REGOLE RTL

lunedì 1 giugno 2020 16:24

1) Posso copiare il valore di un registro in un altro e allo stesso tempo scrivere su questo registro
   es. A -> B, INC(A); //sposto il valore di A in B e allo stesso tempo modifico il valore in A
2) I segnali beta vengono utilizzati nelle condizioni di if
   I segnali beta possono essere:
   1) bit del supero di capacità (bisogna avere un flip flop in cui, in una somma, viene salvato il bit di supero e con segnale di ingresso z che indica l'azzeramento se serve azzerarlo)
3) L'iterazione in RTL viene utilizzata tramite un registro a decremento/incremento che tiene in considerazione del numero di cicli da effettuare e che diventa, tramite il segnale beta uscente da tale registro, la condizione da verificare per interrompere un ciclo (o per farlo continuare attraverso il goto e le etichette).
   In questo caso un passaggio è rilegato al caricare il valore nel registro
4) Se devo scambiare i valori del registro X in X+1 basta utilizzare X come registro a incremento e decremento e introdurre un registro di "salvataggio"
5) Il segnale alpha di controllo con lettera k indica se caricare dall'esterno (k=0) o fare l'operazione (k=1) in un registro funzione.
   Il segnale alpha di controllo con lettera A indica se mantenere lo stato inalterato (A=1,k=x, cioè qualsiasi) o cambiare lo stato (A=1, k che dice cosa fare) in un qualsiasi registro (normale o funzione)
6) Bisogna sempre prevedere un passaggio in cui il valore viene caricato nel registro
7) Mentre carico dei registri o svolgo operazioni dove non è coinvolta, i segnali dell'ALU sono tutti non specificati
8) I segnali beta messi come condizione di un if devono rimanere invariati per tutto l'if
9) Il bus non può essere usato per scrivere due informazioni in contemporanea
10) I segnali di abilitazione vanno sempre specificati (0 o 1) così come i segnali S e L. Solo quelli dell'ALU non vanno specificati
11) Nel fare la tabella dei segnali alpha si deve indicare in bus indirizzi sulle x il codice di chi scrive e sulle y il codice di chi legge. I codici si trovano sullo schema della macchina ad accumulatore
12) Se uno dei due bus non viene utilizzato si mette a non specificato
13) L'incremento del MAR è una modifica dell'architettura (come tutte le modifiche ai registri A e B come shift, incremento, ecc.) e dunque va introdotto un segnale k
14) Un registro va abilitato solo se viene scritto, non quando viene letto
15) Posso scrivere nella memoria e nel MAR contemporaneamente
    MBR -> M[MAR], IRx -> MAR;
16) Anche quando si fa -B -> B si occupa il bus dati
17) Quando si scrivono le istruzioni nella stessa riga bisogna considerare quali bus/circuiti vengono utilizzati
18) Se sposto una costante in un registro non
19) Se introduco dei registri sui bus devono valutare se “allungare” i codici x e y di lettura/scrittura. Tuttavia non è una soluzione senza costi, quindi si può optare per introdurli divisi per bus in modo da sfruttare i codici liberi già presenti
20) E ha due valori:
    0 -> non estende il segno (va bene per gli indirizzi) -> estende con zeri
    1 -> estende il segno (va bene per i numeri in complemento a 2)
21) Spostare IRx in un registro sul bus dati (es. A) prevede che venga specificato E
22) Per spostare il valore di A o B si deve farli passare tramite l’ALU (cosa che è possibile fare solo con B), A non è direttamente scrivibile) utilizzando il codice 0 0 0 (operazione B dell’ALU)
23) La micro istruzione vuota prevede Air a 1 e Zir a 1
24) Due righe della tabella di ROM che sono rispettivamente i primi elementi dell’if e dell’else di un ciclo sono uguali e differiscono solamente per il segnale beta che fa entrare nell’if o nell’else (dunque stato presente e futuro e altri segnali beta sono uguali)
25) Nel redarre la tabella di ROM bisogna mettere lo stesso stato presente per le due configurazioni (0 o 1) del segnale beta che identificano l'ingresso nell'if e nell'else:
    Nella tabella di ROM, quando vi è un goto bisogna mettere come stato futuro lo stato che identifica lo stato presente delle due microoperazioni che portano dentro l'if e l'else

TIPI DI ESERCIZI

1) Moltiplicazione per numero divisibile per 2: copio il valore in un registro a scorrimento sinistro
2) Moltiplicazione per n qualsiasi: aggiungo un registro contatore (cioè a decremento) e sposto il valore da moltiplicare in un registro collegato alla ALU in entrata e uscita così somma tale valore sempre a se stesso per n volte attarverso un goto e un if sul segnale beta in uscita dal registro contatore
3) Somma di due numeri: copio i valori in due registri semplici collegati alla alu e salvo il risultato in un terzo registro.
   Se mi serve conoscere il supero di capacità introduco un flip flop collegato al segnale sup dove salvo il bit. All'uscita del flip flop lego un SEGNALE BETA che posso utilizzare come condizione in un if.
   Se mi serve azzerare un flip flop o un registro introduco per ogni registro da azzerare un segnale di azzeramento da attivare
4) Acquisizione dati dall'esterno: se devo utilizzare un registro solo di memorizzazione, esso avrà un solo segnale alpha di abilitazione (An) che mi dice se
   - non cambiare lo stato (A=0)
   - cambiare lo stato (A=1)

   Se ho bisogno di caricare dati in un registro funzione basta introdurre un segnale k che mi dice se
   - prelevare un valore dall'esterno (k=0)
   - fare un'operazione (k=1)

   il cui valore ha senso se A=1; se A=0 è ininfluente.
5) Rendere output 0 o azzerare: bisogna introdurre un segnale di azzeramento ad ogni registro da azzerare (e che quindi potrebbe anche contenere solo zeri da dare poi in output) tramite un segnale alpha indicato con z
6) Spostare un valore indirizzo di loc. x in loc. x+1: costruisco x come registro a incremento
7) Scambiare due valori indirizzo di loc. x e x+1: costruisco x come registro a incremento e aggiungo dopo M un registro per memorizzare temporaneamente un valore
8) Confronto fra due numeri: si calcola la sottrazione fra i numeri di due registri normali e si salva il risultato in uno dei due. Si verifica poi se il primo bit (quellon più significativo) è 0 (quindi il primo numero era maggiore del secondo) o 1 (il secondo numero era maggiore del primo)
9) Copiare parole in memoria: devo stare attenta alla differenza di lunghezza fra entrate, informazione e memoria
10) Ciclo fino ad interruzione: si pone un if con la condizione complementare al tappo del ciclo e si inserisce un goto con etichetta all'interno del ciclo
11) Addizione di numeri negativi: si fa la complementazione della loro somma
12) Verifica che una condizione sia 0: si fa l'OR del segnale beta e cioè if OR(SBETA)='1' then… (quando l'or darà 0 allora non si entrerà nel ciclo)
13) Modifica di un registro normale: se si vuole modificare un registro bisogna aggiungere un segnale alpha k e un multiplexer che sceglie se nel registro verrà caricato il valore dal bus o quello ottenuto da un modulo combinatorio f
14) Analizzare due metà di un vettore: si introducono due registri contatori con i due indirizzi. Se l'indirizzo è X si porta direttamente IRx nel primo registro, in un registro ausiliare T si porta il valore pari alla metà della lunghezza del vettore, si sposta Irx in A e il valore di T in B e si sommano attraverso l'AlU per poi ottenere così il secondo indirizzo da mettere nel secondo registro indirizzo. Per un vettore di 32 elementi si ha:
   IRx -> IND1, 16 -> T1;
   IRx -> A, T1 -> B;
   A + B -> IND2;
15) Scambiare due elementi:
   Sposto il primo indirizzo nel mar, ottengo l'elemento e lo sposto in un registro;
   Sposto il secondo indirizzo nel mar, ottengo l'elemento e lo sposto in un secondo registro e lo sovrascrivo con l'elemento del primo registro;
   Riscrivo il primo indirizzo e ci copio l'elemento del secondo registro
   OPPURE
   Se in MBR si trova già un valore da scambiare e B si trova il secondo valore da scambiare:
   IND1 -> MAR;
   MBR -> M[MAR], B -> MBR, IND2 -> MAR;
   MBR -> M[MAR], INCR(IND1) -> IND1, INCR(IND2) -> IND2, DECR(T1) -> T1;
16) Verificare che due elementi siano uno l'opposto dell'altro:
   Si sposta il primo valore in un registro, si sposta il secondo valore in un secondo registro, si fa la somma e si verifica se l'OR

sia zero