CONTENTS

11.0 FIFO OVERVIEW

The S5933 has two internal FIFOs. One FIFO is for PCI bus to Add-On bus, the other FIFO is for Add-On bus to PCI bus transfers. Each of these has eight 32-bit registers. The FIFOs are both addressed through a single PCI/Add-On Operation Register offset, but which internal FIFO is accessed is determined by whether the access is a read or write.

The FIFO may be either a PCI target or a PCI initiator. As a target, the FIFO allows a PCI bus master to access Add-On data. The FIFO also allows the S5933 to become a PCI initiator. Read and write address registers and transfer count registers allow the S5933 to perform DMA transfers across the PCI bus. The FIFO may act as initiator and a target at different times in the same application.

The FIFO can be configured to support various Add-On bus configurations. FIFO status and control signals allow simple cascading into an external FIFO, the Add-On bus can be 8-, 16-, or 32-bits wide, and data endian conversion is optional to support any type of Add-On CPU. PCI and Add-On interrupt capabilities are available to support bus mastering through the FIFO.

11.1 FUNCTIONAL DESCRIPTION

The S5933 FIFO interface allows a high degree of functionality and flexibility. Different FIFO management schemes, endian conversion schemes, and advance conditions allow for a wide variety of Add-On interfaces. Applications may implement the FIFO as either a PCI target or program it to enable the S5933 to be a PCI initiator (bus master). The following sections describe, on a functional level, the capabilities of the S5933 FIFO interface.

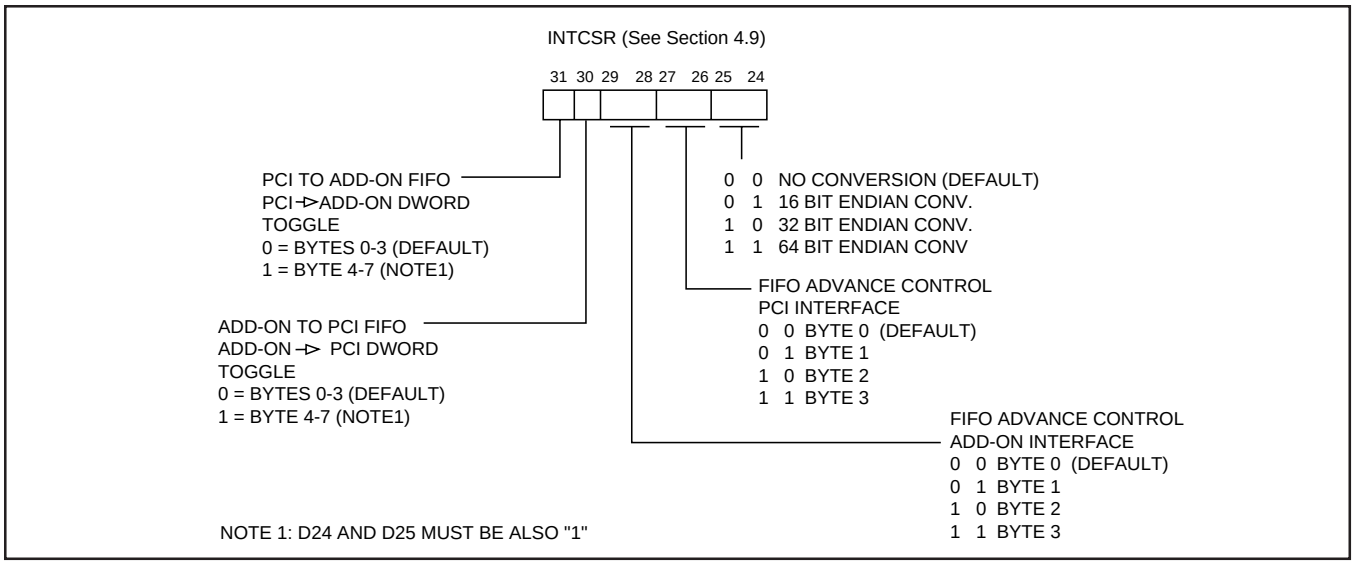
11.1.1 FIFO Buffer Management and Endian Conversion

The S5933 provides a high degree of flexibility for controlling the data flow through the FIFO. Each FIFO (PCI to Add-On and Add-On to PCI) has a specific FIFO advance condition. For FIFO writes, the byte which signifies a location is full is configurable. For FIFO reads, the byte which signifies a location is empty is configurable. This ability is useful for transferring data through the FIFO with Add-Ons which are not 32-bits wide. Endian conversion may also be performed on data passing through the FIFO.

11.1.1.1 FIFO Advance Conditions

The specific byte lane used to advance the FIFO, when accessed, is determined individually for each FIFO interface (PCI and Add-On). The control bits to set the advance condition are D29:26 of the Interrupt Control/Status Register (INTCSR) in the PCI Operation Registers (Figure 11-1). The default FIFO advance condition is set to byte 0. With the default setting, a write to the FIFO with BE0# asserted indicates that the FIFO location is now full, advancing the FIFO pointer to the next location. BE0# does not have to be the only byte enable asserted. Note, the FIFO advance condition may be different for the PCI to Add-On FIFO and the Add-On to PCI FIFO directions.

Figure 11-1. INTCSR FIFO Advance and Endian Control Bits



The configurable FIFO advance condition may be used to transfer data to and from Add-On interfaces which are not 32-bits wide. For a 16-bit Add-On bus, the Add-On to PCI FIFO advance condition can be set to byte 2. This allows a 16-bit write to the lower 16-bits of the FIFO register (bytes 0 and 1) and a second write to the upper 16-bits of the FIFO register (bytes 2 and 3). The FIFO does not advance until the second access. This allows the Add-On to operate with 16-bit data, while the PCI bus maintains a 32-bit datapath.

Notes:
1. During operation, the INTCSR FIFO advance condition bits (D29:26) should only be changed when the FIFO is empty and is idle on both the Add-On and PCI interfaces.

11.1.1.2 Endian Conversion

Bits D31:30 and D25:24 of the INTCSR PCI Operation Register control endian conversion operations for the FIFO (Figure 11-1). When endian conversion is performed, it affects data passing in either direction through the FIFO interface. Figures 11-2a and 11-2b show 16-bit and 32-bit endian conversion. It is important to note that endian conversion is performed on data BEFORE it enters the FIFO. This affects the FIFO advance condition. Example: the FIFO is configured to perform 32-bit endian conversion on data, and the FIFO advance condition is set to byte 0. Byte 3 is written into the FIFO (BE3# asserted). After the endian conversion, byte 3 becomes byte 0, and the FIFO advances. This behavior must be considered when not performing full 32-bit accesses to the FIFO.

Notes:
1. During operation, the INTCSR FIFO endian conversion bits (D25:24) and 64-bit access bits (D31:30) should only be changed when the FIFO is empty and is idle on both the Add-On and PCI interfaces.

Figure 11-2a. 16-bit Endian Conversion

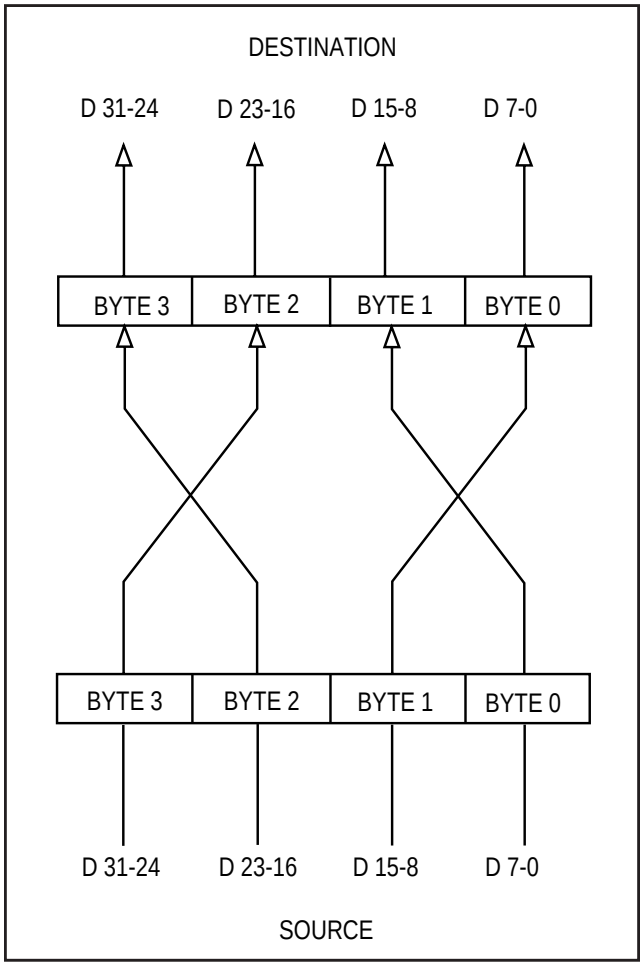
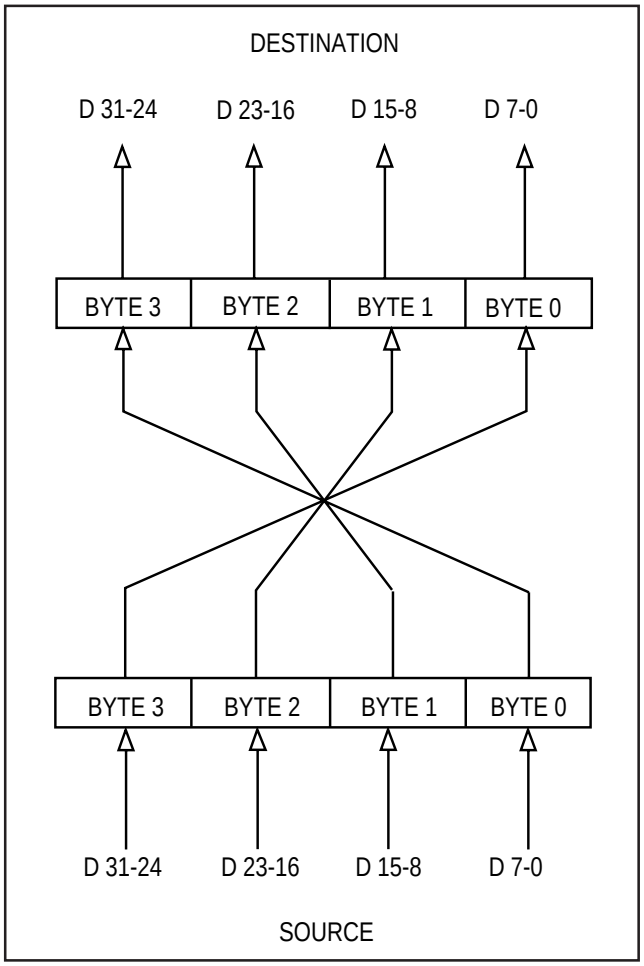


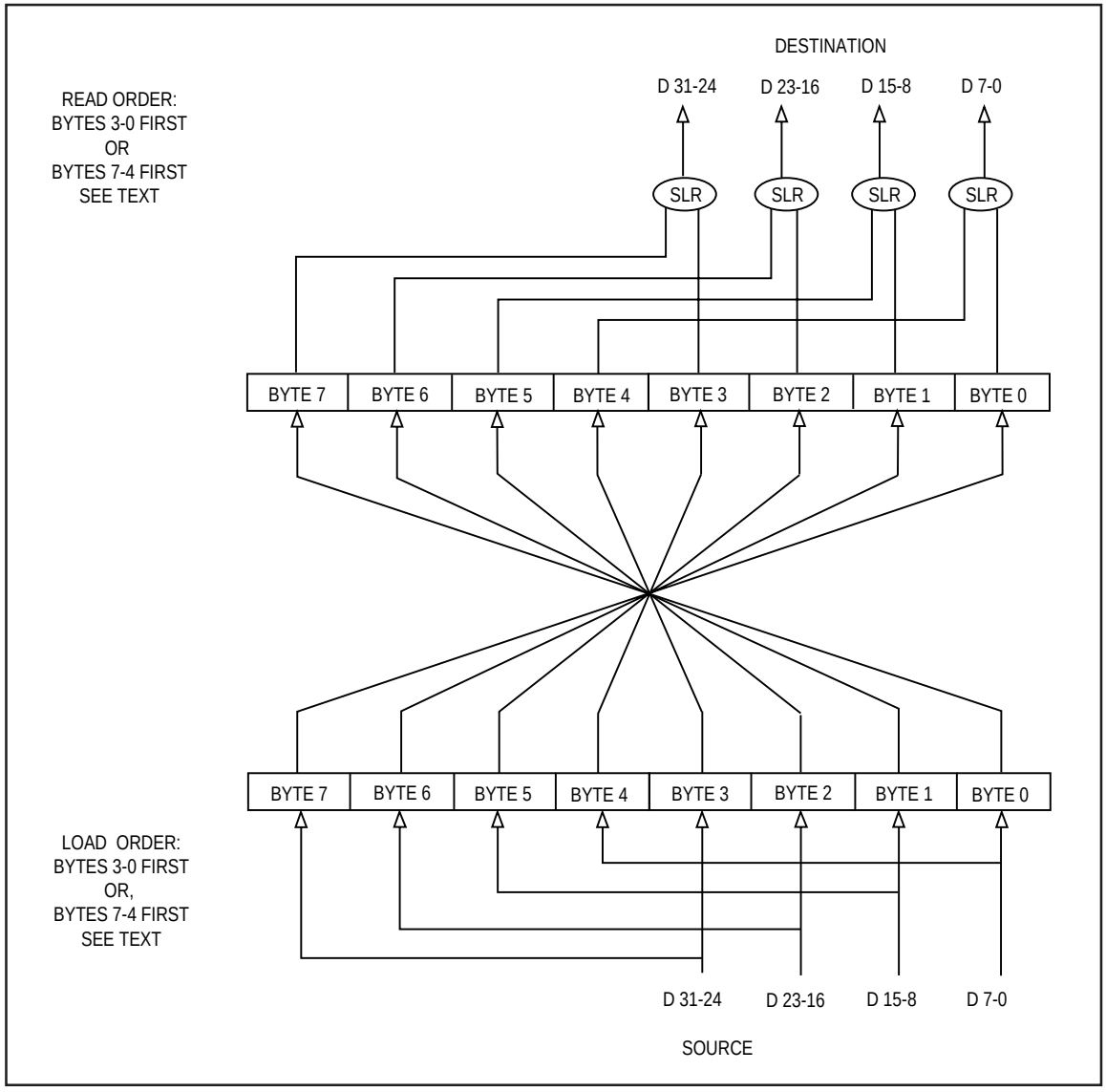
Figure 11-2b. 32-bit Endian Conversion



11.1.1.3 64-Bit Endian Conversion

Because the S5933 interfaces to a 32-bit PCI bus, special operation is required to handle 64-bit data endian conversion. Figure 11-2c shows 64-bit endian conversion. The S5933 must know whether the lower 32-bits enter the FIFO first or the upper 32-bits enter the FIFO first. INTCSR D31:30 identify which method is used by the application. These bits toggle after each 32-bit operation to indicate if half or all of a 64-bit data operation has been completed. The initial state of these bits establishes the loading and emptying order for 64-bit data during operation.

Figure 11-2c. 64-bit Endian Conversion



11.1.2 Add-On FIFO Status Indicators

The Add-On interface implements FIFO status pins to indicate the full and empty conditions of the PCI to Add-On and Add-On to PCI FIFOs. These may be used by the Add-On to allow data transfers between the FIFO and memory, a peripheral, or even a cascaded external FIFO. The RDEMPTY and WRFULL status outputs are always available to the Add-On. Additional status signals are multiplexed with the byte-wide, non-volatile memory interface pins. If the S5933 is configured for Add-On initiated bus mastering, these status signals also become available to the Add-On. FIFO status is also indicated by bits in the Add-On General Control/Status and Bus Master Control/Status Registers. The table below lists all FIFO status outputs and their functions.

Signal	Function
RDEMPTY	Indicates empty condition of the PCI to Add-On FIFO
WRFULL	Indicates full condition of the Add-On to PCI FIFO
FRF	Indicates full condition of the PCI to Add-On FIFO (note 1)
FWE	Indicates the empty condition of the Add-On to PCI FIFO (note 1)

Notes:
1. These signals are only available when a serial non-volatile memory is used and the device is configured for Add-On initiated bus mastering (Section 11.3.1).

11.1.3 Add-On FIFO Control Signals

The Add-On interface implements FIFO control pins to manipulate the S5933 FIFOs. These may be used by Add-On to control data transfer between the FIFO and memory, a peripheral, or even a cascaded external FIFO. The RDFIFO# and WRFIFO# inputs are always available. These pins allow direct access to the FIFO without generating a standard Add-On register access using RD#, WR#, SELECT#, address pins and the byte enables.

Additional control signals are multiplexed with the byte-wide, non-volatile memory interface pins. If a serial non-volatile memory is used and the S5933 is configured for Add-On initiated bus mastering, these control signals also become available. For PCI initiated bus mastering, AMREN, AMWEN, FRC#, and FWC# functionality is always available through bits in the Bus Master Control/Status and Add-On General Control/Status Registers. The FIFO control inputs are listed below.

Signal	Function
RDFIFO#	Reads data from the PCI to Add-On FIFO
WRFIFO#	Writes data into the Add-On to PCI FIFO
FRC#	Reset PCI to Add-On FIFO pointers and status indicators (note 1)
FWC#	Reset Add-On to PCI FIFO pointers and status indicators (note 1)
AMREN	Enable bus mastering for Add-On initiated PCI reads (note 1)
AMWEN	Enable bus mastering for Add-On initiated PCI writes (note 1)

Notes:
1. These signals are only available when a serial non-volatile memory is used and the S5933 is configured for Add-On initiated bus mastering (see Section 11.3.1).

11.1.4 PCI Bus Mastering with the FIFO

The S5933 may initiate PCI bus cycles through the FIFO interface. The S5933 allows blocks of data to be transferred to and from the Add-On by specifying a source/destination address on the PCI bus and a transfer byte count. This DMA capability allows data to be transferred across the PCI bus without host CPU intervention.

Initiating a bus master transfer requires programming the appropriate address registers and transfer byte counts. This can be done from either the PCI interface or the Add-On interface (configurable at reset, see section 11.3.1). Initiating bus master transfers from the add-on is advantageous because the host CPU does not have to intervene for the S5933 to become a PCI Initiator. At the end of a transfer the S5933 may generate an interrupt to either the PCI bus (for PCI initiated transfers) or Add-On interface (for Add-On initiated transfers).

11.1.4.1 Add-On Initiated Bus Mastering

If bit 7 in location 45h of an external serial non-volatile memory is zero, the Master Read Address Register (MRAR), Master Write Address Register (MWAR), Master Read Transfer Count (MRTC), and Master Write Transfer Count (MWTC) are accessible only from the Add-On interface. Add-On initiated bus mastering is not possible when a byte-wide boot device is used due to shared device pins (see Section 11.3.1). When configured for Add-On initiated bus mastering, the S5933 transfers data until the transfer count reaches zero, or it may be configured to ignore the transfer count.

For bus master transfers initiated by the Add-On interface, some applications may not know the size of the data block to be transferred. To avoid constantly updating the transfer count register, the transfer count may be disabled. Bit 28 in the Add-On General Control/Status Register (AGCSTS) performs this function. Disabling the transfer count also disables the interrupt capabilities. Regardless of whether Add-On transfer count is enabled or disabled, the Add-On Master Read Enable (AMREN) and Add-On Master Write Enable (AMWEN) inputs control when the S5933 asserts or deasserts its request to the PCI bus. When Add-On transfer count is enabled, the S5933 will only request the bus when both the transfer count (read or write) is not zero and the appropriate enable line (AMREN or AMWEN) is active. For Add-On initiated bus mastering, AMWEN and AMREN override the read and write bus mastering enable bits in the Bus Master Control/Status Register (MCSR).

11.1.4.2 PCI Initiated Bus Mastering

If bit 7 in location 45h of the external non-volatile memory is one, the Master Read Address Register (MRAR), Master Write Address Register (MWAR), Master Read Transfer Count (MRTC), and Master Write Transfer Count (MWTC) are accessible only from the PCI bus interface. In this configuration, the S5933 transfers data until the transfer count reaches zero. The transfer count cannot be disabled for PCI initiated bus mastering. If no external nv memory boot device is used, the S5933 defaults to PCI initiated bus mastering.

11.1.4.3 Address and Transfer Count Registers

The S5933 has two sets of registers used for bus master transfers. There are two operation registers for bus master read operations and two operation registers for bus master write operations. One operation register is for the transfer address (MWAR and MRAR). The other operation register is for the transfer byte count (MWTC and MRTC).

The address registers are written with the first address of the transfer before bus mastering is enabled. Once a transfer begins, this register is automatically updated to reflect the address of the current transfer. If a PCI target disconnects from an S5933 initiated cycle, the transfer is retried starting from the current address in the register. If bus grant (GNT#) is removed or bus mastering is disabled (using AMREN or AMWEN), the value in the address register reflects the next address to be accessed. Transfers must begin on DWORD boundaries.

The transfer count registers contain the number of bytes to be transferred. The transfer count may be written before or after bus mastering is enabled. If bus mastering is enabled, no transfer occurs until the transfer count is programmed with a non-zero value. Once a transfer begins, this register is automatically updated to reflect the number of bytes remaining to be transferred. If the transfer count registers are disabled (for Add-On initiated bus mastering), transfers begin as soon as bus mastering is enabled.

Although transfers must begin on DWORD boundaries, transfer counts do not have to be multiples of four bytes. For example, if the write transfer count (MWTC) register is programmed with a value of 10 (decimal), the S5933 performs two DWORD writes and a third write with only BE0# and BE1# asserted.

11.1.4.4 Bus Mastering FIFO Management Schemes

The S5933 provides flexibility in how the FIFO is managed for bus mastering. The FIFO management scheme determines when the S5933 requests the bus to initiate PCI bus cycles. The management scheme is configurable for the PCI to Add-On and Add-On to PCI FIFO (and may be different for each). Bus mastering must be enabled for the management scheme to apply (via the enable bits or AMREN/AMWEN).

For the PCI to Add-On FIFO, there are two management options. The PCI to Add-On FIFO management option is programmed through the Bus Master Control/Status Register (MCSR). The FIFO can be programmed to request the bus when any DWORD location is empty or only when four or more locations are empty. After the S5933 is granted control of the PCI bus, the management scheme does not apply. The device continues to read as long as there is an open FIFO location. When the PCI to Add-On FIFO is full or bus mastering is disabled, the PCI bus request is removed by the S5933.

For the Add-On to PCI FIFO, there are two management options. The Add-On to PCI FIFO management option is programmed through the Bus Master Control/Status Register (MCSR). The FIFO can be programmed to request the bus when any DWORD location is full or only when four or more locations are full. After the S5933 is granted control of the PCI bus, the management scheme does not apply. The device continues to write as long as there is data in the FIFO. When the Add-On to PCI FIFO is empty or bus mastering is disabled, the PCI bus request is removed by the S5933.

There are two special cases for the Add-On to PCI FIFO management scheme. The first case is when the FIFO is programmed to request the PCI bus only when four or more locations are full, but the transfer count is less than 16 bytes. In this situation, the FIFO ignores the management scheme and finishes transferring the data. The second case is when the S5933 is configured for Add-On initiated bus mastering with transfer counts disabled. In this situation, the FIFO management scheme must be set to request the PCI bus when one or more locations are full. AMREN and AMWEN may be used to implement a specific FIFO management scheme.

11.1.4.5 FIFO Bus Master Cycle Priority

In many applications, the FIFO is used as a PCI initiator performing both PCI reads and writes. This requires a priority scheme be implemented. What happens if the FIFO condition for initiating a PCI read and a PCI write are both met?

Bits D12 and D8 in the Bus Master Control/Status Register (MCSR) control the read and write cycle priority, respectively. If these bits are both set or both clear, priority alternates, beginning with a read cycle. If the read priority is set and the write priority is clear, read cycles take priority. If the write priority is set and the read priority is clear, write cycles take priority. Priority arbitration is only done when neither FIFO has control of the PCI bus (the PCI to Add-On FIFO would never interrupt an Add-On to PCI FIFO transfer).

11.1.4.6 FIFO Generated Bus Master Interrupts

Interrupts may be generated under certain conditions from the FIFO. If PCI initiated bus mastering is used, INTA# is generated to the PCI interface. If Add-On initiated bus mastering is used, IRQ# is generated to the Add-On interface. Interrupts may be disabled.

FIFO Interrupts may be generated from one or more of the following during bus mastering: read transfer count reaches zero, write transfer count reaches zero, or an error occurs during bus mastering. Error conditions include a target or master abort on the PCI bus. Interrupts on PCI error conditions are only enabled if one or both of the transfer count interrupts are enabled.

The Add-On Interrupt Control/Status Register (AINT) or the Interrupt Control Status Register (INTCSR) indicates the interrupt source. The interrupt service routine may read these registers to determine what action is required. As mailboxes are also capable of generating interrupts, this must also be considered in the service routine. Interrupts are also cleared through these registers.

11.2 BUS INTERFACE

The S5933 FIFO may be accessed from the Add-On interface or the PCI interface. Add-On FIFO control and status signals allow a simple interface to the FIFO with either an Add-On CPU or programmable logic. The following section describes the PCI and Add-On interface behavior and hardware interface.

11.2.1 FIFO PCI Interface (Target Mode)

The S5933 FIFO may act as a standard PCI target. FIFO empty/full status may be determined by the PCI initiator by reading the status bits in the PCI Bus Master Control/Status Register (MCSR).

The FIFO occupies a single 32-bit register location within the PCI Operation Registers. **A PCI initiator may not perform burst accesses on the FIFO**. Each data phase of a burst causes the PCI initiator to increment its address counter (even though only the first address is driven at the beginning of the burst). The initiator keeps track of the current address in case a disconnect occurs. This allows the initiator to continue the burst from where the disconnect occurred. If the S5933 FIFO initiated a disconnect during a PCI burst to the FIFO register, the burst would be resumed at an address other than the FIFO location (because the initiator address counter has incremented). The S5933 always signals a disconnect if a burst to any PCI Operation Register is attempted.

Because the PCI to Add-On FIFO and the Add-On to PCI FIFO occupy a single location within the PCI and Add-On Operation Registers, which FIFO is accessed is determined by whether the access is a read or write. This means that once data is written into the FIFO, the value written cannot be read back.

For PCI reads from the Add-On to PCI FIFO, the S5933 asserts TRDY# and completes the PCI cycle (Figure 11-3). If the PCI bus attempts to read an empty FIFO, the S5933 immediately issues a disconnect with retry (Figure 11-4). The Add-On to PCI FIFO status indicators change one PCI clock after a PCI read.

For PCI writes to the PCI to Add-On, the S5933 asserts TRDY# and completes the PCI cycle (Figure 11-5). If the PCI bus attempts to write a full FIFO, the S5933 immediately issues a disconnect with retry (Figure 11-6). The PCI to Add-On FIFO status indicators change one PCI clock after a PCI write.

11.2.2 FIFO PCI Interface (Initiator Mode)

The S5933 can act as an initiator on the PCI bus. This allows the device to gain control of the PCI bus to transfer data to or from the FIFO. Internal address and transfer count registers control the number of PCI transfers and the locations of the transfers. The following paragraphs assume the proper registers and bits are programmed to enable bus mastering (see section 11.3.3).

PCI read and write transfers from the S5933 are very similar. The FIFO management scheme (section 11.1.4.4) determines when the S5933 asserts its PCI bus request (REQ#). When bus grant (GNT#) is returned, the device begins running PCI cycles. Once the S5933 controls the bus, the FIFO management scheme is not important. It only determines when PCI bus control is initially requested. PCI bus reads and writes are always performed as bursts by the S5933, if possible.

Figure 11-3. PCI Read from a Full S5933 FIFO

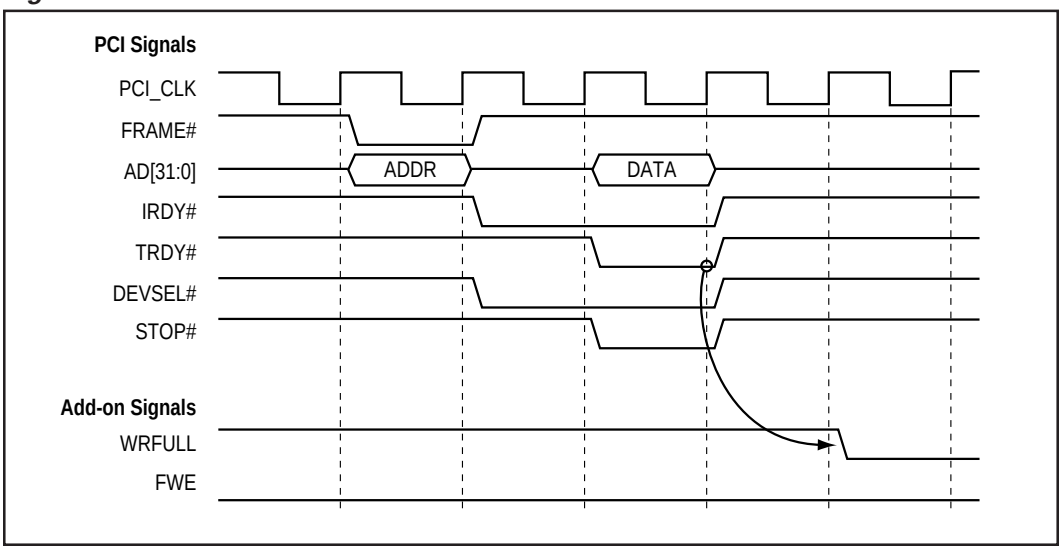


Figure 11-4. PCI Read from an Empty S5933 FIFO (Target Disconnect)

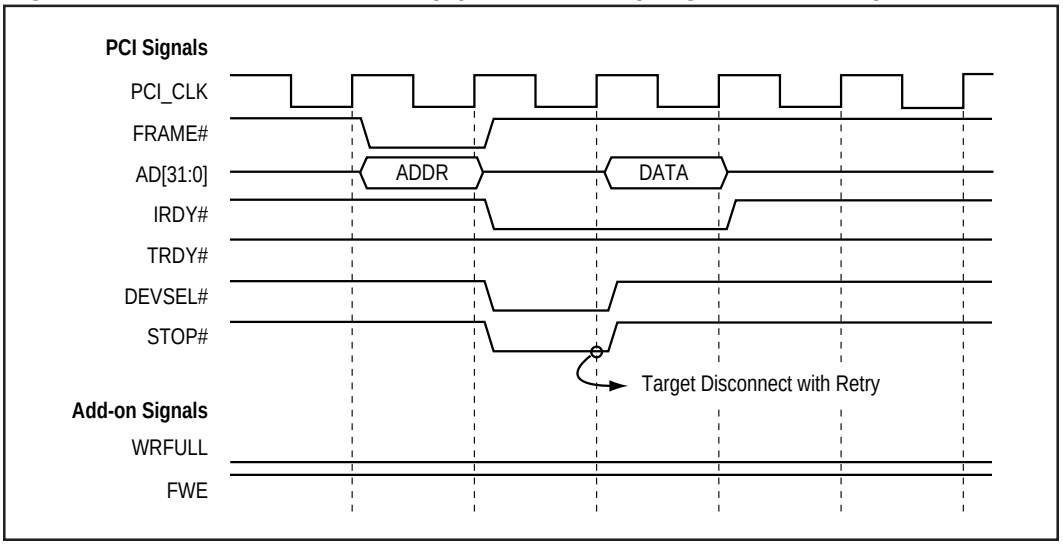


Figure 11-5. PCI Write to an Empty S5933 FIFO

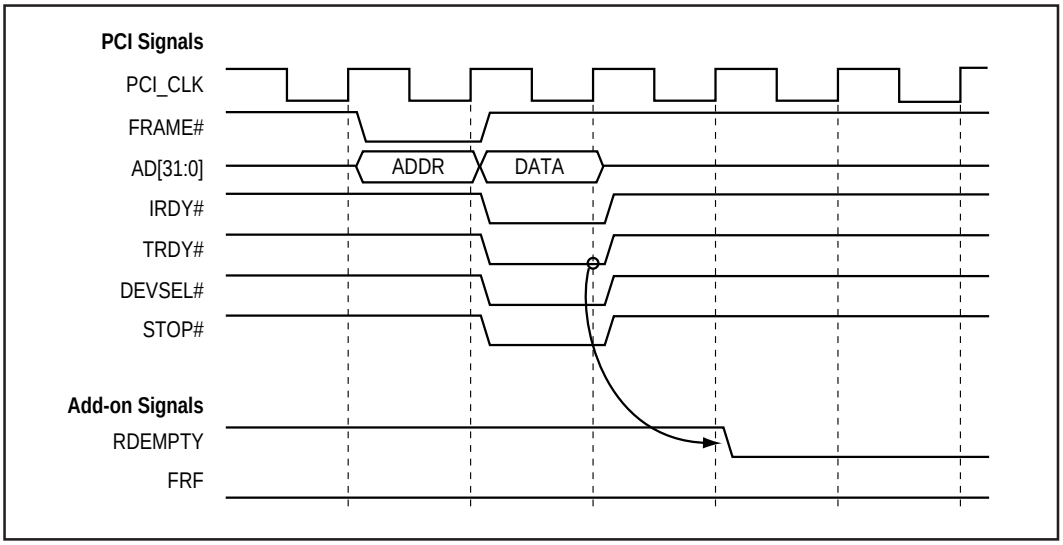
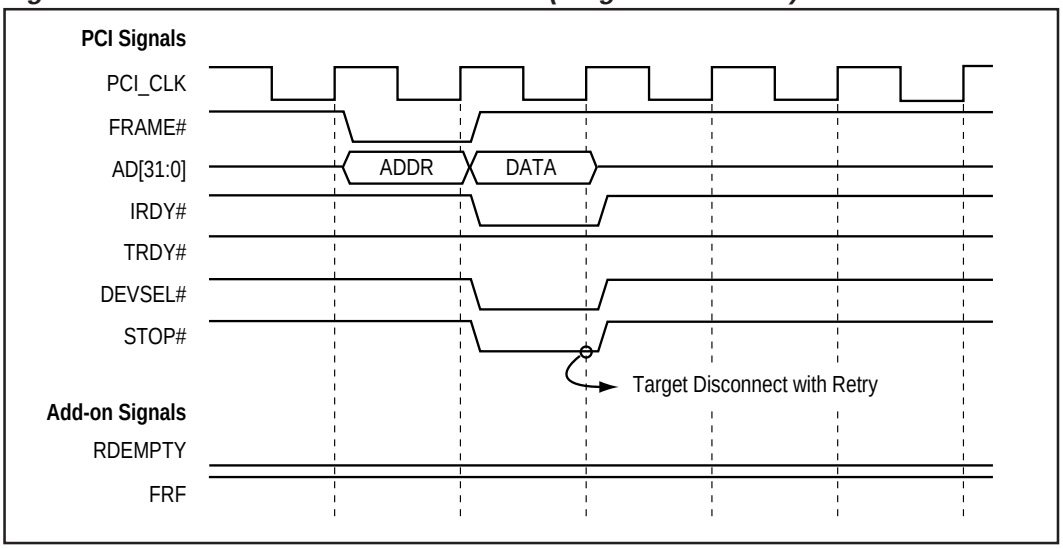


Figure 11-6. PCI Write to a Full S5933 FIFO (Target Disconnect)



11.2.2.1 FIFO PCI Bus Master Reads

For PCI read transfers (filling the PCI to Add-On FIFO), read cycles are performed until one of the following occurs:

- Bus Master Read Transfer Count Register (MRTC), if used, reaches zero
- The PCI to Add-On FIFO is full
- GNT# is removed by the PCI bus arbiter
- AMREN is deasserted (See section 11.1.4.1)

If the transfer count is not zero, GNT# remains asserted, and AMREN is asserted, the FIFO continues to read data from the PCI bus until there are no empty locations in the PCI to Add-On FIFO. If the Add-On can empty the FIFO as quickly as it can be filled from the PCI bus, very long bursts are possible. The S5933 deasserts REQ# when it completes the access to fill the last location in the FIFO. Once REQ# is deasserted, it will not be reasserted until the FIFO management condition is met.

11.2.2.2 FIFO PCI Bus Master Writes

For PCI write transfers (emptying the Add-On to PCI FIFO), write cycles are performed until one of the following occurs:

- Bus Master Write Transfer Count Register (MWTC), if used, reaches zero
- The Add-On to PCI FIFO is empty
- GNT# is removed by the PCI bus arbiter
- AMWEN is deasserted (See section 11.1.4.1)

If the transfer count is not zero, GNT# remains asserted, and AMWEN is asserted, The FIFO continues to write data to the PCI bus until there are is no data in the Add-On to PCI FIFO. If the Add-On can fill the FIFO as quickly as it can be emptied to the PCI bus, very long bursts are possible. The S5933 deasserts REQ# when it completes the access to transfer the last data in the FIFO. Once REQ# is deasserted, it will not be reasserted until the FIFO management condition is met.

11.2.3 Add-On Bus Interface

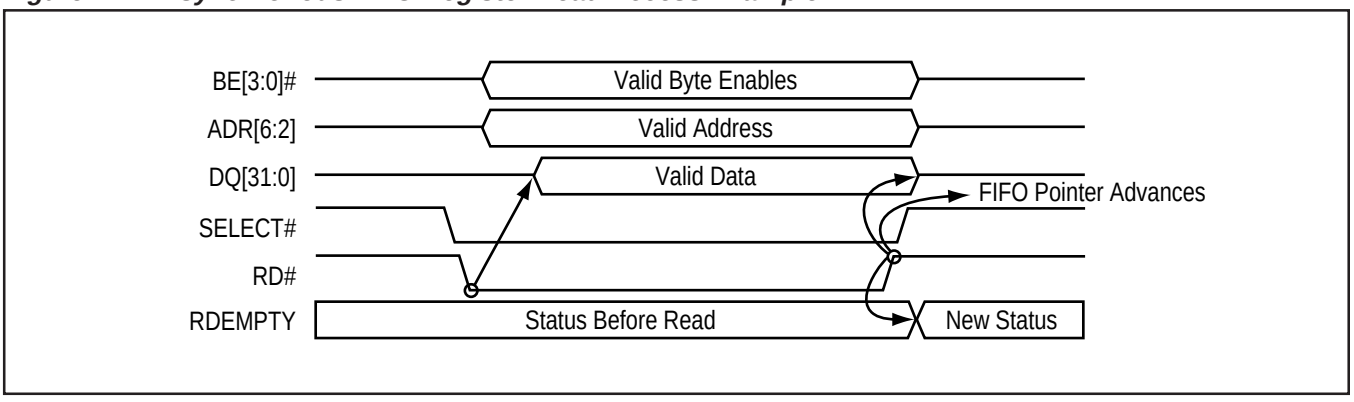
The FIFO register may be accessed in two ways from the Add-On interface. It can be accessed through normal register accesses or directly with the RDFIFO# and WRFIFO# inputs. In addition, the FIFO register can be accessed with synchronous or asynchronous to BPCLK, depending on the S5933 configuration. The Add-On interface also supports datapaths which are not 32-bits. The method used to access the FIFO from the Add-On interface is independent of whether the FIFO is a PCI PCI target or a PCI initiator.

11.2.3.1 Add-On FIFO Register Accesses

The FIFO may be accessed from the Add-On interface through the Add-On FIFO Port Register (AFIFO) read or write. This is offset 20h in the Add-On Operation Registers. Depending on the device configuration, this register can be accessed either synchronous BPCLK or asynchronous to BPCLK. To access the FIFO as a normal Add-On Operation Register, ADR[6:2], BE[3:0]#, SELECT#, and RD# or WR# are required. The major differences between synchronous and asynchronous modes are when the FIFO pointers advance and the ability to perform burst accesses. The following examples are shown for Add-On FIFO reads. FIFO write waveforms are shown in Chapter 13.

Figure 11-7 shows an asynchronous FIFO register read. SELECT# must meet setup and hold times relative to the rising edge of RD#. RD# and SELECT# both asserted enables the DQ outputs, and the first data location in the FIFO is driven onto the bus. The FIFO address and the byte enables must be valid before valid data is driven onto the DQ bus. Data remains valid as long as the address, byte enables, SELECT# and RD# are asserted. Deasserting RD# or SELECT# causes the data bus to float. The rising edge of RD# causes the FIFO pointer to advance. The status outputs are updated to reflect the FIFO condition after it advances.

Figure 11-7. Asynchronous FIFO Register Read Access Example



When the last location in the PCI to Add-On FIFO is read by the Add-On, the FIFO pointer does not change. If another read is performed before more data enters the FIFO, the previous data is driven. When a write to a full Add-On to PCI FIFO is attempted, nothing happens. No FIFO data is overwritten and the FIFO pointers are not changed. This behavior is the same whether the FIFO is accessed using the direct access inputs or normal Operation Register accesses.

Figure 11-8 shows a synchronous FIFO register burst access. SELECT# must meet setup and hold times relative to the rising edge of BPCLK. RD# and SELECT# both asserted enables the DQ outputs, and the first data location (data 0) in the FIFO is driven on to the bus. The FIFO address and the byte enables must be valid before valid data is driven onto the DQ bus. Data 0 remains valid until the next rising edge of BPCLK. The rising edge of BPCLK causes the FIFO pointer to advance to the next location (data 1). The next rising edge of BPCLK also advances the FIFO pointer to the next location (data 2). The status outputs reflect the FIFO condition after it advances, and are updated off of the rising edge of BPCLK. When RD# or SELECT# is deasserted, the DQ bus floats. The next time a valid FIFO access occurs and RD# and SELECT# are asserted, data 2 is presented on the DQ bus (as there was no BPCLK edge to advance the FIFO).

11.2.3.2 Add-On FIFO Direct Access Mode

Instead of generating an address, byte enables, SELECT# and a RD# or WR# strobe for every FIFO access, the S5933 allows a simple, direct access mode. Using RDFIFO# and WRFIFO# is functionally identical to performing a standard AFIFO Port Register access, but requires less logic to implement. Accesses to the FIFO register using the direct access signals are always 32-bits wide. The only exception to this is when the MODE pin is configured for 16-bit operation. In this situation, all accesses are 16-bits wide (see Section 11.2.3.5). The RD# and WR# inputs must be inactive when RDFIFO# or WRFIFO# is active. The ADR[6:2] and BE[3:0]# inputs are ignored.

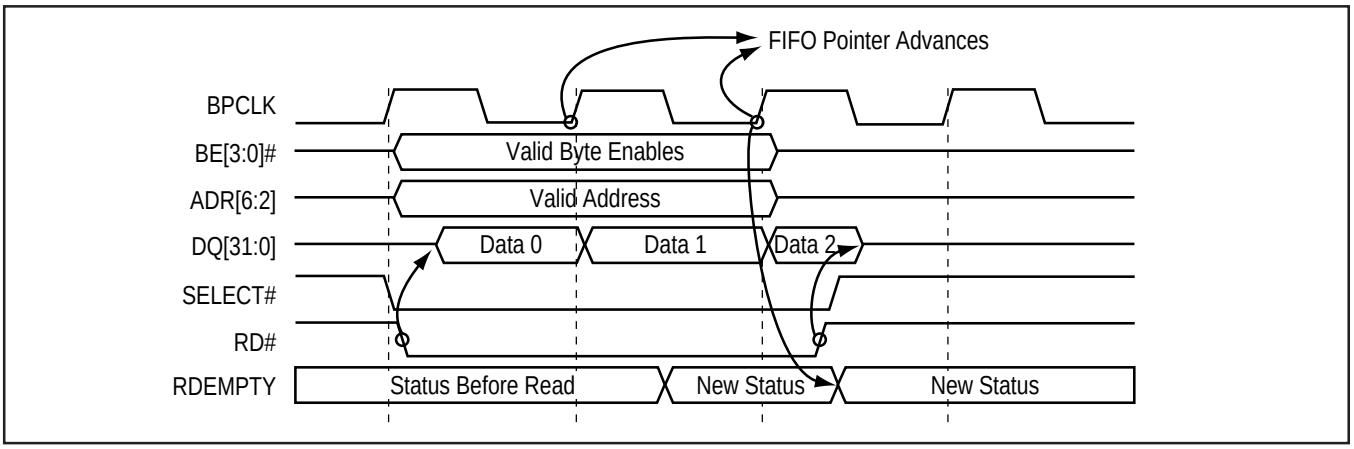
Depending on the device configuration, RDFIFO# and WRFIFO# can act as clocks for data or enables with BPCLK acting as the clock. A Synchronous interface allows higher data rates, and an asynchronous interface is better for slow Add-On logic which may require wait states. The major difference between the synchronous and asynchronous modes is when the FIFO advances.

Figure 11-9 shows an asynchronous FIFO register direct access using RDFIFO#. The first location in the FIFO is driven onto the bus when RDFIFO# is asserted. Data remains valid as long as RDFIFO# is asserted. The rising edge of RDFIFO# causes the data bus to float and acts as the clock causing the FIFO pointer to advance. The status outputs reflect the FIFO condition after it advances.

Figure 11-10 shows a synchronous FIFO register direct burst access using RDFIFO#. RDFIFO# acts as an enable and the first data location (data 0) in the FIFO is driven on to the bus when RDFIFO# is asserted. Data 0 remains valid until the next rising edge of BPCLK. The rising edge of BPCLK causes the FIFO pointer to advance to the next location (data 1). The next rising edge of BPCLK advances the FIFO pointer to the next location (data 2). The status outputs reflect the FIFO condition after it advances, and are updated off of the rising edge of BPCLK. When RDFIFO# is deasserted, the DQ bus floats. The next time RDFIFO# is asserted, data 2 is presented on the DQ bus (as there was no BPCLK edge to advance the FIFO).

A synchronous FIFO interface has the advantage of allowing data to be accessed more quickly (in bursts) by the Add-On. As a target, if a full S5933 FIFO is

Figure 11-8. Synchronous FIFO Register Burst Read Access Example



written (or an empty FIFO is read) by a PCI initiator, the S5933 requests a retry. The faster the Add-On interface can empty (or fill) the FIFO, the less often retries occur. With the S5933 as a PCI initiator, a similar situation occurs. Not emptying or filling the FIFO quickly enough results in the S5933 giving up control of the PCI bus. Higher PCI bus data transfer rates are possible through the FIFO with a synchronous interface.

11.2.3.3 Additional Status/Control Signals for Add-On Initiated Bus Mastering

If a serial non-volatile memory is used to configure the S5933, and the device is configured for Add-On initiated bus mastering, two additional FIFO status signals and four additional control signals are available to the Add-On interface. The FRF and FWE outputs provide additional FIFO status information. Inputs FRC#, FWC#, AMREN, and AMWEN provide additional FIFO control. Applications may use these signals to monitor/control FIFO flags and PCI bus requests. These new signals are some of the lines that were used for byte-wide nvram interface, but now are reconfigured. The reconfigured lines are as follows:

Outputs:

E_ADDR (15) FRF
FIFO Read Full: Indicates that the PCI to Add-On FIFO is full.

E_ADDR (14) FWE
FIFO Write Empty: Indicates that the Add-On to PCI FIFO is empty.

Inputs:

EQ (7) AMWEN
Add-On bus Mastering Write ENable: This input is driven high to enable bus master writes.

EQ (6) AMREN
Add-On bus Mastering Read ENable: This input is driven high to enable bus master reads.

EQ (5) FRC#
FIFO Read Clear: This line is driven low to clear the PCI to Add-On FIFO.

EQ (4) FWC#
FIFO Write Clear: This line is driven low to clear the Add-On to PCI FIFO.

FRF (PCI to Add-On FIFO full) and FWE (Add-On to PCI FIFO empty) supplement the RDEMPTY and WRFULL status indicators. These additional status outputs provide additional FIFO status information for Add-On FIFO control logic.

Figure 11-9. Asynchronous FIFO Register RDFIFO# Access Example

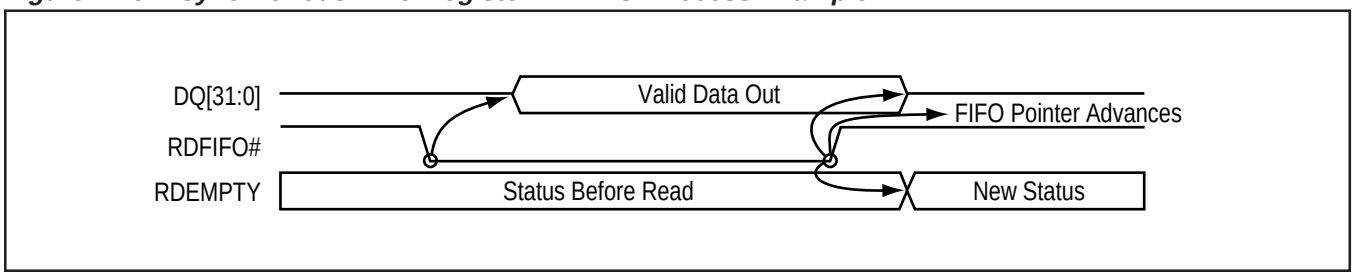
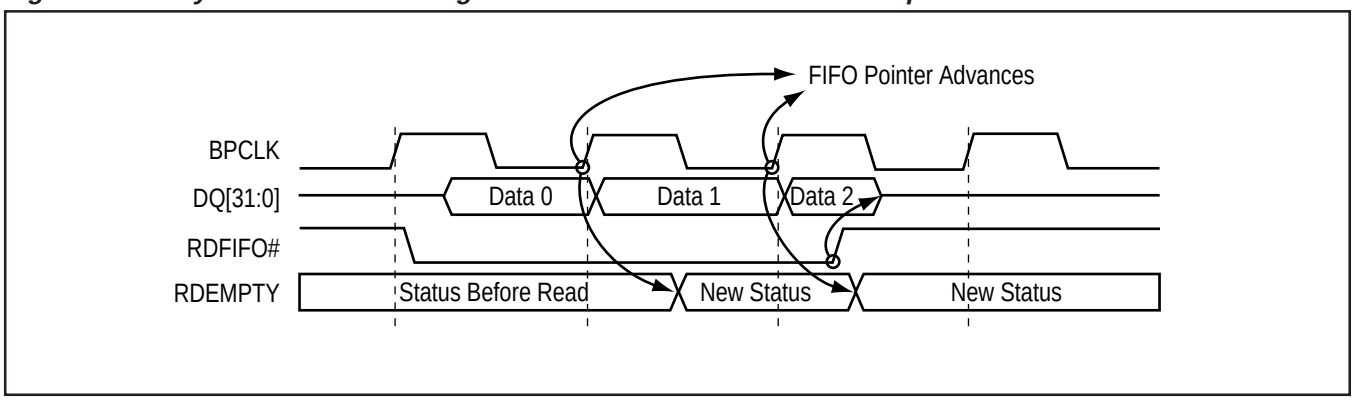


Figure 11-10. Synchronous FIFO Register Burst RDFIFO# Access Example



The FRC# and FWC# inputs allow Add-On logic to reset the PCI to Add-On or Add-On to PCI FIFO flags. The FIFO flags can always be reset with software through the Add-On General Control/Status Register (AGCSTS) or the Bus Master Control/Status Register (MCSR), but these hardware inputs are useful for designs which do no implement a CPU on the Add-On card. Asserting the FRC# input resets the PCI to Add-On FIFO. Asserting the FWC# input resets the Add-On to PCI FIFO.

The AMREN and AMWEN inputs allow Add-On logic to individually enable and disable bus mastering for the PCI to Add-On and Add-On to PCI FIFO. These inputs override the Bus Master Control/Status Register (MCSR) bus master enable bits. The S5933 may request the PCI bus for the PCI to Add-On FIFO when AMREN is asserted and may request the PCI bus for the Add-On to PCI FIFO when AMWEN is asserted. If AMREN or AMWEN is deasserted, the S5933 removes its PCI bus request and gives up control of the bus.

AMREN and AMWEN are useful for Add-Ons with external FIFOs cascaded into the S5933. For PCI bus master write operations, the entire S5933 Add-On to PCI FIFO and the external FIFO may be filled before enabling bus mastering, providing a single long burst write rather than numerous short bursts.

In some applications, the amount of data to be transferred is not known. During read operations, the S5933, attempting to fill its PCI to Add-On FIFO, may access up to eight memory locations beyond what is required by the Add-On before it stops. In this situation, AMREN can be deasserted to disable PCI reads, and then FRC# can be asserted to flush the unwanted data from the FIFO.

11.2.3.4 FIFO Generated Add-On Interrupts

For Add-On initiated bus mastering, the S5933 may be configured to generate interrupts to the Add-On interface for the following situations:

- Read transfer count reaches zero
- Write transfer count reaches zero
- An error occurred during the bus master transaction

The interrupt is posted to the Add-On interface with the IRQ# output. A high-to-low transition on this output indicates an interrupt condition. Because there is a single interrupt output and multiple interrupt conditions, the Add-On Interrupt Control/Status Register (AINT) must be read to determine the interrupt source. This register is also used to clear the interrupt, returning IRQ# to its high state. If mailbox interrupts are also used, this must be considered in the interrupt service routine.

11.2.3.5 8-Bit and 16-Bit FIFO Add-On Interfaces

The S5933 FIFO may also be used to transfer data between the PCI bus and 8-bit or 16-bit Add-On interfaces. This can be done using FIFO advance conditions or the S5933 MODE input pin.

The FIFO may be used as an 8-bit or 16-bit wide FIFO. To use the FIFO as an 8-bit interface, the advance condition should be set for byte 0 (no data is transferred in the upper 3 bytes). To use the FIFO as a 16-bit interface, the advance condition should be set for byte 1 (no data is transferred in the upper 2 bytes). This allows a simple Add-On bus interface, but it has the disadvantage of not efficiently utilizing the PCI bus bandwidth because the host is forced to perform 8-bit or 16-bit accesses to the FIFO on the PCI bus. This is the only way to communicate with an 8-bit Add-On through the FIFO without additional logic to steer byte lanes on the Add-On data bus. Pass-Thru mode is more suited to 8-bit Add-On interfaces.

Implementing a 16-bit wide FIFO is a reasonable solution, but to avoid wasting PCI bus bandwidth, the best method is to allow the PCI bus and the FIFO to operate with 32-bit data. The S5933 can assemble or disassemble 32-bit quantities for the Add-On interface. This is possible through the MODE pin. When MODE is low, the Add-On data bus is 32-bits. When MODE is high, the Add-On data bus is 16-bits. When MODE is configured for 16-bit operation, BE3# becomes ADR1.

With the FIFO direct access signals (RDFIFO# and WRFIFO#), the MODE pin must reflect the actual Add-On data bus width. With MODE = 16-bits, the S5933 automatically takes two consecutive, 16-bit Add-On writes to the FIFO and assembles a 32-bit value. FIFO reads operate in the same manner. Two consecutive Add-On reads empty the 32-bit FIFO register. The 16-bit data bus is internally steered to the lower and upper words of the 32-bit FIFO register.

One consideration needs to be taken when using the FIFO direct access signals and letting the S5933 do byte lane steering internally. The default condition used to advance the FIFO is byte 0. This must be changed to byte 2 or 3. When MODE is configured for a 16-bit Add-On bus, the first 16-bit cycle to the FIFO always accesses the low 16-bits. If the FIFO advance condition is left at byte 0, the FIFO advances after the first 16-bit cycle and the data in the upper 16-bits is directed to the next FIFO location, shifting the data.

Some applications hold the RDFIFO# and WRFIFO# inputs active for a synchronous interface. In 16-bit mode, designs must avoid writing to a full FIFO. The data for the write is lost, but the internal mechanism to direct the 16-bit external data bus to the upper 16-bits of the FIFO register is triggered. This creates a situation where the FIFO is out of step. The next 16-bit FIFO write is directed to the upper 16-bits of the FIFO, and the FIFO advances incorrectly. The WRFULL output should be used to gate the WRFIFO# input to avoid this situation. A similar problem can occur if Add-On logic attempts to read an empty FIFO in 16-bit mode. RDEMPTY should be used to gate the RDFIFO# input to avoid problems with the FIFO getting out of step. In 32-bit mode (MODE = low), these situations do not occur.

If FIFO accesses are done without the direct access signals with MODE configured for 16-bits (using ADR, SELECT#, etc.), external hardware must toggle ADR1 between consecutive 16-bit bus cycles. The FIFO advance condition must be set to correspond to the order the application accesses the upper and lower words in the FIFO register.

11.3 CONFIGURATION

The FIFO configuration takes place during initialization and during operation. During initialization, the FIFO hardware interface method and bus master register access rights are defined. During operation, FIFO advance conditions, endian conversion, and bus mastering capabilities are defined. The following section describes the bits and registers which are involved with controlling and monitoring FIFO operation.

11.3.1 FIFO Setup During Initialization

Location 45h in an external non-volatile memory may be used to configure the S5933 FIFO during initialization. If no external non-volatile memory is used, the S5933 defaults to PCI initiated bus master transfers with asynchronous operation for FIFO accesses.

The value of bit 7 in location 45h determines if the address and transfer count registers used in bus mastering are accessible from the PCI bus or from the Add-On bus. Once the configuration information is downloaded from non-volatile memory after reset, the bus mastering initialization method can not be changed. Access to the bus master address and transfer count registers cannot be alternated between the PCI bus and the Add-On interface during operation.

Bits 6 and 5 in location 45h determine if FIFO register accesses using the RDFIFO#, WRFIFO#, RD# and WR# inputs operate asynchronous or synchronous to BPCLK. For asynchronous operation, RDFIFO#, WRFIFO#, RD# and WR# operate as clocks for data. For synchronous operation, RDFIFO#, WRFIFO#, RD# and WR# operate as enables, using BPCLK to clock data. Synchronous operation allows higher data transfer rates.

Location 45h Configuration Bits

Bit 7 Bus Master Register Access

0 Address and transfer count registers only accessible from the Add-On interface

1 Address and transfer count registers only accessible from the PCI interface (default)

Bit 6 RDFIFO#, RD# Operation

0 Synchronous Mode - RDFIFO# and RD# functions as enables

1 Asynchronous Mode - RDFIFO# and RD# functions as clocks (default)

Bit 5 WRFIFO#, WR# Operation

0 Synchronous Mode - WRFIFO# and WR# functions as enables

1 Asynchronous Mode - WRFIFO# and WR# functions as clocks (default)

Bit 0 Target Latency Timer Enable

0 Disable PCI Latency Timer Time Out - Will not disconnect with retry if cannot issue TRDY in specified time

1 Enable PCI Latency Timer Time Out - Will be PCI 2.1 compliant

11.3.2 FIFO Status and Control Bits

The FIFO status can be monitored and the FIFO operation controlled from the PCI Operation Registers and/or the Add-On Operation Registers. The FIFO register resides at offset 20h in the PCI and Add-On Operation Registers.

The Bus Master Control/Status (MCSR) PCI Operation register allows a PCI host to monitor FIFO activity and control FIFO operation. Reset controls allow the PCI to Add-On FIFO and Add-On to PCI FIFO flags to be reset (individually). Status bits indicate if the PCI to Add-On FIFO is empty, has four or more open spaces, or is full. Status bits also indicate if the Add-On to PCI FIFO is empty, has four or more full locations or is full. Finally, FIFO PCI bus mastering is monitored/controlled though this register (see Section 11.3.3).

The Add-On General Control/Status (AGCSTS) Add-On Operation Register allows an Add-On CPU to monitor FIFO activity and control FIFO operation. Reset controls allow the PCI to Add-On FIFO and Add-On to PCI FIFO flags to be reset (individually). Status bits indicate if the PCI to Add-On FIFO is empty, has four or more open spaces, or is full. Status bits also indicate if the Add-On to PCI is empty, has four or more full spaces or is full. FIFO bus mastering status may be monitored through this register, but all bus master configuration is through the MCSR PCI Operation Register (see Section 11.3.3).

11.3.3 PCI Initiated FIFO Bus Mastering Setup

For PCI initiated bus mastering, the PCI host sets up the S5933 to perform bus master transfers. The following tasks must be completed to setup FIFO bus mastering:

1) Define interrupt capabilities. The PCI to Add-On and/or Add-On to PCI FIFO can generate a PCI interrupt to the host when the transfer count reaches zero.

INTCSR	Bit 15	Enable Interrupt on read transfer count equal zero
INTCSR	Bit 14	Enable Interrupt on write transfer count equal zero

2) Reset FIFO flags. This may not be necessary, but if the state of the FIFO flags is not known, they should be initialized.

MCSR	Bit 26	Reset Add-On to PCI FIFO flags
MCSR	Bit 25	Reset PCI to Add-On FIFO flags

3) Define FIFO management scheme. These bits define what FIFO condition must exist for the PCI bus request (REQ#) to be asserted by the S5933.

MCSR	Bit 13	PCI to Add-On FIFO management scheme
MCSR	Bit 9	Add-On to PCI FIFO management scheme

4) Define PCI to Add-On and Add-On to PCI FIFO priority. These bits determine which FIFO has priority if both meet the defined condition to request the PCI bus. If these bits are the same, priority alternates, with read accesses occurring first.

MCSR	Bit 12	Read vs. write priority
MCSR	Bit 8	Write vs. read priority

5) Define transfer source/destination address. These registers are written with the first address that is to be accessed by the S5933. These address registers are updated after each access to indicate the next address to be accessed. Transfers must start on DWORD boundaries.

MWAR	All	Bus master write address
MRAR	All	Bus master read address

6) Define transfer byte counts. These registers are written with the number of bytes to be transferred. The transfer count does not have to be a multiple of four bytes. These registers are updated after each transfer to reflect the number of bytes remaining to be transferred.

MWTC	All	Write transfer byte count
MRTC	All	Read transfer byte count

7) Enable Bus Mastering. Once steps 1-6 are completed, the FIFO may operate as a PCI bus master. Read and write bus master operation may be independently enabled or disabled.

MCSR	Bit 14	Enable PCI to Add-On FIFO bus mastering
MCSR	Bit 10	Enable Add-On to PCI FIFO bus mastering

The order of the tasks listed above is not particularly important. It is recommended that bus mastering be enabled as the last step. Some applications may choose to leave bus mastering enabled and start transfers by writing a non-zero value to the transfer count registers. This also works, provided the entire transfer count is written in a single access. As a number of the configuration bits and the two enable bits are all in the MCSR register, it may be most efficient for the FIFO configuration bits to be set with the same register access that enables bus mastering.

If interrupts are enabled, a host interrupt service routine is also required. The service routine determines the source of the interrupt and resets the interrupt. As mailbox registers may also be configured to generate interrupts, the exact source of the interrupt is indicated in the PCI Interrupt Control/Status Register (INTCSR). Typically, the interrupt service routine is used to setup the next transfer by writing new addresses and transfer counts, but some applications may also require other actions. If read transfer or write transfer complete interrupts are enabled, master and target abort interrupts are automatically enabled. These indicate a transfer error has occurred. Writing a one to these bits clears the corresponding interrupt.

INTCSR	Bit 21	Target abort caused interrupt
INTCSR	Bit 20	Master abort caused interrupt
INTCSR	Bit 19	Read transfer complete caused interrupt
INTCSR	Bit 18	Write transfer complete caused interrupt

11.3.4 Add-On Initiated FIFO Bus Mastering Setup

For Add-On initiated bus mastering, the Add-On sets up the S5933 to perform bus master transfers. The following tasks must be completed to setup FIFO bus mastering:

1) Define transfer count abilities. For Add-On initiated bus mastering, transfer counts may be either enabled or disabled. Transfer counts for read and write operations cannot be individually enabled.

AGCSTS	Bit 28	Enable transfer count for read and write bus master transfers

2) Define interrupt capabilities. The PCI to Add-On and/or Add-On to PCI FIFO can generate an interrupt to the Add-On when the transfer count reaches zero (if transfer counts are enabled).

AINT	Bit 15	Enable interrupt on read transfer count equal zero
AINT	Bit 14	Enable interrupt on write transfer count equal zero

3) Reset FIFO flags. This may not be necessary, but if the state of the FIFO flags is not known, they should be initialized.

AGCSTS	Bit 25	Reset Add-On to PCI FIFO flags
AGCSTS	Bit 26	Reset PCI to Add-On FIFO flags

4) Define FIFO management scheme. These bits define what FIFO condition must exist for the PCI bus request (REQ#) to be asserted by the S5933. This must be programmed through the PCI interface.

MCSR	Bit 13	PCI to Add-On FIFO management scheme
MCSR	Bit 9	Add-On to PCI FIFO management scheme

5) Define PCI to Add-On and Add-On to PCI FIFO priority. These bits determine which FIFO has priority if both meet the defined condition to request the PCI bus. If these bits are the same, priority alternates, with read accesses occurring first. This must be programmed through the PCI interface.

MCSR	Bit 12	Read vs. write priority
MCSR	Bit 8	Write vs. read priority

6) Define transfer source/destination address. These registers are written with the first address that is to be accessed by the S5933. These address registers are updated after each access to indicate the next address to be accessed. Transfers must start on DWORD boundaries.

MWAR	All	Bus master write address
MRAR	All	Bus master read address

7) Define transfer byte counts. These registers are written with the number of bytes to be transferred. The transfer count does not have to be a multiple of four bytes. These registers are updated after each transfer to reflect the number of bytes remaining to be transferred. If transfer counts are disabled, these registers do not need to be programmed.

MWTC	All	Write transfer byte count
MRTC	All	Read transfer byte count

8) Enable Bus Mastering. Once steps 1-7 are completed, the FIFO may operate as a PCI bus master. Read and write bus master operation may be independently enabled or disabled. The AMREN and AMWEN inputs control bus master enabling for Add-On initiated bus mastering. The MCSR bus master enable bits are ignored for Add-On initiated bus mastering.

It is recommended that bus mastering be enabled as the last step. Some applications may choose to leave bus mastering enabled (AMREN and AMWEN asserted) and start transfers by writing a non-zero value to the transfer count registers (if they are enabled).

If interrupts are enabled, an Add-On CPU interrupt service routine is also required. The service routine determines the source of the interrupt and resets the interrupt. As mailbox registers may also be configured to generate interrupts, the exact source of the interrup is indicated in the Add-On Interrupt Control Register (AINT). Typically, the interrupt service routine is used to setup the next transfer by writing new addresses and transfer counts (if enabled), but some applications may also require other actions. If read transfer or write transfer complete interrupts are enabled, the master/target abort interrupt is automatically enabled. These indicate a transfer error has occurred. Writing a one to these bits clears the corresponding interrupt.

AINT	Bit 21	Master/target abort caused interrupt
AINT	Bit 19	Read transfer complete caused interrupt
AINT	Bit 18	Write transfer complete caused interrupt