E&CE 327 Final Solution

2012t2 (Spring)

Q1 (20 Marks) Dataflow Diagram

In this question, you will design and analyze a dataflow diagram for the equation: $a - (b + c) + (a - b) \times c \times 7 - 3$.

NOTES:

1. Timing

Register setup	2.2 ns
Register hold	1.0 ns
Register clock-to-Q	3.0 ns
Clock skew	0.4 ns
Clock jitter	0.9 ns
Clock latency	0.7 ns
Delay through adder	6.0 ns
Delay through subtracter	7.0 ns
Delay through multiplier	16.0 ns
Delay through multiplexer	0.0 ns

NOTES:

1. Requirements
   (a) Registered inputs.
   (b) Combinational outputs.
   (c) The throughput shall be at least 1/3.
2. Goals (from highest priority to lowest)
   (a) Minimize clock period
   (b) Minimize area
   - Order of priority from highest to lowest: Multipliers, total of adders and subtracters, inputs, registers.
   - Multiplexers are free

   (c) Maximize throughput
3. You may use *algebraic optimizations*.
4. You may schedule the input values to arrive in any order, but you may read each input value only once.
5. You do *not* need to do any allocation.

Q1a (15 Marks) Design

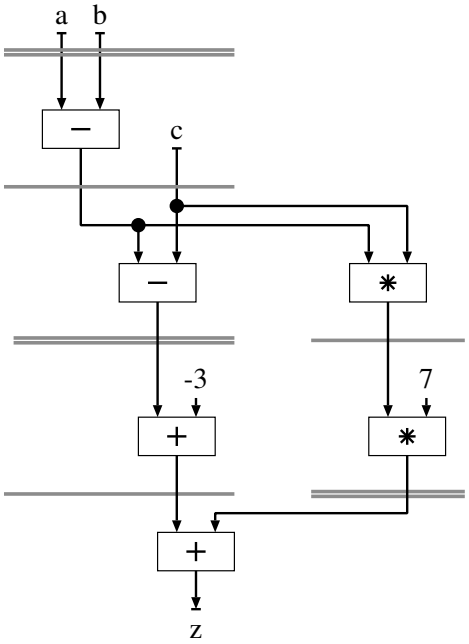
Design a dataflow diagram that satisfies the requirements and maximizes the goals listed above. (There is more space on the next page.)

Answer:

1. *Optimal solution (overlapping stages)*

   (a) *Algebraic optimizations*

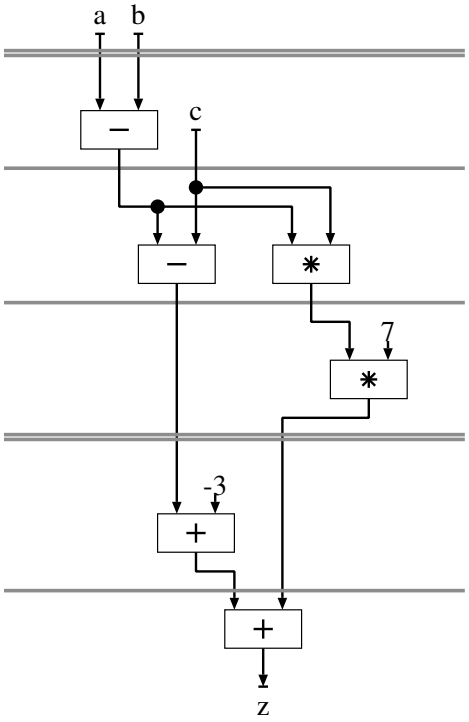
$$\begin{aligned} & a-(b+c)+(a-b)\times c\times 7-3 \\ = \;& (a-b)-c+(-3)+(a-b)\times c\times 7 \end{aligned}$$



marks	*mul*	*add+sub*	*inp*	*reg*	*tput*
15	*1*	*2*	*2*	*4*	*1/2*

inputs	*2 + 0 +*	*0*	*=*	*2*
registers	*2 + 1 +*	*1*	*=*	*4*
multipliers	*0 + 1 +*	*0*	*=*	*1*
adders	*0 + 0 +*	*1*	*=*	*1*
subtracters	*0 + 1 +*	*0*	*=*	*1*
outputs	*0 + 0 +*	*1*	*=*	*1*
Latency				*4*
Throughput	*1/2 1/2*	*1/2*	*=*	*1/2*
Clock period	*0.4 + 0.9 + 3.0 + 16 + 2.2*		*=*	*22.5 ns*
	skew + jitter + clock-to-Q + mult + setup			

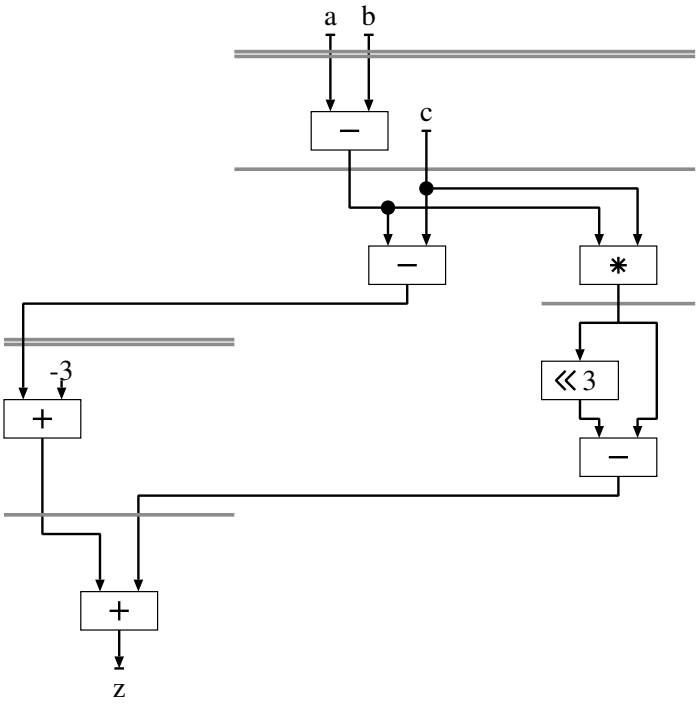
2. Suboptimal solution #1



marks	*mul*	*add+sub*	*inp*	*reg*	*tput*
13	*1*	*2*	*2*	*4*	*1/3*

inputs	*2*
registers	*4*
multipliers	*1*
adders	*1*
subtracters	*1*
outputs	*1*
Latency	*4*
Throughput	*1/3*
Clock period	*0.4 + 0.9 + 3.0 + 16 + 2.2 = 22.5 ns*

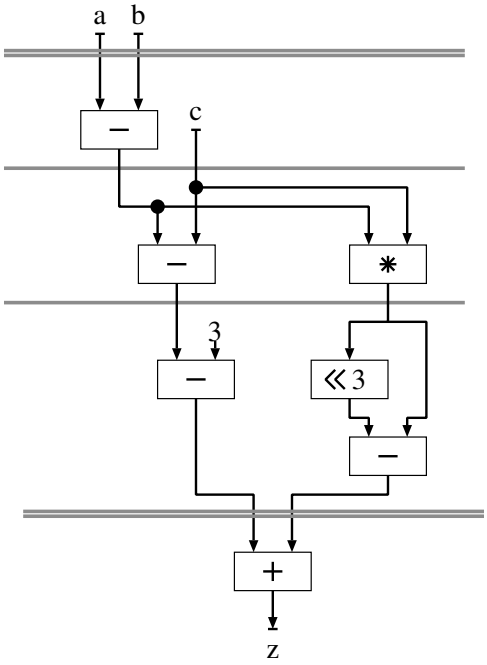
3. Suboptimal solution #2



marks	*mul*	*add+sub*	*inp*	*reg*	*tput*
13	*1*	*2*	*2*	*4*	*1/3*

inputs	*2*
registers	*4*
multipliers	*1*
adders	*1*
subtracters	*1*
outputs	*1*
Latency	*4*
Throughput	*1/3*
Clock period	*0.4 + 0.9 + 3.0 + 16 + 2.2 = 22.5 ns*

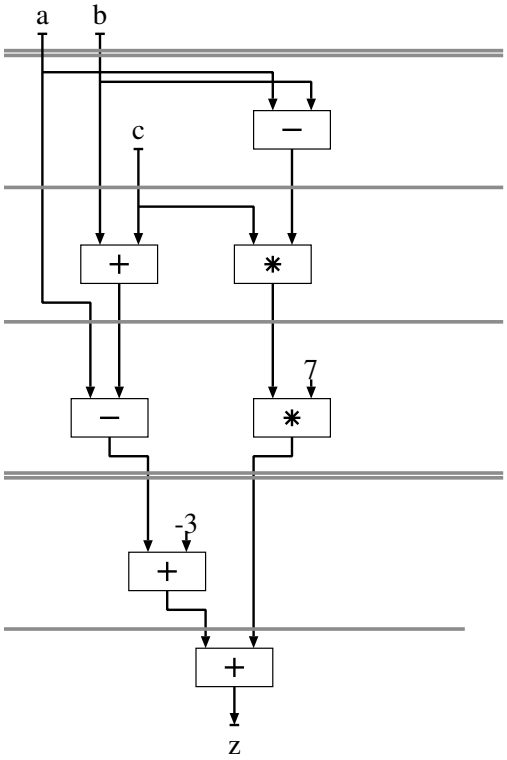
4. Suboptimal solution #3



marks	*mul*	*add+sub*	*inp*	*reg*	*tput*
12	*1*	*3*	*2*	*4*	*1/3*

inputs	*2*
registers	*4*
multipliers	*1*
adders	*1*
subtracters	*2*
outputs	*1*
Latency	*4*
Throughput	*1/3*
Clock period	*0.4 + 0.9 + 3.0 + 16 + 2.2 = 22.5 ns*

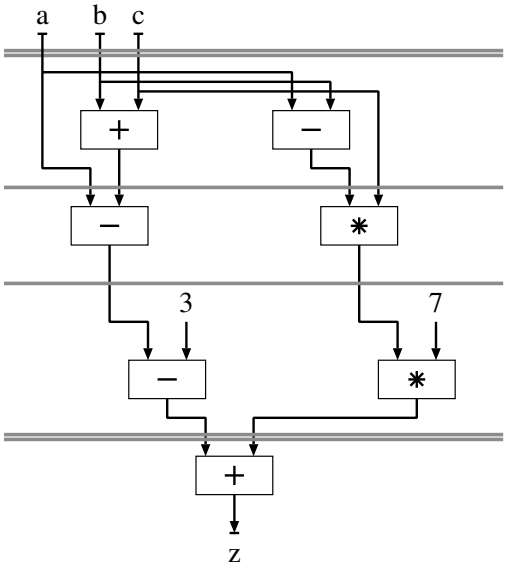
5. Suboptimal solution #4



marks	*mul*	*add+sub*	*inp*	*reg*	*tput*
11	*1*	*3*	*2*	*6*	*1/3*

inputs	*2*
registers	*6*
multipliers	*1*
adders	*2*
subtracters	*1*
outputs	*1*
Latency	*5*
Throughput	*1/3*
Clock period	*0.4 + 0.9 + 3.0 + 16 + 2.2 = 22.5 ns*

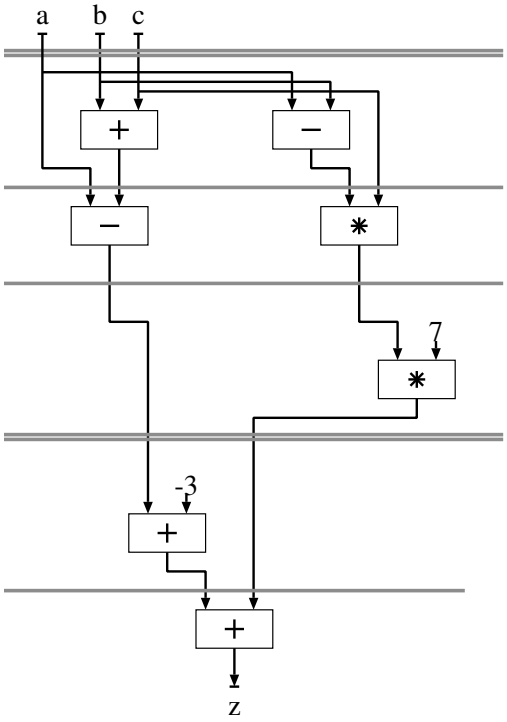
6. Suboptimal solution #5



marks	*mul*	*add+sub*	*inp*	*reg*	*tput*
10	*1*	*3*	*3*	*6*	*1/3*

inputs	*3*
registers	*6*
multipliers	*1*
adders	*2*
subtracters	*1*
outputs	*1*
Latency	*4*
Throughput	*1/3*
Clock period	*0.4 + 0.9 + 3.0 + 16 + 2.2 = 22.5 ns*

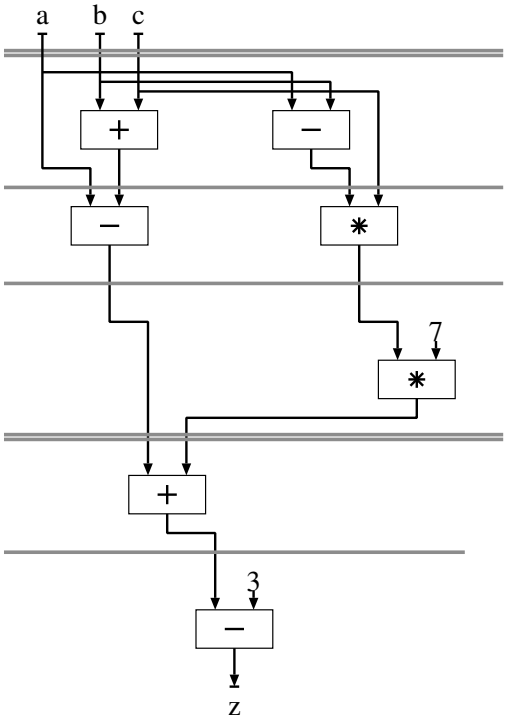
7. Suboptimal solution #6



marks	*mul*	*add+sub*	*inp*	*reg*	*tput*
10	*1*	*3*	*3*	*6*	*1/3*

inputs	*3*
registers	*6*
multipliers	*1*
adders	*2*
subtracters	*1*
outputs	*1*
Latency	*4*
Throughput	*1/3*
Clock period	*0.4 + 0.9 + 3.0 + 16 + 2.2 = 22.5 ns*

8. Suboptimal solution #7



marks	*mul*	*add+sub*	*inp*	*reg*	*tput*
8	*1*	*4*	*3*	*6*	*1/3*

inputs	*3*
registers	*6*
multipliers	*1*
adders	*2*
subtracters	*2*
outputs	*1*
Latency	*4*
Throughput	*1/3*
Clock period	*0.4 + 0.9 + 3.0 + 16 + 2.2 = 22.5 ns*

Marking:

Dataflow diagram

5 marks *syntactically correct*

-1 mark *combinational inputs*

-1 mark *registered outputs*

Analysis

1 mark *area analysis*

1 mark *latency*

1 marks *throughput*

2 marks *clock period*

Q1b (5 Marks) Analysis

Answer:

See above

Q2 (15 Marks) Performance

With the recent discovery of the Higgs-boson-like particle, you have decided that the next hot area in digital hardware will be to develop filters for detecting and analyzing subatomic particles. You have created a startup company, Bozo Filters Inc, and are working on a new boson-detection filter.

One of your engineers, Olivia, has proposed a performance optimization that will provide a performance gain of 15% compared to the current design, but will delay the project by 10 weeks. The project leader, Claire, wants to stick with her current design. Your task is to choose between option O (Olivia's performance optimization) and option C (Claire's current design).

You do *not* know how much the performance of the average boson-detection filter increases each week, because boson-detection filters are such a new type of product.

Because of the uncertainy in the data (amount of performance gain, delay in schedule, rate of performance increase), you decide that you will pursue Olivia's optimization only if it provides a significant advantage over Claire's current design. More precisely, you decide that you will choose Olivia's optimization only if: the ratio of the performance of Olivia's optimization compared to the average boson filter at the time that Olivia's design is done will be 5% higher than the ratio of the performance of Claire's design compared to the average boson filter at the time that Claire's design is done.

What is the maximum increase in average performance per week for boson filters such that you will choose Olivia's optimization over Claire's current design?

NOTES:

1. Some new equations have been added to the page of "potentially useful information", which might be useful in solving this problem.

Answer:

1. Derive equation for rate of performance increase

$$\begin{aligned} P(t_1) &= P(t_1)n^{(t_1-t_0)/k} \\ \frac{P(t_1)}{P(t_0)} &= n^{(t_1-t_0)/k} \\ \frac{P(t_1)}{P(t_0)} &= n^{(t_1-t_0)/k} \\ \left(\frac{P(t_1)}{P(t_0)}\right)^{k/(t_1-t_0)} &= n \end{aligned}$$

2. Initial equations

$$\begin{aligned} \frac{P_o}{P_{avg}(t_o)} &= 1.05\frac{P_c}{P_{avg}(t_c)} \\ t_o - t_c &= 10 \\ P_o &= 1.15P_c \\ n &= \left(\frac{P_{avg}(t_o)}{P_{avg}(t_c)}\right)^{k/(t_o-t_c)} \\ k &= 1 \quad (n = \text{increase in 1 week}) \end{aligned}$$

3. Solve for n

$$\begin{aligned}
\frac{P_{avg}(t_o)}{P_{avg}(t_c)} &= \frac{P_o}{1.05P_c} \\
n &= \left(\frac{P_o}{1.05P_c}\right)^{k/(t_o-t_c)} \\
n &= \left(\frac{1.15P_c}{1.05P_c}\right)^{k/(t_o-t_c)} \\
n &= \left(\frac{1.15}{1.05}\right)^{k/(t_o-t_c)} \\
n &= \left(\frac{1.15}{1.05}\right)^{1/10} \\
n &= 0.009139 \\
&= 0.91\%
\end{aligned}$$

Marking:

4 marks $\frac{P_{avg}(t_o)}{P_{avg}(t_c)} = \frac{P_o}{1.05P_c}$
2 marks $P_{avg}(t_o) = P_{avg}(t_c)n^{(t_o-t_c)/k}$
1 mark $k = 1$
1 mark $t_o - t_c = 10$
2 marks $P_o = 1.15P_c$
2 marks *use performance eqn to solve for rate of performance increase per week*
2 marks *put all equations together and solve for n = 0.91%.*
1 mark *neatnes and clarity*

Q3 (10 Marks) Functional Verification

Your friend, Imran, is a functional verification manager for a secret project. His group is falling behind schedule and he asks you for some advice. His original plan had been to develop assertions, coverage monitors, and reference models for his system, but he now has time for only two of the three tasks. He is considering three options:

Option 1 Assertions and coverage monitors, but no reference models.

Option 2 Assertions and reference models, but no coverage monitors.

Option 3 Coverage monitors and reference models, but no assertions.

Without knowing anything about Imran's system, offer your best advice for how he should choose between the three options.

NOTES:

1. Your advice may be conditional based upon potential characteristics of the system. For example, you may write "If you use Altera, then you should choose option 1, because Google has 6.7 million hits for 'altera coverage monitors' but only 0.6 million hits for 'altera reference model' ".

Answer:

Imran should definitely not *choose option 2, which skips the coverage monitors. Coverage monitors provide the best indication of whether you have done enough verification (run a sufficient number of tests).*

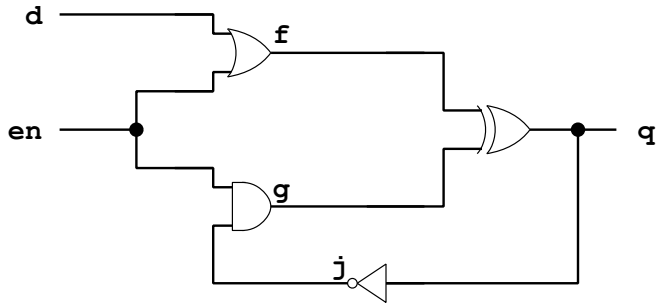
The choice between options 1 and 3 depends largely on the type of system that Imran is building. If the system is datapath-centric, then option 3 is best (use reference models), because reference models are well suited to datapath circuits. If the system is control-centric, then optoin 2 is best (use assertions), because reference models are pooly suited to control circuits and state machines.

Marking:

2.5 marks *Need coverage monitors*
2.5 marks *Reference models are good for datapath*
2.5 marks *Assertions are good for control circuitry and/or state machines*
2.5 marks *Logical reasoning and final conclusions*

Q4 (10 Marks) Latch Analysis

In this question, you will analyze the circuit below to determine if it is correct implementation of a latch.



NOTES:
1. The delay through each gate is 1 ns.

Q4a (3 Marks) Good or Bad?

Answer:
Yes, the circuit is a correct implementation of a latch.

Marking:

3 marks *correct answer*
1 mark *incorrect answer (if rest of question is answered)*

Q4b (7 Marks) Analysis

This part of the question is divided into two columns. Use the left column if you answered *yes* above. Use the right column if you answered *no* above.

Determine if the latch is active-high or active-low and calculate the timing parameters below.

Answer:

Active-high or active-low?	*Active low*	*1 mark*
Clock-to-Q	*2*	*2 marks*
Setup	*3*	*2 marks*
Hold	*0*	*2 marks*

$$\begin{aligned} \text{Clock} - \text{to} - \text{Q} &= \text{delay}(\texttt{d} \rightarrow \texttt{q}) \\ &= 2 \\ \text{Setup} &= \text{delay}(\texttt{d} \rightarrow \texttt{g}) - \text{delay}(\texttt{en} \rightarrow \texttt{g}) \\ &= 4 - 1 \\ &= 3 \\ \text{Hold} &= \text{delay}(\texttt{en} \rightarrow \texttt{f}) - \text{delay}(\texttt{d} \rightarrow \texttt{f}) \\ &= 1 - 1 \\ &= 0 \end{aligned}$$

Marking:

latch is correct

1 mark *answer is* ± 1 *the correct answer*

latch is incorrect

7 marks *glitch on transition from load to store caused by delay from d to q being exactly the same as delay from en to q*

5 marks *problem on transition from load to store caused by improper choice of gate delays*

4 marks *glitch on transition from load to store*

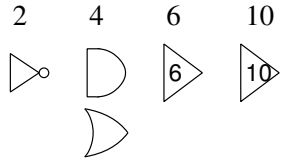
2 marks *glitch on transiton from store to load, or no mention of which mode transition has the glitch*

Q5 (20 Marks) Critical Path

For the circuit below, the longest path $\langle \mathtt{c}, \mathtt{g}, \mathtt{j}, \mathtt{m}, \mathtt{n}, \mathtt{q}, \mathtt{r}, \mathtt{s}, \mathtt{t} \rangle$ and the second-longest path $\langle \mathtt{c}, \mathtt{g}, \mathtt{j}, \mathtt{k}, \mathtt{p}, \mathtt{s}, \mathtt{t} \rangle$ are both false paths.

NOTES:

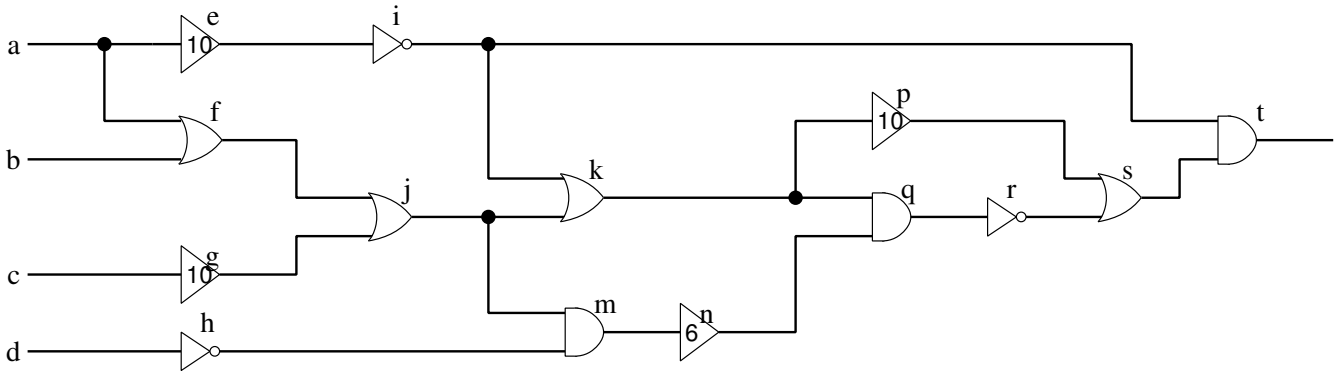
1. A *buffer* is drawn as a triangle with the delay through buffer written inside the triangle. The output of the buffer is simply a delayed version of the input waveform.
2. The delays through the gates are:



3. The equations for some steady-state (static) conditions:

e	$=$	a	$\overline{e}$	$=$	$\overline{a}$
f	$=$	$a+b$	$\overline{f}$	$=$	$\overline{a}\overline{b}$
g	$=$	c	$\overline{g}$	$=$	$\overline{c}$
h	$=$	$\overline{d}$	$\overline{h}$	$=$	d
i	$=$	$\overline{e}$	$\overline{i}$	$=$	a
	$=$	$\overline{a}$			
j	$=$	$f+g$	$\overline{j}$	$=$	$\overline{a}\overline{b}\overline{c}$
	$=$	$a+b+c$			
k	$=$	$i+j$	$\overline{k}$	$=$	False
	$=$	$\overline{a}+a+b+c$			
	$=$	True			

m	$=$	jh	$\overline{m}$	$=$	$\overline{a}\overline{b}\overline{c}+d$
	$=$	$(a+b+c)\overline{d}$			
n	$=$	m	$\overline{n}$	$=$	$\overline{a}\overline{b}\overline{c}+d$
	$=$	$(a+b+c)\overline{d}$			
p	$=$	k	$\overline{p}$	$=$	False
	$=$	True			
q	$=$	kn	$\overline{q}$	$=$	$\overline{a}\overline{b}\overline{c}+d$
	$=$	$(a+b+c)\overline{d}$			
r	$=$	$\overline{q}$	$\overline{r}$	$=$	$(a+b+c)\overline{d}$
	$=$	$\overline{a}\overline{b}\overline{c}+d$			
s	$=$	$p+r$	$\overline{s}$	$=$	False
	$=$	True			
t	$=$	is	$\overline{t}$	$=$	a
	$=$	$\overline{a}$			



Q5a (15 Marks) Third-Longest Path

Find the third-longest path and determine if it is a viable path or a false path.

If the third-longest path is *viable*, give a set of values on the primary inputs that will exercise the path.

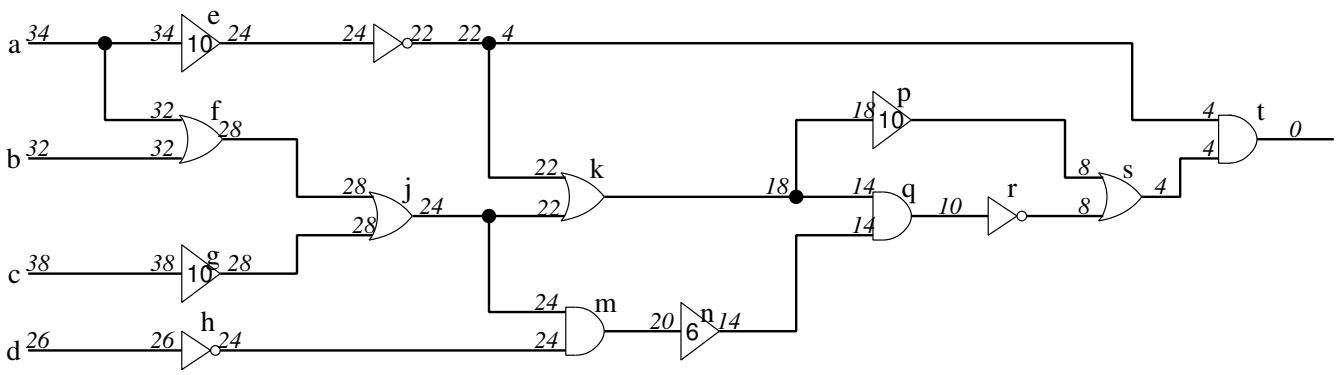
If the third-longest path is a *false* path, explain why.

NOTES:

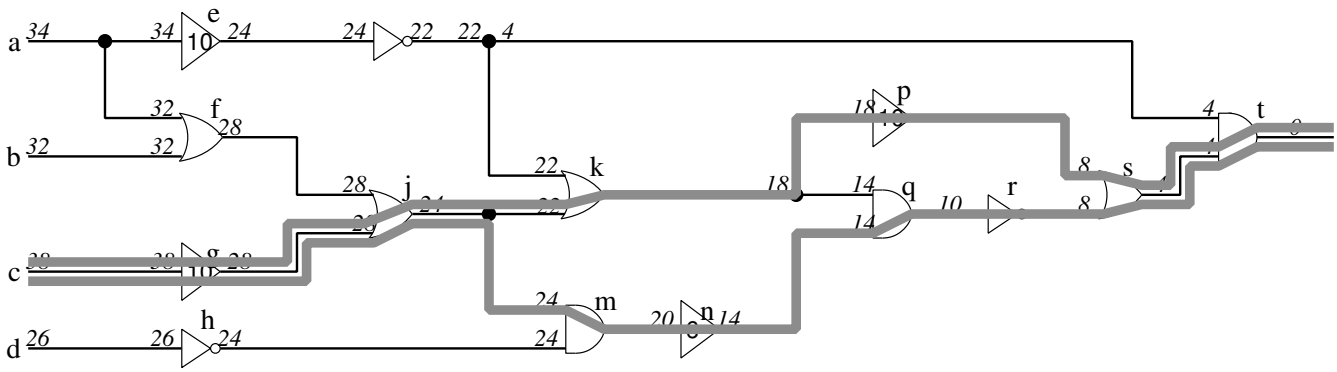
1. If there are multiple third-longest paths with the same delay, choose the path that is alphabetically earlier (e.g., if the third-longest paths are $\langle \mathtt{q}, \mathtt{r}, \mathtt{s}, ... \rangle$ and $\langle \mathtt{q}, \mathtt{r}, \mathtt{u}, ... \rangle$, choose $\langle \mathtt{q}, \mathtt{r}, \mathtt{s}, ... \rangle$)
2. The answer shall be written on the next page.

Answer:

1. annotate with delays



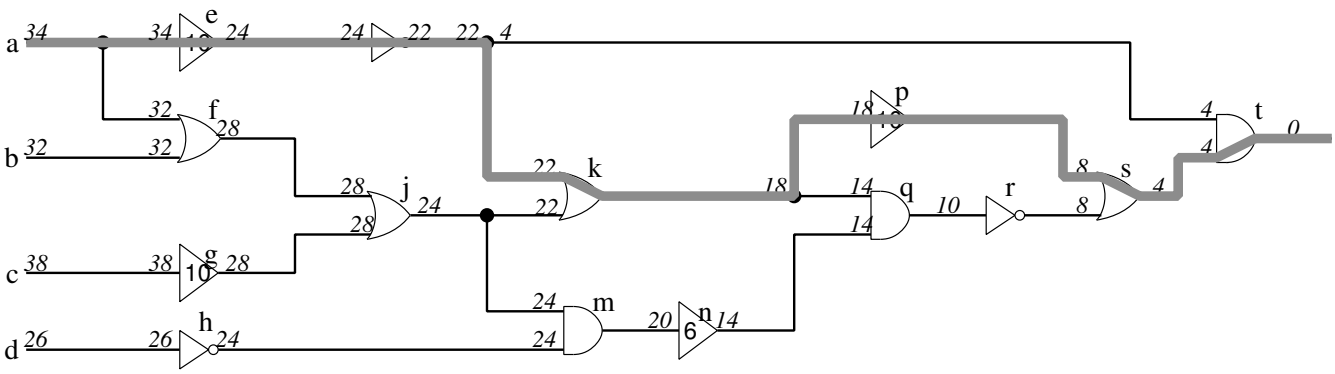
2. previously discovered false paths



3. find third-longest path

Longest path has delay 38. Second-longest path has delay 36. No other path has delay 36. Path from a has delay 34, so follow it.

Third longest path is $\langle \mathtt{a}, \mathtt{e}, \mathtt{i}, \mathtt{k}, \mathtt{p}, \mathtt{s}, \mathtt{t} \rangle$



4. *constraint table for non-controlling side inputs*

   k[j] *0* $\overline{j}$ $\overline{a}\overline{b}\overline{c}$

   s[r] *0* $\overline{r}$ *(a+b+c)*$\overline{d}$

   t[i] *1* $\overline{i}$ $\overline{a}$

   Contradiction between k[j] and s[r].

   k[j] and s[r] are both on previously discovered false paths.

   Try late-arriving side-input for k[j]. We choose k[j] because it is closer to the inputs and so will be simpler to analyze.

5. *late-arriving k[j]*

   (a) viable($\langle \mathsf{c}, \mathsf{g}, \mathsf{j} \rangle$)

$$\begin{aligned} \text{viable}(\langle \mathsf{c}, \mathsf{g}, \mathsf{j} \rangle) &= \text{nonctrl}(j[f]) \\ &= \overline{f} \\ &= \overline{a}\overline{b} \end{aligned}$$

   (b) ctrl(k[i])

$$\begin{aligned} \text{ctrl}(k[i]) &= i \\ &= \overline{a} \end{aligned}$$

6. *update constraint table*

   k[j] *0* $\overline{j}$ $\overline{a}\overline{b}\overline{c} + \text{viable}(\langle \mathsf{c}, \mathsf{g}, \mathsf{j} \rangle)\text{ctrl}(k[i])$

$$\begin{aligned} &= \overline{a}\overline{b}\overline{c} + (\overline{a}\overline{b})(\overline{a}) \\ &= \overline{a}\overline{b}\overline{c} + \overline{a}\overline{b} \\ &= \overline{a}\overline{b} \end{aligned}$$

   s[r] *0* $\overline{r}$ $(a+b+c)\overline{d}$

   t[i] *1* $\overline{i}$ $\overline{a}$

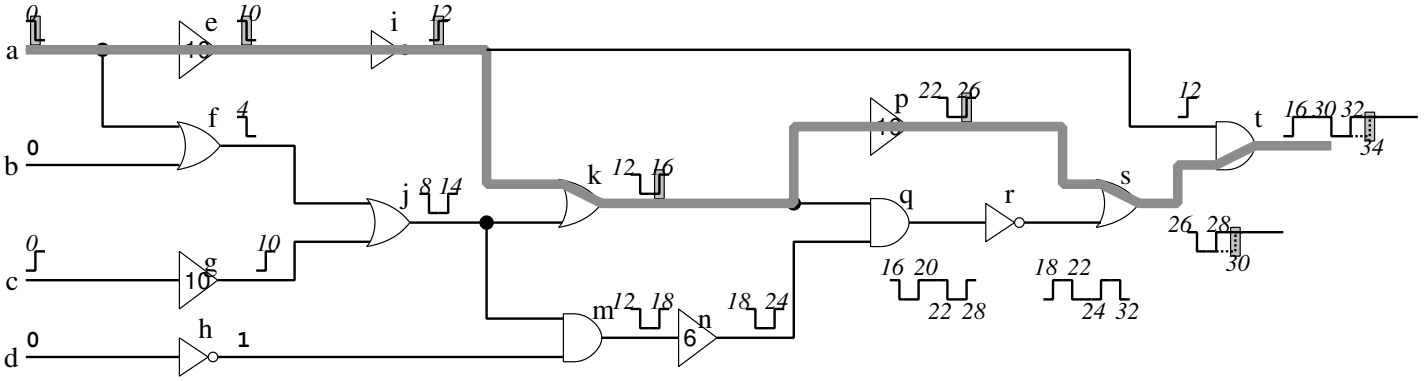
7. *top-level constraint*

$$\begin{aligned} & (\overline{a}\overline{b})((a+b+c)\overline{d})(\overline{a}) \\ = \; & \overline{a}\overline{b}c\overline{d} \end{aligned}$$

8. *excitation*

a	b	c	d
⌞	0	⌜	0

9. *Exercise critical path with excitation:*
(This analysis is just for illustration. It is not necessary for this question.)



We see the effect of monotone speedup at s*, where the edge on the critical path does* not *propagate to* s *with the current delays. But, if the last falling edge on* r *arrived before the last rising edge on* p*, then the last rising edge on* p *would propagate to* s *and on to the output.*

Marking:

2 marks *correct path*
1 marks *viable="yes"*

If deriviation is clear and understandable, or said "false" and chosen path intersects a previous false path

2 marks *non-controlling values on side inputs*
1 mark *propagate constraints to inputs*
1 mark *conjunction of constraints*
1 mark *Boolean algebra, check for contradiction*
1 mark *try late-arriving side input*
2 marks *late side path is viable*
2 marks *controlling value on path input*
2 marks *translate constraint into excitation*

If deriviation is clear and understandable, or said "false" and chosen path does not intersect a previous false path]

3 marks *non-controlling values on side inputs*
3 marks *propagate constraints to inputs*
2 marks *conjunction of constraints*
2 marks *Boolean algebra, check for contradiction*
2 marks *translate constraint into excitation*

If derivation is unclear

3 marks *a=falling edge*
3 marks *b=0*
3 marks *c=rising edge (1 mark for '1')*
3 marks *d=0*

Q5b (5 Marks) Monotone Speedup

Based upon the three longest paths through the circuit, does this circuit illustrate the benefits of taking into account monotone speedup when determining the critical path? **For full marks, you must justify your answer.**

If you are unable to determine whether this circuit illustrates monotone speedup by analyzing the three longest paths, then explain the concept of monotone speedup and why it is beneficial to take into account monotone speedup when finding the critical path through a circuit.

Answer:

The three paths do *illustrate monotone speedup.*

(This analysis is more complicated that would be required.)

The monotone speedup rules are for when we have a late-arriving side input and the side-input is a non-controlling value. The two locations for a possible late arriving side input are `k[j]` *and* `s[r]`*. We already know that the late-side path to* `k[j]` *is viable, because we setup the constraints to make it viable. The viability condition for the late side path to* `s[r]` *is:*

$$\begin{aligned} \text{viable}(\langle \mathtt{c}, \mathtt{g}, \mathtt{j}, \mathtt{m}, \mathtt{n}, \mathtt{q}, \mathtt{r} \rangle) &= (\text{nonctrl}(\mathtt{m[h]}))(\text{nonctrl}(\mathtt{q[k]})) \\ &= (h)(k) \\ &= \overline{d} \end{aligned}$$

Both late-arriving side paths are viable.

For the excitation, the value of `j` *(for the* late side *input* `k[j]`*) is: 1, which is a* controlling value *for* `k`*, so* `k[j]` does not *illustrate monotone speedup.*

For the excitation, the value of `r` *(for the* late side *input* `s[r]`*) is: 0, which is a* non-controlling value *for* `s`*, so* `s[r]` does *illustrate monotone speedup.*

In conclusion, the problem does illustrate monotone speedup, *because with the excitation needed for the critical path, an edge does not propagate along the critical path, but if a late-arriving side-input path is sped up, then an edge will propagate along the critical path.*

Marking:

"Yes"

- **5 marks** *correct justification*
- **3 marks** *identifies that s[r] is a late arriving side input*
- **2 marks** *identifies that k[j] is a late arriving side input*
- **1 mark** *some correct information*

"No"

5 marks *no late arriving side inputs (if chosen path does not have any late-arriving side inputs)*

5 marks *has at least one late arriving side input; but have controlling values on all side inputs, or path input causes edge on output of gate.*

"Unable"

1 mark *clock period must stay the same or decrease if part of the circuit becomes faster*

1 mark *makes a path that appears to be false become viable*

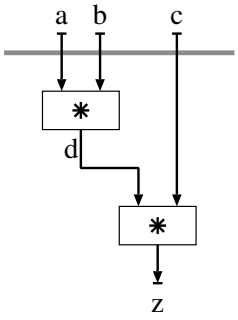
1 mark *late side input has a non-controlling value*

1 mark *speed up a previously discovered false path to make the edge come earlier than the edge on the path input*

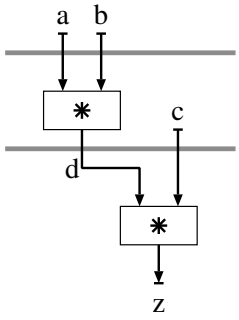
Q6 (15 Marks) Power and Energy

In this question, you will compare the power and energy consumption of three different circuits that implement the equation $z = a \times b \times c$.

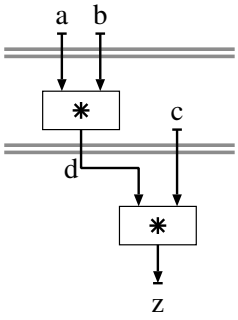
Circuit #1



Circuit #2



Circuit #3



NOTES:

1. All three circuits use the same type of FPGA chip.
2. Each of three circuits is run at its maximum clock speed and maximum throughput.
3. The areas and delays of input ports, output ports, registers, and multiplexers are much less than the area of a multiplier.

Q6a (10 Marks) Power Comparison

Compare the *power consumption* of the circuits by writing one of "<", "=", or ">" in each box below. **For full marks, you must justify your answer.**

Answer:

We need to look at activity factor, capacitance (area), and clock frequency. We assume that all multipliers have the same activity factor.

D delay through mult
A activity factor for a mult
C capacitance of a mult
L leakage power of a mult

We will lump short-circuiting and switching power together, because they are both linearly dependent on area, activity factor, and clock frequency.

In analyzing dynamic power, we will use just D, A, and C, because we are interested only in the relative *amounts of power and all other factors are constant across the three circuits.*

Clock freq	$1/2D$	$1/D$	$1/D$
Area	$2C$	C	$2C$
Dynamic power	$\frac{1}{2D}(2C) = \frac{C}{D}$	$\frac{1}{D}(C) = \frac{C}{D}$	$\frac{1}{D}(2C) = \frac{2C}{D}$
Static power	$2L$	L	$2L$

	$Circuit\#1$		$Circuit\#2$		$Circuit\#3$
Relative dynamic power		$=$		$<$	
Relative static power		$>$		$<$	
Relative total power		$>$		$<$	

Marking:

-2 marks *missing static power*
-4 marks *missing clock frequency*
-1 mark *illogical reasoning*

Q6b (5 Marks) Energy Comparison

Compare the *energy consumption per parcel* of circuits 2 and 3 by writing one of "<", "=", or ">" in the box below. **For full marks, you must justify your answer.**

Answer:

For both circuits, a parcel travels through two multipliers and consumes the same amount of energy in each multiplier. Thus, the energy related to dynamic power is the same for both circuits. However, the energy related to static power is higher for circuit 2, because it has more leakage power than circuit 1.

Marking:

1 mark *static power*
2 marks *observations*
2 marks *reasoning based on observations*

Q7 (10 Marks) Clock Gating

You are designing a clock-gating scheme for a Kirsch edge detector.

NOTES:

1. For adjacent pixels on the *same row* of the image, there are exactly 3 bubbles between valid pixels.
2. At the *end of a row*, there are exactly 100 bubbles before the first valid pixel of the next row.
3. At the *end of an image*, there are exactly 100 bubbles before the first valid pixel of the next image.
4. The latency is 8 clock cycles.
5. The image size is 64×64 pixels.

What is the absolute *minimum* power consumption that you could achieve with clock-gating, as a percentage of the edge-detector's original power?

If you do not know how to answer the question as stated, you may earn part-marks (maximum of 7) by assuming that there are *no bubbles between valid pixels in the same row* (*i.e.*, the system is fully pipelined) and placing a $\surd$ in the box. ☐

List any assumptions that you make (*e.g.*, "frictionless surfaces and point masses"):

Answer:

- *Leakage power is negligible*
- *Power consumption of clock-gating circuit is negligible*
- *Clock-gating scheme is 100% effective*

Analysis:

Answer:

$$\begin{aligned} \textit{time from first i\_valid to last i\_valid} &= 63\times 4 + 1 \\ &= 253 \\ \textit{latency} &= 8 \\ \textit{NumClkEn} &= 253 + 8 \\ &= 261 \\ \textit{Window} &= 253 + 100 \\ &= 353 \\ \textit{PctClk} &= 261/353 \\ &= 73.9\% \end{aligned}$$

Marking:

2 marks *assumptions*
2 marks *window size*
2 marks *time from first i_valid to last i_valid*
2 marks *latency in NumClkEn*
2 marks *putting all of the pieces together*