Címzési módok gyakorlati alkalmazása

Bevezetés

8085-nél láttuk (Digit2):

Utasítás	**Leírás**
MOV r1,r2	Adatmozgatás regiszterek között
MVI r1, adat8	Konstans betöltés regiszterbe
MOV M,r1	Adatmozgatás indirekt címzéssel (HL), tetszőleges regisztebe
MOV r1,M	Adatmozgatás indirekt címzéssel (HL), tetszőleges regiszterből
LDAX rp	Akkumulátor betöltése indirekt címzéssel
STAX rp	Akkumulátor mentése indirekt címzéssel
LDA cím16	Akkumulátor betöltése közvetlen címzéssel
STA cím16	Akkumulátor mentése közvetlen címzéssel
LHLD cím16	16 bites adat beolvasása HL-be közvetlen memória címről
SHLD cím16	16 bites adat kiírása HL-ből közvetlen memória címre
PUSH rp	16 bites regiszterpár mentés, SP indirekt címzéssel, posztdekremetálással M(SP--)
POP rp	16 bites regiszterpár betöltése, SP indirekt címzéssel, posztinkrementálással M(SP++)

1. Táblázat

Ezek a címzési módok a magasszintű nyelvek által kedvelt struktúrák, tömbök kezelését nem igazán támogatják.

Főbb problémák:

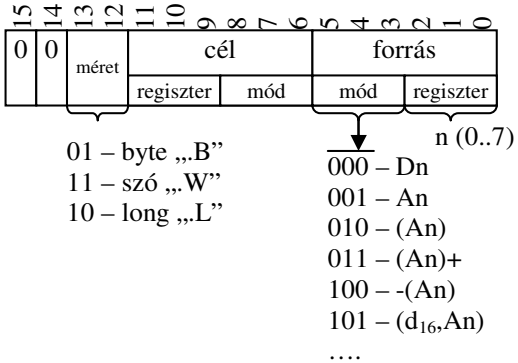
- nem támogatja a különböző adatméretek indexelését (byte, szó, duplaszó)
- hiányzik az automatikus inkrementálás/dekrementlás
- hiányzik a fix offset (displacement) megadás lehetősége az indexhez képest
- nehézkes a regiszter párok töltése, cseréje (pl.: láncolt lista kezelés)

A korszerűbb processzorok már sokféle címzési módot támogatnak, ezek közül a Motorola 68000 MOVE utasítását vesszük példának a gyakorlaton. A címzési módokkal előadáson már foglalkoztunk, itt csak röviden foglaljuk össze a gyakorlati példákhoz szükséges részeket.

Az M68k főbb jellemzői:

- Minden regiszter 32 bites,
- 8db adat és 8db cím regiszter
- Memória tartomány 32 biten címezhető

A MOVE utasítás gépi kódja (OP kód):



Szintaxis:

MOVE.B forrás, cél - byte méretű adat mozgatása
MOVE.W forrás, cél - szó (16 bit) méretű adat mozgatása
MOVE.L forrás, cél - duplaszó (32bit) méretű adat mozgatása

Forrás vagy cél lehet:

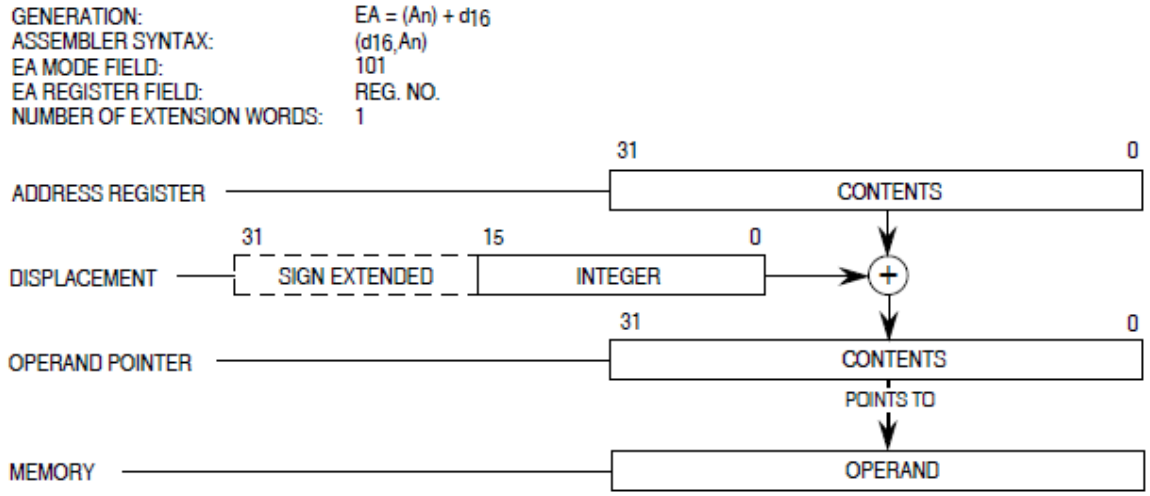
Dn Adat regiszter (n=0..7)
An Cím regiszter (n=0..7)
(An) címregiszter indirekt
(An)+ címregiszter indirekt, automatikus post inkrementálással
-(An) címregiszter indirekt, automatikus pre dekrementálással
(d_{16},An) regiszter indirekt címzés, 16 bites előjeles eltolással (ebben az esetben egy külön kiegészítő szavakban kerül megadásra a d_{16} érték) Lásd 1. ábra.

Amennyiben az utasításban pre/post inkrementálás/dekrementálás van megadva, a címregiszter a mozgatott adat méretének megfelelően változik (1, 2 vagy 4-el).

Amennyiben az utasításhoz kiegészítő paraméterek is tartoznak (pl.: d16, stb.), azt további szavakban tárolják az OP kódot követően. Ezek felépítésével nem foglalkozunk.

A memóriában a Motorola konvenció alapján Big endian (kisebb címen magasabb helyérték) formában vannak tárolva az egyes értékek.

Ha egy utasítással csak 8 vagy 16 bitet töltünk egy 32 bites regiszterbe, a felsőbb bitek változatlanok maradnak. Csak a mozgatott adatnak megfelelő (8, 16 vagy 32 bit) alsóbb bitek íródnak át.



1. ábra

1. feladat

Az alábbi kiinduló adatokat feltételezve adja meg, hogy mi lesz a következő assembly utasítások végrehajtása során az effektív cím és a regiszter értéke. Minden utasítás értelmezésekor a táblázat szerinti kezdőértékeket feltételezze.

Cím regiszter	Értéke
A0	0x0000 1000
A1	0x0000 1002
A2	0x0000 1008
A3	0x0000 100C
A4	0x0000 1010
A5	0x0000 1018
A6	0x0000 1014
A7	0x0000 10FF

Adat regiszter	Értéke
D0	0x0000 0045
D1	0x00F0 0045
D2	0xFFFF 0000
D3	0x0000 1234
D4	0x0000 4343
D5	0x0000 5656
D6	0x0000 7899
D7	0x0000 8888

Memória cím	értéke
0x0000 1018	0x0000 EEE4
0x0000 1014	0x0000 FFF0
0x0000 1010	0x0000 0000
0x0000 100C	0x0000 1008
0x0000 1008	0x0000 1000
0x0000 1004	0x0000 0002
0x0000 1000	0x0000 0001
0x0000 0FFC	0x0000 0000

Adja meg az utasítás végrehajtása után a megváltozott memória és regiszter címeket és tartalmukat.

MOVE.L A5, D3	Cím/reg. név	Új érték

MOVE.L (A5), D2	Cím/reg. név	Új érték

MOVE.W (A5), D2	Cím/reg. név	Új érték

MOVE.B (A5), D3	Cím/reg. név	Új érték

MOVE.W -(A0), D1	Cím/reg. név	Új érték

MOVE.W (A1)+, D1	Cím/reg. név	Új érték

MOVE.W (A0)+, D1	Cím/reg. név	Új érték

MOVE.L (A6)+, D2	Cím/reg. név	Új érték

MOVE.L (#4.W, A0), D3	Cím/reg. név	Új érték

Veremkeret alkalmazása

2. feladat

Egy ***C nyelvű*** programban adott a következő ***függvény***:

```
int f(int i, int j, int k);
{ int m;
….
};
```

8086-os processzornál a fenti függvényt az f(0xA1B1,0xC1D1h,0x1000) aktuális paraméterekkel hívják meg távoli szubrutinhívással (**CS,IP**). A mellékelt ábrán a memóriának az a része látható, amelybe a veremkeret használatakor szokásos szerkezetet (stack frame) is felépítik. A rutin magja használja (rontja) a **BX** és az **SI** regisztereket is. Írja be a memória rekeszekbe a keret felépítése során beírt konkrét értékeket, illetve a szimbolikus jelölésekkel a mentett vagy mentendő regisztereket.

Adja meg az **m** lokális változó elérésére használt címzési móddal előállított effektív címet (képlettel), és adja meg a konkrét értékét (ha kerethez az **SS:BP** regisztert használják).

Adja meg a **keret aktuális** (függvény végrehajtása alatti) hexa értékét.

Mem.cím (hexa)	
SS:SP→ EFA02	
EFA00	
EF9FE	
EF9FC	
EF9FA	
EF9F8	
EF9F6	
EF9F4	
EF9F2	
EF9F0	
EF9EE	
EF9EC	
EF9EA	

IO kezelés

A gyakorlatra való felkészülés során ajánlott az alábbi témakörök átismétlése az előadás vázlatok alapján: Különböző kódolási eljárások a mágneses adatrögzítéshez, sínek felépítése, főbb jellemzői, többprocesszoros rendszerek, buszarbitráció multimasteres környezetben, adatátviteli módok.

3. feladat

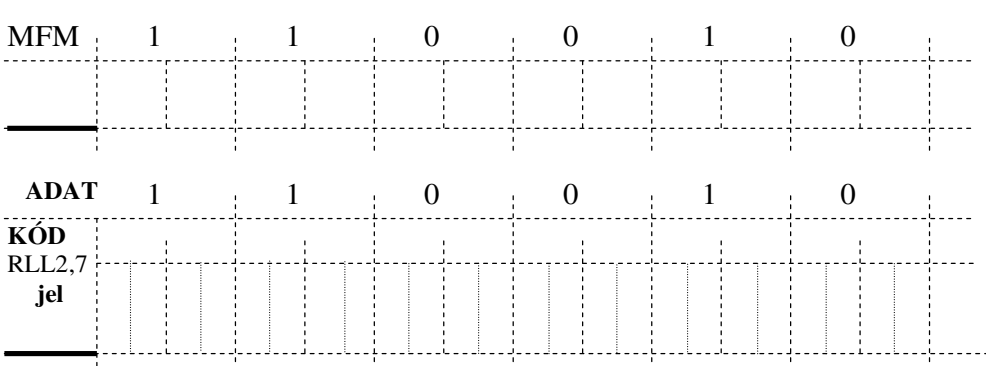
Rajzolja fel egy mágneses tároló, íróáram jelalakját az **10011** bitsorozatra **különböző kódolások esetén (NRZ, NRZI, PE, FM, MFM)**. Tételezze fel, hogy a bitsorozatot megelőző időben az áram 0 értékű. (A függőleges folytonos vonalak a cellahatárokat jelölik)

0…0	1	0	0	1	1
NRZ					
NRZI					
PE					
FM					
MFM					

4. feladat

Rajzolja be az alábbi ábrába az **íróáram jelalakját MFM és RLL2,7** kódolás esetén.

ADAT	RLL2,7 KÓD
11	1000
10	0100
000	000100
010	100100
011	001000
0011	00001000
0010	00100100

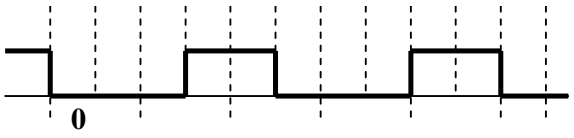


Melyik kódolással és mennyivel több adatot rögzíthetnek azonos fluxusváltási sűrűséget eltételezve?

Mit jelent az RLL2,7 kódolás elnevezésében a **2** és a **7**?

5. feladat

Egy mágneses elvű adattárolónál az íróáram az ábrán látható módon változik. A szaggatott vonalak az ábrán egyforma időközönként vannak.
Az első bit értéke 0.



a) A felírásnál PE vagy FM vagy MFM kódolást alkalmazhattak.
A jelalak alapján **melyik kódolási eljárást használták** a felírásnál?

b) **Adja meg** a jelölt első 0 értékű bitet követő további 4 bit értékét.

c) Azonos fluxusváltási sűrűséget feltételezve MFM kódolással **hányszor több adat** tárolható, mint FM kódolással?

Megoldások

1. feladat

MOVE.L A5, D3	
Cím/reg. név	Új érték
D3	0x0000 1018

MOVE.L (A5), D2	
Cím/reg. név	Új érték
D2	0x0000 EEE4

MOVE.W (A5), D2	
Cím/reg. név	Új érték
D2	0xFFFF 0000

MOVE.B (A5), D3	
Cím/reg. név	Új érték
D3	0x0000 1200

MOVE.W -(A0), D1	
Cím/reg. név	Új érték
A0	0x0000 0FFE
D1	0x00F0 0000

MOVE.W (A1)+, D1	
Cím/reg. név	Új érték
A1	0x0000 1004
D1	0x00F0 0001

MOVE.W (A0)+, D1	
Cím/reg. név	Új érték
A0	0x0000 1002
D1	0x00F0 0000

MOVE.L (A6)+, D2	
Cím/reg. név	Új érték
A6	0x0000 1018
D2	0x0000 FFF0

MOVE.L (#4.W, A0), D3	
Cím/reg. név	Új érték
A0	Nem változik
D3	0x0000 0002

2. feladat

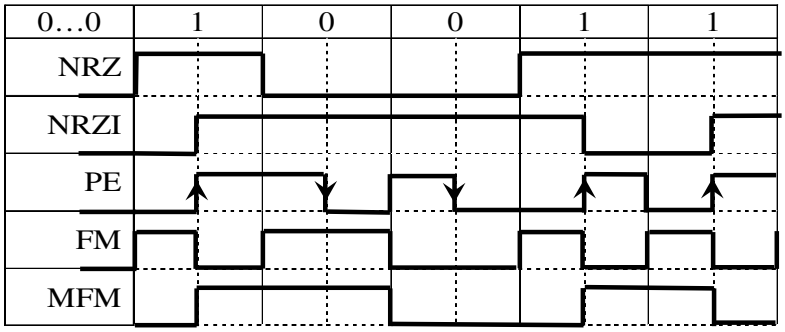
Mem.cím(hexa)	
SS:SP→EFA02	
EFA00	**1000**
EF9FE	**C1D1**
EF9FC	**A1B1**
EF9FA	**CS**
EF9F8	**IP**
EF9F6	**BX**
EF9F4	**SI**
EF9F2	**BP**
EF9F0	**m**
EF9EE	
EF9EC	
EF9EA	

Effektív cím = BP-2

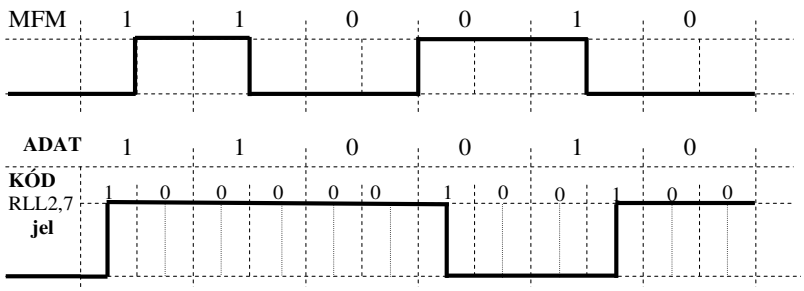
m címe: EF9F0H

Keret: **EF9F2**H

3. feladat



4. feladat



5. feladat

a) MFM b) 01100 c) 2x