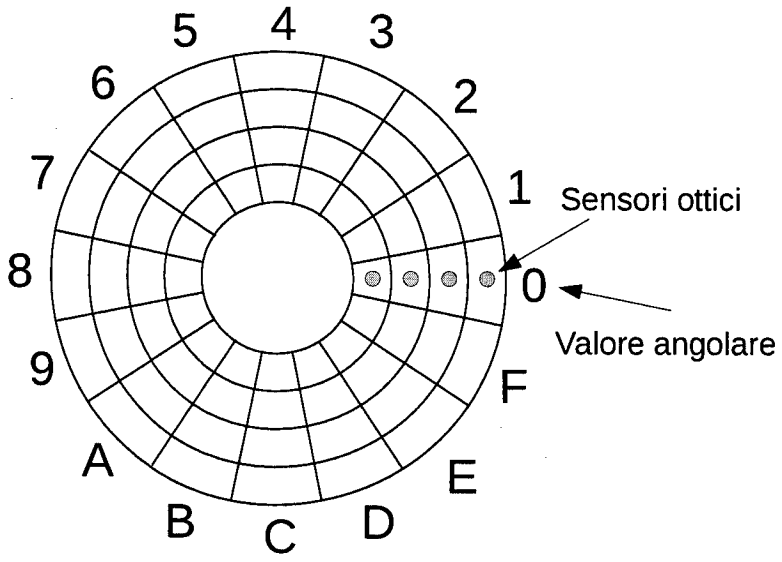
| SCHEDA **ASE1801** | Data: **29 Gennaio 2018** |
|---|---|
| Cognome | Nome |

## ESERCIZIO N°1
8 punti

Un encoder ottico rotativo (il cui disco è rappresentato in figura) è collegato ai 4 pin meno significativi della porta C di un microcontrollore AVR XMEGA256A3BU, mentre ai 4 pin più significativi della stessa porta è collegato un display esadecimale. I fotosensori dell'encoder sono progettati in modo che a un settore bianco corrisponde un contatto chiuso mentre a un settore annerito corrisponde un contatto aperto.

Scrivere un programma assembly che legga continuamente lo stato dell'encoder (il cui disco **deve essere annerito** opportunamente) e scriva sul display la posizione letta. Si ha a disposizione una subroutine "configure" che configura correttamente i pin della porta C (i 4 bit meno significativi sono ingressi con pull-up; i 4 bit più significativi sono uscite totem pole) e li rende disponibili sulla porta virtuale 0. Inoltre si può supporre che nello spazio di memoria non volatile (0x1000-0x1FFF) siano già presenti valori e tabelle di costanti opportune (memorizzate in fase di programmazione).




## ESERCIZIO N°2
6 punti

Realizzare la subroutine "configure" dell'esercizio precedente.

## ESERCIZIO N°3
4 punti

Realizzare con flip-flop $T$, porte logiche elementari e multiplexer, un contatore in discesa con abilitazione e reset che, al termine del conteggio ($Q = 0$) riparta dal valore 12.

## ESERCIZIO N°4
5 punti

Sintetizzare in forma ottima SP una rete combinatoria a 5 ingressi $x_2$, $x_1$, $x_0$ e $y_1$, $y_0$ (che sono le cifre binarie dei valori senza segno $X$ e $Y$) in grado di porre in uscita il bit meno significativo del risultato dell'espressione $\lfloor (X+2)/(Y+1) \rfloor \bmod 3$ .

## ESERCIZIO N°5

4 punti

Progettare una rete di Moore con abilitazione e reset (prioritario) che genera in uscita una sequenza periodica il cui periodo (pari a 7 cicli di clock) è costituito dai valori 0101010. La presenza del reset fa ripartire la sequenza dall'inizio.

## ESERCIZIO N°6

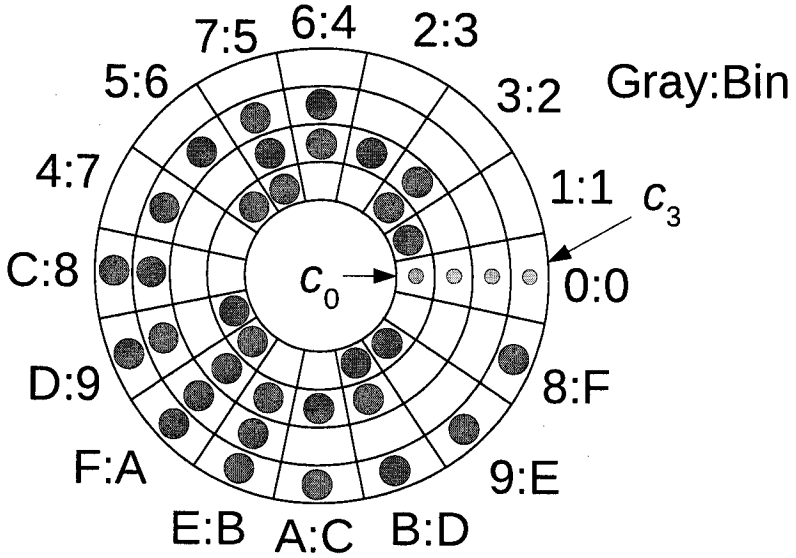
6 punti

Si hanno le rappresentazioni binarie di 3 numeri relativi su 8 bit (con una legge di rappresentazione non nota e non necessariamente uguale nei vari casi, appartenente a una tra MS, C2, C1 e Traslazione): $A$ = 01111110 $B$ = 01010111 e $C$ = 10001101.
Una rete combinatoria esegue correttamente la somma $S$ dei valori dati, ne verifica la rappresentabilità in C2 su 8 bit, e la rappresenta come $S$ = 11001000.
Determinare, se possibile, il valore di $A$, $B$ e $C$ e le relative leggi di rappresentazioni usate, coerenti con il codice binario dato.

# 1

Un encoder ottico rotativo (il cui disco è rappresentato in figura) è collegato ai 4 pin meno significativi della porta C di un microcontrollore AVR XMEGA256A3BU, mentre ai 4 pin più significativi della stessa porta è collegato un display esadecimale. I fotosensori dell'encoder sono progettati in modo che a un settore bianco corrisponde un contatto chiuso mentre a un settore annerito corrisponde un contatto aperto.

Scrivere un programma assembly che legga continuamente lo stato dell'encoder (il cui disco **deve essere annerito** opportunamente) e scriva sul display la posizione letta. Si ha a disposizione una subroutine "configure" che configura correttamente i pin della porta C (i 4 bit meno significativi sono ingressi con pull-up; i 4 bit più significativi sono uscite totem pole) e li rende disponibili sulla porta virtuale 0. Inoltre si può supporre che nello spazio di memoria non volatile (0x1000-0x1FFF) siano già presenti valori e tabelle di costanti opportune (memorizzate in fase di programmazione).




```
/*
Sfruttando il suggerimento, si può eseguire la conversione usando una tabella
contenuta in EEPROM, in 16 locazioni consecutive a partire da 0x1000.*/

.cseg //ritorna alla memoria di programma; serve dopo una direttiva .eseg
init:
 rcall configure //predispone la porta
 ldi XL,low(LUT_BASE+EE_BASE)
 ldi XH,high(LUT_BASE+EE_BASE)
loop:
 in R16,VPORT0_IN //legge l'encoder (0 - contatto chiuso - se bianco)
 andi R16,0x0F //toglie i bit più significativi al dato letto
 andi XL,0x0F //toglie i bit più significativi al puntatore
 add XL,R16 //calcola indirizzo LUT
 ld R17,X
 out VPORT0_OUT,R17 //valore esadecimale da LUT
 rjmp loop //ripeti per sempre
```

# 2

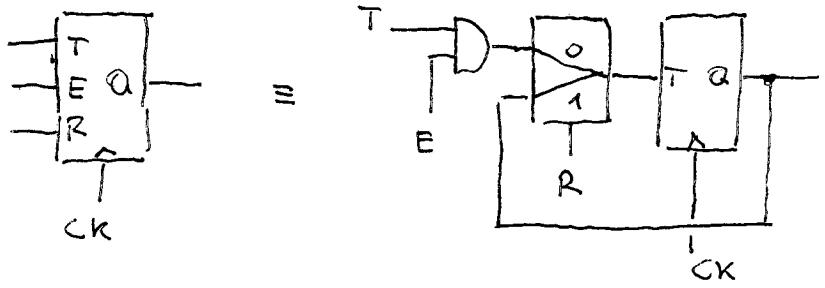
Realizzare la subroutine "configure" dell'esercizio precedente. (Nella soluzione proposta è presenta anche la parte che predispone la EEPROM, non richiesta dal testo dell'esercizio.)

```
.equ LUT_BASE=0x0000 //posizione della LUT con il digit meno significativo nullo
.equ EE_BASE=0x1000 //inizio della eeprom in memoria
.equ IN_POS=0b00001111 //ingressi 4 bit meno significativi
.equ IN_CFG=0b00011000 //pull-up (011 bit 3,4,5)
.equ OUT_POS=0b11110000 //uscite 4 bit più significativi, buono anche per DIR
.equ OUT_CFG=0b00000000 //totem pole senza pull (000 bit 3,4,5)
.equ VPORT0_C=0b00000010 //PORTC in VPORT0

/* Predisposizione per la LUT da Gray (indirizzo) a binario (contenuto)
 * i valori di conversione sono, per un Gray solito (vedi disegno encoder):
 * 0,1,3,2,7,6,4,5,F,E,C,D,8,9,B,A
 */
.eseg
.org LUT_BASE
.db 0x0,0x1,0x3,0x2,0x7,0x6,0x4,0x5,0xF,0xE,0xC,0xD,0x8,0x9,0xB,0xA

/* Qui ci va il codice principale, preceduto dalla direttiva ".cseg"

/* Subroutine di configurazione
 */
configure:
 push R16
 push R17
 ldi R16,IN_POS
 ldi R17,IN_CFG
 sts PORTCFG_MPCMASK,R16
 sts PORTC_PIN0CTRL,R17 //PIN0 è tra gli ingressi
 ldi R16,OUT_POS
 ldi R17,OUT_CFG
 sts PORTCFG_MPCMASK,R16
 sts PORTC_PIN4CTRL,R17 //PIN4 è tra le uscite
 ldi R16,VPORT0_C
 sts PORTCFG_VPCTRLA,R16 //PORTC viene vista come VPORT0
 ldi R16,0 //valore iniziale per l'uscita
 ldi R17,OUT_POS //i 4 pin più significativi in uscita
 sts VPORT0_OUT,R16
 sts VPORT0_DIR,R17
 pop R17
 pop R16
 ret
```

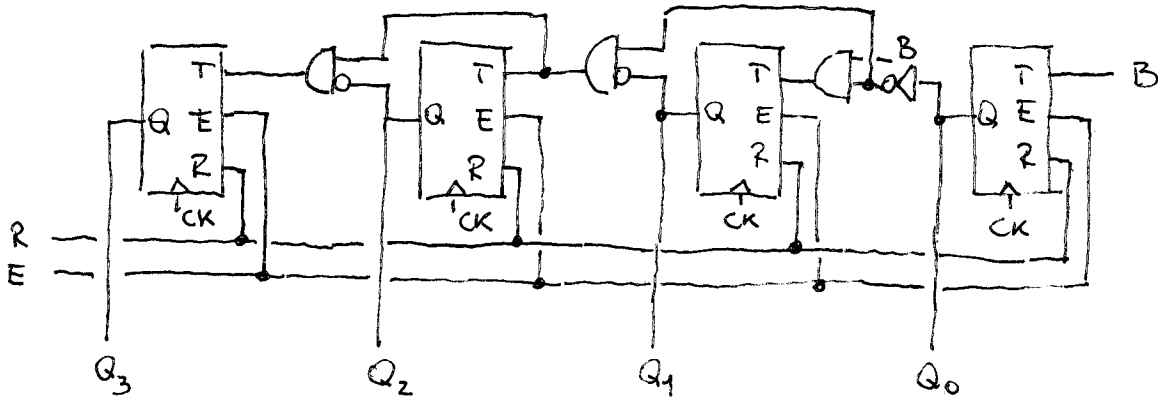
(3) Contatore DOWN modulo 13 con abilitazione E e reset R (priorit.)

sequenza
0 0 0 0 } transizione finale
1 1 0 0 }
- - B B

Viste le specifiche, uso come elemento un T-FF con E ed R priorit. (entrambi sincroni)




Risolta la funzionalità di E ed R, il circuito diventa




(4) MAPPA $\lfloor (x+2)/(Y+1) \rfloor_{mod3,\, mod2}$

$x_2 = 0$

| $y_1y_0$ \ $x_1x_0$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 0 | 0 | 0 | 1* |
| 01 | 1* | 1 | 0 | 0 |
| 11 | 0 | 0 | 1* | 1 |
| 10 | 0 | 1* | 1 | 1 |
| | 2 | 3 | 5 | 4 |

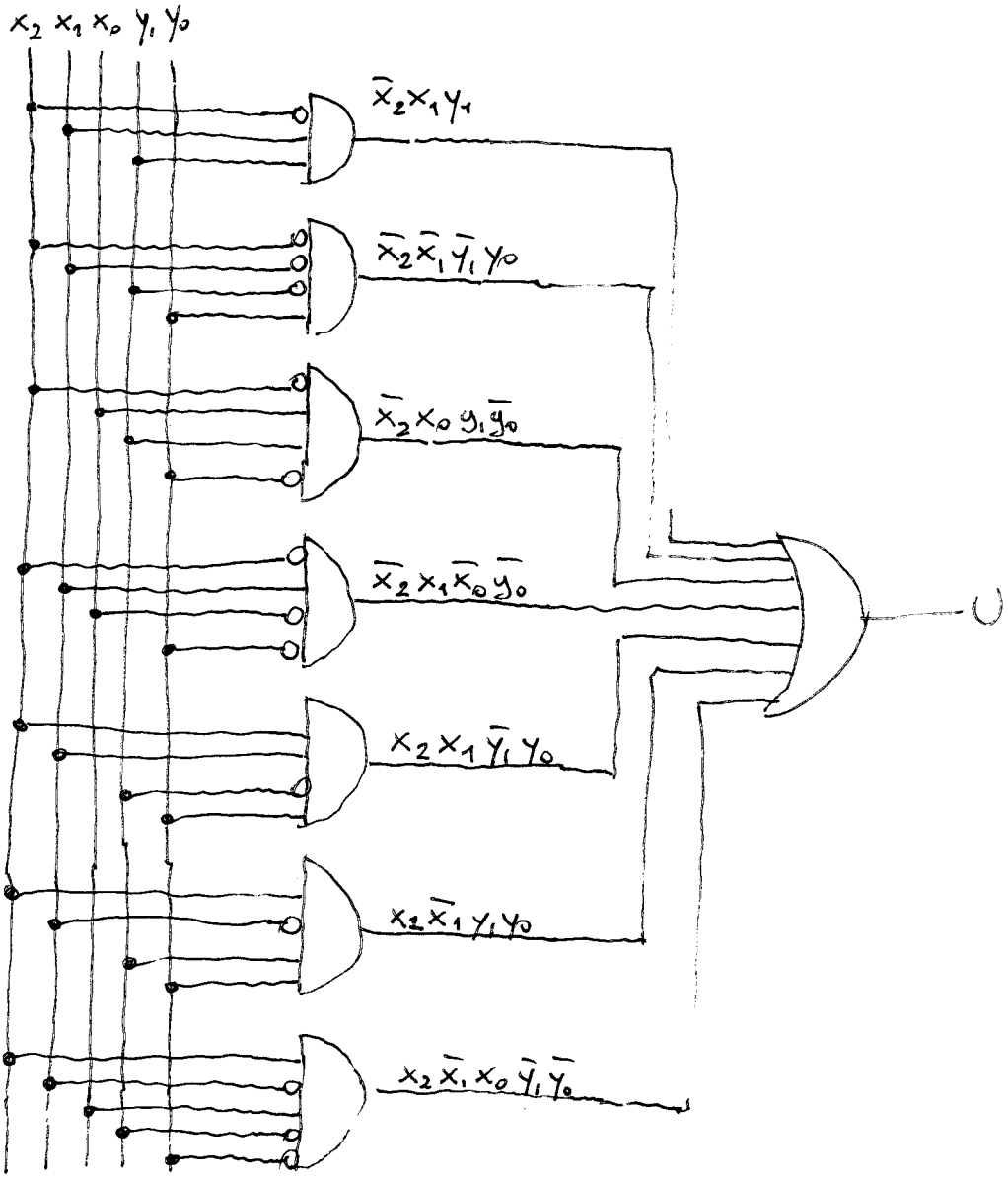
$x_2 = 1$

| $y_1y_0$ \ $x_1x_0$ | 00 | 01 | 11 | 10 | |
|---|---|---|---|---|---|
| 00 | 0 | 1* | 0 | 0 | 1 |
| 01 | 0 | 0 | 1* | 1 | 2 |
| 11 | 1* | 1 | 0 | 0 | 4 |
| 10 | 0 | 0 | 0 | 0 | 3 |
| | 6 | 7 | 9 | 8 | |

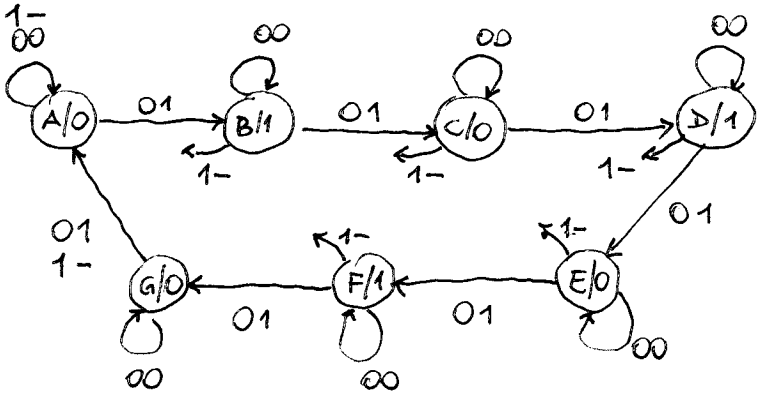
$(Y+1)$

$(X+2)$

La sintesi ottima SP è fatta da 7 implicanti, tutti ESSENZIALI (con * un mintermine coperto esclusivamente dall'implicante indicato)



(5) Grafo (ingressi R, E)




Se R=1 (con E qualunque) lo stato futuro è A

Codifica degli stati (stile "contatore")

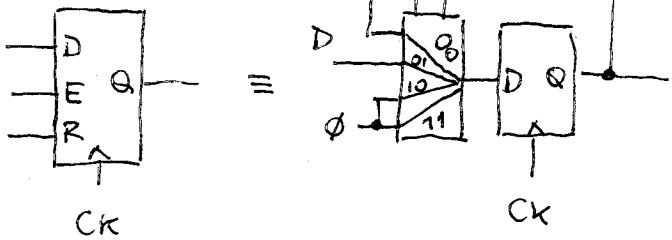
| | |
|---|---|
| A | 000 |
| B | 001 |
| C | 010 |
| D | 011 |
| E | 100 |
| F | 101 |
| G | 110 |
| | $q_2 q_1 q_0$ |

$q_0$ coincide con l'uscita

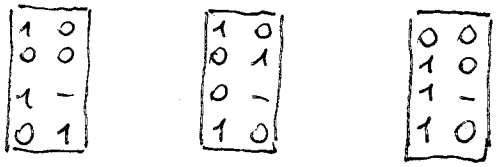
Architettura: scelgo la sintesi con D con E e R prioritario
rete di uscita cortocircuito

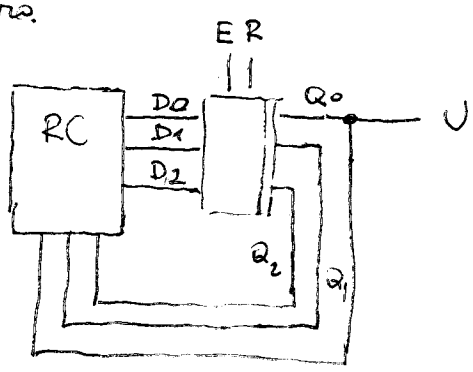
Mappa delle transizioni

| $q_2 q_1$ \ $q_0$ | 0 | 1 |
|---|---|---|
| 00 | 110 | 000 |
| 01 | 001 | 010 |
| 11 | 101 | --- |
| 10 | 011 | 100 |




Architettura



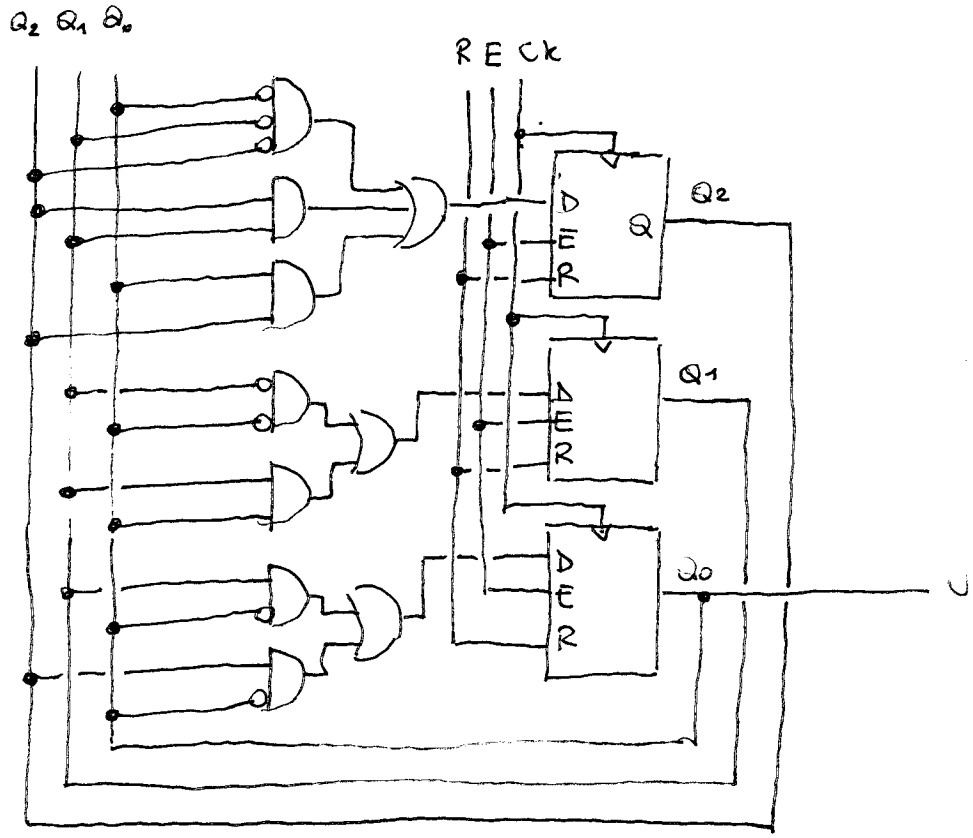




$D_2 = \bar{q}_2 \bar{q}_1 \bar{q}_0 + q_2 q_1 + q_2 q_0$

$D_1 = \bar{q}_1 \bar{q}_0 + q_1 q_0$

$D_0 = q_1 \bar{q}_0 + q_2 \bar{q}_0$

Schema



(6) Determiniamo il valore dei CODICI dati per le diverse codifiche

| | A | B | C |
|---|---|---|---|
| MS | 126 | 87 | -13 |
| C2 | 126 | 87 | -115 |
| C1 | 126 | 87 | -114 |
| T | -2 | -41 | 13 |

Esaminiamo tutte le possibili combinazioni di somma (sono 16) - il risultato deve essere rappresentabile e pari a $S = -56$

(per valori POSITIVI * è indifferentemente C1, C2 e MS)

| A | B | C | (A+B) + C | | Commento |
|---|---|---|---|---|---|
| * | * | MS | 213 | -13 | 200 overflow |
| | | C2 | | -115 | >0 no |
| | | C1 | | -114 | >0 no |
| | | T | | +13 | 226 overflow |
| * | T | MS | 139 | -13 | >0 no |
| | | C2 | | -115 | >0 no |
| | | C1 | | -114 | >0 no |
| | | T | | +13 | >0 e overflow |
| T | * | MS | 85 | -13 | >0 no |
| | | C2 | | -115 | -30 no |
| | | C1 | | -114 | -29 no |
| | | T | | +13 | >0 no |
| T | T | MS | -43 | -13 | -56 OK ← |
| | | C2 | | -115 | overflow |
| | | C1 | | -114 | overflow |
| | | T | | +13 | -30 no |

c'è quindi un'unica possibilità:

A = -2 T

B = -41 T

C = -13 MS