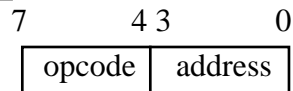


## SOME NOTES ON THE SAP-1

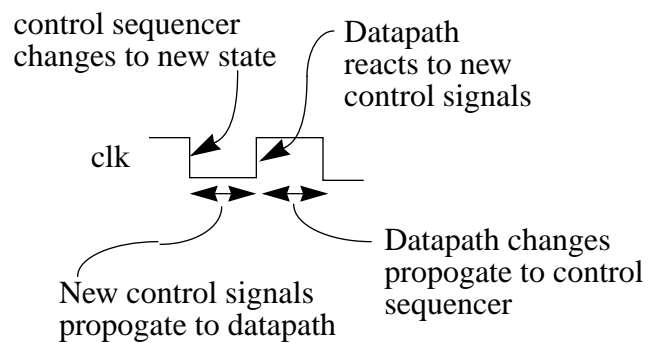
SAP-1 Instruction Format:



SAP-1 Instruction Set:

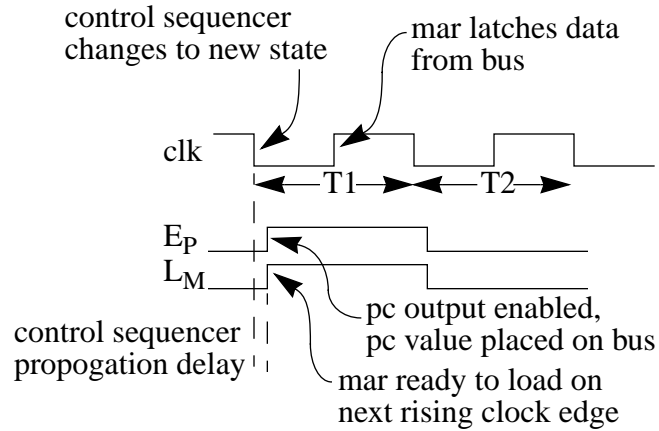
Instruction	opcode	operation
lda	0000	Load the data in M[address] into the accumulator register.
add	0001	Add the value stored in M[address] to the value in the accumulator.
sub	0010	Subtract the value stored in M[address] from the value in the accumulator.
out	1110	Copy the data in the accumulator to the output register
hlt	1111	Stop instruction execution.

Control Sequencer is Negative-Edge Triggered:

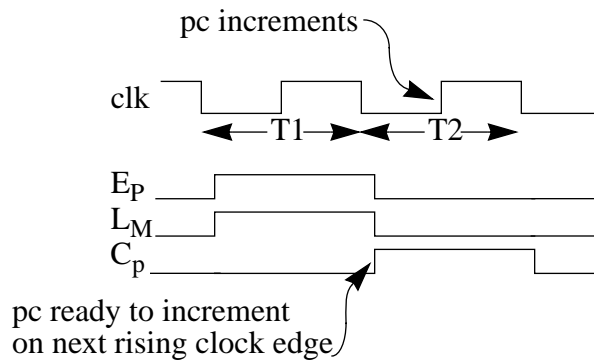


## SAP-1 INSTRUCTION FETCH CLOCK CYCLES

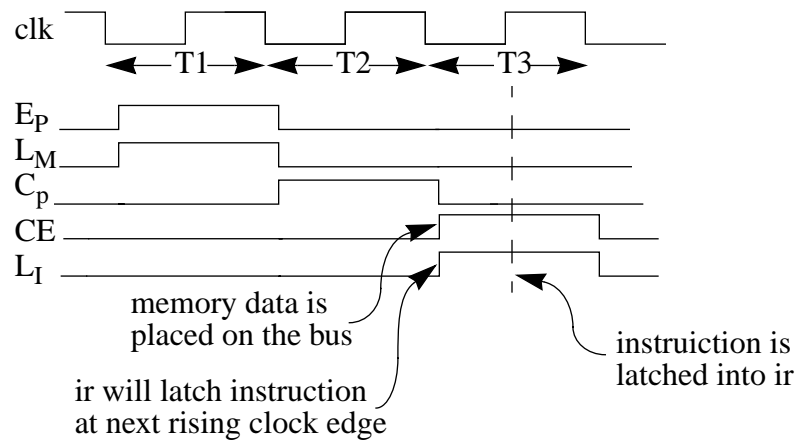
1.  $mar \leftarrow pc$



2.  $pc \leftarrow pc + 1$

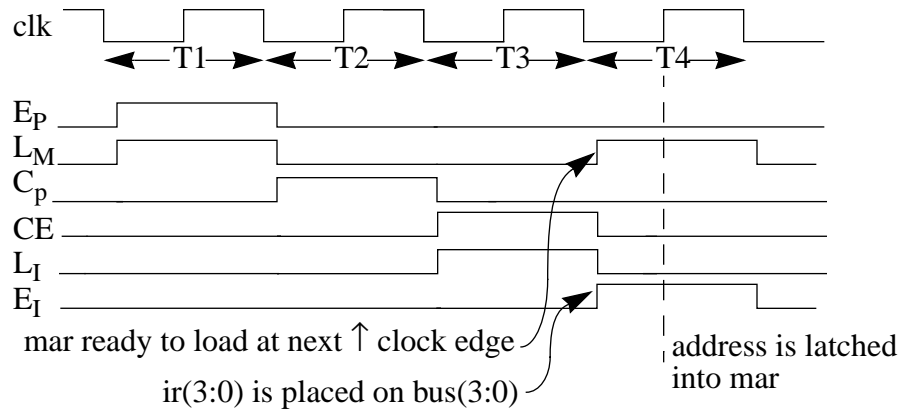


3.  $ir \leftarrow M[mar]$

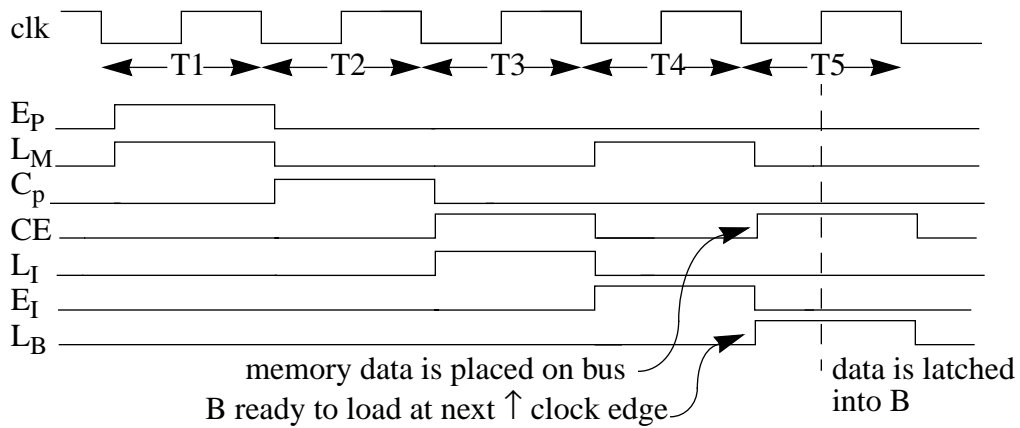


**SAP-1 INSTRUCTION EXECUTION CLOCK CYCLES: sub**

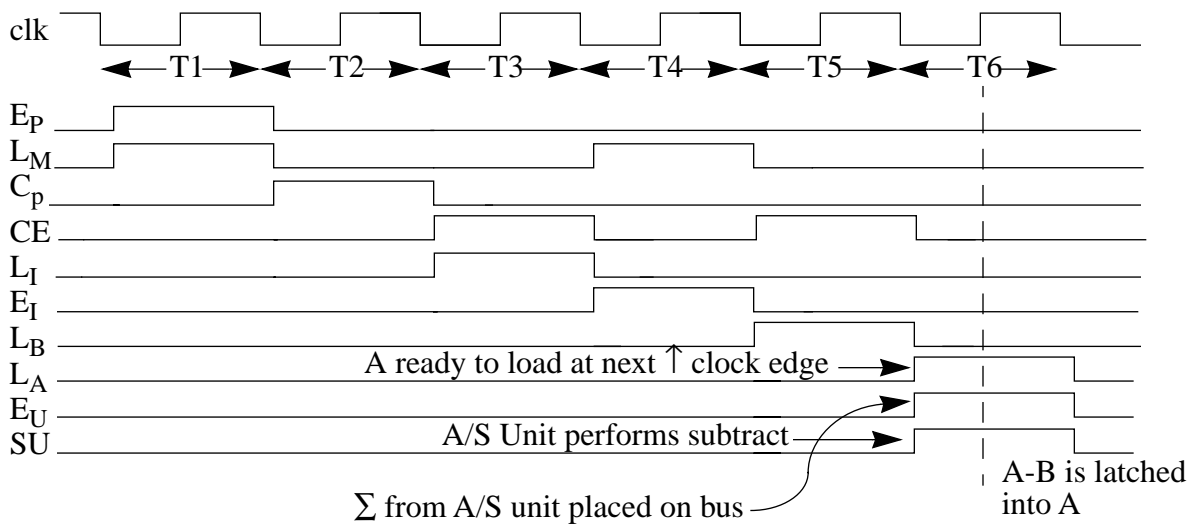
4.  $mar \leftarrow ir(3:0)$



5.  $B \leftarrow M[mar]$



6.  $A \leftarrow A-B$



**CONTROL STEP SEQUENCES FOR EACH INSTRUCTION: SAP-1**

lda (opcode 0000 <sub>2</sub> )			out (opcode 1110 <sub>2</sub> )		
Step	Action	Control Points	Step	Action	Control Points
T1	mar ← pc	E <sub>P</sub> , L <sub>M</sub>	T1	mar ← pc	E <sub>P</sub> , L <sub>M</sub>
T2	pc ← pc+1	C <sub>P</sub>	T2	pc ← pc+1	C <sub>P</sub>
T3	ir ← M[mar]	CE, L <sub>I</sub>	T3	ir ← M[mar]	CE, L <sub>I</sub>
T4	mar ← ir(3:0)	E <sub>I</sub> , L <sub>M</sub>	T4	out ← A	E <sub>A</sub> , L <sub>O</sub>
T5	A ← M[mar]	CE, L <sub>A</sub>	T5	noop	
T6	noop		T6	noop	

add (opcode 0001 <sub>2</sub> )			sub (opcode 0010 <sub>2</sub> )		
Step	Action	Control Points	Step	Action	Control Points
T1	mar ← pc	E <sub>P</sub> , L <sub>M</sub>	T1	mar ← pc	E <sub>P</sub> , L <sub>M</sub>
T2	pc ← pc+1	C <sub>P</sub>	T2	pc ← pc+1	C <sub>P</sub>
T3	ir ← M[mar]	CE, L <sub>I</sub>	T3	ir ← M[mar]	CE, L <sub>I</sub>
T4	mar ← ir(3:0)	E <sub>I</sub> , L <sub>M</sub>	T4	out ← A	E <sub>A</sub> , L <sub>O</sub>
T5	B ← M[mar]	CE, L <sub>B</sub>	T5	B ← M[mar]	CE, L <sub>B</sub>
T6	A ← A+B	E <sub>U</sub> , L <sub>A</sub>	T6	A ← A-B	E <sub>U</sub> , SU, L <sub>A</sub>

hlt (opcode 1111 <sub>2</sub> )		
Step	Action	Control Points
T1	mar ← pc	E <sub>P</sub> , L <sub>M</sub>
T2	pc ← pc+1	C <sub>P</sub>
T3	ir ← M[mar]	CE, L <sub>I</sub>
T4	noop	
T5	noop	
T6	noop	

**INSTRUCTION CONTROL SIGNALS: SAP-1**

$$\begin{aligned} \text{LDA} &= \overline{\text{ir}(7)} \cdot \overline{\text{ir}(6)} \cdot \overline{\text{ir}(5)} \cdot \overline{\text{ir}(4)} & \text{OUT} &= \text{ir}(7) \cdot \text{ir}(6) \cdot \text{ir}(5) \cdot \overline{\text{ir}(4)} \\ \text{ADD} &= \overline{\text{ir}(7)} \cdot \overline{\text{ir}(6)} \cdot \overline{\text{ir}(5)} \cdot \text{ir}(4) & \text{SUB} &= \overline{\text{ir}(7)} \cdot \overline{\text{ir}(6)} \cdot \text{ir}(5) \cdot \overline{\text{ir}(4)} \\ \text{HLT} &= \text{ir}(7) \cdot \text{ir}(6) \cdot \text{ir}(5) \cdot \text{ir}(4) \end{aligned}$$

**CONTROL SIGNAL LOGIC BASED ON INSTRUCTION SIGNALS: SAP-1**

$$\begin{aligned} L_M &= T1 + T4 \cdot (\text{LDA} + \text{ADD} + \text{SUB}) & E_P &= T1 \\ CE &= T3 + T5 \cdot (\text{LDA} + \text{ADD} + \text{SUB}) & C_P &= T2 \\ E_I &= T4 \cdot (\text{LDA} + \text{ADD} + \text{SUB}) & L_I &= T3 \\ L_A &= T5 \cdot \text{LDA} + T6 \cdot (\text{ADD} + \text{SUB}) & E_A &= T4 \cdot \text{OUT} \\ L_B &= T5 \cdot (\text{ADD} + \text{SUB}) & E_U &= T6 \cdot (\text{ADD} + \text{SUB}) \\ SU &= T6 \cdot \text{SUB} & L_O &= T4 \cdot \text{OUT} \end{aligned}$$

## A SIMPLE SAP-1 PROGRAM

; Assembly language code to perform op1 - op2 and display the result

```
add2: lda    op1        ; A <- op1
      sub    op2        ; A <- A - op2 = op1 - op2
      out                    ; display the result
      hlt                    ; stop execution
```

```
op1:  db    4bH
```

```
op2:  db    27H
```

## LISTING FILE GENERATED FROM add2 PROGRAM

; Assembly language code to perform op1 - op2 and display the result

```
0    0f    add2: lda    op1        ; A <- op1
1    2e          sub    op2        ; A <- A - op2 = op1 - op2
2    e0          out                    ; display the result
3    f0          hlt                    ; stop execution

e    4b    op1:  db    4bH
f    27    op2:  db    27H
```