

Application Note

 AN1794/D
 Rev. 0.2, 5/2003

*Backside L2 Timing Analysis
 for PCB Design Engineers*

The backside L2 interfaces on the MPC750 and the G4 processors dramatically increase their performance. Since the L2 design basically connects the processor's L2 controller to the memory SRAMs, the main task for the board designer is to determine what board propagation delays will provide sufficient setup and hold margins for a given target frequency. This document discusses how to determine the propagation delay restrictions for the backside L2 interface of PowerPC™ processors and a method for optimizing the setup and hold margins by using clock offsets.

Setup and hold margins are controlled partially by the processor and memory timing specifications, and partially by the PCB propagation and skew characteristics. Using the various timing specifications from the processor and memory data sheets, one can calculate the minimum and maximum propagation delay allowable. Once these delays are known for all cycle types (that is, address, data write, and data read cycles) one can optimize the setup and hold margins by adding delay to either the processor's L2 feedback clock or to the memory's clock. Once an offset, if any, is calculated, the final allowable propagation delays can be converted to length restrictions for use by the PCB layout designer.

The following definitions are used during the analysis:

- t_{co_src} —Time from the rising edge of the clock until the output becomes valid; specified by the source of the output signal
- t_{oh_src} —Time from the rising edge of the clock until the output becomes invalid; specified by the source of the output signal
- t_{su_rcvr} —Time that an input must be valid before the rising edge of the clock; specified by the receiver of the signal
- t_{ih_rcvr} —Time that an input must remain valid after the rising edge of the clock; specified by the receiver of the signal
- t_{jitter} —Clock jitter from L2 clock source (cycle-to-cycle)
- t_{cksk} —Clock skew introduced from PCB routing and clock loading differences; this includes unintentional length mismatches, skew from PCB impedance differences, and skew due to clock loading differences.
- t_{per} —Clock period of L2 interface
- t_{prop} —Time for a signal to propagate from a driver's output to a receiver's input
- $t_{ckoffset}$ —An intentional offset between the L2 controller's feedback clock and the memory's clock. This offset is calculated later in this document.

1 Procedure and Analysis

The initial analysis is just a typical timing analysis used to calculate setup and hold margins. The L2 feedback clock is assumed to be equal in length to the memory clock length, thus both the processor's L2 controller and the memory devices clock signals in and out at the same instance in time. The second part of this analysis investigates how offsetting the clocks can optimize timing margins.

The equations below are used to calculate the minimum and maximum propagation delays. For the L2 interface, the analysis must consider both address and data cycles. From a propagation delay calculation viewpoint, an address cycle and a data write cycle have the same equation since the L2 controller has the same timing specifications for address and data outputs.

2 Setup Time Margin and Maximum Propagation Delays

The setup margin can be defined as the difference between the clock period, (t_{per}), and the total setup time, t_{su_total} , as shown below:

$$t_{su_margin} = t_{per} - t_{su_total} \quad (EQ 1)$$

The total setup time is the accumulation of the driver's clock to output valid (t_{co_src}), the receiver's input setup (t_{su_rcvr}), the L2 controller's clock jitter, t_{jitter} , the PCB clock skew, t_{cksk} , and the propagation delay of the signal being analyzed which results in:

$$t_{su_total} = t_{cksk} + t_{su_rcvr} + t_{jitter} + t_{cksk} + t_{prop_max} \quad (EQ 2)$$

By substituting for t_{su_total} from EQ. 1, and solving for t_{prop_max} from EQ. 2, the formula for calculating the maximum allowable propagation delay is:

$$t_{prop_max} = t_{per} - t_{co_src} - t_{su_rcvr} - t_{jitter} - t_{cksk} - t_{su_margin} \quad (EQ 3)$$

This propagation delay is used to determine the maximum net length restriction for a given signal group to meet setup time specifications.

3 Hold Time Margin and Minimum Propagation Delays

The hold time margin can be defined as the driver's output hold time (t_{oh_src}), minus the receiver's input hold (t_{ih_rcvr}), plus the minimum propagation delay (t_{prop_min}). The worst case hold margin must also subtract out the clock jitter (t_{jitter}) and the clock skew (t_{cksk}), as shown:

$$t_{ho_margin} = t_{oh_src} - t_{ih_rcvr} + t_{prop_min} - t_{jitter} - t_{cksk} \quad (EQ 4)$$

Solving for the propagation delay, the formula for calculating the minimum allowable propagation delay is:

$$t_{prop_min} = t_{ih_rcvr} - t_{oh_src} + t_{ho_margin} + t_{jitter} + t_{cksk} \quad (EQ 5)$$

This propagation delay is used to determine the minimum net length restriction for a given signal group to meet hold time specifications. An important note is that the hold time margin is not directly frequency dependent. It will vary with frequency if the hold time specifications for either the driver or receiver vary with frequency.

4 Analysis and Clock Offsets

One can now plug in the numbers from the processor and memory data sheets to find the maximum and minimum propagation delays for each signal group. For the backside L2 interface the signals can be grouped into two signal groups, address and data. Since the data bus is bidirectional, timing margins for both reads and writes must be considered. However, the signal source is the processor for both address cycles and data write cycles. Since all outputs from the processor's L2 interface have the same timing specifications, this analysis will consider an address cycle to require the same propagation delays as a data write cycle. When converting from propagation delay time to length restrictions, other factors such as loading and routing topology must be considered.

So, there are two cycle types whose propagation delays must be calculated and analyzed. First is the address cycle, where the processor is the source and the memory is the receiver. Second is the data read cycle, where the memory device is the source and the processor is the receiver. For each cycle type the equations will provide a minimum and maximum propagation delay for any given frequency. This is essentially a range of allowable propagation delays. The range can be plotted on a time-line graph to compare the ranges of the different cycle types as shown in Figure 1. The data write cycle is shown for clarity, but is just the same as the address cycle range. Unless the processor and memory have the same timing specifications, the ranges will be different for the different cycle types.

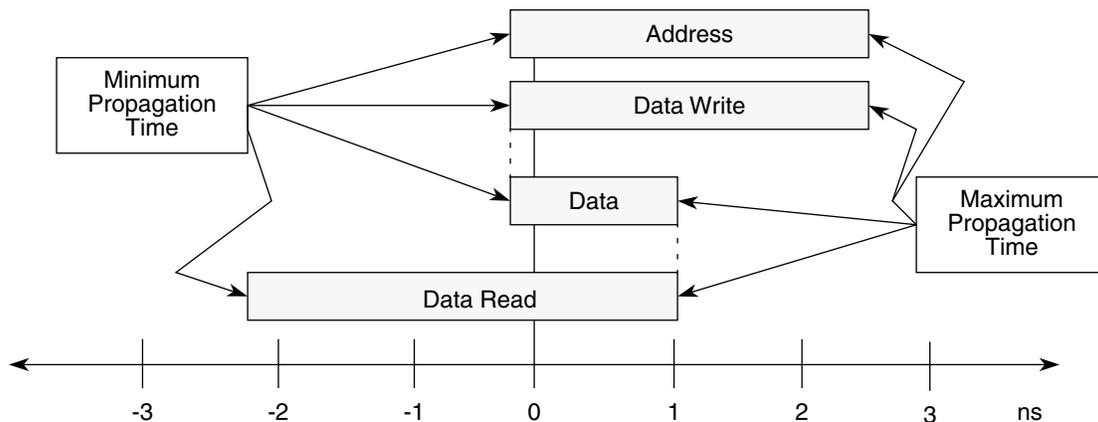


Figure 1. Propagation Delay Time-line Graph

Once the ranges are plotted, they can be analyzed to determine if there is a need to shift the ranges by using clock offsets. As shown in Figure 1, the left side of a range represents the minimum propagation delay while the right side represents the maximum propagation delay for that cycle type. Notice the data signal group has two cycle types—data write and data read. The actual range allowed for a data signal is the overlap of the data write and data read ranges.

For the initial analysis, it was assumed the L2 feedback clock and the memory clocks were aligned in time, therefore the intentional clock offset ($t_{ckoffset}$) was 0 ns. By introducing clock offsets between the L2 feedback clock and the memory clock, the ranges can be shifted to optimize the propagation delays, and thus the setup and hold margins. If additional trace is added to the L2 feedback clock, the address (and data write) range will shift to the left and the data read range will shift to the right as shown in Figure 2. If additional trace is added to the memory clock, the address (and data write) range will shift to the right and the data read range will shift to the left. The ranges will shift an amount equal to the delay introduced by the additional trace. The delay calculation is covered in the example in “Calculating the Length” on page 6.

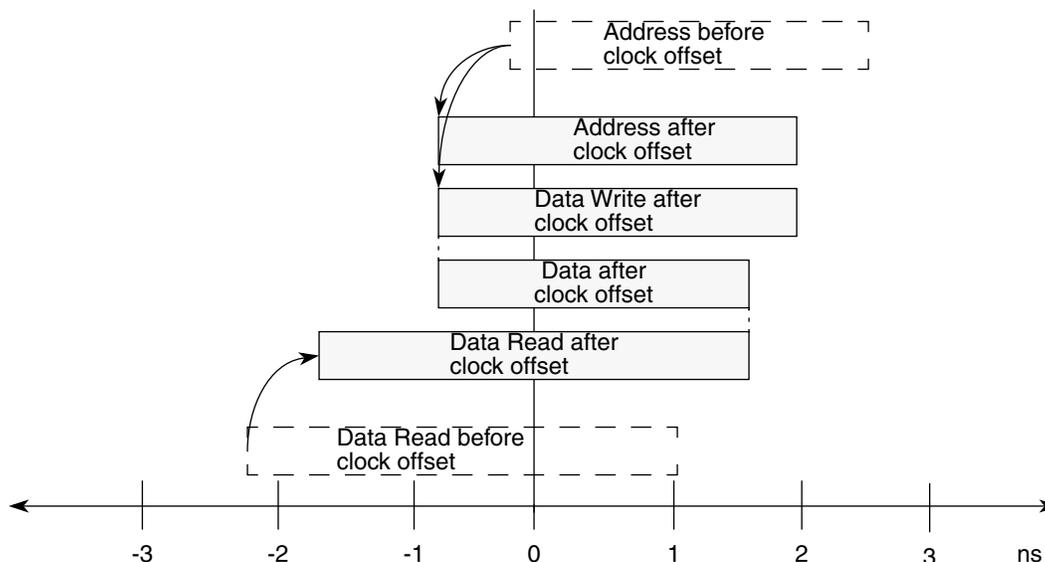


Figure 2. Propagation Delay Time-line Graph with Clock Offset

5 An Example

To illustrate the concepts presented here, an example is analyzed. This example will use a Freescale G4 processor with two Freescale MCM69P737 memories. The G4 is a next generation of processors that implement the PowerPC architecture with backside L2 support, while the MCM69P737 is a 128Kx36 bit pipelined, burst SRAM. From the data sheets, one first creates Table 1.

Table 1. Specifications from Device Data Sheets

Specification	Processor	Memory	Units
t_{co}	2.5	3.0	ns
t_{oh}	0.4	1.5	ns
t_{su}	1.5	1.2	ns
t_{ih}	0.0	0.4	ns
t_{jitter}	150	n/a	ps
load-cap	7.0	7.0	pf

For this example the PCB clock skew ($t_{ck_{sk}}$) used is 0.020 ns. Using Table 1. and the equations for the minimum and maximum propagation delay from the previous sections we create Figure 2 for three different frequencies.

Table 2. Propagation Delays with No Clock Offsets

L2 Freq (MHz)	$t_{ck_{offset}}$ (ns)	Address or Data Write		Data Read	
		t_{prop_min} (ns)	t_{prop_max} (ns)	t_{prop_min} (ns)	t_{prop_max} (ns)
150	0.000	0.170	2.797	-1.330	1.997

Table 2. Propagation Delays with No Clock Offsets (continued)

L2 Freq (MHz)	t _{ckoffset} (ns)	Address or Data Write		Data Read	
		t _{prop_min} (ns)	t _{prop_max} (ns)	t _{prop_min} (ns)	t _{prop_max} (ns)
175	0.000	0.170	1.844	-1.330	1.044
200	0.000	0.170	1.130	-1.330	0.330

In creating Figure 2, the setup and hold margins are set to 0 for this example. This implies the results represent the maximum and minimum propagation delays with 0 ns of setup and hold margin. However, any negative minimum propagation delay translates directly into hold margin, since no trace can have negative propagation delay time. The actual hold margin will be any negative minimum propagation delay from the final table, plus the trace delay of the shortest signal in a given signal group. From Figure 2 we can create the time-line for any of the frequencies. The time-line for 200 MHz is shown in Figure 3.

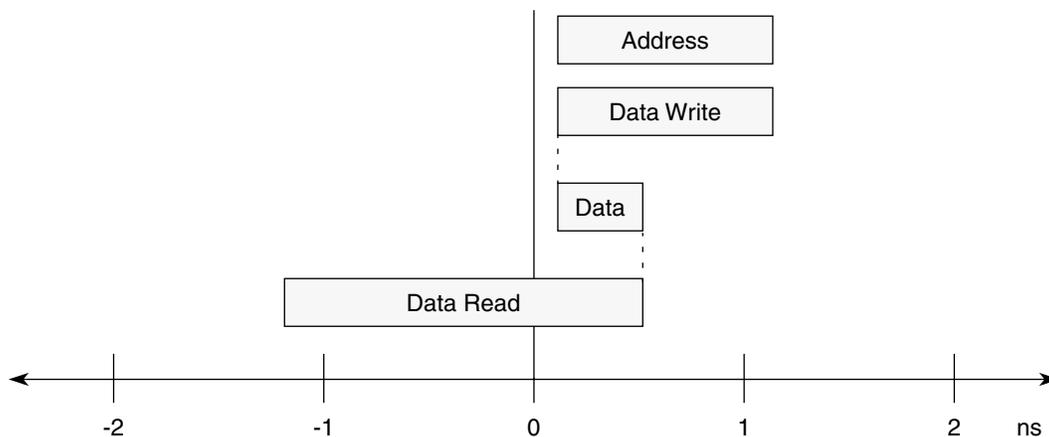


Figure 3. Propagation Delay Time-line for 200 MHz with no Clock Offset

As shown in Figure 3, the data range appears severely restricted due to the small overlap between a data write cycle and a data read cycle. By adding delay to the L2 feedback clock, we can shift the address (and data write) window left and the data read window right. How much to shift the windows depends on the relative length of the longest data trace to the longest address trace. This ratio will be dependent on package types and PCB placement. For this example, the longest data trace is approximately 75% the length of the longest address trace. Once a percentage is chosen, the clock offset to achieve this percentage is calculated with the following equation:

$$t_{ckoffset} = ((\text{percentage} * t_{prop_max}^{addr}) - t_{prop_max}^{data}) / (1 + \text{percentage}) \tag{EQ 6}$$

where the percentage is expressed as a decimal number (that is, 75% is 0.75). Using this equation and Table 2, we calculate the new clock offset. Once the clock offset is known, the equations for propagation delays (EQ 3 and EQ 5) are modified to account for the clock offset as shown below:

$$t_{prop_max} = t_{per} - t_{co_src} - t_{su_rcvr} - t_{jitter} - t_{cksk} - t_{su_margin} - t_{ckoffset} \tag{EQ 7}$$

$$t_{prop_min} = t_{ih_rcvr} - t_{oh_src} + t_{ho_margin} + t_{jitter} + t_{cksk} - t_{ckoffset} \tag{EQ 8}$$

Finally, using these propagation delay equations with clock offsets, we can create our final timing numbers as shown in Table 3.

Table 3. Propagation Delays with Clock Offsets

L2 Freq (MHz)	$t_{ckoffset}$ (ns)	Address or Data Write		Data Read	
		t_{prop_min} (ns)	t_{prop_max} (ns)	t_{prop_min} (ns)	t_{prop_max} (ns)
150	0.058	0.112	2.739	-1.272	2.054
175	0.194	-0.024	1.651	-1.136	1.238
200	0.296	-0.126	0.834	-1.034	0.626

Table 3 shows the new minimum and maximum propagation delays after accounting for the clock offset. Remember, the clock offset shown is implemented by adding the appropriate amount of trace to the L2 feedback clock. Using Table 3 we can draw the new time-line for 200 MHz as shown in Figure 4.

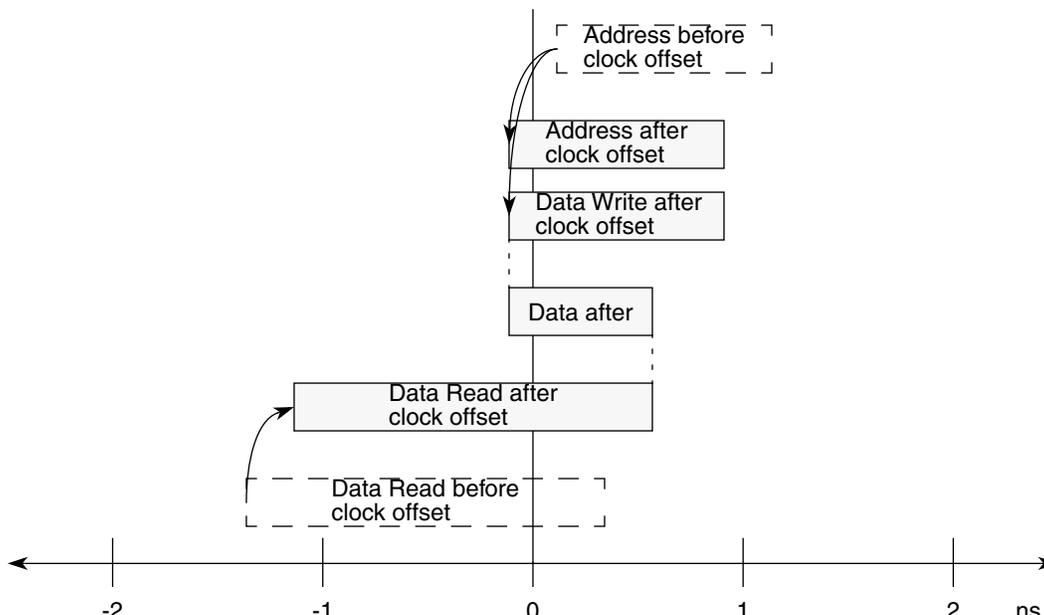


Figure 4. Propagation Delay Time-line for 200 MHz after Clock Offset

Notice the real trade-off has been a decrease in address propagation time for an increase in data propagation time and an increase in both the address and data hold time margin. For this example, the data overlap increased 2x the clock offset, while the address decreased only 1x the clock offset.

6 Calculating the Length

Until now, all units have been in time. Once the final allowable propagation delays are known, these times must be converted to length rules for PCB routing. Due to the various factors affecting the actual propagation delays on a PCB it is highly recommended that a SPICE simulator is used to convert the time delays to length restrictions. However, for those without access to such simulators or for rough estimates, equations using ‘rules of thumb’ can be used. For this example, we will use the following equation to estimate the length conversions:

$$\text{Length} = (\text{Calculated Delay} - \text{Load Delay}) / \text{PCB Propagation Speed} \tag{EQ 9}$$

This equation subtracts a load delay from the propagation delay, then divides the result by the propagation speed to convert to length. It is based on the following assumptions:

- 180 psec/inch unloaded propagation speed
- 10 psec/pf capacitive delay
- Load delay = (# of loads) * (capacitance per load (pf)) * (capacitive delay (ps/pf))

Using this equation, the following table is created from the times in Table 3.

Table 4. Propagation Delays Converted to Length (After Clock Offsets).

L2 Freq (MHz)	Clock offset (ns)	Address or Data Write		Data Read	
		I _{prop_min} (in)	I _{prop_max} (in)	I _{prop_min} (in)	I _{prop_max} (in)
150	0.058	0.0	14.439	0.0	11.024
175	0.194	0.0	8.392	0.0	6.489
200	0.296	0.0	3.857	0.0	3.087

Notice, the minimum length is set to zero if the minimum propagation delay time is negative or less than the load delay. Any negative minimum propagation delay becomes hold time margin. Once the length table is created, the PCB routing restrictions for both address and data signal groups are available. Remember when creating the length table, one must consider the loading differences. This example used two memory chips, so the address signals had two loads while data signals had only one load. Using Table 4, a PCB designed to run the L2 interface at 200 MHz, for this example, should restrict the maximum address trace length to approximately 3.8 inches and should restrict the maximum data trace length to approximately 2.7 inches. Once the PCB has been routed, the designer can compare the actual maximum lengths to the table to determine how much margin was realized. The designer can also re-evaluate whether the data-to-address length percentage was accurate and may want to adjust the clock offset to gain more margin.

Remember that many factors will affect the conversion from time delay to length restrictions, especially loading and routing topology. Capacitive loading will vary from different memory devices and manufacturers. Manufacturers are also starting to specify different loading within a single device dependent on whether the pin is an input only or an I/O pin. One would need to use the I/O capacitance for the data signals, but the input pin capacitance for the address signals. Routing topology can also have a serious impact. This example assumes signals are routed in a manner that effectively achieves incident wave switching.

7 Revision History

Table 5. Revision History

Revision Number	Changes
0.0	Initial release
0.1	Nontechnical reformatting
0.2	Nontechnical reformatting

How to Reach Us:

Home Page:

www.freescale.com

E-mail:

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 or +1-480-768-2130
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 or +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 or 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.

