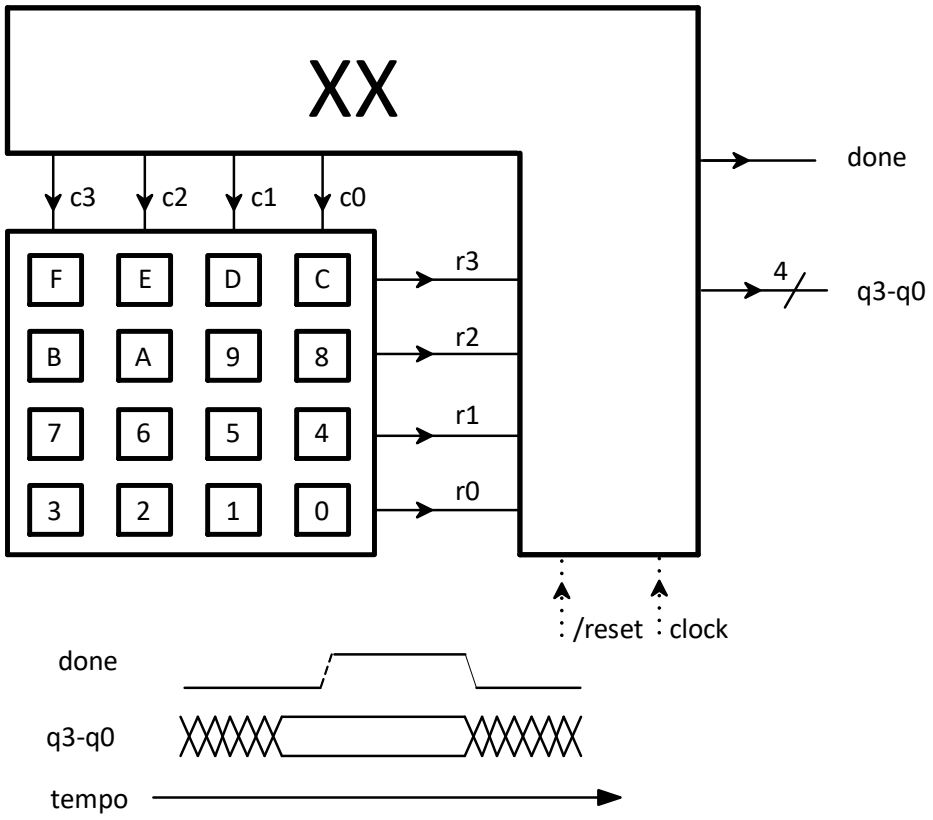
Reti logiche
Prova scritta del 30 Giugno 2020

Esercizio 1

Prendere un circuito moltiplicatore con addizionatore per naturali in base 2, ed aggiungendo ad esso meno logica possibile:

a) Sintetizzare un circuito che ha in ingresso due naturali X, Y ad n bit, ed in uscita la somma S dei due naturali su n bit, se rappresentabile, ed un carry che indica se la somma non è rappresentabile;
b) detti x ed y i due interi rappresentati in C2 da X ed Y, dimostrare che l'uscita S del circuito di cui sopra è la rappresentazione di $x + y$, se questo numero è rappresentabile su n bit; sintetizzare poi l'uscita di overflow.

Esercizio 2



La tastiera di figura viene usata da un operatore che preme un tasto alla volta. Quando viene premuto un tasto, XX deve fornirne in uscita la codifica binaria su 4 bit tramite le variabili *q3-q0*, mettendo l'uscita *done* a 1 finché il tasto non viene rilasciato. Dopo il rilascio del tasto *done* ritorna a zero.
XX attiva le colonne c3-c0 della matrice ciclicamente, mettendo una colonna ad 1 e tenendo le altre a zero. Quando una colonna è ad 1, se nessuno dei tasti sulla colonna è premuto, tutti gli ingressi di riga r3-r0 valgono 0. Se un tasto della colonna è stato premuto, l'ingresso della riga corrispondente vale 1.
Si assuma che non esista nessun problema di temporizzazione (i tasti vengono premuti a lungo, tra due pressioni di tasti intercorre un tempo molto lungo, la tastiera risponde velocemente).
Descrivere il circuito XX e sintetizzarlo come unità con parte operativa e parte controllo.

Soluzione esercizio 1

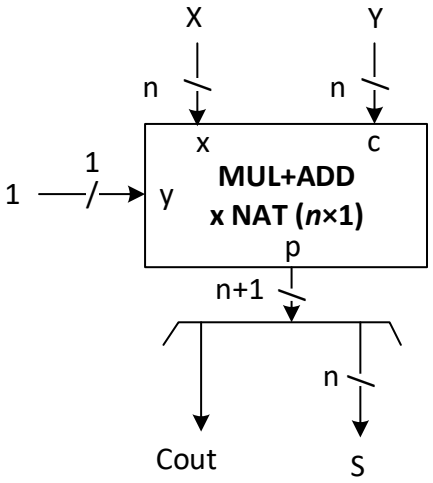
Per quanto riguarda i naturali, il circuito è quello in figura. Per quanto riguarda gli interi, definito $\boldsymbol{s = x + y}$, e detta $\boldsymbol{S}$ la sua rappresentazione in C2 su $\boldsymbol{n}$ bit (se esiste), si ha:

$$S \overset{(*)}{=} |s|_{2^n} = |x + y|_{2^n} = ||x|_{2^n} + |y|_{2^n}|_{2^n} = |X + Y|_{2^n}$$

Dove l'asterisco indica che S potrebbe non essere la rappresentazione di s, se questo non è rappresentabile su n bit. Questo dimostra che il circuito sopra stante può essere usato anche per sommare rappresentazioni di interi. Nella somma di interi si ha overflow quando gli addendi sono concordi ed il risultato è discorde rispetto agli addendi. Pertanto, il circuito di overflow ha come ingressi $x_{n-1}, y_{n-1}, s_{n-1}$ ed è così descritto:

$x_{n-1}, y_{n-1}, s_{n-1}$	ov
001	1
110	1
others	0

La sintesi è: $ov = \overline{x_{n-1}} \cdot \overline{y_{n-1}} \cdot s_{n-1} + x_{n-1} \cdot y_{n-1} \cdot \overline{s_{n-1}}$



Soluzione esercizio 2

```
module XX(c3_c0, done,q3_q0, r3_r0,clock,reset_);
 input          clock,reset_;
 input  [3:0]   r3_r0;
 output [3:0]   c3_c0;
 output [3:0]   q3_q0;
 output         done;

 reg            DONE;
 reg [3:0]      C3_C0, Q3_Q0;
 reg [1:0]      STAR;
 parameter [2:0] S0=0, S1=1, S2=2;

 assign   done=DONE;
 assign   q3_q0=Q3_Q0;
 assign   c3_c0=C3_C0;

 always @(reset_==0) #1 begin C3_C0<='B1000; DONE<=0; STAR<=S0; end
 always @(posedge clock) if (reset_==1) #3
  casex(STAR)
   S0: begin DONE<=0; C3_C0<={C3_C0[0],C3_C0[3:1]}; STAR<=S1; end
   S1: begin Q3_Q0<=decodifica(C3_C0,r3_r0); STAR<=(r3_r0!='B0000)?S2:S0; end
   S2: begin DONE<=1; STAR<=(r3_r0!='B0000)?S2:S0; end
  endcase

function [3:0] decodifica;
 input [3:0] C3_C0,r3_r0;
 casex({C3_C0,r3_r0})
  'B1000_1000: decodifica='B1111;
  'B0100_1000: decodifica='B1110;
  'B0010_1000: decodifica='B1101;
  'B0001_1000: decodifica='B1100;
  'B1000_0100: decodifica='B1011;
  'B0100_0100: decodifica='B1010;
  'B0010_0100: decodifica='B1001;
  'B0001_0100: decodifica='B1000;
  'B1000_0010: decodifica='B0111;
  'B0100_0010: decodifica='B0110;
  'B0010_0010: decodifica='B0101;
  'B0001_0010: decodifica='B0100;
  'B1000_0001: decodifica='B0011;
  'B0100_0001: decodifica='B0010;
  'B0010_0001: decodifica='B0001;
  'B0001_0001: decodifica='B0000;
   default   : decodifica='BXXXX;
 endcase
endfunction
endmodule
```