Traineeships in Advanced Computing for High Energy Physics (TAC-HEP)

GPU & FPGA module training: Part-2

Week-3: *Hands-on with vivado_hls, output review*

Lecture-5: April 4th 2023

Varun Sharma

University of Wisconsin – Madison, USA

WISCONSIN
UNIVERSITY OF WISCONSIN-MADISON

So Far…

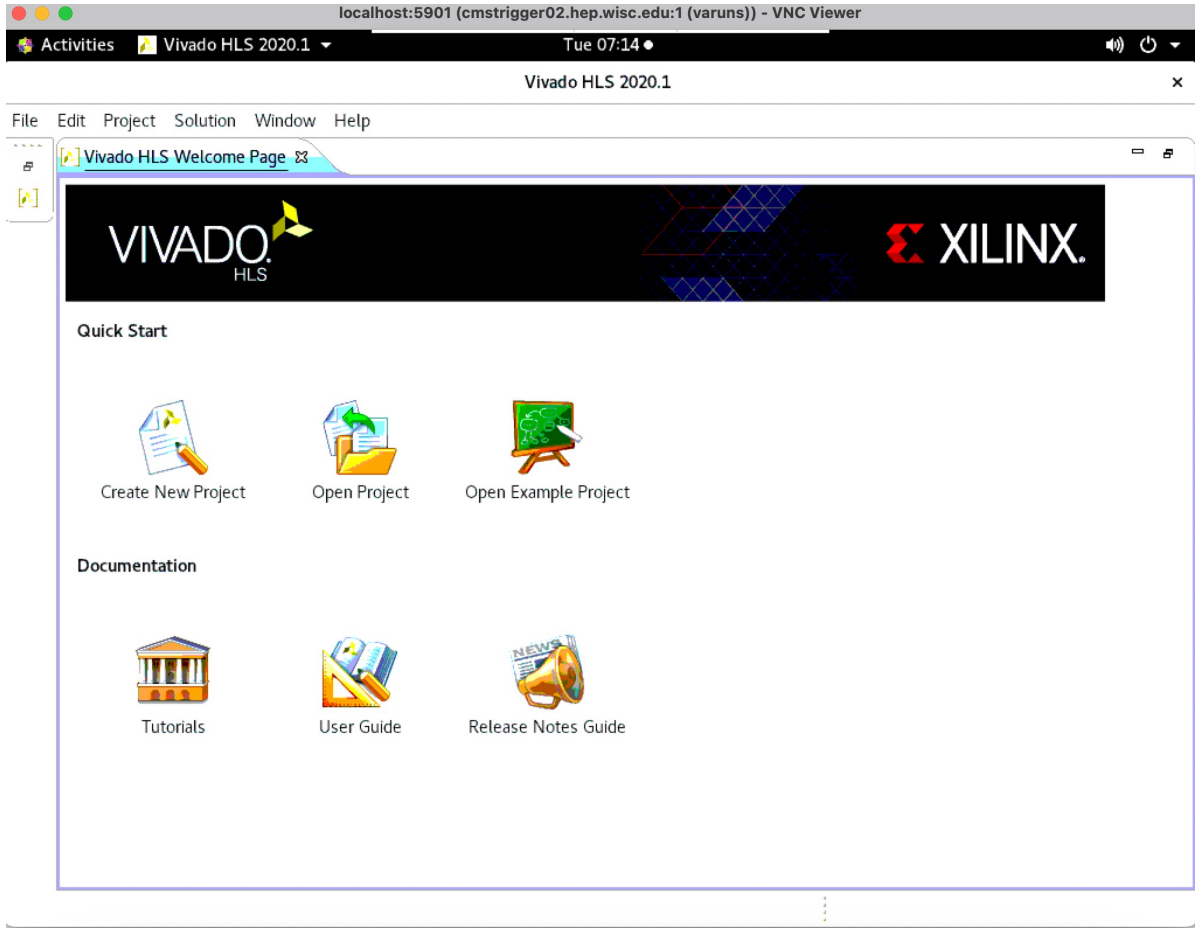
- **FPGA and its architecture**
  - Registor/Flip-Flops, LUTs/Logic Cells, DSP, BRAMs
  - Clock Frequency, Latency
  - Extracting control logic & Implementing I/O ports
- **Parallelism in FPGA**
  - Scheduling, Pipelining, DataFlow
- **Vivado HLS**
  - Introduction, Setup

Today:

- Vivado HLS hands-on with an examples:
  - Using GUI, CLI
- Review output reports (C-Sim, C-Synthesis)

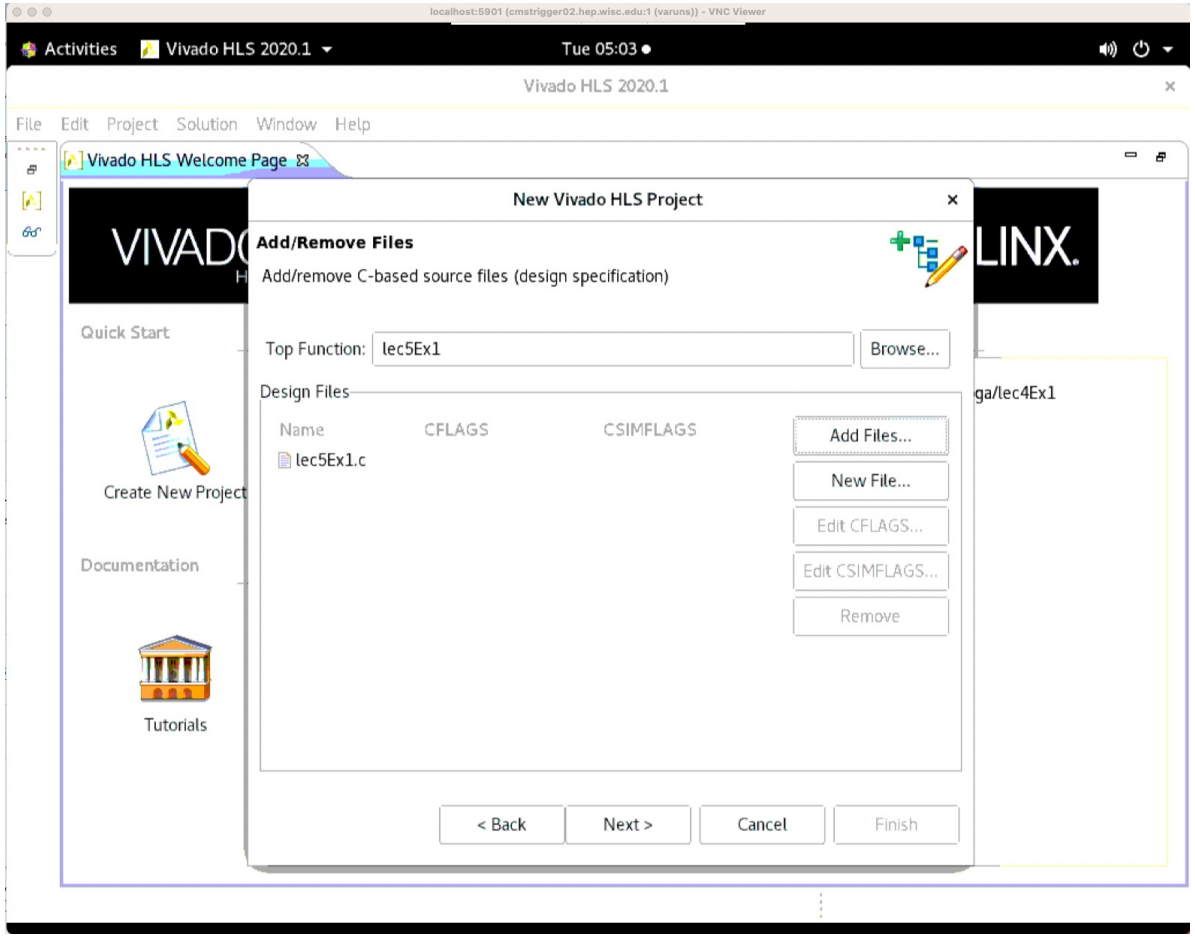
Reminder: Steps to follow

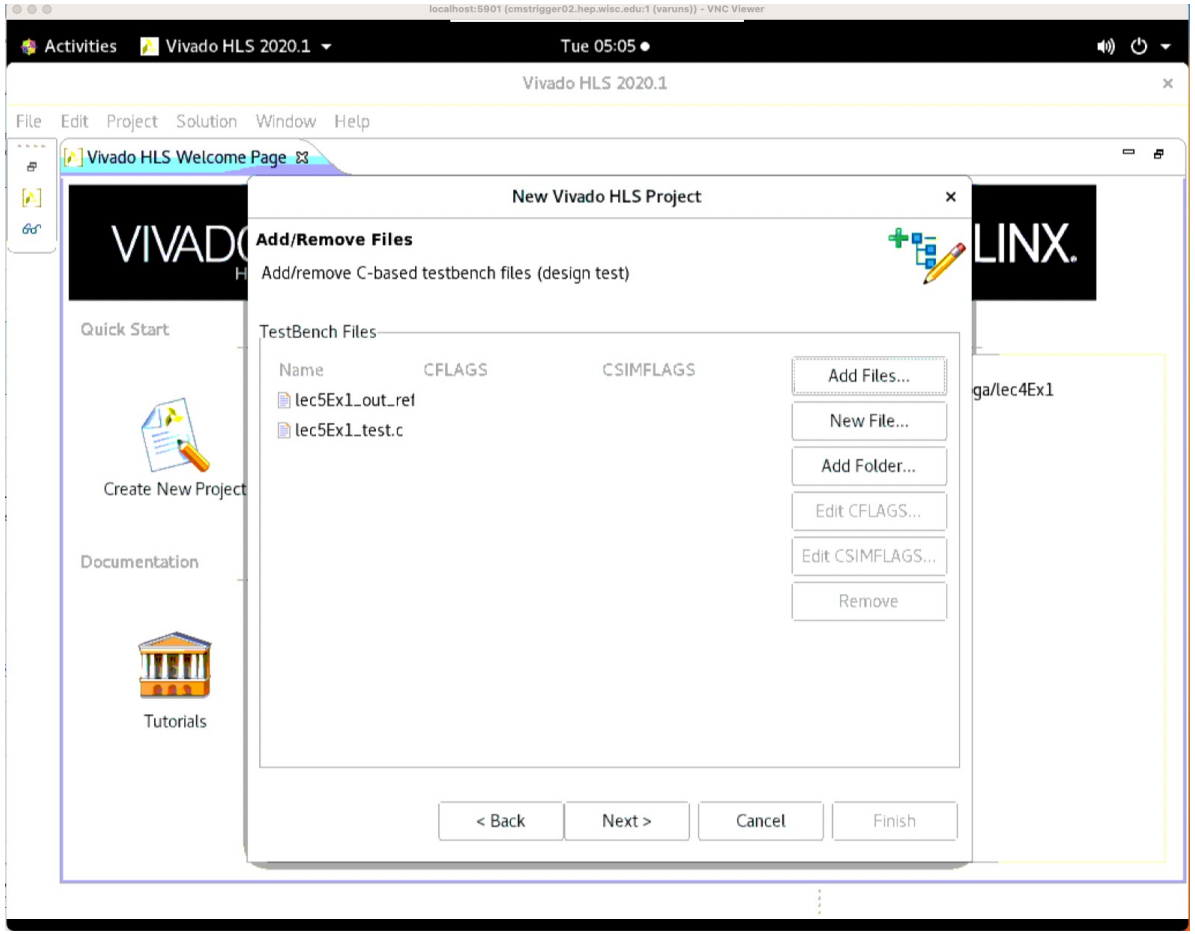
- **Step-1: Creating a New Project/Opening an existing project**
- **Step-2: Validating the C-source code**
- **Step-3: High Level Synthesis**
- **Step-4: RTL Verification**
- **Step-5: IP Creation**



Creating project

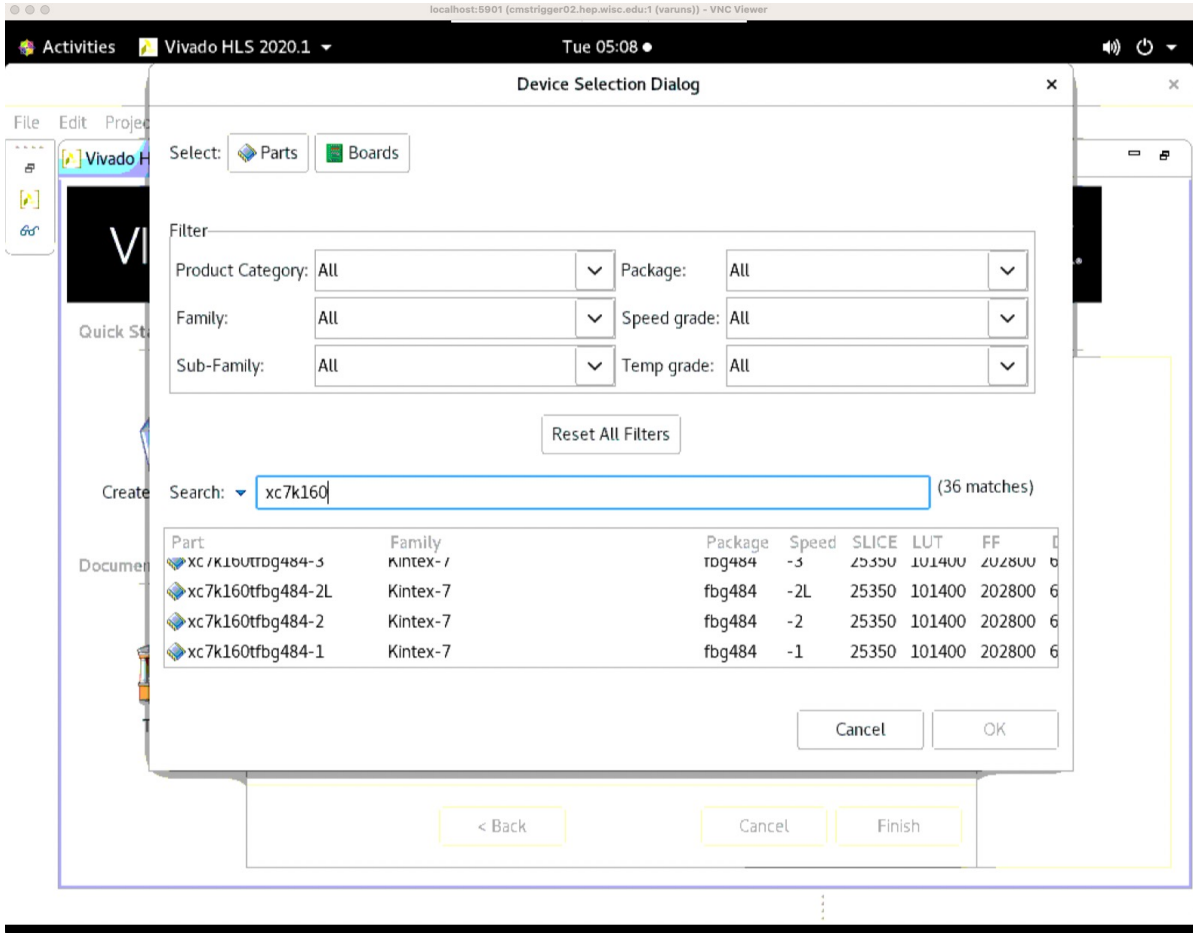
Eg.: lec5Ex1

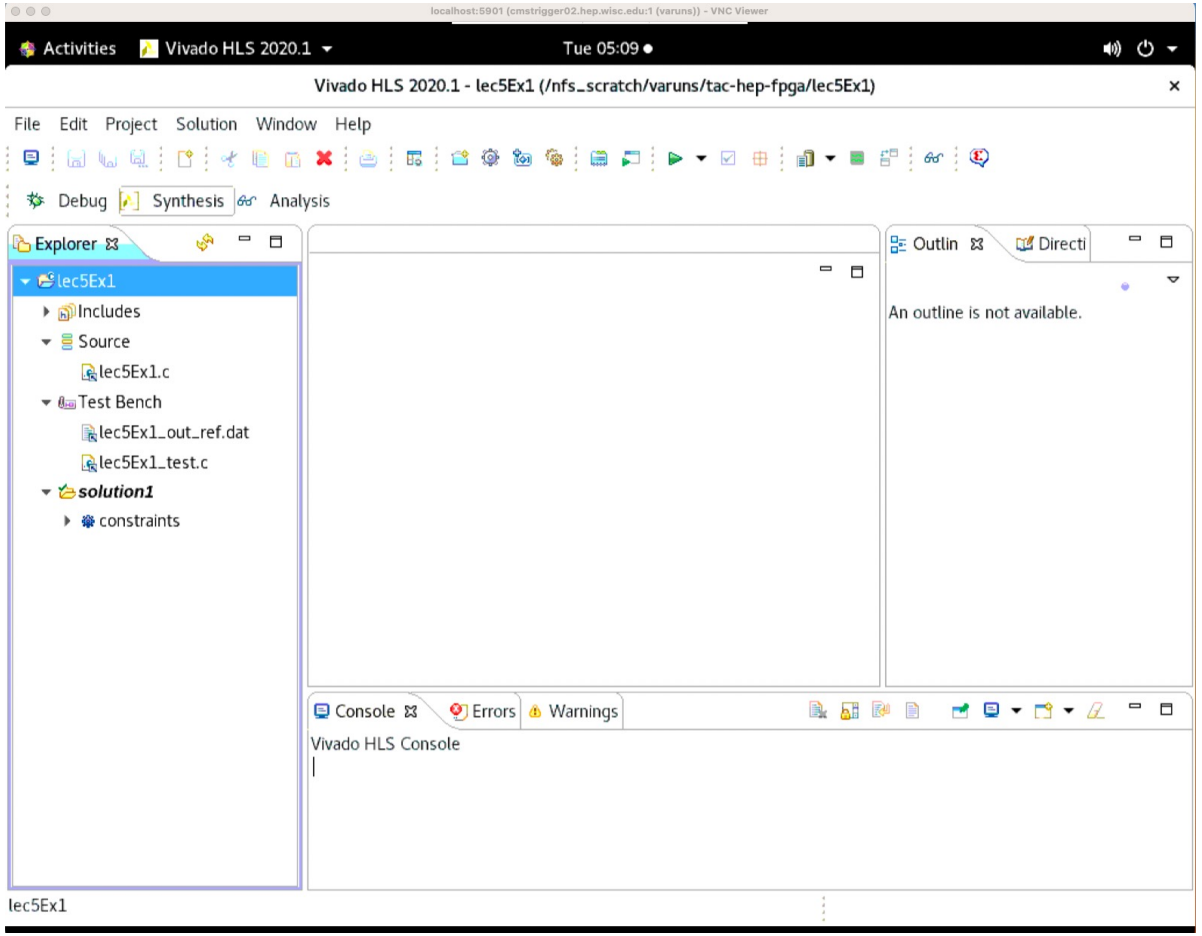




Selecting device

Eg.: lec5Ex1

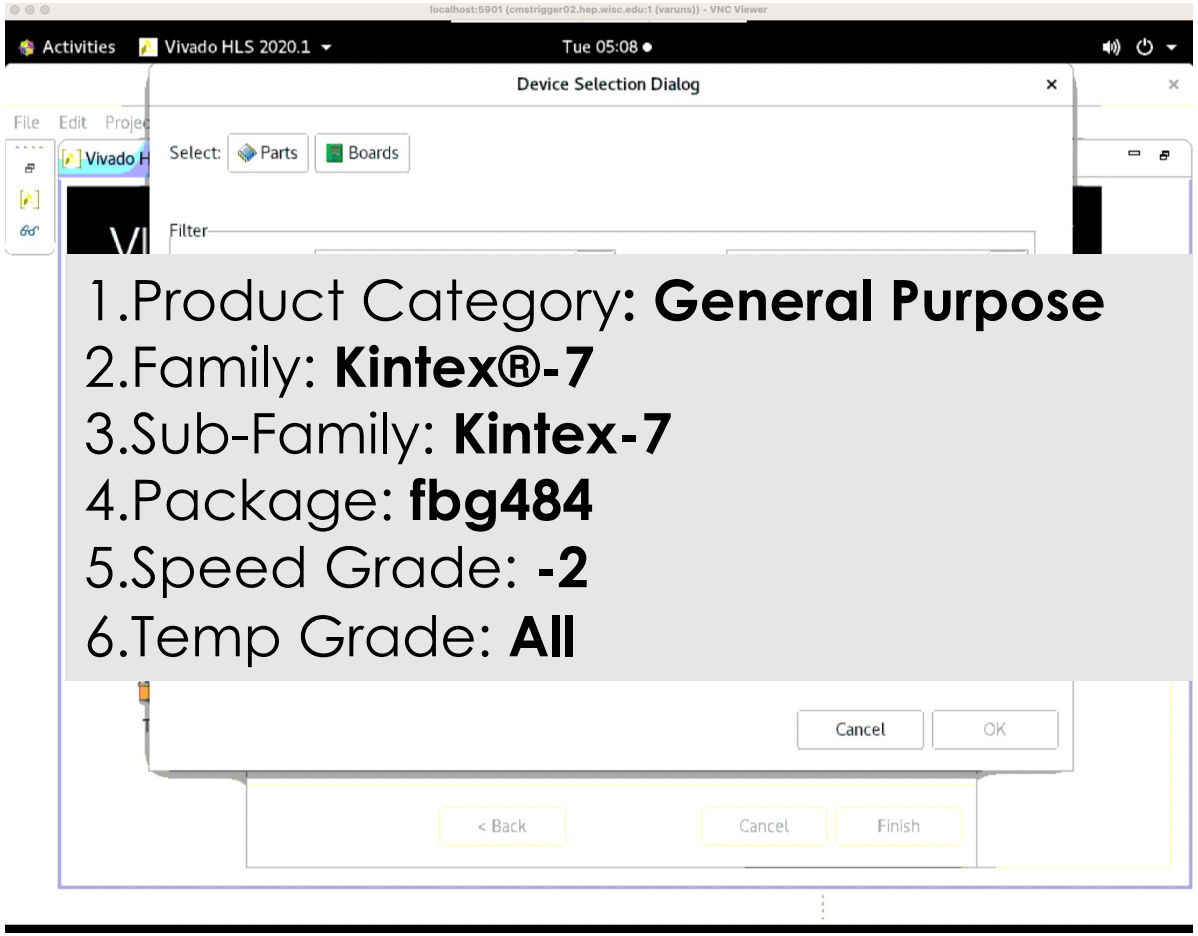




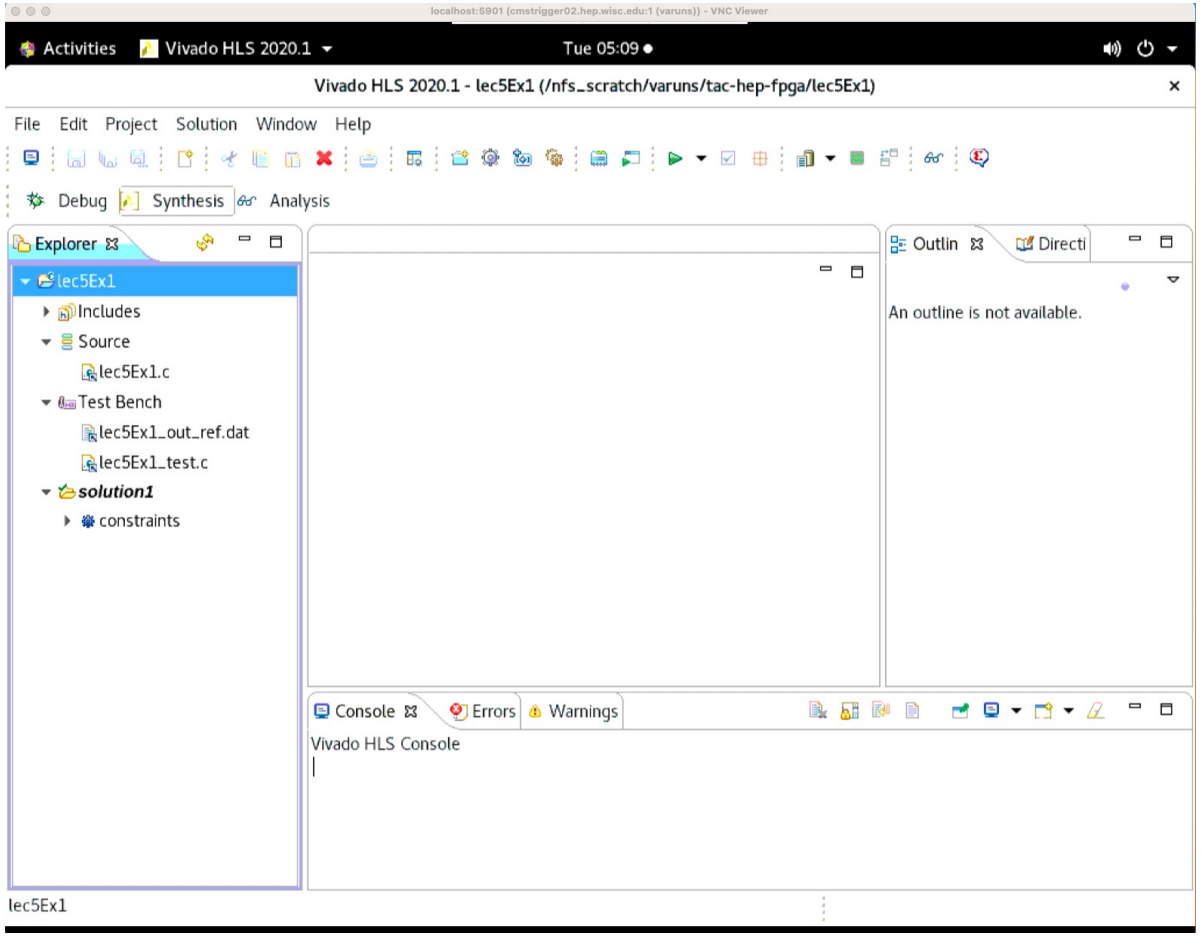
Target Device: xc7k160tfbg484-2

Selecting device

Eg.: lec5Ex1



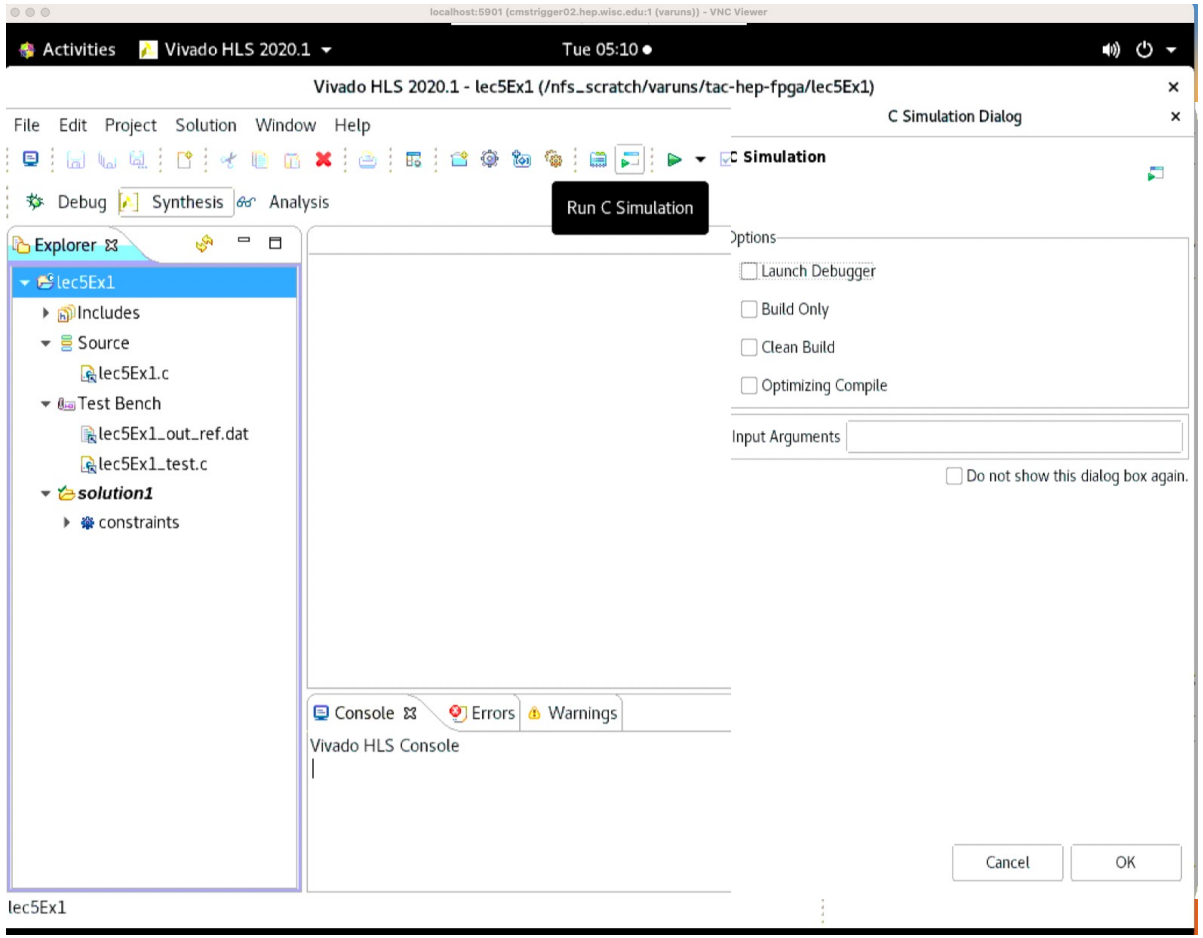
1. Product Category: **General Purpose**
2. Family: **Kintex®-7**
3. Sub-Family: **Kintex-7**
4. Package: **fbg484**
5. Speed Grade: **-2**
6. Temp Grade: **All**



Target Device: xc7k160tfbg484-2

C-Simulation

Eg.: lec5Ex1



Pre-synthesis validation: Correct implementation of C-program for required functionality

Validated using a test bench

Launch Debugger: Compiles the C code & opens the debug perspective

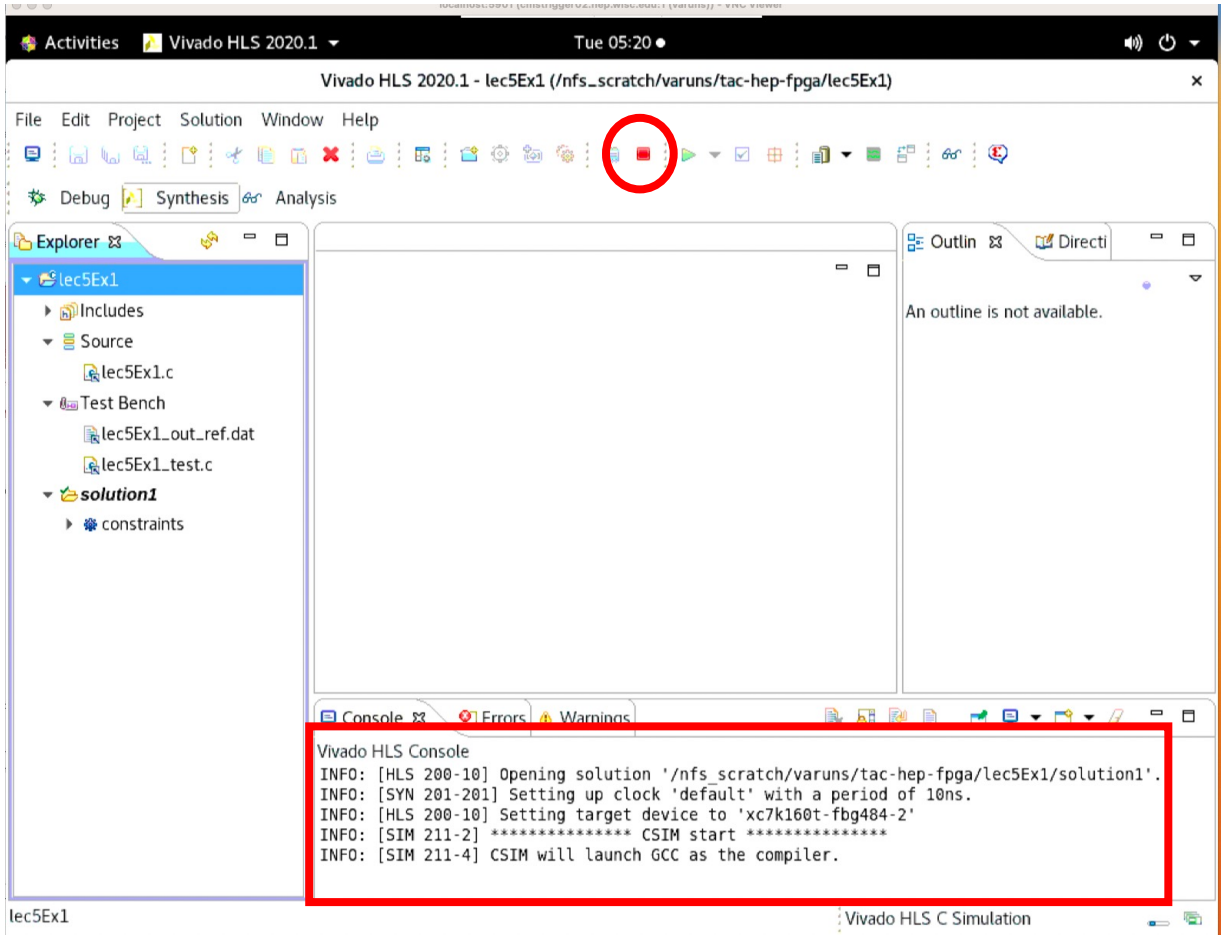
Build Only: C code compiles, but the simulation does not run

Clean Build: Remove any existing executable and object files from the project before compiling the code.

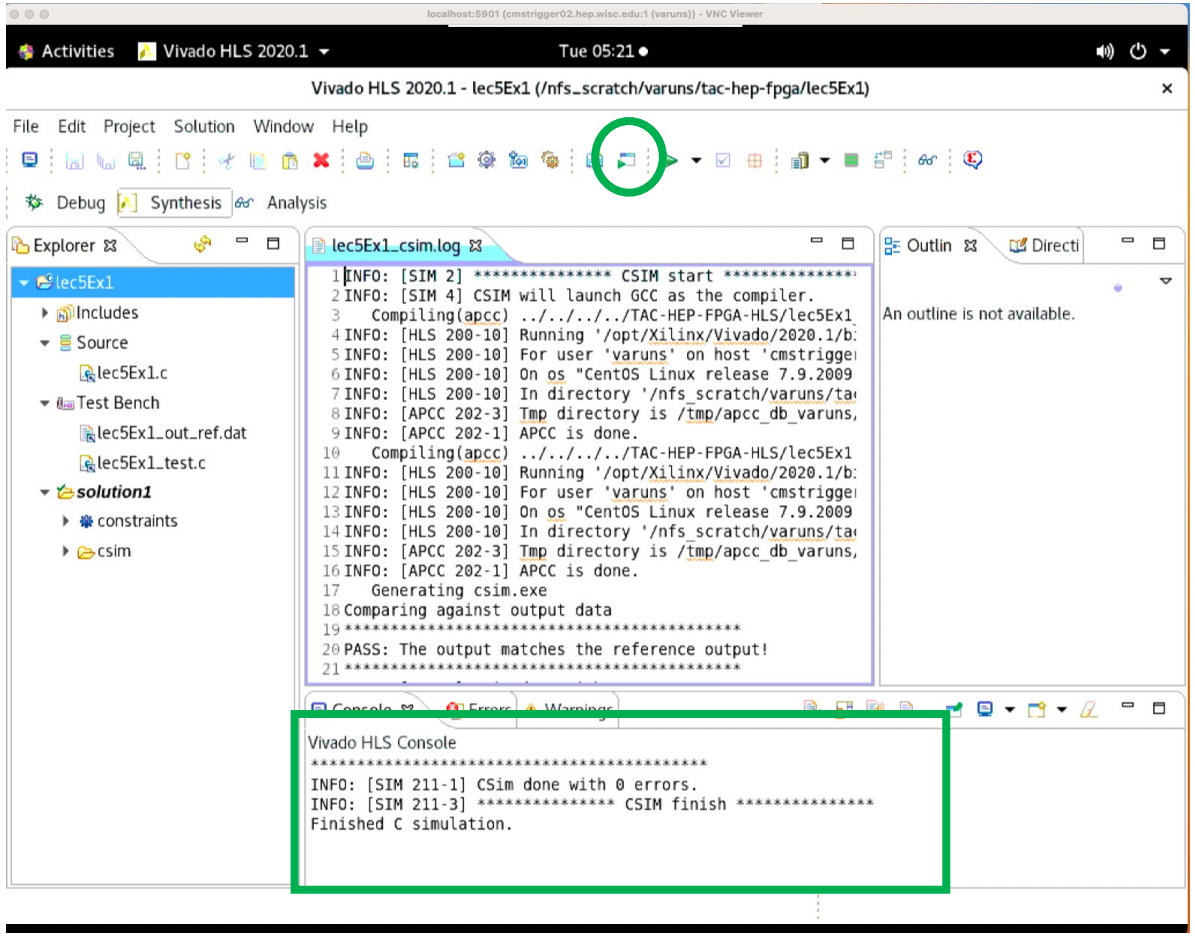
C-Simulation

Eg.: lec5Ex1

On-going…

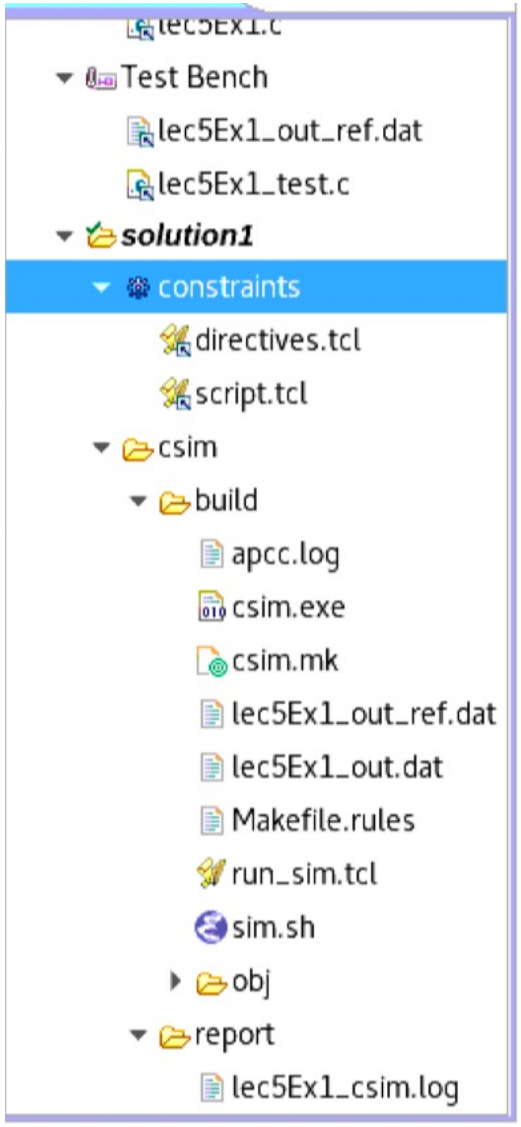


Finished…



C-simulation output

Eg.: lec5Ex1



csim/build is the primary location for all files related to the C-simulation

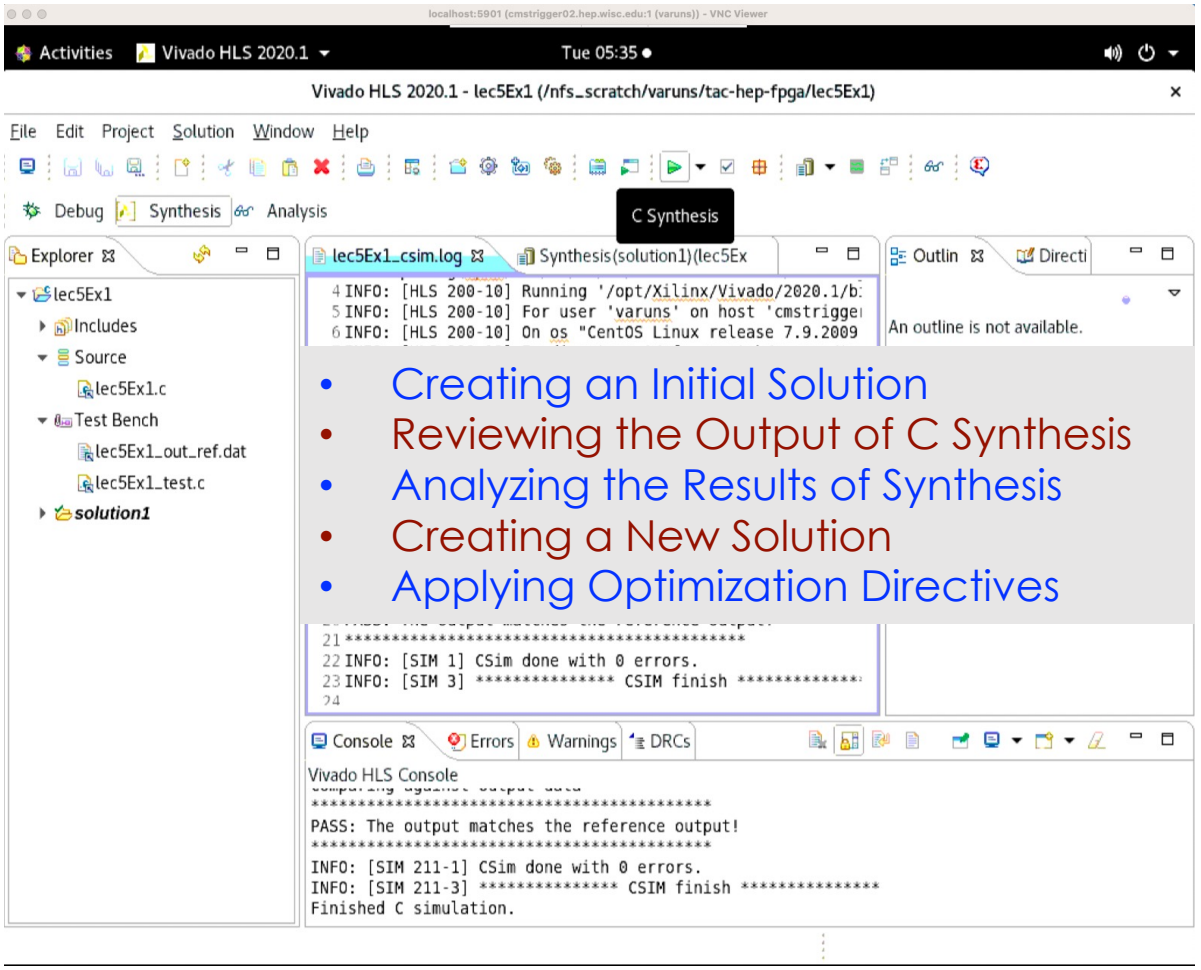
- Any files read by the test bench are copied to this folder
- The C executable file *csim.exe* is created and run in this folder
- Any files written by the test bench are created in this folder
  - Build Only option: exe file is not executed

Folder *csim/report* contains a log file of the C simulation

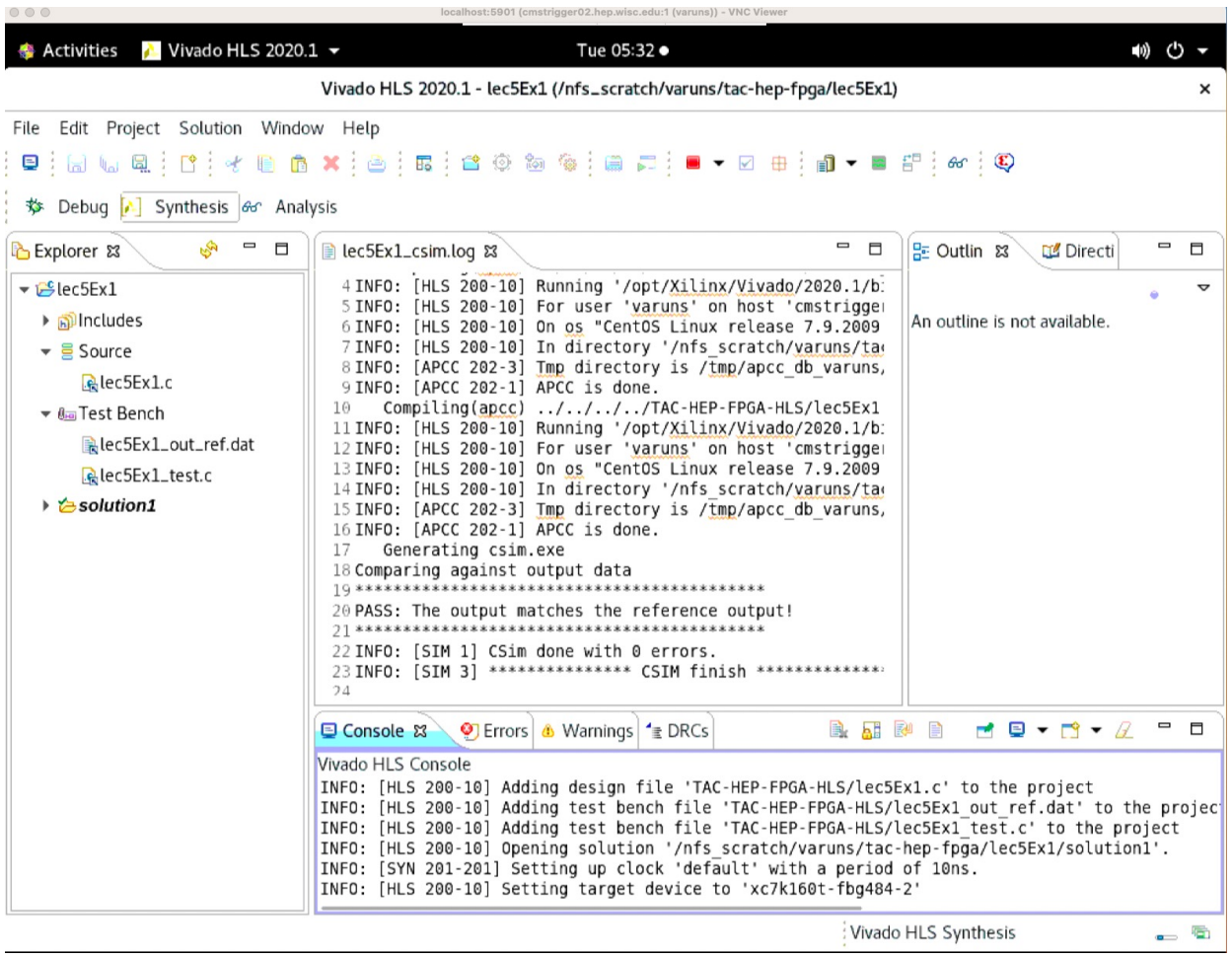
Next: execute synthesis

C-Synthesis

Eg.: lec5Ex1



- Creating an Initial Solution
- Reviewing the Output of C Synthesis
- Analyzing the Results of Synthesis
- Creating a New Solution
- Applying Optimization Directives



C-Synthesis: *On-going*

Eg.: lec5Ex1

Within GUI, some messages may contain links to enhanced information
Clicking them provides more details on why message was issues and possible resolution

```
Vivado HLS Console
/opt/Xilinx/Vivado/2020.1/bin/vivado_hls /nfs_scratch/varuns/tac-hep-fpga/lec5Ex1/solution1/cs
INFO: [HLS 200-10] Running '/opt/Xilinx/Vivado/2020.1/bin/unwrapped/lnx64.o/vivado_hls'
INFO: [HLS 200-10] For user 'varuns' on host 'cmstrigger02.hep.wisc.edu' (Linux_x86_64 version
INFO: [HLS 200-10] On os "CentOS Linux release 7.9.2009 (Core)"
INFO: [HLS 200-10] In directory '/nfs_scratch/varuns/tac-hep-fpga'
Sourcing Tcl script '/nfs_scratch/varuns/tac-hep-fpga/lec5Ex1/solution1/csynth.tcl'
INFO: [HLS 200-10] Opening project '/nfs_scratch/varuns/tac-hep-fpga/lec5Ex1'.
INFO: [HLS 200-10] Adding design file 'TAC-HEP-FPGA-HLS/lec5Ex1.c' to the project
INFO: [HLS 200-10] Adding test bench file 'TAC-HEP-FPGA-HLS/lec5Ex1_out_ref.dat' to the projec
INFO: [HLS 200-10] Adding test bench file 'TAC-HEP-FPGA-HLS/lec5Ex1_test.c' to the project
INFO: [HLS 200-10] Opening solution '/nfs_scratch/varuns/tac-hep-fpga/lec5Ex1/solution1'.
INFO: [SYN 201-201] Setting up clock 'default' with a period of 10ns.
INFO: [HLS 200-10] Setting target device to 'xc7k160t-fbg484-2'
INFO: [SYN 201-201] Setting up clock 'default' with a period of 10ns.
INFO: [SCHED 204-61] Option 'relax_ii_for_timing' is enabled, will increase II to preserve clo
INFO: [HLS 200-10] Analyzing design file 'TAC-HEP-FPGA-HLS/lec5Ex1.c' ...
INFO: [HLS 200-111] Finished Linking Time (s): cpu = 00:00:13 ; elapsed = 00:00:13 . Memory (MI
INFO: [HLS 200-111] Finished Checking Pragmas Time (s): cpu = 00:00:13 ; elapsed = 00:00:13 . I
INFO: [HLS 200-10] Starting code transformations ...
INFO: [HLS 200-111] Finished Standard Transforms Time (s): cpu = 00:00:15 ; elapsed = 00:00:15
INFO: [HLS 200-10] Checking synthesizability ...
INFO: [HLS 200-111] Finished Checking Synthesizability Time (s): cpu = 00:00:15 ; elapsed = 00
INFO: [HLS 200-111] Finished Pre-synthesis Time (s): cpu = 00:00:15 ; elapsed = 00:00:15 . Mem
INFO: [HLS 200-472] Inferring partial write operation for 'arr' (TAC-HEP-FPGA-HLS/lec5Ex1.c:17
INFO: [HLS 200-472] Inferring partial write operation for 'arr' (TAC-HEP-FPGA-HLS/lec5Ex1.c:20
```

C-Synthesis: *On-going*

Eg.: lec5Ex1

Within GUI, some mes
Clicking them provide esolution

```
Vivado HLS Console
/opt/Xilinx/Vivado/202
INFO: [HLS 200-10] Run
INFO: [HLS 200-10] For
INFO: [HLS 200-10] On
INFO: [HLS 200-10] In
Sourcing Tcl script '/
INFO: [HLS 200-10] Ope
INFO: [HLS 200-10] Add
INFO: [HLS 200-10] Add
INFO: [HLS 200-10] Add
INFO: [HLS 200-10] Ope
INFO: [SYN 201-201] Se
INFO: [HLS 200-10] Set
INFO: [SYN 201-201] Se
INFO: [SCHED 204-61] C
INFO: [HLS 200-10] Ana
INFO: [HLS 200-111] Fi
INFO: [HLS 200-111] Fi
INFO: [HLS 200-10] Sta
INFO: [HLS 200-111] Fi
INFO: [HLS 200-10] Che
INFO: [HLS 200-111] Fi
INFO: [HLS 200-111] Fi
INFO: [HLS 200-472] In
INFO: [HLS 200-472] In
```

Message: SCHED 204-61 ×

Message: SCHED 204-61

indicates the final performance achieved by the PIPELINE directive, when issued at the end of the pipeline process.

Explanation

When the PIPELINE directive is used on a loop or function, Vivado HLS seeks to pipeline the loop/function with the specified Initiation Interval (II) using option -II. The II is the number of clock cycles between reading new input values. The fastest possible design has II=1: it reads new inputs every clock cycle. If no II is specified, an II=1 is assumed.

This message specifies the required II (Target), the actual II achieved (Final) and the latency (DEPTH) of the final pipeline.

Solution

If the Final II does not satisfy the requirements, refer to messages SCHED 204-68 or SCHED 204-69 issued before the previous occurrence of this message.

OK

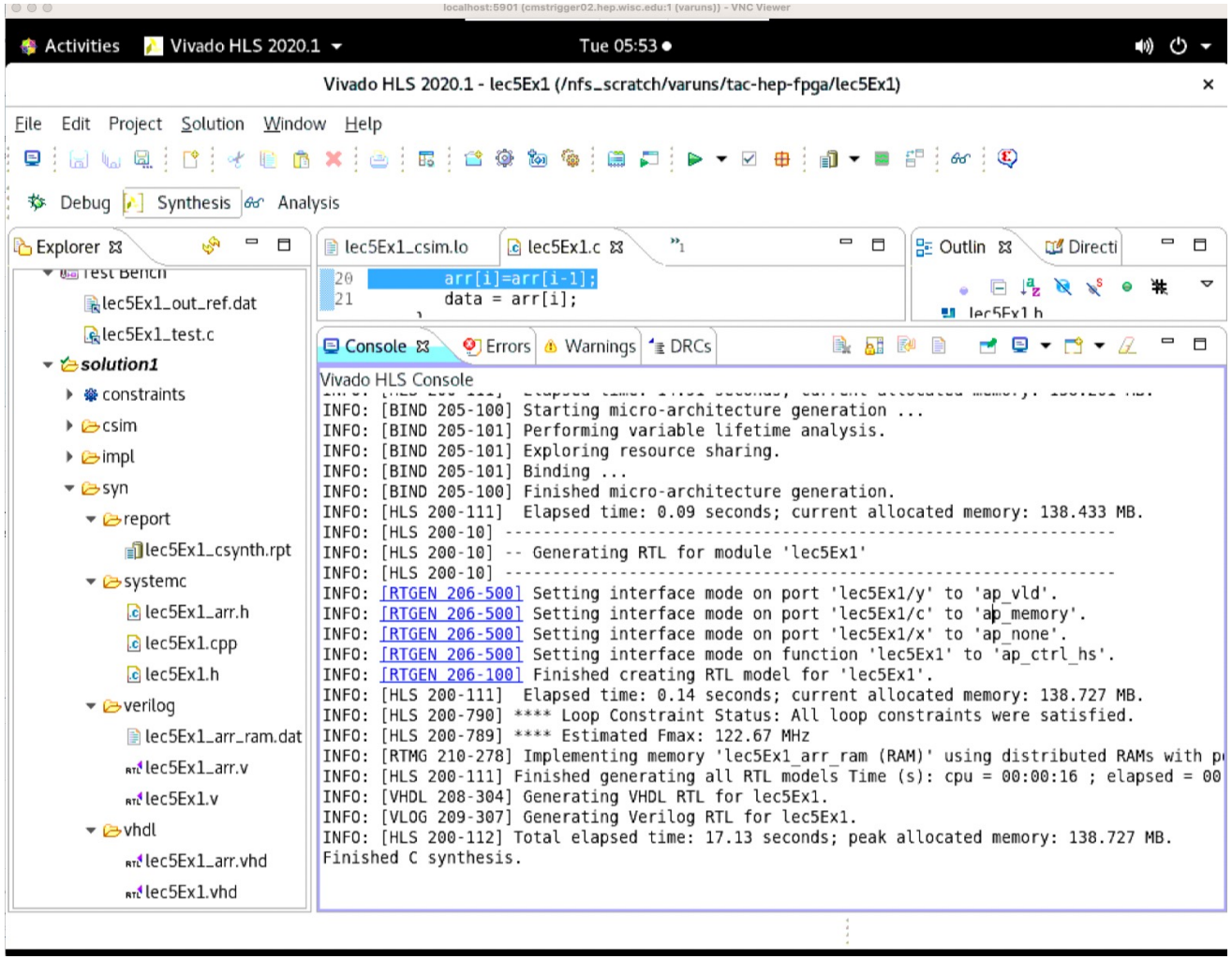
C-Synthesis: *On-going*

Eg.: lec5Ex1

```
Vivado HLS Console
INFO: [HLS 200-10] Synthesizing 'lec5Ex1' ...
INFO: [HLS 200-10] ------------------------------------------------------------
INFO: [HLS 200-42] -- Implementing module 'lec5Ex1'
INFO: [HLS 200-10] ------------------------------------------------------------
INFO: [SCHED 204-11] Starting scheduling ...
INFO: [SCHED 204-11] Finished scheduling.
INFO: [HLS 200-111]  Elapsed time: 14.91 seconds; current allocated memory: 138.261 MB.
INFO: [BIND 205-100] Starting micro-architecture generation ...
INFO: [BIND 205-101] Performing variable lifetime analysis.
INFO: [BIND 205-101] Exploring resource sharing.
INFO: [BIND 205-101] Binding ...
INFO: [BIND 205-100] Finished micro-architecture generation.
INFO: [HLS 200-111]  Elapsed time: 0.09 seconds; current allocated memory: 138.433 MB.
INFO: [HLS 200-10] ------------------------------------------------------------
INFO: [HLS 200-10] -- Generating RTL for module 'lec5Ex1'
INFO: [HLS 200-10] ------------------------------------------------------------
INFO: [RTGEN 206-500] Setting interface mode on port 'lec5Ex1/y' to 'ap_vld'.
INFO: [RTGEN 206-500] Setting interface mode on port 'lec5Ex1/c' to 'ap_memory'.
INFO: [RTGEN 206-500] Setting interface mode on port 'lec5Ex1/x' to 'ap_none'.
INFO: [RTGEN 206-500] Setting interface mode on function 'lec5Ex1' to 'ap_ctrl_hs'.
INFO: [RTGEN 206-100] Finished creating RTL model for 'lec5Ex1'.
INFO: [HLS 200-111]  Elapsed time: 0.14 seconds; current allocated memory: 138.727 MB.
INFO: [HLS 200-790] **** Loop Constraint Status: All loop constraints were satisfied.
INFO: [HLS 200-789] **** Estimated Fmax: 122.67 MHz
INFO: [RTMG 210-278] Implementing memory 'lec5Ex1_arr_ram (RAM)' using distributed RAMs with p
```

C-Synthesis: *Finished*

Eg.: lec5Ex1



Synthesis Report for 'lec5Ex1'

General Information

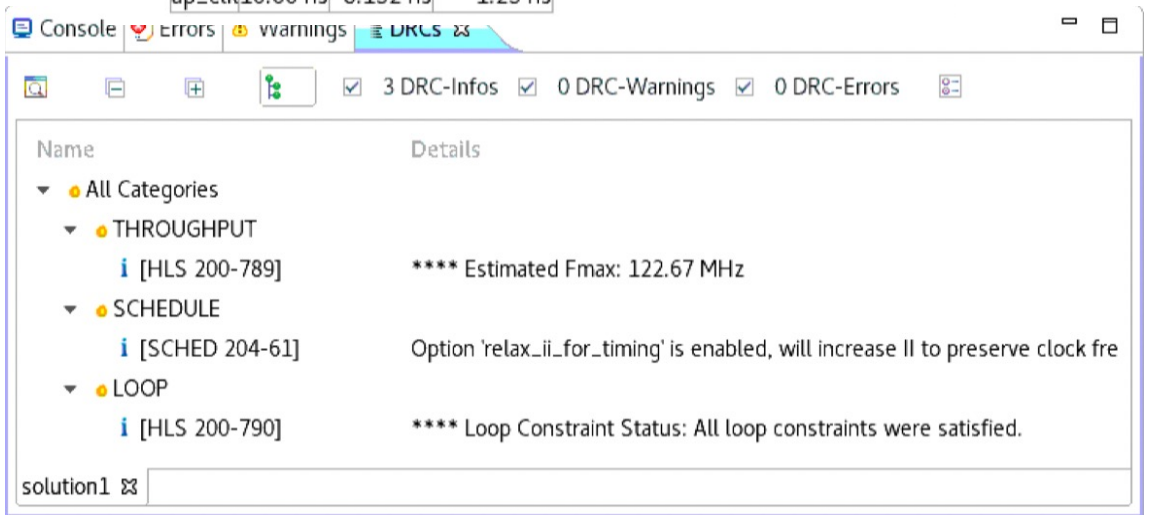
Date:	Tue Apr 4 05:32:18 2023
Version:	2020.1 (Build 2897737 on Wed May 27 20:21:37 MDT 2020)
Project:	lec5Ex1
Solution:	solution1
Product family:	kintex7
Target device:	xc7k160t-fbg484-2

Performance Estimates

Timing

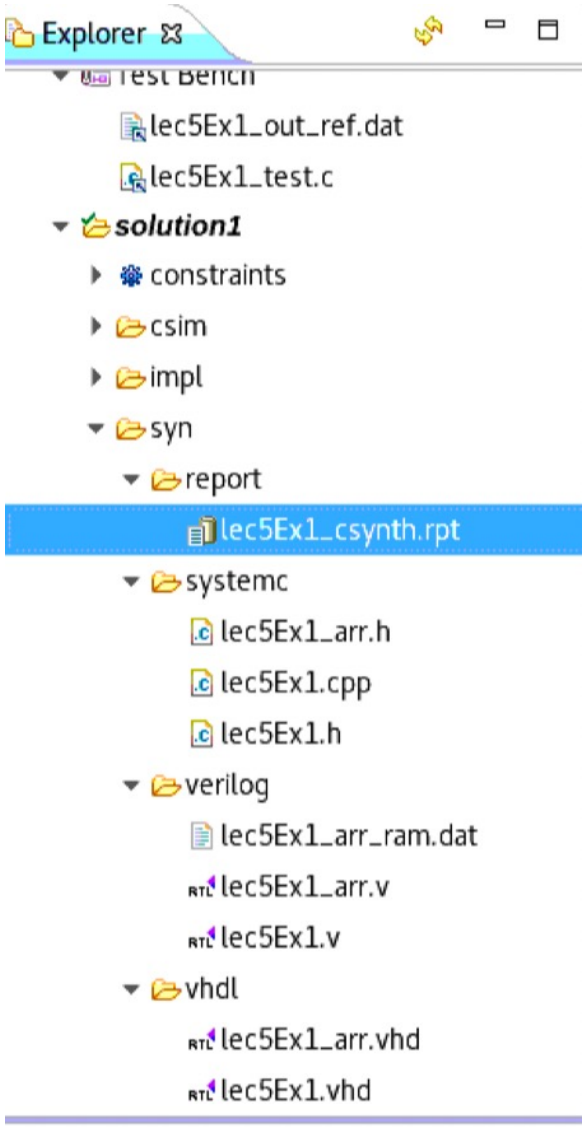
Summary

Clock	Target	Estimated	Uncertainty
ap_clk	10.00 ns	8.152 ns	1.25 ns



C-Synthesis: *Review the output*

Eg.: lec5Ex1



Folder *syn* is now available in the solution folder

report folder contains a report file for the top-level function and one for every sub-function in the design

verilog, *vhdl*, and *systemc* folders contain the output RTL files

The top-level file has the same name as the top-level function for synthesis

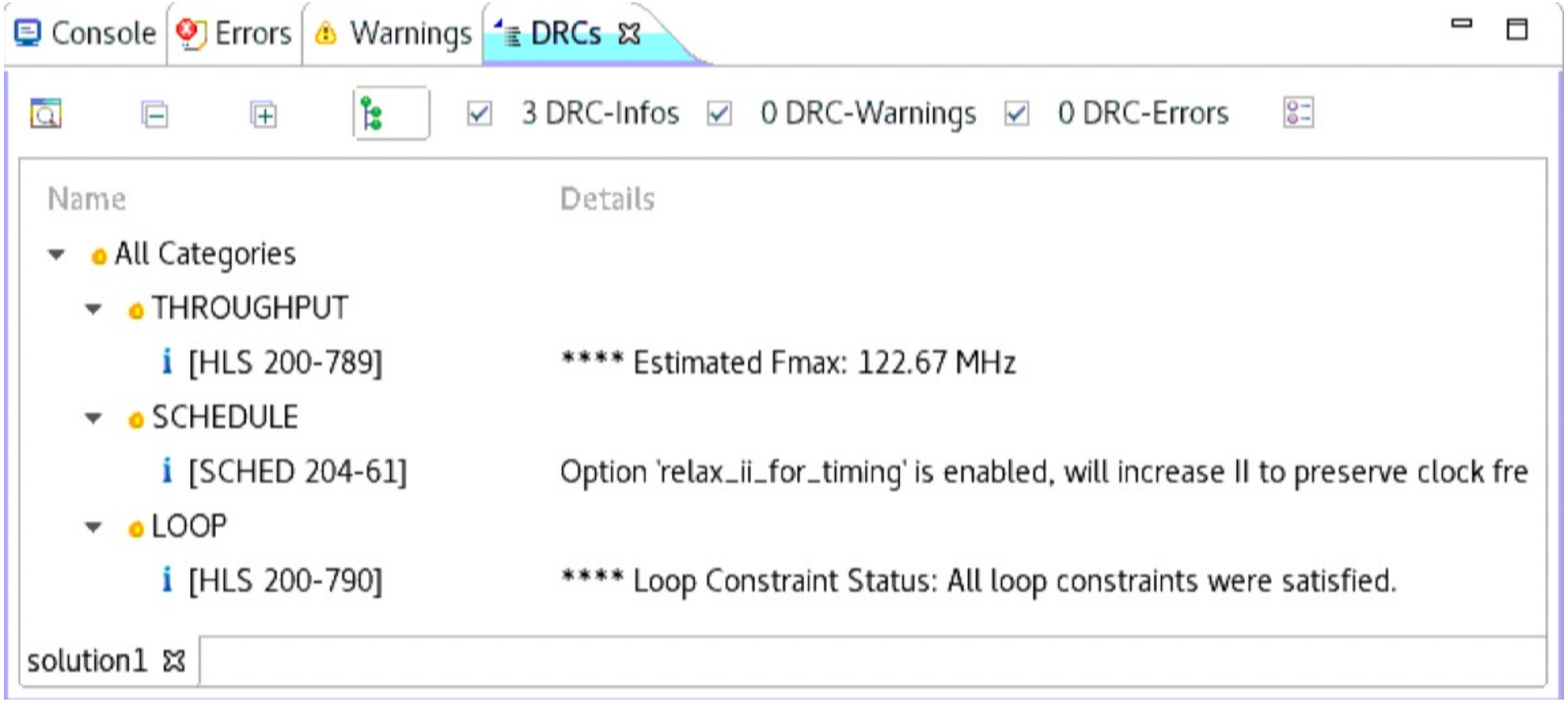
One RTL file for each function

Might be additional RTL files to implement sub-blocks (block RAM, pipelined multipliers, etc)

C-Synthesis: *Review the output*

Eg.: lec5Ex1

Design Rule Checks



C-Synthesis: *Review the output*

Eg.: lec5Ex1

Synthesis Report for 'lec5Ex1'

General Information

Date:	Tue Apr 4 05:32:18 2023
Version:	2020.1 (Build 2897737 on Wed May 27 20:21:37 MDT 2020)
Project:	lec5Ex1
Solution:	solution1
Product family:	kintex7
Target device:	xc7k160t-fbg484-2

Performance Estimates

Timing

Summary

Clock	Target	Estimated	Uncertainty
ap_clk	10.00 ns	8.152 ns	1.25 ns

Latency

Summary

Latency (cycles)		Latency (absolute)		Interval (cycles)		
min	max	min	max	min	max	Type
34	34	0.340 us	0.340 us	34	34	none

General Information: Details on when the results were generated, software version used, project name, solution name, & technology details

Timing: target clock frequency, clock uncertainty, & the estimate of the fastest achievable clock frequency

Latency Summary:

- Reports the latency and II for this block and any sub-blocks instantiated in this block
- Each sub-function called at this level in the C source is an instance in this RTL block unless it was inlined
- Latency: # of CLK cycle to get output
- II: # of CLK cycle before new inputs can arrive

- Without PIPELINE directives: Latency = II -1
  - Next input is read when the final output is written

C-Synthesis: *Review the output*

Eg.: lec5Ex1

- Latency
  - Summary

Latency (cycles)		Latency (absolute)		Interval (cycles)		
min	max	min	max	min	max	Type
34	34	0.340 us	0.340 us	34	34	none

  - Detail
    - Instance

N/A

    - Loop

	Latency (cycles)			Initiation Interval			
Loop Name	min	max	Iteration Latency	achieved	target	Trip Count	Pipelined
- Loop	33	33	3	-	-	11	no

Latency Detail:

- Latency and II for the instances sub-functions & loops in this block
  - For sub-loops, the loop hierarchy is shown
- Min and max latency values indicate the latency to execute all iterations of the loop
  - Conditional branches might make the min and max different

- *Iteration Latency*: latency for a single iteration of the loop
- For variable loop latency, values can't be determined and are shown as a "?"
- Any specified target initiation interval is shown beside the actual initiation interval achieved
- The *tripcount* shows the total number of loop iterations

C-Synthesis: *Review the output*

Eg.: lec5Ex1

- Latency
  - Summary

Latency (cycles)		Latency (absolute)		Interval (cycles)		
min	max	min	max	min	max	Type
34	34	0.340 us	0.340 us	34	34	none

  - Detail
    - Instance

N/A

    - Loop

	Latency (cycles)			Initiation Interval			
Loop Name	min	max	Iteration Latency	achieved	target	Trip Count	Pipelined
- Loop	33	33	3	-	-	11	no

Latency Detail:

- Latency and II for the instances sub-functions & loops in this block
  - For sub-loops, the loop hierarchy is shown
- Min and max latency values indicate the latency to execute all iterations of the loop
  - Conditional branches might make the min and max different

- No sub-blocks in this design.: Expanding the Instance section shows no sub-modules in the hierarchy
- All the delay is due to the RTL logic synthesized from the loop named "Loop"
- Total latency is one clock cycle greater than the loop latency
  - Requires one clock cycle to enter and exit the loop (in this case, the design finishes when the loop finishes, so there is no exit cycle)

C-Synthesis: *Review the output*

Eg.: lec5Ex1

Utilization Estimates

Summary

Name	BRAM_18K	DSP48E	FF	LUT	URAM
DSP	-	-	-	-	-
Expression	-	3	0	86	-
FIFO	-	-	-	-	-
Instance	-	-	-	-	-
Memory	0	-	64	6	0
Multiplexer	-	-	-	91	-
Register	-	-	111	-	-
Total	0	3	175	183	0
Available	650	600	202800	101400	0
Utilization (%)	0	~0	~0	~0	0

Detail

- Instance
- DSP48E
- Memory
- FIFO
- Expression
- Multiplexer
- Register

Report shows the resources: LUTS, Flip-Flops, DSPs, used to implement the design

Instance: Resources used by sub-blocks instantiated at this hierarchy

- If no RTL hierarchy, no instances reported
- For any instance, detail report is presented

Memory: Resources used in implementation of memories

FIFO: Resources used in implementation of any FIFO

Expressions: Resources used by any expressions such as multipliers, adders, and comparators along with bit widths of the input ports to the expressions

Multiplexers: Resources used to implement multiplexors along with bit-widths of input ports

Registers: List of all shift registers used

C-Synthesis: *Review the output*

Eg.: lec5Ex1

Utilization Estimates

- Summary

Name	BRAM_18K	DSP48E	FF	LUT	URAM
DSP	-	-	-	-	-
Expression	-	3	0	86	-
FIFO	-	-	-	-	-
Instance	-	-	-	-	-
Memory	0	-	64	6	0
Multiplexer	-	-	-	91	-
Register	-	-	111	-	-
Total	0	3	175	183	0
Available	650	600	202800	101400	0
Utilization (%)	0	~0	~0	~0	0

- Detail
  - Instance
  - DSP48E
  - Memory
  - FIFO
  - Expression
  - Multiplexer
  - Register

The design uses 3 DSP48s, 175 flip-flops and 183LUTs.

At this stage, the area numbers are estimates

RTL synthesis might be able to perform additional optimizations, and these figures might change after RTL synthesis

C-Synthesis: *Review the output*

Eg.: lec5Ex1

Interface

Summary

RTL Ports	Dir	Bits	Protocol	Source Object	C Type
ap_clk	in	1	ap_ctrl_hs	lec5Ex1	return value
ap_rst	in	1	ap_ctrl_hs	lec5Ex1	return value
ap_start	in	1	ap_ctrl_hs	lec5Ex1	return value
ap_done	out	1	ap_ctrl_hs	lec5Ex1	return value
ap_idle	out	1	ap_ctrl_hs	lec5Ex1	return value
ap_ready	out	1	ap_ctrl_hs	lec5Ex1	return value
y	out	32	ap_vld	y	pointer
y_ap_vld	out	1	ap_vld	y	pointer
c_address0	out	4	ap_memory	c	array
c_ce0	out	1	ap_memory	c	array
c_q0	in	32	ap_memory	c	array
x	in	32	ap_none	x	scalar

Export the report(.html) using the Export Wizard

Open Analysis Perspective Analysis Perspective

Shows how the the function arguments have been synthesized into RTL ports

The RTL port names are grouped with their protocol and source object: these are the RTL ports created when that source object is synthesized with the stated I/O protocol

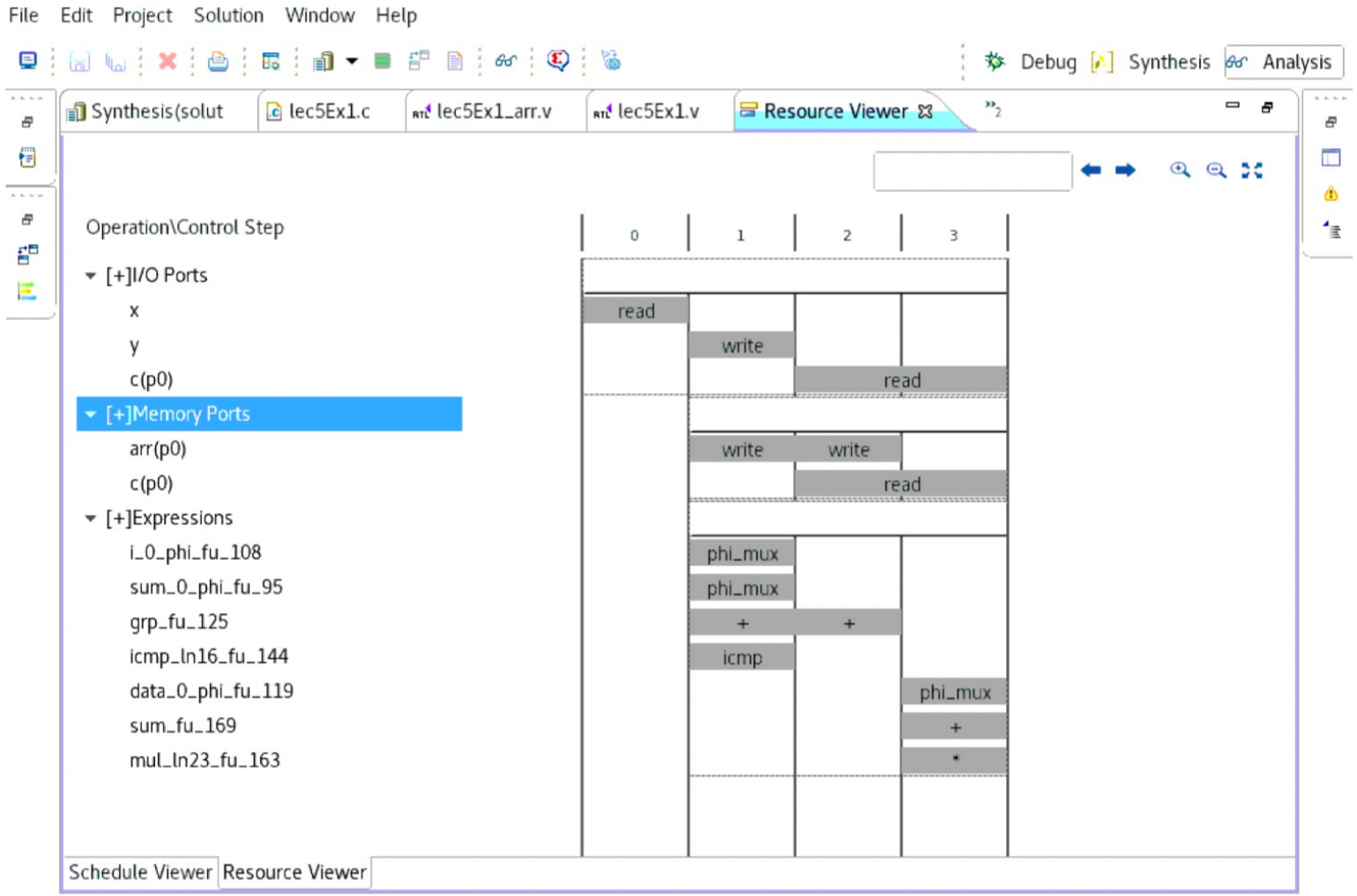
The design has a clock and reset port

Synthesis has automatically added some block level control ports : *ap_start*, *ap_done*, *ap_idle* and *ap_ready*

Analysis Perspective

Eg.: lec5Ex1

Analysis Perspective to analyze the results

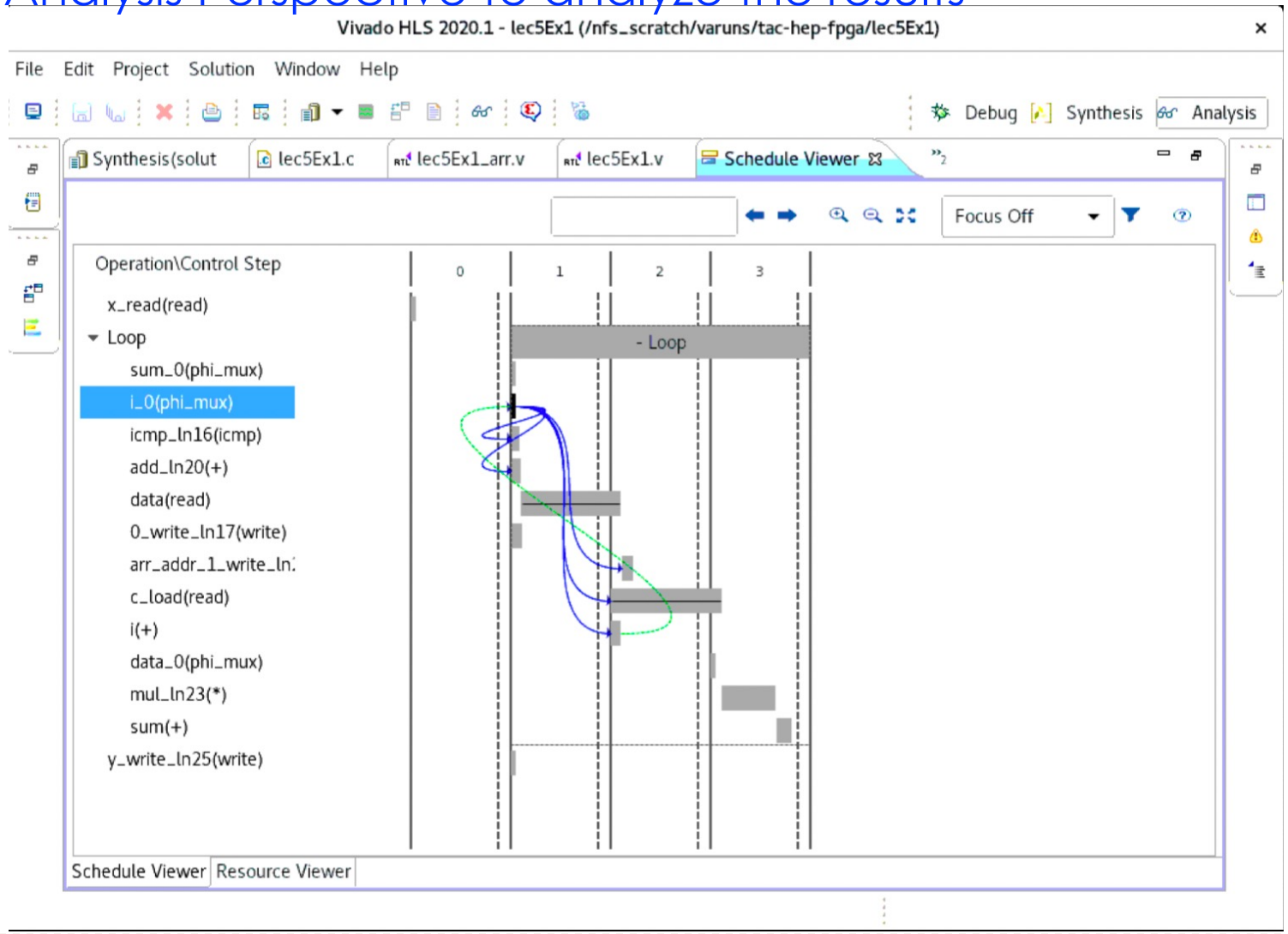


Resource view

Analysis Perspective

Eg.: lec5Ex1

Analysis Perspective to analyze the results



Schedule viewer

Vivado HLS via CLI

Eg.: lec5Ex2

```
1  open_project lec5Ex2
2  set_top lec5Ex1
3  add_files lec5Ex1.c
4  add_files -tb lec5Ex1_test.c
5  add_files -tb lec5Ex1_out_ref.dat
6
7  open_solution "solution1"
8  set_part  {xc7k160tfbg484-2}
9  create_clock -period 10
10
11 #source "./lec5Ex2/solution1/directives.tcl"
12
13 csim_design
14 csynth_design
15 cosim_design
16 #export_design -format ip_catalog
17
18 # Exit Vivado HLS
19 exit
```

https://github.com/varuns23/TAC-HEP-FPGA-HLS/blob/main/lec5Ex2.tcl

Execute: *vivado_hls lec5Ex2.tcl*

Vivado HLS via CLI

What to expect…

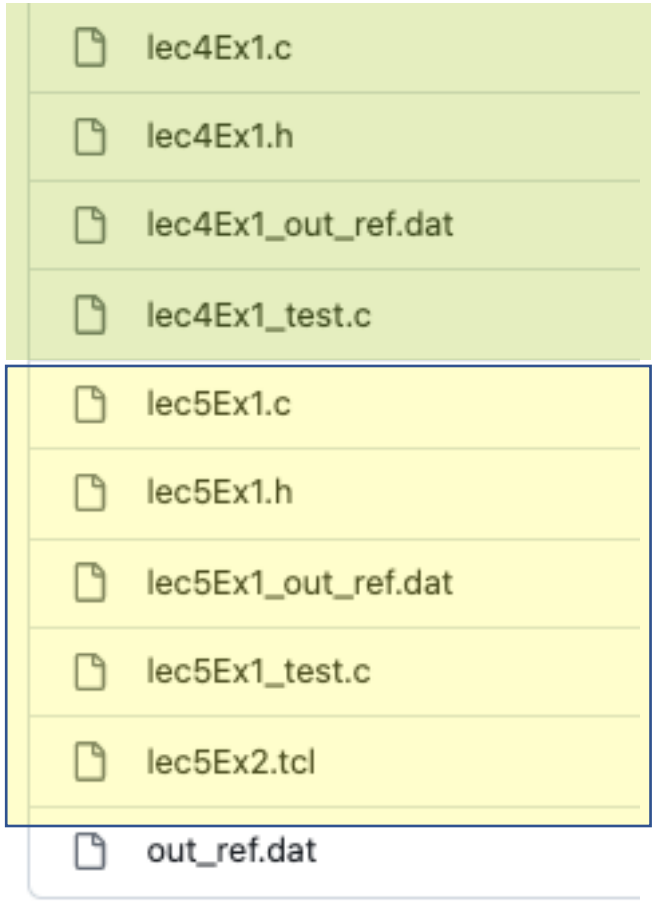
```
[varuns@cmstrigger02 TAC-HEP-FPGA-HLS]$ vivado_hls lec5Ex2.tcl

****** Vivado(TM) HLS - High-Level Synthesis from C, C++ and SystemC v2020.1 (64-bit)
  **** SW Build 2902540 on Wed May 27 19:54:35 MDT 2020
  **** IP Build 2902112 on Wed May 27 22:43:36 MDT 2020
    ** Copyright 1986-2020 Xilinx, Inc. All Rights Reserved.

source /opt/Xilinx/Vivado/2020.1/scripts/vivado_hls/hls.tcl -notrace
INFO: [HLS 200-10] Running '/opt/Xilinx/Vivado/2020.1/bin/unwrapped/lnx64.o/vivado_hls'
INFO: [HLS 200-10] For user 'varuns' on host 'cmstrigger02.hep.wisc.edu' (Linux_x86_64 version 5.4.236-1.el7.elrepo.x86_64) on Tue Apr 04 08:09:19 CDT 2023
INFO: [HLS 200-10] On os "CentOS Linux release 7.9.2009 (Core)"
INFO: [HLS 200-10] In directory '/nfs_scratch/varuns/tac-hep-fpga/TAC-HEP-FPGA-HLS'
Sourcing Tcl script 'lec5Ex2.tcl'
INFO: [HLS 200-10] Creating and opening project '/nfs_scratch/varuns/tac-hep-fpga/TAC-HEP-FPGA-HLS/lec5Ex2'.
INFO: [HLS 200-10] Adding design file 'lec5Ex1.c' to the project
INFO: [HLS 200-10] Adding test bench file 'lec5Ex1_test.c' to the project
INFO: [HLS 200-10] Adding test bench file 'lec5Ex1_out_ref.dat' to the project
INFO: [HLS 200-10] Creating and opening solution '/nfs_scratch/varuns/tac-hep-fpga/TAC-HEP-FPGA-HLS/lec5Ex2/solution1'.
INFO: [HLS 200-10] Setting target device to 'xc7k160t-fbg484-2'
INFO: [SYN 201-201] Setting up clock 'default' with a period of 10ns.
INFO: [SIM 211-2] *************** CSIM start ***************
INFO: [SIM 211-4] CSIM will launch GCC as the compiler.
   Compiling(apcc) ../../../../lec5Ex1_test.c in debug mode
INFO: [HLS 200-10] Running '/opt/Xilinx/Vivado/2020.1/bin/unwrapped/lnx64.o/apcc'
INFO: [HLS 200-10] For user 'varuns' on host 'cmstrigger02.hep.wisc.edu' (Linux_x86_64 version 5.4.236-1.el7.elrepo.x86_64) on Tue Apr 04 08:09:25 CDT 2023
INFO: [HLS 200-10] On os "CentOS Linux release 7.9.2009 (Core)"
INFO: [HLS 200-10] In directory '/nfs_scratch/varuns/tac-hep-fpga/TAC-HEP-FPGA-HLS/lec5Ex2/solution1/csim/build'
INFO: [APCC 202-3] Tmp directory is /tmp/apcc_db_varuns/31896168061376550128 2
INFO: [APCC 202-1] APCC is done.
   Compiling(apcc) ../../../../lec5Ex1.c in debug mode
INFO: [HLS 200-10] Running '/opt/Xilinx/Vivado/2020.1/bin/unwrapped/lnx64.o/apcc'
INFO: [HLS 200-10] For user 'varuns' on host 'cmstrigger02.hep.wisc.edu' (Linux_x86_64 version 5.4.236-1.el7.elrepo.x86_64) on Tue Apr 04 08:09:38 CDT 2023
INFO: [HLS 200-10] On os "CentOS Linux release 7.9.2009 (Core)"
INFO: [HLS 200-10] In directory '/nfs_scratch/varuns/tac-hep-fpga/TAC-HEP-FPGA-HLS/lec5Ex2/solution1/csim/build'
INFO: [APCC 202-3] Tmp directory is /tmp/apcc_db_varuns/32026168061377840338 6
INFO: [APCC 202-1] APCC is done.
   Generating csim.exe
```

Exercise files

- Git clone: https://github.com/varuns23/TAC-HEP-FPGA-HLS.git



Assignment

- Use target device: **xc7k160tfbg484-2**
- Clock period of 10ns

1. Execute the code (lec5Ex2.tcl) using CLI (slide-25) and compare the results with GUI results for C-Simulation, C-Synthesis

2. Vary following parameters for two cases: high and very high values and compare with 1 for both CLI and GUI
    - Variable: "samples"
    - Variable: "N"

Assignment submission

- Where to submit:
  - https://pages.hep.wisc.edu/~varuns/assignments/TAC-HEP/
- Use your login machine credentials
- Submit one file per week
  - Week-2 & 3 can be merged together
- Try to submit by next Tuesday

TAC-HEP 2023

Questions?

TAC-HEP 2023

Additional material

Correct Time

From 03.28.2023 onwards

- Tuesdays: 9:00-10:00 CT / 10:00-11:00 ET / 16:00-17:00 CET
- Wednesday: 11:00-12:00 CT / 12:00-13:00 ET / 18:00-19:00 CET

Jargons

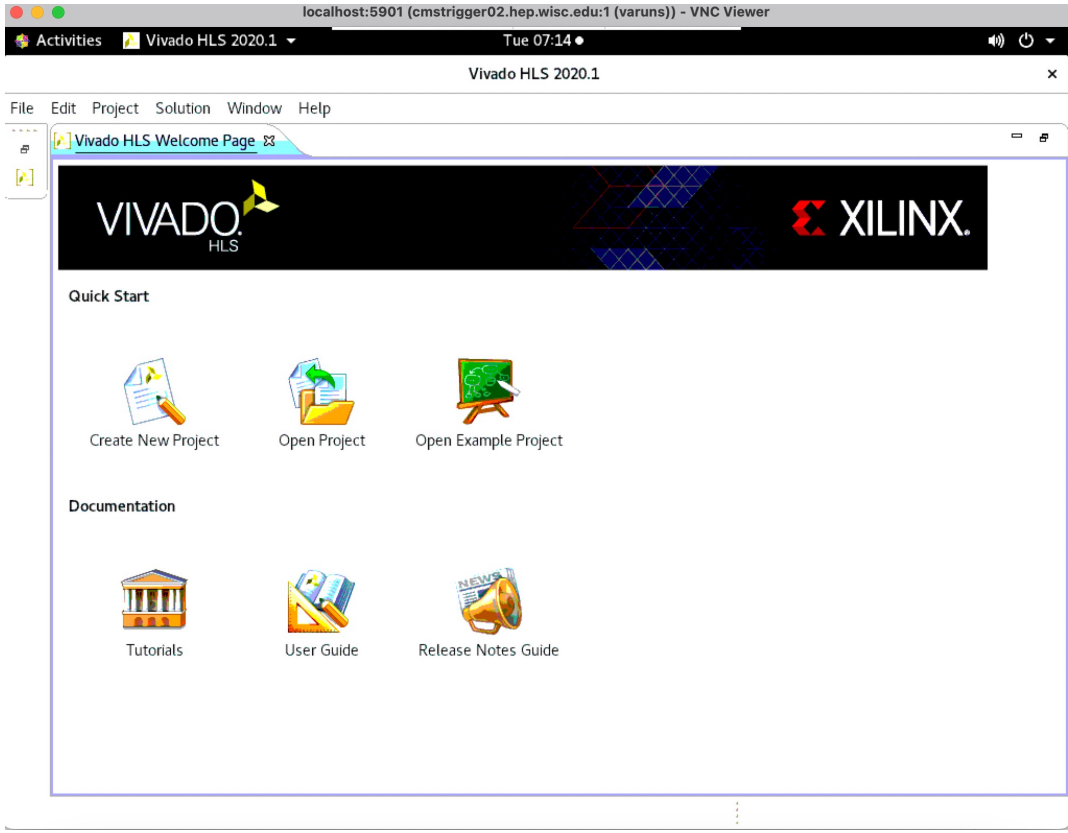
- **ICs - Integrated chip:** assembly of hundreds of millions of transistors on a minor chip
- **PCB:** Printed Circuit Board
- **LUT - Look Up Table aka 'logic'** - generic functions on small bitwidth inputs. Combine many to build the algorithm
- **FF - Flip Flops** - control the flow of data with the clock pulse. Used to build the pipeline and achieve high throughput
- **DSP - Digital Signal Processor** - performs multiplication and other arithmetic in the FPGA
- **BRAM - Block RAM** - hardened RAM resource. More efficient memories than using LUTs for more than a few elements
- **PCIe or PCI-E - Peripheral Component Interconnect Express**: is a serial expansion bus standard for connecting a computer to one or more peripheral devices
- **InfiniBand** is a computer networking communications standard used in high-performance computing that features very high throughput and very low latency
- **HLS -** High Level Synthesis - compiler for C, C++, SystemC into FPGA IP cores
- **DRCs -** Design Rule Checks
- **HDL** - Hardware Description Language - low level language for describing circuits
- **RTL** - Register Transfer Level - the very low level description of the function and connection of logic gates
- **FIFO** – First In First Out memory
- **Latency** - time between starting processing and receiving the result
  - Measured in clock cycles or seconds
- **II - Initiation Interval** - time from accepting first input to accepting next input

All set for hands-on

Everytime

Summary

- ssh varuns@cmstrigger02-via-login -L5901:localhost:5901
  - Or whatever ***:1*** display number
  - Sometimes you may need to run vncserver -localhost -geometry 1024x768 again to start new vnc server
- Connect to VNC server (remote desktop) client
- Open terminal
- Source /opt/Xilinx/Vivado/2020.1/settings64.sh
- vivado_hls



Homework: You are able to connect and bring this screen
Let me know in case of any issue