Wojciech Zatorski, lab 7, gr. 38, rok 3.

SPRZĘT KOMPUTEROWY

Sprawozdanie nr 2

1. Asembler

Asembler jest to język programowania niskiego poziomu, wykorzystujący instrukcje procesora. Program napisany w asemblerze jest tłumaczony na (binarny) kod maszynowy. Program w asemblerze jest bardzo efektywny w porównaniu do programów napisanych w innych językach, jednakże jego tworzenie jest stosunkowo trudne.

Współcześnie praktycznie nie używa się asemblera do pisania całych programów. Jest on za to wciąż używany do pisania fragmentów wymagających bardzo wysokiej wydajności lub mających inne specjalne wymagania.

2. BCD

BCD (ang. Binary Coded Decimal czyli liczby dziesiętne zakodowane binarnie) - sposób zapisu liczb w komputerze. Polega na zapisaniu jednej cyfry dziesiętnej (dwójkowy zapis pozycyjny) przy użyciu dokładnie czterech bitów.

Cyfry:

0	0000	5	0101
1	0001	6	0110
2	0010	7	0111
3	0011	8	1000
4	0100	9	1001

Żeby zapisać liczbę 127 kodujemy każdą z cyfr: (0001, 0010, 0111). Ponieważ większość komputerów zapisuje liczby na ośmiu bitach, powstają dwie możliwości zapisu BCD:

- wyzerowanie lub ustawienie najstarszych czterech bitów i zapisywanie cyfr na czterech najmłodszych bitach.
- zapis dwóch cyfr w każdym bajcie - tak zwane spakowane BCD.

3. Odczytanie czasu rzeczywistego.

Do odczytania aktualnego czasu wykorzystałem przerwanie BIOSu 1Ah udostępniające funkcje do operowania na kalendarzu i zegarze czasu rzeczywistego.

przerwanie:
1Ah

wyjście:
CH – godzina (bcd)
CL – minuta (bcd)
DH - sekunda (bcd)

Wynikiem działania poniższego programu, będzie wyświetlenie aktualnego czasu w formacie:
gg:mm:ss -> 10:15:32

```
var godz, mint, sek : byte;
begin
asm
        MOV AH, 02h
        INT 1Ah

MOV AL, CH {godzina }
AND CH, 0Fh
SHR AL, 4
MOV BL, 10
MUL BL
ADD CH, AL
MOV godz, CH


MOV AL, CL { minuta}
AND CL, 0Fh
SHR AL, 4
MOV BL, 10
MUL BL
ADD CL, AL
MOV mint, CL


MOV AL, DH {sekunda}
AND DH, 0Fh
SHR AL, 4
MOV BL, 10
MUL BL
ADD DH, AL
MOV sek, DH

end;
        writeln (godz:2, ':', mint:2, ':', sek:2);
end.
```

4. Zamiana liczby binarnej na BCD.

Notacja BCD jest powszechnie używana do wprowadzania i wyprowadzania informacji numerycznych, a także podczas kodowania w sygnalizacji optycznej za pomocą wskaźników typu LCD (Liquid Crystal Display) i LED (Light Emitting Diode).

Przykłady podstawowych konwersji liczbowych

Liczby dziesiętne (podstawa 10)	Liczby binarne (podstawa 2)	Liczby ósemkowe (podstawa 8)	Liczby szesnastkowe (podstawa 16)	Liczby dziesiętne (kodowanie BDC)
00	0000	00	0	0000 0000
01	0001	01	1	0000 0001
02	0010	02	2	0000 0010
03	0011	03	3	0000 0011
04	0100	04	4	0000 0100
05	0101	05	5	0000 0101
06	0110	06	6	0000 0110
07	0111	07	7	0000 0111
08	1000	10	8	0000 1000
09	1001	11	9	0000 1001
10	1010	12	A	0001 0000
11	1011	13	B	0001 0001
12	1100	14	C	0001 0010
13	1101	15	D	0001 0011
14	1110	16	E	0001 0100
15	1111	17	F	0001 0101

```
var wynik:WORD;
begin
    asm
    MOV AX, 3214
    MOV CX, 10
    MOV DX, 0
    DIV CX
    MOV BX, DX
    MOV DX, 0
    DIV CX
    SHL DX, 4
    ADD BX, DX
    MOV DX, 0
    DIV CX
    SHL DX, 8
```

```
ADD BX, DX
MOV DX, 0
DIV CX
SHL DX, 12
ADD BX, DX
MOV wynik,BX
end;

writeln(wynik);
readln;
end.
```

5. Zamiana BCD na liczbę binarną.

```
var wynik : WORD;
begin
asm

MOV BX, 00400002b
MOV CX, BX
AND CX, 0Fh
SHR BX, 4
MOV AX, BX
AND AX, 0Fh
MOV DX, 10d
MUL DX
ADD CX, AX
SHR BX, 4
MOV AX, BX
AND AX, 0Fh
MOV DX, 100d
MUL DX
ADD CX, AX
SHR BX, 4
MOV AX, BX
AND AX, 0Fh
MOV DX, 1000d
MUL DX
ADD CX, AX
MOV wynik, CX

end;
writeln (wynik);
readln;
end.
```