**Print:**

First Name:. . . . Solutions . . . . . . Last Name:. . . . . . . . . . . . . . . . . . . . . . . . . . . .

Student Number:. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

# University of Toronto
# Faculty of Applied Science and Engineering

Test II – April 13, 2009
BA 2165

ECE532S – Digital Systems Design

Examiner – Paul Chow

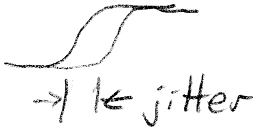
1. There are 4 questions and **10** pages. Do **all** questions. The total number of marks is 50. The duration of the test is 50 minutes.
2. **ALL WORK IS TO BE DONE ON THESE SHEETS!** Use the back of the pages if you need more space. Be sure to indicate clearly if your work continues elsewhere.
3. **Faculty approved calculators are permitted.**
4. Closed book. No aids permitted.

| | |
|---|---|
| 1 [10] | |
| 2 [ 8] | |
| 3 [20] | |
| 4 [12] | |
| Total [50] | |

1. Starting with some short answer questions:

[2 marks] (a) What is *jitter* in the context of timing analysis for digital signals.

Uncertainty in placement of a clock edge from cycle to cycle



[2 marks] (b) What is *static timing analysis*?

Checking timing by tracing through circuit netlist

[2 marks] (c) What happens during the *place and route* phase of an FPGA CAD flow?

placement : logic gates + FF's are assigned specific sites in FPGA

route: wires are selected and switches set to connect logic

[2 marks] (d) What is the difference between a *blocking* and *non-blocking* statement in Verilog?

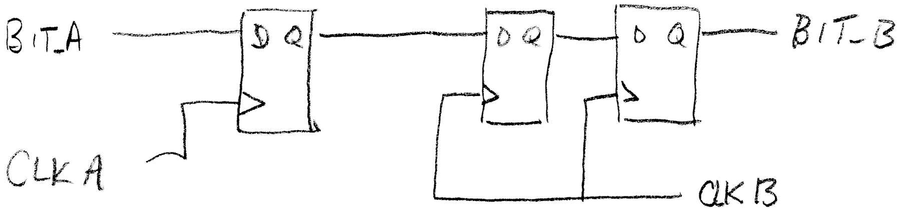
Blocking: Simulation is blocked until RHS is evaluated and assigned

Non-blocking: LHS is not updated until the end of the always block.

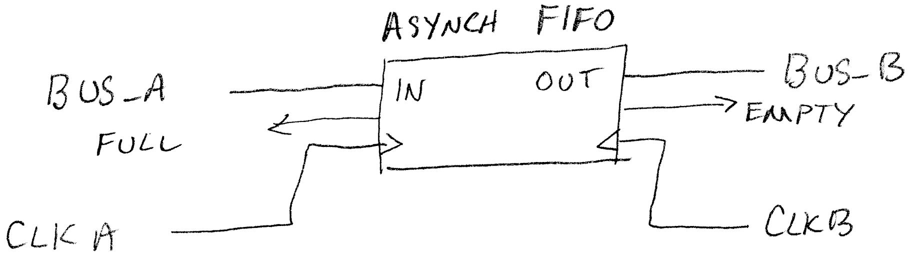
[2 marks] (e) What is the function of a *clock tree*?

Distribute clock across the chip in a low skew manner and drive the high fanout

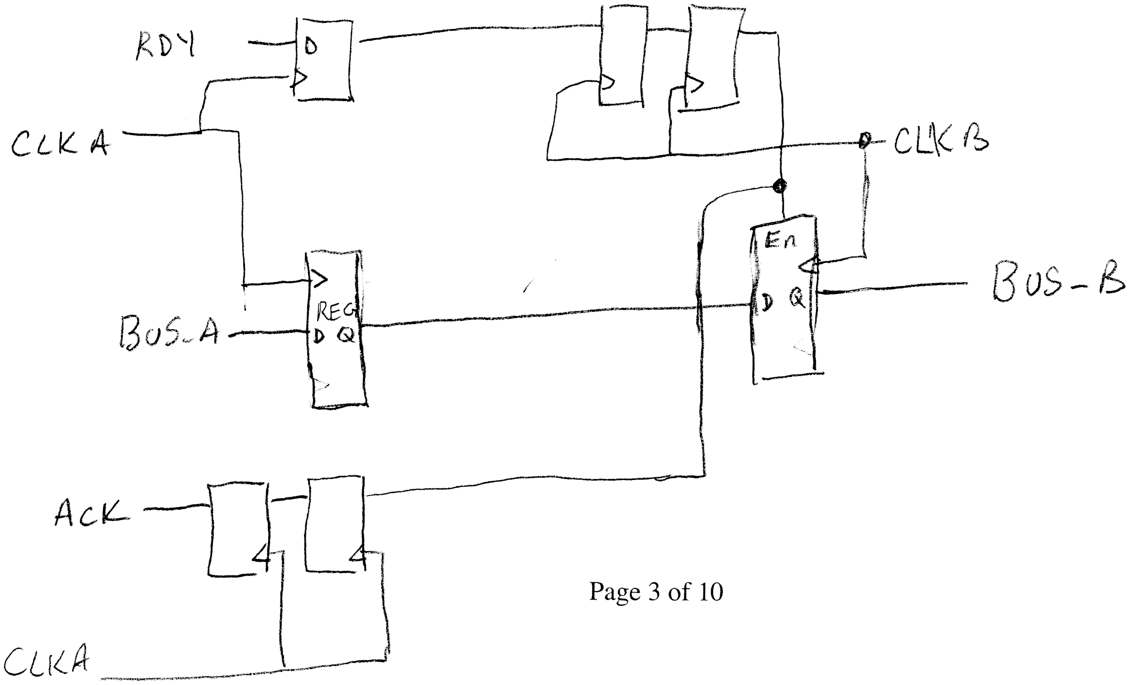
[4 marks] 2. (a) Show a synchronizer for transferring a single bit single across a clock domain.



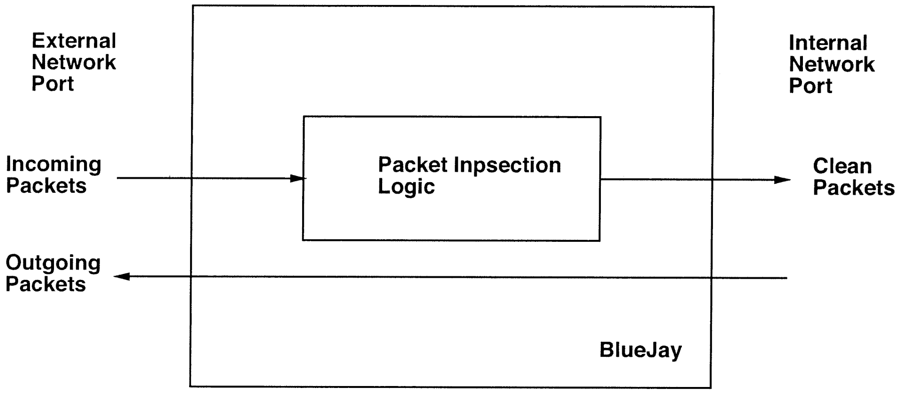
[4 marks] (b) Show a synchronizer for transferring data on a bus across a clock domain.



— OR —



[20 marks] 3. You have been made chief architect of a new product at Disco Networking Systems. The product, code named BlueJay, is to be used to inspect network packets looking for virus signatures. BlueJay will be installed inline with a network connection so that all network traffic entering a site can be inspected. All network traffic leaving a site will pass through BlueJay untouched. The figure below shows the high-level functionality required.



BlueJay will have two ethernet connections: an *External* port for connecting to the outside world and an *Internal* port for the internal network. The ethernet should be capable of running up to the 1 GigE standard (Gigabit Ethernet). Incoming packets are buffered in memory by one process. A separate process removes the packets from memory and runs them through the packet inspection logic.

Assume that the packet inspection logic is an IP core that you can obtain. It takes as input a packet streamed on a 32-bit wide bus. The output of the packet inspection logic has two 32-bit buses. One is a *clean* stream and the other stream is for *infected* packets.

The *clean* stream is sent on to the internal network and the *infected* stream is saved in memory for later processing.

Your first task is to define the architecture of the system at a block diagram level. You will be designing your own board so you have complete control over the components that will be used. You choose to use a Xilinx FPGA.

You also know that it is important to be able to easily debug and control your system, so you will include a processor and a means for it to get input from a keyboard and print messages.

As with all initial design specifications, there is probably not enough information. Under normal cases, you should be asking more questions. In this case, you cannot do this so clearly state any assumptions you need to make.

Draw the block diagram of your proposed system with enough detail so that it could be presented at a design review of your peers where they can provide comments.

Your diagram must show all of the necessary components, including the external chips, how the components are connected and how they are implemented (indicate clearly whether a component is in FPGA logic, an external chip, etc.). To help the reviewers understand the design environment, you should include how the board is connected to a development PC and how the tools interact with the board.

Question 3 continued. . .

For each block in your diagram, briefly explain the function of the block and whether it is custom-designed logic or a block that you know is available from the EDK library of IP. If the block is custom, indicate whether you have to write it from scratch or whether there is a tool that can help you.

The evaluation of your solution will also be based on how easy it is to understand! Clarity of presentation is very important.

There is more space on the next two pages to answer this question.

Bandwidth check:

1 Gbps = 125 MB/s

Worst case: 1 Gbps in and 1 Gbps out

Processor Bus @ 100MHz & 32 bits = 400 MB/s

1 Processor is not going to be able to keep up with all the data transfers required using a single bus

Assuming use of DMA, one PLB bus and one processor could handle each interface

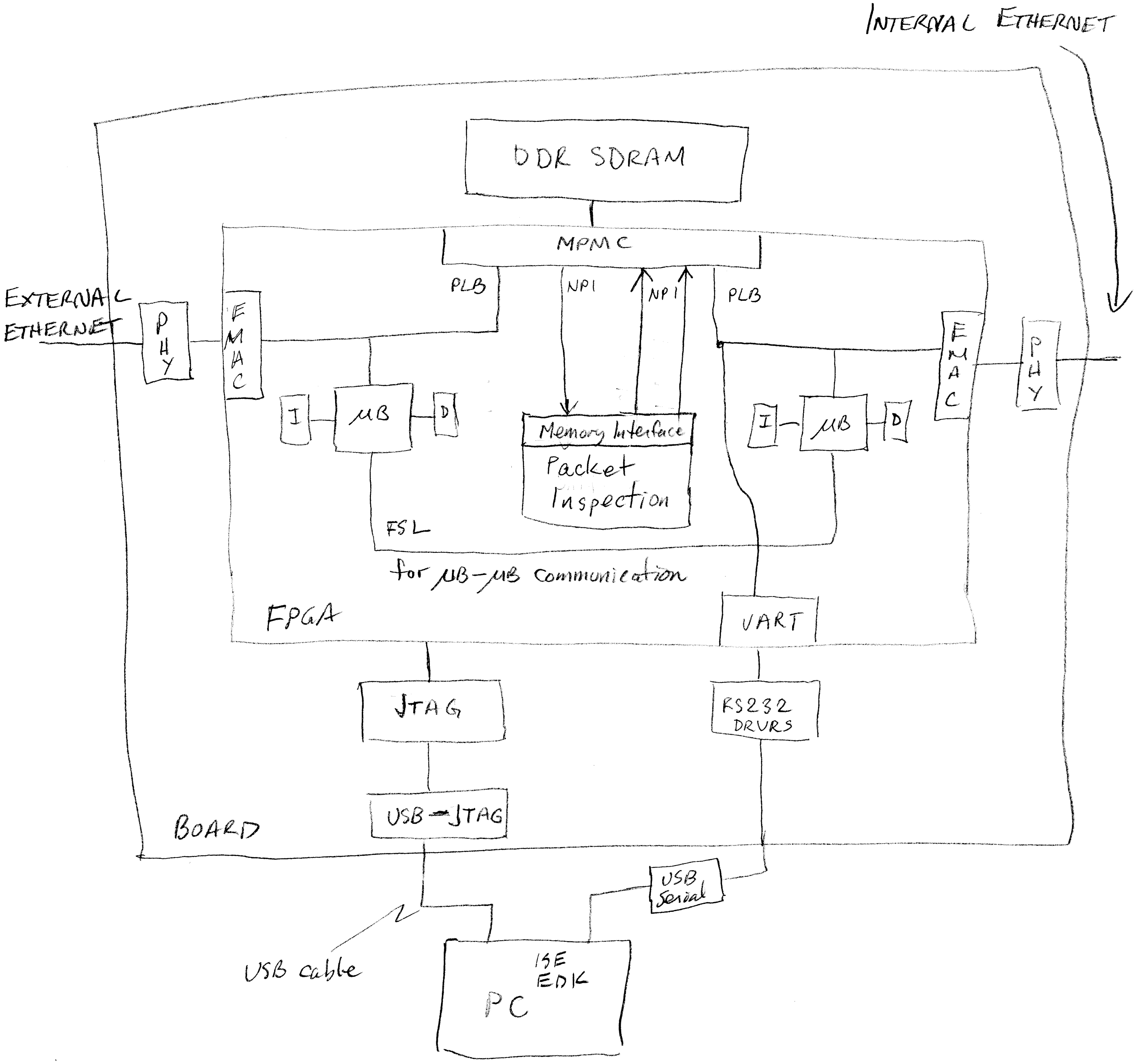
Packet Inspect talks directly to memory with one input port and one output port

Assume for simplicity, packet inspection writes all packets, infected and clean to memory. The output processor will read the clean packets and send them out the interface.

SDRAM: Assume 200 MHz DDR, 32-bit bus (4 bytes)

⇒ 1.6 GB/s - lots of bandwidth

Question 3 continued...



Question 3 continued. . .

EMAC - ethernet MAC from EDK library

PHY - ethernet physical interface

μB - MicroBlaze from EDK library, I+D memory using BRAM and local memory bus interface - rest of memory in DDR

FSL - connects two μB together

MPMC - Multi-port memory controller from EDK

UART - from EDK for serial communication

RS232 Drivers - serial port physical interface

DDR SDRAM - external memory for packets & MicroBlaze

Packet Inspection - Core that checks packets

Memory Interface - Interface to MPMC for packet inspection core

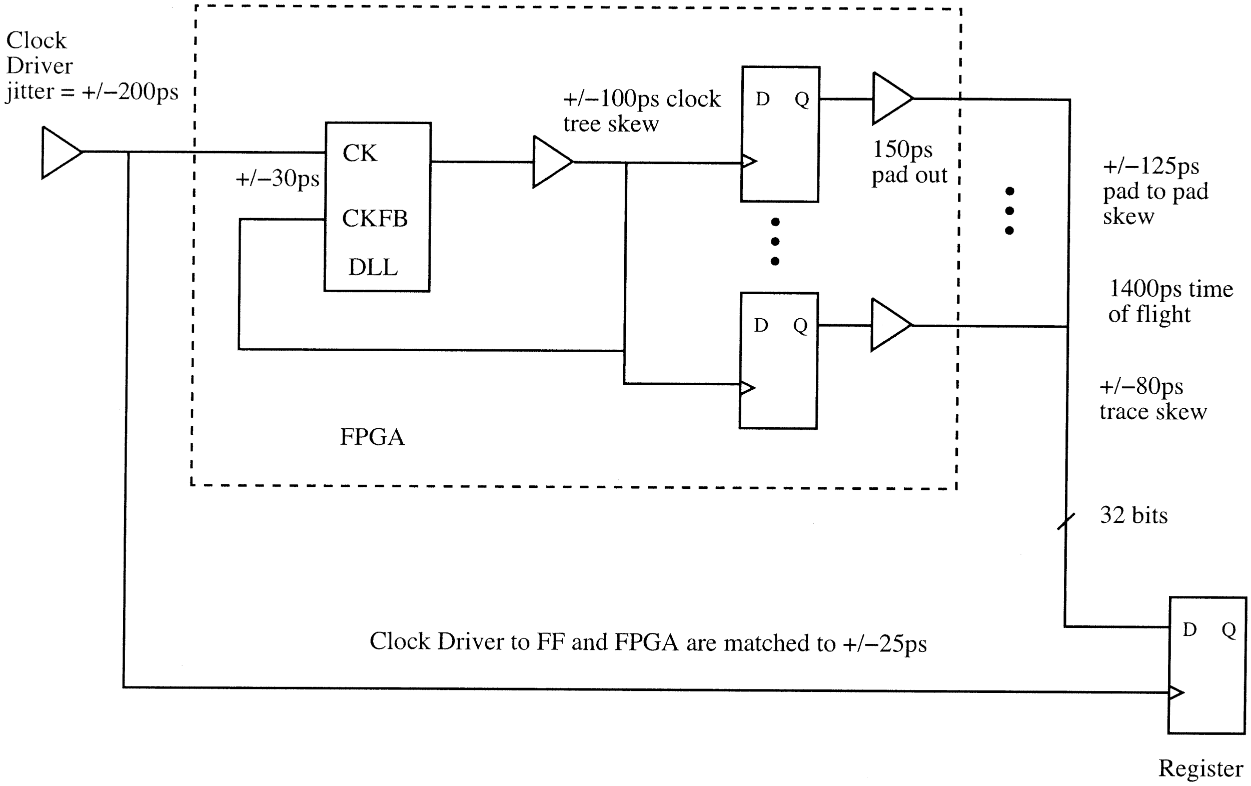
JTAG - JTAG interface to FPGA

USB-JTAG - USB to JTAG interface

USB-Serial - USB to Serial converter

PC - runs ISE, EDK, Hyperterminal

4. Consider the circuit shown below:



From the data sheets you have also gathered the following information about the flip flops:

| | FPGA | External |
|---|---|---|
| $t_{setup}$ | 0.8 ns | 1.0 ns |
| $t_{hold}$ | -0.4 ns | 0.5 ns |
| $t_{ClockToQ}$ | 0.2 ns | 0.3 ns |

Question 4 continued...

You are required to do a timing analysis. Note that unlike the analysis done in class where the uncertainty numbers were conservatively doubled, you are asked to be more careful here to determine the best possible clock frequency.

Assume the rise time is 1.0 ns.

[2 marks] (a) What is the delay from the rising clock edge at a transmitting flip flop to the data arriving at the receiving flip flop? Do not consider any of the uncertainties or rise time. Clearly indicate which components you are including.

FPGA FF Clock Q + pad out + time of flight

$0.2 + 0.15 + 1.4 = 1.75$

Delay = 1.75 ns

[2 marks] (b) Assuming an ideal system where the uncertainties are zero and rise time is zero, what is the minimum clock period?

Ideal Case Minimum Clock Period = 2.75 ns

$\text{Delay} + t_{setup} = 1.75 + 1.0$

$= 2.75 \text{ ns}$

[2 marks] (c) When analyzing for the minimum clock period in the presence of uncertainties, what is the assumption about the placement of the transmitting and receiving clock edges? i.e., what directions (early/late) do they move in the worst case for setup time? Explain your answer.

Transmitting edge is = late

Receiving edge is = early

Shortens clock period giving less time for signal to stabilize

Question 4 continued...

[2 marks] (d) What is the minimum clock period required when accounting for any non-ideal factors? Be sure to show what components you are including and how you include them.

Minimum Clock Period = 4.31 ns

Need to increase clock period by the amount the uncertainty can shorten it. Also account for rise time

Transmit path = Clk driver + DLL + clock tree + pad to pad + trace skew + $t_{rise}$
= 0.2 + 0.03 + 0.1 + 0.125 + 0.08 + 1.0
= 1.535 ns

Receive path = 0.025 ns

Min = 2.75 + 1.535 + 0.025
= 4.31 ns

[2 marks] (e) When checking to see if the hold time is met, what is the assumption about the placement of the transmitting and receiving clock edges? i.e., what directions (early/late) do they move in the worst case? Explain your answer.

Transmitting edge is = early

Receiving edge is = late

Widen clock period -
Gives more time for signal to change, reducing time available for hold time

[2 marks] (f) Is the hold time met using the clock period determined in Part (d) above and accounting for all non-ideal factors? Explain your answer.

Clock driver jitter affects both paths, so ignore

Transmit path - assume $t_{rise} = 0$ for worst case

Clock edge early by 0.03 + 0.1 = 0.13 ns

Path shorter by 0.125 + 0.08 = 0.205 ns

Receive Path

Late by 0.025 ns

Delay path = 1.75 ns (Part a)

Shorten by 0.13 + 0.205 + 0.025 = 0.36 ns

Hold time ≤ 1.75 − 0.36 = 1.39 ns − $\boxed{\text{met}}$