23VLS1401: Microcontroller and Computer architecture

Lecture 1 (U5)

Data Transfer instructions and programming for Microcontroller 8051

A presentation by

Dr. Shubhangi Rathkanthiwar

Professor

Department of Electronics Engineering, YCCE, Nagpur, India

Session objectives

- To overview instructions related to the process of data transfer in Microcontroller 8051
- To develop the programming technique in assembly language for given problem statement, store the source data, execute the program and observe the result in destination register or memory location.

Data Transfer instructions

- **MOV**
- **MOVX**
- **MOVC**
- **PUSH and POP**
- **XCH**

Data transfer instructions

Opcode	Operand	Operation	Addressing mode
MOV	A,#n	Copy immediate data ‘n’ to accumulator	Immediate
MOV	A,Rr	Copy data from register Rr to accumulator	Register
MOV	Rr,A	Copy data from accumulator to register Rr	Register
MOV	Rr,#n	Copy immediate data ‘n’ in register Rr	Immediate
MOV	DPTR,#nn	Copy 16 bit to DPTR	Immediate

Data transfer instructions

Opcode	Operand	Operation	Addressing mode
MOV	A,#n	Copy immediate data ‘n’ to accumulator	Immediate
MOV	A,Rr	Copy data from register Rr to accumulator	Register
MOV	Rr,A	Copy data from accumulator to register Rr	Register
MOV	Rr,#n	Copy immediate data ‘n’ in register Rr	Immediate
MOV	DPTR,#nn	Copy 16 bit to DPTR	Immediate

Caution:

Immediate data can not be a destination

All numbers must start with decimal number 0-9. Otherwise assembler will treat it as a label

Register to register data transfer is not possible

Data transfer instructions

Opcode	Operand	Operation	Addressing mode
MOV	A,add	Copy data from direct address ‘add’ to A	Direct
MOV	add,A	Copy data from A to direct address ‘add’	Direct
MOV	Rr,add	Copy data from direct address to register Rr	Direct
MOV	add,Rr	Copy data from register Rr to direct address add	Direct
MOV	add,#n	Copy data ‘n’ to direct address add	Direct
MOV	add1,add2	Copy data from direct address ‘add2’ to direct address ‘add1’	Direct

Data transfer instructions

Opcode	Operand	Operation	Addressing mode
MOV	A,add	Copy data from direct address ‘add’ to A	Direct
MOV	add,A	Copy data from A to direct address ‘add’	Direct
MOV	Rr,add	Copy data from direct address to register Rr	Direct
MOV	add,Rr	Copy data from register Rr to direct address add	Direct
MOV	add,#n	Copy data ‘n’ to direct address add	Direct
MOV	add1,add2	Copy data from direct address ‘add2’ to direct address ‘add1’	Direct

Caution:

Address above 7F do not exist

Moving data from a Direct address to itself is invalid

Data transfer instructions

Opcode	Operand	Operation	Addressing mode
MOV	@Rp,#n	Load immediate data ‘n’ to the address in Rp	Indirect
MOV	@Rp,add	Copy contents in address add to the address in Rp	Indirect
MOV	@Rp,A	Copy contents in accumulator to the address in Rp	Indirect
MOV	add, @Rp	Copy contents in address at Rp to address ‘add’	Indirect
MOV	A, @Rp	Copy contents in address at Rp to accumulator	Indirect

Caution:

1. A number in Rp must be a RAM address
2. Only R0 and R1 are used for indirect addressing

Data transfer instructions for external data moves

Opcode	Operand	Operation	Addressing mode
MOVX	A,@Rp	Copy the contents of external address in Rp to A	Indirect
MOVX	A,@DPTR	Copy the contents of external address in DPTR to A	Indirect
MOVX	@Rp,A	Copy the contents of A to external address in Rp	Indirect
MOVX	@DPTR,A	Copy the contents of A to external address in DPTR	Indirect

Data transfer instructions

Opcode	Operand	Operation
MOVC	A,@A+DPTR	Copy the code byte found at the ROM address formed by adding A and the DPTR to A
MOVC	A,@A+PC	Copy the code byte found at the ROM address formed by adding A and the PC to A
PUSH	addr	Increment SP and copy the data in specified address to the internal RAM whose address is given in SP
POP	addr	Copy the contents of internal RAM whose address is given in SP to the specified address

Exchange instructions

Opcode	Operand	Operation
XCH	A,Rr	Exchange data bytes between Rr and A
XCH	A,addr	Exchange data bytes between addr and A
XCH	A,@Rp	Exchange data bytes between A and address at Rp
XCHD	A,@Rp	Exchange lower order nibbles between A and address at Rp

Caution:

1. All exchanges are internal to the 8051
2. All exchanges use A
3. XCHD do not modify upper nibbles

Additional instructions required in developing ALP

Opcode	Operand	Operation
INC	A	Increment A
INC	Rr	Increment Rr
INC	addr	Increment contents at addr
INC	@Rp	Increment contents of memory location pointed by Rp
INC	dptr	Increment DPTR
DEC	A	Decrement A
DEC	Rr	Decrement Rr
DEC	addr	Decrement contents at addr
DEC	@Rp	Decrement contents of memory location pointed by Rp

JUMP instructions

- JC raddr
- JNC raddr
- JZ raddr
- JNZ raddr
- DJNZ Rn, raddr
- DJNZ add, raddr

Problem statement 1: WAP to load the data bytes 55H, 66H, 77H and 2501H in Accumulator, registers R0, R1 and DPTR respectively

ALP:

```
ORG 0000H
MOV A,#55H
MOV R0,#66H
MOV R1,#77H
MOV DPTR,#2501H
END
```

Problem statement 2: WAP to load the data bytes 55H, 66H, 77H and 88H in the internal RAM locations starting at 25H

ALP:

```
ORG 0000H
MOV R0,#25H
MOV @R0,#55H
INC R0
MOV @R0,#66H
INC R0
MOV @R0,#77H
INC R0
MOV @R0,#88H
END
```

Problem statement 3: A block of 5 data bytes is stored in the internal RAM locations starting at 25H. WAP to copy the block in external RAM locations starting at 2501H.

ALP:

```
ORG 0000H
MOV R0,#25H
MOV DPTR,#2501H
MOV R1,#05H
L1: MOV A,@R0
    MOVX @DPTR,A
    INC R0
    INC DPTR
DJNZ R1,L1
END
```

Problem statement 4: A block of 5 data bytes is stored in the internal RAM locations starting at 25H. WAP to copy the block in external RAM locations starting at 2501H in reverse order.

ALP:

```
ORG 0000H
MOV R0,#29H
MOV DPTR,#2501H
MOV R1,#05H
L1: MOV A,@R0
    MOVX @DPTR,A
    DEC R0
    INC DPTR
DJNZ R1,L1
END
```

Problem statement 5: A series of 10 16-bit temperature readings is stored in external RAM locations starting at 2501H. Higher order bytes of all temperature readings is constant. WAP to copy lower bytes from internal RAM locations starting at 25H.

ALP:

```
ORG 0000H
MOV R0,#25H
MOV DPTR,#2501H
MOV R1,#0AH
L1: MOVX A,@DPTR
    MOV @R0,A
    INC R0
    INC DPTR
    INC DPTR
DJNZ R1,L1
END
```

Thank you