

# Automazione

27 Marzo 2018

## Esercizio 1

Al livello fisico del mezzo trasmissivo di una rete di automazione, l'informazione binaria da trasmettere può essere codificata con diverse modalità. Con riferimento alla Fig. 1, si vuole trasmettere un segnale binario in modo sincrono, avendo a disposizione un clock a frequenza doppia di quella del segnale utile. Si richiede l'uso di due differenti codifiche, denominate Manchester e Miller. In entrambi i casi la codifica dovrà essere bipolare, in modo da poter associare lo stato di riposo del canale ad assenza di segnale da trasmettere. La codifica Manchester è cosidefinita:

- lo stato 1 è rappresentato con una transizione al semi-periodo dal livello alto a quello basso;
- lo stato 0 è rappresentato con una transizione al semi-periodo dal livello basso a quello alto.

La codifica Miller è di tipo dinamico ed è definita come segue:

- lo stato 1 è rappresentato mantenendo all'inizio del periodo il livello dello stato precedente ed effettuando una transizione al semi-periodo;
- lo stato 0 è rappresentato in due modi:
  - se lo stato precedente era 1, si mantiene il livello precedente per tutto il periodo;
  - se lo stato precedente era 0, si ha una transizione a inizio periodo e si mantiene poi il livello raggiunto.

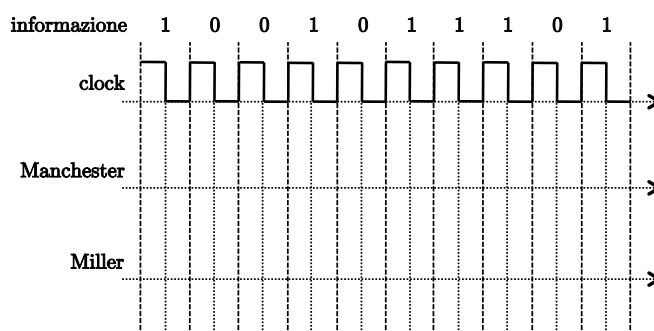


Figura 1: Tracciare i diagrammi per il segnale  $\{1, 0, 0, 1, 0, 1, 1, 1, 0, 1\}$  con le codifiche indicate.

## Esercizio 2

Una linea di produzione deve gestire cinque prodotti  $\{A, B, C, D, E\}$  che necessitano di due lavorazioni in sequenza, la prima effettuata sulla macchina #1 e la seconda sulla macchina #2. In Tab. 1 sono riportati i tempi necessari per le lavorazioni dei diversi prodotti su ciascuna macchina. Sequenziare le lavorazioni sulle due macchine in modo da minimizzare il tempo totale di completamento dei cinque prodotti, descrivendo i singoli passi dell'algoritmo applicato. Riportare il risultato graficamente su un diagramma di Gantt.

Prodotto	A	B	C	D	E
Macchina #1	10	2	1	4	5
Macchina #2	3	5	3	5	6

Tabella 1: Tempi di lavorazione (in minuti) dei cinque prodotti sulle due macchine.

### Esercizio 3

Nella cella in Fig. 2, un robot  $R$  deve prelevare una sequenza di pezzi di due tipi  $A$  e  $B$  che arrivano in ordine sparso e con intervalli casuali su un nastro trasportatore  $N$  in ingresso, smistandoli a seconda del tipo sui due nastri trasportatori  $NA$  e  $NB$  in uscita. Ogni volta che viene effettuato uno smistamento, il supervisore di cella segnala l'evento con il tipo di pezzo. Durante il trasporto dal nastro  $N$  a uno dei due nastri in uscita e nel tragitto di ritorno, è possibile che un pezzo in ingresso venga perso perché il robot è ancora impegnato. Anche per questo evento va segnalato il tipo di pezzo perso. Modellare il comportamento della cella come automa a stati finiti rappresentato mediante una macchina di Mealy, descrivendo l'insieme degli stati, degli eventi in ingresso e di quelli in uscita.

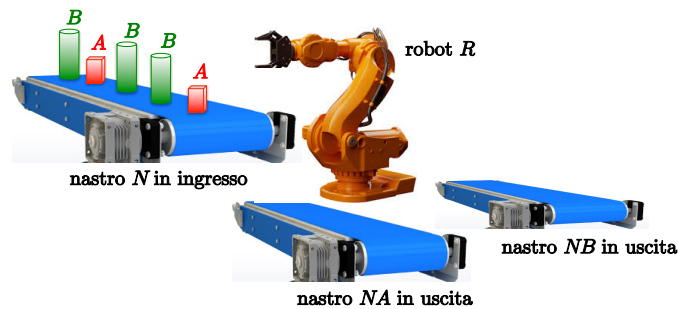


Figura 2: Una cella con un robot che smista pezzi in ingresso su due nastri in uscita.

### Esercizio 4

Si consideri un sistema di automazione in cui, a livello di coordinamento, è necessario portare a termine i seguenti task periodici:

1. ogni 4 t.u. un filo metallico viene tagliato e deformato impiegando 1 t.u.;
2. ogni 16 t.u. i fili metallici vengono saldati assieme impiegando 3 t.u.;
3. ogni 8 t.u. i fili metallici vengono spruzzati da un antiossidante impiegando 1 t.u.

A questi task si aggiunge un task aperiodico di pulizia degli elettrodi di saldatura, il quale deve intervenire con un intervallo minimo di 32 t.u. e durare al massimo 8 t.u. Per semplicità si ipotizzi che i task periodici siano indipendenti l'uno dall'altro. I quattro suddetti task devono essere gestiti con una modalità di scheduling hard real time.

Il sistema di automazione viene supervisionato da un tecnico il quale, attraverso un'interfaccia uomo-macchina costituita da un touch screen, può visualizzare una schermata di stato del sistema. Il task di interfacciamento uomo-macchina è aperiodico e cosiccaratterizzato: l'activation time della prima occorrenza è all'istante  $a_5(1) = 20$  t.u., il computation time è pari a  $C_5(1) = 6$  t.u. e la deadline assoluta è pari a  $D_5(1) = 32$  t.u. Questo task deve essere servito con una modalità di scheduling soft real time.

1. Verificare che il problema di scheduling hard real time sia ammissibile.
2. Nel caso il problema sia ammissibile, verificare se sussiste una condizione sufficiente di schedulabilità con algoritmo RMPO.
3. Mostrare il risultato dello scheduling usando RMPO come algoritmo di task scheduling hard real time e un servizio in background con scheduling di tipo FIFO per i task soft real time.
4. Determinare se il task soft real time viene eseguito entro la deadline assoluta.

[120 minuti; libri aperti]

# Soluzioni

27 Marzo 2018

## Esercizio 1

Le codifiche Manchester e Miller del segnale richiesto sono riportate nella Fig. 3. Per la codifica di Miller, in virtù della dinamicità della stessa, occorre ipotizzare un livello di segnale nell'intervallo precedente a quello indicato. In questo caso, si è ipotizzato che il segnale precedente fosse alto (il grafico sarebbe quello logicamente negato nell'ipotesi di segnale precedente a livello basso). Si noti che il livello alto (valore binario 1) corrisponde a una tensione  $V > 0$  e il livello basso (valore binario 0) a una tensione  $-V < 0$ . Il livello di riposo del canale è quindi riservato ad assenza di segnale utile.

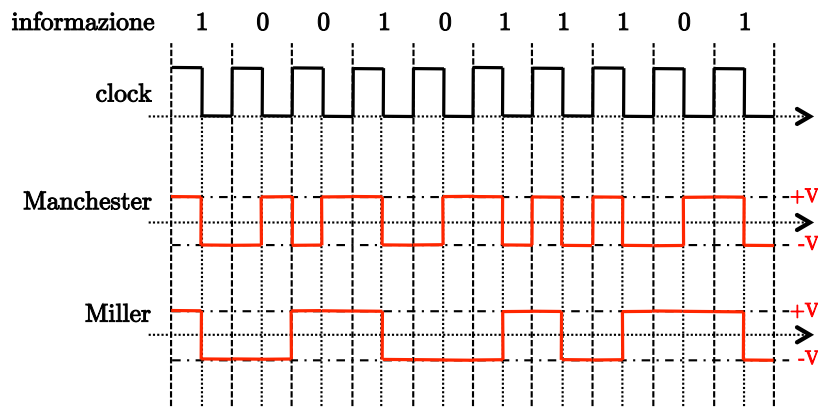


Figura 3: Diagrammi temporali delle codifiche bipolari Manchester e Miller per il segnale  $\{1, 0, 0, 1, 0, 1, 1, 1, 0, 1\}$ .

## Esercizio 2

Si tratta di un problema di flow shop scheduling con  $n = 5$  lavori su  $m = 2$  macchine, per il quale è possibile applicare l'algoritmo di Johnson che minimizza il tempo totale di completamento (*makespan*). L'algoritmo processa per primo il prodotto  $i$  ( $i = A, \dots, E$ ) con il minimo tempo di lavorazione  $t_{ij}$ . In questo caso,  $i = C$  (il minimo tempo nella tabella è  $t_{C1} = 1$ ). Poiché  $t_{C1} = 1 < 3 = t_{C2}$ , il prodotto  $C$  verrà schedulato per primo. Rimosso  $C$  dalla lista dei prodotti, il minimo tempo di lavorazione sui restanti è  $t_{B1} = 2$ . Poiché  $t_{B1} = 2 < 5 = t_{B2}$ , il prodotto  $B$  verrà posto come secondo, ossia immediatamente dopo il primo. Rimosso anche  $B$  dalla lista dei prodotti, il minimo tempo di lavorazione sui restanti è  $t_{A2} = 3$ . Poiché  $t_{A1} = 10 > 3 = t_{A2}$ , il prodotto  $A$  verrà invece schedulato per ultimo. Tra i due rimanenti lavori, quello con il minore tra i tempi di lavorazione è  $D$ , con  $t_{D1} = 4$ . Poiché  $t_{D1} = 4 < 5 = t_{D2}$ , il prodotto  $D$  verrà invece sequenziato come terzo. Di conseguenza  $E$  sarà in quarta posizione. La soluzione ottima fornita dall'algoritmo è quindi  $\{C, B, D, E, A\}$ , illustrata nel diagramma di Gantt in Fig. 4.



Figura 4: Diagramma di Gantt con la soluzione del problema di flow shop scheduling.

Il tempo totale è pari a  $T = 25$  minuti. Si noti che questo sequenziamento ottimo è unico (non è stato infatti necessario operare scelte tra tempi di lavorazione uguali). Le due somme dei tempi

delle lavorazioni sulla macchina #1 e sulla #2 sono tra loro uguali e pari a 22 minuti. La prima macchina rimane quindi a riposo negli ultimi 3 minuti. La seconda macchina rimane in attesa per un totale di 3 minuti, 1 minuto all'inizio della sequenza e 2 minuti in attesa del termine della lavorazione del prodotto  $A$  da parte della prima macchina.

### Esercizio 3

L'automa che descrive il comportamento della cella è mostrato in Fig. 5.

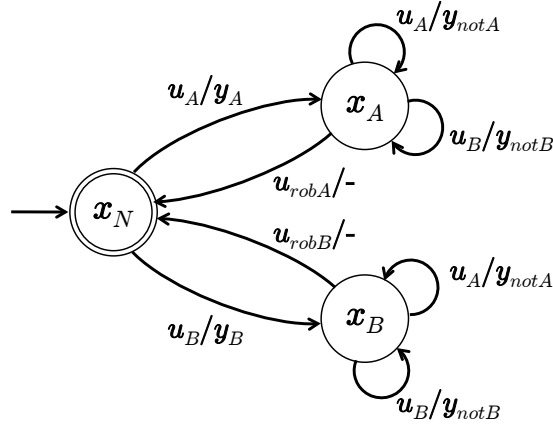


Figura 5: Automa che rappresenta il comportamento della cella in Fig. 2.

I 3 stati, i 4 eventi in ingresso e i 4 in uscita sono:

- $x_N$  = robot disponibile in attesa davanti al nastro  $N$
- $x_A$  = robot impegnato nello smistamento di un pezzo  $A$  verso il nastro  $NA$  e ritorno
- $x_B$  = robot impegnato nello smistamento di un pezzo  $B$  verso il nastro  $NB$  e ritorno
- $u_A$  = arrivo pezzo  $A$  sul nastro  $N$
- $u_B$  = arrivo pezzo  $B$  sul nastro  $N$
- $u_{robA}$  = fine smistamento di un pezzo  $A$  con rientro del robot
- $u_{robB}$  = fine smistamento di un pezzo  $B$  con rientro del robot
- $y_A$  = pezzo  $A$  correttamente smistato
- $y_B$  = pezzo  $B$  correttamente smistato
- $y_{notA}$  = pezzo  $A$  perso
- $y_{notB}$  = pezzo  $B$  perso

Nel diagramma dell'automa, si è indicato con un trattino (-) l'assenza di un evento in uscita.

### Esercizio 4

La verifica di non inammissibilità si effettua calcolando il fattore di utilizzazione dei task periodici hard real time:

$$U = \frac{1}{4} + \frac{3}{16} + \frac{1}{8} + \frac{8}{32} = \frac{8 + 6 + 4 + 8}{32} = \frac{26}{32} = 0.8125 < 1.$$

Soddisfatta la condizione necessaria, verifichiamo se sussiste almeno una condizione sufficiente:

$$U_{lsm}(RMPO) = n \left( 2^{1/\sqrt{n}} - 1 \right) = 4 \left( 2^{1/\sqrt{4}} - 1 \right) \simeq 0.7568.$$

Dato che  $U > U_{lsm}$ , questa condizione sufficiente non è verificata. Si può però notare che i 4 task sono legati tra loro da relazioni armoniche. Pertanto abbiamo individuato una condizione sufficiente. La soluzione dello scheduling RMPO con servizio in background FIFO è mostrata in Fig. 6.

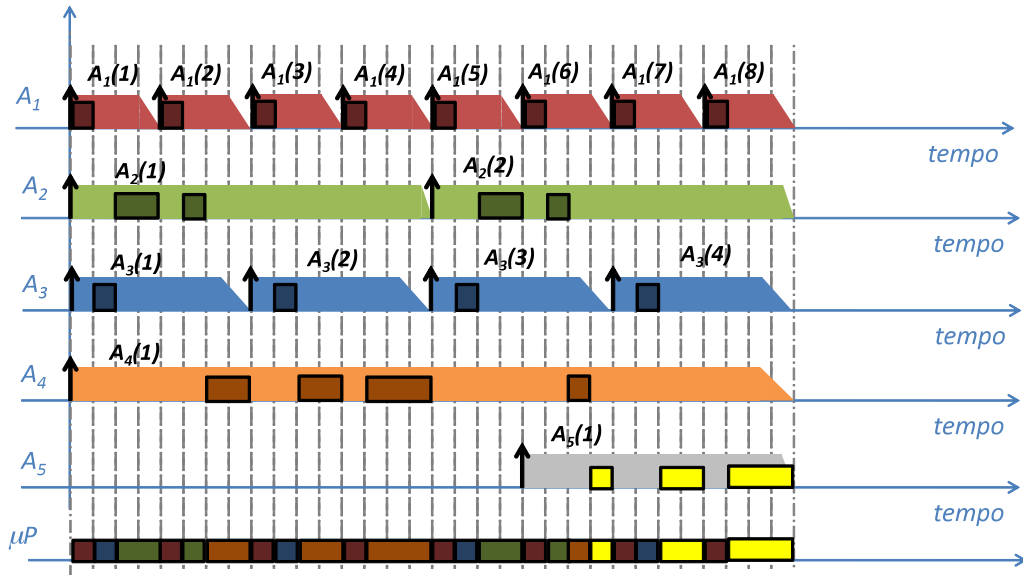


Figura 6: Scheduling RMPO con servizio in background FIFO.

Da questa si evince che il task soft real time può essere portato a termine entro la deadline assoluta.

\* \* \* \* \*