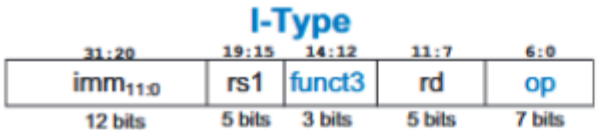
Course: Computer Organization – ENCM 369
Lab #: Lab 6
Instructor Name: Norm Bartley
Student Name: Stephen Ravelo
Lab Section: B03
Date submitted: March 7, 2025

Exercise A

1)



textbook p.334

imm = -160 $imm_{11:0}$ = 1111 0110 0000

rs1 = 00010 rd = 00010

funct3 = 000 op = 19 = 0010011

Ans. 111101100000_00010_000_00010_0010011

2)

```
  1
  0111_[12 ones]_1110_1111_1010_0000   <- sp
+ 1111_[12 ones]_1111_1111_0110_0000   <- imm
-------------------------------------
  0111_[12 ones]_1110_1111_0000_0000   <- sp
```

Exercise B

1)
```
  1011
  1011_0100
+ 1011_0011
-----------
  0100 0111
```
Signed overflow: MSB carry-out/in don't match.

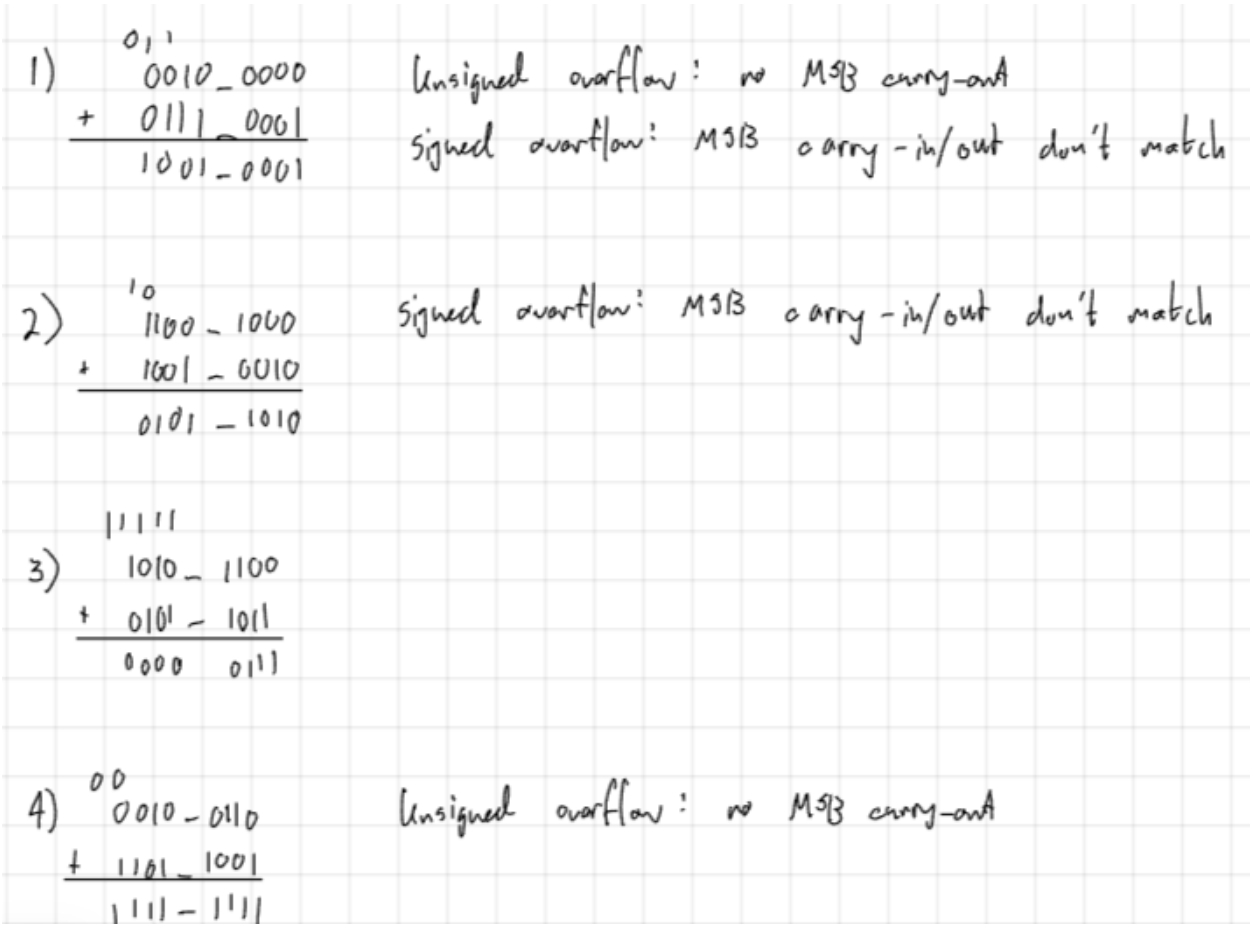
Unsigned overflow: MSB carry-out = 1

2)
```
  11
  1101_0000
+ 1110_0000
-----------
  1011_0000
```
Unsigned overflow: MSB carry-out = 1

3)
```
  01111
  0111_1000
+ 0000_1011
-----------
  1000_0011
```
Signed overflow: MSB carry-out/in don't match.

4)
```
  0 0111 1 1
  0011_0101
+ 0010_1101
-----------
  0110_0010
```

Exercise E


1)
0 1
0010_0000
+ 0111_0001
1001_0001
Unsigned overflow: no MSB carry-out
signed overflow: MSB carry-in/out don't match
2)
1 0
1100_1000
+ 1001_0010
0101_1010
signed overflow: MSB carry-in/out don't match
3)
1 1 1 1
1010_1100
+ 0101_1011
0000 0111
4)
0 0
0010_0110
+ 1101_1001
1111_1111
Unsigned overflow: no MSB carry-out

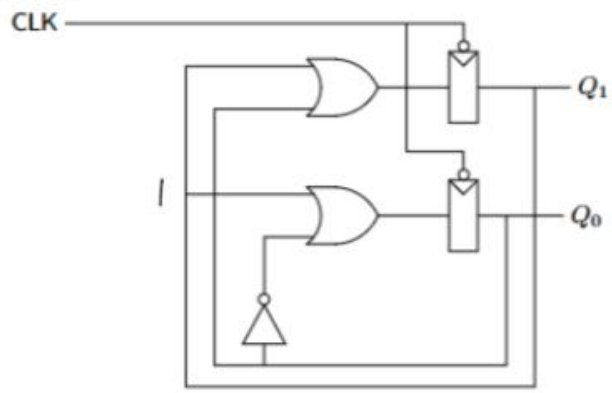
Exercise G

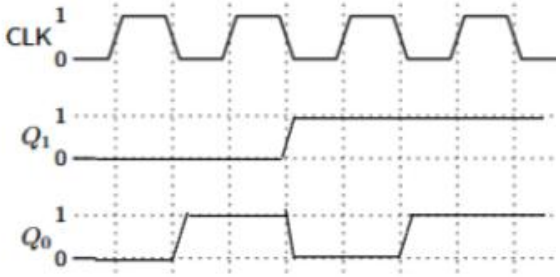
```
    .text
    .globl    int2str
int2str:
# Replace these two comment lines and the following instruction with
# code to match the definition of int2str in int2str.c.
    bne       a1, zero, L1            # if (src != 0) goto L1
    li        t6, '0'                 # t6 = '0'
    sb        t6, (a0)                # dest[0] = t6
    sb        zero, 1(a0)             # dest[1] = '\0'
    jr        ra
L1:
    li        t6, -2147483648         # t6 = -2147483648
    bne       a1, t6, L2              # if (src != t6) goto L2
    lui       t0, 0x80000             # abs_src = 0x80000000
    j         L4
L2:
    bge       a1, zero, L3            # if (src >= 0) goto L3
    sub       t0, zero, a1            # abs_src = -src
    j         L4
L3:
    mv        t0, a1                  # abs_src = src
L4:
    mv        t4, a0                  # p = dest
    li        t1, 10                  # ten = 10
L5:
    beq       t0, zero, L6            # if (abs_src == 0) goto L6
    remu      t2, t0, t1              # rem = abs_src % ten
    la        t6, digits              # t6 = &digits
    add       a3, t6, t2              # a3 = &digits[rem]
    lbu       a4, (a3)                # a4 = digits[rem]
    sb        a4, (t4)                # *p = a4
    addi      t4, t4, 1               # p++
    divu      t0, t0, t1              # abs_src = abs_src / ten
    j         L5
L6:
    bge       a1, zero, L7            # if (src >= 0) goto L7
    li        t6, '-'                 # t6 = '-'
    sb        t6, (t4)                # *p = t6
    addi      t4, t4, 1               # p++
L7:
    sb        zero, (t4)              # *p = '\0'

    addi      t5, t4, -1              # q = p - 1
```

```
    mv      t4, a0                  # p = dest
L8:
    bge     t4, t5, L9              # if (p >= q) goto L9
    lbu     t3, (t4)                # temp = *p
    lbu     t6, (t5)                # t6 = *q
    sb      t6, (t4)                # *p = t6
    sb      t3, (t5)                # *q = temp
    addi    t4, t4, 1               # p++
    addi    t5, t5, -1              # q--
    j       L8
L9:
    jr      ra
```

Exercise H

Part I





$Q_1' = Q_1 + Q_0$

$Q_2' = Q_1 + \overline{Q_0}$

Part II

$Q_0' = A$

$Q_1' = Q_0$

$Q_2' = Q_1$