(2 pts) Name:

AME 3623: Embedded Real-Time Systems: Final Exam

Solution Set

May 9, 2006

Problem	Topic	Max	Grade
0	-	2	
1	Interrupts and I/O	40	
2	Multi-Tasking	46	
3	Finite State Machines	45	
4	Analog Processing	28	
5	Microprocessor Design	15	
6	Logic	26	
Total		202	

1. **Interrupts and I/O** (40 pts)

(a) (5 pts) Define (in brief) "bus arbitration."

Bus arbitration is the process by which the system establishes which device is currently in control in the bus.

(b) (10 pts) For the given clock rate and prescaler configuration of the mega8 timer0, give the time (in microseconds) between interrupts (reduced fractions and exponents of 2 are fine).

Prescaler	*1 MHz clock*	*16 MHz clock*
No prescaler	$2^8 \mu s$	$2^4 \mu s$
Div 8	$2^{11} \mu s$	$2^7 \mu s$
Div 64	$2^{14} \mu s$	$2^{10} \mu s$
Div 256	$2^{16} \mu s$	$2^{12} \mu s$
Div 1024	$2^{18} \mu s$	$2^{14} \mu s$

(c) (6 pts) List at least three operations that are performed by the processor in response to an interrupt

- *Stop execution of the running program*
- *Save the program counter (on the stack)*
- *Begin execution of the interrupt routine*
- *The first thing that the routine will do is save the state of the general-purpose registers that will be used*

(d) (5 pts) Explain (in brief) the purpose of a buffer.

A buffer is a temporary storage area for incoming (or outgoing data). The buffer allows (in the case of incoming data) for a delay between the time that the data arrives and the time that the data is processed.

(There is also a hardware buffer answer.)

(e) (14 pts) Below is an interrupt routine that is supposed to produce a pulse-width-modulated signal on port C, pin 5 (counting from 0). However, there exist several bugs (errors). Make the necessary changes to this code to remove these bugs. The frequency should be constant; the width of the pulse is specified by the variable **duration**. You may assume that the hardware has been initialized correctly.

```
volatile unit8_t counter;
volatile uint8_t duration;   // Duration of the high side of the pulse

SIGNAL(SIG_OVERFLOW0) {
  ++counter;
  if(counter == 0)
     PORTB &= 0x10;          // Turn on line
  if(counter <= duration) {
     PORTB |= ~0x10;         // Turn line off
     counter = 0;
  }
}
```

The repaired code is as follows:

```
volatile unit8_t counter;
volatile uint8_t duration;   // Duration of the high side of the pulse

SIGNAL(SIG_OVERFLOW0) {
  ++counter;
  if(counter == 0)
     PORTC |= 0x20;
  if(counter >= duration) {
     PORTC &= ~0x20;
  }
}
```

Key changes: must get the port and bit mask correct; must get masking right (swap AND and OR operators); remove counter $= 0$ *line; must fix the duration test*

2. **Multi-Tasking** (46 pts)

(a) (8 pts) List two necessary conditions for there to be a shared data problem.

i. The data structure is accessed both by the main program and by an interrupt routine (more generally: a data structure is accessed by two or more processes).

ii. The interrupt routine can be executed while the main program is in the middle of an access to the shared data structure (more generally: the processes accessing the shared data structure can execute asynchronously).

(b) (8 pts) What is the distinction between a process and a processor?

A process consists of the program and the memory/register state (it is a passive entity).

A processor is the active entity that executes the program.

Consider three processes that are involved in the control of a helicopter:

- **Planner** uses information about the current location of the craft, the surrounding terrain, and the immediate goals to determine a course of flight. This computation is performed once per second and requires $400ms$ to execute.
- **Yaw Control** takes input from a digital compass at $20Hz$, filters this signal, and computes a rudder command. This computation requires $10ms$ in the worst case.
- **Local Navigation** takes input from a camera, processes the images, and produces lateral velocity signals. This process needs to be scheduled at $5Hz$ and requires $50ms$ in the worst case.

(c) (15 pts) Show the schedule that results from a round-robin, timesliced scheduler for time 0 through time $160ms$ (show both "running" and "ready" processes). Assume timeslice duration of $10ms$, and that the processes are initially in the ready queue as ordered above. Also assume that at the end of a timeslice, the exiting process (if it still needs to execute) is placed on the ready queue before any newly-scheduled processes.

Time (ms)	Running	Ready	Comments
0	T1a	T2a, T3a	
10	T2a	T3a, T1a	
20	T3a	T1a	T2a has now completed
30	T1a	T3a	
40	T3a	T1a	
50	T1a	T3a, T2b	T2 is rescheduled
60	T3a	T2b, T1a	
70	T2b	T1a, T3a	
80	T1a	T3a	T2b has now completed
90	T3a	T1a	
100	T1a	T3a, T2c	T2 is rescheduled
110	T3a	T2c, T1a	
120	T2c	T1a	T3a has now completed
130	T1a		T2c has now completed
140	T1a		
150	T1a	T2d	T2 is rescheduled
160	T2d	T1a	

(d) (15 pts) Show the schedule (including "running" and "ready" processes) that results from the Rate Monotonic Scheduling algorithm.

This algorithm is priority based with preemption. The priorities are as follows: T2, T3, T1

Time (ms)	Running	Ready	Comments
0	T2a	T3a, T1	
10	T3a	T1a	T2a has now completed
20	T3a	T1a	
30	T3a	T1a	
40	T3a	T1a	
50	T2b	T3a, T1a	T2 is rescheduled
60	T3a	T1a	T2b has now completed
70	T1a		T3a has now completed
80	T1a		
90	T1a		
100	T2c	T1a	T2 is rescheduled
110	T1a		T2c has now completed
120	T1a		
130	T1a		
140	T1a		
150	T2d	T1a	T2 is rescheduled
160	T1a		T2d is now completed

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

Consider a robot that is facing down a hallway and has the ability to detect doorways using a sensor that is mounted on a turret (not unlike the turrets on our robots). In addition, the robot is able to detect when it runs into a wall. Assuming an ability to make perfect movements, we will design a finite state machine that will take the robot to the third door on the left side of the hallway, turn, enter it, and stop (the robot knows that it has moved through the door when it can no longer "see" the door in front of it). In addition, if there is ever a collision, the robot should stop and signal an error condition.

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

- No doors found (D0)
- One door found (D1)
- Two doors found (D2)
- Three doors found (D3)
- done
- error

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

- Door in view
- No door in view
- Collision

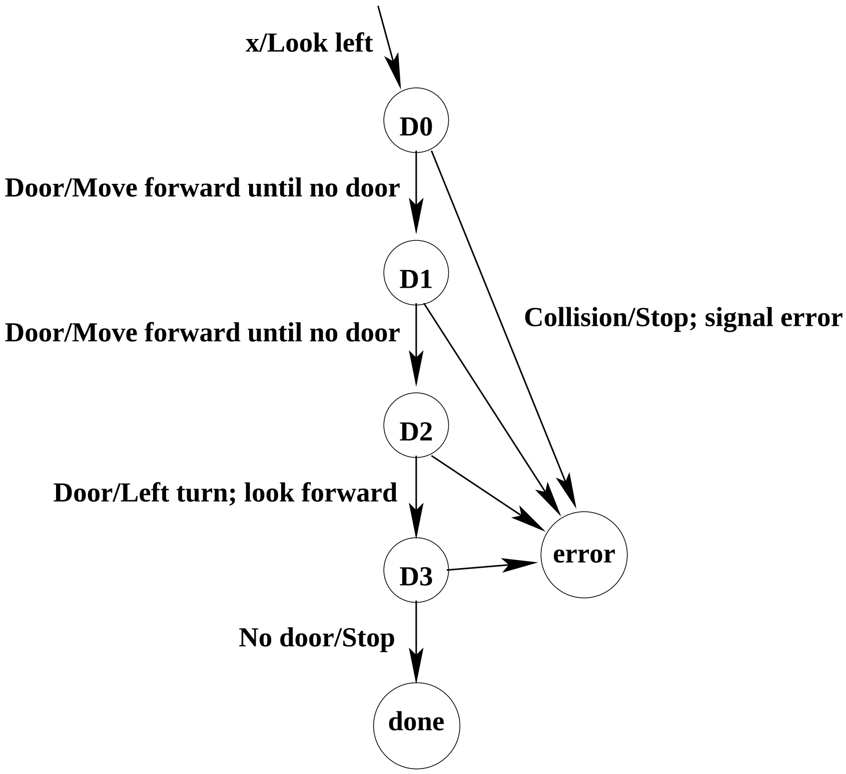
(c) (5 pts) What are the actions? You are responsible for motion of the robot and of the turret (you may separate these in your action list).

Actions can be expressed as sequences of:

- Move forward
- Move forward until no door
- Left turn
- Stop
- Signal error
- Look left (turret)
- Look forward (turret)

(d) (10 pts) Show the finite state machine.

When the robot is in states D0-D3, it is taking a forward motion action.



Suppose that instead of the third door from the robot's initial location, we want to move to the third door from the end of the hallway (along the same direction as the robot's initial orientation). Modify your FSM design to accommodate this.

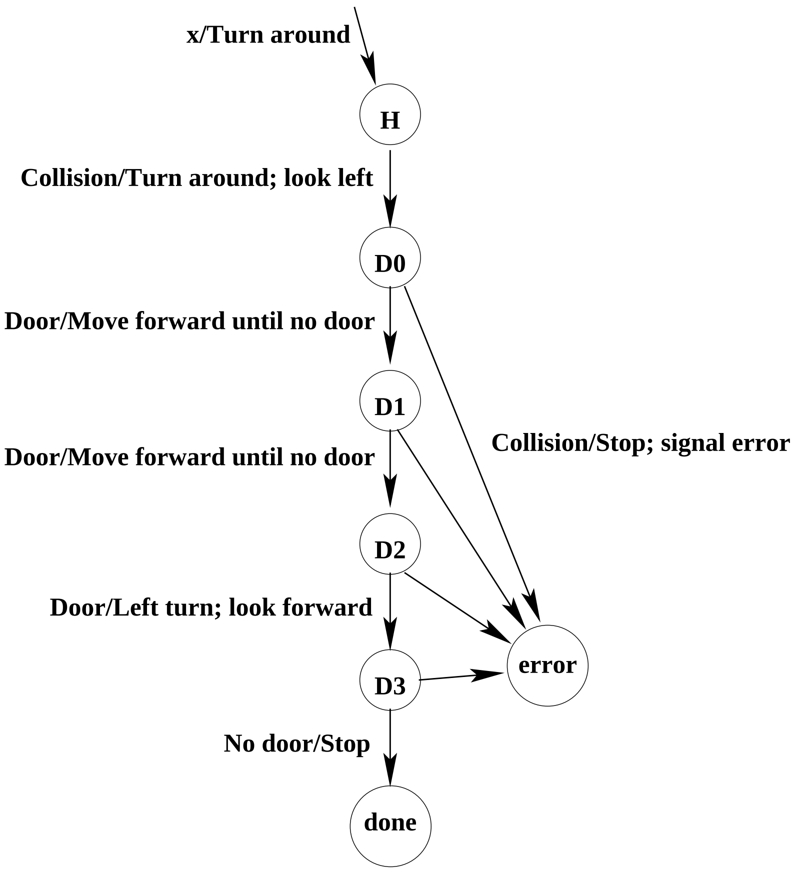
(e) (5 pts) What are the additional states?

- Move along hall (H)

(f) (5 pts) What are the additional actions?

- Turn around

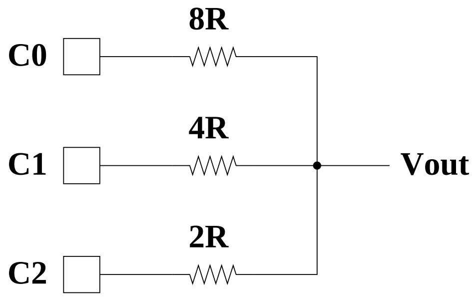
(g) (10 pts) Show the modified FSM.



(We will also accept the solution in which the robot moves forward to the end of the hall and then turns around.)

4. **Analog Processing** (28 pts)

Consider the following digital-to-analog conversion circuit:



(a) (10 pts) Give the expression for V_{out} in terms of binary digits $C0$, $C1$, and $C2$.

Note that the extra factor of 2 is divided out.

$V_{out} = \frac{5}{7}(C_0 + 2C_1 + 4C_2)$

(b) (8 pts) Given that a process wants to generate a voltage $V_{out} = 4V$. What choice of binary values could it use to best approximate this value?

$C[2, 1, 0] = 110$ would yield a voltage of $30/7V$ (which is $4.3V$)

(c) (10 pts) Assume an analog-to-digital converter that uses successive approximation to estimate the digital equivalent of an input voltage. Assume also that the 3-bit digital-to-analog converter from above is used. Given an input voltage of $2V$, show the steps taken by the successive approximation algorithm, as well as its final choice.

Target is $14/7V$

C_0	C_1	C_0	V	
1	0	0	$20/7V$	too high
0	1	0	$10/7V$	low
0	1	1	$15/7V$	too high
0	1	0	$10/7V$	ANSWER

5. **Microprocessor Design** (15 pts)

(a) (5 pts) What is the function of the program counter?

The program counter contains the address of the memory location that contains the instruction that is currently being executed.

(b) (5 pts) What is the function of the general-purpose registers?

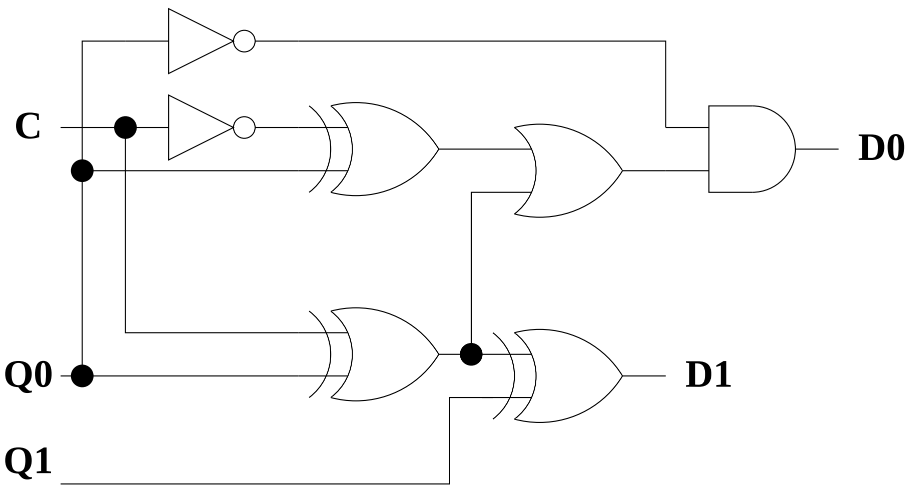
General purpose registers are a fast form of memory that are used to temporarily store results. In the Atmel Mega8 processor (and many others), these registers provide the inputs into components such as the Arithmetic Logical Unit.

(c) (5 pts) (True/False) The clock signal is used by a RAM during a "read" operation from the device. Explain.

False. As soon as the chip is selected and the read operation is selected, the chip will begin to drive the data bus.

6. **Logic** (26 pts)

Given the following circuit:



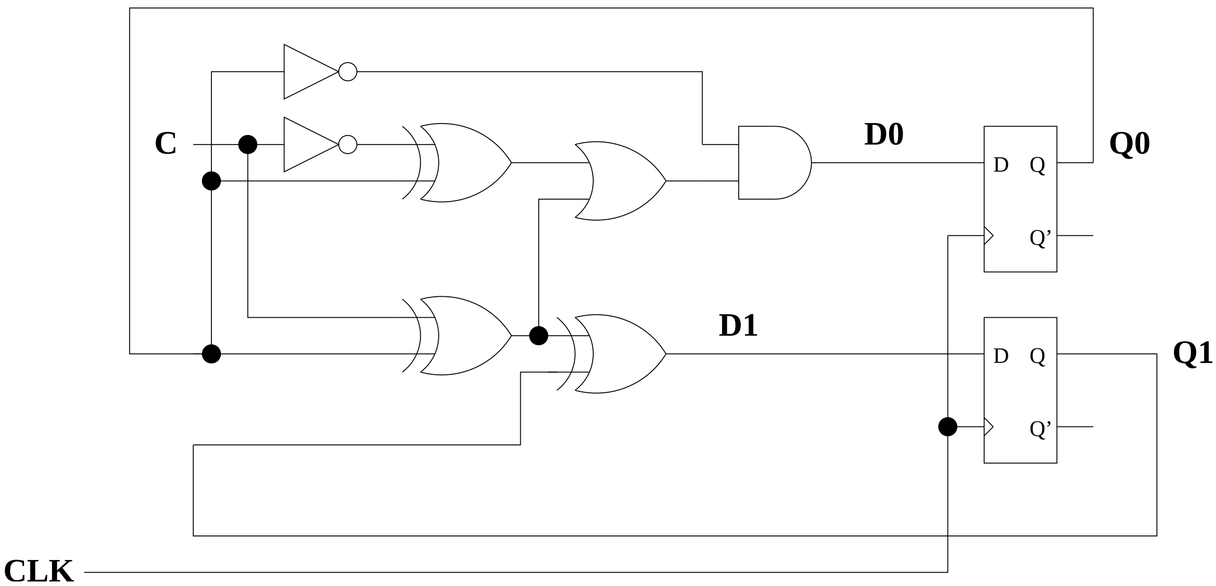
(a) (10 pts) What is the truth table?

C	$Q1$	$Q0$	$D1$	$D0$
0	0	0	0	1
0	0	1	1	0
0	1	0	1	1
0	1	1	0	0
1	0	0	1	1
1	0	1	0	0
1	1	0	0	1
1	1	1	1	0

(b) (8 pts) Give the simplified algebraic expression for $D0$

$D0 = \overline{Q0}$

Now consider the following circuit:



(c) (8 pts) Assume that $C = 1$ and the initial state is $Q1 = 1$ and $Q0 = 0$. What is the sequence of states for five clock ticks (CLK transitions from high to low)?

$Q[10] = 10; 01; 00; 11; 10; 01$