ECE 327 Solution to Final

2014t1 (Winter)

		Total Marks	Approx. Time	Page
Q1	Simulation	5	7	2
Q2	Retiming	10	18	3
Q3	Dataflow Diagram Design	20	20	5
Q4	Optimality	15	20	10
Q5	Longest Path	10	15	14
Q6	Glitches	10	15	16
Q7	Power and Performance	15	20	17
Q8	Clock Gating	15	20	20
Totals		100	135	

Q1 (5 Marks) Simulation

(*estimated time: 7 minutes*)

What would happen if you tried to use the register-transfer-level simulation algorithm to simulate a circuit that the algorithm could not simulate? Obviously there will be a problem. Describe where in the simulation algorithm a problem will first occur and describe what the problem will be.

Answer:

Circuits with combinational loops are not supported by register-transfer-level simulation.

The problem will occur when the register-transfer-level simulation algorithm attempts to do a topological sort of the decomposed combinational processes.

The processes in the combinational loop do not have a topological order. The sorting algorithm will either get stuck in an infinite loop or will detect the loop and fail.

Marking:

+2 mark *combinational loop*

+2 mark *topo sort*

+1 mark *can't topo sort, or get stuck in infinite loop*

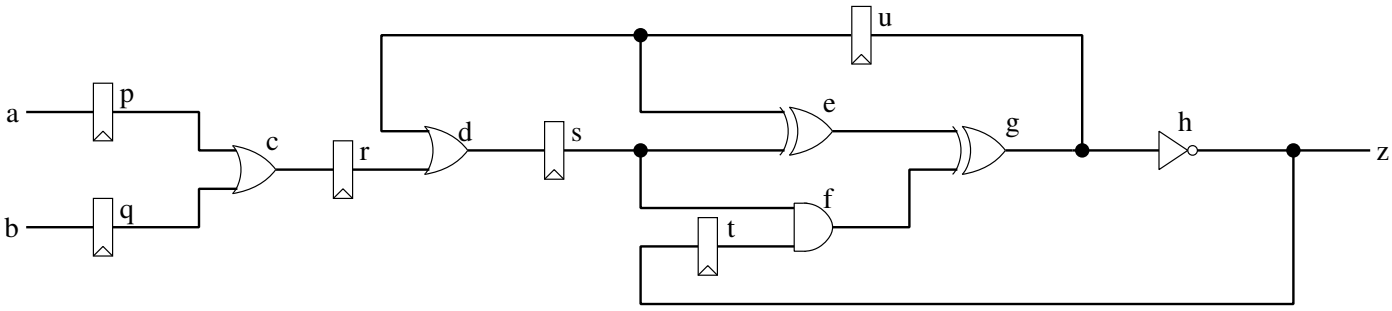
Q2 (10 Marks) Retiming

(*estimated time: 18 minutes*)

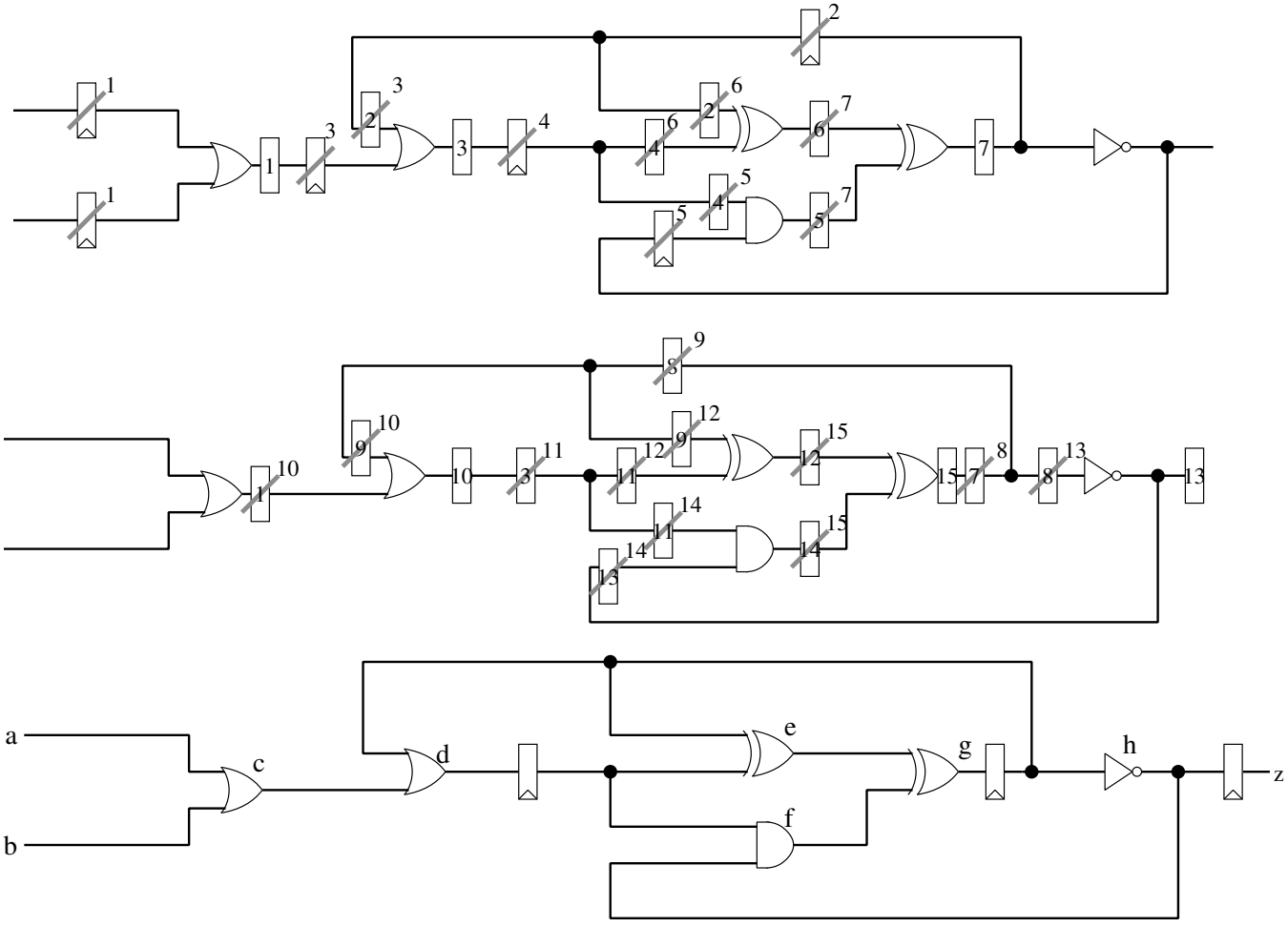
Use retiming to modify the circuit below without changing the behaviour of the output `z`.

NOTES:

1. Goals, in order of *decreasing* importance:
   (a) Minimize the number of registers in the circuit.
   (b) Make `z` be the output of a register.
2. The inputs and combinational gates shall *not* be modified.



Answer:



Marking:

+1 mark *no flops on a, b, c*

+1 marks *exactly 1 flop on e, g loop*

+1 marks *exactly 2 flops on d, e, g loop*

+1 marks *exactly 2 flop on d, f, g loop*

+1 marks *exactly 3 flops on (a,b), c, d, e, g, h, z path and (a,b), c, d, f, g, h, z path*

+1 mark *z is a flop*

+4 marks *flop count (2 marks max if functional bugs)*

- **4 marks** *3 flops*
- **3 marks** *4 flops*
- **1 mark** *5 flops*
- **0 mark** *6 flops*

Q3 (20 Marks) Dataflow Diagram Design

(*estimated time: 20 minutes*)

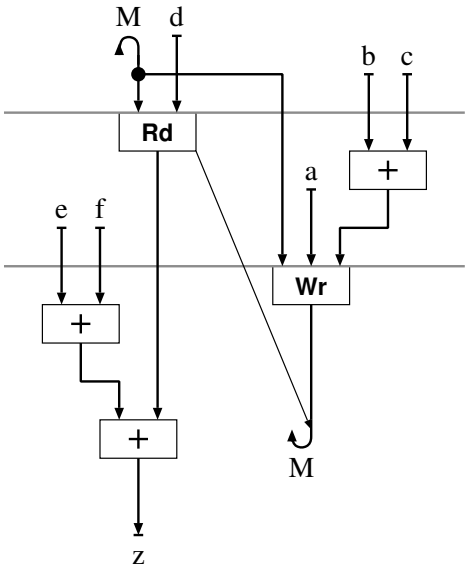
Draw a dataflow diagram for the system described below:

```
M[b+c] = a;
     p = M[d];
     z = p + e + f;
```

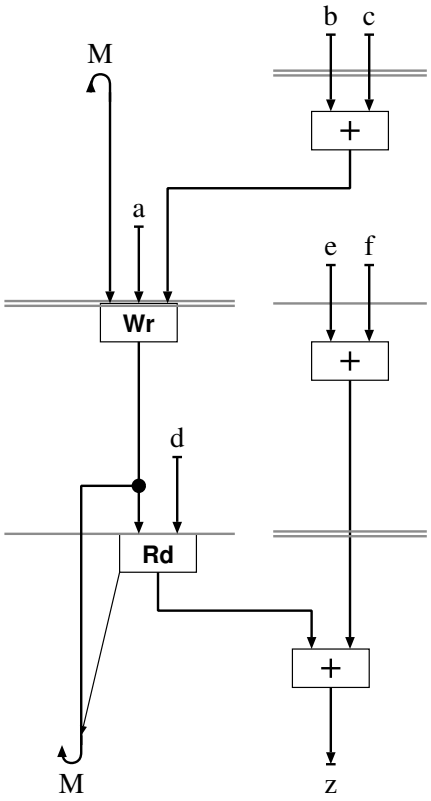
NOTES:

1. The inputs are `a`, `b`, `c`, `d`, `e`, and `f`.
2. It is guaranteed that `b+c` $\neq$ `d`.
3. The output is `z`.
4. The *inputs* shall be registered.
5. The *output* may be either registered or combinational.
6. The memory shall be *single-ported*.
7. Optimization goals, in order of *decreasing* importance:
   (a) Maximum *throughput*
   (b) Minimum number of *inputs*
   (c) Minimum *latency*
   (d) Minimum number of *adders*
   (e) Minimum number of *registers* (*excluding memory*)
   (f) Minimum *clock period*
8. You may use *algebraic optimizations*, as long as the values of `z` and `M` are correct.
9. Memory read, memory write, and a 2-input adder all have the same delay of t.
10. The memory shall have registered inputs and combinational outputs (same as in class).
11. Input values may be read in any clock cycle, but each input value shall be read exactly once.
12. You do *not* need to do any allocation.

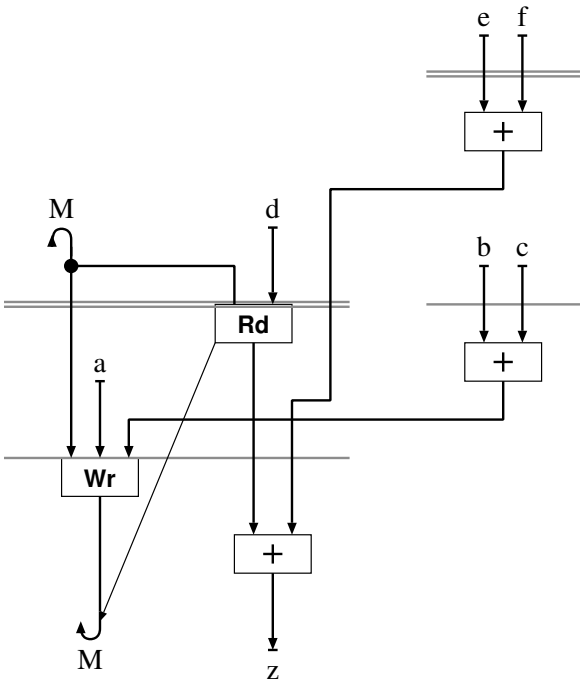
Answer:



Latency	*2*	*Inputs*	*3*
Throughput	$\frac{1}{2}$	*Regs*	*3*
Period	*Flop + 2t*	*Adds*	*2*



Latency	*3*	*Inputs*	*3*
Throughput	$\frac{1}{2}$	*Regs*	*3*
Period	*Flop + 2t*	*Adds*	*2*



Latency	*2*	*Inputs*	*3*
Throughput	$\frac{1}{2}$	*Regs*	*3*
Period	*Flop + 2t*	*Adds*	*2*

functional correctness	+3 marks
stage boundaries clear and correct	+2 marks
optimality	15 marks

mem operations not on clock cycle boundaries	-2 marks
dual port memory	-2 marks
memory rd and wr in diff stages	-2 marks
combinational inputs	-1 mark

Other criteria

anti-dependency arrow
DFD uses M
M is an inter-parcel variable
M has one write port and one read port
Wr produces M
correct analysis of performance and area

1 mistake	-1 mark
2–3 mistakes	-2 marks
4–5 mistakes	-3 marks
6–8 mistakes	-4 marks
9–12 mistakes	-5 marks

Throughput	1/2							1/3	1/1
Inputs	3							4	
Latency	2		3			4	4	4	
Adds	2		2			3	4	2	
Regs	3	4	3		4			4	
Period	2t	2t	t	2t	t			t	
Mark	20	19.5	19	18.5	18	17.5	17	16	15

Marking:

+3 marks *functional correctness*
+2 marks *stage boundaries clear and correct*
+4 marks *maximum throughput* ($\frac{1}{2}$)
+3 marks *minimum inputs (3)*
+3 marks *minimum latency (3)*
+2 marks *minimum adders (2)*
+2 marks *minimum registers (3)*
+1 marks *minimum clock period* (t)

-2 marks *mem operations on clock cycle boundaries*
-1 mark *registered inputs*
Other criteria

- *anti-dependency arrow*
- *DFD uses M*
- *M is an inter-parcel variable*
- *M has one write port and one read port*
- *Wr produces M*
- *correct analysis of performance and area*

-1 mark *1 mistake*
-2 marks *2–3 mistakes*
-3 marks *4–5 mistakes*

Q4 (15 Marks) Optimality

(*estimated time: 20 minutes*)

For Waterluvian filters, optimality is a ratio of the number of pixels produced to the resources (time and hardware) used to produce the pixels:

$$Opt = \frac{Pixels}{Time \times Hardware}$$

From this basic concept, we can define different versions of optimality using different measures of time and hardware.

The *consumers' measure of optimality* for Waterluvian filters is the number of pixels produced per second by a Waterluvian chip:

$$\begin{aligned} Opt_c &= \text{consumers' measure of optimality} \\ &= \frac{Pixels}{Sec \times Chip} \end{aligned}$$

As the size of transistors decreases with each new generation of chips:

- The delay through an FPGA cell decreases
- The number of cells per chip increases

We define the *design optimality* as a measure that is independent of the characteristics of a particular FPGA chip. For design optimality:

- The base unit of time is the delay through a single FPGA cell.
- For area, we count the number of FPGA cells that the Waterluvian filter uses.

$$\begin{aligned} D &= \text{delay through a single FPGA cell} \\ Opt_d &= \text{design optimality} \\ &= \frac{Pixels}{D \times Cells} \end{aligned}$$

Using the data below, calculate the average *percentage change in design optimality per month* for Waterluvian filters.

- Consumer's optimality increases by 2.5% per month
- Delay through an FPGA cell decreases by 1% per month
- Number of cells per chip doubles every 2 years

Answer:

First, a brief discussion about measuring time.

*D is the delay through a cell (*e.g.*, 0.6 ns).*

If we port a design to a new chip that has half the delay through a cell and the same number of cells per chip, the consumers' view of optimality will be that optimality has doubled, because the performance in terms of pixels-per-second will have doubled. However, we know that the increase in performance came only from the chip, the design optimality *has not changed: the pixels per clock cycle is unchanged.*

Because different designs have different clock periods, pixels per clock cycle is not a good measure of optimality. We use pixels per cell-delay, because this measure allows a fair comparison of different choices for how many cell delays to have per clock cycle, and is independent of differences in timing for different chips.

We can think of design optimality as measuring the number of pixels produced per 1000 cell delays divided by the number of cells.

1. *Given information*

Consumer's optimality	Opt_c	*increases by 2.5% per month*
Delay through an FPGA cell	$\frac{Sec}{D}$	*decreases by 1% per month*
Number of cells per chip	$\frac{Cells}{Chip}$	*doubles every 2 years*

   When we use $\frac{Sec}{D}$ *to say that the delay through a cell decreases, we are measuring the number of seconds per gate delay. When we say that a quantity increases, we need to specify the scope for measuring the quantity. For example, it is meaningless to say that the number of students increases, because we do not know whether we are talking about the number of students per class, per university, per province, or ... Even when we discuss the number of stars, we implicitly scope it with "in the sky", or "in the universe".*
   The correct ratio is $\frac{Sec}{D}$*, not* $\frac{D}{Sec}$ *because if D decreases from 0.6 ns to 0.4 ns, the value of* $\frac{D}{Sec}$ increases.
2. *Strategy*
   (a) Find rate of increase for cells per chip by month
   (b) Equation for consumers' optimality in terms of design optimality and chips' characteristics
   (c) Solve for rate of increase in design optimality

3. *Find rate of increase for cells per chip by month*

$$\begin{aligned}\frac{Cells}{Chip} &= 2^{1/24} \\ &= 1.0293\end{aligned}$$

4. *Equation for consumers' optimality in terms of design optimality and chips' characteristics*

$$\begin{aligned}Opt_c &= Opt_d \times ... \\ \frac{Pixels}{Sec \times Chip} &= \frac{Pixels}{D \times Cells} \times \frac{D}{Sec} \times \frac{Cells}{Chip} \\ Opt_c &= Opt_d \times \frac{D}{Sec} \times \frac{Cells}{Chip} \\ Opt_d &= Opt_c \times \frac{Sec}{D} \times \frac{Chip}{Cells}\end{aligned}$$

5. *Solve for rate of increase in design optimality*

$$\begin{aligned}\frac{Opt_{d2}}{Opt_{d1}} &= \frac{Opt_{c2}}{Opt_{c1}} \times \frac{\left(\frac{Sec_2}{D}\right)}{\left(\frac{Sec_1}{D}\right)} \times \frac{\left(\frac{Chip}{Cell_2}\right)}{\left(\frac{Chip}{Cell_1}\right)} \\ &= \frac{Opt_{c1}}{Opt_{c2}} \times \frac{\left(\frac{Sec_2}{D}\right)}{\left(\frac{Sec_1}{D}\right)} \times \frac{\left(\frac{Cell_1}{Chip}\right)}{\left(\frac{Cell_2}{Chip}\right)} \\ &= 1.025 \times \frac{1}{1.01} \times \frac{1}{1.0293} \\ &= 0.9860\end{aligned}$$

6. *Final answer:*
   Design optimality changes by $\frac{0.9860-1}{1} = -1.4\%$ *each month.*

Cells-per-chip per month	+3 marks
Top-level equation relating different optimalities	+3 marks
Delay through FPGA cell rate decrease to increase	+2 marks
Solve for design optimality rate of change	+3 marks
Interpretation of final answer	+2 marks
Clarity of strategy	+2 marks

Marking:

- **+3 marks** *Cells-per-chip per month*
- **+3 marks** *Top-level equation relating different optimalities*
- **+2 marks** *Delay through FPGA cell rate of decrease to rate of increase*
- **+3 marks** *Solve for design optimality rate of change*
- **+2 marks** *Interpretation of final answer*
- **+2 marks** *Clarity of strategy*

Q5 (10 Marks) Longest Path

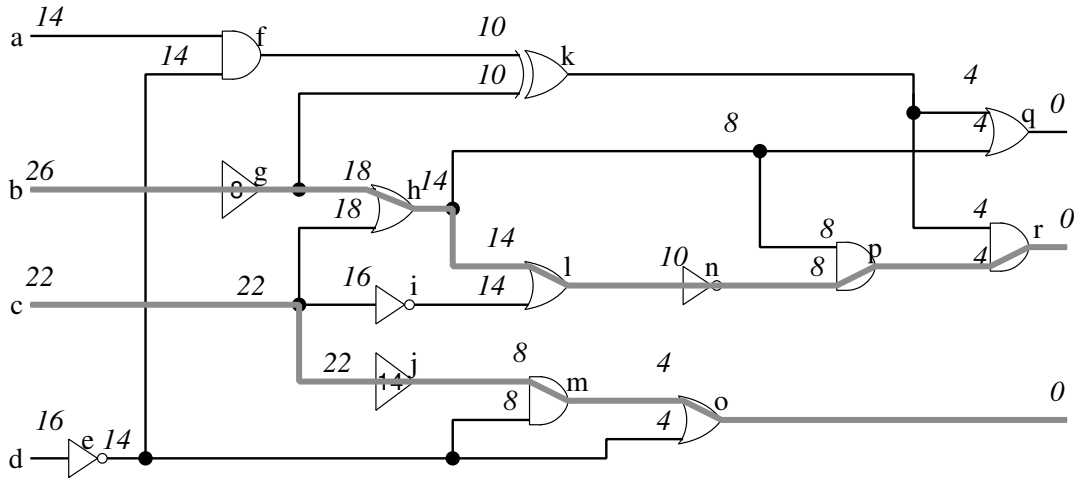
(*estimated time: 15 minutes*)

In the circuit below, the two longest paths, $\langle \mathtt{b},\mathtt{g},\mathtt{h},\mathtt{l},\mathtt{n},\mathtt{p},\mathtt{r}\rangle$ and $\langle \mathtt{c},\mathtt{j},\mathtt{m},\mathtt{o}\rangle$ are false paths. Find the third longest path.

NOTES:

1. A *buffer* is drawn as a triangle with its delay written inside the triangle. The output of the buffer is simply a delayed version of the input.

Answer:



14	*f*	*a*
16	e^{16}	*d*
22	h^{18}, i^{16}, j^{22}	*c*
20 ~~*26*~~	k^{10}, ~~h^{18}~~	*b, g*
$26-(14-8)=20$ ~~*26*~~	p^{8}, q^{4}, ~~l^{14}~~	*b, g, h*
~~*26*~~	$\emptyset$	*b, g, h, l, n, p, r* ***false***

14	*f*	*a*
16	e^{16}	*d*
20	k^{10}, ~~h^{18}~~	*b, g*
20	p^{8}, q^{4}	*b, g, h*
$22-(22-18)=18$ ~~*22*~~	h^{18}, i^{16}, ~~j^{22}~~	*c*
~~*22*~~	$\emptyset$	*c, j, m, o* ***false***

14	*f*	*a*
16	e^{16}	*d*
18	h^{18}, i^{16}	*c*
20	k^{10}, ~~h^{18}~~	*b, g*
20	~~p^{8}~~, q^{4}	*b, g, h*
20	$\emptyset$	*b, g, h, p, r*

The third longest path is $\langle \mathtt{b},\mathtt{g},\mathtt{h},\mathtt{p},\mathtt{r}\rangle$

$\langle \mathtt{b}, \mathtt{g}, \mathtt{h}, \mathtt{p}, \mathtt{r} \rangle$	10 marks
used path table	+3 marks
updated potential delay for $\langle \mathtt{b}, \mathtt{g}, \mathtt{h} \rangle$	+2 marks
updated potential delay for $\langle \mathtt{b}, \mathtt{g} \rangle$	+2 marks
updated potential delay for $\langle \mathtt{c} \rangle$	+2 marks
correct algebra	+1 mark

Marking:

10 marks $\langle \mathtt{b}, \mathtt{g}, \mathtt{h}, \mathtt{p}, \mathtt{r} \rangle$
+3 marks *used path table*
+2 marks *updated potential delay for* $\langle \mathtt{b}, \mathtt{g}, \mathtt{h} \rangle$
+2 marks *updated potential delay for* $\langle \mathtt{b}, \mathtt{g} \rangle$
+2 marks *updated potential delay for* $\langle \mathtt{c} \rangle$
+1 mark *correct algebra*

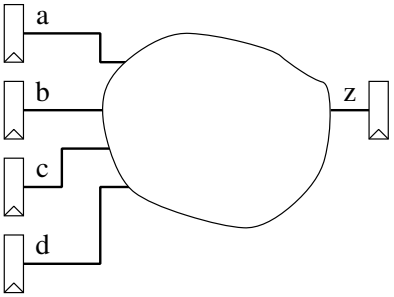
Q6 (10 Marks) Glitches

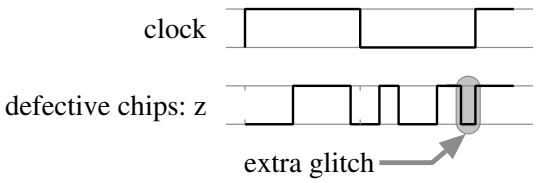
(*estimated time: 15 minutes*)

You have begun to manufacture a new circuit, but a few of the chips are defective. The post-silicon verification engineers have identified that the defective chips have a setup violation on the critical path.

NOTES:

1. The critical path is from `b` to `z`. The specific gates in the circuit are irrelevant to this question.
2. The critical path was found by running timing simulation for all possible excitations of each combinational path using the maximum delay of each gate. (This complete simulation was feasible because it is a small circuit.)
3. The setup violation is the extra glitch in the waveform below, which does not occur in timing simulation with maximum delays or with correct chips.





Answer:

*The most likley cause is monotone speedup. The circuit was simulated using the **maximum** delay through each gate. Most of the actual circuits have less than the maximum delay. With monotone speedup, an edge does not propagate along the critical path when all gates have the maximum delay. But, if some gates on a false path are sufficiently fast, then an edge can propagate along the critical path. The chips with the glitch have faster-than-average gates on a false path, which allows an edge to propagate along the critical path.*

Marking:

+2 marks *monotone speedup*

+2 marks *simulated with max delays*

+2 marks *some gates have better than worst-case delays / variability*

+2 marks *late-arriving side input with non-controlling value from false path*

+2 marks *path cannot be exercised with max delays, but can be exercised with less-than-max delays*

+1 mark *define setup violation*

+1 mark *explain how a glitch could be caused with early side input*

+1 mark *explain how fix setup violation*

Q7 (15 Marks) Power and Performance

(*estimated time: 20 minutes*)

You are the project leader for a Waterluvian filter and are planning to switch to a new FPGA chip.

Summary of characteristics of new and old FPGA chips:

Characteristic	Ratio of value on New FPGA to Old FPGA
Capacitance per cell	0.20
Area per cell	0.25
Delay per cell	0.30
Supply voltage	1.00

Your task is to determine the maximum performance gain that you can achieve by switching to the new FPGA chip.

NOTES:

1. Performance is measured in mega-pixels per second.
2. The only change that you may make to your design is to adjust (add, delete, and/or move) clock cycle boundaries.
3. The *total power consumption* (Watts) and the *power density* (Watts per mm^2) must stay the same or decrease when you switch to the new FPGA chip.

   If you do not know how calculate the power density, you may earn *part marks* by writing a $\surd$ in the box below and ignoring power density.

   I am ignoring power density. ☐

1. List any assumptions you need:

Answer:

- *Short circuiting and leakage power are negligible.*
- *Optimizations do not change total cell count.*
- *Can ignore LUT vs reg in analysis, because only given information for* cells*.*

Marking:

+1 mark *short circuiting power*

+1 mark *leakage power*

2. What is the maximum performance gain that can be achieved?

Answer:

(a) Performance is mega-pixels per second, which is a measure of throughput in time. Maximize increase in clock speed to maximize performance gain. Keep activity factor constant, because do not change design.

(b) Identifiers

P	*Power*
P_d	*Power density*
C	*Capacitance per cell*
A	*Area per cell*
V	*Supply voltage*
α	*Activity factor*
F	*Clock frequency*

(c) Strategy

i. find maximum clock speed while keeping power constant

ii. find maximum clock speed while keeping power density constant

(d) Constant power

$$\begin{aligned} P &= \frac{1}{2}\alpha FCV^2 \\ P_1 &= P_2 \\ \frac{1}{2}\alpha F_1 C_1 V^2 &= \frac{1}{2}\alpha F_2 C_2 V^2 \\ F_1 C_1 &= F_2(0.2C_1) \\ F_2 &= \frac{F_1}{0.2} \\ &= 5F_1 \end{aligned}$$

Keeping power constant allows clock speed to increase by a factor of 5.
Keeping power constant allows clock speed to increase by 400%.

(e) Constant power density

$$\begin{aligned} P_d &= \frac{1}{2}\alpha F C V^2 \frac{1}{A} \\ P_{d1} &= P_{d2} \\ \frac{1}{2}\alpha F_1 C_1 V^2 \frac{1}{A_1} &= \frac{1}{2}\alpha F_2 C_2 V^2 \frac{1}{A_2} \\ F_1 \times C_1 \times \frac{1}{A_1} &= F_2 \times 0.2 C_1 \times \frac{1}{0.25 A_1} \\ F_2 &= \frac{0.25}{0.2} F1 \\ &= 1.25 F1 \end{aligned}$$

Keeping power density constant allows clock speed to increase by 25%.

(f) Limiting factor is power density.

(g) Maximum performance gain is 25%.

Marking:

+1 mark *Performance proportional to frequency*
+1 mark *Keep activity factor constant*
+1 mark *Keep power and power-density constant*
+1 mark *Equation for power*
+1 mark *Equation for power density*
+1 mark *Delay through cell is irrelevant*
+1 mark *Solve for maximum clock speed*
+2 marks *Clarity and correctness of strategy*
+1 mark *Interpretation of result*

3. What is the primary limitation to further increases in performance?

Answer:
Power density

Marking:

3 marks *power density*
1.5 marks *clock speed or power*
0.5 marks *other answer with some merit*

Q8 (15 Marks) Clock Gating

(*estimated time: 20 minutes*)

After 16-months of covert surveillance of Piazza groups, Interpol has determined that a professor at an obscure Canadian university outside of Toronto is using Waterluvian filters to make counterfeit Bitcoins. Because of your expertise in Waterluvian filters, you have been hired by Interpol to infiltrate this criminal organization and pose as the project leader for the next generation Waterluvian filter.

An intern from some other university was assigned the task of adding clock-gating to the Waterluvian filter. When doing the power analysis, the component *without* clock gating has lower power than the component *with* clock gating. Unfortunately, the intern has now returned to school and the organization's policy prevents any communication with students once they have finished their employment.

NOTES:

1. The component with clock gating is identical to the original component except for the addition of clock gating.
2. Both components were analyzed for the same FPGA chip, clock speed, and supply voltage.
3. The clock gating is functionally correct.

Based only upon the information above, give two hypotheses for the most likely reason that adding clock gating caused the power consumption to increase.

Answer:

Hypothesis 1 *The area of the clock enable state machine is so large that it consumes more power than is saved by clock-gating the main circuit.*

Hypothesis 2 *The percentage of time that the main circuit has valid data is so high that the power savings is less than the power consumption of the clock-enable state machine.*

Hypothesis 3 *The activity factor of the main main circuit is so low that the power savings is less than the power consumption of the clock-enable state machine.*

Test procedure

1. *Use synthesis to measure area of main circuit.*
2. *Use synthesis to measure area of clock enable state machine.*
3. *Use timing simulation to measure activity factor of clock-enable state machine and main circuit.*
4. *Use simulation to measure percentage of time that main circuit is turned off.*
5. *Calculate potential power saving in main circuit with clock gating.*
6. *Calculate power consumption of clock gating circuit.*

Marking:

+3 marks *Hypothesis 1*

- **+1 mark** *clock gating ckt uses more power than saves*

+3 marks *Hypothesis 2*

+9 marks *Test procedure*

- **+3 marks** *Methodical*
- **+3 marks** *Detailed*
- **+3 marks** *Tests hypotheses*