



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

Digitális technika

VIMIAA01

Fehér Béla
BME MIT

A megszakításrendszer

- A mikroprocesszoros rendszerek egyik legfontosabb alrendszere
- Feladata a processzoron futó program működésének támogatása, több szempontból is
 - Használatával a program felépítés, a program szerkezet jelentősen egyszerűsödhet
 - Használatával a program normál vagy rendkívüli eseményekre történő reakcióideje jelentősen javulhat
- **Veszélyek:**
 - Használatával a program valódi komplexitása jelentősen megnő (ha ez nem is nyilvánvaló)
 - A program virtuálisan párhuzamos „konkurrens” módon végrehajtódó részfunkciókat tartalmaz

A megszakításrendszer

- **Összetett számítógép rendszerekben a megszakítások alatt gyakran két kategóriát is együtt értenek:**
 - Megszakítások (Interrupt, IT)
 - Kivételek (Exceptions, EX)
 - Néha nincs megkülönböztetve, vagy keverednek a fogalmak
- **Megszakítások:**
 - Elsősorban a hardverhez kapcsolódó események jelzése
- **Kivételek:**
 - Elsősorban a programvégrehajtás során keletkező, ritka, de kezelendő események kezelése
- **Együttes említésüket az eltérő keletkezésük ellenére a nagyon hasonló kezelési módjuk, kiszolgálásuk indokolja**

A megszakításrendszer

- **Megszakítások:**
 - Elsősorban a hardverhez kapcsolódó események jelzése
 - Normál működés közbeni kiszolgálási kérések
 - I/O periféria események
 - Időzítő által generált ütemező jelek
 - A megszakítás használatának előnye az egyszerűbb programszervezés
 - Hibákra utaló jelzések
 - Memória paritáshiba
 - Tápfeszültség probléma, kimaradás
 - A megszakítás használatának előnye, az egyébként nagyon ritka eseményekre történő megbízható gyors reakció

A megszakításrendszer

- **Kivételek:**
 - Elsősorban a programvégrehajtás során keletkező, ritka, de kezelendő események
 - Alkalmazói programok, aritmetikai kivételek
 - Osztás 0-val, művelet lebegőpontos NaN operandussal
 - Túlcsordulás, alucsordulás, átfordulás
 - Állandó ellenőrzésük túl sok időt venne igénybe, nem hatékony
 - Rendszerprogram jelzései
 - Rendszer/felhasználó programvédelmi hiba
 - Verem alul/túlcsordulás
 - Érvénytelen utasításkód
 - Ezeket az eseteket az operációs rendszer kivétel kezelési mechanizmusa kezeli

A megszakításrendszer

- **Megszakítások és kivételek:**
 - Láttuk, a kiváltó események teljesen más jellegűek
 - Ugyanakkor ritkák és speciális kezelést igényelnek
 - Ezek a kódrészletek nem kell, hogy a fő program részei legyenek
 - A speciális programrészletek meghívása, aktiválása történhet teljesen hasonló módon, mind a (hardver) megszakítás, mind a (szoftver) kivételkezelés során
 - Egy azonos jelzés esetén, ettől a ponttól kezdve egységes kezelhetőségük kifejezetten előnyös
 - A továbbiakban a **megszakításokkal** foglalkozunk, a MiniRISC környezetben nincs kivételkezelés

A megszakításrendszer

- **Megszakítások tehát:**
- **Alapvetően események jelzésére szolgálnak:**
 - **Aszinkron események:** A külső események a rendszer működésétől függetlenül történnek (A perifériák a külső fizikai jelzéseket a rendszer órajelhez szinkronizálják)
 - **Szinkron események:** Valamilyen értelemben a program által kiváltott, annak működésével összefüggő események (pl. időzítő indítása adott értékkel, összetett periféria működésének indítása, ami adott idő múlva kiszolgálást kér)

A megszakításrendszer

- **Megszakításrendszerek feladatai:**
 - Megszakítási események jelzése
 - Megszakításkérés prioritásának kezelése
 - Megszakításkérések engedélyezése / tiltása
 - Megszakításforrás azonosítása
 - Megszakítás kérés kiszolgáló szubrutin kezdőcímének meghatározása
 - Megszakításkérés kiszolgáló szubrutin (ISR, Interrupt Service Routine) végrehajtása, a szükséges adminisztráció elvégzése
 - Visszatérés a program eredeti állapotának visszaállításával

A megszakításrendszer

- **Megszakítási események jelzése:**
 - A perifériaeszközök részéről egyedi vagy közösített IRQ vonalakon
 - Hagyományos rendszerekben voltak láncra fűzött „daisy chain” kialakítások (elosztott IT rendszer és vezérlő)
→ manapság ez már ritkán használt
 - PCI buszon 4 részben közösen használt IRQ vonal
 - Egyedi IRQ vonalak minden eszköz számára
 - Szintvezérelt vagy élvezérelt megoldással
 - Az esemény hatására $IRQ = 1$ lesz
 - Az esemény hatására IRQ egy $0 \rightarrow 1$ átmenetet generál
 - Mindkettő használható több forrás közösített jelzésére is

A megszakításrendszer

- **Megszakítások prioritása:**
 - A kérések kiszolgálásának sorrendjét a prioritási beállítások határozzák meg
 - Egyidejűség esetén mindig az aktuálisan legnagyobb prioritású kérés jut érvényre
 - Egyszintű IT rendszer esetén egy IT kérés elfogadása automatikusan tiltja további kérések érvényre jutását (átmeneti késleltetés)
 - Ha a megszakításrendszer többszintű, és az aktív IT kérés kiszolgálása engedélyezi újabb kérések elfogadását, akkor a beérkező magasabb prioritású kérések megszakíthatják az alacsonyabb szintű IT kiszolgáló rutinját is

A megszakításrendszer

- **Megszakítások prioritása:**
 - Prioritások kezelése:
 - Közvetlenül a CPU-ban, ha a CPU-nak van több megszakítás kérési (IRQ_i) bemenete
 - Egy speciális külső egységben, az ún. megszakítás vezérlőben (Interrupt Controller, ITC).
 - Az ITC, mint slave periféria fogadja az egyedi IRQ_i vonalakat, működési paraméterei programozhatóak, és a beállításoknak megfelelően továbbítja az aktuális kérést/kéréseket a CPU felé
 - Túl sok IT forrás esetén prioritás csoportok alakíthatók ki, a csoportok között definiált prioritási szintekkel

A megszakításrendszer

- **Megszakítások prioritása:**
 - Tipikus prioritáskezelési módszerek
 - Fix prioritás: A források egymáshoz viszonyított prioritása rögzített, a tervezés, vagy a rendszer konfiguráció során dől el
 - Körbenforgó: Azonos prioritású egységek között a legutoljára kiszolgált egység a sor végére kerül
 - Lehetséges kevert prioritási rendszer is:
 - Fix a csoportok között és körbenforgó a csoporton belül

A megszakításrendszer

- **Megszakítások engedélyezése/tiltása:**
 - **Ezt maszkolásnak hívjuk**
 - Lehet globális és lokális is
 - Globális engedélyezés / tiltás :
 - Rendszerszintű, a CPU-ban STI/CLI, EI/DI utasításokkal
 - Ha van a rendszerben, akkor a programozható megszakítás vezérlőben
 - Lokális, eszköz vagy funkció szintű enged. / tiltás
 - Az egyes perifériák parancsregiszterében
 - A teljes perifériára vagy akár egyedi eseményekre is

A megszakításrendszer

- **Megszakítások engedélyezése/tiltása:**
 - Egyes processzoroknál létezik ún. nem maszkolható megszakítás is
 - NMI Non-Maskable Interrupt
 - Ennek feladata, hogy **MINDIG** érvényre jusson!
 - Használata egyedi, abszolút sürgős esetekre:
 - Pl. tápfeszültség kimaradás
 - Az NMI csak élvezérelt jelzésű lehet
 - A kérés egy adott pillanatban lép fel és kiszolgálásra kerül
 - A statikus NMI kérés kezelés mindig megszakítaná önmagát is!

A megszakításrendszer

- **A megszakításforrások azonosítása:**
 - A források azonosítása fontos a megfelelő kiszolgáló rutin (ISR) kezdőcímének meghatározásához
 - Amennyiben minden forrásnak önálló jelvezetéke van, akkor az azonosítás automatikusan teljesül
 - A közös vezetéken beérkező kérések kiszolgálásának megkezdéséhez a forrás azonosítása szükséges
 - Programozott lekérdezéssel a státuszregiszterek jelzőbitjei alapján, az első aktív állapotot kiválasztva
 - A lekérdezés sorrendje rögzíti a prioritást ezen a csoporton belül, de szoftverből ez könnyen konfigurálható

A megszakításrendszer

- **A megszakításforrások azonosítása:**
- **Az azonosítás után az ISR kezdőcíme meghatározható**
 - Programozott módon
 - A periféria által támogatott módon, ezt vektoros megszakítás rendszernek nevezzük
 - A megszakítás kiszolgáló rutinok kezdőcímei egy táblázatban vannak (Interrupt Vector Table), ennek báziscíme fix, vagy egy regiszterben konfigurálható
 - A periféria egy indexet ad ehhez a táblázathoz, ami alapján indirekt címzéssel a második lépésben az ISR belépési címe könnyen meghatározható

A megszakításrendszer

- **A megszakításrendszerek változatai:**
 - **Egyszintű megszakítás rendszer:**
 - A CPU vagy a normál programot hajtja végre vagy valamelyik megszakítás kiszolgáló szubrutint
 - Ha már átváltott a megszakításra, újabb megszakítást nem tud fogadni, mert az IRQ elfogadásakor a továbbiakat automatikusan letiltja (és visszatéréskor újból engedélyezi)
 - **Többszintű megszakítás rendszer**
 - A CPU a normál program végrehajtási szinten túl a megszakítások alatt is fogadhat további megszakításkérést
 - Az alacsony prioritású megszakítás rutinok futása is megszakítható a magasabb prioritású kérések által
 - A prioritás tehát nem csak az elfogadásakor, hanem a kiszolgálás alatt is érvényre juthat

A megszakításrendszer

- **A megszakításrendszer változatai:**
 - **A megfelelő megszakítási rendszer kiválasztása a rendszer követelmények alapján lehetséges**
 - **A cél mindig a CPU terhelés optimalizálása, a kérések kiszolgálásának gyorsítása**
 - **A legkedvezőbb kialakítás gyakran extra hardver beépítésével lehetséges, a rendszer minden pontján**
 - **Speciális kiegészítések a periféria egységekben**
 - **Speciális kiegészítések a rendszer kommunikációs kapcsolataiban**
 - **Speciális kiegészítések a processzorban**

A megszakításrendszer

- **A periféria egységek speciális hardver egységei a megszakításos kiszolgálás támogatásához**
- **A megszakítás alrendszer alapvető elemei:**
 - **A megszakítást engedélyező parancsbit (bitek)**
 - Egyedi, bitenkénti vagy a periféria egészére szóló
 - **A megszakítási esemény azonosítása és ennek alapján egy statikus, de törölhető jelző flag előállítása**
 - **Az IRQ vonalat meghajtó kimeneti logika**
 - Ez esetleg speciális nyitott kollektoros kimenetet kíván
- **A megszakítás kérés törlése, nyugtázása**
 - Automatikusan, a lekérdezéssel
 - Külön törlő paranccsal

A megszakításrendszer

- **A processzoros rendszer speciális hardver egységei a megszakításos kiszolgálás támogatásához**
- **A rendszerbusz megszakítás jelei**
 - **Megszakítás kérés jel/jelek (NMI + IRQ_i):**
 - Önálló jelek perifériánként
 - Egyetlen jel az IRQ_i jelek logikai VAGY kapcsolatával
 - Egyetlen jel az IRQ_i jelek speciális „közösíthető” áramköri megoldásával („huzalozott” VAGY)
 - **Megszakítás elfogadás jelzés (IACK):** Egyes processzorok ezzel jelzik a perifériák számára az IRQ elfogadás fázisát, amikor pl. az ISR vektor címe (indexe) a CPU számára elküldhető

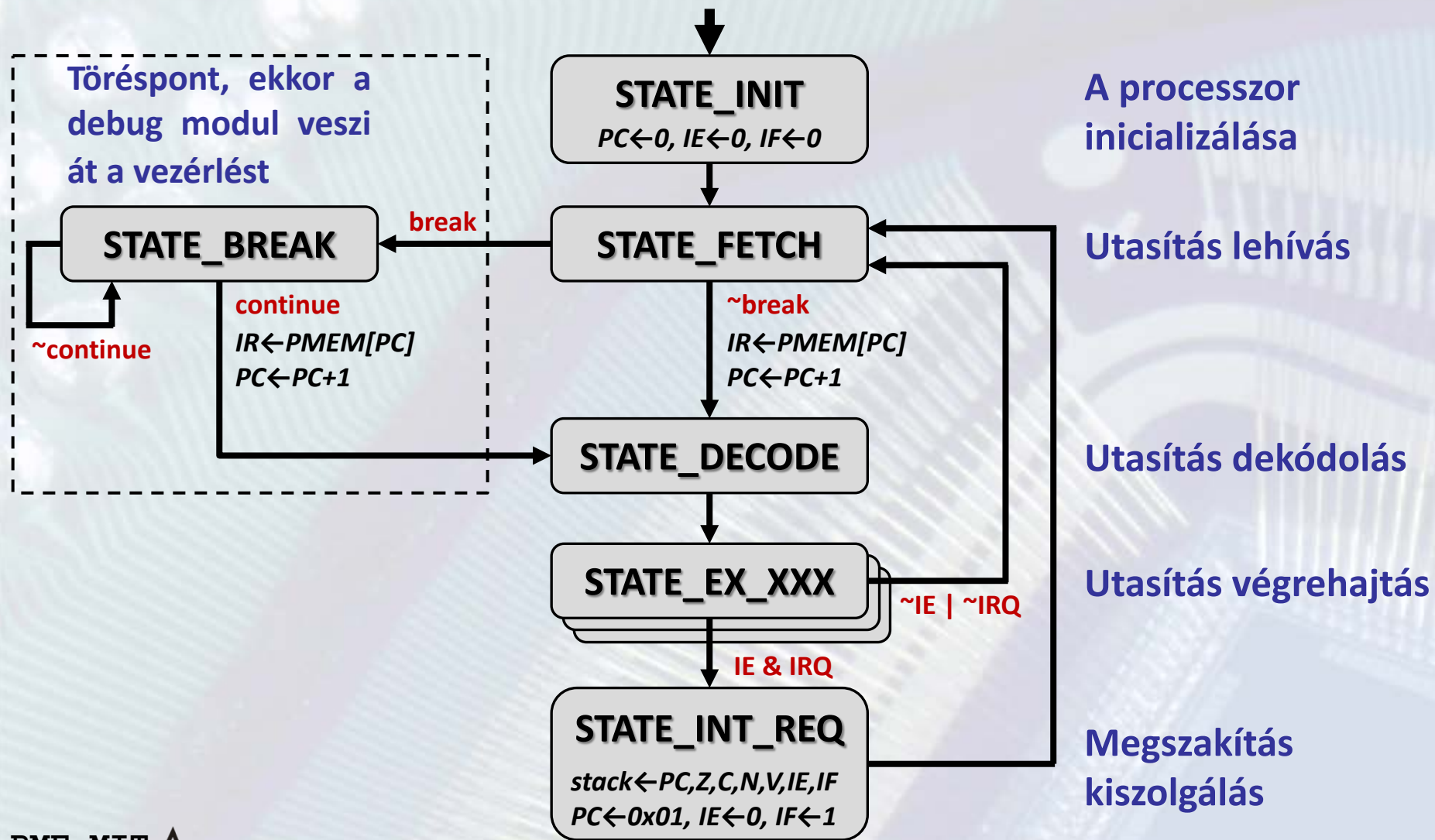
A megszakításrendszer

- **A processzoros rendszer speciális hardver egységei a megszakításos kiszolgálás támogatásához**
- **Az ITC megszakítás vezérlő kialakítási lehetőségei**
- **Központi megszakítás vezérlő**
 - Egyetlen egység, sok IRQi bemeneti vonallal
 - Eng./tiltás, prioritások, ISR vektorok a CPU felé
 - A perifériák egyszerűbbek
- **Elosztott megszakítás vezérlő („daisy chain”)**
 - Minden egység egy közös IRQ vonalat vezérel
 - Az IACKIN-IACKOUT vonal láncba kötve fut végig, az aktív kérő blokkolja a továbbadást
 - Bonyolult egységek, de külső busznál hatékony

A CPU megszakításrendszere

- **A processzor felkészítése a megszakítások fogadására és kiszolgálására**
 - A processzorok általában kevés (1,2) megszakítás kérő bemenettel rendelkeznek (IRQ, NMI)
 - A processzor felé a megszakítás kérések tetszőleges időpillanatban kiadhatók
 - A megszakítás hatására a processzor az utasítás végrehajtási sorrendet az előzetesen lefordított kódhoz képest módosítja
 - A processzor a megszakítást csak utasítás végrehajtás végén engedi érvényre jutni, amennyiben az IT engedélyezett ($IE = 1$)

MiniRISC processzor – Vezérlő állapotgép



A processzor inicializálása

Utasítás lehívás

Utasítás dekódolás

Utasítás végrehajtás

Megszakítás kiszolgálás

A CPU megszakításrendszere

- A megszakítás kiszolgálásának folyamata a MiniRISC processzor szempontjából a következő:
 - A CPU észleli az IRQ vonal aktiválását
 - A CPU az aktuális utasítást végrehajtja
 - Ha az ($IE = 1$ és $IRQ = 1$), akkor az EXECUTE fázis után belép az IRQ_REQ állapotba. (Egyébként soha.)
 - Az IRQ_REQ állapotban lényegében végrehajt egy szubrutinhívást a **Megszakításvektor** címére (0x01). Egyúttal a legfontosabb adatokat, állapotjelző biteket a hardver verembe menti (PC, ZCNV, IE, IF)
 - $PC = 0x01$, $IE = 0$ és $IF = 1$ állapottal végrehajtja az ISR megszakítás kiszolgáló rutinra ugró utasítást.

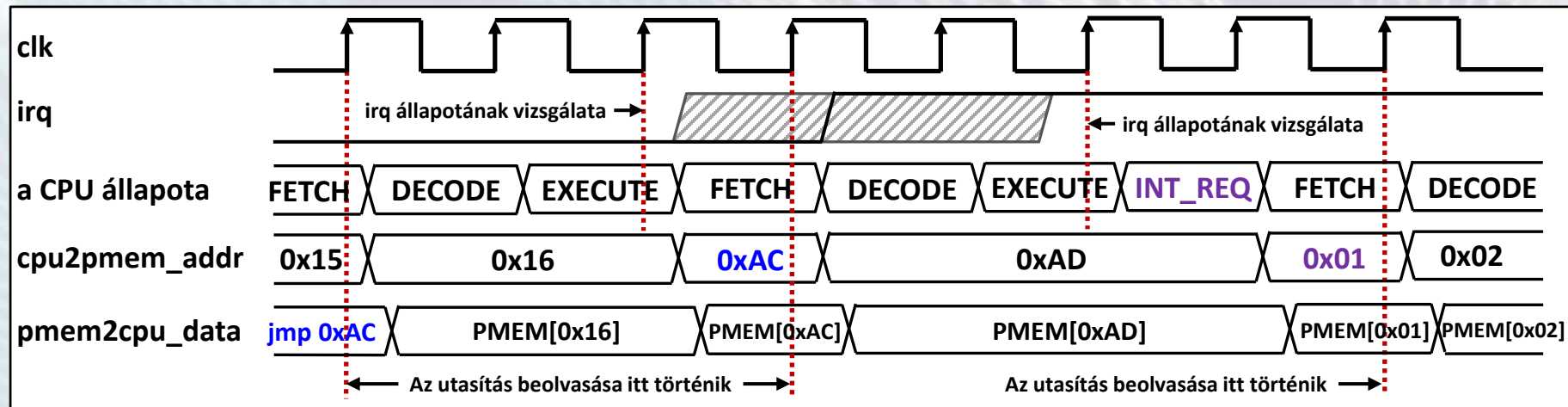
MiniRISC processzor – IT elfogadás

Programmémória interfész

- Ugrás vagy szubrutinhívás esetén a programszámláló értéke módosul(hat) a végrehajtási (execute) fázisban
- Megszakításkérés kiszolgálása esetén az INT_REQ állapotban a programszámlálóba betöltődik a megszakítás vektor (0x01)
- Az aktuális utasítás végrehajtása 4 órajel ciklus hosszú:

FETCH_DECODE_EXECUTE_INT_REQ

- A fenti két esetben a következő lehívási (fetch) fázisra éppen időben megjelenik az új cím a programmemória címbuszán



A CPU megszakításrendszere

- **Processzor állapot értelmezése:**
- **A megszakítás elfogadásakor a CPU aktuális állapotát úgy kell elmenteni, hogy az ISR rutin befejezése után, a normál program egyértelműen folytatható legyen.**
 - Ez általában hardveresen, automatikusan megtörténik
- **A megszakításrutin elindításakor gondoskodni kell a programállapot megőrzéséről is**
 - Az ISR-en belül használt regiszterek értékét el kell menteni és visszatérés előtt vissza kell állítani a korábbi értékre

A CPU megszakításrendszere

- **Processzor és program állapot értelmezése:**
- **A regiszter állapotmentés mértéke, módja, támogatása processzoronként eltérő → Késleltetési idő csökkentése**
 - Egyenként, memóriába, vagy az ott lévő verembe
 - Vektorosan egy regisztercím tartományt automatikusan
 - Független, új regiszterkészlet elérhetősége, automatikus, vagy egy utasításos váltással
- **Egy érdekes gondolat: A megszakítási állapot olyan, mintha egy új processzort működtetnénk**
 - IRQ hatására a „fő” processzor leáll, de bizonyos változói, memóriája elérhetők az újonnan indított „IT” processzor számára

A megszakításrendszer

- **Megszakítások engedélyezése:**
- **A megszakításkérések elfogadása, kiszolgálása programozottan engedélyezhető**
 - **MiniRISC STI, CLI utasítások (Set IE, Clear IE)**
 - Lehetnek olyan működési állapotok amikor ez szükséges (pl. kritikus programrészlet végrehajtása, ami nem megszakítható)
 - Ugyanez a helyzet az ISR-be belépéskor, ekkor automatikusan tiltódik a további IT elfogadás ($IE=0$)
 - A RESET állapotban az $IE = 0$, azaz ha használni szeretnénk megszakítást, akkor azt külön engedélyezni kell

MiniRISC processzor –Megszakítás

A MiniRISC megszakítási rendszere

- Aktív magas megszakításkérő bemenet (IRQ)
- Egyszerű megszakítási rendszer
 - A perifériától csak jelzés jön, a kérés azonosítása a programban történik
 - A megszakításkezelő rutin címe fix 0x01

Megszakítások kiszolgálása a MiniRISC esetén

- Alapvetően a szubrutinhívásra hasonlít
- Ha IE=1 és IRQ=1, akkor az aktuális utasítás végrehajtása után
 - A visszatérési cím és a flag-ek (ALU státusz bitek, IE, IF) elmentésre kerülnek a **verembe (stack)**
 - A megszakítás vektor (0x01) betöltésre kerül a programszámlálóba és az IE bit törlődik

Visszatérés a megszakításból: **RTI utasítás**

- A PC-be betöltésre kerül a visszatérési cím és visszaállításra kerülnek a flag-ek a veremből

```
00:  jmp start
01:  jmp usrt_rx

02:  start:
02:  mov r0, #0
03:  mov LD, r0
04:  mov r0, #0x3b
05:  mov UC, r0
06:  sti
07:  loop:
07:  jsr delay  IRQ
08:  mov r8, #str
09:  jsr print_str
0A:  jmp loop
```

```
stack←{PC (0x09),flag-ek}
PC←0x01, IE←0
```

```
30:  usrt_rx:
30:  mov r15, UD
31:  mov LD, r15
32:  rti
```

{PC,flag-ek}←stack

Az ISR megszakítás rutin

- **Az ISR megszakításrutin egy speciális programrész**
- **Lényegében egy szubrutin, de nem tartozik hozzá hívó utasítás**
 - Aktiválása a IRQ elfogadáson keresztül történik (Visszatérés pedig az RTI-vel lehetséges)
 - Nincsenek bemeneti paraméterei → Nem hívjuk, jön!
 - Nincsenek visszatérési értékek → Kinek, miért?
 - Természetesen a regiszterértékekhez, globális változókhoz, memóriához, perifériákhoz hozzáférhet, azokat használhatja, módosíthatja, ha kell
 - Sok esetben assembly szintű programrészlet a hatékony, gyors végrehajtás érdekében

Programszervezési alternatívák

- **A megszakításrendszer tervezése a programszervezésre is hatással lehet**
 - A programok feladatai megoszlanak a főprogram és a periféria kiszolgálást végző ISR között.
 - Ennek szintjei a következők:
 - Az ISR csak a legszükségesebb perifériaműveleteket hajtja végre. Minden más a főprogramban történik.
 - Az ISR a minimális aktivitáson túl általános feladatokat is elvégez, a perifériához kapcsolódóan.
 - Az ISR tartalmazza az összes szükséges feladatot, a főprogram szinte egy aktivitás nélküli várakozó hurok.
 - „A rendszer üresjáratú folyamata”

Digitális technika

12. EA vége