# INTRODUCTION TO FPGA PROGRAMMING

LESSON 14: ADVANCED VIVADO USAGE

Dr. Davide Cieri<sup>1</sup>

<sup>1</sup>Max-Planck-Institut für Physik, Munich

September 2024



#### **REMINDER: VIVADO FLOW**

#### 1. HDL Design

- 2. Behavioural Simulation
- 3. Synthesis
- 4. Post-Synthesis Simulation
- 5. Implementation
- 6. Bitstream Generation

PROJECT MANAGER

 Add Sources
 Language Templates
 OP IP catalog

 PI INTEGRATOR
 Create Block Design
 Generate Block Design
 Generate Block Design
 Generate Block Design
 SIMULATION
 Run Simulation

 KTLANALYSIS

- > Open Elaborated Design
- ✓ SYNTHESIS
  - Run Synthesis
  - > Open Synthesized Design
- ✓ IMPLEMENTATION
  - Run Implementation
  - > Open Implemented Design
- ✓ PROGRAM AND DEBUG
  - Generate Bitstream
  - > Open Hardware Manager

#### **VIVADO PROJECTS**

- Vivado projects are device specific
  - Board or FPGA part (down to to the speed grade)
- The project include all the source file to compile your design
  - Design Sources (RTL, IPs), Simulation, Constraints
- Supported Languages in Vivado Synthesis 2022.2
  - SystemVerilog:(IEEE Std 1800-2012
  - Verilog: (IEEE Std 1364-2005)
  - VHDL (2008)

### SYNTHESIS CONFIGURATION

- The synthesis process can be highly configured, by changing several Settings
- Settings > Synthesis
- Vivado provides several predefined strategies, for different applications
- Full list of settings here

Q.					
Project Settings	Synthesis Specify various settings as	sociated to Synthes	is	4	
Simulation Elaboration Dataflow	Constraints	strs_1 (active)		~	
Synthesis	Report Settings				
Implementation Bitstream	Strategy: 🌆 Vivado	Synthesis Default F	teports (Vivado Synthesis 2022)	~	
> IP	Settings				
Tool Settings Project	Incremental synthesis:	Automatically se	ected checkpoint	··· î	
IP Defaults	Strategy: 🔓 Vivado Synthesis Defaults		sis Defaults (Vivado Synthesis 2022)	~ 🔛	
> vivado store	Description:	Vivado Synthesis	Defaults		
Display	V Sunth Design Inivado				
Help	tcl.pre	,			
> Text Editor	tcl.post				
3rd Party Simulat	-flatten_hierarchy		rebuilt	*	
> Colors	-gated_clock_conver	rsion	off	~	
Selection Rules	-bufg		12		
Shortcuts	-directive		Default	~	
> Strategies	-retiming				
> Remote Hosts	-no_retiming				
> WINDOW Benavior	Select an option above to	see a description o	fit		
$\langle $					

### SYNTHESIS SETTINGS

- Most used synthesis settings:
  - tcl.pre and tcl.post: Run a custom tcl script before/after the synthesis
  - flatten\_hierarchy: none (same hierarchy as the RTL), full (flats full hierarchy), rebuilt (flats the hierarchy and rebuilds it after synthesis)
  - gated\_clock\_conversion: allows synthesis of clocked logic with enables

*	Settings		~ ^		
ą.	Sunthasis				
Project Settings	Specify various settings associated to Sy	nthesis	- A		
General					
Simulation	Constraints				
Elaboration					
Dataflow	Constraints 🖾 constrs_1 (active)		~		
Synthesis	Report Settings				
Implementation	former in the state of the state	Report seconds			
Bitstream	Strategy: In vivado synthesis Der	aut Reports (vivado Synthesis 2022)	Ŷ		
> IP	Settings				
ool Settings	≤ Synth Design (vivado)		^		
Project	tcl.pre				
IP Defaults	tcl.post				
Vivado Store	-flatten_hierarchy	rebuilt	~		
Source File	-gated_clock_conversion	off	× II		
Display	-bufg	12			
Help	-directive	Default	~		
> Text Editor	-retiming				
3rd Party Simulators	-no_retiming				
> Colors	-fsm_extraction	auto	~		
Selection Rules	-keep_equivalent_registers				
Shortcuts	-resource_sharing	auto	~		
> strategies	-control_set_opt_threshold	auto	~		
Window Behavior	-no_lc	0			
THING TO BEIGHT	-no_srlextract				
	-shreg_min_size	3			
	-max_bram	-1			
	-max_uram	-1			
	-max_dsp	-1			
	-max_bram_cascade_height	-1			
	-max_uram_cascade_height	-1			
	-cascade_dsp	auto	× _		
	Select an option above to see a descript	ion of it			
	ок	Cancel Apply	Restore		

### SYNTHESIS SETTINGS

- Most used synthesis settings:
  - Directive: Run synthesis with different optimisations
  - bufg/max\_bram/max\_dsp: Set a limit on the maximum resources to be used
  - fsm\_extraction: Set the encoding scheme for all FSM in the design
  - retiming: Improve circuit timing performance

*	Settings		~ ^
a,	Synthesis		λ
Project Settings	Specify various settings associated to Sy	nthesis	
General			
Simulation	Constraints		
Elaboration	Constraints 🗇 constrs_1 (active)		~
Datanow			
Implementation	Report Settings		
Bitstream	Strategy: 🎄 Vivado Synthesis Def	ault Reports (Vivado Synthesis 2022)	~
> IP	Settings		
ool Settings	South Design (visado)		^
Project	tcl.pre		-
IP Defaults	tcl.post		
> Vivado Store	-flatten bierarchy	rebuit	~
Source File	-gated clock conversion	off	× 1
Display	-bufg	12	
Help	-directive	Default	~
> Text Editor	-retiming	0	
3rd Party Simulators	-no retiming		
> Colors	-fsm.extraction	auto	~
Selection Rules	-keep equivalent registers	0	
Shortcuts	-resource sharing	auto	~
> Strategies	-control set opt threshold	auto	~
> Remote Hosts	-no_lc		
> Window Behavior	-no_srlextract		
	-shreg_min_size	3	
	-max_bram	4	
	-max_uram	-1	
	-max_dsp	4	
	-max_bram_cascade_height	4	
	-max_uram_cascade_height	-1	
	-cascade_dsp	auto	v .
	Select an option above to see a descript	ion of it	
?)	ок	Cancel Apply	Restore

### SYNTHESIS RUN STRATEGIES

- Vivado provides preconfigured run strategies, for different requirements
- You can also save the current synthesis settings in a *User Defined Strategy*
- This will be available in any Vivado project, on your pc
- Strategies can be also exported, and shared with colleagues



#### SYNTHESIS ATTRIBUTES

- We already met a few attributes that you can set inside your VHDL to force behaviours in the synthesis (e.g. RAM\_STYLE, FSM\_ENCODING)
- Attributes can be set inside your VHDL, but also in the XDC constraint file

Syntax	Syntax		
<pre>attribute <attribute_name> : string;</attribute_name></pre>	set_property <attribute_name> &lt;</attribute_name>		
attribute of <object> : signal is <value></value></object>	ATTRIBUTE_VALUE> <object>;</object>		
Example	Example		
<pre>attribute ram_style : string;</pre>	<pre>set_property ram_style "distributed" [</pre>		
<pre>attribute ram_style of myram : signal is "</pre>	get_cells myram]		
distributed";			

#### VHDL Syntax

XDC Syntax

#### AVAILABLE SYNTHESIS ATTRIBUTES

- Full List here
- ASYNC\_REG: Informs the tool that a register is capable of receiving asynchronous data in the D input pin relative to the source clock (e.g. double-flopping synchroniser)
- DONT\_TOUCH: Prevents the tool to optimise away a signal
- MARK\_DEBUG: Add a signal for debug (creating an ILA)
- USE\_DSP: Force the tool to place logic into a DSP or not
- GATED\_CLOCK: To use in conjunction with the setting gated\_clock\_conversion. Informs the tool that the clock has a clock enable.

- Once our design has been synthesised, we know:
  - What logic blocks have been generated;
  - What delay each block adds.
- Two types of simulation can be run after synthesis:
  - Post-Synthesis Functional Simulation: faster, runs the simulation with the synthetised nets
  - Post-Synthesis Timing Simulation: Slower, add a 100ps delay to each storage element

#### VIVADO IMPLEMENTATION FLOW

- The Vivado implementation is divided in several steps:
  - 1. **Opt Design**: Optimises the logical design to make it easier to fit onto the FPGA;
  - 2. Power Opt Design (optional): Optmises the design to reduce power demands
  - 3. Place Design: Places the design onto FPGA, and replicates logic to improve timing
  - 4. **Post-Place Power Opt Design** (optional): Additional optimization to reduce power after placement.
  - 5. **Post-Place Phys Opt Design** (optional): Optimises logic and placement using estimated timing
  - 6. Route Design: Routes the design onto the FPGA
  - 7. Post-Route Phys Opt Design (optional): Optmises design using the actual routed delays

#### IMPLEMENTATION CONFIGURATION

?

- Also the implementation process can be configured
- **Settings > Implementation** •
  - **Each Implementation** stage can be configured with different directives
  - Tcl scripts can be run before/after each step
- Full list of settings here

		settings		0	0
2-	Implementation				
Project Settings General	Specify various settings associat	ed to Implementat	ion	!	<u> </u>
Simulation Elaboration	Constraints				
Synthesis	Default constraint set: 🗁 constrs_1 (active)				
Implementation Bitstream	Report Settings				
> IP	Strategy: 🔓 Vivado Imple	mentation Default	Reports (Vivado Implementation 2020)	,	1
Fool Settings	Settings				
IP Defaults	Incremental implementation:	Not set		•••	î
Source File	Strategy:	🔓 Vivado Imple	ementation Defaults (Vivado Implem 🕤	-	L
Display WebTalk	Description:	Default settings	for Implementation.		
Help	~Design Initialization (init_de	sign)			
Text Editor	tcl.pre			•••	
3rd Party Simulators	tcl.post				
Colors	~Opt Design (opt_design)				
Selection Rules	is_enabled		V		
Shortcuts	tcl.pre				
Strategies	tcl.post				
Remote Hosts	-verbose				
Manual Configurat	-directive		Default	~	
Cluster Configurat	More Options				
Window Behavior	~Power Opt Design (power_o	pt design)			
		description of the			~

#### **IMPLEMENTATION STRATEGIES**

- Similarly to the synthesis, several pre-configured implementation strategies are available
- User settings can be saved in a user run strategies as well



#### **DESIGN RUNS**



- Vivado allows you to create multiple design runs, to compare results of different strategies.
- Design runs can be created independently for synthesis and implementation

#### INCREMENTAL IMPLEMENTATION

- Large projects can take long time to implement (days)
- With incremental implementation,
   Place & Route starts from a reference checkpoint, before RTL changes
- Can speed up design implementation



X16627-040716

#### SCRIPTING WITH VIVADO

- Vivado can also be run in batch mode
- Everything we did in this course can be done using only TCL scripts
- Every command you launch in the GUI is printed in the TCL console
- An Export feature allows you to export all command to recreate the project in a tcl script (File > Project > Write Tcl)

```
# Open a Vivado project in batch mode
vivado -mode tcl project.xpr
# Source a TCL script with Vivado in batch mode
vivado -mode batch -source mytclscript.tcl
```

#### TCL EXAMPLE

lab22.tcl	_ D @ X
/home/dcieri/Work/fpga-course-tum/labs-solutions/lab22/sim/lab22.tcl	×
Q 🔛 ♠ ≫ X 🗉 🛍 🗙 // 💷 Q	•
142 create project \${_xil_proj_name} ./\${_xil_proj_name} - part xc7a35tcpg236-1	~
143	
14; set the directory path for the new project 14; set proj dir (get_property directory (current_project))	
149; set project properties	
149 set_property -name "board_part_repo_paths" -value "[file normalize "\$origin_dir/////.Xilinx/Vivado/2022.2/xhub/board_store/xi"	linx_board_stor
150; set_property -name "board_part" -value "digilentinc.com:basyssipart0:1.2" -objects \$obj	
152 : set_property -name "enable_resource_estimation" -value "0" -objects \$obj	
153; set_property -name "enable_vhdl_2008" -value "1" -objects \$obj	
15% set_property "name "ip_cache_permissions" -value "ead write" -oojects \$00] 155 set property -name "ip_output repo" -value "\$\$proj dir/\${ xil proj name }.cache/ip" -objects \$00]	
156 set_property -name 'mem.enable_memory_nap_generation'' -value ''1'' -objects \$obj	
157; set_property -name "platform.board_id" -value "basys3" -objects \$obj 158; set_property -name "revised directory structure" -value "]" -objects \$obj	
159 set_property -name "sim.central_dir" -value "\$proj_dir/\${_xil_proj_name_}.ip_user_files" -objects \$obj	
160; set_property -name "sim.ip.auto_export_scripts" -value "1" -objects \$obj	
162: set_property -name simulator_tanguage -value nixed -objects sobj	
163 set_property -name "target_language" -value "VHDL" -objects \$obj	
165 · set_property -name 'webtalk.xsim_launch_sim' -value ''6'' -objects \$obj 165 ·	
166 # Create 'sources_1' fileset (if not found)	
167; if {[string equal [get_filesets -quiet sources_1] ***]} {	
169 : )	
170	
171 # Set "sources_1" Tileset sources 1	
173 set files (list \	
174: [file normalize "\${origin_dir}//lab22.srcs/sources_1/inports/src/LED_Counter.vhd"] \ 175:11	
176 add_files -norecurse -fileset \$obj \$files	
179 # Set sources_1 illeset lite properties for remote illes 179 set file "Soriain dir/Ab22.srcs/sources l/inports/src/LED Counter.vhd"	
180 set file (file normalize \$file)	
181; set file obj (get files -of objects (get filesets sources 1) [List "*\$file"])	~
	>

#### USEFUL VIVADO TCL COMMNANDS

```
# Create the project
create_project -part <target_device > <project_name > <project_directory >
# Open Existing Project
open project -file <project file >
# Add files to synthesis fileset sources 1
add files -fileset [get filesets sources 1] <files >
# Add files to constraint fileset constrs 1
add_files -fileset [get_filesets constrs_1] <files >
# Add files to simulation fileset sim 1
add files -fileset [get filesets sim 1] <files >
# Launch Run (e.g. synth_1, impl_1)
launch_runs <run> -jobs <no. parallel jobs>
# Wait for run to finish
wait on run <run>
# Reset run
reset_run <run>
# Write Bitstream
launch runs <implementation run> -to step write bitstream -jobs <no. parallel jobs>
```

#### USEFUL TCL DOCUMENTATION

- Tcl 8.6 Manual https://www.tcl.tk/man/tcl8.6/
- Vivado Design Suite Tcl Command Reference Guide (link)
- Most of the commands have a help flag. E.g. from the Vivado Tcl console

```
# Syntax
<command> -help
# Example
open_project -help
```

## LAB 22: ADVANCED VIVADO FLOW

The figures in these slides are taken from:

- Digital Design: Principles and Practices, Fourth Edition, John F. Wakerly, ISBN 0-13- 186389-4. ©2006, Pearson Education, Inc, Upper Saddle River, NJ. All rights reserved
- 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
- adafruit.com
- www.icdesigntips.com
- techdocs.altium.com
- anysilicon.com
- Yngve Hafting 2021, University of Oslo
- www.myomron.com