ECE 464-001 / 520-001
Midterm, 2012

Name: ______________________
Student Number: ___________
Course/Section: _____________

Each question is worth 2 points. The correct answer earns 2 points, the incorrect 0. Some questions have space for a short explanation. If your answer is incorrect, we will look at this for potential partial credit.

The exam is open book, open notes. No computers are permitted. **75 minutes.**

Write your name and student number on top of this sheet of paper and turn that in. **Take care answering these questions. Work carefully through your answers and check them before turning them in. Silly errors can cost you a lot in a multiple choice exam. Don’t rush the test.**
**Question 1**
What statement best explains the impact of Moores law on ASIC-based products?

A. Every 18 months the cost of new consumer products will approximately halve due to the decline in transistor cost.
B. Every 18 months the power consumption of new consumer products will approximately halve due to the reduction in transistor parasitic capacitance.
C. Every 18 months the clock speed used in consumer products doubles due to the improvement in transistor performance.
D. Every 18 months the capability of consumer products increases due to the ability to deliver more performance at the same cost and power level.
E. None of the above are true.

Brief Explanation (optional): D. Note D is more correct than C. Hence D is the accepted answer. The question asks for the “best” answer.

**Question 2**
You are designing a digital chip set intended to be used for drug discovery. Performance is important but the computations can be done in parallel. You are only going to build 10 such units, and the algorithms to be employed change with time. Which ASIC implementation style would you use?

A. Full custom design.
B. Standard Cell ASIC.
C. Gate Array.
D. FPGA or a set of FPGAs
E. None of the above

Brief Explanation (optional): (By the way this is a real example.) D
Question 3
Consider the following code (implemented exactly as described):

```verbatim
reg A, B, C, D;
always@(posedge clock)
    begin
    A <= C;
    B <= D ? ((A | C) & B) : B;
    C <= B;
    D <= C;
    end
```

Sketch the logic.

Questions 4, 5, 6 (Use logic in Q3)
The minimum and maximum delays between each set of successive gates are marked as 
#(min: typical: max) ns and are marked on the output node of the driving gate. 
(Remember the timing equations are ≤ and ≥ constraints). You also need the 
following:
- T_setup = #(1 : 1 : 2) ns.
- T_hold = #(1: 1 :1) ns.
- T_skew = #(1: 1 : 1) ns for the clock.
- T_clock-Q = #(2 : 3 : 5) ns.
- T_logic = #(1 : 2 : 3) ns for each and every logic gate from any input to any 
output.

Question 4
What is the fastest possible clock period that satisfies setup constraints across all 
variations?
A. 5 ns
B. 14 ns
C. 17 ns
D. 18 ns
E. None of the above are correct.

Calculation (optional)

C
Question 5
Is there a potential hold violation?
A. Yes, the fastest logic is at least 2 ns too fast.
B. Yes, the fastest logic is 1 ns too fast.
C. No, but there is zero margin
D. No, there is a 1 ns safety margin.
E. No, there is at least a 2 ns safety margin.
F. No, there is at least a 2 ns safety margin.

Calculation (optional)

C

Question 6
Why is cycle stealing of limited value?

Can only steal a total of t_ck_high over the entire pipeline. Note this question was meant to be general, not referring to the HW above.

Question 7
What does the following code block build? Sketch your answer.

```verilog
reg B;
always@(*)
begin
  casex(A)
    2'b00 : B = C;
    2'b01 : B = D;
    default : B = E;
  endcase

Mux, inputs 2 and 3 = E
```
**Question 8**

Consider the following code. It is the only code in the module.

```verilog
always@(posedge clock)
begin
    A <= C & D;
    B <= C | D;
end
assign E = A + B;
```

If it is synthesized with the following constraints

```verilog
Create_clock -period 8 -waveform {0 4} -name clock
set_clock_skew -uncertainty 1.0 clock

set_input_delay 1.0 -clock clock all_inputs() - clock
set_output_delay 2.0 -clock clock all_outputs() - clock
```

Flip flop tck-Q delay is 1 ns, setup time is 1 ns, and hold time 1 ns, all logic gate delays vary between 1 and 2 ns.

What is the maximum allowed delay for the logic A+B?

A. 3 ns
B. 4 ns
C. 5 ns
D. 6 ns
E. None of the above

Calculations (optional):

B

**Question 9**

You have to tell another designer how to set the output_delay in her script so that it can properly interface to the C and D inputs of your design. What script element do you give your coworker?

```verilog
Set_output_delay 3 -clock clock all_outputs() - clock
```
Question 10
In the following test fixture, what is the timing diagram being specified?

```plaintext
initial
  dec = 0; run = 0; ck = 0;
  #21 run=~dec;
  #10 run = ~dec;
  #10 $finish;
end
always #5 ck = ~ck;
always #10 dec=~dec;
```

<table>
<thead>
<tr>
<th></th>
<th>10</th>
<th>20</th>
<th>30</th>
<th>40</th>
</tr>
</thead>
<tbody>
<tr>
<td>ck</td>
<td>00001111000011110000111100001111</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>run</td>
<td>0000000000000000000011111111100000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>dec</td>
<td>00000000111111111000000001111111</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

```
**Question 11**  
In the class synthesis script, what is the difference between `compile` and `compile -incremental`?

Use `compile` for the first optimization run, use `–incremental` thereafter.

**Question 12**  
Rewrite the following code fragment using blocking assignment, so that it is correct.

```verilog
define reg A, B, C;
define always@ (posedge clock)  
begin  
A <= B;  
C <= A;  
end

C = A;  
A = B;
```