Xgate Co-Processor (xgate) Specification

Author: Robert Hayes
rehayes@opencores.org

Rev. 0.1
October 7, 2009

This page has been intentionally left blank.

Revision History

Rev.	Date	Author	Description
0.1	09/29/09	Robert Hayes	

Contents

1

Introduction

The Xgate Co-processor Module, Xgate, is a 16 bit programmable RISC processor that is managed by a host CPU to reduce the host load in handling interrupts. Because the Xgate is user programmable there is a great deal of user control in how to preprocess data from peripheral modules. This module may be configured as a simple DMA controller to organize data such that the host only operates with whole messages and not individual words or bytes. The Xgate may also handle higher levels of messaging protocols than the peripheral hardware recognizes. Encryption algorithms are also supported by the instruction set.

The ideal application for the Xgate co-processor is in an ASIC environment where a specific host processor is required but the host as a standalone processor lacks sufficient computational resources to service the target application. Such a situation might occur in an existing application that has an extensive software base or the host processor has a significant history in similar applications. In this environment the Xgate provides the additional resources needed for the application while also providing the user programmability that may allow the ASIC to be used in other applications.

The use of an Xgate co-processor in an FPGA may not be as advantageous as in an ASIC environment because of the reprogramability of the FPGA hardware. Because the Xgate co-processor offers a generic solution to a variety of problems and the functionality can be changed by upgrading user software without changing the FPGA hardware implementation it may still be a valuable solution to certain problems. The use of the Xgate module may also reduce development time since the application can begin debug sooner and changes can be prototyped in user software.

FEATURES

- **Instruction set compatible with Freescale XGATE co-processor**
- **Handles up to 127 interrupt inputs**
- **Eight software triggerable interrupt channels.**
- **Eight semaphore registers to coordinate host/Xgate shared memory.**

- **Static synchronous design**
- **Fully synthesizable**

2

Architecture

The Xgate core is built around five primary blocks; the WISHBONE Slave Interface, WISHBONE Master Interface, the Control Registers, Interrupt Priority, and RISC Processor Core.

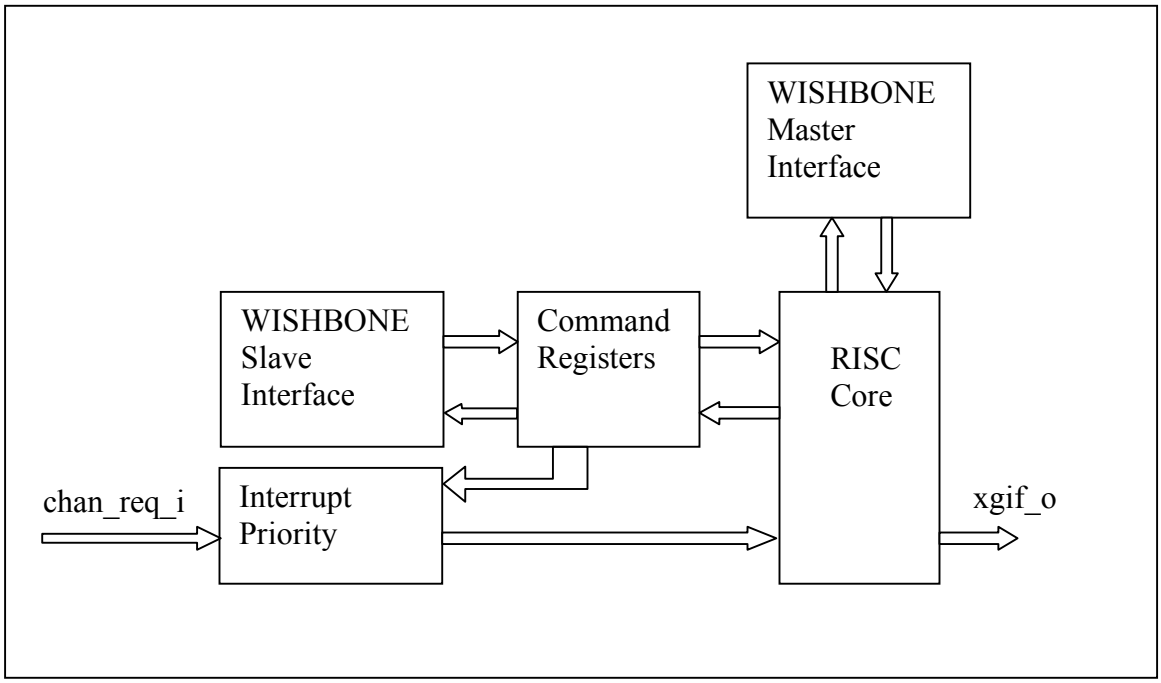


Fig. 2.1 Internal structure Xgate Core

2.1 WISHBONE Slave Interface

The host processor uses this bus to access the Xgate control and status registers. The WISHBONE Slave Interface isolates the Xgate functionality from the WISHBONE bus. This interface takes the bus specific signals and generates a generic set of control signals to drive the Xgate control registers. Isolating the WISHBONE bus should help promote Xgate module reusability by localizing the scope of changes needed to retarget the Xgate module to another bus environment.

2.2 Control Registers

This module receives the generic write signals from the WISHBONE Slave Interface and applies these signals to the registers that the host processor uses to control Xgate operation.

2.3 Interrupt Interface

The Interrupt Interface takes the input interrupts from other slave peripheral modules and prioritizes them for processing by the Xgate RISC Processor Core.

2.3.1 Simple Interrupt Interface

The current implementation uses the simplest interrupt interface with the least hardware and least functionality. This module takes the 127 interrupt inputs and does a simple encoding to generate the six bit bus that becomes the active channel select signal. The active interrupt input with the highest index number becomes the chosen channel select enable.

Because the Xgate module is in the signal path from the interrupt sources to the host CPU there must always be always be some code running in the Xgate module to activate the appropriate interrupt output to the CPU.

2.3.2 Bypass Interrupt Interface

This is a proposed functional improvement and is not currently implemented. This module will do the same basic interrupt encoding as the “Simple Interrupt Interface” except it will include the additional functionality to allow an interrupt input to be bypassed around the Xgate module and connected directly to the host CPU. This will eliminate the latency and software overhead when it is required for the host CPU to process some interrupts directly while still reserving the option to enable the Xgate module to preprocess those same interrupt sources at some other time. To implement this functionality additional control registers will need to be added to the WISHBONE slave interface.

2.3.3 Programmable Interrupt Interface

This is a proposed functional improvement and is not currently implemented. This module will include the functionality of the "Bypass Interrupt Interface" with additional functionality to allow interrupt priorities to be set by the host CPU. This will eliminate the latency and software overhead when it is required for the host CPU to process some interrupts directly while still reserving the option to enable the Xgate module to preprocess those same interrupt sources at some other time. To implement this functionality additional control registers will need to be added to the WISHBONE slave interface.

2.4 WISHBONE Master Interface

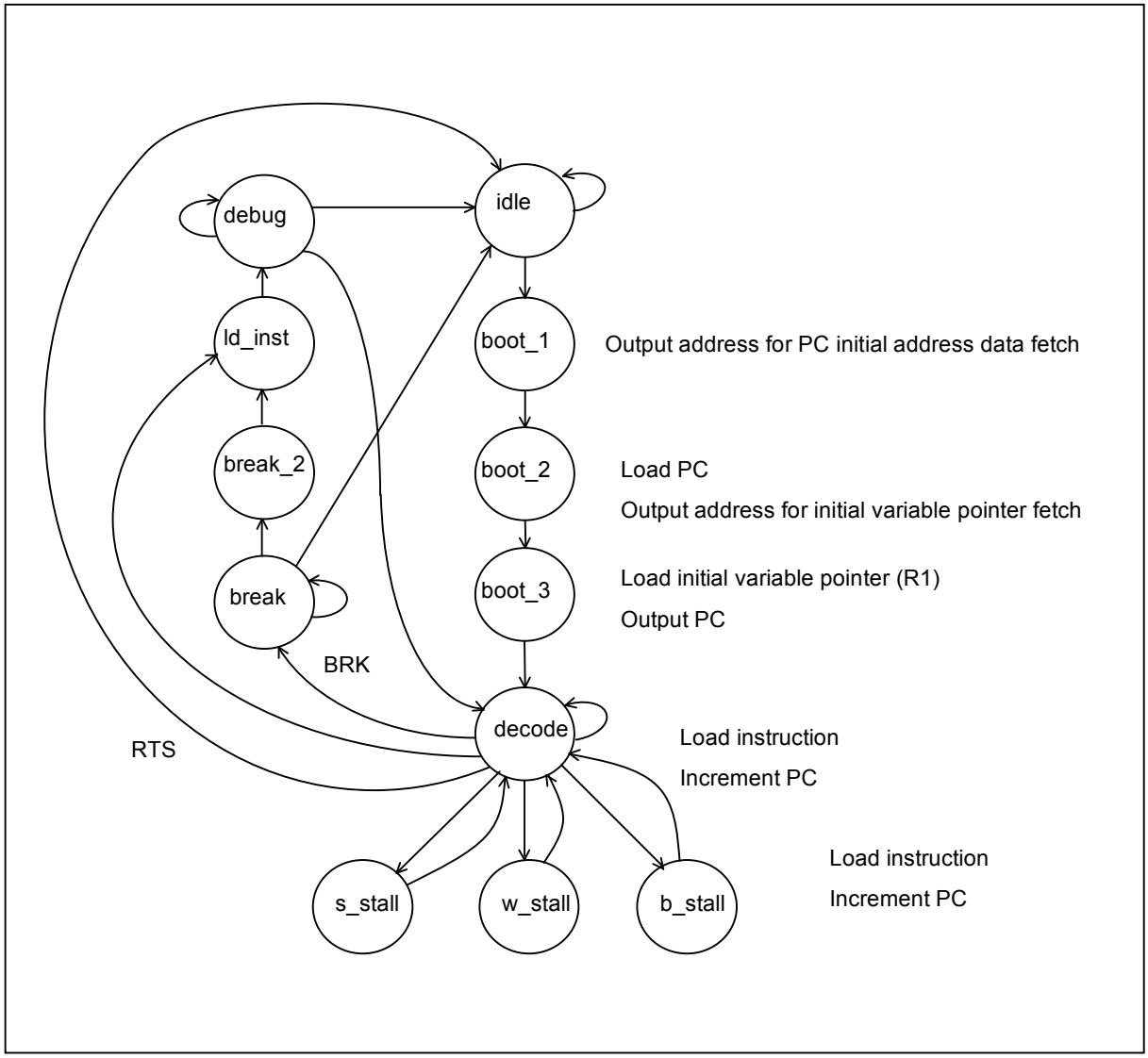
The Xgate module uses the WISHBONE Master Interface to access the shared memory space where it's software code, data, and other slave peripheral registers may be accessed. The WISHBONE Master Interface isolates the Xgate functionality from the WISHBONE bus. This interface takes the bus specific signals and generates a generic set of control signals to drive the Xgate RISC Processor. Isolating the WISHBONE bus should help promote Xgate module reusability by localizing the scope of changes needed to retarget the Xgate module to another bus environment.

2.5 RISC Processor Core

The RISC Processor Core is the key element of the Xgate module. This module implements the Xgate instruction set.

2.5.1 Hardware Model

2.5.2 State Machine



Idle: The idle state is the condition where the xgate RISC processor waits to be enabled and an interrupt input to be activated.

Boot_1: State entered after an input interrupt is detected. Output the address stored in XGVBR + 2*XGCHID to fetch PC (Program Counter).

Boot_2: State entered after Boot_1. Load the PC and output the address of the variable pointer, XGVBR + 2*XGCHID + 2.

Boot_3: State entered after Boot_2. Load the variable pointer. Output the PC and prepare to load the first instruction.

Decode: In normal user operation this is the state entered after Boot_3 and the instruction is decoded and executed. For single cycle instructions, if Debug Mode is not active, then the Program Counter is incremented and the next instruction is loaded. For multi-cycle instructions one of the three instruction continue states is selected. If an RTS instruction is decoded then the state machine is returned to the Idle state to wait for the next change in XGCHID.

S_stall: Simple stall is the state that branch instructions use to change the PC when a change of flow is required. This state is also used for Store instructions.

W_stall: Word stall is the state Load instructions use to retrieve a word of RAM memory data.

B_stall: Byte stall is the state Load instructions use to retrieve a byte of RAM memory data.

Break: If the Decode state detects a BRK command then the Break state is entered. The RISC processor remains in the Break state till:

Single Step Command

XGEN is set to zero

XGCHID is set to zero

Break_2: After a Single Step Command is issued in the Break state the PC is incremented in this state.

Ld_inst: The Load Instruction state is used in debug mode to load the next instruction to be executed into the RISC instruction register.

Debug: debug is the state where the RISC processor waits for an input to execute the next command Exit conditions for this state are:

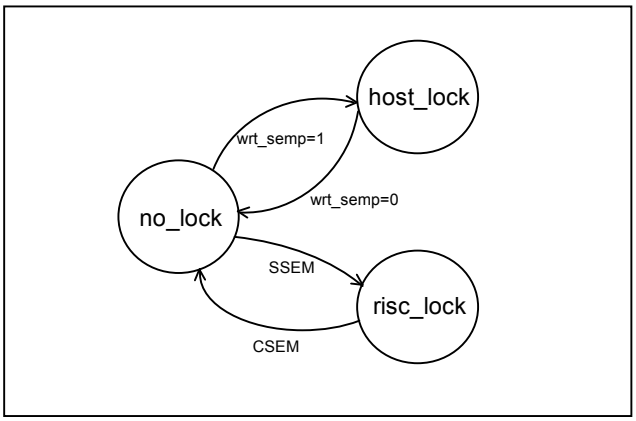
Single Step Command

XGEN is set to zero

XGCHID is set to zero

2.5.3 Semaphore Bit

Each semaphore bit is a simple state machine that is controlled by write commands from either the host or RISC processor. The first write command to set a semaphore bit takes control of the semaphore state until the bit is cleared by the setting processor. The host has priority if the host and RISC processor attempt to write the semaphore bit in the same clock cycle. Each processor has a different read view of the semaphore bit so it can determine if it has successfully set the bit.

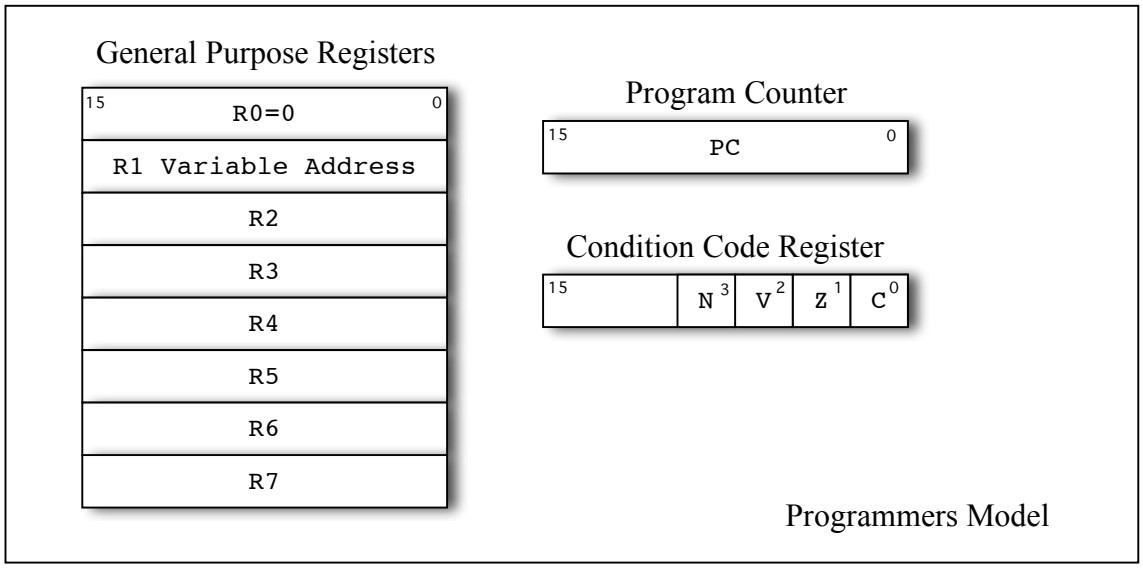


3

Operation

The Xgate Module is a simple RISC CPU with an instruction set ..... then the cop_rst_o will become active to force a system reset.

The COP module also has the capability to generate an interrupt at a programmed number of cycles before the cop_rst_o initiates a system reset. This functionality is primarily intended as a debug feature.



A very readable article about watchdog timers and their use:

3.1 Software Example

The recommended software procedure for using the Xgate Module:

1. Initialize Xgate
   a) Load RAM with Xgate software.
   b) Set XGVBR register to base address of input interrupt vectors.

c) Enable Xgate by setting the XGE bit.

d) Enable host to accept interrupts.

e) Enable peripheral modules to output interrupts as required.

2. Normal Operation

a) Service Xgate interrupt requests as they are received by the host.

3.2 Software Triggers

3.3 Semaphore Bits

3.4 Program Memory Connection Options

3.5 Debug Mode

3.6 Instruction Set Summary

The RISC Processor Core is the key element of the Xgate module. This module implements the Xgate instruction set.

Function	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Return to Scheduler and other																
BRK	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
NOP	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

3.7 Inconsistent Instruction Set Documentation

The following instructions have problems in the Freescale documentation.

3.4.1 SUBL

Freescale Documentation:

V: Set if a two's complement overflow resulted from the 8-bit operation; cleared otherwise.

RD[15]old & !RD[15]new
C: Set if there is a carry from the bit 7 to bit 8 of the result; cleared otherwise.
!RD[15]old & RD[15]new

Xgate implementation:

For signed subtraction the rules for overflow are:

Negative number minus Positive number creates a Positive number OR

Positive number minus Negative number creates a Negative number

So the overflow calculation should be:

V = RD[7] & !IMM8[7] & !result[7] | !RD[7] & IMM8[7] & result[7]

The Carry calculation should be:

C = !RD[7] & IMM8[7] | !RD[7] & result[7] | IMM8[7] & result[7]

These calculations for setting the Carry and Overflow bits are also consistent with the calculations used to set the Carry and Overflow bits using the CMPL instruction.

3.4.2 ADDL

Freescale Documentation:

V: Set if a two's complement overflow resulted from the 8-bit operation; cleared otherwise.
RD[15]old & RD[15]new
C: Set if there is a carry from the bit 7 to bit 8 of the result; cleared otherwise.
RD[15]old & RD[15]new

Xgate implementation:

For signed addition the rules for overflow are:

Positive number plus Positive number creates a Negative number OR

Negative number plus Negative number creates a Positive number

So the overflow calculation should be:

V = !RD[7] & !IMM8[7] & result[7] | RD[7] & IMM8[7] & !result[7]

The Carry calculation should be:

C = !RD[7] & IMM8[7] | RD[7] & !result[7] | IMM8[7] & !result[7]

4

Registers

The Xgate Module has a WISHBONE bus interface configured for a 16-bit bus width 16-bit granularity.

This module address map will probably change to a byte addressable to be consistent with the byte addressable functionality of the Xgate RISC processor.

List of Registers

Name	Address	Width	Access	Description
XGMCTL	0x00	16	RW	Xgate Module Control Register
XGCHID	0x01	16	RW	Xgate Channel ID Register
XGISPHI	0x02	16	RW	Xgate Interrupt Stack Pointer High
XGISPLO	0x03	16	RW	Xgate Interrupt Stack Pointer Low
XGVBR	0x04	16	RW	Xgate Vector Base Address
XGIF_7	0x05	16	RW	Xgate Interrupt Flag Register #7
XGIF_6	0x06	16	RW	Xgate Interrupt Flag Register #6
XGIF_5	0x07	16	RW	Xgate Interrupt Flag Register #5
XGIF_4	0x08	16	RW	Xgate Interrupt Flag Register #4
XGIF_3	0x09	16	RW	Xgate Interrupt Flag Register #3
XGIF_2	0x0A	16	RW	Xgate Interrupt Flag Register #2
XGIF_1	0x0B	16	RW	Xgate Interrupt Flag Register #1
XGIF_0	0x0C	16	RW	Xgate Interrupt Flag Register #0
XGSWT	0x0D	16	RW	Xgate Software Trigger Register
XGSEM	0x0E	16	RW	Xgate Semaphore Register
	0x0F	16	R	Reserved
XGCCR	0x10	16	RW	Xgate Condition Code Register

Name	Address	Width	Access	Description
XGPC	0x11	16	RW	Xgate Program Counter
	0x12	16	R	Reserved
XGR1	0x13	16	RW	Xgate Register 1
XGR2	0x14	16	RW	Xgate Register 2
XGR3	0x15	16	RW	Xgate Register 3
XGR4	0x16	16	RW	Xgate Register 4
XGR5	0x17	16	RW	Xgate Register 5
XGR6	0x18	16	RW	Xgate Register 6
XGR7	0x19	16	RW	Xgate Register 7

Table 1: List of registers

4.1 Xgate Module Control Register (XGMCTL)

Bit #	Access	Description
15	W	XGEM, XGE Mask – The XGE bit can only be changed when the XGEM bit is set in the same register access. ‘0’ No change allowed to XGE control bit. ‘1’ Change allowed to XGE control bit. Always reads as ‘0’.
14	W	XGFRZM, XGFRZ Mask – The XGFRZ bit can only be changed when the XGEM bit is set in the same register access. ‘0’ No change allowed to XGFRZ control bit. ‘1’ Change allowed XGFRZ control bit. Always reads as ‘0’.
13	W	XGDGBM, XGDGB Mask – The XGDGB bit can only be changed when the XGDGBM bit is set in the same register access. ‘0’ No change allowed to XGDBG control bit. ‘1’ Change allowed to XGDBG Control bit. Always reads as ‘0’.
12	W	XGSSM, XGSS Mask – The XGSS bit can only be changed when the XGSSM bit is set in the same register access. ‘0’ No change allowed to XGSS control bit.

Bit #	Access	Description
		'1' Change allowed to XGSS control bit. Always reads as '0'.
11	W	XGFACTM, XGFACT Mask – The XGFACT bit can only be changed when the XGFACTM bit is set in the same register access. '0' No change allowed to XGFACT control bit. '1' Change allowed to XGFACT control bit. Always reads as '0'.
10		BRK_IRQ_ENM, BRK_IRQ_EN Mask – The BRK_IRQ_EN bit can only be changed when the BRK_IRQ_ENM bit is set in the same register access. '0' No change allowed to BRK_IRQ_EN control bit. '1' Change allowed to BRK_IRQ_EN control bit. Always reads as '0'.
9	RW	XGSWEIFM, XGSWEIF Mask – The XGSWEIF bit can only be changed when the XGSWEIFM bit is set in the same register access. '0' No change allowed to XGSWEIF control bit. '1' Change allowed to XGSWEIF control bit. Always reads as '0'.
8	RW	XGIEM, XGIE Mask – The XGIE bit can only be changed when the XGIEM bit is set in the same register access. '0' No change allowed to XGIE control bit. '1' Change allowed to XGIE control bit. Always reads as '0'.
7	RW	XGE, XG Enable – The XGE bit can only be changed when the XGEM bit is set in the same register access. '0' The Xgate module is disabled. The currently active interrupt request will be completed and then no new interrupt processing requests will be accepted. '1' The Xgate module is enabled to start processing channel interrupts.
6	RW	XGFRZ, XG Freeze Mode – The XGE bit can only be changed when the XGEM bit is set in the same register access. '0' The Xgate module is disabled. The currently active channel interrupt will be completed and then no new interrupt channel processing requests will be accepted.

Bit #	Access	Description
		'1' The Xgate module is enabled to start processing channel interrupts.
5	RW	XGDBG, XG Debug Mode – The XGDBG bit can only be changed when the XGDBGM bit is set in the same register access. '0' The Xgate module is not in DEBUG Mode. '1' The Xgate module is in DEBUG Mode. Along with writing to the XGGBG bit from the host interface DEBUG Mode can also be entered by executing the 'BRK' instruction by the RISC Processor Core.
4	RW	XGSS, XG Single Step – The XGSS bit can only be changed when the XGSSM bit is set in the same register access. '0' The Xgate module is disabled. The currently active channel interrupt will be completed and then no new interrupt channel processing requests will be accepted. '1' Write - Xgate module should execute a single RISC instruction. Read – A RISC instruction is being processed.
3	RW	XGFACT, XG Fake Activity – The XGFACT bit can only be changed when the XGFACTM bit is set in the same register access. '0' The Xgate module is disabled. The currently active channel interrupt will be completed and then no new interrupt channel processing requests will be accepted. '1' The Xgate module is enabled to start processing channel interrupts.
2	RW	BRK_IRQ_EN, Break Interrupt Enable – The BRK_IRQ_EN bit can only be changed when the BRK_IRQ_EN M bit is set in the same register access. '0' The Xgate module is operating in a mode compatible with the Freescale XGATE module. BRK instructions do not generate an Xgate error interrupt output. '1' The Xgate module is enabled to output an error interrupt whenever a BRK command is encountered. Set this bit whenever no BRK commands are expected to be used in the software.
1	RW	XGSWEIF, XG Software Error Interrupt Flag – The XGSWEIF bit can only be changed when the XGSWEIFM bit is set in the same register access. '0' The Xgate module is operating correctly. Writing '0' has

Bit #	Access	Description
		no effect. '1' The Xgate RISC processor has detected an error condition. Writing '1' will clear a set XGSWEIF bit and terminate the current interrupt process. The Xgate module will go to an IDLE state if no other interrupt inputs are active.
0	RW	XGIE, XG Interrupt Enable – The XGIE bit can only be changed when the XGEM bit is set in the same register access. '0' All Xgate module interrupt outputs are disabled. '1' All Xgate module interrupt outputs are enabled.

Reset Value:

XGMCTL: 0004h

Table 2: CNTRL Register Bits

4.2 Xgate Channel ID Register (XGCHID)

Bit #	Access	Description
15:7	R	Reserved, write zeros for future compatibility.
6:0	RW	XGCHID, XG Channel ID – The XGCHID register reflects the value of the current interrupt channel being processed. '0' No interrupt currently being processed. '1' The encoded value of the input interrupt currently being processed. The XGCHID can only be written in DEBUG Mode. When a value is written to the XGCHID register the RISC Processor Core will start executing the code associated with that interrupt input.

Reset Value:

XGCHID: 0000h

Table 4: Channel ID Register Bits

4.3 Xgate Stack Pointer Register (XGISPHI)

Bit #	Access	Description

Bit #	Access	Description
15:0	RW	XGISPHI, Read returns the current state of the COP Watchdog Counter. Writes to the CNT register access the COP service words. To reset the COP Watchdog Counter to the TOUT_VAL state two specific service words must be written in the correct order to prevent a COP reset.

Reset Value:
XGISP74: 0000h

Table 5: XGIPS74 Register Bits

4.4 Xgate Stack Pointer Register (XGISPLO)

Bit #	Access	Description
15:0	RW	XGISPLO, Read returns the current state of the COP Watchdog Counter. Writes to the CNT register access the COP service words. To reset the COP Watchdog Counter to the TOUT_VAL state two specific service words must be written in the correct order to prevent a COP reset.

Reset Value:
XGISP31: 0000h

Table 5: XGIPS31 Register Bits

4.5 Xgate Vector Base Address Register (XGVBR)

Bit #	Access	Description
15:1	RW	XGVBR, Sets the vector base address that the Xgate RISC Core uses to determine where in the memory map the code and data space is for a specific interrupt is reserved.

Reset Value:
XGVBR: FE00h

Table 5: XGVBR Register Bits

4.6 Xgate Interrupt Flag Register 7 (XGIFR_7)

The bits in this register reflect the state of interrupts that are being output from the Xgate module to the host CPU. A bit is set by the Xgate module with the "SIF" instruction usually at the end of a process and the host CPU clears the XGIF bit as part of its interrupt service routine.

Clear the interrupt output by writing a '1' to the corresponding XGIF register bit.

The highest numbered interrupt is in the MSB position of the register bank.

Some bits in this register may be unimplemented in a specific hardware design based on the setting of the "MAX_CHANNEL" parameter and will always read as "0".

Bit #	Access	Description
15:0	RW	XGIF_127 – XGIF_112 '0' No interrupt. '1' Interrupt output active.

Reset Value:
XGIFR_7: 0000h

Table 5: XGIF_7 Register Bits

4.6 Xgate Software Trigger Register (XGSWT)

Bit #	Access	Description
15:8	W	XGSWTM[7:0], Read returns the current state of the COP Watchdog Counter. XGSWTM, XGSWT Mask – The XGSWT bit can only be changed when the XGSWTM bit is set in the same register access. '0' No change allowed to XGSWT control bit. '1' Change allowed to XGSWT control bit. Always reads as '0'.
7:0	RW	XGSWT[7:0], XG Software Trigger – The XGSWT bit can only be changed when the XGSWTM bit is set in the same register access. '0' The Xgate module is disabled. The currently active interrupt request will be completed and then no new interrupt processing requests will be accepted.

Bit #	Access	Description
		'1' The Xgate module is enabled to start processing channel interrupts.

Reset Value:
XGSWT: 0000h

Table 5: XGSWT Register Bits

4.6 Xgate Semaphore Register (XGSEM)

Bit #	Access	Description
15:8	W	XGSEMM, XGSEM Mask – The XGSEM bit can only be changed when the XGSEMM bit is set in the same register access. '0' No change allowed to XGSEM control bit. '1' Change allowed to XGSEM control bit. Always reads as '0'.
7:0	RW	XGSWT[7:0], XG Software Trigger – The XGSWT bit can only be changed when the XGSWTM bit is set in the same register access. '0' The semaphore bit is either unlocked or locked by the Xgate module. If the host attempts to set the bit by writing a '1' and a subsequent read of the bit returns a '0' the bit is locked by the Xgate module. '1' The semaphore bit is locked by the host processor. The host writes a '0' to clear the bit and unlock the semaphore.

Reset Value:
XGSEM: 0000h

Table 5: XGSEM Register Bits

4.6 Xgate Condition Code Register (XGCCR)

Bit #	Access	Description
15:4	R	Reserved, write zeros for future compatibility.

Bit #	Access	Description
3	RW	XGN, Xgate Negative – Reflects the current state of the "Negative" bit of the Xgate condition code register.
2	RW	XGZ, Xgate Zero – Reflects the current state of the "Zero" bit of the Xgate condition code register.
1	RW	XGV, Xgate Overflow – Reflects the current state of the "Overflow" bit of the Xgate condition code register.
0	RW	XGC, Xgate Carry – Reflects the current state of the "Carry" bit of the Xgate condition code register.

Reset Value:

XGCCR: 0000h

Table 5: XGCCR Register Bits

4.6 Xgate Program Counter Register (XGPC)

Bit #	Access	Description
15:0	RW	XGPC, Read returns the current state of the Xgate Program Counter. Writes to the XGPC register can only be done in Debug Mode.

Reset Value:

XGPC: 0000h

Table 5: XGPC Register Bits

4.6 Xgate Register 1 (XGR1)

Bit #	Access	Description
15:0	RW	XGR1, Read returns the current state of the Xgate General Purpose Register 1. Writes to the XGR1 register can only be done in Debug Mode.

Reset Value:

XGR1: 0000h

Table 5: XGR1 Register Bits

5

Clocks

Name	Source	Rates (MHz)			Remarks	Description
		Max	Min	Resolution		
wbs_clk_i	System	200	-	-	Master clock for all COP bus registers. Positive edge active.	System clock.
risc_clk	System					RISC Processor clock

Table 3: List of clocks

The wbs_clk_i has no timing constraints based on the RTL implementation although there may be constrains applied for synthesis results to be compatible with the target physical implementation. If the Xgate is targeted for an ASIC implementation then [wbs_clk_i] should be used as the scan clock, any clock multiplexing required to make [wbs_clk_i] the scan clock should be done at the system level external to the Xgate Module.

The risc_clk is the clock used to run the Xgate RISC processor submodule. The frequency of this clock should be the same as or 2X the frequency of the wbs_clk_i input. The risc_clk input is assumed to be in phase with the wbs_clk_i input.

6

IO Ports

Port	Width	Direction	Description
wbs_clk_i	1	Input	WISHBONE Bus Clock Input, Master Clock
wbs_rst_i	1	Input	WISHBONE Synchronous Reset
wbs_adr_i	3	Input	WISHBONE Lower address bits
wbs_dat_i	16	Input	WISHBONE Bus Data
wbs_dat_o	16	Output	WISHBONE Bus Data
wbs_we_i	1	Input	WISHBONE Write enable
wbs_stb_i	1	Input	WISHBONE Strobe signal/Core select
wbs_cyc_i	1	Input	WISHBONE Valid bus cycle
wbs_sel_i	2	Input	WISHBONE Data Bus Byte Select
wbs_ack_o	1	Output	WISHBONE Bus cycle acknowledge
xgif_o	128	Output	Xgate Interrupt signal
xgswt	8	Output	Xgate software trigger
chan_req_i	128	Input	Xgate Interrupt request
risc_clk	1	Input	Clock for RISC submodule
	1	Input	
debug_mode_i	1	Input	System is in DEBUG Mode
	1	Input	
arst_i	1	Input	Asynchronous Reset

Port	Width	Direction	Description
scantestmode_i	1	Input	Scan Test Mode Enable

Table 4: List of IO ports

6.1 WISHBONE Interface

The core features a WISHBONE RevB.3 compliant WISHBONE Classic interface that operates in SLAVE mode. All output signals are registered. Each access takes 2 clock cycles. To limit a WISHBONE access to just two clock cycles the following synthesis rules should be used:

- Single cycle timing for wbs_cyc_i and wbs_stb_i
- Two cycle timing for wbs_adr_i, wbs_data_i, and wbs_data_o. (Single cycle timing could be used but it would be a waste of resources to meet an over constrained timing path.)

Note: Use the "SINGLE_CYCLE" parameter to do a WISHBONE bus access in one clock cycle.

WISHBONE DATASHEET		
Description	Specification	
General description:	8-bit SLAVE	
Supported Cycles:	SLAVE, READ/WRITE	
Data port, size: Data port, granularity: Data port, maximum operand size: Data transfer ordering: Data transfer sequencing:	Default: 16, option 8 bit Default: 16, option to match 8 bit port size	
Supported signal list and cross reference to equivalent WISHBONE signals:	Signal Name	WISHBONE Equiv.
	wbs_clk_i	CLK_I
	wbs_rst_i	RST_I
	wbs_adr_i	ADR_I()

	wbs_dat_i	DAT_I()
	wbs_dat_o	DAT_O()
	wbs_we_i	WE_I
	wbs_stb_i	STB_I
	wbs_cyc_i	CYC_I
	wbs_sel_i	SEL_I
	wbs_ack_o	ACK_O

6.1.1 wbs_rst_i

The synchronous reset signal has a minimum pulse width requirement of one [wbs_clk_i] clock period. It will take two [wbs_clk_i] clock cycles for all registers in the COP Module to initialize. Also see information on pin [arst_i]. It is assumed that elsewhere in the system cop_rst_o will be OR'ed with this signal and it will be active for some time during and after a COP watchdog timeout.

6.1.2 wbs_adr_i

Connections to the WISHBONE address pin will depend on the size of the WISHBONE data bus that is set by the DWIDTH parameter. If DWIDTH=8 the all address pins should be connected, if DWIDTH=16 then [wbs_adr_i(2)] should be tied low.

6.1.3 wbs_sel_i

The [wbs_sel_i] is the WISHBONE byte lane select signal. It is currently unimplemented in the PIT module and should be tied hi.

6.2 Xgate signals

6.2.1 xgif[127:0]

The Xgate output signal. This output is the result of executing the SIF command by the Xgate RISC processor. The normal connection is to the interrupt inputs of the host processor.

6.2.2 chan_req_i[127:0]

These signals are the interrupt inputs from the prepherial modules that the Xgate services. The eight outputs of the xgswt bus may also connect to these inputs.

6.2.3 xgswt[7:0]

The [por_reset_i] signal is active low. This signal is used to initialize a limited number flip-flops that need to maintain state when [arst_i] or [wb_rst_i] are active. A separate reset signal is required for these flip_flops because it is assumed that [cop_rst_o] will effect [arst_i] and [wb_rst_i] and may cause a shortening of the [cop_rst_o] pulse width because of a combinational feedback path.

6.2.4 arst_i

The signal [arst_i] is an asynchronous reset signal that goes to all flops in the COP. It is provided for FPGA implementations and test methodologies that require this function for initialization. Using [arst_i] instead of [wb_rst_i] can result in lower cell-usage and higher performance for a FPGAs implementation because the standard FPGA cell already provides a dedicated asynchronous reset path. Using [wb_rst_i] for an ASIC implementation might synthesize to a smaller module because smaller non_reset flops can be used. Use either [arst_i] or [wb_rst_i], tie the other to a negated state. The active level of [arst_i] is determined by the parameter ARST_LVL that defaults to active low.

6.2.6 stop_mode_i

The Watchdog Counter can be frozen in its current state if this signal is active and the STOP_ENA bit is set to zero. If there is no stop mode signal available from the system then this input signal should be tied low.

6.2.7 wait_mode_i

The Watchdog Counter can be frozen in its current state if this signal is active and the WAIT_ENA bit is set to zero. If there is no stop mode signal available from the system then this input signal should be tied low.

6.2.8 debug_mode_i

The Watchdog Counter can be frozen in its current state if this signal is active and the DEBUG_ENA bit is set to zero. If there is no stop mode signal available from the system then this input signal should be tied low.

6.2.9 scantestmode_i

The [scantestmode_i] input is an optional signal used to put the module into scan test mode. When [scantestmode_i] is active the [startup_osc_clk_i] is replaced by the [wb_clk_i] clock so all register are clocked by a common clock source.

6.3 Xgate Core Parameters

Parameter	Type	Default	Description
ARST_LVL	Bit	1'b0	Asynchronous reset level
MAX_CHANNEL	Integer	127	Number of input an output interrupts
SINGLE_CYCLE	Bit	1'b0	WISHBONE wait state

6.3.1 ARST_LVL

The asynchronous reset level can be set to either active high (1'b1) or active low (1'b0).

Allowed values: 1'b0, 1'b1

6.3.2 MAX_CHANNEL

The maximun index value for chan_req_i and xgif.

Allowed values: any integer value between 1 and 127

6.3.3 SINGLE_CYCLE

The default operation of the COP WISHBONE bus interface is to insert one wait state by delaying the assertion of the wb_ack_o by one wb_clk_i period. Setting the SINGLE_CYCLE parameter generates the wb_ack_o combinationaly so a WISHBONE bus cycle can be completed in one wb_clk_i period.

Allowed values: 1'b0, 1'b1

Appendix A

Name

[This section may be added to outline different specifications.]

Index

[This section contains an alphabetical list of helpful document entries with their corresponding page numbers.]