INTRODUCTION TO FPGA PROGRAMMING

Lesson 11: Timing on FPGAs

Dr. Davide Cieri[1]

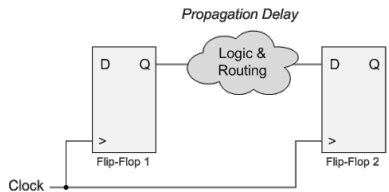
[1]Max-Planck-Institut für Physik, Munich

September 2024

MAX-PLANCK-INSTITUT FÜR PHYSIK

PROPAGATION DELAYS

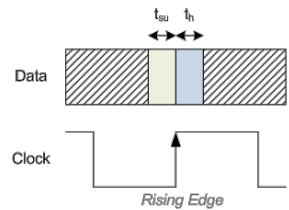
- Propagation delay is the time needed for a signal to travel from the a source to a destination component
- Sources of Propagation Delay:
  - Logic Delay: The delay introduced by the combinational logic (e.g., LUTs, multiplexers).
  - Routing Delay: The delay caused by the interconnects or wiring between logic elements.
- FPGA software include a *time analyser* tool which checks whether the design meets the timing



SETUP AND HOLD TIMES

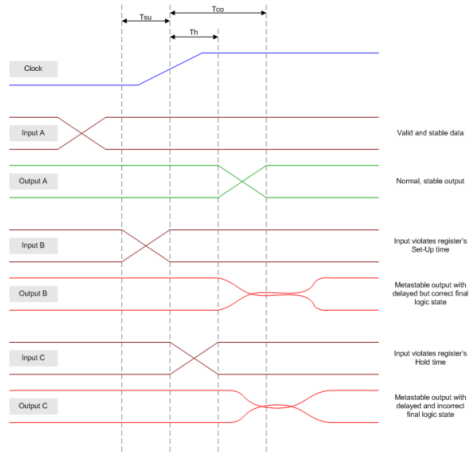
- Signal transitions on FPGA are not instantaneous of course
- Data input to a Flip-Flop must be stable for a certain time to guarantee a valid output (Setup and Hold time)
  - The *Setup* time (t_{su}) is the time required for the input to be stable *before* a clock edge
  - The *Hold* time (t_h) is the minimum amount of time required for the input to be stable *after* the clock edge
- Summing the propagation delay t_p to the setup and hold times, one gets the smallest allowed clock period (or largest clock frequency) for a design

$$t_{\text{clk}}(\text{min}) = t_{su} + t_h + t_p$$



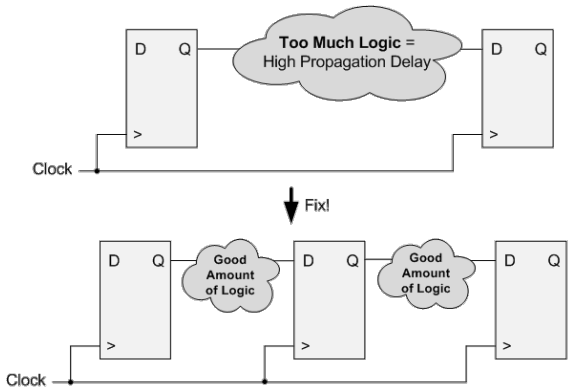
METASTABILITY

- A metastable state occurs when the output of flip-flop is unknown
- This occurs when setup or hold times are violated



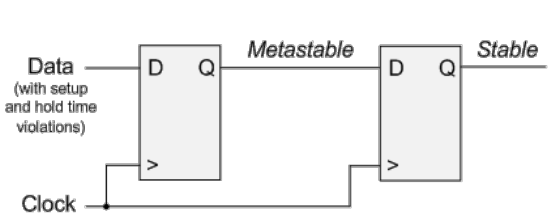
FIX TIMING ERRORS

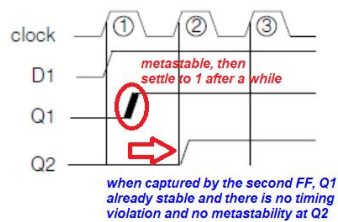
- Two main options to solve timing error inside an FPGA
  - Slow down clock frequency
  - Pipeline your logic, breaking it into stages (extra latency)



DOUBLE FLOPPING

- Other causes of metastability might can appear when
  - Sampling a signal asynchronous to the FPGA clock
  - Crossing Clock Domains
- In both cases, we can fix by "double-flopping" the data
- In the example, the output of the first flip-flop is metastable, when sampling on the first clock edge, but it will get stable when sampling on the second FF (second clock)



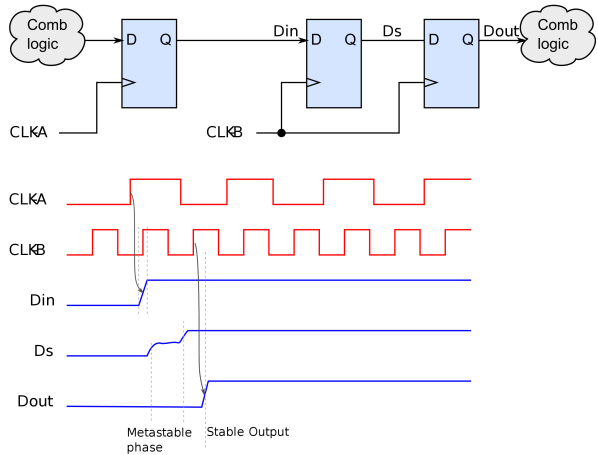


CLOCK DOMAIN CROSSING (CDC)

- In an FPGA design, you might have to work with multiple clock frequencies, especially when interfacing with external peripherals
  - E.g. HDMI running at 148.5 MHz and a camera running at 25.725 MHz
- Even if the clock relation is predictable, no guarantee that they are aligned (expect when using PLLs and MMCMs, next lesson)
- Moving from one domain to another requires a synchroniser:
  - Double Flip-Flop Synchroniser
  - Handshaking Synchroniser
  - FIFO Synchroniser

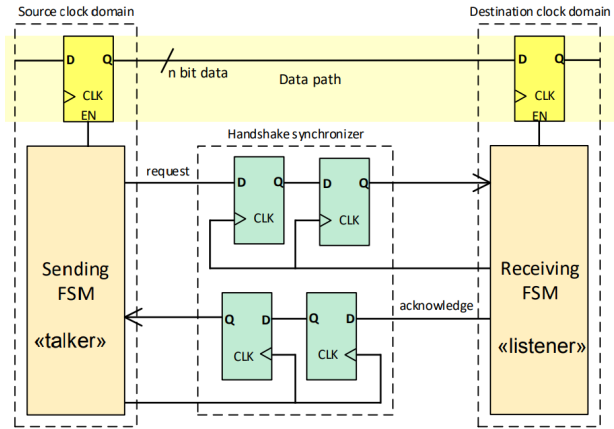
DOUBLE FLIP-FLOP SYNCHRONISER

- This Technique can be used if the following criteria is satisfied
  - The source clock domain is slower than destination clock domain
  - CDC is on a control signal (either single bit or multibit).
- It is enough to input the output data from the slower clock domain into a double FF synchroniser, running with the faster clock



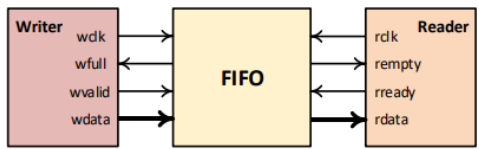
HANDSHAKING SYNCHRONISER

- Guarantees sampling of correct data independently from source and destination clock relation
  - Mainly used to synchronise vector signals not changing continuously or very often
  - Requires additional latency for the handshake acknowledge/requests
  - Data must be stable for the entire period of handshaking



FIFO SYNCHRONISER

- We introduced the concept of FIFOs in lesson 7
  - We considered in that case a synchronous FIFO, read and write clocks are the same
  - An asynchronous FIFO (independent clocks) can be used to synchronise large amount of data between clock domains
- Fundamental to keep track of FIFO status (empty/full)
  - Requires internal CDC between read/write address (multi-bit)



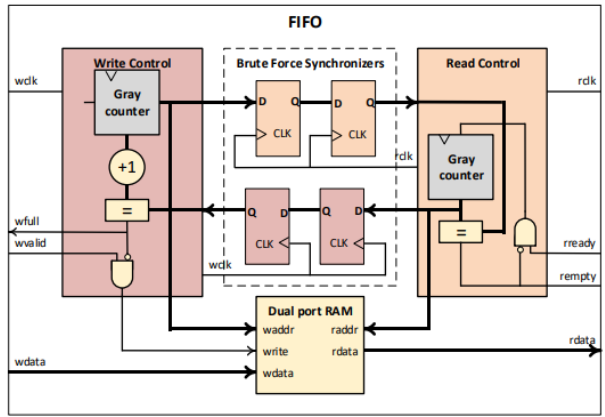
GRAY CODE

- To deal with synchronisation issues between write and read addresses, let's introduce the gray code
- Gray Code is a way to order binary numbers, such that two successive values always differ in only one bit
- Read and Write address in the FIFO can be translated to gray code

Decimal	Binary	Gray Code
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100

ASYNCHRONOUS FIFO SYNCHRONISER

- Translating the write/read address to gray code, we can deal with the comparison to check the FIFO status
  - A double-FF synchroniser is enough when moving read and write address between clock domains
  - Gray code ensures that only one bit is changed (avoiding multiple-bit metastability)

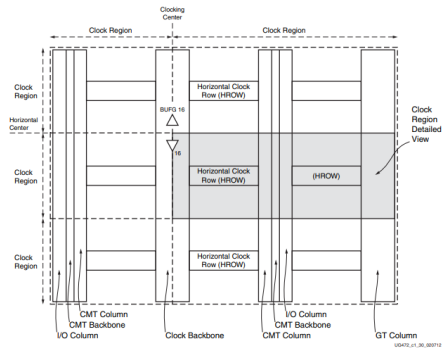


CLOCKING RESOURCES ON 7-SERIES AMD FPGA

- FPGA are divided in clock regions
- Each clock region has
  - 50 rows of Logic Slices
  - 50 IOs
  - 10 36k Block RAMs
  - 20 DSPs
- Clocks can be
  - Global
  - Regional
  - Regional plus adjacent

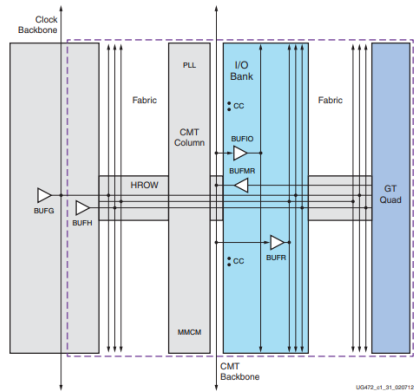
GLOBAL CLOCKS (BUFG)

- There are 32 global clock lines that can clock and provide control signals to all sequential elements in the device
  - 16 in the top and 16 in the bottom half
- Global clock buffers (BUFG) drive global clocks and are used to access global clock lines
- BUFG can only be driven by things in their own half
- Each clock region can support up to 12 of these clock lines (HROW)



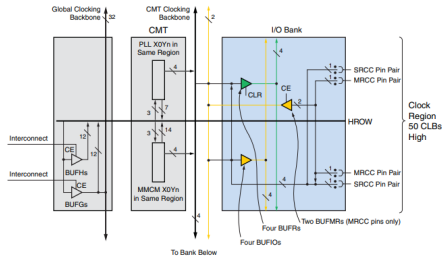
BUFH

- Each clock region has also access to 12 horizontal clock buffers (BUFH)
  - One for each HROW
- BUFH spans the full clock region and the horizontal adjacent region
- Preferred when logic spans one or two clock regions
- Same performance as BUFG



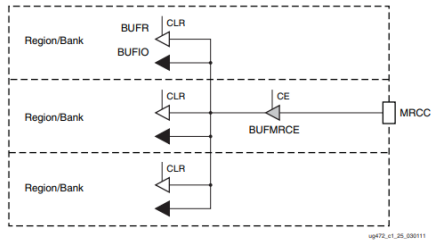
BUFR AND BUFIO

- Regional clock (BUFR)
  - Only available in a single clock region
  - 4 BUFRs per region
- I/O Clock Buffers (BUFIO)
  - Drives or can be driven by the I/O banks in a region
  - Cannot drive logic in the device
  - Ideal in source-synchronous applications where a forwarded clock is used to capture incoming data
  - 4 BUFIO per region



BUFMR

- Multi Region Clock Buffer (BUFMR)
  - Can drive BUFIO and BUFR in vertically adjacent regions
- Mostly used for multi-region I/O interfaces
- 2 BUFMRs per clock region

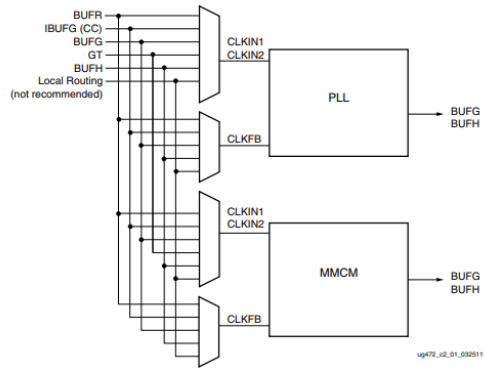


CLOCK PINS

- Four pair of clock pins per IO bank per clock region
  - Artix7 has one IO bank per region
- Differential Clocks
  - P (master) and N (slave)
  - Requires an Differential Signal Input Buffer (IBUFDS)
- Single Ended Clocks
  - Use P (master)
  - Other pin can be used for logic

CLOCK GENERATION INSIDE AN FPGA

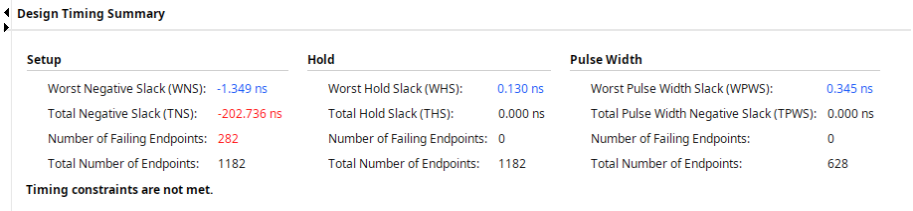
- Each Clock Region includes a Clock Management Tile (CMT) that can be used to generate clocks from several clock sources
- Each CMT consists of:
  - One Phase Locked Loop (PLL)
  - One Mixed Mode Clock Manager (MMCM)
- MMCM and PLL can be instantiated from the IP catalogue
- PLL has a subset of the features of MMCM
- Maximum allowed input/output clock frequency for CMT is 800 MHz on Artix-7



TIMING ANALYSIS

- Vivado analyses the timing conditions of your design, when running the implementation
- When transferring between sequential cells or ports, the data is:
  - Launched by one of the edges of the source clock, which is called the launch edge (t_l)

    Captured by one of the edges of the destination clock, which is called the capture edge (t_c).
- Calculates the most pessimistic Setup, Hold and Pulse Width Slacks (S)
  - $S(Su) = t_c - t_l + t_{su} - \sigma(clk) + Delay(dest.clk) - Delay(src.clk) - Delay(data)$
  - $S(h) = t_c - t_l - t_h + \sigma(clk) + Delay(dest.clk) - Delay(src.clk) - Delay(data)$
  - Pulse Width Slacks checks the actual clock pulse width and period



Design Timing Summary

Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	-1.349 ns	Worst Hold Slack (WHS):	0.130 ns	Worst Pulse Width Slack (WPWS):	0.345 ns
Total Negative Slack (TNS):	-202.736 ns	Total Hold Slack (THS):	0.000 ns	Total Pulse Width Negative Slack (TPWS):	0.000 ns
Number of Failing Endpoints:	282	Number of Failing Endpoints:	0	Number of Failing Endpoints:	0
Total Number of Endpoints:	1182	Total Number of Endpoints:	1182	Total Number of Endpoints:	628

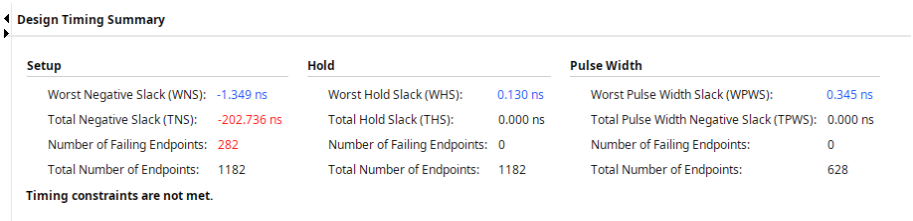
Timing constraints are not met.

TIMING CONSTRAINTS IN VIVADO

- We already saw that timing constraint in Vivado are applied in `.xdc` files
- You can define there the following constraints
  - Clocks (Primary, Virtual, Generated, Automatically derived). Documentation
  - Input and Output Delays. Documentation
  - Timing Exceptions. Documentation
- Vivado includes a Constraint Wizard that checks the code for missing constraints (runs at synthesis or implementation stage)

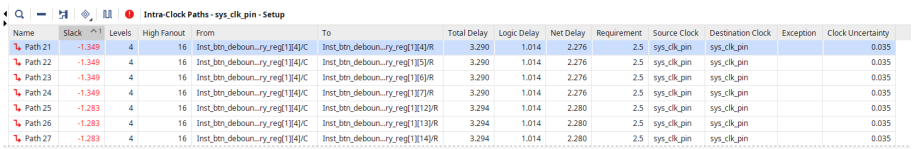
DOING TIMING ANALYSIS

- You can check the timing of your design after synthesis or implementation
  - If Total Negative Slack (TNS)
  - And Total Hold Slack (THS)
  - And Total Pulse Width Slack (TPWS)
  - ... are greater than 0, you have met timing.



CHECK THE FAILING PATHS

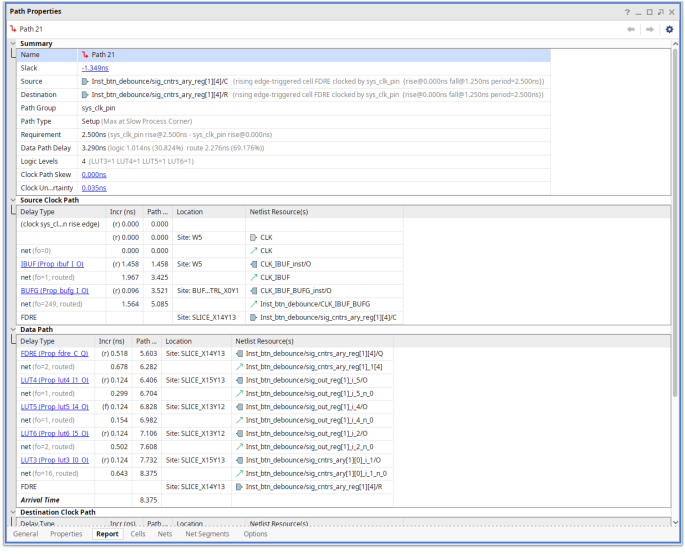
- In the timing report, you can check the failing paths in the design
- Even when timing is met, you can check for the near offenders
- Open the path for more info



Intra-Clock Paths - sys_clk_pin - Setup

Name	Slack	Levels	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source Clock	Destination Clock	Exception	Clock Uncertainty
Path 21	-1.349	4	16	Inst_btn_deboun...ry_reg[1][4]/C	Inst_btn_deboun...ry_reg[1][4]/R	3.290	1.014	2.276	2.5	sys_clk_pin	sys_clk_pin		0.035
Path 22	-1.349	4	16	Inst_btn_deboun...ry_reg[1][4]/C	Inst_btn_deboun...ry_reg[1][5]/R	3.290	1.014	2.276	2.5	sys_clk_pin	sys_clk_pin		0.035
Path 23	-1.349	4	16	Inst_btn_deboun...ry_reg[1][4]/C	Inst_btn_deboun...ry_reg[1][6]/R	3.290	1.014	2.276	2.5	sys_clk_pin	sys_clk_pin		0.035
Path 24	-1.349	4	16	Inst_btn_deboun...ry_reg[1][4]/C	Inst_btn_deboun...ry_reg[1][7]/R	3.290	1.014	2.276	2.5	sys_clk_pin	sys_clk_pin		0.035
Path 25	-1.283	4	16	Inst_btn_deboun...ry_reg[1][4]/C	Inst_btn_deboun...ry_reg[1][12]/R	3.294	1.014	2.280	2.5	sys_clk_pin	sys_clk_pin		0.035
Path 26	-1.283	4	16	Inst_btn_deboun...ry_reg[1][4]/C	Inst_btn_deboun...ry_reg[1][13]/R	3.294	1.014	2.280	2.5	sys_clk_pin	sys_clk_pin		0.035
Path 27	-1.283	4	16	Inst_btn_deboun...ry_reg[1][4]/C	Inst_btn_deboun...ry_reg[1][14]/R	3.294	1.014	2.280	2.5	sys_clk_pin	sys_clk_pin		0.035

PATH REPORT


Path Properties
Path 21
Summary
Name
Path 21
Slack
-1.349ns
Source
Inst_btn_debounce/sig_cntrs_ary_reg[1][4]/C (rising edge-triggered cell FDRE clocked by sys_clk_pin {rise@0.000ns fall@1.250ns period=2.500ns})
Destination
Inst_btn_debounce/sig_cntrs_ary_reg[1][4]/R (rising edge-triggered cell FDRE clocked by sys_clk_pin {rise@0.000ns fall@1.250ns period=2.500ns})
Path Group
sys_clk_pin
Path Type
Setup (Max at Slow Process Corner)
Requirement
2.500ns (sys_clk_pin rise@2.500ns - sys_clk_pin rise@0.000ns)
Data Path Delay
3.290ns (logic 1.014ns (30.824%) route 2.276ns (69.176%))
Logic Levels
4 (LUT3=1 LUT4=1 LUT5=1 LUT6=1)
Clock Path Skew
0.000ns
Clock Un...rtainty
0.035ns
Source Clock Path
Delay Type
Incr (ns)
Path ...
Location
Netlist Resource(s)
(clock sys_cl...n rise edge)
(r) 0.000
0.000
(r) 0.000
0.000
Site: W5
CLK
net (fo=0)
0.000
0.000
CLK
IBUF (Prop_ibuf_I_O)
(r) 1.458
1.458
Site: W5
CLK_IBUF_inst/O
net (fo=1, routed)
1.967
3.425
CLK_IBUF
BUFG (Prop_bufg_I_O)
(r) 0.096
3.521
Site: BUF...TRL_X0Y1
CLK_IBUF_BUFG_inst/O
net (fo=249, routed)
1.564
5.085
Inst_btn_debounce/CLK_IBUF_BUFG
FDRE
Site: SLICE_X14Y13
Inst_btn_debounce/sig_cntrs_ary_reg[1][4]/C
Data Path
Delay Type
Incr (ns)
Path ...
Location
Netlist Resource(s)
FDRE (Prop_fdre_C_Q)
(r) 0.518
5.603
Site: SLICE_X14Y13
Inst_btn_debounce/sig_cntrs_ary_reg[1][4]/Q
net (fo=2, routed)
0.678
6.282
Inst_btn_debounce/sig_cntrs_ary_reg[1]_1[4]
LUT4 (Prop_lut4_I1_O)
(r) 0.124
6.406
Site: SLICE_X15Y13
Inst_btn_debounce/sig_out_reg[1]_i_5/O
net (fo=1, routed)
0.299
6.704
Inst_btn_debounce/sig_out_reg[1]_i_5_n_0
LUT5 (Prop_lut5_I4_O)
(f) 0.124
6.828
Site: SLICE_X13Y12
Inst_btn_debounce/sig_out_reg[1]_i_4/O
net (fo=1, routed)
0.154
6.982
Inst_btn_debounce/sig_out_reg[1]_i_4_n_0
LUT6 (Prop_lut6_I5_O)
(r) 0.124
7.106
Site: SLICE_X13Y12
Inst_btn_debounce/sig_out_reg[1]_i_2/O
net (fo=2, routed)
0.502
7.608
Inst_btn_debounce/sig_out_reg[1]_i_2_n_0
LUT3 (Prop_lut3_I0_O)
(r) 0.124
7.732
Site: SLICE_X15Y13
Inst_btn_debounce/sig_cntrs_ary[1][0]_i_1/O
net (fo=16, routed)
0.643
8.375
Inst_btn_debounce/sig_cntrs_ary[1][0]_i_1_n_0
FDRE
Site: SLICE_X14Y13
Inst_btn_debounce/sig_cntrs_ary_reg[1][4]/R
Arrival Time
8.375
Destination Clock Path
Delay Type
Incr (ns)
Path ...
Location
Netlist Resource(s)
General
Properties
Report
Cells
Nets
Net Segments
Options

FIXING TIMING

- If you fail timing:
  - Understand where is the problem
  - Change your design (reduce clock speed, add pipeline stages)
  - Run implementation (if timing issue arises from synthesis)
  - Optimise implementation (More in another lesson)

PHYSICAL CONSTRAINTS

- Physical Constrains refers to Netlist, IO, placement, Routing and Configurations
- Reminder of general form

```
set_property <property> <value> <object_list>
```

- Cells and nets are specified hierarchically

```
[get_cells top/mod1/mod2/net0]
```

- Consider using variables, if applying multiple property to a net/cell

```
set myCell [get_cells top/mod1/mod2/net0]
set_property <property> <value> \$myCell
```

MOST COMMON PHYSICAL CONSTRAINTS

- `MARK_DEBUG (TRUE/FALSE)`: Used in hardware Test (future lecture)
- `DONT_TOUCH (TRUE/FALSE)`: Tells Vivado to not optimise away or merge this particular net. Useful when duplicating signals
- `IOSTANDARD`: Set an I/O standard for an I/O port (e.g. `LVCMOS33`)
- `LOC`: Places a logical element from the netlist to a site on the device.
- Full list available here

LAB 17: TIMING CONSTRAINTS

LAB 18: GENERATING CLOCKS

The figures in these slides are taken from:

- Digital Design: Principles and Practices, Fourth Edition, John F. Wakerly, ISBN 0-13- 186389-4.

- allaboutfpga.com
- nandland.com
- docs.amd.com
- https://www.symmetryelectronics.com/
- https://www.edn.com/
- Stephen A. Edwards, Columbia University, Fundamentals of Computer Systems, Spring 2012
- https://medium.com/well-red/state-machines-for-everyone-part-1-introduction-b7ac9aaf482e
- www.icdesigntips.com
- techdocs.altium.com
- anysilicon.com
- Yngve Hafting 2021, University of Oslo