By: Ali Hassan Soomro
BSCS from UBIT, University of Karachi
DAE in Electronics from SBTE
Facebook: www.facebook.com/AliiHassanSoomro
Gmail: alisoomro666@gmail.com
Youtube: www.youtube.com/Prgramology

Assembly Language Theory Important Questions

Q1. Difference among Machine, assembly and high level language

Machine Language	Assembly Language	High level Language
It is the native language of machine	It is low level computer programming language which means it is more close to machine	It is a computer programming language that is more close to human.
Consists of 0's and 1's	Consists of a symbolic representation (i.e. Mnemonics)	Consists of English like statements
Known as machine code	Known as assembly code and also asm	There are many high level level languages e.g. c, java and they are called by their names.

Q2. Difference among Assembler, Compiler and Interpreter

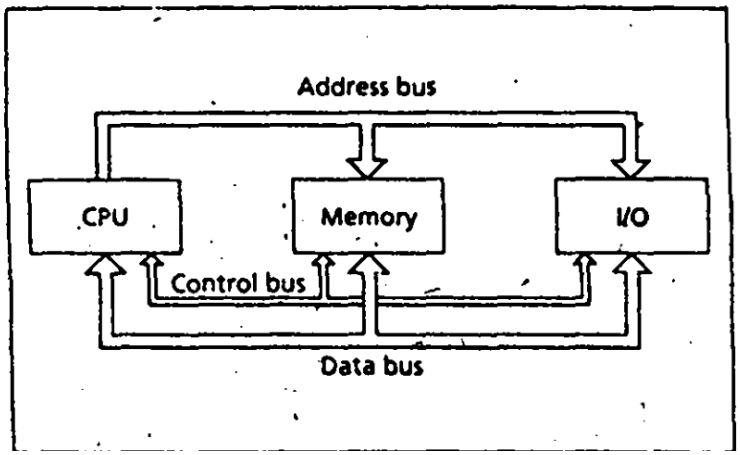
Assembler	Compiler	Interpreter
Translate assembly code to machine code	Translate the entire high level language code to machine code	Translate the high level code line by line (single instruction at a time) and then convert to machine code

Q3. What are Buses and their types? Also draw the diagram

Buses are the wires that connect the components. (CPU, memory and input/output devices) There are three types of wires:

Address Bus: holds the address of memory location, address signals
Control Bus: Inform the memory to perform read operation, control signals
Data Bus: Holds the actual data, data signals

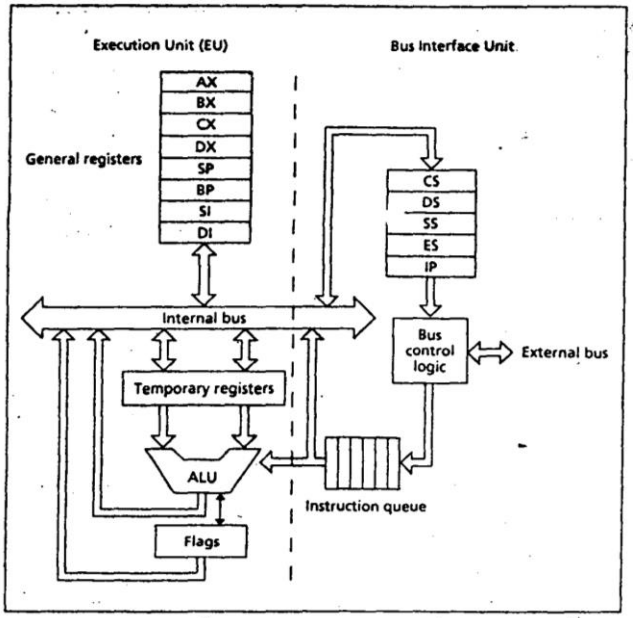
Buses Diagram:



Q4. Difference between Execution unit and bus interface unit? Draw diagram

Execution unit (EU) executes the instructions, it contains a circuit called the arithmetic and logic unit (ALU) and registers. While, the bus Interface unit (BIU) facilitates communication between the EU and the memory or I/0 circuits.

Diagram:



Q5. How CPU executes the instruction? What is fetch-execute-cycle?

CPU executes the instruction with fetch-execute-cycle as;

Fetch

1. Fetch an instruction from memory.
2. Decode the instruction to determine the operation.
3. ·Fetch data from memory if necessary.

Execute

1. Perform the operation on the data.
2. Store the result in memory if needed.

Q6. Suppose a processor uses 20 bits for an address. How many memory bytes can be accessed?

Solution: A bit can have two possible values, so In a 20-blt address there can be 2^{20} = 1,048,576 different values, with each value being the potential address of a memory byte. In computer terminology, the number 2^{20} is called 1 mega. Thus, a 20-bit address can be used to address 1 megabyte or 1 MB.

Q7. Difference between opcode and mnemonic?

Mnemonic is a symbolic representation e.g. AX, BX, ADD, MOV, DB and DW etc. While, opcode is a mnemonic that performs the operation e.g. MOV, ADD, SUB, PUSH etc

Q8. Difference between memory location and register?

Memory locations are locations of external main memory (RAM). While, register are the fastest location inside the microprocessor (ax, bx, cx, dx, ss, cs, si, and di etc)

Q9. Calculate the physical address of A4FB:4872

Step 1:

Multiply A4FB with 10h that is equal to A4FBO

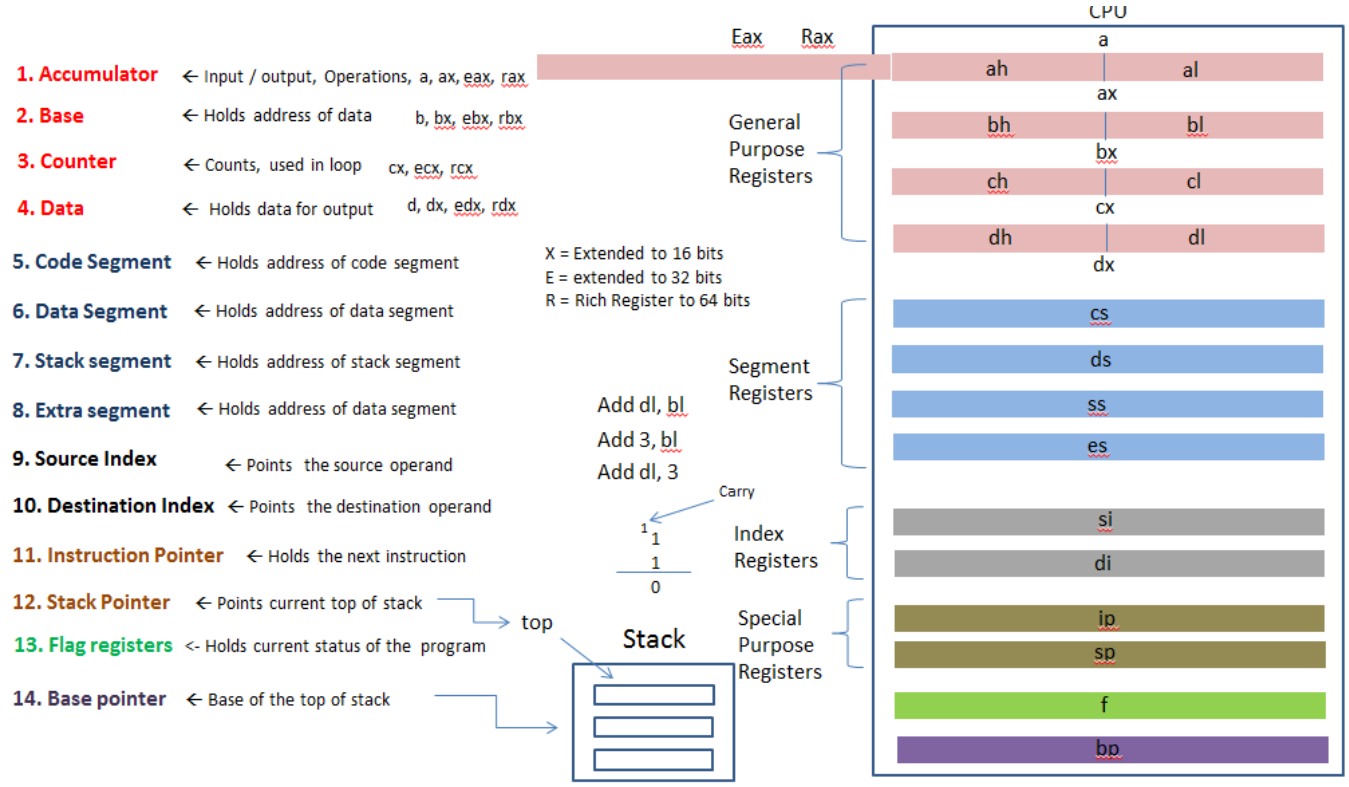
Step 2:

Add

A4FBOh
+ 4872h
A9822h (20-bit physical address)

<u>Q10. What are Registers, describe all registers?</u>

Registers are the fastest memory locations built into microprocessor.
Following are the 14 types of registers



Q11. What is flag register? Describe all bits?

Flag register is a register that contains the current state of the processor. Flag register consists of status flags and control flags.

				OF	DF	IF	TF	SF	ZF		AF		PF		CF

Status Flags: To handle the result of an operation.

1. Carry : CF (1 if carry out)
2. Parity : PF (1 if number of 1 bits is even)
3. Auxiliary : AF (1 if there is 3rd bit carry)
4. Zero: ZF (1 if result is 0)
5. Sign: Flag (1 if result is negative)
6. Overflow flag (1 if result if out of range of destination location)

Controls Flags: To Control the operations of CPU

1. Trap: TF (1 if debugging is used)

2. Direction: DF (1 if reverse function is called)
3. Interrupt: IF (1 if interruption is called)

Q12. What are Assembler directives?

Assembly directives are instructions that direct the assembler to do something. They are not converted to machine code**.** There are different directives in assembly language, some of them are;

- .model
- .stack
- .data
- .code
- Db, dw, dt

Q13. Describe Data representation in Assembly language programming?

In assembly, data can be represented by following ways

- 11011 Decimal
- 11011B BINARY
- 64223 DECIMAL
- -21843D DECIMAL
- OFFFFH Hexadecimal number

Q14. Draw the Basic Program structure?

```
;Title of the program
Dosseg
.model small
.stack 100h
.data
        ; variables are defined here
.code
Main proc     ; procedure start
        ; executable instructions
```

```
Main endp    ; procedure end
End main
```

Q15. Describe five logic instructions?

The five logic instructions are AND, OR, NOT, XOR, and TEST.

- The AND Instruction can be used to clear individual bits in the destination.
- The OR instruction Is useful In setting individual bits in the destination. It can also be used to test the destination for zero.
- The XOR instruction can be used to complement individual bits in the destination. It can also be used to zero out the destination.
- The NOT Instruction performs the one's complement operation on the destination.
- The TEST Instruction can be used to examine individual bits of the destination. For example, It can determine If the destination contains an even or odd number.

Q16. Perform the following operations

1. 10101010 AND 11110000
2. 10101010 OR 11110000
3. 10101010 XOR 11110000
4. NOT 10101010

Solutions:

1. 10101010
   AND 11110000
   = 10100000

2. 10101010
   OR 11110000
   = 11111010

3. 10101010
   XOR 11110000
   = 01011010

4. NOT 10101010
   = 01010101

Q17. Write some code to multiply the value of AX by 8. Assume that overflow will not occur.

Solution: To multiply by 8, we need to do three left shifts.

```
MOV  CL,3          ;number of shifts to do
SAL  AX,CL         ;multiply by 8
```

Q18. If CS = 7646H and IP =12BAH, show

1. The Logical Address
2. The offset Address
3. Physical Address

Solution:

Segment Address = CS = 7646H

Offset Address = IP = 12BAH

Logical Address= Segment: Offset
= 7646H : 12BAH

Physical Address:
First Multiply 7646H with 10h = 76460H
Now add 76460H and 12BAH = 7771Ah
So, physical address is 7771Ah

Q19. Define the following terms:

1. SAL and SHL
2. SHR
3. SAR
4. ROL
5. RCL and RCR

SAL AND SHL

SAL and SHL shift each destination bit "left one place. The most significant bit goes into CF, and a 0 Is shifted Into the least significant bit.

SHR

SHR shifts each destination bit right one place. The least significant bit goes into CF, and a 0 ls shifted into the most significant bit.

SAR

SAR operates like SHR, except that the value of the most significant bit is preserved.

ROL

ROL shifts each destination bit left one position; the most significant bit is rotated Into the least significant bit.

RCL and RCR

RCL and RCR operate like ROL and ROR, except that a bit rotated out goes Into CF, and the value of CF rotates into the destination.

Q20. What is STACK? What are stack instructions?

Stack

The stack is a temporary storage area that works on LIFO (last-in, first-out data) principle.

Stack instructions

The stack-altering instructions are PUSH, PUSHF, POP and POPF.
PUSH : adds a new top word to the stack
POP : removes the top word.
PUSHF : saves the FLAGS register on the stack
POPF : puts the stack top into the FLAGS register.

SP decreases by 2 when PUSH or PUSHF is executed, and It increases by 2 when POP or POPF is executed