

**Digitális rendszerek III. (GEVAU505B) c. tantárgy**  
előadásának ütemterve  
Villamosmérnöki (BSc) Alapszak  
G2BV1, G2BV2 tanulókörök számára

<b>Naptári hét</b>	<b>Előadás</b>
37.	Bevezetés a mikroprocesszorteknikába: CPU felépítés, sínrendszer
38.	Neumann, Harvard architektúra, címdekódolás
39.	Tipikus $\mu$ P műveletek: MR, MW, I/OR, I/OW, INT. A mikroprocesszorok utasításkészlete I: aritmetikai utasítások
40.	A mikroprocesszorok utasításkészlete II: logikai utasítások, III: vezérlésátadó utasítások
41.	Szubrutin hívás, megszakításkezelés lefolytatása
42.	Párhuzamos I/O-k felépítése, programozása
43.	Soros I/O-k felépítése, programozása
44.	ZH
45.	Időzítők, számlálók használata, időzítési feladatok programozása
46.	Az assembly nyelvű programozás szabályai, példák
47.	C nyelvű programozás sajátosságai mikrovezérlők esetén
48.	A RISC programok jellemzése, utasításlapolós üzem mód
49.	Szoftver fejlesztőeszközök használata, boot-loader, nyomkövetés
50.	Ismétlés

Miskolc, 2019. szeptember 9.

Dr. Trohák Attila  
intézet igazgató, egyetemi docens

Drótos Dániel  
tanszéki mérnök  
előadó

**Digitális rendszerek III. (GEVAU505B) c. tantárgy**  
gyakorlatának ütemterve  
Villamosmérnöki (BSc) Alapszak  
G2BV1, G2BV2 tanulókörök számára

<b>Naptári hét</b>	<b>Gyakorlat</b>
37.	C programnyelvű fejlesztés ismétlése
38.	C programozás ismétlése
39.	Fejlesztőeszközök bemutatása
40.	I. feladat: C nyelvű programozás
41.	Alkatrész átvétel
42.	II. feladat: EB134 mérés
43.	III. feladat: Oszcillátor mérés
44.	Mikrovezérlős kártya építése
45.	Mikrovezérlős kártya építése
46.	Mikrovezérlős kártya beüzemelése
47.	IV. feladat: mikrovezérlő programozás I.
48.	V. feladat: mikrovezérlő programozás II.
49.	Pótlás
50.	Pótlás

Miskolc, 2019. szeptember 9.

Dr. Trohák Attila  
intézet igazgató, egyetemi docens

Drótos Dániel  
tanszéki mérnök  
gyakorlatvezető

**Digitális rendszerek III. (GEVAU505B) c. tantárgy**  
követelményrendszere  
Villamosmérnöki (BSc) Alapszak  
G2BV1, G2BV2 tanulókörök számára

**Aláírás feltételei:**

- Legalább elégséges zárthelyi dolgozat (a félév során, illetve az aláírás pótlási időszakban pótolható).
- Gyakorlaton az 5 értékelt feladat legalább elégséges átlaggal való megoldása (a félév során, illetve az aláírás pótlási időszakban pótolható).
- Óralátogatás: legalább 4 gyakorlati feladat teljesítése a szorgalmi időszakban (igazolt hiányzás esetén a félév során, illetve az aláírás pótlási időszakban pótolható).

**Gyakorlat:** 5 feladat, amelyek 1-5 osztályzattal értékelték. A gyakorlat eredménye az 5 jegy átlaga.

**Félév értékelése:** Aláírás megszerzése után a kiadott kérdéssor alapján írásbeli vizsgát kell tenni. A félév teljesítéséhez legalább elégségest kell elérni a vizsgán. A féléves jegy

- a ZH (20%),
- a gyakorlat eredménye (40%),
- és a vizsga jegyének (40%)

súlyozott átlaga.

Miskolc, 2019. szeptember 9.

Dr. Trohák Attila  
intézet igazgató, egyetemi docens

Drótos Dániel  
tanszéki mérnök  
gyakorlatvezető

## ZH Digitális rendszerek 3

Neptun kód: \_\_\_\_\_ Név: \_\_\_\_\_

1. Mikroprocesszoros rendszer felépítése, elemei. \_\_\_\_\_
2. 2 irányú adatvezeték kialakítása TS kapuval. \_\_\_\_\_
3. READ busz művelet idődiagramja. \_\_\_\_\_
4. Címdekóder: 12 bites címbusz, 3 db 1024 byte-os eszköz. \_\_\_\_\_
5. Tervezzen assembly programot az R1 regiszter 2-ik bitjének a 0-ba állítására!  
Adja meg a használt utasítások funkcióját! \_\_\_\_\_
6. A=0x8B, R=0x41, C=0 esetén az ADD R utasítás eredménye (A, jelzőbitek). \_\_\_\_\_
7. Speciális célú regiszterek, felhasználásuk. \_\_\_\_\_
8. CPU működési algoritmus. \_\_\_\_\_
9. DMA működése. \_\_\_\_\_
10. Címzési módok. Milyen C nyelvű utasítás használhatja ezeket? \_\_\_\_\_

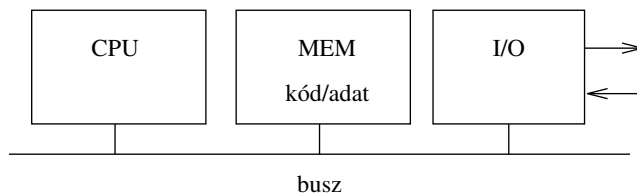
**1:** 0-4.5    **2:** 4.6-5.9    **3:** 6.0-7.4    **4:** 7.5-8.9    **5:** 9.0-10.0

Összesen: \_\_\_\_\_

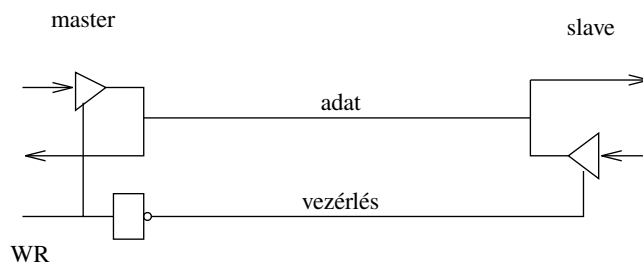
Eredmény: \_\_\_\_\_

## ZH megoldás minta

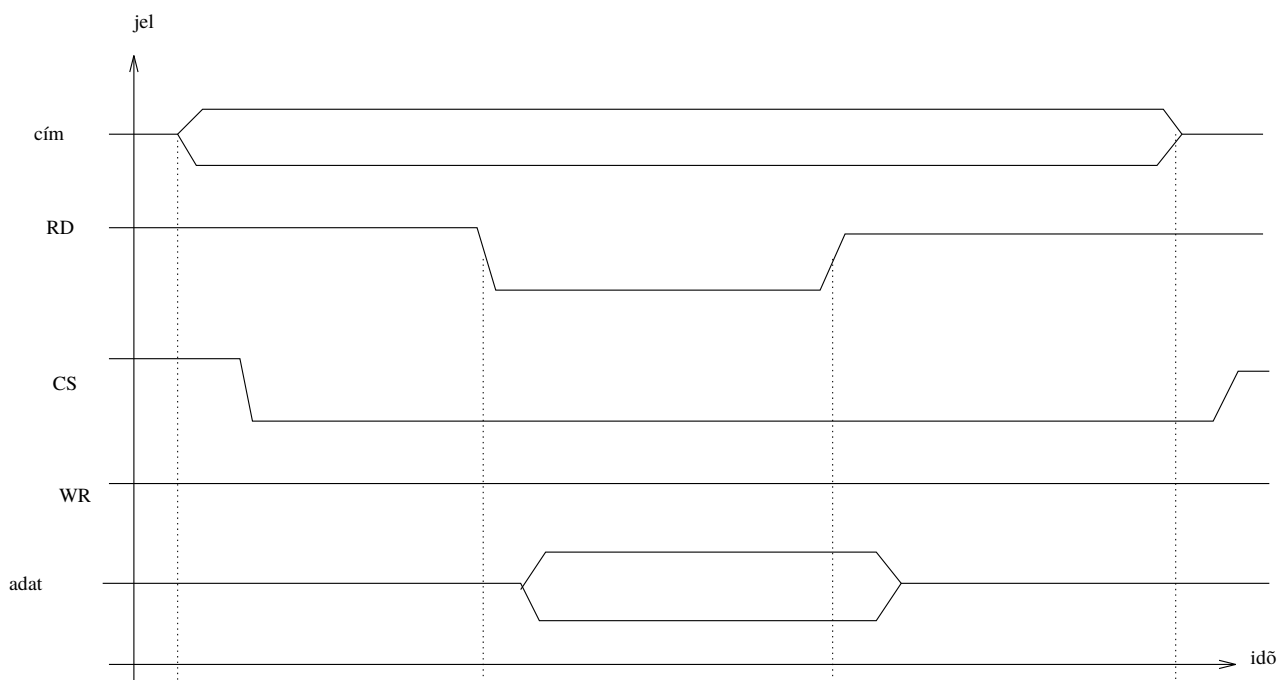
### 1. Mikroprocesszoros rendszer felépítése, elemei.



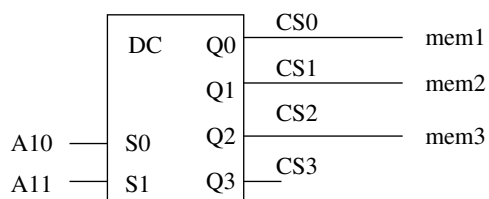
### 2. 2 irányú adatvezeték kialakítása TS kapuval.



### 3. READ busz művelet idődiagramja.



### 4. Címdekóder: 12 bites címbusz, 3 db 1024 byte-os eszköz.



### 5. Tervezzen assembly programot az R1 regiszter 2-ik bitjének a 0-ba állítására! Adja meg a használt utasítások funkcióját!

```
LD    R0, #4    ; mask, 2^2
NOT   R0        ; mask negálása
AND   R1, R1, R0 ; R1= R0 & ~mask
```

6. A=0x8B, R=0x41, C=0 esetén az ADD R utasítás eredménye (A, jelzőbitek).

```
Átvitel: 000000110
Adat A:  10001011
Adat R:  01000001
Eredmény: 11001100
Z=0, S=1, C=0, O=0
```

7. Speciális célú regiszterek, felhasználásuk.

IR: Instruction Register a FETCH művelettel beolvasott utasítást tárolja

PC: Program Counter a következő utasítás címét tárolja (számláló funkcióval növekszik)

SP: Stack Pointer a verem adatszerkezet mutatója, a verem kezelő utasítások automatikusan változtatják

F: Flag az aritmetikai, logikai utasítások eredményének jellemzőit tárolja

8. CPU működési algoritmusa.

- FETCH (PC címmel RD művelet a kódmemóriára, eredmény az IR-be kerül)
- PC nő az utasítás méretével
- Dekódolás
- Adat memória olvasás (RISC ld utasítás, vagy CISC memória operandus esetén)
- Művelet (ALU)
- Memória írás (RISC st utasítás, vagy CISC memória eredmény esetén)
- WriteBack (eredmény írása a regiszterekbe – utasítás függő)
- Ismétlés a elejétől

9. DMA működése.

- Előkészítés: kezdőcímek, adatmennyiség beállítása, művelet engedélyezése
- DRQ: az IO eszköz DMA kérést jelez
- az aktuális utasítás befejeződik, CPU felfüggeszti a működését
- CPU nagy impedanciás állapotba kapcsolja a cím, adat és vezérlő busz kivezetéseit
- DACK: a CPU nyugtázza a kérést
- DMA vezérlő átveszi a busz működtetését
- Memória és IO RD/WR műveletekkel elvégzi a beállított méretű adatblokk másolását
- DMA vezérlő nagy impedanciás állapotba kapcsolja a cím, adat és vezérlő busz kivezetéseit
- DRQ kérés megszűnik

- CPU átveszi a busz működtetését
- CPU folytatja az utasítások végrehajtását

10. Címzési módok. Milyen C nyelvű utasítás használhatja ezeket?

Regiszter: az adat a regiszterben van, pl  $a = b$ ;

Immediate: az adat konstans, az utasítás kódjában van, pl  $a = 1$ ;

Közvetlen: az adat a memóriában van, a cím konstansként az utasításban található, pl  $a = b + c$ ;

Közvetett: az adat a memóriában van, a cím egy regiszterben található, pl  $a = *p$ ;

Indexelt: az adat a memóriában van, a cím egy regiszter tartalmának és az utasításban lévő konstansnak az összege, pl  $a = t[i]$ ;

## GEVAU505B, GEVAU505BL Digitális Rendszerek 3 vizsga

Név: .....

Neptun kód: .....

1. A Harvard és a Neumann architektúra összehasonlítása. ....
2. Memória olvasás idődiagramja. ....
3. A CPU működési algoritmus (utasítás végrehajtás lépései). ....
4. Bemenet változás érzékelésének szoftveres algoritmus (polling). ....
5. Egy RISC processzor órajele 100 MHz, egy utasítás átlagosan 2 ciklus. Egy 100x100 pixeles kép minden pontját egy 500 utasítású algoritmus dolgozza fel. Mennyi ideig tart a teljes kép feldolgozása? .....
6. Időzítő, PWM generátor mód felépítése, működése, felhasználása. Mi a különbség a „center aligned” és az „edge aligned” módok között? .....
7. Alacsonyabb prioritású ISR (magasabb prioritású kéréssel való megszakításának lefolyása). ....
8. Egy Harvard processzor az adatmemóriát 20 bites címbusszal éri el. Egy program két 16 bites ADC-vel sztereó audió jelet olvas be, 50 kHz mintavételi frekvenciával. Milyen hosszú hangot tud tárolni? .....
9. Egyszerű kimeneti és bemeneti portok megvalósítása FF-al, illetve TS kapuval. ....
10. Egy 8 bites CPU **ADD** utasításával összeadjuk a **01110110** és a **01100111** számokat. Mi lesz az eredmény és mennyi lesz az értéke előjeles, ill. előjel nélküli kódolásban, mi lesz a jelzőbitek (C, O, Z, S) értéke? .....

Összesen: .....

Értékelés: 1 0-4.5p; 2 4.6-5.9p; 3 6.0-7.4p; 4 7.5-8.9p; 5 9.0-10p



## Vizsga megoldási minta

### 1. A Harvard és a Neumann architektúra összehasonlítása.

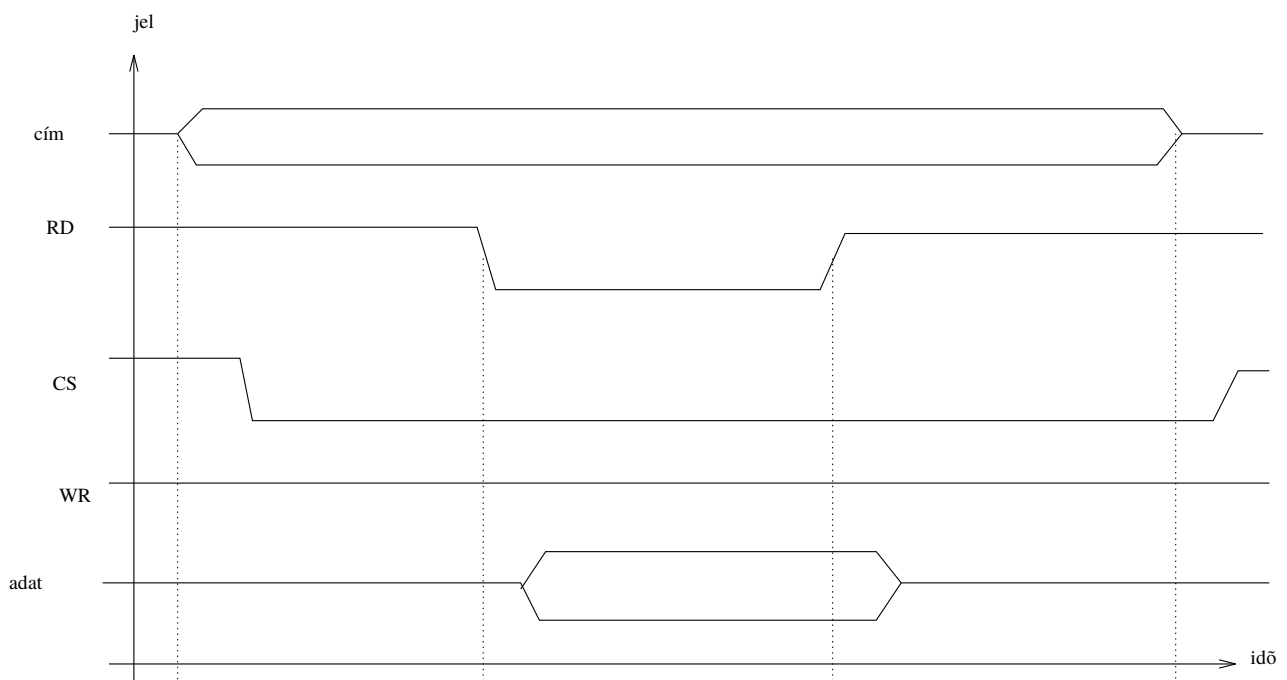
Harvard:

- 2 külön busz illesztő, egy a kód memória, egy pedig az adatmemória eléréséhez
- külön címtér
- két buszon egyidejű művelet lehet, ezért gyorsabb lehet
- kód önmódosítása nem lehetséges
- biztonságosabb
- beágyazott rendszerekben gyakoribb
- OS készítése nehéz

Neumann:

- 1 közös busz illesztő a kód és adat memóriához
- közös címtér
- egy buszon egy időben egy művelet, lassabb lehet
- kód önmódosító lehet (tiltásához memória manager kell)
- kevésbé biztonságos
- asztali gépekben gyakoribb
- OS készítése könnyű

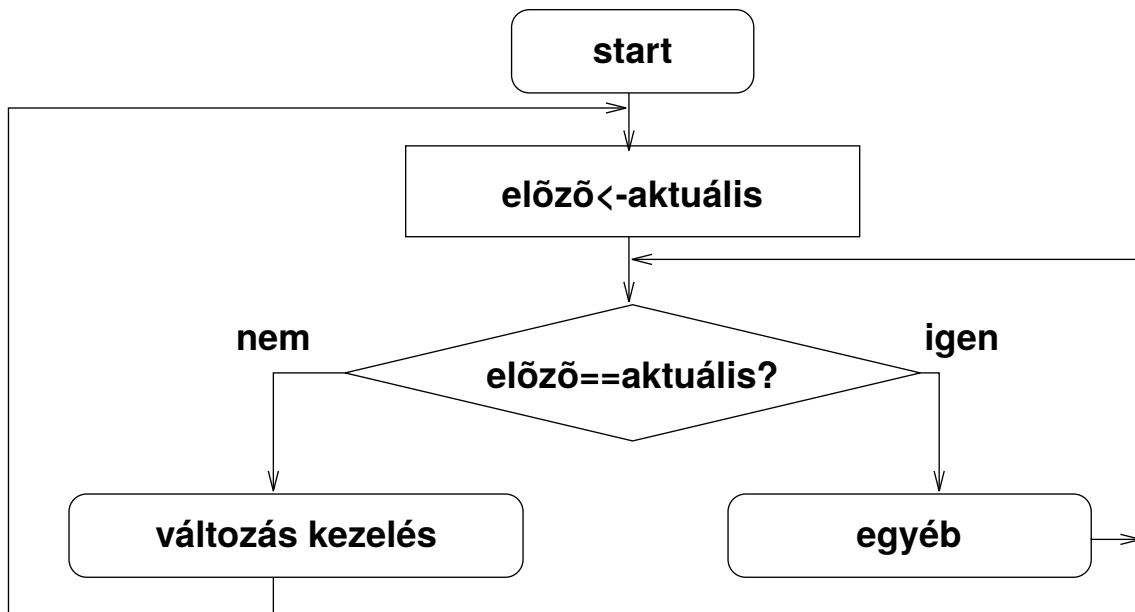
### 2. Memória olvasás idődiagramja.



### 3. A CPU működési algoritmus (utasítás végrehajtás lépései).

- FETCH (PC címmel RD művelet a kódmemóriára, eredmény az IR-be kerül)
- PC nő az utasítás méretével
- Dekódolás
- Adat memória olvasás (RISC ld utasítás, vagy CISC memória operandus esetén)
- Művelet (ALU)
- Memória írás (RISC st utasítás, vagy CISC memória eredmény esetén)
- WriteBack (eredmény írása a regiszterekbe – utasítás függő)
- Ismétlés a elejétől

4. Bemenet változás érzékelésének szoftveres algoritmus (polling).



5. Egy RISC processzor órajele 100 MHz, egy utasítás átlagosan 2 ciklus.  
Egy 100x100 pixeles kép minden pontját egy 500 utasítású algoritmus dolgozza fel. Mennyi ideig tart a teljes kép feldolgozása?

Egy ciklus ideje:  $t = 1/100 \cdot 10^6 = 1/10^8 = 10^{-8}$  sec

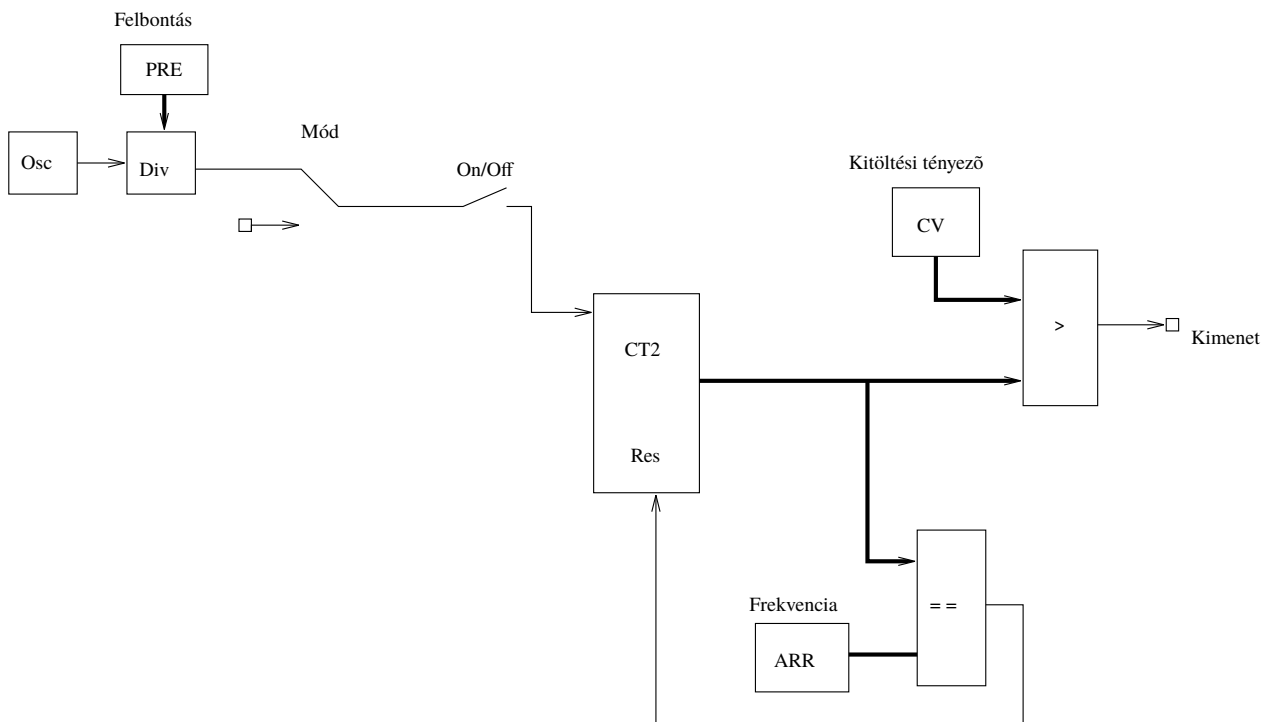
Egy utasítás ideje:  $t_{ut} = 2 \cdot t = 2 \cdot 10^{-8}$  sec

Pixelek száma:  $P = 100 \cdot 100 = 10^4$  db

Összes szükséges utasítás:  $U = P \cdot 500 = 5 \cdot 10^6$  db

Teljes idő:  $T = U \cdot t_{ut} = 5 \cdot 10^6 \cdot 2 \cdot 10^{-8} = 10 \cdot 10^{-2} = 10^{-1} = 0.1$  sec

6. Időzítő, PWM generátor mód felépítése, működése, felhasználása.  
Mi a különbség a „center aligned” és az „edge aligned” módok között?

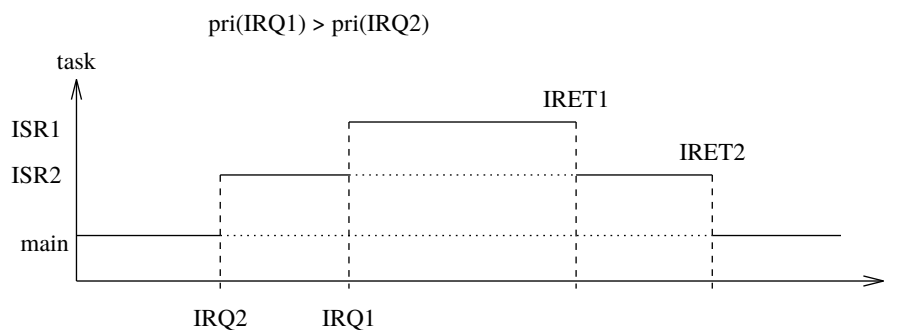


Felhasználása: motor fordulatszám szabályozás, analóg jel előállítás (kimeneten szűrővel), LED fényerő változtatás, stb.

Több kimeneti csatorna esetén:

- edge align: egyirányú számláló, a kimeneti jelek éle egy időben
- center align: kétirányú számláló (fr feleződik), a kimenetei jelek közepe egy időben

7. Alacsonyabb prioritású ISR (magasabb prioritású kéréssel való megszakításának lefolyása.



8. Egy Harvard processzor az adatmemóriát 20 bites címbusszal éri el. Egy program két 16 bites ADC-vel sztereó audió jelet olvas be, 50 kHz mintavételi frekvenciával. Milyen hosszú hangot tud tárolni?

Adatmemória kapacitás:  $K = 2^{20}$  byte

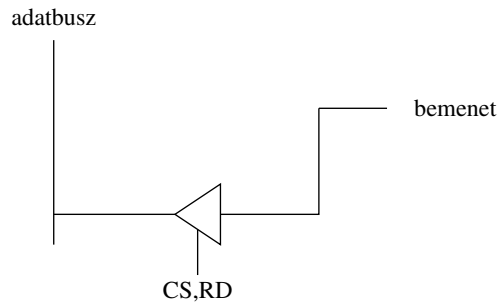
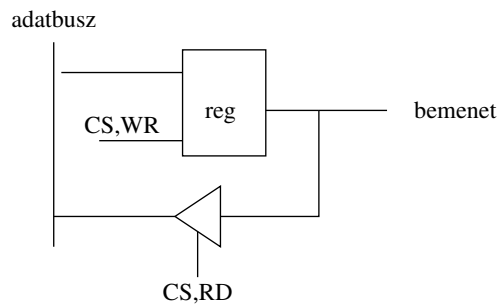
Mintaméret byte-ban:  $M = 2 \cdot (16/8) = 4$  byte

Minták száma másodpercenként:  $\text{MPS} = 50 \cdot 10^3$  db/sec

Adatmennyiség másodpercenként:  $\text{DPS} = \text{MPS} \cdot M = 2 \cdot 50 \cdot 10^3 = 10 \cdot 10^4 = 10^5$  byte/sec

Tárolható idő:  $t = K/\text{DPS} = 2^{20}/10^5 \sim 10^6/10^5 = 10$  sec

9. Egyszerű kimeneti és bemeneti portok megvalósítása FF-al, illetve TS kapuval.



10. Egy 8 bites CPU **ADD** utasításával összeadjuk a **01110110** és a **01100111** számokat. Mi lesz az eredmény és mennyi lesz az értéke előjeles, ill. előjel nélküli kódolásban, mi lesz a jelzőbitek (C, O, Z, S) értéke?

Átvitel bitek: 011001100  
 Adat 1: 01100111  
 Adat 2: 01110110  
 Eredmény: 11011101

C=0, Z=0, S=1, O=1

Eredmény értéke előjel nélküli kódolásban: 221

Eredmény értéke előjeles kódolásban: -35

(Abs= ~11011101+1= 00100010+1= 00100011=> 35)