8.3 Branch Group

PTU - Dec. 2004, May 2005, May 2006, Ma

The microprocessor executes machine codes in a sequential manner. It goes on executing from memory location to the next. Branch group of instructions instructs the microprocessor to go to a diffe memory location. The microprocessor continues executing machine codes from that new location. address of the new memory location is either specified in the instruction or supplied by the micropro or given by extra hardware. The branch group instructions are classified in 3 categories :

(a) **CALL instructions**

1. JMP address
2. Conditional CALL instructions
3. PCHL

(b) **Call and return instructions**

1. CALL address
2. Conditional call instructions
3. RET
4. Conditional RET instructions.

(c) **Restart instructions**

RST N

1. JMP Address

Mnemonic	JMP Address
Operation	PC = Address
No. of Bytes	3 bytes **First byte :** Opcode of JMP **Second byte :** Low order byte of address **Third byte :** High order byte of address
Machine Cycles	3 (OF + MR + MR)

Algorithm	PC ← Address
Flags	No flags are affected .
Addr. Mode	Implied addressing mode.
T-states	10 (4 + 3 + 3)

Description	• This instruction loads the PC with the address given within the instructio continues with the program execution from this location.
Example JMP C200 H.	• This instruction will load the PC with C200 H and the processor will fetc instruction from this address. • This is a 3 byte instruction. So it requires 3 machine cycles to fetch the instru (1) OPCODE fetch (2) Memory Read (3) Memory Read.

(1) **OPCODE fetch :** The program counter places address on the lower order address bus and the higher order address bus. This cycle is used to read OPCODE of JMP instruction i.e. C3 H. The address for this machine cycle is given by PC. The PC is then incremented by 1.

(2) **Memory read :** The program counter places address on the lower order address bus and the higher order address bus. The lower order byte of the address specified is read using PC. The PC is then incremented by one.

(3) **Memory read :** The program counter retains the address on the lower order address bus and the higher order address bus. The higher order byte of the address is read using PC. The PC is then incremented by one.

- The timing diagram of JMP address is as shown in Fig. 8.3.1.

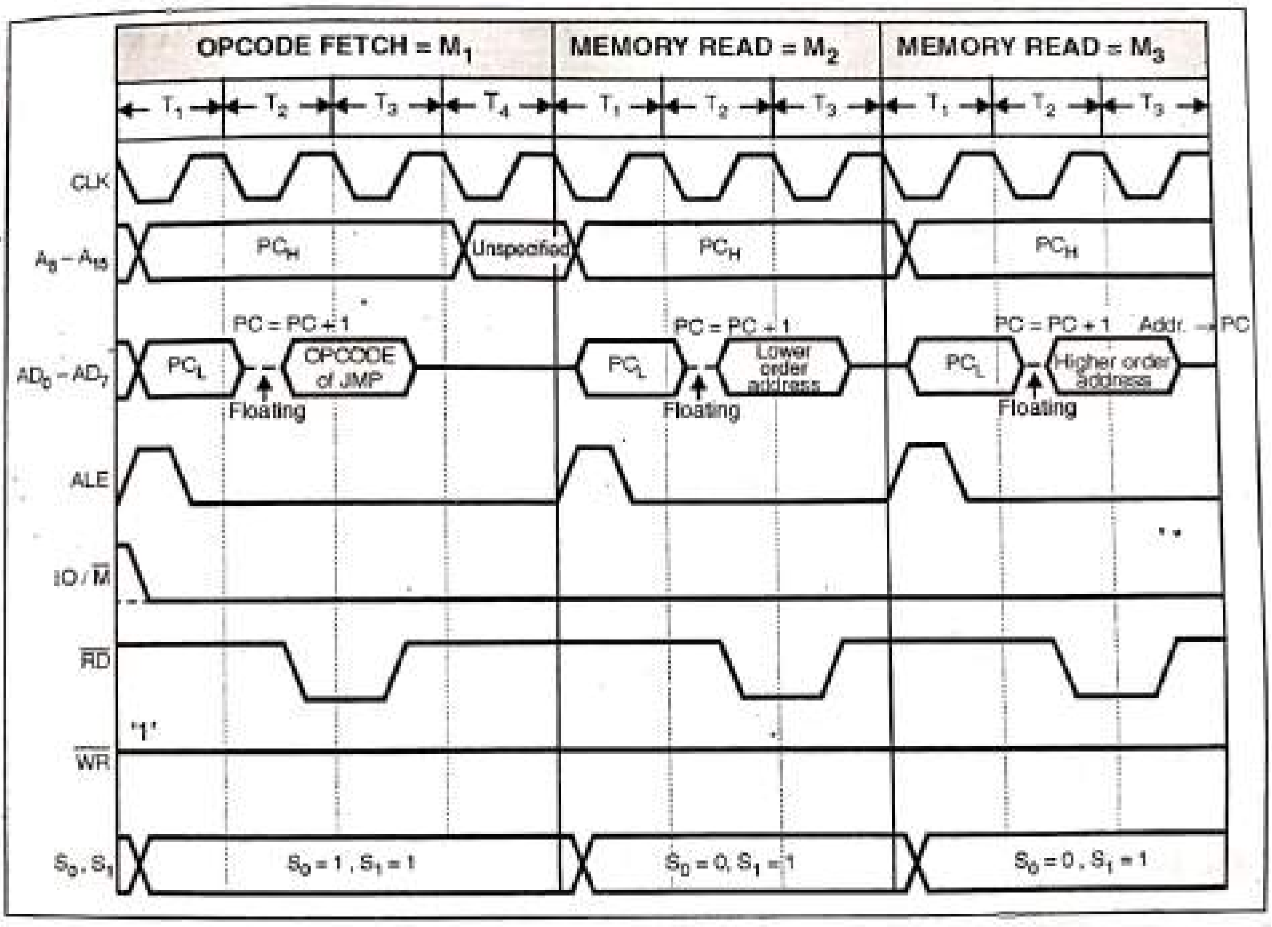


Fig. 8.3.1 : Timing diagram of JMP address

2. Conditional JUMP Instructions

Mnemonic	Jcond Address
Operation	if condition true, PC = address else, PC = PC + 3
No. of Bytes	3 bytes **First byte :** Opcode **Second byte :** Low order byte of the address **Third byte :** High order byte of the address.
Machine Cycles	If condition is false, 2 (OF + MR) If condition is true, 3 (OF + MR + MR)

Algorithm	If condition is true PC ← Address Else PC ← PC + 3
Flags	No flags are modified, only flags are checked.
Addr. Mode	Implied addressing mode
T-states	If condition is false, 7 (4 + 3) If condition is true, 10 (4 + 3 + 3)

Description	• In conditional JUMP instructions, when the condition is true or satisfied then on JUMP is made at the specified address. • If condition is false or not satisfied it will just check and proceed further to execu the next instruction after it.
Example JZ C200	• Let ZF = 1 [illegible] • This instruction will cause a JUMP to an address C200 H. i.e. program counter wi load with C200 H as ZF =1. • This is a 8 byte instruction. In this instruction there are two possible states condition : (1) Condition is satisfied, (2) Condition is not satisfied. (1) If the condition is satisfied the instruction will be similar to JMP address instructio So the timing diagram of this condition will be same as Fig. 8.3.1. (2) If the condition is not satisfied the instruction will go to execute the next instructi after that. In it the instruction requires 2 machine cycles : (1) OPCODE fetch, (2) Memory read. So the timing diagram of this condition is shown in Fig. 8.3.2.

Table below shows the possible condition for jumps.

Instruction code	Description	Condition for JUMP
JZ	JUMP if zero	ZF = 1
JNZ	JUMP if not zero	ZF = 0
JP	JUMP if positive	SF=0
JM	JUMP on minus	SF = 1
JPO	JUMP if parity odd	PF = 0
JPE	JUMP if parity even	PF = 1
JC	JUMP if carry	CF = 1
JNC	JUMP if no carry	CF = 0

Remember, there is no JUMP on auxiliary carry flag.

- During OPCODE fetch, the OPCODE of instruction is fetched from the memory. It is then loa
in instruction register, then it is decoded. Now the microprocessor knows that this is a conditiona
JMP instruction. Meanwhile, the OPCODE fetch cycle is completed and microprocessor will start th
next memory read cycle.
- During this cycle, the condition of flags is checked by microprocessor (in state of memory read cy
- If the condition is satisfied microprocessor will proceed for next memory read cycle.
- But if the condition is not satisfied the PC contents are incremented by 2 and microprocess
will leave the conditional jump instruction and using PC contents, it starts OPCODE fetch cycle f
next instruction.

The timing diagram of conditional jump instruction when condition is not satisfied is as shown in Fig. 8.3.2.

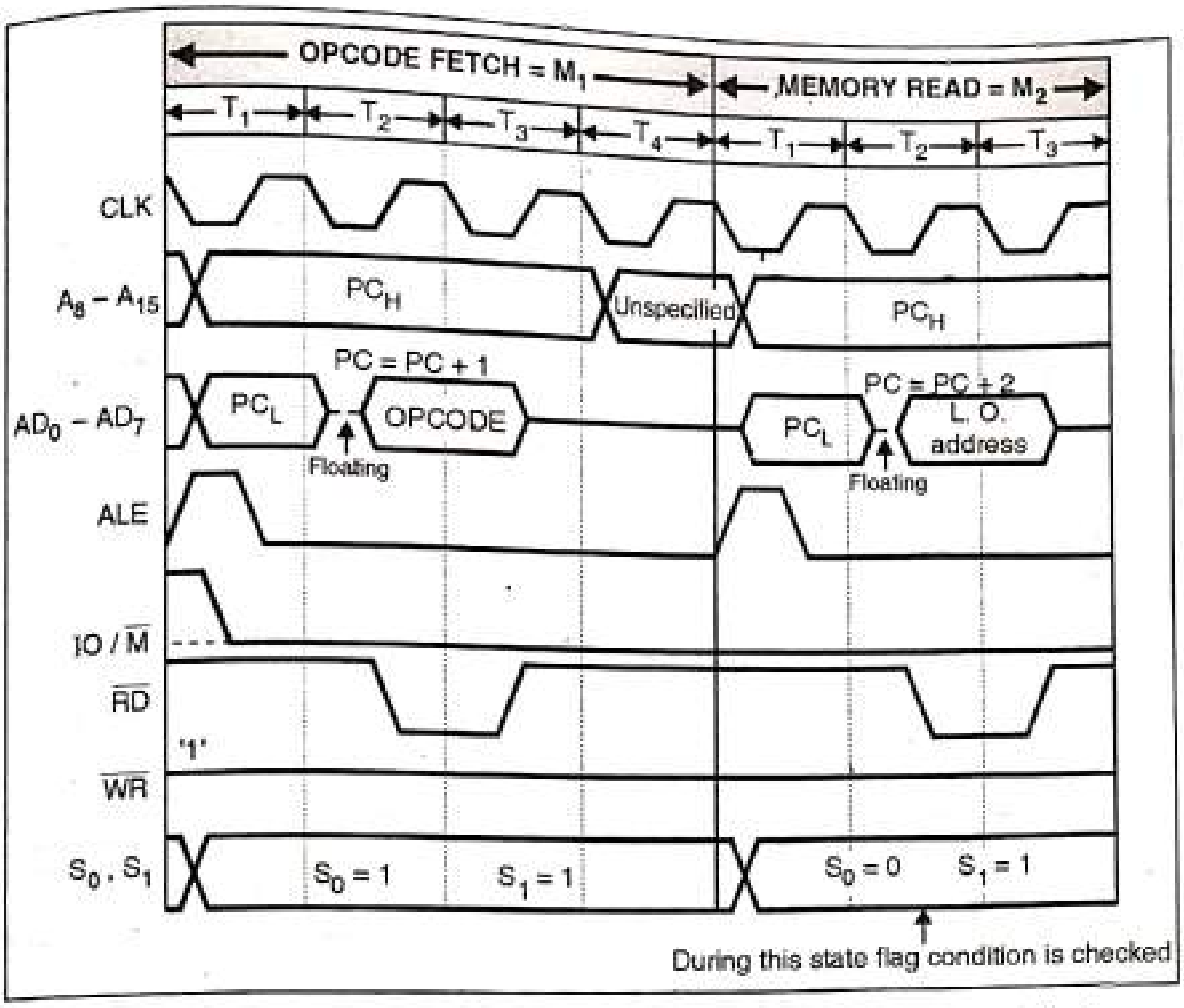


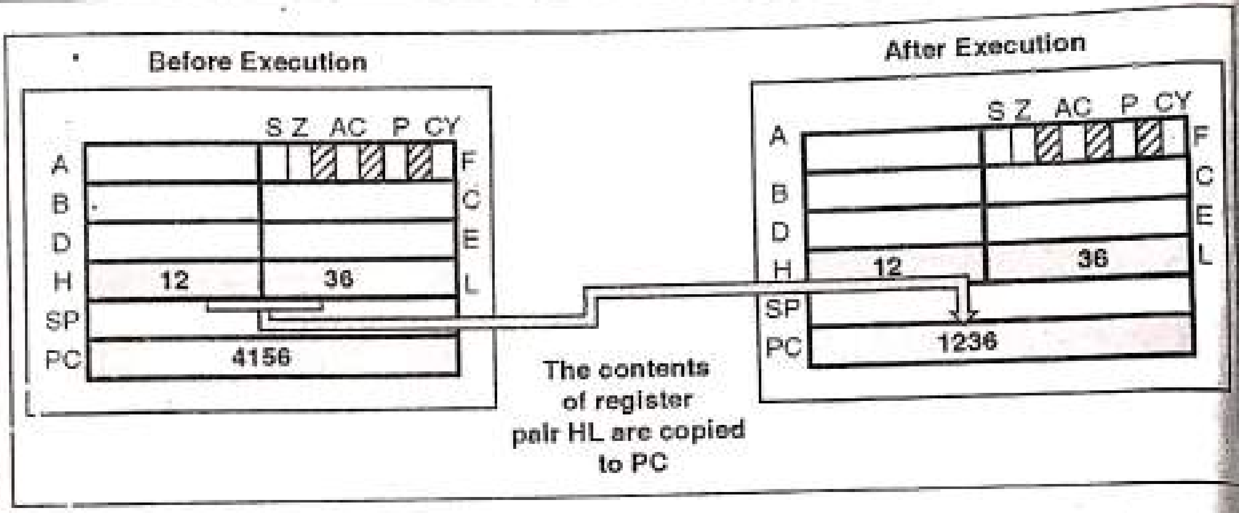
Fig. 8.3.2 : Timing diagram of JMP conditional when condition is not satisfied

PCHL

Mnemonic	PCHL
Operation	PC = HL
No. of Bytes	1 byte (Opcode of PCHL)
Machine Cycles	1 (OF)

Algorithm	PC ← HL
Flags	No flags are modified.
Addr. Mode	Implied addressing mode
T-states	6

Description	• The contents of H and L registers are transferred to program counter. • The H contents to high order 8 bits and L contents to low order 8 bits of program counter. • This instruction is equivalent to a 1 byte unconditional JUMP instruction, provided the address of JUMP is specified by the HL register pair. The program sequence is transferred to address specified by the HL register pair.

Example: PCHL	 Before Execution After Execution S Z AC P CY A B D H SP PC F C E L 12 36 4156 1236 The contents of register pair HL are copied to PC **Fig. 8.3.3 : PHCL**

- The PC gives address on the low order and high order address bus.
- It requires 6T states. The 8085 reads the opcode of PCHL and then decodes it. The opcode of PCH

is E9H.

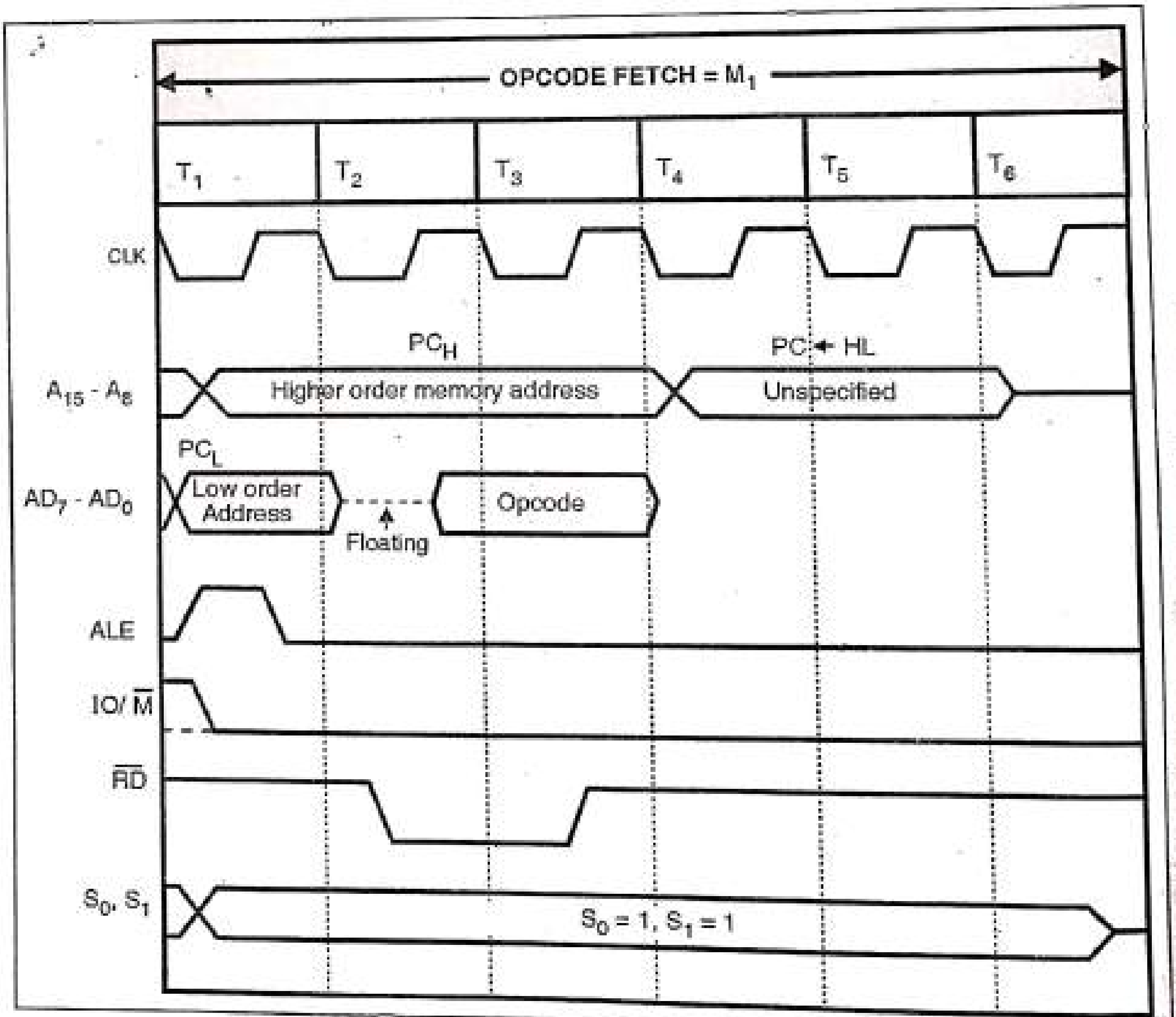


Fig. 8.3.4 :Timing diagram of PCHL

4. CALL Address

Mnemonic	CALL Address

Operation	(SP – 1) = PC_H
	(SP – 2) = PC_L
	SP = SP – 2
	PC = Address

No. of Bytes	3 bytes **First byte :** Opcode of CALL. **Second byte :** Low order byte of the address. **Third byte :** High order byte of the address.
Machine Cycles	5 (OF + MR + MR + MW + MW)

Algorithm	(SP – 1) ← PC_H
	(SP – 2) ← PC_L
	SP ← SP – 2
	PC ← Address

Flags	No flags are affected.
Addr. Mode	Implied addressing mode.
T-states	18 (6 + 3 + 3 + 3 + 3)

Description	• This instruction is used to transfer the program control to a subprogram or subroutine. • i.e. whenever this instruction is executed the program control is transferred to the address specified in the instruction. • The current contents of the PC are pushed onto the stack i.e. the address of the next instruction is pushed onto the stack. • The Stack pointer is decremented by 2. • After the execution of the subroutine programmer can transfer the program control back to the calling program. To do this the processor has to remember the address of the next instruction after the CALL instruction. Processor saves this address on to the stack when the stack CALL instruction is executed.
Example CALL C200	• Suppose this instruction is stored at location. C006 OPCODE of CALL C007 00 C008 C2 C009 next instruction • When this instruction is executed, program counter contents C009 will be stored on to the stack and microprocessor will load the PC with C200 H and starts executing instructions from C200 onwards.

Note : The Stack is a part of read/ write memory set aside for storing intermediate data and addresses.

- 5 machine cycles are required. To execute the CALL address instruction. They are :

(1) **OPCODE fetch :** The program counter places address on the lower order address bus and the higher order address bus. This cycle is used to read OPCODE of CALL instruction. The opcode of the CALL instruction is CD H. The address for this machine cycle is given by PC. The PC is then incremented by 1. This machine cycle requires 6 T-states.

(2) **Memory read :** The program counter places address on the lower order address bus and the higher order address bus. The lower order byte of the address specified is read using PC. The PC is then incremented by one.

(3) **Memory read :** The program counter places the address on the lower order address bus and the higher order address bus. The higher order byte of the address specified is read using PC. The PC is then incremented by one.

(4) **Memory write :** The stack pointer places the address on the lower order address bus and the

higher order address bus. This machine cycle is used to store higher order byte of the program counter contents on to stack. The Stack pointer is decremented by 1.

(5) **Memory write :** The stack pointer places the address on the lower order address bus and higher order address bus. This machine cycle is used to store lower order byte of the program counter contents on to stack.

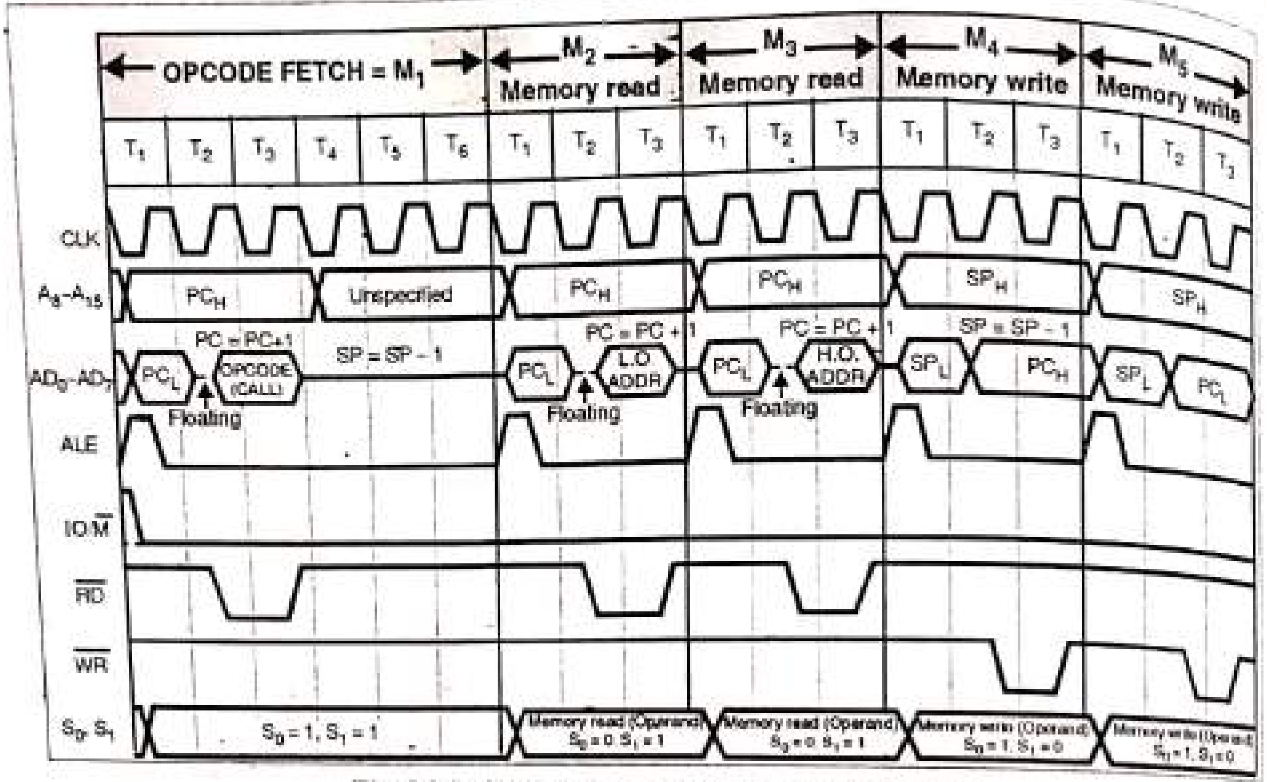


Fig. 8.3.5 : Timing diagram of CALL instruction

5. **Conditional CALL Instructions**

Mnemonic	Cond. address .
Operation	(SP − 1) = PC_H (SP − 2) = PC_L PC = Address Else PC = PC + 3
No. of Bytes	3 bytes **First byte :** Opcode of Instruction **Second byte :** Low order byte of the address **Third byte :** High order byte of the address
Machine Cycles	If condition is false : 2 (OF + MR) If condition is true : 5 (OF + MR + MR + MW + MW)

Algorithm	If condition true (SP − 1) ← PC_H (SP − 2) ← PC_L PC ← Address Else PC ← PC + 3
Flags	No flags are affected . Flags are only checked.
Addr. Mode	Implied addressing mode
T-states	If condition is false : 9 (6 + 3 If condition is true : 18 (6 + 3 + 3 + 3 + 3)

Description	• In conditional CALL instruction, when the condition is true, then a CALL at NEW address is made. • If the condition is not satisfied then the instruction that after the CALL instruction is satisfied . • Before Call, the address of the instruction after the CALL instruction is stored on the Stack and the Stack pointer is decremented by 2.

Example CC C200	• CALL if carry flag is set, the program written from address C200 onwards will be executed. If carry flag is reset microprocessor will execute next instruction after CC C200. • Fig. 8.3.6 shows the timing diagram for condition table .

Table below shows the possible condition for calls .

Instruction code	Description	Condition for CALL
CZ	CALL on zero	ZF = 1
CNZ	CALL if not zero	ZF = 0
CP	CALL if positive	SF = 0
CM	CALL on minus	SF = 1
CPO	CALL if parity odd	PF = 0
CPE	CALL if parity even	PF = 1
CC	CALL if carry	CF = 1
CNC	CALL if no carry	CF = 0

Remember there is no CALL on auxiliary carry flag.

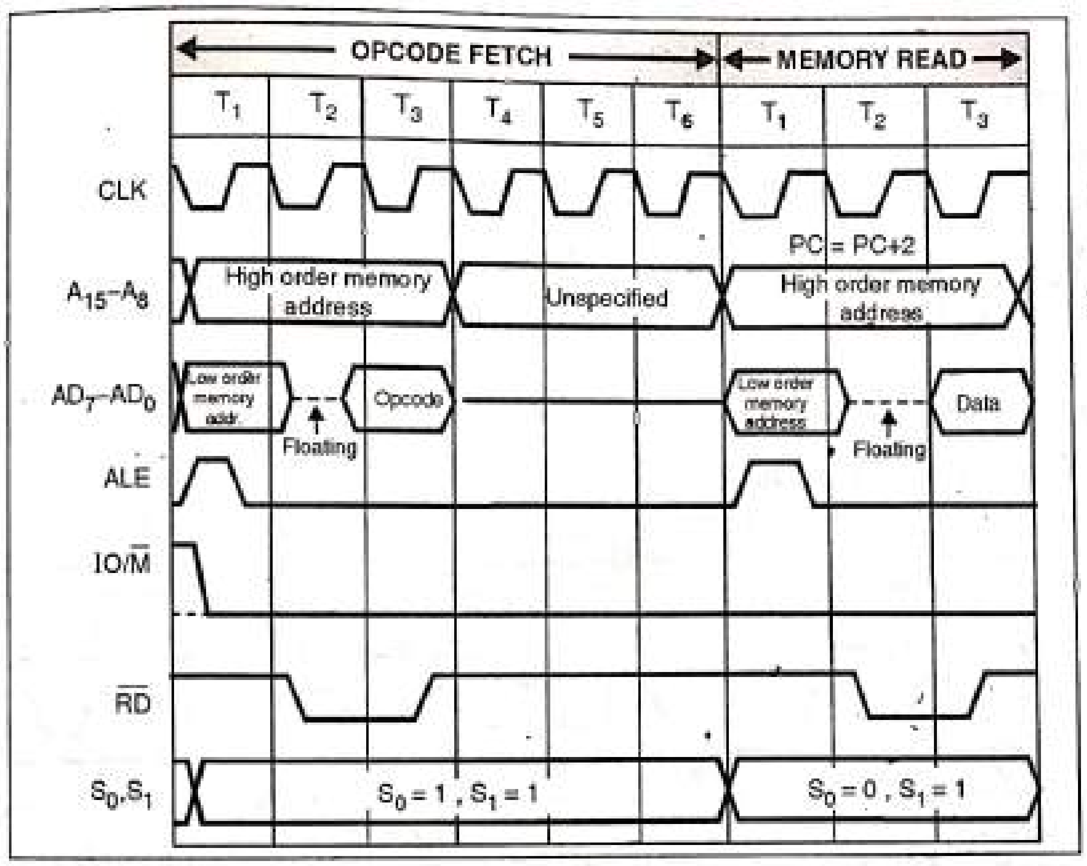


Fig. 8.3.6 : Timing diagram of C condition

6. **RET**

Mnemonic	RET
Operation	$PC_L = (SP)$ $PC_H = (SP + 1)$ $SP = SP + 2$
No. of Bytes	1 byte Opcode of RET
Machine Cycles	3 (OF + MR + MR)

Algorithm	$PC_L \leftarrow (SP)$ $PC_H \leftarrow (SP + 1)$ $SP \leftarrow SP + 2$
Flags	No flags are affected .
Addr. Mode	Indirect addressing mode
T-states	10 (4 + 3 + 3)

Description	• When this instruction is executed program control is transferred from the subroutine to the calling program. • The return address is popped / taken from stack (where the call instruction has stored its PC contents i.e. return address) this address is loaded in PC and the program execution begins at address taken from stack. • Thus, the program control is transferred to the next instruction after CALL in the main program.
Example RET	• Suppose the CALL C200 instruction is written at C006 and is executed by microprocessor. The microprocessor will call the subroutine .It will store the return address at C7FE and C7FD, start executing instructions from C200 onwards. At C209, RET instruction is present. When RET instruction is executed by microprocessor, it will take return address from stack (C7FD and C7FE) and load in program counter. So the next instruction executed will be from C009.

- This instruction requires 3 machine cycles . They are :

(1) **OPCODE fetch :** The program counter places address on the lower order address bus and the higher order address bus. This cycle is used to read OPCODE of RET instruction. The opcode of the RET instruction is C9 H. The address for this machine cycle is given by PC. The PC is then incremented by 1.

(2) **Memory read :** In this machine cycle data is read from the memory location whose address is pointed by the stack pointer. The data read is the low order byte of the address where the program control is to be transferred. The SP is then incremented by one.

(3) **Memory read :** In this machine cycle data is read from the memory location whose address is pointed by the stack pointer. The data read is the higher order byte of the address where the program control is to be transferred. The SP is then incremented by one.

The timing diagram of RET instruction is as shown in Fig. 8.3.7.

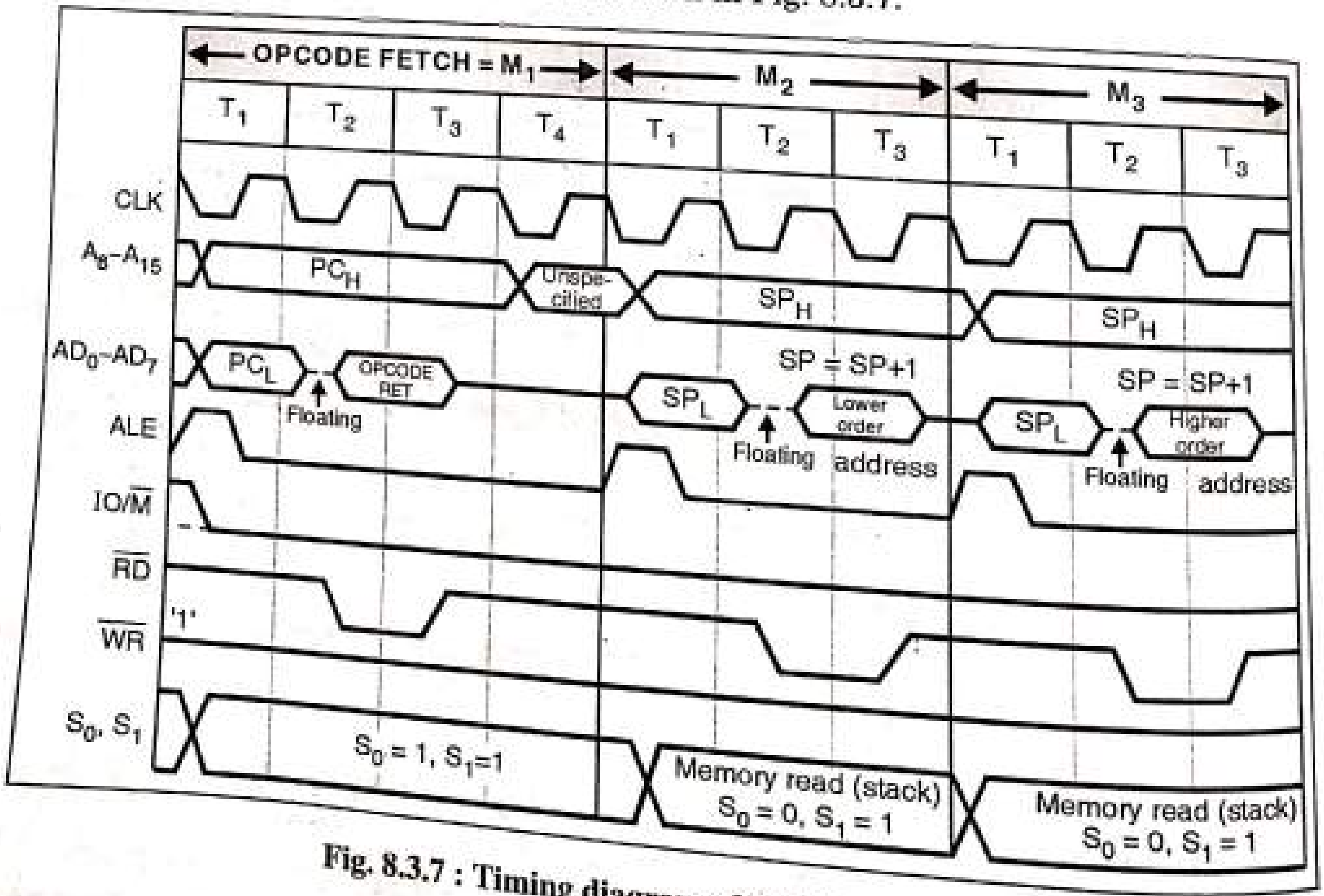


Fig. 8.3.7 : Timing diagram of RET instruction

7. Conditional RET Instructions

Mnemonic	R cond.

Operation	PC$_L$ = (SP)
	PC$_H$ = (SP + 1)
	SP = SP + 2
	Else
	PC = PC + 1

No. of Bytes	1 byte Opcode of RET
Machine Cycles	If condition is false, 1 (OF) If condition is true, 3 (OF + MR + MR)

Algorithm	PC$_L$ ← (SP)
	PC$_H$ ← (SP + 1)
	SP ← SP + 2
	Else PC ← PC + 1

Flags	No flags are modified, flags are only checked.
Addr. Mode	Indirect addressing mode
T-states	If condition is false, 6 If condition is true, 12 (6 + 3 + 3)
Description	• In conditional RET instruction, when the condition is true, then the control is transferred to the main program.

Table below shows the possible condition for returns .

Instruction code	Description	Condition for RET
RZ	RET on zero	ZF = 1
RNZ	RET if not zero	ZF = 0
RP	RET if positive	SF = 0
RM	RET on minus	SF = 1
RPO	RET if parity odd	PF = 0
RPE	RET if parity even	PF = 1
RC	RET if carry	CF = 1
RNC	RET if no carry	CF = 0

Remember there is no return on auxiliary carry flag.

The timing diagram of R condition, when condition is satisfied is shown in Fig.8.3.8.

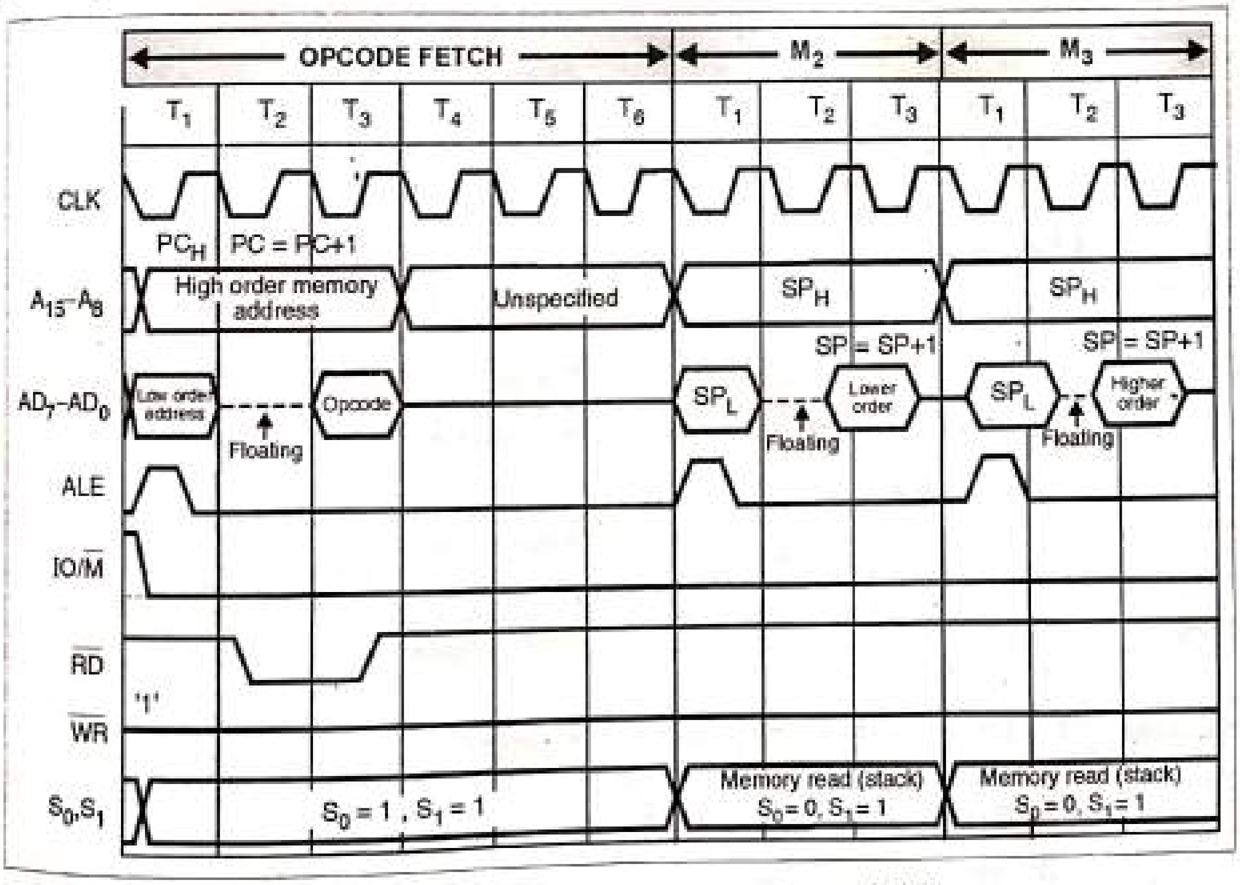


Fig. 8.3.8 : Timing diagram of R condition satisfied

This instruction requires opcode fetch machine cycle. Fig. 8.3.9 gives the timing diagram of this instruction when condition is not satisfied.. It requires 6T states. This instruction transfers the program control to the instruction that is written after this instruction. The program counter places address on high and low order address bus. The opcode of R condition is read into the microprocessor from the addressed memory location. The program counter is incremented by one.

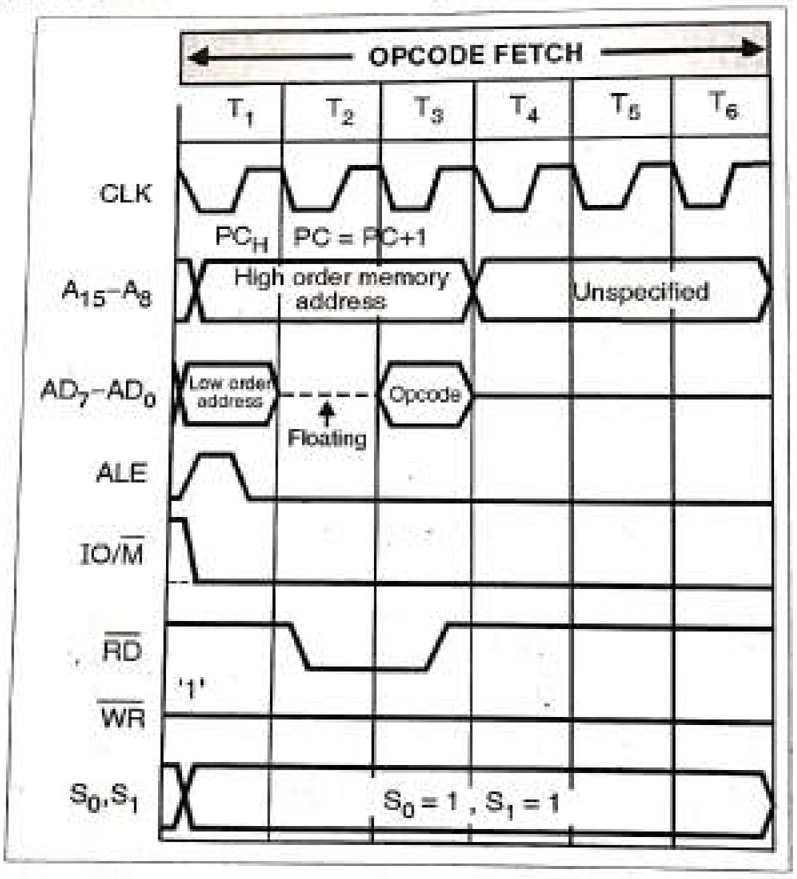


Fig. 8.3.9 : Timing diagram of R condition not satisfied.