

Architektura komputerów i systemy operacyjne

Lista 9

1. Rozkoduj następujące instrukcje zapisane dla prostego procesora przedstawionego na wykładzie:

- 0010000000000111
- 1001000000001011
- 0011000000001001

2. Wyjaśnij dokładnie w notacji RTN co robi poniższy program oraz napisz go w kodzie maszynowym szesnastkowo (ręczna aseblacja):

Etykieta	Adres	Instrukcja
	100	Load A
	101	Add B
	102	Jump S1
S2.	103	Add B
	104	Store A
	105	Halt
S1.	106	Add A
	107	Jump S2
A.	108	HEX 0023
B.	109	HEX 0001

3. Zapisz poniższy program w assemblerze prostego procesora:

```
if (X > 1) {  
    Y = X + X;  
    X = 0;  
}  
Y = Y + 1;
```

4. Zapisz poniższy program w assemblerze prostego procesora:

```
X = 1;  
while (X < 10) {  
    X = X + 1;  
}
```

5. Zapisz poniższy program w assemblerze prostego procesora:

```
Sum = 0;  
for (X=1; X<=10; X++)  
    Sum = Sum + X;
```

6. Wydaje się, że większa liczba rejestrów pomaga znacząco w redukcji dostępu procesora do pamięci głównej. Podaj prosty przykład, który pokazuje, że faktycznie tak jest, następnie napisz program z zadania 5 przy założeniu, że posiadamy dwa rejestry (AC oraz np. BC) i policz liczbę odwołań do pamięci głównej.
7. Załóżmy, że dodamy jeszcze jedną instrukcję do prostego procesora:

IncSZ Operand

Instrukcja zwiększa wartość pod adresem „Operand” i jeśli ta nowo zwiększona wartość jest równa 0, to licznik programu (PC) jest dodatkowo zwiększony o 1. Czyli zwiększamy wartość o jeden i jeśli ta nowa wartość jest równa zero to przeskakujemy następną instrukcję. Napisz tą nową instrukcją w notacji RTN.