
First Name

Last Name

First letter
of last name

UW Userid

ECE 327 Final

2013t2 (Spring)

Instructions and General Information

- 100 marks total
- Time limit: 2.5 hours (150 minutes)
- No books, no notes, no computers. Calculators are allowed
- If you need extra paper, request some from a proctor.
- Write neatly.
- To earn part marks, you must show the formulas you use and all of your work.
- **The proctors and instructors will not answer questions, except in cases where an error on the exam is suspected. If you are confused about a question, write down your assumptions or interpretation.**
- **Justifications of answers will be marked according to correctness, clarity, and concision.**

		Total Marks	Approx. Time	Page	
Q0	!!Almost Free!!	<input type="text"/>	1	0	3
Q1	Dataflow Diagram Design	<input type="text"/>	15	20	4
Q2	Allocation and Control	<input type="text"/>	15	25	6
Q3	Optimality Metrics	<input type="text"/>	10	10	10
Q4	Performance	<input type="text"/>	15	15	11
Q5	Latch Analysis	<input type="text"/>	15	20	13
Q6	Timing Analysis	<input type="text"/>	15	20	15
Q7	Power	<input type="text"/>	15	25	17
Totals		<input type="text"/>	100	135	

Potentially Useful Information

$$P = \frac{1}{2}(A \times C \times V^2 \times F) + (\tau \times A \times V \times \text{ISh} \times F) + (V \times \text{IL})$$

$$T = \frac{\text{Ins} \times C}{F}$$

$$F \propto \frac{(V - V_t)^2}{V}$$

$$P = V \times I$$

$$P = \frac{W}{T}$$

$$\text{IL} \propto e^{\frac{-q \times V_t}{k \times T}}$$

$$S = \frac{T_1}{T_2}$$

$$M = \frac{F/10^6}{\left(\sum_{i=0}^n \text{PI}_i \times C_i\right)}$$

$$A' = (1 - E(1 - P_v))A$$

$$q = 1.60218 \times 10^{-19} \text{C}$$

$$k = 1.38066 \times 10^{-23} \text{J/K}$$

$$\log_x y = \frac{\log y}{\log x}$$

$$(x^y)^z = x^{(yz)}$$

$$(x^y)(x^z) = x^{(y+z)}$$

$$a = b^c \text{ is equivalent to:}$$

$$a^{1/c} = b$$

Q0 (1 Mark) !!Almost Free!!*(estimated time: 0 minutes)*

Ten years from now, what, if anything, will you remember about this course, other than TimBits?

Q1 (15 Marks) Dataflow Diagram Design

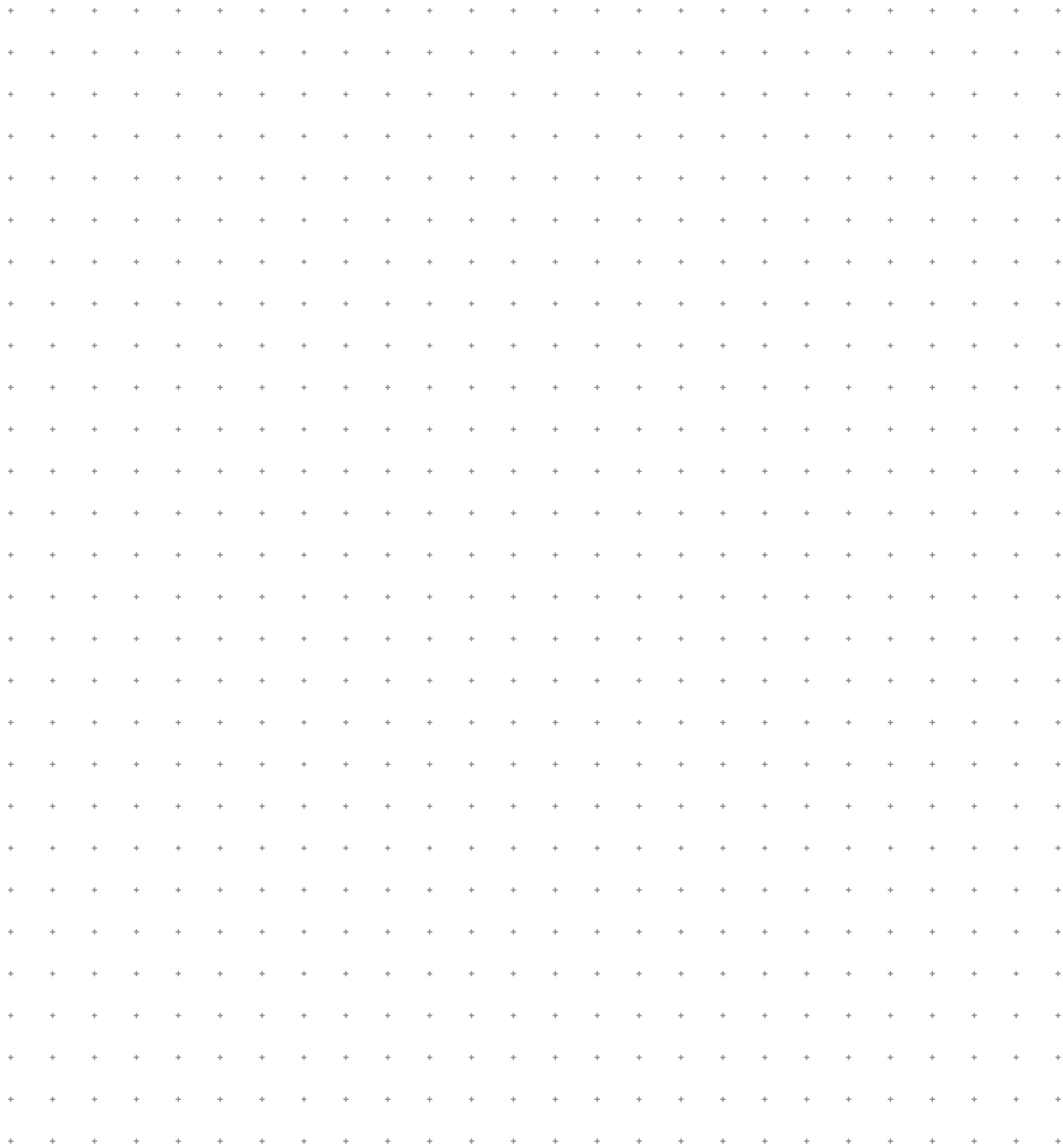
(estimated time: 20 minutes)

Draw a dataflow diagram for the system described below:

$$z = \max(3a - (b+c+d), 3b - (a+c+d), 3c - (a+b+d), 3d - (a+b+c))$$

NOTES:

1. The inputs are a , b , c , and d .
2. The output is z .
3. Hint: $3i - j = 4i - (i + j)$.
4. Hint: $\max(i - k, j - k) = \max(i, j) - k$
5. The *inputs* shall be registered.
6. The *output* may be either registered or combinational.
7. The \max operation shall be implemented using MAX components with two inputs (*e.g.*, $m = \text{MAX}(a,b)$).
8. The *throughput* shall be at least $1/2$.
9. Clock period:
 - Each 2-input MAX component has a delay of t_m
 - Each 2-input MULT component has a delay of t_m (same as a MAX)
 - Each 2-input ADD component has a delay of t_n
 - Each 2-input SUB component has a delay of t_n (same as an ADD)
 - $t_n < t_m < 2t_n$
 - The clock period shall be at most $\text{FLOP} + 2t_n$. Reducing the clock period below this limit is an optimization goal.
10. Optimization goals, in order of *decreasing* importance:
 - (a) Minimal area, in order of *decreasing* importance:
 - i. Inputs
 - ii. Registers
 - iii. Multipliers
 - iv. MAX components
 - v. Total number of adders and subtracters
 - (b) Minimal clock period
 - (c) Minimal latency
11. You may use *algebraic optimizations*, as long as the value of z is correct.
12. You may schedule the input values to arrive in any order, but you may read each input value only once.
13. You do *not* need to do any allocation.



Latency

Throughput

Clock period

Number of inputs

Number of registers

Number of multipliers

Number of MAXs

Number of adders

Number of subtracters

Q2 (15 Marks) Allocation and Control

(estimated time: 25 minutes)

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 *not* change the allocations that have already been done.
2. You may use 2:1 multiplexers, and may combine 2:1 multiplexers to create larger multiplexers. (e.g. two 2:1 multiplexers may be combined to create a 3:1 multiplexer)
3. Optimization goals in order of highest priority to lowest:
 - (a) Minimize the number of adders
 - (b) Minimize the number of input and output ports
 - (c) Minimize the number of registers
 - (d) Minimize the number of 2:1 multiplexers (including the 2:1 multiplexers used to build larger multiplexers)
4. You may *not* perform scheduling optimizations on the dataflow diagram.
5. The *only* algebraic optimization that you may perform is *commutativity*.
6. The system has combinational inputs. If you do not know how to do allocation and write the control table for combinational inputs, you may earn *part marks* by writing a \surd in the box below and using registered inputs.

I am using registered inputs, not combinational inputs.

There are extra copies of the dataflow diagram and space for control tables on the next page.

Q3 (10 Marks) Optimality Metrics

(estimated time: 10 minutes)

You have just been promoted to your dream job: drawing dataflow diagrams for the next-generation Waterluvian filter. The details of Waterluvian filters are not relevant to this question. At your first design review, you present several different dataflow diagrams.

Your manager asks you if you can define a formula to *estimate the optimality of a dataflow diagram*. The optimality of Waterluvian filters is measured using Megapixels-per-second per Watt (MPPS/Watt).

The estimate of optimality will be used to compare the *relative* optimality of dataflow diagrams. The estimate is *not* intended to give the actual optimality of a system that implements the dataflow diagram. That is, if you *estimate* the optimality of DFD1 to be 500 and DFD2 to be 400, then an implementation of DFD1 will have a higher actual optimality than an implementation of DFD2, but the actual optimality of DFD1 is not necessarily 500 MPPS/Watt.

NOTES:

- The inputs to the formula shall be some or all of the following, as measured on a dataflow diagram:
 - **A** = area (FPGA cells)
 - **F** = clock speed (MHz)
 - **L** = latency (clock cycles)
 - **T** = throughput (pixels per clock cycle)
- The output of the formula shall be an estimate of optimality.
- The estimate of optimality will be used only to compare Waterluvian filters for the same type of FPGA.

Is it possible to estimate optimality? **For full marks, you must justify your answer.**

Yes Formula: estimated optimality = _____

No

Explanation of formula, or justification that cannot estimate optimality:

Q4 (15 Marks) Performance

(estimated time: 15 minutes)

The simple version of Amdahl's law is "make the common case fast". This question explores the issues of "how common" and "how fast".

You are the project leader for an edge-detection circuit. Two of your group members have proposed performance optimizations. You have time to implement only one of the optimizations.

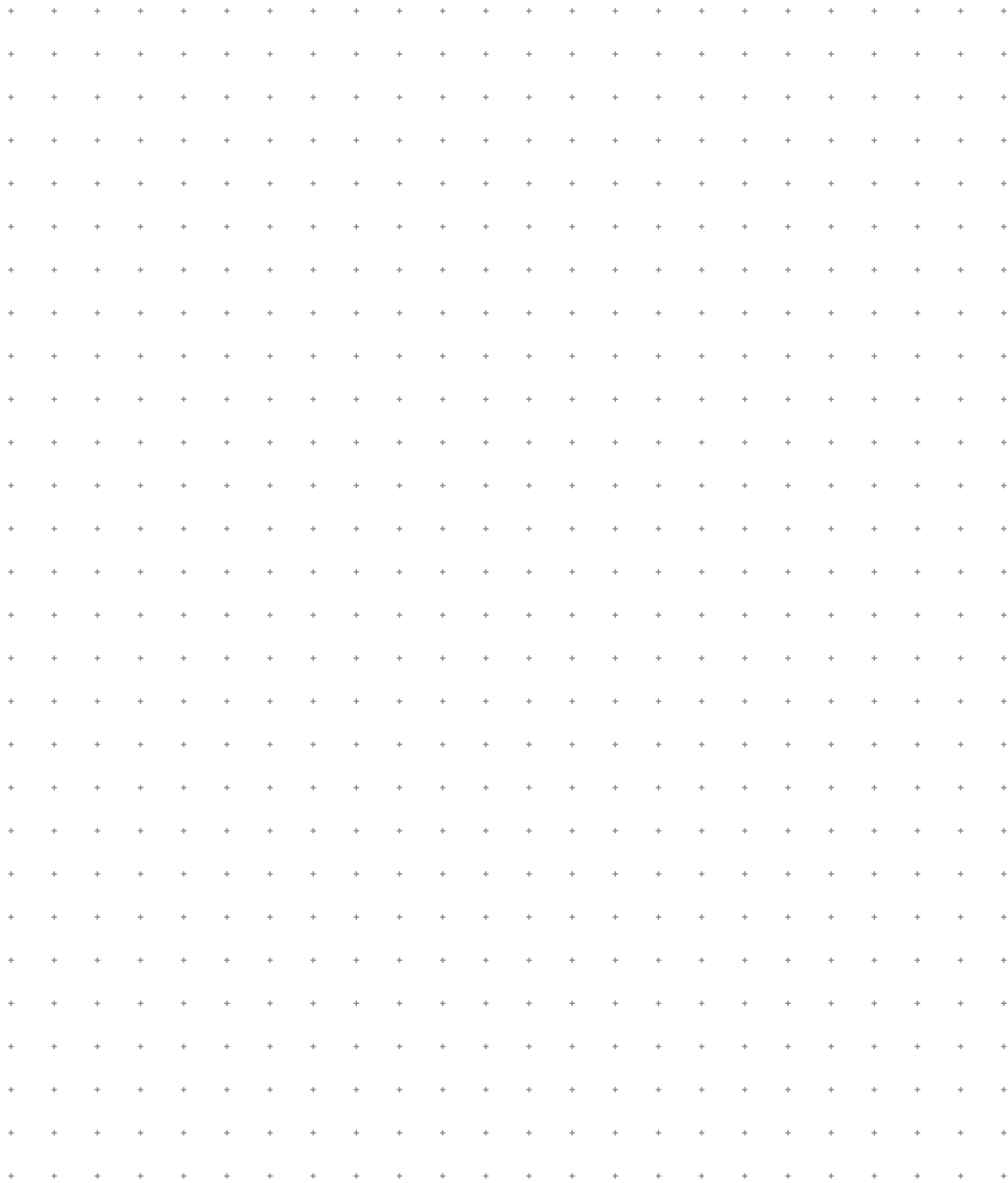
NOTES:

1. Eric Edge has proposed an optimization that will give a 90% performance improvement for pixels that are on an edge.
2. Mark Dark has proposed an optimization that will give a 10% performance improvement for pixels that are *not* on an edge.
3. On average, 10% of the pixels are on an edge.
4. Currently, it takes n_e clock cycles to process a pixel that is on an edge and n_m clock cycles to process a pixel that is *not* on an edge
5. Your system is not pipelined. Neither Eric's nor Mark's optimization will introduce pipelining.

Your task is to determine under what condition will Eric Edge's optimization be better than Mark Dark's optimization. **For full marks, you must justify your answer.**



The answer shall be written on the next page



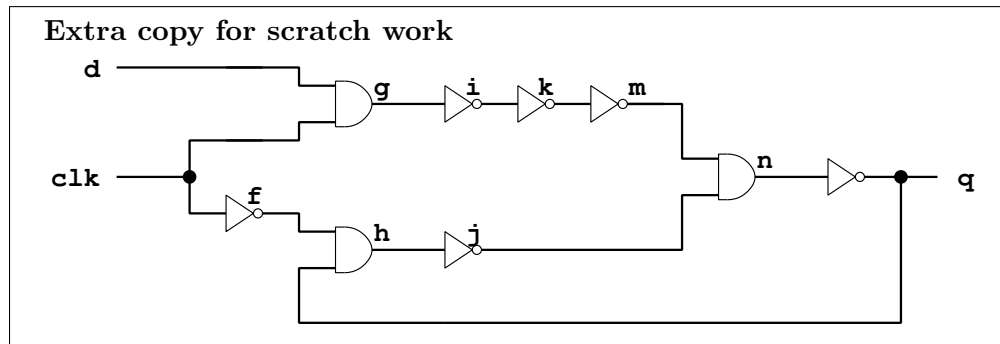
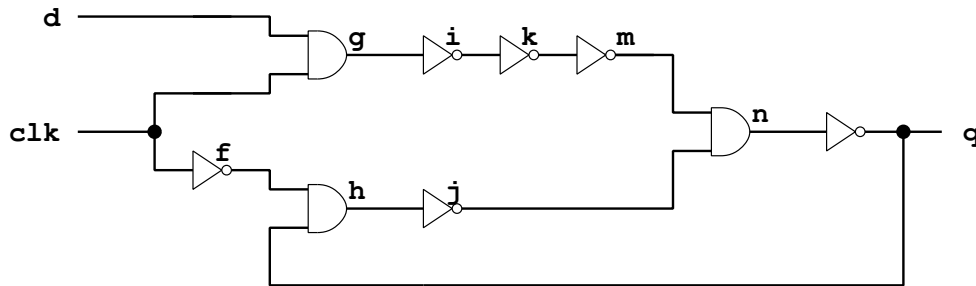
- 90% performance improvement for edges is always better
- 10% performance improvement for non-edges is always better
- Both optimizations will give the same performance improvement
- 90% performance improvement for edges is sometimes better

If your answer is “sometimes”, then give the condition for when a 90% performance improvement for edges is better:

Q5 (15 Marks) Latch Analysis

(estimated time: 20 minutes)

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



NOTES:

1. The delay through each gate is 1 ns.

Q5a (3 Marks) Good or Bad?

Is the circuit a correct implementation of a latch?

Yes	No
<input type="checkbox"/>	<input type="checkbox"/>

The two remaining parts of this question (Q5b and Q5c) are divided into two columns. Use the left column if you answered *yes*. Use the right column if you answered *no* above.

Q5b (7 Marks) Analysis

If the circuit is a correct implementation of a latch, determine if it is active-high or active-low and calculate the timing parameters below.

Active-high or active-low?	<input type="text"/>
Clock-to-Q	<input type="text"/>
Setup	<input type="text"/>
Hold	<input type="text"/>

If the circuit is *not* a correct implementation of a latch, explain why.

Q5c (5 Marks) Modification

If the circuit is a correct implementation of a latch, modify the circuit using the diagram below to *reduce the clock period* as much as possible, while preserving the functionality of the latch.

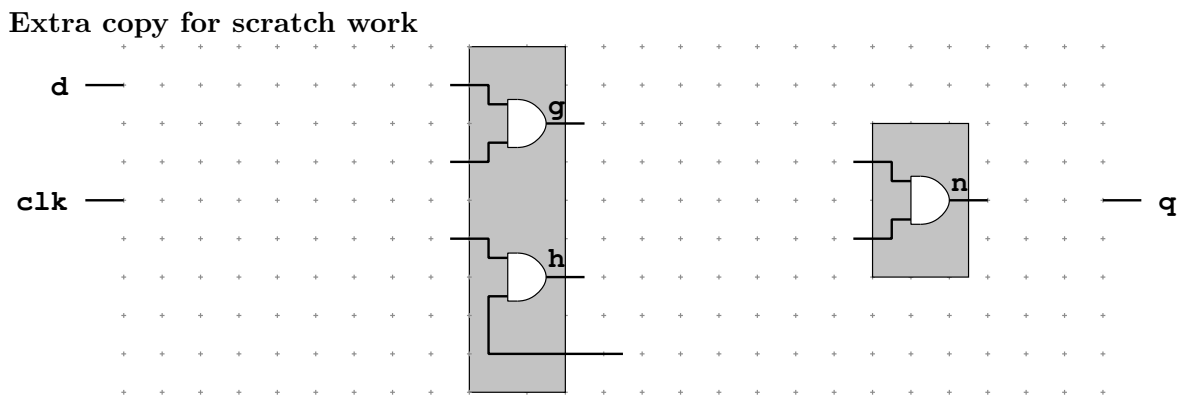
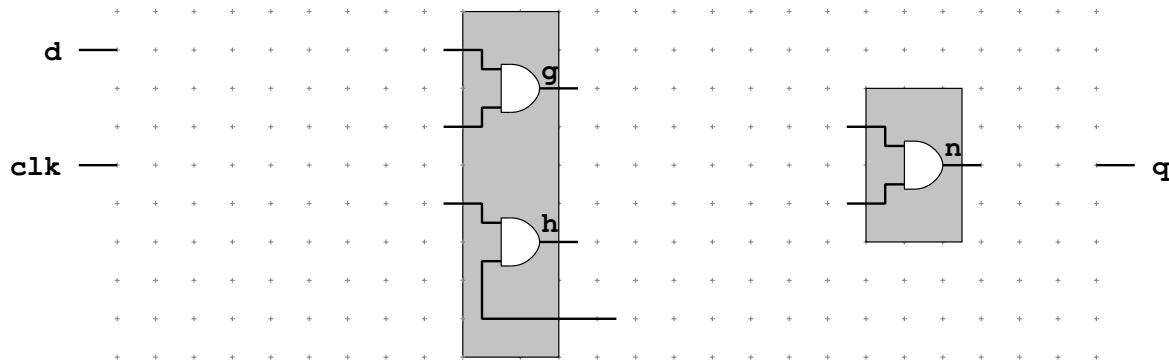
NOTES:

1. You may *not* modify the part of the circuit in the gray rectangles.
2. If you cannot reduce the clock period by modifying the latch, then justify that you cannot.

If the circuit is *not* a correct implementation of a latch, modify the circuit using the diagram below to turn it into a correctly functioning latch.

NOTES:

1. Make the minimum modifications necessary.
2. You may *not* modify the part of the circuit in the gray rectangles.



Justification, if the latch is correct and you are unable to reduce the clock period:

Q7 (15 Marks) Power*(estimated time: 25 minutes)*

You are developing a circuit that has a strict power budget. The current plan for meeting the power budget is to run the clock well below the maximum clock speed that the circuit could support. Your task is to determine how much you could increase the clock speed if you add a clock-gating circuit. More precisely: *calculate the maximum clock speed at which you can run the circuit such that the total power with clock gating is the same as the power consumption now.*

Statistics of main circuit

Latency	10
Throughput	1
Average number of consecutive valid parcels	5
Average number of consecutive bubbles	20
Activity factor	25%
Clock frequency	300 MHz
Area	500 cells
Switching power	27.0 mW
Leakage power	5.0 mW
Short circuiting power	3.0 mW

Statistics of clock-gating circuit

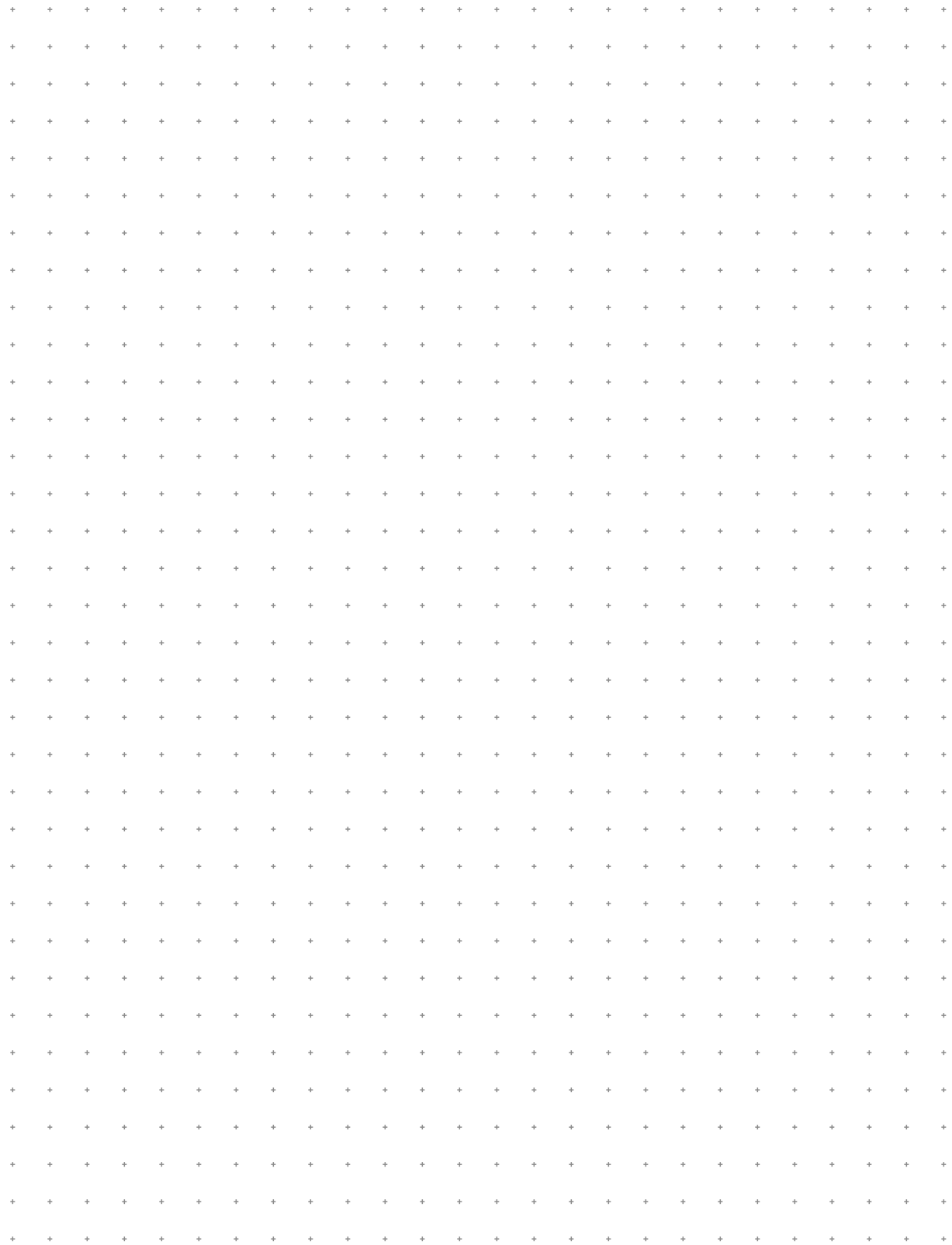
Area	60 cells
Effectiveness	85%

NOTES:

1. The clock-gating circuit and the main circuit shall be run at the same clock speed.
2. Even with clock gating, the power budget will be the limiting factor for the clock speed. The delay through the circuit will *not* limit the clock speed.
3. All circuits shall use the same type of FPGA, same supply voltage, and same threshold voltage.
4. Clearly list any assumptions you use.

Assumptions:

Work space continues on next page



Maximum clock speed with clock gating: