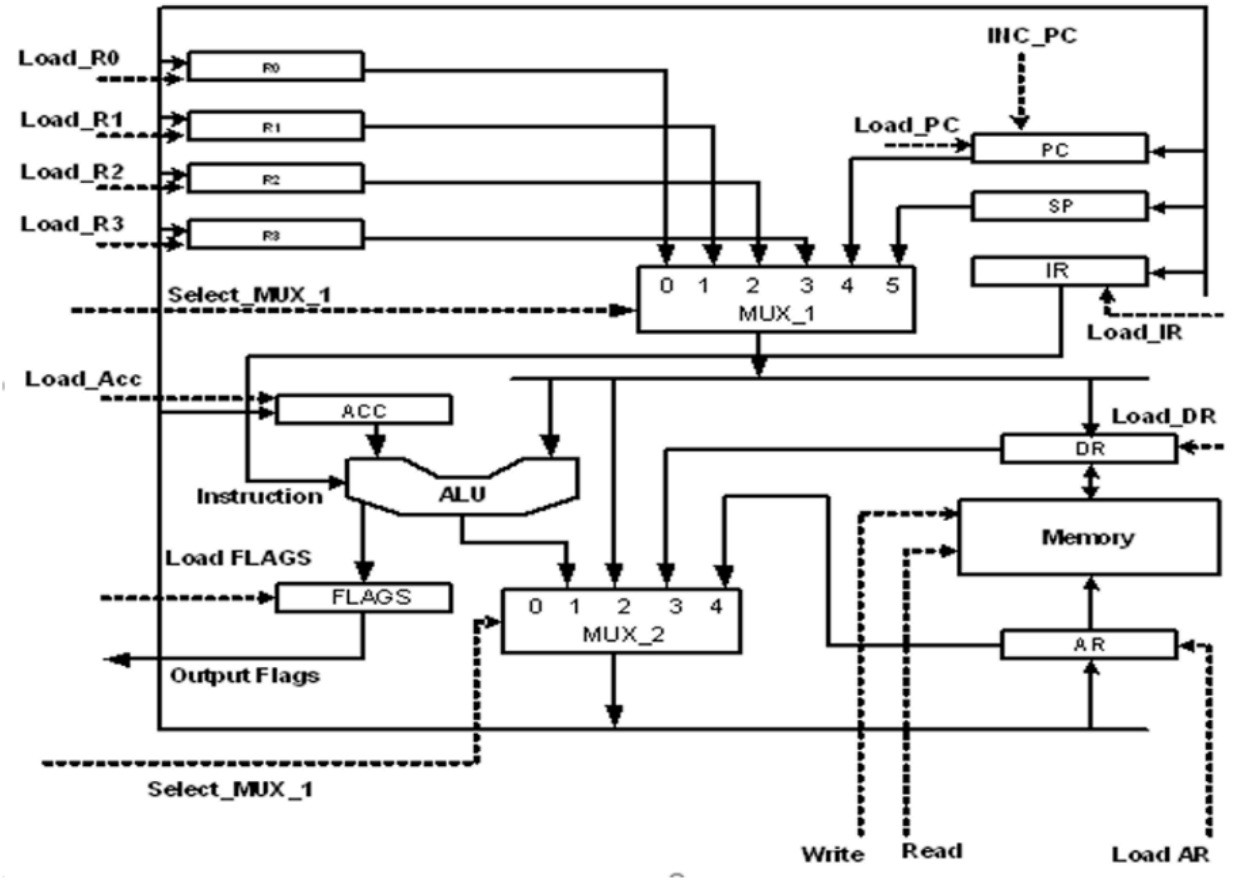
Which of the following instructions is not possible on the following data path?



- A. Add Acc, Acc, SP — Acc ← Acc + SP
- B. Add R2, R1, R3 — R2 ← R1 + R3
- C. Add Acc, Acc, R3 — Acc ← Acc + R3
- D. Add PC, Acc, PC — PC ← Acc + PC
- E. None of them

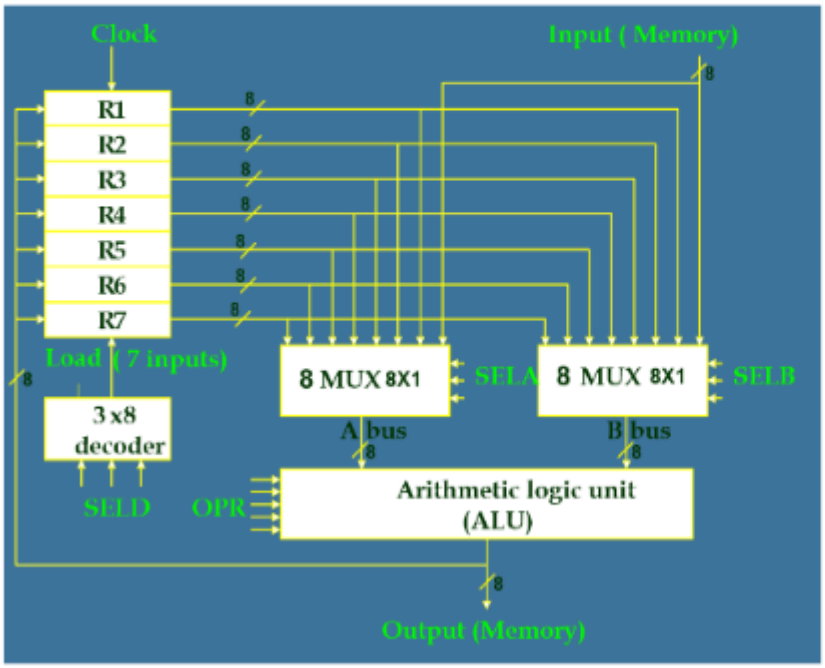
B is correct

○ A.	SUB Acc, Acc, SP	Acc ← Acc - SP
○ B.	SUB R2, Acc, PC	R2 ← Acc - PC
◉ C.	ADD Acc, PC, R3	Acc ← PC + R3
○ D.	Add PC, Acc, PC	PC ←Acc + PC

○ E. None of them

C is correct

How many bits of the control word is cleared on the following data path?



A. 12 bits
B. 15 bits
C. 14 bits
D. 9 bits
E. None of them

SELA (3) + SELB (3) + SELD(3) + OPR(5) = 14 bits

TABLE 8-3 Examples of Microoperations for the CPU

Microoperation	Symbolic Designation SELA	SELB	SELD	OPR	Control Word
$R1 \leftarrow R2 - R3$	R2	R3	R1	SUB	010 011 001 00101
$R4 \leftarrow R4 \vee R5$	R4	R5	R4	OR	100 101 100 01010
$R6 \leftarrow R6 + 1$	R6	—	R6	INCA	110 000 110 00001
$R7 \leftarrow R1$	R1	—	R7	TSFA	001 000 111 00000
Output $\leftarrow R2$	R2	—	None	TSFA	010 000 000 00000
Output $\leftarrow$ Input	Input	—	None	TSFA	000 000 000 00000
$R4 \leftarrow$ shl $R4$	R4	—	R4	SHLA	100 000 100 11000
$R5 \leftarrow 0$	R5	R5	R5	XOR	101 101 101 01100

What is the mode of addressing for the following instruction?

CMP R1 : R1'-> R1

- **A. Register Indirect Mode**
- **B. Register Mode**
- **C. Implied Mode**
- **D. Relative Mode**
- **E. None of them**

What is the mode of addressing for the following instruction?

INC R1 : R1'-> R1

- **A. Register Indirect Mode**
- **B. Register Mode**
- **C. Implied Mode**
- **D. Relative Addressing Mode**
- E. **None of them**

What is the mode of addressing for the following instruction?

LD R2, R4: R4→ R2

A Register Mode

B. Direct Address Mode

C Index Addressing Mode

D. Relative Addressing Mode

E. None of them

What is the mode of addressing for the following instruction?
LD A, (40H) : M(40H)→ A

A Immediate Mode
B. Direct Address Mode
C. Implied Mode
D. Relative Addressing Mode

What is the mode of addressing for the following instruction?
JR 40H : PC+40H→ PC

A Register Indirect Mode
B: Register Mode
C Implied Mode
D. Relative Addressing Mode

What is the mode of addressing for the following instruction?
LD A, 40H : 40H→ A

A Immediate Mode
B. Direct Address Mode
C Implied Mode
D. Relative Addressing Mode

What is the mode of addressing for the following instruction?

LD A, (IX+40H) : M(IX+40H)➔ A

- A. Immediate Mode
- B. Direct Address Mode
- C. Index Addressing Mode
- D. Relative Addressing Mode
- E. None of them

Which of the following components can not be in an MCU?
A. A/D converter
B. D/A converter
C. serial I/O controller
D. Timer
E. None of them
Explanation of the options:

- **A. A/D converter (Analog-to-Digital converter)**: Often included in MCUs for converting analog signals into digital values.
- **B. D/A converter (Digital-to-Analog converter)**: Not typically found in many MCUs; if needed, it may be added as an external component.
- **C. Serial I/O controller**: Commonly integrated in MCUs for handling serial communication (e.g., UART, SPI, I2C).
- **D. Timer**: A standard feature in MCUs for timing operations and generating delays or managing events.

What is The difference between SOC & NOC?

A. SOC has CPU but NOC doesn't have CPU
B. SOC has internal memory but NOC doesn't have internal memory
C. NOC may contain multi SOC cores
D. SOC may contain multi NOC cores
E. NOC contains I/O but SOC doesn't

- **System on Chip (SoC)** is a complete integrated circuit that includes a CPU, memory, I/O ports, and other components, all on a single chip. It is designed to handle a wide range of processing tasks and is typically used in smartphones, tablets, and other embedded systems.
- **Network on Chip (NoC)** is a communication subsystem within a system on a chip (SoC) that enables communication between different cores or components of the SoC. It provides a scalable and efficient way to interconnect various processing elements within a multi-core SoC.

Explanation of the options:

- **A.** Incorrect. Both SoC and NoC may contain CPUs.
- **B.** Incorrect. SoCs often have internal memory, and NoCs are part of SoCs, which can have memory.
- **C.** Correct. A NoC can interconnect multiple SoC cores, serving as the communication framework within a multi-core SoC.
- **D.** Incorrect. An SoC contains a NoC, not the other way around.

  E. Incorrect. Both SoC and NoC can contain I/O components.

Which of the following components can not be in an MCU?
A. RAM memory
B. PROM memory
C. I/O block
D. CPU
E. All of them could be in an MCU

ARM processors are designated for?
• A. low-power operation
• B. mobile devices
• C. Longer battery life
• D. simple design as possible
• E. All of them

E. All of them

ARM processors are designed with several goals in mind, including:

- **A. Low-power operation**: ARM processors are known for their energy efficiency, making them ideal for battery-operated devices.
- **B. Mobile devices**: ARM processors are widely used in mobile devices such as smartphones and tablets due to their power efficiency and performance.
- **C. Longer battery life**: The power-efficient design of ARM processors contributes to longer battery life in portable devices.
- **D. Simple design as possible**: ARM processors are designed with a relatively simple and efficient instruction set, which helps in reducing power consumption and improving performance.

ARM processors address all of these aspects, making them a popular choice for various applications where power efficiency and performance are critical.

Which of the following items are not drive the wearable market?

- ○ A. Fitness
- ○ B. Healthcare
- ○ C. Longer battery life
- ○ D. Fashion
- ◉ E. All of them

How many multiplexers are used on the following data path with an 8bit data size?

- ○ A. 32 MUX
- ◉ B. 16 MUX
- ○ C. 8 MUX
- ○ D. 64 MUX
- ○ E. None of them

In CISC processors PC register has?

A. address of memory for access to a new address

B. address of memory for access to a data

C. address of memory for access to an instruction

D. address of memory for both data and instruction simultaneously

E. A, B and C items are correct

MEMORY

256 × 8

AR =Address Register

PC=Program Counter
DR=Data Register
AC=Accumulator
IR=Instruction Register

What is the instruction for following RTL?

```
PC[10H]      ⇨    AR
PC+1         ⇨    PC
M(10H)       ⇨    DR
DR           ⇨    IR&ID
PC[11H]      ⇨    AR
PC+1         ⇨    PC
M(11H)       ⇨    DR[40H]
DR[40H]      ⇨    AR
AC           ⇨    DR
DR           ⇨    M(AR)
0            ⇨    SC
```

LD (40H), A

What is the instruction for following RTL?

```
PC[10H]      ⇨    AR
PC+1         ⇨    PC
M(10H)       ⇨    DR
DR           ⇨    IR&ID
PC[11H]      ⇨    AR
PC+1         ⇨    PC
M(11H)       ⇨    DR[40H]
DR[40H]      ⇨    AR
M(40H)       ⇨    DR[50H]
DR[50H]      ⇨    AC
0            ⇨    SC
```

LD A,(40H) M(40H) => A

- ○ A. LD A , 40H — 40H ⇒ A
- ◉ B. LD A , (40H) — M(40H) ⇒ A
- ○ C. LD 40H, A — A⇒ M(40H)
- ○ D. LD A , R1 — R1 ⇒ A

What is the instruction for following RTL?

PC[10H]	⇨	AR
PC+1	⇨	PC
M(10H)	⇨	DR
DR	⇨	IR&ID
R1	⇨	AR
M(R1)	⇨	DR
DR	⇨	AC
0	⇨	SC

LD A, (R1) M(R1)=>A

- ○ A. LD A , 40H — 40H ⇒ A
- ◉ B. LD A , (R1) — M(R1) ⇒ A
- ○ C. LD 40H, A — A⇒ M(40H)
- ○ D. LD A , R1 — R1 ⇒ A
- ○ E. Non of them

What is the instruction for following RTL?

PC[10H] ⇨ AR
PC+1 ⇨ PC
M(10H) ⇨ DR
DR ⇨ IR&ID
PC[11H] ⇨ AR
PC+1 ⇨ PC
M(11H) ⇨ DR[40H]
ADD A, DR[40H] ⇨ A
0 ⇨ SC

ADD A, (40H) A+M(40H) -> A

- A. ADD A , 40H — A+40H → A
- B. ADD A , (40H) — A+M(40H) → A
- C. LD (40H), A — A → M(40H)
- D. ADD A , R1 — A+R1 → A

What is the instruction for following RTL?

PC[10H] ⇨ AR
PC+1 ⇨ PC
M(10H) ⇨ DR
DR ⇨ IR&ID
R1 ⇨ AC
0 ⇨ SC

LD A, R1 R1=>A

○ A. LD A , 40H	40H ⇒ A
○ B. LD A , (40H)	M(40H) ⇒ A
○ C. LD 40H, A	A⇒ M(40H)
◉ D. LD A , R1	R1 ⇒ A

What is the instruction for following RTL?

PC[10H]	⇨	AR
PC+1	⇨	PC
M(10H)	⇨	DR
DR	⇨	IR&ID
PC[11H]	⇨	AR
PC+1	⇨	PC
M(11H)	⇨	DR[40H]
DR[40H]	⇨	AC
0	⇨	SC

LD A , 40H

Immediate

PC[10H]	⇨	AR
PC+1	⇨	PC
M(10H)	⇨	DR
DR	⇨	IR&ID

Fetch Cycle →

PC[11H]	⇨	AR
PC+1	⇨	PC
M(11H)	⇨	DR[40H]
DR[40H]	⇨	AC
0	⇨	SC

What is the instruction for following RTL?

PC[10H]	⇨	AR
PC+1	⇨	PC
M(10H)	⇨	DR
DR	⇨	IR&ID
SP	⇨	AR
R3	⇨	DR
DR	⇨	M(AR)
SP - 1	⇨	Sp
0	⇨	SC

- ○ A. POP R3 — M(SP) ⇒ R3
- ○ B. LD A , (R3) — M(R3) ⇒ A
- ○ C. LD (R3), A — A ⇒ M(R3)
- ◉ D. Push R3 — R3 ⇒ M(SP)

The operation SP - 1 → SP is typically associated with a PUSH operation.

What is the instruction for following RTL?

```
PC[10H]   ⇨   AR
PC+1      ⇨   PC
M(10H)    ⇨   DR
DR        ⇨   IR&ID

SP +1     ⇨   SP
SP        ⇨   AR
M(AR)     ⇨   DR

DR        ⇨   R3
0         ⇨   SC
```

The operation SP + 1 → SP is typically associated with a Pop operation.

What is the instruction for following RTL?

```
PC[10H]   ⇨   AR
PC+1      ⇨   PC
M(10H)    ⇨   DR
DR        ⇨   IR&ID

PC[11H]   ⇨   AR
PC+1      ⇨   PC
M(11H)    ⇨   DR[40H]
DR[40H]   ⇨   AR
M(40H)    ⇨   DR

DR        ⇨   PC
0         ⇨   SC
```

◉ A.	JP (40H)	(40H) → PC
○ B.	LD A , PC	PC → A
○ C.	JP 40H	40H → PC
○ D.	Push R3	R3 → M(SP)

What is the instruction for following RTL?

```
PC[10H]   ⇨   AR
PC+1      ⇨   PC
M(10H)    ⇨   DR
DR        ⇨   IR&ID
```

```
PC[11H]     ⇨  AR
PC+1        ⇨  PC
M(11H)      ⇨  DR[40H]
DR[40H]     ⇨  PC
0           ⇨  SC
```


JP 40H

```
PC[10H]   ⇨   AR
PC+1      ⇨   PC
M(10H)    ⇨   DR
DR        ⇨   IR&ID
```

```
PC[11H]     ⇨  AR
PC+1        ⇨  PC
M(11H)      ⇨  DR[40H]
DR[40H]     ⇨  PC
0           ⇨  SC
```

What is the instruction for following RTL?

PC[10H] ⇨ AR
PC+1 ⇨ PC
M(10H) ⇨ DR
DR ⇨ IR&ID
R1+1 ⇨ R1
0 ⇨ SC

○ A.

INC A A+1 ➔ A

○ B. INC PC PC+1 ➔ Pc

○ C. LD (40H), A A ➔ M(40H)

◉ D. INC R1 R1 + 1 ➔ R1

○ E. None of them

In stack structure with post-increment, top full, and 16bit as word size,

- A. with each POP, SP-1➔ SP
- B. with each POP, SP-4➔ SP
- C. with each POP, SP-2➔ SP
- D. with each PUSH, SP+1➔ SP
- E. none of them

In a stack structure with **post-increment**, **top full**, and a **16-bit word size**, the correct behavior for a POP operation is:

- After popping a value from the stack, the stack pointer (SP) is incremented by the word size, which is 2 bytes (16 bits).

Given this, the correct answer is:

C. with each POP, SP-2 → SP

In stack structure with post-increment, top full, and 16bit as word size,

- A. with each PUSH, SP+1➔ SP
- B. with each PUSH, SP+4➔ SP
- C. with each PUSH, SP+2➔ SP
- D. with each POP, SP+1➔ SP

In stack structure with post-increment, top full, and 8bit as word size, which of the following items is correct?
A.with each PUSH, first, there is a write-on memory then SP+1-> SP
B.with each PUSH, first, there is a write-on memory then SP+2-> SP
C.with each PUSH, first, there is SP+1-> SP, then write on memory

D.with each PUSH, first, there is SP+2-> SP, then write on memoryE.none of themC is correct

---

In stack structure with post-increment, top full, and 8bit as word size,

- A. with each POP, first, there is a read from memory then SP-1➔ SP
- B. with each POP, first, there is a read from memory then SP-2 ➔ SP
- C. with each POP, first, there is SP-1➔ SP, then read from memory
- D. with each POP, first, there is SP-2➔ SP, then read from memory