Digital Design and Computer Architecture (E85)

K. Wang & J. Spjut

Fall 2012

Midterm

" There are 10 kinds of people in the world – those who understand binary and those who don't."

This is a closed-book closed-notes exam. You are permitted a calculator and one 8.5x11" sheet of paper with notes. You may have notes on both sides of the single 8.5x11" sheet.

You are allowed at most **1 hour and 30 minutes** to take the exam.

Along side each question, the number of points is written in brackets. The entire exam is worth 100 points. Plan your time accordingly. All work and answers should be written directly on this examination booklet. Use the backs of pages if necessary. Write neatly; illegible answers will be marked wrong. **Show your work** for partial credit.

Name: Aswer Keys

This is a closed-book closed-notes exam. You are permitted a calculator and one 8.5x11" sheet of paper with notes. You may have notes on both sides of the single 8.5x11" sheet.

You are allowed at most **1 hour and 30 minutes** to take the exam.

Along side each question, the number of points is written in brackets. The entire exam is worth 100 points. Plan your time accordingly. All work and answers should be written directly on this examination booklet. Use the backs of pages if necessary. Write neatly; illegible answers will be marked wrong. **Show your work** for partial credit.

Name: _____

Do Not Write Below This Point

Page 3: _____/12
Page 4: _____/12
Page 5: _____/24
Page 6: _____/13
Page 7: _____/13
Page 8: _____/10
Page 9: _____/8
Page 10: _____/8

Total: _____/100

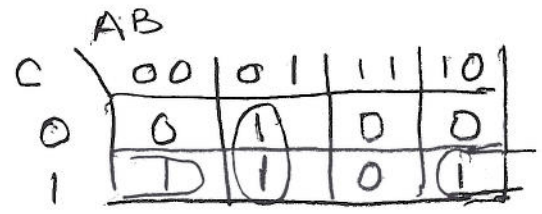
1. [24 pts] For the Boolean equation

$$Y = (A \oplus B)\,C + \overline{A}\,B\,\overline{C} + \overline{A}\,\overline{B}\,C$$

6pts (a) Show the truth table for Y as a function of inputs A, B, and C.

	A	B	C	Y
	0	0	0	0
	0	0	1	1
→	0	1	0	1
	0	1	1	1
	1	0	0	0
	1	0	1	1
	1	1	0	0
	1	1	1	0

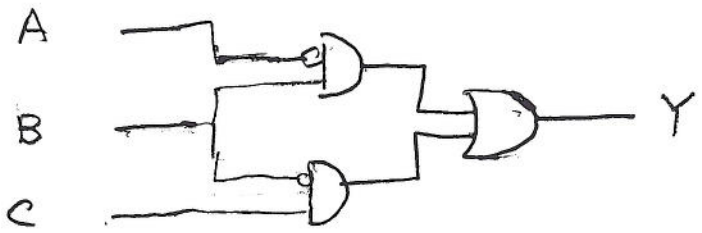
6pts (b) Simply the Boolean equation for Y as a function of inputs A, B, and C.



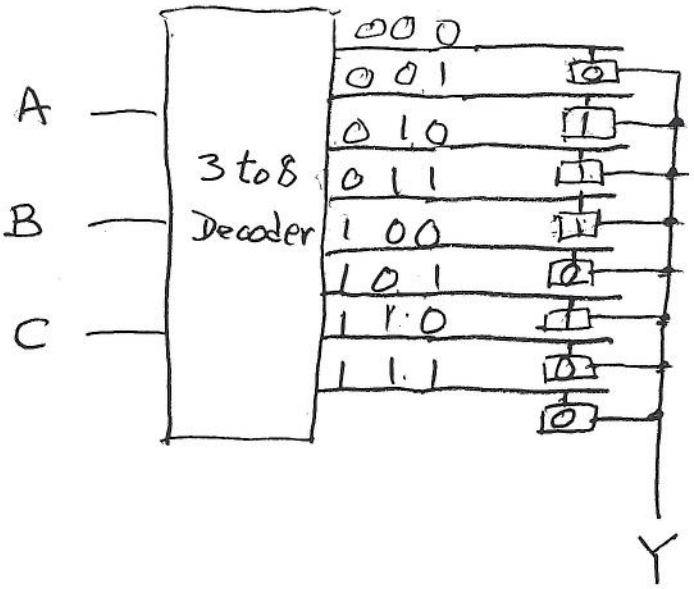
$$Y = \overline{A}\,B + \overline{B}\,C$$

1 (cont.)

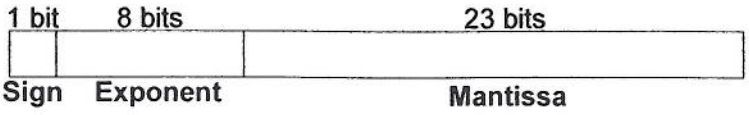
6 pts (c) Show the gate schematic for inputs A, B, and C and out Y. Use the minimum numbers of AND and OR gates.



6 pts (d) Show a lookup table (LUT) implementation of Y as a function of inputs A, B, and C.



2) [24pts] Perform floating point subtraction C= A – B in 32-bit IEEE 754 floating point format



where A = 0x40D00000

B = 0x3FC00000

As floating point numbers

Write the answer C in hexadecimal format.

Extract

(4pts) A: 0 | 1000 0001 | 101 0 ... → 4 0 D

(4pts) B: 0 | 011 1111 | 100 0 ... → 3 F C

(4pts) A: S = 0, E = 129, M = 1.101 (append)

(4pts) B: S = 0, E = 127, M = 1.1

Compare exponent, 129 − 127 = 2

shift B: B << 2 → M = 0.011, E = 129

(4pts) Subtraction

$$\begin{array}{r} 1.101 \\ -\ 0.011 \\ \hline 1.010 \end{array}$$

(drop)

C: S = 0, E = 129, M = 1.01

(2pts) C: 0 | 1000 0001 | 010 ... → 4 0 A

(2pts) C= 0x40A00000

3. [26 pts] In a PCB board design, the output of flip-flop A is buffered and the output signal q is driven 3 cm to the input of flip-flop B. The clk signal and q are carefully laid out next to each other, both traveling with the same speed of 1.5×10^8 m/sec. The flip-flop and buffer timing parameters are given in Tables 1 and 2 respectively.

Table 1 Flip-Flop Timing Parameters

Tpcq	50 ps
Tccq	10 ps
Tsetup	50 ps
Thold	20 ps

Table 2 Buffer Timing Parameters

Tpq	400 ps
Tcq	200 ps

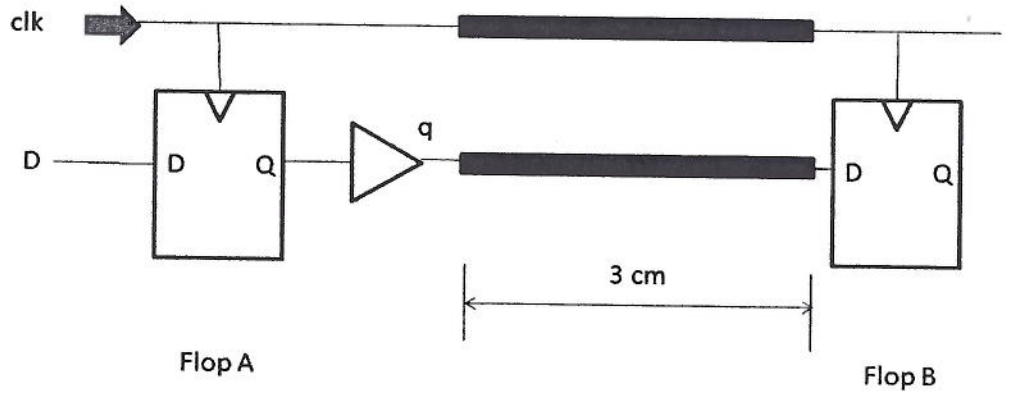
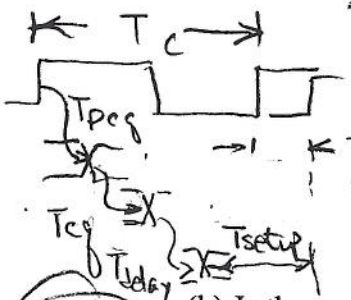


Figure 1

6pts (a) If clk and q travel in the same direction as shown in Figure 1, what is the maximum frequency that the clk can operate?

$T_{skew} = T_{delay} = \frac{3 \times 10^{-2}}{1.5 \times 10^{8}} = 2 \times 10^{-10} = 200\ ps$



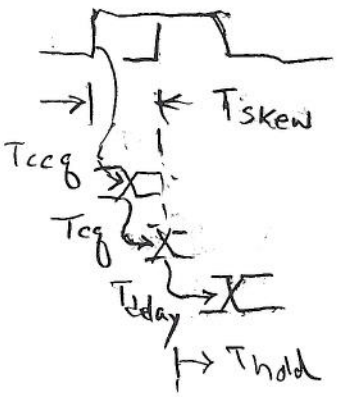
$T_c = T_{pcq} + T_{pq} + T_{delay} + T_{setup} - T_{skew}$

$= 50 + 400 + 200 + 50 - 200$

$= 500\ ps = 0.5\ ns$

$f_{max} = 1/T_c = 2\ GHz$

7pts (b) Is there a hold time problem? Justify your answer.



check

$T_{ccq} + T_{cq} + T_{delay} > T_{hold} + T_{skew}$

$10 + 200 + 200 > 20 + 200$

$210 > 20$ yes

Therefore, no hold time proble

3, (cont.)

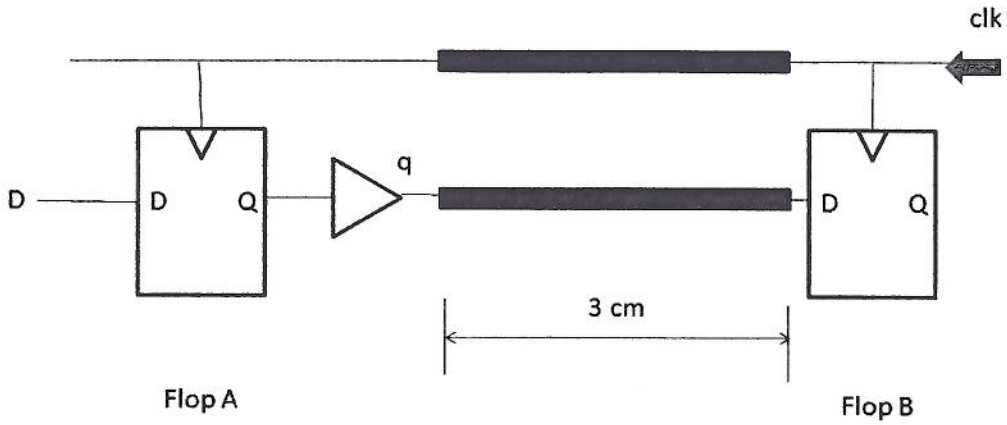
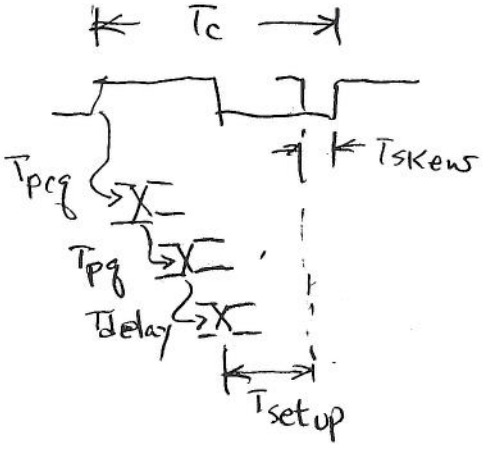


Figure 2

6pts (c) If clk and q travel in the opposite direction as shown in Figure 2, what is the maximum frequency that the clk can operate?



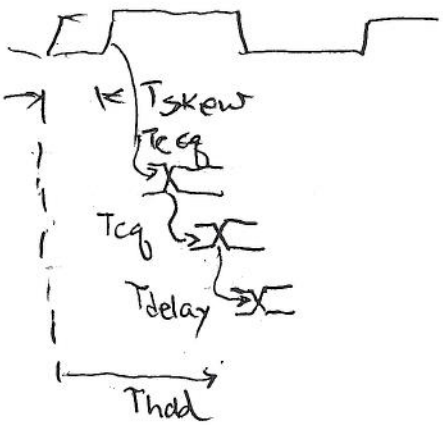
$T_c = T_{pcq} + T_{pq} + T_{delay} + T_{setup} + T_{skew}$

$= 50 + 400 + 200 + 50 + 200$

$= 900 \cdot ps = 0.9\ ns$

$f_{max} = 1/T_c = 1.11\ GHz$

7pts (d) Is there a hold time problem? Justify your answer.



$T_{ccq} + T_{cq} + T_{delay} > T_{hold} - T_{skew}$

$10 + 200 + 200 > 20 - 200$

$410 > -180$ yes

Therefore, no hold time problem

4.[26] A NRZ1 generator for a input data sequence of 1's and 0's is defined as follows:

- For each 0 in the input data sequence, the bit output is the same as the previous bit output.
- For each 1 in the in the input data sequence, the bit output is the complement of the previous output.

For example:

Data input sequence 0 1 1 1 0 0 1 0

Data output sequence 0 1 0 1 1 1 0 0

Assume output is reset to 0 and the input sequence starts at 0.

Design a FSM that generates the output sequence Dout from the input data sequence Din as shown in Figure 4.

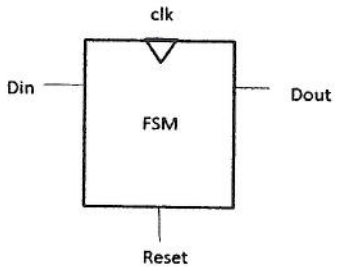
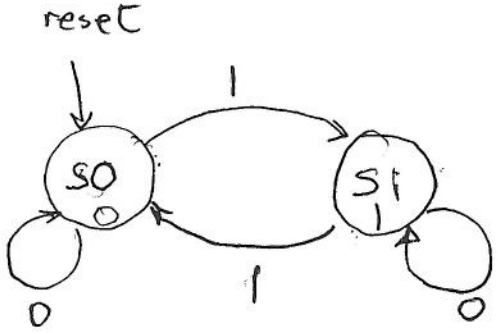


Figure 4

10 pts (a) Show state transition diagram.



4. (cont.)

8pts

(b) Show gate schematic for the FSM design.

Define

S = current state

S' = next state

	S or S'
S0	0
S1	1

transition table

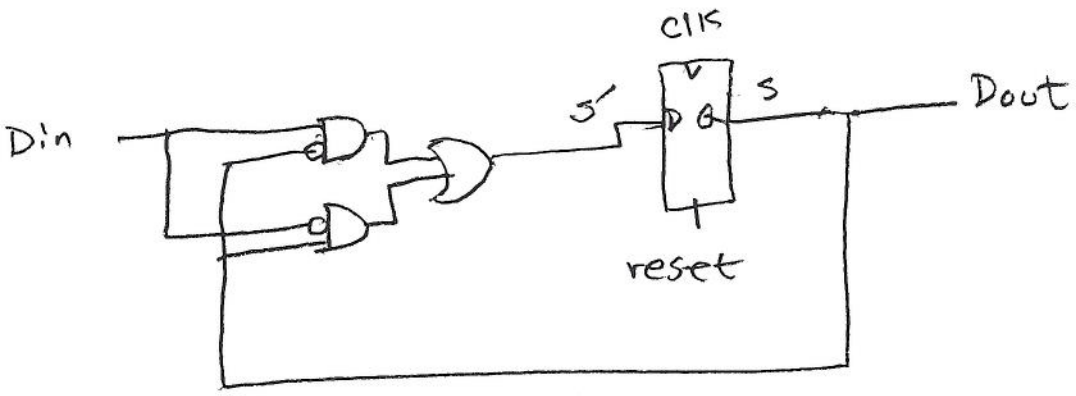
Din	S	S'
0	0	0
1	0	1
0	1	1
1	1	0

Output table

S	Dout
0	0
1	1

$S' = Din \cdot \bar{S} + \overline{Din}\, S$

$Dout = S$



4. (cont.)

8pts (c) Write a HDL description of the FSM design using system Verilog.

```
module NRZI ( input logic din, reset, clk,
              output logic dout );

typedefine enum logic[0] {S0,S1}statetype;
statetype state, nextstate;

always_ff @ (posedge clk or reset);
  if (reset) state <= S0;
  else  state <= nextstate;
// next state logic
  always_comb
  case (state)
      S0 : if (~din)  nextstate = S0 ;
           else       nextstate = S1 ;

      S1 : if (~din)  nextstate = S1;
           else       nextstate = S0;
      default :       nextstate = S0;
  endcase
// output logic
  assign dout = (state == S0) ? 0 : 1 ;

endmodule
```