ECE 327 Final

2018t1 (Winter)

Instructions and General Information

- 100 marks total
- There are extra pages for scratch work at the end of the exam.
- If you need *additional scratch* paper, request some from a proctor. The work done on the *additional scratch* paper will not be marked. **All answers to be marked must be on the exam paper, which includes the pages at the end of the exam.**
- 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.
- To earn part marks, you must show the formulas you use and all of your work.

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 (6 Marks) Metrics

(*estimated time: 15 minutes*)

NOTES:

1. You do *not* need to justify your answers for this question.

Q1a (2 Marks) Optimizing Studying

You are trying to apply the principles you learned in ECE-327 to optimize your study plan for finals. You know that after some amount of studying, you begin to hit diminishing returns, in that you can put in much more time studying but earn only a few more marks. You want to maximize the efficiency of your studying while still earning a reasonable mark on the exam. Which one of the optimality measures below best captures your goal?

M	Mark you earn		$\frac{M}{t}$	$\frac{M^2}{t}$	$\frac{1}{(100\%-M)\cdot t}$	$\frac{M}{(100\%-M)\cdot t}$
t	Time you spend studying					
		Best:	☐	☐	☐	☐

Q1b (2 Marks) Battery Life

You are comparing several Waterluvian filters and want to choose the one that will process the most pixels on one battery charge. All of the filters use the same type and size of battery. Which one of the metrics below is best for this?

E	Energy					
P	Power		$\frac{E}{pix}$	$\frac{P}{pix}$	$\frac{E\cdot F}{pix}$	$\frac{P\cdot T}{pix}$
T	Throughput					
F	Clock frequency					
pix	Pixel	**Best:**	☐	☐	☐	☐

Q1c (2 Marks) Chip Temperature

You are comparing several Waterluvian filters and want to choose the one that will have the lowest temperature on the chip (*e.g.*, so that you can use a smaller and cheaper fan to cool it). Each filter is on a chip by itself: the only circuitry on the chip is the Waterluvian filter and the entire Waterluvian filter fits on one chip. Which one of the metrics below is best for this?

E	Energy					
P	Power		$\frac{E\cdot T}{A}$	$\frac{P}{A}$	$\frac{P\cdot T\cdot F}{A}$	$\frac{P\cdot T\cdot A}{F}$
T	Throughput					
F	Clock frequency					
A	Area of chip	**Best:**	☐	☐	☐	☐

Q2 (20 Marks) The Spring, the Chirping Birds, the Snow

(*estimated time: 20 minutes*)

For each of the code fragments Q2a–Q2d:

1. Answer whether the code is *legal*
2. If the code is *illegal*: explain why, and proceed to the next code fragment.
3. Answer whether the code is *synthesizable*.
4. If the code is *unsynthesizable*: explain why, and proceed to the next code fragment.
5. Answer whether the code adheres to good coding practices, according to the guidelines for ECE 327.
6. If the code does *not follow good coding practices*: explain why.

NOTES:

1. All signals are `std_logic`.

Q2a

```
process (a, b)
begin
  if e = '1' then
    d <= a;
  else
    d <= b;
  end if;
end process;

process begin
  wait until rising_edge( clk );
  e <= d;
end process;
```

	Yes	No
Legal	☐	☐
Synthesizable	☐	☐
Good Practice	☐	☐

Explanation if illegal, unsynthesizable, or bad practice:

Q2b

```
process begin
  wait until rising_edge( clk );
  if reset = '1' then
    d <= '0';
  end if;
end process;

process begin
  wait until rising_edge( clk );
  if reset /= '1' then
    d <= a;
    e <= d xor e;
  end if;
end process;
```

	Yes	No
Legal	☐	☐
Synthesizable	☐	☐
Good Practice	☐	☐

Explanation if illegal, unsynthesizable, or bad practice:

Q2c

```
process ( i_valid ) begin
  if rising_edge( i_valid ) then
    b <= i_data;
  end if;
end process;

process begin
  wait until rising_edge( clk );
  c <= b xor c;
end process;
```

	Yes	No
Legal	☐	☐
Synthesizable	☐	☐
Good Practice	☐	☐

Explanation if illegal, unsynthesizable, or bad practice:

Q2d

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

f <=   e when (d xor e) = '1'
  else a xor b;
```

	Yes	No
Legal	☐	☐
Synthesizable	☐	☐
Good Practice	☐	☐

Explanation if illegal, unsynthesizable, or bad practice:

Q3 (16 Marks) Area Analysis

(*estimated time: 20 minutes*)

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

NOTES:

1. The signals `a`, `b`, `c`, `d`, `e`, `f`, `g`, `y` and `z` are all declared to be `signed( 15 downto 0 )`.
2. The signal `s` is declared to be `std_logic`.
3. The inputs are `a`, `b`, `c`, `d`, and `s`.
4. The outputs are `y` and `z`.
5. Optimizations are allowed, as long as the externally visible input-to-output behaviour of the system does not change.
6. For full marks, you must justify your answer with a drawing and/or text.

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

y <= g;
z <= f + d;
```

Number of FPGA cells:

Q4 (20 Marks) DFD

(*estimated time: 30 minutes*)

Design and analyze a dataflow diagram that implements the pseudocode below.

```
for idx in 0 to 255 {
  z = z - a + b + c + 123;
}
```

NOTES:

1. Read the expression for `z` carefully. It is *not*: `z - (a + b + c + 123)`.
2. `a`, `b`, and `c` are inputs.
3. `idx` is an interparcel varaible.
4. `z` is an interparcel variable and an output.
5. Optimization goals, in order of decreasing importance
   - Minimize clock period
   - Maximize throughput, up to a maximum of 1/2 parcels/clock-cycle. (There is no benefit, and no additional marks will be given, for achieving a throughput that is greater than 1/2.)
   - Minimize number of input ports
   - Minimize total number of datapath components (sum of number of adders and number of subtracters)
   - Minimize number of registers
   - Minimize latency
6. Your DFD *shall* show the increment of `idx` and the computation of `z`.
7. Your DFD shall *not* show the initialization of `idx` or the comparison of `idx` to detect the end of 256 iterations of the loop.
8. Your DFD shall *not* show the initialization of `z`.
9. You may ignore the area consumed by multiplexers.
10. You may schedule the input values to arrive in any order within an iteration.
11. You do *not* need to do any allocation.
12. Algebraic optimizations are allowed, so long as the externally visible functionality remains unchanged.

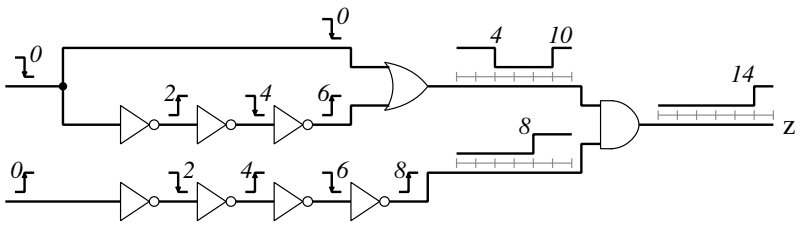
Clock period	
Throughput	
Input ports	
Adders	
Subtracters	
Total datapath	
Registers	
Latency	

Q5 (16 Marks) Monotone Speedup

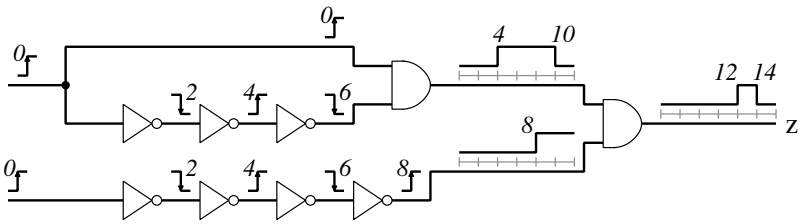
(*estimated time: 20 minutes*)

Use *one* of the excitations and circuits below to demonstrate a problem that can occur if your critical path algorithm does not satisfy the monotone speedup correctness criterion.

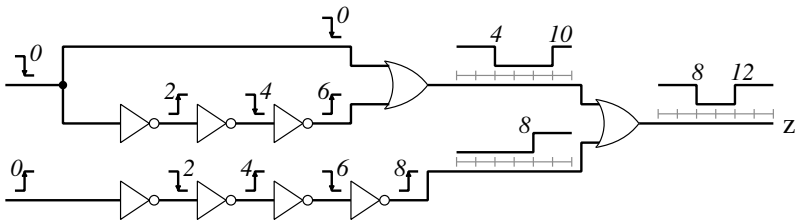
A



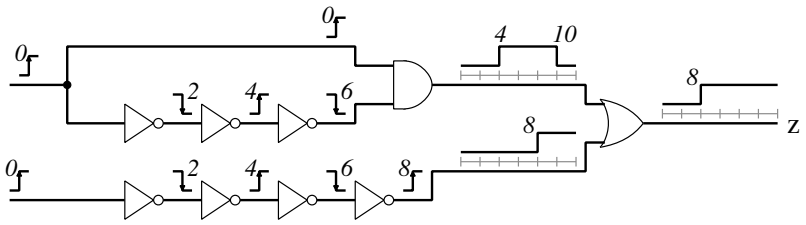
B



C



D

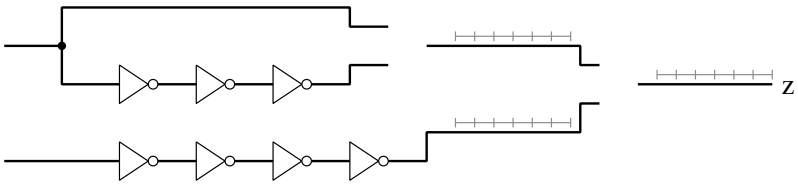


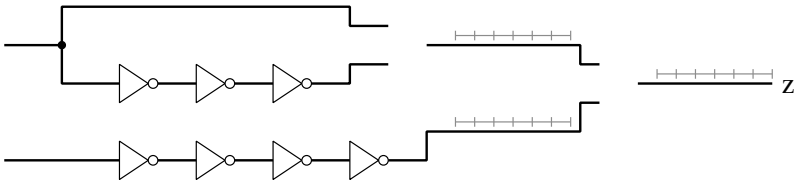
List any assumptions you use:

Circuit and excitation to illustrate problem:

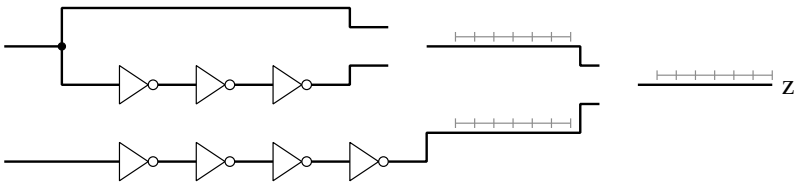
NOTES:
1. In the circuit below, draw the missing gates from the circuit you chose.
2. Draw the modified behaviour to demonstrate the problem.
3. Explain the problem.

Modified behaviour:





(Extra copy if needed)



(Extra copy if needed)

Explanation:

Q6 (22 Marks) Power and Delay

(*estimated time: 30 minutes*)

You work for Pbπ, the *lead*ing developer of *pipe*lined Waterluvian filters. The CEO, Cassandra Scarlet, has promoted you to the top-secret R&D lab, codenamed "The Library". Dr. Scarlet has asked you to investigate the potential use of "approximate computing" in Waterluvian filters.

You are clueless about approximate computing, so you do some web surfing and learn that approximate computing is an optimization that can be used to increase performance or decrease power in digital hardware systems. The disadvantage is that the system will sometimes compute an incorrect result. A typical application is image processing, where users usually will not notice if a few pixels are incorrect.

Approximate computing works by decreasing the supply voltage and/or decreasing the clock period so that the delay through the combinational logic in some parts of the system is sometimes longer than the clock period. When this happens, some flip-flops might store incorrect values.

Your task is to evaluate the effect on power and energy if approximate computing is applied to Pbπ's Waterluvian filter.

The Pbπ Waterluvian filter currently has the following parameters:

	Current Values		
Supply voltage	1.5	V	
Threshold voltage	0.7	V	
Clock period	4.0	ns	
Dynamic power	15	mW	
Static power	2	mW	
Performance	100	MPPS	(mega-pixels per second)

Dr Scarlet has asked you to analyze the effects of the following new values with the same circuit:

	New Values	
Supply voltage	1.3	V
Clock period	3.5	ns

NOTES:

1. Short circuiting power is negligible.

List any assumptions you use:

Q6a (12 Marks) Power

Calculate the new total power consumption of the Waterluvian filter

Total power:

The exam continues on the back side of this page

Q6b (10 Marks) Energy

Calculate the new energy per pixel of the Waterluvian filter.

Energy per pixel:

This page is for scratch work for any question

This page is for scratch work for any question