(2 pts) Name:

AME 3623: Embedded Real-Time Systems: Final Exam

May 13, 2009

- This examination booklet has 16 pages.
- Do not forget to write your name at the top of the page and to sign your name below.
- The exam is closed book, closed notes, and closed electronic device. The exception is that you may have one page of your own notes.
- The exam is worth a total of 200 points (and 20% of your final grade).
- Explain your answers clearly and be concise. Do not write long essays (even if there is a lot of open space on the page). A question worth 5 points is only worth an answer that is at most 1.5 sentences.
- You have 2 hours to complete the exam. Be a smart test taker: if you get stuck on one problem go on to the next. Don't waste your time giving details that the question does not request. Points will be taken off for answers containing extraneous information.
- Show your work. Partial credit is possible, but only if you show intermediate steps.

Problem	Topic	Max	Grade
0	Name	2	
1	Interrupts and I/O	45	
2	Finite State Machines	60	
3	Sequential Logic	35	
4	Analog Processing	10	
5	Microprocessor Design	30	
6	Logic	20	
Total		202	

On my honor, I affirm that I have neither given nor received inappropriate aid in the completion of this exam.

Signature: ______________________________

Date: ______________________________

1. **Interrupts and I/O** (45 pts)

   (a) (15 pts) Below is an interrupt service routine that is supposed to produce a signal of 50% duty cycle and a varible period on portD, pin 6. However, there exist several bugs (errors). Make the necessary changes to this code to remove these bugs. The number of counts for each "half cycle" is specified by the variable **duration** (i.e., for a single cycle, the pin should be 1 for **duration** counts and then 0 for another **duration** counts). You may assume that the I/O hardware has been initialized correctly.

```
volatile uint8_t duration;

ISR(TIMER0_OVF_vect) {
  static uint8_t counter = 0xff;

  ++counter;

  if(counter <= duration) {

     PIND &= 0x80;
  }
}
```

(b) (5 pts) Assume that the code above has been corrected. Given a prescaler of 8 and $duration = 40$, what is the resulting interrupt frequency? (set up the fraction, but do not reduce it)

(c) (5 pts) Given a prescaler of 8 and $duration = 15$, what is the frequency of the signal produced on the PORTD output pin? (again, set up the fraction, but do not reduce it)

(d) (5 pts) (True/False) In the Atmel Mega8, the highest overflow interrupt frequency of timer 0 is less than the highest interrupt frequency of timer 1. Explain.

```
uint16_t get(void) {
  uint16_t val = 0;
  char c;

  while(1) {
     c = getchar();
     if('0' <= c && c <= '9'){
         val = val * 16 + c - '0';
     }else if('a' <= c && c <= 'f'){
         val = val * 16 + c - 'a' + 10;
     }else{
         return(val);
     }
  }
}
```

(e) (5 pts) What does the function **get()** do when the following characters are sent through the serial port: '5', '\n'?

(f) (5 pts) What does the function **get()** do when the following characters are sent through the serial port: '5', 'e', '\n'?

(g) (5 pts) Explain what the function **get()** does in general. Be brief.

2. **Finite State Machines** (60 pts)

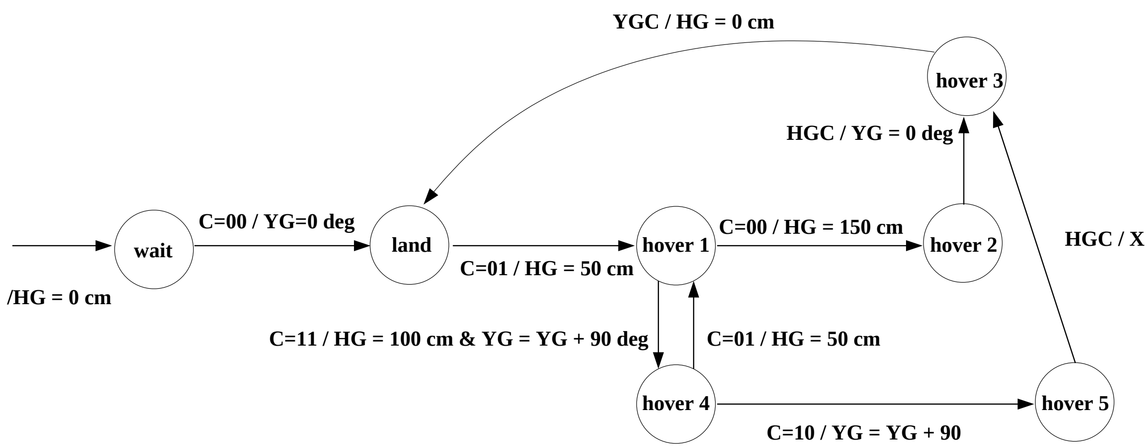
Consider a finite state machine controller of a helicopter not unlike the ones that we used for our projects. This controller has access to the following actions:

- HG = xxx cm. Set the height goal to xxx. You may assume that the underlying PID controller will eventually bring the height to the currently set goal.
- YG = yyy deg. Set the yaw goal to yyy. The associated PID controller will eventually bring the true yaw to the yaw goal.
- X. Do nothing

The FSM also has the following events:

- HGC. Height goal complete. The helicopter has reached its height goal.
- YGC. Yaw goal complete.
- C = bb. A control input that can be one of four different values: 00, 01, 10 or 11 (so, 4 different events). This control input is set by some remote pilot.

The FSM is as follows:



Assume that events that are not listed result in the FSM remaining in the current state.

(a) (5 pts) Starting from the *wait* state, assume that the following commands are issued in this order: 00, 01, 00. What is the state that the FSM stops in?

(b) (5 pts) What is the orientation of the heli at this state?

(c) (5 pts) What is the maximum height obtained by the heli during the sequence?

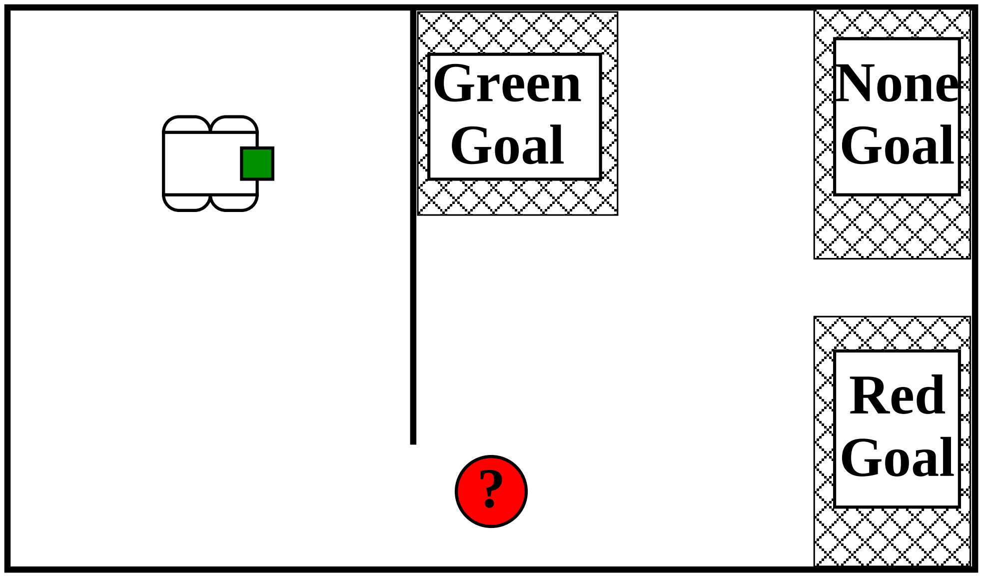
(d) (5 pts) Starting from the *wait* state, assume that the following commands are issued in this order: 00, 01, 11, 10. What is the state that the FSM stops in?

(e) (5 pts) What is the orientation of the heli at this state?

(f) (5 pts) What is the maximum height obtained by the heli during the sequence?

(g) (10 pts) What sequence of commands would result in a landing orientation of 270 degrees?

Consider the following robot world in which the mobile robot is facing to the right.



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
- Turn right (R)
- Grasp (G): grasp an object that is in front of the robot

The robot is sensitive to the following events:

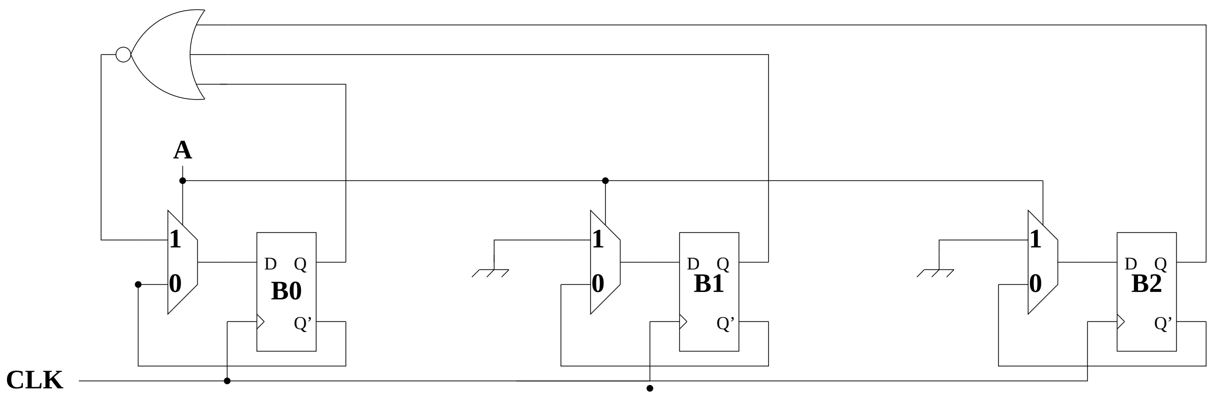
- Bump Wall (BW): bumped into a wall
- Bump Red Ball (BR): bumped into a red ball
- Bump Green Ball (BG): bumped into a green ball
- Turn complete (TC)
- Grasp complete (GC)

Your task is to design a finite state machine that takes the robot from its current position to one of the goal locations. If the robot runs into a ball along the way, it should also grasp it and carry it to the corresponding goal. If it does not encounter a ball, then the robot should stop in the *None goal* region. Note: the ball can only occur at the designated location.

(h) (20 pts) Draw the finite state machine diagram that will control the robot through this sequence. Note that your FSM should perform properly in all three cases.

3. **Sequential Logic** (35 pts)

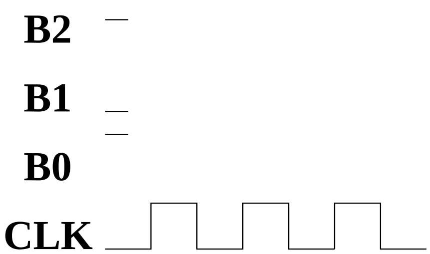
Consider the following circuit:



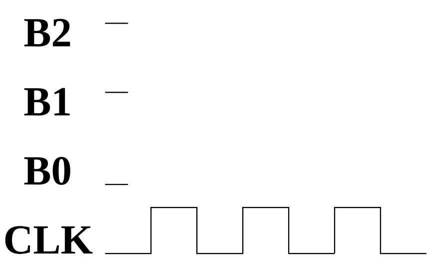
(a) (15 pts) Fill in the following truth table:

A	B_2	B_1	B_0	D_2	D_1	D_0
0	0	0	0			
0	0	0	1			
0	0	1	0			
0	0	1	1			
0	1	0	0			
0	1	0	1			
0	1	1	0			
0	1	1	1			
1	0	0	0			
1	0	0	1			
1	0	1	0			
1	0	1	1			
1	1	0	0			
1	1	0	1			
1	1	1	0			
1	1	1	1			

(b) (5 pts) Assuming that $A = 0$ and an initial state of $B_2, B_1, B_0 = 101$, fill in the following timing diagram:



(c) (5 pts) Assuming that $A = 1$ and an initial state of $B_2, B_1, B_0 = 110$, fill in the following timing diagram:

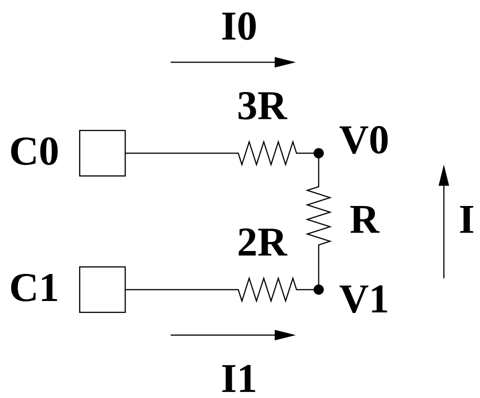


(d) (5 pts) Interpreting B_2, B_1, B_0 as a 3-bit binary number, when $A = 0$, what mathematical operation does this circuit perform when the clock changes from high to low? Your answer should apply to all possible combination of bits.

(e) (5 pts) When $A = 1$, what mathematical operation does this circuit perform when the clock changes from high to low?

4. **Analog Processing** (10 pts)

Consider the following digital-to-analog circuit:



(a) (10 pts) Derive an equation for V_1 as a function of C_1 and C_0. Show your work.

5. **Microprocessor Design** (30 pts)

(a) (5 pts) Briefly explain the function of the *chip select* signal for a memory chip.

(b) (5 pts) (True/False) The data line is an output from a memory chip. Explain.

(c) (5 pts) (True/False) In the Atmel Mega8, PIND is a general purpose register. Explain.

(d) (5 pts) (True/False) The arithmetic logical unit receives arguments (values) from the general purpose registers. Explain.

(e) (10 pts) What is the value of variable *baz* at the end of this segment of code (in hexadecimal)?

```
foo = 0xe5;

bar = 135;

baz = foo & bar | 29;
```

6. **Logic** (20 pts)

Given the following truth table:

A	B	C	f
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

(a) (5 pts) Give the "minterm" form of the corresponding algebraic expression.

(b) (10 pts) Derive a simplified algebraic description for f. Justify each step (provide the name of the rule that you are using).

(c) (5 pts) Draw the corresponding circuit.