CS151B/EE116C – Solutions to Homework #2

1. (2.24)

Answer: Not possible for either.

The jump instruction uses pseudodirect addressing in an attempt to provide the ability to jump to any location. However, because MIPS instructions are 32-bits and must include a 6-bit opcode, the instruction only has a 26-bit address field. This field is multiplied by 4 (or alternatively, shifted left by 2) to get a 28-bit byte address. The lowest 28 bits of the incremented PC are then set to these 28 bits. This means the "j" instruction is only able to change the lower 28-bits of the PC. If the current PC is 0x20000000, we can only jump to addresses in the range of 0x20000000 to 0x2FFFFFFC. To jump from 0x20000000 to 0x40000000, you would need to change bits 29 and 30.

The beq has a 16 bit immediate field and sets PC = PC+4+(imm*4) when the two registers are equal. Because the immediate field is a signed integer, it ranges between -2^{15} and $2^{15} – 1$. As a result when used in a beq, this is multiplied by 4 and allows for branching 2^{17} addresses in the negative direction and $2^{17} – 4$ addresses in the positive direction. As a result, if PC = 0x2000 0000, the highest address that can be branched to is 0x2000 0000 + 0x4 + 0x0001 FFFC = 0x2002 0000.

2. (2.26.1)

Answer: 20

For each loop, the code will subtract 1 from $t1 and add 2 to $s2. After 10 loops (where 2 is added to $s2 each time), $t1 will be zero. When the slt is executed, $0 will not be less than $t1, which results in $t2 being set to 0. Afterwards, the beq will branch to DONE.

3. (2.26.3)

Answer: 5*N + 2

All five instructions in the loop body will be executed N times, after which $t1 will be zero. The slt and beq are then each executed once more to exit the loop.

4. (2.46.1)

Answer: Based only on the performance of this program before and after the change, adding the new instructions would not be a good design choice.

$$ET_{before} = IC_{before} * CPI_{before} * CT_{before}$$
$$= Cycles\_Executed_{before} * CT_{before}$$

$$Cycles\_Executed_{before} = 500*10^6 \text{ (arith instructions)} * 1\ (CPI_{arith}) + 300*10^6 \text{ (loads/stores)} *10\ (CPI_{l/s}) + 100*10^6 \text{ (branches)} * 3\ (CPI_{branch}) = 3800*10^6$$

ET_{before} = $\mathbf{3800*10^6 * CT_{before}}$

$$ET_{after} = IC_{after} * CPI_{after} * CT_{after}$$
$$= Cycles\_Executed_{after} * CT_{after}$$

$$Cycles\_Executed_{after} = 375*10^6 \text{ (arith instructions)} * 1\ (CPI_{arith}) + 300*10^6 \text{ (loads/stores)} *10\ (CPI_{l/s}) + 100*10^6 \text{ (branches)} * 3\ (CPI_{branch}) = 3675*10^6$$

$$CT_{after} = 1.1*CT_{before}$$

$$ET_{after} = 3675*10^6 * 1.1 * CT_{before}$$
$$\mathbf{= 4042.5 * 10^6 * CT_{before}}$$