

Omerovic Muhamed 18772

Logicki Dizaj Zadaca 3

2. Zadatak

a) Projektovati 4-struko grupno-asocijativni keš kapaciteta 64K sa blokom veličine 16 riječi (64B) i 32-bitnom adresom.

$$\text{Cap} = 64K$$

$$\text{Ass} = 4$$

$$\text{Block} = 64B$$

$$\text{Adr} = 32$$

Propracun :

$$N = \frac{\text{cap}}{\text{block} * \text{ass}} = \frac{64K}{64B * 4} = \frac{64K}{256} = 0.25K$$

$$n_{\text{index}} = \log_2 N = \log_2 256 = 8$$

$$n_{\text{offset}} = \log_2 \text{block} = \log_2 64 = 6$$

$$n_{\text{tag}} = \text{adr} - \text{index} - \text{offset} = 32 - 8 - 6 = 18$$

Rezultat:

$$\text{Tag} = 18b, \text{index} = 8b, \text{offset} = 6b$$

b) Projektovati direktno preslikani keš kapaciteta 64K sa blokom veličine 16 riječi (64B) i 64-bitnom adresom.

$$\text{Adr} = 64$$

$$\text{Block} = 64B$$

$$\text{Cap} = 64K$$

$$\text{Ass} = 1$$

Propracun:

$$N = \frac{\text{cap}}{\text{block} * \text{ass}} = \frac{64K}{64B * 1} = \frac{64K}{64} = 1K$$

$$n_{\text{index}} = \log_2 N = \log_2 1024 = 10$$

$$n_{\text{offset}} = \log_2 \text{block} = \log_2 64 = 6$$

$$n_{\text{tag}} = \text{adr} - \text{index} - \text{offset} = 64 - 10 - 6 = 48$$

Rezultat:

$$\text{Tag} = 48b, \text{index} = 10b, \text{offset} = 6b$$

Omerovic Muhamed 18772

Logicki Dizaj Zadaca 3

3. Zadatak

Koristeći mikroarhitekturu oglednog procesora, napisati mikrokod za instrukciju MULCO ($ac := ac * x$) koja će pomnožiti sadržaj AC registra sa konstantom specificiranom u 8 donjih bita instrukcije (x).
Primjer upotrebe instrukcije: MULCO A7

80: A := smask; //0000 0000 1111 1111

81: A := band(IR,A); //0000 0000 xxxx xxxx

82: B := AC;

83: A:= A + (-1);

84: ALU := A; if Z then goto 0;

85: ac := ac + B;

86: A := A + (-1); goto 84;

Objasnjenje:

Sa instrunkcijama 80. I 81. dobivamo posljednjih 8 bita sa Instrunkcijskog Registra (IR) I smjestamo ga u registar A, to nam oznacava mnozilac sa kojim treba da pomnozimo vrijednost na akumulatoru (AC).

Linija 82. smjesta pocetnu vrijednost akumulatora u registar B. Posto nemamo instrukciju za mnozenje to mozemo tretirati kao sabiranje akumulatora samim sa sobom n puta to nam je ekvivalnet $ac * x$.

Linija 83. Moramo vrijednost smanjiti za jedan jer u slucaju da posalje 1 ne trebamo nikako ulaziti u petlju.

Linija 84. smjesta trenutni brojac iz registra A u ALU I provjerava da li je Zero flag 0 ako jeste znaci da smo zavrшили sa instrukcijom a ako nije nastavljamo dalje sa izvravanjem.

Linija 85. sabira trenutnu vrijednsot akumulatora sa svojom pocetnom vrijednosti.

Linija 86. oduzima od vrijednosti sabirnice A , odnosno dekrementira vrijednost sabirnice A I vracamo se na liniju 84 gdje cemo opet provjeravati da li smo završili sa sabiranjem ili ne.