**5. Megszakítások**

A megszakítási rendszer a folyamatok közben keletkező események feldolgozására szolgál. Ezen események lehetnek:

* **Szinkron jellegűek:** melyek keletkezése egy program futása közben meghatározható helyen és időpontban várható (például túlcsordulás)
* **Aszinkron események:** melyek várhatóak, de időpontjuk előre nem ismert (például adatbeolvasás)
* **Váratlan aszinkron események:** amelyek keletkezése nem várható (ilyen lehet egy hardverhiba).

**Megszakítások kiszolgálása**

Megszakítások kiszolgálásánál a rendszernek meg kell állapítania a keletkezés helyét, szabályoznia a megszakítási lehetőségeket (egyes utasítások megszakításkérelmi lehetőségének maszkolása). Szükség van a prioritás szabályozására több, egyidőben bekövetkező kérelem kiszolgálási sorrendjének meghatározására, valamint megoldandó feladat a kiszolgáló folyamat közben beérkező kérelmek kezelése is.

**Megszakítás kérés figyelése**

* A processzor szabályos időközönként lekérdezi a perifériát (polling) – időpazarlás
* A periféria megszakításkéréssel jelzi az adatátviteli kérelmet
* Integrált áramköri láb szükséges: Int

Ha Int = 1, a processzor felfüggeszti tevékenységét és a megszakításkezelő-rutint futtatja (Interrupt Service Routine vagy ISR); Megszakítás vezérelt I/O

Alapvetően a megszakítás lekérdezése “int polling” hardveres művelet és nem igényel processzor időt

**A megszakításkezelő program** (ISR) **címének megadása** (interrupt address vector)

* **Rögzített cím** 
  + Hardverben rögzített cím, nem lehet változtatni
  + Az ISR tárolt cím vagy egy ugrás (jump) mutat az ISR címre ha nem elég a rendelkezésre álló memória
* **Vektoros megszakítási cím** 
  + A periféria adja a megszakítás kezelő rutin címét
  + Gyakori abban az esetben ha a microprocesszorhoz több perifériát illesztünk egy rendszersínre
* **Táblázatos megszakításkezelés** (interrupt address table)

**Rögzített ISR cím esetén**

* μP a főprogram végrehajtását végzi
* adat érkezik a periféria, mely megszakítást kér Int jellel
* aktuális utasítás végrehajtása után, μP érzékeli az *Int* jelet és a PC-be betölti a megszakításkezelő szubrutin címet
* az ISR beolvassa a beérkezett adatot, módosítja, kiírja az eredményt
* adat beolvasás után a periféria törli az *Int* jelet
* ISR a PC tartalmát 1-el növeli, a μP folytatja a felfüggesztett program végrehajtását

**Vektoros ISR kezelés**

* μP a főprogram végrehajtását végzi
* adat érkezik a periféria, mely megszakítást kér Int jellel
* aktuális utasítás végrehajtása után, μP érzékeli az *Int* jelet, PC mentése és Inta jellel a megszakítás nyugtázása
* periféria érzékeli az *Inta* jelet és kiteszi a megszakításkezelő rutin címét az adatsínre
* μP a címvezetéken megadott szubrutin kezdőcímre ugrik, az ISR beolvassa az adatokat a címről, módosítja, kiírja az eredményt
* adat beolvasás után a periféria törli az *Int* jelet
* ISR a PC tartalmát 1-el növeli, a μP folytatja a felfüggesztett program végrehajtását

**Táblázatos megszakítás kezelés**

Kompromisszum a rögzített és vektoros megszakítás kezelés között

* Egy megszakítás láb
* A memóriában egy táblázat tartalmazza a megszakításkezelő címeket ISR (~ 256 szó)
* A periféria nem az ISR címet adja hanem a táblázatban lévő indexet

A periféria kevesebb bittel mutat a megszakításkezelő rutin címére

Az ISR cím változtatható a periféria módosítása nélkül

**Maszkolt és nem maszkolt megszakítások**

* **Maszkolt megszakítás:** a programozó beállíthat egy bit értéket amely tiltja a megszakítást
* Valós idejű , idő kritikus feldolgozások esetében fontos
* **Nem maszkolható megszakítás:** olyan megszakítás amely nem tiltható le (külön IC lábbal)
* Végszükség esetén használt megszakítások; például tápfeszültség kimaradás – azonnali beavatkozást igényel - adatok mentése a háttértárolóra

**Ugrás a megszakításkezelő rutinra ISR**

* Egyes mikroprocesszorok a „jump” utasítást úgy kezelik mint a szubrutin „call” hívást
* A teljes állapot elmentése (PC, registers) – időtartama órajel ciklusok százai
* Más μP csak a részleges állapotot mentik (PC állapot),
* Ebben az esetben az ISR nem módisíthatja a regisztereket vagy elmenti a regiszterek tartalmát
* Asszembly programok esetében a regiszterek visszaállítására is figyelni kell

**Cortex-M4 megszakítások**

**Megszakítások típusai:**

* Megszakítás kérések (IRQs)
  + - *Aszinkron:* független a kódtól melyet a processzor éppen futtat

Példák: megerősített megszakítás, karakter érkezik soros porton, ADC konverter befejete az átalakítást

* Nem maszkolható megszakítás (NMI)
  + - hasonló a megszakítás kérésekhez, de nem tiltható le
* Kivételek, Hibák, szoftveres megszakítások
  + - *Szinkron:* adott utasítások végrehajtásának eredménye

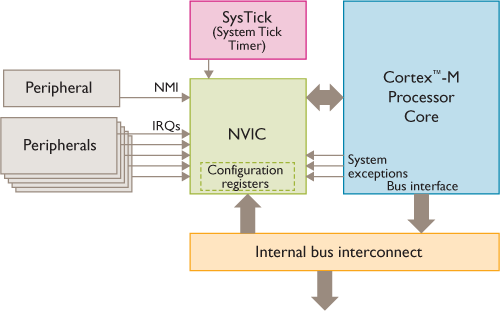
Példák: nem definiált utasítások, adott utasításnál túlcsordulás

* SysTick Timer

**Megszakítás kezelő rutin (ISR)**

Szubrutin mely a processzort egy adott eseményre történő válaszra kényszeríti

**Beágyazott vektoros megszakítás vezérlő (NVIC)**

* NVIC menedzseli és rangsorolja a külső megszakításokat a Cortex-M4 számára
* A megszakítások a kivételek egy típusa (CM4 240 megszakítás kérést támogat)
* Módok
* Szál mód: Reset esetén
* Kezelő mód: kivétel végrehajtás esetén
* Prioritás szintek:
  + Reset: -3 (legmagasabb prioritás)
  + Nem maszkolható megszakítások: -2
  + Hard Fault: -1
* más (perifériás) kivételek prioritása

beállítható

(STM32F4 családnál, prioritás szint 4 bittel kezelhető ami 16 programozható prioritás szintet jelent)

* Stack pointers:
* Main Stack Pointer, MSP
* Process Stack Pointer, PSP
* Kivétel állapotok: inaktív, függőben lévő, aktív, aktív és függőben lévő

**Megszakítások max. gyakorisága:**

Mikor nincs megszakítás a rendszerben:

FMax\_Int = FCPU/(CISR+ COverhead)

FMax\_Int: megszakítások max. frekvenciája

FCPU: CPU órajel frekvencia

CISR: ISR végrehajtásához szükséges ciklusszám

COverhead: overhead ciklusszám (mentés állapot, vektorálás, visszaállító állapot stb.)

Megszakítások esetén:

UInt = 100%\*FInt\* (CISR+COverhead)/FCPU

UInt: feldologzás során felhasznált hasznosítás (processzor idő töredék)