

Politechnika Świętokrzyska

Laboratorium
Mikrokontrolerów

Ćwiczenie 2
Instrukcje arytmetyczne, logiczne

dr inż. Robert Kazała

Kielce 2015

Cel ćwiczenia

Celem ćwiczenia jest poznanie instrukcji arytmetycznych, logicznych, przesunięć. Zapoznanie się z działaniem instrukcji IT oraz skoków z wykorzystaniem adresów zapisanych w tablicy.

Instrukcje ADD, SUB, RSB, ADC, SBC, RSC.

Sposób wywołania

op{S} {cond} {Rd}, Rn, Operand2

op{cond} {Rd}, Rn, #imm12 ; dla Thumb-2 tylko ADD i SUB

imm12 – jest dowolną wartością z zakresu 0-4095

Instrukcja	Opis
ADD	Dodaje wartość Rn i Operand2
SUB	Odejmuje wartość Operand2 od Rn
RSB	Odejmuje wartość Rn od Operand2
ADC	Dodaje wartość Rn i Operand2 oraz bit przeniesienia
SBC	Odejmuje wartość Operand2 od Rn. Jeżeli bit przeniesienia jest wyzerowany to wynik jest zmniejszany o jeden.
RSC	Odejmuje wartość Rn od Operand2. Jeżeli bit przeniesienia jest wyzerowany to wynik jest zmniejszany o jeden.

Przykłady wywołań

ADD r2, r1, r3

SUBS r8, r6, #240

RSB r4, r4, #1280

ADCHI r11, r0, r3

SBC r5, r8, r11

Instrukcje AND, ORR, EOR, BIC, ORN, TST , TEQ.

Sposób wywołania

op{S} {cond} Rd, Rn, Operand2

Instrukcja	Opis
AND	Logiczne AND
ORR	Logiczne OR
EOR	Logiczne Exclusive OR
BIC	Logiczne AND NOT
ORN	Logiczne OR NOT

Instrukcje TST , TEQ.

Sposób wywołania

op{cond} Rn, Operand2

Instrukcja	Opis
TST	Instrukcja porównuje bitowo Rn i Operand2, odpowiednik AND lecz nie zwraca wyniku tylko ustawia flagi.
TEQ	Instrukcja wykonuje operację Exclusive OR na Rn i Operand2, lecz nie zwraca wyniku tylko ustawia flagi.

Instrukcje REV, REV16, REVSH, RBIT.

Sposób wywołania

op{cond} Rd, Rn

Instrukcja	Opis
REV	Zamienia ustawienie bajtów w słowie.
REV16	Zamienia ustawienie bajtów niezależnie w każdym półsłowie.
REVSH	Zamienia ustawienie bajtów w dolnym półsłowie i rozszerza znak do 32 bitów.
RBIT	Zmienia porządek bitów w słowie 32-bitowym.

Instrukcje ASR, LSL, LSR, ROR, RRX.

Instrukcje są synonimem instrukcji MOV z użyciem instrukcji przesuwania dla drugiego argumentu.

Sposób wywołania

op{S} {cond} Rd, Rm, Rs

op{S} {cond} Rd, Rm, #sh

RRX{S} {cond} Rd, Rm

Instrukcja	Opis
ASR	Arytmetyczne przesunięcie w prawo. Kopiuje bit znaku na dopisywane pozycje z lewej strony. Wykonuje operację dzielenia przez potęgę liczby 2.
LSL	Logiczne przesunięcie w lewo. Wykonuje operację mnożenia przez potęgę liczby 2.
LSR	Logiczne przesunięcie w prawo. Wykonuje operację dzielenia liczby bez znaku przez potęgę liczby 2.
ROR	Wykonuje operację rotacji bitu w prawo. Bity opuszczające rejestr z prawej strony dopisywane są po stronie lewej.
RRX	Wykonuje operację przesunięcia o jeden bit w prawo. Poprzednia flaga przeniesienia jest wpisywana do bitu [31]. Jeżeli występuje przyrostek S to bit [0] jest wpisywany do flagi przeniesienia.

Przykłady

ASR r7, r8, r9

LSLS r1, r2, r3

LSR r4, r5, r6

ROR r4, r5, r6

Instrukcje MUL, MLA, MLS.

Sposób wywołania

MUL{S}{cond} {Rd}, Rn, Rm

MLA{S}{cond} Rd, Rn, Rm, Ra

MLS{cond} Rd, Rn, Rm, Ra

Instrukcja	Opis
MUL	Mnożenie wartości Rn i Rm, mniej znaczące 32-bity wyniku zapisywane są do rejestru Rd.
MLA	Mnożenie wartości Rn i Rm oraz dodanie zawartości rejestru Ra, mniej znaczące 32-bity wyniku zapisywane są do rejestru Rd.
MLS	Mnożenie wartości Rn i Rm oraz odjęcie zawartości rejestru Ra, mniej znaczące 32-bity wyniku zapisywane są do rejestru Rd.

Instrukcje UMULL, UMLAL, SMULL, SMLAL.

Sposób wywołania

Op{S}{cond} RdLo, RdHi, Rn, Rm

Instrukcja	Opis
UMULL	Mnożenie 32-bitowych liczb bez znaku zawartych w rejestrach Rn i Rm. Wynik 64-bitowy zapisywany jest do rejestrów RdLo i RdHi.
UMLAL	Mnożenie 32-bitowych liczb bez znaku zawartych w rejestrach Rn i Rm. Wynik 64-bitowy dodawany jest do rejestrów RdLo i RdHi.
SMULL	Mnożenie 32-bitowych liczb ze znakiem zawartych w rejestrach Rn i Rm. Wynik 64-bitowy zapisywany jest do rejestrów RdLo i RdHi.
SMLAL	Mnożenie 32-bitowych liczb ze znakiem zawartych w rejestrach Rn i Rm. Wynik 64-bitowy dodawany jest do rejestrów RdLo i RdHi.

Instrukcje SDIV, UDIV

Sposób wywołania

op{cond} {Rd}, Rn, Rm

Instrukcja	Opis
SDIV	Dzielenie liczby Rn przez Rm, ze znakiem.
UDIV	Dzielenie liczby Rn przez Rm, bez znaku.

Instrukcja IT

W procesorach pracujących w trybie ARM możliwe jest warunkowe wykonanie większości instrukcji asemblera. W trybie Thumb-2 nie jest to możliwe. W celu określenia warunków wykonania instrukcji można wykorzystać polecenie IT.

Instrukcja IT realizuje wyrażenie If-Then dla czterech następujących po niej instrukcji. Jest to tzw. blok IT. Warunki mogą być wszystkie takie same lub niektóre mogą być inwersją logiczną innych warunków. Instrukcja IT jest 16-bitowa i realizowana jest tylko w trybie Thumb-2.

Składnia instrukcji jest następująca

`IT{x{y{z}}}{cond}`

gdzie:

x jest warunkiem dla drugiej instrukcji w bloku IT .

y jest warunkiem dla trzeciej instrukcji w bloku IT.

z jest warunkiem dla czwartej instrukcji w bloku IT.

cond jest warunkiem dla pierwszej instrukcji w bloku IT.

Warunkiem dla drugiej, trzeciej i czwartej instrukcji w bloku IT może być także:

T Then. Warunek zastosowany do każdej z instrukcji jest cond.

E Else. Warunek zastosowany do instrukcji jest inwersją cond.

16-bitowe w bloku IT inne niż CMP, CMN i TST nie zmieniają znaczników warunków. Dla pozostałych instrukcji można zastosować IT z warunkiem AL.

Następujące instrukcje nie są dozwolone w bloku IT:

- IT,
- rozgałęzień warunkowych,
- CBZ i CBNZ,
- TBB i TBH,
- CPS, CPSID i CPSIE,
- SETEND.

Dodatkowo rozgałęzienie bezwarunkowe jest dozwolone w bloku IT, jeżeli jest ostatnią instrukcją w bloku.

Przykłady

```
ITTE NE
ANDNE r0, r0, r1
ADDSNE r2, r2, #1
MOVEQ r2, r3
ITT AL
```

```
ADDAL r0,r0,r1
SUBAL r2,r2,#1
ADD r0,r0,r1
IT NE
ADD r0,r0,r1 ;błąd składni:brak warunku przy instrukcji w bloku IT
```

Instrukcje TBB i TBH

Instrukcje TBB i TBH umożliwiają wykonywanie rozgałęzień w ramach tablicy bajtów lub pół-słów. Mają one następującą składnię:

TBB [Rn, Rm]

TBH [Rn, Rm, LSL #1]

gdzie:

Rn to rejestr bazowy zawierający adres tablicy,

Rm jest rejestrem indeksowym zawierającym liczbę całkowitą wskazującą na bajt docelowy.

Przesunięcie w ramach tablicy zależy od rodzaju instrukcji:

TBB jest równy indeksowi,

TBH jest podwójną wartością indeksu.

Instrukcje te są dostępne tylko w trybie Thumb-2.

Instrukcje te umożliwiają wykonanie rozgałęzienia do przodu względem PC z wykorzystaniem przesunięcia odczytanego z tablicy bajtów lub pół-słów. Odległość rozgałęzienia jest podwójną wartością odczytana z tablicy.

Przebieg ćwiczenia

1. Uruchomić i przeanalizować programy znajdujące się w instrukcji.
2. Napisać programy wykorzystujące instrukcję IT.
3. Napisać programy wykorzystujące instrukcje TBB i TBH.
4. Napisać programy prezentujące działanie instrukcji ADD, SUB, RSB, ADC, SBC, RSC.
5. Napisać programy prezentujące działanie instrukcji AND, ORR, EOR, BIC, ORN, TST ,
TEQ.
6. Napisać programy prezentujące działanie instrukcji REV, REV16, REVSH, RBIT.
7. Napisać programy prezentujące działanie instrukcji ASR, LSL, LSR, ROR, and RRX.
8. Napisać programy prezentujące działanie instrukcji MUL, MLA, and MLS, UMULL,
UMLAL, SMULL, and SMLAL.
9. Napisać programy prezentujące działanie instrukcji SDIV, UDIV.