Project Description

Objective:

The objective of the project is to implement an 8-bit computer based on the architecture of SAP-1 (Simple-As-Possible) using combinational and sequential circuits.

Project Specification:

The computer must have the following features and functionalities-

1. A single 8-bit bus for address and data transfer.
2. 16x8 bit random access memory (RAM) for storing program and data.
3. A program counter for containing the address of the next instruction (should be capable of counting from 0000 to 1111).
4. Input unit for writing the program into RAM before running the computer.
5. Memory address register (MAR) for accessing RAM data.
6. An arithmetic and logic unit (ALU) capable of 8-bit addition and subtraction. Two registers Accumulator (register A) and register B (both 8-bit) are also needed for storing data for ALU operation.
7. A control unit for controlling all other blocks (registers, memory, ALU etc.)
8. Output unit capable of showing the output in binary form.
9. Must have the ability to perform following instruction set-

Operation	Description
Load	Load data from memory to the accumulator
Add	Add data from memory to value in the accumulator
Subtract	Subtract data from memory from value in the accumulator
Output	Load data from accumulator to the output register
Halt	Stop processing

10. Main clock signal driving the computer should have two modes of operation- (i) Auto and (ii) Manual.
11. The computer must have all the blocks shown in project diagram (Fig. 1). Specifications of each block are described later.

Complete diagram of the project is shown in the next page.

Project Diagram:

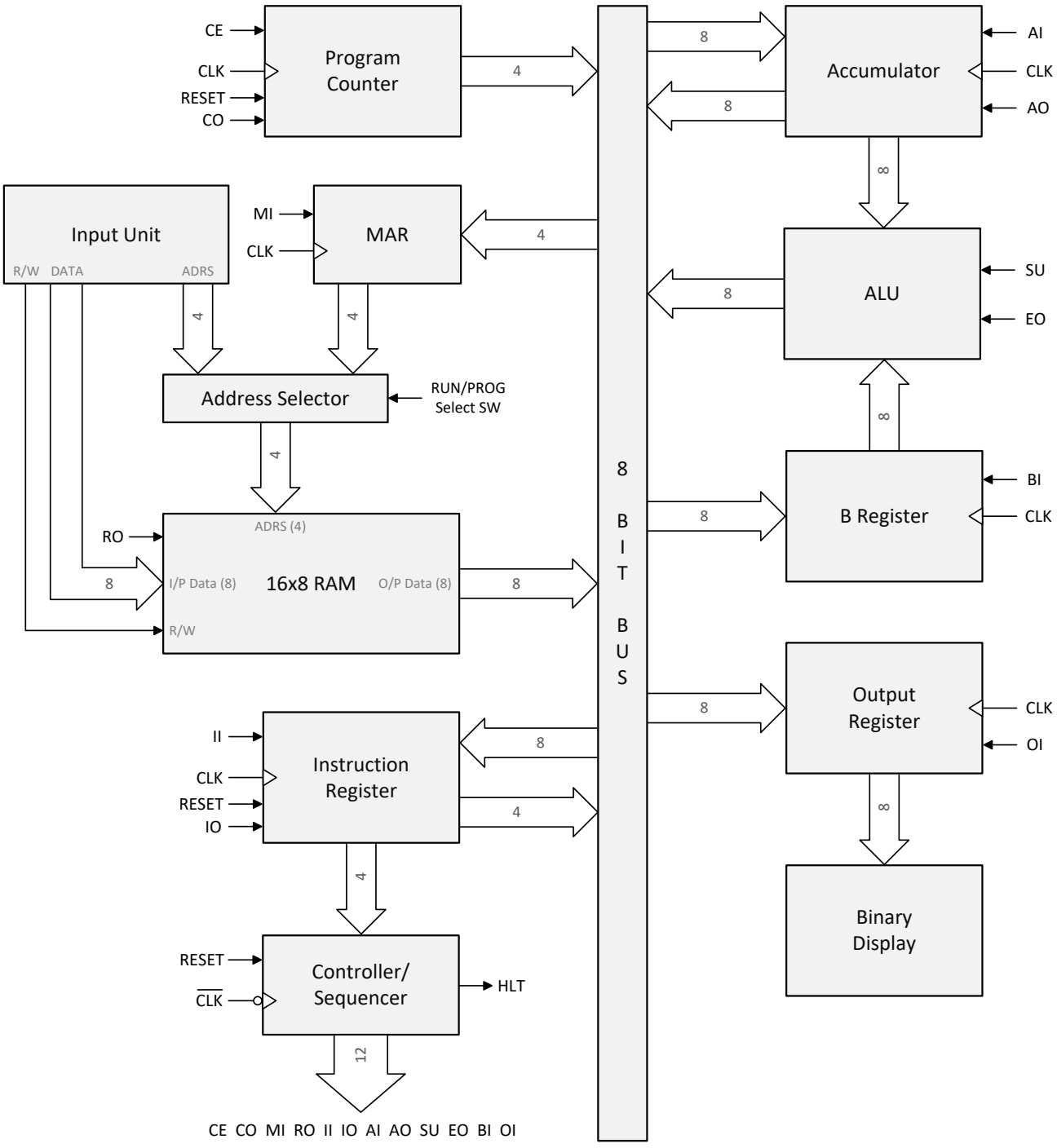


Fig. 1: Block diagram of the project.

Submodule/Block Specification:

This section discusses the specifications/requirements for each block/submodule shown in the project diagram (Fig. 1) -

8-bit Bus:

1. Bus must have 8 lines for address and data transfer.
2. It should be designed in such a way, so that at a single time instance only one module will send data to the bus, and only one module will receive that data.
3. Two modules can never send data to the bus at the same time (bus contention). When a module is sending data to the bus, all other modules' outputs must be disconnected.

Program Counter:

Program counter (PC) is a 4-bit counter which can count from 0000 up to 1111. Program counter's job is to store and send out the memory address of the next instruction to be fetched and executed. It will increment its value after each instruction cycle (completion of an instruction). Functions of its I/O pins are as follows-

Pin	Type	Function
CE	In	Program Counter Enable: Increments the program counter on the next clock-cycle.
CO	In	Program Counter Out: Puts the current program counter value on the bus. Output is disconnected when CO is low.
RESET	In	Resets the program counter to 0000
CLK	In	Clock pulse input.
OUT	Out	4-bit counter output.

Input Unit and Memory Address Register (MAR):

Memory address register (MAR) is a 4-bit register. MAR stores the 4-bit address of data or instruction which are placed in RAM. When the computer is in the RUN state, the 4-bit address is obtained via the bus from the Program Counter and then stored. This stored address is sent to the RAM where data or instructions are read from. Functions of its I/O pins are as follows-

Pin	Type	Function
MI	In	Memory Address In: Stores the current values of the bus into the Memory address register (MAR).
CLK	In	Clock pulse input.
OUT	Out	4-bit address data.

Input unit is used when computer is needed to be programmed by writing data into the RAM (PROG state). There should be a selector to choose between MAR output (4-bit) and Input address output (4-bit) depending on the state (RUN/PROG). Output of selector should be

connected to address pins of RAM. The following is a description of switches that input unit should contain-

Switches	Description
Input data switches	8 switches (D7-D0) for writing 8-bit data to specific address.
Address switches	4 switches (A3-A0) for accessing a particular address of RAM for writing.
R/W switch	For switching between read and write mode.

Random Access Memory (RAM):

RAM size should be 16 x 8 bit (16 memory locations each storing 8 bits of data). The RAM will be programmed by means of the address switches and data switches of input unit. During a computer run, the RAM will receive its 4-bit address from the MAR.

Pin (Bit)	Type	Function
ADRS (4)	In	4-bit address input for accessing RAM locations.
I/P Data (8)	In	In PROG state, used for writing 8-bit data to addressed location.
R/W	In	Read or write mode.
CS	In	Chip select pin.
RO	In	RAM Out: Put the currently selected RAM byte onto the bus. Output is disconnected when RO is low.
O/P Data (8)	Out	8-bit output data.

Accumulator, B Register and Output Register:

All of these registers will be of 8-bit size. Accumulator (also called as A register) should have the ability to output intermediate results after each ALU operation. Accumulator and B register will be used to store 8-bit data as input to ALU input. The purpose of output register is to store result to display when requested. Controller/sequencer unit sends control signals to these registers for controlling input/output enable. Description of these control bits-

Pin	Name	Function
AI	Accumulator In	Stores the current values of the bus into Accumulator.
AO	Accumulator Out	Accumulator sends its stored content to the bus. Output of accumulator is disconnected when AO is low.
BI	B Register In	Stores the current values of the bus into B Register.
OI	Output Register In	Stores the current values of the bus into Output Register.

Arithmetic and Logic Unit (ALU):

ALU should be capable of 8-bit addition and subtraction. For addition and subtraction, 2's complement approach should be used. Description of two control bits of ALU-

Pin	Name	Function
SU	Addition/Subtraction Select	When SU=0, addition operation, when SU=1, subtraction operation will be executed.
EO	ALU Out	Puts the ALU output onto the bus. Output of ALU should be disconnected when RO is low.

Instruction Register:

Function of instruction register is to receive and store the 8-bit instruction placed on the bus from the RAM. The contents of the instruction register are then split into two nibbles (nibble means 4-bit). The upper nibble goes into the controller-sequencer while the lower nibble should be sent to the bus. A reset pin is required for resetting the instruction register, when computer starts. Control bits of instruction register are-

Pin	Name	Function
II	Instruction Register In	Writes the current values of the bus into the instruction register.
IO	Instruction Register Out	Sends the stored data (only lower nibble) onto the bus. Output pins are disconnected when AO is low.

Controller-Sequencer:

Controller-sequencer generates necessary control signals for each block so that actions occur in a desired sequence. 12 control bits come out of the controller-sequencer block. These control bits determine how all other blocks will react to the next positive CLK edge. So, controller-sequencer must be designed in such a way so that correct control bits are already available to all the blocks before the next positive CLK edge. 12 control bits are –

CON = CE CO MI RO II IO AI AO SU EO BI OI

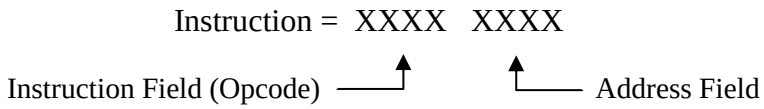
Functions of all 12 control bits are already described. Besides these 12 signals, controller-sequencer also generates a HLT (halt) signal, which can halt the process of computer by stopping the main clock.

Main Clock Signal:

Clock signal should be generated using 555 timer IC. The clock module needs to have two modes of operation available- (i) Auto and (ii) Manual. In manual mode, pressing a switch/push button will advance each clock cycle.

Specification of Instruction Set:

An instruction in the SAP-1 architecture consists of 1 byte. The upper nibble of this byte (bits 7-4) is called "opcode" and it defines the instruction. The lower nibble (bits 3-0) can be used to pass parameters to the instruction (for example a RAM Address).



The computer must be able to perform the following five instructions-

Instruction	Mnemonic	Opcode	Description
Load Accumulator	LDA	0000	Load the value stored in a particular RAM address which is specified by the address field of the instruction, then store it in the accumulator.
Add	ADD	0001	Add the value stored in a particular RAM address to the value in the accumulator. RAM address is given by address field of instruction.
Subtract	SUB	0010	Subtract value stored in a particular RAM address from the value in the accumulator. RAM address is given by address field of instruction.
Output	OUT	0011	Load data from the accumulator and send that to the output register.
Halt	HLT	1111	Stop processing.

Instruction Cycle:

After each instruction is executed, program counter points to the next instruction stored in RAM. But each instruction consists of several smaller instructions called **microinstruction**. The time required by the CPU to execute one single instruction is called **instruction cycle**. For SAP-1, length of this instruction cycle is 6 time states (T states), where **T state** is the time required to complete one microinstruction.

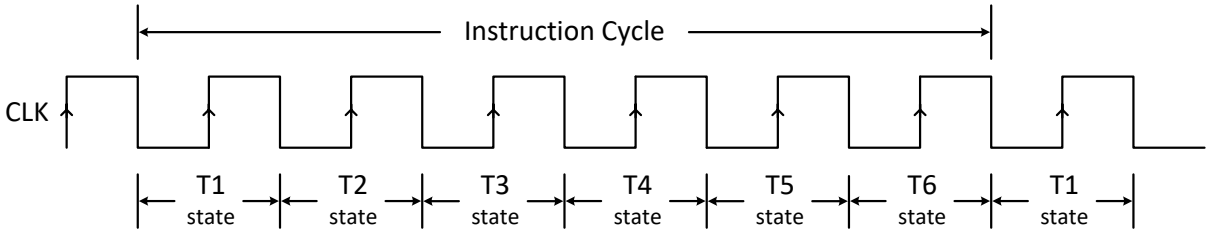


Fig. 2: T1-T6 states in SAP-1 computer.

At the beginning of a T state, controller-sequencer modifies the value of 12 control bits depending on the microinstruction. During that T state, all control bits are kept unchanged. Then, at the start of next T state, control bits are again modified based on the requirement of next microinstruction.

One important thing to observe, each T state starts at the negative edge of the clock pulse (fig. 2), where all the submodules/blocks of the computer are triggered at the positive edge of clock pulse. This is required, so that correct control bits are already available to all the submodules/blocks before the submodules trigger.

Instruction cycle can be further divided into two parts-

i. Fetch cycle (First 3 T states: T1, T2, T3)
ii. Execution cycle (Last 3 T states: T4, T5, T6)

Fetch Cycle:

Fetch cycle (T1, T2, T3) is responsible for fetching the instruction from memory. Fetch cycle is exactly same for all instructions. So, register transfers (or microinstructions) during T1-T3 states will be always same for SAP-1 for any instruction.

T State	Name	Activated Control Bits	Function
T1	Address State	CO MI	Address stored in the program counter (PC) is transferred to the memory address register (MAR).
T2	Increment State	CE	Program counter (PC) is incremented.
T3	Memory State	RO II	Instruction stored at the addressed location of RAM is transferred to instruction register.

Execution Cycle:

During execution cycle (T4, T5, T6) fetched instruction gets executed. Microinstructions during these three states can vary depending upon the instruction that was fetched. Execution cycle for each instruction (LDA, ADD, SUB, OUT, HLT) are as follows-

LDA Routine

T State	Activated Control Bits	Function
T4	IO MI	Upper nibble of instruction register output (IR) goes to controller-sequencer. Lower nibble is loaded into MAR.
T5	AI RO	Addressed data of RAM gets loaded into the accumulator.
T6		No operation.

ADD Routine

T State	Activated Control Bits	Function
T4	IO MI	Upper nibble of instruction register output (IR) goes to controller-sequencer. Lower nibble is loaded into MAR.
T5	RO BI	Addressed data of RAM gets loaded into B register.
T6	EO AI	ALU output is loaded into accumulator.

SUB Routine

T State	Activated Control Bits	Function
T4	IO MI	Upper nibble of instruction register output (IR) goes to controller-sequencer. Lower nibble is loaded into MAR.
T5	RO BI	Addressed data of RAM gets loaded into B register.
T6	EO AI SU	SU bit sets up the ALU to do subtraction. Then subtraction output of ALU gets loaded into the accumulator.

OUT Routine

T State	Activated Control Bits	Function
T4	AO OI	Content of accumulator is stored in output register.
T5		No operation.
T6		No operation.

HLT Routine

HLT does not require any control routine because no registers are involved in execution of halt instruction. After HLT instruction is fetched, controller-sequencer activates the HLT bit, which stops the main clock.