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

Student Number:. .....

University of Toronto
Faculty of Applied Science and Engineering

Test I – February 22, 2010
BA 2155

ECE532S – Digital Systems Design

Examiner – Paul Chow

1. There are 3 questions and **9** 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 [25]	
3 [15]	
Total [50]	

1. Starting with some short answer questions:

[5 marks] (a) Explain or define the following jargon/terms associated with using the Xilinx tools and board as if you were speaking to a classmate that has no knowledge of the environment you are using:

ISE

Xilinx tool for FPGA design that provides simulation, synthesis, place & route, bitstream generation & programming

Base System Builder

Wizard tool that helps build a working EDK System targetted at a specific board

XMD

Xilinx microprocessor debugger. Provides a debug monitor for the Micro Blaze processor

Bitstream

The configuration bits that are loaded into the FPGA

mhs file

Microprocessor Hardware Specification file
Defines the hardware configoration of the system including connectivity, modules, processor, address spaces

[2 marks] (b) What does the keyword *volatile* mean in a C program?

Indicates a variable where the valve may be changed by processes outside the current program so the optimizer should not optimize accesses to it.

[1 marks] (c) What is the *modelsim* command to set a signal called `Input_Enable` to the value *zero*?

force Input-Enable 0

[2 marks] (d) The FPGA on the board you are using is programmed via JTAG. What is JTAG? (Note: Explaining the letters of the acronym is not the desired response.)

A standard for a test and debugging port that uses a small number of pins, usually four. Data is transferred serially

[25 marks] 2. Knowing that you have experience with FPGA designs and the Xilinx XUP board, the first project assigned to you at your new job is to prototype a system that takes in multiple video streams and then displays the videos on a single monitor in several ways ranging from just picking one of the videos for output, to various options displaying multiple videos at once, such as picture-in-picture, or tiling the videos. Note that some video scaling hardware will be required because the videos may need to be scaled in size.

The system will have four digital cameras as inputs and one output monitor. The video resolution is 1024×768 pixels for the cameras and the output. The cameras and monitor use non-interlaced video at 60 frames per second, meaning a complete frame is transmitted every 1/60th of a second. The digital cameras output RGB video with eight bits per colour on a 24-bit bus with some added control signals that provide the clocking and video sync pulses. The output video stream is also in RGB mode with eight bits per colour.

The camera interfaces have a number of configuration registers that are accessed via a two-bit serial bus protocol called IIC, or I^2C. You note that the Xilinx EDK library includes an IIC controller for the PLB bus.

You 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.

You will design an FPGA-based prototype of this video switching and display system. 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.

You will develop your system with the XUP board connected to a host PC that runs all of the tools you need. A daughter card containing the four camera interface chips is provided. The daughter card will be designed to plug into the connectors on the XUP board. Assume that you have enough pins on the board and the FPGA to do what you need to do.

Draw a 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.). Do not forget to sketch out how the daughter card should work. To help the reviewers understand the design environment, you should include how the prototype is connected to the PC and how it interacts with the tools.

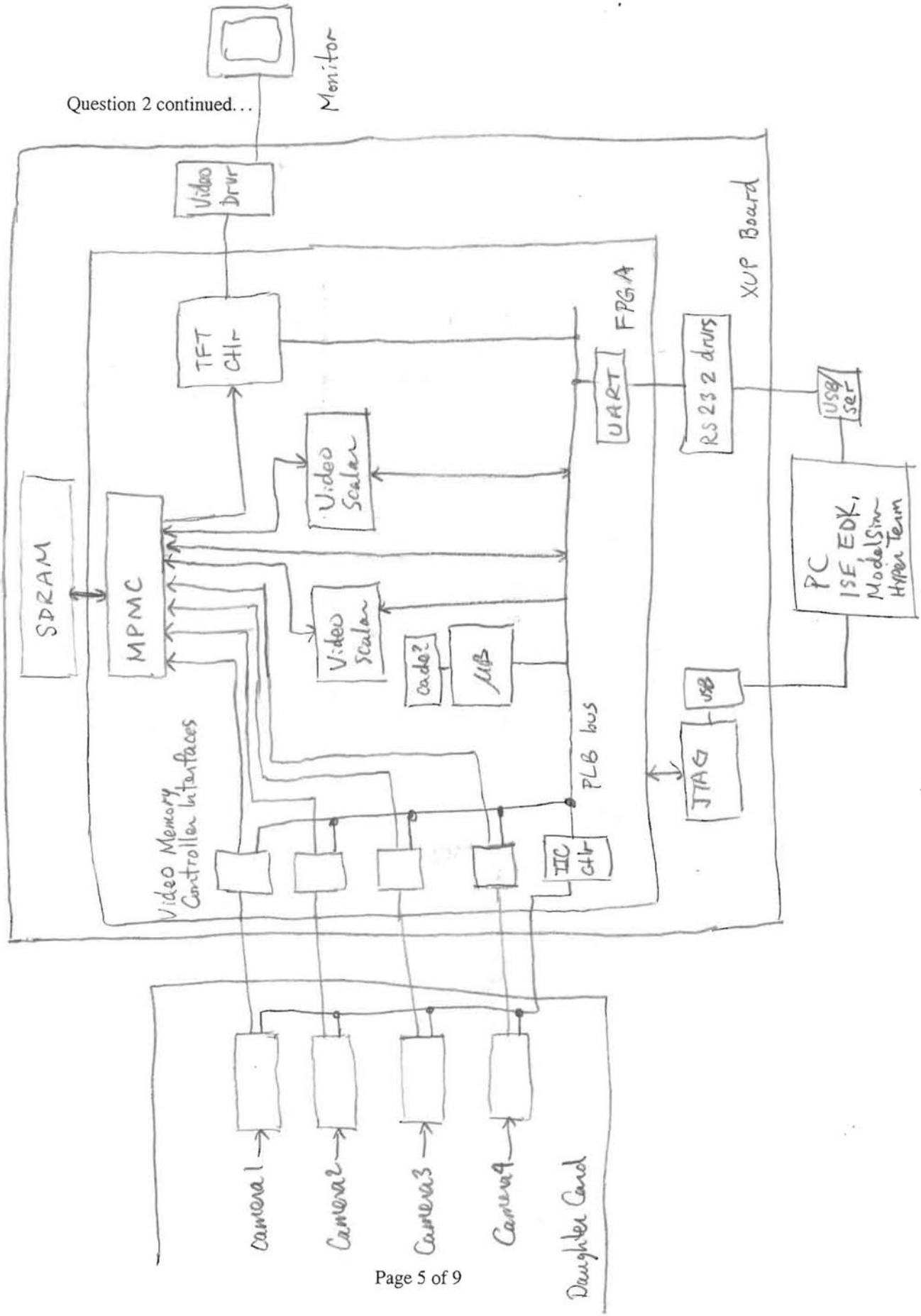
For each block in your diagram, briefly explain the function of the block and whether it is custom-designed logic or a block that you know is available from the EDK library of IP. If the block is custom, indicate whether you have to write it from scratch or whether there is a tool that can help you.

Include bandwidth estimations to show that your architecture is capable of sustaining the necessary data rates.

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 three pages to answer this question.

Question 2 continued...



Basic system operation

- Video is streamed directly to SDRAM to avoid bus bandwidth issues
- μB does control, status, debug functions
- TFT controller outputs from a frame buffer in memory to video driver
- Video scalars – reads videoframes and writes scaled video

Bandwidth Check

- main potential problem area is at memory interface
- ports - 4 input video – writes only
  - 1 output video - only reads
  - 2 video scaling – assume read and write, so need 2x BW
  - μB
- video bandwidth - $1024 \times 768 \times 4 \times 60 = 189$ MB/s
  - note use 4 bytes instead of 3 to put 1 pixel/address
  - memory BW = $(4 + 1 + 2*2) \times 189 = 1.7$ GB/s
    → assume μB is negligible
- memory BW - memory is DIMM - 64 bits = 8 bytes datapath
  - assume 133 MHz clock
  - DDR

  BW = $8 \times 133 \times 2 = 2.1$ GB/s peak

This seems marginal for providing 1.7 GB/s. Could use faster memory or one less video scalar. If video scalar is producing smaller images, it won't need as much bandwidth

Function of blocks

Daughter card - contains 4 camera interface chips and connections to cameras

Video Memory Controller Interfaces - Accepts data from camera interface chips and writes pixels to frame buffers in memory. Configured by µB to set frame buffer pointers
- custom block to write in Verilog

IIC Controller - connects to camera chips so they can be controlled by µB - in EDK library

µB - MicroBlaze Processor, provided by EDK

Cache? - may want to add a cache or local memory for faster execution, available in EDK

MPMC - multi-ported memory controller - available in EDK

TFT Controller - reads frame buffer and outputs to display
- avail. in EDK

Video Drivers - D/A on board to drive monitor

SDRAM - Memory DIMM

Video scalar - custom hardware to scale video images
- controlled by µB - provide two scalars, if memory BW allows

JTAG - JTAG controller on board

USB - USB to JTAG converter on board - connect to PC via USB cable

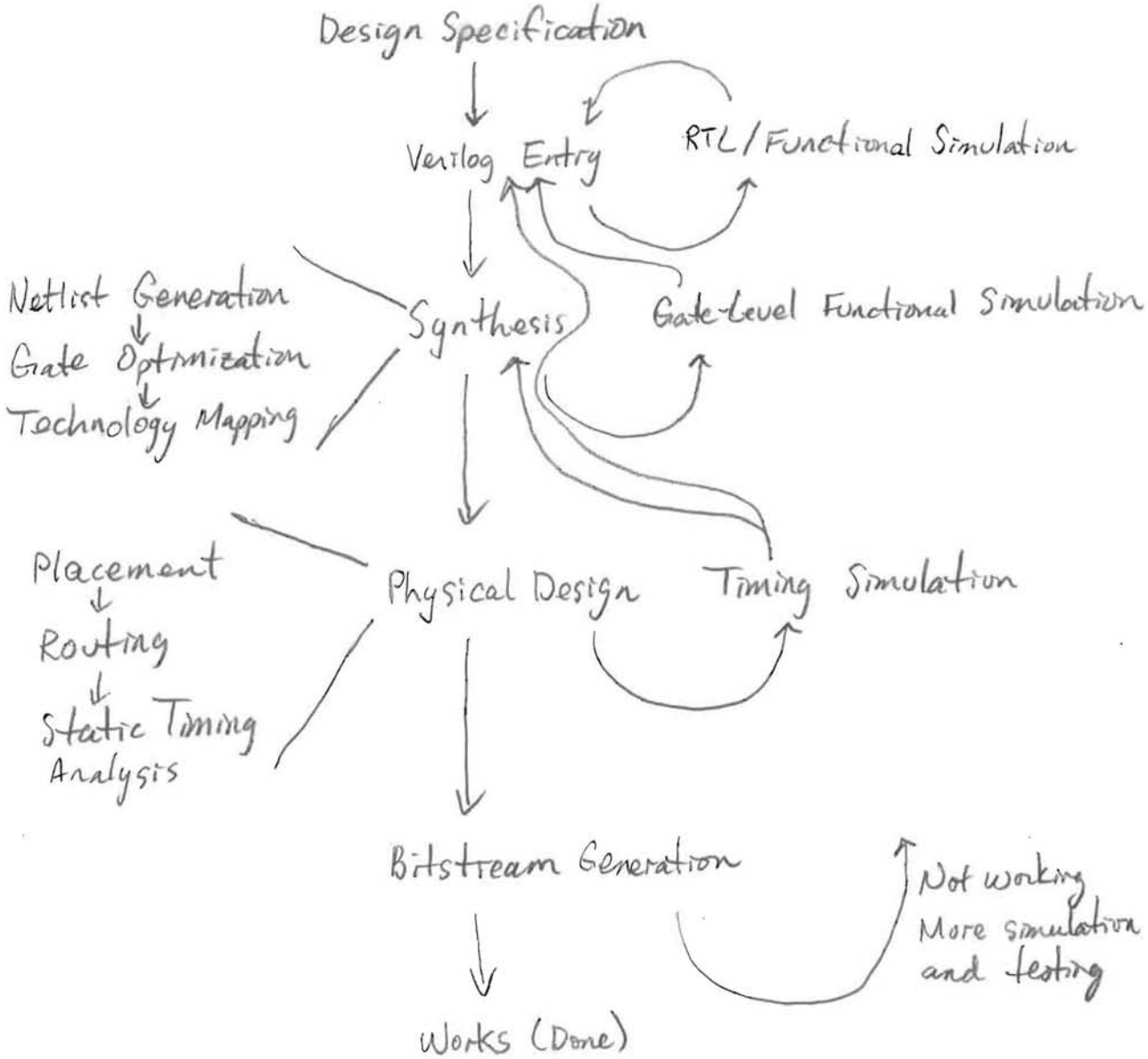
UART - For serial communication with PC, in EDK

RS 232 drivers - analog drivers for serial interface

USB/serial - converts serial cable to USB for connection to PC

PC - runs tools - EDK, ISE, ModelSim, Hyperterm

[15 marks] 3. Describe the basic CAD flow for an FPGA by showing a flow diagram and explaining each of the steps in the flow. Be sure to show where various types of simulations can be done. You should include the following steps, which are not listed in any particular order: timing simulation, placement, Verilog entry, gate optimization, synthesis, bitstream generation, design specification, technology mapping, netlist generation, physical design, routing, static timing analysis. Note that the list is not necessarily the complete list of steps.



Design Specification – Figure out basic functionality, often MatLab or C program model

Verilog Entry – design entry

Synthesis comprises

Netlist Generation – map Verilog into a netlist of gates FFs, and modules

Gate Optimization – improve netlist to reduce logic, improve timing

Tech Mapping – map netlist to the resources on the FPGA

Physical Design comprises

Placement – find a specific resource site for each element in the tech mapped net list.

Routing – connect the resources using the available wires

Static timing analysis – analyse circuit performance and report critical paths and timing violations

Bitstream Generation – create the configuration file for programming the FPGA