XILINX®

PowerPC 60X Bus Interface to a Virtex-E Device

Author: Steve Trynosky

Summary

This application note describes a reference design using a PowerPC™ 60X bus interface with interfaces to Synchronous Static RAM (SSRAM) and flash memory. The design supports two PowerPC 60X bus microprocessors (PowerPC 750 and 750CX) and implements a pipelined address bus and split address/data transactions on the 60X bus.

This reference design uses a processor bus functional model to verify the 60X bus interface to a memory system. Having the capability to generate bus traffic and look inside the Virtex™-E device, in a simulation environment, resolves system issues during the course of a complex system development. Design approaches using Virtex-E FPGAs accommodate evolutionary changes in microprocessor bus protocol, memory, and I/O standards through the ability to reuse and reprogram the design.

Support for evolving microprocessors, memory device densities, I/O standards and higher system bus speeds can be accomplished through design reuse and modification, without waiting for the development of a custom controller from an external supplier. The Virtex-E architecture provides bus keeper I/O and eliminates the requirement for external pull-up resistors on the address, data, and control bus signals to maintain the bus at valid logic levels when not active.

Introduction

Figure 1 shows a block diagram of the reference controller design. The controller interfaces the PowerPC to an external SSRAM and flash memory. Each processor supports one level of address pipelining and out-of-order bus transactions. Each bus master competes for system resources through a central arbiter implemented in the Virtex-E device. Single-beat and burst data transfers for memory accesses and memory-mapped I/O operations are supported. Single-beat transfers are also used for byte and misaligned memory accesses. Burst cycles are used for cache-line operations. The PowerPC, SSRAM, and flash memory are connected in Big Endian mode.

The smallest device that the design fits into is an XCV400E-8-BG432. All larger sized Virtex-E devices will also work, providing additional room to add additional design features to the 60X bus interface. Performance capabilities of the design are currently limited to the speed of the flow through SSRAM device. The SSRAM maximum clock rate is 100 MHz, so both the 60X bus and memory bus operate at that clock rate.
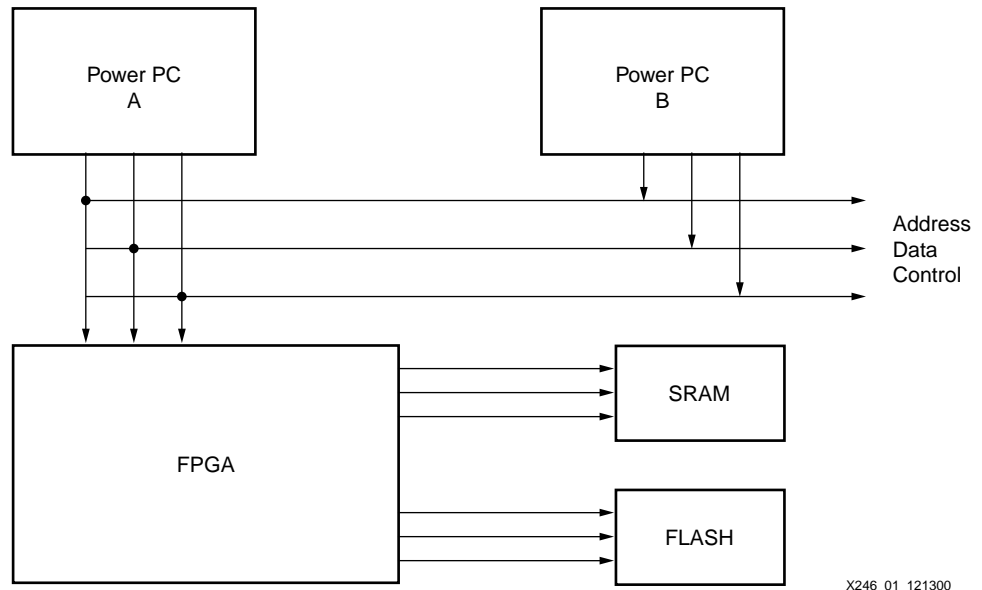
Figure 1: **60X Bus Memory Controller Block Diagram**

The PowerPC 750 is a reduced instruction set computer (RISC) microprocessor. The PowerPC 750 implements the 32-bit portion of the PowerPC architecture, which provides 32-bit effective addresses; 8-, 16-, and 32-bit integer data; and single- and double-precision floating point data. The PowerPC 750, a superscalar processor, completes two instructions simultaneously and contains six execution units: the floating point unit (FPU), a branch processing unit (BPU), a system register unit (SRU), a load/store unit (LSU), and two integer units (IU).

PowerPC System Bus Interface

The PowerPC 750 has both 60X and Level2 (L2) cache bus interfaces. The PowerPC 750CX is a reduced pin count package that provides an internal 256KB L2 cache. However, this reference design provides an interface to the 60X bus and does not address the L2 cache requirements of a system design. The 60X bus interface signals are shown grouped in Figure 2. For detailed descriptions of the PowerPC bus operation and interface signal definition, refer to the *PowerPC 750 RISC Microprocessor User's Manual*.

A system interface transfers data and instructions between the processor and the system memory. Because first level caches on the PowerPC 750CX are write-back caches, burst-read memory operations are the most common memory accesses, followed by burst-write memory operations and single-beat (noncacheable or write-through) memory read and write operations. The PowerPC supports address-only operations for atomic and global memory operations that are snooped by another microprocessor. Address retry is supported when a snooped read access hits a modified block in cache.

The address and data buses operate independently. Address and data tenures for a memory access are decoupled to permit address pipelining and split bus operations. Address pipelining permits the address tenure of one transaction to overlap the data tenure of another. The extent of the pipelining depends on the controller implementation. This reference design has a pipeline depth of one; however, deeper pipelines can be designed. While one data request is being serviced, the next request is waiting in the queue for the data bus to become available. The PowerPC supports split-bus transactions for systems containing multiple bus masters. One device can use the address bus, while another device uses the data bus. There are two types of memory accesses:

- Single-beat transfers. These memory accesses allow transfer sizes of 8, 16, 24, 32, or 64 bits in one bus clock cycle. Single-beat transactions are caused by noncacheable

read/write operations that access memory directly, cache-inhibited operations, cacheable or write-through memory operations.

- Four-beat burst instruction/data transfers always transfer an entire cache block of 32 bytes.

60X Bus System Interface Signal Description

The system interface is divided into groups called address bus arbitration, address start, transfer attributes, address transfer, address termination, data bus arbitration, data transfer, and data bus termination, as shown in Figure 2.
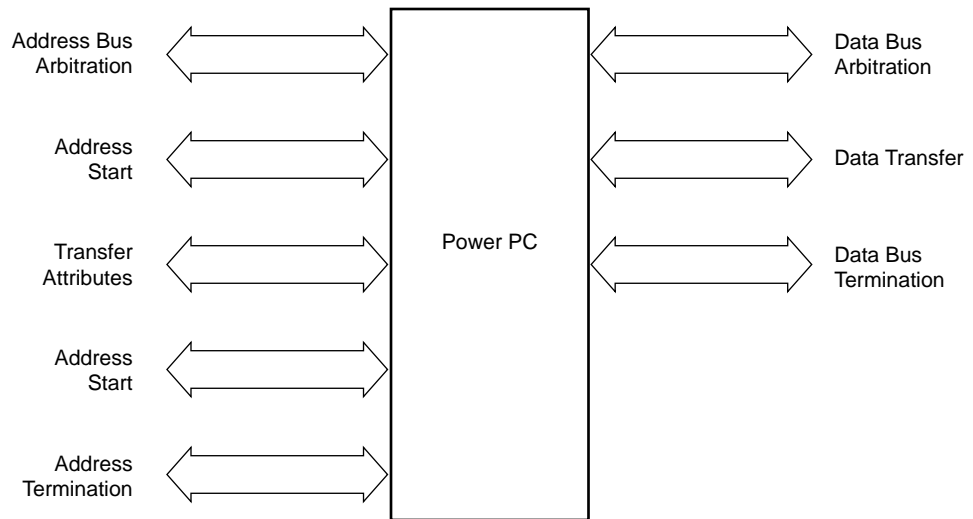
X246_02_121300

Figure 2: **PowerPC System Interface**

- Address Bus Arbitration: Each processor has a unique address bus request, N_BR, and address bus grant, N_BG, signal used for address bus arbitration.

- Address Start: The transfer start signal, N_TS, indicates that a valid address is on the bus.

- Transfer Attributes: The transfer type, TT[0:4], transfer size, TSIZ[0:2], and transfer burst, N_TBST, signals provide information about the type of transfer (read, write, or address only), the transfer size (1, 2, 3, 4, 8, or 32 byte), and single or burst memory access cycle.

- Address Transfer: The address bus, A[0:31], and address parity AP[0:3] signals are in this group.

- Address Termination: These signals are used to acknowledge the end of the address phase of a transaction and indicate whether a condition exists on the bus that requires the address phase to be repeated. The group includes address acknowledge, N_AACK, and address retry N_ARTRY.

- Data Bus Arbitration: The data bus grant signals, N_DBGA and N_DBGB, are in this group.

- Data Transfer: The data bus, DH[0:31] and DL[0:31], and the data parity, DP[0:7], signals are in this group. The PowerPC is a Big Endian machine.

- Data Bus Termination: These signals are required after each data beat in a data transfer, indicating whether a condition exists on the data bus that requires the data bus phase to be retried or extended. The transfer acknowledge, N_TA, and transfer error acknowledge, N_TEA, signals are in this group. The data bus busy and data retry signals are optional signals that have been removed from the 750CX interface.

Split Address/Data Bus Operation

The processor must first arbitrate for the address bus with an address bus request. Once granted the bus, the processor begins the address bus tenure by asserting the transfer start, transfer attributes, and the address of the transfer. The controller monitors these signals and if a valid bus cycle is decoded, the address tenure is terminated with an address acknowledge. Once the address has been acknowledged, the next address tenure can begin. The processor must now arbitrate for the data bus. The transfer start address start signal is an implied data bus request. The processor waits for a data bus grant before starting the data bus tenure. Once granted the bus, the processor begins receiving or transmitting data on the bus. The Virtex-E device paces the data flow using the transfer acknowledge bus termination signal.

In a multiple processor system, a unique address bus request, address bus grant, and data bus grant signal are required for each processor.

SSRAM Memory

A Micron Semiconductor 8 Mb SYNCBURST™ flow-through SSRAM is used in the reference design. The memory is organized as 256 K x 72 bits. The memory has a 7.5 ns access time, 1.5 ns setup time and 0.5 ns hold times. The memory supports either 2.5 V or 3.3 V I/O. To permit connection of both the SSRAM and flash to the memory data bus, 3.3 V I/O levels are used. SSRAM memory device specifications are available on the web at: **http://www.micronsemi.com/datasheets/syncds.html.**

In order to implement a pipelined address bus, the Virtex-E device must provide the address to the external memory devices. The internal address counter of the SSRAM is used for burst accesses, and the counter operates in linear address mode. The design supports byte accesses and uses the byte write signals for misaligned transfers. The global write function is not used in the reference design. The PowerPC uses Big Endian bit naming conventions, while the memory model uses Little Endian. Instantiating the memory model in the testbench requires care to get the byte lanes properly aligned with the write enable signals.

A Micron Semiconductor MT58L256L36F Verilog model is used in the testbench simulation for read/write command verification. Micron Technology memory modeling information can be viewed at: **http://www.micronsemi.com/models/index.htm**. The minimum clock frequency for the memory model was changed to 10 ns for the testbench simulation. The web version is for a 66 MHz device.

Flash Memory

An AMD29LV400B 4-Megabit flash memory, organized as 256K x 16 bits, is used in the reference design. The memory has a 70 ns access time. The flash requires 3.3 V I/O levels. For read operations, the Virtex-E device performs data assembly, prior to asserting transfer acknowledge signal to terminate the data bus tenure. Burst mode is not supported for the flash memory interface. Flash write cycles are not supported in the reference design. Flash memory device specifications are available on the web at: **http://www.amd.com/products/nvd/nvd.html**.

Memory Map

Assertion and deassertion of the system reset causes the processor to branch to location 0xFFF0_0100, which is memory mapped to flash memory. This is the first instruction fetched by the 750CX. The processor memory map is shown in Table 1. The physical location of the

exception vectors is the only system requirement that influences the memory map. All other memory mapped I/O elements can be defined by the system designer.
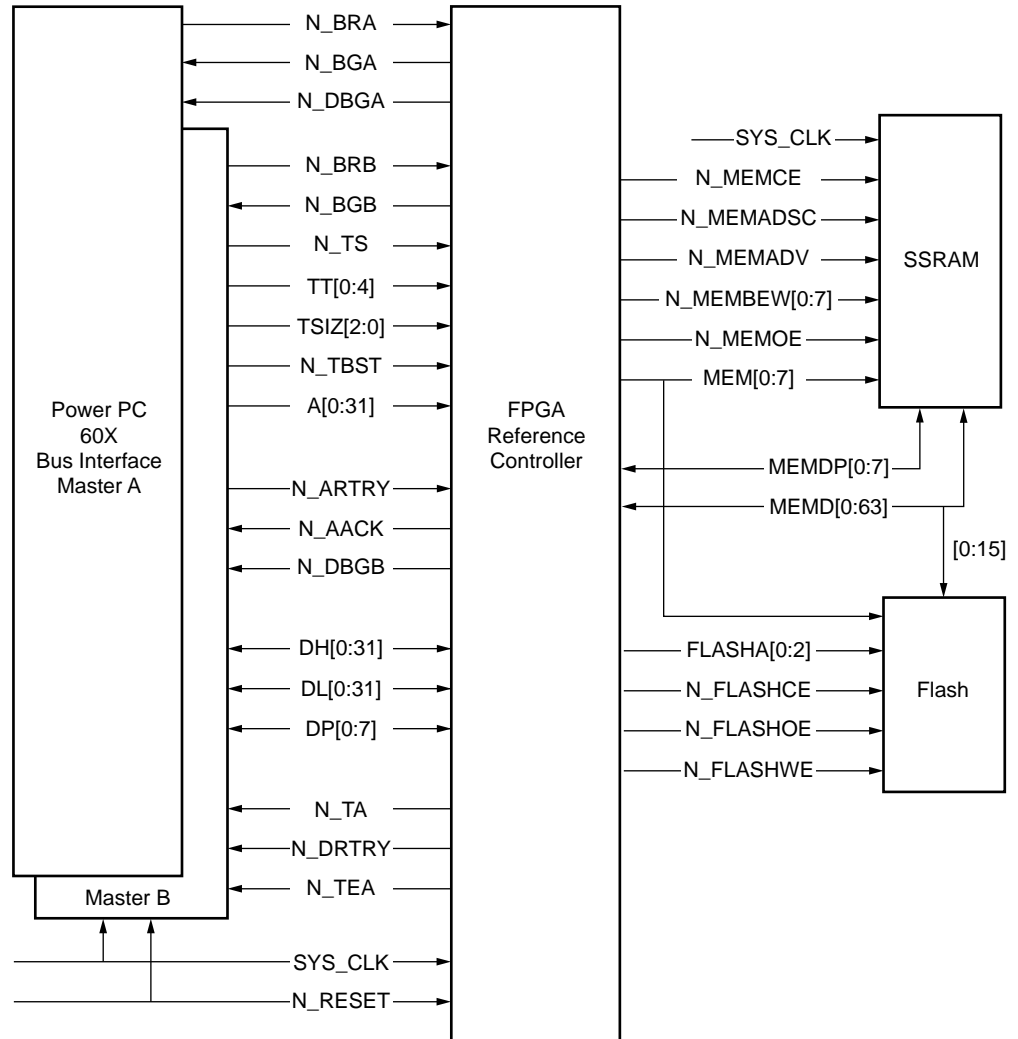
Table 1: **PowerPC Memory Map**

Physical Memory Range	I/O Device
FFF0_0000 to FFF7_FFFF	512 KB Flash Memory
4000_0000 to 400F_FFFF	1 MB SRAM Memory

Virtex-E Controller Design

Figure 3 is a block diagram of the Virtex-E I/O interface. The 60X bus interface is on the left of the diagram and the Flash and SRAM interfaces are on the right. Each bus master has a separate address bus request, address bus grant, and a data bus grant. All other 60X bus interface signals are tied together.

Both the address and data bus use the Big Endian bus and bit ordering convention. For a 32-bit bus, Bit 0 is the most significant bit and bit 31 is the least significant bit. DH contains the most significant word and DL contains the least significant word. Byte 0 is the most significant byte and byte 7 is the least significant byte. The flash memory is connected to the two most significant bytes of the memory bus. The design supports byte access to the SSRAM. Odd data parity is carried through the 60X bus to the SRAM memory.

Any unused I/O for the SSRAM and flash memory should be deasserted through pull-up or pull-down resistors on the printed circuit board or in the testbench. The data bus busy signal, N_DBB, is not used in the design.

Power PC
60X
Bus Interface
Master A

Master B

N_BRA
N_BGA
N_DBGA
N_BRB
N_BGB
N_TS
TT[0:4]
TSIZ[2:0]
N_TBST
A[0:31]
N_ARTRY
N_AACK
N_DBGB
DH[0:31]
DL[0:31]
DP[0:7]
N_TA
N_DRTRY
N_TEA
SYS_CLK
N_RESET

FPGA
Reference
Controller

SYS_CLK
N_MEMCE
N_MEMADSC
N_MEMADV
N_MEMBEW[0:7]
N_MEMOE
MEM[0:7]
MEMDP[0:7]
MEMD[0:63]

SSRAM

[0:15]

FLASHA[0:2]
N_FLASHCE
N_FLASHOE
N_FLASHWE

Flash

X246_03_121300

Figure 3: **Virtex-E 60X Bus Interface Controller I/O Block Diagram**

Virtex-E Internal Block Diagram

Figure 4 is the block diagram for the memory controller design. The design contains an address bus arbiter, a data bus arbiter, address pipeline module, a data bus module, a SRAM memory controller, and a flash memory controller. The PowerPC system bus interface I/O is on the left of the diagram and the memory I/O is on the right side.

The names of the blocks correspond to the Verilog and VHDL modules included in the reference design. The top module, TOP_60X, provides the interconnection of all the lower level modules. All modules receive the system clock and reset signals. The design and testbench are written in the Verilog HDL language. The design was synthesized using FPGA Express and implemented in an XCV400E-8-BG432 device. The following sections describe operation of each module in the design hierarchy.
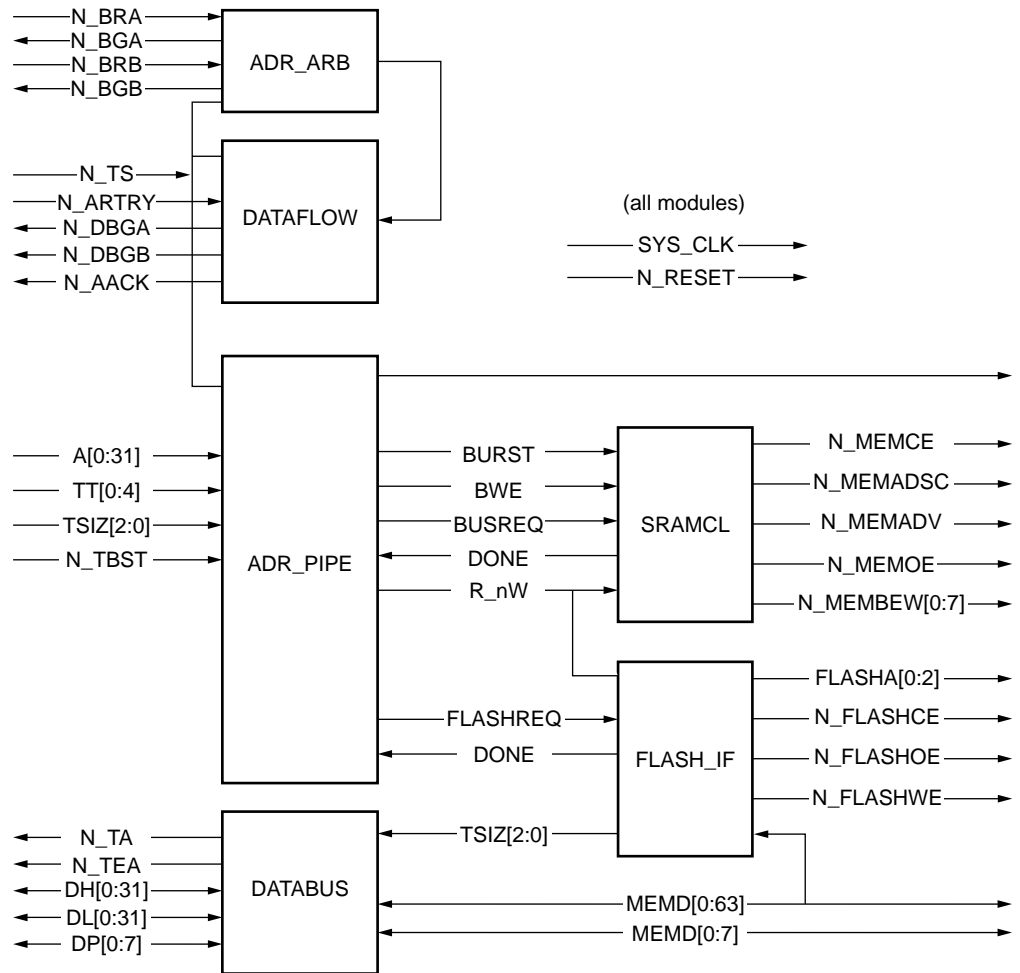
X246_04_121300

Figure 4: **Virtex-E Controller Block Diagram**

Address Bus Arbitration

Before a microprocessor can access memory, it must request mastership of the address bus. Address bus arbitration is controlled in the ADR_ARB module. The arbitration signals consist of an address bus request (N_BRA and N_BRB) and an address bus grant (N_BGA and N_BGB) for each processor. The arbiter generates an address master tag that travels down the address pipeline with the address bus information for subsequent data bus arbitration.
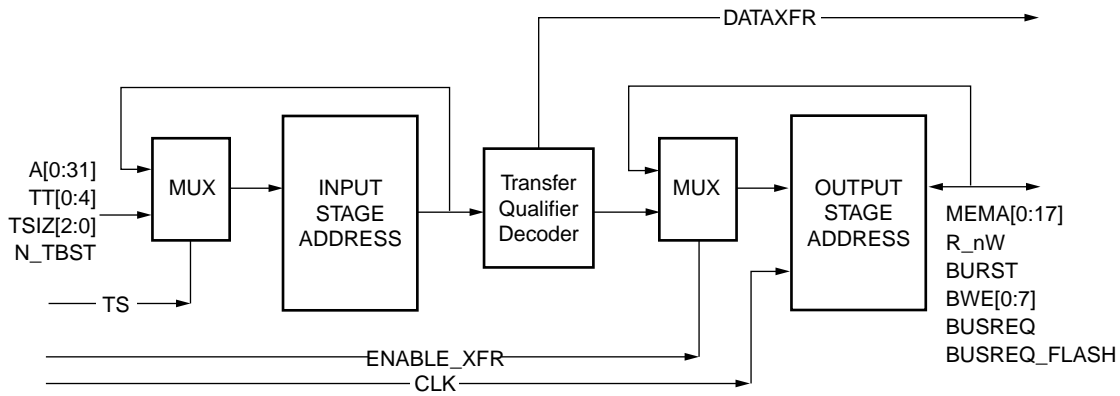
Address Pipeline

The address pipeline permits overlapping address bus transactions on the 60X bus, and the control logic is contained in the ADRPIPE module. Figure 5 is a block diagram of the address pipeline logic. The address and transfer qualifiers from the first bus tenure are stored internal to the Virtex-E device when the transfer start signal, N_TS, is asserted. The next address bus tenure cannot begin until the Virtex-E device asserts the address acknowledge signal, N_AACK. Once N_ACCK has been asserted, the next bus master can begin an address bus tenure. The address information is held on the bus until the output stage is ready to accept another data request. Decode logic is used to determine the following information from the transfer qualifiers:

• If the transaction is an address only transfer, the data tenure is not required, and no further

processing of this bus transaction is performed. The Address Acknowledge, N_AACK, is asserted to the bus master by a state machine in the DATAFLOW module.

• If another bus master asserts the address retry signal, N_ARTRY, the data bus request is cancelled. Supporting address retry induces a memory latency delay of one clock.

• If the address falls within a valid region for either flash or SRAM memory controlled by this device, and a data bus tenure is required, DATAXFR is signaled to the data bus arbiter.

X246_05_121400

Figure 5: **Address Pipeline Logic Block Diagram**

The transfer qualifiers, TT[0:4], N_TBST, and address bits, A[0:7], are decoded to determine if a data tenure is required for a device controlled by the Virtex-E device. The transfer type qualifiers and the associated PowerPC 60X bus commands are listed in Table 2. If the output queue is not busy, the DATAFLOW control logic enables the first stage address qualifiers into the output stage to be used for memory access. If the output stage is busy, the transfer qualifiers and address information are held in the input stage. The transfer qualifiers are used to generate the Read not Write signal (R_nW), BURST, and Byte Write Enables (BWE) for the FLASH_IF and SRAMCTL module. The decoder generates BUS_REQ and BUSREQ_FLASH signals for SRAM or Flash memory bus cycles, respectively.

The output stage stores enough address information to access the external memory devices. The reference design provides address outputs MEMA[0:17] to the SRAM. Internally, processor address bits A[9:10] are used for SRAM bank selection (one of four) and address bits A[11:28] are routed to MEMA[0:17].

The reference design supports misaligned data transfers. Misaligned data transfers occur when string operations are performed. Misaligned data transfers severely degrade system bus performance as the PowerPC bus interface unit must break the transfer into two separate bus cycles, with data being transferred on a word boundary. Consider a four-byte transfer that has a starting address A[29:31] = 3'b001. The first data transfer writes three bytes to byte lanes 1, 2, and 3. The second data transfer writes one byte to byte lane 4. The address decoder handles misaligned data transfers by using three least significant address bits, A[29:31], along with the transfer size attributes, TSIZ[0:2], to generate byte write enables to the SRAMCTL module. Table 3 shows the logic used to create the byte write enable signals passed to the SRAMCTL module.

Table 2: **Transfer Type Encodings for PowerPC 750 Bus Master**

PowerPC 750 Bus Master Transaction	TT[0:4]	60X Bus Specification Command
Address only (dcbst)	00000	Clean block
Address only (lwarx)	00001	Lwarx reservation set
Write with flush	00010	Cache inhibited or write-through store
Reserved	00011	FPGA treats as address only cycle
Address only (dcbf)	00100	Flush block
Reserved	00101	FPGA treats as address only cycle
Write Burst (non-global)	00110	Write with kill. Cast out or snoop copy-back
Reserved	00111	FPGA treats as address only cycle
Address only (sync)	01000	Sync
Address only (tlbsync)	01001	Tlbsync
Read	01010	Read: cache inhibited load or instruction fetch
Read no cache	01011	Read with no intent to cache
Address only (dcbz or dcbi)	01100	Kill block
Address only (lcbi)	01101	cbi
Read Burst	01110	Read with intent to modify. Load or instruction fetch
Reserved	01111	FPGA treats as address only cycle
Address only	10000	EIEIO (enforced in-order execution of I/O)
Reserved	10001	FPGA treats as address only cycle
Write single-beat (stwcx)	10010	Write with flush atomic
Reserved	10011	FPGA treats as address only cycle
Write single-beat (non GBL)	10100	External control word write
Reserved	10101	FPGA treats as address only
Reserved	10110	FPGA treats as address only cycle
Reserved	10111	FPGA treats as address only cycle
Address only	11000	TLB invalidate
Reserved	11001	FPGA treats as address only
Read single-beat (lwarx)	11010	Read atomic
Reserved	11011	FPGA treats as address only cycle
Read single-beat (non GBL)	11100	External control word read
Reserved	11101	FPGA treats as address only
Read Burst (lwarx)	11110	Read with intent to modify atomic
Reserved	11111	FPGA treats as address only cycle

The Verilog testbench uses all four write commands, all six read commands, and all nine address only commands. Reserved commands are treated as address only cycles. The processor bus functional model does not support reserved commands.

Table 3: **Generation of Byte Write Enables for SRAM Control**

Transfer size (bytes)	N_TBST	TSIZ[0:2]	A[29:31]	BWE[0:7]
32	0	010	000	8'hFF
8	1	000	000	8'hFF
1	1	001	000	8'h80
			001	8'h40
			010	8'h20
			011	8'h10
			100	8'h08
			101	8'h04
			110	8'h02
			111	8'h01
2	1	010	000	8'hC0
			001	8'h60
			010	8'h30
			100	8'h0C
			101	8'h06
			110	8'h03
3	1	011	000	8'hE0
			001	8'h70
			100	8'h0E
			101	8'h07
4	1	100	000	8'hF0
			100	8'h0F

Data Bus Arbitration

Data bus arbitration is contained in the DATAFLOW module. Overlapping data transactions are not supported on the 60X bus. The arbiter assumes responsibility for scheduling the data bus tenures by keeping track of the current bus tenure (single or burst) and only asserting the next data bus grant when the current tenure has completed. The data bus grant is asserted to the next bus master on the clock cycle before the master can begin its data tenure. The data bus busy signal, N_DBB, is not used by the arbiter. The 750CX package does not have a data bus busy pin.

The transfer start signal, N_TS, is an implied data request from the PowerPC. The address decoder asserts DATAXFR, if the transfer qualifier information indicates a data tenure is required. Address bus mastership information is passed to the data bus arbiter when transfer start is asserted by the bus master. This guarantees that the correct data bus grant is issued when the data bus is available. If the address retry signal, N_ARTRY, is asserted, the arbiter discards the request and waits for the next address bus tenure to commence. If address retry is not asserted, the data bus arbiter asserts the ENABLE_XFR signal, and the bus request is

moved to the output stage of the address pipeline queue for processing by either the SRAM or Flash memory controllers.

The design supports bus snooping using the address retry input, N_ARTRY. A single clock memory latency penalty is induced in order to wait for the address retry check following assertion of address acknowledge. If address retry is not asserted, the incoming address bus qualifiers are passed to the output stage to begin memory access. Once the address information has moved to the output stage, the input stage is ready to accommodate the next address tenure on the 60X bus.

Data Bus

The DATABUS module provides an interface between the 60X bus and the memory bus. The SRAM or the flash memory data information is presented to the processor for read operations. The module provides 3-state I/O on all processor and memory data bus pins. In order to support higher bus clock rates, all data inputs are registered at the rising edge of the SYS_CLK input to the Virtex-E device. This applies to both the PowerPC data bus as well as the SRAM memory data bus. This induces a memory latency delay of one clock period. As bus speeds increase, synchronous design practices are required in order to ensure data integrity in the system.

Bus keeping circuits are provided in the Virtex-E family to eliminate external pull-up resistors to maintain the data bus at valid logic levels during periods where the bus is 3-stated. Routing the data bus I/O through the Virtex-E device permits the I/O standards on the processor side and the memory side of the interface to change independently.

Flash Memory Control

The FLASH_IF module controls the external flash memory, which contains the processor boot code. All instructions are fetched in single-beat, 64-bit cycles. The flash memory controller assembles the doubleword (8 bytes) using multiple flash read cycles. Burst reads from flash are not supported. The access time for flash reads is 70 ns.

A counter provides the two least significant address bits to the flash. When the counter gets to a count of three, the state machine terminates the multiple read sequence. The counter bits are used to steer the 16-bit read data to a 64-bit holding register. The four half-words are assembled into one double word. Once the four memory read cycles have finished, the transfer acknowledge signal is asserted to terminate the data bus tenure. Odd parity is generated on each byte lane prior to assertion of N_TA. Address bit A[29] is used for flash address[2] and the counter provides flash address bits [0:1].

SRAM Control

The SRAMCTL module performs either a single or a four beat access to the external flow-through SSRAM memory. A bus cycle starts when the BUS_REQ input is asserted. A four beat access is performed if BURST = 1'b1.The design supports a 64-bit bus plus 8 parity bits. The byte enable signals (BWE) are only used for write cycles when the R_nW signal is 1'b0.

Cache fill operations are filled in four beats of 64-bits, each with the critical double word loaded first. This means that if the processor needs a data word at a location that is not on a 64 byte boundary, the memory access will need to wrap around to fill the entire cache line. Synchronous SRAM permits address wraparound using the internal counter in the SSRAM.

As shown in Table 4, using processor address bits A[27:28] as the two LSB inputs of the SSRAM permits critical double-word cache fills to operate properly.

Notes:
1.   A[29:31] bits are always 3b'000 during burst transfers.

Table 4: **Critical Cache Fill Double Word Ordering**

Data Transfer	A[27:28] = 00	A[27:28] = 01	A[27:28] = 10	A[27:28] = 11
First beat	DW0	DW1	DW2	DW3
Second beat	DW1	DW2	DW3	DW0
Third beat	DW2	DW3	DW0	DW1
Fourth beat	DW3	DW0	DW1	DW2

In order to use the internal address counter, the advance input to the SSRAM is used. When advance is deasserted, the external address is loaded into the memory. When advance is asserted, the internal counter increments by one.

Design Verification

A Synopsys MPC740_FX bus functional model (BFM) is used to create the 60X bus stimuli for design verification. The Mentor Graphics Modelsim_SE product is required in order to use the Synopsys FlexModel in a Verilog simulation environment. The BFM contains a MPC740_FX and a MPC750_FX model that are connected together in the testbench. The MPC740_FX model permits the user to enter a sequence of read and write commands that cause bus activity on the external bus interface. The MPC750_FX provides the L2 cache interface support. The testbench developed for this reference design is concerned only with verification of the bus interface design.

The BFM supports both cycle accurate and timing mode simulations that are enabled through defparam statements in the testbench. The benefit to using a bus functional processor model is that the processor bus interface is modeled rather than a user's concept of the behavior of the processor. Once the model is installed and running in the simulation environment, the designer is extremely productive at generating test cases. The model provides cycle accurate and full timing mode support. The MPC740_FX FlexModel supports the following functionality of the PowerPC 750 microprocessor:

- Address arbitration and data arbitration
- All read and write cycles. Pipelined, non-pipelined, single and burst transfers
- All address-only cycles
- Address and data parity generation and checking
- Exception service routines
- Bus snooping and snoop response
- Address retry terminated cycles

For information on Synopsys logic modeling products, go to: **http:\\www.synopsys.com** and look in the models directory. The *FlexModel User Guide* and the *Mpc740_fx FlexModel Datasheet* are important reference documents. Some model data sheet notes pertinent to this reference design usage of the model are:

1. Normally, the model drives a High-Z value on unused bits of the data bus while performing a write of a size less than 8 bytes, and the parity bits are calculated accordingly.
2. The MPC740_fx model can be made to drive valid data on unused data bits by setting the MPC740_DRIVE_REG register. See mpc740_set_reg command in the MPC740_FX FlexModel datasheet for details. (Pull-up resistors were added on the processor and memory data buses in the testbench).

The Verilog testbench validates the reference design using all four write commands, all six read commands, all nine address only transfer types, and performs byte, halfword, three byte, word, doubleword, and quad doubleword transfers. The simulation was performed using minimum, typical, and maximum timing modes for the IBM25PPC740LGB_500 device. Different processor timings can be selected by changing a parameter in the testbench. The following statements are required in the testbench to turn on the timing mode support for the BFM.

```
defparam u1.FlexTimingMode  = `FLEX_TIMING_MODE_ON;
defparam u1.TimingVersion   = "IBM25PPC740LGB_500";
defparam u1.DelayRange      = "TYP";
```

TOP_60X.V is at the top of the hierarchical design and the testbench file is named
MPC750_TST.

Bus Functional Model Simulation Results and Bus Performance

The testbench begins with five write/read cycles to the SSRAM memory to verify all four write
commands and all six read commands. The MPC740_FX command language includes a read
result command that permits the testbench to compare an expected result to a returned result.
A status message is posted on the simulator main window at the completion of each read
command. The message includes a pass/fail status and the read result returned from the read
command.

The testbench includes a 32-byte burst read and write access to SSRAM. Single-beat writes for
1, 2, 3, 4, and 8 byte writes to memory. All nine address only commands are included as well.

The testbench bus cycles demonstrate the overlapped address bus transactions. A mix of
single and burst, load and store, data transfers demonstrate the data bandwidth capabilities of
the reference design. Maximum bus bandwidth is obtained by using back-to-back burst
transfers; however, this does not always reflect the way the processor behaves. Burst transfers
are usually performed when the processor loads or stores cache information, or invalidates the
address translation look-aside buffers. Unlike direct memory access (DMA) controllers, the
RISC processor is not efficient at moving large amounts of data.

Pull-up resistors are included in the behavioral simulation to permit both the processor and
memory data buses to be a known state when the model is not driving the bus. For example,
when the processor does a byte access, the processor drives a "z" level on the other byte lanes.
Keeper circuits in the Virtex-E devices enable these resistors to be removed from the design.

Achieving Speed

The following techniques were used to achieve the maximum system clock speed in the design.

Delay-Locked Loops (DLL)

A DLL is used to eliminate clock delay amongst registers communicating with I/O. This gives
the user more of the clock period to utilize because there is less than a 25 ps clock skew.

For more detailed information, refer to **XAPP 132**, *Using the Virtex Delay-Locked Loop*.

Registering I/Os

Registered I/Os are registers placed on the edge of the die as close to the I/O pad as possible.
These registers are inside the IOB (Input/Output Block). By using I/O registers, one guarantees
the shortest path between an I/O and a register.

To use the registered I/O with a device based on the Virtex-E architecture, all of the flip-flops
must use the same clock and reset signals. One cannot have two I/O registers with different
clocks. Also, no logic is permitted between the flip flop and the I/O pad, because there is no
logic in the IOB and all logic must be implemented via a Configuration Logic Block (CLB)
outside of the IOB. To pull the registers into the IOBs, use the map option:

$$-\texttt{pr [i | o | b]}$$

where `i` = input, `o` = output, and `b` = both.

For more information regarding map options, refer to *Development System Reference Guide* at
http://toolbox.xilinx.com/docsan/3_1i.

SelectI/O™ Resource

The SelectI/O resource allows one to specify the use of different I/O standards with Virtex-based families. The PowerPC bus interface used LVCMOS2 and LVTTL was used in the memory interface of the reference design.

For more information on using the SelectI/O resource refer to **XAPP133**, *Using the Virtex SelectI/O Resource*.

Constraints

In the reference design files, TOP60X.ucf has the I/O bus standard, keeper, pullups, clock timing, and slew rate constraints.

References

AMD

AMD29LV400B 4Megabit (256K x 16-bit) CMOS 3.0 V-only Boot Sector Flash Memory

IBM

PowerPC 750 RISC Microprocessor User's Manual, PowerPC 750CX Supplement to the PowerPC 750 RISC Microprocessor User Manual

Micron

8Mb 256K x 32/36 Flow-Through SYNCBURST SRAM Data Sheet

Synopsys

MPC740_FX FlexModel Datasheet

Xilinx

XAPP132: Using the Virtex Delay-Locked Loop

XAPP133: Using the Virtex SelectI/O Resource

Reference Design

Table 5 lists the files that are included for review of the reference design and testbench. The reference design can be downloaded from the Xilinx web site: **xapp246.zip**.

Table 5: **Reference Design and Testbench File Attachments**

Attachment File Description	File Name(s)
Verilog and VHDL Design Files	TOP_60X ADR_ARB ADRPIPE DATAFLOW DATABUS SRAMCTL FLASH_IF
Verilog Testbench Files	ST_MPC750_TST.V PULLUP.V PULLUPS.V MT58L256L36F.V
Modelsim Waveforms depicting different write scenarios of burst, 8 byte, 1, 2, 3, and 4 byte writes	ST_MPC750_4W4R_wave ST_MPC750_8BW_4x_wave ST_MPC750_1BW_8x_wave ST_MPC750_2BW_4x_wave ST_MPC750_4BW_4x_wave

Conclusion

Over the course of a products lifetime, evolutionary changes occur that impact the system design. Using a Virtex-E device to design the microprocessor bus interface helps mitigate the schedule impact these changes have on time-to-market through design reuse. The 750CX is a reduced pinout package that eliminates several of the optional 60x bus protocol pins. Table 6 highlights changes to the 60X bus interface (as the microprocessor evolves from a 750 to a 750CX) that can be handled by modifying the Virtex-E design.

- Change the Virtex-E I/O voltage to accommodate the lower I/O voltage requirement for the 750CX.
- Modify the bus interface design to generate data parity on writes and check data parity on reads.
- Remove weak pull-up resistors (in a 750 design), on the bidirectional I/O included in the Virtex-E interface, by recompiling the Virtex-E design for a 750CX design.
- Modify the Virtex-E bus protocol design by removing support for address bus busy, data bus busy, data bus disable, and data retry.

Table 6: **Evolutionary 60X Bus Interface Changes**

60X Bus Interface Characteristic	PowerPC 750	PowerPC 750CX
I/O voltage	1.8 V, 2.5 V, or 3.3 V	1.8 V or 2.5 V
Address & Data Parity	4 bit and 8 bit	Removed from 60X protocol
Weak output drivers	No (bus pull-ups required)	Yes (no pull-ups)
Data Retry Mode	Supported	Removed from 60X protocol

The address bus, transfer attributes, and data bus can float in a high impedance state during periods of inactivity. This has the potential to cause excessive power to flow in the bus receivers, both on the processor and on other devices attached to these signals. To ensure that these signals do not float, pull-up resistors are usually added on the system board. An alternative approach is to use output drivers that can weakly drive these bidirectional signals during periods of inactivity. The Virtex-E output drivers have this capability. For 750CX designs, the processor has this capability as well. In this case, the new Virtex-E design could remove the weak output drivers and let the processor provide this function.

As design complexities grow, system level simulations are required to help debug the complex operation of the system and eliminate expensive design spins for problems discovered in system integration. System solutions using a Virtex-E FPGAs have several advantages over off-the-shelf controller ICs. FPGAs permit insight into the "guts" of the design using system level simulation. When models are unavailable, hardware modelers must be used to simulate off-the-shelf integrated circuits. Unfortunately, the modeler only permits insight into how the I/O is behaving. With time-to-market pressures ever increasing and design complexity growing, the system designer need to have every possible means to quickly resolve hardware/software integration issues.

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
12/15/00	1.0	Initial Xilinx release.