N1 → 2502 H
N2 → 2503 H
LDA 2502 H, MOV B, A
LDA 2503 H → Accumulator
ADD B
STA 2504 H

5) BUN (Branch unconditionally)
It transfer the program to the instruction specified by the effective address.

6) BSA (Branch and Save Address)
Used for branching to a portion of a program called sub routine. When executed BSA instruction stores the address of next instruction in a sequence into a memory locations specified by EA (Effective Address)

7) ISA (Increment and Skip if zero)
This instruction increment the word specified by the effective address. And if the incremented value is equal to 0, PC is incremented by 1.

Symbol	Description
CLA	Clears Accumulator (AC)
CLE	Clears Extended Accumulator
CMA	Complement Accumulator
CME	Complement E
CIR	Circulate Right AC and E
CIL	Circulate Left AC and E
INC	Increment Accumulator
SPA	Skip Next Instruction, if AC is positive.
SNA	Skip Next Instruction, if AC is negative.
SZA	Skip Next Instruction, if AC is 0.
SZE	Skip Next Instruction, if E is 0.
HLT	Halt the computer
INP	Input character to AC.
OUT	Output character from AC.
ION	Interrupt on.
IOF	Interrupt off
SKI	Skip on input flag
SKO	Skip on output flag.

MUX / Multiplexer / Data selector / Many-to-one device.

Date
Page

* Microprogrammed Control Unit

* CPU (Central Processing Unit)

→ The part of the computer that performs the bulk of data processing operator is called the central processing unit (CPU). It is made up of three major parts :-

- Register
  The register sets store intermediate data used during the execution of instruction and also the result obtained after the operation.

- ALU (Arithmetic Logic Unit)
  ALU performs the required micro operation for executing the instruction.

- CU (Control Unit)
  Control Unit supervises the transfer of information among the registers and instructs the ALU as to which instruction to perform.

Generate the control word for the operation.

$$R_2 \leftarrow R_4 + R_5$$

SELA SELB

and OPR for the addition → 01101

1	0	0	1	0	1	0	1	0	0	1	1	0	1

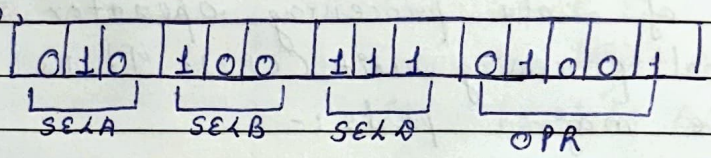
SELA SELB SELA OPR

* Create the control word for CPU for the given operation.

$R_7 \leftarrow R_2 - R_4$

OPR → 01001

Solution,



for the

* Data Transfer Instruction
Data transfer cause transfer of data from one place to another without changing the binary information container. Example :- load (ld), store (st), move (mov), exchange (XCH), input (IN), output (OUT), PUSH (PUSH), pop (POP).

* Data Manipulation Instruction
Data Manipulation instruction are those that performs arithmetic operation and shift operation. It includes arithmetic, logical and bit manipulation and shift instruction.

* Program Control Instruction
Program control instructions provide decision making capabilities and change the path taken by a program when executed in the computer.

1) Arithmetic Instructions

Instructions	Mnemonic
Increment	INC
Decrement	DEC
Add	ADD
Subtract	SUB
Multiply	MUL
Devide	DIV
Add with Carry	ADDC
Sub with Borrow	SUBB
Negate (2's complement)	NCG

2) Logical and Bit Instructions

Instructions	Mnemonic
Clear	CLR
Complement	COM
AND	ANL
OR	ORL
Exclusive OR	XOR
Clear Carry	CLRC
Set Carry	STEC
Complement Carry	COMC
Enable Interrupt	EI
Disable Interrupt	DI

3) Shift Instructions

Instructions	Mnemonic
Logical Shift Right	SHR
Logical Shift Left	SHL
Arithmetic Shift Right	SHRA
Arithmetic Shift Left	SHLA
Rotate Right	ROR
Rotate Left	ROL
Rotate Right Through Carry	RORC
Rotate Left Through Carry	ROLC

4) Program Control Instructions:

Instructions	Mnemonic
Branch	BR
Jump	JUMP
Skip	SKP (always conditional)
Call	CALL (use with sub-routines)
Return	RET (use with sub-routines)
Compare by Sub	CMP
Test (By ANDing)	TST

* Stack

The stack in digital computer is essentially a memory unit with an address register that can count only after an initial value is loaded into it.

- The register that holds the address for the stack is called stack pointer. It's value always points at the top item in the stack. Two operation of stacks are :-

* PUSH

→ The operation of data insertion is called PUSH.

* POP

→ The operation of data deletion is called POP.

1) PUSH OPERATIONS

Initially SP is cleared to 0, EMTY is set to 1 and FULL is cleared to 0. If the stack is not full (FULL = 0). New item is inserted with the push operation following the sequence of micro operations for PUSH :-

1) SP ← SP + 1 (Increment SP)
2) M[SP] ← DR [write item on top of stack]
3) If (SP = 0) then (FULL ← 1), check if stack is full
4) EMTY ← 0 (Mark the stack not empty)

2) POP OPERATION.

The pop operation consists of following sets of micro operations :-

Date
Page

1) DR ← M[SP] (Read item from top of stack)
2) SP ← SP - 1 (Decrement stack pointer)
3) IF (SP = 0) then (EMTY ← 1) (Check if stack is empty)
4) FULL ← 0 (Mark the stack not full)

Instruction Formats [three address, two, one and zero addresses]
A computer have variety of instruction code format. It is the function of the control unit within the CPU to interpret each instruction code and provide the necessary control functions needed to process the instructions.

* Note:
Most computers fall onto 1 of 3 types of CPU organisations :-
i) Single Accumulator organisations
ii) General Register organisations
iii) Stack organisations

Question:
Evaluate the arithmetic statement X = (A+B) * (C+D) using 3 address, 2 address, 1 address and 0 address instructions. Symbols (ADD, SUB, MUL and DIV) are four arithmetic operations.

MOV for transfer type operations, LOAD and STORE for transfer between memory and accumulator. General purpose registers R_1 to R_7 can also be used for temporary storage of information.

→ Three address Instructions

```
ADD   R1 , A, B
ADD   R2 , C, D
MUL   X , R1, R2
```

→ Two address Instructions

```
ADD   A, B
ADD   C, D
MUL   A, C
MOV   X, A
```

→ One Addres Instructions

```
LDA   A     [AC]
ADD   B     [B+AC = AC]
STA   T     [A+B = T]
LDA   C
ADD   D     [D+AC = AC]
MUL   T     [T x AC = AC]
STA   X
```

→ Zero address Instruction.

```
PUSH  A
PUSH  B
ADD
PUSH  C
PUSH  D
ADD
MUL
POP   X
```