

Procesory ARM. Część 1

Wykład 5

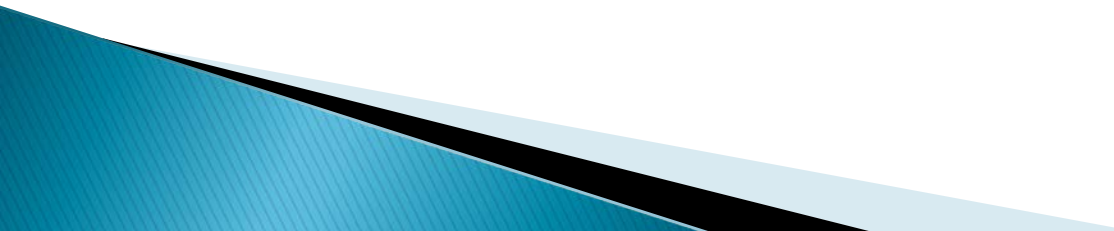
Projektowanie cyfrowych układów elektronicznych

Łukasz Kirchner

lukasz.kirchner@cs.put.poznan.pl

<http://www.cs.put.poznan.pl/lkirchner>

Plan wykładu

- ▶ Zapoznanie się z architekturą ARM
 - ▶ Porównanie cech rdzeni ARM7, ARM9, Cortex
 - ▶ Peryferia występujące w procesorach z rodziny ARM
- 

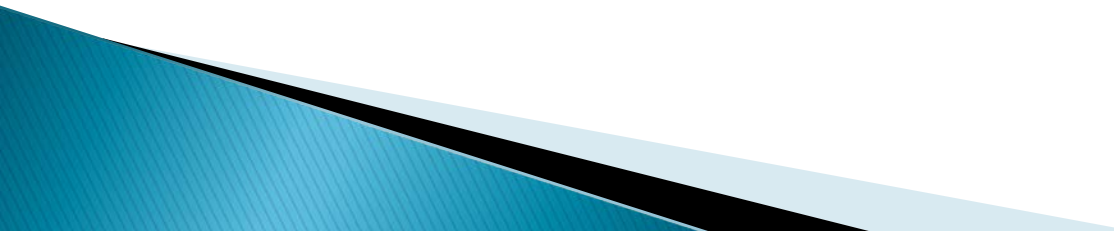
Bibliografia

- ▶ www.arm.com
- ▶ www.atmel.com – ARM7, ARM9
- ▶ www.luminarymicro.com – Cortex–M3
- ▶ www.ti.com
- ▶ www.st.com – STM32
- ▶ www.nxp.com – Cortex
- ▶ <http://www.arm.com/files/pdf/IntroToCortex-M3.pdf>
- ▶ http://www.arm.com/files/downloads/ARMv8_Architecture.pdf – ARMv8

Historia ARM

- ▶ 1983:(Acorn Computers; Roger Wilson, Steve Furber; MOS 6502)
- ▶ 1985: ARM1 (wersja testowa)
- ▶ 1987: ARM2 (wersja produkcyjna): dane 32bit, 26bit przestrzeń adresowa, 16x32bit Rejestry, 30tyś tranzystorów – brak mikro kodu, brak cache, szybszy od 80286
- ▶ 1989: ARM3
- ▶ Apple + Acorn -> 1990: Advanced RISC Machines (ARM Ltd.)
- ▶ 1991: ARM6 (PDA– Apple Newton – ARM610)
- ▶ 1993: ARM7
- ▶ 1995: ARM9
- ▶ 1998: ARM10

Architektura ARM

- ▶ ARM: Advanced RISC Machine, pierwotnie Acorn RISC Machine
 - ▶ Architektura 32 bitowa
 - ▶ Niski pobór mocy (stosowanie w systemach wbudowanych)
 - ▶ Obecnie stosowany prawie we wszystkim (HDD, telefony komórkowe, routery, kalkulatory, zabawki)
- 

Architektura ARM

- ▶ Wszystkie rozkazy wykonują się w ściśle określonym czasie – zwykle 1 cykl
- ▶ 4 bitowy kod warunkowy na początku każdej instrukcji
- ▶ Łączenie operacji przesunięcia i obrotu w rejestrze z instrukcjami arytmetycznymi, logicznymi, przesyłania danych np.:
`"a += (j << 2);"`

Algorytm Euklidesa – NWD

- ▶ Kod C

```
int gcd(int i, int j)
{
    while (i != j)
        if (i > j)
            i -= j;
        else
            j -= i;
    return i;
}
```

- ▶ Kod asm

```
loop    b test
        subgt Ri,Ri,Rj
        suble Rj,Rj,Ri
test    cmp  Ri,Rj
        bne  loop
```

Obecnie stosowane ARM

- ▶ Obecnie rdzeń ARM wspiera 32-bit ARM i 16-bit Thumb® Instruction Set Architectures (ISAs)

Rozszerzenia dla

- ▶ Java acceleration (Jazelle™)
- ▶ security (TrustZone™)
- ▶ Intelligent Energy Manager (IEM)
- ▶ SIMD
- ▶ NEON™ technologies.

Obecnie stosowane ARM

ARMv4

- ▶ Najstarszy z obecnie jeszcze wspieranych rdzeni, implementowany w niektórych ARM7™ oraz Intel StrongARM®
- ▶ 32bit ISA w 32bit przestrzeni adresowej

ARMv4T

- ▶ Dodany 16-bit tryb Thumb® – oszczędność do 35% kodu

Obecnie stosowane ARM

ARMv5TE (1999)

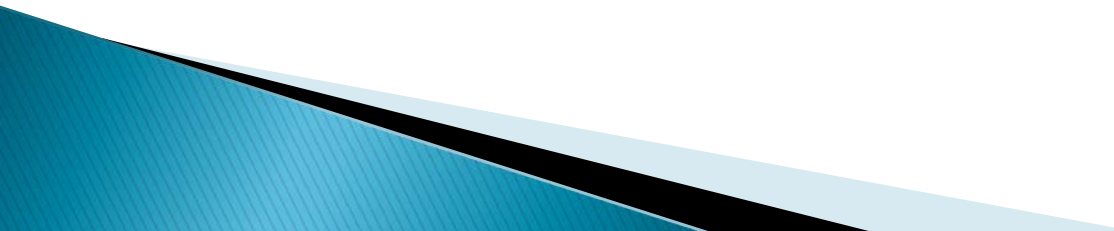
- ▶ Wprowadzone ulepszenia Thumb
- ▶ ARM 'Enhanced' DSP – możliwości DSP na zwykłym CPU

ARMv5TEJ (2000)

- ▶ Jazelle Technology – wspieranie sprzętowe a nie programowe (JVM) obsługi Java. Przyspieszenie 8x, 80% mniejsze zapotrzebowanie na moc.

Obecnie stosowane ARM

ARMv6 (2001)

- ▶ Ulepszenia w systemie pamięci, przechwytywanie wyjątków, środowisko wieloprocessorowe
 - ▶ SIMD (Single Instruction Multiple Data)
 - ▶ Thumb-2 oraz TrustZone jako opcje ARMv6
- 

Obecnie stosowane ARM

ARMv7 – żyje pod rodziną procesorów Cortex

3 Profile:

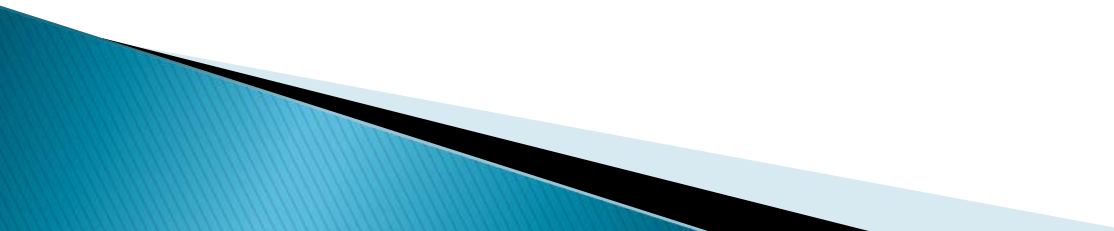
- ▶ A – systemy z pamięcią wirtualną oraz aplikacje użytkownika
- ▶ R – Real Time Systems
- ▶ M – mikrokontrolery oraz aplikacje o niskich kosztach (Cortex-M3)

Każdy profil implementuje w sobie technologie Thumb-2

Technologia NEON

- ▶ NEON Media Acceleration Technology
- ▶ Jest to architektura hybrydowa 64/128 bit SIMD
- ▶ Stworzona w celu przyspieszenia wydajności aplikacji multimedialnych oraz przetwarzania sygnałów (kodowanie/dekodowanie video, grafika 3D, przetwarzanie mowy, dekodowanie dźwięku, przetwarzanie obrazu, telefonia, synteza dźwięku)
- ▶ OpenMAX API

Vector Floating Point (VFP)

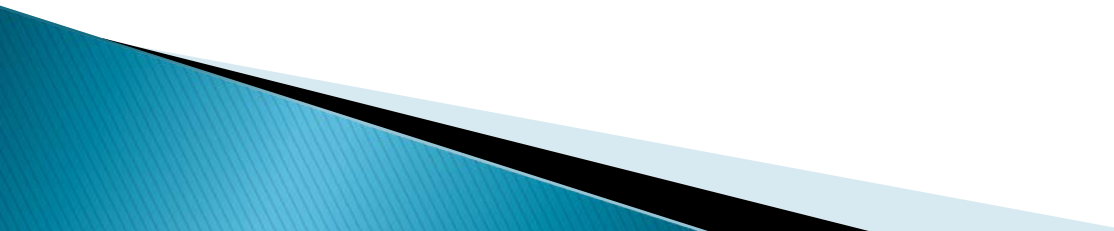
- ▶ Opcjonalna architektura koprocessora
 - ▶ Obsługuje arytmetykę pojedynczej oraz podwójnej precyzji
 - ▶ W pełni kompatybilna z IEEE 754
 - ▶ Wpomaga takie operacje jak skalowanie, transformacja 2D oraz 3D, filtracja cyfrowa
 - ▶ Obecnie implementowana w ARM9, ARM10, ARM11
- 

ARM TrustZone

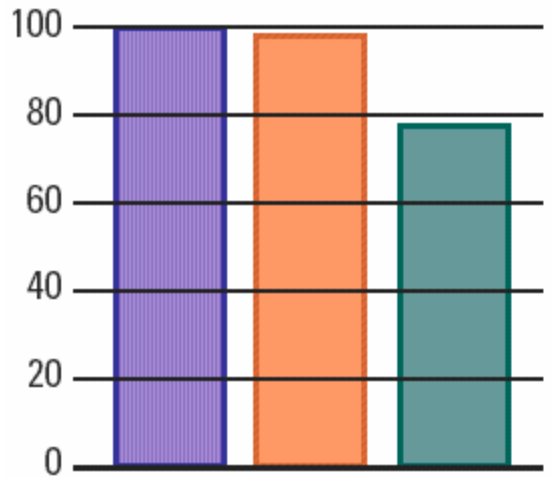


- ▶ Rozszerzenie bezpieczeństwa dla architektury ARM
- ▶ Implementacja w procesorze chroniąca pamięć wewnętrzną jak i zewnętrzną przed atakami ze strony programu
- ▶ Technologia dostarcza bezpieczne środowisko dla elementów systemowych takie jak: zarządzanie kluczami, authentication

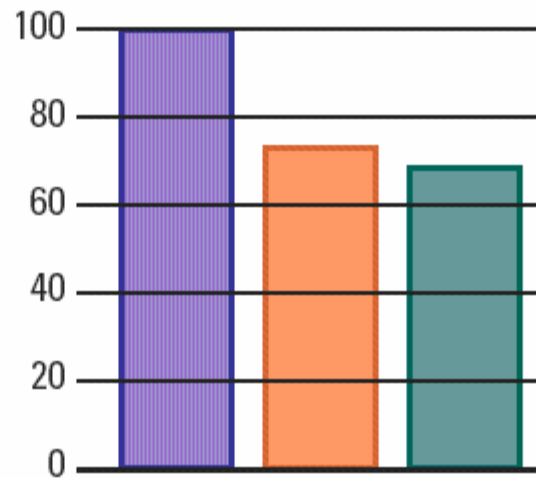
Thumb

- ▶ Znakomite upakowanie kodu
 - ▶ Wydajność 32-bitowa na pamięciach 8/16 bit przy magistrali 8/16bit
 - ▶ MIPS/Wat , RISC
 - ▶ Małe wymiary układu - niski koszt produkcji
 - ▶ Dekompresja oraz dekodowanie bez strat na wydajności
- 

ARV vs Thumb vs Thumb-2



Thumb-2 Performance
25% faster than Thumb

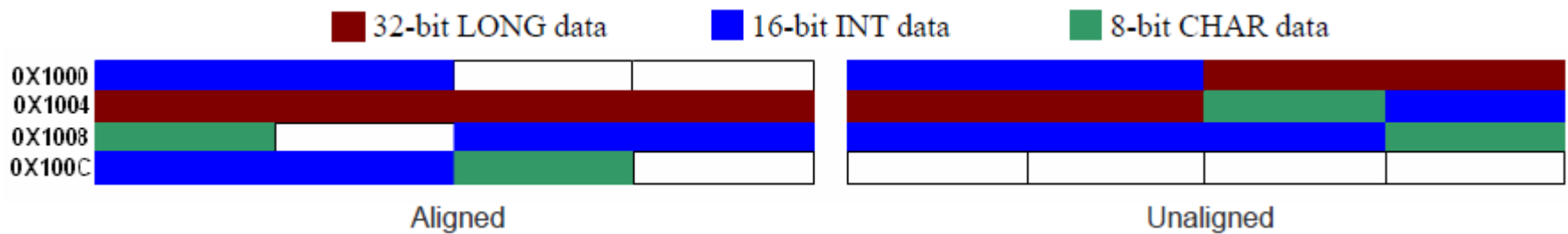


Thumb-2 code size
26% smaller than ARM



Dostęp do pamięci

- ▶ Bit-banging – manipulacja bitami: 1MB pamięci jest rzutowany na 32MB lokalizacji bitowej.
- ▶ Dostęp do pamięci z adresem niewyrównanym

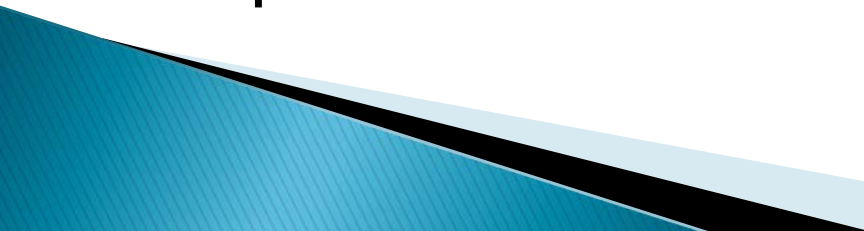


Profile procesorów

- ▶ Aplikacje („Application”) – Cortex–A
- ▶ Czasu rzeczywistego („Real–Time) – Cortex–R
- ▶ Mikrokontroler („Microcontroller”) – Cortex–M

- ▶ Przykłady:
 - ▶ Rdzeń: Cortex–M3 profil: ARMv7–M (thumb2)
 - ▶ Rdzeń: Cortex–M0 profil: ARMv6–M (mniej instrukcji)

Systemy operacyjne

- ▶ Windows CE, .NET Micro Framework, Symbian OS, FreeRTOS, eCos, INTEGRITY, Nucleus PLUS, MicroC/OS-II, QNX, RTEMS, BRTOS, RTXC Quadros, ThreadX, Unison OS, uTasker, VxWorks, MQX oraz OSE
 - ▶ Apple iOS, Android, BSD, GNU/Linux, Inferno, Plan 9 from Bell Labs, Solaris, WebOS
 - ▶ Linux
 - ▶ BSD (FreeBSD, iOS, OpenBSD)
 - ▶ OpenSolaris
- 

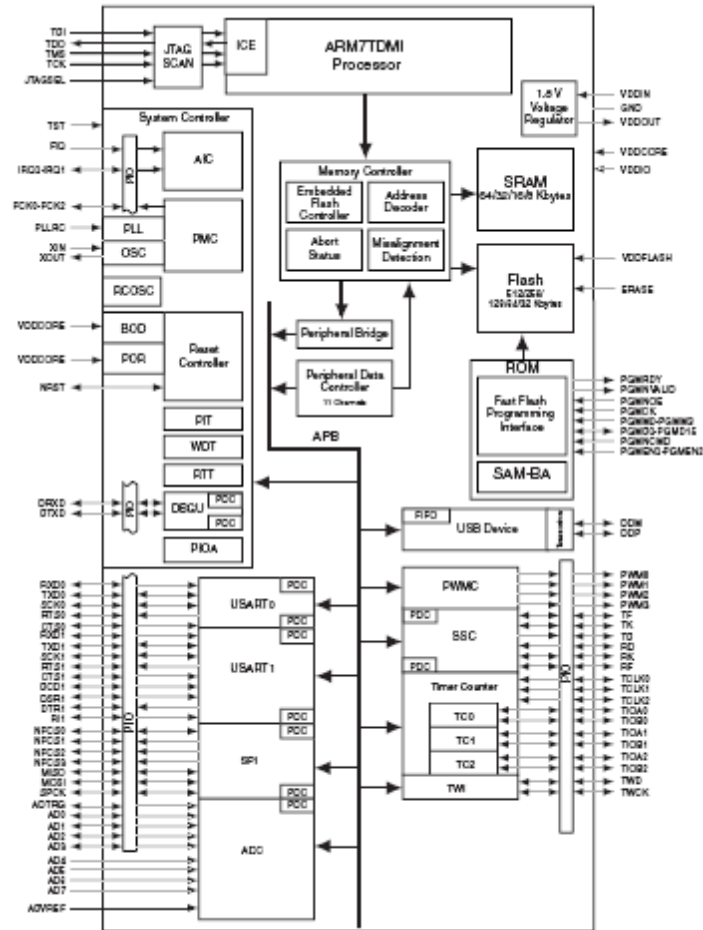
ARM7TDMI

AT91SAM7S64

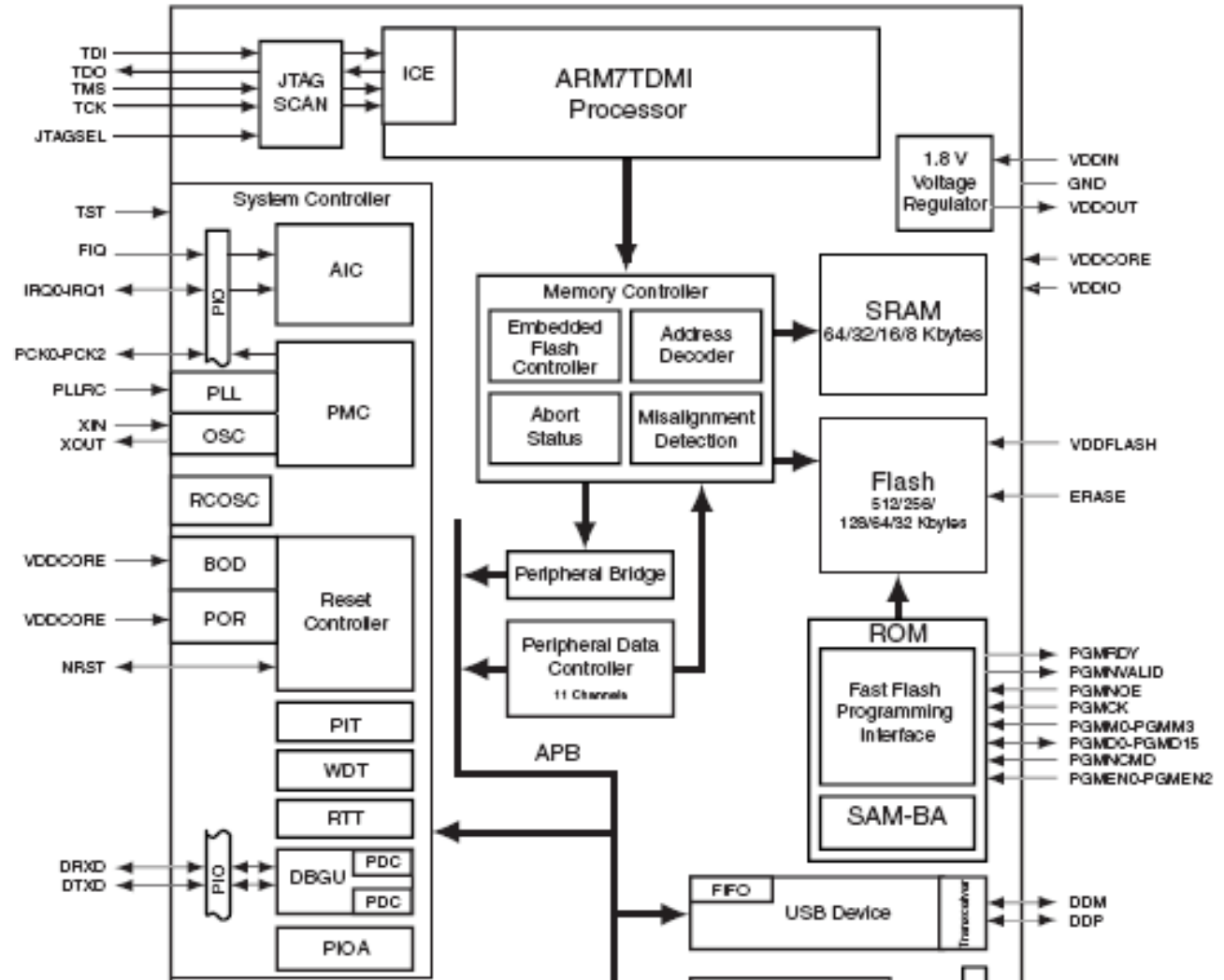
- ▶ 64kB FLASH
- ▶ 512 stron / 128B każda
- ▶ Dostęp do pamięci w pojedynczym cyklu
- ▶ Flagi zabezpieczania sektorów oraz pamięci
- ▶ 16kB SRAM

Wykonywanie rozkazów w 3 stopniowym pipeline (Instruction Fetch (F), Decode (D), Execute (E))

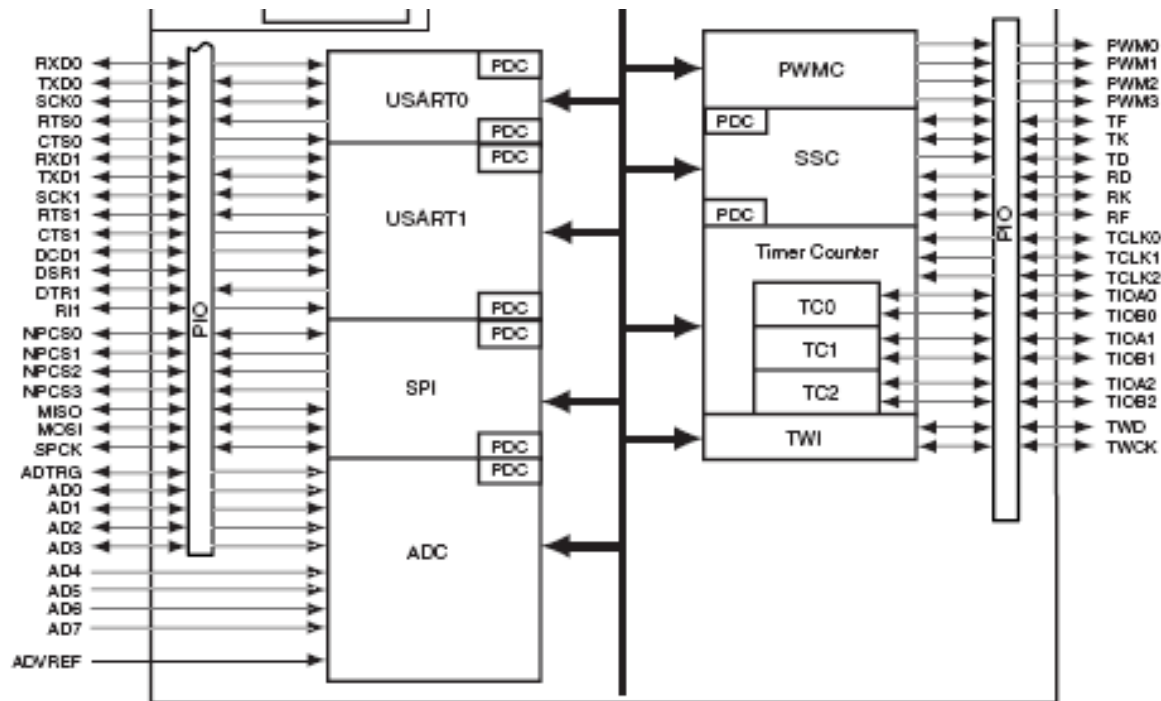
AT91SAM7S



AT91SAM7S



AT91SAM7S



AT91SAM7S

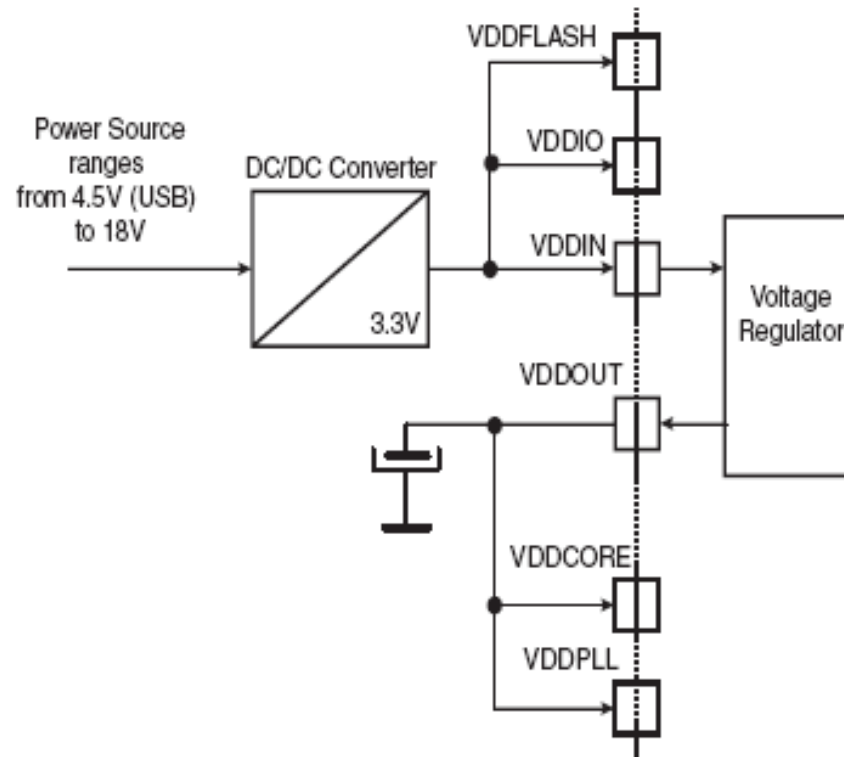
Zasilanie:

- ▶ VDDIN (zasilanie Vreg, ADC – 3.0 – 3.6V)
- ▶ VDDOUT (wyjście z Vreg, 1,8V)
- ▶ VDDIO (zasilanie lini wejścia/wyjścia oraz USB 3.3V lub 1.8V)
- ▶ VDDFLASH (3.3V)
- ▶ VDDCORE (1.8V)
- ▶ VDDPLL (VDDOUT)

Pobór prądu: statyczny 60uA, dynamiczny 50mA

AT91SAM7S

► Zasilanie



AT91SAM7

Tryb DEBUG

Zintegrowany EmbeddedICE (embedded in-circuit emulator) :

- ▶ Dostęp przez JTAG
- ▶ Dwa układy watchpoint
- ▶ Kanał dla wymiany informacji debugującej

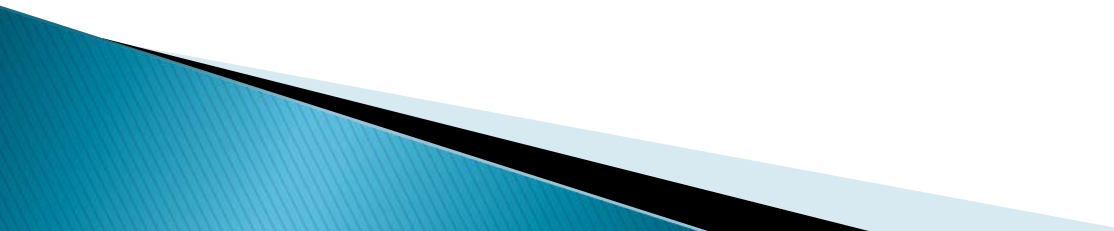
Układ debug:

- ▶ 2 pin UART, chip ID, debug interrupt

JTAG

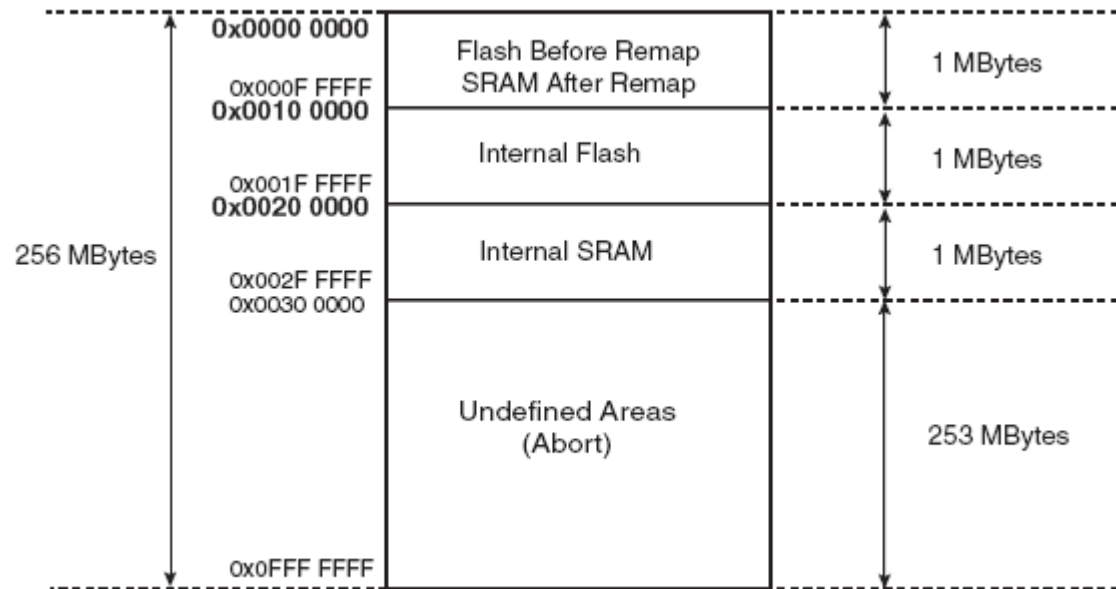
AT91SAM7S

Kontroler Pamięci (Memory Controller)

- ▶ Blokowanie dostępu do pamięci
 - ▶ Zapis/odczyt z pamięci
 - ▶ Sprawdzanie poprawności adresowania danych oraz wyrównania danych
 - ▶ Przemapowanie pamięci (uruchamianie programu z pamięci SRAM)
- 

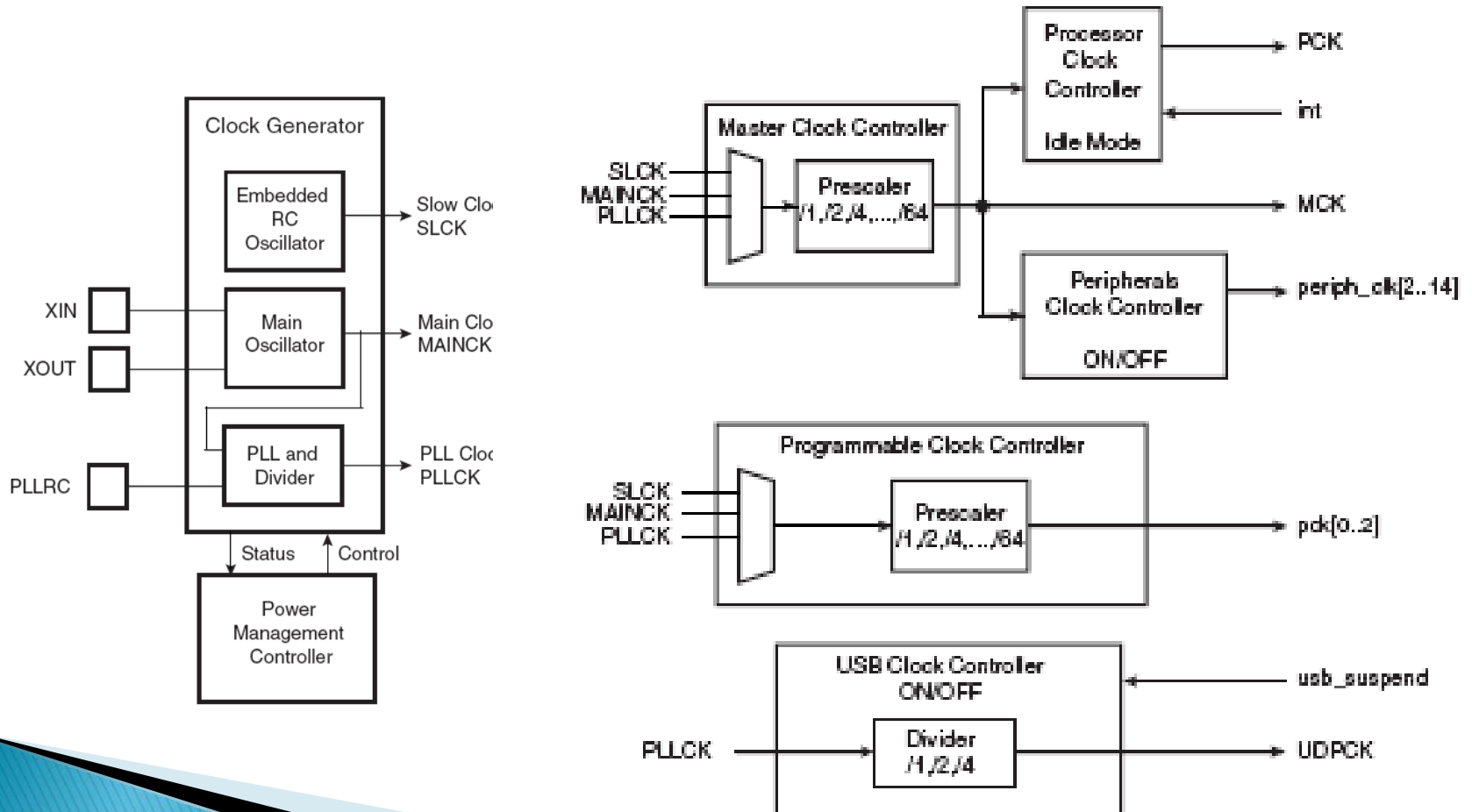
AT91SAM7S

► Organizacja pamięci



AT91SAM7S

▶ Zegar oraz Power Management Controller



AT91SAM7S

- ▶ AIC (Advanced Interrupt controller) – sprawdza nIRQ, nFIQ; przerwania maskowalne oraz wektorowane, 8 poziomów priorytetów,
- ▶ PIT (Periodic Interval Timer) – 20bit programowalny licznik + 12bit licznik okresu
- ▶ WatchDog timer – 12bit klucz, SCK, może generować reset lub programowy int, może być wyłączony gry w trybie debug
- ▶ Real-Time Timer – 32bity, alarm, SCK, programowalny preskaler 16bit

AT91SAM7S

Interfejsy

- ▶ SPI – Serial Peripheral Interface (4 pinowy wybór układu slave, Master lub Slave, 8/16bit data, programowalna faza i polaryzacja pinów wyboru, programowanie opóźnienia)
- ▶ TWI – two-wire interface
- ▶ USART – Universal Synchronous asynchronous Receiver Transceiver (RS232, RS485, IrDA) możliwość ustawienia lokalnego echa
- ▶ SSC – Serial Synchronous Controller
- ▶ Timer Counter – pomiar częstotliwości, zliczanie zdarzeń, pomiar okresu, generowanie impulsów, Delay, PWM

AT91SAM7S

- ▶ PWM (Pulse Width modulation) – 4 kanały
- ▶ USB (2.0)
- ▶ ADC (Analog Digital Converter) – 8 kanałów, 10-bit 384k sample/sec; Wyzwalanie sprzętowe/programowe, poprzez pin zewnętrzny, TC 0 do 2
- ▶ Kontroler DMA – komunikacja pomiędzy pamięcią a urządzeniem peryferyjnym (USART, debug, SPI, SSC, ADC)

Cortex-M3



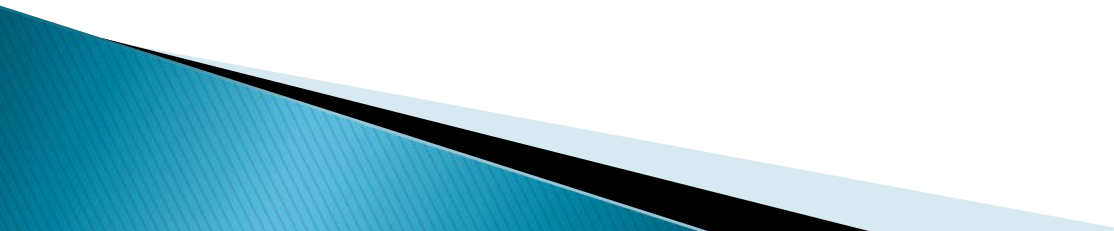
LM3S811

- ▶ 32bit ARM Cortex-M3 v7M
- ▶ Thumb, Thumb-2
- ▶ 50MHz
- ▶ 64kB FLASH
- ▶ 8KB SRAM

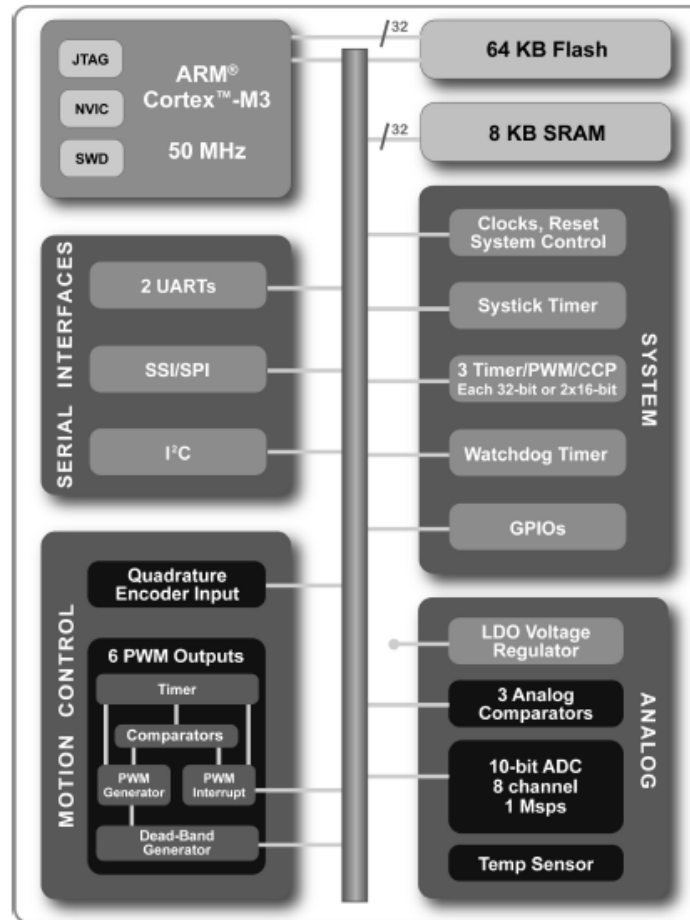
LM3S811

- ▶ SystemTimer (SysTick)
- ▶ Sprzętowe dzielenie oraz jedno-cyklowe mnożenie
- ▶ Zintegrowany NVIC (Nested Vectored Interrupt Controller)
- ▶ 26 przerw z 8 poziomami
- ▶ Memory protection Unit
- ▶ Pozwala na dostęp do danych niewyrównanych
- ▶ Atomowa manipulacja bitami

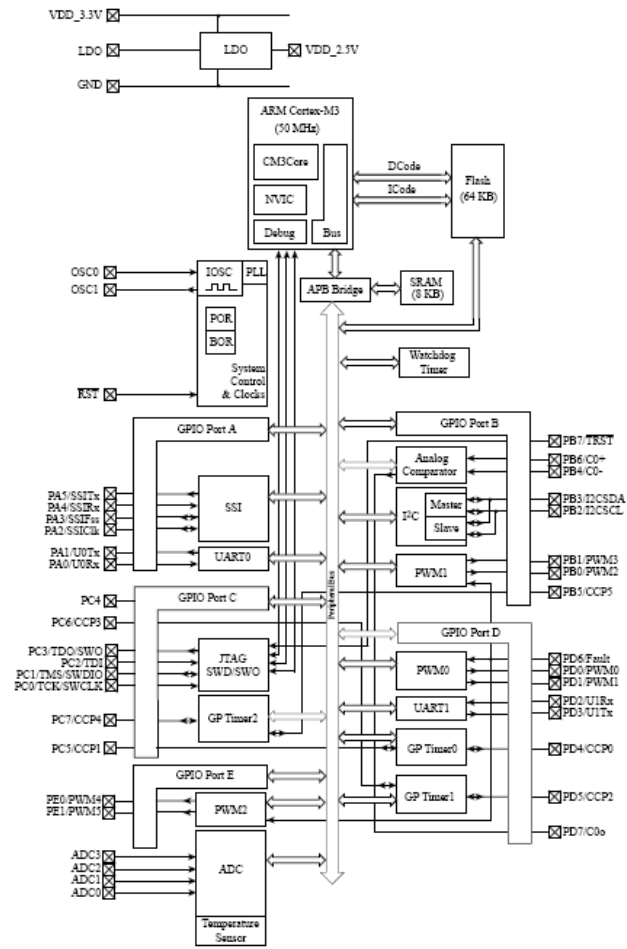
LM3S811

- ▶ General-Purpose Timers (3 sztuki, 16bit, 32bit)
 - ▶ Watchdog
 - ▶ SSI (wewnętrzny loopback)
 - ▶ UART (16x8 TX; 16x12 RX)
 - ▶ ADC (4 x 10bit, 500ksample/sec, temp sens)
 - ▶ Analog Comperator
 - ▶ I2C
 - ▶ PWM (3bloki, 16bit, każdy dwa komparatory)
- 

LM3S800

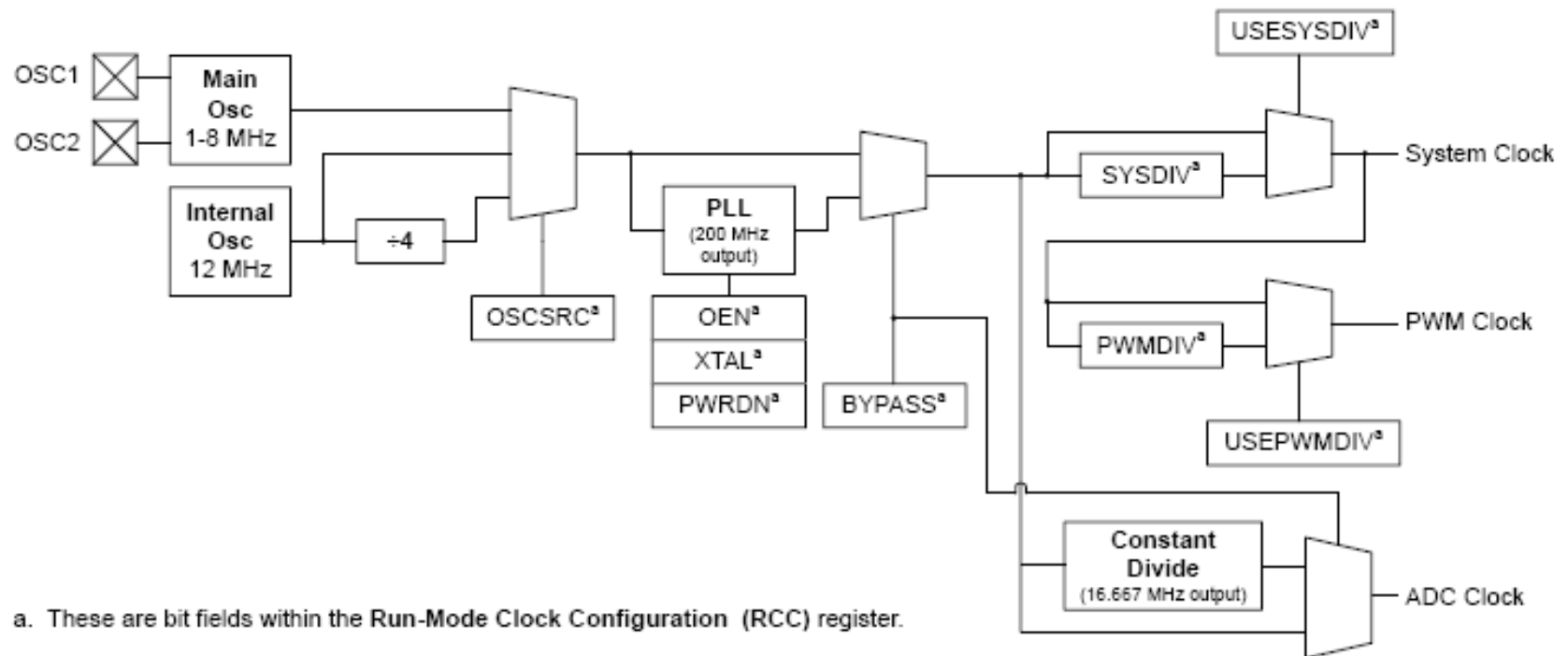


LM3S811



LM3S811

▶ ZEGAR



a. These are bit fields within the Run-Mode Clock Configuration (RCC) register.

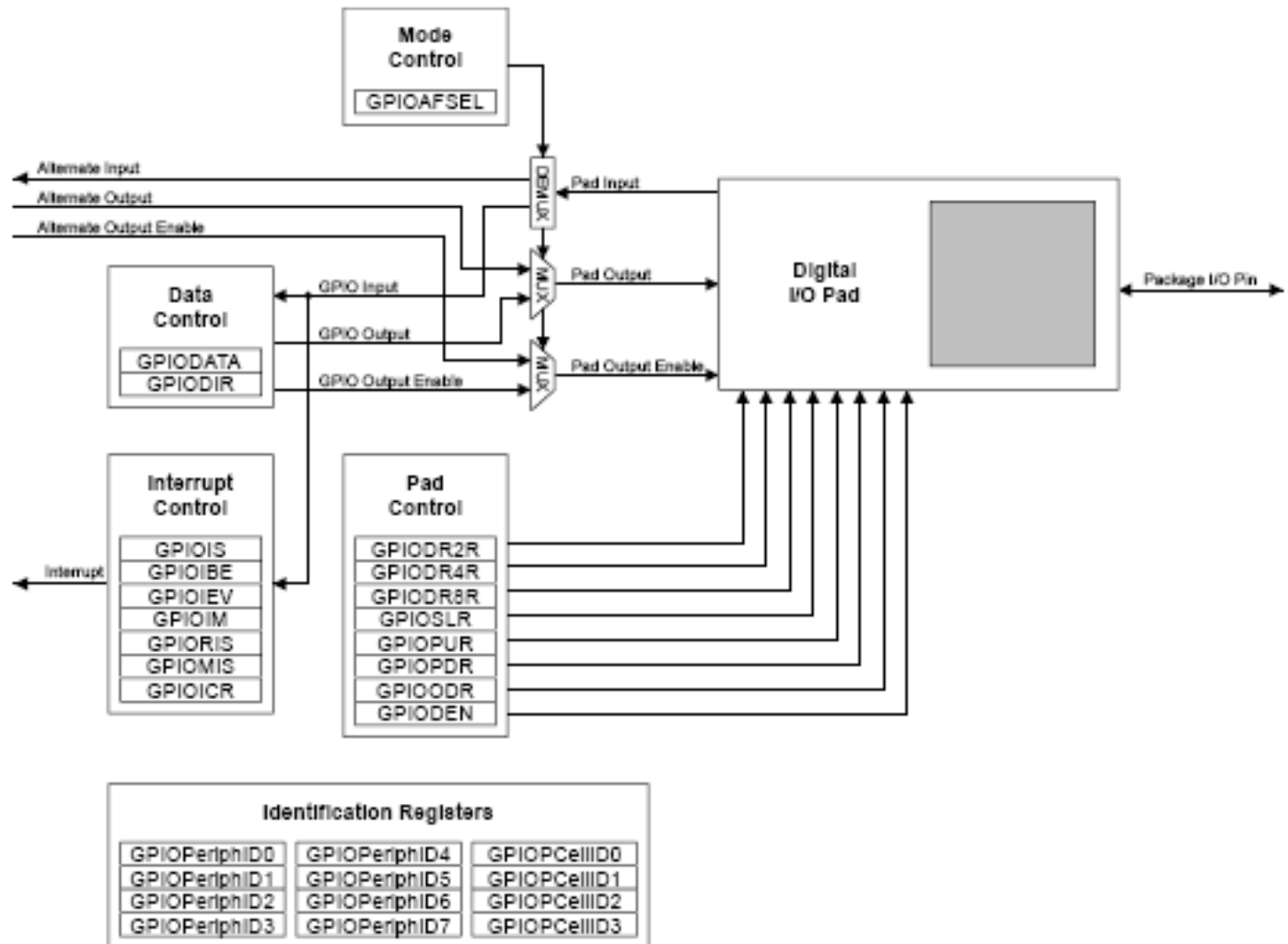
LM3S811

Zasilanie

- ▶ Supply 3.0 – 3.6V
- ▶ Zintegrowany LDO (Low Drop-Out) regulator 2.25 – 2.75V
- ▶ Pobór prądu max 95mA (50MHz, PLL, while(1), FLASH, Periph ON)

LM3S811

▶ GPIO



STM32F103

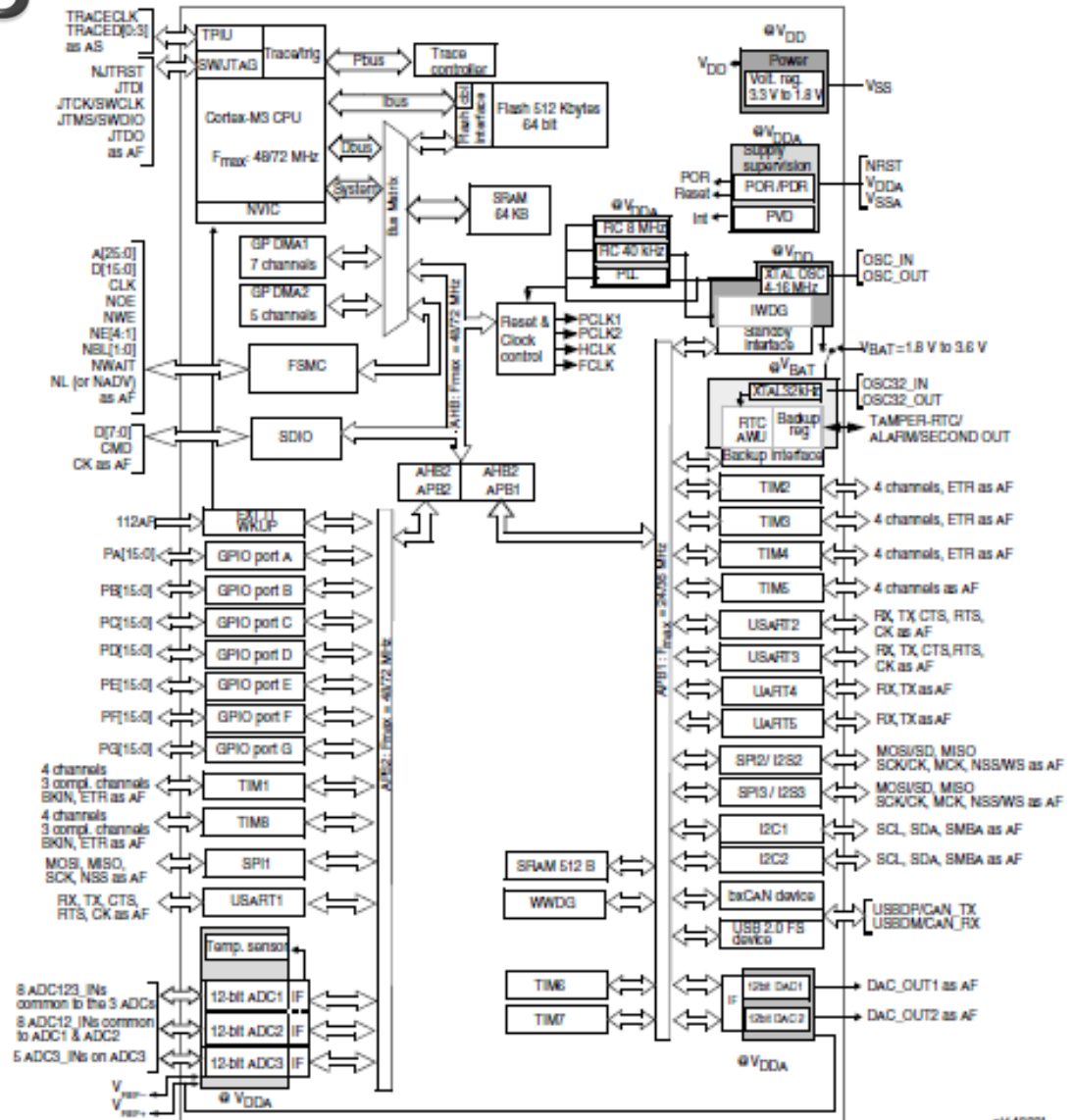
- ▶ ARM® Cortex™-M3
- ▶ Flash aż do 512kB
- ▶ Moduł CRC
- ▶ SRAM aż do 64kB
- ▶ FSMC (Flexible Static Memmory Controller):
PC Card/Compact Flash, SRAM, PSRAM, NOR,
NAND
- ▶ Port równoległy LCD
- ▶ NVIC (Nested Vector Interrupt Controller) – 60
przerwać i 16 poziomów
- ▶ EXTI (External Interrupt Controller)

STM32F103

- ▶ Tryby bootowania:
 - Flash
 - Pamięć systemowa (bootloader z aktywnym USART1)
 - SRAM
- ▶ Zasilanie – 2,0 – 3,6V
- ▶ POR (Power-On Reset)/PDR (Power-Down Reset)
- ▶ Tryby oszczędzania zasilania
- ▶ DMA, RTC, Timery, I2C, USART, SPI, I2S, SDIO(SD/SDIO/MMC), CAN, USB, GPIO,
- ▶ ADC, DAC, Czujnik temperatury

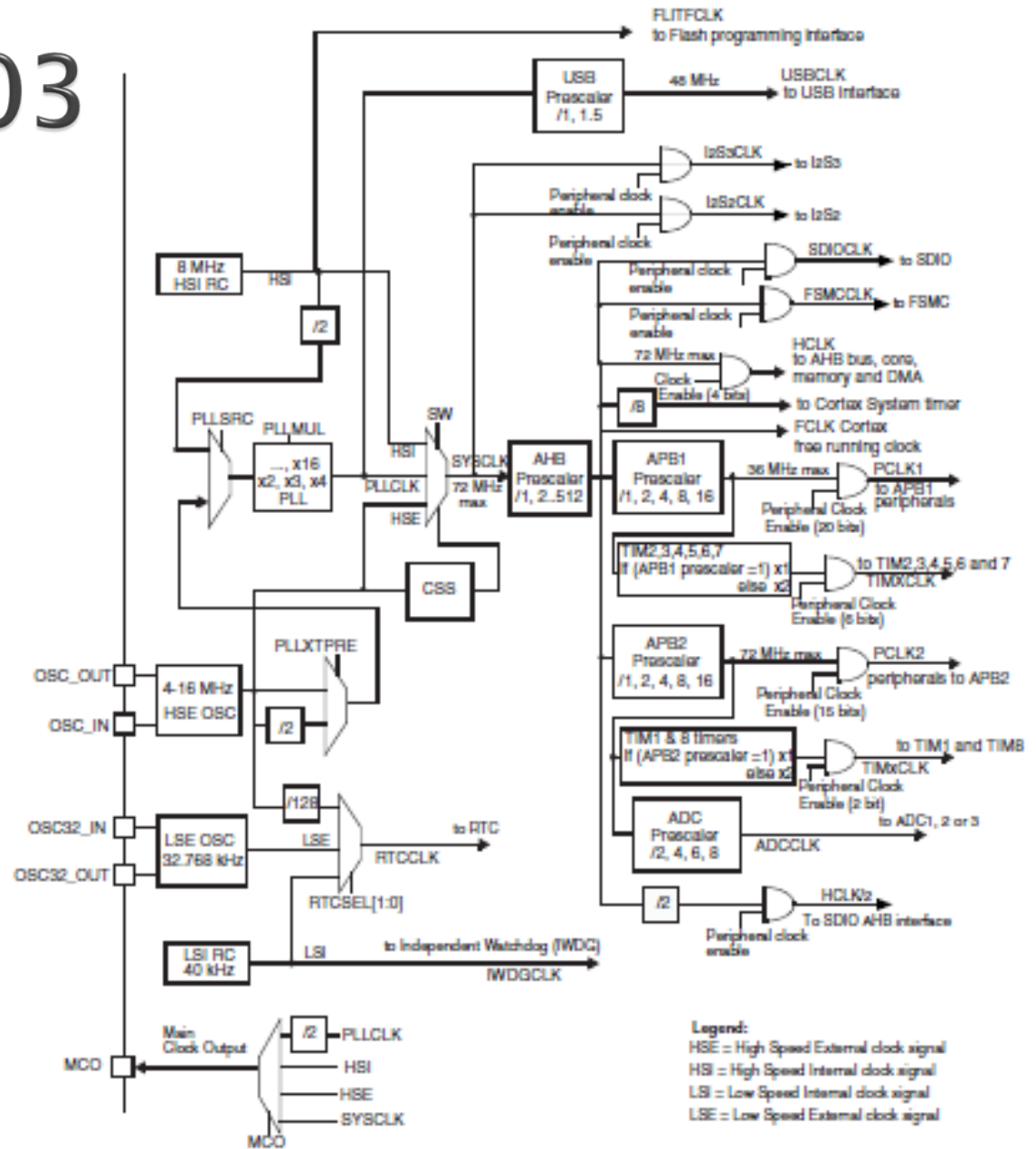
STM32F103

Schemat blokowy



STM32F103

Zegar



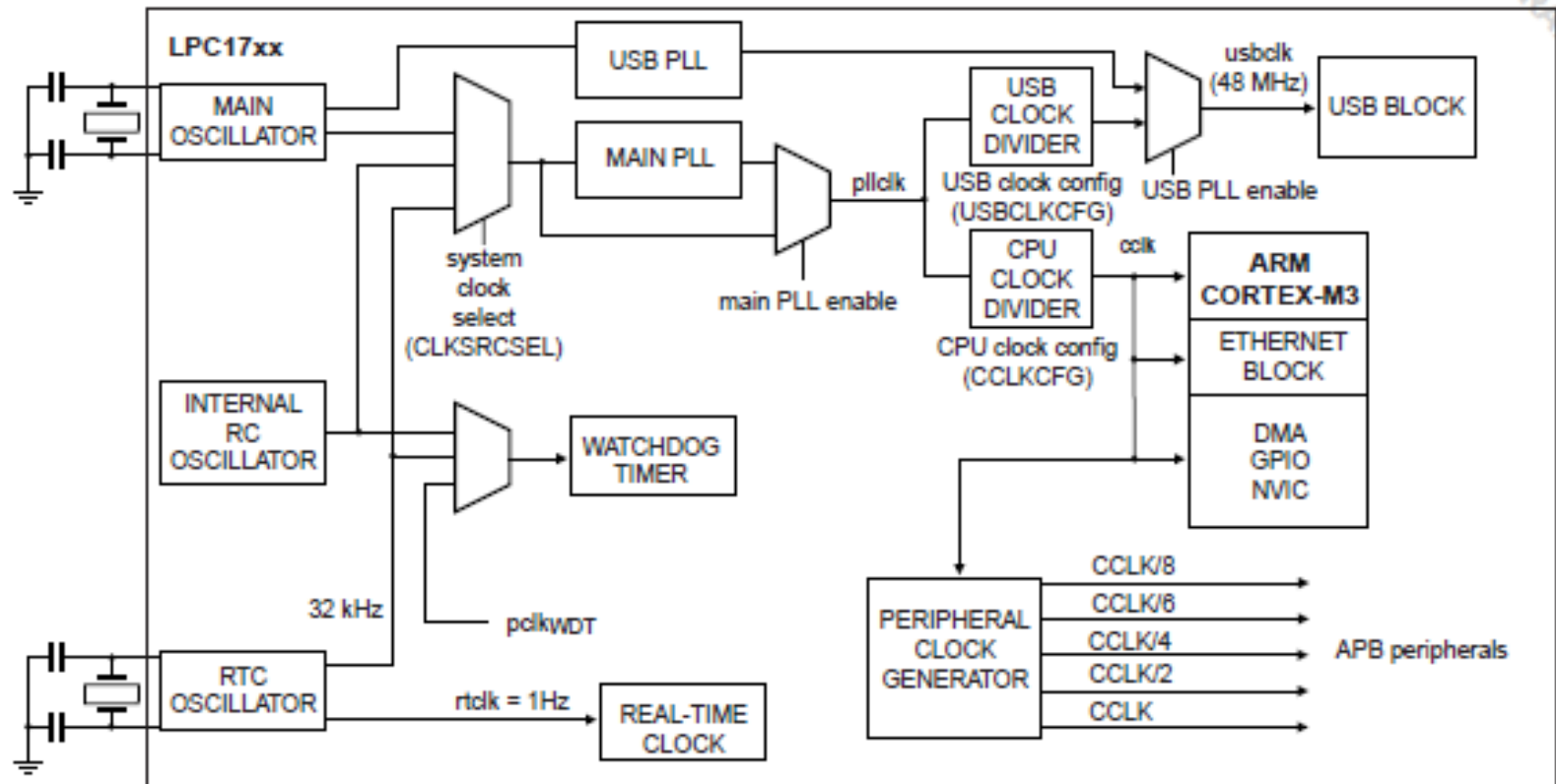
STM32F103 – kompatybilność

Pinout	Low-density devices		Medium-density devices		High-density devices		
	16 KB Flash	32 KB Flash ⁽¹⁾	64 KB Flash	128 KB Flash	256 KB Flash	384 KB Flash	512 KB Flash
	6 KB RAM	10 KB RAM	20 KB RAM	20 KB RAM	48 KB RAM	64 KB RAM	64 KB RAM
144					5 × USARTs		
100					4 × 16-bit timers, 2 × basic timers		
64	2 × USARTs 2 × 16-bit timers 1 × SPI, 1 × I ² C, USB, CAN, 1 × PWM timer		3 × USARTs 3 × 16-bit timers 2 × SPIs, 2 × I ² Cs, USB, CAN, 1 × PWM timer 2 × ADCs		3 × SPIs, 2 × I ² Ss, 2 × I ² Cs		
48					USB, CAN, 2 × PWM timers		
36					3 × ADCs, 2 × DACs, 1 × SDIO FSMC (100- and 144-pin packages ⁽²⁾)		

LPC1768

- ▶ Do 100MHz
- ▶ NVIC
- ▶ ISP (In-System Programming) IAP (In-Application Programming)
- ▶ Flash max 512kB
- ▶ 32/16kB SRAMs dla CPU oraz jeden/dwa 16kB SRAM z osobną ścieżką
- ▶ Ethernet, USB Host/Device/OTG, UART (RS232,RS485,IrDA), CAN 2.0B, SPI, SSP, I2C, I2S
- ▶ 70x GPIO, 12bit ADC, 10bit DAC, timery, PWM, RTC, ...

LPC1768 - zegar

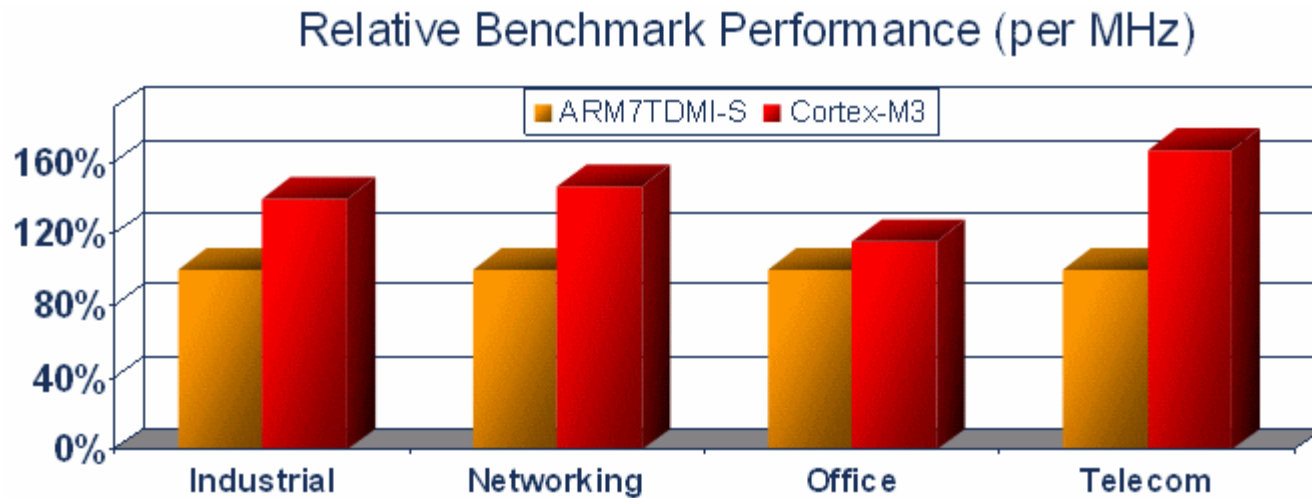


DRAFT

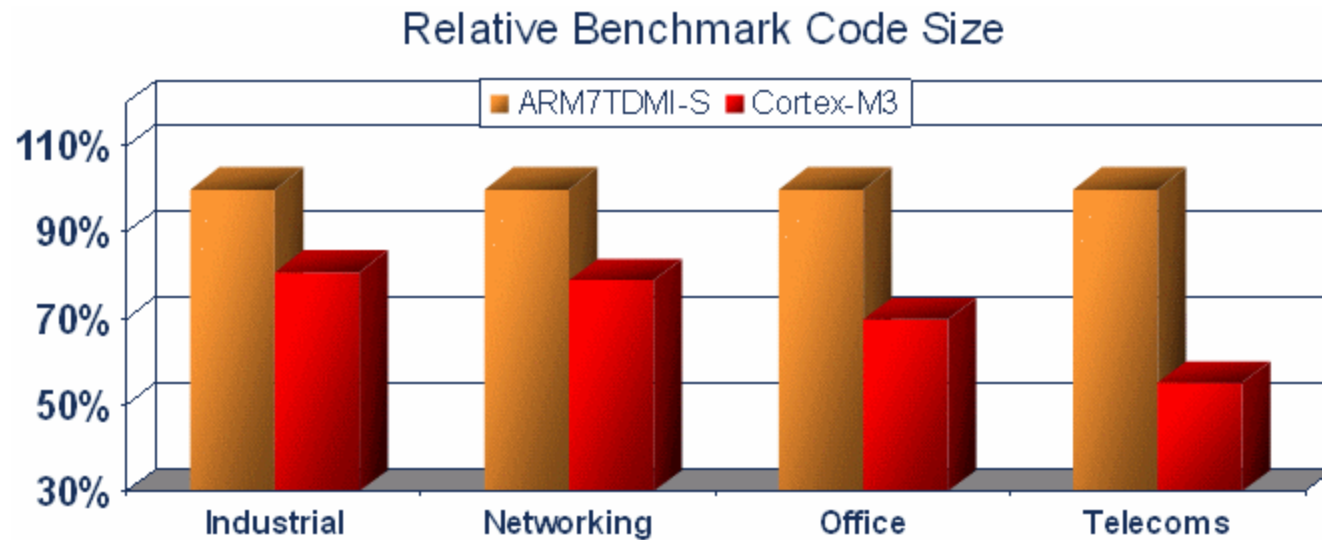
ARM7TDMI vs Cortex-M3

Features	ARM7TDMI	ARM Cortex-M3
Architecture	ARMv4T (von Neumann)	ARMv7-M (Harvard)
ISA Support	Thumb/ARM	Thumb/Thumb-2
Pipeline	3-stage	3-stage + branch speculation
Interrupts	FIQ/IRQ	NMI + 1 to 240 physical interrupts
Interrupt Latency	24 - 42 cycles	12 cycles
Inter-Interrupt Latency	24 cycles	6 cycles
Sleep Modes	None	Integrated
Memory Protection	None	8 region MPU
Dhrystone	0.95DMIPS/MHz (ARM) 0.74DMIPS/MHz (Thumb)	1.25DMIPS/MHz
Power Consumption	0.28mW/MHz	0.19mW/MHz
Area	0.62mm ²	0.82mm ²

ARM7TDMI-S vs Cortex-M3



ARM7TDMI-S vs Cortex-M3



CMSIS

- ▶ **Cortex Microcontroller Software Interface Standard**
- ▶ Warstwa opisu sprzętu niezależna od producenta
- ▶ Składa się z następujących komponentów:
 - CMSIS-CORE – oferują interfejs do Cortex-M0, Cortex-M3, Cortex-M4, SC000, SC300
 - CMSIS-DSP – biblioteka DSP (ponad 60 funkcji w stałoprzecinkowej i zmiennoprzecinkowej implementacji)
 - CMSIS-RTOS API
 - CMSIS-SVD – System View Description XML