

STM32F2 czy STM32F4: który do czego?

Rodzina mikrokontrolerów STM32 stała się w wielu krajach synonimem rdzenia Cortex-M3, którego to poglądu nie zaburzyły nawet ostatnie nowości w ofercie produkcyjnej STMicroelectronics: mikrokontrolery z rdzeniem Cortex-M4 (STM32F4) i z rdzeniem Cortex-M0 (STM32F0). Pomimo podobieństw pomiędzy Cortex-M3 i M4, nie są to takie same rdzenie, a pozornie drobne różnice mogą być czasami odczuwalne jak w promowane w reklamach „prawie, które robi wielką różnicę”...

W artykule zajmiemy się przybliżeniem cech i różnic pomiędzy rdzeniami Cortex-M3 (stosowanymi we wszystkich podrodzinach STM32F1 oraz F2) i Cortex-M4 (stosowanymi w podrodzinie STM32F4), szczególnie istotnymi w przypadku korzystania z mikrokontrolerów należących do podrodzin STM32F2 i STM32F4, które są ze sobą kompatybilne mechanicznie i elektrycznie.

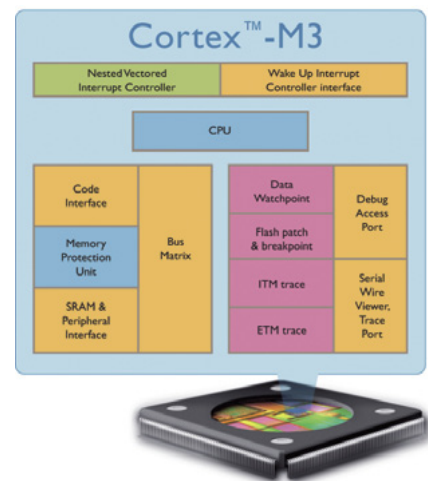
Na **rysunku 1 i 2** przedstawiono (nieco uproszczone) schematy blokowe obydwu rdzeni, co pozwala porównać ich budowę i ocenić różnice w wyposażeniu. Jak widać, na poziomie blokowym, różnice pomiędzy rdzeniami nie są duże, co powoduje że w typowych aplikacjach przy zbliżonych częstotliwościach taktowania ich wydajności są porównywalne (prędkość wykonywania standardowych programów przez mikrokontrolery F2 i F4 wynosi 1,25 MIPS/

MHz @Dhystone 2.1). Nie oznacza to jednak, że nie ma pomiędzy podrodzinami STM32F2 i STM32F4 różnic wpływających na wypadkową wydajność ich CPU.

STM32F2 vs STM32F4 – podstawowe różnice

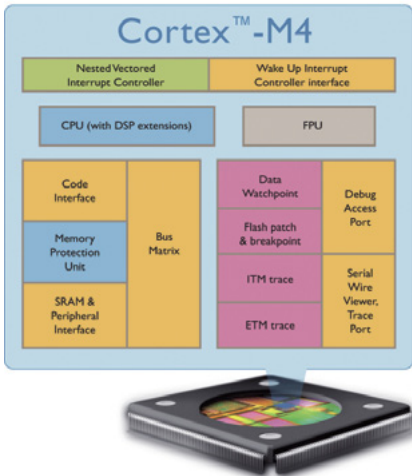
Rdzeń ARM Cortex-M4 to jedno z najnowszych opracowań „mikrokontrolerowych” firmy ARM, przeznaczone dla aplikacji cyfrowych układów regulacji i sterowania, którym stawia się wymagania: aby były wydajne i miały łatwe w użyciu funkcje sterowania i przetwarzania sygnałów w aplikacjach mikrokontrolerów.

Rdzeń Cortex-M4 jest standardowo wyposażony w jednostkę obliczeniową MAC (*Multiply and Accumulate*), która służy do wykonywania operacji mnożenia i sumowania w jednym cyklu zegarowym, wykonuje także instrukcje SIMD (*Single*



Rysunek 1. Schemat blokowy rdzenia Cortex-M3

Instruction, Multiple Data) – zoptymalizowanych operacji na wielu danych, instrukcje arytmetyki nasyceniowej (*saturating arithmetic*). Mikrokontrolery STM32F4 są także standardowo wyposażone w zmienoprzecinkowy koprocesor obliczeniowy FPU (*Floating-Point Unit*) służący do wykonywania operacji na liczbach zmienoprzecinkowych pojedynczej precyzji. Koprocessor FPU zastosowany w mikrokontrolerach STM32F4 jest implementacją wariantu rozszerzenia zmienoprzecinkowego *Flo-*



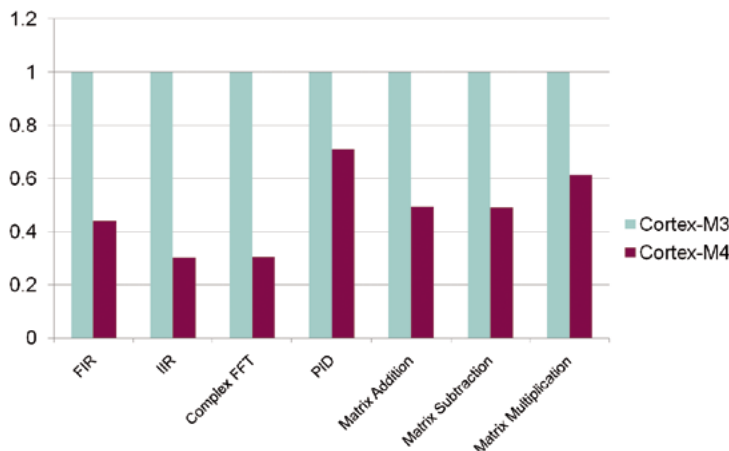
Rysunek 2. Schemat blokowy rdzenia Cortex-M4

ating-Point Extension (FPv4-SP) architektury ARMv7-M. Zapewnia on obliczenia na liczbach zmiennoprzecinkowych, spełniając normy ANSI/IEEE STD 754-2008 oraz normy IEEE dotyczące obliczeń na liczbach zmiennoprzecinkowych (Standard for Binary Arithmetic Floating-Point, IEEE 754).

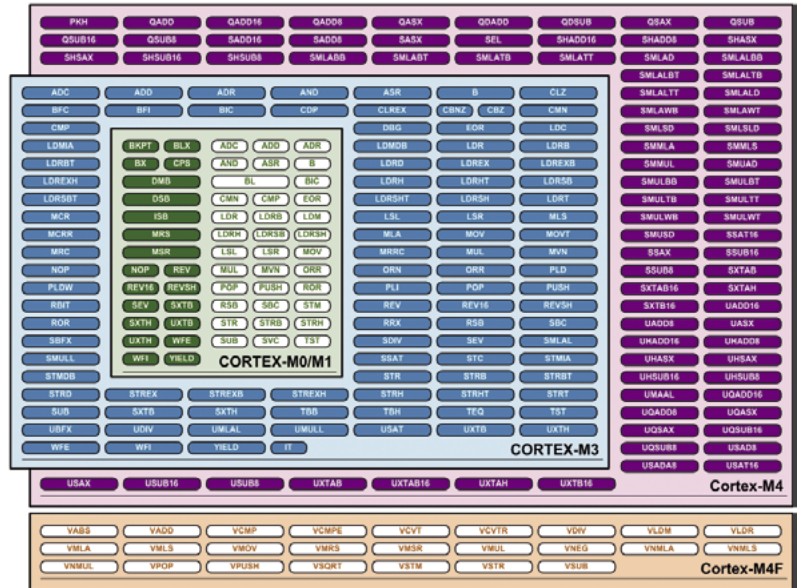
Wbudowane w STM32F4 FPU ma cztery główne funkcjonalności:

- udostępnia rozszerzone rejestry zmiennoprzecinkowe, które oprogramowanie traktuje jak 32 rejestry pojedynczej precyzji lub jak 16 rejestrów podwójnego słowa,
- wykonuje operacje arytmetyczne na liczbach zmiennoprzecinkowych pojedynczej precyzji,
- przeprowadza konwersję formatów między liczbami całkowitymi, zmiennoprzecinkowymi pojedynczej precyzji, i zmiennoprzecinkowymi połowkowej precyzji (16-bitowymi),
- przesyła dane z rejestrów pojedynczej precyzji i podwójnego słowa.

Wbudowana w rdzeń mikrokontrolerów STM32F4 jednostka MAC wykonuje w jednym taktie zegara jednoczesne operacje mnożenia i sumowania liczb 16 i 32-bitowych,



Rysunek 4. Porównanie czasu wykonania różnych obliczeń przez rdzenie Cortex-M3 i Cortex-M4



Rysunek 3. Instrukcje obsługiwane przez rdzenie Cortex-M0, M1, M3, M4 i M4F

owych, a zestaw instrukcji SIMD (Single Instruction Multiple Data) umożliwia wykonywanie w jednym cyklu pracy operacji na dwóch liczbach 16-bitowych i czterech liczbach 8-bitowych.

Jednostka MAC może wykonać:

- jedną operację mnożenia dwóch liczb 32-bitowych i następnie sumowania wyniku z liczbą 64-bitową, dając w rezultacie liczbę 64-bitową: $32 \times 32 + 64 > 64$,
- dwie operacje mnożenia dwóch liczb 16-bitowych: 16×16 .

Jednostka MAC powoduje, że cyfrowe przetwarzanie sygnałów jest bardziej efektywne i zmniejsza zużycie zasobów rdzenia, co wynika z udostępnianych przez MAC funkcjonalności:

- wielu instrukcji operacji mnożenia i sumowania,
- sprzętowego mnożenia liczb 16- lub 32-bitowych oraz sumowania liczb 32- lub 64-bitowych,
- wykonywania tych instrukcji w jednym cyklu pracy.

Ponieważ wyposażenie sprzętowe rdzeni Cortex-M4 jest bogatsze niż w Cortex-M3, rdzenie te obsługują większą liczbę instrukcji - ich listy pokazano na **rysunku 3**.

Na **rysunku 4** pokazano porównanie czasów wykonania różnych obliczeń przez mikrokontrolery STM32F2 i STM32F4 taktowanymi sygnałami zegarowymi o jednakowych częstotliwościach, przy wykorzystaniu oferowanych przez nie możliwości sprzętowych. Porównanie przygotowała firma STMicroelectronics bazując na przykładowych aplikacjach bazujących na bibliotekach CMSIS i kompilatorze GCC w pakiecie MDK firmy Keil/ARM.

Obsługa instrukcji SIMD pozwala optymalizować operacje wykonywane na wielu danych jednocześnie. W skład zestawu instrukcji SIMD wchodzi niektóre instrukcje DSP, do wykonywania takich operacji jak dodawanie, odejmowanie, mnożenie, mnożenie i sumowanie, które są wykorzystywane do implementacji algorytmów DSP, w tym filtrowania sygnałów cyfrowych za pomocą filtrów o skończonej odpowiedzi impulsowej FIR (Finite Impulse Response) lub o nieskończonej odpowiedzi impulsowej IIR (Infinite Impulse Response), obliczania szybkiej transformaty Fouriera FFT (Fast Fourier Transform) na liczbach zespolonych, obliczania parametrów regulatorów proporcjonalno-całkująco-różniczkujących PID (Proportional-Integral-Derivative) oraz sumowania, odejmowania i mnożenia macierzy.

Instrukcje SIMD umożliwiają (w jednym taktie zegara):

- równoległe wykonanie czterech operacji: dodawania lub odejmowania liczb 8-bitowych,
- wykonanie dwóch operacji równoległych: dodawania lub odejmowania liczb 16-bitowych.

W skrócie można stwierdzić, że rdzeń Cortex-M4 jest lepiej wyposażoną wersją rdzenia Cortex-M3 (stosowanego m.in. w mikrokontrolerach STM32F2), silniejszą przede wszystkim o możliwości łatwej implementacji algorytmów cyfrowego przetwarzania sygnałów (DSP – *Digital Signal Processing*) oraz wbudowaną FPU.

STM32F2 vs STM32F4 – wydajność

Obiektywne porównanie wydajności mikrokontrolerów jest zadaniem dość trudnym: żaden z benchmarków nie jest miarodajny dla wszystkich przypadków aplikacyjnych, a wyniki pomiarów są zależne m.in. od zastosowanego kompilatora i jego konfiguracji, lokalizacji wykonywanego programu, sposobu działania interfejsów dostępu do pamięci, zdefiniowania rodzaju testu (badanie samego rdzenia, układu przerwań, bloków peryferyjnych itp.), a także wielu innych czynników trudnych do określenia, ze względu na często występujący brak dokładnych informacji o szczegółach konstrukcyjnych testowanych układów.

Najprostszym sposobem oceny wydajności jest skorzystanie z wyników zestandardyzowanych testów, jak na przykład CoreMark opracowanych przez EEMBC (*Embedded Microprocessor Benchmark Consortium*), które są ograniczone do badania wydajności rdzenia (CPU), bez analizy (jak tylko się da) wpływu współpracującego z nim otoczenia na wypadkową wydajność. W tabeli 2 zestawiono kilka przykładowych wyników testów CoreMark dla mikrokontrolerów STM32F2 i STM32F4. Jak widać z przedstawionych wyników, same zmiany kompilatora powodują znaczne zmiany wypadkowej wydajności tych samych mikrokontrolerów z CPU taktowanych taką samą częstotliwością. Ponieważ ze względów konstrukcyjnych nie jest możliwe całkowite odizolowanie rdzenia od pozostałych elementów mikrokontrolera – jak choćby od pamięci programu i interfejsów wspomagających dostęp do niej – podane wyniki zawsze będą prawdziwe w określonym kontekście.

Wszystkie wymienione trudności nie oznaczają jednak, że nie ma sposobu szacowanego porównania wydajności mikrokontrolerów, zwłaszcza w przypadku podrodzin tak podobnych jak STM32F2 i STM32F4. Ponieważ ich wewnętrzne peryferia i sposób komunikacji CPU z nimi są w obydwu podrodzinach bardzo podobne, a sprzętowe opóźnienia niezależne podczas dostępu CPU do pamięci Flash identyczne w odpowiednich zakresach częstotliwości taktowania CPU (tabela 3), można założyć – tak wyszło nam podczas eksperymentów w redakcyjnym laboratorium – zmiany wy-

Tab. 1. Porównanie cech rdzeni Cortex-M3 i Cortex-M4/Cortex-M4F mających wpływ na prędkość wykonywania programów

Parametr	Cortex-M3	Cortex-M4
Architektura	ARMv7-M (Harvard)	ARMv7-M (Harvard)
ISA	Thumb/Thumb-2	Thumb/Thumb-2
Rozszerzenia DSP	-	Jednotaktowa, 16- /32-bitowa jednostka MAC. Jednotaktowa, 2x 16-bitowa jednostka MAC. zestaw 8-/16-bitowych instrukcji SIMD Zaimplementowane sprzętowo operacje dzielenia. (2-12 cykli).
FPU	-	Jednostka zmiennooprzecinkowa pojedynczej precyzji. Zgodność z normą IEEE 754. Tylko w Cortex-M4F.
Przetwarzanie potokowe	3-stopniowe + wykonywanie spekulatywne instrukcji za skokiem warunkowym.	3-stopniowe + wykonywanie spekulatywne instrukcji za skokiem warunkowym.
Szybkość przetwarzania – test Dhrystone	1,25 DMIPS/MHz	1,25 DMIPS/MHz
Ochrona pamięci	Opcjonalna jednostka ochrony pamięci (MPU – <i>Memory Protection Unit</i>) dla 8 regionów z opcją wyłączenia podregionów (SRD – <i>Sub Region Disable</i>) i domyślną segmentacją.	Opcjonalna jednostka ochrony pamięci (MPU – <i>Memory Protection Unit</i>) dla 8 regionów z opcją wyłączenia podregionów (SRD – <i>Sub Region Disable</i>) i domyślną segmentacją.
Przerwania	Przerwania niemaskowalne (NMI – <i>Non-Maskable Interrupt</i>) + od 1 do 240 przerwania fizycznych	Przerwania niemaskowalne (NMI – <i>Non-Maskable Interrupt</i>) + od 1 do 240 przerwania fizycznych
Opóźnienie obsługi przerwania	12 taktów	12 taktów
Opóźnienie między przerwaniem	6 taktów	6 taktów
Poziomy priorytetów przerwania	Od 8 do 256 poziomów priorytetów	Od 8 do 256 poziomów priorytetów
Kontroler przerwania wake-up (Wake-up Interrupt Controller)	Do 240 przerwania <i>wake-up</i>	Do 240 przerwania <i>wake-up</i>
Tryby uśpienia	Instrukcje WFI (<i>Wait for Interrupt</i>) i WFE (<i>Wait for Event</i>) oraz opcja przejścia do trybu uśpienia po wyłączeniu (<i>Sleep On Exit</i>). Sygnały trybu uśpienia i głębokiego uśpienia (<i>Sleep & Deep Sleep Signals</i>). Opcjonalny tryb zachowania (<i>Retention Mode</i>) z zestawem zarządzania energią (<i>ARM Power Management Kit</i>).	Instrukcje WFI (<i>Wait for Interrupt</i>) i WFE (<i>Wait for Event</i>) oraz opcja przejścia do trybu uśpienia po wyłączeniu (<i>Sleep On Exit</i>). Sygnały trybu uśpienia i głębokiego uśpienia (<i>Sleep & Deep Sleep Signals</i>). Opcjonalny tryb zachowania (<i>Retention Mode</i>) z zestawem zarządzania energią (<i>ARM Power Management Kit</i>).
Operacje bitowe	Zintegrowane instrukcje oraz atomowy dostęp do bitów (<i>bit banding</i>).	Zintegrowane instrukcje oraz atomowy dostęp do bitów (<i>bit banding</i>).

Tab. 2. Wyniki certyfikowanych testów CoreMark dla wybranych mikrokontrolerów STM32F i STM32F4 (za www.coremark.org)

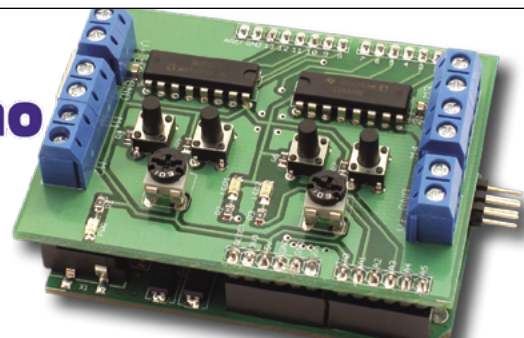
Typ	Pamięć	Kompilator	f _{cpu} [MHz]	CoreMark/MHz	CoreMark
STM32F417IGT6	SRAM	GreenHills Multi 6.1 Compiler 2012	168	2,91	488,59
STM32F417IGT6	Flash	GreenHills Multi 6.1 Compiler 2012	168	2,81	472,96
STM32F417IGT6	Flash	MDK-ARM 4.20.03.0	168	2,16	363,72
STM32F217IGT6	Flash	MDK-ARM 4.1.0.481	120	2,12	253,99
STM32F217IGT6	Flash	MDK-ARM 4.0.0.524	120	1,91	228,60

AVTduino Motor - driver silników dla Arduino

AVT1619

www.sklep.avt.pl

Więcej informacji:



Tab. 3. Liczba wait-state'ów niezbędnych podczas dostępu do pamięci Flash w mikrokontrolerach STM32F2/F4 (przy napięciu zasilającym mikrokontroler 2,7...3,6 V)

Liczba niezbędnych wait-state'ów (cykli f_{CPU})	Liczba cykli zegarowych CPU	Zakres HCLK [MHz]
0	1	0...30
1	2	30...60
2	3	60...90
3	4	90...120
4	5	120...150
5	6	150...168

dajności proporcjonalne do zmian częstotliwości taktowania, z drobnym przeskokiem „w dół” wynikającym z konieczności dodania kolejnego wait-state'u po przekroczeniu określonego przez producenta progu częstotliwości taktującej. Taki sposób szaco-

wania nie dotyczy fragmentów programów korzystających z FPU, SIMD lub instrukcji DSP, bowiem w takich przypadkach są one automatycznie implementowane przez kompilator do wykonania w odpowiednich koprocessorach sprzętowych.

Podsumowanie

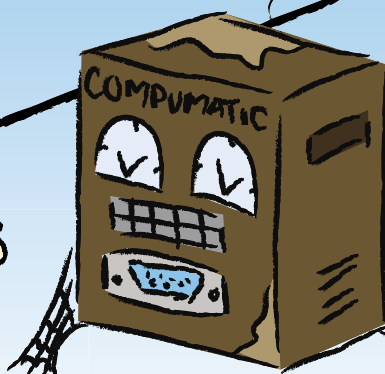
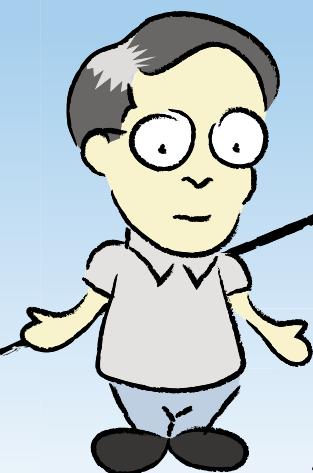
Ponieważ zagadnienia związane z porównywaniem wydajności obliczeniowej mikrokontrolerów cieszą się coraz większym zainteresowaniem – co częściowo wynika z rosnącej konkurencji pomiędzy ich producentami – w jednym z kolejnych numerów EP przedstawimy wyniki naszych testów laboratoryjnych, które pokażą realne możliwości mikrokontrolerów STM32F w kontekście wymogów współczesnych aplikacji. Wrócimy także do prób przetaktowywania CPU w tych mikrokontrolerach, bo – jak pokazują wstępne wyniki – mikrokontrolery STM32 są na nie dość podatne.

Tomasz Starak

REKLAMA

MYŚLISZ, ŻE MOŻEMY WPIĄĆ TO STARE PUDŁO DO SIECI?

NO PEWNIEM, WIDZĘ, ŻE MA PORT SZEREGOWY.



UDOSTĘPNIJ W SIECI PORTY SZEREGOWE

Serwery portów szeregowych do sieci Ethernet

245 zł



serwer 1 portu szeregowego

495 zł



serwer 2 portów szeregowych

525 zł



przemysłowy serwer 1 portu szeregowego

995 zł



przemysłowy serwer 4 portów szeregowych