



wyższą odpornością na zewnętrzne sygnały zakłócające. Można również przypuszczać, że powinien być również dobrze zabezpieczony i „uzbrojony” w mechanizmy umożliwiające reakcję CPU mikrokontrolera na różne nietypowe sytuacje.

## Architektura

Mikrokontroler ST7 ma 6 podstawowych rejestrów (rys. 1). Są to:

- 8-bitowy, wyróżniony rejestr specjalny, na którym wykonywane są operacje arytmetyczne i logiczne - akumulator (A),
- dwa 8-bitowe rejestry indeksowe X i Y,
- 16-bitowy wskaźnik stosu SP,
- 16-bitowy licznik rozkazów PC,
- 8-bitowy rejestr kodów warunkowych CCR.

Dwa rejestry indeksowe X i Y umożliwiają dostęp do 64 kB pamięci. Oczywiście np. w przypadku ST7LITE nie będzie to miało większego znaczenia, ponieważ nie ma on możliwości obsługi pamięci zewnętrznej, a rozmiar wewnętrznej nie przekracza 8 kB. Rejestry X i Y mogą również chwilowo przechowywać dane. Nie są one używane bez wyraźnej intencji programisty.

Wskaźnik stosu SP jest rejestrem adresowym wskazującym obszar pamięci RAM (zwykle na końcu wolnego obszaru). Ten zarezerwowany obszar pamięci ma za zadanie przechowywać dane przy obsłudze procedur obsługi przerwania oraz przechowywać adres powrotu przy wywołaniu podprogramów oraz procedur obsługi przerwania. Za jego pośrednictwem można również przekazywać dane do innych procedur. Przy zapamiętywaniu danych na stosie wskaźnik jest zmniejszany o 1, przy pobieraniu danych ze stosu wskaźnik jest zwiększany o 1.

Licznik rozkazów jest rejestrem, który zawiera adres następnego rozkazu do wykonania lub argumentu rozkazu. Po załadowaniu rozkazu zawartość licznika jest automatycznie zwiększana o 1 i adresuje on następną komórkę pamięci. Licznik rozkazów jest modyfikowany przez rozkazy skoków, wywołań podprogramów, a więc również procedury obsługi przerwania. Po restarcie mikrokontrolera (niezależnie od źródła sygnału *reset*) do licznika rozkazów jest ładowany adres wskazywany przez wektor restartu leżący pod adresem 0xFFFF:0xFFFF

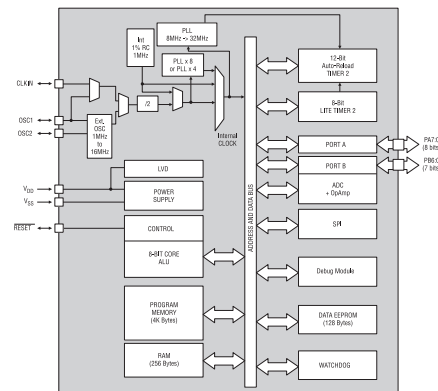
i od niego zaczyna się wykonywanie programu.

Rejestr znaczników CCR zawiera flagi stanów, których ustawienia są wynikiem ostatnio realizowanej instrukcji. Rejestr CCR, jako że ma pojemność pojedynczego bajtu, może zawierać do 8 flag. Aktualnie zaimplementowane są: flaga przeniesienia połówkowego H, flaga zezwolenia na obsługę przerwania I, flaga wyniku negatywnego N oraz zerowego Z, flaga przeniesienia C.

## ST7FLITE19

Seria ST7LITE pojawiła się na rynku jako odpowiedź firmy ST Microelectronics na zapotrzebowanie klientów na tani mikrokontroler przeznaczony do prostych aplikacji sterujących. Nazwa jest trochę myląca. W praktyce bowiem różnica polega nie tyle na zubożeniu mikrokontrolera o pewne funkcje, ile na umieszczeniu go w małej obudowie (np. o 20-nóżkowej) i wyposażeniu w mniejszą pamięć programu i danych. Nie mniej jednak, w strukturze mikrokontrolera pozostały liczne układy peryferyjne (np. watchdog, generatory PWM, pamięć EEPROM i inne), a CPU nie zostało w żaden sposób zubożone. Wszystkie mikrokontrolery ST7 wykonane są w oparciu o ten sam rdzeń i realizują tę samą, rozszerzoną listę rozkazów. Podstawą cyklu maszynowego jest częstotliwość zegarowa podzielona przez 2. Jako reprezentanta rodziny postanowiłem wybrać ST7FLITE19, który jest moim zdaniem, bardzo ciekawym mikrokontrolerem i co najważniejsze: prawie wszystkie narzędzia niezbędne do napisania dla niego aplikacji dostępne są za darmo.

ST7FLITE19 (rys. 2) wyposażony jest w 4 kB pamięci Flash, 128 bajtów pamięci EEPROM oraz 256 bajtów pamięci RAM. Na zewnątrz mikrokontrolera dostępnych jest 15 dwukierunkowych linii wejścia/wyjścia, które oprócz ogólnego przeznaczenia pełnią też funkcje specjalne. Mogą być na przykład wyjściem generatora PWM lub interfejsu SPI. Trzeba mieć to na uwadze projektując własną aplikację. Siedem linii portu A (PA0...5, PA7) jest przystosowanych do obciążenia dużym, jak na mikrokontroler prądem. Dokumentacja techniczna podaje, że maksymalna wartość prądu, która jeszcze nie powinna zniszczyć portu mikrokontrolera, a którą można pobrać z jego wyjścia, wynosi aż 50 mA!



Rys. 2. Schemat blokowy mikrokontrolera ST7FLITE19

Wszystkim, którzy chcieliby użyć tej własności ST7 chciałbym jednak przypomnieć o maksymalnej, dopuszczalnej mocy strat. Raczej nie da się w sposób ciągły obciążać wszystkich portów aż tak wysokim prądem. Po prostu struktura mikrokontrolera przegrzeje się i ulegnie uszkodzeniu. Można jednak bez przeszkód zasilac chociażby wyświetlacz LED lub stopień wejściowy drivera mocy PWM.

Linie portu PB poprzez wewnętrzny multiplexer analogowy dołączone są do wejścia przetwornika analogowo-cyfrowego. Przetwornik ten nie posiada wejścia zewnętrznego napięcia odniesienia, więc wszystkie pomiary są wykonywane w odniesieniu do napięcia zasilającego. Rozdzielczość przetwornika jest równa 10 bitów, co odpowiada wartości ziarna równej ok. 4,9 mV dla napięcia zasilającego równego 5 V.

Na wejściu przetwornika znajduje się wzmacniacz o stałym wzmocnieniu równym 8 V/V, który może być załączony przez ustawienie bitu ADON w rejestrze ADCCSR. Po załączeniu tego wzmacniacza staje się możliwy pomiar nawet niewielkich zmian wielkości elektrycznych, na przykład zmian napięcia na czujniku temperatury, ponieważ w tym trybie napięcia w zakresie od 0 do 430 mV mogą być mierzone z rozdzielczością ok. 0,61 mV (przy zasilaniu 5 V). Tu jedna ważna uwaga. W niektórych materiałach na temat ST7 pojawia się wzmianka, że rozdzielczość przetwornika wynosi przy załączonym wzmacniaczu 13 bitów. Jest to prawda tylko w odniesieniu do napięcia zasilania, a nie do napięcia mierzonego! Jeśli jest włączony wzmacniacz, to nie można do wejścia analogowego przyłożyć napięcia o wartości równej napięciu zasilania. W tym trybie

maksymalne napięcie wejściowe jest 8-krotnie mniejsze, od napięcia zasilającego. Oczywiście – jak wspominałem wcześniej – rozdzielczość przetwornika w takim przypadku to 0,61 V przy zasilaniu mikrokontrolera napięciem 5 V. Odnosząc to do tej wartości napięcia zasilającego otrzymamy wspomniane 13 bitów rozdzielczości (5 V/0,61 mV 8192 tj. 12 bitów). Jest to jednak tylko rozdzielczość pozorną, ponieważ właściwości przetwornika nie zmieniają się: nadal udostępnia on słowo o długości 10 bitów.

Porty wejścia/wyjścia ST7FLITE19 posiadają bardzo ciekawą cechę. Otóż każda z linii może być skonfigurowana w taki sposób, że może uruchamiać procedurę obsługi przerwania zewnętrznego. Nie ma tu wejść oznaczonych INT0, INT1 czy INT2. Czasami jest to cecha bardzo, bardzo użyteczna w aplikacji.

ST7FLITE19 posiada 12 wektorów przerw w wewnętrznych, jeśli uwzględnić przerwy generowane przez instrukcje *TRAP* i *RESET*. Pozostałymi źródłami przerw mogą być układy wbudowanych w strukturę timerów (tu dostępne są liczne warunki generowania przerw), przerwy zewnętrzne, przetwornik A/C, interfejs SPI, również funkcja *AUTOWAKEUP* może generować przerwy. Używając przerw w ST7, a zwłaszcza przerw zewnętrznych, trzeba zachować uwagę. Myślę, że sposób używania przerw najlepiej opiszą przykłady aplikacji w dalszych częściach artykułu.

Mikrokontroler ST7FLITE19 posiada możliwość programowania w układzie. Przesyłanie i zapis programu w pamięci Flash lub EEPROM odbywa się za pośrednictwem interfejsu ICC. Jest to interfejs szeregowy, do transmisji wykorzystywane są 2 linie: ICCDATA i ICCCLK. Dodatkowo podłączenie wymaga doprowadzenie RESET oraz czasami, w sytuacjach gdy na przykład używany jest wewnętrzny oscylator RC, należy podać sygnał zegarowy na doprowadzenie PB4, będące wejściem zewnętrznego zegara. Do zapisania programu w pamięci procesora najlepiej posłużyć się firmowym programatorem sprzedawanym pod oznaczeniem *ST7 Flash Stick*. Jest on dostępny nie tylko w ofercie firmy STM, ale również innych producentów akcesoriów.

ST7FLITE19 posiada możliwość pracy z różnymi źródłami sygnału zegarowego:

- z rezonatorem kwarcowym o częstotliwości od 32768 Hz (kwarc zegarkowy) do 16 MHz,
- z wbudowanym, kalibrowanym generatorem RC (nie wymaga żadnych elementów zewnętrznych) o częstotliwości 1 MHz,
- z zewnętrznym generatorem sygnału zegarowego.

Ze względu na wbudowane w strukturę generatory PLL, na pierwszy „rzut oka” budowa generatora sygnału zegarowego wygląda na bardzo skomplikowaną. Nie mniej jednak, w porównaniu z konkurencją, nastawy rejestru opcji są bardzo proste do wykonania i nie następują żadnych trudności. Układ ST7FLITE19 oprócz „głównego” źródła sygnału zegarowego, posiada również źródła alternatywne, różniące się znacznie częstotliwością od częstotliwości podstawowego zegara CPU, a umożliwiające pracę timerów z częstotliwością inną niż ta, którą taktowane jest CPU. Jest to funkcja użyteczna i przydatna zwłaszcza wtedy, gdy wykorzystuje się generator PWM. Te źródła alternatywne, to dwa generatory zsynchronizowane z częstotliwością podstawową przy pomocy pętli PLL, mogące generować częstotliwość będącą wielokrotnością x4 lub x8 częstotliwości podstawowej. Generatory są załączane przy pomocy bitów opcji. Dla przykładu wyborem źródła sygnału zegarowego dla timera AT steruje nastawa rejestru ATCSR (bity CK0...CK1).

Częstotliwość generatora zegarowego jest wewnętrznie dzielona przez 2, co dla kwarcu 16 MHz daje cykl maszynowy trwający 250 ns. Uwaga: przez producenta ustawiany jest jako domyślny generator RC. W celu przełączenia się na zewnętrzny kwarc, należy ustawić odpowiednie bity opcji przy pomocy programatora! Bity te uwzględniają również częstotliwość dołączonego kwarcu dla doboru odpowiedniego czasu startu mikrokontrolera. Jeśli o tym zapomnimy, możemy mieć problem ze skonfigurowaniem generatora PWM. Okazuje się, że nie działa on zgodnie z naszymi oczekiwaniami. Będziemy szukać błędu w ustawieniu parametrów PWM, podczas gdy mimo

dołączonego rezonatora kwarcowego pracuje wewnętrzny generator RC i to niego zależy przebieg sygnału na wyjściu PWM.

W strukturę mikrokontrolera ST7FLITE19 wbudowano 4 prawie niezależne generatory PWM. Generowana przez nie częstotliwość jest taka sama, ale współczynnik wypełnienia może być indywidualnie ustawiany dla każdego z nich. Oprócz tego, każdy z sygnałów PWM może być indywidualnie negowany, co daje możliwość odpowiedniego (najczęściej dla stopnia mocy) doboru fazy sygnału lub współczynnika wypełnienia przebiegu wyjściowego. Programowanie PWM jest przejrzyste i nie następują żadnych trudności nawet początkującym.

## ST7 Assembler

Przygotowując front dla następnych artykułów zawierających przykłady programów użytkowych oraz wykorzystania peryferii, nie sposób pominąć listy rozkazów assemblera. Lista rozkazów ST7 obejmuje 63 instrukcje. Ich zestawienie, wraz z krótkim opisem funkcjonalnym, podano w **tab. 1**. Praktyczne użycie poszczególnych instrukcji wyjaśniają przykłady programowania.

## Dostępne narzędzia

Do pisania i uruchamiania programów w assemblerze najlepiej będzie się nadawał firmowy *ST7 Visual Develop*. Jest to kompletne narzędzie zawierające kompilator, linker, debugger oraz program do obsługi *ST7 Flash Stick*. Program można pobrać ze strony producenta: <http://www.st.com>. IDE to współpracuje z kompilatorami języka C firm Cosmic (<http://www.cosmic-software.com/st7.php>) i Metrowerks (<http://www.metrowerks.com>), umożliwiając symulację programu. Dostępny jest również kompilator języka C znanej z produktów przeznaczonych dla 8051 i ARM, firmy Raisonance (<http://www.raisonance.com>). Wszystkie wyżej wymienione kompilatory są dostępne w wersjach demonstracyjnych, umożliwiając generowanie do 4 kB kodu wynikowego. Można więc powiedzieć, że narzędzia przeznaczone do programowania ST7FLITE19 dostępne są za darmo.

Opis programatora ISP i zestawu startowego opublikujemy w EP7/2006.

**Jacek Bogusz, EP**  
jacek.bogusz@ep.com.pl