DSP (Digital Signal Processing), czyli cyfrowe przetwarzanie sygnałów, to wyzwanie większe niż zabawa zwykłymi mikrokontrolerami. Dla wielu elektroników tematyka ta jest z pewnością fascynująca, a zmierzenie się z tymi procesorami stanowi emocjonującą przygodę. Stawiającym dopiero pierwsze kroki w tej dziedzinie, tematyka ta może wydawać się trudna i odległa od tego czym do tej pory się zajmowali. Na dobry początek polecam specjalny zestaw ewaluacyjny - 56F800 Demonstration Board.

Hybrydowe mikrokontrolery DSP

56F800DEMO to zestaw ewaluacyjny Motoroli opracowany w celu zaprezentowania możliwości procesorów sygnałowych rodziny 56800. Rodzina liczy kilka typów układów różniących się wielkością pamięci programu i danych, wyposażeniem w peryferia i obudowami. Rdzeń procesorów (**rys. 1**) wykonano w oparciu o architekturę harwardzką. Trzy równolegle pracujące jednostki pozwalają na wykonanie do sześciu operacji w jednym cyklu rozkazowym. Na płytce demonstracyjnej (**fot. 1**) zamontowano układ 56F801. Użytkownik dostaje również w komplecie bogatą dokumentację w postaci elektronicznej oraz niezbędne do prób oprogramowanie narzędziowe. Najważniejszym jego elementem jest kompletne środowisko uruchomieniowe IDE - CodeWarrior, umożliwiające pisanie i uruchamianie programów przeznaczonych dla omawianych procesorów. Sterowniki urządzeń peryferyjnych oraz biblioteki i interfejsy pozwalające na tworzenie specyficznych aplikacji w języku C są zawarte w SDK (*Software Develop-*

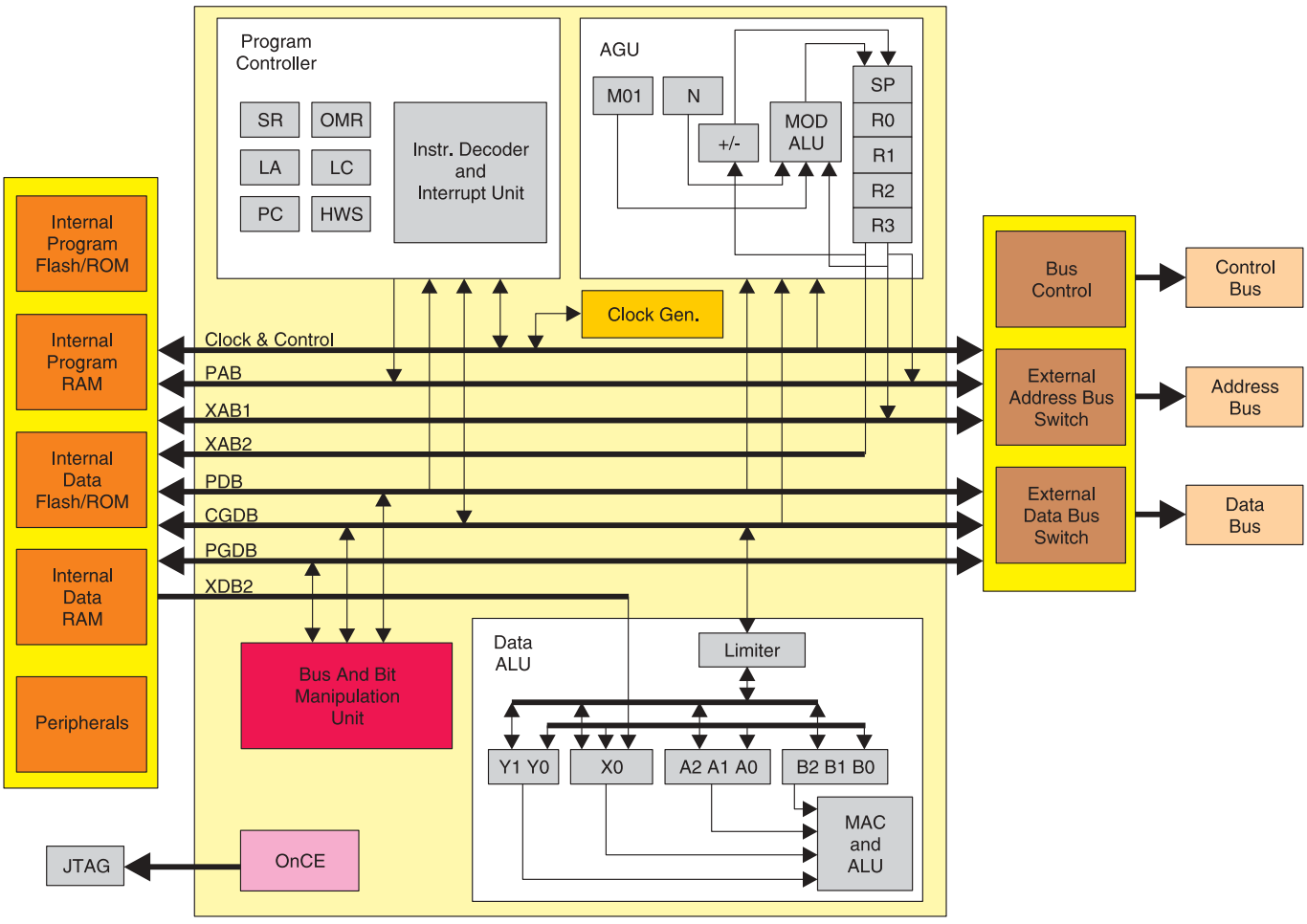
ment Kit). Na CD-ROM-ie jest też multimedialna prezentacja programu. Do płytki ewaluacyjnej dołączono zasilacz i kabel równoległy. Zanim zajmiemy się samym zestawem przyjrzyjmy się pobieżnie procesorowi DSP56F801.

56F801 - 16-bitowy procesor hybrydowy

Tak został nazywany przez producenta układ, który zamontowano na w płytce ewaluacyjnej 56F800DEMO. W układach rodziny 56F8xx połączono typowy mikrokontroler z procesorem DSP. Takie połączenie zastosowane przez Motorolę jest czymś wyjątkowym wśród producentów. Większość z nich wyraźnie rozgranicza klasyczne mikrokontrolery i mikroprocesory od procesorów DSP. Dzięki cechom charakterystycznym dla zwykłych mikrokontrolerów, przy zachowaniu mocy obliczeniowej procesorów DSP, układy 56F8xx doskonale nadają się do budowy takich urządzeń jak: tachometry, reduktory szumu, testery kabli, kompresory, systemy HVAC (*Heating, Ventilating and Air Conditioning*), systemy zdalnego monitorowania, sterowniki pomp, wentylatorów, itp. Można go również z powodzeniem stosować we wszelkich urządzeniach sterowanych głosem, aparatach zgłoszeniowych, modemach działających w sieci energetycznej, systemach alarmowych itp. Na podstawie powyższych przykładów można wnioskować, że aplikacje DSP muszą być zdolne do pracy w czasie rzeczywistym z sygnałami analogowymi o szerokim widmie częstotliwości. Od strony sprzętowej narzuca to konieczność implementowania w strukturze procesorów wielokanałowych przetworników analogowo-cyfrowych i cyfrowo-analogowych wysokiej jakości, wydajnej magistrali wewnętrznej i niekiedy specjalizowanych bloków wspomagających. Jednostka centralna procesora DSP musi być zdolna do wykonywania szybkich operacji dodawania, odejmowania, a przede wszystkim dużo wolniejszego na ogół mnożenia i dzielenia. Jest to niezbędne do sprawnego korzystania z często wykorzystywanymi w praktyce, złożonymi strukturami danych umieszczanych w wielowymiarowych tablicach. Aplikacje DSP w sposób czasami dość brutalny ujawniają związek pomiędzy reprezentacją danego sygnału w dziedzinie czasu i częstotliwości. Przykładowo cyfrowe przetworzenie stereofonicznego sygnału akustycznego o częstotliwości 20 kHz wymaga od procesora wydajności rzędu setek MIPS (operacje są prowadzone w dziedzinie czasu). Związek procesorów DSP z sygnałami analogowymi jest czymś naturalnym, lecz w praktyce mamy również do czynienia z zastosowaniem DSP w systemach, w których występują jedynie sygnały cyfrowe. Przykładem może być grafika komputerowa wykorzystywana w programach prognozowania pogody, symulacjach, modelowaniu itp.

Wszystkie powyższe wymagania dotyczące budowy procesora DSP zostały uwzględnione przy projektowaniu procesorów rodziny 56F8xx. Powróćmy więc do układu 56F801 i zobaczmy na jego przykładzie, jak te założenia zrealizowano przez Motorolę.

Schemat blokowy procesora przedstawiono na **rys. 1**. Jego wydajność dochodzi do 30 MIPS przy taktowaniu zegarem o częstotliwości 80 MHz. Jednostkę centralną zoptymalizowano dla programów pisanych w języku C. Podczas wykonywania rozkazów wymiana danych pomiędzy poszczególnymi blokami funkcjonalnymi jest realizowana przez trzy wewnętrzne szyny adresowe i cztery szy-



Rys. 1

Trochę DSP, trochę mikrokontroler

Mikroprocesory 56F8xx firmy Motorola, ze względu na specyficzną architekturę, można zakwalifikować zarówno do grupy DSP jak i mikrokontrolerów uniwersalnych.

ny danych. Wszystkie rodzaje pamięci mają organizację 16-bitową, ale magistrala danych i jednostka centralna są 32-bitowe. Głębokość stosu jest ograniczona jedynie wielkością dostępnej pamięci RAM. Program jest zapisywany w pamięci Flash o wielkości 8 ksłów, a także w wydzielonym obszarze pamięci RAM (1 ksłów). Ponadto dostępne są pamięci: Boot Flash (2 ksłów), Data Flash (2 ksłów) i Data RAM (1 ksłów). Jednostka centralna obsługuje 14 trybów adresowania, potrafi także wykonać w pojedynczym takcie zegarowym operację podwójnego przesunięcia równoległego. Funkcje sterowania wspomagane są sprzętowymi blokami: 6-kanałowego PWM, dwoma 4-kanałowymi, 12-bitowymi przetwornikami ADC, interfejsem SCI (*Serial Communications Interface*), SPI (*Serial Peripherial Interface*) oraz poczwórnym timerem ogólnego zastosowania. W strukturze zaimplementowano specjalny blok wspomagający sprzętowo realizację pętli DO...REP. Wbudowano też wewnętrzny oscylator współpracujący z pętlą PLL. Użytkownik ma do dyspozycji 11 linii ogólnego zastosowania. Sprzętowe uruchamianie programów jest możliwe dzięki interfejsowi JTAG/OnCE. W układzie uwzględniono również kilka bloków pozwalających na maksymalne uproszczenie aplikacji. Są to m.in.: wewnętrzny stabilizator napięcia zasilającego, dostarczający również napięć zasilających dla zewnętrznych układów analogowych i cyfrowych, układ zarządzania mocą, układ korekcji zniekształceń sygnału PWM i układ *brown-out* generujący przerwanie w przypadku pojawienia się problemów z zasilaniem. Procesor 56F801 jest zasilany pojedynczym napięciem 3,3 V, ale jego wejścia i wyjścia tolerują 5-woltowe sygnały TTL. Może być ustawiony w stany *Wait* i *Stop*, pozwalające ograniczyć pobór energii.

Płytka ewaluacyjna 56F800 DEMO

Za pomocą opisywanego zestawu można stosunkowo łatwo wejść w świat cyfrowego przetwarzania sygnałów. Dziedzina ta kojarzy się z koniecznością posiadania dość obszernej wiedzy z zakresu matematyki wyższej, a także umiejętności postrzegania pewnych zjawisk w nieco inny sposób niż czynią to zwykli „mikroprocesorowcy". Przyznać trzeba, że coś w tym jest, nie sądzę bowiem, żeby przeciętny użytkownik „atmelków" potrafił zrealizować np. filtr cyfrowy o zadanych parametrach, nawet przy założeniu, że jego mikrokontrolerek "udźwignąłby" taki problem pod względem wydajności obliczeniowej. Nie wspomnę już implementacji szybkiej transformaty Fouriera (FFT), czy realizacji sterownika 3-fazowego silnika o przełączanej reluktancji, którego teoria we fragmencie jest przedstawiona w dokumencie PDF dołączonym do prezentowanego zestawu (**rys. 2**).

Na płytce ewaluacyjnej 56F800 DEMO zmieszczono wszystkie elementy niezbędne do przeprowadzenia kilku interesujących ćwiczeń zapoznających z techniką programowania układów DSP Motoroli. Oprócz procesora są więc tu uwzględnione również takie elementy jak: mikrofon wraz z odpowiednim wzmacniaczem, 10 diod LED, specjalny potencjometr wykorzystywany w eksperymentach, przyciski zerowania systemu i symulacji przerwań. Uwzględniono też wyprowadzenia portu SPI, timera/układu PWM, przetwornika analogowo-cyfrowego, interfejsu JTAG/OnCE oraz RS232. Do wszystkich

wyżej wymienionych wejść/wyjść nie wlutowano na płytce gniazd (są tylko punkty lutownicze). Jedyne gniazda w jakie jest fabrycznie wyposażona płytka to *Host JTAG* i zasilające. Na płytce nie ma również układu dopasowania poziomów dla interfejsu RS232 (przygotowano miejsce pod SP/MAX3232 i kondensatory 0,1µF). Płytka 56F800 DEMO może być zasilana napięciem od +7 do +15 VDC, np. z zasilacza wtyczkowego dołączonego zresztą do zestawu. Wewnętrzne stabilizatory zapewniają wymagane napięcia: +3,3 V oraz +5 V. Sygnał z wewnętrznego mikrofonu po przepuszczeniu przez filtr dolnoprzepustowy o częstotliwości granicznej 4000 Hz trafia na wejście ANA2 przetwornika A/C. Potencjometr użytkownika jest dołączony pomiędzy masę i napięcie +3,3 V. Z jego suwaka można podawać regulowane napięcie na wejście ANA6 DAC-a. LED-y są dołączone do portów procesora, które jak zwykle oprócz zastosowań ogólnych najczęściej pełnią jeszcze funkcje dodatkowe. Należy to uwzględnić podczas planowania własnych ćwiczeń.

Rejestracja SDK

Aby skorzystać z bogactwa dokumentacji dostępnej w SDK wymagane jest zarejestrowanie się na internetowej stronie Motoroli. Można to zrobić bez obaw, operacja jest całkowicie bezpłatna. W wyniku rejestracji dostajemy drogą e-mailową indywidualny klucz umożliwiający korzystanie z *Software Development Kit*.

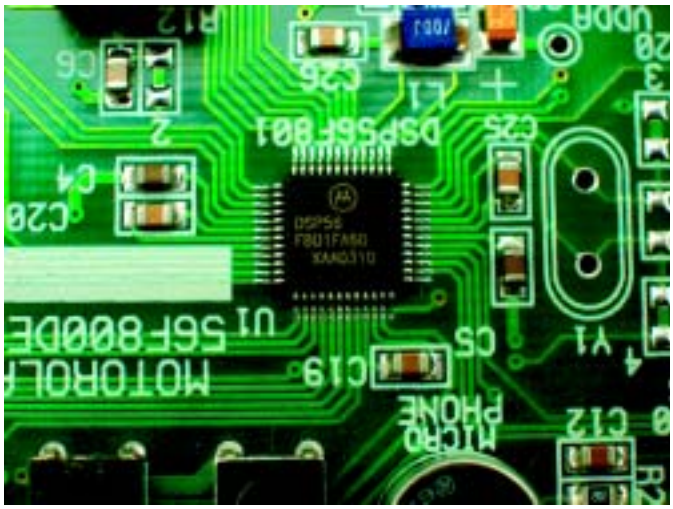
Ćwiczenia

Dołączony do zestawu CD-ROM zawiera kilka interesujących ćwiczeń wprowadzających w środowisko IDE - CodeWarrior, w którym będą uruchamiane projekty i zapoznających z możliwościami płytki ewaluacyjnej i samego procesora. Są one dobrze opisane w plikach PDF, a ich pełniejsze poznanie jest możliwe dzięki prezentacjom *vmf*. Animacje w sposób bardzo wyrazisty objaśniają krok, po kroku, jakie czynności należy wykonać, aby stworzyć dany projekt i uruchomić program w nim zawarty.

Ćwiczenie 1 jest w całości poświęcone tworzeniu projektu. Poznajemy zasady zarządzania plikami projektu, ich tworzenia, kasowania, kopiowania, grupowania, itp. Uczymy się poddawać edycji poszczególne pliki źródłowe. Pod koniec lekcji umiemy już kompilować i debugować nasz program - podglądamy zawartość zmiennych, korzystamy z pułapek.

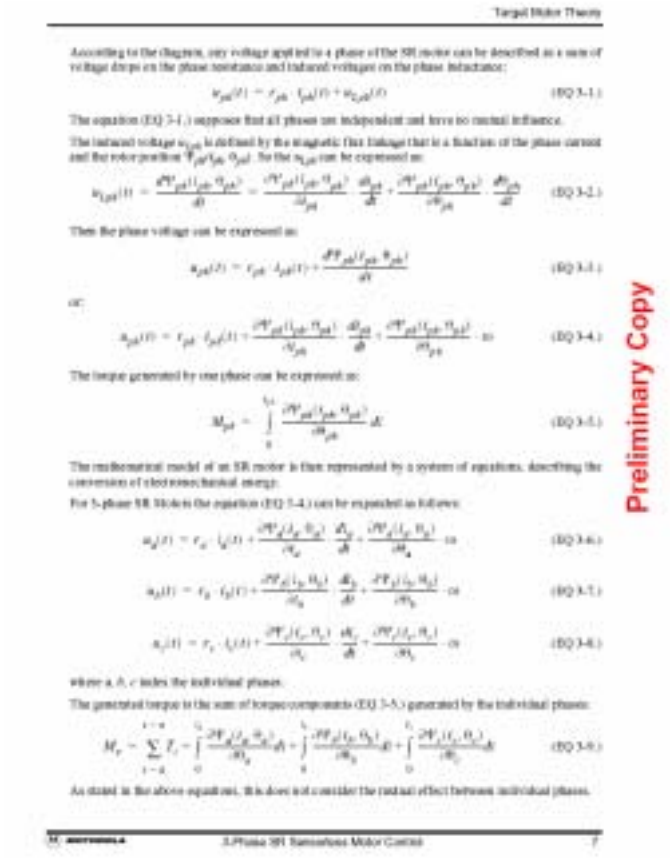
W ćwiczeniu 2. postawiono bardzo konkretne zadanie - musimy napisać program wyszukujący najmniejszą liczbę umieszczoną w tablicy. W tym celu uczymy się korzystać z systemu pomocy oraz biblioteki funkcji udostępnionych przez SDK.

Typ	Wewnętrzna pamięć Xdata (RAM) [B]	Pamięć danych SRAM [kB]	Pamięć danych Flash [B]	Pamięć programu Flash [kB]	Interfejsy szeregowe	Liczba timerów	Wydajność DSP [MMAC]	Szerokość wewnętrznej magistrali danych [b]	Szerokość zewnętrznej magistrali danych [b]	Liczba kanałów A/C	Rozdzielczość przetwornika A/C [b]	Liczba kanałów PWM	Liczba programowanych linii I/O
56F8322	8192		8192	32768	FlexCAN, SCI, SPI	8	60	32		6	12	6	36
56F8323	8192		8192	32768	FlexCAN, SCI, SPI	8	60	32		8	12	6	46
56F8345	8192		8192	131072	FlexCAN, SCI, SPI	16	60	32		16	12	6	102
56F8346	8192		8192	131072	FlexCAN, SCI, SPI	16	60	32	16	16	12	6	118
56F8356	16384		8192	262144	FlexCAN, SCI, SPI	16	60	32	16	16	12	6	118
56F8357	16384		8192	262144	FlexCAN, SCI, SPI	16	60	32	16	16	12	6	133
56F801	2	2	4096	8192	SCI, SPI	8	40	16	16	4	12	6	11
56F801 FA60			4096	8192		8	30						11
56F802			2048	8192		8	40						8
56F802 TA60			2048	8192		8	30						8
56F803	4	1	4096	32256	SCI, SPI	16	40	16	16	4	12	6	16
56F805	4	1	4096	32256	SCI, SPI	16	40	16	16	5	12	12	32
56F807	8	4	8192	61440	SCI, SPI	16	40	16	16	4	12	12	32
56F826	8	2	2048	32256	SCI, SPI, SSI	4	40	16	16				46
56F827	8	2	4096	65536	SCI, SPI, SSI	4	40	16	16	10	12		64
56852	8	12			ISSI, SCI, SPI	4	60, 120	16	16				11, 36
56853	8	24			ESSI, SCI, SPI	8	120	16	16				41
56854	32	32			ESSI, SCI, SPI	8	120	16	16				41
56855	48	48			ESSI, SCI	8	120	16	16				18
56857	48	80			ESSI, SCI, SPI	8	120	16	16				47
56858	48	80			ESSI, SCI, SPI	8	120						



W ćwiczeniu 3. nabieramy coraz większej chęci do zabawy z płytką ewaluacyjną. Nasza wiedza i umiejętności są już wystarczające do zapalania i gaszenia LED-ów i to przy wykorzystaniu systemu przerwań. Przy okazji tak banalnego z pozoru zadania, poruszane są problemy statycznego i dynamicznego instalowania procedur obsługi przerwań - ISR (*Interrupt Service Routine*), a także priorytetowości i zagnieżdżania przerwań. Debugowanie programów z przerwaniami wymaga również pewnej wprawy, której nabieramy podczas ćwiczeń. Tak więc, mimo prostego tematu, lekcja pewnie tym razem przeciągnie się na dwie godziny.

Kolejne ćwiczenie to już poziom niemal ekspercki. Uczymy się tworzyć własne biblioteki i oczywiście korzystać z nich w swoich późniejszych projektach. Czynimy to na bazie źródeł asemblerowych.



Rys. 2

W ćwiczeniu 5. powracamy do sprzętu i znowu bawimy się LED-ami, tym razem wykorzystując timery i mikroprzełączniki umieszczone na płytce. Budujemy nawet zegar czasu rzeczywistego. Przy okazji mamy sporo plików źródłowych do analizy, a jak wiadomo możliwość podejrzenia „jak to robią inni" jest jedną z lepszych metod nauki.

W ostatnim już ćwiczeniu robimy próby z uruchamianiem programu z Flasha procesora, a nie z zewnętrznej pamięci RAM, co czyniliśmy do tej pory.

No dobrze, ale przecież powyższe ćwiczenia nie mają nic wspólnego z DSP. Niby tak, ale ich przeznaczeniem nie było zapoznanie nas z tą tematyką, a jedynie zaznajomienie z płytką testową zawartą w zestawie. Jeśli chcemy dowiedzieć się czegoś więcej na temat samego DSP, to musimy sięgnąć po dokumenty dostarczone razem z SDK. Znajduje się tu pokaźny katalog z plikami PDF, w których opisane są funkcje przydatne do realizacji aplikacji DSP. Zamieszczone tam przykładowe listingi dostarczają nam prawie gotowego materiału do zaprojektowania własnego systemu. Przykładowo, dostępne są biblioteki przydatne do tworzenia urządzeń takich jak: układ rozpoznawania głosu, eliminator echa akustycznego (służący np. do minimalizacji sprzężenia między mikrofonem a słuchawką objawiającego się przeraźliwym piskiem), detektor DTMF, CPT i MFCR2, różnego rodzaju kodeki, modemy, systemy kryptograficzne, sterowniki niemal wszystkich typów silników od DC do 3-fazowych.

Tematy są dość frapujące, więc do dzieła.

Jarosław Doliński, EP
jaroslaw.dolinski@ep.com.pl