MPC8245/MPC8241 Memory Clock Design Guidelines: Part 2

by *Esther C. Alexander*
CPD
Freescale Semiconductor, Inc.
Austin, TX

Contents

This application note is the second of a two-part series that explains the details of designing the memory clock interface of the MPC8245/MPC8241. Understanding and implementing the requirements for DLL locking and T_{os} (SDRAM_SYNC_IN to *sys_logic_clk*) are important to the proper functioning of the MPC8245/MPC8241. Therefore, it is important to precede this application note by reviewing the first part of the two-part series, *MPC8245/MPC8241 Memory Clock Design Guidelines: Part 1* (AN2164).

The following Freescale documents are referenced throughout this application note:

- *MPC8245/MPC8241 Memory Clock Design Guidelines: Part 1* (AN2164)
- *MPC8245 Integrated Processor User's Manual* (MPC8245UM)
- *MPC8245 Integrated Processor Hardware Specifications* (MPC8245EC)
- *MCP8241 Integrated Processor Hardware Specifications* (MPC8241EC)
- *MPC8245 Part Number Specification for the MPC8245ARZUnnnX Series* (MPC8245ARZUPNS)
- *MPC8245 Part Number Specification for the XPC8245TXXnnnx Series* (MPC8245TXXPNS)
- *MPC8245 Part Number Specification for the XPC8245TZUnnnx Series* (MPC8245TZUPNS)

freescale semiconductor

- *MPC8241 Part Number Specification for the XPC8241TXXnnnx Series* (MPC8241TXXPNS)
- *MPC8245/MPC8241 Integrated Processor Chip Errata* (MPC8245CE)

1 DLL Review

The MPC8245/MPC8241 allows for multiple clock options to accommodate various system configurations. Internally, the MPC8245/MPC8241 uses one phase-locked loop (PLL) circuit to generate master clocks to the system logic and a second PLL to generate the processor clock. The system logic PLL is synchronized to the PCI_SYNC_IN input signal.

The DLL uses the *sys_logic_clk* signal as the reference clock to synchronize the memory clock signals with the core clock. The speed of memory clock signals and *sys_logic_clk* may be set to a multiple of the PCI bus frequency as defined in the PLL tables of the MPC8245 and MCP8241 hardware specifications. The PLL_CFG[0:4] signals configure the MPC8245/MPC8241 system logic PLL at reset. To help reduce the amount of discrete logic required in a system, the MPC8245/MPC8241 provides PCI clock fanout buffers.

Figure 1 shows a block diagram of the clocking signals in the MPC8245/MPC8241.

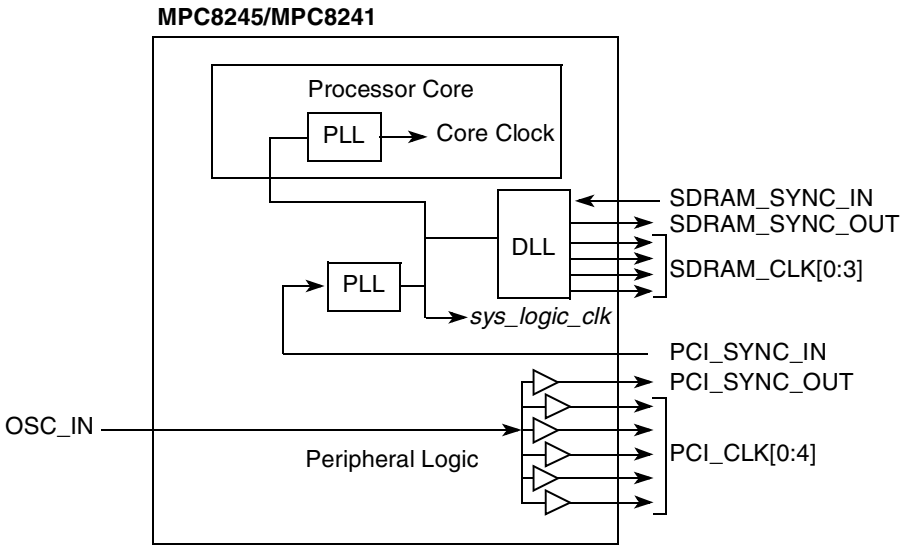


Figure 1. Clocking Signals of the MPC8245/MPC8241

DLL locking is required for proper operation of the MPC8245/MPC8241, and certain requirements must be met to ensure the DLL locks successfully. These requirements include using the design recommendations for SDRAM_SYNC_OUT to SDRAM_SYNC_IN timing and propagation delay time for the DLL to lock. There are four possible DLL locking modes. They are determined by the values of the following bits:

- The DLL_EXTEND bit of power management configuration register 2 (PMCR2[7], at offset 0x72)
- The DLL_MAX_DELAY bit of the miscellaneous I/O control register 1 (MIOCR1[2], at offset 0x76)

The register settings which define each DLL locking mode are shown in Table 1.

Table 1. DLL Mode Definition

DLL Mode	MIOCR1[DLL_MAX_DELAY]	PMCR2[DLL_EXTEND]
Normal tap delay, No DLL extend	0	0
Normal tap delay, DLL extend	0	1
Maximum tap delay, No DLL extend	1	0
Maximum tap delay, DLL extend	1	1

NOTE

These bits should be written only during system initialization and should not be altered during normal operation. Bit 5 (DLL_RESET) of the address map B options register (AMBOR[5], at offset 0xE0) controls the initial tap point of the DLL. This bit must be explicitly set and then cleared by software during initialization to guarantee correct operation of the DLL and the SDRAM_CLK[0:3] signals.

Each DLL mode has a particular region for which the DLL is guaranteed to lock. DLL locking graphs for each mode are given in the MPC8245/MPC8241 hardware specifications document so that a designer could ensure that the DLL will lock for the mode, memory bus speed and the length of SDRAM_SYNC_OUT to SDRAM_SYNC_IN (T_{loop})that will be used. For more details on determining regions of DLL locking guaranteed for memory bus speeds in each DLL mode see *MPC8245/MPC8241 Memory Clock Design Guidelines: Part 1* (AN2164).

For evaluation purposes, bits 6–0 (DLL_TAP_COUNT) of the DLL tap count register (DTCR, located at offset 0xE3) can be read to determine the value of the current tap point once the DLL has locked. The DLL mode that provides the smallest tap point value seen in DTCR should be used. This is because the bigger the tap point value, the more jitter that can be expected for clock signals. Try to keep a tap value that is at least above tap 12 since a value that is too close to the lower tap range may wrap off the edge, causing the DLL to loose lock due to drastic changes in noise, voltage, or temperature. Similarly, if the tap point is too close to the 127 point of the delay line, the DLL could also loose lock.

2 Memory Configuration and Load Impact

Several factors are involved in considering whether a memory interface design will work. It is important that the combination of trace length, loading, and drive strength (among others things) be considered when doing timing analysis. For example, the variation of drive strength will have a more visible impact in marginal designs where the difference may be as great as hundreds of picoseconds in the positive or negative marginal direction. Bits 7 to 0 of the configuration register at offset 0x73 control the drive strength of various signals of the MPC8245/MPC8241. If the memory system is heavily loaded, the drive strength will need to be increased to avoid signal rise times slowing down enough not to meet the AC timing requirements of the memory or the MPC8245/MPC8241. The stronger the drive strength, the faster the transition from low to high and high to low. Note that the choice of drive strength should be in conjunction with the termination of the trace or transmission line in order to avoid a mismatch situation. The latter could result in reflections and poor signal integrity. See Section 2.1, “Trace Length, Capacitance and Termination” for more details.

Figure 2 illustrates what the rising edge of an optimal SDRAM clock signal would look like with various drive strengths.

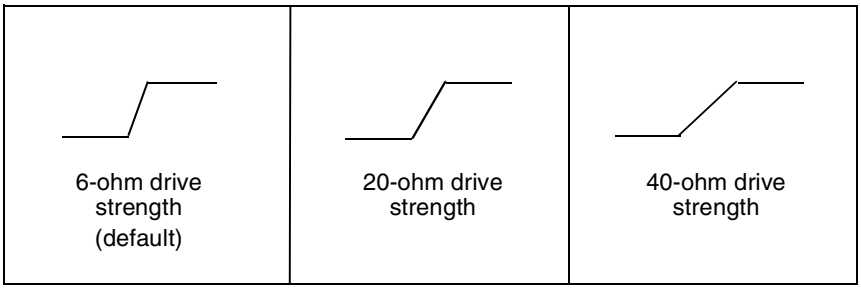


Figure 2. Drive Strength Impact

Increasing trace length of the memory signals will increase the delay due to time of flight. While modifying trace lengths and drive strengths are important factors to consider in memory interface timing, of equal importance is the capacitive load of the memory. The latter must be considered along with the specifications of the memory and the MPC8245/MPC8241. The more load that a control signal is required to support the more delay that is added to propagation time between the MCP8245 and the memory device. Not only will signal values be delayed, but they will also be increasingly degraded which contributes to poor signal integrity. All output timings of the MPC8245/MPC8241 assume a purely resistive 50-ohm load and are measured at the pin. The test load for these measurements is shown in Figure 3.

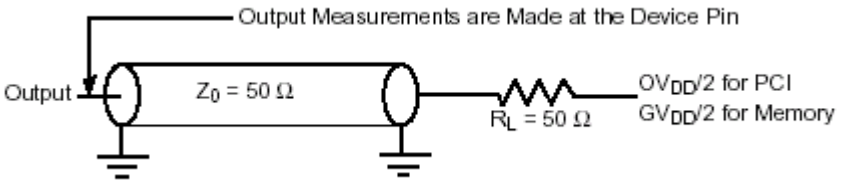


Figure 3. Test Load for the MPC8245/MPC8241.

The best method of early assessment of the impact of design factors is with the use of I/O Buffer Information Specification (IBIS) models. Both the IBIS model for the MPC8245/MPC8241 and the intended memory should be used together to evaluate what is the best layout plan for creating the desired memory bus performance capability. For more details of the use of IBIS models consult the web page of the Electronic Industries Alliance at http://www.eigroup.org. Designers may also use additional tools, available in the industry, for assessment of their memory interface design.

2.1 Trace Length, Capacitance and Termination

Capacitance per unit length increases with increase in trace width. The wider the trace length, the lower the impedance. Both the impedance and capacitance factor into determining the time of flight for transmission through a trace. The propagation delay of a trace is the product of the capacitance per unit length and the characteristic impedance of a trace.

$$\text{Delay} = C_O \times Z_O \qquad \textit{Eqn. 1}$$

where C_O is the distributed intrinsic capacitance per unit length and Z_O is the characteristic impedance of the trace.

The characteristic impedance can be expressed in terms of the inductance and capacitance of the trace, hence delay can be expressed in terms of the inductance and capacitance.

$$\text{Delay} = \sqrt{L_O \times C_O} \qquad \textit{Eqn. 2}$$

L_O and C_O are the distributed inductance and intrinsic capacitance, respectively, per unit length. Inductance is usually defined in terms of microHenrys (µH) per unit length per trace type. Capacitance is given in terms of picoFarads (pF) per unit length. Delay is given in terms of nanoseconds per unit length. This assumes that there is no loading and the main dependency is on the dielectric constant of the board material.

To take into account distributed loading, the distributed load capacitance must be included in the delay calculation. Therefore the delay with distributed load is expressed as follows:

$$\text{Delay(loaded)} = \sqrt{L_O \times (C_O + C_D)} \qquad \textit{Eqn. 3}$$

Where the additional term C_D represents the distributed load capacitance. The latter capacitance is the result of the summed capacitive load divided by the total length of trace having the load. Therefore, the values of the inductance and capacitance of the trace type and expected capacitive load should be used to determine the expected delay for traces. The time of flight of a trace is based on the delay characteristic for the kind of trace used.

An impedance mismatch between a source, load, and characteristic impedance of a transmission line results in reflection of signals across the line. Reflections can occur on the rising edge of a waveform or may occur as overshoot or undershoot. Also, the results may vary if the source impedance is dramatically mismatched from the impedance of a transmission line. The gain of the signal may be reduced going from one trace end to another. Likewise a capacitive load causes the rise time (and therefore slew rate) of a signal to slow down. Hence the quality of a signal may deteriorate significantly due to impedance mismatch or heavy capacitive load. So, termination adjustments need to be made for the trace of concern when there is potential for an impedance mismatch and signal integrity loss due to heavy capacitive load. This could be assessed with the use of simulations tools.

3 Timing Analysis

Once the DLL graphs have been considered and a range for T_{loop} is determined, the particular trace lengths used for T_{loop}, the SDRAM clocks, and the other memory interface signals must be analyzed to produce the optimal memory bus design. T_{loop} is the propagation delay of the DLL synchronization feedback loop (PC board runner) from SDRAM_SYNC_OUT to SDRAM_SYNC_IN in ns; 6.25 inches of loop length (unloaded PC board runner) corresponds to approximately 1 ns of delay for time of flight calculation. Note that Section 2, “Memory Configuration and Load Impact” includes a review of factors affecting propagation delay.

Beyond the desired memory bus speed and the range available for T_{loop}, other factors to consider are the required timing parameters for the memory, the timing parameters of the MPC8245/MPC8241, the expected load on the memory control signals, and the range of the internal offset, T_{OS}. Planned values for design parameters such as load and trace length or even frequency may change as a result of a timing analysis. Deciding on the load, for example, is based on simulation tools providing signal integrity and expected delay information.

In this section, an example of a timing analysis will be done for a 133 MHz memory bus speed design. Note that this example has some assumptions and should only be used as a reference guide for timing analysis in general. It is not the way to do a timing analysis, but will have to be modified according to the specifications of a design for which it may be used as a resource.

The timing parameters of a typical 133 MHz SDRAM are as follows:

- Memory output valid time = 5.4 nanoseconds, Memory output hold time = 3.0 nanoseconds
- Memory input setup time = 1.5 nanoseconds, Memory input hold = 0.8 nanoseconds

The delay on the control lines due to load is based on the memory module used. For the timing analysis some assumptions will be made. The timing parameters of the MPC8245/MPC8241 that must be considered are as follows:

- Output Valid time = 4.0 nanoseconds, Output Hold time = 1.0 nanoseconds
  (All output timings assume a purely resistive 50-ohm load and are measured at the pin.)
- T_{OS} range = 0.4 to 1.0 nanoseconds
- Clock Period = 7.5 nanoseconds (for 133 MHz bus speed)
- Clock to clock jitter = 190 picoseconds

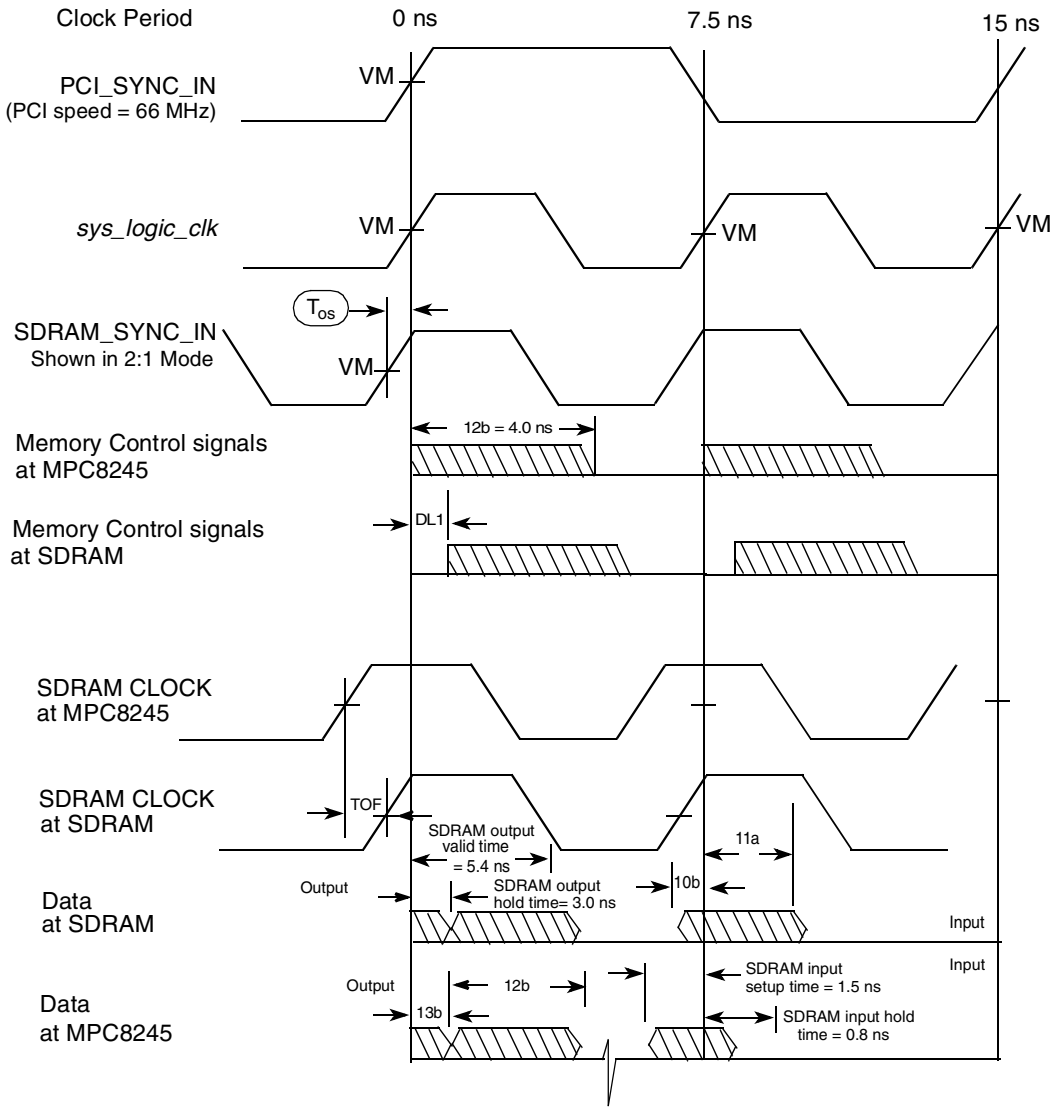
Table 2 shows variable input setup and hold time based on the bit value of bits 5 and 4 of register at offset 0x77 (during initialization).

Table 2. MPC8245/MPC8241 Programmable Input Timing

Bit 5 of 0x77	Bit 4 of 0x77	Input Setup Time (nanoseconds)	Input Hold time (nanoseconds)
0	0	2.6	0.0
0	1	1.9	0.7
1	0	1.2	1.4
1	1	0.5	2.1

T_{OS} is an internal delay in the feedback path for SDRAM_SYNC_IN with respect to the internal *sys_logic_clk* signal of the MPC8245/MPC8241. The DLL tries to compensate for this delay when it compares the clock of SDRAM_SYNC_IN to that of *sys_logic_clk*. The result is that the adjustment causes the DLL to try to lock earlier than it would without the delay. This causes SDRAM_SYNC_IN to be seen earlier than *sys_logic_clk* when the two signals are observed after the DLL locks. Based on characterization, the accommodation for the delay ranges from 0.4 to 1.0 ns. The feedback trace length of SDRAM_SYNC_OUT to SDRAM_SYNC_IN should be shortened by 0.7 ns so that the impact of T_{os} can be reduced. This is recommended because 0.7 ns is the midpoint of the range of T_{os}.

Figure 4 illustrates the timing reference information for the memory bus interface between the MPC8245/MPC8241 and a 133 MHz SDRAM device.



Notes:
VM = Midpoint voltage (1.4 V).
10b = (Variable) Input setup time of the MPC8245.
11a = (Variable) Input hold time of the MPC8245.
12b = *O*utput valid timing of the MPC8245.
13b = Output hold time of the MPC8245.
DL1 = delay due to time of flight of trace and capacitive load of memory.
T_{os} = Offset timing required to align *sys_logic_clk* with SDRAM_SYNC_IN. The SDRAM_SYNC_IN signal is adjusted by the DLL to accommodate for internal delay. This causes SDRAM_SYNC_IN to be seen before *sys_logic_clk* once the DLL locks.
TOF = Trace time of flight.
There is no accounting for T_{loop} in this figure as the timing analyses to follow have T_{loop} = 0 ns.

Figure 4. Input/Output Timing Diagram for 133 MHz

The impact on delay due to load must be considered particularly for the control signals. A worse case analysis should take into account the connections from the pins of the memory device to the control signals of the MPC8245/MPC8241. For the analysis the assumptions for load delay on the longest control line that will impact meeting the memory's input setup time will be presented for two cases.

The following directions of transaction requirements must be considered for a proper timing analysis:

1. (The MPC8245/MPC8241) Meeting the memory's input setup time
2. (The MPC8245/MPC8241) Meeting the memory's input hold time
3. (The memory) Meeting the MPC8245/MPC8241's input setup time
4. (The memory) Meeting the MPC8245/MPC8241's input hold time

Each of the above items will be considered separately for the given timing parameters. Note that propagation delay due to time of flight of an unloaded trace for the analyses is one nanosecond for 6.25 inches.

Suppose the desired length for T_{loop} and the memory trace lengths (including clocks, data, and control signals) is about 0.32 nanoseconds (2 inches). Note that based on the MPC8245/MPC8241 hardware specifications document T_{loop} needs to be at least 0.7 nanoseconds (4.375 inches) less than the SDRAM clock trace lengths. Therefore, if the desired trace length for the SDRAM clocks is two inches (MPC8245/MPC8241 pin to SDRAM pin), then T_{loop} in that case cannot be more than zero inches. Furthermore, if T_{loop} is zero, the trace length for the SDRAM clocks needs to be about 4.4 inches to accommodate the full range of T_{OS}.

3.1 Analysis 1

Assume a lightly loaded memory interface so that there is a 0.7 ns delay impact on meeting input setup time of memory and 0.3 ns delay impact on memory input hold time for the longest control line. Table 3 shows the analysis for meeting the SDRAM's input timing requirements.

Table 3. Timing Analysis from the MPC8245/MPC8241 to the Memory Device

(The MPC8245/MPC8241) Meeting the memory's input setup time		
Parameter	**Trace (inches)**	**Delay (nanoseconds)**
Clock period	—	+7.500
Memory clock trace length	4.4	+0.704
Clock jitter/skew	—	-0.190
MPC8245/MPC8241 output valid time	—	-4.000
Control signal maximum trace length	1.9[1]	-0.304
SDRAM_SYNC_OUT to SDRAM_SYNC_IN (T_{loop})	0	-0.000
Worse case T_{OS}	—	-1.000
Load impact on setup time for control lines	—	-0.700
Total time	—	+2.01
Memory input setup time	—	1.500
Margin (Total time – input setup time)	—	**+0.510**

Table 3. Timing Analysis from the MPC8245/MPC8241 to the Memory Device (continued)

(The MPC8245/MPC8241) Meeting the memory's input hold time		
Parameter	**Trace (inches)**	**Delay (nanoseconds)**
Memory clock trace length	4.4	-0.704
Clock jitter/skew	—	-0.190
MPC8245/MPC8241 Output hold time	—	+1.000
Control signal minimum trace length	1.9 [1]	+0.304
SDRAM_SYNC_OUT to SDRAM_SYNC_IN (T_{loop})	0	+0.000
Worse case T_{OS}	—	+0.400
Load impact on hold time for control lines	—	+0.300
Total time	—	+1.110
Memory input hold time	—	0.800
Margin (Total time – input hold time)	—	**+0.310**

[1] Note that 1.9 inches is used based on the assumption that control trace lengths are equal. For control signals with various trace lengths, the minimum control trace length should be used for this analysis.

Table 3 shows that the parameters chosen will provide at least 300 picoseconds of margin to meet the SDRAM's input timing requirements. When considering the transactions from the memory to the MPC8245/MPC8241, the load on the control lines do not factor into the analysis. Table 4 shows the analysis for meeting the MPC8245/MPC8241's input timing requirements.

The ability to use various input setup and hold times on the MPC8245/MPC8241 allows for timing analyses to consider the various options. From Table 4 it will be clear that depending on the tap delay option chosen, there may or may not be any margin for transactions originating from the memory to the MPC8245/MPC8241.

Table 4. Timing Analysis from the Memory Device to the MPC8245/MPC8241

(The MPC8245/MPC8241) Meeting the 8245/8241's input setup time				
Parameter	**Trace (inches)**		**Delay (ns)**	
Clock period	—		+7.500	
Memory clock trace length	4.4		- 0.704	
Clock jitter/skew	—		-0.190	
Memory output valid time	—		-5.400	
Data signal maximum trace length	1.9[1]		-0.304	
SDRAM_SYNC_OUT to SDRAM_SYNC_IN (T_{loop})	0		+0.000	
Worse case T_{OS}	—		+0.400	
Total time	—		+1.302	
Programmable input timing option	Tap delay 0 (Bits 5:4 of 0x77=00)	Tap delay 1 (Bits 5:4 of 0x77=01)	Tap delay 2 (Bits 5:4 of 0x77=10)	Tap delay 3 (Bits 5:4 of 0x77=11)
MPC8245/MPC8241 input setup time	2.600	1.9	1.2	0.5
Margin (total – input setup time)	**-1.298**	**-0.598**	**+0.102**	**+0.802**
(The memory) Meeting the MPC8245/MPC8241's input hold time				
Parameter	**Trace (inches)**		**Delay (ns)**	
Memory clock trace length	4.4		+0.704	
Clock jitter/skew	—		-0.190	
Memory output hold time	—		+3.000	
Data signal minimum trace length	1.9 [1]		+0.304	
SDRAM_SYNC_OUT to SDRAM_SYNC_IN (T_{loop})	0		+0.000	
Worse case T_{OS}	—		-1.000	
Total time	—		+2.818	
Programmable input timing option	Tap delay 0 (Bits 5:4 of 0x77=00)	Tap delay 1 (Bits 5:4 of 0x77=01)	Tap delay 2 (Bits 5:4 of 0x77=10)	Tap delay 3 (Bits 5:4 of 0x77=11)
MPC8245/MPC8241 input hold time	0.000	0.700	1.400	2.100
Margin (total – input hold time)	**+2.818**	**+2.118**	**+1.418**	**+0.718**

[1] Note that 1.9 inches is used based on the assumption that control and data trace lengths are equal. For data signals with various trace lengths, the minimum data trace length should be used for this analysis.

The options that seem to work are tap delay 2 and 3 which involve setting bits 5 and 4 of the configuration register at offset 0x77 to 10 or 11, for these respective tap delays. Section 3.2, "Analysis 2," will illustrate how load delay and clock trace length can impact timing margins.

3.2 Analysis 2

This analysis is different from Section 3.1, "Analysis 1," in that the load is heavier and the memory clock trace length is longer. Assume a heavily loaded memory interface (1.7 ns delay impact on meeting input setup time of memory and 0.5 ns delay impact on memory input hold time) for the longest control line. Table 5 shows the analysis for meeting the SDRAM's input timing requirements.

Table 5. Timing Analysis from the MPC8245/MPC8241 to the Memory Device

(The MPC8245/MPC8241) Meeting the memory's input setup time		
Parameter	**Trace (inches)**	**Delay (nanoseconds)**
Clock period	—	+7.500
Memory clock trace length	7.5 [1]	+1.200
Clock jitter/skew	—	-0.190
MPC8245/MPC8241 output valid time	—	-4.000
Control signal maximum trace length	1.9[2]	-0.304
SDRAM_SYNC_OUT to SDRAM_SYNC_IN (T_{loop})	0	-0.000
Worse case T_{OS}	—	-1.000
Load impact on setup time for control lines	—	-1.700
Total time	—	+1.506
Memory input setup time	—	1.500
Margin (Total time – input setup time)	—	**+0.006**
(The MPC8245/MPC8241) Meeting the memory's input hold time		
Parameter	**Trace (inches)**	**Delay (nanoseconds)**
Memory clock trace length	7.5[1]	-1.200
Clock jitter/skew	—	-0.190
MPC8245/MPC8241 output hold time	—	+1.000
Control signal maximum trace length	1.9 [2]	+0.304
SDRAM_SYNC_OUT to SDRAM_SYNC_IN (T_{loop})	0	+0.000
Worse case T_{OS}	—	+0.400
Load impact on hold time for control lines	—	+0.500
Total time	—	+0.814
Memory input hold time	—	0.800
Margin (Total time – input hold time)	—	**+0.014**

[1] With long trace lengths there should be concern for decreased slew rates and increased delay.

[2] Note that 1.9 inches is used based on the assumption that control trace lengths are equal. For control signals with various trace lengths, the minimum control trace length should be used for this analysis.

From Table 5, a heavy capacitive load on the control lines, caused reduced margins for transactions going from the MPC8245/MPC8241 to the memory. To accommodate for this, the memory clock trace length had to be increased. This in turn caused less timing margin for transactions going from the memory to the MPC8245/MPC8241. As before, for the transactions from the memory to the MPC8245/MPC8241, the load on the control lines do not factor into the analysis.

Table 6. Timing Analysis from the Memory Device to the MPC8245/MPC8241

(The Memory) Meeting the 8245/8241's input setup time				
Parameter	**Trace (inches)**		**Delay (ns)**	
Clock period	—		+7.500	
Memory clock trace length	7.5		-1.200	
Clock jitter/skew	—		-0.190	
Memory output valid time	—		-5.400	
Data signal maximum trace length	1.9 [1]		-0.304	
SDRAM_SYNC_OUT to SDRAM_SYNC_IN (T_{loop})	0		+0.000	
Worse case T_{OS}	—		+0.400	
Total time	—		+0.806	
Programmable input timing option	Tap delay 0 (Bits 5:4 of 0x77=00)	Tap delay 1 (Bits 5:4 of 0x77=01)	Tap delay 2 (Bits 5:4 of 0x77=10)	Tap delay 3 (Bits 5:4 of 0x77=11)
MPC8245/MPC8241 input setup time	2.600	1.9	1.2	0.5
Margin (total – input setup time)	**-1.794**	**-1.094**	**-0.394**	**+0.306**

Table 6. Timing Analysis from the Memory Device to the MPC8245/MPC8241 (continued)

(The memory) Meeting the MPC8245/MPC8241's input hold time				
Parameter	**Trace (inches)**		**Delay (ns)**	
Memory clock trace length	7.5		+1.200	
Clock jitter/skew	—		-0.190	
Memory output hold time	—		+3.000	
Data signal minimum trace length	1.9[1]		+0.304	
SDRAM_SYNC_OUT to SDRAM_SYNC_IN (T_{loop})	0		+0.000	
Worse case T_{OS}	—		-1.000	
Total time	—		+3.314	
Programmable input timing option	Tap delay 0 (Bits 5:4 of 0x77=00)	Tap delay 1 (Bits 5:4 of 0x77=01)	Tap delay 2 (Bits 5:4 of 0x77=10)	Tap delay 3 (Bits 5:4 of 0x77=11)
MPC8245/MPC8241 input hold time	0.000	0.700	1.400	2.100
Margin (total – input hold time)	**+3.314**	**+2.614**	**+1.914**	**+1.214**

1 Note that 1.9 inches is used based on the assumption that control and data trace lengths are equal. For data signals with various trace lengths, the minimum data trace length should be used for this analysis.

From the four available input setup and hold options, only Tap Delay 3, where bits 5 and 4 of 0x77 are set, was able to meet the margin deficit produced due to the increased memory trace length.

4 IBIS Simulations

The use of IBIS models can prove to be a valuable tool in memory interface design. The availability of several design options, which can be considered in an analysis, allows the designer to assess what is the best option for the project. Dependencies include trace lengths, loads, design topology, and the number of memory modules. The results of analyses provide an idea of the signal integrity expected for a configuration and an estimate of the expected propagation delay due to load and time of flight for signals.

This section illustrates how IBIS simulations can be used to compare routing topologies. The topology types which will be considered for this section are the Daisy Chain (bus) and the Tee routing methods. Both use small outline dual inline memory modules (SO-DIMMs). Note that the lengths chosen in the two cases are not based on the timing analysis in Section 3, "Timing Analysis."

Figure 5 and Figure 6 show how the two topologies/configurations being considered are routed.

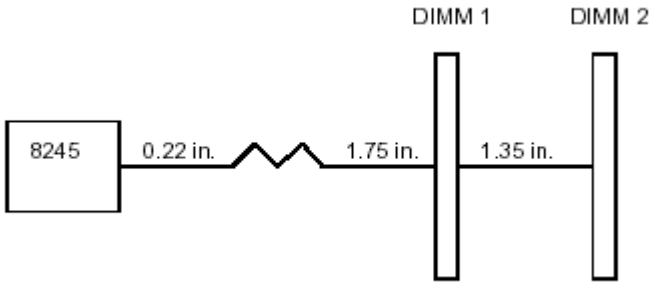


Figure 5. Daisy Chain (Bus) Configuration

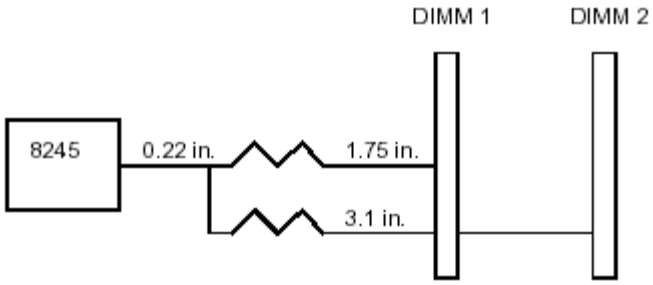


Figure 6. Tee Configuration

IBIS simulations of the two configurations were done with variations including the values the resistor(s) and the load on each SO-DIMM. The listed trace length represents the average length of traces going from the MPC8245/MPC8241 to the SO-DIMMs. A control signal will be used to show the impact of loading on propagation delay. The IBIS models used are for the MPC8245/MPC8241 and 133 MHz memory. The control signal of choice for this analysis is column address strobe (CAS).

Measurements for the delay are considered on the trace end close to the memory on the SO-DIMM. It is important to determine at what reference point of the waveform that the delay should be measured from CAS to the SDRAM clock signal. Some memory devices call for a midpoint-to-midpoint measurement while others go further to consider the slew rate of the control signal being assessed. For this case, the delay measurement is from the 0.8 volt point on the falling edge of CAS to the 2.0 volt level on the SDRAM clock because of the slow slew rate on the measured CAS signal.

Figure 7 is an example of the IBIS waveform output for the CAS signal at the two SO-DIMM locations (DIMM1 and DIMM2) for the Daisy Chain configuration shown in Figure 5 with a 10-ohm series resistor and 20-ohm drive strength.

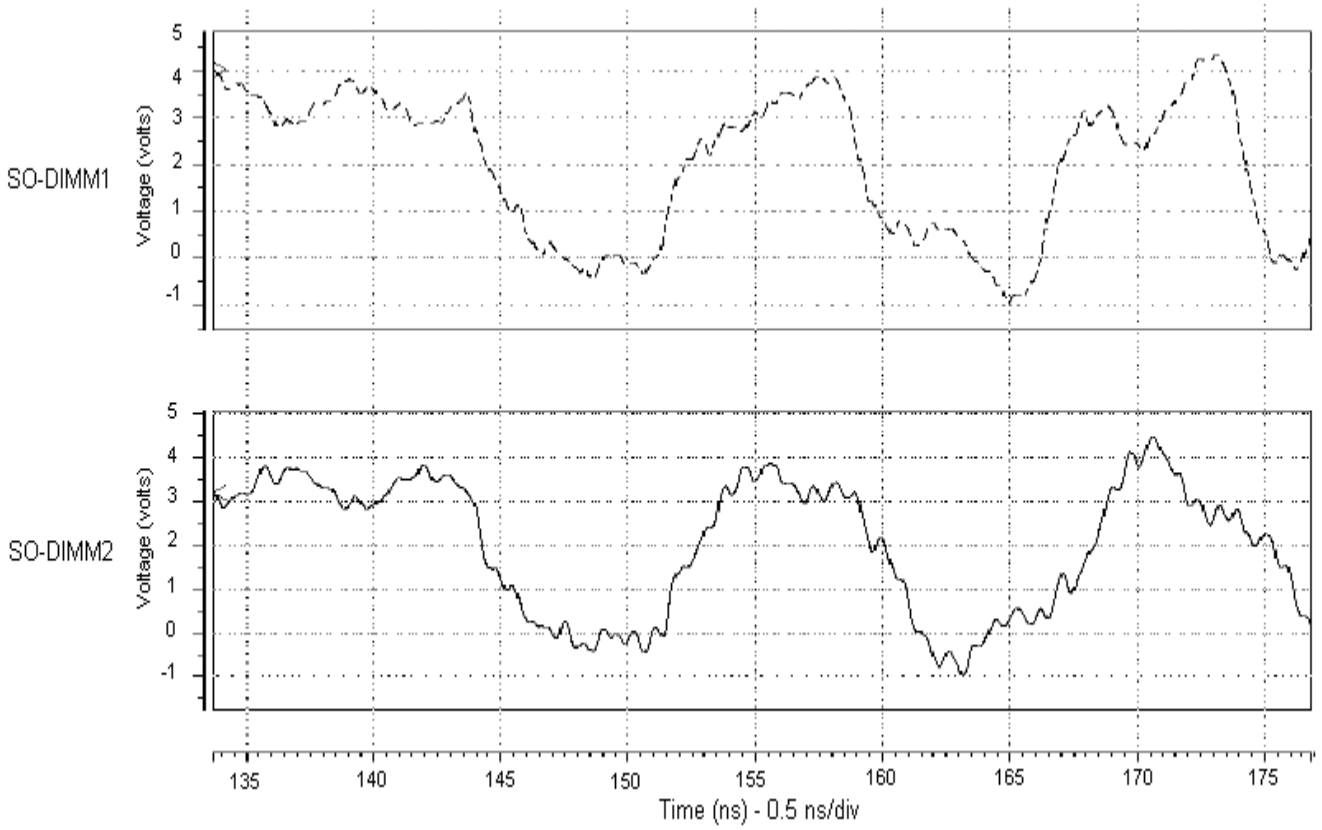


Figure 7. CAS Seen at SO-DIMMs for Daisy Chain Configuration

The placement of the SO-DIMM affects how clean the waveform for the signal is and likewise affects the delay. This is seen in Table 7 which shows loads at various SO-DIMM locations and the measured delay for the IBIS simulation. Note that the drive strength for this analysis was 6 ohms. In general the lower the resistor value of the series termination, the less the delay.

Table 7. Route Propagation Delay for Daisy Chain

Route Configuration	Loads on SO-DIMM #1	Loads on SO-DIMM #2	Termination	Average Delay (ns) [1]
Daisy Chain	9	0	Series 10	2.3
Daisy Chain	9	0	Series 15	2.4
Daisy Chain	9	0	Series 22	2.4
Daisy Chain	0	9	Series 10	2.6
Daisy Chain	0	9	Series 15	2.8
Daisy Chain	0	9	Series 22	2.9
Daisy Chain	9	9	Series 10	3.3
Daisy Chain	9	9	Series 15	3.6
Daisy Chain	9	9	Series 22	4.4

[1] The average delay includes the trace length and the delay due to loading. In the case where the delay is shown for two SO-DIMMs, the average delay was calculated for the total delay at the further SO-DIMM.

In the Daisy Chain configuration the waveforms are subjected to more reflection than the waveforms in the Tee configuration. This is seen when comparing waveforms of Figure 7 and Figure 8 where CAS observed by IBIS simulation for both the Daisy Chain and Tee configurations.

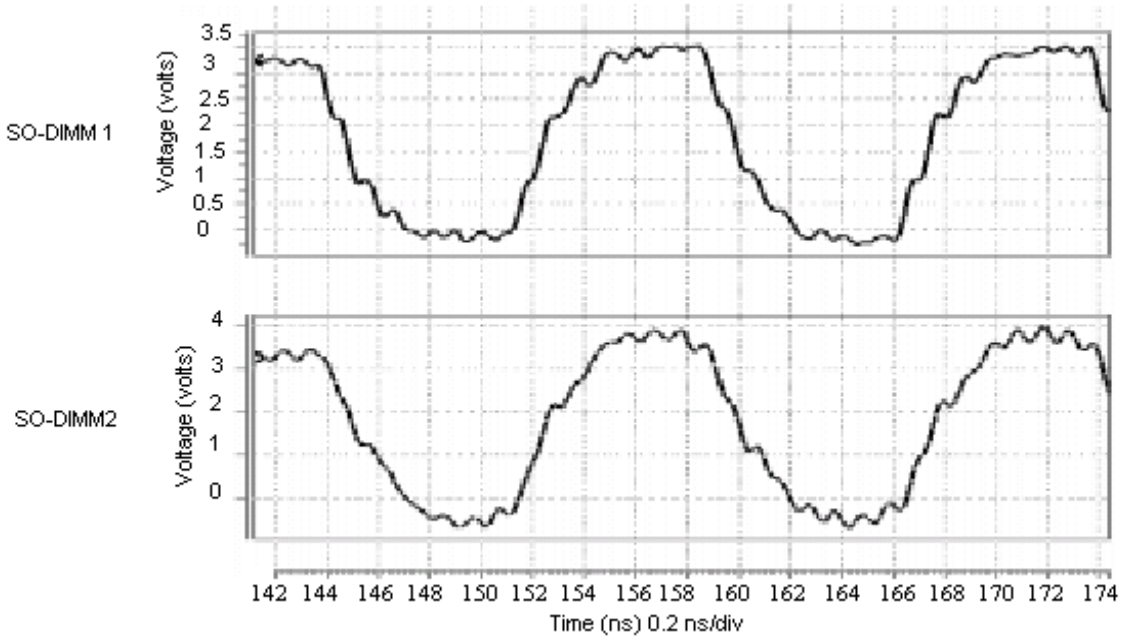


Figure 8. CAS Seen at SO-DIMMs with Tee Configuration

The memory load has less delay impact in the Tee configuration than with the Daisy Chain configuration. This is seen when the observed results were compared in the last two columns of Table 8. The Tee configuration (in Figure 6) helped reduce the delay due to load. The drive strength used for the analysis was 6 ohms. Therefore for the amount of load that is involved for the SO-DIMMs used, the better design between the two options is the Tee configuration.

Table 8. Route Propagation Delay for Tee Configuration

Route Configuration	Loads on SO-DIMM #1	Loads SO-DIMM #2	Termination	Average Tee Delay (ns)	Daisy Chain delay average (ns) [1]
Tee	9	0	Series 10	2.20	2.3
Tee	9	0	Series 22	2.35	2.4
Tee	0	9	Series 10	2.35	2.6
Tee	0	9	Series 22	2.35	2.9
Tee	9	9	Series 10	2.75	3.3
Tee	9	9	Series 22	3.05	4.4

[1] The average delay includes the trace length and the delay due to loading. In the case where the delay is shown for two SO-DIMMs, the average delay was calculated for the total delay at the further SO-DIMM.

This comparison clearly shows that there is some benefit to analyzing design configurations before a final decision is made on the routing of the memory interface.

5 Harmonic Noise: A Case Study

Memory interface designs should also take into consideration the effects of harmonic noise for applications that are required to meet IEEE 802.11 standards. These standards dictate the requirements for wireless ethernet transmission and have to be met by wireless local area network access protocols (WLAN AP). Beyond that, ethernet applications must also conform to the requirements of the Federal Communications Commission (FCC) in order to be licensed for use. Several frequency bands have restrictions on their use and as such, some attention must be paid to how components for a radio frequency design conform to these restrictions. Harmonic noise must be reduced for the frequency bands which apply to a particular radio frequency application.

On the memory interface of the MPC8245/MPC8241 it has been observed that in a lightly loaded memory system such as a WLAN AP, the default memory clock output drive strength of 6 ohms creates harmonic noise around the memory bus interface. The harmonics are a multiple of the memory clock frequency and go well into the 18th and 19th harmonics (in 802.11g frequency band; 18th H = 2.376 GHz, 19th H = 2.508 GHz). The gain, measured in decibels (dB), was seen to change relative to drive strength. Therefore lightly loaded designs that involve the MPC8245/MPC8241 and a need to meet IEEE 802.11 b and g standards, should try to reduce this noise.

Figure 9 shows the overall detection of harmonic noise for the 500 MHz to 2.5 GHz range by placing an antenna from a spectrometer a few centimeters over the MPC8245/MPC8241.

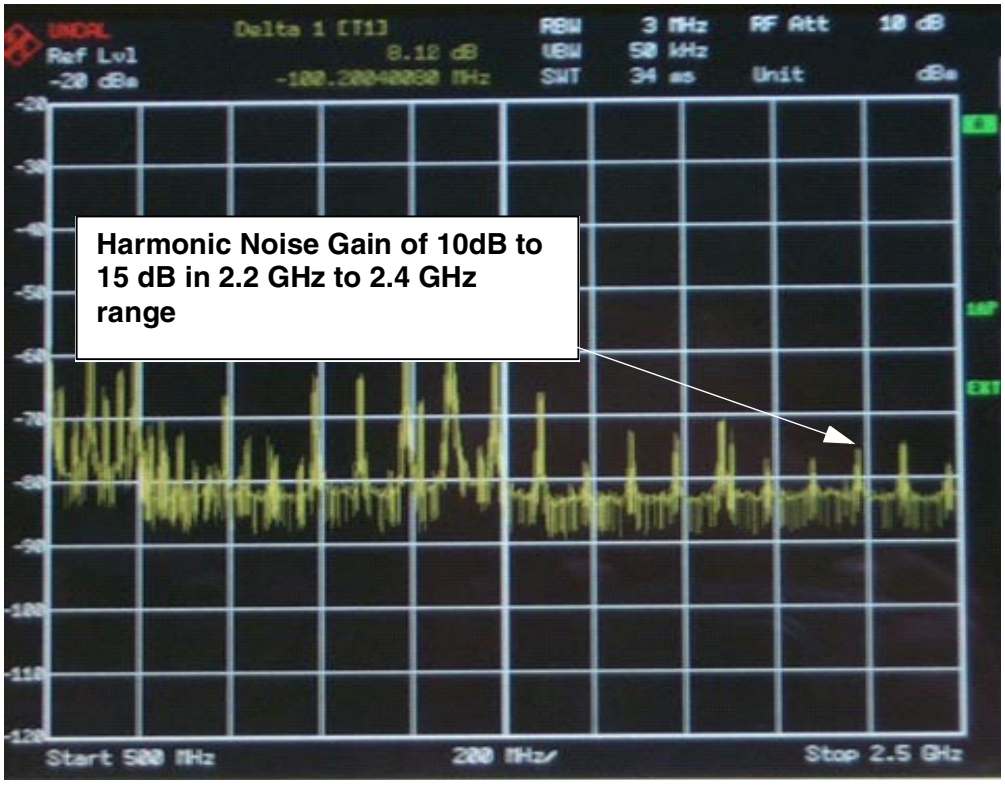


Figure 9. Harmonic Noise for 500 MHz–2.5 GHz at 100 MHz Memory Bus Speed

Note that references to Figure 9 through Figure 12 focus on the amplitude of harmonic noise and as such, the lack of clarity on the axes of these figures, should be ignored.

The amplitude of the harmonic noise is directly related to the drive strength used in conjunction with lightly loaded memory systems. The gain of harmonic noise seen in the 2.2 to 2.4 GHz range is 10 dB to 15 dB when using the 6-ohm memory clock driver. This is the region that IEEE specification 802.11g covers.

The noise is also more pronounced with increase in memory bus activity. One solution that has been verified to work is changing the memory clock drive strength by using software to change bits 0 and 1 of the configuration register at offset 0x73. This register is used to program the drive strength of various signals. The most pronounced reduction occurs when the 40-ohm drive strength is used to control the SDRAM clocks. This drastic difference in harmonic noise level going from a 6-ohm drive strength to 40-ohm drive strength for a 100 MHz memory bus speed is illustrated by comparing Figure 10 to Figure 11.

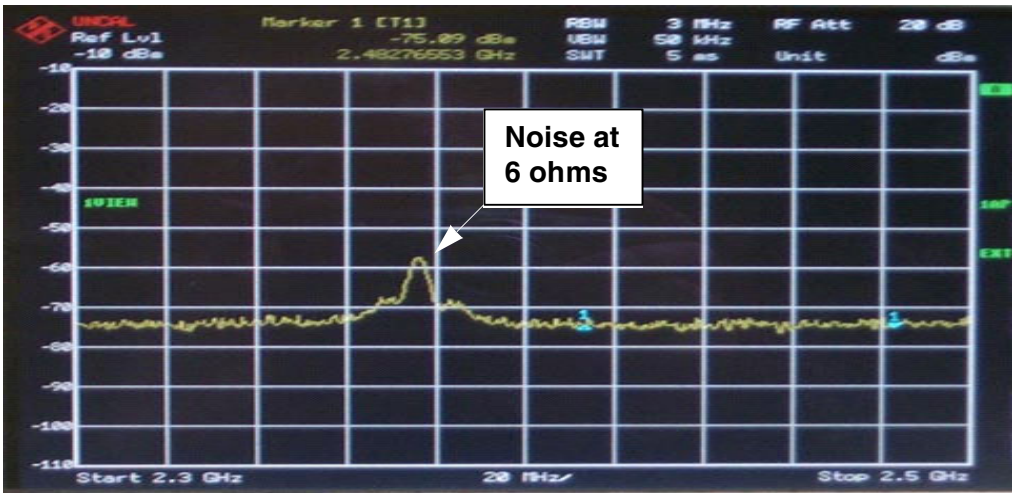


Figure 10. Noise Seen for 2.3–2.5 GHz with 6-Ohm Memory Clock Driver at 100 MHz Memory Bus Speed

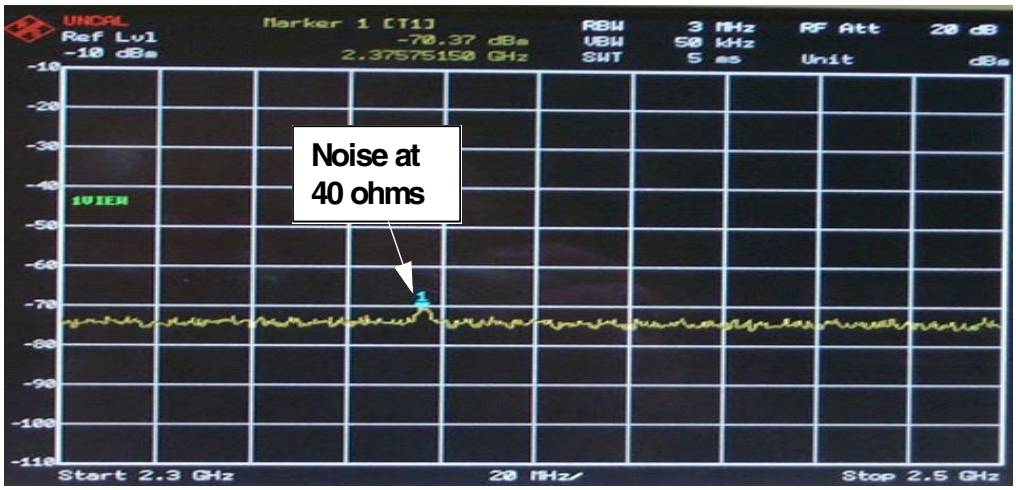


Figure 11. Noise Reduced for 2.3–2.5 GHz with 40-Ohm Memory Clock Driver at 100 MHz Memory Bus Speed

In the case of a 133 MHz memory bus speed, a drastic reduction in gain happens just going from a 6-ohm to 20-ohm drive strength and at 40 ohms the gain is negligible. This progression is shown in Figure 12.

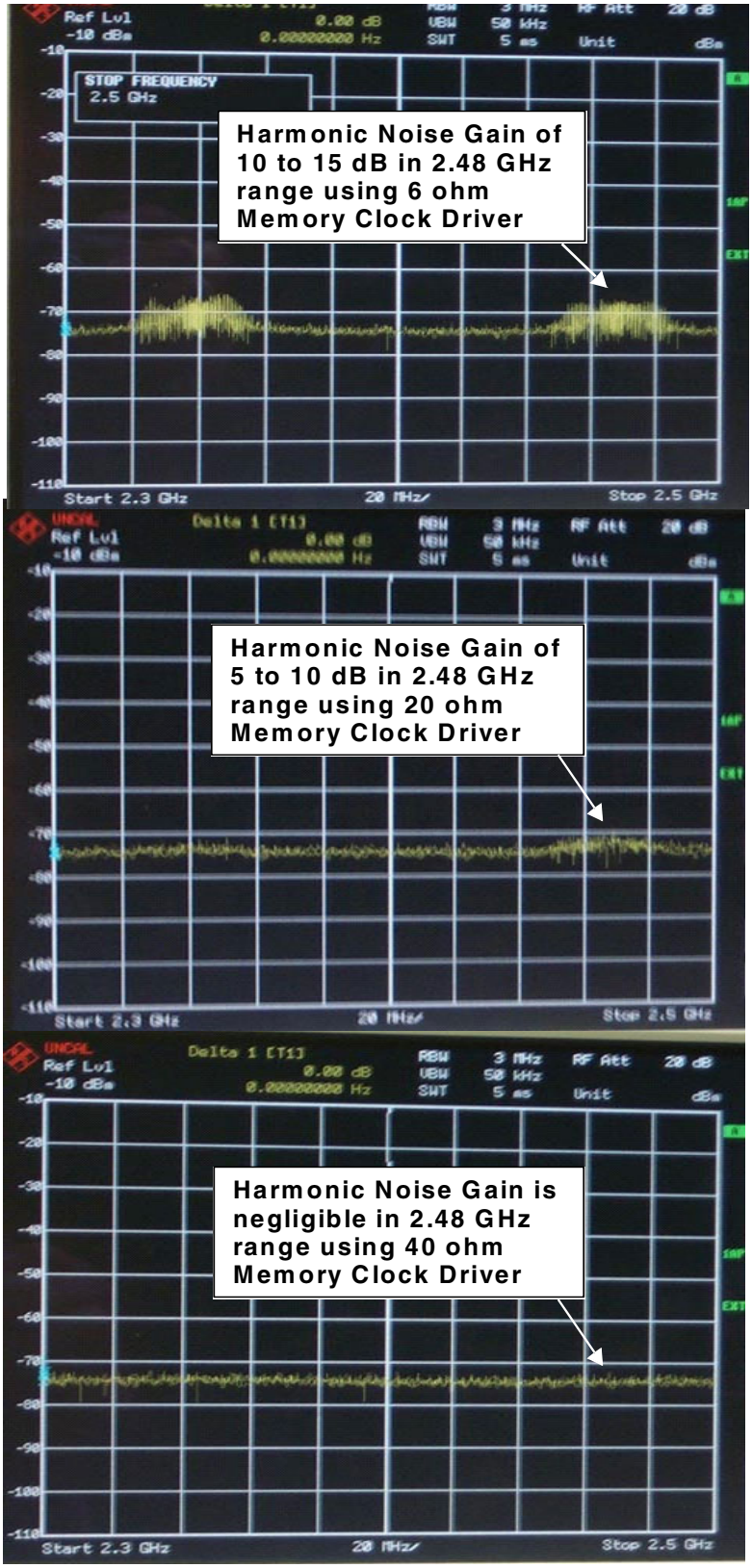


Figure 12. Drive Strength Impact on Harmonic Noise for 133 MHz

Beyond the drive strength solution, another option is to increase the capacitive load on the memory clock trace. The downside of this solution is that the signal integrity may be compromised with the reduction in drive strength and memory timing margins will be affected by the increased delay that both methods will create. The use of IBIS simulations for a design will help to indicate the expected delay for a design with change in capacitive load.

6 Conclusion

Designing memory bus interfaces involve deciding on the expected trace lengths, loads, and routing topology. The planning stages of these decisions can be aided with the use of timing analysis, and IBIS simulations. Designers should be concerned about timing delay and signal integrity to ensure proper memory bus functionality. In addition, there are tools available from memory vendors to help determine expected memory performance. Beyond creating a design that is optimal for the desired trace lengths and loads, products which must meet the IEEE 802.11g standard, should take into consideration adjustments to reduce the amplitude of harmonic noise which may be seen in lightly loaded systems.

7 References

[1] “Transmission Line Theory,” Freescale Semiconductor Design Guide, Section 2

[2] “Tsi107 PowerPC Host Bridge Design Guide,” Tundra Semiconductor Corporation, October 2003

[3] Chris Hanke, Gary Tharaison, “Low Skew Clock Drivers and Their System Considerations,” Freescale Semiconductor, Application Note AN1091

8 Revision History

Table 9 provides a revision history for this application note.

Table 9. Document Revision History

Revision Number	Date	Substantive Change(s)
0	08/31/04	Initial release

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
(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.
Technical Information Center
3-20-1, Minami-Azabu, Minato-ku
Tokyo 106-0047 Japan
0120 191014
+81 3 3440 3569
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
(800) 441-2447
303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor@
hibbertgroup.com

AN2746
Rev. 0
08/2004

freescale™
semiconductor