
ECE 327 Solution to Final

2013t1 (Winter)

		Total	Approx.	
		Marks	Time	Page
Q1	Simulation	10	5	2
Q2	Area Analysis	10	10	3
Q3	Pipelining	10	15	5
Q4	Performance	20	30	9
Q5	Timing Analysis	20	30	11
Q6	Energy Model	15	20	14
Q7	Clock Gating	15	20	17
<hr/> Totals		100	130	

Q1 (10 Marks) Simulation

Q1a (5 Marks) Advantage of Delta-Cycle Simulation

What is the most significant advantage of delta-cycle simulation over RTL simulation, and how does delta-cycle simulation achieve this advantage?

Answer:

Advantage *Delta-cycle simulation supports combinational loops. RTL simulation does not.*

Achieved by *Running multiple simulation cycles per moment in time (simulation round) until values on signals stabilize.*

Marking:

Advantage

2 marks *combinational loops*

Achieved by

3 marks *multiple simulation or delta cycles, and run until stabilize*

2 marks *multiple simulation or delta cycles*

1 mark *simulation or delta cycle*

Q1b (5 Marks) Advantage of RTL Simulation

What is the most significant advantage of RTL simulation over delta-cycle simulation, and how does RTL simulation achieve this advantage?

Answer:

Advantage *RTL simulation is faster than delta-cycle simulation.*

Achieved by *Sorting combinational processes into topological order and running each process only once per moment in time.*

Marking:

Advantage

2 marks *faster*

Achieved by

3 marks *sort combinational processes into topological order and run each process only once stabilize*

2 marks *sort combinational processes into topological order (or by dependency)*

1 mark *don't need delta cycles*

Q2 (10 Marks) Area Analysis

Calculate the minimum number of FPGA cells needed to implement the VHDL code below.

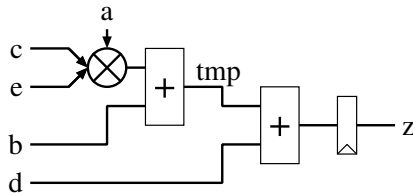
NOTES:

1. The signal `a` is `std_logic`.
2. The signals `b`, `c`, `d`, `e`, and `z` are 12-bit **unsigned**.
3. Each FPGA cell has a 4:1 LUT with carry-in and carry-out signals, and a 1-bit register. All four of the normal inputs to the LUT may be used at the same time as the carry-in signal to the LUT.
4. For full marks, you must justify your answer with a drawing and/or text.

```
process begin
  wait until rising_edge(clk);
  if a='1' then
    z <= b+c+d;
  else
    z <= b+e+d;
  end if;
end process;
```

Answer:

```
tmp <= c when a='1'
      else e;
process begin
  wait until rising_edge(clk);
  z <= b+tmp+d;
end process;
```



signal	circuitry	cells per bit	num bits	total cells
<code>tmp</code>	<i>mux and adder</i>	1	12	12
<code>z</code>	<i>adder and reg</i>	1	12	12
Total				24

With this type of FPGA cell, we can put a 2:1 mux and an adder in the same LUT.

Each cell has a LUT and a flip-flop, so we can fit both the adder and the reg for `z` into one cell per bit.

Marking:

10 marks 24 cells with correct justification

If scaled by number of bits Sum of the following:

- 2 marks *Baseline*
- 1 marks *Need a 2:1 mux, two adders, and a flop*
- 1 mark *At most 4 inputs + carry in per LUT*
- 1 mark *At most 1 output + carry out per LUT*
- 2 marks *Do not include cells for a, b, c, d, e*
- 2 marks *Use both LUT and flop in same cell*
- 1 mark *Put 2:1 mux in same cell as an adder*

If did not scale by number of bits One of the following:

- 4 marks *2 cells with explanation of 2 adds + mux + reg*

3 marks *2 cells with justification that have 5 inputs and can do 4 inputs / cell*

1 mark *1 cell*

Penalties

-2 marks *missing register for z*

-2 marks *missing optimization for mux pushing*

Q3 (10 Marks) Pipelining

In this question, you will optimize the dataflow diagram below. First, try to find an optimization that satisfies Option 1 below. If it is impossible to satisfy the goals of Option 1, then find an optimization that satisfies the goals of Option 2.

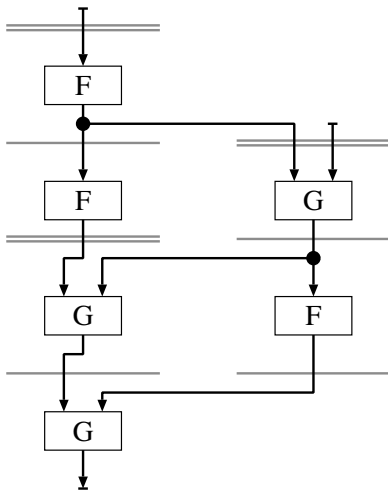
Option 1 Increase throughput *without* any increase in latency, clock period, or area.

Option 2 Optimization goals in order of decreasing importance:

1. Minimum clock period
2. Minimum area; in order of decreasing importance: F, G, Registers, Inputs, Multiplexers
3. Maximum throughput
4. Minimum latency

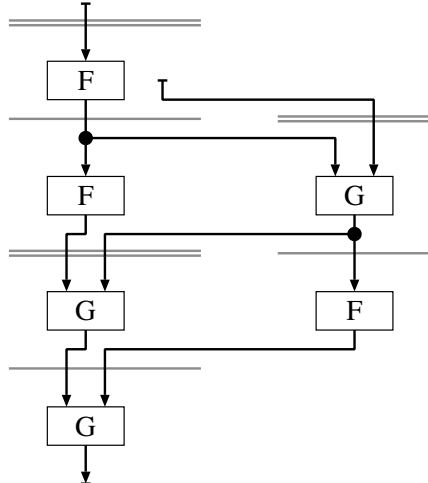
NOTES:

1. Draw the optimized DFD in the space to the left of the original DFD.
2. Compare the original and optimized DFDs by filling in the tables below the DFDs.
3. Inputs shall be registered, outputs shall be combinational.
4. The function G is commutative.
5. The only algebraic optimization that is allowed is commutativity.



If said that Option 1 is possible:

Answer:



	Original	Optimized
Latency	4	same
Throughput	1/3	1/2
Number of F	2	same
Number of G	2	same
Number of registers	4	same
Number of inputs	2	same

Marking:

DFD *7 marks total*

Analysis *3 marks total*

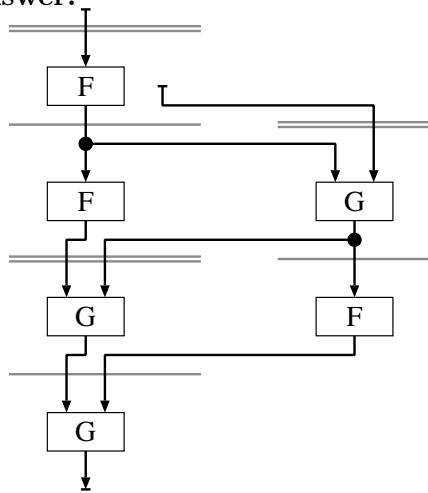
3 marks *0 mistakes*

2 marks *1–3 mistakes*

1 mark *4 or more mistakes*

If said that Option 1 is not possible:

Answer:



	Original	Optimized
Latency	4	same
Throughput	1/3	same
Number of F	2	1
Number of G	2	1
Number of registers	4	3
Number of inputs	2	1

Marking:

DFD 5 marks total

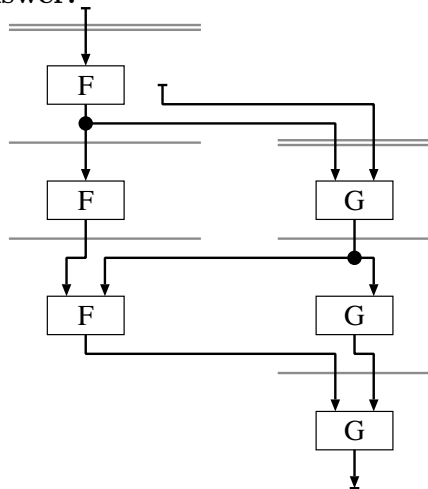
Analysis

3 marks 0 mistakes

2 marks 1-3 mistakes

1 mark 4 or more mistakes

Answer:



	Original	Optimized
Latency	4	same
Throughput	1/3	1/3
Number of F	2	1
Number of G	2	1
Number of registers	4	3
Number of inputs	2	1

Marking:

DFD 3 marks total

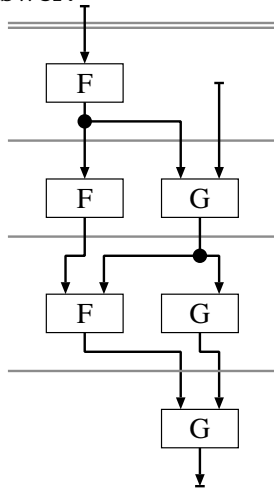
Analysis

3 marks 0 mistakes

2 marks 1-3 mistakes

1 mark 4 or more mistakes

Answer:



	Original	Optimized
Latency	4	same
Throughput	1/3	1/4
Number of F	2	1
Number of G	2	1
Number of registers	4	3
Number of inputs	2	1

Marking:

DFD 2 marks total

Analysis

3 marks 0 mistakes

2 marks 1-3 mistakes

1 mark 4 or more mistakes

Q4 (20 Marks) Performance

You work for Inchworm Productions, a company that designs Waterluvian filters. You just released your current product, which has below average performance. Your manager has asked you to analyze the performance gains needed for the performance of future Inchworm products to be 10% above average in two years.

NOTES:

1. The average performance of Waterluvian filters doubles every 18 months.
2. The average Waterluvian filter now is 20% faster than the current Inchworm product.

By what percentage must your performance increase each year so that the product you release in two years will be 10% faster than the average Waterluvian filter in two years?

Answer:

1. Initial equations

$$\begin{aligned} P_{avg}(t_0) &= 1.20 \times P_i(t_0) \\ P_{avg}(t_1) &= 2^{(t_1-t_0)/18} \times P_{avg}(t_0) \\ P_i(t_1) &= 1.1 \times P_{avg}(t_1) \\ t_1 - t_0 &= 24 \\ P_i(t_1) &= n^{(t_1-t_0)/12} \times P_i(t_0) \end{aligned}$$

2. Only one equation has n , so start with that and solve for n

$$\begin{aligned} P_i(t_1) &= n^{(t_1-t_0)/12} \times P_i(t_0) \\ \frac{P_i(t_1)}{P_i(t_0)} &= n^{(t_1-t_0)/12} \\ \left(\frac{P_i(t_1)}{P_i(t_0)}\right)^{12/(t_1-t_0)} &= \left(n^{(t_1-t_0)/12}\right)^{12/(t_1-t_0)} \\ \left(\frac{P_i(t_1)}{P_i(t_0)}\right)^{12/(t_1-t_0)} &= n \\ n &= \left(\frac{P_i(t_1)}{P_i(t_0)}\right)^{12/(t_1-t_0)} \end{aligned}$$

3. Solve for $P_i(t_0)$

$$\begin{aligned} P_{avg}(t_0) &= 1.20 \times P_{avg}(t_0) \\ P_i(t_0) &= \frac{P_{avg}(t_0)}{1.20} \end{aligned}$$

4. Solve for $P_i(t_1)$

$$\begin{aligned} P_i(t_1) &= 1.1 \times P_{avg}(t_1) \\ &= 1.1 \times 2^{(t_1-t_0)/18} \times P_{avg}(t_0) \end{aligned}$$

5. Substitute into equation for n

$$\begin{aligned}n &= \left(\frac{1.1 \times 2^{(t_1-t_0)/18} \times P_{avg}(t_0)}{P_{avg}(t_0)/1.20} \right)^{12/(t_1-t_0)} \\&= \left(1.1 \times 2^{24/18} \times 1.20 \right)^{12/24} \\&= 1.82378 \\&= 1.824\end{aligned}$$

6. Final answer: Performance must improve at a rate of 82.4% per year

Marking:

3 marks $P_{avg}(t_0) = 1.20 \times P_i(t_0)$

3 marks $P_{avg}(t_1) = 2^{(t_1-t_0)/18} \times P_{avg}(t_0)$

3 marks $P_i(t_1) = 1.1 \times P_{avg}(t_1)$

3 marks $P_i(t_1) = n^{(t_1-t_0)/12} \times P_i(t_0)$

1 mark $t_1 - t_0 = 24$

2 marks Use second $P_i(t_1)$ equation to solve for n

2 marks exponent of 24/18

2 marks exponent of 12/24

1 mark neatness and clarity

Q5 (20 Marks) Timing Analysis

You have just been promoted to a research and development group working on a new cell library for ASICs (*e.g.*, implementations of AND gates, OR gates, flip-flops, *etc.*).

Your new group does not yet have a product, but they already have hired a branding consultant who has come up with a name, “Precidel” (for “precise delay”) and a slogan: “*Precidel: same delay, every day*”.

The goal of the Precidel cell library is to eliminate the variability in delay through gates. For example, each Precidel AND gate will always have a delay of precisely 1.0 ns.

Normal (*i.e.*, non-Precidel) gates have variability in delay. Due to manufacturing tolerances, there is variability in delay across different AND gates (*e.g.*, one AND gate will have a delay of 1.1 ns and another will have a delay of 0.9 ns). Due to changes in supply voltage and temperature there is variability in delay across time for an individual gate (*e.g.*, today a particular AND gate has a delay of 1.1 ns and tomorrow the same gate will have a delay of 0.9 ns).

Your job is to work on the timing analysis tool for Precidel circuits. Your first task is to choose a definition for *viability*. Your manager has given you three possible definitions of viability to choose from:

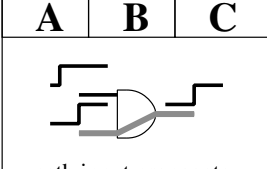
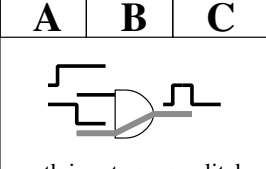


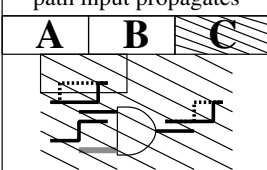
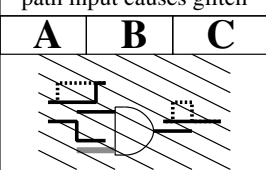
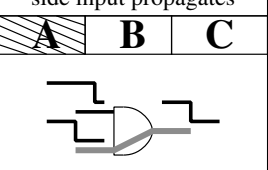
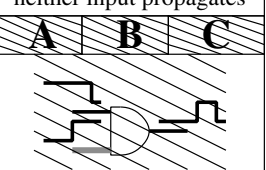
- A** : A path is viable *if-and-only-if* for each gate along the path:
 - the side input has a non-controlling value.
- B** : A path is viable *if-and-only-if* for each gate along the path:
 - the side input has a non-controlling value
 - or** the side input arrives late and the path input has a controlling value
- C** : A path is viable *if-and-only-if* for each gate along the path:
 - the side input arrives early and has a non-controlling value
 - or** the side input arrives late and the path input has a controlling value

Worksheet is on next page

Which definition (A–C) is the best choice for the definition of viability for Precidel circuits?

Answer:

C

	side=non-ctrl path=non-ctrl	side=non-ctrl path=CTRL	side=CTRL path=CTRL	side=CTRL path=non-ctrl
	A B C	A B C	A B C	A B C
Early Side	 <p>path input propagates</p>	 <p>path input causes glitch</p>	 <p>side input propagates</p>	 <p>neither input propagates</p>
Late Side	 <p>monotone speedup</p>	 <p>monotone speedup</p>	 <p>path input propagates</p>	 <p>side input causes glitch</p>

Justify your answer by describing the most significant advantage of your chosen definition over each of the other two definitions.

My choice is better than **A** because:

Answer:

Definition A says that if the side-input has a controlling value, then the path is false. But, if the side input arrives after the path input and the path input has a controlling value, then the edge on the path input will propagate and the edge on the side input will not propagate. Definition A incorrectly says that some paths are false. Thus, definition A will underestimate the minimum clock period and overestimate the maximum clock speed.

My choice is better than **B** because:

Answer:

Our choice of definition, C, is more restrictive than definition B. The difference is that definition C says that a path with a late-arriving side input and both the path input and side input have a non-controlling values is false. This situation is one of the two cases of monotone speedup. Because Precidel circuits have no variability, speedup will not occur, so it is safe to exclude this situation from the definition of viability.

Marking:

Better than A

10 marks A says incorrectly that a path is false when in fact an edge could propagate: late-arriving side input with controlling value.

9 marks A does not take into account late-arriving side inputs

6 marks *doesn't take into account situation where both inputs have controlling values*

4 marks *monotone speedup depends on when the side input arrives*

Better than B

10 marks *Precidel eliminates need to consider monotone speedup. B says that a late-arriving side input with a non-controlling value is viable, but in fact, the path input will not cause the last edge on the output.*

9 marks *Precidel eliminates needs to consider monotone speedup.*

Better than C

8 marks *C takes into account monotone speedup*

7 marks *C says that a late arriving side input with non-controlling values on both the path input and side input is not viable, but with if the side input speeds up, an edge could propagate from the path input*

6 marks *C does not take into account monotone speedup*

5 marks *C says that a late arriving side input with non-controlling values on both the path input and side input is not viable*

5 marks *monotone speedup is not required with precise delays*

5 marks *if side input is non-controlling, it doesn't matter whether it is early or late*

4 marks *early side input and non-controlling leads to a glitch*

3 marks *C takes into account late side inputs and a non-controlling value on the path input*

3 marks *describes how late side inputs work*

Penalties

4 marks *understanding of non-controlling value*

2 marks *understanding of late-arriving vs early-arriving side inputs*

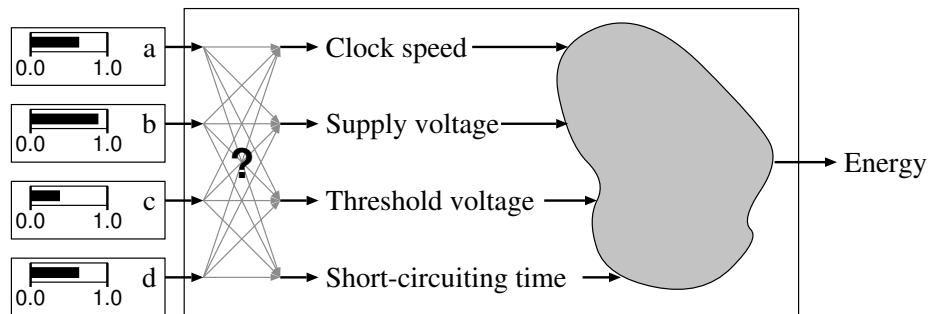
2 marks *understanding of normal definition of viability*

2 marks *understanding of monotone speedup*

Q6 (15 Marks) Energy Model

You have just started your co-op job at Silicon Waffles, a startup company working on a new power and energy analysis tool. The previous co-op student wrote a program to calculate the *total energy consumption* of a 6-bit one-hot counter for six clock cycles. The model is functionally correct, but the program is written so poorly that no one can understand it. Your job is to reverse engineer the meaning of the input variables by running the program and examining the output.

The inputs are named **a**, **b**, **c**, and **d**. These inputs control the circuit parameters of clock speed, supply voltage, threshold voltage, and short-circuiting time. But, no one knows which input corresponds to which circuit parameter:



The inputs **a**, **b**, **c**, and **d** are scaling factors that range between 0.0 and 1.0. Each scaling factor controls the value of a circuit parameter: 0.0 = minimum realistic value, 0.5 = average value, 1.0 = maximum realistic value.

For example, if this was a model for the amount of pollution generated by a vehicle on a highway and the parameter for vehicle speed ranged from 80 km/hr to 120 km/hr, then the relationship between the input and speed would be as follows:

input	speed
1.0	120 km/hr
0.5	100 km/hr
0.0	80 km/hr

The output value for energy consumption ranges between 0 and 1000.

It is easy to run many tests and analyze the results, because it takes less than 2 seconds to run the program for a given set of input values.

Your answer shall describe a sequence of steps that you could use to determine which input corresponds to which circuit parameter.

An example answer that is in the correct format, but has no helpful technical content:

1. Run a series of tests that gradually increase the inputs together from 0.0 to 1.0.
2. If energy consumption increases as the variables increase, then threshold voltage = a, because “a” is the first letter in “aagaard”. Otherwise, threshold voltage = d, because “d” is the last letter in “aagaard”.
3. Set the input for threshold voltage to 1.0. For each of the remaining three inputs, make the chosen input oscillate as a sine wave and hold the other inputs constant at 0.5. The input that causes the energy consumption to oscillate corresponds to clock speed.

Answer:

1.

- (a) For each input (a,b,c,d): set the chosen input to 0.0 and set all of the other inputs to 1.0, then run the program
- (b) Exactly one of the inputs should cause the output to be 0, independent of the other inputs. This input controls the supply voltage, because supply voltage is the only parameter that affects all three types of power.

2.

- (a) For each input other than supply voltage: set all of the other inputs to 0.5 and run a series of tests as the chosen input is gradually increased from 0.0 to 1.0.
- (b) Two of the inputs should cause the energy consumption to decrease as the input value increases and one of the inputs should cause the energy consumption to increase.
- (c) The input that causes the energy consumption to increase is short-circuiting time.
- (d) The two remaining parameters are threshold voltage and clock speed.

Threshold voltage affects leakage energy. As described next, clock speed also affects leakage power, because we are measuring energy consumption per clock cycle.

Clock speed affects dynamic power, but we are measuring energy consumption for six clock cycles. Power consumption is energy consumption per unit time. Energy consumption per clock cycle is the dual of power consumption, in the sense that the roles of dynamic and static power are reversed.

The dynamic energy consumption per clock cycle is independent of clock period, because it describes power consumed by events that happen within a clock cycle, independent of how long the clock cycle lasts. Static energy consumption per clock cycle is dependent upon the length of the clock cycle, and so is dependent upon clock speed.

As clock speed increases, leakage energy will decrease linearly. As threshold voltage increases, leakage energy will decrease non-linearly. Thus, the input parameter with a linear effect on energy consumption is clock speed and the input parameter with the non-linear effect is threshold voltage.

Marking:

6 marks Primary relationships. Sum of the following:

1.5 marks increasing clock speed causes energy to decrease

1.5 marks increasing supply voltage causes energy to increase OR setting supply voltage to 0 will cause energy to be 0

1.5 marks increasing threshold voltage causes energy to decrease

1.5 marks increasing short-circuiting time causes energy to increase

5 marks (max) Secondary relationships. Sum of the following, up to a max of 5 marks:

2 marks increasing clock speed causes energy to decrease linearly

2 marks changing clock speed will cause larger change in energy than changing short-circuiting time

2 marks increasing supply voltage causes energy to increase quadratically

2 marks increasing threshold voltage causes energy to decrease non-linearly

3 marks overall correctness of strategy

1 mark *clarity and neatness*

-1 mark *In $F \propto (V_{DD} - V_{Th})^2 / V_{DD}$, F is the maximum clock speed at which the circuit will work correctly. The input parameter to the energy model is actual clock speed.*

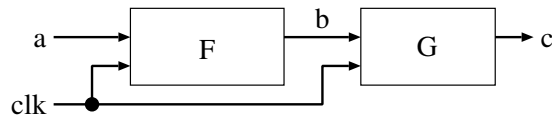
-2 marks *Ignore leakage current*

-2 marks *Confusion between energy and power*

-1 mark *Confusion between exponential and quadratic*

Q7 (15 Marks) Clock Gating

In this question you will analyze the potential power savings of a clock gating scheme for the system below.



	F	G
Latency	15	20
Throughput	1	1
Area	200	180
Activity factor	0.20	0.30
Supply voltage	1.2 V	1.2 V
Clock speed	500 MHz	500 MHz

Calculate the maximum percentage of power of the original system that could be saved with a clock gating scheme is 90% effective.

NOTES:

- The typical sequence of inputs is 25 valid parcels followed by 55 bubbles.
- You may use either a single clock-enable state machine for both F and G, or one clock enable state machine for F and one for G. If you use two clock enable state machines, each state machine is 90% effective for the circuit (F or G) whose clock it enables.

Answer:

Assumptions:

- Leakage power is negligible
- Power consumption of clock enable state machine is negligible in comparison to main circuit

The best clock gating scheme will be to have a separate clock-enable state machine for each component (F and G).

- Window length

$$25 \text{ valid parcels} + 55 \text{ bubbles} = 80 \text{ clock cycles for window}$$

- Useful equations

$$PctValid = \frac{NumValidInput + Latency}{Window}$$

$$Eff = \frac{1 - PctClk}{1 - PctValid}$$

$$PctClk = 1 - Eff(1 - PctValid)$$

$$A' = PctClk \times A$$

- F

$$PctValid_F = \frac{25 + 15}{80} = 0.5$$

$$PctClk_F = 1 - 0.9(1 - 0.5) = 0.55$$

$$A'_F = 0.55 \times 0.2 = 0.11$$

4. G

$$\begin{aligned} PctValid_G &= \frac{25 + 20}{80} \\ &= 0.5625 \end{aligned}$$

$$\begin{aligned} PctClk_G &= 1 - 0.9(1 - 0.5625) \\ &= 0.60625 \end{aligned}$$

$$\begin{aligned} A'_G &= 0.60625 \times 0.3 \\ &= 0.1819 \end{aligned}$$

5. Total Power

$$\begin{aligned} P'/P &= \frac{A'_F \times C_F + A'_G \times C_G}{A_F \times C_F + A_G \times C_G} \\ &= \frac{0.1100 \times 200 + 0.1819 \times 180}{0.2 \times 200 + 0.3 \times 300} \\ &= 0.5823 \end{aligned}$$

With clock gating, power is 58.23% of original power.

Clock gating can save $100\% - 58.23\% = 41.77\%$ of the original power.

Marking:

- 2 marks** Use two clock gating circuits
- 1 mark** Assume leakage power is negligible
- 1 mark** Assume clock enable state machine power is negligible
- 1 marks** Window
- 2 marks** Percentage valid
- 2 marks** Percentage clocked
- 2 marks** Activity factor with clock gating
- 2 marks** Overall power
- 1 marks** Percentage reduction in power
- 1 mark** Neatness and clarity