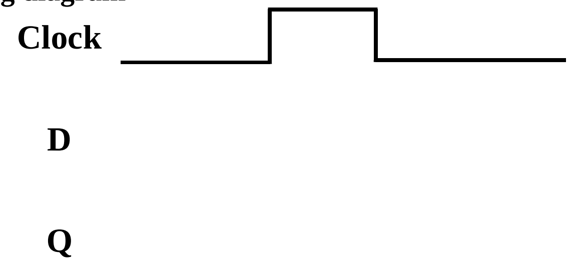
Jonathan W. Valvano
April 9, 2003, 1:00pm-1:50pm

First:__________________ Last:______________________

(5) Question 1.	Choose A-F
(5) Question 2.	Choose A-E
(5) Question 3.	Choose A-D
(5) Question 4.	Resolution=
(5) Question 5.	R=
(10) Question 6.	Maximum time=

(5) Question 7. Draw timing diagram



(5) Question 8. Show the C code

(2) Question 9.	
(2) Question 10.	
(2) Question 11.	
(2) Question 12.	
(2) Question 13.	
(2) Question 14.	
(2) Question 15.	
(2) Question 16.	
(2) Question 17.	
(2) Question 18.	

(2) Question 19a) RDRF	
(2) Question 19b) TDRE	
(2) Question 19c) KWIFH.2	
(2) Question 19d) TOF	
(2) Question 19e) RTI	

(20) Question 20. Show the ritual and interrupt service routine.

Jonathan W. Valvano May 8, 2003, 2-5pm

This is a closed book exam. You must put your answers on the special answer pages only. You have 3 hours, so please allocate your time accordingly. ***Please read the entire quiz before starting.***

(5) Question 1. When considering the issue of critical sections, how does one categorize I/O ports?

A) Local variables
B) Global variables
C) Public variables
D) Private variables
E) Volatile variables
F) Nonvolatile variables

(5) Question 2. Three events must occur for a key wakeup interrupt on PJ0 to be generated:
1) Software sets the arm bit (KWIEJ bit 0)
2) Software enables interrupts (I=0)
3) Hardware sets the flag bit (KWIFJ bit 0)
Which sequence of events cause the interrupt to be generated

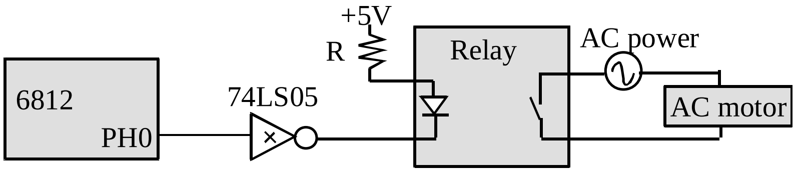
A) Only the order 1,2,3
B) Only the order 3,2,1
C) Only the order 2,1,3
D) Either the order 1,2,3 or 2,1,3
E) Any order as long as all three occur

(5) Question 3. What happens if an interrupt service routine does not acknowledge or disarm?

A) Software crashes because no more interrupts will be requested
B) The next interrupt is lost
C) This interrupt is lost
D) Software crashes because the interrupts is requested over and over.

(5) Question 4. A 12-bit ADC has an input range of –10V to +10V. The sampling rate is 1000 Hz. What is the ADC resolution?

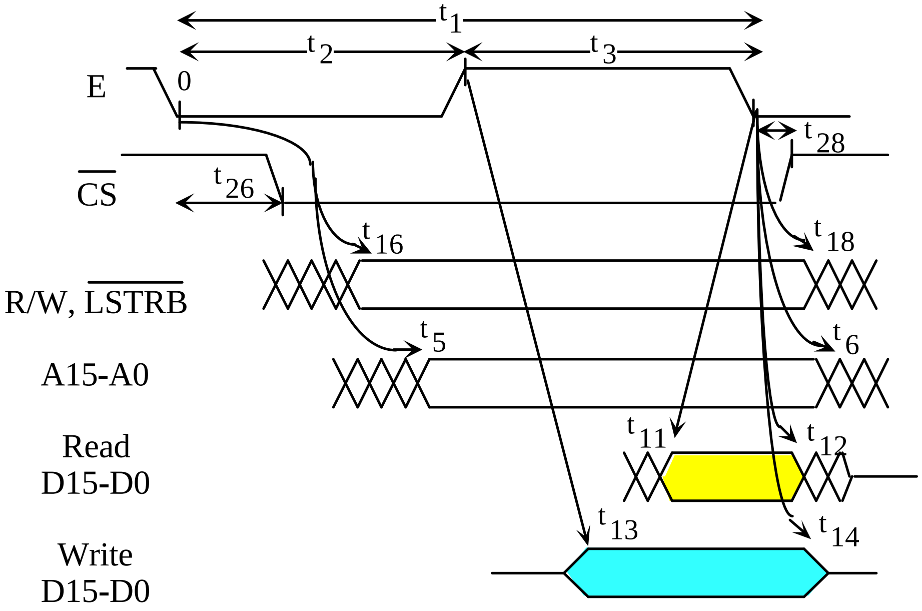
(5) Question 5. A solid-state relay can be used to switch 120 VAC power to a load. For example, the software can turn on/off an AC motor. To activate the relay (apply power to the motor), you must deliver about 2.6 V at 2 mA to the control diode. To deactivate the relay, the diode current should be zero. Choose the proper resistor value.



V_{IL}	V_{IH}	V_{OL}	I_{IL}	I_{IH}	I_{OL}
0.8 V	2.0 V	0.4 V	0.4 mA	20 µA	4 mA

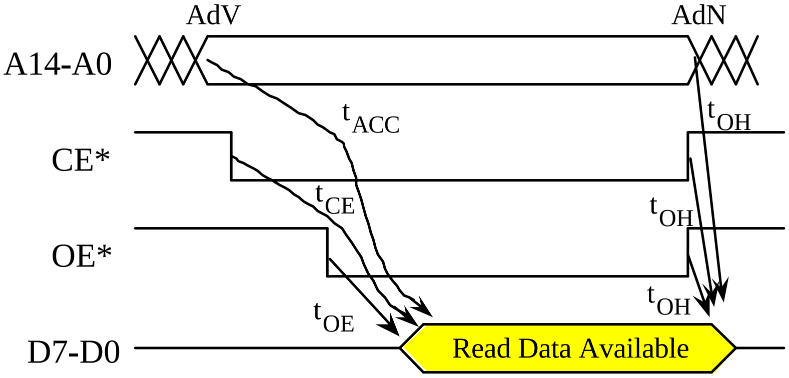
Specifications for 74LS05

(10) Question 6. Assume an expanded mode 6812 is initialized to have 2 stretches.



Num	Characteristic	0 stretch	1 stretch	2stretch	3 stretch	Units
t_1	Cycle Time	125	250	375	500	ns
t_2	Pulse Width E low	60 min	60 min	60 min	60 min	ns
t_3	Pulse Width E high	60 min	185 min	310min	435 min	ns
t_5	A15-A0, R/W delay	60 max	60 max	60 max	60 max	ns
t_6	address hold time	20 min	20 min	20 min	20 min	ns
t_{11}	Read data setup time	30 min	30 min	30 min	30 min	ns
t_{12}	Read data hold time	0 min	0 min	0 min	0 min	ns
t_{13}	Write data delay time	46 max	46 max	46 max	46 max	ns
t_{14}	Write data hold time	20 min	20 min	20 min	20 min	ns
t_{16}	R/W delay time	49 max	49 max	49 max	49 max	ns
t_{18}	R/W hold time	20 min	20 min	20 min	20 min	ns
t_{26}	CS delay time	60 max	60 max	60 max	60 max	ns
t_{28}	CS hold time	10 max	10 max	10 max	10 max	ns

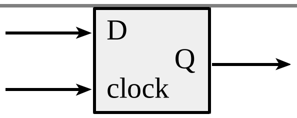
A ROM is interfaced to the 6812 with **CSP0** connected to **CE*** and **OE*** is grounded



What is the maximum possible value for the access time (t_{ACC} and t_{CE}) that can work with 2 stretches?

(5) Question 7. The objective of this question is to draw a timing diagram for this D flip-flop. On the rising edge of the **Clock**, and input data on **D** is copied to the **Q** output. The timing parameters are:

Term	description	min	max
t_{PHL}, t_{PLH}	Maximum Propagation Delay Clock to Q		32 ns
t_S	Setup Time	20 ns	
t_H	Hold Time	5 ns	



Draw a timing diagram showing the **D** and **Clock** inputs together with the **Q** output. Your figure should illustrate the required timing that allows the **D** input to be properly saved into the **Q** output.

(5) Question 8. Write the C code to implement the following equation using fixed-point math. X and Y are 8-bit signed integers. **Y = 0.123*X + 4.56**

For questions 9-18, choose the term that best fits the definition.

(2) Question 9. A high-speed communication protocol where both the clock and data are passed from transmitter to receiver.

(2) Question 10. A communication system that can transfer data in two directions, but only one direction at a time.

(2) Question 11. The difference between the true value and the measured value.

(2) Question 12. A debugging technique that uses paper and pencil to determine in advance specific input/output behaviors of our software, then runs the system and comparing actual results with expected values.

(2) Question 13. The number of information bits transferred per second.

(2) Question 14. The amount of time from when new input data is ready until the time the computer to reads the data.

(2) Question 15. The interrupt mechanism, like RDRF and TDRE, where multiple potential interrupt requests share the same interrupt vector, but have separate interrupt flags, separate interrupt arm bits, and separate acknowledge sequences.

(2) Question 16. A variable or function that can only be accessed by functions within the same module (e.g., functions within the same file).

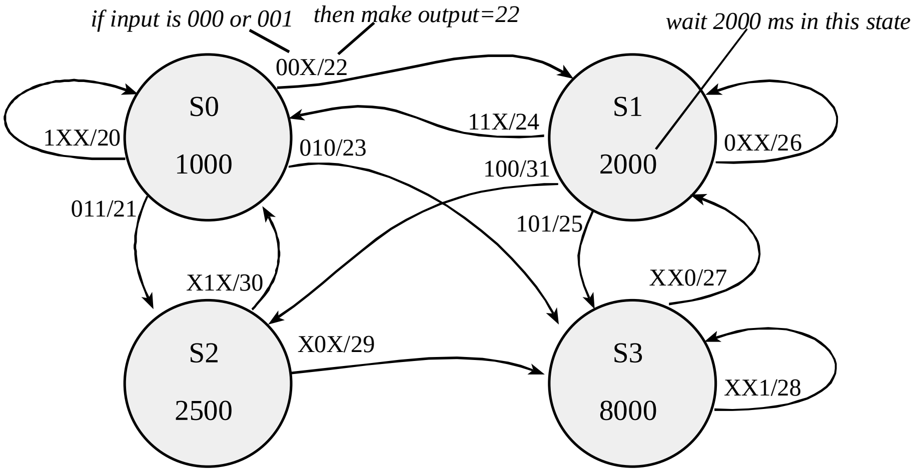
(2) Question 17. A debugging term that means the act of debugging itself has a small but not too noticeable effect on the system being tested.

(2) Question 18. A multithreaded system where the direct operations of input and output occur in background interrupt service routines, the foreground thread (main program) processes inputs and generates new outputs, and FIFO queues are used to pass data between the foreground and background.

A) accuracy
B) asynchronous serial
C) atomic
D) bandwidth
E) baud rate
F) bit time
G) breakpoint
H) buffered I/O
I) critical section
J) desk check
K) even parity
L) frame
M) friendly
N) full duplex
O) half-duplex
P) latency
Q) minimally intrusive
R) nonintrusive
S) nonvolatile
T) periodic polling
U) polled interrupt
V) precision
W) private
X) profile
Y) promotion
Z) public
AA) range
BB) real-time
CC) reentrant
DD) resolution
EE) scanpoint
FF) simplex
GG) stabilize
HH) synchronous serial
II) vectored interrupt
JJ) vulnerable window

(10) Question 19. Write 1 2 or 3 lines of C code that acknowledges each type of interrupt.
Part a) How do you acknowledge an SCI RDRF interrupt? (i.e., clear RDRF)
Part b) How do you acknowledge an SCI TDRE interrupt? (i.e., clear TDRE)
Part c) How do you acknowledge a key wakeup PH2 interrupt? (i.e., clear KWIFH.2)
Part d) How do you acknowledge a TOF interrupt? (i.e., clear TOF)
Part e) How do you acknowledge an RTI interrupt? (i.e., clear RTIF)

(20) Question 20. You will use my linked data structure and periodic output compare interrupts to implement this Mealy finite state machine. Each state has a time to wait in milliseconds, eight 5-bit output values, and eight next state pointers. In a Mealy machine both the outputs and the next states depend on the input. For each state your software will 1) wait the specified amount of time; 2) read the 3-bit input; 4) perform the 5-bit output that depends on the current state and the input; and 5) jumps to the next state. Which state to go to next also depends on the current state and the input.



The hardware uses all 8 bits of PORTH, in particular your software will configure PORTH bits 4,3,2,1,0 to be outputs and bits 7,6,5 to be input. I am giving you the specification of the linked data structure, and you must write the output compare initialization and the output compare interrupt service routine. You may assume the main program executes your ritual, then performs other unrelated functions. Other than initialization, YOU MAY NOT PERFORM ANY FUNCTIONS IN THE FOREGROUND. S0 is the initial state.

```
const struct State {
  unsigned short wait;           // time in ms to wait
  unsigned char out[8];          // 5-bit outputs
  const struct Node *next[8];};  // Next if 3-bit input is 0-7
typedef const struct State StateType;
typedef StateType *StatePtr;
#define S0 &fsm[0]
#define S1 &fsm[1]
#define S2 &fsm[2]
#define S3 &fsm[3]
StateType fsm[4]={
  {1000,{22,22,23,21,20,20,20,20},{S1,S1,S3,S2,S0,S0,S0,S0}},
  {2000,{26,26,26,26,31,25,23,23},{S1,S1,S1,S1,S2,S3,S0,S0}},
  {2500,{29,29,30,30,29,29,30,30},{S3,S3,S0,S0,S3,S3,S0,S0}},
  {8000,{27,28,27,28,27,28,27,28},{S1,S3,S1,S3,S1,S3,S1,S3}},
};
```