

REASON: REsearch and training Action for System On chip desigN An IST Project (IST-2000-30193) of the Fifth Framework Program

Tutorial:

A modern HDL-based design flow for FPGA prototyping of ASICs



University of Zielona Góra Institute of Computer Engineering and Electronics



Table of Contents

- Introduction to a new and modern ASIC design flow.
- Comparison of typical and new design flows for ASIC device (advantages and disadvantages).
- Presentation of the new design ASIC flow, based on an example.
- Overview of hardware accelerated simulation technology.
- Benchmark results & hardware accelerated simulation profits.





Tutorial Purposes

- Elimination of ASIC design verification bottlenecks through innovative hardware solutions.
- Presentation of new trends of developing ASIC complex designs.
- Acquaintance of the audience with the recent and modern EDA solutions.
- Presentation and practical application of new technology used to accelerate simulation of huge HDL designs.



<u>PART1</u>

Modern ASIC design flow conception based on hardware accelerated simulation





REASON

The Traditional Design Flow – validation steps



The Traditional Design Flow – incremental prototyping process







The Traditional Design Flow – the property

- Software netlists (post-synthesis, postimplementation) simulation; the total simulation time strongly depends on the computer performance (CPU speed, RAM capacity).
- Long verification period. Each verification of developed design at the post-implementation simulation level requires re-run of the functional and post-synthesis simulation processes.
- Attached to the simulated design ready-made IP-cores slow a whole design verification.
 - This flow is dedicated only for small designs.





REASON

New conception of design flow for ASICs





New conception of design flow for ASICs – incremental prototyping

- After the module is verified it is "pushed" into prototyping board. New blocks may be added into the software simulator as well.
 - Since the simulation time depends only on the design size in the software simulator, it's speeds up system simulation.







New conception of design flow for ASICs – functional verification

- All blocks destined for accelerated simulation have to be synthesized and implemented with Virtex technology.
- If the software and hardware simulation results are the same, the user knows that the synthesis and P&R process does not change the functionality of the used blocks.







New conception of design flow for ASICs – time costs

- Total simulation time is the sum of simulation times for all blocks.
 - The simulation time of "accelerated" blocks is drastically decreased (it is approximately "ZERO" in comparison with software simulation).









New conception of design flow for ASICs – advantages

- New FPGA prototyping design flow provides:
 - hardware accelerated simulation,
 - hardware verification of user designs.
- This flow decreases the design verification time.
- Assures dramatic simulation acceleration.
- Hardware verification confirms the real functionality of design in the hardware part – design is almost ready for ASIC.





New conception of design flow for ASICs – summary

- Hardware Accelerated Simulation can speedup the "re-simulation" time by over 100X.
- The simulation process can speed-up also by acceleration only selected design components (IP CORE simulation acceleration – the partial simulation).
- Faster Performance than C Model Flows.
- Proposed technology capacity: about 15 Million ASIC gates.



<u>PART 2</u>

Hardware accelerated simulation technology





- overview

- There are available many hardware accelerated simulators on the market, like IKOS, QuickTurn, and others.
- It is important (e.g. from academic point of view) to use hardware simulation system that is powerful, easy to share, easy to use, and not very expensive. It is provided by Alatek HES (Hardware Embedded Simulation) technology. Our tutorial is based on the Alatek HES board.
- This solution is very flexible and is useful for various engineer applications.





- hardware embedded simulation

- Design Verification Manager (DVM)
 - Implements Incremental Prototyping
 - Manages block insertion into HES
- HES boards
 - Virtex based: XCV800, XCV2000 and XCV2000 with Daughter Board
 - Apex based: Apex1000
- Interface & Simulators
 - PLI, FLI or VHPI interfaces
 - Model Technology ModelSim
 - Aldec's Active-HDL and Riviera
 - Cadence Verilog XL and NC-sim, Synopsys VCS
 - Workstations Supported
 - PC with Windows/Linux

SUN with UNIX

REASON





- design types of acceleration
- Whole Design Acceleration; Engineers may offload or "drop" the entire design onto the HES accelerator. Speed-up about 10X or higher.
- Partial Design Acceleration; Engineers may offload or "drop" sections of the design onto the HES accelerator, using Incremental Design Prototyping (IDP[™]) technology. Speed-up about 100X or higher.
- Hardware & Software Co-verification; Hardware & Software co-verification is the bottleneck in the SoC designs and usually is very expensive and time consuming. It is possible to speed-up SoC co-verification about 98X or higher.





- the partial design acceleration $\mathrm{IP^{TM}}$





- Incremental Prototyping and/or Co-simulation

- After each VHDL/Verilog/EDIF block is verified, it is 'pushed' into the HES prototyping module
- The design remains 'connected' by HES through the PCI interface.
- Any new added logic blocks will be automatically connected with the VHDL1, EDIF1, and Verilog1 through the signal names
- Co-simulation is performing by PLI, FLI or VHPI interface.
- Debugging can be performed in the user-familiar simulator environment. No need to learn a new simulator.

HES Prototyping Module







- HES technology advantages
- Uses existing HDL Software Simulator through PLI & VHPI interfaces.
- Performance/FAST RTL Simulation (10X or more & partial RTL modules 100X).
- Hardware/software co-simulation.
- Universal Test Development platform.
- IP Core Simulation.
- Mixing VHDL, Verilog, EDIF and hard macros modules.
- Incremental Prototyping[™] methodology calls for development of designs in multiple, logical block systems.
- PCI Card, small form factor.



<u>PART 3</u>

Presentation of new design flow with a project example





HDL Design

- the exercise steps
- Design overview DVB stream filters unit.
- Using the Aldec Active-HDL system for complete user project designing and managing.
- The HES DVM graphical user interface.
- Performing hardware accelerated simulation.
- Benchmark comparison.



The DVB stream filters unit – a design example





REASON

The DVB stream filters unit design

- digital TV receiver overview

The DVB streams filter units are used in the digital TV receivers. Such unit consists of:

- the front-end; this part is responsible for processing an input analog signal,
- the digital part; it is responsible for decoding and approving digital input data.





The DVB stream filters unit design

- overview

- Analysis of the transporting stream.
- Transfer errors detection. Cooperation with initial correction systems, like Philips TDA 8043H (FEC).
- Demultiplexing of the DVB stream and switching correct program data for appropriate audio and video decoders.
- Consolidation of the PSI service tables for feature processing.
- Cooperation with external MPEG-1,2 stream decoders (e.g. SAA7201).



Minimum processing speed is about 72Mb/s.



The DVB stream filters unit design - flow diagram





Using the Aldec Active-HDL system for complete user project designing and managing





Integrated Development Environment

Active-HDL is an integrated development environment dedicated for VHDL, Verilog, EDIF and mixed VHDL- Verilog-EDIF designs.







Workspace

- access to several designs at the same time

- Workspace/Design Explorer facilitates management of workspaces and designs so that user does not have to worry about physical locations of workspace and design files on the computer.
- Each workspace and design is represented in the Workspace/Design Explorer window by a shortcut icon.
- Workspace/Design Explorer allows a user to group the icons in special folders.





Design Browser

- project sources manager
- Design Browser is a tool designed to help user managing projects, sources, resources, workspace.
- The Design Browser window includes three tabs:
 - The Files tab,
 - The Structure tab,
 - The Resources tab.







An example – block diagram

Each of these (yellow) blocks has been developed as an independent system component. Such technique is a property for ASIC design methodology, where, basically, a whole system consists of several IP CORE components.







Block Diagram Editor

- simple hierarchy tool for designing a complex hardware models based on IP Cores
- Any compiled HDL source code, e.g. IP Core, can be placed at the BDE sheet as a symbol.
- Block Diagram Editor is a graphical tool used to create hierarchical block diagrams.
- The editor automatically translates graphically designed diagrams into VHDL or Verilog code as hierarchical structure.
- Any compiled HDL source can be placed on the BDE sheet as a simple symbol ready to be connected to designed system (application for IP Core based systems).





Simulation Results

- The simulation results could be saved as a waveform document for future (post-synthesis, timing and hardware accelerated simulations) comparisons.
- The comparison process compares signal patterns with actual values and marks all detected differences. This can be highly customized through a number of options that specify comparison:
 - range
 - offset
 - tolerance
 - compared signals hierarchy paths
 - maximum differences limit.





DVB system settings

- the functional simulation

- While designing process of DVB project, there appears requirements to use additional external applications (developed in C).
- The Active-HDL environment makes possible to run external applications (executables) by user for various purposes.
- For the DVB design purposes, there has been developed two applications:
 - IP CORE Configurator; customizes and starts simulation process,
 - MemoryAgent; graphical representation of the simulation results.

IP-Core Configurator	×	IP-Core Configurator	
Settings Simulation Filters Run Filter Settings Image: Short Filter 1 Image: Short Filter 1 Image: Short Filter 2 Image: Short Filter 1 Image: Short Filter 3 Image: Short 1 Image: Short Fi	Byte Value Byte Mask 1 00000000 00000000 2 00000000 00000000 3 00000000 00000000 4 00000000 00000000 5	Settings Simulation Type of simulation: Behavioral Synthesis Post-Synthesis Post-Synthesis Post-Implementation Source files from: Exemplar Leonardo FPGA Express Synplify Pro Time: ms Start 	DVB transport stream file E:\MyProjects\hdl\DVB_filters\src\T Browse Description: DVB transport stream PID Description 0000000000000 Program Association Table 0000010110000 Program Map Table 0000010110100 Video 000001110000 Audio Simulation Time: 45 ms





IDE Active-HDL Flow







Flow Configuration

- The flow supports the most known and widely used synthesis and FPGA vendor implementation tools, e.g.:
 - Leonardo Spectrum
 - Synplify Synplicity
 - XST
 - Xilinx Foundation 3.3
 - ISE 5.1
 - Altera Quartus
- The server farm technique makes possible to share and use servers performance over local network.






RTL Simulation

- synthesis parameters

- Both synthesis and implementation processes are fully configurable.
- The configuration settings could be stored in the TCL file for future command line operations.

Synthesis Options	×
Design Files:	
Image: TestBenich_Source Top Level Unit: IMPEG2_decoder Image: Project_Source Image: Top Level Unit: ImpEG2_decoder Image: Behavioral Family : Xilinx-3.3 VIRTEX Image: Behavioral Image: Source Image: Behavioral Family : Xilinx-3.3 VIRTEX	
Speed : -4 Speed : -4 Simulation Output Format: O None O VHDL O Verilog	
Image: Crc_table.vnd Image: Crc_table.vnd Image: Crc_table.vnd Image: Crc_table.as Image: Crc_table.as	
tenerate script → tenerate script → tenerate script → tenerate script → tenerate script → tenerate script	
OK Cancel Help	





RTL Simulation - design complexity

Synthesis Results: Exemplar Leonardo v.2002_1d				
Project Properties		FPGA Family		
	Spartan 2	Virtex	Virtex II	
Logic Blocks	1845	1636	1725	
Function Generators	3690	3271	3449	
Flip-flops	2025	2068	2080	
Input/Output Ports	78	78	78	
Max Frequency	36.0 MHz	40.0 MHz	64.1 MHz	
Suggested Device Part	2S200fg256	V800hq240	2V500fg256	





Summary (software) Simulation - Results

- All simulations have been performed with the same simulation parameters and computer performance:
 - Simulation parameters:
 - Simulation time: 10ms
 - DVB stream file: test_03.dvb
 - Simulation results:
 - functional simulation lasted 10 seconds,
 - post-synthesis simulation takes about 70 minutes,
 - timing simulation has to go on for 90 minutes.





Part 3, Summary

- The software simulator performance abilities pose a bottleneck of huge ASIC designs.
- Acceleration of simulation is expecting and required.
 - Most of them meets set demands, but almost all require long time period of design preparation for accelerated simulation, e.g. synthesis and implementation steps.
- The Incremental Prototyping of hardware embedded simulation technology cuts the time-to-market developing period.



<u>PART 4</u> Hardware accelerated simulation

- the exercise





Agenda of Part 4

- Design Verification Manager overview.
- HES and Active-HDL quick start.
- Design Verification Manager prepare design for hardware acceleration
 - Exercise 1; accelerated simulation of five DVB components
 - Exercise 2; incremental prototyping technology



Simulation verification & Benchmark Results.



- DVM 2.x series provides many features and enhancements that increase speed of design preparation for acceleration with HES and come closer to ASIC design acceleration with FPGA technology.
- The flow of design simulation with HES can be summarized in three steps:
 - Extracting data for DVM.
 - Preparing design for acceleration.
 - Configuring design and HES boards to start co-simulation.



DVM Flow

- Synthesis is performed incrementally.
- User can use up to four HES boards simultaneously.
- After design partitioning and modules selection for acceleration user specifies additional properties as well as internal signals for debugging if required.
- At the end, FPGA implementation is performed and bit-stream files are produced for HES boards configuration.







Hardware Accelerated Simulation

- acceleration environment







HES DVM – general overview

	Active-HDL	Riviera	ModelSim	Verilog XL NC-SIM
Platform	MS Windows	Linux, MS Windows, Solaris	Linux, MS Windows, Solaris	Solaris
HDL	VHDL Verilog, mixed	VHDL Verilog, mixed	VHDL Verilog, mixed	VHDL Verilog
Board Type	V800 ÷ V2000DB	V800 ÷ V2000DB	V800 ÷ V2000DB	V800 ÷ V2000DB
Synopsys FPGA Express/Compiler II	yes		yes	yes
Exemplar Leonardo Spectrum	yes	Not supported on Linux OS	yes	yes
Synplicity Synplify or Certify	yes		yes	yes



HES and Active-HDL – quick start





Compilation of project sources

- To start DVM manager and setup the acceleration environment, it is necessary to compile all project files.
 - Open DVB design,
 - Execute the compile.do macro located in the \$DSN\src\Project_Sourc e\Synthesis folder,
 - This compiles all project and test-bench files.





HES Data Generation (1)

- The DVM requires specific simulation data file, which can be reached by following steps:
 - 1. Set the simulation toplevel for DVB design, the testbench component in this case.





HES Data Generation (2)

- 2. In order to carry out the *hes.dat* generation, the *HES Data Generation* option from the **Design/Settings** menu has to be checked.
- 3. It is necessary to specify path to the HES Library (*libhes_riviera2.dll*), which contains PLI functions for *hes.dat* generation.

Design Settings			<u> </u>
Verilog Libraries General Co	Verilog PLI mpilation SDF	EDIF Libraries	Code Coverage Trace/Debug
General Disable warnings FPGA Booster FES Data Gener	from SDF reader		HESLibrary
VHDL Specific Ignore vital glitches Enable Vital acceleration			
Verilog Specific Disable timing ch Disable timing ch Delay Selection:	ecks eck messages Typ Delay	-	



HES Data Generation (3)

4. Initialize simulation to generate hes.dat.



5. After the initialization is completed, the information about *hes.dat* file generation will appear on the console window. It is placed in the design root directory by default and the path to *hes.dat* is displayed in the Console window.

	• #	Simulation has been initialized
×	□ #	Selected Top-Level: testbench (testbench)
•	□ #	PLI: Loading library 'C:\CAD\Aldec\Active-HDL 5.2\bin/systf.dll'
8	□ #	PLI: Loading library 'C:\CAD\DVM23\bin\libhes_riviera2.dll'
É	□ #	HES Riviera/Active-HDL Generator. Version 2.2
8	□ #	Simulator: Active-HDL
	□ #	Writing : 'E:\MyProjects\hdl\DVB_filters\hes.dat'
	□ #	Preparing hes.dat time: 2 [s]
	>	
		Console /





Design Verification Manager – preparing design for hardware acceleration



DVB Components

- The Incremental Prototyping Technology of hardware embedded simulation accelerates the time-to-market developing period of ASIC designs through partial/total components simulation.
- Each of DVB project components could be developed in a different time and by different designers.
- The simulation has been performed incrementally.





- exercise 1, step A







- exercise 1, step B

- HES Design Verification Manager will guide a designer through all steps required to simulate a design in hardware.
- The resulting bit-stream will be downloaded into FPGA device at the beginning of simulation process.
- The DVM wizard generates a number of files required to perform the appropriate software/hardware co-simulation.





- exercise 1, step C

Start the DVM manager from Active-HDL tools menu.







REASON

DVB Project settings

- exercise 1, step D

Create a new project with the *hes.dat* file generated during simulation initialization.

😹 Design Verification Manager - Noname *				
File Edit View Boards Synthesis Module Port Implement	ntation Tools Help		v av Avlassian s. s.	
121 🗃 📂 🖬 121 📂 🔛 🗠 121 122 123 124 124 124 124 124 124 124 124 124 124	🔹 🖑 🕷 🏶 Ø 10 Ø 1	<u>୍</u> କ ରୋକ ଜୋକ	4 省 🖉 💛 🏠 🕷	
FLOW		D		
📕 📴 Design 🛛 🗱 Debug 🗍 🍘 Memories 🛛 📆 Daught	ter Board (🦉 HES Files	1 Simplementation	1	
💱 Structure 🛛 🎕 Find	Boards Information			×
▼ Design Structure	Board	Instance FPC	A Resources Interf	ace Resources
TestBench: TestBench(testbench)	F 🐺 <u>HES1X800_1</u>	0%	In:0.0	0% Out:0.0%
U1: FEC(fec_body)	HES Modules	0%	In:O	Out:0 Debug:0
U2: MPEG2_system_demultiplexer(mpeg2	Clock Buffers	U%		
U3: User_Interface(user_interface)	Low Skew Lines	U%		
U4: MPEG2_decoder(mpeg2_decoder)				
US. SRAW(sram)				
р	71			
🔷 🔜 👋 Synthesis 🖓	, Implementation			
				_
				-
Ready			TestBench Active-HDL	Leonardo Spectrum

#57 of 82



- exercise 1, step E

Selecting components for hardware accelerated simulation.



#58 of 82





- exercise 1, step F

Generating HES simulation environment.

File Edit View Boards Synthesis M	Madala Bash Zaralasa a Kati		😹 Design Verification Manager - E:/MyProjects/hdl/DVB_filters/DVB.dvm *		
File Edit View Boards Synthesis Module Port Implementation Tools Help					
🖓 🛩 🖬 🎽 🚅 📨 ా 😫 🎦 🍱 🍯 🖧 😓 🚴 🌲 🆓 🐳 🌒 관 🧐 🥸 🌒 관 🌾 🌒 관 🌾 🌒 관 관 🌾 🌒 관 관 🎉 🕷					
FLOW					
🛛 🕞 Design 🗎 🗱 Debug 🗋 🍻 Me	emories 🛛 🚯 Daughter E	oard 🛄 HES Files 🛛 🏶 Implementation			
Options		Files			
HES Files Folder	-Encoding	Generate Files Do not ask for overwrite			
HES_practice1	Encoding for STD_LOGIC type	HES Directory Path: E:/MyProjects/hdl/DVB_filters/src	\$		
HES Architecture Name Arch_HES Configuration Options Do not use configuration Configuration Name: HES_Cfg Levels limit:	'U' => 0 'U' => 0 'U' => 0 'U' => 1 'U' => 0	Generated Files F HES_practice1 G dvb_filters_crc_table_HES.vhd dvb_filters_Long_Filters_Group_HES.vhd dvb_filters_Parser_HES.vhd dvb_filters_PCR_processing_HES.vhd dvb_filters_Short_Filters_Group_HES.vhd mapping.map Simulate_HES.do work_cfg.vhd HES1X800_1 D1_package_package_0.vhd curv_dvb_filters_crc_table_vbd			



Ready



- exercise 1, step G

Performing the synthesis and implementation process.

Design Verification Manager - E:/MyProjects/hdl/DVB_filters/DVB.dvm *	
File Edit View Boards Synthesis Module Port Implementation Tools Help	
\$\$\$ ☞ 🖬 🎦 ☞ 🐨 ♀ ♀ ♀ ♀ ♀ ♀ ♀ ● ◆ ◆ ◆ ● ○ ◇ ◇ ◇ ◇ ◇ ◇ ◇ ◇ ◇ ◇ ◇ ◇ ◇ ◇ ◇ ◇ ◇ ◇	
FLOW	
🕼 Design 🛛 🗱 Debug 🛛 🥔 Memories 🛛 🔂 Daughter Board 🔍 🖳 HES Files 🥻 Implementation 🔪	
FPGA Status F IM HES1X800_1 Stop Implementation IM Run all boards concurrently	
Chip: 1 Place & Route Implementation flow for HES1X800_1 - chip: 1	
Synthesis Translate Map Place and Route	Trace
Done Done Running	?
	F
🗙 🕞 🗙 Console 🕼 Transcript 🔒 Synthesis 🍇 Implementation	
HES1X800_1:1	
Loading device database for application par from file "map.ncd".	
"HES_top" is an NCD, version 2.32, device xcv800, package hq240, speed -6	
C:/CAD/Xilinx.	
	<u> </u>
Implementation in progress //TestBench/U2 Active-HDI //Le	onardo Spectrum

#60 of 82

RFASO



- exercise 1, the partitioning table

Software Simulator	HES board
testbench	/U2/U2:parser
/U1:fec	/U2/U3:crc_table
/U2:mpeg2_system_demultiplexer	/U2/U4:short_filters_group
/U2/U1:filters_fifo	/U2/U6:long_filters_gourp
/U2/U10:av_filetr_module	/U2/U9:pcr_processing
/U2/U11:status_unit	
/U2/U5:gen_crc_table	
/U2/U7:addr_decoder	
/U2/U8:subtitling_filter_module	
/U3:user_interface	
/U4:mpeg_decoder	
/U5:sram	





- exercise 1
- DVB project has been split into two parts:
 - Hardware: contains chosen components for accelerated simulation,
 - Software: behavioral or under development components.

HES (FPGA device)









- exercise 1, start simulation

 To start accelerated simulation, execute generated by DVM the simulation macro file: \$DSN\src\HES_practice1 \simul\Simulate_HES.do







- exercise 1, simulation process
- The console window reports about connection between software simulator and HES board while simulation initialization. This indicates, that selected for hardware accelerated simulation modules have been downloaded into an FPGA device on HES board.
 - The structure tab of Design Browser presents all design components and design hierarchy. Components accelerated by HES are listed with "(arch_hes)" as architecture definition.







- exercise 1, simulation review
- The simulation user interface has not been changed.
- The only difference is the significant acceleration of the simulation process.
- The RTL (software) simulator performs simulation process (computation) for a part (*defined software components*) of the DVB project only.





Simulation results

- end exercise 1

Components with the "arch_hes" architecture name (listed in the simulation structure) are located in the FPGA device, on the HES board.







- exercise 2
- The Incremental Prototyping Technology is based on existing and already implemented design components.
- It is not necessary to perform all synthesis and implementation steps for components computed during "exercise 1".
- The DVM manager only performs synthesis and implementation steps for new selected (delegated for hardware accelerated simulation) DVB design components.
- This significantly increases, in addition, preparation of design for verification process.





- exercise 2, step A







- exercise 2, operating system information
- The HES technology supports many operating systems.
- Next DVM steps (generation of HES files and implementation bitstream) will be performed on SUN workstation with Solaris 8.0 operating system.





REASON

DVB Project settings

- exercise 2, step B

Loading the DVM project (saved during exercise 1).



#70 of 82



- exercise 2, step C

Adding additional design components to the HES board.







- exercise 2, step D

Generating HES simulation environment for exercise 2.

Design Verification Manager – /expor	t/home/staff/gore/projects/MTI/dvb_2/DVB.dvm * 🛛 👘 🗖
<u>File E</u> dit <u>View B</u> oards <u>S</u> ynthesis <u>M</u> odule <u>P</u> ort <u>I</u> mplementation	Iools <u>H</u> elp
🚳 🚅 🖵 🗃 🚄 🔛 🔛 🔕 🔯 🐴 🐎 🔈 🔌 🔌 🍬	🏽 🌒 🗿 🚳 🕷 🏷 🎯 🦗 🖉 🦉 🛤
FLOW	
🕞 Design 🛛 🗱 Debug 🛛 🥔 Memories 🛛 🏠 Daughter Board 🛛 🖳 HES	6 Files Files Files
Options [File	3
HES Files Folder Encoding for HES_practice2 Encoding for STD_LOGIC type HES Architecture Name 'U' => 0 Arch_HES 'X' => 0	Generate Files Do not ask for overwrite S Directory Path: /export/home/staff/gore/projects/MTI/dvb_2
Configuration Options $0' \Rightarrow 0$ $0' \Rightarrow 0$ $0' \Rightarrow 1$ $0' \Rightarrow 1$ $1' \Rightarrow 1$ $1' \Rightarrow 1$ $1' \Rightarrow 1$ HES_Cfg $1' \Rightarrow 1$ $1' \Rightarrow 0$ $1' \Rightarrow 1$ $1' \Rightarrow 1$ $1' \Rightarrow 1$ $1' \Rightarrow 0$ $1' \Rightarrow 1$	<pre>imul work_Parser_HES.vhd work_crc_table_HES.vhd work_Short_Filters_Group_HES.vhd work_Long_Filters_Group_HES.vhd work_PCR_processing_HES.vhd work_Filters_FIFO_HES.vhd work_Gen_crc_table_HES.vhd work_ADDR_Decoder_HES.vhd work_ADDR_Decoder_HES.vhd work_AV_Filter_Module_HES.vhd</pre>
Console Transcript Synthesis By Implementation • GENERATOR: Writing configuration file(s) • GENERATOR: Writing simulation script	
Senerator: Files generated successfully.	




REASON

DVB Project settings

- exercise 2, step E

Performing the synthesis and implementation process.

Design Verification Manager – /export/home/staff/gore/projects/MTI/dvb_2/DVB.dvm *	• [
<u>F</u> ile <u>E</u> dit <u>V</u> iew <u>B</u> oards <u>S</u> ynthesis <u>M</u> odule <u>P</u> ort <u>I</u> mplementation <u>T</u> ools	<u>H</u> elp
\$	
FLOW	
🛛 📭 Design 🛛 💥 Debug 🛛 🥔 Memories 🛛 🏠 Daughter Board 🔍 📑 HES Files 🦓 Implementation	
FPGA Status F-GP HES2X1000MB_1 Implementation	
Chip: 1 Place & Route	
Image: sector of the sector	
Synthesis Translate Map Flace and houte Trace Done Done Done Running ?	
× Sonsole 🤬 Transcript 📴 Synthesis 🎉 Implementation	
IMPLEMENTATION: HES2X1000ME_1:1: Phase 3.23 (Checksum: 99527a) REAL time: 17 secs IMPLEMENTATION: HES2X1000ME_1:1: IMPLEMENTATION: HES2X1000ME_1:1: Phase 4.2	7
Z INPLEMENTATION: HESZX1000MB_1:1: Phase 4.2 ≫	Z
nglementation in progress I/TestBench/u2 ModelSim Leonardo Si	pectru

#73 of 82



DVB Project settings

- exercise 2, the partitioning table

Software Simulator	Hardware Accelerated
testbench	/U2/U2:parser
/U1:fec	/U2/U3:crc_table
/U2:mpeg2_system_demultiplexer	/U2/U1:filters_fifo
/U3:user_interface	/U2/U4:short_filters_group
/U4:mpeg_decoder	/U2/U6:long_filters_gourp
/U5:sram	/U2/U9:pcr_processing
	/U2/U10:av_filetr_module
	/U2/U5:gen_crc_table
	/U2/U8:subtitling_filter_module /U2/U11:status_unit
	/U2/U7:addr_decoder





DVB Project settings

- exercise 2
- DVB project has been split into two parts:
 - Hardware: contains all components resides at the U2 level of design hierarchy,
 - Software: behavioral or under development components.

HES (FPGA device)







DVB Project settings

- end exercise 2

Components with the "arch_hes" architecture name (listed in the simulation structure) are located in the FPGA device, on the HES board.

Design Browser ×	X 🖻 🖬 🖌	Si 🔓 🔍	ム 🛚 🔍 🔍 🔍 🔩 🗤 🗤 熊 🕵 🖭 🔟 👬 🥵 🥕 ≫
Ches_cfg	Name	Value	ı 100 ı 200 ı 300 ı 400 ı 500 ı 600 ı 700 ı 800 ı 1000 us
-+ hes_cfg : TestBench (testbench)		FF	
⊕- € U1 : fec (fec_body)		0021	03FF X0017 X0021
E = E U2 : mpeg2_system_demultiplexer (mpeg2_sy		14D	(000 X 14D
		CO	00 XC0
- U11 : status_unit (arch_hes)		00	80 00
⊕-↓ U2 : parser (arch_hes)		FF	
Image: International and the second secon		FF	
- US : gen_crc_table (arch_hes)	AT NET5249	0	
⊕-↓ U6 : long_filters_group (arch_hes)	AT NET6559	0	
⊕ ⊕ U8 : subtitling_filter_module (arch_hes)	# NET6953	0	
U9 : pcr_processing (arch_hes)	# NET6998	1	
⊕-↓ U3 : user_interface (user_interface)	M NET77	0	
	AT NET85	1	
- E std.standard	AT NET89	0	
□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □	- Interrupt	1	
-P ieee.std_logic_unsigned	PWM_out	1	
- B std.textio			
Files Structure Caresources	🔝 waveform e		



Results Verification & & Benchmarks





Results Verification (1) - graphical comparison

- There are two variants of waveform comparison operation:
 - compare all signals in two waveform files,
 - compare signals selected in source waveform file.
- Any of these two operations can be initiated from the Waveform Viewer/Editor toolbar or from the Waveform menu.
- The signals are compared in accordance with their names.





Results Verification (2) - graphical comparison

Comparison of exercise 1 and exercise 2 simulation results.

Name	Value	Sti	т. • 100 • т. • 200 • т. • 300 • т. • 400 • т. • 500 • т. • 600 • т. • 700 • т. • 800 • т. • 900 • т. 🖓	
± # BUS5579			(00 X C0	
± # BUS5583			(uu <u>X</u> 00	
± # BUS6175			<u></u>	
표 🕶 BUS97			©	
# NET6276				
# NET6559				
# NET6953		Ì		
# NET6998				
™ NET77				
4 NET85				
₩ NET89				
P Interrupt				
PWM_out		ļ		
			\ \ \ \ \	
🚪 design flow 🔏	practive1_si.			
<pre> # ELBREAD: Elaboration time 0.6 [s]. # Comparing waveforms: "practive1_simulation.awf" - "practive2_simulation.awf" #</pre>				





Results Verification (3) - graphical comparison

Comparison of behavioral and exercise 1 simulation results.

Name	Value	1 100 1 2	0 • • • 300 • • • 400 • •	. 500 600	700 · · · 800 · 963,563264 u
► PKTBAD	0				_
► PKTBCLK	0				
🗉 🖻 PKTDAT	C6	00		() ()	
► PKTDATV	1				
► PKTSYNC	0				
► RST	0				
	0				
⊞ 🗣 F_ADDR	FFFF	FFFF			
. ♥ F_WR_DATA	E1	00	<u> </u>		
🗄 🗝 AVDAT	DD	00	$ \bigcirc \bigcirc$	<u> </u>	
 VSTROBE 	0				
 ASTROBE 	0				
 interrupt 	1				
<pre> * practice1_s * * * * * * * * * * * * * * * * * *</pre>					
Console	Console /				





Benchmark Results

Simulation Type	Simulation Level	Results	
So: Sim	Functional	10 sec	
ftwa ulat	Post-Synthesis	4200 sec (1h 10min)	
Ire	Post-Implementation	5400 sec (1h 30 min)	
Hard Accel Simu	Exercise 1 design configuration	69 sec (60x speed up vs. post-synthesis)	
ware erated lation	Exercise 2 design configuration	28 sec (150x speed up vs. post-synthesis)	



Thank You

