# Wykład 6

## Instrukcje arytmetyczne, logiczne

# 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

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

| 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. |

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

- 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 operacje 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
```

# Instrukcje ASR, LSL, LSR, ROR, RRX.

| Instrukcja | Opis |
|---|---|
| ASR | Arytmetyczne przesunięcie w prawo. Kopiuje bit znaku na dopisywane pozycje z lewej strony. Wykonuje operacje dzielenia przez potęgę liczby 2. |
| LSL | Logiczne przesunięcie w lewo. Wykonuje operacje mnożenia przez potęgę liczby 2. |
| LSR | Logiczne przesunięcie w prawo. Wykonuje operacje dzielenia liczby bez znaku przez potęgę liczby 2. |
| ROR | Wykonuje operacje 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. |

# Instrukcje ASR, LSL, LSR, ROR, RRX.

- 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. |