

CSCI-21 Programming Assignment #1, due 2/10/16

In the Settings menu of QTSpim set Bare Machine ON, Allow Pseudo Instructions OFF, Load Exception Handler OFF, Delayed Branches OFF, Delayed Loads OFF, Mapped IO OFF, Load Exception File OFF. Email me the source file(s) attached to an email with "CS21 your name Assign 1" in the subject line.

For these programming exercises, you may use ONLY these instructions:

```
and nor or ori sll srl xor
```

Run the programs by verifying the value of the PC is 0x00400000 (it defaults there in QTSpim) and then single stepping (pushing F10) or by multiple stepping (push F11 and enter a number of steps), observing the results in the SPIM register display window.

Exercise 1:

Start a program with the instruction that puts a single one-bit into the low order bit of register eight (\$8 or \$t0):

```
ori    $t0, $0, 0x01
```

Now, by using only shift logical instructions and register to register logic instructions (use NO more immediate instructions), put the pattern 0xAAAAAAAA into register \$9 (or \$t1). You may not use another `andi`, `ori` or `xori` instruction to set another bit from scratch, you must work from the single bit you set in the first instruction or created directly from that bit. You will need to use more registers than just \$9. See how few instructions you need to do this. Doing this in approximately ten to twelve instructions is reasonable. It can be done in fewer. Again, you MAY NOT create any other data bits from scratch for this, only the single bit you started with.

Exercise 2:

Again, start a program with the instruction that puts the single one-bit into register eight (\$t0):

```
ori    $t0, $0, 0x01
```

Now, by using only shift logical instructions and register to register logic instructions (use NO more immediate instructions), put the pattern 0x96969696 into register \$9. Do not use another `andi`, `ori` or `xori` instruction. You will need to use other registers than just \$9. See how few instructions you need to do this. Again, this can be done in approximately ten to twelve instructions. Again, you may only work from the one initial bit.

Exercise 3:

Again, start a program with the instruction that puts the single one-bit into register eight (\$t0):

```
ori    $t0, $0, 0x01
```

Now, by using only shift logical instructions and register to register logic instructions (use NO more immediate instructions), put the pattern 0xFFFFFFFF into register \$9. Don't use another `andi`, `ori` or `xori` instruction. You will need to use other registers than just \$9. See how few instructions you need to do this. This can be done in very few instructions. Again, you may only work from the one initial bit.