

# A Detailed Illustration of VLSI Block Design Implementation Process Using VIVADO HLS and Arty Kit

K. Pranitha<sup>1,\*</sup>, G. Kavya<sup>1</sup>, M. Arun Kumar<sup>2</sup>

<sup>1</sup>Department of ECE, S.A Engineering College, Anna University, Chennai -77, India

<sup>2</sup>Department of ECE, GITAM University, Bangalore – 561203, India

*Received January 24, 2020; Revised May 6, 2020; Accepted May 13, 2020*

Copyright ©2020 by authors, all rights reserved. Authors agree that this article remains permanently open access under the terms of the Creative Commons Attribution License 4.0 International License

**Abstract** This paper aims to provide consolidated procedure to design and implement VLSI block in Arty A7 kit using VIVADO HLS. Sine wave signal with appropriate pulse width is implemented and default hardware part is selected in VIVADO HLS tool and verilog code has been written in VIVADO IDE for the implementation of Arty A7 kit. Using high level synthesis, an engineer has the contingency to add libraries for the corresponding project and further proceeds with the achievement at superior level of entrancement. The procedure of implementing sinusoidal signal and the performance of implementation in Arty A7 kit using vivado HLS is illustrated along with simulation, synthesis, implementation results and project summary report. Timings, Design rule check (DRC) violations, utilization, total on-chip power are charted in project summary report. The proposed research work is based on FPGA implementation based image compression/decompression using VIVADO HLS. Proposed work is widely used in medical application especially in telehealth where patient records are transferred from remote area to the hospital to make consultation in easier manner. Likewise FPGA implementation based image compression/ decompression is used in fields like remote sensing, oceanography, earth observation etc. to predict the future weather conditions and it helps to take precautionary steps in order to detect the disaster before it occurs. Initial proposed research work has been done based on FPGA implementation of sinusoidal signal using VIVADO HLS. Future research work, FPGA implementation based image compression/decompression will be carried out using VIVADO HLS. The main advantage of Vivado HLS tool is to obtain high speed performance and low power consumption while implementing in hardware.

**Keywords** VIVADO HLS, Flow Navigator, IP Catalog, DRC, VIVADO IDE, TCL

---

## 1. Introduction

VIVADO IDE (Integrated Developed Environment) contributes perspective graphics user interface (GUI) along extraordinary features. Restraint practices will be carried out during entire design process. Timing analysis can be done after power estimations, synthesis, and placement and routing. Modifications are done for the designs accordingly in real time, instead of going for re-implementation. The VIVADO Design Suite with an emphasis on the different project types, using the tool through the GUI and TCL, with a project and without. VIVADO IDE includes, Register Transfer Level design languages used are VHDL, Verilog and System Verilog, IP catalog, Performing Behavioral, functional, timing simulation using vivado simulator, Synthesis design, Optimized Implementation design, Serial input and output ports, Logic analyzer for debugging, Power and Timing analysis, SDC based xilinx design constraints, Floor planning in higher level, Detailed Placement and Routing, Bitstream configuration and Generation. VIVADO adopts an approach of viewing design arrangement in memory. Viewing an arrangement leads to create a design netlist for the appropriate design flow, it allots design constraints to the particular design arrangement and after that dumping the particular design to the targeted equipment. This feature contributes capability of viewing and considering the designs at each phase and it can be done with various implementation methods with retuned timing constraints. Hardware functions are synthesized into

hardware in the Programmable Logic (PL) by the VIVADO HLS compiler. This compiler automatically translates C/C++ code into an FPGA hardware implementation, and as with all compilers, using compiler defaults. There are two flows for optimizing the hardware functions. Top-down flow: In this flow, program decomposition into hardware functions proceeds top down within the SDSoC environment, letting the system compiler create pipelines of functions that automatically operate in dataflow mode. The microarchitecture for each hardware function is optimized using VIVADO HLS. Bottom-up flow: In this flow, the hardware functions are optimized in isolation from the system using the VIVADO HLS compiler provided in the VIVADO Design suite. The hardware functions are analyzed, optimizations directives can be applied to create an implementation other than the default, and the resulting optimized hardware functions are then incorporated into the SDSoC environment. To perform optimizations using VIVADO HLS, it consists of directives which specify a methodology how optimizations are applied in a deterministic and efficient manner. Bottom-up flow is often used in organizations where the software and hardware are optimized by different teams and can be used by software programmers to take advantage for hardware implementations. Both flows are supported and results in the same high-performance system. Xilinx sees the choice as a workflow decision made by individual teams and organizations and provides no recommendation on which flow to use.

## 2. Literature Review

Marcin kowalczyk et al [1] [12] [15] [17] explains about hardware implementation preferred for contingent based image pre-processing sections for a 3840x2160 at 60fps video stream in a Zynq UltraScale and MPSOC is explained. Successive processes like simple averaging, Gaussian filter, Edge detection using SOBEL and CANNY methods, median filter and morphological erosion and dilation are undergone. Video stream is considered in the pattern of 2 and 4 pixels per clock. Logic methods are characterized in VHDL code using high level synthesis tool, VIVADO HLS. Elaborated segments backing real time processing of 4k at 60fps video stream. The above concept clearly states that employment of computations with lower clock frequency acquires clear development on energy utilization against increment in the count of utilized logic resources. Bhaumik vaidya et al [5] [3] [14] [16] describes about effective implementation results for discrete image processing algorithms using VIVADO HLS. FPGA kits like virtex 7, virtex 6, Spartan 6 are utilized to give comparative implementation results. Parameters considered are speed, latency and resource utilization. Basic image processing algorithms like evaluation of

histogram, histogram equalization, averaging filter and laplacian filter are selected to produce complete kit expedition for using VIVADO HLS. This process starts from dumping high level C/C++/System C Code in FPGA kits mentioned above using VIVADO HLS is elucidated or described including implementation results for different image processing algorithms. Ajay Rupani et al [4] [7] [13] discusses about image processing has been done by coordinating raspberry-PI and FPGA using IOT. IOT means Internet of Things, which widen the connectedness of substantial equipment and usual gadgets. IOT plays a vital role in day to day life. Image processing followed by image filtering also done by using above mention interfacing. The mode of process has been carried out by using ZEDBOARD ZYNQ 7000 FPGA and raspberry-PI. Image filtering process like Gaussian filter, sharpening filter and average filter, edge detection and gray scale operations are also done and the results are visualized using VGA monitor. Usage of chip is indicated. A.cortes et al [12] [9] [11] explains about image processing algorithms are constructed in Zynq SOC using VIVADO HLS and are matched with mandated architectures. In VIVADO HLS, an engineer has a scope to add library files to open CV, an open CV is extensively used by the software designers. Here is the process is explained in view of area resources with library files and without library files. This case study is about data binning, a step row filter and sobel filter. The above algorithms are very common in image processing filed so they are adopted for above process. In VIVADO Hls tool, excluding of library files, there is a reduction in time for whole process. On the other hand, including of the library files it increases the FPGS resources i.e. area. Ghislain Takam Tchendjou et al [6] [2] [10] evaluate the functions of machine learning techniques for the technique, image quality assessment. Here the proposed methods are divided into two classifications (linear discriminant analysis and k-nearest neighbors) and four non-linear regressions accesses (artificial neural network, non-linear polynomial regression, decision tree and fuzzy logic). Establishment and the vitality of the architecture of calculated by applying (monte-carlo cross validation), with inconstantly 1000 verification sets. Simulation results determine the fuzzy logic method which has largest reliable approach and better optic opinion. Implemented methods produces ultimate fuzzy logic modeling employing Gaussian and derived membership functions. The proposed implementation is carried out on Kintex 7 FPGA kit with Xilinx VIVADO HLS tool.

## 3. Proposed Work

The Proposed research work focuses on describing about the performance results like simulation, synthesis, implementation and generating project summary report and verilog implementation of Arty A7 kit using xilinx

VIVADO HLS tool. Work starts by creating a RTL project, Here you will be able to add sources, create block designs in IP integrator, generate IP, run RTL analysis, synthesis, implementation, design planning and analysis. Then add sources and constrains, set the target language as Verilog and simulator language as "mixed". Then choose default part or board for your project, this can be changed later. After adding all those things, a new project summary will be displayed then click Finish, a new project will be created. After creating a new project, project settings have to be done. In VIVADO IDE, click on flow navigator, project manager with wide variety of options like settings, IP catalog, IP integrator, simulation, RTL analysis, synthesis, implementation, program and debug are displayed. Steps to be performed for project settings, Click on settings under project manager. Here options with project settings and tool settings are displayed. In tool settings, choices like text editor, web talk, color, strategies, window behavior are listed these options are default options there won't be any changes made here. In project settings, tabs like general, simulation, elaboration, synthesis, implementation, bit stream, IP are listed. In general tab, specify values for various settings used throughout the design flow. These settings are applied to the current project. Settings like project device, target language as "Verilog", default library file, top module name. Language options like loop count are displayed default. In simulation tab, specifies various settings like target simulator as "VIVADO simulator", simulator language as "Mixed", simulation set as SIM1 and simulation top module name. Then in compilation and elaboration check the library files, in simulation give run time as 1000 ns, in netlist set the process as "Slow" and in advanced tab, enable incremental compilation it preserves simulation files during successive simulation runs. In elaboration tab, specifies various settings associated to elaboration like LINK IP model options as "netlist model" and constrains options as "Load constrains". In synthesis, it specifies various settings like default constraint sets as CONSTRS1. Strategy will display the VIVADO synthesis default reports. In synthesis description, default stated as higher

performance designs, resource sharing is turned off, the global fan out guide is set to a lower number, FSM extraction forced to one-hot, LUT combining is disabled, equivalent registers are preserved, SRL are inferred with the larger threshold. In implementation, settings like constrains and report options are indicated. Default options like design initialization, opt design, power opt design, place design, post-place power opt design, post-place phy opt design, route design, post route phy op design are enabled. In bit stream tab, settings are related to writing bit stream, here (-bin\_file\*) is enabled. In IP tab, enabled pre compiled IP simulation libraries, automatically generate simulation scripts for IP, generate log file. IP location where output products and customization are stored. IP cache set us "local". After finishing all these settings, click on Apply and OK. Complete project settings are done to the newly created project. Now move on to give the input signal as "sine wave" signal. Click on IP catalog under flow navigator, a separate window with cores and interfaces are opened in that type "DDS COMPILER" selects the core as wave form synthesis → DDS compiler → AXI4-stream → production → version 6.0.

#### **IMPLEMENTATION OF ARTY A7 KIT USING VIVADO TOOL:**

**In VIVADO IDE,** Click on add sources and add or create design sources then click on next and create file with file name as verilog file (.v file), then click on finish now the IDE environment has been opened for hardware implementation. Verilog code has been written for implementation with input signal as 'CLOCK' and output signal as 'LED' then the register count as [24:0] are adopted. By incrementing the register count value LED will blink accordingly. Run simulation, synthesis, and implementation. After implementation, bitstream generation will be completed (.bit file) has been generated accordingly. Now click on "Open hardware manager" then "open target" it will interface the Arty A7 kit and VIVADO tool along with the connection of local host. Then the implementation result can be viewed by blinking of "LED" light.

## 4. Results

### Simulation result

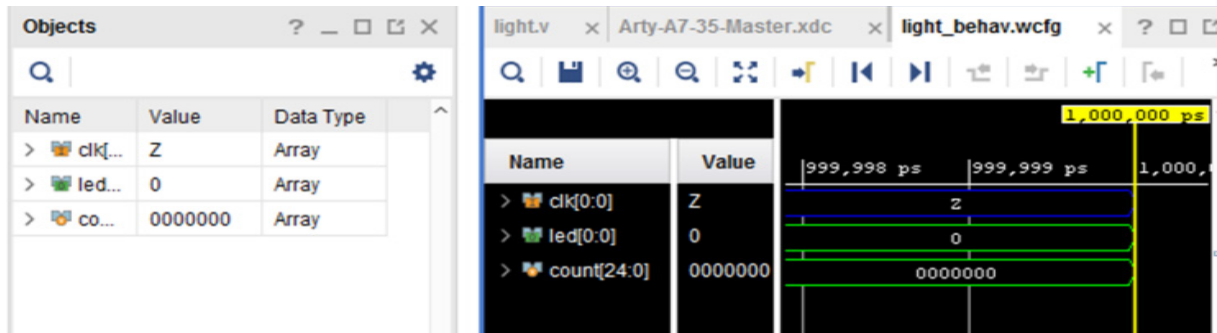


Figure 1. Simulation result using VIVADO for Arty A7 kit

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

### Synthesis design

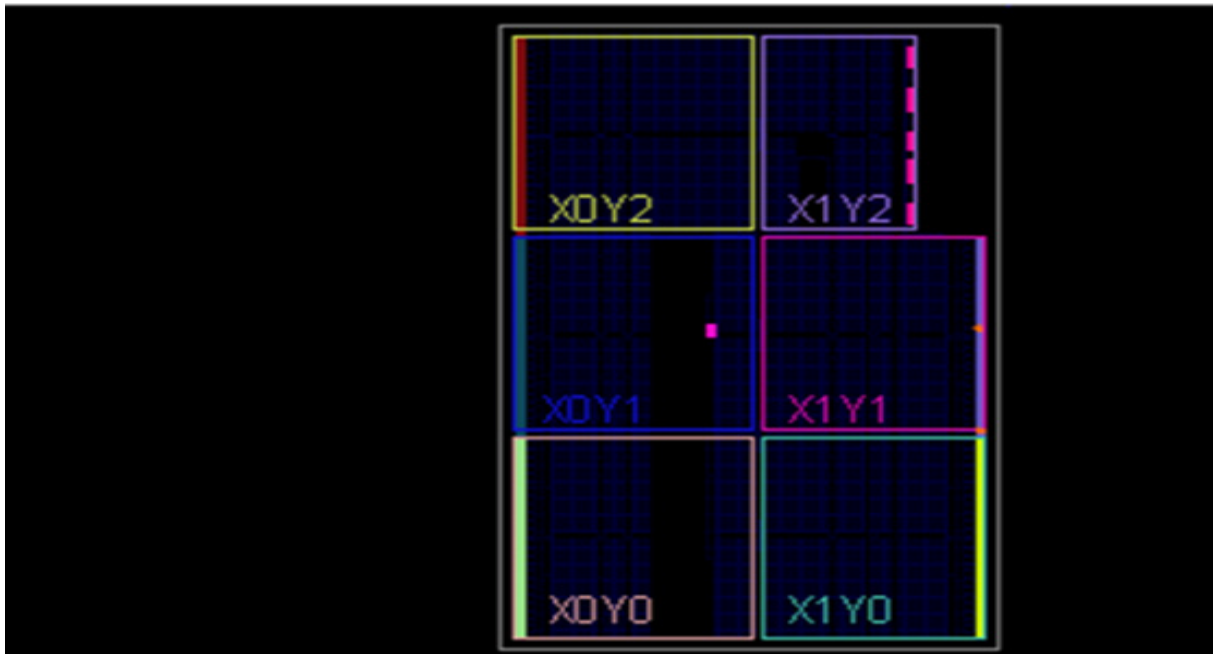
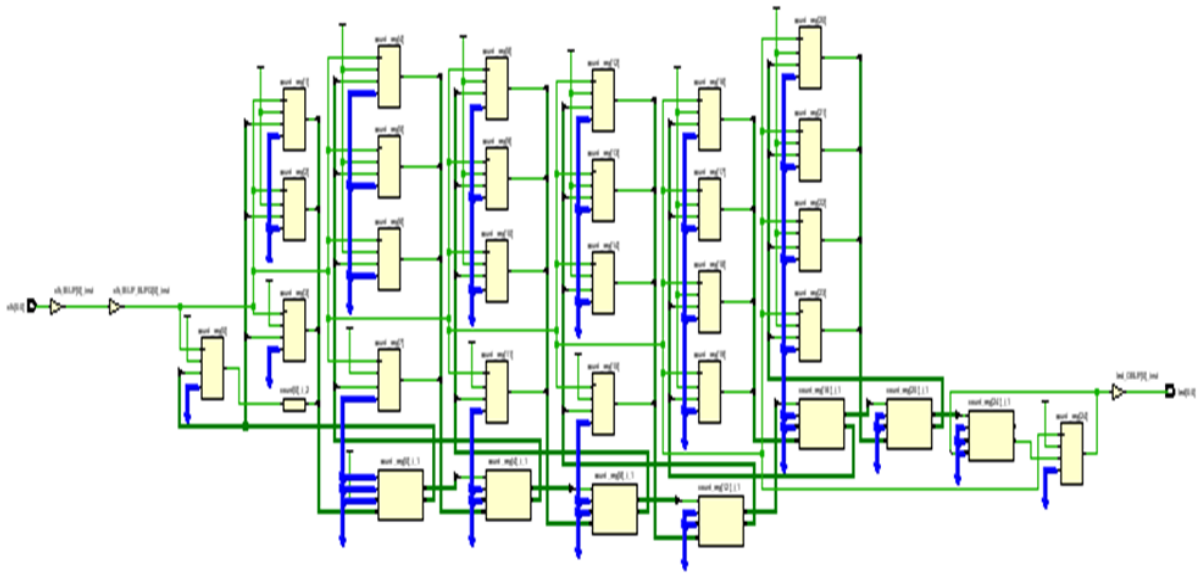


Figure 2. Synthesis design using VIVADO for Arty A7 kit

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 above.

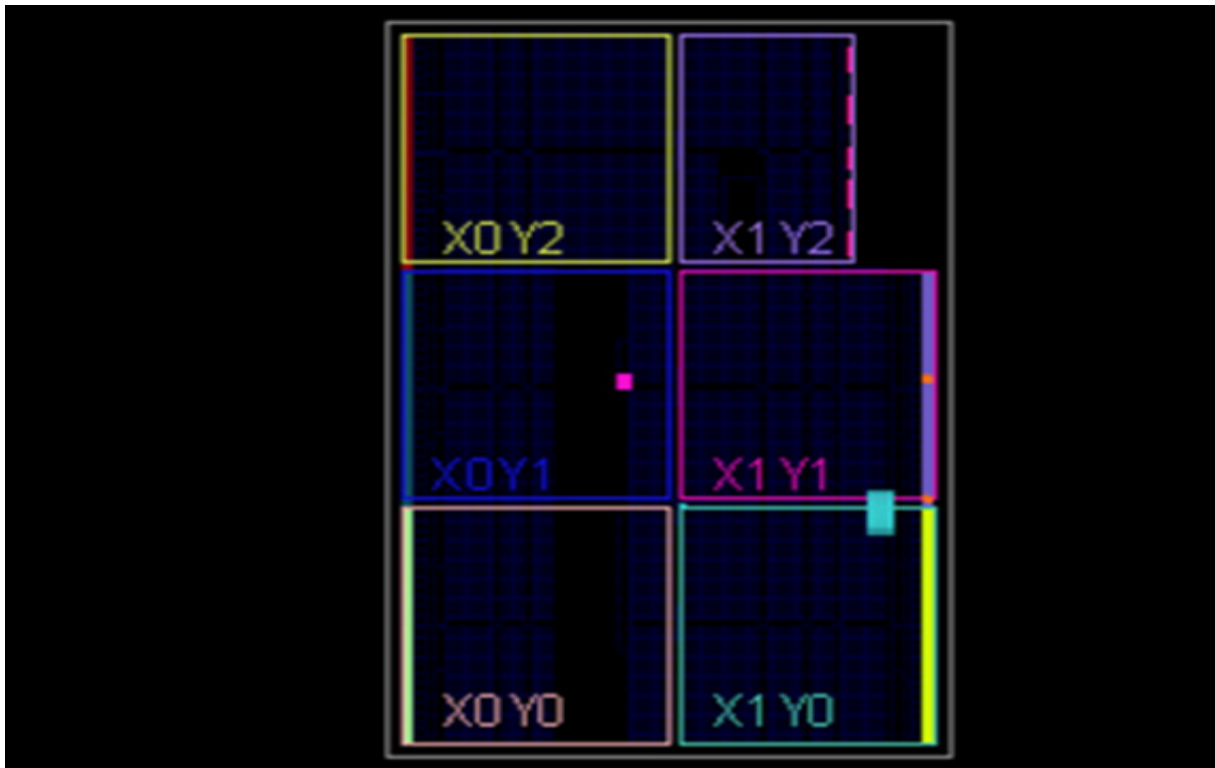
**Schematic view**



**Figure 3.** Schematic view using VIVADO for Arty A7 kit

Schematic view → Click on open schematic view and click OK in dialog box for starting the implementation, it will take time to perform, schematic view of the design will be displayed as shown above.

**Implementation design**



**Figure 4.** Implementation design using VIVADO for Arty A7 kit

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 above.

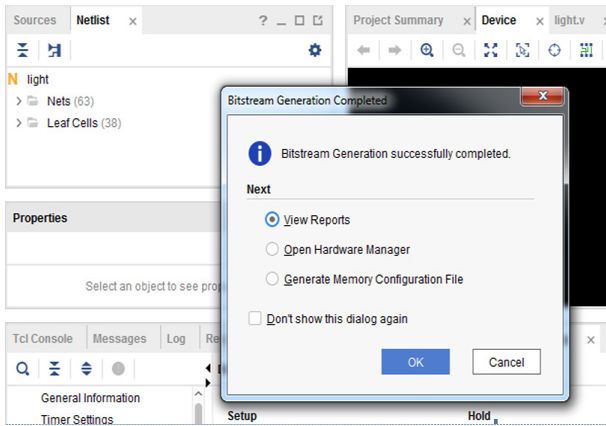
**Design Detail**



**Figure 5.** Elaborated design using vivado for Arty A7 kit

Elaborated design → Click on open elaborated design and click OK in dialog box for starting the implementation, it will take time to perform and elaborated design will be displayed as shown above.

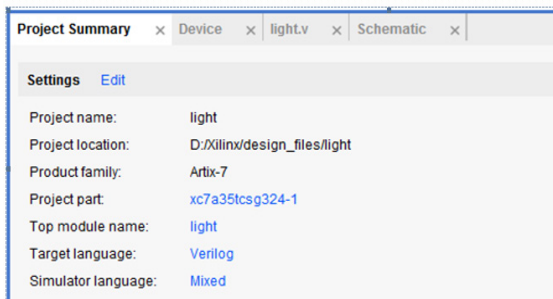
**Generation of Bitstream file**



**Figure 6.** Generation of bitstream file using vivado for Arty A7 kit

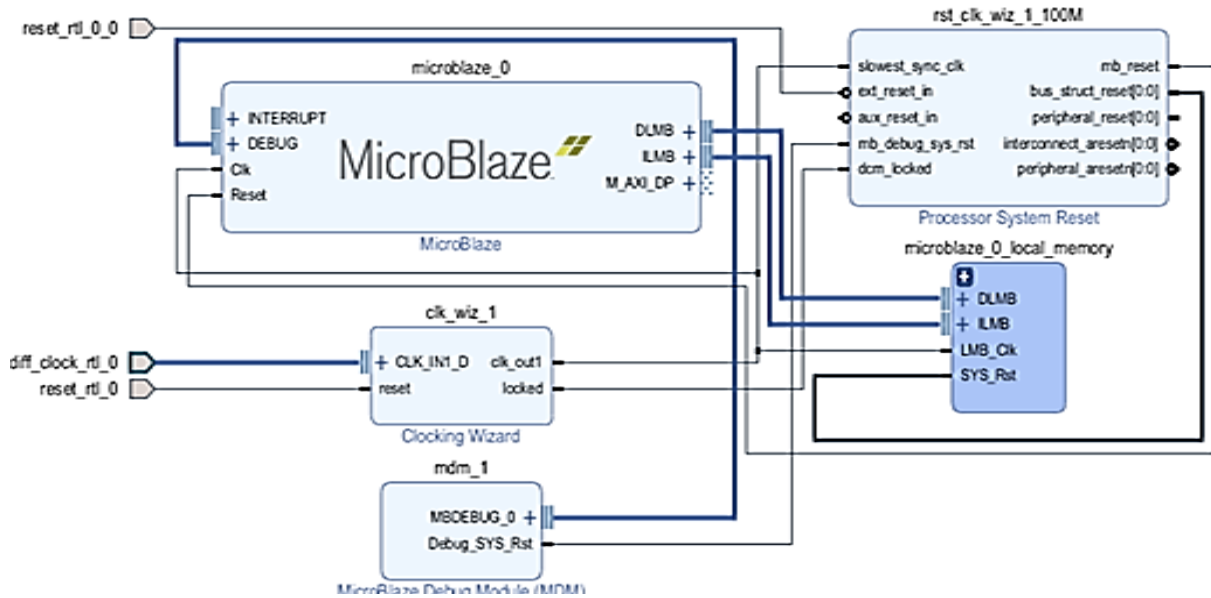
**Project summary report**

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



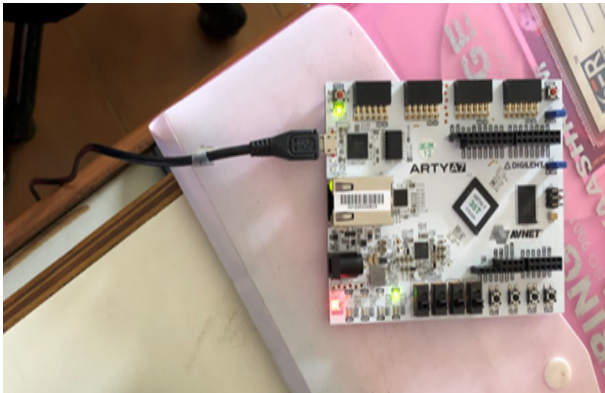
**Figure 7.** Project summary report for implementation of Arty A7 kit

**Block design**



**Figure 8.** Block design for implementation of Arty A7 kit

**Implementation of Arty A7**



**Figure 9.** Implementation of Arty A7 kit

**5. Conclusions**

The above procedure shows that design and implementation using VIVADO HLS has been done in Arty A7 kit. Results for simulation, synthesis design, implementation design, schematic view, and bitstream generation, block design, snapshot for implementation Arty A7 kit 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. The main advantage of using VIVADO HLS, improves the design in better way, decreases device resource utilization and latency with increased throughput.

**REFERENCES**

- [1] Marcin kowalczyk, Dominika przewlocka, Tomasz kryjak, Real time implementation of contextual image processing operations for 4k video stream in zynq ultrascale + MPSOC, 2018, 978-1-5386-8237-1/18, IEEE.
- [2] Amir. Hajirassoulina, Andrew J. Tabenner, Martyn P. Nash, Poul M.F. Nielsen, Suitability of recent hardware accelerators (DSPs, FPGAs, and GPUs) for computer vision and image processing algorithms, Signal processing: image communication, 2018, 68(2018)101-119[Elsevier].
- [3] Pranitha. K, Dr. G. Kavya, 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), 2018, volume 7, No (4.6).
- [4] Ghislain takam Tchendjou, Emmanuel Simeu, Rshdee Alhakim, Fuzzy logic based objective image quality assessment with FPGA implementation, Journal of system architecture, 2017, doi: 10.1016/j.sysarc.2017.12.002.
- [5] Bhaumik vaidya, Mustafa surti, Parthvaghya siya, Jay Bordiya, Jenish Jain, 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, A robust technique for image processing based on interfacing of raspberry-PI and FPGS using IOT, International conference on computer, communications and Electronics (comptelix 2017).
- [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] Pranitha. K, Dr. G. Kavya, A Systematic Method for Hardware Software Codesign using Vivado HLS, *International Journal of Recent Technology and Engineering (IJRTE)*, Volume-8 Issue-4, November 2019, PP 467-472.
- [10] A. cortes, I. velez, A. Irizar, High level synthesis using vivado HLS for zynq SOC: image processing case studies, 978-1-5090-4565-5/16, 2016, IEEE.
- [11] Z. Shi, "Rapid Prototyping of an FPGA-Based Video Processing System," PhD thesis, Virginia Tech, 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] M. Nieto, A. Cortés, O. Otaegui, J. Arrospide, and L. Salgado, Real-time lane tracking using rao–blackwellized particle filter. *Journal of Real-Time Image Processing*, vol. 1, no. 1, 2016, pp. 179-191, DOI :10.1007/s11554–012–0315–0.
- [14] H. M. Abdelgawad, M. Safar, and A. M. Wahba, High level synthesis of canny edge detection algorithm on zynq platform, *International Journal of Computer, Electrical, Automation, Control and Information Engineering*, 2015, vol 9(1) pp 148–152.
- [15] Jinalkumar. D, Joshi et al., Implementing a Cloud based Xilinx ISE FPGA Design Platform for Integrated Remote Labs, in *Proceedings of IEEE ICACCI 2015*, pp-533-537.
- [16] Shivank. D, Zalak. D, Joshi et. Al, Reconfigurable Image Processor using an FPGA-Raspberry pi Interface, *International Journal of Computer Applications Proceedings ICCT 2015*, pp 11-15.
- [17] Xilinx. High-Level Synthesis. [http://www.xilinx.com/support/documentation/sw\\_manuals/xilinx2015\\_4/ug902-vivadohigh-level-synthesis.pdf](http://www.xilinx.com/support/documentation/sw_manuals/xilinx2015_4/ug902-vivadohigh-level-synthesis.pdf), November 2015.