A Systematic Method for Hardware Software Codesign using Vivado HLS

Pranitha.K, Kavya.G

Abstract: ***This paper aims to provide increased productivity for designing, integrating and implementing systems using xilinx vivado design suite. It can accelerate design implementation with place and route tools that analytically optimize for multiple and concurrent design metrics such as timing, congestion, total wire length, utilization and power; it also provides design analysis capabilities at each design stage. An overview of vivado design suite is illustrated with configuration, implementation, detailed implementation, summary, settings along with component name. Here the component DDS compiler has been chosen and the waveform repository, design settings are added to it. Improved productivity results are indicated through simulation, synthesis, implementation, bitstream generation.***

Keywords: Vivado IDE, DDS, Implementation.

I. INTRODUCTION

The Vivado® Design Suite is designed to improve productivity. This tool suite is architected to increase the overall productivity for designing, integrating, and implementing systems using the Xilinx® 7 series, Zynq®-7000 All Programmable (AP) SoC, and UltraScale™ devices. Xilinx devices are now much larger and come with a variety of new technology, including stacked silicon interconnect (SSI) technology, up to 28 gigabyte (GB) high speed I/O interfaces, hardened microprocessors and peripherals, analog mixed signal, and more. These larger and more complex devices create multidimensional design challenges, when handled incorrectly, that can prevent the achievement of faster time-to-market and increased productivity. With the Vivado Design Suite, you can accelerate design implementation with place and route tools that analytically optimize for multiple and concurrent design metrics, such as timing, congestion, total wire length, utilization and power. The Vivado Design Suite provides you with design analysis capabilities at each design stage. This allows for design and tool setting modifications earlier in the design processes where they have less overall schedule impact, thus reducing design iterations and accelerating productivity. The Vivado Design Suite replaces the existing Xilinx ISE® Design Suite of tools. It replaces all of the ISE Design Suite point tools, such as Project Navigator, Xilinx Synthesis Technology (XST), implementation, CORE Generator™ tool, Timing Constraints Editor, ISE Simulator (ISim), ChipScope™ Analyzer, Xilinx Power Analyzer, FPGA Editor, PlanAhead™ design tool, and SmartXplorer. All of these capabilities are now built directly into the Vivado Design Suite and leverage a shared scalable data model.

Built on the shared scalable data model of the Vivado Design Suite, the entire design process can be executed in memory without having to write or translate any intermediate file formats, which accelerates runtimes, debug, and implementation while reducing memory requirements. All of the Vivado Design Suite tools are written with a native tool command language (Tcl) interface. All of the commands and options available in the Vivado Integrated Design Environment (IDE), which is the graphical user interface (GUI) for the Vivado Design Suite, are accessible through Tcl.

The Vivado Design Suite also provides powerful access to the design data for reporting and configuration as well as the tool commands and options. It interacts with the Vivado Design Suite using, GUI-based commands in the Vivado IDE, Tcl commands entered in the Tcl Console in IDE or saved to a Tcl script file that is run either in the Vivado IDE or in the Vivado Design Suite Tcl shell. A mix of GUI-based and Tcl commands, Tcl script can contain Tcl commands covering the entire design synthesis and implementation flow, including all necessary reports generated for design analysis at any point in the design flow.

II. LITERATURE REVIEW

FPGA Design of Real Time MDFD System Using High Level Synthesis (Chuliang Wei et.al, 2019) in this paper, a novel technique (MDFD) multi input depth from defocus has been introduced. This technique has been implemented on FPGA device, to improve the latency and interval of the proposed technique using vivado HLS (high level synthesis).Initially it is designed through C/C++ code and it is converted to verilog or VHDL code. Output images are obtained in the form of pixel size of 640x480. A continuous image like about 22 images can be processed at a time at 20MHZ frequency along with the depth of accuracy of 93.2% i.e represents the state of art of MDFD technique. Frame – based programming, stream - based processing for medical image processing applications (Joost Hoozemans et.al., 2019) this paper explains about the technique to set up the image and video processing pipelines which has been established from framed based programming and stream based processing especially for FPGA.

Revised Manuscript Received on November 15, 2019

K.Pranitha, Research Scholar at Anna University, Chennai, India.
G.Kavya, Professor in Electronics and Communication Engineering at S.A.Engineering College, Chennai.

This method adopts soft core VLIW preprocessors with 'C' compiler to estimate the difficulties while using software before entering into high level synthesis design flow. This technique takes high time consumption to design or introduce new product, it act as a bridge to connect software developers and hardware designers to provide efficient circuit simulation and synthesis time.

Optimized Memory Allocation and Power Minimization for FPGA-based Image Processing (Paulo Garcia et.al., 2019) in this paper, proposed methods adopted are optical flow and mean shift tracking. Nowadays in FPGA implementation for image processing, power and memory consumption is the main challenge for building or designing high level architectures. Proposed methods will reduce the utilization, power consumption and will produce efficient FPGA based image processing. It brings out memory efficient architectures from hardware description language and high level synthesis designs, 60% utilization efficiency, 70% dynamic power consumption and total power reduced upto 25 to 30% without affecting the performance.

A New FPGA based Architecture to Improve Performance of Deflectometry Image Processing Algorithm (Faraz Bhatti et.al, 2017) in this paper, proposed method, Deflectometry image processing algorithm has been introduced to investigate effects on reflecting surfaces. Proposed architecture has been designed using vivado HLS. Hardware implementation provides flexibility for reconfigurability, parallelization and pipelining which improves latency, execution time and gain. Proposed architecture along with simulation results are explained. It shows the correlation between productivity and resource utilization.

Generating FPGA based Image Processing Accelerators with Hipacc (Oliver Reichie et.al., 2017) this paper aims to provide high comfort zone among hardware target architectures. Hipacc is the proposed framework which is the combination of DSL (Domain specific language) and source to source compiler for image processing applications. Here the proposed framework generates software C based implementation using DSL for targeted FPGA's; it includes individual memory requirement and provides effective distinguished conversions. As a final result it provides comparison of efficient implementation cuda software (CPU) and hardware accelerators using DSL source code.

Very High Level Synthesis for Image Processing Applications (YanjingBI et.al., 2016) this paper aims to produce effective model for FPGA designs in Matlab environment by using very high level synthesis method. The proposed VHLS method contains behavior specification, source to source compiler, control and data path extraction, RTL generation, after performing these entire steps target RTL will be generated. The above method is designed with composite design flow for evaluating it in real life applications. This technique provides efficient RTL productivity and in conversion of C to RTL and it decreases the complication of design flow.

III. PROPOSED WORK

Procedure for implementing hardware software codesign in the DDS compiler:

Double click on DDS COMPILER 6.0, on left side of the window, tab with IP symbol enabled show input and output ports are displayed and information like resource estimates, AXI4- stream port structure are indicated. On right on side, starting with component name and tabs like configuration, implementation, detailed summary are displayed. In configuration tab, set configuration option as "SINCOSLUT" only. In system requirements, set system clock frequency as 100 MHZ, number of channels as '1', mode of operation as "Standard", parameter section as "hardware parameters", noise shaping as "none". In hardware parameters, set phase width as "16", range will be "3 to 16" and output width as "16", range will be "3 to 26". Click on implementation; make output section as "SINE", amplitude mode as "Full range". Implementation options like memory type, optimization goal, Dsp480SE are indicated as default values. In detailed implementation tab, AXI channel options as "packet framing". In TUSER options, input as "user field", data output as "user field", phase output "not required", user field width as "1", Output form as "sign and magnitude", synchronization mode as "on vector".In latency configuration, set mode as "auto" and latency value as "4". Select control signals as "ARESET n (active low) it must be asserted for a minimum of two cycles. Then the summary will provide the selected complete details. After setting all these mode of configurations, Result analysis is done. Screenshots for implementing hardware software codesign in DDS compiler are shown below:

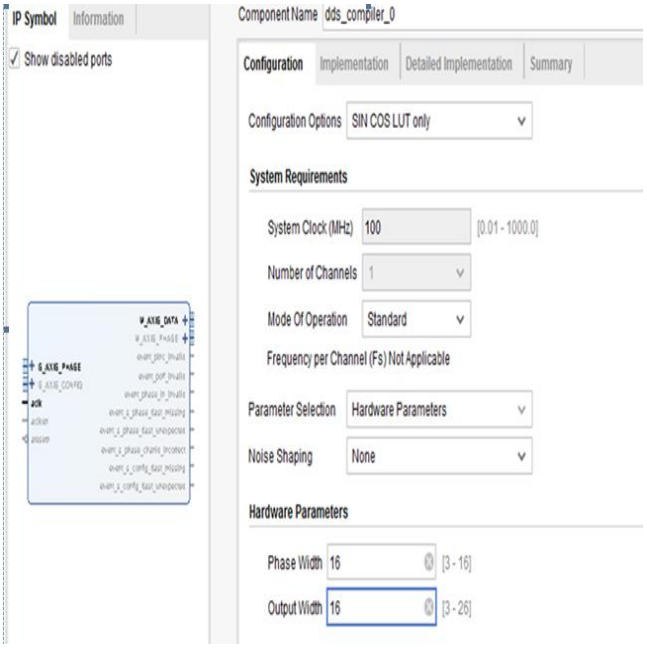


Figure 1 Configuration of hardware software codesign in DDS compiler

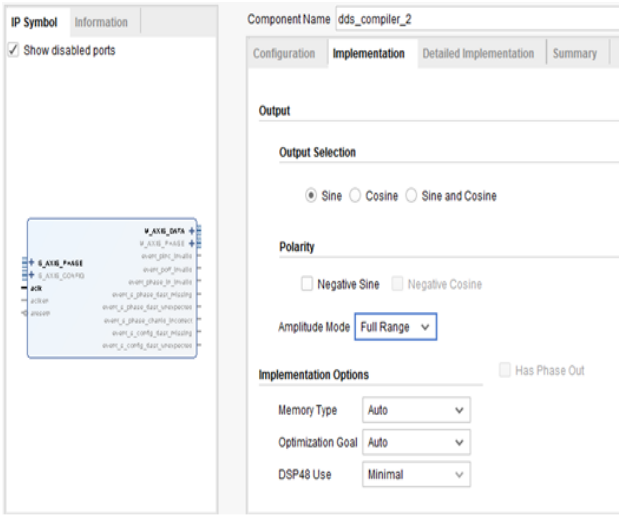


Figure 2 Implementation of hardware software codesign in DDS compiler

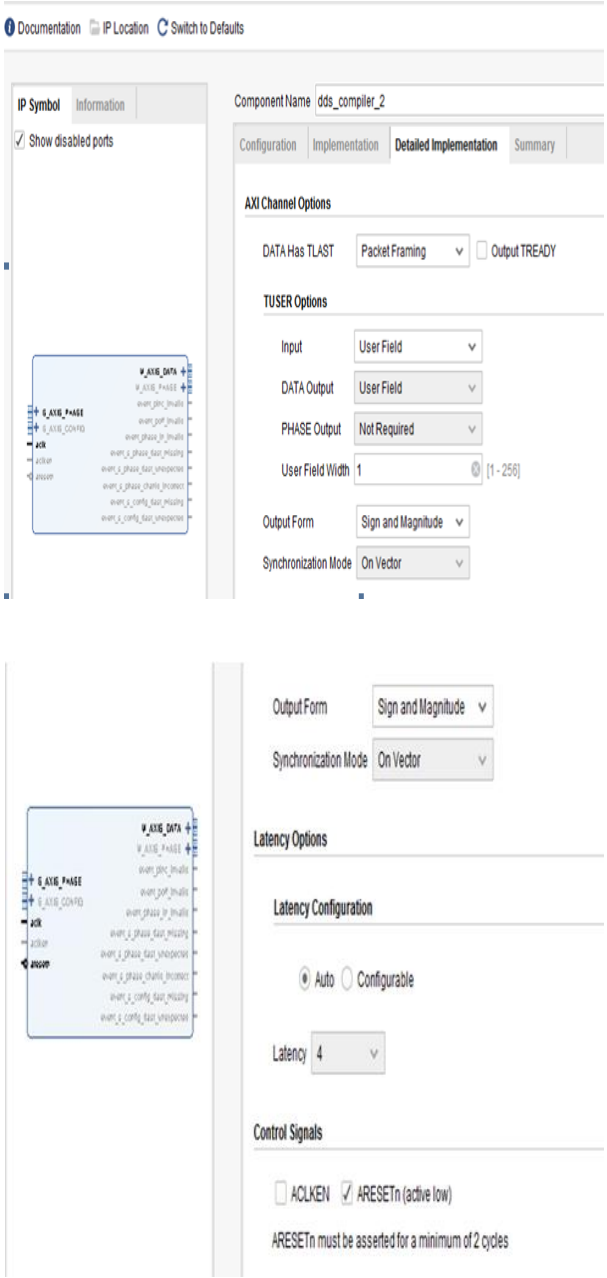


Figure 3 Detailed Implementation of hardware software codesign in DDS compiler

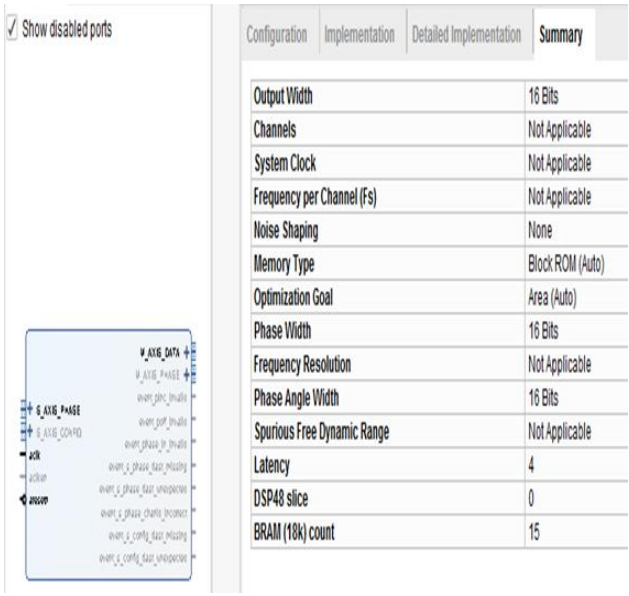
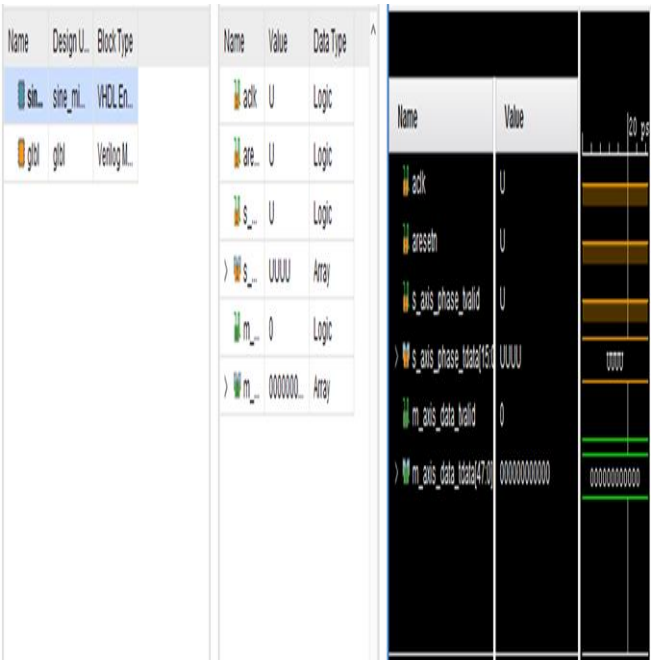


Figure 4 Summary of hardware software codesign in DDS compiler

Procedure for analyzing performance results based on simulation, synthesis, implementation and bit stream generation:

Click on simulation → run behavior simulation. Simulation scripts are generated and also it will generate simulation wave form for the given input SINE wave signal according to the phase width and output width values. Output simulation waveforms from 0 to 1000 ns are displayed below.

Behavioral Simulation results:



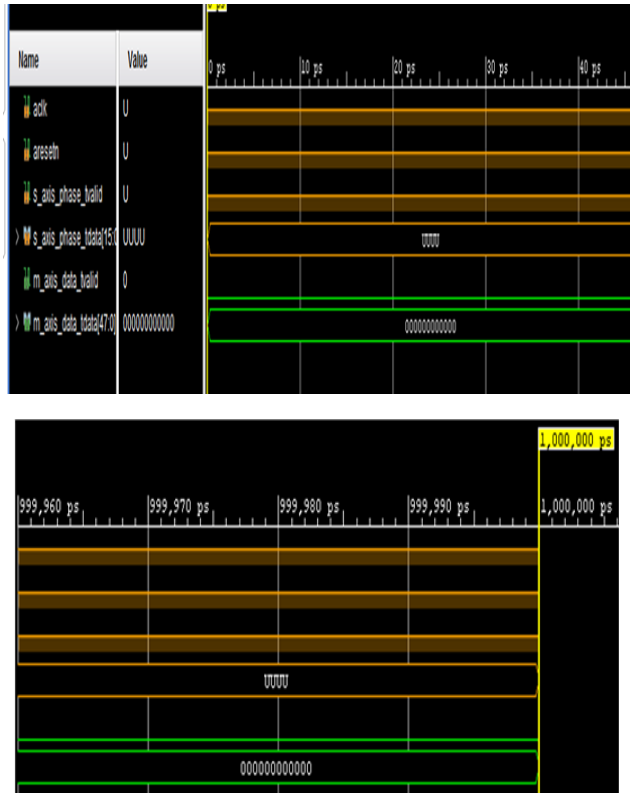


Figure 5 Simulation results using DDS compiler for implementation

Right click on synthesis →run synthesis → launch synthesis run and click OK in dialog box for starting synthesis. It will take time to perform, synthesis scripts are generated and the designs are produced accordingly which is displayed below.

Synthesis design result:

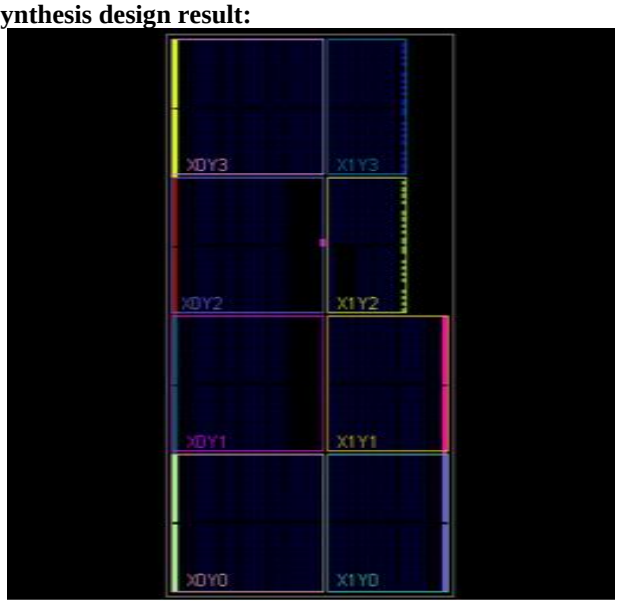


Figure 6 Screenshot for Synthesis design

Right click on implementation → run implementation → launch implementation run and click OK in dialog box for starting the implementation, it will take time to perform, here route_design will be completed after implementation process which is shown below.

Implementation design result:

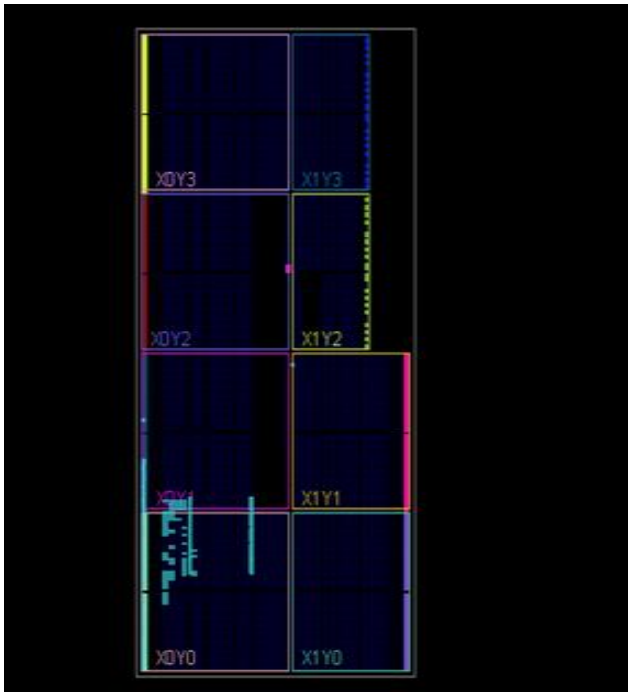


Figure 7 Screenshot for implementation design

Bit stream configuration and generation has been done which is displayed below. Bit stream file has been generated by using the command, write_bitstream –bin_file*.bitstream_test.

```
Loading data files...
Loading site data...
Loading route data...
Processing options...
Creating bitmap...
Creating bitstream...
Writing bitstream ./bitstrream_test.bit...
Writing bitstream ./bitstrream_test.bin...
INFO: [Vivado 12-1842] Bitgen Completed Successfully.
INFO: [Project 1-118] WebTalk data collection is enabled (User setting is ON. Install Setting is ON.).
INFO: [Common 17-186] 'D:/Xilinx/design_files/project_xsim1/usage_statistics_webtalk.xml' has been successfully sent to Xilinx
INFO: [Common 17-83] Releasing license: Implementation
10 Infos, 3 Warnings, 0 Critical Warnings and 0 Errors encountered.
write_bitstream completed successfully
write_bitstream: Time (s): cpu = 00:00:34 ; elapsed = 00:00:37 . Memory (MB): peak = 2193.660 ; gain = 298.855
bitstrream_test
```

Figure 8 Screenshot for bitstream generation

Schematic view for the elaborated design using the FPGA kit (xc7k70tfbg484-2 (active)) which is selected while creating a project.

Elaborated design:

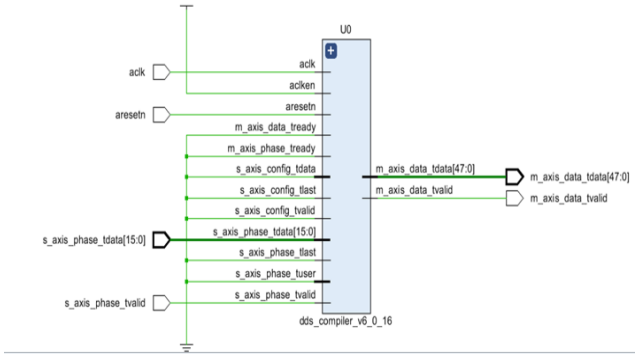


Figure 8 Screenshot for Elaborated design

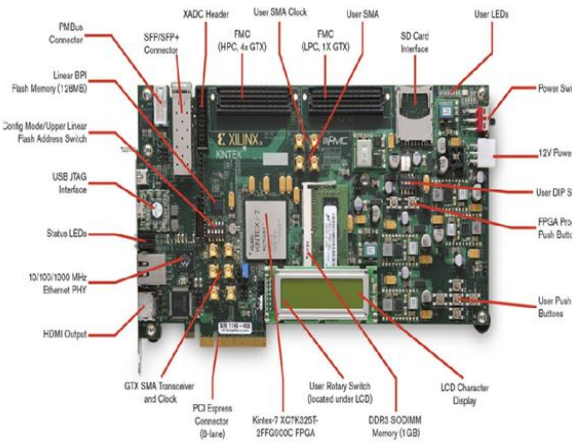
Project summary report has been generated along the details with settings, board part, synthesis, implementation, DRC violations, timing, utilization, power screenshots are displayed below.

Project name:	project_xsim1
Project location:	D:/Xilinx/design_files/project_xsim1
Product family:	Kintex-7
Project part:	Kintex-7 KC705 Evaluation Platform (xc7k325tffg900-2)
Top module name:	sine_mid
Target language:	Verilog
Simulator language:	Mixed

Board Part

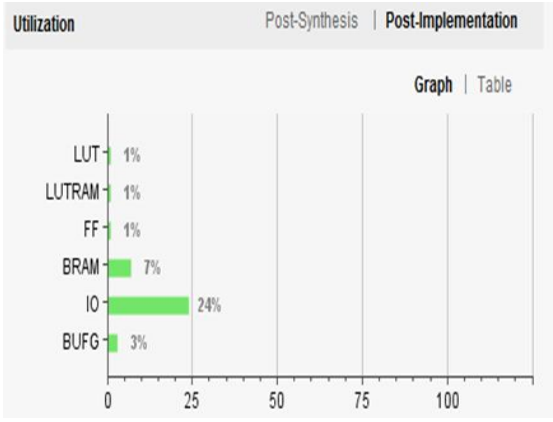
Display name:	Kintex-7 KC705 Evaluation Platform
Board part name:	xilinx.com:kc705:part0:1.5
Connectors:	No connections
Repository path:	D:/Xilinx/Vivado/2018.2/data/boards/board_files
URL:	www.xilinx.com/kc705
Board overview:	Kintex-7 KC705 Evaluation Platform

Changes



Synthesis		Implementation	
Status:	✓ Complete	Status:	✓ Complete
Messages:	112 warnings	Messages:	8 warnings
Active run:	synth_4	Active run:	impl_4
Part:	xc7k70tfbg484-2	Part:	xc7k70tfbg484-2
Strategy:	New3_strategy	Strategy:	Vivado Implementation Defaults
Report Strategy:	Vivado Synthesis Default Reports	Report Strategy:	Vivado Implementation Default Reports
		Incremental compile:	None

DRC Violations		Timing	
Summary:	2 critical warnings	Worst Negative Slack (WNS):	NA
	1 warning	Total Negative Slack (TNS):	NA
Implemented DRC Report		Number of Failing Endpoints:	NA
		Total Number of Endpoints:	NA



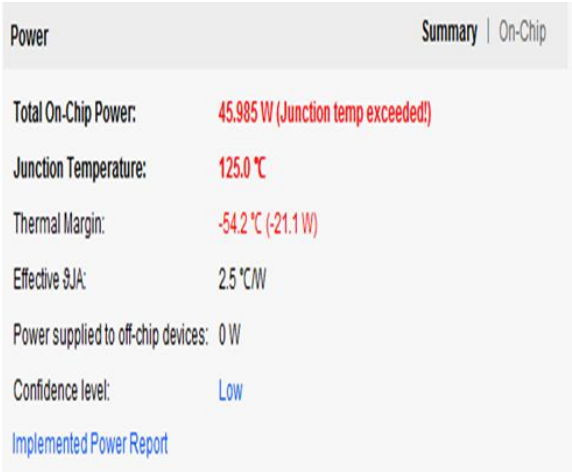


Figure 9 Screenshot for Project Summary Report

IV. CONCLUSION

In this paper, experiment shows that hardware implementation using software tool VIVADO HLS 2018.2 version has been done. Results for simulation, synthesis design, implementation design, schematic view, bitstream configuration and generation are illustrated. Project summary report along with board part is indicated. RTL design (or) RTL implementation has an advantage of both time consuming and error prone. Creating RTL project using VIVADO HLS, it improves the design in better way, decreases latency, increases the throughput, lower the area and device resource utilization.

REFERENCES

1. Marcin kowalczyk, Dominika przewlocka, Tomasz kryjak, 2018, Real time implementation of contextual image processing operations for 4k video stream in zynq ultrascale + MPSOC, 978-1-5386-8237-1/18,IEEE.
2. Amir.Hajirassoulina, Andrew J.Tabenner, Martyn P.Nash, Poul M.F.Nielsen,2018, Suitability of recent hardware accelerators (DSPs, FPGAs, and GPUs) for computer vision and image processing algorithms, Signal processing: image communication 68(2018)101-119[Elsevier].
3. Pranitha.K, Dr.G.Kavya, 2018, Data compression with high peak signal to noise ratio using Bisectional Cylindrical wavelet transform for a satellite image, International journal of Engineering and Technology(UAE) volume 7, No (4.6) (2018).
4. Ghislain takam Tchendjou, Emmanuel Simeu, Rshdee Alhakim, 2017, Fuzzy logic based objective image quality assessment with FPGA implementation, Journal of system architecture, doi: 10.1016/j.sysarc.2017.12.002.
5. Bhaumik vaidya, Mustafa surti, Parthvagha siya, Jay Bordiya, Jenish Jain, 2017, Hardware acceleration of image processing algorithms using vivado high level synthesis tool, International conference on intelligent computing and control systems (ICICCS 2017).
6. Ajay Rupani, Pawan Whig, Gajendra sujediya, Piyush vyas, 2017, A robust technique for image processing based on interfacing of raspberry-PI and FPGS using IOT, International conference on computer, communications and Electronics (comptelix).
7. Pranitha.K, Dr.G.Kavya, Literature Survey Of Image Compression/Decompression Techniques For Telehealth applications in Proceedings IEEE International Conference on Photonics and High Speed Optical Networks ICPHON 2018 pp. 63-68.
8. R. Liu, M. Liu, D. Li, Y. Zhang, and Y. Zheng, “Fpga architecture for real-time ultra-high definition glasses-free 3d system,” in Computer Vision, J. Yang, Q. Hu, M.-M. Cheng, L. Wang, Q. Liu, X. Bai, and D. Meng, Eds. Singapore: Springer Singapore, 2017, pp. 380–392.
9. A.cortes, I.velez, A.Irizar, 2016, High level synthesis using vivado HLS for zynq SOC : image processing case studies, 978-1-5090-4565-5/16, IEEE.
10. Z. Shi, “ Rapid Prototyping of an FPGA-Based Video Processing System,” PhD thesis, Virginia Tech, 2016.
11. M. Nieto, A. Cort´es, O. Otaegui, J. Arr´ospide, and L. Salgado, ”Real-time lane tracking using rao–blackwellized particle filter. Journal of Real-Time Image Processing”, vol. 1, no. 1, pp. 179-191, DOI :10.1007/s11554–012–0315–0, 2016.
12. Pranitha.K, Dr.G.Kavya, Literature Survey Of Image Compression/Decompression Techniques For space and Telehealth applications, oxidation communications, book 2 volume 42(2019), pp 151 – 159.
13. Xilinx. High-Level Synthesis. http://www.xilinx.com/ support/documentation/sw_manuals/xilinx2015_4/ ug902-vivadohigh-level-synthesis.pdf, November 2015.
14. H. M. Abdelgawad, M. Safar, and A. M. Wahba, “ High level synthesis of canny edge detection algorithm on zynq platform“, Int. J. Comput. Electr. Autom. Control Inf. Eng, 9(1):148–152, 2015.)
15. Jinalkumar. D, Joshi et Al ”Implemening a Cloud based Xilinx ISE FPGA Design Platform for Integrated Remote Labs ”, Proc. IEEE ICACCI 2015, pp-533-537.
16. Shivank. D, Zalak. D, Joshi et. Al “Reconfigurable Image Processor using an FPGA-Raspberry pi Interface”, IJCA Proc. ICCT 2015, pp 11-15.
17. M H Lohithaswa, “Canny edge detection algorithm on fpga,” IOSR Journal of Electronics and Communication Engineering (IOSRJECE), 2015.
18. G. Chaple, and R.D. Daruwala, ”Design of Sobel operator based image edge detection algorithm on FPGA”, 2014 International Conference on Communications and Signal Processing (ICCSP), 788 – 792, DOI: 10.1109/ICCSP.2014.6949951, 2014.

AUTHORS PROFILE

K.Pranitha, completed B.E in Electrical and Electronics Engineering (2008) at KSR college of Engineering, Tiruchengode. M.E in VLSI (2012) at Kongu Engineering College, Perundurai. Pursuing PhD in Information and Communication Engineering as a Full Time Research Scholar at Anna University, Chennai, India. Her works have been published in reputed journals, international conferences. Her research interests include VLSI, Image Processing, and Software Testing. **Mail ID:** pranithakarthik@gmail.com

G.Kavya received her PhD degree in Electronics Engineering from Sathyabama University, Chennai, India in the year 2015. She has a great flair for teaching and research and has a total experience of fifteen years in teaching Engineering students. Currently she is working as a Professor in Electronics and Communication Engineering at S.A.Engineering College, Chennai. Her professional interests include VLSI, Wireless Communication, Embedded Systems and Telemedicine. She is a member of many professional societies such as IEEE, IETE ISTE and IAENG. She has published nearly sixteen papers in reputed journals, conference proceedings and magazines. She is the Co-Investigator of a research project funded by All India Council for Technical Education (AICTE), Government of India. Mail ID**:** drgkavya@saec.ac.in