EE224 End-semester Test

18 April 2016

Write the *final answers* to 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.

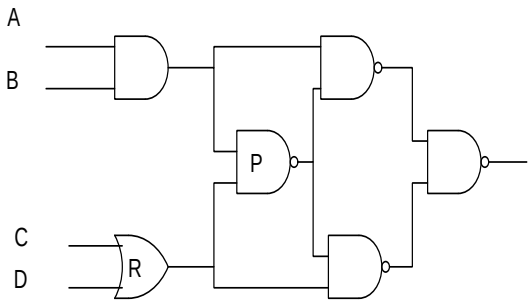


Figure 1: Circuit for problem 1

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). That is, find an input combination which distinguishes between the good circuit and the circuit with the fault introduced.

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

> **Solution:** In terms of P, A, B, C, D, we can write the output $f = A.B.P + P.(C+D)$. Thus $df/dP = A.B+C+D$. For the test, we must have $\overline{P}.df/dP = 1$, and $\overline{P} = A.B.(C + D)$. Thus we must have $(A.B + C + D).A.B.(C + D) = 1$, which is achieved by $A = B = C = 1$, $D = 0$.

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

Solution: In terms of R, A, B, C, D, we can write the output as $f = A.B.\overline{R} + \overline{A.B}.R$. Thus $df/dR = 1$. For the test we must have $R.df/dR = 1$, so that $(C + D) = 1$. Thus $A = B = C = D = 1$ is a possible test.

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

Solution: For the test, we must have $df/dP = A.B + C + D = 1$ and further the input combination to the gate P must exclude the combination when both inputs to the gate are equal, because for this combination the NAND and NOR functions are indistinguishable. Thus $A.B \oplus (C+D)$ must also be 1. So we must have $(A.B + (C + D)).(A.B \oplus (C + D)) = 1$, so that $A = B = 1$, $C = D = 0$ is a possible test.

2. Consider the circuit shown in Figure 2.

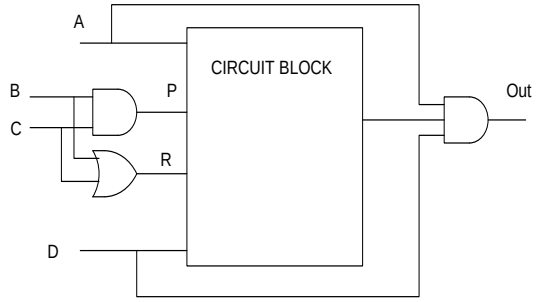


Figure 2: Circuit for problem 2

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.

Solution: Since $P = B + C$, and $R = B.C$, $P = 1, R = 0$ is a don't care since this input combination can never occur at the circuit-block. Further if either $A = 0$ or $D = 0$, then the output of the circuit-block will not influence the overall circuit output, thus $A = 0$ is a don't care and so is $D = 0$.

(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.

Solution: Based on the don't cares, the K-map for the circuit block looks like

```
  AP
RD   00 01 11 10
00   d  d  d  d
01   d  d  d
11   d  d     1
10   d  d  d  d
```

Choose the PI

```
  AP
RD   00 01 11 10
00
01
11   d        1
10   d        d
```

so the circuit-block just needs to implement the formula $\overline{P}.R$.

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

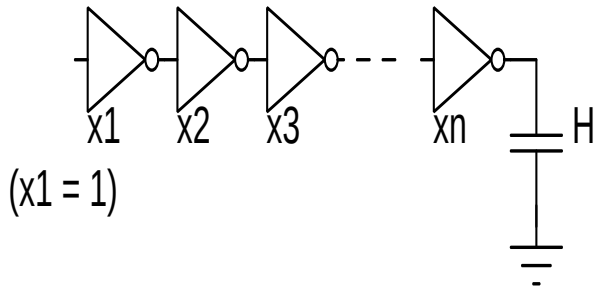


Figure 3: Circuit for problem 3

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 $x_i's$.

Solution: The delay is

$$\frac{x_2}{x_1} + \frac{x_3}{x_2} + \ldots \frac{x_{n-1}}{x_{n-1}} + \frac{H}{x_n}$$

(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 $u_1 = u_2 = \ldots = u_n$.

Solution: Let $u_i = \frac{x_{i+1}}{x_i}$, with $u_n = \frac{H}{x_n}$. Then the delay is

$$\sum_{i=1}^{n} u_n$$

while we have a constraint that

$$\Pi_{i=1}^{N} u_i \;=\; H.$$

The minimum (by the inequality mentioned above) sum is attained when

$$u_1 = u_2 = u_3 = \ldots u_n = H^{1/n}$$

and the best delay is $n.H^{1/n}$.

(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?

Solution: Let $D(n) = n.H^{1/n}$. To get the best n, set $dD/dn = 0$. we get $n = \ln H$. For this n, the best delay is $e \times \ln H$.

4. Consider the system shown in Figure 4.

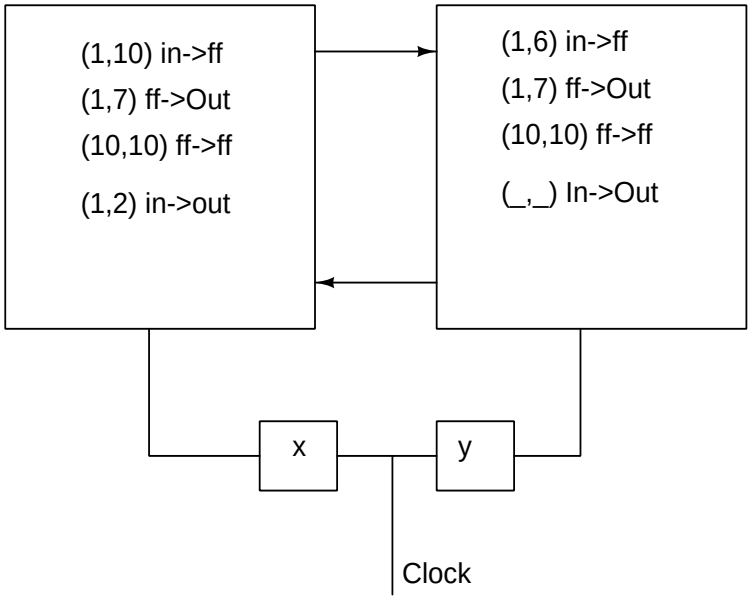
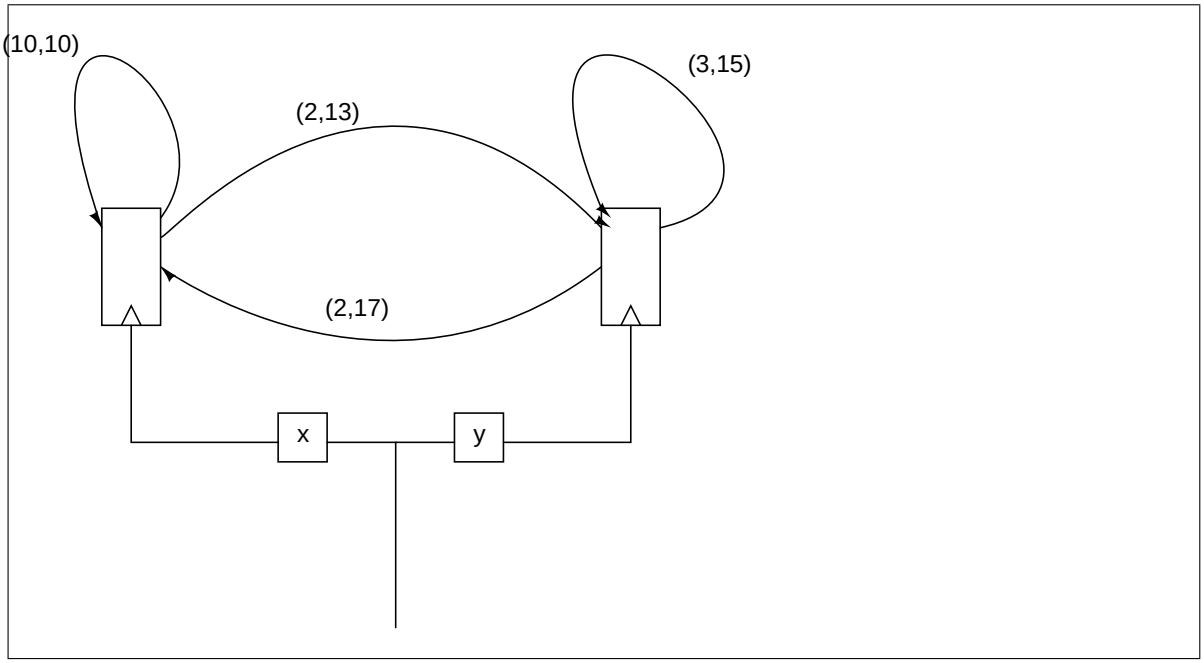


Figure 4: Circuit for problem 4

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.

Solution:

Lets do a bit of analysis. If we look at the possible paths in the circuit, we get the following picture:



(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?

Solution: The circuit will work correctly for clock periods ≥ 17 units. This is because the hold constraints are satisfied and the setup constraint is satisfied for $T \geq 17$ units.

(b) (3 marks) If clock period is fixed to 25 units, for which values of x and y will the system operate correctly?

Solution: As long as $|x - y| \leq 2$, the hold constraints will be satisfied and the circuit will work correctly (note that if $|x - y| \leq 2$, the setup constraints will be met).

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

Solution: We must have $|x - y| \leq 2$, and further

$$\begin{aligned} T &\geq 17 + (y - x) \\ T &\geq 13 + (x - y) \end{aligned}$$

Choose $x - y = 2$ to find that the circuit will work correctly at $T = 15$ units.

5. Using at most one register and exactly one control state, design a data multiplexor whose interface is shown in Figure 5.

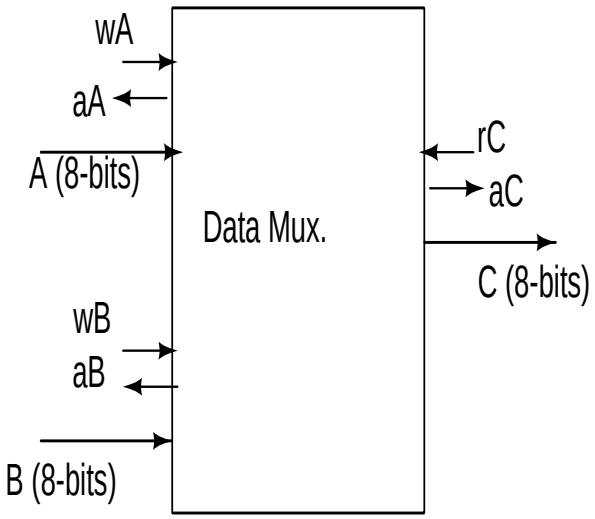


Figure 5: Circuit for problem 5

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) (5 marks) Design an RTL algorithm for the data multiplexor using exactly one control state and one register. Describe your algorithm using a state-transition diagram or pseudo-code in the space provided.

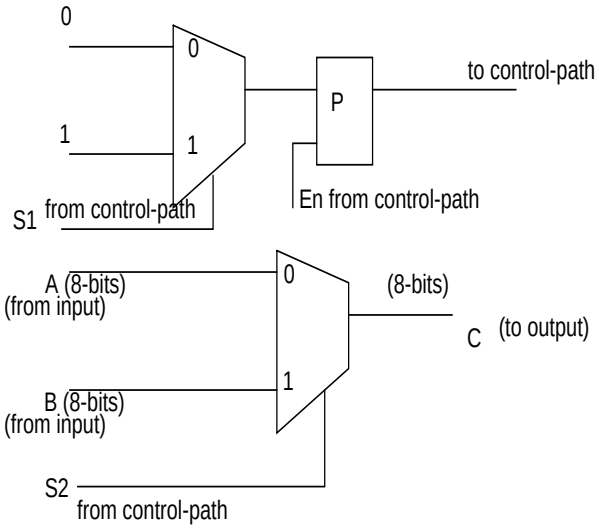
Solution: We use a single bit register P and a single state algorithm:

```
register P[0:0] // single bit.
SingeState:
   if ( (wA.(~wB).rC) +
          (wA.wB.rC.P) ) {
         aA = 1, aC = 1, C = A, P := 0
   }
   else if ((wB.(~wA).rC) +
               (wA.wB.rC.(~P))) {
         aB = 1, aC = 1, C = B, P := 1
   }
   goto SingleState
```

The one-bit register P works like a priority bit. If A and B both are requesting the mux, the priority will switch between the two and guarantee fairness.

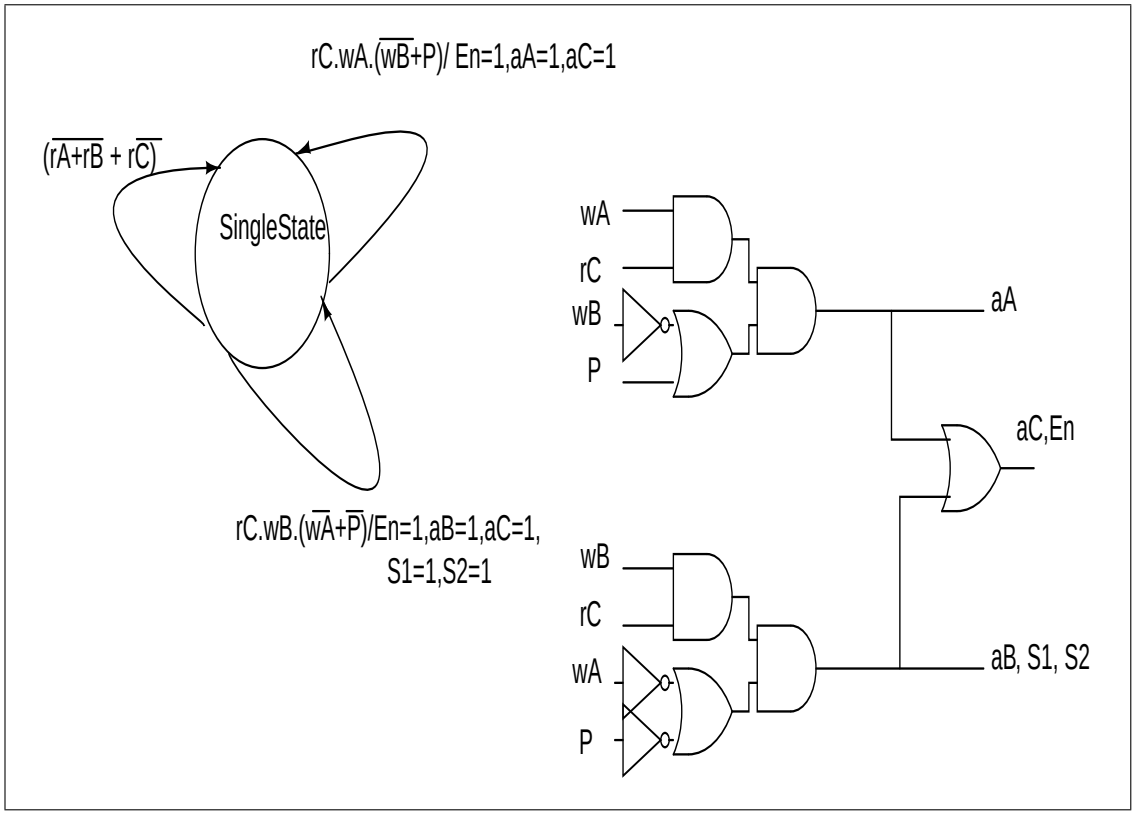
(b) (3 marks) Implement the data path corresponding to your RTL algorithm (using registers, multiplexors, NOT gates and two-input AND/OR gates). Draw the final circuit in the space provided.

Solution:



(c) (2 marks) 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.

Solution: Note that no flip-flop is needed since there is only one state. The control path is just combinational logic. After a bit of simplification, we get the following implementation.


rC.wA.(wB+P)/ En=1,aA=1,aC=1
(rA+rB + rC)
SingleState
rC.wB.(wA+P)/En=1,aB=1,aC=1,
S1=1,S2=1
wA
rC
wB
P
aA
aC,En
wB
rC
wA
P
aB, S1, S2