to receive the data.</p>
BUSY (PtrBusy/ PeriphAck)	83	<p>This signal is normally inactive and is driven active high by the peripheral to indicate that it cannot accept data.</p> <p>IEEE 1284 Modes - Used to transfer Data bits 3 and 7 in Nibble Mode. Used in the data transfer handshake for ECP mode.</p>
PE (AckDataReq)	93	<p>Compatible Mode - Driven high to indicate an out of paper condition has occurred in the peripheral device. Peripherals must set Error low whenever they set PError high.</p> <p>IEEE Std. 1284 Mode - Set high to indicate IEEE Std. 1284 support during Negotiation 15 phase, then follows nDataAvail (#Error). In Nibble 7 Mode, used to transfer Data bits 2 and 6. In ECP 5 mode driven low to acknowledge nReverseRequest.</p>
SELECT (Xflag)	76	<p>Compatible Mode - Driven high to indicate the peripheral device is on line.</p> <p>IEEE Std. 1284 Mode - In Negotiation, used to respond to requested extensibility byte. State of this line depends on the mode requested. See Reference 3 on page 5 for details. Also used to transfer Data bits 1 and 5 during Nibble mode.</p>
\overline{ERROR} (nDataAvail/ nPeriohRequest)	80	<p>Compatible Mode - Driven low to indicate an error has occurred in the peripheral device. This signal is sometimes referred to as nFault.</p> <p>IEEE Std. 1284 Mode - Set high to indicate IEEE Std. 1284 compatibility during Negotiation. In reverse transfer modes, used to indicate that the peripheral has data to send to HOST.</p>

Table 6-3. IEEE 1284 Interface Signals

Symbol	Pin	Name and Function
PDATA(0:7) (AD(0:7))	95:98 101: 103 106	All Modes These eight lines make up the parallel port data bus. The data bus is driven by the HOST when it is transferring data to the peripheral; likewise, the data bus is driven by the peripheral when it is transferring data to the HOST. Communication mode setup determines which direction the data bus is driven. AD0 is the least significant bit and AD7 is the most significant bit.
<u>STROBE</u> (HostClk)	88	This active low signal is used to clock data into the peripheral device. When the peripheral device is sending data to the HOST, this signal functions as an acknowledge of data received by the HOST. There exist an internal digital filter on this pin. Any change of state on this signal should be stable for at least one clock cycle (41.67ns + / - 5%), if it is to be detected. This line should have a pullup, so that it is high when the HOST cable is disconnected.
<u>INIT</u> -Reverse Request	89	Compatible Mode - driven low for at least 50 microseconds to initialize the printer. Driven high during normal operation. IEEE Std. 1284 Mode - Used in ECP mode to place the channel in the reverse direction. There exist an internal digital filter on this pin. Any change of state on this signal should be stable for at least one clock cycle (41.67ns + / - 5%), if it is to be detected. This line should have a pullup, so that it is high when the HOST cable is disconnected.
<u>SelectIn</u> (Active)	75	Compatible 8 Mode - Not used in current IBM printers. EEE Std. 1284 4 Mode - Functions as an input. Set high by host during Negotiation 15 to request a IEEE Std. 1284 4 mode. During various transfer modes, set low by HOST 2 to return link to Compatible 8 mode. Converged 14 Mode - This pin becomes an output. This line should have a pullup, so that it is high when the HOST 2 cable is disconnected.

Table 6-3. IEEE 1284 Interface Signals

Symbol	Pin	Name and Function
<u>AUTO_FD</u> (HostBusy)	107	Compatible 8 Mode - no meaning for IBM printers. IEEE Std. 1284 4 Mode - Functions as an input. Set low during Negotiation 15 to request a IEEE Std. 1284 4 mode. In reverse transfer modes, used as part of the handshake during data transfer. Converged 14 Mode - This pin becomes an output. This line should have a pullup, so that it is high when the host cable is disconnected.
<u>Pdir</u> (PDIR)	74	Parallel Port Data Direction - This output pin controls the direction of the data bus +v=output, 0v=input (default is input).

6.4 Programming Issues

The parallel interface can be configured in several ways to maximize the throughput and interface performance for a particular system architecture.

Standard compatible pseudocode and IEEE Std. 1284 pseudocode are in Section 6.6.7 and Section 6.6.8, to aid software developers in writing programs for the parallel interface.

The Auto-Ack feature, DCR bit 0(AACK), can be used by software developers to reduce the microcode effort in programming the parallel interface. Table 6-4 shows the required microcode effort needed to program the parallel interface to support the IEEE Std.1284 with the Auto-Ack feature enabled.

Table 6-4. IEEE 1284 Signal Transition Event Summary

Event #s	Communication Phase	Handled By
0-6	Negotiation	Microcode
7-13	Nibble, Byte Mode Data Transfer	Chip in Auto Ack mode
14	Nibble Mode Host Busy	Chip in Auto Ack mode
15-17	Byte Mode Data xfer and handshake	Chip in Auto Ack mode
18-21	Nibble Mode Interrupt Phase	Chip in Auto Ack mode
22-29	Termination phase	Microcode
30-31	ECP mode Setup	Chip in Auto Ack mode
32-37	ECP Mode Forward Phase	Chip in Auto Ack mode
38-40	ECP Mode Fwd2Rev Phase	Chip in Auto Ack mode

Table 6-4. IEEE 1284 Signal Transition Event Summary

Event #s	Communication Phase	Handled By
41-46	ECP Mode Reverse Phase	Chip in Auto Ack mode
47-49	ECP Mode Rev2Fwd Phase	Chip in Auto Ack mode
50-55	Negotiation with Extensibility Link	Microcode
72-75	Host Transfer Recovery	Chip in Auto Ack mode

6.5 Register Descriptions

6.5.1 RDR

D0 (msb)	D1	D2	D3	D4	D5	D6	D7 (lsb)
DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0

- Type: R/O; DIR bit = 0 & RFIFO bit = 0;
- Address: Primary and Auxiliary, base addr+0
- POR state: 0x0

DB0:7 Data received across the parallel port bidirectional data bus when Receive FIFO mode is disabled.

Data read from RDR register, when DCR[DIR] = 0 and PCR[ENAB RFIFO] = 0, will be the last data byte strobed from parallel data bus.

Data is latched into this register on the rising edge of STROBE, if PCR[ENAB RFIFO] = 0 and the mode is not IEEE Std. 1284 reverse mode (Nibble, Byte, or ECP) or, if PCR[ENAB RFIFO]=1 and the mode is IEEE Std. 1284 Negotiation mode.

The internal busy signal, which is ORed with DCR[BUSY], will be asserted for the above conditions and will drive the BUSY pin to its active state. It will remain in that state until the RDR register is read by the PPC403GB. The internal busy will also be asserted when the DCR[AACK]=0 and data is latched into the RDR register. The internal busy will remain in the asserted state until the RDR register is read by the PPC403GB.

Note: Failing to read the RDR register will hold the internal busy signal to a 1 and will drive the BUSY pin to its active state.

6.5.2 RDF

D0 (msb)	D1	D2	D3	D4	D5	D6	D7 (lsb)
DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0

- Type: R/O; RFIFO bit = 1
- Address: Primary and Auxiliary, base addr+0
- POR state: unknown

DB0:7 Data received across the parallel port bidirectional data bus and stored in a 128x8 Dual Port RAM when Receive FIFO mode is enabled.

Data read from RDF, when PCR[ENAB RFIFO] = 1, is the byte pointed to by a internal circular pointer. The circular pointer is decremented after each read until the receive FIFO is declared empty (read circular pointer = write circular pointer).

Data is stored into the receive FIFO on the rising edge of STROBE, if PCR[ENAB RFIFO] = 1 and the mode is not IEEE Std. 1284 negotiation mode or, if PCR[ENAB RFIFO] = 1 and the mode is not IEEE Std. 1284 reverse byte mode.

The internal busy signal goes active when the FIFO is full. It remains in that state until the number of bytes in the receive FIFO is less than 128. Internal busy is ORed with DCR[BUSY] and drives the BUSY pin to its active state.

6.5.3 TDR

D0 (msb)	D1	D2	D3	D4	D5	D6	D7 (lsb)
DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0

- Type: R/W; DIR bit = 1 & SFIFO bit = 0;
- Address: Primary and Auxiliary, base addr+0
- POR state: 0x00

DB0:7 Data placed across the parallel port bidirectional data bus when Send FIFO mode is disabled.

Data written into the TDR register is placed on the parallel data bus while the DCR[DIR] = 1.

Data read from TDR register, when DCR[DIR] = 1, is the last data byte written to the TDR register

Note: Setting DCR[DIR] bit should be used with Caution. This bit should control the bidirectional drivers and could cause BUS CONFLICTS and possibly chip damage.

6.5.4 TDF

D0 (msb)	D1	D2	D3	D4	D5	D6	D7 (lsb)
DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0

- Type: W/O; SFIFO bit = 1;
- Address: Primary and Auxiliary, base addr+0
- POR state: unknown

DB0:7 Data transmitted across the parallel port bidirectional data bus when Send FIFO mode is enabled and Auto Acknowledge mode is enabled.

Data written into the TDF is stored in a 64x8 dual port ram, and placed on the parallel data bus or on the parallel control signals (Nibble mode only), after a supported IEEE Std. 1284 reverse mode (Nibble, Byte, or ECP) has been requested by the host and properly responded by microcode. The macro handles communication (Handshaking) with the host until it terminates the requested mode or a signal protocol error occurs.

6.5.5 DCR

D0 (msb)	D1	D2	D3	D4	D5	D6	D7 (lsb)
+BUSY	-ACK	+PE	+SLCT	-ERR	DIR	RESET	AACK

- Type: R/W
- Address: Primary, Base addr+1
- POR state: 0xC8

+BUSY Used to directly/indirectly control the state of the BUSY output pin. This bit is ORed with the internally generated BUSY signal. Internally generated Busy conditions are:

Data latched into RDR

Receive FIFO full condition

INIT input pin active.

In general, internal busy is generated for as long as it takes the macro to safely dispose of the data byte being transferred or an exception condition occur (ex. INIT going low). If the data destination is the RDR register, internal busy is active until the PPC403GB reads the data out of the RDR. If the data destination is the receive FIFO, internal busy goes active when the receive FIFO is full. Internal busy remains active while the receive FIFO is full. If INIT input pin is active while the mode is Compatible, the internal busy signal goes active and remains active until ISR[INIT IRQ] is set to 0 and INIT input pin is inactive (HIGH).

In Manual Mode (AACK = 0), the state of this bit is ORed with the above internally generated BUSY conditions.

If the mode is Auto-Ack mode (AACK = 1), the state of this bit is set to the Compatible mode BUSY status when it is not being used to respond to an IEEE Std. 1284 Negotiation mode request. In this mode, all internally generated busy conditions should be cleared. This bit can be set whenever a Busy condition occurs within the system.

Note: In Auto Ack mode (AACK = 1) and while the mode is reverse data transfer mode (Nibble or Byte), the state of this bit is ORed with the internal busy conditions and will be presented to the BUSY output pin during the peripheral busy status period.

ACK - Used to directly/indirectly control the state of the ACK output pin.

In Manual Mode (AACK = 0), this bit directly controls the ACK output pin.

In Auto Ack mode (AACK = 1), this bit should be set to the Compatible mode initial setting (=1) or be used to respond to an IEEE Std. 1284 negotiation mode request.

+PE - Used to directly/indirectly control the state of the PE output pin.

In Manual Mode (AACK = 0), this bit directly controls the PE output pin.

In Auto Ack mode (AACK = 1), this bit should reflect the Compatible mode paper error status or be used to respond to an IEEE Std. 1284 negotiation mode request.

Note: In Auto Ack mode (AACK =1), the state of this bit is presented to the PE output pin at predefined times. These times are:

At the end of a Compatible mode data transfer.

During a IEEE Std. 1284 negotiation mode request, the state of this bit will be drive the PE output pin immediately.

+SLCT - Used to directly/indirectly control the state of the SELECT output pin.

In Manual mode (AACK = 0), this bit directly controls the SELECT output pin.

In Auto Ack mode (AACK = 1), this bit should be set to the Compatible mode initial setting (=1) or be used to respond to an IEEE Std. 1284 negotiation mode request.

-ERR - Used to directly or indirectly control the state of the $\overline{\text{ERROR}}$ output pin.

In Manual mode (AACK = 0), this bit directly controls the $\overline{\text{ERROR}}$ output pin.

In Auto Ack mode (AACK = 1), this bit should reflect the Compatible mode error status or be used to respond to an IEEE Std. 1284 negotiation mode request.

Note: In Auto Ack mode (AACK =1), the state of this is be presented to the $\overline{\text{ERROR}}$ output pin at predefined times. These times are:

At the end of a Compatible mode data transfer.

During a IEEE Std. 1284 negotiation mode request, the state of this bit drives the $\overline{\text{ERROR}}$ output pin immediately.

DIR - Used to directly/indirectly control the direction of Data flow via the $\overline{\text{Pdir}}$ output pin.

If set to 0 (default), $\overline{\text{Pdir}}$ will be at 0v, and data flows from remote device to STBP chip. If set to 1, $\overline{\text{Pdir}}$ is at +5v, and data flows from STBP chip to remote device.

When the mode is Auto Ack mode (AACK=1), this bit is ignored, and the macro determines the appropriate state of the $\overline{\text{Pdir}}$ output pin. This bit should be left in the "0" state if Auto Ack is used.

RESET - If this bit is set to 1, the macro resets.

The macro clears this bit after initialization is complete (one clock cycle).

AACK -

If this bit is set to 1, the macro generates the $\overline{\text{ACK}}$ pulse in Compatible mode, its width determined by the AAT. In IEEE Std. 1284 modes this bit enables Auto-Handshaking between the STBP chip and the remote device. It is strongly recommended that this bit be used for Compatible and IEEE Std. 1284 modes of operation.

6.5.6 DSR

D0 (msb)	D1	D2	D3	D4	D5	D6	D7 (lsb)
BSYP	ACKP	DIRP	ECP CMD	SLCTIN	INIT	AUFD	STRB

- Type: R/W; Bits 1 & 3 R/W; all others R/O;
- Address: Primary, Base addr+2
- POR state: Hi byte: x'4' Low byte: reflects current state of corresponding input pins.

BSYP - This bit indicates the true state of the BUSY pin on the STBP chip.

ACKP - This bit indicates the true state of the ACK pin on the STBP chip.

DIRP - This bit indicates the true state of the Pdir pin on the STBP chip.

ECP CMD - This bit indicates that an IEEE 1284 ECP mode command has been received.

This bit is set when the macro detects that ECP mode is being used for forward data transfer and a CMD byte is latched into the macro. This bit w clears when the data in the receive FIFO i s emptied.

Note: When this bit is set, the interrupt pin is driven high until the receive FIFO is emptied. This interrupt source does not appear in ISR register and cannot be masked on themacro.

SLCTIN - State of the SelectIn line on the interface when Converged mode is inactive.

This bit l drives the SelectIn line in Converged mode.

INIT - State of the INIT signal.

AUFD - State of the $\overline{\text{AUTO_FD}}$ line on the interface.

This bit drives the $\overline{\text{AutoFd}}$ line in Converged mode.

STRB - State of the $\overline{\text{STROBE}}$ line on the interface.

6.5.7 IER

D0 (msb)	D1	D2	D3	D4	D5	D6	D7 (lsb)
ENAB RFULL	ENAB REMP	ENAB SFULL	ENAB SEMPT	ENAB SELIN	ENAB INIT	ENAB AUFD	ENAB STRB

- Type: R/W
- Address: Primary, base addr+3
- POR state: 00

This register is used to mask interrupts which will cause the +INT_P pin to go active. The corresponding status bit in the interrupt status register will still be set if the interrupt condition occurs.

ENAB RFULL - When this bit is set to 1, an interrupt occurs when the number of bytes in the receive FIFO reaches the threshold set in the receive FIFO threshold register.

ENAB REMPT - When this bit is set to 1, an interrupt occurs when the number of bytes in the receive FIFO has gone to zero.

ENAB SFULL - When this bit is set to 1, an interrupt occurs when the number of bytes in the send FIFO reaches the threshold set in the send FIFO threshold register.

ENAB SEMPT - When this bit is set to 1, an interrupt occurs when the number of bytes in the send FIFO has gone to zero.

ENAB SLIN - When this bit is set to 1, a rising or falling edge of SelectIn generates an interrupt. A rising edge also puts the macro in IEEE 1284 mode. This interrupt should always be enabled if the system expects to use any of the IEEE 1284 communication modes.

ENAB INIT When this bit is set to 1, a rising or falling edge of INIT generates an interrupt.

ENAB AUFD - When this bit is set to 1, a rising or falling edge of AUTOFD generates an interrupt.

ENAB STRB - When this bit is set to 1, a rising or falling edge of $\overline{\text{STROBE}}$ generates an interrupt.

6.5.8 ISR

D0 (msb)	D1	D2	D3	D4	D5	D6	D7 (lsb)
RFULL IRQ	REMP IRQ	SFULL IRQ	SEMP IRQ	SELIN IRQ I	INIT IRQ	AUFD IRQ	STRB IRQ

- Type: R/W
- Address: Primary, base addr+4
- POR state: 00

The Interrupt Status Register is a read/write register; reading the register gives the status of the various interrupt sources. These bits are set even if the interrupt is masked in the IER. Interrupts are cleared by writing zero to the corresponding bit position, at this same address.

RFULL IRQ - This bit is set to 1, when the receive FIFO contains RFIFO threshold number of bytes. The RFIFO threshold can be changed to generate an interrupt condition. It is cleared by writing zero to this bit position.

REMP IRQ - This bit is set to 1 when the receive FIFO goes from containing a byte(s) of data to containing no bytes. It is cleared by writing zero to this bit position.

SFULL IRQ - This bit is set to 1 when the send FIFO contains SFIFO threshold number of bytes. The SFIFO threshold can be changed to generate an interrupt condition. It is cleared by writing zero to this bit position.

SEMP IRQ - This bit is set to 1 when the send FIFO goes from containing one byte of data to containing no bytes. It is cleared by writing zero to this bit position.

SELIN IRQ - This bit is set to 1 if the macro detects a rising or falling edge on the SelectIn line, and the CONV bit is 0. It is cleared by writing zero to this bit position.

INIT IRQ - This bit is set to 1 if the STBP chip detects a rising or falling edge on the $\overline{\text{INIT}}$ line. It is cleared by writing zero to this bit position. In Compatible mode, clearing this bit when $\overline{\text{INIT}}$ is high will lower the Busy signal if it was raised during a high to low transition on $\overline{\text{INIT}}$.

AUFD IRQ - When this bit is set to 1, the STBP chip detected a rising or falling edge on the AUTO_FD line, and PCR[CONV] is 0. It is cleared by writing zero to this bit position.

STRB IRQ - When this bit is set to 1, the macro latched a byte from the remote device on the rising edge of strobe.

6.5.9 PCR

D0 (msb)	D1	D2	D3	D4	D5	D6	D7 (lsb)
AUX	CONV	RCV DMA	SND DMA	ENAB RFIFO	ENAB SFIFO	CLR RFIFO	CLR SFIFO

- Type: R/W
- Address: Primary and Auxiliary, base addr+5
- POR state: 00

AUX - When this bit is set to 0, primary registers are accessed. When this bit is set to 1, auxiliary registers are accessed.

CONV - When this bit is set to 1, Converged mode is selected. $\overline{\text{INIT}}$ line and $\overline{\text{SelectIn}}$ lines become outputs, and the CONV output pin is driven low.

RCV DMA - When this bit is set to 1 and $\text{DCR}[\text{DIR}] = 0$, the PDRQ output pin can be used as a DMA request line for received data. PDRQ is driven high to request that a DMA transfer take place to empty the receive FIFO, if it is enabled and the number of bytes in the FIFO is equal to the threshold value set up in the RFIFO threshold register.

The state of the PDRQ signal reflects the above condition if the $\text{DCR}[\text{DIR}]$ is set to 0. In Auto-Acknowledge mode, the state of this bit does not change the DIR output signal (bidirectional buffers) after the macro is put into an IEEE Std. 1284 mode.

PDRQ is driven low when the receive FIFO has been emptied.

Note: Once PDRQ has gone low, it will not go high again until the receive FIFO has filled to the threshold level. PDRQ can also be forced low (with $\text{PCR}[\text{SND DMA}] = 0$) by clearing this bit, setting $\text{PCR}[\text{CLR RFIFO}] = 1$ or by setting $\text{DCR}[\text{DIR}] = 1$.

Note: DMA controller transfer size can be at maximum, 128 bytes.

SND DMA - When this bit is set to 1, the PDRQ output pin can be used as a DMA request line for transmitted data. PDRQ is driven high to request that a DMA transfer take place to fill the send FIFO to the threshold value, if it is enabled and the FIFO is currently below the threshold value

The state of the PDRQ signal reflects the above condition if the $\text{DCR}[\text{DIR}]$ is set to 1. In Auto-Acknowledge mode the state of this bit will not change the DIR output signal (bidirectional buffers) after the macro is put into an IEEE Std. 1284 mode.

PDRQ will be driven low when the send FIFO is equal to or above the threshold level. PDRQ can also be forced low (with $\text{PCR}[\text{RCV DMA}] = 0$) by clearing this bit, setting $\text{PCR}[\text{CLR SFIFO}] = 1$ or by setting $\text{DCR}[\text{DIR}] = 0$.

Note: DMA controller transfer size can be at maximum, 128 bytes.

ENAB RFIFO - When this bit is set to 1, the receive FIFO is active.

The receive FIFO receives bytes from the remote device as long as the receive FIFO has not reached its maximum capacity (128 bytes). Interrupts are latched into the ISR register when the number of bytes in the receive FIFO are equal to or greater than receive FIFO threshold.

ENAB SFIFO - When this bit is set to 1, the send FIFO is active.

Data stored in the send FIFO is sent to the remote device if AACK bit is set to 1 (Auto-Handshaking).

Note: Clearing this bit while in an IEEE Std. 1284 reverse and forward mode will result in the macro returning to compatible mode.

CLR RFIFO - Write 1 to clear the receive FIFO. This bit is cleared in hardware once the pointers are reset.

CLR SFIFO - Write 1 to clear the send FIFO. This bit is cleared in hardware once the pointers are reset.

6.5.10 RCR

D0 (msb)	D1	D2	D3	D4	D5	D6	D7 (lsb)
8 bit value							

- Type: R/W
- Address: Primary, base addr+6
- POR state: 00

8 bit value - This register holds the number of bytes currently in the receive FIFO. Maximum value = 128.

Note: This register should not be read while the host is transferring data. Busy could be used to hold prevent the host from sending data while this register is being read. An alternate method would be to use RFULL, REMPTY and receive FIFO threshold bits/byte to determine the status of the receive FIFO count.

6.5.11 SCR

D0 (msb)	D1	D2	D3	D4	D5	D6	D7 (lsb)
	7 bit value						

- Type: R/W
- Address: Primary, base addr+7
- POR state: 00

7 bit value - This register holds the number of bytes currently in the send FIFO. Maximum value=64.

6.5.12 RTR

D0 (msb)	D1	D2	D3	D4	D5	D6	D7 (lsb)
	8 bit value						

- Type: R/W Address:
- Auxiliary, base addr+1
- POR state: 0x80

8 bit value - This register sets the RFIFO Threshold for the receive FIFO. When number of bytes in the receive FIFO is equal to or greater than this value, the RFULL IRQ bit in the ISR is set.

The maximum threshold value is 128. Upon reset, the threshold is at the maximum setting.

6.5.13 STR

D0 (msb)	D1	D2	D3	D4	D5	D6	D7 (lsb)
	7 bit value						

- Type: R/W
- Address: Auxiliary, base addr+2
- POR state: 0x40

7 bit value - This register sets the SFIFO Threshold for the Send FIFO. When number of bytes in the send FIFO is equal to or greater than this value, the SFULL IRQ bit in the ISR is set.

The maximum threshold value is 64. Upon reset, the threshold is at the maximum setting.

6.5.14 AAT

D0 (msb)	D1	D2	D3	D4	D5	D6	D7 (lsb)
8 bit value							

- Type: R/W
- Address: Auxiliary, base addr+3
- POR state: 00

8 bit value - This register sets the ACK pulse width for use with the Auto Acknowledge feature of the macro. When DCR[AACK] is set to 1, the macro will generate a \overline{ACK} pulse after the rising edge of \overline{STROBE} . The pulse width is dependent upon the value of this register. Writing zero to this register, results in 0x0C being stored. Any other value will produce a pulse of width = (value)*41.67ns (assuming a 24 MHz clock input to the macro). Upon reset, this register will be 0x0C, indicating a 500ns pulse width.

6.5.15 CVR

D0 (msb)	D1	D2	D3	D4	D5	D6	D7 (lsb)
PROT ERR	NEGO	ECP	BYTE	NIBL	COMP	100 MS	1 MS

- Type: R/O
- Address: Auxiliary, base addr+4
- POR state: 00

PROT ERR - This bit will be set if a protocol error is detected by the macro. This bit is cleared after this register is read. The macro defaults to Compatible mode when this bit is set.

NEGO - When this bit is high, the macro is in Negotiation mode.

ECP - When this bit is high, the macro is in ECP mode.

BYTE - When this bit is high, the macro is in Byte mode.

NIB - When this bit is high, the macro is in Nibble mode.

COMP - When this bit is high, the macro is in Compatibility mode.

100 MS - This bit is used in the Converged Mode interface to indicate 100ms has passed since BUSY was raised. This bit is only operational if the CONV bit is set.

1 MS - This bit is used in the Converged Mode interface to indicate that 1ms has passed since BUSY was lowered. This bit is only operational if the CONV bit is set.

6.5.16 EDR

D0 (msb)	D1	D2	D3	D4	D5	D6	D7 (lsb)
Error Condition				ASTRB	AINIT	AAUFD	ASELI

- Type: R/O
- Address: Auxiliary, base addr+5
- POR state: 00

Error Condition - This field should be used to report the internal state of macro when a protocol error occurs. This field can also be used to track IEEE Std. 1284 Negotiation events in the chip in the absent of a protocol error. Refer to Table 6-5 for additional information

ASTRB - When this bit is high, this signal was detected causing a protocol error.

AINIT - When this bit is high, this signal is detected causing a protocol error.

AAUFD - When this bit is high, this signal is detected causing a protocol error.

ASELI - When this bit is high, this signal is detected causing a protocol error.

Table 6-5. Negotiation Mode Events using EDR

IEEE 1284 Negotiation Events	CVR Register Value	Error condition field (bits 7-4)
1	0x40	1
2	0x40	2
3	0x40	3
4	0x40	4
5	0x40	4
6	0x40	--

6.6 External Interface

Figure 6-7 shows the recommended external interface and filter designs for the IEEE 1284 port.

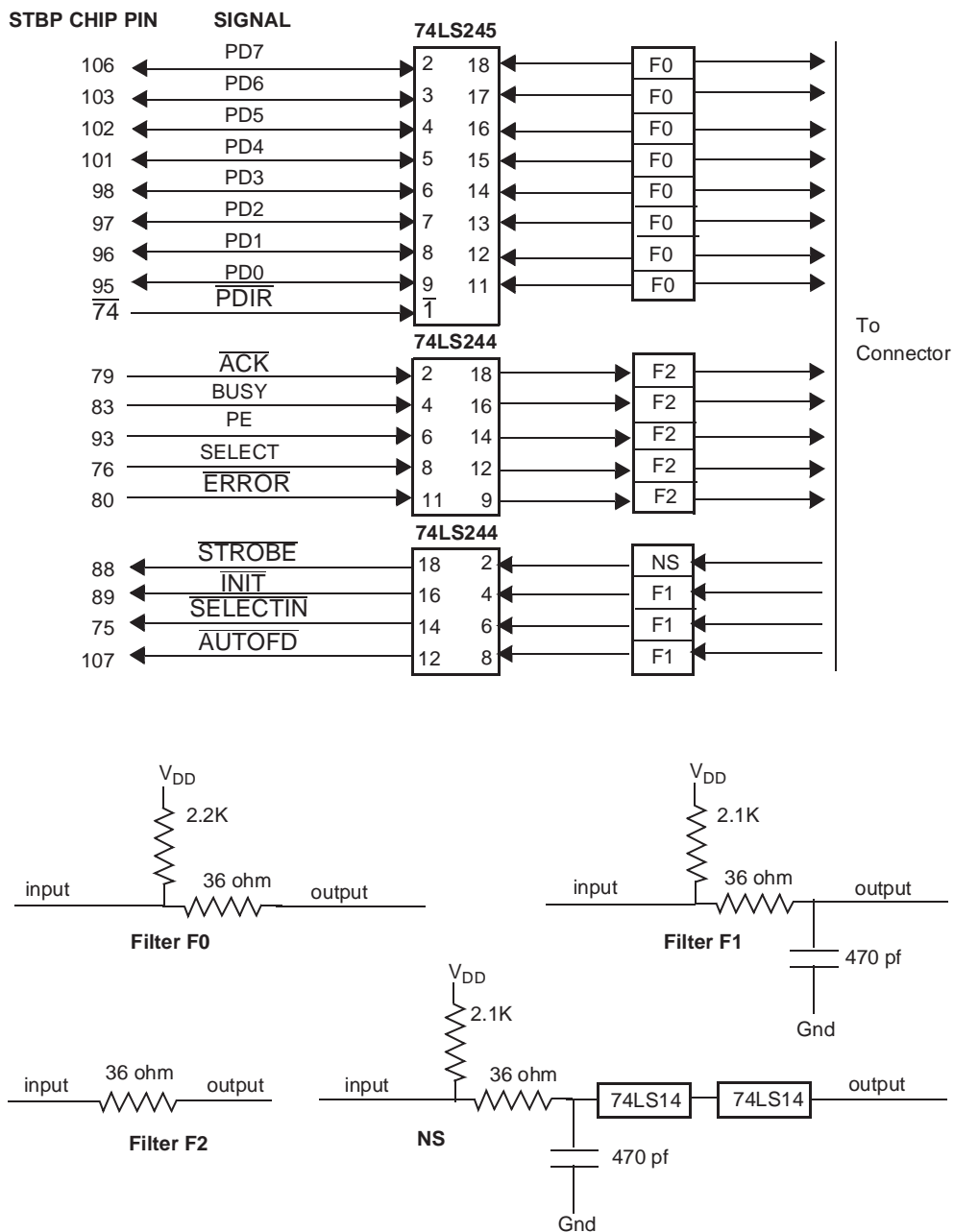


Figure 6-7. External Interface and Filters

6.7 Standard Compatible Pseudo-Code

This pseudocode can be used as an aid to develop a program to support one or all of the possible chip configurations (DMA, Receive FIFO, Auto-Acknowledge) for the standard parallel interface mode.

The standard parallel interface can be configured for maximum throughput by enabling DMA, Receive FIFO and Auto-Ack bits in the chip. The device function(s) and interrupt routine below, should be used as an example for low level device control. A high level interface example is not shown since it may vary across different architectures. The order in which the device function(s) appear below, should be the order in which the upper level device driver (application) call these functions or similar functions to setup the chip for operation. Some of these functions are optional. Each device function and interrupt routine can be modified to support only 1 or all possible chip configuration. The device mode check in most functions should be used as the indicator as to whether the line(s) of code are needed for a particular mode of operation. A buffer manager example is also illustrated to show the flow of data in a system. The buffering scheme may be different in other systems.

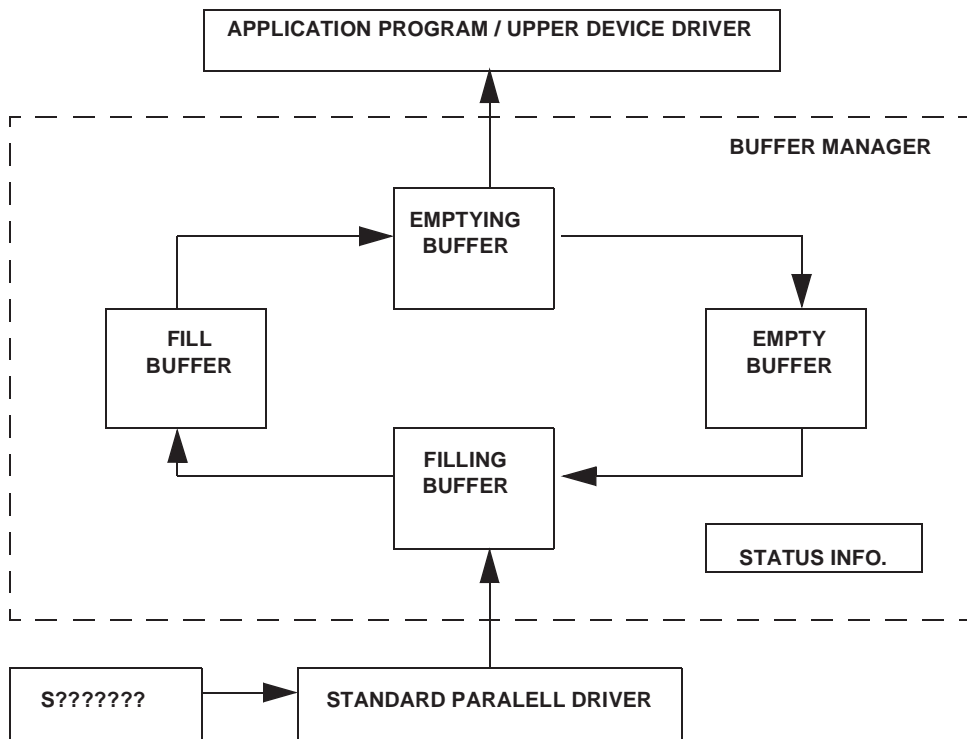


Figure 6-8. Buffer Manager (Standard Driver)

initialize()

```

{

Set DCR bit 0(RESET);                                // soft reset chip.

}
```

check_interface_reset()

```

{

Read all Register(s) and compare to Default Values;

if (Error = TRUE) {
    call Error_Recovery_Routine();
}

}
```

configure_device_std_mode (dev_mode)

```

{

    if (dev_mode = 0) {                                // Standard Compatible.

        Set IER bit 0,2(ENAB STRB,ENAB INIT);

    elseif (dev_mode = 1)                                // Standard Compatible with AACK

        Set DCR bit 0(AACK);                            // Enabled.

        Set IER bit 2(ENAB INIT);

    elseif (dev_mode = 2)                                // Standard Compatible with

        Set PCR bit 3(ENAB RFIFO);                        // Recv. Fifo Enabled.

        Set RTR value(Threshold);

        Set IER bit 0,2,7(ENAB STRB,ENAB INIT,ENAB RFULL);

    elseif (dev_mode = 3)                                // Standard Compatible with

        Set PCR bit 3(ENAB RFIFO);                        // Recv. Fifo and AACK Enabled.

}
```

configure_device_std_mode (dev_mode)

```
Set DCR bit 0(AACK);

Load RTR value(Threshold);

Set IER bit 2,7(ENAB INIT,ENAB RFULL);

elseif (dev_mode = 4)                                // Standard Compatible with
Set PCR bit 5(RCV DMA);                              // Recv. DMA Enabled.

Set IER bit 0,2(ENAB STRB,ENAB INIT);

elseif (dev_mode = 5)                                // Standard Compatible with
Set PCR bit 5(RCV DMA);                              // Recv. DMA and AACK Enabled.

Set DCR bit 0(AACK);

Set IER bit 2(ENAB INIT);

elseif (dev_mode = 6) // Standard Compatible with
Set PCR bit 3,5(ENAB RFIFO,RCV DMA);                // Recv. Fifo and Recv. DMA
Set RTR value(Threshold);                            // Enabled.

Set IER bit 0,2(ENAB STRB,ENAB INIT);

elseif (dev_mode = 7)                                // Standard Compatible with
Set PCR bit 3,5(ENAB RFIFO,RCV DMA);// Recv. Fifo , Recv. DMA
Set RTR value(Threshold);                            // and AACK Enabled.

Set IER bit 2(ENAB INIT);                            // Recommended mode of
Set DCR bit 0(AACK);                                // operation. }
```

check_interface_mode (dev_mode)

```
{

Read CVR Register;

if (dev_mode = 1,3,5,7) {                            // check mode selected.
compare value to 0x04(CVR bit 2 set);
else
compare value to 0x00(CVR bit 2 reset);
```

check_interface_mode (dev_mode)

```
        {  
  
    }  
  
}
```

Interrupt Handler()

```
    {  
  
    ** Interrupt sources are in order of priority.  
    ** Interrupt pin(s) are to be disabled when Handler is called  
    ** and enabled at return.  
    ** Interrupt sources are:  
        INIT IRQ =====> chi  
        STRB IRQ =====> chip  
        RFULL IRQ =====> chip  
        DMA TC =====> DMA Controller  
  
    Read IER Register;                // read chip interrupt mask.  
    Read ISR Register;                // read chip interruptsources.  
    if (Interrupt source != chip) {    // compare ISR ANDed with IER  
        Check_System_Status();        // with 0. If equal to 0,  
    }                                  // check for DMA or Timerintr.  
  
    goto interrupt source algorithm;
```

INIT IRQ

```
        Read DSR;  
  
    if (DSR bit 2 != 0) {              // low-to-high transition?  
        Reset ISR bit 2;              // This will lower Busy signal  
    } else                             {
```

INIT IRQ

```

                                Read DSR;

Reset ISR bit 2;                                // clear #init interrupt
initialize();                                  ****OPTIONAL ****
check_interface_reset();                      ****OPTIONAL ***
configure_device_std_mode (dev_mode);        ****OPTIONAL ****
check_interface_mode (dev_mode);              ****OPTIONAL ****
}
Return;
```

STRB IRQ

```

                                if (dev_mode != 0,2,4,6) {                                // check modes which enabled

call Error_Recovery_Routine();                // strobe interrupt.

}else {

Set DCR bit 7(+BUSY);                          // raise Busy signal

if (dev_mode = 0) {                            // Standard Compatible only;

Read PDR Register;                            // store_ptr is pointer to

store PDR at store_ptr;                       // buffer location available to

increment store_ptr;                          // store data.

}end if;

Reset DCR bit 6(-ACK);                        // lower -ACK signal.

Wait >= 500ns;                               // assume via software.

Set DCR bit 6(-ACK);                          // raise -ACK signal.

Reset ISR bit 1(STRB IRQ);                    // clear strobe interrupt.

if (Buffer_Full = TRUE) {                     // check buffer store count.

Buffer_Manager();                             // Pass full buffer to manage

if (Buffer_Avail = TRUE) {                   // and request for new Buffer.

store_ptr = new Base Pointer;                // store new ptr. to store_ptr.
```

STRB IRQ

```
        if (dev_mode != 0,2,4,6) {                                // check modes which enabled
            Reset DCR bit 7(+BUSY);                                // lower Busy signal.
        }                                                         // No buffer? Busy should be
    }else {                                                         // reset when buffer is avail.
        Reset DCR bit 7(+BUSY);                                    // lower busy signal.
    }end if;
}end if;
Return;
```

RFULL IRQ

```
        if (dev_mode != 2,3) {                                    // check modes which enabled
            call Error_Recovery_Routine();                          // Recv Fifo.
        }else {
            temp = Read RCR Register;                                // read byte count
            while (temp != 0) {                                     // if byte count not equal to 0,
                Read RDF ;                                          // read data out of Recv. Fifo
                store RDF at store_ptr;                             // and store at store_ptr.
                increment store_ptr;                                // Increment store_ptr and
                decrement temp;                                     // decrement byte count.
                if (Buffer_Full = TRUE) {                           // check buffer count.
                    Buffer_Manager();                               // Pass full buffer to manager.
                    if (Buffer_Avail = TRUE) {                     // check to see if buffer avail.
                        store_ptr = new Base Pointer;              // set new pointer to store_ptr.
                    }else {
                        Set DCR bit 7(+BUSY);                       // raise Busy signal.
                        Reset ISR bit 7(RFULL IRQ);                // clear Recv. Fifo full IRQ.
                        Return;                                     // Busy should be reset when a
```


RFULL IRQ

```
                if (dev_mode != 2,3) {                                // check modes which enabled
                    }end if;                                           // buffer becomes available.
                end if;
            }end while;
            Reset ISR bit 7(RFULL IRQ);                                // clear Recv. Fifo IRQ.
        }end if;
    Return;
```

DMA TERMINAL COUNT(TC) IRQ

```
                if (dev_mode != 4,5,6,7) {                            // check modes which enabled
                    call Error_Recovery_Routine();                    // Recv. DMA.
                }else {
                    Reset PCR bit 5(RECV DMA);                        // disable Recv. DMA.
                    Buffer_Manager();                                  // Pass full buffer to manager.
                    if (Buffer_Avail = TRUE) {                        // check for available buffer.
                        setup DMA CONTROLLER(start address,size = new Base Pointer,size;
                        Set PCR bit 5(RECV DMA);                      // enable Recv. DMA.
                    }else {
                        Set DCR bit 7(+BUSY);                          // raise Busy signal.
                        Return;                                         // Recv DMA should be set when
                    }end if;                                           // a buffer becomes available.
                }end if;
            Return;
```

6.8 IEEE Std. 1284 Pseudo-Code

This pseudocode can be used as an aid to develop a program to support one or all of the possible chip configurations(DMA,FIFO, Auto-Acknowledge) for the IEEE Std. 1284

Interface mode. The standard Compatible interface mode is also supported in this psuedo-code.

The IEEE Std. 1284 Interface can be configured for maximum throughput by enabling DMA's, FIFO's and Auto-Ack bits in the chip. The device function(s) and interrupt routine below, should be used as an example for low level device control. A high level interface example is not shown since it may vary across different architectures. The order in which the device function(s) appear below, should be the order in which the upper level device driver (application) call these functions or similar functions to setup the chip for operation. Some of these functions are optional. Each device function and interrupt routine can be modified to support only 1 or all possible chip configuration. The device mode check in most functions should be used as the indicator as to whether the line(s) of code are needed for a particular mode of operation. A Buffer Manager example is also illustrated to show the flow of data in a system. The buffering scheme may be different in other systems.

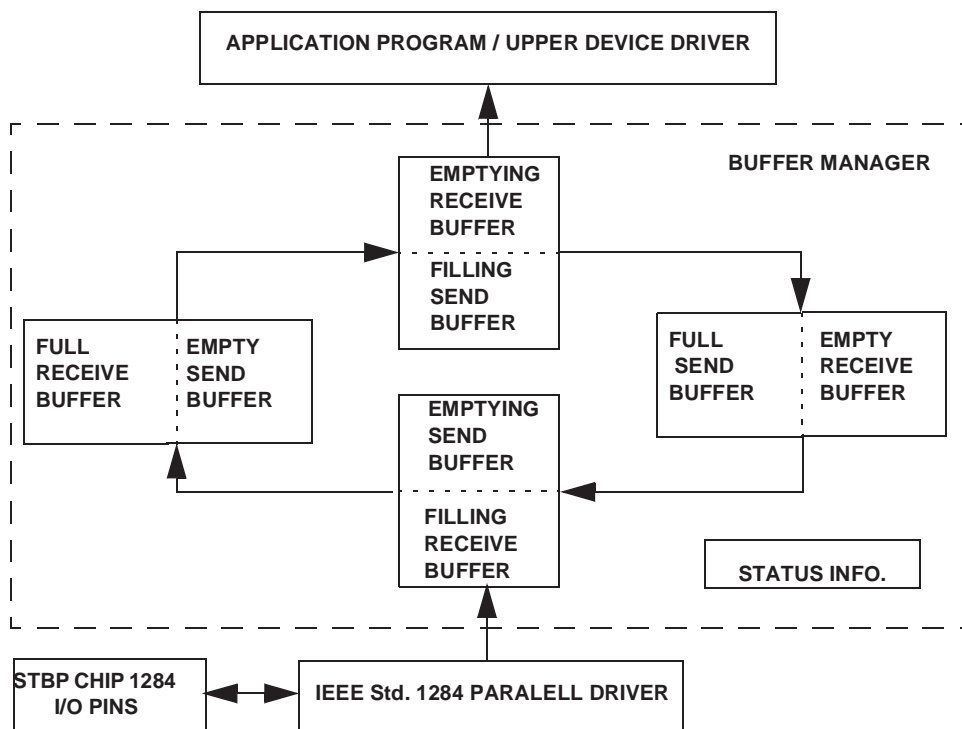


Figure 6-9. Buffer Manager (IEEE Std. 1284 Parallel Driver)

initialize()

```

{
    Set DCR bit 0(RESET);                // soft reset chip.
}

```

check_interface_reset()

```

{
    Read all Register(s) and compare to Default Values;
    if (Error = TRUE) {
        call Error_Recovery_Routine();
    }
}

```

configure_device_IEEE1284_mode (dev_mode)

```

{
    if (dev_mode = 0) {                // IEEE Std. 1284
        Set DCR bit 0(AACK);
        Set PCR bit 2(ENAB SFIFO);
        Set IER bit 0,2,3,6(ENAB STRB,ENAB INIT,ENAB SELIN,ENAB SEMPT);
    elseif (dev_mode = 1)              // IEEE Std. 1284 with RFIFO
        Set DCR bit 0(AACK);           // Enabled.
        Set RTR value(Threshold);
        Set PCR bit 2,3(ENAB SFIFO,ENAB RFIFO);
        Set IER bit 0,2,3,6,7(ENAB STRB,ENAB INIT,ENAB SELIN,ENAB SEMPT);
        Set IER bit 7(ENAB RFULL);
    elseif (dev_mode = 2)              // IEEE Std. 1284 Receive

```

configure_device_IEEE1284_mode (dev_mode)

```
{  
    Set DCR bit 0(AACK);                // DMA Enabled.  
    Set PCR bit 2,5(ENAB SFIFO,ENAB RCV DMA)  
    Set IER bit 0,2,3,6(ENAB STRB,ENAB INIT,ENAB SELIN,ENAB SEMPT);  
    elsif (dev_mode = 3)                // IEEE Std. 1284 with RFIFO  
    Set DCR bit 0(AACK);                // and Receive DMA Enabled.  
    Set RTR value(Threshold);  
    Set PCR bit 2,3,5(ENAB SFIFO,ENAB RFIFO,ENAB RCV DMA);  
    Set IER bit 0,2,3,6(ENAB STRB,ENAB INIT,ENAB SELIN,ENAB SEMPT);  
    elsif (dev_mode = 4)                // IEEE Std. 1284 with Send  
    Set DCR bit 0(AACK)                 // DMA Enabled.  
    Set IER bit 0,2,3(ENAB STRB,ENAB INIT,ENAB SELIN);  
    elsif (dev_mode = 5)                // IEEE Std. 1284 with Send  
    Set DCR bit 0(AACK);                // DMA and RFIFO Enabled.  
    Set IER bit 0,2,3,7(ENAB STRB,ENAB INIT,ENAB SELIN,ENAB RFULL);  
    elsif (dev_mode = 6)                // IEEE Std. 1284 with Send DMA  
    Set DCR bit 0(AACK);                // Recv. DMA Enabled.  
    Set IER bit 0,2,3(ENAB STRB,ENAB INIT,ENAB SELIN);  
    elsif (dev_mode = 7)                // IEEE Std. 1284 with RFFIO,  
    Set DCR bit 0(AACK);                // Recv. DMA & Snd. DMA Enabled.  
    Set IER bit 0,2,3(ENAB STRB,ENAB INIT,ENAB SELIN);  
}
```

Interrupt Handler()

```
{  
  
    ** Interrupt sources are in order of priority.  
  
    ** Interrupt pin(s) are to be disabled when Handler is called
```

Interrupt Handler()

```

{

** and enabled at return.

** Interrupt sources are:

    INIT IRQ =====> chip
    STRB IRQ =====> chip
    RFULL IRQ =====> chip
    SEMPT IRQ =====> chip
    SELIN IRQ =====> chip
    DMA TC =====> DMA Controller

Read IER Register;                // read chip interrupt mask.
Read ISR Register;                // read chip interrupt sources.
if (Interrupt source != chip) {   // compare ISR ANDed with IER
    Check_System_Status();        // with 0. If equal to 0,
}                                  // check for DMA or Timer intr.

goto interrupt source algorithm;
```

INIT IRQ

```

    Read CVR;

if (CVR = 4) {
    Read DSR;
    if (DSR bit 2 != 0) {         // low-to-high transition?
        Reset ISR bit 2;         // This will lower Busy signal
    } else {

        Reset ISR bit 2;         // clear #init interrupt

        initialize();            ****OPTIONAL ****
        check_interface_reset();  ****OPTIONAL ****
    }
}
```

INIT IRQ

```
                Read CVR;

configure_device_std_mode (dev_mode);          ****OPTIONAL ****
check_interface_mode (dev_mode);               ****OPTIONAL ****
}
}

Return;
```

STRB IRQ

```
Read CVR Register;                                // check mode
if (CVR = 0x40) {                                // Negotiation Mode?
    Reset ISR bit 0(STRB IRQ);                   // reset strobe interrupt.
    Read PDR Register;                           // get ext. link byte
    if ( PDR = 0x10,0x14) {                      // check to see if ECP mode
        ECP_MODE = TRUE;                        // set ECP_MODE flag.
        Setup DCR for Event 5                   // set DCR to ECP Event 5.
        wait for >= 500ns < 35ms               // Wait tp time.
        Setup DCR for Event 6                   // set DCR for ECP Event 6.
        Read CVR Register;                      // chip in ECP mode ?
        if (CVR != 0x20) {                      // ECP ?
            call Error_Routine();               // Call Error Routine
        }                                       // reset when buffer is avail.
    }
}elseif( PDR = 0,4 ) {
    NIBBLE_MODE = TRUE;                        // set NIBBLE_MODE flag.
    Read SCR Register;                          // get SFIFO count.
    if (SCR != 0) {                             // FIFO not Empty
        if (PDR = 4) {                         // DEVICE ID NIBBLE TRANSFER?
            Set DCR bit 4(SELECT);              // Set Xflag = 1.
```

STRB IRQ

```
}else {                                     // otherwise

    Reset DCR bit 4(SELECT);                 // Set Xflag = 0.
}

Setup DCR; Event 5 w/PE,nERROR=1           // set DCR to NIBBLE Event 5.
wait for >= 500ns < 35ms                   // Wait tp time.

Setup DCR; Event 6 w/PE,nERROR=1           // set DCR for NIBBLE Event 6.

Read CVR Register;                         // chip in NIBBLE mode ?

if (CVR != 0x20) {                          // NIBBLE ?
    call Error_Routine();                   // Call Error Routine
}                                           // reset when buffer is avail.

}else{

    if (PDR = 4) {                          // DEVICE ID NIBBLE TRANSFER?
        Set DCR bit 4(SELECT);             // Set Xflag = 1.
    }else {                                 // otherwise
        Reset DCR bit 4(SELECT);           // Set Xflag = 0.
    }

    Setup DCR; Event 5 w/PE,nERROR=0       // set DCR to NIBBLE Event 5.
    wait for >= 500ns < 35ms               // Wait tp time.

    Setup DCR; Event 6 w/PE,nERROR=0       // set DCR for NIBBLE Event 6.

    Read CVR Register;                     // chip in NIBBLE mode ?

    if (CVR != 0x20) {                     // NIBBLE ?
        call Error_Routine();              // Call Error Routine
    }                                       // reset when buffer is avail.

}elseif( PDR = 1,5 ){

    BYTE_MODE = TRUE;                      // set BYTE_MODE flag.

    Read SCR Register;                     // get SFIFO count.

    if (SCR != 0) {
```

STRB IRQ

```
Setup DCR; Event 5 w/PE,nERROR=1           // set DCR to BYTE Event 5.
wait for >= 500ns < 35ms                   // Wait tp time.
Setup DCR; Event 6 w/PE,nERROR=1           // set DCR for BYTE Event 6.
Read CVR Register;                          // chip in BYTE mode ?
if (CVR != 0x20) {                          // BYTE ?
    call Error_Routine();                   // Call Error Routine
}                                           // reset when buffer is avail.
}else{
Setup DCR; Event 5 w/PE,nERROR=0           // set DCR to BYTE Event 5.
wait for >= 500ns < 35ms                   // Wait tp time.
Setup DCR; Event 6 w/PE,nERROR=0           // set DCR for BYTE Event 6.
Read CVR Register;                          // chip in BYTE mode ?
if (CVR != 0x20) {                          // BYTE ?
    call Error_Routine();                   // Call Error Routine
}                                           // reset when buffer is avail.
}else {
NO_SUPPORT_MODE=TRUE;                      // set NO_SUPPORT_MODE flag
Reset DCR bit 4(SELECT);                   // Set Xflag = 0.
Setup DCR; Event 5 w/PE,nERROR=1           // set DCR to BYTE Event 5.
wait for >= 500ns < 35ms                   // Wait tp time.
Setup DCR; Event 6 w/PE,nERROR=1           // set DCR for BYTE Event 6.
if (CVR != 0x40) {                          // Negotiation?
    call Error_Routine();                   // Call Error Routine
}                                           // reset when buffer is avail.
}end if;
}elseif ( CVR =4 ){
Reset ISR bit 1(STRB IRQ);                 // clear strobe interrupt.
```


STRB IRQ

```
Set DCR bit 7(+BUSY);           // raise Busy signal
Read PDR Register;              // store_ptr is pointer to
store PDR at store_ptr;         // buffer location available to
increment store_ptr;            // store data.
if (Buffer_Full = TRUE) {       // check buffer store count.
    Buffer_Manager();           // Pass full buffer to manager
    if (Buffer_Avail = TRUE) {  // and request for new Buffer.
        store_ptr = new Base Pointer; // store new ptr. to store_ptr.
        Reset DCR bit 7(+BUSY);  // lower Busy signal.
    }                           // No buffer? Busy should be
}                                // reset when buffer is avail.
} else {                        // lower busy signal.
    Reset DCR bit 7(+BUSY);
}
} end if;
} else {
    Reset ISR bit 1(STRB IRQ);  // clear strobe interrupt.
    call Error_Routine();       // call Error Routine.
}
Return;
```

SELIN IRQ

```
Read CVR Register;              // check mode.
if ( CVR = 0x40 ) {             // Negotiation Mode?
    Reset ISR bit 3(SELIN IRQ); // reset SELIN Interrupt.
    Reset IER bit 2(ENAB INIT); // reset INIT Enable.
    IEEE_MODE = TRUE;          // Set IEEE_MODE flag.
    Setup DCR for Event 2      // Setup for Negot. Event 2
} else {                        // Termination?
```

SELIN IRQ

Read CVR Register;	// check mode.
Reset ISR bit 3(SELIN IRQ);	// reset SELIN Interrupt.
Read DSR Register;	// Read DSR for SELIN State. .
if (SELECTIN=0 & IEEE_MODE=TRUE){	// high-to-low transition.
if (NIBBLE_MODE = TRUE){	// NIBBLE MODE previous.
Perform NIBBLE TERMINATION PHASES(23-29)	
NIBBLE_MODE=FALSE;	// Reset NIBBLE_MODE flag.
IEEE_MODE=FALSE;	// Reset IEEE_MODE flag.
Set IER bit 2(ENAB INIT)	// Enable INIT interrupts.
}elseif (BYTE_MODE = TRUE){	// BYTE MODE previous.
Perform NIBBLE TERMINATION PHASES(22-29)	
BYTE_MODE=FALSE;	// Reset BYTE_MODE flag.
IEEE_MODE=FALSE;	// Reset IEEE_MODE flag.
Set IER bit 2(ENAB INIT)	// Enable INIT interrupts.
}elseif (ECP_MODE = TRUE){	// ECP MODE previous.
Perform NIBBLE TERMINATION PHASES(22-29)	
ECP_MODE=FALSE;	// Reset ECP_MODE flag.
IEEE_MODE=FALSE;	// Reset IEEE_MODE flag.
Set IER bit 2(ENAB INIT)	// Enable INIT interrupts.
}elseif(NO_SUPPORT_MODE=TRUE){	// NOT SUPPORTED MODE previous.
Perform NIBBLE TERMINATION PHASES(22-29)	
NO_SUPPORT_MODE=FALSE;	// Reset NO_SUPPORT_MODE flag.
IEEE_MODE=FALSE;	// Reset IEEE_MODE flag.
Set IER bit 2(ENAB INIT)	// Enable INIT interrupts.
}else{	
call Error_Routine();	
}end if;	

SELIN IRQ

```
                Read CVR Register;                // check mode.

    }else {

        call Error_Routine();

    }end if;

}end if;

Return;
```

RFULL IRQ

```
                if (dev_mode != 1,5) {                // check modes which enabled

    call Error_Recovery_Routine();                // Recv Fifo.

    }else {

        temp = Read RCR Register;                // read byte count

        while (temp != 0) {                // if byte count not equal to 0,

            Read RDF ;                // read data out of Recv. Fifo

            store RDF at store_ptr;                // and store at store_ptr.

            increment store_ptr;                // Increment store_ptr and

            decrement temp;                // decrement byte count.

            if (Recv_Buffer_Full = TRUE) {                // check buffer count.

                Buffer_Manager();                // Pass full buffer to manager.

                if (Recv_Buffer_Avail = TRUE) {                // check to see if buffer avail.

                    store_ptr = new Base Pointer;                // set new pointer to store_ptr.

                }else {

                    Set DCR bit 7(+BUSY);                // raise Busy signal.

                    Reset ISR bit 7(RFULL IRQ);                // clear Recv. Fifo full IRQ.

                    Return;                // Busy should be reset when a

                }end if;                // buffer becomes available.

            }end if;

        }end if;
```

RFULL IRQ

```
        if (dev_mode != 1,5) {                                // check modes which enabled

    }end while;

    Reset ISR bit 7(RFULL IRQ);                                // clear Recv. Fifo IRQ.

}end if;

Return;
```

SEMPY IRQ

```
        Reset ISR bit 4(SEMPY IRQ);                            // reset SEMPY IRQ.

    if (dev_mode != 0,1,2,3) {                                // check modes which enabled

        call Error_Recovery_Routine();                        // Recv Fifo.

    }else {

        if (Send_Buffer != EMPTY)                            // check buffer status.

            temp=Read SCR Register;                            // get SCR count;

        if(temp = 0){                                         // check to see if empty.

            while (temp != 64) {                               // byte count not equal to 64,

                Xmit_ptr data in SEND FIFO;                   // put data in Send FIFO.

                increment Xmit_ptr;                            // increment Send_Buffer ptr.

                increment temp;                                // increment send count.

                if (Send_Buffer = EMPTY) {                    // check buffer count.

                    Buffer_Manager();                          // Request another Buffer.

                    if (Buffer_Avail = TRUE) {

                        Xmit_ptr = new Base Pointer;

                    }else {

                        Return;

                    }end if;

                }end if;

            }end while;
```

SEMPY IRQ

```
Reset ISR bit 4(SEMPY IRQ);           // reset SEMPY IRQ.

}else {

    call Error_Routine();

}

}

}end if;                               // If Send_Buffer is EMPTY, upper level will
                                        start xfer.

Return;
```

DMA TERMINAL COUNT(TC) IRQ

```
// Upper level will control the setting of DMA Controller when Buffer

// is available.

// Remember: Only one control signal from chip to control both channels.

if (RECV_DMA = TRUE) {                 // check for RECV DMA?

    Reset PCR bit 5(RECV DMA);         // disable Recv. DMA.

    Buffer_Manager();                   // Pass full buffer to manager.

}elseif (SEND_DMA = TRUE) {

    Reset PCR bit 4(SEND DMA);         // disable Send DMA..

    Buffer_Manager();                   // Request another Buffer.

}else {

    call Error_Routine();

}end if;                               // a buffer becomes available.

Return;
```


Chapter 7. General Purpose Input/Output Controller

The General Purpose Input/Output (GPIO) controller enables multiplexing of module I/Os with many different functions. It can significantly reduce the quantity of module I/Os required.

7.1 GPIO Logic

The GPIO macro provides 24 bidirectional latched signals that are connected to the interrupt controller on the STBP chip. A 24 bit register sets the direction of each pin independently. A second 24 bit register contains the data value for each pin.

7.2 Operation

When a direction bit is set to 1, the GPIO data register is driven out to the chip GPIO pin. When a direction bit is set to 0, the GPIO data register reflects the external level applied to the associated GPIO pin. The actual implementation has the register read function connected directly to the receivers of the bidirectional GPIO pins. If a GPIO in output mode is shorted to a level, the register read function reflects the shorted levels, not the programmed level.

All GPIO pins are implemented in the same way. GPIO pins are not assigned functional names on the STBP chip.

All GPIO feed the interrupt logic described in "Interrupt Programming," on p. 1-6. GPIO pins, programmed as outputs, can also cause interrupts. The microcode programmer must use the interrupt mask bit in the interrupt logic to prevent GPIO outputs from causing unwanted system interrupts.

At STBP chip reset, all direction bits are set to 0 so that all GPIO pins are in receive mode.

7.2.1 Addressing

Address bits 0:3 address the macros of the STBP chip. Bit 0 is msb. Address bits 4:7 are used internally by each macro.

:

Table 7-1 GPIO Address

Address 4:7	Description
0000	DataBit 0:7

Table 7-1 GPIO Address

0001	DataBit 8:15
0010	DataBit 16:23
0011	Reserved
0100	DirectionBit 0:7
0101	DirectionBit 8:15
0110	DirectionBit 16:23
0111	Reserved

7.2.2 Functional Equations

Note: gpio0 = DirectionBit0 and DataBit0 (driving outbound mode)
DataBit0 = gpio0 (receiving inbound mode)

Chapter 8. Pulse Width Modulation

The pulse width modulation (PWM) function produces three square wave outputs with a period of $\text{SYSCLK}/1024$. The duty cycle of this square wave is variable under program control. The active time of this signal ranges from $\text{SYSCLK}/4$ to $\text{SYSCLK}/1020$ in increments of $\text{SYSCLK}/4$.

There is a control register with two bits for each PWM. This register will control if the PWM is active or not, and what its inactive output level should be. When the PWM control register is set to disable a PWM, the 8-bit period counter will be inactive to minimize power. The clock will be active, but flops will not be switching.

8.1 Operation

The pulse width modulation portion of this unit contains three identical blocks, each containing an 8-Bit programmable and reloadable down counter and control logic. At each $\text{SYSCLOCK}/1024$ pulse from the timebase generator, the control logic loads the PWM period register value into the counter and starts the counter counting down. The counter counts down each time it receives a $\text{SYSCLOCK}/4$ pulse from the timebase generator. The output of the PWM is set to a 1 at this time. When the PWM period counter reaches 0, the output of the PWM is set to a 0. It remains a 0 until the next $\text{SYSCLOCK}/1024$ pulse which starts the process all over again. SYSCLOCK is 27Mhz for the STBP chip.

After the PWM is enabled, and before the first $\text{SYSCLOCK}/1024$ pulse, the output of the PWM is set to 0.

The control logic contains two bits. The first bit indicates whether or not the output is permitted to switch as described above. If that bit is set to disable switching, the second bit is used to determine the output value which the signal is held at.

Figure 8-1 shows the output waveform for one cycle of a single PWM unit.

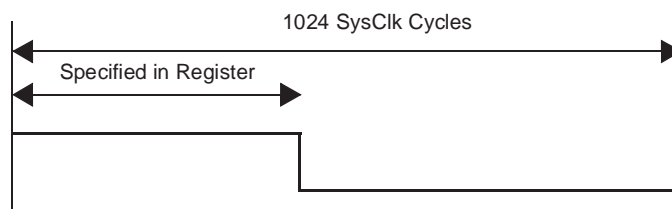


Figure 8-1. Output Waveform for PWM Unit

8.1.1 IC Timers

Three InterCharacter (IC) time-out timers are also implemented in this functional unit. These timers receive the count signal from the macro which they are timing. (See the UART, Chapter 5, and IEEE1284, Chapter 6, sections for descriptions of how this signal is generated in those macros.) Each timer is an 8 bit count-down timer. The timer is originally programmed by microcode. When the timer reaches 0, it has expired.

When a timer expires, it sets its corresponding bit in the IC interrupt status register below. These three bits are ORed together and sent to the Interrupt logic where they set bit 24 of interrupt status. Note that bit 24 of Interrupt status should be programmed to be level sensitive. To reset an IC time-out, a one should be written to the IC interrupt status register.

When a timer expires and causes an interrupt, the timer remains at zero and doesn't start counting again until the interrupt has been reset.

The UART IC timers use a timebase of 27Mhz/262,144 which is 103hz or a period of 9.7ms. This gives a timing range of 9.7ms to 2.47sec in steps of 9.7ms. 262144 is 2^{18} . The IEEE 1284 IC timer uses a timebase of 27Mhz/32,768 which is 823hz or a period of 1.2ms. This gives a timing range of 1.2ms to 301ms in steps of 1.2ms. 32768 is 2^{15} .

If a value of 0 is programmed into the time-out value, the timer is disabled and no interrupts are generated.

8.1.2 Timebase Generator

There is a single string of 18 divide-by-two circuits that generate the 27Mhz/4, 27Mhz/1024, 27Mhz/32,768, and 27Mhz/262144 time base values which are used by the PWM and the IC timers.

The output pulses of the timebase generator are all in-phase with each other.

8.2 Programming Information

Table 8-1. PWM and Timers Register Addressing

Address 4:7	Definition
0000	Pwm 0 period register
0001	Pwm 1 period register
0010	Pwm 2 period register
0011	Pwm control register
0100	Uart 1 IC time-out value
0101	Uart 2 IC time-out value
0110	IEEE1284 parallel port IC time-out value

Table 8-1. PWM and Timers Register Addressing

0111	IC interrupt status
------	---------------------

Table 8-2. PWM Control Registers

Bit	Definition
2	pwm 0 enable, 1 = enabled, 0 = disabled
3	pwm 0 output value when pwm disabled.
4	pwm 1 enable, 1 = enabled, 0 = disabled
5	pwm 1 output value when pwm disabled.
6	pwm 2 enable, 1 = enabled, 0 = disabled
7	pwm 2 output value when pwm disabled.

Table 8-3. IC Interrupt Status Register

Bit	Definition
5	Uart 1 timer expired
6	Uart 2 timer expired
7	IEEE1284 parallel port timer expired

8.2.1 Resetting:

All registers in this section reset to the value 0 except for the 3 PWM “disable value” bits.

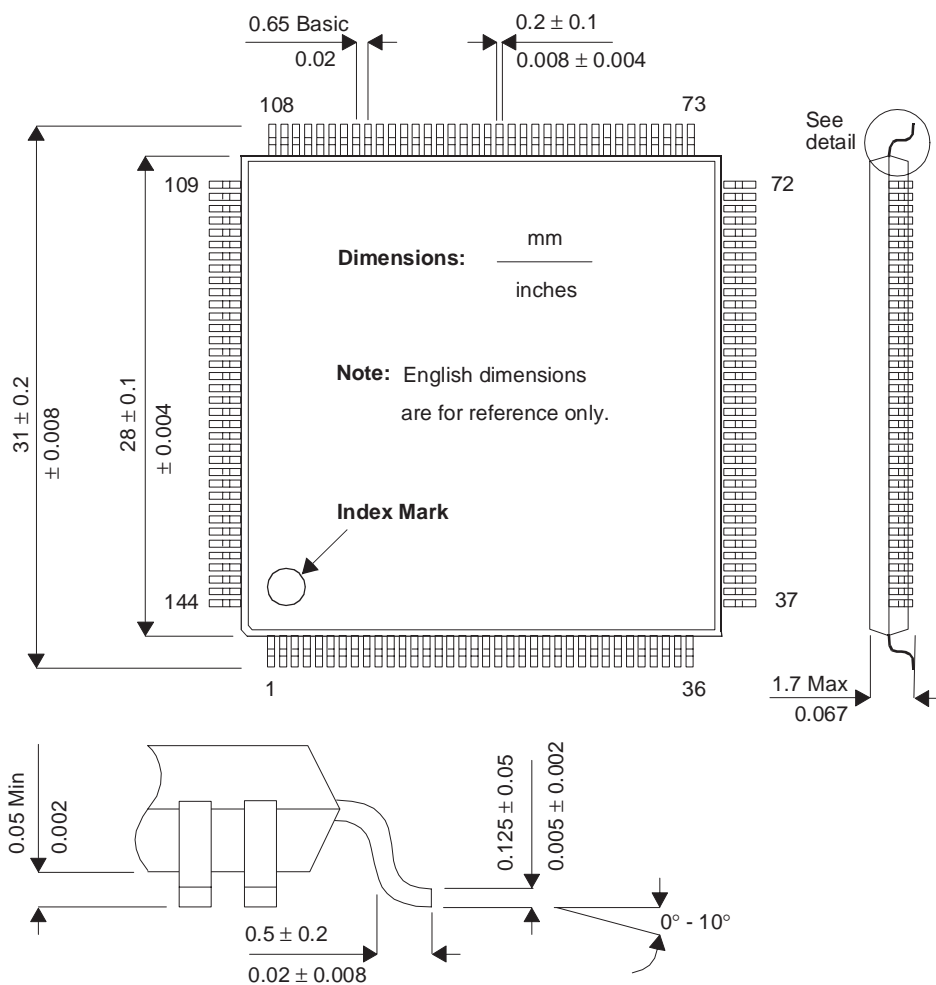


Figure A-1. STBP Mechanical Drawing (Top View)

A.1 Pin Functional Descriptions

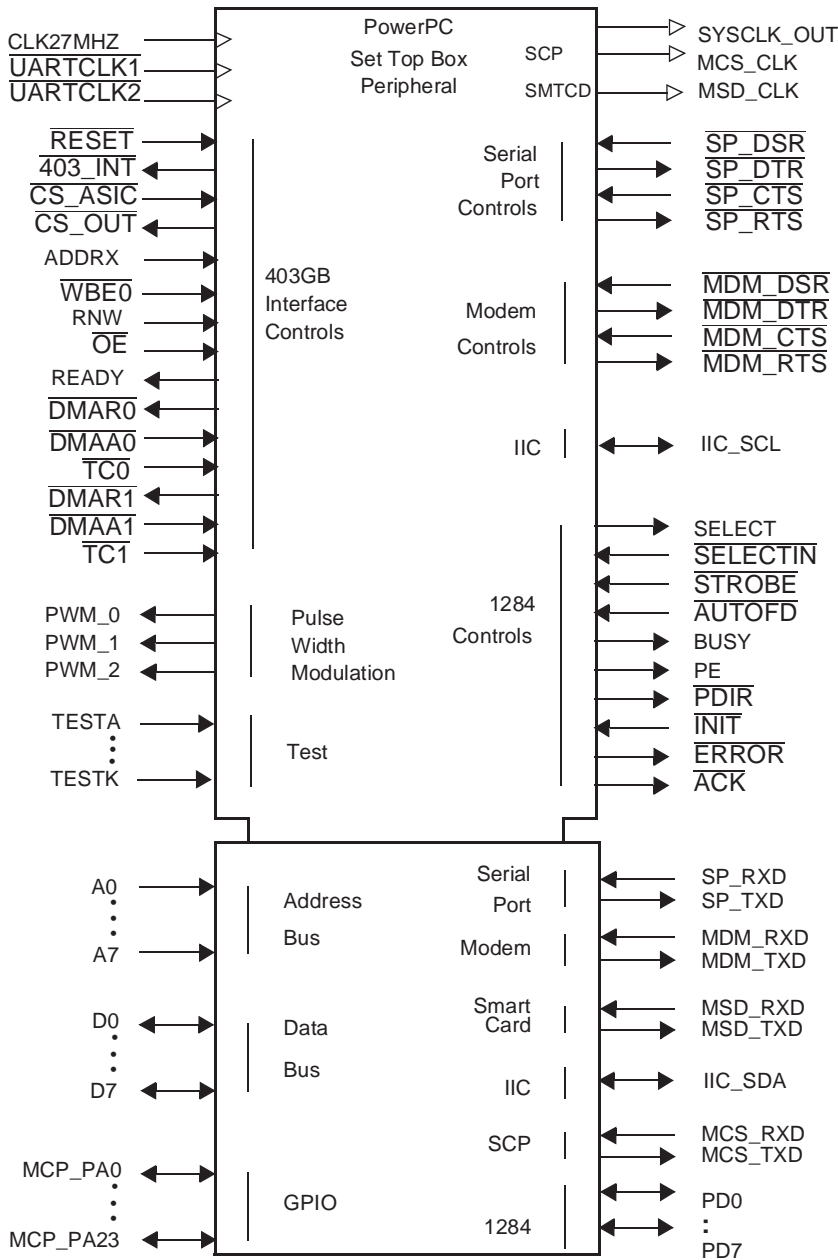


Figure A-2. STBP Logic Symbol

Active-low signals are shown with overbars: $\overline{\text{DMAR0}}$.

Table A-1. Signal Descriptions

Signal Name	Pin	I/O Type	Function												
$\overline{\text{ACK}}$	79	O	IEEE 1284 Peripheral Device.Acknowledge (Tx type - TTL4ma)												
ADDR0	110	I	403 Interface.Address Bus Bit 0. (RX type - TTL) Address bits 0:3 address the macros of STBP . Bit 0 is Most significant. Bit 3 is least significant. Address bits 4:7 are used internally by each macro. Table 2: <table><tr><th>Address0:3</th><th>Macro</th></tr><tr><td>0000</td><td>GPIO</td></tr><tr><td>0001</td><td>UART 1</td></tr><tr><td>0010</td><td>UART 2</td></tr><tr><td>0011</td><td>Smart card interface</td></tr><tr><td>0100</td><td>IIC interface</td></tr></table>	Address0:3	Macro	0000	GPIO	0001	UART 1	0010	UART 2	0011	Smart card interface	0100	IIC interface
Address0:3	Macro														
0000	GPIO														
0001	UART 1														
0010	UART 2														
0011	Smart card interface														
0100	IIC interface														
ADDR1	111	I	403 Interface.Address Bus Bit1. See description of ADDR0.												
ADDR2	112	I	403 Interface. Address Bus Bit 2. See description of ADDR0.												
ADDR3	113	I	403 Interface. Address Bus Bit 3. See description of ADDR0.												
ADDR4	114	I	403 Interface. Address Bus Bit 4. See description of ADDR0.												
ADDR5	115	I	403 Interface. Address Bus Bit 5. See description of ADDR0.												
ADDR6	116	I	403 Interface. Address Bus Bit 6. See description of ADDR0.												
ADDR7	117	I	403 Interface. Address Bus Bit 7. See description of ADDR0.												
ADDRX	125	I	403 Interface. (RX type - TTL)												
$\overline{\text{AUTO_FD}}$	107	I	IEEE 1284 Peripheral Device. (RX type - TTL)												
BUSY	83	O	IEEE 1284 Peripheral Device. (Tx type - TTL4ma)												

Table A-1. Signal Descriptions

Signal Name	Pin	I/O Type	Function				
CLK27M	87	I	403 Interface. 27 MHz Clock. (RX type - TTL)				
$\overline{\text{COMBOS}}$	11	I	Tied high for normal operation.				
$\overline{\text{CsAsic}}$	140	I	403 Interface. Chip Select. (RX type - TTL) Address logic truth table: Table 3:				
			$\overline{\text{CS}}_{\text{asic}}$	ADDRx	$\overline{\text{Int}}_{\text{CS}}$	$\overline{\text{CS}}_{\text{Out}}$	function
			0	0	1	0	External chip selected
			0	1	0	1	STBP selected
$\overline{\text{CS}}_{\text{OUT}}$	71	O	403 Interface. Chip Select Out. (Tx type - TTL4ma)				
Data0	130	I/O	403 Interface. Data bus bit 0 (Most significant bit). (RX type - TTL, Tx type - TTL4ma)				
Data1	131	I/O	Data bus bit 1. See description of Data0.				
Data2	132	I/O	Data bus bit 2. See description of Data0.				
Data3	133	I/O	Data bus bit 3. See description of Data0.				
Data4	136	I/O	Data bus bit 4. See description of Data0.				
Data5	137	I/O	Data bus bit 5. See description of Data0.				
Data6	138	I/O	Data bus bit 6. See description of Data0.				
Data7	139	I/O	Data bus bit 7. See description of Data0.				
$\overline{\text{DMAA0}}$	141	I	403 Interface. DMA Channel0 Acknowledge. $\overline{\text{DMAA0}}$ has an active level when a transaction is taking place between the 403 and a peripheral. (RX type - TTL)				
$\overline{\text{DMAA1}}$	142	I	403 Interface. DMA Channel 1 Acknowledge. See description of $\overline{\text{DMAA0}}$.				

Table A-1. Signal Descriptions

Signal Name	Pin	I/O Type	Function
$\overline{\text{DMAR0}}$	4	O	403 Interface. DMA Channel 0 Request. When the 403 is the bus master, external devices request a DMA transfer on channel 0 by putting a logic 0 on $\overline{\text{DMAR0}}$. (Tx type - TTL4ma)
$\overline{\text{DMAR1}}$	7	O	403 Interface. DMA Channel 1 Request. See description of $\overline{\text{DMAR0}}$. (Tx type - TTL4ma)
$\overline{\text{DTACK}}$	14	I	Tied high for normal operation.
$\overline{\text{ERROR}}$	80	O	IEEE 1284 Peripheral Device. (Tx type - TTL4ma)
GND	9		Ground. All ground pins must be used.
	19		Ground. All ground pins must be used.
	28		Ground. All ground pins must be used.
	36		Ground. All ground pins must be used.
	46		Ground. All ground pins must be used.
	55		Ground. All ground pins must be used.
	63		Ground. All ground pins must be used.
	72		Ground. All ground pins must be used.
	81		Ground. All ground pins must be used.
	91		Ground. All ground pins must be used.
	100		Ground. All ground pins must be used.
	108		Ground. All ground pins must be used.
	117		Ground. All ground pins must be used.
	127		Ground. All ground pins must be used.
	135		Ground. All ground pins must be used.
	144		Ground. All ground pins must be used.

Table A-1. Signal Descriptions

Signal Name	Pin	I/O Type	Function																		
IIC_SCL2	16	I/O	Phillips I ² C bus. Clock. See document iic_spec.ps for additional component information. Macro implemented with bit 0 as LSB. (RX type - TTL, TX type - OD 4ma)																		
IIC_SDA2	15	I/O	Phillips I ² C bus. Data. (RX type - TTL, TX type - OD 4ma)																		
INIT	89	I	IEEE 1284 Peripheral Device. (RX type - TTL)																		
LDS	12	O	Float. Do not connect.																		
MCP_PA0	64	I/O	<p>General Purpose Input Output Interface. Signal 0. (RX type - TTL, TX type - OD 4ma)</p> <p>The GPIO macro will provide 24 bidirectional latched signals which are connected to the interrupt controller on the chip. This macro will consist of a 24 bit register which will set the direction of each pin independently, along with a second 24 bit register which will contain the data value for each pin.</p> <p>GPIO addressing:</p> <table><tr><th>Address 4:7</th><th></th></tr><tr><td>0000</td><td>Data bit 0 to 7</td></tr><tr><td>0001</td><td>Data bit 8 to 15</td></tr><tr><td>0010</td><td>Data bit 16 to 23</td></tr><tr><td>0011</td><td>Reserved</td></tr><tr><td>0100</td><td>Direction Bit 0 to 7</td></tr><tr><td>0101</td><td>Direction Bit 8 to 15</td></tr><tr><td>0110</td><td>DirectionBit 16 to 23</td></tr><tr><td>0111</td><td>Reserved</td></tr></table>	Address 4:7		0000	Data bit 0 to 7	0001	Data bit 8 to 15	0010	Data bit 16 to 23	0011	Reserved	0100	Direction Bit 0 to 7	0101	Direction Bit 8 to 15	0110	DirectionBit 16 to 23	0111	Reserved
Address 4:7																					
0000	Data bit 0 to 7																				
0001	Data bit 8 to 15																				
0010	Data bit 16 to 23																				
0011	Reserved																				
0100	Direction Bit 0 to 7																				
0101	Direction Bit 8 to 15																				
0110	DirectionBit 16 to 23																				
0111	Reserved																				
MCP_PA1	61	I/O	General Purpose Input Output Interface. Signal 1. See description of MCP_PA0.																		
MCP_PA2	60	I/O	General Purpose Input Output Interface. Signal 2. See description of MCP_PA0.																		
MCP_PA3	59	I/O	General Purpose Input Output Interface. Signal 3. See description of MCP_PA0.																		

Table A-1. Signal Descriptions

Signal Name	Pin	I/O Type	Function
MCP_PA4	58	I/O	General Purpose Input Output Interface. Signal 4. See description of MCP_PA0.
MCP_PA5	57	I/O	General Purpose Input Output Interface. Signal 5. See description of MCP_PA0.
MCP_PA6	56	I/O	General Purpose Input Output Interface. Signal 6. See description of MCP_PA0.
MCP_PA7	53	I/O	General Purpose Input Output Interface. Signal 7. See description of MCP_PA0.
MCP_PA8	52	I/O	General Purpose Input Output Interface. Signal 8. See description of MCP_PA0.
MCP_PA9	51	I/O	General Purpose Input Output Interface. Signal 9. See description of MCP_PA0.
MCP_PA10	50	I/O	General Purpose Input Output Interface. Signal 10. See description of MCP_PA0.
MCP_PA11	49	I/O	General Purpose Input Output Interface. Signal 11. See description of MCP_PA0.
MCP_PA12	48	I/O	General Purpose Input Output Interface. Signal 12. See description of MCP_PA0.
MCP_PA13	45	I/O	General Purpose Input Output Interface. Signal 13. See description of MCP_PA0.
MCP_PA14	44	I/O	General Purpose Input Output Interface. Signal 14. See description of MCP_PA0.
MCP_PA15	43	I/O	General Purpose Input Output Interface. Signal 15. See description of MCP_PA0.
MCP_PA16	42	I/O	General Purpose Input Output Interface. Signal 16. See description of MCP_PA0.
MCP_PA17	41	I/O	General Purpose Input Output Interface. Signal 17. See description of MCP_PA0.
MCP_PA18	40	I/O	General Purpose Input Output Interface. Signal 18. See description of MCP_PA0.
MCP_PA19	39	I/O	General Purpose Input Output Interface. Signal 19. See description of MCP_PA0.
MCP_PA20	38	I/O	General Purpose Input Output Interface. Signal 20. See description of MCP_PA0.

Table A-1. Signal Descriptions

Signal Name	Pin	I/O Type	Function
MCP_PA21	32	I/O	General Purpose Input Output Interface. Signal 21. See description of MCP_PA0.
MCP_PA22	31	I/O	General Purpose Input Output Interface. Signal 22. See description of MCP_PA0.
MCP_PA23	30	I/O	General Purpose Input Output Interface. Signal 23. See description of MCP_PA0.
MCS_CLK	35	O	SCP Interface. Clock (TX type - OD 4ma)
MCS_RXD	34	I	SCP Interface. Receive Data. (RX type - TTL)
MCS_TXD	33	O	SCP Interface. Transmit Data (TX type - OD 4ma)
MDM_CTS	84	I	UART 2 (Modem). Clear To Send. (RX type - TTL) When low, indicates that the MODEM or data set is ready to exchange data. The CTS_N signal is a MODEM status input whose condition can be tested by the CPU reading bit 4 (CTS) of the MODEM Status Register. Bit 4 is the complement of the CTS_N signal. Bit 0 (DCTS) of the MODEM Status Register indicates whether the CTS_N input has changed state since the previous reading of the MODEM Status Register. CTS_N has no effect on the Transmitter. When CTS_N is de-activated, the uart will not transmit data.
MDM_DSR	94	I	UART 2 (Modem). Data Set Ready. (RX type - TTL) When low, indicates that the MODEM or data set is ready to establish a communications link with the UART. The DSR_N signal is a MODEM status input whose condition can be tested by the CPU reading bit 5(DSR) of the MODEM Status Register. Bit 5 is the complement of the DSR_N signal. Bit 1 (DDSR) of the MODEM Status Register indicates whether the DSR_N input has changed state since the previous reading of the MODEM
MDM_DTR	105	O	UART 2 (Modem). Data Terminal Ready.(TX type - TTL 4ma) When low, this informs the MODEM or data set that the UART is ready to establish a communications link. The DTR_N output signal can be set to an active low by programming bit 0 (DTR) of the MODEM Control Register to a high level. A Master Reset operation sets this signal to its inactive (high) state. Loop mode operation holds this signal in its inactive state.

Table A-1. Signal Descriptions

Signal Name	Pin	I/O Type	Function						
MDM_RTS	85	O	UART 2 (Modem). Request to Send. (TX type - TTL 4ma) When low, this informs the MODEM or data set that the UART is ready to exchange data. The RTS_N output signal can be set to an active low by programming bit 1 (RTS) of the MODEM Control Register. A Master Reset operation sets this signal to its inactive (high) state. Loop mode operation holds this signal in its inactive state. RTS_N shall be de-activated when DMA reaches Terminal Count, the DMA channel is unavailable, or when the receiver buffer is full.						
MDM_RXD	86	I	UART 2 (Modem). Receive Data. (RX type - TTL)						
MDM_TXD	77	O	UART 2 (Modem). Transmit Data. (TX type - TTL 4ma)						
MSD_CLK	92	O	Smartcard Interface.Clock. (TX type - TTL 4ma) <div>Table 4:<table><tr><th>clk sel 0,1</th><th>Clock used</th></tr><tr><td>00</td><td>sysclk/2</td></tr><tr><td>01</td><td>UartClk1</td></tr></table></div>	clk sel 0,1	Clock used	00	sysclk/2	01	UartClk1
clk sel 0,1	Clock used								
00	sysclk/2								
01	UartClk1								
MSD_RXD	21	I	Smartcard Interface. Receive Data. (RX type - TTL)						
MSD_TXD	20	O	Smartcard Interface.Transmit Data (TX type - OD 4ma)						
\overline{OE}	143	I	403 Interface. Output Enable. (RX type - TTL)						
\overline{Pdir}	74	O	IEEE 1284 Peripheral Device. (TX type - TTL 4ma)						
PD0	95	I/O	IEEE 1284 Peripheral Device. Bit 0 of 8 bit FIFO (RX type - TTL, TX type - TTL 4ma)						
PD1	96	I/O	IEEE 1284 Peripheral Device. Bit 1 of 8 bit FIFO (RX type - TTL, TX type - TTL 4ma)						
PD2	97	I/O	IEEE 1284 Peripheral Device. Bit 2 of 8 bit FIFO (RX type - TTL, TX type - TTL 4ma)						
PD3	98	I/O	IEEE 1284 Peripheral Device. Bit 3 of 8 bit FIFO (RX type - TTL, TX type - TTL 4ma)						

Table A-1. Signal Descriptions

Signal Name	Pin	I/O Type	Function
PD4	101	I/O	IEEE 1284 Peripheral Device. Bit 4 of 8 bit FIFO (RX type - TTL, TX type - TTL 4ma)
PD5	102	I/O	IEEE 1284 Peripheral Device. Bit 5 of 8 bit FIFO (RX type - TTL, TX type - TTL 4ma)
PD6	103	I/O	IEEE 1284 Peripheral Device. Bit 6 of 8 bit FIFO (RX type - TTL, TX type - TTL 4ma)
PD7	106	I/O	IEEE 1284 Peripheral Device. Bit 7 of 8 bit FIFO (RX type - TTL, TX type - TTL 4ma)
PE	93	O	IEEE 1284 Peripheral Device. (TX type - TTL 4ma)
PWM0	22	O	Pulse Width Modulation. (TX type - TTL 4ma)
PWM1	24	O	Pulse Width Modulation. (TX type - TTL 4ma)
PWM2	29	O	Pulse Width Modulation. (TX type - TTL 4ma)
READY	122	O	403 Interface. Ready. Enabled by CS_ASIC being active (low) and ADDR _x being high. (TX type - TTL 4ma TriS)
RNW	129	I	403 Interface. (RX type - TTL)
$\overline{\text{RST}}$	26	I	403 Interface. Reset. Negative active pulse at least 8 clock cycles long, and the system clock must be running during reset time. (RX type - TTL)
$\overline{\text{SelectIn}}$	75	I	IEEE 1284 Peripheral Device. (RX type - TTL)
SELECT	76	O	IEEE 1284 Peripheral Device. (TX type - TTL 4ma)
SP_CTS	66	I	UART 1 (Serial Port). Clear To Send. (RX type - TTL)
SP_DSR	67	I	UART 1 (Serial Port). Data Set Ready. (RX type - TTL)
SP_DTR	68	O	UART 1 (Serial Port). Data Terminal Ready. (TX type - TTL 4ma)
SP_RTS	65	O	UART 1 (Serial Port). Request to Send. (TX type - TTL 4ma)
SP_RXD	69	I	UART 1 (Serial Port). Receive Data. (RX type - TTL)
SP_TXD	70	O	UART 1 (Serial Port). Transmit Data. (TX type - TTL 4ma)
$\overline{\text{STROBE}}$	88	I	IEEE 1284 Peripheral Device. (RX type - TTL)

Table A-1. Signal Descriptions

Signal Name	Pin	I/O Type	Function
SysClkOut	78	O	System Clock Output. Feed the 403 system clock to achieve proper latching of data. (TX type - TTL 8ma)
$\overline{TC0}$	2	I	403 Interface. (RX type - TTL)
$\overline{TC1}$	3	I	403 Interface. (RX type - TTL)
TESTA	17	I/O	Reserved for manufacturing test. Tied low for normal operation.
TESTB	124	I	Reserved for manufacturing test. Tied low for normal operation.
TESTC	104	I	Reserved for manufacturing test. Tied low for normal operation.
TESTD	8	I	Reserved for manufacturing test. Tied low for normal operation.
TESTE	5	I	Reserved for manufacturing test. Tied low for normal operation.
TESTF	6	I	Reserved for manufacturing test. Tied low for normal operation.
TESTG	128	I	Reserved for manufacturing test. Tied low for normal operation.
UARTCLK1	25	I	(RX type - TTL)
UARTCLK2	23	I	(RX type - TTL)
UDS	13	O	Float. Do not connect.

Table A-1. Signal Descriptions

Signal Name	Pin	I/O Type	Function
V_{DD}	1		Power. All power pins must be connected to 3.3V supply.
	10		Power. All power pins must be connected to 3.3V supply.
	18		Power. All power pins must be connected to 3.3V supply.
	27		Power. All power pins must be connected to 3.3V supply.
	37		Power. All power pins must be connected to 3.3V supply.
	47		Power. All power pins must be connected to 3.3V supply.
	54		Power. All power pins must be connected to 3.3V supply.
	62		Power. All power pins must be connected to 3.3V supply.
	73		Power. All power pins must be connected to 3.3V supply.
	82		Power. All power pins must be connected to 3.3V supply.
	90		Power. All power pins must be connected to 3.3V supply.
	99		Power. All power pins must be connected to 3.3V supply.
	109		Power. All power pins must be connected to 3.3V supply.
	119		Power. All power pins must be connected to 3.3V supply.
	126		Power. All power pins must be connected to 3.3V supply.
	134		Power. All power pins must be connected to 3.3V supply.
<u>WBE0</u>	120	I	403 Interface. Write Byte Enable 0. (RX type - TTL)
WBE1	121	I	403 Interface. Write Byte Enable 1. (RX type - TTL)
<u>403_INT</u>	123	O	403 Interface. A common interrupt signal which combines interrupts from all macros. (TX type - OD 4ma)

Table A-2 Pin Number-Signal Name

Pin	Signal Names	Pin	Signal Names	Pin	Signal Names	Pin	Signal Names
1	V _{DD}	37	V _{DD}	73	V _{DD}	109	V _{DD}
2	TC0	38	MCP_PA20	74	Pdir	110	ADDR0
3	TC1	39	MCP_PA19	75	SelectIn	111	ADDR1
4	DMAR0	40	MCP_PA18	76	SELECT	112	ADDR2
5	TESTE	41	MCP_PA17	77	MDM_TXD	113	ADDR3
6	TESTF	42	MCP_PA16	78	SysClkOut	114	ADDR4
7	DMAR1	43	MCP_PA15	79	ACK	115	ADDR5
8	TESTD	44	MCP_PA14	80	ERROR	116	ADDR6
9	GND	45	MCP_PA13	81	GND	117	ADDR7
10	Vdd	46	GND	82	Vdd	118	GND
11	COMBOS	47	Vdd	83	BUSY	119	Vdd
12	LDS	48	MCP_PA12	84	MDM_CTS	120	WBE0
13	UDS	49	MCP_PA11	85	MDM_RTS	121	WBE1
14	DTACK	50	MCP_PA10	86	MDM_RXD	122	READY
15	IIC_SDA2	51	MCP_PA9	87	CLK27M	123	403_INT
16	IIC_SCL2	52	MCP_PA8	88	STROBE	124	TESTB
17	TESTA	53	MCP_PA7	89	INIT	125	ADDRX
18	V _{DD}	54	V _{DD}	90	V _{DD}	126	V _{DD}
19	GND	55	GND	91	GND	127	GND
20	MSD_TXD	56	MCP_PA6	92	MSD_CLK	128	TESTG
21	MSD_RXD	57	MCP_PA5	93	PE	129	RNW
22	PWM0	58	MCP_PA4	94	MDM_DSR	130	Data0
23	UARTCLK2	59	MCP_PA3	95	PD0	131	Data1
24	PWM1	60	MCP_PA2	96	PD1	132	Data2
25	UARTCLK1	61	MCP_PA1	97	PD2	133	Data3
26	RST	62	Vdd	98	PD3	134	Vdd
27	Vdd	63	GND	99	Vdd	135	GND
28	GND	64	MCP_PA0	100	GND	136	Data4
29	PWM2	65	SP_RTS	101	PD4	137	Data5
30	MCP_PA23	66	SP_CTS	102	PD5	138	Data6
31	MCP_PA22	67	SP_DSR	103	PD6	139	Data7
32	MCP_PA21	68	SP_DTR	104	TESTC	140	CsAsic
33	MCS_TXD	69	SP_RXD	105	MDM_DTR	141	DMAA0
34	MCS_RXD	70	SP_TXD	106	PD7	142	DMAA1
35	MCS_CLK	71	CS_OUT	107	AUTO_FD	143	OE
36	GND	72	GND	108	GND	144	GND

A.2 Package Thermal Specifications

The STBP is designed to operate within the case temperature range from 0°C to 120°C.

Note 1: Case temperature T_{m_C} is measured at top center of case surface with device soldered to circuit board.

Note 2: $T_{m_A} = T_{m_C} - P \times \theta_{CA}$, where T_{m_A} is ambient temperature.

Note 3: $T_{m_{CMax}} = T_{m_{JMax}} - P \times \theta_{JC}$, where $T_{m_{JMax}}$ is maximum junction temperature and P is power consumption.

Note 4: The above assumes that the chip is mounted on a card with at least one signal and two power planes.

A.3 Electrical Specifications

A.3.1 Absolute Maximum Ratings

The absolute maximum ratings in Table A are stress ratings only. Operation at or beyond these maximum ratings may cause permanent damage to the device.

Table A-3. STBP Maximum Ratings

Parameter	Maximum Rating
Supply voltage with respect to GND	-0.5V to +3.8V
Voltage on other pins with respect to GND	-0.5V to +5.5V
Case temperature under bias	0°C to +120°C
Storage temperature	0°C to +150°C

A.3.2 Operating Conditions

The STBP can interface to either 3V or 5V technologies. The range for supply voltages is specified for five-percent margins relative to a nominal 3.3V power supply.

Device operation beyond the conditions specified in tables in this appendix is not recommended. Extended operation beyond the recommended conditions may affect device reliability: TBD.

A.3.3 Recommended Connections

Power and ground pins should all be connected to separate power and ground planes in the circuit board to which the STBP is mounted. Unused input pins must be tied inactive, either high or low.

A.3.4 DC Specifications

Table A-4. STBP DC Characteristics

Symbol	Parameter	Min	Max	Units
V_{IL}	Input low voltage	GND - 0.1	0.8	V
V_{IH}	Input high voltage (except for SysClk) ¹	2.0	3.3 + 0.1	V
V_{OL}	Output low voltage		0.4	V
V_{OH}	Output high voltage	2.4	V_{DD}	V
I_{OH}	Output high current		2	mA
I_{OL}	Output low current		4	mA
I_{LI}	Input leakage current		50	mA
I_{LO}	Output leakage current		10	μ A

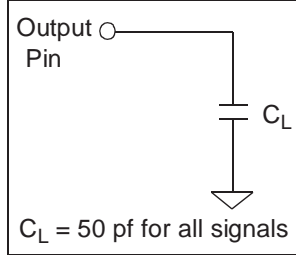
Table A-5. STBP I/O Capacitance

Symbol	Parameter	Min	Max	Units
C_{IN}	Input capacitance		5	pF
C_{OUT}	Output capacitance ¹		7	pF
$C_{I/O}$	I/O pin capacitance		8	pF

Note 1: C_{Out} is specified as the load capacitance of a floating output in high impedance.

A.4 AC Specifications

Clock timing and switching characteristics are specified in accordance with recommended operating conditions noted above. AC specifications are tested at $V_{DD} = 3.14V$ and $T_J = 85^{\circ}C$ with the 50pF test load shown in the figure below.



Misc. Chip Delays	min. ns	max. ns
SysClk to SysClkOut delay	4.0	11.5
Cs to CsOut delay	3.7	11.2
SysClk to Internal Clock	1.7	5.0

STBP Output Timing	ns
output data setup ¹ , ref SysClk	16.6
output data hold, ref SysClk	5.6
output data setup, ref SysClkOut	28.1
output data hold, ref SysClkOut	2.0

Note 1: This refers to the setup time provided by STBP to PPC403GB. It can be compared directly to the setup time required specification time in the PPC403GB manual.

STBP Input Requirements	ns
input data setup, ref SysClk	0.0
input data hold, ref SysClk	3.8
input data setup, ref SysClkOut	9.6
input data hold, ref SysClkOut	0.0

In order to achieve proper latching of data between STBP and the PPC403GB, it is necessary to feed the PPC403GB system clock input from the SysClkOut output of the STBP chip. This allows the PPC403GB clocks to be synchronized with the STBP clocks. The PPC403GB uses an internal PLL, which causes it's internal clocks, at the latch inputs, to rise at the same time as the clock input. STBP does not have a PLL, therefore there is a significant internal delay between the rise of it's clock input and the rise of a clock at an internal flop.

The SysClkOut signal from STBP is designed so that the rising edge of the clock will occur slightly before the change of data on the STBP outputs. This is achieved by using a higher performance driver for SysClkOut than the rest of the outputs, combined with the input to this driver being taken slightly ahead of the flops in the clock repowering tree. This will assure correct latching of STBP data by the PPC403GB.

Additionally, The SysClkOut signal will occur slightly after (due to Off-Chip-Driver delay) the internal Bclk signals, (which feed the flops) therefore assuring correct latching of PPC403GB data into STBP.

Appendix B. Errata

The 'Summary' presentation order is

1. Problem
2. Description
3. Impact
4. Workaround

The 'Solution' presentation order (same number reference) is

- Overview
- Detailed Description
- Projected Impact
- Workaround
- Projected Solution

Summary 1.

1. UART doesn't restart after clearing FIFO.
2. The UART logic does not assert a DMA request after a FIFO clearing operation.
3. Logic appears to hang.
4. Software workaround is available.

Summary 2.

1. Asynchronous GPIO interrupts..
2. Asynchronous GPIO inputs do not reliably generate interrupts.
3. Expected interrupt not presented to 403.
4. Software workaround is available.

Summary 3.

1. Parallel port doesn't start DMA in nibble mode.
2. When data is being sent from STBP to the PC, DMA mode doesn't work.
3. Parallel port doesn't start.
4. Software workaround is available.

Summary 4.

1. IIC repeated start detects false error.
2. Write 1 byte, Restart, Read 1 byte. This causes IIC logic to detect false error.
3. False error detected. Data OK.
4. Software workaround is available.

Summary 5.

1. Parallel port doesn't transfer bytes left in FIFO in DMA mode after direction switch.
2. If the direction of the interface in ECP mode is changed before the receive FIFO is emptied, not all of the received data will be transferred to the processor in DMA mode.
3. Data appears to be incomplete.
4. Software workaround is available.

Summary 6.

1. Parallel port receive FIFO not available for reading during negotiation mode.
2. If the parallel port is receiving data from the host, and the STPB software has not emptied the receive FIFO before negotiation mode is entered, then the software is not allowed to read the receiverb FIFO data.
3. Duplicate false data bytes received
4. Software workaround is available.

Summary 7.

1. Smart card interface detects error assertion as false start bit.
2. If the smart card interface is receiving data from the smar card, and it detects a parity error, it will assert the parity error condition. The interface sees the error condition as a false start bit.
3. False byte received.
4. Software workaround is available.

Solution 1.

Overview:

UART doesn't restart after clearing FIFO.

Detailed Description:

The UART logic does not assert TxRdy after a FIFO clearing operation. The DMA logic uses this signal to know when to request data from the processor. Writing a byte in MMIO mode to the TX FIFO allows the TxRdy signal to be asserted normally afterwards.

This FIFO clearing operation here refers to the UART command where the transmit buffer is cleared before all of the data has been sent. For example, if an error is detected, this command could be used to eliminate unsent data from the FIFO so a retransmission of clean data can be started.

Projected Impact:

The logic appears to hang.

Work Around:

After clearing FIFO, write the first byte of the next transfer in MMIO mode to the transmit holding register. Then finish the transfer in DMA mode. If the transfer is only one byte, just use the MMIO mode.

Projected Solution:

No fix is planned for the this design. The software workaround should be used with this design.

Solution 2.

Overview:

The interrupt logic handling GPIO inputs cannot reliably detect an a edge change when programmed to generate an interrupt on the rising or falling edge of the GPIO input signal.

Detailed Description:

The GPIO signals are not registered before being fed into interrupt logic. If the signal changes level within a few ns of the sysclock rising edge, the interrupt edge detector will not detect the edge change.. This happens approx. 10% of the time, and can be reliably repeated if the signal is made to transition near the clock.

Projected Impact:

Expected interrupts do not occur.

Work Around:

There are several workarounds available:

- Use level detection.

- Use interrupt pin on processor for interrupt function.

- Ensure that GPIO input signal is synchronous to the Spring system clock.

Projected Solution:

No fix is planned for the this design. The software workaround should be used with this design.

Solution 3.

Overview:

Parallel port doesn't start DMA in nibble mode.

Detailed Description:

When data is being sent from STBP to the PC in nibble mode, DMA mode doesn't work. The reason is that nibble mode is a reverse direction but the main data bus is in the forward direction. Because of the direction of the data bus, the internal direction signal reflects the forward direction. This is necessary to prevent the data bus from being driven by both ends at once. This internal direction signal is used to qualify the DMA request.

Projected Impact:

Parallel port doesn't start. No DMA request is presented to the processor.

Work Around:

Use MMIO mode instead of DMA mode when Nibble mode is desired. Due to the very low performance of nibble mode in the first place, this should not be a severe restriction.

Projected Solution:

No fix is planned for the this design. The software workaround should be used with this design.

Solution 4.

Overview:

IIC repeated start detects false error.

Detailed Description:

Write 1 byte, Restart, Read 1 byte. This causes IIC logic to detect false error. The incomplete transfer and transfer aborted bits are set in the extended status register. These are bits 1 and 0, respectively.

Projected Impact:

False error detected. Data OK.

Work Around:

Ignore false error.

Projected Solution:

No fix is planned for the this design. The software workaround should be used with this design.

Solution5.

Overview:

Parallel port doesn't transfer bytes left in FIFO in DMA mode after direction switch.

Detailed Description:

1. PC requests ECP mode.
2. PC send data in DMA mode.
3. As soon as BUSY line down (transfer complete from PC viewpoint), PC issues INIT=0 to change xfer direction.

The internal direction signal is used to gate the DMA request. If the direction changes before the data is read from the FIFO, the DMA request is dropped. Data is not lost, and is still in the receive FIFO awaiting transfer to the PC. However, the DMA request signal will not be asserted due to the direction change.

This problem is related to number 3 above.

Projected Impact:

Data received from the PC appears to be missing the last few bytes.

Work Around:

Enable direction switch interrupt to trigger 'clean-up routine' which will inspect 1284 RX FIFO for data and retrieve it via MMIO.

Another workaround is to set FIFO trigger level to 1.

Projected Solution:

No fix is planned for the this design. The software workaround should be used with this design.

Solution 6.

Overview:

Parallel port receive FIFO not available for reading during negotiation mode.

Detailed Description:

1. PC sending data to STBP
2. PC completes transfer and immediately enters negotiation mode.
3. STBP receive FIFO still contains data.

4. Because STBP makes the negotiation byte available at the same address as the receive data, the 1284 logic switches the read location to the negotiation byte when in negotiation mode. In negotiation mode, the receive FIFO data is not available for reading.

In MMIO mode, the software driver should not attempt to access the receive FIFO during negotiation mode. After negotiation mode is complete, the receive FIFO can be read normally.

In DMA mode, the DMA request signals are still active during negotiation mode, even though you can't access the FIFO. This causes many duplicate and false bytes to be read by the 403 DMA controller.

Projected Impact:

False extra bytes received.

Work Around:

Use MMIO mode with the 1284 and ensure that software driver implemented in such a way as to not access the receive FIFO during negotiation mode.

In DMA mode, the only way to ensure correct operation is for the PC software end to allow sufficient time for STBP to empty the receive FIFO before entering negotiation mode for another transfer.

Projected Solution:

No fix is planned for the this design. The software workaround should be used with this design.

Solution 7.

Overview:

Smart card interface detects error assertion as false start bit.

Detailed Description:

If the smart card interface is receiving data from the smart card, and it detects a parity error, it will assert the parity error condition. The interface sees the error condition as a false start bit. This false start bit causes the smart card interface to receive a false byte. This false byte is typically FF since that is the idle condition of the interface. If the parity is set to even, then another parity error (false error) is detected and the loop continues.

Projected Impact:

Smart card can enter loop where error assertion causes false byte to be received, which can cause another error assertion to be generated, which results in another false byte.

Work Around:

There are two workarounds available:

Disable parity error detection and assertion.

When software detects looping of error condition, temporarily disable error detection and issue reset to smart card. This allows for continued checking for parity errors but doesn't allow for automatic resending of bytes which contained parity error.

Projected Solution:

No fix is planned for the this design. The software workaround should be used with this design.

Appendix C. Acronyms and Abbreviations

ASIC	Application-Specific Integrated Circuit
ASSP	Application-Specific Solutions Products
ADC	Analog-to-digital conversion
ARM	Alternate refresh mode
ATM	Asynchronous transfer mode
ANSI	American National Standards Institute
BME	Burst mode enable
CSSP	Customer-Specific Solutions Products
CMP	Composite video
CMOS	Complimentary Metal Oxide Semiconductor
CVBS	Combined video blanking and sync
CCIR	International Radio Consultive Committee
CRC	Cyclic redundancy check
CPM	Clock and Power Management

CAS	Column address strobe
CIF	Common Interface Format
CIP	Common Isochronous Packet
CTS	Clear to send
CPU	Central Processing Unit
DENC	Digital video encoder
DAC	Digital to analog converter
DTR	Data transmit ready
DVB	Digital video broadcast
D/A	Digital to analog converter
DMSD	Digital multi-standard decoder
DSF	Designated special function
DVB	Digital Video Broadcasting System
DVI	Digital video interactive
DCR	Device control register
D-Cache	Data cache

DSP	Digital signal processor
DMA	Direct memory access
DTR	Data transmit ready
DTSC	Data Tri-state Control
DCE	Data Communications Equipment
DSR	Data set ready
DCT	Discrete Cosine Transform
DCD	Data carrier detect
DTO	Discrete time oscillator
DVD	Digital Versatile Disk
DAVIC	Digital Audio-Visual Council
EOT	End of transfer
ERM	Early RAS mode
EDO	Extended data output
EBIU	Extended bus interface unit
FIFO	First in, first out

FPM	Fast page mode
FIR	Finite Impulse Response
FIT	Fixed interval timer
FM	Frequency modulation
GPIO	General Purpose Input/Output
GUI	Graphical user interface
GPR	General purpose register
HSI	Hue, saturation, and Intensity
HSL	Hue, saturation, and Lightness
HSV	Hue, saturation, and Value
HSMC	High Speed Memory Controller
HSYNC	Horizontal sync
IFD	Interface device
ISO	International Standards Organization
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers

IBM	International Business Machines
I-Cache	Instruction cache
IrDA	Infrared Data Association
I/O	input/output
I²C	Inter-Integrated Circuit
I/F	Interface
IMD	IBM Microelectronics Division
JPEG	Joint Photographic Experts Group
JTAG	Joint Test Action Group
LSB	Least significant bit
LL	Link layer
LSSD	Level sensitive scan design
MPEG	Moving Pictures Experts Group
MMU	Memory management unit
MSB	Most significant bit
NIM	Network Interface Module

NIU	Network Interface Unit
NC	Network Computer
NTSC	National Television Systems Committee
NS	National Semiconductor
OPB	On-chip Peripheral Bus
OSD	On-screen display
OCM	On-chip memory
OE	Output enable
PLB	PowerPC Local Bus
PLL	Phase-lock loop
PIT	Programmable interval timer
PID	Packet Identifier
PCR	Program clock reference
PSI	Program-specific information
PES	Packetized elementary stream
PWM	Pulse width modulation

PAL	Phase alternate line
PPC	PowerPC
PIA	Peripheral interface adapter
PCM	Pulse Code Modulation
POF	Plastic optical fiber
RTS	Ready to send
RZ	Return to zero
RGB	Red, green, blue
ROM	Read-only memory
RE	Ready enable
RAS	Row address strobe
RISC	Reduced information set computer/cycles
RTOS	Real-time operating system
SNR	Signal-to-noise ratio
SIF	Standard (or source) input format
SAP	Secondary audio program

SBB	System Building Blocks
STB	Set Top Box
SOC	System-on-a-chip
SRAM	Static random access memory
SCI	SmartCard Interface
SDRAM	Synchronous dynamic random access memory
SICC	Serial and Infrared Communications Controller
SPR	Special purpose register
STC	System time clock
SI	Service information
S/PDIF	Sony/Philips Digital Interface Format
SGRAM	Synchronous graphics random access memory
SCC	Serial Communications Controller
SMPTE	Society of Motion Picture and Television Engineers
STU	Set Top Unit

SAR	Segmentation and reassembly
STP	Shielded twisted pair
S-Video	Separate video
TC	Terminal count
TBD	To be determined
TAP	Transport Assist Processor
TLB	Translation Lookaside Buffer
TS	Transport stream
UART	Universal Asynchronous Receiver/Transmitter
UIC	Universal Interrupt Controller
UTP	Unshielded twisted pair
VBI	Vertical blanking interval
VESA	Video Electronic Standards Association
VCO	Voltage-controlled oscillator
VCXO	Voltage-controlled crystal oscillator
VITC	Vertical interval time code

VMI	Video Module Interface
VPS	Video Program System
VSYNC	Vertical sync
WDT	Watchdog timer
WSS	Wide-screen signalling
WE	Write enable
YUV	A color image encoding scheme that separates luminance (Y) and two color signals: red minus Y (U) and blue minus Y (V).

Index

Numerics

1284 **6-1**

- byte mode **6-4**
- compatibility mode **6-2**
- compatible pseudo-code **6-29**
- ecp mode **6-5**
- i/o signals **6-1**
- intercharacter timer **6-2**
- interface signals **6-9**
- nibble mode **6-3**
- operating modes **6-2**
- register definitions **6-8**
- register descriptions **6-14**
- signal transition event **6-12**
- std pseudo-code **6-35**

23049

- Chapter
- Chapter 2. **2-1**

B

book **xiii**

- how to use this book **xiii**
- notation conventions **xiv**
- who should use this book **xiii**

C

clocks **1-5**

G

gpio

- address **7-1**
- operation **7-1**

I

iic

- programming interface **2-2**
- Inter-Integrated Circuit **2-1**
- inter-integrated circuit **2-1**
- interrupt
- addressing **1-6**

- implementation **1-9**
- level functions **1-9**
- programming **1-6**

O

- on-chip peripheral bus **1-3**
 - peripheral function interface **1-4**

P

- pwm **8-1**
 - operation **8-1**

R

- related publications **xiv**

S

- scc

- dma operation **3-5**
 - features **3-1**
 - interfaces **3-2**
 - overview **3-1**
 - uarts 1 and 2 **3-4**

- sci

- clocking **4-4**
 - features **4-1**
 - operation **4-3**
 - programming interface **4-4**
 - protocols **4-3**
 - registers **4-5**

- scp **5-1**

- addressing **5-1**
 - dma operation **5-3**

- serial communications controller **3-1**
 - overview **3-1**

- Serial Communications Protocol **5-1**

- serial communications protocol **5-1**

- signals

- by signal name **A-3**

- smart card interface **4-1**

- STBP

- connection to 403 **1-3**
 - features **1-1**
 - function address **1-4**
 - interface connections **1-3**

- stbp

- ac specifications **A-15**
- electrical specifications **A-14**
- hardware specifications **A-1**
- pinout description **A-2**

U

- uart **3-5**
 - addressing **3-9**
 - handshaking **3-4**
 - io signals **3-5**
 - registers **3-12**
 - signals **3-5**

