

Compito dell'11 aprile 2018

Cognome e Nome dello studente: _____

Microprocessore L'istruzione `LOOP <label>` dell'8086 decrementa il valore del registro contatore `CX` e, se tale valore è maggiore di zero, salta a `<label>` (inizio del ciclo), altrimenti esce dal ciclo. Si vuole riprodurre questa funzionalità, estendendola, in un microprocessore CISC byte addressable a singolo bus interno con dati e indirizzi a 16 bit e memoria non segmentata, attraverso l'istruzione

`CYCLE [register] <jump_location> .`

L'operando `[register]` è opzionale, e indica quale, tra gli 8 registri general purpose della macchina, è quello impiegato come contatore. Il contatore di default è `R7`. L'operando `<jump_location>`, che individua la locazione di memoria a cui saltare, è espresso tramite un'etichetta (nel quale caso il salto è relativo, e l'operando rappresenta uno scostamento rispetto al valore attuale del PC) o un registro (nel quale caso il salto è assoluto, e il registro contiene il puntatore alla locazione di salto). Esempi di sintassi: `CYCLE pippo` (usa `R7`), `CYCLE R4 pippo`, `CYCLE R2 [R3]`.

Dopo aver disegnato la sezione di parte operativa strettamente necessaria, fornire una codifica plausibile dell'istruzione e disegnare l'automa di controllo relativo alla sua esecuzione, evidenziando con chiarezza e completezza ingressi e uscite della parte di controllo in ogni stato (ciò sarà alla base della soluzione dell'esercizio successivo). Calcolare il numero di cicli di bus necessari al fetch e il numero di cicli macchina necessari all'esecuzione di `CYCLE` al variare dei possibili formati dell'istruzione.

Reti sequenziali Progettare l'hardware di controllo per l'automa disegnato nell'esercizio precedente (dell'automa devono fare parte anche i due stati di partenza dal fetch e di rientro al fetch) secondo il classico schema "monoblocco": funzione di transizione di stato f , registro di stato M , funzione di uscita g (macchina di Moore). In particolare, specificare la funzione f attraverso il metodo orientato alla sintesi con multiplexer, cioè scrivendo una diversa tabella per ciascuno stato S_i dell'automa ($i = 1 \dots n$, con n numero degli stati della macchina). La tabella i -sima deve specificare lo stato futuro in funzione dei soli ingressi campionati in S_i .