

MASSACHUSETTS INSTITUTE OF TECHNOLOGY  
DEPARTMENT OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

**6.375 Complex Digital Systems**  
**Spring 2008 - Quiz - March 21, 2008**  
**80 Minutes**

NAME: \_\_\_\_\_ SCORE: \_\_\_\_\_

Please write your name on every page of the quiz.

Not all questions are of equal difficulty, so look over the entire quiz and budget your time carefully.

Please carefully state any assumptions you make.

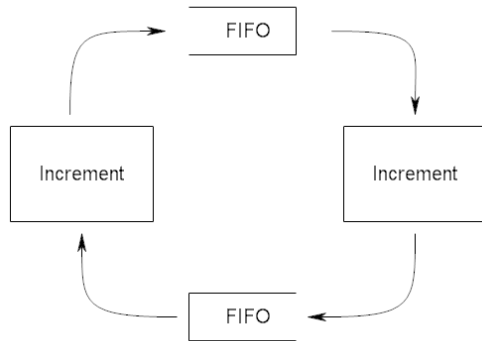
Enter your answers in the spaces provided below. If you need extra room for an answer or for scratch work, you may use the back of each page but please *clearly indicate where your answer is located*.

**You must not discuss the quiz's contents with other students who have not yet taken the quiz. If, prior to taking it, you are inadvertently exposed to material in a quiz — by whatever means — you must immediately inform the instructor or a TA.**

	Points	Score
Problem 1	20	
Problem 2	25	
Problem 3	25	
Problem 4	10	
Problem 5	20	

**Problem 1 : Throughput (20 total points)**

Ben Bitdiddle made a dual upcounter:



using the following code:

```

module temp();
  Reg#(Bit#(1)) stage <- mkReg(0);
  FIFO#(int)    fifoA <- mkFIFO1();
  FIFO#(int)    fifoB <- mkFIFO1();
  rule init (stage == 0);
    fifoA.enq(1);
    fifoB.enq(1);
    stage <= 1;
  endrule
  rule inc1 (stage == 1);
    let temp = fifoA.first();
    fifoA.deq();
    $display("Inc1: %d", temp);
    fifoB.enq(temp+1);
  endrule
  rule inc2 (stage == 1);
    let temp = fifoB.first();
    fifoB.deq();
    $display("Inc2: %d", temp);
    fifoA.enq(temp+1);
  endrule
  rule exit ((fifoA.first() == 6) || (fifoB.first() == 6));
    $finish();
  endrule
endmodule

```

He was expecting to see the following display:

Inc1: 1 Inc2: 1 Inc1: 2 Inc2: 2 Inc1: 3 Inc2: 3 Inc1: 4 Inc2: 4 Inc1: 5 Inc2: 5

**1.1 (10 points)**

The code compiled, but on simulation, he did not see any display statements. Why is the code not executing correctly?

**1.2 (10 points)**

Modify the code to get the desired execution. You can use library elements such as those used in the labs.

**Problem 2 : Bluespec Semantics (25 total points)**

Consider the code given below:

```
module sem();
  Reg#(int)    count <- mkReg(1);
  Reg#(int)    a <- mkReg(1);
  Reg#(int)    b <- mkReg(2);
  Reg#(int)    c <- mkReg(3);
  rule counter (True);
    count <= count + 1;
  endrule
  rule mod1 (True);
    a <= b + c;
  endrule
  rule mod2 (True);
    b <= c + count;
  endrule
  rule mod3 (True);
    c <= b + count;
  endrule
  rule exit (count >= 4);
    $finish();
  endrule
endmodule
```

**2.1 (6 points)**

What are the sequential composability conditions deduced by the compiler?

**2.2 (9 points)**

Using the above conditions, assume an overall order and determine the values of all the state elements at finish.

**2.3 (10 points)**

Modify the code such that all the rules fire every cycle in the following order:

```
counter | mod1 < mod2 < mod3 < exit
```

You can use EHRs of any order.

**Problem 3 : Bluespec Synthesis (25 total points)**

Consider the code shown below:

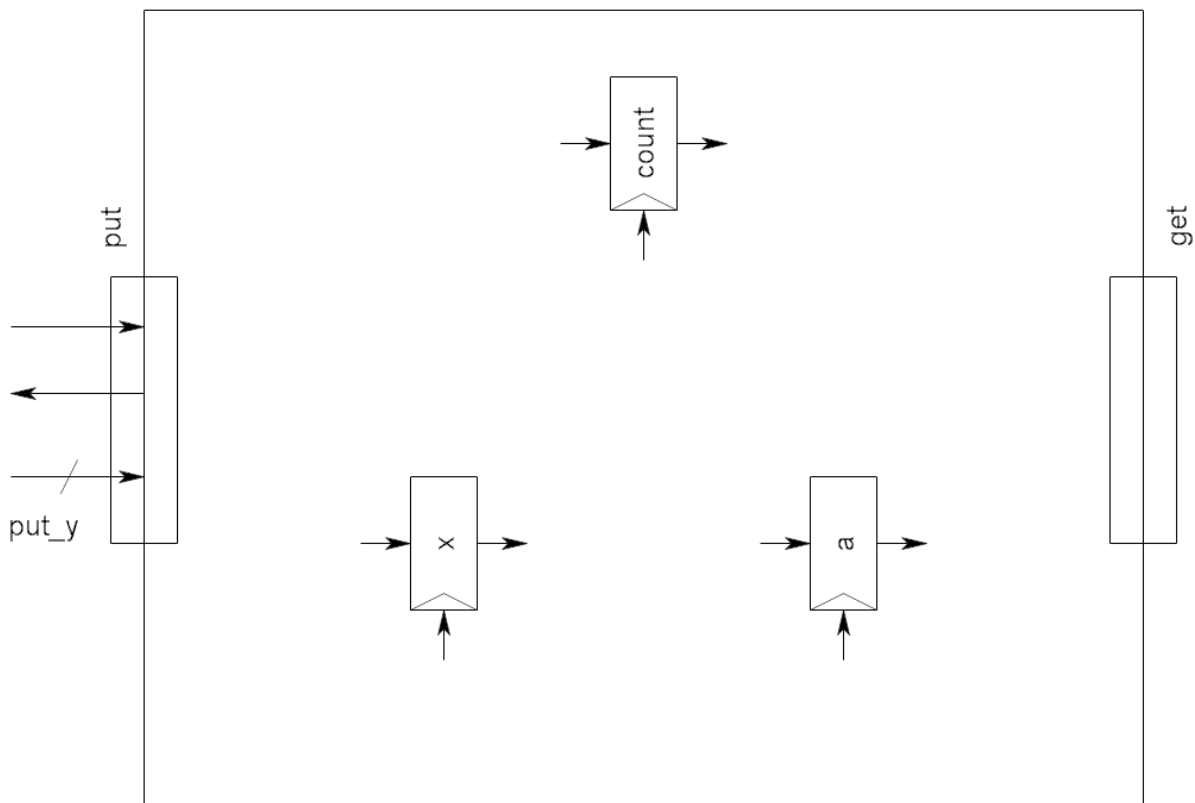
```

module mkMultBySixDyn(Foo#(int));
  Reg#(int) a      <- mkReg(0);
  Reg#(int) x      <- mkReg(0);
  Reg#(int) count <- mkReg(0);
  rule mulDyn (count>0 && count<6);
    count <= count+1;
    a     <= a+x;
  endrule
  method Action put (int y) if (count==0);
    a <= y; x <= y;
    count <= 1;
  endmethod
  method ActionValue#(int) get if (count==6);
    count <= 0;
    return a;
  endmethod
endmodule

```

**3.1: 15 points**

Sketch the hardware produced on compiling this code. Label the interface signals, scheduling logic and signals corresponding to CAN\_FIRE\_mulDyn and WILL\_FIRE\_mulDyn.



**3.2: 10 points**

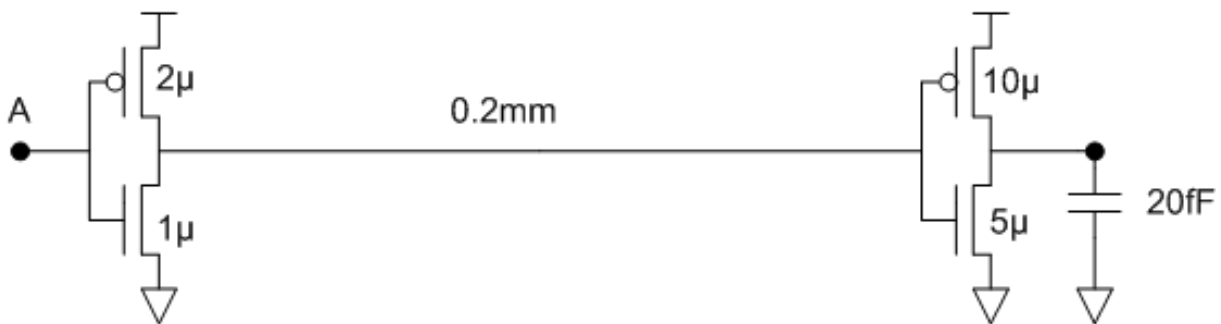
The rule mulDyn is replaced by the following rule:

```
rule mulStat (count>0 && count<6);
  for(int i = 1; i<6; i++)
    if(count==i)
      begin
        count <= i+1; a <= a+x;
      end
endrule
```

How does this change affect the hardware generated? Which implementation mulDyn or mulStat has more adders? More muxes? Compare the overall area and critical paths of the implementations.

**Problem 4 : RC Delay (10 points)**

Assume you have an inverter (nmos width =  $1\ \mu\text{m}$ , pmos width =  $2\ \mu\text{m}$ ) driving an interconnect of length  $0.2\ \text{mm}$  as shown in the figure. The interconnect has an inverter (nmos width =  $5\ \mu\text{m}$ , pmos width =  $10\ \mu\text{m}$ ) at the other end which drives a flipflop whose input capacitance is  $20\ \text{fF}$ . Calculate the delay of the circuit from point A to point B (see figure). You can use a  $\pi$  model (as shown in the attached slide) for the bitline and assume that the transistors turn on after one RC time constant. The process parameters are given in the table below.



Process Parameters	Value
PMOS gate capacitance per $\mu\text{m}$ of transistor width	$1.5\text{fF}/\mu\text{m}$
NMOS gate capacitance per $\mu\text{m}$ of transistor width	$1.5\text{fF}/\mu\text{m}$
PMOS drain capacitance per $\mu\text{m}$ of transistor width	$0.3\text{fF}/\mu\text{m}$
NMOS drain capacitance per $\mu\text{m}$ of transistor width	$0.3\text{fF}/\mu\text{m}$
PMOS effective on resistance	$6.6\text{k}\Omega\mu\text{m}$
NMOS effective on resistance	$3.3\text{k}\Omega\mu\text{m}$
Metal 2 wire resistance per $\mu\text{m}$ of length	$0.4\Omega/\mu\text{m}$
Metal 2 wire capacitance per $\mu\text{m}$ of length	$0.2\text{fF}/\mu\text{m}$



**Problem 5 : Power (20 total points)**

The following two unit designs implement the same signal processing function, with the following performance characteristics.

Unit	Throughput (million tasks/sec)	Vdd (volts)	Energy/Task (nanojoules)
Unit 1	20	1.2	5
Unit 2	10	1.2	3

The following table lists the effect of changing supply voltage on circuit delay and energy per operation, **normalized** to that at 1.2V.

Vdd	Delay	Energy/Task
1.5	0.7	1.9
1.4	0.8	1.5
1.3	0.9	1.2
<b>1.2</b>	<b>1.0</b>	<b>1.0</b>
1.1	1.2	0.8
1.0	1.5	0.6
0.9	2.0	0.4
0.8	10.0	0.3
0.7	non-functional	non-functional

a) Which unit is the most energy efficient for a minimum throughput of million 12.5 tasks/second? Show your work. **6 points**

b) Which unit gives the highest performance at a maximum power dissipation of 30mW? Show your work. **6 points**

c) Assume the signal processing function is perfectly parallelizable. What is the lowest power parallel configuration to process 20 million tasks/second? Ignore the area cost. List which unit is used, the operating voltage, and the number of parallel instances. **8 points**