INTERNAL PROCESSOR OPERATION

INDEX REGISTER INSTRUCTIONS

INSTRUCTION CODING D_7D_6	$D_5D_4D_3$	$D_2D_1D_0$	OPERATION	# OF STATES TO EXECUTE INSTRUCTION	MEMORY CYCLE ONE (1) T1 (2)	T2	T3	T4 (3)	T5
1 1	D D D	S S S	Lr_1r_2	(5)	PC_LOUT (4)	PC_HOUT	FETCH INSTR. (5) TO IR & REG. b	SSS TO REG. b (6)	REG. b TO DDD
1 1	D D D	1 1 1	LrM	(8)	PC_LOUT	PC_HOUT	FETCH INSTR. TO IR & REG. b	(7)	
1 1	1 1 1	S S S	LMr	(7)	PC_LOUT	PC_HOUT	FETCH INSTR. TO IR & REG. b	SSS TO REG. b	
0 0	D D D	1 1 0	Lrl	(8)	PC_LOUT	PC_HOUT	FETCH INSTR. TO IR & REG. b		
0 0	1 1 1	1 1 0	LMI	(9)	PC_LOUT	PC_HOUT	FETCH INSTR. TO IR & REG. b		
0 0	D D D	0 0 0	INr	(5)	PC_LOUT	PC_HOUT	FETCH INSTR. TO IR & REG. b	X	ADD OP - FLAGS AFFECTED
0 0	D D D	0 0 1	DCr	(5)	PC_LOUT	PC_HOUT	FETCH INSTR. TO IR & REG. b	X	SUB OP - FLAGS AFFECTED

ACCUMULATOR GROUP INSTRUCTIONS

1 0	P P P	S S S	ALU OP r	(5)	PC_LOUT	PC_HOUT	FETCH INSTR. TO IR & REG. b	SSS TO REG. b	ALU OP - FLAGS AFFECTED
1 0	P P P	1 1 1	ALU OP M	(8)	PC_LOUT	PC_HOUT	FETCH INSTR. TO IR & REG. b		
0 0	P P P	1 0 0	ALU OP I	(8)	PC_LOUT	PC_HOUT	FETCH INSTR. TO IR & REG. b		
0 0	0 0 0	0 1 0	RLC	(5)	PC_LOUT	PC_HOUT	FETCH INSTR. TO IR & REG. b	X	ROTATE REG. A CARRY AFFECTED
0 0	0 0 1	0 1 0	RRC	(5)	PC_LOUT	PC_HOUT	FETCH INSTR. TO IR & REG. b	X	ROTATE REG. A CARRY AFFECTED
0 0	0 1 0	0 1 0	RAL	(5)	PC_LOUT	PC_HOUT	FETCH INSTR. TO IR & REG. b	X	ROTATE REG. A CARRY AFFECTED
0 0	0 1 1	0 1 0	RAR	(5)	PC_LOUT	PC_HOUT	FETCH INSTR. TO IR & REG. b	X	ROTATE REG. A CARRY AFFECTED

PROGRAM COUNTER AND STACK CONTROL INSTRUCTIONS

0 1	X X X	1 0 0	JMP	(11)	PC_LOUT	PC_HOUT	FETCH INSTR. TO IR & REG. b		
0 1	0 C C	0 0 0	JFc	(9 or 11)	PC_LOUT	PC_HOUT	FETCH INSTR. TO IR & REG. b		
0 1	1 C C	0 0 0	JTc	(9 or 11)	PC_LOUT	PC_HOUT	FETCH INSTR. TO IR & REG. b		
0 1	X X X	1 1 0	CAL	(11)	PC_LOUT	PC_HOUT	FETCH INSTR. TO IR & REG. b		
0 1	0 C C	0 1 0	CFc	(9 or 11)	PC_LOUT	PC_HOUT	FETCH INSTR. TO IR & REG. b		
0 1	1 C C	0 1 0	CTc	(9 or 11)	PC_LOUT	PC_HOUT	FETCH INSTR. TO IR & REG. b		
0 0	X X X	1 1 1	RET	(5)	PC_LOUT	PC_HOUT	FETCH INSTR. TO IR & REG. b	POP STACK	X
0 0	0 C C	0 1 1	RFc	(3 or 5)	PC_LOUT	PC_HOUT	FETCH INSTR. TO IR & REG. b	POP STACK (13)	X
0 0	1 C C	0 1 1	RTc	(3 or 5)	PC_LOUT	PC_HOUT	FETCH INSTR. TO IR & REG. b	POP STACK (13)	X
0 0	A A A	1 0 1	RST	(5)	PC_LOUT	PC_HOUT	FETCH INSTR. TO REG. b AND PUSH STACK (0→REG. a)	REG. a TO PC_H	REG. b TO PC_L (14)

I/O INSTRUCTIONS

0 1	0 0 M	M M 1	INP	(8)	PC_LOUT	PC_HOUT	FETCH INSTR. TO IR & REG. b		
0 1	R R M	M M 1	OUT	(6)	PC_LOUT	PC_HOUT	FETCH INSTR. TO IR & REG. b		

MACHINE INSTRUCTIONS

0 0	0 0 0	0 0 X	HLT	(4)	PC_LOUT	PC_HOUT	FETCH INSTR. TO IR & REG. b & HALT (18)		
1 1	1 1 1	1 1 1	HLT	(4)	PC_LOUT	PC_HOUT	FETCH INSTR. TO IR & REG. b & HALT (18)		

NOTES:

1. The first memory cycle is always a PCI (instruction) cycle.
2. Internally, states are defined as T1 through T5. In some cases more than one memory cycle is required to execute an instruction.
3. Content of the internal data bus at T4 and T5 is available at the data bus. This is designed for testing purposes only.
4. Lower order address bits in the program counter are denoted by PC_L and higher order bits are designated by PC_H.
5. During an instruction fetch the instruction comes from memory to the instruction register and is decoded.
6. Temporary registers are used internally for arithmetic operations and data transfers (Register a and Register b.)
7. These states are skipped.
8. PCR cycle (Memory Read Cycle).
9. "X" denotes an idle state.
10. PCW cycle (Memory Write Cycle).
11. When the JUMP is conditional and the condition fails, states T4 and T5 are skipped and the state counter advances to the next memory cycle.

MEMORY CYCLE TWO					MEMORY CYCLE THREE				
T1	T2	T3	T4	T5	T1	T2	T3	T4	T5
REG. L OUT (8)	REG. H OUT	DATA TO REG. b	X (9)	REG. b TO DDD					
REG. L OUT (10)	REG. H OUT	REG. b TO OUT							
PC_LOUT (8)	PC_HOUT	DATA TO REG. b	X	REG. b TO DDD					
PC_LOUT (8)	PC_HOUT	DATA TO REG. b	→		REG. L OUT(10)	REG. H OUT	REG. b TO OUT		

REG. L OUT (8)	REG. H OUT	DATA TO REG. b	X	ALU OP - FLAGS AFFECTED					
PC_LOUT (8)	PC_HOUT	DATA TO REG. b	X	ARITH OP - FLAGS AFFECTED					

PC_LOUT (8)	PC_HOUT	LOWER ADD. TO REG. b	→		PC_LOUT(8)	PC_HOUT	HIGHER ADD. REG. a	REG. a TO PC_H	REG. b TO PC_L
PC_LOUT (8)	PC_HOUT	LOWER ADD. TO REG. b	→		PC_LOUT(8)	PC_HOUT	HIGHER ADD. REG. a (11)	REG. a TO PC_H	REG. b TO PC_L
PC_LOUT (8)	PC_HOUT	LOWER ADD. TO REG. b	→		PC_LOUT(8)	PC_HOUT	HIGHER ADD. REG. a (11)	REG. a TO PC_H	REG. b TO PC_L
PC_LOUT (8)	PC_HOUT	LOWER ADD. TO REG. b	→		PC_LOUT(8)	PC_HOUT	HIGHER ADD. REG. a	REG. a TO PC_H	REG. b TO PC_L
PC_LOUT (8)	PC_HOUT	LOWER ADD. TO REG. b	→		PC_LOUT(8)	PC_HOUT	HIGHER ADD. REG. a (12)	REG. a TO PC_H	REG. b TO PC_L
PC_LOUT (8)	PC_HOUT	LOWER ADD. TO REG. b	→		PC_LOUT(8)	PC_HOUT	HIGHER ADD. REG. a (12)	REG. a TO PC_H	REG. b TO PC_L

REG. A TO OUT (15)	REG. b TO OUT	DATA TO REG. b	COND ff OUT (16)	REG. b TO REG. A					
REG. A TO OUT (15)	REG. b TO OUT	X (17)							

12. When the CALL is conditional and the condition fails, states T4 and T5 are skipped and the state counter advances to the next memory cycle. If the condition is true, the stack is pushed at T4, and the lower and higher order address bytes are loaded into the program counter.
13. When the RETURN condition is true, pop up the stack; otherwise, advance to next memory cycle skipping T4 and T5.
14. Bits D_3 through D_5 are loaded into PC_L and all other bits are set to zero; zeros are loaded into PC_H.
15. PCC cycle (I/O Cycle).
16. The content of the condition flip-flops is available at the data bus: S at D_0, Z at D_1, P at D_2, C at D_3.
17. A READY command must be supplied for the OUT operation to be completed. An idle T3 state is used and then the state counter advances to the next memory cycle.
18. When a HALT command occurs, the CPU internally remains in the T3 state until an INTERRUPT is recognized. Externally, the STOPPED state is indicated.