

**Ahsanullah University of Science & Technology**

**Department of Computer Science & Engineering**

**Course No : CSE3110**

**Course Title : Digital System Design Lab**

**Assignment Name : Implementation of SAP computer using Proteus**

**Date of Submission : 21.04.2019**

**Submitted To : Ms. Mahzabeen Emu**

**Ms. Nudrat Nawal**

**Submitted By-**

**Group : 05**

**Id : 16.02.04.024**

**Section : A1**

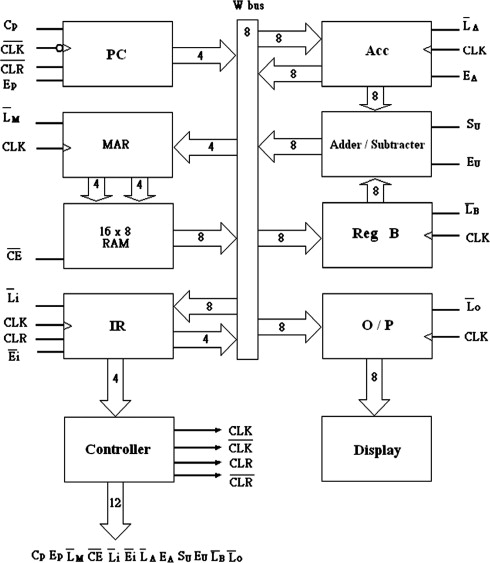
**Introduction:**

SAP or Simple-As-Possible computer is the first stage towards the evolution of modern computers. It gives us the most basic and necessary ideas about the basic building blocks of a computer without immersing us with the more complicated details.

**Usage of SAP Computer:**

The main use of SAP computer is to perform addition and subtraction operations on 8-bit data. 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 simple.

**Diagram of SAP-1 Architecture:**



**Components:**

SAP computer has mainly the following components:

**Program counter (PC):**

It’s a 4 bit register that counts from 0 to 15 and acts like a pointer register. It generates and holds the addresses of the instructions to be executed next. It contains 4 signals:

* Cp: Used to increment the value in the PC
* Ep: Used to place the value in the PC onto the W bus
* CLK’: Negative edge triggered clock pulse
* CLR’: Used to clear the PC to 0

**Input and MAR (Memory Address Register):**

The address of the current instruction to be executed is passed from PC to MAR through W bus and consequently to the RAM. It can take 4 bit input too. It contains 2 signals:

* LM’: Used to load data in the W bus onto the MAR
* CLK: Positive edge triggered clock pulse

**16x8 RAM (Random Access Memory):**

It contains 16 memory locations, each of 8-bit length. It is programmed beforehand to store data and instructions before the SAP computer is actually run. It takes 8-bit input. It contains 1 signal:

* CE’: Used to pass the data in the RAM onto the W bus

**Instruction register (IR):**

It stores the instruction placed on the W Bus on a positive clock trigger and divides it into two nibbles. The upper nibble goes to the Controller-Sequencer and the lower nibble can be placed onto the W bus when needed. It contains 4 signals:

* LI’: Used to load the 8-bit data on the W bus onto the IR
* CLK: Positive edge triggered clock pulse
* CLR: Used to clear the IR to 0
* EI’: Used to pass the data in the IR onto the W bus

**Accumulator:**

It holds the value of a number when addition or subtraction is to be performed on two numbers. It also holds the result from the adder/subtractor after an operation is performed. Its value is passed to the output register. It contains 3 signals:

* LA’: Used to load the 8-bit data on the W bus onto the accumulator
* CLK: Positive edge triggered clock pulse
* EA: Used to pass the data in the accumulator onto the W bus

**Adder/Subtractor:**

It performs addition or subtraction on two 8-bit numbers placed in accumulator and register B respectively, depending on the signal SU. Subtraction is done in 2’s complement method. As it’s asynchronous, its contents changes as soon as the input changes in accumulator or register B. It contains 2 signals:

* SU: Used to control what type of operation is performed on the data in the accumulator and register B. If SU=0, addition is performed. If SU=1, subtraction is performed.
* EU: Used to pass the data in the adder/subtractor onto the W bus

**Register B:**

It holds the second 8-bit number on which addition/subtraction operation is performed. It contains 2 signals:

* LB’: Used to load the 8-bit data on the W bus onto register B
* CLK: Positive edge triggered clock pulse

**Output Register:**

It loads the 8-bit result of addition/subtraction operation performed in adder/subtractor that was initially placed in the accumulator. This is then passed to the binary display. It contains 2 signals:

* LO’: Used to load the 8-bit data on the W bus onto output register
* CLK: Positive edge triggered clock pulse

**Binary Display:**

It is a row of 8 light emitting diodes (LEDs) that glow depending on the value in the output register. The value that was passed to the output register from the accumulator is what makes the LEDs glow.

**Controller-Sequencer:**

There are 12 signals coming out of the controller-sequencer, which forms the **control word**. It consists of the following 12 signals:

Cp, Ep, LM’, CE’, LI’, EI’, LA’, EA, SU, EU, LB’, LO’.

These are what controls the whole SAP computer. Without these, no operations could be performed. Before each operation is performed, a clear signal resets the whole computer.

**Directions of use:**

At first, we have to include the binary image file to the RAM in order to load the instructions we want to make the SAP computer perform. To turn on the SAP computer in proteus, we have to put the CLR’ toggle to 1. It will clear the contents of the memory and consequently the SAP computer will perform each instruction loaded in the RAM serially due to the clock pulse it receives through a digital clock generator. CLK signal is generated by using the HLT signal and the clock pulse generator.

**Proteus implementation:**

The implementation of the various components of SAP computer in proteus are described below:

**Program Counter**

To construct the 4-bit PC, we have used four 1-bit JK flipflops **(IC: 74LS107).** We have input the control signal Cp to all the J and K input ports of the flipflops as the purpose of the Cp is to increment the PC. We included the CLR’ signal to all the R(reset) ports and CLK signal was provided to only the LSB bit of the 4-bit counter. The other 3 flipflops use their predecessor bits’ outputs as clock pulse. All the outputs are connected to a seven segment BCD as well as 4 logic probes in order to display the current value in the PC. Tristate buffers **(IC: 74LS126)** were used in order to control the flow of data between PC and W bus. They only allow flow of data to the W bus (last 4 bits) when Ep is activated.

**Input and MAR**

For this we used one 4-bit D registers **(IC: 74LS173)** whichreceives the last 4 bits of the W bus as input. The control signal LM’ is connected to the E1 and E2 ports to control when the data from the W bus is loaded. CLR and CLK’ are connected as well to the respective ports. OE1’ and OE2’ are grounded.

**16x8 RAM**

For this, we used a 32 kilobit EPROM **(IC: 2732).** It receives the 4-bit output from MAR as input in the first 4 bits(A0-A3) and the rest 8-bits and port CE’ are grounded. Control signal CE’ is connected to the OE’/VPP port to control when the contents in the RAM are put onto the W bus. The 8-bit output is connected to the W bus.

**Instruction Register**

For this we used two 4-bit D registers **(IC: 74LS173)** as IR is 8-bit in length. The 8-bit input is connected to the W bus. The control signal LI’ is connected to the E1 and E2 ports to control when the data from the W bus is loaded. CLR and CLK’ are connected as well to the respective ports. OE1’ and OE2’ are grounded for the register used for the higher nibble, whereas for the lower nibble, signal EI’ is connected to control when data is transferred to the bus. The output for the upper nibble goes to the controller-sequencer while the lower nibble’s output goes back to the bus.

**Controller-Sequencer**

It consists of basic gates such as AND, OR and NOT. Six 1-bit JK flipflops **(IC: 74LS107)** were used to construct a ring counter to control the T-states. The 4-bit output received from the IR was used to form the mechanism for the 5 opcodes (LDA, ADD, SUB, OUT, HLT). The 12 control signals were formed using the T-states and basic gates. Logic probes were used to see the state of the control word.

**Accumulator**

For this we used two 4-bit D registers **(IC: 74LS173)** as accumulator is 8-bit in length. The 8-bit input is connected to the W bus. The control signal LA’ is connected to the E1 and E2 ports to control when the data from the W bus is loaded. CLR and CLK’ are connected as well to the respective ports. OE1’ and OE2’ are grounded for both registers. The 8-bit output is connected to the input A(A7-A0) of the adder/subtractor directly as well as to the W bus through tristate buffers **(IC: 74LS126)** which are controlled by the signal EA.

**Adder/Subtractor**

We used two 4-bit adders **(IC: 74LS83)** to construct this. The input for A(A7-A0) comes from the accumulator whereas the input for B(B7-B0) comes from register B. Signal SU is in XOR connection with the input of B to control when addition and subtraction occurs. The 8-bit output is connected to the W bus through tristate buffers **(IC: 74LS126)** which are controlled by the signal EU.

**Register B**

For this we used two 4-bit D registers **(IC: 74LS173)** as register B is 8-bit in length. The 8-bit input is connected to the W bus. The control signal LB’ is connected to the E1 and E2 ports to control when the data from the W bus is loaded. CLR and CLK’ are connected as well to the respective ports. OE1’ and OE2’ are grounded for both registers. The 8-bit output is connected to the input B(B7-B0) of the adder/subtractor directly.

**Output Register**

For this we used two 4-bit D registers **(IC: 74LS173)** as output register is 8-bit in length. The 8-bit input is connected to the W bus. The control signal LO’ is connected to the E1 and E2 ports to control when the data from the W bus is loaded. CLR and CLK’ are connected as well to the respective ports. OE1’ and OE2’ are grounded for both registers. The 8-bit output is connected to logic probes to see the required output of the SAP.

**Binary Display**

8 logic probes were used to see the output in binary format.