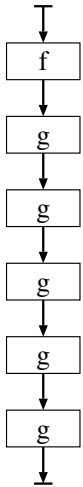

E&CE 327 Final Solution

2010t1 (Winter)

Q1 (17 Marks) Pipelining

Design a dataflow-diagram for the data-dependency graph shown below. The circuit will be implemented on an ASIC. Your goal is to maximize optimality, as defined by $\frac{1000}{\text{ClkPeriod} \times \text{Area}}$, where the clock period is measured in nanoseconds and area is measured in square microns (μm^2).



NOTES:

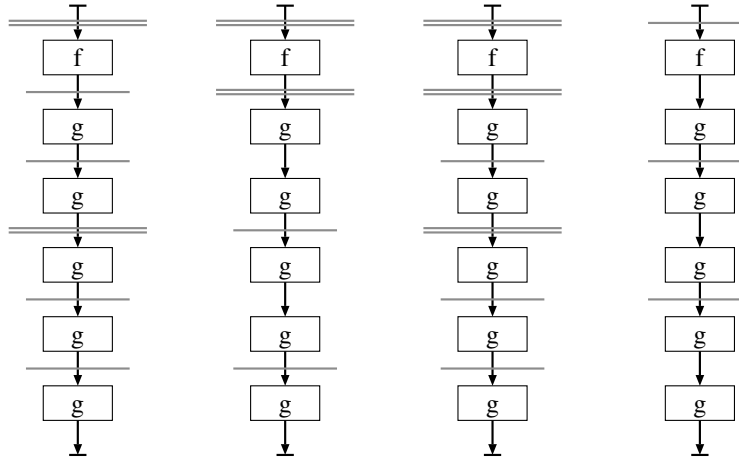
1. The circuit will be implemented on an ASIC, *not* on an FPGA. The area and delay for each component is given in the table below.
2. All of the datapaths are 10 bits wide: each f and g component has one 10-bit input and one 10-bit output.
3. Clock skew, clock jitter, and clock latency are all negligible.
4. The minimum throughput is 1/3.
5. The maximum latency is 8.
6. You must register the inputs, you do *not* need to register the outputs.

| | Area (μm^2) | Delay (ns) |
|-------------------|--------------------------|--|
| f | 12 | 1.7 |
| g | 10 | 1.0 |
| 10-bit reg | 10 | setup: 0.05 hold: 0.07 tco: 0.04 |

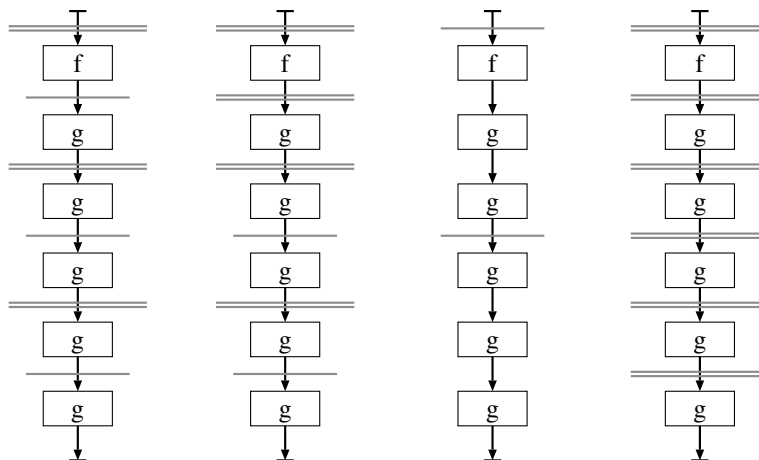
Q1a (10 Marks) Design

Annotate a data-dependency graph below to create a dataflow diagram with the maximum optimality. Multiple copies of the datadependency graph are provided to allow for scratch work. Put a $\sqrt{\quad}$ in the box below the diagram that you wish to be marked.

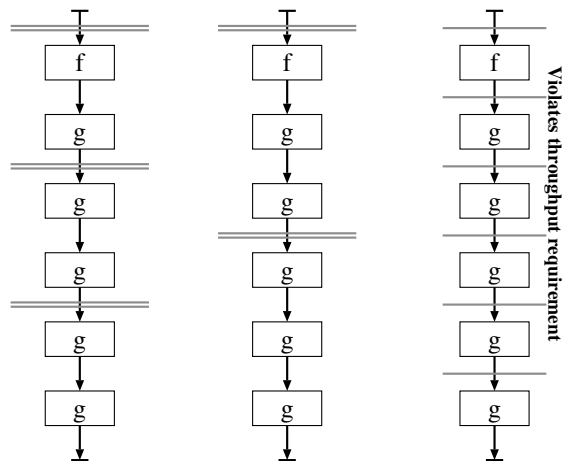
Q1b (7 Marks) Analysis



| | | | | |
|-------------|-----------------|-----------------|-----------------|-----------------------|
| stages: | 2 | 2 | 3 | 1 |
| f: | 1 | 1 | 1 | 1 |
| g: | 2 | 2 | 2 | 2 |
| area: | 52 | 52 | 62 | 42 |
| period: | $1.7+0.09=1.79$ | $2.0+0.09=2.09$ | $1.7+0.09=1.79$ | $1.0 + 1.7+0.09=2.79$ |
| opt: | 10.74 | 9.2 | 9.01 | 8.53 |
| throughput: | 1/3 | 1/3 | 1/3 | 1/3 |
| latency: | 6 | 4 | 6 | 3 |



| | | | | |
|-------------|-----------------|-----------------|-------------------------|-----------------|
| stages: | 3 | 4 | 1 | 6 |
| f: | 1 | 1 | 1 | 1 |
| g: | 3 | 3 | 3 | 5 |
| area: | 72 | 82 | 52 | 122 |
| period: | $1.7+0.09=1.79$ | $1.7+0.09=1.79$ | $1.7+1.0+1.0+0.09=3.79$ | $1.7+0.09=1.79$ |
| opt: | 7.76 | 6.81 | 5.07 | 4.58 |
| throughput: | 1/2 | 1/2 | 1/2 | 1 |
| latency: | 6 | 6 | 2 | 6 |



| | | | |
|-------------|---------------------------|---------------------------------|---------------------|
| stages: | 3 | 2 | 1 |
| f: | 1 | 1 | 1 |
| g: | 5 | 5 | 1 |
| area: | 92 | 82 | 32 |
| period: | $1.0 + 1.7 + 0.09 = 2.79$ | $1.7 + 1.0 + 1.0 + 0.09 = 3.79$ | $1.7 + 0.09 = 1.79$ |
| opt: | 3.90 | 3.22 | 17.46 |
| throughput: | 1 | 1 | 1/6 |
| latency: | 3 | 2 | 6 |

For dataflow diagram:

Marking:

If satisfies design requirements and $\text{opt} \geq 5$, then $\text{mark} = \lfloor \text{opt} \rfloor$

If violate requirements, or optimality is less than 5:

- 7** *Good optimality, but violate requirements*
- 4** *Legal and complete dataflow diagram*
- 2** *Partial dataflow diagram with substantial work*
- 1** *Partial work*
- 2** *missing register on inputs*
- 1** *ambiguous stage boundaries*

For analysis:

Marking:

- 1** *area*
- 2** *period*
- 1** *optimality*
- 2** *throughput*
- 1** *latency*

Q2 (17 Marks) Functional Verification**Q2a (3 Marks) Functional Simulation**

For a typical digital circuit, what percentage of its behaviour can be verified with functional simulation during the design process?

In the table below, x represents the percentage of a typical circuit's behaviour that can be verified with functional simulation during the design process. Answer the question by writing a \checkmark in the box of the row that best characterizes the range in which x would lie.

Answer:

| | | |
|--------------------------|-------------------------------------|--------|
| $0.00\% \leq x < 0.01\%$ | <input checked="" type="checkbox"/> | 3marks |
| $0.01\% \leq x < 1.00\%$ | <input type="checkbox"/> | 2marks |
| $1\% \leq x < 25\%$ | <input type="checkbox"/> | 1mark |
| $25\% \leq x < 75\%$ | <input type="checkbox"/> | 1mark |
| $75\% \leq x \leq 100\%$ | <input type="checkbox"/> | 1mark |

Q2b (5 Marks) Coverage Monitors

Give an example of a coverage monitor that could be used in a Kirsch edge detector. (You may use whatever notation is most appropriate: equation, schematic, VHDL code, diagram, or text.)

Answer:

There are many correct answers. An example answer is:

A coverage monitor for a derivative could detect different ranges of values for the derivative: 3825, 3825..390, 389..384, 383, 382..375, 374..1, 0: where 3825 is the maximum value of a derivative and 383 is the threshold.

Either describe the purpose of your coverage monitor, or describe how would you use your coverage monitor.

Answer:

The coverage monitor reports which range the derivative is in. If a range does not appear in the report, then we know that no tests exercised that range. We would then create additional tests to exercise those cases that are not triggered.

The marking rules list the key points for this question:

Marking:

2 marks coverage monitors are for functional verification (not fault testing)

2 marks coverage refers to whether a case is tested (does not check for correctness of result)

1 mark add test cases until coverage monitor is triggered

Q2c (9 Marks) Simulation Options

One day at the lunch table in the cafeteria, your manager says that she recently learned that Y⁰-Sim, the VHDL simulator that you use, has split their simulator into two products: one for functional simulation and one for timing simulation. She is considering buying either just the functional simulator or just the timing simulator, and using the money that she saves to buy more computers to speed up simulation.

Your current functional verification methodology uses a mixture of functional simulation, timing simulation, and running on the FPGA board, just as you were taught in ECE-327.

Your manager describes three options: “FTB”, “FCB”, and “TCB”; where “F” means functional simulation, “T” means “timing simulation”, “C” means new computers, and “B” means FPGA board. All three options cost the same.

FTB Buy both the functional simulator and timing simulator, and continue with the current methodology.

FCB Don't buy the timing simulator; use functional simulation and FPGA boards for functional verification. Use the money saved by not buying the timing simulator to buy more computers, which will allow you to *run functional simulation 10-times faster than you do now*.

TCB Don't buy the functional simulator; use timing simulation and FPGA boards for functional verification. Use the money saved by not buying the functional simulator to buy more computers, which will allow you to *run timing simulation at the same speed as you currently run functional simulation*.

For each option, answer whether you think it should be chosen as the best option, considered as a possibility, or rejected. If you recommend that an option be chosen, then you must reject the other two options.

For each option, briefly justify your recommendation in terms of its advantages and/or disadvantages.

Answer:

There is no single right or wrong answer. Good points to make in the analysis are listed below under marking.

Marking:

2 marks *FTB allows current methodology to be continued: methodology is trusted and reliable, no need for additional training.*

2 marks *Functional simulation is insufficient; need timing simulation or running on board to detect timing errors*

2 marks *Running functional simulation faster provides a minimal increase in actual coverage, because functional coverage is so low*

2 marks *Debugging is much easier with functional simulation than with timing simulation, and debugging with timing simulation is much easier than debugging on the board.*

2 marks *Could use the board as an alternative to timing simulation.*

2 marks *Without functional simulation, the design would need to be synthesizable before it could be simulated.*

Q4 (16 Marks) Elmore Delay

In this question you will use the Elmore delay model to analyze the maximum clock speeds of three different layouts of the gate-level schematic shown below.

Answer:

First, identify the slowest node(s) in each layout:

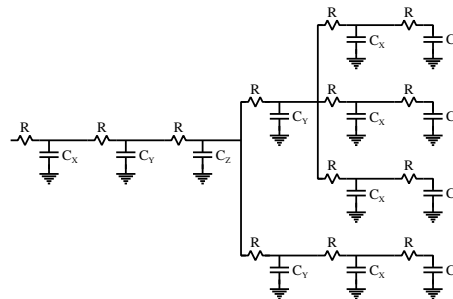
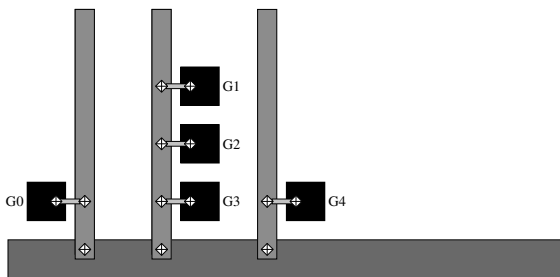
- *Layout-1:* G1, G2, and G3. These nodes will be slower than G4, because the Z-Y via leading to these nodes has greater downstream capacitance than the Z-Y via leading to G4.
- *Layout-2:* G1 and G2. Same reasoning as for Layout-1.
- *Layout-3:* G1, G2, G3, and G4.

We can answer the question without doing detailed Elmore-delay calculations:

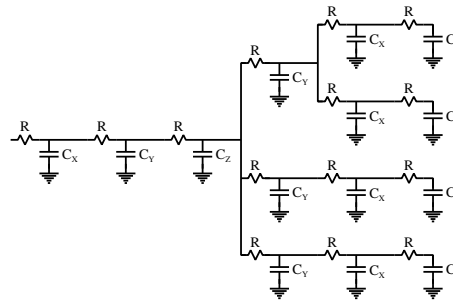
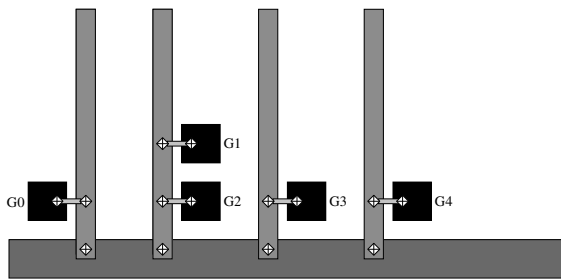
Layout-1 vs layout-2 For the Y-Z via coming from G0, layout-2 has more downstream capacitance than layout-1, because of the additional C_Y . However, the downstream capacitance for the Z-Y via leading to node G1 is greater for layout-1 than for layout-2. Thus, in the delay equations for G1, some factors will be greater for layout-1 and some will be greater for layout-2.

Layout-2 vs Layout-3 For each resistor (via) on the path from G0 to G1, the downstream capacitance is the same except for the additional C_Y in layout-2. Therefore, layout-2 is slower than layout-3.

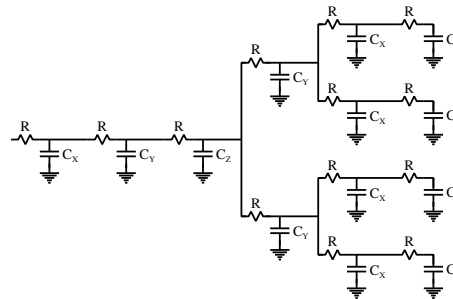
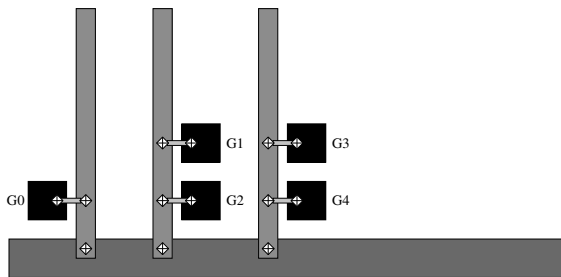
If we do detailed Elmore-delay calculations:



$$\begin{aligned}
 & 5RC_X + 6RC_g \\
 & + 4R(C_Y + 2(C_X + C_g)) \\
 & + 3R(C_Y + C_X + C_g) \\
 & + RC_x + 2RC_Y + 3RC_Z
 \end{aligned}$$



$$\begin{aligned}
 &5RC_X + 6RC_g \\
 &+ 4R(C_Y + C_X + C_g) \\
 &+ 3R(2(C_Y + C_X + C_g)) \\
 &+ RC_x + 2RC_Y + 3RC_Z
 \end{aligned}$$



$$\begin{aligned}
 &5RC_X + 6RC_g \\
 &+ 4R(C_Y + C_X + C_g) \\
 &+ 3R(C_Y + 2(C_X + C_g)) \\
 &+ RC_x + 2RC_Y + 3RC_Z
 \end{aligned}$$

Summing up the RC values for each layout:

| Layout-1 | | | | Layout-2 | | | | Layout-3 | | | |
|----------|-------|-------|-------|----------|-------|-------|-------|----------|-------|-------|-------|
| C_X | C_Y | C_Z | C_g | C_X | C_Y | C_Z | C_g | C_X | C_Y | C_Z | C_g |
| 5 | | | 6 | 5 | | | 6 | 5 | | | 6 |
| 8 | 4 | | 8 | 4 | 4 | | 4 | 4 | 4 | | 4 |
| 3 | 3 | | 3 | 6 | 6 | | 6 | 6 | 3 | | 6 |
| 1 | 2 | 3 | | 1 | 2 | 3 | | 1 | 2 | 3 | |
| 17 | 9 | 3 | 17 | 16 | 12 | 3 | 16 | 16 | 9 | 3 | 16 |

$$\begin{aligned}
 \text{Layout - 1} & 17RC_X + 9RC_Y + 3RC_Z + 17RC_g \\
 \text{Layout - 2} & 16RC_X + 12RC_Y + 3RC_Z + 16RC_g \\
 \text{Layout - 3} & 16RC_X + 9RC_Y + 3RC_Z + 16RC_g
 \end{aligned}$$

The speed of Layout-1 cannot be compared to layout-2. Because in comparing the Elmore delay of layout-1 to layout-2: C_X is greater in layout-1 but C_Y is less.

Layout-2 is slower than layout-3, because the only capacitance that is different is C_Y , which is greater for Layout-2.

If use G4 as slowest node:

| <i>Layout-1</i> | | | | <i>Layout-2</i> | | | |
|----------------------------|-------|-------|-------|----------------------------|-------|-------|-------|
| $4RC_Y + 5RC_X + 6RC_g$ | | | | $4RC_Y + 5RC_X + 6RC_g$ | | | |
| + $3R(C_Y + 3(C_X + C_g))$ | | | | + $3R(C_Y + C_X + C_g)$ | | | |
| + $RC_x + 2RC_Y + 3RC_Z$ | | | | + $3R(C_Y + 2(C_X + C_g))$ | | | |
| + $RC_x + 2RC_Y + 3RC_Z$ | | | | + $RC_x + 2RC_Y + 3RC_Z$ | | | |
| C_X | C_Y | C_Z | C_g | C_X | C_Y | C_Z | C_g |
| 5 | 4 | | 6 | 5 | 4 | | 6 |
| 9 | 3 | | 9 | 3 | 3 | | 3 |
| 1 | 2 | 3 | | 6 | 3 | | 6 |
| 15 | 9 | 3 | 15 | 1 | 2 | 3 | |
| | | | | 15 | 12 | 3 | 15 |

Layout-1 is faster than layout-2, because the only capacitance that is different is C_Y , which is less for layout-1.

The speed of layout-2 cannot be compared to the speed of layout-3, because C_X is greater for layout-3 and C_Y is greater for layout-2.

Marking:

For parts a and b:

For answers that did not do a detailed Elmore delay analysis, marks were awarded based on the correctness and clarity of the answer, roughly with the same weighting as answers with detailed Elmore delay analysis.

2 marks *RC-network*

2 marks *RC-equation*

1 mark *RC answer*

1 mark *speed answer*

Q4a (4 Marks) Nodes to Measure

To determine the maximum clock speed at which layout-1 would work correctly, at which node(s) would you measure the delay?

NOTES:

- If any other nodes will have the same delay as the node that you would measure, list these nodes as well.

Marking:

4 marks *G1, G2, G3*

3 marks *G1*

2 marks *G4*

1 mark *some partial work*

Q5 (17 Marks) Clock Gating

Your task is to analyze a proposed clock-gating scheme.

NOTES:

1. The latency through the main circuit is 15 clock cycles.
2. The average length of a continuous sequence of valid parcels is 45.
3. The area of the clock-gating circuit is 1/8 that of the main circuit.
4. Short-circuiting and leakage power are negligible.

What is the minimum number of bubbles between valid parcels, if the circuit with clock gating is to have a maximum of 90% of the power of the original circuit? If you are unable to reduce the power to be 90% of the power of the original circuit, then calculate the minimum power that you can achieve with clock gating.

Answer:

$$\begin{aligned} P_{tot} &= P_{main} \\ P'_{tot} &= P'_{main} + P_{cg} \end{aligned}$$

$$\begin{aligned} P &= P_{switch} + P_{short} + P_{leak} \\ P_{short} &= 0 \\ P_{leak} &= 0 \\ P_{switch} &= \frac{1}{2}ACfV^2 \\ \text{Same } f \text{ and } V \text{ for all circuits} \end{aligned}$$

Equations from problem statement

$$\begin{aligned} \frac{P'_{tot}}{P_{tot}} &= 0.9 \\ C_{cg} &= \frac{1}{8}C_{main} \end{aligned}$$

$$\begin{aligned} P_{cg} &= \frac{1}{2}A\frac{1}{8}C_{main}fV^2 \\ P'_{main} &= \frac{1}{2}A'C_{main}fV^2 \end{aligned}$$

Solve for A'

$$\begin{aligned} 0.9 &= \frac{P'_{main} + P_{cg}}{P_{main}} \\ &= \frac{(\frac{1}{2}A'C_{main}fV^2) + (\frac{1}{2}A\frac{1}{8}C_{main}fV^2)}{\frac{1}{2}AC_{main}fV^2} \\ &= \frac{A' + \frac{1}{8}A}{A} \\ A' &= 0.9A - 0.125A \\ &= 0.775A \end{aligned}$$

$$\begin{aligned} \text{NumClkEn} &= \text{NumValid} + \text{Latency} \\ &= 45 + 15 \\ &= 60 \end{aligned}$$

$$A' = \frac{\text{NumClkEn}}{w} A$$

$$\begin{aligned} w &= \text{NumClkEn} \frac{A}{A'} \\ &= 60 \frac{A}{0.775A} \\ &= 77.419 \end{aligned}$$

$$\begin{aligned} \text{NumBubbles} &= w - \text{NumValid} \\ &= 77.419 - 45 \\ &= 32.419 \end{aligned}$$

The minimum number of bubbles is 33.

Marking:

4 marks $A' = 0.775A$

4 marks $\text{NumClkEn} = 60$

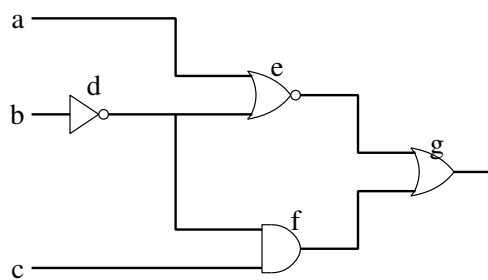
5 marks $w = 77.419$

4 marks $\text{NumBubbles} = 33$

Q6 (17 Marks) Testing

The travel department at your company accidentally sent your ticket to the “Manufacturing and Faults Conference (MFC)” to the VP of Marketing, and sent his ticket to the “Marketing Frolics Cabanal ($\mathcal{M}\mathcal{C}$)” to you. When the VP of Marketing returned from the faults conference, he was all excited about the hot topic of the conference, which was the “Single-stuck-at-1” (SS1) fault model. Unfortunately, despite his enthusiasm, he knows almost nothing about faults or testing, and so he has asked you to prepare a report for him to present to senior management.

Find the minimum set of test vectors to catch all SS1 faults in the circuit below, and list the test vectors in the order to run them, from first to last.



NOTES:

1. If you do not know how to answer the question using the SS1 fault-model, then you for part-marks, you may answer the question using the fault model that we use in ECE-327. If you do so, write a \checkmark in the box:
2. The probability of a fault occurring on a wire driving a 1-input gate (e.g. inverter) is half that of the probability of a fault occurring on other wires. Your test vectors must still detect such a fault.
3. Write an “X” in the box for any test vector that is *not* needed.
4. There are copies of the circuit and Karnaugh-map templates on the pages following this question.

Answer:

kmap notation:

```

c=  0  1  1  0
b=  0  0  1  1
a=0 [ ] [#] [#] [#]
a=1 [ ] [#] [ ] [ ]

```

good expression

```

[ ] [#] [#] [#]
[ ] [#] [ ] [ ]

```

faults difference k

| | |
|-----------------|-----------------|
| L1@1 | L1@1 |
| [] [#] [] [] | [] [] [#] [#] |
| [] [#] [] [] | [] [] [] [] |

| | |
|-----------------|-----------------|
| L2@1 | L2@1 |
| [#] [#] [#] [#] | [#] [] [] [] |
| [] [] [] [] | [] [#] [] [] |

| | |
|-----------------|-----------------|
| L3@1 | L3@1 |
| [#] [#] [#] [#] | [#] [] [] [] |
| [#] [#] [] [] | [#] [] [] [] |

| | |
|-----------------|-----------------|
| L4@1 | L4@1 |
| [] [#] [#] [] | [] [] [] [#] |
| [] [#] [#] [] | [] [] [#] [] |

| | |
|-----------------|-----------------|
| L5@1 | L5@1 |
| [] [#] [] [] | [] [] [#] [#] |
| [] [#] [] [] | [] [] [] [] |

| | |
|-----------------|-----------------|
| L6@1 | L6@1 |
| [] [#] [#] [#] | [] [] [] [] |
| [] [#] [#] [] | [] [] [#] [] |

| | |
|-----------------|-----------------|
| L7@1 | L7@1 |
| [#] [#] [#] [#] | [#] [] [] [] |
| [#] [#] [#] [#] | [#] [] [#] [#] |

| | |
|-----------------|-----------------|
| L8@1 | L8@1 |
| [#] [#] [#] [#] | [#] [] [] [] |
| [#] [#] [#] [#] | [#] [] [#] [#] |

| | |
|-----------------|-----------------|
| L9@1 | L9@1 |
| [#] [#] [#] [#] | [#] [] [] [] |
| [#] [#] [#] [#] | [#] [] [#] [#] |

gate collapsing

L1@1 L5@1

L9@1 L7@1 L8@1

fault domination

L4@1 dom by [L6@1]

L9@1 dom by [L3@1,L6@1]

 required test vectors

111 (from L6@1)

[][][][]

[][][#][]

 unrequired test vectors

L1@1

[][][#][#]

[][][][]

L2@1

[#][][][]

[][#][][]

L3@1

[#][][][]

[#][][][]

 minimum test vectors

000

[#][][][]

[][][][]

010

[][][][#]

[][][][]

 catches #1

000 [L2@1,L3@1,L7@1,L8@1,L9@1]

010 [L1@1,L4@1,L5@1]

111 [L4@1,L6@1,L7@1,L8@1,L9@1]

 test vector #1: L6@1

catches #2

000 [L2@1,L3@1]

010 [L1@1,L5@1]

test vector #2: 010

test vector #3: 000

Marking:

- 1 mark *behaviour of correct circuit*
- 2 marks *fault locations*
- 1 mark *diff kmaps*
- 2 mark *gate collapsing*
- 2 marks *fault domination*
- 2 marks *required test vectors*
- 2 marks *"unrequired" test vectors*
- 1 mark *minimum set of test vectors*
- 3 marks *ordering of vectors*
- 2 marks *used single-stuck-at fault model*