

Jonathan W. Valvano

First: _____ Last: _____

April 9, 2003, 1:00pm-1:50pm

(30) Question 1. Place one letter A-PP for each. (3 points each)

Arm key wakeup J bit 3.	
Acknowledge key wakeup J bit 7.	
Enable interrupts.	
Disable interrupts.	
Select falling edge of key wakeup PJ7.	
Select rising edge of key wakeup PJ3.	
Select pullup resistor on PJ7.	
Select pulldown resistor on PJ3.	
Toggle PORTJ bit 0.	
Disarm key wakeup J bit 7.	

(25) Question 2. Show the C code that executes the finite state machine.

(5) Question 3a.	Letter A-F
(5) Question 3b.	Letters A-E
(5) Question 3c.	Resistor value, in Ω
(5) Question 4.	Letter A-D

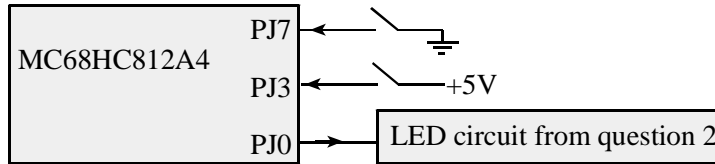
(25) Question 5. Design C code that spins two stepper motors at about 1 rotation per sec.

```
unsigned short volatile Ack;
#pragma interrupt_handler RTIHan()
void RTIHan(void){
    PORTT |= 0x01;
    RTIFLG = 0x80;
    if(Ack==1){
        Ack = 0;
    }
    PORTT &= ~0x01;
}
#pragma abs_address:0xffff0
void (*RTI_vector[])() = { RTIHan };
#pragma end_abs_address
void RTIinit(void){
    asm(" sei");
    COPCTL = 0x00;
    DDRT |= 0x41;
    TSCR = 0x80;
    RTICTL = 0x81;
    Ack = 1;
    asm(" cli");
}
```

Jonathan W. Valvano April 9, 2003, 1:00pm-1:50pm

This is a closed book exam. You must put your answers on the special answer pages only. You have 50 minutes, so please allocate your time accordingly. *Please read the entire quiz before starting.*

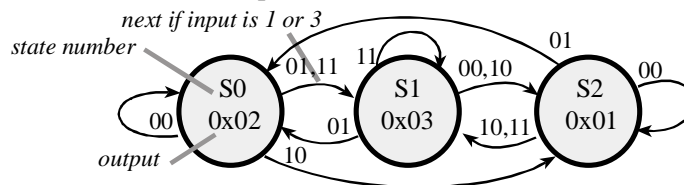
(30) Question 1. All code must be friendly. The inputs are PJ7 and PJ3, and the output is PJ0. The other five PORTJ pins will be used for another unrelated module.



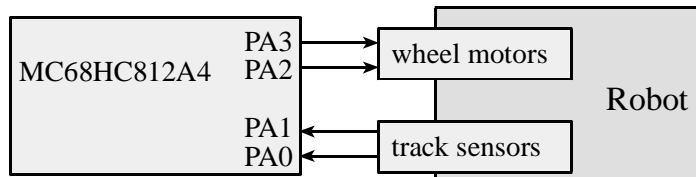
PLACE A CORRECT SINGLE LETTER FOR EACH QUESTION SHOWN ON THE ANSWER PAGE.

- | | | |
|-------------------------------------|-------------------------------------|--------------------------------------|
| A) <code>asm(" cli");</code> | R) <code>KWIEJ&= ~0x08;</code> | JJ) <code>PUPSJ&= ~0x08;</code> |
| B) <code>asm(" rti");</code> | S) <code>KWIEJ&= ~0x80;</code> | <code>PULEJ = 0x08;</code> |
| C) <code>asm(" sei");</code> | T) <code>KWIEJ = 0x08;</code> | KK) <code>PUPSJ &= ~0x80;</code> |
| D) <code>asm(" enable");</code> | U) <code>KWIEJ = 0x80;</code> | <code>PULEJ = 0x80;</code> |
| E) <code>asm(" swi");</code> | V) <code>KWIEJ = 0x00;</code> | LL) <code>PUPSJ = 0x08;</code> |
| F) <code>DDRJ &= ~0x01;</code> | W) <code>KWIEJ = 0x08;</code> | <code>PULEJ = 0x08;</code> |
| G) <code>DDRJ &= ~0x88;</code> | X) <code>KWIFJ&= ~0x80;</code> | MM) <code>PUPSJ = 0x80;</code> |
| H) <code>DDRJ = 0x01;</code> | Y) <code>KWIFJ = 0x80;</code> | <code>PULEJ = 0x80;</code> |
| I) <code>DDRJ = 0x88;</code> | Z) <code>KWIFJ = 0x00;</code> | NN) <code>PUPSJ = 0x00;</code> |
| J) <code>DDRJ = 0x00;</code> | AA) <code>KWIFJ = 0x08;</code> | <code>PULEJ = 0x08;</code> |
| K) <code>DDRJ = 0x01;</code> | BB) <code>KWIFJ = 0x80;</code> | OO) <code>PUPSJ = 0x80;</code> |
| L) <code>KPOLJ &= ~0x08;</code> | CC) <code>KWIFJ = 0x88;</code> | <code>PULEJ = 0x80;</code> |
| M) <code>KPOLJ &= ~0x80;</code> | DD) <code>PJ0 = 0x00;</code> | PP) none of the above |
| N) <code>KPOLJ = 0x08;</code> | EE) <code>PJ0 = 0x01;</code> | |
| O) <code>KPOLJ = 0x80;</code> | FF) <code>PORTJ&= ~0x01;</code> | |
| P) <code>KPOLJ = 0x00;</code> | GG) <code>PORTJ = 0x01;</code> | |
| Q) <code>KPOLJ = 0x08;</code> | HH) <code>PORTJ = 0x00;</code> | |
| | II) <code>PORTJ = 0x01;</code> | |

(25) Question 2. You will use my linked data structure to implement this line-tracking robot. Each state has a 2-bit output value, and four next state pointers.



The hardware uses 4 bits of PORTA, so your software will configure PORTA bits 3,2 to be outputs and bits 1,0 to be input. The other four bits of PORTA should not be modified.

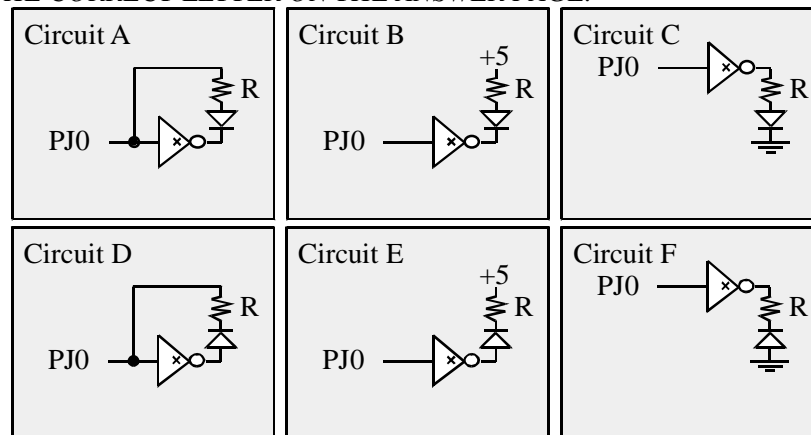


I am giving you the specification of the linked data structure, and you must write the main program (no interrupts and no time delays) that executes the Moore finite state machine. S1 is the initial state.

```
const struct State {
    unsigned char out; // 2-bit output
    const struct Node *next[4]; // Next if 2-bit input is 0-3
} typedef const struct State StateType;
typedef StateType *StatePtr;
#define S0 &fsm[0]
#define S1 &fsm[1]
#define S2 &fsm[2]
StateType fsm[3]={
    {0x02, {S0, S1, S2, S1}},
    {0x03, {S2, S0, S2, S1}},
    {0x01, {S2, S0, S1, S1}}
};
```

(15) Question 3. The objective of this question is to interface a LED to PJ0 using an open collector driver chip. The LED voltage is 2V and its current is 10 mA.

(5) Part a) Choose the appropriate circuit for the interface. You may assume PJ0 is an output pin of the 6812. *PLACE THE CORRECT LETTER ON THE ANSWER PAGE.*



(5) Part b) List all the open collector driver chips that could be used for the interface. There may be none, one, two, three or four correct answers.

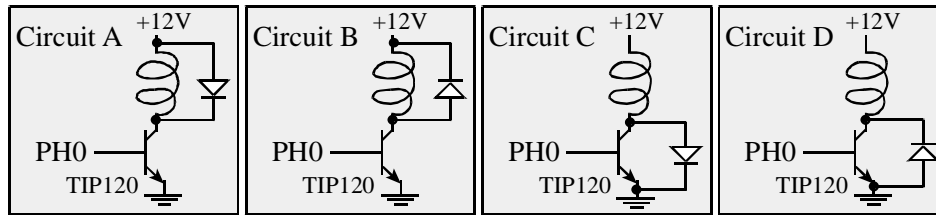
	family	example	V_{OL}	I_{OL}	I_{IH}	I_{IL}
A	standard TTL	7405	0.5 V	16 mA	40 μ A	1.6 mA
B	Schottky TTL	74S05	0.5 V	20 mA	50 μ A	2 mA
C	Low Power Schottky TTL	74LS05	0.5 V	4 mA	20 μ A	0.4 mA
D	High speed CMOS	74HC05	0.5 V	4 mA	1 μ A	1 μ A

PLACE THE CORRECT LETTER(S) ON THE ANSWER PAGE.

Choose **E** if none of the above will work.

(5) Part c) Choose the appropriate resistor value for the interface. *GIVE YOUR ANSWER IN OHMS.*

(5) **Question 4.** The objective of this question is to interface a solenoid to PH0 using a Darlington transistor. The specific issue addressed in this question is the configuration of the snubber diode, used to eliminate back EMF. Choose the appropriate circuit for the interface. You may assume PH0 is an output pin of the 6812. **PLACE THE CORRECT LETTER ON THE ANSWER PAGE.**



(25) **Question 5.** Design C code that spins two stepper motors at about 1 rotation per second. You are given a function called `step` that will rotate both motors exactly 1/30 of a rotation. The two stepper motors are connected to PORTH. Here is the C code for `step`.

```
void step(void){
    if (PORTH==0x55)    PORTH=0x66;
    else if (PORTH==0x66) PORTH=0xAA;
    else if (PORTH==0xAA) PORTH=0x99;
    else                PORTH=0x55;
}
```

Basically, you will use RTI interrupts to call the function `step` approximately every 33 ms. Comments are not necessary. Add code to make PORTH an output. On the answer sheet you will find the starter file `rti.c` that you have seen in lab. There are no bugs in this code, but you do have to make changes in this example to solve this problem For each line of this code decide to:

- 1) leave the code *unmarked*, because this code is **absolutely necessary** to solve the problem;
- 2) draw a *looped BLUE* line through the code with the modifications at the end, if you wish to **delete and replace**; or
- 3) draw a *caret, ^*, and *BLUE* line with additional code at the end, if you wish to **add code**;
- 4) (optional step) draw a *straight BLUE* line through the code, because this code is **not absolutely necessary** to solve this problem.