Arithmetic Logic Unit

Does the computations for Additition and Subtraction instructions

input1 `00000000`b

c in
+
c out

0
MUX
`00`h ALUoutput

input2 `00000000`b

b in
b out

operation `0`b
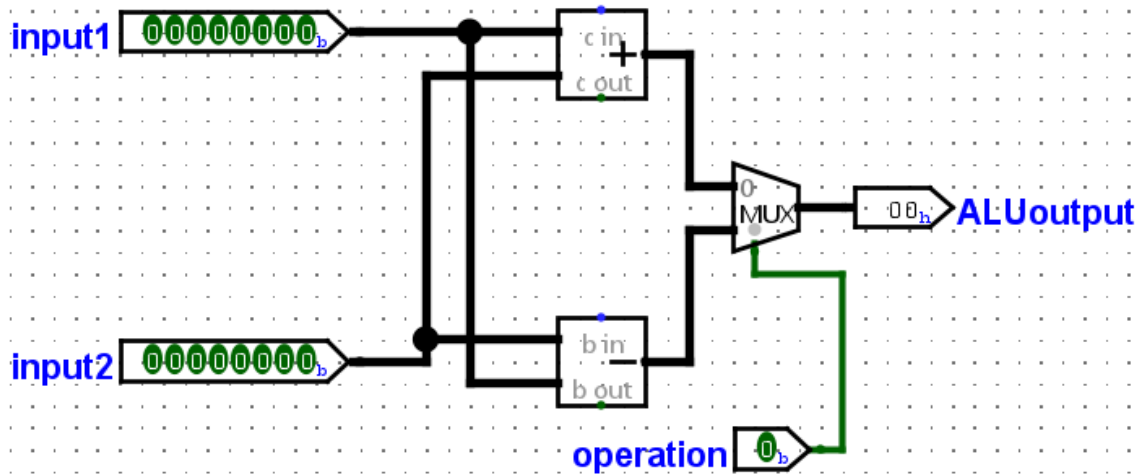
Register File

Holds the Read/Write Registters. There are four general purpose registers: X1, X2, X3, and X4

Clocky `0`b

Decd

X1

`00`
D   Q
WE
R

0
MUX
`00`b ReadReg1
`00`h RegData1

X2

`00`
D   Q
WE
R

X3

`00`
D   Q
WE
R

0
MUX
`00`h RegData2
`00`b ReadReg2

WriteReg `00`b

RegWrite `0`b

X4

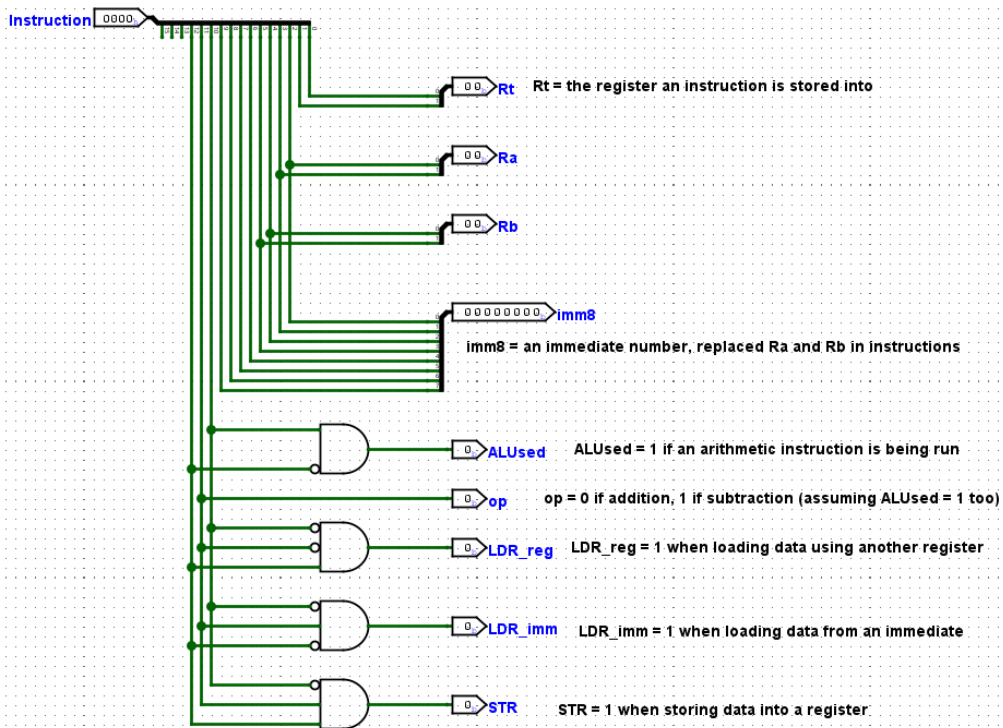`00`
D   Q
WE
R

RegDataW `00000000`b
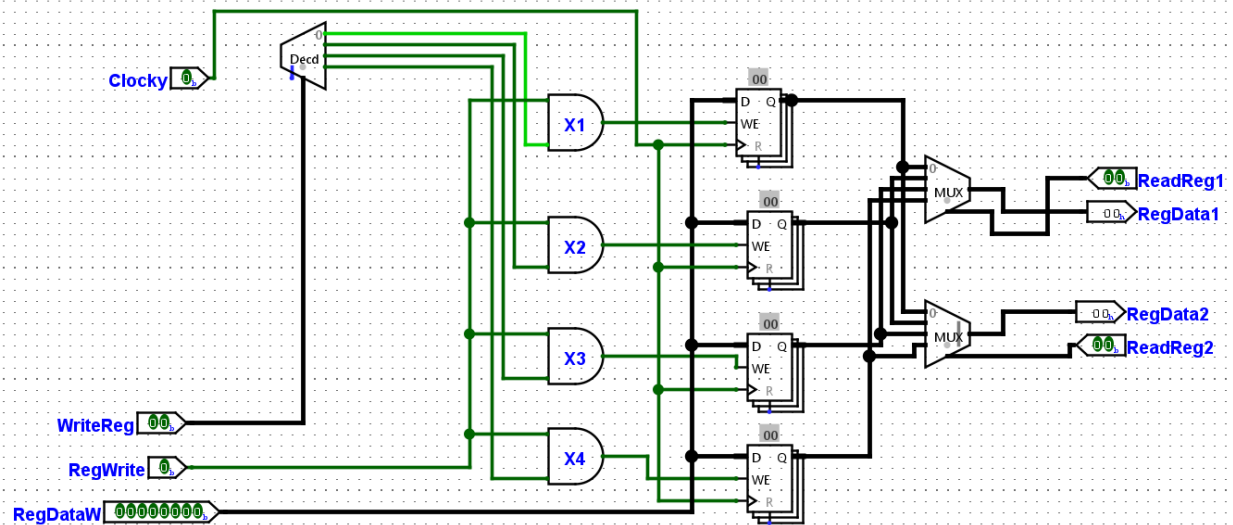
Control Unit

This translates the opcode into different outputs ulitzied by the CPU

- need 2 for second Reg
- need 2 for destination Reg
- need 8 for 8-bit imm
- need 1 for arithmetic (0 = add, 1 = sub)
- need 1 (mayb 2) for storing / loading

LDR/STR = address + register

Instruction [0000]

Rt = the register an instruction is stored into

Ra

Rb

imm8

imm8 = an immediate number, replaced Ra and Rb in instructions

ALUsed = 1 if an arithmetic instruction is being run

op = 0 if addition, 1 if subtraction (assuming ALUsed = 1 too)

LDR_reg = 1 when loading data using another register

LDR_imm = 1 when loading data from an immediate

STR = 1 when storing data into a register

Computer Processing Unit

Names: James Schorle and Lucas Vanzelli
Pledge: I pledge my honor that I have abided by the Stevens Honor System.
CPU Name: Cuntina Pookie Unit (CPU)

PC

instruction memory

ROM 256 x 16

IR

Instruction

Rt   target
Ra   inputA
Rb   inputB
imm8 immediate

immediate also stores the address

ALUsed   ALUsed
op   opALU
LDR_reg   LDR
LDR_imm   LDR_imm
STR   STR

ControlUnit

immediate

DataMemory
RAM 256 x 8

STR
LDR

Data

inputA
inputB
target

Clocky   RegData1
ReadReg1   RegData2
ReadReg2
WriteReg
RegWrite
RegDataW

RegisterFile

ALUsed
LDR
LDR_imm

input1   ALUoutput
input2
operation

opALU   ALU

immediate
LDR_imm

Data
LDR

ALUsed