Corso di Laurea in Ingegneria Informatica

Calcolatori — a.a. 2017–2018

Compito del 21 febbraio 2018

Cognome e Nome dello studente: _____

La divisione binaria di due interi $a \geq 0$ e $b > 0$ entrambi espressi su n bit è realizzata attraverso scorrimenti a sinistra, confronti, addizioni e sottrazioni. La parte operativa della macchina include i registri `RH` *e* `B` *da $n+1$ bit, posti all'ingresso di una* ALU *che opera in complemento a due, e i registri* `RL` *e* `Q` *da n bit. I registri* `RH` *e* `RL` *sono collegati in modo da formare uno shift register* `R` *da $2n+1$ bit. All'inizializzazione, a e b sono posti rispettivamente in* `RL` *e* `B`*, mentre* `RH` *e* `Q` *sono azzerati. Al termine del funzionamento, gli interi $q \geq 0$ e $0 \leq r < b$ tali che $a = q \times b + r$ si trovano rispettivamente in* `Q` *e* `RH`*, mentre* `RL` *contiene* 0*. L'algoritmo per la divisione binaria è il seguente:*

0. initialize (as above)
1. do the following n times:
   1.1. if $(\texttt{RH} < 0)$: $(\texttt{R} \ll 1)$; $\texttt{RH} \leftarrow (\texttt{RH} + \texttt{B})$
   else: $(\texttt{R} \ll 1)$; $\texttt{RH} \leftarrow (\texttt{RH} - \texttt{B})$ ['shift and operate']
   1.2. if $(\texttt{RH} < 0)$: $\texttt{Q} \leftarrow (\texttt{Q} \ll 1) + 0$
   else: $\texttt{Q} \leftarrow (\texttt{Q} \ll 1) + 1$ ['adjust quotient']
2. if $(\texttt{RH} < 0)$: $\texttt{RH} \leftarrow \texttt{RH} + \texttt{B}$ ['adjust remainder']

Reti Logiche Facendo riferimento al metodo di progettazione "parte operativa/parte di controllo": (I) disegnare la parte operativa di una macchina sequenziale sincrona che realizzi l'algoritmo sopra riportato; (II) specificarne il controllo attraverso il diagramma degli stati, facendo attenzione a riportare tutti i necessari segnali di comando (output) e di condizione (input); (III) simulare il funzionamento temporale della macchina per $n = 5$, $a = 23$, $b = 3$; (IV) realizzare la parte di controllo con la tecnica "registro di stato e multiplexer".

Programmazione LM Scrivere un programma in linguaggio Assembly 8086 che, date due variabili di memoria di tipo "byte" `a` e `b`, calcoli q e r secondo l'algoritmo sopra riportato, ponendoli rispettivamente in AL e DH.