(2 pts) Name:

AME 3623: Embedded Real-Time Systems: Final Exam

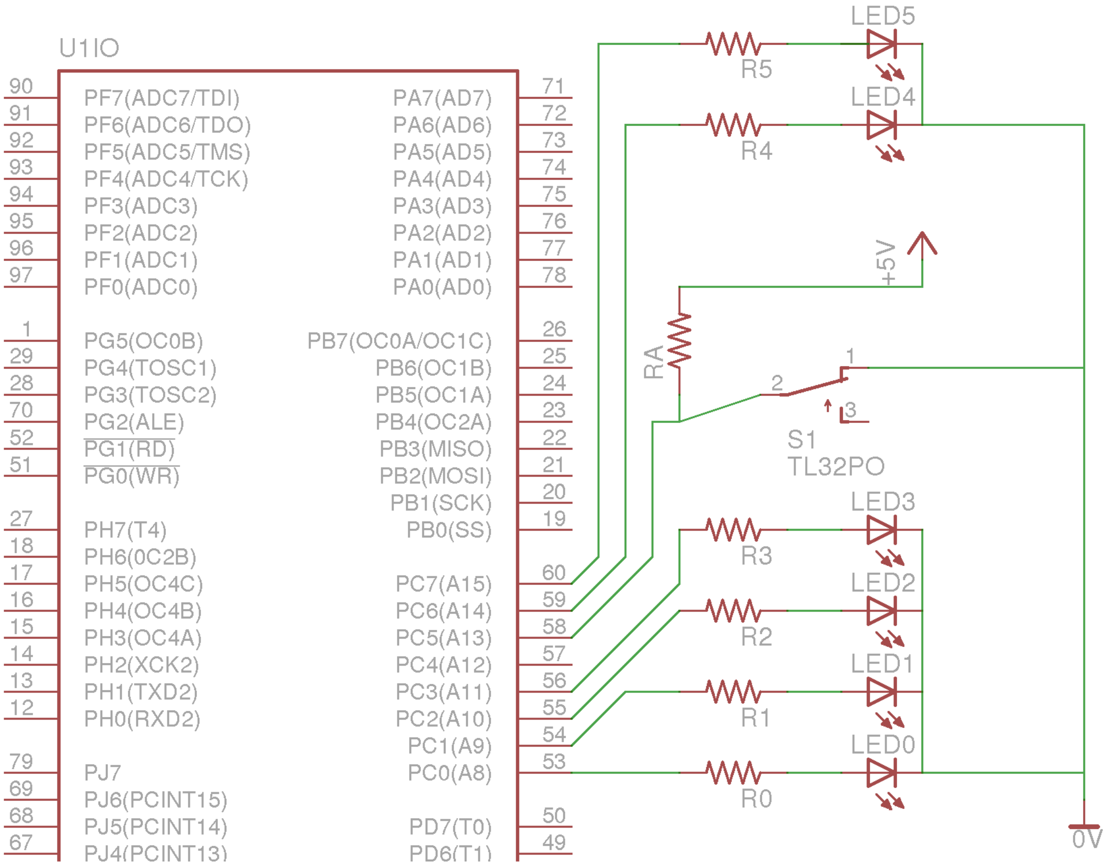
Solution Set

May 10, 2012

Problem	Topic	Max	Grade
0	Name	2	
1	Interrupt Service Routines and Digital I/O	35	
2	Number Representation and Arithmetic	20	
3	Finite State Machines	45	
4	Analog Circuits	35	
5	Analog Processing	40	
6	Microprocessors	25	
Total		202	

1. **Interrupt Service Routines and Digital I/O** (35 pts)

Carefully consider the following circuit:



And consider the following program:

```
ISR(TIMER5_OVF_vect) {
  static uint8_t counter = 0;

  if(counter == 9) {
      PORTC ^= 0x40;
  }

  PORTC = PORTC & 0xF0 | counter & 0xF;
  counter += 3;
}

int main(void) {
  DDRC = 0xCF;
  PORTC = 0;
  timer5_config(TIMER5_PRE_64);
  timer5_enable();
  sei();

  while(1) {
  }
}
```

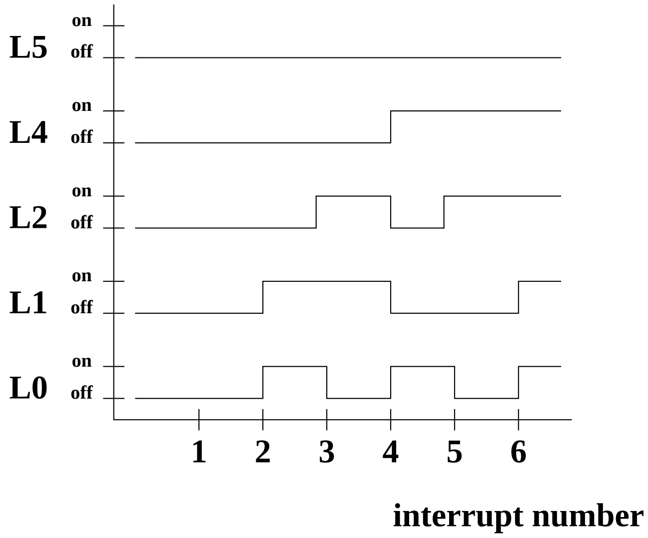
(a) (5 pts) Assuming a system clock of $16MHz$, at what frequency is the timer 5 counter incrementing? (give the ratio with units)

$$\frac{16{,}000{,}000\ ticks/sec}{64\ ticks/tock}$$

(b) (5 pts) At what frequency is the timer 5 overflow interrupt being generated? (give the ratio with units)

$$\frac{16{,}000{,}000\ ticks/sec}{64\ ticks/tock \times 256^2\ tocks/int}$$

(c) (25 pts) Show the state of LEDs 0, 1, 2, 4 and 5 as a function of interrupt number for interrupts 1 through 6.



2. **Number Representation and Arithmetic** (20 pts)

(a) (5 pts) What is the binary equivalent of hexadecimal 78? Show your work.

111 1000

(b) (5 pts) Take the two's complement (the negative) of the above number (assume an 8-bit, signed representation). Show your work.

1000 1000

(c) (5 pts) Compute $78 - 0x78$ using binary arithmetic. Show your work and give your answer in 8-bit binary two's complement.

decimal 78 is binary 01001110

```
  10001000
+ 01001110
----------
  11010110
```

(d) (5 pts) What is the decimal equivalent of the above result?

This is a negative number. The two's complement inverse is 00101010, which is decimal 42
So: the original result is -42.
Alternatively: -128+64+16+4+2 = -42

3. **Finite State Machines and Control** (45 pts)

Consider the problem of controlling the door of an elevator. The properties and rules of behavior are as follows:

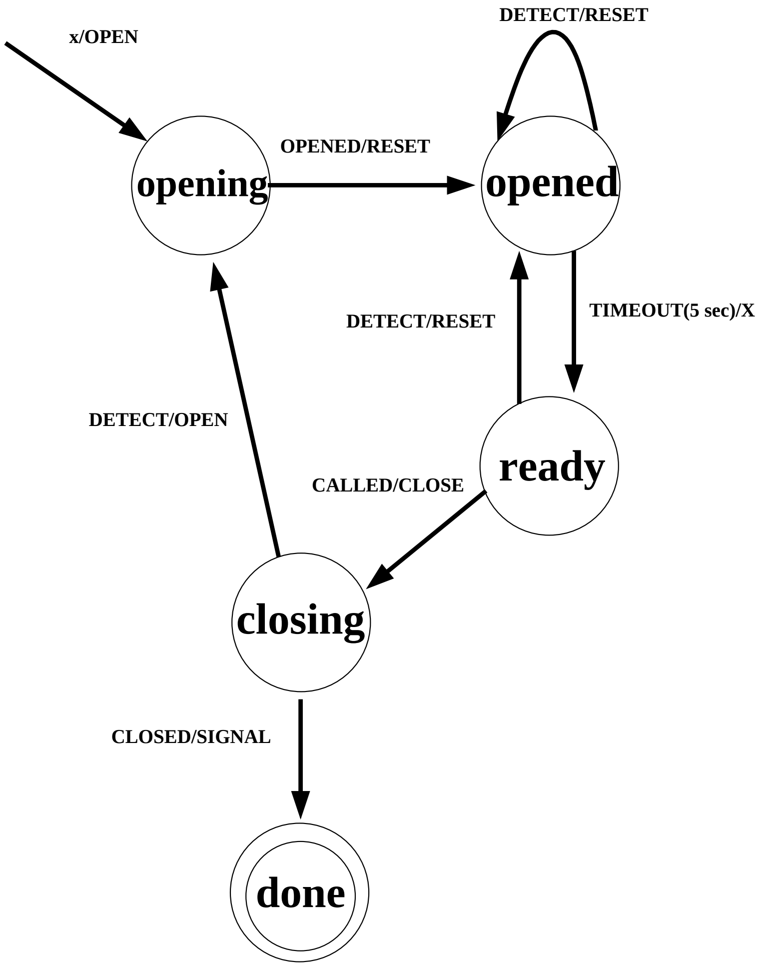
- The door may be commanded to CLOSE or OPEN. Once commanded, these take time to complete.
- A sensor indicates when the door is CLOSED. Once closed, a SIGNAL must be sent to the elevator monkeys to indicate that they may move the elevator car.
- A sensor indicates when the door is OPENED.
- A timer may be RESET or may TIMEOUT(t).
- The door has a sensor that detects whether a passenger has crossed the threshold of the door (DETECT).
- The door must remain open until a request is received to move the elevator (CALLED).
- Once a passenger has been detected, the door must be opened (if it isn't already). After it is opened, the door must wait at least 5 seconds before starting to close.
- On initialization, the FSM must ensure that the door is open before doing anything else.
- The FSM can do nothing (Z).
- The FSM also has a non-event (X).

All of the possible events and actions are listed above in bold.

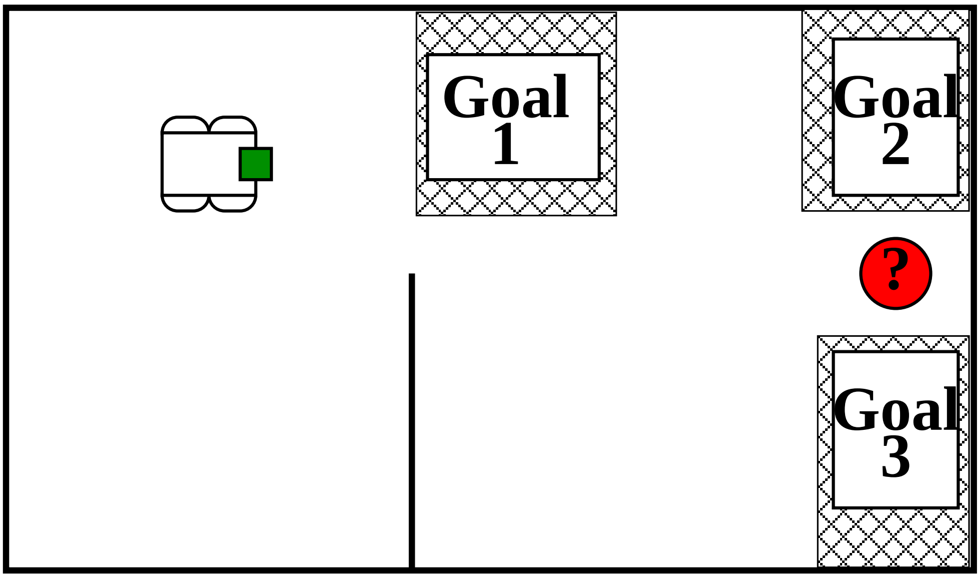
(a) (5 pts) Which are the events?

- *CLOSED*
- *OPENED*
- *TIMEOUT(t)*
- *DETECT*
- *CALLED*
- *X*

(b) (20 pts) Draw the corresponding FSM



Consider the following robot world:



The robot is able to execute the following actions:

- Move forward (F): continues to move forward until another action is issued
- Stop (S)
- Turn left (L): initiate a turn to the left. This turn will complete exactly at 90 degrees within a short period of time
- Turn right (R)
- Grasp (G): grasp an object that is in front of the robot

The robot is sensitive to the following events:

- Bump Wall (BW): bumped into a wall
- Bump Red Ball (BR): bumped into a red ball
- Bump Green Ball (BG): bumped into a green ball
- Turn complete (TC)
- Grasp complete (GC)

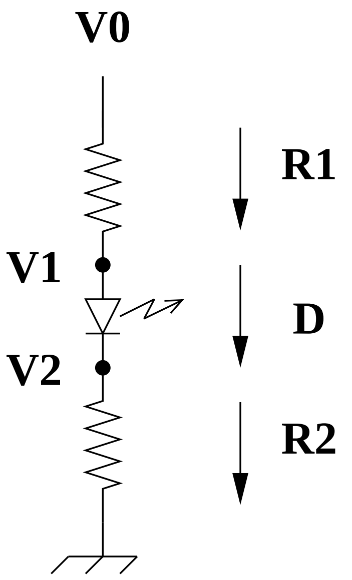
Th robot always starts in the indicated configuration (with the front facing to the right). A red or green ball is located at the red circle. Your task is to design a finite state machine that will navigate to and grasp the ball. If the ball is red, then the robot must navigate to goal 2 and stop. If the ball is green, then the robot must navigate to goal 1 and stop.

(c) (20 pts) Draw the FSM diagram that describes the behavior of this robot. Only use the given actions and events.

An exercise for the interested reader.

4. **Analog Circuits** (35 pts)

Consider the following circuit:



Assume that $R1 = 750\Omega$, $R2 = 250\Omega$, and $V_f = 2V$.

(a) (5 pts) List the equations that are **always** true.

$$V0 - V1 = I_1 R1 \quad (1)$$
$$V2 - 0 = I_2 R2 \quad (2)$$
$$I_1 = I_2 = I_D \quad (3)$$

(b) (15 pts) Assume that $V0 = 1.5V$. What are $V1$, $V2$ and I_D?

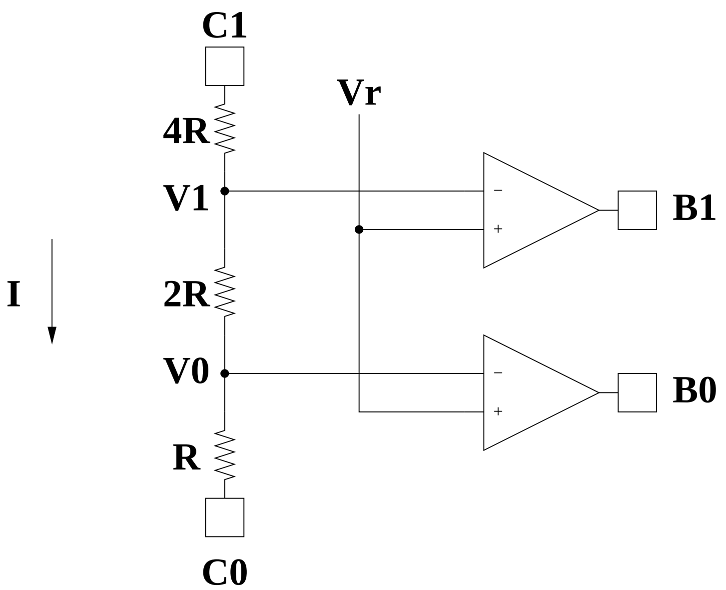
We will assume that the LED is turned off. This means that $I_D = 0$.
This means that there is no voltage drop across either of the resistors. Therefore: $V1 = 1.5V$ and $V2 = 0V$.
But: we must also check that $V1 - V2 < 2V$. This is true. Therefore, our original assumption holds.

(c) (15 pts) Assume that $V0 = 3V$. What are $V1$, $V2$ and I_D?

We will assume that the LED is turned on. This means that $V1 - V2 = 2V$.
Adding eq 2 and eq 1, we have:
$V0 - (V1 - V2) = I_1 R1 + I_2 R2$
Plugging in the above and using eq 3, we have:
$3 - 2 = I_D(R1 + R2)$
which means that $I_D = 1mA$.
Therefore:
$V2 = 0.25V$, *and*
$V1 = 2.25V$.
Checking our assumtion: $I_D > 0$. *Therefore, our original assumption holds.*

5. **Analog Processing** (40 pts)

Consider the following circuit:



C_1 and C_0 are logical values determined by your Atmel Mega processor (i.e., 0 and 1). The voltage at pin i is $5C_i$. B_1 and B_0 are inputs into the processor. Assume that the analog comparators produce an output of $0V$ or $5V$.

(a) (20 pts) For each combination of C_1 and C_0, derive the voltage at points V_1 and V_0.

Key observation: the inputs to the analog comparators require zero current. Therefore, we can treat the resistors separately and solve for V1 and V0.

Total resistance across the set of resistors is $7R$. *Therefore,* $I = \frac{5(C1-C0)}{7R}$.

When $C1 = C0$, *we have* $I = 0$. *Therefore,* $V1 = V0 = 5C1$.

When $C1, C0 = 1, 0$, *we have* $I = \frac{5}{7R}$.
Applying Ohm's law to the bottom resistor, we have $V0 = IR = \frac{5}{7}V$.
Applying Ohm's law to the bottom two resistors, we have $V1 = I3R = \frac{15}{7}V$.

When $C1, C0 = 0, 1$, *we have* $I = -\frac{5}{7R}$.
Applying Ohm's law to the bottom resistor, we have $V0 = 5 + IR = \frac{30}{7}V$.
Applying Ohm's law to the bottom two resistors, we have $V1 = 5 + I3R = \frac{20}{7}V$.

To summarize:

C_1	C_0	V_1	V_0
0	*0*	*0*	*0*
0	*1*	*20/7 V*	*30/7 V*
1	*0*	*15/7 V*	*5/7 V*
1	*1*	*5*	*5*

(b) (10 pts) Assume that $V_r = 2\ V$ and $C_1, C_0 = 10$. What are B_1 and B_0?

$V_r = 14/7V$.

Therefore: logic 0 and 1, respectively.

(c) (10 pts) Assume that a standard 5-bit digital-to-analog converter in which the minimum voltage is zero Volts and the maximum voltage is 5 Volts. Assume a digital value of 0x11. What is the corresponding voltage?

We know that $0V$ *is represented as zero and* $5V$ *is represented as 31 (the maximum value that can be represented with 5 bits).*
The equation for a standard D2A converter is:
$V_{out} = \frac{5}{31}(C_0 + 2C_1 + 4C_2 + 8C_3 + 16C_4)$
Since $0x11$ *is binary* 10001, $V_{out} = \frac{5 \times 17}{31} V$
(This happens to be: $\frac{85}{31} = 2.7419V$*)*

6. **Microprocessors** (25 pts)

(a) (5 pts) Briefly explain the function of the *instruction decoder*.

The instruction decoder translates the bit pattern of the currently executing instruction into a set of control signals for the different parts of the microprocessor.

(b) (5 pts) True or False and briefly explain. The status register is updated by the general purpose registers.

False. The status register is updated by the ALU as the result of some operation.

(c) (7 pts) List two ways in which the information in the status register may be used.

- *The carry bit can be used in an addition operation*
- *The zero or negative bits can be used to conditionally change the program counter.*

(d) (8 pts) List two key properties of every type of memory.

- *Stores values that can be read.*
- *Composed of multiple memory elements, each of which has a unique address.*