ELEX 2117 : Digital Techniques 2
2022 Fall Term

MIDTERM EXAM 2

15:30 – 17:20
Friday, November 4, 2022
SW03-1710

This exam has five (5) questions on two (2) pages. The marks for each question are as indicated. There are a total of seventeen (17) marks. Answer all questions. Write your answers and all rough work in this paper and nowhere else. Show your work. Draw a box around your final answer. Numerical answers must include units. Books and notes are allowed. No electronic devices other than calculators are allowed. **Show your work.**

This exam paper is for:

Sample Exam 1 A00123456

Each exam is equally difficult.

Answer your own exam.

Do not start until you are told to do so.

Name: ______________________

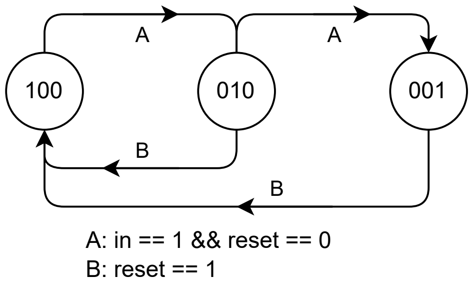
BCIT ID: ______________________

Signature: ______________________

0

Question 1 — 5 marks

A sequence detector detects input sequences consisting of a 1, followed by any number of zeros (including none), followed by a 1.
There are two inputs named **reset** and **in**. There is one output named **out**. When **reset** is asserted the output is set to 0. When the desired sequence is detected, the output is set to 1. The output remains at 1 until **reset** is asserted again.
The state transition diagram and state transition table are shown at right. The state does not change for other input conditions. A second table shows the value of **out** for each state.
Fill in the missing code in the following Verilog module so as to implement the sequence detector state machine described above.



state	reset	in	next state
X	1	X	100
100	0	1	010
010	0	1	001

state	out
100	0
010	0
001	1

```
module detector
  ( input logic reset, clk, in,
    output logic out ) ;


   logic [2:0] state ;
endmodule
```

Question 2 — 2 marks

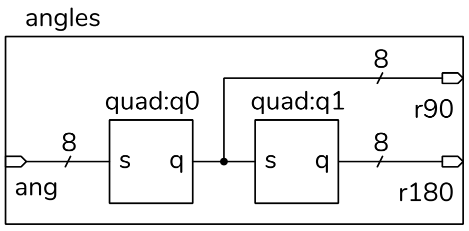
(a) Should an input named **RESET** be set high or low to cause a reset?

(b) Should an output named $\overline{\texttt{INTERRUPT}}$ be set high or low to cause an interrupt?

Question 3 — 4 marks

The following Verilog shows the declaration of a module named **quad**. The diagram shows how two of these are connected within a module named **angles**:

```
module quad
  ( input  logic [7:0] s,
    output logic [7:0] q ) ;
   // ...
endmodule
```



The numbers above the lines show the signal (bus) widths. The identifiers above the boxes show the module and instance names. The identifiers below the ports show the **angles** module port names.

Write a System Verilog module named **angles** that implements the diagram above. Declare any signals required to implement the **angles** module. Do not write the **quad** module. Follow the course coding conventions

Question 4 — 4 marks

Fill in the following testbench module with code that does the following:

(a) sets the value of **n** to **0** in an **initial** block,

(b) terminates the simulation with **$stop()** if the value of **n** is equal to **8'd255**,

(c) adds 1 to **n** every 5 µs, and

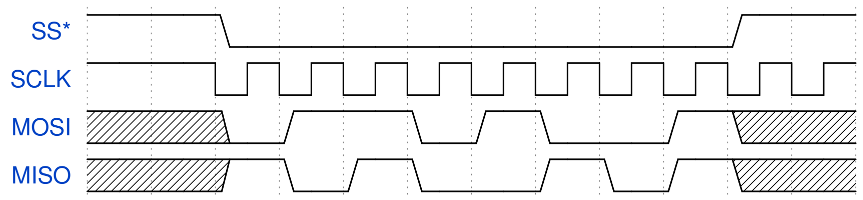
(d) uses **$display()** to print the value of **n** every 20 µs.

```
module midterm_tb ;


   logic [7:0] n ;
endmodule
```

Question 5 — 2 marks

The following waveform shows the signals on an SPI interface. What value was transmitted from the slave to the master assuming the bits were transferred most-significant-bit first? Give you answer as a hexadecimal number. Show your work.



ELEX 2117 : Digital Techniques 2

2022 Fall Term

MIDTERM EXAM 2

15:30 – 17:20

Friday, November 4, 2022

SW03-1710

This exam has five (5) questions on two (2) pages. The marks for each question are as indicated. There are a total of seventeen (17) marks. Answer all questions. Write your answers and all rough work in this paper and nowhere else. Show your work. Draw a box around your final answer. Numerical answers must include units. Books and notes are allowed. No electronic devices other than calculators are allowed. **Show your work.**

This exam paper is for:

Sample Exam 2 A01234567

Each exam is equally difficult.

Answer your own exam.

Do not start until you are told to do so.

Name: ____________________

BCIT ID: ____________________

Signature: ____________________

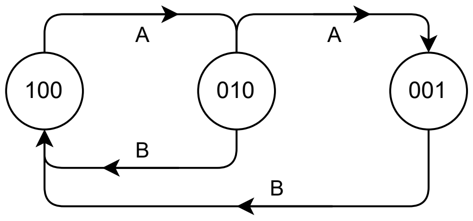
Question 1 5 marks

A sequence detector detects input sequences consisting of a 1, followed by any number of zeros (including none), followed by a 1.

There are two inputs named **reset** and **in**. There is one output named **out**. When **reset** is asserted the output is set to 0. When the desired sequence is detected, the output is set to 1. The output remains at 1 until **reset** is asserted again.

The state transition diagram and state transition table are shown at right. The state does not change for other input conditions. A second table shows the value of **out** for each state.

Fill in the missing code in the following Verilog module so as to implement the sequence detector state machine described above.



A: in == 1 && reset == 0
B: reset == 1

state	reset	in	next state
X	1	X	100
100	0	1	010
010	0	1	001

state	out
100	0
010	0
001	1

```
module detector
  ( input logic reset, clk, in,
    output logic out ) ;


   logic [2:0] state ;
endmodule
```

Question 2 2 marks

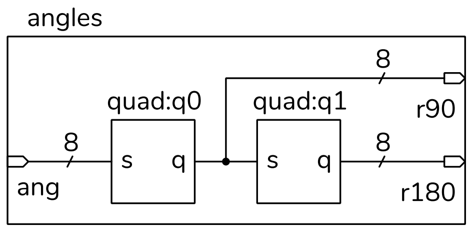
(a) Should an input named $\overline{\texttt{RESET}}$ be set high or low to cause a reset?

(b) Should an output named **INTERRUPT** be set high or low to cause an interrupt?

Question 3 4 marks

The following Verilog shows the declaration of a module named **quad**. The diagram shows how two of these are connected within a module named **angles**:

```
module quad
  ( input  logic [7:0] s,
    output logic [7:0] q ) ;
   // ...
endmodule
```



The numbers above the lines show the signal (bus) widths. The identifiers above the boxes show the module and instance names. The identifiers below the ports show the **angles** module port names.

Write a System Verilog module named **angles** that implements the diagram above. Declare any signals required to implement the **angles** module. Do not write the **quad** module. Follow the course coding conventions

Question 4 4 marks

Fill in the following testbench module with code that does the following:

(a) sets the value of **n** to **0** in an **initial** block,

(b) terminates the simulation with **$stop()** if the value of **n** is equal to **8'd255**,

(c) adds 1 to **n** every 5 µs, and

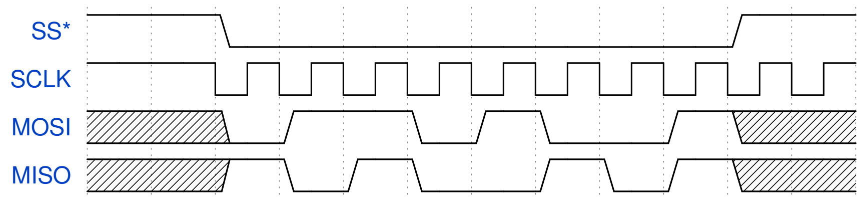
(d) uses **$display()** to print the value of **n** every 20 µs.

```
module midterm_tb ;

   logic [7:0] n ;
endmodule
```

Question 5 2 marks

The following waveform shows the signals on an SPI interface. What value was transmitted from the master to the slave assuming the bits were transferred most-significant-bit first? Give you answer as a hexadecimal number. Show your work.


SS*
SCLK
MOSI
MISO