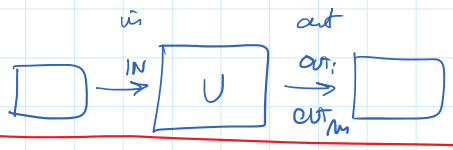
Esercizio 4 | 2017 (appello straordinario)

martedì 17 ottobre 2017 15:59



0. $(RDY_{in} = 0)$ nop, ∅
   $(=1)$ $I+1 \to I$, 1
1. $M[IN] \to X$, $IN \to TEMP$, 2

0. $(RDY_{in} \neq 0)$ nop, ∅
   $(=1)$ $I+1 \to I$, $M[IN] \to X$, $IN \to TEMP$, 2

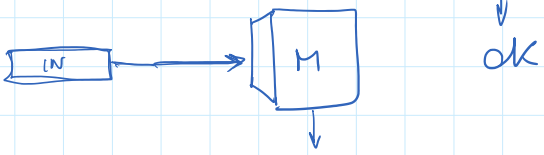
2. $(segn(\overset{M[IN]}{X} - k), Ack_{out} = 0 -)$ $M[\overset{IN}{TEMP}] + 1 \to M[\overset{IN}{TEMP}]$, set Ack_{in}, reset RDY_{in}, ∅
   $(=11)$ $\overset{M[IN]}{X} \to OUT_M$, $\overset{I+1}{I} \to OUT_i$, set RDY_{out}, reset Ack_{out}, set Ack_{in}, reset RDY_{in}, ∅
   $(=10)$ nop, 2

$W(0) = \{I, X, TEMP\}$

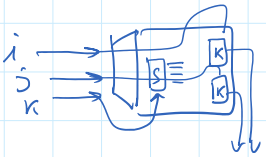
$R(2) = \{X, K, TEMP, M[I]\}$

0. $(RDY_{in}, segn(M[IN] - K), Ack_{out} = 0 - -)$ nop, ∅
   $(=110)$ nop, ∅
   $(=10-)$ $I+1 \to I$, $M[IN]+1 \to M[IN]$, set Ack_{in}, reset RDY_{in}, ∅
   $(=111)$ $I+1 \to I$, $M[IN] \to OUT_M$, $I+1 \to OUT_i$, set - - - - - - - , ∅

condizione di correttezza? (Po è una rete di Moore?)

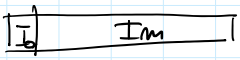


$R[i] + R[j] \to R[k]$



Esercizio 2 — 1 luglio 2013

mercoledì 18 ottobre 2017 08:49



U

U_1 U_1

op_esf $IN \to M[IND]$

op_esf $S \to U_2$

$S = \exists (M[I] \neq \emptyset)$

S_0 $\emptyset . (RDY_{IN}, OP = \emptyset -) nop, \emptyset$ z_1

$(=10)$ reset RDY_{in}, set ACK_{in}, $\emptyset \to I, \emptyset \to S, 1$ z_2

$(=11)$ " " , $IN \to M[IND], \emptyset$ z_3

S_1 1. $(I_0, OR[M[I_m]], ACK_{out} = 00 -) \; I+1 \to I, 1$ z_4

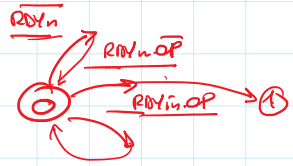
$(=01-) \; I+1 \xrightarrow{I} S + M[I_m] \to S, 1$ z_5

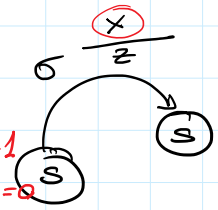
$(=1-0) \; nop, 1$ z_6

$(=1-1)$ set RDY_{out}, reset ACK_{out}, $S \to OUT, \emptyset .$ z_7

Automa Mealy

S	Stati
σ	$X \times S \to S$
ω	$X \times S \to Z$
X	
Z	





$S = \{\emptyset, 1\}$

$X = \{ \underset{x_0}{RDY_{IN}}, \underset{x_1}{RDY_{in}\,OP}, \underset{x_2}{RDY_{in}\,OP}, \underset{x_3}{I_0\,OR}, \underset{x_4}{I_0\,OR}, \underset{x_5}{I_0\,ACK_{out}}, \underset{x_6}{I_0\,ACK_{out}} \}$

$Z = \{$ conf di $\begin{vmatrix} \alpha_{K_I} & \alpha_{K_S} \\ \beta_{RDY_{in}} & \beta_{ACK_{in}} \\ \beta_{RDY_{out}} & \beta_{ACK_{out}} \; \beta_I \; \beta_S \; \beta_M \; \beta_{out} \end{vmatrix} \}$

$z_3 \doteq \beta_{RDY_{in}} = \beta_{ACK_{in}} = \beta_I = \beta_S = 1$
$\beta_{ACK_{out}} = \beta_{RDY_{out}} = \beta_M = \beta_{out} = 0$
$\alpha_{K_I} = \alpha_{K_S} = \emptyset$

ω	X_2
S_0	z_3
S_1	z_x

$\sigma =$

	x_0	x_1	x_2	x_3	x_4	x_5	x_6
S_0	S_0	S_1	S_0	—	—	—	—
S_1	—	—	—	S_1	S_1	S_1	$S_\emptyset$

$\emptyset$. $(RDY_{IN}, op = \emptyset -)$ nop, $\emptyset$ FR1

$(=10)$ reset RDY_{in}, set Ack_{in}, $\emptyset \to I$, $\emptyset \to S$, 1 FR2

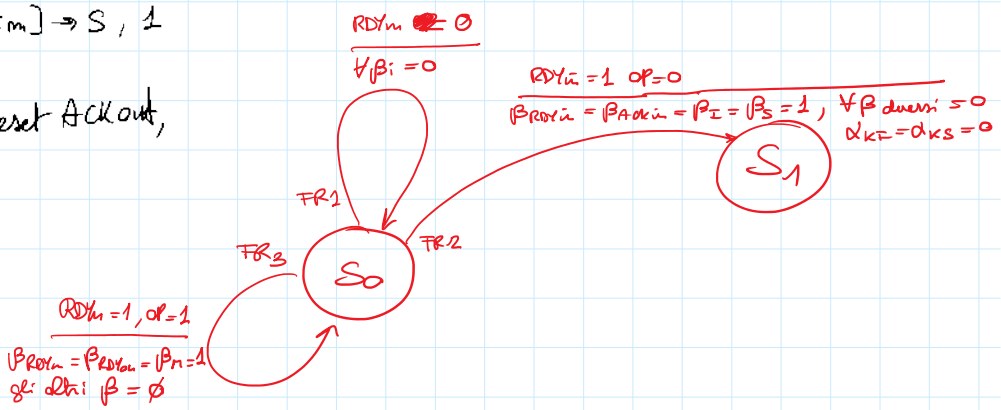
$(=11)$ " " , $IN \to M[IND]$, $\emptyset$ FR3

1. $(I_0, OR[M[I_m]], Ack_{out} = 00 -)$ $I+1 \to I$, 1

$(=01-)$ $I+1 \to I$, $S+M[I_m] \to S$, 1

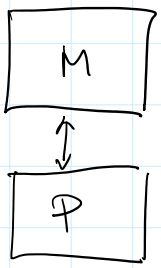
$(=1-0)$ nop, 1

$(=1-1)$ set RDY_{out}, reset Ack_{out},



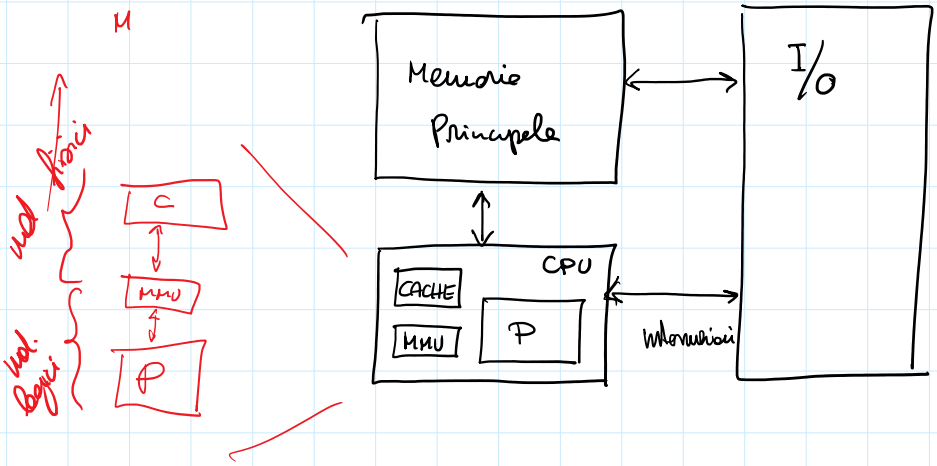
Livello Assembler

mercoledì 18 ottobre 2017 08:49



M:: while (true) {
accettano una richiesta { Lettura / Scrittura } dal P
la eseguono
restituiscono un risultato
}

P:: while (true) {
prende un'istruzione (ling. macchina assembler) dalla M
la esegue
scrive i risultati.
gestire interruzioni
}



dati su cui si lavora

D-RISC
32 bit

1) registri generali
2) memoria

{ memoria veloce
all'interno del processore
numero limitato
1 registro contiene una parola

M

Gli registri
Reg[0 ... 63]

Spazio di indirizzamento logico
(memoria virtuale)

M[0 ... MAX]

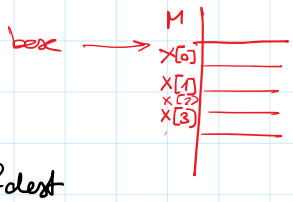
M[i] è una parola

Reg[0] = 0
costante

Reduced Instruction Set Computer
vs
CISC
Complex " "

Modi di indirizzamento

↳ modi x esprimere gli operandi delle istruzioni del linguaggio macchina



REGISTRO

ADD R_0, R_2, R_3

REG[1] + REG[2] → REG[3]

BASE + INDICE

LOAD $R_{base}, R_{indice}, R_{dest}$

M [REG[base] + REG[indice]] → REG[dest]

RELATIVO

alla istruzione che sto eseguendo

IC instruction counter → mi dice qual'è la prossima istr da eseguire
reg del livello FW

IF> R_x, R_y, etichetta $^{+6}_{-9}_{-3}$

se REG[x] > REG[y] allora prossima ic = ic + etichetta
altrimenti ic = ic + 1

IMMEDIATO

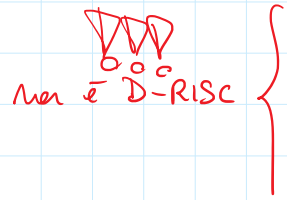
dato nell'istruzione

LOAD R_{base}, #3, R_{dest}

M[REG[base] + 3] → REG[dest]

INDIRETTO

Usa un reg x indicare l'indirizzo in memoria dove trovo il ~~dato~~



ADD (R1), R2, R3

M[REG[1]] + REG[2] → REG[3]

6

mercoledì 18 ottobre 2017 08:49

10

mercoledì 18 ottobre 2017 08:49

11

mercoledì 18 ottobre 2017 08:49

12

mercoledì 18 ottobre 2017 08:49

13

mercoledì 18 ottobre 2017 08:49

14

mercoledì 18 ottobre 2017 08:50