Wstęp

Popularne procesory firmy Microchip z rodziny PIC12 (w obudowach 8-nóżkowych) zyskały opinię tanich i łatwych w obsłudze, lecz o małych możliwościach i ze skromnym zestawem układów peryferyjnych. Przeglądając notę katalogową procesora PIC12F675, Czytelnik musi zrewidować swoją opinię o „małych" procesorach PIC. Za cenę niewiele przekraczającą 1,5 $ otrzymujemy procesor z wielokrotnie programowalną pamięcią programu (Flash), 4-kanałowym, 10-bitowym przetwornikiem A/C, komparatorem analogowym, programowanym źródłem napięcia odniesienia, dwoma licznikami (8- i 16-bitowy), wewnętrzną pamięcią EEPROM, układem przerwań i wieloma innymi układami, spotykanymi dotąd w większych i droższych procesorach. Zastosowanie tego procesora pozwala na budowanie interesujących układów o dużych możliwościach i niskiej cenie, a więc konkurencyjnych wobec rozwiązań wymagających zastosowania procesora i zewnętrznych układów peryferyjnych.

Procesor, pomimo nazwy PIC12 sugerującej pokrewieństwo z rodziną *Base-Line*, ma architekturę procesorów rodziny *Mid-Range*. Ponadto zastosowano w nim wiele udogodnień i rozszerzeń funkcjonalnych obejmujących m.in. możliwość podłączania rezystorów podciągających do poszczególnych wejść, czy możliwość definiowania przerwań od zmiany stanu na poszczególnych liniach I/O. Brak takiej możliwości jest wyraźnie odczuwalny w procesorach z rodziny *Mid-Range*. Uproszczono też sposób konfigurowania linii I/O współpracujących z przetwornikiem analogowo-cyfrowym. Interesującym rozwiązaniem jest możliwość bramkowania licznika 1 za pomocą sygnału zewnętrznego, co rozszerza możliwości tego 16-bitowego licznika z preskalerem. Jeśli dodamy do tego możliwość taktowania procesora z częstotliwością do 20 MHz, to określenie układu PIC12F675 mianem „mały" procesor wydaje się bardzo nieodpowiednie.

W książce stosunkowo wiele uwagi poświęcono na omówienie architektury procesora, która w dużej mierze decyduje o efektywnym wykorzystaniu jego możliwości. Zapewne wielu Czytelników zna inne procesory PIC, dlatego starano się specjalnie wyróżnić nowe rozwiązania i różnice w stosunku do spotykanych w rodzinie *Mid-Range*. Być może pozwoli to na szybsze przygotowanie własnego programu czy też modyfikację kodu napisanego dla innego procesora.

Do najważniejszych cech procesorów PIC12F675 i PIC12F629 należy zaliczyć:

- 8-nóżkową obudowę z sześcioma uniwersalnymi liniami I/O i dwoma wyprowadzeniami zasilania (V_{DD} i V_{SS});
- pamięć programu o pojemności 1024 słów 14-bitowych;
- pamięć danych RAM złożoną z 64 rejestrów 8-bitowych;
- nieulotną pamięć EEPROM o pojemności 128 słów 8-bitowych;
- dwa liczniki (timery): 8-bitowy licznik 0 i 16-bitowy licznik 1;

Od czasu rozpoczęcia pracy nad książką pojawiło się kilka nowych procesorów znacznie rozszerzając możliwości rodziny procesorów w obudowach z 6, 8 i 14 wyprowadzeniami. Więcej informacji można znaleźć w dodatku A.

- 10-bitowy przetwornik analogowo-cyfrowy (tylko w PIC12F675);
- komparator analogowy wraz z programowanym źródłem napięcia odniesienia;
- rozbudowany układ generatora zegarowego zawierający: wewnętrzny generator RC, ekonomiczny generator RC i kwarcowy generator: 32 kHz...20 MHz;
- rozbudowany układ zerowania zawierający m.in. obwód zerowania po załączeniu zasilania i obwód zerowania przy spadku napięcia zasilania;
- szeroki zakres napięcia zasilania: 2...5,5 V;
- możliwość programowania po zamontowaniu w systemie.

Budowa procesorów PIC opiera się na architekturze harvardzkiej, tzn. mają oddzieloną pamięć danych od pamięci programu i dwie oddzielne magistrale o różnej szerokości słowa. Dla pamięci programu są stosowane słowa 14-bitowe, natomiast pamięć danych jest 8-bitowa. Dlatego procesory PIC są zaliczane do grupy procesorów 8-bitowych.

Dzięki tak długim słowom pamięci programu w pojedynczej instrukcji może być zawarty zarówno kod, jak i argument (argumenty) operacji, wskutek czego liczba słów możliwych do zapamiętania w pamięci odpowiada liczbie możliwych do wykonania instrukcji. W przypadku architektury tradycyjnej jedna komórka zawiera zwykle kod operacji, a kolejne – argumenty. Dlatego kod wynikowy takich samych programów jest zazwyczaj dla procesorów PIC nieco krótszy.

W procesorach zastosowano potokowy system pracy (*pipeline*), w którym procesor podczas wykonywania rozkazu pobiera z pamięci następny, co sprawia wrażenie, że wykonanie rozkazu trwa tylko jeden cykl maszynowy.

Wszystkie rejestry specjalne procesora – *SFR* (*Special Function Register*) – odwzorowane są w pamięci RAM i są dostępne tak samo jak inne komórki pamięci, tzn. poprzez adresowanie bezpośrednie lub pośrednie. Pomysł ten nazwano koncepcją tablicy rejestrów (*Register File Concept*).

Inną interesującą cechą procesorów PIC jest to, że w każdym słowie pamięci programu może być umieszczony zarówno argument, jak i wynik operacji, tradycyjnie przesyłany do akumulatora. Jego rolę pełni w tych procesorach rejestr roboczy oznaczony jako W (*Working Register*). Brak wyjątków, obecnych zazwyczaj w takich przypadkach, bardzo upraszcza programowanie procesora. Cechę tę producent nazywa *symetrią* lub *ortogonalnością* instrukcji.

Skrócony opis instrukcji

Podczas opisu budowy i działania procesora będziemy często posługiwać się przykładami, dlatego już na początku warto, nawet pobieżnie, zaznajomić się z listą instrukcji procesorów PIC zawierającą tylko 35 poleceń. Instrukcje podano w uporządkowaniu funkcjonalnym, odbiegającym od alfabetycznego, spotykanego w dokumentacji procesorów. Każda instrukcja dla procesorów z rodziny *Mid-Range* ma postać 14-bitowego słowa i dzieli się na pole określające kod instrukcji i pole zawierające jeden lub dwa argumenty.

Wszystkie instrukcje są wykonywane w jednym cyklu maszynowym, z wyjątkiem instrukcji skoków, które są wykonywane w dwóch cyklach maszynowych.

Mnemonik	Arg.	Opis w jęz. ang.	Opis
NOP		No Operation	Cykl pusty
CLRW		Clear W	Zeruj W
CLRF	f	Clear F	Zeruj F
COMF	f, d	Complement F	Zaneguj F, wynik prześlij do d
MOVF	f, d	Move F to d	Prześlij F do d
MOVWF	f	Move F to W	Prześlij W do F
MOVLW	k	Move k to W	Prześlij k do W
ADDWF	f, d	Add W to F	Dodaj W do F, wynik prześlij do d
ADDLW	k	Add k to W	Dodaj k do W
INCF	f, d	Increment F	Zwiększ F o 1, wynik prześlij do d
SUBWF	f, d	Subtract W from F	Odejmij W od F, wynik prześlij do d
SUBLW	k	Subtract k from W	Odejmij k od W
DECF	f, d	Decrement F	Zmniejsz F o 1, wynik prześlij do d
ANDWF	f, d	AND W and F	Iloczyn logiczny W i F, wynik prześlij do d
ANDLW	k	AND k and W	Iloczyn logiczny k i W
IORWF	f, d	OR W with F	Suma logiczna W i F, wynik prześlij do d
IORLW	k	OR k with W	Suma logiczna k i W
XORWF	f, d	XOR W with F	Suma modulo 2 W i F, wynik prześlij do d
XORLW	k	XOR k with W	Suma modulo 2 k i W
RLF	f, d	Rotate Left F	Przesuń F o 1 bit w lewo, wynik prześlij do d
RRF	f, d	Rotate Right F	Przesuń F o 1 bit w prawo, wynik prześlij do d
SWAPF	f, d	Swap nibbles in F	Zamień tetrady w F, wynik prześlij do d
BCF	f, b	Bit Clear F	Wyzeruj bit b w F
BSF	f, b	Bit Set F	Ustaw bit b w F
BTFSC	f, b	Bit Test F, Skip if Clear	Jeśli bit b w F = 0, to omiń następną instrukcję
BTFSS	f, b	Bit Test F, Skip if Set	Jeśli bit b w F = 1, to omiń następną instrukcję
INCFSZ	f, d	Inc. F Skip if Zero	Zwiększ F o 1, jeśli wynik = 0 omiń następną instr.
DECFSZ	f, d	Dec. F Skip if Zero	Zmniejsz F o 1, jeśli wynik = 0 omiń następną instr.
GOTO	s	Go To address s	Skok bezwarunkowy do etykiety s
CALL	s	Call subroutine	Wywołaj podprogram o etykiecie s
RETURN		Return from subroutine	Powrót z podprogramu
RETLW	k	Return with k in W	Powrót ze stałą k w W
RETFIE		Return From Interrupt	Powrót z przerwania
CLRWDT		Clear WDT	Zeruj licznik WDT
SLEEP		Go to standby mode	Przejdź w stan uśpienia

gdzie:

f – 7-bitowy adres rejestru (0...127)
F – zawartość rejestru o adresie f (0...255)
W – zawartość rejestru roboczego (0...255)
d – adres wyniku operacji (0,1):
gdy *d* = 0, wynik operacji jest przesyłany do *W*
gdy *d* = 1, wynik operacji jest przesyłany do *F*
b – numer bitu (0...7)
k – 8-bitowa stała liczbowa (0...255)
s – 11-bitowa etykieta (0...2047)

Uwagi końcowe

W książce przyjęto konwencję pisania programów małymi literami, dlatego w celu uniknięcia błędów należy przed kompilacją ustawić opcję kompilatora *Case sensitivity* na *off*. Pozostałe opcje kompilatora mogą być ustawione jako domyślne – i takie przyjęto w książce. Zrezygnowano również z możliwości użycia polskich liter w komentarzach w kodach źródłowych programów.

Nazwy rejestrów

W książce przyjęto nazwy rejestrów zgodne z opisem podanym w plikach inicjalizacyjnych procesorów. Poniżej przedstawiono nazwy i adresy najważniejszych rejestrów. Pełny opis zamieszczono w dodatku B.

Nazwa rejestru	Adres
INDF	0000h
TMR0	0001h
PCL	0002h
STATUS	0003h
FSR	0004h
GPIO	0005h
PCLATH	000Ah
INTCON	000Bh
PIR1	000Ch
TMR1L	000Eh
TMR1H	000Fh
T1CON	0010h
CMCON	0019h
ADRESH	001Eh
ADCON0	001Fh
OPTION_REG	0081h
TRISIO	0085h
PIE1	008Ch
PCON	008Eh
OSCCAL	0090h
WPU	0095h
IOC	0096h
IOCB	0096h
VRCON	0099h

Autor składa podziękowania panom Klaudiuszowi Gawlikowi i Andzejowi Niedworokowi – studentom Wydziału Automatyki, Elektroniki i Informatyki Politechniki Śląskiej w Gliwicach za pomoc w przygotowaniu i przetestowaniu części programów zawartych w niniejszej książce.