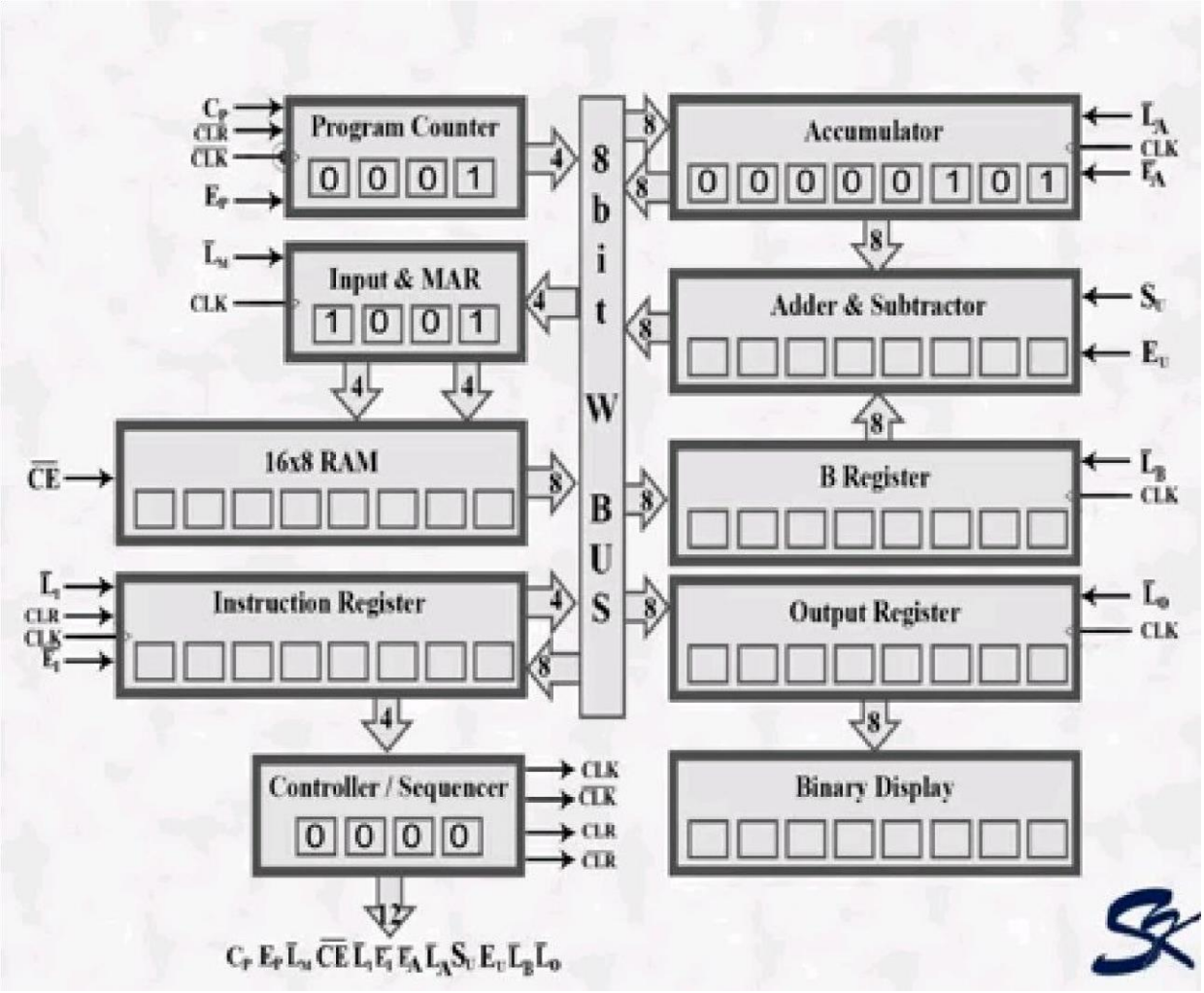
SAP-1(Simple as Possible-1)

Architecture Introduction

Introduction:

The simple as possible(SAp-1) computer is a very basic model of a microprocessor explained by Albert Paul Makvino.

The sap-1 design contains the basic necessities for a functional Microprocessor. Its primary purpose is to develop a basic understanding of how microprocessor works, interact with memory and other parts of the system like input and output. The instruction set is very limited and simple



The features in Sap-1 computer are:

W bus - A single 8 bit bus for address and data transfer.16 bytes memory **RAM(16 X 8).**Register counter - initializes from 00H(od) to FFH(15d) during program execution–Memory address Register(MAR) to store memory addresses. Adder/sub for addition and subtraction instruction. A control unit .A simple output 6 machine reserved for each instruction

SAp-1 instruction set contains following instruction

1: Mnemonic operation OPCODE.

2: LDA load addressed memory contents into accumulator 0000

3: ADD add addresses memory contents to accumulator 0001

4: SUB subtract addressed memory contents from accumulator 0010

5: OUT load accumulator data into Register 1100

6: HLT strop processing 1111

Example: if 0000 1000 is stored at memory location 0000 of the RAM then SAp-1

Computer interprets it as follows:

Machine cycle and instruction cycle

SAP-1 has six T-states (three fetch and three Execute cycle) reserved for each instruction. Not all instruction requires all the six T-states for excution. The unused T-states is marked as No operation (NOP) cycle. Each T-states is called a machine cycle for SAP1. A ring counters used to generate a T-states at every falling edge of clock pulse. The ring counter output is reset after the 6th T-sates

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 counter from 0000 to 1111 and its signals the memroy address of the next instruction to be fetched and executed

2: INPUT AND MAR (MAR)

***** implemented in "inputMar.v" file

* During computer run, the address in pc is latched into the memory address register (MAR)

3: RAM

* implemented in "mem16k.v" file

* The program code to be executed and data for SP-1 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 computer

* It is asynchronous RAM, which means that's 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 SEQUANCE

* implemented in "cu.v" file

* It generated the control signals for each block so that's action occurred in desired sequance.CLK is used to synchronous the overall operation of the SAP1

* A 12 bit word comes out of the controller-sequence block. This control word determines how the Register will react to the next positive CLK edge.

6: Accumulator

***** implemented in "accumulator.v" file

* It is an 8 bit buffer register that's stored intermediate results during a computer run.

* It is always one of the operand of add, sub, and out instruction.

8: REGISTER

***** implemented in "registed.v" file

* It is 8 bit buffer register which is primarily used to hold the other operands (one operands is always accumulator) of mathematical operations.

9: OUTPUT REGISTER

***** This register hold the output of out instruction

10: BINARY DISPLAY

***** It is a row of eight LED'S to the show the contents of output register