SAP-1

Simple As Possible Computers

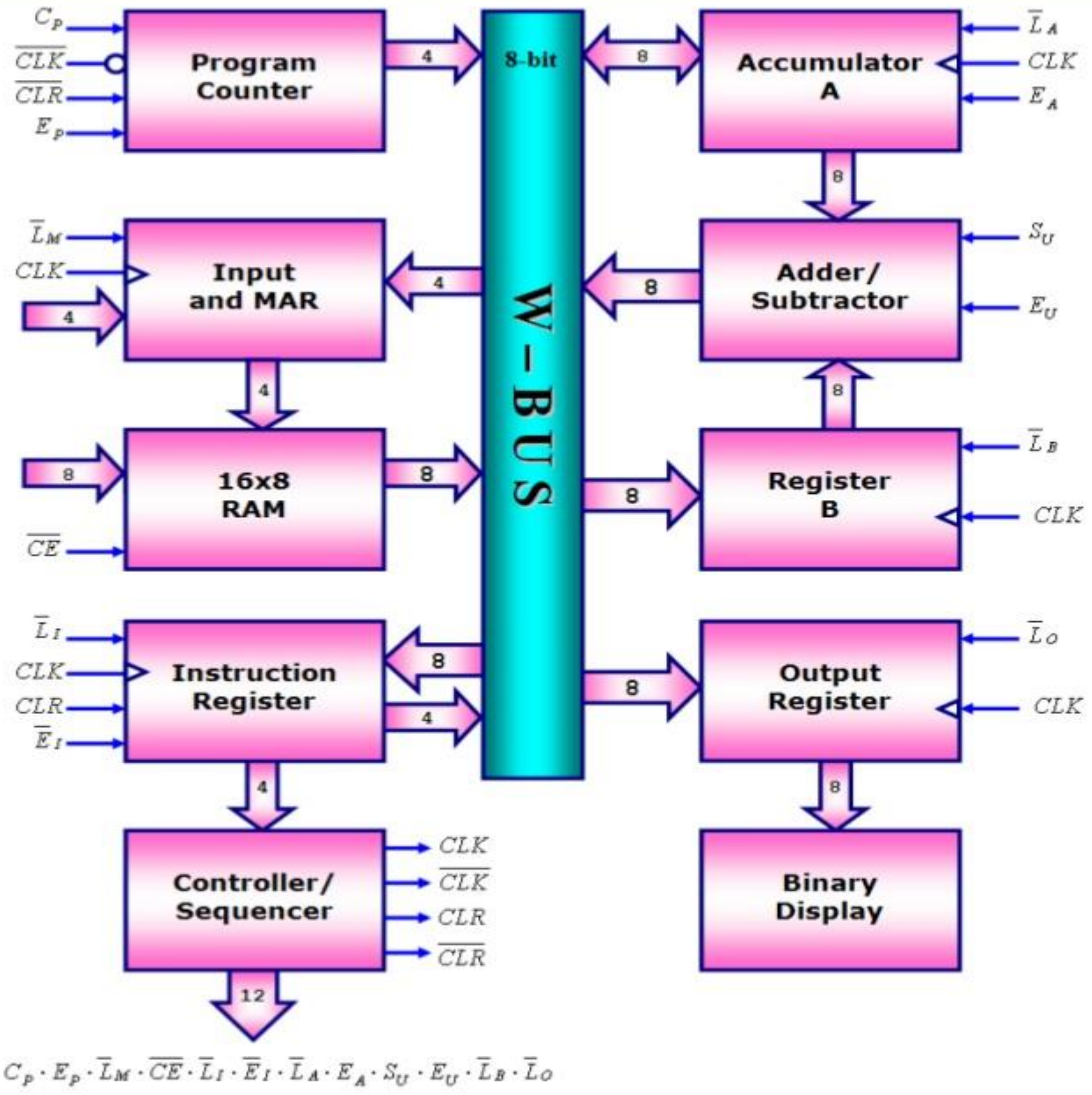
SAP-1 Introduction

- SAP-1 is the first stage in the evolution towards modern computers.
- The main purpose of SAP is to introduce all the crucial ideas behind computer operations.
- Being a simple computer, SAP-1 also covers many advanced concepts.
- SAP-1 is a bus organized computer. All registers are connected to the W bus with the help of tri-state buffers.

10-1 Architecture

SAP-1

Block Diagram



10-1 Architecture

PROGRAM COUNTER

✓Program counter is a part of the control unit of SAP and counts from 0000 to 1111.

✓The program is stored at the beginning of the memory with the first instruction at binary address 0000, second instruction at 0001 and so on.

✓Its job is to send to the memory the address of the next instruction to be fetched and executed.

✓The PC is reset to 0000 before each computer run.

✓When computer run begins, PC sends address 0000 to the memory.

✓The PC is then incremented to get 0001.

✓After the first instruction is fetched and executed, the PC send address 0001 to memory and PC is incremented to 0010….

✓In this way program counter is keeping track of the next instruction to be fetched and executed

10-1 Architecture

INPUT AND MEMORY ADDRESS REGISTER (MAR)

✓MAR is a part of SAP-1 memory.

✓During a computer run, the address in the PC is latched into MAR. A bit later, the MAR applies this 4 bit address to the RAM, where a read operation is performed.

THE RAM

✓The RAM is a 16 X 8 static TTL RAM.

✓During a computer run, the RAM receives 4-bit addresses from the MAR and a read operation is performed.

✓In this way, the instruction or data word stored in the RAM is placed on W-bus for use in other parts of the computer

10-1 Architecture

INSTRUCTION REGISTER (IR)

✓IR is a part of the control unit.

✓To fetch an instruction from the memory, the computer does a memory read operation.

✓This places the contents of the addressed memory location on the W bus.

✓At the same time, the instruction register is set up for loading on the next positive clock edge.

✓The contents of the IR are split into 2 nibbles. The upper nibble goes to the controller sequencer and the lower nibble is read onto the W-bus when needed.

10-1 Architecture

CONTROLLER SEQUENCER

✓Before each computer run, a CLR’ signal is sent to the PC and a CLR signal is sent to IR. This resets the PC to 0000 and wipes out the last instruction in the IR.

✓A clock signal CLK is sent to all the buffer registers (MAR, Accumulator, B Register and Output Register)

✓A CLK’ signal also goes to PC.

✓The 12 bits that come out of the controller-sequencer form a word controlling the rest of the computer. The 12 wires carrying the control word are called the control bus.

✓Control word has the format:

$$\mathbf{C_P E_P L_M' CE' \quad L_1' E_1' L_A' E_A \quad S_U E_U L_B' L_O'}$$

10-1 Architecture

ACCUMULATOR (A)

✓Accumulator is a buffer register that stores intermediate answers during computer run.

✓The accumulator has two outputs. One output goes directly to the adder-subtractor. The other output goes to the W-bus when E_A is high.

10-1 Architecture

THE ADDER-SUBTRACTOR

SAP-1 uses a 2’s complement adder-subtractor. When S_U is low, the sum out of the adder-subtractor is

S = A + B

When S_U is high, the difference appears

S = A + B’+1

(Recall that 2’s complement is equivalent to a decimal sign change)

The adder-subtractor is asynchronous (unclocked). This means that the contents can change as soon as the input word changes. However, these contents appear on the W bus only when E_U is high.

10-1 Architecture

B REGISTER

✓B Register is another buffer register, used in arithmetic operations.

✓A low L_B’ and CLK loads the word on W bus into B Register. This is fed into the adder-subtractor to be added or subtracted

OUTPUT REGISTER

✓At the end of a computer run, the accumulator contains the answer. At this point, we need to transfer the answer to the outside world. This is where the output register is used. When E_A is high and L_O’ is low, the next positive clock edge loads the accumulator word into the output register.

BINARY DISPLAY

✓Row of 8 LEDs connected to the output register/port

10-2 Instruction Set

LDA

✓LDA stands for “load the accumulator”.

✓A complete LDA instruction includes the hexadecimal address of the data to be loaded.

✓LDA 8H means – load the accumulator with the contents of memory location 8H

If, R8 = 1111 0000

The execution of LDA 8H results in A = 1111 0000

10-2 Instruction Set

ADD

✓ADD is another SAP-1 instruction

✓A complete ADD instruction includes the hexadecimal address of the word to be added.

✓ADD 9H means – add the contents of the memory location 9H to the accumulator contents. The sum replaces the original contents of the accumulator.

✓Suppose A = 0000 0010 and R9 = 0000 0011

During execution of ADD 9H, R9 is loaded in B register to get, B = 0000 0011

And almost instantly the adder-subtractor forms the sum of A and B and loads output onto Accumulator A to give A = 0000 0101

10-2 Instruction Set

SUB

✓SUB is another SAP-1 instruction

✓A complete SUB instruction includes the hexadecimal address of the word to be subtracted.

✓SUB CH means – subtract the contents of the memory location CH from the accumulator contents. The difference replaces the original contents of the accumulator.

✓Suppose A = 0000 0111 and RC = 0000 0011

During execution of SUB CH, RC is loaded in B register to get, B = 0000 0011

And almost instantly the adder-subtractor forms the difference of A and B and loads output onto Accumulator A to give A = 0000 0100

10-2 Instruction Set

OUT

✓The instruction OUT tells the SAP-1 computer to transfer accumulator contents to the output port/register.

✓OUT is complete by itself since the instruction does not involve data in memory

HLT

✓HLT stands for halt

✓HLT marks the end of a program

✓HLT is complete by itself since the instruction does not involve data in memory

10-2 Instruction Set

Memory-Reference Instructions

✓LDA, ADD and SUB are called memory-reference instructions because they use data stored in memory.

✓OUT and HLT are not memory-reference instructions

Mnemonics

✓LDA, ADD ,SUB, OUT and HLT are the instruction set for SAP-1

✓These are called mnemonics

10-3 Programming SAP-1

MNEMONIC	OP CODE
LDA	0000
ADD	0001
SUB	0010
OUT	1110
HLT	1111

The Op-code goes to the upper nibble and the operand goes to the lower nibble.
LDA FH = 0000 1111
ADD EH = 0001 1110
HLT = 1111 XXXX

Assembly language involves working with mnemonics when writing a program
Machine language involves working with 0s and 1s