

## Esercizi logisim

### Esercizio 1

- Realizzare un circuito per la somma tra 2 numeri binari di 4 bit (senza segno) utilizzando 4 sommatore binari in parallelo.
- Ampliare il circuito precedente con un input binario S che determina se i numeri in input vadano somati o sottratti. *Suggerimento:  $A - B = A + (\text{complemento a 2 di } B)$*

### Esercizio 2

Realizzare un file register con 4 registri a 8 bit con un singolo input e un singolo output. Il circuito deve comprendere un bit per write enable, uno per read enable e un indice da due bit per selezionare il registro da utilizzare.

## 1 Esercizi Assembly

### Esercizio 1

Tradurre la seguente istruzione in codice macchina: Sottrarre a R2 il valore in memoria all'indirizzo 0xA00 e, se il risultato è uguale al valore in R1, salvare il valore di R0 in memoria all'indirizzo 0xB00.

### Esercizio 2

Tradurre in linguaggio umano le seguenti espressioni assembly:

```
MOV    R3, #0x070
LDR     R0, [R3]
MOV     R3, #0x700
LDR     R1, [R3]
SUB     R2, R0, R1
MOV     R3, #0x770
STR     R2, [R3]
```

```

                                CMP    R4, #1
                                BNE    opt2
                                MOV    R0, #0
                                MOV    R1, #1
                                MOV    R2, #2
                                B       exit
opt2  MOV    R0, #0
      MOV    R1, #0
      MOV    R2, #0
exit
```

```

                                MOV    R4, #0xAB0
                                LDR    R2, [R4]
                                CMP    R2, #0
                                BLT    opt2
                                ORR    R3, R0, R1
                                B       save
opt2  EOR    R3, R0, R1
save  MOV    R4, #0xBA0
      STR    R3, [R4]
```

### Esercizio 3

Dato il seguente codice in C, tradurre il frammento in codice assembly.

```
int main(int argc, char** argv){

    int sum = 0;

    for(int i = 0; i < 5; i++){
        sum += i * i;
    }

    return 0;
}
```