Microprocessor & Interfacing
Lecture 21
Branching Instructions

PARUL BANSAL
ASST PROFESSOR
ECS DEPARTMENT
DRONACHARYA COLLEGE OF ENGINEERING

Contents

- Branching Instructions
- Jump Conditionally
- Control Instructions
- Summary

Branching Instructions

- Alter the normal sequential flow
- Alter either unconditionally or conditionally

Opcode	Operand	Description
JMP	16-bit address	Jump unconditionally

- The program sequence is transferred to the memory location specified by the 16-bit address given in the operand.
- **Example:** JMP 2034 H.

Opcode	Operand	Description
Jx	16-bit address	Jump conditionally

- The program sequence is transferred to the memory location specified by the 16-bit address given in the operand based on the specified flag of the PSW.
- **Example:** JZ 2034 H.

Jump Conditionally

Opcode	Description	Status Flags
JC	Jump if Carry	CY = 1
JNC	Jump if No Carry	CY = 0
JP	Jump if Positive	S = 0
JM	Jump if Minus	S = 1
JZ	Jump if Zero	Z = 1
JNZ	Jump if No Zero	Z = 0
JPE	Jump if Parity Even	P = 1
JPO	Jump if Parity Odd	P = 0

Branching Instructions

Opcode	Operand	Description
CALL	16-bit address	Call unconditionally

- The program sequence is transferred to the memory location specified by the 16-bit address given in the operand.
- Before the transfer, the address of the next instruction after CALL (the contents of the program counter) is pushed onto the stack.
- **Example:** CALL 2034 H.

Opcode	Operand	Description
RET	None	Return unconditionally

- The program sequence is transferred from the subroutine to the calling program.
- The two bytes from the top of the stack are copied into the program counter, and program execution begins at the new address.
- **Example:** RET.

Control Instructions

Opcode	Operand	Description
NOP	None	No operation

- No operation is performed.
- The instruction is fetched and decoded but no operation is executed.
- **Example:** NOP

Opcode	Operand	Description
HLT	None	Halt

- The CPU finishes executing the current instruction and halts any further execution.
- An interrupt or reset is necessary to exit from the halt state.
- **Example:** HLT

Control Instructions

Opcode	Operand	Description
DI	None	Disable interrupt

- The interrupt enable flip-flop is reset and all the interrupts except the TRAP are disabled.
- No flags are affected.
- **Example:** DI

Opcode	Operand	Description
EI	None	Enable interrupt

- The interrupt enable flip-flop is set and all interrupts are enabled.
- No flags are affected.
- This instruction is necessary to re-enable the interrupts (except TRAP).
- **Example:** EI

Summary – Data transfer

- MOV: Move
- MVI: Move Immediate
- LDA: Load Accumulator Directly from Memory
- STA: Store Accumulator Directly in Memory
- LHLD: Load H & L Registers Directly from Memory
- SHLD: Store H & L Registers Directly in Memory

Cont..

- An 'X' in the name of a data transfer instruction implies that it deals with a register pair (16-bits);
- LXI: Load Register Pair with Immediate data
- LDAX: Load Accumulator from Address in Register Pair
- STAX: Store Accumulator in Address in Register Pair
- XCHG: Exchange H & L with D & E
- XTHL: Exchange Top of Stack with H & L

Cont..

Add, Subtract, Increment / Decrement data in registers or memory.

- ADD Add to Accumulator
- ADI Add Immediate Data to Accumulator
- ADC Add to Accumulator Using Carry Flag
- ACI Add Immediate data to Accumulator Using Carry
- SUB Subtract from Accumulator
- SUI Subtract Immediate Data from Accumulator
- SBB Subtract from Accumulator Using Borrow (Carry) Flag
- SBI Subtract Immediate from Accumulator Using Borrow (Carry) Flag
- INR Increment Specified Byte by One
- DCR Decrement Specified Byte by One
- INX Increment Register Pair by One
- DCX Decrement Register Pair by One
- DAD Double Register Add; Add Content of Register Pair to H & L Register Pair

Summary Logical Group

This group performs logical (Boolean) operations on data in registers and memory and on condition flags.

- These instructions enable you to set specific bits in the accumulator ON or OFF.
- ANA: Logical AND with Accumulator
- ANI: Logical AND with Accumulator Using Immediate Data
- ORA: Logical OR with Accumulator
- OR: Logical OR with Accumulator Using Immediate Data
- XRA: Exclusive Logical OR with Accumulator
- XRI: Exclusive OR Using Immediate Data

Cont..

The Compare instructions compare the content of an 8-bit value with the contents of the accumulator;

- CMP Compare
- CPI Compare Using Immediate Data
- The rotate instructions shift the contents of the accumulator one bit position to the left or right:
- RLC Rotate Accumulator Left
- RRC Rotate Accumulator Right
- RAL Rotate Left Through Carry
- RAR Rotate Right Through Carry
- Complement and carry flag instructions:
- CMA Complement Accumulator
- CMC Complement Carry Flag
- STC Set Carry Flag

Summary - Branch Group

Unconditional branching

- JMP Jump
- CALL Call
- RET Return

Conditions

- NZ Not Zero (Z = 0)
- Z Zero (Z = 1)
- NC No Carry (C = 0)
- C Carry (C = 1)
- PO Parity Odd (P = 0)
- PE Parity Even (P = 1)
- P Plus (S = 0)
- M Minus (S = 1)

Conditional branching

Summary - Stack

- PUSH: Push Two bytes of Data onto the Stack
- POP: Pop Two Bytes of Data off the Stack
- XTHL: Exchange Top of Stack with H & L
- SPHL: Move content of H & L to Stack Pointer

I/O instructions

- IN: Initiate Input Operation
- OUT: Initiate Output Operation

Summary - Machine Control instructions

- EI Enable Interrupt System
- DI Disable Interrupt System
- HLT Halt
- NOP No Operation