

# ECE2031 In-Class Exam C Spring 2004

## ANSWER SHEET

Name \_\_\_\_\_ Section \_\_\_\_\_

Student No. \_\_\_\_\_

**Closed Books, Closed Notes, No computers or calculators.**

Having read the Georgia Institute of Technology Academic Honor Code, I understand and accept my responsibility as a member of the Georgia Tech Community to uphold the Honor Code at all times. In addition, I understand my options for reporting honor violations as detailed in the code.

\_\_\_\_\_  
(Signature)

\_\_\_\_\_  
(Date)

1. (15 points)

X Y

|    | 00 | 01 | 11 | 10 |
|----|----|----|----|----|
| 00 |    |    |    |    |
| 01 |    |    |    |    |
| 11 |    |    |    |    |
| 10 |    |    |    |    |

$Q_1Q_0$

Z= \_\_\_\_\_

**CIRCLE YOUR SELECTED ANSWERS**

- |              |             |                 |             |
|--------------|-------------|-----------------|-------------|
| 2. a b c     | (5 pts)     | 7. a b c d      | (10 points) |
| 3. a b c d e | (5 points)  | 8. a b          | (10 points) |
| 4. a b       | (10 points) | 9. a b c        | (5 points)  |
| 5. a b c d e | (10 points) | 10. a b c d e f | (10 points) |
| 6. a b c d e | (10 points) | 11. a b c d e   | (10 points) |

Name \_\_\_\_\_

## ECE2031 In-Class Exam C Spring 2004

**Closed Books, Closed Notes, No computers or calculators.  
Mark all answers on the answer sheet.**

1. (15 pts) The truth table below shows the output Z for some state machine. (The transition tables for next states Q1+ and Q0+ are not shown and do not matter here.) Fill in the Karnaugh map on the answer sheet for Z, circle the prime implicants, and write the minimal sum of products expression for Z.

| Q1 | Q0 | X | Y | Z |
|----|----|---|---|---|
| 0  | 0  | 0 | 0 | 0 |
| 0  | 0  | 0 | 1 | 0 |
| 0  | 0  | 1 | 0 | 1 |
| 0  | 0  | 1 | 1 | 1 |
| 0  | 1  | 0 | 0 | 0 |
| 0  | 1  | 0 | 1 | 0 |
| 0  | 1  | 1 | 0 | 0 |
| 0  | 1  | 1 | 1 | 1 |
| 1  | 0  | 0 | 0 | 0 |
| 1  | 0  | 0 | 1 | 0 |
| 1  | 0  | 1 | 0 | 0 |
| 1  | 0  | 1 | 1 | 1 |
| 1  | 1  | 0 | 0 | 1 |
| 1  | 1  | 0 | 1 | 1 |
| 1  | 1  | 1 | 0 | 0 |
| 1  | 1  | 1 | 1 | 0 |

2. (5 pts) Is the state machine described in problem 1 a Moore state machine? You may assume that Q1 and Q0 are the only state variables, X and Y are the only inputs, and Z is the only output.
- a) YES
  - b) NO
  - c) Impossible to tell without more information
3. (5 pts) In the implementation of a state machine with D flip-flops and discrete gates (i.e., TTL chips), what signals are connected to the D inputs of the flip flops? (Select one.)
- a) The state machine inputs
  - b) The state machine outputs
  - c) The current-state signals
  - d) The next-state signals
  - e) None of the above
4. (10 pts) In the train lab, it was possible for trains A and B to run in opposite directions on the same segment of track.
- a) TRUE
  - b) FALSE

Name \_\_\_\_\_

5. (10 points) The VHDL code for the switch debouncer is shown below

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.all;
USE IEEE.STD_LOGIC_ARITH.all;
USE IEEE.STD_LOGIC_UNSIGNED.all;

ENTITY debounce IS
    PORT( pb, clock_100Hz          : IN      STD_LOGIC;
          pb_debounced           : OUT     STD_LOGIC);
END debounce;

ARCHITECTURE a OF debounce IS
    SIGNAL SHIFT_PB                : STD_LOGIC_VECTOR(3 DOWNTO 0);
BEGIN

    -- Debounce clock should be approximately 10ms or 100Hz
    PROCESS
    BEGIN
        WAIT UNTIL (clock_100Hz'EVENT) AND (clock_100Hz = '1');
        SHIFT_PB(2 DOWNTO 0) <= SHIFT_PB(3 DOWNTO 1);
        SHIFT_PB(3) <= NOT PB;
        IF SHIFT_PB(3 DOWNTO 0)="0000" THEN
            PB_DEBOUNCED <= '0';
        ELSE
            PB_DEBOUNCED <= '1';
        END IF;
    END PROCESS;
END a;
```

**Suppose you changed the last seven lines to**

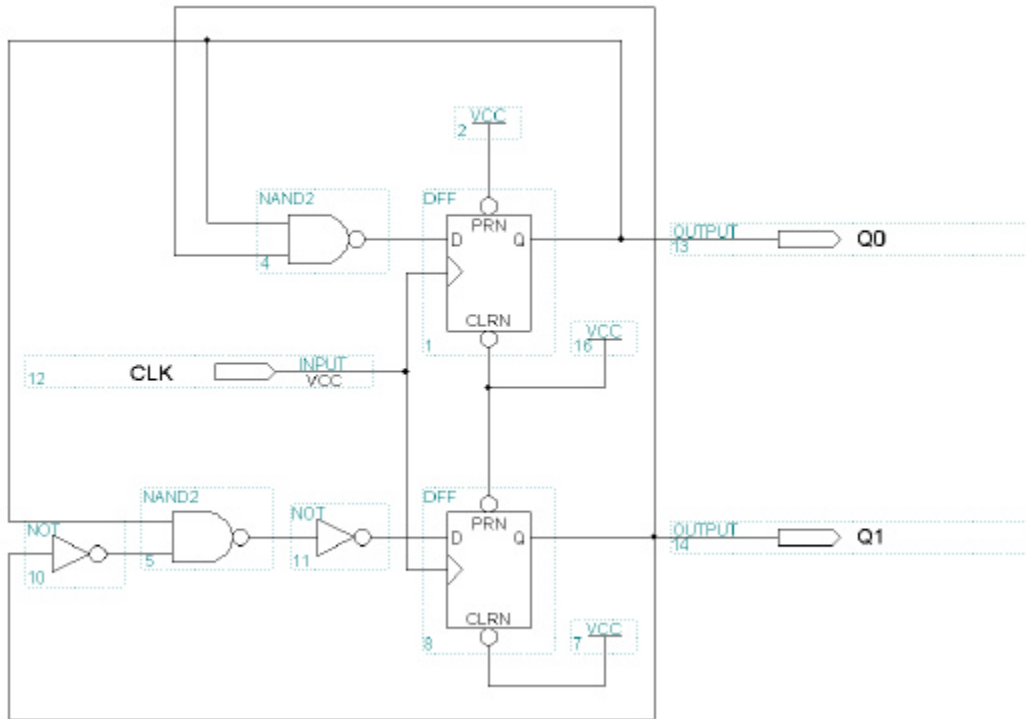
```
IF SHIFT_PB(3 DOWNTO 0)="1111" THEN
    PB_DEBOUNCED <= '1';
ELSE
    PB_DEBOUNCED <= '0';
END IF;
END PROCESS;
END a;
```

What would be the effect on the operation of the switch debouncer device? (Select all that apply.)

- It originally inverted the input signal as it debounced it, but now it will not (regardless of whether it actually debounces).
- It originally did NOT invert the input signal as it debounced it, but now it WILL invert it (regardless of whether it actually debounces).
- It will no longer have a debouncing effect.
- It will not debounce in exactly the same way, but it will still have a debouncing effect.
- Given an input that does NOT bounce, but simply changes from high to low, the output will be identical to the output of the original device.

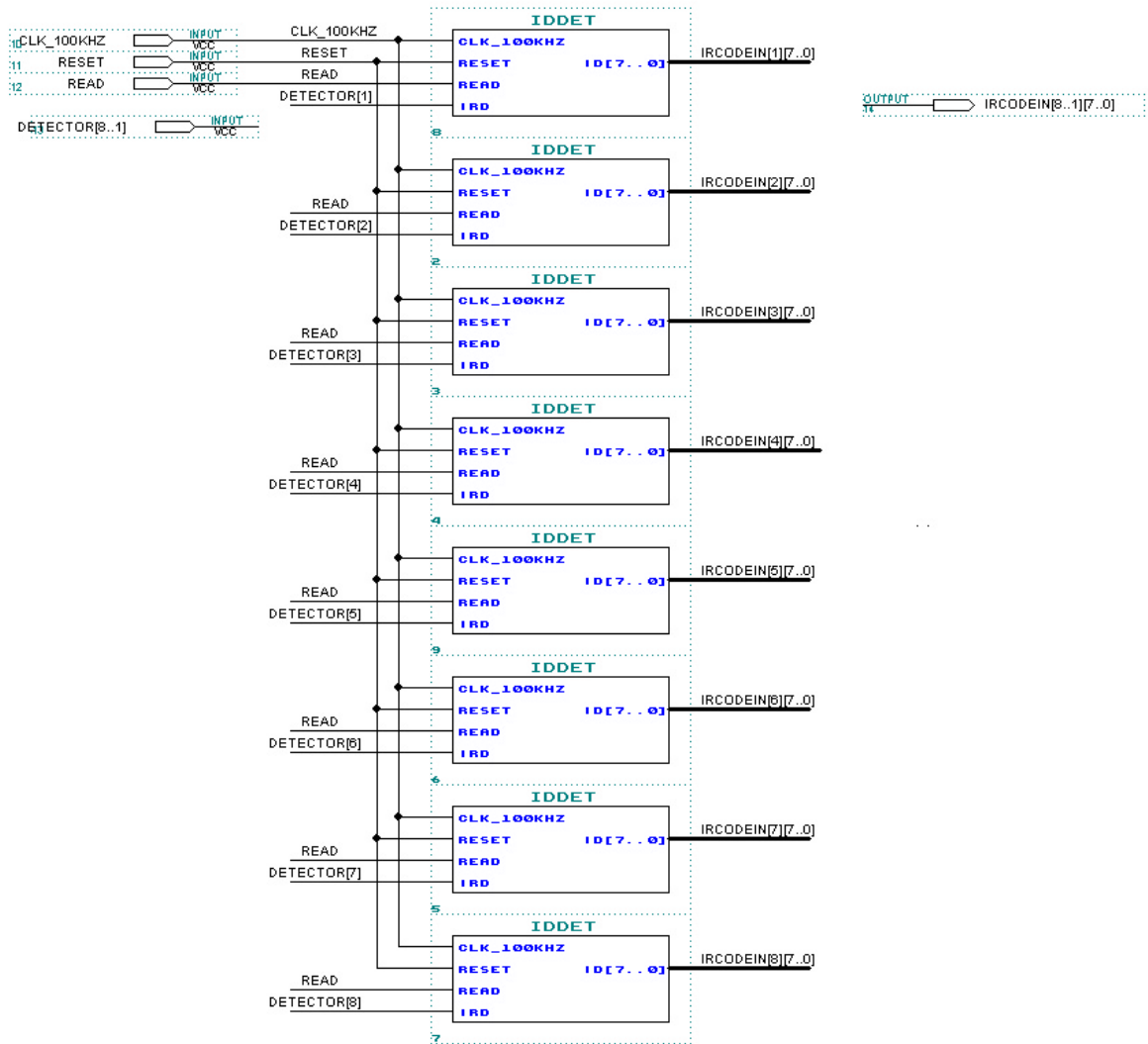
Name \_\_\_\_\_

6. (10 points) For the circuit shown below, assume that all NAND gates have a propagation delay of 12 ns, and all inverters have a propagation delay of 5 ns. The D flip-flops have a propagation delay of 10 ns, a setup time of 3 ns, and a hold time of 2 ns. What is the maximum clock frequency of the circuit?



- a)  $\frac{1}{32} \times 10^{-9}$  Hz  
 b)  $\frac{1}{37} \times 10^{-9}$  Hz  
 c)  $\frac{1}{35} \times 10^{-9}$  Hz  
 d)  $\frac{1}{45} \times 10^{-9}$  Hz  
 e) none of the above
7. (10 pts) Suppose you just implemented an eight-bit counter in a device like the FLEX chip on the UP2 board, and you have access to all eight bits (i.e., you can probe them). If you wanted to see if it was counting properly, it would be most effective to use which of the following tools? (Select one.)
- a) A logic probe  
 b) An oscilloscope similar to the one in the lab  
 c) A logic analyzer similar to the one in the lab  
 d) A voltmeter
8. (5 points) Under the stated rules of the project demo, the puck may be moving faster than the highest speed of the robot
- a) TRUE  
 b) FALSE

Name \_\_\_\_\_



The module IDDET shown in the figure above is used inside of the IR\_COMM device in your project files and will be used for the next two questions. You don't need to know any more about IDDET than the following:

- The CLK\_100KHZ signal has to be connected to a 100 kHz clock
- The RESET signal, when held high, forces the 8-bit ID output to be “00000000”
- The IRD signal is connected directly to a detector exactly like one of the eight detectors on the AlteraBot daughterboard
- If RESET is not high, and if READ is not high, the 8-bit ID output will change to whatever IR code is received by the IR detector, as soon as the last bit of that code is received (there are extra error detection bits as part of that IRD signal, but don't concern yourself with that)
- If READ goes high, the 8-bit ID output WILL NOT change as long as it is high, but it will change to “00000000” as soon as READ goes low again

9. (5 points) The IDDET modules are not running concurrently as shown – they must take turns processing an IR detector signal in order to produce valid outputs

- a) TRUE
- b) FALSE
- c) Impossible to determine from the information given

Name \_\_\_\_\_

10. (10 points) Also referring to the previous figure, if you could fit something like the circuit shown in that figure inside your `top_scomp.gdf` file used for the project, you would be able to read 8 detected IR codes simultaneously. Aside from getting it to fit, what else would have to be done for your SCOMP to read the detected codes? (Select all that apply.)
- a) The eight DETECTOR signals would have to be connected to the actual pins used for the eight IR detectors
  - b) The signal bus `IRCODEIN[8..1][7..0]` would have to be directly connected to SCOMP's I/O data bus
  - c) The READ signal would have to be directly connected to the inverted `IO_WRITE` signal from SCOMP
  - d) A way would have to be found to drive SCOMP's I/O data bus with one, or at most two, of the detected IR codes at a given time, and only when those codes are requested by SCOMP (i.e., when SCOMP reads an I/O address reserved for that function)
  - e) It would be necessary to generate eight different clock signals, because clocks can only be used once to drive inputs within the FLEX chip
  - f) RESET must be tied to VCC
11. (10 points) The ASM code on the next page is a complete program extracted from the example given for the first week of experiments with the robot. Only comments, extraneous constants, etc. have been removed, so it will compile and do exactly what it did in the lab. Which of the statements below are true statements about what the code does? (Select all that apply.)
- a) As long as the result of the `IN Status` instruction (in the AC register) is `&H7032`, the code will stay in the loop starting at `CheckPower`
  - b) The loop beginning at `Start` (and ending at `JZERO Start`) will be exited if the instruction `IN Switches` produces the value `&H01FF` (in the AC register).
  - c) If the program runs up to its final instruction, the seven-segment display will show "03".
  - d) IR code values are being read by SCOMP.
  - e) The loop starting at `Goal1` is trying to stop the robot when its X position becomes greater than or equal to 984 (decimal).

Name \_\_\_\_\_

```
Start:      ORG      &H00
            LOAD     Zero
            ADDI     &H01
Temp:       OUT      SevenSeg
            IN       Switches
            AND      UserPBmask
CheckPower: JZERO    Start
            IN       Status
            SHIFT    -8
            OUT      SevenSeg
            SUB      MinBattLevel
            JPOS     Go
            JUMP     CheckPower
Go:         LOAD     One
            OUT      Enable
            LOAD     Safespeed
            OUT      Velocity
            LOAD     Three
            OUT      SevenSeg
Goall:     IN       Xposition
            SUB      HalfMeter
            JNEG     Goall
            LOAD     Zero
            OUT      Enable
            LOAD     Four
            OUT      SevenSeg
AllDone:   JUMP     AllDone

Zero:      DW       &H0000
One:       DW       &H0001
Two:       DW       &H0002
Three:     DW       &H0003
Four:      DW       &H0004
UserPBmask: DW      &H0100
MinBattLevel: DW    &H0070
SafeSpeed: DW      100
HalfMeter: DW      984

Enable:    EQU      &H80
Velocity:  EQU      &H81
Status:    EQU      &H80
Xposition: EQU      &H81
Yposition: EQU      &H82

Switches:  EQU      &H00
SevenSeg:  EQU      &H01
```