



BUDAPESTI MŰSZAKI ÉS GAZDASÁGTUDOMÁNYI EGYETEM
VILLAMOSMÉRNÖKI ÉS INFORMATIKAI KAR
MÉRÉSTECHNIKA ÉS INFORMÁCIÓS RENDSZEREK TANSZÉK

Digitális technika (VIMIAA03)

12. gyakorlat és laboratórium

Raikovich Tamás
BME MIT

Stopperóra

- A mai feladat egy 0,1 s felbontású stopperóra megvalósítása a MiniRISC processzoros rendszerre assembly nyelven
- Megjelenítés a hétszegmenses kijelzőn
 - DIG0: tizedmásodpercek (0 – 9)
 - DIG1: másodpercek (0 – 9)
 - DIG2: tíz másodpercek (0 – 5)
 - DIG3: percek (0 – 9)
- Ciklikus állapotváltás a BT0 nyomógomb lenyomására
 - IDLE (0x02): a stopper áll és nullát mutat (kezdőállapot)
 - RUN (0x01): a stopper fut
 - STOP (0x00): a stopper áll és mutatja a mért időt
- Az időzítő és a nyomógomb perifériákat megszakításosan fogjuk kezelni

Megszakításkezelés

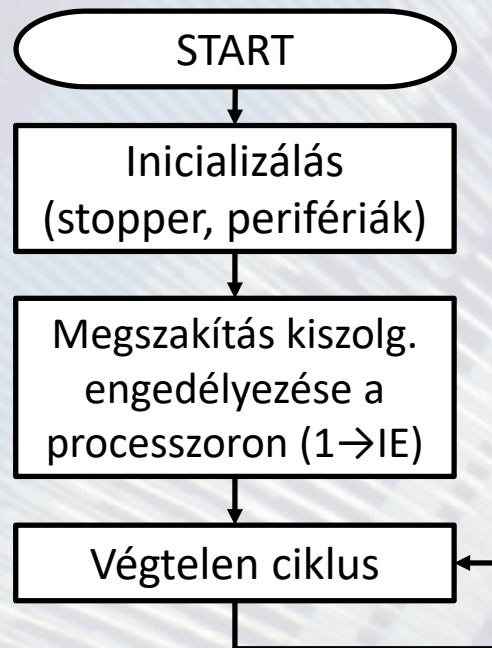
- A megszakításkérés hardveres jelzés a perifériától a processzor felé kiszolgálási igényre
- A MiniRISC processzor ezt az aktuális utasítás végrehajtása után elfogadja, ha az IE flag 1
 - PC, flag-ek → verem
 - 0x01 (megszakítás vektor) → PC
 - Ide ugró utasítás kerül a megszakításkezelő rutinra
 - Visszatérés az RTI utasítással (verem → PC, flag-ek)
- A MiniRISC processzornak egy megszakítás bemenete van, ezért a forrást azonosítani kell a megszakításkezelő rutinban
- A kiszolgálás során a jelzés(eke)t törölni, nyugtázni kell a perifériá(k)ban

Megszakításkezelés

- **A megszakítások nem tudni, hogy mikor következnek be**
 - A kiszolgálás előtt általában el kell menteni a processzor állapotát, utána pedig vissza kell állítani
 - A MiniRISC processzor esetén ez nehézkes, ezért a megszakításkezelő rutinhoz csak ott használt regisztereket rendelünk
- **A regiszterhasználat legyen az alábbiak szerint**
 - ***r0 – r7***: megszakításkezelésen kívül
 - ***r8 – r15***: megszakításkezeléshez
 - r8 – r9: paraméter átadás a szubrutinok számára
 - r10: ideiglenes adatok tárolása
 - r11: a stopper állapota (IDLE, RUN vagy STOP)
 - r12 – r15: a stopper számlálói

Főprogram

- A stopperóra megvalósítása a megszakításkezelő rutinban történik, ezért a főprogramot végtelen ciklussal zárjuk
- Az egyes inicializálásokhoz önálló szubrutinok tartoznak



Inicializálás

- **Stopper (stopper_init)**
 - Számlálók törlése, kezdőállapot (IDLE) beállítása
- **Időzítő periféria (tmr_init)**
 - Már használtuk, részletek a 11. labor anyagában
 - A 100 ms időzítési periódushoz milyen számláló kezdőállapot és előosztás szükséges?
 - Milyen értéket írjunk a parancsregiszterbe?
- **Nyomógomb periféria (btn_init)**
 - A regisztereket lásd a következő dián
 - A BT0 nyomógomb megszakítást kell engedélyezni

A nyomógomb periféria

4 bites bemeneti periféria pergésmentesítéssel és megszakításkérésrel

- **Adatregiszter**

- BASEADDR + 0x00, 8 bites, csak olvasható
- A BT_i bit az i -edik nyomógomb állapotát adja meg

7. bit	6. bit	5. bit	4. bit	3. bit	2. bit	1. bit	0. bit
0	0	0	0	BT3	BT2	BT1	BT0
R	R	R	R	R	R	R	R

- **Megszakítás engedélyező (Interrupt Enable, IE) regiszter**

- BASEADDR + 0x01, 8 bites, írható és olvasható
- Az IE_i bit engedélyezi az i -edik nyomógombra a megszakításkérését

7. bit	6. bit	5. bit	4. bit	3. bit	2. bit	1. bit	0. bit
0	0	0	0	IE3	IE2	IE1	IE0
R	R	R	R	R/W	R/W	R/W	R/W

- **Megszakítás flag (Interrupt Flag, IF) regiszter**

- BASEADDR + 0x02, 8 bites, írható és olvasható
- Az IF_i bit jelzi a BT_i állapotának megváltozását, 1 beírásával törölhető

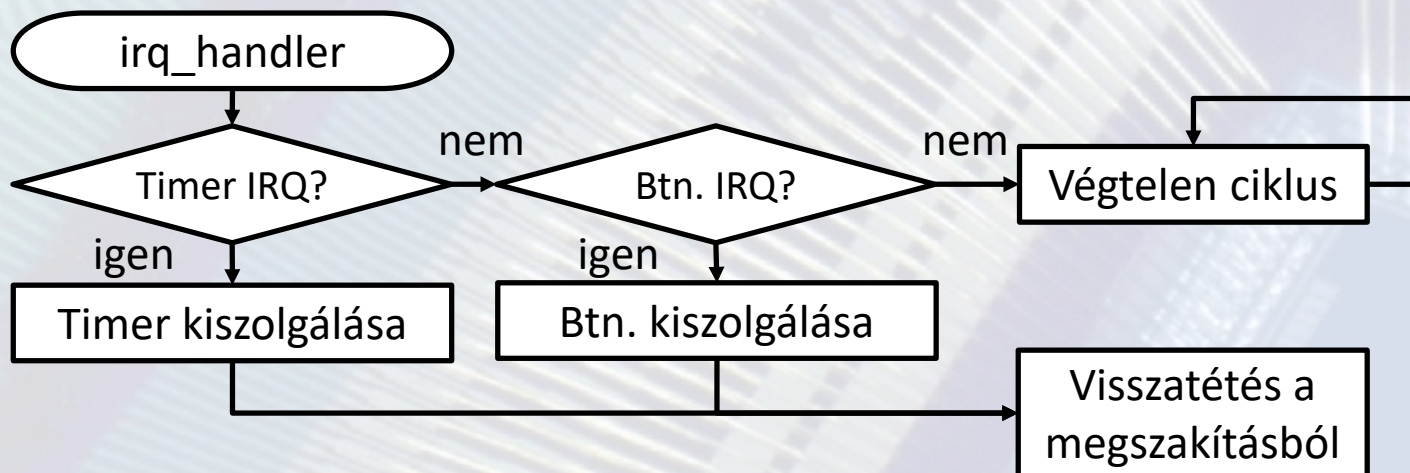
7. bit	6. bit	5. bit	4. bit	3. bit	2. bit	1. bit	0. bit
0	0	0	0	IF3	IF2	IF1	IF0
R	R	R	R	R/W1C	R/W1C	R/W1C	R/W1C

Megjelenítés a kijelzőn

- **A 10. laboron már csináltunk ilyen**
- **1 digitos megjelenítés: disp_wr szubrutin**
 - r8: az alsó 4 bit a megjelenítendő értéket tartalmazza, az MSb pedig a tizedespont állapota
 - r9: a megjelenítéshez használandó digit adatregiszter címe
- **Hétszegmenses dekóder megvalósítása tömbbel**
 - Elem címe = kezdőcím + méret·index (a méret most 1 byte)
 - A szegmensképen be kell állítani a tizedespont állapotát
- **Milyen címzést kell használni a tömb elemek olvasásához és a digit adatregiszter eléréséhez?**

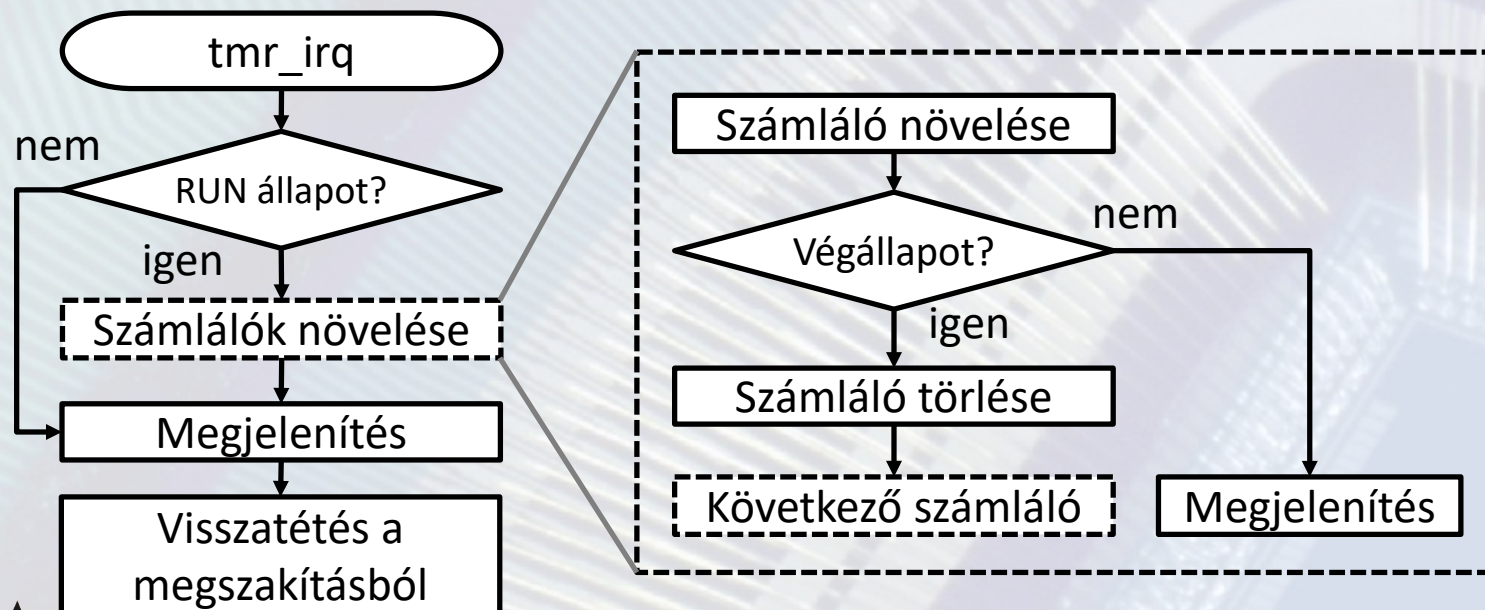
Megszakításkezelő rutin

- **A forrás azonosítása és kiszolgálása**
 - Timer: TS regiszter olvasása, ez törli TIT/TOUT-ot
 - Nyomógomb: BTIF olvasása és IFO vizsgálata
 - A BTIF bitek 1 beírásával törölhetők
- **Nem várt megszakítás esetén végtelen ciklus (hiba)**



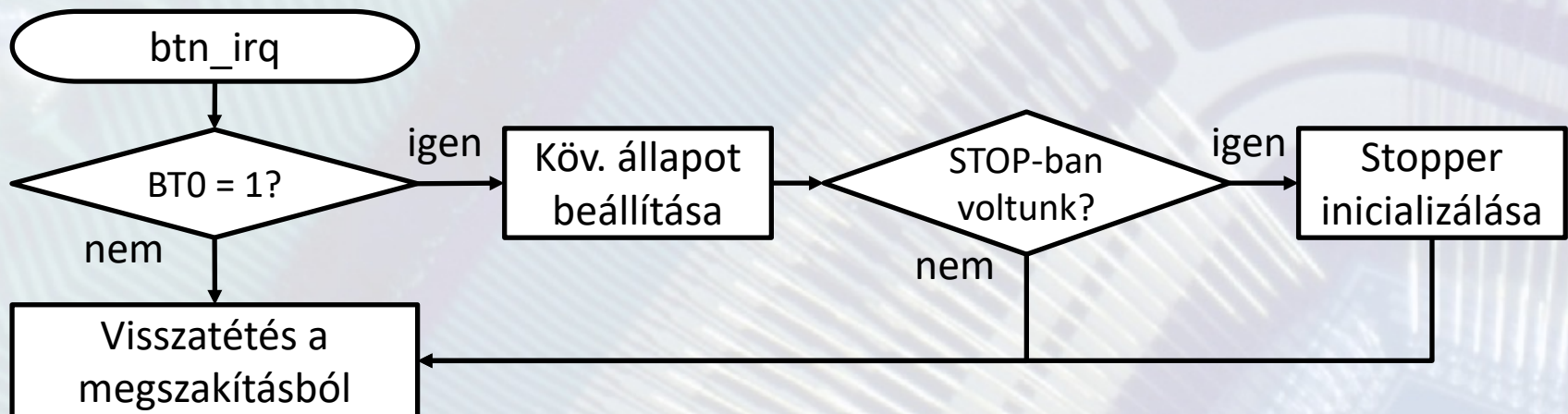
Időzítő megszakítás kiszolgálása

- Itt végezzük a stopper számlálóinak növelését és a kijelzőn történő megjelenítést
- A tizedespont legyen bekapcsolva DIG3-nál és DIG1-nél
- A számlálókat csak a RUN állapotban kell növelni
- A következő számlálót csak akkor kell növelni, ha az előző a végállapotban volt



Nyomógomb megszakítás kiszolgálása

- Itt a stopper következő állapotát kell beállítani
- A használt állapotkódolás esetén hogy tehető ez meg?
- Hogy vizsgálható, hogy az előző állapot a STOP volt?



Szorgalmi feladat

- **Módosítsuk a programot úgy, hogy a stopperóra századmásodperces (10 ms) felbontású legyen**
- **Megjelenítés a hétszegmenses kijelzőn**
 - DIG0: századmásodpercek (0 – 9)
 - DIG1: tizedmásodpercek (0 – 9)
 - DIG2: másodpercek (0 – 9)
 - DIG3: tíz másodpercek (0 – 5)
 - DIG2-nél legyen bekapcsolva a tizedespont