Answer the questions in the spaces provided. For rough work, you will be provided with supplementary answer sheets. Submit the supplementaries together with this answerbook.

Name and Roll Number: ______________________________

1. Consider the circuit shown in Figure 1. In each of the following fault situations, find an input combination which detects the fault (only the value at the output of the circuit can be observed).

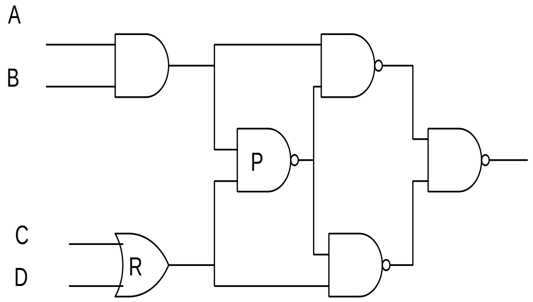

Figure 1: Circuit for problem 1

(a) (3 marks) The output of gate P is stuck at 1. (3)

(b) (3 marks) The output of gate R is stuck at 0. (3)

(c) (4 marks) Gate P behaves like a NOR gate instead of a NAND gate. (4)

2. Consider the circuit shown in Figure 2.

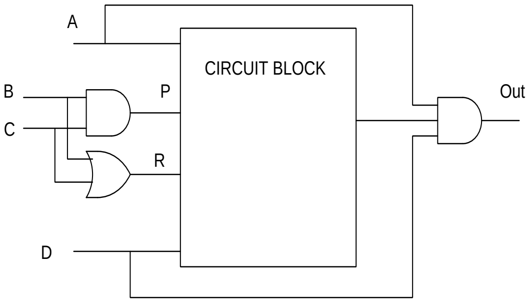

Figure 2: Circuit for problem 2

The circuit contains a circuit-block which needs to be designed. The circuit-block is supposed to compute the XOR of its inputs, and will be used only as part of a larger circuit as shown in Figure 2.

(a) (5 marks) Identify the complete set of don't-care combinations at the input to the circuit-block.

(b) (5 marks) Use the don't cares to obtain a minimum literal sum-of-products realization of the circuit. Show the K-map, prime-implicants and the final simplified formula.

3. Consider the chain of CMOS inverters shown in Figure 3.

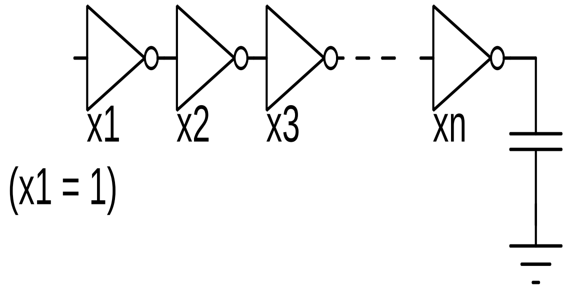

Figure 3: Circuit for problem 3

The i^{th} inverter has a parameter x_i which is the input capacitance of the inverter. The first inverter has $x_1 = 1$, and the last inverter drives a capacitance of value H. You may assume that the delay of an inverter is C_{out}/C_{in} units, where C_{out} is the load capacitance it drives and C_{in} is its input capacitance (that is, ignore the parasitic delay).

(a) (2 marks) For a fixed n, find a formula for the delay of the inverter chain as a function of the $x_i's$.

(b) (3 marks) For a fixed n, find the values of $x_2, \ldots x_n$ which give the minimum possible delays. You may assume the following inequality: Given n positive numbers $u_1, u_2, \ldots u_n$,

$$(\sum_{i=1}^{n} u_i)/n \;\geq\; (\Pi_{i=1}^{n} u_i)^{1/n}.$$

with equality if and only if $u_1 = u_2 = \ldots = u_n$.

(c) (5 marks) What is the value of n for which you will get the smallest possible delay through the inverter chain? How does this delay depend on H?

4. Consider the system shown in Figure 4.

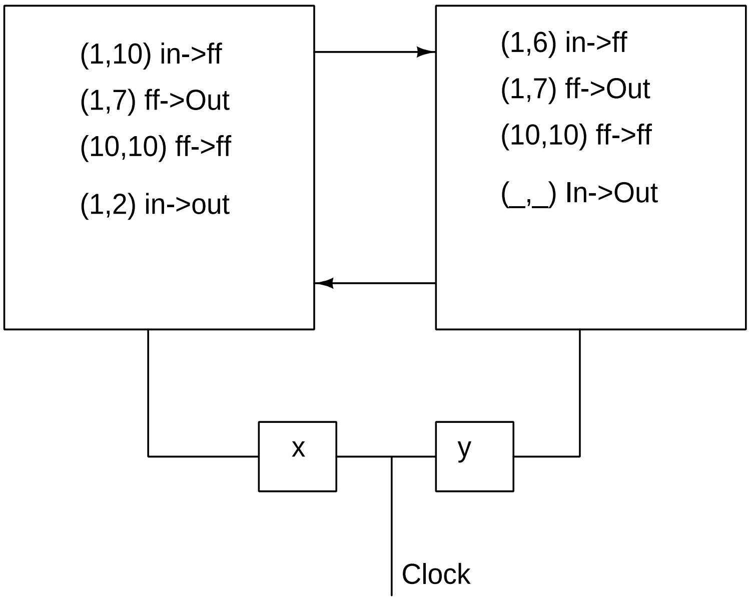

Figure 4: Circuit for problem 4

Both subsystems work on the positive edge of clock. For each subsystem, (min,max) values for four types of delays are indicated in Figure 4: input to flip-flop, flip-flop to output (including the flip-flop clock to q delay), flip-flop to flip-flop (including the flip-flop clock to q delay), and the input to output combinational path delay. A _ indicates that no path of that type exists in the subsystem. The clock wiring delays are also indicated by x and y in Figure 4. Assume that flip-flop setup and hold times are 0.

(a) (2 marks) Assume that the clock wiring delays are 0 (that is, $x = y = 0$). At what clock periods will the circuit operate correctly?

(b) (3 marks) If clock period is fixed to 10 units, for which values of x and y will the system operate correctly?

(c) (5 marks) What is the minimum possible clock-period you can achieve if you are able to control x and y?

5. Using at most two registers and at most two control states, design a data multiplexor whose interface is shown in Figure 5.

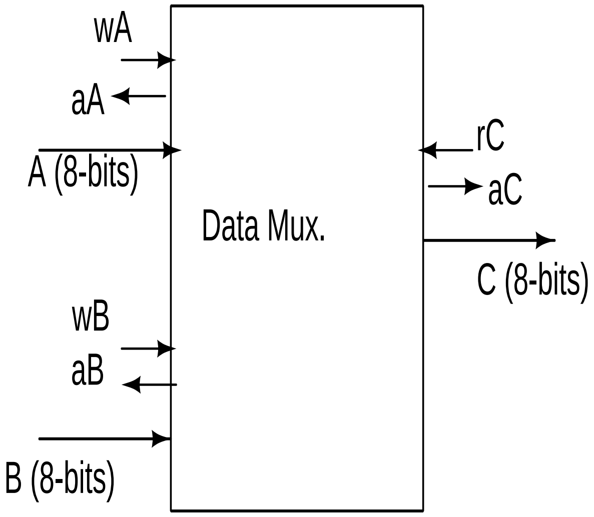

Figure 5: Circuit for problem 5

The behaviour of the multiplexor is as follows: There are two unrelated writers who wish to write values A, B, to the multiplexor using the ready-ready protocol regulated by the signals wA, aA and wB, aB respectively. There is also a reader who wishes to read C using a ready-ready protocol regulated by the signals rC, aC. Whatever is written into the multiplexor cannot be lost, that is it must be preserved until it has been read at C. The multiplexor is to operate at full-rate. That is, at every clock cycle, data must be written into the muliplexor if it is possible to do so. Finally, the multiplexor must be fair: that is if both writers are always trying to write, each should have an equal chance of its data getting into the multiplexor.

(a) Implement an RTL algorithm for the data multiplexor (with at most two control states and at most two registers). Describe using a diagram or pseudo-code in the space provided. (3)

(b) Implement the data path corresponding to your RTL algorithm (using registers, multiplexors, NOT gates and two-input AND/OR gates). Draw the final circuit in the space provided. (4)

(c) Implement the control path corresponding to your algorithm (using D-flipflops, two-input AND/OR gates, NOT gates). Draw the state transition graph and the final circuit in the space provided. (3)