

# Relazione progetto Architettura degli Elaboratori

Studente: Davide Cortellucci  
Matricola: 260321

Specifiche progetto: moltiplicatore a due operandi interi maggiori uguali a 0 di 4 bit ciascuno ( $2^4 = 16$ , dunque cifre da 0 a 15) e 8 bit di risultato ( $2^8 = 256$ , quindi numeri da 0 a 255). Risultato massimo a fronte dei possibili ingressi: 225 ( $15 \times 15$ ); pertanto nessuna perdita di dati nella rappresentazione dell'output.

Legenda: gli operandi sono stati chiamati A3 A2 A1 A0 e B3 B2 B1 B0; A0 e B0 stanno ad indicare i bit MENO significativi, mentre A3 e B3 denotano i bit più significativi dei rispettivi operandi. Lo stesso ragionamento per la soluzione, rappresentata come S7 S6 S5 S4 S3 S2 S1 S0.

Per la realizzazione del progetto ho proceduto considerando che la moltiplicazione binaria si esegue come la moltiplicazione in decimale.

Bisogna moltiplicare ogni cifra del primo operando con quella meno significativa del secondo, poi con la seconda meno significativa e così via; ogni prodotto tra operandi di 1 bit ciascuno è assimilabile all'AND, quindi basta eseguire l'AND tra ogni bit tenendo conto delle regole della moltiplicazione binaria, e infine sommare correttamente tutti i prodotti per fornire il risultato. Nella spiegazione procederò colonna per colonna, cioè partendo dalla colonna più a destra dell'immagine e passando alla colonna adiacente a sinistra.

- S0:

In qualunque caso l'AND tra A0 e B0 (prima colonna a destra nell'immagine) non deve essere sommato ad alcuna cifra, quindi il risultato è direttamente S0, cioè il bit meno significativo della soluzione, e non c'è riporto.

$$\begin{array}{r} 1100 \text{ x} \\ 1111 \text{ =} \\ 1100 \\ 1100 - \\ 1100 - \\ 1100 - \\ \hline 10110100 \end{array}$$

-S1:

Considerando che non c'è riporto in entrata e che gli operandi da sommare sono 2, è bastato realizzare un modulo (FAdd) a 2 entrate e 2 uscite, cioè S1 e il Cout. FAdd somma A1B0 e A0B1.

$$\begin{array}{r} 1100 \text{ x} \\ 1111 \text{ =} \\ 1100 \\ 1100 - \\ 1100 - \\ 1100 - \\ \hline 10110100 \end{array}$$

-S2:

Ci troviamo nella terza colonna partendo da destra; qui bisogna sommare 3 operandi + il riporto di FAdD. Per lo scopo ho realizzato un modulo (FA3alt) a 4 entrate (3 operandi + Cin). Le uscite saranno necessariamente 3, poiché ci sono 4 bit da sommare e quindi nel caso massimo (tutti = 1) il risultato sarà 4, cioè 1 0 0, non rappresentabile con soli 2 bit.

Prendendo questo esempio in considerazione, FA3alt somma gli 1 e restituisce 1 0 0: lo 0 a destra è un output del moltiplicatore (S2), l'altro 0 è il primo riporto (C1), mentre l'1 è il secondo riporto (C2), che deve essere sommato nella quinta colonna. Nel caso del disegno in basso S2=1, C1=0, C2=0.

Gli operandi di FA3alt sono A2B0, A1B1 e A0B2 + il riporto di FAdD.

Il modulo in questione è stato realizzato con un Full Adder e due dei moduli che nel caso di S1 ho chiamato FAdD.

$$\begin{array}{r} 1100 \times \\ 1111 = \\ \hline 1100 \\ 1100 - \\ 1100 - \\ 1100 - \\ \hline 10110100 \end{array}$$

-S3:

Stavolta gli input sono 5 (4 operandi + il riporto C1 di FA3alt), perciò ho realizzato un modulo FA4a con 5 input e 3 output. Come per FA3alt, il bit meno significativo risultante dalla somma è S3, poi C1 e C2, quest'ultimo riportato nella sesta colonna.

Gli operandi in questo caso sono A3B0, A2B1, A1B2, A0B3 + il riporto C1 di FA3alt.

FA4a è stato realizzato utilizzando un FA3alt e 2 FAdD.

$$\begin{array}{r} 1100 \times \\ 1111 = \\ \hline 1100 \\ 1100 - \\ 1100 - \\ 1100 - \\ \hline 10110100 \end{array}$$

-S4:

Ci troviamo nella quinta colonna, ma anche stavolta serve un modulo identico al precedente (FA4b), dal momento che gli operandi sono 3, ma ci sono due riporti (il C2 di FA3alt e il C1 di FA4a).

FA4b ha lo stesso funzionamento del precedente, con la differenza che ci sono 2 riporti in entrata piuttosto che 1, dunque il bit meno significativo della somma è S4, poi abbiamo due riporti C1 e C2, rispettivamente da aggiungere alla sesta e settima colonna.

Tale modulo somma A3B1, A2B2, A1B3 + i due riporti sopracitati.

$$\begin{array}{r} 1100 \times \\ 1111 = \\ \hline 1100 \\ 1100 - \\ 1100 - \\ 1100 - \\ \hline 10110100 \end{array}$$

-S5:

Sesta colonna, gli operandi sono 2, ma anche i riporti, quindi ho optato per un altro FA3alt (che qui ho chiamato FA3b).

FA3b somma A3B2, A2B3, il primo riporto di FA4b e il secondo di FA4a e restituisce 3 output allo stesso modo degli altri moduli: S5 e 2 riporti (C1 e C2).

$$\begin{array}{r} 1100 \times \\ 1111 = \\ \hline 1100 \\ 1100 - \\ 1100 - \\ 1100 - \\ \hline 10110100 \end{array}$$

-S6:

Settima colonna, un solo operando ma 2 riporti; 3 input in totale e 2 output, quindi è sufficiente un Full Adder, che in questo caso ho chiamato FAb.

FAb somma A3B3, il C2 di FA4b e il C1 di FA3b, e restituisce S6 + riporto (Cout).

$$\begin{array}{r} 1100 \times \\ 1111 = \\ \hline 1100 \\ 1100 - \\ 1100 - \\ 1100 - \\ \hline 10110100 \end{array}$$

-S7:

Ultima colonna a sinistra, rimangono da sommare due input, cioè il secondo riporto di FA3b e quello di FAb. Normalmente servirebbe un addizionatore a due input e due output, ma in questo caso è sufficiente un OR, poiché non accade mai, date le specifiche, che i due riporti siano allo stesso tempo entrambi uguali a 1; ciò è dimostrato dal fatto che, a fronte di 2 ingressi a 4 bit ciascuno, il risultato massimo possibile è 225, mentre un output a 8 bit sarebbe sufficiente a rappresentare numeri fino a 255. Dunque non c'è riporto (in quel caso sarebbe un output a 9 bit).

Al di fuori del modulo MUL ho raggruppato gli operandi in un bus a 4 bit ciascuno, agganciando un DIP switch a testa, mentre per il risultato ho adottato un bus ad 8 bit agganciato ad un LED che converte il tutto in decimale.