Control Logic

Instruction	State #	Condition	Data Path	PCen	PCLD	Aen	Aout	IRen	Ten	Tout	ALUout	ALUM	ROMout	DEV1en	DEV2en	DEV0en	MEMen	MEMin	MEMout	ALUCn	Examples	Encoding	Description
FETCH	**S0**	**X**	**ROM -> IR**					0				X	0							X	/	/	Fetch phase for every instruction.
	S1		**PC+1 -> PC**	0								X								X	/	/	
OUT	**S2**	**OPC=0 & OPC2=0**	**A -> DEV1**				0					X		0						X	*OUT DEV1, A*	*20*	Output the content of register A or T to the output device DEV1 or DEV2.
		OPC=0 & OPC2=1	**A -> DEV2**												0						*OUT DEV2, A*	*24*	
		OPC=1 & OPC2=0	**T -> DEV1**							0				0							*OUT DEV1, T*	*28*	
		OPC=1 & OPC2=1	**T -> DEV2**												0						*OUT DEV2, T*	*2C*	
IN	**S3**	**OPC=0**	**DEV0 -> A**			0						X				0				X	*IN A*	*30*	Obtain one byte from the input device DEV0 to register A or T.
		OPC=1	**DEV0 -> T**						0												*IN T*	*38*	
LDR	**S4**	**X**	**RAM -> A**			0						X					0		0	X	*LDR 0x123*	*41 23*	Load one byte from the given location of RAM into register A.
	S5		**PC+1 -> PC**	0								X								X			
SUB*	**S6**	**OPC=0**	**A-T -> A**			0					0	0								0	*SUB A*	*60*	Subtract the content of register T from that of A, and store the result into A or T.
		OPC=1	**A-T -> T**						0												*SUB T*	*68*	
LDI	**S7**	**OPC=0**	**ROM -> A**			0						X	0							X	*LDI A, #0xAB*	*70 AB*	Load a one-byte immediate number from ROM to register A or T.
		OPC=1	**ROM -> T**						0												*LDI T, #0xEF*	*78 EF*	
	S8	**X**	**PC+1 -> PC**	0								X								X	/	/	
ADD*	**S9**	**OPC=0**	**A+T -> A**			0					0	0								1	*ADD A*	*90*	Add the contents of register A and T, and store the result into A or T.
		OPC=1	**A+T -> T**						0												*ADD T*	*98*	
JMP	**S10 (A)**	**X**	**IR+ROM -> PC**		0							X	0							X	*JMP 0x123*	*A1 23*	Unconditional jump to the given address.
AND*	**S11 (B)**	**OPC=0**	**A&T -> A**			0					0	1								X	*AND A*	*B0*	Execute AND operation for registers A and T, and store the result into A or T.
		OPC=1	**A&T -> T**						0												*AND T*	*B8*	
STR	**S12 (C)**	**X**	**A -> RAM**				0					X					0	0		X	*STR 0x123*	*C1 23*	Store the content of register A to the given address of RAM.
	S13 (D)		**PC+1 -> PC**	0								X								X			
OR*	**S14 (E)**	**OPC=0**	**A | T -> A**			0					0	1								X	*OR A*	*E0*	Execute OR operation for registers A and T, and store the result into A or T.
		OPC=1	**A | T -> T**						0												*OR T*	*E8*	
JNZ	**S15 (F)**	**Zflag=0 (A=0)**	**IR+ROM -> PC**	0								X								X	*JNZ 0x123*	*F1 23*	Jump to the given address (ROM) if the Zflag is 0.
		Zflag=1 (A≠0)	**PC+1 -> PC**		0								0										

Note:
(1) Every blank cell in the table means setting to 1 (filled by 1), symbol ‘X’ means don’t care (can be 0 or 1), and symbol ‘/’ means not applicable.
(2) For all instructions with ‘*’ (i.e., SUB, ADD, AND, and OR), the opcodes correspond to 74X181 chip’s function codes.
(3) Every instruction using the lowest 4 bits of IR register as the highest 4 address bits cannot make these lowest 4 bits from IR as any condition bits, because these bits are occupied for address.