

Design of a Generalized Coprocessor for Software Defined Radios on a Reconfigurable Computing Platform

Krishnamurthy Vaidyanathan¹, P. Nagaraju², Barun Sharma³ & K.Shivaprakash⁴.

¹EI Labs India Pvt Ltd

^{2&3}Telecommunication Department, RVCE, Bangalore-560059

⁴SASTRA University

E-mail: krishna@eilabs.co.in¹, nagarajup@rvce.edu.in²,
barunshrm@gmail.com³, prakaashkarthik@gmail.com⁴

Abstract – Software defined Radio platform developed so far are designed for particular application. Current reconfigurable computing technology allows designers to implement complete embedded computing systems on a single FPGA. This paper concentrates on designing a reconfigurable platform which consists of reconfigurable coprocessors that can be used in various applications. The architecture that we have developed for the generic reconfigurable coprocessor is well suited for multiple application domains. It ties in to the sequential RISC processor based C programming framework where in the coprocessor performs specific functions, replacing a C function call. The methodology adopted to develop reconfigurable platform can be implemented for performing mathematical operation such as addition, subtraction and to perform complex operation such as Autocorrelation, it can also be used in Image Processing, Cryptography etc. As an example to illustrate this methodology reconfigurable coprocessor consisting of wrapper module along with controller module are designed to control external device like LCD. The architecture of the proposed methodology presented in this paper were prototyped using a Cyclone IV Starter Board (DEO Nano), which is based on Nios II Embedded Evaluation Kit.

Keywords – FPGA, Reconfigurable, Quartus, LCD.

I. INTRODUCTION

Embedded computing systems typically comprise both processors and dedicated logic modules to meet design specifications that include performance, area, power, and cost constraints. This has led researchers to investigate numerous issues that arise from Hardware/Software co design. Although older systems combined fixed processors and integrated circuits, current technology allows designers to combine both processors and dedicated logic to implement complete embedded computing systems as Systems-on-Chip (SoC) using either ASIC or FPGA platforms[1].

Communications devices designed with application-specific integrated circuit (ASIC) technology suffer from one very significant limitation—the integrated circuits are not programmable. Therefore, deploying a new algorithm or an updated standard requires new hardware. Field-programmable gate arrays (FPGAs) solve this problem by introducing what is essentially reconfigurable hardware.

In this paper we prepare a design methodology to develop reconfigurable SDR platform using FPGA. The remainder of the paper can be categorized of following sections which are described as follows. Section II gives overview of SDR and basic definition related to reconfigurable platforms .Section III gives the descriptive view of various platforms that can be used to design Reconfigurable platform . Section IV gives the general proposed methodology and its application that can be used in various technologies. Section V deals with Software package required. Section VI deals with results obtained and implementation of the proposed methodology using these software package .Section VII deals with conclusion and at last section VIII deals with the references required to come up with this methodology.

II. SDR OVERVIEW

Software Defined Radio is a collection of hardware and software technologies that enable reconfigurable system architectures for wireless networks and user terminals. SDR provides an efficient and comparatively in-expensive solution to the problem of building multi-mode, multi-band, multi-functional wireless devices that can be enhanced using software upgrades. It is applicable across a wide range of areas within the wireless industry. Reconfigurability for an SDR can be “static” or “dynamic.” Static reconfigurability is the

ability to reconfigure equipment capability ‘off-line’ (e.g., at supply), or via a smart card. Pseudo-static reconfigurability is the ability to reconfigure equipment capability over-the-air (OTA), while at the same time leaving it un-changed during a call. Dynamic reconfigurability, on the other hand, is the ability to auto-re-configure equipment capability during a call. Fundamental differences exist in the manner in which systems using SDR technology are implemented in the commercial, government, and military wireless-market segments.

III. PLATFORM DEFINED FOR SDR

Software-defined radios have been employed on numerous platforms, including general purpose microprocessors (GPP), digital signal processors (DSP), and graphics processing units (GPU). General purpose microprocessors, such as the Intel and AMD devices commonly found in personal computers, are not specialized for any particular application. Therefore, they are very flexible[2]. However, SDR systems using GPPs are often wasteful since these processors are designed for speed and generality rather than power efficiency or mathematical operations. Digital signal processors solve these two problems. DSPs, such as those manufactured by Texas Instruments, are specialized for performing mathematical operations and typically contain less hardware, increasing power efficiency. On the other hand, their narrow focus makes them potentially slow for other applications. Finally, graphics processing units employ massively parallel architectures that are optimized for vector manipulations and other graphical operations. Such parallel designs are well-suited for signal processing, but GPUs are still relatively difficult to program and they are extremely power hungry. A fourth option is the field-programmable gate array (FPGA). FPGAs are reconfigurable logic devices that enable highly parallel implementations of digital signal processing algorithms. They are becoming increasingly power efficient while simultaneously including specialized hardware for some DSP operations. In addition, they can be reprogrammed remotely. Due to these features, FPGAs make an ideal software-defined radio platform. Table 1 shows the comparison of microelectronics platform for SDR.

Feature	GPP	DSP	GPU	FPGA
DSP Operations	Moderate	Good	Good	Good
General Operations	Good	Poor	Poor	Poor
Flexibility	High	Low	Moderate	High
Size	Moderate	Small	Large	Large
Power Efficiency	Moderate	Good	Poor	Moderate
Common Brands	Intel, AMD	Texas Instruments	nVidia, AMD	Xilinx, Altera
Programming Language	C, Java	C, Assembly	CUDA, C	Verilog, VHDL

Table 1 : Comparison of different SDR platforms.

IV. PROPOSED METHODOLOGY

Design of the specified methodology contains one processor with storage memory along with buffers and parallel to serial converter. For verification and testing external device is required. Figure 1 gives the descriptive view of predefined methodology.

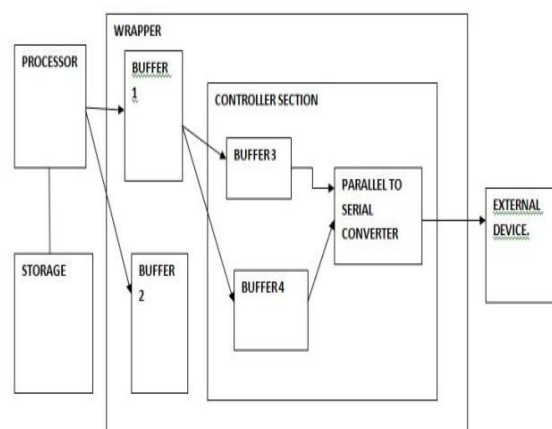


Fig. 1 : Defined Methodology

In figure 1 defined processor can be any processor but according to user point of view and due to fast processing speed preferred processor is NIOS processor. This methodology can be used to carry out any mathematical operation like calculating Autocorrelation, simple binary operation can be performed easily and we can reconfigure the defined system as per our requirements. It can also be used in cryptography and many more technologies. In this section focus will be more on illustrating the use of methodology with two suitable examples.

A. Performing mathematical operation.

As in the methodology defined there is one processor defined with one storage memory. Let us take a suitable example to perform Autocorrelation operation using this. Suppose buffer 1 defined in the above consist of one matrix of (2X2) and buffer 2 consist of other matrix of (2X2). Now to perform autocorrelation one has to perform matrix multiplication.

$$\begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix} = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \cdot \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix} \quad (1)$$

Information from buffer 1 is passed to buffer 3 and information from buffer 2 to buffer 4. Buffer 3 and buffer 4 forwards the existing matrix in them to multiplier which is defined as parallel to serial converter in our case, and after multiplication the required output which consist of matrix C as defined in equation 1 is

displayed on external device that can be LCD which is compatible to display defined result or one can choose any suitable display source.

B. Cryptography.

The defined methodology can also be used in cryptography and can be used to transfer secured data in terms of any media types .Let us take an example of transferring a secured image. Consider buffer 3 and buffer 4 consist of two images, in which buffer 3 consist of real image and buffer 4 with duplicate. As we have defined our structure as reconfigurable on the software side so we are removing the buffer 1 and buffer 2 and connect buffer3 and buffer 4 directly to processor such as NIOS. At the encoder side we multiply these two images and transfer and at decoder side these two are decoded and original image can be extracted. Image multiplication can be done as shown in Figure 2.As we can see in the figure defined that two pixel values a(1,1) and a(1,2) multiplies with b(1,2) and b(2,2) to achieve the resultant pixel value . Similarly all the resultant image can be obtained by multiplication pixel values in matrix form.

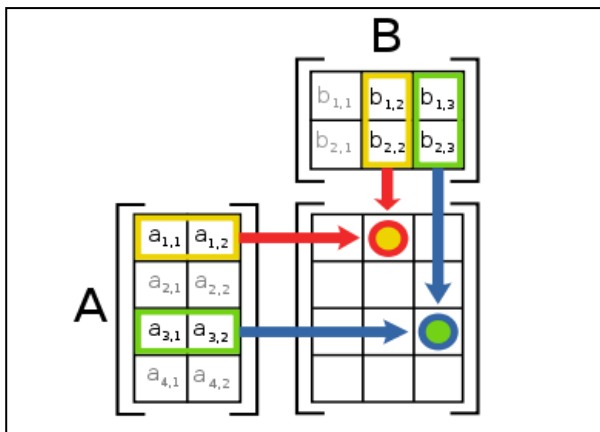


Fig. 2 : Image Multiplication Example

In this case memory are used to store pixel values which are stored in the code which can be in C or C++ and that code can be dumped in to the storage memory and NIOS processor instantiate the code and pixel values when the system is executed .

The example what has shown is an on overview of how to reconfigure the defined platform in software and different application can be performed.

V. REQUIRED SOFTWARE PACKAGE.

Basic software tools required to accomplish this project is as follows:

Modelsim - Simulation and verification of code is done using this software tool.

Quartus 12.0 - The software package by Altera enables the user to compile and synthesize the code for NIOS processor.

Qsys - Inbuilt software package by Altera Quartus helps in system on chip development and create hardware system platform for NIOS processor and joins all the necessary modules internally on FPGA board.

Altera Monitor Programme - Creates software system for reconfigurable platform. One can run and compile there own code in C/C++ and instantiate with the reconfigurable platform by NIOS using this software.

VI. IMPLEMENTATION AND RESULTS.

The proposed methodology is implemented by controlling external device lcd by using the reconfigurable platform .The implemented design of the reconfigurable platform is as shown in Figure 3 .

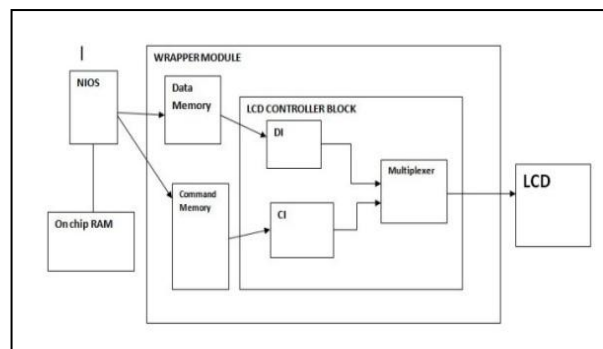


Fig. 3 : Reconfigurable platform to control lcd.

The implemented methodology can be broadly divided in two categories that are hardware system definition and software system definition . Hardware system synthesis can be implemented and designed using software package Qsys and Quartus . On the other hand software system synthesis can be implemented using Altera Monitor Programme. The defined modules in this are data memory, command memory, data interpreter, command interpreter, command interpreter, multiplexer, NIOSprocessor, on chip memory and LCD. Flow of implementation of proposed methodology is as follows:

A. Simulation and verification using modelsim.

The modules defined in this application are data memory, command memory, data interpreter, command interpreter. These modules are coded and executed separately to create a design flow. Data memory consists of data to be displayed on the LCD and command memory consists of commands that are to be executed to initialize the LCD. Both Command and Data memory are 64x8 user-defined modules with 64 address lines each of 8 bit width. After coding each module

separately, a wrapper module is created where each defined module is instantiated and inter-module connections are made. This wrapper module is then simulated on ModelSim, in which desired inputs are forced and outputs are observed. The output waveforms thus obtained are verified manually.

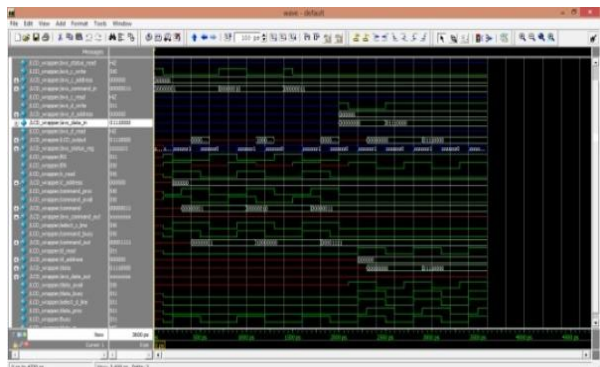


Fig. 4 : Wrapper module simulation

B. Compilation and simulation using quartus.

The System on Programmable Chip building tool Qsys from Altera is used to implement the system. The Data Interpreter and Command Interpreter modules, which are cascaded together along with the Data Multiplexer as the LCD Controller block, is defined as an Avalon Memory Mapped Slave. The architecture of this Controller block includes a Status Register in which the LSB signifies ‘Busy’ status of this module. The Data Memory and Command Memory blocks are also defined as Avalon Memory Mapped Slaves since these modules are directly connected to the Avalon bus, and thus to the NIOS II processor. This definition is done by creating a “New Component” in Qsys and ensuring that the signals and registers defined in the Verilog code match their suitable roles in the system.

Once these modules have been defined as Avalon MM Slaves, the other components of the system are included viz. the NIOS II processor, On-Chip RAM (8K), JTAG UART and a General Purpose I/O controller along with the Reconfigurable platform components. The necessary clock connections, Avalon MM Master to Avalon MM Slave connections, Interrupt vector address and interrupt network connections and export of conduits to the surface-level so that internal connections may be customized are then made. Once the connections are made in Qsys, base addresses must be assigned to the memory mapped slaves, which is done by choosing the “Assign Base address” option.

After the system thus created above is saved, the .qsys, .sopcinfo, .qip files are generated by the software. These files are to be included in the Quartus project.

In Altera Quartus 12.0, the defined modules are included in the project directory along with the .qip file generated in the previous step. After this step, the top level Verilog module must be written. This will contain declaration of all signals that are input to or output from the FPGA along with the instantiation of the system designed in Qsys. The conduits in the system must be connected to their respective pins or other modules, as the case may be. Once this top level module has been saved and included in the project directory, it must be compiled. After compilation RTL logic is obtained which gives the descriptive view of our code written in Verilog which is as follows:

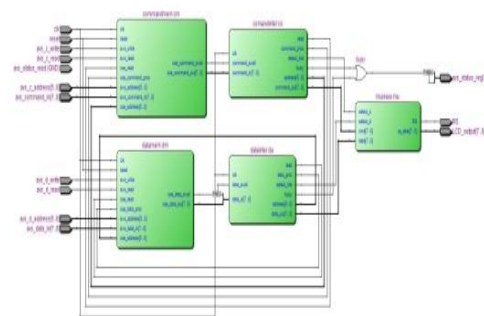


Fig. 5 : RTL view of wrapper.

The qsys file generated is shown in figure 6 as follows:

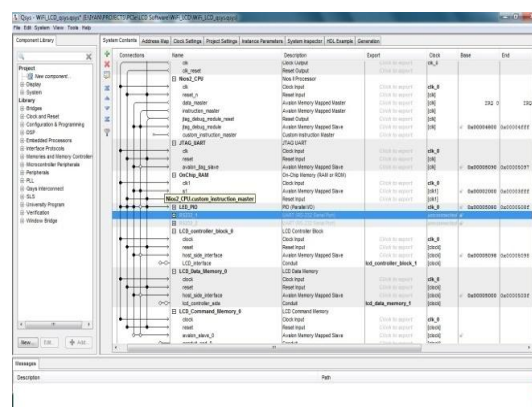


Fig. 6 : Window displaying Design flow in Qsys.

After defining the hardware system next step is to define software system. For controlling LCD proper C code is written which consist of command memory instruction and description of the of the method involve to control LCD. In brief Register select (Rs), Enable (E) and Read write (R/W) instruction is used to control LCD. The descriptive way of running LCD is shown in Table 2 which is coded in C ,Wrapper module in verilog and for verification purpose in Modelsim also. Altera Monitor Programme is used to dump the C file we obtain after converting .sopcinfo file from Qsys to .elf file which contains C code.

RS	R/W	E	OPERATION
0	0		Write instruction code
0	1		Read busy flag and address counter
1	0		Write data
1	1		Read data

Table 2 : Relation between the operation and combination of RS and R/W.

NIOS processor defined play an important role by connecting on chip ram with specific design and the overall design is controlled by NIOS processor. To check the output on Deo Nano board which consists of Altera FPGA with cyclone IV device pin assignment is done in Quartus package which gives the description that outputs are assigned to particular GPIO pins. Descriptive view of pin assignment in Quartus 12.0 is shown in fig 7.

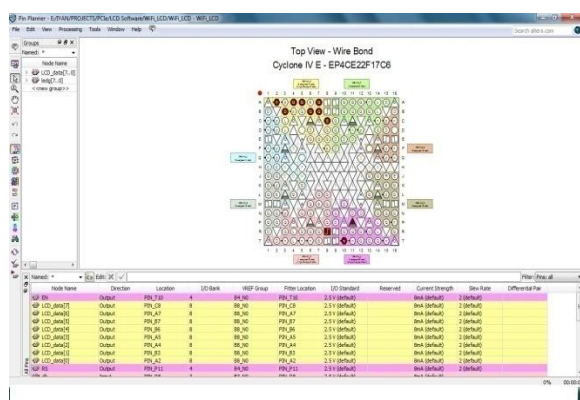


Fig. 7 : Descriptive View of pin Assignment in Quartus

To check the flow design and output in a defined way and to check the output waveform Signal Tapper is used which gives the flow with description according to the clock pulse. All the modules are exported and Hardware kit is specified before running dumping the programme.

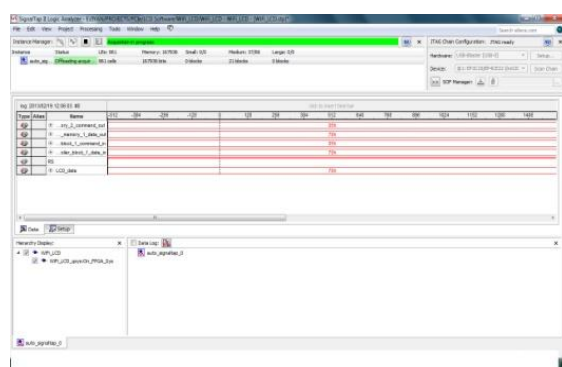


Fig. 8 : Signal Tapper output.

After checking the logic in signal tapper .sof file is burn on to the Deo Nano board and LCD (20X4) is controlled using our reconfigurable platform as shown in figure 9. One can display according to there requirements by changing the code without changing the hardware connection and system created. One can control any character LCD using this reconfigurable platform.



Fig. 9 : Lcd Character Display.

VII. CONCLUSION

A Software Defined Radio prototype is developed using Altera FPGA. The designed system consisting of coprocessors that can be used as a reconfigurable platform and one can control any external computing devices like LCD, DAC, WIFI module etc, using this prototype. Innovative design enhancements from engineering community, coupled with process improvements from FPGA vendors can play a crucial role in in producing fast, small and efficient communications systems that can be used in commercial arenas.

VIII. REFERENCES

- [1] Lesley Shannon and Paul Chow-“Levaraging Reconfigurablity in the Hardware/Software codesign process”.
- [2] Steven J. Olivieri-“Modular FPGA-Based Software Defined Radio for CubeSats”, thesis published in May 2011.
- [3] Nicholas Peter Sedcole-“Reconfigurable Platform-Based Design in FPGAs for Video Image Processing”, thesis published in January 2006.
- [4] S.Haykin, “Cognitive Radio: Brain –empowered wireless communications,” Selected areas in communications,IEEE journal on,vol.23,no.2,pp.201-220,Feb 2005.
- [5] J.Mitola III, “ Software radios: Survey,critical evaluation and future directions ,” Aerospace and

- electronic systems magazines ,IEEE ,vol.8,no.4,pp.25-36,APR 1993.
- [6] J.E.cooley,"A day in the life of rf spectrum,"Master'sthesis,Massachusetts Institute of Technology,2005.
- [7] Alexander Loula, Open BTS installation and Configuration guide v0.1,2009-05-25.
- [8] R.Vidano," Speakeasy II-an IPT approach to software programmable radio development ," MILCOM 97 Proceedings,vol.3,pp.1212-1215,Nov.1997.
- [9] J,Chapin,V.Bose , " The Vanu Software Radio System," in 2002 Software Defined Radio Technical Conference,San Diego, 2002 .
- [10] Gary J.Mullet ,"Introduction to Wireless Telecommunication System,"text book,2010.

