



20 YEARS *of*  
**ALTERA**®  
INNOVATION

## SDRAM PLL Tuning

# SDRAM PLL Tuning

- Objective: Find the Correct Skew Needed for the SDRAM Clock with Respect to the System Clock
- Two Methods:
  - “Scientific” vs. “Trial and Error”

# Design Recommendations

- Use “Zero Delay Buffer” Mode of the PLL
  - Allows Control of Phase of External Clock with Respect to the Input Clock
- SDRAM Clock Output of PLL (e0)
- SDRAM Pins:
  - Use Fast I/O's Settings
  - Group Together to Minimize Skew

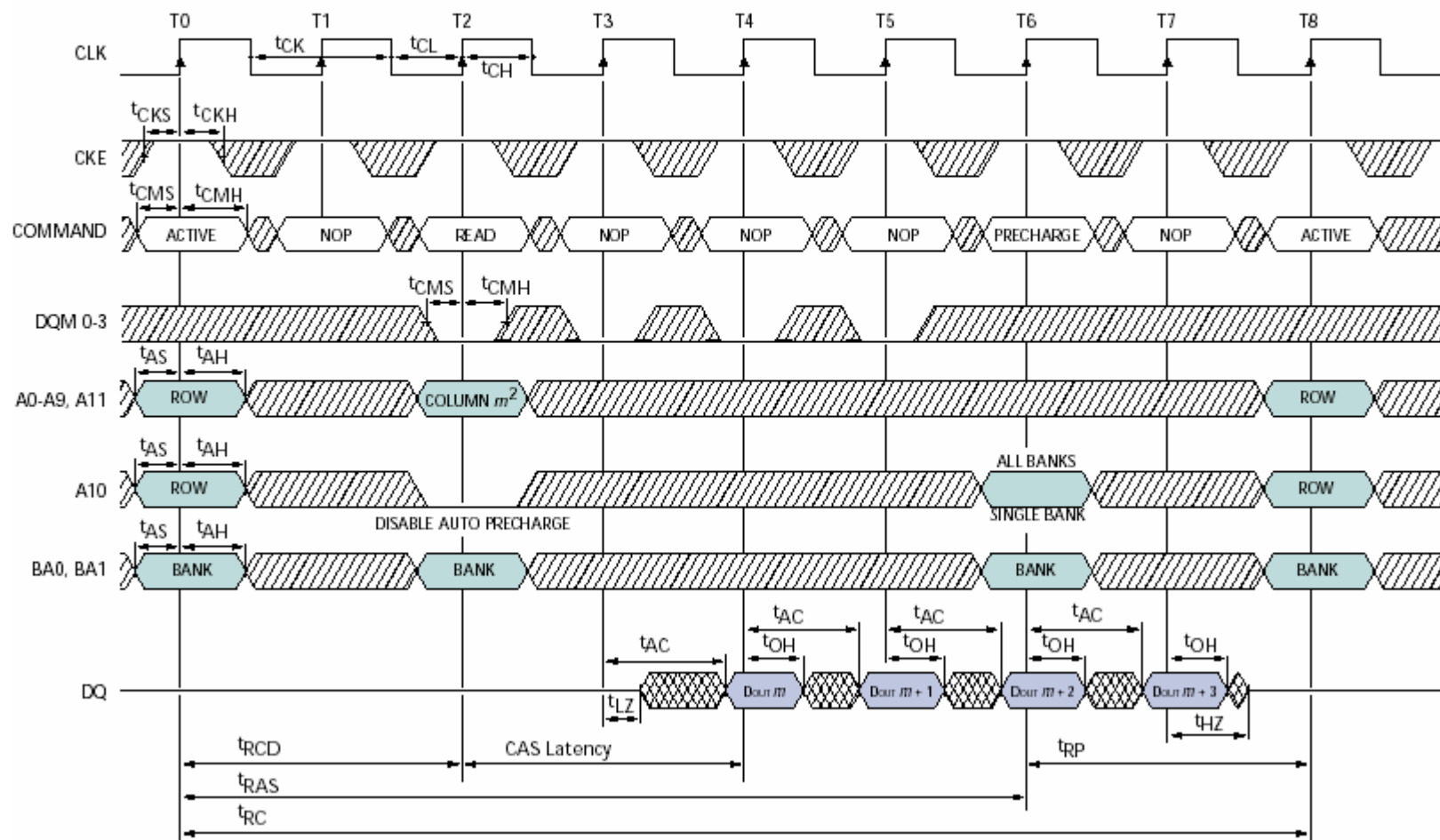
# First, Read the Datasheets

## ■ SDRAM (MT48LC4M32B2-7)

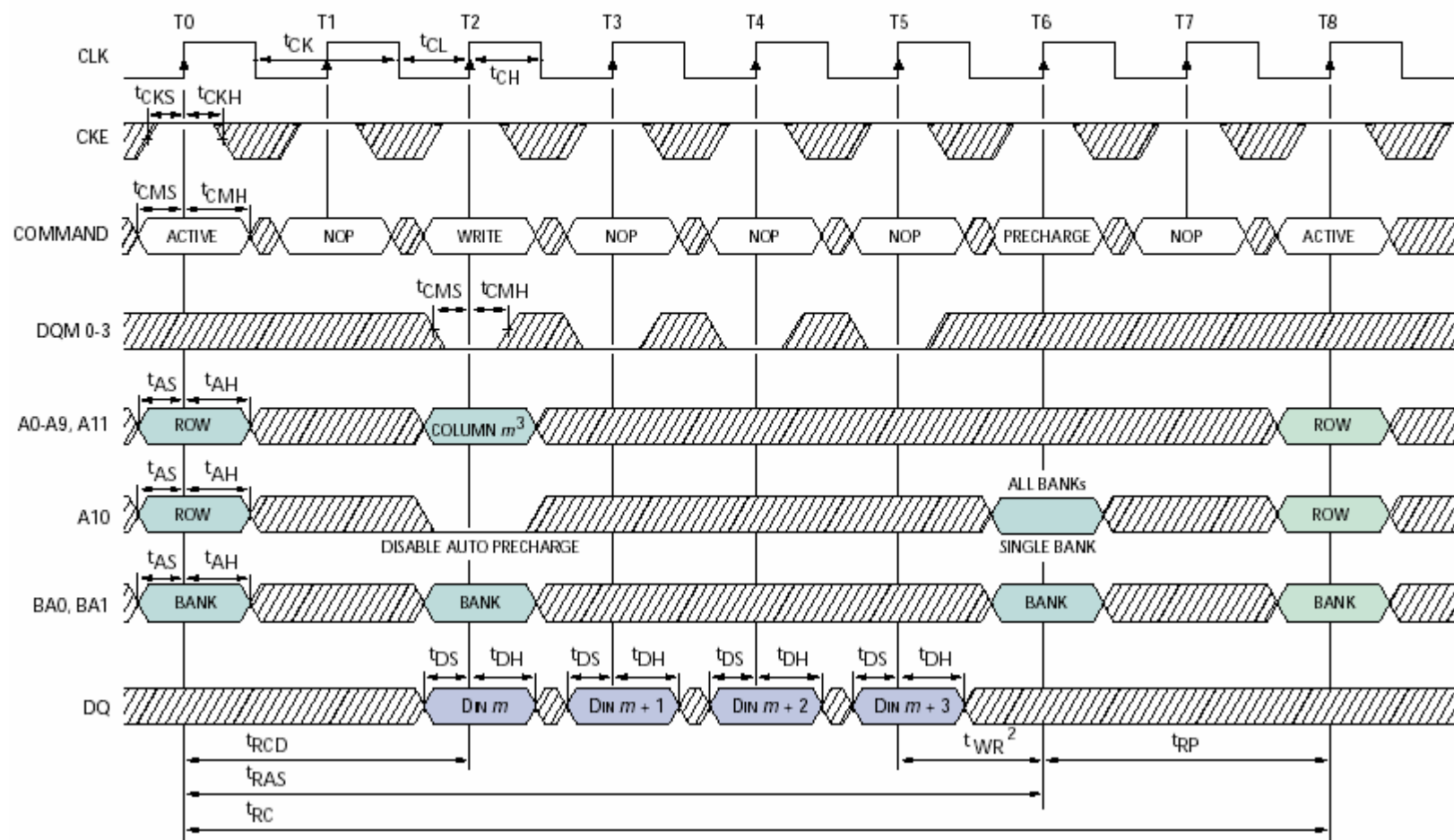
- Data In:  $t_{su} = 2 \text{ ns}$ ,  $t_h = 1 \text{ ns}$
- Data Out:  $t_{oh} = 2.5 \text{ ns}$ ,  $t_{hz}/t_{ac} = 5.5 \text{ ns}$  (CL=3)
  - 2.5 – 5.5 ns (Data Undefined)

AC CHARACTERISTICS			-6		-7			
PARAMETER		SYMBOL	MIN	MAX	MIN	MAX	UNITS	NOTES
Access time from CLK (pos. edge)	CL = 3	$t_{AC} (3)$		5.5		5.5	ns	
	CL = 2	$t_{AC} (2)$		7.5		8	ns	
	CL = 1	$t_{AC} (1)$		17		17	ns	
Address hold time		$t_{AH}$	1		1		ns	
Address setup time		$t_{AS}$	1.5		2		ns	
CLK high-level width		$t_{CH}$	2.5		2.75		ns	
CLK low-level width		$t_{CL}$	2.5		2.75		ns	
Clock cycle time	CL = 3	$t_{CK} (3)$	6		7		ns	23
	CL = 2	$t_{CK} (2)$	10		10		ns	23
	CL = 1	$t_{CK} (1)$	20		20		ns	23
CKE hold time		$t_{CKH}$	1		1		ns	
CKE setup time		$t_{CKS}$	1.5		2		ns	
CS#, RAS#, CAS#, WE#, DQM hold time		$t_{CMH}$	1		1		ns	
CS#, RAS#, CAS#, WE#, DQM setup time		$t_{CMS}$	1.5		2		ns	
Data-in hold time		$t_{DH}$	1		1		ns	
Data-in setup time		$t_{DS}$	1.5		2		ns	
Data-out high-impedance time	CL = 3	$t_{HZ} (3)$		5.5		5.5	ns	10
	CL = 2	$t_{HZ} (2)$		7.5		8	ns	10
	CL = 1	$t_{HZ} (1)$		17		17	ns	10
Data-out low-impedance time		$t_{LZ}$	1		1		ns	
Data-out hold time		$t_{OH}$	2		2.5		ns	

## READ - WITHOUT AUTO PRECHARGE<sup>1</sup>



## WRITE - WITHOUT AUTO PRECHARGE<sup>1</sup>



# Cyclone Parameters

- FPGA (Cyclone 1C20-7)
  - Column I/O's and Global Clock
  - Data In:  $t_{su} = 2.4 \text{ ns}$ ,  $t_h = 0 \text{ ns}$
  - Data Out:  $t_{outco} = 2 \text{ ns (min) } 4.4 \text{ ns (max)}$ 
    - 2.0 - 4.4 ns (Data Undefined)

<i>Table 4-36. EP1C20 Column Pin Global Clock External I/O Timing Parameters</i>							
Symbol	-6 Speed Grade		-7 Speed Grade		-8 Speed Grade		Unit
	Min	Max	Min	Max	Min	Max	
$t_{INSU}$	2.226		2.406		2.585		ns
$t_{INH}$	0.000		0.000		0.000		ns
$t_{OUTCO}$	2.000	3.926	2.000	4.358	2.000	4.795	ns

# Calculate the Window - Cyclone

## ■ SDRAM Clock Can Lead System Clock by:

Minimum of:

$$t_{\text{coutmin}}(\text{FPGA}) - t_{\text{h}}(\text{SDRAM}) = 2 \text{ ns} - 1 \text{ ns} = \mathbf{1 \text{ ns}}$$

$$t_{\text{clk}} - t_{\text{hz}}(\text{SDRAM}) - t_{\text{su}}(\text{FPGA}) = 10 \text{ ns} - 5.5 \text{ ns} - 2.4 \text{ ns} = 2.1 \text{ ns}$$

## ■ SDRAM Clock Can Lag System Clock by:

Minimum of:

$$t_{\text{oh}}(\text{SDRAM}) - t_{\text{h}}(\text{FPGA}) = 2.5 \text{ ns} - 0 \text{ ns} = \mathbf{2.5 \text{ ns}}$$

$$t_{\text{clk}} - t_{\text{coutmax}}(\text{FPGA}) - t_{\text{su}}(\text{SDRAM}) = 10 \text{ ns} - 4.4 \text{ ns} - 2 \text{ ns} = 3.6 \text{ ns}$$

## ■ Window Between +1 ns to – 2.5 ns

# Stratix Parameters

- FPGA (Stratix 1S10-6)
  - Column I/O's and Global Clock
  - Data In:  $t_{su} = 1.75 \text{ ns}$ ,  $t_h = 0 \text{ ns}$
  - Data Out:  $t_{outco} = 2 \text{ ns}(\text{min}) \text{ } 5.5 \text{ ns}(\text{max})$ 
    - 2.0 - 5.5 ns (Data Undefined)

<i>Table 4-54. EP1S10 Column Pin Global Clock External I/O Timing Parameters</i>							
Symbol	-5 Speed Grade		-6 Speed Grade		-7 Speed Grade		Unit
	Min	Max	Min	Max	Min	Max	
$t_{INSU}$	1.699		1.748		1.993		ns
$t_{INH}$	0.000		0.000		0.000		ns
$t_{OUTCO}$	2.000	5.143	2.000	5.504	2.000	6.308	ns

# Calculate the Window - Stratix

- SDRAM Clock Can Lead System Clock by:  
Minimum of:

$$t_{\text{coutmin}}(\text{FPGA}) - t_{\text{h}}(\text{SDRAM}) = 2 \text{ ns} - 1 \text{ ns} = \mathbf{1 \text{ ns}}$$

$$t_{\text{clk}} - t_{\text{hz}}(\text{SDRAM}) - t_{\text{su}}(\text{FPGA}) = 10 \text{ ns} - 5.5 \text{ ns} - 1.75 \text{ ns} = 2.75 \text{ ns}$$

- SDRAM Clock Can Lag System Clock by:  
Minimum of:

$$t_{\text{oh}}(\text{SDRAM}) - t_{\text{h}}(\text{FPGA}) = 2.5 \text{ ns} - 0 \text{ ns} = \mathbf{2.5 \text{ ns}}$$

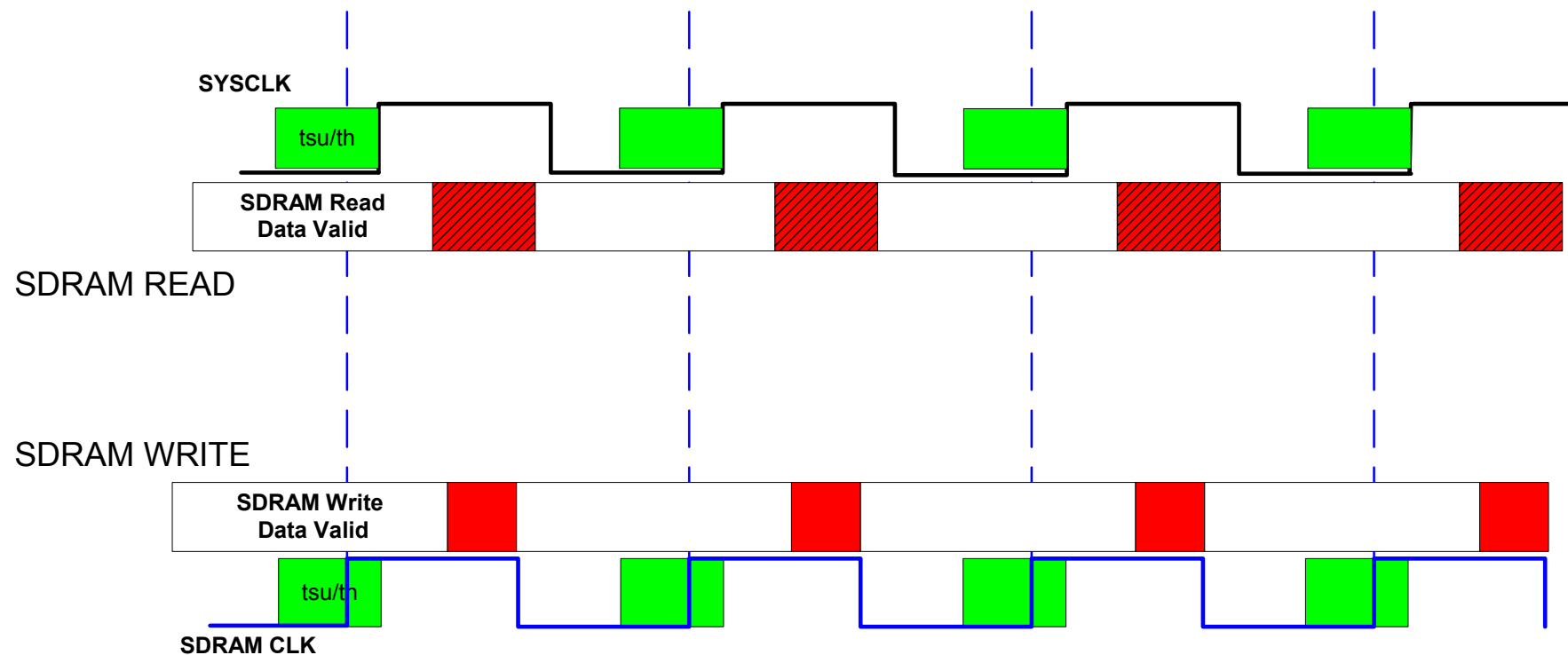
$$t_{\text{clk}} - t_{\text{coutmax}}(\text{FPGA}) - t_{\text{su}}(\text{SDRAM}) = 10 \text{ ns} - 5.5 \text{ ns} - 2 \text{ ns} = 2.5 \text{ ns}$$

- Window = +1 ns to – 2.5 ns
  - Same As Cyclone at 100 MHz
  - Windows Different at Higher Frequencies

# PLL Tuning

- Only a 3.5 ns Window (at 100 MHz)
  - +1 ns to -2.5 ns
- Center the Phase Shift in the Middle of Window
  - -0.75 ns Phase Shift
- Window Can Change, Dependent On:
  - SDRAM
  - SDRAM CAS Latency
  - FPGA Device:
    - Global Clock Versus Regional Clock
    - Column Versus Row I/O's
    - Speed Grade

# PLL Tuning (100 MHz)

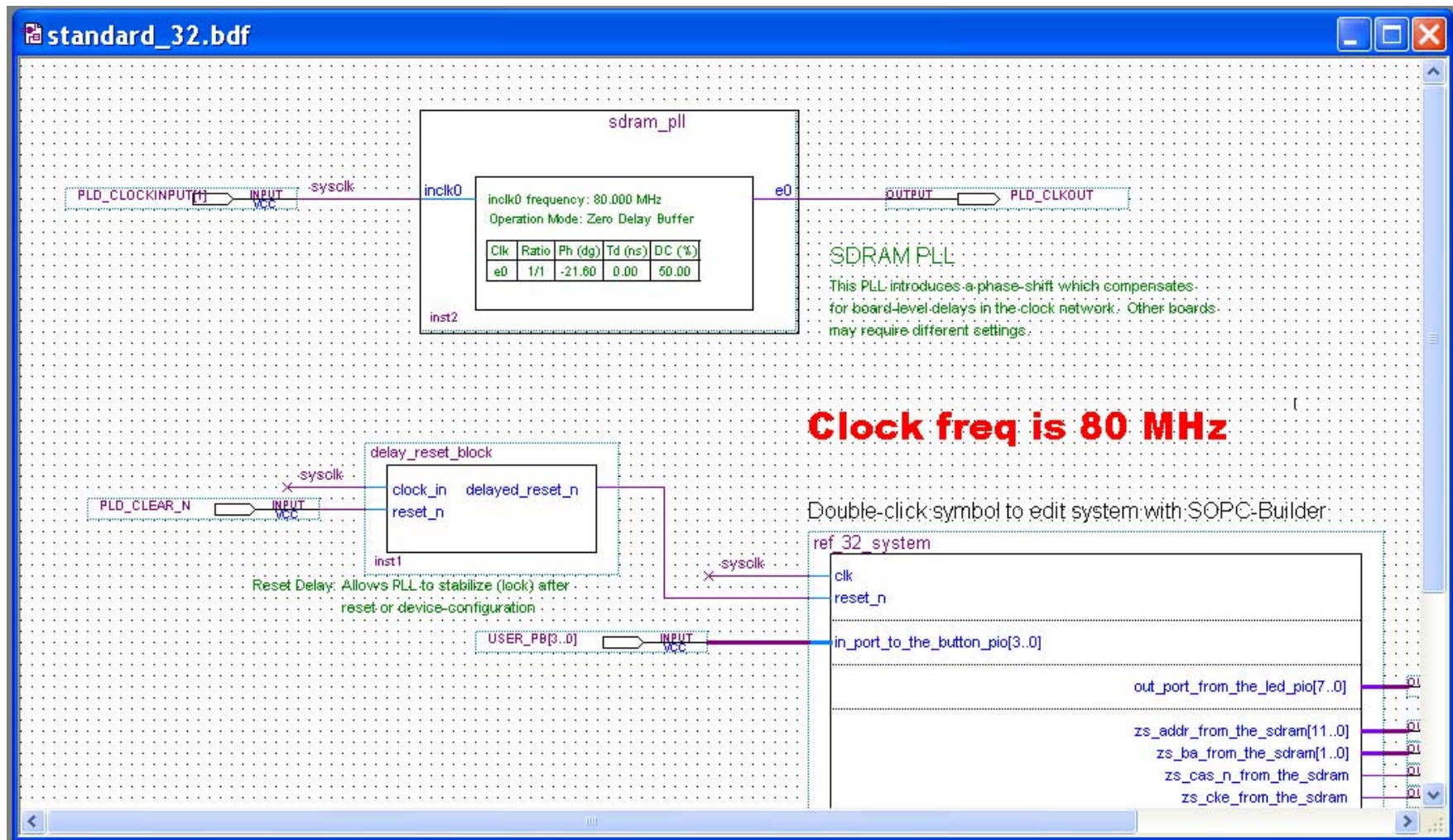


# Setting Up Clock Skew

## ■ PLL Setup

- Zero Delay Buffer
- Use Input Clock for System Clock
- Use e0 for SDRAM Clock
- Add -0.75 ns Phase Shift

# Clock Setup



# Set Phase Shift

MegaWizard Plug-In Manager - ALTCLKLOCK [page 1 of 15]

Able to implement the requested PLL

Jump to page for: General/Modes

General

Which device family will you be using? Stratix

☐ Use Fast PLL

What is the frequency of the inclk0 input? 80.000 MHz

☐ Create an 'pllena' input to selectively enable the PLL

☐ Create an 'areset' input to asynchronously reset the PLL

☐ Create an 'pfdena' input to selectively enable the phase/freq. detector

Operation mode

How will the PLL outputs be generated?

☒ Use the feedback path inside the PLL

☐ In Normal Mode

☒ In Zero Delay Buffer Mode

☐ With no compensation

☐ Create an 'fbin' input for an external feedback (External Feedback Mode)

Which output clock will be compensated for? e0

Cancel < Back Next > Finish

sdram\_pll

inclk0

e0

inclk0 frequency: 80.000 MHz

Operation Mode: Zero Delay Buffer

Clk	Ratio	Ph (dg)	Td (ns)	DC (%)
e0	1/1	-21.60	0.00	50.00

# Shift Clock e0

MegaWizard Plug-In Manager - ALTCLKLOCK [page 11 of 15]

sdram\_pll

inclk0

e0

inclk0 frequency: 80.000 MHz  
Operation Mode: Zero Delay Buffer

Clk	Ratio	Ph (dg)	Td (ns)	DC (%)
e0	1/1	-21.60	0.00	50.00

**e0 - External Output Clock**

Jump to page for: ☒ Clock e0

☒ Use this clock

Able to implement the requested PLL

Requested settings	Actual settings
Clock multiplication factor	1
Clock division factor	1
Clock phase shift	-0.75 ns -0.78
Clock time shift (nsec)	0.00
Clock duty cycle (%)	50.00

☐ Create a clock enable input

C0	C1	C2	C3	C4	C5
E0	E1	E2	E3		

Cancel < Back Next > Finish

# Skew After Compile

**standard\_32 Compilation Report**

**Timing Analyses**

**tco (Clock to Output Delays)**

	Output Name -- Register Name -- Clock Name	Actual tco
58	LEDG[3]	8.465 ns
61	LEDG[4]	8.708 ns
64	LEDG[5]	8.999 ns
67	LEDG[6]	8.677 ns
70	LEDG[7]	8.440 ns
73	PLD_CLKOUT	-0.780 ns
74	sdram_pll:inst2[altpll_component_extclk0]	-0.780 ns
76	SDRAM_A[0]	5.751 ns
79	SDRAM_A[1]	5.751 ns
82	SDRAM_A[2]	5.751 ns
85	SDRAM_A[3]	5.751 ns
88	SDRAM_A[4]	5.751 ns
91	SDRAM_A[5]	5.751 ns
94	SDRAM_A[6]	5.751 ns
97	SDRAM_A[7]	5.751 ns
100	SDRAM_A[8]	5.751 ns
103	SDRAM_A[9]	5.751 ns
106	SDRAM_A[10]	5.751 ns
109	SDRAM_A[11]	5.751 ns



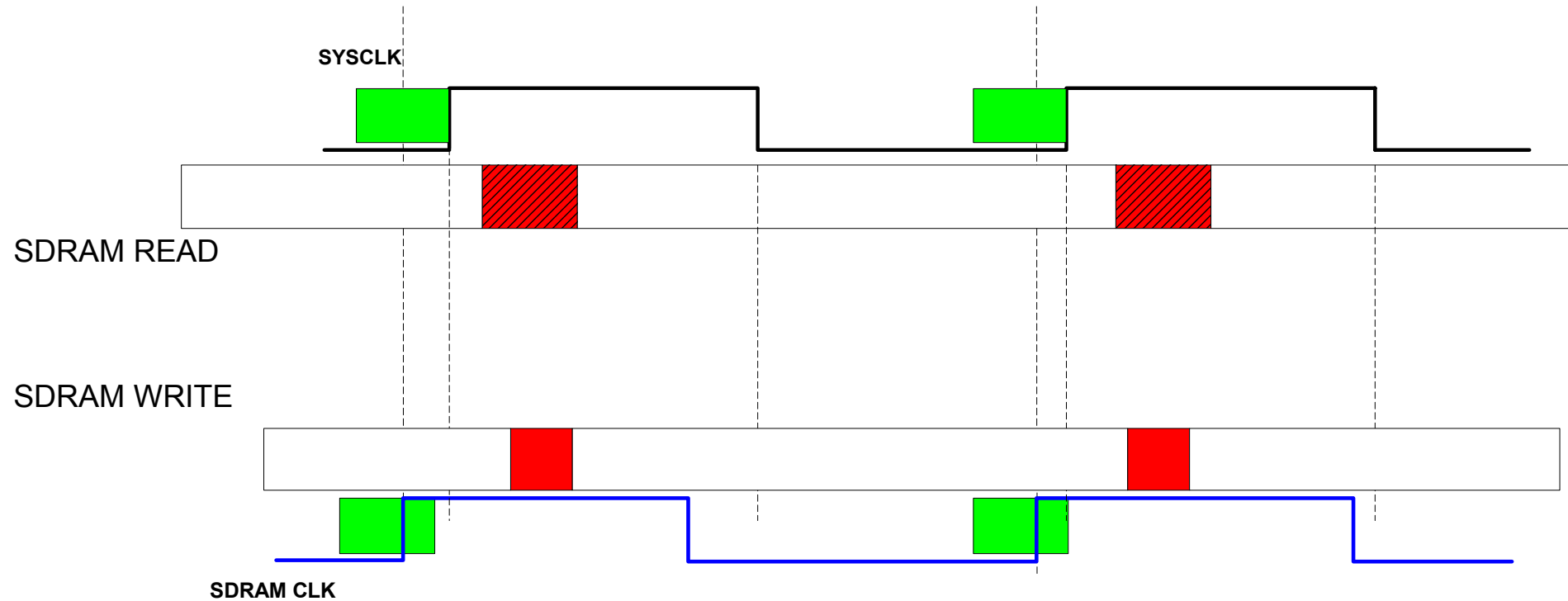
20 YEARS *of*  
**ALTERA**®  
INNOVATION

# Overtuning

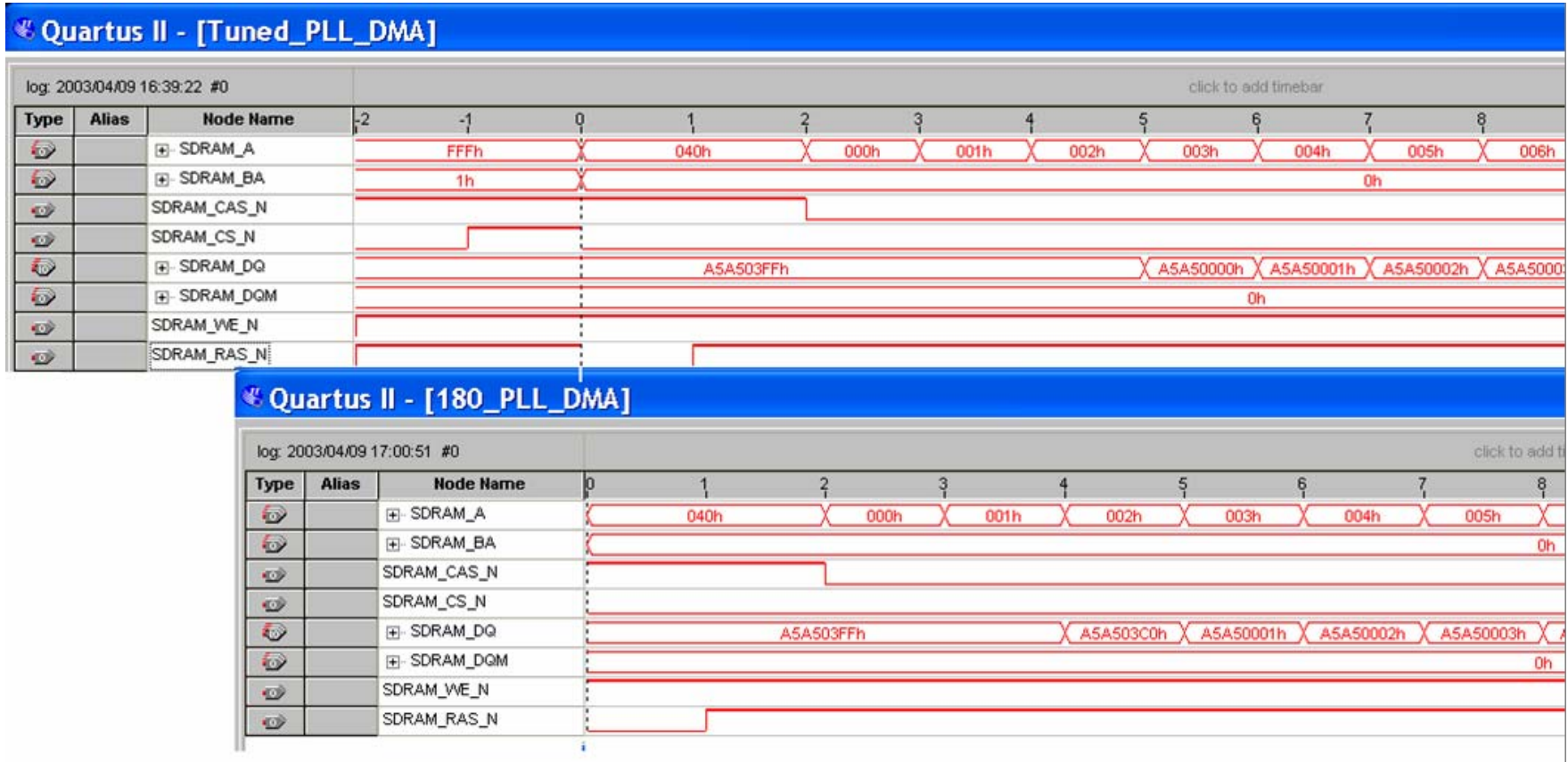
# Overtuning Symptoms

- Caused by “Trail and Error” Method
- False Tuning Window
- Single Read/Writes to SDRAM Work
- Running Code From SDRAM Fails
  - Best Way to Test Tuning
- DMA Transfer Provide Skewed Data
- This Causes the SDRAM to Provide the Data One Clock Cycle Early

# Overtuning (50 MHz)



# Overtuning: Data Received Early



# Conclusions

- Tight Control is Needed for the SDRAM Clock and System Clock For System to Function Correctly
- Verify PLL's are Correct by Executing Software from SDRAM or Using DMA to Read/Write