

ARM Cortex magú mikrovezérlők

2. Cortex-M3 mag

Scherer Balázs



Méréstechnika és
Információs Rendszerek
Tanszék

A Cortex-M3 mag

A Cortex-M3 mag és Cortex-M3 processzor

Cortex-M3 mag

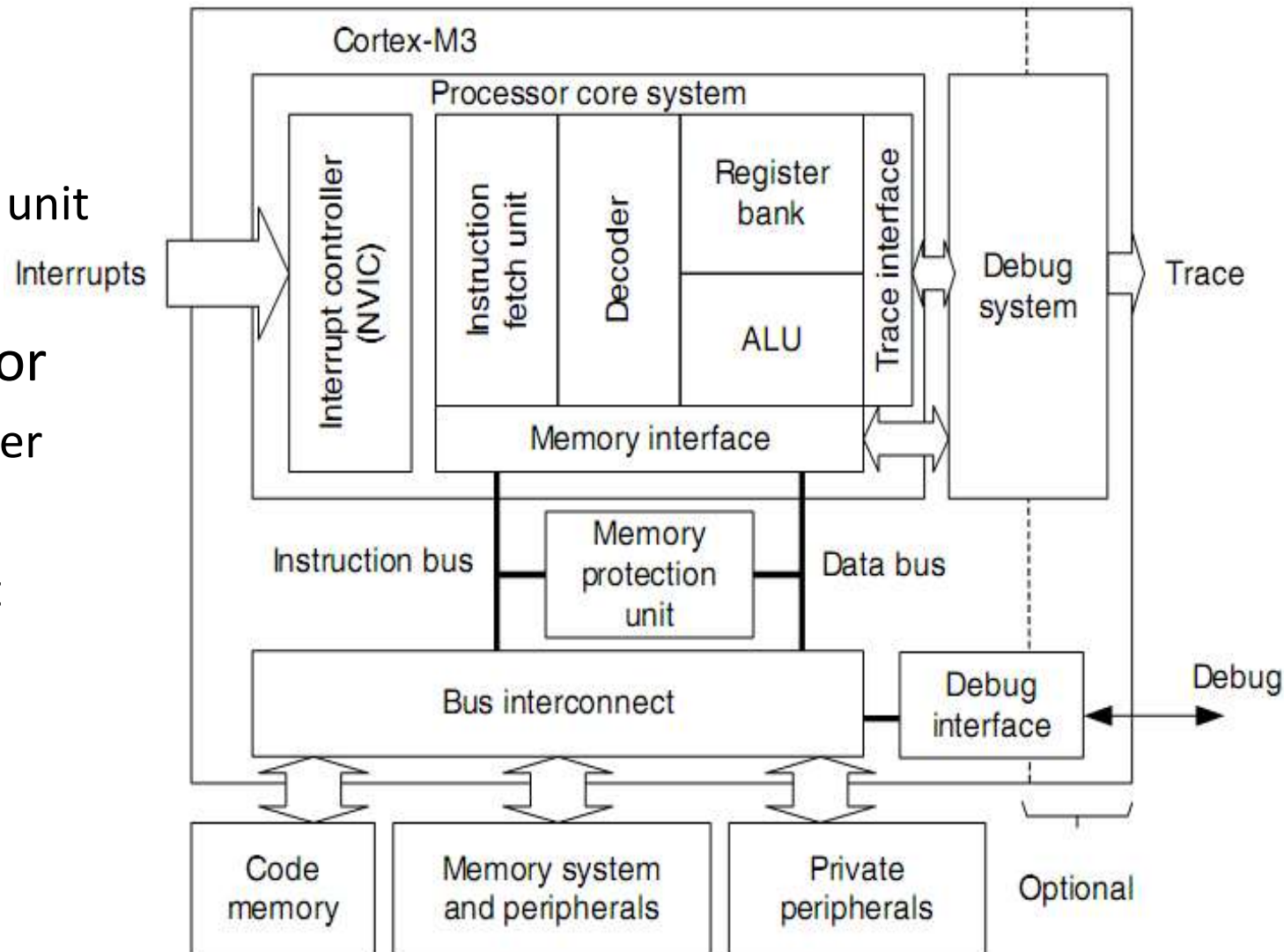
- ALU
- Instruction fetch unit
- Regiszterbank

Cortex-M3 processzor

- Interrupt Controller
- Debug rendszer
- Bus Interconnect

Mikrovezérlő

- Perifériák
- Memória
- Órajel források



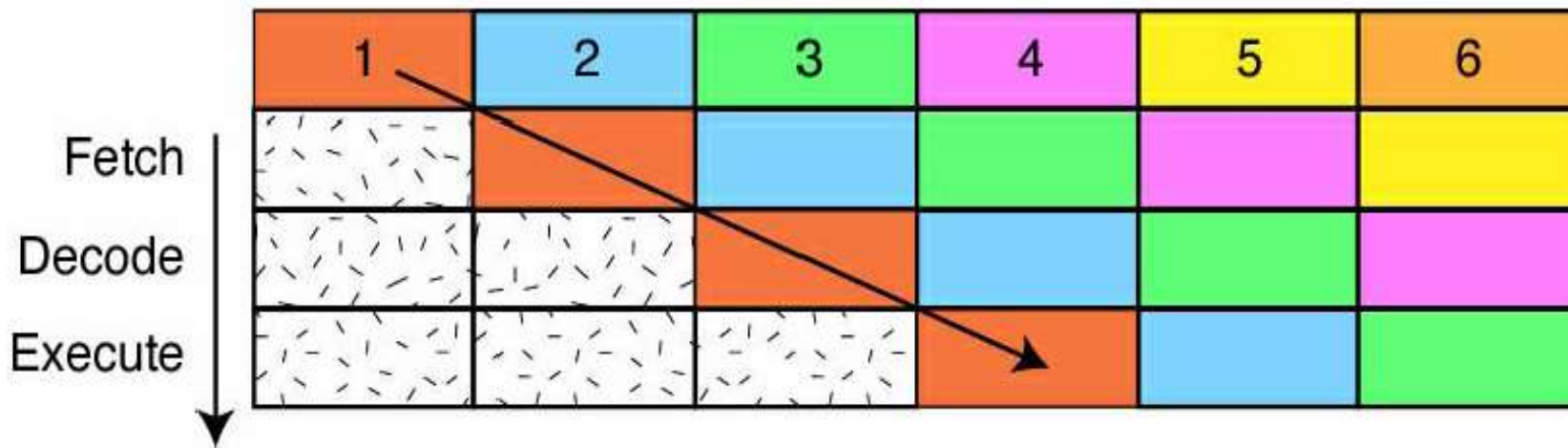
A Cortex-M3 alapok

- ARMv7M architektúra (Az ARM7 az ARMv4-es architektúrát használta.)
- Harvard architektúra
 - Külön utasítás- és adatbusz. Párhuzamos utasítás- és adatfelhozást tesz lehetővé.
- Thumb-2 utasításkészlet, nincs külön ARM és Thumb mód.
 - A Thumb2 16 és 32 bites utasításokat tartalmaz elég nagy kódsűrűséget eredményezve.
- Egyszerűbb programozó modell, mint az ARM7-nél
- Nagyobb teljesítményű utasítások

Utasításvégrehajtás és regiszterek

A Cortex-M3 pipeline-ja

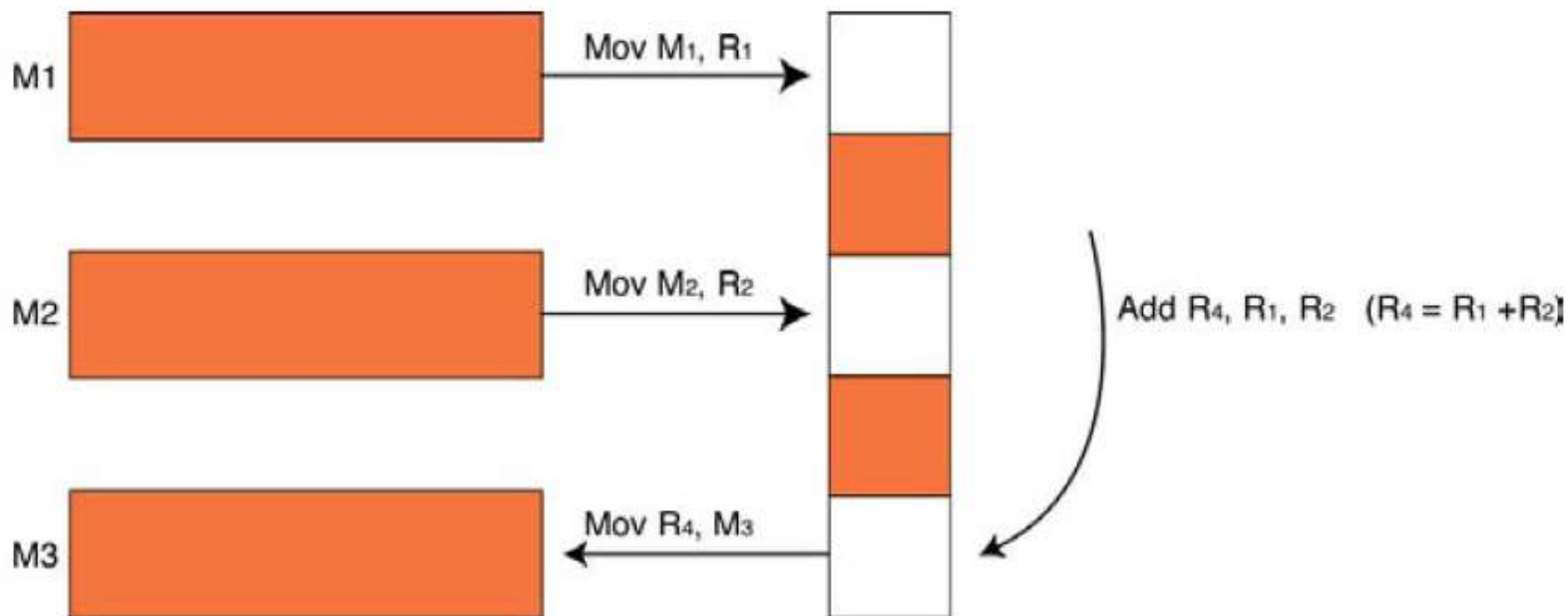
- Háromlépcsős pipeline



- Branch target forwarding: Bizonyos ugró utasításoknál az execute fázis előtt már megkezdí a végrehajtást a rendszer.
 - Pl. feltétel nélküli ugrásnál már a decode fázisban értelmezi az új címet.

A Cortex-M3 programozói modellje

- Load and Store architektúra
 - Minden adatot először be kell mozgatni a memóriából a regiszterekbe. Utána lehet műveletet végrehajtani, majd a végén ki kell írni a műveletek eredményét a regiszterből a memóriába.
 - Semmi különbség az ARM7, ARM9-hez képest.



A Cortex-M3 regiszterei

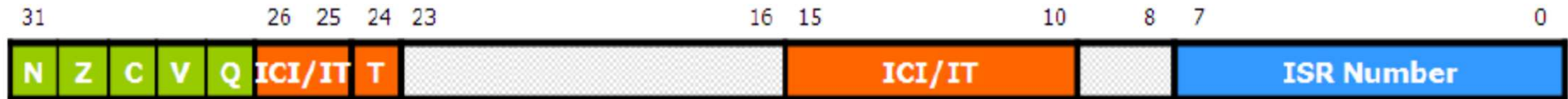
- 16 darab 32 bites regiszter



- Mint az ARM7, ARM9 esetében, itt is:
 - R13 a stack pointer
 - R14 a link regiszter (visszatérési cím)
 - R15 a PC (utasításszámláló, közvetlenül írható olvasható)
- Az R13 egy „bank”-olt regiszter, hogy a processzor két stacket használhasson a Process Stack-et és a Main Stack-et.

Extended program status register

- Az xPSR Nem része a regiszter banknak, csak speciális utasításokkal kezelhető

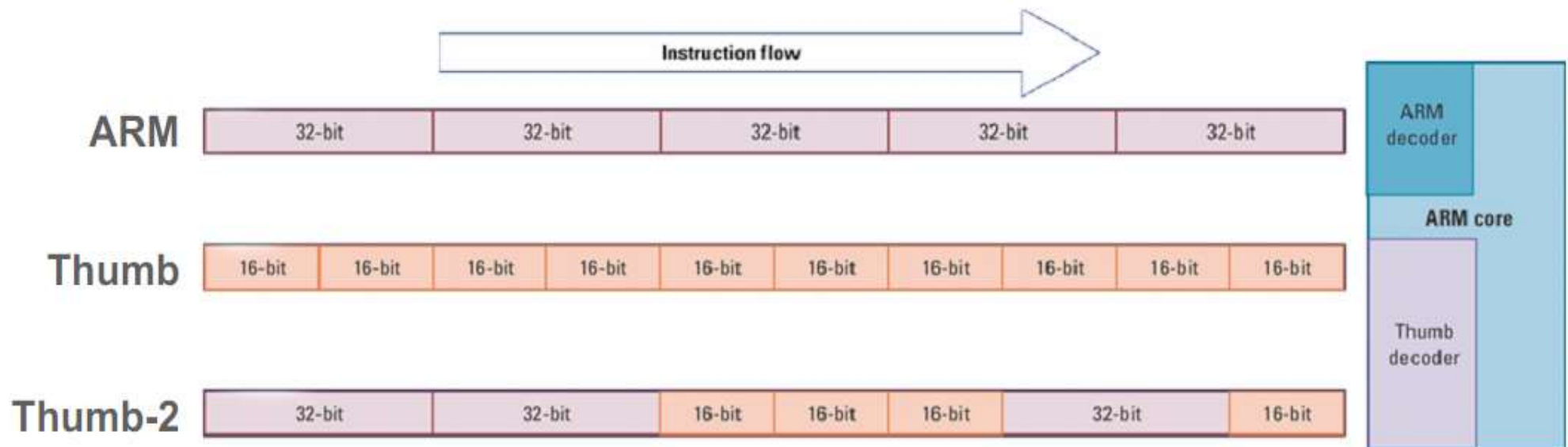


- Van 3 alias regisztere:
- APSR (Application PSR) : Condition code flags (Negatív, Carry, Overflow, Saturated math overflow)
- IPSR (Interrupt PSR): Az aktuális megszakítás száma
- EPSR (Execution PSR),
 - T: Thumb state, mindig 1, nem lehet törölni.
 - IF-THEN field
 - Lehetőség van a Thumb2-ben egyszerű IF-THEN blokkokat megvalósítani
 - Interrupt continuable instruction field
 - PL a Load, Store multiple utasítások nem egy órajel alatt hajtódnak végre
 - A determinikus IT kezeléshez ezeket is meg kell szakítani
 - Ebben a részben tárolja a megszakított Load, Store multiple utasítás következő paraméterét

Utasításkészlet

16 és 32 bites utasítások

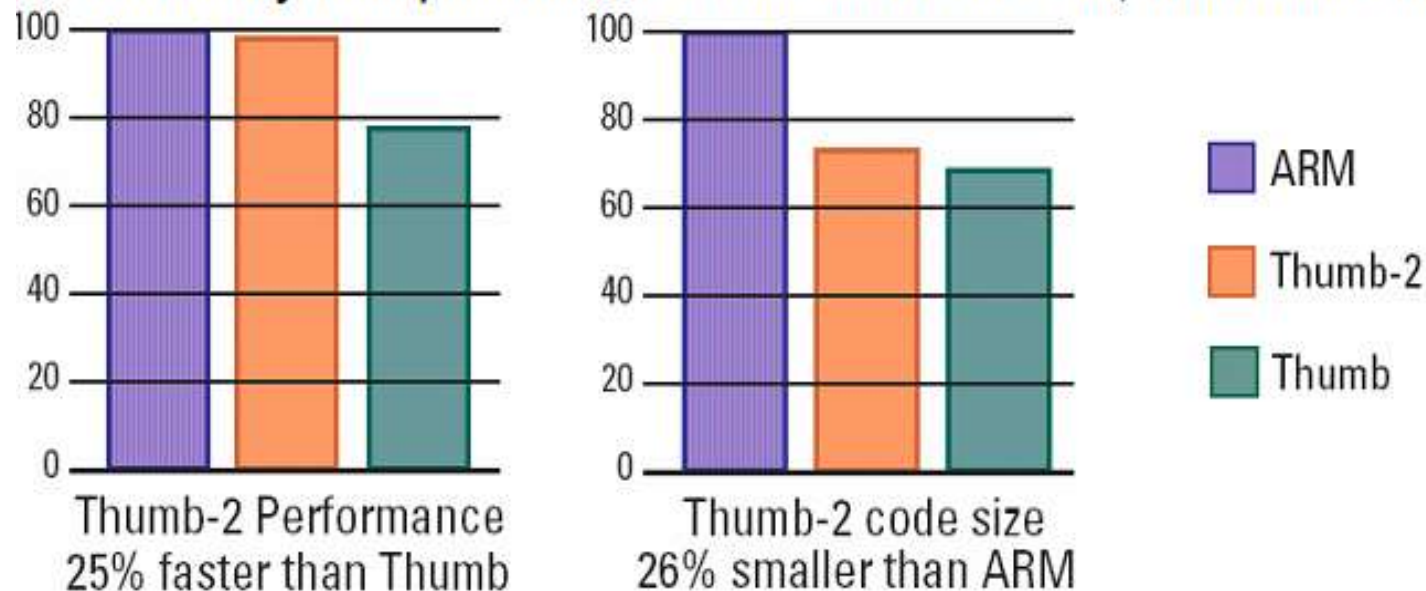
- Nem minden ciklusban kell feltétlenül utasítást felhoznia a magnak.



A Thumb2 utasításkészlet

- Az ARM7, ARM9 processzorokhoz képest csak egy utasításkészlet a Thumb2.
 - 26%-kal tömörebb, mint az ARM 32 bites utasításkészlet.
 - 25%-kal hatékonyabb, mint a hagyományos Thumb.
 - Hardveres szorzás, osztás.

Relative Dhrystone performance and code size for ARM, Thumb and Thumb-2



A Cortex-M3 IF-THEN block

- Maximum 4 utasítást lehet blokkba zárni.
 - IT<x><y><z> <cond>
 - <x><y><z>: lehet T:Then, E: else
 - <cond>normál feltételek: EQ: egyenlő, NE: nem egyenlő stb.

```
if (R0 equal R1) then {  
    R3 = R4 + R5  
    R3 = R3 / 2  
} else {  
    R3 = R6 + R7  
    R3 = R3 / 2  
}
```

This can be written as:

```
CMP    R0, R1      ; Compare R0 and R1  
ITTEE  EQ          ; If R0 equal R1, Then-Then-Else-Else  
ADDEQ  R3, R4, R5  ; Add if equal  
ASREQ  R3, R3, #1  ; Arithmetic shift right if equal  
ADDNE  R3, R6, R7  ; Add if not equal  
ASRNE  R3, R3, #1  ; Arithmetic shift right if not equal
```

A Cortex-M3 IF-THEN block

- Maximum 4 utasítást lehet blokkba zárni
 - IT<x><y><z> <cond>
 - <x><y><z>: lehet T:Then, E: else
 - <cond>normál feltételek: EQ: egyenlő, NE: nem egyenlő stb.

```
if (R0 equal R1) then {  
    R3 = R4 + R5  
    R3 = R3 / 2  
} else {  
    R3 = R6 + R7  
    R3 = R3 / 2  
}
```

- Ha nem kell végrehajtani a blokkot, akkor NOPként hajtódnak végre.
- Segíti a pipe-line kitöltését, nem kell üríteni, újra tölteni az ugrások miatt.

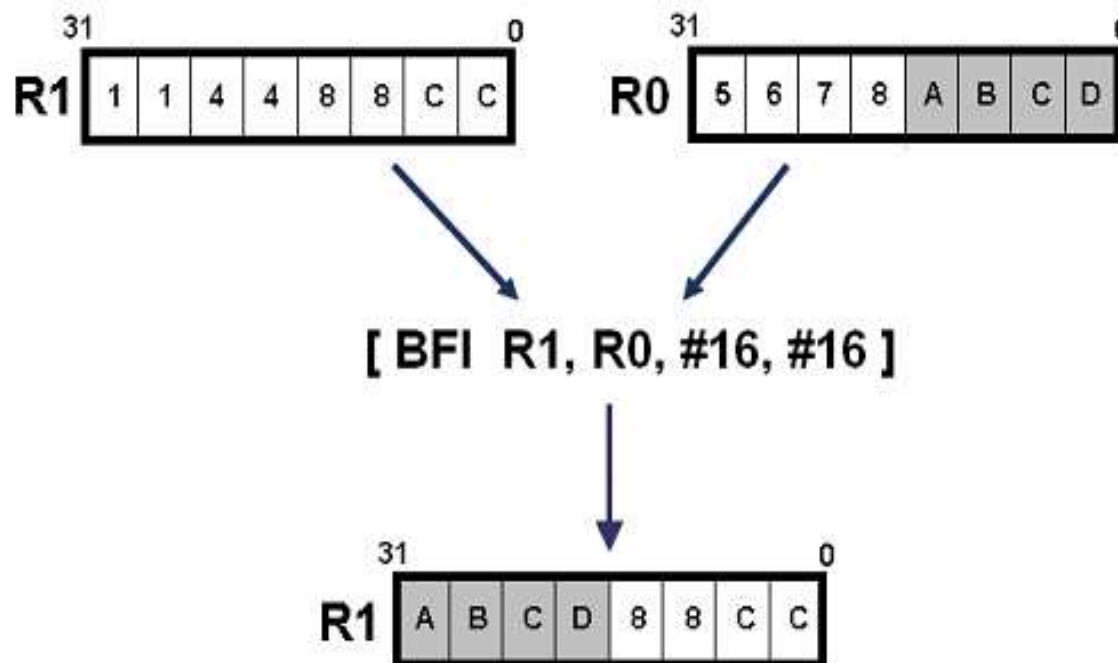
This can be written as:

```
CMP    R0, R1      ; Compare R0 and R1  
ITTEE  EQ          ; If R0 equal R1, Then-Then-Else-Else  
ADDEQ  R3, R4, R5  ; Add if equal  
ASREQ  R3, R3, #1  ; Arithmetic shift right if equal  
ADDNE  R3, R6, R7  ; Add if not equal  
ASRNE  R3, R3, #1  ; Arithmetic shift right if not equal
```

Bitmezőtörlés és -beszúrás

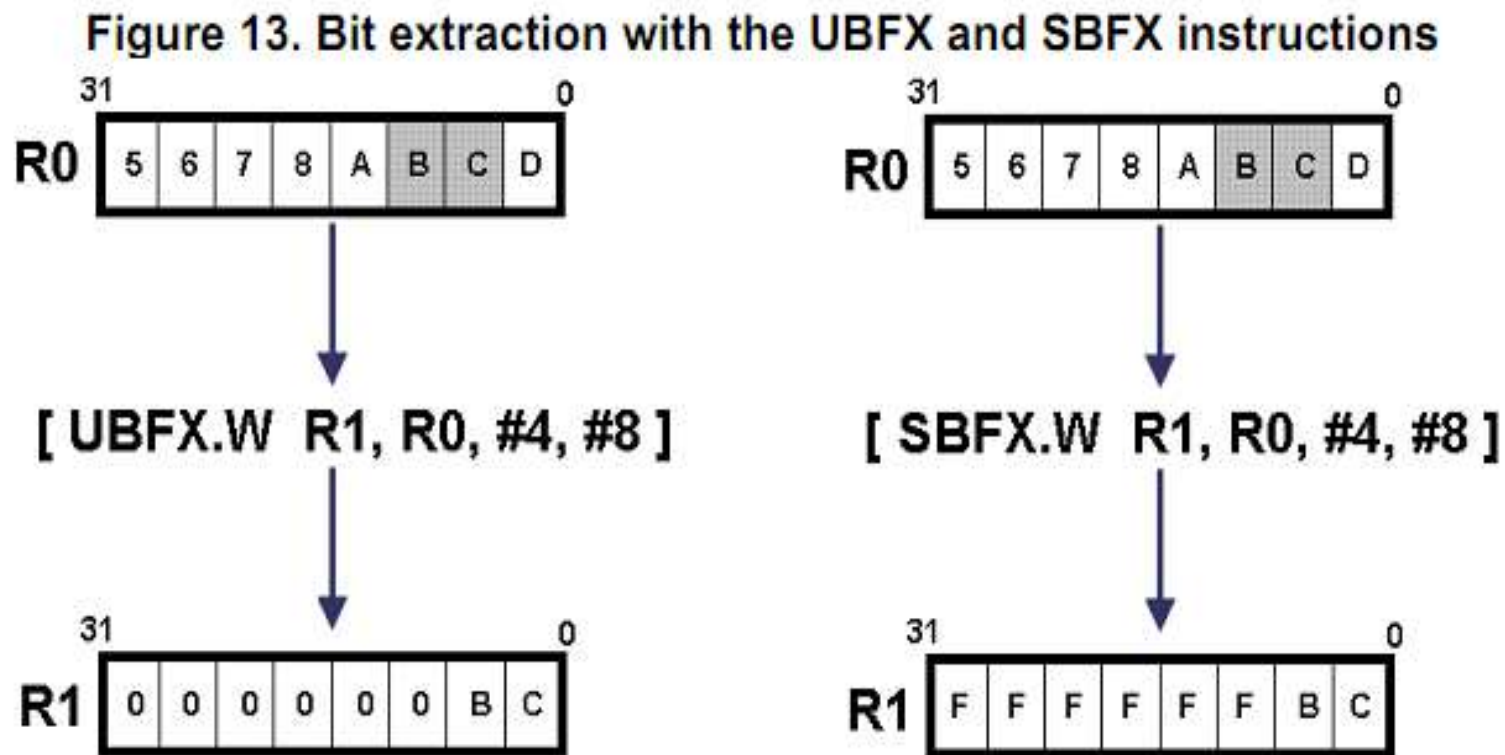
- Bit Field Clear (BFC), bármely bitmező törölhető.
- Bit Field Insert (BFI), tetszőleges bitmező beszúrható egy másik regiszterből.

Figure 15. Insertion of any number of adjacent bits using BFI



Bitkinyerő utasítások

- UBFX (zero extend) és SBFX (sign extend)



Egyéb különleges utasítások

- Bit és byte sorrend felcserélése
- 64 bites adatok kétregiszteres transzferje
 - Egy utasításként mozgathatóak.
- Egy utasításos ugrótábla
- Előjeles vagy előjel nélküli osztás

Belső perifériák

System Timer

- 24 bites lefelé számláló
- Egységesített rendszerszámláló a Cortex-M3 core-ra épülő mikrovezérlőkhöz.
- Rendszer órajelről vagy annak 1/8-áról mehet.
- Három regiszter
 - Számláló
 - Reload
 - Status:
 - IT engedélyezés
 - Timer konfiguráció, Start stop

System Timer

- 24 bites lefelé számláló
- Egységesített rendszerszámláló a Cortex-M3 core-ra épülő mikrovezérlőkhöz.
 - Elsősorban RTOS Heart-beat timer-nek szánták.
- Rendszer órajelről vagy annak 1/8-áról mehet
- Három regiszter
 - Számláló
 - Reload
 - Status:
 - IT engedélyezés
 - Timer konfiguráció, Start stop

Cortex-M3 NVIC

- Nested Vector Interrupt Controller
 - Gyártófüggetlen standard tartozék, ebből következően gyártófüggetlen interrupt-struktúra.
 - Könnyű portolhatóság
 - A Thumb2 utasításkészlet több órajelig tartó utasításai megszakíthatóak, így az IT-kezelés determinisztikus.
 - Nested interruptokat támogatja
 - Az STM32-n 16 prioritási szint van.
 - Bár az NVIC processzorfüggetlen, az erőforrás-használat minimalizálása miatt a processzortervezők megszabhatják az NVIC bemenő vonalainak számát.
 - Az NVIC képes: 1 nem maszkolható +240 külső periféria + 15 belső Cortex-es IT vonal forrást kezelni.
 - Az STM32 43-at használ.

Az NVIC ugrótábla

- Az ugrótábla a címtartomány alján a 0x00000004-ről indul.
 - A 0x00000000-án a kezdő stack pointer van, hogy minél hamarabb lehessen C-t használni.

No.	Exception Type	Priority	Type of Priority	Descriptions
1	Reset	-3 (Highest)	fixed	Reset
2	NMI	-2	fixed	Non-Maskable Interrupt
3	Hard Fault	-1	fixed	Default fault if other handler not implemented
4	MemManage Fault	0	settable	MPU violation or access to illegal locations
5	Bus Fault	1	settable	Fault if AHB interface receives error
6	Usage Fault	2	settable	Exceptions due to program errors
7-10	Reserved	N.A.	N.A.	
11	SVCall	3	settable	System Service call
12	Debug Monitor	4	settable	Break points, watch points, external debug
13	Reserved	N.A.	N.A.	
14	PendSV	5	settable	Pendable request for System Device
15	SYSTICK	6	settable	System Tick Timer
16	Interrupt #0	7	settable	External Interrupt #0
.....			settable	
256	Interrupt#240	247	settable	External Interrupt #240

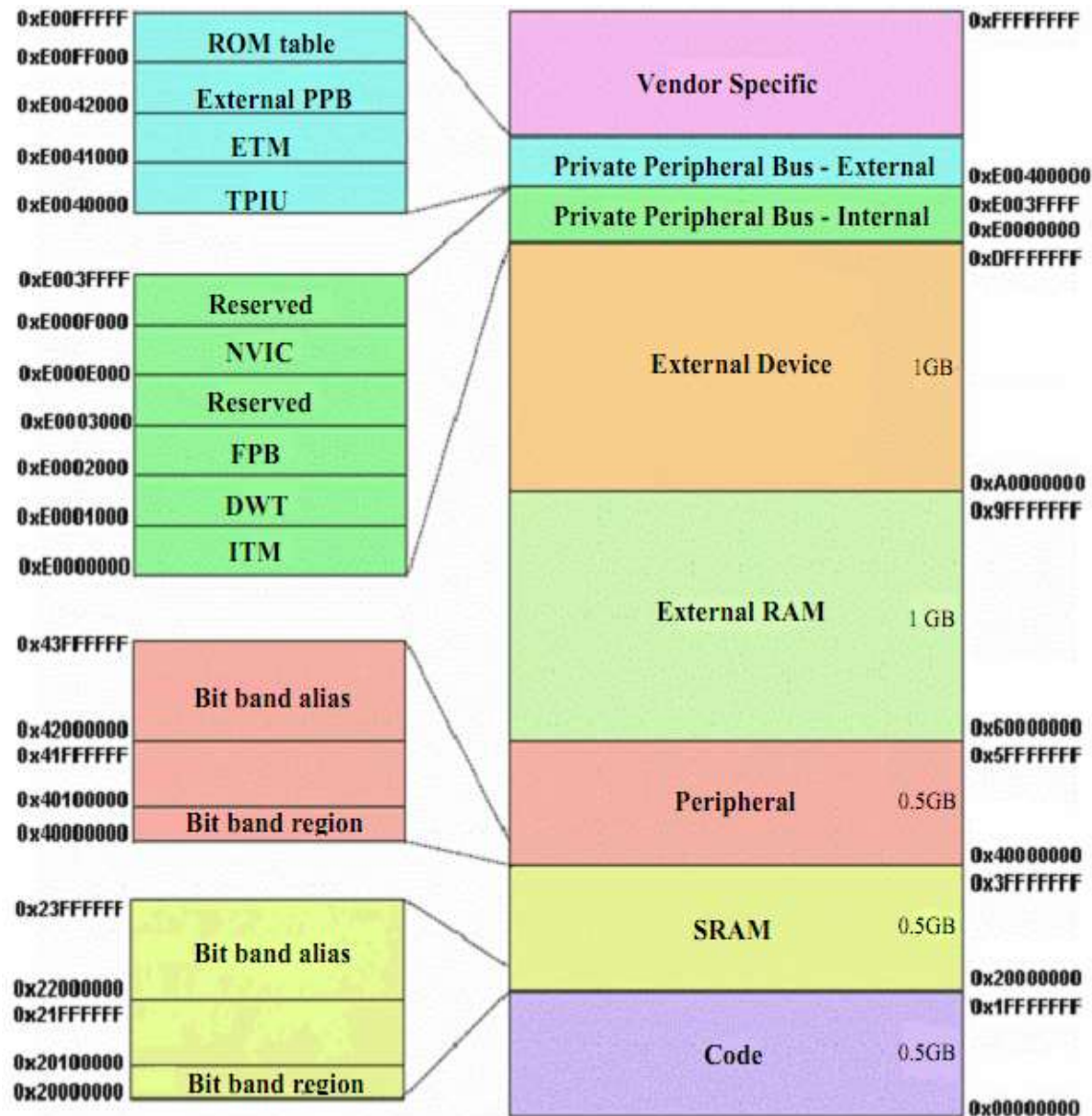
Gyártó-
specifikus

Memóriaelrendezés

A Cortex-M3 memóriatérképe

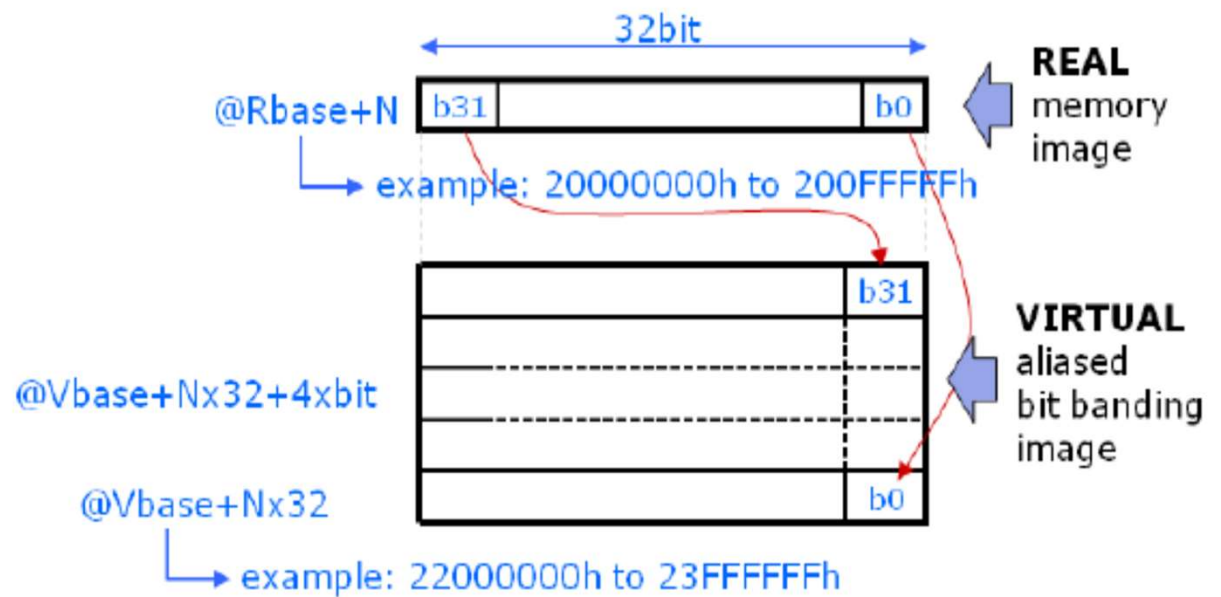
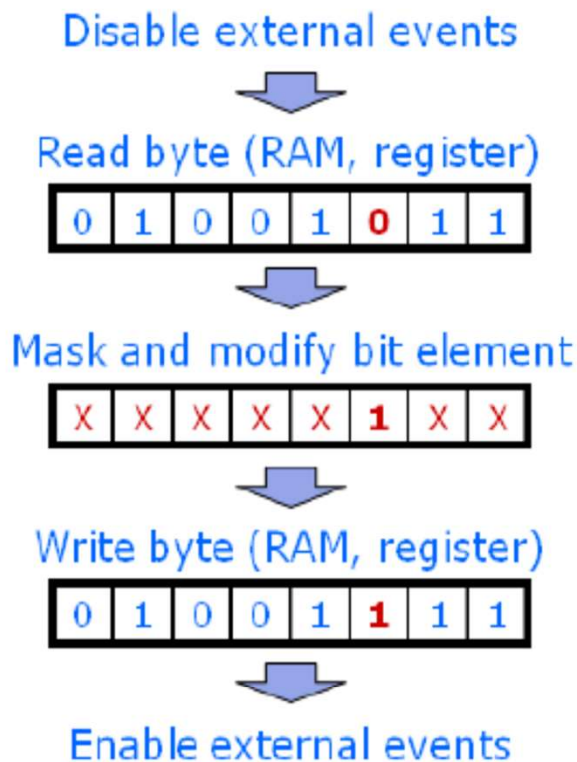
- Az ARM7, ARM9-essel ellentétben itt pontosan specifikálva van az egységes memóriatérkép.
- 4 Gbyte címtartomány
 - 1 Gbyte Code és SRAM terület
 - 0,5 G Code terület optimalizálva az I-Code busz számára
 - 0,5 G SRAM terület
 - Code lehet az SRAM-ból is végrehajtva (lassabb)
 - 0,5 Gbyte On chip periféria
 - 2 Gbyte külső memória és külső egységek
 - 0,5 Gbyte Cortex regiszterek és mikrovezérlő gyártóspecifikus részek

Előre kiosztott memóriatartományok



Bit banding

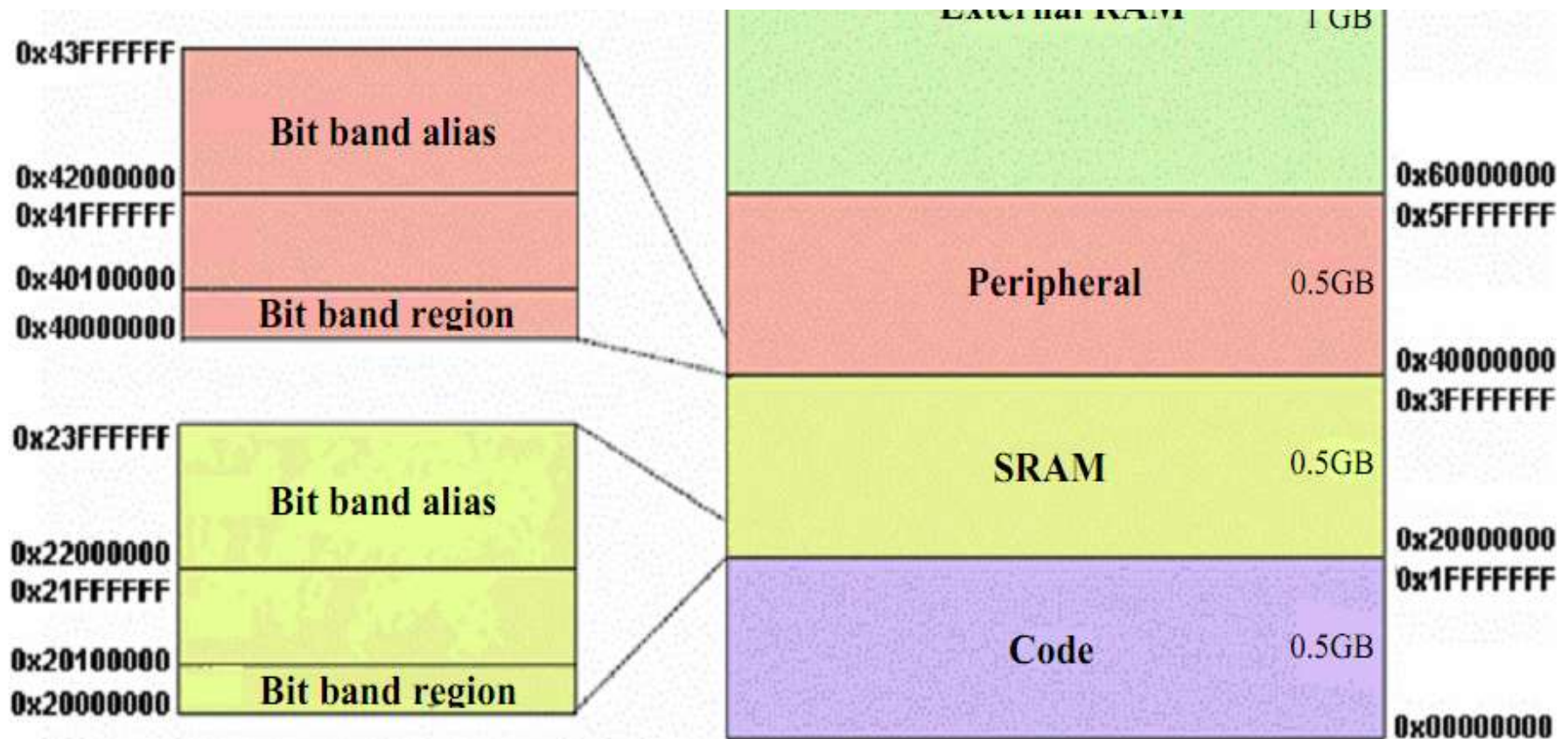
- Direkt bitvezérlés, nincs szükség AND, OR maszkolásra.



The bit banding technique allows atomic bit manipulation while keeping the Cortex-M3 CPU to a minimal gate count.

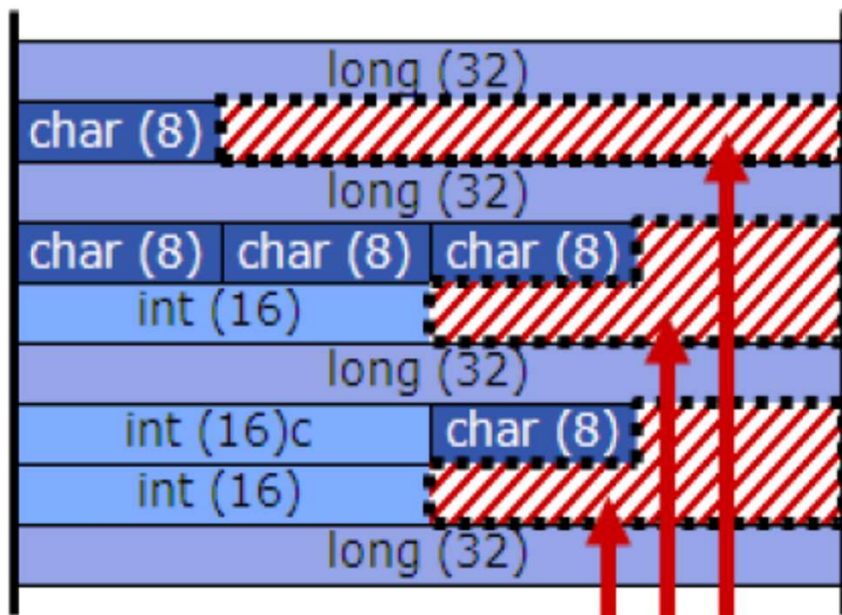
Bit band memóriaterületek

- Az SRAM és a perifériablokk első 1 Mbyte-ja
 - Nincs szükség többre.

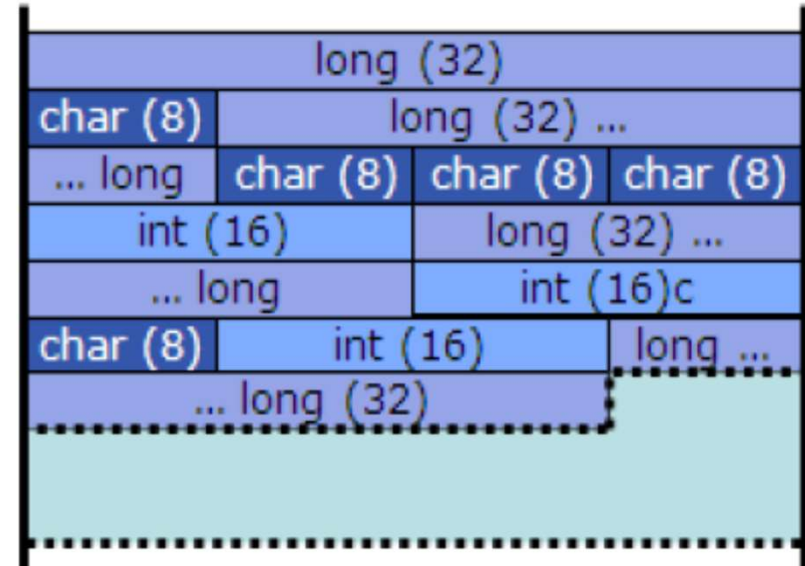


Alignment nélküli memória-hozzáférés

- ARM7,ARM9 aligned (csak adott címeken kezdődhetnek), Cortex-M+ non-aligned memóriakezelés
 - Kellemetlenségek (Vektor CCP), kihasználatlan terület (akár 25%)



Unused (wasted) space



Hozzáférési és működési módok

A Cortex-M3 hozzáférési módjai

- A Cortex-M3 kétféle hozzáférési módot specifikál
 - Privileged működés
 - Nevezik supervizor módnak is.
 - Automatikusan ez aktív reset után.
 - Kivétel vagy interrupt hatására automatikusan ebbe lép a processzor.
 - Minden processzor-erőforráshoz hozzáférést biztosít.
 - Unprivileged operation
 - Hívják felhasználói hozzáférési módnak is.
 - Korlátozott hozzáférés
 - Néhány utasítástípus letiltva, például az xPSR-t manipulálók.
 - Nem lehet a System Control Space (SCS) regiszterekhez hozzáférni. Ilyen például az NVIC (vektoros interrupt-kezelő) és SysTick (Rendszer timer)

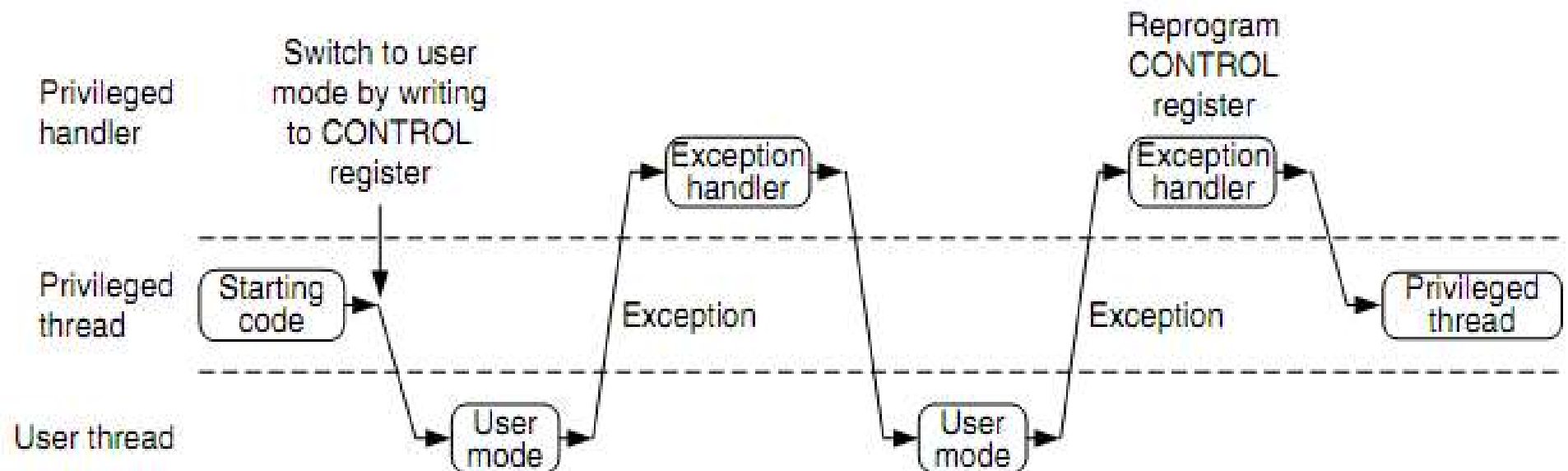
A Cortex-M3 működési módjai

- Egyszerűsített két működési mód
 - Thread mód:
 - Normál működés
 - Lehet privilegizált vagy nem privilegizált hozzáférésben.
 - Ha átváltja magát nem privilegizáltba, csak a Handler tudja visszaváltani.
 - Handler mód:
 - Kivételkezelés
 - Interrupt-kezelés
 - Mindig privilegizált végrehajtás

A Cortex-M3 stackjei

- Main stack
 - Operációs rendszer és a kivételek részére fenntartva.
- Process stack
 - Elsősorban a thread mód számára (A Thread mód használhatja a Main stacket is. Szoftverben választható, hogy a Thread mód a Main vagy a Process stacket használja).
- A jól különválasztott stack biztosíthatja, hogy az alkalmazás nem tudja kilőni az oprendszert, ez biztonságkritikus rendszereknél nagyon fontos.

Minta a privilégizált mód használatára



Hozzáférési működési módok

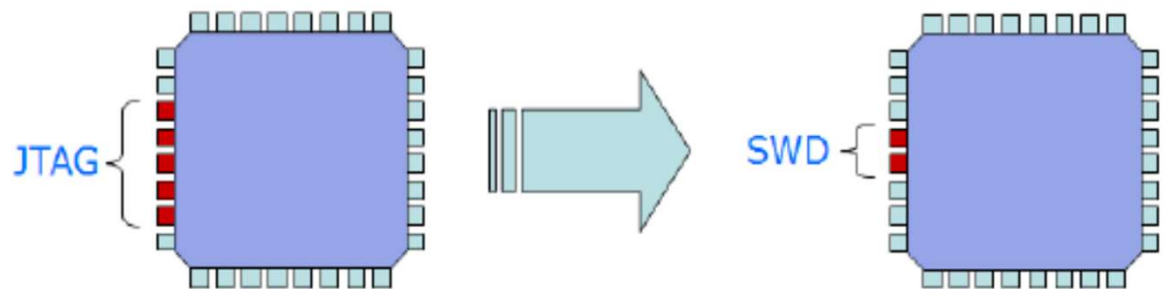
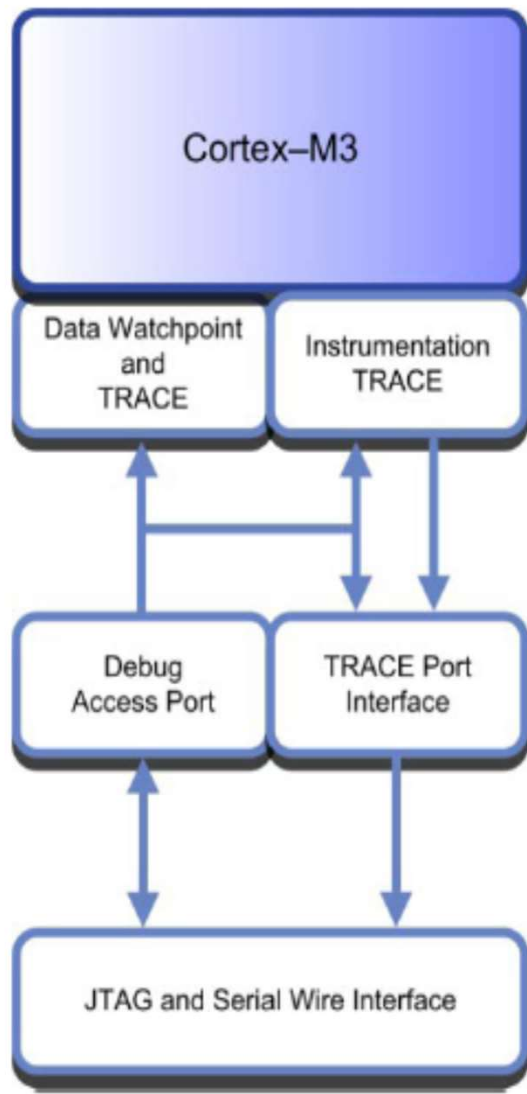
		Operations (privilege out of reset)	Stacks (Main out of reset)
	Handler - An exception is being processed	Privileged execution Full control	Main Stack Used by OS and Exceptions
	Thread - No exception is being processed - Normal code is executing	Privileged/Unprivileged	Main/Process

Egyéb specialitások

Alacsony fogyasztási módok

- Sleep mód
 - Processzor inaktív
 - Az NVIC egy része aktív, így a processzor felébreszthető.
- Sleep now
 - WFI utasítás: Wait For Interrupt
 - Power down-ba megy, és egy megszakítás ébreszti fel.
 - WFE utasítás: Wait For Event
 - Periféria megszakítás vonal, ami felébreszti a processzort, de nem kell tényleges IT kiszolgálást csinálni, hanem a főprogram folytatódik.
- Sleep on Exit
 - IT után rögtön visszaalszik megint, takarékos eseményvezérelt működés.
- Deep Sleep
 - A Cortex-M3 core jelzi a külső gyártóspecifikus egységeknek, hogy menjenek energiatakarékos módba. Ilyenkor lehet a perifériákat és a PLL-t lekapcsolni.

Debug rendszer



The Cortex CoreSight debug system uses a JTAG or serial wire interface. CoreSight provides run control and trace functions. It has the additional advantage that it can be kept running while the STM32 is in a low power mode. This is a big step on from standard JTAG debugging.