

### **Esercizio 1**

- Si progetti un addizionatore HW operante su 3 bit (ogni operando ha 3 bit – in uscita il risultato avrà 3 bit); si includa un circuito per la detezione dell'overflow.
- Hint -> per la progettazione del circuito, si creino prima i moduli half-adder e full-adder e li si accorpino in maniera opportuna.
- Hint -> si salvi il circuito per gli esercizi successivi.

### **Esercizio 2**

- Si modifichi il circuito precedente per effettuare la differenza in complemento a due tra due numeri a 3 bit. Si modifichi (solo se necessario) il circuito di detezione dell'overflow.
- Hint -> per effettuare il complemento a due negando il numero in ingresso è necessario negare i bit e aggiungere 1. Si utilizzino in modo opportuno i moduli full-adder a tale scopo.
- Hint -> si salvi il circuito per gli esercizi successivi.

### **Esercizio 3**

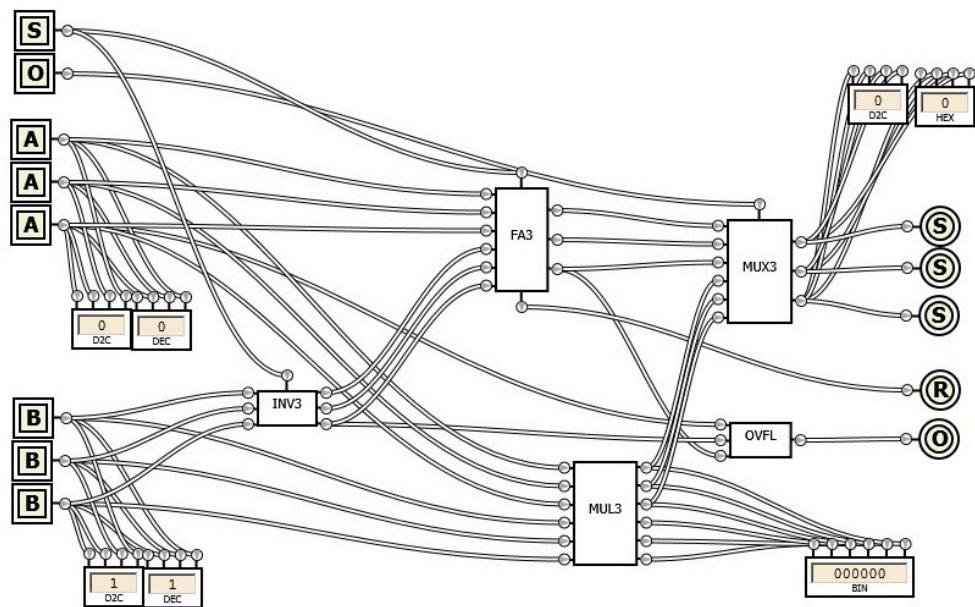
- Si progetti un moltiplicatore HW tra due numeri a 3 bit, utilizzando in modo opportuno i moduli half-adder e full-adder precedentemente realizzati. E' necessario in questo caso un modulo di detezione dell'overflow?
- Hint -> si salvi il circuito per gli esercizi successivi.

### **Esercizio 4**

- Si utilizzino i circuiti realizzati negli esercizi precedenti per realizzare una mini-alu in grado di realizzare operazioni di somma, sottrazione e prodotto tra due numeri a 3 bit. L'operazione da realizzare viene selezionata dall'utente.
- Hint -> si utilizzino dei moduli multiplexer / demultiplexer per la selezione dell'operazione scelta dall'utente (somma, sottrazione, prodotto) e per la selezione del risultato in uscita.

Soluzione:

La ALU a 3 bit finale può essere realizzata come segue:

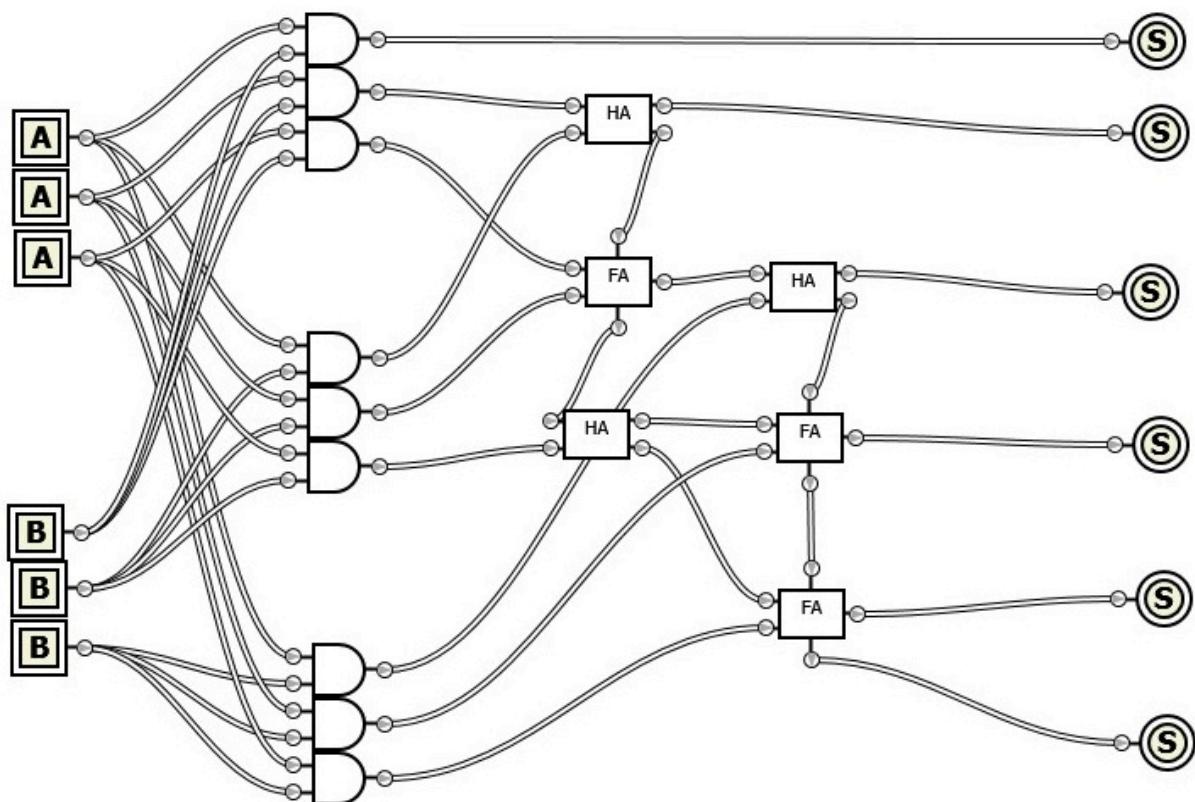


Per la realizzazione dei moduli somma su tre bit, inversione bit a bit su tre bit, mux su tre bit,circuito di overflow si faccia riferimento alle esercitazioni precedenti.

Per realizzare il modulo moltiplicatore su tre bit si consideri lo svolgimento dell'usuale algoritmo della moltiplicazione realizzato in base 2:

$$\begin{array}{r}
 \begin{array}{ccc} A_2 & A_1 & A_0 \end{array} * \\
 \begin{array}{ccc} B_2 & B_1 & B_0 \end{array} = \\
 \hline
 \begin{array}{ccc} R_{13} & R_{12} & R_{11} \\ A_2B_0 & A_1B_0 & A_0B_0 \\ + \\ A_2B_1 & A_1B_1 & A_0B_1 \end{array} = \\
 \hline
 \begin{array}{ccccc} R_{23} & R_{22} & R_{21} \\ X_4 & X_3 & X_2 & X_1 & X_0 \\ + \\ A_2B_2 & A_1B_2 & A_0B_2 \end{array} = \\
 \hline
 \begin{array}{ccccccc} z_5 & z_4 & z_3 & z_2 & z_1 & z_0 \end{array}
 \end{array}$$

Il modulo può essere realizzato come segue:



Il circuito computa correttamente le moltiplicazioni 3bitx3bit ->6bit senza segno. In aggiunta computa correttamente anche le moltiplicazioni 3bitx3bit->3bit con segno purché il risultato sia rappresentabile con 3bit segno compreso.