ECE3221 Lab 0
Name: Stephen Cole
ID: 3553803

Pre-lab

Some examples

- dd6E -> 1101 1101 0110 1110
- 98b9 -> 1001 1000 1011 1001
- d9C6 -> 1101 1001 1100 0110

Figure 1.01

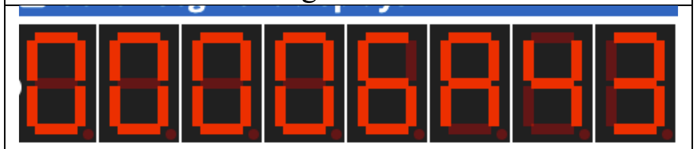
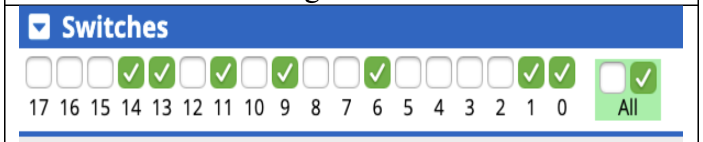


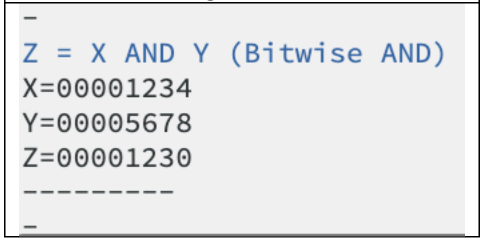
Figure 1.02



Lab

Example 1

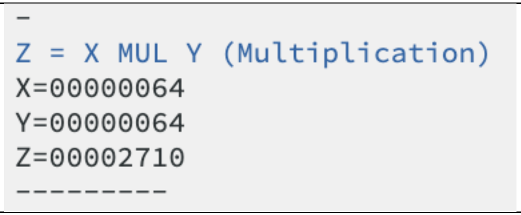
Figure 1.1



```
-
Z = X AND Y (Bitwise AND)
X=00001234
Y=00005678
Z=00001230
---------
-
```

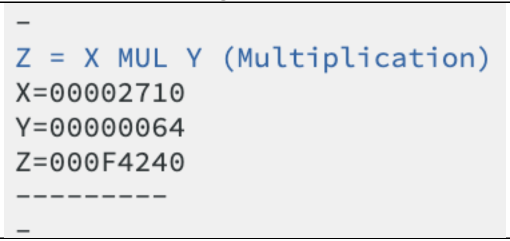
Example 2

Figure 2.1



```
-
Z = X MUL Y (Multiplication)
X=00000064
Y=00000064
Z=00002710
---------
```

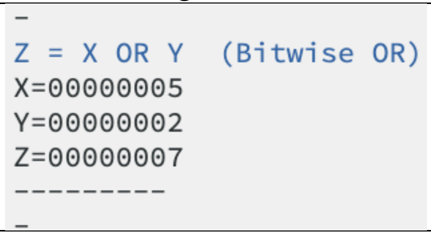
Figure 2.2



```
-
Z = X MUL Y (Multiplication)
X=00002710
Y=00000064
Z=000F4240
---------
-
```

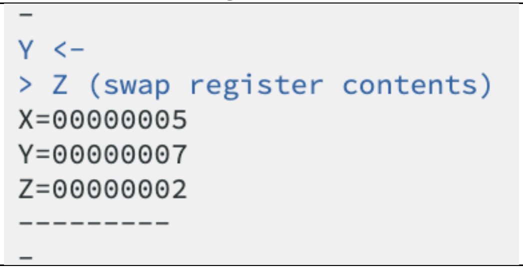
Example 3

Figure 3.1



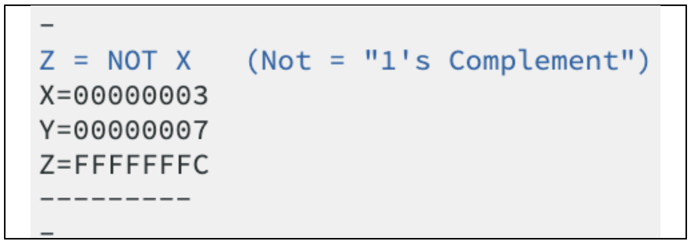
```
-
Z = X OR Y  (Bitwise OR)
X=00000005
Y=00000002
Z=00000007
---------
-
```

Figure 3.2



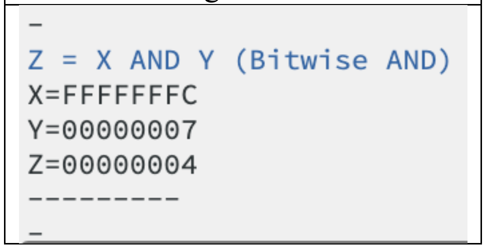
```
-
Y <-
> Z (swap register contents)
X=00000005
Y=00000007
Z=00000002
---------
-
```

Figure 3.3



```
-
Z = NOT X    (Not = "1's Complement")
X=00000003
Y=00000007
Z=FFFFFFFC
---------
-
```

Figure 3.4



```
-
Z = X AND Y (Bitwise AND)
X=FFFFFFFC
Y=00000007
Z=00000004
---------
-
```

Procedure

1a.

AND

1	0	1	1	0	1	1	1	0	0	1	1	1	=	9
0	0	0	0	0	1	1	1	1	1	1	1	1	=	0
0	0	0	0	0	1	1	1	0	0	1	1	1	=	0

Table 1a

b.

OR

1	0	1	1	0	1	1	1	0	0	1	1	1	=	9
1	1	1	0	0	0	0	0	0	1	1	1	1	=	3
1	1	1	1	0	1	1	1	0	1	1	1	1	=	B

Table 1b

c.

XOR

1	0	1	1	0	1	1	1	0	0	1	1	1	=	9
0	0	0	0	0	1	1	1	1	1	1	1	1	=	0
1	0	1	1	0	0	0	0	1	1	0	0	0	=	9

Table 1c

d.

ADD

1	0	1	1	0	1	1	1	0	0	1	1	1	=	1
0	0	0	0	0	1	1	1	1	1	1	1	1	=	0
1	0	1	1	1	1	1	1	0	0	1	1	0	=	9

Table 1d

2.

Results:

0x00000FF0 AND 0x00001234 = 0x00000230
BIN: 0000 0000 0000 0000 0010 0011 0000

Explanation: In this case the AND function is used to force all but two hexadecimal values of Y low. As you can see in the example below hexadecimal values 3 and 0 have been forced low by the operation.

3 2 1 0
Y = 0001 0010 0011 0100
Z = 0000 0010 0011 0000

0x00000FF0 OR 0x00001234 = 0x00001FF4
BIN: 0000 0000 0000 0001 1111 1111 0100

Explanation: In this case the OR function is used to force certain values of Y to a high state. As you can see from the example below values 2 and 1 of Y have been forced high.

3 2 1 0
Y = 0001 0010 0011 0100
Z = 0001 1111 1111 0100

0x00000FF0 XOR 0x00001234 = 0x00001DC4
BIN: 0000 0000 0000 0001 1101 1100 0100

Explanation: In this case the XOR function is used to invert selected values of Y. As you can see from the example below hexadecimal values 2 and 1 have been inverted

3 2 1 0
Y = 0001 0010 0011 0100
Z = 0001 1101 1100 0100

3.

Results:

0x0000FFFF SLL 0x00000001 = 0x0001FFFE
BIN: 0000 0000 0000 0001 1111 1111 1110

0x0000FFFF SLL 0x00000002 = 0x0003FFFC
BIN: 0000 0000 0000 0011 1111 1111 1100

0x0000FFFF SLL 0x00000003 = 0x0007FFF8

BIN: 0000 0000 0000 0111 1111 1111 100

0x0000FFFF SLL 0x00000004 = 0x000FFFF0
BIN: 0000 0000 0000 1111 1111 1111 0000

What is the overall effect when Y=4?

The value of X 0x0000FFFF is multiplied by 16 as each shift left accounts for a multiplication of 2^x where x = Y.

4.

Results:

0x0000FFFF SLL 0x00000014 = 0xFFF00000
BIN: 1111 1111 1111 0000 0000 00000 0000 00000

0x0000FFFF ROL 0x00000014 = 0xFFF0000F
BIN: 1111 1111 1111 0000 0000 00000 0000 1111

Explain the difference between shift left and rotate left:

The difference between shift left and roll left is that if the value is shifted to a value that is greater than the maximum number of bits allotted, the values contained in the bits those locations will be forgotten. Whereas roll left will move those values to the start of the allocated bits.

5.

Results:

0x00000065 AND 0x00000001 = 0x00000001

a. What does the result of this operation tell you about X?

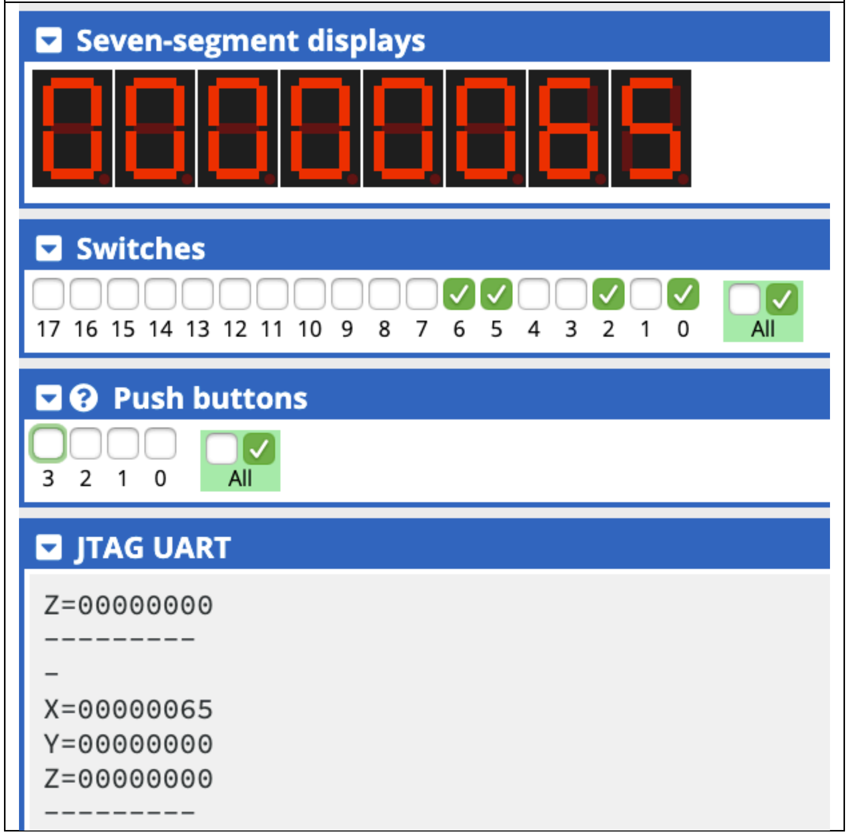
The result of this operation tells us that X is used to force all bits but the LSB to a low state.

b. Write the operation that would complement only the least significant bit in X.

0x00000001 XOR 0x00000001 = 0x00000000

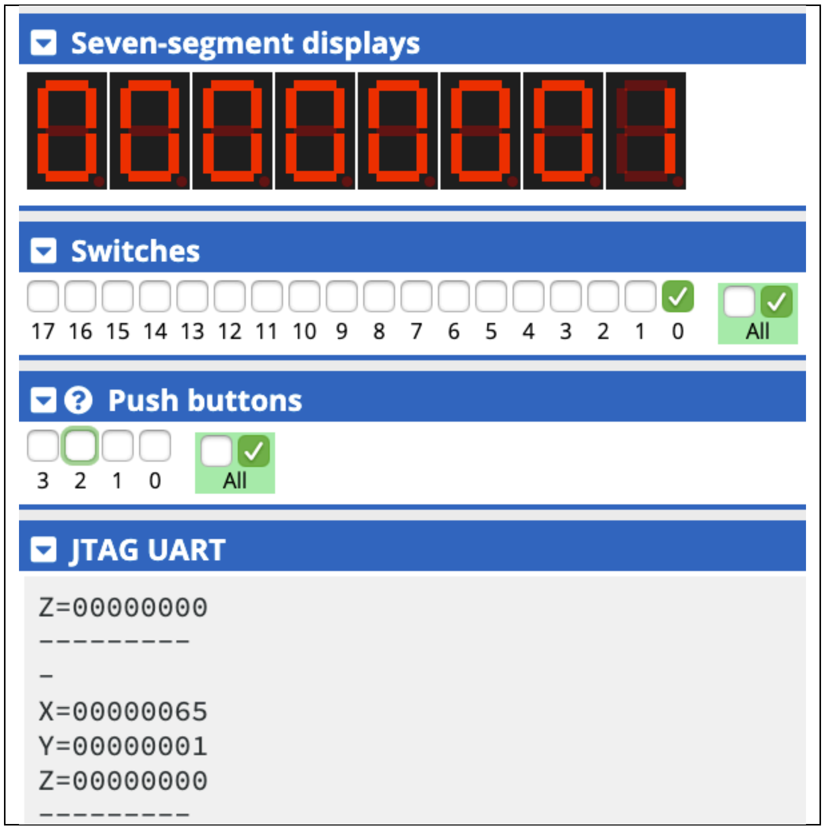
c. Write the operation that would set only the least significant bit to 0 in X. Show the steps necessary to do this using the calculator.

0x00000065 XOR 0x00000001 = 0x00000064

Figure 5.1
Set X=0x00000065


```
Z=00000000
---------
-
X=00000065
Y=00000000
Z=00000000
---------
```

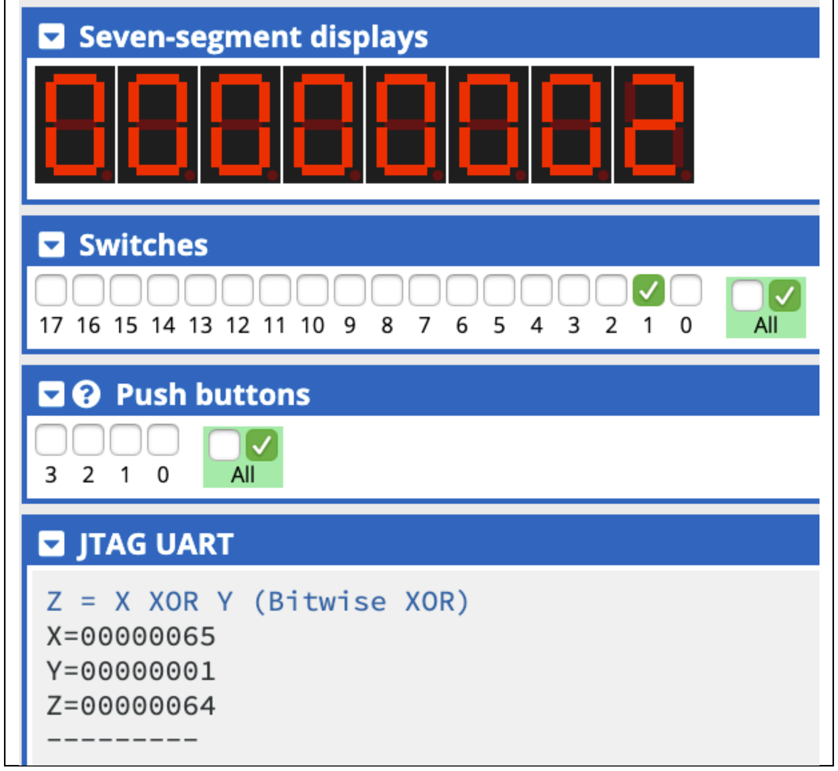
Figure 5.2
Set Y = 0x00000001



```
Z=00000000
---------
-
X=00000065
Y=00000001
Z=00000000
---------
```

Figure 5.3

Use XOR (0010) to set the least significant bit of X to 0.



```
Z = X XOR Y (Bitwise XOR)
X=00000065
Y=00000001
Z=00000064
---------
```