

---

# **Computer Aided Digital Systems Design - EE 4743/6743**

**Sherif Abdelwahed**

---

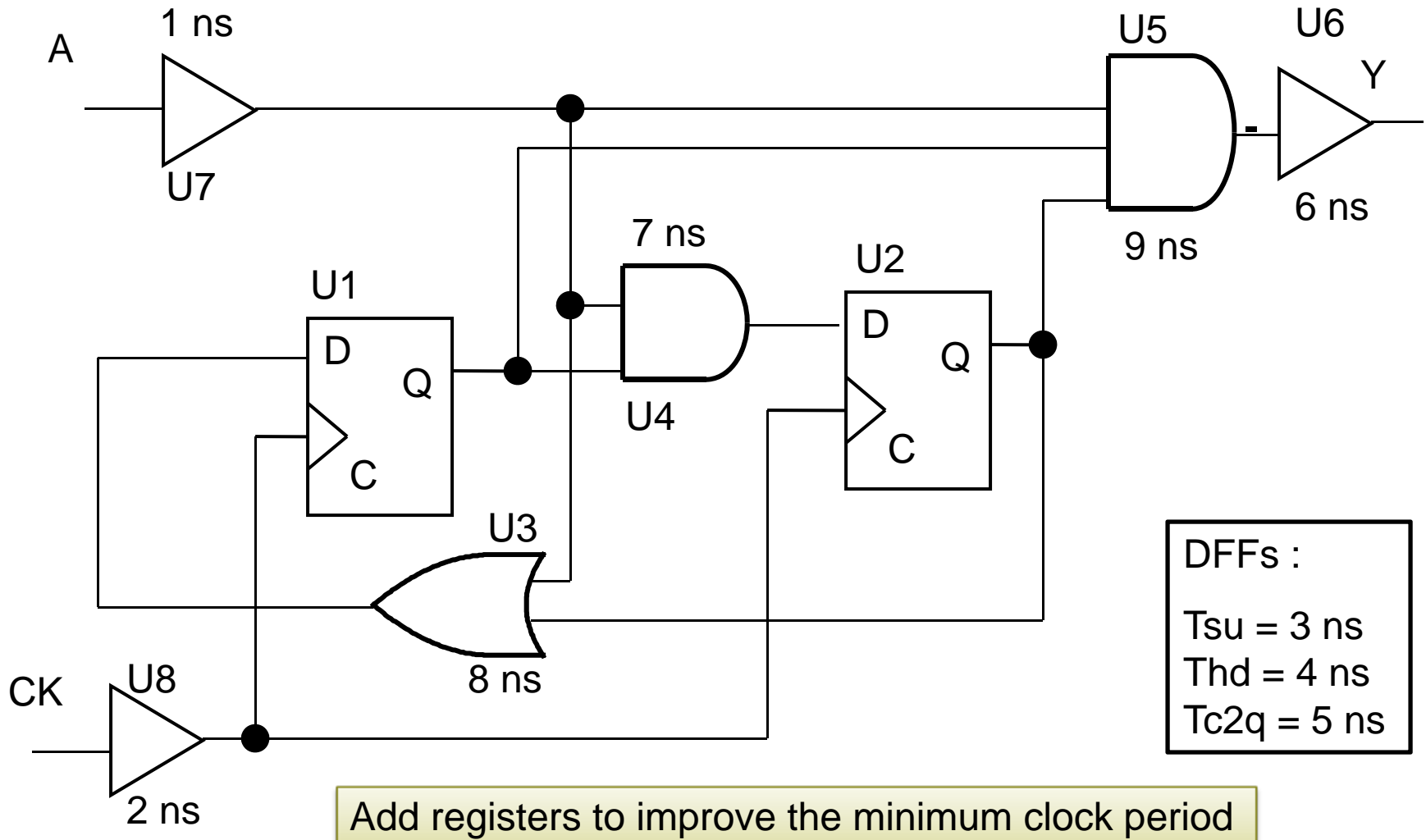
**Increasing Clock Rate**

**Department of Electrical and Computer Engineering  
Mississippi State University**

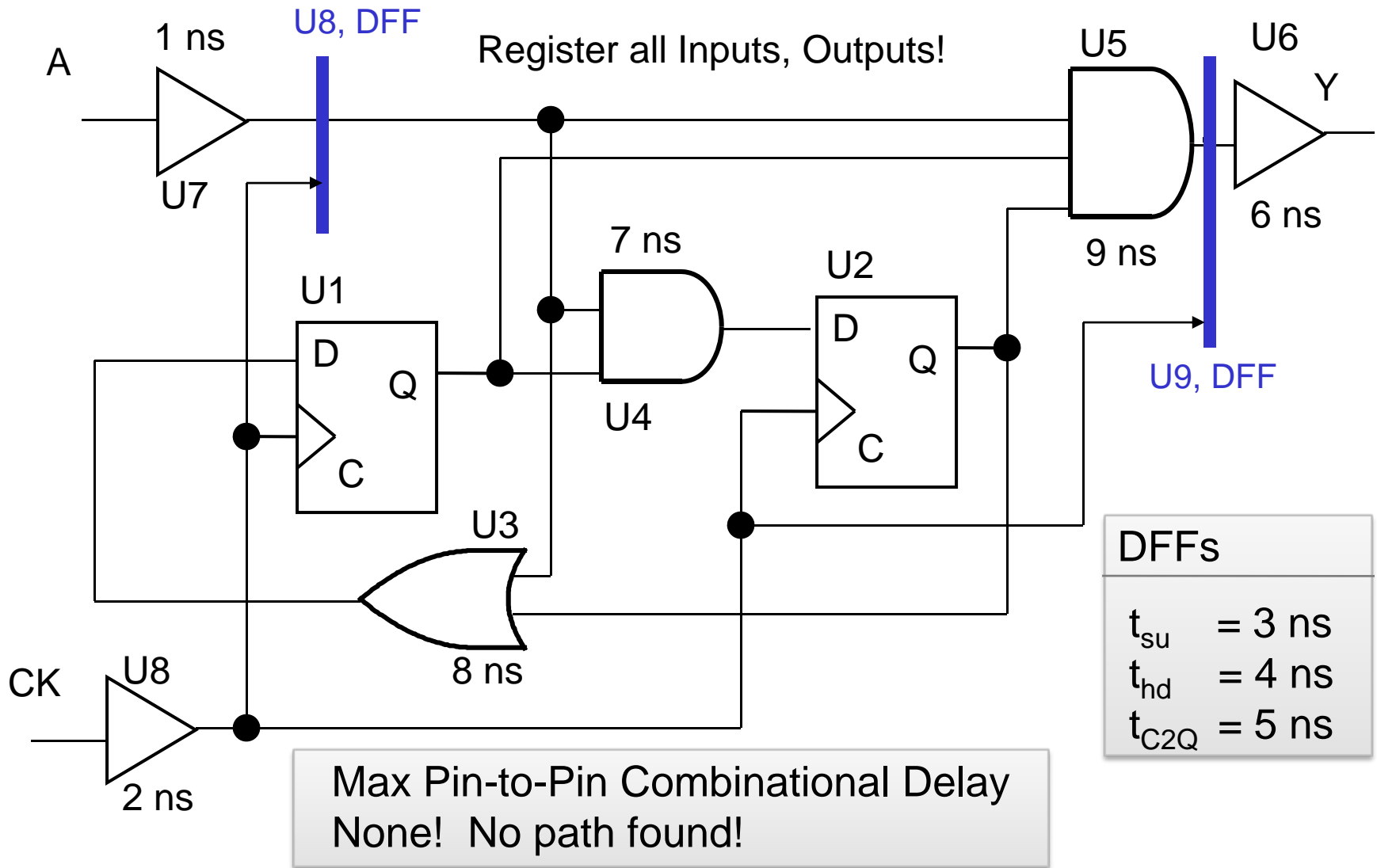
# Increasing Clock Rate

- The three types of delay –  $t_{P2P}$  ,  $t_{C2Q}$ ,  $t_{R2R}$  - determine the maximum clock rate.
  - The maximum clock rate can be changed by:
    - Reducing the propagation delay of registers and gates
    - Changing the circuit architecture
  - The design approach:
    - Pin-to-pin delay can be eliminated by registering the inputs or the
    - Clock-to-output delay can be minimized by reducing combinational paths between the clock input and output
    - Registering outputs will remove all combinational delay paths
    - Adding registers may increase register-to-register delay but can also improve the maximum clock frequency by reducing  $t_{P2P}$  ,  $t_{C2Q}$
-

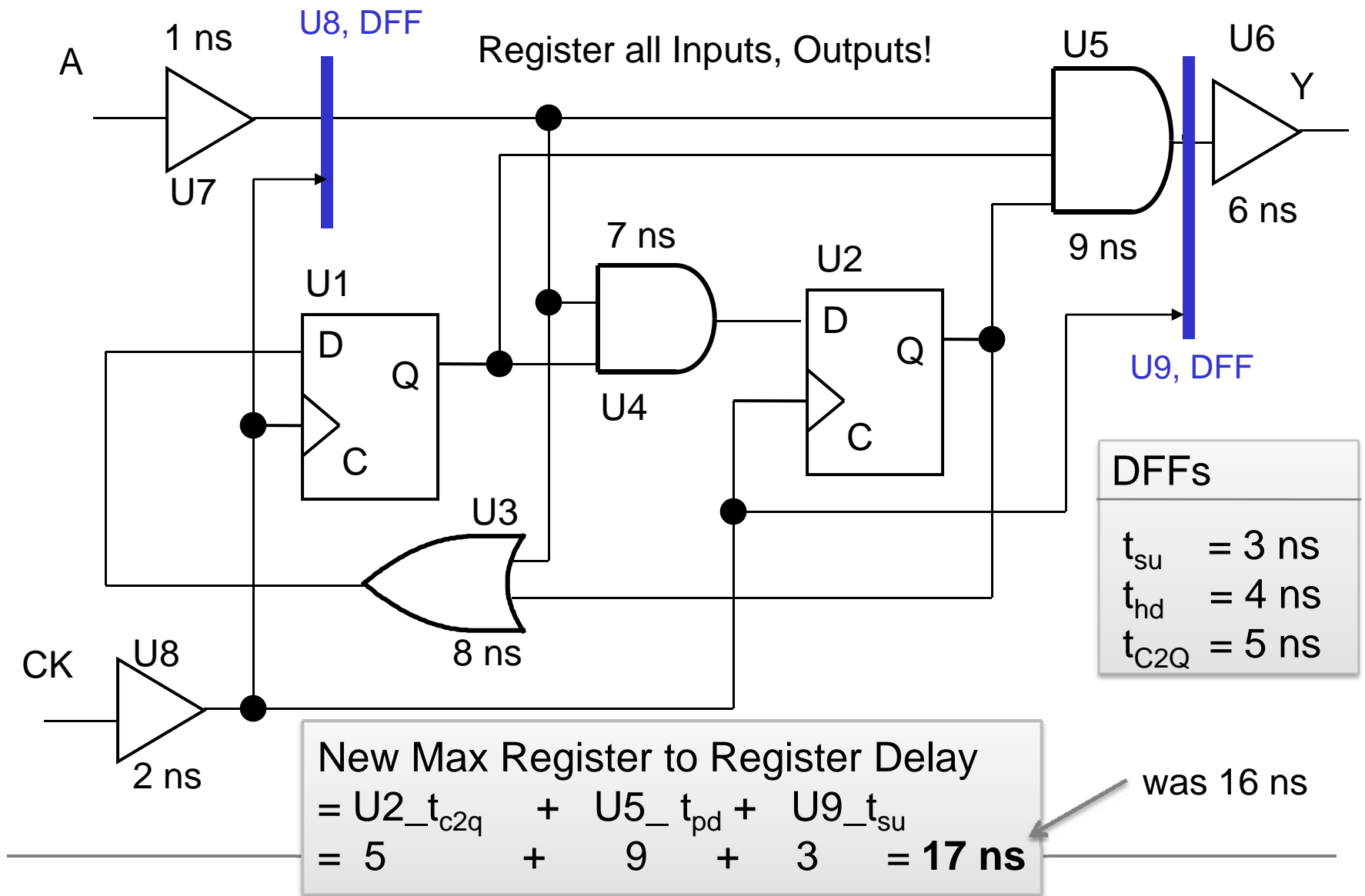
# A Timing Example



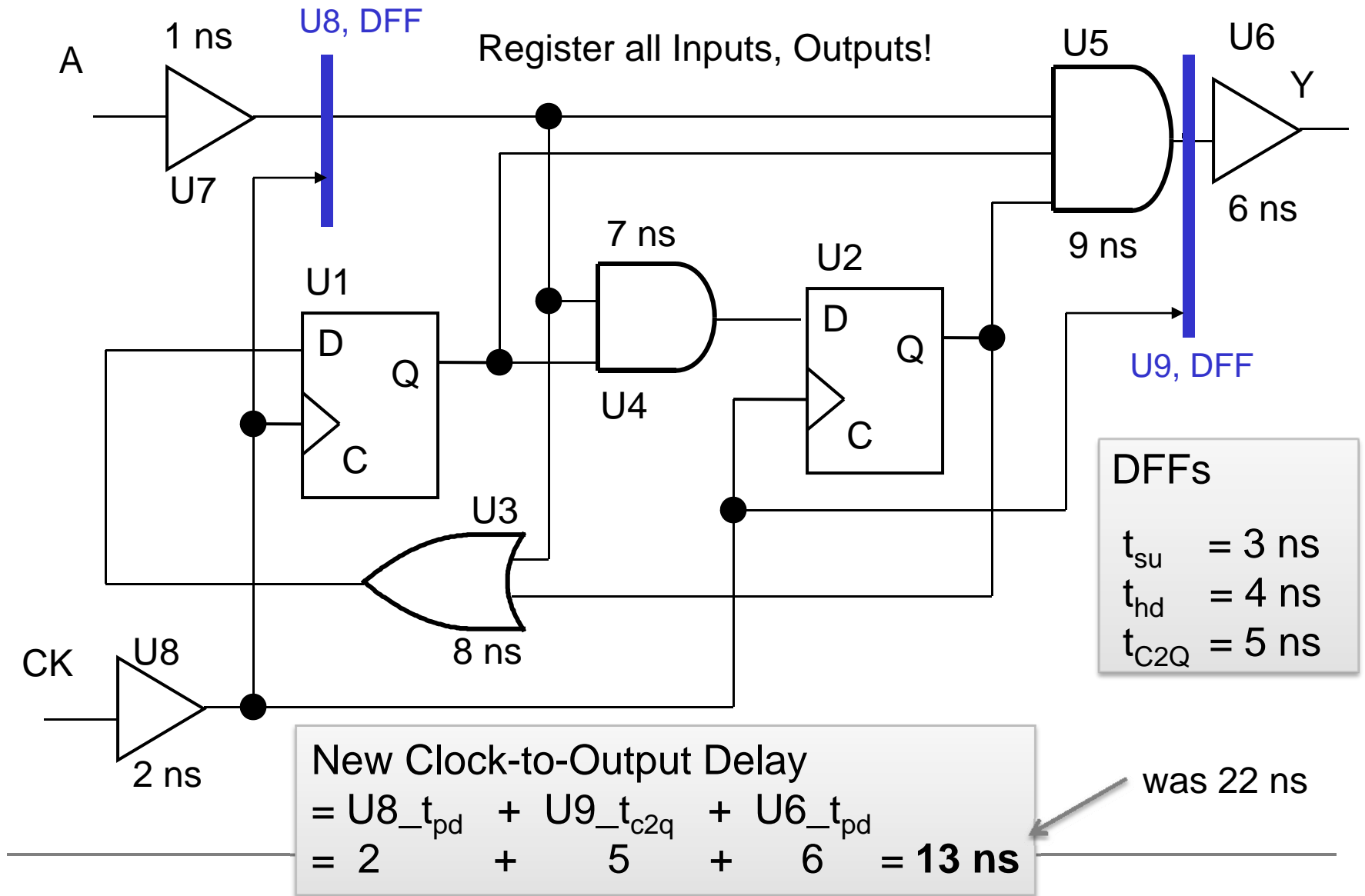
# How do we improve clock rate?



# How do we improve clock rate?



# How do we improve clock rate?



# New Timings

$$\begin{aligned}\text{Max Clock Freq} &= 1 / \text{Max} (\text{Reg2reg}, \text{Clk2Out}, \text{Pin2Pin}) \\ &= 1 / \text{Max} (17, 13, 0) \\ &= 58.8 \text{ MHz} \longleftarrow \text{was } 45.5 \text{ MHz}\end{aligned}$$

$$\begin{aligned}\text{New setup time} &= t_{\text{su}} + \text{A2D } t_{\text{pd}} \text{ max} - \text{Clk } t_{\text{pd}} \text{ min} \\ &= t_{\text{su}} + \text{U7\_}t_{\text{pd}} \text{ U7} - \text{U8\_}t_{\text{pd}} \longleftarrow \text{was } 10 \text{ ns} \\ &= 3 + 1 - 2 = 2 \text{ ns}\end{aligned}$$

$$\begin{aligned}\text{New hold time} &= t_{\text{hd}} + \text{Clk } t_{\text{pd}} \text{ max} - \text{A2D } t_{\text{pd}} \text{ min} \\ &= t_{\text{hd}} + \text{U8\_}t_{\text{pd}} - \text{U7\_}t_{\text{pd}} \longleftarrow \text{was } -2 \text{ ns} \\ &= 4 + 2 - 1 = 5 \text{ ns}\end{aligned}$$

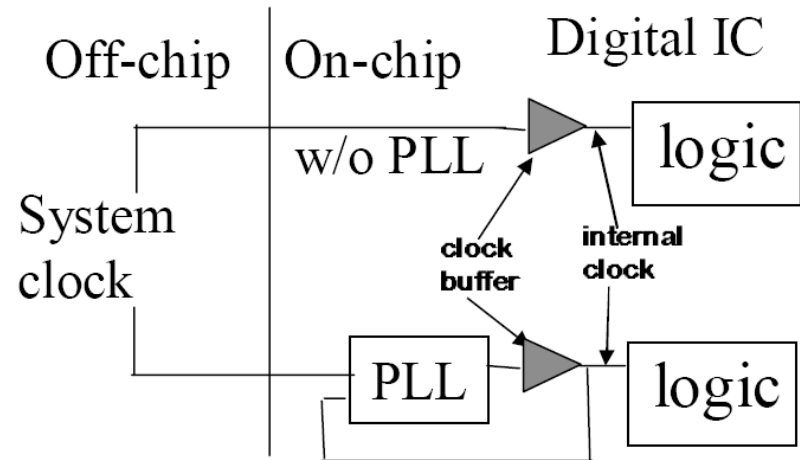
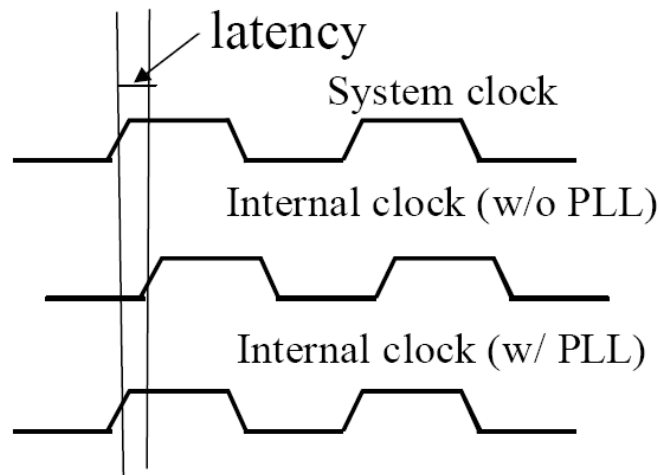
# New Datasheet

Parameter	Description	Min	Max	Units
Tclk	Clock Period	17		ns
Fclk	Clock Frequency		58.8	Mhz
Atsu	A setup time	2		ns
Athd	A hold time	5		ns
A2Y	A to Y Tpd			ns
Ck2Y	Clock to Y tpd		13	ns

Most designs have all inputs, outputs registered

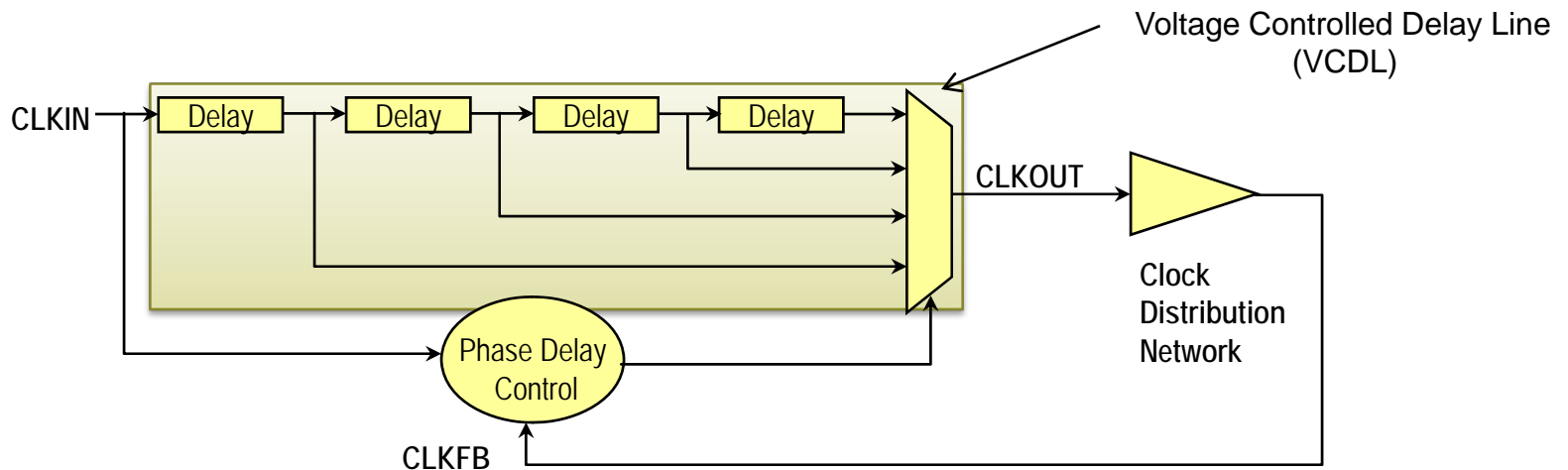
# PLLs/DLLs

- A Delay Locked Loop (DLL) or Phased Locked Loop (PLL) circuit is used to align the external clock edge at the pin with the internal clock edges at the DFF clk pins
- Applications of these circuits include: system synchronization, skew reduction, clock synthesis, clock and data synchronization



# Generic DLL Operation

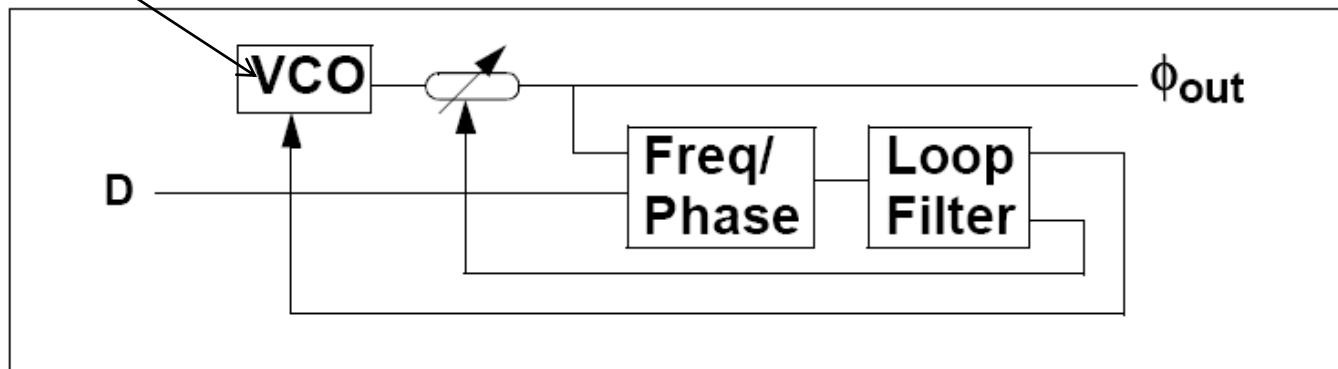
- A DLL inserts delay on the clock net until the clock input rising edge is in phase with the clock feedback rising edge
- Requires a well-designed clock distribution network: the clock edges arrive simultaneously everywhere in the part



# Phased Lock Loops

- A Phased Lock Loop adjust the frequency of the internal oscillator (Voltage Controlled Oscillator) until the generated local clock matches the external clock
- Local reference clock may be frequency multiple of input clock
- PLL depends on data input to provide “enough” signal transitions to lock onto, else PLL could lose coherency.

Voltage Controlled Oscillator



# PLL vs. DLL

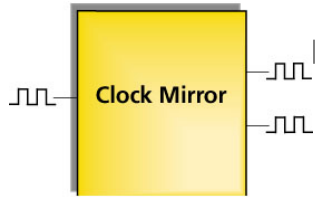
## PLL

- Frequency synthesis possible (uses a VCO)
- Input jitter is filtered
- Takes longer to acquire lock (phase error accumulates)
- Limited frequency capture range, unlimited phase capture range.
- Used more for frequency synthesis, synchronization for digital, communications, and clock recovery
- Stability is an issue (2<sup>nd</sup> or 3<sup>rd</sup> order system)

## DLL

- Not able to adjust its frequency (uses VC DL)
- No self-generated jitter
- Fast lock (phase error does not accumulate)
- Limited phase capture range
- Very attractive alternative when no frequency synthesis required.
- Always stable (1<sup>st</sup> order system).

# DLL Capabilities



- Easy clock duplication

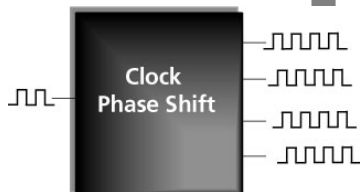
- System clock distribution

- Cleans and reconditions incoming clock



- Quick and easy frequency adjustment

- Single crystal easily generates multiple clocks



- Faster state machine utilizing different clock phases

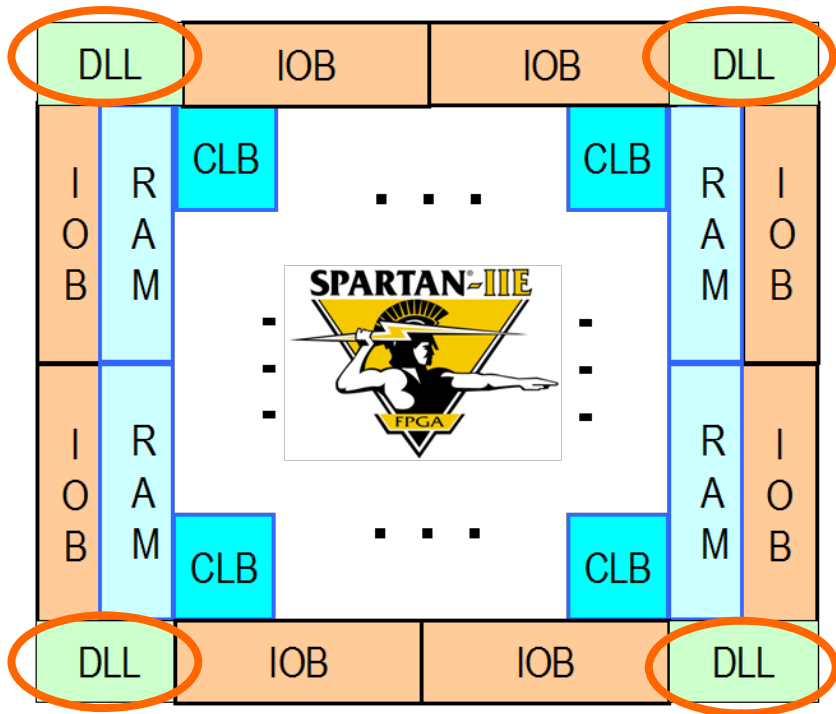
- Excellent for advance memory types



- De-skew incoming clock

- Generate fast setup and hold time or fast clock-to-outs

# DLL Blocks in Xilinx Spartan Chips



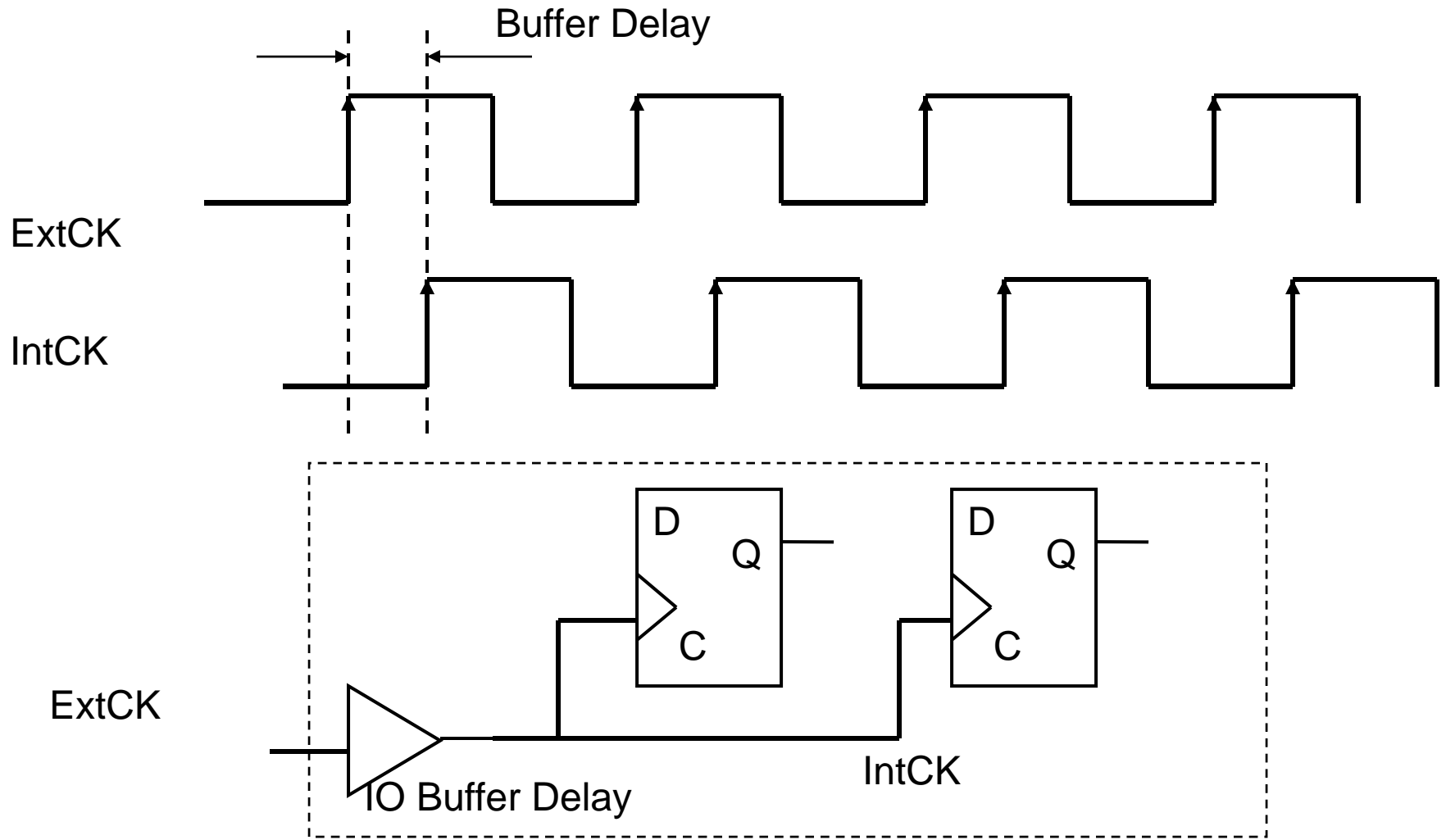
- Digital DLL Design
  - Noise insensitive
  - Scalable to new processes
  - Jitter specifications
    - +/- 100ps, << 50ps Typical
    - No cumulative phase error
  - Used in advanced memories
- Spartan IIIE has
  - 4 DLLs
  - External clock outputs

---

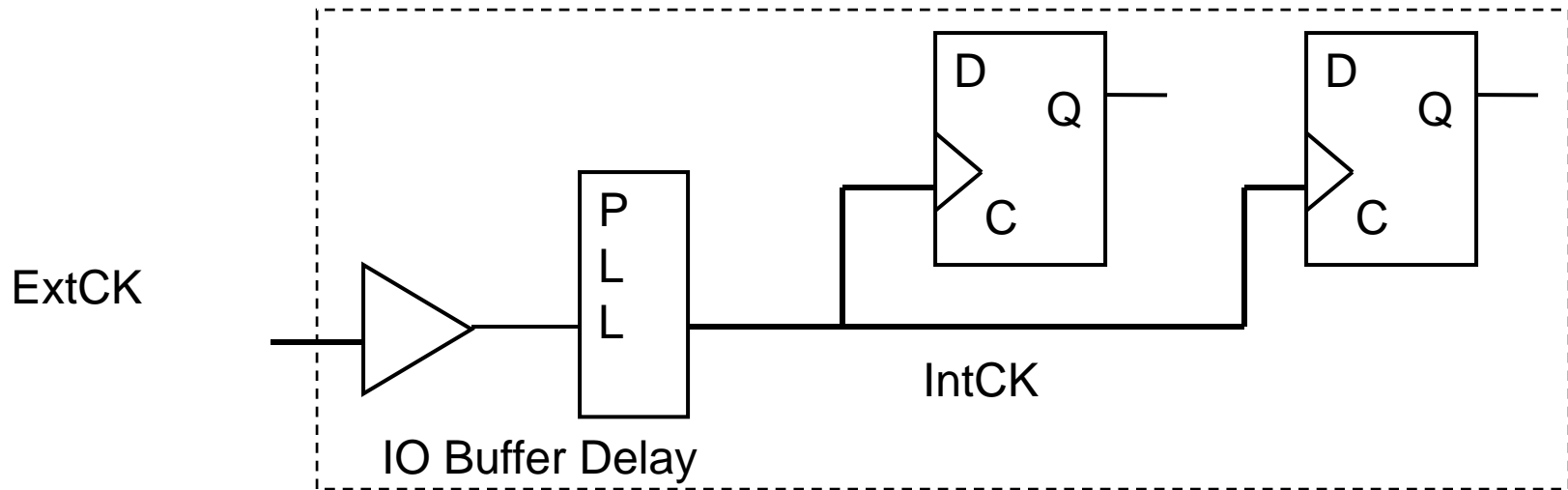
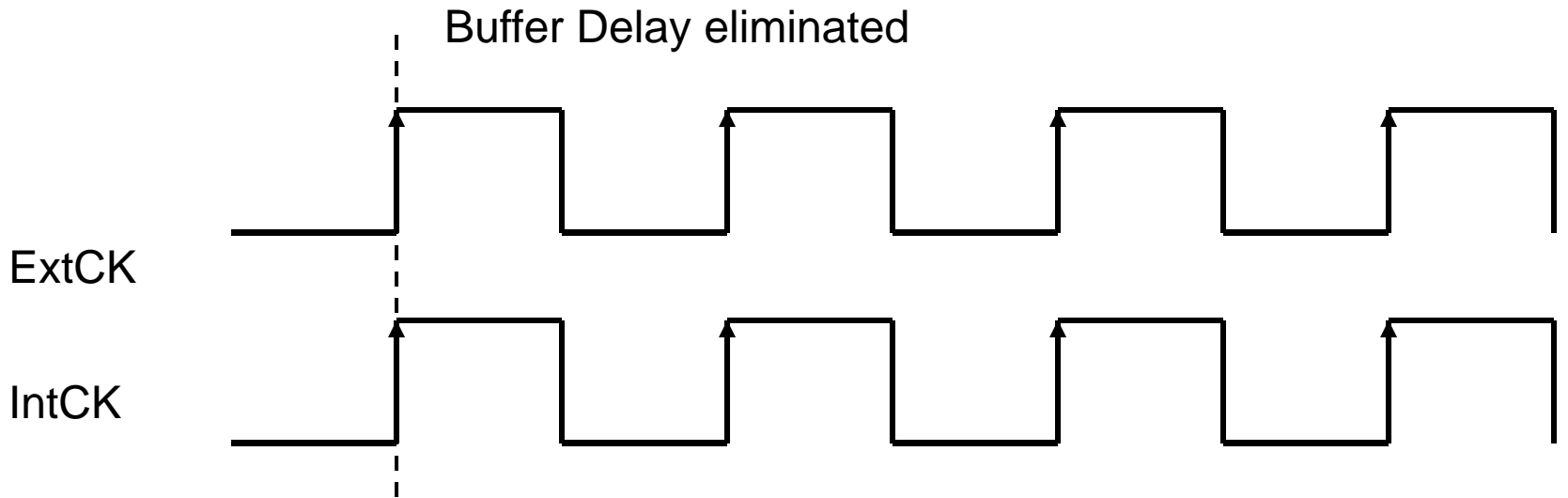
# How does a DLL/PLL Help?

- A Delay Locked Loop (DLL) or Phased Locked Loop circuit is used to align the external clock edge at the pin with the internal clock edges at the DFF clk pins
    - Some clock skew due to clock routing network from PLL will still be present, but input buffer delay eliminated
  - This means that we can drop out the Clk Tpd term from the equations
  - How does this change things?
-

# Without PLL/DLL



# With PLL/DLL



# New Timings (PLL, Inputs/Outputs Reg)

Max Register to Register Delay

$$U2_{t_{c2q}} + U5_{t_{pd}} + U9_{t_{su}} = 5 + 9 + 3 = 17 \text{ ns.}$$

$$\begin{aligned} \text{A setup time} &= t_{su} + A2D t_{pd} \text{ max} - \text{Clk } t_{pd} \text{ min} \\ &= t_{su} + (U7_{t_{pd}}) - 0 \text{ (due to PLL)} \\ &= 3 + 1 - 0 = 4 \text{ ns} \end{aligned}$$

$$\begin{aligned} \text{A hold time} &= t_{hd} + \text{Clk } t_{pd} \text{ max} - A2D t_{pd} \text{ min} \\ &= t_{hd} + 0 \text{ (due to PLL)} - (t_{pd} U7) \\ &= 4 + 0 - (1) = 3 \text{ ns} \end{aligned}$$

---

# New Timings (PLL, Inputs/Outputs Reg)

Clock to Out

$$\begin{aligned} &= U8_{t_{pd}} + U9_{t_{c2q}} + U6_{t_{hd}} \\ &= 0 \text{ (due to PLL)} + 5 + 6 = 11 \text{ ns} \end{aligned}$$

NO pin to Pin combinational delay! All inputs/outputs registered!

$$\begin{aligned} \text{Max Clock Freq} &= 1 / \text{Max (Reg2reg, Clk2Out, Pin2Pin)} \\ &= 1 / \text{Max (17, 11, 0)} \\ &= 58.8 \text{ Mhz} \end{aligned}$$

---

# New Datasheet (PLL, I/O Reg)

Parameter	Description	Min	Max	Units
Tclk	Clock Period	17		ns
Fclk	Clock Frequency		58.8	Mhz
Atsu	A setup time	4		ns
Athd	A hold time	3		ns
A2Y	A to Y Tpd			ns
Ck2Y	Clock to Y tpd		11	ns

Clock to Output improved; important in multiple chip designs. External Setup/Hold times closer to setup/hold times of internal DFFs.

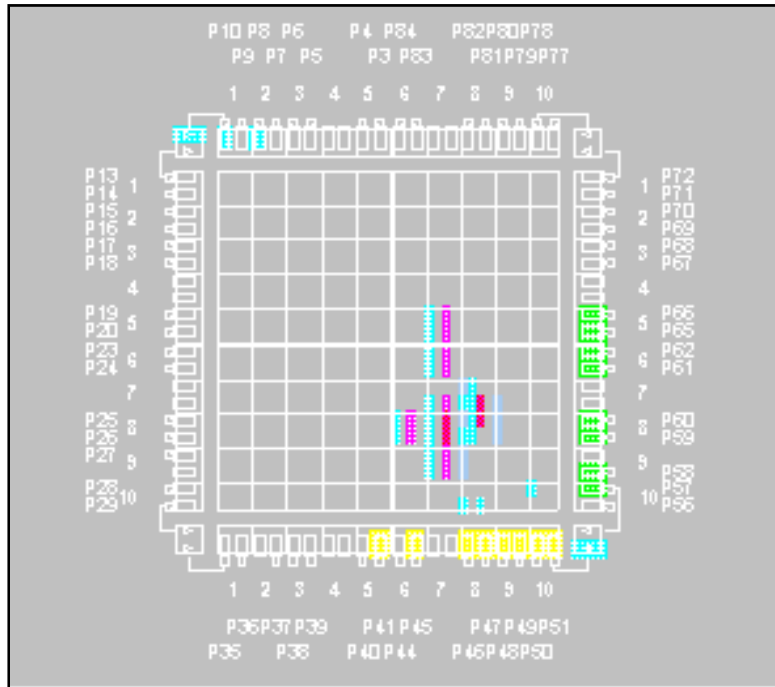
---

# Timing Constraints

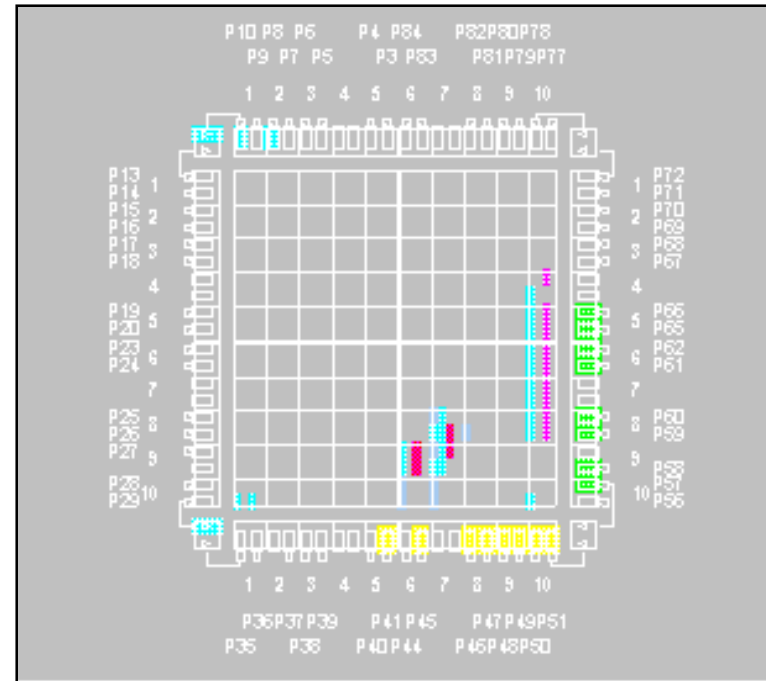
- When doing designs with FPGA devices, you need a mechanism to specify how you want the circuit to perform.
    - To meet a performance specification.
    - To evaluate how your design operates.
  - Most Design tools provide several ways to specify timing constraints as well as other design constraint such as layout area.
  - Timing constraints improve the design performance by placing logic closer together so that shorter routing resources can be used
-

# Logic Placement and Timing Constraints

Without global timing constraints



With global timing constraints



- Note that with timing constraints the logic is placed closer to the I/O pins
- Moving the logic closer to the pins improves on-chip and off-chip timing

---

# Creating a Timing Constraint

**Step 1:** Create groups of path end points

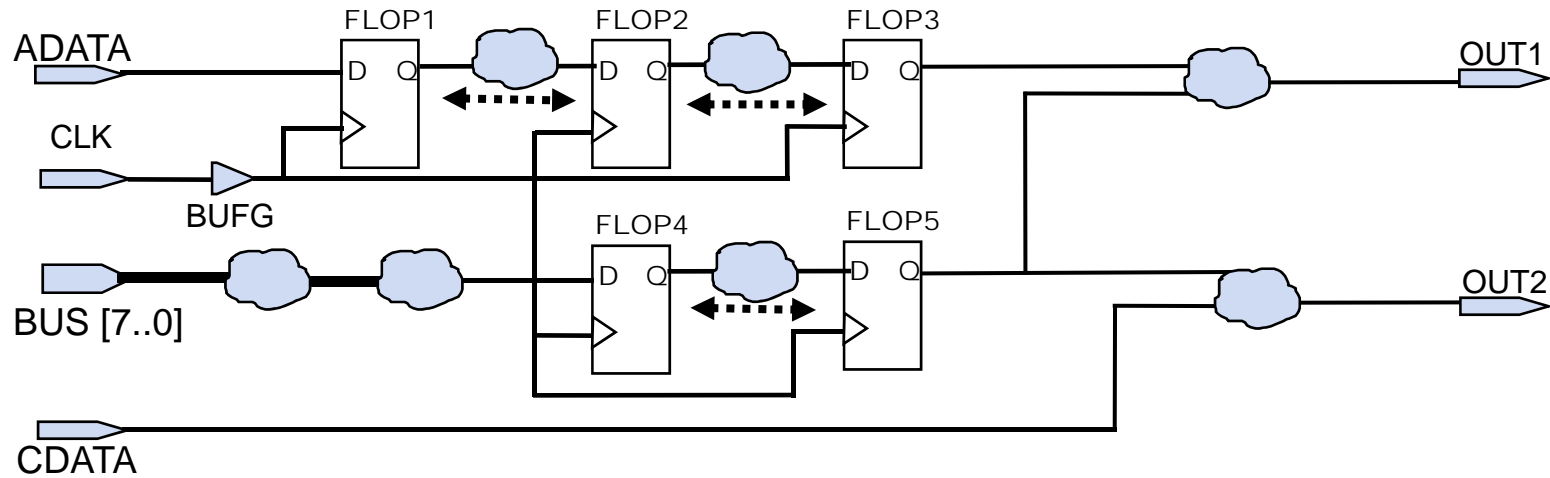
**Step 2:** Specify a timing requirement between the groups

---

# PERIOD Constraints

PERIOD constraints cover paths between synchronous elements

PERIOD constraints do not cover paths from input pads to output Pads

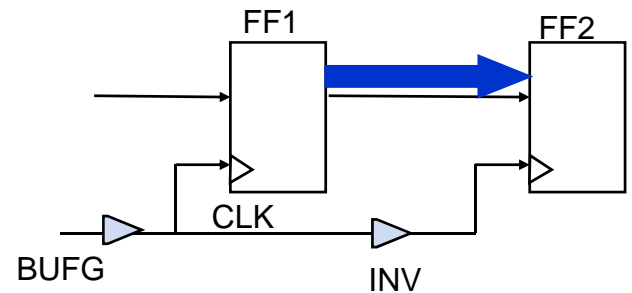


PERIOD Constraints take into consideration all available timing information

- ✓ Clock skew between the source and destination flip-flops
- ✓ Synchronous elements clocked on the negative edge
- ✓ Unequal clock duty cycles
- ✓ Clock input jitter

# An Example of a PERIOD Constraint

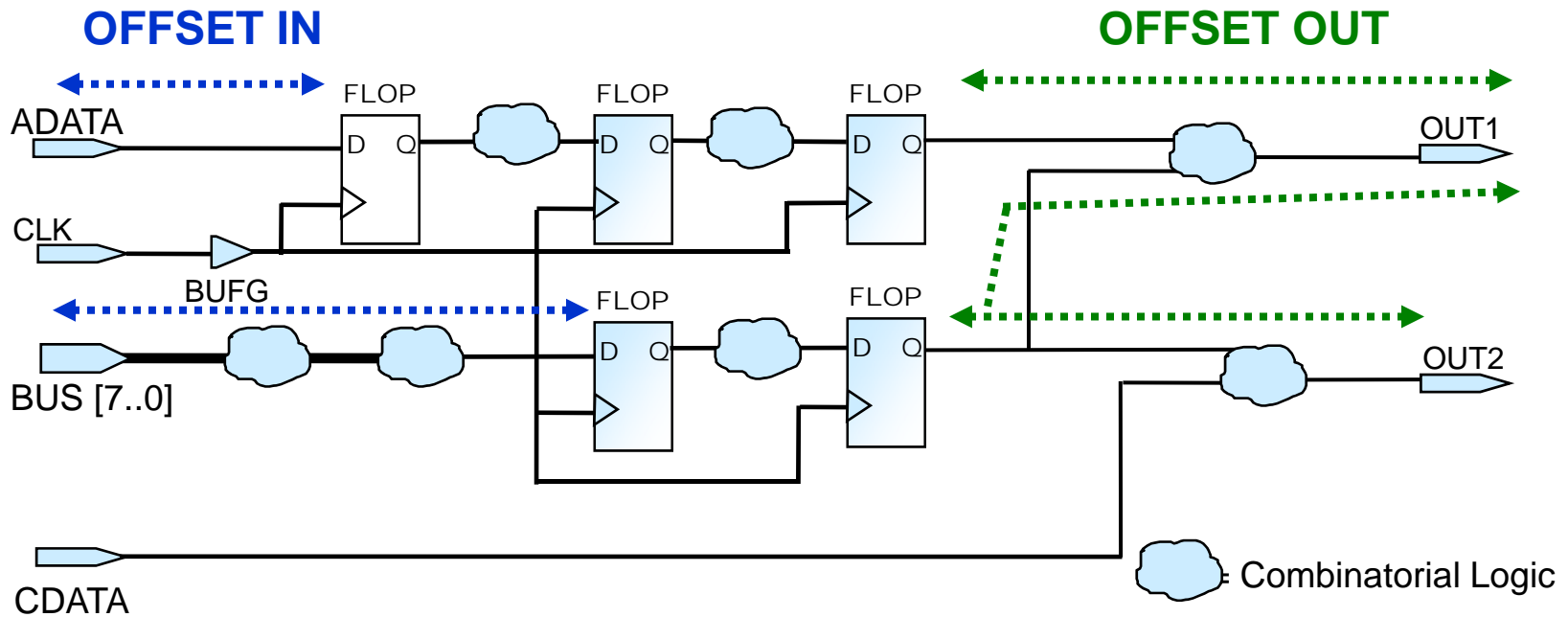
- Assume:
  - 50 percent duty cycle on CLK
  - PERIOD constraint of 10 ns
  - Because FF2 will be clocked on the falling edge of CLK, the path between the two flip-flops will be constrained to 50 percent of 10 ns = 5 ns





# OFFSET Constraints

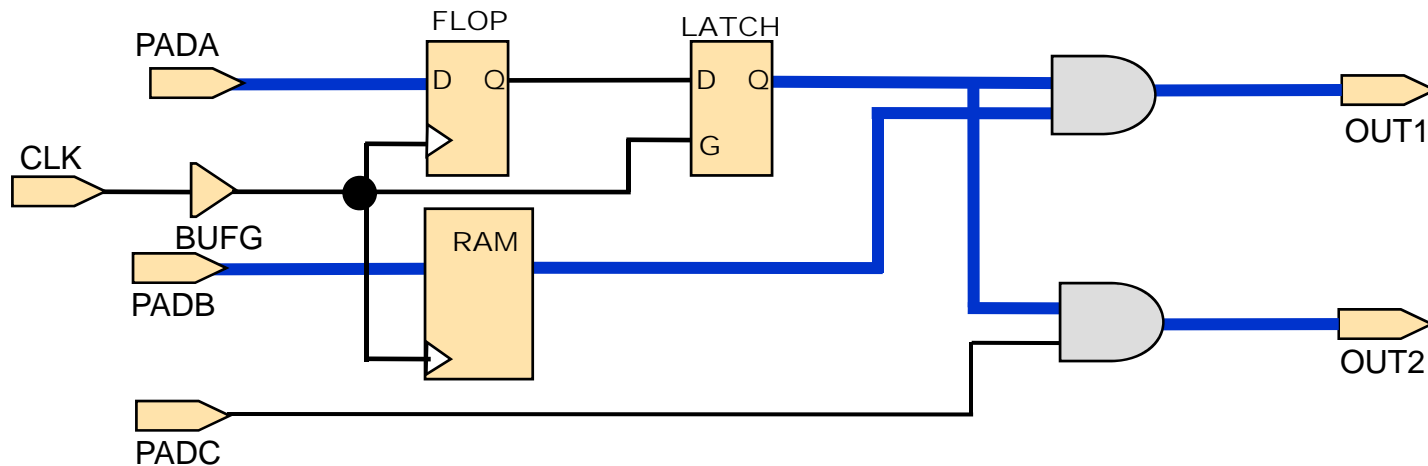
Offset In/Out constrains I/O pads to/from synchronous elements





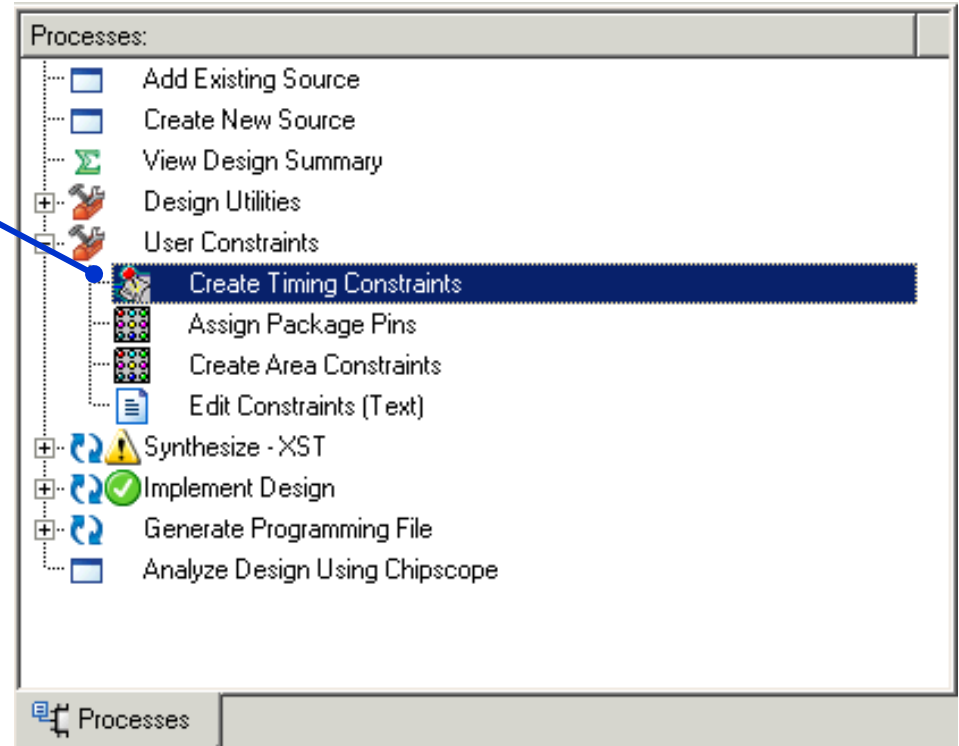
# Example

- Which paths are constrained by an OFFSET IN and an OFFSET OUT constraint on CLK?
  - OFFSET IN: PADA to FLOP and PADB to RAM
  - OFFSET OUT: LATCH to OUT1, LATCH to OUT2, and RAM to OUT1



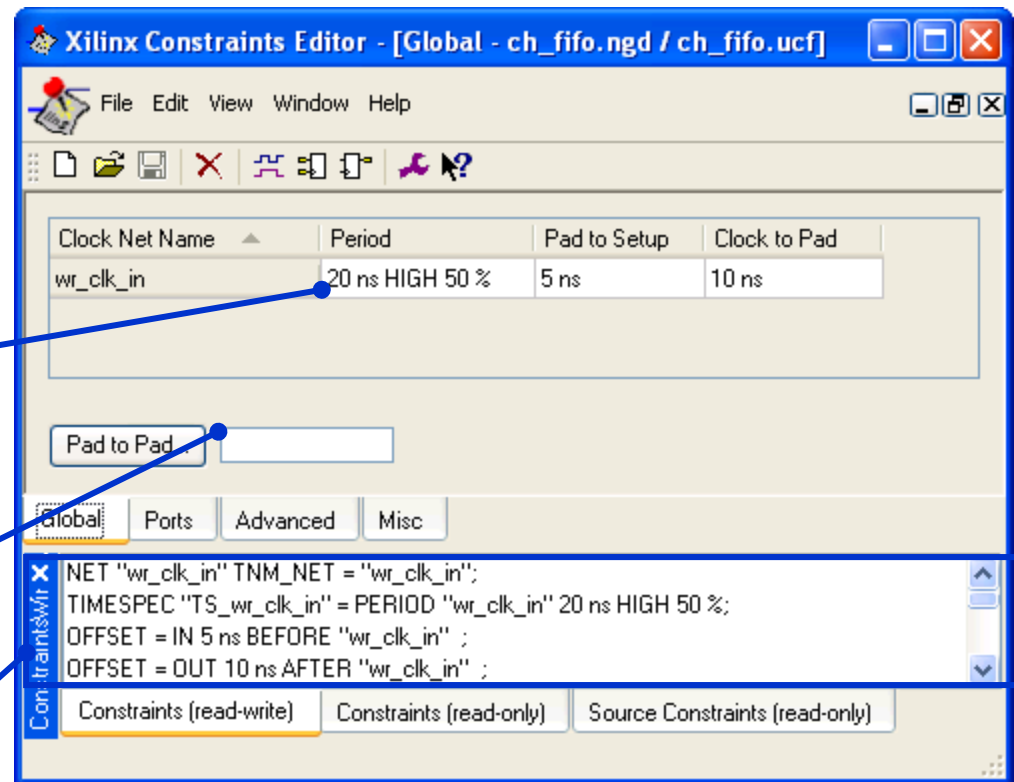
# Constraints Editor (Xilinx ISE)

- Expand **User Constraints** in the Processes for Source window
- Double-click **Create Timing Constraints**



# Constraint Editor (Xilinx ISE)

- PERIOD and Pad-to-Pad constraints can be entered on the Global tab
- Double-click here to make a PERIOD constraint
- Global Pad-to-Pad constraint
- Constraints can be deleted by selecting the **constraint in the text window** and pressing **<Delete>**



# PERIOD Constraint Options

- **TIMESPEC Name**
- **Specific constraint value**
  - Active clock edge
  - Duty cycle
- **Relative to other PERIOD TIMESPEC**
  - Useful for designs with multiple clock signals
  - Can define both frequency and phase relationships
- **Input Jitter**

**Clock Period**

Initial active edge used for OFFSET value is set to HIGH

PERIOD

INPUT\_JITTER

TIMESPEC Name: TS\_wr\_clk\_in

Clock Net Name: wr\_clk\_in

Clock Signal Definition

Specify Time

Time: 20 Units: ns

Start HIGH  Start LOW

Time HIGH: 50 Units: %

Relative to other PERIOD TIMESPEC

Reference TIMESPEC: [dropdown]

Multiply by  Divide by

Factor: 1

PHASE:

Plus  Minus

Value: 0 Units: ns

Input Jitter

Time: 0 Units: ps

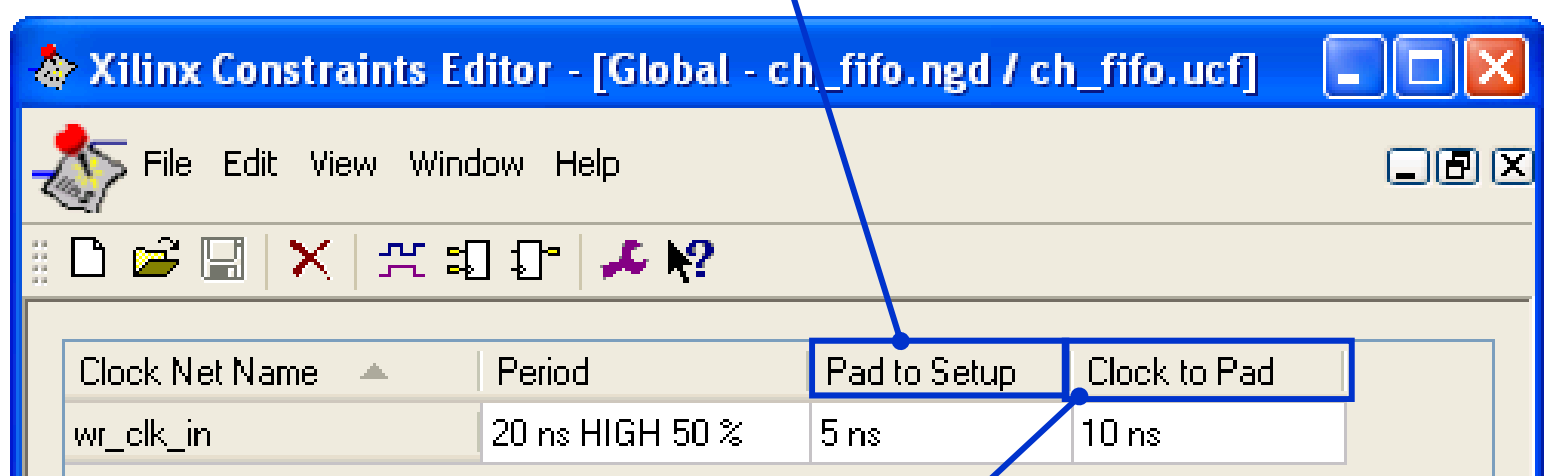
Comment:

OK Cancel Help

# Entering OFFSET Constraints

- Global OFFSET IN and OFFSET OUT constraints can be made on the **Global** tab

Pad to Setup = OFFSET IN

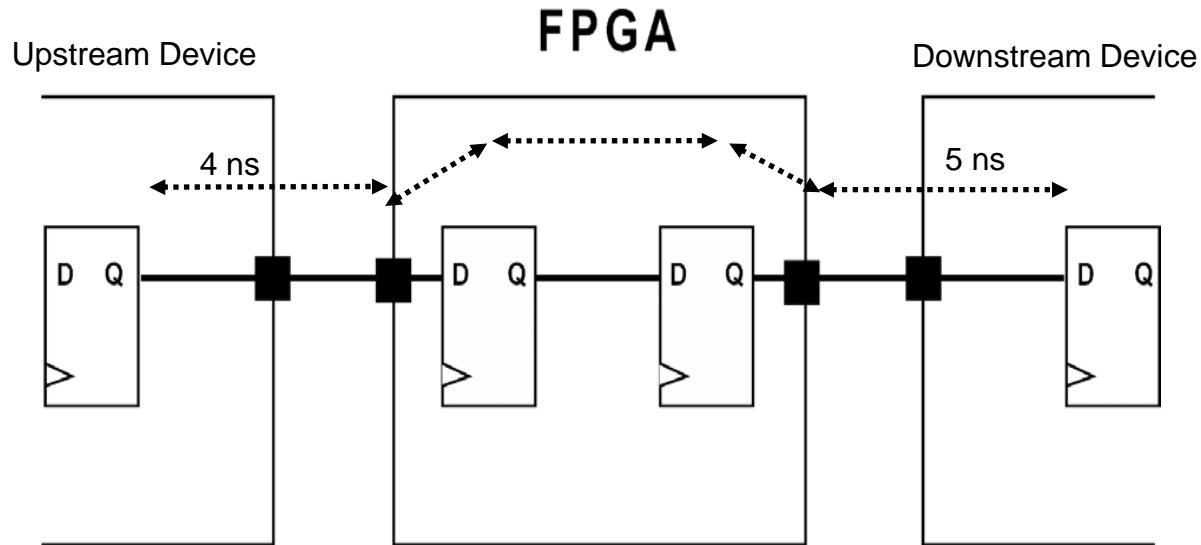


Clock to pad = OFFSET OUT

Clock Net Name	Period	Pad to Setup	Clock to Pad
wr_clk_in	20 ns HIGH 50 %	5 ns	10 ns

# Review Question

- Given the system diagram below, what values would you put in the Constraints Editor so that the system will run at 100 MHz?
  - Assume no clock skew between devices



Answer: PERIOD = 10 ns , OFFSET IN = 6 ns, and OFFSET OUT = 5 ns