

Ovaj dokument je namijenjen isključivo za osobnu upotrebu studentima koji slušaju predmet ODR i u druge svrhe se ne smije upotrebljavati.

Primjer 1.

Koliko iznosi 8-bitni podatak 01011001 ako ga promatramo kao: NBC, 2'k,1'k, prikaz s bitom za predznak, BCD pakirani i nepakirani, ASCII znak

| | |
|----------------------------|--|
| NBC | $1+8+16+64=89$ |
| 2'k | pozitivan, 89 |
| 1'k | pozitivan, 89 |
| prikaz s bitom za predznak | pozitivan, 89 |
| BCD pakirani | 59 |
| BCD nepakirani | neispravno, ne smije imati ništa u gornja 4 bita |
| ASCII znak | iz tablice: veliko slovo y: 'Y' |

Primjer 2.

Koliko iznosi 8-bitni podatak 10111001 ako ga promatramo kao: NBC, 2'k,1'k, prikaz s bitom za predznak, BCD pakirani i nepakirani, ASCII znak

| | |
|----------------------------|--|
| NBC | $1+8+16+32+128 = 185$ |
| 2'k | negativan, $-(01000110+1) = -01000111 = -(1+2+4+64) = -71$ |
| 1'k | negativan, $-(01000110) = -(2+4+64) = -70$ |
| prikaz s bitom za predznak | negativan, $-(0111001) = -(1+8+16+32) = -57$ |
| BCD pakirani | neispravan jer ima nedozvoljenu kombinaciju 1011 |
| BCD nepakirani | neispravno, ne smije imati ništa u gornja 4 bita |
| ASCII znak | nije čisti ASCII, jer je ASCII 7-bitni kod |

Primjer 3.

Odrediti stanje općih zastavica nakon zbrajanja dva 6-bitna broja: 011001 i 010110

$$\begin{array}{r}
 010100 \\
 010101 \\
 + 010110 \\
 \hline
 101011
 \end{array}$$

$$C_n = 0, C_{n-1} = 1$$

$$c = C_n = 0$$

$$ov = C_n \oplus C_{n-1} = 0 \oplus 1 = 1$$

$$s = r_n = 1$$

$$z = 0 \text{ (jer je rezultat različit od 0)}$$

Primjer 4.

Interpretirajte rezultate prošlog zadatka ako promatrate brojeve kao da su to podatci u NBC-u i u 2^k-u:

a) Ako su podatci u NBC-u onda se radi o zbrajanju brojeva 25 i 22, a rezultat je 47 što je unutar opsega 6-bitnog NBC-a (opseg od 0 do 63). Da je rezultat ispravan vidi se i po stanju zastavice c koja je 0.

b) Ako su podatci u 2^k-u onda se radi o zbrajanju brojeva 25 i 22, a rezultat je negativan što je greška jer bi ispravan rezultat 47 bio izvan opsega 6-bitnog 2^k-a (opseg od -32 do 31). Da je rezultat neispravan vidi se i po stanju zastavice ov koja je 1.

Primjer 5.

Odrediti stanje općih zastavica nakon oduzimanja dva 6-bitna broja: 010110 i 011001

$$\begin{array}{r} 000110 \\ 010110 \\ 100110 \\ + \quad \quad \underline{1} \\ \hline 111101 \end{array}$$

$$C_n = 0, C_{n-1} = 0$$

$$c = C_n = 0$$

$$ov = C_n \oplus C_{n-1} = 0 \oplus 0 = 0$$

$$s = r_n = 1$$

$$z = 0 \text{ (jer je rezultat različit od 0)}$$

Odredimo dodatno i stanje posudbe, budući da se radi o oduzimanju:

$$b = \text{not } c = 1 \text{ (b je kratica za borrow, tj. posudbu)}$$

Primjer 6.

Interpretirajte rezultate prošlog zadatka ako promatrate brojeve kao da su to podatci u NBC-u i u 2^k-u:

a) Ako su podatci u NBC-u onda se radi o oduzimanju brojeva 22 i 25, a rezultat bi trebao biti -3 što nije unutar opsega 6-bitnog NBC-a (opseg od 0 do 63). Da je rezultat neispravan vidi se i po stanju zastavice c koja je 0.

b) a) Ako su podatci u 2^k-u onda se radi o oduzimanju brojeva 22 i 25, a rezultat je -3 što je ispravno jer je unutar opsega 6-bitnog 2^k-a (opseg od -32 do 31). Da je rezultat ispravan vidi se i po stanju zastavice ov koja je 0.

Primjer 7.

Ako se oduzimanje radi tako da se prvo izračuna dvojni komplement drugog operanda, a tek nakon toga se taj dvojni komplement zbroji s prvim operandom treba biti oprezan kod određivanja zastavica (iako se skoro uvijek dobije dobro stanje). Primjer za koji to ne funkcionira je, npr. 3-0. Ako se provede oduzimanje tako da se doda jedinični komplement i dodatna jedinica, onda su zastavice sigurno dobro određene:

$$\begin{array}{r} 111111 \\ 000011 \\ 111111 \\ + \quad \underline{1} \\ \hline 000011 \end{array}$$

$$C_n = 1, C_{n-1} = 1$$

$$\begin{aligned} c &= C_n = 1 \\ ov &= C_n \oplus C_{n-1} = 1 \oplus 1 = 0 \\ s &= r_n = 0 \\ z &= 0 \text{ (jer je rezultat različit od 0)} \end{aligned}$$

Ako se prvo izračuna dvojni komplement pa se tek onda on pribroji prvom operandu onda je postupak kako slijedi. Prvo računamo dvojni komplement od ničtice:

$$\begin{array}{r} 111111 \\ + \quad \underline{1} \\ \hline 000000 \end{array}$$

Zatim ovaj međurezultat dodajemo broju 3:

$$\begin{array}{r} 000000 \\ 000011 \\ + \quad \underline{000000} \\ \hline 000011 \end{array}$$

$C_n = 0, C_{n-1} = 0$, što je očito različito nego kod prvog postupka.

Problem nastaje zbog toga što smo "zaboravili" ("zanemarili") prijenos do kojeg je došlo pri operaciji računanja dvojnog komplementa drugog operanda, a koja zapravo glasi ovako:

$$\begin{array}{r} 111111 \\ 111111 \\ + \quad \underline{1} \\ \hline 000000 \end{array}$$

Prijenosi ukupne operacije oduzimanja dobiju se kada se napravi logički ILI između prijenosa u ovom koraku (operaciji dvojnog komplementa) i prijenosa kod zbrajanja prvog operanda s rezultatom ovog koraka.

Primjer 8.

Primjer za zbrajanje BCD-a s dvije korekcije (iz knjige)

Moguće je da je potrebna korekcija na obje znamenke, tj. grupe od četiri bita. Također je moguće da korekcija jedne znamenke prouzroči promjenu više znamenke tako da viša znamenka postane neispravna pa je onda treba korigirati iako je u početku (nakon zbrajanja) bila ispravna. Zato se korekcija provodi u više koraka. Ovo je pokazano sljedećim primjerom.

$$\begin{array}{r} 49 \\ + 56 \\ \hline 105 \end{array} \quad \begin{array}{r} 1 \\ 0100 \ 1001 \\ + 0101 \ 0110 \\ \hline 1001 \ 1111 \end{array}$$

Niža znamenka je neispravna, a viša je ispravna. Poluprijenosa nema ni na jednoj znamenki. Potrebno je korigirati samo nižu znamenku:

$$\begin{array}{r} 11 \ 11 \\ 1001 \ 1111 \\ + \quad \quad 0110 \\ \hline 1010 \ 0101 \end{array}$$

Nakon korekcije niže znamenke ona postaje ispravna. Dolazi do poluprijenosa s niže znamenke, ali kao što je objašnjeno on nije razlog za nove korekcije. Međutim, viša znamenka koja je prije korekcije bila ispravna i iznosila je 9, nakon zbrajanja s poluprijenosom postaje 10, odnosno neispravna, te je treba korigirati:

$$\begin{array}{r} 111 \\ 1010 \ 0101 \\ + 0110 \\ \hline 0000 \ 0101 \end{array}$$

Nakon druge korekcije dobivamo ispravne znamenke rezultata: 0 i 5. Za znamenku 1 nema mjesta jer su u osam bita prikazive samo dvije znamenke. Ovdje dolazi do prekoračenja opsega od dvije dekadске znamenke što se može prepoznati po zastavici prijenosa koja će nakon druge korekcije biti postavljena u jedinicu. Poluprijenos nakon druge korekcije ponovno nije uzrok za novu korekciju, ali kada bi postojala dodatna četiri bita iznad najviših, u njima bi se pojavila jedinica čime bi dobili puni rezultat - broj 105.

Primjer 9.

Prikažite stanje 8-bitnog registra s početnim stanjem 00110001 nakon operacija:

a) pomak ulijevo za 3 bita

10001000

b) logički pomak udesno za 2 bita

00001100

c) aritmetički pomak u desno za 2 bita

00001100

d) rotacija u lijevo za 3 bita

10001001

e) rotacija u desno za 4 bita

00010011

e) rotacija u desno za 2 bita, ako je prethodno zastavica kroz koju se rotacija izvodi jednaka 0

10001100, novo stanje zastavice je 0