Reti Logiche
Appello del 9 gennaio 2007 – Seconde prove

(D2) Qualunque funzione di commutazione di due variabili $f(x,y)$ può essere espressa nella forma

$$f(x,y) = a \oplus bx \oplus cy \oplus dxy$$

Ricavare i coefficienti *a*, *b*, *c*, *d* in funzione dei valori assunti da $f(x,y)$.

◆ ◆ ◆ ◆ ◆ ◆

Sostituendo nell'espressione le quattro possibili combinazioni di *x*, *y* si ottiene:

$$\begin{aligned}
f(0,0) &= a \oplus (b\cdot 0) \oplus (c\cdot 0) \oplus (d\cdot 0\cdot 0) = a \\
f(0,1) &= a \oplus (b\cdot 0) \oplus (c\cdot 1) \oplus (d\cdot 0\cdot 1) = a \oplus c \\
f(1,0) &= a \oplus (b\cdot 1) \oplus (c\cdot 0) \oplus (d\cdot 1\cdot 0) = a \oplus b \\
f(1,1) &= a \oplus (b\cdot 1) \oplus (c\cdot 1) \oplus (d\cdot 1\cdot 1) = a \oplus b \oplus c \oplus d
\end{aligned}$$

In primo luogo si ha:

$$a = f(0,0)$$

Da $f(0,1) = a \oplus c$ segue $a \oplus f(0,1) = a \oplus a \oplus c = c$ e dunque

$$c = a \oplus f(0,1) = f(0,0) \oplus f(0,1)$$

Da $f(1,0) = a \oplus b$ segue $a \oplus f(1,0) = a \oplus a \oplus b = b$ e dunque

$$b = a \oplus f(1,0) = f(0,0) \oplus f(1,0)$$

Da $f(1,1) = a \oplus b \oplus c \oplus d$ segue $a \oplus b \oplus c \oplus f(1,1) = a \oplus b \oplus c \oplus a \oplus b \oplus c \oplus d = d$ e dunque

$$d = a \oplus b \oplus c \oplus f(1,1) = f(0,0) \oplus f(0,1) \oplus f(1,0) \oplus f(1,1)$$

In definitiva:

$$\begin{aligned}
a &= f(0,0) \\
b &= f(0,0) \oplus f(1,0) \\
c &= f(0,0) \oplus f(0,1) \\
d &= f(0,0) \oplus f(0,1) \oplus f(1,0) \oplus f(1,1)
\end{aligned}$$

A titolo di esempio, i coefficienti per la funzione $f(x,y) = x + \overline{y}$ sono:

$$\begin{aligned}
a &= f(0,0) = 1 \\
b &= f(0,0) \oplus f(1,0) = 1 \oplus 1 = 0 \\
c &= f(0,0) \oplus f(0,1) = 1 \oplus 0 = 1 \\
d &= f(0,0) \oplus f(0,1) \oplus f(1,0) \oplus f(1,1) = 1 \oplus 0 \oplus 1 \oplus 1 = 1
\end{aligned}$$

con conseguente espansione

$$x + \overline{y} = 1 \oplus y \oplus xy$$

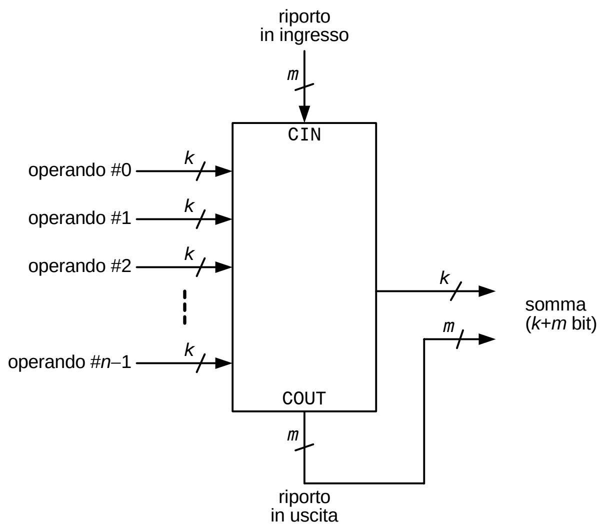
Infatti

$$\begin{aligned}
1 \oplus y \oplus xy &= \overline{y} \oplus xy = \\
&= xy + \overline{y}(\overline{x} + \overline{y}) = \\
&= xy + \overline{y} = \\
&= x + \overline{y}
\end{aligned}$$

(D3) Definire la struttura di una rete iterativa per la somma aritmetica di *tre* numeri binari, e sviluppare il progetto della cella elementare usando solo porte NAND.

◆ ◆ ◆ ◆ ◆ ◆

Occorre innanzitutto stabilire quanti bit di carry (riporto) devono lasciare la singola cella ed entrare nella cella immediantamente successiva. Si ossrvi innanzitutto che in un addizionatore multi-operando i bit di carry devono codificare il riporto come *numero binario*, in modo che la somma finale ottenuta possa essere espressa non solo dai bit di somma generati dalle singole celle, ma anche dai bit di carry-out della cella più significativa (riporto finale in uscita). Nella figura che segue è illustrato un addizionatore a n operandi da k bit ciascuno, con m bit di carry: la somma risultante deve poter essere espressa su $k+m$ bit.



Poiché il massimo valore del carry è $2^m - 1$ e il massimo valore di un operando è $2^k - 1$, il massimo valore della somma è allora

$$S_{\max} = n(2^k - 1) + 2^m - 1$$

Questo valore deve poter essere espresso su $k+m$ bit, ossia deve essere $S_{\max} \le 2^{k+m} - 1$. In definitiva, m deve assumere il minimo valore intero che soddisfa la disequazione

$$n(2^k - 1) + 2^m \le 2^{k+m}$$

da cui

$$2^m (2^k - 1) \ge n(2^k - 1)$$

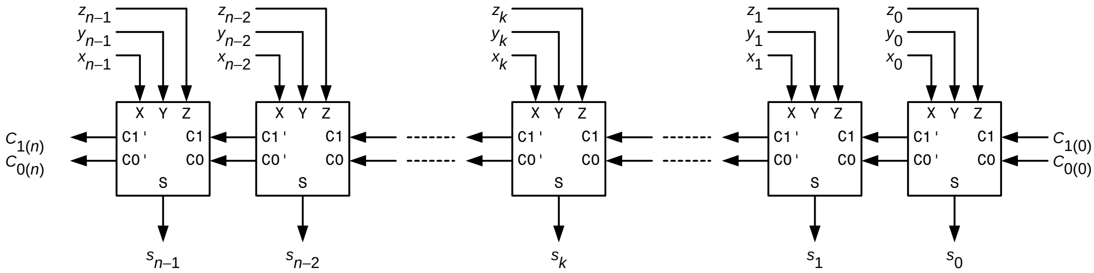
e infine

$$2^m \ge n$$

Se nel normale Full Adder a due operandi (n = 2) è sufficiente un solo bit di carry (m = 1), l'equazione mostra che in una cella di addizione per tre operandi (n = 3) sono necessari m = 2 bit di carry per codificare il riporto tra una cella e l'altra. Per la singola cella, allora, detti c_1 e c_0 i bit di carry-in, il valore numerico del riporto in ingresso sarà $C = c_0 + 2c_1$; detti c_1' e c_0' i bit di carry-out, il valore numerico del riporto in uscita sarà $C' = c_0' + 2c_1'$; detti x, y, z i tre operandi da un bit e s il bit di somma in uscita, il valore numerico della somma generata dalla cella sarà

$$S = x + y + x + c_0 + 2c_1 = s + 2c_0' + 4c_1'$$

La rete iterativa avrà pertanto la struttura:



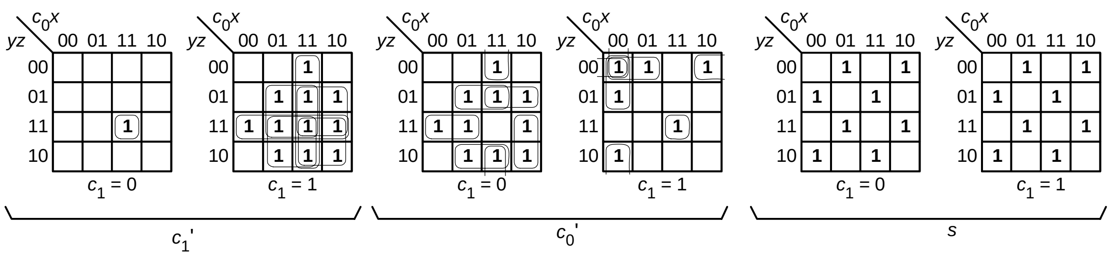
Il comportamento della singola cella è descritto dalla seguente tavola di verità:

c_1	c_0	x	y	z	C	T	S	c_1'	c_0'	s
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	1	1	0	0	1
0	0	0	1	0	0	1	1	0	0	1
0	0	0	1	1	0	2	2	0	1	0
0	0	1	0	0	0	1	1	0	0	1
0	0	1	0	1	0	2	2	0	1	0
0	0	1	1	0	0	2	2	0	1	0
0	0	1	1	1	0	3	3	0	1	1
0	1	0	0	0	1	0	1	0	0	1
0	1	0	0	1	1	1	2	0	1	0
0	1	0	1	0	1	1	2	0	1	0
0	1	0	1	1	1	2	3	0	1	1
0	1	1	0	0	1	1	2	0	1	0
0	1	1	0	1	1	2	3	0	1	1
0	1	1	1	0	1	2	3	0	1	1
0	1	1	1	1	1	3	4	1	0	0

c_1	c_0	x	y	z	C	T	S	c_1'	c_0'	s
1	0	0	0	0	2	0	2	0	1	0
1	0	0	0	1	2	1	3	0	1	1
1	0	0	1	0	2	1	3	0	1	1
1	0	0	1	1	2	2	4	1	0	0
1	0	1	0	0	2	1	3	0	1	1
1	0	1	0	1	2	2	4	1	0	0
1	0	1	1	0	2	2	4	1	0	0
1	0	1	1	1	2	3	5	1	0	1
1	1	0	0	0	3	0	3	0	1	1
1	1	0	0	1	3	1	4	1	0	0
1	1	0	1	0	3	1	4	1	0	0
1	1	0	1	1	3	2	5	1	0	1
1	1	1	0	0	3	1	4	1	0	0
1	1	1	0	1	3	2	5	1	0	1
1	1	1	1	0	3	2	5	1	0	1
1	1	1	1	1	3	3	6	1	1	0

dove C indica il valore numerico presente sui bit di carry input c_1, c_0 e T indica quanti dei bit x, y, z degli operandi sono diversi da zero; il valore numerico della somma è allora pari a $S = C + T$, che va rappresentato in binario sui bit di uscita c_1', c_0', s.

Le mappe di Karnaugh per $c_1{}', c_0{}', s$, con delle possibili coperture, sono le seguenti:



Da queste si possono ricavare le corrispondenti espressioni in somme di prodotti e infine le relative reti NAND-NAND a due livelli (più un livello di inversione per le variabili di ingresso). Si noti come le mappe per s siano identiche, il che significa che l'uscita s non dipende dall'ingresso c_1.

Si può inoltre ottenere una realizzazione non ottimale osservando che:

- La mappa per c'_0 e $c_1 = 0$ è esattamente complementare alla mappa per c'_0 e $c_1 = 1$;
- A meno del mintermine $c_0xyz = 1111$, la mappa per c'_1 e $c_1 = 1$ è esattamente identica alla mappa per c'_0 e $c_1 = 0$;
- Il mintermine $c_0xyz = 1111$ è comune a entrambe le mappe per c'_1.

Una soluzione alternativa per l'addizionatore a tre operandi prevede l'uso di normali Full Adder per la realizzazione della singola cella. In tal caso, tuttavia, occorre considerare con attenzione i *pesi* associati a ciascuna variabile di cella. Ricordando che $C = 2c_1 + c_0$, l'operazione da eseguire è

$$S = x + y + z + C = x + y + z + 2c_1 + c_0$$

I primi tre termini della somma possono essere applicati a un primo Full Adder (FA1) che genererà una somma parziale $S_1 = p_1 + 2q_1$, dove p_1 e q_1 sono rispettivamente il bit di somma e il bit di carry uscenti dal Full Adder; dunque la somma finale diventa

$$S = S_1 + 2c_1 + c_0 = p_1 + c_0 + 2(c_1 + q_1)$$

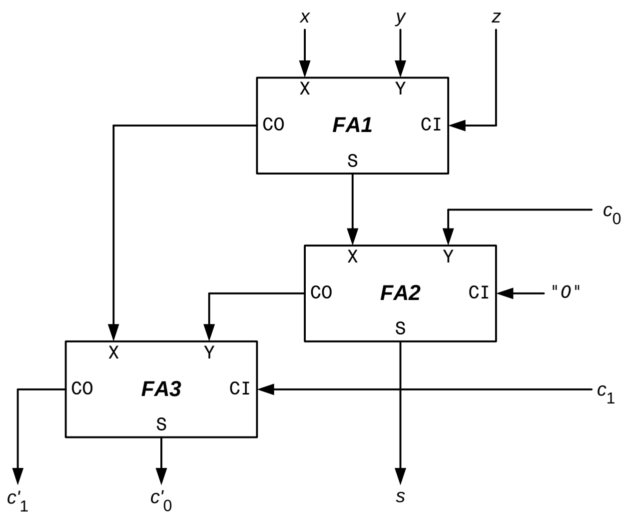
I primi due termini di quest'ultima espressione possono poi essere applicati a un secondo Full Adder (FA2) che produrrà una somma parziale $S_2 = p_2 + 2q_2$; l'espressione della somma S allora diventa

$$S = S_2 + 2(c_1 + q_1) = p_2 + 2(c_1 + q_1 + q_2)$$

I termini tra parentesi possono infine essere applicati a un terzo Full Adder (FA3) che produrrà una somma parziale $S_3 = p_3 + 2q_3$; la somma finale S diventa così

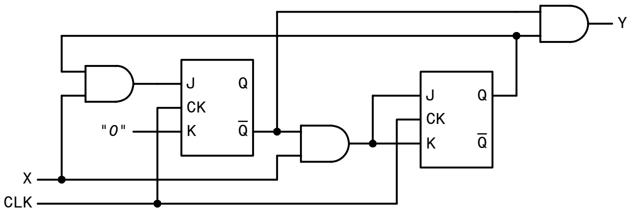
$$S = p_2 + 2S_3 = p_2 + 2(p_3 + 2q_3) = p_2 + 2p_3 + 4q_3$$

Da questa espressione si deduce che l'uscita s della cella deve coincidere con p_2 (uscita di somma di FA2), il bit di carry-out c_0 deve coincidere con p_3 (uscita di somma di FA3), e il bit di carry-out c_1 deve coincidere con q_3 (uscita di carry di FA3). Il circuito corrispondente è allora il seguente:



Naturalmente, il Full Adder FA2, essendo solo parzialmente utilizzato, può essere sostituito con un Half Adder.

(D4) Analizzare il comportamento della rete sequenziale in figura e ricavarne una rappresentazione come diagramma degli stati oppure come tavola di transizione; quindi per ciascuno stato stabilire se esso è transiente, ricorrente, persistente o isolato.



◆◆◆◆◆◆

Il circuito ha un singolo ingresso X e una singola uscita Y; esso è inoltre di tipo sincrono, poiché gli elementi di memoria presenti sono soltanto flip-flop sincroni, né sono presenti loop combinatori.

Detti J_1, K_1, Q_1 i terminali del flip-flop di sinistra e J_0, K_0, Q_0 i terminali del flip-flop di destra, le equazioni di eccitazione dei flip-flop sono le seguenti:

$$J_1 = Q_0 X \qquad K_1 = 0 \qquad J_0 = K_0 = \overline{Q}_1 X$$

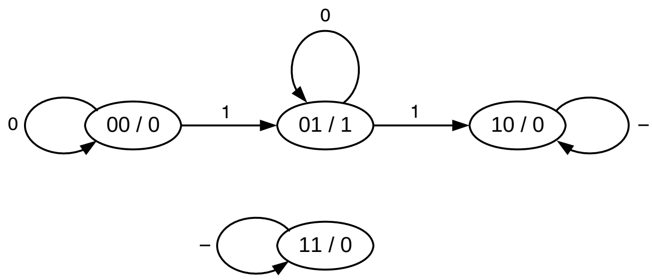
mentre l'equazione di uscita è

$$Y = \overline{Q}_1 Q_0$$

Dato che l'uscita dipende solo dallo stato e non dall'ingresso, la macchina sequenziale corrispondente è di tipo Moore. Assunto come stato la coppia $Q_1 Q_0$, la tavola di transizione si ricava facilmente dalle equazioni di cui sopra:

Q_1	Q_0	X	J_1	K_1	J_0	K_0	Q_1'	Q_0'	Y
0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	1	1	0	1	0
0	1	0	0	0	0	0	0	1	1
0	1	1	1	0	1	1	1	0	1
1	0	0	0	0	0	0	1	0	0
1	0	1	0	0	0	0	1	0	0
1	1	0	0	0	0	0	1	1	0
1	1	1	1	0	0	0	1	1	0

Il diagramma di stato sarà allora il seguente:



Da tale diagramma possiamo in conclusione osservare che:

- lo stato 00, non avendo rami entranti, è *transiente*;
- lo stato 01, avendo rami sia entranti che uscenti, è *ricorrente*;
- lo stato 10, non avendo rami uscenti, è *persistente*;
- lo stato 11, non avendo né rami entranti né rami uscenti, è *isolato*.