

XMUT202 Digital Electronics

Tutorial 3: Assembly Language (Solution)

A. Assembly Language

1. With regard to high-level and low-level programming languages, answer the following questions
 - a. What are these programming languages? [2 marks]
 - b. Provide at least two examples of these programming languages? [2 marks]
 - c. What are the advantages and disadvantages of one to the other? [2 marks]

Answer

- a. The high-level programming languages provide high level of abstraction between language and hardware and the low-level programming languages' commands are very closely related to the hardware's actual operation.
 - b. C, Python, Javascript, LabView, Lua, C#, Java, Ada, Smalltalk, Swift, and Forth are examples of high-level programming languages and assembly languages are considered low-level often one line per instruction cycle with machine code is the lowest of all.
 - c. High-level programming languages are faster to program, easier to debug, more portable, and myriad other advantages. But on the other hand, low-level programming languages have fast execution, can be deterministic, affords understanding of hardware.
2. Compare the characteristics of CISC and RISC of the instruction set of the microprocessor. [2 marks]

Answer

CISC (Complex Instruction Set:

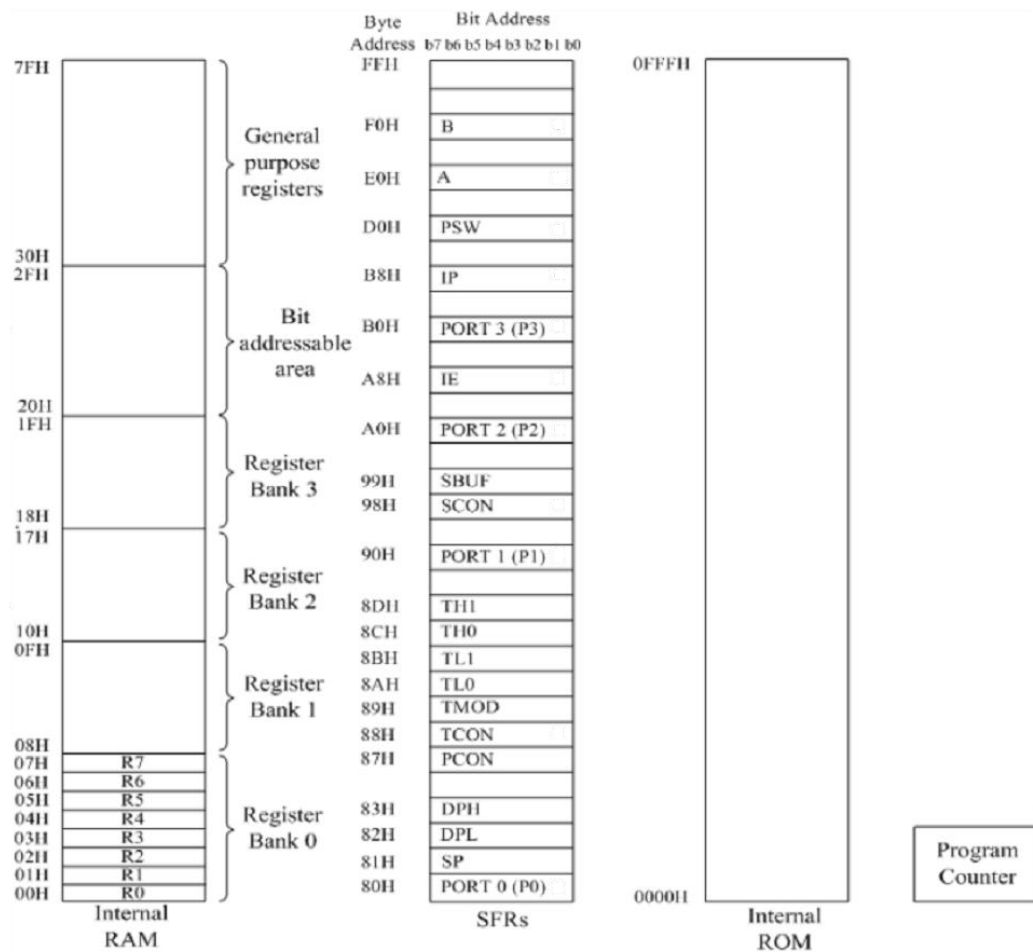
- Common in many early computers.
- Many detailed operations that allow for single instructions to have fine-grained control over the computer.
- Pentium, 8051, etc. are CISC
- Good for low-level programming, as fewer commands are needed.

RISC (Reduced Instruction Set:

- As high-level languages became prevalent in the 1980's and 1990's, there was less need for very specific processor instructions.

- Compilers could make use of a restricted (and fast!) instruction set to efficiently realise high-level programs.
- ARM processors are the most prevalent RISC processors today.
- Assembly language programs for CISC systems are often relatively human readable. RISC assembly language programs are significantly less so.

3. The diagram below shows the mapping of the register memory locations in the 8051 microcontrollers.



Which of the registers given above are:

a. Byte addressable. [2 marks]

b. Bit addressable. [2 marks]

Answer

c. Byte addressable registers are general registers (i.e. R0-R7), most of the SFRs, and general-purpose registers (i.e. memory address 0x30H-0x7FH)

d. Bit addressable registers are 0x20H - 0x2FH, SFRs such as: PORT 0, 1, 2 and 3, TCON, SCON,

IE, IP, PSW, A, and B.

4. What is the size of the following registers in the 8051 microcontrollers. [4 marks]
- Accumulator (A).
 - Program Counter (PC).
 - Data Pointer (DTPR).
 - Stack Pointer (SP).

Answer

- Accumulator (A) – 8 bits.
- Program Counter (PC) – 16 bits.
- Data Pointer (DTPR) – 16 bits.
- Stack Pointer (SP) – 8 bits.

5. Describe the addressing mode of the following instructions in the 8051 microcontrollers. [4 marks]

| Instruction | Addressing Mode |
|-------------|-----------------|
| MOV A, B | |
| MOV 30H, A | |
| ADD A, @R0 | |
| ADD A, #80H | |

Answer

The addressing modes of those instructions are listed on the table below.

| Instruction | Addressing Mode |
|-------------|--------------------|
| MOV A, B | Register |
| MOV 30H, A | Direct |
| ADD A, @R0 | Indirect |
| ADD A, #80H | Immediate/Constant |

6. Describe at least two examples of bit-addressing instruction sets in the 8051 microcontrollers. [2 marks]

Answer

Two examples of bit addressing instruction sets are:

```
SETB P1.6      ;Set bit 7 of Port 1 into 1
CLR P2.2       ;Clear bit 3 of Port 2 into 0
CPL P0.0       ;If bit 0 of Port 0 was 1, it is now 0, or vice versa
```

B. Programming with Basic Assembly Language

B.1. Memory Addressing Instructions

1. Describe the following assembly language program for immediate addressing mode program in the 8051 microcontrollers. [2 marks]

Program:

```
ADD A, #030H
```

Solution

This program adds 8-bit value of 0x30H to the accumulator register (which is an 8-bit register).

2. Describe the following assembly language program for register addressing mode program in the 8051 microcontrollers. [2 marks]

Program:

```
MOV A, P3
```

Solution

The program initiates a transfer the contents of Port 3 to the accumulator.

3. Describe the following assembly language program for direct addressing mode program in the 8051 microcontrollers. [2 marks]

Program:

```
MOV A, 020H
```

Solution

This program transfers the contents of RAM location 0x20H to the accumulator.

4. Describe the following assembly language program for indirect addressing mode program in the 8051 microcontrollers. [2 marks]

Program:

```
MOV @R0, A
```

Solution

This program stores the content of accumulator into the memory at a location pointed to by register R0. The R0 could have an 8-bit address, such as 0x60H.

5. Describe the following assembly language program for indirect addressing mode program from a 16-bit data pointer register (DPTR) to Accumulator A in the 8051 microcontrollers. [2 marks]

Program:

```
MOVX A, @DPTR
```

Solution

This program transfers the contents from the memory location pointed to by DPTR into the accumulator. DPTR could have a 16-bit address, such as 0x1234H.

6. Describe the following assembly language program for immediate addressing mode program to a 16-bit data pointer register (DPTR) in the 8051 microcontrollers. [2 marks]

Program:

```
MOV DPTR, #0FE00H
```

Solution

This program moves 16-bit data constant 0xFE00H into the 16-bit Data Pointer Register.

7. Describe the following assembly language program for addressing mode program in the 8051 microcontrollers. [2 marks]

Program:

```
GoBack:  DEC A
          JNZ GoBack
```

Solution

In the first line of the code, the program decrements Accumulator A.

The second line of the code is about a decision branching that evaluate the value of A.

- If A is not zero, loop back to the address labelled as GoBack.
- If A is 0, then the program will skip the loop to GoBack.

8. Describe the following assembly language program for addressing mode program in the 8051 microcontrollers. What is the scope of ACALL for switching the instruction. [2 marks]

Program:

```
ACALL      PORT_INIT
           NOP
           NOP
           NOP
PORT_INIT: MOV P0, #0FH
```

Solution

This program would call an address labelled as PORT_INIT. The PORT_INIT address should be located within 2k bytes.

9. Describe the following assembly language program for addressing mode program in the 8051 microcontrollers. What is the equivalent function in C programming language. [2 marks]

Program:

```
LCALL TIMER_INIT
      NOP
      NOP
TIMER_INIT: ORL TMOD, #01H
```

Solution

The program TIMER_INIT is a subroutine. The TIMER_INIT address (i.e. 16-bits long) is specified as the operand. In C programming, this will be a function call i.e. Timer_Init().

B.2. Bit-Addressing Instructions

10. Describe the following assembly language program for the bit rotating op code-based program in the 8051 microcontrollers. [2 marks]

Program:

```
MOV A, #11000011B
RL A
RR A
```

Solution

The code line-by-line comments for the bit rotating op code-based program are listed below.

```
MOV A, #11000011B      ;A=11000011B
RL A                   ;A=10000110B
RR A                   ;A=01000011B
```

11. Describe the following assembly language program for the bit rotating op-code with carry-based program in the 8051 microcontrollers. [4 marks]

Program:

```
MOV A, #11000011B
RLC A
RLC A
RRC A
RRC A
```

Solution

The code line-by-line comments for the bit rotating op-code with carry-based program are listed below.

```
MOV A, #11000011B      ;A=10000000B
RLC A                   ;A=00000000B and Carry=1
RLC A                   ;A=00000001B and Carry=0
RRC A                   ;A=00000000B and Carry=1
RRC A                   ;A=10000000B
```

12. Describe the following assembly language program for bit setting, clearing and complementing program in the 8051 microcontrollers. [3 marks]

Program:

```
MOV P1, #10101111B
SETB P1.6
CLR P1.2
CPL P1.0
```

Solution

The code line-by-line comments program for bit setting, clearing and complementing program are listed below.

```
MOV P1, #10101111B    ;P1=10101111B
SETB P1.6             ;P1=11101111B
CLR P1.2              ;P1=11101011B
CPL P1.0              ;P1=10101010B
```

13. Describe the following assembly language program for bit swapping program in the 8051 microcontrollers. [1 mark]

Program:

```
MOV A, 11111010B
SWAP A
```

Solution

The code line-by-line comments for bit swapping program are listed below.

```
MOV A, 11111010B    ;A=11111010B
SWAP A               ;A=10101111B
```

14. Describe the following assembly language program for AND and OR bit masking program in the 8051 microcontrollers. [5 marks]

Program:

```
MOV P2, #10101111B
ANL C, P2.0
ORL C, P2.7
ANL P2.6, C
ANL P1, #00000001B
ORL P1, #00000010B
```

Solution

The code line-by-line comments for AND and OR bit masking program are listed below.

Program:

```
MOV P2, #10101111B    ;P2=10101111B
ANL C, P2.0           ;As P2.0 is 0, so Carry=0
```

```

ORL C,P2.7          ;As P2.7 is 1, so Carry=1
ANL P2.6,C          ;C=1 and P2.6=1,so bit P2.6=1, P2=11101111B
ANL P1,#00000001B  ;P1=00000001B
ORL P1,#00000010B  ;P1=#00000011B

```

15. Describe the following assembly language program for bit conditional branching-based program in the 8051 microcontrollers. [6 marks]

Program:

```

MOV A, #80H
RLC A
JC LOOP
JNC LOOP
LOOP: JB P0.2,LOOP
      JNB P1.4,LOOP
      JBC P0.2,LOOP

```

Solution

The code line-by-line comments for bit conditional branching-based program are listed below.

Program:

```

MOV A, #80H ;A=10000000H
RLC A      ;A=00000000H and Carry=1
JC LOOP   ;Program will not jump to LOOP if C=1
JNC LOOP  ;Program will not jump to LOOP if C=0
LOOP: JB P0.2,LOOP ;If P0.2=1 program jump to LOOP
      JNB P1.4,LOOP ;If P1.4=0 program jump to LOOP
      JBC P0.2,LOOP ;If P0.2=1 program jump to LOOP, P0.2=0

```