

ARM – Cortex M

1. Vývoj ARM procesorov

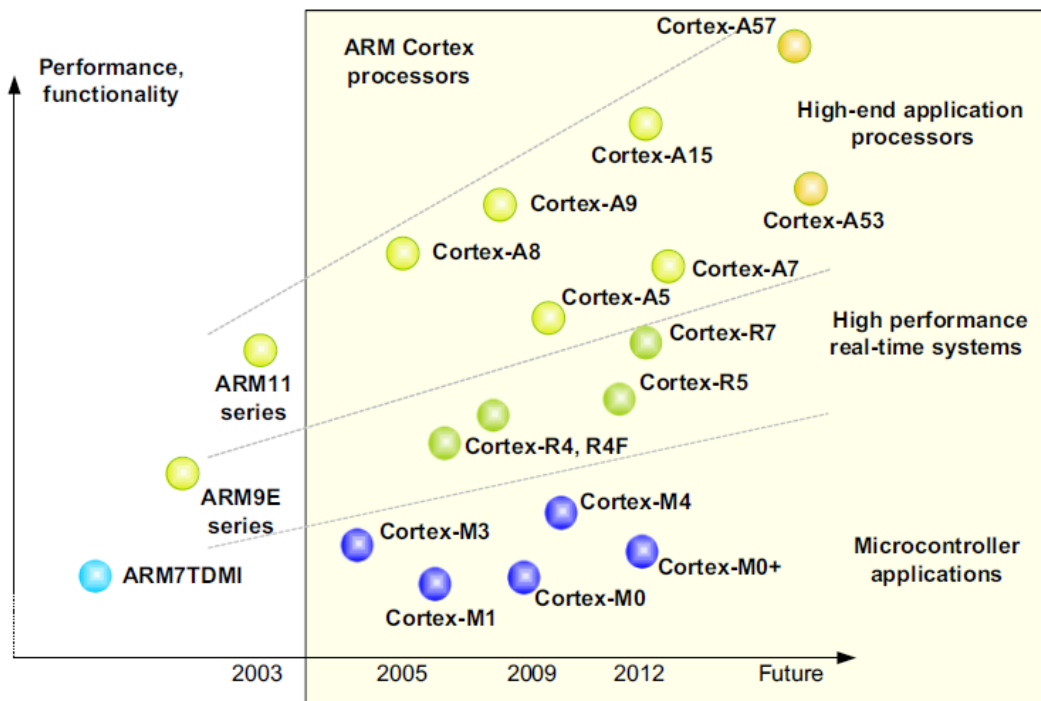
ARM bola založená v roku 1990 ako **Advanced RISC Machines Ltd.**, spoločný podnik medzi Apple Computers, Computer Group a VLSI Technology.

Na rozdiel od mnohých polovodičových firiem, ARM nevyrába procesory a nepredáva čipy priamo, ale ARM dáva licencie obchodným partnerom, vrátane väčšiny z popredných svetových polovodičových výrobcov, na procesorový návrh s využitím technológie ARM. Výrobcovia s licenciou následne navrhujú, vyrábajú a predávajú svoje procesory a mikropočítačové systémy, ktoré dodržiavajú podmienky týchto licencovaných návrhov. Takýto obchodný model je bežne nazýva IP licencia.

ARM označoval o začiatku svoje procesory číslom série napr. ARM5, ARM6, ARM7,...ARM11 alebo verzie architektúry (ARMv5, ARMv6, ARMv7,...).

Počas niekoľkých posledných rokov, ARM rozšíril svoje produktové portfólio rozčlenením návrhu CPU, čím vznikol nový názov procesorov "Cortex." Procesory sú rozdelené do troch skupín:

1. **CORTEX-A** - aplikačné procesory sú navrhované pre vysoko výkonné aplikačné platformy, pre komplex = operačné systémy + užívateľské aplikácie (napr.: iOS, Android, Linux a Windows)
2. **CORTEX-R** - procesory sú navrhované pre systémy, ktoré potrebujú výkon v reálnom čase „Real-Time Systems“ a krátku dobu reakcie
3. **CORTEX-M** – procesory sú navrhované pre „microcontroller-type systems“ –mikropočítačové systémy (mikrokontroléry), kde je potrebný dostatočný výkon procesora, rýchla odozva na prerušenie (low interrupt latency), ale dôležitým kritériom je aj nízka cena a nízka spotreba energie.



Obrázok 1: Rozčlenenie procesorov ARM Cortex

ARM – Cortex M

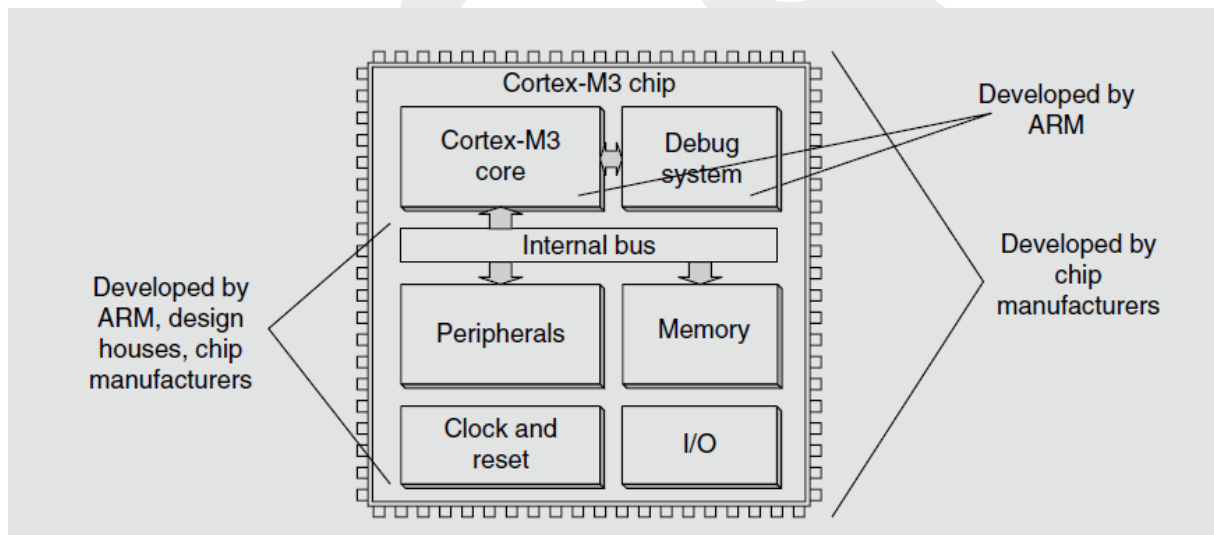
2. ARM Cortex-M3

Procesor Cortex-M3 je jeden z procesorov, ktoré sú založené na architektúre **ARMv7**.

2.1. Cortex-M3 procesor a mikropočítačový systém (mikrokontrolér) na báze Cortex M3 procesora

Procesor Cortex - M3 je centrálna procesorová jednotka (Central Processing Unit - CPU), ktorá sa nachádza v čipe mikrokontroléra.

Okrem toho je treba celý rad ďalších prvkov pre mikrokontrolér. Potom, čo výrobcovia čipov získajú licenciu na procesor Cortex-M3, môžu urobiť svoj vlastný návrh mikrokontroléra, pridať pamäť, periférie, vstupy/výstupy (I / O), a ďalšie funkcie (viď. obrázok).

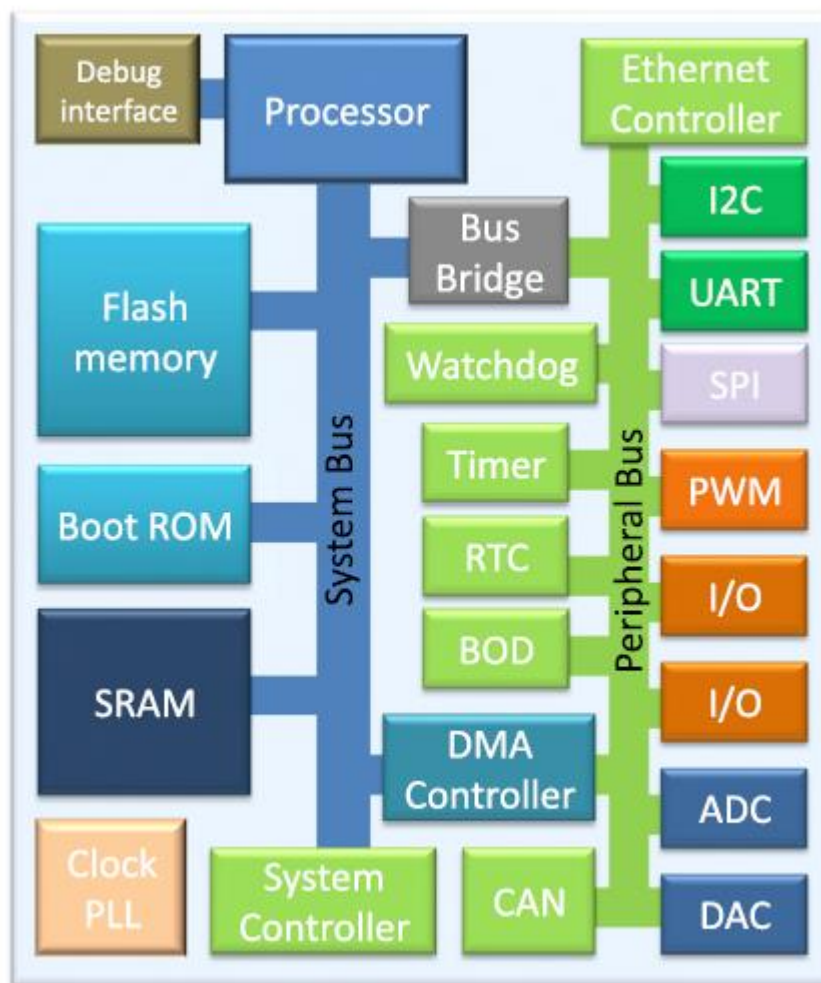


Obrázok 2: Cortex-M3 procesor a mikrokontrolér na báze Cortex M3 procesora

Mikrokontroléry na báze Cortex-M3 procesorov od rôznych výrobcov môžu mať rôzne veľkosti pamäte, typy, periférie, a funkcie. Spoločné alebo rovnaké v mikrokontroléroch je len to, čo je dané licenciou (z obrázku: Developed by ARM)

ARM – Cortex M

Schéma mikrokontroléra na báze Cortex-M3



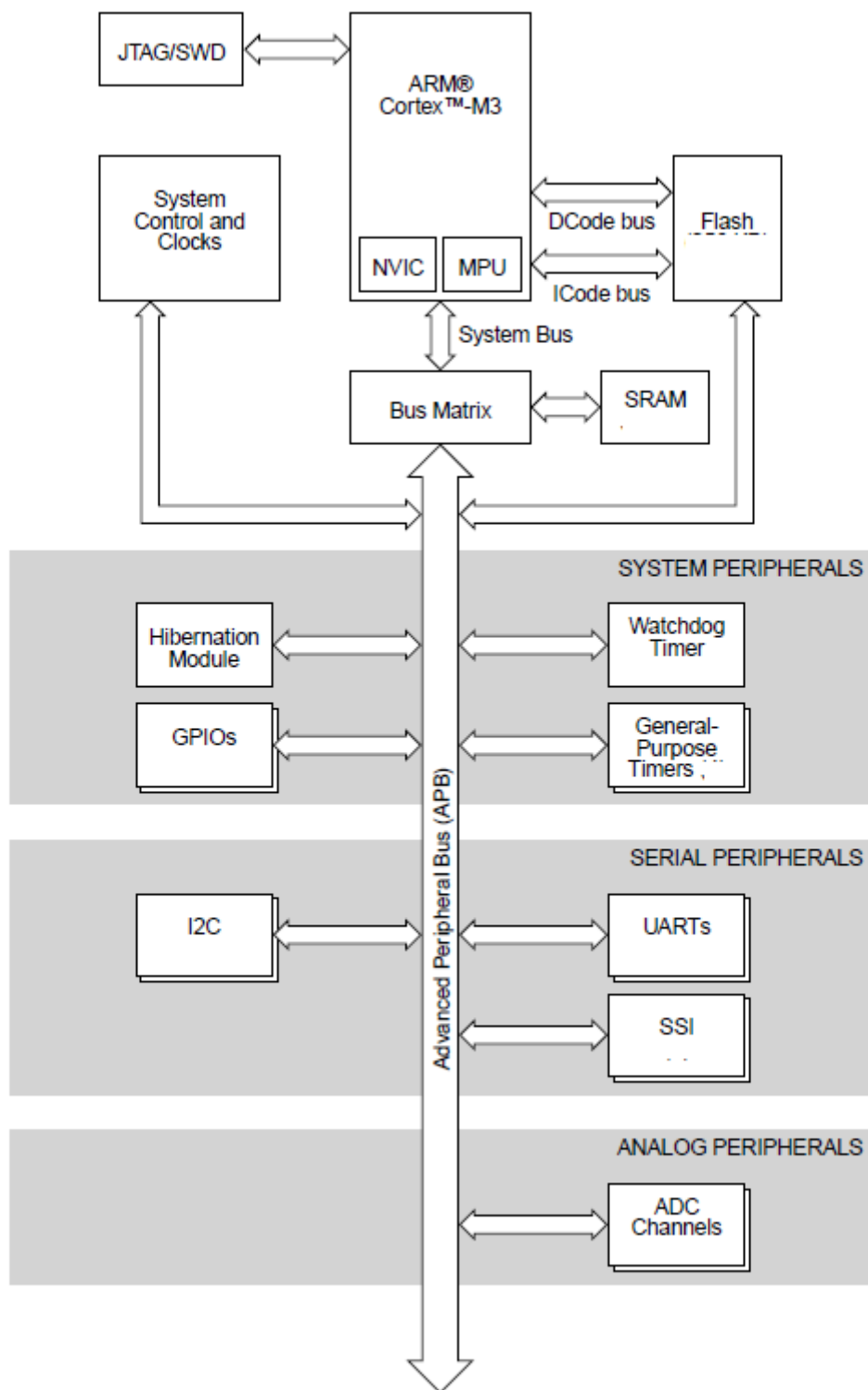
Obrázok 3: Navrhnutý mikrokontrolér môže obsahovať mnoho rôznych blokov

Mikrokontrolér na báze procesora Cortex-M3:

- 1) mikroprocesor Cortex-M3 (CPU) – získaný licenciou
- 2) pamäťový systém (32 bitový adresný rozsah bez jednotky Virtual Memory Manager)
- 3) zbernicový systém (I-bus, D-bus, S-bus)
- 4) periférie
 - systémové (watchdog, časovače, programovateľné GPIOs. ...)
 - sériové (zbernice UART, SSI, I2C, CAN,...)
 - analógové (A/D, D/A prevodníky)

ARM – Cortex M

Príklad blokovej schémy mikrokontroléra na báze Cortex-M3 vyrábanej spoločnosťou Texas Instruments (t.j. konkrétny vyrábaný mikrokontrolér s výrobcom určenou veľkosťou pamäte a perifériami)



Obrázok 4: Bloková schéma ARM CORTEX-M3 - Texas Instruments

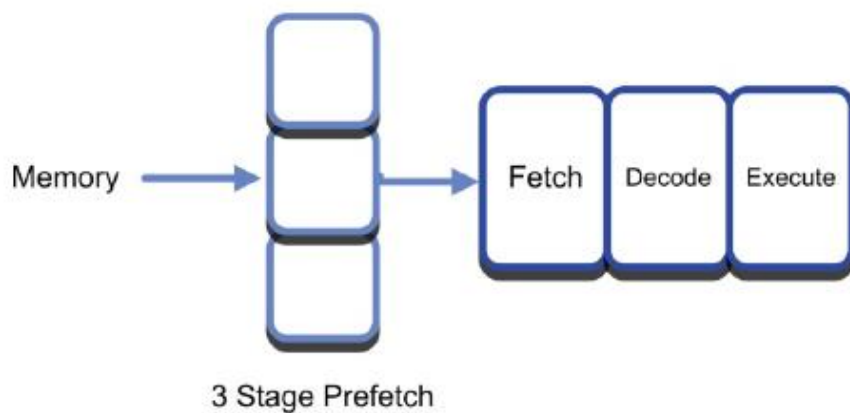
ARM – Cortex M

2.2. Prehľad procesora Cortex-M3 - CPU

Cortex-M3 je **32 bitový** procesor, ktorý má **Harvardskú architektúru** t.j. má oddelenú pamäť pre program (kód) a pamäť pre dáta, má zvlášť zbernicu do pamäte programu a zbernicu do pamäte dát, čo umožňuje paralelné čítanie inštrukcií a čítanie/zápis dát.

Súčasne využíva trojstupňové prúdové spracovania inštrukcií (**pipeline**), vďaka čomu je schopný vykonávať viac operácií súčasne a tým sa dosahuje **vyšší výkon systému**.

Pipeline



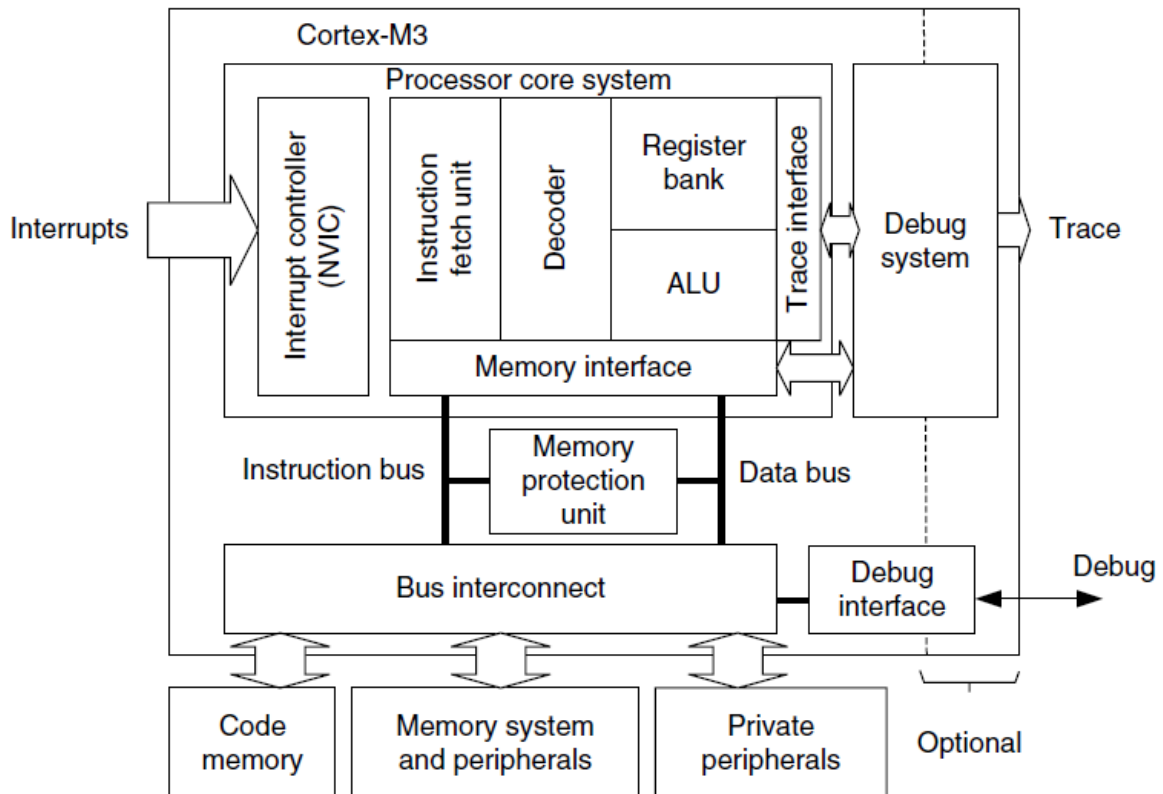
Obrázok 5: Trojstupňové prúdové spracovania inštrukcií (pipeline)

Trojstupňové prúdové spracovania inštrukcií (pipeline) **vykonáva súčasne v 1 hodinovom cykle** viac operácií. Pokiaľ sa 1 inštrukcia vykonáva (execute state), nasledujúca inštrukcia sa dekóduje (decode state) a ďalšia sa načítava (vzdvihuje/vyberá) z pamäte (fetch state).

V nasledujúcom hodinovom cykle sa inštrukcia, ktorá sa v predchádzajúcom cykle dekodovala, vykonáva (execute state). Inštrukcia, ktorá sa v predchádzajúcom cykle načítavala, sa dekóduje (decode state) a ďalšia sa načítava z pamäte (fetch state). A takto to pokračuje.

ARM – Cortex M

Jadro Cortex-M3 - CPU



Obrázok 6: Bloková schéma CPU

Cortex-M3 obsahuje:

- radič prerušení (NVIC - Nested Vectored Interrupt Controller) – riadi prerušovací systém procesora (prerušovací systém si vysvetlíme neskôr)
- jednotku načítania inštrukcií (Instruction fetch unit) – **vykonáva fetch state**
- dekodér inštrukcií – **vykonáva decode state**
- registre
- aritmeticko-logickú jednotku (ALU) – ALU vykonáva aritmeticko-logické operácie nad dátami uloženými v registroch – **vykonáva execute state**
- pamäťové rozhranie (Memory interface) – slúži na komunikáciu jadra procesora s pamäťou (zabezpečuje, aby si jadro procesora s pamäťou rozumeli)
- rozhranie pre krokovanie programu (Trace interface)
- podporu a rozhranie pre funkcie ladenia programu (Debug interface)
- voliteľnú jednotku ochrany pamäte (Memory protection unit)

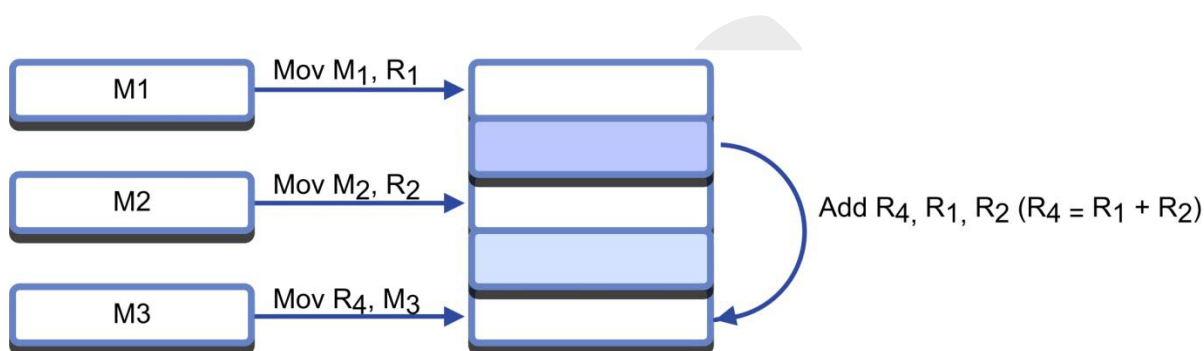
(Obsah Cortex-M3 si kontrolujte podľa obrázka)

ARM – Cortex M

2.3. Programátorský model

Cortex-M3 CPU je RISC procesor (Reduced instruction set computer - procesor s redukovanou inštrukčnou sadou), ktorý využíva Thumb-2 inštrukčný súbor. Thumb-2 obsahuje zmes 16-bitových a 32-bitových inštrukcií. Obsahuje v sebe inštrukcie jednotaktové, ale i load and store inštrukcie, ktoré sú viactaktové.

Load and store architektúra (načítaj a ulož) T.j. aby mohli byť vykonané inštrukcie, operandy musia byť načítané do súboru registrov procesora, operácia je vykonaná nad týmito registrami a výsledok je uložený späť do pamäte.

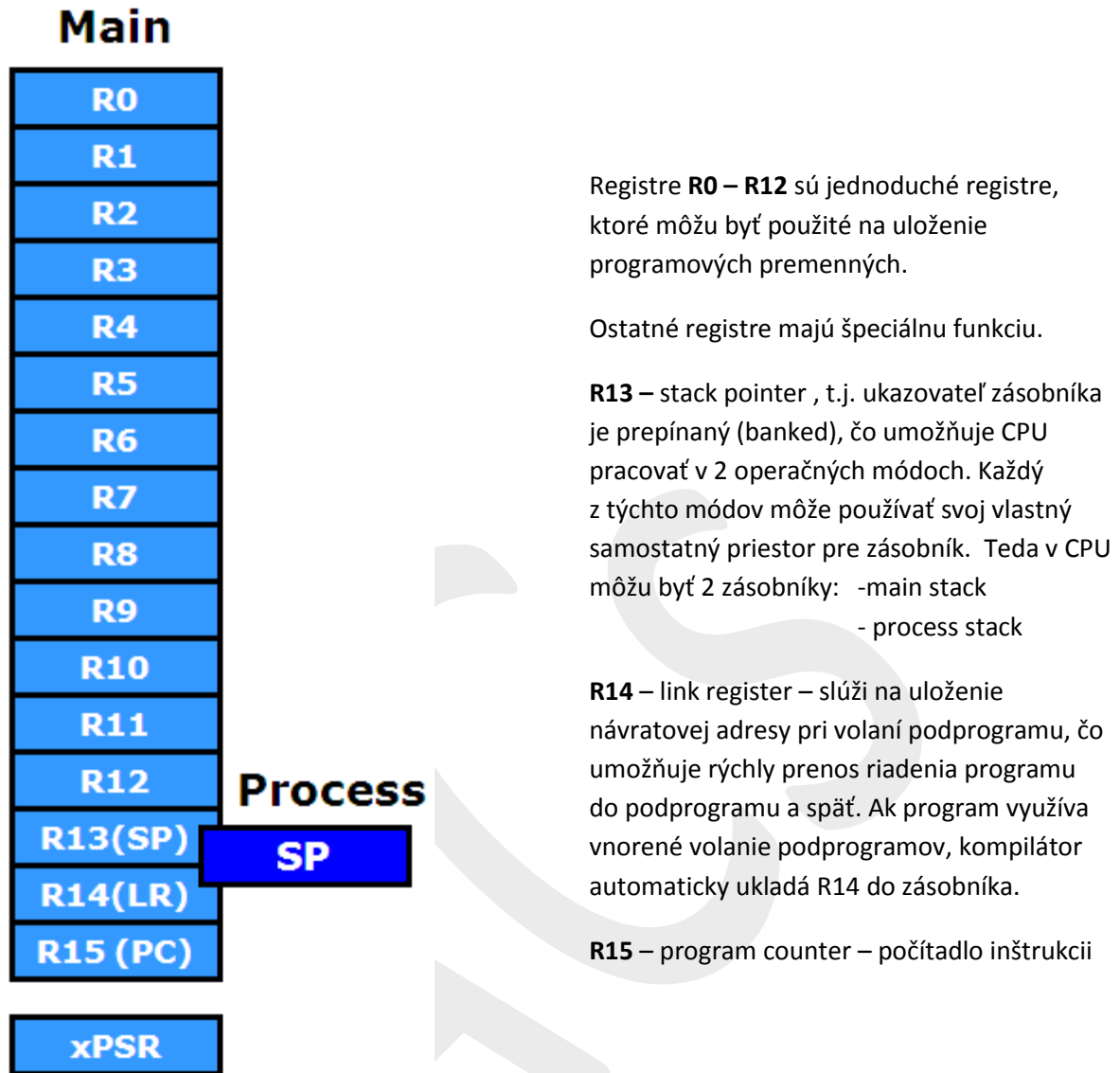


Obrázok 7: ARM Cortex-M3 má load and store architektúru.

2.3.1. Registre procesora

Teda celá činnosť programu je zameraná na prácu so súborom registrov CPU. Tento súbor registrov obsahuje **šestnásť** 32-bitových registrov.

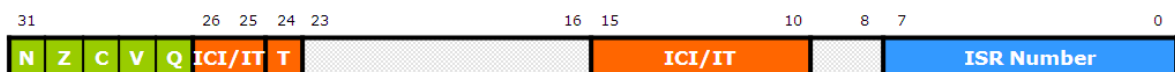
ARM – Cortex M



Obrázok 8: Súbor registrov CPU

Okrem súboru 16 registrov sa v CPU nachádza register nazývaný - **program status register** (register stavu programu). Nie je súčasťou hlavného registrového súboru a je prístupný len cez 2 určené inštrukcie. Obsahuje oblasti, ktoré majú vplyv na vykonávanie inštrukcií.

2.3.2. Program status register



Obrázok 9: Program status register xPSR (APSR, EPSR, IPSR)

Register xPSR možno tiež pristupovať pomocou troch špeciálnych aliasov, ktoré umožňujú prístup do čiastkových rozsahov bitov v rámci xPSR

ARM – Cortex M

Vrchných (najvyšších) 5 bitov sú príznaky inštrukcií a sú aliasované ako **Application Program Status Register** (APSR). Prvé 4 príznaky N, Z, C, V (Negative, Zero, Carry and Overflow) sú nastavované a čistené ako výsledok vykonania inštrukcií. Bit Q je nastavovaný pri niektorých matematických inštrukciách, ak premenná nadobudla jej minimálnu alebo maximálnu hodnotu. Niektoré inštrukcie sú vykonávané len keď podmienka inštrukčného kódu sa zhoduje s príznakmi v APSR. Ak sa nezhoduje, inštrukčný tok sa nenarušuje, ale inštrukcia prechádza plynule cez pipeline ako NOP (prázdna inštrukcia) a tým minimalizuje vyprázdňovanie pipeline.

Bity 26 – 8 sú aliasované ako **Execution Program Status Registers** (EPSR). Obsahuje 3 polia “If then”, “interrupt continuable instruction” a Thumb. Thumb je pozostatok z iných ARM procesorov a v CORTEX-M je vždy nastavený na 1. Inštrukcie Thumb-2 (IBA INŠTRUKCIE THUMB-2), ktoré sa používajú v CORTEX-M, umožňujú vytvárať malé if-then bloky. Ak je splnená podmienka if-then bloku (bity 10-15), nastavené IT pole povie CPU, koľko inštrukcií má vykonať (max. 4 inštrukcie – 2 bity, 25-26).

Thumb-2 inštrukcie väčšinou obsahujú jednocyklové inštrukcie, ale load and store inštrukcie sú viaccyklové a sú teda prerušiteľné. Do príslušných bitov sa zapisuje adresa, v ktorej bude pokračovať load and store inštrukcia po návrate z prerušenia a v ktorom stave vykonávania bola prerušená.

Interrupt Program Status Registers (IPSR) obsahuje identifikáciu aktuálneho vektoru prerušenia (interrupt service routine) – t.j.: informáciu o aktuálnej požiadavke na prerušenie, ktoré je vykonávané.

2.3.3. Operačné módy CPU

Procesor CORTEX-M bol navrhovaný na rýchle a ľahké využívanie Jadra mikrokontroléra a na podporu používania real-time operačného systému (RTOS).

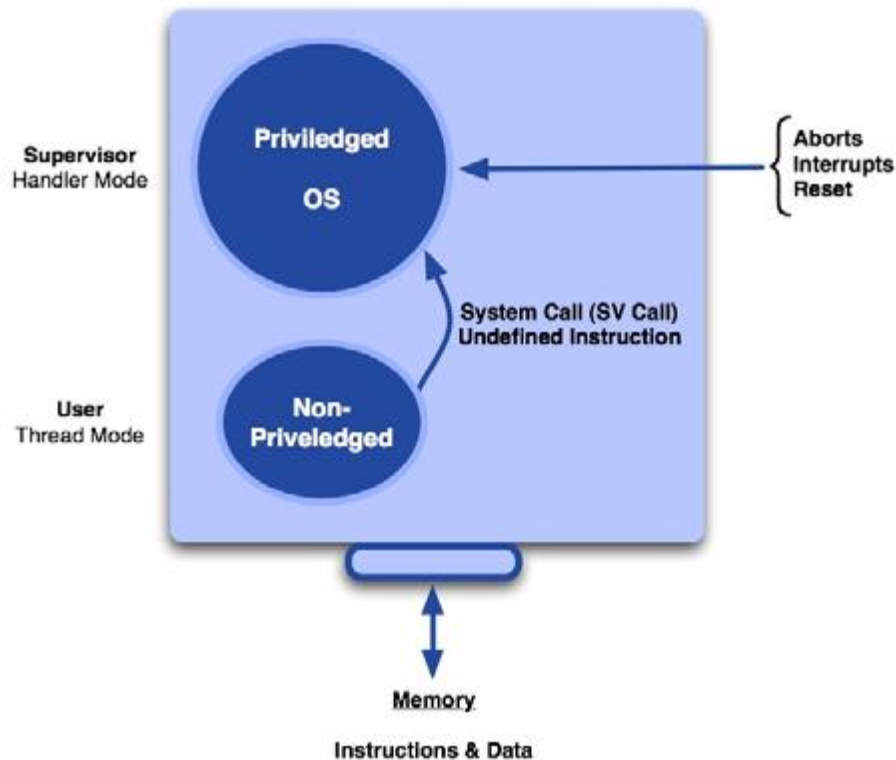
Procesor môže pracovať v 2 módoch:

- **Thread mode**
- **Handler mode**

CPU pracuje v **thread** móde, pokiaľ vykonáva kód bez prerušenia, ale keď nastane nejaká výnimka alebo prerušenie, prepne do **handler** módu

Navyše procesor môže pracovať v **privilegovanom** alebo **neprivilegovanom** režime. V privilegovanom režime má CPU prístup k celému setu inštrukcií a pracuje s main stackom. V neprivilegovanom režime sú niektoré inštrukcie zakázané (napr. inštrukcie prístupujúce k xPSR a jeho aliasom), pracuje s process stackom a navyše prístup k niektorým registrov procesora je zablokovaný.

ARM – Cortex M



Obrázok 10: Módy CPU

Použitie zásobníka môže byť teda nakonfigurované podľa špecifikácie použitého OS alebo programu. Main stack R13 môže byť používaný aj thread (aplikačné programy) aj handler (prerušená, výnimky) módom, ale thread mód môže byť nakonfigurovaný aj na používanie process stack (R13 –banked).

		Operations (privilege out of reset)	Stacks (Main out of reset)
Modes (Thread 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

Obrázok 11: Možnosti použitia módou CPU

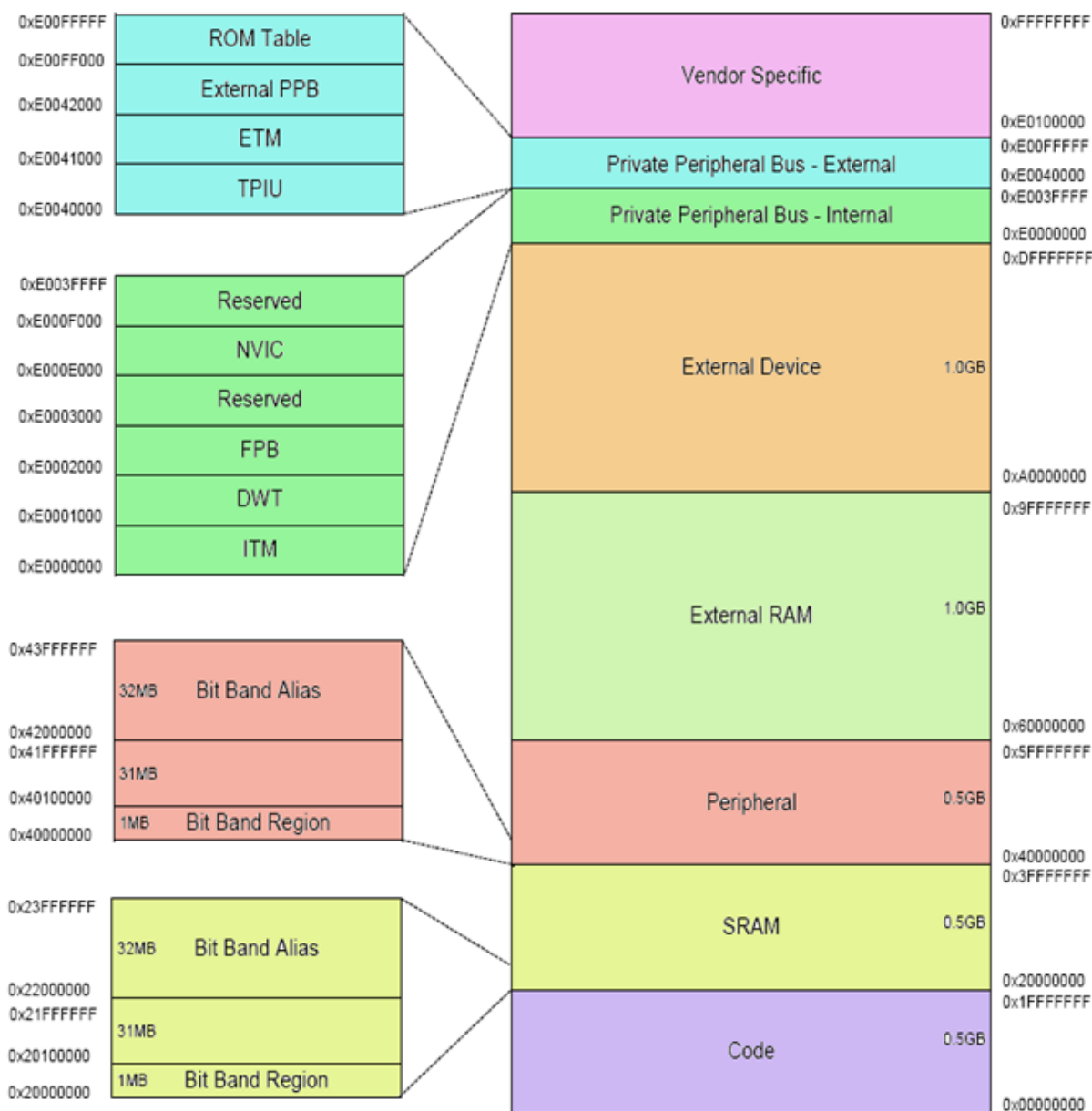
ARM – Cortex M

2.4. Pamäťový model

Cortex-M3 procesor je štandardizované mikropočítačové jadro a ako také má dobre definovaný pamäťový model. Má 32 bitovú adresnú zbernicu. Tým je daná veľkosť pamäte, ktorú môže procesor používať. (1 adresa adresuje 1 Byte)

32 bitov => 2^{32} adres => $2^{32}B = 2^2 \cdot 2^{10} \cdot 2^{10} \cdot 2^{10}B = 2^2 \cdot 2^{10} \cdot 2^{10}KB = 2^2 \cdot 2^{10}MB = 2^2GB = 4GB$

Obsahuje 4GB adresovateľného pamäťového priestoru.



Obrázok 12: Pamäťový model

ARM – Cortex M

4GB pamäťového priestoru má určené oblasti pre kód, SRAM, periférie, externú pamäť a periférie a pre Cortex systémové registre. Tento pamäťový model je spoločný pre všetky zariadenie s jadrom Cortex.

CODE region – 0,5GB – oblasť vykonateľného programového kódu. Môžu tu byť vkladane aj data – napr. konštanty. V tejto časti pamäte modelu býva interná FLASH pamäť.

SRAM region – 0,5GB – oblasť pre spracovávané dáta.

Peripheral region – 0,5 GB – všetky užívateľské periférie poskytnuté výrobcom mikropočítačového systému sú zaradené do tejto oblasti.

SRAM aj Peripheral region obsahujú 1MB bitovo adresovateľnej pamäte používanej techniku, ktorú nazývame bit-banding.

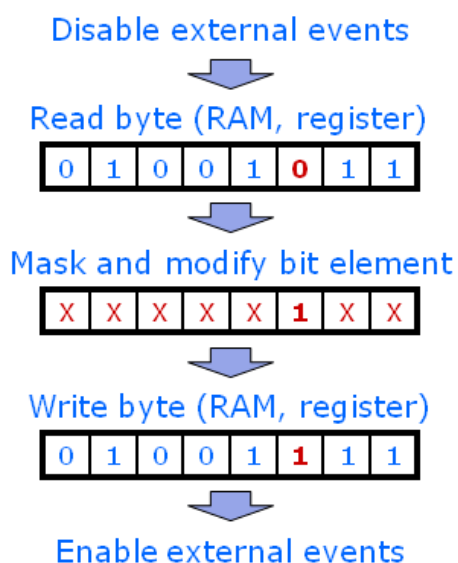
External RAM region – 1GB – externú RAM pamäť – oblasť, ktorá je určená pre dáta.

Vykonávaný programový kód môže byť uložený v CODE, SRAM aj v External RAM, ale odporúča sa ho ukladať v oblasti CODE, pretože procesor má oddelené zbernice, ktoré umožňujú vyberať inštrukcie a pristupovať k dátam súčasne. Teda program vykonávaný zo SRAM alebo external RAM regionu beží pomalšie, lebo inštrukcie sa prenášajú inou zbernicou.

Private Peripheral Bus – obsahuje NVIC, systémový časovač a systémový riadiaci blok.

2.4.1. Bit Banding

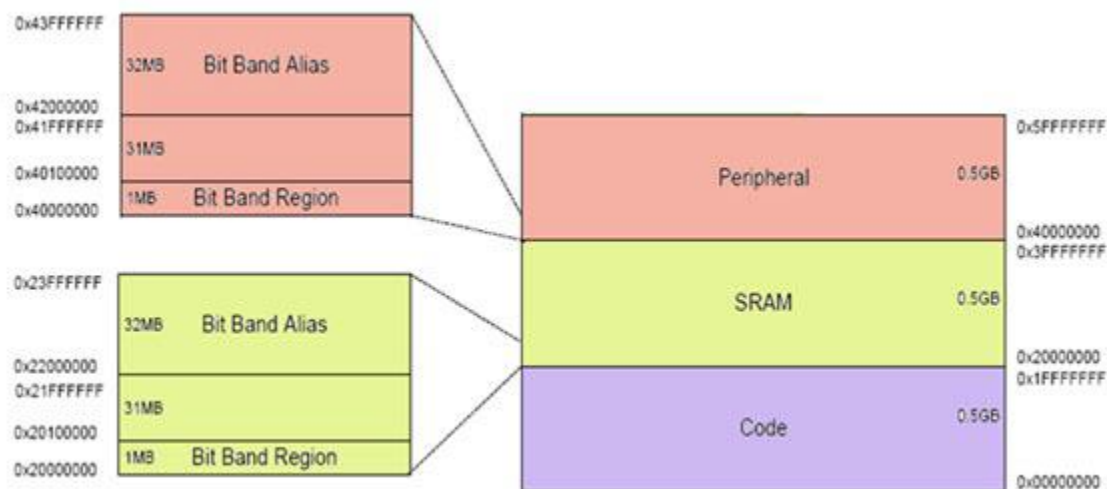
V prípade, že potrebujeme zmeniť len 1 bit v údají, nasleduje proces načítania, maskovania a modifikácie bitu a následný zápis upraveného údaju.



Tento proces nastavenia alebo nulovania 1 bitu je zdĺhavý, a trvá viac cyklov. Pre urýchlenie manipulácie s bitmi bola použitá technológia Bit Banding.

ARM – Cortex M

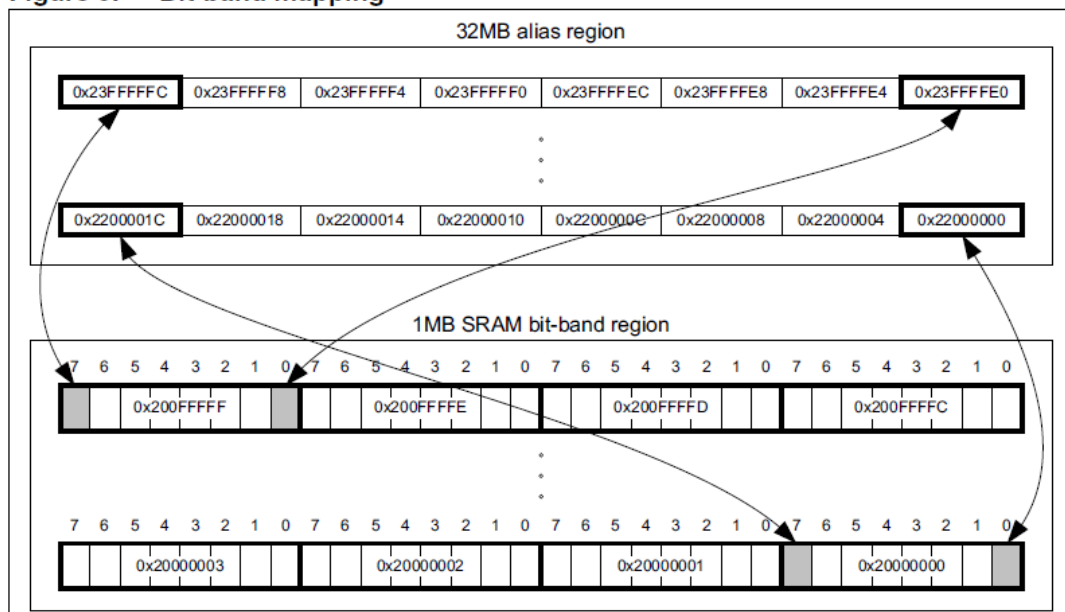
Bit Banding umožňuje v SRAM a v Peripheral mapovať každé slovo z Bit Band Alias na jednotlivé bity z Bit Band Region.



Obrázok 13: Bit Banding

Bit Band Region obsadzuje 1MB (t.j. $1\text{MB} \times 8\text{b} = 8\text{Mb}$, pre každý bit je určený 32 bitov (4B) v Alias t.j. $8\text{Mb} \times 4\text{B} = 32\text{MB}$) a teda Bit Band Alias obsadzuje 32 MB.

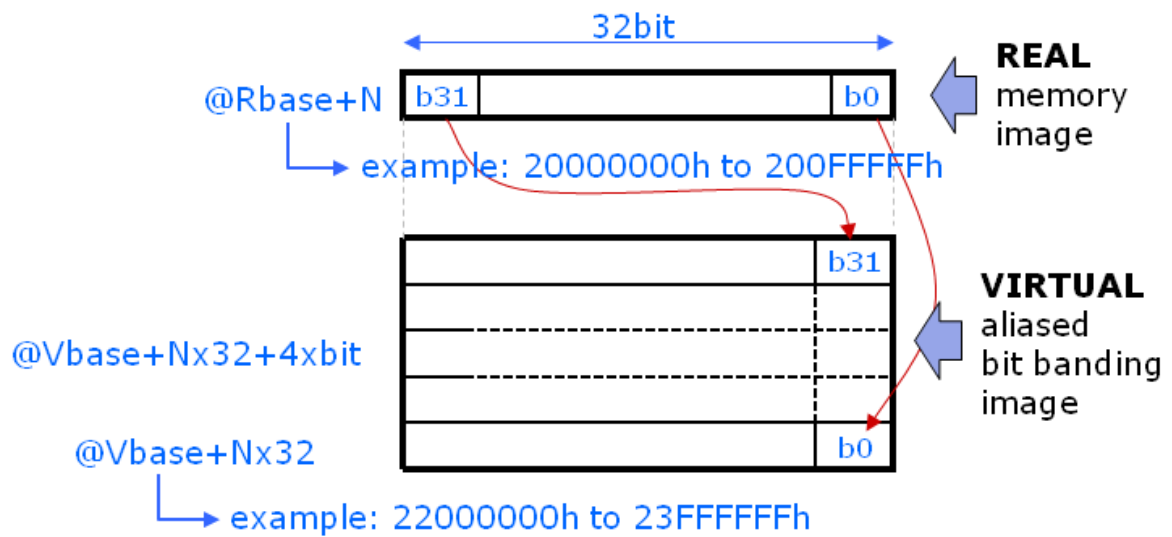
Figure 9. Bit-band mapping



Obrázok 14: Bit Band mapovanie

Z obrázku je vidieť, že 1 Byte z Bit Band Alias je mapovaný na príslušný bit v Bit Band Region. Do budúcnosti je počítané, že sa budú mapovať až 4B na 1 bit a teda je vytvorená rezerva (viď adresy).

ARM – Cortex M



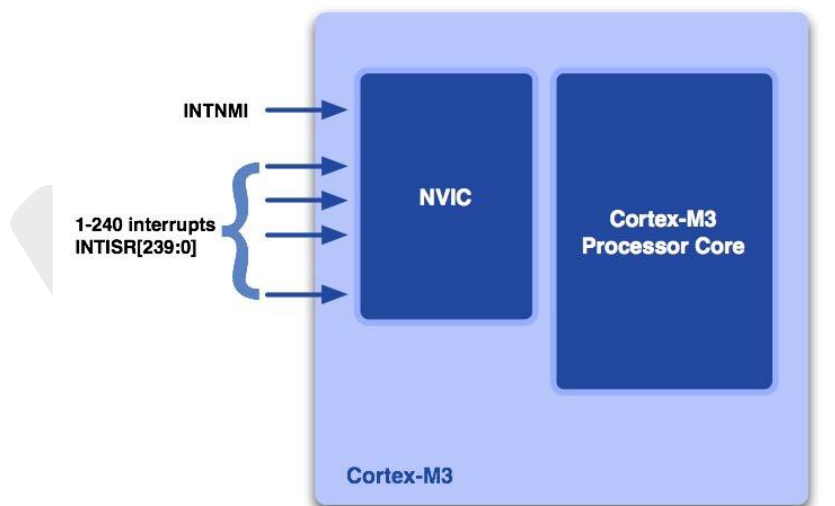
Obrázok 15: Adresa v Bit Band Alias = $@Vbase + N \times 32$ (8b x 4B) + poradie bitu v registri x 4B

ARM – Cortex M

2.5. Prerušovací systém

Požiadavka na prerušenie je požiadavka externého zariadenia na procesor, aby procesor prerušil vykonávanie aktuálneho programu – postupnosť inštrukcií, vykonal podprogram určený na obsluhu prerušenia (t.j. spracoval požiadavku na prerušenie) a následne pokračoval v prerušenom programe. Súbor takýchto požiadaviek na prerušenie, ich spôsob spracovania a priorít nazývame prerušovací systém.

Prerušovací systém v CORTEX-M jadre je realizovaný štandardnou jednotkou nazývanou radič vnorených prerušení - **Nested Vector Interrupt Controller** –NVIC. To znamená, že mikropočítačové systémy na báze CORTEX-M jadra majú rovnakú štruktúru prerušenia bez ohľadu na výrobcu. Ako už názov napovedá, NVIC je navrhnutý na podporu vnorených prerušení a pre STM32 systém podporuje 16 úrovní priority. NVIC je plne programovateľný a svojou architektúrou spĺňa požiadavky na real-time použitie (t.j. použitie v aplikáciách s presne definovanou odozvou na vznik prerušenia).



NVIC obsahuje:

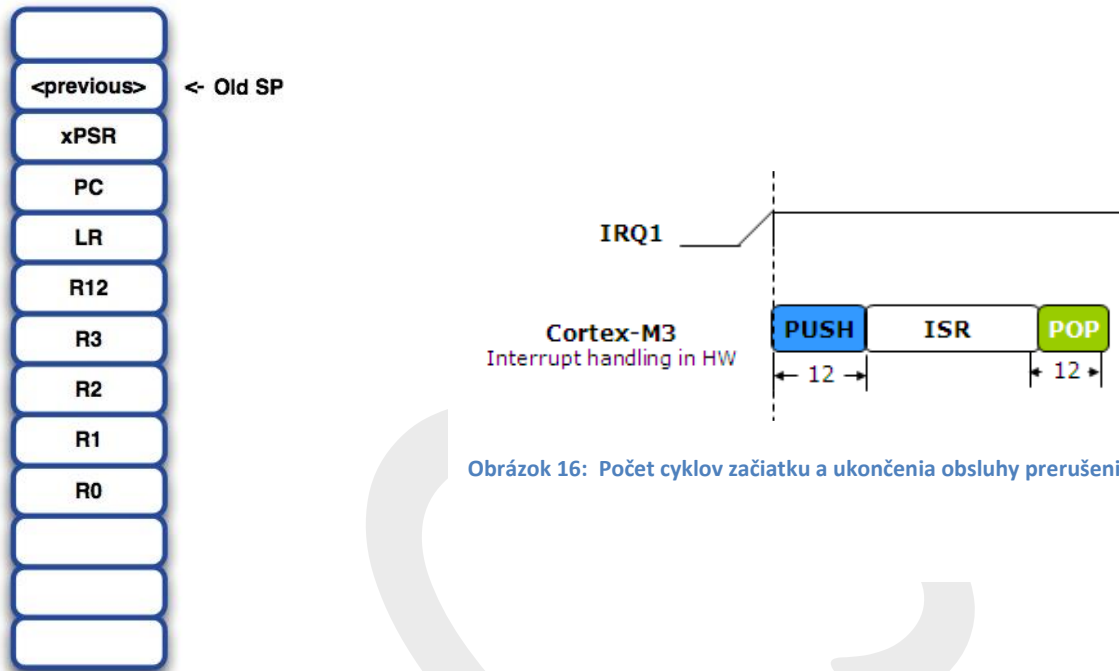
- 1 nemaskovateľné prerušenie (non-maskable interrupt – NMI, pin na puzdre procesora),
- 15 zdrojov prerušenia v CORTEX jadre, ktoré sa používajú na spracovanie výnimiek v rámci samotného jadra, na riadenie pri použití RTOS, atď,
- 1-240 externých prerušení, na ktoré môžu byť pripojené užívateľské periférie.

2.5.1. Začiatok a koniec obsluhy prerušenia

Keď príde požiadavka na prerušenie od periférie, NVIC naštartuje CORTEX-M CPU na proces obsluhy prerušenia. Z dôvodu, že CORTEX-M CPU prechádza do režimu prerušenia, je potrebné uchovať informácie o prerušenom programe. Obsah určených registrov sa uloží do zásobníka (PSP alebo MSP). Keď je obsah registrov uložený do zásobníka, vyberie sa cez inštrukčnú zbernicu adresa podprogramu na obsluhu prerušenia.

ARM – Cortex M

Čas od požiadavky na prerušenie po prvú inštrukciu obslužného podprogramu prerušenia je **12 cyklov**, počas ktorých procesor zrealizuje vyššie uvedené úkony.



Obrázok 16: Počet cyklov začiatku a ukončenia obsluhy prerušenia

Obrázok 17: Registre, ktoré sa pri obsluhu prerušenia ukladajú do zásobníka

2.5.2. Pokročilý mód spracovania prerušenia

Rovnako, ako je schopný rýchlo spracovať jedno prerušenie, NVIC je navrhnutý, aby efektívne spracovával viac prerušení v reálnom čase. Má metódy na spracovanie viacerých zdrojov prerušenia s minimálnym oneskorením medzi prerušeniami a na zabezpečenie, aby požiadavky s najvyššou prioritou boli spracované ako prvé.

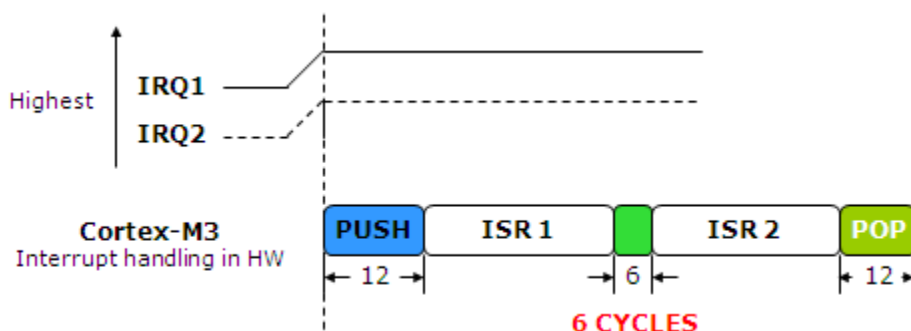
2.5.2.1. Prerušenie prerušenia (Interrupt Pre-emption)

Ak príde požiadavka na prerušenie s vyššou prioritou v čase obsluhy prerušenia s nižšou prioritou, obsluha prerušenia s nižšou prioritou sa zastaví. Vybrané registre sa uložia v 12 cykloch do zásobníka, načíta sa adresa obslužného podprogramu prerušenia s vyššou prioritou. Po ukončení obsluhy prerušenia s vyššou prioritou sa obnovia registre zo zásobníka a pokračuje obsluha prerušenia s nižšou prioritou.

ARM – Cortex M

2.5.2.2. Metóda koncového zret'azenia (Tail Chaining)

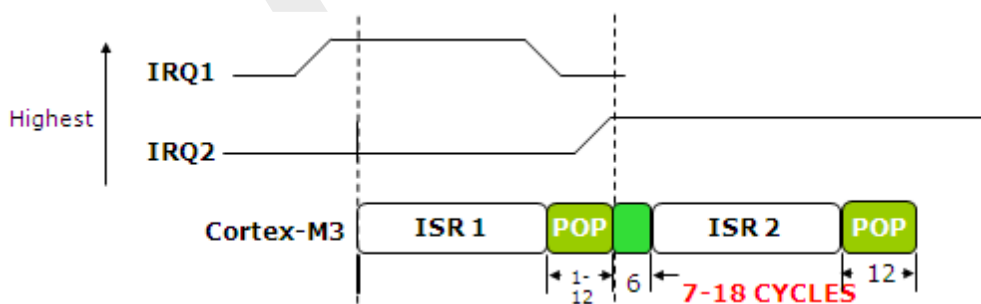
Ak beží obsluha prerušenia s vyššou prioritou a príde požiadavka na prerušenie s nižšou prioritou, použije sa metóda Tail Chaining, aby sa zabezpečilo minimálne oneskorenie medzi obsluhými podprogramami.



Obrázok 18: Tail Chaining

Ako vidieť z obrázka, počas 12 cyklov sa uložia registre do zásobníka, vyberie sa adresa obslužného podprogramu. Po ukončení obslužného podprogramu sa neobnovia registre zo zásobníka, ale načíta sa adresa obslužného podprogramu ďalšieho prerušenia podľa priority – 6 cyklov. Po ukončení obsluhy prerušenia sa obnovia registre zo zásobníka – 12 cyklov.

Ak počas obsluhy prerušenia príde požiadavka na prerušenie s nižšou prioritou, od obnovovania registrov bude upustené, ukazovateľ zásobníka sa obnoví na pôvodnú hodnotu, pridá sa 6 cyklov na vybratie adresy obslužného podprogramu prerušenia s nižšou prioritou t.j. oneskorenie je 7 – 18 cyklov pred obsluhou nového prerušenia. (viď. nasledujúci obrázok)

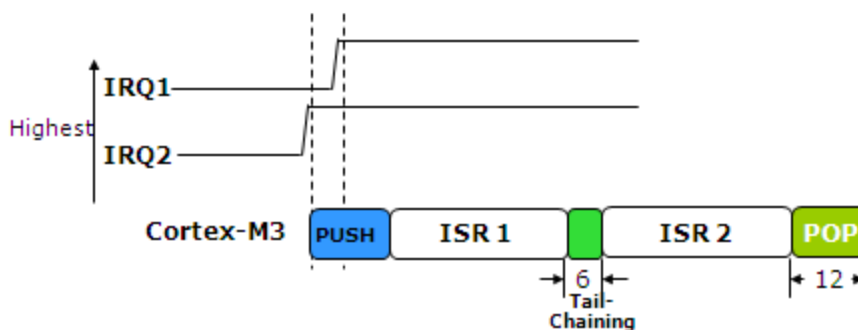


Obrázok 19: Ret'azenie obsluhy prerušení

2.5.2.3. Oneskorený príchod požiadavky s vyššou prioritou

V realite sa často stáva, že začne obsluha požiadavky na prerušenie s nižšou prioritou a príde následne požiadavka s vyššou prioritou. Pokiaľ sa tak stane v inicializačnom ukladaní registrov do zásobníka (PUSH), prepne sa na obsluhu prerušenia s vyššou prioritou, po jej ukončení nasleduje Tail Chaining.

ARM – Cortex M



Obrázok 20: Oneskorený príchod požiadavky na prerušenie s vyššou prioritou

2.5.3. Konfigurácia a použitie NVIC

Aby sme mohli použiť NVIC, musíme:

1. Nakonfigurovať vektorovú tabuľku pre zdroje prerušenia, ktoré chceme používať.
2. Nakonfigurovať NVIC registre na povolenie a nastavenie priorít prerušení.
3. Nakonfigurovať periférie a povoliť ich podporu prerušení.

2.5.3.1. Vektorová tabuľka výnimiek

V CORTEX-M sa vektorová tabuľka výnimiek (prerušení) nachádza v pamäti v dolnej časti pamäťového modelu, konkrétne začína na adrese 0x00000004. Prvé 4B sú určené pre inicializačnú hodnotu zásobníka - „stack pointer“.

Každý vektor prerušenia vo vektorovej tabuľke má 4B a obsahuje adresu začiatku obslužného podprogramu určeného na obsluhu príslušného prerušenia. Prvých 15 položiek tabuľky sú výnimky, ktoré môžu nastať v jadre procesora (resetovací vektor, NMI, poruchy a riadenie chýb, výnimky debuggera, prerušenie od SysTick Timer prerušenie, ...). Perifériám sú vyhradené prerušenia od 16 po max. počet určený výrobcom (označované ako Interrupt_0 .. Interrupt_240).

Tabuľku adries vektorov prerušenia je možné presunúť z pôvodnej preddefinovanej adresy (0x00000004) na ľubovoľnú inú adresu v rozsahu adresného priestoru procesora.

ARM – Cortex M

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

Obrázok 21: Tabuľka výnimiek – prerušení



Registre NVIC sú sprístupnené len ak CPU pracuje v privilegovanom režime.

Interné CORTEX výnimky sú konfigurované v System Control a System Priority registroch.

Vonkajšie prerušenia užívateľských periférií sú konfigurované cez IRQ registre.

Obrázok 22: Registre NVIC

Ako vidieť z tabuľky Reset, NMI a HardFault majú fixne danú prioritu, ostatné vnútorné výnimky majú nastaviteľnú prioritu (v System Priority) – každá výnimka má 8 bitov na nastavenie priority (keďže má 16 úrovní priority = využívajú sa len 4 bity).

ARM – Cortex M

2.6. Systémový časovač

CORTEX jadro obsahuje 24 bitové počítadlo počítajúce dozadu s možnosťou automatického obnovenia počiatkovej hodnoty (cyklické počítanie) a vyvolaním prerušenia pri ukončení cyklu počítania. (Ak na vstup počítadla impulzov privedieme vstup z oscilátora, hovoríme o časovači). Toto počítadlo poskytuje štandardný časovač pre všetky CORTEX-M a nazývame ho systémový časovač – **System Timer (SysTick)** s prideleným vektorom/číslom prerušenia **15**. Ako zdroj impulzov môže byť použitý interný hodinový signál CORTEX-M procesora alebo externé referenčné hodiny, ktoré sa ale môžu líšiť medzi výrobcami.

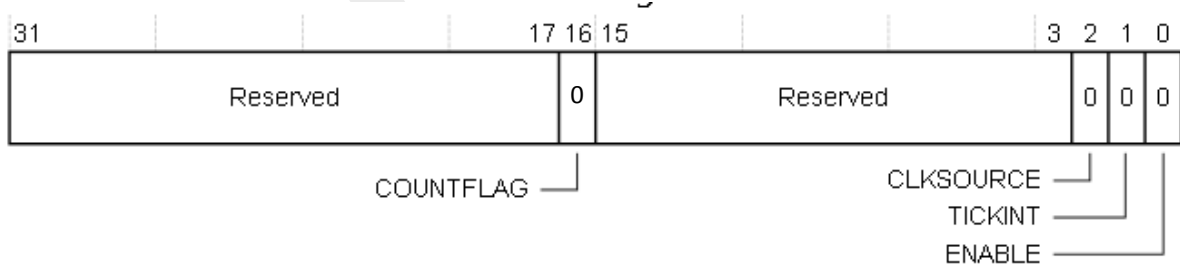
SysTick časovač je určený hlavne na určenie systémového ticku (taktu) pre RTOS, na generovanie periodických prerušení pre plánované úlohy, meranie času...



Obrázok 23: Registre systémového časovača

SysTick časovač má 3 registre:

- SysTick Control and Status register - riadiaci a stavový register, ktorý obsahuje napr. ENABLE bit na spustenie časovača a TICKINT bit na povolenie jeho prerušenia,
- SysTick Reload Value register - register obnovovacej hodnoty, je nastavovaná s novým cyklom počítania,
- SysTick Current Value register - aktuálna hodnota počítadla.



Obrázok 24: SysTick Control and Status register

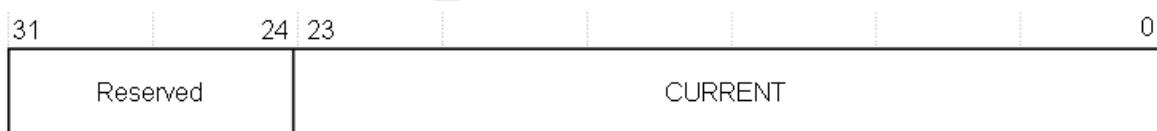
ARM – Cortex M

Bits	Name	Function
[31:17]	-	Reserved.
[16]	COUNTFLAG	Returns 1 if timer counted to 0 since last time this was read.
[15:3]	-	Reserved.
[2]	CLKSOURCE	Indicates the clock source: 0 = external clock 1 = processor clock.
[1]	TICKINT	Enables SysTick exception request: 0 = counting down to zero does not assert the SysTick exception request 1 = counting down to zero asserts the SysTick exception request. Software can use COUNTFLAG to determine if SysTick has ever counted to zero.
[0]	ENABLE	Enables the counter: 0 = counter disabled 1 = counter enabled.



Obrázok 25: SysTick Reload Value register

Bits	Name	Function
[31:24]	-	Reserved.
[23:0]	RELOAD	Value to load into the SYST_CVR register when the counter is enabled and when it reaches 0,



Obrázok 26: SysTick Current Value register

Bits	Name	Function
[31:24]	-	Reserved.
[23:0]	CURRENT	Reads return the current value of the SysTick counter. A write of any value clears the field to 0, and also clears the SYST_CSR COUNTFLAG bit to 0.

ARM – Cortex M

Časovač (počítadlo) po príchode impulzu zníži stav SysTick Current Value register o 1. Po dosiahnutí 0 sa obsah registra SysTick Current Value obnoví na hodnotu, ktorú obsahuje SysTick Reload Value register a od tejto hodnoty ďalej počíta. Pri prechode nulou (konkrétne pri prechode z 1 do 0) sa do príznaku COUNTFLAG zapíše 1 a vyšle požiadavku na výnimku (prerušenie), ak je v bite TICKINT povolené posielať takéto požiadavky.

Zapísaním 0 do SysTick Current Value registra sa vynuluje aj príznak COUNTFLAG.

Príklad obslužného podprogramu pre obsluhu prerušenia od systémového časovača:

```
Void SysTick_Handler (void)
{
    ...
}
```

2.7. Zbernice

CORTEX-M3 procesor má Harvardskú architektúru s oddelenou kódovou a dátovou zbernicou, ktoré sa nazývajú **Icode bus** a **Dcode bus**. Obe z týchto zberníc majú prístup ku kódu (preloženému – skompilovanému programu) a dátam v rozsahu 0x00000000 – 0x1FFFFFFF (Code region).

Navyše má **systémovú zbernicu** na prístup do pamäťového priestoru v rozsahu 0x20000000-0xDFFFFFFF and 0xE0100000-0xFFFFFFFF (Region SRAM, Peripheral, External RAM, External Device, Vendor Specific).

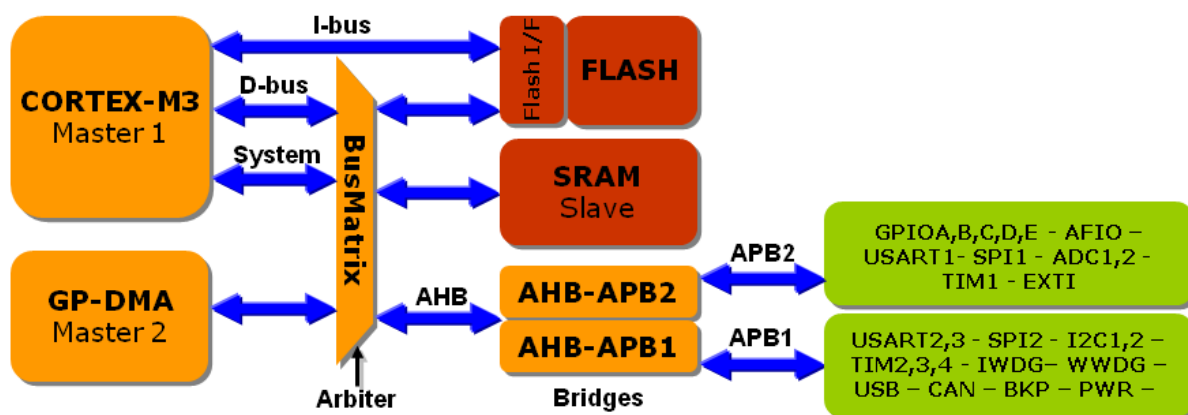
Ladiaci systém v čipe mikroprocesora má zbernicu nazývanú **Private Peripheral Bus**.

CORTEX-M3 CPU je prepojený s kódovou pamäťou (Code region) pomocou **Icode bus**. Systémová a dátová zbernice sú pripojené k externému mikropočítačovému systému pomocou sady vysokorýchlostných zberníc usporiadaného ako zbernica Matrix (**BusMatrix**). To umožňuje vytvoriť paralelné cesty medzi CORTEX zbernicami a ďalšími externými zbernicami - **Advanced High Speed Busses (AHB)**.

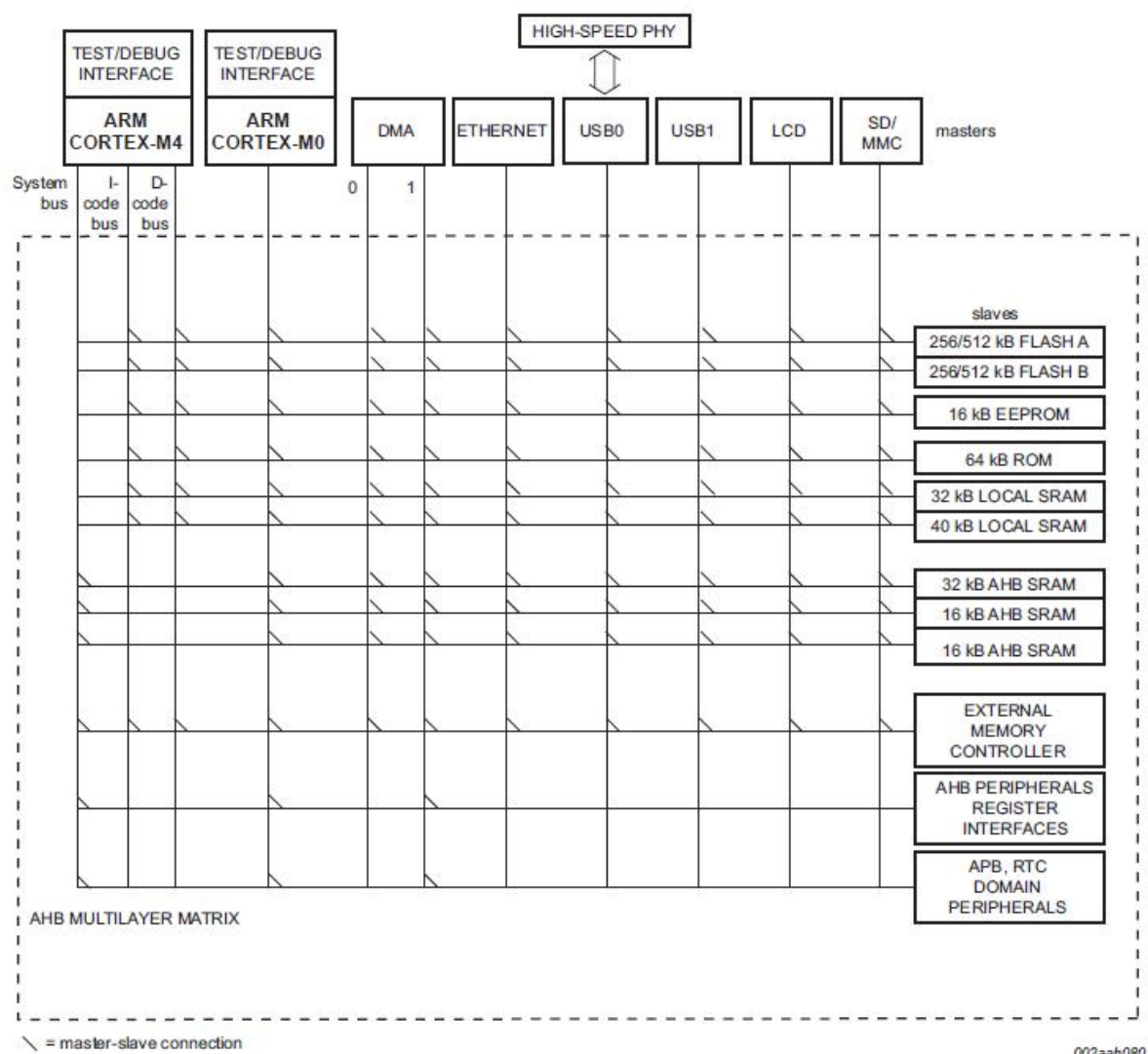
Periférie sú pripojené prostredníctvom 2 periférnych zberníc – **Advanced Peripheral Busses (APB)**. Každá z APB zberníc je premostená na AHB bus matrix (Bridge – s funkciou prispôsobenia rýchlosti jednotlivých zberníc).

AHB zbernica Matrix je taktovaná rovnakou rýchlosťou ako CORTEX jadro, ale pre úsporu energie môže byť taktovaná aj na nižšiu rýchlosť.

ARM – Cortex M



Obrázok 27: Vnútorý zbernicový systém poskytuje prepojenie procesora a jednotlivých častí mikropočítačov. Systému



Obrázok 28: AHB viacvrstvové matrix master a slave prepojenie

Periférie mikropočítačových systémov sa líšia podľa výrobcov a programujú sa odlišne v závislosti od výrobcu. Napr. ethernet od Texas Instruments sa programuje inak ako ethernet od STM a pod.