International Journal of Electronics Communication and Computer Engineering
Volume 5, Issue 6, ISSN (Online): 2249–071X, ISSN (Print): 2278–4209

RISC & DSP System Application Design using VHDL

Rachana Solanki
Email: rachana17solanki@gmail.com

Vinay Gupta
Email: vinaygupta.jnct@gmail.com

Abstract **– The Reduced Instruction Set Computer (RISC) processor use fewer instructions with simple constructs, therefore they can be executed much faster within the CPU without having to use memory as often. It reduce execution time by simplifying the instruction set of the computer. The DSP processors are perform the operation such as Discrete Cosine transform (DCT), Inverse DCT, Discrete Fourier Transform (DFT) and Fast Fourier Transform (FFT) are performed by DSP system. This paper represent the design of a Reduced Instruction Set Computer (RISC) and Digital Signal Processor (DSP) system described using VHDL with both the single-cycle and pipelined processors and implement in a Field Programmable Logic Array FPGA). These days most microprocessor and microcontroller designs are based on Reduced Instruction Set Computer (RISC) core and many operation such as Discrete Cosine transform (DCT), Inverse DCT, Discrete Fourier Transform (DFT) and Fast Fourier Transform (FFT) are performed by DSP system. The goal of this work is to incorporate Digital Signal Processor (DSP) system operation in RISC processor.**

Keywords **– RISC, DFT, DSP, VHDL, FFT.**

I. INTRODUCTION

Reduced Instruction Set Computer (RISC) are specific type of microprocessors, which recognizes limited number of instructions. One advantage of reduced instruction set computers is, they execute their instructions very fast because the instructions are so simple also its architecture requires less number of transistors which reduces its cist. The programing code is store in memory is decoded through number of stages including arithmetic logic unit, (ALU) and input/output port devices. The Arithmetic and logical operations as add, subtract, exclusive or, inclusive or, complement instructions are more than one cycle instructions but the branching instructions are single cycled instruction. This RISC controller uses Harvard architecture, i.e. program and data are accessed on separate buses as shown in fig.2.

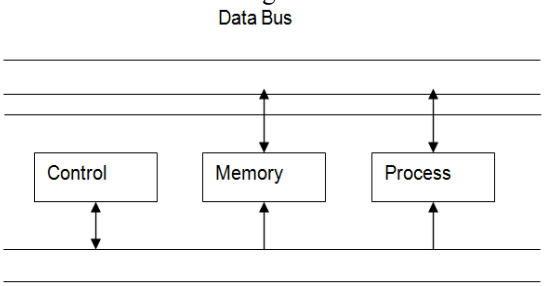

Fig.1. General Structure of Harvard Architecture

In Harvard architecture the address and data bus are separate so single clock cycle is require for its operation. A general block diagram of a 8-bit RISC micro controller is shown in Figure

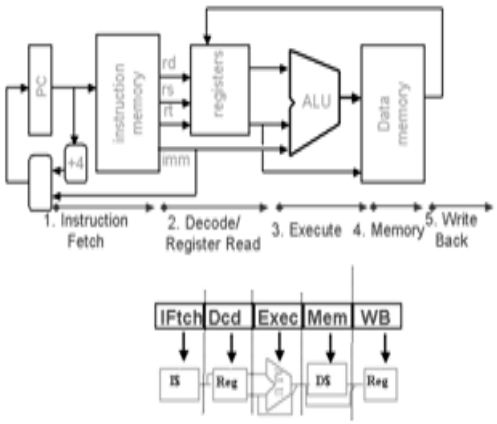

Fig.2. Block Diagram of 8 Bit RISC Microcontroller

The main features of DSP- Digital Signal Processor are-
1. Special arithmetic operations, such as Multiply and accumulates (MACs)
2. Perform DCT (Discrete Cosine transform) and IDCT (Inverse Discrete Cosine transform).
3. Perform DFT (Discrete Fourier Transform) and FFT (Fast Fourier Transform).
4. It can be implemented in general purpose computers or with embedded processors that may or may not consist of specialized microprocessors called digital signal processors.
5. Use VLIW (Very Large Instruction Word) techniques so every instruction supports multiple arithmetic units in parallel.

II. RISC ARCHITECTURE

The Architecture of RISC system is shown in Fig. 1. It includes Decoder, fetch machine, Arithmetic and logic machine, and register set.

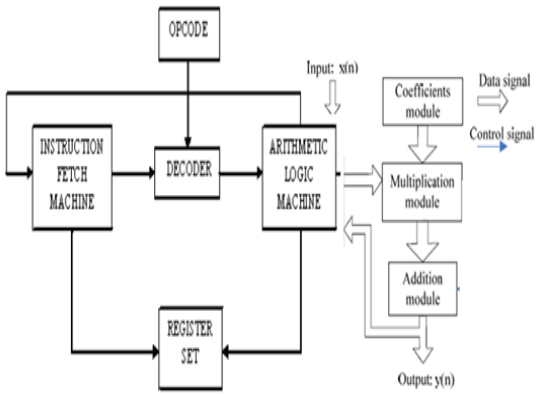

Fig.3. The RISC architecture

The MIPS pipelined processor involves five steps, the division of an instruction into five stages implies a five-stage pipeline:

International Journal of Electronics Communication and Computer Engineering
Volume 5, Issue 6, ISSN (Online): 2249–071X, ISSN (Print): 2278–4209

1. Instruction Fetch (IF): It fetches the instructions.
2. Instruction Decode (IDIt decodes instructions from instruction register.
3. Execution (EX): executing an operation or calculating an address
4. Data Memory (MEM): Access data from data memory.
5. Write Back (WB): writing the result into a register.

2.1 Execution

The instructions executes in RISC are in a pipelined way i.e when one instruction is executing; another is being fetched from memory so that the instruction fetch/execution cycle takes only one clock cycle. While branch instruction requires two cycles.

2.2 Data transfer instruction

The load and store instructions are executed for Data transfer operation to read or write the data to is destination memory or register rd. The mov instruction moves the data between the register within the CPU. It also place the ALU function control signal to cause the ALU to add its operand and allow the memory address register (MAR) input to accept the result. At the end of the clock cycle, when the effective address has been store in the memory address register, system disables the MAR input.

2.3 Arithmetic and Logical Instruction

The arithmetic and logical operations include add, subtract, multipy, logical OR, AND , NOT operations. The source operands were stored in the Ra, Rb register during the instruction decoding stage. According the opcode the controller select, which function the ALU should perform .It also enables the Rc register input to accept the ALU result. At the end of clock cycle, it disables the Ra, Rb register output and the Rc register input the result is written back to the destination register.

2.4 Control transfer instruction

These instructions like branch, Jump conditionally or unconditionally, wait etc transfer control within a program. The status of flag regiters play major role in control transfer instructions. At the end of clock cycle when the branch target address has been stored in the program counter, the program counter outputs and then it disable the program counter inputs. Some of the instructions are summarize as

Instruction	Operands	Description
Ib	rd,rs	Load byte
sb	rs,rd	Store byte
mov	rd,rs	Move source register to destination register
mov	rs,rd	Move destination register to source register
add,sub	rd,rs	Add/sub sign or unsigned
addi,subi	rd,rs	Add/sub sign or unsigned immediate
mult	fd,fs	Multiply unsigned
and,or xor	rd,rs	Bit wise logical and or xor
sll	rd,rs	Shift left logical
slli	rd,rs	Shift left logical Immediate
j	disp	Jump unconditional
jal		Jump conditional

2.5 Bus Read and Write

The signal "we" is use to write or read data from memory. When "we" =1 then the data is written to memory otherwise data read from the memory. After the rising edge of clock, the system place the address on memory address bus, sets the control signal to the required state and sets me to '1'.

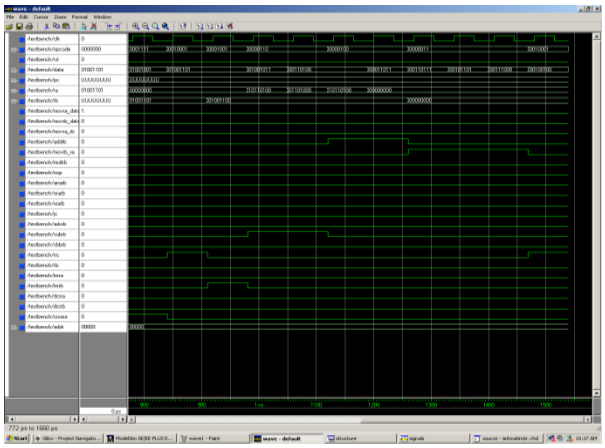

Fig.4. Trusted Instruction Set Operation

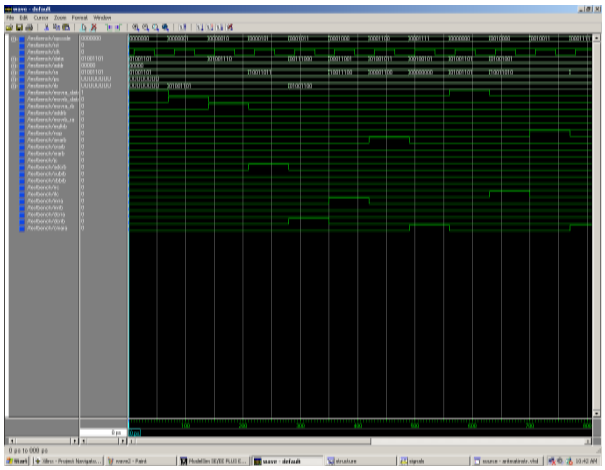

Fig.5. Trusted Instruction Set Operation

III. DISCRETE FOURIER TRANSFORM

It is a kind of Discrete Transform which is used in Fourier analysis. It transforms one function into another, which is called the frequency domain representation, or simply the DFT, of the original function. The formula for DFT is:

DFT:

$$X(k) = \sum_{n=0}^{N-1} x(n).e^{-j\left(\frac{2\pi}{N}\right)nk} \ (k = 0,1,\dots,N-1)$$

3.1 Fast Fourier Transform:-

FFT is an efficient algorithm or fast way to compute a DFT. Radix-2 Decimation-in-time (DIT) Fast Fourier Transform (FFT) is dividing the DFT in to two portions. The Basic Butterfly [1] operation of radix-2 DIT FFT algorithm of 8 signals is shown in Fig.6.

International Journal of Electronics Communication and Computer Engineering
Volume 5, Issue 6, ISSN (Online): 2249–071X, ISSN (Print): 2278–4209

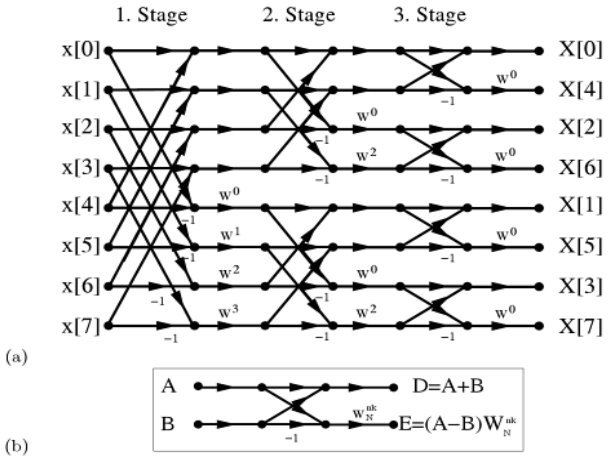

Fig.7. a) Signal Flow graph b) Butter fly algorithm for FFT.

The processor implementation, consists of two main types of logic elements: combinational and sequential elements. Combinational elements are elements that operate on data values, meaning that their outputs depend on the current status of inputs. Such elements in the processor implementation include the arithmetic logic unit (ALU) and memory. Sequential elements are elements that depends on current status of input and previous status of output i.e it acts as latch. Each state element has at least two inputs and one output. The two inputs are the data value to be written and a clock signal. The output signal provides the data values that were written in an earlier clock cycle. Logic units are design and implement in VHDL.

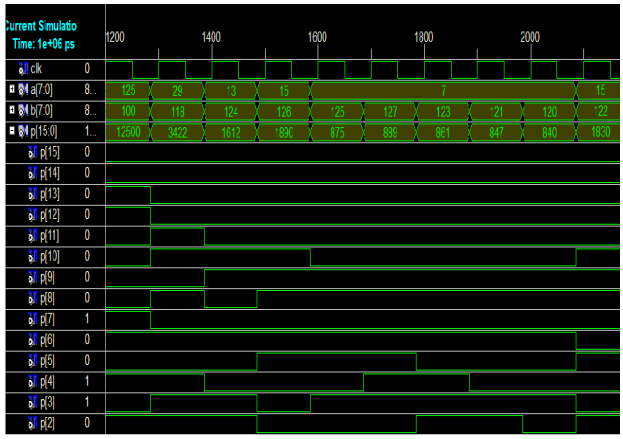

Fig.8. Timing simulation of multiplier unit.

IV. Instruction Fetch Unit

The instruction fetch unit fetches the instructions from memory whose address is specified by the content in program counter. Every time the PC increments its content for the next instruction. The instruction fetch component contains the following logic elements that are implemented in VHDL: 8-bit program counter (PC) register, an adder to increment the PC by four, the instruction memory, a multiplexor, and an AND gate used to select the value of the next PC.

4.1 Instruction Decode Unit:

The main function of the instruction decode unit is to use the 16-bit instruction given from the earlier instruction fetch unit to index the register file and obtain the register data values. The logic elements to be implemented in VHDL include several multiplexors and the register file.

4.2 The Control Unit:

The control unit of the single-cycle processor checks the instruction opcode bits and decodes the instruction to generate control signals. The Jump control signal selects the jump address to be sent to the PC. The Branch control signal is used to select the branch address to be sent to the PC. The Memory read control signal is asserted during a load instruction when the data memory is read to load a register with its memory contents. The Memory Write control signal is asserted when during a store instruction when a registers value is stored in the data memory

Fig.9. Timing simulation of control Unit

4.3 Execution Unit:

The execution unit of the processor contains the arithmetic logic unit (ALU) which performs the operation determined by the ALU control signal. The logic elements to be implemented in VHDL include a multiplexor, an adder, the ALU and the ALU control.

4.4 Pro-Program counter (Pro PC):

It is a one kind of clock pulse generator and address counter. It generates clock pulse for driving the Program counter (PC) from the main clock pulse. In this design it generates a pulse with one positive edge against four main clock pulses.

4.5 Control ROM.

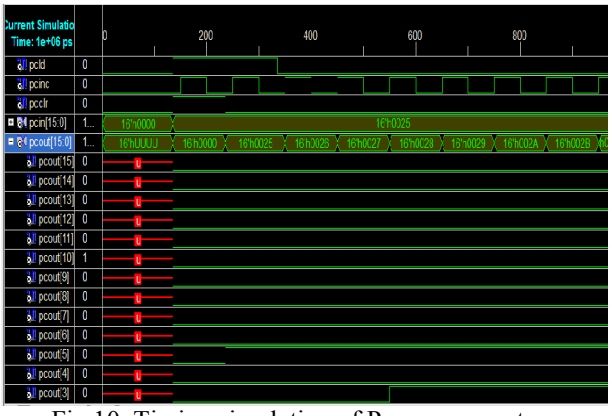

Fig.10. Timing simulation of Program counter.

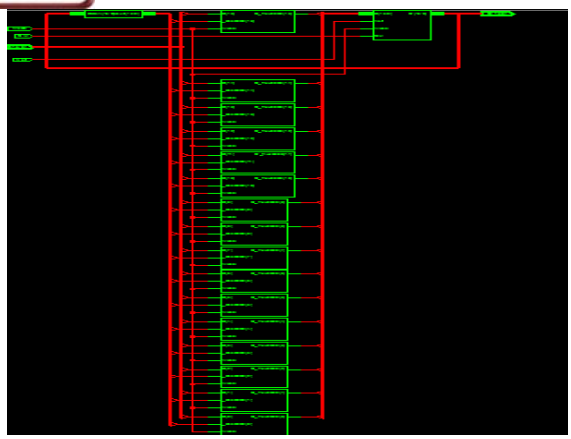

Fig.11. RTL view of program counter.

4.6 Data Memory Unit:

The data memory unit is only accessed by the load and store instructions. The load instruction asserts the MemRead signal and uses the ALU Result value as an address to index the data memory. The read output data is then subsequently written into the register file. A store instruction asserts the Memory Write signal and writes the data value previously read from a register into the computed memory address. The VHDL implementation of the data memory was described earlier. Figure shows the signals used by the memory unit to access the data memory. Appendix C contains the complete VHDL code used to create the memory state of the MIPS single-cycle processor.

REFERENCES

[1] Amit Kumar Singh Tomar, Rita Jain "20-Bit RISC & DSP System Design in an FPGA" Prceeding of ieee transaction sept 2013 1070-9924/$26.00 2013 IEEE.

[2] C. G. Penteado, E. D. Moreno" A Specialized Processor for Emulating Peripherals of the PIC Microcontroller" IEEE Latin America Transactions, Vol. 7, No. 2, June 2009 pp no. 133.

[3] Kris Myny,Erik van Veenendaal, Gerwin H. Gelinck, Jan Genoe, Wim Dehaene, and Paul Heremans "An 8-Bit, 40-Instructions-Per-Second Organic Microprocessor on Plastic Foil" IEEE Journal Of Solid-State Circuits, Vol. 47, No. 1, January 2012 pp no. 284.

[4] Raj Kamal “Architecture, Programming, Interfacing and System Design” Pearson Education Dorling Kindersley (India), 2007.