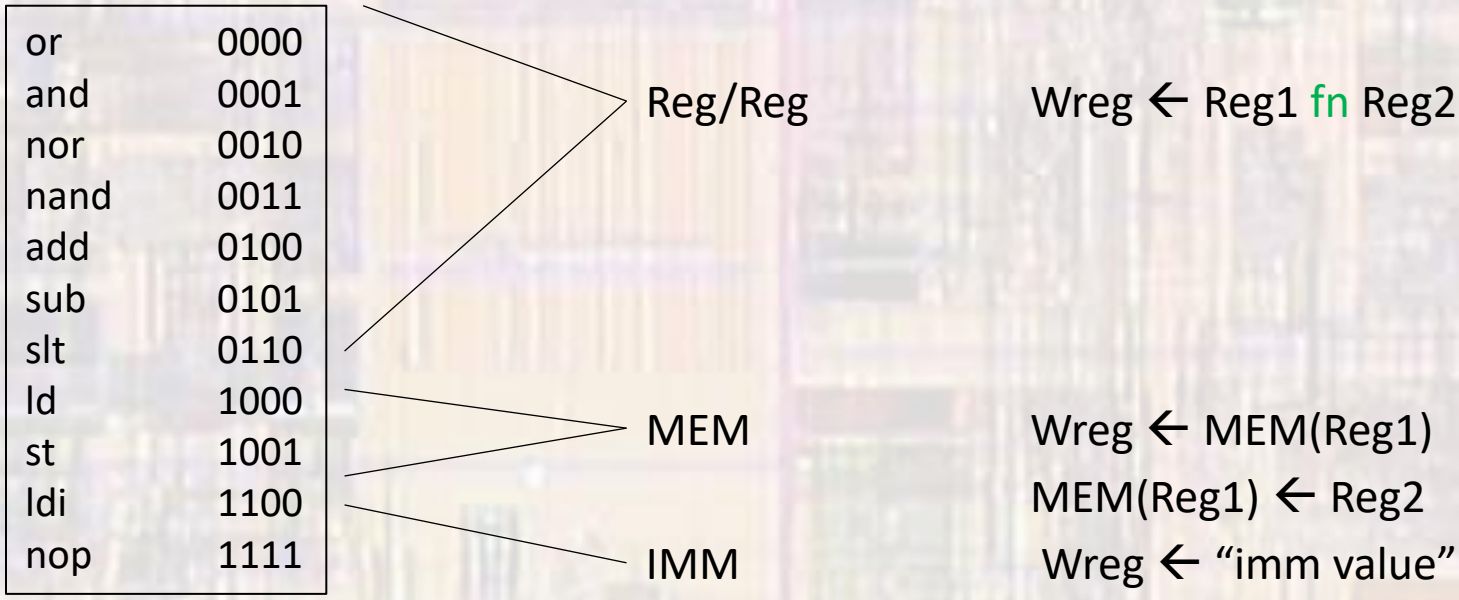
Single Cycle Processor Data Path

Last updated 7/18/23

Single Cycle Processor – Data Path

- Instruction Format

Instruction				Reg 1		Reg 2		W Reg		Immediate Value					

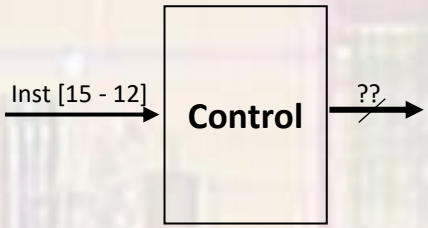


Single Cycle Processor – Data Path

- Decode / Register Access
  - Instruction Decode
    - Logic to decode the instruction
    - Pull off the relevant bits from the instruction

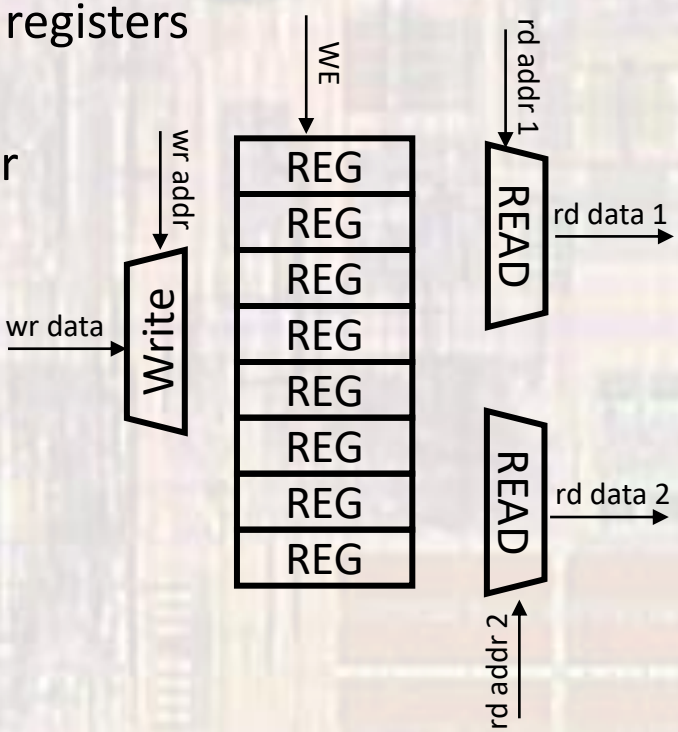
Instruction				Reg 1		Reg 2		W Reg		Immediate Value					
0	1	0	1	0	0	0	1	1	0	0	0	0	0	0	0

  - Create logic to drive control signals to other blocks



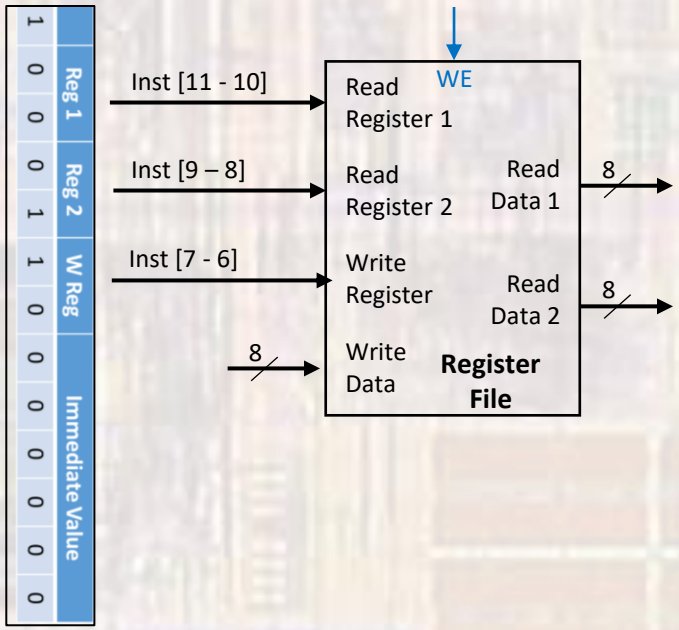
Single Cycle Processor – Data Path

- Decode / Register Access
  - Register File
    - Series of registers
    - 2 read multiplexors to select one of the registers for one of 2 outputs
    - Write multiplexor to choose one register to write to
    - Write data input
    - Write enable (or WE_b)



Single Cycle Processor – Data Path

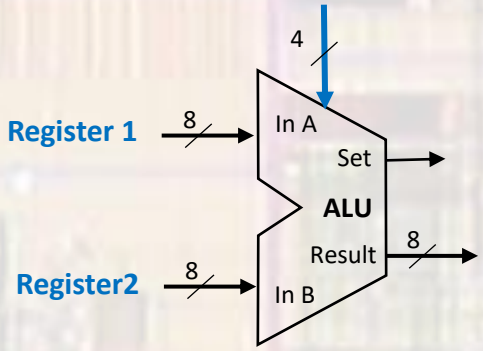
- Decode / Register Access
  - Register File – implementation comments
    - Instruction bit mapping to select registers
      - Wire specific bits from the instruction to the address ports of the register file.
    - 4 registers → 2 bits of address
    - Each register 8 bits wide
    - No rstb signal



Single Cycle Processor – Data Path

- Execute
  - ALU executes all arithmetic and logical instructions
  - Inputs are Register outputs
  - Control is decoded from instructions

Instruction				Reg 1		Reg 2		W Reg	Immediate Value						
0	1	0	1	0	0	0	1	1	0	0	0	0	0	0	0



Single Cycle Processor – Data Path

- Memory Access

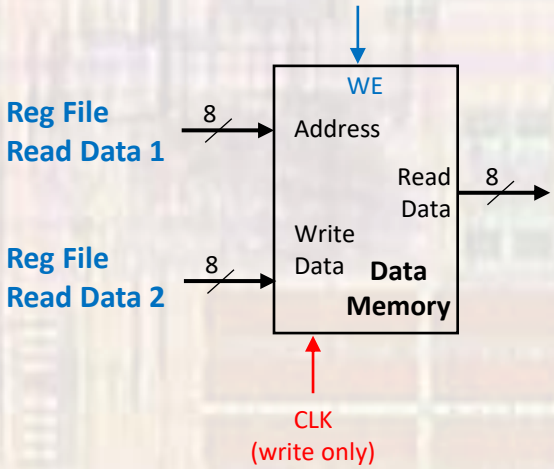
Instruction				Reg 1		Reg 2		W Reg		Immediate Value					
0	1	0	1	0	0	0	1	1	0	0	0	0	0	0	0

LD Wreg ← MEM(Reg1)

Instruction				Reg 1		Reg 2		W Reg		Immediate Value					
0	1	0	1	0	0	0	1	1	0	0	0	0	0	0	0

ST MEM(Reg1) ← Reg2

  - Load / Store Instructions
    - R/W from registers to data memory
  - Address
    - Pre-stored in one of the registers
    - Accessed from reg file “Read Data 1”
    - Added to the immediate value in the instruction (zeros)
  - Write Data
    - Pre-stored in one of the registers
    - Accessed from reg file “Read Data 2”
    - Synchronous
  - Read Data
    - Asynchronous read



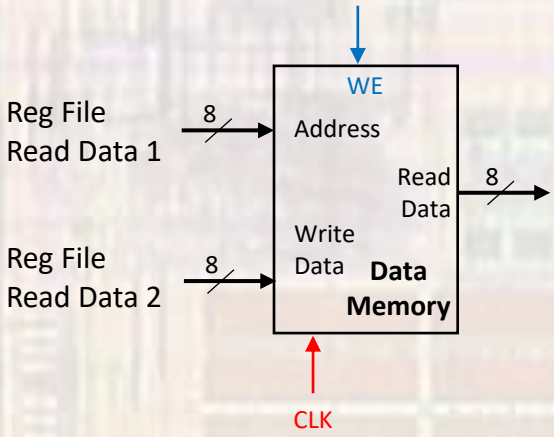
Single Cycle Processor – Data Path

- Memory Access
  - Data memory implementation comments
    - Inferred RAM
      - ?? in a x8 configuration

      → asynchronous address

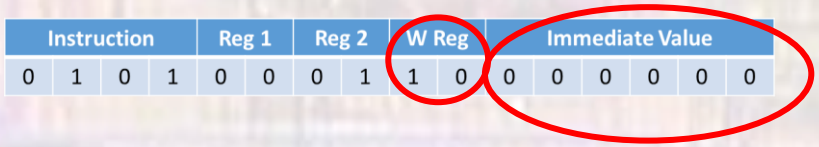
      → asynchronous read

      → synchronous write



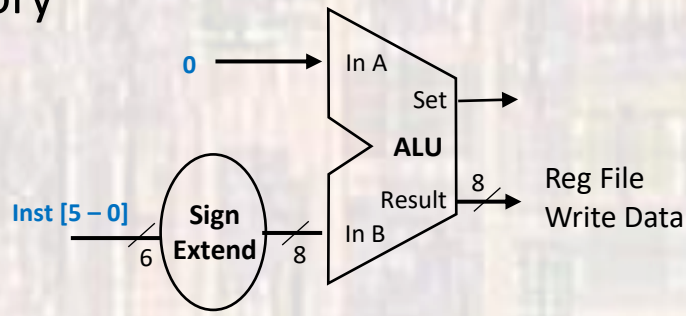
Single Cycle Processor – Data Path

- Memory Access



IMM Wreg ← "imm value"

  - Load Immediate Instruction
    - Load a register from the program memory
  - Value
    - Stored in the instruction
    - Sign-extended from 6 bits to 8 bits
    - **OR'd** with zero in the ALU
    - Uses the writeback mechanism to store the value in a register



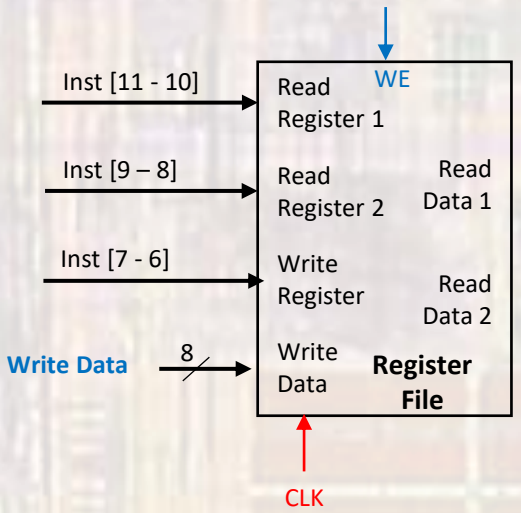
Single Cycle Processor – Data Path

Instruction				Reg 1		Reg 2		W Reg		Immediate Value					
0	1	0	1	0	0	0	1	1	0	0	0	0	0	0	0

- Write Back

Reg/Reg Wreg ← Reg1 fn Reg2
LD Wreg ← MEM(Reg1)

  - Write results or memory value back to a register
  - Write data
    - Comes from ALU (result or ldi)
      or
    - Comes from data memory (ld)
  - Synchronous



Single Cycle Processor – Data Path

- Missing Pieces
  - Program control elements
    - Branches
    - Jumps
    - Hazards

Single Cycle Processor – Data Path

- Full Data path