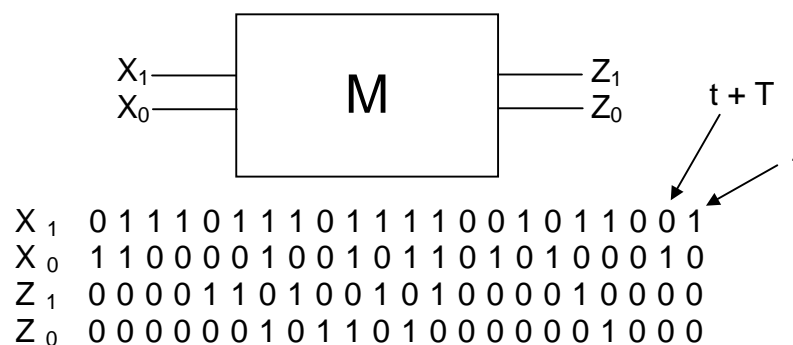


# Corso di Calcolatori elettronici A

## Compito scritto A – 22/3/2005

- 1) Progettare con un automa di Moore ed implementare con sole porte NAND, dopo aver minimizzato il numero delle porte stesse, la macchina sequenziale M, le cui uscite sono normalmente a zero, fino a quando si presenta la coppia (0,0) agli ingressi  $X_1$  e  $X_0$ . Le uscite  $Z_1$  e  $Z_0$  diventano quindi i due bit di un numero che rappresenta il conteggio in modulo 3 dei bit pari a 1 che si presentano sulle linee di ingresso. Quando il conteggio è pari a 0 ovvero quando  $Z_1 Z_0$  coincide con 00, la macchina M è ritornata nello stato iniziale con le uscite a zero in attesa dell'ingresso (0,0) per riprendere eventualmente il conteggio suddetto. In figura è riportato un esempio.



- 2) Un calcolatore con frequenza di clock pari a 2,4GHz e CPI in assenza di fallimenti pari a 2 cicli di clock, dispone di una cache primaria e di una cache secondaria con le seguenti caratteristiche:
- Cache primaria di 512KB a corrispondenza diretta con blocchi di 4 parole. Frequenza di successo per le istruzioni pari al 95% e per i dati pari al 90%. Per l'accesso in caso di hit viene impiegato un ciclo di clock.
  - Cache secondaria di 128KB set-associative a 2 vie con blocchi di 4 parole. Frequenza di fallimento per le istruzioni pari al 3% e per i dati pari al 5%. Per l'accesso in caso di hit vengono impiegati 2 cicli di clock.
- La penalità di fallimento per l'accesso in memoria principale è di 10 cicli di clock. Supponendo si abbia l'architettura MIPS, determinare il tag espresso come numero esadecimale per l'indirizzo 0x7ABCEFE8 in relazione alle due cache e il tempo di CPU, supponendo che il 24% delle istruzioni I di un programma di test facciano accesso ai dati.
- 3) Scrivere una procedura in assembler MIPS che riceva in ingresso l'indirizzo di un vettore di interi ed il numero dei suoi elementi e per ogni elemento del vettore stesso effettui la seguente assegnazione:

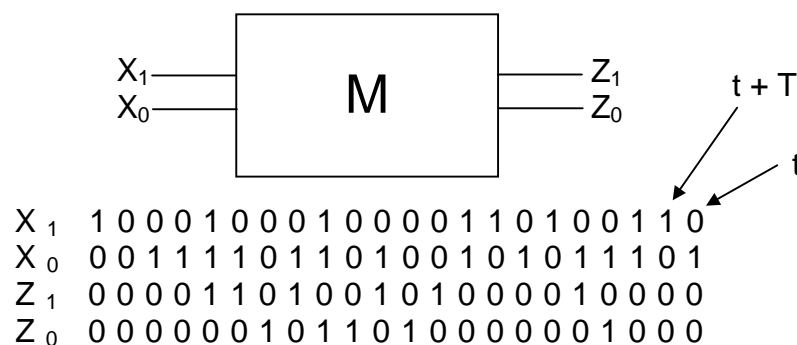
```
if a[i] >= 1024 then
    a[i] = f(a[i]) - g(a[i]) + 128
else
    a[i] = 256 - [g(a[i]) - f(a[i])]
```

Dove con  $f()$  e  $g()$  si sono indicate due funzioni a disposizione del programmatore.

# Corso di Calcolatori elettronici A

## Compito scritto B – 22/3/2005

- 1) Progettare con un automa di Moore ed implementare con sole porte NAND, dopo aver minimizzato il numero delle porte stesse, la macchina sequenziale M, le cui uscite sono normalmente a zero, fino a quando si presenta la coppia (1,1) agli ingressi  $X_1$  e  $X_0$ . Le uscite  $Z_1$  e  $Z_0$  diventano quindi i due bit di un numero che rappresenta il conteggio in modulo 3 dei bit pari a 0 che si presentano sulle linee di ingresso. Quando il conteggio è pari a 0 ovvero quando  $Z_1 Z_0$  coincide con 00, la macchina M è ritornata nello stato iniziale con le uscite a zero in attesa dell'ingresso (1,1) per riprendere eventualmente il conteggio suddetto. In figura è riportato un esempio.



- 2) Un calcolatore con frequenza di clock pari a 2,4GHz e CPI in assenza di fallimenti pari a 2 cicli di clock, dispone di una cache primaria e di una cache secondaria con le seguenti caratteristiche:
- Cache primaria di 512KB a corrispondenza diretta con blocchi di 4 parole. Frequenza di successo per le istruzioni pari al 96% e per i dati pari al 92%. Per l'accesso in caso di hit vengono impiegati 2 cicli di clock.
  - Cache secondaria di 128KB set-associative a 2 vie con blocchi di 4 parole. Frequenza di fallimento per le istruzioni pari al 5% e per i dati pari al 4%. Per l'accesso in caso di hit viene impiegato un ciclo di clock.
- La penalità di fallimento per l'accesso in memoria principale è di 12 cicli di clock. Supponendo si abbia l'architettura MIPS, determinare il tag espresso come numero esadecimale per l'indirizzo 0x7BCAEFE6 in relazione alle due cache e il tempo di CPU, supponendo che il 22% delle istruzioni I di un programma di test facciano accesso ai dati.
- 3) Scrivere una procedura in assembler MIPS che riceva in ingresso l'indirizzo di un vettore di interi ed il numero dei suoi elementi e per ogni elemento del vettore stesso effettui la seguente assegnazione:

```
if v[i] >= 512 then
    v[i] = s(v[i]) - t(v[i]) + 32
else
    v[i] = 1024 - [t(v[i]) - s(v[i])]
```

Dove con  $s()$  e  $t()$  si sono indicate due funzioni a disposizione del programmatore.