

CODICE RTL DEL REPERTORIO ISTRUZIONI

Istruzione	μ Sequenza RTL	μ Operazione
FETCH (COP: 00000000)	PC \rightarrow MAR; M[MAR] \rightarrow MBR; MBR \rightarrow IR, INCR(PC) \rightarrow PC;	μ_1 μ_2 μ_3
LOAD X (COP: 00000001)	IR _x \rightarrow MAR; M[MAR] \rightarrow MBR; MBR \rightarrow AC;	μ_4 μ_2 μ_5
LOAD @X (COP: 01000001)	IR _x \rightarrow MAR; M[MAR] \rightarrow MBR; MBR \rightarrow MAR; M[MAR] \rightarrow MBR; MBR \rightarrow AC;	μ_4 μ_2 μ_6 μ_2 μ_5
LOAD #X (COP: 10000001)	IR _x \rightarrow AC;	μ_7
LOAD %X (COP: 11000001)	IR _x \rightarrow A; PC \rightarrow B; A + B \rightarrow MAR M[MAR] \rightarrow MBR; MBR \rightarrow AC;	μ_8 μ_9 μ_{10} μ_2 μ_5
STORE X (COP: 00000010)	IR _x \rightarrow MAR, AC \rightarrow MBR; MBR \rightarrow M[MAR];	μ_{11} μ_{12}
ADD X (COP: 00000011)	IR _x \rightarrow MAR, AC \rightarrow A; M[MAR] \rightarrow MBR; MBR \rightarrow B; A + B \rightarrow AC;	μ_{13} μ_2 μ_{14} μ_{15}
SUB X (COP: 00000100)	IR _x \rightarrow MAR, AC \rightarrow A; M[MAR] \rightarrow MBR; MBR \rightarrow B; A - B \rightarrow AC;	μ_{13} μ_2 μ_{14} μ_{16}
JUMP X (COP: 00000101)	IR _x \rightarrow PC;	μ_{17}
JUMPZ X (COP: 00000110)	if OR(AC) = 0 then IR _x \rightarrow PC; else Φ ; fi	μ_{17} μ_0
IN (COP: 00000111)	IN \rightarrow AC;	μ_{18}
OUT (COP: 00001000)	AC \rightarrow OUT;	μ_{19}

ELENCO DELLE MICRO-OPERAZIONI

μ Operazione	Codice RTL
μ_0	Φ ;
μ_1	$PC \rightarrow MAR$;
μ_2	$M[MAR] \rightarrow MBR$;
μ_3	$MBR \rightarrow IR, INCR(PC) \rightarrow PC$;
μ_4	$IR_X \rightarrow MAR$;
μ_5	$MBR \rightarrow AC$;
μ_6	$MBR \rightarrow MAR$;
μ_7	$IR_X \rightarrow AC$;
μ_8	$IR_X \rightarrow A$;
μ_9	$PC \rightarrow B$;
μ_{10}	$A + B \rightarrow MAR$
μ_{11}	$IR_X \rightarrow MAR, AC \rightarrow MBR$;
μ_{12}	$MBR \rightarrow M[MAR]$
μ_{13}	$IR_X \rightarrow MAR, AC \rightarrow A$;
μ_{14}	$MBR \rightarrow B$;
μ_{15}	$A + B \rightarrow AC$;
μ_{16}	$A - B \rightarrow AC$;
μ_{17}	$IR_X \rightarrow PC$;
μ_{18}	$IN \rightarrow AC$;
μ_{19}	$AC \rightarrow OUT$;

ELENCO DELLE OPERAZIONI DELLA ALU

AL_0	AL_1	AL_2	OPERAZIONE
0	0	0	B
0	0	1	$B + 1$
0	1	0	\bar{B}
0	1	1	$-B$
1	0	0	$A + B$
1	0	1	$A \wedge B$
1	1	0	$A \vee B$
1	1	1	$A - B$

ELENCO DEI SEGNALI DI COMANDO

SEGNALE α	SEGNALE DI CONTROLLO	DESCRIZIONE
α_0	A_{IR}	Abilitazione scrittura registro IR
α_1	Z_{IR}	Azzeramento registro IR
α_2	A_{PC}	Abilitazione scrittura registro PC
α_3	K_{PC}	Incremento registro PC (se $A_{PC} = 1$)
α_4	A_{AC}	Abilitazione scrittura registro AC
α_5	A_{MAR}	Abilitazione scrittura registro MAR
α_6	A_{MBR}	Abilitazione scrittura registro MBR
α_7	S	Scrittura in memoria
α_8	L	Lettura dalla memoria
α_9	E	Estensione del segno
α_{10}	A_A	Abilitazione scrittura registro A
α_{11}	A_B	Abilitazione scrittura registro B
α_{12}	AL_0	Segnali di comando della ALU
α_{13}	AL_1	
α_{14}	AL_2	
α_{15}	A_{T1}	Abilitazione scrittura registro T1
α_{16}	K°_{T1}	Comandi funzione registro T1: Incremento (01), Decremento (10), <i>Liber</i> o (11)
α_{17}	K'_{T1}	
α_{18}	A_{T2}	Abilitazione scrittura registro T2
α_{19}	K°_{T2}	Comandi funzione registro T1: Incremento (01), Decremento (10), <i>Liber</i> o (11)
α_{20}	K'_{T2}	
α_{21}	x_1	Identificano il componente che scrive sul BUS INDIRIZZI
α_{22}	x_0	
α_{23}	y_1	Identificano il componente che legge dal BUS INDIRIZZI
α_{24}	y_0	
α_{25}	x_2	Identificano il componente che scrive sul BUS DATI
α_{26}	x_1	
α_{27}	x_0	
α_{28}	y_3	Identificano il componente che legge dal BUS DATI
α_{29}	y_2	
α_{30}	y_1	
α_{31}	y_0	

TABELLA DEI COMANDI

μ	CODICE RTL	A_{IR}	Z_{IR}	A_{PC}	K_{PC}	A_{AC}	A_{MAR}	A_{MBR}	S	L	E	A_A	A_B	AL_0	AL_1	AL_2	A_{T1}	$K^{\circ} T_1$	K'_{T1}	A_{T2}	$K^{\circ} T_2$	K'_{T2}	Bus Indirizzi		Bus Dati				
		a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}	a_{11}	a_{12}	a_{13}	a_{14}	a_{15}	a_{16}	a_{17}	a_{18}	a_{19}	a_{20}	$x_1 x_0$	$y_1 y_0$	$x_2 x_1 x_0$	$y_3 y_2 y_1 y_0$			
																							$a_{21} a_{22}$	$a_{23} a_{24}$	$a_{25} a_{26} a_{27}$	$a_{28} a_{29} a_{30} a_{31}$			
μ_0	Φ ;	0	-	0	-	0	0	0	0	0	-	0	0	-	-	-	0	-	-	0	-	-	--	--	---	----			
μ_1	$PC \rightarrow MAR$;	0	-	0	-	0	1	0	0	0	-	0	0	-	-	-	0	-	-	0	-	-	1 0	0 1	---	----			
μ_2	$M[MAR] \rightarrow MBR$;	0	-	0	-	0	0	1	0	1	-	0	0	-	-	-	0	-	-	0	-	-	--	--	---	----			
μ_3	$MBR \rightarrow IR, INCR(PC) \rightarrow PC$;	1	0	1	1	0	0	0	0	0	-	0	0	-	-	-	0	-	-	0	-	-	--	--	0 0 1	0 0 1 1			
μ_4	$IR_x \rightarrow MAR$;	0	-	0	-	0	1	0	0	0	-	0	0	-	-	-	0	-	-	0	-	-	0 1	0 1	---	----			
μ_5	$MBR \rightarrow AC$;	0	-	0	-	1	0	0	0	0	-	0	0	-	-	-	0	-	-	0	-	-	--	--	0 0 1	0 1 0 0			
μ_6	$MBR \rightarrow MAR$;	0	-	0	-	0	1	0	0	0	-	0	0	-	-	-	0	-	-	0	-	-	0 0	0 1	0 0 1	0 0 0 0			
μ_7	$IR_x \rightarrow AC$;	0	-	0	-	1	0	0	0	0	1	0	0	-	-	-	0	-	-	0	-	-	0 1	0 0	0 0 0	0 1 0 0			
μ_8	$IR_x \rightarrow A$;	0	-	0	-	0	0	0	0	0	0	1	0	-	-	-	0	-	-	0	-	-	0 1	0 0	0 0 0	0 1 0 1			
μ_9	$PC \rightarrow B$;	0	-	0	-	0	0	0	0	0	0	0	1	-	-	-	0	-	-	0	-	-	1 0	0 0	0 0 0	0 1 1 0			
μ_{10}	$A + B \rightarrow MAR$	0	-	0	-	0	1	0	0	0	-	0	0	1	0	0	0	-	-	0	-	-	0 0	0 1	1 0 0	0 0 0 0			
μ_{11}	$IR_x \rightarrow MAR, AC \rightarrow MBR$;	0	-	0	-	0	1	1	0	0	-	0	0	-	-	-	0	-	-	0	-	-	0 1	0 1	0 1 1	0 0 0 1			
μ_{12}	$MBR \rightarrow M[MAR]$	0	-	0	-	0	0	0	1	0	-	0	0	-	-	-	0	-	-	0	-	-	--	--	---	----			
μ_{13}	$IR_x \rightarrow MAR, AC \rightarrow A$;	0	-	0	-	0	1	0	0	0	-	1	0	-	-	-	0	-	-	0	-	-	0 1	0 1	0 1 1	0 1 0 1			
μ_{14}	$MBR \rightarrow B$;	0	-	0	-	0	0	0	0	0	-	0	1	-	-	-	0	-	-	0	-	-	--	--	0 0 1	0 1 1 0			
μ_{15}	$A + B \rightarrow AC$;	0	-	0	-	1	0	0	0	0	-	0	0	1	0	0	0	-	-	0	-	-	--	--	1 0 0	0 1 0 0			
μ_{16}	$A - B \rightarrow AC$;	0	-	0	-	1	0	0	0	0	-	0	0	1	1	1	0	-	-	0	-	-	--	--	1 0 0	0 1 0 0			
μ_{17}	$IR_x \rightarrow PC$;	0	-	1	-	0	0	0	0	0	-	0	0	-	-	-	0	-	-	0	-	-	0 1	1 0	---	----			
μ_{18}	$IN \rightarrow AC$;	0	-	0	-	1	0	0	0	0	-	0	0	-	-	-	0	-	-	0	-	-	--	--	0 1 0	0 1 0 0			
μ_{19}	$AC \rightarrow OUT$;	0	-	0	-	0	0	0	0	0	-	0	0	-	-	-	0	-	-	0	-	-	--	--	0 1 1	0 0 1 0			

ROM DELLA PARTE DI CONTROLLO

La Parte di Controllo del Calcolatore (P.C.) utilizza un registro di stato a 3 bit, in quanto la μ -sequenza più lunga da eseguire è composta da 5 μ -operazioni (vedi istruzione LOAD@ X).

La ROM della P.C. ha indirizzi a 11 bit (8 per i segnali istruzione $I_7 \dots I_0$, 3 per i segnali condizione β_{AC} , β_{T1} e β_{T2} e 3 per lo stato attuale $y_2 y_1 y_0$) e parole di memoria a 35 bit (32 per i segnali di comando $\alpha_0 \dots \alpha_{31}$ e 3 per lo stato futuro $y'_2 y'_1 y'_0$). Il registro IR deve essere azzerato al termine dell'esecuzione di ogni istruzione (a eccezione della FETCH). Il valore del segnale Z_{IR} è riportato in una colonna a parte (vedi ultima colonna).

ISTRUZIONE	INDIRIZZO DELLA ROM							CONTENUTO DELLA ROM											
	Segnali Istruzione							Segnali Condizione			Stato corrente			Stato futuro			Segnali di comando		
	I ₇	I ₆	I ₅	I ₄	I ₃	I ₂	I ₁	I ₀	β _{AC}	β _{T1}	β _{T2}	y ₂	y ₁	y ₀	y' ₂	y' ₁	y' ₀	α ₀ ... α ₃₁ (μ)	Z _{IR}
FETCH	00000000							–	–	–	0	0	0	0	0	0	1	μ ₁	0
	00000000							–	–	–	0	0	1	0	1	0	μ ₂	0	
	00000000							–	–	–	0	1	0	0	0	0	μ ₃	0	
LOAD X	00000001							–	–	–	0	0	0	0	0	1	μ ₄	0	
	00000001							–	–	–	0	0	1	0	1	0	μ ₂	0	
	00000001							–	–	–	0	1	0	0	0	0	μ ₅	1	
LOAD @X	01000001							–	–	–	0	0	0	0	0	1	μ ₄	0	
	01000001							–	–	–	0	0	1	0	1	0	μ ₂	0	
	01000001							–	–	–	0	1	0	0	1	1	μ ₆	0	
	01000001							–	–	–	0	1	1	1	0	0	μ ₂	0	
	01000001							–	–	–	1	0	0	0	0	0	μ ₅	1	
LOAD #X	10000001							–	–	–	0	0	0	0	0	0	0	μ ₇	1
LOAD %X	11000001							–	–	–	0	0	0	0	0	1	μ ₈	0	
	11000001							–	–	–	0	0	1	0	1	0	μ ₉	0	
	11000001							–	–	–	0	1	0	0	1	1	μ ₁₀	0	
	11000001							–	–	–	0	1	1	1	0	0	μ ₂	0	
	11000001							–	–	–	1	0	0	0	0	0	μ ₅	1	
STORE X	00000010							–	–	–	0	0	0	0	0	1	μ ₁₁	0	
	00000010							–	–	–	0	0	1	0	0	0	μ ₁₂	1	
ADD X	00000011							–	–	–	0	0	0	0	0	1	μ ₁₃	0	
	00000011							–	–	–	0	0	1	0	1	0	μ ₂	0	
	00000011							–	–	–	0	1	0	0	1	1	μ ₁₄	0	
	00000011							–	–	–	0	1	1	0	0	0	μ ₁₅	1	
SUB X	00000100							–	–	–	0	0	0	0	0	1	μ ₁₃	0	
	00000100							–	–	–	0	0	1	0	1	0	μ ₂	0	
	00000100							–	–	–	0	1	0	0	1	1	μ ₁₄	0	
	00000100							–	–	–	0	1	1	0	0	0	μ ₁₆	1	
JUMP X	00000101							–	–	–	0	0	0	0	0	0	μ ₁₇	1	
JUMPZ X	00000110							0	–	–	0	0	0	0	0	0	μ ₁₇	1	
	00000110							1	–	–	0	0	0	0	0	0	μ ₀	1	
IN	00000111							–	–	–	0	0	0	0	0	0	μ ₁₈	1	
OUT	00001000							–	–	–	0	0	0	0	0	0	μ ₁₉	1	