ECE 327 Solution to Final

2014t2 (Spring)

		Total Marks	Approx. Time	Page
Q1	Dataflow Diagram	20	25	2
Q2	Allocation and Control	20	30	5
Q3	Latch Design and Analysis	15	20	10
Q4	Elmore	15	20	13
Q5	Power and Performance	15	20	16
Q6	Clock Gating	15	20	18
Totals		100	135	

Q1 (20 Marks) Dataflow Diagram

(*estimated time: 25 minutes*)

In this question, you will design and analyze a dataflow diagram for the equation:

$$a+b+c+10-3\times a\times b-3\times b\times c$$

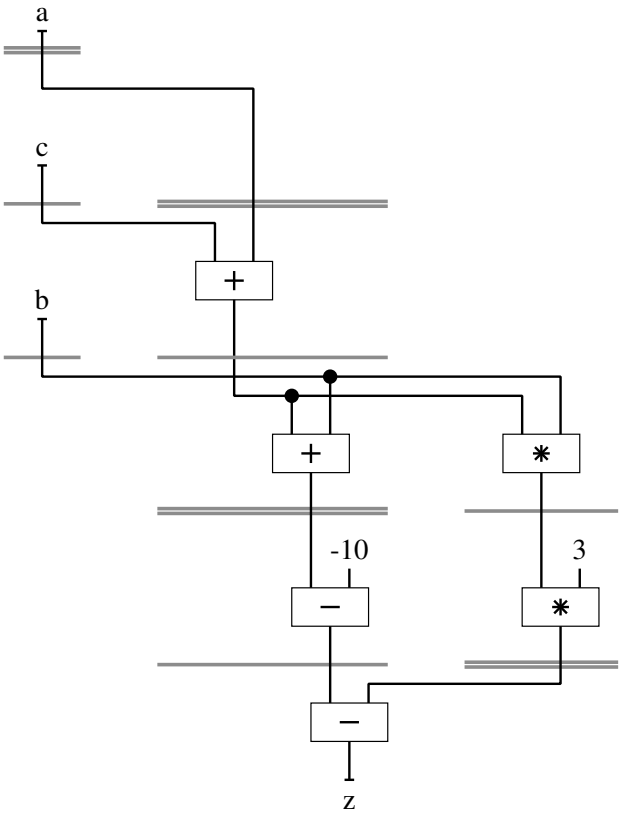
NOTES:

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, from highest priority to lowest:
      • Multipliers
      • 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.

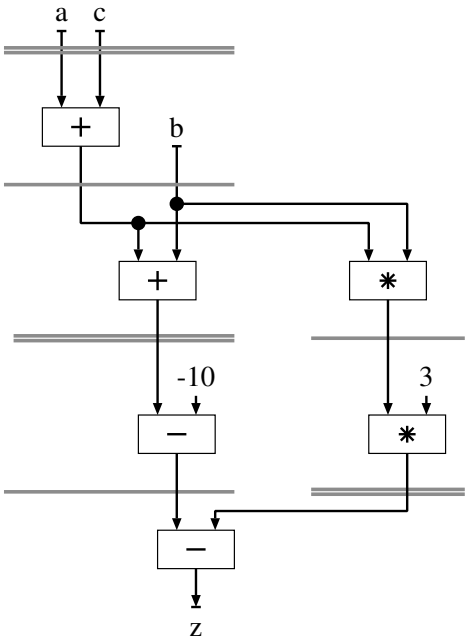
On the next page, draw a dataflow diagram that satisfies the requirements and is optimized according to the goals above.

Answer:

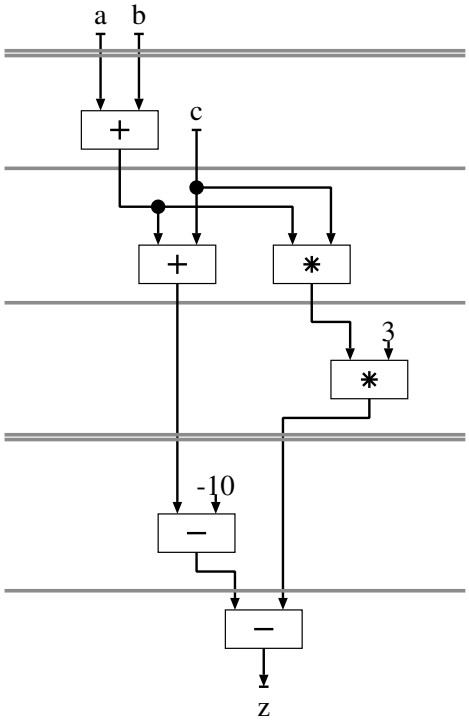
Optimal answer



Suboptimal answer #1



Suboptimal answer #2



Marking:

-5 marks *violates throughput requirement*

-2 marks *signal skips clock-cycle boundary*

-1 mark *registered output*

Some example marks:

Clk	Mul	Add+Sub	Inp	Reg	Tput	Mark
1	1	2	1	4	1/3	20
1	1	2	2	4	1/2	18
1	1	2	2	4	1/3	17
2	1	3	2	7	1/2	12
2	1	3	3	4	1/3	10

Q2 (20 Marks) Allocation and Control

(*estimated time: 30 minutes*)

This question examines the implementation of a dataflow diagram.

Q2a: perform allocation and draw a control table

Q2b: choose a state encoding

Q2c: write VHDL code for the datapath registers

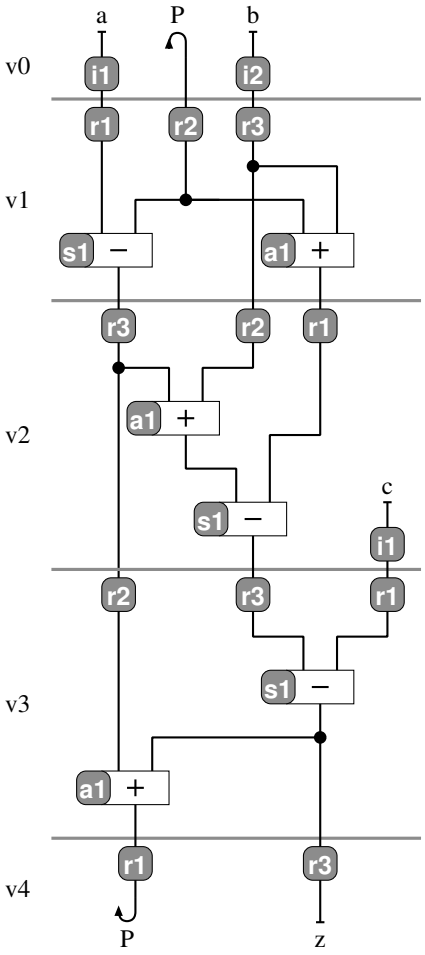
Q2a (10 Marks) Allocation and Control Table

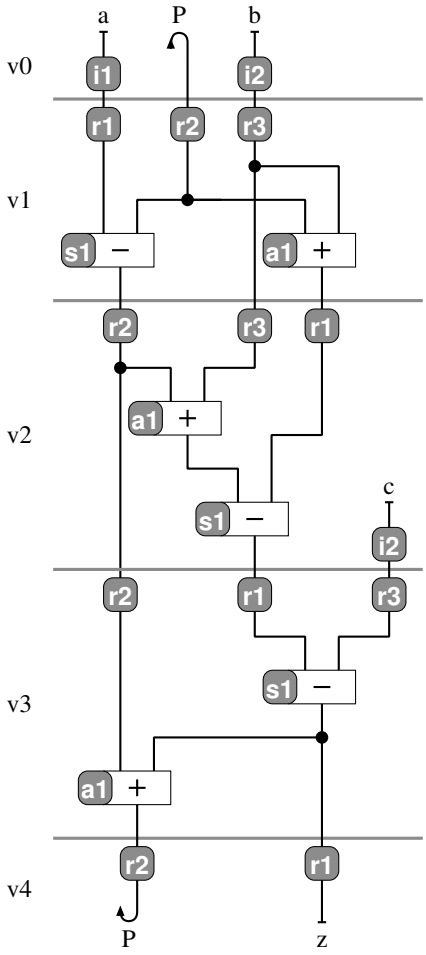
For the dataflow diagram on the next page: perform input/output allocation, datapath allocation, and register allocation; then draw the control table.

NOTES:

1. You may use 2:1 multiplexers, and may combine 2:1 multiplexers to create larger multiplexers.
2. The optimization goals, in order of highest priority to lowest, are to minimize the number of:
   (a) Input ports
   (b) Output ports
   (c) Registers
   (d) Adders
   (e) Subtracters
   (f) 2:1 multiplexers and chip enables
3. You shall *not* perform scheduling optimizations on the dataflow diagram.
4. The *only* algebraic optimization that you may perform is *commutativity*.
5. You shall *not* perform don't-care instantiation

Answer:





	r1 ce	r1 d	r2 ce	r2 d	r3 ce	r3 d	a1 src1	a1 src2	s1 src1	s1 src2	o1	
const			1		1						r1	
idle	0	—	—	—	—	—	—	—	—	—	—	
v0	1	i1	1	r1	1	i2						
v1	1	a1	1	r3	1	s1	r2	r3	r1	r2	—	
v2	1	i1	1	r3	1	s1	r2	r3	a1	r1	—	
v3	1	a1	—	—	1	s1	r2	s1	r3	r1	—	totals
v4											r1	
ce	1											1
mux		1		1		1		1	2	1		7
								total cost				8

	r1 ce	r1 d	r2 ce	r2 d	r3 ce	r3 d	a1 src1	a1 src2	s1 src1	s1 src2	o1	
const	1					i2	r2					
idle	—	—	0	—	—	—	—	—	—	—	—	
v0	1	i1	0	—	1	i2						
v1	1	a1	1	s1	0	—	r2	r3	r1	r2	—	
v2	1	s1	0	—	1	i2	r2	r3	a1	r1	—	
v3	1	s1	1	a1	—	—	r2	s1	r1	r3	—	
v4											r1	totals
ce			1		1							2
mux		2		1				1	1	2		6
										total cost		9

Marking:

10 marks *8 mux+ce*
8 marks *9 mux+ce*
7 marks *10 mux+ce*
6 marks *11 mux+ce*
5 marks *12 or more mux+ce*
-1 mark *1 mistake*
-2 marks *2–3 mistakes*
-3 marks *4–5 mistakes*
-4 marks *6–8 mistakes*
-5 marks *9–12 mistakes*
-6 marks *13 or more mistakes*

Mistakes

2 *extra register*
2 *incorrect number of clock cycles*
2 *missing idle state*
2 *missing chip-enables, or used ce on add or sub*
2 *missing don't-cares*
2 *missing unuseds*
2 *missing reg for P, or allocated input or output port for P*
2 *commutativity on sub*
1 *missing ce=0 on reg for P*
1 *missing allocation for output port*
1 *mismatch between DFD and ctrl table*

Q2b (3 Marks) State Encoding

What is the best choice for the state encoding for your system, and how many bits will you need for your state signal(s)?

NOTES:

1. The system is required to support an indeterminate number of bubbles.

Answer:

Valid bits, because the system has an inter-parcel variable.

There are four clock cycles in the dataflow diagram, so four valid bits are needed for the state signal.

The state signal will be named `v(0 to 3)`.

Marking:

3 marks *valid bits*
2 marks *one-hot*
1 mark *binary*
0.5 marks *grey*
-1 mark *incorrect number of bits*

Q2c (7 Marks) VHDL Coding

Write the VHDL code for each datapath register in your design.

NOTES:

1. Your VHDL code may use your registers (`r1`...`r`*n*), inputs (`i1`...`i`*n*), adders (`a1`...`a`*n*), subtracters (`s1`...`s`*n*), and state signal(s).
2. Use don't-care instantiation and peephole optimizations to simplify the control circuitry.
3. The primary marking criteria are the functional correctness and simplicity of your control circuitry, not the details of VHDL syntax.

Answer:

```
process begin
  wait until rising_edge(clk);
  if v(0) = '1' or v(2) = '1' then
    r1 <= i1;
  elsif v(1) = '1' or v(3) = '1' then
    r1 <= a1;
  end if;
end process;

process begin
  wait until rising_edge(clk);
  if v(0) = '1' then
    r2 <= r1;
    r3 <= i2;
  else
    r2 <= r3;
    r3 <= s1;
  end if;
end process;
```

Marking:

+2 marks *correct coding structure for registered asns*

+2 marks *code for data inputs to all regs*

+2 marks *code for chip-enable to regs that need ce*

+1 marks *used encoding to simplify circuitry*

-1 mark *extraneous code*

-1 mark *each mistake*

Q3 (15 Marks) Latch Design and Analysis

(*estimated time: 20 minutes*)

You were recently hired by Juju Circuits of Arendelle. Your manager, Anna, was called away on a family emergency and has left you with the job of finishing a latch design for a customer named Hans.

Your task is:

Q3a Add gates and wires to the interiors of *box1*, *box2*, and *box3* in the diagram below to complete the design of the required latch.

Q3b Analyze the timing parameters of the latch, or explain why the circuit cannot be used to construct a latch.

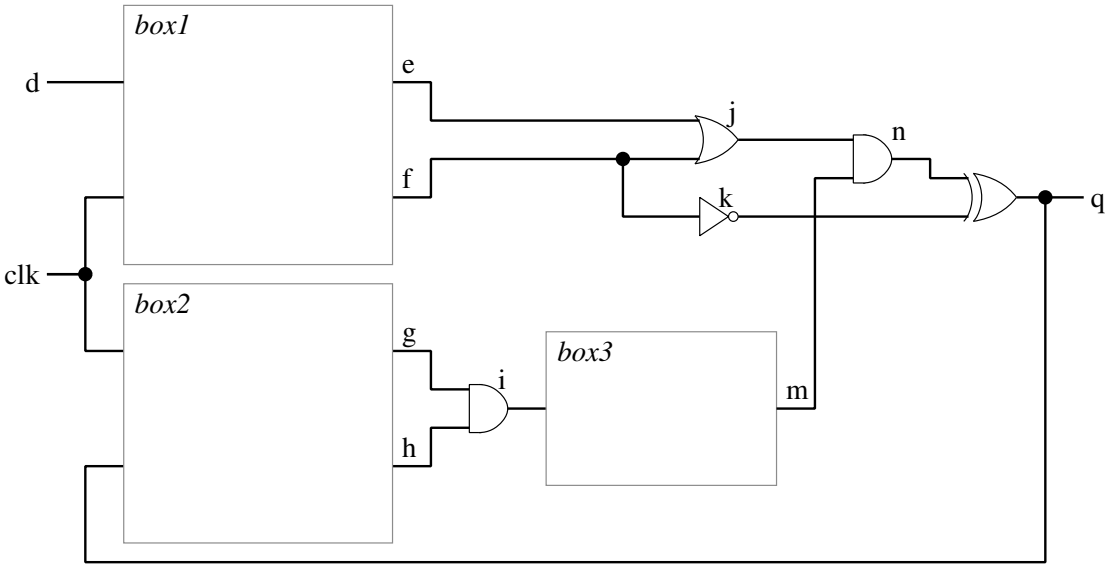
NOTES:

1. The latch shall be *active low*.
2. The existing gates and wires in the circuit shall *not* be changed.
3. The only connections between the gates that you add inside the boxes and the rest of the circuit shall be the signals:

	Inputs	Outputs
box1	`d`, `clk`	`e`, `f`
box2	`clk`, `q`	`g`, `h`
box3	`i`	`m`

4. Optimization goals, from highest priority to lowest:
   - Minimize clock-to-Q
   - Minimize area
5. The delay through each gate is 1 ns.

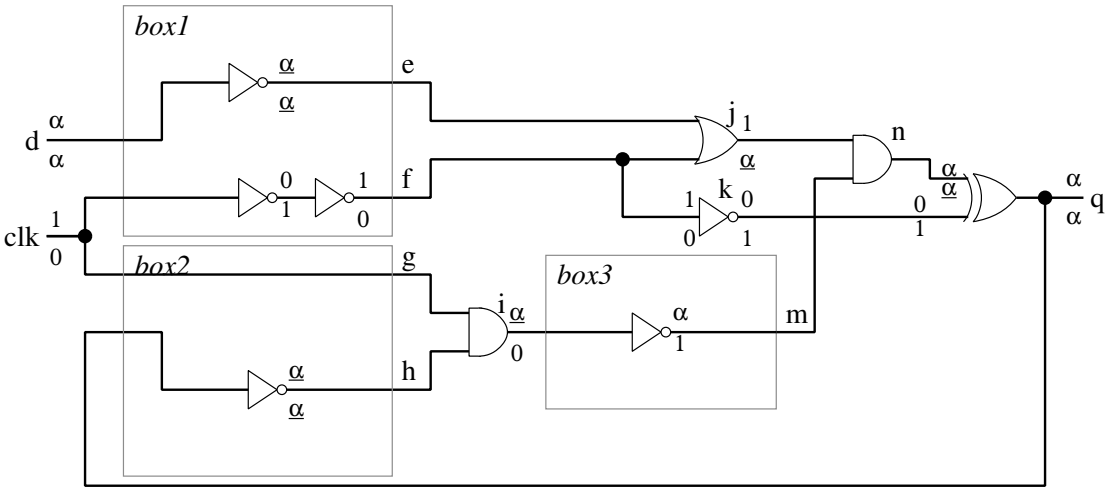
Q3a Latch Design



Answer:

The circuit below shows values above and below each signal line. The value above corresponds to when clk=1 and the value below corresponds to when clk=0.

The XOR *gate acts like an inverter when clk=0 and a wire when clk=1.*



Clock to Q	*delay($clk \rightarrow j \rightarrow q$)*	*= 5*
Setup	*delay($d \rightarrow i$) – delay($clk \rightarrow i$)*	*= 5*
Hold	*delay($clk \rightarrow j$) – delay($d \rightarrow j$)*	*= 1*

Q3b Latch Analysis or Explanation

If you were able to design a correctly functioning active-low latch, then calculate the timing parameters, otherwise, explain why the circuit cannot be used to construct an active-low latch.

Answer: see above

If made a working latch:

Latch design

Marking:

- **+2 marks** *even number of inversions around storage loop when in store mode*
- **+2 marks** *even number of inversions on load path when in load mode*
- **+2 marks** *store mode and load mode are mutually exclusive*
- **+2 marks** *delay of clk $\rightarrow$ store-enable $\rightarrow$ join is less than delay of clk $\rightarrow$ load-enable $\rightarrow$ join*
- **+3 marks** *clock-to-Q delay (-1 if latch is incorrect)*
  - **3 marks** *5*
  - **2 marks** *6*
  - **1 marks** *7 or more*

Latch analysis:

Marking:

4 marks *all correct*

3 marks *one answer off by 1*

2 marks *one mistake or two answers off by 1*

1 mark *some correct information*

$\frac{1}{2}$ *at least two parts answered*

If unable to make a working latch, marking was based upon a combination of the attempt to create a working latch and the explanation for why the attempt was unsuccesful.

Q4 (15 Marks) Elmore

(*estimated time: 20 minutes*)

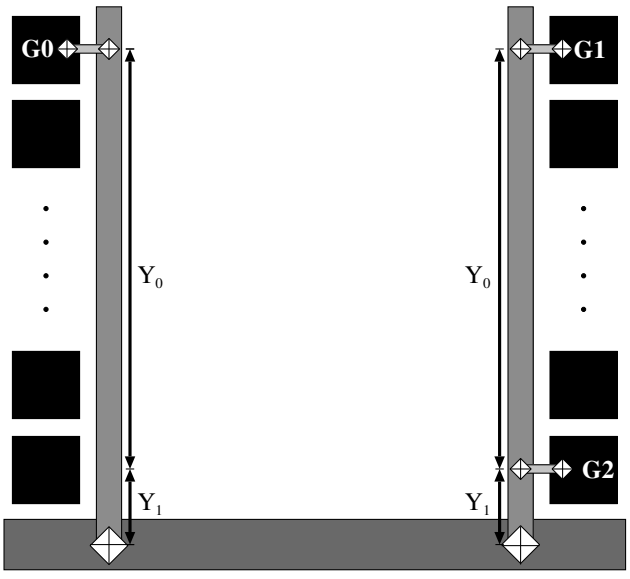
You were recently hired by Fred and George Weasley to develop a circuit for a new autopilot system for a high-speed personal aircraft. A high clock speed is very important to provide sufficient stability and performance.

Your task is to analyze the delay *from G0 to G1* for the two layouts below. If one layout has a smaller delay, then choose that layout, otherwise find an equation for the ratio $Y0/Y1$ in terms of C_L, C_Y, and C_Z such that the two layouts have the same delay.

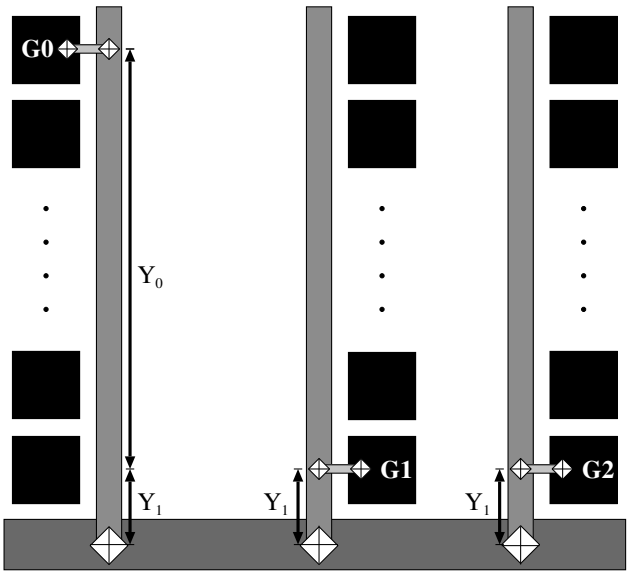
NOTES:

1. G0 is the source gate. G1 and G2 are the load gates.
2. A "switchbox" is equivalent to an "antifuse" or "via".
3. The capacitance of a node on a wire is independent of the location of the node on the wire.
4. The resistance of level-1 wires is negligible, because they are extremely short.
5. The resistance between nodes on a level-2 wire is proportional to the distance between the nodes.
6. The resistance of level-3 wires is negligible, because the cross-section of the wire is so large.
7. **For full marks, you must justify your answer.**

Symbol	Description	Capacitance	Resistance
	Gate	C_L	0
	Switchbox	0	0
	Level 1 wire	0	0
	Level 2 wire	C_Y	proportional to distance
	Level 3 wire	C_Z	0



Layout-1



Layout-2

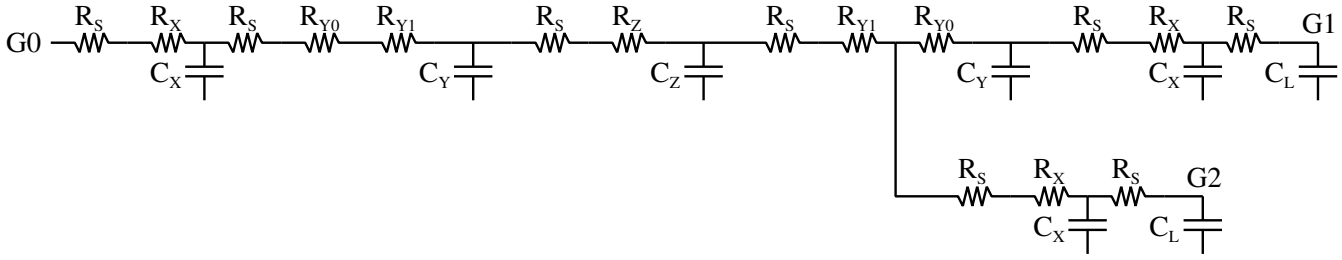
Space for your justification and answer are on the next page

Answer:

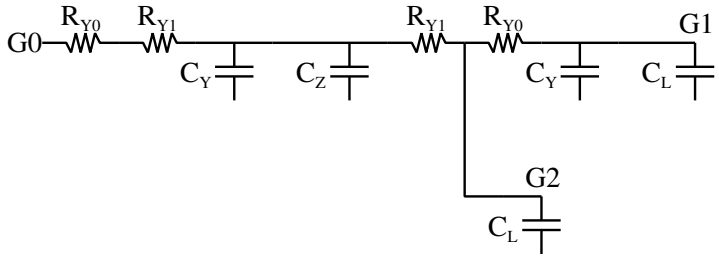
1. *Layout-1 showing all resistances and capacitances, even those that are negligible. (This is not needed for the exam.)*

   R_X *and* C_X *level-1 wire*

   R_S *switchbox*



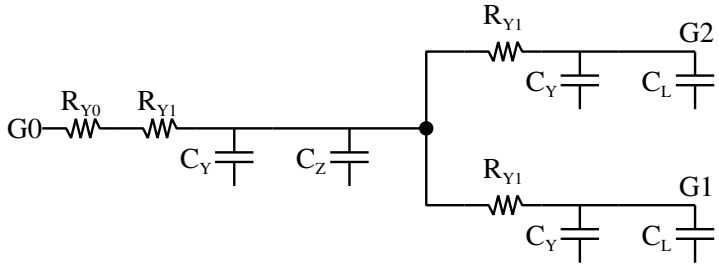
2. *Layout-1 showing only non-negligible resistances and capacitances*



3. *Layout-1, delay to G1:*

$$\begin{array}{rl} & R_{Y0}(2C_Y + C_Z + 2C_L) \\ + & R_{Y1}(2C_Y + C_Z + 2C_L) \\ + & R_{Y1}(C_Y + 2C_L) \\ + & R_{Y0}(C_Y + C_L) \\ \hline & R_{Y0}(3C_Y + C_Z + 3C_L) \\ + & R_{Y1}(3C_Y + C_Z + 4C_L) \end{array}$$

4. *Layout-2 showing only non-negligible resistances and capacitances*



5. *Layout-2, delay to G1:*

$$\begin{array}{rl} & R_{Y0}(3C_Y + C_Z + 2C_L) \\ + & R_{Y1}(3C_Y + C_Z + 2C_L) \\ + & R_{Y1}(C_Y + C_L) \\ \hline & R_{Y0}(3C_Y + C_Z + 2C_L) \\ + & R_{Y1}(4C_Y + C_Z + 3C_L) \end{array}$$

6. *Layout-1 delay minus Layout-2 delay:*

$$\begin{array}{rlcl} & R_{Y0}(3C_Y+C_Z+3C_L) & + & R_{Y1}(3C_Y+C_Z+4C_L) \\ - & R_{Y0}(3C_Y+C_Z+2C_L) & + & R_{Y1}(4C_Y+C_Z+3C_L) \\ \hline & R_{Y0}C_L & - & R_{Y1}C_Y+R_{Y1}C_L \end{array}$$

Depending on relative values of $Y0$, $Y1$, C_Y, and C_L, either Layout-1 or Layout-2 will have less delay.

7. *Solve for equal delay:*

$$\begin{aligned} 0 &= R_{Y0}C_L - R_{Y1}C_Y + R_{Y1}C_L \\ R_{Y0}C_L &= R_{Y1}C_Y - R_{Y1}C_L \\ \frac{R_{Y0}}{R_{Y1}} &= \frac{C_Y - C_L}{C_L} \\ \frac{Y0}{Y1} &= \frac{C_Y - C_L}{C_L} \end{aligned}$$

Marking:

+7 marks *RC layouts*

- **+1 mark** *branches to G1 and G2*
- **+1 mark** *includes all capacitors*
- **+1 mark** *includes all resistors*
- **+1 mark** *used R_{YO} and R_{Y1} correctly*
- **+1 mark** *capacitors follow resistors*
- **+1 mark** *each branch terminates with C_L*
- **+1 mark** *on layout-2, G2 branches out between R_{Y1} and R_{y0}*

+5 marks *delay equations*

- **+2 marks** *sum of resistances on path from G0 to G1*
- **+2 marks** *sum of downstream capacitors*
- **+1 marks** *correct algebra*

+3 marks *analysis*

- **+1 marks** *compare delay equations for layout-1 and layout-2*
- **+1 marks** *correct algebra*
- **+1 marks** *correct conclusion*

Q5 (15 Marks) Power and Performance

(*estimated time: 20 minutes*)

You are the project leader for a Waterluvian filter. You currently use the Cyclops-1 FPGA chip and are planning to upgrade to the Cyclops-2. The changes from the Cyclops-1 to the Cyclops-2 for various characteristics are given in the table below:

Characteristic	Cyclops-2
Delay for flip-flops and LUTS	$\frac{1}{2}$ of Cyclops-1
Area of flip-flops and LUTS	$\frac{1}{2}$ of Cyclops-1
Supply voltage	no change
Threshold voltage	no change
Number of LUTs	no change
Number of flip-flops	no change
Number of I/O pins	no change

You are considering two options for how to take advantage of the Cyclops-2:

Option 1: Speed demon Double the clock speed and make no changes to the design.

Option 2: Braniac Keep the clock speed the same, but change the design at the register-transfer level to double the length of the critical path. For example, if the critical path on the Cyclops-1 is Flop+Add, the critical path for the Cyclops-2 will be Flop+2Add. Assume that this change will not affect the number of FPGA cells in the design.

Use the table on the next page to describe the effects that each option will have on a variety of criteria.

NOTES:

1. Waterluvian filters are used in image processing. Their input and output are images with 1920x1200 pixels. The exact algorithm performed on the image is irrelevant and unknown even to those who invented the filter.
2. Your analysis should be *quantitative*, *justified*, and *brief*. For example, writing "profits will double because Speed demon is a cool name" will earn more marks than writing "profits will increase".
3. The environment is able to send the input image and receive the output image at whatever throughput and clock speed you choose.

Answer:

Initial analysis:

	Speed demon	*Braniac*
Area	$\frac{1}{2}$	$\frac{1}{2}$
Capacitance	$\frac{1}{2}$	$\frac{1}{2}$
Clock speed	2	1
Latency (in clock cycles)	1	$\frac{1}{2}$
Time to process a pixel (in ns)	$\frac{1}{2}$	$\frac{1}{2}$
Throughput (in pixels per clock cycle)	*1*	*1*

	Speed demon	***Braniac***
Power consumption of the Waterluvian filter	*Small decrease. Leakage power will be $\frac{1}{2}$, because of the smaller area. Switching and short circuiting power will remain unchanged, because capacitance is $\frac{1}{2}$ and clock speed is 2.*	*$\frac{1}{2}$, because capacitance is $\frac{1}{2}$ and clock speed is unchanged*
Energy per pixel	*$\frac{1}{2}$, because energy is independent of time. Capacitance is $\frac{1}{2}$, so total energy to process a pixel is $\frac{1}{2}$.*	*Same as speed demon*
Time from first input available until first output produced	*$\frac{1}{2}$, because the clock speed doubles and latency remains unchanged.*	*$\frac{1}{2}$, because the clock speed is unchanged and latency is cut in half.*
Performance in megapixels per second	*Doubles, because throughput in pixels per clock cycle remains unchanged and clock speed doubles.*	*Unchanged, because throughput in pixels per clock cycle is unchanged and clock speed is unchanged.*

Q6 (15 Marks) Clock Gating

(*estimated time: 20 minutes*)

You are the project leader for a Waterluvian filter that is 45% over its power budget. Your clock-gating design group did a preliminary design, and then everyone went away on holidays. It is now your task to finish their work.

NOTES:

1. 25% of the clock cycles have an incoming parcel.
2. The latency of the main circuit is 20 clock cycles.
3. The maximum throughput of the main circuit is 0.5.

   If you do not know how to answer the question for a throughput of 0.5, you may earn part marks by writing a $\surd$ in the box immediately below and ignoring throughput.

   I am ignoring *throughput* ☐
4. The average behaviour of the system is to receive a sequence of 15 parcels at the maximum throughput, then receive a contiguous sequence of some number of bubbles.

   If you chose to ignore throughput above, then you shall assume that the 15 parcels arrive contiguously.
5. For the main circuit by itself, without clock gating:
   - 5% of the power consumption is short-circuiting power
   - 9% of the power consumption is leakage power
6. The area of the clock-enable state machine will be 12% of the area of the main circuit.

What is the minimum clock-gating effectiveness that is needed to reduce the total power consumption to be within budget?

Answer:

Strategy:

1. *Assume that the clock-enable state machine has the same activity factor as the main circuit*
2. *Lump switching and short circuiting power together as dynamic power*
3. *Calculate the PctClk needed to bring power down to within budget.*
4. *Calculate PctBusy*
5. *Calculate Eff*

1. *Initial equations*

$$\begin{aligned}
P_{main} &= \textit{power of main circuit before clock gating} \\
P_{main.sw} &= \textit{switching power of main circuit before clock gating} \\
P_{main.sh} &= \textit{short-circuiting power of main circuit before clock gating} \\
P_{main.d} &= \textit{dynamic power of main circuit before clock gating} \\
&= P_{main.sw} + P_{main.sh} \\
&= 0.91 P_{main} \\
P_{main.lk} &= \textit{leakage power of main circuit before clock gating} \\
&= 0.09 P_{main} \\
P'_{main} &= \textit{power of main circuit with clock gating} \\
P_{tot} &= \textit{total power before clock gating} \\
&= P_{main} \\
P_{clken} &= \textit{power of clock-enable state machine} \\
&= 0.12 P_{main}
\end{aligned}$$

$$\begin{aligned}
P'_{tot} &= \textit{total power with clock gating} \\
&= P'_{main} + P_{clken}
\end{aligned}$$

$$\begin{aligned}
P_{tot} &= 1.45 P'_{tot} \\
P'_{tot} &= \textit{PctClk} \times P_{main.d} + P_{main.lk} + P_{clken}
\end{aligned}$$

2. *Calculate PctClk*

$$\begin{aligned}
P_{tot} &= 1.45 P'_{tot} \\
P_{main} &= 1.45 \left(\textit{PctClk} \times P_{main.d} + P_{main.lk} + P_{clken} \right) \\
&= 1.45 \left(\textit{PctClk} \times 0.91 P_{main} + 0.09 P_{main} + 0.12 P_{main} \right) \\
\textit{PctClk} &= 0.5271
\end{aligned}$$

3. *Calculate PctBusy*

   The length of time that the system is busy must take into account the throughput.

$$\begin{aligned} \textit{LenBusy} &= \textit{NumPcls-1}/\textit{Tput}+1+\textit{Lat} \\ &= (15-1)/0.5+1+20 \\ &= 49 \\ \textit{LenWindow} &= \textit{NumPcls}/\textit{PctValid} \\ &= 15/0.25 \\ &= 60 \end{aligned}$$

$$\begin{aligned} \textit{PctBusy} &= \textit{LenBusy}/\textit{LenWindow} \\ &= 49/60 \\ &= 0.8167 \end{aligned}$$

4. *Calculate Eff*

$$\begin{aligned} \textit{Eff} &= \frac{1-\textit{PctClk}}{1-\textit{PctBusy}} \\ &= \frac{1-0.5271}{1-0.8167} \\ &= 2.5799 \end{aligned}$$

The minimum required effectiveness is greater than 1, therefore clock gating cannot be used to reduce the power to within budget.

Marking:

–2 marks *ignored throughput*
+1 mark $P_{tot} = P_{main}$
+1 mark P'_{tot} *includes* P_{clken} *without clock gating*
+1 mark P'_{tot} *includes* $P_{main.lk}$ *without clock gating*
+1 mark P'_{tot} *includes* $P_{main.dyn}$ *scaled by PctClk*
+1 mark $P_{tot} = 1.45P'_{tot}$
+1 marks *calculate PctClk*
+2 marks *with throughput, input parcels for 30 clock cycles*
+1 marks *calculate LenBusy*
+2 marks *calculate LenWindow*
+2 marks *calculate PctBusy*
+2 marks *calculate Effectiveness*