ECGR 3183 Spring 2018
Dr. Mihail Cutitaru
Project 2 (13%) – Draft

Assigned: Wednesday, March 21, 2018
Due: by Wednesday, April 25, 5PM

Project Overview

You are tasked with designing and simulating a floating point co-processor. You are provided with:

- an instruction set
- architecture parameters

You will provide:

- documented ISA
- architecture design
- controller design
- VHDL/C++/other simulation

Project Milestones:

Milestones are provided for the project to give you guidelines to completing the work. Milestones will not be graded. However, if you complete a milestone by the designated date, feedback will be provided as quickly as possible, allowing you to catch problems as early as possible. You will also be given an opportunity to schedule time to sit down with me and discuss your milestone. If you miss the deadline, feedback will be provided if and when the opportunity arises, but will receive very lower priority. All work should be submitted electronically on Canvas.

Milestone 1	ISA Specifiction Instruction formats Opcodes Condition codes Architectural features	Mar 28
Milestone 2	Architecture Design Architecture for your design	Apr 4
Milestone 3	Controller Design Architecture	Apr 11
Milestone 4	Simulation: complete with testing Controller Complete Testing	Apr 18
Final Project	Final Report and Results	Apr 25 (hard deadline)

Results submitted for a milestone are not cast in stone. You may rework milestones when working on future milestones as the need arises.

System Parameters:

Data: IEEE single-precision floating point numbers

Register file: 16 registers

Grading:

ISA: 20%

Architecture: 20%

Controller: 20%

Simulation: 20%

Report: 20%
(includes self- and peer-evaluations)

Instruction Set:

Set	`Set Ri, #value`	Ri ← value
Get	`Get Ri`	Return Ri
Move	`Move Ri, Rj`	Ri ← Rj
Add	`Fadd Ri, Rj, Rk`	Ri ← Rj + Rk
Subtract	`Fsub Ri, Rj, Rk`	Ri ← Rj – Rk
Negate	`Fneg Ri, Rj`	Ri ← -Rj
Multiply	`Fmul Ri, Rj, Rk`	Ri ← Rj * Rk
Divide	`Fdiv Ri, Rj, Rk`	Ri ← Rj ÷ Rk
Floor	`Floor Ri, Rj`	Ri ← $\lfloor Rj \rfloor$
Ceiling	`Ceil Ri, Rj`	Ri ← $\lceil Rj \rceil$
Round	`Round Ri, Rj`	Ri ← round(Rj)
Absolute Value	`Fabs Ri, Rj`	Ri ← $\lvert Rj \rvert$
Inverse	`Finv Ri, Rj`	Ri ← 1/Rj
Minimum	`Min Ri, Rj, Rk`	Ri ← min(Rj, Rk)
Maximum	`Max Ri, Rj, Rk`	Ri ← max(Rj, Rk)

Power	`Pow Ri, Rj, #integer_value`	$Ri \leftarrow Rj^{integer\_value}$
Sine	`Sin Ri, Rj`	$Ri \leftarrow \sin(Rj)$
Cosine	`Cos Ri, Rj`	$Ri \leftarrow \cos(Rj)$
Tangent	`Tan Ri, Rj`	$Ri \leftarrow \tan(Rj)$
Exponent	`Exp Ri, Rj`	$Ri \leftarrow e^{Rj}$
Logarithm	`Log Ri, Rj`	$Ri \leftarrow \ln(Rj)$
Square Root	`Sqrt Ri, Rj`	$Ri \leftarrow \sqrt{Rj}$

Note: It is suggested that you implement the transcendental functions as Taylor Series, but you may opt for another implementation approach. Also, the round, ceiling, and floor functions would round up to the nearest integer, expressing the result in FP format.

The intermediate milestones are optional and require a high-level simulation of this system, but the simulation will have to be present in the final submission. Work will be done in teams of 3.

The following is provided as a general outline for your final report. This is not to say that you need to follow this, but it is intended to give you a starting point.

1. Introduction
2. ISA (instruction set, instruction formats)
3. Architecture
   3.1. Datapath
   3.2. Controller
4. VHDL/HLL Description (do not include code, upload your entire project zipped up via Canvas, this should describe the model)
5. Testing
6. Conclusion

Make sure you focus on your design. Your design is not captured in your VHDL/HLL code, it should be laid out in your report. Figures are excellent for showing your structure. This includes architecture diagrams, state machine diagrams, etc. Then your VHDL/HLL code should reflect these diagrams. When showing waveforms demonstrating the functionality of your architecture, do not just show the waveform. The waveform should be annotated, i.e. put comment blocks on the image with pointers to explain the waveform, or explain it in your write-up.

Your testing section should include a test plan, some sample tests, and the test results. Your tests should be small sections of code that focus on testing one aspect of your architecture, maybe one instruction. Some sample programs are shown below.

Test programs:

1) The following test program computes the force in Newtons necessary to move an object up a ramp at constant speed. The equation is:

$$F = 9.8m(\sin\theta + k\cos\theta)$$

where m is the mass of the object in kilograms, k is the coefficient of friction, and θ is the angle the ramp makes with the horizontal. The program computes the force if a 100kg crate is moved on a 30° incline with a coefficient of friction of 0.4.

```
Set R1, #30.0          --angle of inclined plane
--convert to radians
Set R2, #3.14159265    --pi
Fmul R1, R1, R2
Set R2, #180.0
Fdiv R1, R1, R2
--compute equation
Cos R2, R1
Set R3, #0.4
FMul R2, R2, R3
Sin R1, R1
Fadd R1, R1, R2
Set R2, #100.0         --mass of crate
Fmul R1, R1, R2
Set R2, #9.81          --acceleration of gravity
Fmul R1, R1, R2
Get R1
```

2) The following test program computes Newton's Law of Universal Gravitation. The equation is:

$$F = G\frac{m_1 \times m_2}{r^2}$$

for two objects of masses m_1 and m_2 separated by distance r.

```
Set R0, #2.3           --m1
Set R1, #5.7           --m2
Fmul R0, R1
Set R1, #13.2          --r
Pow R1, #2
Fdiv R0, R0, R1
Set R1, #6.673E-11     --gravitational constant
Fmul R0, R1
```

3) The following test program computes the Quadratic Formula. The equation is:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

```
Set R1, #4.0
Set R2, #2.0          --a
Fmul R1, R1, R2
Set R3, #2.0          --c
Fmul R1, R1, R3
Set R3, #5.0          --b
Pow R4, R3, 2
Fsub R1, R4, R1
Sqrt R1, R1
Fneg R3, R3
Fadd R0, R3, R1
Fsub R1, R3, R1
Set R3, #2
Fmul R2, R2, R3
Fdiv R0, R0, R2
Get R0
Fdiv R1, R1, R2
Get R1
```