

Mikrokontrolery z rdzeniem ARM, część 2

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.

Rejestry

Rdzeń ARM7TDMI-S posiada łącznie 31 rejestrów ogólnego przeznaczenia oraz 6 rejestrów statusu (flag), przy czym programista ma dostęp jednocześnie do 16 rejestrów R0...R15. Pozostałe rejestry są używane zamiennie z rejestrami R0...R15. Gdy mikroprocesor zostanie przełączony z trybu ochrony użytkownika na inny tryb część rejestrów zostaje podmieniona. Na przykład, gdy w wyniku wystąpienia przerwania zostanie zmieniony tryb z użytkownika na IRQ rejestry R0...R12 i R15 pozostaną te same, natomiast w miejsce rejestrów R13 i R14 zostanie podstawiony prywatny rejestr wykorzystywany przez tryb IRQ, nazwany R13_IRQ i R14_IRQ. Wykonanie instrukcji `MOVS R13,#100` podczas gdy procesor znajduje się w trybie IRQ powoduje wpisanie do rejestru R13_IRQ wartości 100, natomiast rejestr R13 pozostanie nienaruszony. Rejestry prywatne w trybach ochrony przyjmują identyfikatory przedstawione

w tab. 1. Na przykład rejestr R13 w trybie systemowym zamieniany jest na rejestr R13_sys.

Musimy pamiętać, że przy zmianie trybu ochrony nie wszystkie rejestry są podmieniane. W trybie FIQ podmieniane są rejestry R8...R14. Natomiast w pozostałych trybach tylko rejestry R13 oraz R14.

Tryb FIQ podmienia więcej rejestrów niż pozostałe, ponieważ jest on wykorzystywany do obsługi krytycznych czasowo przerwania. Gdyby tryb ten zapisywał tylko rejestry R13, R14 pozostałe trzeba by było zachować w sposób programowy, co spowodowałoby to dodatkową stratę czasu.

Oprócz rejestrów R0...R15 do dyspozycji mamy rejestr CPSR (Current Program Status Register), który jest rejestrem słowa stanu procesora. Zawiera on znaczniki wyniku wykonanych operacji arytmetycznych i logicznych, flagi pozwalające na kontrolę przyjmowa-

nia przerwania, 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 przykład w momencie przyjęcia przerwania rejestr statusu CPSR jest kopiowany do SPSR, a po zakończeniu procedury obsługi przerwania jest on ponownie kopiowany do CPSR. Gdyby zawartość rejestru flag nie była kopiowana program sam musiałby troszczyć się o odtworzenie jego zawartości.

Wszystkie rejestry R0...R15 są równouprawnione i można na nich wykonywać wszystkie operacje znaj-



Tryb ARM - rejestr ogólnego przeznaczenia

System and User	FIQ	Supervisor	Abort	IRQ	Undefined
r0	r0	r0	r0	r0	r0
r1	r1	r1	r1	r1	r1
r2	r2	r2	r2	r2	r2
r3	r3	r3	r3	r3	r3
r4	r4	r4	r4	r4	r4
r5	r5	r5	r5	r5	r5
r6	r6	r6	r6	r6	r6
r7	r7	r7	r7	r7	r7
r8	r8_fiq	r8	r8	r8	r8
r9	r9_fiq	r9	r9	r9	r9
r10	r10_fiq	r10	r10	r10	r10
r11	r11_fiq	r11	r11	r11	r11
r12	r12_fiq	r12	r12	r12	r12
r13	r13_fiq	r13_svc	r13_abt	r13_irq	r13_und
r14	r14_fiq	r14_svc	r14_abt	r14_irq	r14_und
r15 (PC)	r15 (PC)	r15 (PC)	r15 (PC)	r15 (PC)	r15 (PC)

Tryb ARM - rejestry znaczników

CPSR	CPSR	CPSR	CPSR	CPSR	CPSR
	SPSR_fiq	SPSR_svc	SPSR_abt	SPSR_irq	SPSR_und

▲ - rejestry bankowane

Rys. 3. Mapa rejestrów procesora (w trybie ARM)

Tryb Thumb - rejestr ogólnego przeznaczenia

System and User	FIQ	Supervisor	Abort	IRQ	Undefined
r0	r0	r0	r0	r0	r0
r1	r1	r1	r1	r1	r1
r2	r2	r2	r2	r2	r2
r3	r3	r3	r3	r3	r3
r4	r4	r4	r4	r4	r4
r5	r5	r5	r5	r5	r5
r6	r6	r6	r6	r6	r6
r7	r7	r7	r7	r7	r7
SP	SP_fiq	SP_svc	SP_abt	SP_irq	SP_und
LR	LR_fiq	LR_svc	LR_abt	LR_irq	LR_und
PC	PC	PC	PC	PC	PC

Tryb Thumb - rejestry znaczników

CPSR	CPSR	CPSR	CPSR	CPSR	CPSR
	SPSR_fiq	SPSR_svc	SPSR_abt	SPSR_irq	SPSR_und

 - rejestry bankowane

Rys. 4. Mapa rejestrów procesora (w trybie Thumb)

dujące się na liście rozkazów. Dwa z nich spełniają szczególne funkcje: Rejestr R15 jest licznikiem rozkazów PC (*Program Counter*), do jego zawartości można się odwoływać za pomocą wielu instrukcji. Na przykład skok pod wskazany adres można zrealizować poprzez wpisanie określonej wartości instrukcją

MOV do rejestru R15. W typowych mikrokontrolerach np. w AVR nie ma możliwości wykonywania jakichkolwiek operacji bezpośrednio na zawartości licznika rozkazów. Kolejnym rejestrem pełniącym dodatkową funkcję jest rejestr R14, wykorzystywany jako *Link Register* (LR). W mikrokontrolerach takich

jak AVR czy 8051 wywołanie podprogramu odbywa się za pomocą instrukcji CALL, która powoduje zapamiętanie licznika rozkazów na stosie oraz skok do podprogramu. Powrót z podprogramu wykonywany jest za pomocą instrukcji RET, która odtwarza ze stosu zawartość licznika rozkazów. Ponieważ w ARM7 stos jest realizowany na drodze programowej i nie jest niezbędny do pracy systemu, skok do podprogramu realizowany jest za pomocą instrukcji BL (*Branch Link*). Instrukcja ta działa w ten sposób, że kopiuje zawartość licznika rozkazów PC (R15), do rejestru LR (R14) *Link Register*, a następnie wykonuje skok pod wskazany adres. Powrót z podprogramu odbywa się poprzez skopiowanie zawartości rejestru LR z powrotem do licznika rozkazów PC. Wyraźnie widać tutaj zaletę równouprawnienia wszystkich rejestrów, ponieważ zbędne staje się użycie specjalnych rozkazów operujących na liczniku rozkazów np. RET czy RETI. W przypadku zagnieżdżenia podprogramów, czyli skoku z podprogramu do innego podprogramu, użytkownik musi sam zatroszczyć się o zapamiętanie rejestru LR (R14) przed skokiem do kolejnego podprogramu, a następnie po wyjściu z zagnieżdżonego podprogramu powinien odtworzyć jego zawartość. Do zapamiętania rejestru LR (R14) można wykorzystać np. stos lub dowolny obszar pamięci. W pozostałych przypadkach, gdy nie używamy rozkazu BL, rejestr R14 może być wykorzystany w dowolnym celu. Chociaż R13 nie ma szczególnego zastosowania wyznaczonego przez architekturę procesora, najczęściej oprogramowanie wykorzystuje ten

Tab. 1. Tryby ochrony procesora i wynikające z nich dozwolone ustawienia bitów

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

Mikrokontrolery z rdzeniem

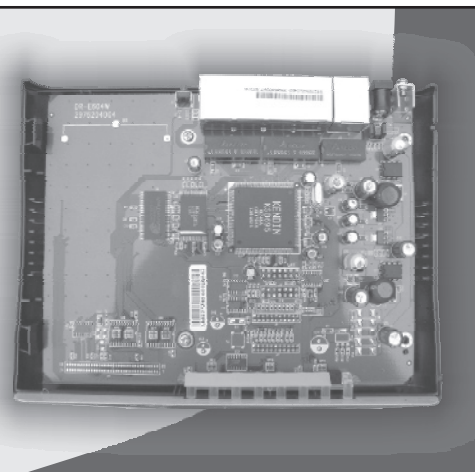
ARM

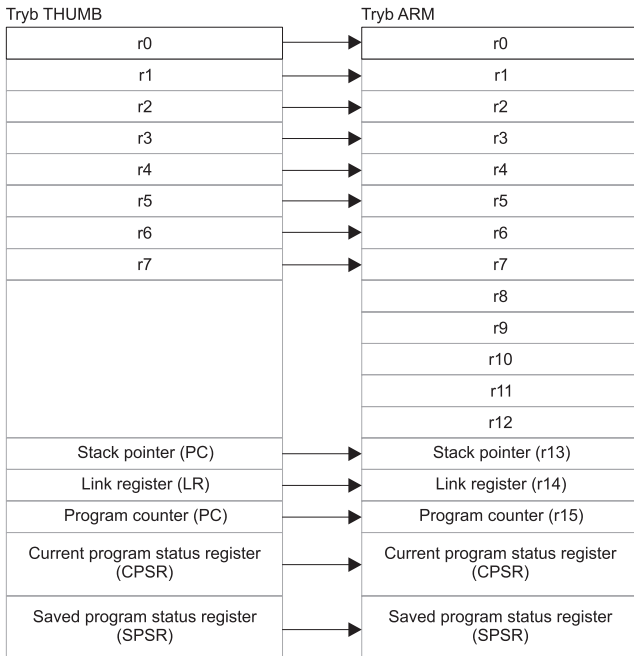
Oferujemy mikrokontrolery z rdzeniem ARM7TDMI i pamięcią Flash 32...512 kB, SRAM 8...64 kB, interfejsami UART, SPI, I2C, CAN, USB, rozbudowanymi funkcjonalnie timerami (do 20 w jednym układzie!), wielokanałowymi przetwornikami A/C, C/A i PWM...

Po prostu mamy wszystko!



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





Rys. 5. Sposób mapowania rejestrów z trybu Thumb na tryb ARM

rejestr jako wskaźnik stosu (SP) (*Stack Pointer*), ponieważ jest on bankowany we wszystkich trybach ochrony. Umożliwia to implemen-

tu operacyjnego i nie jest sztywno ustalone, jak w przypadku standardowych architektur.

Na rys. 3 przedstawiono reje-

tację oddzielnych stosów dla każdego trybu. Na dobrą sprawę gdybyśmy chcieli używać tylko jednego stosu dla wszystkich trybów, równie dobrze jako wskaźnik stosu można użyć innego rejestru z zakresu R0...R12.

Instrukcje procesora ARM umożliwiają implementację wszystkich rodzajów stosu: rosnący lub malejący, wskazujący na ostatni zajęty element lub pierwszy wolny. Wszystko zależy od programu lub, w bardziej zaawansowanych aplikacjach, od syste-


stry procesora (Tryb ARM) w poszczególnych trybach ochrony.

W 16-bitowym trybie THUMB mamy do dyspozycji 32-bitowe rejestry R0...R7, LR, SP, PC, CPSR, SPSR. Wszystkie rejestry są podzbiorem rejestrów trybu ARM. Ponieważ lista rozkazów w tym trybie jest zbliżona do zestawu instrukcji standardowego mikroprocesora do dyspozycji mamy rejestr wskaźnika stosu implementowany sprzętowo. Rejestry tego trybu są przedstawione na rys. 4.

Rejestry z trybu THUMB są mapowane na rejestry trybu ARM w sposób przedstawiony na rys. 5.

W trybie THUMB rejestry R0...R7 są nazywane zestawem dolnym, natomiast rejestry R8...R15 są nazywane górnym zestawem rejestrów. Górne rejestry nie są dostępne w standardowym zestawie, ale można je wykorzystać jako pamięć tymczasową. Dostęp do górnych rejestrów jest możliwy za pomocą specjalnego wariantu instrukcji MOV.

Lucjan Bryndza SQ7FGB
lucjan.bryndza@ep.com.pl



**Oficjalny Przedstawiciel
w Polsce firmy 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