

Esercizi sulla strutturazione firmware

1) Una unità di elaborazione U contiene un componente logico memoria M di capacità $N = 16K$ interi non negativi di 32 bit. Riceve da U1 messaggi (A, IND), con A intero non negativo di 32 bit e IND di 14 bit, ed invia ad U2 messaggi (OUT) di 15 bit.

Una *prima versione* di U è espressa dal seguente microprogramma, dove *zero* ($M[IND] - A$) funge da codice operativo di due operazioni esterne equiprobabili, e la funzione booleana $f(x)$, con x intero non negativo, è vera se e solo se x è una potenza di due ≥ 256 Mega:

0. (RDY1, zero($M[IND] - A$) = 0 -) nop, 0;
(= 1 1) reset RDY1, set ACK1, $0 \rightarrow I, 0 \rightarrow C, 1$;
(= 1 0) reset RDY1, set ACK1, $A \rightarrow M[IND], 0$
1. (I_0 , ACK2 = 0 -) $f(M[I_m]) \rightarrow H, 2$;
(= 1 0) nop, 1;
(= 1 1) $C \rightarrow OUT$, set RDY2, reset ACK2, 0;
2. ($H = 0$) $I + 1 \rightarrow I, 1$;
(= 1) $I + 1 \rightarrow I, C + 1 \rightarrow C, 1$

Si chiede una *seconda versione* di U tale che, esprimendo il tempo medio di elaborazione della prima versione come $T_1 \sim p N$, la seconda versione abbia un tempo medio di elaborazione $T_2 \sim q N$, con $q < p$.

Detto t_p il ritardo di una porta logica con al massimo 8 ingressi, le ALU e la memoria M hanno ritardo di $5t_p$.

Scrivere il microprogramma della seconda versione, realizzare $f(x)$ e valutare le costanti p e q . Spiegare chiaramente come si è ragionato per definire la seconda versione.

2) Una unità di elaborazione U

- contiene un componente logico memoria M di capacità 64 parole ognuna di 32 bit;
- riceve in ingresso, da una stessa unità U_M , messaggi (A, B, C, J) con A e B parole di 32 bit, C di 64 bit, e J di 4 bit. I valori di C hanno sempre la seguente caratteristica: uno ed un solo bit vale 1 e tutti gli altri 0; nel seguito con $P(C)$ si indicherà la posizione del bit uguale a 1 in C;
- invia in uscita messaggi di una parola ad una delle 16 unità U_0, \dots, U_{15} .

Il funzionamento è il seguente:

- se $A < B$: scrive nella locazione di M di indirizzo uguale a $P(C)$ il contenuto della stessa locazione incrementato di 1, ed invia il valore scritto all'unità U_J ;
- se $A = B$: scrive nella locazione di M di indirizzo uguale a $P(C)$ il contenuto della stessa locazione incrementato di 1;
- se $A > B$: invia il contenuto della locazione di M di indirizzo uguale a $P(C)$ incrementato di 1 all'unità U_J .

È noto il ritardo t_p di una porta logica con al più 8 ingressi. Il ritardo di una ALU vale $5t_p$.

Spiegare e mostrare in dettaglio la struttura del componente logico memoria M in modo da utilizzare direttamente il valore di C per l'indirizzamento.

Scrivere e spiegare il microprogramma e valutare il tempo di elaborazione di U, in funzione di t_p , rispettando entrambi i seguenti requisiti:

- minimizzare il numero di variabili di condizionamento,
- minimizzare il numero di cicli di clock.

Mostrare la struttura della Parte Operativa.

3) Una unità di elaborazione U

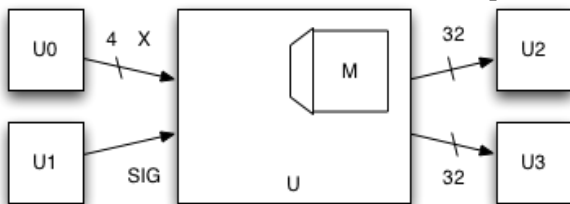
- contiene 256 registri di una parola, considerati appartenere a due insiemi disgiunti denominati $A0, \dots, A127$ e $B0, \dots, B127$. I registri del primo insieme sono in sola lettura e tutti di valore diverso, quelli del secondo modificabili;
- riceve in ingresso messaggi (OP, X, Y, I) , con OP di 2 bit, X e Y di una parola, e I di 7 bit, e può inviare messaggi $(OUT, ESITO)$ con OUT di una parola ed $ESITO$ di 1 bit;
- se $OP = 00$, scrive il valore Y nel registro Bh , con $h = I$;
- se $OP = 01$, invia su OUT il risultato di $(Bh + Y) \text{ div } 128$, con $h = I$, e su $ESITO$ il valore 1;
- se $OP = 1-$, invia su OUT il risultato di $(Bj + Y) \text{ mod } 128$, con j tale che $Aj = X$, e su $ESITO$ il valore 1. Se non esiste alcun j tale che $Aj = X$, invia su $ESITO$ il valore 0.

Fornendo adeguate spiegazioni, progettare U in modo che il tempo di elaborazione sia uguale ad *un ciclo di clock*, e valutare il ciclo di clock in funzione del ritardo di stabilizzazione t_p di una porta logica con al più 8 ingressi, considerando che una ALU ha tempo di stabilizzazione $5t_p$.

Di tutte le reti combinatorie utilizzate, diverse da una ALU, deve essere esplicitamente data la definizione, l'espressione logica ed il ritardo di stabilizzazione.

4) Un'unità U è interfacciata ad una unità U0 che le fornisce un valore X su 4 bit, ad una unità U1, che le fornisce un segnale di pura sincronizzazione, e a due unità U2 ed U3 cui invia parole di 32 bit.

L'unità U contiene una memoria M con capacità 1K parole da 32 bit.



Per ogni X ricevuto da U0, U conta quante sono le celle di M, con indirizzo i tale che $i\%16 = X$, che contengono un numero pari e quante quelle che contengono un numero dispari. I risultati (numero_pari e numero_dispari) vengono inviati rispettivamente ad U2 ed U3.

Ogni qualvolta viene rilevato il segnale SIG, il calcolo corrente viene immediatamente *abortito*, *senza inviare risultati in uscita* e, in seguito alla *prossima* ricezione di un nuovo X, l'invio dei risultati verso U2 ed U3 viene invertito: il numero_pari, precedentemente inviato ad U2 (U3), viene inviato ad U3 (U2), e il numero_dispari, precedentemente inviato ad U3 (U2), viene inviato ad U2 (U3).

La memoria M ha un tempo di accesso pari a $10t_p$, con t_p ritardo di stabilizzazione di una porta logica con al massimo otto ingressi. Una ALU a 32 bit opera in $5t_p$ e gli eventuali overflow non vanno considerati.

Fornendo adeguate spiegazioni, è richiesto il microprogramma dell'unità U e la valutazione della sua banda di elaborazione per $t_p = 10^{-2}$ nsec, assumendo trascurabile la probabilità che si verifichi un segnale SIG.

5) Una unità di elaborazione U ha al suo interno due memorie, A e B, di capacità $N = 16K$ parole, dove la parola è di 32 bit.

U riceve dall'unità U₀ valori OP di 1 bit, dall'unità U₁ coppie (IND, H), dove IND è un indirizzo delle memorie e H è un valore di 5 bit. U invia all'unità U₂ valori Z0 di 1 bit, e ad U₃ valori Z1 rappresentati su un numero di bit opportuno.

Le operazioni esterne sono definite come segue:

- *prima operazione esterna* ($OP = 0$): considerando le due locazioni di A e B indirizzate da IND , se esse hanno il bit H -esimo uguale: copiare la locazione di A nella locazione di B e inviare 1 a U_2 ; altrimenti non modificare le memorie e inviare 0 a U_2 ;
- *seconda operazione esterna* ($OP = 1$): considerando A e B suddivise logicamente in blocchi di 512 parole, per ogni locazione i ($i = 0 \dots 511$) del blocco H -esimo di A e di B :

$$\text{se } A[i] > B[i] \text{ eseguire } B[i] = A[i] + B[i]/4$$

Inviare a U_3 il numero di modifiche apportate al blocco di B .

È noto il ritardo t_p di una porta logica con al massimo 8 ingressi. Una ALU ha ritardo uguale a $5t_p$. Le memorie A e B sono realizzate partendo da componenti logici memoria aventi capacità 256 parole e tempo di accesso $3t_p$.

a) Con il vincolo che la prima operazione esterna sia eseguita in un singolo ciclo di clock, scrivere il microprogramma di U e valutarne il tempo medio di elaborazione nell'ipotesi che la probabilità della prima operazione esterna sia uguale a $1/4$ e quella della seconda uguale a $3/4$.

È necessario fornire opportune e sintetiche spiegazioni sulle scelte più significative effettuate.

b) Il sistema è modificato come segue: U_0 e U_3 sono eliminate, e U esegue solo la prima operazione esterna. Realizzare U come singola rete sequenziale esprimendone le funzioni che la definiscono, *spiegando sinteticamente come si è ragionato*.

c) Con riferimento al punto a):

- esprimere la funzione di transizione dello stato interno della Parte Operativa relativamente alla generica locazione della memoria B , e la funzione di transizione dello stato interno della Parte Controllo;
- spiegare se, con gli stessi vincoli e nelle stesse ipotesi, lo stesso microprogramma potrebbe essere eseguito in un tempo minore rispetto a quello valutato.

6) Progettare una unità di elaborazione U che implementa una pila capace di contenere fino a 1024 elementi, ognuno costituito da 16 interi. U

- riceve da una unità U_1 valori degli elementi da inserire nella pila; ogni elemento è ricevuto un intero alla volta,
- invia elementi estratti dalla pila ad una unità U_2 , che ne fa richiesta; U_2 è capace di ricevere due interi alla volta.

In caso di richieste contemporanee da U_1 e U_2 , U applica una disciplina di priorità pseudo-casuale.

7) Progettare una unità di memoria, di capacità massima 512M parole, capace di servire richieste da parte di 128 unità U_0, \dots, U_{127} . Le possibili operazioni richieste da una unità sono:

- lettura di una parola,
- scrittura di una parola,
- lettura di un blocco di 8 parole a indirizzi consecutivi,
- scrittura di un blocco di 8 parole a indirizzi consecutivi.

Nel caso di richieste contemporanee, viene applicata una disciplina a priorità fissa, secondo la quale tra U_i e U_j la priorità maggiore viene data all'unità U_i con $i < j$.

Inoltre, supponendo che:

- tutte le unità impiegate eccetto M abbiano ciclo di clock τ uguale,

- la tecnologia con cui è realizzata la memoria abbia $t_p = \tau$, ritardo di stabilizzazione di una porta logica con al più 4 ingressi,
- la latenza di trasmissione dei collegamenti esterni ad un'unità valga 5τ ,

calcolare il tempo di accesso in memoria come visto da una unità appartenente a U_0, \dots, U_{127} , in assenza di conflitti (cioè, supponendo che l'unità sia l'unica a fare richiesta di accesso).