

# Mikrokontrolery z rdzeniem ARM, część 3

*W większości obecnie prezentowanych (nie tylko w EP) projektów królują mikrokontrolery 8-bitowe. Dawniej była to nieśmiertelna rodzina 8051, obecnie panuje moda na AVR-y. Dotychczas użycie 32-bitowych mikrokontrolerów pozostawało w sferze marzeń przeciętnego konstruktora ze względu na wysoką cenę, skomplikowane rozwiązania układowe wymagające prowadzenia na płytkach 32-bitowych magistral systemowych oraz konieczność stosowania dodatkowych układów zewnętrznych. Sytuacja uległa diametralnej zmianie w momencie pojawienia się na rynku mikrokontrolerów z rdzeniem ARM i wbudowaną pamięcią Flash, najpierw firmy Philips LPC21xx i wkrótce później Atmela AT91SAM7, ATM – rodzina STR700 i wielu innych producentów.*

## Słowo stanu procesora (CPSR, SPSR)

Rejestr CPSR zawiera znaczniki wyniku wykonanych operacji arytmetycznych i logicznych, flagi pozwalające na kontrolę przyjmowania przerw, oraz bity ustawień trybu ochrony. Gdy procesor znajduje się w trybie uprzywilejowanym dodatkowo widoczny jest rejestr SPSR (*Saved Program Status Register*), będący kopią rejestru CPSR, utworzoną tuż przed zmianą trybu ochrony na inny. Bitową reprezentację rejestru CPSR przedstawiono na **rys. 6**.

Flagi warunków N, Z, C, V są zmieniane w wyniku operacji arytmetycznych i logicznych. Flagi mogą być także zmieniane za pomocą instrukcji MSR, której wykonanie powoduje przepisanie rejestru R0...R15 do rejestru stanu procesora (CPSR). Rdzeń sprawdza znaczniki flag warunkowych w celu określenia czy instrukcja ma być wykonana czy też nie. W trybie ARM wszystkie instrukcje, zawierają 4-bitowy kod warunkowy i mogą być wykonywane warunkowo w zależności od flag N,Z,C,V, natomiast w trybie THUMB warunkowo mogą być wykonywane tylko instrukcje rozgałęzień. Bity I, F pozwalają na kontrolę przyjmowania przerw. Usta-

wienie bitu I, F powoduje zablokowanie przyjmowania odpowiednio przerwania IRQ i FIQ. Bit T informuje o tym, czy CPU znajduje się w 16-bitowym trybie THUMB (flaga ustawiona) lub w trybie ARM (flaga skasowana). Nie wolno bezpośrednio modyfikować tego bitu, pod groźbą nieokreślonego zachowania CPU. Do zmiany trybu pracy służy specjalna odmiana instrukcji skoku BX. Bity M4, M3, M2, M1, M0 określają tryb ochrony w jakim znajduje się procesor. Nie wszystkie kombinacje stanów są dozwolone i należy uważać, aby używać tylko prawidłowych kombinacji bitów. W **tab. 2** przedstawiono tryby ochrony procesora i wynikające z nich dozwolone ustawienia bitów.

## Wyjątki

Wyjątki działają na bardzo podobnej zasadzie jak przerwania w mikrokontrolerach 8-bitowych, mianowicie gdy zdarzy się sytuacja wyjątkowa mikroprocesor przerywa wykonania bieżącego programu i skacze do podprogramu obsługi wyjątku. Po zakończeniu obsługi wyjątku mikroprocesor powraca do poprzedniego miejsca gdzie zostało przerwane wykonywanie programu. Jedyna różnica polega na tym że wyjątki są generowane nie tylko w reakcji na przerwania, od urządzeń peryferyjnych, ale mogą być wywoływane przez sam mikroprocesor na przykład na skutek wystąpienia odwołania do obszaru pamięci, który nie istnieje. Jest jeszcze jedna różnica, mianowicie tuż przed wejściem do procedury obsługi wyjątku CPU zmienia tryb ochrony z trybu bieżącego na tryb, który jest przypisany do danego wyjątku. Po zakończeniu podprogramu obsługi wyjątku przywracany pierwotny tryb ochrony. Np. gdy mikroprocesor wykonujący program w trybie użytkownika napotka instrukcję przerwania programowego SWI generuje wyjątek Software Interrupt. Wyjątek ten zmienia tryb ochrony na Supervisor, a po zakończeniu podprogramu obsługi wyjątku przywracany jest tryb użytkownika. Podczas gdy zostaje zgłoszony wyjątek CPU wykonuje następujące operacje:

- W rejestrze LR (R14) zapisuje adres następnej instrukcji do wykonania.
- Do rejestru SPSR trybu Rys. 6.

ochrony do którego przypisany jest wyjątek kopiuje zawartość rejestru CPSR.

- Zmienia tryb ochrony na taki jaki jest przypisany do danego wyjątku.
- Do licznika rozkazów (PC) wpisuje adres wektora wyjątku.

Jeśli zgłaszane są wyjątki obsługujące przerwanie CPU ustawia bity F, I co zapobiega ponownemu przyjmowaniu przerw do czasu zakończenia obsługi bieżącego przerwania. O kolejności przyjmowania wyjątków decyduje ich priorytet. Gdy w momencie zgłoszenia wyjątku procesor znajduje się w trybie THUMB, zostaje on automatycznie przełączony w tryb ARM. Po zakończeniu podprogramu obsługi wyjątku należy powrócić do realizacji programu głównego poprzez skopiowanie do PC (R15) zawartości rejestru LR (R14) odejmując odpowiednie przesunięcie. W momencie skopiowania adresu powrotu do licznika rozkazów PC(R15), poprzednia zawartość rejestru CPSR jest odtwarzana automatycznie z rejestru SPSR. W **tab. 3** przedstawiono informacje na temat adresów wektorów wyjątków, priorytetów oraz zalecanej instrukcji kończącej obsługę wyjątku.

Obserwując tablicę wektorów przerw możemy zauważyć, że wektor 0x00000014 jest niewykorzystany. Jest on pozostałością po wcześniejszych implementacjach architektury ARM. W procesorach LPC21xx wektor ten powinien być zapisany sumą kontrolną w kodzie uzupełnienia do dwóch tablicy wektorów przerw. Suma ta jest sprawdzana przez bootloader mikrokontrolera, w celu stwierdzenia poprawności programu zapisanego w pamięci. Warto podkreślić, że maksymalny czas, jaki upływa od zgłoszenia przerwania FIQ do jego obsługi wynosi 27 cykli procesora, co przy zegarze 60 MHz daje czas reakcji na przerwanie równy maksymalnie 450 ns. Czas reakcji na przerwanie IRQ jest podobny, jednak może ono zostać zgłoszone później, jeżeli jest wykonywany inny wyjątek o wyższym priorytecie.



**Tab. 2. Tryby ochrony procesora i wynikające z nich dozwolone ustawienia bitów M4...M9**

Bity M[4..0]	Tryb	Widoczne rejestry trybu THUMB	Widoczne rejestry trybu ARM
10000	User	R0–R7, SP, LR, PC, CPSR	R0–R14, PC, CPSR
10001	FIQ	R0–R7, SP_fiq, LR_fiq, PC, CPSR, SPSR_fiq	R0–R7, R8_fiq–R14_fiq, PC, CPSR, SPSR_fiq
10010	IRQ	R0–R7, SP_irq, LR_irq, PC, CPSR, SPSR_irq	R0–R12, R13_irq–R14_irq, PC, CPSR, SPSR_irq
10011	Supervisor	R0–R7, SP_svc, LR_svc, PC, CPSR, SPSR_svc	R0–R12, R13_svc–R14_svc, PC, CPSR, SPSR_svc
10111	Abort	R0–R7, SP_abt, LR_abt, PC, CPSR, SPSR_abt	R0–R12, R13_abt–R14_abt, PC, CPSR, SPSR_abt
11011	Undefined	R0–R7, SP_und, LR_und, PC, CPSR, SPSR_und	R0–R12, R13_und–R14_und, PC, CPSR, SPSR_und
11111	System	R0–R7, SP, LR, PC, CPSR	R0–R14, PC, CPSR

Za kontrolę przerwania od urządzeń zewnętrznych oraz ich przydzielenie jako przerwanie FIQ lub IRQ w mikrokontrolerach LPC21xx odpowiada wektoryzowany kontroler przerwania (VIC), który zostanie opisany później.

Poszczególne wyjątki zgłaszane są w następujących sytuacjach:

Wyjątek **FIQ (Fast Interrupt)** jest generowany przez sygnał wewnętrzny mikrokontrolera nFIQ pochodzący z kontrolera przerwania VIC. Tryb ochrony FIQ, w który wchodzi CPU w momencie zgłoszenia tego wyjątku, zawiera 7 bankowanych rejestrów, których nie trzeba zapisywać na stosie przy wejściu do procedury obsługi przerwania. Wyjątek FIQ można zablokować w uprzywilejowanym trybie ochrony poprzez ustawienie flagi F w rejestrze CPSR.

Wyjątek **IRQ (Interrupt)** jest generowany przez sygnał wewnętrzny mikrokontrolera nIRQ pochodzący z kontrolera przerwania VIC. Wyjątek FIQ można zablokować w uprzywilejowanym trybie ochrony poprzez ustawienie flagi F w rejestrze CPSR.

Wyjątek **Prefetch Abort** jest generowany, gdy pobierana instrukcja jest nieprawidłowa.

Wyjątek **Data Abort** jest generowany przez instrukcje LDM, STM, LDR, STR, SWP odwołujące się do nieistniejącego obszaru pamięci. Mechanizm ten umożliwia implementację przez system operacyjny pamięci wirtualnej (np. na dysku lub karcie pamięci FLASH). W typowej pracy z mikrokontrolerami LPC21xx najczęściej podprogram obsługi tego wyjątku będzie się sprowadzał do wyświetlenia informacji o błędzie i wejściu w nieskończoną pętlę. W przypadku, gdy implementujemy mechanizm pamięci wirtualnej, podprogram obsługi przerwania po usunięciu błędu powraca do miejsca jego wystąpienia i instrukcja, która spowodowała wyjątek jest ponownie wykonywana. Mechanizm pamięci wirtualnej jest dobrze znany wszystkim użytkownikom 32-bitowych systemów operacyjnych. Działanie jego polega na emulacji przez system operacyjny części pamięci operacyjnej za pomocą pliku

na dysku lub partycji wymiany.

Wyjątek **Software Interrupt (SWI)** jest wywoływany programowo instrukcją SWI, umożliwia to realizację przerwania programowych, tzw. funkcji systemu operacyjnego. Identyfikacyjny mechanizm posiada mikroprocesor x86. Wszyscy którzy kiedyś mieli kontakt z programowaniem w assemblerze w systemie MS-DOS znają mechanizm przerwania programowych generowanych rozkazem INT. Podczas programowania mikrokontrolerów LPC21xx tryb ten będziemy najczęściej wykorzystywali do zmiany trybu ochrony na tryb uprzywilejowany (supervisor) w celu np. zablokowania systemu przerwania.

Wyjątek **Undefined Instruction** wyjątek ten jest generowany, gdy CPU napotka instrukcję nieznaną dla procesora. Mechanizm ten można wykorzystać np. do emulacji instrukcji koprocesora, którego mikrokontrolery LPC21xx nie posiadają.

**Reset** – podczas startu systemu CPU rozpoczyna wykonanie programu od tego wektora. W stanie początkowym bity trybu ochrony ustawiane są na tryb *supervisor* oraz blokowane jest zgłaszanie przerwania zewnętrznych poprzez ustawienie odpowiednich bitów w CPSR.

### Lista instrukcji procesora ARM7TDMI-S

Podstawą niniejszego kursu będzie język C, więc przedstawienie listy rozkazów nie jest niezbędne. Niemniej jednak zrozumienie architektury mikroprocesora umożliwi pisanie bardziej efektywnych programów. W mikrokontrolerach

LPC21xx rdzeń ARM7TDMI-S ma na sztywno ustalony porządek bajtów little-endian, o czym musimy pamiętać podczas ustawiania opcji kompilatora. Procesor ARM posiada architekturę

**Load, Store**, więc wszystkie operacje muszą być wykonywane na rejestrach. Aby wykonać jakąś operację na danej znajdującej się w pamięci musimy jej zawartość przesłać do rejestru, wykonać żądaną operację, a następnie wynik operacji przesłać ponownie do pamięci. Bardzo ciekawą cechą ARM7TDMI-S wyróżniającą go od innych mikroprocesorów jest to, że każda instrukcja może być wykonywana warunkowo. Najstarsze cztery bity kodu rozkazu są porównywane z bitami flag warunków w rejestrze CPSR, jeżeli bity te nie są zgodne, dana instrukcja nie jest wykonywana (w jej miejsce do potoku wstawiany jest rozkaz NOP). Wykonywanie każdej instrukcji w sposób warunkowy pozwala w dużej liczbie przypadków na uniknięcie używania instrukcji rozgałęzień, których wykonanie powoduje ponowne zapełnianie potoku nowymi instrukcjami i dodatkową stratą czasu. Do większości mnemoników assemblera w trybie ARM można dodać sufix z mnemonikami warunkowymi. Listę mnemoników warunków przedstawiono w **tab. 4**.

Przykładowa instrukcja **ADDMI R1,R2,#0x8000000** dodaje do rejestru R1 zawartość R2+0x80000000 tylko wtedy, gdy wynikiem wykonania poprzedniej operacji było ustawienie bitu N w rejestrze CPSR, w przeciwnym razie zawartość rejestru R1 pozostanie niezmienną. Instrukcje rozgałęzień potrafią skoczyć o wartość w przód lub w tył o 32 MB od bieżącej instrukcji. Podstawową instrukcją skoku jest instrukcja **B (branch)**, której wykonanie powoduje skok pod wskazany adres. Modyfikacją instrukcji **B** jest instrukcja **BL (Branch**

**Tab. 3. Adresy wektorów wyjątków, priorytetów oraz zalecanej instrukcji kończącej obsługę wyjątku**

Wyjątek	Priorytet	Adres wektora	Tryb ochrony wyjątku	Instrukcja powrotu z podprogramu obsługi	Zawartość LR ARM (THUMB)
Reset	1 (Najwyższy)	0x00000000	Supervisor	–	–
Undefined Instruction	6 (Najniższy)	0x00000004	Undefined	MOVS PC,R14_und	PC+4 (PC+2)
Software Interrupt (SWI)	6 (Najniższy)	0x00000008	Supervisor	MOVS PC,R14_svc	PC+4 (PC+2)
Prefetch Abort	5	0x0000000C	Abort	SUBS PC,R14_abt,#4	PC+4 (PC+4)
Data Abort	2	0x00000010	Abort	SUBS PC,R14_abt,#8	PC+8 (PC+8)
IRQ (interrupt)	4	0x00000018	IRQ	SUBS PC,R14_irq,#4	PC+4 (PC+4)
FIQ (Fast interrupt)	3	0x0000001C	FIQ	SUBS PC,R14_fiq,#4	PC+4 (PC+4)

Link), która działa tak samo jak instrukcja B, ale dodatkowo zapisuje zawartość PC+4 w rejestrze LR. Umożliwia to realizację mechanizmu podprogramów. Używając wraz z instrukcją B i BL mnemoników warunkowych można wykonywać skoki warunkowe oraz warunkowe wywołania podprogramów. W innych mikrokontrolerach np. 8051 rozgałęzienia warunkowe najczęściej są wykonywane do +128 bajtów do przodu i -127 bajtów wstecz. Powoduje to konieczność użycia dodatkowych instrukcji skoków bezwarunkowych, co komplikuje i wydłuża program. W ARM-ach możemy wykonywać skoki w zakresie aż do  $\pm 32$  MB co jest dużą zaletą. Specjalną odmianą instrukcji B i BL są instrukcje BX i BLX, które oprócz skoku dodatkowo zmieniają tryb pracy procesora z THUMB na ARM i odwrotnie. Instrukcje

te są jedynym dozwolonym sposobem na zmianę trybu pracy.

Kolejnym typem instrukcji są instrukcje przetwarzania danych.

Każda instrukcja przetwarzania danych posiada rejestr przeznaczenia oraz dwa rejestry zawierające argumenty. Pierwszy operand musi być rejestrem natomiast drugi operand może być rejestrem lub wartością stałą. Dodatkowo jako czwarty argument możemy podać przesunięcie logiczne do 32 bitów, które nie powoduje wydłużenia wykonania rozkazu. Flaga „S” informuje czy rozkaz ma zaktualizować flagi warunkowe w CPSR czy też nie. Jeżeli jako rejestr przeznaczenia wyspecyfikowany jest rejestr PC (R15) flaga „S” ma dodatkowe znaczenie polegające na skopiowaniu rejestru SPSR z bieżącego trybu ochrony do rejestru CPSR.

Oki, Network Solutions  
for a Global Society

## OKI

### 32-bitowe mikrokontrolery ARM7

- rdzeń ARM7TDMI
- zegar do 60MHz
- do 512 kB pamięci FLASH
- do 128 kB pamięci RAM
- do 42 linii we/wy
- 10 bitowy przetwornik ADC
- 16 bitowe wyjścia PWM
- do 7 liczników/timerów 16-bitowych
- we/wy szeregowo UART/I2C/I2S/SSIO
- zegar czasu rzeczywistego
- do 4 kanałów DMA

Autoryzowany dystrybutor:

02-493 Warszawa  
ul. Krańcowa 49  
tel.: (22) 33 60 200  
fax: (22) 33 60 201  
e-mail: acte@acte.pl

[www.acte.pl](http://www.acte.pl)

## GAMMA

Oficjalny Przedstawiciel  
w Polsce firmy Digi:

**Digi**

**GAMMA Sp. z o.o.**  
01-013 Warszawa ul. Kacza 6 Lok A  
tel. 022 862 75 00, fax. 022 862 75 01  
e-mail: info@gamma.pl  
www.gamma.pl

Rodzina modułów Digi Connect firmy Digi to gotowe rozwiązania pozwalające na bezpieczną komunikację między dowolnymi urządzeniami posiadającymi interfejs RS232 lub bezprzewodowy interfejs sieciowy 802.11b z siecią Ethernet

Podstawowe Parametry techniczne modułów:

**Digi Connect:**

- Procesor: 32-Bit ARM, NetSilicon NS7520;
- Pamięć: 2 MB Flash, 8 MB RAM;
- Interfejs sieciowy: 10/100Base-T;
- Bezprzewodowy interfejs sieciowy: 802.11b;
- interfejs High Speed TTL, szybkość do 230 Kbps.

Digi Connect ME    Digi Connect Wi-ME    Digi Connect EM    Digi Connect Wi-EM

Instrukcje te używane są do powrotu z podprogramu obsługi wyjątku. Do najczęściej wykorzystywanych instrukcji przetwarzających dane należą:

- AND – Logiczny AND
- EOR – Exclusive-OR
- SUB – odejmowanie
- RSB – odejmowanie w odwróconej kolejności argumentów.
- ADD – Dodawanie
- ADC – Dodawanie z przeniesieniem
- SBC – Odejmowanie z pożyczką
- RSB – odejmowanie z pożyczką w odwróconej kolejności argumentów
- TST – testowanie AND
- TEQ – testowanie XOR
- CMP – porównanie
- CMN – odwrotne porównanie
- ORR – logiczny OR
- MOV – przeniesienie
- BIC – kasowanie bitu
- MVN – przenieś zanegowany (Rd:=0xFFFFFFFF xor Operand2)
- Kolejną grupą instrukcji są instrukcje transferu danych pamięć-rejestry:
- LDR – Załaduj 32-bitową daną z pamięci
- LDRH – Załaduj 16-bitową daną z pamięci
- LDRSH – Załaduj 16-bitową daną ze znakiem z pamięci
- LDRB – Załaduj 8-bitową daną z pamięci
- LDRSB – Załaduj 16-bitową daną ze znakiem z pamięci
- STR – Prześlij 32-bitową daną do pamięci
- STRH – Prześlij 16-bitową daną do pamięci
- STRSH – Prześlij 16-bitową daną ze znakiem do pamięci
- STRB – Prześlij 8-bitową daną do pamięci
- STRSB – Prześlij 8-bitową daną ze znakiem do pamięci

SWP – Zamień daną z zawartością rejestru

Ponieważ wszystkie rejestry są ortogonalne (równouprawnione), możemy użyć instrukcji modyfikujących bezpośrednio licznik rozkazów PC (R15), co jest użyteczne w przypadku, gdy adres skoku znajduje się poza zasięgiem instrukcji B (*branch*). ARM7TDMI-S posiada także instrukcje (LDM, STM), które potrafią przepisać do i z pamięci zawartość wszystkich rejestrów lub wybranego podzbioru rejestrów. Dostęp do rejestrów stanu procesora CPSR,SPSR możliwy jest za pomocą dwóch instrukcji MSR i MRS, które umożliwiają przesłanie zawartości CPSR i SPSR do rejestrów ogólnego przeznaczenia. Zmiana trybu ochrony procesora lub załączenie i wyłączenie przerwań możliwe jest tylko z uprzywilejowanego trybu ochrony. Ciekawą instrukcją jest instrukcja SWI (*Software Interrupt*), która jest odpowiednikiem instrukcji *int* procesora 80x86. Powoduje ona wygenerowanie wyjątku przerwania programowego SWI. Jako argument SWI można podać 24-bitową liczbę, którą w podprogramie obsługi wyjątku można odczytać i na jej podstawie określić czynności jakie ma wykonać wyjątek. Instrukcja SWI powoduje zmianę trybu ochrony na Supervisor i często za jej pomocą wykonuje się odwołania do funkcji systemu operacyjnego. Pracując bez systemu będziemy często wykorzystywać generowanie tego wyjątku do zmiany trybu ochrony na *Supervisor* w celu np. zablokowania przerwań. Oprócz wyżej przedstawionych instrukcji procesor posiada także instrukcje mnożenia liczb ze znakiem, bez znaku oraz instrukcję charakterystyczną dla procesorów DSP – mnożenie i akumulację. Cechą charakterystyczną ARM-ów są trójargumentowe rozkazy np. ADDS R0, R1, R2.

Tab. 4. Lista mnemoników warunków w ARM7TDMI-S

Mnemonik	Flagi warunków	Działanie
EQ	Z=1	=
NE	Z=0	!=
CS	C=1	>= (liczba bez znaku)
CC	C=0	< (liczba bez znaku)
MI	N=1	Ujemny
PL	N=0	Dodatni lub 0
VS	V=1	Przepelnienie
VC	V=0	Brak przepelnienia
HI	C=1 i Z=0	> (liczba bez znaku)
LS	C=0 i Z=1	<= (liczba bez znaku)
GE	N=V	>=
LT	N!=V	<
GT	Z=0 i (N=Y)	>
LE	Z=1 lub (N!=V)	<=
AL.	Bez znaczenia	Zawsze

Gdzie do rejestru R0 wpisywana jest zawartość R1+R2 w innych mikroprocesorach rozkazy najczęściej operują na dwóch argumentach gdzie operacje wykonuje się na rejestrze przeznaczenia np. dla mikroprocesora x86 instrukcja ADD EAX, ECX do rejestru EAX dodaje zawartość EAX + ECX. Unikalną cechą listy instrukcji ARM-ów jest także łączenie operacji przesunięcia i obrotu z instrukcjami arytmetycznymi, logicznymi, lub rozkazami przesłań. Jako czwarty argument tych rozkazów możemy wyspecyfikować o ile bitów należy obrócić lub przesunąć rejestr np. *orr r1,r2,r3,ls! #4* powoduje zapis do rejestru R1 sumy logicznej rejestru r2 z rejestrem R3 przesuniętym o 4 bity w lewo. Na tym zakończymy opis architektury ARM7TDMI-S.  
**Lucjan Bryndza SQ7FGB**  
**lucjan.bryndza@ep.com.pl**

# ARM-ia

W naszej ofercie znajdują się mikrokontrolery z rdzeniem ARM dwóch dużych producentów:

**PHILIPS**

Jeśli potrzebujesz ARM-a wyposażonego w pamięć Flash i rozbudowane peryferia – dobrze traficie!

FUTURE ELECTRONICS POLSKA Sp. z o.o.  
 03-704 Warszawa, ul. Paniańska 9  
 tel.: (22) 618 92 02, fax: (22) 618 80 50  
<http://www.futureelectronics.com>




