Esercizi con l'encoder

Esercizio 1

Sia dato un encoder ottico, connesso ad una macchina rotativa con velocità massima di 12000 giri al minuto, che trasmette su un bus digitale 4 bit in parallelo: 3 bit rappresentano con codifica Gray 8 settori ognuno di 45 gradi (vedi figura). 1 bit è il bit di parità (pari).

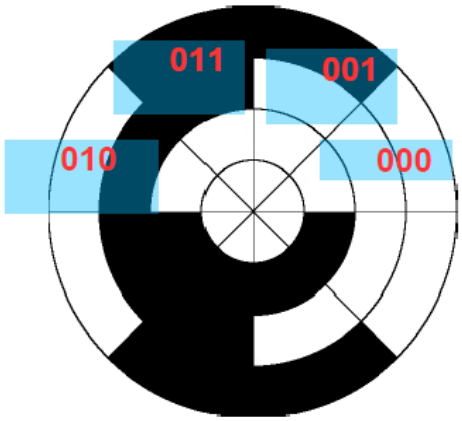
Nell'esercizio si chiede di progettare, lato ricevitore, una unità digitale che effettua le seguenti operazioni:

- metta a 1 l'uscita *data valid* ogni qual volta il controllo di parità sul bus è corretto (*data valid* vale 0 altrimenti).

- mette a 1 una uscita *dir* se il senso di rotazione della macchina è antiorario, altrimenti mette a 0 l'uscita *dir.*

- conta il numero di giri completi compiuti dalla macchina rotativa e salva il valore in un registro a 8 bit.

E' possibile utilizzare per la progettazione della macchina sia un approccio asincrono che sincrono.



Esercizio 1

Un sistema di misura tramite encoder ottico della posizione/velocità di rotazione di un motore elettrico trasmette su un bus digitale 4 bit in parallelo: 3 bit rappresentano con codifica Gray 8 settori ognuno di 45 gradi (vedi figura). 1 bit è il bit di parità (pari).

Si progetti una unità digitale (dotata di segnali di reset ed enable) sincrona che effettua le seguenti operazioni:

- ri-campiona i segnali a 4 bit sul proprio dominio di clock

- lavorando sui segnali ri-campionati, mette a 0 una uscita *data valid* ogni qual volta rileva un errore di parità sul bus (*data valid* vale 1 altrimenti) e memorizza in un registro A a 16 bit il numero degli errori rilevati.

- lavorando sui segnali ri-campionati, conta il numero di giri completi compiuti dal motore e salva il valore in un registro B a 16 bit.

Esercizio 1

Sia dato un encoder ottico, connesso ad una macchina rotativa con velocità da 0 a 6000 giri al minuto, che trasmette su un bus digitale 4 bit in parallelo: 3 bit rappresentano con codifica Gray 8 settori ognuno di 45 gradi (vedi figura). 1 bit è il bit di parità (dispari).

Nell'esercizio si chiede di progettare, lato ricevitore, una unità digitale che effettua le seguenti operazioni:

- metta a 1 l'uscita *data valid* ogni qual volta il controllo di parità sul bus è corretto (*data valid* vale 0 altrimenti).

- conta il numero di giri completi compiuti dal motore e salva il valore in un registro a 8 bit.

- mette a 1 una uscita *dir* se il senso di rotazione della macchina è orario, altrimenti mette a 0 l'uscita *dir.*

E' possibile utilizzare per la progettazione della macchina sia un approccio asincrono che sincrono.

Esercizi su trasmissioni in bus digitali

Esercizio 1

Su un bus digitale a n+1 bit, ad ogni ciclo di clock sys_clk, viene trasmesso in modo casuale una cifra codificata BCD su n bit più un bit di parità (dispari). Si progetti una unità digitale (dotata di segnali di reset ed enable) che permette di monitorare il bus e :

- mette a 1 una uscita *err* ogni qual volta rileva errori sul bus (es. codifica su n bit di una cifra BCD non valida e/o errore sul bit di parità) e memorizza in un registro A a 16 bit il numero totale delle cifre ricevute non correttamente. L'uscita *err* vale 0 se invece la cifra ricevuta è ritenuta corretta.

- solo nel caso in cui *err*=0, l'unità mette a 1 una uscita *cod* (per un solo ciclo di clock) ogni qual volta rileva che si è presentata sul bus la seguente sequenza di 4 cifre 1, 3, 5, 7 e salva in un registro B a 8 bit il numero totale delle volte che si è presentata sul bus tale sequenza.

Esercizio 1

Su un bus digitale a n+1 bit, ad ogni ciclo di clock sys_clk, viene trasmesso in modo casuale una cifra codificata BCD su n bit più un bit di parità (pari). Si progetti una unità digitale (dotata di segnali di reset ed enable) che permette di monitorare il bus e :

- mette a 1 una uscita *err* ogni qual volta rileva errori sul bus (es. codifica su n bit di una cifra BCD non valida e/o errore sul bit di parità) e memorizza in un registro A a 32 bit il numero totale delle cifre ricevute non correttamente. L'uscita *err* vale 0 se invece la cifra ricevuta è ritenuta corretta.

- mette a 1 una uscita *fib* (per un solo ciclo di clock) ogni qual volta rileva che si è presentata sul bus la seguente sequenza di 7 cifre 0, 1, 1, 2, 3, 5, 8 e salva in un registro B a 8 bit il numero totale delle volte che si è presentata sul bus tale sequenza.

Esercizio 1

Su un bus digitale, ad ogni ciclo di clock, vengono trasmessi in parallelo una cifra in codice GRAY a 3 bit più un bit di parità (dispari). Si progetti una unità digitale (dotata di segnali di reset ed enable) che permette di monitorare il bus e:

- mette a 1 una uscita *err_c* ogni qual volta rileva errori sul bus dovuti a una codifica non corretta di una cifra in codice GRAY; il numero totale delle cifre ricevute non correttamente viene memorizzato in un registro A a 16 bit

- mette a 1 una uscita *err_p* ogni qual volta rileva errori di parità sul bus e memorizza in un registro B a 16 bit il numero totale degli errori di parità rivelati.

Le uscita *err_c ed err_p* valgono 0 in assenza di errori.

Esercizio 1

Su un bus digitale, ad ogni ciclo di clock, vengono trasmessi in parallelo una cifra in codice GRAY a 3 bit più un bit di parità (dispari). Si progetti una unità digitale (dotata di segnali di reset ed enable) che permette di monitorare il bus e:

A) Mette a 1 una uscita *err_p* ogni qual volta rileva errori di parità sul bus e memorizza in un registro A a 16 bit il numero totale degli errori di parità rivelati. L' uscita *err_p* vale 0 in assenza di errori.

B) Nel caso in cui dal controllo di parità non risultano errori, l'unità digitale mette a 1 una uscita *err_c* ogni qual volta rileva errori sul bus dovuti a una codifica non corretta di una cifra in codice GRAY. L'uscita *err_c* vale 0 in assenza di errori. Il numero totale delle cifre ricevute non correttamente viene memorizzato in un registro B a 16 bit

Esercizio 1

Su un bus digitale, in modo sincrono rispetto ai fronti in salita di un clock *sys-clk*, viene trasmessa una sequenza di simboli a 4 bit in cui i primi 3 bit rappresentano una sequenza crescente di cifre in codifica GRAY mentre il quarto bit è un bit di parità (pari). Si progetti una unità digitale (con reset e enable), sincrona con *sys-clk*, che permette di monitorare il bus e:

a) Mette a 1 una uscita *err_p* ogni qual volta rileva errori di parità sul bus e memorizza in un registro X a 8 bit il numero totale degli errori di parità rivelati. L'uscita *err_p* vale 0 in assenza di errori.

b) Nel caso in cui dal controllo di parità non risultano errori, l'unità digitale mette a 1 una uscita *err_c* ogni qual volta rileva errori sul bus dovuti a una codifica non corretta (sequenza di cifre in codice GRAY con ordine crescente). L'uscita *err_c* vale 0 in assenza di errori. Il numero totale delle cifre ricevute non correttamente viene memorizzato in un registro Y a 8 bit

Esercizi con rivelatori di eventi

Esercizio 1

Un rivelatore di eventi, che lavora con un clock di 100 MHz, emette un impulso largo 5 cicli di clock per ogni evento rivelato. Considerando che la distanza temporale minima tra due eventi successivi è di 100 ns, si realizzi una macchina sincrona in grado di calcolare il numero di eventi rilevati modulo 64. Si presti attenzione a non contare più volte lo stesso evento. La macchina sia dotata di segnale di abilitazione attivo alto (sincrono) e reset asincrono.

Esercizio 1

Un detector per esperimenti di fisica nucleare genera, in modo sincrono con un clock di sistema sys_clk=10 MHz, una sequenza di 4 bit 1001, ogni qual volta viene rivelato un evento. Il segnale che viene trasmesso è a 5 bit in quanto in coda ai 4 bit viene trasmesso un bit di parità P (il numero complessivo di bit trasmessi uguali a 1 sia pari). Due eventi successivi sono temporalmente distanti almeno un tempo T > 5*T sys_clk.
Se l'evento non viene rivelato il detector trasmette, sempre in modo sincrono con sys_clk, 1 se è pronto alla rivelazione, 0 se invece è in stand-by.
Progettare lato ricevitore una macchina a stati sincrona, dotata di segnali di start ed enable, che registra modulo 128 il numero di eventi rivelati.

Esercizio 1

Un detector per esperimenti di fisica nucleare genera, in modo sincrono con un clock di sistema sys_clk=10 MHz, una sequenza di 4 bit 1001, ogni qual volta viene rivelato un evento. Il segnale che viene trasmesso è a 5 bit in quanto in coda ai 4 bit viene trasmesso un bit di parità P (il numero complessivo di bit trasmessi uguali a 1 sia dispari). Due eventi successivi sono temporalmente distanti almeno un tempo T > 5*T sys_clk.
Se l'evento non viene rivelato il detector trasmette 0, sempre in modo sincrono con sys_clk.
Progettare lato ricevitore una macchina a stati sincrona, dotata di segnali di enable e reset, che registra modulo 128 il numero di eventi rivelati.

Progetto di unità digitali

Esercizio 1

Progettare una unità digitale con segnali generali di clock, reset asincrono e Start (sincrono con il clock) che:
A) Allo start scandisce, in modo sequenziale dall'indirizzo 0000000000 a quello 1111111111 una memoria con locazioni a 4 bit; ogni locazione contiene una cifra di cui il LSB (Last Significant Bit) è il bit di segno e i 3 bit rimanenti sono il modulo in codifica Gray. Nota: Se la macchina viene resettata riparte dall'indirizzo 0000000000; se invece Start viene messo a 0 e poi rimesso a 1 riparte dall'ultima locazione scandita.
B) Verifica che il contenuto di locazioni successive della memoria sia coerente con la codifica Gray e

conta il numero di errori eventualmente presenti (salvati in un registro Gray_err, di cui va determinata la dimensione in bit necessaria)
C) Su una uscita seriale, che di default è settata a 1, per ogni cifra letta genera un flusso di bit così composto: 1o bit di sincronizzazione vale 0, dal 2o al 4o bit indicano il modulo codificato GRAY, il 5o bit è il bit di parità pari.

Esercizio 1
Progettare una unità digitale che riceve su un ingresso seriale, in modo sincrono con un clock Iclk, cifre decimali con segno codificate su 6 bit: dal 1o al 4o bit viene indicato il modulo codificato BCD, il 5o bit indica il segno, il 6o bit codifica la parità (dispari). La macchina sincrona effettua la media mobile sulle ultime 10 cifre ricevute (se il bit di parità corrispondente ad una cifra è errato la cifra è esclusa dal conto) e:

- con cadenza Oclk mette a 1 una uscita POS se la media è positiva, mette a 1 una uscita NULL se la media è nulla, mette a 1 una uscita ERR se è stato rivelato un errore di parità (altrimenti le uscite valgono 0)

Che relazione ci deve essere tra Iclock ed Oclock affinchè il sistema funzioni correttamente?

Esercizio 1
Progettare una unità digitale che riceve su un ingresso seriale, con cadenza IN_clock, cifre decimali con segno codificate su 6 bit in questo modo: 1o bit indica il segno, dal 2o al 5o bit indicano il modulo codificato BCD, la 6o è il bit di parità pari. La macchina sincrona effettua la media mobile sulle ultime 8 cifre ricevute (se il bit di parità corrispondente ad una cifra è scorretto la cifra viene esclusa dal computo) e con cadenza OUT_clock:
- mette a 1 una uscita NEG se la media è negativa, (altrimenti tale uscita vale zero),
- mette a 1 una uscita NULL se la media è nulla
- mette a 1 una uscita ERR se è stato rivelato un errore di parità
Che relazione ci deve essere tra IN_clock ed OUT_clock affinchè il sistema funzioni correttamente?

Esercizio 1
Progettare un'unità digitale che riceve su un ingresso seriale, con cadenza IN_clock, cifre decimali con segno codificate su 6 bit in questo modo: 1o bit indica il segno, dal 2o al 5o bit indicano il modulo codificato BCD, la 6o è il bit di parità (dispari). La macchina effettua la media mobile sulle ultime 4 cifre ricevute (se il bit di parità corrispondente ad una cifra è scorretto la cifra viene esclusa dal computo) e con cadenza OUT_clock:
- mette a 1 una uscita POS se la media è positiva, (altrimenti tale uscita vale zero),
- mette a 1 una uscita NULL se la media è nulla
- mette a 1 una uscita ERR se è stato rivelato un errore di parità.
La macchina deve essere dotata di segnali di Reset asincrono e Enable sincrono con IN_clock.
Che relazione ci deve essere tra IN_clock ed OUT_clock affinché il sistema funzioni correttamente?

Esercizio 1
Progettare una unità digitale con segnali generali di clock, reset asincrono e Start (sincrono con il clock) che:
A) Allo start scandisce, in modo sequenziale dall'indirizzo 0000000000 a quello 1111111111 una memoria con locazioni a 4 bit; ogni locazione contiene una cifra di cui il MSB è il bit di segno (MSB) e i 3 bit rimanenti sono il modulo in codifica Gray. (se la macchina viene resettata riparte da indirizzo 0000000000; se invece Start viene messo a 0 e poi rimesso a 1 riparte dall'ultima locazione scandita)

B) Verifica che il contenuto di locazioni successive della memoria sia coerente con codifica Gray e conta il numero di errori eventualmente presenti (salvati in un registro ERR)
C) Su una uscita seriale, che di default vale 1, per ogni cifra letta genera un flusso di bit così composto: 1o bit di sincronizzazione vale 0, 2o bit indica il segno, dal 3o al 5o bit indicano il modulo codificato GRAY, il 6o bit è il bit di parità dispari.

Esercizi con il sistema "pick&place"

Esercizio 1

Un sistema “pick and Place”, per l’assemblaggio di schede elettroniche, verifica il corretto collocamento di 4 dispositivi su 8 piazzole numerate da 0 a 7. Il sistema di verifica genera in modo sincrono con un clock di sistema s-clk un segnale su 1 byte in cui i bit sono tutti a “1” nel caso di corretto posizionamento, i primi 4 bit sono a "0" e gli ultimi 4 bit sono a "1" nel caso di errato posizionamento, tutti e gli 8 bit sono a "0" se non sono presenti schede da assemblare. Il segnale trasmesso è di 10 bit, in quanto in testa al byte viene trasmesso un bit a 1 di START ed in coda al byte viene aggiunto un bit di parità P (pari). Il segnale così generato viene inviato a distanza su una linea seriale con clock di sistema s-clk. Due verifiche successive sono temporalmente distanti almeno un tempo T > 30*T_s-clk.
Progettare lato ricevitore una macchina a stati sincrona, dotata di segnali di enable sincrono e reset asincrono, che conta, modulo 1024, sia il numero di verifiche di piazzamento correttamente rilevate (salva risultato in registro X) sia gli errori commessi dovuti a errore di parità e/o errore di piazzamento (salva risultato in registro Y).

Esercizio 1

Un sistema “pick and Place”, per l’assemblaggio di schede elettroniche, verifica, con un sistema visivo, il corretto collocamento di tre dispositivi su cinque piazzole. I dispositivi devono occupare le piazzole 1, 3 e 5, il sistema di verifica fornisce un segnale su cinque bit nel quale c’è un “1” se la piazzola è occupata dal dispositivo e uno “0” se la piazzola è libera. Il segnale è di 6 bit, in quanto in coda ai 5 bit viene aggiunto un bit di parità P (il numero complessivo di bit trasmessi uguali a 1 è pari). Il segnale così generato viene inviato a distanza su una linea seriale con un clock di sistema sys_clk=1 MHz. Due verifiche successive sono temporalmente distanti almeno un tempo T > 100*T sys_clk. Quando non sono presenti schede sotto il sistema di controllo trasmette 0, sempre in modo sincrono con sys_clk.
Progettare lato ricevitore una macchina a stati sincrona, dotata di segnali di enable e reset, che registra, modulo 256, sia gli errori di bit di parità in un registro A che il numero di verifiche di piazzamento correttamente rilevate in un registro B.

Esercizio 1

Un sistema “pick and Place”, per l’assemblaggio di schede elettroniche, verifica, con un sistema visivo, il corretto collocamento di 4 dispositivi su 8 piazzole numerate da 1 a 8. I dispositivi DEVONO occupare le piazzole dispari. Il sistema di verifica fornisce in modo sincrono con un clock di sistema sys_clk un segnale su 1 byte in cui i bit sono a “1” per le piazzole occupate da dispositivi e a “0” per le piazzole libere. Quando non sono presenti schede da assemblare il sistema di controllo genera bit pari a "0", sempre in modo sincrono con sys_clk. Il segnale trasmesso è di 10 bit, in quanto in testa al byte viene trasmesso un bit a 1 di START ed in coda al byte viene aggiunto un bit di parità P (il numero complessivo di bit trasmessi uguali a 1 è pari). Il segnale così generato viene inviato a distanza su una linea seriale con clock di sistema sys_clk. Due verifiche successive sono temporalmente distanti almeno un tempo T > 50*T sys_clk.

Progettare lato ricevitore una macchina a stati sincrona, dotata di segnali di enable sincrono e reset asincrono, che conta, modulo 1024, sia il numero di verifiche di piazzamento correttamente rilevate (salva risultato in registro A) sia gli errori commessi dovuti a errore di parità e/o errore di piazzamento (salva risultato in registro B).

Esercizio 1

Un sistema “pick and Place”, per l’assemblaggio di schede elettroniche, verifica, con un sistema visivo, il corretto collocamento di 4 dispositivi su 8 piazzole numerate da 1 a 8. I dispositivi DEVONO occupare le piazzole pari. Il sistema di verifica fornisce in modo sincrono con un clock di sistema sys_clk un segnale su 1 byte in cui i bit sono a “1” per le piazzole occupate da dispositivi e a “0” per le piazzole libere. Quando non sono presenti schede da assemblare il sistema di controllo genera bit pari a "0", sempre in modo sincrono con sys_clk. Il segnale trasmesso è di 10 bit, in quanto in testa al byte viene trasmesso un bit a 1 di START ed in coda al byte viene aggiunto un bit di parità P (il numero complessivo di bit trasmessi uguali a 1 è dispari). Il segnale così generato viene inviato a distanza su una linea seriale con clock di sistema sys_clk. Due verifiche successive sono temporalmente distanti almeno un tempo T > 100*T sys_clk.
Progettare lato ricevitore una macchina a stati sincrona, dotata di segnali di enable e reset, che registra, modulo 256, sia gli errori di bit di parità in un registro A che il numero di verifiche di piazzamento correttamente rilevate in un registro B.

Esercizio 1

Un sistema “pick and Place”, per l’assemblaggio di schede elettroniche, verifica, con un sistema visivo, il corretto collocamento di 4 dispositivi su 8 piazzole numerate da 1 a 8. I dispositivi DEVONO occupare le piazzole pari. Il sistema di verifica fornisce in modo sincrono con un clock di sistema sys_clk un segnale su 1 byte in cui i bit sono a “1” per le piazzole occupate da dispositivi e a “0” per le piazzole libere. Quando non sono presenti schede da assemblare il sistema di controllo genera bit pari a "0", sempre in modo sincrono con sys_clk. Il segnale trasmesso è di 10 bit, in quanto in testa al byte viene trasmesso un bit a 1 di START ed in coda al byte viene aggiunto un bit di parità P (il numero complessivo di bit trasmessi uguali a 1 è dispari). Il segnale così generato viene inviato a distanza su una linea seriale con clock di sistema sys_clk. Due verifiche successive sono temporalmente distanti almeno un tempo T > 100*T sys_clk.
Progettare lato ricevitore una macchina a stati sincrona, dotata di segnali di enable e reset, che registra, modulo 256, sia gli errori di bit di parità in un registro A che il numero di verifiche di piazzamento correttamente rilevate in un registro B.

Altri sistemi digitali

Esercizio 1

Un sensore di allarme genera, in modo sincrono con un clock di sistema sys_clk=20 MHz, una sequenza di 4 bit 1001 (in modo seriale), ogni qual volta viene rivelata una intrusione. Il segnale che viene trasmesso è a 5 bit in quanto in coda ai 4 bit viene trasmesso un bit di parità P (dispari). Due eventi successivi sono temporalmente distanti almeno un tempo T > 1 ms.

Se l'evento non viene rivelato il detector trasmette 0, sempre in modo sincrono con sys_clk.

Progettare lato ricevitore una macchina a stati sincrona, dotata di segnali di start ed enable, che:

- Mette a 1 l'uscita False_ALL se rileva l'arrivo della sequenza 1001 ma la parità non è corretta (False_ALL vale 0 altrimenti). Conta il numero totale di falsi allarmi ricevuti e li memorizza, modulo 356, nel registro A

- Mette a 1 l'uscita True_ALL se rileva l'arrivo della sequenza 1001 e la parità è corretta (True_ALL vale 0 altrimenti). Conta il numero totale di allarmi veri ricevuti e li memorizza, modulo 356, nel registro B

Esercizio 1

Progettare una macchina sincrona dotata di Clock, Reset e Enable che:

A) Se Enable=1 scandisce, dall'indirizzo 0000000000 a quello 1111111111 una memoria con locazioni a 4 bit; ogni locazione contiene una cifra di cui il MSB è il bit di segno e i 3 bit rimanenti sono il modulo in codifica Gray. (al reset la macchina parte dall'indirizzo 0000000000; se durante la scansione Enable viene messo a 0 e poi rimesso a 1 la macchina riparte dall'ultima locazione scandita)

B) Verifica che il contenuto di locazioni successive della memoria sia coerente con la codifica Gray e conta il numero di errori eventualmente presenti (questo numero è salvato in un registro ERR a 8 bit)

C) Su una uscita seriale, che di default vale 0, per ogni cifra letta genera un flusso di bit così composto: 1o bit di Start vale 1, 2o bit indica il segno, 3o bit vale 0 se la cifra è multipla di 2 altrimenti vale 1, il 4o bit è un bit di parità PARI.