



Digital Systems

EEE4084F



FINAL EXAM

29 May 2012

3 hours

*Examination Prepared by:
Simon Winberg*

Last Modified: 10-May-2012

REGULATIONS

This is a closed-book exam. Scan through the questions quickly before starting, so that you can plan your strategy for answering the questions. If you are caught cheating, you will be referred to University Court for expulsion procedures. Answer on the answer sheets provided. Make sure that you **put your student name and student number**, the course code **EEE4084F** and a title **Final Exam** on your answer sheet(s). Answer each section on a separate page.

DO NOT TURN OVER UNTIL YOU ARE TOLD TO

Exam Structure

Marked out of 100 marks. Time per mark = 1min 48sec

<u>Section 1</u>	<u>Section 2</u>	<u>Section 3</u>	<u>Appendices</u>
Short Answers (2x 11 mark questions 2 x 10 mark questions) [42 marks] pg 2	Multiple choice (5x 4-mark questions) [20 marks] pg 4	Long Answers (3x questions) [38 marks] pg 6	A: Formulae B: Verilog cheatsheet pg 8

RULES

- NB →
- You must write your name and student number on each answer book.
 - Write the question numbers attempted on the cover of each book.
 - Start **each section on a new page**.
 - Make sure that you cross out material you do not want marked. Your first attempt at any question will be marked if two answers are found.
 - Use a part of your script to plan the facts for your written replies to questions, so that you produce carefully constructed responses.
 - Answer all questions, and note that the time for each question relates to the marks allocated.

Section 1: Short Answers [42 marks]

Question 1.1 [11 marks]

This question relates to FPGA-based Reconfigurable Computing (RC) platforms.

- Describe the differences between the temporal and spatial computing paradigms. Motivate which of these paradigms is probably better suited to developing applications that run on RC platforms. [3 marks]
- RC platforms that incorporate FPGAs often include a CPLD as well. The CPLD in such a RC platform is usually used as configuration and glue logic whereas the FPGA(s) are used for computation.
 - Describe the main differences between a FPGA and a CPLD [2 marks].
 - Motivate why a CPLD is often chosen to implement this glue logic instead of using a microprocessor processor or simply by clocking the FPGA programming lines directly from the PC. [2 marks]
- The overall design of the Programmable Active Memories (PAM) reconfigurable computing platform is illustrated on the right. As you can see each FPGA has its own SRAM bank, each with dedicated address and data lines. FPGAs connect to one another via a shared bus (not shown). Explain one advantage and one disadvantage of this design in terms of high speed data processing [2 marks]. Briefly argue a type of applications suited to this architecture, and a type not suited to this design. [2 marks].



Figure 1: PAM design overview

Question 1.2 [11 marks]

This question relates to general concepts and facts about parallel and high performance computing.

- Name two applications that commonly run on large scale parallel computing systems. [2 marks]
- Explain why much of the parallel code in a parallel computer system is often hand crafted? (i.e. much care and effort going into it, writing in a fairly low-level language like C.) [2 marks]
- Consider that in a business meeting you recommend using a parallel language to implement a parallel computing solution, and that automatic parallelization might be worth considering. But a colleague responds: “No way! There aren’t parallel computing languages and automatic parallelization is useless”. Present an argument to stand by your statement and illustrate examples confirming that parallel languages do indeed exist – also (to be a smart ass) clarify the current situation of automatic parallelization techniques. [5 marks]
- Nowadays, server class machines tend to use SMP architectures. Briefly explain: what is a SMP and what does the acronym SMP stand for. [2 marks]

Question 1.3 [10 marks]

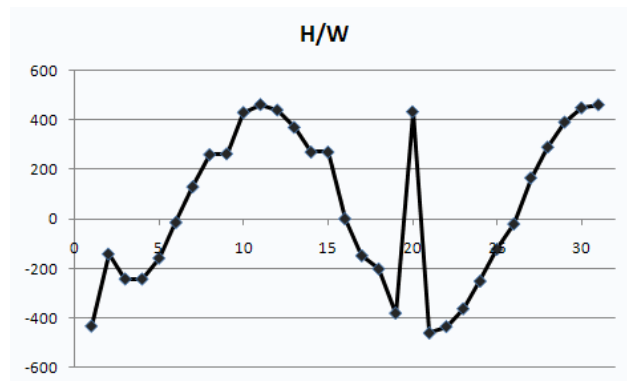
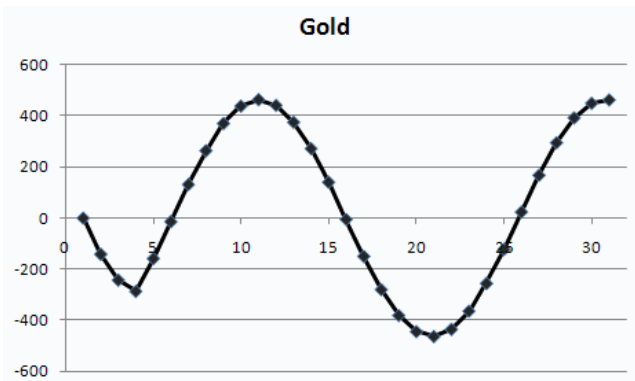
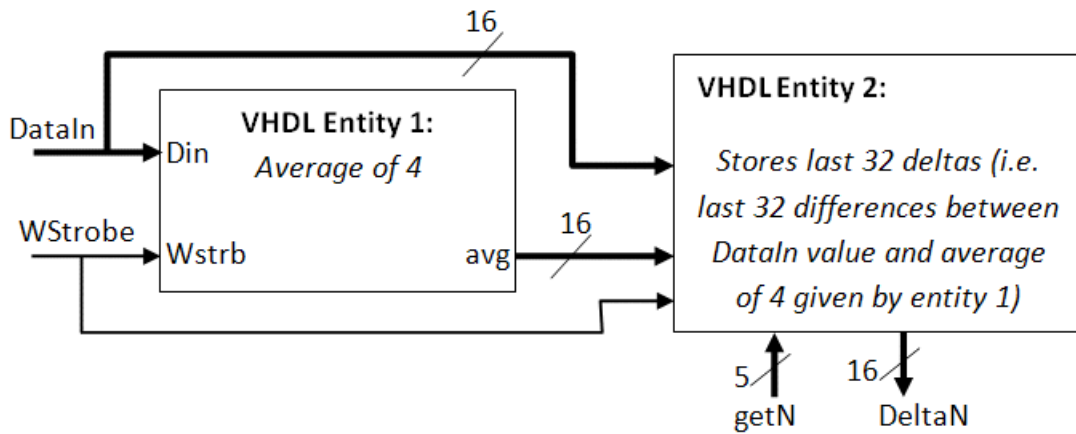
This question relates to interconnection fabrics and bisection bandwidth.

Consider you want to connect up eight processing nodes. These processing nodes are independent computers all to be networked using crossbar switches as needed. Assume all links have the same maximum speed (1Gbps).

- Contrast a drawback and a benefit between using a simple binary tree network topology compared to a binary fat tree network to connect these eight processor nodes. Determine the bisection bandwidth for each of the two cases. [4 marks]
- If the processing nodes were instead to be implemented using a VLSI approach, instead of individual computers, motivate why a fat tree approach is probably the better option, especially if there are only 8 processors in the design. [2 marks]
- Consider that this 8-processor system works cooperatively on a 128x128 matrix of 32-bit floating point values. Each processor node starts with only part of this matrix: specifically each processor has a sub-matrix of size 128 columns x 16 rows of floats. Processor 1 has the top 16 rows, processor 2 has the next 16 rows, etc. If the sum of all elements of the matrix (i.e. the 32-bit float value $\sum_{i=1:128, j=1:128} A_{i,j}$) was to be calculated, determine how many bytes would have to be transferred around the network, assuming the result of the final sum is to end up at node 1. Ensure you explain your answer logically. [4 marks]

Question 1.4 [10 marks]

- (a) Define what is understood to be a *reconfigurable computer*, highlighting how to discriminate between a reconfigurable computer and a non-reconfigurable / regular networked computer. [2 marks]
- (b) The Xilinx Virtex-6 comprises an architecture composed of two types of configuration logic blocks (CLBs), namely SLICEM and SLICEL blocks. Explain what a CLB is – surely it is just the same as a LUT (or is it)? Discuss the differences between SLICEL and SLICEM blocks and how different slides can be beneficial in terms of FPGA manufacture and programming. [4 marks]
- (c) Consider the design below. The system comprises two VHDL entities, imported into a schematic editor (such as the Xilinx ISE block editor). But when the design is synthesized and executed on hardware, weird results occur (see the graphs below), even though the data is streamed into the system at a much slower rate than the speed at which the entries operate. Indeed, much of the output from the hardware looks correct (see right hand graph below) yet doesn't match the golden measure (left hand graph). The correlation coefficient between the datasets is 0.834, verifying that something is wrong. Comment on what may be causing this problem and how it might be solved. [4 marks]



Section 2: Multiple Choice [20 marks]

NOTE: Choose only one option (i.e. either a, b, c, d or e) for each question in this section.

Q2.1 Consider that the following VHDL code:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity E1 is
    port ( A: in STD_LOGIC; B: in STD_LOGIC; X: out STD_LOGIC );
end E1;
```

Consider that you want this entity to output the result of the C expression $X = A \ \&\& \ !B$ (assuming that all these variables are Boolean or bit values). Which code snippet below will implement this architecture for the entity without syntax errors:

(a)
architecture Elbehavior of E1 is
begin
 X = A and not(B);
end;

(b)
architecture Elb of E1 is
begin
 X <= A and not(B);
end Elb;

(c)
module Elb of E1
{
 X = and(A,not(B));
};

(d)
process Elb of E1 is
begin
 X := A and not(B);
end;

(e)
architecture E1 :
begin
 X := A and not(B);
end E1;

[4 marks]

Q2.2 What is another term for asynchronous communications?

- (a) Non-blocking communications.
- (b) Blocking communications.
- (c) Handshaking communications.
- (d) Serialized data transfer.
- (e) Untimed communications

[4 marks]

Q2.3 Consider that the following variables are defined:

Define f = fraction of computation that can be parallelized

Define n = number of processors for parallel case

Then, according to Amdahl's law, the maximum speed-up achievable for the parallel case over a sequential case is (choose one option below):

- (a) $\text{Speedup} = f / (f - 1)(n+1)$
- (b) $\text{Speedup} = (n + 1) / (1 - f)$
- (c) $\text{Speedup} = (1 - f) / ((1 - n \cdot f))$
- (d) $\text{Speedup} = 1 / ((1 - f) + f/n)$
- (e) $\text{Speedup} = (1 - f) / (f/n - 1)$

[4 marks]

Q2.4 Which of the following is *not* an acronym describing one of Flynn's classifications for parallel computer architectures.

- (a) SISD
- (b) SIMS
- (c) MIMD
- (d) SIMD
- (e) MISD.

[4 marks]

Q2.5 Answer **true** or **false** to each question below (each answer is 1 mark).

- (i) A correlation between two identical data sets returns the value 0.
- (ii) iVerilog is a free Verilog HD code compiler and simulator.
- (iii) 8 GFLOPS is the *peak computation rate* of a processor that can complete four FP operations per clock and is being clocked at 2 GHz.
- (iv) The commonly used HPEC acronym "SWAP" means SoftWare And Power matrix.

[4 x 1 mark each = 4 marks]

Section 3: Long Answers [38 marks]

Question 3.1 [14 marks]

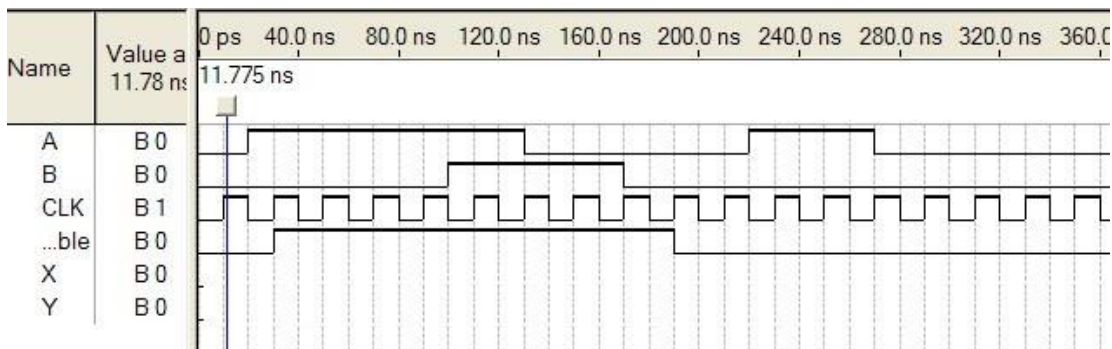
Consider the Verilog code provided below and then answer (a) and (b) below.

```
/* Test Program for EEE4084F Final Exam 2012 */
module Test1 ( A, B, CLK, Enable, X, Y );
  input A, B, CLK, Enable;
  output X, Y;
  reg    temp;

  // On every positive edge...
  always@(posedge CLK)
    begin
      temp = A;
    end

  assign X = Enable & (temp & ~B);
  assign Y = Enable & (temp | B);
endmodule
```

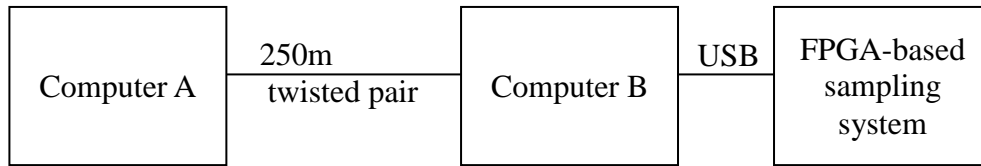
- (a) Convert the Verilog code given above into VHDL code. You need not specify any library includes, just focus on the entity and behavioral implementation. [8 marks]
- (b) The timing waveform configuration shown below is to be used to simulate the code given above. The simulation is set to work at 1ps resolution (i.e. much finer than the visualization can shown).



- (i) At time 200ns will X be at 1 or 0? [3 marks]
- (ii) If X is initially 0, explain when X will first transition to 1. Clearly you can't give a perfectly accurate figure as you haven't been given propagation delay information (but you can assume the maximum propagation time for the circuit is faster than the CLK clock rate). [3 marks]

Question 3.2 [14 marks]

This question concerns networked computer systems and calculating effective bandwidth. There are two computer systems in the network, Computer A and Computer B connected via a 250m twisted pair connection. Computer B connects to a FPGA-based sampling system via a short USB cable.

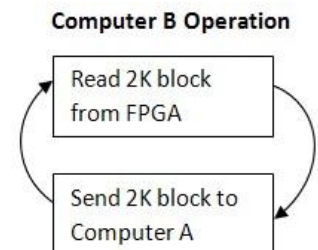


The raw bandwidth for the twisted pair connection is 1Mbps (i.e. bits per second) between computer. The signal propagation in the copper twisted pair is $0.8c$ (i.e. 80% speed of light). The sending overhead, to start transmissions is 200us. The receiving overhead is 100us. Assume each byte is encoded as a 10-bit sequence for both the twisted pair link and for the USB link.

The FPGA based sampling system captures a set of 2Kbyte samples every 500ms. The sample block is sent to Computer B from the FPGA via the 400MBps USB link. Computer B captures the whole sample block and sends the data on to Computer A via the twisted pair link.

Complete the following:

- What is the difference between *communication latency* and *bandwidth*? Present an example to aid your explanation that draws on the network system scenario given above. [4 marks]
- What is the *effective bandwidth* for sending the 2K byte block over the twisted pair link that runs between Computer A and B? [5 marks]
- Explain, with a clear argument using paper-based calculations, whether the connections described here will allow Computer B to continuously capture 2Kbyte blocks from the FPGA and send the data blocks to Computer A without missing any samples (i.e. as illustrated by the flowchart on the right). Assume the USB connection has no send and receive overhead (i.e. $<10\mu\text{s}$), and ignore latency and delays in memory access performed for either computer. [5 marks]



Question 3.3 [10 marks]

Cloud Computing has become a popular topic in recent year.

- Explain clearly what cloud Computing is, using an example to help your explanation. Also briefly indicate how a company can make an income from this. [4 marks]
- What is your view on the use of Cloud Computing in the next five years? In your discussion clarify the advantages and disadvantages of Cloud Computing; be particularly clear on how the potential disadvantages may be a hindrance to people using the technology, and how the advantages may help to improve the use of this technology. [6 marks]

END OF EXAMINATION

Appendix A:

Formulae

TABLE 14-1
Formulas for Crossbar Tree Networks

	Uniform	General
Processors	d^q	$\prod_{i=1}^q d_i$
Bisection Width (links)	$\frac{du^{q-1}}{2}$	$\frac{d_q}{2} \prod_{i=1}^{q-1} u_i$
I/O Links	u^q	$\prod_{i=1}^q u_i$
Switches	$\frac{d^q - u^q}{d - u}$	$\sum_{k=1}^q U_{q-k-1} D_k$ $U_j = \frac{\prod_{i=1}^j u_i}{u_j} \quad D_j = \frac{\prod_{i=1}^j d_i}{d_j}$

Table from pg 291 of Martinez, Bond and Vai 2008

Appendix B: Verilog Cheat sheet

Numbers and constants

Example: 4-bit constant 10 in binary, hex and in decimal: 4'b1010 == 4'ha -- 4'd10

(numbers are unsigned by default)

Concatenation of bits using { }

4'b1011 == {2'b10, 2'b11}

Constants are declared using parameter:

parameter myparam = 51

Operators

Arithmetic: and (+), subtract (-), multiply (*), divide (/) and modulus (%) all provided.

Shift: left (<<), shift right (>>)

Relational ops: equal (==), not-equal (!=), less-than (<), less-than or equal (<=), greater-than (>), greater-than or equal (>=).

Bitwise ops: and (&), or (|), xor (^), not (~)

Logical operators: and (&&) or (||) not (!) note that these work as in C, e.g. (2 && 1) == 1

Bit reduction operators: [n] n=bit to extract

Conditional operator: ? to multiplex result

Example: (a==1)? funcif1 : funcif0

The above is equivalent to:

```
((a==1) && funcif1)
```

```
|| ((a!=1) && funcif0)
```

Registers and wires

Declaring a 4 bit wire with index starting at 0:

```
wire [3:0] w;
```

Declaring an 8 bit register:

```
reg [7:0] r;
```

Declaring a 32 element memory 8 bits wide:

```
reg [7:0] mem [0:31]
```

Bit extract example:

r[5:2] returns 4 bits between pos 2 to 5 inclusive

Assignment

Assignment to wires uses the assign primitive outside an always block, e.g.:

```
assign mywire = a & b
```

Registers are assigned to inside an always block which specifies where the clock comes from, e.g.:

```
always@(posedge myclock)
```

```
cnt = cnt + 1;
```

Blocking vs. unblocking assignment <= vs. =

The <= assignment operator is non-blocking (i.e. if use in an always@(posedge) it will be performed on every positive edge. If you have many non-blocking assignments they will all updated in parallel. The <= operator must be used inside an always block – you can't use it in an assign statement.

The blocking assignment operator = can be used in either an assign block or an always block. But it causes assignments to be performed in sequential order. This tends to result in slower circuits, so avoid using it (especially for synthesized circuits) unless you have to.

Case and if statements

Case and if statements are used inside an always block to conditionally update state. e.g.:

```
always @(posedge clock)
  if (add1 && add2) r <= r+3;
  else if (add2) r <= r+2;
  else if (add1) r <= r+1;
```

Note that we don't need to specify what happens when add1 and add2 are both false since the default behavior is that r will not be updated. Equivalent function using a case statement:

```
always @(posedge clock)
  case({add2,add1})
    2'b11 : r <= r+3;
    2'b10 : r <= r+2;
    2'b01 : r <= r+1;
    default: r <= r;
  endcase
```

Module declarations

Modules pass inputs, outputs as wires by default.

```
module ModName (
  output reg [3:0] result, // register output
  input [1:0] bitsin, input clk );
  ... code ...
endmodule
```