

EsD4 ADC-DAC: Convertitore sigma-delta

Gruppo 1.AC
Matteo Rossi, Bernardo Tomelleri

11 maggio 2022

Misura componenti dei circuiti

Riportiamo per completezza il valore della tensione continua di alimentazione per i circuiti integrati misurata con il multimetro

$$V_{CC} = 4.99 \pm 0.03 \text{ V}$$

e il valore di capacità del condensatore di disaccoppiamento che collega le linee di alimentazione a massa (sempre misurato con il multimetro)

$$C_d = 97 \pm 4 \text{ nF}$$

1 Analisi e costruzione del circuito

Si vuole costruire una macchina a stati finiti in grado di emulare il funzionamento delle luci di un semaforo stradale, nelle modalità

ABILITATO si ripetono in ciclo gli stati: LED verde acceso → LED giallo e verde accesi → LED rosso acceso.

DISABILITATO si alternano gli stati LED giallo acceso e spento (giallo lampeggiante).

in cui tutti gli stati hanno durata pari ad un impulso del clock inviato all'ingresso dei circuiti integrati a disposizione.

2 Descrizione delle misure e acquisizione dati

Si vuole ricostruire il circuito precedente per il controllo di un semaforo tramite un microcontrollore (nel nostro caso utilizzeremo Eleego UNO R3). Dopo aver assemblato il circuito descritto in fig. 1, si procede con

Figura 1: Schematica utilizzata per la gestione del semaforo con software tramite Eleego.

l'implementazione software del semaforo.

2.a Definizione del segnale in ingresso

Ci basiamo sullo schema per il semaforo FSM di Mealy presente in ?? per controllare il funzionamento del semaforo tramite segnale di controllo ENABLE, riportiamo il listato del programma nel listing

2.b Campionamento e acquisizione del segnale

Si introduce un quarto stato nel caso in cui il semaforo sia abilitato, successivo al rosso; per fare ciò è necessaria una semplice modifica all'interno dello switch nel caso in cui il semaforo risulti abilitato. Riportiamo le modifiche apportate al codice precedente nel listing 1.

```
1 void loop() {  
  //viene controllato se il semaforo \’e abilitato  
3  if(CheckEnable()){  
  
5    // nel caso in cui il semaforo sia abilitato  
    switch(CurrentState){  
7    case 1://se lo stato precedente risulta essere solo rosso
```

```

9      CurrentState=2;
      digitalWrite(6,HIGH); //imposta lo stato corrente a rosso e giallo
      delay(tickREDYELLOW); //viene fatto passare il tempo specificato all'inizio del file
11     break;
      case 2: //se lo stato precedente risulta essere rosso e giallo
13         CurrentState=3; //lo stato corrente viene spostato a verde
            digitalWrite(5,LOW);
            digitalWrite(6,LOW);
            digitalWrite(7,HIGH);
15         delay(tickGREEN);
            break;
17     .
19     .
21     .

```

Listing 1: Svizzero.ino

2.c Acquisizione con Protocol

3 Analisi dei dati

Si vuole realizzare una FSM che riceve uno stream di bit su una linea di ingresso e che accende un LED tutte le volte che su questo si presenta un fronte di discesa secondo il modello di Moore e un'altra secondo quello di Mealy.

3.a Ricostruzione dei segnali in ingresso

Seguendo lo stesso procedimento usato nella sezione 1 per realizzare il semaforo disegniamo i diagrammi degli stati delle macchine, ne codifichiamo in bit gli stati, ricaviamo la tabella di verità per le transizioni tra questi e i valori in uscita, così da ottenere le equazioni logiche che le governano da tradurre nella realizzazione pratica delle connessioni tra memoria e circuiti di logica combinatoria.

3.b Fit sinusoidale

Si è scelto di costruire la FSM di Moore per riconoscere fronti di discesa con 3 stati, dunque 2 bit di memoria Q_1Q_0 secondo il diagramma degli stati in fig. 2. Per cui se il segnale di rumore in ingresso rimane a livello logico basso o alto la macchina resta in stato LOW (00) o HIGH (01) in cui l'uscita è $OUT = 0$; se invece il segnale IN scende a 0 durante lo stato HIGH la macchina entra nell'unico stato (EDGE := 10) in cui l'uscita è $OUT = 1$ prima di tornare ad uno degli stati precedenti a seconda del valore logico dell'ingresso al successivo fronte di salita del clock.

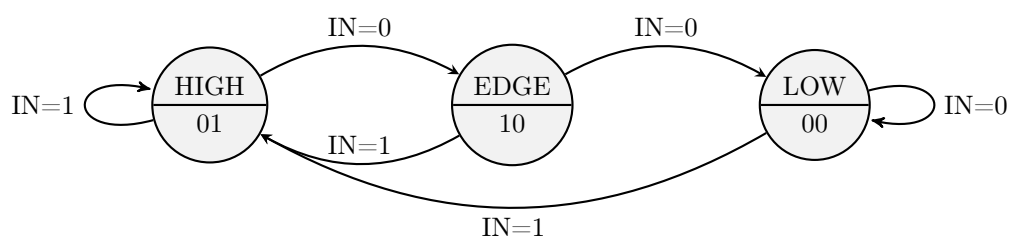


Figura 2: Edge-detector FSM di Moore

Dal diagramma ricaviamo la tabella di verità delle transizioni di stato con la stessa convenzione usata prima tra lo stato corrente Q_i e il successivo D_i come riportata in tabella 1.

Dunque per semplificare le espressioni logiche delle transizioni e delle uscite si sono costruite le mappe di Karnaugh riportate in tabella 2, tabella 3 e tabella 4.

Finalmente dalle equazioni per gli stati futuri e delle uscite della FSM si è costruito il circuito riportato in fig. 3 a partire da due positive-edge-triggered D Flip-Flop (IC 74LS74), 1 porta AND (74LS08) e una porta NOT (74LS04).

3.c Risposta in frequenza dell'ADC

Per la realizzazione della FSM di Mealy per riconoscere fronti di discesa invece sono sufficienti 2 stati, quindi 1 bit di memoria, come si vede anche dal diagramma degli stati riportato in fig. 4. Anche in questo caso il

State	IN	Q_1	Q_0	D_1	D_0	OUT
LOW	0	0	0	0	0	0
	1	0	0	0	1	0
EDGE	0	1	0	0	0	1
	1	1	0	0	1	1
HIGH	0	0	1	1	0	0
	1	0	1	0	1	0

Tabella 1: Tabella di verità per le transizioni tra gli stati del detector di Moore.

IN	Q_1Q_0			
	00	01	11	10
0	0	0	X	0
1	1	1	X	1

Tabella 2: Mappa di Karnaugh per $D_0 = \text{IN}$

IN	Q_1Q_0			
	00	01	11	10
0	0	1	X	0
1	0	0	X	0

Tabella 3: Mappa di Karnaugh per $D_1 = \overline{\text{IN}} \cdot Q_0$

IN	Q_1Q_0			
	00	01	11	10
0	0	0	X	1
1	0	0	X	1

Tabella 4: Mappe di Karnaugh per $\text{OUT} = Q_1$

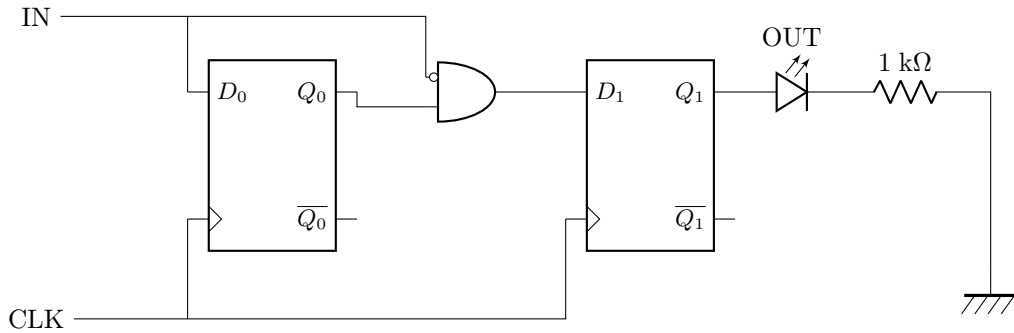


Figura 3: schema del detector Moore.

segnale di rumore viene inviato all'ingresso D del FF, ma ora viene comparato continuamente con lo stato precedentemente registrato in memoria Q . Dunque il valore in uscita dal circuito è sempre $OUT = 0$ apparte quando la macchina registra la discesa del segnale IN a 0 durante lo stato HIGH ($Q = 1$), per cui $OUT: 0 \rightarrow 1$ lungo il fronte di discesa del segnale di clock.

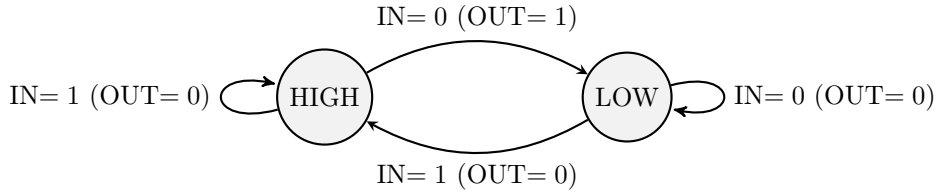


Figura 4: Edge detector FSM di Mealy

Come prima ricaviamo dal diagramma la tabella di verità delle transizioni di stato riportata in tabella 5 con le solite convenzioni; stavolta non è necessario ricorrere alle mappe di Karnaugh per semplificare le espressioni dello stato futuro $D = IN$ e dell'uscita $OUT = \overline{IN} \cdot Q$

$D = IN$	Q	OUT
0	0	0
0	1	1
1	0	0
1	1	0

Tabella 5: codifica binaria degli stati del detector di Mealy. $D = IN$; $OUT = \overline{IN} \cdot Q$

A partire da queste equazioni si è costruito il circuito riportato in fig. 5 con un positive-edge-triggered D Flip-Flop (IC 74LS74), 1 porta AND (74LS08) e una porta NOT (74LS04).

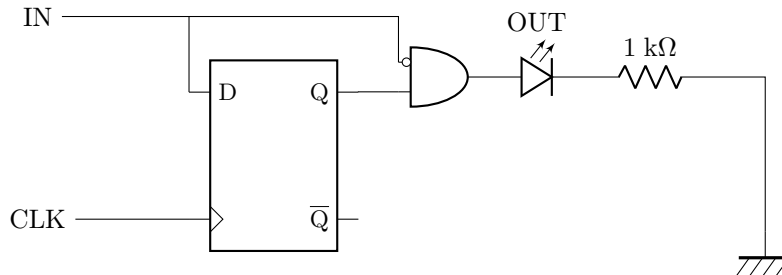


Figura 5: schema del detector Mealy.

3.d Stima del fattore di calibrazione del convertitore

Con la funzione Patterns di Waveform si invia un segnale (in DIO 5) di clock di frequenza $f_{clk} = 1 \text{ kHz}$ al pin (CLK) dei Flip-Flop e si genera con il canale DIO 6 uno stream di dati random alla stessa frequenza $f = f_{clk}$, già dalla schermata di Patterns è possibile notare come le commutazioni di stato del segnale pseudocasuale avvengano in corrispondenza dei fronti di discesa del segnale di clock.

Come prima si sono mantenuti i pin PRESET e CLEAR dei D-FF collegati a V_{CC} per evitare reset o clear spurii.

3.e Misura del signal/noise ratio (SNR)

Si sono acquisiti i segnali in ingresso (CLK = DIO 5, IN = DIO 6) e in uscita (OUT = DIO 7) dai circuiti edge-detector con la funzione Logic Analyzer dell'AD2, di cui riportiamo i risultati per il circuito di Moore in fig. 6 e per la FSM di Mealy in fig. 7.

Notiamo nel primo caso come l'uscita assume valore alto in maniera sincrona rispetto al fronte di salita del segnale di clock, mentre nell'implementazione di Mealy l'uscita sale in maniera sincrona rispetto al fronte di discesa del segnale di rumore IN.

Questo risulta compatibile con quanto ci si aspetta per le diverse temporizzazioni delle macchine a stati finiti realizzate secondo i modelli di Moore e Mealy

Figura 6: Acquisizione di un ciclo completo (frequenza 1 kHz) con Logic Analyzer dei segnali in ingresso e in uscita dall'edge detector di Moore.

Figura 7: Acquisizione di un ciclo completo (frequenza 1 kHz) con Logic Analyzer dei segnali in ingresso e in uscita dal detector di Mealy.

Conclusioni e commenti finali

Si è riusciti a progettare, costruire e verificare il corretto funzionamento di circuiti logici combinatori di diversa complessità e svariate applicazioni (e.g., sistemi di controllo e misura) costruiti con porte NOT, NAND, OR e D-FF. Inoltre si è riusciti ad apprezzare le diverse modalità di funzionamento delle macchine a stati finiti implementate secondo i modelli Moore e Mealy, ponendo particolare attenzione alle loro diverse temporizzazioni nei cambiamenti di stato, nonostante la bassa risoluzione temporale dell'AD2.

Dichiarazione

I firmatari di questa relazione dichiarano che il contenuto della relazione è originale, con misure effettuate dai membri del gruppo, e che tutti i firmatari hanno contribuito alla elaborazione della relazione stessa.