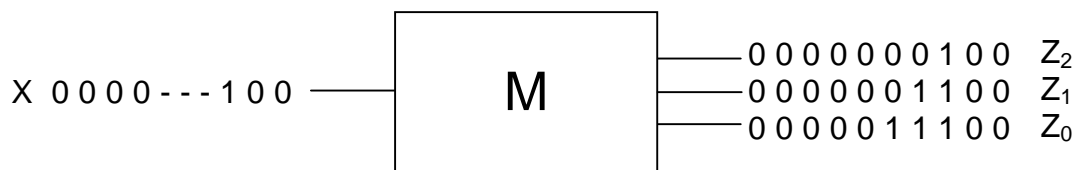


Corso di Calcolatori elettronici A

Compito scritto A – 2/7/2004

- 1) Progettare con un automa di Moore ed implementare con sole porte NAND, dopo aver eventualmente minimizzato il numero delle porte stesse, la macchina sequenziale M che, in corrispondenza di un 1 ricevuto sulla linea di ingresso X, produce sulle uscite Z_2 Z_1 Z_0 nei 4 cicli di clock successivi i bit (1, 1, 1), (0, 1, 1), (0, 0, 1) e (0, 0, 0) come mostrato nell'esempio in figura. Al termine le uscite rimangono a zero, finché non si ripresenta un 1 all'ingresso.



- 2) Un programma in assembler MIPS legge in sequenza i 10 elementi di un vettore di parole che inizia all'indirizzo 0x10000A00. Il computer che esegue il codice dispone di una cache per i dati di 64 byte, set-associative a 2 vie e con blocchi di 2 parole.

Supponendo la cache inizialmente vuota:

- Specificare per ognuna delle 10 letture se si tratta di un hit o un miss.
 - Per ogni accesso rappresentare graficamente lo stato della cache evidenziando i tag dei blocchi caricati.
 - Modificare poi la cache in modo da ridurre i miss del 50% circa e ripetere quindi l'esercizio per dimostrare tale riduzione.
 - Spiegare gli eventuali inconvenienti della cache modificata rispetto a quella iniziale.
- 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:

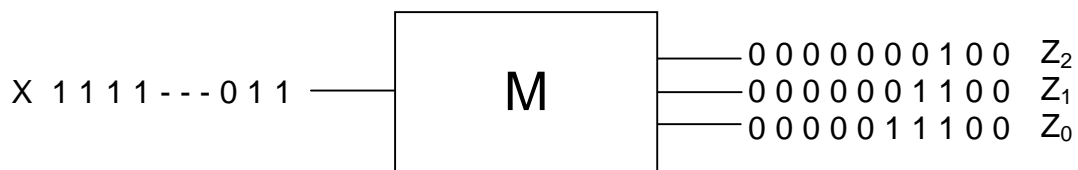
$$a[i] = 144.500 - a[i]^2$$

Con $a[i]$ si è indicato l'i-esimo elemento del vettore passato alla procedura.

Corso di Calcolatori elettronici A

Compito scritto B – 2/7/2004

- 1) Progettare con un automa di Moore ed implementare con sole porte NAND, dopo aver eventualmente minimizzato il numero delle porte stesse, la macchina sequenziale M che, in corrispondenza di uno 0 ricevuto sulla linea di ingresso X, produce sulle uscite Z_2 Z_1 Z_0 nei 4 cicli di clock successivi i bit (1, 1, 1), (0, 1, 1), (0, 0, 1) e (0, 0, 0) come mostrato nell'esempio in figura. Al termine le uscite rimangono a zero, finché non si ripresenta uno 0 all'ingresso.



- 2) Un programma in assembler MIPS legge in sequenza i 10 elementi di un vettore di parole che inizia all'indirizzo 0x10000C00. Il computer che esegue il codice dispone di una cache per i dati di 64 byte, set-associative a 2 vie e con blocchi di 2 parole.

Supponendo la cache inizialmente vuota:

- Specificare per ognuna delle 10 letture se si tratta di un hit o un miss.
 - Per ogni accesso rappresentare graficamente lo stato della cache evidenziando i tag dei blocchi caricati.
 - Modificare poi la cache in modo da ridurre i miss del 50% circa e ripetere quindi l'esercizio per dimostrare tale riduzione.
 - Spiegare gli eventuali inconvenienti della cache modificata rispetto a quella iniziale.
- 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:

$$a[i] = 137.800 - a[i]^2$$

Con $a[i]$ si è indicato l'i-esimo elemento del vettore passato alla procedura.