

**BUILDING CUSTOM IMAGE FOR ZEDBOARD USING PETALINUX: CASE  
STUDY**

Yash Naik<sup>1</sup>, Rikin Nayak<sup>2</sup>, Arpita Shah<sup>3</sup>

<sup>1</sup>Student, U & P.U. Patel Department Of Computer Engineering, Chandubhai S. Patel Institute of Technology (CSPIT), Charotar University of Science and Technology (CHARUSAT)

<sup>2</sup>Assistant Professor, V.T. Patel Department of E & C, Chandubhai S. Patel Institute of Technology(CSPIT), Charotar University of Science and Technology (CHARUSAT)

<sup>3</sup>Assistant Professor, U & P.U. Patel Department Of Computer Engineering, Chandubhai S. Patel Institute of Technology (CSPIT), Charotar University of Science and Technology (CHARUSAT)

**Abstract**—This paper is focuses on Building a Custom Image of Linux Operating System(OS). The custom Image of Linux OS is Developed using the Petalinux. “Petalinux is the Embedded Linux System Development Kit” This OS will be developed on Xilinx’s Zynq-7000 All Programmable System on chip which contains the dual core ARM processor. With the use of SD card, OS will be Boot on the Zedboard. The advantage of design customized image for Zedboard, which is consists of reconfiguration system (FPGA) and software programming, to solve complex problem at lower cost, compare to the graphics processing unit(GPU) and General purpose computer. This paper gives information about steps of building the custom image of Linux Operating System(OS).

**Keywords**— Petalinux, FPGA, System on Chip,

**I.INTRODUCTION**

The general purpose processors consume high power and does not give high performance that is why hardware accelerators are required. The hardware accelerator is used to perform some function faster and effective compare to perform on software which is running on general purpose processors CPU.

Different hardware accelerator architectures are shown in the figure 1. Each of these has its own advantage and disadvantage but the third architecture is the faster architecture because the communication between CPU and accelerator being sort Which reduce the length, capacitance of the bus which makes them easier to obtain the high throughput. In the normal computer use the first architecture for implementation and the Zynq Soc use the third architecture for implementation [1].

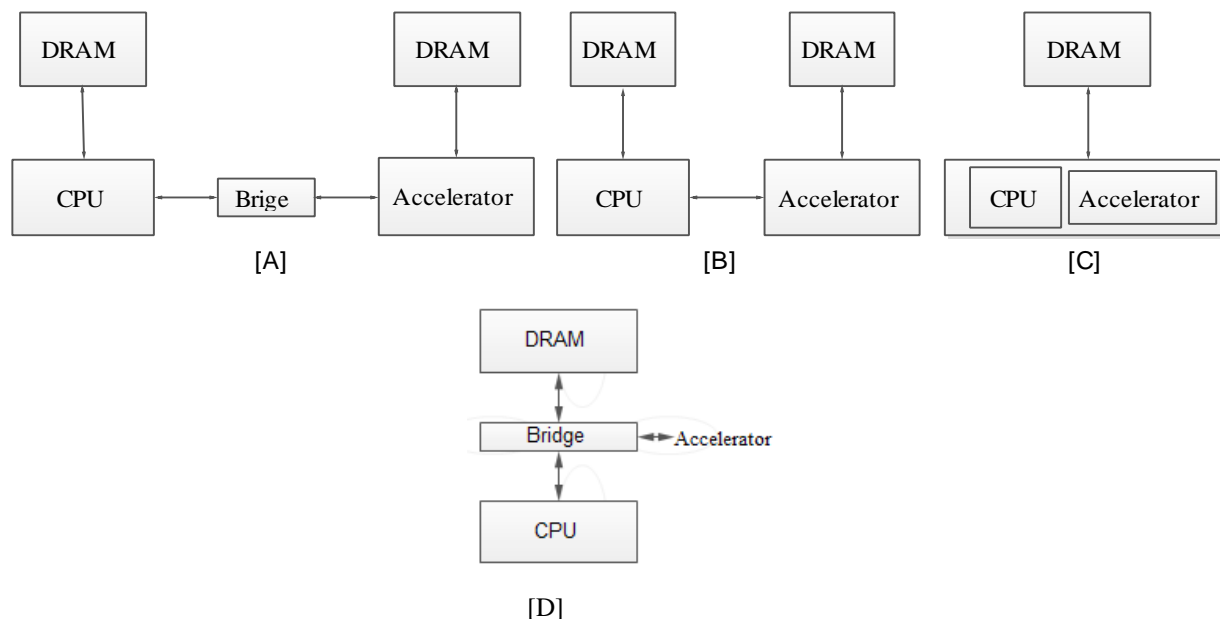


Figure 1 Hardware accelerator architecture[1]

The hardware for the project used is Zedboard development board. Zedboard is the product of Xilinx where a processor and programmable logic are on the same board and it contains the ARM Dual-core processor connected with the programmable gates [2]. The Board Contains the combination of general-purpose processor and programmable logic on one board which allows hardware acceleration with the high data-throughput and memory controls [1]. Custom OS is used for booting the OS on the board. We can build the custom Linux OS for the Zedboard with the use of petalinux. It involves creating a kernel image with the required libraries and packets for supporting hardware device and software application for the board.

## **II.PROCESS OF BUILDING CUSTOM COMPONENTS USING TOOLS**

### **A. Zynq SOC and Zedboard**

Zynq SOC are processor that offer integration of processor and FPGA architecture into a single board. Combination of two technologies provide the higher integrity, lower power consumption, small board size and higher bandwidth communication between the FPGA and processor [8].

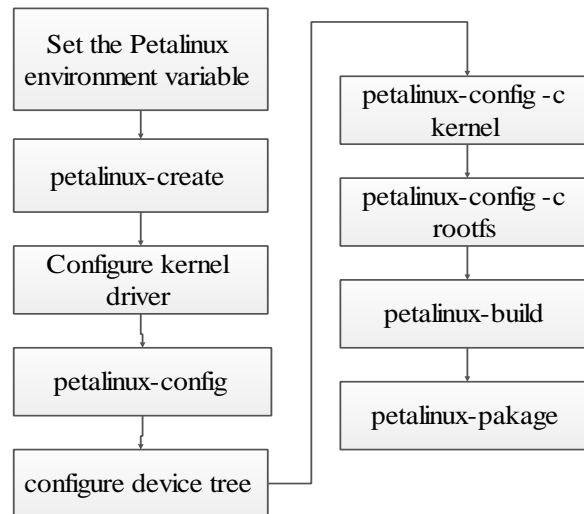
Xilinx's Zynq-7000 All Programmable System on chip which contains the dual core ARM processor with programmable logics [9]. There is a high speed AXI interface between Programmable Logic (FPGA) and Processing System (ARM Processor) through which it is possible to transfer data between PL and PS. In [10] data transfer between PL and DDR Memory using high speed AXI bus is explained. From PS it is possible to access same DDR memory. So by performing some operations on PL it is possible to accelerate the performance at PS.

### **B. Vivado**

Vivado and SDK used for create a network's block diagram, bitstream and generate a hardware description file (\*.hdf file). Vivado Design Suite is a software developed by Xilinx for analysis of HDL designs. It is used for system on a chip development and high-level synthesis.

### **C. Petalinux**

Software development for the board, we need to install the Petalinux tool in Linux. "Petalinux is the Embedded Linux System Development Kit specifically for FPGA-based System-On-Chip designs" [3]. It is used to customize, build and deploy embedded Linux system in ZYNQ. Petalinux involves of three key elements: pre-configured binary bootable images, fully customized Linux for the Xilinx device, and Petalinux SDK, which includes tools and utilities to computerize complex tasks for configuration, build, and deployment. With the petalinux, we can easily create a kernel. Petalinux provides the customized Linux distribution. Following are the steps of the Petalinux simulation.



*Figure 2 Flow of Petalinux Simulation*

### **D. Implementation**

After the decompressed the packets of petalinux in the file system of Linux run the script file to enter or register the environment \$PETALINUX and complete the installation. [5] In process of petalinux kernel configuration, need to select the hardware description file (\*.hdf file). The Hardware description file contains the structure, behavior and all the details about the hardware components. which is created in the vivado software [4]. The Figure 3 shows hardware design which is used in experimental setup.

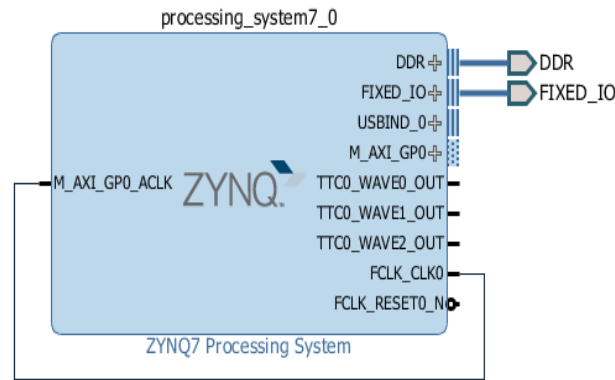


Figure 3 Hardware design

The hardware description file is stored in the vivado project directory [6]. The operation is petalinux-config --get-hw-description=/path for \*.hdf file. This command automatically takes \*.hdf file and the BSP and configure the project. BSP is the Board Support Package which contains the device drives, IP core, kernel, and boot loader configurations. To run and start up an embedded target processor and fundamental software in board BSP will be used. A Board Support Package delivers a standardized interface between the Board and the operating system. BSP provides an interface to device drivers which allow to the kernel to communicate with the Board's resources likes microprocessor, memory, internal and external busses, and device controllers. BPS saved in the single board computer, system on chip, wireless hardware [2]. BSP performs the specific task, initialize processor, bus, clock, RAM and RUN the boot loader.

When the petalinux kernel compilation is completed get the binary file image.ub, as well as the uboot configuration file. Image.ub is Linux image. It compares the version of Linux. It will be used for loading the Linux os on the board. After the above set is complete configure the project for that run these command petalinux-config

After the project is configured to build the project and generate the Boot file. There are three files are required to generate a boot.bin: first stage boot loader (FSBL), Bitstream and uboot files. BOOT.BIN image is to boot the system on the hardware [3]. For loading the bitstream and configure the zynq architecture at Boot time, FSBL will use and All the programmable information about the FPGA will be contain on the Bitstream. The Boot file handle the initial boot process and load the main Linux image [4].

The next step to configure the Linux kernel to include support for drivers and set the default option primary SD because Zedboard can be booted with the use of SD card. Petalinux provide the inbuilt option to install the third party application like python which shown in Figure 4 [7]. This can be done by these simple command petalinux-config -c kernel

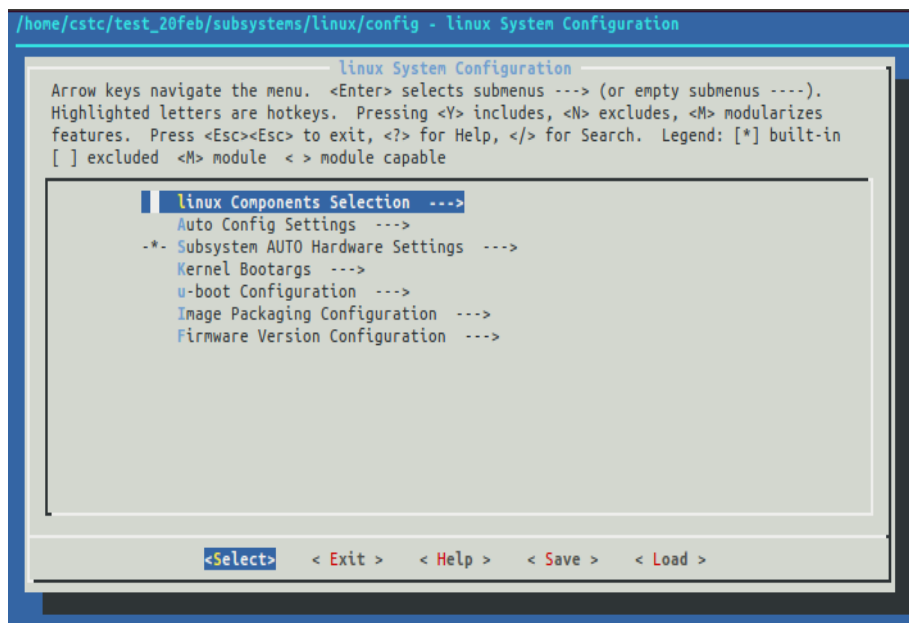


Figure 4 Configuration of Linux System

Finally, the files are generated, which is used to run the Linux OS on an FPGA which is BOOT image, device Tree file, Linux Kernel Image and the root file system. Linux kernel is running on the hardware, so they know each detailed of the hardware where it is running on, so the device tree binary file is used which contains the hardware details. There is SD card which is used for the Booting the OS on the Board. There are two partitions in the SD card, first is the fat32 partition (BOOT partition) which assessable by the Linux and Window OS and the Second partition is the ext4 (ROOT partition). It is readable/writable by the Linux OS.

TABLE 1  
PARTITION OF SD CARD

	Partition 1	Partition 2
Lable	BOOT	root
System	Fat32	Ext4
Size	512MBI	Remaining all

The BOOT partition contains the BOOT image, Device tree file, Linux Kernel Image. The root partition contains the root file system; it contains all the necessary operating system files as shown in Table1. It saved only important folders which going to be used, which is a dev, lib or lib32, root, home, mnt, user, proc, sys, init.

### III.EXPERIMENT ON BOARD USING CUSTOM HARDWARE-SOFTWARE DESIGN

Zedboard includes the custom installation of the various hardware and software modules that allows to modify and isolate the board configuration according to the custom requirement of the system. It supports the custom installation of the tiny OS and various connection configuration that matches the corresponding requirements of the project and does not additionally includes the optional components that are already integrated on the other boards. Once the OS Boot on the Zedboard, Perform the simple python program of calculation of vector and non-vector multiplication with the size of 10k with the use of numpy library.

#### A. Result

All the required files are copied into the SD card. The SD card can be inserted into the Zedboard and connect power supply to Zedboard barrel jack, connect USB-UART port of Zedboard to Pc using the MicroUSB cable. After that Board can be connected with the Tera Term. If the prompt, ask for the login it means OS booted on Board. Petalinux provide the installation of third party application like python that can be add when configure the Linux kernel. Calculate the vector and non-vector multiplication with the size of 10k. The time is less required for the multiplication is performed on Zedboard instead of computer.

TABLE 2  
RESULT OF ZEDBOARD V/S JUPYTER NOTEBOOK

	ZEDBOARD	JUPYTER NOTEBOOK
Vector Multiplication	23.6ms	27.96ms
Non-vector multiplication	65.84ms	66.00ms

```
COM4:115200baud - Tera Term VT
File Edit Setup Control Window Help
SF: Detected S25FL256S_64K with page size 256 Bytes, erase size 64 KiB, total 32
MiB
In: serial
Out: serial
Err: serial
Net: ZYNQ GEM: e000b000, phyaddr 0, interface rgmii-id
eth0: ethernet@e000b000
Hit any key to stop autoboot: 0
boot Petalinux
Device: sdhci@e0100000
Manufacturer ID: 3
OEM: 5344
Name: SL08G
Iran Speed: 50000000
Rd Block Len: 512
SD version 3.0
High Capacity: Yes
Capacity: 7.4 GiB
Bus Width: 4-bit
Erase Group Size: 512 Bytes
reading image.ub
7556700 bytes read in 647 ms (11.1 MiB/s)
## Loading kernel from FIT Image at 01000000 ...
Using 'conf@1' configuration
Verifying Hash Integrity ... OK
Trying 'kernel@1' kernel subimage
Description: Petalinux Kernel
Type: Kernel Image
Compression: gzip compressed
Data Start: 0x010000f0
Data Size: 7540372 Bytes = 7.2 MiB
Architecture: ARM
OS: Linux
Load Address: 0x00008000
Entry Point: 0x00008000
Hash algo: crc32
Hash value: 1ed37650
Verifying Hash Integrity ... crc32+ OK
## Loading fdt from FIT Image at 01000000 ...
Using 'conf@1' configuration
Trying 'fdt@1' fdt subimage
Description: Flattened Device Tree blob
Type: Flat Device Tree
Compression: uncompressed
Data Start: 0x01731068
Data Size: 15032 Bytes = 14.7 KiB
Architecture: ARM
Hash algo: crc32
Hash value: da915a25
Verifying Hash Integrity ... crc32+ OK
Booting using the fdt blob at 0x1731068
Uncompressing Kernel Image ... OK
Loading Device Tree to 07ff9000, end 07ffab7 ... OK
Starting kernel ...
```

Figure 5 Booting The OS

#### IV.CONCLUSION

The implementation of custom image of Linux operating System using petalinux is the major work of this paper. The Target hardware platform is the zynq-700, which is low cost, low power consumption and fast implementation due to its ARM processor. Using Programmable logic (PL) – Processing System(PS) interface and common shared memory between PL-PS it is possible to perform certain task on PL to accelerate the process. Advantage of the hardware is, reconfigurable logic blocks and easy to maintenance.

#### REFERENCE

- [1] Sanjay Patel and Wen Mei W. Hwu. "Accelerator architectures.", IEEE Micro, Vol. 28, No.4, pp. 4-12, 2008
- [2] ZedBoard. Zedboard Product page. [online]http://zedboard.org/product/zedboard.
- [3] Xilinx Zynq "Petalinux SDK User GuideUg977", Vol 13.04 April2013
- [4] Xilinx. Xilinx vivado Product page. [online]http://www.xilinx.com/products/design-tools/vivado.html.
- [5] Ji-gang Tong, Zhen-xin Zhang, Qing-lin Sun, Zeng-qiang Chen "Design of Wireless Sensor Network Node with Hyperchaos Encryption Based on FPGA", International Workshop on Chaos-Fractals Theories and Applications, pp.190-194 2009
- [6] Xilinx UG998: Introduction to FPGA Design with Vivado High-Level Synthesis, [Online].  
http://www.xilinx.com/support/documentation/sw\_manuals/ug998- vivado-intro-fpga-design-hls.pdf
- [7] Petalinux Application Development Guide from Xilinx Inc., UG981. [online]
- [8] Apurva Choudhary, Jaimin B Chavda, Amit P Ganatra, Rikin J Nayak. "Performance evaluation PL330 DMA controller for bulk data transfer in Zynq SoC", IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), pp. 1811-1815, 2016
- [9] Xilinx Zynq, "Zynq-7000All Programmable SoCTechnical Reference manualUG585" Vol. 1.10, February 2015.
- [10]Nayak, Rikin J., and Jaiminkumar B. Chavda. "Proficient Design Space Exploration of ZYNQ SoC using VIVADO Design Suite: Custom Design of High Performance AXI Interface for High speed data transfer between PL and DDR Memory using Hardware-Software Co-Design." International Journal of Applied Engineering Research, pp. 8991-8997,2018