VS2000

32-Bit Lightfoot™ CPU with Ethernet MAC

Preliminary
Product Manual

5673 West Las Positas Boulevard
Suite 206
Pleasanton, CA 94588
Phone: (925) 924-0550
Fax: (925) 924-0555

Unit 5 Windmill Business Village
Brooklands Close
Sunbury
Middlesex TW16 7DY
United Kingdom
Phone: +44 (0)1932-784858
Fax: +44 (0)1932-782811

info@velocitysemi.com

www.velocitysemi.com

0080-v4–06/07/04–xM

Table of Contents

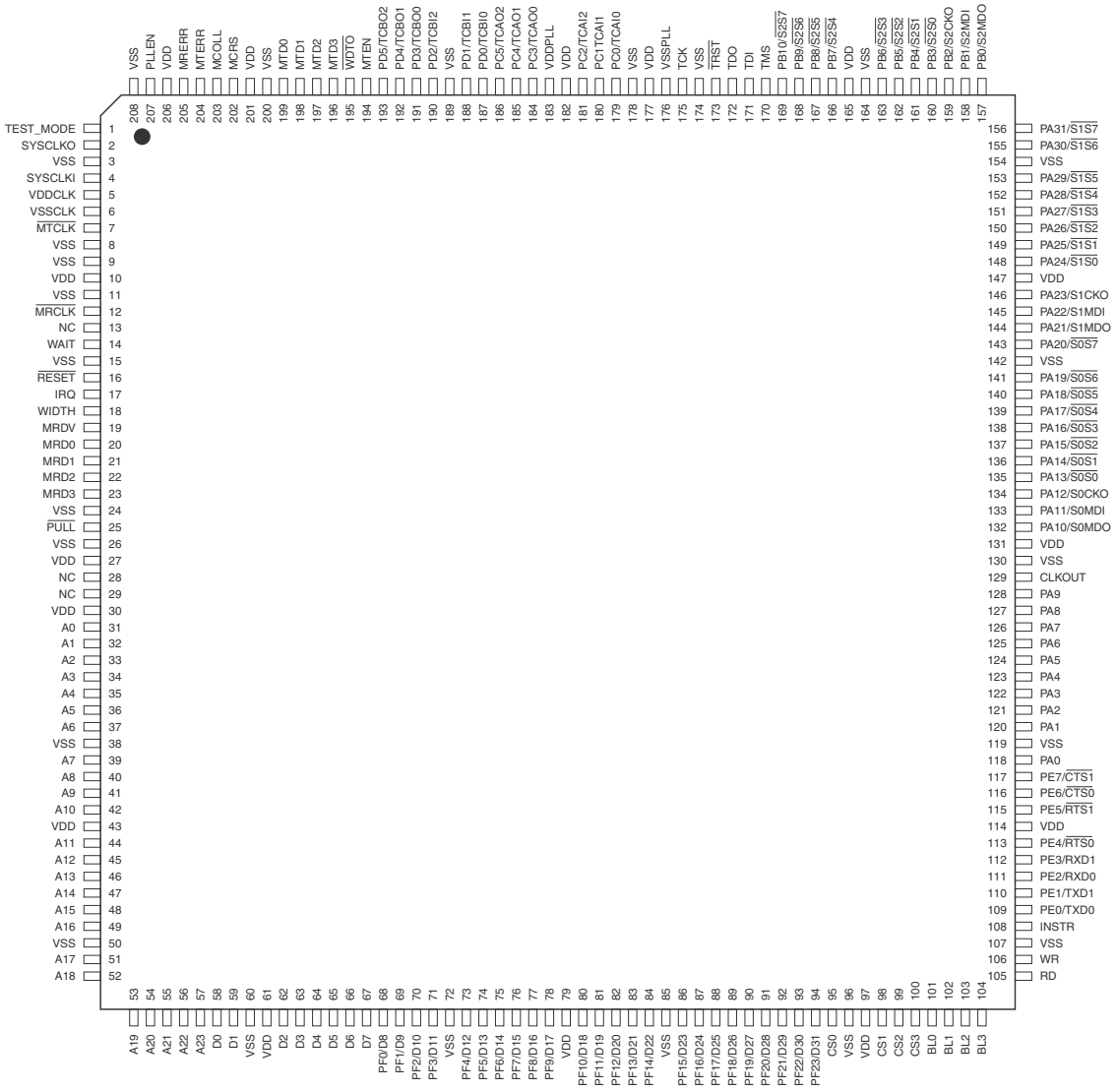
Chapter 1. General Description

The VS2000, a member of the Typhoon™ family, is a highly-integrated and optimized single-chip network-oriented microcontroller solution. The VS2000 microcontroller incorporates the Lightfoot CPU core, a low-complexity, high-performance 32-bit stack RISC CPU. The Lightfoot CPU makes Typhoon ideal for implementing embedded systems that use object-oriented technologies such as Java™ and .NET™. In addition to the Lightfoot CPU, the Typhoon microcontroller features 4 KB of instruction and data caches, a memory controller with an 8- or 32-bit external interface, an Ethernet MAC unit and a wide range of peripherals including UARTs, timers, GPIOs and SPI in a single-package solution.

The unparalleled code density offered by the Lightfoot architecture and instruction-set offers a dramatic reduction in system memory costs, delivering 32-bit computing at a cost usually associated with 8- and 16-bit solutions. The VS2000 is targeted at a wide range of embedded industrial, consumer, networking and machine-to-machine applications.

1.1 Pin Definitions

Figure 1-1. 208-Lead LQFP Pinout Diagram (top view)



1.1.1 Power Pins

Table 1-1. Power and Ground

Name	Location	Type	Description
VDDPLL	183	PWR	**PLL Power** — VDDPLL is an isolated power dedicated to Phase Lock Loop (PLL) use. The voltage should be well-regulated and the input should be provided with an extremely low-impedance path to the V_{DD} power rail. VDDPLL should be bypassed to V_{SS} by a 0.47 µF capacitor located as close as possible to the chip package. There is one V_{DDPLL} input.
VDD (15)	10, 27, 30, 43, 61, 79, 97, 114, 131, 147, 165, 177, 182, 201, 206	PWR	**Power** — VDD is the device power supply. These inputs must be tied externally to all other chip power inputs. The user must provide adequate external decoupling capacitors. There are 15 VDD inputs.
VDDCLK	5	PWR	**Clock Power** — VDDCLK is an isolated power dedicated to the clock buffers. This input must be tied externally to all other chip power inputs. The user must provide adequate external decoupling capacitors. There is one VDDCLK input.
VSSPLL	176	GND	**PLL Ground** — VSSPLL is an isolated ground dedicated to Phase Lock Loop (PLL) use. The connection should be provided with an extremely low impedance path to ground. VSSPLL should be bypassed to V_{SS} by a 0.47 µF capacitor located as close as possible to the chip package. There is one VSSPLL connection.
VSS (24)	3, 8, 9, 11, 15, 24, 26, 38, 50, 60, 72, 85, 96, 107, 119, 130, 142, 154, 164, 174, 178, 189, 200, 208	GND	**Ground** — VSS is the device ground return. This input must be tied externally to all other chip ground inputs. The user must provide adequate external decoupling capacitors. There are 24 VSS inputs.
VSSCLK	6	GND	**Clock Ground** — VSSCLK is an isolated power dedicated to the clock buffers. This input must be tied externally to all other chip ground inputs. The user must provide adequate external decoupling capacitors. There is one VSSCLK input.

1.1.2 Clock Pins

Table 1-2. Clocks

Name	Location	Type	Description
SYSCLKI	4	I	**System Clock/Crystal Input** — SYSCLKI interfaces the internal crystal oscillator input to an external crystal or an external clock.
SYSCLKO	2	O	**System Clock/Crystal Output** — SYSCLKO connects the internal crystal oscillator output to an external crystal. If an external clock is used, leave SYSCLKO unconnected.
CLKOUT	129	O	**Clock Output** — CLKOUT provides an output clock synchronized to the internal core clock phase.

1.1.3 System Control Pins

Table 1-3. System Control

Name	Location	Type	Description
$\overline{\text{RESET}}$	16	I	**System Reset** — $\overline{\text{RESET}}$ is an active-low, Schmitt-trigger input with pull-up. When asserted, the chip is placed in the Reset state and the internal phase generator is reset. The Schmitt-trigger input allows a slowly rising input (such as a capacitor charging) to reset the chip reliably. If the Phase Lock Loop is used, $\overline{\text{RESET}}$ must be asserted for a minimum of 600 µs from power-on. This guarantees that the PLL has locked. If the PLL is not used, $\overline{\text{RESET}}$ must be asserted for a minimum of ten system clock cycles where, in this case, the system clock frequency equals the external clock frequency divided by two. $\overline{\text{RESET}}$ must be asserted after power is applied. $\overline{\text{RESET}}$ is 5V tolerant.
IRQ	17	I	**Interrupt Request** — IRQ is a programmable-edge/level Schmitt-trigger pull-down input maskable interrupt request internally synchronized to CLKOUT. IRQ is 5V tolerant.

1.1.4 External Memory Interface Pins

Table 1-4. External Memory Interface

Name	Location	Type	Description
A[23:17] A[16:11] A[10:7] A[6:0]	57-51 49-44 42-39 37-31	O	**Address Bus** — A[23:0] are active-high outputs that specify the address for external program and data memory accesses. To minimize power consumption, A[23:0] do not change state unless external memory spaces are being accessed. The address bus is tri-stated during reset.
D[31:23]/ PF[23:15] D[22:18]/ PF[14:10] D[17:12]/ PF[9:4] D[11:8]/ PF[3:0] D[7:2] D[1:0]	94-86 84-80 78-73 71-68 67-62 59-58	I/O	**Data Bus** — D[31:0] are active-high, bidirectional data bus pins for external program and data memory accesses. The data bus defaults to inputs during reset and are only driven during a write operation. If configured in 8-bit mode, D[7:0] are high-impedance and PF[23:0] are bidirectional GPIO port F. Pins D[31:8] default to GPIO inputs.
CS[3:1] CS0	100-98 95	O	**Chip Select** — CS[3:0] provide four programmable-polarity external chip selects. These signals are tristated during reset.
BL[3:0]	104-101	O	**Byte Lane Select** — BL[3:0] are used to select destination bytes for 32-bit external memory write accesses. These programmable-polarity outputs are tristated during reset.
RD	105	O	**Read Enable** — RD is a programmable-polarity output that is asserted to read external memory on the data bus (D[31:0]). RD is tristated during reset.
WR	106	O	**Write Enable** — WR is a programmable-polarity output that is asserted to write external memory on the data bus (D[31:0]). WR is tristated during reset.
WAIT	14	I	**Wait** — WAIT is an active-high input used by external devices to delay completion of bus cycles. The WAIT input must be asserted synchronously with CLKOUT to guarantee proper operation.
WIDTH	18	I	**Data Bus Width** — The WIDTH input configures the width of the external memory interface. When high the data bus is configured as an 8 bit interface, otherwise the data bus is configured as a 32-bit interface.
INSTR	108	O	**Instruction Access** — INSTR is an active-high output indicating that the current memory access is in the VS2000's program memory space. It is deasserted for external data memory space accesses.

1.1.5 10/100 PHY MII Pins

Table 1-5. MII

Name	Location	Type	Description
$\overline{\text{MTCLK}}$	7	I	**Transmit Symbol (nibble) Clock** — $\overline{\text{MTCLK}}$ is supplied by the PHY at 25 MHz for 100 Mbits/sec. operation, at 2.5 MHz for 10 Mbits/sec. $\overline{\text{MTCLK}}$ is a 5V-tolerant pull-down input. MTD[3:0], MTEN and MTERR are synchronous to this clock.
MTD[3:0]	196-199	O	**Transmit Data Nibble** — MTD[3:0] are active-high transmit data nibble outputs. MTD[3:0] are synchronized to the rising edge of MTCLK.
MTEN	194	O	**Transmit Enable** — MTEN is an active-high output that indicates valid data is present on MTD[3:0]. MTEN remains asserted until all nibbles have been clocked into the external PHY. It is deasserted following the clock edge loading the final frame.
MTERR	204	O	**Transmit Coding Error** — MTERR is an active-high output that indicates to the PHY that a coding error has occurred when asserted at the same time as MTEN.
$\overline{\text{MRCLK}}$	12	I	**Receive Symbol (nibble) Clock** — $\overline{\text{MRCLK}}$ is supplied by the PHY at 25 MHz for 100 Mbits/sec. operation, at 2.5 MHz for 10 Mbits/sec. $\overline{\text{MRCLK}}$ is a 5V-tolerant pull-down input.
MRDV	19	I	**Receive Data Valid** — MRDV is an active-high input synchronized to the rising-edge of $\overline{\text{MRCLK}}$. MRDV is asserted by the PHY to indicate that the MRD[3:0] contain valid data. MRDV is a 5V-tolerant pull-down input.
MRD[3:0]	23-20	I	**Receive Data** — MRD[3:0] are active-high receive data nibble inputs. These signals are synchronized to their rising edge of $\overline{\text{MRCLK}}$. MRD[3:0] are 5V-tolerant pull-down inputs.
MRERR	205	I	**Receive Symbol Error** — MRERR is an active-high input signal. It is asserted when a media error is detected by the PHY during transmission of the current frame. MRERR is synchronized with $\overline{\text{MRCLK}}$ and is asserted for one or more clock periods. MRERR is a 5V-tolerant pull-down input.
MCOLL	203	I	**Collision Detected** — MCOLL is an asynchronous active-high input signal indicating that a collision has occurred on the medium. MCOLL is a 5V-tolerant pull-down input.
MCRS	202	I	**Carrier Sense** — MCRS is an asynchronous active-high signal indicating that the medium is in a non-idle state. Otherwise an idle state is indicated (and transmission may start). MCRS is a 5V-tolerant pull-down input.

1.1.6 UART Pins

Table 1-6. Serial Communication Interface(1)

Name	Location	Type	Description
TXD0/PE0	109	I/O	**Transmit Data/GPIO** — TXD0 is an active-high UART0 transmit-data output or bidirectional GPIO port E bit 0. RS-232 applications need external RS-232 transmitters to convert voltage levels. This pin defaults to GPIO input.
TXD1/PE1	110	I/O	**Transmit Data/GPIO** — TXD1 is an active-high UART1 transmit-data output or bidirectional GPIO port E bit 1. RS-232 applications need external RS-232 transmitters to convert voltage levels. This pin defaults to GPIO input.
RXD0/PE2	111	I/O	**Receive Data/GPIO** — RXD0 is an active-high UART0 receive-data input or bidirectional GPIO port E bit 2. RS-232 applications need external RS-232 receivers to convert voltage levels. This pin defaults to GPIO input.
RXD1/PE3	112	I/O	**Receive Data/GPIO** — RXD1 is an active-high UART1 receive-data input or bidirectional GPIO port E bit 3. RS-232 applications need external RS-232 receivers to convert voltage levels. This pin defaults to GPIO input.
$\overline{\text{RTS0}}$/PE4	113	I/O	**Request to Send/GPIO** — $\overline{\text{RTS0}}$ is an active-low UART0 flow-control output that indicates the UART is ready to receive data or bidirectional GPIO port E bit 4. RS-232 applications need external RS-232 transmitters to convert voltage levels. This pin defaults to GPIO input.
$\overline{\text{RTS1}}$/PE5	115	I/O	**Request to Send/GPIO** — $\overline{\text{RTS1}}$ is an active-low UART1 flow-control output that indicates the UART is ready to receive data or bidirectional GPIO port E bit 5. RS-232 applications need external RS-232 transmitters to convert voltage levels. This pin defaults to GPIO input.
$\overline{\text{CTS0}}$/PE6	116	I/O	**Clear to Send/GPIO** — $\overline{\text{CTS0}}$ is an active-low UART0 flow-control input that indicates the UART may send data or bidirectional GPIO port E bit 6. RS-232 applications need external RS-232 transmitters to convert voltage levels. This pin defaults to GPIO input.
$\overline{\text{CTS1}}$/PE7	117	I/O	**Clear to Send/GPIO** — $\overline{\text{CTS1}}$ is an active-low UART1 flow-control input that indicates the UART may send data or bidirectional GPIO port E bit 7. RS-232 applications need external RS-232 transmitters to convert voltage levels. This pin defaults to GPIO input.

Note 1. All SCI pins have pull-ups and are not 5V-tolerant.

1.1.7 Timer Pins

Table 1-7. Timer/Counter Pins(1)

Name	Location	Type	Description
$\overline{\text{WDTO}}$	195	O	**Watchdog Time-Out** — $\overline{\text{WDTO}}$ is an active-low output that indicates the on-chip watchdog timer has overflowed. This output is active for 16 system clock cycles.
TCAI[2:0]/ PC[2:0] TCBI[1:0]/ PD[1:0] TCBI2/PD2	181-179 188-187 190	I/O	**Timer Counter Input/GPIO** — TCAI0, TCAI1, TCAI2, TCBI0, TCBI1, TCBI2 are programmable-polarity edge-sensitive timer inputs or bidirectional GPIO port C bits 0, 1, 2 and GPIO port D bits 0, 1, 2. These pins have pull-ups and are not 5V-tolerant. These pins default to GPIO inputs.
TCAO[2:0]/ PC[5:3] TCBO[2:0]/ PD[5:3]	186-184 193-191	I/O	**Timer Counter Output/GPIO** — TCAO0, TCAO1, TCAO2, TCBO0, TCBO1, TCBO2 are programmable-polarity timer outputs or bidirectional GPIO port C bits 3, 4, 5 and GPIO port D bits 3, 4, 5. These pins have pull-ups and are not 5V-tolerant. These pins default to GPIO inputs.

Note 1. With the exception of $\overline{\text{WDTO}}$, all timer I/O pins have pull-ups and are not 5V-tolerant.

1.1.8 GPIO/SPI Pins

Table 1-8. GPIO Port A/B, SPI2, SPI1, SPI0(1)

Name	Location	Type	Description
PA[31:30]/ $\overline{\text{S1S}}$[7:6] PA[29:24]/ $\overline{\text{S1S}}$[5:0]	156-155 153-148	I/O	**SPI1 Slave Select** — $\overline{\text{S1S}}$[7:0], are active-low output signals used to select the target SPI slave device. **GPIO** — PA[31:24] are general purpose input/output signals. These pins default to GPIO inputs.
PA23/S1CKO	146	I/O	**SPI1 Clock Output** — S1CKO is a programmable SPI clock output. **GPIO** — PA23 is a general purpose input/output signal. This pin defaults to GPIO input.
PA22/S1MDI	145	I/O	**SPI1 Master Data In** — S1MDI is an active-high serial data input for receiving data from SPI slave devices. **GPIO** — PB1 is a general purpose input/output signal. This pin defaults to GPIO input.
PA21/S1MDO	144	I/O	**SPI1 Master Data Out** — S1MDO is an active-high serial output for transmitting data to SPI slave devices. **GPIO** — PB0 is a general purpose input/output signal. This pin defaults to GPIO input.

Note 1. All GPIO and SPI I/O pins have pull-ups and are not 5V-tolerant.

Table 1-8. GPIO Port A/B, SPI2, SPI1, SPI0[1] (continued)

Name	Location	Type	Description
PA[20]/$\overline{S0S}$[7] PA[19:13]/ $\overline{S0S}$[6:0]	143 141-135	I/O	**SPI0 Slave Select** — $\overline{S0S}$[7:0] are active-low output signals used to select the target SPI slave device. **GPIO** — PA[20:13] are general-purpose input/output signals. These pins default to GPIO inputs.
PA12/S0CKO	134	I/O	**SPI0 Clock Output** — S0CKO is a programmable SPI clock output. **GPIO** — PA12 is a general-purpose input/output signal. This pin defaults to GPIO input.
PA11/S0MDI	133	I/O	**SPI0 Master Data In** — S2MDI is an active-high serial data input for receiving data from SPI slave devices. **GPIO** — PA11 is a general-purpose input/output signal. This pin defaults to GPIO input.
PA10/S0MDO	132	I/O	**SPI0 Master Data Out** — S0MDO is an active-high serial output for transmitting data to SPI slave devices. **GPIO** — PA10 is a general-purpose input/output signal. This pin defaults to GPIO input.
PA[9:1], PA0	128-120, 118	I/O	**GPIO/Interrupt** — PA[9:0] are general-purpose input/output signals. When configured as inputs, bits PA[3:0] can be programmed to function as programmable level-sensitive or edge-sensitive interrupt inputs. When configured as outputs PA[9:0] can drive 12 mA.
PB[10:7]/ $\overline{S2S}$[7:4] PB[6:3]/ $\overline{S2S}$[3:0]	169-166 163-160	I/O	**SPI2 Slave Select** — $\overline{S2S}$[7:0] are active-low output signals used to select the target SPI slave device. **GPIO** — PB[10:3] are general-purpose input/output signals. These pins default to GPIO inputs.
PB2/S2CKO	159	I/O	**SPI2 Clock Output** — S2CKO is a programmable SPI clock output. **GPIO** — PB2 is a general-purpose input/output signal. This pin defaults to GPIO input.
PB1/S2MDI	158	I/O	**SPI2 Master Data In** — S2MDI is an active-high serial data input for receiving data from SPI slave devices. **GPIO** — PB1 is a general-purpose input/output signal. This pin defaults to GPIO input.
PB0/S2MDO	157	I/O	**SPI2 Master Data Out** — S2MDO is an active-high serial output for transmitting data to SPI slave devices. **GPIO** — PB0 is a general-purpose input/output signal. This pin defaults to GPIO input.

Note 1. All GPIO and SPI I/O pins have pull-ups and are not 5V-tolerant.

1.1.9 Miscellaneous Pins

Table 1-9. Miscellaneous Pins

Name	Location	Type	Description
TMS	170	I	**Test Mode Select Input** — Boundary scan input.
TDI	171	I	**Test Data Input** — Boundary scan input.
TDO	172	O	**Test Data Output** — Boundary scan output.
$\overline{\text{TRST}}$	173	I	**Test Reset** — Active-low boundary scan test reset input.
TCK	175	I	**Test Clock Input** — Boundary scan test clock input.
TEST_MODE	1	I	**Device Test Mode Input** — Active-high, 5V-tolerant pull-down production test input. This pin should normally be tied low.
$\overline{\text{PULL}}$	25	I	**Pull-Up/Pull-Down Enable** — Active-low input, enables pull-ups and pull-downs on inputs. This pin is normally tied to V_{SS}, but for power-sensitive applications it may be tied to V_{DD} to disconnect all pull-ups and pull-downs from associated input pins, reducing static power consumption.
PLLEN	207	I	**PLL Enable Input** — Active-high PLL enable Schmitt-trigger input. The PLL may be bypassed by tying this pin low; the system clock frequency will be half the external clock frequency. If the PLL is enabled it is required that PLLEN is driven low for a minimum of 500 ns following application of power.

1.2 VS2000 Functional Blocks

1.2.1 Cache Units

The VS2000 has internal instruction and data caches, each of 4K bytes. To raise the cache-hit ratio, the cache is configured using two-way, set-associative addressing. The replacement algorithm is LRU (Least-Recently Used). Each set uses a 12-bit tag address. The cache line size is four words (16 bytes). Valid, dirty and lock bits indicate the status of each cache line. The valid bit indicates the associated line has valid data, the dirty bit indicates that the line contains data that needs to be written to main memory and the lock bit indicates that the line cannot be replaced. When a miss occurs, four words must be fetched consecutively from external memory. Typically, RISC processors take advantage of instruction/data caches to improve performance. Without an instruction cache, bottlenecks that occur during instruction fetches from external memory may seriously degrade performance.

1.2.2 10/100 Ethernet MAC with DMA

The VS2000 has an Ethernet controller that operates at either 100 Mbits or 10 Mbits per second in half-duplex or full-duplex mode. The MII supplies the transmit and receive clocks of 25 MHz for 100 Mbits/second operation or 2.5 MHz at the 10 Mbits/second speed. An integrated DMA controller transfers data between the MII interface and frame buffers. The Ethernet MAC memory system contains a 1 KB transmit buffer and a 3 KB receive buffer. An external PHY is needed for the complete Ethernet solution.

1.2.3 Memory Controller

The memory controller generates the signals that control the access to external memory or peripheral devices. The unit is programmable and can address up to 32 MB of off-chip unified instruction/data memory. It has four chip selects and a 24-bit address bus. The external data bus can be configured to interface with 8- or 32-bit external devices. Separate read and write control signals allow for direct memory and peripheral interfacing.

1.2.4 Interrupt Controller

The VS2000 has a 16-level priority, individually-maskable, vectored interrupt controller. This feature substantially reduces the software and real-time overhead in handling internal and external interrupts. The 16-level priority encoder allows the customer to define the priority between the different interrupt sources. Internal sources are programmed to be level sensitive or edge triggered. External sources can be programmed to be positive- or negative-edge triggered or high- or low-level sensitive.

1.2.5 UARTs

The VS2000 provides two identical, full-duplex, universal synchronous/asynchronous receiver/transmitters that support standard features such as parity, framing and overrun error detection, line break generation and detection, interrupt generation and flow control. Each UART also includes a programmable baud-rate generator.

1.2.6 SPI Ports

The VS2000 provides three identical master-only SPI-compatible serial communication ports that directly supports a number of SPI operational features such as programmable frame transfer length, clock rate, clock polarity and clock phase. SPI slave devices typically include D/A and A/D converters, UARTs, memories and sensors. Each SPI port supports the direct connection of up to eight SPI slave devices.

1.2.7 Timers

The VS2000 features two timer/counter blocks, each containing three identical 32-bit timer/counter channels. Each channel can be independently programmed to perform a wide range of functions including event counting, interval measurement, pulse generation, delay timing and pulse width modulation. Each timer/counter channel has three external clock inputs, five internal clock inputs, and two multi-purpose input/output signals that can be configured by the user. Each channel drives an internal interrupt signal that can be programmed to generate processor interrupts via the interrupt controller.

1.2.8 Watchdog Timer

The Typhoon series microcontrollers have an internal watchdog timer that can be used to prevent system lock-up if the software becomes trapped in a deadlock. In normal operation the user reloads the watchdog at regular intervals before the timer timeout condition occurs. A watchdog timeout condition occurs when the watchdog control register has not been written to within a programmable timeout period. When a timeout condition occurs, the watchdog timeout signal is driven low for a duration of 16 system clock cycles. This output can be used by other devices within the system to reset or respond in an appropriate manner.

1.2.9 General-Purpose I/Os

The VS2000 has 87 programmable I/O lines. Each of the I/O lines is multiplexed with an external signal of a peripheral to optimize the use of available package pins. Six separate GPIO controllers control these lines. Eight of the GPIO pins also provide an external internal interrupt signal to the interrupt controller; the interrupt signals can be programmed as edge sensitive or level sensitive.

Chapter 2. Central Processing Unit

The Lightfoot 32-bit core features a hybrid 8-bit instruction path, 32-bit data path Harvard stack RISC architecture with a unique soft instruction set that allows application specific customization for virtual machine application environments such as Java or .NET. The memory referenced by programs falls into two categories: instruction memory and data memory. Instruction memory is eight bits wide and is used to store program instructions and constant data. Data memory is 32 bits wide and is byte addressable. Words (32-bit quantities) must be word-aligned, half-words (16-bit quantities) must be half-word aligned. The memory interface detects illegal accesses and signals a Bus Error trap.

The instruction and data memory interfaces each use 24-bit addresses allowing a total of 32 MB of off-core memory to be addressed. In addition to these interfaces, the Lightfoot core also features a 256 word, single-cycle 32-bit register interface that is used to access on-chip peripheral and core resources. The register port uses 8-bit addresses.

The Lightfoot core provides the following functional blocks:

- Control Unit
- ALU
- Data and Return Stacks
- Core Registers

2.1 Control Unit

The Control Unit is responsible for fetching, decoding and sequencing the execution of instructions in the processor. It also contains modules for implementing run-time checks and trap handling.

2.2 Arithmetic and Logic Unit

The ALU performs all the arithmetic and logical operations on data operands present in the data stack. The 32-bit ALU features a multi-cycle 32-bit barrel shifter and a 2-bit multiply step unit (allowing a 32 x 32-bit multiply to execute in 16 cycles) in addition to the usual arithmetic and logic capabilities.

2.3 Data And Return Stacks

The Data Stack plays the role of a register bank in traditional architectures. It consists of a hardware part (implemented as a bank of eight 32-bit core registers) and a memory extension unit. The memory extension unit is supported by a dedicated register (the EP, or Extension Pointer) together with a fill/spill circuit. The data stack is used to hold temporary data; it is not used to implement the stack frame for which special support is provided. The top elements of the Data Stack are coupled to the inputs of the ALU.

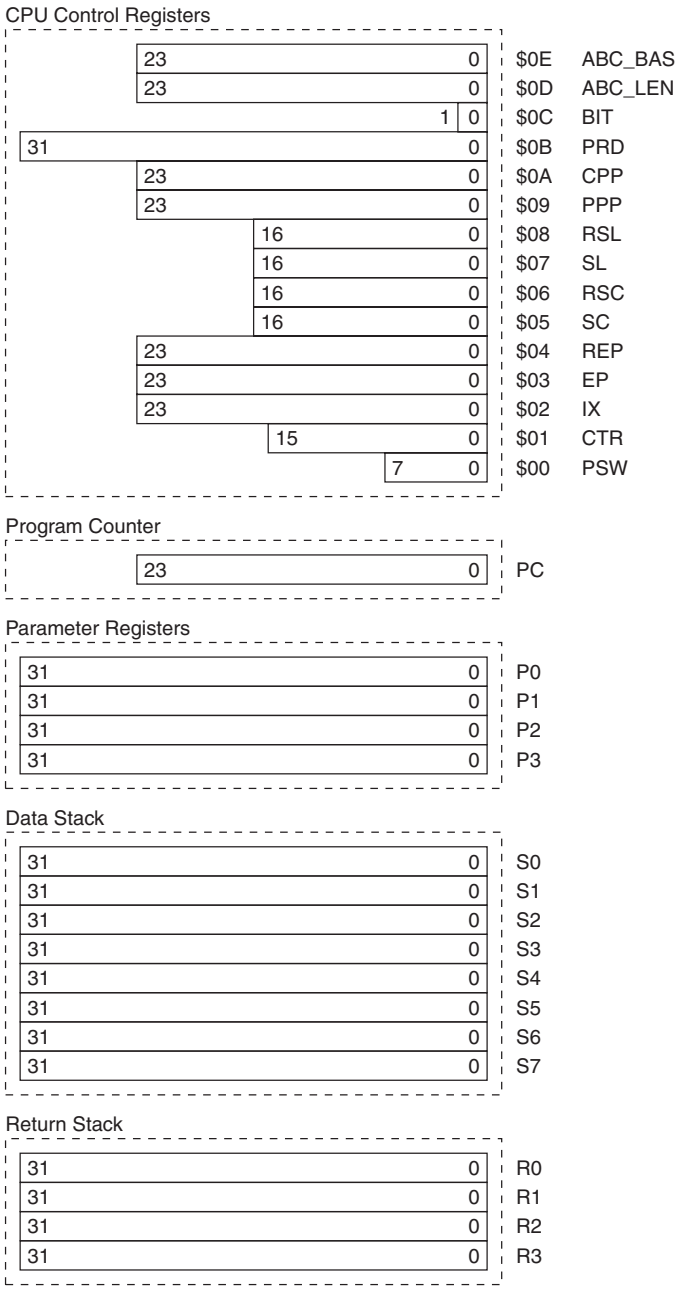
The Return Stack plays a threefold role in the processor: it holds return addresses for subroutines, its top-of-stack element is used as an index register to access program memory, and it can be used as an auxiliary stack for programs. The organization of the Return Stack is similar to the organization of the Data Stack, in that it also consists of a hardware part and a memory extension unit. The hardware part of the Return Stack consists of four 32-bit registers. The memory extension unit is supported by a dedicated CPU register (called the REP, Return Extension Pointer) and a fill/spill circuit.

2.4 Core Registers

Core registers fall into two categories, those that can be read or written via the 256-word register bus and those that can not. The bottom 32 address slots are reserved for core use, the remainder are available for interfacing to system peripherals such as memory management units or cryptographic co-processors. Of the 32 reserved slots, 15 are currently used as indicated in the figure below.

Address pointer registers: IX, EP, REP, CPP and PPP are always word-aligned, i.e. the two LSBs are always set to zero.

Figure 2-1. Lightfoot Programming Model



2.5 Instruction Set

The Lightfoot instruction set has been designed to allow highly-efficient execution of programs developed using object-oriented technologies such as Java, .NET, while still remaining highly flexible through the Lightfoot soft byte code mechanism.

The opcode field of a Lightfoot instruction is always eight bits wide; having three instruction formats which are:

- Soft Byte Code
- Fast Return
- Non-Returnable

A total of 128 of the possible 256 Lightfoot instructions are fixed instructions. These instructions comprise all Fast-Return and Non-Returnable instructions. The remaining 128 of the 256 possible Lightfoot instruction codes are user-configurable. These instructions are known as soft byte codes. Soft byte codes deliver the time efficiency of in-line code without sacrificing code density, making the processor ideal for creating efficient virtual machines (to support technologies such as Java), legacy instruction set emulation (8051) and application-specific instructions (for encryption support).

A number of Lightfoot instructions use operands which immediately follow the instruction opcode. These operands may be either eight or 16 bits wide, signed or unsigned.

2.5.1 Fast Return Instructions

A subset of 32 of the 128 fixed Lightfoot instructions which are single byte-only instructions (they do not require immediate data) and have Fast Return variants. Fast Return instructions permit zero overhead return from subroutines to be folded with their execution.

During the cycle after a Fast Return instruction is fetched from program memory, the Program counter takes on the value from the top of the Return Stack (RS0).

2.5.2 Non-Returnable Instructions

A subset of 64 of the 128 hard-coded Lightfoot instructions are non-returnable, i.e. their execution may not be folded with a subroutine return. These non-returnable instructions typically have immediate data or use the Return Stack during their execution.

2.5.3 Instruction Prefixes

A number of instructions may be prefixed with either WIDE or UNSGN instruction opcodes. These prefix opcodes alter how the instruction is executed by Lightfoot. The following subsections describe each of these prefix instruction opcodes.

2.5.3.1 WIDE Instruction

If an instruction, which uses an immediate operand is prefixed by the WIDE opcode, the immediate operand is taken to be 16 bits wide, with the most significant eight bits coming first followed by the least significant eight bits (big endian ordering). The resulting 16-bit value is interpreted as a signed or unsigned value, depending on the particular instruction.

2.5.3.2 UNSGN (Unsigned) Instruction

A number of instructions which normally use signed data in their execution may be forced to use unsigned data by prefixing their opcode (or associated WIDE opcode) with an UNSGN (unsigned) opcode.

There are three types of instruction which may use the UNSGN opcode, these include data memory byte and half-word loads, program memory immediate constant loads and branch condition comparisons which compare the top two Data Stack elements. The following summarizes the use of the UNSGN opcode with each of these instruction types:

- In the case of unsigned byte and half-word data memory loads, an UNSGN instruction prefix will prevent sign extension of the data loaded onto the top Data Stack as a result of the instruction execution. Instead, the data loaded onto the top Data Stack element is zero extended.
- In the case of unsigned program memory immediate constant loads, an UNSGN instruction prefix will prevent sign extension of constant data loaded onto the Data Stack. Instead, the data loaded onto the top Data Stack element is zero extended.
- In the case of branch condition comparisons, an UNSGN instruction prefix will force top two stack elements to be treated as 32-bit unsigned numbers. The branch offset values are not affected by the UNSGN instruction (they remain signed).

Instruction Byte Sequences

Lightfoot instructions (assuming that WIDE and UNSGN are part of an instruction) can thus be 8, 16, 24, 32 or 40 bits wide.

Instruction Set Summary

Table 2-1. Arithmetic and Logical Instructions

Mnemonic	Description	Mnemonic	Description
ADD	Add	ASR	Arithmetic shift right
ADC	Add with carry	ROT	Rotate left
AND	Bitwise AND	MUL	Multiplication step
IOR	Bitwise inclusive OR	NEG	Negate
XOR	Bitwise exclusive OR	CPL	One's complement
SUB	Subtract	INC	Increment
SBC	Subtract with carry	DEC	Decrement
SHL	Shift left	SEXB	Sign-extend byte
SHR	Shift right	SEXH	Sign-extend half-word

Table 2-2. Stack Manipulation Instructions

Mnemonic	Description	Mnemonic	Description
POP	Drop top data stack element	RROT	Right rotate top three data stack elements
DUP	Copy top data stack element	TOVER	Copy third data stack element to top
OVER	Copy second data stack element to top	RPOP	Move top return stack element to data stack
SWAP	Swap top two data stack elements	RPUSH	Move top data stack element to return stack
LROT	Left rotate top three data stack elements		

Table 2-3. Constant, Local Variable, Register and Parameter Register Instructions

Mnemonic	Description	Mnemonic	Description
CNSTI	Immediate constant	LP2	Load P2
CNST	Constant	LP3	Load P3
LP	Load parameter	SP0	Store P0
SP	Store parameter	SP1	Store P1
LPAR	Load parameter (direct)	SP2	Store P2
SPAR	Store parameter (direct)	SP3	Store P3
LPA	Load parameter address	LR	Load register
LP0	Load P0	SR	Store register
LP1	Load P1		

Table 2-4. Program and Data Memory Access Instructions

Mnemonic	Description	Mnemonic	Description
LIP	Load indexed program memory	LXB	Load indexed byte
SIP	Store indexed program memory	SXB	Store indexed byte
LW	Load word	LOW	Load offset word
SW	Store word	SOW	Store offset word
LH	Load half-word	LOH	Load offset half-word
SH	Store half-word	SOH	Store offset half-word
LB	Load byte	LOB	Load offset byte
SB	Store byte	SOB	Store offset byte
LWIX	Load word via index register	LDW	Load direct word
SWIX	Store word via index register	SDW	Store direct word
LXW	Load indexed word	LDH	Load direct half-word
SXW	Store indexed word	SDH	Store direct half-word
LXH	Load indexed half-word	LDB	Load direct byte
SXH	Store indexed half-word	SDB	Store direct byte

Table 2-5. Control Flow Instructions

Mnemonic	Description	Mnemonic	Description
DBNZ	Loop while counter register is greater than zero	BGEZ	Branch on greater than or equal to zero
BR	Unconditional branch	BLEZ	Branch on less than or equal to zero
JMP	Jump	BEQ	Branch on equal
BSR	Branch to subroutine	BNE	Branch on not equal
JSR	Jump to subroutine	BGT	Branch on greater than
BZ	Branch on zero	BLT	Branch on less than
BNZ	Branch on not zero	BGE	Branch on greater than or equal to
BGZ	Branch on greater than zero	BLE	Branch on less than or equal to
BLZ	Branch on less than zero		

Table 2-6. Stack Frame Support and Other Instructions

Mnemonic	Description	Mnemonic	Description
SSF	Set up stack frame	USGN	Unsigned prefix
PARS	Parameter store	WIDE	Wide prefix
REGS	Register store	NOP	No operation (return)

Chapter 3. Memory Maps

The VS2000 has three memory areas: instruction memory, data memory and register port. The Ethernet MAC unit is configured using the data memory interface, all other peripherals are configured using the register port.

Table 3-1. Register Port Memory Map

Register Port Address	Register Name	Width (bits)	Description	Reset Value
$00	PSW	8	Processor status word register, in-core	$00
$01	CTR	16	Counter register, in-core	$0000
$02	IX	24	Index register, in-core	$000000
$03	EP	24	Data stack extension pointer register, in-core	$000000
$04	REP	24	Return stack extension pointer register, in-core	$000000
$05	SC	17	Stack counter register, in-core	$1FFFC
$06	RSC	17	Return stack counter register, in-core	$1FFFF
$07	SL	17	Data stack limit register, in-core	$1FFFC
$08	RSL	17	Return stack limit register, in-core	$1FFFF
$09	PPP	24	Parameter pool pointer register, in-core	$000000
$0A	CPP	24	Constant pool pointer register, in-core	$000000
$0B	PRD	32	Product register, in-core	$00000000
$0C	BIT	2	Bit register, in-core	$000000
$0D	ABC_LEN	16	Array bounds check length register, in-core	$0000
$0E	ABC_BAS	24	Array bounds check base register, in-core	$000000
$0F-$1F	Reserved	n/a	Reserved for future in-core expansion	n/a
$20	TACSR0	8	Triple timer module A, timer 0 write command, read status register	$00
$21	TAMR0	4	Triple timer module A, timer 0 mode register	$0
$22	TAIOR0	4	Triple timer module A, timer 0 IO register	$4
$23	TACVR0	32	Triple timer module A, timer 0 count value register	$00000000
$24	TAPLR0	32	Triple timer module A, timer 0 pre-load register	$00000000

Note 1. Undefined at power-up, unmodified by reset.

2. Bits 11-0 are undefined, unmodfied by reset.

Table 3-1. Register Port Memory Map (continued)

Register Port Address	Register Name	Width (bits)	Description	Reset Value
$25	TAMCVR0	32	Triple timer module A, timer 0 middle count value register	$FFFFFF00
$26	TAECVR0	32	Triple timer module A, timer 0 end count value register	$FFFFFFFF
$27	TAMVR0	32	Triple timer module A, timer 0 measurement value register	$00000000
$28	TACSR1	32	Triple timer module A, timer 1 write command, read status register	$00
$29	TAMR1	32	Triple timer module A, timer 1 mode register	$0
$2A	TAIOR1	32	Triple timer module A, timer 1 IO register	$4
$2B	TACVR1	32	Triple timer module A, timer 1 count value register	$00000000
$2C	TAPLR1	32	Triple timer module A, timer 1 pre-load register	$00000000
$2D	TAMCVR1	32	Triple timer module A, timer 1 middle count value register	$FFFFFF00
$2E	TAECVR1	32	Triple timer module A, timer 1 end count value register	$FFFFFFFF
$2F	TAMVR1	32	Triple timer module A, timer 1 measurement value register	$00000000
$30	TACSR2	8	Triple timer module A, timer 2 write command, read status register	$00
$31	TAMR2	4	Triple timer module A, timer 2 mode register	$0
$32	TAIOR2	4	Triple timer module A, timer 2 IO register	$4
$33	TACVR2	32	Triple timer module A, timer 2 count value register	$00000000
$34	TAPLR2	32	Triple timer module A, timer 2 pre-load register	$00000000
$35	TAMCVR2	32	Triple timer module A, timer 2 middle count value register	$FFFFFF00
$36	TAECVR2	32	Triple timer module A, timer 2 end count value register	$FFFFFFFF
$37	TAMVR2	32	Triple timer module A, timer 2 measurement value register	$00000000
$38-$3E	Reserved	n/a	Reserved	n/a
$3F	TAISR	3	Triple timer module A interrupt status register	$0
$40	TBCSR0	8	Triple timer module B, timer 0 write command, read status register	$00

Note 1. Undefined at power-up, unmodified by reset.

2. Bits 11-0 are undefined, unmodfied by reset.

Table 3-1. Register Port Memory Map (continued)

Register Port Address	Register Name	Width (bits)	Description	Reset Value
$41	TBMR0	4	Triple timer module B, timer 0 mode register	$0
$42	TBIOR0	4	Triple timer module B, timer 0 IO register	$4
$43	TBCVR0	32	Triple timer module B, timer 0 count value register	$00000000
$44	TBPLR0	32	Triple timer module B, timer 0 pre-load register	$00000000
$45	TBMCVR0	32	Triple timer module B, timer 0 middle count value register	$FFFFFF00
$46	TBECVR0	32	Triple timer module B, timer 0 end count value register	$FFFFFFFF
$47	TBMVR0	32	Triple timer module B, timer 0 measurement value register	$00000000
$48	TBCSR1	8	Triple timer module B, timer 1 write command, read status register	$00
$49	TBMR1	4	Triple timer module B, timer 1 mode register	$0
$4A	TBIOR1	4	Triple timer module B, timer 1 IO register	$4
$4B	TBCVR1	32	Triple timer module B, timer 1 count value register	$00000000
$4C	TBPLR1	32	Triple timer module B, timer 1 pre-load register	$00000000
$4D	TBMCVR1	32	Triple timer module B, timer 1 middle count value register	$FFFFFF00
$4E	TBECVR1	32	Triple timer module B, timer 1 end count value register	$FFFFFFFF
$4F	TBMVR1	32	Triple timer module B, timer 1 measurement value register	$00000000
$50	TBCSR2	8	Triple timer module B, timer 2 write command, read status register	$00
$51	TBMR2	4	Triple timer module B, timer 2 mode register	$0
$52	TBIOR2	4	Triple timer module B, timer 2 IO register	$4
$53	TBCVR2	32	Triple timer module B, timer 2 count value register	$00000000
$54	TBPLR2	32	Triple timer module B, timer 2 pre-load register	$00000000
$55	TBMCVR2	32	Triple timer module B, timer 2 middle count value register	$FFFFFF00
$56	TBECVR2	32	Triple timer module B, timer 2 end count value register	$FFFFFFFF

Note 1. Undefined at power-up, unmodified by reset.

2. Bits 11-0 are undefined, unmodfied by reset.

Table 3-1. Register Port Memory Map (continued)

Register Port Address	Register Name	Width (bits)	Description	Reset Value
$57	TBMVR2	32	Triple timer module B, timer 2 measurement value register	$00000000
$58-$5E	Reserved	n/a	Reserved	n/a
$5F	TBISR	3	Triple timer module B interrupt status register	$0
$60-$6F	Reserved	n/a	Reserved	n/a
$70	ICCSR	32	Instruction cache write control/read status register	$00800xxx(2)
$71	DCCSR	32	Data cache write control/read status register	$00800xxx(2)
$72	WDTCSR	8	Watchdog timer write command/read status register	$80
$73	WDTCR	32	Watchdog timer count value register	$00000000
$74	Reserved	n/a	Reserved	n/a
$75	Reserved	n/a	Reserved	n/a
$76	Reserved	n/a	Reserved	n/a
$77	Reserved	n/a	Reserved	n/a
$78	PACR	32	Port A I/O configuration register	$00000000
$79	PBCR	11	Port B I/O configuration register	$000
$7A	PCCR	6	Port C I/O configuration register	$00
$7B	PDCR	6	Port D I/O configuration register	$00
$7C	PECR	8	Port E I/O configuration register	$00
$7D	PFCR	24	Port F I/O configuration register	$000000
$7E-$7F	Reserved	n/a	Reserved	n/a
$80	UDR0	8	UART0 write transmit/read receive data register or divisor low	$00
$81	UIER0	8	UART0 interrupt enable register or divisor high	$00
$82	UIIR0	8	UART0 write FIFO control/read interrupt identification register	$C1
$83	ULCR0	8	UART0 line control register	$03
$84	UMCR0	8	UART0 modem control register	$00
$85	ULSR0	8	UART0 line status register	$60
$86	UMSR0	8	UART0 modem status register	$00

Note 1. Undefined at power-up, unmodified by reset.

2. Bits 11-0 are undefined, unmodfied by reset.

Table 3-1. Register Port Memory Map (continued)

Register Port Address	Register Name	Width (bits)	Description	Reset Value
$87	Reserved	n/a	Reserved	n/a
$88	UDR1	8	UART1 write transmit/read receive data register or divisor low	$00
$89	UIER1	8	UART1 interrupt enable register or divisor high	$00
$8A	UIIR1	8	UART1 write FIFO control/read interrupt identification register	$C1
$8B	ULCR1	8	UART1 line control register	$03
$8C	UMCR1	8	UART1 modem control register	$00
$8D	ULSR1	8	UART1 line status register	$60
$8E	UMSR1	8	UART1 modem status register	$00
$8F-$9F	Reserved	n/a	Reserved	n/a
$A0	PADR	32	Port A data register	$00000000
$A1	PADDR	32	Port A data direction register	$FFFFFFFF
$A2	PAICR	16	Port A interrupt control register	$0000
$A3	PAISR	4	Port A interrupt status register	$0
$A4	PBDR	11	Port B data register	$000
$A5	PBDDR	11	Port B data direction register	$7FF
$A6	PBICR	16	Port B interrupt control register	$0000
$A7	PBISR	4	Port B interrupt status register	$0
$A8	PCDR	6	Port C data register	$00
$A9	PCDDR	6	Port C data direction register	$3F
$AA	PDDR	6	Port D data register	$00
$AB	PDDDR	6	Port D data direction register	$3F
$AC	PEDR	8	Port E data register	$00
$AD	PEDDR	8	Port E data direction register	$FF
$AE	PFDR	24	Port F data register	$000000
$AF	PFDDR	24	Port F data direction register	$FFFFFF
$B0	SCSR0	8	SPI 0 write control/read status register	$00
$B1	SSLVC0	16	SPI 0 slave configuration register	$0208

Note 1. Undefined at power-up, unmodified by reset.

2. Bits 11-0 are undefined, unmodfied by reset.

Table 3-1. Register Port Memory Map (continued)

Register Port Address	Register Name	Width (bits)	Description	Reset Value
$B2	SWD0	16	SPI 0 write data register	$0000
$B3	SRD0	16	SPI 0 read data register	$0000
$B4	SBPR0	16	SPI 0 bit rate count preset register	$000F
$B5-$B7	Reserved	n/a	Reserved	n/a
$B8	SCSR1	8	SPI 1 write control/read status register	$00
$B9	SSLVC1	16	SPI 1 slave configuration register	$0208
$BA	SWD1	16	SPI 1 write data register	$0000
$BB	SRD1	16	SPI 1 read data register	$0000
$BC	SBPR1	16	SPI 1 bit rate count preset register	$000F
$BD-$BF	Reserved	n/a	Reserved	n/a
$C0	SCSR2	8	SPI 2 write control/read status register	$00
$C1	SSLVC2	16	SPI 2 slave configuration register	$0208
$C2	SWD2	16	SPI 2 write data register	$0000
$C3	SRD2	16	SPI 2 read data register	$0000
$C4	SBPR2	16	SPI 2 bit rate count preset register	$000F
$C5-$DF	Reserved	n/a	Reserved	n/a
$E0	Z0CFG	24	Zone 0 configuration register	$027F00
$E1	Z0TMR	32	Zone 0 timers configuration register	$00080004
$E2	Z1CFG	24	Zone 1 configuration register	$030000
$E3	Z1TMR	32	Zone 1 timers configuration register	$00010000
$E4	Z2CFG	24	Zone 2 configuration register	$030000
$E5	Z2TMR	32	Zone 2 timers configuration register	$00010000
$E6	Z3CFG	24	Zone 3 configuration register	$030000
$E7	Z3TMR	32	Zone 3 timers configuration register	$00010000
$E8-$EF	Reserved	n/a	Reserved	n/a
$F0	IPL0	32	Interrupt priority vector assignment register 0	$76543210
$F1	IPL1	32	Interrupt priority vector assignment register 1	$FEDCBA98
$F2	IMASK	16	Interrupt mask register	$0000
$F3	ICFG0	16	Interrupt configuration register 0	$FFFF

Note 1. Undefined at power-up, unmodified by reset.

2. Bits 11-0 are undefined, unmodfied by reset.

Table 3-1. Register Port Memory Map (continued)

Register Port Address	Register Name	Width (bits)	Description	Reset Value
$F4	ICFG1	16	Interrupt configuration register 1	$FFFF
$F5	ICLR	16	Interrupt clear register	$0000
$F6	IFLAG	16	Interrupt flag register	$0000
$F7-$FE	Reserved	n/a	Reserved	n/a
$FF	DVR	32	Device version register	$01020001

Note 1. Undefined at power-up, unmodified by reset.

2. Bits 11-0 are undefined, unmodfied by reset.

Data memory is divided into three areas: cached, direct mapped and Ethernet peripheral.

Table 3-2. Data Memory Map

Memory Address	Register Name	Width (bits)	Description	Reset Value
$000000-$7FFFFF	n/a	8/32	8 MB of cached memory area	n/a
$800000-$FFEFFF	n/a	8/32	direct mapped external memory	n/a
$FF0000	EMODER	32	eMAC mode register	$0000A800
$FF0004	EISOURCE	32	eMAC interrupt source register	$00000000
$FF0008	EIMASK	32	eMAC interrupt mask register	$00000000
$FF000C	EIPGT	32	eMAC b2b inter-packet gap register	$00000012
$FF0010	EIPGR1	32	eMAC non-b2b inter-packet gap register 0	$0000000C
$FF0014	EIPGR2	32	eMAC non-b2b inter-packet gap register 1	$00000012
$FF0018	EPKLEN	32	eMAC packet length register	$003C0600
$FF001C	ECOLLCFG	32	eMAC collision & retry configuration register	$000F003F
$FF0020	ERXBDNUM	32	eMAC transmit buffer descriptor number register	$00000080
$FF0024	ECTLMR	32	eMAC control module mode register	$00000000
$FF0028	EMIIMR	32	eMAC MII mode register	$00000064
$FF002C	EMIICR	32	eMAC MII command register	$00000000
$FF0030	EMIIADR	32	eMAC MII address register	$00000000
$FF0034	EMIITDR	32	eMAC MII transmit data register	$00000000
$FF0038	EMIIRDR	32	eMAC MII receive data register	$00000000
$FF003C	EMIISR	32	eMAC MII status register	$00000000
$FF0040	EMADR0	32	eMAC MAC address register 0	$00000000
$FF0044	EMADR1	32	eMAC MAC address register 1	$00000000

Table 3-2. Data Memory Map (continued)

Memory Address	Register Name	Width (bits)	Description	Reset Value
$FF0048	EHASH0	32	eMAC hash register 0	$00000000
$FF004C	EHASH1	32	eMAC hash register 1	$00000000
$FF0050-$FF03FF	Reserved	n/a	Reserved	n/a
$FF0400-$FF07FC	n/a	32	256-word single-port Ethernet buffer descriptor memory	n/a
$FF0800-$FF0FFF	n/a	32	Reserved	n/a
$FF1000-$FF13FF	n/a	32	1 kB dual-port Ethernet transmit buffer memory	n/a
$FF1400-$FF1FFF	n/a	32	3 kB dual-port Ethernet receive buffer memory	n/a
$FF2000-$FFFFFF	Reserved	n/a	Reserved	n/a

VS2000 instruction memory is divided into two areas: direct mapped and cached.

Table 3-3. Program Memory Map

Memory Address	Width (bits)	Description
$000000-$7FFFFF	8	8 MB cached memory area
$800000-$FFFFFF	8	8 MB direct mapped external memory

Chapter 4. Clock System

The VS2000 is a fully-synchronous device which uses a single system-clock signal to synchronously perform all internal CPU and peripheral transfers.

There are two additional clocks used in the VS2000: the real-time clock counter input clock and the Ethernet MAC MII clock. Synchronization between these additional clock inputs and the VS2000 system clock is performed in the respective blocks.

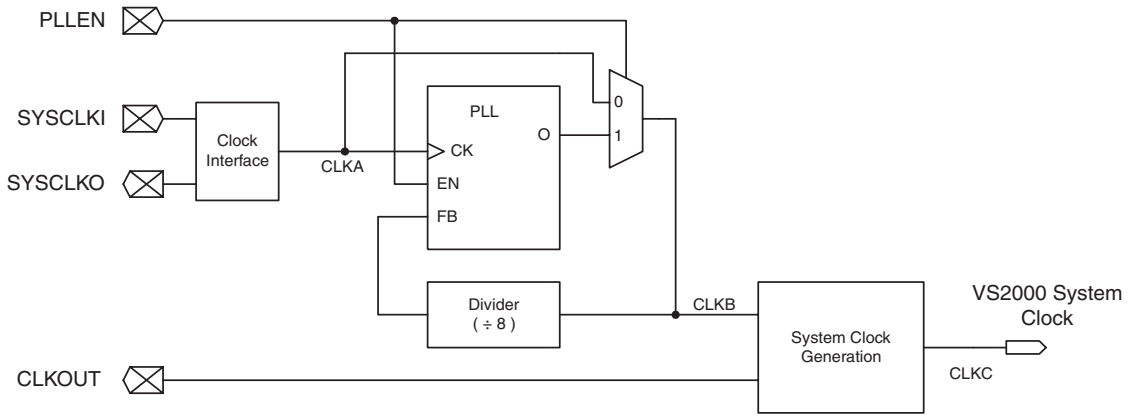
The Ethernet MAC MII clock interface is described in Chapter 12, "Ethernet MAC".

4.1 Internal System Clock

The VS2000 has a single system-clock signal used to clock the VS2000 CPU core and all associated peripheral and control blocks. The system clock frequency is configurable depending upon the user's performance and power consideration requirements. An internal clock multiplication circuit provides a means of clocking the VS2000 with a lower rate external clock using either a direct crystal connection or oscillator output. The clock multiplication circuit includes a phase-lock loop (PLL) block which can be enabled or disabled (bypassed) as required.

When enabled, the PLL provides external clock-source frequency multiplication resulting in the system clock frequency being four times the input clock frequency. The PLL itself multiplies the external clock input frequency by eight and outputs this clock signal (CLKB) to the system clock generation circuit. The system clock generation circuit divides CLKB by two to produce the 4x system clock frequency signal (CLKC) which is buffered to the entire VS2000 device.

Figure 4-1. VS2000 System Clock Derivation



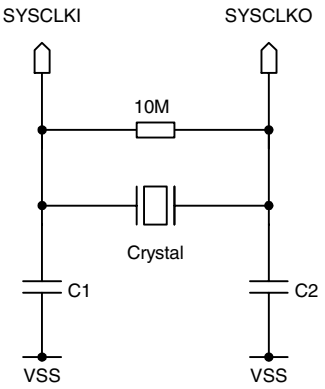
4.1.1 VS2000 Clock Input Interface

The VS2000 clock input interface allows direct connection to either a crystal or oscillator device output via the external SYSCLKI and SYSCLKO pins.

If an oscillator output is to be used to clock the device, it should be connected directly to the SYSCLKI pin. In this case, the SYSCLKO pin should be left unconnected. The input frequency range when driving the SYSCLKI input using an oscillator output is DC to 80 MHz.

If a crystal is to be used, it should be connected as shown in Figure 4-2. The values of C1 and C2 should be as per the crystal manufacturers recommendations. A 10 MΩ resistor is required to be connected between SYSCLKI and SYSCLKO. The frequency range for the crystal device is 1 MHz to 20 MHz.

Figure 4-2. Crystal-Controlled Clock Input

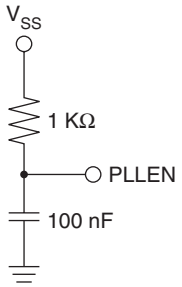


4.1.2 Phase-Lock Loop

The PLL is enabled when the device's PLLEN pin is tied to V_{DD}. If the PLL is not required, it should be disabled by connecting PLLEN to V_{SS} (0V).

When enabled, the PLL requires a maximum of 600 µs from power-on to achieve lock. In addition to this, the PLLEN input pin must be driven low for a minimum of 500 ns from power-on. The PLLEN pin is a schmitt input, and therefore, the following circuit may be used to implement this function.

Figure 4-3. PLLEN RC Network



If the PLL is disabled, the resulting system clock frequency will equal the external clock frequency divided by two.

It is not recommended that the PLLEN signal changes state while the device is in an operating (non-reset) state. This may cause erroneous behavior as the clock switches between sources.

4.1.3 Clock Output Pin

The VS2000 Clock Output (CLKOUT) pin is used to output a copy of the internal system clock signal. The CLKOUT signal itself has a defined phase relationship with the internal system clock and can be used by external devices (such as synchronous memories or peripherals) for synchronization or timing purposes.

Chapter 5. General-Purpose I/O Ports

The VS2000 has six general-purpose I/O ports: A, B, C, D, E and F. Ports A, B, C, D, E share pins with other peripherals. The Pad MUX controller is used to select between GPIO and Peripheral functionality for each I/O pin on the VS2000. The Port F pins are shared with the upper 24 bits of the data bus and can only be used as GPIO when the VS2000 is running in 8-bit mode.

Table 5-1. General-Purpose I/O Ports

Port	Primary Function	Secondary Function
A	32-Bit I/O Port	SPI0 / SPI1
B	11-Bit I/O Port	SPI2
C	6-Bit I/O Port	TCA
D	6-Bit I/O Port	TCB
E	8-Bit I/O Port	UART1 / UART2
F	24-Bit I/O Port (8-bit mode)	Data Bus[31:8] (32-bit mode)

All general purpose I/O ports have a data register and a data direction register. Ports A and B also have an interrupt control register and an interrupt status register. The data register drives output pins and latches input signals. The data direction register configures each pin as either input or output. The interrupt control register enables a GPIO pin as an interrupt, programs its sensitivity and allows it to be masked. The interrupt status register shows the state of the interrupt line and is used to clear an edge detected interrupt.

5.1 Operation

The GPIO ports all have standard GPIO pins. Ports A and B each have four pins which can be used as external interrupts. The interrupt-programmable pins therefore have two modes of operation set by the status of the interrupt enable bit for that pin in the interrupt control register.

The two modes of operation are:

- Standard GPIO operation
- Interrupt operation

5.1.1 Standard GPIO Operation

When the interrupt enable bit is set to "0" the pin behaves as a standard GPIO pin. Writing to the direction register sets the pin as an input or an output. Writing the data register sets the state of all output pins. Reading from the data register samples the input pins and for output pins will return the last value written to them.

5.1.2 External Interrupt Operation

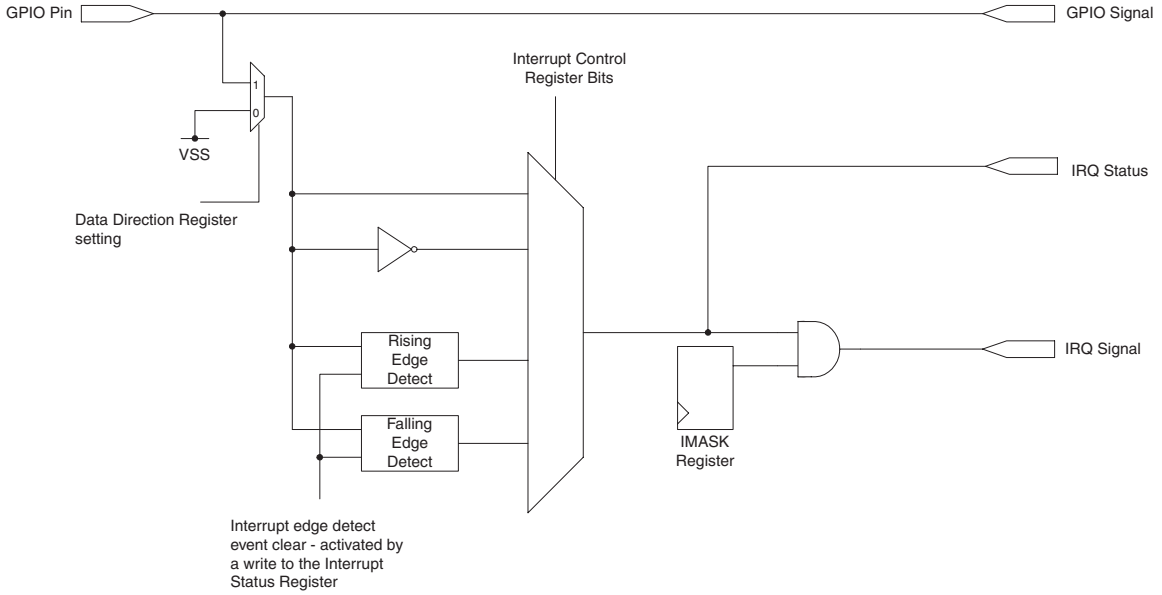
When configured as an external interrupt, by setting the interrupt enable bit to "1", the pin is automatically configured as an input. Note that if the interrupt enable pin is subsequently set to "0" the pin will remain configured as an input.

The type of interrupt that the external interrupt pin will respond to can be configured. The four types are:

1. Rising edge sensitive
2. Falling edge sensitive
3. Active high level sensitive
4. Active low level sensitive

The interrupt type is configured by the two most-significant bits of each of the four interrupt control nibbles.

Figure 5-1. 4-to-1 Multiplexed Interrupt Generator



The interrupt type is selected using a 4-to-1 multiplexer to generate a single, level-sensitive, active-high interrupt from the interrupt source specified. Edge-sensitive interrupts are latched and must be cleared by the interrupt service routine by writing to the interrupt status register.

Note: If the interrupt pin is disabled (set as standard GPIO) no edges will be latched by the hardware above.

The IRQ signal for the GPIO port is given by the logical OR of the four post-mask interrupt signals. The interrupt status register allows system software to examine which interrupts are active.

5.2 GPIO Register Set

5.2.1 Data Registers

The data register holds the current state of the input pins and the value to be written to the output pins. A read of an input pin returns the state of the pin. A read of an output pin returns the state of the correspond-

ing device output pin. A write to a pin which is configured as an input, interrupt, or peripheral has no effect on the state of the pin.

5.2.1.1 PADR Register

Table 5-2. PADR Register – Bits 31-24

Address	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24	Register Name
$A0	PA31	PA30	PA29	PA28	PA27	PA26	PA25	PA24	PADR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	0	0	0	

Table 5-3. PADR Register – Bits 23-16

Address	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16	Register Name
$A0	PA23	PA22	PA21	PA20	PA19	PA18	PA17	PA16	PADR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	0	0	0	

Table 5-4. PADR Register – Bits 15-8

Address	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Register Name
$A0	PA15	PA14	PA13	PA12	PA11	PA10	PA9	PA8	PADR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	0	0	0	

Table 5-5. PADR Register – Bits 7-0

Address	Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register Name
$A0	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0	PADR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	0	0	0	

5.2.1.2 PBDR Register

Table 5-6. PBDR Register – Bits 15-8

Address	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Register Name
$A4	-	-	-	-	-	PB10	PB9	PB8	PBDR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	1	1	1	
Peripheral	–	–	–	–	–	S2S7	S2S6	S2S5	

Table 5-7. PBDR Register – Bits 7-0

Address	Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register Name
$A4	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	PBDR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	0	0	0	
Peripheral	S2S4	S2S3	S2S2	S2S1	S2S0	S2CKO	S2MDI	S2MDO	

5.2.1.3 PCDR Register

Table 5-8. PCDR Register

Address	Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register Name
$A8	-	-	PC5	PC4	PC3	PC2	PC1	PC0	PCDR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	0	0	0	
Peripheral	–	–	TCAO2	TCAO1	TCAO0	TCAI2	TCAI1	TCAI0	

5.2.1.4 PDDR Register

Table 5-9. PDDR Register

Address	Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register Name
$AA	-	-	PD5	PD4	PD3	PD2	PD1	PD0	PDDR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	0	0	0	
Peripheral	–	–	TCBO2	TCBO1	TCBO0	TCBI2	TCBI1	TCBI0	

5.2.1.5 PEDR Register

Table 5-10. PEDR Register

Address	Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register Name
$AC	PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0	PEDR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	0	0	0	
Peripheral	CTS1	CTS0	RTS1	RTS0	RXD1	RXD0	TXD1	TXD0	

5.2.1.6 PFDR Register

Table 5-11. PFDR Register – Bits 23-16

Address	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16	Register Name
$AE	PF23	PF22	PF21	PF20	PF19	PF18	PF17	PF16	PFDR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	0	0	0	
Data Bus	D31	D30	D29	D28	D27	D26	D25	D24	

Table 5-12. PFDR Register – Bits 15-8

Address	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Register Name
$AE	PF15	PF14	PF13	PF12	PF11	PF10	PF9	PF8	PFDR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	0	0	0	
Data Bus	D23	D22	D21	D20	D19	D18	D17	D16	

Table 5-13. PFDR Register – Bits 7-0

Address	Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register Name
$AE	PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0	PFDR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	0	0	0	
Data Bus	D15	D14	D13	D12	D11	D10	D9	D8	

5.2.2 Data Direction Registers

The direction register configures the direction of each pin of the general purpose I/O port.

For all bits:

0 = Output

1 = Input

On reset all general purpose I/O pins default to input.

5.2.2.1 PADDR Register

Table 5-14. PADDR Register – Bits 31-24

Address	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24	Register Name
$A1	DDA31	DDA30	DDA29	DDA28	DDA27	DDA26	DDA25	DDA24	PADDR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	1	1	1	1	1	1	1	1	

Table 5-15. PADDR Register – Bits 23-16

Address	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16	Register Name
$A1	DDA23	DDA22	DDA21	DDA20	DDA19	DDA18	DDA17	DDA16	PADDR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	1	1	1	1	1	1	1	1	

Table 5-16. PADDR Register – Bits 15-8

Address	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Register Name
$A1	DDA15	DDA14	DDA13	DDA12	DDA11	DDA10	DDA9	DDA8	PADDR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	1	1	1	1	1	1	1	1	

Table 5-17. PADDR Register – Bits 7-0

Address	Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register Name
$A1	DDA7	DDA6	DDA5	DDA4	DDA3	DDA2	DDA1	DDA0	PADDR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	1	1	1	1	1	1	1	1	

5.2.2.2 PBDDR Register

Table 5-18. PBDDR Register – Bits 15-8

Address	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Register Name
$A5	-	-	-	-	-	DDB10	DDB9	DDB8	PBDDR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	1	1	1	

Table 5-19. PBDDR Register – Bits 7-0

Address	Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register Name
$A5	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0	PBDDR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	1	1	1	1	1	1	

5.2.2.3 PCDDR Register

Table 5-20. PCDDR Register

Address	Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register Name
$A7	-	-	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0	PCDDR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	1	1	1	1	1	1	

5.2.2.4 PDDDR Register

Table 5-21. PDDDR Register

Address	Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register Name
$AB	-	-	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	PDDDR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	1	1	1	1	1	1	

5.2.2.5 PEDDR Register

Table 5-22. PEDDR Register

Address	Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register Name
$AD	DDE7	DDE6	DDE5	DDE4	DDE3	DDE2	DDE1	DDE0	PEDDR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	1	1	1	1	1	1	1	1	

5.2.2.6 PFDDR Register

Table 5-23. PFDDR Register – Bits 23-16

Address	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16	Register Name
$AF	DDF23	DDF22	DDF21	DDF20	DDF19	DDF18	DDF17	DDF16	PFDDR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	1	1	1	1	1	1	1	1	

Table 5-24. PFDDR Register – Bits 15-8

Address	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Register Name
$AF	DDF15	DDF14	DDF13	DDF12	DDF11	DDF10	DDF9	DDF8	PFDDR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	1	1	1	1	1	1	1	1	

Table 5-25. PFDDR Register – Bits 7-0

Address	Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register Name
$AF	DDF7	DDF6	DDF5	DDF4	DDF3	DDF2	DDF1	DDF0	PFDDR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	1	1	1	1	1	1	1	1	

5.2.3 Interrupt Control Registers

The interrupt control registers configure up to four of the least significant pins of general purpose I/O ports A and B as external interrupts.

ICS - Interrupt Control Sensitivity

The Interrupt Control Sensitivity bits are used to select the type of input the interrupt will be sensitive to. The input can be either edge or level sensitive.

0 = Edge sensitive

1 = Level sensitive

ICP - Interrupt Control Polarity

The Interrupt Control Polarity bits are used to select the input signal polarity. This setting refers to either high or low levels for level sensitive inputs, or rising or falling edges for edge sensitive inputs.

0 = High level/rising edge

1 = Low level/falling edge

ICE - Interrupt Control Enable

The Interrupt Control Enable bits are used to enable and disable the interrupt operation for the associated bit.

0 = Disable interrupt

1 = Enable interrupt

ICM - Interrupt Control Mask

The Interrupt Control Mask bits are used to mask interrupt events.

0 = Mask interrupt (prevent interrupt from being detected)

1 = Unmask interrupt (allow interrupt to be detected)

On reset all interrupt pins are by default disabled, masked and initially set as rising edge type.

5.2.3.1 PAICR Register

Table 5-26. PAICR Register – Bits 15-8

Address	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Register Name
$A2	ICSA3	ICPA3	ICEA3	ICMA3	ICSA2	ICPA2	ICEA2	ICMA2	PAICR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	0	0	0	

Table 5-27. PAICR Register – Bits 7-0

Address	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Register Name
$A2	ICSA1	ICPA1	ICEA1	ICMA1	ICSA0	ICPA0	ICEA0	ICMA0	PAICR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	0	0	0	

5.2.3.2 PBICR Register

Table 5-28. PBICR Register – Bits 15-8

Address	Bit 15	Bit 14	Bit 13	Bit12	Bit11	Bit10	Bit9	Bit8	Register Name
$A6	ICSB3	ICPB3	ICEB3	ICMB3	ICSB2	ICPB2	ICEB2	ICMB2	PBICR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	0	0	0	

Table 5-29. PBICR Register – Bits 7-0

Address	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Register Name
$A6	ICSB1	ICPB1	ICEB1	ICMB0	ICSB0	ICPB0	ICEB0	ICMB0	PBICR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	0	0	0	

5.2.4 Interrupt Status Registers

The interrupt status register performs two functions. Reading the register shows the state of each interrupt. Writing a "1" to any bit of the register clears the corresponding latched edge detected interrupt.

Read Mode:

0 = Interrupt un-triggered

1 = Interrupt triggered

Write Mode:

0 = No effect

1 = Edge detected interrupt cleared

5.2.4.1 PAISR Register

Table 5-30. PAISR Register

Address	Bit 3	Bit 2	Bit 1	Bit 0	Register Name
$A3	ISA3	ISA2	ISA1	ISA0	PAISR
Read/Write	R	R	R	R	
Reset	0	0	0	0	

5.2.4.2 PBISR Register

Table 5-31. PBISR Register

Address	Bit 3	Bit 2	Bit 1	Bit 0	Register Name
$A7	ISB3	ISB2	ISB1	ISB0	PBISR
Read/Write	R	R	R	R	
Reset	0	0	0	0	

Chapter 6. I/O Pin Controller

The VS2000 has six ports: A, B, C, D, E and F. Each port represents a group of pins with a common peripheral function. The pins of each port are multiplexed between general purpose I/O and VS2000 peripherals. The pin functions for each port are described in the tables below.

Table 6-1. Port A Pin Function Definitions

VS2000 Pin	Function		
	GPIO Bit	Peripheral	Peripheral Direction
PA0	Port A [0]	None - GPIO only	-
PA1	Port A [1]	None - GPIO only	-
PA2	Port A [2]	None - GPIO only	-
PA3	Port A [3]	None - GPIO only	-
PA4	Port A [4]	None - GPIO only	-
PA5	Port A [5]	None - GPIO only	-
PA6	Port A [6]	None - GPIO only	-
PA7	Port A [7]	None - GPIO only	-
PA8	Port A [8]	None - GPIO only	-
PA9	Port A [9]	None - GPIO only	-
PA10/S0MDO	Port A [10]	SPI 0 master data output	O
PA11/S0MDI	Port A [11]	SPI 0 master data input	I
PA12/S0CKO	Port A [12]	SPI 0 clock output	O
PA13/S0S0	Port A [13]	SPI 0 chip select output 0 (active low)	O
PA14/S0S1	Port A [14]	SPI 0 chip select output 1 (active low)	O
PA15/S0S2	Port A [15]	SPI 0 chip select output 2 (active low)	O
PA16/S0S3	Port A [16]	SPI 0 chip select output 3 (active low)	O
PA17/S0S4	Port A [17]	SPI 0 chip select output 4 (active low)	O
PA18/S0S5	Port A [18]	SPI 0 chip select output 5 (active low)	O
PA19/S0S6	Port A [19]	SPI 0 chip select output 6 (active low)	O
PA20/S0S7	Port A [20]	SPI 0 chip select output 7 (active low)	O
PA21/S1MDO	Port A [21]	SPI 1 master data output	O
PA22/S1MDI	Port A [22]	SPI 1 master data input	I

Table 6-1. Port A Pin Function Definitions (continued)

VS2000 Pin	Function		
	GPIO Bit	**Peripheral**	**Peripheral Direction**
PA23/S1CKO	Port A [23]	SPI 1 clock output	O
PA24/S1S0	Port A [24]	SPI 1 chip select output 0 (active low)	O
PA25/S1S1	Port A [25]	SPI 1 chip select output 1 (active low)	O
PA26/S1S2	Port A [26]	SPI 1 chip select output 2 (active low)	O
PA27/S1S3	Port A [27]	SPI 1 chip select output 3 (active low)	O
PA28/S1S4	Port A [28]	SPI 1 chip select output 4 (active low)	O
PA29/S1S5	Port A [29]	SPI 1 chip select output 5 (active low)	O
PA30/S1S6	Port A [30]	SPI 1 chip select output 6 (active low)	O
PA31/S1S7	Port A [31]	SPI 1 chip select output 7 (active low)	O

Table 6-2. Port B Pin Function Definitions

VS2000 Pin	Function		
	GPIO Bit	**Peripheral**	**Peripheral Direction**
PB0/S2MDO	Port B [0]	SPI 2 master data output	O
PB1/S2MDI	Port B [1]	SPI 2 master data input	I
PB2/S2CKO	Port B [2]	SPI 2 clock output	O
PB3/S2S0	Port B [3]	SPI 2 chip select output 0 (active low)	O
PB4/S2S1	Port B [4]	SPI 2 chip select output 1 (active low)	O
PB5/S2S2	Port B [5]	SPI 2 chip select output 2 (active low)	O
PB6/S2S3	Port B [6]	SPI 2 chip select output 3 (active low)	O
PB7/S2S4	Port B [7]	SPI 2 chip select output 4 (active low)	O
PB8/S2S5	Port B [8]	SPI 2 chip select output 5 (active low)	O
PB9/S2S6	Port B [9]	SPI 2 chip select output 6 (active low)	O
PB10/S2S7	Port B [10]	SPI 2 chip select output 7 (active low)	O

Table 6-3. Port C Pin Function Definitions

VS2000 Pin	Function		
	GPIO Bit	**Peripheral**	**Peripheral Direction**
PC0/TCAI0	Port C [0]	Timer/Counter A input 0	I
PC1/TCAI1	Port C [1]	Timer/Counter A input 1	I
PC2/TCAI2	Port C [2]	Timer/Counter A input 2	I
PC3/TCAO0	Port C [3]	Timer/Counter A output 0	O
PC4/TCAO1	Port C [4]	Timer/Counter A output 1	O
PC5/TCAO2	Port C [5]	Timer/Counter A output 2	O

Table 6-4. Port D Pin Function Definitions

VS2000 Pin	Function		
	GPIO Bit	**Peripheral Function**	**Peripheral Direction**
PD0/TCBI0	Port D [0]	Timer/Counter B input 0	I
PD1/TCBI1	Port D [1]	Timer/Counter B input 1	I
PD2/TCBI2	Port D [2]	Timer/Counter B input 2	I
PD3/TCBO0	Port D [3]	Timer/Counter B output 0	O
PD4/TCBO1	Port D [4]	Timer/Counter B output 1	O
PD5/TCBO2	Port D [5]	Timer/Counter B output 2	O

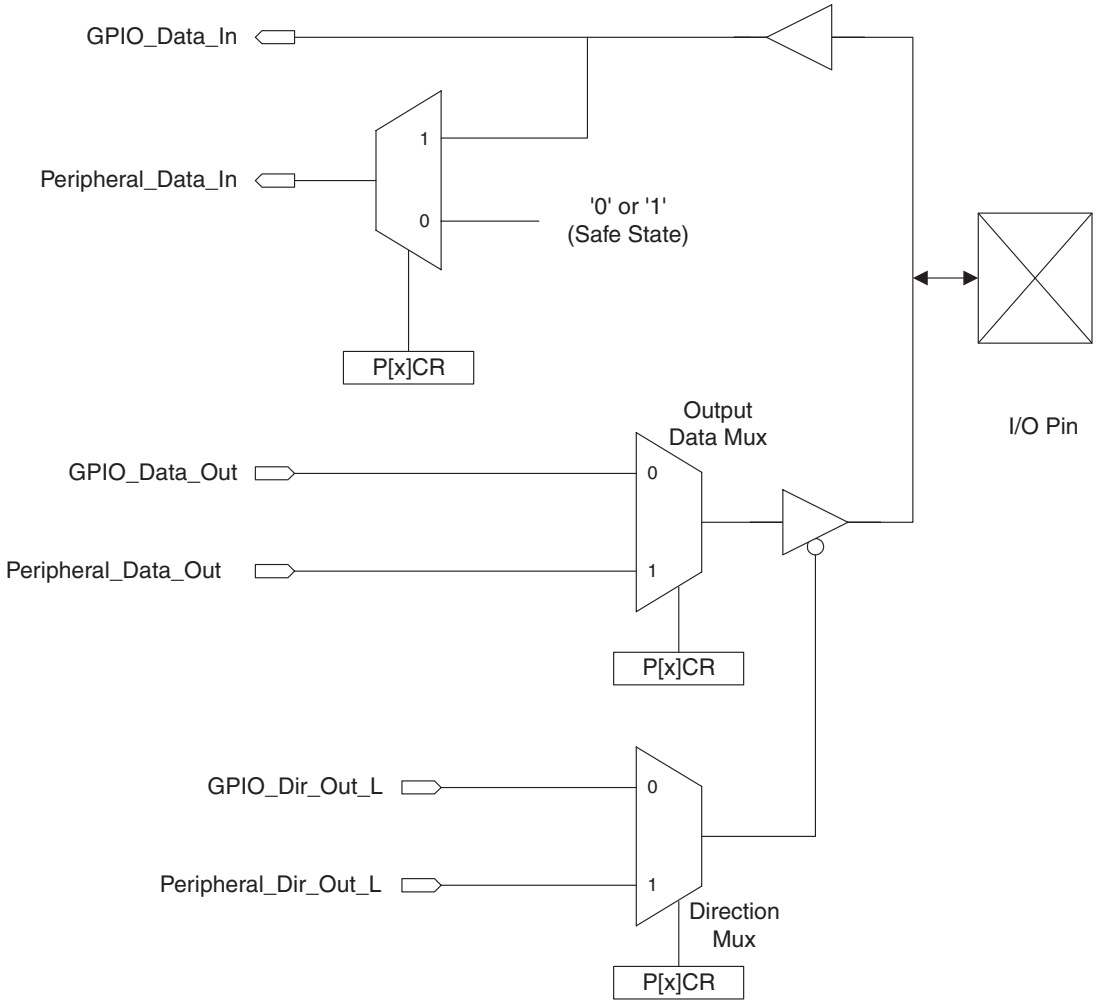
Table 6-5. Port E Pin Function Definitions

VS2000 Pin	Function		
	GPIO Bit	**Peripheral Function**	**Peripheral Direction**
PE0/TXD0	Port E [0]	UART 0 transmit data output	O
PE1/TXD1	Port E [1]	UART 1 transmit data output	O
PE2/RXD0	Port E [2]	UART 0 receive data input	I
PE3/RXD1	Port E [3]	UART 1 receive data input	I
PE4/RTS0	Port E [4]	UART 0 request to send output (active low)	O
PE5/RTS1	Port E [5]	UART 1 request to send output (active low)	O
PE6/CTS0	Port E [6]	UART 0 clear to send input (active low)	I
PE7/CTS1	Port E [7]	UART 1 clear to send input (active low)	I

6.1 I/O Pin Multiplexing

Each VS2000 I/O pin shared between GPIO and peripheral I/O functions is multiplexed using the circuit shown below. In the case where the peripheral function is an input, then the output data multiplexer is replace by a single connection to the GPIO output signal. In the case where the GPIO function is selected, then the peripheral's default input state is set by a predetermined logic level, leaving the peripheral input in a default "safe" state.

Figure 6-1. I/O Pin Multiplexers



The peripheral direction signal (Peripheral_Dir_Out_L) state is set by a predetermined logic level, representing the appropriate direction of the connected peripheral. For example, in the case of Timer/Counter A input 0 (TCAI0), the peripheral direction signal is fixed to logic "1" representing an input.

6.2 I/O Pin Control Registers

6.2.1 Port A Control Register

Table 6-6. Port A Control Register (PACR)

Address	Bits[31:0]	Register Name
$78	–	PACR
Read/Write	R/W	
Reset	$00000000	

6.2.1.1 Port A Configuration Bits

These are used on a bitwise basis to select either GPIO or peripheral connections to the VS2000 configurable I/O pins.

0 = GPIO port A data

1 = Peripheral connection (applies to bits 31 through 10 only)

Note: PACR bits correspond to GPIO port pins in a bitwise fashion, for example, PACR[31] corresponds to GPIO Port A [31].

6.2.2 Port B Control Register

Table 6-7. Port B Control Register (PBCR)

Address	Bits[15:0]	Register Name
$79	–	PBCR
Read/Write	R/W	
Reset	$0000	

6.2.2.1 Port B Configuration Bits

These are used on a bitwise basis to select either GPIO or peripheral connections to the VS2000 configurable I/O pins.

0 = GPIO port B data

1 = Peripheral connection

Note: PBCR bits correspond to GPIO port pins in a bitwise fashion, for example, PBCR[15] corresponds to GPIO Port B [15].

6.2.3 Port C Control Register

Table 6-8. Port C Control Register (PCCR)

Address	Bits[5:0]	Register Name
$7A	–	PCCR
Read/Write	R/W	
Reset	$00	

6.2.3.1 Port C Configuration Bits

These are used on a bitwise basis to select either GPIO or peripheral connections to the VS2000 configurable I/O pins.

0 = GPIO Port C data

1 = Peripheral connection

Note: PCCR bits correspond to GPIO port pins in a bitwise fashion, for example, PCCR[5] corresponds to GPIO Port C [5].

6.2.4 Port D Control Register

Table 6-9. Port D Control Register (PDCR)

Address	Bits[5:0]	Register Name
$7B	–	PDCR
Read/Write	R/W	
Reset	$00	

6.2.4.1 Port D Configuration Bits

These are used on a bitwise basis to select either GPIO or peripheral connections to the VS2000 configurable I/O pins.

0 = GPIO port D data

1 = Peripheral connection

Note: PDCR bits correspond to GPIO port pins in a bitwise fashion, for example, PDCR[5] corresponds to GPIO Port D [5].

6.2.5 Port E Control Register

Table 6-10. Port E Control Register (PECR)

Address	Bits[7:0]	Register Name
$7C	–	PECR
Read/Write	R/W	
Reset	$00	

6.2.5.1 Port E Configuration Bits

These are used on a bitwise basis to select either GPIO or peripheral connections to the VS2000 configurable I/O pins.

0 = GPIO port E data

1 = Peripheral connection

Note: PECR bits correspond to GPIO port pins in a bitwise fashion, for example, PECR[7] corresponds to GPIO Port E [7].

6.2.6 Port F Control Register

Table 6-11. Port F Control Register (PFCR)

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register Name
$7D	–	–	–	–	–	–	–	DBEN	PFCR
Read/Write	R	R	R	R	R	R	R	R	
Reset	–	–	–	–	–	–	–	(1)	

The PFCR is a read-only register that is used to monitor the state of the Port F GPIO pins. GPIO on Port F may only be used when VS2000 is configured to operate in 8-bit mode otherwise the I/O pins are used for data bus bits (D[31:8]).

Note 1. The reset state of the pin is determined by the VS2000 operating mode selected. This bit is read as a "1" when 8-bit mode is selected. When 32-bit mode is selected, this bit is read as a "0".

0 = Data bus connected (32-bit mode selected)

1 = GPIO port F connected

Chapter 7. Timer System

The timer system provides the VS2000 device with a total of two timer/counter blocks (TimerA and TimerB), each containing three identical 32-bit timer/counter channels. Each channel can be independently programmed to perform a wide range of I/O functions, including external event counting, input period and pulse width measurement, pulse generation output, and pulse width modulation (PWM) output generation.

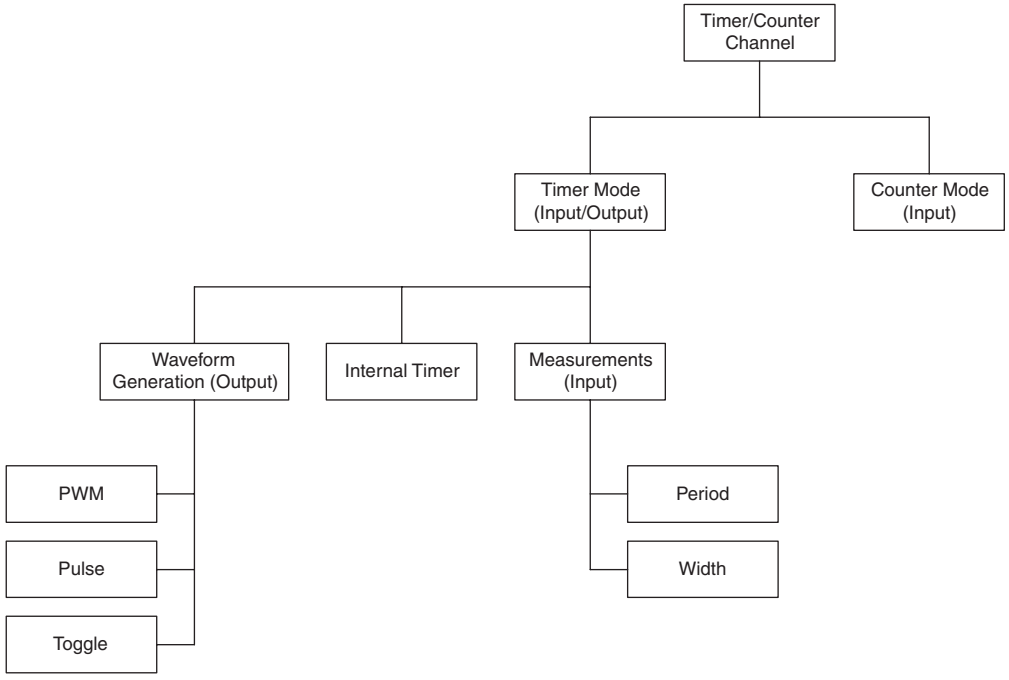
Each timer/counter channel is clocked using the system clock frequency, where one timer/counter bit period is equal to one system clock period.

Each timer/counter channel has one input and one output, provided for user configurable I/O functions. TimerA I/O pins are located in PC[5:0] and TimerB is located in PD[5:0]. If timer I/O functions are required in either of these timer/counter blocks, then the associated port configuration register will need to be configured appropriately. In the case of TimerA, the PCCR I/O configuration register would need to be configured and in the case of TimerB, the PDCR I/O configuration register would need to be configured.

Each timer/counter channel can be used to generate an internal interrupt signal that can be programmed to generate processor interrupts via the interrupt controller block. Interrupts can be generated as a result of internal timer system events or external input events.

The timer/counter channels can be configured in either a timer or counter mode of operation, each of these modes has its own set of associated operating functions. The available timer/counter functions are shown in Figure 7-1.

Figure 7-1. Timer/Counter Function Summary



7.1 Counter Mode

Counter mode allows external rising or falling edge events on the timer/counter's input pin to be counted. All counter events are capable of generating system interrupts to indicate that events have occurred, for example, when a fixed number of external edge events have been counted.

In counter mode, either rising or falling external edge events may be counted. The edge-detection circuit for each timer/counter channel requires that the input waveform is stable in either a high or low state for a minimum of three system clock periods. This results in a maximum timer/counter input frequency of f_{SYS}/6, where f_{SYS} is the system clock frequency.

When counter mode is enabled, the Timer Count Value Register (TCVR) starts counting the selected edge events from the value stored in the Timer Pre-Load Register (TPLR). When the number of counted events reaches the value stored in the Timer End Count Value Register (TECVR), the timer status register is updated and an interrupt is generated, if enabled, to reflect that the maximum count has been reached.

7.2 Timer Mode

Timer mode makes use of the Timer/Counter Count Value Register (TCVR) to provide various user configurable timer functions. These functions include output waveform generation, internal timing and input waveform measurement functions.

7.2.1 Output Waveform Generation

Each timer/counter channel may be configured to generate one of three output waveform formats, these include Pulse Width Modulation (PWM), pulse, and toggle waveforms. All output waveform generation functions require that the timer/counter channel is placed in a continuous mode of operation using the Timer Mode Register (TMR). All output waveform generation functions are capable of generating system interrupts and use the TCVR.

PWM output waveforms use the Timer Pre-Load Register (TPLR), Timer Middle Count Value Register (TMCVR) and the Timer End Count Value Register (TECVR). When running, the output is initially set to the level specified in the Timer I/O Register (TIOR), the Timer/Counter Count Value Register (TCVR) counts upwards from the start value specified in the Timer Pre-Load Register (TPLR) until it reaches the value set in the Timer Middle Count Value Register (TMCVR). At this point the timer/counter's output toggles its state and the Timer Count Value Register continues counting upwards towards the value set in the Timer End Count Value Register (TECVR). Once the value in the TCVR reaches and equals the TECVR, the output toggles its state once again while an interrupt is generated, if enabled, and the process repeats.

Toggle output waveforms are suitable for generating external clock waveforms of programmable frequency. Toggle outputs use the TPLR and TECVR registers. When running, the output is initially set to the level specified in the TIOR, the TCVR counts upwards from the start value specified in TPLR until it reaches the value set in TECVR. At this point the timer/counter's output toggles its state and the count value register is re-loaded with the TPLR value while an interrupt is generated, if enabled, and the process repeats.

Pulse output waveforms are suitable for generating external synchronisation pulses for triggering external events. A pulse is generated with a duration equal to the system clock cycle period in a configurable polarity. Pulse outputs use the TPLR and TECVR registers and are generated in a similar manner to toggle outputs. When running the output is initially set to the level specified in the TIOR, the TCVR counts upwards from the start value specified in the TPLR until it reaches the value set in the TECVR. At this point, the timer/counter's output toggles its state for one clock cycle only and the count value register is re-loaded with the TPLR value while an interrupt is generated, if enabled, and the process repeats.

7.2.2 Internal Timer

Each timer/counter channel may be simply used as a system timer. When configured to perform this function, the timer can be used to generate system interrupts to notify the system that a particular time event has occurred. These can be either run continuously or in a one-shot mode of operation depending upon system requirements. If interrupts are not desirable, the timer status can be polled to see if a timeout condition has occurred.

Timer operation makes use of the TPLR and TECVR registers. When timer mode is started, the TCVR begins counting from the value stored in the TPLR until it reaches the value stored in the TECVR. At this point either a system interrupt is generated (if enabled) and the timer stops (if in one-shot mode) or it continues by automatically re-loading itself with the value stored in the TPLR and the process repeats.

7.2.3 Input Waveform Measurements

Each timer/counter can be configured to perform either period or pulse measurements of the waveform on its input pin. An edge-detection circuit is used to detect the appropriate starting edge and continuously count from the value stored in the TPLR until the selected edge is detected. At this point, a system interrupt is generated (if enabled) and the timer stops (if in one-shot mode) or it continues in its programmed measurement mode waiting for the next event.

In the case of continuous period measurements, the end of a measurement is the start of the next measurement assuming that the detection conditions have not been modified.

7.3 Timer System Registers

Both of the VS2000 timer system blocks (TimerA and TimerB) have identical functionality and register sets. Because of this, only one description for each register has been documented, that said, all registers are shown and corresponding bits are shown for clarification.

7.3.1 Control/Status Registers

Table 7-1. Timer A Control/Status Register 0 (TACSR0) - Write

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register Name
$20	–	–	EN	IE	–	–	MINTCLR	INTCLR	TACSR0
Read/Write	W	W	W	W	W	W	W	W	
Reset	–	–	–	–	–	–	–	–	

Table 7-2. Timer A Control/Status Register 0 (TACSR0) - Read

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register Name
$20	–	EXP	EN	IE	MINTOVR	INTOVR	MINT	INT	TACSR0
Read/Write	R	R	R	R	R	R	R	R	
Reset	0	0	0	0	0	0	0	0	

Table 7-3. Timer A Control/Status Register 1 (TACSR1) – Write

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register Name
$28	–	–	EN	IE	–	–	MINTCLR	INTCLR	TACSR1
Read/Write	W	W	W	W	W	W	W	W	
Reset	–	–	–	–	–	–	–	–	

Table 7-4. Timer A Control/Status Register 1 (TACSR1) – Read

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register Name
$28	–	EXP	EN	IE	MINTOVR	INTOVR	MINT	INT	TACSR1
Read/Write	R	R	R	R	R	R	R	R	
Reset	0	0	0	0	0	0	0	0	

Table 7-5. Timer A Control/Status Register 2 (TACSR2) – Write

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register Name
$30	–	–	EN	IE	–	–	MINTCLR	INTCLR	TACSR2
Read/Write	W	W	W	W	W	W	W	W	
Reset	–	–	–	–	–	–	–	–	

Table 7-6. Timer A Control/Status Register 2 (TACSR2) – Read

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register Name
$30	–	EXP	EN	IE	MINTOVR	INTOVR	MINT	INT	TACSR2
Read/Write	R	R	R	R	R	R	R	R	
Reset	0	0	0	0	0	0	0	0	

Table 7-7. Timer B Control/Status Register 0 (TBCSR0) – Write

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register Name
$40	–	–	EN	IE	–	–	MINTCLR	INTCLR	TBCSR0
Read/Write	W	W	W	W	W	W	W	W	
Reset	–	–	–	–	–	–	–	–	

Table 7-8. Timer B Control/Status Register 0 (TBCSR0) - Read

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register Name
$40	–	EXP	EN	IE	MINTOVR	INTOVR	MINT	INT	TBCSR0
Read/Write	R	R	R	R	R	R	R	R	
Reset	0	0	0	0	0	0	0	0	

Table 7-9. Timer B Control/Status Register 1 (TBCSR1) - Write

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register Name
$48	–	–	EN	IE	–	–	MINTCLR	INTCLR	TBCSR1
Read/Write	W	W	W	W	W	W	W	W	
Reset	–	–	–	–	–	–	–	–	

Table 7-10. Timer B Control/Status Register 1 (TBCSR1) - Read

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register Name
$48	–	EXP	EN	IE	MINTOVR	INTOVR	MINT	INT	TBCSR1
Read/Write	R	R	R	R	R	R	R	R	
Reset	0	0	0	0	0	0	0	0	

Table 7-11. Timer B Control/Status Register 2 (TBCSR2) - Write

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register Name
$50	–	–	EN	IE	–	–	MINTCLR	INTCLR	TBCSR2
Read/Write	W	W	W	W	W	W	W	W	
Reset	–	–	–	–	–	–	–	–	

Table 7-12. Timer B Control/Status Register 2 (TBCSR2) - Read

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register Name
$50	–	EXP	EN	IE	MINTOVR	INTOVR	MINT	INT	TBCSR2
Read/Write	R	R	R	R	R	R	R	R	
Reset	0	0	0	0	0	0	0	0	

Each timer/counter channel's control and status register (TCSR) is an 8-bit wide register used to implement control and monitor status of the timer/counter. Reads correspond to reading the status register and writes are to the control register.

7.3.1.1 Control Register

Bits[7:6] - Reserved

These bits have no function and writes to them are ignored.

EN - Enable

The Enable bit is used to enable or disable the timer/counter channel. When enabled, timer/counter will operate as per its configuration.

0 = Disable the timer/counter channel

1 = Enable the timer/counter channel

IE - Interrupt Enable

The Interrupt Enable bit is used to control interrupt generation following completion of a timer or counter operation.

0 = Disable interrupt generation

1 = Enable interrupt generation

Bits[3:2] - Reserved

These bits have no function and writes to them are ignored.

MINTCLR - Measurement Interrupt Clear

The Measurement Interrupt Clear bit is used to clear measurement interrupts. Measurement interrupts are generated when the timer/counter channel is in timer mode and is configured to measure input periods or widths and a measurement completes.

0 = Leave measurement interrupt status unmodified

1 = Clear an existing measurement interrupt

INTCLR - Interrupt Clear

The Interrupt Clear bit is used to clear all interrupts other than measurement interrupts. INTCLR clears interrupts that are generated when the timer/counter channel reaches the value stored in the TECVR when operating in timer mode, for internal timer and waveform generation, and counter mode.

0 = Leave interrupt status unmodified

1 = Clear an existing interrupt

7.3.1.2 Status Register

Bits[7:6] - Reserved

These bits have no function and return zeros when read.

EXP - Expired

The Expired bit is used to indicate that a one-shot timer/counter operation has completed or the timer is idle.

0 = Timer/counter operation in one-shot mode not completed (if enabled)

1 = Timer/counter one-shot operation completed

EN - Enable

The Enable bit is used to indicate the status of the EN bit in the control register.

0 = Timer/counter channel disabled

1 = Timer/counter channel enabled

IE - Interrupt Enable

The Interrupt Enable bit is used to indicate the status of the IE bit in the control register.

0 = Timer/counter interrupts disabled

1 = Timer/counter interrupts enabled

MINTOVR - Measurement Interrupt Overrun

The Measurement Interrupt Overrun bit is used to indicate that one or more measurement interrupt events have occurred before a previous measurement interrupt was cleared. This indicates that data may have been lost as a result of an overrun measurement interrupt. The MINTOVR bit is cleared when the MINTCLR bit is set in the control register.

0 = No measurement interrupt overrun condition detected

1 = Measurement interrupt overrun condition detected

INTOVR - Interrupt Overrun

The Interrupt Overrun bit is used to indicate that one or more interrupt events, not including measurement interrupts, have occurred before a previous interrupt was cleared. This indicates that data may have been lost as a result of an overrun interrupt. The INTOVR bit is cleared when the INTCLR bit is set in the control register.

0 = No interrupt overrun condition detected

1 = Interrupt overrun condition detected

MINT - Measurement Interrupt

The measurement interrupt bit indicates the status of the timer/count measurement interrupt.

0 = No measurement interrupt present

1 = A measurement interrupt has occurred

INT - Interrupt

The interrupt bit indicates the status of the timer/counter channel interrupt.

0 = No interrupt present

1 = An interrupt has occurred

7.3.2 Mode Registers

Table 7-13. Timer A Mode Register 0 (TAMR0)

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register Name
$21	–	–	–	–	TM[2]	TM[1]	TM[0]	CONT	TAMR0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	0	0	0	

Table 7-14. Timer A Mode Register 1 (TAMR1)

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register Name
$29	–	–	–	–	TM[2]	TM[1]	TM[0]	CONT	TAMR1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	0	0	0	

Table 7-15. Timer A Mode Register 2 (TAMR2)

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register Name
$31	–	–	–	–	TM[2]	TM[1]	TM[0]	CONT	TAMR2
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	0	0	0	

Table 7-16. Timer B Mode Register 0 (TBMR0)

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register Name
$41	–	–	–	–	TM[2]	TM[1]	TM[0]	CONT	TBMR0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	0	0	0	

Table 7-17. Timer B Mode Register 1 (TBMR1)

Addr (hex)	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register Name
49	-	-	-	-	TM[2]	TM[1]	TM[0]	CONT	TBMR1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	0	0	0	

Table 7-18. Timer B Mode Register 2 (TBMR2)

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register Name
$51	–	–	–	–	TM[2]	TM[1]	TM[0]	CONT	TBMR2
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	0	0	0	

Each mode register is an 8-bit wide read/write register used to configure the operating mode of the timer/counter channel.

Bits[7:4] - Reserved

These bits have no function. Writes to them are ignored and reads return zeros.

TM[2:0] - Timer/Counter Mode

The Timer/counter Mode bits are used to configure the operating mode of the timer/counter as per the description in the Table 7-19.

Table 7-19. Timer/Counter Mode Configuration Bits

TM[2]	TM[1]	TM[0]	Mode	Description
0	0	0	Timer	Internal timer
0	0	1	Timer	Pulse output waveform generation
0	1	0	Timer	Toggle output waveform generation
0	1	1	Timer	PWM output waveform generation
1	0	0	Timer	Input width measurement
1	0	1	Timer	Input period measurement
1	1	0	Counter	External event counter
1	1	1	Reserved	Reserved

CONT - Continuous

The Continuous bit is used to place the timer/counter channel in a continuous mode of operation. In continuous mode, the timer/counter will automatically restart a subsequent operation immediately following completion of a previous one. The timer/counter is required to be placed in continuous mode when operating in the output waveform timer mode of operation. A timer/counter is required to be in a one-shot mode of operation when used in timer input measurement mode.

0 = One-shot mode selected

1 = Continuous mode selected

7.3.3 I/O Registers

Table 7-20. Timer A I/O Register 0 (TAIOR0)

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register Name
$22	–	–	–	–	–	TPOL	TOL	TOE	TAIOR0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	1	0	0	

Table 7-21. Timer A I/O Register 1 (TAIOR1)

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register Name
$2A	–	–	–	–	–	TPOL	TOL	TOE	TAIOR1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	1	0	0	

Table 7-22. Timer A I/O Register 2 (TAIOR2)

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register Name
$22	–	–	–	–	–	TPOL	TOL	TOE	TAIOR2
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	1	0	0	

Table 7-23. Timer B I/O Register (TBIOR0)

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register Name
$42	–	–	–	–	–	TPOL	TOL	TOE	TBIOR0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	1	0	0	

Table 7-24. Timer B I/O Register 1 (TBIOR1)

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register Name
$4A	–	–	–	–	–	TPOL	TOL	TOE	TBIOR1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	1	0	0	

Table 7-25. Timer B I/O Register 2 (TBIOR2)

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register Name
$52	–	–	–	–	–	TPOL	TOL	TOE	TBIOR2
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	1	0	0	

Each I/O register is an 8-bit wide read/write register used to control the timer/counter's external I/O interface.

Bits[7:3] - Reserved

These bits have no function. Writes to them are ignored and reads return zeros.

TPOL - Input Polarity

The Timer/counter Input Polarity bit is used during timer measurement and counter modes of operation to define the type of edge the timer/counter channel is to detect to commence measuring or counting depending on the mode selected. Table 7-26 defines which edges are used in either timer measurement mode or counter mode based on the TPOL value set.

Table 7-26. Timer/Counter Polarity Bit

	Timer Measurement Mode				Counter Mode
	Period		Width		
TPOL	Start Edge	End Edge	Start Edge	End Edge	Edge Events Counted
0	Falling	Falling	Falling	Rising	Falling
1	Rising	Rising	Rising	Falling	Rising

For clarification, the effect of TPOL on measurements and input edge counting is shown in the diagram below. The shaded areas highlight the measurement durations for both period and width measurements.

TOL - Output Level

The Output Level bit is used to set the initial state of the timer/counter channel's output. This is used when the timer/counter is in timer output waveform generation mode. The value is automatically updated to reflect the value on the output pin.

TOE - Output Enable

The Output Enable bit is used to enable and disable the timer/counter channel's output. This provides a means of disabling the output whilst allowing the output generation function to continue.

0 = Output disabled (output forced to high level)

1 = Output enabled

7.3.4 Count Value Registers

Table 7-27. Timer A Count Value Register 0 (TACVR0)

Address	Bits[31:0]	Register Name
$23		TACVR0
Read/Write	R	
Reset	$00000000	

Table 7-28. Timer A Count Value Register 1 (TACVR1)

Address	Bits[31:0]	Register Name
$2B		TACVR1
Read/Write	R	
Reset	$00000000	

Table 7-29. Timer A Count Value Register 2 (TACVR2)

Address	Bits[31:0]	Register Name
$33		TACVR2
Read/Write	R	
Reset	$00000000	

Table 7-30. Timer B Count Value Register 0 (TBCVR0)

Address	Bits[31:0]	Register Name
$43		TBCVR0
Read/Write	R	
Reset	$00000000	

Table 7-31. Timer B Count Value Register 1 (TBCVR1)

Address	Bits[31:0]	Register Name
$4B		TBCVR1
Read/Write	R	
Reset	$00000000	

Table 7-32. Timer B Count Value Register 2 (TBCVR2)

Address	Bits[31:0]	Register Name
$53		TBCVR2
Read/Write	R	
Reset	$00000000	

Each count value register (TCVR) is a 32-bit wide read-only register used to read the current count value of the timer. When timer/counter operations commence, the TCVR is automatically loaded with the pre-load register value (TPLR) and starts timing by incrementing (by one) every system clock cycle.

7.3.5 Pre-Load Value Registers

Table 7-33. Timer A Pre-Load Value Register 0 (TAPLR0)

Address	Bits[31:0]	Register Name
$24		TAPLR0
Read/Write	R/W	
Reset	$00000000	

Table 7-34. Timer A Pre-Load Value Register 1 (TAPLR1)

Address	Bits[31:0]	Register Name
$2C		TAPLR1
Read/Write	R/W	
Reset	$00000000	

Table 7-35. Timer A Pre-Load Value Register 2 (TAPLR2)

Address	Bits[31:0]	Register Name
$34		TAPLR2
Read/Write	R/W	
Reset	$00000000	

Table 7-36. Timer B Pre-Load Value Register 0 (TBPLR0)

Address	Bits[31:0]	Register Name
$44		TBPLR0
Read/Write	R/W	
Reset	$00000000	

Table 7-37. Timer B Pre-Load Value Register 1 (TBPLR1)

Address	Bits[31:0]	Register Name
$4C		TBPLR1
Read/Write	R/W	
Reset	$00000000	

Table 7-38. Timer B Pre-Load Value Register 2 (TBPLR2)

Address	Bits[31:0]	Register Name
$54		TBPLR2
Read/Write	R/W	
Reset	$00000000	

Each pre-load register (TPLR) is a 32-bit wide read/write register used to set the timer count start value. When timer/counter operations commence, the value stored in the TPLR is loaded into the TCVR and the timer starts timing by counting from the TPLR value.

It should be noted that the value stored in the TPLR assumes one cycle for the initial value. For example, if the TPLR has a value of $00000000, and the TECVR has a value of $00000004, then a total of five cycles will be required to increment the TCVR to the TECVR.

7.3.6 Middle Count Value Registers

Table 7-39. Timer A Middle Count Value Register 0 (TAMCVR0)

Address	Bits[31:0]	Register Name
$25		TAMCVR0
Read/Write	R/W	
Reset	$FFFFFF00	

Table 7-40. Timer A Middle Count Value Register 1 (TAMCVR1)

Address	Bits[31:0]	Register Name
$2D		TAMCVR1
Read/Write	R/W	
Reset	$FFFFFF00	

Table 7-41. Timer A Middle Count Value Register 2 (TAMCVR2)

Address	Bits[31:0]	Register Name
$35		TAMCVR2
Read/Write	R/W	
Reset	$FFFFFF00	

Table 7-42. Timer B Middle Count Value Register 0 (TBMCVR0)

Address	Bits[31:0]	Register Name
$45		TBMCVR0
Read/Write	R/W	
Reset	$FFFFFF00	

Table 7-43. Timer B Middle Count Value Register 1 (TBMCVR1)

Address	Bits[31:0]	Register Name
$4D		TBMCVR1
Read/Write	R/W	
Reset	$FFFFFF00	

Table 7-44. Timer B Middle Count Value Register 2 (TBMCVR2)

Address	Bits[31:0]	Register Name
$55		TBMCVR2
Read/Write	R/W	
Reset	$FFFFFF00	

Each middle count value register (TMCVR) is a 32-bit wide read/write register used to set the middle count value used during timer mode's PWM output waveform generation. When PWM output waveform generation is selected and the TCVR reaches the value stored in the TMCVR, the output toggles its state.

7.3.7 End Count Value Registers

Table 7-45. Timer A End Count Value Register 0 (TAECVR0)

Address	Bits[31:0]	Register Name
$26		TAECVR0
Read/Write	R/W	
Reset	$FFFFFFFF	

Table 7-46. Timer A End Count Value Register 1 (TAECVR1)

Address	Bits[31:0]	Register Name
$2E		TAECVR1
Read/Write	R/W	
Reset	$FFFFFFFF	

Table 7-47. Timer A End Count Value Register 2 (TAECVR2)

Address	Bits[31:0]	Register Name
$36		TAECVR2
Read/Write	R/W	
Reset	$FFFFFFFF	

Table 7-48. Timer B End Count Value Register 0 (TBECVR0)

Address	Bits[31:0]	Register Name
$46		TBECVR0
Read/Write	R/W	
Reset	$FFFFFFFF	

Table 7-49. Timer B End Count Value Register 1 (TBECVR1)

Address	Bits[31:0]	Register Name
$4E		TBECVR1
Read/Write	R/W	
Reset	$FFFFFFFF	

Table 7-50. Timer B End Count Value Register 2 (TBECVR2)

Address	Bits[31:0]	Register Name
$56		TBECVR2
Read/Write	R/W	
Reset	$FFFFFFFF	

Each end count value register (TECVR) is a 32-bit wide read/write register used to set the end count value used during different timer/counter modes of operation. The timer/counter modes that use the TECVR are timer output waveform generation, internal timer, and counter modes. Note that timer input measurement mode does not make use of this register.

When the timer/counter channel is in a mode where the TECVR is used and the TCVR equals the value in the TECVR, the TCVR is loaded with the value in the TPLR and the TCVR begins incrementing from the TPLR value. If continuous mode is selected, CONT is set in TMR. Otherwise the EXP (expired) bit in the status register is set and the TCVR halts.

7.3.8 Measurement Value Registers

Table 7-51. Timer A Measurement Value Register 0 (TAMVR0)

Address	Bits[31:0]	Register Name
$27		TAMVR0
Read/Write	R	
Reset	$00000000	

Table 7-52. Timer A Measurement Value Register 1 (TAMVR1)

Address	Bits[31:0]	Register Name
$2F		TAMVR1
Read/Write	R	
Reset	$00000000	

Table 7-53. Timer A Measurement Value Register 2 (TAMVR2)

Address	Bits[31:0]	Register Name
$37		TAMVR2
Read/Write	R	
Reset	$00000000	

Table 7-54. Timer B Measurement Value Register 0 (TBMVR0)

Address	Bits[31:0]	Register Name
$47		TBMVR0
Read/Write	R/W	
Reset	$00000000	

Table 7-55. Timer B Measurement Value Register 1 (TBMVR1)

Address	Bits[31:0]	Register Name
$4F		TBMVR1
Read/Write	R/W	
Reset	$00000000	

Table 7-56. Timer B Measurement Value Register 2 (TBMVR2)

Address	Bits[31:0]	Register Name
$57		TBMVR2
Read/Write	R/W	
Reset	$00000000	

Each measurement value register (TMVR) is a 32-bit wide read-only register used to hold the result of the previous (completed) measurement. The value stored in the register represents the number of system clock cycles that elapsed during the selected input measurement. The user must note that enough time should be allowed for the duration of measurement. It is therefore recommended that measurements are performed with the TPLR set to $00000000.

To ensure that the correct result is returned for a measurement, it is recommended that input measurements are performed in a one-shot mode of operation, where the CONT bit in the timer/counter channel's mode register is cleared. This ensures that a measurement value is not overwritten if external input events cause the TMVR to be updated before the register value is read.

7.3.9 Interrupt Status Registers

Table 7-57. Timer A Interrupt Status Register (TAISR)

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register Name
$3F	-	-	-	-	-	INTT2	INTT1	INTT0	TAISR
Read/Write	R	R	R	R	R	R	R	R	
Reset	0	0	0	0	0	0	0	0	

Table 7-58. Timer B Interrupt Status Register (TBISR)

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register Name
$5F	-	-	-	-	-	INTT2	INTT1	INTT0	TBISR
Read/Write	R	R	R	R	R	R	R	R	
Reset	0	0	0	0	0	0	0	0	

Each timer interrupt status register is an 8-bit read-only register used to monitor the status of each of the three timers within each of the timer/counter blocks.

INTT2 - Timer/Counter Channel 2 Interrupt Status

The Timer/counter Channel 2 Interrupt Status is set whenever an interrupt is present in timer/counter channel 2. The interrupt may either be an interrupt or measurement interrupt. When cleared, timer/counter channel 2 has no interrupts pending.

INTT1 - Timer/Counter Channel 1 Interrupt Status

The Timer/counter Channel 1 Interrupt Status is set whenever an interrupt is present in timer/counter channel 1. The interrupt may either be an interrupt or measurement interrupt. When cleared, timer/counter channel 1 has no interrupts pending.

INTT0 - Timer/Counter Channel 0 Interrupt Status

The Timer/counter Channel 0 Interrupt Status is set whenever an interrupt is present in timer/counter channel 0. The interrupt may either be an interrupt or measurement interrupt. When cleared, timer/counter channel 0 has no interrupts pending.

Chapter 8. Watchdog

The watchdog module provides a system watchdog timer function for the VS2000 device. When enabled, the system watchdog timer can be used to reset the VS2000 system, preventing system lock-up conditions if software becomes trapped in a deadlock.

In normal operation, the user reloads the watchdog at regular intervals before the timer timeout condition occurs, effectively restarting the timer which automatically begins counting upwards towards the user programmable time limit value. A watchdog timeout condition occurs when the watchdog control register has not been written to within the user programmable timeout period. When the watchdog function is enabled and timeout occurs, the watchdog can be configured to generate either an interrupt or full system reset.

Full system reset results in the VS2000 device being held in a reset state for 16 system clock cycles as a result of the watchdog timeout condition. During this time, the $\overline{\text{WDTO}}$ pin is driven low for the duration watchdog generate reset (16 system clock cycles). Normally, $\overline{\text{WDTO}}$ is driven high. The $\overline{\text{WDTO}}$ pin may be used by other devices in a system as a signal to reset.

For system software development purposes, the watchdog can be configured to generate a system reset in the event of a watchdog timeout condition.

8.1 Watchdog Registers

The watchdog registers are located in the VS2000 register memory space. The registers are described in the following sections.

8.1.1 Watchdog Control and Status Register

Table 8-1. Watchdog Control and Status Register (WDTCSR) – Write

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register Name
$72	TPS[3]	TPS[2]	TPS[1]	TPS[0]	STB	MODE	EN	INTCLR	WDTCSR
Read/Write	W	W	W	W	W	W	W	W	
Reset	-	-	-	-	-	-	-	-	

Table 8-2. Watchdog Control and Status Register (WDTCSR) – Read

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register Name
$72	TPS[3]	TPS[2]	TPS[1]	TPS[0]	-	MODE	EN	INT	WDTCSR
Read/Write	R	R	R	R	R	R	R	R	
Reset	1	0	0	0	0	0	0	0	

The Watchdog Control and Status Register (WDTCSR) is an 8-bit register used to implement control and monitor status of the real-time clock counter system. Reads correspond to reading the status register and writes are to the control register.

8.1.1.1 Control Register

TPS[3:0] - Timeout Period Select

The Timeout Period Select (TPS) bits are used to control and select the watchdog timeout period. The timeout period is defined as the number of system clock cycles that are counted from zero to the selected timeout count limit. The TPS bits may only be updated when the watchdog timer is disabled. Attempts to update these bits while the watchdog is enabled will be ignored.

The system clock count values for each TPS combination are defined in Table 8-3.

Table 8-3. Timeout Period Select

TPS[3]	TPS[2]	$TPS[1]	TPS[0]	System Clock Cycles Counted
0	0	0	0	$00010000
0	0	0	1	$00020000
0	0	1	0	$00040000
0	0	1	1	$00080000
0	1	0	0	$00100000
0	1	0	1	$00200000
0	1	1	0	$00400000
0	1	1	1	$00800000
1	0	0	0	$01000000
1	0	0	1	$02000000
1	0	1	0	$04000000
1	0	1	1	$08000000
1	1	0	0	$10000000
1	1	0	1	$20000000
1	1	1	0	$40000000
1	1	1	1	$80000000

STB - Strobe

The Strobe bit is used to reset the watchdog timeout counter to zero (only when the watchdog is enabled).

0 = Leave watchdog counter unmodified

1 = Reset watchdog counter to zero

MODE - Watchdog Mode

The Mode bit is used to control the watchdog's operating mode when the watchdog is enabled. The two possible modes are Reset and Interrupt. When in Reset mode, the VS2000 is fully reset when a watchdog timeout condition occurs. When in Interrupt mode, an interrupt is generated as a result of a watchdog timeout condition.

0 = Reset mode

1 = Interrupt mode

EN - Enable

The Enable bit is used to enable or disable the watchdog function. When enabled, the watchdog timer will continuously count resulting in generation of either a reset or interrupt when a watchdog timeout condition occurs. When disabled, the watchdog timer is held in a reset state and an existing interrupt, if present, is cleared.

0 = Disable the watchdog

1 = Enable the watchdog

INTCLR - Interrupt Clear

The Interrupt Clear bit is used to clear an interrupt condition which has been generated as a result of a watchdog timeout condition when the watchdog is enabled. If an interrupt is cleared at the same time that a new interrupt is generated, the new interrupt is not lost.

0 = Leave interrupt status unmodified

1 = Clear current interrupt

8.1.1.2 Status Register

TPS[3:0] - Timeout Period Select

The Timeout Period Select bits indicate the status of the TPS[3:0] bits in the control register. The control register description includes a description of the timer period values selected by the TPS bits.

Bit[3] - Reserved

This bit has no function and returns a zero when read.

MODE - Watchdog Mode

The Mode bit is used to indicate the status of the MODE bit in the control register.

0 = Reset mode

1 = Interrupt mode

EN - Enable

The Enable bit is used to indicate the status of the EN bit in the control register.

0 = Watchdog disabled

1 = Watchdog enabled

INT - Interrupt

The Interrupt bit indicates the status of the watchdog interrupt.

0 = No interrupt present

1 = Interrupt present

8.1.2 Watchdog Timer Count Value Register

Table 8-4. Watchdog Timer Count Value Register (WDTCR)

Address	Bits[31:0]	Register Name
$73		WDTCR
Read/Write	R	
Reset	$00000000	

The Watchdog Timer Count Value Register (WDTCR) is a 32-bit read-only register that holds the current value of the watchdog timer count value. The WDTCR is provided to allow software developers to check timeout headroom when developing software that uses the watchdog system.

8.2 Watchdog Timeout Condition

When enabled, a watchdog timeout condition occurs when the WDTCR reaches the count value selected using the TPS[3:0] bits in the control register.

8.3 Watchdog Interrupt

When enabled, the watchdog interrupt is generated by a watchdog timeout condition. An interrupt is cleared by setting the INTCLR bit in the control register. It is also cleared by disabling the watchdog module which is done by clearing the EN bit in the control register.

8.4 Watchdog Reset

When the external reset input is asserted, the $\overline{\text{WDTO}}$ pin is negated (driven to logic "1"). An external device reset circuit design must take this into account.

Chapter 9. Serial Peripheral Interface

The serial peripheral interface system provides the VS2000 device with master-only SPI-compatible serial communication ports. These ports allow the VS2000 to communicate synchronously with peripheral devices such as D/A and A/D converters, UARTs, memories and sensors. Each SPI port directly supports a number of SPI-operational features such as programmable frame transfer length, clock rate, clock polarity, and clock phase.

There are three SPI ports in the VS2000. SPI0 is located in PA[19:10], SPI1 is located in PA[31:21], and SPI2 is located in PB[10:0]. The SPI system is configured as Master-only for all three interfaces with a fully-programmable transfer rate in the range $f_{SYS}/131072$ to $f_{SYS}/4$ bits per second (where f_{SYS} is the system clock frequency). Each SPI Master supports direction connection of up to 8 slave peripheral devices with further device connection using GPIO.

Read and write SPI data is transferred simultaneously for each SPI data transfer.

9.1 Serial Peripheral Interface Registers

Each of the three VS2000 serial peripheral interface blocks (SPI0, SPI1, SPI2) have identical functionality and register sets. Because of this, only one description for each register has been documented. However, all registers and corresponding bits are shown for clarification.

9.1.1 Control/Status Registers

Each Serial Peripheral Interface Control and Status Register (SCSR0, SCSR1, SCSR2) is an 8-bit wide register used to implement control and monitor status of the serial peripheral interface module functions. Reads correspond to reading the status register and writes are to the control register.

Table 9-1. Serial Peripheral Interface Control and Status Register 0 (SCSR0) – Write

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register Name
$B0	ST	IE	CONT	-	-	-	-	INTCLR	SCSR0
Read/Write	W	W	W	W	W	W	W	W	
Reset	-	-	-	-	-	-	-	-	

Table 9-2. Serial Peripheral Interface Control and Status Register 0 (SCSR0) – Read

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register Name
$B0	COMP	IE	CONT	-	-	-	INTOVR	INT	SCSR0
Read/Write	R	R	R	R	R	R	R	R	
Reset	1	0	0	0	0	0	0	0	

Table 9-3. Serial Peripheral Interface Control and Status Register 1 (SCSR1) – Write

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register Name
$B8	ST	IE	CONT	-	-	-	-	INTCLR	SCSR1
Read/Write	W	W	W	W	W	W	W	W	
Reset	-	-	-	-	-	-	-	-	

Table 9-4. Serial Peripheral Interface Control and Status Register 1 (SCSR1) – Read

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register Name
$B8	COMP	IE	CONT	-	-	-	INTOVR	INT	SCSR1
Read/Write	R	R	R	R	R	R	R	R	
Reset	1	0	0	0	0	0	0	0	

Table 9-5. Serial Peripheral Interface Control and Status Register 2 (SCSR2) – Write

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register Name
$C0	ST	IE	CONT	-	-	-	-	INTCLR	SCSR2
Read/Write	W	W	W	W	W	W	W	W	
Reset	-	-	-	-	-	-	-	-	

Table 9-6. Serial Peripheral Interface Control and Status Register 2 (SCSR2) – Read

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register Name
$C0	COMP	IE	CONT	-	-	-	INTOVR	INT	SCSR2
Read/Write	R	R	R	R	R	R	R	R	
Reset	1	0	0	0	0	0	0	0	

9.1.1.1 Control Register

ST - SPI Start Transfer

Setting the Start Transfer bit results in commencing an SPI data transfer.

0 = Do not start SPI transfer

1 = Start SPI transfer

IE - SPI Interrupt Enable

Setting the Interrupt Enable bit enables interrupt generation following completion of the SPI data transfer.

0 = Disable SPI interrupt generation

1 = Enable SPI interrupt generation

CONT - SPI Transfer Frame Continuation Enable

This continuation bit controls the state of the appropriate SPI chip select signal (if used) at the end of an SPI data transfer. Some SPI slave devices require that their chip select is kept asserted between blocks of SPI data transfer to/from the device. This bit allows a user to implement this functionality. Once an SPI data transfer has commenced, the CONT bit may be written and updated up to any time before the completion of a data transfer itself.

0 = Negate SPI chip select output at end of SPI data transfer (force to V_{DD}).

1 = Keep SPI chip select output asserted at the end of SPI data transfer (maintain at V_{SS})

Bits[4:1] - Reserved

These bits have no function and writes to them are ignored.

INTCLR - SPI Interrupt Clear

Setting the Interrupt Clear bit clears an existing interrupt. If a new interrupt occurs at the same time that an interrupt is cleared then the new interrupt is not lost. When INTCLR is set, any interrupt overrun condition is also cleared.

0 = Leave interrupt status unmodified

1 = Clear an existing interrupt

9.1.1.2 Status Register

COMP - SPI Transfer Complete

The Complete bit indicates the current data transfer status of the SPIx module.

0 = SPI data transfer in operation

1 = SPI idle, no data transfer currently occurring

IE - SPI Interrupt Enable Status

The Interrupt Enable Status bit indicates the status of the interrupt enable bit set in the control register.

0 = SPI interrupt generation disabled

1 = SPI interrupt generation enabled

CONT - SPI Transfer Frame Continuation Status

The Continue bit indicates the status of the transfer frame continuation enable bit in the control register.

0 = SPI continuation bit cleared

1 = SPI continuation bit set

Bits[4:2] - Reserved

These bits have no function and return zeros when read.

INTOVR - SPI Interrupt Overrun

The Interrupt Overrun bit is used to indicate that one or more interrupt events have occurred before a previous interrupt was cleared. This indicates that data may have been lost as a result of an overrun interrupt. The INTOVR bit is cleared when the interrupt clear bit is set in the control register.

0 = No interrupt overrun condition detected

1 = Interrupt overrun condition detected

INT - SPI Interrupt Status

The Interrupt Status bit is used to indicate that a data transfer has completed and an interrupt has subsequently been generated. The INT bit is cleared by setting the interrupt clear bit in the control register.

0 = No interrupt present

1 = Interrupt present

9.1.2 Slave Configuration Registers

Table 9-7. Slave Configuration Register 0 (SSLVC0) – Bits 15-8

Address	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Register Name
$B1	-	-	-	-	-	-	SE	SS[2]	SSLVC0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	0	1	0	

Table 9-8. Slave Configuration Register 0 (SSLVC0) – Bits 7-0

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register Name
$B1	SS[1]	SS[0]	CPOL	CPHA	BC[3]	BC[2]	BC[1]	BC[0]	SSLVC0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	1	0	0	0	

Table 9-9. Slave Configuration Register 1 (SSLVC1) – Bits 15-8

Address	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Register Name
$B9	-	-	-	-	-	-	SE	SS[2]	SSLVC1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	0	1	0	

Table 9-10. Slave Configuration Register 1 (SSLVC1) – Bits 7-0

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register Name
$B9	SS[1]	SS[0]	CPOL	CPHA	BC[3]	BC[2]	BC[1]	BC[10]	SSLVC1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	1	0	0	0	

Table 9-11. Slave Configuration Register 2 (SSLVC2) – Bits 15-8

Address	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Register Name
$C1	-	-	-	-	-	-	SE	SS[2]	SSLVC2
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	0	1	0	

Table 9-12. Slave Configuration Register 2 (SSLVC2) – Bits 7-0

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register Name
$C1	SS[1]	SS [0]	CPOL	CPHA	BC[3]	BC[2]	BC[1]	BC[0]	SSLVC2
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	1	0	0	0	

The serial peripheral interface slave configuration registers (SSLVC0, SSLVC1, SSLVC2) are 16-bit wide registers used to configure the operation of each individual slave device. Each time communication is to be established with a new slave device, the slave configuration register must be modified with an appropriate value for the new slave.

Updates to the slave configuration register may only made while the SPI module is idle (no SPI data transfer is in progress) otherwise updates are ignored.

Bits[15:10] - Reserved

These bits have no function and writes to them are ignored. Reads return zeros.

SE - SPI Slave Enable

The Slave Enable bit is used to enable or disable assertion of the slave chip select signals. If an alternative signal (GPIO bit for example) is to be used as a slave chip select then SE can be used to prevent any of the eight chip select outputs from being asserted during an SPI data transfer.

0 = Disable all slave chip select outputs during SPI transfers

1 = Enable the selected chip select output during SPI transfers

SS[2:0] - SPI Slave Select

The slave select bits are used to select the slave for the intended SPI data transfer. Slave selection is shown in Table 9-13 below.

Table 9-13. SPI Slave Select Configuration

SS[2]	SS[1]	SS[0]	Chip Select Asserted During Transfer
0	0	0	$\overline{SxS0}$
0	0	1	$\overline{SxS1}$
0	1	1	$\overline{SxS2}$
1	0	0	$\overline{SxS3}$
1	0	1	$\overline{SxS4}$
1	1	0	$\overline{SxS5}$
1	0	1	$\overline{SxS6}$
1	1	1	$\overline{SxS7}$

CPOL - SPI Clock Polarity

The Clock Polarity bit is used to set the polarity of the SPI clock output pin (SxCKO). When the clock polarity bit is cleared and SPI data is not being transferred, the SxCKO pin is in a steady-state low value. Refer to the SPI transfer timing diagrams in "SPI Transfer Waveform Diagrams" on page 9-11 for further details.

CPHA - SPI Clock Phase

The Clock Phase bit is used to configure the phase relationship between the SPI clock output and the input and output data.

Some SPI slave peripherals present their first data bit as soon as they are selected. In this format the slave peripheral does not need to be clocked in order to present their first data bit, resulting in the need to delay the clock output by half a cycle. This is achieved by clearing the CPHA bit to "0". An example of this is shown in "SPI Transfer Waveform Diagrams" on page 9-11.

Other SPI slave peripherals need to be clocked before the first active value is output from them, resulting in the need to advance the clock output by half a cycle. This is achieved by setting the CPHA bit to "1". An example of this is shown in "SPI Transfer Waveform Diagrams" on page 9-11.

BC[3:0] - SPI Bit Count

The Bit Count bits are used to set the number of data bits to be used during an SPI data transfer. The number of bits is programmable between 1 and 16. Selection of the number of bits is shown in Table 9-14 on page 9-7.

Table 9-14. SPI Bit Count Configuration

BC[3]	BC[2]	BC[1]	BC[0]	Number of Bits Used During Transfer
0	0	0	0	1
0	0	0	1	2
0	0	1	0	3
0	0	1	1	4
0	1	0	0	5
0	1	0	1	6
0	1	1	0	7
0	1	1	1	8
1	0	0	0	9
1	0	0	1	10
1	0	1	0	11
1	0	1	1	12
1	1	0	0	13
1	1	0	1	14
1	1	1	0	15
1	1	1	1	16

9.1.3 Write Data Registers

Table 9-15. Write Data Register 0 (SWD0) – Bits 15-8

Address	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Register Name
$B2	WD[15]	WD[14]	WD[13]	WD[12]	WD[11]	WD[10]	WD[9]	WD[8]	SWD0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	0	0	0	

Table 9-16. Write Data Register 0 (SWD0) – Bits 7-0

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register Name
$B2	WD[7]	WD[6]	WD[5]	WD[4]	WD[3]	WD[2]	WD[1]	WD[0]	SWD0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	0	0	0	

Table 9-17. Write Data Register 1 (SWD1) – Bits 15-8

Address	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Register Name
$BA	WD[15]	WD[14]	WD[13]	WD[12]	WD[11]	WD[10]	WD[9]	WD[8]	SWD1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	0	0	0	

Table 9-18. Write Data Register 1 (SWD1) – Bits 7-0

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register Name
$BA	WD[7]	WD[6]	WD[5]	WD[4]	WD[3]	WD[2]	WD[1]	WD[0]	SWD1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	0	0	0	

Table 9-19. Write Data Register 2 (SWD2) – Bits 15-8

Address	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Register Name
$C2	WD[15]	WD[14]	WD[13]	WD[12]	WD[11]	WD[10]	WD[9]	WD[8]	SWD2
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	0	0	0	

Table 9-20. Write Data Register (2 SWD2) – Bits 7-0

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register Name
$C2	WD[7]	WD[6]	WD[5]	WD[4]	WD[3]	WD[2]	WD[1]	WD[0]	SWD2
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	0	0	0	

The serial peripheral interface write data registers (SWD0, SWD1, SWD2) are 16-bit read/write registers used to store transmit data.

The most-significant bit of the SWD is always transmitted first. For transfers sizes of less than 16-bits, the transmitted data bit is sourced from the most-significant bit position as write data is shifted out bit-by-bit on the SxMDI pin. For example, an 8-bit write data transfer would result in WD[7] being transmitted first and WD[0], the least-significant bit, being transmitted last.

Writes to SWD are ignored while SPI transfers are in progress.

Data written to the SWD may be read prior to being transmitted. The data content of the SWD will change during an SPI data transfer as data is shifted out.

9.1.4 Read Data Registers

Table 9-21. Read Data Register 0 (SRD0) – Bits 15-8

Address	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Register Name
$B3	RD[15]	RD[14]	RD[13]	RD[12]	RD[11]	RD[10]	RD[9]	RD[8]	SRD0
Read/Write	R	R	R	R	R	R	R	R	
Reset	0	0	0	0	0	0	0	0	

Table 9-22. Read Data Register 0 (SRD0) – Bits 7-0

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register Name
$B3	RD[7]	RD[6]	RD[5]	RD[4]	RD[3]	RD[2]	RD[1]	RD[0]	SRD0
Read/Write	R	R	R	R	R	R	R	R	
Reset	0	0	0	0	0	0	0	0	

Table 9-23. Read Data Register 1 (SRD1) – Bits 15-8

Address	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Register Name
$BB	RD[15]	RD[14]	RD[13]	RD[12]	RD[11]	RD[10]	RD[9]	RD[8]	SRD1
Read/Write	R	R	R	R	R	R	R	R	
Reset	0	0	0	0	0	0	0	0	

Table 9-24. Read Data Register 1 (SRD1) – Bits 7-0

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register Name
$BB	RD[7]	RD[6]	RD[5]	RD[4]	RD[3]	RD[2]	RD[1]	RD[0]	SRD1
Read/Write	R	R	R	R	R	R	R	R	
Reset	0	0	0	0	0	0	0	0	

Table 9-25. Read Data Register 2 (SRD2) – Bits 15-8

Address	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Register Name
$C3	RD[15]	RD[14]	RD[13]	RD[12]	RD[11]	RD[10]	RD[9]	RD[8]	SRD2
Read/Write	R	R	R		R	R	R	R	
Reset	0	0	0	0	0	0	0	0	

Table 9-26. Read Data Register 2 (SRD2) – Bits 7-0

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register Name
$C3	RD[7]	RD[6]	RD[5]	RD[4]	RD[3]	RD[2]	RD[1]	RD[0]	SRD2
Read/Write	R	R	R	R	R	R	R	R	
Reset	0	0	0	0	0	0	0	0	

The serial peripheral interface read data registers (SRD0, SRD1, SRD2) are 16-bit read-only registers used to store receive data.

The most-significant bit is always received first, being shifted initially into SRD[0]. For transfer sizes of less than 16 bits, completion of the read data transfer results in the most-significant be in the SRD bit position corresponding to the number of bits received. For example, an 8-bit transfer would result in the most-significant bit residing in SRD[7] and the least-significant bit in SRD[0]. The least-significant bit always resides in SRD[0].

Data in the SRD register is only valid while the SPI interface is idle.

9.1.5 Serial Bit Rate Count Preset Registers

Table 9-27. Serial Bit Rate Count Preset Register 0 (SBPR0) – Bits 15-0

Address	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Register Name
$B4	SBPR[15]	SBPR[14]	SBPR[13]	SBP[12]	SBPR[11]	SBPR[10]	SBPR[9]	SBPR[8]	SBPR0
Read/Write	R	R	R	R	R	R	R	R	
Reset	0	0	0	0	0	0	0	0	

Table 9-28. Serial Bit Rate Count Preset Register 0 (SBPR0) – Bits 7-0

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register Name
$B4	SBPR[7]	SBPR[6]	SBPR[5]	SBPR[4]	SBPR[3]	SBPR[2]	SBPR[1]	SBPR[0]	SBPR0
Read/Write	R	R	R	R	R	R	R	R	
Reset	0	0	0	0	0	0	0	0	

Table 9-29. Serial Bit Rate Count Preset Register 1 (SPBR1) – Bits 15-0

Address	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Register Name
$BC	SBPR[15]	SBPR[14]	SBPR[13]	SBP[12]	SBPR[11]	SBPR[10]	SBPR[9]	SBPR[8]	SBPR1
Read/Write	R	R	R	R	R	R	R	R	
Reset	0	0	0	0	0	0	0	0	

Table 9-30. Serial Bit Rate Count Preset Register 1 (SB{R1) – Bits 7-0

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register Name
$BC	SBPR[7]	SBPR[6]	SBPR[5]	SBPR[4]	SBPR[3]	SBPR[2]	SBPR[1]	SBPR[0]	SBPR1
Read/Write	R	R	R	R	R	R	R	R	
Reset	0	0	0	0	0	0	0	0	

Table 9-31. Serial Bit Rate Count Preset Register 2 (SBPR2) – Bits 15-0

Address	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Register Name
$C4	SBPR[15]	SBPR[14]	SBPR[13]	SBP[12]	SBPR[11]	SBPR[10]	SBPR[9]	SBPR[8]	SBPR2
Read/Write	R	R	R	R	R	R	R	R	
Reset	0	0	0	0	0	0	0	0	

Table 9-32. Serial Bit Rate Count Preset Register 2 (SBPR2) – Bits 7-0

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register Name
$C4	SBPR[7]	SBPR[6]	SBPR[5]	SBPR[4]	SBPR[3]	SBPR[2]	SBPR[1]	SBPR[0]	SBPR2
Read/Write	R	R	R	R	R	R	R	R	
Reset	0	0	0	0	0	0	0	0	

The serial peripheral interface serial bit rate count preset registers (SBPR0, SBPR1, SBPR2) are 16-bit read/write registers used to set the SPI transfer clock rate.

The SPI transfer clock frequency is defined as $f_{SYS}/2n$, where:

- f_{SYS} is the system clock frequency
- n is the integer value of the unsigned binary value stored in the SBPR, where SBPR[15] is the most-significant bit.

Writing values of either $0000 or $0001 to the SBPR will result in the value $0004 being automatically loaded into the register. The minimum value that may be written to the SBPR is $0002, resulting in a maximum clock rate of $f_{SYS}/4$. The lowest clock rate is $f_{SYS}/131072$.

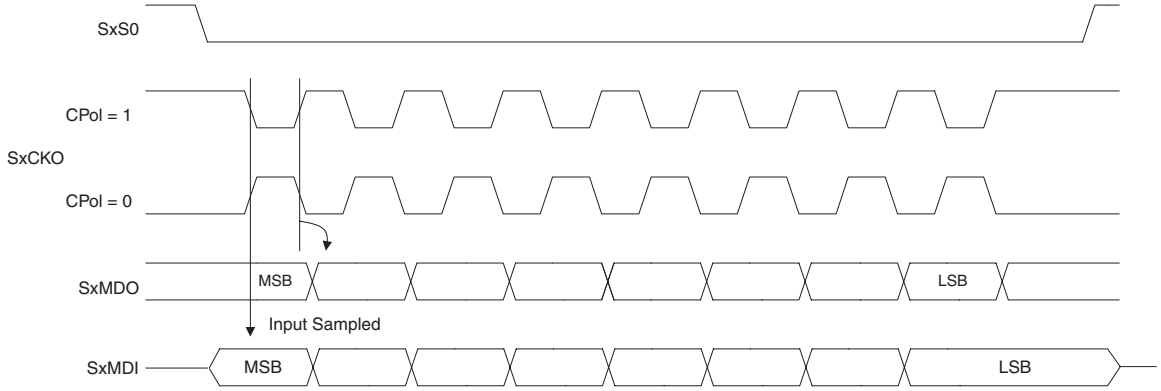
9.2 SPI Transfer Waveform Diagrams

The following are examples of 8-bit SPI transfers using both phase settings (CPHA = 0 and CPHA = 1).

9.2.1 Example SPI Transfer: CHPA = 0

When CPHA is cleared (set to "0"), data is shifted into the SPI module (MSB first) using the first active edge of the SxCKO clock. Figure 9-1 on page 9-12 shows an example of a non-continuous 8-bit transfer with CPHA cleared. The selected slave is 0. In this example, the master clock output has been shown for both CPOL settings.

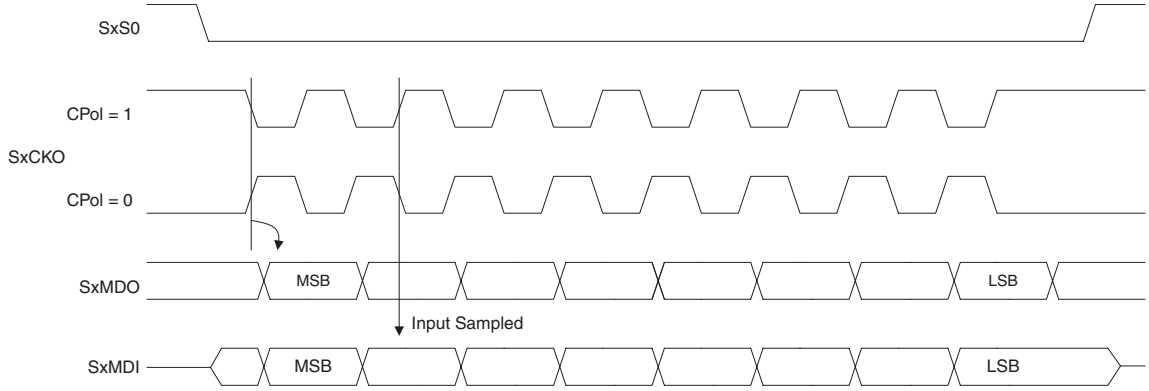
Figure 9-1. Example SPI 8-Bit Transfer: CPHA = 0



9.2.2 Example SPI Transfer: CHPA = 1

When CPHA is set (set to "1"), data is shifted into the SPI module (MSB first) using the second active edge of the SxCKO clock. Figure 9-2 shows an example of a non-continuous 8-bit transfer with CPHA set. The selected slave is 0. In this example, the master clock output has been shown for both CPOL settings.

Figure 9-2. Example SPI 8-Bit Transfer: CPHA = 1



Chapter 10. Caches

The VS2000 has two identical 2-way set-associative caches, one for program bus accesses and one for data bus accesses. Each cache module provides single-cycle read accesses for cached data and implements a write-back system for write accesses. Line lock and flush functions are also provided.

10.1 Cache Features

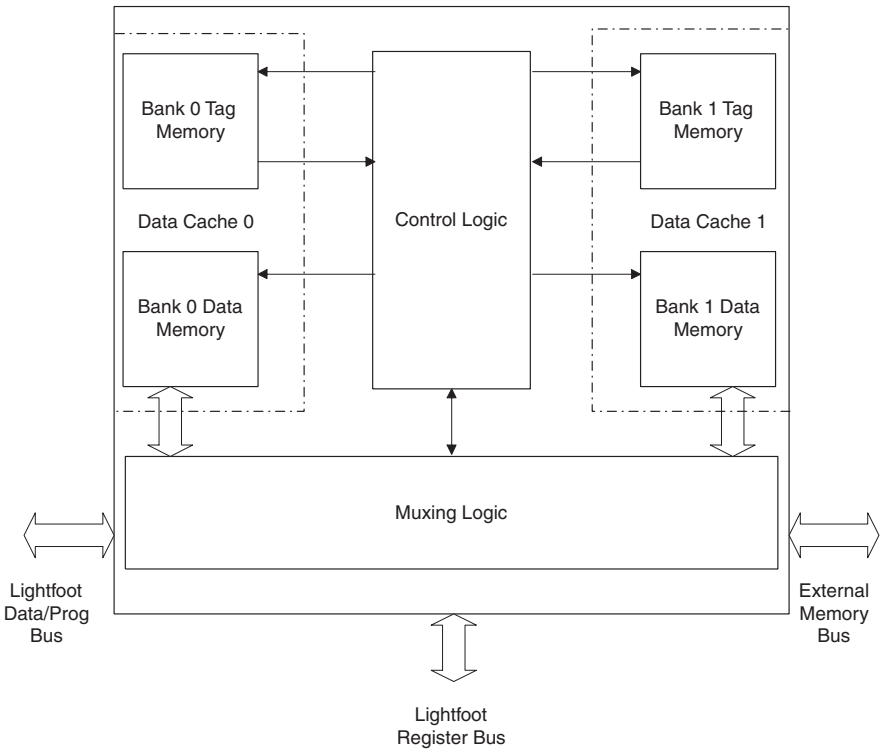
The cache has the following features:

- 4 KB cache in two banks of 2 KB
- Single-cycle read-hit accesses
- Write-back system to prevent unnecessary write accesses to slow memory
- Line lock facility to keep code in the cache
- Manual line flush functionality

10.2 Architecture

The internal architecture of the cache module is shown in Figure 10-1.

Figure 10-1. Cache Unit Block Diagram



The two banks of the cache each contain a 2 KB 32-bit data RAM and a 192-byte 12-bit tag RAM. The control logic block implements the state machine which controls the line fills, flushes, and implements all the commands. The multiplexing logic is responsible for implementing the bypass function and also for multiplexing the signals of the two main data buses with the data and tag memory.

10.3 Operation

In normal operation the cache is invisible to the programmer and the only action needed is to enable it on system start-up (see "Control Register" on page 10-4).

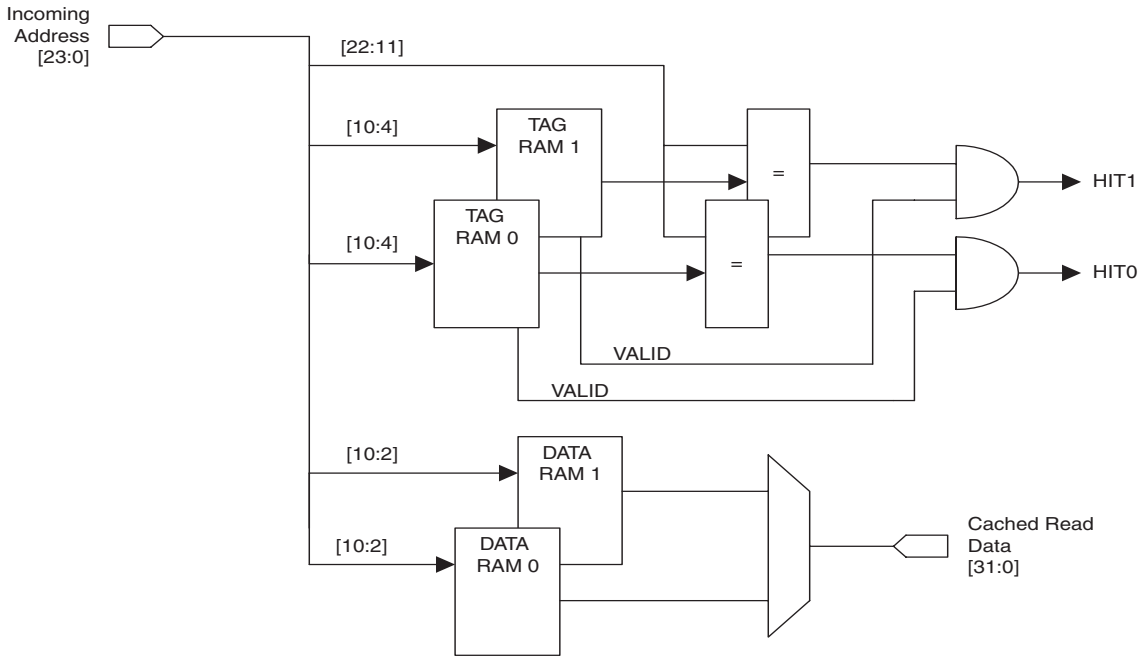
The cache starts up in bypass mode. This means that any access will be passed straight through to main memory. The cache will also be in bypass mode if the most-significant bit (bit 23) of the address is set. This means that data in the top half of the memory map cannot be cached.

The following subsections explain how the cache operates in the four conditions: read miss, read hit, write miss, and write hit.

10.3.1 Read

When a read request is made to a cached memory area (i.e., a transfer in the first 8 MB of memory, between $000000 and $7FFFFF) and the cache is enabled, the cache checks to see whether the requested data is currently cached. The check is performed by comparing the contents of the tag RAM for the given line address (represented by bits 10 through 4 of the incoming address) with the incoming address itself (bits 22 through 11 representing the location of the 2 KB potentially cached area). If data corresponding to the external read transfer is in the cache, and it is valid, then a hit signal is generated. The hit signal is used to select the source of the read data returned to the host (this is either from bank 0 or bank 1).

Figure 10-2. Cache Read Mechanism



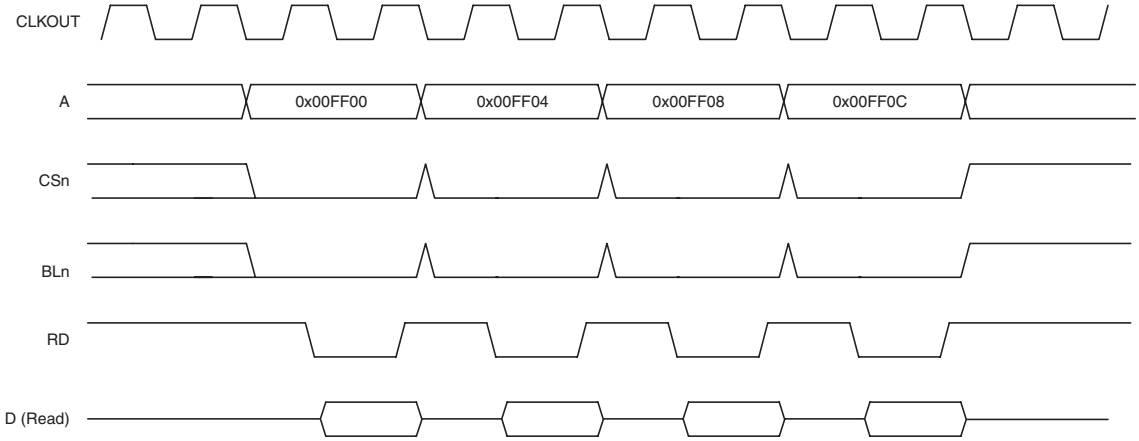
10.3.1.1 Read Miss

If the cache does not contain the required data then a line fill is performed. Figure 10-3 shows the timing for a cache line fill in 32-bit mode resulting from a cache read miss at address $00FF08. An cache line fill in 8-bit mode would result in 16 single byte transfers. This means that the 16-byte segment of memory that contains the requested data is read from main memory. The line is stored into one of the banks of data RAM according to the following system:

1. If the lock bit is set for the line in question, use bank 1 (no examination of the "least-recently used" bit is required)
2. If the line is empty in both banks, fill bank 0
3. If the line in question is full in one bank, use the other bank
4. If the line is full in both banks but the lock bit is clear, then examine the LRU (least-recently used) bit and overwrite the line in the bank that was least-recently used

Note that if the data to be overwritten (if the line is already full) is marked as "dirty", a line flush will be performed first to update the relevant section of main memory.

Figure 10-3. Cache Line Fill Timing



10.3.1.2 Read Hit

If the cache contains the required data, data is then output from the data RAM in time for the host processor to read the data in the same cycle as the generated request. Cache read hits therefore result in zero wait-state operation. The least-recently used (LRU) bit is updated accordingly in the following cycle, reflecting fact that the read hit occurred.

10.3.2 Write

The cache uses a write-back system for data writes. This means that if data is written to a line which is stored in cache then no external access is made to main memory to keep it updated. The data in main memory is effectively "dirty" or not in sync with the cached copy.

10.3.2.1 Write Miss

Dealing with a write miss is much the same as dealing with a read miss. The current copy of the data in question is fetched from main memory via a 16-byte line fill. The data word being written is then copied over the appropriate section of the cached data thus making the line instantly "dirty". The LRU bit will be updated accordingly.

10.3.2.2 Write Hit

As explained above, to avoid lengthy accesses to main memory write accesses are not always copied through to main memory. In the case of a write hit, a write to a line of data which is currently stored in the cache, only the cached copy of the data is updated. The "dirty" bit for the line is set and the LRU bit for the given index is updated accordingly. Writes to the cache take two clock cycles.

10.4 Cache Register Set

10.4.1 Control Register

The cache uses one control register to implement all control functionality. The functions that can be accessed through this register are as follows:

- Enable/disable the cache
- Write/read a tag
- Flush a line
- Clear all control bits (effectively clearing the cache)

The following tables describes the control register fields for read and write:

Table 10-1. Instruction Cache Control Register (ICCSR) – Read

Address	Bits 31:28	Bit 27	Bit 26	Bit 25	Bit 24	Bit 23	Bits 22:12	Bits 11:0	Register Name
$70	-	ENABLE	VALID	DIRTY	LOCK	READY	-	TAG	ICCSR
R/W	R	R	R	R	R	R	R	R	
Reset	0	0	0	0	0	1	0	$00	

Table 10-2. Instruction Cache Control Register (ICCSR) – Write

Address	Bit 31	Bit 30	Bit 29	Bit 28	Bits 27:16	Bits 10:4	Bits 3:0	Register Name
$70	VALID	DIRTY	LOCK	BANK	TAG	INDEX	COMMAND	ICCSR
R/W	W	W	W	W	W	W	W	
Reset								

Table 10-3. Data Cache Control Register (DCCSR) – Read

Addess	Bits 31:28	Bit 27	Bit 26	Bit 25	Bit 24	Bit 23	Bits 22:12	Bits 11:0	Register Name
$71	-	ENABLE	VALID	DIRTY	LOCK	READY	-	TAG	DCCSR
R/W	R	R	R	R	R	R	R	R	
Reset	0	0	0	0	0	1	0	$00	

Table 10-4. Data Cache Control Register (DCCSR) – Write

Address	Bit 31	Bit 30	Bit 29	Bit 28	Bits 27:16	Bits 10:4	Bits 3:0	Register Name
$71	VALID	DIRTY	LOCK	BANK	TAG	INDEX	COMMAND	DCCSR
R/W	W	W	W	W	W	W	W	
Reset								

The cache control commands, shown in Table 10-5, are encoded in the 4-bit COMMAND field of the Cache Control Register.

Table 10-5. COMMAND Field of the Cache Control Registers

Command	Opcode	Function
Disable	0000	Disable the cache
Enable	0001	Enable the cache
Write Tag	0010	Writes the tag value to the specified line
Read Tag	0011	Reads the tag value from a specified line
Flush Line	0100	Flushes the specified data line out to main memory
Clear Control Bits	0101	Clears all of the lock, valid and dirty bits

10.4.1.1 Enabling the Cache

The cache starts up in a disabled state. To enable the cache, write the *Enable* command to the cache control register. The cache will switch to the enabled state the next time the processor-side bus is inactive.

10.4.1.2 Disabling the Cache

The cache must be disabled before commands are issued to it. To do this write the *Disable* command to the control register. The cache will switch to the disabled state the next time the processor-side bus is inactive.

10.4.1.3 Read Tag

To read the tag address and control bits associated with a particular index of a bank in question, the *Read Tag* command is written along with an index address and a bank bit. The control register's READY bit must then be polled to determine when the command has been completed. The required data can then be read back as per the read field specification above.

10.4.1.4 Write Tag

The *Write Tag* command writes the a tag address to the tag RAM in the specified bank while updating the relevant control bits: Lock, Dirty, and Valid. The control register READY bit must be polled after the command has been issued before performing any other commands or re-enabling the cache.

10.4.1.5 Flush Line

The *Flush Line* command flushes a dirty line out to main memory. The line is specified according to bank and index address. Note that if the line is not dirty, no flush will be performed.

10.4.1.6 Clear Control Bits

The *Clear Control Bits* command resets all the control bits for all lines of the cache. This has the effect of resetting the cache. Any previously cached data will be lost as without the valid bits set, the data will be re-fetched from main memory.

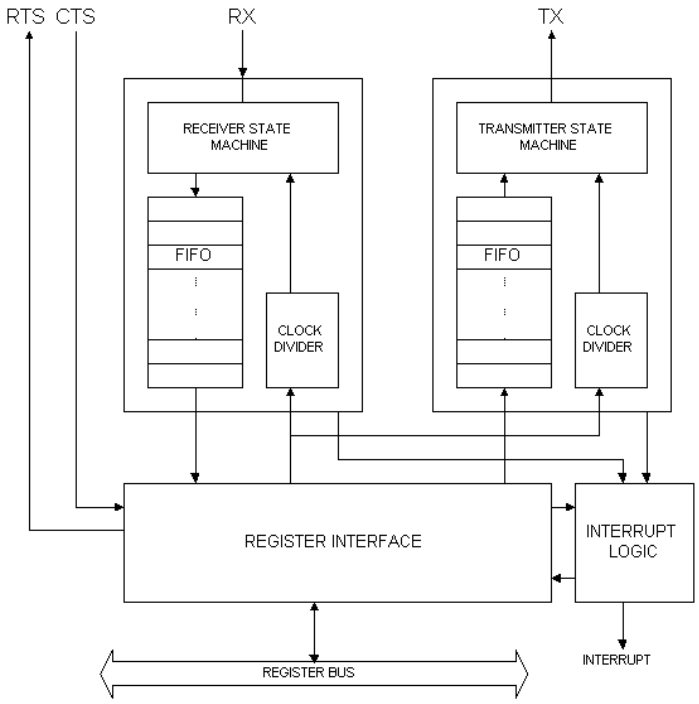
Chapter 11. UARTs

UART0 and UART1 are 16550-compatible, full-duplex asynchronous communications modules, the design was sourced from opencores (see www.opencores.org for more information). This module's description is based on the opencores UART documentation. The two UARTs on the device share their I/O ports with GPIO port E (UART1: PE7, PE5, PE3 and PE1; UART0: PE6, PE4, PE2 and PE0). Both UARTs support the following features:

- Programmable number of data bits: 5, 6, 7 and 8 data bits supported
- Programmable number of stop bits: 1, 2 and 1.5 (5 data bits only)
- Programmable baud rate
- 16-byte FIFO on transmit and receive
- Hardware flow control (CTS/RTS) built-in

The transmitter and receiver are functionally independent but use the same data format and baud rate. The user will have to provide external level-conversion buffers to convert the 0-3.3V signals to RS-232 standard levels (typically ±5V or ±12V).

Figure 11-1. UART Block Diagram



Each UART has four I/O signals associated with it:

- TX - Transmit Data
- RX - Receive Data

- RTS - Ready to Send, flow control line out
- CTS - Clear to Send, flow control line in

Baud rate is selected via a programmable divisor which divides down the system clock. This allows serial communication at almost any baud rate irrespective of system clock frequency.

Both the receiver and transmitter have 16-byte deep FIFOs allowing up to 16 characters to be transmitted or received before interaction with Lightfoot core is required.

11.1 Operation

11.1.1 Transmit Operation

Data is transmitted via the TX pin from the Transmitter Shift Register. The shift register gets its data from the 16-byte FIFO which is loaded via writes to the Transmit Data Register.

There are two bits in the Line Status Register, TXFMT and TXMT, which indicate the state of the transmitter. TXFMT indicates that the FIFO is empty. TXMT indicates that the Transmitter Shift Register is empty. When TXFMT is asserted, up to 16 characters can safely be written to the Transmitter Data Register. When TXMT is asserted, it indicates that the transmitter is idle.

The transmitter can be run in interrupt mode by enabling the THRE interrupt (THRIEx in UIERx). This interrupt will be asserted when the transmitter FIFO is empty. Note that the Transmitter Shift Register can still be processing a character. By linking the THRE interrupt to the TXFMT bit, the maximum data rate can be achieved while the FIFO can be re-loaded when the transmitter is still sending the last character from the shift register.

A break character can be transmitted by asserting the relevant BREAK bit in the Line Control Register for the required amount of time.

11.1.2 Receive Operation

Data is received from the RX pin into the receiver shift register. When a byte is received and checked for parity and stop bits, it is loaded into the receiver FIFO. Reading from the Receive Data Register returns the first byte stored in the FIFO.

There are five bits in the Line Status Register that indicate the status of received characters. DATRD is asserted when there are one or more characters in the FIFO. The other four bits (OVERN, PARITY, FRAME, and BRKE) each indicate an error condition associated with the character at the top of the FIFO, the next character that will be read from the Receive Data Register.

The receiver has three associated interrupt conditions. Which condition triggered the interrupt can be determined by reading the Interrupt Identification Register (UIIR0 and UIIR1).

Note: The Received Data Available (RDA) and Time-Out Indication (TOI) conditions are both enabled by bit RDAIEx in the Interrupt Enable Register (UIER0 or UIER1). The Receiver Line Status (RLS) interrupt is enabled by bit RLSIEx.

The RDA interrupt condition indicates that the receive FIFO has reached the programmed trigger level.

The TOI interrupt condition occurs when a character has been received, the FIFO has not reached its trigger level, and no activity has occurred for four character times.

The RLS interrupt condition indicates one of four possible conditions:

1. Overrun – This occurs when the FIFO and the Receiver Shift Register are full and a character is received. The data in the shift register will be overwritten with the incoming data but the FIFO will not be affected.
2. Parity – This indicates that the character at the top of the FIFO was received with a parity error.
3. Frame – This indicates that the character at the top of the FIFO was received with a framing error. A framing error occurs when the stop bit is not valid indicating a mismatch in the number of data bits being sent or a baud rate mismatch with the device transmitting characters.
4. Break – This indicates that the character at the top of the FIFO was received as a break character.

As mentioned above the LSR indicates which of these four possible conditions caused the RLS interrupt to occur.

11.1.3 Configuring the UART

After a reset, the UART is configured with eight data bits, one stop bit, no parity mode and the divisor is set to zero. This ensures no activity takes place, receive or transmit, until the UART is set up. The FIFO trigger level is set to 14 bytes.

The data bits, stop bits, and parity settings are set via the Line Control Registers (ULCR0 and ULCR1). Bit 7 of the LCR is the Divisor Latch Access bit (DLAB). When set high, this bit allows the first two addresses for each UART ($80 and $81 for UART0, $88 and $89 for UART1) to access the two bytes of the Divisor Register.

Note: The Divisor Register should be loaded with the upper-byte first.

```
UART0_DIV_UP      .DEFINE        $01
UART1_DIV_LOW     .DEFINE        $23

;Set the Divisor Latch Access bit in ULCR0
          cnsti          $80
          sr             $83
;Store $01 to the upper byte of the Divisor Register
          cnsti          UART0_DIV_UP
          sr             $81
;Store $23 to the lower byte of the Divisor Register
          cnsti          UART0_DIV_LOW
          sr             $80
;Clear DLAB0 and set to 8 data bits, 1 stop bit, no parity
          cnsti          $03
          sr             $83
```

Setting up the data format and baud rate for UART0 in Lightfoot Assembler

The formula for calculating the baud rate divisor is:

$$\text{Divisor} = \frac{\text{System Clock Speed}}{16 \times \text{Required Baud Rate}}$$

The above formula calculates an exact baud rate divisor. To round the baud rate divisor to an integer value, the following permissible rounding errors must be taken into consideration:

- Rounding-up error (1 stop-bit): 0.625%
- Rounding-up error (2 stop-bits): 0.588%
- Rounding-down error: 5.26%

The UART can only be guaranteed to perform correctly if the rounding error is within these limits.

The receive FIFO trigger level is set in the FIFO Control Register (FCR) and determines the number of bytes in the receive FIFO that will cause a Received Data Available (RDA) interrupt to occur.

11.2 UART Register Set

11.2.1 Receive/Transmit Data Registers

Table 11-1. UART0 Data Register

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register Name
$80	D7	D6	D5	D4	D3	D2	D1	D0	UDR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	0	0	0	

Table 11-2. UART1 Data Register

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register Name
$88	D7	D6	D5	D4	D3	D2	D1	D0	UDR1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	0	0	0	

The data registers are used to transmit and receive data to/from the UARTs. A write to these registers pushes the data written into the transmit FIFO. A read from these registers reads back the first byte in the receive FIFO.

11.2.2 Interrupt Enable Registers

These registers allow interrupt generation to be enabled/disabled for each possible UART interrupt source.

Table 11-3. UART0 Interrupt Enable Register

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register Name
$81	-	-	-	--	MSIE0	RLSIE0	THRIE0	RDAIE0	UIER0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	0	0	0	

Table 11-4. UART1 Interrupt Enable Register

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register Name
$89	-	-	-	-	MSIE1	RLSIE1	THRIE1	RDAIE1	UIER1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	0	0	0	

Bits [7:4] - Reserved

These bits have no function, writes are ignored and reads return zero.

MSIEx - Modem Status Interrupt

0 = Disabled

1 = Enabled

RLSIEx - Receiver Line Status Interrupt

0 = Disabled

1 = Enabled

THRIEx - Transmitter Holding Register Empty Interrupt

0 = Disabled

1 = Enabled

RDAIEx - Received Data Available Interrupt

0 = Disabled

1 = Enabled

11.2.3 Interrupt Identification Registers

The UART interrupt identification registers (UIIR) enable the programmer to determine the current highest priority pending interrupt.

Table 11-5. UART0 Interrupt Identification Register

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register Name
$81	-	-	-	-	INT2	INT1	INT0	IPND0	UIIR0
R/W	R	R	R	R	R	R	R	R	
Reset	1	1	0	0	0	0	0	1	

Table 11-6. UART1 Interrupt Identification Register

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register Name
$89	-	-	-	-	INT2	INT1	INT0	IPND1	UIIR1
R/W	R	R	R	R	R	R	R	R	
Reset	1	1	0	0	0	0	0	1	

Bits [7:4] - Reserved

These bits have no function, writes are ignored and reads return zero.

Table 11-7 on page 11-6 displays the list of possible interrupts along with the bits they enable, their priority, source, and reset control:

Table 11-7. Possible UART Interrupts

Bit 3	Bit 2	Bit 1	Priority	Interrupt Type	Interrupt Source	Interrupt Reset Control
0	1	1	1st	Receiver line status	Parity, Overrun or Framing Errors or Break Interrupt	Reading the Line Status Register
0	1	0	2nd	Receiver data available	FIFO trigger level reached	FIFO drops below trigger level
1	1	0	2nd	Timeout indication	At least one character is present in the FIFO but no character has been input to the FIFO or read from it for the last four character times	Reading from the FIFO (Receiver Buffer Register)
0	0	1	3rd	Transmitter Holding Register empty	Transmitter Holding Register empty	Writing to the Transmitter Holding Register or reading the IIR
0	0	0	4th	ModemsStatus	CTS	Reading the Modem Status Register.

11.2.4 FIFO Control Registers

The FIFO Control Registers allow selection of the receiver FIFO trigger level (the number of bytes in the receive FIFO which will cause a Received Data Available interrupt). Bits 2-1 control the reset to the TX and RX FIFOs allowing them to be reset under software control.

Table 11-8. UART0 FIFO Control Register

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register Name
$82	TRIG01	TRIG00	RSVD	RSVD	RSVD	TXCLR0	RXCLR0	RSVD	UFCR0
R/W	W	W	W	W	W	W	W	W	
Reset	1	1	0	0	0	0	0	0	

Table 11-9. UART1 FIFO Control Register

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register Name
$8A	TRIG11	TRIG10	RSVD	RSVD	RSVD	TXCLR1	RXCLR1	RSVD	UFCR1
R/W	W	W	W	W	W	W	W	W	
Reset	1	1	0	0	0	0	0	0	

Bits [5:3, 0] - Reserved

These bits have no function, writes are ignored and reads return zero.

TRIGx[1:0] - Define the Receiver FIFO Interrupt Trigger Level

00 = 1 byte

01 = 4 bytes

10 = 8 bytes

11 = 14 bytes

TXCLRx

Writing a "1" to bit 2 clears the transmitter FIFO and resets its logic. The shift register is not cleared and transmitting of the current character continues.

RXCLRx

Writing a "1" to bit 1 clears the receiver FIFO and resets its logic. However, it doesn't clear the Shift Register (receiving of the current character continues).

11.2.5 Line Control Registers

The Line Control Register allows the specification of the format of the asynchronous data communication used. Bit 7 in the register also allows access to the Divisor Registers which define the baud rate. Reading from the register is allowed to check the current settings.

Table 11-10. UART0 Line Control Register

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register Name
$83	DLAB0	BREAK0	PRSTK0	PRSEL0	PREN0	STPBT0	DTBIT01	DTBIT00	ULCR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	0	1	1	

Table 11-11. UART1 Line Control Register

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register Name
$8B	DLAB1	BREAK1	PRSTK1	PRSEL1	PREN1	STPBT1	DTBIT11	DTBIT10	ULCR1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	0	1	1	

DLABx - Divisor Register Access Bit

1 = The divisor registers can be accessed

0 = The normal registers are accessed

BREAKx - Break Control Bit

1 = The serial out is forced into logic 0 (break state)

0 = Break is disabled

PRSTKx - Stick Parity Bit

0 = Stick parity disabled

1 = If bits 3 and 4 are high, the parity bit is transmitted and checked as logic 0. If bit 3 is high and bit 4 is low, the parity bit is transmitted and checked as logic "1".

PRSELx - Even Parity Select

0 = Odd number of ones are transmitted and checked in each word (data and parity combined). If the data contains an even number of ones, the parity bit is high.

1 = Even number of ones are transmitted in each word

PRENx - Parity Enable

0 = No parity

1 = Parity bit is generated on each outgoing character and is checked on each incoming one.

STPBTX - Number of Generated Stop Bits

0 = 1 stop bit

1 = 1.5 stop bits when 5-bit character length selected and 2 bits otherwise

Note: The receiver always checks the first stop bit only.

DTBITX[1:0] - Number of Bits in Each Character

00 = 5 bits

01 = 6 bits

10 = 7 bits

11 = 8 bits

11.2.6 Modem Control Registers

The Modem Control Registers displays the current state of the modem control lines. The four LSBs provide flags that tells the programmer that one of the modem status lines have changed state. These bits are set high when a change in corresponding line has been detected and they are reset when the register is read.

Table 11-12. UART0 Modem Control Register

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register Name
$84	-	-	-	LPBK0	-	-	RTS0	-	UMCR0
R/W	W	W	W	W	W	W	W	W	
Reset	0	0	0	0	0	0	0	0	

Table 11-13. UART1 Modem Control Register

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register Name
$8C	-	-	-	LPBK1	-	-	RTS1	-	UMCR1
R/W	W	W	W	W	W	W	W	W	
Reset	0	0	0	0	0	0	0	0	

Bits [7:5, 3:2, 0] - Reserved

These bits have no function, writes are ignored and reads return zero.

LPBKx - Loop-Back Mode

0 = Normal operation

1 = Loop-back mode. When in loop-back mode, the TX pin is set to logic 1. The signal of the Transmitter Shift Register is internally connected to the input of the receiver shift register. In addition, the RTS output is connected to the CTS input.

RTSx - Request to Send ($\overline{RTS}$) Signal Control

0 = $\overline{RTS}$ is deasserted

1 = $\overline{RTS}$ is asserted

11.2.7 Line Status Registers

Table 11-14. UART0 Line Status Register

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register Name
$85	ERROR0	TXMT0	TXFMT0	BRKE0	FRAME0	PARITY0	OVERN0	DATRD0	ULSR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	0	0	0	

Table 11-15. UART1 Line Status Register

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register Name
$8D	ERROR1	TXMT1	TXFMT1	BRKE1	FRAME1	PARITY1	OVERN1	DATRD1	ULSR1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	0	0	0	

ERRORx - Error Indicator

1 = At least one parity error, framing error, or break indication has been received and is inside the FIFO. The bit is cleared upon reading from the register.

0 = No parity errors, framing errors, or break indications received

TXMTX - Transmitter Empty Indicator

1 = Both the transmit FIFO and Transmitter Shift Register are empty. The bit is cleared when data is being been written to the transmitter FIFO.

0 = The transmit FIFO and Transmitter Shift Register are not empty

TXFMTX - Transmit FIFO Empty

1 = The transmit FIFO is empty. A Transmitter Holding Register Empty interrupt is generated. The bit is cleared when data is being been written to the transmitter FIFO.

0 = The transmit FIFO is not empty

BRKEx - Break Interrupt (BI) Indicator

1 = A break condition has been reached in the current character. The break occurs when the line is held in logic 0 for a time of one character (start bit + data + parity bit + stop bit). In that case, one zero character enters the FIFO and the UART waits for a valid start bit to receive the next character. The bit is cleared upon reading from the register and generates a Receiver Line Status interrupt.

FRAMEx - Framing Error (FE) Indicator

1 = The received character at the top of the FIFO did not have a valid stop bit. This usally means that all of the following data is corrupt. The bit is cleared upon reading from the register and generates a Receiver Line Status interrupt.

0 = No framing error in the current character

0 = No break condition in the current character

PARITYx - Parity Error (PE) Indicator

1 = The character that is currently at the top of the FIFO has been received with a parity error. The bit is cleared upon reading from the register and generates a Receiver Line Status interrupt.

0 = No parity error in the current character

OVERNx - Overrun Error (OE) Indicator

1 = The FIFO is full and another character has been received in the receiver shift register. If another character is starting arrives, it will overwrite the data in the shift register but the FIFO will remain intact. This bit is cleared upon reading from the register and generates a Receiver Line Status interrupt.

0 = No overrun state

DATRDx - Data Ready (DR) Indicator

0 = No characters in the FIFO

1 = At least one character has been received and is in the FIFO

11.2.8 Modem Status Registers

Table 11-16. UART0 Modem Status Register

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register Name
$86	-	-	-	CCTS0	-	-	-	DCTS0	UMSR0
R/W	R	R	R	R	R	R	R	R	
Reset	0	0	0	0	0	0	0	0	

Table 11-17. UART1 Modem Status Register

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register Name
$8E	-	-	-	CCTS1	-	-	-	DCTS1	UMSR1
R/W	R	R	R	R	R	R	R	R	
Reset	0	0	0	0	0	0	0	0	

Bits [7:5, 3:1] - Reserved

These bits have no function, writes are ignored and reads return zero.

CCTSx - CTS Input

Linked to RTS output in loop-back mode.

DCTSx - Delta Clear To Send (DCTS) Indicator

1 = The CTS line has changed its state

Chapter 12. Ethernet MAC

The VS2000 has an integrated Ethernet Media Access Controller (MAC) which was sourced from opencores (see www.opencores.org for more information). This module's description is based on the opencores Ethernet MAC documentation. The MAC is capable of operation at 10Mbps or 100Mbps in both half- and full-duplex modes. The MAC provides a Media Independent Interface (MII), allowing direct connection to most Ethernet PHY devices. A Direct Memory Access (DMA) controller trans fers data between the MII interface and Ethernet transmit and receive data frame buffers.

The Ethernet MAC consists of four functional modules:

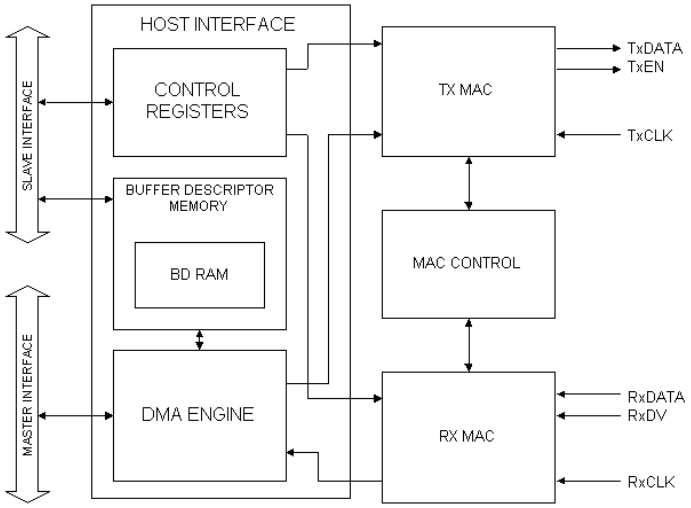
- The host interface connects the Ethernet core to the Lightfoot core and to frame buffer memory using DMA transfers. Registers are also part of the host interface.
- The TX Ethernet MAC performs transmit functions.
- The RX Ethernet MAC performs receive functions.
- The MAC Control Module performs full duplex flow control functions.

All modules combined deliver full-function 10/100 Mbps Media Access Control. The Ethernet MAC can operate in half- or full-duplex mode and is based on the CSMA/CD (Carrier Sense Multiple Access/Collision Detection) protocol.

When a station wants to transmit in half-duplex mode, it must observe the activity on the media (Carrier Sense). As soon as the media is idle (no transmission), any station can start transmitting (Multiple Access). If two or more stations are transmitting at the same time, a collision on the media is detected. All stations stop transmitting and back off for a random time. After the back-off time, the station checks the activity on the media again. If the media is idle, it starts transmitting. All other stations wait for the current transmission to end.

In full-duplex mode, the Carrier Sense and the Collision Detect signals are ignored. The MAC Control Module automatically handles reception and transmission of PAUSE control frames to achieve flow control.

Figure 12-1. Ethernet MAC Block Diagram



12.1 Host Interface

The host interface is connected to Lightfoot and the frame buffer module via two independent data buses. The slave interface is used by Lightfoot to write to and read from the registers and buffer descriptors. The master interface is used by the Ethernet MAC's internal DMA engine to fetch data from memory for transmission and write received data out to memory.

12.2 TX Ethernet MAC

The TX Ethernet MAC generates 10Base-T/100Base-TX transmit MII nibble data streams in response to the byte streams the transmit logic (host) supplies. It performs the required deferral and back-off algorithms, takes care of the inter-packet gap (IPG), computes the checksum (FCS), and monitors the physical media (by monitoring Carrier Sense and collision signals). The TX Ethernet MAC is divided into several modules that provide the following functionality:

- Generation of the control signals for the PHY during the transmission process
- Generation of the status signals the host uses to track the transmission process
- Random time generation used in the back-off process after a collision has been detected
- CRC generation and checking
- Pad generation
- Data nibble generation

12.3 RX Ethernet MAC

The RX Ethernet MAC transmits the data streams to the host in response to the 10Base-T or 100Base-TX received MII nibbles. It searches for the SFD (start frame delimiter) at the beginning of the packet, verifies the FCS, and detects any dribble nibbles or receive code violations.

The module is divided into several sub-modules providing the following functionality:

- Pre-amble removal
- Data assembly (from input nibble to output byte)
- CRC checking for all incoming packets
- Generation of the signal that can be used for address recognition (in the hash table)
- Generation of the status signals the host uses to track the reception process

12.4 MAC Control Module

The MAC Control Module performs the real-time flow control function for full-duplex operation. If the receive buffers start filling up when the user cannot continue dealing with the incoming packets, the user can send a PAUSE control frame to the transmitting station before an overflow occurs. This control frame inhibits the transmission of the data frames for a specified period of time. See "Control Frame Generation" on page 12-3 for details.

When the MAC Control Module receives a PAUSE control frame, it loads the pause timer with the value sent in the pause timer value field. The TX MAC is stopped from transmitting data frames for the a number of slot times equal to the value of the pause timer field.

The MAC Control Module has the following functionality:

- Control frame detection
- Control frame generation
- TX/RX MAC interface
- PAUSE timer
- Slot timer

12.4.1 Control Frame Detection

Incoming data packets are passed from the receiver via the MAC Control Module to the upper layers while the control frames are usually dropped. The RXFLOW bit in the ECTLMR register defines whether the pause control frame causes the transmitter to break the transmission or not. If RXFLOW bit is set then the RXC interrupt is set in the EISOURCE register. The PASSALL bit in the ECTLMR register defines whether the control frames are stored to the memory or not (regardless to the RXFLOW bit). If the PASSALL bit is set then the control frame is stored to the memory and its related buffer descriptor has the control frame bit (CF) set to "1". RXB interrupt in the EISOURCE register is set to "1".

Note: If both RXFLOW and PASSALL bits are set to "1", then only RXC interrupt is set in the EISOURCE register when a control frame is received.

The destination address must be a reserved multicast address (01-80-c2-00-00-01) or a destination address equal to the Ethernet IP Core MAC address. The Length/Type field must be equal to 8808 and the opcode to 0001 for a PAUSE control frame. When the receive flow control and the MAC Control Module are enabled (RXFLOW asserted and PASSALL deasserted), a PAUSE Timer Value from the PAUSE control frame is passed to the PAUSE timer.

12.4.2 Control Frame Generation

When the host wants to send a PAUSE control frame the Transmit Pause Request bit in the Transmit Flow Control Register (TXPAUSERQ bit in ETXCTRL register) is asserted. When a request is detected, the control module waits for the current transmission to end, and then starts transmitting the PAUSE control frame. The TX MAC pads and appends the FCS as normal. When the control frame has been sent, control of the TX MAC is returned to the host interface.

Asserting the TXFLOW bit in the MODER register enables the transmission of the PAUSE control frame. The Transmit Pause Timer Value TXPAUSETV[15:0] is set prior to the transmit pause request. The TPAUSETV contains the value to be sent as a Pause Timer Value in the pause control frame. The incoming data packets are passed from the receiver via the MAC Control Module to the upper layers while the control frames are usually dropped.

12.4.3 TX/RX MAC Interface

The TX/RX MAC interface controls the interaction between the TX and RX MACs and the host interface. This module takes over control of the TX MAC to initiate a transmission and feed it the appropriate data when a control frame needs to be sent. When normal data frames are being sent the relevant signals are switched to the control of the host interface. When a PAUSE control frame is received the TX/RX MAC interface determines whether to drop the frame or pass it through to the host interface, depending on the state of the PASSALL bit in the ECTLMR register.

12.4.4 PAUSE Timer

The 16-bit PAUSE timer is loaded with a pause timer value when a PAUSE control frame is received. The timer inhibits the data frame transmissions for the timer value time slots. This is done by:

- Preventing the TX MAC module from seeing the "start frame" signal from the host
- Preventing the host from seeing the signal "used data" signal from the TX MAC

The timer decrements by one each time a time slot passes by.

12.4.5 Slot Timer

The Slot Timer is activated when a PAUSE Timer is preloaded. It counts slot times and generates pulses to the PAUSE Timer for every slot time passed. One time slot is equal to the time taken to transmit 64 bytes.

12.5 Buffer Descriptors (BD)

The transmission and the reception processes are based on buffer descriptors.

Each buffer descriptor is 64 bits long. The least-significant 32 bits are reserved for length and status while the most-significant 32 bits contain the pointer to the associated buffer in memory (where data is stored). The Ethernet MAC core has an internal RAM that can store up to 128 buffer descriptors (for both RX and TX).

The buffer descriptor RAM resides at addresses $FF0400 to $FF07FF in the Lightfoot data bus memory map. The transmit descriptors are located between the start address ($FF0400) and the address given by the value written in the TX_BD_NUM register multiplied by eight plus the start address. The receive descriptors are located between the end of the TX buffer descriptors and the end word address ($FF07FC). In the following two subsections the term OFFSET is used to indicate the buffer descriptor base address.

The transmit and receive status of the packet is written to the associated buffer descriptor once its transmission/reception is finished.

12.5.1 TX Buffer Descriptors

The transmit descriptors contain information about associated buffers (length, status). They also contain pointers to the buffers holding the relevant data.

Table 12-1. TX Buffer Descriptor Control/Status – Bits 31-16

Address	Bits 31:16
OFFSET	TXLENGTH
R/W	R/W

Table 12-2. TX Buffer Descriptor Control/Status – Bits 15-8

Address	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bits 10:9	Bit 8
OFFSET	RDY	IRQ	WR	PAD	CRC	RSVD	UR
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 12-3. TX Buffer Descriptor Control/Status – Bits 7-0

Address	Bits 7:4	Bit 3	Bit 2	Bit 1	Bit 0
OFFSET	RETRY	RL	LC	DF	CS
R/W	R/W	R/W	R/W	R/W	R/W

12.5.1.1 Control Bits (set by software before transmission occurs)

Bits [10:9] - Reserved

These bits have no function. Writes are ignored and reads return zero.

TXLENGTH - Frame Length

Number of bytes to be transmitted in the frame.

RD - TX BD Ready

0 = The buffer associated with this buffer descriptor is not ready to be transmitted.

1 = The buffer associated with this descriptor is ready to be transmitted or is currently being transmitted. After the frame has been transmitted this bit is reset to zero.

IRQ - Interrupt Request

0 = No interrupt is generated after the transmission.

1 = When the frame has been transmitted, either the TXE or TXB interrupt will be asserted (see EISOURCE register for details).

WR - Wrap

0 = After transmitting this frame, increment the BD address and move onto the next buffer descriptor.

1 = This BD is the last in the buffer descriptor table, after this BD is used, move back to the first TX buffer descriptor.

PAD - Pad Enable

0 = No pads will be added at the end of short packets.

1 = Short packets will be padded out until they are MINFL bytes long.

CRC - CRC Enable

0 = CRC Checksum won't be added at the end of the packet.

1 = CRC Checksum will be added at the end of the packet.

12.5.1.2 Error Conditions (updated by the MAC after transmission is complete)

UR - Underrun

Underrun occurred when transmitting this buffer.

RETRY - Retry Count

This field indicates the number of retries required before the frame was successfully sent.

RL - Retransmission Limit

This bit is set when the transmitter fails to transmit a frame due to collisions on the medium after MAXRET + 1 retry attempts (see ECOLLCFG register for details).

LC - Late Collision

This bit is set when a late collision occurred while sending this buffer. The transmission is aborted immediately when this happens. Late Collision is defined in the ECOLLCFG register definition.

DF - Defer Indication

The frame was deferred before being sent successfully. This occurs when the transmitter had to wait for Carrier Sense before sending because the line was busy. This is NOT a collision indication and does not indicate a failure to transmit.

CS - Carrier Sense Lost

Carrier Sense Lost during a frame transmission. This bit is set after the MAC has finished sending the frame.

Table 12-4. TX Buffer Descriptor Pointer

Address	Bits 31:24	Bits 23:0
OFFSET + 4	-	TXPOINTER
R/W	R/W	R/W

Bits [31:24] - Reserved

These bits have no function. Writes are ignored and reads return zero.

TXPOINTER - Transmit Pointer

This is the base address in data memory where the data for this frame is stored

12.5.2 RX Buffer Descriptors

The receive BDs contain information about the received frames (length, status). When the internal DMA is selected, they also contain pointers to the buffers holding the relevant data.

Table 12-5. RX Buffer Descriptor Control/Status – Bits 31-16

Address	Bits 31:16
OFFSET	RXLENGTH
R/W	R/W

Table 12-6. RX Buffer Descriptor Control/Status – Bits 15-8

Address	Bit 15	Bit 14	Bit 13	Bit 12:8
OFFSET	E	IRQ	WR	RSVD
R/W	R/W	R/W	R/W	R/W

Table 12-7. RX Buffer Descriptor Control/Status – Bits 7-0

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
OFFSET	M	OR	IS	DN	TL	SF	CRC	LC
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

12.5.2.1 Control Bits (set by software before reception starts)

Bits [12:8] - Reserved

These bits have no function. Writes are ignored and reads return zero.

E - Empty

0 = The buffer associated with this BD has been filled with data or has stopped due to an error. While this bit is 0 the buffer descriptor won't be used.

1 = The buffer is empty and ready for receiving data or is currently receiving data.

IRQ - Interrupt Request

0 = No interrupt is generated after frame reception.

1 = When a frame has been received either the RXB or RXE interrupt will be asserted (see EISOURCE register for details).

WR - Wrap

0 = After filling this buffer with a received frame the MAC will increment the address and sample the next BD.

1 = This is the last buffer descriptor in the table, after this buffer has been filled the MAC will sample the first RX BD again.

12.5.2.2 Status Bits (updated by the MAC after reception has finished)

RXLENGTH - Received Frame Length

Number of bytes in the received frame.

M - Miss

0 = The frame is received because of an address recognition hit.

1 = The frame is received because the MAC is running in promiscuous mode.

OR - Overrun

This bit is set when a receiver overrun occurs during frame reception.

IS - Invalid Symbol

This bit is set when the reception of an invalid symbol is detected by the PHY.

DN - Dribble Nibble

This bit is set when a received frame is made up of a non-integer number of bytes (an extra nibble has been received).

TL - Too Long

This bit is set when a received frame is too long (bigger than the MAXFL value – see "Packet Length Register" on page 12-15 for details).

SF - Short Frame

This bit is set when a received frame is too short (smaller than the MINFL value - see "Packet Length Register" on page 12-15 for details).

CRC - RX CRC Error

This bit is set when the received frame contains a CRC error.

LC - Late Collision

This bit is set when a Late Collision occurred whilst receiving a frame.

Table 12-8. RX Buffer Descriptor Pointer

Address	Bits 31:24	Bits 23:0
OFFSET + 4	-	RXPOINTER
R/W	R/W	R/W

Bits [31:24] - Reserved

These bits have no function. Writes are ignored and reads return zero.

RXPOINTER - Received Frame Pointer

This is the address to which the MAC will start writing received data.

12.6 Operation

12.6.1 Frame Transmission

To transmit a frame the software must perform the following actions:

- Store the frame to the memory.
- Associate a buffer descriptor with the frame (via the pointer) and set the appropriate flags regarding CRC operation, interrupt generation etc. See "TX Buffer Descriptors" on page 12-4 for more information.
- Enable the Transmitter of the Ethernet MAC by asserting the TXEN bit in the mode register (EMODER).

When the transmitter is enabled, it samples the first buffer descriptor (BD). If the descriptor is marked as ready, the core reads the pointer to the memory storing the associated data and starts reading data to the internal FIFO. When the FIFO is filled, transmission begins.

At the end of the transmission, the transmit status is written to the buffer descriptor and, if specified in the buffer descriptor, an interrupt will be generated (assuming the relevant Interrupt is enabled and unmasked).

The transmitter will then sample the next BD according to the following rule:

- If the WR bit has not been set, the BD address is incremented and the next descriptor is sampled. When the READY bit for this BD is set, frame transmission will start.
- If the WR bit has been set, the first BD address is loaded again. When the BD is marked as ready, transmission will start.

12.6.2 Frame Reception

To receive a frame the software must perform the following actions:

- Set the first Receive Buffer Descriptor to be associated with the received packet and mark it as empty.
- Enable the Receiver of the Ethernet MAC by asserting the RXEN bit in the mode register (EMODER).

When the receiver is enabled it samples the first receiver BD. If it is marked as empty the MAC will receive any incoming frames nibble by nibble and process them according to the settings in the BD. The received data will be stored in the RX FIFO and written out to external memory on a word-by-word basis.

At the end of the frame the frame status bits and the length of the BD are updated. If specified in the buffer descriptor an interrupt will be generated (assuming the relevant Interrupt is enabled and unmasked).

The receiver will then sample the next RX BD according to the following rule:

- If the WR bit is not set the RX BD address is incremented and the next descriptor is sampled. If the READY bit in the new BD is set then the MAC can receive further frames.
- If the WR bit is set then the first RX BD is sampled again. If the READY bit in the new BD is set then the MAC can receive further frames.

The BD address is incremented and the next BD loaded. If the new BD is marked as empty, another frame can be received; otherwise the operation stops and any further frames will be lost.

12.7 Ethernet Register Set

12.7.1 Mode Register

Table 12-9. Ethernet Mode Register (EMODER) – Bits 31-16

Address	Bits 31:17	Bit 16	Register Name
$FF0000	RSVD	RECSMALL	EMODER
R/W	R/W	R/W	
Reset	0	0	

Table 12-10. Ethernet Mode Register (EMODER) – Bits 15-8

Address	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Register Name
$FF0000	PAD	HUGEN	CRCEN	DLYCRCEN	RST	FULLD	EXDFREN	NOBCKOF	EMODER
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	1	0	1	0	1	0	0	0	

Table 12-11. Ethernet Mode Register (EMODER) – Bits 7-0

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register Name
$FF0000	LOOPBCK	IFG	PRO	IAM	BRO	NOPRE	TXEN	RXEN	EMODER
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	0	0	0	

Bits [31:17] - Reserved

These bits have no function. Writes are ignored and reads return zero.

RECSMALL - Receive Small Packets

0 = Packets smaller than MINFL are ignored (see "Packet Length Register" on page 12-15 for details of MINFL)

1 = Packets smaller than MINFL are accepted

PAD - Padding Enabled

0 = Do not add padding to small frames

1 = Add pads to small frames (until frame size is equal to MINFL)

HUGEN - Huge Packets Enabled

0 = The maximum frame length is MAXFL. Bytes after that are discarded

1 = Frames up to 64kB are transmitted

CRCEN - CRC Enable

0 = The TX MAC will not append the CRC

1 = The TX MAC will append the CRC to every frame

DLYCRCEN - Delayed CRC Enable

0 = Normal operation (CRC calculation starts immediately after the SFD)

1 = CRC calculation starts four bytes after SFD

RST - Reset MAC

0 = Normal operation

1 = MAC is reset

FULLD - Full Duplex

0 = Half-duplex mode

1 = Full-duplex mode

EXDFREN - Excess Defer Enabled

0 = When the excessive deferral limit is reached a packet is aborted

1 = MAC waits for the carrier indefinitely

NOBCKOF - No Back-Off

0 = Normal operation (a binary exponential back-off algorithm is used)

1 = The TX MAC starts re-transmitting immediately after the collision

LOOPBCK - Loop Back

0 = Normal operation

1 = TX is looped back to the RX

IFG - Inter-frame Gap for Incoming Frames

0 = Normal operation (minimum IFG is required for a frame to be accepted)

1 = All frames are accepted regardless to the IFG

PRO - Promiscuous Mode

0 = Check the destination address of the incoming frames

1 = Receive frames regardless of the address

IAM - Individual Address Mode

0 = Normal operation (physical address is checked when the frame is received)

1 = The individual hash table is used to check all individual addresses that are received

BRO - Broadcast Address

0 = Receive all frames containing the broadcast address

1 = Reject all frames containing the broadcast address unless the PRO bit is set to "1"

NOPRE - No Pre-amble

0 = Normal operation (7-byte pre-amble is added)

1 = No pre-amble is sent

TXEN - Transmit Enable

0 = Transmitter is disabled

1 = Transmitter is enabled

RXEN - Receive Enable

0 = Receiver is disabled

1 = Receiver is enabled

12.7.2 Interrupt Source Register

Table 12-12. Ethernet Interrupt Source Register (EISOURCE)

Address	Bits 31:7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register Name
$0XFF0004	RSVD	RXC	TXC	BUSY	RXE	RXB	TXE	TXB	EISOURCE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	0	0	0	

Bits [31:7] - Reserved

These bits have no function. Writes are ignored and reads return zero.

RXC - Receive Control Frame

This bit indicates that a control frame was received.

TXC - Transmit Control Frame

This bit indicates that a control frame was transmitted.

BUSY - Busy

This bit indicates that a buffer was received ad discarded due to a lack of receive buffers.

RXE - Receive Error

This bit indicates that an error occurred while receiving data.

RXB - Receive Frame

This bit indicates that a frame was successfully received

TXE - Transmit Error

This bit indicates that a buffer was not transmitted due to a transmit error.

TXB - Transmit Buffer

This bit indicates that a buffer was successfully transmitted.

These bits are cleared by writing a "1" to the bit in question, multiple bits can be cleared with a single write.

12.7.3 Interrupt Mask Register

Table 12-13. Ethernet Interrupt Mask Register (EIMASK)

Address	Bits 31:7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register Name
$FF0008	RSVD	RXCM	TXCM	BUSYM	RXEM	RXBM	TXEM	TXBM	EIMASK
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	0	0	0	

Bits [31:7] - Reserved

These bits have no function, writes are ignored and reads return zero.

RXCM - Receive Control Frame Mask

0 = Event masked

1 = Event causes an interrupt

TXCM - Transmit Control Frame Mask

0 = Event masked

1 = Event causes an interrupt

BUSY_M - Busy Mask

0 = Event masked

1 = Event causes an interrupt

RXEM - Receive Error Mask

0 = Event masked

1 = Event causes an interrupt

RXBM - Receive Frame Mask

0 = Event masked

1 = Event causes an interrupt

TXEM - Transmit Error Mask

0 = Event masked

1 = Event causes an interrupt

TXBM - Transmit Buffer Mask

0 = Event masked

1 = Event causes an interrupt

12.7.4 Back-to-Back Inter-Packet Gap Register

Table 12-14. Ethernet Back-to-Back Inter-Packet Gap Register (EIPGT)

Address	Bits 31:7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register Name
$FF000C	RSVD	IPGT6	IPGT5	IPGT4	IPGT3	IPGT2	IPGT1	IPGT0	EIPGT
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	1	0	0	1	0	

Bits [31:7] - Reserved

These bits have no function, writes are ignored and reads return zero.

IPGT - Back-to-Back Inter-Packet Gap

Full-duplex: The recommended value is $15, which is equal to 0.96 ms (100 Mbps) or 9.6 ms (10 Mbps). The value in this register should equal the required period in nibble-times minus six.

Half-duplex: The recommended value (and the default for this register) is $12, which is equal to 0.96 ms (100 Mbps) or 9.6 ms (10 Mbps). The value in this register should be equal to the required period in nibble-times minus three.

12.7.5 Non Back-to-Back Inter-Packet Gap Register 1

Table 12-15. Ethernet Non Back-to-Back Inter-Packet Gap Register 1 (EIPGR1)

Address	Bits 31:7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register Name
$FF0010	RSVD	IPGR16	IPGR15	IPGR14	IPGR13	IPGR12	IPGR11	IPGR10	EIPGR1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	0	0	0	

IPGR1 - Non Back-to-Back Inter-Packet Gap 1

When the carrier sense appears within the IPGR1 window, the TX MAC defers and the IPGR counter is reset.

When the carrier sense appears later than the IPGR1 window, the IPGR counter continues counting. The recommended and default value for this register is 0xC and must be within the range 0 - IPGR2.

12.7.6 Non Back-to-Back Inter-Packet Gap Register 2

Table 12-16. Ethernet Non Back-to-Back Inter-Packet Gagp Register 2 (EIPGR2)

Address	Bits 31:7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register Name
$FF0014	RSVD	IPGR26	IPGR25	IPGR24	IPGR23	IPGR22	IPGR21	IPGR20	EIPGR2
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	0	0	0	

Bits [31:7] - Reserved

These bits have no function. Writes are ignored and reads return zero.

IPGR2 - Non Back-to-Back Inter-Packet Gap 2

The recommended and default value for this is $12 which equals 0.96 ms (100 Mbps) or 9.6 ms (10 Mbps).

12.7.7 Packet Length Register

Table 12-17. Ethernet Packet Length Register (EPKLEN)

Address	Bits 31:16	Bits 15:0	Register Name
$FF0018	MINFL	MAXFL	EPKLEN
R/W	R/W	R/W	
Reset	$003C	$0600	

MINFL - Minimum Frame Length

The minimum Ethernet packet is 60 bytes long. If reception of smaller frames is required, then the RECSMALL bit in the MODER register can be asserted, or the value of this field can be changed. To transmit small packets, the PAD bit in the MODER register can be asserted or the MINFL field can be changed.

MAXFL - Maximum Frame Length

The maximum Ethernet packet is 1518 bytes long. To support this and to leave additional space for the tags, a default maximum packet length equal to 1536 bytes is set. If larger packets need to be supported then the HUGEN bit in the MODER register can be asserted or the value of MAXFL can be changed.

12.7.8 Collision and Retry Configuration Register

Table 12-18. Ethernet Collision and Retry Configuration Register (ECOLLCFG)

Address	Bits 31:20	Bits 19:16	Bits 15:6	Bits 5:0	Register Name
0xFF001C	RSVD	MAXRET	RSVD	COLLVALID	ECOLLCFG
R/W	R/W	R/W	R/W	R/W	
Reset	0x000	0xF	0x000	0x03	

Bits [31:20] - Reserved

These bits have no function. Writes are ignored and reads return zero.

MAXRET - Maximum Retry Attempts

This field specifies the maximum number of sequential re-transmission attempts after a collision is detected. When the maximum number is reached the TX MAC reports an error and stops transmitting the current packet. In accordance with the IEEE 802.3 standard, the default value is set to $F.

COLLVALID - Collision Valid Window

This field specifies the collision time window. Collisions that occur later than the time windows are reported as “Late Collision” and transmission of the current packet is aborted.

12.7.9 Transmit BD Number Register

Table 12-19. Ethernet Transmit BD Number Register (ETXBDNUM)

Address	Bits 31:10	Bits 9:0	Register Name
$FF0020	RSVD	TXBDNUM	ETXBDNUM
R/W	R/W	R/W	
Reset	0x00000	0x080	

Bits [31:10] - Reserved

These bits have no function. Writes are ignored and reads return zero.

TXBDNUM - Number of Transmit Buffer Descriptors

This register holds the number of Buffer Descriptors in the BD RAM which are assigned for frame transmission.

12.7.10 Control Module Mode Register

Table 12-20. Ethernet Control Module Mode Register (ECTLMR)

Address	Bits 31:3	Bit 2	Bit 1	Bit 0	Register Name
$FF0024	RSVD	TXFLOW	RXFLOW	PASSALL	ECTLMR
R/W	R/W	R/W	R/W	R/W	
Reset	0x0000000	0	0	0	

Bits [31:3] - Reserved

These bits have no function. Writes are ignored and reads return zero.

TXFLOW - Transmit Flow Control

0 = PAUSE control frames are blocked

1 = PAUSE control frames can be sent (ETXCTRL register is used to initiate transmission of pause control frames)

RXFLOW - Receive Flow Control

0 = Received PAUSE control frames are ignored

1 = Transmit function is blocked when a PAUSE control frame is received

PASSALL - Pass All Received Frames

0 = Control frames are not passed to the host interface. MAC Control Module is enabled

1 = All received frames are passed to the host interface. MAC Control Module is disabled. This allows flow control to be performed in software.

12.7.11 MAC Address Registers

Table 12-21. Ethernet MAC Address Register 0 (EMADR0)

Address	Bits 31:24	Bits 23:16	Bits 15:8	Bits 7:0	Register Name
$FF0040	MADR2	MADR3	MADR4	MADR5	EMADR0
R/W	R/W	R/W	R/W	R/W	
Reset	0x00	0x00	0x00	0x00	

Table 12-22. Ethernet MAC Address Register 1 (EMADR1)

Address	Bits 31:16	Bits 15:8	Bits 7:0	Register Name
$FF0044	RSVD	MADR0	MADR1	EMADR1
R/W	R/W	R/W	R/W	
Reset	0x0000	0x00	0x00	

EMADR1, bits [31:16] - Reserved

These bits have no function. Writes are ignored and reads return zero.

MADRx - Byte x of the MAC Address

12.7.12 Hash Registers

Table 12-23. Ethernet Hash Register 0 (EHASH0)

Address	Bits 31:0	Register Name
$FF0048	HASH0	EHASH0
R/W	R/W	
Reset	0x00000000	

Table 12-24. Ethernet Hash Register 1 (EHASH1)

Address	Bits 31:0	Register Name
$FF004C	HASH1	EHASH1
R/W	R/W	
Reset	0x00000000	

HASH0 - Hash 0 Value

HASH1 - Hash 1 Value

12.7.13 Transmit Flow Control Register

Table 12-25. Ethernet Transmit Flow Control Register (ETXCTRL)

Address	Bits 31:17	Bit 16	Bits 15:0	Register Name
$FF0050	RSVD	TXPAUSERQ	TXPAUSETV	ETXCTRL
R/W	R/W	R/W	R/W	
Reset	0x00000	0	0x0000	

Bits [31:17] - Reserved

These bits have no function. Writes are ignored and reads return zero.

TXPAUSERQ - Pause Control Frame Transmit Request

Writing a "1" to this bit causes the MAC Control Module to transmit a pause control frame with the pause timer value given in TXPAUSETV. This bit is automatically cleared to zero.

Note: For a control frame to be transmitted Transmit Flow Control must be enabled (TXFLOW must be asserted in the ECTLMR register).

TXPAUSETV - Pause Control frame Timer Value

This is the 16-bit pause timer value that is sent as part of the pause control frame and determines the number of slot times the transmitting station will wait for before recommencing transmission.

Chapter 13. Ethernet Frame Buffers

The VS2000 includes two blocks of dual-port RAM for storing Ethernet TX and RX frames. The TX frame buffer is 1KB and the RX frame buffer is 3KB. The RX frame buffer can be written to by the Ethernet MAC and read from by Lightfoot. The TX frame buffer can be written to by Lightfoot and read from by the Ethernet MAC.

When the Ethernet MAC DMA engine performs a read access (to fetch data for a frame transmission) the Ethernet memory controller examines the address. If the address is in the range $FF1000-$FF13FF, the access is made from the TX frame buffer. If the address is outside this range the access passes through the internal data bus and off-chip via the external memory controller.

When the Ethernet MAC DMA engine performs a write access (to write out data from a received frame) the Ethernet memory controller examines the address. If the address is in the range $FF1400-$FF1FFF, the access is made from the RX frame buffer. If the address is outside this range the access passes through the internal data bus and off-chip via the external memory controller.

Both dual port memories have a single read and single write port, therefore neither can be used as read/wrire memory. A write to a read port will have no effect, a read from a write port will return undefined data.

It is important to note that all off-chip Ethernet MAC-generated accesses are non-cached, this is irrespective of the address.

Chapter 14. External Memory Controller

The VS2000 has an external memory controller (EMC) that interfaces the external memory bus to the internal memory buses. This controller allows glueless connection of up to four external devices on the external bus. Each external device occupies its own configurable zone. Internally the Harvard architecture Lightfoot CPU uses separate program and memory data buses, the EMC also unifies these buses into the single external bus, thus reducing the number of external memory devices required.

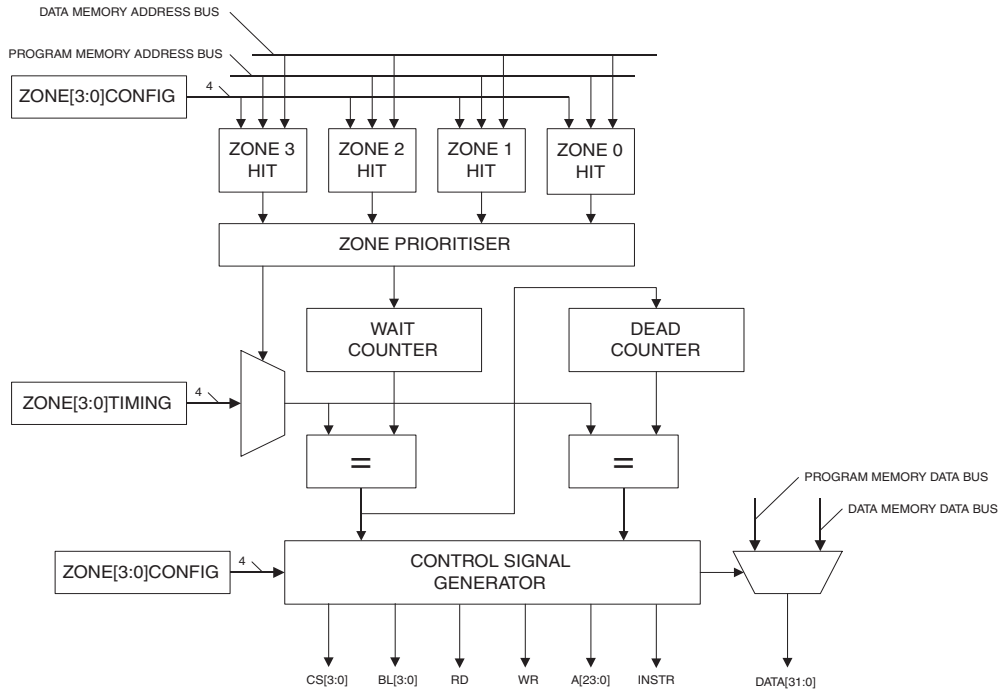
Internally, Lightfoot uses 24-bit addresses for both program and data memory, giving 16 MB of addressable space on each bus. The external memory bus also uses 24-bit addresses, with an additional control signal output that defines the type of access, i.e. program or data, thus permitting 32 MB of external memory space to be directly addressed.

14.1 External Memory Controller Features

The external memory controller has the following features:

- Supports glueless connection of up to four external devices.
- Programmable zone timing: wait and dead time.
- Programmable zone ownership: program, data, or unified.
- Address inversion permits single memory device to be used for program and data accesses.
- Programmable zone signal polarity: chip select and byte lane.

Figure 14-1. External Memory Controller Block Diagram



14.2 Operation

The purpose of the external memory controller is to allow up to four external memory resources to be directly connected to the VS2000 device using a simple interface. The interface consists of the signals described in Table 14-1.

Table 14-1. External Memory Interface

VS2000 Pin(s)	I/O	Description
A[23:0]	O	VS2000 byte-aligned address output
D[31:0]	I/O	VS2000 bi-directional data bus (in 8-bit mode only bits D[7:0] are used)
CS[3:0]	O	Chip select outputs, one for each programmable zone
BL[3:0]	O	Byte lane select outputs for 32-bit transfers
RD	O	Read strobe output (defaults to active low)
WR	O	Write strobe output (defaults to active low)

The set-up and operation of the external memory controller is described in the following subsections.

14.2.1 Programmable Chip Select Zone

The VS2000 has four programmable zones each with a corresponding chip select output (CS[3:0]).

14.2.1.1 Zone Address Ranges

The zone range is used to map a given range of logical addresses to one or more physical memory resources. The range is specified by the most-significant eight bits of the address (address bits A[23:16]).

For example an 64KB EPROM could be mapped to zone 0 at addresses $000000 to $00FFFF and a 2 MB SRAM mapped to zone 1 at addresses $200000 to $3FFFFF using the following set-up:

- Zone 0: upper boundary = $00, lower boundary = $00
- Zone 1: upper boundary = $3F, lower boundary = $20

Note that zone boundaries are inclusive. In the above example, setting the lower and upper boundaries to $00 gives zone 0 the 64 KB block from $000000 to $00FFFF.

14.2.1.2 Zone Timing

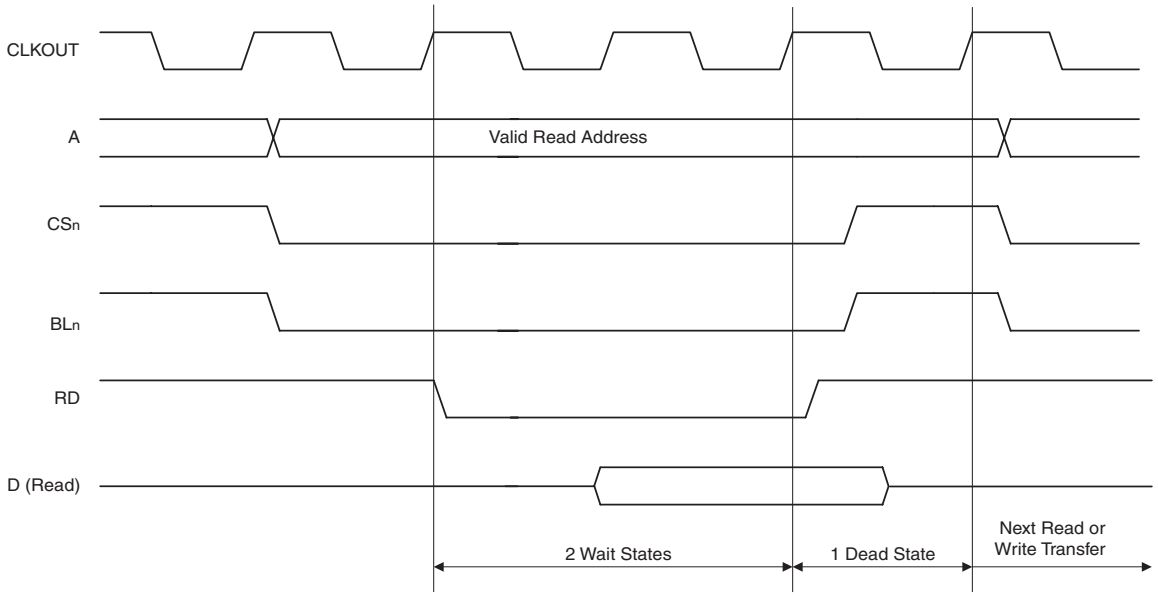
The timing of an access to external memory is controlled by two values: the wait time and the dead time. Figure 14-2 shows a typical read access.

The dead time is the time from when either of the read or write strobes and the chip selects are de-asserted to the end of the access. This is included because certain memory resources take a considerable amount of time to tri-state their data buses following a read access.

The reset state for all zones is as follows:

- Wait count = 8 cycles
- Dead count = 4 cycles

Figure 14-2. Timing Diagram of a Read with Multiple Wait and Dead States



Note: If the number of dead states is set to zero, the Lightfoot transfer will be completed in the final cycle of the wait count, thus ending the external memory transfer as the read or write strobe returns high.

The wait time is determined by the access time of the memory. The memory control read and write strobes, RD, WR, are asserted for a programmable number of wait state cycles.

14.2.1.3 Zone Ownership

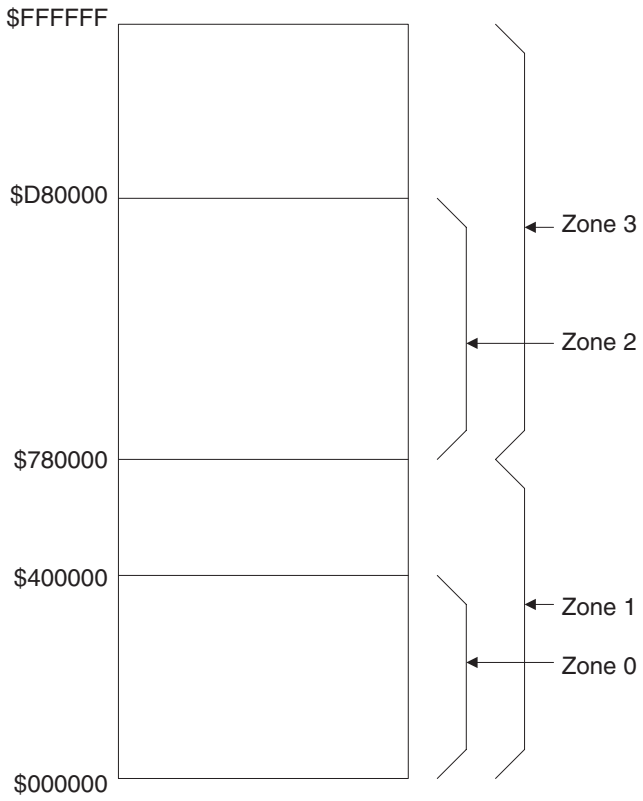
Zone ownership can be used to make certain memory resources only accessible to either the program bus or data bus. Memory zones can also be configures as unified, i.e. they will be accessible to both buses.

If the address ranges of two zones overlap, the lower zone has priority over accesses to the shared address range (unless the two zones have different ownerships).

14.2.1.4 Zone Configuration Example

To demonstrate the system of zone ownership, boundary, and priority, the following example is given.

Figure 14-3. Zone Configuration Example



The ownerships in Figure 14-3 are set as follows:

- Zone 0 = program
- Zone 1 = data
- Zone 2 = program
- Zone 3 = unified

In the case of a program access to an address in the range $000000 to $3FFFFF, zone 0 is accessed. In the case of a data access in the same address range, range zone 1 is accessed. This is because the ownerships are mutually exclusive.

In the case of a program access to an address in the range $780000 to $D7FFFF zone 2 will the accessed. This is because in the case of two overlapping zones, the lower zone takes priority.

In the case of a data access in the same range zone 3 will be accessed. This is because zone 2 is configured as a program only zone.

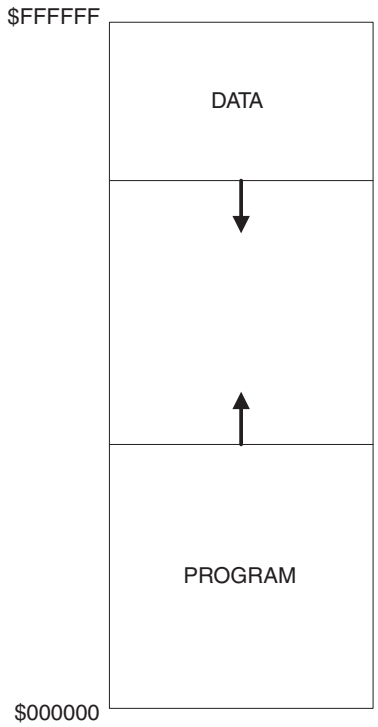
Note 1. If address inversion was switched on for program accesses in zone 3 in this case a program access to an address in the range $780000 to $D7FFFF would still map to zone 2 as the address comparison between incoming address and the zone boundaries is performed **before** address inversion.

2. If address inversion is to be used, memory blocks must be mapped onto appropriate boundaries. In the example above, the 2MB SRAM must be mapped onto a 2MB boundary in the address range to allow address inversion to be used.

14.2.1.5 Zone Address Inversion

Address inversion can be configured on any zone on either program or data bus accesses. This function is included to allow logically identical program and data memory memory addresses to share a unified memory resource without causing data corruption.

Figure 14-4. Zone Address Inversion



In this example one memory resource has been set up as a unified zone on zone 0 using the entire address range. Address inversion has been switched on for data accesses. In this system a program access at $000000 accesses physical memory address $000000 whereas a data access at address $000000 accesses physical memory address $FFFFFC (the two LSBs are not inverted).

This causes the program area to grow up from the bottom of memory and the data area to grow down from the top of the memory area, keeping them separate without having to change the addresses used in software.

14.2.2 Configurable External Data Bus Width

The VS2000 can be connected directly to either an 8- or 32-bit memory system. This is selected using the WIDTH input pin on the VS2000 device. When conected to V_{DD}, 8-bit mode is selected and all external memory transfers are performed using D[7:0]. When connected to V_{SS}, 32-bit mode is selected and all external memory transfers use a 32-bit external data bus, D[31:0].

The state of the WIDTH pin should be fixed at power-on and not modified during operation as changes to the state of this pin while power is applied may result in erroneous device behavior.

14.2.3 Byte Lane Signals

The VS2000 byte lane output signals (BL[3:0]) are used in 32-bit mode to select the active bytes during external memory read and write transfers.

During all 32-bit read transfers (of any size), all byte lane signals are asserted. Logic internal to the VS2000 is responsible for routing data to the Lightfoot core as required. During all write transfers, write data is broadcast across the external data bus and the byte lane outputs indicate which byte positions are being written to.

During all 32-bit write transfers, the byte lane signals are asserted according to the transfer size and address, as shown in Table 14-2.

Table 14-2. Byte Lane Signal Assertions During Write Transfers

Access Size	Address[1:0]	Byte Lane [3:0](1)
Byte	00	0111
Byte	01	1011
Byte	10	1101
Byte	11	1110
Half-word	00	0011
Half-word	10	1100
Word	00	0000

Note 1. The byte lane signals are asserted low by default.

When connecting the VS2000 to a 32-bit memory resource only the word address bits from the device (A[23:2]) are typically connected to the memory device's address bus. This configuration assumes that the memory resource has byte enable signals that may be used to control byte and half-word write transfers.

14.3 External Memory Controller Register Set

The external memory controller is configured via the Lightfoot register bus interface. Each zone has two configuration registers.

14.3.1 External Memory Controller Configuration Registers

Table 14-3. External Memory Controller Configuration Register 0 (Z0CFG) – Bits 23-16

Address	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16	Register Name
$E0	WRPOL	RDPOL	DINV0	PINV0	CSPOL0	BLPOL0	OWNR0[1]	OWNR0[0]	Z0CFG
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	0	1	0	

Table 14-4. External Memory Controller Configuration Register 0 (Z0CFG) – Bits 15-0

Address	Bits 15:8	Bits 7:0	Register Name
$E0	HIGHADDR0	LOWADDR0	Z0CFG
Read/Write	R/W	R/W	
Reset	0x7F	0x00	

Table 14-5. External Memory Controller Configuration Register 1 (Z1CFG) – Bits 23-16

Address	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16	Register Name
$E2			DATAINV1	PROGINV1	CSPOL1	BLPOL1	OWNR1[1]	OWNR1[0]	Z1CFG
Read/Write			R/W	R/W	R/W	R/W	R/W	R/W	
Reset			0	0	0	0	1	1	

Table 14-6. External Memory Controller Configuration Register 1 (Z1CFG) – Bits 15-0

Address	Bits 15:8	Bits 7:0	Register Name
$E2	HIGHADDR1	LOWADDR1	Z1CFG
Read/Write	R/W	R/W	
Reset	0x00	0x00	

Table 14-7. External Memory Controller Configuration Register 2 (Z2CFG) – Bits 23-16

Address	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16	Register Name
$E4			DATAINV2	PROGINV2	CSPOL2	BLPOL2	OWNR2[1]	OWNR2[0]	Z2CFG
Read/Write			R/W	R/W	R/W	R/W	R/W	R/W	
Reset			0	0	0	0	1	1	

Table 14-8. External Memory Controller Configuration Register 2 (Z2CFG) – Bits 15-0

Address	Bits 15:8	Bits 7:0	Register Name
$E4	HIGHADDR2	LOWADDR2	Z2CFG
Read/Write	R/W	R/W	
Reset	0x00	0x00	

Table 14-9. External Memory Controller Configuration Register 3 (Z3CFG) – Bits 23-16

Address	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16	Register Name
$E6			DATAINV3	PROGINV3	CSPOL3	BLPOL3	OWNR3[1]	OWNR3[0]	Z3CFG
Read/Write			R/W	R/W	R/W	R/W	R/W	R/W	
Reset			0	0	0	0	1	1	

Table 14-10. External Memory Controller Configuration Register 3 (Z3CFG) – Bits 15-0

Address	Bits 15:8	Bits 7:0	Register Name
$E6	HIGHADDR3	LOWADDR3	Z3CFG
Read/Write	R/W	R/W	
Reset	0x00	0x00	

WRPOL - Polarity of the Write Strobe Signal

0 = Active low (reset state)

1 = Active high

RDPOL - Polarity of the Read Strobe Signal

0 = Active low (reset state)

1= Active high

DINVx - Inverting the External Address Bus during Data Accesses for Zone x

0 = Data memory address bus normal (reset state)

1 = Data memory address bus inverted

PINVx - Inverting the External Address Bus during Program Accesses for Zone x

0 = Program memory address bus normal (reset state)

1 = Program memory address bus inverted

CSPOLx - Polarity of the Chip Select Signal for Zone x

0 = Active low (reset state)

1 = Active high

BLPOLx - Polarity of the Byte Lane Signals for Zone x

0 = Active low (reset state)

1 = Active high

OWNRx - Ownership of Zone x

00 = Data bus only

01 = Program bus only

10 = Unified (reset state)

11 = Zone disabled

HIGHADDRx - Upper Bound of the Top Eight Address Bits

This specifies the address range which maps zone x. The upper and lower bounds are inclusive.

LOWADDRx - Lower Bound of the Top Eight Address Bits

This specifies the address range which maps to zone x. The upper and lower bounds are inclusive.

14.3.2 External Memory Controller Timing Registers

Table 14-11. External Memory Controller Timing Register 0 (Z0TMG)

Address	Bits 31:16	Bits 15:0	Register Name
$E1	WAIT0	DEAD0	Z0TMG
Read/Write	R/W	R/W	
Reset	0x08	0X04	

Table 14-12. External Memory Controller Timing Register 1 (Z1TMG)

Address	Bits 31:16	Bits 15:0	Register Name
$E3	WAIT1	DEAD1	Z1TMG
Read/Write	R/W	R/W	
Reset	0x01	0X00	

Table 14-13. External Memory Controller Timing Register 2 (Z2TMG)

Address	Bits 31:16	Bits 15:0	Register Name
$E5	WAIT2	DEAD2	Z2TMG
Read/Write	R/W	R/W	
Reset	0x01	0x00	

Table 14-14. External Memory Controller Timing Register 3 (Z3TMG)

Address	Bits 31:16	Bits 15:0	Register Name
$E7	WAIT3	DEAD3	Z3TMG
Read/Write	R/W	R/W	
Reset	0x01	0X00	

WAITx - Number of Cycles the External Memory Controller will Spend in the Wait State for Zone X

The mininum number of cycles is one. Writing zero to this field will set the wait count to one.

DEADx - Number of Cycles the External Memory Controller will Spend in the Dead State for Zone X

See Chapter 4, "Clock System" for an explanation of the wait and dead time counts.

Chapter 15. Interrupt Controller

The VS2000 has a system Interrupt Controller Module (ICM) which provides control over all system interrupt sources. The ICM controls the interrupt and associated 4-bit interrupt identification vector that is pas sed to the Lightfoot CPU. The ICM is configurable, allowing the user to assign priority to interrupt sources, define interrupts as either level or edge sensitive, and allowing interrupts to be masked.

The VS2000 interrupt sources are both internally generated (by on-chip peripherals) and externally generated using configurable external interrupts (GPIO Port A[3:0], GPIO Port B[3:0] and the external IRQ pin). The device interrupts and associated default priorities are shown in Table 15-1. It should be noted that interrupt vectors $D to $F (inclusive) are not used in the VS2000.

Table 15-1. Lightfoot CPU Interrupt Vector Assignments

Interrupt Source	Lightfoot Interrupt Vector	Default Priority (1 = Highest)
Watchdog	$0	1
Triple Timer Counter A	$1	2
UART A	$2	3
Ethernet MAC	$3	4
UART B	$4	5
Triple Timer Counter B	$5	6
GPIO Port A	$6	7
GPIO Port B	$7	8
SPI 0	$8	9
SPI 1	$9	10
SPI 2	$A	11
External Interrupt Input	$B	12
RTC Interrupt	$C	13
unused	$D	14
unused	$E	15
unused	$F	16

15.1 Operation

15.1.1 Configuring the Interrupt Controller

After a reset, the interrupt controller is configured with all interrupts as level-sensitive active-high interrupts and the priority is such that interrupt source 0 is the highest priority and interrupt source 15 the lowest. All interrupts are masked after a reset.

To configure the interrupt controller, the priority and configuration registers must be loaded according to the instructions given in the register set section. In the VS2000, all internal interrupts are level-sensitive active-high interrupts so no configuration change is required. All set-up of the priorities and configuration should be performed BEFORE unmasking any interrupts to prevent spurious interrupts reaching Lightfoot.

15.2 Interrupt Controller Register Set

15.2.1 Interrupt Priority Registers

Table 15-2. Interrupt Priority Register 0 (IPL0) – Bits 31-16

Address	Bits 31:28	Bits 27:24	Bits 23:20	Bits 19:16	Register Name
$F0	PRIVEC8	PRIVEC9	PRIVEC10	PRIVEC11	IPL0
R/W	R/W	R/W	R/W	R/W	
Reset	0x7	0x6	0x5	0x4	

Table 15-3. Interrupt Priority Register 0 (IPL0) – Bits 15-0

Address	Bits 15:12	Bits 11:8	Bits 7:4	Bits 3:0	Register Name
$F0	PRIVEC12	PRIVEC13	PRIVEC14	PRIVEC15	IPL0
R/W	R/W	R/W	R/W	R/W	
Reset	0x3	0x2	0x1	0x0	

Table 15-4. Interrupt Priority Register 1 (IPL1) – Bits 31-16

Address	Bits 31:28	Bits 27:24	Bits 23:20	Bits 19:16	Register Name
$F1	PRIVEC0	PRIVEC1	PRIVEC2	PRIVEC3	IPL1
R/W	R/W	R/W	R/W	R/W	
Reset	0xF	0xE	0xD	0xC	

Table 15-5. Interrupt Priority Register 1 (IPL1) – Bits 15-0

Address	Bits 15:12	Bits 11:8	Bits 6:4	Bits 3:0	Register Name
$F1	PRIVEC4	PRIVEC5	PRIVEC6	PRIVEC7	IPL1
R/W	R/W	R/W	R/W	R/W	
Reset	0xB	0xA	0x9	0x8	

PRIVECx[3:0]

To program the priorities of the interrupt sources, the vector of the interrupt in question is written to the appropriate "priority slot". Therefore, PRIVECx is the Lightfoot interrupt vector associated with interrupt priority x.

15.2.2 Interrupt Mask Register

Table 15-6. Interrupt Mask Regsiter (IMASK) – Bits 15-8

Address	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Register Name
$F2	MSK15	MSK14	MSK13	MSK12	MSK11	MSK10	MSK9	MSK8	IMASK
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	0	0	0	

Table 15-7. Interrupt Mask Regsiter (IMASK) – Bits 7-0

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register Name
$F2	MSK7	MSK6	MSK5	MSK4	MSK3	MSK2	MSK1	MSK0	IMASK
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	0	0	0	

MSKx - Interrupt Mask

The IMASK register holds the sixteen interrupt mask bits. These bits individually mask all sixteen possible interrupt sources.

1 = Interrupt un-masked

0 = Interrupt masked

15.2.3 Interrupt Configuration Registers

Table 15-8. Interrupt Configuration Register 0 (ICFG0) – Bits 15-8

Address	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Register Name
$F3	EL15	EL14	EL13	EL12	EL11	EL10	EL9	EL8	ICFG0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	1	1	1	1	1	1	1	1	

Table 15-9. Interrupt Configuration Register 0 (ICFG0) – Bits 7-0

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register Name
$F3	EL7	EL6	EL5	EL4	EL3	EL2	EL1	EL0	ICFG0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	1	1	1	1	1	1	1	1	

ELx - Edge/Level Sensitivity

These bits determine the type of interrupt the ICM responds to. If an edge-sensitive interrupt is selected, the edge-detection hardware is enabled to catch edge-sensitive interrupts and convert them to level-sensitive.

0 = Edge sensitive

1 = Level sensitive

Table 15-10. Interrupt Configuration Register 1 (ICFG1) – Bits 15-8

Address	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Register Name
$F4	LH15	LH14	LH13	LH12	LH11	LH10	LH9	LH8	ICFG1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	1	1	1	1	1	1	1	1	

Table 15-11. Interrupt Configuration Register 1 (ICFG1) – Bits 7-0

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register Name
$F4	LH7	LH6	LH5	LH4	LH3	LH2	LH1	LH0	ICFG1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	1	1	1	1	1	1	1	1	

LHx - Active Level

These bits determine the active edge/level of each interrupt source. When an interrupt source is configured as edge-sensitive, this bit determines whether the controller responds to rising or falling edge interrupts. When an interrupt source is configured as level-sensitive, this bit determines whether the controller responds to active high or active low interrupts.

0 = Falling edge/active-low

1 = Rising edge/active-high

15.2.4 Interrupt Clear Register

Table 15-12. Interrupt Clear Register (ICLR) – Bits 15-8

Address	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Register Name
$F5	CLR15	CLR14	CLR13	CLR12	CLR11	CLR10	CLR9	CLR8	ICLR
R/W	W	W	W	W	W	W	W	W	
Reset	0	0	0	0	0	0	0	0	

Table 15-13. Interrupt Clear Register (ICLR) – Bits 7-0

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register Name
$F5	CLR7	CLR6	CLR5	CLR4	CLR3	CLR2	CLR1	CLR0	ICLR
R/W	W	W	W	W	W	W	W	W	
Reset	0	0	0	0	0	0	0	0	

CLRx - Interrupt Clear

These bits are used to clear latched edge-sensitive interrupts. This is a write-only register.

0 = Leave interrupt state unmodified

1 = Clear interrupt

15.2.5 Interrupt Flag Register

Table 15-14. Interrupt Flag Register (IFLAG) – Bits 15-8

Address	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Register Name
$F6	FLG15	FLG14	FLG13	FLG12	FLG11	FLG10	FLG9	FLG8	IFLAG
R/W	W	W	W	W	W	W	W	W	
Reset	0	0	0	0	0	0	0	0	

Table 15-15. Interrupt Flag Register (IFLAG) – Bits 7-0

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register Name
0xF6$	FLG7	FLG6	FLG5	FLG4	FLG3	FLG2	FLG1	FLG0	IFLAG
R/W	W	W	W	W	W	W	W	W	
Reset	0	0	0	0	0	0	0	0	

FLGx - Interrupt Flag

The Interrupt Flag register allows the status of the interrupts to be read. Note that an active interrupt is shown as a "1" irrespective of edge/level sensitivity or active edge/level. These values indicate the status of the interrupts **before** masking, changing the contents of the mask register will have no affect on this register.

0 = Interrupt inactive

1 = Interrupt active

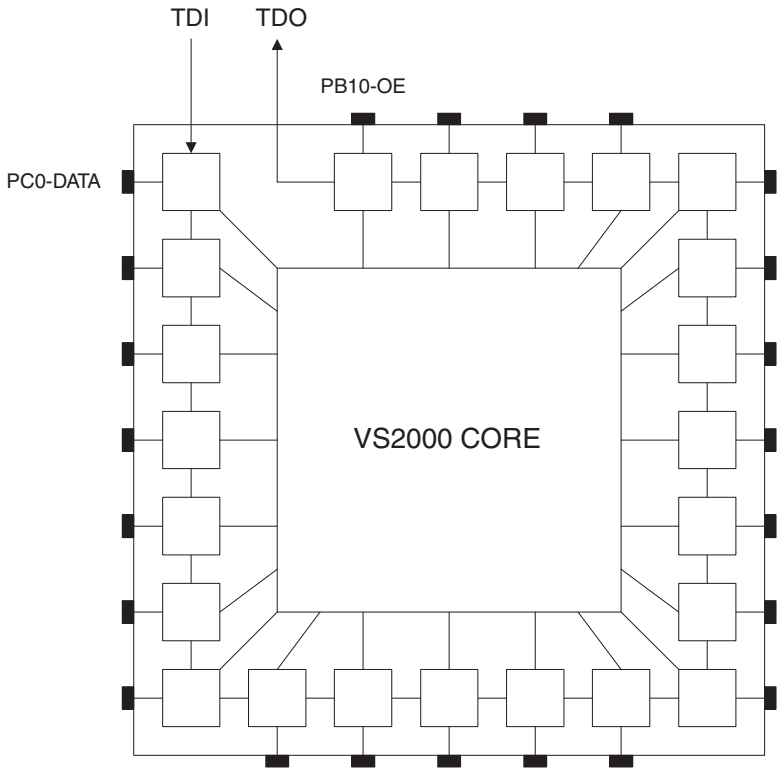
Chapter 16. Boundary Scan Interface

16.1 Boundary Scan Structure

16.1.1 VS2000 Boundary Scan Chain

The VS2000 has a Boundary Scan interface which links all device I/O pins in a serial chain. The Boundary Scan register chain contains 255 boundary scan cells which are inserted between the VS2000 I/O pads and the device core. In the case of bi-directional pads, such as the data bus, there are two boundary scan cells per pad: one bi-directional cell which is dedicated to the data in/out signals and one which is dedicated to pin direction control.

Figure 16-1. Architecture of the Boundary Scan Chain



The Boundary Scan Chain is formed by connecting the scan cells in a series chain. The input to the chain is connected to the TDI pad and the output from the chain can be connected through a multiplexer to the TDO pad. Three other external pins are involved in the operation of the boundary scan circuit: TCK (the boundary scan clock signal), TRST_L (the asynchronous boundary scan reset signal), and TMS (the test mode select signal). All boundary scan pins are described in Table 16-1.

16.1.2 Boundary Scan Signals

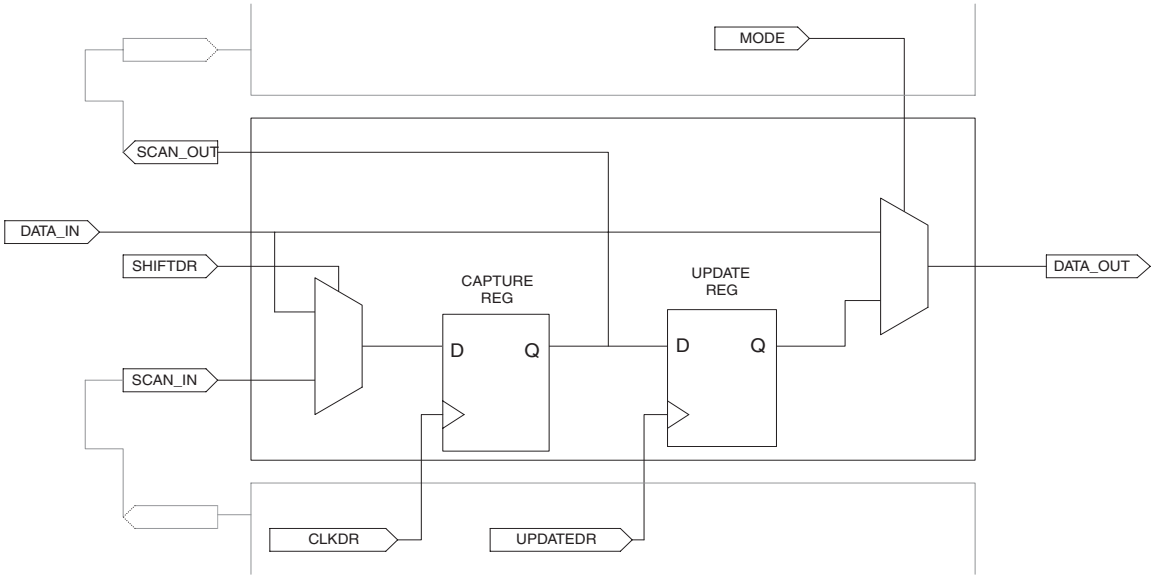
Table 16-1. Boundary Scan Signals

VS2000 Pin	Function Name	Description
TDI	Test Data Input	Serial input for test data as well as test instructions. Data is shifted in on the rising edge of TCK.
TDO	Test Data Output	Serial output for instructions as well as test and programming data. Data is shifted out on the falling edge of TCK. This pin is tri-state if data is not being shifted out.
TMS	Test Mode Select	Input pin which determines the state transitions of the TAP controller state machine. State machine transitions occur on the rising edge of TCK so TMS must be set up BEFORE the rising edge of TCK.
TCK	Test Clock Input	The clock input to the boundary scan logic. Both the rising and falling edges are used in the logic.
TRST	Test Reset Input	Active-low asynchronous reset input used to reset the boundary scan logic

16.1.3 Boundary Scan Cell

Each boundary scan cell contains two flip-flops. The input of the capture flip-flop is connected by a multiplexer between the scan chain serial input or (one of) the signal input(s). The capture flip-flop output is connected to the scan chain serial output. The capture flip-flop is clocked by a gated version of TCK. Each boundary scan cell also contains an update flip-flop whose data input is the scan chain serial output from that cell. The output from the update flip-flop can be connected to the signal output via the multiplexer.

Figure 16-2. Schematic of the Boundary Scan Cell



16.1.4 Boundary Scan Cell Control Signals

There are four principal control signals to each boundary scan cell.

CLKDR - Clock Data Register

Connected to the clock input on the capture flip-flop.

SHIFTDR - Shift Data Register

This controls the capture flip-flop input multiplexer.

0 = Signal input

1 = Serial input

UPDATEDR - Update Data Register

Connected to the update flip-flop clock.

MODE - Mode Data

This selects the signal output.

0 = Signal in

1 = Update flip-flop out

In normal operation, the signal inputs and outputs to each boundary scan cell are routed straight through via the output multiplexers making each cell transparent.

16.2 VS2000 Boundary Scan Control Registers

16.2.1 Bypass Register

This is just a single flip-flop which allows boundary scan data to be passed synchronously from TDI to TDO.

16.2.2 Identification Register

This is 32-bit register holds the chip ID. The chip ID can be clocked out of TDO by issuing the IDCODE command and then clocking TDO in the SHIFT_DR state for 32 cycles. The chip ID is hardwired at device fabrication.

Note: The VS2000 JTAG ID is $02001427.

16.2.3 Instruction Register

This is 5-bit register is used to determine the mode of operation of the boundary scan circuit. Principally, this means selecting which of the data registers is connected to TDO and what mode of operation the boundary scan cells are in.

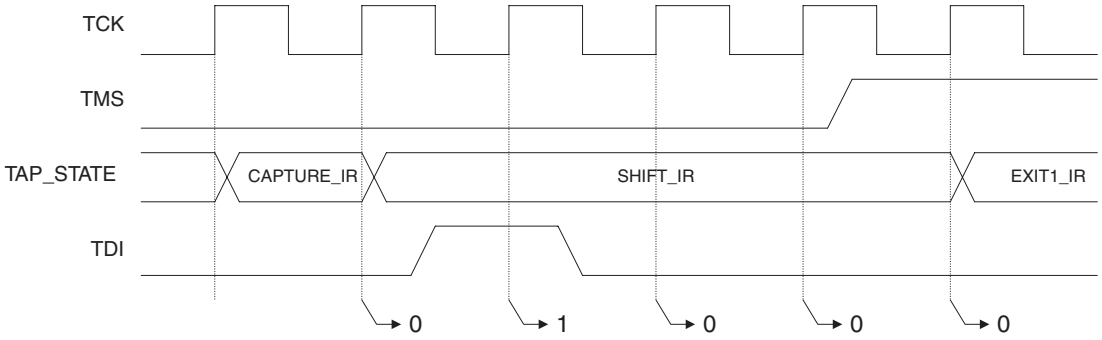
16.3 Boundary Scan Commands

The VS2000 supports four boundary scan commands:

- SAMPLE/PRELOAD
- BYPASS
- EXTEST
- IDCODE

Each of these commands has an associated 5-bit opcode. These opcodes are shifted LSB-first into the instruction register while the TAP controller state machine is in the SHIFT_IR state. Figure 16-3 demonstrates this process.

Figure 16-3. Opcode IDCODE (00010) Shifted into the Instruction Register



Note that the first bit is clocked in on the same clock edge on which the state machine transitions to the SHIFT_IR state and the last bit is clocked in on the clock edge on which the state machine transitions to the EXIT1_IR state.

16.3.1 SAMPLE/PRELOAD (00001)

The SAMPLE/PRELOAD command allows the user to take a snapshot of the device data without interrupting normal device operation.

During the capture phase, the multiplexers preceding the capture register select the device data signals which are then clocked into the capture registers. The multiplexers at the outputs of the update registers also select the device data signals to prevent interruption to the normal function of the device.

During the shift phase, the data stored in the capture registers is shifted around the device periphery and out on the TDO pin while new data is shifted in on the TDI pin.

During the update phase the data in the capture registers is passed through to the update registers. This data can then be used in a subsequent EXTEST command.

Figure 16-4. Capture Phase of SAMPLE/PRELOAD Command

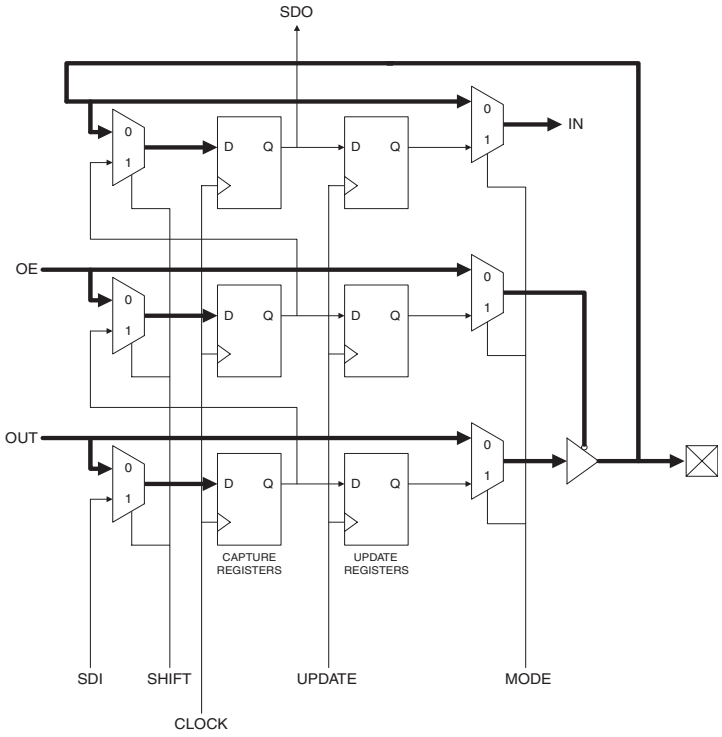
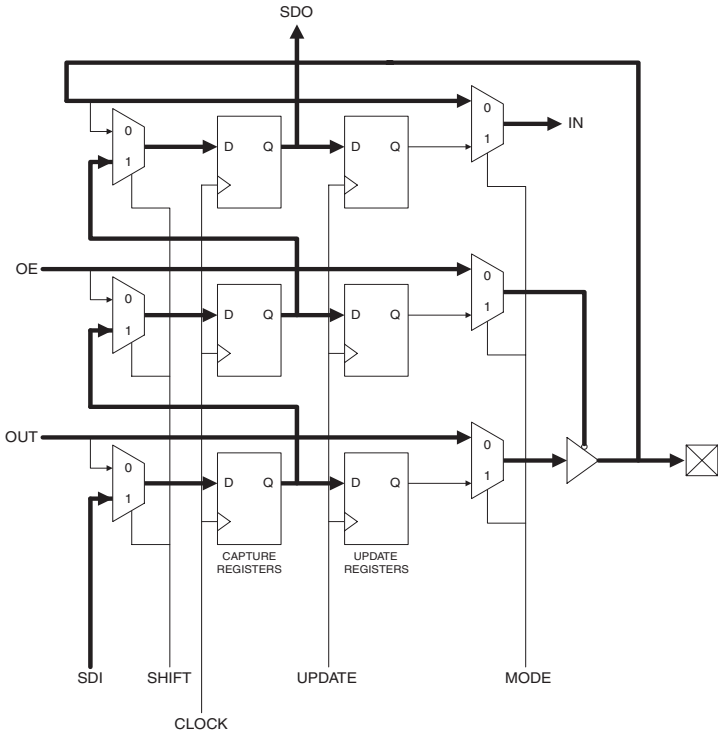


Figure 16-5. Shift/Update Phase of SAMPLE/PRELOAD Command



16.3.2 EXTEST (00000)

This command can be used to check external pin connections between devices and also to initialize memory attached to the pins of the VS2000. Unlike the SAMPLE/PRELOAD mode, EXTEST allows data to be forced onto the pin signals.

EXTEST uses data from the update registers to drive the OUT, OE and IN signals. When the EXTEST command is entered, the multiplexers select the update register data. This means that data stored in the update registers from a previous SAMPLE/PRELOAD or EXTEST test run can be driven onto the pin signals.

Figure 16-6. Capture Phase of EXTEST Command

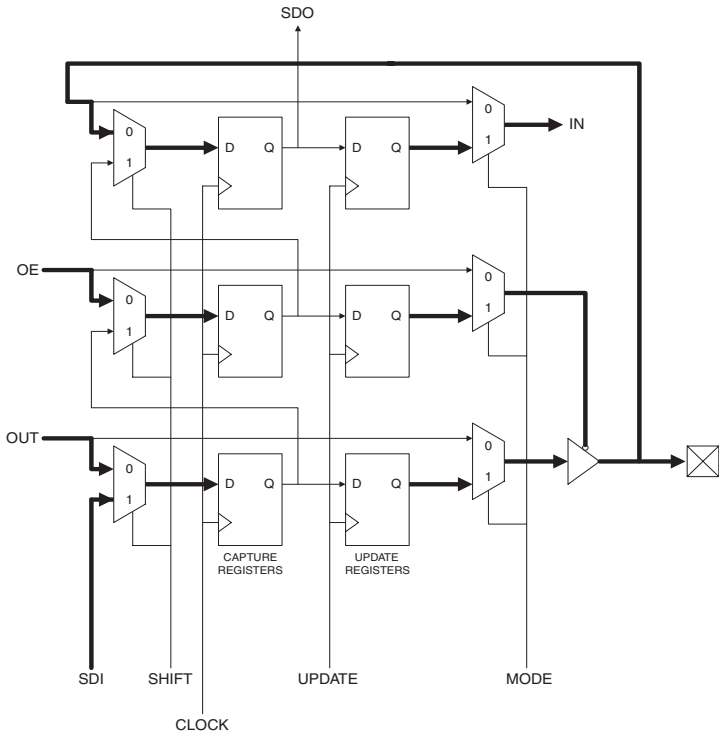
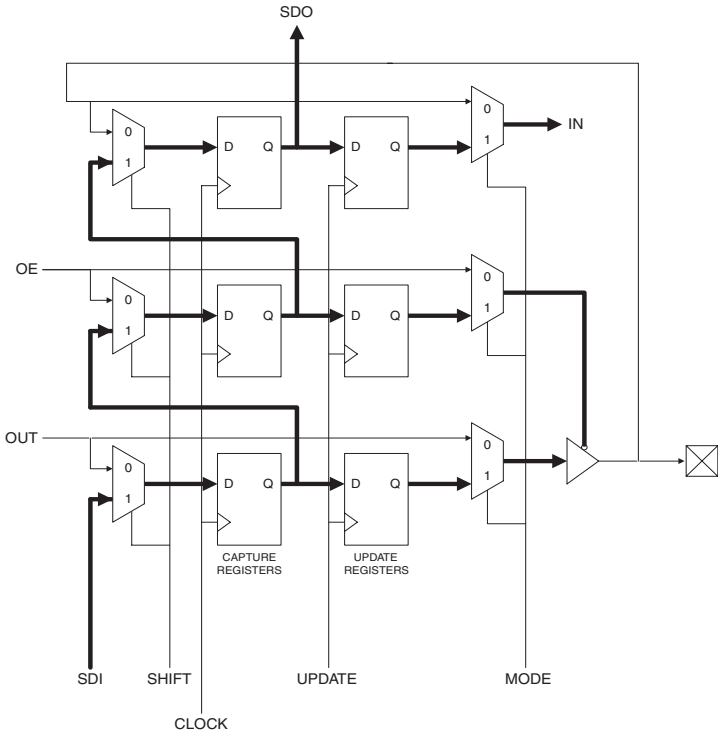


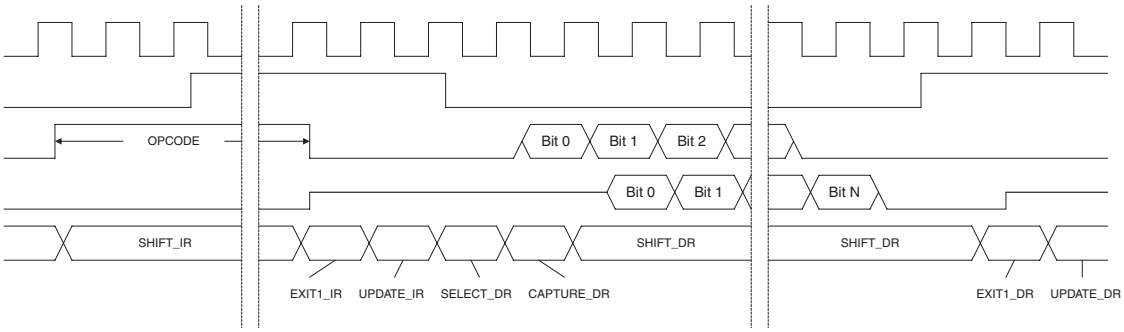
Figure 16-7. Shift/Update Phase of EXTEST Command



16.3.3 BYPASS (11111)

The BYPASS command causes data shifted into TDI on the rising edge of TCK to be shifted out of TDO on the falling edge of the same TCK clock pulse.

Figure 16-8. Timing Diagram of the Bypass Command



16.3.4 IDCODE (00010)

The IDCODE command allows devices in a JTAG chain to be identified. When the IDCODE command is shifted into the instruction register, the IDCODE for the VS2000 will be loaded between the TDI and TDO ports and can be shifted out of TDO during the SHIFT_DR phase.

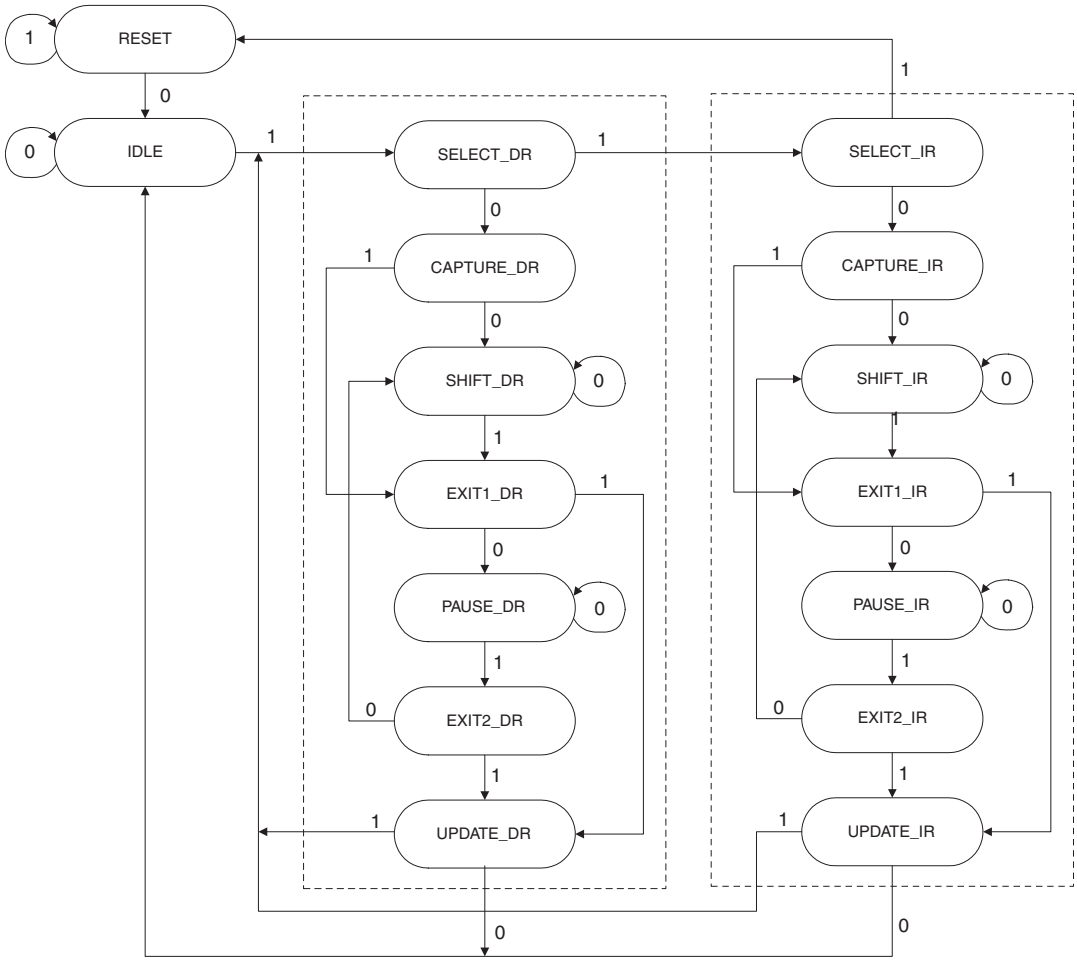
16.4 TAP Controller

16.4.1 TAP State Machine

The Test Access Port (TAP) controller is a state machine with 16 states which is split into two modes. One mode is concerned with manipulating the instruction register, the other mode with manipulating the three data registers that can be connected to TDO. The state machine transitions are controlled by the TMS signal.

The states and the state progression are shown in Figure 16-9.

Figure 16-9. TAP Controller State Machine Transition Diagram



To initialise the TAP controller, set TMS = 1 and pull TRST_L low for a clock cycle. Subsequently pull TMS low. This will advance the state machine to the idle state and hold it there.

Note: Holding TMS high for five clock cycles will also return the TAP controller to the reset state.

16.5 Boundary Scan Order

Table 16-2 lists the VS2000 signals and their order on the boundary scan chain. This list is needed to determine the order of data bits to be shifted in on TDI and to allow the serial data from TDO to be decoded.

Table 16-2. Boundary Scan Order

Position	Cell Type	External Pin	Pin Signal
254	BC_7	PC0	BIDIR
253	BC_2	PC0	OE
252	BC_7	PC1	BIDIR
251	BC_2	PC1	OE
250	BC_7	PC2	BIDIR
249	BC_2	PC2	OE
248	BC_7	PC3	BIDIR
247	BC_2	PC3	OE
246	BC_7	PC4	BIDIR
245	BC_2	PC4	OE
244	BC_7	PC5	BIDIR
243	BC_2	PC5	OE
242	BC_7	PD0	BIDIR
241	BC_2	PD0	OE
240	BC_7	PD1	BIDIR
239	BC_2	PD1	OE
238	BC_7	PD2	BIDIR
237	BC_2	PD2	OE
236	BC_7	PD3	BIDIR
235	BC_2	PD3	OE
234	BC_7	PD4	BIDIR
233	BC_2	PD4	OE
232	BC_7	PD5	BIDIR
231	BC_2	PD5	OE
230	BC_1	MTEN	OUTPUT
229	BC_1	$\overline{\text{WDTO}}$	OUTPUT

Note 1. DATA ORDER: TDI is connected to the boundary scan cell for PC0 - BIDIR and TDO is connected to the boundary scan cell for PB10 - OE. When shifting data in, therefore, data for PB10 - OE should be shifted in first.

Table 16-2. Boundary Scan Order (continued)

Position	Cell Type	External Pin	Pin Signal
228	BC_1	MTD3	OUTPUT
227	BC_1	MTD2	OUTPUT
226	BC_1	MTD1	OUTPUT
225	BC_1	MTD0	OUTPUT
224	BC_1	MCRS	INPUT
223	BC_1	MCOLL	INPUT
222	BC_1	MTERR	OUTPUT
221	BC_4	MRERR	CLOCK
220	BC_4	PLLEN	CLOCK
219	BC_4	TESTMODE	CLOCK
218	BC_4	$\overline{\text{MTCLK}}$	CLOCK
217	BC_4	$\overline{\text{MRCLK}}$	CLOCK
216	BC_1	MWAIT	INPUT
215	BC_4	$\overline{\text{SYSRST}}$	CLOCK
214	BC_4	IRQ	CLOCK
213	BC_4	WIDTH	CLOCK
212	BC_1	MRDV	INPUT
211	BC_1	MRD0	INPUT
210	BC_1	MRD1	INPUT
209	BC_1	MRD2	INPUT
208	BC_1	MRD3	INPUT
207	BC_4	PWDN	CLOCK
206	BC_1	MDC	OUTPUT
205	BC_7	MDIO	BIDIR
204	BC_2	MDIO	OE
203	BC_1	A0	OUTPUT
202	BC_2	A[23:0], RD, WR, CS[3:0], BL[3:0]	OE
201	BC_1	A1	OUTPUT
200	BC_1	A2	OUTPUT

Note 1. DATA ORDER: TDI is connected to the boundary scan cell for PC0 - BIDIR and TDO is connected to the boundary scan cell for PB10 - OE. When shifting data in, therefore, data for PB10 - OE should be shifted in first.

Table 16-2. Boundary Scan Order (continued)

Position	Cell Type	External Pin	Pin Signal
199	BC_1	A3	OUTPUT
198	BC_1	A4	OUTPUT
197	BC_1	A5	OUTPUT
196	BC_1	A6	OUTPUT
195	BC_1	A7	OUTPUT
194	BC_1	A8	OUTPUT
193	BC_1	A9	OUTPUT
192	BC_1	A10	OUTPUT
191	BC_1	A11	OUTPUT
190	BC_1	A12	OUTPUT
189	BC_1	A13	OUTPUT
188	BC_1	A14	OUTPUT
187	BC_1	A15	OUTPUT
186	BC_1	A16	OUTPUT
185	BC_1	A17	OUTPUT
184	BC_1	A18	OUTPUT
183	BC_1	A19	OUTPUT
182	BC_1	A20	OUTPUT
181	BC_1	A21	OUTPUT
180	BC_1	A22	OUTPUT
179	BC_1	A23	OUTPUT
178	BC_7	D0	BIDIR
177	BC_2	D0	OE
176	BC_7	D1	BIDIR
175	BC_2	D1	OE
174	BC_7	D2	BIDIR
173	BC_2	D2	OE
172	BC_7	D3	BIDIR
171	BC_2	D3	OE

Note 1. DATA ORDER: TDI is connected to the boundary scan cell for PC0 - BIDIR and TDO is connected to the boundary scan cell for PB10 - OE. When shifting data in, therefore, data for PB10 - OE should be shifted in first.

Table 16-2. Boundary Scan Order (continued)

Position	Cell Type	External Pin	Pin Signal
170	BC_7	D4	BIDIR
169	BC_2	D4	OE
168	BC_7	D5	BIDIR
167	BC_2	D5	OE
166	BC_7	D6	BIDIR
165	BC_2	D6	OE
164	BC_7	D7	BIDIR
163	BC_2	D7	OE
162	BC_7	D8	BIDIR
161	BC_2	D8	OE
160	BC_7	D9	BIDIR
159	BC_2	D9	OE
158	BC_7	D10	BIDIR
157	BC_2	D10	OE
156	BC_7	D11	BIDIR
155	BC_2	D11	OE
154	BC_7	D12	BIDIR
153	BC_2	D12	OE
152	BC_7	D13	BIDIR
151	BC_2	D13	OE
150	BC_7	D14	BIDIR
149	BC_2	D14	OE
148	BC_7	D15	BIDIR
147	BC_2	D15	OE
146	BC_7	D16	BIDIR
145	BC_2	D16	OE
144	BC_7	D17	BIDIR
143	BC_2	D17	OE
142	BC_7	D18	BIDIR

Note 1. DATA ORDER: TDI is connected to the boundary scan cell for PC0 - BIDIR and TDO is connected to the boundary scan cell for PB10 - OE. When shifting data in, therefore, data for PB10 - OE should be shifted in first.

Table 16-2. Boundary Scan Order (continued)

Position	Cell Type	External Pin	Pin Signal
141	BC_2	D18	OE
140	BC_7	D19	BIDIR
139	BC_2	D19	OE
138	BC_7	D20	BIDIR
137	BC_2	D20	OE
136	BC_7	D21	BIDIR
135	BC_2	D21	OE
134	BC_7	D22	BIDIR
133	BC_2	D22	OE
132	BC_7	D23	BIDIR
131	BC_2	D23	OE
130	BC_7	D24	BIDIR
129	BC_2	D24	OE
128	BC_7	D25	BIDIR
127	BC_2	D25	OE
126	BC_7	D26	BIDIR
125	BC_2	D26	OE
124	BC_7	D27	BIDIR
123	BC_2	D27	OE
122	BC_7	D28	BIDIR
121	BC_2	D28	OE
120	BC_7	D29	BIDIR
119	BC_2	D29	OE
118	BC_7	D30	BIDIR
117	BC_2	D30	OE
116	BC_7	D31	BIDIR
115	BC_2	D31	OE
114	BC_1	CS0	OUTPUT
113	BC_1	CS1	OUTPUT

Note 1. DATA ORDER: TDI is connected to the boundary scan cell for PC0 - BIDIR and TDO is connected to the boundary scan cell for PB10 - OE. When shifting data in, therefore, data for PB10 - OE should be shifted in first.

Table 16-2. Boundary Scan Order (continued)

Position	Cell Type	External Pin	Pin Signal
112	BC_1	CS2	OUTPUT
111	BC_1	CS3	OUTPUT
110	BC_1	BL0	OUTPUT
109	BC_1	BL1	OUTPUT
108	BC_1	BL2	OUTPUT
107	BC_1	BL3	OUTPUT
106	BC_1	RD	OUTPUT
105	BC_1	WR	OUTPUT
104	BC_1	INSTR	OUTPUT
103	BC_2	INSTR	OE
102	BC_7	PE0	BIDIR
101	BC_2	PE0	OE
100	BC_7	PE1	BIDIR
99	BC_2	PE1	OE
98	BC_7	PE2	BIDIR
97	BC_2	PE2	OE
96	BC_7	PE3	BIDIR
95	BC_2	PE3	OE
94	BC_7	PE4	BIDIR
93	BC_2	PE4	OE
92	BC_7	PE5	BIDIR
91	BC_2	PE5	OE
90	BC_7	PE6	BIDIR
89	BC_2	PE6	OE
88	BC_7	PE7	BIDIR
87	BC_2	PE7	OE
86	BC_7	PA0	BIDIR
85	BC_2	PA0	OE
84	BC_7	PA1	BIDIR

Note 1. DATA ORDER: TDI is connected to the boundary scan cell for PC0 - BIDIR and TDO is connected to the boundary scan cell for PB10 - OE. When shifting data in, therefore, data for PB10 - OE should be shifted in first.

Table 16-2. Boundary Scan Order (continued)

Position	Cell Type	External Pin	Pin Signal
83	BC_2	PA1	OE
82	BC_7	PA2	BIDIR
81	BC_2	PA2	OE
80	BC_7	PA3	BIDIR
79	BC_2	PA3	OE
78	BC_7	PA4	BIDIR
77	BC_2	PA4	OE
76	BC_7	PA5	BIDIR
75	BC_2	PA5	OE
74	BC_7	PA6	BIDIR
73	BC_2	PA6	OE
72	BC_7	PA7	BIDIR
71	BC_2	PA7	OE
70	BC_7	PA8	BIDIR
69	BC_2	PA8	OE
68	BC_7	PA9	BIDIR
67	BC_2	PA9	OE
66	BC_1	CLKOUT	OUTPUT
65	BC_7	PA10	BIDIR
64	BC_2	PA19	OE
63	BC_7	PA11	BIDIR
62	BC_2	PA11	OE
61	BC_7	PA12	BIDIR
60	BC_2	PA12	OE
59	BC_7	PA13	BIDIR
58	BC_2	PA13	OE
57	BC_7	PA14	BIDIR
56	BC_2	PA14	OE
55	BC_7	PA15	BIDIR

Note 1. DATA ORDER: TDI is connected to the boundary scan cell for PC0 - BIDIR and TDO is connected to the boundary scan cell for PB10 - OE. When shifting data in, therefore, data for PB10 - OE should be shifted in first.

Table 16-2. Boundary Scan Order (continued)

Position	Cell Type	External Pin	Pin Signal
54	BC_2	PA15	OE
53	BC_7	PA16	BIDIR
52	BC_2	PA16	OE
51	BC_7	PA17	BIDIR
50	BC_2	PA17	OE
49	BC_7	PA18	BIDIR
48	BC_2	PA18	OE
47	BC_7	PA19	BIDIR
46	BC_2	PA19	OE
45	BC_7	PA20	BIDIR
44	BC_2	PA20	OE
43	BC_7	PA21	BIDIR
42	BC_2	PA21	OE
41	BC_7	PA22	BIDIR
40	BC_2	PA22	OE
39	BC_7	PA23	BIDIR
38	BC_2	PA23	OE
37	BC_7	PA24	BIDIR
36	BC_2	PA24	OE
35	BC_7	PA25	BIDIR
34	BC_2	PA25	OE
33	BC_7	PA26	BIDIR
32	BC_2	PA26	OE
31	BC_7	PA27	BIDIR
30	BC_2	PA27	OE
29	BC_7	PA28	BIDIR
28	BC_2	PA28	OE
27	BC_7	PA29	BIDIR
26	BC_2	PA39	OE

Note 1. DATA ORDER: TDI is connected to the boundary scan cell for PC0 - BIDIR and TDO is connected to the boundary scan cell for PB10 - OE. When shifting data in, therefore, data for PB10 - OE should be shifted in first.

Table 16-2. Boundary Scan Order (continued)

Position	Cell Type	External Pin	Pin Signal
25	BC_7	PA30	BIDIR
24	BC_2	PA30	OE
23	BC_7	PA31	BIDIR
22	BC_2	PA31	OE
21	BC_7	PB0	BIDIR
20	BC_2	PB0	OE
19	BC_7	PB1	BIDIR
18	BC_2	PB1	OE
17	BC_7	PB2	BIDIR
16	BC_2	PB2	OE
15	BC_7	PB3	BIDIR
14	BC_2	PB3	OE
13	BC_7	PB4	BIDIR
12	BC_2	PB4	OE
11	BC_7	PB5	BIDIR
10	BC_2	PB5	OE
9	BC_7	PB6	BIDIR
8	BC_2	PB6	OE
7	BC_7	PB7	BIDIR
6	BC_2	PB7	OE
5	BC_7	PB8	BIDIR
4	BC_2	PB8	OE
3	BC_7	PB9	BIDIR
2	BC_2	PB9	OE
1	BC_7	PB10	BIDIR
0	BC_2	PB10	OE

Note 1. DATA ORDER: TDI is connected to the boundary scan cell for PC0 - BIDIR and TDO is connected to the boundary scan cell for PB10 - OE. When shifting data in, therefore, data for PB10 - OE should be shifted in first.

Appendix A. List of Figures

Appendix B. List of Tables

Warranties and Disclaimers: This document is published for informational purposes only and is provided "as is" and without warranty of any kind. Velocity Semiconductor™ disclaims any and all warranties with respect to this information, whether express or implied, including without limitation, the implied warranties of merchantability or fitness for a particular purpose. To the extent permitted by law, in no event shall Velocity Semiconductor be liable for any direct, indirect, special, incidental or consequential damages arising from the use of, or reliance upon, any information in this document or any error or omission in such information. Velocity Semiconductor reserves the right to update, change or remove portions of the information provided herein without notice.

Velocity Semiconductor and the Velocity Semiconductor logo are trademarks of Velocity Semiconductor Inc.

Lightfoot is a trademark of Digital Communication Technologies, Ltd.

VELOCITY™ SEMICONDUCTOR

5673 West Las Positas Boulevard
Suite 206
Pleasanton, CA 94588
Phone: (925) 924-0550
Fax: (925) 924-0555

Unit 5 Windmill Business Village
Brooklands Close
Sunbury
Middlesex TW16 7DY
United Kingdom
Phone: +44 (0)1932-784858
Fax: +44 (0)1932-782811

info@velocitysemi.com

www.velocitysemi.com

0080-v4–06/07/04–xM