Interfacing the PowerSpan II™ with the Wintegra Winpath™

80A1010_AN005_05

November 4, 2009

6024 Silver Creek Valley Road San Jose, California 95138
Telephone: (408) 284-8200 • FAX: (408) 284-3572
Printed in U.S.A.

GENERAL DISCLAIMER

Integrated Device Technology, Inc. ("IDT") reserves the right to make changes to its products or specifications at any time, without notice, in order to improve design or performance. IDT does not assume responsibility for use of any circuitry described herein other than the circuitry embodied in an IDT product. Disclosure of the information herein does not convey a license or any other right, by implication or otherwise, in any patent, trademark, or other intellectual property right of IDT. IDT products may contain errata which can affect product performance to a minor or immaterial degree. Current characterized errata will be made available upon request. Items identified herein as "reserved" or "undefined" are reserved for future definition. IDT does not assume responsibility for conflicts or incompatibilities arising from the future definition of such items. IDT products have not been designed, tested, or manufactured for use in, and thus are not warranted for, applications where the failure, malfunction, or any inaccuracy in the application carries a risk of death, serious bodily injury, or damage to tangible property. Code examples provided herein by IDT are for illustrative purposes only and should not be relied upon for developing applications. Any use of such code examples shall be at the user's sole risk.

1. Interfacing the PowerSpan II with the Wintegra Winpath

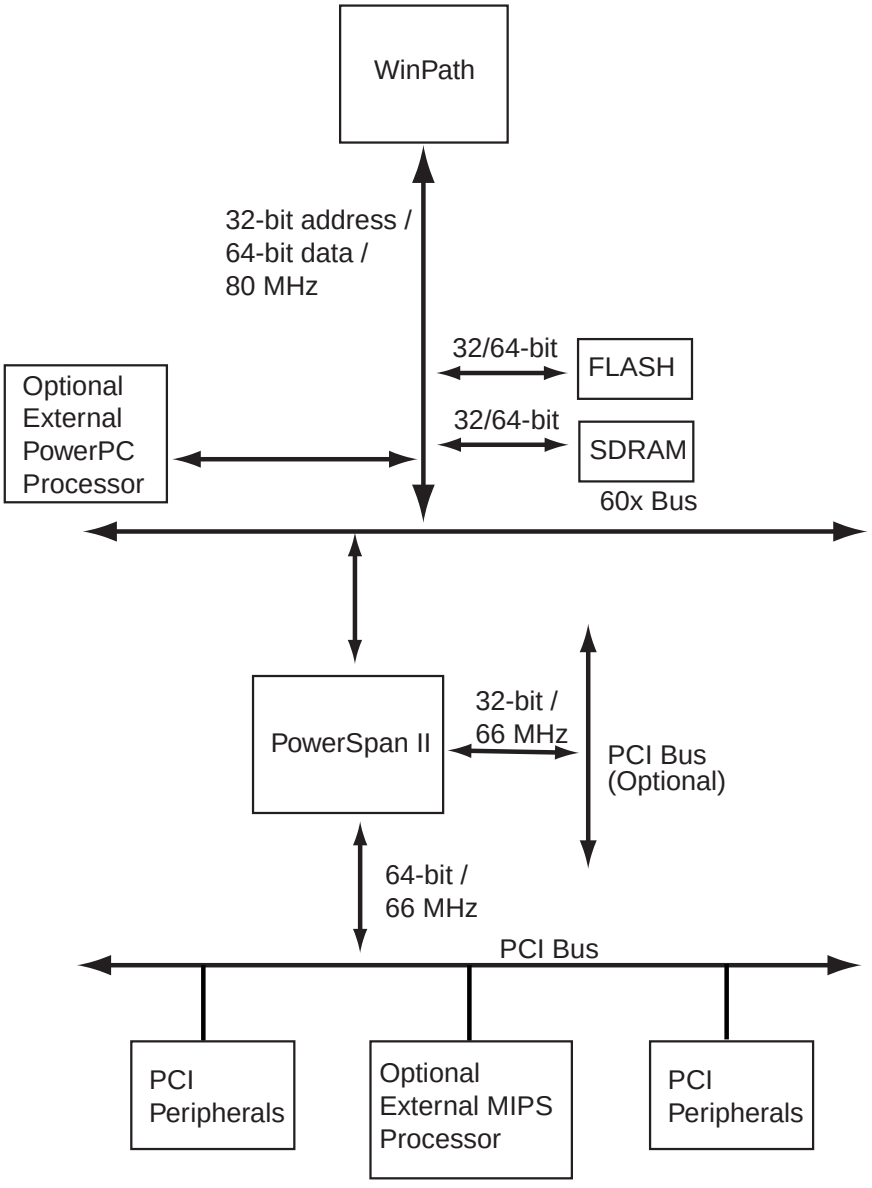
This application note details the interface requirements for using the Wintegra WinPath and the PowerSpan II in an application. The following sections are detailed in the application note:

1.1 Overview

The Wintegra WinPath and the IDT PowerSpan II can be used in a wide range of system applications. This application note only focuses on the requirements for connecting the host bus interface between the two devices. In “Typical Applications” on page 28 of this document, specific applications such as host and adapter card implementations are discussed in detail.

Figure 1 illustrates a simplified block diagram between the WinPath and the PowerSpan II.

Figure 1: System Diagram



1.1.1 Architectural Components

The major components of this application note are the WinPath and the PowerSpan II.

1.1.1.1 WinPath

The WinPath is the flagship semiconductor product family from Wintegra. Targeted for the Access Infrastructure, WinPath offers designers a comprehensive method to handle the data path. ATM and IP can be independently selected as a transport protocol per port. A full set of IP services may be offered over any L2 protocol, including ATM, PPP and Ethernet. Any port can instantly migrate from ATM to IP transport with zero hardware changes.

The WinPath provides a world-class set of ATM features for both termination and switching. It also includes interworking from ATM to TDM and from ATM to IP transport media.

The WinPath complements the protocol offering with an ideal mix of IP quality-of-service support, Layer 2/3 Ethernet switching and Point-to-Point protocol (PPP) support features, all which are increasingly important in access equipment designs.

1.1.1.2 PowerSpan II

The IDT PowerSpan II is a Multi-port PCI Bus Switch that bridges PCI to PowerPC^{TM} processors and the Wintegra WinPath^{TM} processor. PowerSpan II is available in either a single PCI or dual PCI variant. PowerSpan II defines a new level of PCI bus switch flexibility.

PowerSpan II has a Switched PCI architecture, which maximizes transfer concurrency and improves the efficiency and burst performance on the PCI and processor buses. Along with a revolutionary Switched PCI architecture, PowerSpan II features integrated PCI-to-PCI bridging, concurrent prefetch read capability, multi-port Direct Memory Access (DMA) operation and CompactPCI Hot Swap support. It is now available in 23mm and 27mm BGA packages.

1.2 Hardware Connection

This section defines how the signals must be connected between the WinPath and the PowerSpan II devices to ensure proper operation.

1.2.1 Host Bus Signal Connection

The host bus interface signals between the WinPath and the PowerSpan II can be directly connected. External pull-up resistors must be connected to all bus control signals to ensure that they are held in an inactive state.

The host bus interface implements the PowerPC 60x bus protocol.

Figure 2 illustrates the signals that must be connected in a joint WinPath and PowerSpan II application. The WinPath processor is the arbiter in the system diagram.

Figure 2 does not illustrate the external pull-up resistors for the directly connected signals. For pull-up resistor information, refer to Table 1.

Figure 2: WinPath and PowerSpan II Connection Diagram

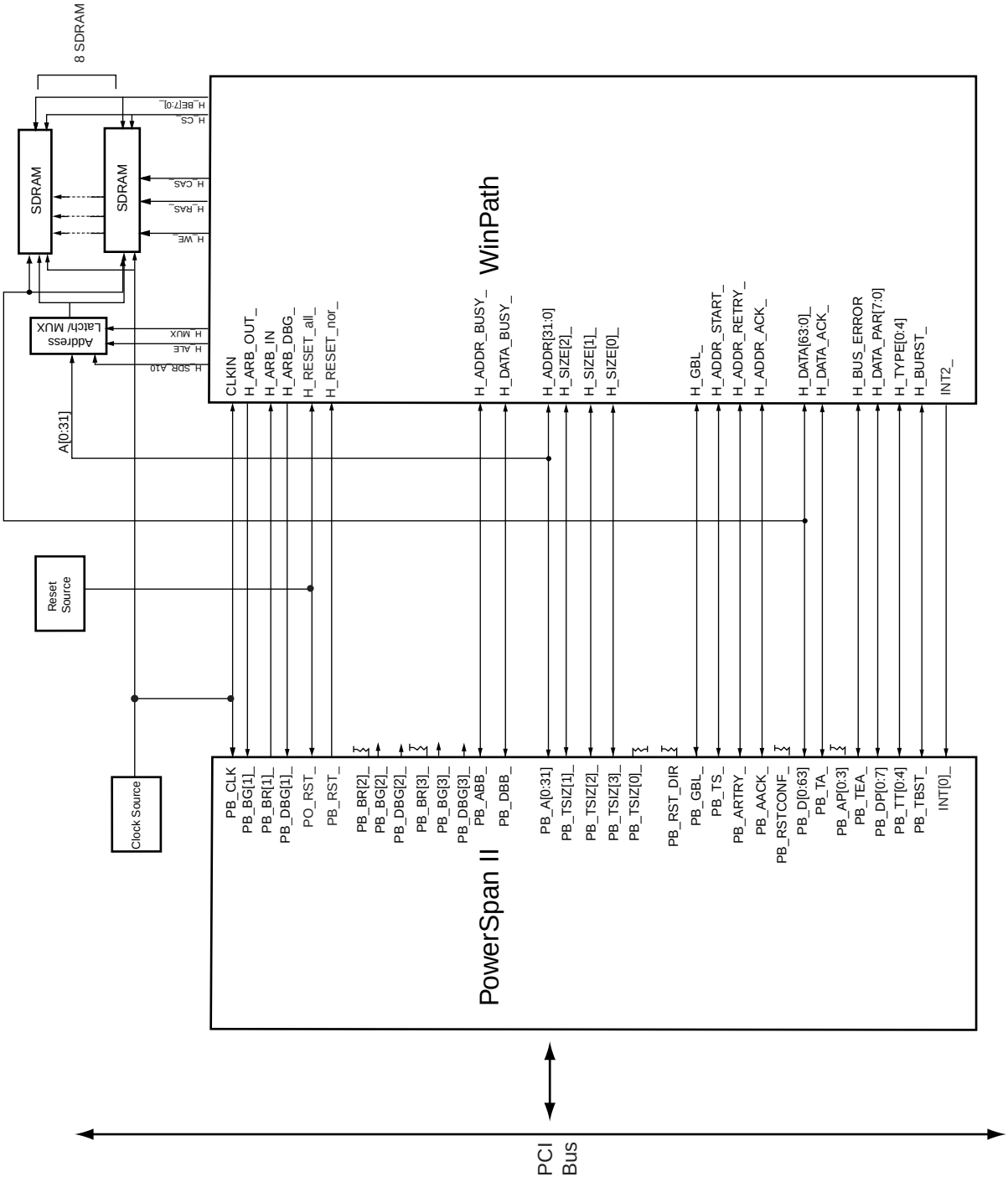


Figure 2 assumes WinPath's arbiter is enabled. The routing of the interrupts is dependent on the application. This diagram shows the interrupts being routed from the WinPath processor through PowerSpan II to the PCI host.

The SDRAM memory is controlled by the WinPath SDRAM memory controller. This memory must be 64-bit in a WinPath and PowerSpan II application. The SDRAM interface requires external address multiplexing to support accesses by the PowerSpan II.

The FLASH memory is controlled by the WinPath memory controller. This memory must be 64-bit in a WinPath and PowerSpan II application.

WinPath includes an optimized interface for controlling latch and multiplexing devices. For example, an integrated latch/mux device, such as the 74AVCLH16260 device, can be used to limit the delay through two devices.

1.2.1.1 Signal Connection and Terminations

The processor bus signals in Table 1 are connected between the WinPath and the PowerSpan II. The table also shows termination requirements for the connections.

Pull-up resistors are not required on the processor bus address (PB_A[0:31]) and data (PB_D[0:63]) signals to guarantee functional operation of the PowerSpan II. However, adding resistors to the address and data signals minimizes the current drawn by the PowerSpan II's tristated buffers when the bus is in an idle condition. The system designer must decide whether to add these resistors to the address and data bus.

Table 1: Signal Connection between WinPath and PowerSpan II

WinPath Signal	PowerSpan II Signal	Termination
H_ADDR[31:0][a]	PB_A[0:31]	No requirement
H_DATA[63:0][b]	PB_D[0:63]	No requirement
H_DATA_PAR[7:0][c]	PB_DP[0:7]	No requirement
H_ADDR_PAR[3:0][d]	PB_AP[0:3]	No requirement
H_TYPE_4	PB_TT[4]	No requirement
H_TYPE_3	PB_TT[3]	No requirement
H_TYPE_2	PB_TT[2]	No requirement
H_TYPE_1	PB_TT[1]	No requirement
H_TYPE_0	PB_TT[0]	No requirement
H_SIZE_2	PB_TSIZ[1]	No requirement
H_SIZE_1	PB_TSIZ[2]	No requirement

Table 1: Signal Connection between WinPath and PowerSpan II *(Continued)*

WinPath Signal	PowerSpan II Signal	Termination
H_SIZE_0	PB_TSIZ[3]	No requirement
H_ADDR_START_	PB_TS_	Pull-up 10 Kohm resistor
H_ADDR_BUSY_	PB_ABB_	Pull-up: 10 Kohm resistor
H_ADDR_ACK_	PB_AACK_	Pull-up: 10 Kohm resistor
H_ADDR_RETRY_	PB_ARTRY_	Pull-up: 10 Kohm resistor
H_DATA_ACK_	PB_TA_	Pull-up: 10 Kohm resistor
H_BUS_ERROR_	PB_TEA_	Pull-up: 10 Kohm resistor
H_BURST_	PB_TBST_	Pull-up: 10 Kohm resistor
H_DATA_BUSY_	PB_DBB_	Pull-up: 10 Kohm resistor
H_GBL_	PB_GBL_	Pull-up: 10 Kohm resistor
INT2_	INT[0]	Pull-up: 10 Kohm resistor
H_RESET_all	PO_RST_	Pull-up: 1.2 Kohm resistor
H_RESET_nor	PB_RST_	Pull-up: 1.2 Kohm resistor

a. Mapping for the address signals: H_ADD[31] - PB_A[0], H_ADDR[0] - PB_A[31]

b. Mapping for the data signals: H_DATA[63] - PB_D[0], H_DATA[0] - PB_D[63]

c. Mapping for the data parity signals: H_DATA_PAR[7] - H_DATA_PAR[0], PB_DP[0] - PB_DP[7].

d. Mapping for the address parity signals: H_ADDR_PAR[3] - H_ADDR_PAR[0], PB_AP[0] - PB_AP[3]. Available on WinPath Revision B and later devices.

1.2.1.2 PowerSpan II Specific Signals

PowerSpan II has signals on its host bus interface that are not supported on the WinPath.

Table 2: PowerSpan II Processor Signals not Connected to WinPath

PowerSpan II Processor Bus Signal	Signal Description	Termination
PB_FAST	Input (multiplexed pin): Used for PowerSpan II power-up configuration/PB clock PLL frequency range.	This signal can be driven by an external CPLD or pulled-up/ pulled-down to a fixed level (depends on the system configuration). Refer to the *PowerSpan II User Manual* for more information.
PB_RSTCONF_	Input: Used for PowerSpan II configuration slave mode.This mode is not a supported configuration by WinPath.	Pull-up: 10 Kohm resistor
PB_RST_DIR	Input: Used for determining PB_RST_ pin direction. 1= PB_RST_ is an output 0= PB_RST_ is an input	Pull-up (10 Kohm resistor) or pull-down (1.0 Kohm resistor) depending on where the host resides.
PB_AP[0:3]	Bidirectional: Use for the processor supporting address parity check.This functionality is supported by WinPath Revision B and later devices.	Pull-up: 10 Kohm resistor if function is not used.
PB_CI_	Output: Use for cache inhibit functionality. This functionality is not supported by WinPath.	Open
PB_DVAL_	Bidirectional: Used for extended cycles on the host bus. This functionality is not a supported by WinPath.	Pull-up: 10 Kohm resistor
PB_TSIZ[0]	Bidirectional: Used for PowerSpan II to specify the type of the transaction. This functionality is not a supported by WinPath.	Pull-down: 1.0 Kohm resistor
INT[5:1]	Bidirectional (multiplexed pin): Used for PowerSpan II power-up configuration/General purpose interrupt pins.	If not using for interrupt purpose, pull-up (10 Kohm resistor) or pull-down (1.0 Kohm resistor) depending on the system configuration.

PCI Signals

PowerSpan II's PCI signals can be directly connected to the appropriate PCI signal on the motherboard or the PCI connector. Pull-up resistors may, depending on the application, need to be added to the PCI bus control signals. If a local PCI bus is designed on the motherboard, then pull-up resistors are required (refer to the *PCI 2.2 Specification* for more information). For host bridging applications, the PowerSpan II P1_IDSEL and P2_IDSEL signals are not required to be implemented. In this application, the P1_IDSEL and P2_IDSEL signals must be pulled low through a pull-down resistor.

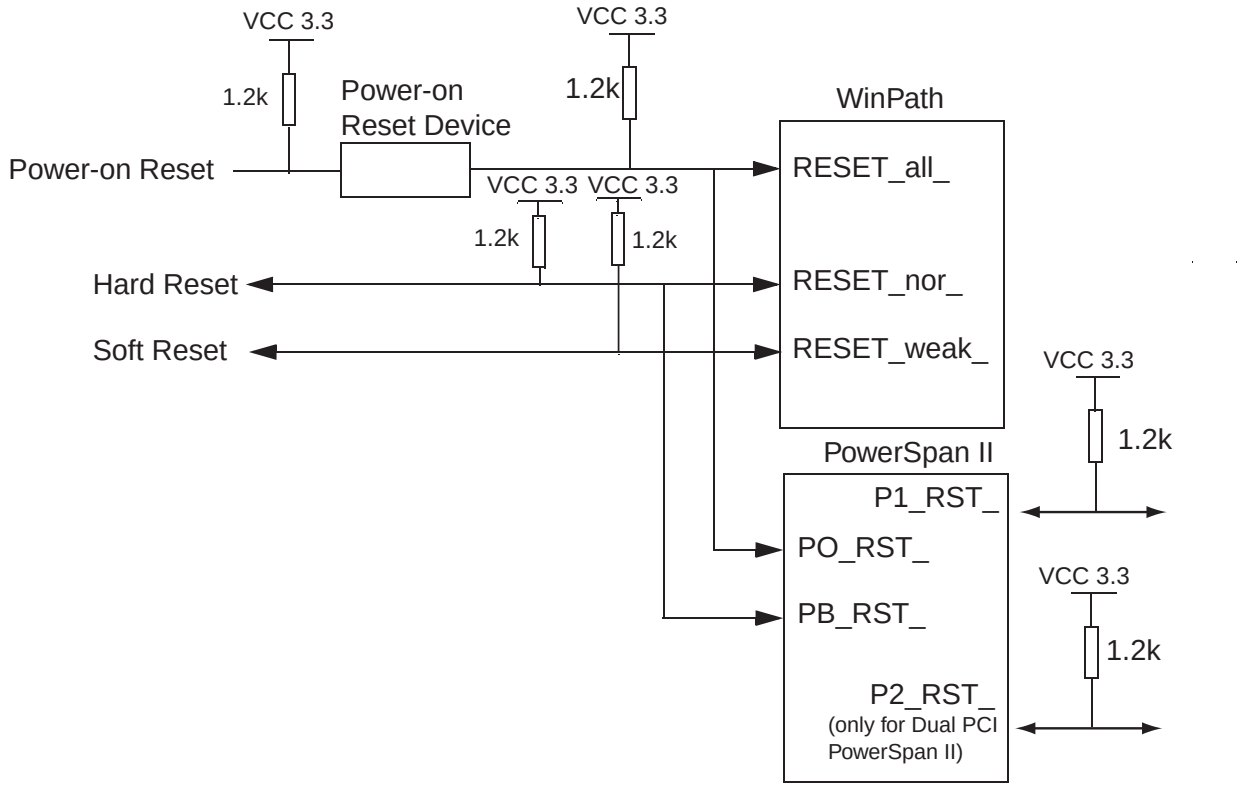
1.3 Clock, Reset, and Power-up Options

1.3.1 Resets

In a WinPath and PowerSpan II system, reset control is a requirement for proper operation. Both devices require various resets including a power-on reset and processor bus reset signals.

Figure 3 shows one possible system reset circuit for a WinPath and PowerSpan II system. Refer to “Typical Applications” on page 28 for additional details on how reset signals should be connected in certain applications.

Figure 3: System Reset Circuit



During power up, a Power-on Reset device asserts RESET_all and PO_RST_ until the power supplies are stable. During this phase, both device’s PLL clocking mechanisms gain lock and execute their reset configuration routine.

If a bus error condition occurs during normal operation the hard reset can be asserted by the Software watchdog time-out in the WinPath, or another PowerPC processor, to bring the system back to a known state.

In PCI adapter card application, the central reset resource is on the PCI host side. The PowerSpan II PB_RST_ can be configured as an output to drive the host bus' hard reset signal when it detects PCI bus reset from its PCI interface.

While in the PCI host application, the PowerSpan II propagates the host bus hard reset to its PCI bus in order to reset all the other PCI peripheral devices.

1.3.2 Power-up Options

The WinPath processor requires configuration from a serial I^2C bus or flash memory.

Some of PowerSpan II's features must be configured before the completion of the power-up reset sequence. PowerSpan II's. Multiplexed System Pins mode is used in this WinPath and PowerSpan II application. This configuration mode uses the following system pins: PB_FAST, P1_M66EN, P2_M66EN, and INT[5:1]. Either resistors (pull-up or pull-down) or external programmable logic can be used to ensure the signals are in the correct state during the power-up reset sequence to configure PowerSpan II's power-up options.

For more information on PowerSpan II power-up options refer to the *PowerSpan II User Manual*.

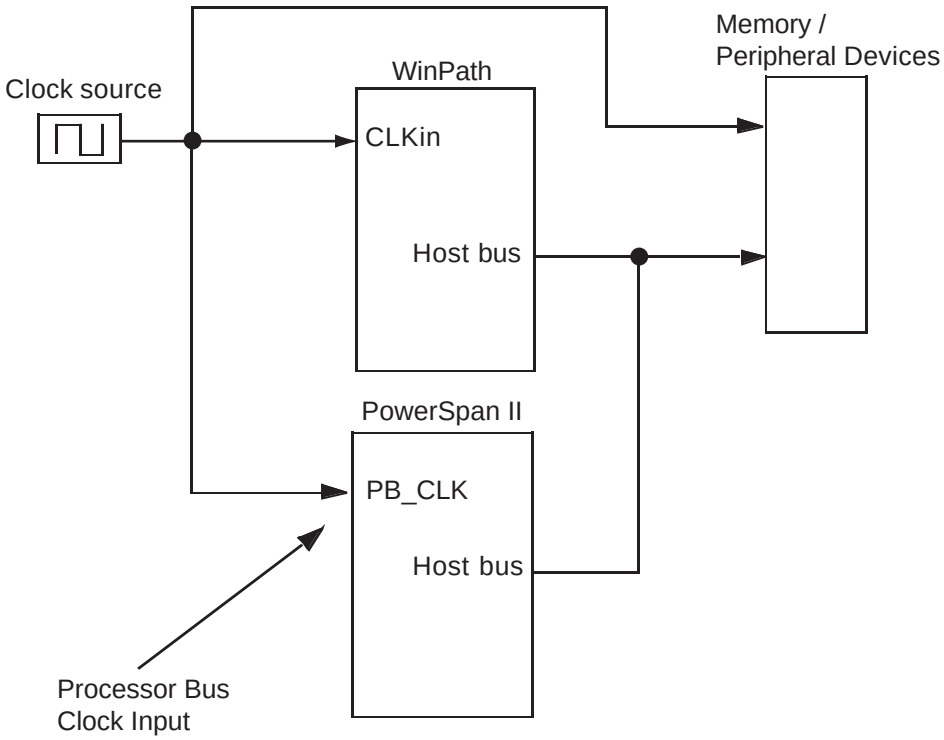
1.3.3 Clocks

The external clock generator provides the host bus clock to the WinPath, PowerSpan II, and memory devices. The WinPath processor feeds the input clock source to its PLL to generate clock sources for internal, packet and parameter bus interface clocks. The recommended mode to program the WinPath host bus clock generator is in synchronous multi-master bus mode with externally generated system clock. This mode ensures that accesses by each device to the host bus are synchronized. Refer to the WinPath *Hardware Developers Guide* for more information.

The PowerSpan II clock input should be connected to the WinPath CLK_in pin because the WinPath must work in multi-master mode with the PowerSpan II. A PLL-based clock driver can be used to ensure the zero clock skew between the devices (see Figure 4).

PowerSpan II P1_CLK, P2_CLK, and PB_CLK clock frequencies must be stable before PO_RST_ is deasserted. If a frequency change is required, a new power-up sequence (with the assertion of PO_RST_) must be initiated in order for the PLLs to lock.

Figure 4: Multi-master with DLL-based Processor Bus Clock



1.4 Arbitration

In a WinPath and PowerSpan II system, either of the devices can be programmed as the host bus arbiter. Both devices have internal host bus arbiters which support multiple masters, and can also work with external bus arbiters.

Either the WinPath or PowerSpan II arbiter can be used effectively in any application. However, when considering system level performance the use of the WinPath arbiter is recommended because of its flexibility in allocating bus bandwidth between internal WinPath resources and external resources.

1.4.1 WinPath as Host (60x) Bus Arbiter

The WinPath arbiter controls up to two external masters.

Table 3 shows the required signal connection between the WinPath and PowerSpan II devices when the WinPath is the arbiter in the system.

Table 3: Signal Connection When WinPath is the Bus Arbiter

PowerSpan II Arbitration Signal	WinPath Arbitration Signal	
	PowerSpan II as Master One[a]	PowerSpan II as Master Two[a]
PB_BR[1]_ This signal is an output when the PowerSpan II 60x bus arbiter is disabled.	H_ARB_IN_ This signal is an input when WinPath is the arbiter.	H_ABR2_ This signal is an input when WinPath is the arbiter.
PB_BG[1]_ This signal is an input when the PowerSpan II 60x bus arbiter is disabled.	H_ARB_OUT_ This signal is an output when WinPath is the arbiter.	H_ABG2_ This signal is an output when WinPath is the arbiter.
PB_DBG[1]_ This signal is an input when the PowerSpan II 60x bus arbiter is disabled.	H_ARB_DBG_ This signal is an output when WinPath is the arbiter.	H_DBG2_ This signal is an output when WinPath is the arbiter.

a. The WinPath arbiter controls up to two external masters. When PowerSpan II is the first external bus master (master one) it connects through the following WinPath signals: H_ARB_IN_, H_ARB_OUT_, H_ARB_DBG_. When PowerSpan II is the second external bus master (master two) it connects through the following WinPath signals: H_ABR2_, H_ABG2_, H_DBG2_. The WinPath H_DATA_BUSY_ signal connects to PB_DBB_ signal of PowerSpan II.

1.4.2 Configuring PowerSpan II as Host (60x) Bus Arbiter

The PowerSpan II host (60x) bus arbiter is enabled through a power-up option. The Arbiter Enable (PB_ARB_EN) bit (read-only) in the PowerSpan II Reset Control and Status register reflects this option. The bit is set to 1 during power-up when PowerSpan II internal arbiter is enabled.

When PowerSpan II is sharing the host bus with two or more WinPath processors, the WinPath's host bus arbiter must be used.

The following functions can be programmed in the Arbiter Control Register:

- Bus Parking
- External Master Enable
- Priority Level

Table 4 shows the point-to-point signal connection required between the WinPath PowerSpan II when the PowerSpan II is the arbiter in the system.

Table 4: Signal Connection when PowerSpan II is the Bus Arbiter

PowerSpan II Arbitration Signal	WinPath Arbitration Signal
PB_BR[*n*]_[a] This signal is an input when PowerSpan II is the arbiter.	H_ARB_OUT_ This signal is an output when the WinPath arbiter is disabled.
PB_BG[*n*]_ This signal is an output when PowerSpan II is the arbiter.	H_ARB_IN_ This signal is an input when the WinPath arbiter is disabled.
PB_DBG[*n*]_ This signal is an output when PowerSpan II is the arbiter.	H_ARB_DBG_ This signal is an input when the WinPath arbiter is disabled.

a. PowerSpan II's internal 60x bus arbiter can support up to three external bus masters. The value of *n* can equal 1,2, or 3 depending on the arbitration signal connection.

1.5 Bus Pipelining Support

In joint WinPath and PowerSpan II applications bus pipelining mode is supported but must be used under certain constraints.

1.5.1 Programming Requirements

The pipeline depth field (PDEPTH) in the WinPath External Bus Arbiter Control register defines how many address tenure grants are generated by the arbiter relative to data tenure. The following list shows the programmable values in the PDEPTH field:

- 001: One address grant is generated before one data phase is completed (No pipeline mode).
- 010: Two address grants may be asserted before one data phase is completed. (One-level pipeline mode).
- 000: Four address grants may be asserted before one data phase was completed.

PowerSpan II does not allow more than level-one address pipelining. Assuming a transaction (Transaction A) takes place on the host (60x) bus, the PowerSpan II allows the next transaction's address phase (Transaction B) to begin after the Address Phase of the Transaction A was acknowledged by the assertion of AACK. The PowerSpan II's pipeline allows the address phase of Transaction B to start before the data phase of Transaction A has finished. However, PowerSpan II's pipeline does not allow the address phase of Transaction B to be acknowledged before the data phase of Transaction A has finished.

This characteristic applies not only to the PowerSpan II itself, but for every device in the system, therefore the pipeline size is not only limited by the arbiter itself (for WinPath, the host (60x) bus arbiter can be configured to support three different levels of pipelining), but it is limited by all the slaves in the bus.

The WinPath Bus Arbiter can grant the bus to any master if the number of Address Phases waiting for their corresponding Data Phase does not exceed a specific number. The Multi-master Slave Module in the WinPath has an internal buffer of four locations that allows it to support of a bus pipeline of four.

In a WinPath and PowerSpan II system, if a pipeline depth of four is enabled within the WinPath, the host bus interface could hang.

To avoid a situation where the pipeline mode is set to a level PowerSpan II does not support, the WinPath processor includes a configuration bit that will force the Multi-master Slave Module to work with a pipeline of reduced depth. The RSPD (Reduced Slave Pipeline Depth) bit in the BIU_EBACR register enables or disables this capability. If this bit is 0, the WinPath Multi-master Slave Interface limits the maximum pipeline depth to four. If this bit is 1, the pipeline depth is limited to one.

1.6 Interrupt Control

The WinPath processor can both generate and process interrupt requests to and from PowerSpan II.

The PowerSpan II has six individual bidirectional interrupt pins INT[5:0] that can be connected to the WinPath or the PowerPC processor.

PowerSpan II's INT[5:1] signals are also the multiplexed input signals for PowerSpan II power-up configuration options (see "Power-up Options" on page 13).

If only one interrupt signal is required in the system, the INT[0] signal can be directly connected to aWinPath interrupt input with a pull-up resistor to the 3.3 V I/O power supply. The WinPath MIPs core can then process these interrupts which are routed through the internal interrupt controller.

Interrupt Routing is discussed in "Typical Applications" on page 28.

1.7 Memory Interface

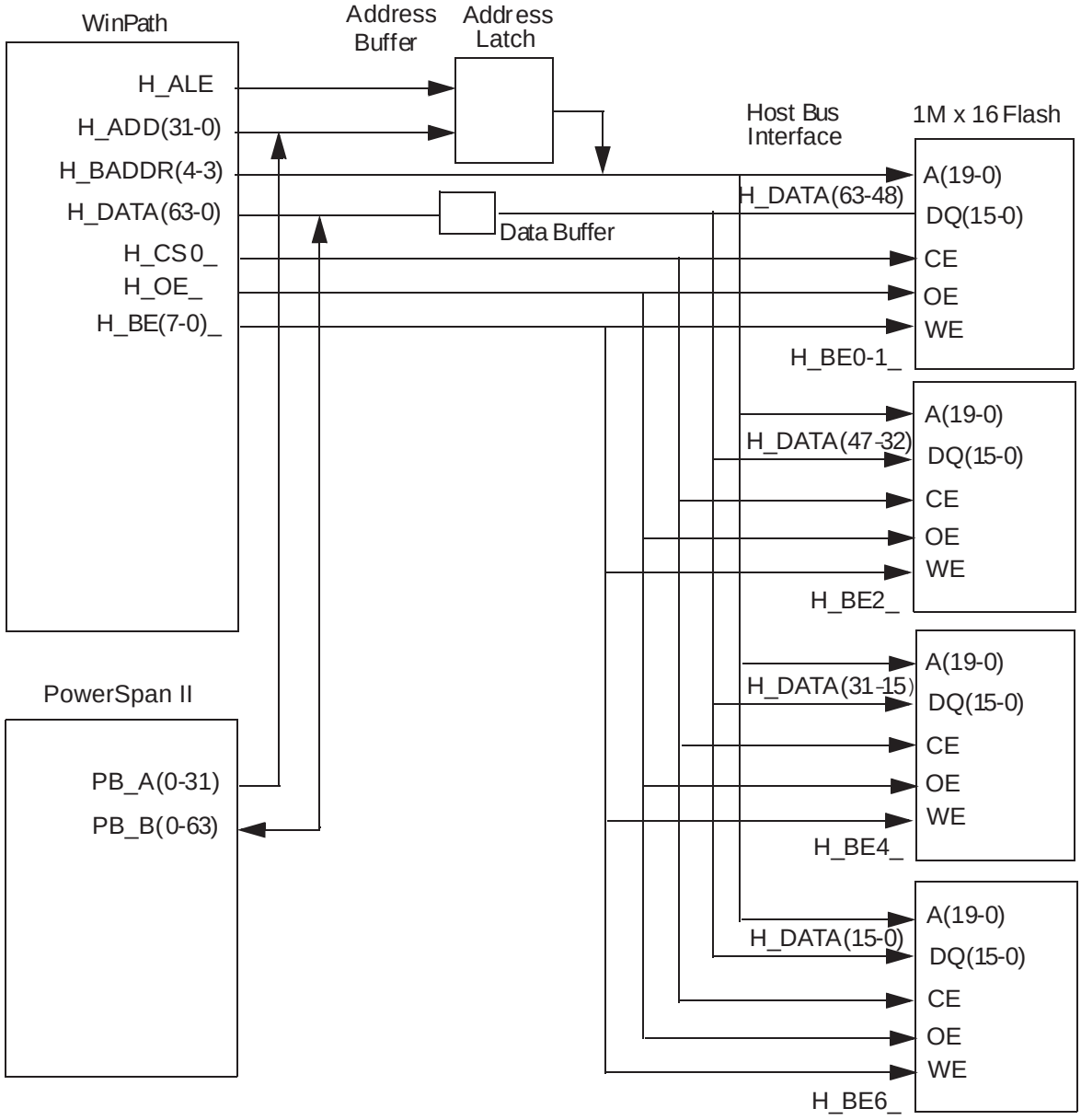
The memory implementation for a WinPath and PowerSpan II system can be controlled by the WinPath host bus memory controller. In addition to WinPath's ability to master the host bus, the PowerSpan II's internal DMA engine or PCI target image channel can be used for mastering the host bus to access both SDRAM and FLASH memory.

1.7.1 Flash Memory Interface

The Flash memory is controlled by the asynchronous memory controller. This memory is 64 bits wide to meet the requirements of the PowerPC device for boot port size and is controlled by host chip select 0 (H_CS0) the WinPath boot chip select. The reset configuration routine reprograms the chip select to the address area of the reset vector associated with the boot microprocessor. For MIPS based boot the address is 0x1FC00000, for PowerPC based boot the address is 0xFFF00100.

Figure 5 illustrates the connectivity of the memory interface.

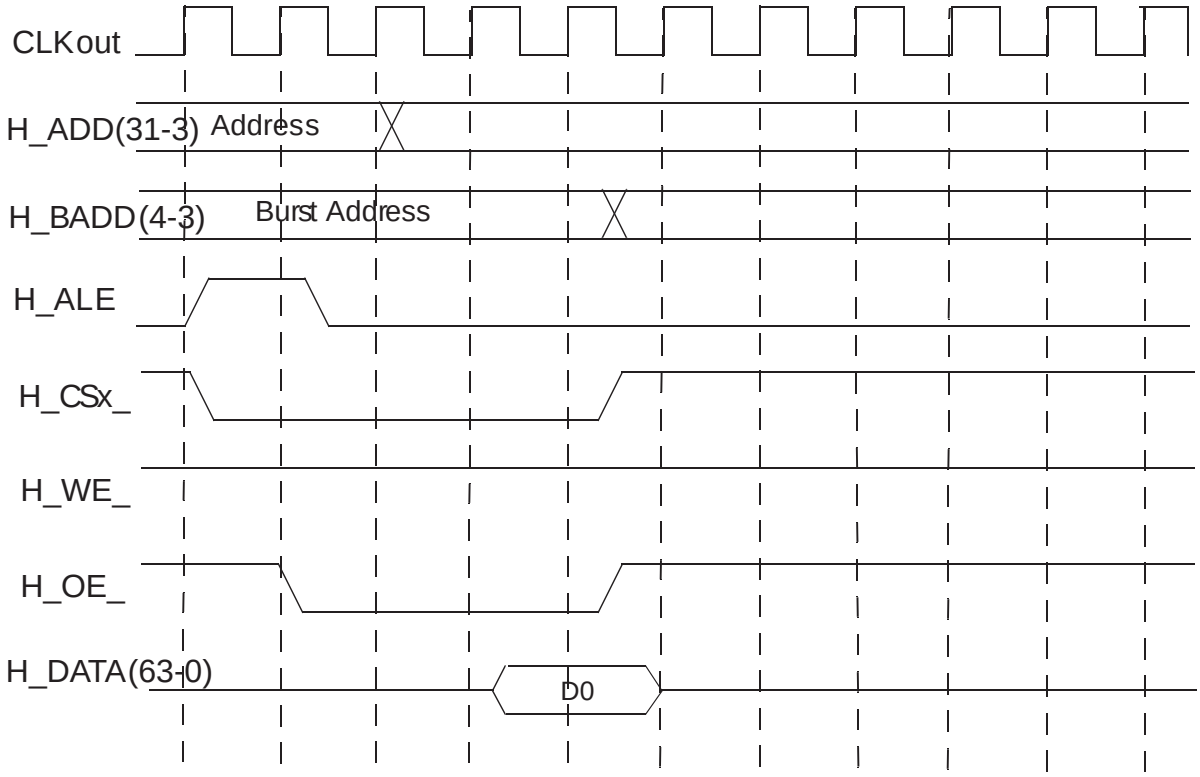
Figure 5: Flash Memory Interface



1.7.1.1 FLASH Read Cycle

Figure 6 illustrates the data phase of a single 64-bit read cycle initiated by either WinPath or the PowerSpan II device. The bus master drives the address until is it acknowledged. WinPath drives the burst address line and the ASRAM controller strobes the transaction by asserting the chip select (H_CSx) and output enable (H_OE) according to chip select programming. The transaction terminates and the data is latched by the data bus master.

Figure 6: FLASH Read Cycle



1.7.2 SDRAM Memory Interface

The SDRAM memory is controlled by the SDRAM memory controller. The memory is 64 bits wide taking advantage of the data bus width supported by WinPath and the PowerSpan II device.

The SDRAM interface requires external address multiplexing to support accesses by the PowerSpan II. WinPath includes an optimized interface for controlling latch and multiplexing devices. An integrated latch/mux device (for example, 74AVCLH16260) can be used to limit the delay through two devices.

Figure 7 illustrates the connectivity of this interface.

Figure 7: SDRAM Memory Interface

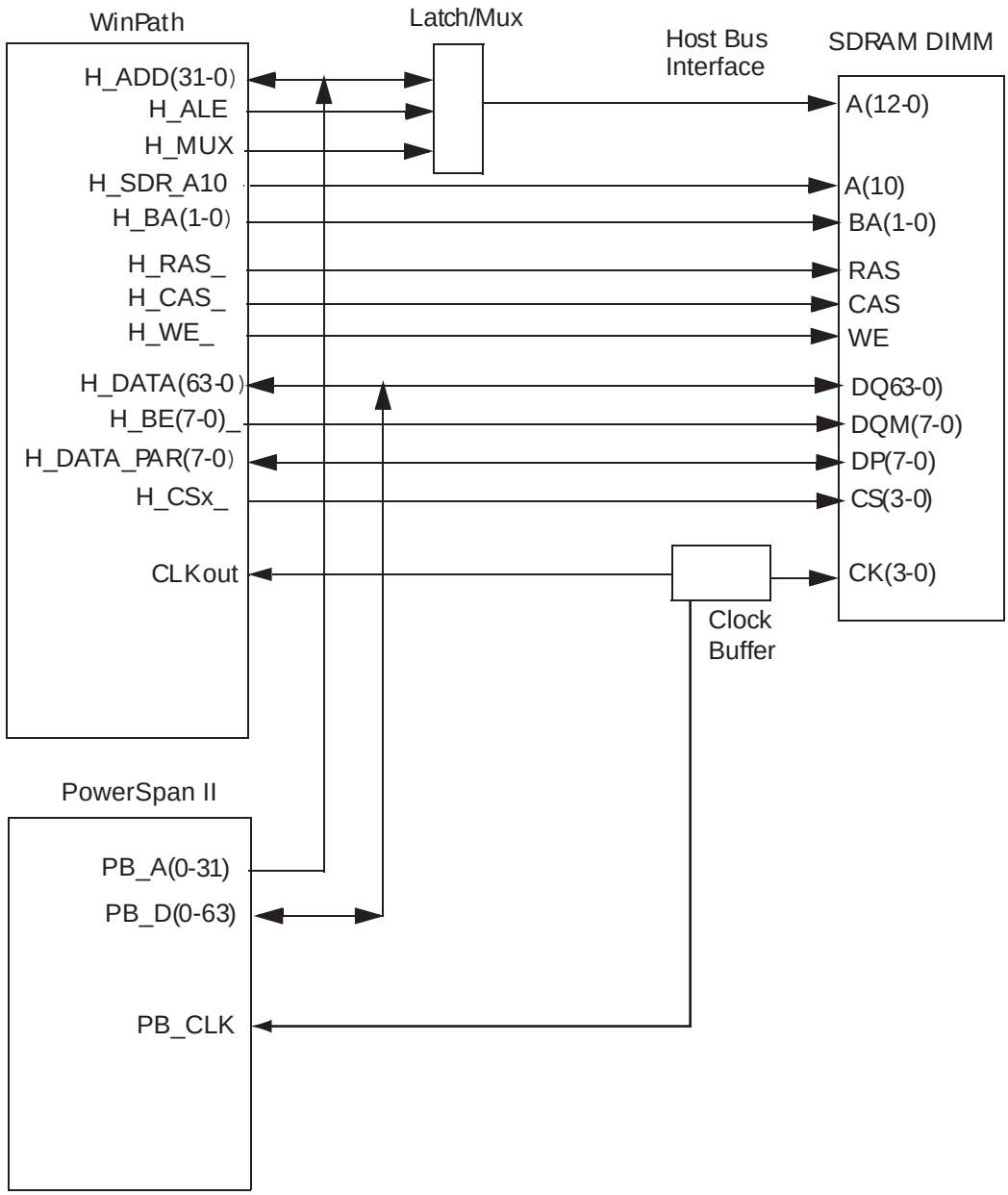
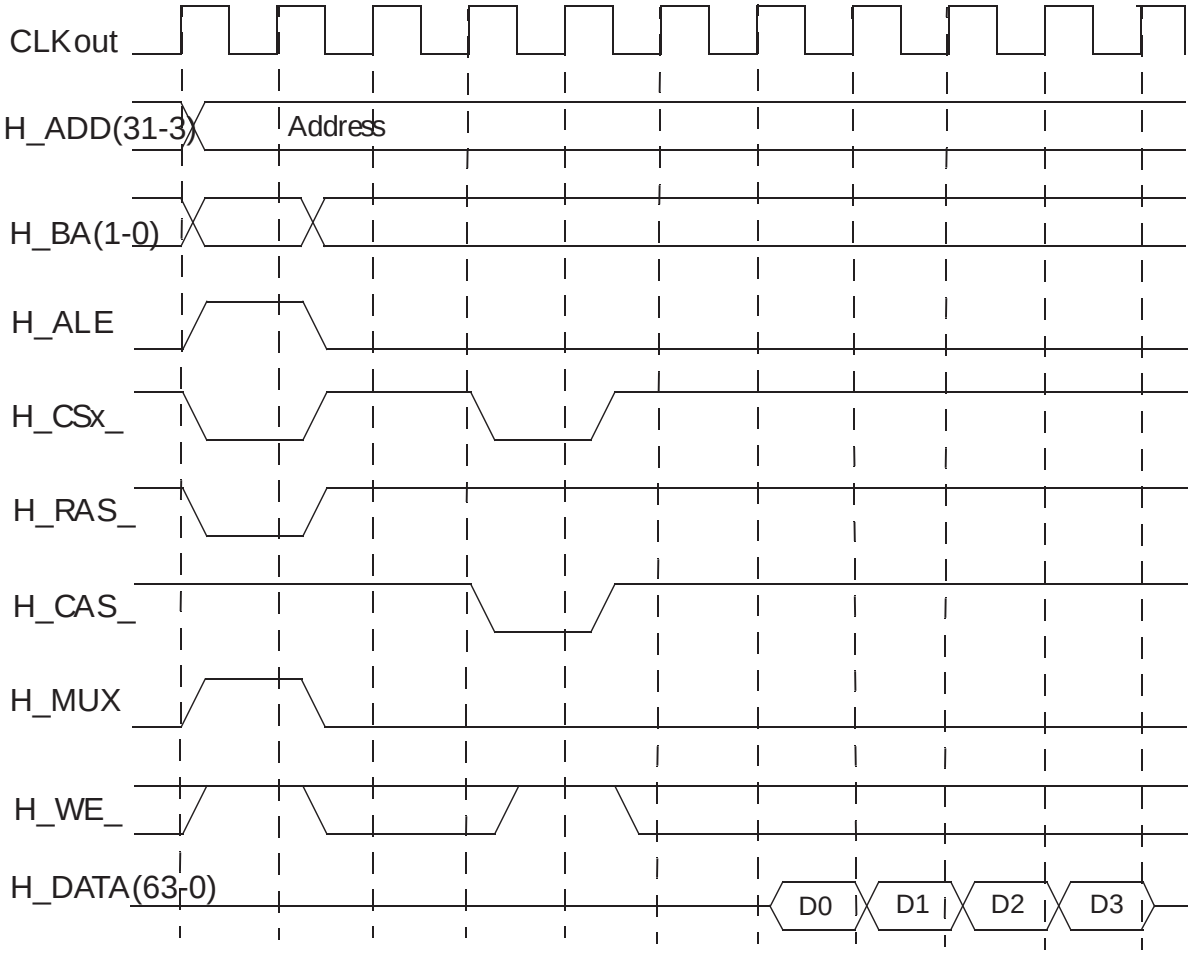


Figure 8 illustrates the timing relationships of the address latch and multiplex control.The address latch control signal (H_ALE) and multiplexor control signal (H_MUX) assert during the row address phase in the same clock cycle as the chip select (H_CSx) and row address strobe (H_RAS_) assertion. These signals de-assert in the following clock cycle. The latched address is multiplexed during the column address phase with the assertion of the chip select and column address strobe (H_CAS_).

Figure 8: SDRAM Cycle- Address Latch and Multiplex Control



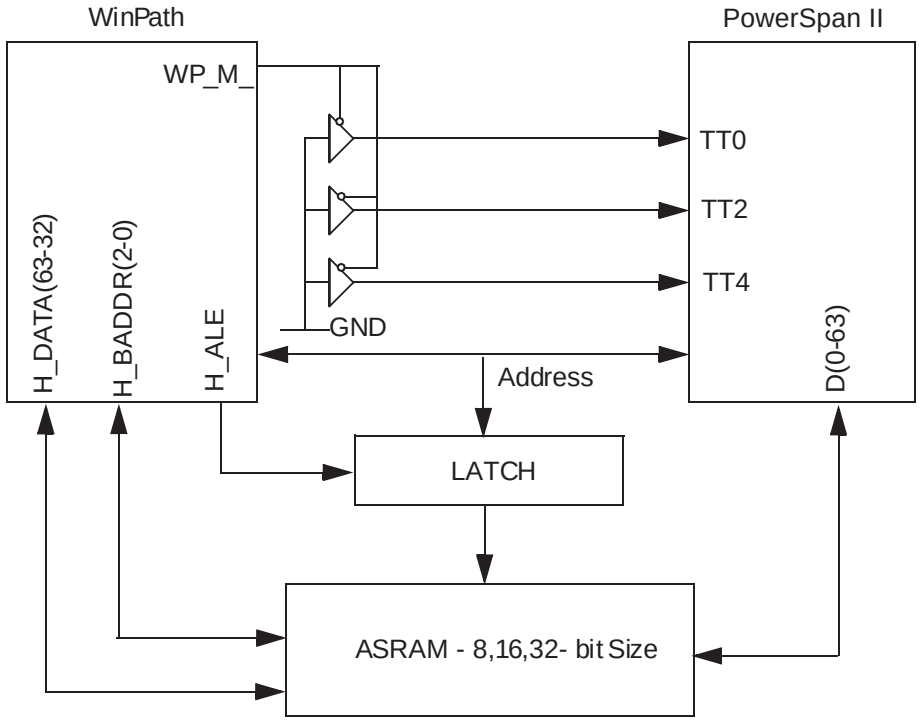
1.7.3 Application Supporting Smaller Asynchronous Memory Port Sizes

In an interface configuration with the PowerSpan II, a typical application is a PCI mezzanine card where PCB space is a premium. WinPath supports small port sizes on its asynchronous memory interface and this may be exploited when interfacing to Flash memory. The size of this memory can be tailored to meet the exact requirements for code storage. Since the code will be typically copied to faster memory e.g. SDRAM, for execution during runtime, the flash port size can be minimized since it has little effect on the system performance. The Host bus burst address pins H_BADDR(2-0) are required to support access decomposition to the smaller port size.

However due to pin multiplexing on the Host bus where the burst address pins 2,1,0 are multiplexed with 60x transfer type pins TT0, TT2, TT4 there is a requirement to implement logic to overcome the fact that transfer type pins become unavailable when burst addresses are selected. The transfer type signals must be driven with the appropriate logic level for proper operation. WinPath however only samples TT1,TT3, these signals convey all the information required for transactions supported by WinPath.

Figure 9 illustrates the recommended logic. The signal WP_M is asserted by WinPath indicating it is mastering the current transaction on the Host bus. The signal is used as the tristate enable to a set of buffers which drive the logic level (logic 0) to transfer type pins.

Figure 9: Small Port Size Implementation



The smaller asynchronous memory port sizes (8, 16. and 32-bit) are supported by WinPath on the host bus for SDRAM and ASRAM memory. These port sizes only apply if WinPath is the only device that can access the memory. When the memory can be accessed by another device, for example PowerPC processor or PowerSpan II, the memory port size must be 64-bit.

1.7.4 Multi-master Operation - Shared Data

The WinPath and PowerSpan II devices operate together in a multi-master configuration. However, global data between the processors cannot be shared as cacheable.

The WinPath processor does not implement cache snooping of external masters and therefore all shared data must be non-cached. The WinPath drives the host (60x) bus global signal (GBL_) high for all initiated transactions; it never samples this signal. The WinPath Memory Management Units (MMU) must be used to create an area of non-cached memory to allow data to be shared between the WinPath and the PowerSpan II.

The PowerSpan II PB_GBL_ is an output only signal. PowerSpan II never samples this signal. Whether PB_GBL_ is asserted during the transaction can be controlled by the PB_GBL_ bit in the control register.

Refer to the *PowerSpan II User Manual* for more information.

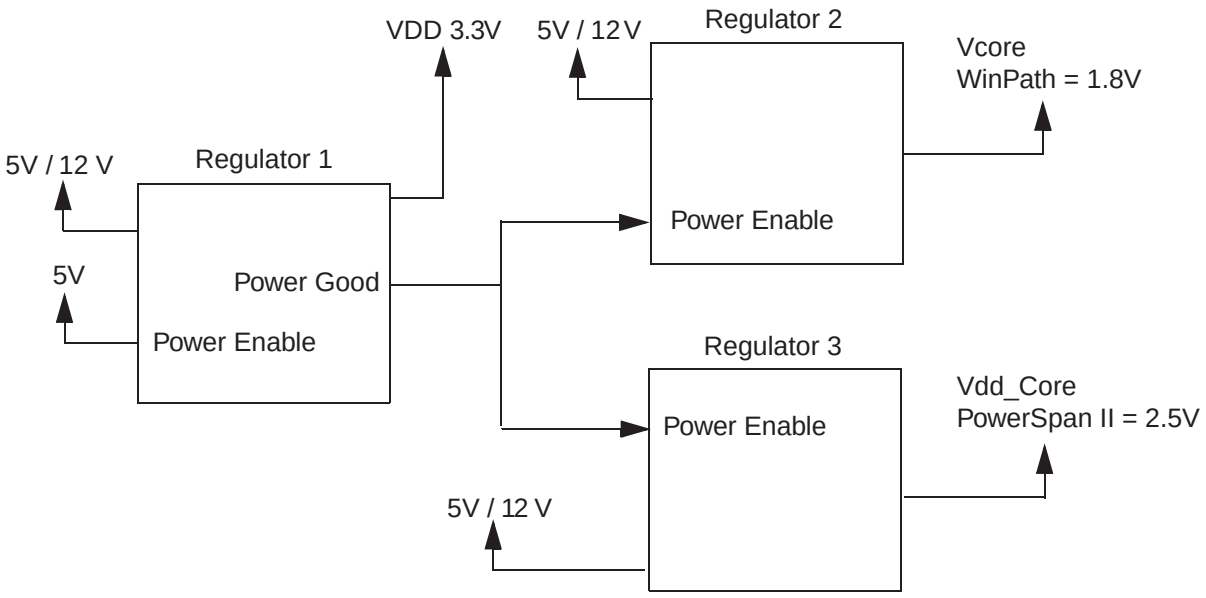
1.8 Power Supplies

When designing a WinPath and PowerSpan II system, there are different power supply requirements for each device. Both of these devices use the same 3.3 V I/O power voltage. However, WinPath specifies a 1.8 V core supply while PowerSpan II's core voltage is 2.5 V.

Flexible power distribution can be implemented with programmable regulators providing voltage selection via sets of input configuration pins. The regulators are used to generate the core and I/O voltages and provide the sequencing of the power supply. The sequencing is important to meet specifications of each device.

Figure 10 illustrates the architecture of an implementation for a WinPath and PowerSpan II system. Each regulator has 5V and 12 volt supply.

Figure 10: Power Supply Regulation Circuit



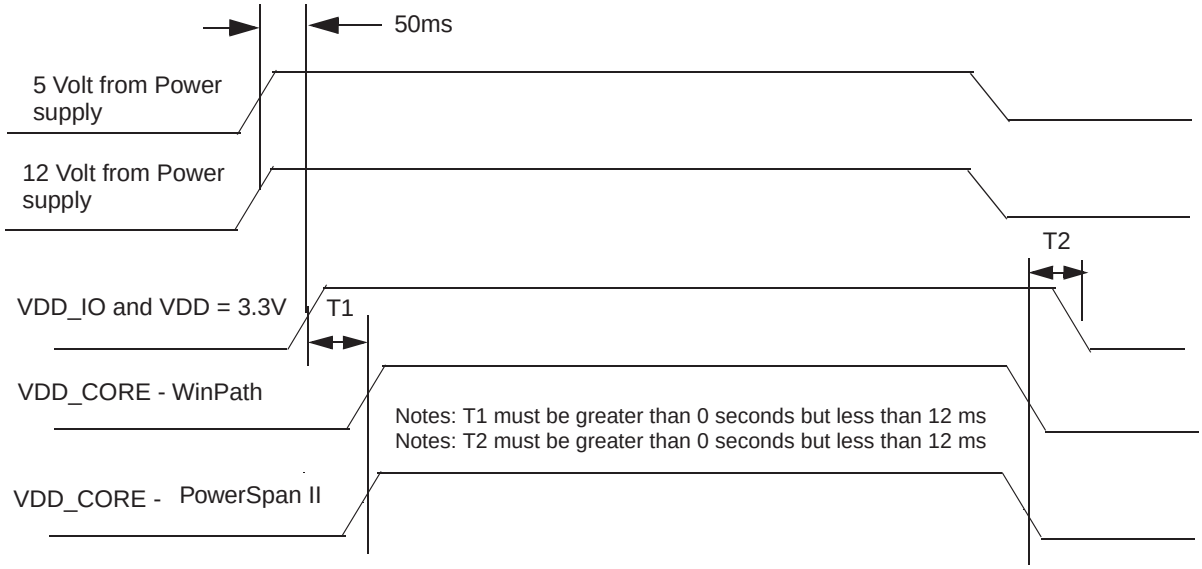
Regulator 1 generates the I/O voltage for both devices as well as the general 3.3 V supply. Regulator 1 has an output pin (Power Good) indicating power has reached it nominal value. This signal enables Regulator 2 and Regulator 3, which then generate their output voltages. During power down phase, as the 3.3 V supply drops the Power Good signal is negated. The negation of the Power Good signal turns off Regulator 2 and Regulator 3. This circuit ensures that power sequencing is adhered to for each supply.

Refer to the WinPath and PowerSpan II documentation to determine the required current rating for each regulator.

1.8.1 Power Sequencing

The voltages are supplied to the device sub-modules in sequence. The power supply ensures that VDD_IO rises to full voltage before VDD_CORE is applied. For power-down VDD_CORE should fall before VDD_IO.

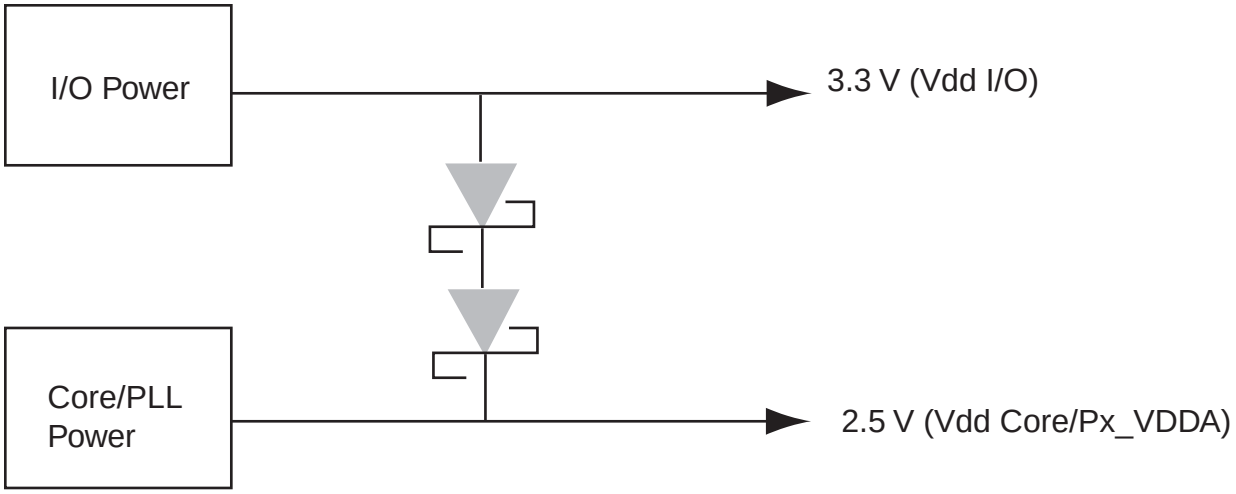
Figure 11: Power Supply Sequence



1.8.2 PowerSpan II Recommended Bootstrap Diode

IDT recommends the use of a bootstrap diode between the power rails. The bootstrap diodes that are used in the system must be configured so that a nominal Core Supply Voltage (Vdd Core) is sourced from the I/O Supply Voltage (Vdd I/O) until the power supply is active. In Figure 12, two Schottky barrier diodes are connected in series. Each of the diodes has a forward voltage (V_F) of 0.6V at high currents which provides a 1.2V current drop. This drop maintains 2.1V on the 2.5V power line. Once the Core/PLL power supply stabilizes at 2.5V, the bootstrap diode(s) are reverse biased with small leakage current.

Figure 12: Bootstrap Diodes for Power-up Sequencing



The forward voltage must be effective at the current levels required by PowerSpan II (< 1 amp). Do not use diodes with only a nominal V_F which drops too low at high current.

1.9 Typical Applications

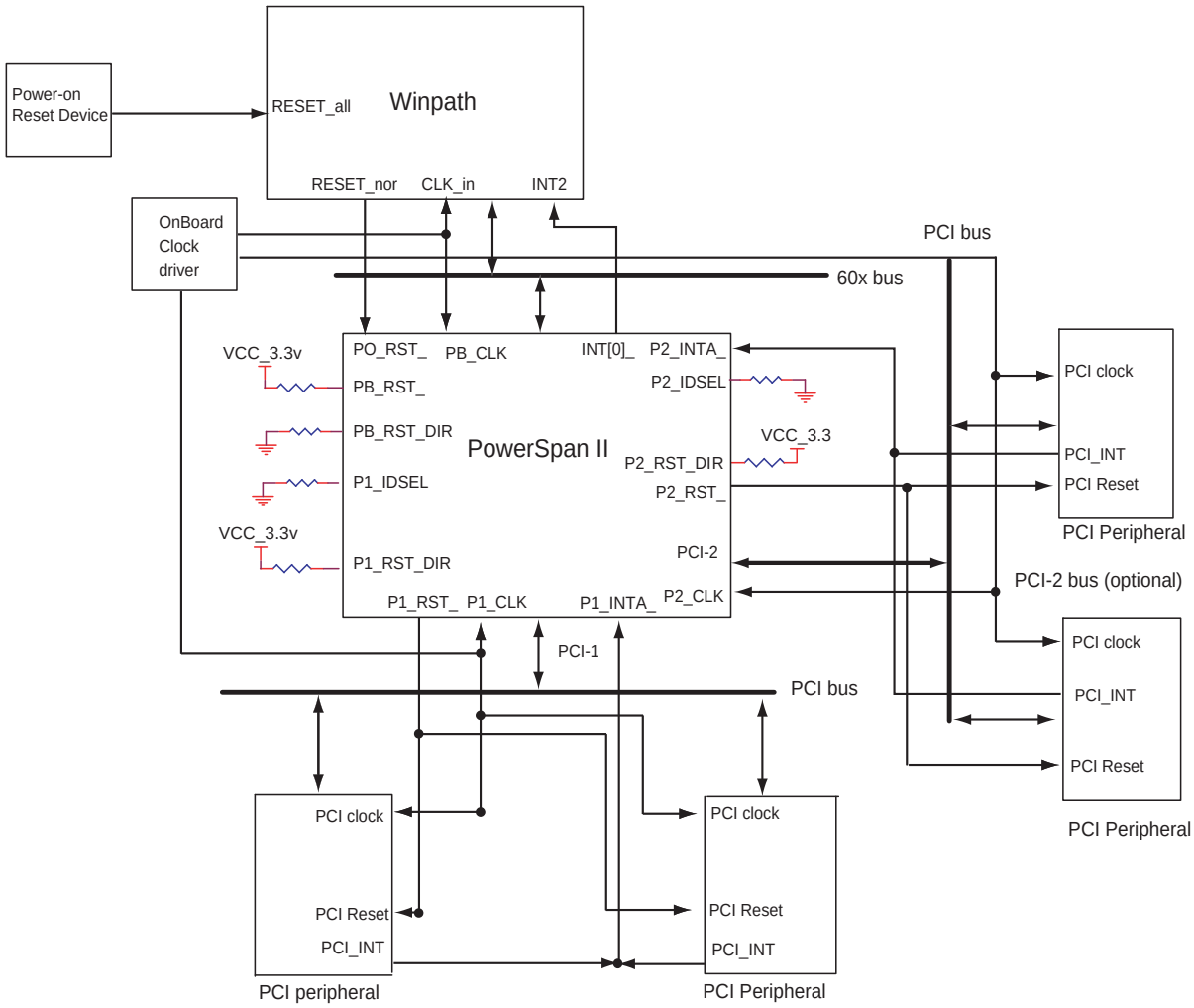
PowerSpan II is able to support the WinPath processor as both a PCI host bridge and a PCI peripheral bridge. Section 1.2 on page 6 described the connections between WinPath's host bus interface and PowerSpan II's processor bus interface. The following section discusses in greater detail how the clocks, resets, and interrupt signals should be connected based on the application.

1.9.1 WinPath as the Host Processor in the System

Figure 13 illustrates an typical application where the WinPath processor is using PowerSpan II as a host bridge. In this design, the internal MIPs core is enabled within the WinPath processor.

In this application there is an on-board clock generator to distribute all the necessary clocks to the WinPath, PowerSpan II, and memory devices (including the PowerSpan II PCI clocks).

Figure 13: WinPath as the Host Processor



The routing of the interrupts is dependent on the application. This diagram shows the interrupts being routed from the PCI Peripheral to the WinPath through PowerSpan II. The interrupt input pin to the WinPath processor can occur on any one of four maskable interrupts pins: INT1, INT2, INT3, or INT4.

The PowerSpan II also drives the PCI reset signal and the REQ64_ signal during the power reset sequences. This is a host feature according to the *PCI Local Bus Specification*. When the WinPath processor asserts the hard reset (RESET_nor) signal, which is connected to PowerSpan II's PB_RST_, PowerSpan II drives the P1_RST_ and the P2_RST_ signal low.

Make sure the PowerSpan II power-up option pins are driven to the correct value during each assertion of its PO_RST_ caused by the WinPath's RESET_nor signal.

During the board initialization, the PowerSpan II is initialized by the MIPS processor core within the WinPath device.

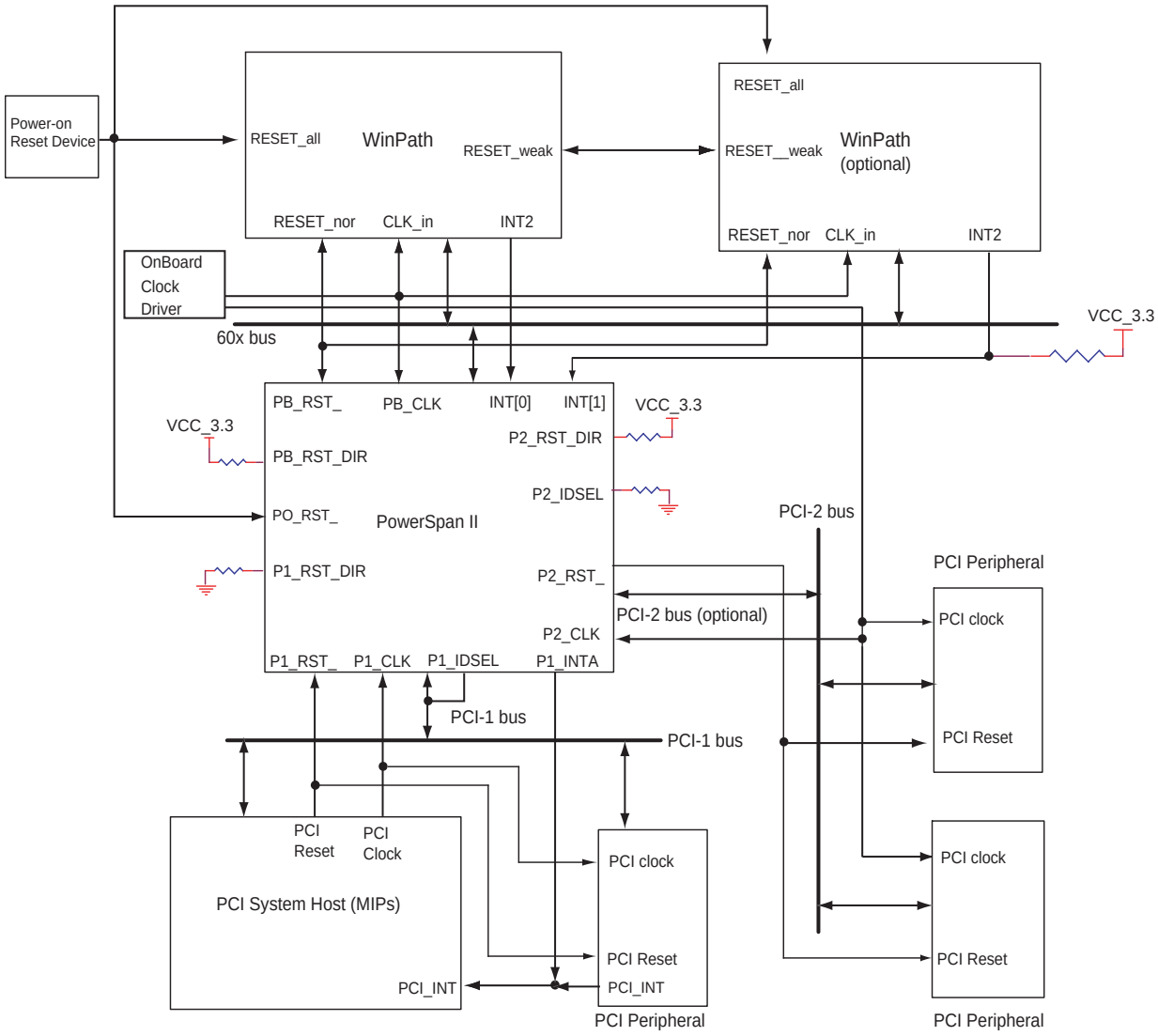
In order to configure PowerSpan II in this application, the following design considerations must be reviewed:

1. Pull-up the P1_RST_DIR pin to allow PowerSpan II to drive PCI reset from its P1_RST_ pin.
2. Pull-down the PB_RST_DIR pin and pull-up the PB_RST_ signal because WinPath drives the host bus reset to PowerSpan II's PO_RST_ signal.
3. Pull-down the P1_IDSEL signal to ground to disable PowerSpan II from receiving PCI configuration cycle. This is required because PowerSpan II is the PCI host bridge device in this application.
4. Select power-up option (PWRUP_BOOT) as PB_BOOT to allow PowerSpan II to be initialized from the host bus.
   - This option can be implemented either by pulling-up the INT[3] signal or, alternatively, using a CPLD to drive it to a logic high during the negation of PO_RST_.
5. Enable power-up option PWRUP_P1_R64_EN to allow PowerSpan II to drive P1_REQ64_when it drives PCI reset out.
   - This option programs whether PowerSpan II drives the P1_REQ64_ during the negation of P1_RST_. This is a PCI host feature in 64-bit mode according to the *PCI Local Bus Specification*.
   - Enable PWRUP_P1_R64_EN either by pulling-down the INT[4] signal or, alternatively, the signal can be driven by any CPLD to ensure its at a logic low during the negation of PO_RST_.
6. Optionally, configure the interrupt pin INT[0] as a output and P1_INTA as a input to route the hardware interrupt request from the PCI peripheral devices to the WinPath host processor.
   - The interrupt pins direction can be individually configured by the PowerSpan II register IDR (offset 0x444) through a host software access after power-up configuration.

1.9.2 WinPath Application with the Host Processor Residing on the PCI bus

Figure 14 illustrates a typical application where the host processor in the system resides on the PCI bus. In this design, PowerSpan II is used as a PCI bridge to communicate between the WinPath devices and the host processor.

Figure 14: Host Processor on the PCI Bus and Dual WinPath Devices



The routing of the interrupts is dependent on the application. This diagram shows the interrupts being routed from the WinPath to the PCI System Host through PowerSpan II. The interrupt input pin to the WinPath processor can occur on any one of four maskable interrupts pins: INT1, INT2, INT3, or INT4.

In this application an on-board clock generator distributes the necessary clocks to the WinPath, PowerSpan II, and memory devices. The only clock not supplied by the clock generator is PowerSpan II's P1_CLK.

According to the *PCI Specification*, the PCI host system is responsible to provide the clock and reset signals for the devices on the primary PCI interface.

The PCI host system initializes the WinPath and PowerSpan II peripheral device after power-up.

When the PCI host asserts the bus reset, PowerSpan II monitors the P1_RST_ signal and propagates the signal to its processor bus by asserting its PB_RST_ signal (this signal is connected to WinPath's RESET_nor signal) and P2_RST_ signal (when the Dual PCI PowerSpan II device it used). This reset configuration scheme allows the WinPath and PowerSpan II peripheral card to be reset by the PCI host.

In order to configure PowerSpan II in this application, the following design considerations must be reviewed:

1. Pull-down P1_RST_DIR pin to allow PowerSpan II to propagate PCI reset from its P1_RST_ pin.
2. Pull-up PB_RST_DIR pin to allow PowerSpan II to drive PB_RST_ to reset WinPath.

The PowerSpan II's PB_RST_ signal is a bidirectional signal. Even when PB_RST_ is configured as a output, the PowerSpan II still monitors PB_RST_ and reset its PB interface accordingly to the appropriate status when the hard reset is asserted from external host bus processor. For example, when WinPath loads the boot code from FLASH memory it drives the H_ADDR_START_ signal and the H_ADDR_ACK_ signal while asserting the RESET_nor signal. PowerSpan II's PB_RST_ signal must be connected to RESET_nor to make sure its PB interface is kept in reset during the loading period.

3. Connect the P1_IDSEL signal to one of the PCI-1 bus address lines (P1_AD[31:16]) to allow PowerSpan II to receive PCI configuration cycle (Type 0) issued by the PCI host device. Review the manual of the PCI bridge device on host side of the application to make sure the proper AD bus line is used for generating PCI configuration cycle.
4. Select the power-up option (PWRUP_BOOT) as PCI_BOOT to enable PowerSpan II to be initialized first from the PCI bus.
   - This option can be implemented by pulling-down the INT[3] signal or using a CPLD to drive it to a logic low during the negation of PO_RST_.

5. Disable the power-up option PWRUP_P1_R64_EN to allow PowerSpan II to sample P1_REQ64_ signal during the negation of PCI reset.

   — This option supports PowerSpan II driving the P1_REQ64_ signal during the negation of P1_RST_. This is a PCI host feature, according to the *PCI Local Bus Specification*.

   — In the application shown in Figure 14, this option must be disabled by pulling-up the INT[4] signal, or alternatively using a CPLD to drive it to logic high during the negation of PO_RST_.

6. Optionally, configure the INT[0] interrupt pin as an input and P1_INTA interrupt pin as output. This routes the hardware interrupt from WinPath to the PCI host.

   This application uses both the PowerSpan II's INT[0] and INT[1] signals as inputs and the P1_INTA signal as an output to route the hardware interrupt from the two WinPath devices to the PCI host processor. Please note, the INT[1] signal is also used for the multiplexed system pin mode for the power-up option, which is sampled during the negation of PO_RST_. Therefore, the design must ensure INT[1] is at a logic high during the negation of PO_RST_. The INT[1] pin must not be driven by any device attached to this interrupt signal during the PowerSpan II power-up period.

   ⚠ On the PowerSpan II the multiplexed system pin mode power-up option, INT[5:1] has a 10 ns the setup time before the negation of PO_RST_ and a 10 ns hold time after the negation of PO_RST_.

   The direction of the interrupt pins can be individually configured by the PowerSpan II register IDR (offset 0x444). They can be configured through a host software access after power-up configuration.

   For a complete design reference, refer to the *PowerSpan II User Manual* and the WinPath *Hardware Developer's Guide*.

1.9.2.1 Alternative Reset Strategy

For a design which requires the PCI Host to completely reset WinPath, one solution is to route ta PowerSpan II general purpose interrupt pin (one of the INT[5:0] signals), and the WinPath's RESET_all signal to a CPLD device. Inside the CPLD, the interrupt output status is latched to drive the WinPath RESET_all signal.

The PCI Host can access either PowerSpan II's mailbox or doorbell register which is mapped to one of the interrupt signals to completely reset the WinPath processor through the CPLD.

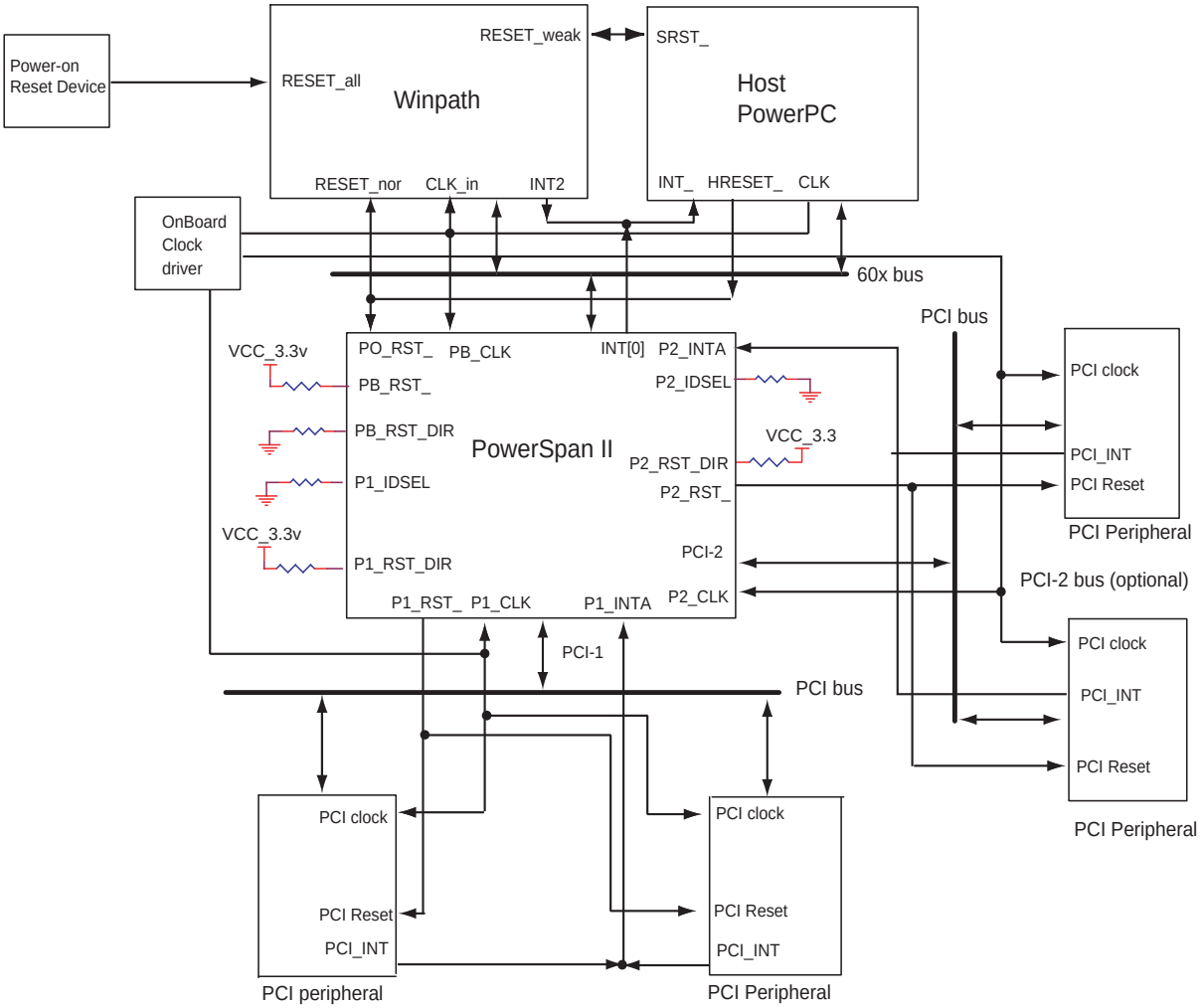
The CPLD must latch the interrupt signal status long enough to reset WinPath. This is necessary because the PowerSpan II tristates the interrupt output when its PB_RST_ signal is asserted by WinPath during power-up.

Another recommended solution to create a flexible system reset strategy is to route all of PowerSpan II's and WinPath's reset signals to an external CPLD. When using this CPLD solution, the board designer has more control of the reset sequence while prototyping the board.

1.9.3 WinPath Application with a PowerPC Host Processor

Figure 15 illustrates an application where the internal MIPS core within the WinPath device is disabled and the PowerPC processor is the host in the system. The PowerSpan II device is providing PCI host bridging capability to the I/O peripherals.

Figure 15: WinPath with PowerPC in a Host Application



The routing of the interrupts is dependent on the application. This diagram shows the interrupts being routed from the PCI Peripheral to the Host PowerPC through PowerSpan II. The interrupt input pin to the WinPath processor can occur on any one of four maskable interrupts pins: INT1, INT2, INT3, or INT4.

In this application, there is an on-board clock generator to distribute all the necessary clocks to the WinPath, PowerSpan II, and memory devices.

The PowerSpan II drives the PCI reset signal and the REQ64_ signal during the power reset sequence. This functionality is a host feature according to the *PCI Local Bus Specification.* When the PowerPC processor asserts the hard reset (HRESET_) signal, which is connected to PowerSpan II's PO_RST_ signal, PowerSpan II drives the P1_RST_ and the P2_RST_ signal low accordingly.

During the board initialization, both the WinPath and the PowerSpan II are initialized from the PowerPC host processor through their host bus interfaces.

In order to configure PowerSpan II in this application, the following design considerations must be reviewed:

1. Pull-up the P1_RST_DIR pin to allow PowerSpan II to drive PCI reset from its P1_RST_ pin.
2. Pull-down the PB_RST_DIR pin and pull-up PB_RST_ signal because the PowerPC processor and WinPath drives the host bus reset to PowerSpan II's PO_RST_ signal.
3. Pull-down the P1_IDSEL signal to ground to disable PowerSpan II from receiving PCI configuration cycle. This is required because PowerSpan II is the PCI host bridge device in this application.
4. Select power-up option (PWRUP_BOOT) as PB_BOOT to allow PowerSpan II to be initialized from the host bus.
   - This option can be implemented by pulling-up the INT[3] signal or, alternatively, using a CPLD to drive it to a logic high during the negation of PO_RST_.
5. Enable power-up option PWRUP_P1_R64_EN to allow PowerSpan II to drive P1_REQ64_when it drives PCI reset out.
   - This option programs whether PowerSpan II drives the P1_REQ64_ during the negation of P1_RST_. This is a PCI host feature in 64-bit mode according to the *PCI Local Bus Specification*.
   - The PWRUP_P1_R64_EN option can be implemented either by pulling-down the INT[4] signal, or using a CPLD to drive it to a logic low during the negation of PO_RST_.

6. Optionally, configure the interrupt pin INT[0] as a output and P1_INTA as a input to route the hardware interrupt request from the PCI peripheral devices to the PowerPC host processor.
   - These interrupt pins direction can be individually configured by the PowerSpan II register IDR (offset 0x444) through a host software access after power-up configuration.

1.10 Register Configuration

In a system that has both the WinPath and PowerSpan II, particular registers in both devices must be set to specific values in order to ensure proper operation. The information in this application note is based on the testing of the WinPath and PowerSpan II configurations. The system can still work with other register settings, but the values represented in Table 5 and Table 6 are derived from extensive testing.

1.10.1 WinPath Register Settings

Table 5 shows which registers in the WinPath processor must be set to the corresponding value in order for a WinPath and PowerSpan II system to function properly.

Table 5: WinPath Required Register Setting for PowerSpan II Applications

WinPath Host Bus (SIU4) Register	Register Description	Required Bit	Bit Description	Bit Setting for WinPath
BIU_CONF (SIU4)	BIU General Configuration Register	Multi-Master (MM)	Indicates if the bus is multi-master bus or a stand alone bus. 0= stand alone bus 1= Multi master bus (default)	MM=1
BIU_PCR (SIU4)	BIU Pin Configuration register	Active Pull up Enable (ACTPU)	0= Host bus signal is tristated normally (default) 1= host bus signal is actively driven before tristating	ACTPU= 1
BIU_EBACR (SIU4)	BIU External Bus Arbiter Control Register	Setup Time Control (SUTC)	The field define the setup time options for Host bus signal H_DACK, H_ADDR_RETRY, H_BUS_ERROR, H_ADDR_ACK, H_ADDR_BUSY, H_ADDR_START, H_ARB_IN, H_DATA_ACK. 0= Standard setup time. No additional clock cycles added for memory control signals assertion (default) 1= Reduced setup time. Additional clock cycle added for memory control signal assertion	Set SUTC=1 when one of the following conditions is met: • Working with non-registered SDRAM on the Host bus. • Host bus frequency is above 66 MHz. Set SUTC=0 when neither of these conditions are met.
BIU_EBACR (SIU4)	BIU External Bus Arbiter Control Register	Retry Disable (RTD)	Defines if ARTRY supported signal active in the system. Disabling ARTRY allows a master to initiate back to back transactions without disconnecting from the bus for one clock cycle. 0=ARTRY supported (default) 1=ARTRY not supported	RTD=0

Table 5: WinPath Required Register Setting for PowerSpan II Applications

WinPath Host Bus (SIU4) Register	Register Description	Required Bit	Bit Description	Bit Setting for WinPath
BIU_EBACR (SIU4)	BIU External Bus Arbiter Control Register	Data Bus Busy internally generated (DBBI)	Defines if an external DBB signal is used in the system. In the case it is not used, DBB is internally generated so that it is asserted in the cycle immediately following the assertion of DBG and negated in the cycle immediately following the assertion of the last TA of the current transaction. 0=DBB external (default) 1=DBB internally generated	DBBI=0
BIU_EBACR (SIU4)	BIU External Bus Arbiter Control Register	Address Bus Busy Internally generated. (ABBI)	Defines if an external ABB signal is used in the system. In the case it is not used, ABB is internally generated such that it is asserted in the cycle immediately following the assertion of ABG and negated in the cycle immediately following the assertion of AACK of the current transaction. 0=ABB external (default) 1=ABB internally generated	ABBI=0
BIU_EBACR (SIU4)	BIU External Bus Arbiter Control Register	DBGTS	When the pipeline is empty, the WinPath (acting as the arbiter) can normally assert H_DBG_ in the same cycle as H_ADDR_START_ (TS_) 0 = H_DBG_ can be asserted in the same cycle as H_ADDR_START_ (TS_) 1= H_DBG_ can be asserted at least one cycle after H_ADDR_START_ (TS_). In this mode, H_DATA_BUSY_ must be deasserted in the cycle immediately following the last data phase of the transaction. H_DATA_BUSY_ may not be extended to provide additional turn-off time on the bus.	DBGTS = 1 Note: This setting is only valid for Rev. B0 and later revisions

Table 5: WinPath Required Register Setting for PowerSpan II Applications

WinPath Host Bus (SIU4) Register	Register Description	Required Bit	Bit Description	Bit Setting for WinPath
BIU_EBACR (SIU4)	BIU External Bus Arbiter Control Register	Pipeline Depth (PDEPTH)	Defines the maximum pipeline depth allowed in the system. This is limited by the system mechanism in each 60x device that tracks correspondence of the address-phase to data-phase. This defines for the internal arbiter when an additional ABG can be asserted without overflowing the number of address-phases initiated that have not yet completed their corresponding data phases. 001 = 1 transaction pending allowed 010 = 2 transactions pending allowed 011 = Reserved 000 = 4 transactions pending allowed (default) 1xx = Reserved	PDEPTH = 000 Note: The RSPD bit in the BIU External Bus Arbiter Control register must be set to 1 when the PDEPTH field is set to 000.
BIU_EBACR (SIU4)	BIU External Bus Arbiter Control Register	Data Bus Grant transition from IDLE (DGI)	Defines how the Data Bus Grant is asserted by the Bus Arbiter when the bus transitions from the IDLE state (no current transaction running). 0=DBG will be asserted together with ABG (default) 1=DBG will be asserted one clock cycle after the assertion of the ABG line.	When WinPath arbiter is enabled DGI=1
BIU_EBACR (SIU4)	BIU External Bus Arbiter Control Register	Reduced Slave Pipeline Depth (RSPD)	0= the WinPath Multi-master Slave Interface will work with a maximum pipeline depth of four. (default) 1=the pipeline depth is limited to one.This adds compatibility for device that have a maximum address pipeline depth of 1, such as PowerSpan II. Caution: If the WinPath RSPD bit is not set to 1 and data pipelining is enabled in a PowerSpan II system, the host bus could hang.	RSPD=1 Note: The RSPD bit must be set to 1.
BIU_EBACR (SIU4)	BIU External Bus Arbiter Control Register	Arbitration Mode (ARM)	Defines the priority mechanism for masters requesting bus ownership. 0 = Strict Priority (default) 1 = Weighted Round Robin	When the WinPath arbiter is enabled ARM=1

Table 5: WinPath Required Register Setting for PowerSpan II Applications

WinPath Host Bus (SIU4) Register	Register Description	Required Bit	Bit Description	Bit Setting for WinPath
BIU_EBACR (SIU4)	BIU External Bus Arbiter Control Register	Internal Bus Arbiter Enable (IABE)	0 = Internal arbiter disabled 1 = Internal arbiter enabled Note: Either the WinPath or PowerSpan II internal arbiter can be used effectively in any application. However, when considering system level performance the use WinPath arbiter is recommended because of its flexibility in allocating bus bandwidth between internal WinPath resources and external resources.	IABE = 1
MC_B_x_MSK (Chip Select)	Address Mask Register	Turn-off Time on Read Operation (TOFF)	00 = 0 clock cycles 01 = 1 clock cycle 10 = 4 clock cycles 11 = 8 clock cycles	TOFF=00
MC_B_x_MSK (Chip Select)	Address Mask Register	Wait State (WS)	Number of additional cycles (Wait states) inserted during read or write cycle where external termination is not enabled. 00000 = 0 wait states 00001 = 1 wait states . . . 11111 = 46 wait states **Note:** The programming options for this field continue up to 46 wait states.	When working with ASRAM memory on the 60x bus the WS field must be set to a value greater, or equal to, 1 when the SUTC bit is set to 0. The WS field can be set to any value when the SUTC bit is is set to 1.

1.10.1.1 Special Register Considerations

SUTC Bit Setting

WinPath must have the SUTC bit set to 1 when PowerSpan II accesses non-registered SDRAM through the host bus. This requirement is caused because PowerSpan II does not function properly with minimum cycle 60x bus systems where the TA signal occurs in the cycle immediately following the TS signal. Setting the SUTC bit to 0 may cause WinPath to generate minimum 60x bus cycles.

TOFF Bit Setting

When working with PowerSpan II, any memory peripheral device placed on the host bus that is controlled by a WinPath chip select and configured as a ASRAM device must have a TOFF parameter in the MC_B_x_MSK register set to zero clock cycles (00). When a peripheral requires a non-zero turn-off time, the data buffers must be used to isolate the data bus connected to the peripheral. The buffers can then be turned off without requiring a positive value programmed into the TOFF field.

TS/DBG Interaction

In a system with both the WinPath and PowerSpan II devices, the host bus can hang under the following conditions:

- WinPath is the 60x bus arbiter with the pipeline depth set 1.
- PowerSpan II initiates two consecutive transactions to the 60x bus. Two consecutive transactions can be initiated in the following ways:
  - PowerSpan II is doing DMA from one bus to the other with sizes other than 8 bytes or 32 bytes
  - A PCI master is performing access to the 60x bus with size other than 8 bytes or 32 bytes

For the first case, assume a PCI master request 16 bytes to read from the 60x bus, since the 60x bus can transfer only up to 8 bytes (single) or 32 bytes (burst), PowerSpan II will initiate two consecutive single transactions.

For the second case, assume a PCI master request 8 bytes to write from address 0x1 (not align to 8 bytes), the PowerSpan II will initiate two consecutive single transactions, first from address 0x1 with size of 7 and the second from address 0x8 with size of 1.

In the above two examples there is no difference if the initiator is the PCI master or the PowerSpan II's DMA.

Workaround (All WinPath revisions)

In order to avoid these situations the following actions can be taken:

- Ensure the PCI master (or PowerSpan II's DMA) is performing only 8 byte single or 32 byte burst transactions. The transactions starting address must be aligned to the address boundary.
- Enable WinPath's internal arbiter and configure it to pipe 0 (set PDEPTH field in BIU_EBACR register to '001').
- Use the PowerSpan II's internal 60x bus arbiter instead of the internal WinPath arbiter.

Workaround (WinPath Revision B.0 and Later Revisions)

In order to avoid these situations the following actions can be taken:

- When using the WinPath's internal arbiter, set the DBGTS bit in the BIU_EBACR register (see Table 5).

1.10.2 PowerSpan II Register Settings

Table 6 shows what registers in PowerSpan II must be set to the corresponding value in order for a WinPath and PowerSpan II system to function properly.

Table 6: PowerSpan II Required Register Settings for PowerSpan II Applications

PowerSpan II Register	Register Description	Required Bit	Bit Description	Bit Setting for PowerSpan II
Processor Bus Miscellaneous Control and Status register (PB_MISC_CSR)	The PB_MISC_CSR register contains general control and status bits.	Extended Cycle Enable (EXTCYC)	Determines of the PowerSpan II Master is able to generate extended cycles of 16 bytes or 24 bytes. 0=does not generate extended cycles (default) 1= generates extended cycles	EXTCYC =0
Processor Bus Miscellaneous Control and Status register (PB_MISC_CSR)	The PB_MISC_CSR register contains general control and status bits.	Address Retry Enable (ARTRY_ EN)	Controls PowerSpan II's use of the PB_ARTRY_ signal during the servicing of transactions. 0= PowerSpan III does not assert PB_ARTRY_ (default) 1=PowerSpan II does assert PB_ARTRY_	ARTRY_EN =1
Processor Bus Miscellaneous Control and Status register (PB_MISC_CSR)	The PB_MISC_CSR register contains general control and status bits.	Address Parity Enable (AP_EN)	Controls PowerSpan II's use of the PB_AP_ signal during the servicing of transactions. 0= PowerSpan II does not support Address Parity 1=PowerSpan II does support Address Parity	AP_EN = 0 if address parity is not used AP_EN=1 if address parity is used

1.10.2.1 Example PowerSpan II Configuration Register Programming

Table 7 and Table 8 show specific registers that must be programmed in PowerSpan II. However, the list of registers is only a sub-set of all the possible registers that can be configured in the PowerSpan II device. Refer to the *PowerSpan II User Manual* and the PowerSpan II *initialization Application Note* for a complete listing of all processor bus and PCI registers and configuration requirements.

PowerSpan II Register Programming - Host on the Processor Bus

Table 7 shows what registers in PowerSpan II must be programmed when the host is on the 60x bus. The example values shown in the table are system dependent and are meant only as a guide when programming PowerSpan II.

Table 7: Host on the Processor Bus

PowerSpan II Register	Register Description	Register Values
Processor Bus Register Image Base Address Register PB_REG_BADDR	WinPath internal core or PowerPC processor writes to this register can relocate PowerSpan II PB register space within 32-bit host address space.	BA=0x30000000 (default)
PCI-1 Control and Status Register P1_CSR		The bus master bit (BM) must be set to 1 to enable PowerSpan II to master the PCI-1 bus
PCI-2 Control and Status Register P2_CSR		The bus master bit (BM) must be set to 1 to enable PowerSpan II to master the PCI-2 bus.

Table 7: Host on the Processor Bus

PowerSpan II Register	Register Description	Register Values
Processor Bus PCI Configuration Cycle Information Register PB_CONF_INFO	WinPath internal MIPs core or PowerPC processor writes to these two registers to generate PCI configuration cycle to initialize PCI peripherals residing on the PCI-1 or PCI-2 bus.	
Processor Bus PCI Configuration Cycle Data Register PB_CONF_DATA		
Processor Bus Slave Image Control Register PB_SI*x*_CTL	WinPath internal MIPs core or PowerPC processor writes to these registers to configure the PB slave image. The PB slave interface compares the address and transaction type of any 60x bus cycle and acknowledges it if it falls in the configured address range. PowerSpan II then completes the transaction on either the PCI-1 or PCI-2 bus. The bus that PowerSpan II completes the transaction on is programmed in the Destination (DEST) bit in the PB_SI*x*_CTL register.	For more information about possible register values that enable specific features, see "Processor Bus Slave Image Programming Register Values" on page 47. Also refer to the *PowerSpan II User Manual* for a complete listing of all processor bus and PCI registers
Processor Bus Slave Image Base Address PB_SI*x*_BADDR		
Processor Bus Slave Image Translation Address PB_SI*x*_TADDR		

Processor Bus Slave Image Programming Register Values

The following register examples show how to set-up the PowerSpan II's Processor Bus Slave Image 0. PowerSpan II has a total of eight independent images. The slave images must not have overlapping addresses.

In these examples Master-based decode is disabled and the Endian (END) field must be set to Big Endian in the PB_SI*x*_CTL register.

Register Setting Example One

When the PowerSpan II Processor Bus Slave Image is programmed with the following values, PowerSpan II maps 60x bus transactions within address range 0xA0000000~0xA0007FFF (image size is 32 Kbytes) to the memory space of the PCI-1 bus at the address of 0xA0000000~0xA0007FFF. The prefetching size is 32 bytes.

Register settings:

- PB_SI0_BADDR: 0xA0000000
- PB_SI0_CTL: 0x830000C2

The programming value for the PB_SI0_TADDR register is not specified because address translation is not enabled in this example (TA_EN is set to 0 in the PB_SI0_CTL register).

Register Setting Example Two

When the PowerSpan II Processor Bus Slave Image is programmed with the following values, PowerSpan II maps 60x bus transactions within address range 0x70000000~0x70FFFFFF (image size is 16 Mbytes) to the memory space of the PCI-2 bus at the translated address of 0x40000000~0x40FFFFFF. The minimum read size is 8 bytes because prefetching is disabled.

Register settings:

- PB_SI0_BADDR: 0x70000000
- PB_SI0_CTL: 0xCC400040
- PB_SI0_TADDR: 0x40000000

Register Setting Example Three

When the PowerSpan II Processor Bus Slave Image is programmed with the following values, PowerSpan II maps 60x bus transactions within address range 0x12345000~0x12345FFF (image size is 4 Kbytes) to the memory space of the PCI-1 bus at the translated address of 0x56789000~0x56789FFF. The minimum read size is 4 bytes because prefetching is disabled and the MODE and MEM_IO bits are set to 1 in the PB_SI0_CTL register.

Register settings:

- PB_SI0_BADDR: 0x12345000
- PB_SI0_CTL: 0xC0A00040
- PB_SI0_TADDR: 0x56789000

Register Setting Example Four

When the PowerSpan II Processor Bus Slave Image is programmed with the following values, PowerSpan II maps 60x bus transactions within address range 0x80000000~0x80003FFF (image size is 16 Kbytes) to the I/O space of the PCI-1 bus at the translated address of 0x40000000~0x40003FFF. Prefetching is disabled on the I/O cycle because the minimum read size can be 1,2,3, or 4 bytes.

Register settings:

- PB_SI0_BADDR: 0x80000000
- PB_SI0_CTL: 0xC2800040
- PB_SI0_TADDR: 0x40000000

PowerSpan II Register Programming - Host on the PCI-1 Bus

Table 8 shows what registers in PowerSpan II must be programmed when the host is on the PCI-1 bus. The example values shown in the table are system dependent and are meant only as a guide when programming PowerSpan II.

Table 8: PowerSpan II Register Configuration Example When the Host is on the PCI-1 Bus

PowerSpan II Register	Register Description	Register Values
P1_MISC_CSR		The BSREG_BAR_EN bit must be kept to 1 to allow the PCI host using memory cycle to access PowerSpan II register space. BSREG_BAR_EN=1 (default)
P1_BSREG	PCI bus host writes to this register through a PCI Type 0 configuration cycle to assign PCI memory space to PowerSpan II's 4 Kbyte register space.The register space above the address offset 0x100 is visible to PCI host memory cycle.	
P1_CSR		The Memory Space bit (MS) must be set to 1 to allow PowerSpan II's PCI Target Image to acknowledge PCI memory cycles.

Table 8: PowerSpan II Register Configuration Example When the Host is on the PCI-1 Bus

PowerSpan II Register	Register Description	Register Values
P1_BST*x*	The PCI host writes to these registers to configure a PCI target image. The target image maps PCI bus to PowerSpan II PB bus or PCI-2 bus. The defined PCI target image monitors any external PCI bus transaction on the address and transfer type, only acknowledges it if it falls in the address range, then transfer to the corresponding PowerSpan II PB bus or PCI-2 bus. The bus that PowerSpan II completes the transaction on is programmed in the Destination (DEST) bit in the P1_TI*x*_CTL register.	For more information about possible register values that enable specific features, see "PowerSpan II Register Programming - Host on the PCI-1 Bus" on page 49. Also refer to the *PowerSpan II User Manual* for a complete listing of all processor bus and PCI registers
P1_TI*x*_CTL		
P1_TI*x*_TADDR		
P2_CSR		The Bus Master (BM) bit must be set to 1 to allow PowerSpan II to master the PCI-2 bus interface.
P1_CONF_INFO	Writing to these two registers from PCI host through memory cycle in order to generate PCI configuration cycle out of PowerSpan II PCI-2 interface.	For more information about possible register values that enable specific features, see "PowerSpan II Register Programming - Host on the PCI-1 Bus" on page 49.
P1_CONF_DATA		

PCI Target Image Programming Register Values

The following examples show how to set-up the PowerSpan II's PCI Target Image 0 on PCI-1 bus. PowerSpan II has a total of four independent images. The target images must not have overlapping addresses.

Always disable Master-based decode in the target image and set the Endian (END) field to Big Endian.

Register Setting Example One

When the PowerSpan II PCI Target Image is programmed with the following values, PowerSpan II maps PCI transactions within address range 0xA0000000~0xA003FFFF (image size is 256 Kbytes) to the memory space of the PB bus at the translated address of 0xA0000000~0xA003FFFF. The prefetching size is 64 Kbytes.

Register settings:

- P1_BST0: 0xA0000008
- P1_TI0_CTL: 0xA20AC2D3

The programming value for the PB_TI0_TADDR register is not specified because address translation is not enabled in this example (TA_EN is set to 0 in the P1_TI0_CTL register).

Register Setting Example Two

When the PowerSpan II PCI Target Image is programmed with the following values, PowerSpan II maps PCI-1 transactions within address range 0x70000000~0x70FFFFFF (image size is 16 Mbytes) to the memory space of the PCI-2 bus at the translated address of 0x40000000~0x40FFFFFF. The minimum read size is 8 bytes because prefetching is disabled.

Register values

- P1_BST0: 0x70000008
- P1_TI0_CTL: 0xE84AC240
- P1_TI0_TADDR: 0x40000000

Register Setting Example Three

When the PowerSpan II PCI Target Image is programmed with the following values, PowerSpan II maps PCI transactions within address range 0x12340000~0x1234FFFF (image size is 64 Kbytes) to the memory space of the PB bus at the translated address of 0x56780000~0x5678FFFF. The minimum read size is 4 bytes because prefetching is disabled and the MODE and MEM_IO bits are set to 1 in the PB_TI0_CTL register.

Register values

- P1_BST0: 0x12340008
- P1_TI0_CTL: 0xE0AAC240
- P1_TI0_TADDR: 0x56780000

This example can be used for the PCI bus master accessing the WinPath register space with a 32-bit access

Register Setting Example Four

When the PowerSpan II PCI Target Image is programmed with the following values, PowerSpan II maps PCI transactions within address range 0x00000000~0x0000FFFF (image size is 64 Kbytes) to the memory space of the PB bus at the translated address of 0xA0000000~0xA000FFFF. The prefetching size is 128 bytes.

Register settings:

- P1_BST0: 0x00000008
- P1_TI0_CTL: 0xE00AC2D4
  - P1_TI0_TADDR: 0xA0000000

PowerSpan II supports a base address of 0x00000 if the BAR_EQ_0 bit is set to 1 in the MISC_CSR register.

1.11 AC Timing

This section describes the timing for joint WinPath and PowerSpan II applications. For more information on the AC timing of both devices, refer to the WinPath *Hardware Developer's Guide* and the *PowerSpan II User Manual*.

1.11.1 WinPath and Single PCI PowerSpan II Timing

Table 9 shows the timing analysis for the maximum bus frequency in an application with a WinPath, with an internal frequency of 166 MHz or 200 MHz, and a Single PCI PowerSpan II.

In order to determine the maximum operating frequency between the two devices an analysis of the AC timing was performed for each host bus interface signal. The results of this analysis show the Single PCI PowerSpan II device is able to support the WinPath processor at 80 MHz on the Host Interface.

Table 9: Timing Analysis for WinPath and Single PCI PowerSpan II

WinPath Signal	Output Delay (ns)		Input Setup (ns)	Minimum Clock Period Analysis (ns) (note 3)		Note
	WinPath Output Delay (note 1)		Single PCI PowerSpan II Input Setup			
	166 MHz	200 MHz		166 MHz	200 MHz	
H_ADDR	7.00	7.00	3.00	11.50	11.50	
H_SIZE	7.00	5.50	3.00	11.50	10.00	
H_TYPE	7.00	5.50	3.00	11.50	10.00	
H_BURST_	7.00	5.50	3.00	11.50	10.00	
H_DATA	8.00	7.00	3.00	12.50	11.50	
H_DATA_PAR	8.00	7.00	3.00	12.50	11.50	
H_ADDR_START_	7.00	5.50	3.00	11.50	10.00	
H_DATA_ACK_	7.00	7.00	3.00	11.50	11.50	
H_BUS_ERROR	6.50	6.00	3.00	11.00	10.50	
H_ADDR_ACK_	9.00	7.50	3.00	13.50	12.00	
H_ADDR_RETRY_	5.50	5.00	3.00	10.00	9.50	
H_ARB_OUT_	7.00	7.00	3.00	11.50	11.50	6
H_ARB_DBG_	7.00	5.50	3.00	11.50	10.00	6
H_ABG2_	7.00	7.00	3.00	11.50	11.50	6
H_DBG2_	7.00	5.50	3.00	11.50	10.00	6
H_ADDR_BUSY_	5.50	5.00	3.00	10.00	9.50	
H_DATA_BUSY_	7.00	7.00	3.00	11.50	11.50	

Table 9: Timing Analysis for WinPath and Single PCI PowerSpan II *(Continued)*

WinPath Signal	Output Delay (ns)	Input Setup (ns)		Minimum Clock Period Analysis (ns) (note 3)		Note
	Single PCI PowerSpan II Output Delay (note 8)	WinPath Input Setup (note 2)				
		166 MHz	200 MHz	166 MHz	200 MHz	
H_ADDR	5.00	3.50	3.50	10.00	10.00	
H_SIZE	5.00	5.50	5.00	12.00	11.50	
H_TYPE	5.00	5.50	5.00	12.00	11.50	
H_BURST_	5.00	5.50	5.00	12.00	11.50	
H_DATA	5.00	4.80	4.80	11.30	11.30	
H_DATA_PAR	5.00	4.80	4.80	11.30	11.30	
H_ADDR_START_	5.00	5.50	5.50	12.00	12.00	2
H_DATA_ACK_	5.00	5.00	5.00	11.50	11.50	2
H_BUS_ERROR	5.00	5.70	5.70	12.20	12.20	2
H_ADDR_ACK_	5.00	5.50	5.50	12.00	12.00	2
H_ADDR_RETRY_	5.00	5.70	5.70	12.20	12.20	2
H_ARB_IN	5.00	5.50	5.50	12.00	12.00	2
H_ARB_DBG_	5.00	5.50	5.50	12.00	12.00	
H_ARB2_	5.00	5.50	5.00	12.00	11.50	
H_DBG2_	5.00	5.50	5.50	12.00	12.00	
H_ADDR_BUSY_	5.00	5.50	5.50	12.00	12.00	2
H_DATA_BUSY_	5.00	5.50	5.50	12.00	12.00	
Minimum Clock Period for all Signals				13.50	12.20	4
Maximum Operating Frequency				74 MHz	80 MHz (note 9)	5

Table 9 Notes:

1. WinPath AC timing specification is measured with a load 30pF for the maximum output delay.
2. WinPath has a SUTC bit that can alter the input set-up timing. When operating the host bus interface at a frequency greater than 66 MHz, or when PowerSpan II accesses non-registered SDRAM through the WinPath, set SUTC to 1. Otherwise, set SUTC to 0.
3. When determining the minimum clock period, an additional 1.5 ns was added to the number to allow for the PC-board level design considerations: clock skew, jitter, propagation delay, and capacitive loading delay.
4. The minimum clock period for all signals is calculated by choosing the maximum number from the Minimum Clock Period column in the table.
5. The maximum operating frequency is calculated by the following equation:

   (max operation frequency) = 1000/(min clock period for all signals)
6. The de-rating factor given in the WinPath Hardware Developer's Guide gives an extra 0.35 ns of margin when a compact layout limits these point-to-point signals to 20pF of capacitance. (PowerSpan II input pin capacitance is 10pF)
7. WinPath input-hold time is specified as 0.5 ns. The PowerSpan II has minimum output delay as 1 ns. PowerSpan II input-hold time is specified as 0.3 ns. The WinPath has minimum output delay as 1 ns. Therefore, both devices input-hold timing requirements are met.
8. PowerSpan II’s output delays are measured with a 35pF load.
9. The maximum operating frequency for the WinPath processor and PowerSpan II is 80 MHz. The frequency is limited by the maximum operating frequency of the WinPath’s system clock input.

1.11.2 WinPath and Dual PCI PowerSpan II Timing

Table 10 shows the timing analysis for the maximum bus frequency in an application with a WinPath, with an internal frequency of 166 MHz or 200 MHz, and a Dual PCI PowerSpan II.

In order to determine the maximum operating frequency between the two devices an analysis of the AC timing was performed for each host bus interface signal. The results of this analysis show the Dual PCI PowerSpan II device is able to support the WinPath processor at 80 MHz on the Host Interface.

Table 10: Timing Analysis for WinPath and Dual PCI PowerSpan II

WinPath Signal	Output Delay (ns)		Input Setup (ns)	Minimum Clock Period Analysis (ns) (note 3)		Note
	WinPath Output Delay (note 1)		Dual PCI PowerSpan II Input setup			
	166 MHz	200 MHz		166 MHz	200 MHz	
H_ADDR	7.00	7.00	3.00	11.50	11.50	
H_SIZE	7.00	5.50	3.00	11.50	10.00	
H_TYPE	7.00	5.50	3.00	11.50	10.00	
H_BURST_	7.00	5.50	3.00	11.50	10.00	
H_DATA	8.00	7.00	3.00	12.50	11.50	
H_DATA_PAR	8.00	7.00	3.00	12.50	11.50	
H_ADDR_START_	7.00	5.50	3.00	11.50	10.00	
H_DATA_ACK_	7.00	7.00	3.00	11.50	11.50	
H_BUS_ERROR	6.50	6.00	3.00	11.00	10.50	
H_ADDR_ACK_	9.00	7.50	3.00	13.50	12.00	
H_ADDR_RETRY_	5.50	5.00	3.20	10.20	9.70	
H_ARB_OUT_	7.00	7.00	3.50	12.00	12.00	6
H_ARB_DBG_	7.00	5.50	3.20	11.70	10.20	6
H_ABG2_	7.00	7.00	3.50	12.00	12.00	6
H_DBG2_	7.00	5.50	3.20	11.70	10.20	6
H_ADDR_BUSY_	5.50	5.00	3.00	10.00	9.50	
H_DATA_BUSY_	7.00	7.00	3.00	11.50	11.50	

Table 10: Timing Analysis for WinPath and Dual PCI PowerSpan II *(Continued)*

WinPath Signal	Output Delay (ns)		Input Setup (ns)		Minimum Clock Period Analysis (ns) (note 3)		Note
	Dual PCI PowerSpan II Output Delay (note 8)		WinPath Input Setup (note 2)				
			166 MHz	200 MHz	166 MHz	200 MHz	
H_ADDR	5.00		3.50	3.50	10.00	10.00	
H_SIZE	5.00		5.50	5.00	12.00	11.50	
H_TYPE	5.00		5.50	5.00	12.00	11.50	
H_BURST_	5.00		5.50	5.00	12.00	11.50	
H_DATA	5.00		4.80	4.80	11.30	11.30	
H_DATA_PAR	5.00		4.80	4.80	11.30	11.30	
H_ADDR_START_	5.00		5.50	5.50	12.00	12.00	2
H_DATA_ACK_	5.00		5.00	5.00	11.50	11.50	2
H_BUS_ERROR	5.00		5.70	5.70	12.20	12.20	2
H_ADDR_ACK_	5.00		5.50	5.50	12.00	12.00	2
H_ADDR_RETRY_	5.00		5.70	5.70	12.20	12.20	2
H_ARB_IN	5.00		5.50	5.50	12.00	12.00	2
H_ARB_DBG_	5.00		5.50	5.50	12.00	12.00	
H_ARB2_	5.00		5.50	5.00	12.00	11.50	
H_DBG2_	5.00		5.50	5.50	12.00	12.00	
H_ADDR_BUSY_	5.00		5.50	5.50	12.00	12.00	2
H_DATA_BUSY_	5.00		5.50	5.50	12.00	12.00	
Minimum clock period for all signals					13.50	12.20	4
Maximum operating frequency					74 MHz	80 MHz (note 9)	5

Table 10 Notes:

1. WinPath AC timing specification is measured with a load 30pF for the maximum output delay.
2. WinPath has a SUTC bit that can alter the input set-up timing. When operating the host bus interface at a frequency greater than 66 MHz, or when PowerSpan II accesses non-registered SDRAM through the WinPath, set SUTC to 1. Otherwise, set SUTC to 0.
3. When determining the minimum clock period, an additional 1.5 ns was added to the number to allow for the PC-board level design considerations: clock skew, jitter, propagation delay, and capacitive loading delay.
4. The minimum clock period for all signals is calculated by choosing the maximum number from the Minimum Clock Period column in the table.
5. The maximum operating frequency is calculated by the following equation:

   (max operation frequency) = 1000/(min clock period for all signals)
6. The de-rating factor given in the WinPath Hardware Developer's Guide gives an extra 0.35 ns of margin when a compact layout limits these point-to-point signals to 20pF of capacitance. (PowerSpan II input pin capacitance is 10pF)
7. WinPath input-hold time is specified as 0.5 ns. The PowerSpan II has minimum output delay as 1 ns. PowerSpan II input-hold time is specified as 0.3 ns. The WinPath has minimum output delay as 1 ns. Therefore, both devices input-hold timing requirements are met.
8. PowerSpan II’s output delays are measured with a 35pF load.
9. The maximum operating frequency for the WinPath processor and PowerSpan II is 80 MHz. The frequency is limited by the maximum operating frequency of the WinPath’s system clock input.

IMPORTANT NOTICE AND DISCLAIMER

RENESAS ELECTRONICS CORPORATION AND ITS SUBSIDIARIES (“RENESAS”) PROVIDES TECHNICAL SPECIFICATIONS AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT OF THIRD-PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for developers who are designing with Renesas products. You are solely responsible for (1) selecting the appropriate products for your application, (2) designing, validating, and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. Renesas grants you permission to use these resources only to develop an application that uses Renesas products. Other reproduction or use of these resources is strictly prohibited. No license is granted to any other Renesas intellectual property or to any third-party intellectual property. Renesas disclaims responsibility for, and you will fully indemnify Renesas and its representatives against, any claims, damages, costs, losses, or liabilities arising from your use of these resources. Renesas' products are provided only subject to Renesas' Terms and Conditions of Sale or other applicable terms agreed to in writing. No use of any Renesas resources expands or otherwise alters any applicable warranties or warranty disclaimers for these products.

(Disclaimer Rev.1.01 Jan 2024)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Contact Information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit www.renesas.com/contact-us/.

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.