SB4002A
PCI Intelligent Target Controller

Ver 0.96

SystemBase

Contents

1. Introduction 6
2. PCI BUS 8
2.1 PCI Bus Protocol 8
2.1.1 PCI Feature 8
2.1.2 PCI Signal Description 8
2.1.3 PCI Command 9
2.1.4 PCI Addressing 10
2.1.5 PCI Basic Transaction 11
2.1.6 PCI Target Termination 14
2.1.7 PCI Parity Generation & Checking 17
2.1.8 PCI Error Reporting 17
2.2 PCI Configuration Space Header 19
2.3 PCI Power Management Interface 29
2.4 Vital Product Data (VPD) 33
2.5 CompactPCI Hot Swap 35
2.6 Exclusive Access 37
3. Legacy Bus 39
3.1 Real Access/Delayed Access 39
3.1.1 Real Access 39
3.1.2 Delayed Access 40
3.2 Single Access/Burst Access 41
3.3 Data Split and Combine 44
3.4 Endian Conversion 46
4. Configuration Data in Serial ROM 48
5. Miscellaneous Feature 58
5.1 Interrupt 58
5.2 General Purpose I/O Interface 59
5.3 Serial ROM Interface 59
6. Control Register 60
7. Physical Specification 70
7.1 Pin Location 70
7.2 Pin Description 71
7.3 Package Dimension 75
7.3 Electrical Specification 76
8. Timing Specification 77
8.1 PCI BUS Timing Specifications 77
8.2 Legacy Bus Timing Specifications 79
9. Company Information 81

Table of Contents

Table 2-1. PCI Bus Signals 9
Table 2-2. PCI Bus Command 9
Table 2-3. PCI Bus I/O Addressing Rule 10
Table 2-4. PCI Configuration Space Addressing 10
Table 2-5. PCI Configuration Header Space 19
Table 2-6. Vendor ID Register Layout 20
Table 2-7. Device ID Register Layout 20
Table 2-8. Command Register Layout 20
Table 2-9. Status Register Layout 22
Table 2-10. Revision ID Register Layout 23
Table 2-11. Class Code Register Layout 23
Table 2-12. Cache Line Size Register Layout 24
Table 2-13. Latency Timer Register Layout 24
Table 2-14. Header Type Register Layout 24
Table 2-15. BIST Register Layout 25
Table 2-16. Base Address Register for Memory space 25
Table 2-17. Base Address Register for I/O space 26
Table 2-18. Subsystem Vendor ID Register Layout 27
Table 2-19. Subsystem ID Register Layout 27
Table 2-20. Capability Pointer Register Layout 27
Table 2-21. Interrupt Line Register Layout 27
Table 2-22. Interrupt Pin Register Layout 28
Table 2-23. MIN_GNT Register Layout 28
Table 2-24. MAX_LAT Register Layout 28
Table 2-25. Power Management Interface Registers 29
Table 2-26. Capability ID0 Register for Power Management Interface Layout 29
Table 2-27. Next Item Pointer0 Register Layout 29
Table 2-28. PMC Register Layout 30
Table 2-29. PMCSR Layout 31
Table 2-30. Data Register 31
Table 2-31. Vital Product Data Registers 33
Table 2-32. Capability ID1 Register Layout for Vital Product Data 33
Table 2-33. Next Item1 Pointer Register Layout 33
Table 2-34. VPD Address Register Layout 34
Table 2-35. VPD Data Register Layout 34
Table 2-36. CompactPCI Hot Swap Registers 35
Table 2-37. Capability ID2 Register Layout for CompactPCI Hot Swap 35
Table 2-38. Next Item Pointer2 Register Layout 35
Table 2-39. HS_CSR Layout 36
Table 3-1. DWORD Data path Endian Conversion 46

Table 3-2. WORD Data path Endian Conversion 46
Table 3-3. BYTE Data path Endian Conversion 47
Table 4-1. Serial ROM Address Map 49
Table 6-1. Control Register Map 60
Table 6-2. Base Address Space Setting Register 61
Table 6-3. Reset Register Layout 65
Table 6-4. GPIO Out Enable Register Layout 65
Table 6-5. GPIO Output Register Layout 66
Table 6-6. GPIO Input Register Layout 66
Table 6-7. Interrupt Enable Register Layout 67
Table 6-8. Interrupt Polarity Register Layout 67
Table 6-9. Poll Register Layout 68
Table 6-10. Endian & LOCK Control Register Layout 68
Table 6-11. Muxed Pin Control Register Layout 69
Table 7-1. SB4002A Pin Description 74
Table 7-2. Recommended Operation Conditions 76
Table 7-3. DC Characteristics 76
Table 8-1. PCI Bus Timing Specifications 77
Table 8-2. Legacy Bus Timing Specifications for Write Access 79
Table 8-3. Legacy Bus Timing Specifications for Read Access 80

Figure of Contents

Figure 2-1. PCI BUS Basic Read Transaction 12
Figure 2-2. PCI BUS Basic Write Transaction 13
Figure 2-3. DEVSEL# timing 13
Figure 2-4. Retry 14
Figure 2-5. Disconnect With Data 15
Figure 2-6. Disconnect Without Data 16
Figure 2-7. Target Abort 16
Figure 2-8. Parity Operation (First Read, Second Write) 18
Figure 2-9. Starting an Exclusive Access 37
Figure 2-10. Continuing or Releasing Exclusive Access 38
Figure 3-1. Legacy Bus Real Access 39
Figure 3-2. Legacy Bus Delayed Write Access 40
Figure 3-3. Legacy Bus Delayed Read Access 41
Figure 3-4. Legacy Bus Real Write Burst Access 41
Figure 3-5. Legacy Bus Real Read Burst Access 42
Figure 3-6. Legacy Bus Delayed Write Burst Access 42
Figure 3-7. Legacy Bus Delayed Read Burst Access 43
Figure 3-8. Legacy Bus DWORD Delayed Write Split Access for Byte Device 44
Figure 3-9. Legacy Bus DWORD Delayed Read Split Access for Byte Device 45
Figure 5-1. Interrupt Generator Schematic 58
Figure 5-2. GPIO Controller Schematic 59
Figure 7-1. SB4002A Pin Location 70
Figure 8-1. Output Timing Measurement Conditions 78
Figure 8-2. Input Timing Measurement Conditions 78
Figure 8-3. Legacy Bus Timing Specifications for Write Access 79
Figure 8-4. Legacy Bus Timing Specifications for Read Access 80

1. Introduction

Summary of SB4002A features:

- Compliant with PCI Local Bus Specification 2.3
- Supports 32-bit Bus / 33MHz and 66MHz
- Supports data transmission of max. 264MB/sec
- Supports PCI Power Management 1.1
- Supports CompactPCI and CompactPCI Hot Swap
- Supports Vital Product Data
- Provides the real time access and delay time access on the rear side Legacy bus (provides internal FIFO 16 DWORD for delay time access)
- Provides the real time access on the rear side Legacy bus
- Provides 5 address spaces (selective between memory and I/O)
- Provides the chip select (CS) signal for each address space / interrupt
- Supports burst access
- Supports the write/read signal timing control
- Supports selection of data path (8-bit/16-bit/32-bit)
- Provides interrupt control register
- Provides big/little endian conversion
- Downloads the Configuration Space Header data from external Serial ROM at booting
- 8 x GPIO ports
- 3.3V I/O, 5V tolerance
- 176pin TQFP package

Over the last few years, PCI Local Bus has emerged as the strongest and most widely used bus system in the option card bus industry that used to have been dominated by ISA/EISA. We expect that this trend will continue for substantial amount of time thanks to convenience and high performance of PCI BUS. It is also expected that PCI Local Bus will expand its application even further as the PCI families such as CompactPCI, PCI-X and mini PCI are developed, over the conventional PCI types, to provide various types of solution. Therefore, PCI Local Bus becomes the most important factor to be considered in the electronics industry. While PCI Local Bus is powerful and has many convenient functions, however, difficulty in developing PCI becomes a burden for the developers. There are companies who provide the PCI target board developers with the PCI target interface one chip solution or the FPGA embedded PCI core solution. Because these suppliers are from USA and Taiwan, the developers in other countries suffer difficulty in analyzing solutions, lack of technical support and high price.

In this context, SystemBase has released SB4001 which enabled the users to develop PCI target boards with reduced cost. SystemBase releases SB4002A with improved features at the same price level. Because SB4002A is compatible with its previous model, you can easily use the new model without trouble. SB4002A also provides an easy-to-use interface for the developers who have never used SystemBase chip before. SystemBase has developed the PCI boards with the PCI target interface core built-in, and SB4001 adopted in the products of SystemBase has been proven in the field for years. Developed based on the know-how

acquired through SB4001, SB4002A offers more robustness and convenience than any other solutions. SB4002A will enable the users to develop the PCI target boards at reduced time and effort, and to replace the existing boards with cost-effective PCI option boards.

2. PCI BUS

2.1 PCI Bus Protocol

2.1.1 PCI Feature

- PCI is a synchronous bus which supports 33MHz/66MHz clock and 32-bit/64-bit data path. (SB4002A supports 33MHz/66MHz clock and 32-bit data path.)
- PCI increases transmission efficiency via Linear/Cacheline wrap burst access. (SB4002A supports linear burst access.)
- PCI supports plug & play, and automatically reads the required resources from BIOS/OS and allocates them to the device.
- PCI provides high reliability of bus by supporting parity checking and target termination.
- PCI multiplexes the address/data pins to increase efficiency of chip package.

2.1.2 PCI Signal Description

Signal name	Type	Description
CLK	in	Provides timing for all actions of PCI bus. The signal gives input for all the PCI devices, and its rising edge becomes the reference for input/output and timing constant of all signals. (exception: RST#, INTx#, PME#, CLKRUN#)
RST#	in	The reset signal of the PCI system.
AD[31:0]	t/s	Address/Data multiplexed signal. This signal is used during the PCI transmission as the address signal in the address phase or as the data signal in the data phrase.
C/BE[3:0]#	t/s	Bus Command/Byte Enable multiplexed signal. It is used during the PCI transmission as BUS Command in the address phase or as Byte Enable in the data phrase.
PAR	t/s	Provides the even parity for AD[31:0] and C/BE[3:0].
FRAME#	s/t/s	Indicates beginning and duration of PCI access.
IRDY#	s/t/s	Initiator Ready. The signal indicates the bus master's ability to complete the current data phase of the transaction.
TRDY#	s/t/s	Target Ready. The signal indicates the target device's ability to complete the current data phase of the transaction.
STOP#	s/t/s	Indicates the current target is requesting the master to stop the current transaction.
LOCK#	s/t/s	Provides for the exclusive use of a resource.
IDSEL	in	Indicates that a device is selected during configuration read and write transactions.

Signal name	Type	Description
DEVSEL#	s/t/s	Device Select. Used by the selected target to indicate that it is selected.
PERR#	s/t/s	Parity Error. Indicates that a parity error occurs in the data phase.
SERR#	o/d	System Error. Indicates occurrence of a critical error such as a parity error in the address phase.
INTA#/INTB# INTC#/INTD#	o/d	PCI interrupt signals. These signals except INTA# are the interrupt signals for multi functions.

Table 2-1. PCI Bus Signals

* The above table describes the main signals used in the PCI target device only.

in: Input signal

out: Totem Pole Output signal

t/s: Tri-State. Bi-directional, tri-state in/out signal.

s/t/s: Sustained Tri-State. The low enable signal drives only one device at a time. At least one clock must be driven to high before the Hi-Z state.

o/d: Open Drain. It is shared by numbers of devices.

2.1.3 PCI Command

C/BE[3:0]#	Command Type
0000	Interrupt Acknowledge
0001	Special Cycle
0010	I/O Read
0011	I/O Write
0100	Reserved
0101	Reserved
0110	Memory Read
0111	Memory Write
1000	Reserved
1001	Reserved
1010	Configuration Read
1011	Configuration Write
1100	Memory Read Multiple
1101	Dual Address Cycle
1110	Memory Read Line
1111	Memory Write and Invalidate

Table 2-2. PCI Bus Command

Out of the commands mentioned in the above table, SB4002A does not support Interrupt Acknowledge, Special Cycle, Dual Address Cycle and the reserved commands. It does not support Memory Read Multiple, Memory Read Line, and Memory Write and Invalidate Command as well as SB4002A disregards the commands which are not supported.

2.1.4 PCI Addressing

I/O Space Addressing

In the I/O Address Space, all 32 AD lines(AD[31:0]) are used to provide a full byte address. The following table shows the starting byte and BE# [3:0] in the data phrase depending on the value of AD [1:0].

AD[1:0]	Starting Byte	BE#[3:0] Combinations
00	Byte 0	xxx0 or 1111
01	Byte 1	xx01 or 1111
10	Byte 2	x011 or 1111
11	Byte 3	0111 or 1111

Table 2-3. PCI Bus I/O Addressing Rule

If the above combination is not satisfied, SB4002A disregards the command.

Memory Space Addressing

In the Memory Address Space, accesses are decoded to a DWORD address using AD[31:2]. AD[1:0] determines the method of address increment for burst access. SB4002A supports the linear incrementing only (00), and disconnect others after the first transmission.

Configuration Space Addressing

In the Configuration Address Spaces, accesses are decoded to a DWORD address using AD[7:2]. Configuration Space Addressing is divided into type0 and type1. Type1 is supported in the PCI-to-PCI bridge device only, and is disregarded in SB4002A. The following table shows the Configuration Space addressing of type0.

31 ... 11	10 ... 8	7 ... 2	1	0
Reserved	Function Number	Register Number	0	0

Table 2-4. PCI Configuration Space Addressing

Function Number is an encoded value used to select one of eight possible functions on a multifunction device. Register Number is an encoded value used to index a DWORD in

Configuration Space of the intended target. Because SB4002A supports the single function only, only Function Number 0 is available. For accessing Configuration Space, SB4002A supports the IDSEL signal 1 only in the address phase.

2.1.5 PCI Basic Transaction

All the functions specified below are based on the rising edge of CLK.

Read Transaction

All transactions are triggered as FRAME# is asserted. The bus master outputs the address on AD [31:0] and the command on C/BE [3:0]# as it asserts FRAME#. For Configuration Space access, the bus master outputs 1 on IDSEL. This is called as address phase. Only 1 clock is maintained for address phase. The target shall receive and decode address, command and IDSEL. The selected target shall assert DEVSEL# to notify the master of its selection. The data phrase starts at Address phase.

In the read transaction, the target outputs the signal on AD [31:0] and the master receives the signal, which is the contrary to the address phase. Therefore, it is required to insert the turn around cycle by deasserting TRDY# for at least one clock. If the data requested by the master is prepared, the target outputs the data on AD[31:0], and asserts TRDY#. If the master is ready to receive the data, i.e. IRDY# is asserted, the data transmission is terminated and the data phase is over. If FRAME# is asserted instead of IRDY#, which means that the transaction is not terminated and the burst is in progress, the target outputs the data on AD [31:0] and asserts TRDY#. When the master is prepared to receive the data, IRDY# is asserted and the data transmission is terminated. This process is repeated until FRAME# is deasserted when both TRDY# and IRDY# are asserted at the same time and the data transmission is completed. In this case, TRDY#, IRDY# and DEVSEL# are all deasserted and a transaction is terminated.

In Figure 2-1, the master device starts transaction of PCI bus at CLK 1.

At this point, as the address phase is started, the master asserts FRAME#, outputs the address on AD, and asserts the READ command on C/BE#. The target device reads the address and the command at CLK2, and prepares a transaction. A turnaround cycle is inserted at CLK 2. The master device asserts IRDY#, indicating that it is ready to transfer data.

At CLK 3, the target device asserts DEVSEL# to indicate its selection, and asserts TRDY# and outputs the read data on AD to indicate that it is ready to transfer data,

At CLK 4, IRDY# and TRDY# are asserted at the same time, and a data transfer occurs. As FRAME# is asserted, a burst access occurs. And as TRDY# is deasserted, a wait cycle is inserted by the target.

At CLK 5, the target reasserts TRDY#, indicating that it is ready to transfer data. At CLK 6, IRDY# and TRDY# are asserted at the same time, and a data transfer occurs.

As Frame# is asserted, a burst access occurs. Because IRDY# is deasserted, a wait cycle is inserted by the master.
At CLK 7, IRDY# is reasserted, and a data transfer occurs at CLK 8. At this point, because FRAME# is deasserted, the transaction is terminated. The master deasserts IRDY#, and the target deasserts TRDY# and DEVSEL#. Therefore a PCI BUS read transaction is completed.

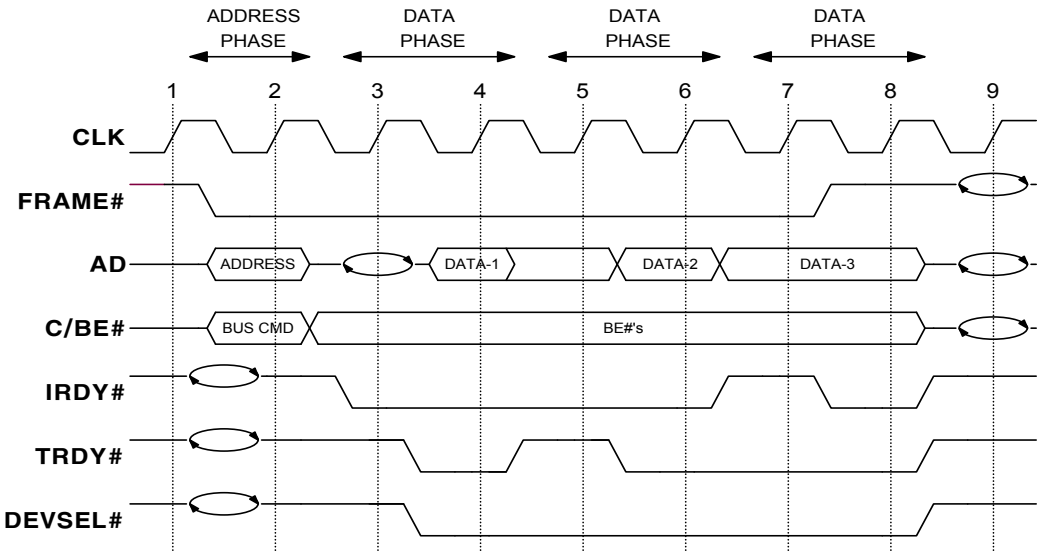


Figure 2-1. PCI BUS Basic Read Transaction

Write Transaction

The address phase is processed in the same way as in the read transaction. In the write transaction, because the master outputs the signal on AD [31:0] at both the address phase and the data phase, unlike the read transaction, no turn around cycle is required. The data phase may be completed right after the address phase depending on the capability of the master and the target. The data phase is also the same as that in the read transaction, except that the master outputs data on AD[31:0] in the write transaction. When the data is ready, the master outputs the data on AD [31:0] and asserts IRDY#. If the target is prepared to receive the data, i.e. TRDY# is asserted, the data transmission is completed and the data phase is terminated. This process is repeated until FRAME# is deasserted when a data transfer is completed, which means it is the last data phrase. In this case TRDY#, IRDY# and DEVSEL# are all deasserted and a transaction is terminated.

In Figure 2-2, the master device starts PCI bus transaction at CLK 1. At this point, as the address phase is started, the master asserts FRAME#, outputs the address on AD, and asserts the WRITE command on C/BE#.
At CLK 2, the target reads and decodes the address and command, and asserts DEVSEL# to indicate that the target is selected. The master device asserts IRDY# and the target device asserts TRDY#, indicating that they are prepared to transfer data.
At CLK 3, IRDY# and TRDY# are asserted at the same time, and a data transfer occurs. As

FRAME# is asserted, a burst access occurs. At CLK 4, IRDY# and TRDY# are asserted at the same time, and a data transfer occurs. As FRAME# is asserted, a burst access occurs. And as IRDY# and TRDY# are deasserted, a wait cycle is inserted.

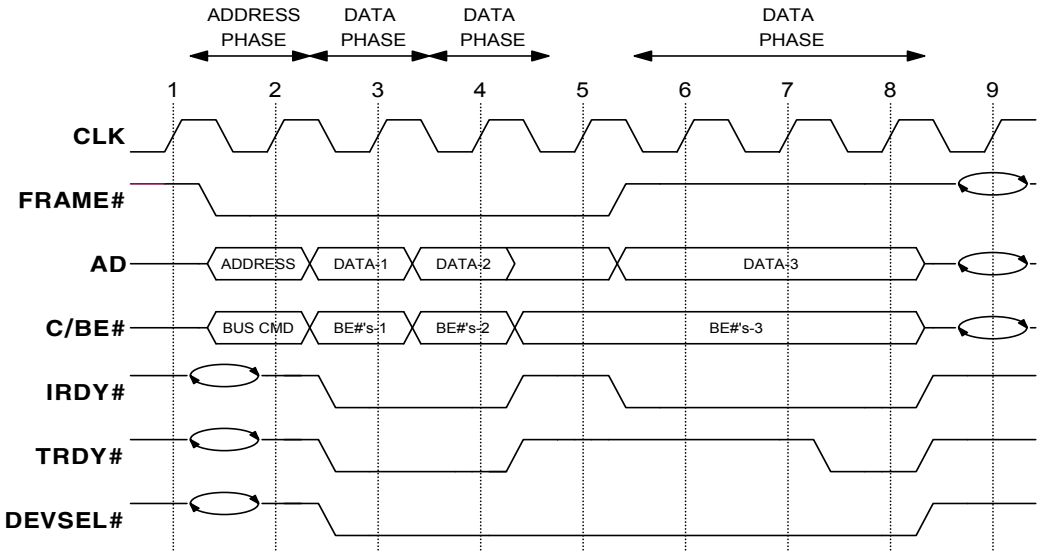


Figure 2-2. PCI BUS Basic Write Transaction

At CLK 5, the master device asserts IRDY#, indicating that it is ready to transfer data, while TRDY# keeps deasserted. The target extends the wait cycle. At CLK 8, the target asserts TRDY#, indicating that it is ready to transfer data. At CLK 9, IRDY# and TRDY# are asserted at the same time, and a data transfer occurs. As FRAME# is deasserted, the transaction is terminated. The master deasserts IRDY# and the target deasserts TRDY# and DEVSEL#. Therefore a PCI bus write transaction is completed.

The following rules are applied to the basic transaction.
- The point that DEVSEL# is asserted depens on the characteristics of the target (Refer to config).

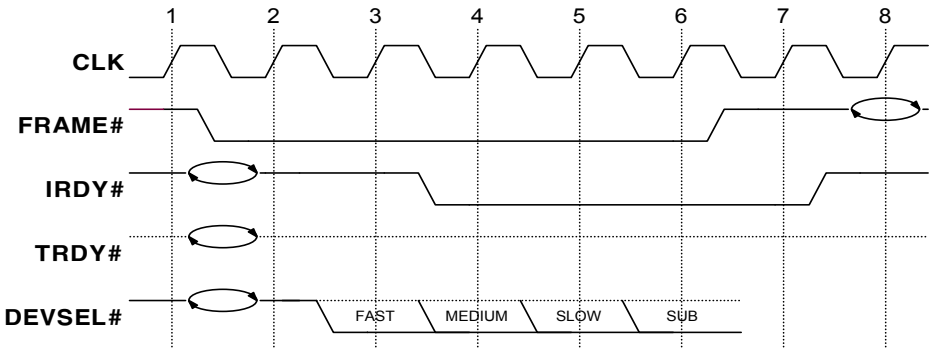


Figure 2-3. DEVSEL# timing

The above figure shows the points DEVSEL# is asserted depending on the characteristics of the

device. The device sets the DEVSEL timing of status register of the configuration space header depending on the time point it can assert DEVSEL#. In the above figure, if DEVSEL# is asserted at CLK 2, CLK 3 and CLK 4, it is set to FAST (00b), MEDIUM (01b) and SLOW (10b), respectively. If no device has asserted DEVSEL# by CLK 4, the subtractive decoding device asserts DEVSEL#. If the system has no subtractive decoding device, DEVSEL# is not asserted at CLK 4, and the master terminates the transaction with Master Abort. (SB4002A supports MEDIUM.)

- In any case, a data transfer occurs if both TRDY# and IRDY# are asserted.
- Once TRDY# and IRDY# are asserted in a data phase, they cannot be deasserted by the end of the data phase.
- In a data phase, C/BE[3:0]# means byte enable. It always has an effective value through the data phase. Once a data phase is started, the value must not be changed. In a burst transfer, as soon as a data phase is completed, the byte enable value for the next data phase is assigned on C/BE[3:0]#.

2.1.6 PCI Target Termination

PCI bus provides the target termination, as well as the master termination which is provided by other bus, in order to handle the situation that the target fails to execute the request of the master or that it is necessary for the target to stop execution.
Target termination is divided into Retry, Disconnect and Target-Abort.

Retry
The target terminates transaction when it cannot perform the request of the master due to other internal process. No data transfer occurs in this case, and the master must retry the request.
At the first data phase, the target asserts STOP# instead of TRDY# to execute retry. It also asserts IRDY# and STOP# together to terminate the transaction with RETRY.

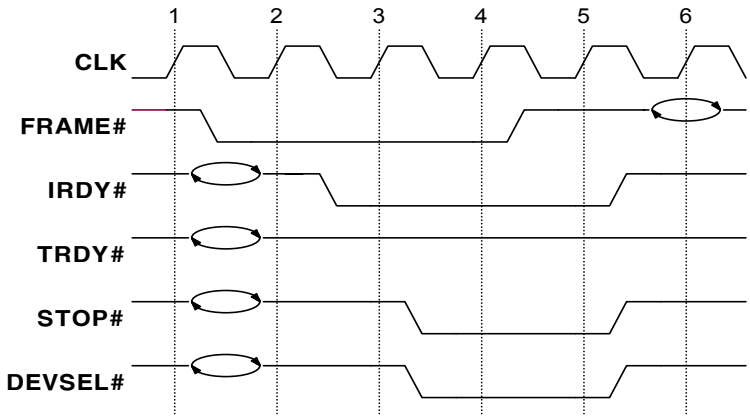


Figure 2-4. Retry

In the above figure, a transaction is started normally at CLK 1 or CLK 2. If the target determines

that it cannot execute the transaction, it asserts STOP# instead of TRDY# at CLK 3. The master checks STOP# at CLK 4, and deasserts FRAME#. At CLK 5, the master deasserts IRDY#, and the target deasserts STOP# and DEVSEL#, terminating the transaction with RETRY,

Disconnect

The target terminates transaction when it cannot continue the burst from the master after it transfers at least one data signal. If the master outputs the request for burst access to the target that cannot continue the burst access, the target may complete the first data phase and terminates the transaction with Disconnect.

Disconnect is divided into with data and without data. For with data, the target terminates transaction after transfering the data at the last data phase, and for without data, the target terminates transaction without transfering the data at the last data phase. For with data, both TRDY# and STOP# are asserted at the same time at the last data phase, and for without data, the target deasserts TRDY# and asserts STOP#, terminating the transaction with Disconnect.

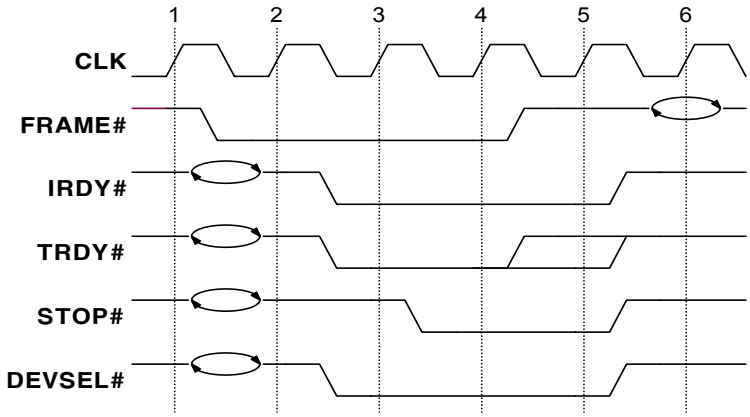


Figure 2-5. Disconnect With Data

In the above figure, a transaction is started normally at CLK 1 or CLK 2. If the burst access is unavailable at CLK 3, the target asserts STOP#3, indicating that it will disconnect. At CLK 4, the master checks STOP#, deasserts FRAME#, and then, terminates the transaction at CLK 5. In this transaction, a data transfer occurs at CLK 3 (disconnect transaction), and at CLK 4 in which STOP# is asserted (with data). If TRDY# is asserted at CLK 5, a data transfer occurs.

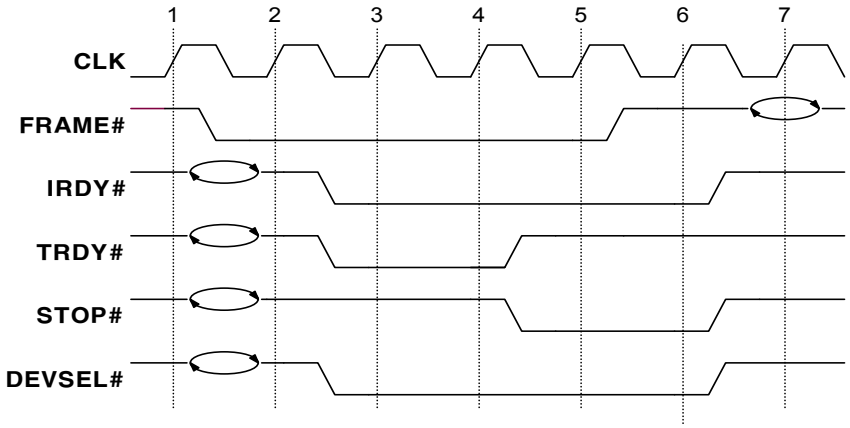


Figure 2-6. Disconnect Without Data

In the above figure, a transaction is started at CLK 1, CLK 2 and CLK 3. If the burst access is unavailable at CLK 4, the target asserts STOP#3, indicating that it will disconnect. It also deasserts TRDY#, indicating that it will not transfer data any more. The master checks STOP# at CLK 5, deasserts FRAME#, and then, terminates the transaction at CLK 6. In this transaction, data is transferred at CLK 3 and CLK 4 (disconnect transaction). After that, because TRDY# is deasserted, no more data are transferred (without data).

Target-Abort

The target terminates transaction if a critical error has occurred, or the target cannot complete the request from the master.

Target-Abort is available when DEVSEL# is asserted. In this state, the target deasserts DEVSEL# and asserts STOP# to terminate the transaction with Target-Abort.

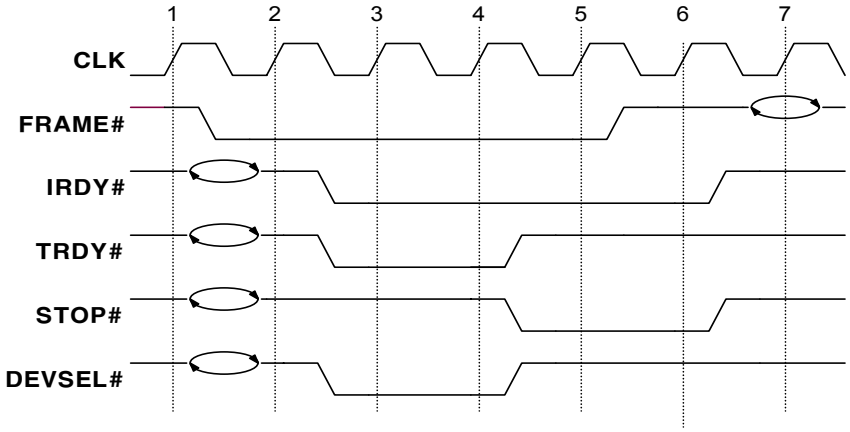


Figure 2-7. Target Abort

In the above figure, a transaction is started at CLK 1, CLK 2 and CLK 3. The target finds a critical error at CLK 4, deasserts TRDY# and DEVSEL#, and asserts STOP#. At CLK 5, the master checks this, recognizes Target-Abort, and deasserts FRAME#. At CLK 6, IRDY# and STOP# are deasserted, and the transaction is terminated with Target-Abort.

2.1.7 PCI Parity Generation & Checking

In order to verify that address and data are transferred properly, PCI outputs the even parity signal via PAR. The devices must output the even parity on AD[31:0]. In other words, the master outputs the parity corresponding to the address phase and the data phase of the write transaction, and the target outputs the parity corresponding to the data phase of the read transaction.
The even parity of AD [31:0] and C/BE [3:0]# are output on PAR. Even a meaningless value on AD[31:0] must be included in the parity. The address parity is output on PAR for a clock after the address phase. The data parity is output on PAR for a clock after a data phase with valid data, in other words, after both IRDY# and TRDY# are asserted.
In figure 2-8, the master starts outputting the PAR for the address phase at CLK 2 and CLK 6, and the PAR for the data phase for writing at CLK 7. The target starts outputting the PAR for the data phase at CLK 4.

2.1.8 PCI Error Reporting

PCI enables the target to report transaction errors to the master.
PERR# is the error reporting signal output by the target for a data parity error, and SERR# is the error reporting signal output for a critical error on the system.
For a write transaction, if the even parity is not consistent in PAR, AD [31:0] and C/BE[3:0]# from the next clock after IRDY# is asserted, the target asserts PERR#. For a read transaction, if the even parity is not consistent in PAR, AD [31:0] and C/BE[3:0]# from the next clock after TRDY# is asserted, the master asserts PERR#. If PERR# is asserted, PERR# is maintained for two clocks after the data phase is terminated.
SERR# is asserted if an address parity error occurs. If the even parity is not consistent between PAR, the address value and the command value after the address phase, SERR# must be asserted for the next one clock. SERR# is also asserted for other critical errors.
(* SB4002A supports the PERR# for the data parity error only. If an address parity error occurs, the device disregards the transaction, and terminates the transaction with Master-Abort.)

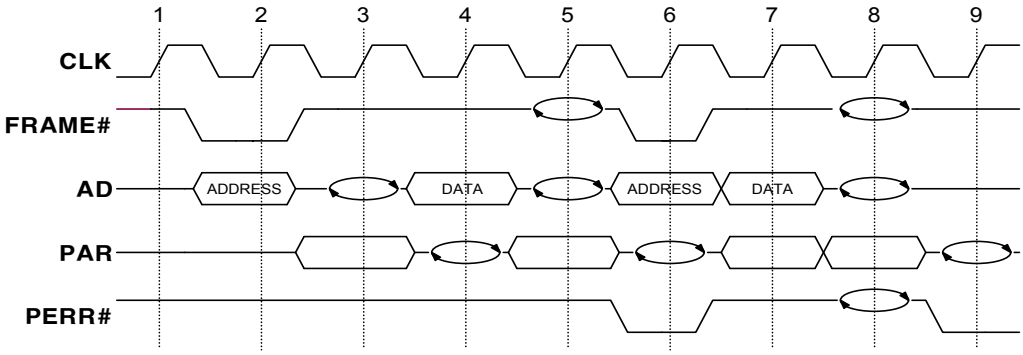


Figure 2-8. Parity Operation (First Read, Second Write)

In figure 2-8, for read, the master checks PAR output by the target at CLK 5, and asserts PERR# if a parity error occurs. For write, the target checks PAR sent by the master at CLK 8, and asserts PERR# if a parity error occurs.

2.2 PCI Configuration Space Header

All PCI devices must implement Configuration Space Header. The Configuration Space Header contains information on the device and the resources (base address space, interrupt, and etc.) used by the device. At system booting or system configuration, BIOS or OS reads the Configuration Space Header, runs P&P, and allocates the system resources.

This I/O space can be accessed by DWORD, WORD and BYTE. SB4002A responses for type0 access only.

The following table describes the details of the Configuration Space Header. SB4002A downloads header information from the external serial ROM at booting.

Address	BYTE3	BYTE2	BYTE1	BYTE0
00h	Device ID		Vendor ID	
04h	Status		Command	
08h	Class Code			Revision ID
0Ch	BIST	Header Type	Latency Timer	Cache Line Size
10h	Base Address0 Register			
14h	Base Address1 Register			
18h	Base Address2 Register			
1Ch	Base Address3 Register			
20h	Base Address4 Register			
24h	Base Address5 Register			
28h	Cardbus CIS Pointer			
2Ch	Subsystem ID		Subsystem Vendor ID	
30h	Expansion ROM Base Address			
34h	Reserved			Capabilities Ptr
38h	Reserved			
3Ch	Max_Lat	Min_Gnt	Interrupt Pin	Interrupt Line

Table 2-5. PCI Configuration Header Space

Device ID Register

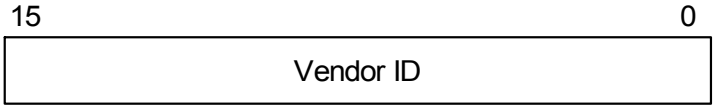


Table 2-6. Vendor ID Register Layout

Manufacturer of the device. The Valid ID is allocated by PCI SIG. If no specific vendor ID is assigned, the 14A1h value allocated by PCI SIG may be used. [RO : downloadable]

Device ID Register

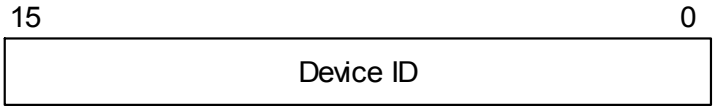


Table 2-7. Device ID Register Layout

A unique ID of each device. Device ID can be assigned by the manufacturer. [RO : downloadable]

Command Register

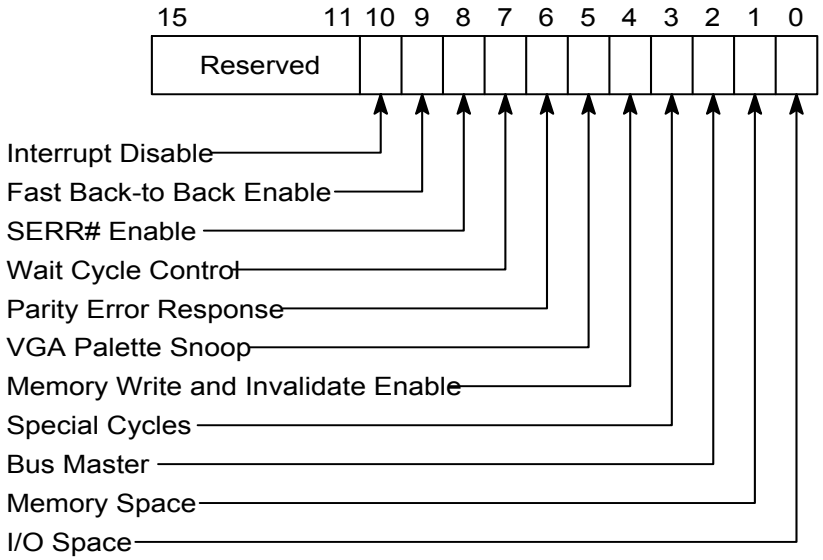


Table 2-8. Command Register Layout

The register that controls reply for the PCI cycle. [non-downloadable]

Bit[15:11] : Reserved

Bit[10] : Interrupt Disable. The bit enables/disables INTX#. The value 0b enables INTX# and 1b disables INTX#. The default is 0b. [R/W] (* Supported in PCI Specification 2.3)

Bit[9] : Fast Back-to-Back Enable. The bit indicates capability of fast back-to-back transaction between the master and other devices. This is activated when all the targets on the bus support fast back-to-back transaction. The value 1b indicates that the master supports fast back-to-back transaction with other device, while 0b indicates that the master does not support fast back-to-back transaction. Since this bit is for the master, SB4002A does not support this bit. The default is 0b. [RO]

Bit[8] : SERR# Enable. The enable bit for SERR# driver. The value 1b enables the target to assert SERR#, and 0b disables assertion of SERR#. The default is 0b. [RO]

Bit[7] : Stepping Control. The bit indicates whether or not the device supports address/data stepping. The default is 0b. [R] (* Not supported in PCI Specification 2.3.)

Bit[6] : Parity Error Response. The bit controls parity error processing. The value 1b executes normal action against a parity check error, and 0b sets the parity error status bit only and does not assert PERR# against a parity check error. The default is 0b. [R/W]

Bit[5] : VGA Palette Snoop. The bit controls processing of palette register of the VGA graphic card. SB4002A does not support this bit. The default is 0b. [RO]

Bit[4] : Memory Write and Invalidate Enable. The bit indicates whether the master can use the Memory Write and Invalidate command. If the value is 1b, the master can use this command, and if 0b, the master must use the Memory Write Command. Since this bit is for the master, SB4002A does not support this bit. The default is 0b. [RO].

Bit[3] : Special Cycles. The bit controls reply for a Special cycle on the PCI bus. The value 1b supports the special cycle, and 0b disregards the special cycle. Because SB4002A does not support a special cycle, the default is 0b. [RO]

Bit[2] : Bus Master. The bit enables the device to act as a master. The value 0b disables the device to act as a master, and 1b enables the master to start a PCI transaction. Because SB4002A is a target device, the default is 0b. [RO]

Bit[1] : The bit controls response to the Memory Space cycle. The value 1b enables the device to respond. The default is 0b. [R/W]

Bit[0] : I/O Space. The bit controls response to the I/O Space cycle. The value 1b enables the device to respond. The default is 0b. [R/W]

Status Register

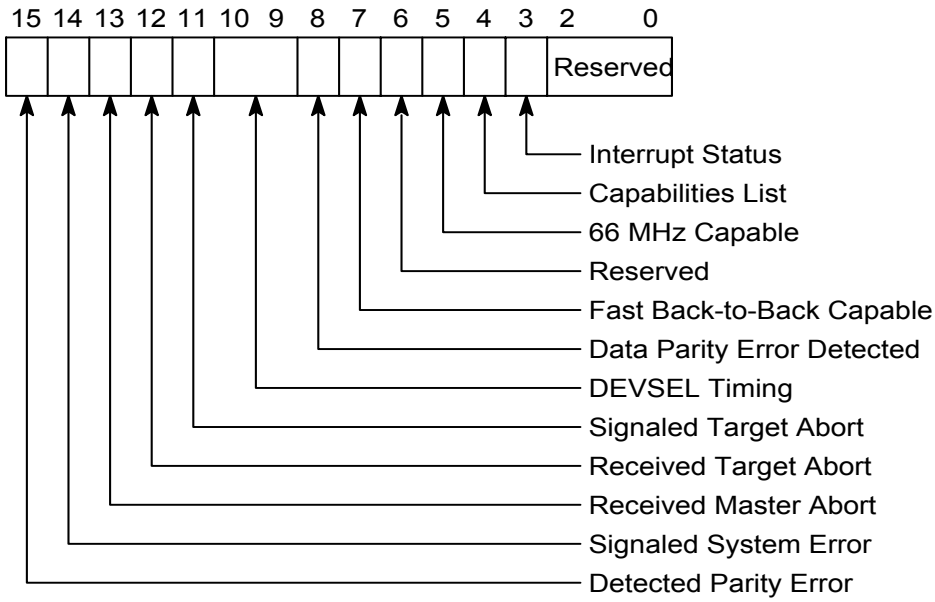


Table 2-9. Status Register Layout

The status of the PCI bus of a device [downloadable]

Bit[15] : Detect Parity Error. This bit is set if a parity error occurs in a device. This bit is set regardless of the parity error response of the command register. [R/WC]

(* WC(Write Clear) : This bit is reset when the value is 1b.)

Bit[14] : Signal System Error. This bit is set when a device asserts SERR#. SB4002A does not support this bit since it does not use SERR#. The default is 0b. [RO]

Bit[13] : Received Master Abort. The bit indicates that the master terminates the transaction with Master-Abort. Since this bit is for the master, SB4002A does not support this bit. The default is 0b. [RO]

Bit[12] : Received Target Abort. The bit indicates that the master terminates the transaction with Target-Abort. Since this bit is for the master, SB4002A does not support this bit. The default is 0b. [RO]

Bit[11] : Signaled Target Abort. This bit is set when the target device terminates a PCI transaction with Target-Abort. Because SB4002A does not support Target-Abort, this bit is not used. The default is 0b. [RO]

Bit[10:9] : DEVSEL Timing. The bit provides information on the timing for start of the cycle and assertion of DEVSEL#. 00b is fast, 01b is medium, and 10b is slow. Because SB4002A is a medium device, the default is 01b. [RO]

Bit[8] : Master Data Parity Error. The bit is 1b if PERR# is asserted, the master starts the transaction, and the parity error response bit of the common register is enabled. Since this bit is for the master,

SB4002A does not support this bit. The default is 0b. [RO]

Bit[7] : Fast Back-to-Back Capable. The bit indicates whether the target device supports the fast back-to-back transaction for a different device. 1b is support, and 0b is no-support. The default is 1b. [RO]

Bit[6] : Reserved. The default is 0b

Bit[5] : 66MHz Capable. The bit indicates whether the device supports 66MHz. 0b indicates that the device supports 33MHz only, and 1b indicates that the device supports both 33MHz and 66MHz. Because SB4002A supports 66MHz, the default is 1b. [RO]

Bit[4] : Capabilities List. The bit indicates if a new Capabilities Linked List Pointer is implemented on 34h of the Configuration Space Header. The value 1b indicates that the Capability Linked List is supported, and 0b indicates that it is not supported. [RO : downloadable]

Bit[3] : Interrupt Status. The bit indicates the interrupt status of a device or function. If the interrupt disable of the command register is 0b and the interrupt status is 1b, INTx# of the device or the function is asserted. The interrupt disable value 1b does not affect this bit. [RO]

Bit[2:0] : Reserved.

Revision ID Register

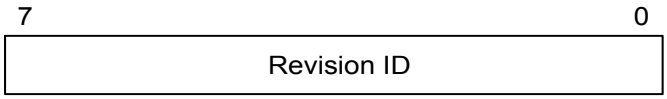


Table 2-10. Revision ID Register Layout

The register indicates the revision of the device. The revision ID is assigned by the manufacturer. [RO : downloadable]

Class Code Register

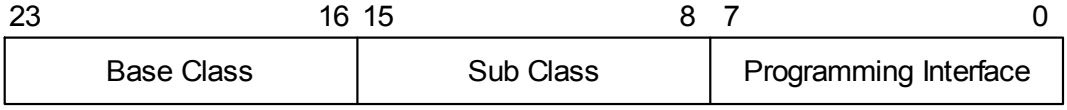


Table 2-11. Class Code Register Layout

This register provides the description on the function implemented by the device. The register is divided into Base Class, Sub Class and Programming Interface by bytes. The register can be set to the value defined in the PCI bus specification. [RO : downloadable]

Cache Line Size Register

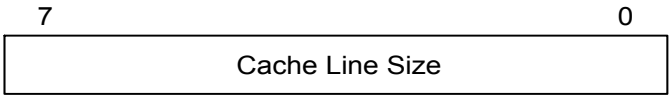


Table 2-12. Cache Line Size Register Layout

The register designates the size of the cache line of the system. Since this register is for the master, SB4002A does not support this register. The default is 0b. [RO : non-downloadable]

Latency Timer Register

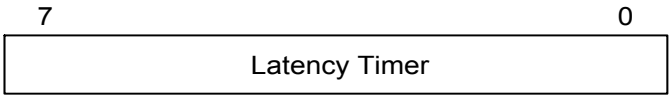


Table 2-13. Latency Timer Register Layout

The register designates the latency clock for the bus master that executes a burst access. Since this register is for the master, SB4002A does not support this register. The default is 0b. [RO : non-downloadable]

Header Type Register

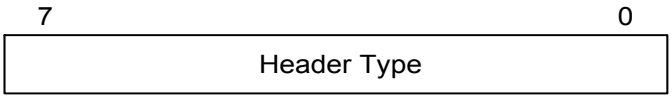


Table 2-14. Header Type Register Layout

Configuration Space Header Type [RO : non-downloadable]

Header Type register identifies the layout of second part of the predefined header(beginning at byte 10h in Configuration Space) and also whether or not the device contains multiple functions.

Bit[7] : This bit indicates whether the device supports multi-function or single function. Because SB4002A supports single function, the default is 0b.

Bit[6:0] : This bit identify the layout of second part of the predefined header. 00h is the target device, 01h is the PCI-to-PCI bridge, and 02h is the Cardbus bridge. Because SB4002A is a target device, the default is 00.

BIST(Built-in Self Test) Register

7	6	5 4	3 0
BIST Capable	Start BIST	Reserved	Completion Code

Table 2-15. BIST Register Layout

This register is used to control BIST.

Bit[7] : The value 1b supports BIST, and 0b does not support BIST.

Bit[6] : The value 1b starts BIST. When the BIST is finished, the device resets BIST by writing 0b on this bit.

Bit[5:4] : Reserved.

Bit[3:0] : The value 0b indicates that BIST is passed. Other values provide information on errors. SB4002A does not support this bit. The default is 0b. [RO : non-downloadable]

Base Address Register0/1/2/3/4/5

Designates the base address for access of a device or a memory on the Local Bus. Bit[0] in all Base Address Registers is read-only and used to determine whether the register maps into Memory or I/O Space. [downloadable]

For Memory Space

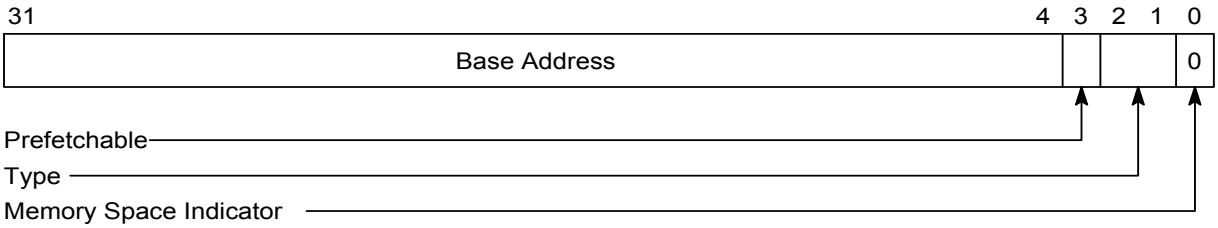


Table 2-16. Base Address Register for Memory space

Bit[31:4] : This bit sets the base address of this Memory space. It is used by the address decoder. [W/R]

Bit[2:1] : 00b indicates 32bit address space, and 10b indicates 64bit address space. The remaining values are reserved. [RO]

Bit[0] : This bit indicates that Base Address Register maps Memory Space. [RO]

For I/O Space

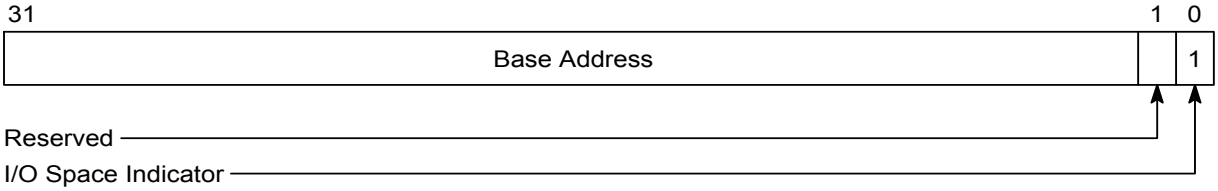


Table 2-17. Base Address Register for I/O space

Bit[31:2] : This bit sets the base address of this I/O space. It is used by the address decoder. [W/R]
Bit[1] : Reserved. [RO]
Bit[0] : This bit indicates that Base Address Register maps I/O Space. [RO]

Tips：How to set the Base Address

Bit[31:4] of the memory base address and Bit[31:2] of the I/O base address are used to indicate the size and location of the I/O Space. BIOS and OS check the size of the I/O Space from this register, and designate the appropriate address.

Set all the address bits of the required I/O Space to read only 0 bit, and other upper bits to read/write. For example, in order to set the 1Kbyte memory space, set Bit[9:4] to hardwired 0h, and set the upper bits to read/write bits. Now BIOS or OS writes FFFFFFFFh on the entire base address register, reads the hardwired 0 bits to check the size of the I/O space, and then, writes the appropriate address on the base address register. The internal address decoder determines access to the I/O space based on this address. The base address registers which are not used are set to all zero.

* SB4002A allocates the Base Address0 or Base Address5 for access of the internal control register (selective). The internal register is 28 bytes, and is in the I/O Space or the Memory Space (selective). The default value of the Base Address0 register is FFFFFFC1h or FFFFFFC0h.

Bit[5:0] is a Read Only bit and Bit[31:6] is a Read/Write bit.

* How to select the I/O Space: The Base Address register and the type of the I/O Space are determined by 00h and 01h of the Serial ROM. If 00h is 0b, the base address is allocated to Base Address0, and if it is 1b, to Base Address5. If 01h is 0b, the base address is allocated to I/O space, and if it is 1b, to the Memory Space.

Subsystem Vendor ID/Subsystem ID Register

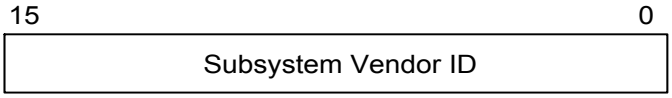


Table 2-18. Subsystem Vendor ID Register Layout

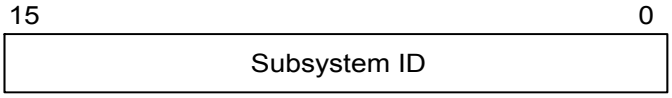


Table 2-19. Subsystem ID Register Layout

Subsystem ID shows information on the manufacturer and the subsystem. The register is used to identify the devices of the same vendor ID and the device ID. The vendor ID and the device ID provide information on the controller chip, and the subsystem vendor ID and the subsystem ID provide information on the PCI card. A vendor ID can be shared by numbers of subsystems. [RO : downloadable]

Capabilities Pointer Register

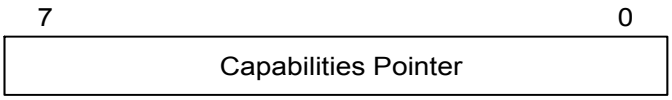


Table 2-20. Capability Pointer Register Layout

The pointer shows the address of the register for new functions of the device. The functions are defined in PCI, and the address designated by the pointer has the ID of the function. The address + 01h provides the address of the register for new functions. It is possible to add functions in this linked list method. If there is no more function to be added, the pointer is set to 00h. [RO : downloadable]

Interrupt Line Register

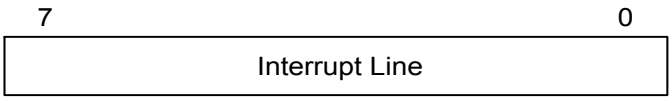


Table 2-21. Interrupt Line Register Layout

This register contains routing information of the interrupt line. This register is set at the system initialization. [R/W : non-downloadable]

Interrupt Pin Register

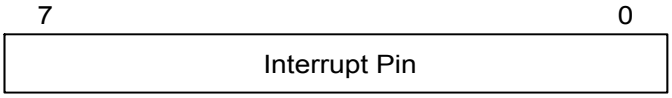


Table 2-22. Interrupt Pin Register Layout

This register indicates the PCI interrupt pin used by the device. Because SB4002A is a single function, the default value is set to 1h via INTA#. [RO : non-downloadable]

MIN_GNT Register

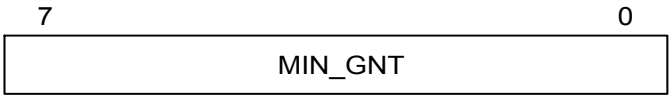


Table 2-23. MIN_GNT Register Layout

This register indicates the time required for burst access. Because SB4002A does not support this register, the default is 00h. [RO : non-downloadable]

MAX_LAT Register

Table 2-24. MAX_LAT Register Layout

This register indicates the frequency of access to the PCI bus. Because SB4002A does not support this register, the default is 00h. [RO : non-downloadable]

2.3 PCI Power Management Interface

In some cases, it is necessary to control the power supply of the PCI bus embedded system. Especially for the mobile system which uses an independent power supply or the PCI device which consumes a large amount of power, it is required to configure a low-power consuming system by limiting power supply when the PCI device is not is operation. The PCI specification defines the power management interface for easier power management.

For the control of power supply in the software level, the power management related registers are located in the configuration I/O space header, and the address can be changed. In SB4002A, the registers are in 40h~47h. In order to apply power management, you need to set the capability pointer to 40h.

SB4002A supports PCI Power Management Interface Specification Revision 1.1. The following tables show the registers contained in the configuration I/O space header.

40h	PMC		Next Item Ptr0	Capability ID0
44h	Data	PMCSR_BSE	PMCSR	

Table 2-25. Power Management Interface Registers

Capability ID0 Register

7	0
Power Management Interface ID	

Table 2-26. Capability ID0 Register for Power Management Interface Layout

Capability ID for the Power Management Interface. The default is 01h. [RO : non-downloadable]

Next Item Point0 Register

7	0
Next Item Pointer0	

Table 2-27. Next Item Pointer0 Register Layout

The pointer indicates the address of the register for the next capability. If there is no next capability, the value must be 00h. [RO : downloadable]

PMC(Power Management Capabilities) Register

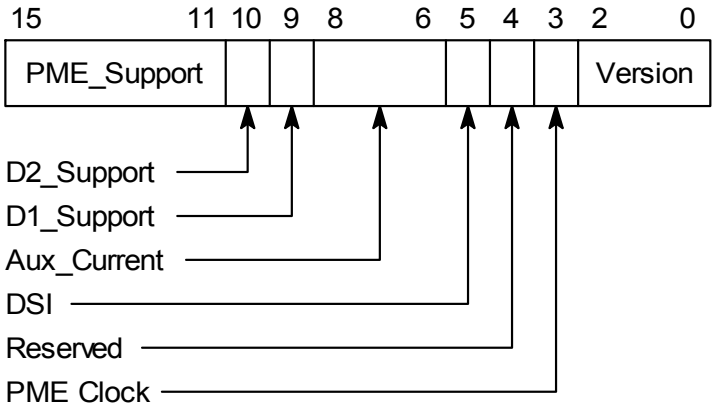


Table 2-28. PMC Register Layout

This register contains information on the power management capability of the device. [RO : non-downloadable]

Bit[15:11] : PME Support. This bit shows the power state in which the device can assert PME#. Bit[11] is allocated to D0, Bit[12] to D1, Bit[13] to D2, Bit[14] to $D3_{hot}$, and Bit[15] to $D3_{cold}$, The value is 1b if the device can assert PME#, or 0b otherwise. Because SB4002A can assert PME# in D0 and $D3_{hot}$ state, the default is 01001b.

Bit[10] : D2_Support. This bit shows if the device supports the D2 power management state. Because SB4002A supports D0 and D3 only, the default value is 0b.

Bit[9] : D1_Support. This bit shows if the device supports the D1 power management state. Because SB4002A supports D0 and D3 only, the default value is 0b.

Bit[8:6] : Aux_Current. This bit shows the current used through 3.3Vaux pin. If the device in $D3_{cold}$ cannot assert PME#, the bit must be set to 000b. Because SB4002A, when in $D3_{cold,}$ cannot assert PME#, the default is 000b.

Bit[5] : DSI. This bit is used by the device which requires a special initialization process before the device driver-triggered initialization. Because SB4002A does not support this process, the default is 0b.

Bit[4] : Reserved.

Bit[3] : PME Clock. This bit indicates if a PCI clock is required for PME# assertion. SB4002 uses the PCI clock, and the default is 1b.

Bit[2:0] : Version. This bit indicates the version of Power Management Interface Specification. Because SB4002 supports the version 1.1, the default is 010b.

PMCSR(Power Management Control/Status Register)

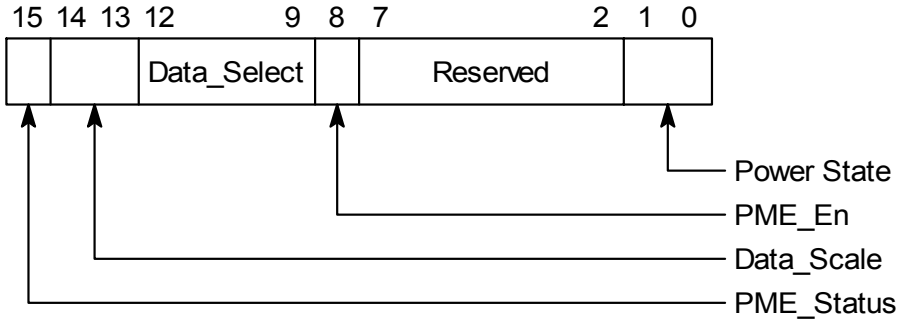


Table 2-29. PMCSR Layout

This register controls power state and PME# of a device. [non-downloadable]

Bit[15] : PME_Status. This bit is set to 1b if PME# is asserted. If this bit is 1b, the register is cleared and the PME# is deasserted. The default is 0b. [R/WC]

Bit[14:13] : Data_Scale. This bit indicates the scale of the data register value. The default is 0b. [RO : downloadable]

Bit[12:9] : Data_Select. This bit designates the value to be read through the data register.

Bit[8] : PME_En. PME# assertion is enabled if this bit is 1b, or disabled if it is 0b. The default is 0b. [R/W]

Bit[1:0] : Power State. This bit shows the current power state. The device is in the D0 state if this bit is 00b, D1 if it is 01b, D2 if it is 10b, and D3 if it is 11b. Because SB4002A supports D0 and D3 only, the values 01b and 10b do not affect the device. The default is 00b.

PMCSR_BSE (PCMSR PCI to PCI Bridge Support Extension) Register

SB4002A does not support this PCI bridge related register. The default is 00h [RO : non-downloadable]

Data Register

This register is used to read the data such as power consumption and dissipation by state. The value to be read is designated and scaled by Data_Select. The following table shows the details. [RO : downloadable]

Data_Scale	Data_Select	Data Register
0 : unknown 1 : x0.1 2 : x0.01 3 : X0.001	0	D0 Power Consumed
	1	D1 Power Consumed
	2	D2 Power Consumed
	3	D3 Power Consumed
	4	D0 Power Dissipated
	5	D1 Power Dissipated
	6	D2 Power Dissipated
	7	D3 Power Dissipated
	8-15	Reserved

Table 2-30. Data Register

PCI bus provides the PME# signal for Power Management Interface. PME# is used to request change of power state of a device. It is an open drain signal which is asynchronous to CLK.

SB4002A provides the power management-related signal to the device in order to support the power management interface in efficient manner. SB4002A provides the input signal PME_REQ and the output signal PME_S. PME_REQ requests PCI bus to change the power state of the device, and PME_S shows the current power state.
If PME_REQ is changed from low to high, the PME# signal is asserted (if PME_En is enabled). PME_S outputs the current power state to the next device. The value is 1b for D0 state, or 1b for D3 state.

2.4 Vital Product Data (VPD)

VPD is a capability register prepared to provide additional information on the system and the device.

VPD is located in the memory devices such as serial EEPROM. The data are accessed through the VPD capability register in the Configuration Space Header. The register addresses can be changed. In SB4002A, the register is in 48h~4Fh. In order to use VPD, you need to set the capability pointer to 48h.

SB4002A adopts the VPD format defined in PCI Specification Revision 2.3. The following figure shows the registers contained in the Configuration Space Header. SB4002A supports read access via VPD, and does not support write access.

48h	F	VPD Address	Next Item Ptr1	Capability ID1
4Ch	VPD Data			

Table 2-31. Vital Product Data Registers

Capability ID1 Register

7	0
Vital Product Data ID	

Table 2-32. Capability ID1 Register Layout for Vital Product Data

Capability ID for the Vital Product Data. The default is 03h. [RO : non-downloadable]

Next Item Pointer1 Register

7	0
Next Item Pointer1	

Table 2-33. Next Item1 Pointer Register Layout

The pointer indicates the address of the register for the next capability. If there is no next capability, the value must be 00h. [RO : downloadable]

VPD Address Register

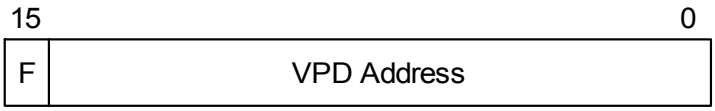


Table 2-34. VPD Address Register Layout

Bit[15] F. This bit indicates that data transfer is finished between the VPD data register and the memory. For a read event, if 0b is written on F for the address, the device sets F when the data is prepared. The default is 0b. [R/W : non-downloadable]

Bit[14:0] VPD Address. This bit indicates the address of VPD to be accessed by DWORD. The default is 00h. [R/W : non-downloadable]

VPD Data Register

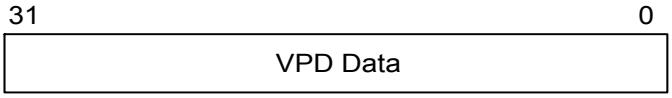


Table 2-35. VPD Data Register Layout

The Configuration Space on which the device reads the VPD Data. The default is 0000h. [R/W : non-downloadable]

In SB4002A, the VPD data are stored in the external serial EEPROM. Because this serial EEPROM is also used to set the Configuration Space Header at system booting, this space for the Configuration Space Header must not be overlapped with the space for VPD data. Up to the address 30h is for the Configuration Space Header setting, and the remaining parts can be used for VPD.

2.5 CompactPCI Hot Swap

CompactPCI is the specification made by applying the existing specification of PCI Local Bus for general PC to the industrial PC. The algorithms such as bus protocol are same as those of the existing PCI, except for the electrical features and the mechanical features. Hot Swap is one of the functions that reinforce the industrial aspect of CompactPCI. The function enables you to delete/insert the CompactPCI board while power is connected to the system.

CompactPCI hot swap capability supports the software aspect of the Hot Swap Function. The control register is in the Configuration Space Header. In SB4002A, the control register is in 50h~53h.

SB4002A follows the CompactPCI Hot Swap Specification 1.0. The following table shows the registers contained in the Configuration Space Header.

50h	Reserved	HS_CSR	Next Item Ptr2	Capability ID2

Table 2-36. CompactPCI Hot Swap Registers

Capability ID2 Register

7	0
CompactPCI Hot Swap ID	

Table 2-37. Capability ID2 Register Layout for CompactPCI Hot Swap

Capability ID for CompactPCI Hot Swap. The default is 06h. [RO : non-downloadable]

Next Item Poinert2 Register

7	0
Next Item Pointer2	

Table 2-38. Next Item Pointer2 Register Layout

The pointer indicates the address of the register for the next capability. Because this is the last capability supported by SB4002A, the default is 00h. [RO : downloadable]

HS_CSR(Hot Swap Control and Status Register)

7	6	5	4	3	2	1	0
INS	EXT	PI		LOO	PIE	EIM	DHA

Table 2-39. HS_CSR Layout

The register enables the host to adjust the hot swap process. [non-downloadable]

Bit[7] : INS. ENUM# Status - Insertion. This bit indicates that ENUM# is asserted during the insert process. If the value is 1b, the bit is cleared and ENUM# is deasserted. The default is 0b. [R/WC]

Bit[6] : EXT. ENUM# Status - Ejection. This bit indicates that ENUM# is asserted if the value is 1b, the bit is cleared and ENUM# is deasserted. The default is 0b. [R/WC]

Bit[5:4] : PI. Programming Interface. This bit shows information on the supporting programming interface. If 00b, the device supports EIM, LOO, INS and EXT only, and if 01b, it also supports Device Hiding, DHA and PIE. Because SB4002A supports the basic functions only, the default is 00b. [RO]

Bit[3] : LOO. LED ON/OFF. LED is on if the value is 1b, or off if it is 0b. The default is 0b. [R/W]

Bit[2] : PIE. Pending Insertion/Extraction. This bit indicates if insertion or extraction process is in progress. Because PI is 0b, the default is 0b. [RO]

Bit[1] : EIM. ENUM# Interrupt Mask. Mask is executed if the value is 1b, and enabled if it is 0 . The default is 0b. [R/W]

Bit[0] : DHA. Device Hiding Arm. This bit supports device hiding. Because PI is 0b in SB4002A, the default is 0b. [RO]

PCI bus supports the ENUM# signal for CompactPCI Hot Swap. This signal asserts ENUM# when a CompactPCI is inserted or rejected for appropriate action.

SB4002A provides HSW and HS_LED signal to support CompactPCI Hot Swap in efficient manner. HSW, located on the front panel of the CompactPCI board, is an input signal that indicates insert/reject of the board by switching on/off after the board is inserted or before the board is rejected. HSW from 0b to 1b indicates that the board is inserted, and from 1b to 0b indicates the board is about to reject. If HSW changes from 0b to 1b, Bit7 of HS_CSR is set, and ENUM# is asserted. If HSW changes from 1b to 0b, Bit 6 of HS_CSR is set, and ENUM# is asserted. HS_LED is the signal that makes the Bit3 value of HS_CSR, and is used to control the external LED.

2.6 Exclusive Access

PCI bus provides the method that the master can exclusively access a specific target. This method is called an exclusive access. It is started by LOCK#, one of the PCI signals.

The first transaction to start an excusive access for a specific target must be the memory READ transaction. If this transaction is started, LOCK# must be deasserted at the first CLK of the address phase, and LOCK# must be deasserted at the next CLK. An exclusive access is started when the following data phase is completed. Once an exclusive access is started, the master must maintain LOCK# in the assert state.

Once an exclusive access is started, in order that the master that has requested exclusive access can re-access the target, the master must deassert LOCK# at the first CLK of the address phase, and then, assert LOCK# at the next CLK. In order that the exclusive access is maintained after the completion of the last data phase, LOCK# must be in the assert state. In order that the exclusive access is released, LOCK# must be deasserted.

In order that the target should reject an exclusive access because it does not support exclusive access or due to any internal reason, the target must terminate the first transaction with retry.

No other master than the one for the exclusive access can access the target because it cannot deassert LOCK# at the address phase. If any other master attempts access, the target terminates the transaction with retry.

Once an exclusive access is started, it is maintained even if retry of transaction or disconnect occurs between the master and the target. If a target abort or master abort occurs, the exclusive access is released.

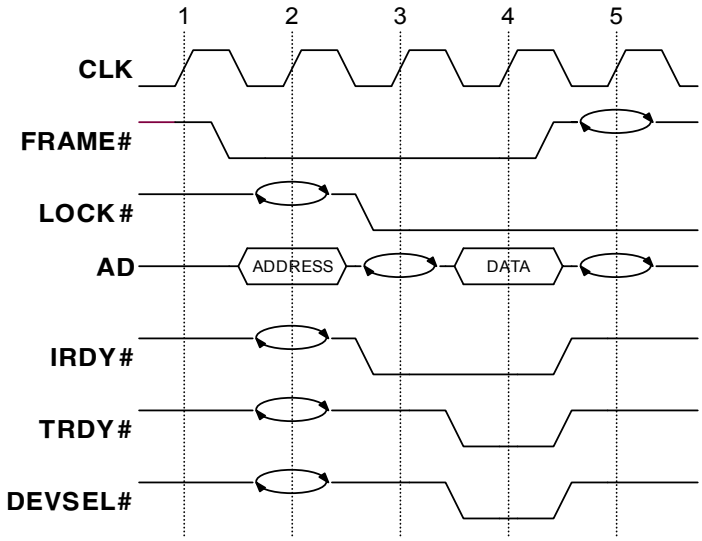


Figure 2-9. Starting an Exclusive Access

In the above figure, FRAME# is asserted at CLK 2 and the address phase is started. As LOCK# is asserted, if the PCI command is Memory Read, the exclusive access request transaction is executed. As LOCK# is asserted at CLK 3 and the data phase is terminated normally at CLK 4, an exclusive access is started.

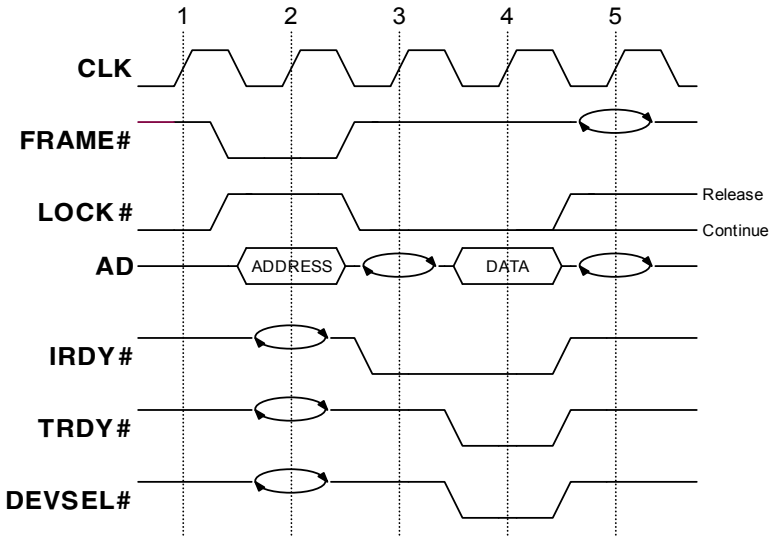


Figure 2-10. Continuing or Releasing Exclusive Access

The above figure shows the state that an exclusive access has been already started. The figure shows the transaction that the master that started the exclusive access accesses the target as LOCK# is deasserted at CLK 2. At CLK 3, LOCK# is asserted for the transaction. At CLK4, data transfer occurs and the transaction is terminated. The exclusive access continues if LOCK# is kept asserted, or is released if LOCK# is deasserted.

3. Legacy Bus

SB4002A provides the ISA like Legacy bus at the back end, in order that it can interface with general devices via interworking with PCI bus. SB4002A provides the chip select signal (CSx#), I/O read/write(IOR#, IOW#) and memory read/write#(MEMR#, MEMW#) signal to support the general devices.
SB4002A also supports a variety of flexible accessing methods.

3.1 Real Access/Delayed Access

3.1.1 Real Access

This method provides the real time data transfer by executing the PCI transaction and the Legacy bus transaction at the same time.
At the same time it is hit, the device sends the address signal, the chip select signal and the R/W signal, and then, depending on R/W, outputs the data on the PCI bus to the Legacy bus, or vice versa.
For the write process, because it is not necessary that the transaction on the PCI bus is extended until the transaction on the Legacy bus is completed, the device completes the transaction on the PCI bus as soon as possible, stores the write data on the buffer, and starts the transaction on the Legacy bus.
For the read process, because the device can send the valid data to the PCI bus at the time the transaction on the Legacy bus is completed, a wait cycle must be inserted to the PCI bus.

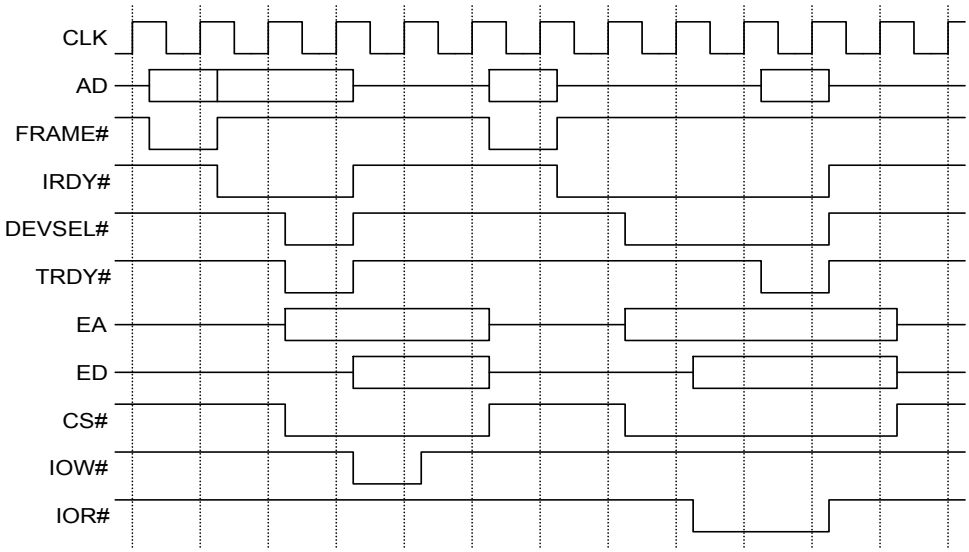


Figure 3-1. Legacy Bus Real Access

3.1.2 Delayed Access

This method transmits data through the transaction on the Legacy bus after the device completes the transaction on the PCI bus. If the timing of the device is slow or the burst size is too big, data transfer in the real access method may seize the PCI bus for a long time, and deteriorate the overall system efficiency.

For the write process, the device finishes the PCI bus transaction at the shortest time without wait cycle, stores the address and data in FIFO, and starts the Legacy bus transaction. This doesn't seem much different from the real access method, but will provide the advantage over the real access method in burst access and data byte split.

For the read process, the device stores the address, command and byte enable in the buffer, and terminates the transaction with retry. Then, the device starts the read transaction independently on the Legacy bus, and stores the read data in FIFO memory. When the transaction is restarted on the PCI bus, the device outputs the data from the FIFO memory.

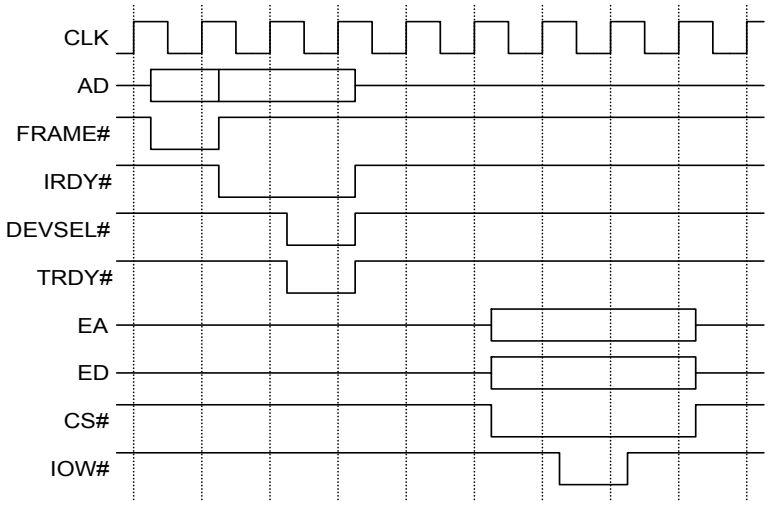


Figure 3-2. Legacy Bus Delayed Write Access

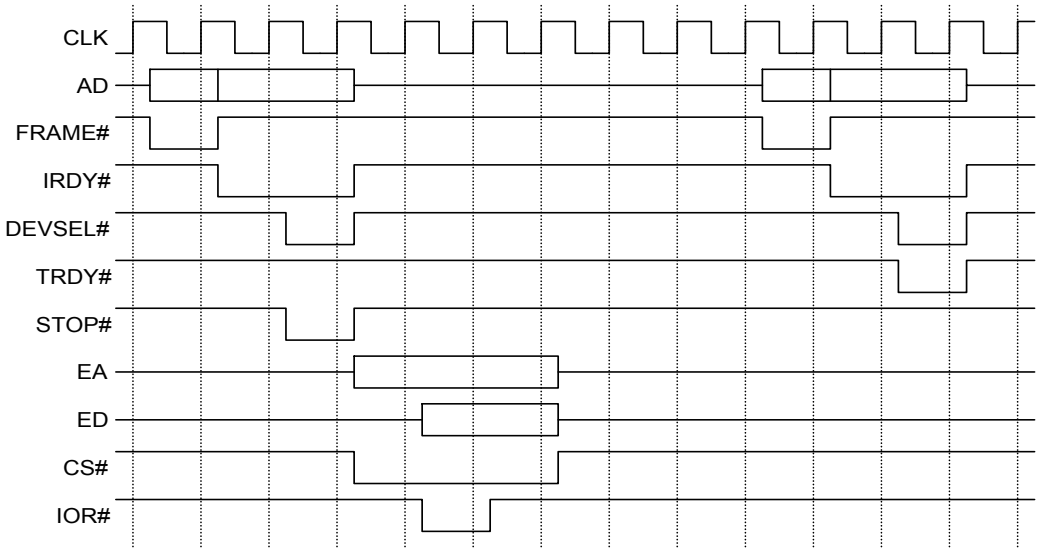


Figure 3-3. Legacy Bus Delayed Read Access

3.2 Single Access/Burst Access

Single and burst access on the Legacy bus are the signal access / burst access on the PCI bus. In other words, the single read/write on the PCI bus is transmitted to the single read/write on the Legacy bus, and the burst read/write on the PCI bus is transmitted to the burst read/write on the Legacy bus. For burst access, the chip select signal is asserted in the entire course of a transaction, and the read/write signals repeat assert/deassert by a single transfer.

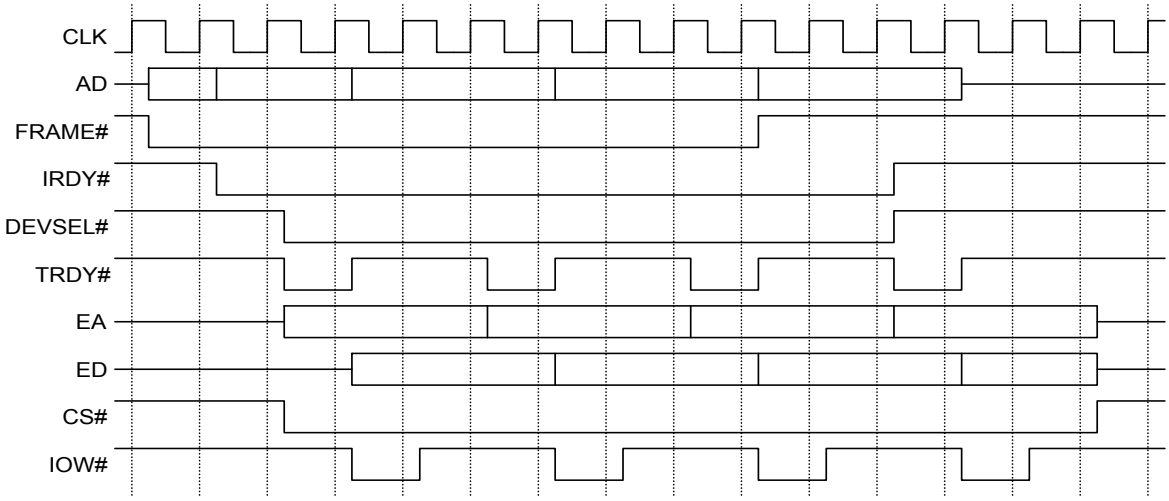


Figure 3-4. Legacy Bus Real Write Burst Access

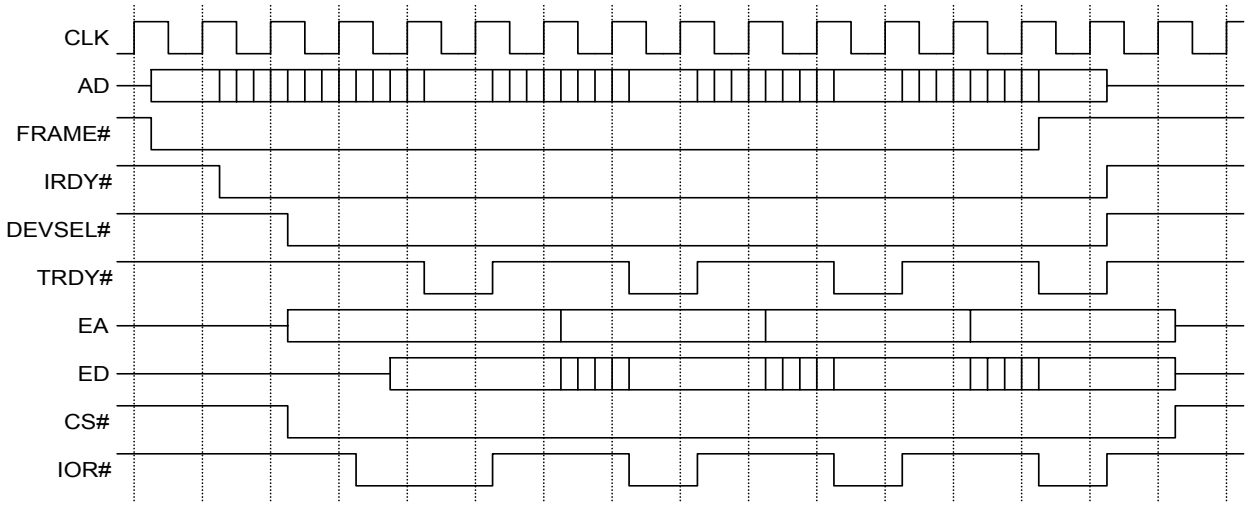


Figure 3-5. Legacy Bus Real Read Burst Access

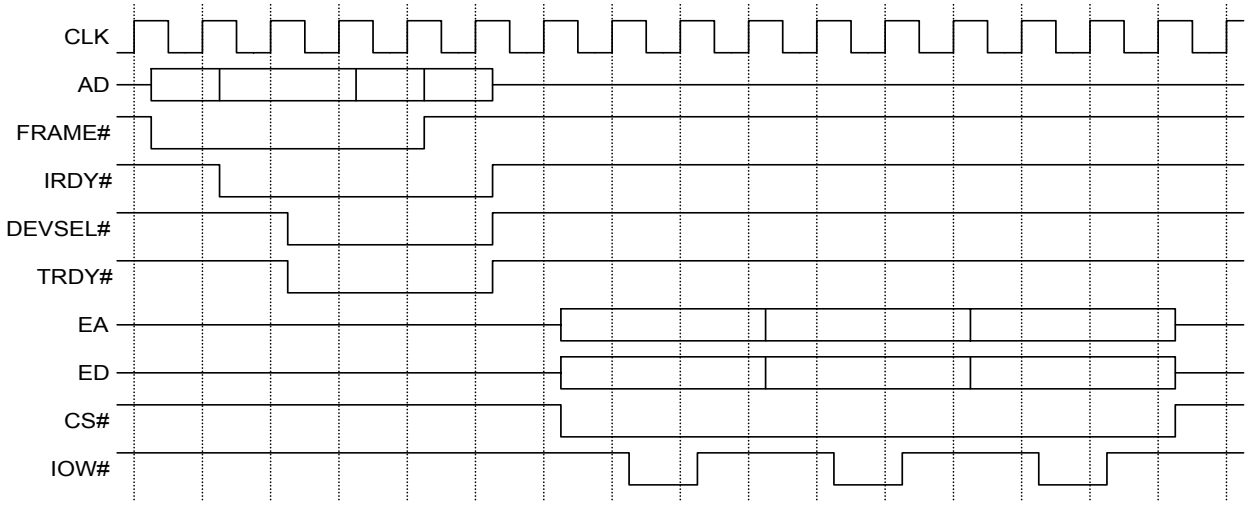


Figure 3-6. Legacy Bus Delayed Write Burst Access

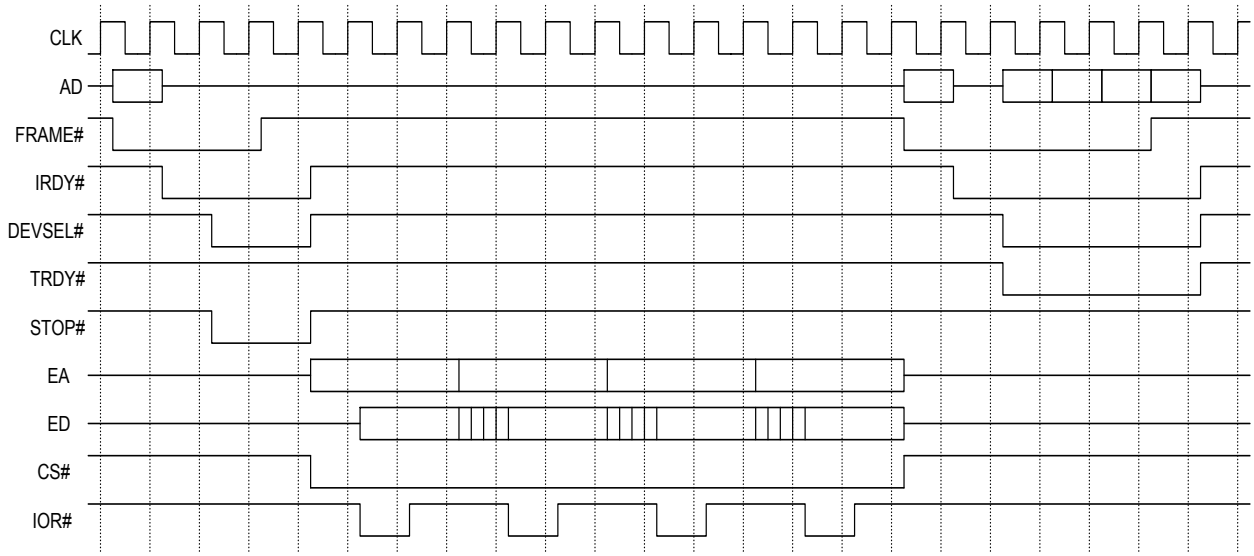


Figure 3-7. Legacy Bus Delayed Read Burst Access

Single and burst in a delayed access are depending on whether it is a write access or a read access.

In case of a delayed write access, a burst access is executed on the PCI bus, and the data are stored in the FIFO memory. (Maximum 16 bytes. If the access request exceeds the limit, the device terminates the transaction with disconnect.) Then the device executes the write burst access on the Legacy bus after the PCI bus access.

In case of a delayed read access, the device predefines the burst transfer size on the Legacy bus. If the PCI bus access is requested, the device terminates the transaction with retry. Then, the device executes the burst access to the designated size (maximum 16 DWORD) on the Legacy bus, and stores the data in FIFO. This process is called as a prefetch. If the transaction is restarted on the PCI bus, the device transmits the data from FIFO. In this case, it is impossible to know how much data the PCI bus will request. The PCI bus may request more or less data than in the memory. In the former case, the device may terminate the transaction with Disconnect. And in the latter case, the data stored in FIFO become invalid. Because the data may be lost in the latter case, the device must be prefetchable, which means that the read access must not affect the address space.

3.3 Data Split and Combine

SB4002A supports the data path in BYTE, WORD and DWORD on the Legacy bus depending on the characteristics of the device. The PCI bus data path is based on DWORD, and access in byte and word is also available. If the supported data path type is not consistent between the PCI bus access and the Legacy bus device - e.g., the PCI bus access is DWORD or WORD, while the Legacy bus device is a BYTE device; or the PCI bus access is DWORD, while the Legacy bus device is a WORD device - the PCI bus can terminate the transaction with Target Abort. SB4002A, however, adopts the concept of Data Split and Combine to complete the transaction in the above case. This concept is applied to the delayed access only.

For the write process, the device stores the data in FIFO, and when the Legacy bus starts the write access, starts burst access to the bandwidth of the Legacy bus device. For example, if the Legacy bus device is a BYTE device, and a DWORD data is stored in FIFO, the Legacy bus device executes access 4 times by dividing the data into 4 sets of BYTE data.

For the read process, the device executes burst access to the data path of the Legacy bus device, and stores the data in FIFO. For example, if the Legacy bus device is a BYTE device, the Legacy bus device executes BYTE access 4 times, stores a set of DWORD data, and transmits the data when the PCI bus starts a transaction.

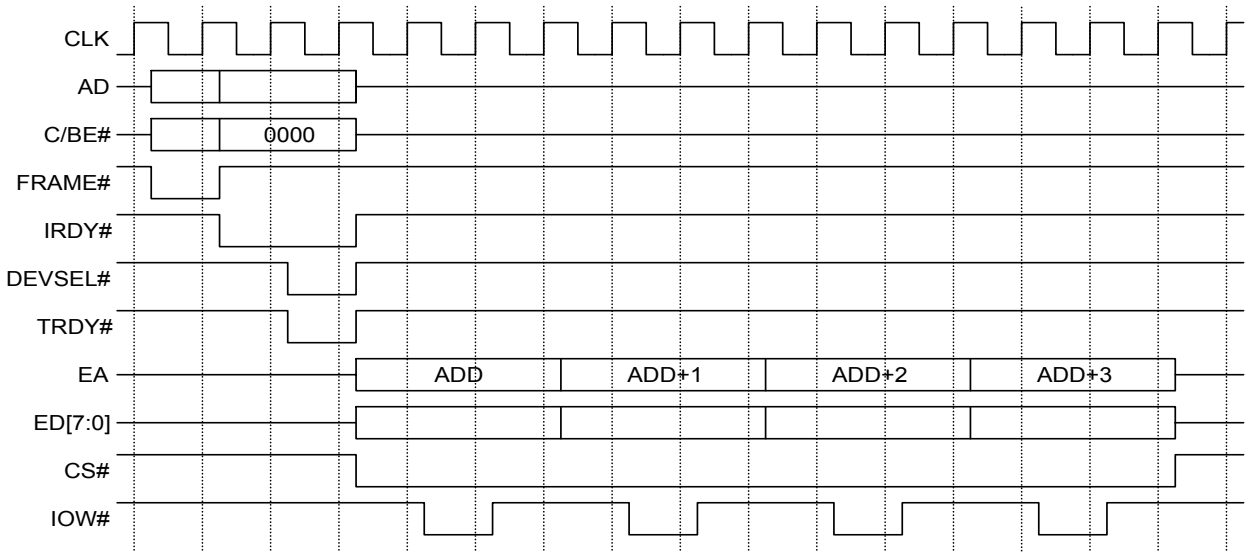


Figure 3-8. Legacy Bus DWORD Delayed Write Split Access for Byte Device

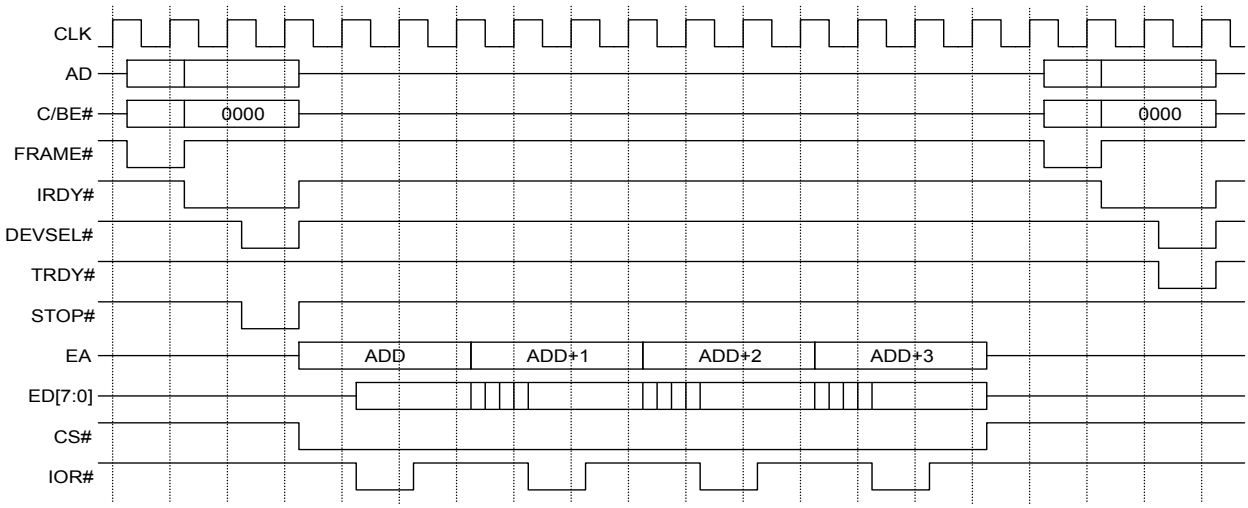


Figure 3-9. Legacy Bus DWORD Delayed Read Split Access for Byte Device

3.4 Endian Conversion

The data are divided into Little Endian and Big Endian depending on the order or BYTE lanes in the system structure.

In case of a DWORD system with the [31:0] data path, in Little Endian, [31:24], [23:16], [15:8] and [7:0] are the fourth, third, second and first byte, respectively. In Big Endian, [7:0], [15:8] and [23:16] are the fourth, third and fourth byte, respectively.

Basically, PCI bus is based on Little Endian. However, because the device connected to the Legacy bus can be the device that supports Little Endian, SB4002A supports conversion of the Little Endian data into the Big Endian data. The following tables show the conversion methods.

- DWORD Data path

Byte Lane	BYTE3	BYTE2	BYTE2	BYTE0
Little Endian PCI bus	AD[31:24]	AD[23:16]	AD[15:8]	AD[7:0]
Little Endian Lagacy bus	ED[31:24]	ED[23:16]	ED[15:8]	ED[7:0]
Big Endian Lagacy bus	ED[7:0]	ED[15:8]	ED[23:16]	ED[31:24]

Table 3-1. DWORD Data path Endian Conversion

- WORD Data path

Byte Lane	BYTE3	BYTE2	BYTE2	BYTE0
Little Endian PCI bus	AD[31:24]	AD[23:16]	AD[15:8]	AD[7:0]
Little Endian Lagacy bus	ED[15:8]	ED[7:0]	ED[15:8]	ED[7:0]
Big Endian Lagacy bus	ED[7:0]	ED[15:8]	ED[7:0]	ED[15:8]

Table 3-2. WORD Data path Endian Conversion

- BYTE Data path

Byte Lane	BYTE3	BYTE2	BYTE2	BYTE0
Little Endian PCI bus	AD[31:24]	AD[23:16]	AD[15:8]	AD[7:0]
Little Endian Lagacy bus	ED[7:0]	ED[7:0]	ED[7:0]	ED[7:0]
Big Endian Lagacy bus	ED[7:0]	ED[7:0]	ED[7:0]	ED[7:0]

Table 3-3. BYTE Data path Endian Conversion

4. Configuration Data in Serial ROM

The PCI Configuration Space Header is an important part of the PCI bus. The Header contains information on the product and the resources, which is a unique value for each product. Because the Header must contain unique information by product, SB4002A reads out information from the external serial ROM. SystemBase adopts ATMEL's 1K 3-wire Serial EEPROM AT93C46 as the external ROM. Any other ROM which is equivalent to AT93C46 can be used.

The following table describes the address map of the serial ROM.

Address	Description	Note
0h	Control Register Setting0	Select BASE0(0) or BASE5(1)
1h	Control Register Setting1	Select I/O(0) or Memory(1)
2h	Vendor ID Low byte	
3h	Vendor ID High byte	
4h	Device ID Low byte	
5h	Device ID High byte	
6h	Status Register Capabilities List Enable	Bit4 is set to 1. Disregards other bits.
7h	Reserved	
8h	Revision ID	
9h	Class Code Low byte	
Ah	Class Code Middle byte	
Bh	Class Code High byte	
Ch	Base Address1 1st byte	
Dh	Base Address1 2nd byte	
Eh	Base Address1 3rd byte	
Fh	Base Address1 4th byte	
10h	Base Address2 1st byte	
11h	Base Address2 2nd byte	
12h	Base Address2 3rd byte	
13h	Base Address2 4th byte	
14h	Base Address3 1st byte	

Address	Description	Note
15h	Base Address3 2nd byte	
16h	Base Address3 3rd byte	
17h	Base Address3 4th byte	
18h	Base Address4 1st byte	
19h	Base Address4 2nd byte	
1Ah	Base Address4 3rd byte	
1Bh	Base Address4 4th byte	
1Ch	Base Address5/Base Address0 1st byte	
1Dh	Base Address5/Base Address0 2nd byte	
1Eh	Base Address5/Base Address0 3rd byte	
1Fh	Base Address5/Base Address0 4th byte	
20h	Subvendor ID Low byte	
21h	Subvendor ID High byte	
22h	Subsystem ID Low byte	
23h	Subsystem ID High byte	
24h	EXROM Base Address 1st byte	
25h	EXROM Base Address 2nd byte	
26h	EXROM Base Address 3rd byte	
27h	EXROM Base Address 4th byte	
28h	Capability Pointer	
29h	Next PTR0	
2Ah	PMC Data Scale	
2Bh	PMC Data0	
2Ch	PMC Data3	
2Dh	PMC Data4	
2Eh	PMC Data7	
2Fh	NEXT PTR1	
30h~	Reserved	

Table 4-1. Serial ROM Address Map

00h~2Fh of the serial ROM are used to set the PCI Configuration Space Header. At the system reset, the device automatically reads this part from the serial ROM and sets the Configuration Space Header.

You can use 30h~7Fh at your own discreet. Because you can read/write data through the serial ROM interface register in the control register, you can easily store data without seizing other resources.

Serial ROM Address(00h) : Control Register Setting0 Byte

Selects the Base Address0 or Base Address5 in the PCI Configuration Space for the SB4002A control register.

If the 00h data is stored in the serial ROM 00h, the SB4002A control register is allocated to the Base Address0, and if the 01h is stored, the SB4002A control register is allocated to the Base Address5.

Serial ROM Address(01h) : Control Register Setting1 Byte

Determines whether the SB4002A Control Registers should be allocated to the I/O Space or the Memory Space. If the 00h data is stored in the serial ROM 00h, the SB4002A Control Register is allocated to the I/O Space, and if the 01h is stored, the SB4002A Control Register is allocated to the Memory Space.

Serial ROM Address(02h) : Vendor ID Low Byte

Serial ROM Address(02h) represents the bit array [7:0] of Vendor ID[15:0] of PCI Configuration Space 00h~01h.

Serial ROM Address(03h) : Vendor ID High Byte

Serial ROM Address(03h) represents the bit array [15:8] of Vendor ID[15:0] of PCI Configuration Space 00h~01h.

Serial ROM Address(04h) : Device ID Low Byte

Serial ROM Address(04h) represents the bit array [7:0] of Vendor ID[15:0] of PCI Configuration Space 02h~03h.

Serial ROM Address(05h) : Device ID High Byte

Serial ROM Address(05h) represents the bit array [15:8] of Vendor ID[15:0] of PCI Configuration Space 02h~03h.

Serial ROM Address (06h) : Status Register Capabilities List Enable

The PCI spec from 2.1 provides a new register set which is a linked list called as the Capabilities List. In order to enable the Capabilities List, the Capabilities List bit (bit 4) of the PCI Status Register must be set to 1, and the Capabilities List pointer must be in the PCI Configuration Space 34h. In 34h, the pointer indicates the first item of the Capabilities List.

In order to use the Capabilities List, store 10h in 06h of the serial ROM. Then 1b is stored in the PCI Status Register bit 4 of PCI Configuration Space 06h~07h. In order not to use the Capabilities List, store 00h. Then 0b is stored in the PCI Status Register bit 4.

Serial ROM Address(07h) : Reserved

Serial ROM Address 07h is reserved for 00h data.

Serial ROM Address(08h) : Revision ID

Serial ROM Address 08h represents Revision ID[7:0] of PCI Configuration Space 08h.

Serial ROM Address(09h) : Class Code Low Byte

Serial ROM Address 09h represents the bit array [7:0] of Class Code [23:0] of PCI Configuration Space 09h~0Bh.

Serial ROM Address(0Ah) : Class Code Middle Byte

Serial ROM Address 0Ah represents the bit array [15:8] of Class Code [23:0] of PCI Configuration Space 09h~0Bh.

Serial ROM Address(0Bh) : Class Code High Byte

Serial ROM Address 0Bh represents the bit array [23:16] of Class Code [23:0] of PCI Configuration Space 09h~0Bh.

Serial ROM Address(0Ch) : Base Address1 1st Byte

Serial ROM Address 0Ch represents the first bit array [7:0] of Base Address1[31:0] in PCI Configuration Space 14h~17h.

Serial ROM Address(0Dh) : Base Address1 2nd Byte

Serial ROM Address 0Dh represents the second bit array [15:8] of Base Address1[31:0] in PCI Configuration Space 14h~17h.

Serial ROM Address(0Eh) : Base Address1 3rd Byte

Serial ROM Address 0Eh represents the third bit array [23:16] of Base Address1[31:0] in PCI Configuration Space 14h~17h.

Serial ROM Address(0Fh) : Base Address1 4th Byte

Serial ROM Address 0Fh represents the fourth bit array [31:24] of Base Address1[31:0] in PCI Configuration Space 14h~17h.

Serial ROM Address(10h) : Base Address2 1st Byte

Serial ROM Address 10h represents the first bit array [7:0] of Base Address2[31:0] in PCI Configuration Space 18h~1Bh.

Serial ROM Address(11h) : Base Address2 2nd Byte

Serial ROM Address 11h represents the second bit array [15:8] of Base Address2[31:0] in PCI Configuration Space 18h~1Bh.

Serial ROM Address(12h) : Base Address2 3rd Byte

Serial ROM Address 12h represents the second bit array [23:16] of Base Address2[31:0] in PCI Configuration Space 18h~1Bh.

Serial ROM Address(13h) : Base Address2 4th Byte

Serial ROM Address 13h represents the fourth bit array [31:24] of Base Address2[31:0] in PCI Configuration Space 18h~1Bh.

Serial ROM Address(14h) : Base Address3 1st Byte

Serial ROM Address 14h represents the first bit array [7:0] of Base Address3[31:0] in PCI Configuration Space 1Ch~1Fh.

Serial ROM Address(15h) : Base Address3 2nd Byte

Serial ROM Address 15h represents the second bit array [15:8] of Base Address3[31:0] in PCI Configuration Space 1Ch~1Fh.

Serial ROM Address(16h) : Base Address3 3rd Byte

Serial ROM Address 16h represents the third bit array [23:16] of Base Address3[31:0] in PCI Configuration Space 1Ch~1Fh.

Serial ROM Address(17h) : Base Address3 4th Byte

Serial ROM Address 17h represents the fourth bit array [31:24] of Base Address3[31:0] in PCI Configuration Space 1Ch~1Fh.

Serial ROM Address(18h) : Base Address4 1st Byte

Serial ROM Address 18h represents the first bit array [7:0] of Base Address4[31:0] in PCI Configuration Space 20h~23h.

Serial ROM Address(19h) : Base Address4 2nd Byte

Serial ROM Address 19h represents the second bit array [15:8] of Base Address4[31:0] in PCI Configuration Space 20h~23h.

Serial ROM Address(1Ah) : Base Address4 3rd Byte

Serial ROM Address 1Ah represents the third bit array [23:16] of Base Address4[31:0] in PCI Configuration Space 20h~23h.

Serial ROM Address(1Bh) : Base Address4 4th Byte

Serial ROM Address 1Bh represents the fourth bit array [31:24] of Base Address4[31:0] in PCI Configuration Space 20h~23h.

Serial ROM Address(1Ch) : Base Address5/Base Address0 1st Byte

If the Control Register Setting0 byte of the Serial ROM Address 00h is 00h, Serial ROM Address (1Ch) represents the bit array [7:0] of Base Address5[31:0] in PCI Configuration Space 24h~27h.

If the Control Register Setting0 byte of the Serial ROM Address 00h is 01h, Serial ROM Address (1Ch) represents the bit array [7:0] of Base Address0[31:0] in PCI Configuration Space 10h~13h.

Serial ROM Address(1Dh) : Base Address5/Base Address0 2nd Byte

If the Control Register Setting0 byte of the Serial ROM Address 00h is 00h, Serial ROM Address (1Dh) represents the bit array [15:8] of Base Address5[31:0] in PCI Configuration Space 24h~27h.

If the Control Register Setting0 byte of the Serial ROM Address 00h is 01h, Serial ROM Address (1Dh) represents the bit array [15:8] of Base Address0[31:0] in PCI Configuration Space 10h~13h.

Serial ROM Address(1Eh) : **Base Address5/Base Address0 3rd Byte**

If the Control Register Setting0 byte of the Serial ROM Address 00h is 00h, Serial ROM Address (1Eh) represents the bit array [23:16] of Base Address5[31:0] in PCI Configuration Space 24h~27h.

If the Control Register Setting0 byte of the Serial ROM Address 00h is 01h, Serial ROM Address (1Eh) represents the bit array [23:16] of Base Address0[31:0] in PCI Configuration Space 10h~13h.

Serial ROM Address(1Fh) : Base Address5/Base Address0 4th Byte

If the Control Register Setting0 byte of the Serial ROM Address 00h is 00h, Serial ROM Address (1Fh) represents the bit array [31:24] of Base Address5[31:0] in PCI Configuration Space 24h~27h.

If the Control Register Setting0 byte of the Serial ROM Address 00h is 01h, Serial ROM Address (1Fh) represents the bit array [31:24] of Base Address0[31:0] in PCI Configuration Space 10h~13h.

Serial ROM Address(20h) : Subsystem Vendor ID Low Byte

Serial ROM Address(20h) represents the bit array [7:0] of Subsystem Vendor ID[15:0] of PCI Configuration Space 2Ch~2Dh.

Serial ROM Address(21h) : Subsystem Vendor ID High Byte

Serial ROM Address(21h) represents the bit array [15:8] of Subsystem Vendor ID[15:0] of PCI Configuration Space 2Ch~2Dh.

Serial ROM Address(22h) : Subsystem ID Low Byte

Serial ROM Address(22h) represents the bit array [7:0] of Subsystem ID[15:0] of PCI Configuration Space 2Eh~2Fh.

Serial ROM Address(23h) : Subsystem ID High Byte

Serial ROM Address(23h) represents the bit array [15:8] of Subsystem ID[15:0] of PCI Configuration Space 2Eh~2Fh.

Serial ROM Address(24h) : EXROM Base Address 1st Byte

Serial ROM Address(24h) represents the bit array [7:0] of Expansion ROM Base Address[31:0] of PCI Configuration Space 30h~33h.

Serial ROM Address(25h) : EXROM Base Address 2nd Byte

Serial ROM Address(25h) represents the bit array [15:8] of Expansion ROM Base Address[31:0] of PCI Configuration Space 30h~33h.

Serial ROM Address(26h) : EXROM Base Address 3rd Byte

Serial ROM Address(26h) represents the bit array [23:16] of Expansion ROM Base Address[31:0] of PCI Configuration Space 30h~33h.

Serial ROM Address(27h) : EXROM Base Address 4th Byte

Serial ROM Address(27h) represents the bit array [31:24] of Expansion ROM Base Address[31:0] of PCI Configuration Space 30h~33h.

Serial ROM Address(28h) : Capabilities Pointer

Serial ROM Address(28h) represents Capabilities Pointer [7:0].

SB4002A provides 3 types of Capability; PCI Power Management Interface(Capability ID = 01h), VPD(Capability ID = 03h) and CompactPCI How Swap(Capability ID = 06h). PCI Spec 3.0 defines total 17 types of Capabilities, and SB4002A supports only 3 types.

The PCI Power Management Interface Capability List is located in 40h~47h of the PCI Configuration Space, and the VPD Capability List is located in 48h~4Fh of the PCI Configuration Space.

The CompactPCI Hot Swap Capability List is in 50h~53h.

If you wish to use the 3 types of Capability above, store 40h in the PCI Configuration Space 34h. Then the PCI Power Management Interface Capability is enabled. Store 48h in 41h, then, the VPD Capability is enabled.

Store 50h in 49h to enable CompactPCI Hot Swap Capability. In 51h, because no more Capability is supported, 00h is stored.

If you want to use the PCI Power Management Interface Capability and the CompactPCI Hot Swap Capability only, store 40h and 50h in 34h and 41h, respectively, of the PCI Configuration Space.

If you want to use the PCI Power Management Interface Capability only, store 40h and 00h in 34h and 41h, respectively, of the PCI Configuration Space.

If you want to use the VPD Capability only, store 48h and 00h in 34h and 49h, respectively, of the PCI Configuration Space.

If you want to use the CompactPCI Hot Swap Capability only, store 50h in 34h of the PCI Configuration Space.
If 00h is stored in 34h of the PCI Configuration Space, the three Capabilities are disabled.

Serial ROM Address(29h) : NEXT PTR0

Serial ROM Address(29h) represents the Next Item Pointer of PCI Power Management Interface Capability List.
This is used to use the VPD Capability or the CompactPCI Hot Swap Capability other than the PCI Power Management Interface Capability.
If 48h is stored, the VPD Capability is enabled, and if 50h is stored, the CompactPCI Hot Swap Capability is enabled.
If 00h is stored, no more Capability is enabled.

Serial ROM Address(2Ah) : PMC Data Scale

Serial ROM Address(2Ah) is used to specify the Data Scale value(PMCSR[14:13]) of PMCSR which is located in PCI Configuration Space 44h.
The bit designates the scale of data to be read through the Data Register of the PCI Power Management Interface Capability. 00h indicates the unknown scale, and 20h, 40h and 60h indicate the data scale of x0.1, x0.01 and x0.001, respectively. (See 2.3 PCI Power Management Interface for further details.)
00h, 20h, 40h and 60h must be used to designate 00h, 01h, 02h and 03h, respectively.

Serial ROM Address(2Bh) : PMC Data 0

Serial ROM Address(2Bh) is used to specify the reading value from Data Register in case Data Select bits(PMCSR[12:9]) of PMCSR equal to 0h. In other words, it indicates the D0 State Power Consumed.
The data register is located in 47h of the PCI Configuration Space. (See 2.3 PCI Power Management Interface for further details.)

Serial ROM Address(2Ch) : PMC Data 3

Serial ROM Address(2Ch) is used to specify the reading value from Data Register in case Data Select bits(PMCSE[12:9]) of PMCSR equal to 3h. In other words, it indicates the D3 State Power Consumed.

The Data Register is located in 47h of the PCI Configuration Space. (See 2.3 PCI Power Management Interface for further details.)

Serial ROM Address(2Dh) : PMC Data 4

Serial ROM Address(2Dh) is used to specify the reading value from Data Register in case Data Select bits(PMCSE[12:9]) of PMCSR equal to 4h. In other words, it indicates the D0 State Power Dissipated.

The Data Register is located in 47h of the PCI Configuration Space. (See 2.3 PCI Power Management Interface for further details.)

Serial ROM Address(2Eh) : PMC Data 7

Serial ROM Address(2Eh) is used to specify the reading value from Data Register in case Data Select bits(PMCSE[12:9]) of PMCSR equal to 7h. In other words, it indicates the D3 State Power Dissipated.

The Data Register is located in 47h of the PCI Configuration Dpace. (See 2.3 PCI Power Management Interface for further details.)

Serial ROM Address (2Fh) : NEXT PTR1

Serial ROM Address(2Fh) represents Next Item Pointer of VPD Capability List.

This is used to use the PCI Power Management Interface Capability or the CompactPCI Hot Swap Capability other than the VPD Capability.

If 40h is stored, the PCI Power Management Capability is enabled, and if 50h is stored, the CompactPCI Hot Swap Capability is enabled.

If 00h is stored, no more Capability is enabled.

Note : The Next Item Pointer in the CompactPCI Hot Swap Capability List is fixed to 00h. It is because the CompactPCI Hot Swap Capability is the last Capability provided by SB4002A, and that the user doesn't need to handle it unlike the Next Item Pointers of other Capabilities.

Serial ROM Address(30h~) : Reserved

Serial ROM Address 30h~ is reserved. You have to store the 00h data.

5. Miscellaneous Feature

5.1 Interrupt

In order to enable the PCI interrupt service, SB4002A provides 5 interrupt pins to the Legacy bus. In general, the I/O devices use the interrupt and provide total 5 base addresses, SB4002A provides 5 interrupt pin sources. However, the interrupt pins can be used regardless of the address. For example, an I/O base address can use 2 or more interrupt pins.

Because the interrupt features vary by the devices, SB4002A provides the interrupt active level setting register, the enable/disable register, and the poll register for each pin. These registers are located in the internal control register. The following figure shows the interrupt block diagram.

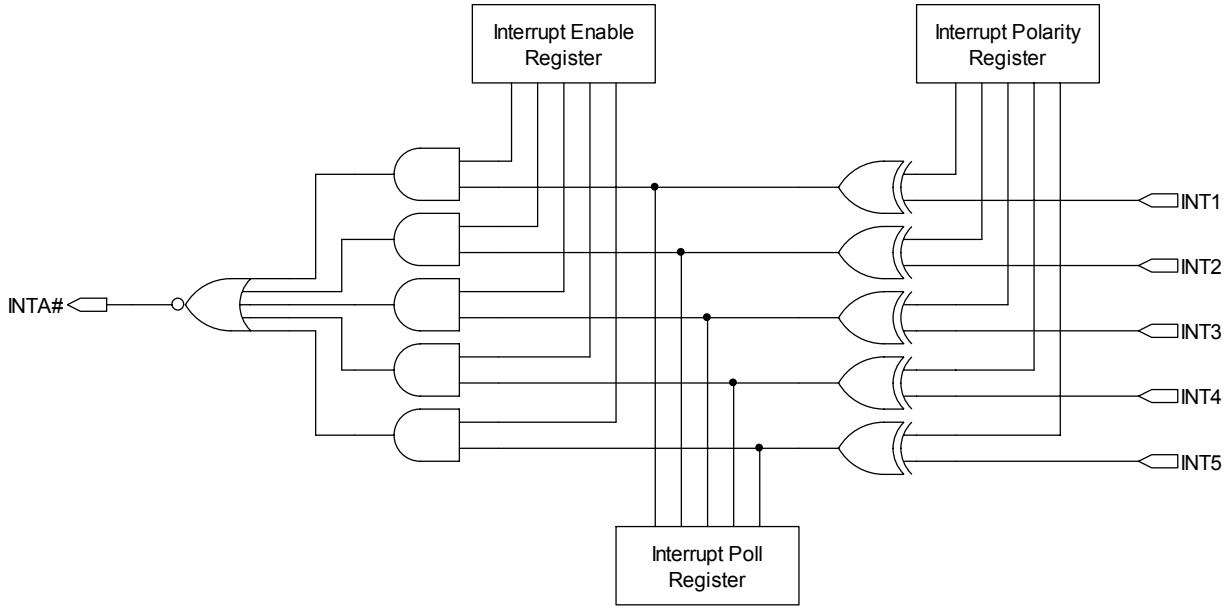


Figure 5-1. Interrupt Generator Schematic

5.2 General Purpose I/O Interface

SB4002A provides total 8 GPIO ports.

In order to control the 8 GPIO ports, SB4002A also provides GPIO Output Enable Register, GPIO Output Register and GPIO Input Register.

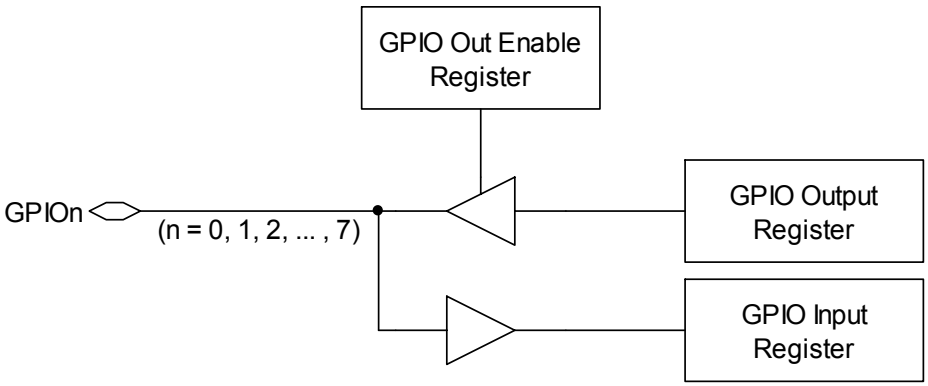


Figure 5-2. GPIO Controller Schematic

5.3 Serial ROM Interface

In order to import the configuration data, SB4002A provides an external serial EEPROM. The EEPROM is 1K in capacity, and allocates 00h~2Fh as the storage of internal configuration data, and 30h~7Eh for the user to store and display data. The serial EEPROM interface other than the configuration data is executed via the serial ROM interface register in the control register. It is necessary to limit the access addresses in order to protect the configuration data stored in the serial EEPROM.

6. Control Register

Theses addresses are used to define various internal selections and to implement the supplementary functions. The Control Register Setting0/1(See 4. Configuration Data in Serial ROM)of the Serial ROM allocates the Base Address0 or the Base Address5 to I/O Space or Memory Space in the PCI Configuration Space Header, and supports access to the unit of DWORD. The table below describes the details of the Control Registers.

Address	Byte3	Byte2	Byte1	Byte0
00h	BASE ADDRESS1 Space Setting Register			
04h	BASE ADDRESS2 Space Setting Register			
08h	BASE ADDRESS3 Space Setting Register			
0Ch	BASE ADDRESS4 Space Setting Register			
10h	BASE ADDRESS5 Space Setting Register			
14h	EXPANSION ROM BASE Address space Setting Register			
18h	Reserved			Reset Register
1Ch	Reserved	GPIO Input Register	GPIO Output Register	GPIO Out Enable Register
20h	Endian & LOCK Control Register	Poll Register	Interrupt Polarity Register	Interrupt Enable Register
24h	Reserved		Muxed Pin Control Register	

Table 6-1. Control Register Map

BASE ADDRESS1/2/3/4/5 & EXPANSION ROM BASE Address Space Setting Register

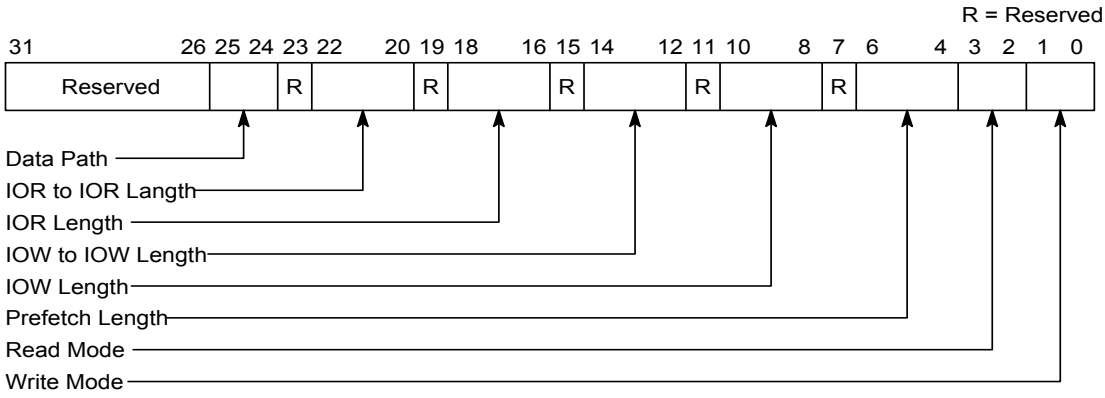


Table 6-2. Base Address Space Setting Register

If burst access is requested for the device which does not support burst access, the device finishes the cycle with Disconnect.

Bit[25:24] : Data Path for Legacy Bus (R/W)

00b : 8-bit data path (Default)

01b : 16-bit data path

10b : 32-bit data path

11b : Reserved

bit[31:22] : Reserved

Bit[22:20] : IOR# to IOR# Adding Length for Legacy Bus (R/W)

Applied to burst read and prefetch read only. Once IOW# is deasserted, the time required for assertion of the next IOW# is at least PCI 2 CLK. The length is extended by the value set in this value.

This bit is used to control the wait cycle on the PCI bus in the real time write.

Bit[18:16] : IOR# Adding Length for Legacy Bus (R/W)

Basically, IOW# has the enable length of PCI 1 CLK.

The length is extended by the value set in this value.

This bit is used to control the wait cycle on the PCI bus in the real time write.

Bit[14:12] : IOW# to IOW# Adding Length for Legacy Bus (R/W)

Applied to burst write only. Once IOW# is deasserted, the time required for assertion of the next IOW# is at least PCI 2 CLK. The length is extended by the value set in this value.

This bit is used to control the wait cycle on the PCI bus in the real time write.

Bit[10:8] : IOW# Adding Length for Legacy Bus (R/W)

Basically, IOW# has the enable length of PCI 1 CLK.

The length is extended by the value set in this value.

This bit is used to control the wait cycle on the PCI bus in the real time write.

Bit[6:5] : READ Prefetch Byte (R/W)

Supported in the delay time read mode only.

00b – Prefetch by 2-DWORD (Default)

01b – Prefetch by 4-DWORD

10b - Prefetch by 8-DWORD

11b - Prefetch by 16-DWORD

Bit[4] : READ Prefetch Enbale (R/W)

1b enables READ prefetch, and 0b disables READ prefetch. The default is 0b.

Bit[3:2] : READ Mode for PCI Bus (R/W)

00 b- Real time read / Burst not supported (Default/FIFO not used)

01b - Real time read / Burst supported (FIFO not used)

10b - Delayed time read (retry used) / Burst not supported (FIFO 1 byte used)

11b - Delayed time read (retry used) / Burst supported (FIFO n byte used / prefetch enable)

Bit[1:0] : WRITE Mode for PCI Bus (R/W)

00 - Real time write / Burst not supported (Default/FIFO not used)

01 - Real time write / Burst supported (FIFO not used)

10 - Delayed time write / Burst not supported (FIFO 1 byte used)

11 - Delayed time write / Burst supported (FIFO n byte used)

Tips : PCI Bus and Legacy Bus Description

1. Real Time Write

In a single transfer, when DEVSEL# is asserted, TRDY# is asserted at the same time to catch the data, and CS# is also asserted to output the address. Then, the device deasserts DEVSEL# and TRDY# to end the PCI cycle, asserts IOW# on the Legacy bus, and outputs the data. The device maintains IOW# for the bit[9:7]+01b before deasserting it, deasserts CS# at the next CLK, and then removes the address and data. In a burst transfer, when DEVSEL# is asserted, TRDY# is asserted at the same time to catch the data, and CS# is also asserted to output the address. Then, the device deasserts TRDY#, asserts IOW# on the Legacy bus, and outputs the data. The device, then, maintains IOW# for the bit[9:7]+01b before deasserting it. The device asserts TRDY# after the next bit[15:13]+01b, and sends the address on the Legacy bus. The device deasserts TRDY# at the next CLK, asserts IOW# on the Legacy bus,

and outputs the data. Then, the above procedure is repeated. Therefore, the TRDY# to TRDY# wait cycle on the PCI bus is bit[9:7]+bit[15:13]+10b.

2. Delayed Time Write

In a single transfer, when DEVSEL# is asserted, TRDY# is asserted at the same time to catch the data, and the device ends the PCI cycle. After an appropriate time period, the device asserts CS# and approves the address and data. The device asserts IOW# at the next CLK, and maintains IOW# for the bit[9:7]+001b before deasserting it. The device deasserts CS# and removes the address and data at the next CLK.

In a burst transfer, when DEVSEL# is asserted, the device asserts TRDY#, and executes the burst cycle in no wait (but within the size of FIFO). After an appropriate time period, the device asserts CS# and approves the address and data. The device asserts IOW# at the next CLK, maintains IOW# for the bit[9:7]+001b before deasserting it, and outputs the data. Then the above procedure is repeated. Therefore the wait cycle on the PCI bus is 0.

If a cycle is executed on the PCI bus before a cycle is finished on a Legacy bus, the device terminates the transaction with retry.

3. Real Time Read

In a single transfer, the device asserts DEVSEL#, outputs the address, and asserts CS#. The device asserts IOR# at the next CLK, maintains IOR# for the bit[12:10], and asserts TRDY#. The device deasserts IOR#, TRDY# and DEVSEL# at the next CLK, and then, at the next CLK, deasserts CS# and removes the address.

In a burst transfer, the device asserts DEVSEL#, outputs the address, and asserts CS#. The device asserts IOR# at the next CLK, maintains IOR# for the bit[12:10], and asserts TRDY# at the next CLK. At the following CLK, the device deasserts IOR#, and after bit[18:16]+2, asserts IOR#. After the time for the bit[12:10], the device asserts TRDY#. At the next CLK, the device deasserts IOR# and TRDY#. Then, the above procedure is repeated. Therefore the TRDY# to TRDY# wait cycle on the PCI bus is bit[12:10]+bit[18:16]+2.

4. Delayed Time Read without Prefetch

It is applied to a single transfer only. On receiving the initial access, the device stores the address and command, finishes the PCI bus cycle with retry, and after an appropriate time period, executes a read transaction on the Legacy bus. Where, the length of IOR# is bit[12:10] +1. Then, if the same cycle is started on the PCI bus, the device responds to the cycle in no wait.

If a new read cycle is started after retry, the previous cycle is disregarded. If a PCI cycle is

started when a read cycle on the Legacy bus is not finished, the device ends the cycle with retry.

5. Delayed Time Read with Prefetch

It is applied to a burst transfer only. On receiving the initial access, the device stores the address and command, finishes the PCI bus cycle with retry, and after an appropriate time period, executes a read transaction on the Legacy bus. The length of IOR# is bit[12:10]+001b, and the time between deassertion of an IOR# and assertion of the next IOR# is bit[18:16] +010b. If the same cycle is started on the PCI bus, the device responds to the cycle in no wait. The size of data prefetched at a time is designated in the bit[6:5].

If the same cycle is started when prefetched data is in FIFO, the device responds to the burst cycle for the prefetched data. If the cycle is requested continuously, the device terminates the transaction with disconnect. The Legacy bus keeps executing the prefetch.

If a new read cycle or a write cycle is started for other address when prefetch is not finished on the Legacy bus, the device terminates the transaction with retry.

※ CS# is PCI 1 CLK length added before and after IOW# and IOR#.

※ The wait cycle must follow the 8 CLK rule. In other words, after TRDY# is deasserted in the burst transfer, TRDY# must be asserted within 8 PCI CLK.

※ In the 8-bit data space mode, if a 16-bit access is started on the PCI bus, two 8-bit accesses must be executed. If a 32-bit access is executed, four 8-bit accesses must be executed.

If a 32-bit access is started on the PCI bus in the 16-bit data path, two 16-bit accesses must be executed.

※ Errata : A real time read burst is executed on the assumption that byte enable is not changed during the access. If the byte enable is changed during the access, an erroneous value can be read. Therefore, if byte enable is changed during the access, it is recommended to use the delayed time mode instead of the real time mode.

Reset Register

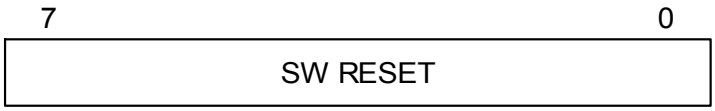


Table 6-3. Reset Register Layout

This register resets the chip internal register and Legacy bus in the software level. (Write Only)

Bit[7:0] : If 52h corresponding to “R" is used, the default is the chip internal register (excluding the register on Configuration Space Header and Reset Register), and the reset signal is output through the reset pin on the Legacy bus. If other values are used, reset is released.

GPIO Out Enable Register

R = Reserved

7	6	5	4	3	2	1	0
R	GEN6	GEN5	GEN4	GEN3	GEN2	GEN1	GEN0

Table 6-4. GPIO Out Enable Register Layout

Sets input/output of each GPIO.

Bit[7] : Reserved

Bit[6] : GEN6, 1b : Enables GPIO6 Out, 0b : Disables GPIO6 Out(Default)

Bit[5] : GEN5, 1b : Enables GPIO5 Out, 0b : Disables GPIO5 Out(Default)

Bit[4] : GEN4, 1b : Enables GPIO4 Out, 0b : Disables GPIO4 Out(Default)

Bit[3] : GEN3, 1b : Enables GPIO3 Out, 0b : Disables GPIO3 Out(Default)

Bit[2] : GEN2, 1b : Enables GPIO2 Out, 0b : Disables GPIO2 Out(Default)

Bit[1] : GEN1, 1b : Enables GPIO1 Out, 0b : Disables GPIO1 Out(Default)

Bit[0] : GEN0, 1b : Enables GPIO0 Out, 0b : Disables GPIO0 Out(Default)

GPIO Output Register

R = Reserved

7	6	5	4	3	2	1	0
R	GO6	GO5	GO4	GO3	GO2	GO1	GO0

Table 6-5. GPIO Output Register Layout

Sets output of each GPIO.

Bit[7] : Reserved

Bit[6] : GO6, Sets the output of GPIO6.

Bit[5] : GO5, Sets the output of GPIO5.

Bit[4] : GO4, Sets the output of GPIO4.

Bit[3] : GO3, Sets the output of GPIO3.

Bit[2] : GO2, Sets the output of GPIO2.

Bit[1] : GO1, Sets the output of GPIO1.

Bit[0] : GO0, Sets the output of GPIO0.

GPIO Input Register

R = Reserved

7	6	5	4	3	2	1	0
R	GI6	GI5	GI4	GI3	GI2	GI1	GI0

Table 6-6. GPIO Input Register Layout

Reads input of each GPIO.

Bit[7] : Reserved

Bit[6] : GI6, Reads the input of GPIO6.

Bit[5] : GI5, Reads the input of GPIO5.

Bit[4] : GI4, Reads the input of GPIO4.

Bit[3] : GI3, Reads the input of GPIO3.

Bit[2] : GI2, Reads the input of GPIO2.

Bit[1] : GI1, Reads the input of GPIO1.

Bit[0] : GI0, Reads the input of GPIO0.

Interrupt Enable Register

7 ~ 5	4	3	2	1	0
Reserved	IE4	IE3	IE2	IE1	IE0

Table 6-7. Interrupt Enable Register Layout

Enables/Disables 5 interrupt signals. (R/W)

Bit[4] : IE4, 1b : Enables INT4, 0b : Disables INT4(Default).

Bit[3] : IE3, 1b : Enables INT3, 0b : Disables INT3(Default).

Bit[2] : IE2, 1b : Enables INT2, 0b : Disables INT2(Default).

Bit[1] : IE1, 1b : Enables INT1, 0b : Disables INT1(Default).

Bit[0] : IE0, 1b : Enables INT0, 0b : Disables INT0(Default).

Interrupt Polarity Register

7 ~ 5	4	3	2	1	0
Reserved	IP4	IP3	IP2	IP1	IP0

Table 6-8. Interrupt Polarity Register Layout

Indicates polarities of 5 interrupt signals.(R/W)

Bit[4] : IP4, 1b : INT4 is positive polarity, 0b : INT4 is negative polarity(Default).

Bit[3] : IP3, 1b : INT3 is positive polarity, 0b : INT3 is negative polarity(Default).

Bit[2] : IP2, 1b : INT2 is positive polarity, 0b : INT2 is negative polarity(Default).

Bit[1] : IP1, 1b : INT1 is positive polarity, 0b : INT1 is negative polarity(Default).

Bit[0] : IP0, 1b : INT0 is positive polarity, 0b : INT0 is negative polarity(Default).

Poll Register

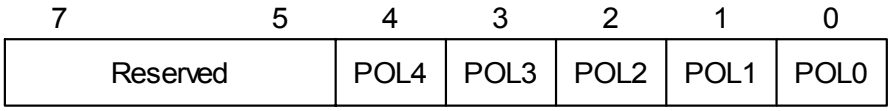


Table 6-9. Poll Register Layout

Indicates the interrupt status on the Legacy bus regardless of the enable register value.

Bit[4] : POL4, 1 : Indicates generation of an INT4.

Bit[3] : POL3, 1 : Indicates generation of an INT3.

Bit[2] : POL2, 1 : Indicates generation of an INT2.

Bit[1] : POL1, 1 : Indicates generation of an INT1.

Bit[0] : POL0, 1 : Indicates generation of an INT0.

Endian & LOCK Control Register

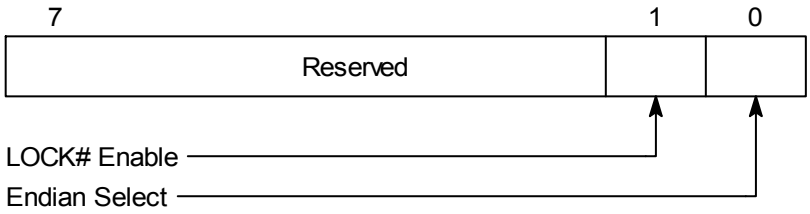


Table 6-10. Endian & LOCK Control Register Layout

Sets the values for other supplementary functions.

Bit[1] : Endian Select (R/W)

Big Endian for 1b and Little Endian for 0b, Default:0b

Bit[0] - LOCK# Enable : If this bit is 1b, the exclusive access is started in response to LOCK#, Default:0b

Muxed Pin Control Register

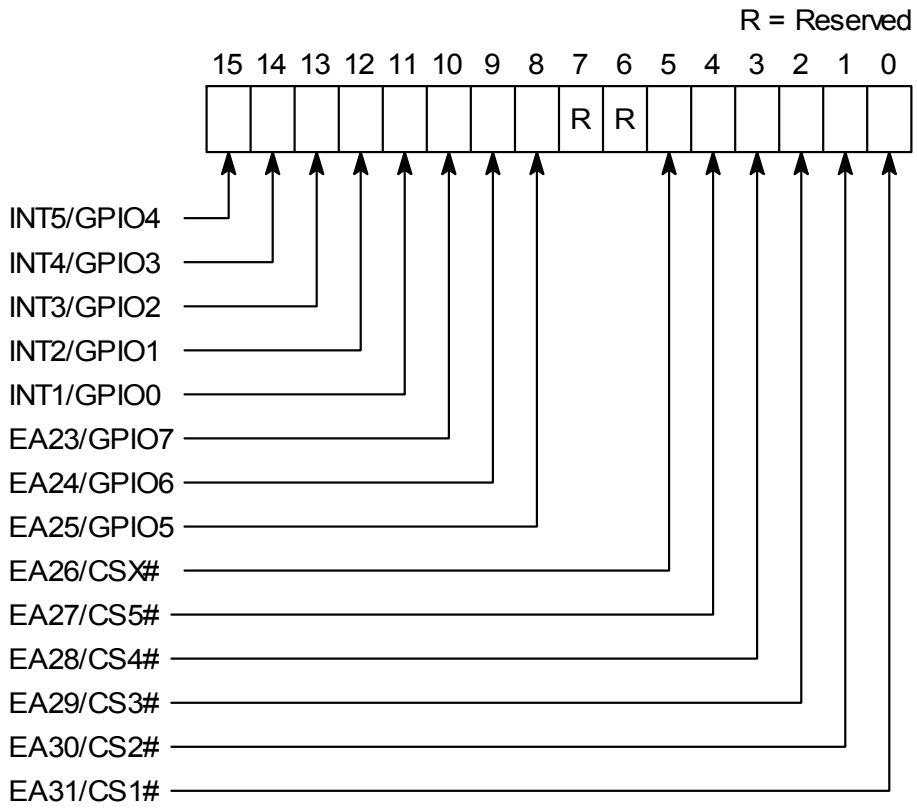


Table 6-11. Muxed Pin Control Register Layout

Sets the values for other supplementary functions.

Bit[15] : INT5/GPIO4, 1b : INT5 selected(Default), 0b : GPIO4 selected

Bit[14] : INT4/GPIO3, 1b : INT4 selected(Default), 0b : GPIO3 selected

Bit[13] : INT3/GPIO2, 1b : INT3 selected(Default), 0b : GPIO2 selected

Bit[12] : INT2/GPIO1, 1b : INT2 selected(Default), 0b : GPIO1 selected

Bit[11] : INT1/GPIO0 : 1b : INT1 selected(Default), 0b : GPIO0 selected

Bit[10] : EA23/GPIO7, 1b : EA23 selected(Default), 0b : GPIO7 selected

Bit[9] : EA24/GPIO6, 1b : EA24 selected(Default), 0b : GPIO6 selected

Bit[8] : EA25/GPIO5, 1b : EA25 selected(Default), 0b : GPIO5 selected

Bit[5] : EA26/CSX#, 1b : EA26 selected, 0b : CSX# selected(Default)

Bit[4] : EA27/CS5#, 1b : EA27 selected, 0b : CS5# selected(Default)

Bit[3] : EA28/CS4#, 1b : EA28 selected, 0b : CS4# selected(Default)

Bit[2] : EA29/CS3#, 1b : EA29 selected, 0b : CS3# selected(Default)

Bit[1] : EA30/CS2#, 1b : EA30 selected, 0b : CS2# selected(Default)

Bit[0] : EA31/CS1#, 1b : EA31 selected, 0b : CS1# selected(Default)

7. Physical Specification

7.1 Pin Location

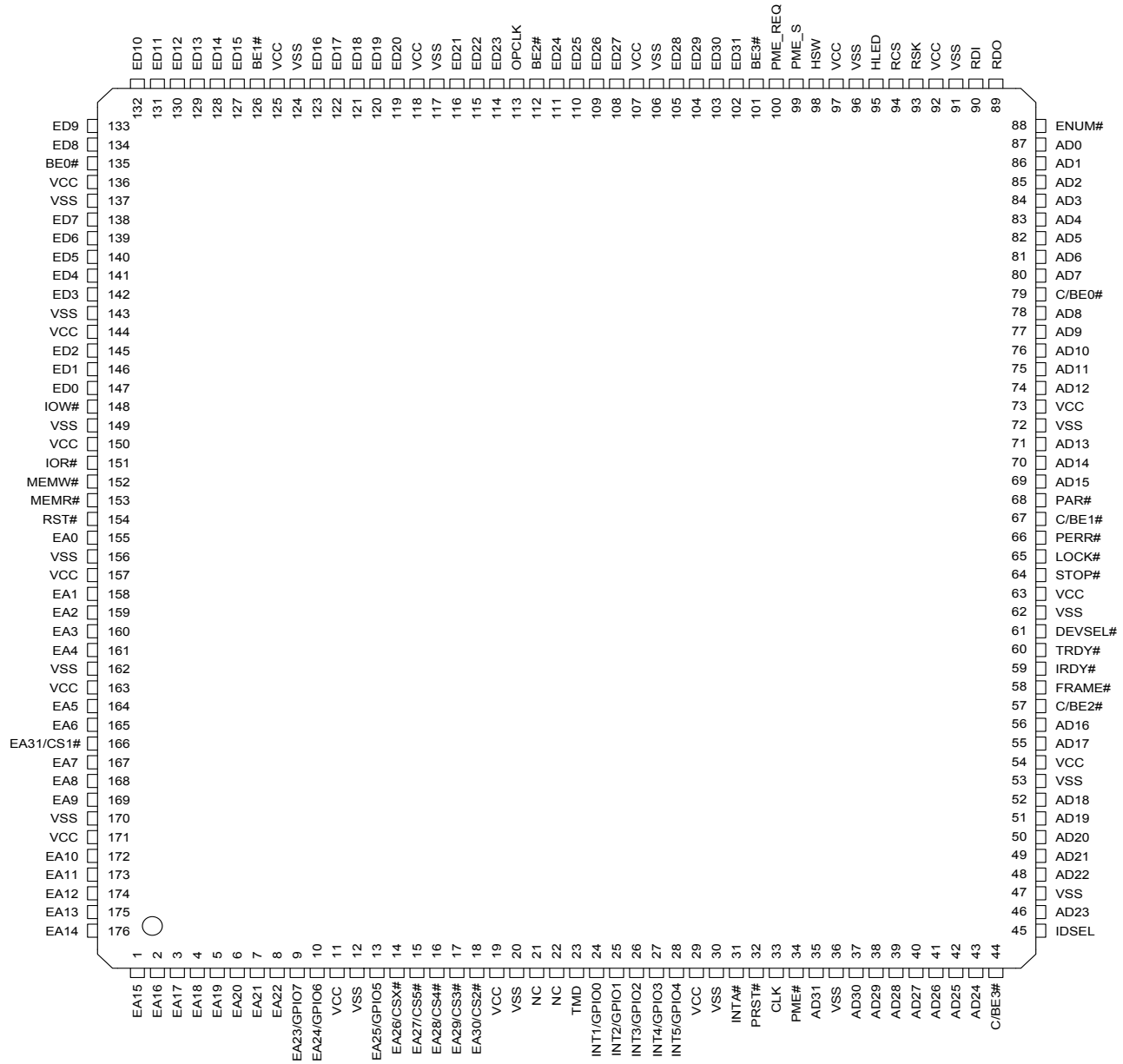


Figure 7-1. SB4002A Pin Location

7.2 Pin Description

Signal Name	Pin No.	Pin Type	Description
PCI Bus Signal			
CLK	33	in PCI	Provides timing for all actions of PCI bus. The signal gives input for all the PCI devices, and its rising edge becomes the reference for input/output and timing constant of all signals. (exception: RST#, INTx#, PME#, CLKRUN#)
PRST#	32	in PCI	Reset signal of the PCI system.
AD[31:0]	-	t/s	Address/Data multiplexed signal. This signal is used during the PCI transaction as the address signal in the address phase or as the data signal in the data phrase. (Pin No. : 35, 37, 38, 39, 40, 41, 42, 43, 46, 48, 49, 50, 51, 52, 55, 56, 69, 70, 71, 74, 75, 76, 77, 78, 80, 81, 82, 83, 84, 85, 86, 87)
C/BE[3:0]#	-	t/s	Bus Command/Byte Enable multiplexed signal. It is used during the PCI transaction as BUS Command in the address phase or as Byte Enable in the data phrase. (Pin No. : 44, 57, 67, 79)
PAR	68	t/s	Provides the even parity for AD[31:0] and C/BE[3:0].
FRAME#	58	s/t/s	Indicates start and continuance of PCI transaction.
IRDY#	59	s/t/s	Initiator Ready. The signal indicates that the BUS master is ready to output data.
TRDY#	60	s/t/s	Target Ready. The signal indicates that the target device is ready to output data.
STOP#	64	s/t/s	Used by the target to request the master to stop PCI transaction.
LOCK#	65	s/t/s	Used for exclusive access.
IDSEL	45	in	Indicates that a device is selected for configuration access.
DEVSEL#	61	s/t/s	Device Select. Used by the selected target to indicate that it is selected.
PERR#	66	s/t/s	Parity Error. Indicates that a parity error occurs in the data phase.

Signal Name	Pin No.	Pin Type	Description
INTA#	31	o/d	PCI interrupt signal.
PME#	34	o/d	Power Management Event. Requests change of power state of a device or a system. Enabled when the power management interface capability is supported.
ENUM#	88	o/d	Enumeration. Shows that the board has been inserted to the system, or is about to be removed. This is enabled when the CompactPCI Hot Swap is supported.
Legacy Bus Signal			
EA[22:0]	-	out	Legacy Bus Address. The lower 26-bit output at transaction on the Legacy bus. (Pin No. : 8, 7, 6, 5, 4, 3, 2, 1, 176, 175, 174, 173, 172, 169, 168, 167, 165, 164, 161, 160, 159, 158, 155)
GPIO7/EA[23]	9	bi /out	The Legacy Bus Address and General Purpose I/O Port muxed pin. The Muxed Pin Select Register [10:8] in the control register space is used to determine the use of the pin (address pin or GPIO pin).
GPIO6/EA[24]	10	bi /out	
GPIO5/EA[25]	13	bi /out	
CSX#/EA[26]	14	out	The Legacy Bus Address and CSn# muxed pin. The Muxed Pin Select Register [5:0] in the control register space is used to determine the use of the pin (address pin or CSn# pin) In CSn#, n indicates the order of the base address spaces. If the control register space is allocated to the base address5, the CS5# pin function is changed to CS0#.
CS5#/CS0#/EA[27]	15	out	
CS4#/EA[28]	16	out	
CS3#/EA[29]	17	out	
CS2#/EA[30]	18	out	
CS1#/EA[31]	166	out	
ED[31:0]	-	bi	Legacy Bus Data. Data for the Legacy bus transaction is input and output through this pin. The effective data line varies by data path, endian and byte enable. (Pin No. : 102, 103, 104, 105, 108, 109, 110, 111, 114, 115, 116, 119, 120, 121, 122, 123, 127, 128, 129, 130, 131, 132, 133, 134, 138, 139, 140, 141, 142, 145, 146, 147)
RST#	154	out	Legacy Bus Reset. The reset signal output to the Legacy bus. Asserted when PRST# of PCI bus is asserted. It is also asserted if 52h is written on the reset register of the control register space.
IOR#	151	out	I/O Read Signal. Asserted at read access in the I/O port.

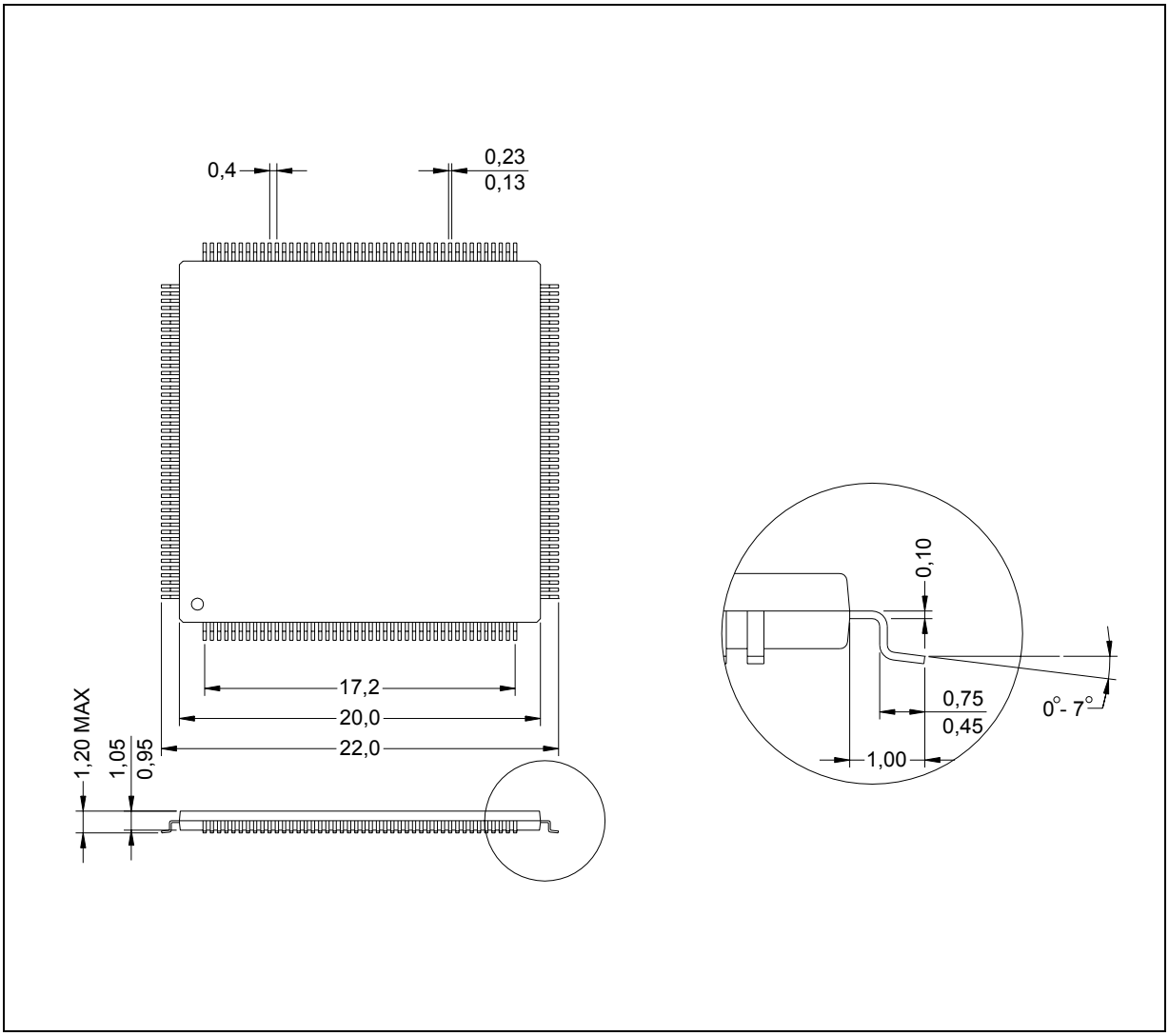
Signal Name	Pin No.	Pin Type	Description
IOW#	148	out	I/O Write Signal. Asserted at write access in the I/O port.
MEMR#	153	out	Memory Read Signal. Asserted at read access in the memory.
MEMW#	152	out	Memory Write Signal. Asserted at write access in the memory.
BE[3:0]#	-	out	Byte Enable Signal. Enable signal for each byte of the data signal on the Legacy bus. BE[3]#,BE[2]#,BE[1]#,BE[0]# are corresponding to ED[31:24], ED[23:16], ED[15:8] and ED[7:0], respectively. If the data path is in bytes, no substantial influence is made. (Pin No. : 101, 112, 126, 135)
INT0/GPIO0	24	in / bi	The interrupt request and general purpose I/O port muxed pin. The Muxed Pin Select Register [15:11] in the control register space is used to determine the use of the pin (INT pin or GPIO pin)
INT1/GPIO1	25		
INT2/GPIO2	26		
INT3/GPIO3	27		
INT4/GPIO4	28		
PME_REQ	100	in	Power Management Event Request Signal. The signal asserts the PME# signal to request the power management event.
PME_S	99	out	Power Management Status Signal. The signal reports the power state of the device.
HSW	98	in	CompactPCI Hot Swap Switch Signal. The pin receives the switch input for insert/eject at hot swap.
HLED	95	out	CompactPCI Hot Swap Switch Signal. The pin outputs the LED on/off signal for insert/eject complete at hot swap. The LOO value of HS_CSR in the configuration is out.
OPCLK	113	out	Output of PCI clock.
Serial EEPROM Interface Signal			
RCS	94	out	Serial EEPROM Chip Select Signal. Connected to CS of the Serial EEPROM.
RSK	93	out	Serial EEPROM Serial Data Clock Signal. Connected to SK of the Serial EEPROM.
RDI	90	in	Serial EEPROM Serial Data Input Signal. Connected to DO of the Serial EEPROM.
RDO	89	out	Serial EEPROM Serial Data Output Signal. Connected to DI of the Serial EEPROM.

Signal Name	Pin No.	Pin Type	Description
Others			
TMD	23	in	Connected to VCC via the 10K resistance.
NC	21 22	-	No Connection

Table 7-1. SB4002A Pin Description

7.3 Package Dimension

THIN QUAD FLATPACK



NOTES: A. All linear dimensions are in millimeters.

B. This drawing is subject to change without notice.

7.3 Electrical Specification

SB4002A is based on 3.3V I/O, and supports 5V tolerance.

Recommended Operation Conditions

Parameters	Minimum	Maximum	Conditions
Power Supply	3.0V	3.6V	
VIL,low level input voltage			Guaranteed Input Low Voltage
PCI input	-0.5V	0.3VDD	
General input	-0.5V	0.3VDD	
VIH,low level input voltage			Guaranteed Input High Voltage
PCI input	0.5VDD	VDD+0.5V	
General input	0.7VDD	VDD+0.5V	
Junction Temperature	0℃	100℃	

Table 7-2. Recommended Operation Conditions

DC Characteristics

Parameter	Minimum	Maximum	VDD	Conditions
VIL , Low level input voltage			2.7V to 3.6V	Guaranteed Input Low Voltage
PCI input	-0.5V	0.3xVDD		
General input	-0.5V	0.3xVDD		
VIH , Hi level input voltage			2.7V to 3.6V	Guaranteed Input High Voltage
PCI input	0.5VDD	VDD+0.5V		
General input	0.7VDD	VDD+0.5V		
VOL , Low level output voltage			2.7V	
PCI output (IOL = 1.8mA)		0.1VDD		
General output (IOL = 0.8mA)		VSS+0.1V		
VOH , High level output voltage			2.7V	
PCI output (IOL = -0.5mA)	0.9VDD			
General output (IOL = 0.8mA)	VDD-0.1V			

Table 7-3. DC Characteristics

* PCI I/O shows the characteristics of 3.3V 66MHz PCI Bus System.

8. Timing Specification

8.1 PCI BUS Timing Specifications

		66MHz		33MHz		
Symbol	**Parameter**	**Min**	**Max**	**Min**	**Max**	**Units**
Tval	CLK to Signal Valid Delay - Bused Signals	2	6	2	11	ns
Tval(ptop)	CLK to Signal Valid Delay - point to point Signals	2	6	2	12	ns
Ton	Float to Active Delay	2		2		ns
Toff	Active to Float Delay		14		28	ns
Tsu	Input Setup Time to CLK - bused Signals	3		7		ns
Tsu(ptop)	Input Setup Time to CLK - point to point signals	5		10,12		ns
Th	Input Hold Time from CLK	0		0		ns
Trst	Reset Active Time after power stable	1		1		ms
Trst-clk	Reset Active Time after CLK stable	100		100		us
Trst-off	Reset Active to output float delay		40		40	ns
Trhfa	RST# high to first Configuration access	2 25		2 25		clocks
Trhff	RST# high to first FRAME# assertion	5		5		clocks

Table 8-1. PCI Bus Timing Specifications

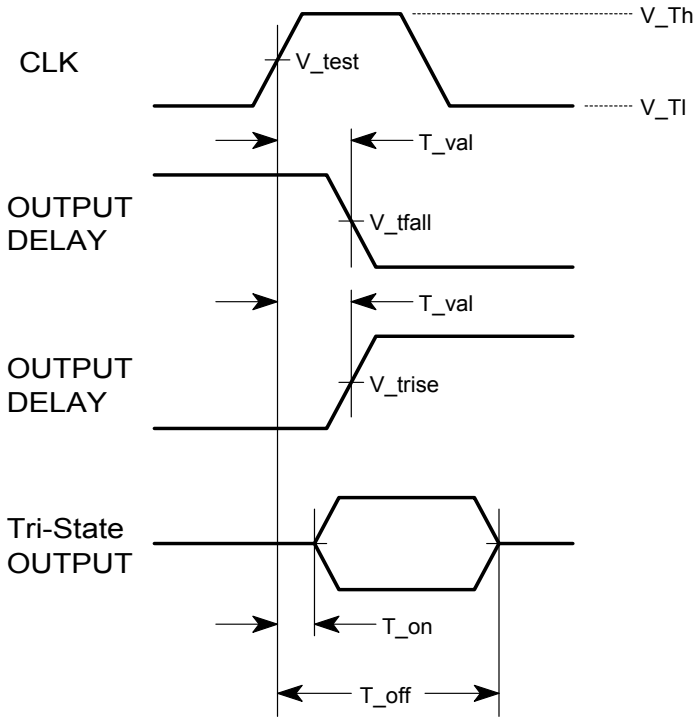


Figure 8-1. Output Timing Measurement Conditions

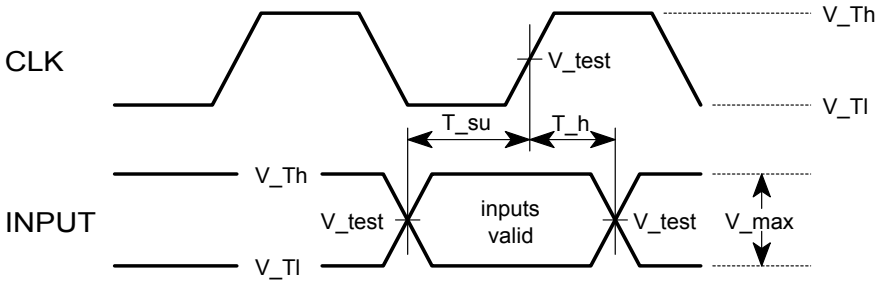


Figure 8-2. Input Timing Measurement Conditions

* PCI Bus timing specification is as defined in the PCI Local Bus Specification Ver 2.3.
* SB4002A satisfies the above mentioned timing specification in the aspect of PCI Bus interface.

8.2 Legacy Bus Timing Specifications

Symbol	Parameter	Min	Typ	Max	Units
Td1	CS#/EA/BYTE_EN# assert to IOW#/MEMW# assert	-	1	-	PCI CLK
Td2	IOW#/MEMW# deassrt to CS#/EA/BYTE_EN# deassert	-	1	-	PCI CLK
Td3	IOW#/MEMW# enable to ED valid	-	-	3	ns
Td4	IOW#/MEMW# disable to ED valid	-	1	-	PCI CLK
Tw1	IOW#/MEMW# width	-	n	-	PCI CLK

* 'n' meams IOW#/MEMW# signal width is 'n' times PCI CLK.

'n' would be determined by Base Address Space Setting Register[10:8], Refer to Chapter 6

Table 8-2. Legacy Bus Timing Specifications for Write Access

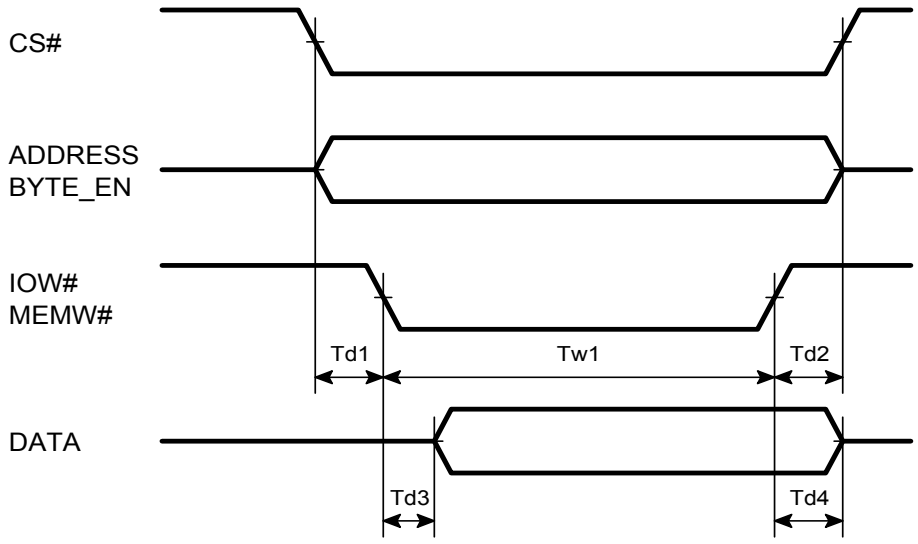


Figure 8-3. Legacy Bus Timing Specifications for Write Access

Symbol	Parameter	Min	Typ	Max	Units
Td5	CS#/EA/BYTE_EN# assert to IOR#/MEMR# assert	-	1	-	PCI CLK
Td6	IOR#/MEMR# deassrt to CS#/EA/BYTE_EN# deassert	-	1	-	PCI CLK
Trs	ED valid to IOR#/MEMR# disable (required setup time for read on 66MHz)	15.5	-	-	ns
	ED valid to IOR#/MEMR# disable (required setup time for read on 33MHz)	19.5	-	-	
Trh	IOR#/MEMR# disable to ED unvalid (reguired hold time for read)	0	-	-	PCI CLK
Tr1	IOR#/MEMR# width	-	n	-	PCI CLK

* 'n' meams IOR#/MEMR# signal width is 'n' times PCI CLK.

'n' would be determined by Base Address Space Setting Register[10:8], Refer to Chapter 6

Table 8-3. Legacy Bus Timing Specifications for Read Access

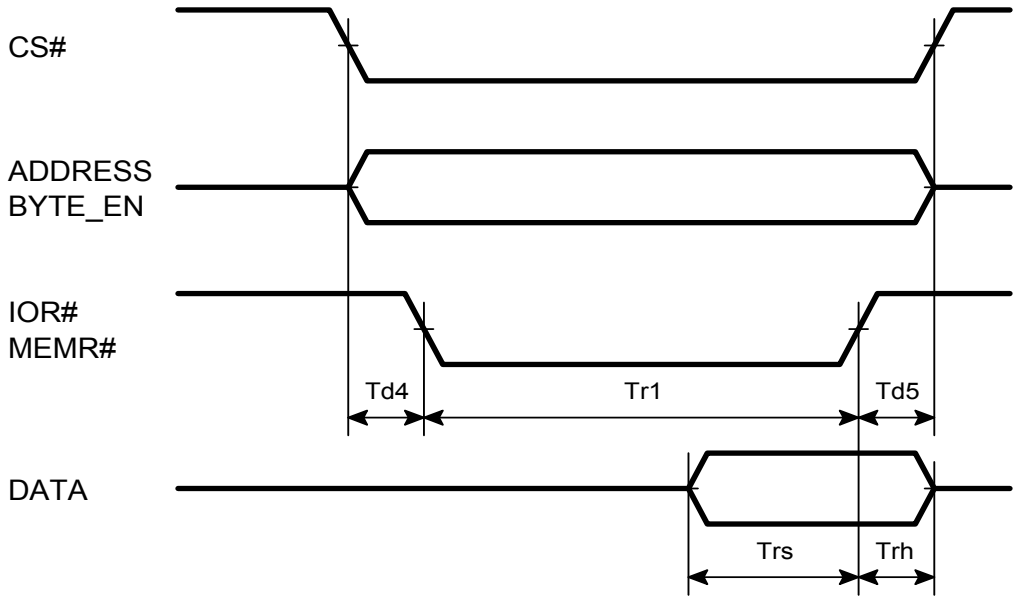


Figure 8-4. Legacy Bus Timing Specifications for Read Access

9. Company Information

SystemBase Co., Ltd.
1601, Daerung Post Tower-1, 212-9, Guro-dong, Seoul, Korea
Tel) 82-2-855-0501
Fax) 82-2-855-0850