

Mikrokontrolery 8- i 16-bitowe zalecane do nowych aplikacji (1)

Mimo inwazji rdzeni 32-bitowych i oczywistych zalet ich stosowania, mikrokontrolery z rdzeniami 8-bitowymi i 16-bitowymi są nadal chętnie stosowane. Jako najważniejszy powód ich użycia wskazałbym to, że bazują na rdzeniach sprawdzonych przez wiele lat użytkowania, więc można się spodziewać, że mają wyeliminowane większość błędów, a firmy – producenci narzędzi, również poradziły sobie z niedomaganiem np. kompilatorów i bibliotek funkcji. Ich ogromną zaletą jest również łatwość tworzenia oprogramowania. Należy również dostrzec fakt, że współcześnie „małe” i tanie mikrokontrolery są uzbrajane w moduły funkcjonalne, które niegdyś były zarezerwowane wyłącznie dla droższych układów, co rozszerza obszar ich zastosowań.

We współczesnym świecie trendem obowiązującym konstruktorów większości urządzeń elektronicznych jest oszczędność energii. W wielu wypadkach jest to szeroko rozumiany obszar związany nie tylko z użytkowaniem urządzenia, ale również ze zużyciem surowców i energii przeznaczonej na jego wyprodukowanie. Skupmy się jednak na zagadnieniach związanych z samym mikrokontrolerem i jego najbliższym otoczeniem.

Takie działania są z jednej strony wymuszane przez regulacje prawne, a z drugiej przez samych użytkowników, którzy żądają możliwie najdłuższego czasu pracy urządzenia przenośnego po naładowaniu jego akumulatora lub wymianie baterii. Energooszczędność najczęściej wymaga takiego trybu pracy mikrokontrolera, w którym rdzeń i możliwie jak najwięcej układów peryferyjnych przebywają w trybie uśpienia/obniżonego poboru energii i są wybudzane jedynie na czas realizacji ważnych zadań.

W większości współczesnych mikrokontrolerów moduły peryferyjne, takie jak interfejsy komunikacyjne czy kontroler DMA, pracują niezależnie od CPU i są w stanie samodzielnie realizować pewne zadania bez jego włączenia. Z punktu widzenia energooszczędności jest bardzo istotne, aby były one wykonywane w jak najkrótszym czasie. Wówczas średni prąd zasilania pobierany przez urządzenie jest niewielki, co przekłada się na dłuższy czas funkcjonowania przy zasilaniu baterijnym. Dlatego wydajność rdzenia jest powodem, dla którego mikrokontrolery 32-bitowe tak świetnie radzą sobie we

współczesnych urządzeniach, praktycznie wypierając inne rozwiązania. Dziś nikogo już nie dziwi fakt, że 32-bitowy rdzeń może mieć obudowę 16-nóżkową i zajmować powierzchnię około 2 milimetrów kwadratowych.

Należę do pokolenia osób, które swoje pierwsze programy tworzyły dla mikrokontrolerów 8048, 8049 i 8051. Wtedy głównie używaliśmy assemblera. Wydaje mi się, że podobnie jak wielu konstruktorów rozpoczynających karierę w tamtych latach, muszę walczyć z samym sobą sięgając po rdzeń 32-bitowy do realizacji prostych czynności, ponieważ jestem obciążony tamtym sposobem myślenia, pochodzącym z czasów, w których do 32-bitowego rdzenia sięgało się przy realizacji bardzo wymagających zadań, ponieważ było to związane z ogromnymi kosztami gotowego systemu. Dziś jest inaczej – mikrokontrolery i mikroprocesory 32-bitowe są tanie, i wszechobecne.

Pomimo bezdyskusyjnych zalet rdzeni 32-bitowych mają one jednak wady. Elastyczność jest okupiona trudnością konfigurowania samego rdzenia i modułów peryferyjnych. Owszem, są do tego odpowiednie narzędzia, które nie tylko podpowiadają, ale nawet generują odpowiedni kod w języku C, jednak pomimo pewnego ryzyka takiego stwierdzenia (a to przez ogromny wybór narzędzi do tworzenia programów) trzeba powiedzieć, że programowanie mikrokontrolerów 32-bitowych jest trudniejsze albo nawet znacznie trudniejsze, niż ich 8-bitowych. Wiem, że takie twierdzenie może wywołać

spzeciwi, ponieważ to zależy od konkretnej aplikacji, ale założmy, że w tym momencie mam na myśli mikrokontrolerowe „Hello world!”, pierwszy program, którego zadaniem jest migotanie diodą LED dołączoną do wyprowadzenia. W 8-bitowcu nie musimy używać kompilatora C, biblioteki, samo oprogramowanie nawet w assemblerze zajmie co najwyżej kilka lub kilkanaście linii kodu. Dlatego w niektórych zastosowaniach, mimo wszystko, warto pamiętać o starszych rdzeniach zwłaszcza, że producenci poprawiają ich wydajność, zmieniają technologie produkcji na bardziej energooszczędne oraz otaczają rdzeń ciekawymi modułami peryferyjnymi.

O ile 8-bitowiec nie bardzo radzi sobie z obsługą multimediiów czy skomplikowanego wyświetlacza LCD (aczkolwiek nie jest to regułą), o tyle w takim zastosowaniu przyda się rdzeń 16-bitowy. W wielu aplikacjach może też przydać się możliwość zasilania rdzenia napięciem 5 V, co może poprawić odporność na zaburzenia elektromagnetyczne. Taka właściwość jest potrzebna w urządzeniach przeznaczonych dla motoryzacji lub medycyny.

Ogromną zaletą starszych mikrokontrolerów jest ich długa obecność na rynku. Dzięki niej stały się dobrze znane oraz znacznie wzrosła liczba narzędzi, w tym darmowych, stosowanych nie tylko przez profesjonalistów, ale i amatorów. Same mikrokontrolery też miały czas na to, aby zostać sprawdzone od A do Z i udoskonalone przez producentów. Nie ulegajmy powszechnej „armizacji”, warto przemyśleć alternatywy.



Atmel – mikrokontrolery z rdzeniem AVR

Osobiście kojarzę firmę Atmel przede wszystkim z mikrokontrolerami AVR. Niegdyś używałem atmelowskich odpo-

WYBÓR KONSTRUKTORA

Tabela 1. Aktualna oferta mikrokontrolerów ATtiny (rodzina AVRtiny)				
Typ układu	RAM	Flash	Najważniejsze parametry	Obudowa
ATtiny4	32 B	512 B	12 MHz, 4×I/O, 4×zewn. IRQ, 16-bit T/C+2×PWM, WDG	SOT-23/6, UDFN 8 pin
ATtiny5	32 B	512 B	12 MHz, 4×I/O, 4×zewn. IRQ, 16-bit T/C+2×PWM, WDG, 4×8-bit A/C	SOT-23/6, UDFN 8 pin
ATtiny9	32 B	1 kB	12 MHz, 4×I/O, 4×zewn. IRQ, 16-bit T/C+2×PWM, WDG	SOT-23/6, UDFN 8 pin
ATtiny10	32 B	1 kB	12 MHz, 4×I/O, 4×zewn. IRQ, 16-bit T/C+2×PWM, WDG, 4×8-bit A/C	SOT-23/6, UDFN 8 pin
ATtiny13	64 B	1 kB	20 MHz, 6×I/O, 6×zewn. IRQ, 8-bit T/C+2×PWM, 4×10-bit A/C, WDG, komparator analogowy	SOIC-8, MLF-20
ATtiny13A	64	1 kB	picoPower, 20 MHz, 6×I/O, 6×zewn. IRQ, 8-bit T/C+2×PWM, 4×10-bit A/C, WDG, komparator analogowy	SOIC-8, PDIP-8, MLF-10, MLF-20
ATtiny20	128 B	2 kB	12 MHz, 12×I/O, 12×zewn. IRQ, 8-bit T/C+2×PWM, 16-bit T/C+2×PWM, 8×10-bit A/C, WDG, TWI, SPI, komparator analogowy	SOIC-14, WLCSP-12, UFBGA-15, VQFN-20
ATtiny24	128 B	2 kB	20 MHz, 12×I/O, 12×zewn. IRQ, 8-bit T/C+2×PWM, 16-bit T/C+2×PWM, 8×10-bit A/C, WDG, USI, komparator analogowy, EEPROM 128 B	SOIC-14, PDIP-14, VQFN-20, MLF-20
ATtiny24A	128 B	2 kB	picoPower, 20 MHz, 12×I/O, 12×zewn. IRQ, 8-bit T/C+2×PWM, 16-bit T/C+2×PWM, 8×10-bit A/C, WDG, USI, komparator analogowy, EEPROM 128 B	SOIC-14, PDIP-14, QFN-20, VQFN-20, MLF-20
ATtiny25	128 B	2 kB	20 MHz, 6×I/O, 6×zewn. IRQ, 8-bit T/C+2×PWM, 8-bit HS T/C+2×PWM, 4×10-bit A/C, WDG, USI, komparator analogowy, EEPROM 128 B	PDIP-8, SOIC-8, QFN-20, MLF-20
ATtiny26	128 B	2 kB	16 MHz, 16×I/O, 11×zewn. IRQ, 8-bit T/C+2×PWM, 8-bit HS T/C+2×PWM, 11×10-bit A/C, WDG, USI, komparator analogowy, EEPROM 128 B	PDIP-20, SOIC-20, QFN-32, MLF-32
ATtiny28L	256 B	2 kB	4 MHz, 11×I/O, 10×zewn. IRQ, 8-bit T/C, driver LED, WDG, komparator analogowy	PDIP-28, TQFP-32, MLF-32
ATtiny40	256 B	4 kB	12 MHz, 18×I/O, 18×zewn. IRQ, QTouch, 8-bit T/C+2×PWM, 8/16-bit T/C, 12×10-bit A/C, SPI, TWI, WDG, komparator analogowy	SOIC-20, TSSOP-20, VQFN-20
ATtiny43U	256 B	4 kB	8 MHz, 16×I/O, 16×zewn. IRQ, 2×8-bit T/C+2×PWM, 4×10-bit A/C, WDG, USI, komparator analogowy, konwerter boost, EEPROM 64 B	SOIC-20, QFN-20, MLF-20
ATtiny44	256 B	4 kB	20 MHz, 12×I/O, 12×zewn. IRQ, 8-bit T/C+2×PWM, 16-bit T/C+2×PWM, 8×10-bit A/C, WDG, USI, komparator analogowy, EEPROM 256 B	SOIC-14, PDIP-14, QFN-20, VQFN-20, MLF-20
ATtiny44A	256 B	4 kB	picoPower, 20 MHz, 12×I/O, 12× zewn. IRQ, 8-bit T/C+2×PWM, 16-bit T/C+2×PWM, 8×10-bit A/C, WDG, USI, komparator analogowy, EEPROM 256 B	SOIC-14, PDIP-14, QFN-20, VQFN-20, MLF-20
ATtiny45	256 B	4 kB	20 MHz, 6×I/O, 6×zewn. IRQ, 8-bit T/C+2×PWM, 8-bit HS T/C+2×PWM, 4×10-bit A/C, WDG, USI, komparator analogowy, EEPROM 256 B	PDIP-8, SOIC-8, QFN-20, MLF-20
ATtiny48	256 B	4 kB	12 MHz, 28×I/O, 28×zewn. IRQ, 8-bit T/C, 16-bit T/C, 6×10-bit A/C, WDG, SPI, TWI, komparator analogowy, EEPROM 64 B,	PDIP-28, QFN-28, TQFP-32, QFN-32, UFBGA-32
ATtiny84	512 B	8 kB	20 MHz, 12×I/O, 12× zewn. IRQ, 8-bit T/C+2×PWM, 16-bit T/C+2×PWM, 8×10-bit A/C, WDG, USI, komparator analogowy, EEPROM 512 B	SOIC-14, PDIP-14, QFN-20, VQFN-20, MLF-20
ATtiny84A	512 B	8 kB	picoPower, 20 MHz, 12×I/O, 12× zewn. IRQ, 8-bit T/C+2×PWM, 16-bit T/C+2×PWM, 8×10-bit A/C, WDG, USI, komparator analogowy, EEPROM 512 B	SOIC-14, PDIP-14, QFN-20, VQFN-20, MLF-20
ATtiny85	512 B	8 kB	20 MHz, 6×I/O, 6×zewn. IRQ, 8-bit T/C+2×PWM, 8-bit HS T/C+2×PWM, 4×10-bit A/C, WDG, USI, komparator analogowy, EEPROM 512 B	PDIP-8, SOIC-8, QFN-20, MLF-20
ATtiny87	512 B	8 kB	16 MHz, 16×I/O, 16×zewn. IRQ, 8-bit T/C, 8-bit HS T/C, 11×10-bit A/C, WDG, USI, LIN, komparator analogowy, EEPROM 512 B	SOIC-20, VQFN-32, TSSOP-20
ATtiny88	512 B	8 kB	12 MHz, 28×I/O, 28×zewn. IRQ, 8-bit T/C, 16-bit T/C, 8×10-bit A/C, WDG, SPI, TWI, komparator analogowy, EEPROM 64 B,	PDIP-28, QFN-28, TQFP-32, QFN-32, UFBGA-32
ATtiny167	512	16 kB	16 MHz, 16×I/O, 16×zewn. IRQ, 8-bit T/C, 8-bit HS T/C, 11×10-bit A/C, WDG, USI, LIN, komparator analogowy, EEPROM 512 B	SOIC-20, VQFN-32, TSSOP-20
ATtiny261A	128 B	2 kB	picoPower, 20 MHz, 16×I/O, 16×zewn. IRQ, 8-bit TC, 8-bit HS T/C+3×PWM, 11×10-bit A/C, WDG, USI, komparator analogowy, EEPROM 128 B	PDIP-20, SOIC-20, MLF-32
ATtiny441	128 B	4 kB	20 MHz, 12×I/O, 12×zewn. IRQ, 8-bit TC+2×PWM, 16-bit HS T/C+2×PWM, 12×10-bit A/C, WDG, SPI, TWI, 2×USART, 2×komparator analogowy, EEPROM 256 B	SOIC-14, MLF-20, QFN-20, VQFN-20
ATtiny461	256 B	4 kB	20 MHz, 16×I/O, 16×zewn. IRQ, 8-bit TC, 8-bit HS T/C+PWM, 11×10-bit A/C, WDG, USI, komparator analogowy, EEPROM 256 B	PDIP-20, SOIC-20, MLF-32
ATtiny461A	256 B	2 kB	picoPower, 20 MHz, 16×I/O, 16×zewn. IRQ, 8-bit TC, 8-bit HS T/C+3×PWM, 11×10-bit A/C, WDG, USI, komparator analogowy, EEPROM 256 B	PDIP-20, SOIC-20, MLF-32
ATtiny828	512 B	8 kB	20 MHz, 28×I/O, 28×zewn. IRQ, 8-bit T/C+2×PWM, 16-bit T/C+2×PWM, 28×10-bit A/C, USART, SPI, TWI, WDG, komparator analogowy, EEPROM 256 B	TQFP-32, QFN-32, MLF-32

Tabela 1. c.d.				
Typ układu	RAM	Flash	Najważniejsze parametry	Obudowa
ATtiny841	256 B	8 kB	16 MHz, 12×I/O, 12×zewn. IRQ, 8-bit TC+2×PWM, 16-bit HS T/C+2×PWM, 12×10-bit A/C, WDG, SPI, TWI, 2×USART, 2×komparator analogowy, EEPROM 256 B	SOIC-14, MLF-20, QFN-20, VQFN-20
ATtiny861	512 B	8 kB	20 MHz, 16×I/O, 16×zewn. IRQ, 8-bit TC, 8-bit HS T/C+3×PWM, 11×10-bit A/C, WDG, USI, komparator analogowy, EEPROM 512 B	PDIP-20, SOIC-20, MLF-32
ATtiny861A	512 B	8 kB	picoPower, 20 MHz, 16×I/O, 16×zewn. IRQ, 8-bit TC, 8-bit HS T/C+3×PWM, 11×10-bit A/C, WDG, USI, komparator analogowy, EEPROM 512 B	PDIP-20, SOIC-20, MLF-32
ATtiny1634	1 kB	16 kB	12 MHz, 18×I/O, 18×zewn. IRQ, 8-bit T/C+2×PWM, 16-bit T/C+2×PWM, 12×10-bit A/C, WDG, TWI, SPI, 2×USART, komparator analogowy, EEPROM 256 B	QFN-20, MLF-20, SOIC-20
ATtiny2313	128 B	2 kB	20 MHz, 18×I/O, 18×zewn. IRQ, 8-bit T/C, 16-bit T/C, 4×PWM, USI, USART, komparator analogowy, EEPROM 128 B	PDIP-20, SOIC-20, QFN-20, MLF-20
ATtiny2313A	128 B	2 kB	picoPower, 20 MHz, 18×I/O, 18×zewn. IRQ, 8-bit T/C, 16-bit T/C, 4×PWM, USI, USART, komparator analogowy, EEPROM 128 B	PDIP-20, SOIC-20, VQFN-20, MLF-20
ATtiny4313	256 B	4 kB	picoPower, 20 MHz, 18×I/O, 18×zewn. IRQ, 8-bit T/C, 16-bit T/C, 4×PWM, USI, USART, komparator analogowy, EEPROM 256 B	PDIP-20, SOIC-20, VQFN-20, MLF-20

wiedników 8052 (nadal są one oferowane przez firmę), ale w mojej świadomości to przede wszystkim producent i wynalazca 8-bitowych rdzeni AVR i mikrokontrolerów w niego wyposażonych. Nie wolno jednak zapominać, że firma oferuje również 32-bitowe wersje tego rdzenia tj. AT32UC3 oraz AT32AP7. Mimo tego 8-bitowce w ofercie Atmela mają się dobrze, a firma zapowiada wprowadzenie do oferty kolejnych mikrokontrolerów z tej grupy.

Mikrokontrolery AVR z rdzeniem 8-bitowym podzielono na trzy rodziny: *tinyAVR* (ATtiny), *megaAVR* (ATmega) oraz *AVR Xmega* (ATxmega).

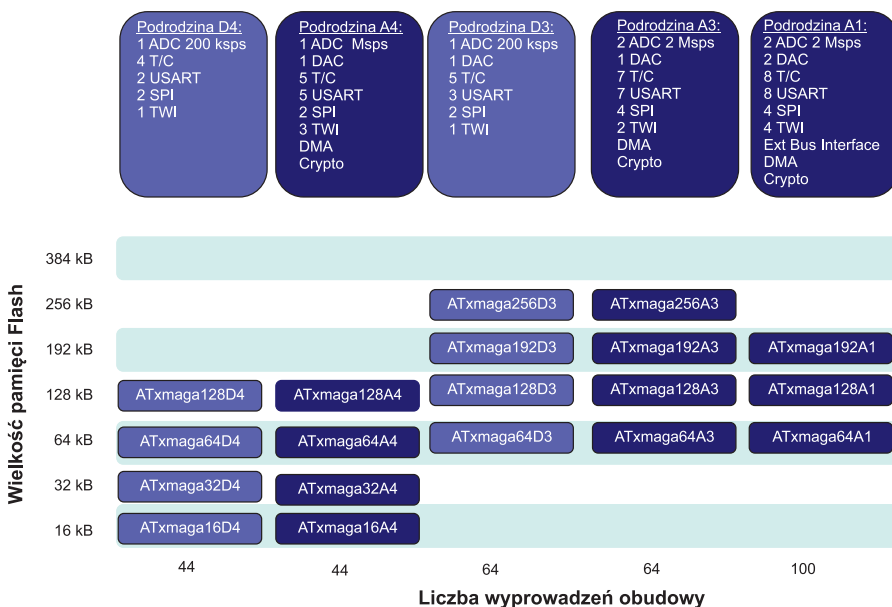
Pierwsza grupa obejmuje tanie układy o podstawowej funkcjonalności, wyposażone w raczej niewielką pamięć Flash (0,512 B...16 kB), pamięć RAM mieszczącą – zależnie od układu – do 4 kB danych (32 B...4 kB), oferowane w obudowach mających od 6 do 32 wyprowadzeń. Aktualnie rodzina ATtiny obejmuje około 36 układów o różnych możliwościach. Ich wykaz umieszczono w tabeli 1.

Mikrokontrolery z tej rodziny mają tę samą architekturę, co pozostałe AVR-y i są z nimi kompatybilne programowo, oczywiście w ramach swoich możliwości i zasobów sprzętowych. Zoptymalizowano je pod kątem aplikacji niewymagających dużej mocy obliczeniowej, w których krytyczny jest koszt opracowania i produkcji. Dzięki miniaturyrowym obudowom (najmniejszy układ z tej rodziny – ATtiny4 – ma obudowę o wymiarach 1,5 mm×1,4 mm) zajmują one mało miejsca na płytce drukowanej, co ma duże znaczenie przy produkcji wielkoseryjnej np. zabawek, a niska cena sprawia, że układów można też używać w roli tzw. logiki doklejanej (*glue logic*), to znaczy do dodawania funkcjonalności lub realizowania funkcji normalnie zapewnianych przez dodatkowe obwody logiczne. Jeszcze innym ciekawym zastosowaniem jest budowanie procesorów zdecentralizowanych, składających się z wielu mikrokontrolerów autonomicznie podejmujących decyzje i wymieniających się informacjami o stanie urzędzenia.

Poszczególne układy mikrokontrolerów w obrębie rodziny tinyAVR różnią się wyposażeniem – od mających podstawowe peryferia, po „uzbrojone” w pamięć EEPROM, przetwornik A/C, generator PWM i interfejs USI, który może być używany jako SPI, UART lub kompatybilny z I²C – TWI. Mimo niewielkich obudów mikrokontrolery z tej rodziny można debugować i programować w aplikacji docelowej, co ma ogromne znaczenie na etapie uruchamiania prototypu. Ciekawostką w tej rodzinie jest ATtiny43U, który dzięki wbudowanej przetwornicy może pracować przy zasilaniu napięcie zaledwie 0,7 V. Kompatybilność z innymi AVR-ami zapewnia możliwość użycia bibliotek programowych – mam tu na myśli głównie QTouch służącej do łatwego wykonywania interfejsu dotykowego. Uwaga: nie wszystkie ATtiny mogą obsługiwać panele dotykowe.

Kolejną grupę 8-bitowców Atmela tworzą układy z rodziny megaAVR. W momencie tworzenia tego artykułu, ta rodzina mikrokontrolerów obejmuje 80 układów, których raczej nie będziemy tu wymieniali. Inaczej niż tinyAVR, „megi” znacznie różnią się wyposażeniem w moduły peryferyjne, wielkością pamięci oraz liczbą wyprowadzeń, ale wszystkie mają ten sam 8-bitowy, znany rdzeń AVR. Maksymalna prędkość przetwarzania danych wynosi do 20 MIPS. Liczba wyprowadzeń I/O rozciąga się od 20 do 69, pojemność pamięci Flash od 4 (ATmega48) do 256 kB (ATmega2560 i 2561). Niektóre „megi” doczekały się swoich wariantów w technologii picoPower, co korzystnie wpłynęło na zmniejszenie poboru energii. Każda ATmega może być programowana w aplikacji oraz aktualizować swoje oprogramowanie za pomocą funkcji wbudowanego *bootloader'a*, bez konieczności wyłączenia urządzenia czy sprzętowego wprowadzania mikrokontrolera w jakiś specjalny tryb pracy.

W ramach rodziny megaAVR konstruktor ma największy wybór układów w znaczeniu wielkości pamięci, liczby wyprowadzeń I/O oraz wbudowanych bloków funkcjonal-



Rysunek 1. Podstawowe parametry mikrokontrolerów z rodziny AVR Xmega

nych. Można wybierać od nieskomplikowanych układów, tanich i wyposażonych tylko w podstawowe peryferia, do specjalizowanych, mających np. liczne generatory PWM, sprzętowy interfejs USB, CAN, LIN, kontroler wyświetlacza LCD i inne. Co ważne, te mikrokontrolery są oferowane od wielu lat, więc można się spodziewać dostępności zarówno wiedzy, jak i narzędzi, a wiele z programatorów i kompilatorów jest dostępnych za darmo. Nie bez znaczenia jest też fakt, że producent zapowiada dalszy rozwój tej rodziny nie tyle przez wprowadzanie nowych układów, ile przez zmianę technologii ich wytwarzania, co w zamiarach ma obniżyć cenę mikrokontrolera oraz zmniejszyć pobór energii.

Trzecią grupą mikrokontrolerów z oferty Atmela jest nowoczesna, niedawno wprowadzona do sprzedaży, rodzina mikrokontrolerów AVR Xmega. Aktualnie obejmuje ona 46 układów. Podstawowe parametry wybranych układów pozwalające na zorientowanie się w ofercie mikrokontrolerów wymieniono w tabeli 2, natomiast w szybkiej orientacji będzie pomocny rysunek 1.

Nowe mikrokontrolery uzbrojono w funkcjonalności dotychczas rzadko spo-

tykane w świecie rdzeni 8-bitowych. Wydaje się, że producent miał na celu przede wszystkim obniżenie poboru energii dzięki nieangażowaniu CPU w mniej wymagające zadania, takie jak na przykład transmisja danych pomiędzy blokami funkcjonalnymi.

Bodaj najważniejszą innowacją w Xmega jest system obsługi zdarzeń, który umożliwia współpracę bloków funkcjonalnych bez udziału CPU, przy jednoczesnym zachowaniu gwarantowanego czasu reakcji pomiędzy wystąpieniem zdarzenia o uruchomieniem procedury jego obsługi. Ta funkcja świetnie nadaje się do użycia w aplikacjach, w których kluczowe znaczenie ma zapewnienie czasu reakcji układu w czasie nie dłuższym, niż wymagany np. w automatyce przemysłowej lub motoryzacji. Ponadto, dzięki pracy samych modułów peryferyjnych i niezależaniu CPU, nowa funkcjonalność może znacząco ograniczyć zużycie energii. Praca systemu obsługi zdarzeń jest wspierana przez kontroler DMA, który umożliwia przesyłanie danych pomiędzy blokami mikrokontrolera bez angażowania CPU.

Niektóre układy Xmega (tabela 2) mają moduł kryptograficzny wspierający 64-bitowy algorytm szyfrowania DES oraz 128-bitowy

wy AES. Dzięki temu, nowe mikrokontrolery świetnie nadają się do różnych aplikacji sieciowych i telekomunikacyjnych, w których jest konieczne szyfrowanie danych zwłaszcza w tzw. sieciach otwartych lub przy korzystaniu z urządzeń dołączanych drogą radiową. W nowej rodzinie układów starano się wyeliminować wady megaAVR, z których jedna była związana z kontrolerem przerwań, niedającym możliwości ustalenia priorytetów ich obsługi. Nowy, wielopoziomowy kontroler przerwań (PMIC) pozwala na nadawanie przerwaniom jednego z trzech priorytetów – obsługa przerwań o priorytecie wyższym odbywa się przed tą o niższym.

W rodzinie ATXmega zastosowano drugą generację technologii picoPower. Jedną z ciekawszych jej cech jest możliwość zasilania układu napięciem 1,8 V. Uzyskano też znacznie ograniczenie poboru prądu – nowe układy potrzebują zaledwie 100 nA dla podtrzymania zawartości pamięci RAM w trybie *Power Down* oraz 550 nA dla zasilania tejże pamięci oraz zegara RTC w trybie *Power Save*. Przejście do trybu aktywnego następuje w czasie 5 mikrosekund.

Programiście (lub projektantowi systemu) oddano do dyspozycji rozbudowa-

Tabela 2. Podstawowe parametry wybranych mikrokontrolerów z rodziny AVR Xmega

Typ układu	RAM	Flash	Najważniejsze parametry	Obudowa
ATXmega64A1U	4 kB	64 kB	32 MHz, 78×I/O, 78×zewn. IRQ, 16-bit T/C, RTC, USB, SPI, TWI, USART, I2C, 4-kan. DMA, 8-kan. ES*, AES/DES, EBI**, A/C, C/A, EEPROM 2 kB...	TQFP-100, VFBGA-100, CBGA-100
ATXmega128A1U	8 kB	128 kB		
ATXmega64A3U	4 kB	64 kB	32 MHz, 50×I/O, 50×zewn. IRQ, A/C, C/A, USART, SPI, TWI, 4-kan. DMA, 8-kan. ES*, AES/DES, CRC16/CRC32, RTC, EEPROM 4 kB...	TQFP-64, QFN-64
ATXmega128A3U	8 kB	128 kB		
ATXmega192A3U	16 kB	192 kB		
ATXmega256A3U	16 kB	256 kB		
ATXmega256A3BU	16 kB	256 kB	32 MHz, 47×I/O, 49×zewn. IRQ, 16-bit T/C, A/C, C/A, USB, 4×komparator analogowy, AES/DES, 6×USART, 2×SPI, 2×TWI, RTC, 4-kan. DMA, 8-kan. ES*, EEPROM 4 kB...	TQFP-64, MLF-64
ATXmega16A4U	3,3 kB	16 kB	32 MHz, 34×I/O, 34×zewn. IRQ, 5×16-bit T/C, SPI, TWI, AES/DES, RTC, A/C, C/A, 4-kan. DMA, 8-kan. ES*, EEPROM 1 kB...	TQFP-44, VQFN-44, QFN-44, VFBGA-49
ATXmega32A4U	4 kB	32 kB		
ATXmega64A4U	4 kB	64 kB		
ATXmega128A4U	8 kB	128 kB		
ATXmega64A1	4 kB	64 kB	32 MHz, 78×I/O, 78×zewn. IRQ, 4-kan. DMA, 8-kan. ES*, EBI**, 8×USART, TWI, SPI, RTC, A/C, C/A, 4×komparator analogowy, EEPROM 2 kB...	TQFP-100, BGA-100, VFBGA-100
ATXmega128A1	8 kB	128 kB		
ATXmega64A3	4 kB	64 kB	32 MHz, 50×I/O, 50×zewn. IRQ, 4-kan. DMA, 8-kan. ES*, 7×16-bit T/C, TWI, SPI, RTC, A/C, C/A, PMIC, EEPROM 2 kB...	TQFP-64, QFN-64
ATXmega128A3	8 kB	128 kB		
ATXmega192A3	16 kB	192 kB		
ATXmega256A3	16 kB	256 kB		
ATXmega256A3B	16 kB	256 kB	32 MHz, 47×I/O, 49×zewn. IRQ, 4-kan. DMA, 8-kan. ES*, 7×16-bit T/C, SPI, TWI, RTC, 6×USART, AES/DES, A/C, C/A, EEPROM 4 kB...	TQFP-64, QFN-64, MLF-64
ATXmega16A4	4 kB	16 kB	Uwaga: niezalecane do nowych urządzeń. Zamiast A4 należy stosować A4U	TQFP-44, QFN-44, VQFN-44, VFBGA-49
ATXmega32A4	4 kB	32 kB		
ATXmega64A4	4 kB	64 kB		
ATXmega128A4	4 kB	128 kB		
ATXmega384C3	32 kB	384 kB	32 MHz, 50×I/O, 50×zewn. IRQ, PMIC, 2-kan. DMA, 4-kan. ES*, 5×16-bit T/C, RTC, USB, 3×USART, PWM, A/C, 2×komparator analogowy, TWI, SPI, AES, EEPROM 4 kB...	TQFP-64, QFN-64
ATXmega384D3	32 kB	384 kB	32 MHz, 50×I/O, 50×zewn. IRQ, PMIC, CRC16/CRC32, 4-kan. ES*, 5×16-bit T/C, PWM, RTC, USB, 3×USART, PWM, A/C, 2×komparator analogowy, TWI, SPI, AES, EEPROM 4 kB...	TQFP-64, QFN-64

* ES – Event System, ** – zewnętrzna szyna danych

ny blok generatora sygnałów zegarowych, dzięki któremu Xmega może korzystać z rezonatorów kwarcowych i ceramicznych, wbudowanego generatora RC oraz pętli PLL, oscylatorów zewnętrznych i wbudowanego preskalera częstotliwości. Do wymienionych cech dodajmy bogate wyposażenie w interfejsy komunikacyjne (USART, SPI, EBI, TWI), przetworniki C/A i A/C, kontrolery LCD (do 160 segmentów), programowane obwody logiczne tzw. *Xmega Custom Logic*, możliwość podtrzymania baterijnego zasilania RTC, wsparcie dla obsługi USB, a otrzymany pełny obraz innowacji wprowadzonych przez Atmel. Wydaje się to o tyle dziwne, że jednocześnie ten producent uległ „armizacji” i wytwarza mikrokontrolery z rdzeniem ARM. Widać jednak, że zapotrzebowanie na 8-bitowce jest tak duże, że nie tylko powstają nowe rodziny, ale jest zapowiadany ich rozwój i wprowadzanie nowych rozwiązań.



Freescale – kontynuacja tradycji

Myślę, że dzięki swoim posunięciom w zakresie promowania mikrokontrolerów Kinetis firma Freescale jest kojarzona przede wszystkim z mikrokontrolerami 32-bitowymi z rdzeniem ARM. Nic bardziej mylącego, ponieważ Freescale oferuje też natywne rozwiązania 32-bitowe, tj. procesory Coldfire, Qorivva oraz 16- i 32-bitowe procesory DSP. Trzeba też pamiętać, że w przeszłości ta firma przejęła półprzewodnikową część Motoroli, która wytwarzała procesory 8- i 16-bitowe oraz inne komponenty (np. doskonałe tranzystory MOS) i była jednym z ważniejszych dostawców tych produktów dla sektora motoryzacyjnego i telekomunikacyjnego. Siłą inercji, jeśli był rynek na produkty Motoroli,



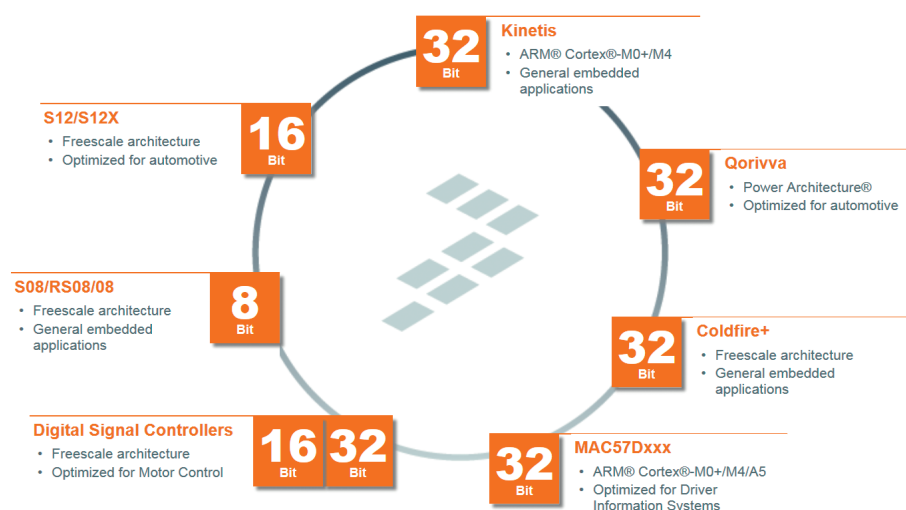
to intuicyjnie można wyczuć, że Freescale na pewno kontynuuje ich sprzedaż i produkcję. W zorientowaniu się w portfolio procesorów wytwarzanych przez Freescale pomocny będzie **rysunek 2**.

W **tabeli 3** umieszczono wykaz mikrokontrolerów Freescale z rdzeniem 8-bitowym. Patrząc na kolumnę „Rdzeń” można zauważyć, że przypuszczenia odnośnie do kontynuowania schedy po Motoroli mają podstawę, ponieważ rdzenie 8-bitowe wykorzystywane przez Freescale to HCS08 i RS08 wzorowane na HC08. Jak przekonamy się dalej, w mikrokontrolerach 16-bitowych jest używany rdzeń HCS12 (i jego warianty) będący zmodyfikowaną wersją HC12, który z kolei swoje korzenie wywodzi z HC11.

Jak wspomniano, stosowany przez Freescale rdzeń HCS08 jest wzorowany na HC08. W najnowszych rozwiązaniach może być on taktowany przebiegiem o częstotliwości do 32 MHz. Rdzeń HCS08 korzysta z tego samego zestawu instrukcji, co rdzeń HC08, a ponadto obsługuje tryb debugowania (instrukcja BGND) do transmisji

danych w tle. Pozwala to na pełne korzystanie z możliwości znajdującego się w mikrokontrolerze debuggera BDG oraz emulatora „w układzie” ICE. Kontroler debuggera obsługuje w tle podstawowej aplikacji rozkazy umożliwiające zapisywanie i odczytywanie danych z pamięci mikrokontrolera, bez przerywania pracy CPU. Emulator „w układzie” zawiera moduły przerwań i śledzenia umożliwiające wstawianie punktów przerwań, opcji wyzwalania oraz wychwytywanie zmiany w schemacie programu i jego wykonywaniu. Obsługa wbudowanego debuggera odbywa się za pomocą jednej linii I/O – BKGD. Po zakończeniu debugowania to wyprowadzenie może być używane na potrzeby aplikacji w roli standardowej linii I/O.

Aby spełnić wymagania obniżonego poboru energii w mikrokontrolerach z rdzeniem HCS08 zaimplementowano specjalny tryb oczekiwania (*Wait*) i trzy dodatkowe tryby zatrzymania (*Stop*). Wewnętrzny moduł generatora sygnału zegarowego (ICS) zapewnia możliwość wyboru kilku źródeł sygnału taktującego. Zawiera on pętlę częstotliwościową – FLL, wewnętrzny zegar odniesienia – IRC, generator współpracujący z oscylatorem zewnętrznym – XOSC oraz układ logiczny do wyboru sygnału zegarowego – CSL. Pętla częstotliwościowa powiela częstotliwość 512 razy eliminując potrzebę stosowania dodatkowych elementów zewnętrznych wymaganych dla pętli fazowych PLL. Regulowany w zakresie 31,25...39,06 kHz wewnętrzny zegar odniesienia IRC może być wykorzystywany jako odniesienie dla FLL lub bezpośrednio jako źródło sygnału zegarowego magistrali i CPU jednostki centralnej. Generator sygnału odniesienia współpracujący z zewnętrznym oscylatorem można skonfigurować do pracy w trzech trybach. Pierwszy z nich, LF, jest przeznaczony do współpracy z rezonatorami o częstotliwości 32...38,4 kHz. Trybu HF można używać w połączeniu z rezonatorami o częstotliwo-



Rysunek 2. Mikrokontrolery wytwarzane przez Freescale

ści 1...16 MHz. Trzeci tryb oscylatora przeznaczono do użytku z zewnętrznym, aktywnym źródłem zegarowym o częstotliwości do 32 MHz. Układ logiczny selekcji sygnału zegarowego (SCL) wybiera i rozdziela sygnał z poszczególnych źródeł zegarowych na jednostkę centralną i dodatkowe bloki mikrokontrolera. Do sterowania jednostką centralną i magistralą można używać źródeł FLL, IRC i XOSC. Częstotliwość sygnału wyjściowego może być w danym czasie dzielona przez 2, 4 lub 8, aby zmniejszyć szybkość wykonywania instrukcji przez CPU i urządzenia peryferyjne dla przedłużenia czasu życia baterii, bez przełączania mikrokontrolera do pracy w tryb oszczędnościowy.

Zawartość pamięci Flash i RAM jest chroniona przez specjalny mechanizm ochrony odznaczający się wysokim poziomem bezpieczeństwa, ponieważ dostęp do tych pamięci można uzyskać jedynie po wprowadzeniu 8-bajtowego kodu. Można też opcjonalnie zabezpieczyć przed przypadkowym zapisem lub skasowaniem ważne „nieulotne” dane i kod, takie jak stałe aplikacyjne oraz *bootloader*.

W rodzinie mikrokontrolerów z rdzeniem HCS08 są dostępne dwa niezależne moduły czasowe (timery). Pierwszy z nich to 2-kanalowy, programowany, 16-bitowy moduł Timer/PWM (TPM). Każdy z kanałów tego modułu można konfigurować niezależnie, co pozwala np. na zliczanie impulsów zewnętrznych, porównywanie przebiegów i generowanie impulsów PWM z przesunięciem do środka lub do krawędzi. Obsługa rejestrów timerów jest prowadzona w taki sposób, aby uzyskać zwarte, 16-bitowe uaktualnianie współczynnika wypełnienia i uniknąć powstawania impulsów o przypadkowej szerokości. Drugi timer jest 8-bitowy, z możliwością generowania przerwań (MTIM). Pozwala on na dołączanie różnych źródeł sygnału zegarowego i generowanie przerwań. Może pracować w trybie pracy ciągłej (*free run*) i jako „*modulo timer*” z funkcją programowania okresowych przerwań po przepełnieniu licznika.

Przetwornik A/C może spełniać rolę prostego interfejsu dla dowolnego czujnika z wyjściem analogowym. Przetwornik może prowadzić konwersję w wybranym kanale, w trybie ciągłym lub jednokrotnym. Każda operacja konwersji jest inicjowana programowo lub z przerwaniem w czasie rzeczywistym (RTI). Przetwornik można skonfigurować tak, żeby porównywał w sposób ciągły jedno napięcie z wcześniej określoną dolną lub górną wartością graniczną napięcia i generował przerwanie w momencie przekroczenia tych wartości. Należy zaznaczyć, że funkcji tej można używać do wyprowadzenia mikrokontrolera z trybu oszczędzania zasilania, pod warunkiem, że jako źródło sygnału zegarowego wybierze się źródło zegarowe tego przetwornika.

Opisywane mikrokontrolery wyposażono także w komparator analogowy (ACMP) porównujący dwa napięcia wejściowe lub jedno wejściowe napięcie analogowe z wewnętrznym napięciem odniesienia (*bandgap reference*). Napięcie z modułu jest doprowadzane do innych układów wewnętrznych mikrokontrolera i do linii I/O. Włącza ono inne, wewnętrzne układy peryferyjne mikrokontrolera i systemy zewnętrzne w celu ciągłego monitorowania i dalszego przetwarzania sygnałów wyjściowych komparatora. Można też wyzwalać opcjonalne przerwania komparatora zbroczem narastającym, opadającym lub też jednym i drugim zbroczem sygnału wyjściowego komparatora. Takie przerwanie może być wykorzystywane do wyprowadzenia mikrokontrolera z trybu oszczędzania zasilania. Wszystkie mikrokontrolery mają czujnik temperatury, którego wyjście jest dołączone do jednego z kanałów przetwornika. Umożliwia to użytkownikowi pomiar temperatury struktury układu scalonego bez poświęcania na ten cel analogowego kanału wejściowego. Ponadto, aby zapewnić poprawną pracę wewnętrznego stabilizatora napięcia odniesienia oraz zasilacza zewnętrznego, można mierzyć w ten sam sposób wewnętrzne napięcie odniesienia i odpowiednio napięcie zasilacza.

W mikrokontrolerach są dostępne trzy interfejsy komunikacyjne. Interfejs SCI zapewnia transmisję między mikrokontrolerem i zdalnymi urządzeniami w standardzie UART (pełny duplex, asynchroniczny, NRZ). Interfejs SPI pozwala na ustanowienie połączeń w trybach: bardzo szybkim w trybie „Master” do 5 Mb/s oraz „Slave” do 2,5 Mb/s. Zintegrowany interfejs I²C jest przystosowany do transmisji danych z prędkością do 100 kb/s.

Oprócz rdzenia HC08 w mikrokontrolerach Freescale jest stosowany również rdzeń RS08, który jest rdzeniem „zredukowanym”. Co oznacza to w praktyce? Konstruktorzy tego rdzenia skupili się na zminimalizowaniu jego fizycznych wymiarów w krzemie (przekłada się to na cenę układu), a osiągnięto to dzięki usunięciu z rdzenia elementów nadmiarowych z punktu widzenia aplikacji dostępnych dla „małych” mikrokontrolerów. Najważniejsze zmiany wprowadzone w rdzeniu RS08 w stosunku do HCS08 to:

- Usunięcie z listy instrukcji poleceń: *Multiply*, *Divide*, *Nibble*, *Swap*, *Decimal Adjust*, *Acc*, *Negate*, *Complement*, *Arithmetic Shift* i niektórych instrukcji skoków oraz operacji na stosie.
- W rejestrze znaczników (flag) CCR pozostawiono jedynie dwie flagi: *Carry* i *Zero*.
- Z rdzenia usunięto rejestr H oraz wskaźnika stosu SP (Stack Pointer).
- Zabroniono wielopoziomowego wywoływania procedur z procedur.

- Licznik PC ma długość 14 bitów, co ogranicza maksymalny rozmiar pamięci programu do 16 kB.
- Wektorowy moduł obsługi przerwań zastąpiono znacznie prostszym mechanizmem, który przerzuca na programistę konieczność ustalenia źródła sygnału przerwania.

Optymalizacji rdzenia RS08 pod kątem dopasowania do „małych” aplikacji nie przeprowadzono wyłącznie za pomocą odcięcia fragmentów rdzenia uznanych za nadmiarowe, ale wprowadzono też kilka modyfikacji rozbudowujących rdzeń, które mają służyć zmniejszeniu zajętości pamięci Flash oraz przyśpieszeniu wykonywania programów:

- Wprowadzono dwa nowe tryby adresowania: *Short* oraz *Tiny* (są to uproszczone tryby adresowania bezpośredniego o ograniczonym zasięgu).
- Zastosowano rejestr *Shadow PC*, który służy do przechowywania adresu powrotu z procedury spełnia rolę jednopoziomowego stosu.
- Operacje na rejestrze *Shadow PC* umożliwiają dwie nowe instrukcje: SHA i SLA (wymiana zawartości rejestru z zawartością akumulatora).
- Wprowadzono tzw. pseudoinstrukcje, czyli mechanizm emulacji niektórych instrukcji niedostępnych na liście rdzenia RS08 (mają z niego automatycznie korzystać kompilatory).
- Wprowadzono mechanizm stronicowania pamięci w oknie 64-bajtowym, który umożliwia m.in. wykonywanie operacji na całej pamięci Flash o pojemności do 16 kB za pomocą rozkazów o „krótkim” adresowaniu.

Dzięki opisanym modyfikacjom uzyskano zmniejszenie powierzchni zajmowanej przez rdzeń o 30% w stosunku do wersji HCS08 wykonywanej w tej samej technologii, a przy umiejętnym podejściu do programowania mikrokontrolerów, możliwe jest także 15...20-procentowe zmniejszenie zapotrzebowania na pamięć programu.

Oprócz mikrokontrolerów 8-bitowych, firma Freescale wytwarza pokazną liczbę układów z rdzeniami 16-bitowymi HCS12 oraz jego wariantami – HCS12X oraz HCS12Z. Ich wykaz umieszczono w **tabeli 4**.

Rdzeń HC12 został wprowadzony w latach dziewięćdziesiątych przez firmę Motorola. Jest on rozszerzeniem stosowanego wcześniej rdzenia HC11 i dlatego oprogramowanie dla HC11 powinno działać również na HC12. Pierwsze mikrokontrolery z rdzeniem HC12 miały magistrale taktowane przebiegiem o częstotliwości 8 MHz i pamięć Flash do 128 kB. Podobnie jak HC11, HC12 ma dwa akumulatory 8-bitowe (A i B), które

Tabela 3. Wykaz mikrokontrolerów 8-bitowych produkowanych przez Freescale										
Typ	Obudowa	Flash [kB]	RAM [B]	EEPROM [kB]	Rdzeń	F _{CPU} [MHz]	Interfejsy szeregowo	Liczba I/O	Konwerter A/C	Inne
68HC705P6A	SOIC 28	-	16	-	-	-	SIOP	21	4×8-bit	-
HC08Q	SOIC 28	16	512	-	HC08	32	SPI, SCI	26	10×10-bit	COP, KBI, LVD, POR
RS08KA	SOIC 16, SOP 16, DFN 6, DIP 16, SOIC 20, DIP 20, SOIC 8, DIP 8	1...8	64...256	-	RS08	20	I ² C	2.0 - 18.0	8 lub 12×10-bit	Analog Comparator, COP, KBI, LVD
RS08KB	SOIC 20, SOP 16, DFN 8, SOIC 16, SOIC 8, QFN 24	2...12	126...254	-	RS08	20	I ² C, SCI	6.0 - 18.0	4...12×10-bit	Analog Comparator, LVD
RS08LA	QFN 48, QFP 48	8	256	-	RS08	20	SCI, SPI	34	6×10-bit	Segment Display Controller, COP, KBI, LVD
S08AC	QFP 64, QFP 44, QFP 32, QFN 48, QFP 80	8...128	768...8 k	-	S08	40	I ² C, SCI, SPI	22.0 - 70.0	6...16×10-bit	COP, Internal Clock Generator, Internal Voltage Regulator, KBI, LVD
S08AW	QFP 44, QFP 64, QFN 48, QFP 32	8...60	768...2 k	-	S08	40, 20	SCI, SPI, I ² C	22.0 - 54.0	6...16×10-bit	COP, KBI, LVD, POR, RTC, Temperature Sensor
S08D	QFP 100, QFP 64, QFP 48, QFP 32	16...128	1...8 k	0.0 - 2.0	S08	40, 20	I ² C, CAN, SCI, SPI	26.0 - 88.0	10...24×12-bit	COP, LVD, POR, RTC, Temperature Sensor, Analog Comparator, Internal Voltage Regulator, KBI
S08EL-SL	SOP 28, SOP 20	32, 16	1	0.512	S08	40, 20	SLIC, SCI, SPI, I ² C	22, 16	12 lub 16×10-bit	Temperature Sensor, RTC, POR, LVD, KBI, COP, Analog Comparator, Internal Clock Generator
S08FL	DIP 32, QFP 32	16, 8	1 k, 768	-	S08	20	SCI	30	12×8-bit	COP, ICS
S08GT	QFP 44, QFN 32, QFN 48	16, 8	2k, 1 k	-	S08	40	SCI, SPI, I ² C	24.0 - 39.0	4 lub 8×10-bit	LVD, RTC, WDG OSC/Timer
S08GW	QFP 80, QFP 64	64, 32	4 k, 2 k	-	S08	20	SCI, SPI, I ² C	57, 45	14×16-bit	Analog Comparator, COP, Internal Clock Generator, KBI, LVD, POR, RTC, Temperature Sensor
S08JE	QFP 80, QFP 64	128, 64	12 k	-	S08	48	SCI, SPI, I ² C	46, 33	12-bit	KBI, Temperature Sensor
S08JM	QFP 44, QFP 64, QFN 48, QFP 32	8...60	1...4 k	-	S08	48	SPI, I ² C, SCI, USB	21.0 - 51.0	4...12×12-bit	Analog Comparator, COP, Internal Voltage Regulator, KBI, LVD, RTC
S08JS	QFN 24, SOIC 20	16, 8	512	-	S08	48	SCI, SPI, USB	14	-	KBI, LVD, POR, RTC
S08LC	QFP 64, QFP 80	36, 60	2,5 k, 4 k	-	S08	40	I ² C, SCI, SPI	18.0 - 24.0	2 lub 8×12-bit	COP, KBI, RTC, LVD
S08LG	QFP 64, QFP 80, QFP 48	32, 18	2 k	-	S08	20, 40	SCI, SPI, I ² C	39.0 - 69.0	9...16×12-bit	WDG OSC/Timer, Temperature Sensor, COP, KBI, LVD, POR, RTC, SPDIF Transmitter
S08LH	QFP 64, QFP 80	64, 36	4 k	-	S08	40	SCI, SPI, I ² C	37, 39	8×16-bit	KBI
S08LL	QFP 48, QFP 64, QFN 48, QFP 80	8...64	2 k, 4 k	-	S08	20, 40	SCI, SPI, I ² C	31.0 - 64.0	8 lub 10×12-bit	COP, KBI, LVD, Analog Comparator

WYBÓR KONSTRUKTORA

Tabela 3. c.d.

Typ	Obudowa	Flash [kB]	RAM [B]	EEPROM [kB]	Rdzeń	F _{CPU} [MHz]	Interfejsy szeregowo	Liczba I/O	Konwerter A/C	Inne
S08MM	QFP 64, MAPBGA 81, QFP 80	32...128	2 k, 12 k	-	S08	48	SCI, SPI, I ² C, USB	33.0 - 47.0	12 lub 16×12-bit	Analog Comparator, COP, PLL, WDG OSC/Timer
S08MP	QFP 48, QFP 32	16	1 k	-	S08	20.0 - 50.0	SCI, SPI, I ² C	40, 25	8...13×12-bit	COP, KBI, WDG OSC/Timer, Analog Comparator, Internal Clock Generator, RTC, Temperature Sensor, PGA
S08P	QFP 48, QFP 44, QFP 32, SOP 20, SOP 16, QFP 64, SOIC 20	4...60	512...4 k	0.256, 0.128	S08	20	SCI, SPI, I ² C	14.0 - 57.0	10...16×12-bit	Internal Clock Generator, LVD, POR, Touch Sensing Interface, Analog Comparator, KBI, ICS
S08QA	DIP 8, DFN 8, SOIC 8	4, 2	256, 168	-	S08	20	-	5	4×10-bit	Analog Comparator, COP, KBI
S08QB	SOP 16, QFN 24, SOIC 28	8, 4	512, 256	-	S08	20	SCI	14.0 - 24.0	8×12-bit	Analog Comparator, COP, KBI, LVD, ICS
S08QD	SOIC 8, DIP 8	4, 2	256, 128	-	S08	8, 16	-	6, 4	4×10-bit	Internal Clock Generator, LVD, WDG OSC/Timer
S08QE	QFN 48, QFP 44, SOP 16, SOIC 20, SOIC 28, QFP 32, QFN 32, QFP 64, DIP 16, QFP 80	4...128	256...8 k	-	S08	50, 20	SCI, SPI, I ² C	12.0 - 70.0	8...24×10 lub 12-bit	Analog Comparator, COP
S08QG	SOP 16, QFN 16, DIP 16, QFN 24, DFN 8, DIP 8, SOIC 8	8, 4	512, 256	-	S08	20	SCI, SPI, I ² C	12, 4	4 lub 8×10-bit	LVD, WDG OSC/Timer
S08R	QFP 44	32	2 k	-	S08	16	SCI, SPI, SPI	39	-	COP, LVD, Analog Comparator, RTC
S08SC4	SOP 16	4	256	-	S08	20, 40	SCI	12	8×10-bit	COP, Internal Clock Generator, KBI, LVD, POR, Temperature Sensor
S08SE	SOIC 28, SOP 16	4, 8	256, 512	-	S08	20	SCI	24, 14	10×10-bit	COP, KBI, LVD, POR, RTC, ICS
S08SF	SOP 20, SOP 16	4	128	-	S08	40	I ² C	18, 14	8×10-bit	Analog Comparator, COP, Internal Voltage Regulator, KBI, LVD, WDG OSC/Timer
S08SG	SOP 16, SOP 20, SOP 28, SOIC 8, QFP 44	4...128	256...8 k	-	S08, HCS08	20.0 - 40.0	SCI, SPI, I ² C	4.0 - 34.0	4...16×10-bit	Analog Comparator, COP, Internal Clock Generator, KBI, LVD, POR, RTC, Temperature Sensor, Internal Voltage Regulator
S08SH	SOP 16, SOIC 8, SOP 20, SOIC 20, DIP 20, SOP 28, SOIC 28, QFN 24	4...32	256...1 k	-	S08	40	SPI, I ² C, SCI	4.0 - 23.0	4...16×10-bit	Analog Comparator, COP, Internal Clock Generator, Internal Voltage Regulator, LVD

Tabela 3. c.d.

Typ	Obudowa	Flash [kB]	RAM [B]	EEPROM [kB]	Rdzeń	F _{cpu} [MHz]	Interfejsy szeregowo	Liczba I/O	Konwerter A/C	Inne
S08SL	SOP 20, SOP 28	16, 8	512	0.256	S08	40, 20	SLIC, SCI, SPI, I ² C	16, 22	12 lub 16×10-bit	RTC, Temperature Sensor, Analog Comparator, COP, Internal Clock Generator, KBI, LVD, POR
S08SV	QFP 32	16	1 k	-	S08	40	SCI, SPI, I ² C	30	12×10-bit	Analog Comparator, COP, KBI, LVD, Temperature Sensor, WDG OSC/Timer

można połączyć w parę i traktować jak pojedynczy akumulator 16-bitowy (D). Oprócz tego, w architekturze rdzenia wyróżniono rejestry indeksowe X i Y, 16-bitowy licznik adresów, 16-bitowy wskaźnik stosu i 8-bitowy rejestr warunków wykonywania programu (CCR).

Na początku roku 2000 rodzinę mikrokontrolerów z rdzeniem HC12 rozszerzono o MC9S12, w której szyna danych była taktowana z częstotliwością do 25 MHz, a pamięć Flash mieściła do 512 kB.

We wrześniu 2004 r. firma Freescale wprowadziła do oferty MC9S12NE64, który był reklamowany jako pierwszy mikrokontroler zapewniający szybkie połączenie Ethernet i wyposażony w reprogramowalną pamięć programu Flash (slogan z materiałów Freescale: „*industry's first single-chip fast-Ethernet Flash microcontroller*”). Zawierał on rdzeń HCS12 taktowany przebiegiem o częstotliwości 25 MHz, 64 kB pamięci Flash, 8 kB pamięci RAM i kontroler Ethernet 10/100 Mbit/s. Kolejny wersja tego

mikrokontrolera – MC9S12XDP512 miała magistralę taktowaną przebiegiem o częstotliwości 40 MHz oraz koprocesor nazwany XGATE. Nowa wersja rdzenia HCS12 – HCS12X, w którą był wyposażony nowy mikrokontroler, wprowadzała kilka nowych instrukcji, których zadaniem była poprawa wydajności CPU.

W maju 2006 r. firma Freescale wprowadziła kolejną nowość wśród mikrokontrolerów 16-bitowych – MC9S12XEP100 z rdzeniem HCS12X. Ma on magistralę taktowaną przebiegiem o częstotliwości 50 MHz i dodatkowy moduł ochrony pamięci (bazujący na mechanizmie segmentacji) oraz możliwość emulowania EEPROM za pomocą Flash. W tych mikrokontrolerach wprowadzono też specjalny mechanizm adresowania pamięci o pojemności większej niż 64 kB (stronicowanie oraz adresowanie globalne umożliwiające „sięgnięcie” aż do 8 MB).

Ważną innowacją w mikrokontrolerach Freescale jest koprocesor XGATE. Jest to 16-bitowy procesor RISC pracujący

z podwojoną częstotliwością magistrali. Odciąża ona jednostkę centralną przejmując obsługę przerwań. Pierwsze wersje tego koprocesora nie pozwalały na przerywanie obsługi bieżących przerwań przez przerywanie o wyższym priorytecie. Pozwalała na to dopiero trzecia wersja tego koprocesora (XGATEV3) zaimplementowana w 9S12XEP100 i innych, późniejszych S12X. W najnowszych rozwiązaniach rdzeń S12X może zgłaszać przerywanie programowe koprocesorowi XGATE, podobnie jak XGATE jednostce centralnej. Producent zaimplementował system flag, dzięki któremu rdzeń i koprocesor synchronizują dostęp do modułów peryferyjnych i unikają kolizji. Typowo, kod programu wykonywanego przez XGATE jest kopiowany do pamięci RAM przy starcie mikrokontrolera w celu uzyskania jak największej szybkości jego wykonywania. Przestrzeń, która może zaadresować XGATE sięga 64 kB (bez stronicowania). Rejestry są wspólne dla CPU i XGATE, jednak obszary pamięci RAM są rozdzielne.

Tabela 4. Wykaz mikrokontrolerów 16-bitowych produkowanych przez Freescale

Typ	Obudowa	Flash [kB]	RAM [kB]	EEPROM [kB]	Rdzeń	F _{cpu} [MHz]	Interfejsy szeregowo	Liczba I/O	Konwerter A/C	Inne
56F801X	QFP 32	16, 12	4, 2	-	-	32	LIN, I ² C, SCI, SPI	26	6 lub 8×12-bit	WDG OSC/Timer, POR, PLL, LVD, Internal Voltage Regulator, Internal Clock Generator, COP
56F802X	QFP 64, QFP 32, QFP 44	32	4	-	-	32	SCI, SPI, CAN, I ² C, LIN	26...53	6...16×12-bit	WDG OSC/Timer, POR, PLL, LVD, Internal Clock Generator, COP, Analog Comparator
56F803X	QFP 44, QFP 64, QFP 32, QFP 48	64	8	-	-	32	SCI, SPI, I ² C, LIN, CAN, QSPI	26...53	6...16×12-bit	-
56F824X_825X	QFP 48, QFP 64, QFP 44	48, 64	6, 8	-	-	60	IIC/SMBus, SCI, QSPI, CAN	-	8...16×12-bit	COP, LVD, PLL, POR
56F8335_8135	QFP 128	64	12, 8	-	-	60, 40	SCI, SPI, CAN	49	16×12-bit	Internal Voltage Regulator, LVD, Quad Decoder, Temperature Sensor, WDG OSC/Timer
56F834X-814X	QFP 144, QFP 128	128	12, 8	-	-	60, 40	SCI, SPI, CAN	62, 49	16×12-bit	Internal Voltage Regulator, LVD, Quad Decoder, Temperature Sensor, POR, PLL, COP

WYBÓR KONSTRUKTORA

Tabela 4. c.d.

Typ	Obudowa	Flash [kB]	RAM [kB]	EEPROM [kB]	Rdzeń	F _{cpu} [MHz]	Interfejsy szeregowe	Liczba I/O	Konwerter A/C	Inne
56F835X-815X	QFP 128, QFP 144, QFP 160	256	20, 16	-	-	60, 40	SCI, SPI, CAN	49...76	16×12-bit	Temperature Sensor, Quad Decoder, LVD, Internal Voltage Regulator
56F836X-816X	QFP 160, QFP 128, QFP 144	512	36, 32	-	-	60, 40	SCI, SPI, CAN	49...76	16×12-bit	Internal Voltage Regulator, LVD, Quad Decoder, Temperature Sensor
S12ZVM	QFP 64	128	8	0,512	S12Z	50, 40	-	64	12-bit	-
DSP56F801	QFP 48	16	2	-	-	80	SCI, SPI	11	8×12-bit	Internal Clock Generator
DSP56F-801FA60	QFP 48	16	2	-	-	60	SCI, SPI	11	8×12-bit	Internal Clock Generator, LVD
DSP56F803	QFP 100	64	4	-	-	80	SCI, SPI, CAN	16	8×12-bit	LVD, Quad Decoder
DSP56F805	QFP 144	64	4	-	-	80	SCI, SPI, CAN	32	8×12-bit	LVD, Quad Decoder
DSP56F807	MAPBGA 160, QFP 160	128	4	-	-	80	SCI, SPI, CAN	32	16×12-bit	LVD, Quad Decoder
MC56F800x	QFP 32, SOIC 28, QFP 48	16, 12	2	-	-	32	LIN, I ² C, SCI, SPI	9...26	36×12-bit	PGA, WDG OSC/Timer, PLL, LVD, Internal Voltage Regulator, Internal Clock Generator, COP, POR
MC56F827xx	QFN 32, QFP 64, QFP 32, QFP 48	16...64	4...8	0	56800 EX	100, 50	QSCI, QSPI, I ² C, CAN	26...54	12-bit	-
MC56F84xxx	QFP 80, QFP 48, QFP 64, QFP 100	64...256	8...32	2	56800 EX	60.0 - 100.0	QSCI, QSPI, I ² C, CAN	39...86	12-bit	-
MC9S12H	QFP 144, QFP 112	256, 128	12, 6	4, 2	HCS12	50	SCI, SPI, I ² C, CAN	83...117	8 lub 16×10-bit	External Memory Controller, LVD
MM912_634	QFP 48	32...64	6, 2	-	HCS12	20, 16	-	-	-	-
MM912_P812	QFP 100	128, 96	6	-	HCS12	32	-	-	10×12-bit	-
MM912_S812	QFP 100	128	8	-	HCS12	40, 400	-	-	-	-
S12A	QFP 80, QFP 112	32...512	4...14	1...4	-	25	SCI, SPI, I ² C	91, 59	8×10-bit	External Memory Controller, LVD
S12B	QFP 80, QFP 112	64...256	4, 2	1, 2	-	-	CAN, SPI, SCI, I ² C	59, 91	8 lub 16×10-bit	COP, Internal Voltage Regulator, External Memory Controller
S12C	QFP 48, QFP 52, QFP 80	32...128	4, 2	-	-	25	SCI, SPI, CAN	31...60	8×10-bit	COP, LVD, POR
S12D	QFP 112, QFP 80	32...512	4...12	1...4	HCS12	25	SPI, SCI, I ² C, CAN, J1850	91, 59	8 lub 16×10-bit	COP, Internal Voltage Regulator, PLL, External Memory Controller
S12E	QFP 80, QFP 112	32...256	2...16	-	HCS12	50	SCI, SPI, I ² C	58, 90	16×10-bit	LVD
S12G	QFP 48, QFP 32, QFP 64, QFN 48, QFP 100, SOP 20	16...240	1...12	0,5...4	HCS12	25	SCI, SPI, CAN	14...86	6...16×10 lub 12-bit	Internal Clock Generator, Internal Voltage Regulator, LVD, PLL, RTC, WDG OSC/Timer, Analog Comparator
S12GC	QFP 48, QFP 80, QFP 52	16...128	1...4	-	HCS12	50	SCI, SPI	31...60	8×10-bit	LVD, COP, Internal Voltage Regulator
S12HY	QFP 100, QFP 64	32...64	2, 4	4	HCS12	32	CAN, SCI, SPI, I ² C, LIN	80, 50	6 lub 8×10-bit	WDG OSC/Timer, RTC, PLL, LVD, Internal Voltage Regulator, Internal Clock Generator, Motor Controller
S12HZ	QFP 112, QFP 80	64...256	4...12	2, 1	HCS12	50	SCI, SPI, I ² C, CAN	85, 58	16×10-bit	-
S12K	QFP 80, QFP 112	128	8	2	HCS12	50	SCI, SPI, I ² C, CAN	59, 91	16×10-bit	COP, LVD, POR
S12P	QFN 48, QFP 80, QFP 64	32...128	2...6	4	HCS12	32	CAN, SCI, SPI	34...64	10×12-bit	Internal Clock Generator, Internal Voltage Regulator, LVD, PLL, RTC, WDG OSC/Timer
S12Q	QFP 48, QFP 80, QFP 52	64...128	2...4	-	HCS12	32, 16	CAN, SCI, SPI	31...60	8×10-bit	POR, PLL, LVD, Internal Voltage Regulator, Internal Clock Generator, COP
S12VR	QFP 32, QFP 48	-	-	-	-	-	-	-	-	-
S12XA	QFP 112, QFP 144, QFP 80	256, 512	16, 32	4	HCS12X	80	SCI, SPI, I ² C	59...119	8...24×10-bit	-

Tabela 4. c.d.

Typ	Obudowa	Flash [kB]	RAM [kB]	EEPROM [kB]	Rdzeń	F _{cpu} [MHz]	Interfejsy szeregowo	Liczba I/O	Konwerter A/C	Inne
S12XB	QFP 80, QFP 112	128, 256	6, 10	1, 2	HCS12X	40, 66	CAN, SCI, SPI, I ² C	59, 91	8 lub 16×10-bit	COP, Internal Clock Generator, Internal Voltage Regulator, LVD, PLL, POR
S12XD	QFP 112, QFP 144, QFP 80, Hardware Only	64...512	4...32	1...4	HCS12X	80, 40	CAN, SCI, SPI, I ² C	59...119	8...24×10-bit	COP, Internal Clock Generator, Internal Voltage Regulator, LVD, PLL, POR
S12XE	QFP 144, QFP 80, QFP 112, MAPBGA 208	128...1000	12...64	4, 2	HCS12X	50	CAN, SCI, SPI, I ² C	59...152	8...32×10-bit	COP, LVD, POR, Temperature Sensor, WDG OSC/Timer
S12XF	QFP 112, QFP 64	256...512	20...32	4, 2	HCS12X	50	CAN, SCI, SPI, FlexRay	-	8 lub 16×10-bit	COP, Internal Clock Generator, Internal Voltage Regulator, LVD, PLL, POR
S12XHY	QFP 100, QFP 112	128, 256	8, 12	8	HCS12	40, 80	CAN, SCI, SPI, I ² C	76, 88	8 lub 12×10-bit	Internal Voltage Regulator, LVD, PLL, RTC, WDG OSC/Timer, Motor Controller
S12XHZ	QFP 144, QFP 112	256...512	16...32	4	HCS12X	40, 80	CAN, SCI, SPI, I ² C	117, 85	16×10-bit	COP, Internal Voltage Regulator, LVD, PLL, POR
S12XS	QFP 80, QFP 64, QFP 112	64...256	4...12	8, 4	HCS12X	40	SCI, SPI, CAN	44...91	8 lub 16×8 lub 12-bit	COP, Internal Clock Generator, Internal Voltage Regulator, LVD, POR
S12ZVH	QFP 144, QFP 100	32...128	2...8	2, 4	S12Z	64	-	73...104	4 lub 8	Stepper Motor Controller



MICROCHIP

Microchip – własne pomysły na mikrokontrolery

Microchip jest jedną z tych firm, które nie uległy powszechnej „armizacji”. Nadal promuje rozwiązania natywne i całkiem dobrze ma się na rynku mikrokontrolerów. Niemniej jednak ostra konkurencja wymusza na firmie wprowadzanie innowacji do swoich wyrobów, w tym modułów funkcjonalnych do mikrokontrolerów 8-bitowych, które kiedyś były zarezerwowane dla mocniejszych rdzeni. W tabeli 5 wymieniono mikrokontrolery z rdzeniem 8-bitowym będące w aktualnej Microchipsa. Napiszmy kilka słów o tym, co nowego można znaleźć w strukturach tych mikrokontrolerów.

Bramki logiczne zintegrowane z mikrokontrolerem (CLC) pozwalają na wykonywanie w czasie rzeczywistym operacji, takich jak: iloczyn logiczny (AND), suma logiczna (OR), alternatywa wykluczająca (XOR), zaprzeczenie (NOT), negacja iloczynu (NAND), negacja sumy (NOR) i negacja alternatywy wykluczającej (XNOR). Dzięki możliwości ich konfigurowania, do każdej z wymienionych operacji wystarczy pojedyncza bramka. Ponadto, jest możliwe tworzenie automatów w oparciu o wbudowane przerzutniki D, JK oraz zatraski D i SR. Jako wejścia do bramek i przerzutników mogą posłużyć sygnały podawane bezpośrednio na wyprowadzenia

obudowy mikrokontrolera lub pochodzące z innych modułów funkcjonalnych. Wyjścia mogą być przekierowane na inne wyprowadzenia lub do pozostałych modułów funkcjonalnych. Co ważne, funkcje logiczne będą realizowane również w czasie, gdy mikrokontroler znajduje się w trybie uśpienia.

Kolejnym ciekawym układem peryferyjnym jest CWG (*Complementary Waveform Generator*) lub COG (*Complementary Output Generator*), który służy do generowania dwóch, częściowo uzależnionych od siebie sygnałów w oparciu o jeden wejściowy. Na jego wejście mogą być doprowadzone sygnały z komparatorów, PWM-u, komórek logicznych lub sterowanego oscylatora. Generator CWG analizuje zmiany na wejściu umożliwiając filtrowanie krótkich impulsów oraz pozwala na opóźnianie wyjścia w fazie względem wejścia. Dwa, powiązane ze sobą sygnały wyjściowe pozwalają na bezpośrednie sterowanie pracą wielu złożonych urządzeń, takich jak np. silniki. Ułatwiają tworzenie zasilaczy, systemów oświetlenia LED, ładowarek akumulatorów, układów korekcji współczynnika mocy i wzmacniaczy audio klasy D. Dzięki scaleniu w obudowie mikrokontrolera zmniejszają koszt i fizyczną wielkość gotowego urządzenia oraz ograniczają pobór mocy.

W niektórych zastosowaniach bardzo przydatny może okazać się oscylator sterowany cyfrowo (*Numerically Controlled Oscillator*, NCO), który umożliwia liniowe regulowanie częstotliwości generowanego przebiegu. Sygnał wyjściowy z NCO jest tworzony z 20-bitową rozdzielczością w oparciu o 16-megahercowy sygnał zegarowy. Generowana częstotliwość jest ustawiana za pomocą liczby 16-bitowej, co daje skok 15 Hz.

Maksymalna częstotliwość przebiegu wyjściowego to 500 kHz. Co więcej, wyjście może pracować w dwóch trybach: standardowym, gdy czas trwania impulsu wynosi połowę jego okresu (tryb FDC) oraz z regulowanym współczynnikiem wypełnienia (tryb PFM).

Układ PSMC (*Programmable Switch Mode Controller*) to zaawansowany sterownik umożliwiający generowanie sygnałów w oparciu o różne wejścia zegarowe (wejście zewnętrzne, systemowe lub niezależny zegar 64 MHz) i różne sygnały sterujące pochodzące z komparatorów i lub doprowadzone do mikrokontrolera z innych obwodów. Pozwala on na generowanie do 6 oddzielnych sygnałów PWM lub do 3 podwójnych, komplementarnych, z brankowanymi wyjściami. Rozdzielczość sterowania PWM to 16 bitów, a dostępne tryby pracy wyjść umożliwiają sterowanie różnymi typami silników. Moduł PSMC znajdzie też zastosowanie w zasilaczach, przetwornicach, układach PFC, systemach sterowania oświetleniem, napędach i aplikacjach z czujnikami. Upraszcza generowanie szybkich, precyzyjnie sterowanych sygnałów PWM.

Prawie każdy z mikrokontrolerów wyposażony w przetwornik A/C ma też wbudowany czujnik temperatury. Jego wskazania mieszczą się w zakresie od -40°C do +85°C, co pozwala na łatwy pomiar temperatury w aplikacji, bez konieczności stosowania dodatkowych komponentów zewnętrznych. Jest przydatny w takich aplikacjach, jak telefony komórkowe, urządzenia o dużej mocy, konsumencki sprzęt domowy i wielu innych. Ułatwia implementację funkcji zabezpieczania urządzeń przed przegrzaniem.

W wielu aplikacjach przyda się generator modulowany cyfrowo. Blok ten umożli-

WYBÓR KONSTRUKTORA

wia modulowanie częstotliwości nośnej za pomocą strumienia danych. Może nim być sygnał z wyprowadzenia układu, zegar odniesienia, przebieg PWM, sygnał UART/SPI/I²C, bit rejestru lub wyjście wbudowanego komparatora. Ułatwia to tworzenie nadajników IrDA oraz przygotowywanie danych do transmisji ASK, FSK, PSK.

Kolejną innowacją jest bramkowany timer, który służy do pomiaru długości impulsów wejściowych. Odliczanie może być rozpoczęte lub wstrzymane za pomocą dowolnego sygnału, którym może być wyjście komparatora, Timer 0 lub wyprowadzenie T1G mikrokontrolera. 16-bitowy licznik Timera 1 może być taktowany przebiegiem z wbudowanego oscylatora (opcjonalnie 4-krotnie spowolnionego) lub oddzielnym z Timera 1. Rozpoczęcie i zakończenie pomiaru może być uruchamiane w oparciu o zbocze opadające lub narastające monitorowanego sygnału. Po zakończeniu pomiaru timer wyzwala przerwanie. Bramkowany timer nie wymaga do działania dodatkowych komponentów i ułatwia dekodowanie sygnału PWM oraz pomiaru czasu i częstotliwości impulsów.

W strukturę nowych układów wbudowano też rejestr typu „zatrask”. Może on być sterowany sygnałem z wbudowanego komparatora, programowalnego zegara, pochodzącym spoza mikrokontrolera lub za pomocą kodu programu. Jego wyjście może być wyprowadzone na zewnątrz lub dołączone wewnątrz mikrokontrolera do któregoś z modułów funkcjonalnych. W praktyce i we współpracy z komparatorem ten zatrask pozwala na wykonanie wszystkiego, co dało się wykonać w oparciu o popularny timer 555 – astabilnych i bistabilnych generatorów sygnałów bez potrzeby stosowania wielu dodatkowych podzespołów. Może znaleźć zastosowanie do budowy modulatora ASK, budowy ściemniaczy, komparatorów fazy i generatorów dźwięków lub generatorów PWM. Uwaga – funkcja zatrasku SR jest zaimplementowana również w omówionym

na początku artykułu bloku konfigurowalnych komórek logicznych CLC.

Dodatkowe bloki peryferyjne sprawiają, że nawet, jeśli do aplikacji zostanie wybrany tani mikrokontroler 8-bitowy o niedużej wydajności, to i tak za jego pomocą można zrealizować zaawansowane funkcje sterowania. Co więcej, zwykle omawiane moduły funkcjonalne nie będą wymagały dodatkowych komponentów, więc ich zastosowanie upraszcza urządzenie, zmniejsza wymagania odnośnie do wielkości płytki drukowanej oraz podnosi niezawodność rozwiązania.

Prawdziwą trudność stanowi zaprezentowanie przeglądu mikrokontrolerów 16-bitowych. Ich wybór w ofercie Microchipsa jest bardzo duży. Pełna lista obejmuje ponad 360

salnych mikrokontrolerów przeznaczonych do użycia w typowych aplikacjach. Układy dsPIC33 są w znacznym stopniu zgodne programowo i pod względem rozmieszczenia wyprowadzeń z dsPIC33. Najważniejsze z punktu widzenia użytkownika będą jednak różnice pomiędzy tymi rodzinami, ponieważ to one będą predysponowały mikrokontroler do danej aplikacji.

W mikrokontrolerach dsPIC33 zwiększono prędkość wykonywania programu do 40 MIPS, dostępną pamięć Flash do 256 kB i RAM do 30 kB. Zwiększono również liczbę timerów 16-bitowych do 9, a 32-bitowych do 4. Również liczba wyprowadzeń I/O wzrosła do 85 (ich liczba zależy od wersji układu). Jednym z elementów wpływających na wzrost



pozycji i składają się na nią następujące rodziny mikrokontrolerów: dsPIC30, dsPIC33 i PIC24. Są one dostępne i doskonalone od około 10 lat. W związku z ogromną liczbą układów poprzestaniemy na ich omówieniu, nie będziemy wymieniali ich w tabeli. Osoby zainteresowane ich przeglądem odsyłamy do strony internetowej, której adres podano w materiałach źródłowych.

Mikrokontrolery dsPIC łączą cechy klasycznych mikrokontrolerów oraz prostszych procesorów DSP. Rodzinę układów dsPIC30 zoptymalizowano pod kątem wykorzystania w układach czujnikowych, w systemach zasilania, sterowania napędami elektrycznymi, a także w celu powiększenia grona uniwer-

wydajności jest kontroler DMA pośredniczący w transmisji danych pomiędzy pamięcią SRAM a większością modułów peryferyjnych. Ponadto, istotną modyfikacją jest zwiększenie maksymalnej częstotliwości próbkowania 10-bitowego przetwornika A/C do 1,1 MSA/s oraz maksymalnej liczby jego kanałów pomiarowych do 32. Mikrokontrolery dsPIC33 zaprojektowano w taki sposób, że mogą one być stosowane wymiennie z odpowiednikami z rodziny PIC24.

Rodzina mikrokontrolerów PIC24 była pierwszym krokiem Microchipsa w stronę mikrokontrolerów 16-bitowych. Jednym z założeń podczas ich projektowania było zapewnienie kompatybilności – także pod względem

Tabela 5. Wykaz mikrokontrolerów Microchip z rdzeniem 8-bitowym

Typ	Flash [kB]	RAM [B]	EEPROM / HEF	Liczba I/O	F _{cpu} [MHz]	Inne
PIC12F1571	1,75	128	HEF	8	32	Analog Comparator, A/C 4×10-bit, C/A 1×5-bit, 3×PWM, CWG
PIC12F1572	3,5	256	HEF	8	32	Analog Comparator, A/C 4×10-bit, C/A 1×5-bit, 3×PWM, UART, CWG
PIC16F1574	7	512	HEF	14	32	2×Analog Comparator, A/C 8×10-bit, C/A 1×5-bit, 5×PWM, UART, CWG
PIC16F1579	14	1024	HEF	14	32	2×Analog Comparator, A/C 12×10-bit, C/A 1×5-bit, 5×PWM, UART, CWG
PIC16F1703	3,5	256	HEF	14	32	2×Analog Comparator, 8×10-bit, SPI, I ² C
PIC16F1704	7	512		14	32	2×Analog Comparator, A/C 8×10-bit, C/A 1×8-bit, UART, SPI, I ² C, 3×CLC, CWG
PIC16F1705	14	1024	HEF	14	32	2×Analog Comparator, A/C 8×10-bit, C/A 1×8-bit, UART, SPI, I ² C, 3×CLC, CWG
PIC16F1707	3,75	256	HEF	20	32	2×Analog Comparator, A/C 8×10-bit, SPI, I ² C
PIC16F1708	7	512	HEF	20	32	2×Analog Comparator, A/C 12×10-bit, C/A 1×8-bit, UART, SPI, I ² C, 3×CLC, CWG
PIC16F1709	14	1024	HEF	20	32	2×Analog Comparator, A/C 12×10-bit, C/A 1×8-bit, UART, SPI, I ² C, 3×CLC, CWG
PIC16F1713	7	512	HEF	28	32	2×Analog Comparator, A/C 17×10-bit, C/A 2×8-bit, UART, SPI, I ² C, 4×CLC, CWG
PIC16F1716	14	1024	HEF	28	32	2×Analog Comparator, A/C 17×10-bit, C/A 2×8-bit, UART, SPI, I ² C, 4×CLC, CWG
PIC16F1717	14	1024	HEF	40	32	2×Analog Comparator, A/C, 28×10-bit, C/A 2×8-bit, UART, SPI, I ² C, 4×CLC, CWG
PIC16F1718	28	2048	HEF	28	32	2×Analog Comparator, A/C 17×10-bit, C/A 2×8-bit, UART, SPI, I ² C, 4×CLC, CWG
PIC16F1719	28	2048	HEF	40	32	2×Analog Comparator, A/C 28×10-bit, C/A 2×8-bit, UART, SPI, I ² C, 4×CLC, CWG
PIC12F1612	3,5	256		8	32	2×Analog Comparator, A/C 4×10-bit, WWDI, CRC/Scan, CWG
PIC16F1613	3,5	256	HEF	14	32	2×Analog Comparator, A/C, 8×10-bit, WWDI, CWG

wyprowadzeń – z procesorami dsPIC33 i niektórymi ze starszej rodziny PIC18. Dzięki temu konstruktorzy mogą łatwo migrować pomiędzy różnymi platformami, o różnej wydajności i procesorami DSC. Rodzina PIC24 składa się z dwóch grup układów: PIC24F o prędkości dochodzącej do 16 MIPS oraz PIC24H o prędkości do 40 MIPS. Ponadto, układy PIC24H wyposażono w DMA, który służy do szybkiej transmisji danych pomiędzy pamięcią, a większością modułów funkcjonalnych. Charakterystycznym elementem wyposażenia jest także 12-bitowy przetwornik A/C o maksymalnej częstotliwości próbkowania 500 kHz (w odróżnieniu od wersji „F” wyposażonej w równie szybki, ale 10-bitowy przetwornik A/C).

W strukturę mikrokontrolera wbudowano 16-pozycyjny potencjometr cyfrowy sterowany za pomocą rejestru 4-bitowego. Dzięki temu można precyzyjnie ustalić i zmieniać dynamicznie wartość napięcia odniesienia za pomocą programu.

Standardowym wyposażeniem mikrokontrolerów PIC24 jest zaawansowany interfejs do komunikacji szeregowej, przystosowany do bezpośredniej współpracy z interfejsami RS232, RS485, LIN 1.2 i IrDA (oczywiście za pomocą odpowiedniego interfejsu fizycznego). Wyposażono go między innymi w automatyczny detektor prędkości transmisji oraz system automatycznego testowania *loopback*. Bezpieczeństwo komunikacji szeregowej można zwiększyć wykorzystując sprzętowy generator CRC, który dzięki wydzielaniu rejestrów dostępowych można wykorzystać do obliczania CRC dowolnych grup danych.

Standardowym wyposażeniem wszystkich mikrokontrolerów z rodziny PIC24 są interfejsy SPI i I²C, a wybrane układy mają także wbudowane sprzętowe interfejsy CAN (lub *Enhanced CAN*). Wyposażeniem, które z pewnością ucieszy wielu projektantów jest kompletny zegar czasu rzeczywistego

z kalendarzem i rejestrami alarmu. Wykrycie ustawionego czasu alarmu może być sygnalizowane przerwaniem oraz zmianą stanu jednej z linii I/O. Znika więc występujący w wielu aplikacjach problem odmierzenia czasu oraz programowego realizowania kalendarza. Mikrokontroler można łatwo skonfigurować do pracy w roli zaawansowanego budzika i to w zdecydowanej części w oparciu o wbudowany w niego sprzęt, nie angażując mocy CPU.

Rdzeń mikrokontrolerów wyposażono w bloki mnożenia sprzętowego (także zmienoprzecinkowego) oraz dzielenia (32:16 bitów), a liniowa przestrzeń adresowa pamięci programu obejmuje aż 12 MB. Dostęp do pamięci danych ułatwiają dwa generatory adresów – osobne dla odczytu i zapisu danych. Maksymalna, liniowa przestrzeń adresowa dla pamięci danych wynosi 64 kB.

Pamięć programu Flash można programować za pomocą interfejsu ICSP lub JTAG. Ten drugi można wykorzystać także do testowania krawędziowego, natomiast do wspomaganie uruchamiania systemu z mikrokontrolerem PIC24 służy specjalny, dwuliniowy interfejs ICE.

Microchip zrezygnował (jak na razie) z podążania modnym kursem „na ARM-y” i forsuje własne, przemyślane rozwiązania. Jak pokazują dotychczasowe doświadczenia, od samego początku istnienia firmy inżynierowie firmy Microchip potrafią przewidzieć zapotrzebowanie rynku, czego przykładem jest choćby sukces rodziny procesorów dsPIC. Jednym z elementów sukcesu, kontynuowanym także w odniesieniu do prezentowanych nowości, jest szanowanie nawyków projektantów korzystających z mikrokontrolerów Microchipa, dlatego właśnie zadbano o ich wzajemną kompatybilność, a także o to, żeby projektanci mogli korzystać z dotychczas stosowanych narzędzi, jak choćby ze środowiska MPLab.

Podsumowanie

W pierwszej części artykułu udało się dokonać przeglądu mikrokontrolerów oferowanych przez Atmel, Freescale i Microchip. W kolejnej przyjrzymy się ofercie innych producentów, w tym rzadziej wybieranych do projektowanych urządzeń. Jednak już nawet po działaniach firm wymienionych w tym przeglądzie widać, że oferta nieskomplikowanych mikrokontrolerów 8-bitowych jest nadal rozwijana i zapewne jeszcze przez wiele lat będą one stosowane w różnych urządzeniach. Dobrze też mają się mikrokontrolery 16-bitowe, a ich producenci silnie konkurują z rozwiązaniami proponowanymi przez firmę ARM.

Jacek Bogusz, EP

Bibliografia:

1. <http://goo.gl/GyL08w>
2. <http://goo.gl/2IK5IS>
3. <http://goo.gl/vgDRN9>
4. <http://goo.gl/IvS0EC>
5. <http://goo.gl/0frWp3>
6. <http://goo.gl/oAqGCP>
7. <http://goo.gl/IOLoZ1>
8. Andrzej Gawryluk „8-bitowe maluchy – historii ciąg dalszy”, *Elektronika Praktyczna* 7/2006
9. Jarosław Doliński „Mikrokontrolery 8-bitowe HCS08”, *Elektronika Praktyczna* 5/2004
10. Tomasz Jabłoński „NCO i CWG w mikrokontrolerach PIC”, *Elektronika Praktyczna* 6/2012
11. Marcin Karbowiczek „Nowoczesne układy peryferyjne 8-bitowych mikrokontrolerów PIC”, *Elektronika Praktyczna* 11/2012
12. Andrzej Gawryluk „dsPIC33/PIC24 – nowe mikrokontrolery firmy Microchip”, *Elektronika Praktyczna* 12/2005”