Question 1 (15 points): For each of the following statements you have three options: (i) Leave it blank; (ii) Mark it with T to indicate that the statement is true; (iii) Mark it with F to indicate that the statement is false. You lose 3 points for each incorrect answer. You win 3 points for each correct answer. You dont win or lose any points if you leave the statement blank. Regardless of how many statements you mark wrong, your score in this question cannot be below zero.

a) (F) The clock cycle time when implementing the 5-stage MIPS pipeline is 5 times faster than the clock cycle time for the single-cycle MIPS datapath.

Some stages are longer than others.

b) (F) Every bus in a datapath (single-cycle or pipelined) must be either 1 bit (for a control signal) or 32 bits (for data).

We can have buses of other sizes (ie. 16 bits)

c) (T) Consider the following C variable declarations:

```
int *x;
char *y;
```

If the address of **x** is in **$a0** and the address of **y** is in **$a1**, then the following MIPS instructions load **x** and **y** into registers **$t0** and **$t1**.

```
lw $t0, 0($a0)
lw $t1, 0($a1)
```

x and y are both pointers (addresses)

d) (T) Assume that PC = 0x8000 0000 and there is a **beq** instruction at that address. Given our current MIPS implementation, the highest possible address of the next instruction executed is 0x8002 0000. If we could add one more bit to the immediate field of the instruction, then the highest possible address would increase to 0x8004 0000.

Adding a bit doubles the range.

e) ( ) When an exception occurs, control is transferred to another MIPS program (the Exception Handler), which has to save any registers it uses to someplace other than the stack.

My intention for this question was "any $s or $t registers", however this was not worded that way. The exception handler must save any registers other than $k0 and $k1, but cannot use the stack.

Due to this, I accepted T/F/blank.