

## Budowa linii asemblera

Pola w linii są oddzielone znakami białymi (spacje, tabulacje).

Wewnątrz pola znaki te nie występują.

Linia programu zawiera cztery pola (być może puste):

| etykieta | mnemonik | operand     | komentarz                 |
|----------|----------|-------------|---------------------------|
| *        |          |             | to jest linia komentarza  |
| LABEL    | NOP      |             | etykieta może wystąpić... |
|          | NOP      |             | ... lub nie               |
| *        |          |             | operand (o ile jest)      |
| *        |          |             | zadajemy:                 |
|          | LDA      | LICZ8       | symbolicznie,             |
|          | LDA      | \$C0        | numerycznie,              |
|          | LDA      | TABL+TLEN-1 | lub przy pomocy wyrażenia |

## Dyrektywy asemblera

```
TABLEN EQU 5 #define TABLEN 5

ORG 0 typedef struct {
ELEM_A DS.B 1 char ELEM_A;
ELEM_B DS.W 1 int ELEM_B;
ELEM_C DS.B TABLEN char ELEM_C[TABLEN];
STRU_L EQU * } T_STRU;

ORG RAM
LICZ8 DS.B 1 char LICZ8;
LICZ16 DS.W 1 int LICZ16;
TABL DS.B TLEN char TABL[TLEN];
STR1 DS.B STRU_L T_STRU STR1,STR2;
STR2 DS.B STRU_L

ORG ROM
STALA8 DC.B 255 const char STALA8=255;
STALA16 DC.W $FFFF const int STALA16=0xffff;
NAPIS DC.B "ula",0 const char * NAPIS="ula";
STABL DC.B 1,2,3 const char STABL[STLEN]={1,2,3};
STLEN EQU *-STABL
```

## Tryby adresowania

**IMM** *immediate* – argument bezpośrednio w programie

```
0300 A6 55          LDA    #$55    $55 -> A
```

**INH** *inherent* – bez argumentów (dotyczy rejestrów CPU)

```
0302 4C           INCA          (A)+1 -> A
```

**EXT** *extended* – 16-bitowy adres argumentu

```
0303 C6 03 FF     LDA    $3FF    ($03FF) -> A
```

**DIR** *direct page* – 8-bitowy adres argumentu

```
0306 B6 FF     LDA    $FF    ($00FF) -> A
```

**IX IX1 IX2** *indexed (0,1,2-byte offset)* – IX jako wskaźnik argumentu

```
0308 F6           LDA    0,X    ((IX)) -> A
```

```
0309 E6 02       LDA    2,X    (2+(IX)) -> A
```

```
030B D6 03 E0    LDA    $3E0,X ($3E0+(IX)) -> A
```

**REL** *relative* – adres wynikowy liczony względem PC

```
030E 27 EF     BRA    $300    (PC)+$0300-$0310 -> PC
0310
```

## Adresowanie indeksowe

\*\*\*\*\*

\* IX jako indeks                   int    i;

\* UWAGA:

\* w HC05 indeks: [0..255]

CLX

```
LOOP1  LDA      TABL,X           for(i=0;i<TLEN;i++){
      STA      PORTA           PORTA=TABL[i];
      INX
      CPX      #TLEN
      BNE      LOOP1           }
```

\*\*\*\*\*

\* IX jako wskaźnik               char \*ptr;

\* UWAGA:

\* w HC05 tylko strona zerowa

LDX       #TABL               ptr=TABLE;

```
LOOP2  LDA      0,X           do{
      STA      PORTA           PORTA=*ptr++;
      INX
      CPX      #(TABLE+TLEN)
      BNE      LOOP2           } while(ptr!=TABLE+TLEN);
```

## Instrukcje typu rejestr–pamięć

tryby: **IMM, DIR, EXT, IX, IX1, IX2**

| skrót | opis   |
|-------|--|
| ADC   | suma z przeniesieniem                              |
| ADD   | suma   |
| AND   | iloczyn bitowy                                     |
| BIT   | test bitowy  |
| CMP   | porównanie   |
| EOR   | różnica symetryczna                                |
| LDA   | ładowanie akumulatora                              |
| LDX   | ładowanie rejestru indeksowego                     |
| MUL   | mnożenie A i X (tylko <b>INH</b> )                 |
| ORA   | suma bitowa  |
| SBC   | różnica z przeniesieniem                           |
| STA   | składowanie akumulatora (bez <b>IMM</b> )          |
| STX   | składowanie rejestru indeksowego (bez <b>IMM</b> ) |
| SUB   | różnica  |

## Instrukcje jednoargumentowe (RMW)

| skrót | opis   |
|-------|--|
| ASL   | arytmetyczne przesuwanie w lewo ( <b>DIR, IX, IX1</b> oraz <b>INH</b> – ASLA, ASLX)  |
| ASR   | arytmetyczne przesuwanie w prawo ( <b>DIR, IX, IX1</b> oraz <b>INH</b> – ASRA, ASRX) |
| BCLR  | zerowanie bitu ( <b>DIR</b> )  |
| BSET  | ustawienie bitu ( <b>DIR</b> )   |
| CLR   | zerowanie ( <b>DIR, IX, IX1</b> oraz <b>INH</b> – CLRA, CLRX)                        |
| COM   | dopełnienie arytmetyczne ( <b>DIR, IX, IX1</b> oraz <b>INH</b> – COMA, COMX)         |
| DEC   | zmniejszenie o 1 ( <b>DIR, IX, IX1</b> oraz <b>INH</b> – DECA, DECX)                 |
| INC   | zwiększenie o 1 ( <b>DIR, IX, IX1</b> oraz <b>INH</b> – INCA, INCX)                  |
| LSL   | logiczne przesunięcie w lewo ( <b>DIR, IX, IX1</b> oraz <b>INH</b> – LSLA, LSLX)     |
| LSR   | logiczne przesunięcie w prawo ( <b>DIR, IX, IX1</b> oraz <b>INH</b> – LSRA, LSRX)    |
| NEG   | dopełnienie bitowe ( <b>DIR, IX, IX1</b> oraz <b>INH</b> – NEGA, NEGX)               |
| ROL   | obrót w lewo z przeniesieniem ( <b>DIR, IX, IX1</b> oraz <b>INH</b> – ROLA, ROLX)    |
| ROR   | obrót w prawo z przeniesieniem ( <b>DIR, IX, IX1</b> oraz <b>INH</b> – RORA, RORX)   |
| TST   | sprawdzanie znaku i zera ( <b>DIR, IX, IX1</b> oraz <b>INH</b> – TSTA, TSTX)         |

# Instrukcje względnych skoków warunkowych

## tryb REL

| skrót | warunek wykonania skoku                   |
|-------|---|
| BCC   | brak przeniesienia                        |
| BCS   | wystąpienie przeniesienia                 |
| BEQ   | relacja równości                          |
| BHCC  | brak przeniesienia połówkowego            |
| BHCS  | wystąpienie przeniesienia połówkowego     |
| BHI   | ostra relacja większości                  |
| BHS   | nieostra relacja większości               |
| BIH   | stan wysoki wejścia $\overline{IRQ}$      |
| BIL   | stan niski wejścia $\overline{IRQ}$       |
| BLO   | ostra relacja mniejszości                 |
| BLS   | nieostra relacja mniejszości              |
| BMC   | przerwania nie zamaskowane                |
| BMI   | wynik ujemny                              |
| BMS   | przerwania zamaskowane                    |
| BNE   | relacja nierówności                       |
| BPL   | wynik nieujemny                           |
| BRA   | zawsze                                    |
| BRCLR | bit wyzerowany (słowo – tryb <b>DIR</b> ) |
| BRN   | nigdy                                     |
| BRSET | bit ustawiony (słowo – tryb <b>DIR</b> )  |

## Inne instrukcje sterujące

### tryb INH

| skrót | opis   |
|-------|--|
| BSR   | względny skok do podprogramu ( <b>REL</b> )                  |
| CLC   | zerowanie flagi przeniesienia                                |
| CLI   | zerowanie maski przerwań                                     |
| JMP   | skok bezwzględny ( <b>DIR, EXT, IX, IX1, IX2</b> )           |
| JSR   | wywołanie podprogramu ( <b>DIR, EXT, IX, IX1, IX2</b> )      |
| NOP   | instrukcja pusta   |
| RSP   | inicjalizacja wskaźnika stosu                                |
| RTI   | powrót z obsługi przerwania                                  |
| RTS   | powrót z podprogramu   |
| SEC   | ustawienie flagi przeniesienia                               |
| SEI   | ustawienie maski przerwań                                    |
| STOP  | zatrzymanie zegara, odblokowanie przerwania $\overline{IRQ}$ |
| SWI   | przerwanie programowe  |
| TAX   | przepisanie akumulatora do rejestru indeksowego              |
| TXA   | przepisanie rejestru indeksowego do akumulatora              |
| WAIT  | zatrzymanie CPU i odblokowanie przerwań                      |