CG3207 Lab 3

Neil Banerjee

neil@nus.edu.sg

Lab 3 Overview

- Implement signed, unsigned division into Mcycle (sim only)
- Incorporate Mcycle into processor; implement MUL, DIV
- Enhance Mcycle

Task 1: Mcycle Divider [13]

- Download design files from wiki
- Complete Mcycle.v and test_Mcycle.v
- Incorporate both signed and unsigned division
- Simulate using a good testbench; hardware implementation not needed *for this task*
  - Make sure to use the correct top module
- Assumptions:
  - Divisor is never 0
  - -5/2 => Quotient -2, Remainder -1; or Quotient -3, Remainder 1

Suggested Test Cases

Multiplication		Division	
	32-bit		32-bit
1111 x 1111	FFFFFFFF x FFFFFFFF	0000 / 1111	FFFFFFFF / FFFFFFFF
0000 x 1111	00000001 x FFFFFFFF	1111 / 1111	FFFFFFFF / 00000001
1111 x 0000	FFFFFFFF x 00000001	1111 / 0001	00000001 / FFFFFFFF
1000 x 0111	00000000 x FFFFFFFF	1111 / 0111	00000000 / FFFFFFFF
0111 x 1000		1000 / 0111	
0000 x 0000		0111 / 1000	
0001 x 1111		0111 / 0010	
0001 x 0111		1001010 / 1000	

Task 2: Incorporate Mcycle into CPU [7]

ARM

- Implement MUL and DIV
- ARMv3 does not include DIV
  - Use MLA instruction format
  - Stop the simulation at any MLA instruction, modify the register manually, and then continue
- Assume DIV is unsigned division
- MUL can set Z and N flags – not required.
- Long multiplication instructions SMULL, UMULL etc. not required.

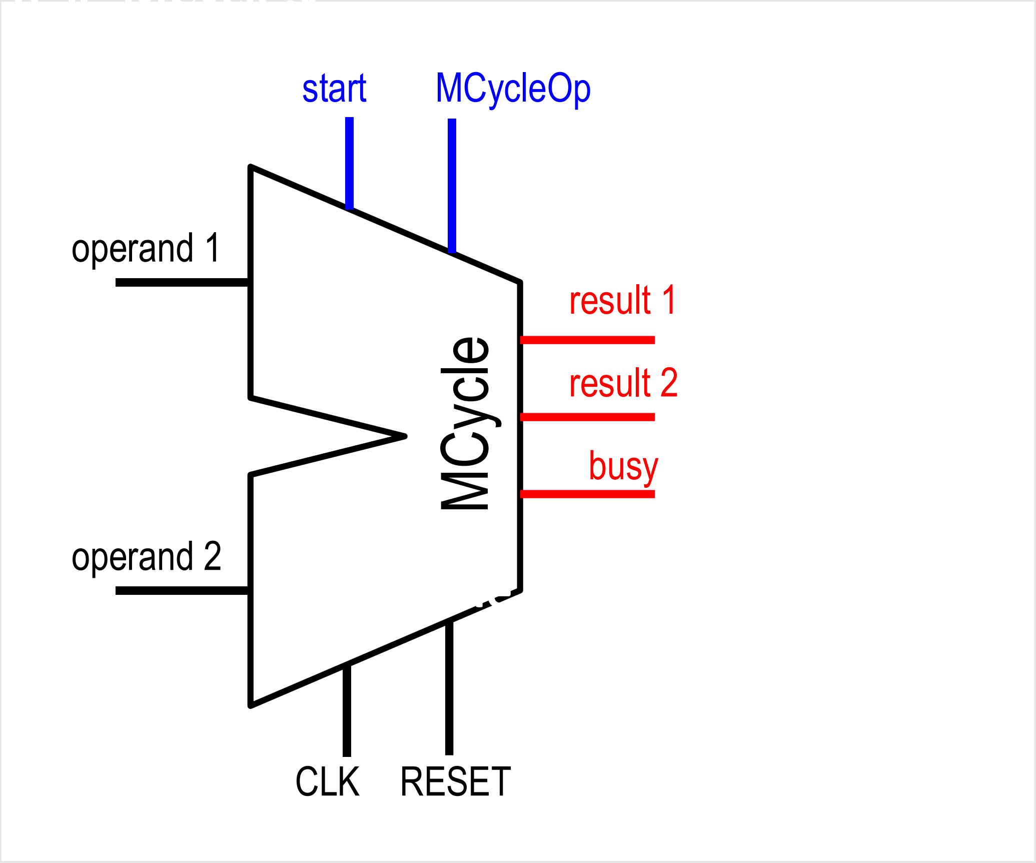
RISC-V

- Implement mul, divu
  - 32-bit versions sufficient
  - divu is part of the Multiply extension – no need to bodge things together
- div (signed), mulh variants, rem not required
  - Not much effort required (except mulhsu)

Task 2: Incorporate Mcycle into CPU [7]

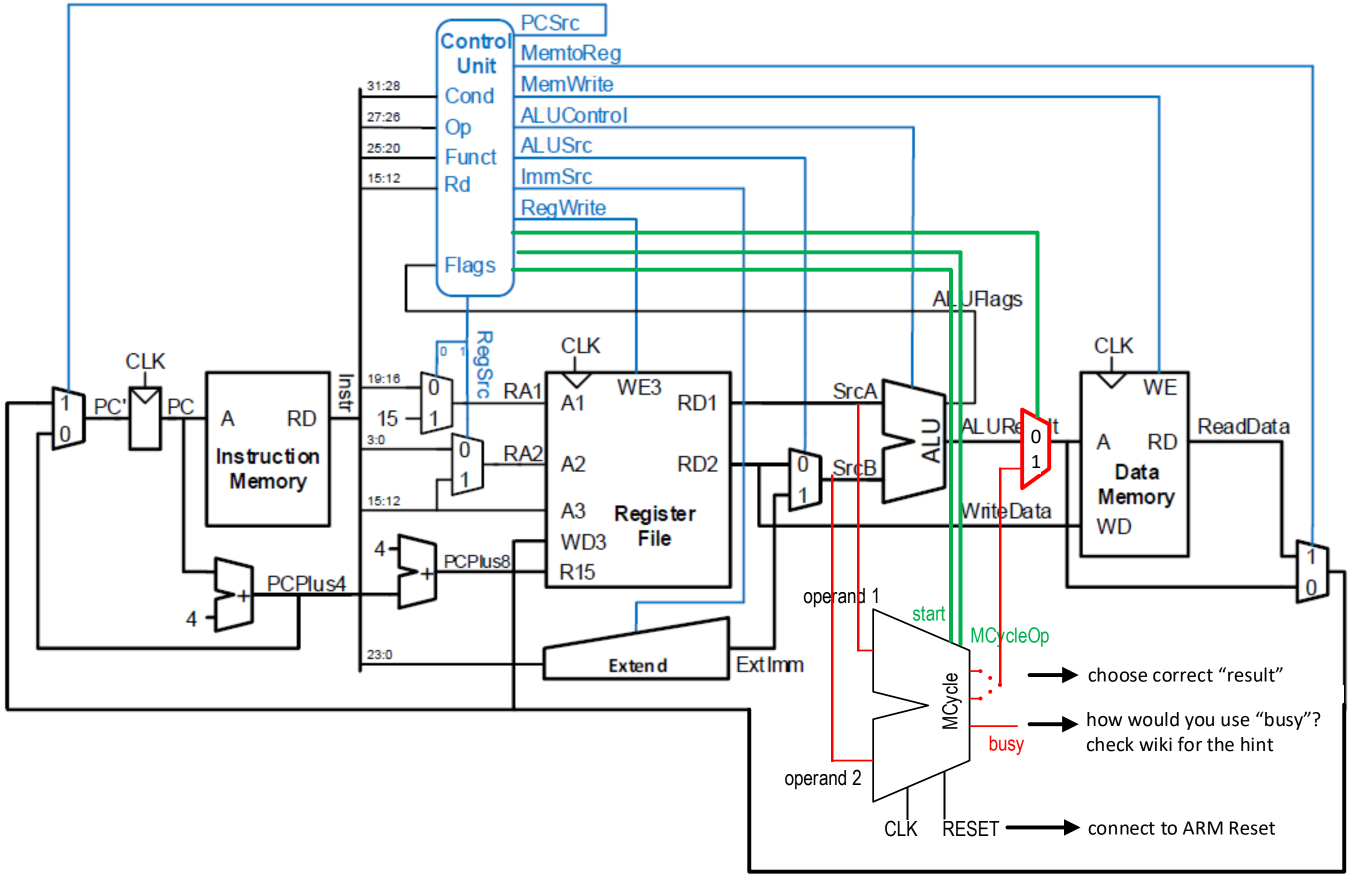
- Destination register contains quotient; discard remainder
- ARM and RISC-V MUL are 32-bit; no diff b/w signed and unsigned
- Modify the Control Unit to generate Start, MCycleOp signals
- !Busy can be used as PC WE; stall PC until mul/div complete
- Make appropriate datapath connections
- Write assembly program to demonstrate functions

Mcycle unit



Neil Banerjee | neil@nus.edu.sg | Image courtesy Animesh Gupta

Mcycle in CPU (ARM shown, RV similar)



choose correct "result"

how would you use "busy"?
check wiki for the hint

connect to ARM Reset

Task 3: Enhancement to Mcycle [5]

- Improve the performance of the given signed multiplier
- Suggestions:
  - Reduce number of cycles by increasing hardware use
  - Use a single adder for multiply/divide; maybe even use the ALU
  - Implement Booth’s algorithm
  - DO NOT implement a single-cycle multiplier using * operator – defeats purpose of this lab

Expectations

- Fully understand Mcycle, including provided code (if not modified)
- Fully understand multiplication algorithm used, enhancements (if applicable)
- Be able to explain algorithm by handwriting the process
- Know modifications done to CPU from Lab 2 onwards
- Be able to explain advantages and tradeoffs for your design decisions, algorithm choices