RESEARCH ARTICLE OPEN ACCESS

Verilog Implementation of 32-Bit CISC Processor

P.Kanaka Sirisha*,P.A.V.Sai Sindhuja**,P.Prasanthi**,K.Kiran****, K.Ankala Rao*****
*(Department of electronics and communication engineering,Bapatla engineering college, Bapatla-522101, **(Department of electronics and communication engineering,Bapatla engineering college, Bapatla-522101, ***(Department of electronics and communication engineering,Bapatla engineering college, Bapatla-522101, ****(Department of electronics and communication engineering,Bapatla engineering college, Bapatla-522101, *****(Department of electronics and communication engineering,Bapatla engineering college, Bapatla-522101,

ABSTRACT
The Project deals with the design of the 32-Bit CISC Processor and modeling of its components using Verilog language. The Entire Processor uses 32-Bit bus to deal with all the registers and the memories. This Processor implements various arithmetic, logical, Data Transfer operations etc., using variable length instructions, which is the core property of the CISC Architecture. The Processor also supports various addressing modes to perform a 32-Bit instruction. Our Processor uses Harvard Architecture (i.e., to have a separate program *and* data memory) and hence has different buses to negotiate with the Program Memory and Data Memory individually. This feature enhances the speed of our processor. Hence it has two different Program Counters to point to the memory locations of the Program Memory and Data Memory.Our processor has 'Instruction Queuing' which enables it to save the time needed to fetch the instruction and hence increases the speed of operation. 'Interrupt Service Routine' is provided in our Processor to make it address the Interrupts.

***Keywords* :** PC-Program counter,DM-Data memory,S_BUS-system bus,ACC-Acumulator,MUX-Multiplexer,DEMUX-Demultiplexer,IR-Instruction register

I. INTRODUCTION

From the architecture point of view, the microprocessor chips can be classified into two categories: Complex Instruction Set Computers (CISC) and Reduce Instruction Set Computers (RISC). In either case, the objective is to improve system performance.

'CISC' stands for 'Complex Instruction Set Computer'. CISC Computers are based on a complex Instruction set in which instructions are executed by microcode. Microcode allows developers to change hardware designs and still maintain backward compatibility with instructions for earlier computers by changing only the microcode, thus make a complex instruction set possible and flexible. Although CISC designs allow a lot of hardware flexibility, the supporting of microcode slows microprocessor performance because of the number of operations that must be performed to execute each CISC instruction. A CISC instruction set typically includes many instructions with different sizes and execution cycles, which makes CISC instructions impossible to pipeline.

Properties:-
CISC processor has most of the following properties:-

- Richer instruction set, some simple, some very complex
- Instructions generally take more than 1 clock cycle to execute
- Instructions of a variable size
- Instructions interface with memory in multiple mechanisms with complex addressing modes
- No pipelining
- Upward compatibility within a family
- Microcode control
- Work well with simpler compiler

II. HARVARD ARCHITECTURE:

The harvard architecture is a computer architecture with physically separate storage and signal path ways instruction and data.The term originated from the harvard mark 1 relay-based computer which stored instructions on punched tape(24bit wide)and data in electromechanical counters.These early machines had data storage entirely contained within the central processing unit,and provided no access to the instruction storage as data.programs needed to be loaded by an operator;the processor could not initialise itself.

Today most processors impleement such separate signal pathways for performance reasons ,but actually implement a modified harvard architecture,so they can support tasks like loading a program from disk storage as data and then executing it.**FIGURES**

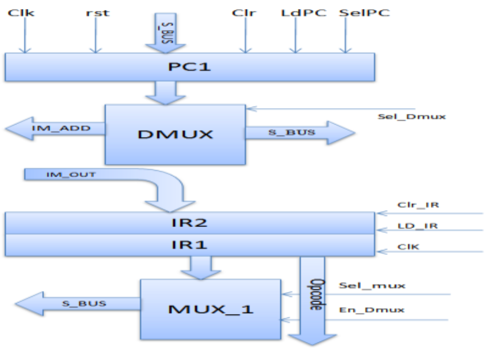

Figure 1:-Instruction Memory Fetching Unit

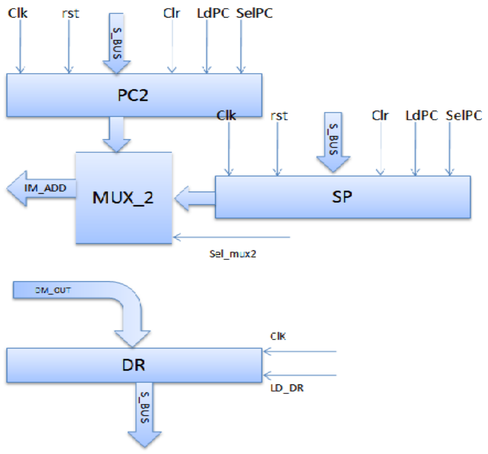

Figure 2:- Data Memory Fetching Unit

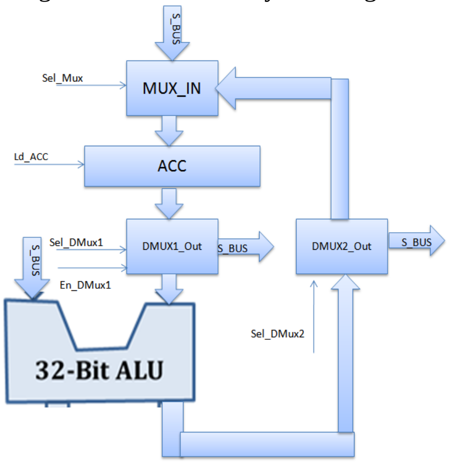

Figure 3:- Arithmetic and Logical Unit

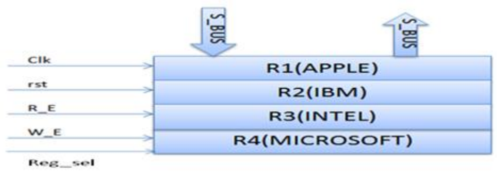

Figure 4:Register file

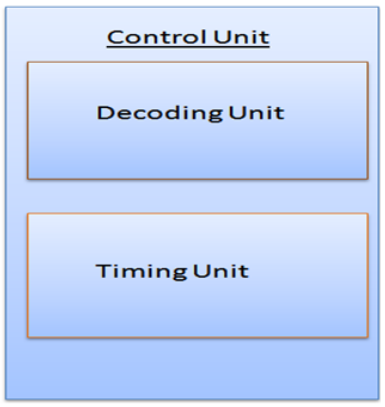

Figure 5:- Control Unit

Fig 1:
At First, the address in the 'Program Counter 1 (PC1)' is either loaded directly from the system bus or incremented from its previous value. The Memory address in the 'PC' is loaded out. Based on the selection pin of the 'DeMUX', the memory address is loaded either into the 'Instruction Memory address (IM_ADD) bus' (in the general case) or to the 'System Bus' (in the case of subroutine). The data in the 'Instruction Memory' pointed by the address is fetched to the 'Instruction Queue (IQ)' by the 'Instruction Memory OUT (IM_OUT) Bus'.

The 'IQ' is then loaded by selecting the 'Load' Pin. The data in the 'IR1' is then loaded into the MUX and the output to the bus is then again decided by the 'Enable MUX' and 'Select MUX' signals. The output to the 'System Bus'is either the entire data in IR1 (in the case of 32-bit memory location) or zero-padded 24-bit (in the case of immediate value). The MSB 8 bits of IR1 are loaded as input 'Opcode' to the control Unit (CU).

Fig 2:
The System Bus is connected to the 'Program Counter 2 (PC2)', 'Stack Pointer (SP)' and directly to the 'Data Memory (DM)'. Based on the respective enable signals, the data in the System Bus enters into PC2 (in the case of loading the PC2 with DM address) or SP (in the case of subroutine) or directly to the DM (in the case of writing to the DM).Hence in any case, the MUX2 present there, provides the address of DM that arrives from either PC2 or SP. In the case of write operation, the address is derived from the PC2 and data to be written is present in the System Bus that is directly connected to the DM. In the case of 'Read' operation, the data in the DM pointed by the address from the PC2 is loaded in to the Data Memory out (DM_OUT) Bus. On the enabling of 'Load' signal of the Data Register (DR), the data in the DM_OUT bus enters into the DR. The data in

the DR is always readily available to the system bus that connects various blocks of the processor.
Fig 3:
As the architecture indicates, the output result of the ALU is again stored in the Accumulator (Acc) itself or transferred to the System Bus based on the Selection pin of the DeMUX2. The MUX present selects the input to the Acc, either the data in the System Bus or the ALU output and the data enters 'Acc' only when the 'Load Acc' is enabled. A DeMUX1 is then present in order to share the data in the Acc to either the input 'A' of the ALU or to the System Bus as the output.The input 'B' to the ALU is also provided by the System Bus itself. When the input enters as 'B' from the System Bus, the Input 'A' doesn't change since the 'Load Acc' is disabled. Hence the operation is performed based on the ALU Opcode provided by the Control Unit, and the result of the Operation is stored again in the 'Acc' itself. The data in the 'Acc' can be used by moving it into the System Bus by selecting the appropriate DeMUX signal.
Fig 4:
The Register File (RF) of the CISC32 Processor contains 4 General Purpose Registers each called Apple (R1), IBM (R2), INTEL (R3) and MICROSOFT (R4). The System Bus attached to the RF is used by the processor to both read and write the RF. Operations 'Read' and 'Write' depend upon the 'Read Enable (R_E)' and 'Write Enable (W_E)' Signals attached to the Register File.
Fig 5:
The control Unit (CU) block consists of two units named Decoding Unit (DU) and Timing Unit (TU). The opcode that comes out of the FU1 acts as the input to the CU. The CU then enables the necessary signals in a sequential manner to perform the operation indicated by the opcode. 'Heart' to the Human Body is what 'CU' to the program.

III. CISC INSTRUCTION FORMAT

1. Opcode:- It occupies 8 bits out of 32 bits, which can accommodate a maximum of 2^8 (256) instructions.
2. The processor has 4 user accessible registers and can be addressed by 2 bits, hence in case of 'register to register' addressing; the entire operation can be installed in a single instruction.
3. In case 'memory memory' or'memory to register' operations 2 to words

IV. EXECUTION CYCLE

FETCH INSTRUCTION: -At First, the instruction will be fetched from the program memory through the bus allocated to it. The instruction is then stored in an instruction queue and there on to be passed to the Control Unit.

Decode: -The instruction will be decoded in the decode unit of control unit and the necessary enable and disable signals will be sent to the necessary blocks (Micro programmed Control Unit) using the timing unit. The operands, if present, are placed on to the Internal Data Bus for the use of the Registers or ALU.

Fetch Operands: -If the opcode requests fetching of operands from data memory, the memory location will be placed on to the internal bus so that the location will be copied on to the Data Memory PC and then to the bus attached to the Data Memory. The Operand will be fetched and placed on the Internal Data Bus making it available for the ALU or the Register Bank as in the previous case.

Execute: -The processor then performs the operation indicated by the opcode.

Store Output Operand: -It then stores the result in the Accumulator (By Default; if it is an ALU Operation). Thus the 'Execution Cycle' is completed.

V. CONCLUSIONS

- The CISC32 Processor is implemented using Verilog Language.
- The Processor is designed using Harvard Architecture and hence uses two Different Program Counters are used in CISC32 Processor to point the Program Memory and Data Memory Individually.
- The speed of the processor is 1 MHz
- The Accumulator Type Instruction Set Architecture (ISA) is used in the CISC32 Processor (Where the result is stored in Acc itself).
- Up to 256 different instructions are compatible.
- 3 types of addressing modes are used in the processor.
- 5 General Purpose Registers are used (including Accumulator).
- Instruction Queuing is also made available which slightly increases the speed of the processor Operations.

REFERENCES

[1]. Computer Organization and Architecture' by Prof. J. K. Deka, IIT- Guwahati
[2]. Computer Organization and Architecture' by William Stallings

[3]. Hasan Krad and Aws Yousif Al-Taie, ‘A New Trend for CISC and RISC Architectures’, IEEE.
[4]. A journal on 8 Bit RISC processor using verilog HDL