CS-472: Design Technologies for Integrated Systems

Exercise Problem Set 8 Solution Date: 11/12/2025

Topics: Boolean methods (cf. slide set 12), timing analysis & retiming (cf. slide sets 13 & 14)

Problem 1

Consider the logic network where inputs are $\{a, b, c, d\}$ and output is $\{f\}$ defined as:

$$
\begin{aligned}
k &= ad \\
n &= c + k \\
m &= \overline{(ab)} \\
f &= m + n
\end{aligned}
$$

Assuming $CDC_{in} = ab$, compute CDC_{out}.

cf: Algorithm in textbook p.385

Ans:

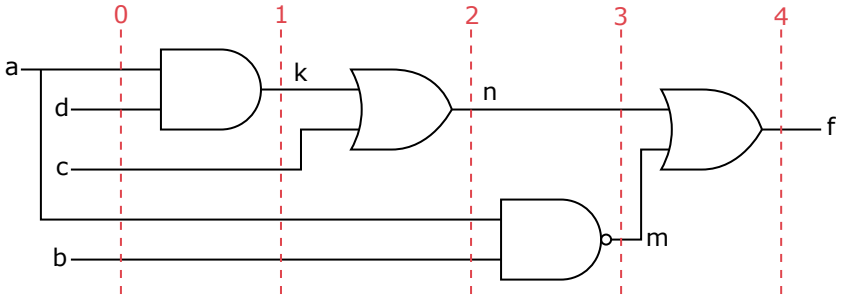

Figure 1: Cuts in the logic circuit during CDC computation

Figure 1 shows some possible succesive *cuts* of the network that we will use to compute the CDC_{out} by network traversal. Remember that, in the algorithm, $C_y(CDC_{cut}) = CDC_{cut_y} \cdot CDC_{cut_{y'}}$ is the consensus of CDC_{cut} with respect to variable y. The CDC computation for each cut proceeds as follows:

0. $CDC_{in} = ab$

1. $CDC_{cut} = ab + (ad \oplus k)$
   drop d: $(ab + (a \oplus k)) \cdot (ab + k) = ab + \bar{a}k$ ($\{a = 1, b = 1\}$ and $\{a = 0, k = 1\}$ will not happen)

2. $CDC_{cut} = ab + \bar{a}k + ((c + k) \oplus n)$
   drop k: $(ab + \bar{a} + \bar{n}) \cdot (ab + (c \oplus n)) = ab + c\bar{n} + \bar{a}\bar{c}n$
   drop c: $(ab + \bar{n}) \cdot (ab + \bar{a}n) = ab$

3. $CDC_{cut} = ab + (\overline{(ab)} \oplus m)$
   drop a: $(b + (\bar{b} \oplus m)) \cdot \bar{m} = \bar{m}$
   drop b: $\bar{m}$

4. $CDC_{cut} = \bar{m} + ((m + n) \oplus f)$
   drop m: $\bar{f}$
   drop n: $\bar{f}$
   $CDC_{cut} = \bar{f}$

Output f is always 1 since m can never be 0.

Problem 2

Consider the logic network from Problem 1. Compute the ODC sets for all internal and input vertices assuming that the output is fully observable.

cf: Algorithm in textbook p.390
Ans:

$$\frac{\partial(m+n)}{\partial m} = n \oplus 1 = \bar{n}, \qquad ODC_m = n = ad + c$$
$$\frac{\partial(m+n)}{\partial n} = m \oplus 1 = \bar{m}, \qquad ODC_n = m = \overline{ab}$$
$$\frac{\partial(c+k)}{\partial k} = c \oplus 1 = \bar{c}, \qquad ODC_k = c + m = c + \overline{ab}$$
$$\frac{\partial(c+k)}{\partial c} = k \oplus 1 = \bar{k}, \qquad ODC_c = k + m = ad + \overline{ab}$$
$$\frac{\partial(ad)}{\partial d} = 0 \oplus a = a, \qquad ODC_d = \bar{a} + c + m = \bar{a} + \bar{b} + c$$
$$\frac{\partial(\overline{ab})}{\partial b} = 1 \oplus \bar{a} = \bar{a}, \qquad ODC_b = \bar{a} + n = \bar{a} + ad + c$$

Signal a has two fanouts: $\{k, m\}$ *cf:* p. 392

$$\frac{\partial(ad)}{\partial a} = 0 \oplus d = d, \qquad ODC_{a,k} = \bar{d} + c + m = \bar{a} + \bar{b} + c + \bar{d}$$
$$\frac{\partial(\overline{ab})}{\partial a} = 1 \oplus \bar{b} = b, \qquad ODC_{a,m} = \bar{b} + n = ad + \bar{b} + c$$

$$\begin{aligned} ODC_a &= ODC_{a,k}|_{a=\bar{a}} \mathbin{\overline{\oplus}} ODC_{a,m} \\ &= (a + \bar{b} + c + \bar{d}) \mathbin{\overline{\oplus}} (ad + \bar{b} + c) \\ &= (\bar{a}b\bar{c}d) \oplus (ad + \bar{b} + c) \\ &= \bar{b} + c + d \end{aligned}$$

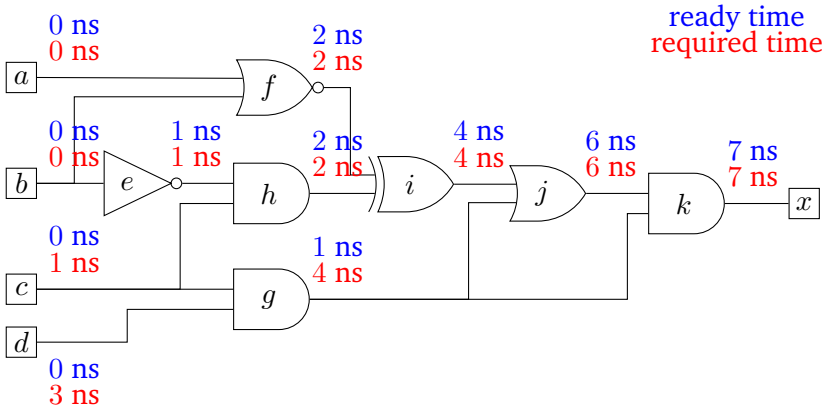

Figure 2: Combinational logic network.

Problem 3

Consider the network in Fig. 2: The inputs are $\{a, b, c, d\}$ and the output is x. The data ready time for all the inputs is 0, and the inputs are stable after time 0. Assume the following delays for logic gates:

INV gate: $t_d = 1$ ns;
AND gate: $t_d = 1$ ns;
OR/NOR/XOR gate: $t_d = 2$ ns.

(a) The data required time for the output x is 7 ns. Find all the topological critical paths.
Ans: $(a, f, i, j, k, x), (b, f, i, j, k, x), (b, e, h, i, j, k, x)$

(b) Is the path (b, e, h, i, j, k, x) a true critical path?
Ans: No. Analysis as follows:

cf: Slide set 13 pp. 26

When $c = 0$ or $d = 0$: $g = 0$, which controls gate k. Either condition (1) or (2) holds.
When $c = d = 1$: $g = 1$, which controls gate j. Either condition (1) or (2) holds.

Problem 4

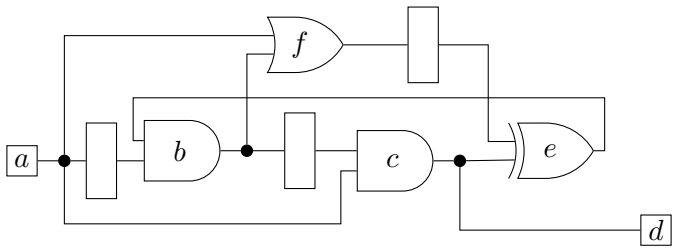

Figure 3: Sequential logic network.

Consider the network in Fig. 3, where the rectangles are registers. Assume the following delays for logic gates:

AND gate: $t_d = 3$ ns;
OR gate: $t_d = 4$ ns;
XOR gate: $t_d = 5$ ns.

(a) Draw the synchronous network graph, where weights of vertices are the combinational delays of the logic gates, and weights of the edges are the numbers of registers between two vertices.
Ans:

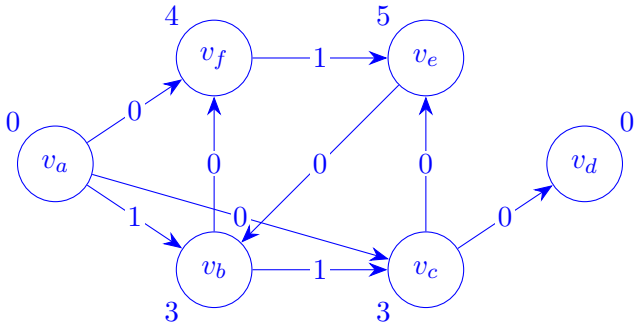

Figure 4: Synchronous network graph.

(b) What is the minimum cycle time currently (before retiming)?
Ans: $3 + 5 + 3 + 4 = 15$ ns (path (v_c, v_e, v_b, v_f))

(c) Draw the constraint graph modeling to search for a legal retiming with a cycle time of 12 ns.
Ans:
$W(v_a, v_b) = 0, D(v_a, v_b) = 11$ $[(v_a, v_c, v_e, v_b)]$.
$W(v_a, v_c) = W(v_a, v_d) = 0, D(v_a, v_c) = D(v_a, v_d) = 3$.
$W(v_a, v_e) = 0, D(v_a, v_e) = 8$ $[(v_a, v_c, v_e)]$. $W(v_a, v_f) = 0, D(v_a, v_f) = 15$ $[(v_a, v_c, v_e, v_b, v_f)]$.
$W(v_b, v_c) = W(v_b, v_d) = 1, D(v_b, v_c) = D(v_b, v_d) = 6$.
$W(v_b, v_e) = 1, D(v_b, v_e) = 12$ $[(v_b, v_f, v_e)]$. $W(v_b, v_f) = 0, D(v_b, v_f) = 7$.
$W(v_c, v_b) = 0, D(v_c, v_b) = 11$. $W(v_c, v_d) = 0, D(v_c, v_d) = 3$.
$W(v_c, v_e) = 0, D(v_c, v_e) = 8$. $W(v_c, v_f) = 0, D(v_c, v_f) = 15$.

$W(v_e, v_b) = 0, D(v_e, v_b) = 8$. $W(v_e, v_c) = W(v_e, v_d) = 1, D(v_e, v_c) = D(v_e, v_d) = 11$.
$W(v_e, v_f) = 0, D(v_e, v_f) = 12$. $W(v_f, v_b) = 1, D(v_f, v_b) = 12$.
$W(v_f, v_c) = W(v_f, v_d) = 2, D(v_f, v_c) = D(v_f, v_d) = 15$. $W(v_f, v_e) = 1, D(v_f, v_e) = 9$.

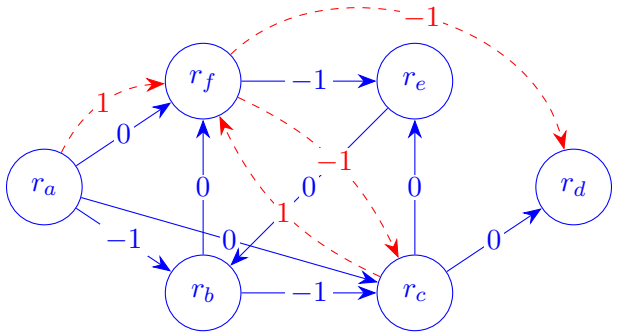

Figure 5: Constraint graph.

(d) To find a retiming solution, apply the Bellman-Ford Algorithm on the constraint graph using r_a as the source (compute the longest distance from r_a to each vertex).
Ans: $r_a = r_b = r_c = r_d = r_e = 0, r_f = 1$ ($\vec{r} = [0, 0, 0, 0, 0, 1]$)

(e) Redraw the retimed synchronous network graph using the retiming vector $\vec{r}$ obtained in (d). Verify that the targeted cycle time 12 ns is met.
Ans: The longest combinational path is now (v_f, v_e, v_b), whose delay is $4 + 5 + 3 = 12$ ns.

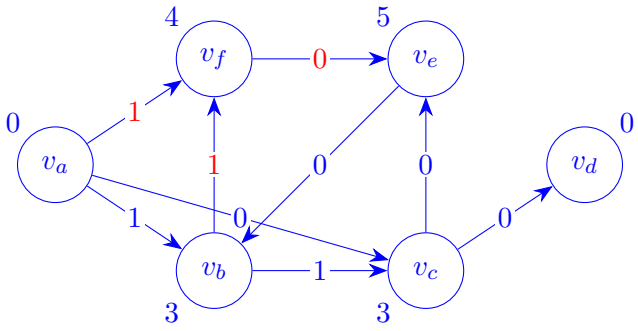

Figure 6: Retimed synchronous network graph.