(2 pts) Name:

AME 3623: Embedded Real-Time Systems: Final Exam

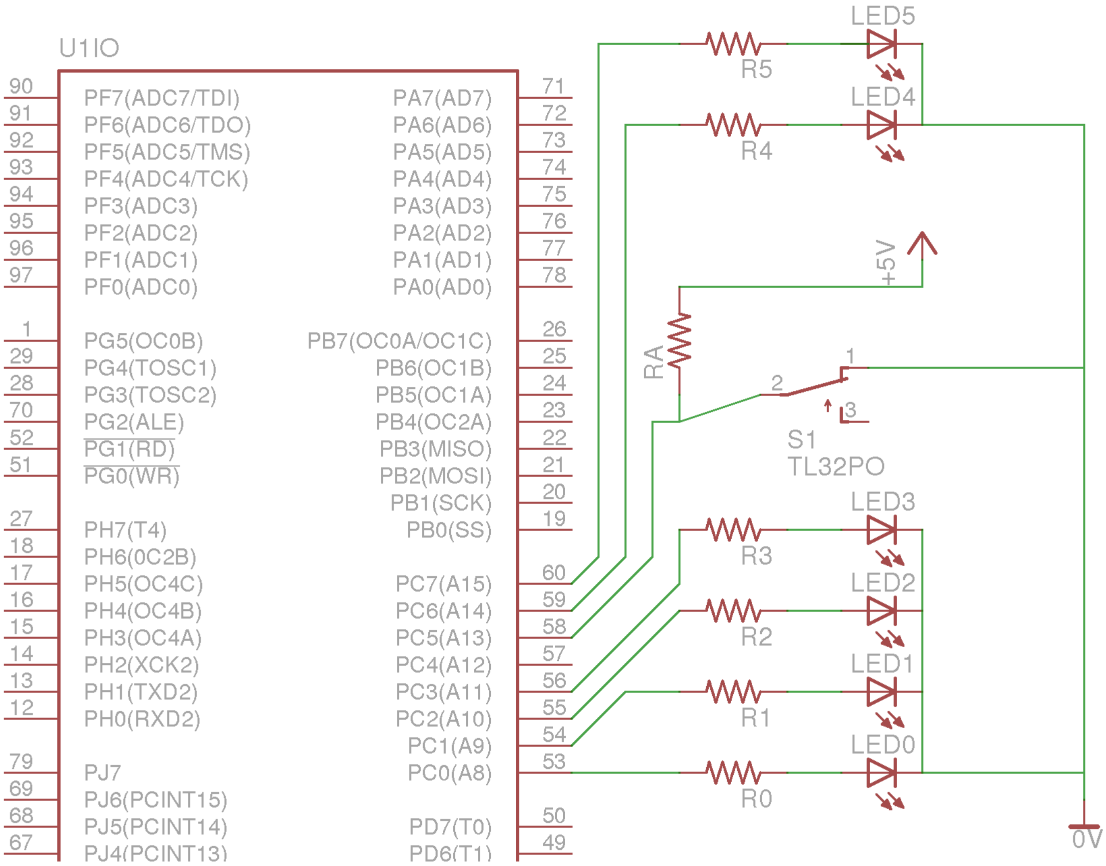
Solution Set

May 10, 2011

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	Device Control	30	
5	Analog Processing	45	
6	Microprocessors	25	
7	Extra Credit	10	
Total		212	

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

   Carefully consider the following circuit:



And consider the following program:

```
ISR(TIMER2_OVF_vect) {
  static uint16_t counter = 0;

  if((counter / 3) == 1) {
     PORTC |= 0x40;
     PORTC ^= 0x80;
  }else{
     PORTC &= ~0x40;
  }

  PORTC ^= 0x02;

  ++counter;

  if(counter == 8)
     counter = 0;
}

int main(void) {
  DDRC = 0xCF;
  PORTC = 0;
  timer2_config(TIMER2_PRE_128);
  timer2_enable();
  sei();

  while(1) {
  }
}
```

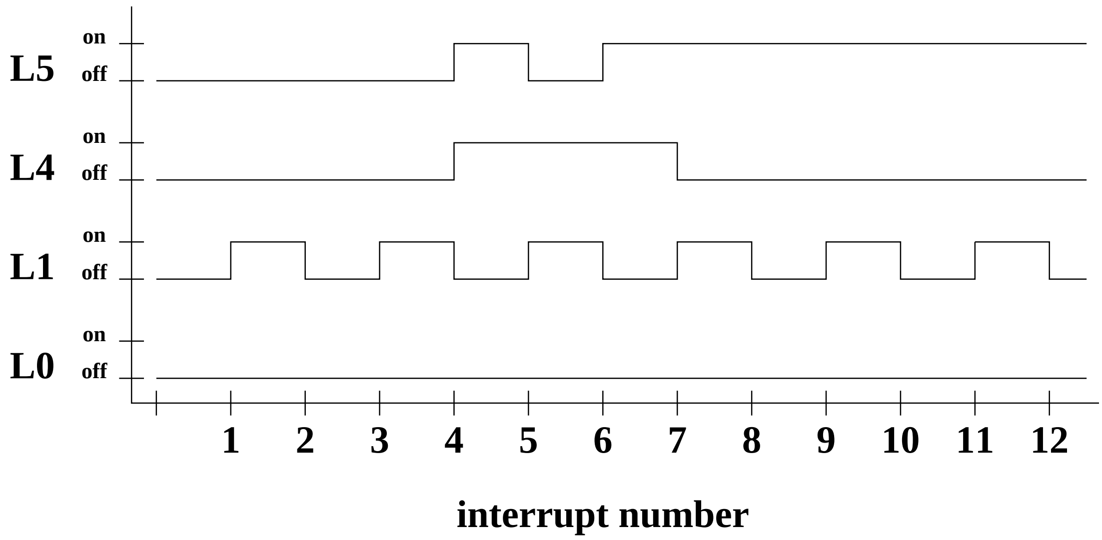
(a) (5 pts) Assuming a system clock of $16MHz$, at what frequency is the timer 2 counter incrementing? (yes, work out the long division by hand)

$\frac{16{,}000{,}000\ ticks/sec}{128\ ticks/tock} = 125,000\frac{tocks}{sec}$

(b) (5 pts) At what frequency is the timer 2 overflow interrupt being generated? (set up the ratio only)

$\frac{16{,}000{,}000\ ticks/sec}{128\ ticks/tock \times 256\ tocks/int}$

(c) (20 pts) Show the state of LEDs 0, 1, 4 and 5 as a function of interrupt number for interrupts 1 through 12.



(d) (5 pts) What is the flashing frequency of LED 1? (set up the ratio)

$\frac{16{,}000{,}000\ ticks/sec}{128\ ticks/tock \times 256\ tocks/int \times 2\ int/cycle}$

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

(a) (5 pts) What is the binary equivalent of hexadecimal 35? Show your work.

110101

(b) (5 pts) What is the binary equivalent of hexadecimal -35 in 8-bit two's complement? Show your work.

11001011

(c) (5 pts) Compute $47 - 0x35$ using binary arithmetic. Show your work and give your answer in binary two's complement.

47 = 0b00101111

```
  00101111
+ 11001011
----------
  11111010
```

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

```
  11111010

This is a negative number.  The two's complement inverse is 110, which
is decimal 6.

So: the original result is -6.
```

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

Consider the problem of controlling the traffic lights for an intersection between a road oriented North/South and another road oriented East/West. The rules of behavior are as follows:

- Each of the light displays can show *red, yellow, or green.*
- The North- and South-facing lights show the same color.
- The East- and West-facing lights show the same color.
- The "green direction" only changes if a car is detected as waiting at the opposing red light. A light must be green at least 30 seconds before going yellow in response to a detected car.
- If the lights in one direction are green or yellow, then the opposite direction **must** be red.
- A light must be yellow for 20 seconds before changing to red.
- After a light has changed to red, the opposing light turns green after 15 more seconds.
- Any time that an "emergency" signal is received, any green light must immediately transition to yellow and then, after 20 seconds, to red.
- After the emergency signal has been removed, the light sequence rules return to normal operation.

Note: the FSM must explicitly handle the timing of its actions.

The actions are as follows:

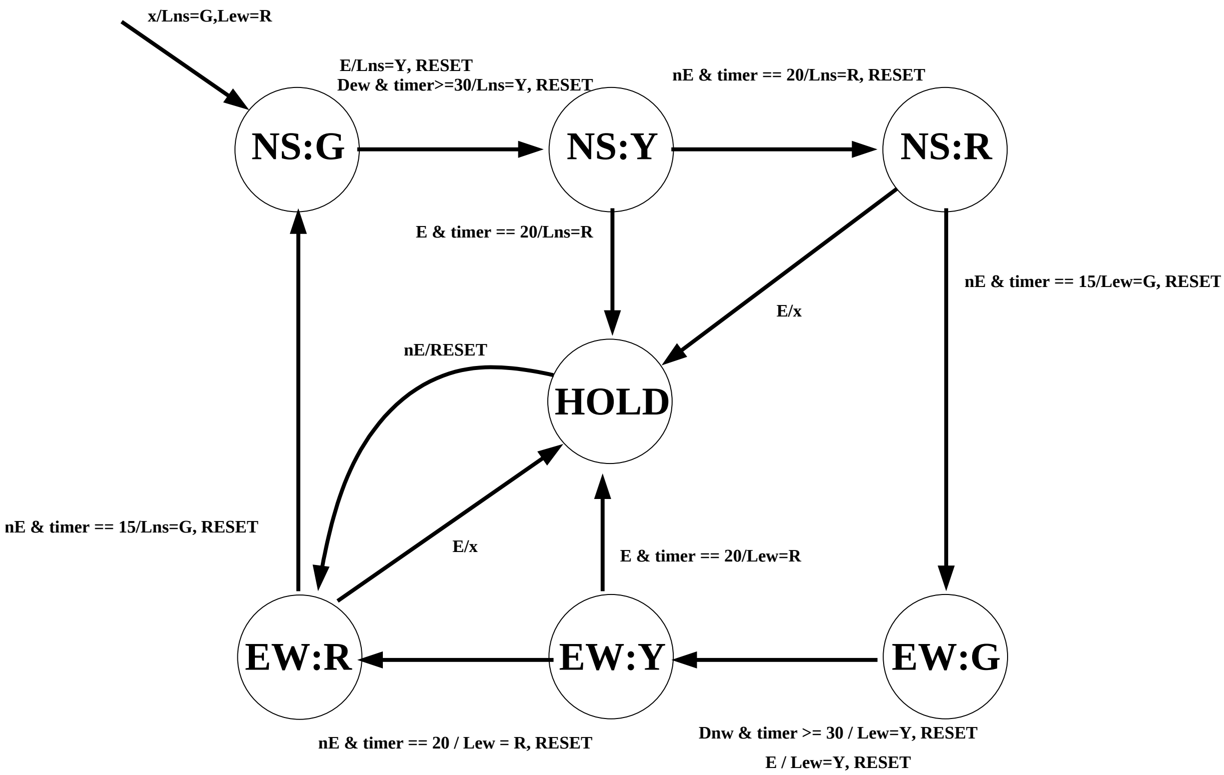
- $L_{ns} = R$
- $L_{ns} = Y$
- $L_{ns} = G$
- $L_{ew} = R$
- $L_{ew} = Y$
- $L_{ew} = G$
- RESET timer

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

  - $Timer = 15$
  - $Timer = 20$
  - $Timer \geq 30$
  - *N/S car detected (Dns)*

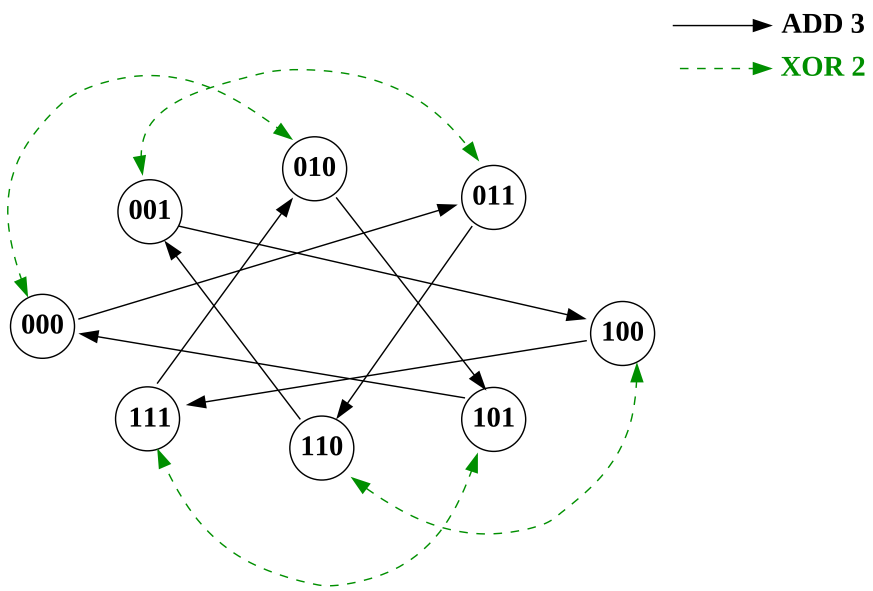
- *E/W car detected (Dew)*
- *Emergency (E)*
- *Not emergency (nE)*

(b) (20 pts) Draw the corresponding FSM



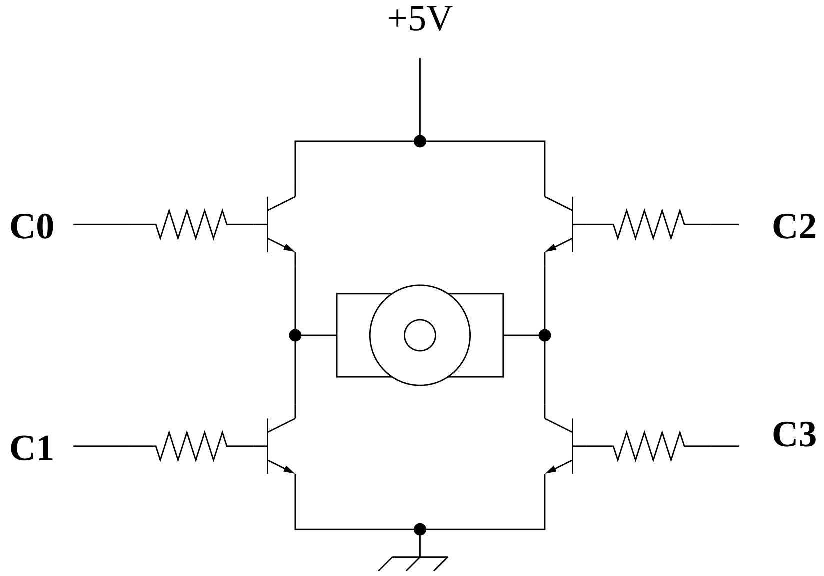
Consider a FSM whose state is described by a three-bit binary number (i.e., there are 2^3 states). This finite state machine has two possible events: XOR with 0x2 (X) and ADD 3 (A).

(c) (20 pts) Draw the FSM diagram that describes the behavior of this device when the different events occur (don't worry about actions in this case).



4. **Device Control** (30 pts)

Given the following H-bridge circuit:



(a) (10 pts) Briefly explain the constraints on the chosen **logical** values of C_0, C_1, C_2, and C_3 for safe operation of the H-bridge.

C_0 and C_1 cannot both be logic 1 at the same time, and C_2 and C_3 cannot both be logic 1. (otherwise, a short circuit will be created).

(b) (10 pts) Briefly describe how motor torque direction is controlled with this circuit.

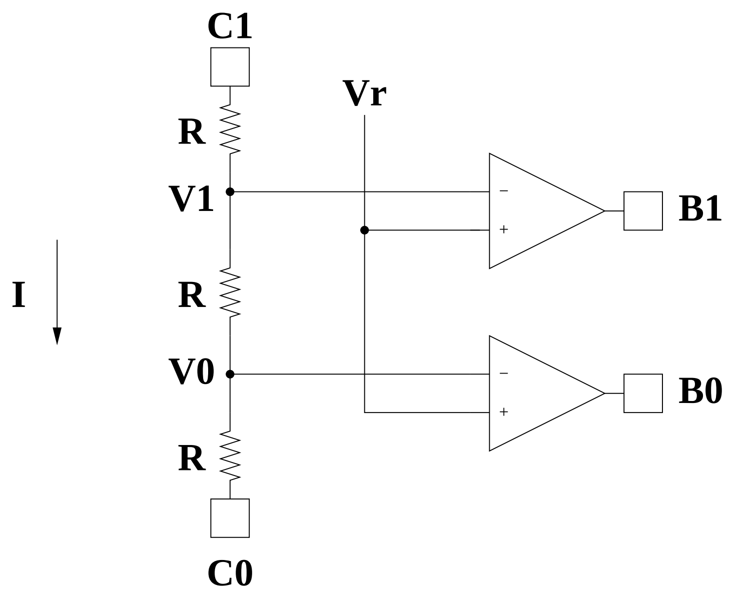
Setting C_0 and C_3 to logic 1 will cause a torque in one direction to be produced. Setting C_1 and C_2 to logic 1 will cause a torque in the opposite direction.

(c) (10 pts) Briefly describe how torque magnitude is controlled in this circuit.

C_0 and C_3 OR C_1 and C_2 are pulsed on and off at some high frequency. The duty cycle of this PWM signal is linearly related to the average torque magnitude (in the ideal case, anyway).

5. **Analog Processing** (45 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 operational amplifiers produce an output of $0V$ or $5V$.

(a) (15 pts) For each combination of C_1 and C_0, derive the voltage at points V_1 and V_0.

C_1	C_0	V_1	V_0
0	*0*	*0*	*0*
0	*1*	*5/3 V*	*10/3 V*
1	*0*	*10/3 V*	*5/3 V*
1	*1*	*5*	*5*

(b) (5 pts) Assume that $V_r = 7/3\ V$ and $C_1, C_0 = 01$. What are B_1 and B_0?

logic 1 and 0, respectively.

(c) (5 pts) Assume that $V_r = 11/3\ V$ and $C_1, C_0 = 11$. What are B_1 and B_0?

logic 0 and 0, respectively.

(d) (5 pts) Assume that a 5-bit digital-to-analog converter in which the minimum voltage is zero Volts and the maximum voltage is 5 Volts. What is the digital value that corresponds to 1 Volt? (give the fraction only)

We know that $0V$ is represented as zero and $5V$ is represented as 31. Therefore, for one Volt:

$31/5 = 6$ *(note that this is integer division)*

(e) (15 pts) Assume that a 5-bit analog-to-digital converter is presented with an input voltage of $120/31\ V$. Show the sequence of guesses that the successive approximation algorithm goes through in order to determine the equivalent binary representation.

10000	$5\frac{16}{31} = \frac{80}{31}$	*too low*
11000	$5\frac{16+8}{31} = \frac{120}{31}$	*too high (by our rule of greater than or equal to)*
10100	$5\frac{16+4}{31} = \frac{100}{31}$	*too low*
10110	$5\frac{16+4+2}{31} = \frac{110}{31}$	*too low*
10111	$5\frac{16+4+2+1}{31} = \frac{115}{31}$	*too low (but we keep it)*

6. **Microprocessors** (25 pts)

(a) (5 pts) Briefly explain the function of the *program counter*.

The program counter tells the microprocessor which instruction is to be executed next.

(b) (5 pts) True or False and briefly explain. The arithmetic logical unit receives arguments (values) from the general purpose registers.

True. This is the only way for arguments to be presented to the ALU. After computing the value, it also places the result into one of the general purpose registers.

(c) (5 pts) True or False and briefly explain. Variables declared in a function are stored in flash memory.

False. They are stored in RAM.

(d) (10 pts) Give four examples of special-purpose registers in the Atmel Mega processors and briefly describe their function.

- *DDRx = this general purpose register determines whether a pin is a digital input or output.*
- *PORTx = if a pin is configured as an output, this register determines whether the pin is pulled to* $0V$ *or* $5V$.
- *PINx = this register allows one to read the state of a pin.*
- *program counter (defined above).*
- *instruction register = the register that stores the bit representation of the instruction that is currently being executed.*

7. **Extra Credit** (10 pts)

(10 pts) (credit to Click & Clack) You are presented with seven stacks of 70 coins each. Each stack is composed of either genuine or counterfeit coins (all coins in a stack are the same). A genuine coin weighs 10 grams; a counterfeit coin weighs 11 grams. You have a scale that is accurate to 0.1 grams. Describe a scheme in which you can determine which stacks are genuine and which are counterfeit using a single weighing.

Hints:

- From a given stack, you may select any number of coins to include in the weighing.
- The solution to this problem is related to your implementation of get_orientation(), with binary numbers replacing hexadecimal numbers.

Select one coin from the first stack, two from the second stack, four from the third stack, eight from the fourth stack, 16 from the fifth, 32 from the sixth, and 64 from the seventh. Let x be equal to the weight of the set of coins minus 1270 (the weight if all coins were genuine). Each bit of x will indicate whether the corresponding stack is counterfeit (0 = genuine; 1 = counterfeit).