Alicja Kapiszka

Paweł Adamski

Informatyka N1, gr. 30C

Sprawozdanie nr 1

Wstęp do mikrokontrolerów rodziny MCS-51

Wstęp

Przedmiotem laboratoriów był mikrokontroler 8051 z rodziny MCS-51. Jest on urządzeniem 8-bitowym. Zawiera: jednostkę arytmetyczno-logiczną (ALU), pamięć wewnętrzną (4 kB) oraz zewnętrzną (64 kB), rejestry (w tym akumulator ściśle związany z ALU oraz 4 bloki rejestrów ogólnego przeznaczenia, oznacoznych R0-R7), a także 4 porty wejścia wyjścia (P0, P1, P2, P3), za pomocą których możliwa jest interakcja z otoczeniem.

Do przetestowania programów napisanych na laboratoriach wykorzystany został zestaw uruchomieniowy ZL2MCS51 oraz oprogramowanie obejmujące kompilator i programator (MIDE-51, Flip). Zestaw podłączono do komputera za pomocą złącza RS-232 oraz zasilono, podłączając do stacji NI ELVIS. Używanymi elementami płyty były przyciski (podłączone do portu P3) oraz diody LED (podłączone do portu P2).

Użytym oprogramowaniem było środowisko MIDE-51 wraz z kompilatorem Asemblera (ASEM-51) oraz programator Flip , za pomocą którego możliwe jest zaprogramowanie mikrokontrolera, używając pliku .hex otrzymanego po kompilacji kodu programu napisanego w Asemblerze. Po każdym programowaniu mikrokontrolera należy wcisnąć przycisk Reset znajdujący się na płycie uruchomieniowej.

Celem laboratoriów było zapoznanie się z dokumentacją mikrokontrolera 8051 oraz zestawu uruchomieniowego ZL2MCS51, a następnie wykonanie zadań związanych z obsługą diód i przycisków poprzez odpowiednie zaprogramowanie mikrokontrolera.

Zadanie 1.

Należało zaimplementować dwa programy i zaobserwować różnice w ich działaniu:

```
ORG 0
JMP 100
ORG 100
start:
     MOV P2, P3
     JMP start
END
```

```
ORG 0
JMP 100
ORG 100
start:
     MOV C, P3.0
     MOV P2.0, C
     JMP start
END
```

Zadaniem obu programów było przekazanie stanu portu P3 na port P2, co skutkowało świeceniem się diód po wciśnięciu przycisku (wciśnięcie przycisku powoduje przejście stanu z wysokiego na niski,

natomiast dioda świeci również przy stanie niskim). W przypadku pierwszego programu działanie to występowało dla wszystkich diód (po wciśnięciu odpowiadającego danej diodzie przycisku), natomiast w przypadku drugiego – połączono tylko bit P2.0 portu P2 z bitem P3.0 portu P3, co skutkowało świeceniem diody SW0 po wciśnięciu przycisku S0.

Zadanie 2.

Należało napisać program, którego efektem miał być efekt pulsowania wybranej diody LED z częstotliwością 1 Hz (co oznacza, że dioda przez 0,5 s powinna się świecić, po czym przez 0,5 s być zgaszona). W tym celu należało wyznaczyć częstotliwość taktowania oraz czas trwania pojedynczego okresu sygnału zegarowego taktującego mikrokontroler, a także określić, ile taktów zegara przypada na jeden cykl rozkazowy mikrokontrolera. Częstotliwość taktowania zegara użytej płytki określona w dokumentacji wynosi 11059200 Hz. Na jeden cykl rozkazowy mikrokontrolera 8051 przypada 12 taktów zegara. Oznacza to, że na 1s przypada 11059200 / 12 = 921600 cyklów rozkazowych, czyli na 0,5s – 460800 cykli. W programie należało zastosować 3 pętle opóźniające trwające 0,5 s, a więc należało wyliczyć ilości iteracji tych pętli tak, aby ilość cykli rozkazowych wyniosła w przybliżeniu 460800.

```
ORG 0
JMP 100
ORG 100
start:
     MOV R3, #0E1h
     loop_a:
          MOV R4, #0FFh
          loop_b:
               DEC R4
               MOV A, R4
          JNZ loop_b
          MOV R5, #0FFh
          loop_c:
               DEC R5
               MOV A, R5
          JNZ loop_c
          DEC R3
          MOV A, R3
     JNZ loop_a
     CPL P2.5
     JMP start
END
```

-> umieszczenie następującego kodu w pamięci programu od 0
-> skok do adresu 100 w pamięci programu i umieszczenie następującego kodu po adresie 100 (ominięcie zarezerwowanych obszarów pamięci)
-> pętla główna
-> ustawienie wartości rejestru R3 na 225
-> pętla zewnętrzna (iterowana 225 razy)
-> ustawienie R4 na 255 (ilość iteracji pętli B)
-> 1. pętla wewnętrzna (iterowana 255 razy)
-> zmniejszenie wartości R4
-> skopiowanie wartości R4 do akumulatora (A)
-> jeśli A != 0 –> kolejna iteracja pętli B
-> ustawienie R5 na 255 (ilość iteracji pętli C)
-> 2. pętla wewnętrzna (iterowana 255 razy)
-> zmniejszenie wartości R5
-> skopiowanie wartości R5 do akumulatora (A)
-> jeśli A != 0 –> kolejna iteracja pętli C
-> zmniejszenie wartości R3
-> skopiowanie wartości R3 do akumulatora (A)
-> jeśli A != 0 –> kolejna iteracja pętli A (zewnętrznej)
-> instrukcja complement – odwrócenie bitów (zaświecenie/zgaszenie diody)
-> powrót do początku pętli głównej

<u>Wyliczenie odpowiednich ilości iteracji:</u>

Pierwszym krokiem było wyliczenie ilości cykli rozkazowych przypadających na każdą pętlę. W każdej z nich występowały 3 instrukcje: DEC (1 cykl rozkazowy), MOV A, Rn (1 cykl rozkazowy) i JNZ (2 cykle rozkazowe), ponadto przed każdą pętlą występowała instrukcja MOV Rn, direct (2 cykle rozkazowe). Oznacza to, że w pętli A zostały wykonane: 4 cykle rozkazowe, 2 cykle rozkazowe przed pętlą B, 2 cykle rozkazowe przed pętlą C oraz 4*x cykli rozkazowych w pętli B i 4*y cykli w pętli C (gdzie x i y to ilość iteracji odpowiednio pętli B i C). Założyliśmy, że x = 255 i y = 255 (maksymalna ilość iteracji). Wówczas

w pętli A zostanie wykonane: 8 + 1020 + 1020 = 2048 cykli rozkazowych. Chcemy uzyskać 460800 cykli rozkazowych, więc pętla A wykona się 460800 / 2048 = 225 razy.

Zadanie 3.

Zadanie polegało na takiej modyfikacji programu z poprzedniego zadania, by uzyskać efekt wędrowania diody, tzn. zapalania się co 0,5 s kolejnych diód na płytce. W tym celu wykorzystaliśmy instrukcję RL, czyli rotacji w lewo.

```
ORG 0
JMP 100
ORG 100
MOV P2, #0FEh
start:
     MOV R4, #0E1h
     loop_a:
          MOV R5, #0FFh
          loop_b:
               DEC R5
               MOV A, R5
          JNZ loop_b
          MOV R3, #0FFh
          loop_c:
               DEC R4
               MOV A, R4
          JNZ loop_c
          DEC R4
          MOV A, R4
     JNZ loop_a
     MOV A, P2
     RL A
     MOV P2, A
     JMP start
END
```

-> zainicjalizowanie portu P2 wartością 254 (diody 1-7 zgaszone – bit 1, dioda 0 świeci – bit 0)

-> kopiowanie wartości P2 do akumulatora
-> rotacja akumulatora w lewo (zmiana świecącej diody na tą po lewej stronie)
-> kopiowanie wartości akumulatora z powrotem do P2