

SCHEDA ASE1807		Data: 10 Settembre 2018
Cognome	Nome	

ESERCIZIO N°1

8 punti

Realizzare un sottoprogramma per il microcontrollore AVR XMEGA256A3BU, in grado di valutare la somma (MS) di due numeri relativi anch'essi rappresentati in modulo e segno, contenuti nei registri R16 e R17. Il risultato deve essere posto in R16. Il flag V al termine del sottoprogramma deve essere coerente con l'operazione svolta (1 se il risultato non è rappresentabile MS; in questo caso il valore di R16 non è significativo).

ESERCIZIO N°2

4 punti

Determinare la rappresentazione delle 5 cifre BCD meno significative dei seguenti numeri interi assoluti, espressi in varie basi:

$|254771|_8$

$|BABB0ACACC1A|_{16}$

$|2200110012|_3$

$|10001010101000101110|_2$

ESERCIZIO N°3

5 punti

Realizzare in forma SP ottima una rete combinatoria in grado di valutare il segno della somma (secondo la convenzione MS e usando 0 per il valore del segno in caso di risultato nullo) di due numeri relativi rappresentati su 3 bit MS (x_2, x_1, x_0 e y_2, y_1, y_0 sono le cifre dei 2 numeri binari X e Y).

ESERCIZIO N°4

5 punti

Progettare una macchina di Moore con un ingresso e una uscita, in grado di riconoscere una qualsiasi delle 3 sequenze (comunque interallacciate) 111, 00, 0101. Quando la macchina riconosce una qualsiasi delle sequenze, pone l'uscita a 1 per 1 ciclo di clock.

ESERCIZIO N°5

6 punti

Disegnare lo schema logico di un comparatore digitale (in grado di fornire l'indicazione $A < B$, $A > B$ oppure $A = B$) tra numeri relativi a 4 bit rappresentati in modulo e segno.

ESERCIZIO N°6

5 punti

Realizzare la seguente funzione booleana

$$u = x + \bar{y}z + \bar{z}(\bar{x}\bar{t} + ty)$$

a) usando dei multiplexer 2:1

b) usando decoder 4:16 e il minimo numero possibile di porte NOR a 3 ingressi

1

Realizzare un sottoprogramma per il microcontrollore AVR XMEGA256A3BU, in grado di valutare la somma (MS) di due numeri relativi anch'essi rappresentati in modulo e segno, contenuti nei registri R16 e R17. Il risultato deve essere posto in R16. Il flag V al termine del sottoprogramma deve essere coerente con l'operazione svolta (1 se il risultato non è rappresentabile MS; in questo caso il valore di R16 non è significativo).

```
addMS:
  push R17
  sbrs R16,7 //controlla il segno di R16
  rjmp R16pos
  andi R16,0b01111111 //trova il modulo di R16
  neg R16 //converte il negativo il C2; vale anche per 0
R16pos:
  sbrs R17,7 //controlla il segno di R17
  rjmp R17pos
  andi R17,0b01111111 //trova il modulo di R17
  neg R17 //converte il negativo il C2; vale anche per 0
R17pos:
  add R16,R17 //somma in C2
  brvs end //c'e` overflow e il risultato non e` significativo
  brpl end //risultato positivo, senza overflow
  neg R16 //converte il risultato negativo nel modulo in MS
  brvs end //si ha con -128, unico valore negativo C2 non rappresentabile MS
  ori R16,0b10000000 //mette il bit del segno
end:
  pop R17
  ret
```

②

Occorre trovare la rappresentazione dei numeri dati in BASE 10, limitandosi a valutare le espressioni MODULO 10^5

$$|254771|_8 = (((((2 \cdot 8 + 5) \cdot 8 + 4) \cdot 8 + 7) \cdot 8 + 7) \cdot 8 + 1) = 88569$$

in BCD (sono 5 cifre)

$$1000.1000.0101.0110.1001$$

$$|2200110012|_3 = (((((2 \cdot 3 + 2) \cdot 3 + 1) \cdot 3 + 1) \cdot 3 + 1) \cdot 3 + 2) = 52817$$

in BCD (sono 5 cifre)

$$0101.0010.1000.0001.0111$$

$$|10001010101000101110|_2 = |8AA2E|_{16} = 5 \downarrow 67854$$

non interessa.

in BCD (solo le 5 cifre meno significative)

$$0110.0111.1000.0101.0100$$

Per l'ultimo numero (molto grosso) valutiamo le potenze di 16 modulo 10^5 , poi eseguiamo il prodotto per la cifra e sommiamo

EXP	$ 16^e _{10^5}$	cifra hex	$ Prod _{10^5}$	
0	1	10	10	Somma _{10⁵} 13370
1	16	1	16	
2	256	12	3072	
3	4096	12	49152	
4	65536	10	55360	
5	48576	12	82912	
6	77216	10	72160	
7	35456	0	0	
8	67296	11	40256	
9	76736	11	44096	
10	27776	10	77760	
11	44416	11	88576	

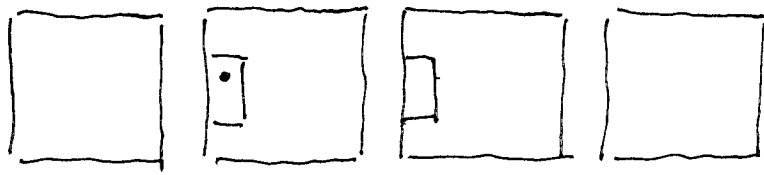
in BCD

$$0001.0011.0011.0111.0000$$

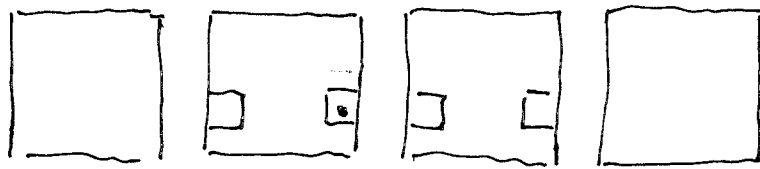
③ Sintesi ottima di rete a 6 ingressi. MAPPE a 6 (segno pos: ϕ)

$y_1 y_0$ \ $x_1 x_0$	00	01	11	10	00	01	11	10	00	01	11	10	00	01	11	10
00	0	0	0	0	0	0	0	0	0	1	1	1	0	1	1	1
01	0	0	0	0	1	0	0	0	1	1	1	1	0	0	1	1
11	0	0	0	0	1	1	0	1	1	1	1	1	0	0	0	0
10	0	0	0	0	1	1	0	0	1	1	1	1	0	0	1	0

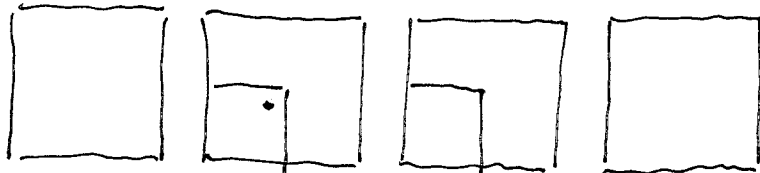
$x_2 y_2 = 00$ 01 11 10



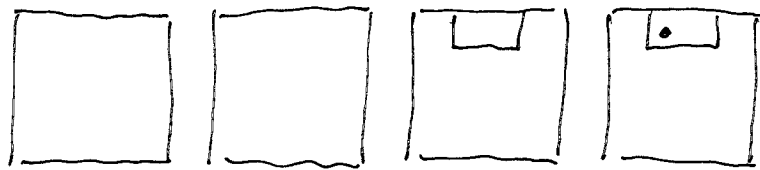
ess. $y_2 y_0 \bar{x}_1 \bar{x}_0$



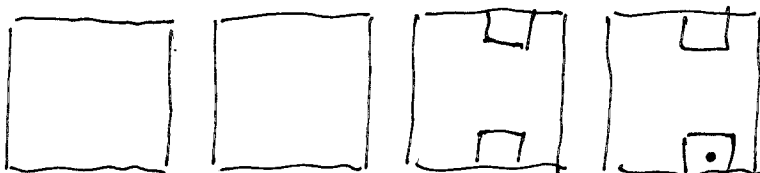
ess. $y_2 y_1 y_0 \bar{x}_0$



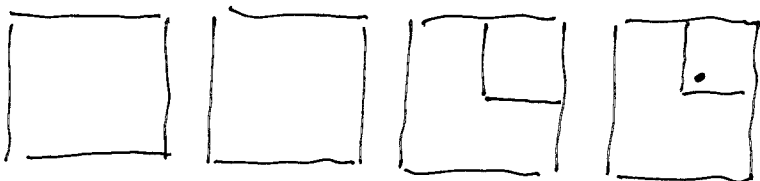
ess. $y_2 y_1 \bar{x}_1$



ess. $x_2 x_0 \bar{y}_1 \bar{y}_0$

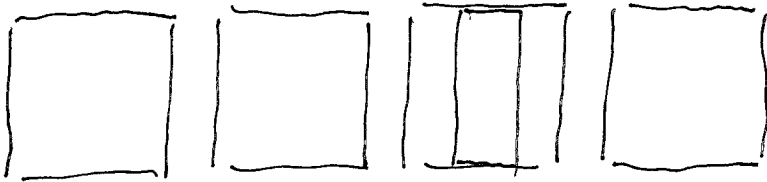


ess. $x_2 x_1 x_0 \bar{y}_0$

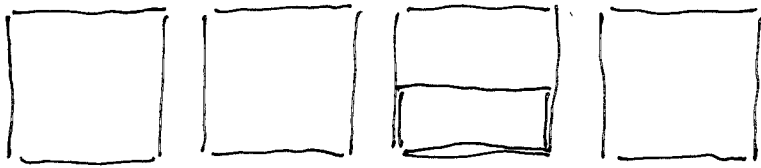


ess. $x_2 x_1 \bar{y}_1$

Dopo aver trovato gli essenziali, ci restano da coprire 3 mintermi. col minimo numero di principali (di ordine 3)



$$x_2 x_0 y_2$$

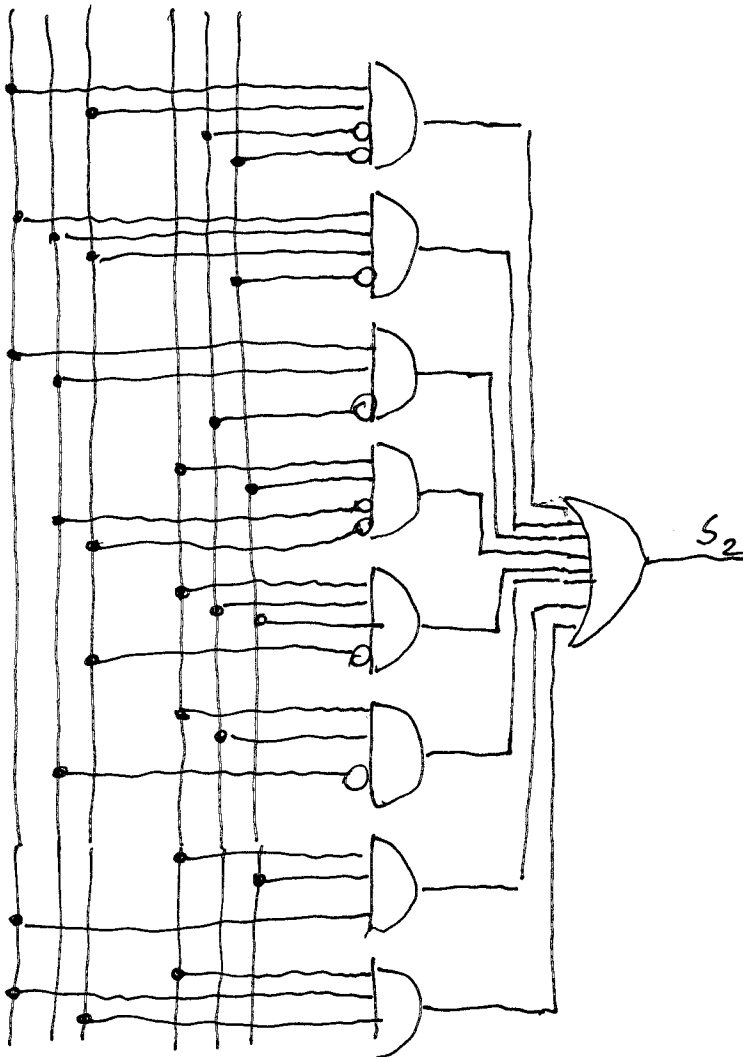


$$x_2 y_2 y_0$$

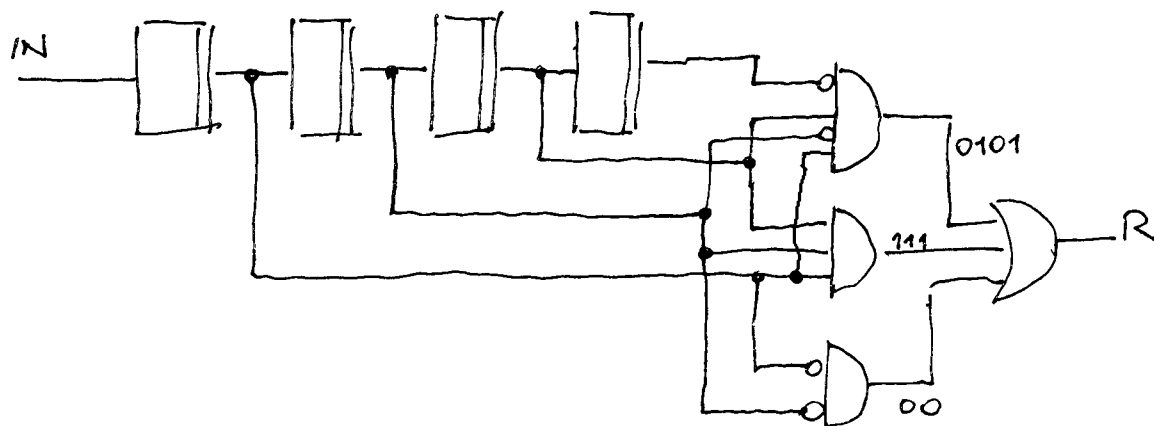
Quindi si ha

$$S_2 = y_2 y_0 \bar{x}_1 \bar{x}_0 + y_2 y_1 y_0 \bar{x}_0 + y_2 y_1 \bar{x}_1 + \\ + x_2 x_0 \bar{y}_1 \bar{y}_0 + x_2 x_1 x_0 \bar{y}_0 + x_2 x_1 \bar{y}_1 + \\ + x_2 x_0 y_2 + x_2 y_2 y_0$$

$y_2 y_1 y_0$ $x_2 x_1 x_0$



④ Possiamo realizzare la macchina con uno shift



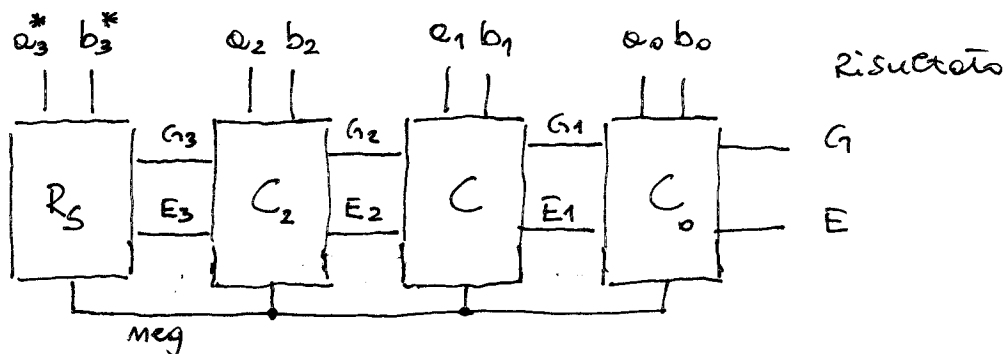
L'uscita è significativa dopo 4 cicli di clock, qualunque sia il valore iniziale dei flip-flop.

5) Uso 2 bit per codificare il confronto

	G	E
$A < B$	0	0
$A > B$	1	0
$A = B$	0	1

(Volevamo, si può avere $L = \bar{G}\bar{E}$)

Realizzo una struttura MSB first



R_3 è la rete che confronta i segni;
per evitare il caso 1000 $a_3^* = a_3 \cdot (a_2 + a_1 + a_0)$; $b_3^* = b_3 (b_2 + b_1 + b_0)$

a_3	b_3	G_3	E_3	neg
0	0	0	1	0
0	1	1	0	0
1	0	0	0	0
1	1	0	1	1

$$G_3 = \bar{a}_3 b_3$$

$$E_3 = \bar{a}_3 \oplus b_3$$

$$neg = a_3 b_3$$

C_2 è un blocco comparatore (che inverte logica, per dati negativi)

G_{i+1}	E_{i+1}	a_i	b_i	neg	G_i	E_i
1	0	-	-	-	1	0
0	0	-	-	-	0	0
0	1	0	0	-	0	1
		1	1	-	0	1
0	1	0	1	0	1	0
		1	0	0	0	0
0	1	0	1	1	0	0
		1	0	1	1	0

$$G_i = G_{i+1} \bar{E}_{i+1} +$$

$$+ \bar{G}_{i+1} E_{i+1} \bar{a}_i b_i \bar{neg} +$$

$$+ \bar{G}_{i+1} E_{i+1} a_i \bar{b}_i neg$$

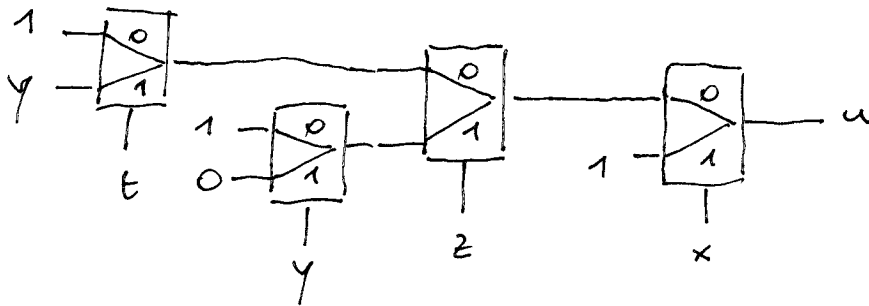
$$E_i = \bar{G}_{i+1} E_{i+1} (\bar{a}_i \oplus b_i)$$

* In alternativa si poteva pensare in TRASLAZIONE e usare un comparatore normale

6) Trovo la mappa

xy		zt			
		00	01	11	10
00	0	1	1	1	1
	1	0	1	1	1
01	0	1	0	1	1
	1	1	0	1	1
11	0	1	1	1	1
	1	1	1	1	1
10	0	1	1	1	1
	1	1	1	1	1

siūtes con MUX 2:1



siūtes con decoder (ricomposco i 3 eeri)

