

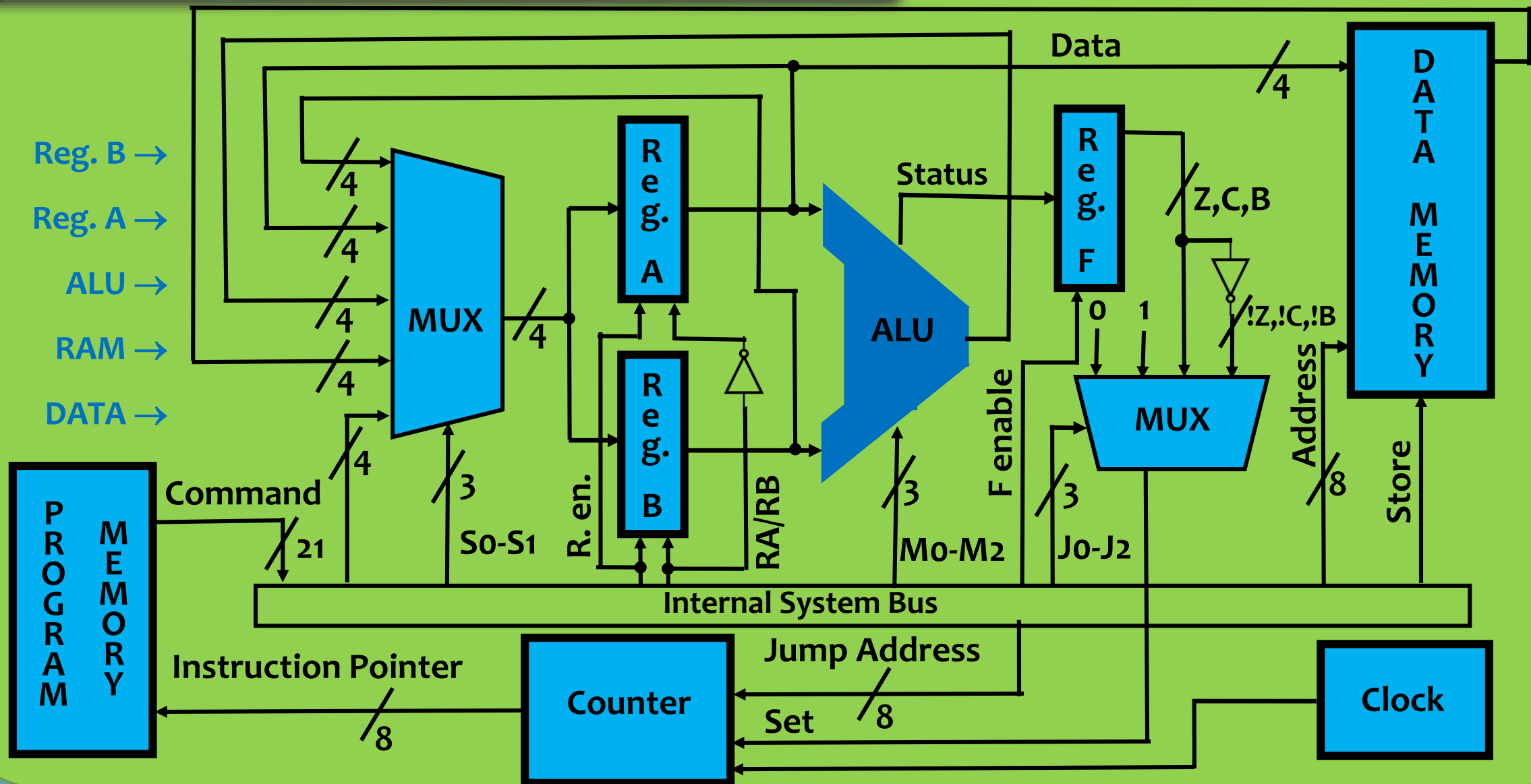


# Mikroprocesszorok építőelemei

# Tartalomjegyzék

Témakör	Főbb tartalmak	
4 bites mikroszámítógép	Elvi felépítés, Működési leírás, Utasítás felépítés.	

# Egy 4 bites CPU felépítése





# Egy 4 bites CPU felépítése

- Az előző dián egy egyszerű 4 bites mikroszámítógép elvi felépítése látható. A processzor 4 bites adatokkal tud műveleteket végezni. Az ALU az A és B regiszterek adatain végzi el a kijelölt aritmetikai műveleteket amik eredményét az A vagy B regiszterbe írja vissza. A két regiszter egy multiplexer segítségével kaphat 5 forrásból adatokat. A műveletek eredményétől függően az F regiszter egyes bitjei jelzik a műveletvégrehajtás státuszát (eredmény nulla (Z), vagy túlcsordulás (C) esetleg alulcsordulás (B) történt).
- A mikroszámítógép tartalmaz egy  $256 \times 4$  bites adatmemóriát és egy  $256 \times 21$  bites programmemóriát. Ez a felépítés a Harvard architektúrának felel meg, mivel az adatok és az utasítások egymástól eltérő helyen és fel nem cserélhető módon tárolódnak.
- A programmemóriában az utasítások dekódolt módon tárolódnak. Ez nem hatékony, ezért a processzorok tartalmaznak egy utasítás dekódoló egységet is, amit jelen esetben lehangyunk.
- A programmemóriának a címzését egy 8 bites bináris, programozható számláló állítja elő (ez valósítja meg az utasítás mutatót (IP)). Normális működés esetében minden végrehajtott utasítást követően a számláló értéke eggyel növekszik. A programozható számláló értékének átírásával az ugrások (Jump) valósíthatók meg.

# Egy 4 bites CPU felépítése

- A processzor működésének megértéséhez elengedhetetlen az egyes részegységek vezérlőjeleinek meghatározása, ismerete.

Bemenet			Kimenet
S2	S1	So	A kimeneten megjelenő adat
0	0	0	Regiszter A
0	0	1	Regiszter B
0	1	0	ALU
0	1	1	RAM
1	0	0	Adat
1	0	1	-
1	1	0	-
1	1	1	-

Az adatokat választó multiplexer igazságtáblázata.

Bemenet			Kimenet
M2	M1	Mo	A kimeneten megjelenő adat
0	0	0	A plusz B
0	0	1	A mínusz B
0	1	0	A and B
0	1	1	A or B
1	0	0	A xor B
1	0	1	A jobbra eltolás
1	1	0	A balra eltolás
1	1	1	not A

Az ALU igazságtáblázata.

Bemenet			Kimenet
J2	J1	Jo	A kimeneten megjelenő adat
0	0	0	0 (nincs ugrás)
0	0	1	1 (ugrás)
0	1	0	Z (ugrás, ha Z)
0	1	1	C (ugrás, ha C)
1	0	0	B (ugrás, ha B)
1	0	1	!Z (ugrás, ha nem Z)
1	1	0	!C (ugrás, ha nem C)
1	1	1	!B (ugrás, ha nem B)

A programszámláló multiplexer igazságtáblázata.

Bemenet	Kimenet
Enable	A kimeneten megjelenő adat
0	Egység tiltása
1	Egység engedélyezése

Az egy bites eszközök engedélyezésének igazságtáblázata.

# Egy 4 bites CPU felépítése

A 4 bites processzor utasításának felépítése.

20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Vezérlés													Cím vagy operandus							
Flag írás	RAM írás	ALU műveletek			Forrásadat választás			Munka-regiszter engedélyezés		Vezérlésátadás (ugrás)			Utasítás cím (8 bit)							
F. en.	Store	M2	M1	Mo	S2	S1	So	R. en.	A/B	J2	J1	Jo	X	X	X	X	Adat (4 bit)			

20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
F. en.	Store	M2	M1	Mo	S2	S1	So	R. en.	A/B	J2	J1	Jo	X	X	X	X	D3	D2	D1	Do
0	0	X	X	X	1	0	0	1	0	0	0	0	X	X	X	X	1	1	1	1

Példa az „A regiszter értéke legyen 15” azaz MOV A,15 utasításra. Az „X”-el jelölt bitek értéke nem releváns, azaz bármi lehet. Ahhoz, hogy az utasításkód kiszámítható legyen meg kell határozni egy szabályt. Legyen minden nem releváns bit értéke „0”. A fentiek szerint a MOV A,15 bináris kódja: 00 000 100 10 000 0000 1111

20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
F. en.	Store	M2	M1	Mo	S2	S1	So	R. en.	A/B	J2	J1	Jo	X	X	X	X	D3	D2	D1	Do
0	0	X	X	X	1	0	0	1	1	0	0	0	X	X	X	X	0	0	1	1

Példa a „B regiszter értéke legyen 3” azaz MOV B,3 utasításra. A fentiek szerint a MOV A,3 bináris kódja: 00 000 100 11 000 0000 0011



# Egy 4 bites CPU felépítése

20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
F. en.	Store	M2	M1	Mo	S2	S1	So	R. en.	A/B	J2	J1	Jo	X	X	X	X	D3	D2	D1	Do
1	0	0	0	0	0	0	0	1	0	0	0	0	X	X	X	X	X	X	X	X

Példa az „A regiszter értéke legyen A plusz B” azaz ADD A,B utasításra. Az eredmény (amennyiben az előző két példa alapján A értéke 15 és B értéke 3) túlsordulással jár, amit az „F” regiszter tárol, mivel az utasítás „F. en.” bitje 1, azaz a Flag regiszter írását engedélyezi.  
A fentiek szerint az ADD A,B bináris kódja: 10 000 000 10 000 0000 0000

20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
F. en.	Store	M2	M1	Mo	S2	S1	So	R. en.	A/B	J2	J1	Jo	A7	A6	A5	A4	A3	A2	A1	A0
0	1	X	X	X	X	X	X	0	X	0	0	0	0	0	0	0	0	0	0	0

Példa az „A regiszter értékének a 0-ás című memória rekeszbe történő írására” azaz STO 0,A utasításra.  
A fentiek szerint a STO 0,A bináris kódja: 01 000 000 00 000 0000 0000

A példákban szereplő négy művelet egymást követő végrehajtásához a program az alábbi módon írható le:

Mnemonic	Bináris kód	Működés (pszekodód)
MOV A,15	00 000 100 10 000 0000 1111	A regiszter értéke 15
MOV B,3	00 000 100 11 000 0000 0011	B regiszter értéke 3
ADD A,B	10 000 000 10 000 0000 0000	A regiszter értéke A és B összege
STO 0,A	01 000 000 00 000 0000 0000	A regiszter értéke a 0-ás memóriacímre mentve

# Egy 4 bites CPU felépítése

20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
F. en.	Store	M2	M1	Mo	S2	S1	So	R. en.	A/B	J2	J1	Jo	A7	A6	A5	A4	A3	A2	A1	A0
0	0	X	X	X	X	X	X	0	X	0	0	1	0	0	0	1	0	0	0	0

Példa a „feltétel nélküli ugrásra a 16-os programcímre” azaz JMP 16 utasításra.  
A fentiek szerint a JMP 16 bináris kódja: 00 000 000 00 001 0001 0000

20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
F. en.	Store	M2	M1	Mo	S2	S1	So	R. en.	A/B	J2	J1	Jo	A7	A6	A5	A4	A3	A2	A1	A0
0	0	X	X	X	X	X	X	0	X	1	0	1	0	0	0	1	0	1	0	0

Példa a „feltételes ugrásra a 20-as programcímre”. Az ugrás akkor hajtódik végre, ha a Zero (Z) flag értéke nulla, azaz az előző művelet végrehajtása során az eredmény nem nulla lett.

A fentiek szerint a JNZ 20 bináris kódja: 00 000 000 00 101 0001 0100

## A példa az elágazásokra:

Program cím	Mnemonic	Bináris kód	Működés (pszeudokód)
018	MOV A,2	00 000 100 10 000 0000 1111	A regiszter értéke 16
019	MOV B,4	00 000 100 11 000 0000 0011	B regiszter értéke 3
020	ADD A,2	10 000 000 10 000 0000 0000	A regiszter értéke A és B összege
021	SUB B,1	10 001 000 11 000 0000 0000	B regiszter értéke B mínusz 1 lesz
022	JNZ 20	00 000 000 00 101 0001 0100	Ha a kivonás eredménye nem nulla, akkor ugrás a 20-as programcímre
023	STO 0,A	01 000 000 00 000 0000 0000	A regiszter értéke a 0-ás memóriacímre mentve