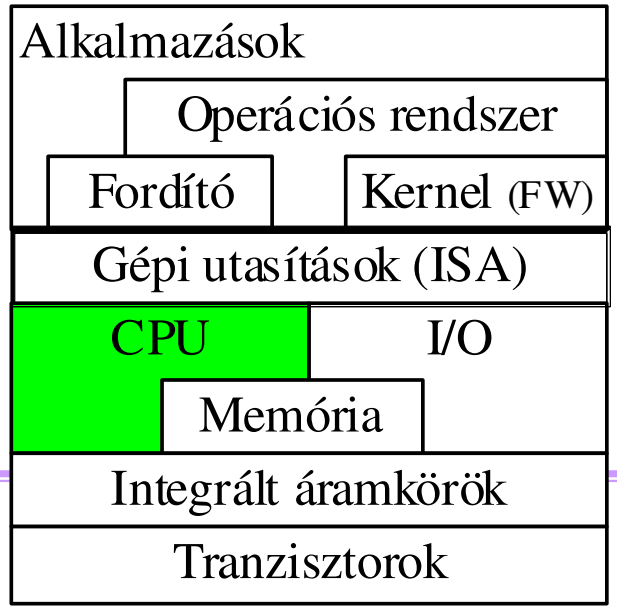
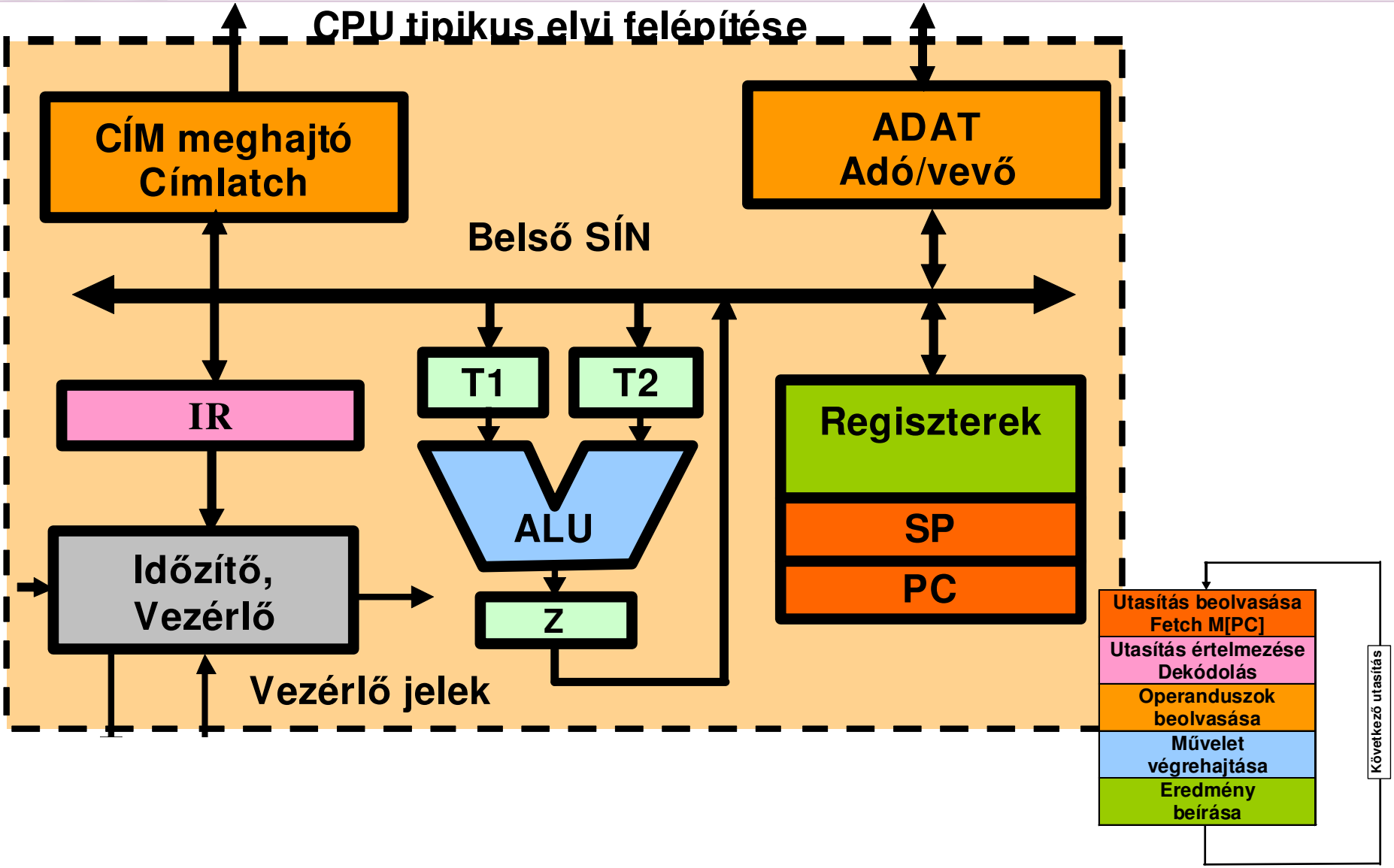
INFORMATIKA I.

BMEVIIIAB08

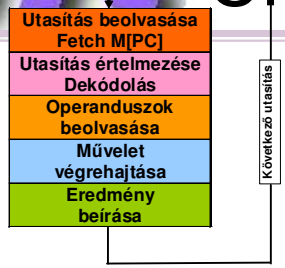
CPU (ALU - Adatút – Vezérlés)

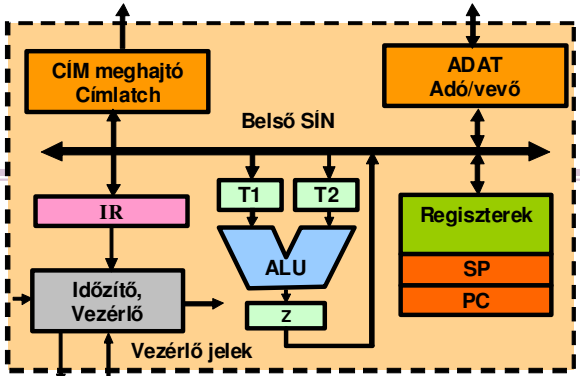


CPU-k szervezési kérdései



CPU-k szervezési kérdései





Feladat

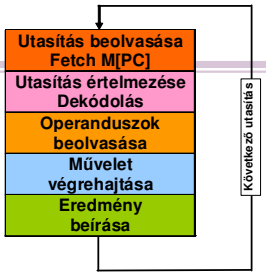
- CPU - memória forgalom
  - Utasítások beolvasása
  - Operandus/ok/ olvasása
- Utasítás értelmezés
  - Operandus címszámítás
- Utasítás végrehajtás
- Eredmény írása
- CPU - periféria forgalom

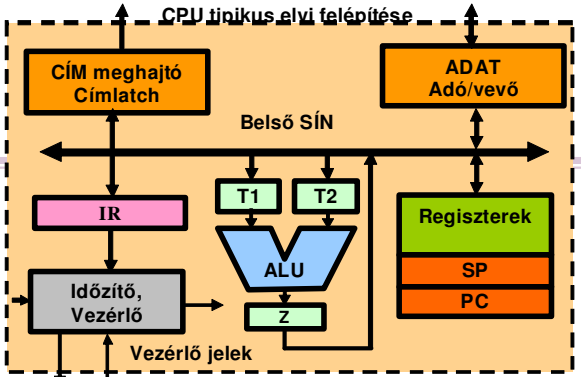
- Megszakítások kezelése

Részegység /modul/

- SÍN interfész
  - PC, CÍML, ADATa/v
  - Rx,SP, CÍML, ADATa/v
- Utasításdekóder,IR, /SÍN/
- ALU/Címszámító egység
- ALU, regiszterek
- Regiszterek,SÍN interfész
- Periféria sínvezérlő
  - /memóriába ágyazott I/O kezelésnél nincs/
- megszakítás logika ***/vezérlő/***

CPU-k szervezési kérdései





Feladat

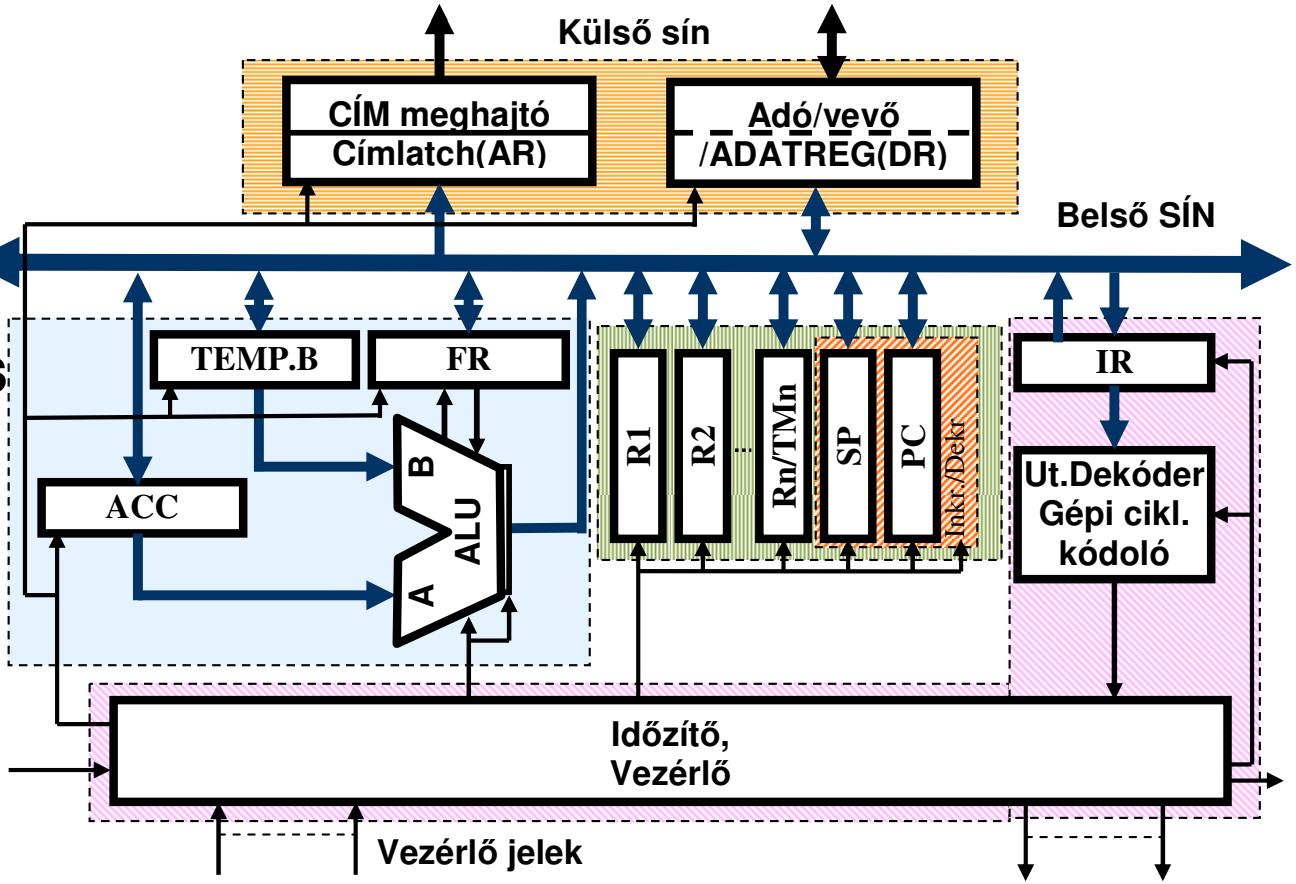
- Részfeladat szekvenciák, adatút/ak/ vezérlése
- Síninterfész vezérlés
  - Sínvezérlés átadás
  - Társprocesszor utasítás
- Memória elérés
  - Kapacitás
    - Virtuális tárkezelés
  - Sebesség

Részegység /modul/

- Vezérlő
- Vezérlő
  - Sínvezérő ***Kérő/elengedő-Arbiter***
  - Társprocesszor (C*oprocesszor)*
  - Virtuális tárkezelő ***/MMU/***
  - Gyorsítótár ***/Cache/***

Utasításkészlet és a belső felépítés egymásra hatása

- ALU szervezés
- Címszámítás
- Sínszervezés
- Regiszter szervezés
- Vezérlés

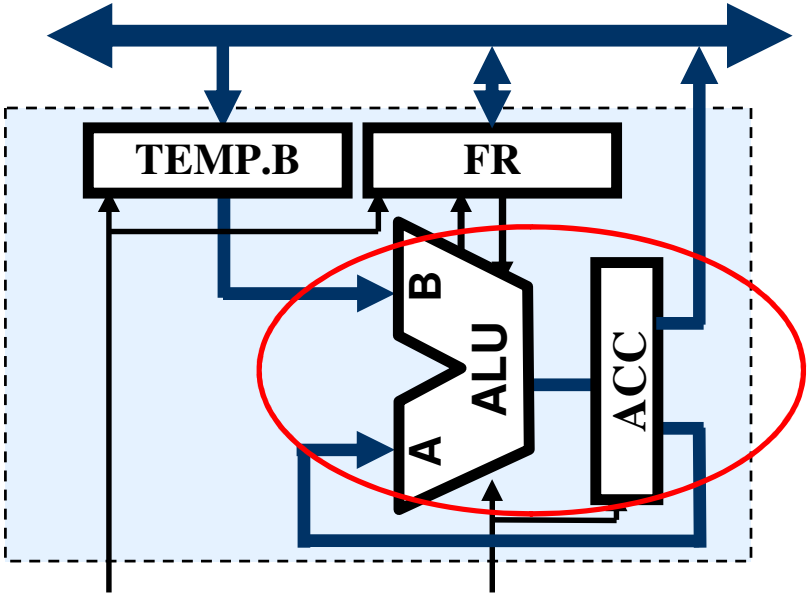


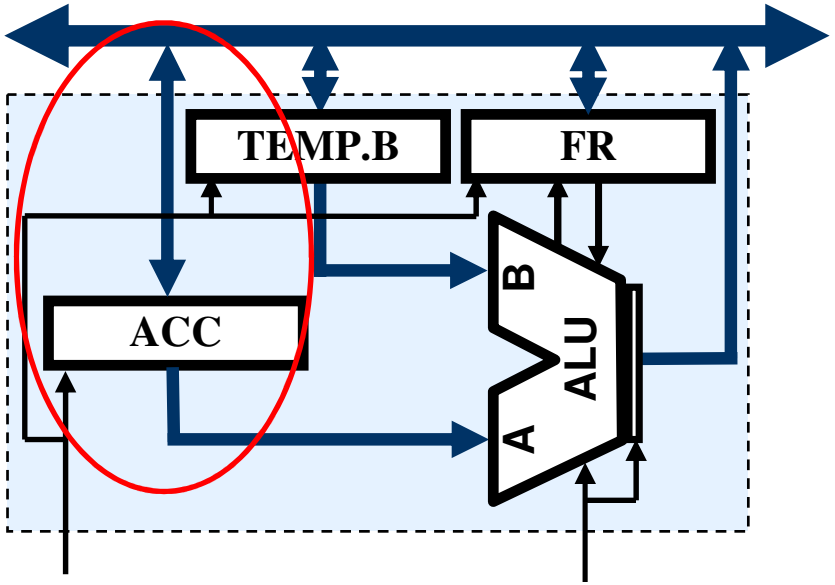
1. Egysínes „ACC központú” CPU tipikus felépítése

ALU szervezés /Operandusok helye?/

Egycímes */Egy operandus az ACC-ben /*

- *Egyszerűbb vezérlés*
- *Egyszerűbb utasításkészlet*
- *Több utasítás*

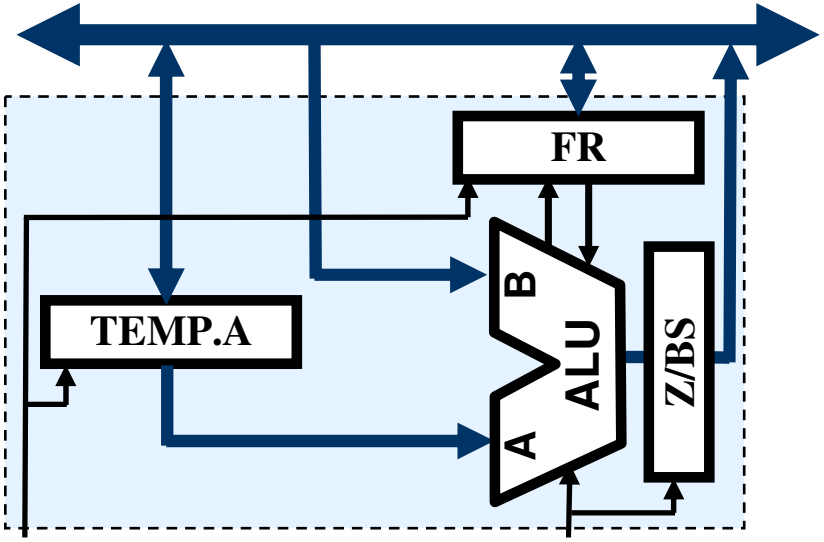




- **2a. Hagyományos ACC**
  - ***ALU⇔ACC kapcsolat***
  - ***Több elemi lépés***
- **2b. Univerzális ACC**
  - ***ACC töltése egyszerűbb***
  - ***ALU⇨ACC csak sínen***

- Másfél címes

/Egy operandus regiszterben/

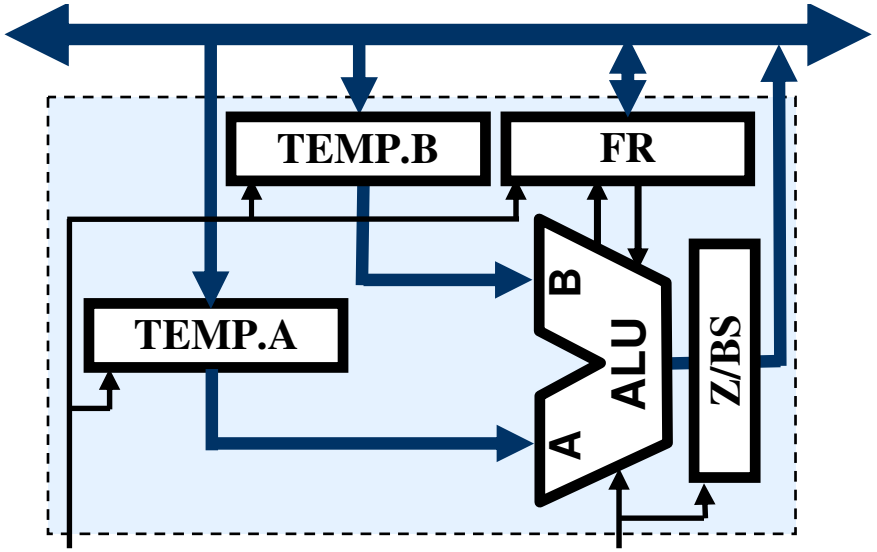


2c. Regiszter centrikus

- ***Nincs ACC, hatékonyabb ALU szervezés*** 👍
- ***Kevesebb Reg.mentés*** 👍

- Kétcímes

/Operandusok tetszőleges helyen/

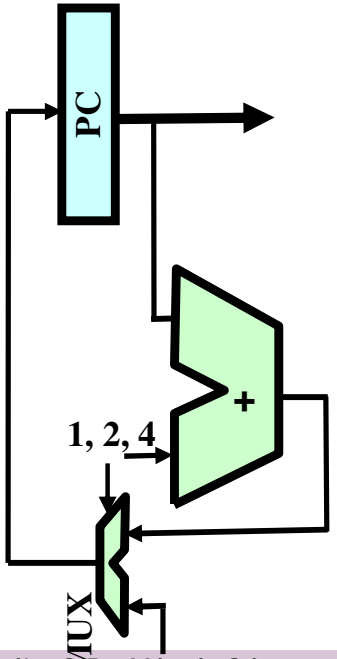


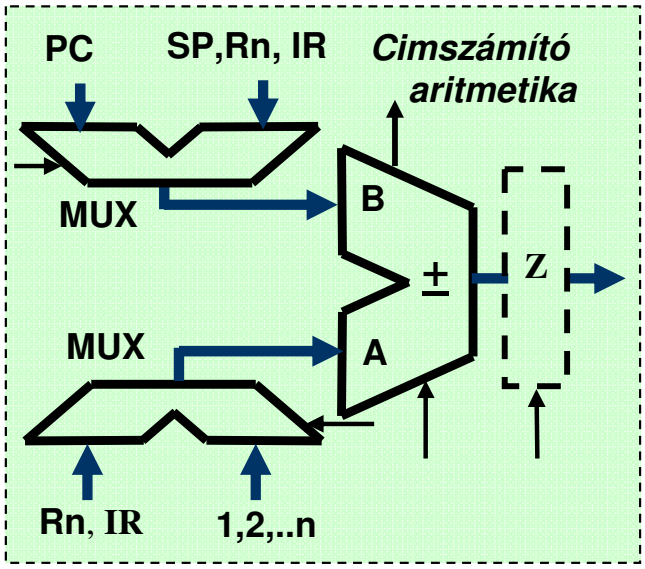
2d. Általános */BS barrel shifter/*

- ***Kevesebb utasítás*** 👍
- ***Több memória elérés*** 👎

Címszámítás kérdése

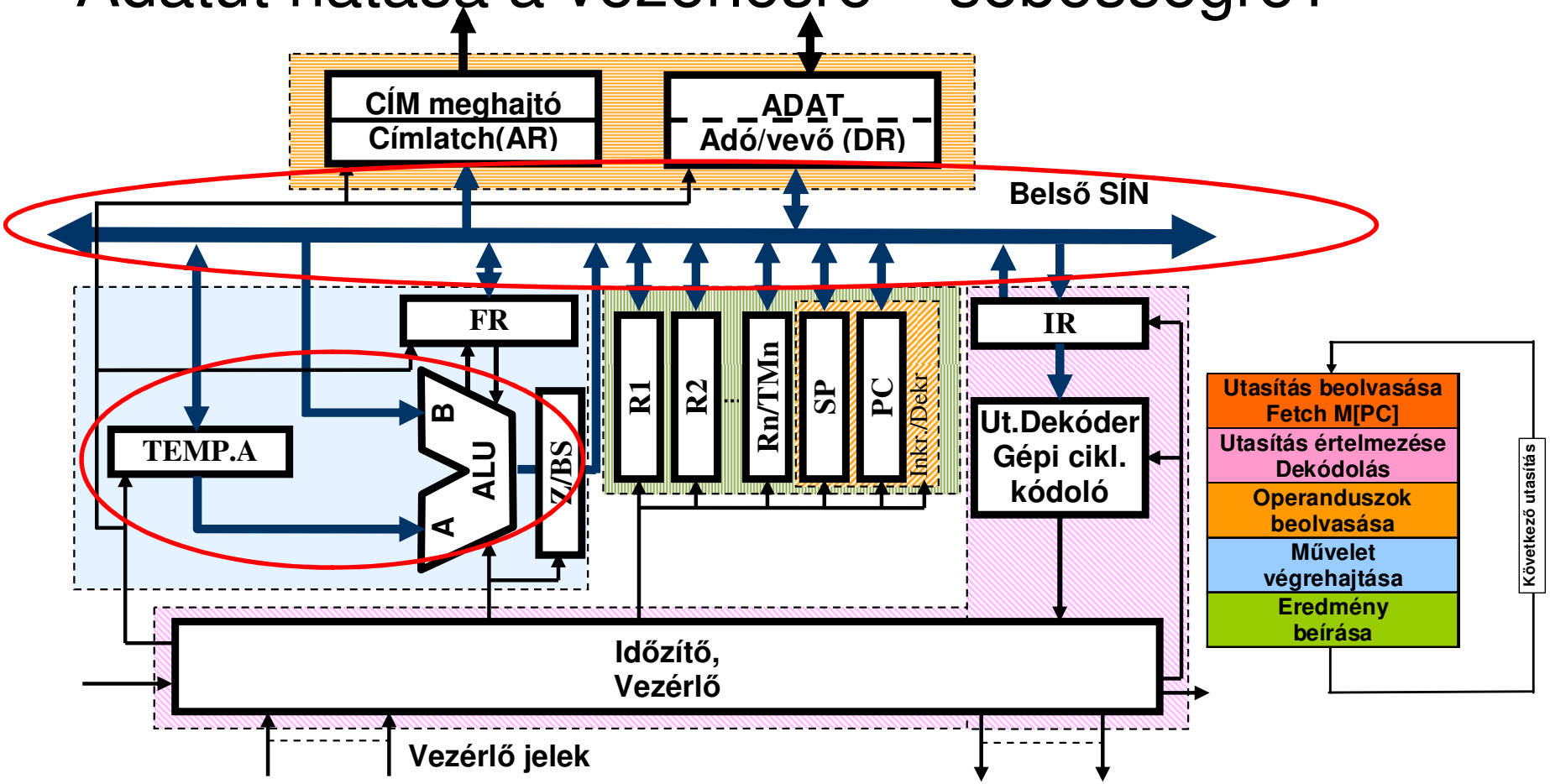
- ALU felhasználása
  - Átmeneti – munkaregiszterek alkalmazása /ACC esetén/
- Külön címszámító mű
  - Egyszerű inkrementer/dekrementer
    - PC, SP esetén
- Összetett címaritmetika /CISC/





Belső adatút szervezés hatása

- Adatút hatása a vezérlésre – sebességre?

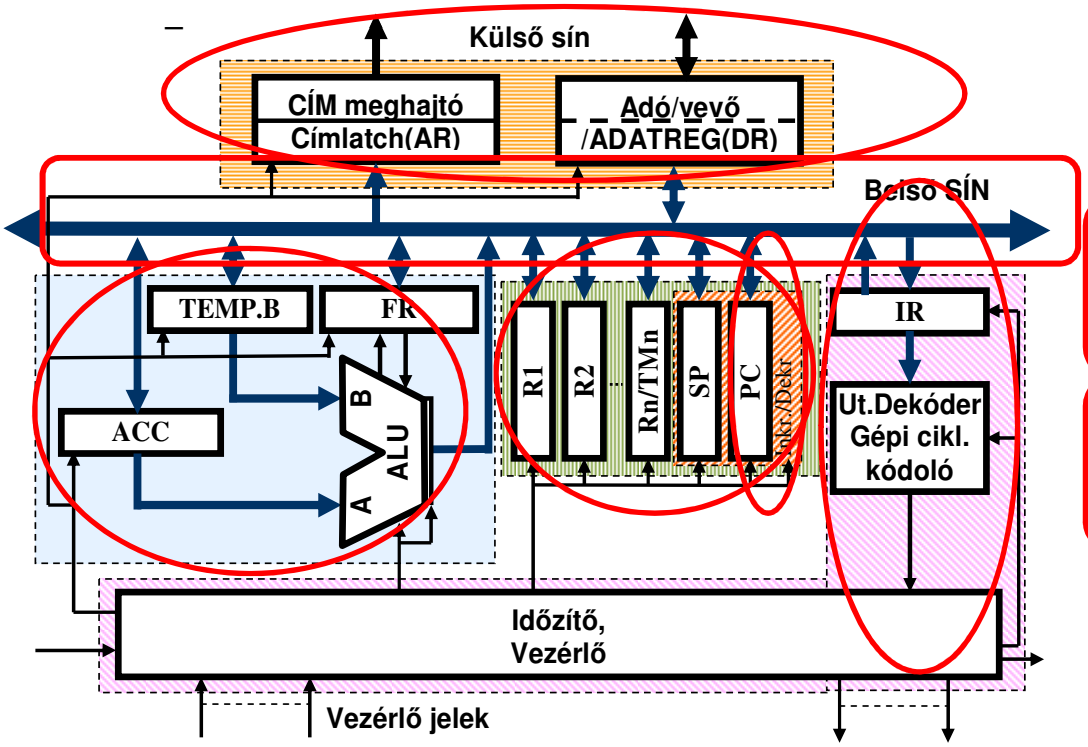


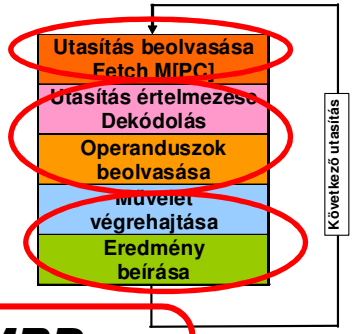
3. Egysínes „Regiszter központú” CPU tipikus felépítése

Adatút hatása a vezérlésre

Az utasítás végrehajtásához szükséges elemi tevékenységek

- *SUB (R1),R2 végrehajtása különböző adatút esetén*
  - *SUB (R1) A=A-(R1) /MOVE R2, A SUB (R1) MOVE A,R2/*
    - /egy címes utasításkészlet, egyik operandus az ACC-ben, eredmény szintén ACC-be kerül/
- SUB (R1) /Elemi lépések *1.ábra szerint/*
  -





1. PCki, ARbe, MRD,
/PC+n külön, n=1,2,4/
2. PCbe, vár MReady
3. DRki, IRbe,
4. R1ki, ARbe, MRD
5. DRki, TEMP.Rbe, vár MReady*
6.SUB(A-B), ALUki, ACCbe,
vége

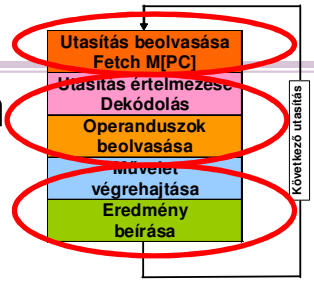
- ***Csak látszólag kevesebb***

1. Egysínes „ACC központú” CPU tipikus felépítése

Adatút hatása a vezérlésre - Az utasítás végrehajtásához szükséges elemi tevékenységek

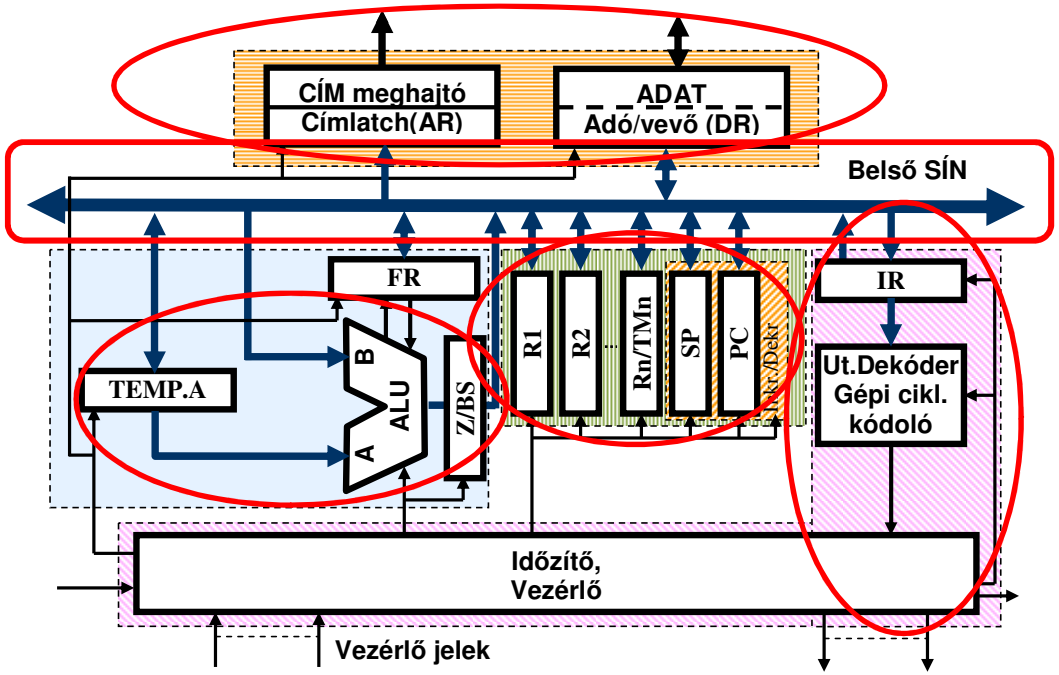
SUB (R1),R2 végrehajtása különböző adatút esetén

- *SUB (R1) A=A-(R1) /MOVE R2, A SUB (R1) MOVE A,R2/*
  - /egy címes utasításkészlet, egyik operandus az ACC-ben , eredmény szintén ACC-be kerül/



- SUB (R1),R2 /Elemi lépések *3.ábra szerint*/*

  /Feltételezzük SUB (R1),R2 >> R2=R2-(R1) ,Motorola konvenció/



3. Egysínes „Regiszter központú" CPU tipikus felépítése

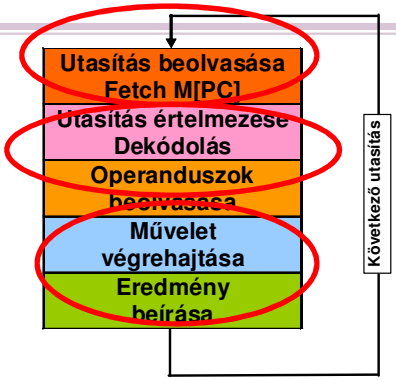
1. **PCki, ARbe, MRD,**
   ALU /F=B+1/*, Zbe
2. **Zki ,PCbe, vár MReady***
3. **DRki, IRbe,**
4. **R1ki, ARbe, MRD**
5. **x DRki, TEMP.Abe, vár MReady**
6. **x R2ki, SUB(B-A), Zbe**
7. **Zki, R2be,**
   vége

* PC+n-t *az ALU számítja*

x 5,6 –ban R2ki, DRki felcserélhető akkor SUB (A-B)

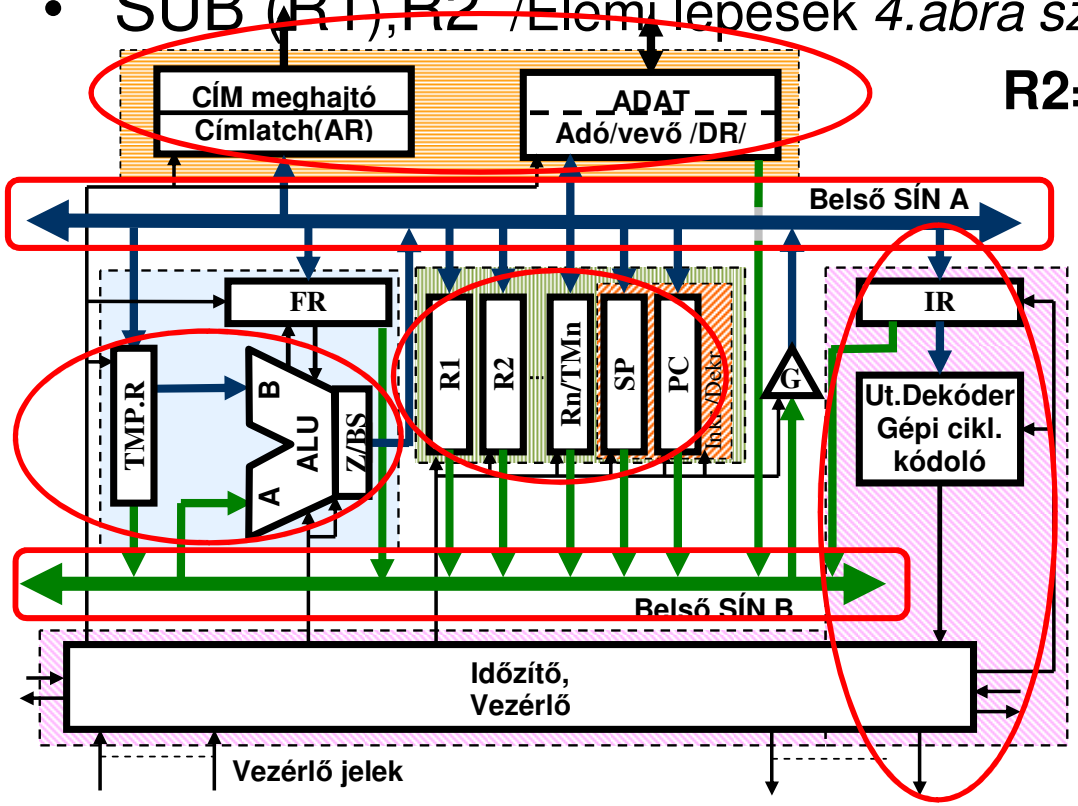
Adatút hatása a vezérlésre - Az utasítás végrehajtásához szükséges elemi tevékenységek

- *SUB (R1),R2 végrehajtása különböző adatút esetén*
  - *SUB (R1) A=A-(R1) /MOVE R2, A SUB (R1) MOVE A,R2/*
    - /egy címes utasításkészlet, egyik operandus az ACC-ben, eredmény szintén ACC-be kerül/



- SUB (R1),R2 /Elemi lépések *4.ábra szerint*/*

R2=R2-(R1)



1 PCki, G_e, ARbe, MRD, ALU/F=A+1/*, Zbe
2 PCbe, vár MReady
3 DRki, G_e, IRbe,
4 R1ki, G_e, ARbe, MRD
5 x R2ki, G_e,TMP.Rbe, vár MReady
6 x DRki,SUB(B-A), Zbe, Zki, R2be, vége

* PC+n-t *az ALU számítja*

x 5,6 –ban R2ki, DRki felcserélhető akkor SUB (A-B)

4. Kétsínes „Regiszter központú" CPU tipikus felépítése

Adatút hatása a vezérlésre - Az utasítás végrehajtásához szükséges elemi tevékenységek

- *Az utasítás végrehajtásához szükséges elemi tevékenységek*

❑ *SUB (R1),R2 végrehajtása különböző adatút esetén*

❑ *SUB (R1) A=A-(R1) /MOVE R2, A SUB (R1) MOVE A,R2/*

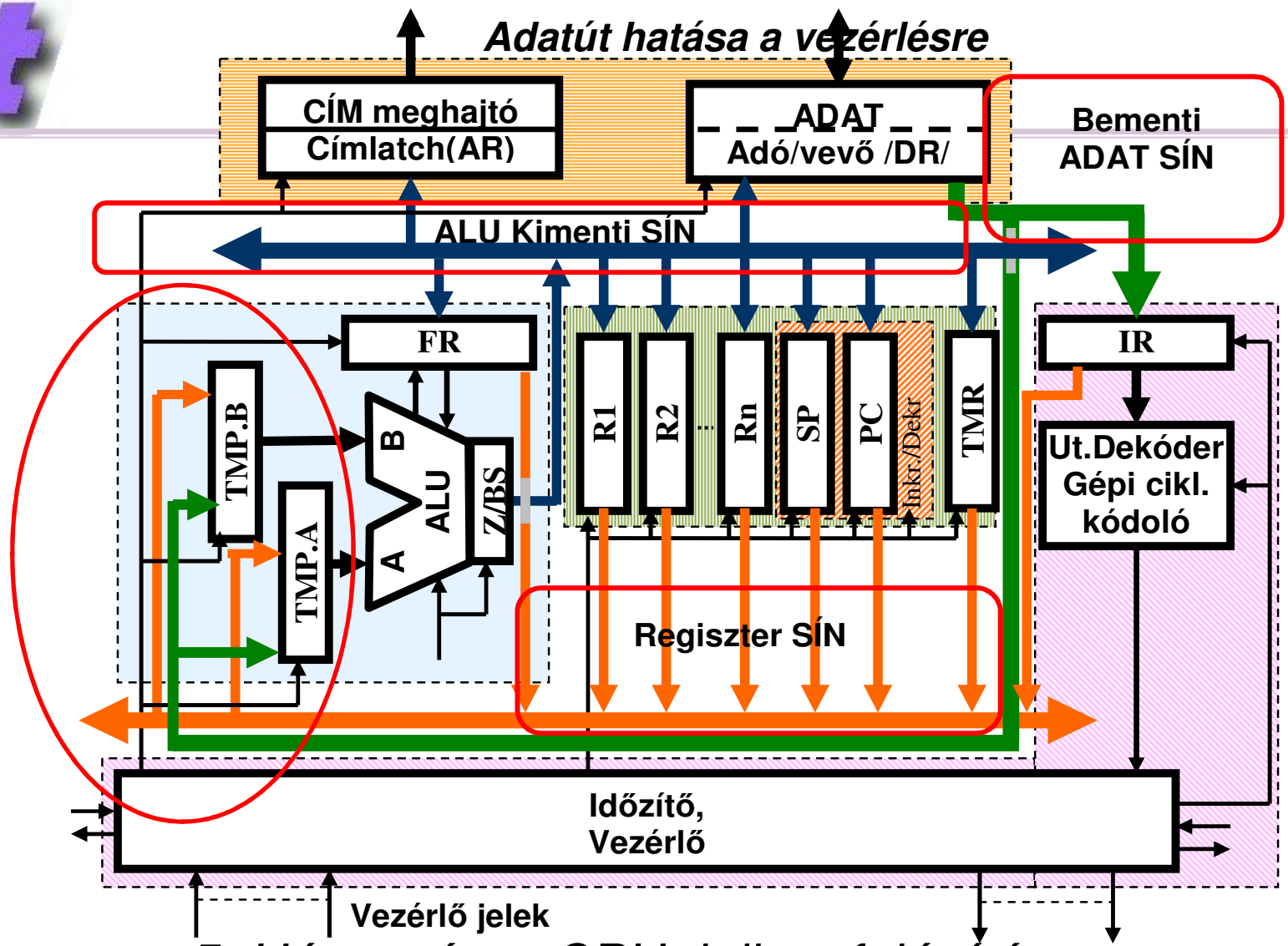
– /egy címes utasításkészlet, egyik operandus az ACC-ben, eredmény szintén ACC-be kerül/

SUB (R1) /1. szerint/	**SUB R2 (R1) /3.szerint*/** 👍	**SUB R2 (R1) /4. szerint*/** 👍 👍
1 PCki, ARbe, MRD, /PC+n külön, n=1,2,4/	1 PCki, ARbe, MRD, ALU /F=B+1/*, Zbe	1 PCki, G_e, ARbe, MRD, ALU/F=A+1/*, Zbe
2 PCbe, vár MReady	2 Zki ,PCbe, vár MReady*	2 PCbe, vár MReady
3 DRki, IRbe,	3 DRki, IRbe,	3 DRki, G_e, IRbe,
4 R1ki, ARbe, MRD	4 R1ki, ARbe, MRD	4 R1ki, G_e, ARbe, MRD
5 DRki, TEMP.Rbe, vár MReady	5 x DRki, TEMP.Abe, vár MReady	5 xR2ki, Ge,TMP.Rbe, vár MReady
6 SUB(A-B), ALUki, ACCbe,	6 x R2ki, SUB(B-A), Zbe	6 xDRki,SUB(B-A), Zbe, Zki, R2be,
vége	7 Zki, R2be,	vége
Ez csak látszólag kevesebb!	vége	

Megjegyzés: *3,4. esetén PC+n-t az ALU számítja

x 5,6 –ban R2ki, DRki felcserélhető, akkor SUB (A-B)

/Feltételezzük SUB (R1),R2 >> R2=R2-(R1) , Motorola konvenció/

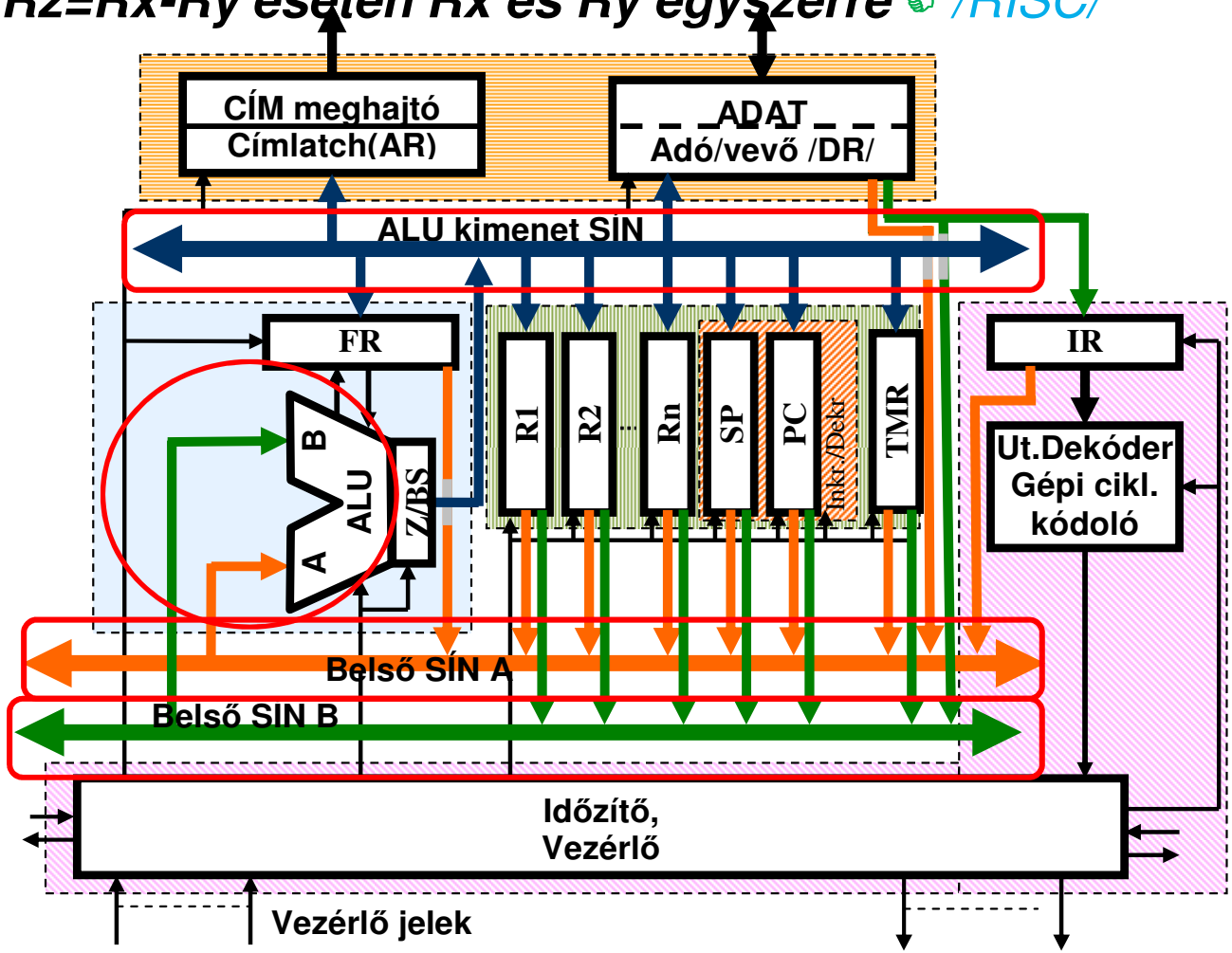


5. Háromsínes CPU tipikus felépítése

- *Két operandus (Rx) és Ry egyszerre*

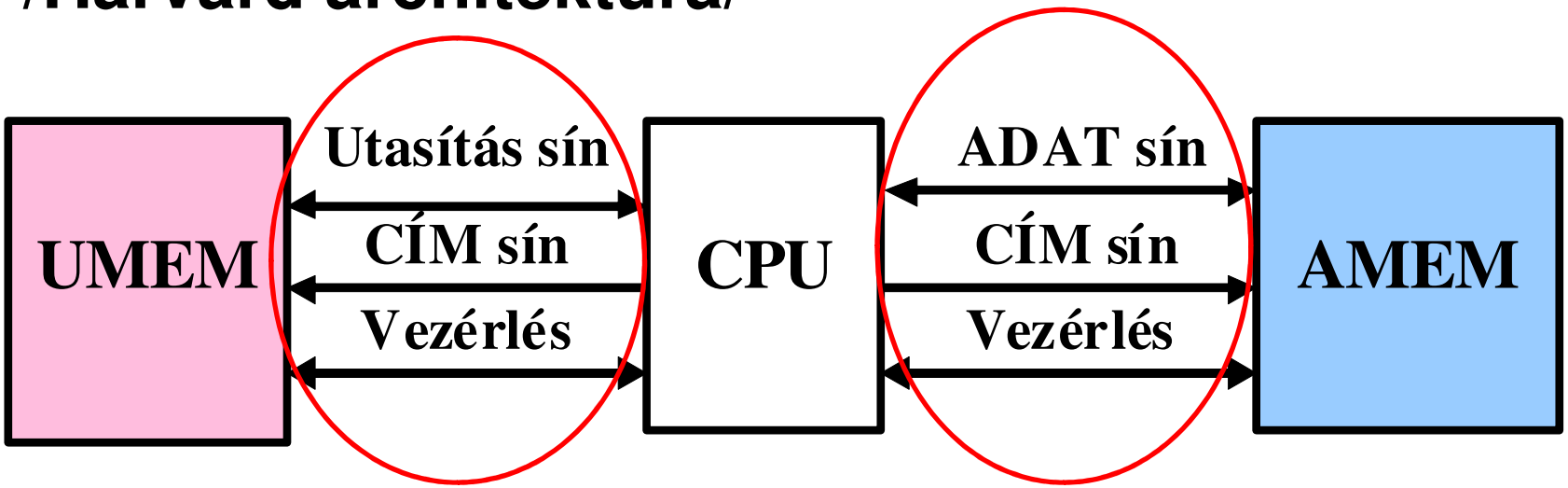
„Regiszter központú” adatút szervezés

- ***Rz=Rx-Ry esetén Rx és Ry egyszerre*** 👍 */RISC/*



6.Háromsínes „Regiszter központú” CPU tipikus felépítése

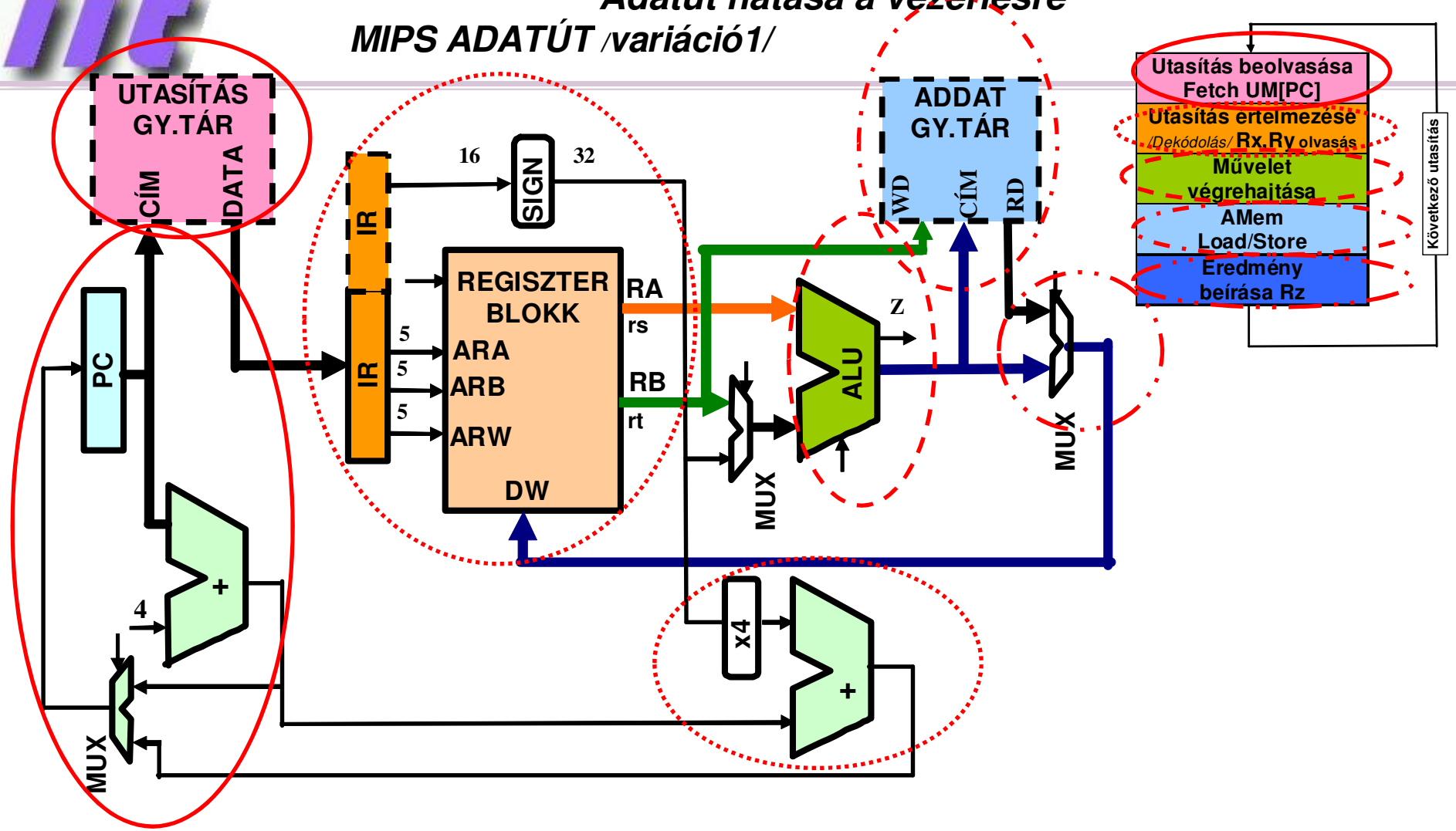
Memória - utasítás/adat-út szétválasztása /Harvard architektúra/



- **Párhuzamos elérés**
- **Neumann elv?**
  - ***Sok esetben csak a CPU-n belül***
    - ***UMEM-AMEM szétválasztott belső gyorstár***

Adatút hatása a vezérlésre

MIPS ADATÚT /variáció1/



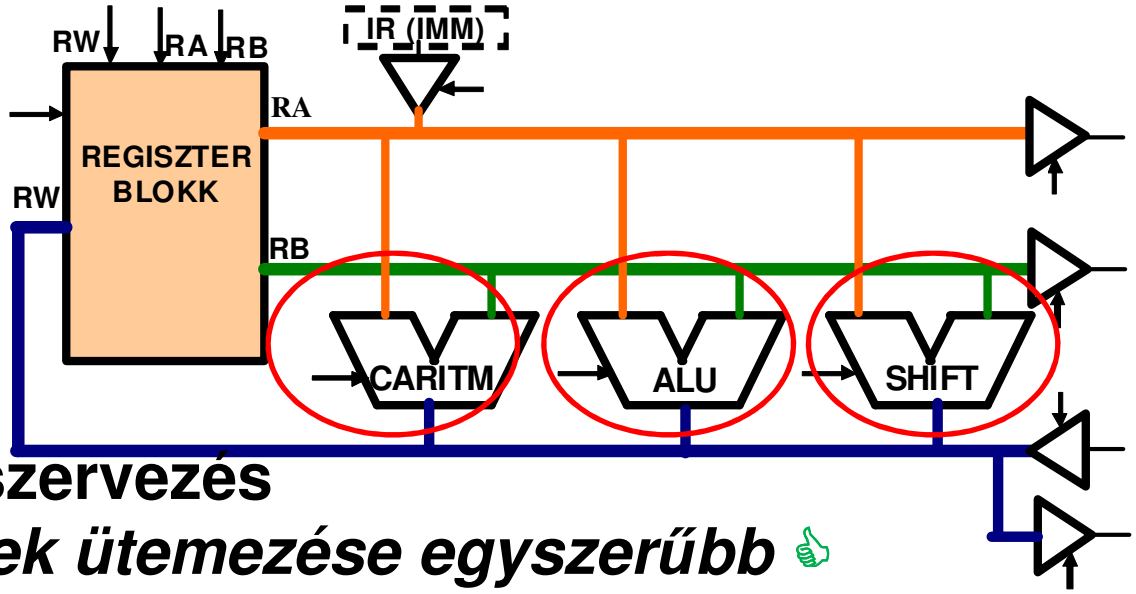
- *Egy kis feladat megoldás: Melyik utasítás típus nem hajtható végre? Mi a hiba? Hogyan javítható?*

  [David A. Petterson and John L. Henessy: Computer organization and design]

Egyciklusú szervezés

- ❑ *Egyszerűbb ütemezés* 👍
  /de a leghosszabb műveletre kell méretezni/ 👎
  - ❑ *Több külön részegység* 👍
    /pl.:Külön aritmetikák, memóriák párhuzamosan működhetnek/

HÁROMSÍNES REGISZTER BLOKK ÖNÁLLÓ MŰVELETVÉGZŐKKEL



- ➢ **Többciklusú szervezés**
  - ❑ ***Elemi lépések ütemezése egyszerűbb*** 👍
    - ❑ *Változó lépésszámot igénylő utasításoknál előnyös*
  - ❑ ***Több ciklus lassíthat***
    - ❑ ***de előnyös is lehet /lsd.pipe-line/*** 👍👍

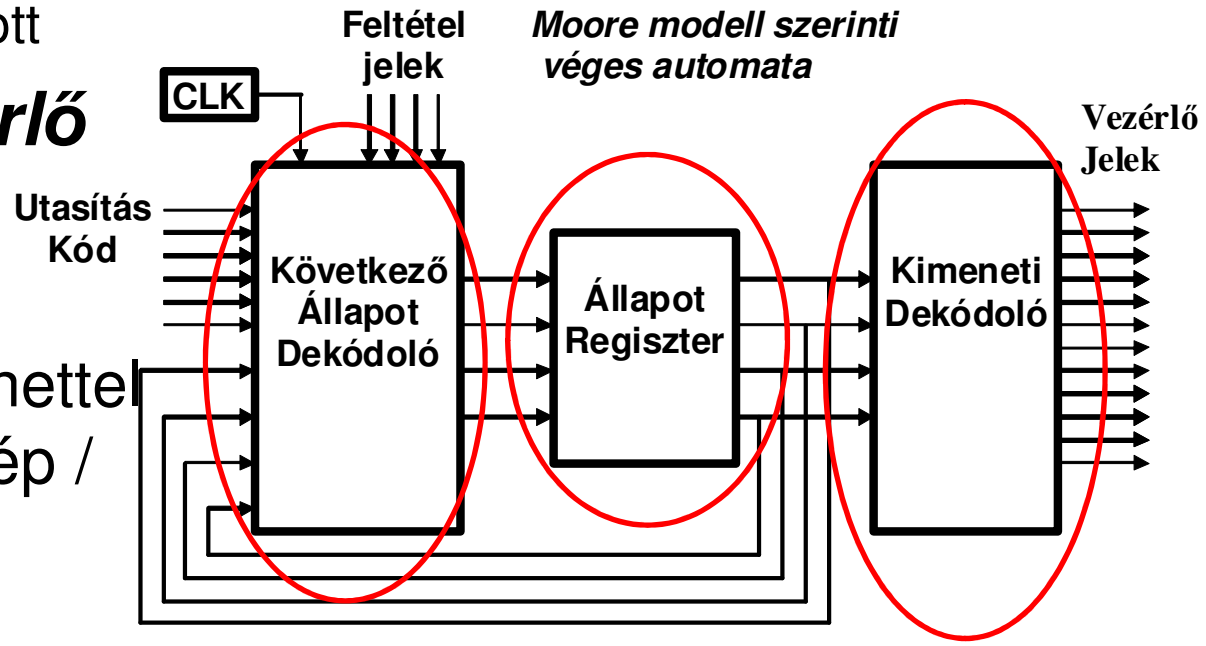
Vezérlőegység

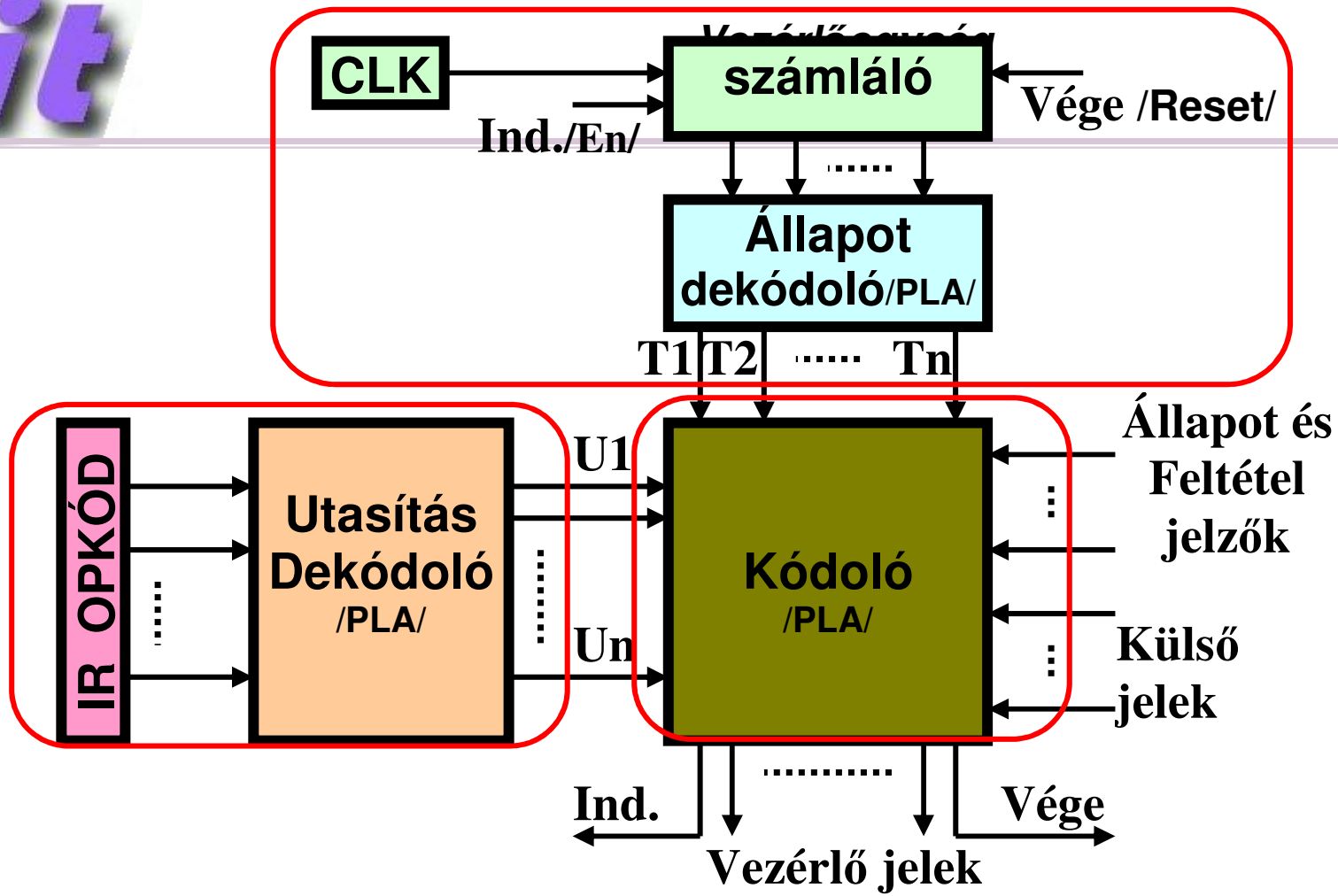
- Az utasítás végrehajtás folyamatának vezérlése
- Az adatút/ak/ vezérlése
- Külső és belső feltételek /állapotjelzők/ figyelése
- Vezérlőjelek előállítása
  - Huzalozott
  - Mikroprogramozott

❖ ***Huzalozott vezérlő egység***

- Véges Automata

– /véges számú átmenettel rendelkező állapotgép /

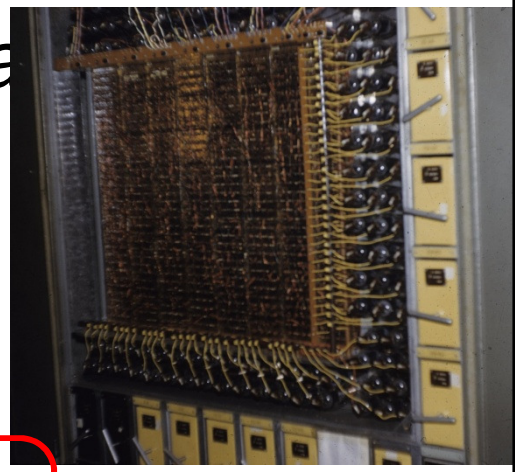


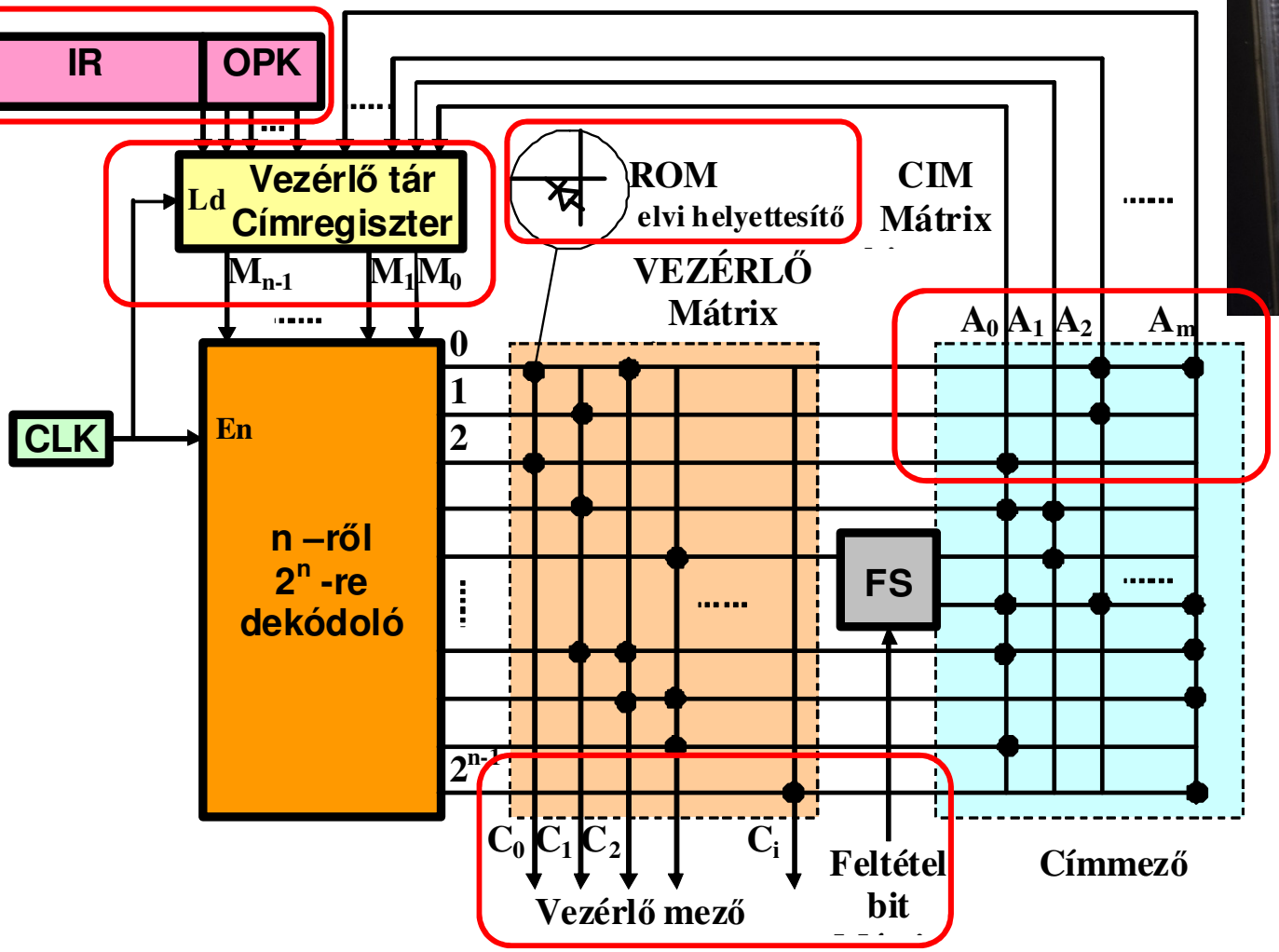


- *Leggyorsabb működés* 👍
  /Pontosan az adott feladatra optimalizálva/
- *Bonyolult, nehezebb tervezés, nehezebb tesztelés* 👎
- *Nem változtatható* 👎

Mikroprogramozott vezérlőegység

- *WILKES modell /1951 ferrit memória*

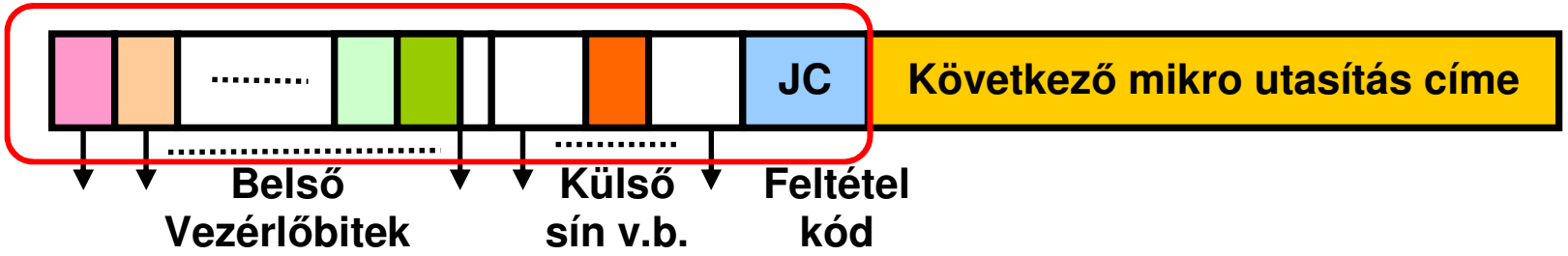




Mikroprogramozott vezérlőegység

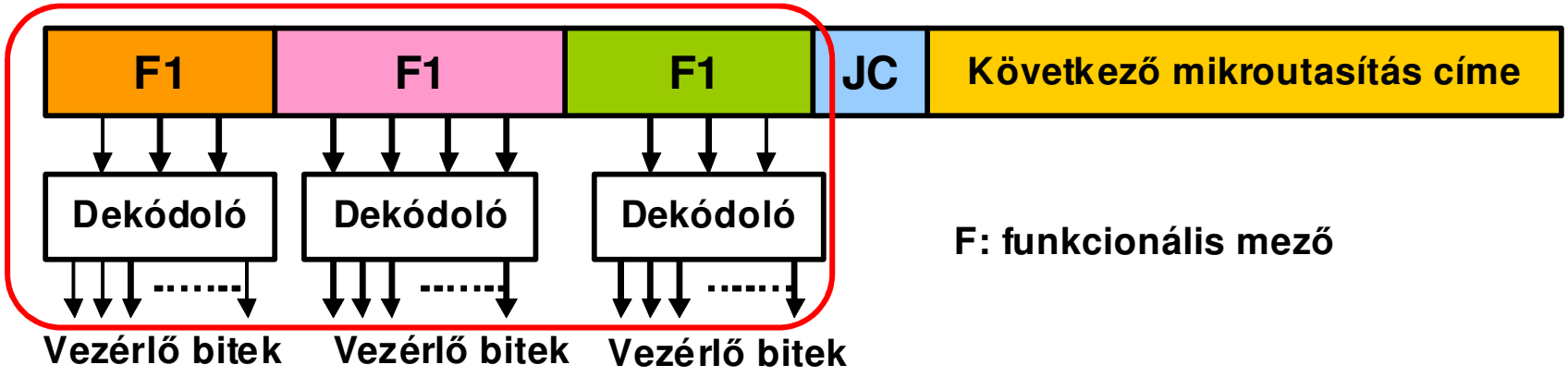
- *Egyszerűbb tervezés /szisztematikus tervezés/*
- *Flexibilis felépítés /könnyen módosítható/*
- *Könnyű emulálni meglévő utasításkészletet*
- *Könnyű összetett utasítást megvalósítani /mi van a sebességgel?!!*
- *Több hardvert igényel /pl.:mikroprogram tároló /*
- *Lassabb működés /a tároló elérés miatt/*
- Tervezési szempontok
  - A lehető legtöbb műveletet párhuzamosan hajtsa végre
  - vezérlő információ közvetlen, vagy kódolt
  - A következő mikroutasítás címe
    /feltételes utasítás esetén két cím?!/

Horizontális mikroutasítás formátum



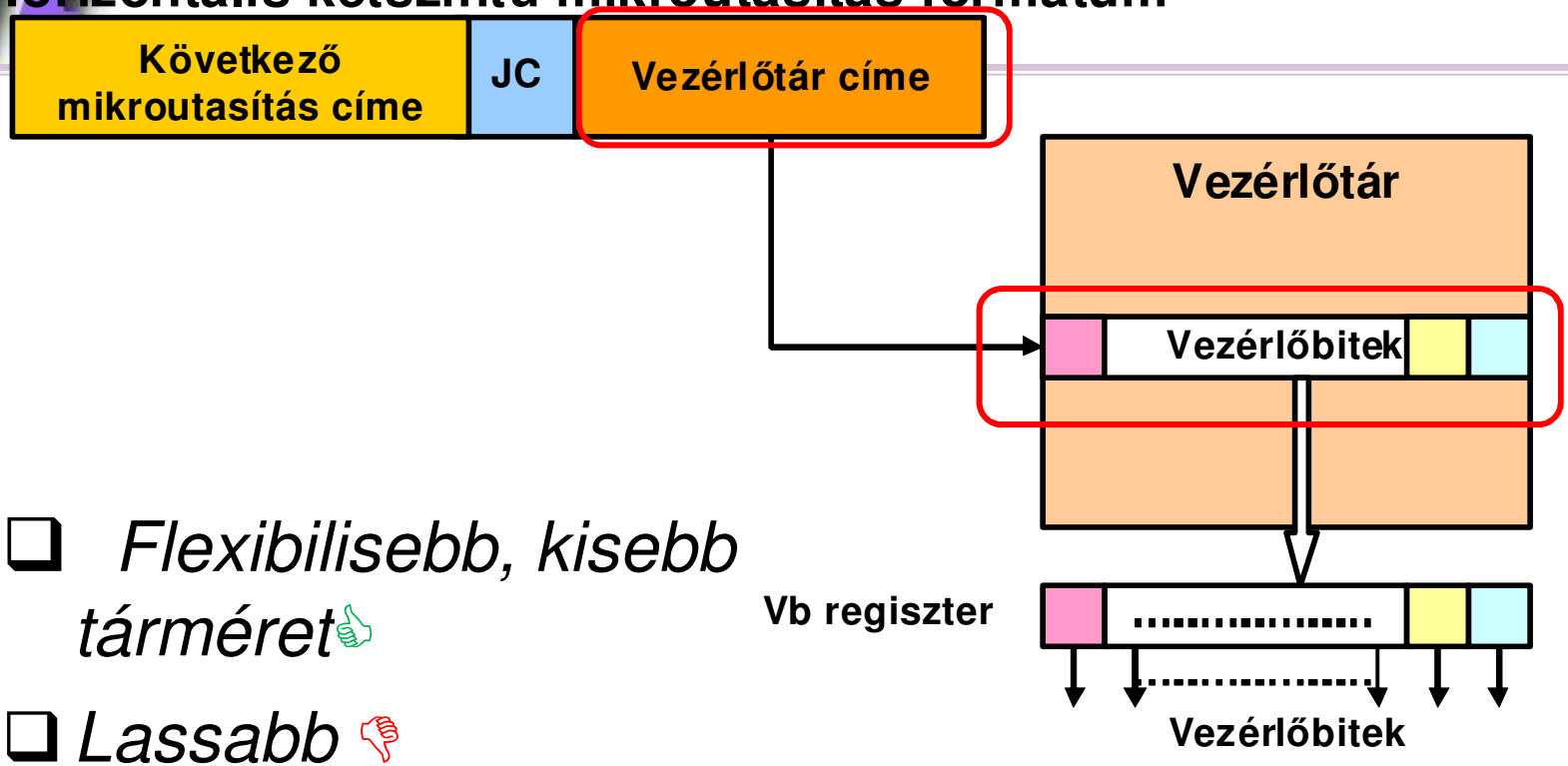
- *Nagymértékű párhuzamosság, gyors*
- *Nagy tárméret*

Mezőnként kódolt horizontális mikroutasítás formátum



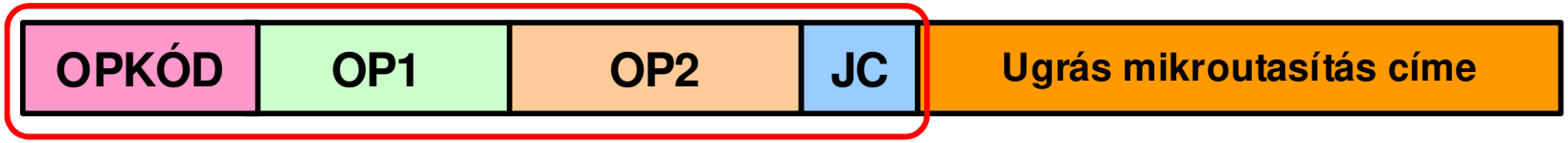
- ***Jó kompromisszum***

Horizontális kétszintű mikroutasítás formátum



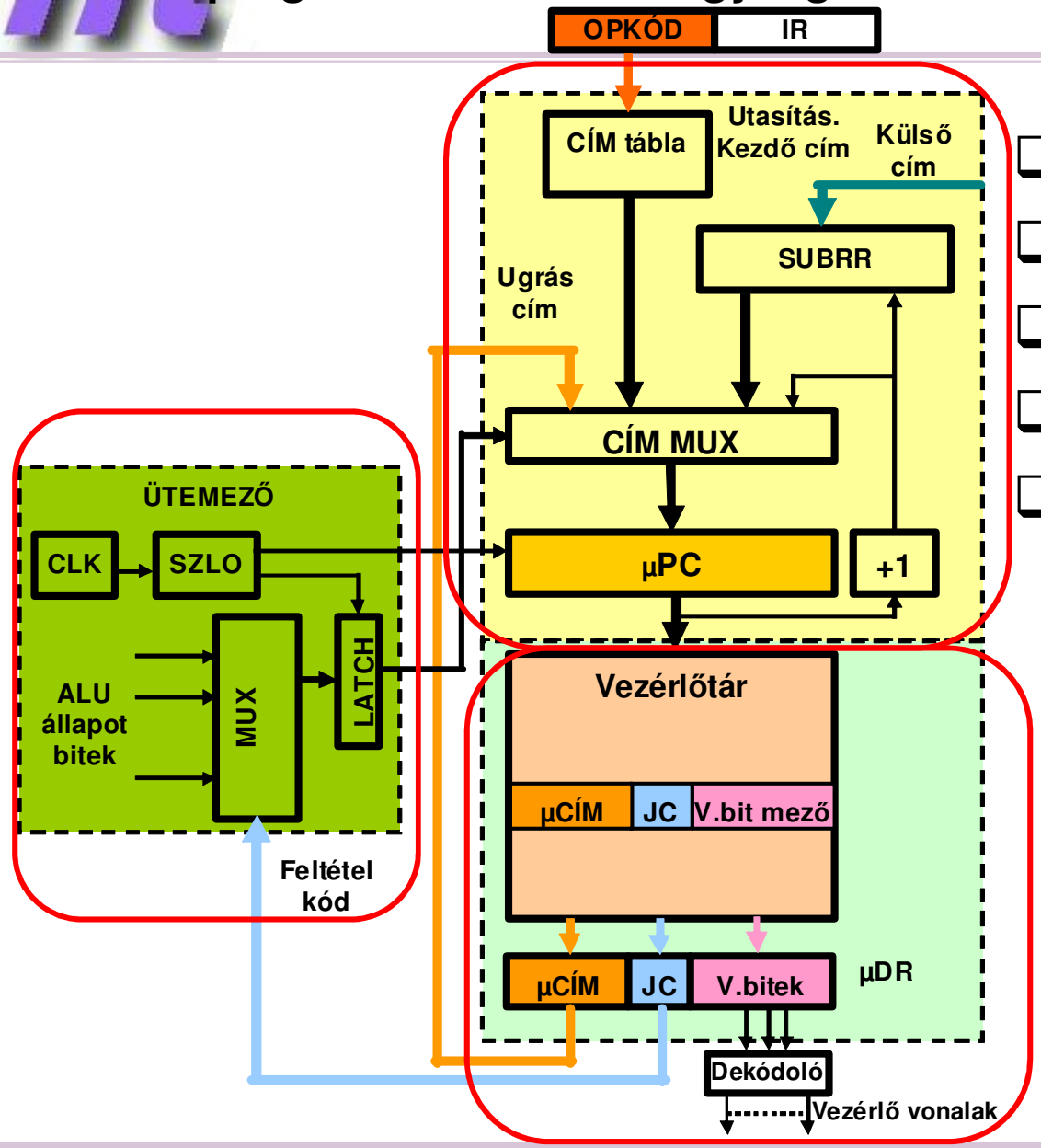
- *Flexibilisebb, kisebb tárméret*
- *Lassabb*

Vertikális mikroutasítás formátum /néhány vagy esetleg egy elemi műveletre/



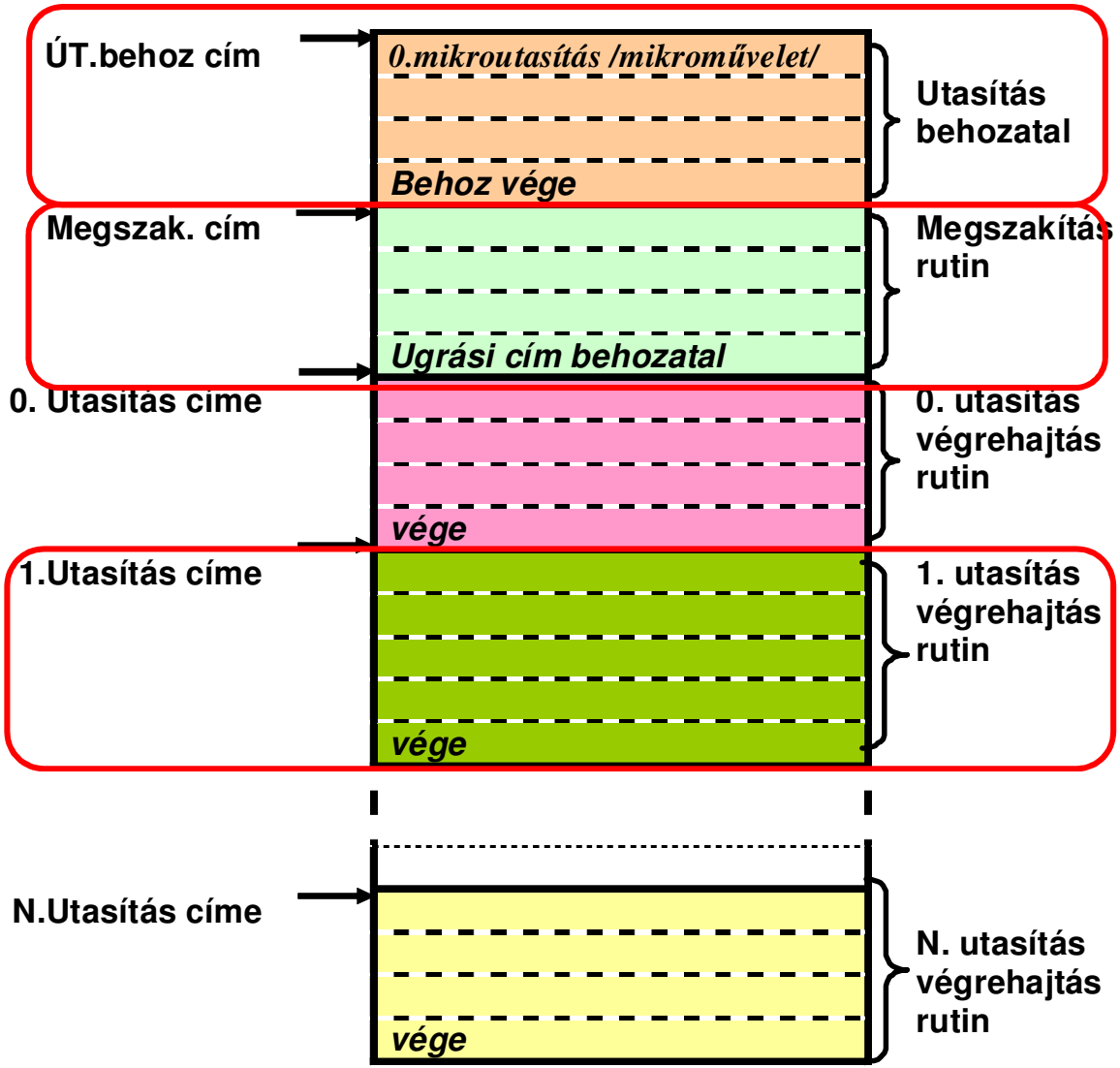
- ***Bonyolult utasítások mikro-rutinként***
- ***Mikroszintű programozott „CPU”***

Mikroprogramozott vezérlő egység blokkvázlat



- ***Flexibilisebb***
- ***Kisebb tárméret***
- ***Nagyobb Hw***
- ***Lassabb***
- ***Kisebb párhuzamosság***

Mikroprogramtár szervezése



Példák:

- *IBM 360/50 90bit mezőnként kódolt horizontális*
- *PDP11 128 bit horizontális*
- *M6800 68 bit kétszintű horizontális*
- *AMD 2900 bitszelet mikrovezérlő IC-k*
- *X86 Vertikális*

Összefoglalás

- CPU-k szervezési kérdései
- ALU szervezés
- Címszámítás
- Belső adatút szervezés hatása a teljesítményre, vezérlésre
- Utasítás és adatút szétválasztása,Harvard architektúra
- Huzalozott vezérlő egység
- Mikroprogramozott vezérlőegység
- Horizontális és vertikális mikroutasítások