Review Lecture II

In this lecture two problems from a previous year's exam will be solved.

State Machines

This question asks you to design a digital logic controller for a ski lift. This lift must operate at a slower-than-normal speed for the first 1.5 seconds after starting to reduce the start-up shock on the skiers already on the lift.

The lift operator has two pushbuttons available: `START` (or "X") and `STOP` (or "Y"). Each pushbutton generates a logic 1 signal when it is being pushed and 0 if it is not being pushed. The lift should start when the `START` button is pushed. If the lift is already running at either slow or normal speed the `START` button has no effect. The lift should be stopped immediately when the `STOP` button is pushed regardless of the current speed.

The lift motor has two speeds and is controlled by two logic signals: `RUN` and `SLOW`. When `SLOW` is logic 1 the motor will run at slow speed and when `SLOW` is 0 the motor will run at normal speed. However, the motor only runs when the `RUN` signal is logic 1. When the `RUN` signal is logic 0 the motor stops.

A timer circuit is available for your controller. The timer has one logical input (`RESET`) and one logical output, (`TIME` or "Z"). When the `RESET` signal goes from 0 to 1 the `TIME` signal becomes 0. The `TIME` signal goes back to 1 after a delay of 1.5 seconds following the last 0-to-1 transition of the `RESET` signal.

Design a state machine controller for the ski lift controller. List the inputs and outputs. List a sufficient number of states and a name for each state. Give a table giving the output conditions for each state. Draw a state transition diagram showing the states and the logical conditions that cause transitions between them. Write out a tabular description of the state machine with the following columns: starting state, input, next state.

C Programming

This question asks you to write two C functions to interface to a communication device.

A data communication device is interfaced to a computer using an 8-bit status register at address 0x2e0 and an 8-bit data register at address 0x2e1. The status register can be read to determine the state of the interface. The least significant bit (bit 0) is 1 only if a character has been received. The most significant bit (bit 7) is 1 only if the transmitter buffer is empty and a new character can be transmitted. Bit 3 of the status register is 1 only when the device is properly connected to a compatible communication device. The values of the other bits in the status register are used for other purposes and you may not assume anything about their values.

Received characters may be retrieved from the interface by reading from the data register. Characters may be sent by writing their values to the data register.

Assume that the function `char peek ( int a )` returns the value of a byte at address `a` and the function `void poke ( int a, char c )` writes a byte of value of `c` to memory address `a`.

Write two C functions, declared as

```
char cget(void)
```

and

```
void cput(char c)
```

The first function should return the value of the next character received over the interface. The second function should send its argument over the interface. Both functions should wait for the communication device to be connected to a compatible device and for either a character to be received or for the transmitter buffer to be empty as appropriate. Your functions should be written using (`#define`'d) symbolic constants rather than numeric constants. Both the syntax and the logic of your program must be correct.