COAL Assignment 1

21K-3153

(Q1) I In real address mode, 16 bit segment registers indicate base addresses of pre assigned memory areas named segments.

In protected mode, segment registers hold pointers to segment descriptor tables, which describe the location, length and access rights of the memory segment.

(Q1) II As a rule, memory access take more machine cycles because reading a single value from memory involves four steps

1. Place the address of the value on the address bus
2. Assert (change) the value of the processor's RD (read) pin
3. Wait one clock cycle for memory chips to respond
4. Copy data from data bus to destination operand

Each of these steps requires a single clock cycle. In contrast, register access requires fewer clock cycles.

(Q1) III (1) Industrial Robots

(2) Gaming Consoles

Date: M T W T F S S

Q

(a)(iv) Executing a machine instruction is known as An instruction execution cycle

The Instruction Pointer (IP) holds the address of the instruction to be executed

(1) The CPU fetches the instruction from the instruction queue and then increments the IP so it points to the next instruction

(2) The CPU decodes the instruction by looking at its binary bit pattern, which could reveal operands etc.

(3) If operands are involved, the CPU fetches operands from registers (general purpose registers like base, counter etc) and sometimes performs calculations (done in the Arithmetic Logic Unit or ALU)

(4) The CPU executes the instructions and update status flags such as Zero flag, Carry flag etc

(5) If the output operand was part of the instruction, the CPU stores the result of the instruction in the operand.

Instructions are decoded using the instruction decoder, which coordinates the ALU and floating point unit

Reading program instructions from memory requires addresses to be placed on the address bus.

Control bus coordinates transfer of data between different units of the CPU.

URBANE

Date M T W T F S S

(V) A machine cycle is when the processor implements an instruction. An instruction cycle is a process in the computer is given a program, understands it and executes it from memory. An instruction cycles can contain multiple machine cycles.

(VI) This is because a program written in Java is translated into Java byte code by a compiler which is executed by a program known as a Java Virtual Machine. (JVM) Since Java byte code is always the same and is always executed by the JVM on almost every computer, the Java code is nearly machine independent.

(VII) When the computer recognizes the ADD instruction, it looks up the value at the address 12FCBD10h, puts the value at one of ALU's input registers, put the value in AL at the other input register. Three parallel data buses are used for each of the two operands and output. Result is stored in a destination register if 12FCBD10h points to a register.

Date: M T W T F S S

Q1 VIII

(A) MOV 2020h, AL

Illegal as a value cannot be assigned to an immediate memory

(B) MOVZX AX, BX

Illegal since AX and BX are both the same size, thus BX values do not need to be zero extended.

(C) MOV AL, WORD PTR [EBX]

The WORD data type is 16 bits, while AL is only 8 bits, thus operands are not the same size. Illegal.

(D) ADD [AL], [CH]

Illegal as destination must be an index or base register.

(E) INC 1Ah

Illegal because add the answer has to be stored in a register and incrementing the value is not allowed.

Immediate.

~~Q1X~~

~~(Q1X) "AB" = 2 byte~~
~~ABh = 2 bits~~

~~20 DUP (10 DUP("AB"), 10 DUP(ABh), "AB", ABh)~~
~~20 bits, 20 bits, 2 bits, 2 bits~~

~~2 byte + 2 byte + 400 byte.~~
~~+ 2 + 400 bits + 400 bits + 40 bits + 40 bits~~

Q1 "AB" = 16 bits , ABh = 16 bits

20 DUP (10 DUP("AB"), 10 DUP(ABh), "AB", ABh)

160, 160, 16, 16

3200 + 3200 + 320 + 320 + 16 + 16 =

7072 bits → 884 bytes → 374h bytes

Q1 X

(A) MOV AX, 8F7Ah
ADD AX, 7AF8h ;

C	O	S	Z
1	0	0	0

8F7A
7AF8
10A72 ~~10738~~

(B) MOV BX, 0FA77h
INC BX

C	O	S	Z
0	0	1	0

FA77
1
FA78

Q2

Seg x 10h + 0ff

I Segment : 560Fh
Offset : 53D9h
Real Address : ?

E = 14h
+ D = 13
27

560F
x 10
560F0
+ 53D9
5B4B9

11011
1B

Real Address = 5B4B9h

II Segment = 0893h
Offset = ?
Real Address = BC893h
Segment x 10 = 08930h

BC893h
− 8930
B3F63

F=15

18h = 24d

10 = 16
11 = 17
12 = 18
13
14
15
16
17
18

OFFSET = B3F63h

14
− 9
15 = F

III Segment = ?

Offset = 50ADh

Real Address = ED32Dh

```
  2
 ED32Dh
-  50AD
 E8280
```

~~E8280/10h = E822h → segment~~

E8280h / 10 = E828h → segment

(03)

(I)

```
.data
A BYTE 10h
B BYTE 5h
.code
mov AX, A        mov AX, A
mov BX, B        xchg AX, B
mov              mov A, AX
```

Q3 II

.data

array1 WORD 20h, 19h, 18h, 17h, 16h, 15h, 14h, 13h, 12h, 11h, 10h, 9h, 8h, 7h, 6h, 5h, 4h, 3h, 2h, 1h

array2 WORD 1h, 2h, 3h, 4h, 5h, 6h, 7h, 8h, 9h, 10h, 11h, 12h, 13h, 14h, 15h, 16h, 17h, 18h, 19h, 20h

.code

```
mov esi, OFFSET array1
mov ebp, OFFSET array2
mov ecx, 20
L1:
    mov AX, [esi]
    xchg AX, [ebp]
    mov [esi], AX
    add esi, 2
    add ebp, 2
LOOP L1
```

Q3 III

```
.data
    mov esi, OFFSET array
    mov eax, 0
    mov ecx, 100
L1:
    add eax, [esi]
    inc esi
    LOOP L1
```