

Birzeit University

Faculty of Information Technology

Computer Systems Engineering Department

Computer Organization

Homework #1

ENCS 238

Student Name: Zeinab Mohammad

Instructor Name: Muhsen Iwaidah

Section No: 1

**Q1:**

A)Big Endian

1. **struct** {

double i; //0x1112131415161718

};

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
| 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 |

2. **struct** {

int i; //0x11121314

int j; //0x15161718

};

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
| 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 |

3. **struct** {

short i; //0x1112

short j; //0x1314

short k; //0x1516

short l; //0x1718

};

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
| 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 |

B)Little Endian

1. **struct** {

double i; //0x1112131415161718

};

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
| 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |

2. **struct** {

int i; //0x11121314

int j; //0x15161718

};

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 16 | 17 | 18 | 11 | 12 | 13 | 14 |
| 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |

3. **struct** {

short i; //0x1112

short j; //0x1314

short k; //0x1516

short l; //0x1718

};

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 17 | 18 | 15 | 16 | 13 | 14 | 11 | 12 |
| 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |

**Q2:**

1. ABC-+
2. Yes, it is equivalent to (A+B)-c and equivalent to A+(B-c)

c) It matters because of rounding.

**Q3:**

a) ABCDE++++// or AB+C+D+E+

b)AB+CD+\*E+

c)AB\*CD\*+E+  
d)AB-CDE\*-F/G/\*H\*

**Q4:**

|  |  |  |  |
| --- | --- | --- | --- |
| Number | Equivalent in Binary | Arithmetic shift | Logical shift |
| 27 | 000011011 | 011011000 | 011011000 |
| 155 | 010011011 | 011011000 | 011011000 |
| -22 | 111101010 | 110101000 | 110101000 |
| 123 | 010111011 | 000010111 | 000010111 |
| -167 | 101011001 | 111110101 | 000010101 |

1. (27)10=(000011011)2

000011011\* 23=011011000=shift left 3-positions

(155)10=(010011011)2

010011011\* 23=010011011000 two overflow bits so we need two additional bits to make the shift operation result correspond to multiplication operation

(-22)10=(111101010)2

111101010\*22=11110101000=shift

This fact hold only for arithmetic shift lift operation.

1. (123)10=(001111011)2

001111011\*2-3=000011011=shift right 3-positions

(-167)10=(101011001)2

101011001\*2-4=111110101=shift right 4-positions

this fact hold for arithmetic shift right .round towards

-∞ .

**Q5:**

M=(A - B) \* (((C - D \* E)/F)/G) \* H

|  |  |  |  |
| --- | --- | --- | --- |
| 0-Address | 1-Address | 2-Address | 3-Address |
| PUSH A | LOAD D | SUB A,B | SUB R1,A,B |
| PUSH B | MUL E | MUL D,E | MUL D,D,E |
| SUB | STORE R | SUB C,D | SUB R2 ,C,D |
| PUSH C | LOAD C | DIV C,F | DIV R2,R2,F |
| PUSH D | SUB R | DIV C,G | DIV R2,R2,G |
| PUSH E | DIV F | MUL C,H | MUL R2,R2,H |
| MUL | DIV G | MUL A,C | MUL R2,R2,R1 |
| SUB | MUL H | MOV M,A |  |
| PUSH F | STORE R | STORE M |  |
| DIV | LOAD A |  |  |
| PUSH G | SUB B |  |  |
| DIV | MUL R |  |  |
| PUSH H | STORE M |  |  |
| MUL |  |  |  |
| MUL |  |  |  |
| POP M |  |  |  |

Q6:

1. move r0, 0 //set r0=0

move r3, 0 //set r3=0

move r1, 1 //set r1=1

loop: BRE r1, 1024, finish // for(r1=1;r1<1024;r1\*2)

add r0, r0, r1 //r0+=r1

and r2, r0, a0 //r2=r0&a0

add r3, r3, r2 //r3+=r2

ashl r1, r1, 1 //r1=r1\*2

BR loop

finish:

move v0, r3//v0=r3

b) ten times

c) R0=0000 0000 0000 0001

R2=0000 0000 0000 0001

R3=0000 0000 0000 0001

R1=0000 0000 0000 0010

R0=0000 0000 0000 0011

R2=0000 0000 0000 0011

R3=0000 0000 0000 0100

R1=0000 0000 0000 0100

R0=0000 0000 0000 0111

R2=0000 0000 0000 0111

R3=0000 0000 0000 1011

R1=0000 0000 0000 1000

R0=0000 0000 0000 1111

R2=0000 0000 0000 0111

R3=0000 0000 0001 0010

R1=0000 0000 0001 0000

R0=0000 0000 0001 1111

R2=0000 0000 0000 0111

R3=0000 0000 0001 1001

R1=0000 0000 0010 0000

R0=0000 0000 0011 1111

R2=0000 0000 0010 0111

R3=0000 0000 0100 0000

R1=0000 0000 0100 0000

R0=0000 0000 0111 1111

R2=0000 0000 0010 0111

R3=0000 0000 0110 0111

R1=0000 0000 1000 0000

R0=0000 0000 1111 1111

R2=0000 0000 0010 0111

R3=0000 0000 1000 1110

R1=0000 0001 0000 0000

R0=0000 0001 1111 1111

R2=0000 0000 0010 0111

R3=0000 0000 1011 0101

R1=0000 0010 0000 0000

R0=0000 0011 1111 1111

R2=0000 0010 0010 0111

R3=0000 0010 1101 1100

R1=0000 0100 0000 0000

**Q7:** a)the address field.

b)memory location 14.

c)the memory location whose address is in memory location 14.

d)register 14.

e)the memory location whose address is in register 14.

**Q8:** PC=256028+3=256031

Displacement -31 : 256031-31=256000=effective address.