| SCHEDA **ASE2307** | Data: **17 Luglio 2023** |
|---|---|

Il testo deve essere riconsegnato nella cartellina. Non è ammessa la consultazione degli appunti e dei compiti precedenti. Si possono consultare i data sheet. Non usare il colore rosso nello svolgimento.

# ESERCIZIO N°1

5 punti

Determinare la rappresentazione nel formato binary 32 (IEEE754) dei seguenti numeri, arrotondando il valore effettivo al codice più vicino, e poi valutare l'errore assoluto con segno definito da $(\hat{x}-x)$ usando, se l'errore non è nullo, 3 (strettamente) cifre significative

$-260!/255!$
$-2^{-160}$
$4 \log_{13}(e^{5000})$

# ESERCIZIO N°2

8 punti

Realizzare un sottoprogramma per il microcontrollore AVR XMEGA256A3BU, che divide per 2 (con eventuale arrotondamento al valore intero superiore) tutti gli elementi (interi su 2 byte, in C2, LSB first) di un vettore di dimensione *n* (contenuto in R0), posto in memoria all'indirizzo contenuto in Z. Nel caso in cui la dimensione *n* sia nulla, la subroutine esce senza effettuare modifiche.

# ESERCIZIO N°3

5 punti

Disegnare lo schema logico di una rete che esegue la funzione delle 5 variabili $X_4$, $X_3$, $X_2$, $X_1$ e $X_0$, facendo riferimento alla forma normale (SP o PS) che presenta il minimo numero di letterali. La funzione, non completamente definita, è caratterizzata dalla seguente tabella di verità:
1, 0, -, 0, -, 1, -, 1, 0, 1, 1, 1, -, 0, -, 1, 0, -, 1, -, -, 0, 1, 0, -, 1, 0, 0, 0, -, 1, 0.

# ESERCIZIO N°4

5 punti

Progettare un contatore sincrono $\overline{U}/D$, modulo 12, con abilitazione, con architettura ripple carry.

# ESERCIZIO N°5

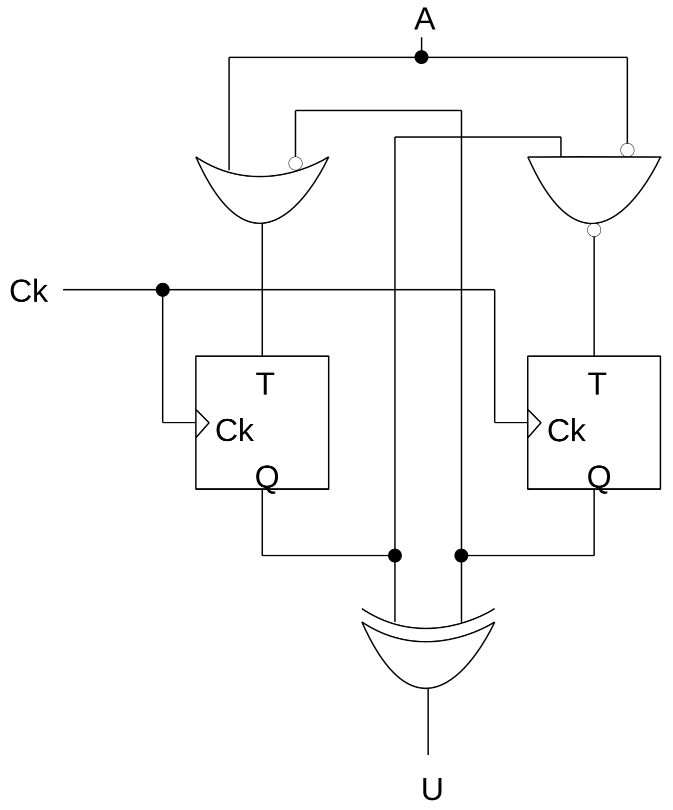
5 punti

Realizzare, usando esclusivamente multiplexer 2:1, la rete combinatoria proposta nell'esercizio 3. Evitare l'uso di multiplexer ridondanti.

## ESERCIZIO N°6

5 punti

Determinare tipologia architetturale e grafo di flusso della seguente macchina sequenziale sincrona.



1

Determinare la rappresentazione nel formato binary 32 (IEEE754) dei seguenti numeri, arrotondando il valore effettivo al codice più vicino, e poi valutare l'errore assoluto con segno definito da usando, se l'errore non è nullo, 3 (strettamente) cifre significative.

Conviene scrivere il primo valore isolando i suoi fattori.
Il valore è rappresentabile come normalizzabile.

$$-\frac{260!}{255!} = -260 \cdot 259 \cdot 258 \cdot 257 \cdot 256 = -2^{40} \cdot 1,03959954344...$$

$S = 0\text{b}1 \qquad E = 40 + 127 = 167 = 0\text{b}10101001$
$T = 332185 = 0\text{b}000\ 0101\ 0001\ 0001\ 1001\ 1001$

$$\epsilon = \hat{x} - x \simeq 6,14 \cdot 10^3$$

Il secondo valore è una potenza di 2.
Visto il suo ordine di grandezza, si può rappresentare in modo esatto come NON normalizzabile.

$$-2^{-160}$$

$S = 0\text{b}1 \qquad E = 0\text{b}0$
$T = 2^{(-160+169)} = 2^9 = 0\text{b}000\ 0000\ 0000\ 0010\ 0000\ 0000$

$$\epsilon = \hat{x} - x = 0$$

Il terzo valore si esprime facilmente usando le proprietà dei logaritmi.
Si può rappresentare come normalizzabile.

$$4\log_{13}(e^{5000}) = \frac{20000}{\ln(13)} = 7797,424905... = 2^{12} \cdot 1,9036681897...$$

$S = 0\text{b}0 \qquad E = 12 + 127 = 0\text{b}10001011$
$T = 7580518 = 0\text{b}111\ 0011\ 1010\ 1011\ 0110\ 0110$

$$\epsilon = \hat{x} - x = -1,00 \cdot 10^{-4}$$

2

```
/* Realizzare un sottoprogramma per il microcontrollore AVR XMEGA256A3BU,
che divide per 2 (con eventuale arrotondamento al valore intero superiore)
tutti gli elementi (interi su 2 byte, in C2, LSB first)
di un vettore di dimensione n (contenuto in R0),
posto in memoria all’indirizzo contenuto in Z.
Nel caso in cui la dimensione n sia nulla,
la subroutine esce senza effettuare modifiche. */

div2_c2:
  tst R0
  breq end_div2
    push R0
    push R24
    push R25
    push ZL
    push ZH
    loop:
      ld R24,Z  //carica il valore da dividere
      ldd R25,Z+1
      asr R25   //divide con segno
      ror R24
      brcc poi
        adiw R25:R24,1  //arrotonda al successivo se valore iniziale dispari
      poi:
      st Z+,R24 //memorizza e incrementa
      st Z+,R25
    dec R0
    brne loop
    pop ZH
    pop ZL
    pop R25
    pop R24
    pop R0
  end_div2:
  ret
```

3

Disegnare lo schema logico di una rete che esegue la funzione delle 5 variabili X4, X3, X2, X1 e X0, facendo riferimento alla forma normale (SP o PS) che presenta il minimo numero di letterali.
La funzione, non completamente definita,
è caratterizzata dalla seguente tabella di verità:
1, 0, -, 0, -, 1, -, 1, 0, 1, 1, 1, -, 0, -, 1, 0, -, 1, -, -, 0, 1, 0, -, 1, 0, 0, 0, -, 1, 0.

$X_4 = 0$

| $X_1, X_0$ \ $X_3, X_2$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 1 * | - | - | 0 |
| 01 | 0 | 1 * | 0 | 1 |
| 11 | 0 | 1 | 1 | 1 |
| 10 | - | - | - | 1 |

$X_4 = 1$

| $X_1, X_0$ \ $X_3, X_2$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 0 | - | 0 | - |
| 01 | - | 0 | - | 1 |
| 11 | - | 0 | 0 | 0 |
| 10 | 1 | 1 | * 1 | 0 |

SP; rossi essenziali
19 letterali
(5 ordine 2, 1 ordine 1)

$X_4 = 0$

| $X_1, X_0$ \ $X_3, X_2$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 1 | - | - | *0 |
| 01 | *0 | 1 | *0 | 1 |
| 11 | 0 | 1 | 1 | 1 |
| 10 | - | - | - | 1 |

$X_4 = 1$

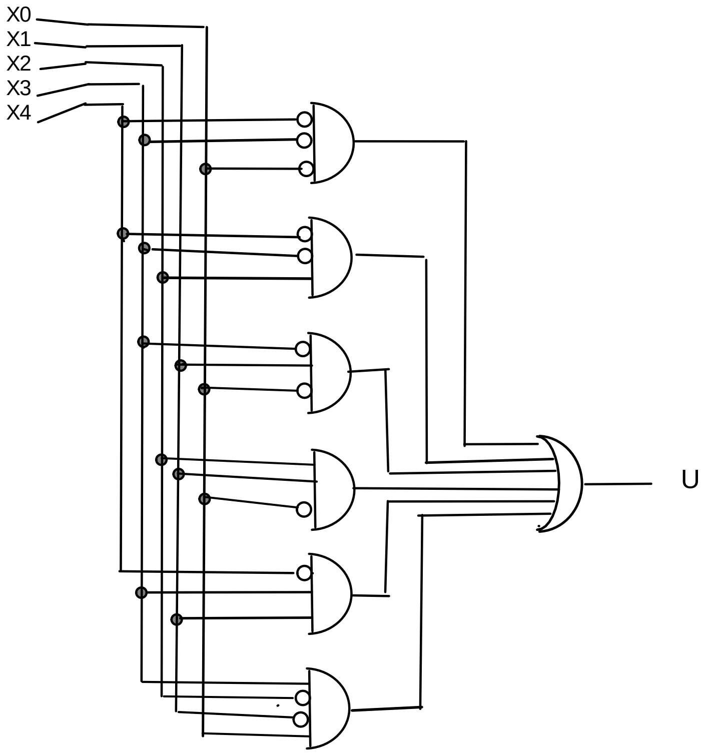
| $X_1, X_0$ \ $X_3, X_2$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 0 | - | 0 | - |
| 01 | - | 0 | - | 1 |
| 11 | - | 0 | 0 | 0 |
| 10 | 1 | 1 | 1 | 0 |

PS; rossi essenziali
19 letterali
5 ordine 2, 1 ordine 1)

Le soluzioni sono equivalenti. Scelgo SP.

$$U = \overline{X_4}\,\overline{X_3}\,\overline{X_0} + \overline{X_4}\,\overline{X_3}X_2 + \overline{X_3}X_1\overline{X_0} + X_2X_1\overline{X_0} + \overline{X_4}X_3X_1 + X_3\overline{X_2}\,\overline{X_1}X_0$$

Schema logico

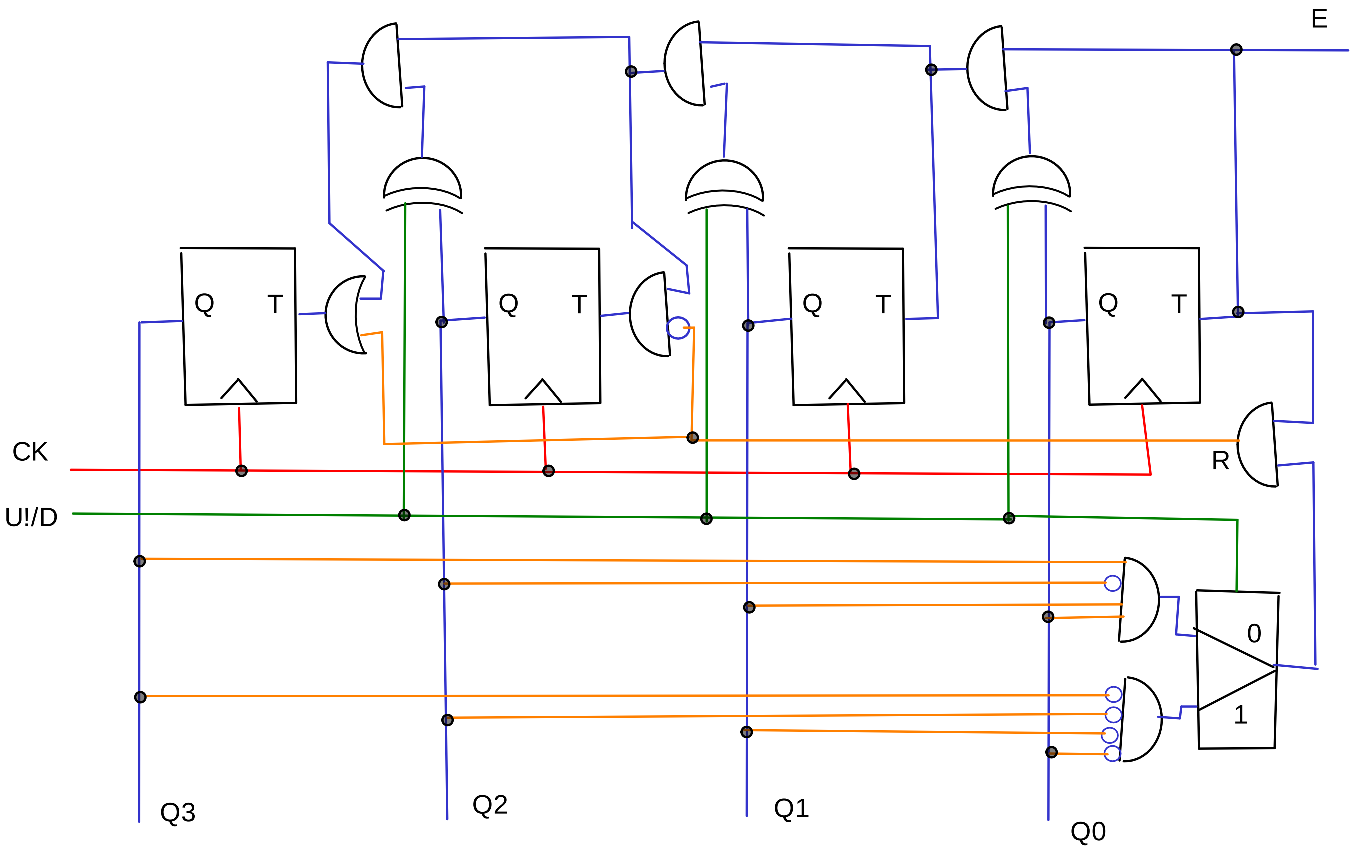


4

Progettare un contatore sincrono U!/D, modulo 12, con abilitazione, con architettura ripple carry.

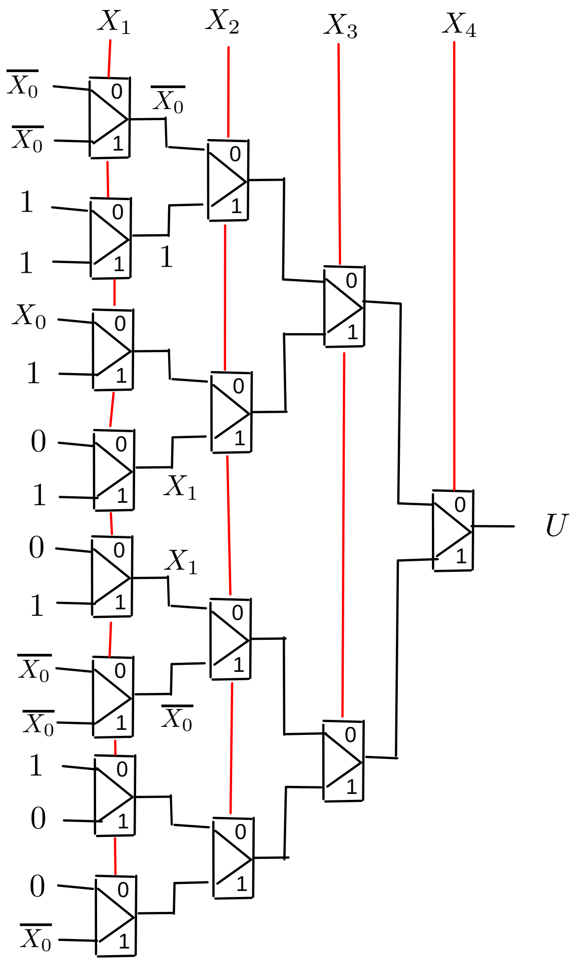
```
Esamino le sequenze di conteggio:
up       down
 ....     ....
 1001     0010
 1010     0001
 1011     0000
(FB--)   (FB--)
 0000     1011
 ....     ....

Lo stesso operatore FB-- va bene
per entrambe le direzioni.
```

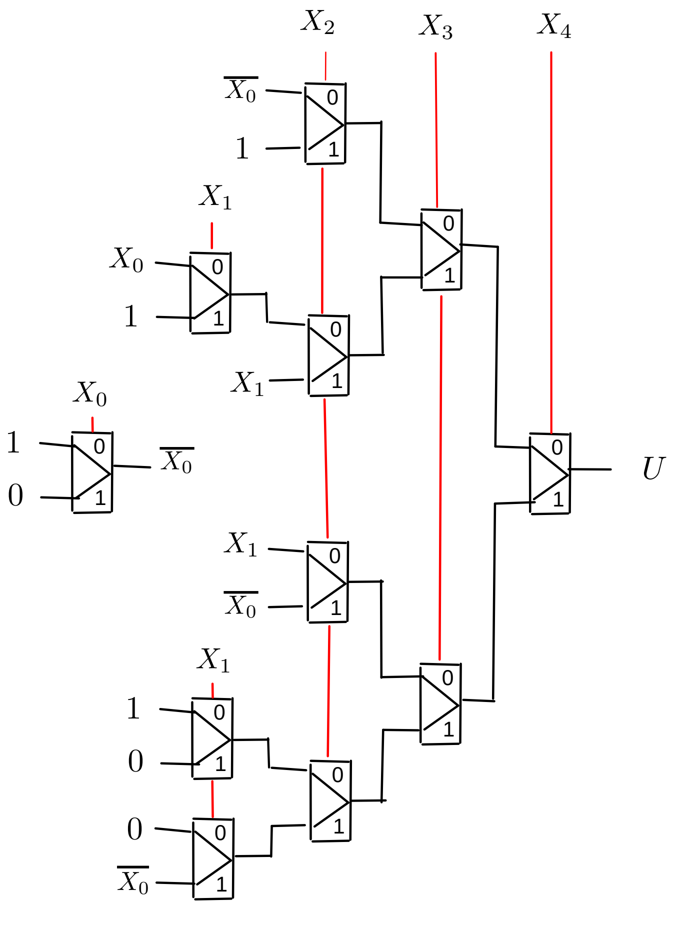


5

Realizzare, usando esclusivamente multiplexer 2:1, la rete combinatoria proposta nell'esercizio 3. Evitare l'uso di multiplexer ridondanti.
1, 0, -, 0, -, 1, -, 1, 0, 1, 1, 1, -, 0, -, 1, 0, -, 1, -, -, 0, 1, 0, -, 1, 0, 0, 0, -, 1, 0.



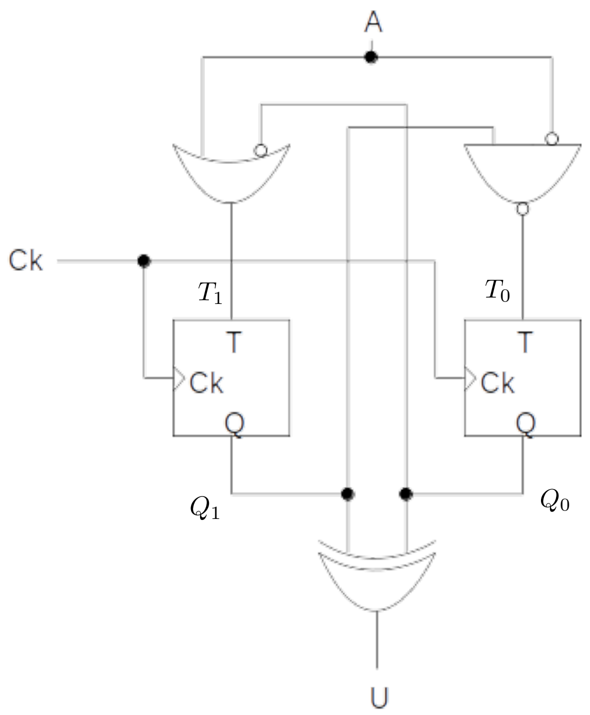

Sfruttando i don't care, posso arrivare a configurazioni semplificate




Eliminando i mux ridondanti ottengo una ulteriore semplificazione.

6

Determinare tipologia architetturale e grafo di flusso della seguente macchina sequenziale sincrona.




Questa è una macchina di Moore,
in quanto l'uscita
dipende soltanto dallo stato (l'uscita dei T-FF)

$$U = Q_1 \oplus Q_0$$

$$T_1 = A + \overline{Q_0}$$

$$T_0 = A + \overline{Q_1}$$

| St | Q1 | Q0 | A | T1 | T0 | Q1+ | Q0+ | St+ |
|---|---|---|---|---|---|---|---|---|
| A | 0 | 0 | 0 | 1 | 1 | 1 | 1 | D |
| | | | 1 | 1 | 1 | 1 | 1 | D |
| B | 0 | 1 | 0 | 0 | 1 | 0 | 0 | A |
| | | | 1 | 1 | 1 | 1 | 0 | C |
| C | 1 | 0 | 0 | 1 | 0 | 0 | 0 | A |
| | | | 1 | 1 | 1 | 0 | 1 | B |
| D | 1 | 1 | 0 | 0 | 0 | 1 | 1 | D |
| | | | 1 | 1 | 1 | 0 | 0 | A |