FPGA BASED RISC AND DSP SYSTEM DESIGN

Jivin M
PG student, VLSI & Embedded Systems, ECE Department
TKM Institute of Technology
Karuvelil P.O, Kollam, Kerala-691505, India
jivinmurali@gmail.com

***Abstract*— Nowadays most of the microprocessor and microcontroller designs are based on Reduced Instruction Set Computer (RISC) core and many operations such as Discrete Cosine transform (DCT), Inverse DCT, Discrete Fourier Transform (DFT) and Inverse Discrete Fourier Transform (IDFT) are performed by DSP system. The concept of RISC architecture involves an attempt to reduce execution time by simplifying the instruction set of the computer and a Digital Signal Processor is a specialized microprocessor with an architecture developed for the fast operational needs of digital signal processing. A RISC (Reduced Instruction Set Computer) and DSP system which can perform Arithmetic, Logic and DSP operations are proposed. The processor use a 4 bit opcode and it can perform 15 different operations which include Arithmetic, Logic and DSP operations like DCT, IDCT, DFT & IDFT. The RISC machine fetches an instruction from memory. The instruction is 20 bit out of which 0-3 bits represent an opcode which decide the operation to be performed, 4-11 and 12-19 bits represent the registers holding the values to be used for the instructions. The output is of 8 bit value. The coding is done in VHDL language, synthesized using Xilinx ISE 13.2 and simulated using ModelSim SE 6.2c.**

Keywords—*RISC, DSP, FPGA, DFT, IDFT, DCT, IDCT, opcode*.

I. INTRODUCTION

Reduced Instruction Set Computer (RISC) architectures represent an important innovation in the area of computer organization. This architecture attempt to produce more CPU power by simplifying the instruction set of the CPU. Reduced Instruction Set Computer (RISC) use fewer instructions with simple constructs, therefore they can be executed much faster within the CPU without having to use memory as often. The concept of RISC architecture involves an attempt to reduce execution time by simplifying the instruction set of the computer [4].

Main features of a RISC processor are-

1. Relatively few instructions.
2. Most instruction is register based.
3. Relatively few Addressing modes.
4. Better compilation.
5. Fixed length, easily decoded instruction format.
6. All operation done within the registers of the CPU.
7. Efficient and optimization of instruction pipeline.
8. Hardwired controller instructions (as opposed to microcoded instructions).

The most important feature of RISC instruction format is to decode the information. It has the ability to execute one instruction per cycle. This is done by overlapping the fetch, decode, and execute phases of two or three instructions by using a procedure referred to as pipelining. Instructions are of fixed number of bytes and take fixed amount of time for execution [13]. RISC implements each instruction in a single cycle using a distinct hard-wired control at lesser amount of circuitry and thus, power dissipation because of its reduced instruction set. A Digital Signal Processor is a specialized microprocessor with an architecture developed for the fast operational needs of digital signal processing. Digital Signal Processor is optimized specially for digital signal processing. It also support features as an applications processor or microcontroller. DSP operations process the continuous signals and data. Digital signal processing is used in many aspects of industry. Examples of applications include speech synthesis, speech recognition, and high-speed modems. The main advantage of digital processing over analog processing is its ability to both process data and to control data based on earlier results [10]. The most important feature of a DSP is its ability to support repetitive and numerically intensive tasks. This ability is used in its calculation of Fourier transforms, multi-filter systems and correlation calculations. The ability to perform a multiply-accumulate operation in a single clock cycle is the key. The multiply-accumulator is integrated into the data path. Digital signal processing algorithms typically require a large number of mathematical operations to be performed quickly and repeatedly on a set of data. Signals are constantly converted from analog to digital, manipulated digitally, and then converted back to analog form. Many DSP applications have constraints on latency; that is, for the system to work, the DSP operation must be completed within some fixed time.

II. PROJECT OVERVIEW

Most of the microprocessor and microcontroller designs are based on Reduced Instruction Set Computer (RISC) core and many operations such as Discrete Cosine transform (DCT), Inverse DCT, Discrete Fourier Transform (DFT) and Inverse Discrete Fourier Transform (IDFT) are performed by DSP system. The concept of RISC architecture involves an attempt to reduce execution time by simplifying the instruction set of the computer and a Digital Signal Processor is a specialized microprocessor with an architecture developed for the fast operational needs of digital signal processing. The project focuses on developing a 20-Bit RISC and DSP System

described using VHDL which can perform Arithmetic, Logic and DSP operations.

III. PROPOSED SYSTEM

Reduced Instruction Set Computer (RISC) architectures represent an important innovation in the area of computer organization. This architecture attempt to produce more CPU power by simplifying the instruction set of the CPU. Fig.1 shows the Block Diagram of RISC system [4]. It includes Decoder, fetch machine, Arithmetic and logic machine, and register set.

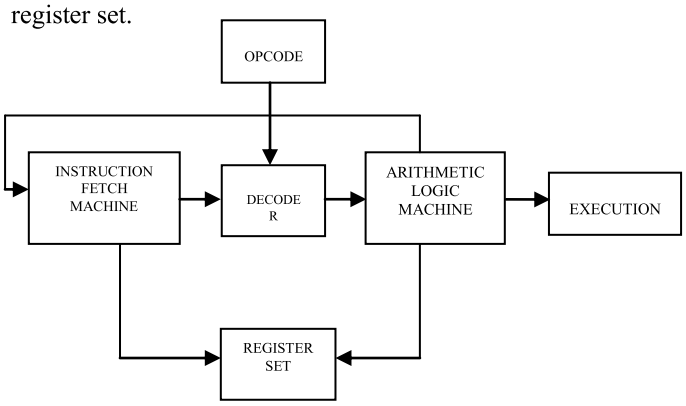

Fig. 1. Block Diagram of RISC System

The Register Set of this system contains the following registers:

- Instruction Register (IR) - holds the current instruction.
- Program Counter (PC) - holds the address of the next instruction.

Instruction Fetch Machine fetches an instruction from external memory, and upon completion of the instruction fetch cycle this machine signals the decoder to decode the instruction. Upon completion of the instruction fetch cycle, the instruction is decoded. The decoder reads bit 3 down to 0 of the Instruction Register, decides which of the sixteen operations the CPU needs to performs, and signals one of the next states to begin its operation. The ALU can perform Arithmetic and Logic operation based on the opcode obtained by decoding the instruction. The data is taken from two GPRs and is moved to the ALU. The result is stored in a GPR. For operations that involve one operand, a GPR can be specified to store the result. RISC system mainly consists of 2 parts shown in Fig.2.

- Control Unit

A Control Unit that coordinates the behavior of the Data Path by issuing appropriate control signals that guarantee the correct sequence of operations.It is typically designed as a single or cooperating FSMs.

- Data Path

A Data Path is a collection of interconnected modules that perform all the relevant computation on the data: it can use both combinational and sequential components. Data Path includes Registers, Memory, ALU etc. The Data Path is controlled by the control signals generated by the Control Unit.

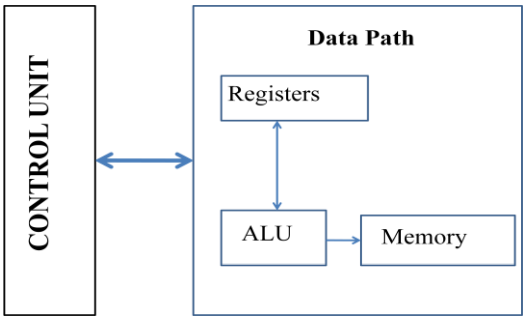

Fig.2. Control Unit and Data Path

IV. ARCHITECTURE OF RISC & DSP SYSTEM

Fig.3. shows the Architecture of RISC and DSP System which can perform Arithmetic, Logic and DSP operations. The RISC & DSP System mainly consist of 2 parts control unit and data path. The Data Path is controlled by the control signals generated by the Control Unit. A data path is a collection of functional units, such as Arithmetic Logic Units or multipliers that perform data processing operations. It is a central part of many central processing units (CPUs) along with the control unit, which largely regulates interaction between the data path and the data itself, usually stored in registers or main memory.

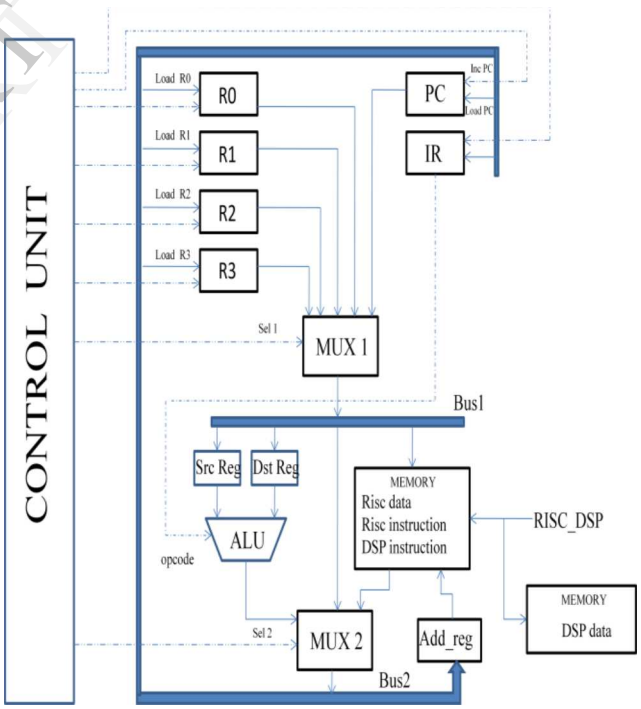

Fig.3. Architecture of RISC & DSP System

Working of RISC & DSP System is explained below:

1. Fetching the Instruction

The instruction is fetched from the memory using the address that is currently stored in the program counter (PC), and stored the instruction in the instruction register (IR). At the end of the fetch operation, the PC points to the next instruction that will be read at the next cycle.

2. Decode the instruction

The decoder interprets the instruction. During this cycle the instruction inside the IR (instruction register) gets decoded.

3. Execute the instruction

The control unit of the CPU passes the decoded information as a sequence of control signals to the relevant function units of the CPU to perform the actions required by the instruction such as reading values from registers, passing them to the ALU to perform mathematical or logic functions on them, and writing the result back to a register. If the ALU is involved, it sends a condition signal back to the Control Unit. Initially the Program Counter is at 00000000. During the Fetch machine cycle it fetches the instruction from the memory at 00000000 location. The instruction thus obtained is stored to Instruction Register after incrementing the Program Counter by 1. Then the Program Counter becomes 00000001. This is the fetch cycle. The instruction thus stored in the Instruction Register is decoded and opcode is obtained. This opcode is given to the ALU. On the basis of opcode the ALU can perform Arithmetic, Logic and DSP operations. If the opcode obtained after decoding the instruction denotes the DSP operation the input data is taken from DSP memory by enabling RISC_DSP signal.

V. INSTRUCTION FORMAT

The RISC machine fetches an instruction from the memory. Each instruction decodes by internal decoder and the value of each instruction is 8 bits. In those 0 to 3 bits is the opcode which decide the operation to be performed. Fig.4. shown the instruction format for RISC processor .4 to 5 bits represents the Source Register and 6 to 7 bits represents the Destination Register.

Destination Register			Source Register			Opcode		
7	2bits	6	5	2bits	4	3	4bits	0

Fig.4. Instruction Format

The Source Register and Destination Register is selected in such a way that,

IF **00 -> R0** is selected
IF **01 -> R1** is selected
IF **10 -> R2** is selected
IF **11 -> R3** is selected

The instruction format for output is shown in Fig.5.

OUTPUT	
7	0

Fig.5. Instruction Format for output

VI. INSTRUCTION SETS

This RISC & DSP System perform 16 operations. It includes 11 Arithmetic and Logic operations and 4 DSP operations. In this instruction set the processor use 4 bit opcode to allow instruction to perform various operations. Table.1 shows the instruction set for RISC & DSP system

Table.1 Instruction Sets for RISC & DSP System

Instruction	Opcode
OR	0000
AND	0001
NAND	0010
NOR	0011
XOR	0100
XNOR	0101
ADD	0110
SUBTRACT	0111
NOT	1000
INCREMENT	1001
DECREMENT	1010
FFT	1011
IFFT	1100
DCT	1101
IDCT	1110
READ	1111

VII. DSP OPERATIONS

The important DSP operations performed by this system are Fast Fourier Transform (FFT), InverseFFT, Discrete Cosine transform (DCT) and Inverse DCT.

- Fast Fourier Transform (FFT) & Inverse FFT

The Discrete Fourier Transform (DFT) plays an important role in the analyses, design and implementation of the discrete-time signal- processing algorithms and systems it is used to convert the samples in time domain to frequency domain. The Fast Fourier Transform (FFT) is simply a fast(computationally efficient) way to calculate the Discrete Fourier Transform (DFT).

$$X(K) = \sum_{n=0}^{N-1} x(n) W_N^{Kn} \quad K=0,1,....,N-1$$

The direct evaluation of *X(K)* requires N^2complex multiplications and N (N-1) complex additions.Inorder to avoid complexity Radix-2 DIT FFT Algorithm is used. The Flow graph of 8-point radix-2 DIT FFT algorithm is shown in Fig.6.

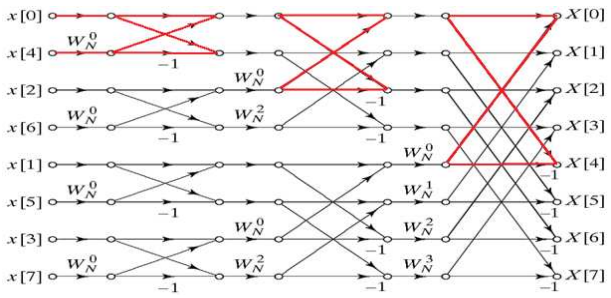

Fig.6. Flow graph of 8-point radix-2 DIT FFT algorithm

In a communication system that uses an FFT algorithm there is also a need for an IFFT algorithm to compute IDFT. The IDFT of an N-point sequence *X(K), K*=0,1,2,.....N-1 is defined as

$$x(n) = \frac{1}{N}\sum_{K=0}^{N-1} X(K) W_N^{-Kn} \quad , \quad n = 0,1,...,N-1$$

- Discrete Cosine Transform (DCT) & Inverse DCT

A discrete cosine transform (DCT) expresses a finite sequence of data points in terms of a sum of cosine functions oscillating at different frequencies.

The N point 1-D DCT is defined as:

$$X_K = \sqrt{\frac{2}{N}}\, C(k) \sum_{j=0}^{N-1} y_j \cos\left[\frac{(2j+1)k\pi}{2N}\right] \quad k = 0, 1, 2 \ldots N-1$$

The N point 1-D IDCT is defined as:

$$Y_j = \sqrt{\frac{2}{N}} \sum_{k=0}^{N-1} C(k)\, X_k \cos\left[\frac{(2j+1)k\pi}{2N}\right] \quad j = 0, 1, 2 \ldots N-1$$

where,

$$C(k) = \begin{cases} \frac{1}{\sqrt{2}} & \text{for } k=0 \\ 1 & \text{for } k \neq 0. \end{cases}$$

VIII. SIMULATION RESULTS

The design entry is modelled using VHDL in Xilinx ISE Design Suite 12.1 and the simulation of the design is performed using Modelsim SE 6.2c to verify the functionality of the design.

Table: 2 Instruction or Data in various addresses of RAM

Ram Address	Instruction or Data In That Address
00000000	00001111
00000001	10000001
00000010	01001111
00000011	11100001
00000100	10001111
00000101	00001001
00000110	11001111
00000111	11111111
00001000	00010000
00001001	10110001
00001010	01000010
00001011	11000011
00001100	11100001

- Simulation of RISC System

The simulation result of RISC system which can perform various Arithmetic and Logic operations based on the data stored in RAM (Table:2)..When reset='0' the program counter becomes '00000000' and all other signals are set to '0'.When reset='1' & rd_wb='1' the system starts to work by reading data from the RAM. The program counter (PC) locate the position '00000000' of RAM and taken out the instruction from that position. Here the instruction is '00001111'. The program counter is then incremented by 1 and the instruction is stored in the instruction register. The instruction stored in the instruction register is then decoded to obtain opcode. Thus the opcode obtained as a result of decode is '1111'.The opcode '1111' represent the next instruction in the memory is data and is taken out and load to corresponding register. After incrementing PC it becomes '00000001'.

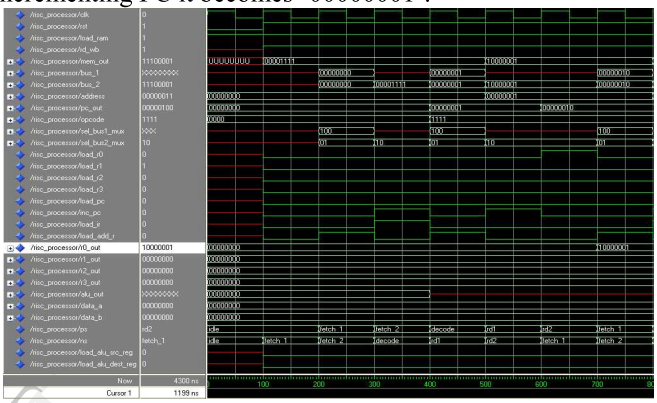

Fig.7. Simulation result of READ operation in RISC system.

Simulation result of various Arithmetic and Logic operations performed by the RISC system is shown Fig.8,9,10. Now the PC is '00001000'. PC locate the RAM position and taken out the instruction from that position. Here the instruction is '00010000'.The program counter is then incremented by 1 and the instruction is stored in the instruction register. The instruction stored in the instruction register is then decoded to obtain opcode.

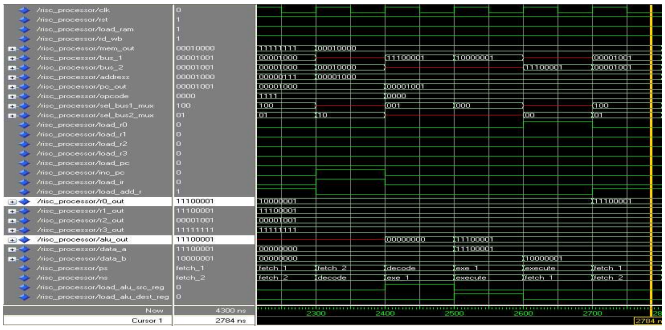

Fig.8. Simulation result of OR operation (opcode='0000') in RISC system.

Thus the opcode obtained as a result of decode is '0000'. The opcode '0000' represent the OR operation between two registers and the result is stored in the destination register. In the instruction'00010000',01 is for the selection of source register (R1) and '00' for the selection of destination register (R0).Here the OR operation takes place between R1 &R0 and the result is stored in the destination register (R0).The simulation result of OR operation is shown in Fig.8 and simulation results of AND, & ADD are shown in Fig. 9 & 10.

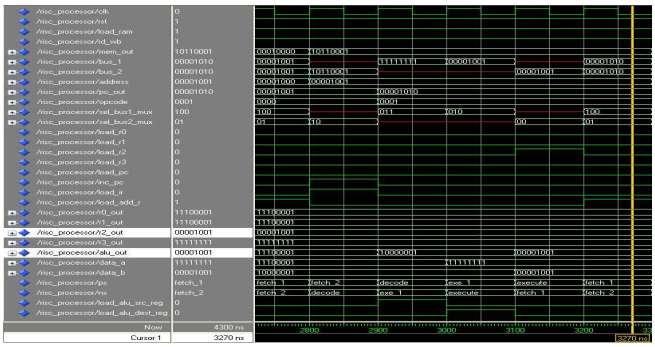

Fig.9.Simulation result of AND operation (opcode='0001') in RISC system

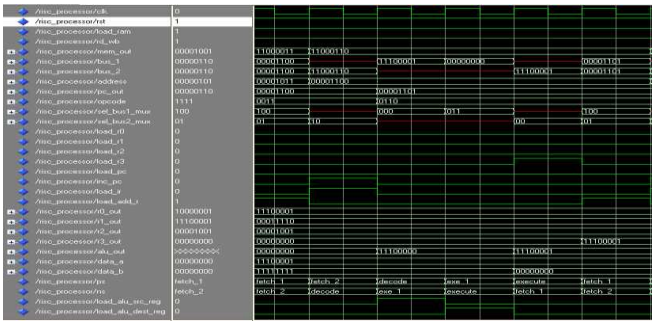

Fig.10.Simulation result of ADD operation (opcode='0110') in RISC system

- Simulation of 8-point FFT

The RISC & DSP System perform 4 DSP operations: FFT, IFFT, DCT & IDCT. Among the four DSP operations the 8 point FFT is designed and simulated using Decimation In Time Radix-2 Algorithm. For designing 8 point FFT, firstly designed 2-point FFT with the help of butterfly diagram. Then designed 4 point FFT using two 2-points FFT and so on. Simulation result of 8-Point FFT is shown in Fig.11.

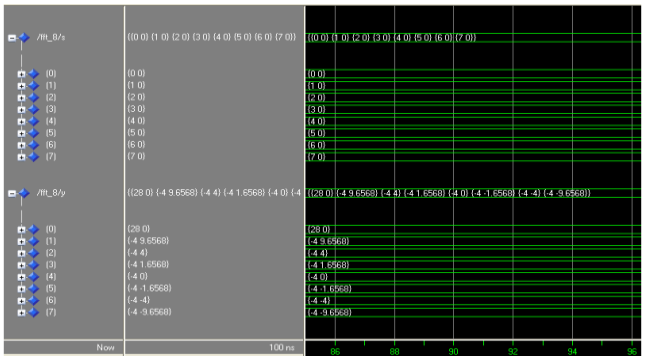

Fig.11. Simulation result of 8-point FFT

IX. CONCLUSIONS

This project intends to design 20 bit RISC & DSP system which can perform Arithmetic, Logic and DSP operations. The system uses a 4 bit opcode and 15 different operations (Arithmetic, Logic and DSP) can be done with these opcode. The various DSP operations performed by the system are FFT, IFFT, DCT and IDCT. The instruction is 20 bit out of which 0-3 bits represent an opcode which decide the operation to be performed, 4-11 and 12-19 bits represent the registers holding the values to be used for the instructions. The output is of 8 bit value. The submodules of the system are designed and simulated. By combining these submodules, designed and simulated various Arithmetic, Logic & DSP operations performed by the RISC system.

REFERENCES

[1] Deepak Kumar, K.Anusudha"Implementation of DSP System for Discrete Transforms using VHDL" *International Journal of Computer Applications* (42-45), Volume 69-No.26, May 2013.

[2] Asmita Haveliya, "Design and Simulation of 32-Point FFT Using Radix-2 Algorithm for FPGA Implementation", *Second International Conference on Advanced Computing & Communication Technologies*, 2012.

[3] Sneha N. Kherde, Meghana Hasamnis, "Efficient Design and Implementation of FFT", *International Journal of Engineering Science and Technology (IJEST),* ISSN : 0975-5462 NCICT Special Issue Feb 2011.

[4] Ryszard Gal, Adam Golda, Maciej Frankiewicz, Andrzej Kos, "FPGA implementation of 8-bit RISC Microcontroller for Embedded System" *MIXDES*, 323-328, 2011.

[5] M. R.S. Balpande, M.R.S. Keota, "Design of FPGA based Instruction Fetch & Decode Module of 32-bit RISC (MIPS) Processor," *Proc. ICCSNT*, p. 409, 2011.

[6] S. Belkouch, M. El Aakif and A. Ait Ouahman "Improved Implementation of a Modified Discrete Cosine Transform on Low-Cost FPGA" *IEEE 5th International Symposium on I/V Communications and Mobile Network*, Oct 2010, Rabat, Morocco.

[7] Zi-Wei Zheng and Zhe Ren "Efficient Design of Fast Fourior Transform Processor Using FPGA Technology" *IEEE International Conference on Electrical and Control Engineering*, pp.5195-5198, Aug 2010, Wuhan, China.

[8] LI Xiao-feng, Chen Long, Wang Shihu "The Implementation of High-speed FFT processor based on FPGA" *IEEE International Conference on Computer*, Mechatronics, Control & Electronic Engineering (CMCE), Vol.2 pp. 236-239, June 2010, Changchun, China.

[9] K. Anand and S. Gupta, "Designing Of Customized Digital Signal Processor" B.T. Thesis, Department of Electrical and Electronics, Indian Institute of Technology, Delhi, 2007.

[10] Jarrod D. Luker and Vinod B. Prasad "RISC System Design in an FPGA" *IEEE Conference Publication,* Vol. 2, pp. 532-536, Aug 2001, Dayton, USA.

[11] Raj Kamal "Architecture, Programming, Interfacing and System Design" *Pearson Education* Dorling Kindersley (India), 2007.