$7\frac{1}{2}$ Architektura ARM

Materiały do wykładu

Marcin Peczarski

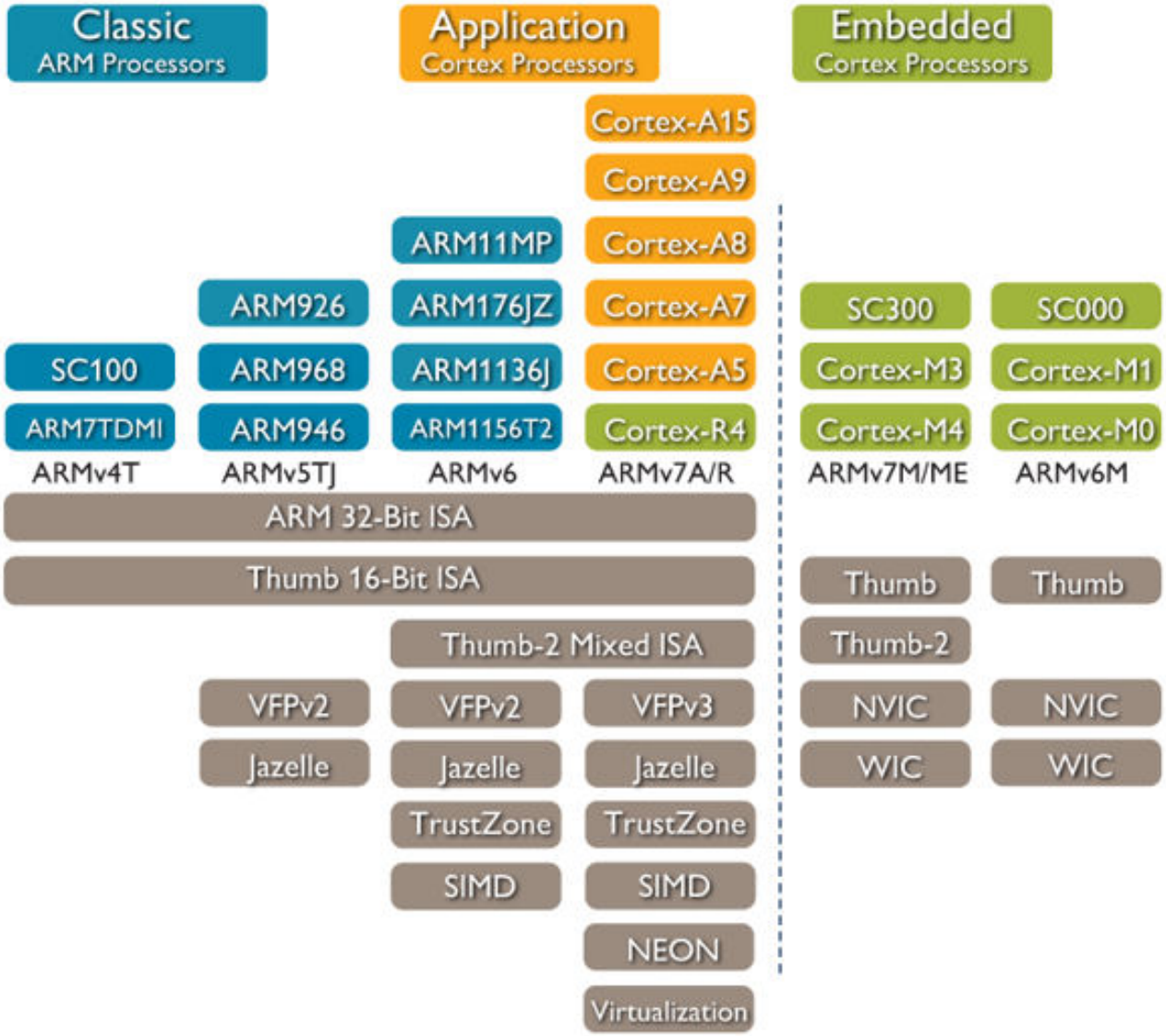
Instytut Informatyki
Uniwersytet Warszawski

19 maja 2015

ARM = Advanced RISC Machines

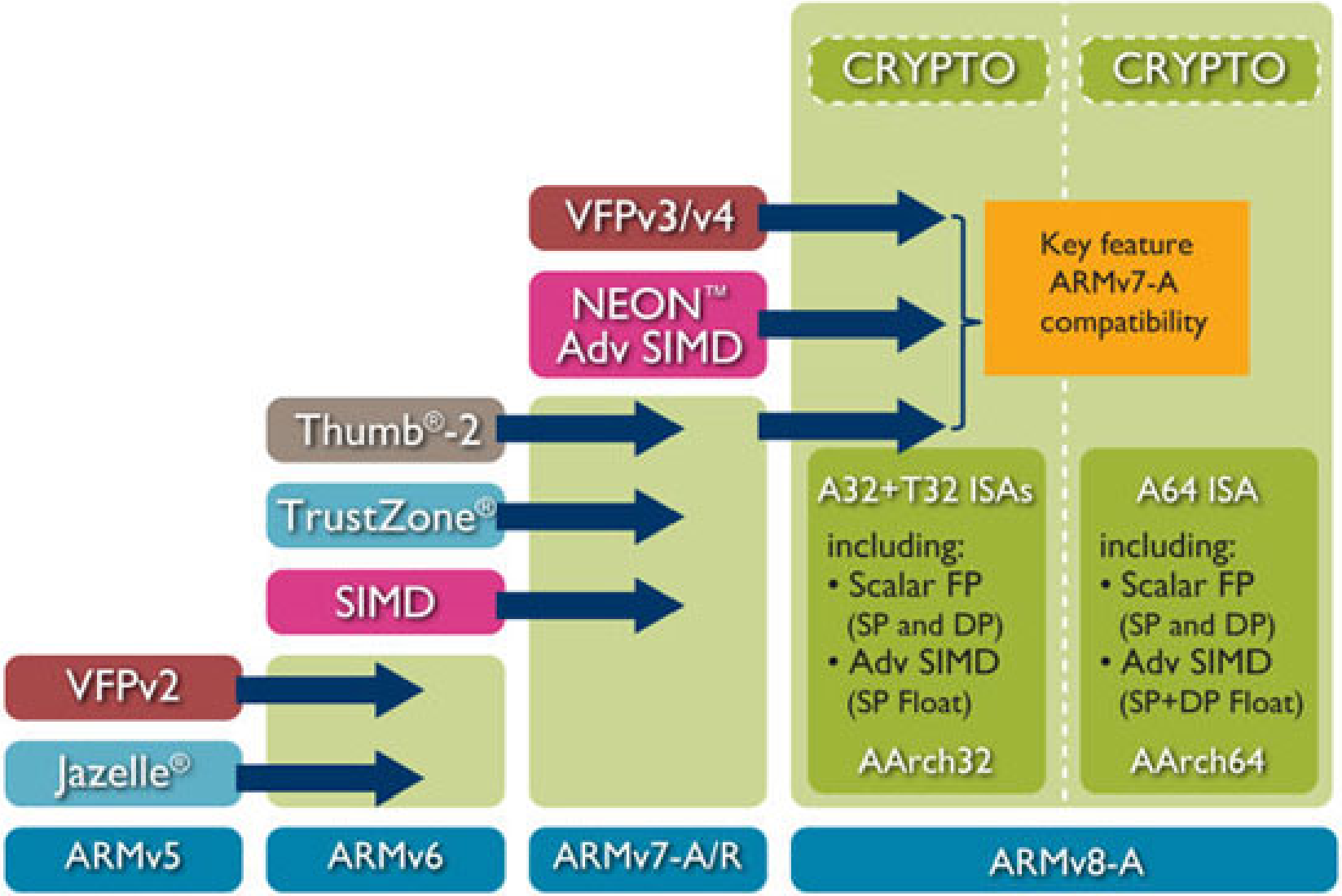
- Międzynarodowa firma, mająca główną siedzibę w Cambrdge w Wielkiej Brytanii
- Projektuje i sprzedaje licencje na rdzenie ARM
- Nie produkuje *krzemu*
- Większość urządzeń mobilnych, w tym głównie telefonów komórkowych, i systemów wbudowanych zawiera rdzenie ARM

Rozwój 32-bitowej architektury ARM



Źródło: `http://www.arm.com`

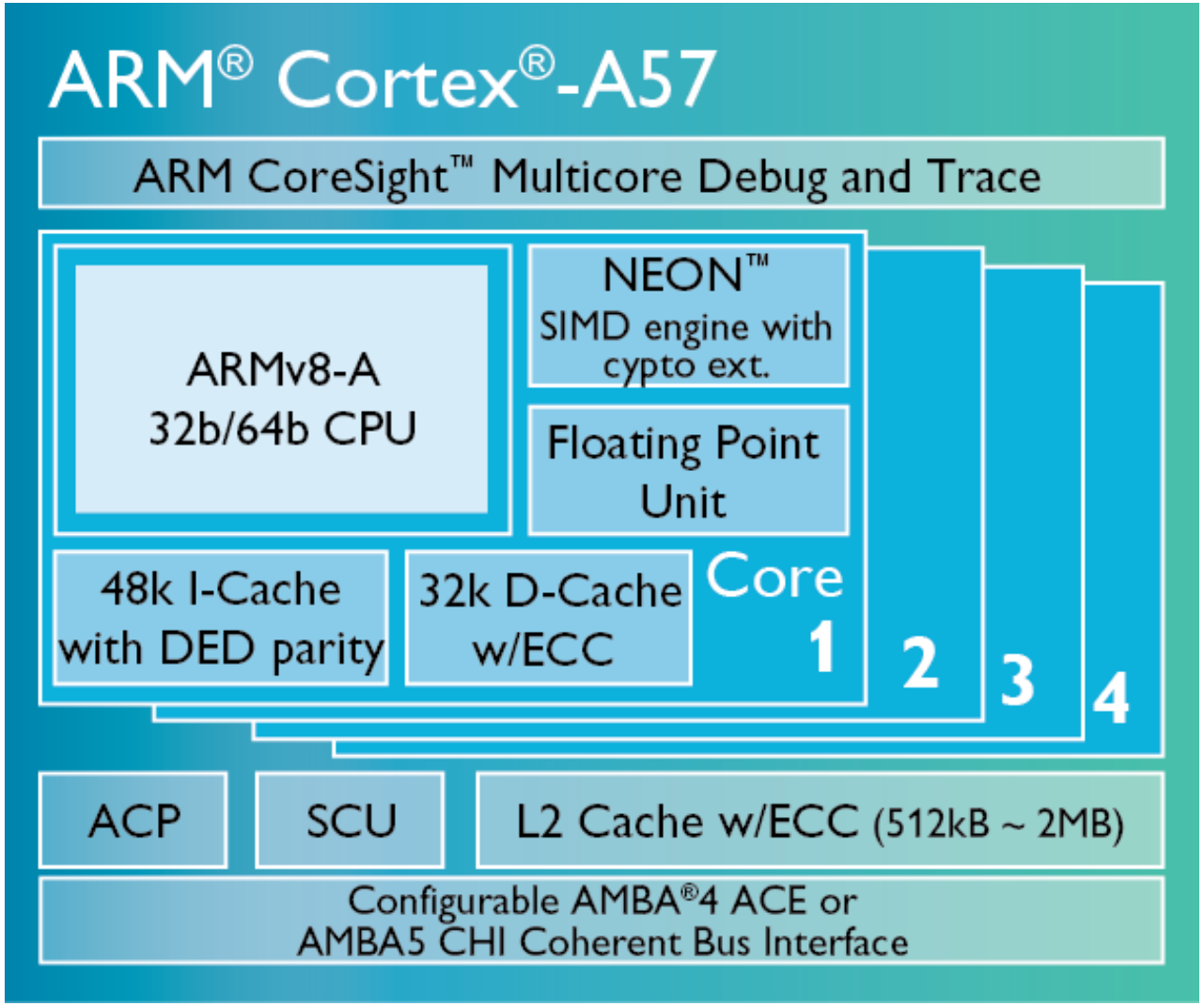
64-bitowa architektura ARM



Źródło: `http://www.arm.com`

ARM Cortex-A50

- Procesory Cortex-A57 i Cortex-A53



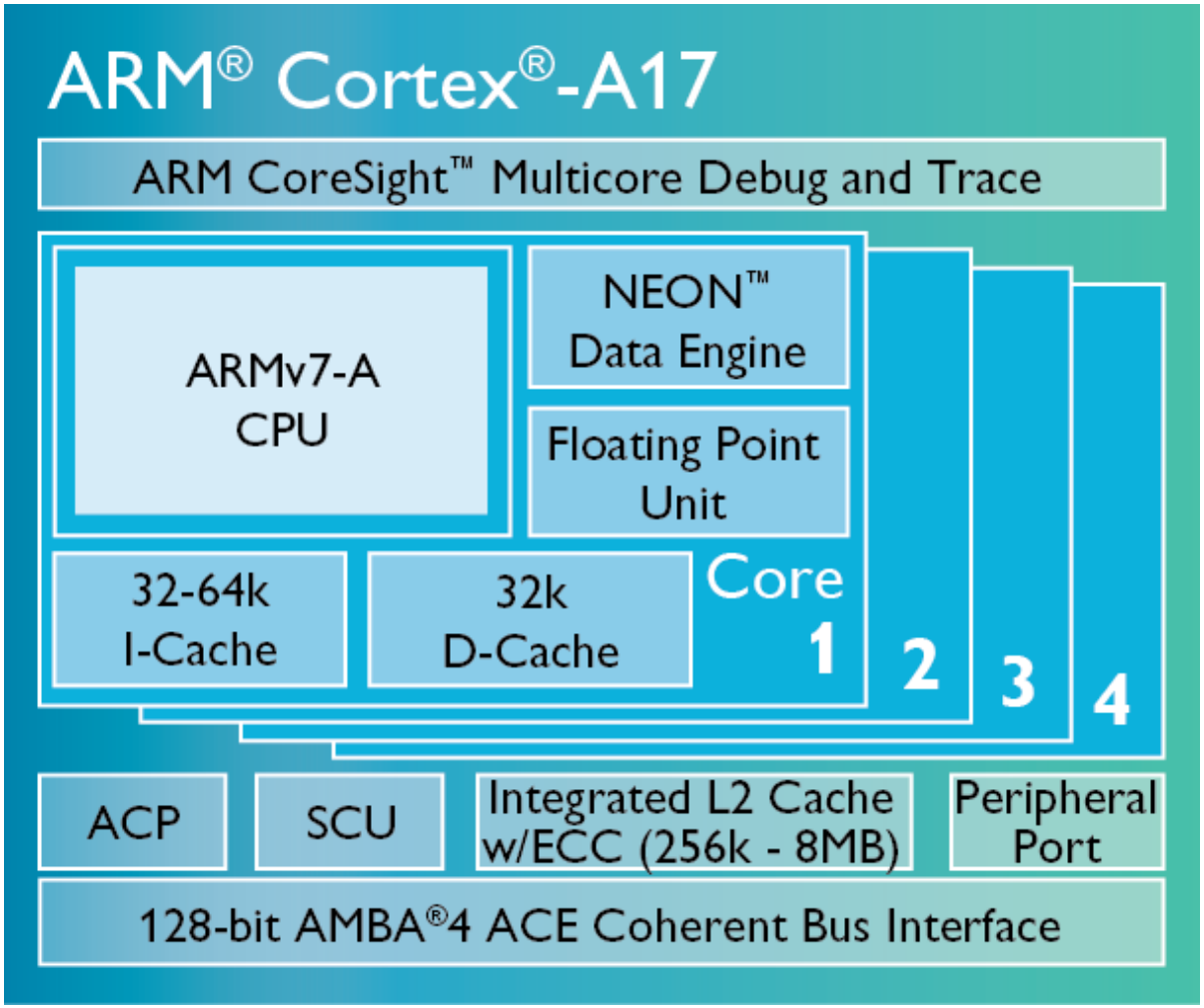
Źródło: `http://www.arm.com`

Różne fajne skróty

- ACP: Accelerator Coherency Port – układ zapewniający spójność pamięci (w tym przypadku podręcznej L2)
- SCU: Snoop Control Unit – układ zapewniający spójność pamięci (w tym przypadku podręcznej L2)
- ECC: Error Correction Code – kod korygujący błędy w pamięci półprzewodnikowej
  - SEC: Single Error Correct – możliwość skorygowania przekłamania jednego bitu
  - DED: Double Error Detect – możliwość wykrycia dwóch przekłamanych bitów
- AMBA 5: Advanced Microcontroller Bus Architecture – szyna wraz z zestawem protokołów do komunikacji między układami cyfrowymi, piąta wersja specyfikacji
  - CHI: Coherent Hub Interface

ARM Cortex-A

- Procesory Cortex-A17, Cortex-A15, Cortex-A12, Cortex-A9, Cortex-A8, Cortex-A7, Cortex-A5



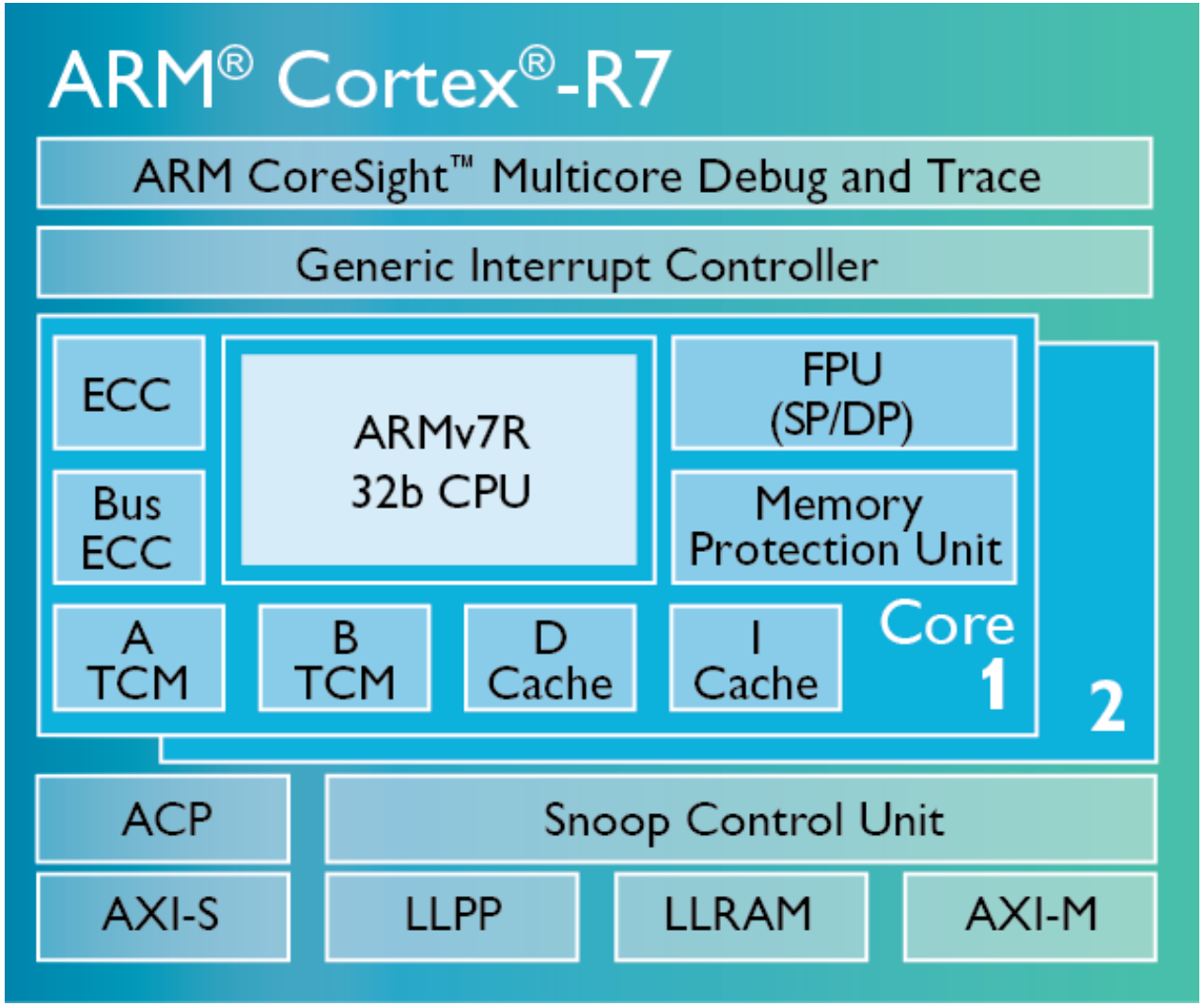
Źródło: `http://www.arm.com`

Różne fajne skróty, uwagi

- AMBA 4: czwarta wersja specyfikacji
  - ACE: AXI Coherency Extensions
  - AXI: Advanced eXtensible Interface
- Rdzenie Cortex-A przeznaczone są do stosowania w urządzeniach wymagających dużych mocy obliczeniowych i umiarkowanego zużycia energii
  - smartfony
  - tablety

ARM Cortex-R

- Procesory Cortex-R7, Cortex-R5, Cortex-R4



Źródło: `http://www.arm.com`

Różne fajne skróty

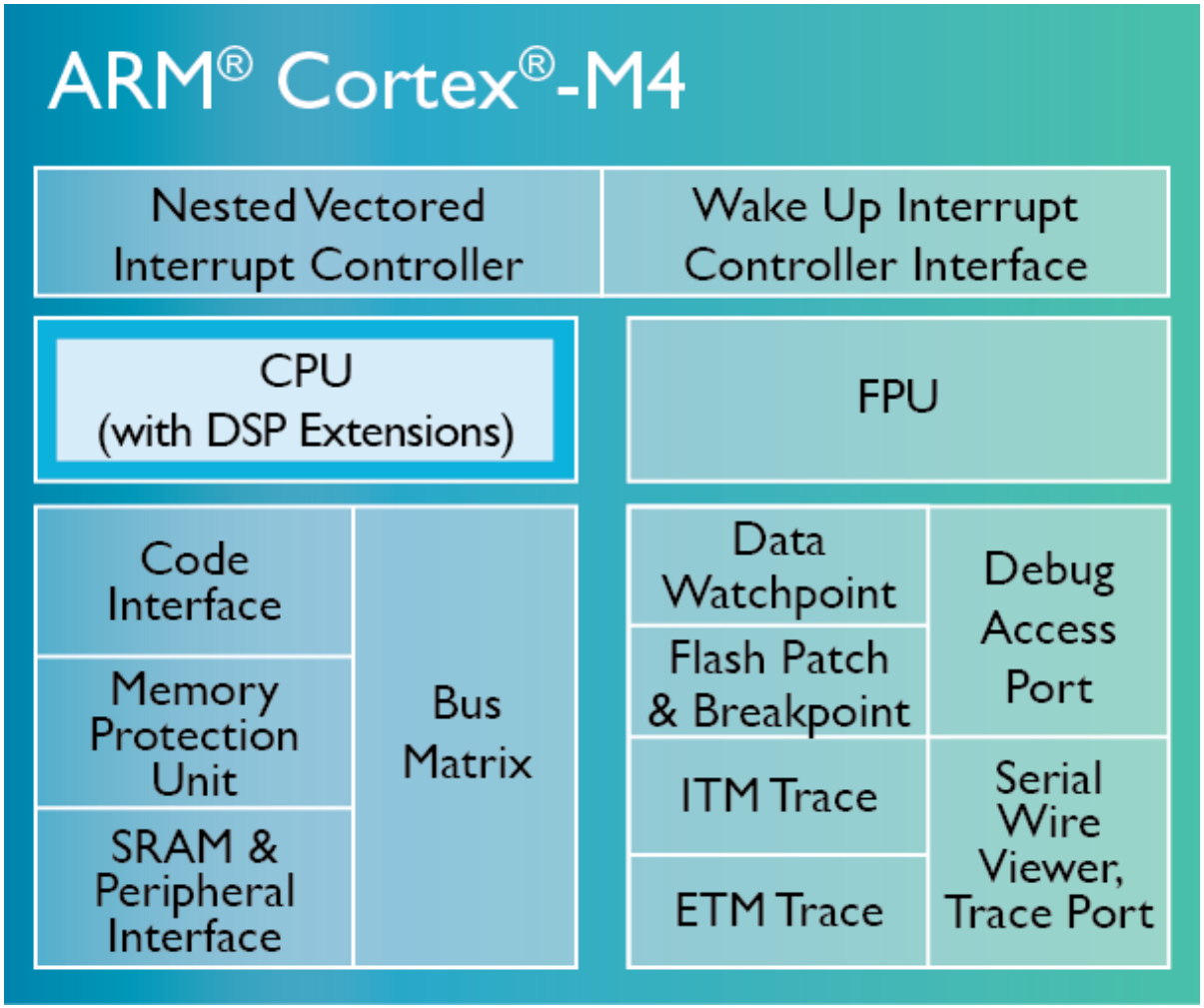
- FPU: Floating Point Unit – jednostka zmiennoprzecinkowa
  - SP: Single Precision – pojedyncza precyzja (32 bity)
  - DP: Dual Precision – podwójna precyzja (64 bity)
- MPU: Memory Protection Unit – układ umożliwiający separację procesów w pamięci operacyjnej
- TCM: Tightly Coupled Memory – pamięć umieszczona blisko procesora, przyspieszająca obsługę przerwań
- AMBA 3: trzecia wersja specyfikacji
  - AXI-S: Advanced eXtensible Interface Slave
  - AXI-M: Advanced eXtensible Interface Master
- LLPP: Low Latency Peripheral Port – port wejścia-wyjścia o krótkim czasie dostępu
- LLRAM: Low Latency RAM – pamięć RAM o krótkim czasie dostępu

Uwagi

- Rdzenie Cortex-R przeznaczone są do stosowania w systemach czasu rzeczywistego (ang. *real time*), gdzie wymagana jest szybka reakcja na zdarzenia, wysoka niezawodność, spora moc obliczeniowa
  - motoryzacja
  - lotnictwo
  - energetyka

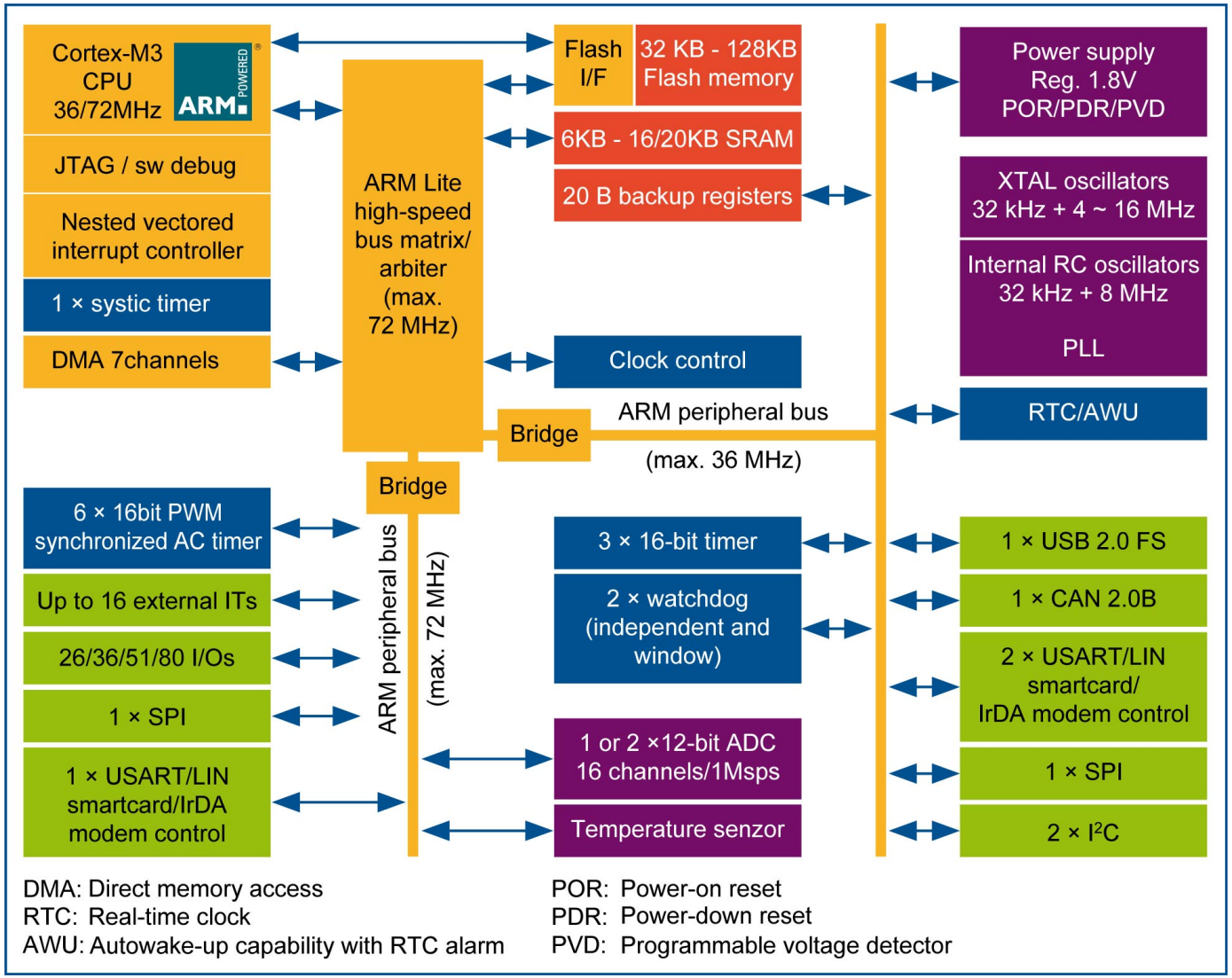
ARM Cortex-M

- Procesory Cortex-M7, Cortex-M4, Cortex-M3, Cortex-M1, Cortex-M0+, Cortex-M0



Źródło: `http://www.arm.com`

Przykład mikrokontrolera z rdzeniem Cortex-M3



Źródło: `http://www.st.com`

Różne fajne skróty

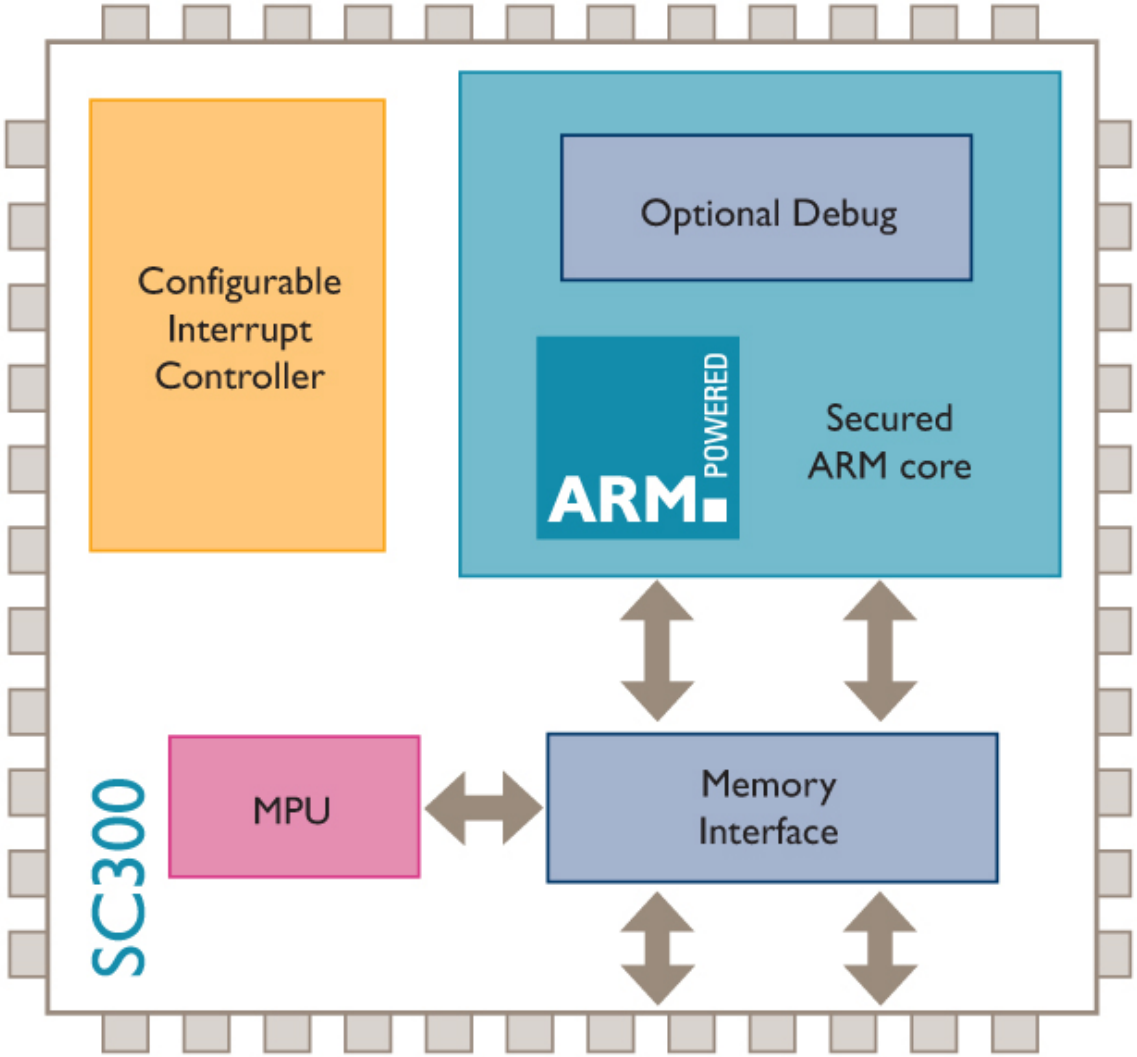
- DSP: Digital Signal Processing – sprzętowe wsparcie dla cyfrowego przetwarzania sygnałów, np. operacje z nasyceniem, dodawanie z akumulacją itp.
- ITM: Instrumentation Trace Macrocell – sprzętowe wsparcie odpluskwiania
- ETM: Embedded Trace Macrocell – sprzętowe wsparcie odpluskwiania
- AMBA 2: druga wersja specyfikacji
  - AHB: Advanced High-performance Bus
  - APB: Advanced Peripheral Bus

Uwagi

- Cortex-M7 przeznaczony do systemów wbudowanych wymagających bardzo dużej wydajności
- Cortex-M4 przeznaczony do systemów wbudowanych wymagających dużej wydajności
  - FPU SP i DSP tylko w tym rdzeniu
- Cortex M3 przeznaczony do szerokiego stosowania w systemach wbudowanych
- Cortex-M1 przeznaczony do implementacji w
  - FPGA (ang. *Field Programmable Gate Array*)
  - ASIC (ang. *Application Specific Integrated Circuit*)
  - ASSP (ang. *Application Specific Standard Product*)
- Cortex-M0+ przeznaczony do urządzeń o małym poborze energii
  - ograniczony zestaw instrukcji
  - uproszczona architektura
  - alternatywa dla architektur 8- i 16-bitowych
- Cortex-M0 to poprzednia (mniej udana) wersja Cortex-M0+

ARM SecureCore

- Procesory SC300, SC100 i SC000



Źródło: `http://www.arm.com`

Uwagi

- SC300 – rdzeń Cortex-M3
- SC100 – rdzeń ARM7TDMI
- SC000 – rdzeń Cortex-M0
- Główny obszar zastosowań to karty czipowe
  - SIM
  - płatnicze
  - do dekoderów TV
  - identyfikacyjne

Ogólne własności architektury

- Cienkokońcówkowa lub grubokońcówkowa, w niektórych rdzeniach jest możliwość wyboru
- Jednolita przestrzeń adresowa – architekura typu Princeton
- Osobne szyny do pamięci danych i programu – organizacja typu Hardward
- Rdzenie Cortex-M praktycznie nie wymagają pisania czegokolwiek w asemblerze

Rejestry w Cortex-M

- R0 do R12 – rejestry ogólnego przeznaczenia
- SP (R13, MSP, PSP) – wskaźnik stosu
- LR (R14) – adres powrotu
- PC (R15) – licznik programu
- PSR (APSR, IPSR, EPSR) – rejestr znaczników
- PRIMASK, FAULTMASK, BASEPRI – rejestry maskujące przerwania
- CONTROL – rejestr sterujący trybami pracy rdzenia

Zestaw instrukcji w trybie 32-bitowym

- Wzorowany na RISC, aby łatwo potokować

```
ldr     r3, [r1, #36]
adds    r2, r3, #1
```

- RISC-owe wywoływanie procedur

```
bl      etykieta
bx      lr
```

- Prawie każda instrukcja może być wykonywana warunkowo

```
movge   r7, r4
```

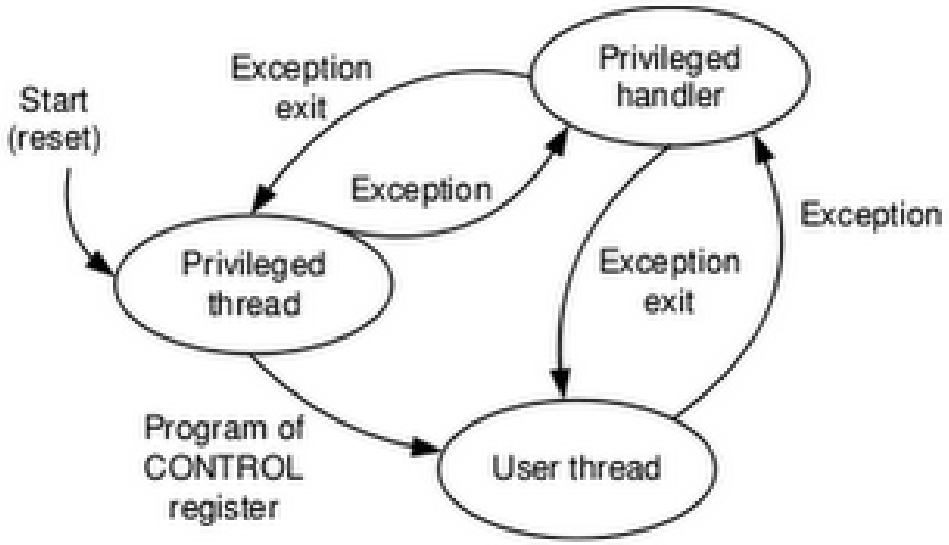
- Argument można przesunąć przed wykonaniem operacji

```
orr     r1, r1, r4, lsl #12
str     r2, [r4, r0, lsl #2]
```

- Występują instrukcje typowo CISC-owe

```
push    {r4, r5, lr}; prolog funkcji
pop     {r4, r5, pc}; epilog funkcji
```

Tryby pracy rdzenia Cortex-M



Źródło: `http://sigalrm.blogspot.com`

- Są dwa stosy:
  - `MSP` dla trybów uprzywilejowanych (ang. *privileged thread*, *privileged handler*),
  - `PSP` dla trybu użytkownika (ang. *user thread*).

ABI funkcji i przerwań w Cortex-M

- Pierwsze 4 argumenty funkcji są przekazywane w rejestrach `R0` do `R3`, pozostałe przez stos.
- Funkcja może dowolnie modyfikować rejestry `R0` do `R3` i `R12`.
- Funkcja musi przywrócić wartości rejestrów `R4` do `R11`, `LR` i `SP` sprzed wywołania.
- Przed wywołaniem procedury przerwania procesor odkłada na stos rejestry `R0` do `R3`, `R12`, adres powrotu, `PSR`, `LR`.
- Funkcja, która ma obsługiwać przerwanie, to zwykła funkcja języka C o sygnaturze

  ```
  void Handler(void);
  ```

- Nie są potrzebne żadne dodatkowe atrybuty!
  ~~`__attribute__ ((interrupt))`~~

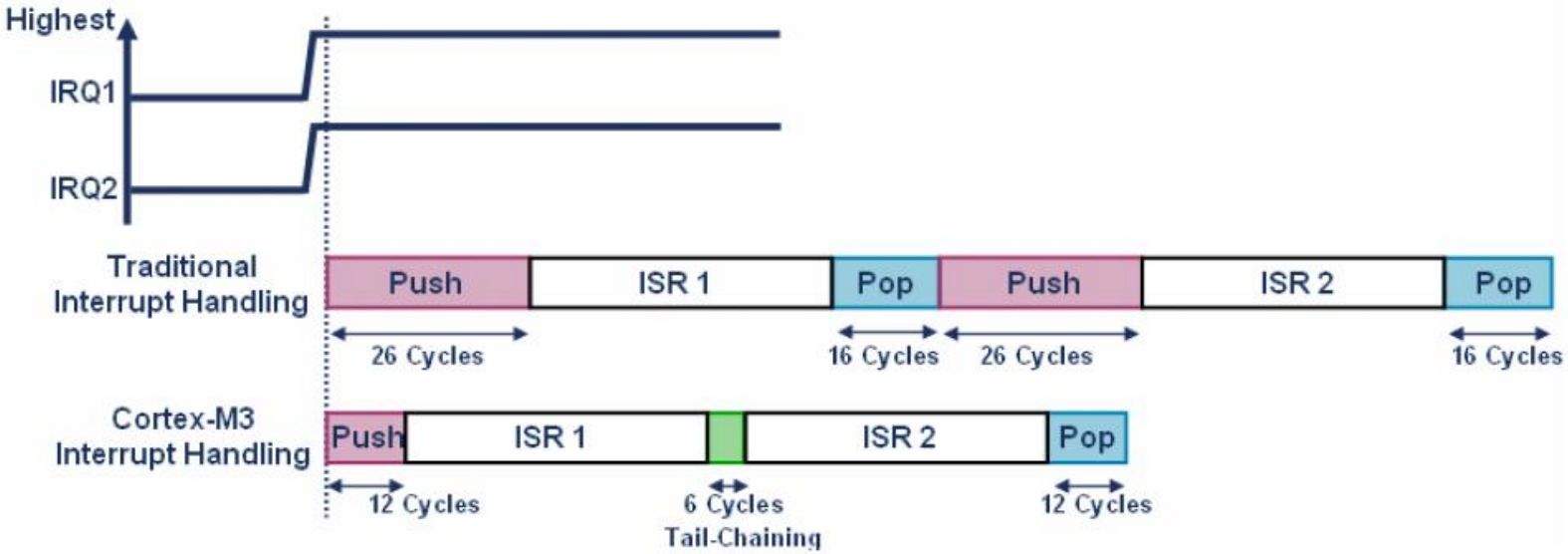
ABI przerwań, problem

- Skąd procesor wie, czy instrukcja `bx lr` jest powrotem ze zwykłej funkcji, czy z obsługi przerwania (trzeba coś zdjąć ze stosu)?
- Czy ktoś zauważył problem?

ABI przerwań, rozwiązanie problemu

- Przed wywołaniem funkcji obsługującej przerwanie procesor zapisuje do rejestru `LR` wartość:
  - `0xfffffff1` – powrót do obsługi innego przerwania (ang. *privileged handler*), odtwórz stan z `MSP`, po powrocie używaj `MSP`;
  - `0xfffffff9` – powrót do wątku uprzywilejowanego (ang. *privileged thread*), odtwórz stan z `MSP`, po powrocie używaj `MSP`;
  - `0xfffffffd` – powrót do wątku użytkownika (ang. *user thread*), odtwórz stan z `PSP`, po powrocie używaj `PSP`;
  - `0xffffffe1`, `0xffffffe9`, `0xffffffed` – jak wyżej, ale z odtwarzaniem stanu rejestrów zmiennoprzecinkowych.

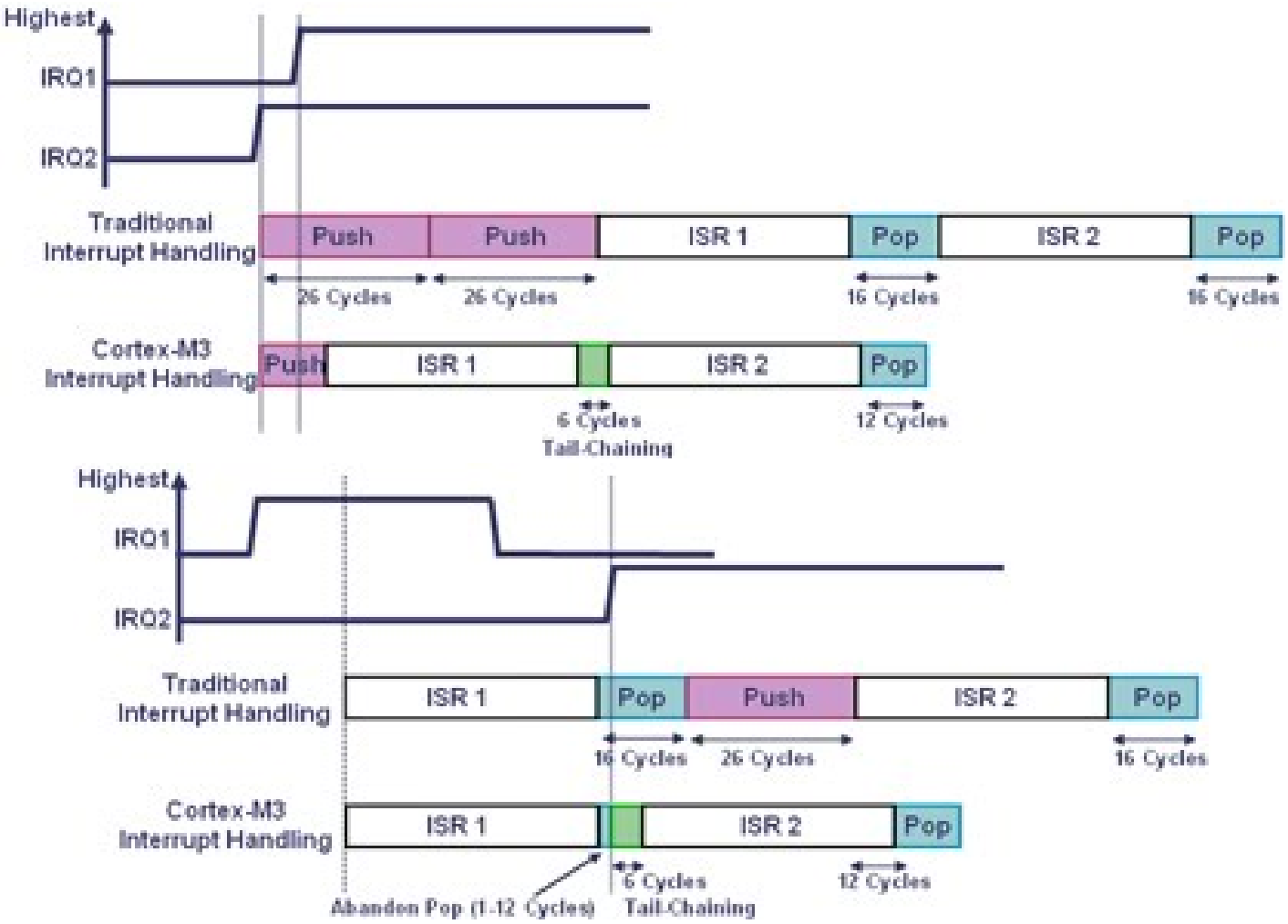
Łańcuchowa obsługa przerwań w Cortex-M



Źródło: `http://www.arm.com`

- Może dochodzić do zagłodzenia!

Łańcuchowa obsługa przerwań, dalsze przykłady



Źródło: `http://www.arm.com`

Priorytety przerwań w Cortex-M

- Przerwanie może mieć priorytet ujemny lub nieujemny.
- Czym mniejsza wartość, tym większy priorytet.
- Trzy przerwania mają ustalony, niekonfigurowalny priorytet:
  - Reset – priorytet -3,
  - NMI – priorytet -2,
  - HardFault – priorytet -1.
- Pozostałym przerwaniom można nadać priorytet, który jest liczbą nieujemną i składa się z dwóch pól:
  - priorytet wywłaszczania,
  - podpriorytet.
- Liczba bitów przeznaczonych na priorytet wywłaszczania jest w większości rdzeni konfigurowalna.
- Pozostałe bity są przeznaczone na podpriorytet.

Numery przerwań w Cortex-M

- Przerwania zgłaszane przez rdzeń mają ujemne numery

```
NonMaskableInt_IRQn    = -14,
...
MemoryManagement_IRQn = -12,
BusFault_IRQn          = -11,
UsageFault_IRQn        = -10,
...
SVCall_IRQn            = -5,
...
SysTick_IRQn           = -1,
```

- Przerwania zgłaszane przez peryferie mają nieujemne numery

```
WWDG_IRQn              = 0,
...
TIM2_IRQn              = 28,
...
```