METRO STATE UNIVERSITY

ICS 232 Computer Organization & Architecture
Homework 9 – Irvine Chapter 7 & 8 - 10 points
Due Date: 7/12/2023

Name: Key

Note: Please post your homework to ICS232 D2L on or before the due date.

Irvine Chapter 7 - Integer Arithmetic
Irvine Chapter 8 - Advanced Procedures

1. Which instruction shifts each bit in an operand to the left and copies the highest bit into both the Carry flag and the lowest bit position?

   ROL

2. Which instruction shifts each bit to the right, copies the lowest bit into the Carry flag, and copies the Carry flag into the highest bit position?

   RCR

3. What is the value of AL after each instruction?

```
MOV   AL, 0D4h
SHR   AL, 1          a. 6Ah
MOV   AL, 0D4h
SAR   AL, 1          b. EAh
MOV   AL, 0D4h
SAR   AL, 4          c. FDh
MOV   AL, 0D4h
ROL   AL, 1          d. A9h
```

4. Write the assembly language instructions to multiple EAX by 24 using shift instructions.

```
MOV   EBX, EAX
SHL   EAX, 4         ; multiply by 16
```

ICS 232 Computer Organization & Architecture
Homework 9 – Irvine Chapter 7 & 8 - 10 points
Due Date: 7/12/2023

```
SHL     EBX, 3              ; multiply by 8
ADD     EAX, EBX
```

5. Explain why overflow cannot occur when the MUL and one-operand IMUL instructions execute.

   The product is stored in registers that are twice the size of the multiplier and multiplicand. If you multiply 0FFh by 0FFh, for example, the product (FE01h) easily fits within 16 bits.

6. When EBX is the operand in a DIV instruction, which register holds the quotient?

   EAX

7. When BX is the operand in a DIV instruction, which register holds the quotient?

   AX

8. What will be the contents of EAX and EDX after the following operation?

```
mov edx,0
mov eax,1234567h
mov ecx,100h
mul ecx
```

   EDX = 00000001, EAX = 23456700

9. What will be the contents of EAX and EDX after the following operation?

```
mov eax,63h
cdq
mov ebx,10h
div ebx
```

   EAX = 00000006, EDX = 00000003

METRO STATE UNIVERSITY

ICS 232 Computer Organization & Architecture
Homework 9 – Irvine Chapter 7 & 8 - 10 points
Due Date: 7/12/2023

10. Implement the following C expression in assembly language, using 32-bit integer signed operands:

```
val1 = (val2 / val3) * (val1 + val2);
```

```
     MOV  EAX, val2
     CDQ
     IDIV val3           ; val2 / val3
     MOV  EBX, EAX       ; save result
     MOV  EAX, val1
     ADD  EAX, val2      ; val1 + val2
     IMUL EAX, EBX       ; multiply
     MOV  val1, EAX      ; save result

val1 SDWORD 0
val2 SDWORD 0
val3 SDWORD 0
```

11. Implement the following C code fragment in assembly language, using 32-bit integer signed operands:

```
int test(int x, int y)
{
   int r;

   if (x > y)
      r = x * y;
   else if (x == y)
      r = x / y;
   else
      r = x + y;
   return (r);
}
```

ICS 232 Computer Organization & Architecture
Homework 9 – Irvine Chapter 7 & 8 - 10 points
Due Date: 7/12/2023

```
test proc
    push  ebp
    mov   ebp, esp
    sub   esp, 4
    mov   eax, DWORD PTR 8[ebp]      ; x
    cmp   eax, DWORD PTR 12[ebp]     ; y
    jle   elseif
    imul  eax, DWORD PTR 12[ebp]
    mov   DWORD PTR -4[ebp], eax     ; r
    jmp   endif
 elseif:
    jne   else
    cdq
    idiv  DWORD PTR 12[ebp]
    mov   DWORD PTR -4[ebp], eax
    jmp   endif
 else:
    add   eax, DWORD PTR 12[ebp]
    mov   DWORD PTR -4[ebp], eax
 endif:
    mov   eax, DWORD PTR -4[ebp]
    mov   esp, ebp
    pop   ebp
    ret
test endp
```

64-bit (-g):

Arguments are in RDI, RSI, RDX, R10, R8, R9

```
  12 0004 55             push rbp
  15 0005 4889E5         mov  rbp, rsp
  17 0008 897DEC         mov  DWORD PTR -20[rbp], edi
  18 000b 8975E8         mov  DWORD PTR -24[rbp], esi
  15:Homework 9.c  ****    int r;
  16:Homework 9.c  ****
  17:Homework 9.c  ****    if (x > y)
```

METRO STATE UNIVERSITY

ICS 232 Computer Organization & Architecture
Homework 9 – Irvine Chapter 7 & 8 - 10 points
Due Date: 7/12/2023

```
20 000e 8B45EC          mov  eax, DWORD PTR -20[rbp]
21 0011 3B45E8          cmp  eax, DWORD PTR -24[rbp]
22 0014 7E0C            jle  .L2
18:Homework 9.c  ****       r = x * y;
24 0016 8B45EC          mov  eax, DWORD PTR -20[rbp]
25 0019 0FAF45E8        imul eax, DWORD PTR -24[rbp]
26 001d 8945FC          mov  DWORD PTR -4[rbp], eax
27 0020 EB1F            jmp  .L3
28                .L2:
19:Homework 9.c  ****     else if (x == y)
30 0022 8B45EC          mov  eax, DWORD PTR -20[rbp]
31 0025 3B45E8          cmp  eax, DWORD PTR -24[rbp]
32 0028 750C            jne  .L4
20:Homework 9.c  ****       r = x / y;
34 002a 8B45EC          mov  eax, DWORD PTR -20[rbp]
35 002d 99              cdq
36 002e F77DE8          idiv DWORD PTR -24[rbp]
37 0031 8945FC          mov  DWORD PTR -4[rbp], eax
38 0034 EB0B            jmp  .L3
39                .L4:
21:Homework 9.c  ****     else
22:Homework 9.c  ****       r = x + y;
41 0036 8B55EC          mov  edx, DWORD PTR -20[rbp]
42 0039 8B45E8          mov  eax, DWORD PTR -24[rbp]
43 003c 01D0            add  eax, edx
44 003e 8945FC          mov  DWORD PTR -4[rbp], eax
45                .L3:
23:Homework 9.c  ****     return (r);
47 0041 8B45FC          mov  eax, DWORD PTR -4[rbp]
24:Homework 9.c  **** }
49 0044 5D              pop  rbp
51 0045 C3              ret
```

64-bit (-O):

```
 8 0000 89F0            mov  eax, esi
 9 0002 39F7            cmp  edi, esi
```

METRO STATE UNIVERSITY

ICS 232 Computer Organization & Architecture
Homework 9 – Irvine Chapter 7 & 8 - 10 points
Due Date: 7/12/2023

```
10 0004 7E04            jle   .L2
11 0006 0FAFC7          imul eax, edi
12 0009 C3              ret
13                .L2:
14 000a 7403            je    .L4
15 000c 01F8            add   eax, edi
16 000e C3              ret
17                .L4:
18 000f B8010000        mov   eax, 1
18      00
19 0014 C3              ret
```

12. What is the equivalent C code?

```
     whatDoIDo      proc
0000 55             push   ebp
0001 89E5           mov    ebp, esp
000d 8B4508         mov    eax, DWORD PTR 8[ebp]
0010 99             cdq
0011 F77D0C         idiv   DWORD PTR 12[ebp]
0014 89D0           mov    eax, edx
0016 5D             pop    ebp
0017 C3             ret
     whatDoIDo      endp
```

```
int whatDoIDo(int x, int y)
{
     return (x % y);
}
```

Prepare for next class by reading Chapter 6 – Memory

Start working on Project 2

Continue working on Your Group Project