International Journal of Engineering Research and General Science Volume 3, Issue 3, May-June, 2015
ISSN 2091-2730

Fast Fourier Transform implementation using Microblaze and uclinux

Ravi Jani[1], Kunjal Mehta[2]

[1]jr4onlyu@gmail.com, Student Dept. EC, LJIET, Ahmedabad

[2]hikunjalmehta@gmail.com, Professor Dept. EC, LJIET, Ahmedabad

Abstract— In spite of this increased computational capacity the FPGAs have proved inefficient in comparison to their ASIC counterpart. In case of certain multimedia and signal processing application the FPGA's computational capacity proves to be inadequate. To overcome this limitations the designers have come up with number of approaches .Two of the approaches have been implemented. One approach is hardware based approach and the other is software based approach. The hardware based approach refers to resorting to multiprocessor architecture to enhance or multiply the performance of System on Chip. Another approach which is software based is useful in case hardware capacity of the FPGA is limited. This approach refers to use of an OS and boost the performance. In this project a quad processor system has been designed as well as the uCLinux RTOS has been ported to FPGA.

Keywords— RTOS, Microblaze, Soft Processor, uclinux, FFT, Fourier Transform.

I. INTRODUCTION

Since their emergence in the mid-1980s, Field Programmable Gate Arrays (FPGAs) have become a popular choice for prototyping and production of products in small to moderate quantities. An FPGA is a special kind of Programmable Logic Device (PLD) that allows implementation of general digital circuits. The circuit to be implemented is defined by programming the device. Over the years, the capabilities of FPGA devices have grown to the level where a complete multiprocessor system can fit on a single device .

FPGA is flexible because its parameters can be changed at any time by reprogramming the device. Traditionally, systems have been built using general-purpose processors implemented as Application Specific Integrated Circuits (ASIC), placed on printed circuit boards that may have included FPGAs if flexible user logic was required. Using soft-core processors, such systems can be integrated on a single FPGA chip, assuming that the soft-core processor provides adequate performance. The widely used soft core processors are Xilinx Microblaze and PowerPC and Altera's NIOS. This combination of FPGAs with embedded soft-core processor gives a cutting edge to the FPGA platform for designing system on chip.[1]

Main objective of the project has been to design a smarter system on chip. With embedded systems being used continuously for increasing number of applications designers constantly look forward to improving the performance. At times a number of applications make stringent demand on FPGA hardware and many a times these demands may not be satisfied. To overcome this limitation OS or a software layer can be used. FPGA are well known for their hardware reconfiguration capability but with the use of an OS even software reconfiguration is possible thereby increasing the flexibility for the designer.[2]

Nowadays, more and more embedded systems are using field programmable gate arrays (FPGAs) to control and process data by making use of inherent parallelism and flexibility concepts of FPGAs. Designers using FPGAs can choose and implement the exact amount and type of peripherals that are needed for the requirements of their application, having also the freedom of changing them while the design process is continuing. Suppose that a system designer prepares the requirements of an incoming project before the design phase of the product as usual. There is always a high possibility that these requirements are changed by the customer after the design phase starts. If the system designers decide to use for example a microprocessor of a particular type at the beginning of the project, software engineers may experience difficulties to fulfill the incoming requirements later because of the inflexible hardware architecture of this particular microprocessor. By using FPGAs on the other hand, software of the product may be protected and may be implemented in a processor independent way and the designers may not suffer from processor obsolescence.

If the decision is to use FPGAs in a project, designers can have more advantages by opting for soft processors embedded in FPGAs. Today's embedded systems must be power-efficient, sufficiently small and above all, cheap, to be commercially viable. If an embedded design uses a microprocessor, one needs to have an extra flash memory and RAM for booting the software when the system is powered up. However, FPGAs have built-in flash memory and RAM, so designers can save area, power and money by not using such extra peripherals. As a matter of fact, if a standard processor is sufficient to fulfill the requirements of an embedded project, it

may be wise to use it. But if an FPGA is already employed for some other purposes then it may be cheaper and more area-efficient to use an embedded processor in the design.[4]

Using an embedded soft processor on the other hand has some disadvantages. Because of the integration of the hardware and software platform design, the design tools are more complex and relatively immature compared to standard processor design tools.

II. MICROBLAZE

Microblaze is the embedded soft core processor used here. A number of features offered by it makes it very flexible and suitable for the designer. Following figure shows the microblaze core. This soft processor core can be implemented in any of the Virtex Architec utre. It typically occupies about 1050 to 2000 slices on FPGA.[7]

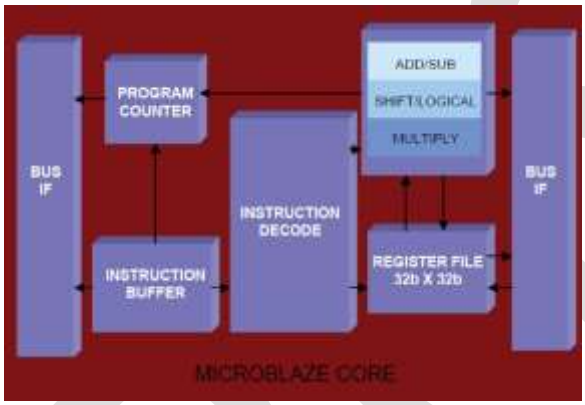

Fig.1 Microblaze Core

Microblaze core is a Reduced Instruction Set Computer(RISC) optimized for the implementation in Xilinx FPGA. The detailed core block diagram is shown in the fig. below. Apart from the selective features Microblaze offers the following fixed features:

- Harvard Architecture.
- Thirty two 32 bit General purpose registers.
- 32 bit address bus
- 32-bit instruction word with three operands and two addressing modes.
- 3 pipeline stages(Fetch ,decode and execute) and single issue pipeline.
- Big-endian address format.

Xilinx ML505 FPGA Board has been used. The architecture is Virtex 5,device size xc5vlx110t with packaging ff1136 and a speed grade of -1. irtex®-5 FPGAs are the world's first 65nm FPGA family fabricated in 1.0v, triple-oxide process technology, providing up to 330,000 logic cells, 1,200 I/O pins, 48 low power transceivers, and built-in PowerPC® 440, PCIe® endpoint and Ethernet MAC blocks, depending upon the device selected.

Features of Viretx-5 FPGA Family are:

- On average, one to two speed grade improvement over Virtex-4 devices
- Cascadable 32-bit variable shift registers or 64-bit distributed memory capability
- Superior routing architecture with enhanced diagonal routing supports block-to-block connectivity with minimal hops
- 3,30,000 logic cells

- 207,360 internally fabricated flip-flops with clock enable.
- 207,360 real 6 input LUTs and upto 13 million total LUT bits.
- 550 MHz clock technology.
- 550 MHz DSP48E slices for enhanced performance in DSP applications.
- LVCMOS(3.3V, 1.8V, 1.5V, 1.2V)
- Each CLB is made up of two slices and each CLB consist of four function generator, four storage elements, arithmetic logic gates, fast carry look ahead chain and multiplexers.[7]

III. UCLINUX

There are many third-party companies giving RTOS support for Xilinx soft processor MicroBlaze. In Table 1, a list of some third-party companies that supports MicroBlaze and their RTOS products are given [1].

Table 1. Third-Party RTOS Companies Supporting MicroBlaze

Company	Product
eSOL Co., Ltd	PrKernel (μITRON4.0)
Express Logic	ThreadX®
Mentor Graphics ESD	Nucleus Plus
Micriμm	μC/OS-II
MiSPO	NORTi/ulTRON
PetaLogix	uClinux and Petalinux 2.6

Using a real time operating system (RTOS) on processors is another trend that designers increasingly follow due to RTOSs' deterministic behavior and efficient resource management characteristics. Ability to create tasks to handle and distribute huge sized codes, existence of scheduling algorithms to manage the tasks and efficient interrupt handling and faster memory allocation are some of the many advantages of using an RTOS.

A standalone system possesses a processor, which has no operating system running on it. By running an RTOS on such a processor, the resources of the embedded system might be managed more efficiently. Using an RTOS, 'tasks' can be created to perform the duty. Priorities can be assigned to these tasks, i.e., software engineers may decide which functions of their software are more important than others. Another feature of RTOSs named as semaphores increase the predictability of the software by helping in switching the tasks safely. Each RTOS company provides a different set of application programmers interfaces (APIs), but in summary, nearly all of these provide fast memory allocation, preemptive scheduling and deterministic latency. The more software engineers have precise information about what's going on in their systems, the more their software becomes reliable.

Linux(prono unced as you see Linux) is an operating system with embedded systems, microprocessors and microcontrollers as its target systems. It is nothing but port of the Linux kernel. Although kernel version is same as Linux uCLinux does not have any Memory Management Unit(MMU).This is in agreement with the fact that target systems of this OS are the embedded systems or microcontrollers which do not possess MMU. With embedded linux going mainstream along with embedded processing on FPGA, uCLinux aptly epitomizes this advancement in embedded designing.

The major motivation behind selecting uCLinux is because it has been already ported to a number of targets including microblaze,coldfire and dragonball,ARM,Blackfin just to name a few. Moreover scheduling, threading and interrupt handling is also available.To understand the difference between Linux and uCLinux its necessary to understand a Linux system and than differentiate between Linux and uCLinux system.[10]

This in detail explanation is enough to point out the difference between the Linux and uCLinux operating systems. The Linux kernel has following characteristics:

- Originated on i386 architecture
- Possesses a MMU.
- Virtual Memory available.
- Memory protection.

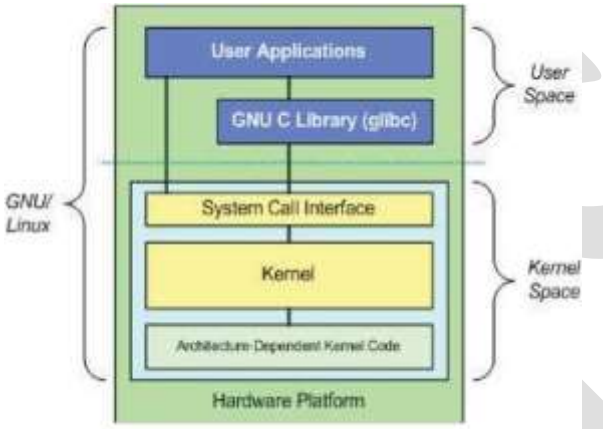

Fig.2 Anatomy of Linux system.[10]

On the other hand the uCLinux kernel possesses following characterstics:

- No MMUs
- No virtual memory.
- No memory protection

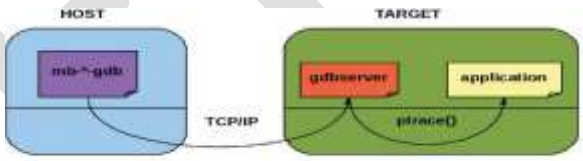

Fig.3 Cross Compilation.

IV. FAST FOURIER TRANSFORM

When number of data points N in DFT is a power of 4 than it is computationally efficient to use Radix 4 algorithm. Of course Radix 2 can be used as well but it will require more no. of iterations as compared to Radix 4. As for example for calculating 16 point DFT using Radix 2 algorithm 4 stages are needed whereas this reduces to 2 stages in case of Radix 4 algorithm.The number of multiplications and additions required is also more in Radix2 as compared to Radix 4. However the only limitation is that N should be power of 4.

The N point sequence is divided into N/4 point sequence. Thereafter these sequences are again divided into four subsequences and it goes on for lo g(N,4) times and finally these subsequences are added together to obtained the DFT. This algorithm has been widely used since it boost up the speed of the computations drastically. And higher the value of N more the speed up. Using a multiprocessor system this speedup can be further increased.

Here the Decimation in frequency algorithm has been implemented. So inputs are in normal order and output in digit-reversed order.

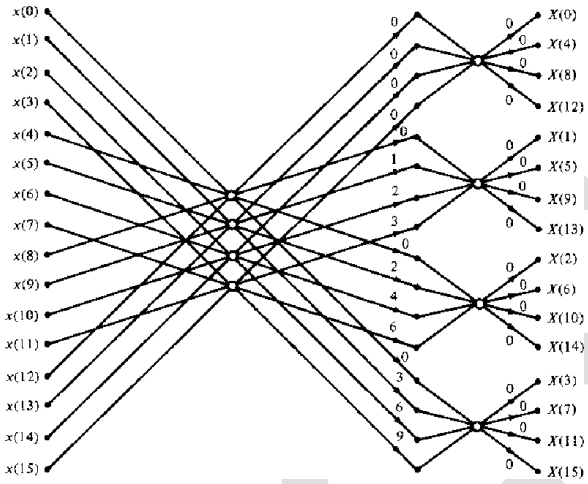

Fig.4 Radix-4 FFT.

V. IMPLEMENTATION

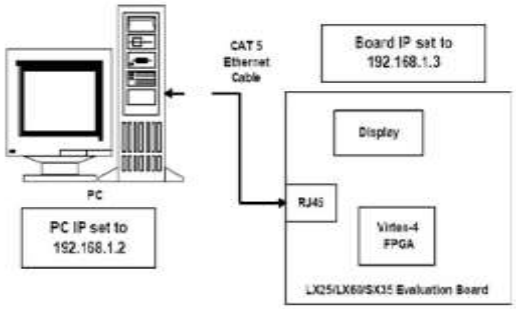

Fig.5 Setup.

As seen above in Xilinx platform studio all microblaze are made and they were implemented on kit. Now for uclinux it was ported to microblaze. Then logic for FFT was implemented as shown in figure.

Using the SPARK tool, successfully implemented a number of C codes which include sorting minimum and maximum element in an array ,average of numbers and inverse discrete cosine transform. All these codes have been successfully downloaded on FPGA and outputs verified accordingly. In each of the codes the to verify the output a done bit was used. Whenever the desired output was obtained the done bit was set and to indicate this LED was used i.e. when done bit was set the LED glows.

- In the first step a Microwave Software Specification file is auto generated but can be modified by user at later stage if need arises. It contains all project software options like C compiler options ,driver info etc
- Library Generator(LibGen) which configures libraries and device drivers. Creates xparameters.h include file for driver.Creates libc.a,libm.a,libxil.a which microprocessor can access.
- GNU base software compilation for microblaze is supported by XPS.

- Finally the generated bitstream can be merged with hardware flow or if it is executable in off-chip memory it can be directly downloaded to FPGA by GDB/XMD.
- FFT algorithm computed.
- Applied on board.
- Results compared for both systems.

VI. RESULTS

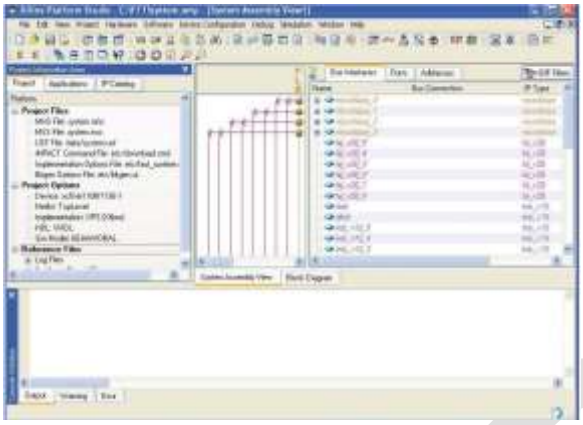

Fig.6 4 Microblaze Configured.

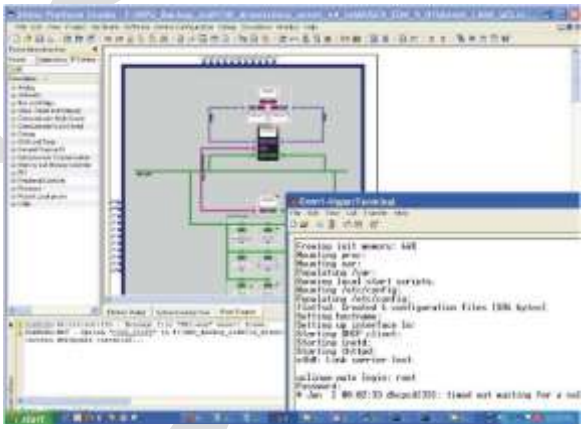

Fig.7 Porting Linux(1)

Fig.8 Porting Linux(2)

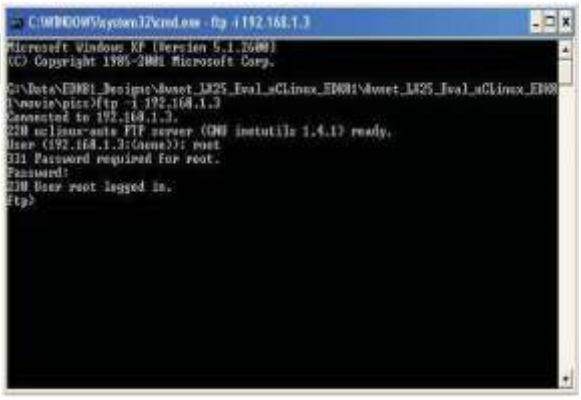

Fig.9 Logging into Linux

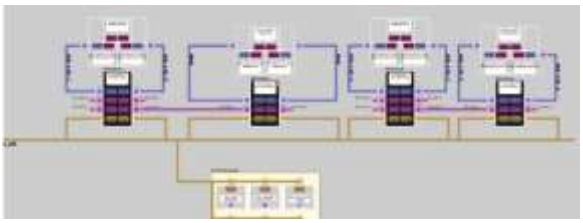

Fig.10 Diagram with BUS logic.

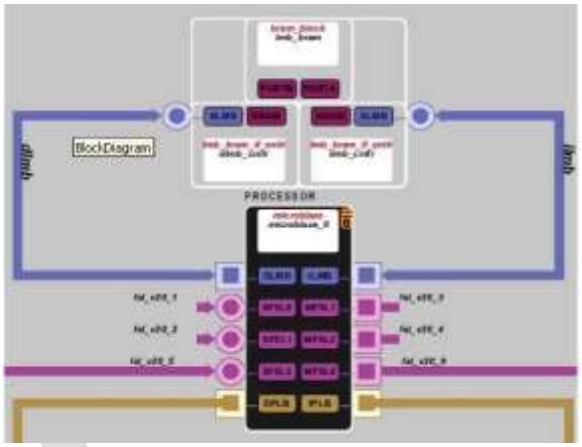

Fig.11 Microblaze with FSL port

Table 2. Comparing Single vs. Multi processor system outputs for Microblaze.

Input	Single Processor System	Multiprocessor System
Unit Impulse	35545700	1407219
Unit Step	35545744	1407248
Ramp	35545786	1407289

VII. CONCLUSION

The FPGAs so far were well known for their hardware flexibility can now also provide software flexibility with uCLinux ported on it. A number of applications which could not be implemented on FPGA due to hardware limitations can now be implemented using software layer. Commercialization can be very useful for the contemporary PDAs, cellphones and embedded systems.**[16]**

The developed project and multiprocessor system designed can be used for number of other applications. A number of algorithms in Digital Signal Processing can be implemented on the designed system. Convolution, Filter Designing, Modulator — demodulator can be implemented on the designed system. The multiprocessor system can be put to use wherever multitasking or parallel operation is required.

I have successfully ported and booted uCLinux on Virtex-4 FPGA. Moreover I have successfully used the SPARK synthesis tool and demonstrated its use by implementing a number of codes and synthesizing them successfully and than downloaded the same on FPGA.I have also developed a quad processor system and implemented Radix4 FFT algorithm and obtained some encouraging results. However means are equally important as the end itself. While developing the system I got to know a number of tools and softwares.

REFERENCES:

1) Stephen Brown and Jonathan Rose, *Architecture of FPGA and CPLD*, Department of ECE , University of Toronto.

2) Franjo Plavec, *Soft Core Processor Design*, University of Toronto.

3) Humberto Calderon, Christian Elena and Stamatis Vassiliadis, *Soft core processors and Embedded processing:survey and analysis*, EEMBCS, TUDelft, Netherlands

4) Gokhan Ugurel, Dept. of Image Processing, Turkey, Cuneyt F. Bazlamacci, Dept. of Electrical and Electronics Eng., Turkey, Context switching time and memory footprint comparison of Xilkernel and µC/OS-II on MicroBlaze , 7th International Conference on, Electrical and Electronics Engineering (ELECO), 2011 (ISBN : 978-1-4673-0160-2).

5) Nan Xiang, Xiaodan Guan,Tong Zhou, Dept. of Electronic Engineering, A design for 2-dimensional DCT transform kernel based on MicroBlaze soft core , World Automation Congress (WAC), 2012 (ISBN : 978-1-4673-4497-5).

6) Poonm S. Isasare, Mahesh T. Kolte, University of Pune, Pune, India, Fast Fourier Transform Implementation on FPGA Using Soft-Core Processor NIOS II , IJETT, International Journal of Engineering Trends and Technology , *Volume 8 No 10* 2014 (ISSN : 2231-5381).

7) Xilinx Microblaze Datasheet.

8) Embedded Linux on Xilinx Microblaze presentation manual, Xilinx University Program

9) Somen Burma,*Design,De velopment and Performance Evaluation Of Multiprocessor System on FPGA.*IIT Delhi

10) Xilinx OS and Libraries Document Collection.

11) Kyan Fraser, *An FPGA running uCLinux on a Microblaze soft processor*, School of Computer Science and Engineering, University of New South Wales.

12) Emi Eto "Difference Based Partial Reconfiguration", XAPP290 (v2.0). Xilinx Inc., December 2007.

13) Theresa Chou, *Microblaze Overview*

14) John G. Proakis & Dimitris G. Manolakis, *Digital Signal Processing:Principles, Algorithms and Applications*-Fourth Edition,Prentice Hall of India Publishers

15) Cooley J.W. and Turkey J.W., *An Algorithmfor the Machine Computation of Complex fourier series,* Math. Comp., Vol. 19,pp.297-301,April 1965.

16) M.Simek, *Embedded Operating System for Microblaze,* Acta Polytechnica.