DEPARTMENT OF COMPUTER SCIENCE

COLORADO SCHOOL OF MINES
EARTH • ENERGY • ENVIRONMENT

# CSCI 341: Computer Organization
## WS 5: Bitwise Instructions

1 How would you bit mask the bottom half of a word? That is, how would you set the top half of a word to 0 using logical operators only?

Solution: without pseudo/enhanced instructions, only 12 bit immediates

```
addi t0, zero, 0xFF # t0 = 0xFF
slli t0, t0, 8      # t0 = 0xFF00
ori t0, t0, 0xFF    # t0 = 0xFFFF
and s0, s0, t0
```

2 The NOR operation does not exist in RISC-V. How would you do this in RISC-V? Here is its truth table:

| A | B | A NOR B |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

Assuming values are in t0 and t1, result in t0:

```
or t0, t0, t1
xori t0, t0, -1
```

3 Assume that x (using s0) and y (using s1) are both ints. Recall that bits are numbered from right to left in a word starting with bit 0.

Write two lines of RISC-V code to set x to the value contained in bits 15 to 10 of y. That is, x has y's bits 15 to 10 as its bits 5 to 0, and 0's in all other bits.

Shift and mask solution:

```
srli s0, s1, 10       # x = y>>10
andi s0,s0,0x3F # x=x&0x3F (=(y>>10)&0x3F)
```

Two shift solution:

```
slli s0,s1,16     # x = y <<16
srli s0, s0, 26 # x = x >> 26 (=(y<<16)>>26)
```

| | |
|---|---|
| 4 | Set s1 to hold the value 0xAACCBBDD<br><br>End up with the value 0x00AACC00 in s0, using only logical operations to erase bits and move bits around, and print it out as a hexadecimal.<br><br>Make it so that it creates a value from s1 that is the upper two bytes of s1 as the middle two bytes of s0 no matter what the starting value is.<br><br>`lui s1, 0xAACCC`<br>`addi s1, s1, 0xFFFFFBDD`<br>`srli s0, s1, 16`<br>`slli s0, s0, 8`<br><br>`addi a7, zero, 34`<br>`add a0, zero, s0`<br>`ecall`<br>`addi a7, zero, 11`<br>`addi a0, zero, 10`<br>`ecall`<br><br>`lui s1, 0xAACCC`<br>`addi s1, s1, 0xFFFFFBDD`<br>`lui t0, 0xFFFF0`<br>`and s0, s1, t0`<br>`srli s0, s0, 8`<br><br>`addi a7, zero, 34`<br>`add a0, zero, s0`<br>`ecall` |