**Homework #2**

Trevor Lund

1. Operations and Operands
   1. Having that many registers decreases 1) physical space on the chip, 2) reliability of the memory during computations, and 3) speed of accessing so many registers.
   2. Use a shift right logical command and divide by two to the seventh.
   3. ABI refers to the calling conventions between functions, meaning what registers are used and what sizes the various C data types are, whereas ISA refers to the instructions and registers a CPU has available. $sp is the stack pointer and $zero just represents 0. If the MIPS code I wrote used $at explicitly, it might get overwritten by the MIPS assembler.
2. Compiling and Assembling by Hand
   1. Loop: add $t0, $s6, $s3

Addi $s3, $s3, 4

Beq $t0, $s5, Loop

Took 52 instructions before optimization, and only 27 after.

* 1. $v0 = 0

$v1 = 1

$a0 = 2

$a1 = 2

Formats:

Addi = I

Lw = I

Sw = I

Beq = I

After the loop is completed.

1. New Operations in MIPS
   1. $s1 is divided by 2, $s0 \* 2^31 is stored as $t0, and $s0 is divided by 2. $s1 either stays the same or is replaced by $t0.
   2. Abs $s0, $s1, Where $s1 is the source and $s0 is the destination

Sub $s0, $zero, $s1

Add $t1, $s0, $s1

Bne $t1, $zero, ABS

Add $s0, $zero, $s1

ABS:

Format is I.

* 1. Self modifying code is a bad idea because if the value at all changes due to a bug or some other fluke, your entire program or even your computer could be severely damaged.