

Introduction to FPGA

Adam Taylor



Xilinx Tools and Frameworks

VitisAI

Enables the implementations of machine learning inference, using Tensor Flow, Caffe and PyTorch. *Requires a SoC/RFSoC/Alveo

Vitis

Embedded and accelerated SW development. Used to develop software solutions for MicroBlaze, Arm R5, A9, A53 and A72.

Petalinux

Embedded Linux solutions

PYNQ

Python framework for rapid prototyping on SoC/RFSoC/Alveo

Vitis / Vivado HLS

High Level Synthesis tool supporting C/C++/OpenCL

Vivado

Design Capture and implementation for the base platform



Session 1	Session 2	Session 3	Session 4
Introduction to	Processing in	Embedded	Accelerating
Xilinx FPGA	Xilinx FPGA	Linux	Solutions



What is an FPGA

A field-programmable gate array is an integrated circuit designed to be configured by a designer after manufacturing.

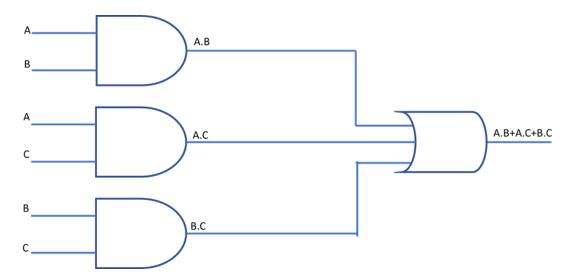
Field-Programmable – Means the user can program it in the field



FPGA Architecture



What is programmable logic



Input A	Input B	Input C	ОР
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

VCC 111 VCC 110 VCC 101 GND 100 VCC 011 GND 010 GND 001 GND 000 В С А

OP



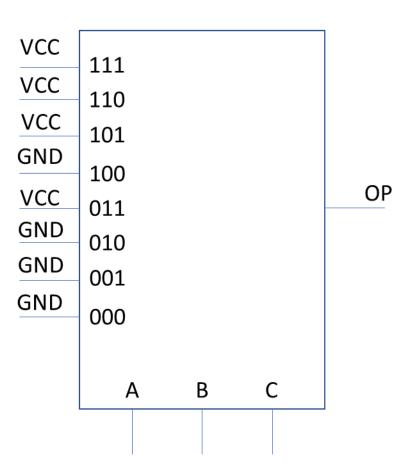
Look Up Table

- The key to programmable logic is how logic equations are implemented in the device.
- It is of course very difficult for the device designers to include a range of AND, OR gates etc. as the number of each gate type will vary for each application
- Device manufacturers addressed this challenge in a very smart manner. In place of discrete gates, they used they use a several input look up table (LUT) which is programmed to implement the combinatorial logic equations.



Look Up Table

Input A	Input B	Input C	ОР
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1



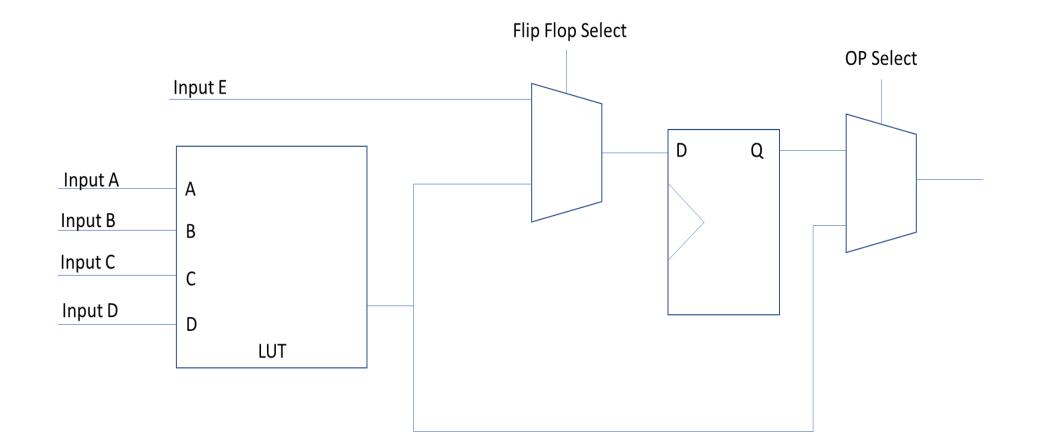


Configurable Logic Block

- LUT allows implementation of Combinatorial Circuit
- BUT we design synchronous circuits what about the FF
- flip flop to act as storage for the combinatorial output such that we can implement sequential structures
- This combination of a LUT and Flip Flop is often called a Configurable Logic Block (CLB) and a programmable logic device will consist of many thousands of these CLBs. To provide the most flexibility additional multiplexers will be used to support a wide range of CLB configurations.



Configurable Logic Block



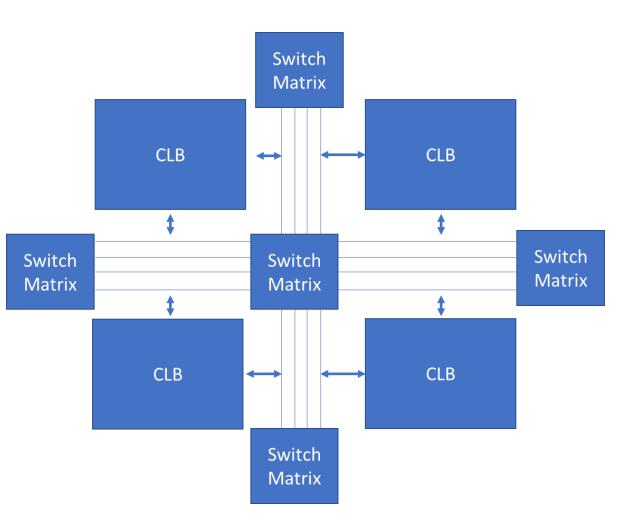


Interconnect Routing

A collection of wires and programmable switches.

These are responsible for connecting CLBs and other building blocks within the FPGA.

These are also called routing channels.





Tools

- Requirements Capture tool
- Editors Ideal with ability to LINT and check structural issues e.g HDL Creator
- Synthesis Tool Third Party or Vendor Supplied
- Implementation Vendor supplied some open source for lattice
- Simulation Third Party or Vendor
- Source Control Subversion, GIT
- Configuration control often called PLM tool
- Specialist e.g. Fault Injection



Vivado Overview

- Foundation of all design and higherlevel tools is Vivado
- Vivado enables us to capture designs using VHDL or Verilog
- Large IP library to accelerate our designs
- Integrates Vivado / Vitis HLS IP cores

i

- Simulate designs using Vivado Simulator
- Synthesize, place and route the design
- Generate power estimations
- Create Xilinx Support Architecture



Vivado Overview

FPGA Implementation Flow

Synthesis – Translates the HDL design into a series of logic equations which are then mapped onto the resources available in the target FPGA.

Synthesis – Place – The logic resources determined by the synthesis tool are placed at available locations within the target device.

Routing – The placed logic resources in the design are interconnected using routing and switch matrixes to implement the final application.

Bit File – The generation of the final programming file for the target FPGA.

i

We can control the flow implementation settings by using Constraints (XDC file) and implementation Strategies



Xilinx Devices





Xilinx Offer a range of devices

FPGA – Spartan, Artix, Kintex, Virtex – Seven / UltraScale / UltraScale+

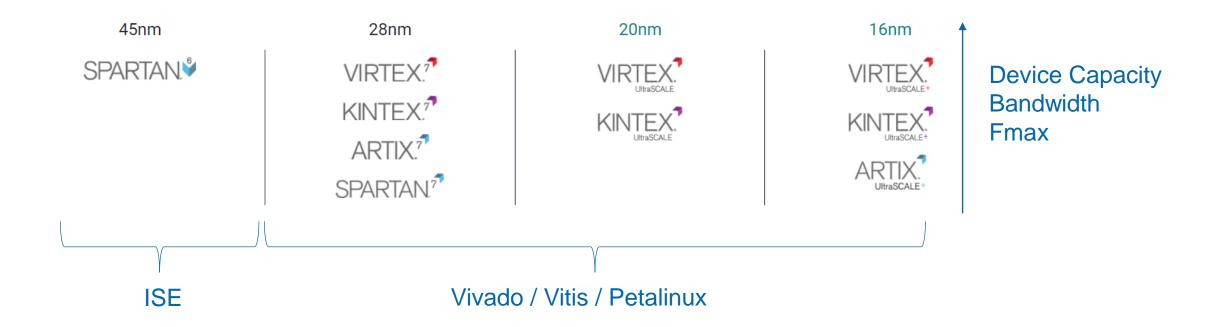
Heterogeneous SoC – Zynq, Zynq MPSoC, RFSoC

Adaptive Compute Acceleration Platform– Versal

Accelerator Card - Alveo



FPGA





Heterogeneous SOC

Combine diverse processing elements with programmable logic

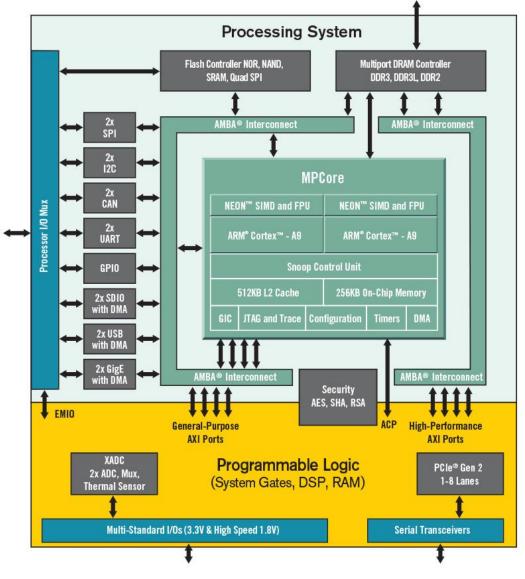
Processing elements are silicon implementations

Processing element is the master – it boots just like any other processor

Enables highly optimized solutions be implemented using processor and logic to exploit natural strengths

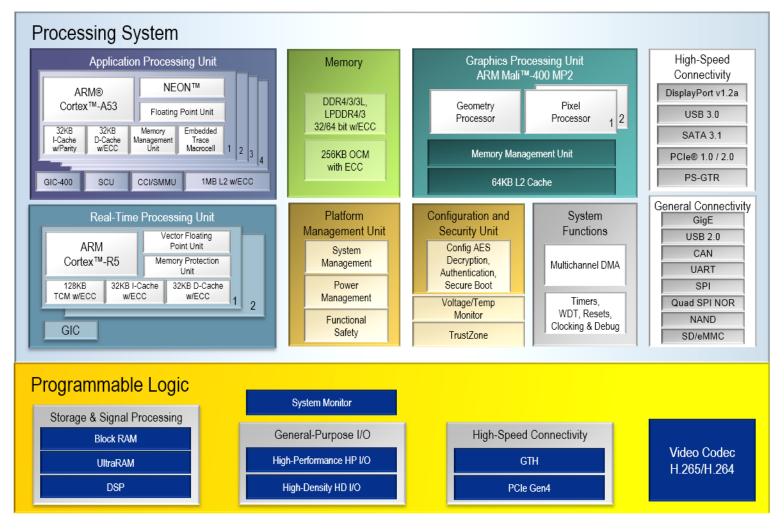


Zynq



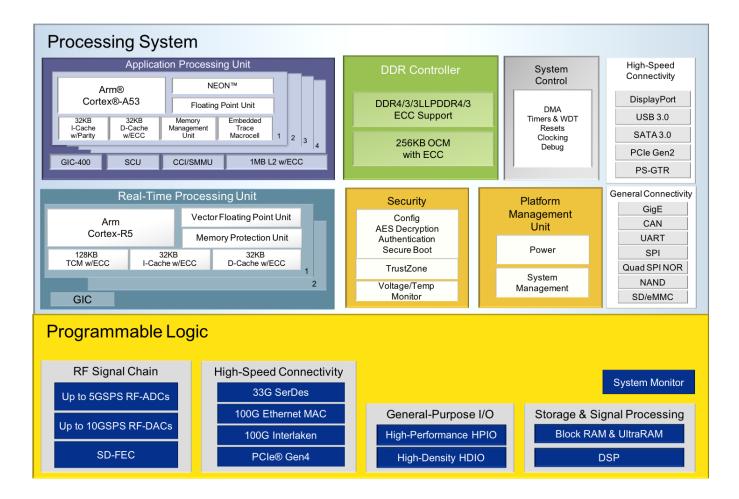


Zynq MPSoC



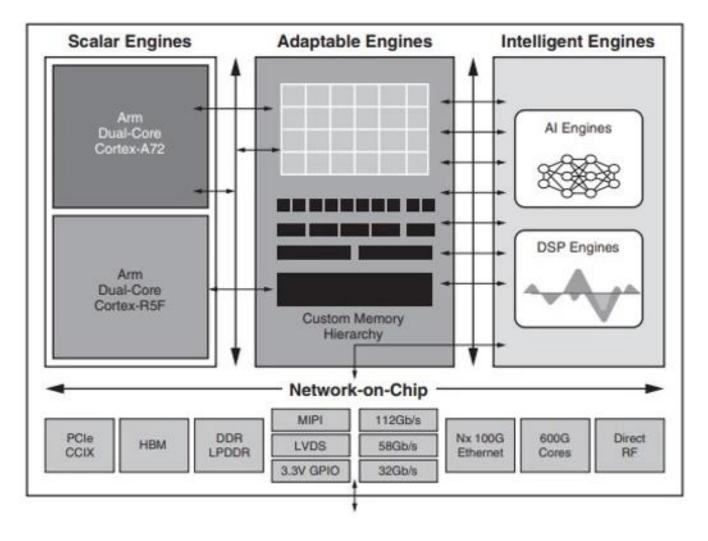


Zynq RFSoC





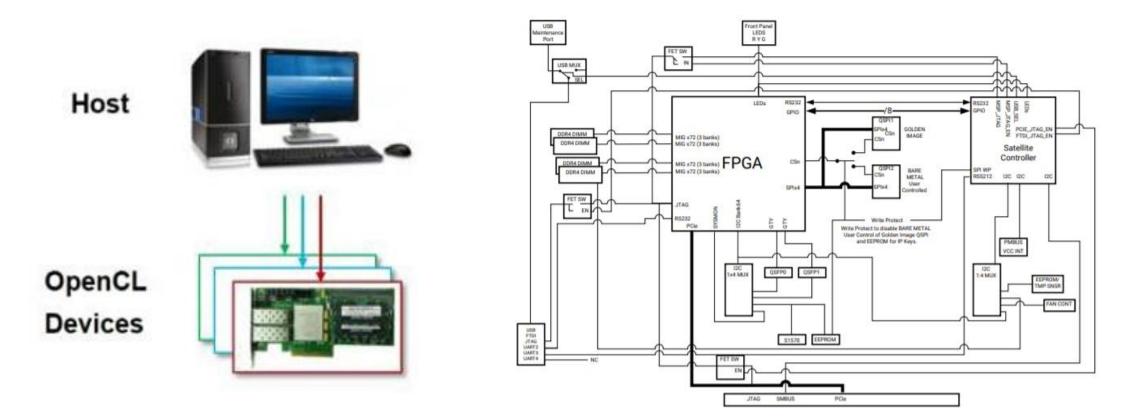
Versal





Alveo

Accelerator card based around OpenCL





Clocking



24



Clocking

FPGA Designs are Synchronous!

Clocks are high fan out, as such dedicated pins are required to be used.

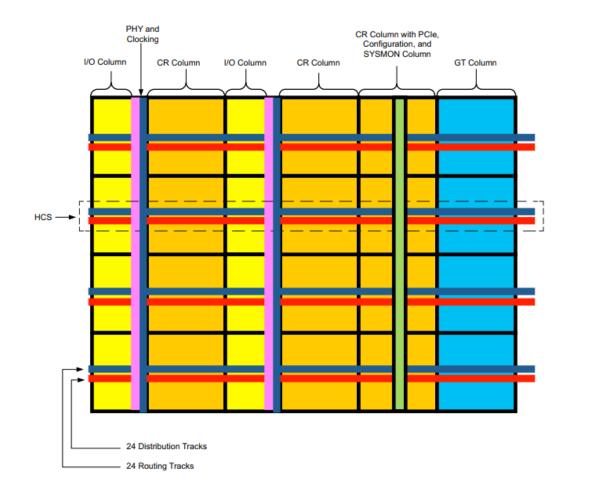
FPGA typically will have multiple clock regions

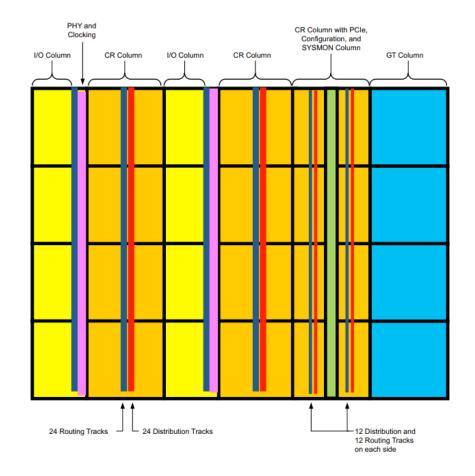
Clock pins are places in IO banks

- UltraScale / UltraScale+ GC pins or global clock pins
- Seven Series CC and GC pins GC global clocks CC restricted to close CR
- Special Clock pins e.g. Byte-Lane Clocks (DBC and QBC) typical in memory applications



Vertical and Horizontal Clocks



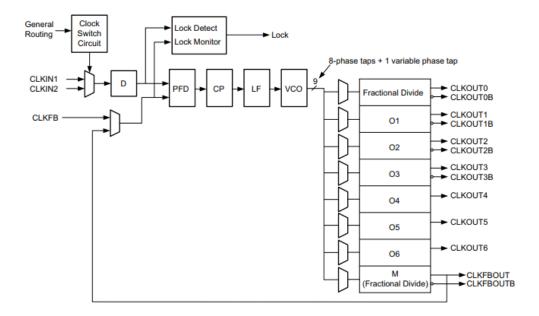




Clocking Resources

FPGA have a range of clock resources to simply solutions

- 1. Clock Buffers e.g. bufgctl
- 2. Clock Management Tiles
 - Mixed Mode Clock Manager (MMCM) & 2 Phase Locked Loops





Clocking Resources

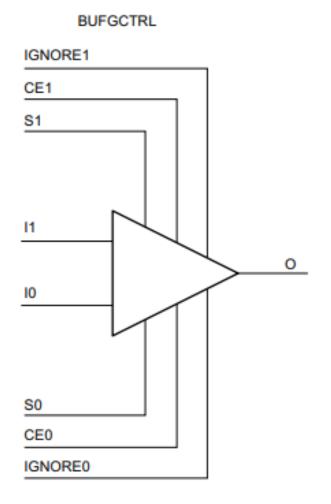
Bufgctl - drive the routing and distribution resources across the entire device.

Can switch between 2 clocks seamlessly.

Basis of many structures including

Bufgce_1, Bufgmux etc.

Read Seven Series, UltraScale clocking guides





What is a Clock Domain?

Same clock domain if:

» Same source and integer multiple frequencies

Different clock domain if:

- » Different source
 - Even if frequency specification is the same!
 - All specifications have error bars
- » Not integer multiple frequencies, even if same source

By default, Vivado assumes all clocks are in the same domain – Unless you tell it otherwise! Not telling it clock relationships can make a significantly longer implementation times as Vivado tries to close timing which is impossible



Clock Domain Crossing

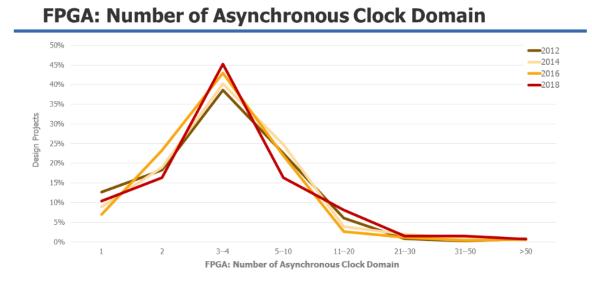




Clock Domain Crossing

Ideal solution uses one clock and the entire design is synchronous.

BUT!



Modern devices have multiple clocks to address different clock domains e.g. ADC / DAC clocks, source synchronous interfaces.

Brings with it the need to transfer data, and signals safely and reliability between the clock domains



Metastability

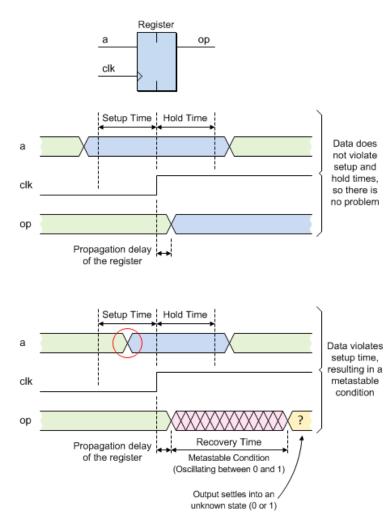
One issue which can arise with incorrect domain crossing is metastability

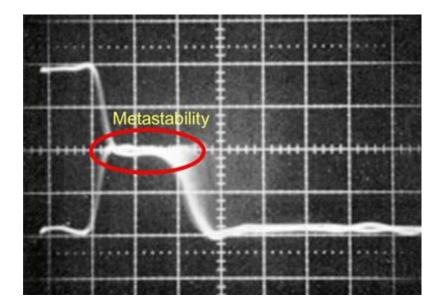
This can lead to corruption of data or incorrect behaviour

Occurs when a flip flops set up or hold time is violated



Metastability







Clock Domain Crossing

Several Techniques which can be used depending upon what needs to be transferred

- Two stage synchroniser Ideal for single bit data
- Grey Code Synchroniser Encodes data bus in grey code and transfer between domains – Ideal for counters as input to be converted to grey code can only decrement / increment by one from previous value
- Hand shake synchroniser Transfers data bus between two clock domains using handshake signals
- Pulse synchroniser transfer pulse from one clock domain to another
- Asynchronous FIFO transfers data from one domain to another, useful for high throughput / burst transfers



Clock Domain Crossing

To support reset functionality across clock domains we may need the following synchronisers structures.

- » Asynchronous Reset Synchroniser enables asynchronous assertion and synchronous de-assertion.
- » Synchronous Reset Synchroniser synchronises a synchronous reset to another clock domain.



How many synchronisers do I need?

Standard two stage synchroniser assumes the first flip flop, will recover from metastability before the signal is clocked into the second flip flop. How do we know this is the case and what if we need more stages

$$MTBF = \frac{e^{t_r/\tau}}{T_o \times f \times a} \times k \left(\frac{e^{t_r/\tau}}{T_o \times f}\right)$$

- t_r = The speed at which the event is being resolved (clock period – setup time)
- T_o = Time window during which the register is susceptible to going metastable.
- $\tau =$ Settling time
- a = Frequency of the asynchronous signal.
- f = Frequency of the clock (as stated in Part 1, this should be faster than "a")
- k = The number of additional stages (so k=1 for a two-stage synchroniser)

Note that τ and T_o are both process constants.

© Adiuvo Engineering and Training, Ltd. 2020



CDC in Xilinx

Xilinx Parameterised Macros (XPM) – provide CDC structures

Use registers optimised for CDC in the fabric

» Registers located close together and have small set up and hold windows

Described within UG953 Libraries Guide

Described within UG 974 UltraScale Architecture Libraries

Untitled 1											_ 8 2
Q 📕 🔍 Q 🐰	•F I4 M 12 27	+ Г Г ∉ ⇒ Г ⊡									
											1,965.000 ns
Name	Value		200 ns	400 ns	600 ns	800 ns	1,000 ns	1,200 ns	1,400 ns	1,600 ns	1,800 ns
🕌 clk_1	1			<u>Ta bada da bada da bada da ba</u>							
🕌 clk_2	0										
👪 reset	0										
reset_safe	0										
🕌 start	0										
<pre>& start_retimed</pre>	0										
> 😼 start_reg[1:0]	0	x		0		3			0		
> 🕫 counter[15:0]	100	x									100
H pulse	1	1									
🖟 terminal_count	100					10)				



CDC Design Analysis tools

Detecting all CDC issues can be a challenge in large designs

- » Are all IP IO on the correct clock domain
- » Very easy to associate signal with wrong domain e.g. FIFO empty and WR clock

CDC issues can be very difficult to find changing on each start up and may be intermittent

Can be hard to find in simulation – Timing simulation required, takes a long time simulate the system

Static analysis tools are better suited to find the CDC issues.



Vivado CDC Report

Following Synthesis – in TCL window run the command report_cdc

onsole		? = 1
DC Report		
D Severity Count Description		
DC-1 Critical 66 1-bit unknown CDC circuitry		
DC-2 Warning 1 1-bit synchronized with missing ASYNC_REG property DC-3 Info 2 1-bit synchronized with ASYNC REG property		
DC-9 Info 2 Asynchronous reset synchronized with ASYNC_REG property DC-15 Warning 32 Clock enable controlled CDC structure detected		
ource Clock: input port clock		
stination Clock: sl_clk		
DC Type: No Common Primary Clock		
	Exception Source (From) Destination (To)	
1 CDC-9 Info Asynchronous reset synchronized with ASYNC_REG property 2	False Path reset_n UF/U0/inst_fifo_gen/gconvfifo.rf/grf.rf/rstblk/ngwrdrst.grst.g7serrst.gnsckt_wrst.rs	t_wr_reg2_inst/arststages_ff_reg[0]/PRE
ource Clock: s2_clk		
estination Clock: sl_clk DC Type: No Common Primary Clock		
		Destination (To)
ow ID Severity Description Depth Exception	Source (From)	Descinación (10)
1 CDC-3 Info 1-bit synchronized with ASYNC_REG property 5 False Path 1 2 CDC-1 Critical 1-bit unknown CDC circuity 0 None 1	UF/U0/inst_fifo_gen/gconvfifo.rf/grf.rf/rstblk/ngwrdrst.grst.g79errst.gnsckt_wrst.gic_rst.sckt_rd_rst_ic_reg/C UE/U0/inst_fifo_gen/gconvfifo.rf/grf.rf/gntv_or_sync_fifo.gl0.rd/gras.rsts/ram_empty_i_reg/C	JF/U0/inst_fifo_gen/gconvfifo.rf/grf.rf/rstblk/ngwrdrst.grst.g7 reqdata_reg/CE
1 CDC-3 Info 1-bit synchronized with ASYNC_REG property 5 False Path 1 2 CDC-1 Critical 1-bit unknown CDC circuity 0 None 1	UF/V0/inst_fifo_gen/gconvfifo.rf/grf.rf/rstblk/ngwrdrst.grst.grserrst.gnsckt_wrst.gic_rst.sckt_rd_rst_ic_reg/C UF/V0/inst_fifo_gen/gconvfifo.rf/grf.rf/gntv_or_sync_fifo.q10.rd/gras.rsts/ram_empty_i_reg/C	JF/U0/inst_fifo_gen/gconvfifo.rf/grf.rf/rstblk/ngwrdrst.grst.g7
1 CDC-3 Info 1-bit synchronized with ASYNC_REG property 5 False Path 1 2 CDC-1 Critical 1-bit unknown CDC circuitry 0 None 1 3 CDC-1 Critical 1-bit unknown CDC circuitry 0 None 1 ource Clock: input port clock	UF/U0/inst_fifo_gen/gconvfifo.rf/grf.rf/rstblk/ngwrdrst.grst.g79errst.gnsckt_wrst.gic_rst.sckt_rd_rst_ic_reg/C UE/U0/inst_fifo_gen/gconvfifo.rf/grf.rf/gntv_or_sync_fifo.gl0.rd/gras.rsts/ram_empty_i_reg/C	JF/U0/inst_fifo_gen/gconvfifo.rf/grf.rf/rstblk/ngwrdrst.grst.g7 reqdata_reg/CE
1 CDC-3 Info 1-bit synchronized with ASYNC_REG property 5 False Path 2 CDC-1 Critical 1-bit unknown CDC circuitry 0 None 1 3 CDC-1 Critical 1-bit unknown CDC circuitry 0 None 1	UF/U0/inst_fifo_gen/gconvfifo.rf/grf.rf/rstblk/ngwrdrst.grst.g79errst.gnsckt_wrst.gic_rst.sckt_rd_rst_ic_reg/C UE/U0/inst_fifo_gen/gconvfifo.rf/grf.rf/gntv_or_sync_fifo.gl0.rd/gras.rsts/ram_empty_i_reg/C	JF/U0/inst_fifo_gen/gconvfifo.rf/grf.rf/rstblk/ngwrdrst.grst.g7 reqdata_reg/CE
1 CDC-3 Info 1-bit synchronised with ASYNC_REG property 5 False Path 1 2 CDC-1 Critical 1-bit unknown CDC circuitry 0 None 1 3 CDC-1 Critical 1-bit unknown CDC circuitry 0 None 1 ource Clock: input port clock evination toost 32 circ DC Type: No Common Primary Clock ow ID Severity Description Depth	UF/U0/inst_fifo_gen/gconvfifo.rf/grf.rf/rstblk/ngwrdrst.grst.grserrst.gnsckt_wrst.gic_rst.sckt_rd_rst_io_reg/C UE/U0/inst_fifo_gen/gconvfifo.rf/grf.rf/gntv_or_sync_fifo.gl0.rd/gras.rsts/ram_empty_i_reg/C UF/U0/inst_fifo_gen/gconvfifo.rf/grf.rf/gntv_or_sync_fifo.gl0.rd/gras.rsts/ram_empty_i_reg/C Exception Source (From) Destination (To)	UF/U0/inst_fifo_gen/goonvfifo.rf/grf.rf/rstblk/ngwrdrst.grst.g7 cegdata_reg/CE cegdata_reg/D
1 CDC-3 Info 1-bit synchronised with ASYNC_REG property 5 False Path 1 2 CDC-1 Critical 1-bit unknown CDC circuitry 0 None 1 3 CDC-1 Critical 1-bit unknown CDC circuitry 0 None 1 ource Clock: input port clock evination toost 32 circ DC Type: No Common Primary Clock ow ID Severity Description Depth	UF/U0/inst_fifo_gen/gconvfifo.rf/grf.rf/rstblk/ngwrdrst.grst.g79errst.gnsckt_wrst.gic_rst.sckt_rd_rst_ic_reg/C UE/U0/inst_fifo_gen/gconvfifo.rf/grf.rf/gntv_or_sync_fifo.gl0.rd/gras.rsts/ram_empty_i_reg/C UF/U0/inst_fifo_gen/gconvfifo.rf/grf.rf/gntv_or_sync_fifo.gl0.rd/gras.rsts/ram_empty_i_reg/C ExceptionSource (From)_Destination (To)	DF/U0/inst_fifo_gen/goonvfifo.rf/grf.rf/rstblk/ngwrdrst.grst.g7 regdata_reg/CE regdata_reg/D
1 CDC-3 Info 1-bit synchronized with ASYNC_REG property 5 False Path 2 CDC-1 Critical 1-bit unknown CDC circuitry 0 None 1 3 CDC-1 Critical 1-bit unknown CDC circuitry 0 None 1 ource Clock: input port clock 0 None 1 D'Type: NC Common Frimary Clock 0 Depth 1 Depth 1 CDC-9 Info Asynchronous reset synchronized with ASYNC_REG property 2	UF/U0/inst_fifo_gen/goonvfifo.rf/grf.rf/rstblk/ngwrdrst.grst.g7serrst.gnsckt_wrst.gio_rst.sckt_rd_rst_io_reg/C UE/U0/inst_fifo_gen/goonvfifo.rf/grf.rf/gntv_or_sync_fifo.gl0.rd/gras.rsts/ram_empty_i_reg/C UF/U0/inst_fifo_gen/goonvfifo.rf/grf.rf/gntv_or_sync_fifo.gl0.rd/gras.rsts/ram_empty_i_reg/C Exception Source (From) Destination (To)	DF/U0/inst_fifo_gen/goonvfifo.rf/grf.rf/rstblk/ngwrdrst.grst.g7 regdata_reg/CE regdata_reg/D
1 CDC-3 Info 1-bit synchronized with ASYNC_REG property 5 False Path 1 2 CDC-1 Critical 1-bit unknown CDC circuitry 0 None 1 3 CDC-1 Critical 1-bit unknown CDC circuitry 0 None 1 ource Clock: input port clock setination croox: s2_clx Description	UF/U0/inst_fifo_gen/goonvfifo.rf/grf.rf/rstblk/ngwrdrst.grst.g7serrst.gnsckt_wrst.gio_rst.sckt_rd_rst_io_reg/C UE/U0/inst_fifo_gen/goonvfifo.rf/grf.rf/gntv_or_sync_fifo.gl0.rd/gras.rsts/ram_empty_i_reg/C UF/U0/inst_fifo_gen/goonvfifo.rf/grf.rf/gntv_or_sync_fifo.gl0.rd/gras.rsts/ram_empty_i_reg/C Exception Source (From) Destination (To)	DF/U0/inst_fifo_gen/goonvfifo.rf/grf.rf/rstblk/ngwrdrst.grst.g7 regdata_reg/CE regdata_reg/D
1 CDC-3 Info 1-bit synchronized with ASYNC_REG property 5 False Path 1 2 CDC-1 Critical 1-bit unknown CDC circuitry 0 None 1 3 CDC-1 Critical 1-bit unknown CDC circuitry 0 None 1 ource Clock: input port clock Setimation Clock # 2 fix DC Type: No Common Primary Clock ow ID Severity Description Depth 1 CDC-9 Info Asynchronous reset synchronized with ASYNC_REG property 2 ource Clock: s1_clk estimation Clock: s2_clk DC Type: No Common Primary Clock	UF/U0/inst_fifo_gen/gconvfifo.rf/grf.rf/rstblk/ngwrdrst.grst.grserrst.gnsckt_wrst.gic_rst.sckt_rd_rst_io_reg/C UE/U0/inst_fifo_gen/gconvfifo.rf/grf.rf/gntv_or_sync_fifo.gl0.rd/gras.rsts/ram_empty_i_reg/C UF/U0/inst_fifo_gen/gconvfifo.rf/grf.rf/gntv_or_sync_fifo.gl0.rd/gras.rsts/ram_empty_i_reg/C Exception Source (From) Destination (To) 	UF/U0/inst_fifo_gen/geonvfifo.rf/grf.rf/rstblk/ngwrdrst.grst.g7 cegdata_reg//E cegdata_reg/D
1 CDC-3 Info 1-bit synchronized with ASYNC_REG property 5 False Path 1 2 CDC-1 Critical 1-bit unknown CDC circuitry 0 None 1 3 CDC-1 Critical 1-bit unknown CDC circuitry 0 None 1 ource Clock: input port clock Setimation Clock # 2 fix DC Type: No Common Primary Clock ow ID Severity Description Depth 1 CDC-9 Info Asynchronous reset synchronized with ASYNC_REG property 2 ource Clock: s1_clk estimation Clock: s2_clk DC Type: No Common Primary Clock	UF/U0/inst_fifo_gen/gconvfifo.rf/grf.rf/rstblk/ngwrdrst.grst.g7serrst.gnsckt_wrst.gic_rst.sckt_rd_rst_ic_reg/C UE/U0/inst_fifo_gen/gconvfifo.rf/grf.rf/gntv_or_sync_fifo.gl0.rd/gras.rsts/ram_empty_i_reg/C UF/U0/inst_fifo_gen/gconvfifo.rf/grf.rf/gntv_or_sync_fifo.gl0.rd/gras.rsts/ram_empty_i_reg/C Exception Source (From) Destination (To)	DF/U0/inst_fifo_gen/goonvfifo.rf/grf.rf/rstblk/ngwrdrst.grst.g7 regdata_reg/CE regdata_reg/D
1 CDC-3 Info 1-bit synchronized with ASYNC_REG property 5 False Path 1 2 CDC-1 Critical 1-bit unknown CDC circuitry 0 None 1 3 CDC-1 Critical 1-bit unknown CDC circuitry 0 None 1 ource Clock: input port clock estimation croox: s2_clx DC Type: No Common Primary Clock ow ID Severity Description Depth 	UF/U0/inst_fifo_gen/gconvfifo.rf/grf.rf/rstblk/ngwrdrst.grst.g7serrst.gnsckt_wrst.gic_rst.sckt_rd_rst_ic_reg/C UF/U0/inst_fifo_gen/gconvfifo.rf/grf.rf/gntv_or_sync_fifo.gl0.rd/gras.rsts/ram_empty_i_reg/C UF/U0/inst_fifo_gen/gconvfifo.rf/grf.rf/gntv_or_sync_fifo.gl0.rd/gras.rsts/ram_empty_i_reg/C Exception Source (From) Destination (To) 	UF/U0/inst_fifo_gen/goonvfifo.rf/grf.rf/rstblk/ngwrdrst.grst.g7 reqdata_reg/CE reqdata_reg/D c_rst.rst_rd_reg2_inst/arststages_ff_reg[0]/PRE Destination (To) c_reg/C UF/U0/inst_fifo_gen/goonvfifo.rf/grf.rf/rstblk/ngwrdrs
1 CDC-3 Info 1-bit synchronized with ASYNC_REG property 5 False Path 2 CDC-1 Critical 1-bit unknown CDC circuitry 0 None 1 3 CDC-1 Critical 1-bit unknown CDC circuitry 0 None 1 ource Clock: input port clock setimation Cucker sg_cik DE Type: No Common Primary Clock ow ID Severity Description	UF/U0/inst_fife_gen/gconvfifo.rf/grf.rf/rstblk/ngwrdrst.grst.g7serrst.gnsckt_wrst.gic_rst.sckt_rd_rst_ic_reg/C UF/U0/inst_fife_gen/gconvfifo.rf/grf.rf/gntv_or_sync_fifo.gl0.rd/gras.rsts/ram_empty_i_reg/C UF/U0/inst_fife_gen/gconvfifo.rf/grf.rf/gntv_or_sync_fifo.gl0.rd/gras.rsts/ram_empty_i_reg/C Exception Source (From) Destination (To) 	<pre>UF/U0/inst_fifo_gen/gconvfifo.rf/grf.rf/rstblk/ngwrdrst.grst.g7 reqdata_reg//E reqdata_reg//D</pre>
1 CDC-3 Info 1-bit synchronized with ASYNC_REG property 5 False Path 1 2 CDC-1 Critical 1-bit unknown CDC circuitry 0 None 0 3 CDC-1 Critical 1-bit unknown CDC circuitry 0 None 0 ource Clock: input port clock 0 0 estimation Cucket signature 0 0 0 D Severity Description Depth 1 CDC-9 Info Asynchronous reset synchronized with ASYNC_REG property 2 0 curce Clock: sl_clk 0 Estimation Clock is g_clk 0 C Type: No Common Primary Clock 0 0 Depth Estimation Clock: sl_clk C Type: No Common Primary Clock 0 0 Depth Estimation Clock: sl_clk 0 C Type: No Common Primary Clock 0 1 Depth Estimation Clock: sl_clk 0 C Type: No Common Primary Clock 0 1 Depth Estimation Clock: sl_clk 0 2 CDC-1 Critical 1-bit unknown CDC circuitry 0 0 None 0 None 2 CDC-1 Critical 1-bit unknown CDC circuitry 0 None 0 None 0	UF/U0/inst_fifo_gen/gconvfifo.rf/grf.rf/rstblk/ngwrdrst.grst.g7serrst.gnsckt_wrst.gic_rst.sckt_rd_rst_io_reg/C UF/U0/inst_fifo_gen/gconvfifo.rf/grf.rf/gntv_or_sync_fifo.gl0.rd/gras.rsts/ram_empty_i_reg/C UF/U0/inst_fifo_gen/gconvfifo.rf/grf.rf/gntv_or_sync_fifo.gl0.rd/gras.rsts/ram_empty_i_reg/C Exception Source (From) Destination (To) 	UF/U0/inst_fifo_gen/goonvfifo.rf/grf.rf/rstblk/ngwrdrst.grst.g7 reqdata_reg/CE reqdata_reg/D c_rst.rst_rd_reg2_inst/arststages_ff_reg[0]/PRE Destination (To) c_reg/C UF/U0/inst_fifo_gen/goonvfifo.rf/grf.rf/rstblk/ngwrdrs
1 CDC-3 Info 1-bit synchronized with ASYNC_REG property 5 False Path 1 2 CDC-1 Critical 1-bit unknown CDC circuitry 0 None 1 3 CDC-1 Critical 1-bit unknown CDC circuitry 0 None 1 ource Clock: input port clock Setimation Clock # 2 CH DC Type: No Common Primary Clock ow ID Severity Description Depth 1 CDC-9 Info Asynchronous reset synchronized with ASYNC_REG property 2 ource Clock: s1_clk estimation Clock: s2_clk DC Type: No Common Primary Clock ow ID Severity Description Depth to Clock: s1_clk estimation Clock: s2_clk DC Type: No Common Primary Clock ow ID Severity Description Depth Exc 1 CDC-3 Info 1-bit synchronized with ASYNC_REG property 6 Fal 2 CDC-1 Critical 1-bit unknown CDC circuitry 0 Non 3 CDC-1 Critical 1-bit unknown CDC circuitry 0 Non	UF/U0/inst_fifo_gen/gconvfifo.rf/grf.rf/rstblk/ngwrdrst.grst.g7serrst.gnsckt_wrst.gic_rst.sckt_rd_rst_io_reg/C UF/U0/inst_fifo_gen/gconvfifo.rf/grf.rf/gntv_or_sync_fifo.gl0.rd/gras.rsts/ram_empty_i_reg/C UF/U0/inst_fifo_gen/gconvfifo.rf/grf.rf/gntv_or_sync_fifo.gl0.rd/gras.rsts/ram_empty_i_reg/C Exception Source (From) Destination (To) False Path reset_n UF/U0/inst_fifo_gen/gconvfifo.rf/grf.rf/rstblk/ngwrdrst.grst.g7serrst.gnsckt_wrst.gl seption Source (From) Exception Source (From) Source (from) Second Source (from) ise Path UF/U0/inst_fifo_gen/gconvfifo.rf/grf.rf/rstblk/ngwrdrst.grst.g7serrst.gnsckt_wrst.gic_rst.sckt_wr_rst_i data_sl_rl_reg1[6]/C data_sl_rl_reg1[6]/C data_sl_rl_reg1[6]/C	UF/U0/inst_fifo_gen/goonvfifo.rf/grf.rf/rstblk/ngwrdrst.grst.g7 cegdata_reg//E cegdata_reg//E c_rst.rst_rd_reg2_inst/arststages_ff_reg[0]/PRE Destination (To) c_reg/C UF/U0/inst_fifo_gen/goonvfifo.rf/grf.rf/rstblk/ngwrdrs multresult_reg[1]/D multresult_reg[2]/D multresult_reg[3]/D
1 CDC-3 Info 1-bit synchronized with ASYNC_REG property 5 False Path 1 2 CDC-1 Critical 1-bit unknown CDC circuitry 0 None 0 3 CDC-1 Critical 1-bit unknown CDC circuitry 0 None 0 ource Clock: input port clock 0 None 0 SetInation Clock is 0 None 0 0 D Severity Description Depth Depth 1 CDC-9 Info Asynchronous reset synchronized with ASYNC_REG property 2 ource Clock: sl_clk Estimation Clock isclk Depth DC Type: No Common Frimary Clock 0 None 1 Ource Clock: sl_clk Estimation Clock is_clk 0 None 2 Ource Clock: sl_clk Estimation Clock is_clk 0 None 1 Ource Clock: sl_clk Estimation Clock is_clk 0 None 1 Ource Clock: sl_clk Estimation Clock is_clk 0 None 1 0 None 1 Ource Clock: sl_clk Estimation Clock is_clk 0 None 1 0 None 1 0 None 1 0 None 1	UP/U0/inst_fifo_gen/gconvfifo.rf/grf.rf/rsblk/ngwrdrst.grst.g7serrst.gnackt_wrst.gic_rst.sckt_id_rst_ic_reg/C UF/U0/inst_fifo_gen/gconvfifo.rf/grf.rf/gntv_or_sync_fifo.gl0.rd/gras.rsts/ram_empty_i_reg/C UF/U0/inst_fifo_gen/gconvfifo.rf/grf.rf/gntv_or_sync_fifo.gl0.rd/gras.rsts/ram_empty_i_reg/C Exception Source (From) Destination (To) 	<pre>UF/U0/inst_fifo_gen/gconvfifo.rf/grf.rf/rstblk/ngwrdrst.grst.g7 reqdata_reg/CE reqdata_reg/D c_rst.rst_rd_reg2_inst/arststages_ff_reg[0]/PRE</pre>
1 CDC-3 Info 1-bit synchronized with ASYNC_REG property 5 False Path 1 2 CDC-1 Critical 1-bit unknown CDC circuitry 0 None 0 None 3 CDC-1 Critical 1-bit unknown CDC circuitry 0 None 0 None ource Clock: input port clock	<pre>UPT/UP/inst_fife_gen/gconvfife.rf/rstblk/ngwrdrst.grst.g7serrst.gnsckt_wrst.gic_rst.sckt_rd_rst_ic_reg/C UP/UD/inst_fife_gen/gconvfife.rf/grf.rf/gntv_or_sync_fife.gl0.rd/gras.rsts/ram_empty_i_reg/C UP/UD/inst_fife_gen/gconvfife.rf/grf.rf/gntv_or_sync_fife.gl0.rd/gras.rsts/ram_empty_i_reg/C UP/UD/inst_fife_gen/gconvfife.rf/grf.rf/gntv_or_sync_fife.gl0.rd/gras.rsts/ram_empty_i_reg/C Exception Source (From) Destination (To) </pre>	<pre>UF/U0/inst_fifo_gen/gconvfifo.rf/grf.rf/rstblk/ngwrdrst.grst.g7 ceqdata_reg//E ceqdata_reg//E rst.rst_rd_reg2_inst/arststages_ff_reg[0]/PRE rst.rst_rd_reg2_inst/arststages_ff_reg[0]/PRE reg/(C UF/U0/inst_fifo_gen/gconvfifo.rf/grf.rf/rstblk/ngwrdrs multresult_reg[0]/D multresult_reg[1]/D multresult_reg[3]/D multresult_reg[3]/D multresult_reg[3]/D multresult_reg[3]/D</pre>
1 CDC-3 Info 1-bit synchronized with ASYNC_REG property 5 False Path 1 2 CDC-1 Critical 1-bit unknown CDC circuitry 0 None 0 3 CDC-1 Critical 1-bit unknown CDC circuitry 0 None 0 ource Clock: input port clock 0 None 0 SetInation Clock is 0 None 0 0 D Severity Description Depth Depth 1 CDC-9 Info Asynchronous reset synchronized with ASYNC_REG property 2 ource Clock: sl_clk Estimation Clock isclk Depth DC Type: No Common Frimary Clock 0 None 1 Ource Clock: sl_clk Estimation Clock is_clk 0 None 2 Ource Clock: sl_clk Estimation Clock is_clk 0 None 1 Ource Clock: sl_clk Estimation Clock is_clk 0 None 1 Ource Clock: sl_clk Estimation Clock is_clk 0 None 1 0 None 1 Ource Clock: sl_clk Estimation Clock is_clk 0 None 1 0 None 1 0 None 1 0 None 1	UPU/UV/inst_fifo_gen/gconvfifo.rf/grf.rf/rstblk/ngwrdrst.grst.g7serrst.gnackt_wrst.gic_rst.sckt_rd_rst_ic_reg/C UP/UV/inst_fifo_gen/gconvfifo.rf/grf.rf/gntv_or_sync_fifo.gl0.rd/gras.rsts/ram_empty_i_reg/C UPU/UV/inst_fifo_gen/gconvfifo.rf/grf.rf/gntv_or_sync_fifo.gl0.rd/gras.rsts/ram_empty_i_reg/C Exception Source (From) Destination (To) 	<pre>UF/U0/inst_fifo_gen/gconvfifo.rf/grf.rf/rstblk/ngwrdrst.grst.g7 reqdata_reg/CE reqdata_reg/D c_rst.rst_rd_reg2_inst/arststages_ff_reg[0]/PRE</pre>
1 CDC-3 Info 1-bit synchronized with ASYNC_REG property 5 False Path 1 2 CDC-1 Critical 1-bit unknown CDC circuitry 0 None 3 CDC-1 Critical 1-bit unknown CDC circuitry 0 None 0 ource Clock: input port clock 0 None 0 SCDC-1 Critical 1-bit unknown CDC circuitry 0 None 0 ource Clock: input port clock 0 None 0 DC Type: No Common Frimary Clock 0 Depth Depth	UF/U0/inst_fifo_gen/gconvfifo.rf/grf.rf/stblk/ngwrdrst.grst.g7serrst.gnsckt_wrst.gic_rst.sckt_rd_rst_io_reg/C UF/U0/inst_fifo_gen/gconvfifo.rf/grf.rf/gntv_or_sync_fifo.gl0.rd/gras.rsts/ram_empty_i_reg/C UF/U0/inst_fifo_gen/gconvfifo.rf/grf.rf/gntv_or_sync_fifo.gl0.rd/gras.rsts/ram_empty_i_reg/C Exception Source (From) Destination (To) 	<pre>UF/U0/inst_fifo_gen/gconvfifo.rf/grf.rf/rstblk/ngwrdrst.grst.g7 reqdata_reg/CE reqdata_reg/D reg/L</pre>

© Adiuvo Engineering and Training, Ltd. 2020



Vivado Timing Analysis – Clock Interaction



Generated From Flow Navigator when implementation is open

Yellow show unrelated clock – Indicates CDC issues as well

Means we need to define constraints

	async					(
Clock Group	s					
<u>G</u> roup 1:	[get_clocks	s2_clk]		\otimes	•••	+ =
Group 2	[get_clocks	s1_clk]		\otimes	••••	+



Vivado Timing Analysis – Inter clock Issues

Q, ¥, ♦ ●	Q —	א ⊗_ ת	JI 😶	Inter-Clock Pa	aths - s2_clk to s1_clk - Setup							
General Information	Name	Slack ^1	Levels	High Fanout	From	То	Total Delay	Logic Delay	Net Delay	Requirement	Source Clock	Destination Clock
Timer Settings	l> Path 61	-1.565	0	5	UF/U0/inst_fifo_gens/ram_empty_i_reg/C	reqdata_reg/CE	1.636	0.456	1.180	0.8	s2_clk	s1_clk
Design Timing Summary	Ъ Path 62	-1.426	0	5	UF/U0/inst_fifo_gens/ram_empty_i_reg/C	reqdata_reg/D	1.636	0.456	1.180	0.8	s2_clk	s1_clk
Clock Summary (2)	I Path 63	35.048	0	1	UF/U0/inst_fifo_gen/src_gray_ff_reg[7]/C	UF/U0/inst_fifoc_ff_reg[0][7]/D	1.050	0.419	0.631	36.4	s2_clk	s1_clk
Check Timing (115)	4 Path 64	35.	0	1	UF/U0/inst_fifo_gen/src_gray_ff_reg[3]/C	UF/U0/inst_fifoc_ff_reg[0][3]/D	1.040	0.419	0.621	36.4	s2_clk	s1_clk
🖻 Intra-Clock Paths	Ъ Path 65	35.158	0	1	UF/U0/inst_fifo_gen/src_gray_ff_reg[0]/C	UF/U0/inst_fifoc_ff_reg[0][0]/D	1.111	0.456	0.655	36.4	s2_clk	s1_clk
Inter-Clock Paths	4 Path 66	35.186		1	UF/U0/inst_fifo_gen/src_gray_ff_reg[4]/C	UF/U0/inst_fifoc_ff_reg[0][4]/D	1.083	0.456	0.627	36.4	s2_clk	s1_clk
✓ a s1_clk to s2_clk	4 Path 67	35.191	0	1	UF/U0/inst_fifo_gen/src_gray_ff_reg[8]/C	UF/U0/inst_fifoc_ff_reg[0][8]/D	1.080	0.456	0.624	36.4	s2_clk	s1_clk
9 Setup -11.052 ns (10)	4 Path 68	35.213	0	1	UF/U0/inst_fifo_gen/src_gray_ff_reg[2]/C	UF/U0/inst_fifoc_ff_reg[0][2]/D	1.058	0.456	0.602	36.4	s2_clk	s1_clk
Hold 0.116 ns (10)	Ъ Path 69	35.227	0		UF/U0/inst_fifo_gen/src_gray_ff_reg[6]/C	UF/U0/inst_fifoc_ff_reg[0][6]/D	1.042	0.456	0.586	36.4	s2_clk	s1_clk
V a s2_clk to s1_clk	4 Path 70	35.235	0		U0/inst_fifo_gen/src_gray_ff_reg[1]/C	UF/U0/inst_fifoc_ff_reg[0][1]/D	0.859	0.419	0.440	36.4	s2_clk	s1_clk
9 Setup -1.565 ns (10)												
Hold 0.114 ns (2)												
🖻 Other Path Groups					Detail	ad noth range	rt C	ouroo	and	Docti	notion	Clocks
User Ignored Paths					Detail	ed path repor	ι – Ο	ource	anu	Desti	nalion	CIUCKS
Unconstrained Paths												

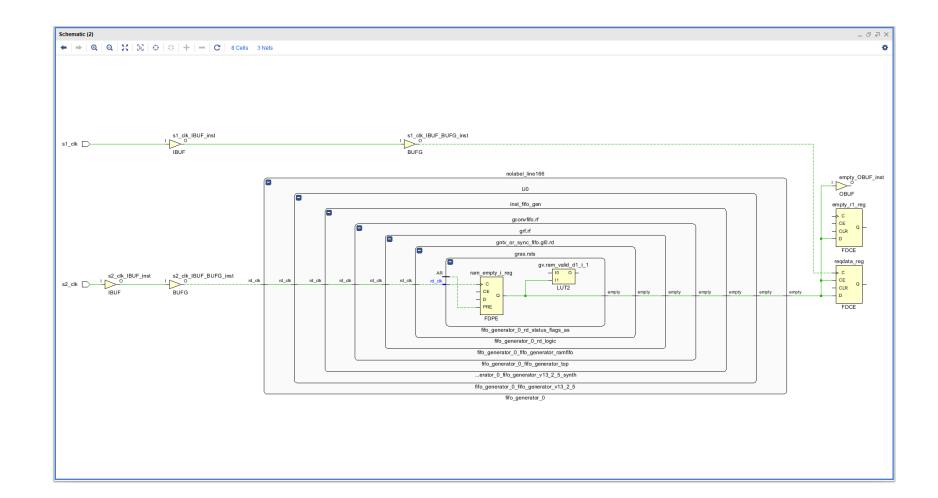
If you forget to report_cdc following synthesis. CDC issues will be apparent in the timing report if we have not correctly addressed the constraints

Summary	
Name	🕨 Path 62
Slack	<u>-1.426ns</u>
Source	UF/U0/inst_fifo_gen/gconvfifo.rf/grf.rf/gntv_or_sync_fifo.gl0.rd/gras.rsts/ram_empty_i_reg/C (rising edge-triggered cell FDPE clocked by s2_clk {rise@0.000ns fall@18.182ns period=36.364ns})
Destination	reqdata_reg/D (rising edge-triggered cell FDCE clocked by s1_clk {rise@0.000ns fall@5.000ns period=10.000ns})
Path Group	s1_dk
Path Type	Setup (Max at Slow Process Corner)
Requirement	0.812ns (s1_clk rise@9710.000ns - s2_clk rise@9709.188ns)
Data Path Delay	1.636ns (logic 0.456ns (27.876%) route 1.180ns (72.124%))
Logic Levels	0
Clock Path Skew	<u>-0.537ns</u>
Clock Uncertainty	<u>0.035ns</u>
Clock DomCrossing	Inter clock paths are considered valid unless explicitly excluded by timing constraints such as set_clock_groups or set_false_path.



Vivado Flow - Schematic

Engineer can use information provided in the text report to navigate to the schematic to understand where CDC might be.





Block RAM





BRAM

Used for storage of the Data in the FPGA, more efficient than using registers.

BRAM can be used as Single / Dual Port – Great for clock conversion

BRAM can be used to implement FIFOs – Great for Clock Domain Crossing

What might we store in the BRAMs

- Filter Coefficients
- Image Lines
- Signal Data
- System Configuration Data



Block RAM

- RAMS can be protected by error protection codes
- Depending upon the device these are hard coded into the RAMS and transparent to the user or require user implementation
- Regardless of how they are implemented it is a good idea to use a scrubbing algorithm which will read back contents out of the memory periodically and ensure correction of any errors to prevent a build up of errors
- If these RAMS are being used to store the configuration of the system it is a good idea to have a copy of this configuration on the ground payload control system to ease recovery in the worse case



BRAM in Xilinx

- Inherent support for ECC
- Cannot initialise BRAM with a COE file it ECC is used
- BRAM 64 bit data width and greater Hard Hamming Implementation
- BRAM less than 64 bit data width soft hamming implementation
- Can optimise for performance / power



ECC In Block Memory Generator

🝌 Re-customize IP		X A Re-customize IP	×
Block Memory Generator (8.4)		Block Memory Generator (8.4)	4
ODocumentation 🕞 IP Location		ODcumentation 🗁 IP Location	
P Symbol Power Estimation Show disabled ports	Component Name blk_mem_gen_0	Image: Power Estimation Show disabled ports Baic Port A Options Power Estimation Mode Stand Alone Image: Port A Options Power Estimation Mode Stand Alone Image: Port A Options Port A Options Power Estimation Mode Stand Alone Image: Port A Options Port B Options Port B Options	erate address interface with 32 bits mon Clock
	OK	Cancel	OK

© Adiuvo Engineering and Training, Ltd. 2020



Injection of Error in to BRAM

Untitled 4*		
Q 🖬 🔍 Q 🔀 📲 I	 ▶ 1 ± ± 	+ F a b F b
Name	Value	420 ns 430 ns 440 ns 450 ns 460 ns 470 ns 480 ns 490 ns 500 ns 510
> V Stim_memory_0_addr_a[12:0]	000f	□ X 0000 X 0001 X 0002 X 0003 X 0004 X 0005 X 0006 X 0007 X 0008 X 0009
> 😽 Stim_memory_0_addr_b[12:0]	0008	000£
Stim_memory_0_clk	0	
> 😼 Stim_memory_0_data_a[31:0]	000000f	
Stim_memory_0_injectdbiterr	0	
Stim_memory_0_injectsbitter	0	
Stim_memory_0_wr	0	
¹⁸ blk_mem_gen_0_dbiterr	0	
> 😼 blk_mem_gen_0_doutb[31:0]	0000007	0000000e 000000£
> 😼 blk_mem_gen_0_rdaddrecc[12:0]	0007	000e 000£
blk_mem_gen_0_sbiterr	0	



Read out of Error

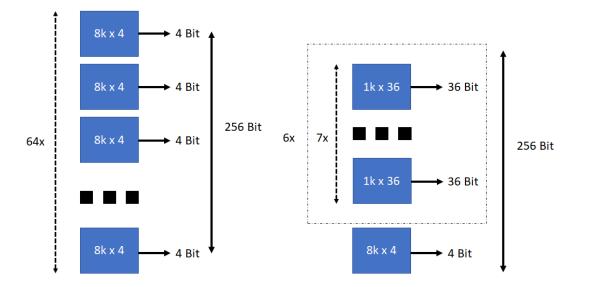
Untitled 4*

Q 📕 @ Q 🔀 ➡ 🛛 🖌 🛨 🛨 + Γ 🖛 ➡ 🖂

																		673.18	1 <mark>0 ns</mark>
Name	Value	590 n	s	600 ns	;	610 ns		620 ns	;	630 ns		640 ns		650 ns	660) ns	670	ns	680_л
> 😻 Stim_memory_0_addr_a[12:0]	000f														000£				
> 😻 Stim_memory_0_addr_b[12:0]	0008	0000	00	01	00	02	00	03	00	04	00	05	000)6	0007		0008		0009
Stim_memory_0_clk	0																		
> 😻 Stim_memory_0_data_a[31:0]	000000f														0000000	E			
Stim_memory_0_injectdbiterr	0																		
Stim_memory_0_injectsbitter	0																		
Stim_memory_0_wr	0																		
blk_mem_gen_0_dbiterr	0																		
> 😼 blk_mem_gen_0_doutb[31:0]	0000007	00000	0000	0000	00000	0001	0000	0002	0000	003	0000	0004	00000	005	000000	6	00000007	0	000008
> Weblk_mem_gen_0_rdaddrecc[12:0]	0007	000£	00	00	00	01	00	02	00	03	00	04	00	05 X	0006		0007		0008
blk_mem_gen_0_sbiterr	0																		

BRAM Optimisations

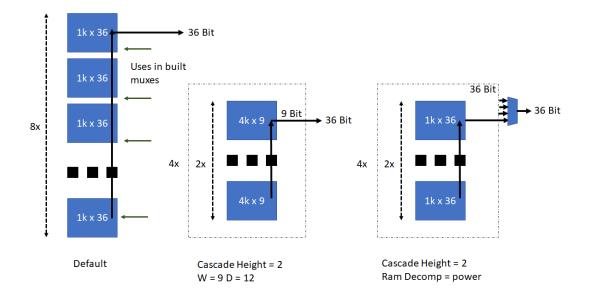
- Optimise BRAM for power / performance
- 6K by 256 RAM implemented using a 64
 BRAMS configured as 8K by 4 bits
- 7 BRAMS configured as 1K by
 36 repeated 6 times + one 8k x4 using 43
 BRAMS
- RAM_decomposition constraint can be used for second structure



BRAM Optimisations

- Can also use RAM cascade_height constraint
- Not just a choice between power and performance
- Also possible to combine the cascade_height and

ram_decomposition









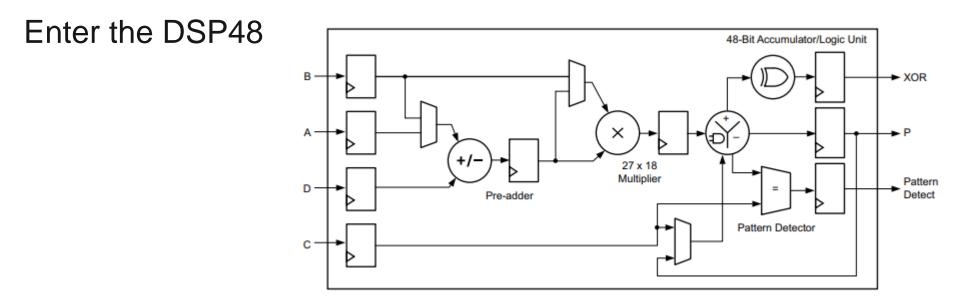


DSP

Mathematical structures such as add / subtract / Multiply / Division can be

implemented in FPGA using LUT and Flip Flops

 Not efficient for high performance - Better to have dedicated resources in the FPGA





DSP

Applications

- Fixed- and floating-point Fast Fourier Transform (FFT) functions
- Systolic FIR filters
- MultiRate FIR filters
- CIC filters
- Wide real/complex multipliers/accumulators



Multi Giga Bit Transcievers





Multi Giga Bit Transceivers

Xilinx FPGA implement high speed interfaces such as SATA, USB3, PCIe, Display port and Chip to Chip communications using MGTs

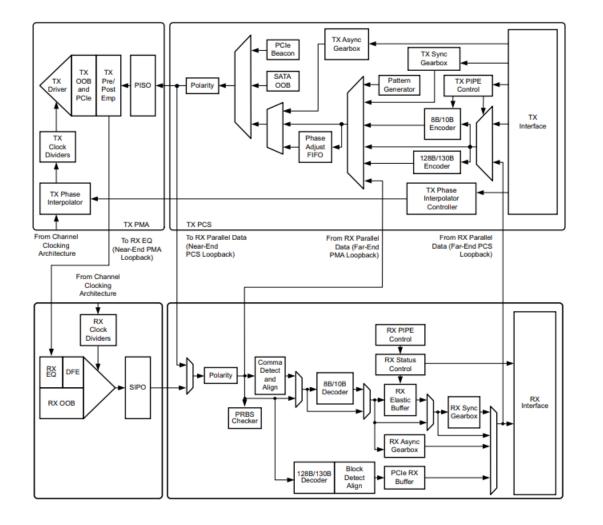
Special dedicated transceivers in the device – Normally in Quads with refence clock as well

Differential signaling and use Current Mode Logic

PMA Physical Media AttachmentPCS Physical Coding Sublayer



MGT Internals





Bandwidths Supported

Device	Туре	Max Bandwidth (Gbs)
Versal ACAP	GTY / GTM	32.75 / 58.0
Veral Premium	GTY / GTM	32.75 / 112.0
Virtex US+	GTY / GTM	32.75 / 58.0
Kintex US+	GTH / GTY	16.3 / 32.75
Virtex US	GTH / GTY	16.3/30.5
Kintex US	GTH	16.3
Virtex 7	GTX / GTH / GTZ	12.5 / 13.1 / 28.05
Kintex 7	GTX	12.5
Artix 7	GTP	6.6
Zynq MPSoC	GTR / GTH / GTY	6.0 / 16.3 / 32.75
Zynq	GTX	12.5



Constraints



-59



Constraints

Help us instruct and guide the implementation tool

Multiple different types of constraints

- Timing constraints— The timing relationships required for correct operation
- Timing Exceptions— Define any exceptions to those constraints e.g. Multi Cycle Paths or False Paths
- Implementation constraints—Constraints used in the design's placement and routing e.g. IO location / type / location in the device



Clock Constraints

Vivado has several different types of clock constraints

- Primary Clocks those that enter through an I/O pin
- Generated Clocks those which are generated automatically via an internal PLL or by the design (for instance, dividing a clock by two with a flip-flop). With generated clocks, one describes how the master clock (either a prime or other generated clock) modifies the waveform
- Virtual Clocks These are not attached to anything within the design netlist but can be used for I/O timing



Clock Constraints

Vivado has three different types of clocks

- Synchronous Clocks Synchronous clocks have a predictable timing/phase relationship, which is normally the case for a primary clock and its generated clocks because they share a common root clock and will therefore have a common period.
- Asynchronous Clocks Asynchronous clocks have no predictable timing/phase relationship, which is normally the case for different primary clocks (and the clocks generated from these primary clocks). Asynchronous clocks have different roots.
- Unexpandable Clocks Two clocks are unexpandable if a common period cannot be determined over 1000 clock cycles. If a common clock period cannot be established, then Vivado uses the worst case set-up relationship over the 1000 cycles. However, there is no guarantee that this relationship truly represents the actual worst case. That estimate is just the best that Vivado can do with the information provided.



Placement Constraints

Vivado has a range of constraints we can use

- •Placement Constraints define cell location
- •Routing Constraints define signal routing
- •I/O Constraints define I/O location and I/O parameters
- Configuration Constraints define configuration methods

Tcl Console Mes	sages Log Rep	ports Design	Runs Package Pine	I/O Ports	×		
Q ≚ ≑ €	+ 14						
Name	Direction	Board Part Pin	Board Part Interface	Neg Diff Pair	Package Pin	$\wedge 1$	Fixed
🗸 🕞 All ports (25)							
> 🤒 data_in (8)	IN						
> 🍯 result (12)	OUT						\checkmark
🗸 🕞 Scalar ports	(5)						
🕢 valid	OUT				B15	~	
🕑 clk	IN				Y9	~	
🕑 empty	IN				AB20	~	
🕢 read	OUT				AB21	~	\checkmark
	IN I				4000		



Placement Constraints

			findes - Duome	adiavo/nat_proj	jects/01_vivado/01_vivado.xp	1- 11000 202011						•
	ts <u>W</u> indow Layout <u>View</u> <u>H</u> elp <u>Q. Quick</u>									Impleme		ete, Failed Timing
		I 🕺									🔛 Defa	ult Layout
ow Navigator	IMPLEMENTED DESIGN - xc7z020clg484-1											
PROJECT MANAGER	Sources Netlist ×	? _ 🗆 🖆 🏼 Pr	oject Summary	× Device ×	timing.xdc x io.xdc x	block average.vhd x						? 🗆
Settings	중 년	•	, , , ,		₩ Po 🔍 🔍							
Add Sources	* 9					\$105_X109		X10/V100 (SAUCIA)	SUC1_X100/130 (St	JCENT SUCE_X207013		x110/130 (SUCIU
Language Templates	(design_1_fifo_generator_0_0_blk_mem_gen_					BRAM-R-X67Y125 CMBLL L						
₽ IP Catalog	(design_1_nio_generator_0_0_bik_inem_gen_	vo_4_4_syn						D 💭 🖉 🗄	HPL.			
	ebmg.native_blk_mem_gen (design_1_fifo_ger	nerator_0_0				₽ 2						
PINTEGRATOR		i i i) #4 a c c c c c c c c c c c c c c c c c c		α P						
Create Block Design	design_1_fifo_generator_0_0_blk_mem_gen_ge	eneric_cstr)	102/120 (SLICE#1 SLICE_X1	SUCE SUCE	x304/129 (SUCEM) SUCE_X305/129 (SUCEL)	SLIDE_X100*	129 (SLICIL) SLICI	X107/129 (SUICEL)	SUC1_X1007129 S	SUCE_X109/1	sizocen ance	1110/129 (SUCILI SU
Open Block Design	9) p[0].ram.r (design_1_fifo_generator_0_0_blk_r	nem den pr					Part I					
Generate Block Design	s (39)					말 >		Parto				
	1_noinit.ram (design_1_fifo_generator_0_0_blk	_mem_gen_	2월 8 발 (₽ ‱ 8				
IMULATION	Nets (40)			09/128 (SUCE)	x104/128 (SUCE/1 SUCE, X107/128 (SUCE)	100 SLICE_X108/	120 50.051	X107/128 (SUICE)	SUCE_X108/128 5	SUCE_X109-11	BISUCEU SUCE	X110-158 Terrort an
Run Simulation	Leaf Cells (2)		1. 陈文明5日27 印,	ten a a a a a a a a a a a a a a a a a a a	<u>씨</u> (1927) 태 , (1947) 8	\ ¢₁	Beer P [™] P [†]	2 2 2	CIERLM_	PB27 6 . 1	₽₽₽₿	
	C DEVICE_/SERIES.NO_BMM_INFO.SDP.SIMPLE								05			
TL ANALYSIS		1	요리 말!									
Open Elaborated Design	Path Properties ?	_ 🗆 🖒 X 🕌			,7104/7127 (SUCE41 SUCE,7105/127 (SUCE41	SLC6_X100		204 12 X107/127 (SUESU	SUCE_X100/12/ [5		и Isucen III висе	x110/12/ (SUCE)
	Ъ Path 1 ◀	- -> 🗢 🚃		Fe-at Cina	MX669[426 m] , 10 m = 2		國 9 ¥ ¥ 6 ि □	. 62		₽¥126 🖬 . 8	₽	L LAX74Y926
YNTHESIS • Run Synthesis	✓ Summary	^					/ ē 6					
	Name 1, Path 1 Slack -0.249ns					5	2-1 B			18		
Open Synthesized Design									SUCE A 100/136 [S			X1109261540560
Constraints Wizard	General Properties Report Cells								anond a new lat			والاستفاق
Edit Timing Constraints												
🐞 Set Up Debug	Tcl Console Messages Log Repo	orts Design Run	s IP Status	Power DRC	Methodology Timing ×							? _ □
🖸 Report Timing Summary	Q ₹ ≑ 9	• Q -	h ⊗, M (Intra-Clock P	aths - clk_0 - Setup							
Report Clock Networks	Timer Settings	^ Name	Slack ^1 Lev	els High Fanout	From	То	Total Delay	Logic Delay	Net Delay	Requirement	Source Clock	Destination Clo
Report Clock Interaction	Design Timing Summary	1, Path 1	-0.249		design_1_i/fam/CLKBWRCLK	design_1_i/avor_reg[11]/D	5.264	3.887	1.377	5.0	clk_0	clk_0
Report Methodology	Clock Summary (1)	🔓 Path 2	-0.137		design_1_i/fam/CLKBWRCLK	design_1_i/avator_reg[7]/D	5.150	3.773	1.377		clk_0	clk_0
Report DRC	> 🔓 Check Timing (20)	1, Path 3	-0.118		design_1_i/fam/CLKBWRCLK	design_1_i/avator_reg[8]/D	5.090	3.763	1.327		clk_0	clk_0
Report Noise		Path 4	-0.108	5 1	<pre>design_1_i/fam/CLKBWRCLK design 1 i/fam/CLKBWRCLK</pre>	design_1_i/avor_reg[10]/D design_1_i/avator_reg[9]/D	5.077	3.783	1.294		clk_0 clk_0	clk_0 clk_0
Report Utilization	 Setup -0.249 ns (10) 	Path 6	-0.024		design_1_i/fam/CLKBWRCLK	design_1_i/avator_reg[9]/D	4.976	3.649	1.105		clk_0	clk_0
Seport Power	Hold 0.100 ns (10)	ly Path 7	0.004		design_1_i/fam/CLKBWRCLK		4.963	3.669	1.294		clk_0	clk_0
	Pulse Width 2.000 ns (30)											



Device Selection





Benefit of FPGA

FPGA's offer several benefits to the system designer

- Flexibility of Design performance, upgrades
- Reduction in NRE and Cost.
- Reliability One Device as opposed to lots if using discrete devices - increased reliability with reduced solder connections.
- Time to market can be reduced.
- Maintainability ability with some FPGA to update in the field.



Device Selection - SRAM, OTP or FLASH?

- SRAM: The FPGA program is stored in an external memory and loaded into the FPGA each time it is powered.
- FLASH: The FLASH architecture of the FPGA also contains the program; no external memory device is needed.
- One Time Programmable (OTP): The FPGA is applied by blowing fuses in the device. Once programmed, it cannot be modified.

Device Selection

• SRAM

- Higher Performance (+)
- Higher Capacity (+)
- Needs configuration at power up (-)
- Higher Power (-)
- Susceptible to Configuration Corruption (-) but there are mitigation schemes

• OTP

- Live at power up (+)
- Cannot be updated / fixed design (-)
- Lower Power (+)
- Lowest Performance (-)

- Flash based
 - Live at power up (+)
 - No Configuration Corruption (+)
 - Updateable in the field (+)
 - Middling Performance (-)



Device Selection

Typical use cases

- SRAM
 - Software defined radio, image processing, satellite communications, EW, AI etc.
- OTP
 - Control and communication, security functions e.g. crypto
- Flash
 - Control & Communication, Crypo, image processing, SDR



Vivado IP Integrator

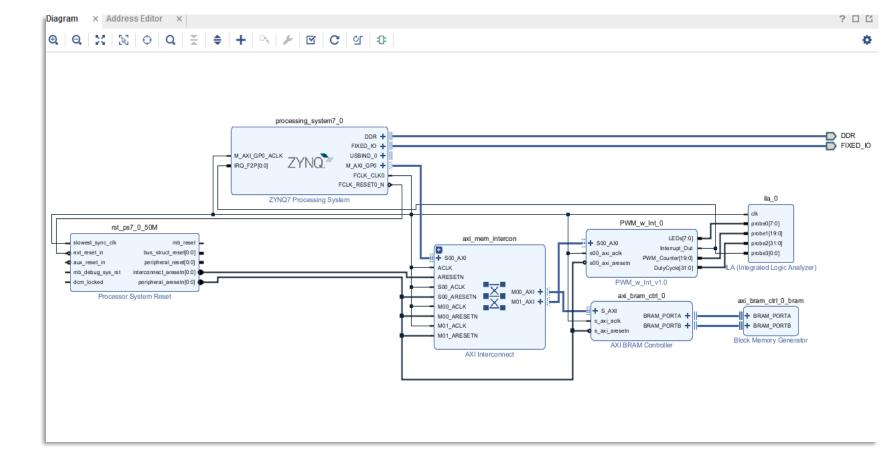


Vivado IP Integrator (IPI)



Create system-level designs

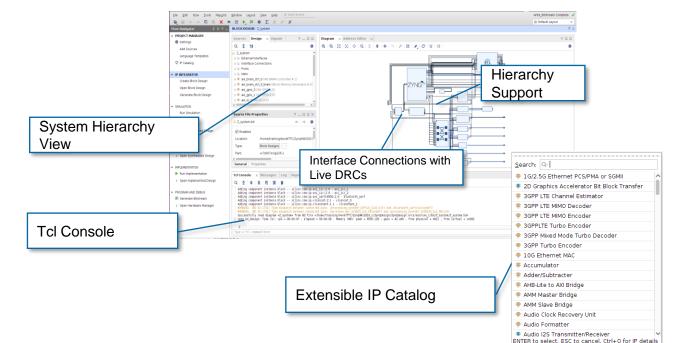
- Instantiate and interconnect IP cores
- IP-centric design flow
 - Plug-and-play IP
 - Vast IP catalog
- Accelerates
 - Integration
 - Productivity
- Example applications
 - Embedded
 - DSP
 - Video
 - Analog
 - Networking





Vivado IP Integrator: Intelligent IP Integration

- Automated IP subsystems
- Block automation for rapid design creation
- One click IP customization
- Board aware
- Support all 7 Series FPGAs and Zynq SoCs
- Built-in presets, accelerating design creation





Vivado IP Integrator: Intelligent IP Integration

Correct-by-construction

- Interface level connections
- Extensible IP repository ullet
- Real-time DRCs and parameter propagation / resolution
- Designer assistance

Diagram × A	Address Editor × IP Catalog >	<	
ଷ୍ ର୍ 🔀	№ ⊕ Q ≚ ♦ +	ା 🕞 🖌 🖂 🖸 ପ୍ର	•
📌 Designer Ass	sistance available. Run Connection A	utomation	
	Run Connection Automation Automatically make connections in your design by ch configuration options on the right.	necking the boxes of the interfaces to connect. S	Select an interface on the left to display its
	Q ¥ ♦ ✓ All Automation (1 out of 1 selected) ✓ Image: All Automation (1 out of 1 selected) ✓ Image: All Automation (1 out of 1 selected) ✓ Image: All Automation (1 out of 1 selected) ✓ Image: All Automation (1 out of 1 selected) ✓ Image: All Automation (1 out of 1 selected) ✓ Image: All Automation (1 out of 1 selected)	Master interface to an unconnected Slave in starting from the Slave Interface. Options Slave:	I) to a selected Slave interface. To connect a iterface, please use connection automation /axi_bram_ctrl_0/S_AXI /axi_mem_intercon Auto Auto
	?	<	OK Cancel



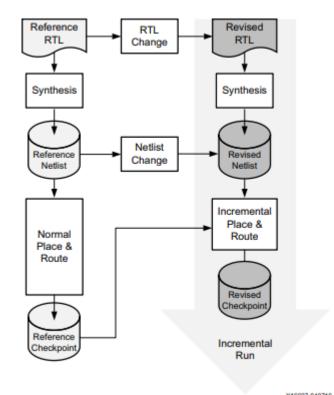
Reducing Run Time





Design Check Points

- Vivado uses a physical design database to store placement and routing information
- Design checkpoint files (.dcp) allow you to save and restore this physical database at key points in the design flow





Vivado – Reducing Compile Time

- Implementing a FPGA design can take a considerable time > 1 Hour
- Iterating the design can therefore be an issue, there are several options which can reduce the implementation time both in synthesis and place and route

Synthesis							
Global	Performs a traditional top-down synthesis of the entire design. Selecting this option takes the longest time because you need to re-run the entire synthesis every time you make a change.						
Out of Context Per IP	Runs synthesis and creates a Design Check Point (DCP) for every individual IP block within your design. These check points are then collected into a black-box at the top-level implementation. Using this option means that only the blocks you change need to be re-synthesized, which saves time. OOC-IP also creates an IP customization file (XCI) for each IP block, allowing for customization and OOC XDC files. OOC-IP is the default setting for synthesis within Vivado. This option applies to all IP within the block diagram.						
Out of Context Per Block Diagram	Like the OOC-IP option however this option allows you to define the entire block diagram as OOC						



Vivado – Reducing Compile Time

- Incremental synthesis can also be used when the changes are small
- Write out incremental synthesis to the post synthesis design check point
- Selecting incremental synthesis then provides two options
 - Automatically use previous DCP
 - Use a defined DCP

•	Synthesis					
roject Settings	Specify various settings as	sociated to Synthesis				
General						
Simulation	Constraints					
Elaboration		-				
Synthesis	Default <u>c</u> onstraint set:	constrs_1 (active)	~			
Implementation	Report Settings					
Bitstream	Strategy: 🌆 Vivado S	Synthesis Default Reports (Vivado Syr	nthesis 2020)			
IP	Strategy.	Synthesis Delaut Reports (Wado Syn	lulesis 2020) •			
ool Settings	Settings					
Project			^			
IP Defaults	Write Incremental Sy	nthesis				
XHub Store	Incremental synthesis:	Notset				
Source File	-					
Display	Strategy:	here is the set of the	do Synt 👻 💾			
WebTalk	Description:	Vivado Synthesis Defaults				
Help Text Editor						
3rd Party Simulators	✓Synth Design (vivado)					
Colors	tcl.pre tcl.post		•••			
Selection Rules		rebuilt	~			
Shortcuts	-flatten_hierarchy		~			
Strategies	-gated_clock_conver	sion off 12	¥			
Window Behavior	-bufg -directive	Default	~			
		Delault				
	-retiming		J ~			
	Select an option above to s	see a description of it				



Vivado – Reducing Compile Time

- Incremental implementation allows use of Design Check Point as the starting point
- Preserves QoR predictability by reusing prior placement and routing from a reference design
- Speeds up place and route run time or attempts last mile timing closure

Q- Project Settings	Implementation	Implementation Specify various settings associated to Implementation						
General	Specily various settings associ	Specily various settings associated to implementation						
Simulation								
Elaboration	Constraints							
Synthesis	Default constraint set:	constrs_1	(active)	~	7			
Implementation		-						
Bitstream	Report Settings							
> IP	Strategy: 🏂 Vivado Impl	lementation	Default Reports (Vivado Impleme	ntati 🦄	-			
	0-11-1				_			
Tool Settings	Settings							
Project IP Defaults	Incremental implementation	n: Not set			î			
> XHub Store	Incremental implementation	Notset						
Source File	Strategy:	🔓 Viva	눰 Vivado Implementation Defaults (😒					
Display				_				
WebTalk	Description:	Default	settings for Implementation.					
Help	✓Design Initialization (init_o	design)						
> Text Editor	tcl.pre			•••				
3rd Party Simulators	tcl.post			•••				
> Colors	└Opt Design (opt_design)							
Selection Rules	is_enabled		\checkmark					
Shortcuts	tcl.pre			•••				
> Strategies	tcl.post			•••				
> Window Behavior	-verbose							
	-directive		Default	~				
	More Options				~			
	Select an option above to see	a descriptio	n of it					



Strategies and Reports





Vivado – Strategies

- Strategies are a defined set of Vivado implementation feature options that control the implementation results.
- These strategies can be used to explore:
 - Timing Performance (e.g., Performance_Explore)
 - Congestion Strategies to reduce routing congestion in areas of the design
 - Area Optimise for area
 - Power Optimise for power
 - Quick Flow Reduced implementation time

We should always of course try to achieve timing closure



Vivado – Reports

- Vivado provides several reports which can be used to help focus in on performance issues in the design:
 - Design Analysis Report Provides information on design timing, congestion, and complexity of design
 - Quality of Result Report Provides overall design assessment and methodology check – QOR can also make suggestions to fix issues in the design.
- Both are very useful to achieve timing closure of the design



Vivado – Design Analysis Report

Design Analysis Report provides information on:

- Timing Provides information on the timing and physical characteristics of timing paths.
- Complexity Provides information on routing complexity and LUT distribution.
- Congestion Provides information on routing congestion



No need to run full implementation. Generate report after running opt_design command in TCL

Report Design Analysis	
merate in-depth analysis reports of top timing paths and quantify the design complexity.	
Results name: design_analysis_2	C
Options Advanced Timer Settings	
✓ Ti <u>m</u> ing	
All <u>p</u> aths	
Path delay type: max 🗸	
Maximum number of paths: 10 🌲	
◯ Sp <u>e</u> cific paths	
Extend analysis: show worst path to startpoint and from endpoint for each critication of the startpoint and from endpoint and	ical path
Include logic level distribution	
Number of paths: 1,000 🗘	
Routes	
Complexity	
<u>C</u> ells to analyze (default: top):	
Hierarchical depth: 2	
✓ Congestion	
Minimum congestion level: 5 \$	
Command: gn_analysis -complexity -timing -setup -max_paths 10 -congestion -name	design englysis (
Zommand: jgn_analysis -complexity-uning -setup -max_paths to -congestion -name. ✓ Open in a new tab	design_analysis_2
) ок	Cancel



Vivado – Design Analysis Report

- Along with timing information, DAR can provide information on design complexity including indicating design risk for implementation
- Low Risk Rent Analysis <0.65 and Fan Out <4
- High Risk Rent Analysis >0.65 <0.85 and Fan Out >4 % <5 May be difficult to place without congestion

Tcl Console Messages Log Reports	Design Runs Power DRC Design Analysis × Methodology Timing														
Q \ X \$ C	$\mathbf{Q} \mid \mathbf{X} \mid \mathbf{a} \mid \mathbf{A} \mid \mathbf{a} \mid \mathbf{A} \mid \mathbf{a} \mid \mathbf{Complexity}$	Characteristics													
General Information Setup Path Characteristics	Instance	Module	Rent	Average Fanout	Total Instances	LUT1	LUT2	LUT3	LUT4	LUT5	LUT6	Memory LUT	DSP	RAMB	MUXF
Complexity Characteristics	V tmr_example_design_wrapper	tmr_example_design_wrapper	0.53	3.43	16437	74	783	2427	1051	1350	3102	283	9	29	342
 Congestion 	I tmr_example_design_i (tmr_example_design)	tmr_example_design	0.55	3.43	16434	74	783	2427	1051	1350	3102	283	9	29	342
Placer Final															



Vivado – Quality of Result

- Quality of Result Assessment (QoRA) and Quality of Result Suggestions (QoRS) since both provide information that can be used to achieve timing closure
- Like Design Analysis Report Run initially after doing the Opt

1. Overall Assessment Summary			ogy Check Details		
+ QoR Assessment Score +	<pre>++ 2 - Implementation may complete. Timing will not meet ++</pre>	+	Description	Criticality	++ Number of Violations
Report Methodology Severity	Critical warnings	TIMING-6	No common primary clock between related clocks	Critical Warning	2
+	Yes		No common node between related clocks No common period between related clocks	Critical Warning Critical Warning	
+ Incremental Compatible	No I	TIMING-9 +	Unknown CDC Logic +	Warning 	1 ++
<pre></pre>	Review methodology warnings and fix or waive them				



Vivado – Quality of Results

SCORE	MEANING	CORRECTIVE ACTION
1	Design will not implement	Redesign RTL / HLS modules
2	Design will implement timing problems	Review constraints & RTL HLS
3	Design Runs have a small chance of success	Use QoR suggestions, review clocking, ML strategies
4	Design should meet timing if directives used	Use QoR suggestions, ML strategies
5	Design will implement without timing issues	Run Implementation



Vivado – Quality of Results

Date Host Command Design Device Design State ML Models	: Sat Aug 29 : DESKTOP-L3 : report_gon : design_l_v : xczu9eg : Fully Plac : v2019.2.0	9 14:13:22 2 30MJCl runni: r_suggestion vrapper ced	020 ng 64-bit major s -file QOR_Sugg	0 Wed May 27 19:5 release (build gestions.txt	9200)			
port QoR Sug	-							
able of Conte								
. ML Strategi . QoR Suggest . QoR Suggest	ions - XDC	5ummary 				+ Incremental Friendly	+ Pescription	
	++	+	+ place_design 	synth_design	+ No 	+ No 	I Tight constraints for given unsafe paths. Fix unsafe paths by amending the design or adding false path, datapath only, or clock group constraints.	Current Run
-			+			+	+	+



Verification





Verification

Before we can deploy an FPGA based solution we must be sure it functions as expected, across all use cases. This is where verification and validation come in, for those unfamiliar with the terms

- Verification Does the FPGA function in line with the specification
- Validation Is the specification correct i.e. does it address the needs of the use case.

While validation takes place mostly within the higher levels of system engineering domain, verification takes place at the FPGA level. Depending upon the needs of the applications verification can range from being very complex and time consuming to simply confirming the expected behaviour.

At the heart of verification is simulation and the simulator.



Simulation

Untitled 1							_ C J X
Q 📕 @ Q 🔀 📲 🔣	H te te	+[]= =[]=					0
			3,180.171 ns				
Name	Value		2 100	10, 200,	10, 000	12 210	la 200 x-
₩ op_fsm	read_out	<u> </u>	3,180 ns	3,200 ns	3,220 ns read_out	3,240 ns	3,260 ns 3,280 ns
Teplay_axi_aclk	0				ouc		
₩ replay_axi_aresetn	1						
> V replay_axi_awaddr[5:0]	00				00		
> M replay_axi_awprot[2:0]	0				0		
le replay_axi_awvalid	0						
> National and the second seco	00000000				0000000		
replay_axi_wstrb[3:0]	0				0		
Teplay_axi_wvalid	0						
🕼 replay_axi_rready	0						
Na replay_axis_aclk	0						
> 🏹 replay_tdata[95:0]	000abf000ab	X 000 X 000 X 000 X 0					
₩ replay_tvalid	1						
Wa replay_axi_bready	0						
> 🛒 replay_axi_araddr[5:0]	00				00		
> 🛒 replay_axi_arprot[2:0]	0				0		
🖟 replay_axi_arvalid	0						
™ replay_tuser	0						
lla replay_tlast	0						
14 dk	0						
¹ ∰ clk_tx	0						
hdr_valid	0						
> 📑 hdr_data_type[5:0]	00				00		
> 📲 hdr_vc[1:0]	0				0		
> 📑 hdr_wc[15:0]	0000				0000		
14 sop	0						
₩ еор	0						
> 🔜 vc_ident[1:0]	0				0		



Simulation

Verification can be used at several different levels

- Functional Simulation only This check if the design is functionally correct
- Functional Simulation & Code Coverage This checks that along with the functional correctness of the design that, all of the code within the design has been tested.
- Gate Level Simulation This verifies the functionality of the design when back annotated with timing information from the final implemented design, this can take a considerable time to perform.



Verification What else can we do?

- Static Timing Analysis This analyses the final design to ensure the timing performance of the module is achieved.
- Formal Equivalence Checking This is used to check the equivalence of netlists against RTL files



www.adiuvoengineering.com



adam@adiuvoengineering.com