

Esercitazione di Elettronica Digitale

Ing. F. Iannuzzo - A.A. 2004/2005

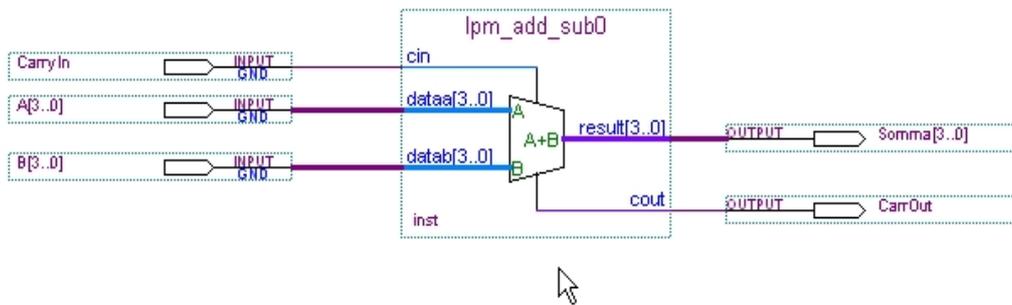
SOMMATORE A 4 e 8 BIT.

Introduzione.

Nella presente esercitazione si realizzerà un sommatore con tecnica FPGA. Come è noto, tale circuito esegue la somma di due addendi rappresentati con stringhe binarie. Si vedrà dapprima la realizzazione a 4 bit e poi quella a 8 bit come cascata di due sommatore a 4 bit.

Progetto del Sommatore a 4 bit.

Lo schema da realizzare è il seguente:



Il blocco centrale è un elemento sommatore presente in libreria e presenta i seguenti pin:

- dataa[3..0]:** bus a 4 linee che identifica i 4 bit del primo addendo. Convenzione: dataa0 è il bit meno significativo (MSB);
- datab[3..0]:** bus a 4 linee che identifica i 4 bit del secondo addendo. Convenzione: datab0 è il bit meno significativo (MSB);
- cin:** riporto entrante (Carry In). Questo piedino è utilizzato quando si collegano più sommatore in cascata in modo tale da eseguire una somma con più bit (come si vedrà in seguito nel caso del sommatore a 8 bit).
- result[3..0]:** bus a 4 linee che identifica i 4 bit del risultato. Convenzione: result0 è il bit meno significativo (MSB);
- cout:** riporto uscente (Carry Out). Questo piedino assume valore alto quando il risultato dell'operazione non è rappresentabile con soli 4 bit (Condizione di overflow).

Per prima cosa si crei una cartella **C:\Elettronica**, e dentro se ne crei un'altra chiamata **Somma4bit**. In questa cartella si inseriranno tutti i file del progetto.



Alla fine dell'esperienza si copi questa cartella su un supporto di memorizzazione rimovibile (FLOPPY o PENDRIVE) in quanto allo spegnimento del PC si perderanno tutti dati salvati in precedenza.

Si apra l'ambiente Quartus II

Si faccia click su: File -> New Project Wizard, all'apertura della finestra, si scelga **Next**,



si scelga, facendo click su  nel primo campo, la directory **Somma4bit** precedentemente creata e si dia il nome "**somma4bit**" al progetto, specificandolo nel secondo e terzo campo. Si faccia click su **Next** per tre volte di seguito fino ad arrivare alla finestra seguente:



In questa finestra si specifichi la famiglia MAX3000A, si selezioni **Yes**, si prema su **Next**.

Si aprirà una finestra, si scelgano i seguenti dati:

- Si scelga dalla casella "PIN COUNT" il valore 100.
- Si scelga dalla casella "SPEED GRADE" il valore 10.
- Si scelga il CPLD EPM3128ATC100-10 dalla lista.

Si clicchi su **Finish**.

Si clicchi sull'icona  e si scelga Block Diagram / Schematics File.

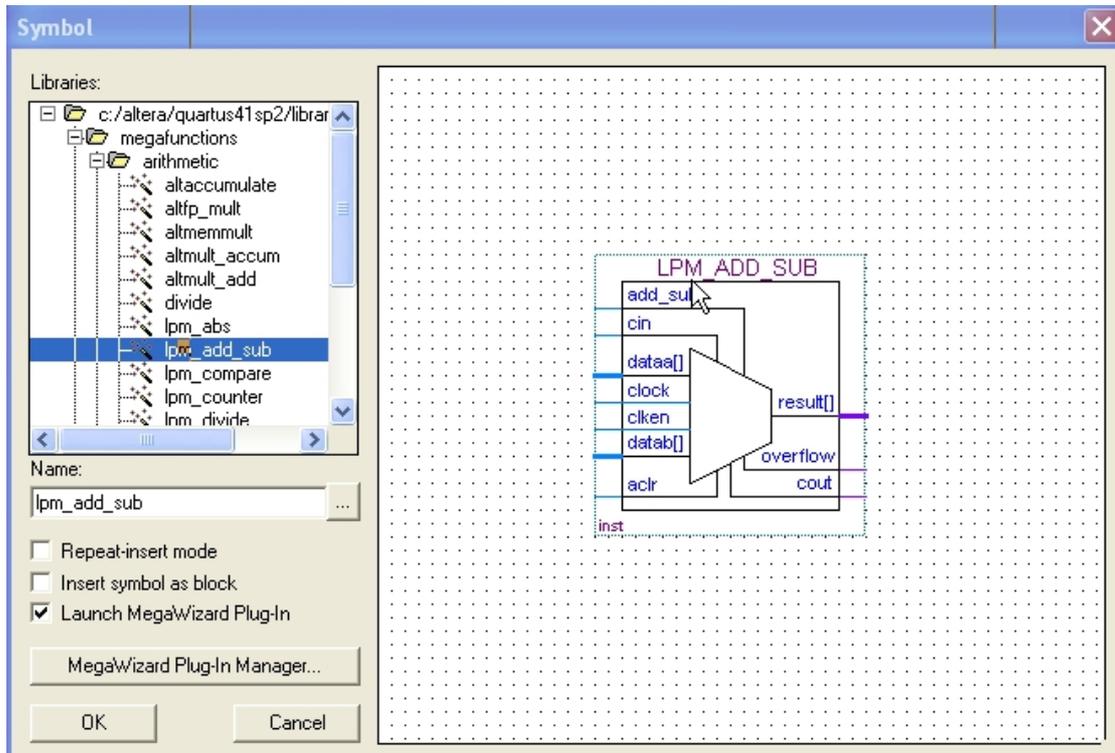
Si clicchi su **OK**

Si aprirà una finestra con una griglia, in cui si dovranno inserire i componenti illustrati in figura di pagina 1.

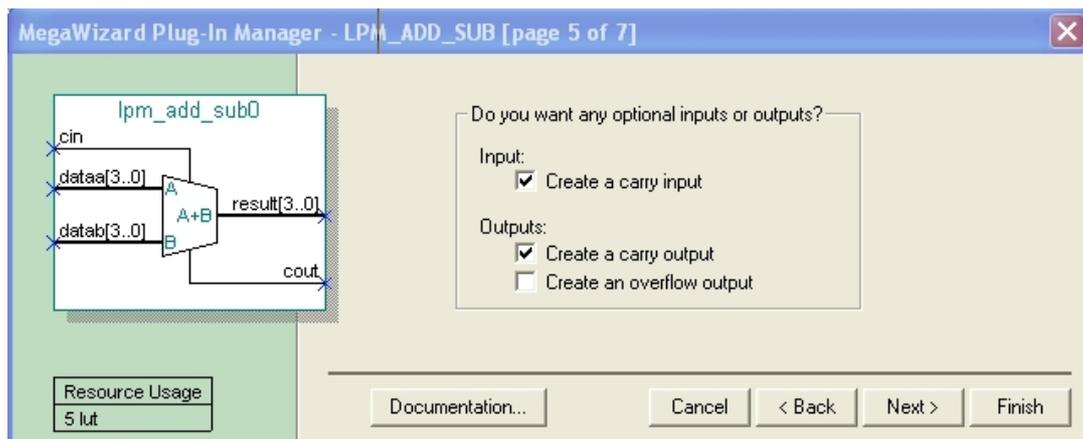
Si clicchi sull'icona . Si aprirà la finestra denominata "**Symbol**". Si apra la libreria cliccando

sull'icona  e si scelga Megafunction e la sottocartella Arithmetic, si scelga

"lpm_add_sub". Si disabiliti la funzione Repeat-insert mode, altrimenti si inseriranno più componenti e si controlli che sia abilitata la funzione LaunchMegaWizard Plug-In, come mostrato nella figura qui sotto.



Si faccia click su OK: si aprirà il wizard. Si imposti AHDL come tipo di output file e si preme Next. Si scelga il numero di bit da realizzare, nel nostro caso 4 bit, si selezioni Addition only e si preme Next per due volte fino ad arrivare alla schermata seguente:



Si spuntino le opzioni Create a carry input e Create a carry output e si clicchi su Finish.

Si inserisca il blocco creato.

Libraries:

Si apra la libreria cliccando sull'icona . Si disabiliti la funzione Repeat-insert mode e si scelga dal gruppo Primitives, la sottocartella Pin, qui si scelga "Input". Si inseriscano nello schema i pin di input e si colleghino ai rispettivi pin del blocco "lpm_add_sub".

Si rinominino i pin di input nel modo seguente: A[3..0] per il dato A, B[3..0] per il dato B e CarryIn per il pin cin. Si scelga anche l'inserimento di pin "Output" e si colleghino al bus Somma[3..0] e al CarryOut (per rinominare il pin basta cliccare sul nome del pin). Per effettuare il collegamento tra i componenti si avvicini il puntatore del mouse all'ingresso o all'uscita degli stessi. Si noti il cambiamento del cursore. A questo punto, tenendo premuto il pulsante sinistro del mouse, si tracci il collegamento tra i componenti.



Si noti che i pin di ingresso presentano un valore di default (VCC o GND) modificabile con doppio click. Questo valore specifica al compilatore quale livello logico assegnare al pin come default.

Si imposti il valore di default di tutti gli ingressi a VCC. Fatto ciò si salvi il file e si proceda alla compilazione tramite l'icona  posta in alto sulla barra dei comandi.

Finita la compilazione, si visualizzi il Report della compilazione. Le informazioni che vi sono indicate riguardano le fasi della compilazione. Le fasi sono le seguenti:

Analysis and Synthesis: in questa fase si analizza il progetto e viene generata la Netlist che specifica i collegamenti tra i vari componenti.

Fitter: in questa fase il compilatore distribuisce nello spazio del CPLD le macrocelle contenenti i componenti del progetto.

Assembler: in questa fase il compilatore assembla il file per la programmazione del CPLD.

Timing Analyzer: in questa fase il compilatore analizza i tempi di propagazione nel CPLD, calcola il tempo peggiore e il tempo migliore specificando il percorso per cui si hanno.

Simulazione del Progetto.

Con N bit si possono rappresentare i numeri da 0 a $(2^N - 1)$, nel nostro caso con 4 bit si rappresentano i numeri decimali da 0 a 15.

Come esempio effettueremo le somme di $(7 + 5)$ e di $(11 + 8)$.

Il numero 7 è rappresentato da 0111 in binario; il numero 5 da 0101, il numero 11 è rappresentato da 1011 e 8 da 1000. Se si effettua la somma si ottiene:

DECIMALE	BINARIO
7 +	0111 +
5 =	0101 =
<hr/>	<hr/>
12	1100
11 +	1011 +

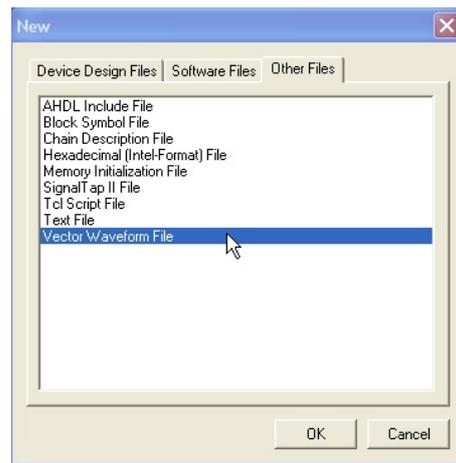
$$\frac{8}{19} = \frac{1000}{0011} \quad \text{il numero rappresentato è 3}$$



CARRY OUT = 1 rappresenta 2^4 , cioè 16

Tenendo conto anche del Carry Out il risultato è 10011, cioè 19. Per essere rappresentato abbiamo, però bisogno di 5 bit (condizione di overflow).

Per creare un nuovo file di tipo simulazione, si clicchi su  e si scelga Vector Waveform File dal sottomenu Other Files e si clicchi su **OK**.

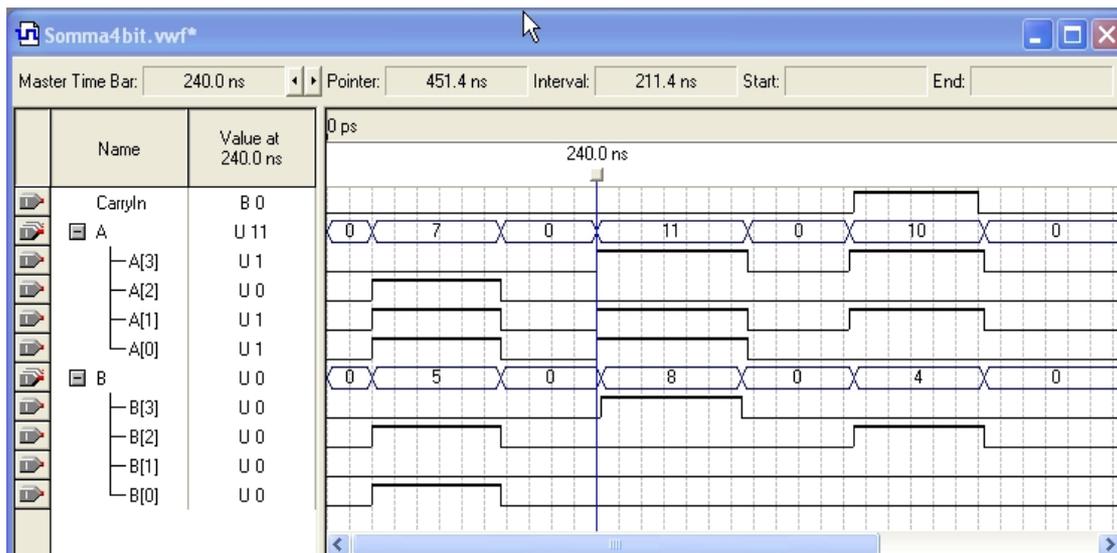


Si aprirà un'ulteriore finestra. Si faccia doppio clic nella parte sinistra (sotto la dicitura Name) e si scelga **Node Finder...** Si utilizzi il menù filter e si scelga "PINS: all" e poi **List**. Apparirà la lista di tutti i pin del progetto. Si selezioni dapprima il pin CarryIn, si clicchi su **OK** e si imposti, nell'opzione Radix il tipo di dato Binary. Poi si ripeta il procedimento e si selezionino gli ingressi A e B e l'output Somma, si clicchi su **OK** e si imposti il tipo di dato in Radix al valore Unsigned Decimal. Si faccia click su **OK**.



E' utile osservare l'intero intervallo temporale. Per ottenere ciò, si faccia click sullo strumento Zoom Tool  e si clicchi ripetutamente col tasto destro del mouse al centro dell'area di lavoro.

Si selezioni con il puntatore un intervallo di tempo sull'input A, si clicchi sull'icona . Si aprirà una finestra; si imposti il valore 7 e si clicchi su **OK**. In questo modo, si sta indicando al simulatore che l'ingresso A deve assumere per quell'intervallo il valore decimale 7. Si selezioni il medesimo intervallo di tempo sull'ingresso B. Si imposti il valore decimale 5 e si clicchi su **OK**. Si ripetano le operazioni selezionando altri intervalli di tempo più a valle e si impostino i valori 11 per A e 8 per B. Si provi, poi, ad impostare, in un altro intervallo altri due valori per i dati A e B (si scelgano a piacere) ma si setti, per l'intervallo considerato, il CarryIn al valore 1.



Si consideri che il CarryIn è un ingresso Binario, quindi non può essere impostato con l'icona  ma con l'icona .

Nella figura sopra si può notare che gli input A e B, essendo bus a 4 linee, se esplosi presentano i valori binari di ogni bit usato per la rappresentazione del numero decimale.

Si salvi il file delle forme d'onda e si clicchi sull'icona  per la compilazione; se non ci sono errori passate alla simulazione cliccando su Tools -> Simulator Tool.

Si imposti il tipo di simulazione tra le due a disposizione.

Functional: simulazione ideale, non tiene conto dei tempi di propagazione.

Timing: simulazione reale, tiene conto dei tempi di propagazione.

Si scelga per prima la simulazione Functional, e si clicchi su **Generate Functional Simulation Netlist**. Si selezioni il file delle forme d'onda (.vwf), e si dia **Start** per iniziare la simulazione. Se non si sono avuto errori, si clicchi su **Report**.

Nella finestra che si aprirà si visualizzeranno gli ingressi da noi impostati e le uscite calcolate dalla simulazione. Si verifichi se il risultato è quello voluto.

Si ripeta la simulazione scegliendo questa volta il tipo Timing, si dia **Start** e si visualizzi, cliccando su **Report**, il risultato ottenuto in modo da confrontare le due simulazioni e notare le differenze.

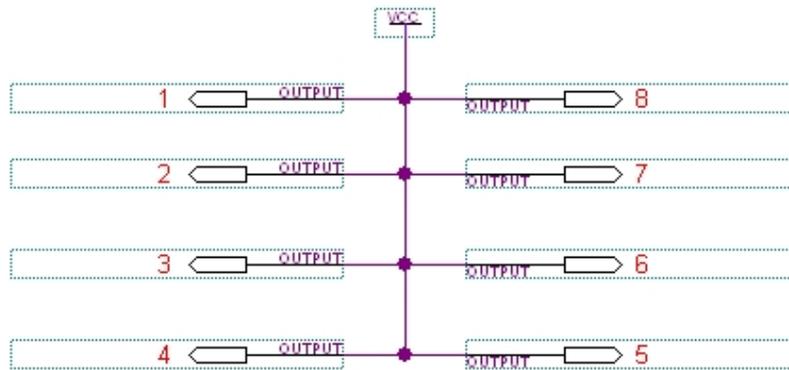


Cosa è successo quando si è impostato il CarryIn ad 1 ?

Programmazione della basetta.

Tra poco si programmerà la basetta ma dobbiamo realizzare una rete di pull-up che servirà alla verifica sperimentale del risultato.

Si realizzi nel file schematics la rete mostrata in figura:



Essa è costituita da 8 pin di output collegati tutti a Vcc. (Si usi lo strumento  per riflettere i componenti).

Si compili il progetto e si passi all'assegnazione dei pin.

Cosa significa assegnare i pin?

Bisogna comunicare all'hardware programmabile quali sono gli input e gli output corrispondenti sulla basetta. Come facciamo ad impostare i due addendi in ingresso? E l'uscita?

L'idea è quella di impostare il bit di carry al pulsante SW3, i bit di A e B a 8 pin sulla morsettiera J2, collocata a sinistra del CPLD, i bit corrispondenti alla somma a 4 segmenti dei display e il bit Carry Out al DP (punto decimale del display).

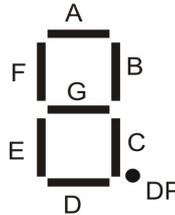
Si entri nel menù `Assignments -> Pins`, si aprirà una finestra, si faccia click sull'icona  posta in alto a destra.

- a) Si faccia doppio click su **new** sotto la colonna **To** ; si aprirà un menù a tendina; si scelga il pin A0.
- b) Si faccia doppio click su **new** sotto la colonna **Location**; si aprirà un menù a tendina, si scelga il Pin_2.

Si ripetano le operazioni ai punti a) e b) per gli altri pin, rispettando le assegnazioni descritte nella tabella seguente.

Corrispondenza Pin progetto - Pin basetta

A[0]	PIN_2
A[1]	PIN_6
A[2]	PIN_8
A[3]	PIN_10
CarryIn	PIN_27
B[0]	PIN_13
B[1]	PIN_16
B[2]	PIN_19
B[3]	PIN_21
Somma[0]	PIN_77
Somma[1]	PIN_76
Somma[2]	PIN_71
Somma[3]	PIN_72
CarryOut	PIN_79



Segmento	PIN
A	70
B	71
C	72
D	75
E	76
F	77
G	80
DP	79

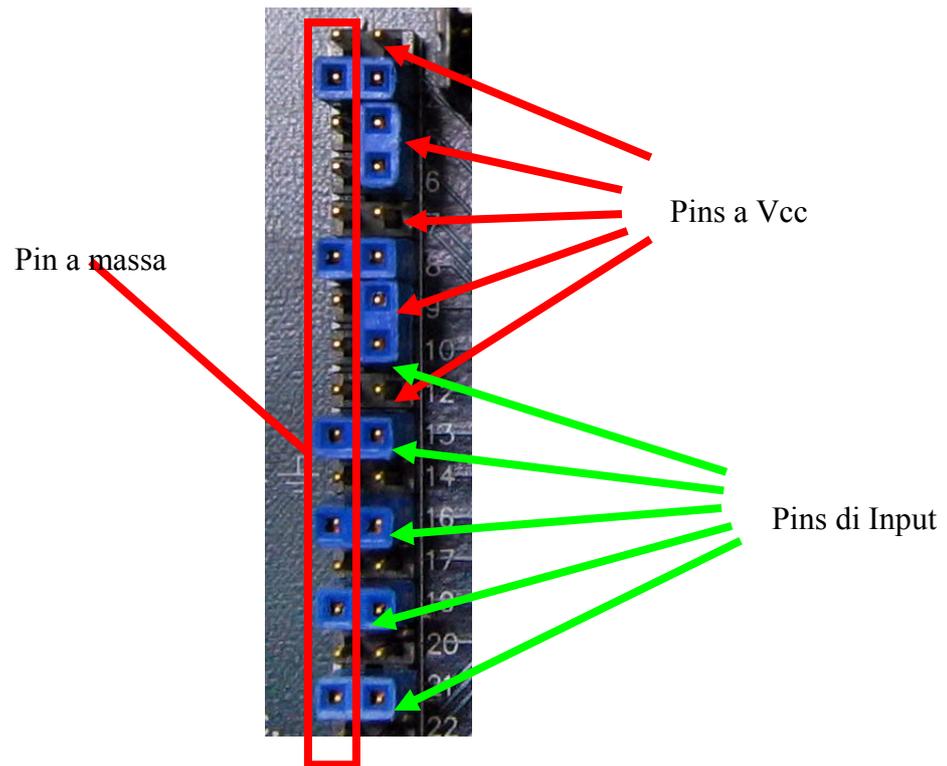
Si assegnino i pin della rete con le uscite a Vcc ai pin del CPLD 1, 5, 7, 9, 12, 14, 17 e 20.

Fatta l'assegnazione dei pin si salvi il file e si compili di nuovo il progetto. Se non si hanno errori, si vada nella cartella sul desktop **Eledig\Somma4bit** e si copi il file "**Somma4bit.pof**" su un floppy. Questo file servirà alla programmazione del CPLD.

Verifica del progetto.

Si apra Quartus II e si faccia click sull'icona . Nella finestra che si aprirà, con il comando **Add file...** si specifichi il file .pof salvato poco fa sul floppy. Si scelga l'opzione Program / Configure e si click su **Start** per iniziare la programmazione.

Per impostare i due addendi, A e B, si facciano dei ponticelli con i jumper per ogni pin di input, prima assegnato, e lo si ponticelli in orizzontale per impostare il bit corrispondente a 0, mentre lo si colleghi in verticale per assegnare il bit a 1. Un esempio si può vedere nella figura qui sotto.

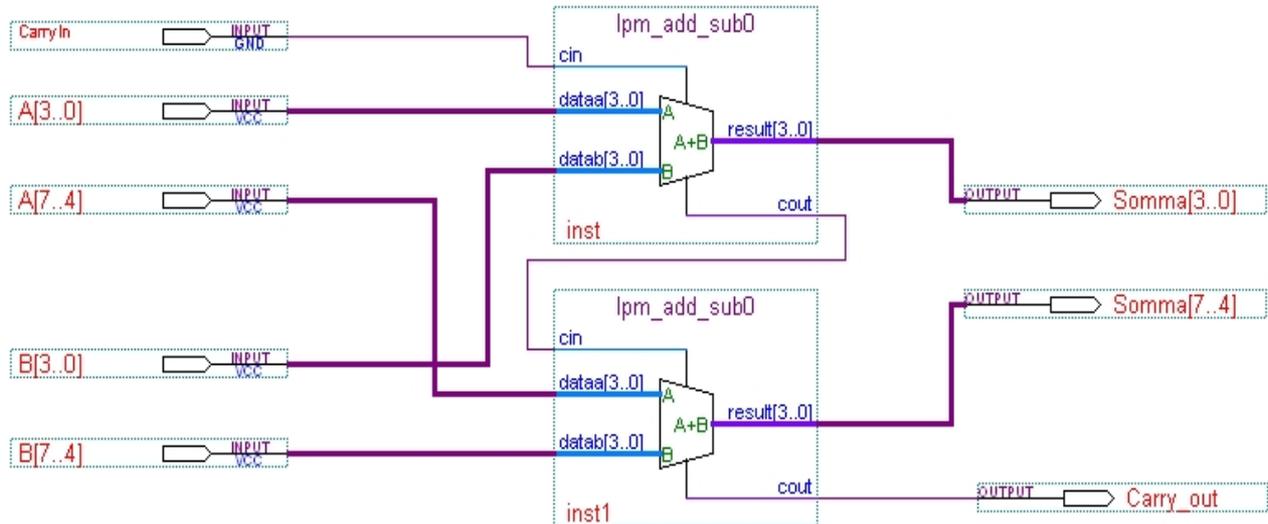


Ora si verifichi il funzionamento del sommatore, scegliendo le opportune combinazioni dei livelli logici sui pin di Input in modo tale da rappresentare gli addendi A e B in binario. Si facciano varie prove ed ad ogni prova si controlli che i display visualizzino correttamente il risultato.

Si provi anche a porre il bit di carry in a 1 e si controlli cosa accade alla somma.

Progetto del Sommatore a 8 bit.

Il sommatore ad otto bit sarà realizzato con la cascata di due sommatori a 4 bit. Lo schema da realizzare è il seguente:

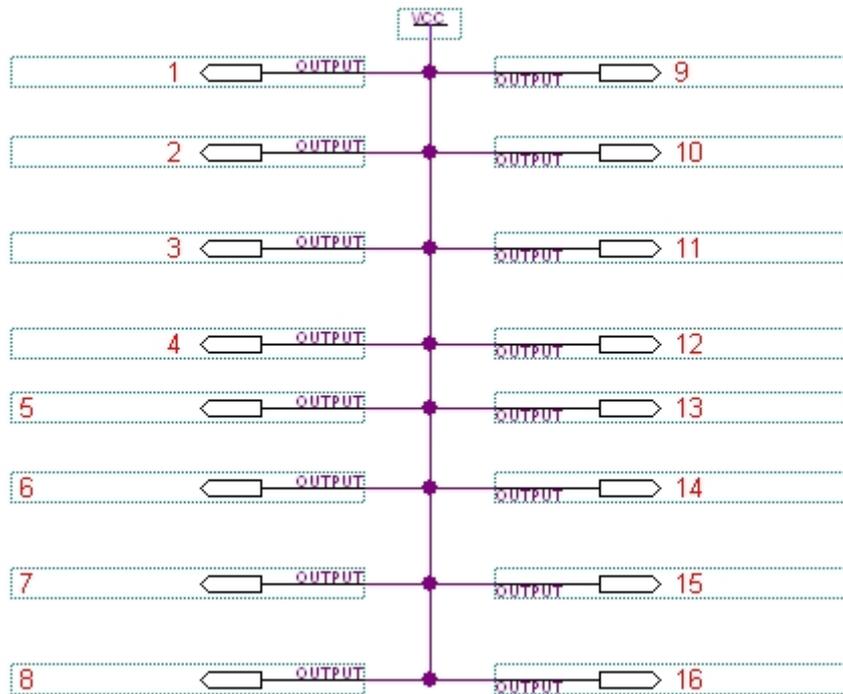


Nel blocco centrale ci sono 2 sommatori e dallo schema si notano i seguenti pin:

- **A[3..0]**: rappresenta un bus con 4 linee che identifica i 4 bit meno significativi del primo addendo, A[0] è il bit meno significativo (MSB);
- **A[7..4]**: rappresenta un bus con 4 linee che identifica i 4 bit più significativi del primo addendo, A[4] è il bit meno significativo (MSB);
- **B[3..0]**: rappresenta un bus con 4 linee che identifica i 4 bit meno significativi del secondo addendo, B[0] è il bit meno significativo (MSB);
- **B[7..4]**: rappresenta un bus con 4 linee che identifica i 4 bit più significativi del secondo addendo, B[4] è il bit meno significativo (MSB);
- **Carry in**: rappresenta il riporto entrante. Nel secondo sommatore, quello che somma la parte più significativa dei 2 addendi, esso è collegato al carry out del primo sommatore.
- **Somma[3..0] e Somma[7..4]**: rappresenta 2 bus con 4 linee che identificano insieme gli 8 bit del risultato, result0 è il bit meno significativo (MSB) della somma;
- **Carry out**: rappresenta il riporto uscente, esso indica se la somma fornisce un numero non rappresentabile con i bit a disposizione.

Si crei un'altra cartella in **Elettronica**, chiamandola **Somma8bit**. In questa cartella si inseriranno tutti i file del progetto.

Si svolgano le operazioni fatte per il sommatore a 4 bit e si realizzi in schematics lo schema precedentemente illustrato.
 Ora nello stesso foglio schematics si realizzi la rete mostrata nella figura qui sotto e si rinominino i pin di output nel modo indicato.



Fatto ciò si salvi il file e si proceda alla compilazione tramite l'icona  posta in alto sulla barra dei comandi.

Finita la compilazione si visualizzi il Report della compilazione.

Simulazione del Progetto.

Come fatto per il sommatore a 4 bit si imposti un file di forme d'onda e si svolgano i due tipi di simulazione e si verifichi se il progetto svolge la funzione voluta, notando le differenze tra i due tipi di simulazione.

Programmazione della basetta.

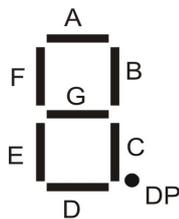
Si assegnino i pin alla basetta utilizzando il metodo usato per il sommatore a 4 bit, le assegnazioni da fare sono riportate nelle tabelle qui sotto.

Corrispondenza Pin progetto - Pin basetta

A[0]	PIN_2
A[1]	PIN_6
A[2]	PIN_8
A[3]	PIN_10
A[4]	PIN_13
A[5]	PIN_16
A[6]	PIN_19
A[7]	PIN_21
CarryIn	PIN_27
B[0]	PIN_30
B[1]	PIN_32
B[2]	PIN_37
B[3]	PIN_41
B[4]	PIN_44
B[5]	PIN_46
B[6]	PIN_48
B[7]	PIN_50
CarryOut	PIN_100
Somma[0]	PIN_70
Somma[1]	PIN_71
Somma[2]	PIN_72
Somma[3]	PIN_75
Somma[4]	PIN_76
Somma[5]	PIN_77
Somma[6]	PIN_80
Somma[7]	PIN_79

Corrispondenza Pin progetto - Pin basetta

1	PIN_1
2	PIN_5
3	PIN_7
4	PIN_9
5	PIN_12
6	PIN_14
7	PIN_17
8	PIN_20
9	PIN_29
10	PIN_31
11	PIN_36
12	PIN_40
13	PIN_42
14	PIN_45
15	PIN_47
16	PIN_49



Segmento	PIN
A	70
B	71
C	72
D	75
E	76
F	77
G	80
DP	79

Fatta l'assegnazione dei pin si salvi il file e si compili di nuovo il progetto. Se non si hanno errori si vada nella cartella sul desktop **Eledig\Somma8bit** e si copi il file .pof, generato dalla compilazione, su un floppy. Questo file servirà alla programmazione del CPLD.

Verifica del progetto.

Si apra Quartus II e si faccia click sull'icona . Nella finestra che si aprirà, con il comando **Add file...** si specifichi il file .pof salvato poco fa sul floppy. Si scelga l'opzione Program / Configure e si click su **Start** per iniziare la programmazione.



Si eseguano diverse prove, sommando vari addendi, impostando di volta in volta i jumper sui pin, così come si è fatto nel sommatore a 4 bit. Si provi anche a premere il tasto SW3 in modo da impostare il carry in a 1 e si veda quello che accade alla somma.

Ringraziamenti.

Desidero ringraziare il Dott. Paolo Mele, laureato di questa Facoltà, per l'insostituibile contributo dato alla preparazione di questa e delle altre esercitazioni sugli hardware programmabili, nell'ambito del suo lavoro di tesi di laurea.