CS 401 NAME(s): ___________________________ Due Apr 24

FINAL GROUP PROJECT: CPU DESIGN PART II: ALU & DPU DESIGN

CATEGORY	Beginning 70% – 79%	Satisfactory 80% – 89%	Excellent 90% – 100%
30 pts possible **ALU DESIGN**	4 ALU operations, VHDL code poorly commented, no ALU hardware design drawing.	5-6 ALU operations, satisfactory VHDL comments, adequate ALU hardware diagram. No testbench that verifies all operations.	7 or more ALU operations, excellent comments, neat and professional ALU hardware diagram. VHDL testbench that verifies all the ALU operations.
30 pts possible **DPU DESIGN**	VHDL code for DPU poorly commented	VHDL code for DPU, satisfactory drawing of DPU, control signals	Well documented, neat hardware diagram of the DPU, well documented VHDL code for the DPU and subcomponents, list of control signals needed by the DPU

2.1 ALU(s) DESIGN

Last project you designed the assembly language for your CPU. What mathematical instructions does your language require? These define the instructions required by the ALU(s). Is your processor general or special purpose? If general purpose you may wish to included more ALU operations. You may reuse the ALU design from mips (pages 248-250 in Harris), or, I encourage you to design your own ALU from scratch.

Hints for ALU Design

- **Do you want to support parallelism in your final CPU design? If so, what implications does this have for your ALU? i.e. do you want multiple simple ALUs?**
- **What ALU operations did the processors your group researched support?**
- **Choose your operations carefully and make sure it will be easy to add new operations if needed.**

Suggestions for possible ALU operations

shifters (shl, shr)
arithmetic (add, sub, mul, div)
comparison (lt, gt, lte, gte, eq)
bitwise logical operations (and, or, xor, inv, etc.)

ALU Design

1. **Make a list of ALU instructions and description of each instruction.**
2. **Draw a neat, detailed hardware diagram of the ALU.**
3. **Create a VHDL implementation of the ALU and a testbench to verify all operations.**

2.2 DPU DESIGN

Review the slides and the examples given in the textbook (pages 377-381) for the Data Path Unit design. Now that you know your target processor application, and you know your ALU’s capabilities, lay out the Data Path Unit for your processor.

1. Now, lay out the data paths (data busses) required to connect memory to your registers and to your ALU.
2. Will it be possible for the Fetch, Decode, Execute cycle to use your data paths, registers and devices to execute your instructions? Modify your data path as necessary to make sure each instruction will be able to run and use the data buses and registers effectively.
3. What multiplexers will you require? Sign extenders? Zero extenders? Adders?
4. Will your processor re-use the ALU to compute the next PC address?
5. Draw out a detailed design of your processors DPU.

Look on the next page for what you should hand in.

What to Hand In:

ALU Design

- **ALU list of operations with description of each operation.**
- **ALU hardware diagram (neat, detailed diagram)**
- **ALU VHDL implementation code.**
- **ALU VHDL testbench to verify all ALU operations.**

DPU Design

- **Neat DPU hardware diagram. (Neatly drawn using Visio (free from DreamSpark) or another drafting tool or incredible hand drafting talent with a ruler and pencil.)**
- **DPU VHDL code.**
- **Include the vhdl code for all of the DPU sub-components too.**
- **List of control signals needed by the DPU (these will eventually come from the CU)**