

Laboratorio 1 – Esercizio 3

Incominciamo dalla rappresentazione in binario dei due numeri.

(op1) 150 = 1001 0110

(op2) 100 = 0110 0100

OP2 ha il MSB a 0 per cui non da problemi

OP1 ha il MSB a 1 quindi se lo si tratta come numero, con il conseguente uso della rappresentazione in CA2 dove il MSB rappresenta il segno, il suo corretto valore è -106 e la successiva somma con il numero 100 avrà come risultato -6

```
lb $t0, op1
```

```
lb $t1, op2
```

```
R4  [a0] = ffffffff
```

```
R8  [t0] = ffffffff96
```

```
R9  [t1] = 64
```

Se ora trattiamo OP1 come binario, tramite la funzione Unsigned, le cose cambiano.

Ricordiamoci che stiamo lavorando su 8 bit, quindi se ci togliamo l'incomodo del segno, che viene esteso sui rimanenti 24 bit, qualunque sia il valore memorizzato negli 8 bit (fino a 255) può essere tranquillamente rappresentato sui 32 bit, l'uso quindi dell'estensione U nelle successive operazioni (LB e ADD) non cambia nulla.

```
lbu $t0, op1
```

```
lb $t1, op2
```

```
add $a0, $t0, $t1
```

```
R4  [a0] = fa
```

```
R8  [t0] = 96
```

```
R9  [t1] = 64
```

Quindi il discriminare sta nella "gestione" di OP1, se Signed o Unsigned