

Design of 8-bit microprocessor using Verilog (SAP-1 architecture)

Computer Architecture & Design Lab Assignment

Submitted by

Abhishek Dutta – 061bct501

Bibek Shrestha – 061bct509

Date - May 22, 2007

Title

Design a 8-bit microprocessor using Verilog and verify it's operations. Use SAP-1 (Simple As Possible) architecture as your reference.

Introduction

The Simple-As-Possible (SAP)-1 computer is a very basic model of a microprocessor explained by Albert Paul Malvino¹. The SAP-1 design contains the basic necessities for a functional Microprocessor. Its primary purpose is to develop a basic understanding of how a microprocessor works, interacts with memory and other parts of the system like input and output. The instruction set is very limited and is simple.

The features in SAP-1 computer are

- W bus - A single 8 bit bus for address and data transfer.
- 16 Bytes memory (RAM)
- Registers are accumulator and B-register each of 8 bits.
- Program counter – initializes from 00H(0d) to FFH(15d) during program execution.
- Memory Address Register (MAR) to store memory addresses.
- Adder/Subtractor for addition and subtraction instructions.
- A Control Unit
- A Simple Output.
- 6 machine reserved for each instruction

The instruction format of SAP-1 Computer is

(XXXX) (XXXX)

the first four bits make the opcode while the last four bits make the operand(address).

1 Albert Paul Malvino, Ph. D. introduced the concept of SAP (Simple As Possible) computers in his book “Digital Computer Electronics – An Introduction to Microcomputers” - 2nd Edition.

SAP-1 instruction set consists of following instructions

Mnemonic	Operation	OPCODE
LDA	Load addressed memory contents into accumulator	0000
ADD	Add addressed memory contents to accumulator	0001
SUB	Subtract addressed memory contents from accumulator	0010
OUT	Load accumulator data into output register	1110
HLT	Stop processing	1111

Example: if 0000 1000 is stored at memory location 0000 of RAM then SAP1 computer interprets it as follows:

Machine cycle and Instruction cycle

SAP1 has six T-states (three fetch and three execute cycles) reserved for each instruction. Not all instructions require all the six T-states for execution. The unused T-state is marked as No Operation (NOP) cycle.

Each T-state is called a machine cycle for SAP1. A ring counter is used to generate a T-state at every falling edge of clock pulse. The ring counter output is reset after the 6th T-state.

FETCH CYCLE – T1, T2, T3 machine cycle

EXECUTE CYCLE - T4, T5, T6 machine cycle

Architecture

1. Program Counter (PC)

- implemented in “pc.v” file
- It counts from 0000 to 1111 and it signals the memory address of next instruction to be fetched and executed

2. Input and MAR (MAR)

- implemented in “inputMAR.v” file
- During a computer run, the address in PC is latched into Memory Address Register (MAR).

3. RAM

- implemented in “mem16k.v” file
- the program code to be executed and data for SAP1 computer is stored here.
- During a computer run, the RAM receives 4-bit addresses from MAR and a read operation is performed. Hence, the instruction or data word stored in RAM is placed on the W bus for use by some other part of the computer.
- It is asynchronous RAM, which means that the output data is available as soon as valid address and control signal are applied

4. Instruction Register (IR)

- implemented in “ir.v” file
- IR contains the instruction (composed of OPCODE+ADDRESS) to be executed by SAP1 computer.

5. Controller- Sequencer

- implemented in “cu.v” file
- it generates the control signals for each block so that actions occur in desired sequence. CLK signal is used to synchronize the overall operation of the SAP1 computer.
- A 12 bit word comes out of the Controller-Sequencer block. This control word determines how the registers will react to the next positive CLK edge.

6. Accumulator

- implemented in “accumulator.v” file
- it is a 8 bit buffer register that stores intermediate results during a computer run.
- It is always one of the operands of ADD,SUB and OUT instructions.

7. Adder-Subtractor

- implemented in “addersubtractor.v” file
- it is a 2's complement adder-subtractor
- this module is asynchronous (unclocked), which means that its contents can change as soon as the input words change

8. B Register

- implemented in “register.v” file
- it is 8 bit buffer register which is primarily used to hold the other operand (one operand is always accumulator) of mathematical operations.

9. Output Register

- this registers hold the output of OUT instruction.

10. Binary Display

- it is a row of eight LEDs to show the contents of output register.

Discussion

The design of modules like Program Counter, Input and MAR, RAM, Instruction Register, Accumulator, Adder-Subtractor, B register, Output register was easy as similar designs were already done in Lab assignments.

The most complicated part was the design of Controller/Sequencer(C/S). The first problem we faced arose due to failure to understand about the settling times for data lines. We tried to check the data lines at the instant it changed and hence we got the old value of the data line. This problem was resolved by adding a 1ms delay (#1;). The second problem we faced in designing C/S was “Maximum HDL code reached”. We deleted the code for test bench of other modules to decrease the size of our project. This was probably the limitation of beta version of Silos workbench (Silos2001.120) which we used in this project for coding and testing of Verilog code for SAP1 computer.

The task of synchronizing each block was also very complicated. The presence of a shared bus (WBUS) also caused several complications during the design.

Conclusion

We became familiar with the working of a 8-bit microprocessor and the way each module contributes to the overall functioning of the computer. Although the concept of SAP1 is very simple, the knowledge gained while designing it can be extended to design of more complex microprocessors.

We successfully designed a 8-bit microprocessor based on SAP1 architecture and verified it's operations in Verilog.

References

1. Albert Paul Malvino. *Digital Computer Electronics – An Introduction to Microcomputers: SAP-1(Page 140)*. Tata McGraw-Hill Publishing Company Limited 1990
2. Peter M. Nyasulu. *Introduction to Verilog* Jan. 2005