

Understanding the MPC7450 Family L3 Cache Hardware Interface

by *Michael Everman*
CPD Applications
Freescale Semiconductor, Inc.
Austin, TX

This document covers many issues that arise when the backside cache is designed for members of the MPC7450 family of microprocessors. Because many other resources discussing general hardware design are available, this application note focuses on features or areas that are specific or new to the MPC7450 family. Topics of interest include clocking, layout, and configuration of the hardware interface.

While this document refers to the MPC7450 throughout, it equally applies to any MPC7450 family microprocessor that features an L3 backside cache interface. In cases where device-specific differences exist, these are mentioned explicitly.

1 Introduction

A well-designed backside cache interface is an important component in obtaining the best performance in a system using one of the microprocessors in the MPC7450 family featuring an L3 interface, such as the MPC7451 and MPC7455. The purpose of this document is to illuminate issues of concern to the designer and address common questions about the MPC7450 backside cache. Additionally, the L3 cache interface implements new features designed to

Contents

1	Introduction	1
1.1	General Design Guidelines	2
1.2	Signals of the L3 Cache Interface	2
2	Configuring the L3 Cache Interface in Software	5
2.1	L3 Cache Control Register (L3CR)	5
2.2	L2 Cache Control Register (L2CR)	9
2.3	Memory Subsystem Control Register (MSSCR0)	10
2.4	L3 Output Hold Control Register (L3OHCN)—MPC7457-Specific	10
2.5	L3 Input Timing Control Registers (L3ITCRn)	11
3	Clocks and Timing	11
3.1	Differential Clocks	12
3.2	Clocking and Address and Control Signals (all SRAM types)	12
3.3	Data Signals and Writes (all SRAM types)	12
3.4	Data Signals and Read Accesses	14
4	Adjusting AC Timing Margins	18
4.1	Adjusting AC Timing Margins Using Software	18
4.2	Adjusting AC Timing Margins Using Hardware	23
4.3	Adjusting AC Timing Margins for MSUG2 DDR SRAM	24
5	Special Considerations for the L3 Address Bus	26
5.1	Burst Accesses and MSUG2 DDR	26
5.2	Address Bus Bit Ordering and Alternative Uses of the Backside Cache Interface	27
6	References and Revision History	28

provide a great deal of flexibility for debugging and for support of future SRAM technologies. This application note also describes many of these new features in detail.

1.1 General Design Guidelines

The guidelines for designing a backside cache interface are as follows:

- Place the SRAM physically as close as possible to the processor
- Make all traces as short as possible
- Match all trace delays as closely as possible
- Use ‘Y’ or ‘T’ topologies with equal stub lengths for address and control signals; daisy-chains are not generally recommended
- Do not use signal terminators unless careful simulation shows them to be necessary

Because of the clocking schemes for the backside cache interface, propagation delays do not directly affect an AC timing analysis of any signal that is delay matched with its associated clocks. All AC timing parameters described in the hardware specifications assume the specified signal is delay matched with the appropriate clock signal; that is, an L3_CLK signal for data (during write accesses), address, and control signals, or an L3_ECHO_CLK signal for data signals during read accesses. Differences in the delays between a signal and the associated clock signal must be accounted for in the AC timing analysis and the guidelines are intended to minimize this. These clock groups are described in the hardware specifications, and [Section 3, “Clocks and Timing”](#) discusses them in detail.

Signal integrity is a primary concern when designing a fast interface. Minimizing transmission line effects and trace loading by the placing of the SRAM in close proximity to the processor and using the shortest possible traces helps to ensure a favorable environment. If the guidelines presented here are followed, signal terminators should not be necessary and are not generally recommended unless detailed signal integrity analysis and simulation show them to be necessary. Finally, electromagnetic interference (EMI) may be also be of concern given the fast clock rates of the L3 bus. Limiting the trace lengths helps to minimize emissions.

1.2 Signals of the L3 Cache Interface

This section describes the uses of the L3 interface signals for each type of SRAM technology. Except for the echo clock signals, the signal use for MSUG2 DDR is essentially identical to pipelined burst and late write SRAM. Note that the L3_ADDR bus is little-endian, whereas all others are big-endian. This is because the PowerPC™ architecture is a big-endian architecture while most SRAM vendors define the address signals on their devices as little-endian. For convenience, the L3_ADDR bus is defined in a little-endian format so that L3_ADDR[0] is connected to SA[0] on the SRAM, and so on.

1.2.1 Signals in Pipelined Burst and Late Write SRAM Modes

[Figure 1](#) shows the typical connectivity for the L3 interface, and [Table 1](#) lists the signals for the MPC7450 L3 cache interface and their functions when the interface is configured in pipelined burst and late write modes.

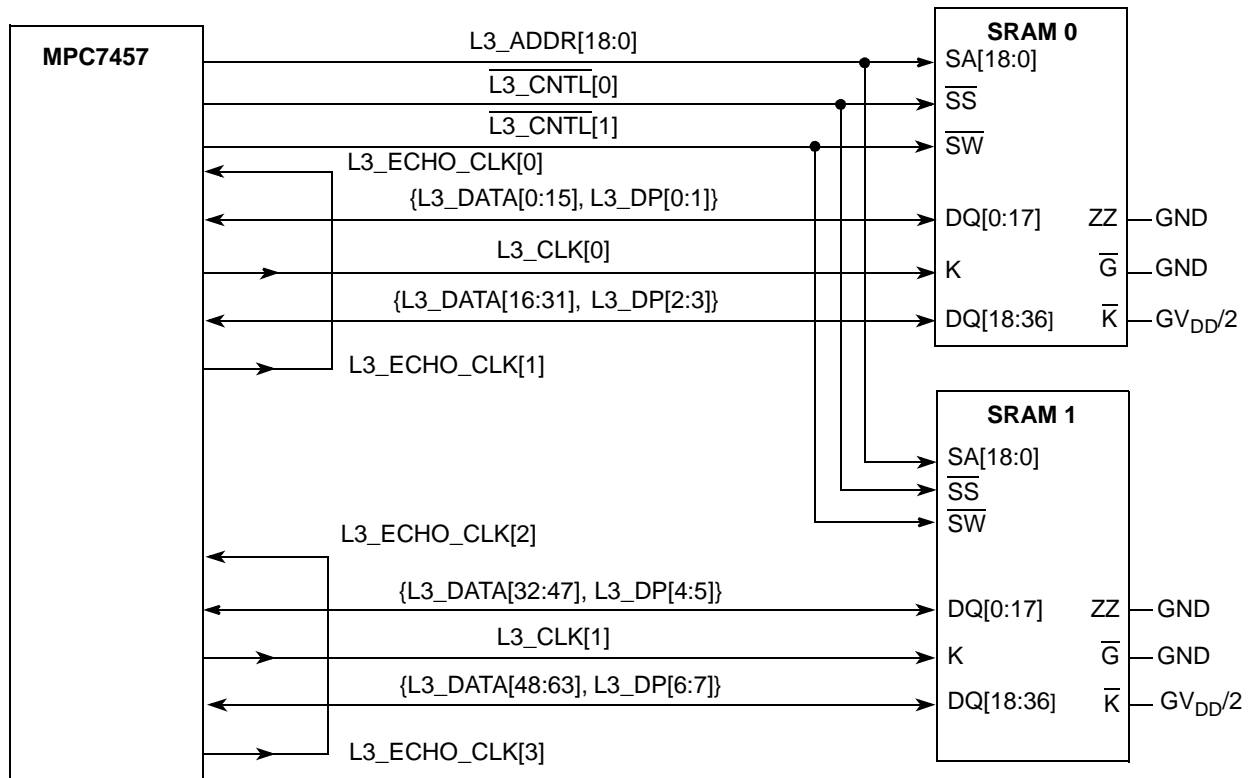


Figure 1. Typical Signal Connections for 4M L3 Cache PB2 and Late Write SRAM

Table 1. L3 Interface Signal Functions for Pipelined Burst and Late Write SRAM

Signals	Description	Comment/Alternate Name
L3CLK[0:1]	SRAM clocks	Each clock drives clock input of one SRAM
L3_ECHO_CLK[1,3]	Synchronization loop clock outputs	L3_SYNC_OUT[0,1]
L3_ECHO_CLK[0,2]	Synchronization loop clock inputs	L3_SYNC_IN[0,1]
$\overline{\text{L3_CTRL0}}$	Chip enable	$\overline{\text{L3CE}}$
$\overline{\text{L3_CTRL1}}$	Write enable	$\overline{\text{L3WE}}$
L3_ADDR[18:0] ¹	Address bus	MSB: L3_ADDR[18] ² , LSB: L3_ADDR[0]
L3_DATA[0:63]	Data bus	MSB: L3_DATA[0], LSB: L3_DATA[63]
L3_DP[0:7]	Data parity (1 bit per byte lane)	Used for address and data parity if address parity enabled

¹ MPC7457 only; MPC7450, MPC7451, and MPC7455 implement L3_ADDR[17:0], supporting up to 2 Mbytes of SRAM. Note that the MPC7457 supports 2M of L3 cache; the remainder must be private memory.

² MPC7457 only; L3_ADDR[17] is the MSB for MPC7450, MPC7451, and MPC7455.

1.2.2 Signals in MSUG2 DDR Mode

Figure 2 shows the typical connectivity for the L3 interface, and Table 2 lists the signals for the MPC7450 L3 cache interface and their functions when the interface is configured in MSUG2 DDR mode.

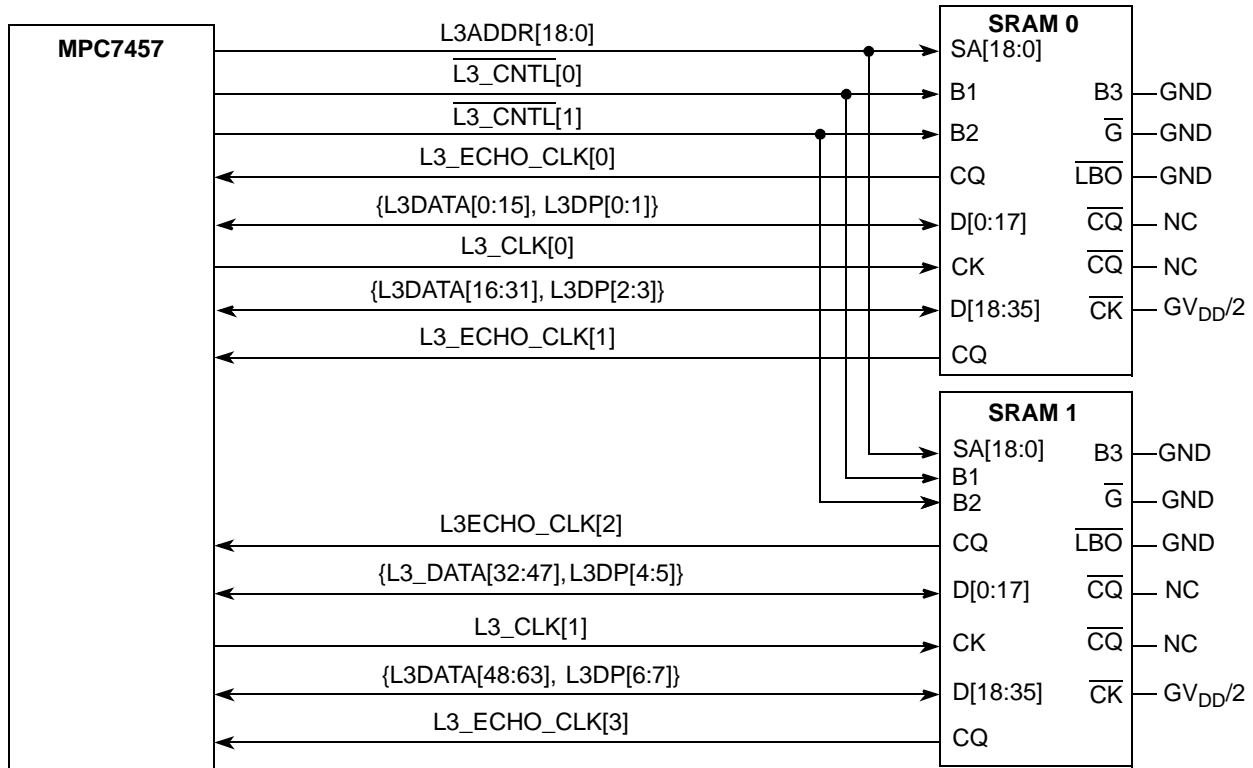


Figure 2. Typical Signal Connections for L3 Interface with 4M DDR SRAM

Table 2. L3 Interface Signal Functions for Pipelined Burst and Late Write SRAM

Signals	Description	Comment/Alternate Name
L3CLK[0:1]	SRAM clocks	Each clock drives clock input of one SRAM
L3_ECHO_CLK[0:4]	Echo clock inputs	Inputs for clock signals driven by SRAM for read data
$\overline{\text{L3_CTRL0}}$	Chip enable	$\overline{\text{L3CE}}$
$\overline{\text{L3_CTRL1}}$	Write enable	$\overline{\text{L3WE}}$
L3_ADDR[18:0] ¹	Address bus	MSB: L3_ADDR[18] ² , LSB: L3_ADDR[0]
L3_DATA[0:63]	Data bus	MSB: L3_DATA[0], LSB: L3_DATA[63]
L3_DP[0:7]	Data parity (1 bit per byte lane)	Used for address and data parity if address parity enabled

¹ MPC7457 only; MPC7450, MPC7451, and MPC7455 implement L3_ADDR[17:0], supporting up to 2 Mbytes of SRAM. Note that the MPC7457 supports 2M of L3 cache; the remainder must be private memory.

² MPC7457 only; L3_ADDR[17] is the MSB for MPC7450, MPC7451, and MPC7455.

2 Configuring the L3 Cache Interface in Software

Two registers, the L3 cache control register (L3CR) and memory subsystem control register (MSSCR0), are used to configure the L3 cache hardware interface. The L2 cache control register (L2CR) also has one bit (L2CR[L3OH0]) that affects the L3 cache for the MPC7455 (only). Each of these registers is discussed in the following sections. The MPC7457 has an additional register, the L3 output hold control register (L3OHCR), which is not on other MPC7450 family devices. Another register, the L3 private memory address register (L3PM), is used to set the base address for private memory space when the L3 cache is operating in private memory mode, but this register is not of concern from a hardware design standpoint and is not discussed in this application note.

2.1 L3 Cache Control Register (L3CR)

The L3CR is the primary configuration register for the L3 cache. This section discusses the fields that affect the hardware aspect of the design. Fields not discussed are associated with other aspects of the design and are detailed in the *MPC7450 RISC Microprocessor Family User's Manual* and in other application notes.

2.1.1 L3 Clock Ratio (L3CLK)

The L3 clock is derived from the processor core clock. L3CLK determines the core:L3 clock ratio. Because the L3 clock frequency cannot be less than the system clock frequency, the minimum L3 clock frequency in any system is the system bus frequency. Though the various device hardware specifications specify a maximum or typical L3 clock frequency, the specifications also explain that the actual maximum frequency is a function of the AC timing of the processor, of the SRAM, circuit loading, and board characteristics such as layout, signal integrity, and so forth. Stable operation of the L3 at clock frequencies higher than the value specified in the hardware specifications is possible in a well-designed, tightly-toleranced system, but the specified value is considered a realistic, approximate limit in a typical system. Likewise, not all designs may be able to achieve the stated maximum frequency.

The MPC7457-specific L3CLK[L3CLKEXT] bit is an extension of the L3CLK field, allowing additional clock divider ratios. These additional ratios are useful when the combination of a high core clock frequency and slow L3 clock frequency is desired, which may be useful during system debug. Some dividers not previously provided, such as 4.5 and 5.5, now allow finer control of the L3 clock frequency. Because the minimum L3 clock frequency is the system bus frequency (f_{SYSCLK}), the higher dividers must be used with caution as it is possible to violate this rule with these dividers.

2.1.2 L3 Sample Point Configuration (L3CKSP, L3PSP, and L3SPO)

Three fields, L3CKSP, L3PSP, and L3SPO, configure the sample point settings for the MPC7450 L3 cache. These settings determine when the processor samples the FIFO to which all data read from the SRAM is forwarded. These settings are configured entirely in software and are mentioned here only because propagation and loading delays on the board affect them, so the delays must be known before the sample points can be configured. For details on configuring the sample point settings, see *Setting the Sample Points on the MPC7450 L3 Backside Cache* (AN2182). Note that the sample points must be

configured regardless of which type of SRAM is used and whether the L3 is used in cache mode or private memory mode.

The MPC7457-specific L3 Clock Sample Point Extension (L3CKSPEXT) field is an extension of L3CKSP and allows higher sample point settings to be selected. It is unlikely these will be needed in a typical system during normal operation, but they may be useful in debugging L3 interface problems.

2.1.3 L3 Non-Integer Ratio Clock Adjustment (L3NIRCA)

The L3NIRCA field is used to adjust the phase of the L3_CLK_x with respect to the address, control, and data signals for a non-integer clock ratio (that is, 2.5:1, 3.5:1). To understand the purpose of this bit, you must understand how the L3 clock signals are generated. The address, control, and data (for writes) are driven based on an internal L3 clock (*internal_L3clk*). To provide adequate setup and hold times at the SRAM inputs, the MPC7450 attempts to drive the corresponding L3_CLK edge three-quarters of an L3 clock period later. The offset is generated using the VCO clock (*VCO_clk*), which runs at twice the processor core clock (*core_clk*) frequency. You can make this delay exactly three-quarters of an L3 clock period in any integer ratio mode because the ratio of *VCO_clk* to L3_CLK is always a multiple of two and L3_CLK_x can be driven on a rising or falling *VCO_clk* edge, as shown in [Figure 3](#); this example shows a data write and 4:1 core:L3 ratio.

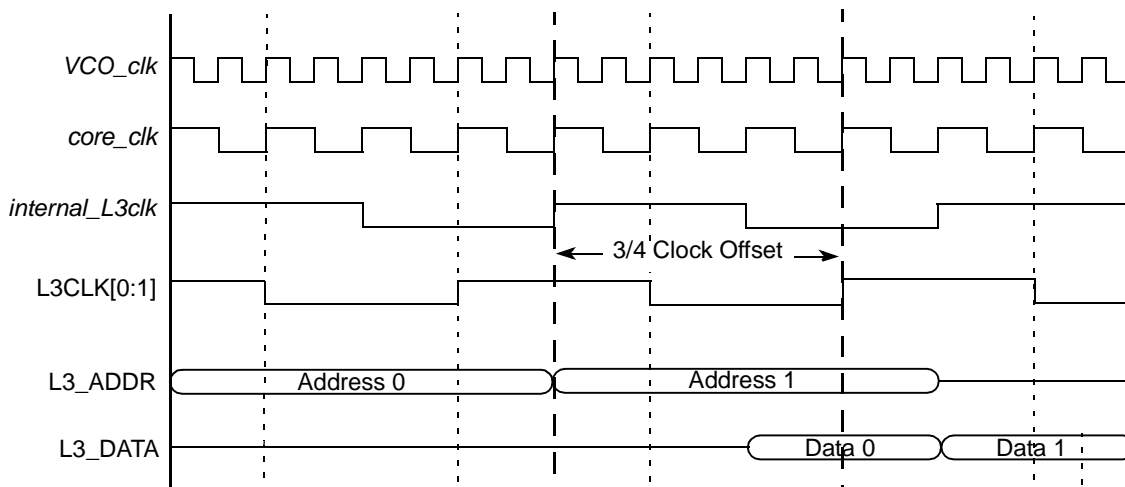


Figure 3. Offset of L3_CLK Signals from Internal L3 Clock with 4:1 Core:L3 Ratio

In non-integer ratios, however, there is not an even number of core clocks in an L3 clock period and thus the *VCO_clk*-to-L3_CLK ratio is not a multiple of two. Therefore, there is no rising or falling *VCO_clk* edge at a point in time exactly three-quarters of an L3 clock period after the *internal_L3clk* edge. Instead, there is a *VCO_clk* edge one-quarter of a VCO period before and after the ideal point. As a result, you must decide which *VCO_clk* edge to use to drive L3CLK[0:1]. In the default state (L3NIRCA = 0), the *VCO_clk* edge just after the ideal point is used, as shown in the 3.5:1 example in [Figure 4](#). If L3NIRCA = 1, the *VCO_clk* edge just before the ideal point is used, as shown in [Figure 5](#).

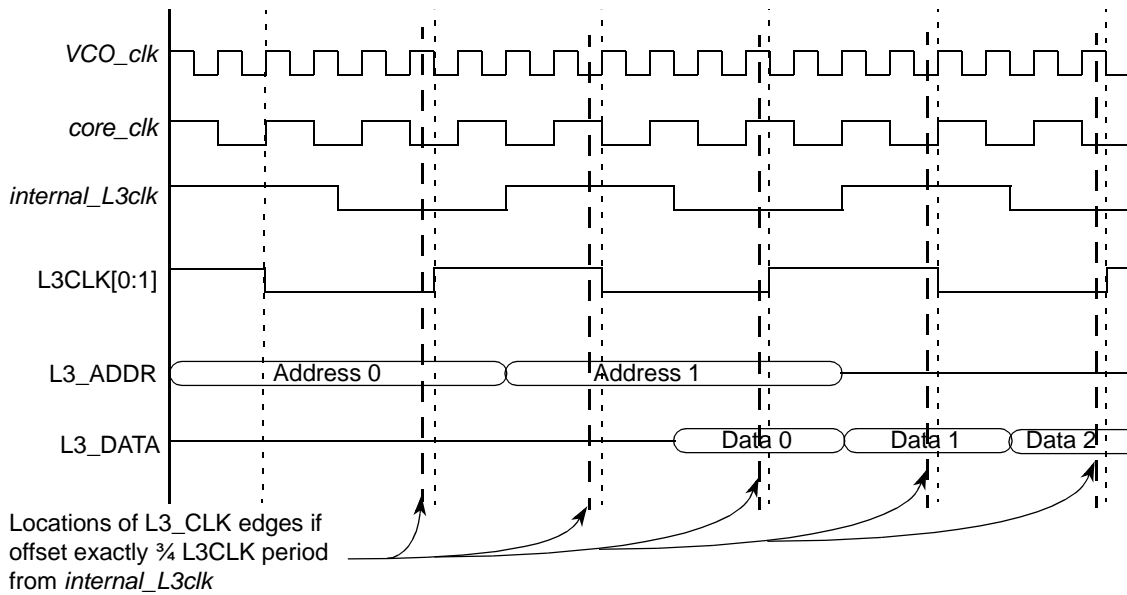


Figure 4. Location of L3_CLK Edges with 3.5:1 Core:L3 Ratio and L3CR[L3NIRCA] = 0 (Default)

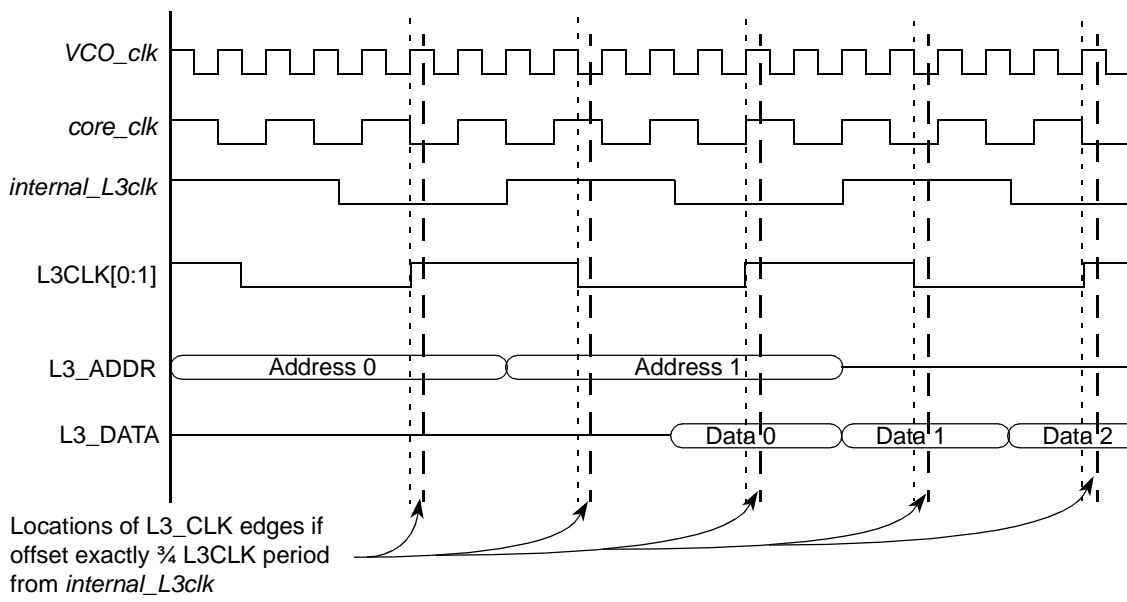


Figure 5. Location of L3_CLK Edges with 3.5:1 Core:L3 Ratio and L3CR[L3NIRCA] = 1

These slight deviations from the ideal case are not comprehended in the L3 AC timing parameters in the hardware specifications. However, because the L3 AC timing specifications are guard banded, devices operating in non-integer ratio clock modes should meet all specifications. Additionally, because the amount of delay incurred is inversely proportional to the core frequency, higher core frequencies further mitigate any effects from it in a typical application. However, if you complete a detailed AC timing analysis and are still concerned that there may be very small margins for the SRAM's input hold, L3NIRCA can be set in order to provide an additional VCO phase (that is, one-quarter core clock) of input hold margin at the SRAM. Because this shifts the clocks, the SRAM input setup margin is reduced by the same amount and this bit should not be set if input setup time margin at the SRAM is very small.

2.1.4 L3 Output Valid Time Adjust (L3OH1)—MPC7455-Specific

For the MPC7455 (Rev. 2.1 and later), L3OH1 (L3CR[12]) can be used in conjunction with L3OH0 (L2CR[12]) to provide additional output hold time from the processor. Only the MPC7455 implements these bits and exact implementations vary according to device revision, as shown in Table 3. For MPC7455 Rev. 2.1, these bits cause the L3CLK[0:1] signals to drive clock edges earlier relative to the address, control, and data signals, similar in effect to the L2CR[L2OH] bits on some earlier processors such as the MPC755 and MPC7410. This increases the amount of input hold time provided to the SRAM. However, it also causes the amount of input setup margin at the SRAM to reduce by a like amount. For MPC7455 Rev. 3.x and later, non-zero values cause the L3 clocks to be delayed relative to the address, control, and data signals, resulting in faster output valid time but less output hold time. It is important to note the distinction. For specific information on the effect of these bits on the L3 AC timing, see the appropriate hardware specifications for a particular device.

Table 3. L3OH0 and L3OH1 Implementations by Device

L3OH0 (L2CR[12])	L3OH1 (L3CR[12])	Device				
		MPC7455 Rev. 2.0 and Prior	MPC7455 Rev. 2.1		MPC7455 Rev. 3.x and Later	
0	0	Not supported	Least output hold time for all address, control and data signals (Default)	Least (best) output valid time	Most output hold time for all address, control and data signals (Default)	Most (worst) output valid time
0	1		Most output hold time for all address, control and data signals latched by L3_CLK1. (Not recommended)		Less output hold time for all address, control and data signals	
1	0		Most output hold time for all address, control and data signals latched by L3_CLK0. (Not recommended)		Even less output hold time for all address, control and data signals	
1	1		Most output hold time for all address, control and data signals	Most (worst) output valid time	Least output hold time for all address, control and data signals	Least (best) output valid time

For MPC7455 Rev. 2.1, each L3OH bit separately controls one L3 clock signal. Therefore, both bits should be cleared or both set for this revision of the MPC7455. Setting one bit but not the other causes only one of the clocks to be driven earlier, effectively increasing the clock-to-clock skew between the L3 clock signals (defined as $t_{L3CSKW1}$ in the *MPC7455 RISC Microprocessor Hardware Specifications*). For example, if the L3OH[0-1] = 0b01 for the L3 interface shown in Figure 6, all signals being latched by SRAM 0 (shown in bold) have modified AC timing, while those latched by SRAM 1 do not. Therefore, a separate AC timing analysis for each SRAM, using the default processor L3 AC timing for SRAM 1 and the modified processor L3 AC timing for SRAM 0, is necessary. This technique is useful, however, in cases where it is not possible to make the data trace lengths for one of the SRAMs match the associated

L3 clock trace length and some additional hold time is needed for that group of signals. Care must be taken, because the L3 address and control signals are vulnerable to the added clock-to-clock skew because they are shared by both SRAMs; see [Section 3.2, “Clocking and Address and Control Signals \(all SRAM types\).”](#)

For MPC7455 Rev. 3.x and later, it is not possible to introduce clock-to-clock skew in this way because the the bits comprise a 2-bit value that selects among 4 possible values and both L3 clock signals are equally affected by L3OH[0-1].

Unlike earlier devices that implement L2OH bits (such as the MPC755 and MPC7410), changing the output AC timing does not impact the L3 input setup and input hold times because the L3_ECHO_CLK signals are not affected by the settings of these bits in any way.

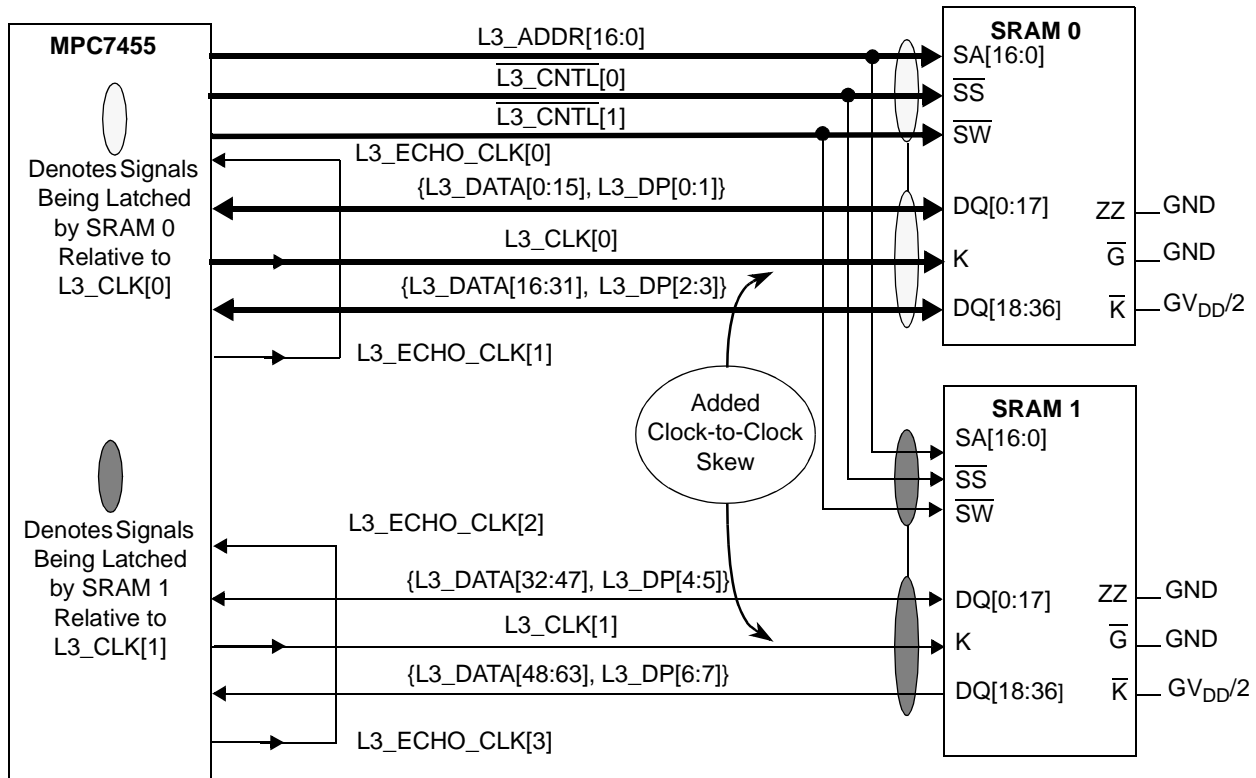


Figure 6. Effects of Setting L3OH[0-1] = 0b01 on L3 Interface Timing for MPC7455 Rev. 2.1 and Prior

2.2 L2 Cache Control Register (L2CR)

For the MPC7455 (only), one bit in the L2CR, L3OH0 (L2CR[12]), is used to configure the L3 interface. For details on this bit, see [Section 2.1.4, “L3 Output Valid Time Adjust \(L3OH1\)—MPC7455-Specific,”](#) and the appropriate hardware specifications for a particular device. This bit is reserved on all MPC7450 family devices except the MPC7455.

2.3 Memory Subsystem Control Register (MSSCR0)

Two bit fields in MSSCR0, L3TCEN, and L3TC, affect the number of cycles the processor will wait before issuing a write transaction to the SRAM following a read transaction, or issuing a read transaction following a write transaction. Note that no delay is added between consecutive read transactions or between consecutive write transactions. Additionally, the MPC7457 supports an additional bit field, L3TCEXT, that allows a longer wait period; this field is not implemented on MPC7450, MPC7451, or MPC7455. For MSUG2 DDR, this turn around time is normally one clock, while it is two clocks for pipelined burst and late write SRAM types. Because it may be desirable to increase this turn around time during system debug, these defaults can be overridden in software by setting L3TCEN. When this bit is set, the turn around time is defined by the value of the L3TC and L3TCEXT fields (and L3TCEXT field for the MPC7457) as shown in Table 4; for reference, this table also shows the default read-to-write turn-around times for each of the supported SRAM types. Adjusting the turn around time may also be useful in special applications where the L3 interface is used to interface with devices other than SRAM; see Section 5.2, “Address Bus Bit Ordering and Alternative Uses of the Backside Cache Interface.” For most applications, L3TCEN and L3TC should normally be cleared.

Table 4. Read-to-Write Turn Around Time Settings

MSSCR0[L3TCEN]	MSSCR0[L3TC]	MSSCR0[L3TCEXT] ¹	L3CR[L3RT]	Turn Around Time (L3CLK Periods)	Comment
0	X	X	0b00	1	Default for MSUG2 DDR
0	X	X	0b01	2	Default for late write
X	X	X	0b10	X	Reserved
0	X	X	0b11	2	Default for pipelined burst
1	0b00	0	X	2	Settings intended for debug use
1	0b01	0	X	3	
1	0b10	0	X	4	
1	0b11	0	X	5	
1	0b00	1	X	6	MPC7457 only; settings intended for debug use
1	0b01	1	X	7	
1	0b10	1	X	8	
1	0b11	1	X	9	

¹ MPC7457-specific; not implemented on MPC7450, MPC7451, or MPC7455.

2.4 L3 Output Hold Control Register (L3OHCR)—MPC7457-Specific

On the MPC7457 is an entire register, L3OHCR, dedicated to controlling the output AC timing. The function of this registers is similar to that of L3OH[0–1] in the MPC7455. However, it is much more flexible and allows for greater control of the output timing. Each L3OHCR field controls the output timing

of a group of signals. [Section 4.1.1, “Using L3OHCR—MPC7457-Specific,”](#) contains examples showing how these bit fields can be used to adjust the AC timing of the interface to correct timing violations.

Table 5. L3OHCR Bit Field Descriptions

L3OHCR Bits	Bit Field Name	Signals Affected	Comment
0–1	L3AOH	L3A[17:0], L3_CTRL[0:1]	Increases input hold time margin and decreases input setup margin at SRAM; affects timing at both SRAM devices
2–4	L3CLK0_OH	L3CLK0	Increases input setup time margin and decreases input hold margin at SRAM
5–7	L3CLK1_OH	L3CLK1	
8-10	L3DOH0	L3_DATA[0:7], L3_DP[0]	Increases input hold time margin and decreases input setup margin at SRAM during write accesses; does not affect read access AC timing
11-13	L3DOH8	L3_DATA[8:15], L3_DP[1]	
14-16	L3DOH16	L3_DATA[16:23], L3_DP[2]	
17-19	L3DOH24	L3_DATA[24:31], L3_DP[3]	
20-22	L3DOH32	L3_DATA[32:39], L3_DP[4]	
23-25	L3DOH40	L3_DATA[40:47], L3_DP[5]	
26-28	L3DOH48	L3_DATA[48:55], L3_DP[6]	
29-31	L3DOH56	L3_DATA[56:63], L3_DP[7]	

2.5 L3 Input Timing Control Registers (L3ITCR_n)

Though L3ITCR_n alter the input AC timing of the MPC745x family, they are intended for factory debug use only. However, they may be useful when debugging suspected AC timing issues associated with data read accesses. These registers are described in the application note *Using the L3ITCR Registers of the MPC7450-family L3 Cache Interface* (AN2580). Note that the effects of these registers are not characterized or tested at this time.

3 Clocks and Timing

The MPC7450 provides two output clocks, L3_CLK[0:1], for use by the SRAM to latch the address, control, and data signals. The L3_ECHO_CLK[0:3] signals are used by the MPC7450 to latch data driven by the SRAM during a read access. The hardware specifications describe several important parameters, namely different types of clock skew, that affect an AC timing analysis of the backside interface. The L3 clock output jitter information is provided so that it can be compared with the input jitter requirements of the SRAM, but the conservative L3 AC timing specifications provided in the hardware specifications already include the output jitter. Therefore, it does not need to be separately considered in an L3 AC timing analysis. Because the clock signals, particularly the L3_ECHO_CLK signals, are used differently for pipelined burst and late write SRAM when compared to MSUG2 DDR SRAM, the subject of clocking for each technology is treated separately for each SRAM technology.

3.1 Differential Clocks

Many SRAM devices have differential clocks while the MPC7450 has single-ended clocks. Most manufacturers do support single-ended clocking of their devices and provide recommendations for how the inverted clock input (\bar{K}) should be terminated. For example, a particular manufacturer may recommend that the \bar{K} input be tied to V_{REF} or $GV_{DD}/2$. These signals are shown as terminated to $GV_{DD}/2$ in the following examples, but the required termination for a particular SRAM device may vary from device to device and the recommendations of the manufacturer should always be followed.

3.2 Clocking and Address and Control Signals (all SRAM types)

As shown in Figure 7, each SRAM latches the shared address and control signals using one of the L3_CLK[0:1] signals. The AC timing of the address and control signals is not associated with one particular L3_CLK signal. Furthermore, the hardware specifications specify $t_{L3CSKW1}$, the skew between L3_CLK[0] and L3_CLK[1], also shown in Figure 7. Because the direction of this skew is not defined, this skew must be deducted from the input setup and input hold margins for the address and control signals of each SRAM. Though Figure 7 shows the diagram for PB2 and late-write SRAM, the above discussion equally applies to DDR SRAM.

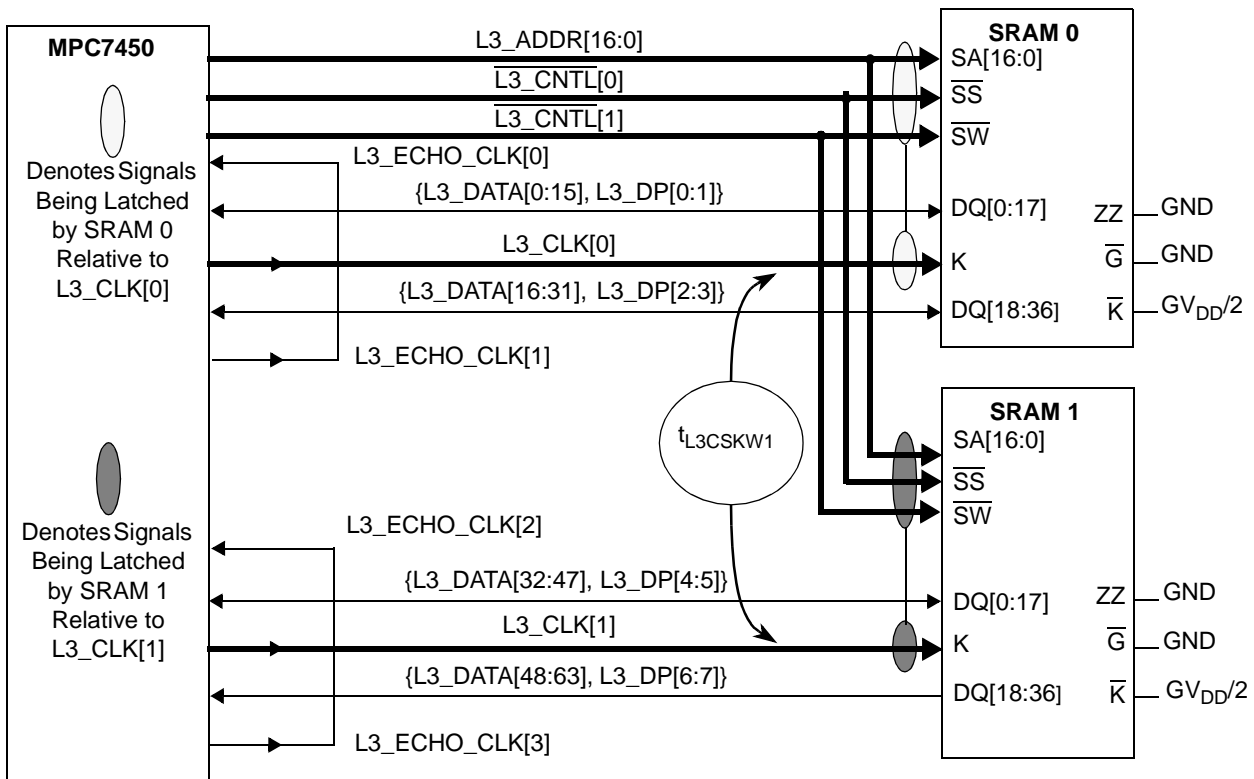


Figure 7. Address and Control Signal Clocking and L3_CLK Clock-To-Clock Skew

3.3 Data Signals and Writes (all SRAM types)

During a write access to the SRAM, each SRAM latches one-half of the data bus (and data parity signals) using one of the L3_CLK[0:1] signals; note that this is true regardless of which SRAM type is used. Unlike

address and control signals, however, the AC timing of each half of the data bus and the associated data parity signals is coupled with an L3 clock signal, as shown in Table 6 and Figure 8. Therefore, $t_{L3CSKW1}$ does not affect the data signals during a write transaction.

Table 6. Clock Groups for Write Accesses (All SRAM Types)

Signal	Associated clock during write access
L3_DATA[0:31]	L3_CLK[0]
L3DP[0:3]	
L3_DATA[32:63]	L3_CLK[1]
L3DP[4:7]	

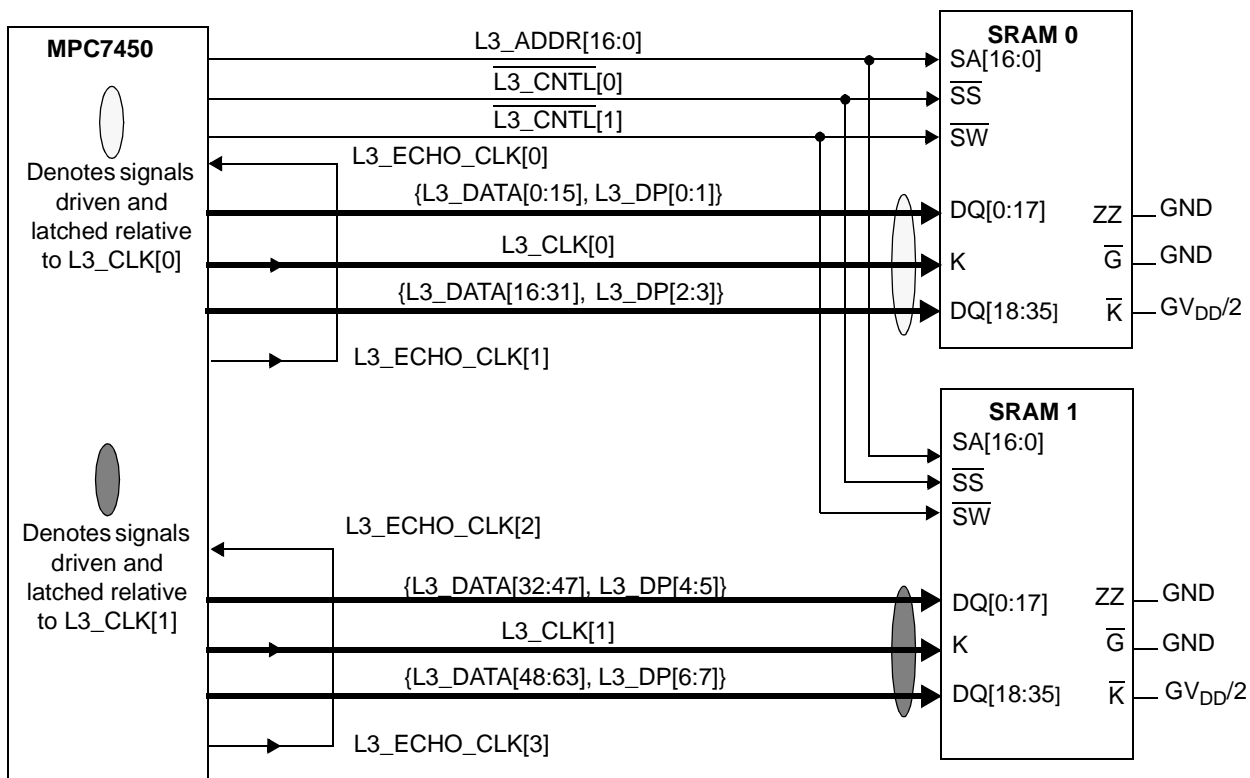


Figure 8. Write Data Clock Grouping

However, this is true only when the signals are routed as shown in Figure 8, or at least such that only signals from L3_DATA[0:31] and L3DP[0:3] (if used) are routed to SRAM 0 regardless of order. Thus, there is a constraint on the amount of optimization during layout. For example, if L3_DATA[32:47] and L3DP[4:5] were routed to SRAM 0 and L3_DATA[16:31] and L3DP[2:3] to SRAM 1, as shown in Figure 9, then $t_{L3CSKW1}$ would affect all of those signals. This is because the MPC7450 is driving L3_DATA[16:31] relative to L3_CLK[0] while SRAM 1 is latching them relative to L3_CLK[1]; likewise, L3_DATA[32:47] are being driven relative to L3_CLK[1] by the processor but being latched relative to L3_CLK[0] by SRAM 0. However, optimizing the data signals within a clock group is

acceptable. For example, no penalty is associated with routing L3_DATA[15] to DQ[16] and L3_DATA[16] to DQ[15] on SRAM 0. While pipelined burst SRAM is shown in the examples in the section, this equally applies other SRAM types.

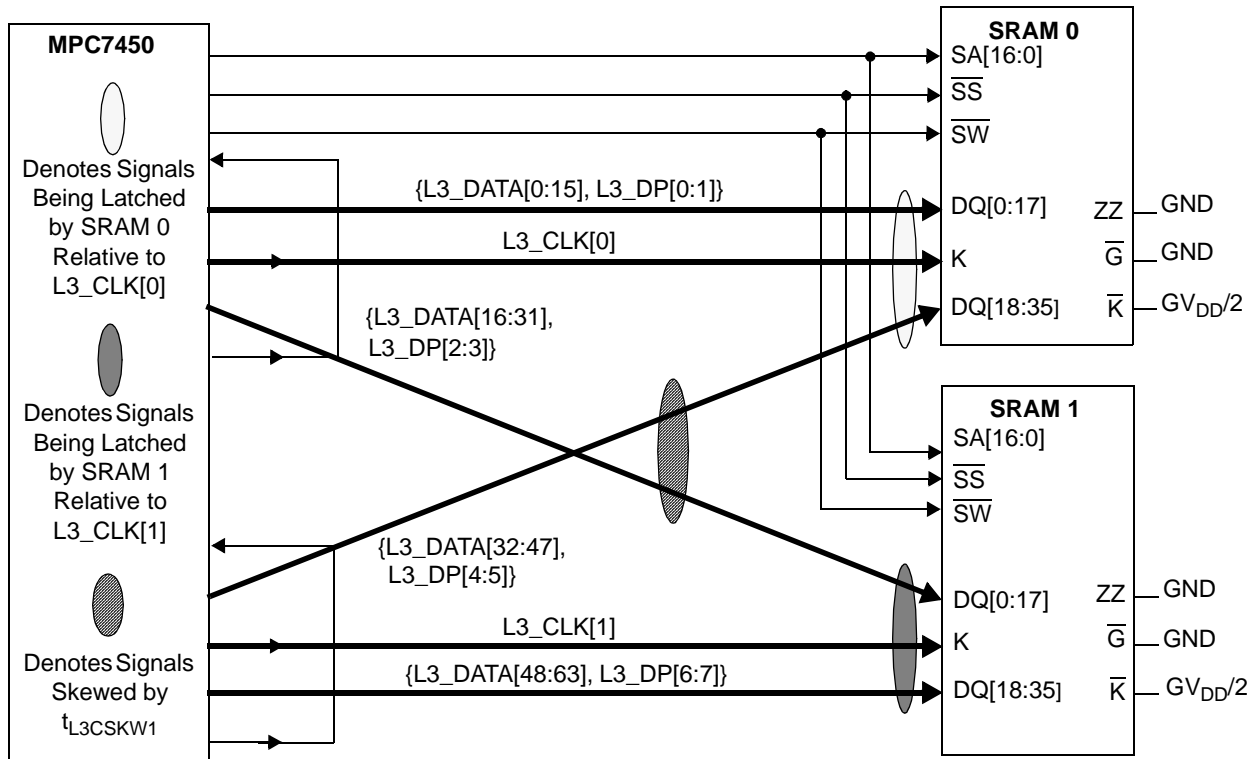


Figure 9. Example Showing Skew Resulting from Incorrect Signal Routing

3.4 Data Signals and Read Accesses

Aside from clocking data on rising and falling edges for DDR SRAM types, the other major difference between the SRAM technology families from the standpoint of system design is how the processor synchronizes data being read from the SRAM. Pipelined burst and late write are handled in the same way and are treated together, while MSUG2 DDR is handled differently.

3.4.1 Data Signals and Reads for Pipelined Burst and Late Write SRAM

Like earlier devices, the MPC7450 uses a synchronization loop to latch data read from pipelined burst and late write SRAM types. Unlike earlier devices that used one loop to synchronize all data being returned from the SRAM, however, the MPC7450 implements two sync loops, one each for half of the data. The first loop is created by routing a trace that begins at L3_ECHO_CLK[1], runs halfway out to SRAM 0, and then back to the L3_ECHO_CLK[0] input. The second loop begins at L3_ECHO_CLK[3], runs halfway out to SRAM 1, and then back to the L3_ECHO_CLK[2] input. As with previous PowerPC ISA processors featuring a backside cache interface, this allows the designer to make corrections in the AC timing budget by adjusting the length of the loop traces. The advantage of two loops, versus the single loop of earlier devices, is that data returning from each SRAM is synchronized individually, allowing more refined adjustments to the timing budget of each SRAM. The disadvantage is that, as shown in Figure 9, fewer

liberties can be taken in optimizing signal routing. Unlike earlier devices, the MPC7450 does not employ a DLL for the L3 interface and the clock edges received at L3_ECHO_CLK[0] and L3_ECHO_CLK[1] are used only to latch data; changing the feedback loop length does not affect L3 output AC timing. See Section 4.2.1, “AC Timing Margins for Pipelined Burst and Late Write SRAM.”

When the length of the synchronization loop is the same as the associated data traces, the input AC timing of the L3 interface meets those published in the hardware specifications. By making this trace longer or shorter, the clocks can be skewed, as described in Section 4.2, “Adjusting AC Timing Margins Using Hardware.” When analyzing AC timing, $t_{L3CSKW2}$, which describes the skew between L3_CLK[0] and L3_ECHO_CLK[1] and between L3_CLK[1] and L3_ECHO_CLK[3], must be subtracted from the processor input setup and input hold margins. This applies only to read transactions for pipelined burst and late write SRAM types. The value of $t_{L3CSKW2}$ is published in the hardware specifications. As shown in Figure 9, signals should not be mixed between clock groups due to the resulting skew ($t_{L3CSKW1}$). Similarly, it is incorrect to connect L3_ECHO_CLK[1] to L3_ECHO_CLK[2] and L3_ECHO_CLK[3] to L3_ECHO_CLK[0].

Table 7. Clock Groups for Read Accesses to Pipelined Burst and Late Write SRAM

Signal	Associated Clock Synchronization Loop During Read Access	Source of Echo Clock Output
L3_DATA[0:31] L3DP[0:3]	L3_ECHO_CLK[1] to L3_ECHO_CLK[0]	L3_CLK[0]
L3_DATA[32:63] L3DP[4:7]	L3_ECHO_CLK[3] to L3_ECHO_CLK[2]	L3_CLK[1]

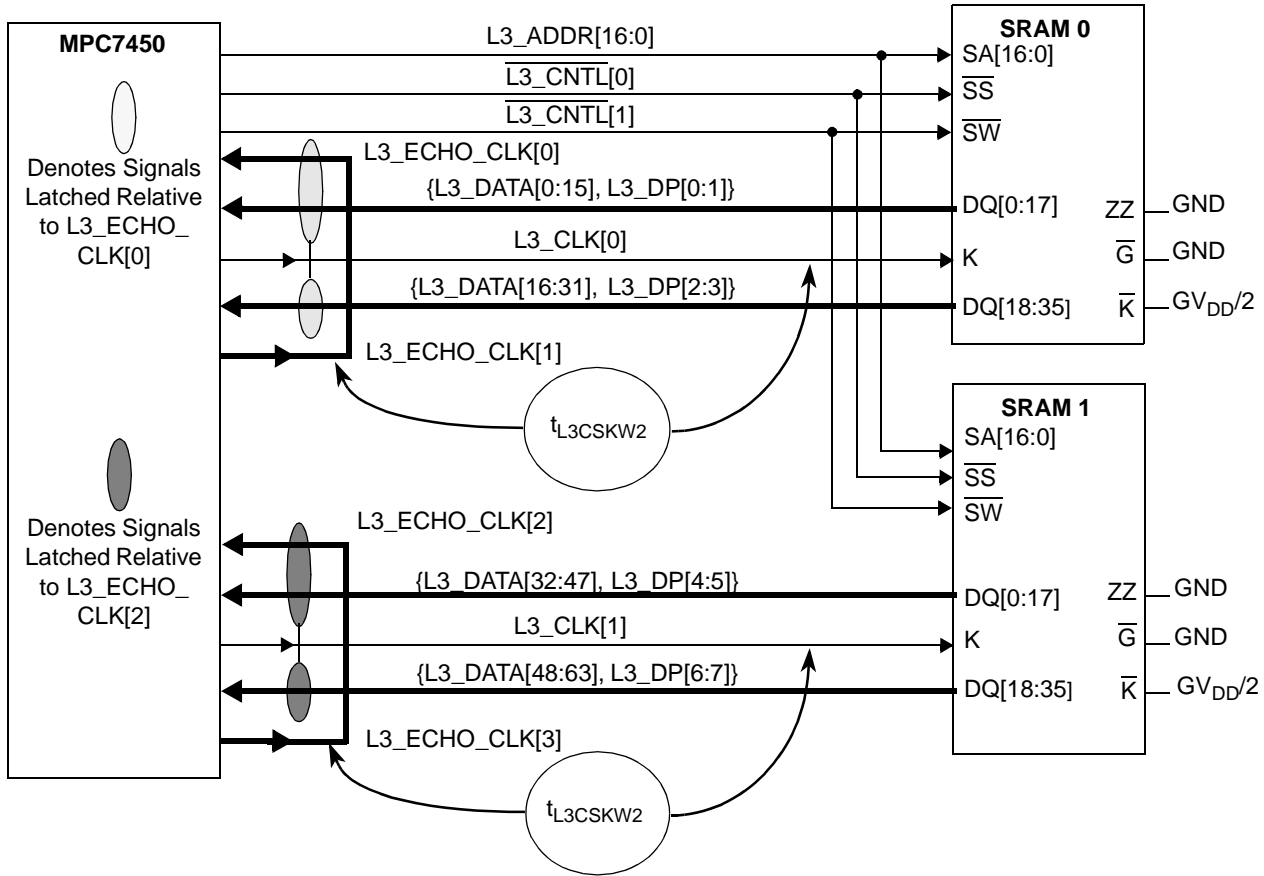


Figure 10. Data Read Clock Grouping for PB2 and Late Write SRAM

3.4.2 Data Signals and Reads for MSUG2 DDR SRAM

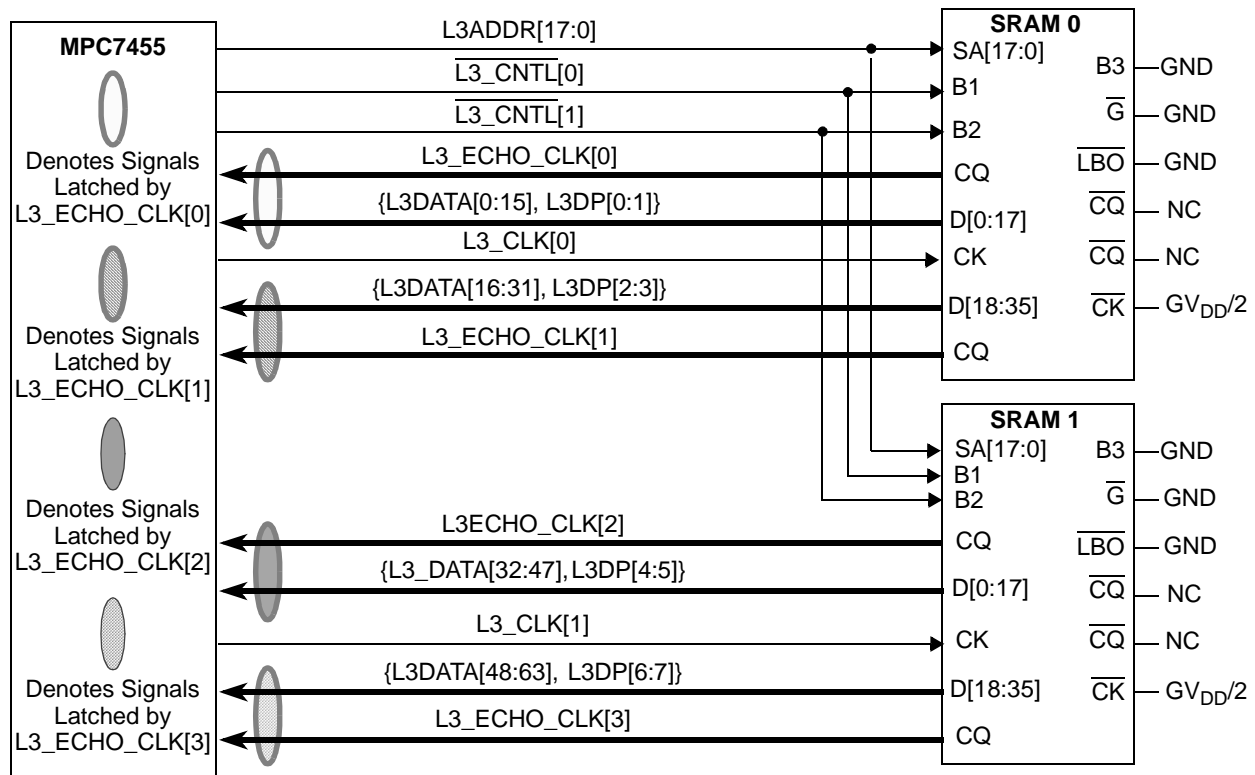
While many of the concerns for pipelined burst and late write SRAM are equally applicable for address and control signals being driven to MSUG2 DDR SRAM, and even for data signals during write transactions, data being returned from DDR SRAM must be considered separately because of the way in which data is being latched by the processor. Feedback loops are not used to synchronize data returning from DDR SRAM. Instead, the SRAM itself supplies a clock that the processor uses to latch the data. Each echo clock input is used to latch two bytes of data and associated parity, as shown in Table 8 and in Figure 11.

Table 8. Clock Groups for Read Accesses to MSUG2 DDR SRAM

Signal	Associated Echo Clock Input
L3_DATA[0:15]	L3_ECHO_CLK[0]
L3DP[0:1]	
L3_DATA[16:31]	L3_ECHO_CLK[1]
L3DP[2:3]	

Table 8. Clock Groups for Read Accesses to MSUG2 DDR SRAM (continued)

Signal	Associated Echo Clock Input
L3_DATA[32:47]	L3_ECHO_CLK[2]
L3DP[4:5]	
L3_DATA[48:63]	L3_ECHO_CLK[3]
L3DP[6:7]	


Figure 11. Data Read Clock Grouping for DDR SRAM

The effects of incorrectly routing signals and mixing clock groups depend on the SRAM and which clock groups are affected. Given the tighter time constraints associated with DDR data transfers, it is all the more critical to avoid adding skew. For example, referencing [Figure 11](#), if the SRAM used specifies a skew between its echo clock (CQ) outputs, this must be subtracted from the processor input setup and input hold margins if L3_DATA[15] and L3_DATA[16] are swapped at the SRAM. (This is not the case at the time of this writing since current MSUG2 DDR SRAM data sheets do not specify any skew between CQ outputs.) Similarly, differences in length of the L3_ECHO_CLK[0] and L3_ECHO_CLK[1] traces must also be included. No signals between the L3_CLK[0] group (that is, L3_DATA[0:31], L3DP[0:3]) should ever be mixed with signals from the L3_CLK[1] group (that is, L3_DATA[32:63], L3DP[4:7]) because the incurred skew ($t_{L3CSKW1}$) during write accesses could be significant in a DDR application and severely limit the maximum attainable frequency.

Note that $t_{L3CSKW2}$ has no meaning in a DDR application and need never be included in any timing analysis because all echo clocks are inputs.

4 Adjusting AC Timing Margins

After a careful timing analysis of the interface, it may be desirable or necessary to adjust the timing. This is most easily accomplished by using the L3OH[0–1] bits (for the MPC7455) or L3OHCR (for the MPC7457). However, there is no such software mechanism for the MPC7450. Similarly, if the available settings do not provide the needed timing relief in an MPC7455 system, it may be necessary to adjust the length of the traces on the routed board. A simple approach to L3 design is as follows:

1. Follow the guidelines in [Section 1.1, “General Design Guidelines.”](#)
2. Initially assume L3_CLK traces equal in length to the average trace length of the associated L3_DATA signals (see [Table 6](#)). Likewise, assume L3_ECHO_CLK feedback trace lengths (for PB2 or late write SRAM), or L3_ECHO_CLK trace lengths (for MSUG2 DDR), equal to the trace length of the associated L3_CLK signal (see [Table 7](#) and [Table 8](#)). This is the baseline and is assumed in the L3 AC timing specifications
3. Perform a detailed timing analysis based on the planned layout (before fabricating the board)
4. Investigate software means (for MPC7455 and MPC7457) of adjusting timing
5. Make adjustments to the trace lengths as needed

Repeat analysis and simulation to verify the correct timing

4.1 Adjusting AC Timing Margins Using Software

For the MPC7455 and MPC7457, it is much better and easier to use software to adjust AC timing margins where possible. The L3OH[0–1] bits allow software to directly adjust the AC timing of the MPC7455 interface. The effects of these bits are straightforward and conveyed directly in the L3 AC timing specifications. See also [Section 2.1.4, “L3 Output Valid Time Adjust \(L3OH1\)—MPC7455-Specific.”](#) The L3OHCR on the MPC7457 is more sophisticated, however, and warrants more discussion of possible uses and benefits.

4.1.1 Using L3OHCR—MPC7457-Specific

The versatility of the L3OHCR allows most address, control, and data write timing errors to be corrected in software. As briefly described in [Section 2.4, “L3 Output Hold Control Register \(L3OHCR\)—MPC7457-Specific,”](#) the L3OHCR can be used to adjust the timing of individual groups of signals. The following examples illustrate how these bits can be used.

4.1.1.1 Example 1: Correcting SRAM Input Hold Time Violations

Referring to [Figure 12](#), suppose L3_DATA[8:15] and L3_DP[1] are routed so that their trace lengths are shorter than the other signals in that clock group. As a result, the transitions on those signals arrive earlier relative to the L3_CLK0 edge than the other signals in the group. This may cause an input hold time violation at the SRAM, as shown in [Figure 13](#). The value of L3DOH8 can be increased to delay these signals such that they meet the SRAM’s requirements, as shown in [Figure 14](#). Note that no other signal groups were affected.

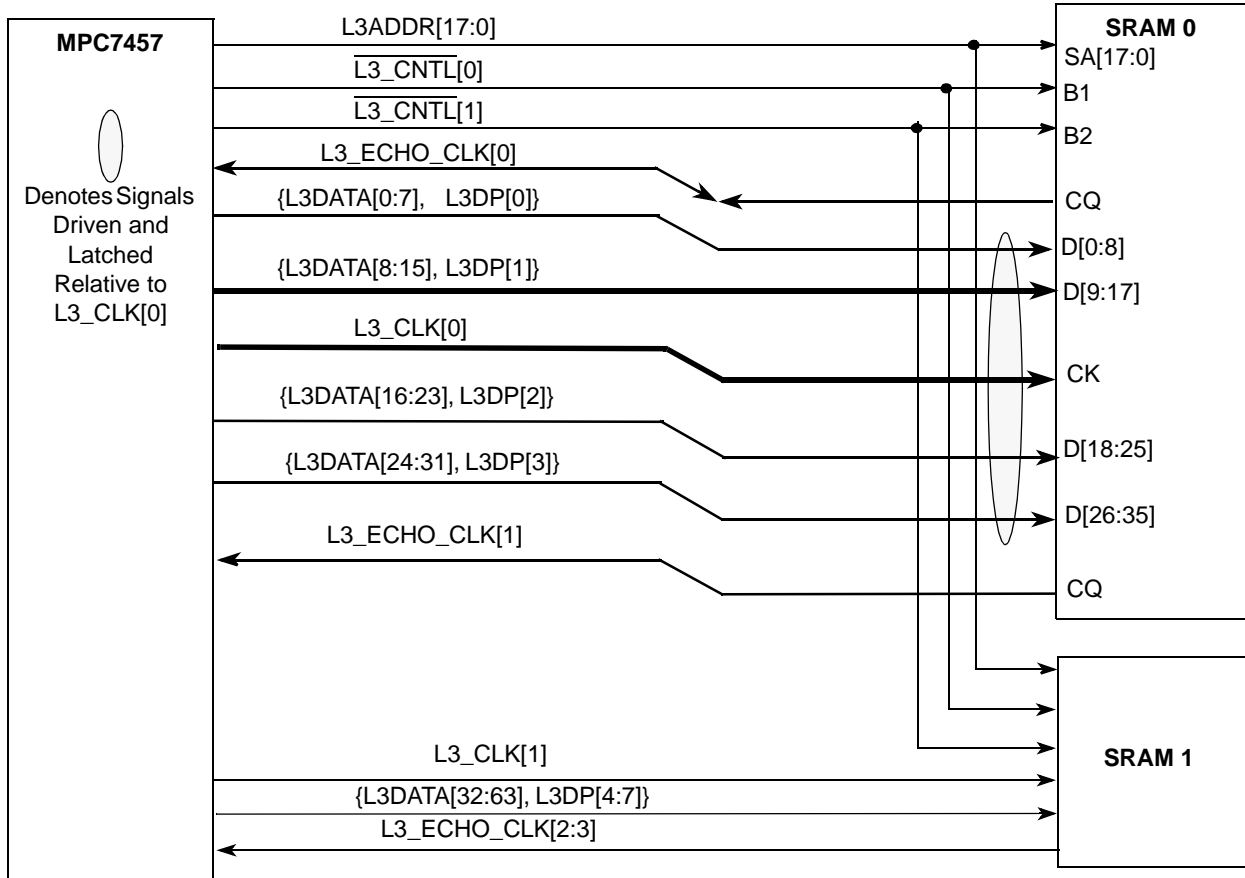


Figure 12. Example 1: Short Traces on L3_DATA[8:15] and L3_DP[1]

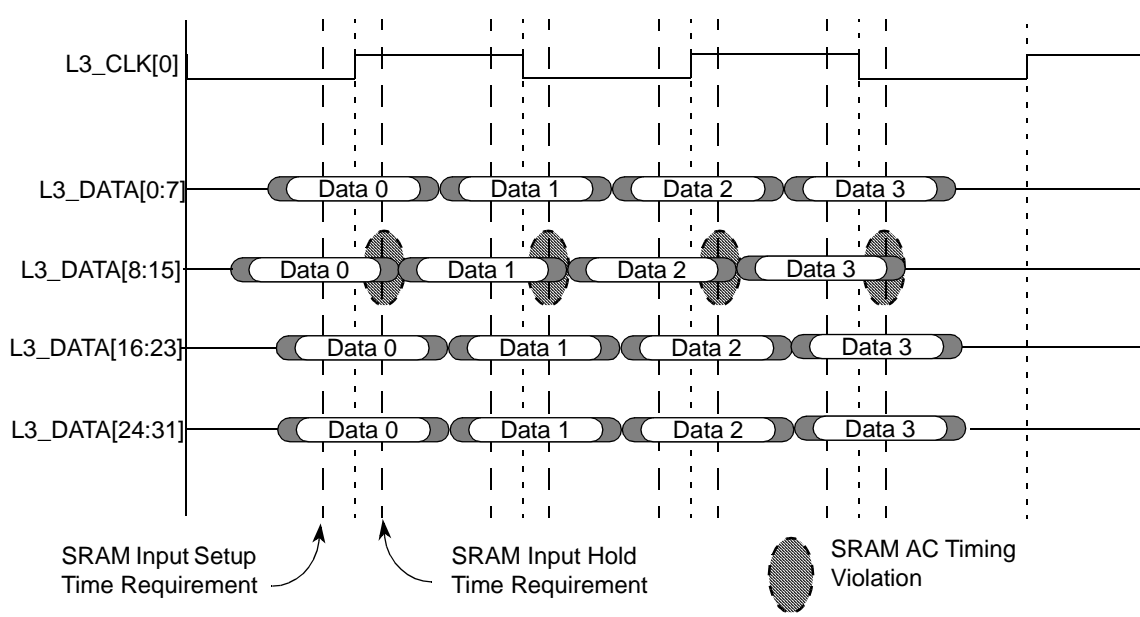


Figure 13. Example 1: Input Hold Time Violation on L3_DATA[8:15] and L3_DP[1]

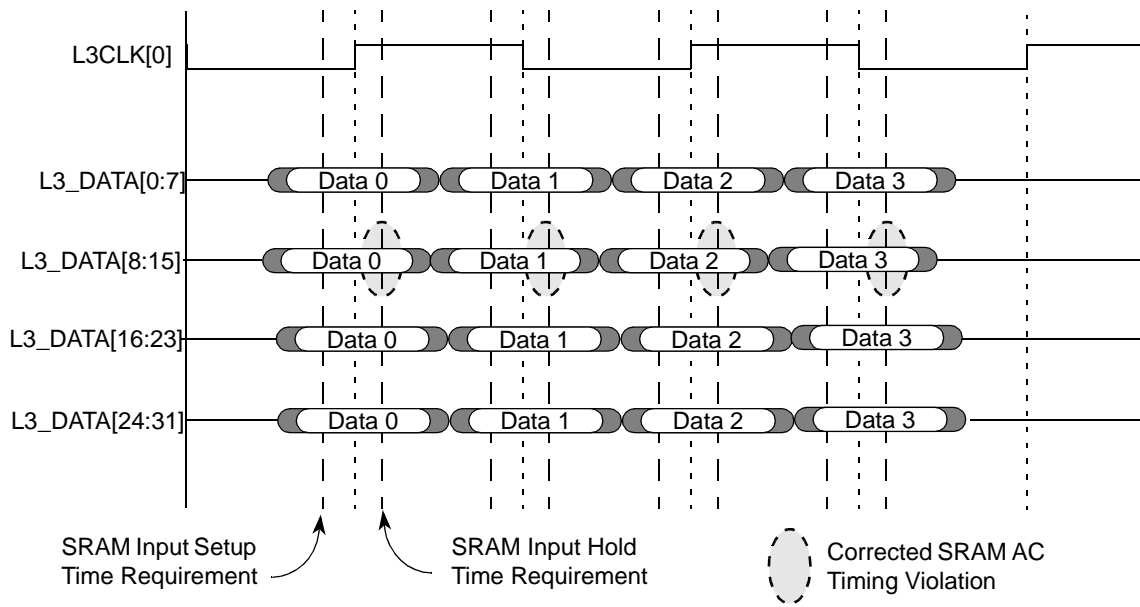


Figure 14. Example 1: Input Hold Time Violation Corrected using L3OHCR[L3DOH8]

4.1.1.2 Example 2: Correcting SRAM Input Setup Time Violations

Just as L3AOH can be used to delay the address and control signals, and L3DOH n to delay the data signals, L3CLK0_OH and L3CLK1_OH can be used to delay the L3 clock signals. However, the goal is not to improve SRAM input hold time margins but to delay the clocks relative to the other signals to improve SRAM input setup time margins. Example 2, shown in Figure 15, considers a case opposite to Example 1. Here the trace lengths for L3_DATA[8:15] and L3_DP[1] are longer than the other traces, causing transitions on these signals to arrive later relative to the L3_CLK0 edge than the other signals in the group. This creates an SRAM input setup time violation, as shown in Figure 16.

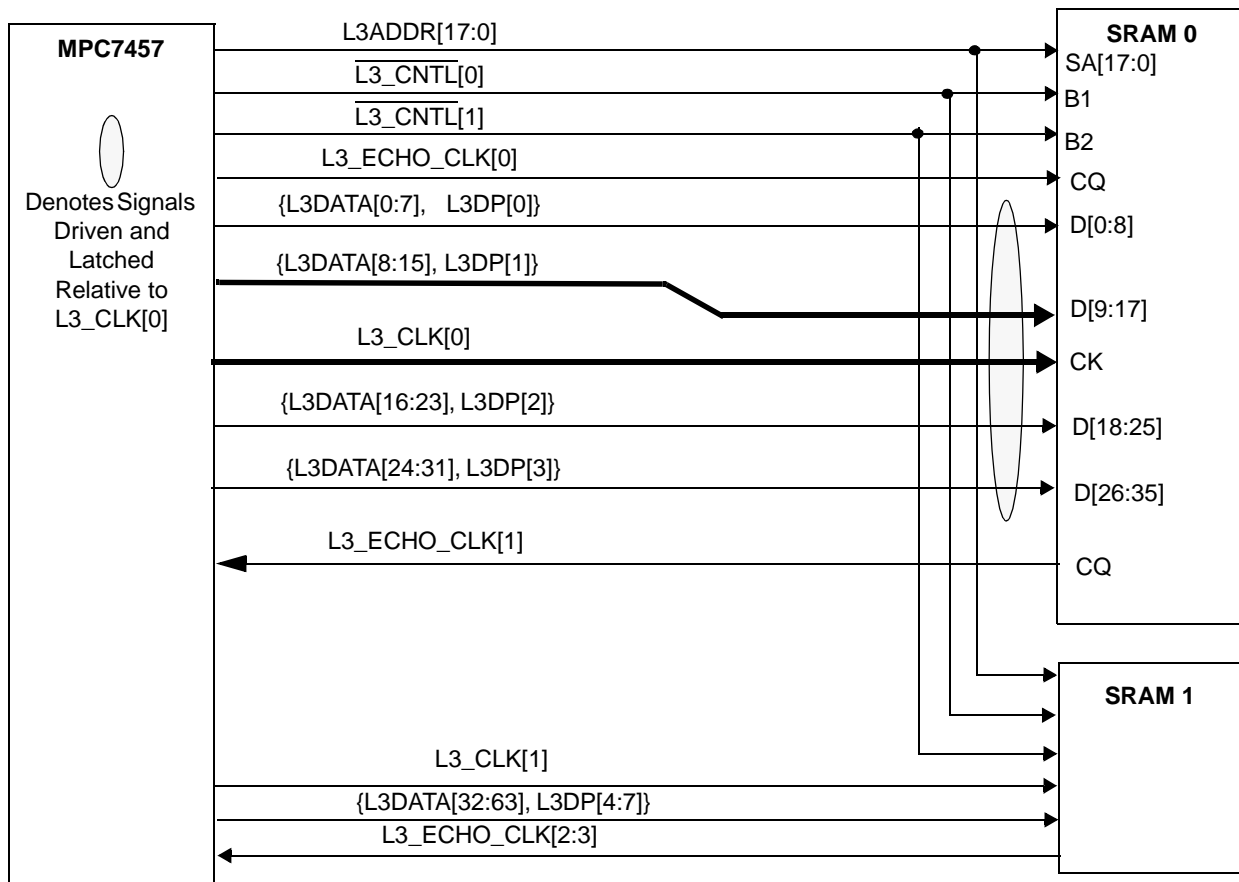


Figure 15. Example 2: Long Traces on L3_DATA[8:15] and L3_DP[1]

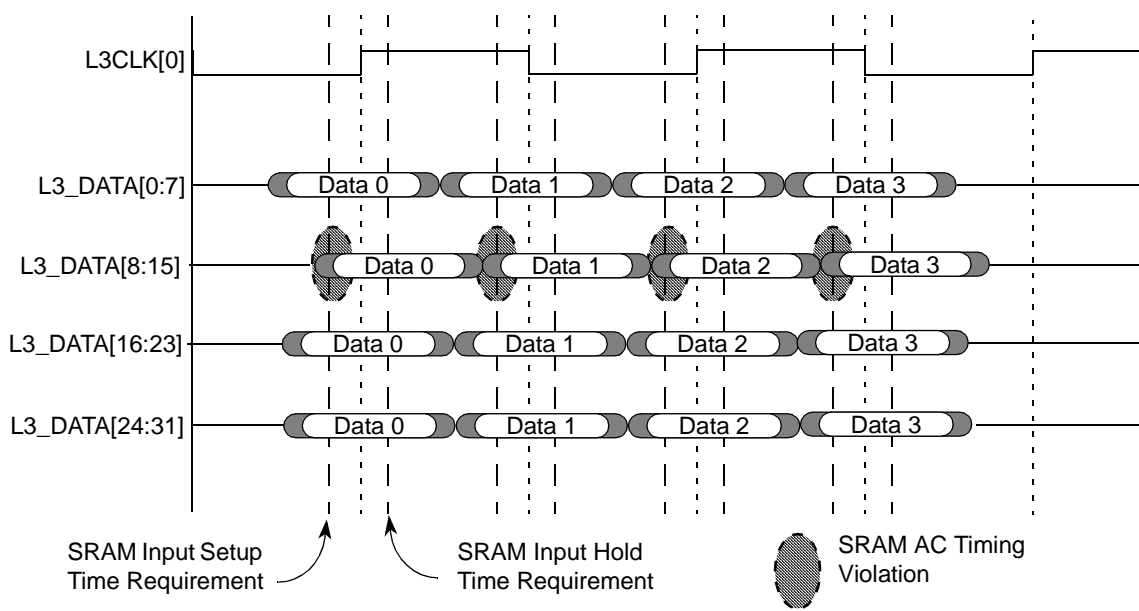


Figure 16. Example 2: Input Setup Time Violation on L3_DATA[8:15] and L3_DP[1]

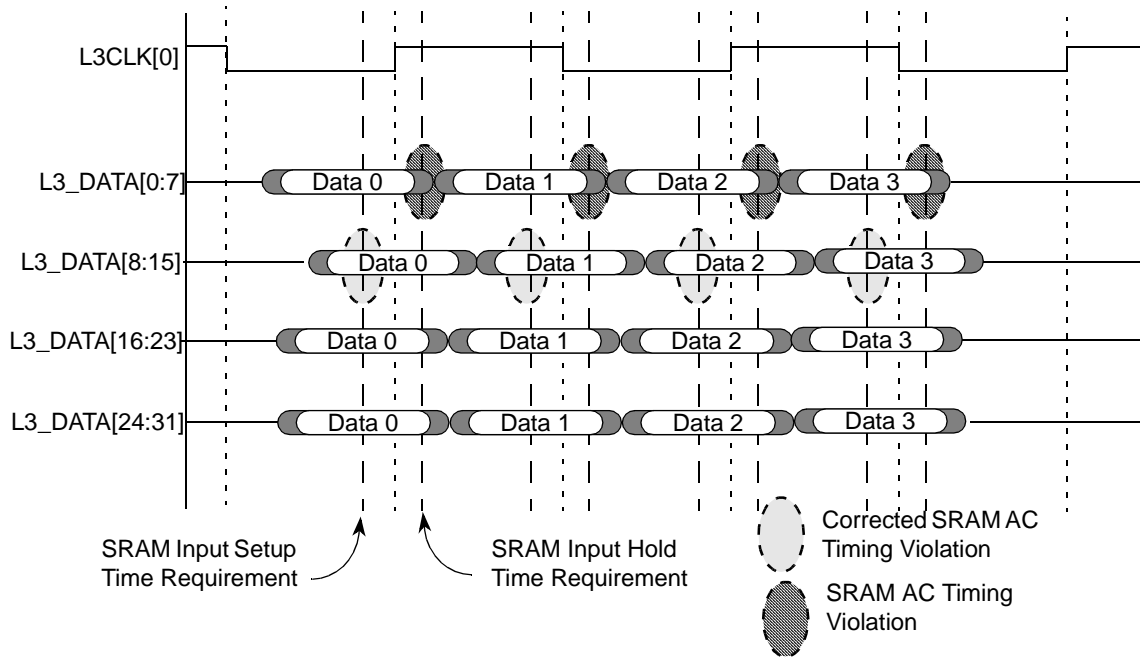


Figure 17. Example 2: Input Setup Time Violation Corrected using L3OHCR[L3CLK0_OH]

As shown in [Figure 17](#), delaying an L3 clock makes all signal transitions appear earlier relative to the clock. Take care when using L3CLK_n_OH because delaying the clocks affects the AC timing of all inputs of the SRAM receiving the clock in question (including the address and control signals, not shown in these examples). In this case, shifting the clocks created an SRAM input hold time violation. Fortunately, control of the output delays in L3OHCR are independent of each other. Therefore, L3DOH0 can be used to correct the timing violation created by adjusting L3CLK0_OH, as shown in [Figure 18](#).

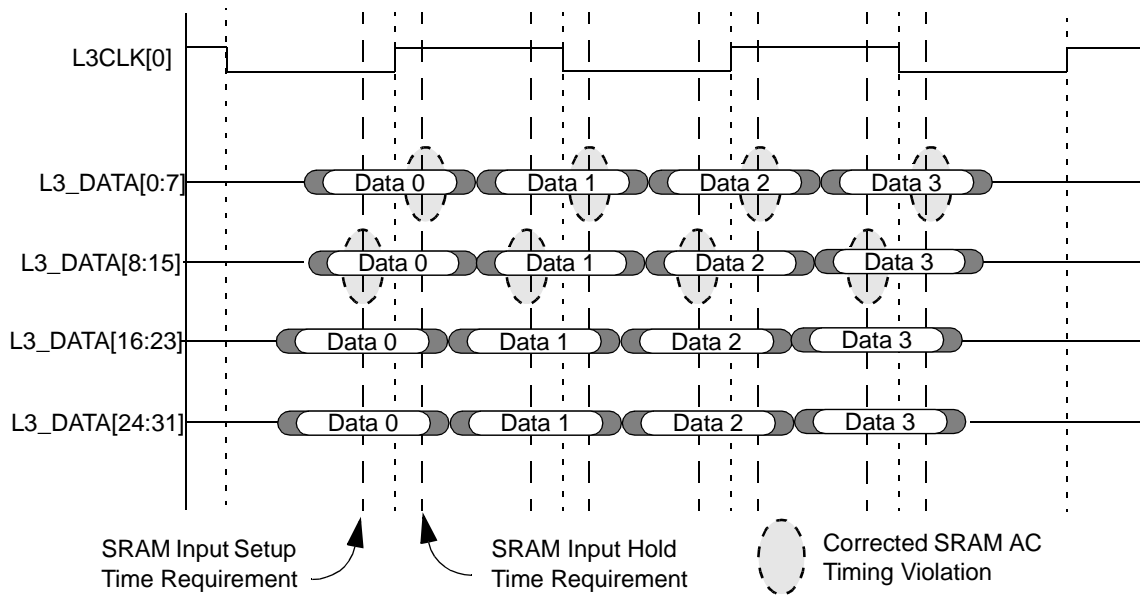


Figure 18. Example 2: Input Setup and Hold Time Violations Corrected using L3OHCR

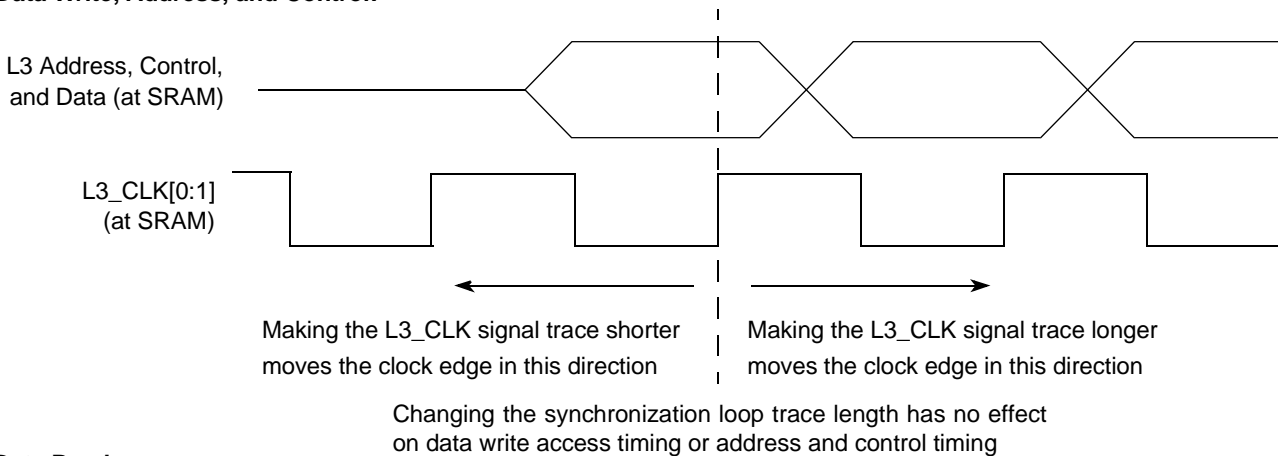
4.2 Adjusting AC Timing Margins Using Hardware

When it is not possible to correct timing margins in software, the only recourse is to adjust trace lengths to skew signals. If you follow the recommendations in [Section 1.1, “General Design Guidelines”](#) most adjustments to timing margins are made by adjusting the lengths of L3 clock or echo clock traces. The following sections briefly describe the main concerns in doing so for the various SRAM technologies.

4.2.1 AC Timing Margins for Pipelined Burst and Late Write SRAM

Adjusting AC timing margins for pipelined burst and late write SRAMs requires care because of the interdependence of read and write timing. Adjusting the feedback loop trace length while leaving the L3_CLK trace length unchanged affects only the AC timing margins for data read accesses, but changing the L3_CLK trace length while leaving the synchronization loop length unchanged affects both read and write access AC timing. The relationship between the relative length of the feedback loop and the associated signals is straightforward and is shown in [Figure 19](#) and summarized in [Table 9](#).

Data Write, Address, and Control:



Data Read:

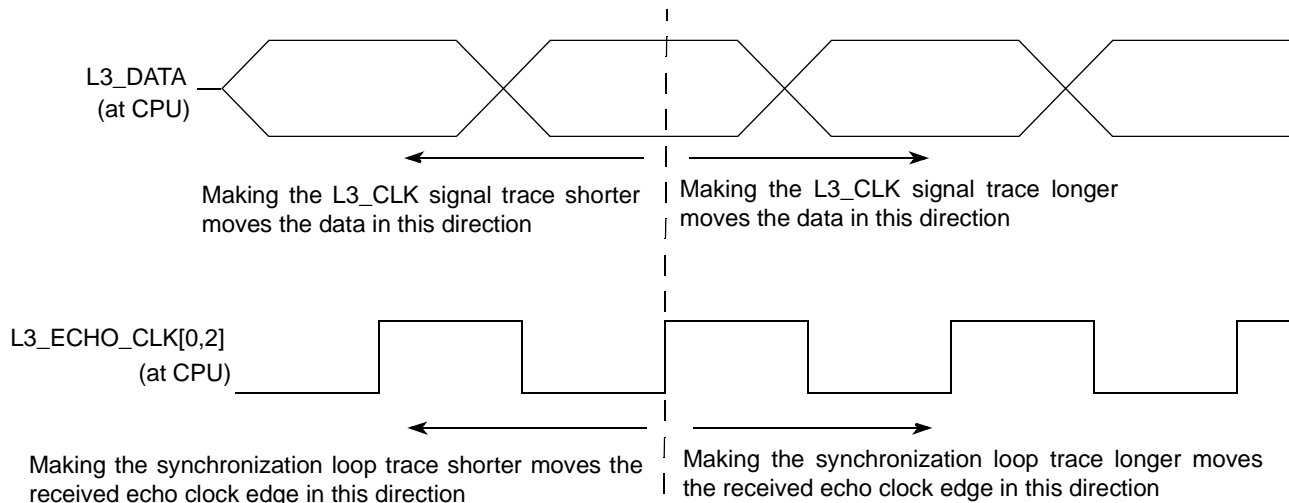


Figure 19. Adjusting the Length of Clock and Synchronization Loop Traces for PB2 and Late Write SRAM

Table 9. Effects of Clock Trace Lengths on AC Timing Margins for PB2 and Late Write SRAM

Parameter	Change in L3_CLK Trace Length (Relative to Other Signals in Clock Group)		Change in Synchronization Loop Length (Relative to Other Signals in Clock Group)	
	Shorten	Lengthen	Shorten	Lengthen
SRAM input setup margin	Decrease	Increase	None	None
SRAM input hold margin	Increase	Decrease	None	None
Processor input setup margin	Increase	Decrease	Decrease	Increase
Processor input hold margin	Decrease	Increase	Increase	Decrease

The amount the parameters increase or decrease is determined by the amount of propagation delay added or subtracted from the trace lengths. Changes in L3_CLK and synchronization loop lengths are independent and cumulative. Shortening or lengthening the L3_CLK trace lengths affects all of the parameters and always by an equal amount. For example, shortening the L3_CLK trace lengths such that 100 ps of propagation delay is added increases the SRAM input setup and processor input hold margins by 100 ps while simultaneously decreasing the SRAM input hold and processor input setup margins by 100 ps. To change only the SRAM input timing margins while leaving the processor input timing margins unchanged, an equal amount of propagation delay must be added to the feedback loop to offset the effects of changing the L3_CLK trace lengths. The amount of actual trace length to remove or add to achieve a given change in a propagation time is design dependent and should be determined after careful analysis and simulation of the board and trace properties. Determining propagation delay is beyond the scope of this document but some references in the bibliography discuss the topic in detail, particularly *High-Speed Digital Design: A Handbook of Black Magic*.

4.3 Adjusting AC Timing Margins for MSUG2 DDR SRAM

Adjusting AC timing margins for MSUG2 DDR is simplified because the independence of the echo clocks from the L3 clocks allows flexibility in adjusting timing margins: the input timing at the SRAM can be adjusted without affecting the input timing at the processor, and *vice versa*. Changing the length of an L3_CLK trace affects only the SRAM input setup and hold margins, and changing the length of an L3_ECHO_CLK trace affects only the processor input setup and hold margins. For the MPC7455, the L3OH bits may be useful, depending on the timing requirement violated and the device revision in question. In other cases, it may not be possible to use L3OH to correct timing problems. For the MPC7457, the L3OHCR provides enough granularity and flexibility to allow adjustments entirely in software without need to resort to hardware fixes in most systems. For the MPC7450 and MPC7451, which have neither L3OH[0–1] or the L3OHCR, adjusting trace lengths is the only way to adjust AC timing margins.

4.3.1 Adjusting Address, Control, and Data Write Timing

For the MPC7455, L3OH[0–1] offer the simplest method of adjusting the AC timing of the L3 interface. Similarly, L3OHCR settings can be used for the MPC7457 to adjust output AC timing. Changing the length of an L3 clock trace length to skew a the clock affects the input setup and hold time margins of all signals (data, address, and control) received by that SRAM. This does not affect the data input setup and input hold margins for data read from the processor because those signals are latched by the processor using an echo clock input.

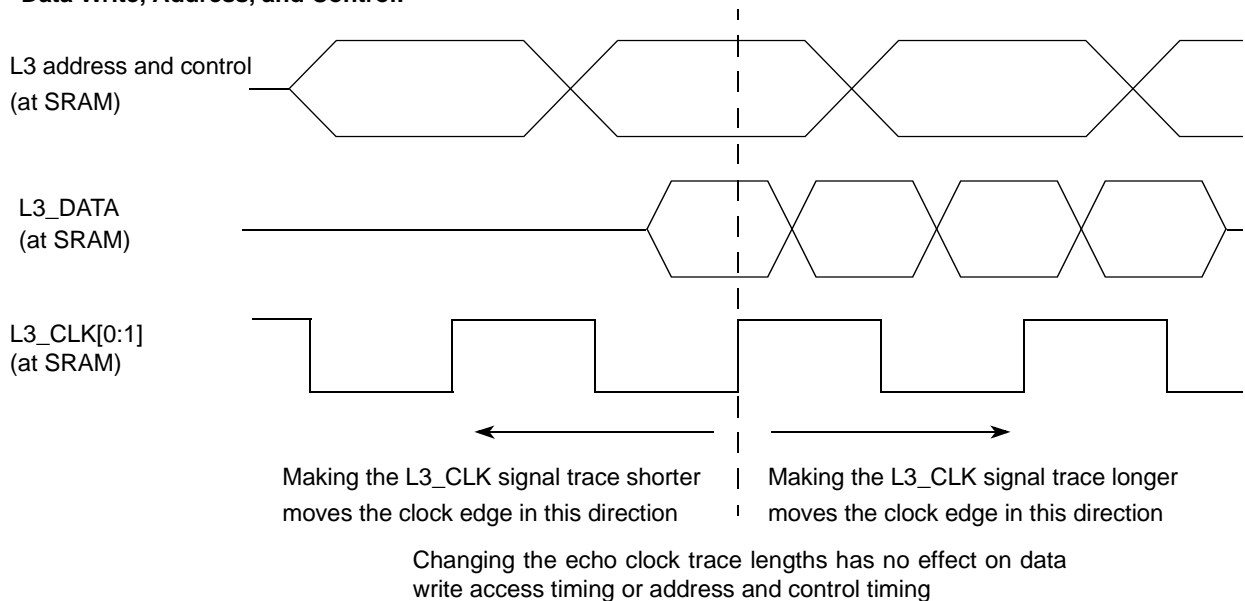
4.3.2 Adjusting Data Read Timing

To change the processor input setup and input hold margins, the length of the corresponding L3 echo clock trace should be changed. This does not affect the input setup and input hold margins at the SRAM.

Table 10. Effects of Clock Trace Lengths on AC Timing Margins for DDR SRAM

Parameter	Change in L3_CLK Trace Length (Relative to Other Signals in Clock Group)		Change in Echo Clock Trace Length (Relative to Other Signals in Clock Group)	
	Shorten	Lengthen	Shorten	Lengthen
SRAM input setup margin	Decrease	Increase	None	None
SRAM input hold margin	Increase	Decrease	None	None
Processor input setup margin	None	None	Decrease	Increase
Processor input hold margin	None	None	Increase	Decrease

Data Write, Address, and Control:



Data Read:

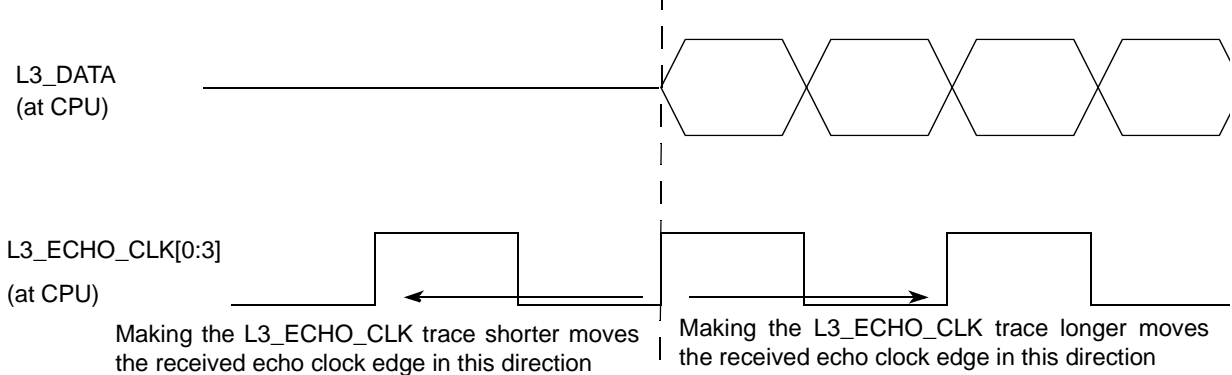


Figure 20. Adjusting the Length of Clock and Echo Clock Traces for DDR SRAM

5 Special Considerations for the L3 Address Bus

Special considerations for the bus include burst accesses and MSUG2 DDR as well as address bus bit ordering and alternative uses of the backside cache interface.

5.1 Burst Accesses and MSUG2 DDR

When pipelined burst or late write SRAM is accessed, the MPC7450 does not use the burst feature of the SRAM. Instead, the processor issues a unique address for each beat of data. However, this is not possible with DDR SRAM types because address and control signals are essentially SDR inputs (that is, they are driven/sampled only on rising clock edges). Therefore, the burst feature of DDR SRAM must be used. For MSUG2 DDR, the MPC7450 uses two-beat bursts to access SRAM. That is, one address is issued for every two beats of data, a cache line access requiring two such bursts to transfer all 32 bytes.

The order of the address bits is significant for DDR. With pipelined burst or late write SRAM, the address bits can be reordered without worry because the SRAM ‘sense’ of the address space has no effect on the operation of the interface and there is a 1:1 relationship between an address and the data stored there. For example, the processor can drive address *X* on the L3 address bus, which the SRAM in turn interprets as address *Y*, but there is no error because the SRAM accesses address *Y* if the processor drives address *X*.

Because the burst feature must be used with DDR SRAM the processor does not supply a unique address for every beat of data. Instead, the processor supplies one address and the SRAM accesses two addresses, the one supplied by the processor and another determined by the SRAM itself (for example, *X* and *X* + 1). Therefore, the SRAM’s ‘sense’ of the address space becomes significant. For example, consider a case with the L3 address bus reversed at the SRAM and connectivity as shown in [Table 11](#).

Table 11. MSUG2 DDR Address Wiring Error Example

MPC7450 Address Pin	SRAM Address Pin
L3_ADDR[17]	SA[0]
L3_ADDR[16]	SA[1]
⋮	⋮
L3_ADDR[1]	SA[16]
L3_ADDR[0]	SA[17]

In this situation, the address interpreted by the SRAM is the mirror image of that driven by the processor. For example, the SRAM interprets address 0x00550 (as driven by the processor) as address 0x0AA00. Now consider the case where the processor accesses L3_ADDR[17:0] = 0x00000. The reversed address bits make no difference here and the SRAM internally accesses 0x00000 and 0x00001, supplying the expected data to the processor. If the processor initiates an access to L3_ADDR[17:0] = 0x20000 some time later, however, the SRAM then internally accesses address 0x00001 because of the reversed address bits. The problem is immediately clear: two addresses (0x00001 and 0x20000) in the processor sense of the address space map to address 0x00001 in the SRAM sense of the address space. The result is corrupted data or instructions or modified data being overwritten and lost. To avoid this situation, always connect the L3 address bus signals in the correct order, particularly the lowest 2 bits (L3_ADDR[1:0]).

5.2 Address Bus Bit Ordering and Alternative Uses of the Backside Cache Interface

The address driven on the L3 address bit is derived from the physical address (PA) of the access that caused it (as opposed to the effective or virtual address). However, the order of the address bits with respect to the physical address is not sequential. The order of some bits changes when compared to the physical address. This is of no particular concern in a typical application where, by nature of being a backside cache bus, only the MPC7450 would ever access the SRAM. Therefore, the actual appearance of the address on the L3 bus is immaterial because no device ever needs to decode it. However, system designers have expressed interest in using the L3 interface with dual-ported SRAM or programmable logic devices as a fast, dedicated bus. While this goes beyond the original intent of the interface, such an application of the interface is possible if carefully considered and properly implemented. The private memory feature is likely to be very useful in such an implementation because it configures the interface in non-cache mode so that the L3 tags are not used and accesses to private memory space never access the system bus interface. However, the private memory space must be cacheable and the L1 data cache enabled for data accesses to reach the interface. This is because cache-inhibited accesses bypass the L3 interface (including private memory) and data accesses are treated as cache-inhibited if the L1 data cache is disabled. (For details, see the *MPC7450 RISC Microprocessor Family User's Manual*.) Also, private memory is non-coherent with respect to system memory and other devices in the system must not issue global (\overline{GBL} asserted) accesses (that is, transactions to be snooped) to the private memory address range. Failure to observe this requirement can result in loss of private memory data if the data resides in the L1 or L2 cache and a snoop hit occurs.

The first step is to use a dual-ported SRAM compatible with one of the supported SRAM types or to design the custom logic such that it conforms to the interface protocol of one of them. If it is not possible to make the external device exactly match the protocol of any of the supported SRAM types, it may still be possible to choose the mode that it most closely matches and then use some of the additional configuration options described in earlier sections, such as L3TC, to alter the configuration of the interface.

The second step is to understand the ordering of the address bits. This is provided in [Table 12](#).

Table 12. L3 Address to Physical Address Bit Mapping

L3 Address Bit	Private Memory Size and Corresponding Physical Address Bit ¹		
	1M	2M	4M ²
L3_ADDR[18] ³	—	PM ⁴	17
L3_ADDR[17]	PM ⁵	18	18
L3_ADDR[16]	19	19	19
L3_ADDR[15]	20	20	20
L3_ADDR[14]	21	21	21
L3_ADDR[13]	22	22	22
L3_ADDR[12]	23	23	23
L3_ADDR[11]	24	24	24

Table 12. L3 Address to Physical Address Bit Mapping (continued)

L3 Address Bit	Private Memory Size and Corresponding Physical Address Bit ¹		
	1M	2M	4M ²
L3_ADDR[10]	25	25	25
L3_ADDR[9]	26	26	26
L3_ADDR[8]	27	27	27
L3_ADDR[7]	28	28	28
L3_ADDR[6]	29	29	29
L3_ADDR[5]	30	30	30
L3_ADDR[4]	16 (Way 2)	15 (Way 2)	14
L3_ADDR[3]	17 (Way 1)	16 (Way 1)	15
L3_ADDR[2]	18 (Way 0)	17 (Way 0)	16
L3_ADDR[1]	31	31	31
L3_ADDR[0]	32	32	32

¹ Entries in parentheses indicate the function of these bits when the L3 is operated in cache mode. These bits reflect which of the eight ways is being accessed. When operating in private memory mode, the bits reflect the stated bits of the physical address.

² MPC7457-specific; not supported on other devices. Note that because the MPC7457 does not support 4M of L3 as cache, 4M of SRAM must be configured as either 4M of private memory or 2M of private memory and 2M of cache; see the *MPC7450 RISC Microprocessor Family User's Manual* for more information.

³ MPC7457-specific; not implemented on other devices.

⁴ For MPC7457 only, the value of this bit depends on the size of the SRAM and the private memory space. For 4M of SRAM configured as 2M of cache and 2M of private memory, this bit will be driven low (0b0) for cache accesses and high (0b1) for private memory accesses. For 2M of SRAM configured as private memory (only), this bit is always driven low (0b0).

⁵ The value of this bit depends on the size of the SRAM and the private memory space. For 2M of SRAM configured as 1M of cache and 1M of private memory, this bit is driven low (0b0) for cache accesses and high (0b1) for private memory accesses. For 1M of SRAM configured as private memory (only), this bit is always driven low (0b0).

The final step is to determine the information that affects the software configuration, such as L3 clock frequency, correct L3OH settings and correct sample point settings.

6 References and Revision History

The reference materials shown in [Table 13](#) may be useful to the reader. Several concepts mentioned in this application note are described in detail in these documents.

Table 13. Reference Documentation

Title	Author	Order Number
<i>High-Speed Digital Design: A Handbook of Black Magic</i>	Howard Johnson and Martin Graham	Prentice-Hall ISBN 0-13-395724-1
<i>PowerPC Backside L2 Timing Analysis for the PCB Design Engineer</i>	Bruce Parker	AN1794
<i>Setting the Sample Points for the MPC7450 L3 Cache</i>	Michael Everman	AN2182
<i>Using the L3ITCR Registers of the MPC7450-family L3 Cache Interface</i>	Michael Everman	AN2580

Table 14 lists the revision history for this application note.

Table 14. Document Revision History Table

Rev. No.	Substantive Changes
0	Initial release.
1	Added Sections 2.1.1.1 and 2.1.2.1
	Corrected L3 turn-around count description in Section 2.3.
	Corrected entries for 4M column in Table 12 .

THIS PAGE INTENTIONALLY LEFT BLANK

THIS PAGE INTENTIONALLY LEFT BLANK

How to Reach Us:

Home Page:

www.freescale.com

email:

support@freescale.com

USA/Europe or Locations Not Listed:

Freescale Semiconductor
Technical Information Center, CH370
1300 N. Alma School Road
Chandler, Arizona 85224
1-800-521-6274
480-768-2130
support@freescale.com

Europe, Middle East, and Africa:

Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
support@freescale.com

Japan:

Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku
Tokyo 153-0064, Japan
0120 191014
+81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:

Freescale Semiconductor Hong Kong Ltd.
Technical Information Center
2 Dai King Street
Tai Po Industrial Estate,
Tai Po, N.T., Hong Kong
+800 2666 8080
support.asia@freescale.com

For Literature Requests Only:

Freescale Semiconductor
Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
1-800-441-2447
303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductor products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters which may be provided in Freescale Semiconductor data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including "Typicals" must be validated for each customer application by customer's technical experts. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify and hold Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.

Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc. The Power Architecture and Power.org word marks and the Power and Power.org logos and related marks are trademarks and service marks licensed by Power.org. All other product or service names are the property of their respective owners.

© Freescale Semiconductor, Inc., 2003, 2006.

Document Number: AN2468

Rev. 1
11/2006

