

Esame 06-07-05

Un'interfaccia IFABS riceve in continuazione da un bus esterno XDATA(0..15) dati numerici paralleli a 16 bit in complemento a due, sincronizzati a un clock esterno XCLK. Su comando della CPU, IFABS procede all'acquisizione di un blocco da 1024 dati, dei quali calcola il massimo e il minimo dei valori assoluti:

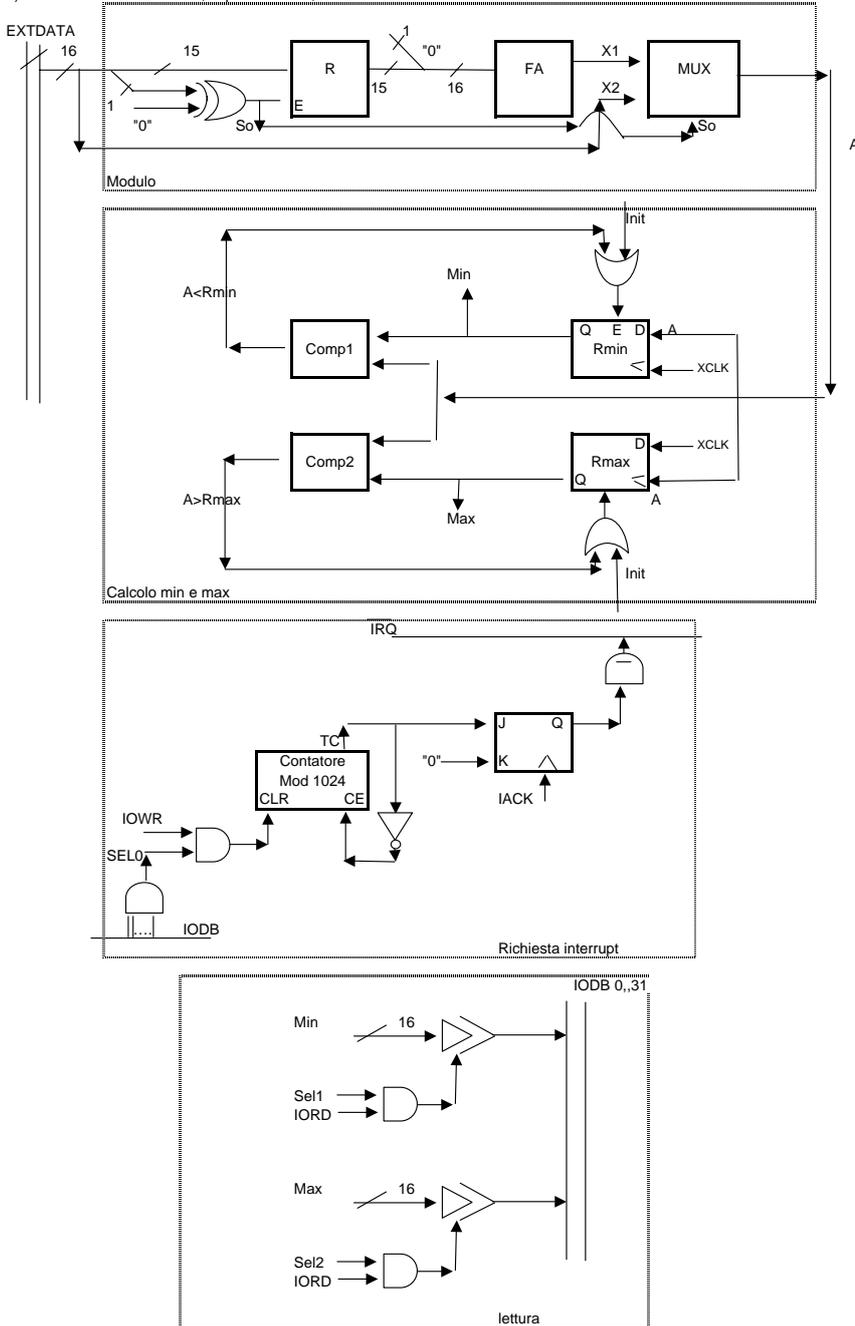
$$A = \max |x_i| \quad B = \min |x_i|$$

questi due risultati vengono quindi trasmessi alla CPU che calcola il valore

$$U = (A+B)/2$$

e aggiorna le seguenti locazioni di memoria:

- 1) AMIN con il minimo tra tutti i valori A acquisiti;
- 2) BMAX con il massimo tra tutti i valori B acquisiti
- 3) UMIN e UMAX col minimo e, rispettivamente, il massimo tra tutti i valori U calcolati.



Funzionamento:

Da Extdata entrano i 16 bit. Di questi 16 bit io devo calcolare il max e il min in modulo. Essendo le stringhe in complemento a due ho la necessità di calcolare il modulo dei numeri negativi entranti. Opero in questo modo: i 16 bit che entrano li mando sia direttamente all'ingresso di un mux e anche dentro un dispositivo "Modulo" che ha il compito di ricavarne il modulo. La prima cosa che questo dispositivo fa è prelevare il primo bit. Tale bit viene mandato all'ingresso di una XOR in confronto con un bit posto preventivamente a zero. In uscita avremo 1, solo se la stringa entrante è negativa (ovvero il bit prelevato vale 1). Se l'uscita della XOR vale 1 questo segnale andrà ad abilitare l'ENABLE del registro R asincrono. Il registro sarà caricato della stringa di 15 bit (cioè senza il primo bit che avevo prelevato). In uscita al registro inserirò un bit posto a zero in testa, in modo da ottenere di nuovo i 16 bit. A questi 16 bit in un Full Adder aggiungerò una stringa di 16 bit equivalenti all'1 binario. Con questo procedimento mi sono ricavato il modulo delle sequenze negative entranti.

Dimostro con un esempio a soli 3 bit il concetto:

000	0	Come si nota dall'esempio qui a sinistra, i numeri positivi sono già in modulo. I negativi invece hanno bisogno di una lavorazione. Sostituisco il
001	1	Primo bit con lo zero e poi aggiungo 1 ottenendo:
010	2	-1: 000+001=001 ovvero 1, cioè il modulo di -1
011	3	-2: 001+001=010 ovvero 2, cioè il modulo di -2
100	-1	
101	-2	Questo procedimento vale per tutti i numeri negativi.
110	-3	
111	-4	

Il segnale So che proviene dalla porta XOR lo uso come segnale di abilitazione per il MUX. Il MUX ha il compito di far passare il segnale entrante diretto se questo era già in modulo, oppure quello che è stato convertito in modulo. Ovviamente la decisione la prende So. Infatti S0 vale 1 solo se la stringa entrante era negativa. Entra dunque X1 se So vale 1 oppure entra X2. La stringa di 16 bit uscente da questo blocco "modulo" va nel blocco che calcola il min e il max.

La stringa entrante che chiamo A andrà direttamente ai comparatori (ne ho usati due perché ho supposto che tutto il calcolo avvenga in un solo colpo di XCLK) e anche in entrata ai due registri Rmax e Rmin. Tuttavia i registri non la caricheranno se l'ingresso Enable non è attivo. All'inizio del procedimento, un segnale di init proveniente dalla CPU abiliterà enable, quindi la prima stringa che arriva verrà caricata sia in Rmax che in Rmin. Al prossimo colpo di XCLK le stringhe presenti nei due registri verranno proposte all'ingresso del comparatore contemporaneamente alla nuova stringa. Il comparatore opera il controllo e abilita l'uscita A<Rmin se la stringa entrante è minore in modulo di quella presente nel registro Rmin. Questo segnale va a pilotare la porta or che a sua volta pilota l'Enable del registro. Essendo ad 1, uno dei due bit d'ingresso alla OR, l'uscita vale 1 e allora Rmin viene abilitato al caricamento della stringa che si trova in attesa al suo ingresso. In questo modo ho aggiornato (se richiesto) il registro Rmin. Il procedimento è analogo con Rmax.

Infine ci sono i due blocchi: "richiesta di interrupt" e "lettura".

Il primo ha il compito di contare le 1024 sequenze di 16 bit entranti. Il contatore inizierà a contare quando la CPU lo abilita e fa richiesta di interrupt quando TC arriva ad 1. TC va ad 1 solo dopo aver contato da 0 a 1023, essendo un contatore mod.1024. A quel punto attenderà che la CPU conceda il bus. Quando la CPU concede il bus abiliterà anche una tra le due periferiche che devono scrivere (ovvero il min o il max), settando ad 1 o SEL1 o SEL2. Le periferiche scriveranno sul bus e il procedimento è finito.

Gentile professore,

Le chiedo di controllare il procedimento che ho usato e possibilmente correggerlo.

Avrei anche un paio di domande.

- 1) ad un certo punto il testo dice che la CPU calcola U e aggiorna delle locazioni di memorie AMIN, BMAX, UMIN e Umax. Queste azioni mi sembrano molto semplici da implementare, ma quello che volevo chiedere è se devo farle io. Il fatto che il testo dica: "La CPU calcola...." mi ha portato a pensare che non devo occuparmene.
  - 2) Nel mio tentativo di risolvere il problema mi sono accorto di aver bisogno della collaborazione della CPU, la quale deve inviarmi 2 segnali. Uno di init per inizializzare i due registri Rmin e Rmax e anche di un segnale che in qualche modo mi resettasse il contatore, ovvero un clear.
- Penso che questo dovrebbe essere il compito dello SCO. Potrebbe, anche in linea di massima, dirmi come faccio a progettare?