First Name	Last Name	First letter of last name
UW Userid		

ECE 327 Midterm

2016t2 (Spring)

Instructions and General Information

- 100 marks total
- Time limit: 1 hour and 20 minutes (80 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!!		2	2	2
Q1	VHDL		25	15	3
Q2	Area Analysis		25	15	4
Q3	FSM		25	20	6
Q4	DFD		25	20	8
Totals			100	72	

Q0 (2 Marks) !!Almost Free!!

(*estimated time: 2 minutes*)

Q0a (1 Mark) Best part

What is the best part of the course?

Q0b (1 Mark) Most improve

What one thing could be done to most improve the course for the remainder of the term?

Q1 (25 Marks) VHDL

(*estimated time: 15 minutes*)

On the first day of your dream co-op job at Amabagookify, the company goes bankrupt and you lose your job. You decide to create a startup company based on your idea for a new hardware description language, "Why Wait?", abbreviated as "yW". To get the company going quickly, you decide to base yW on VHDL, but eliminate `wait` statements and sensitivity lists, because they are too complicated.

Q1a (3 Marks) Disadvantage

What is the most significant disadvantage of eliminating wait statements?

Q1b (22 Marks) Semantics

So far, you have decided that in each simulation cycle:

- each process is executed exactly once
- the processes are executed in the order in which they appear in the architecture.

Can you define the rest of the semantics of yW such that a program with a combinational loop will have the same behaviour as in VHDL?

If **yes**, briefly describe the required semantics.

If **no**, write a sample program with a combinational loop and explain why the program would be illegal in yW or how its behaviour in yW would be different from that in VHDL.

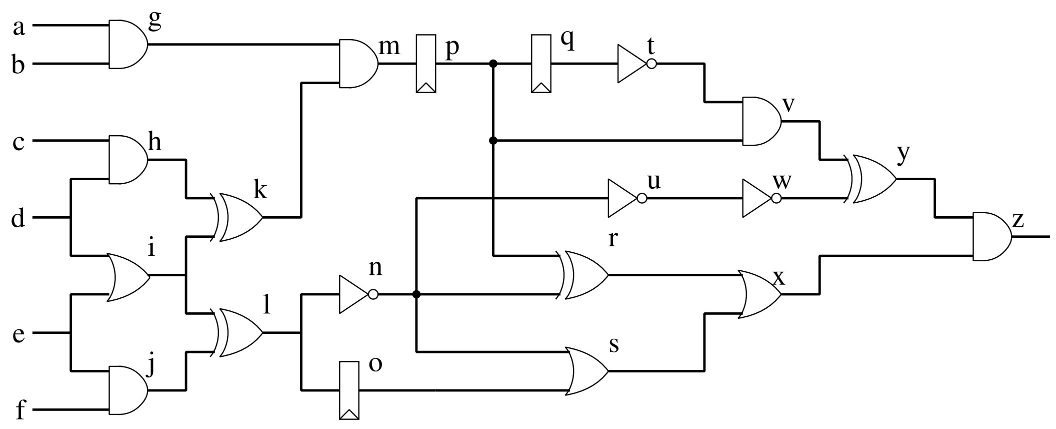
Q2 (25 Marks) Area Analysis

(*estimated time: 15 minutes*)

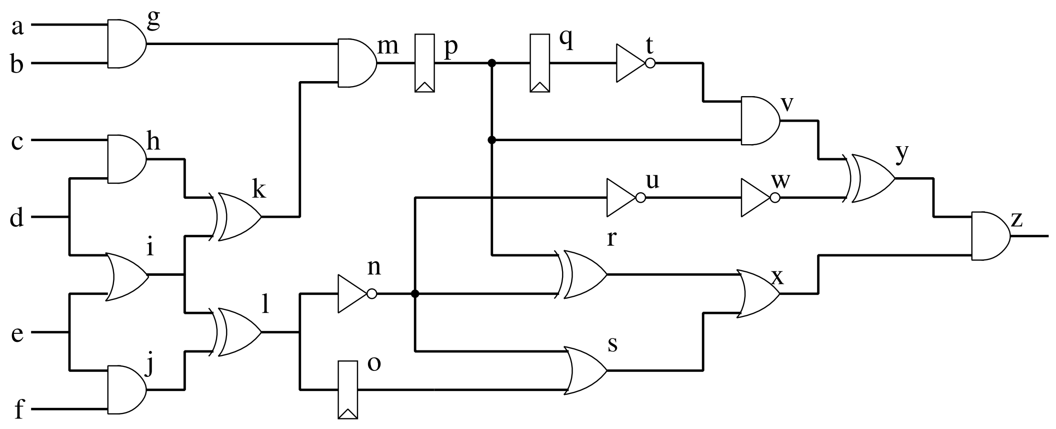
Design an FPGA implementation of the gate-level circuit shown below that uses the minimum number of FPGA cells. Use the FPGA cells on the following page to answer the question.

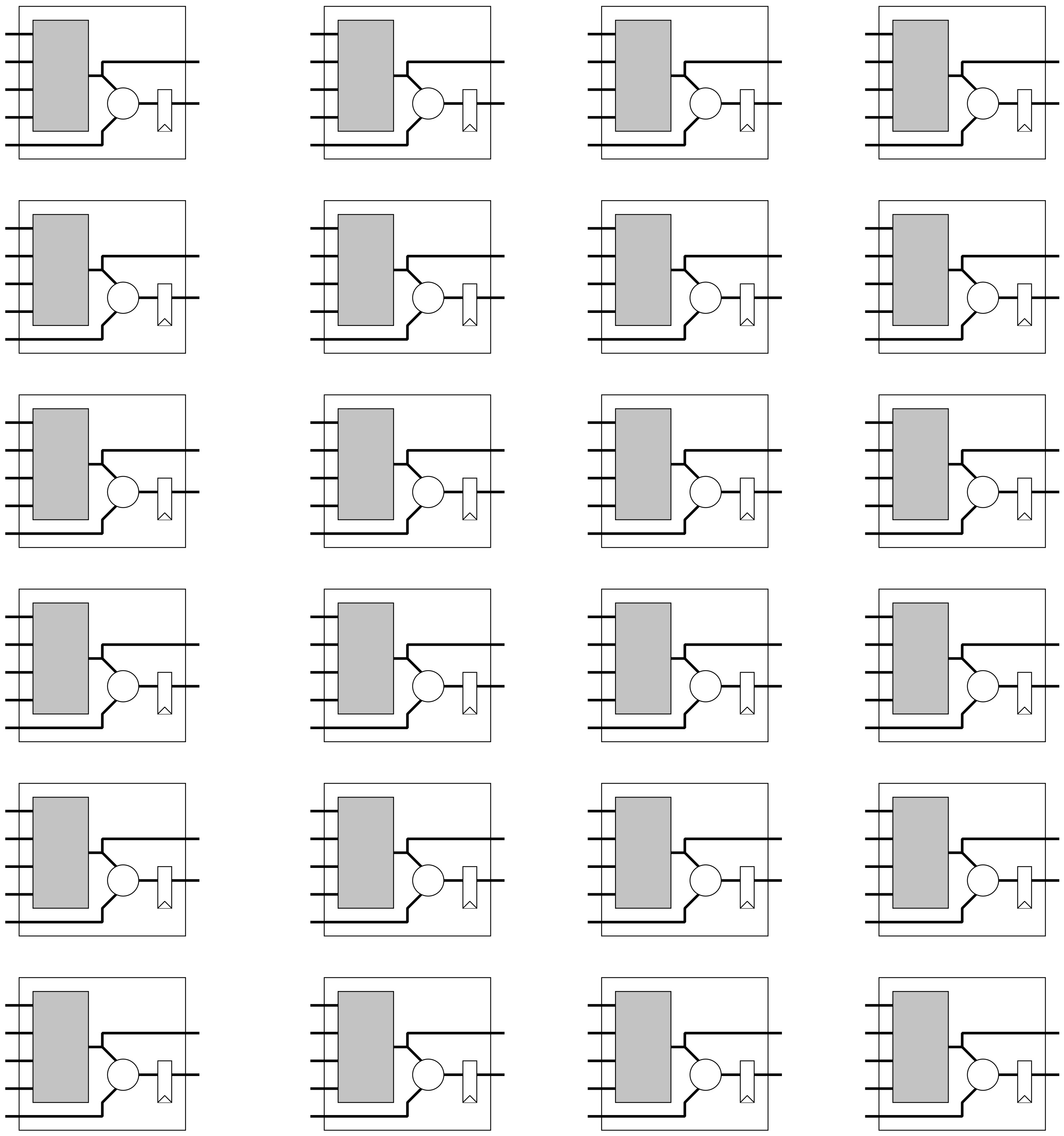
NOTES:

1. The primary inputs of the circuit are: `a`, `b`, `c`, `d`, `e`, and `f`.
2. The primary output of the circuit is: `z`.
3. Do not perform any logic optimizations.
4. For each FPGA cell that you use:
   - Label the input and output ports of the cell using the signal names from the gate-level circuit for ports that you use and **NC** (for no-connect) for ports that you do not use.
   - Show the configuration for the internal multiplexer.



Extra copy for scratch work





Total number of cells used: $\boxed{}$

Q3 (25 Marks) FSM

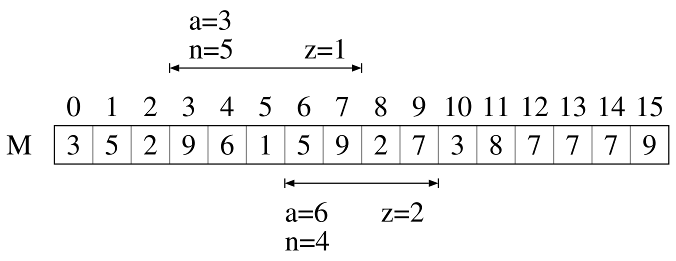
(*estimated time: 20 minutes*)

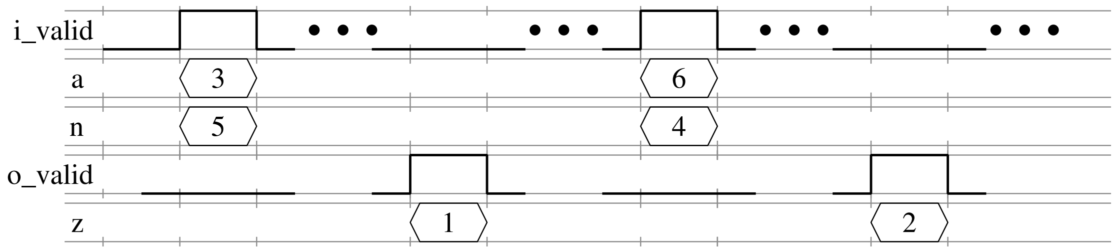
Design a state machine that finds the minimum value in a memory array.

NOTES:

1. The inputs to the system shall be `i_valid`, `a` and `n`.
2. The outputs shall be `o_valid` and `z`.
3. The system shall have an internal *single-port* memory array, named `M`.
4. The system shall set the output `z` to be the minimum value in `M` between addresses `a` and `a+n-1`, inclusive (*i.e.*, including `M[a]` and `M[a+n-1]`, see examples below.)
5. The environment will set the inputs `a` and `n` in the same clock cycle as `i_valid='1'`.
6. The environment will set `i_valid='1'` for exactly one clock cycle, then `i_valid` will remain `'0'` until after `o_valid='1'`. An unpredictable number of clock cycles after the system sets `o_valid='1'`, the environment will set `i_valid='1'`.
7. The system shall set `o_valid='1'` for exactly one clock cycle when it sets `z` to be the correct output value.
8. The system shall ignore the possibility of overflow.

Example:





Draw the state machine on the next page

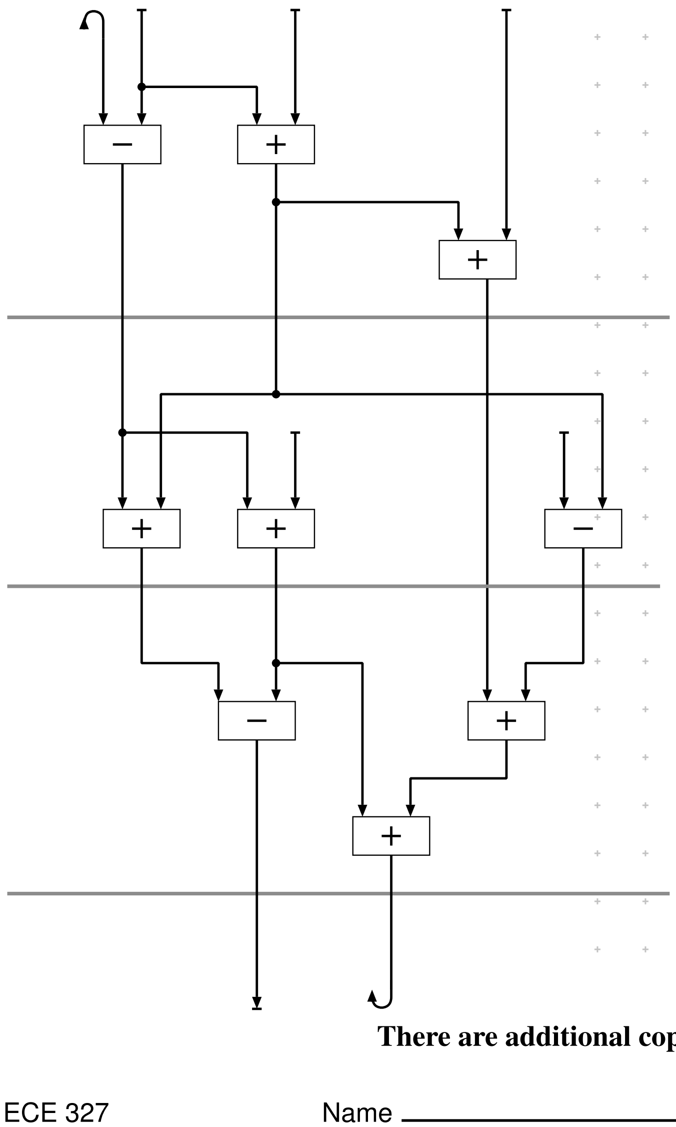
Q4 (25 Marks) DFD

(*estimated time: 20 minutes*)

For the dataflow diagram below: perform allocation, then draw the control table.

NOTES:

1. The system shall use a parcel schedule of "unpredictable number of bubbles".
2. You may use 2:1 multiplexers, and may combine 2:1 multiplexers to create larger multiplexers.
3. The optimization goals, in order of highest priority to lowest, are to minimize the number of:
   (a) adders and subtracters
   (b) input and output ports
   (c) registers
   (d) 2:1 multiplexers (including the 2:1 multiplexers used to build larger multiplexers)
   (e) chip-enables
4. In the control table, clearly show:
   - where *each multiplexer* and *chip-enable* is used
   - the *total* number of multiplexers and the *total* number of chip-enables used.
5. You may *not* perform scheduling optimizations on the dataflow diagram.
6. The *only* algebraic optimization that you may perform is *commutativity*.
7. You shall *not* perform don't-care instantiation



There are additional copies of the DFD on the next page