

SCHEDA ASE2003		Data: 17 Febbraio 2020
Cognome	Nome	

ESERCIZIO N°1

8 punti

Realizzare una subroutine `sum_sub` per il microcontrollore AVR XMEGA256A3BU che, a partire dall'indirizzo contenuto in `Y` e per 256 locazioni totali, somma tutti i valori di indirizzo pari e sottrae tutti i valori di indirizzo dispari. I valori letti sono interpretati come interi senza segno e il risultato, in complemento a 2, deve essere lasciato nella coppia di registri `R25:R24`. Può aversi overflow?

ESERCIZIO N°2

5 punti

- a), Usando blocchi logici noti (porte logiche elementari, full-adder, mux e demux...) realizzare una rete in grado di sottrarre in virgola fissa a 8 bit un numero in `C1` da uno in modulo e segno, fornendo il risultato in traslazione.
- b), Realizzare la rete che valuta l'overflow nella sottrazione precedente.

ESERCIZIO N°3

5 punti

Realizzare una macchina sequenziale sincrona secondo il modello di Mealy sincronizzata con 3 ingressi X_2, X_1, X_0 che sono le cifre binarie di un numero con segno e una uscita U , in grado di individuare ponendo a 1 l'uscita tutti i casi in cui 3 ingressi consecutivi rappresentano numeri `C2` in sequenza strettamente crescente. La macchina deve essere dotata di reset asincrono, che la riporta in uno stato di partenza. Non è richiesta ottimizzazione.

ESERCIZIO N°4

5 punti

Sintetizzare in forma normale ottima (scegliendo tra forma SP e PS) e disegnare lo schema logico della una rete combinatoria a 5 ingressi (X_4, X_3, X_2, X_1, X_0) non completamente specificata, individuata dai seguenti valori di uscita, corrispondenti alla sequenza ordinata degli ingressi: -, -, 0, 1, 1, 1, 0, -, 0, 0, 1, 0, 1, 0, -, -, 0, 1, 1, -, -, 1, 0, 0, 1, 0, 1, 0, 0, -, 1, 0.

ESERCIZIO N°5

5 punti

Mostrare un ciclo di scrittura di una SRAM, evidenziando i tempi di rispetto per il corretto funzionamento.

ESERCIZIO N°6

5 punti

Determinare la rappresentazione in virgola fissa `C2` su 10 bit che rappresenta con il minimo errore assoluto (da valutare) $e_a = (\hat{x} - x)$ tutti i seguenti valori:

$$A = -10/13$$

$$B = 1,03769$$

$$C = -8/7$$

$$D = -1,9943$$

1

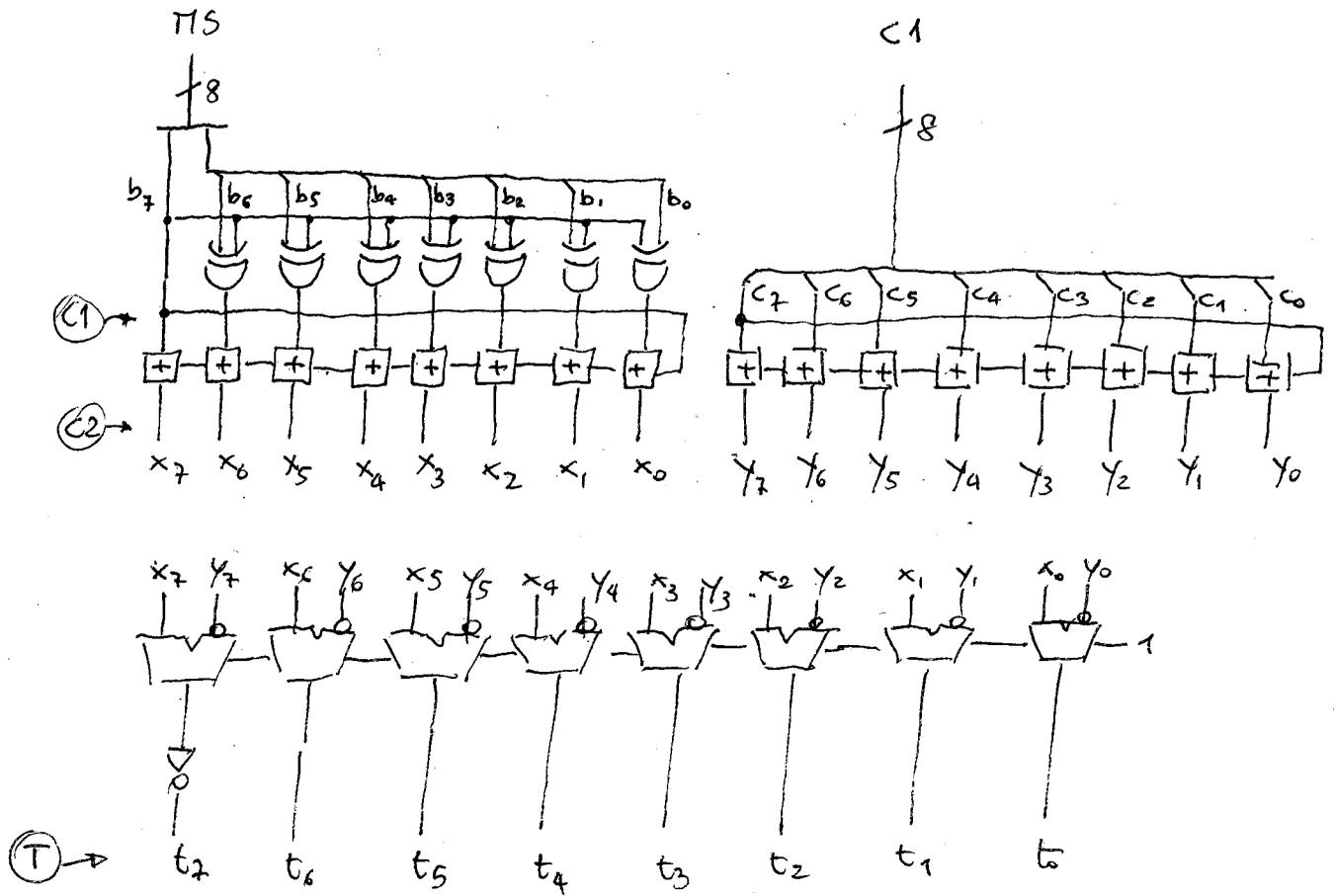
Realizzare una subroutine `sum_sub` per il microcontrollore AVR XMEGA256A3BU che, a partire dall'indirizzo contenuto in `Y` e per 256 locazioni totali, somma tutti i valori di indirizzo pari e sottrae tutti i valori di indirizzo dispari. I valori letti sono interpretati come interi senza segno e il risultato in complemento a 2, deve essere lasciato nella coppia di registri `R25:R24`.

Può aversi overflow?

```
/* No, in quanto il range del risultato dell'operazione richiesta è
   -128*255..128*255, incluso nel range di rappresentabilità C2 a 16 bit,
   -2^15..2^15-1
*/
```

```
sum_sub:
  push R16
  push R17
  push R1
  clr R1
  clr R16 //per eseguire 256 iterazioni
  clr R24
  clr R25
loop:
  ld R17,Y+
  sbrc YL,0 //se YL ora è pari deve sottrarre
  rjmp sum
  sub R24,R17
  sbc R25,R1
  rjmp endc
sum:
  add R24,R17
  adc R25,R1
endc:
  dec R16
  brne loop
  dec YH //ripristina il puntatore
  pop R1
  pop R17
  pop R16
ret
```

② Come convertire entrambi gli operandi in C2, sottrarli con un comune ripple-borrow subtractor e quindi convertire in T.

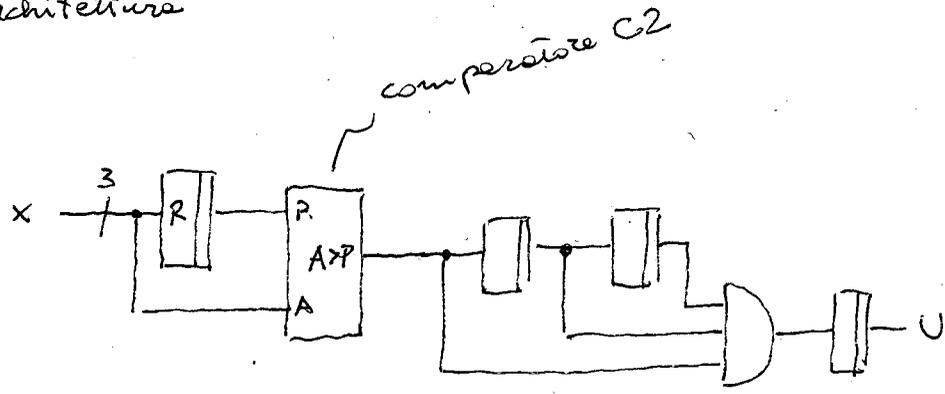


Rete per l'overflow (usa x_7 y_7 e t_7)
 (sottra t_7 e risultato - oppure $-$, $+$ e risultato $+$)

$$OV = x_7 \bar{y}_7 t_7 + \bar{x}_7 y_7 \bar{t}_7$$

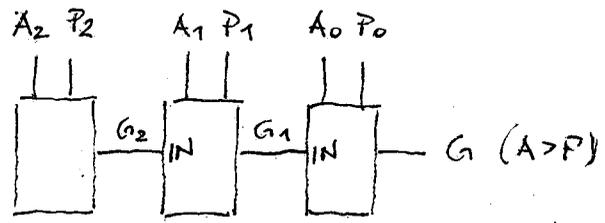
③ Se con un comparatore confronto un dato al precedente, per vedere se è maggiore, è sufficiente riconoscere 111

Architettura



Per evitare falsi riconoscimenti conviene usare registri con reset asincrono e implementare R in modo che sia inizializzato a 011 (il MAX)

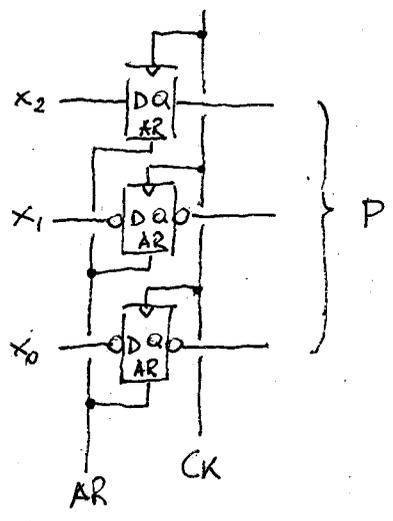
Comparatore C2



$$G_2 = \bar{A}_2 P_2 \quad (A \text{ positivo } P \text{ negativo})$$

$$G_1 = IN + A_1 \bar{P}_1$$

$$G = IN + A_0 \bar{P}_0$$



Registro R

4

Mappe delle funzioni

$x_3 x_2$		$x_4 = 0$			
		00	01	11	10
$x_1 x_0$	00	0	4	12	8
	01	-	1	5	13
	11	1	3	-	7
	10	0	2	6	-

$x_3 x_2$		$x_4 = 1$			
		00	01	11	10
$x_1 x_0$	00	16	-	20	28
	01	17	1	21	29
	11	-	19	0	23
	10	1	18	0	22

sintesi SP: ricerca essenziali e coperture degli 1 rimanenti

-	1	1	0
-	1	0	0
1	-	-	0
0	0	-	1*

0	-	0	1*
1	1	-	0
1	0	0	0
1	0	1	1

Letterali

$$4 + 3 + 3 + 3 + 4 + 4 = 21$$

sintesi PS: ricerca essenziali e coperture degli 0 rimanenti

-	1	1	0*
-	1	0*	0
1	-	-	0
0	0	-	1

0	-	0	1
1	1	-	0
-	0	0	0
1	0	1	1

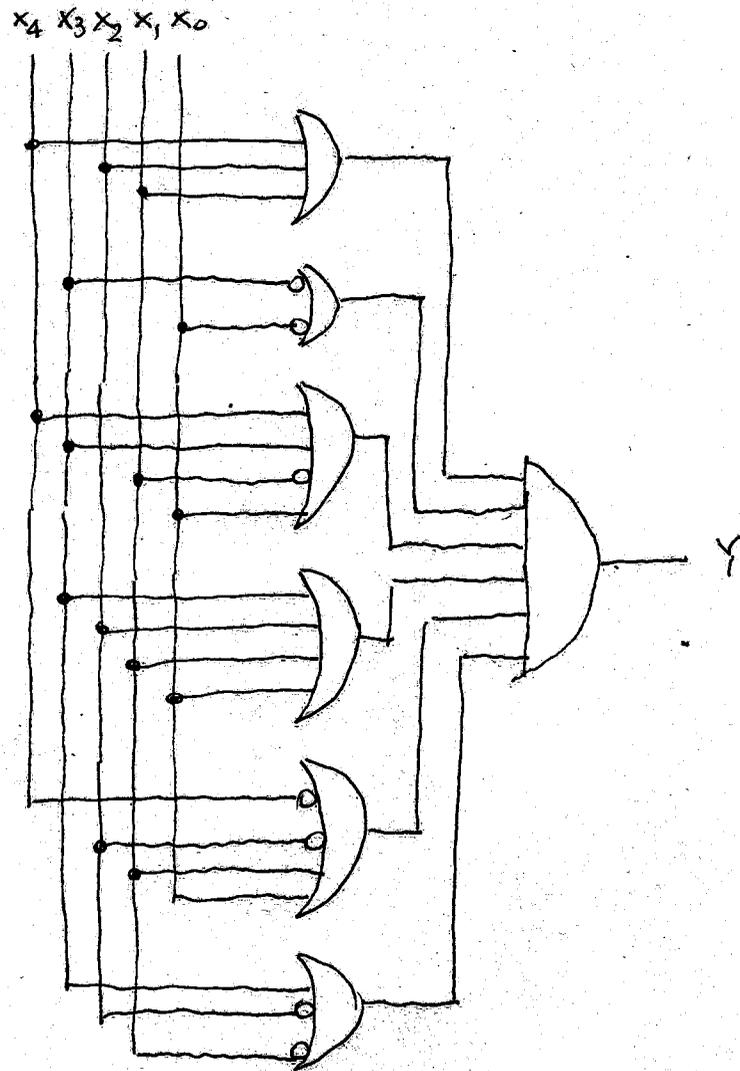
Letterali

$$3 + 2 + 4 + 4 + 4 + 3 = 20$$

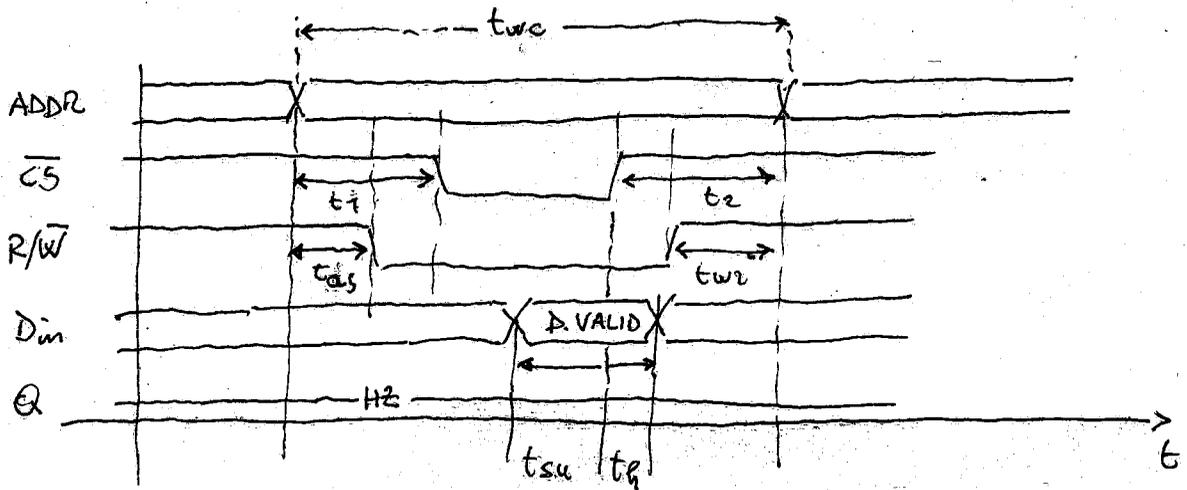
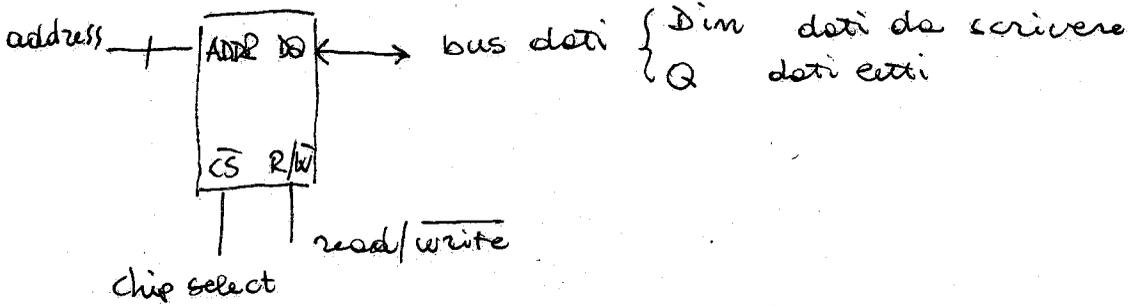
Coverage e sintesi PS.

$$Y = (x_4 + x_2 + x_1)(\bar{x}_3 + \bar{x}_0)(x_4 + x_3 + \bar{x}_1 + x_0)(x_3 + x_2 + x_1 + x_0) \cdot (\bar{x}_4 + \bar{x}_2 + x_1 + x_0)(x_3 + \bar{x}_2 + \bar{x}_1)$$

Schema logico



5 Ciclo di scrittura per interfaccia semplificata



Occorre garantire ADDR stabile durante TUTTO il ciclo di scrittura

I dati devono essere VALIDI e cessare della FINE della scrittura (fronte \downarrow di \overline{CS})

⑥

$$A \approx -0,7692307\dots$$

$$B = 1,03769$$

$$C \approx -1,142857\dots$$

$$D = -1,9943$$

Per la parte intera occorrono 2 bit (compreso il segno)
Restano 8 bit per la parte frazionaria

$$-2 \leq x < 2 - 1/256$$

Procedura: moltiplico per $2^8 = 256$ e arrotondo
se negativo sommo 1024
convertito in binario

A :	11.00111011	errore	$-3,00 \cdot 10^{-4}$
B :	01.00001010		$1,37 \cdot 10^{-3}$
C :	10.11011011		$-1,67 \cdot 10^{-3}$
D :	10.00000001		$-1,79 \cdot 10^{-3}$

Come ci si aspettava, $|e| \leq \text{LSB}/2 = 1,95 \cdot 10^{-3}$