Name: _____

First Name: _____

Student ID: _____

Exercises

Design of Digital Circuits SS2017

Prof. Dr. Onur Mutlu, Prof Dr. Srdjan Capkun

Examination Rules:

1. Written exam, 90 minutes in total.
2. No books, no calculators, no computers or communication devices. Six pages of hand-written notes are allowed.
3. Write all your answers on this document, space is reserved for your answers after each question. Blank pages are available at the end of the exam.
4. Put your Student ID card visible on the desk during the exam.
5. If you feel disturbed, immediately call an assistant.
6. Answers will only be evaluated if they are readable.
7. Write with a black or blue pen (no pencil, no green or red color).
8. Show all your work. For some questions, you may get partial credit even if the end result is wrong due to a calculation mistake.
9. Points for every part are indicated in the exam. They should correspond to the expected difficulty of the questions. Consider this when allocating your time.

Question:	1	2	3	4	5	6	7	8	Total
Points:	5	5	18	9	8	12	10	5	72
Score:									

This page intentionally left blank

1. (a) (2 points) For the following four numbers given in hexadecimal notation, write the corresponding binary number.

   $0x42$: _____

   $0xF9$: _____

   $0x85$: _____

   $0x7E$: _____

   (b) (3 points) Sort the four numbers given above from the *smallest to largest* assuming that the binary numbers are coded using:

   Unsigned: _____

   Two's complement: _____

   Sign Magnitude: _____

2. (a) (4 points) For the following four numbers given in decimal or hexadecimal notation, write the corresponding binary number using the indicated format.

   $(-11)_{10}$ using 6-bit two's complement: _____

   $(51)_{10}$ using 6-bit unsigned: _____

   $(-17)_{10}$ using 6-bit sign magnitude: _____

   $(39)_{16}$ using 6-bit unsigned: _____

   (b) (1 point) What are the problems with the sign/magnitude representation of binary numbers, why are they not used more often than two's complement representation?

3. For this question, use the following truth table for a 4-input logic function called Z.

Input				**Output**
A	B	C	D	Z
0	0	0	0	0
0	0	0	1	0
0	0	1	0	X
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	X
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	X
1	1	1	1	1

(a) (1 point) What is the meaning of X in this truth table?

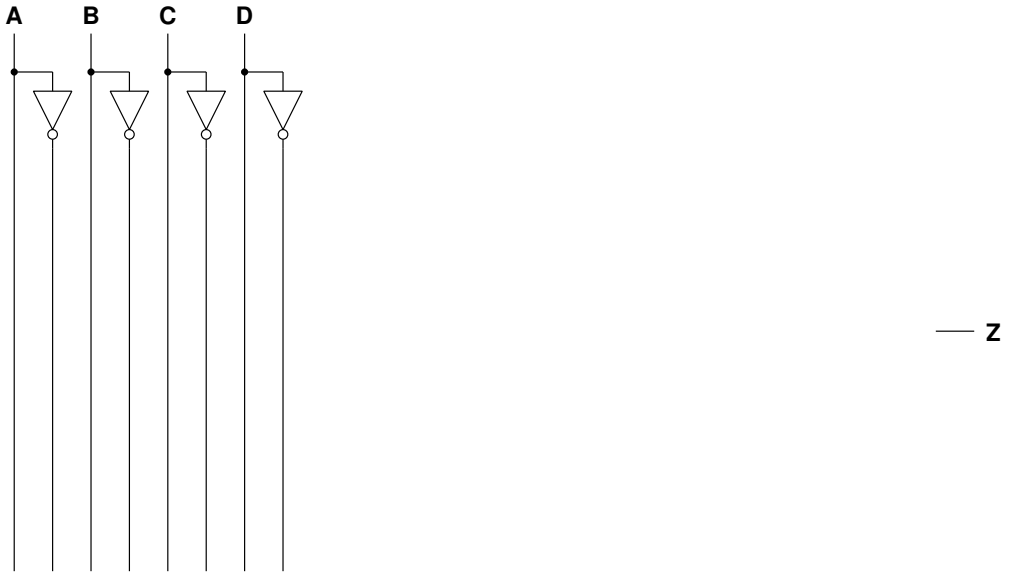
(b) (6 points) A friend of yours has determined the following Boolean equation for Z:

$$Z = (\overline{B} + D) \cdot (A + B + C) \cdot (A + B + \overline{C}) \cdot (\overline{A} + B + \overline{C}) \cdot (\overline{A} + \overline{C} + D)$$

But he is not sure if this is correct. Verify whether or not the given equation matches the truth table given above. If not, please explain how the equation can be corrected.

(c) (5 points) Derive your own *optimized* Boolean equation corresponding to the same truth table using *sums-of-products* form. Try to take advantage of the 'X' values to minimize the equation as much as possible. (*Hint: use a Karnaugh map*)

(d) (4 points) Draw a gate-level schematic that realizes the function Z using **only** 2-input AND, OR gates. Assume that you have all variables (A, B, C, D) available as input. Their complements $(\overline{A}, \overline{B}, \overline{C}, \overline{D})$ are already drawn for you.



(e) (2 points) Assume that all the gates (AND, OR, NOT) in the previous diagram have a propagation delay of 100 ps and a contamination delay of 50 ps. What is the delay of the longest (*critical*) path and the *shortest* path of this circuit?

This page intentionally left blank

4. In this question, you will be asked to derive the Boolean Equations for two 4-input logic functions. Please use the Truth Table below for this part.

Input				**Output**	
D_3	D_2	D_1	D_0	A	B
0	0	0	0		
0	0	0	1		
0	0	1	0		
0	0	1	1		
0	1	0	0		
0	1	0	1		
0	1	1	0		
0	1	1	1		
1	0	0	0		
1	0	0	1		
1	0	1	0		
1	0	1	1		
1	1	0	0		
1	1	0	1		
1	1	1	0		
1	1	1	1		

(a) (4 points) The output A is *one* when there are at least three consecutive 1's or 0's in the word D_3, D_2, D_1, D_0 (for example 1110), output is *zero* otherwise. Write the *simplified* Boolean equation for A using the *Sum of Products* form. (*Hint: use a Karnaugh map*)

(b) (4 points) The output B is *one* when there are *three or more zeroes* in the 4-bit input word D_3, D_2, D_1, D_0, output is *zero* otherwise (for example 0100). Write the *simplified* Boolean equation for B using the *Product of Sums* form. (*Hint: use a Karnaugh map*)

(c) (1 point) In general, how many different 4-input Boolean functions can be defined?

5. In this question you will be asked to draw design the FSM for a power saving control module of a mobile device.

   (a) (4 points) We want to design the power saving control module of a mobile device.

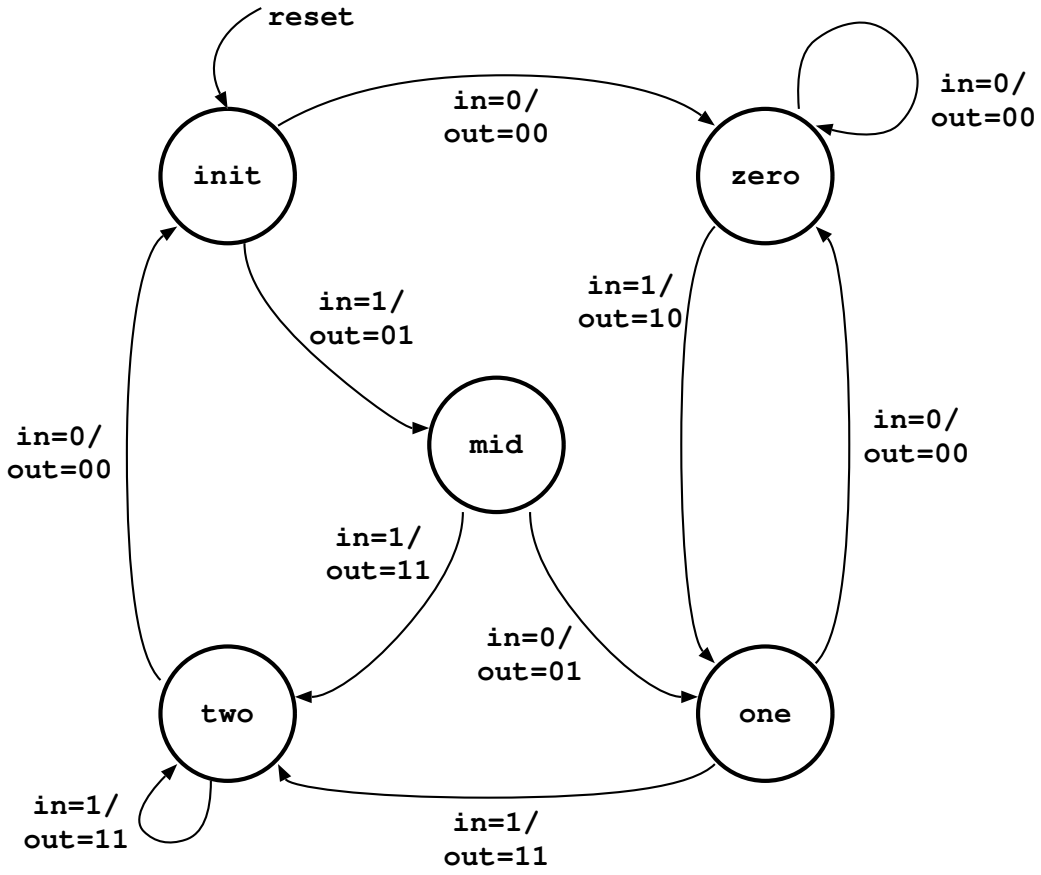
   - There are two inputs: C (charging) and D (discharging)
   - There are four power levels (0,1,2,3) for the device
   - When both inputs (C, D) are the same the power level does not change
   - When only C is active, power level increases until the last level (3) is reached
   - When only D is active, power level decreases until the lowest level (0) is reached
   - There are 3 outputs: DIM (dimmer), LOW (low power), $ANIM$ (animations)
   - DIM is active at power level 1 or lower
   - LOW is active at power level 0 only and signals that we are at low power
   - $ANIM$ is active at power level 3 only and enables power hungry animations on the device
   - the reset state corresponds to power level 2.

   Draw the State Transition Diagram for a Moore type FSM that implements this state machine

(b) (4 points) Using the State Transition Diagram, complete the following table for both State Transitions and the outputs.

Present State	**Inputs**		**Next State**	**Outputs**		
name	C	D	name	DIM	LOW	$ANIM$

6. In this question you will be given the state transition diagram for a Finite State Machine (FSM). The FSM has a two-bit output "`out[1:0]`" as well as a single bit input "`in`".



(a) (2 points) Is this a Mealy or Moore type of FSM, briefly explain why?

(b) (5 points) Using the following state encoding table, complete the state transition and output table on the following page.

State	**Encoding**		
Name	S_2	S_1	S_0
`init`	1	1	1
`mid`	1	0	0
`zero`	0	0	0
`one`	0	0	1
`two`	0	1	0

Present State			**Input**	**Next State**			**Outputs**	
P_2	P_1	P_0	in	N_2	N_1	N_0	out_1	out_0
1	1	1	0	0	0	0	0	0

(c) (5 points) Draw a new state transition diagram for this FSM using a different type (if the original was a Moore, then draw a Mealy type or vice versa) that has the same functionality.

7. (10 points) There are four Verilog code snippets in this section. For each code, first state whether or not there is a mistake. If there is a mistake explain how to correct it. *Note: Assume that the behavior as described, is correct*

(a)

```
module mux2 ( input [1:0] i,  output s, input z);
  assign s= (z) ? i[1]:i[0];
endmodule

module one (input [3:0] data, input sel1, input sel2, output z );
  wire [1:0] temp ;

  mux2 i0 (data[1:0], sel1, temp[0]);
  mux2 i1 (data[3:2], sel1, temp[1]);
  mux2 final (temp, sel2, z);
endmodule
```

(b)

```
module two (input [3:0] a, input [0:3] b, output reg [3:0] z);
 always @ ( *)
     if (a == 0)
       z={b[0],b[1],b[2],b[3]};
     else
       z=a+b;
endmodule
```

(c)

```
module three (clk, rst, a, b, c, z);
  input a,b,c,clk,rst;
  output reg z;
  reg q;

  always @ (*)
    begin
      q <= a ^ b;
      if (c) q <= ~(a^b);
    end
  always @ (negedge clk)
     if (rst) z= 1'b0;
     else     z= q;
endmodule
```

(d)

```
module four (input [2:0] sel, output reg [5:0] z);
 case (sel)
   0: z = 6'b00_0000;
   1: z = 6'b00_0001;
   2: z = 6'b00_0011;
   3: z = 6'b00_0111;
   4: z = 6'b00_1111;
   5: z = 6'b01_1111;
   6: z = 6'b11_1111;
   default: z= 6'b00_0000;
 endcase
endmodule
```

8. (a) (5 points) When considering the performance of caches, what are compulsory cache misses? Is there any way to reduce them? How?