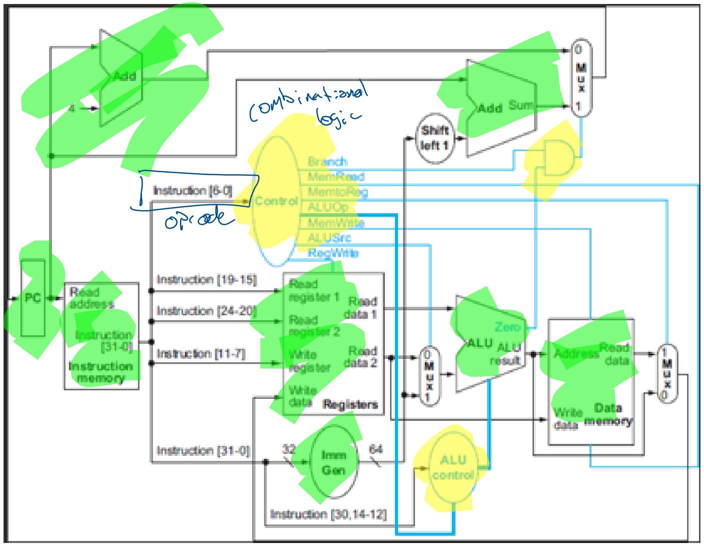
# Lecture 6: Single Cycle CPU

Friday, January 18, 2019    8:53 AM

## Outline

- Data path of a processor
- Controlling the data path
- Examples for Lab 2

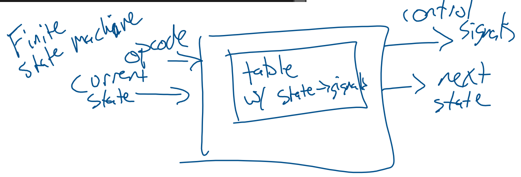
## Example from book



Data Path
does the "work"
moves the data around

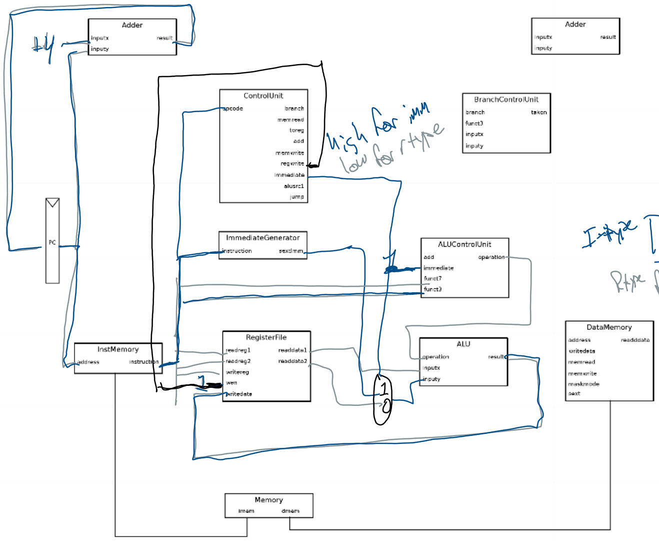
control path
- chooses which data path components to use
- controls the processor

control path could be implemented with microcode instead of combinational logic



## DINO CPU Examples

## I-Type



R-types

I-type

$\underline{R[rd]} = R[rs1] + imm$

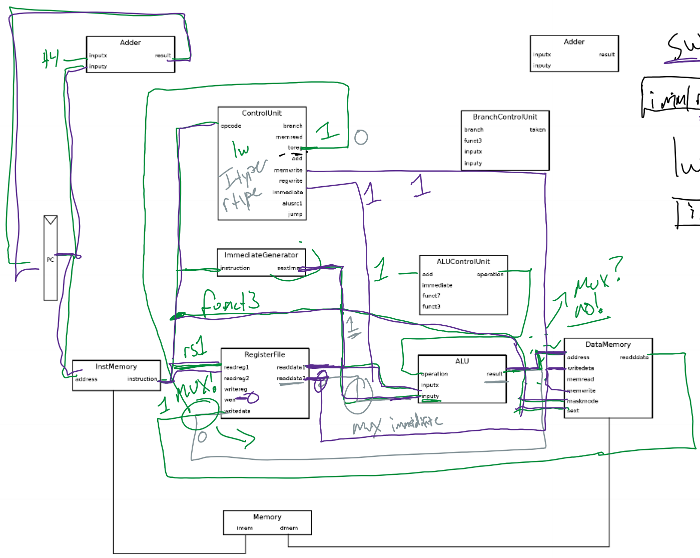
~~I-type~~

| imm | rs1 | funct3 | rd | 0010011 |
|---|---|---|---|---|
|  |  |  |  | 0110011 |

Rtype funct7

|  | opcode | immediate | regwrite | toreg | write memory | read memory |
|---|---|---|---|---|---|---|
| Itype | 0010011 | 1 use imm | 1 | 0 from alu result | 0 | 0 |
| Rtype | 0110011 | 0 use readdata2 | 1 | 0 | 0 | 1 |
| lw | 0000011 | 1 | 1 | 1 from memory | 0 | 0 |
| sw | 0100011 | 1 | 0 | X | 1 | 0 |

## lw/sw



sw M[R[rs1] + imm] = R[rs2]

| imm | rs2 | rs1 | funct3 | imm | 0100011 |
|---|---|---|---|---|---|

lw R[rd] = M[R[rs1] + imm]

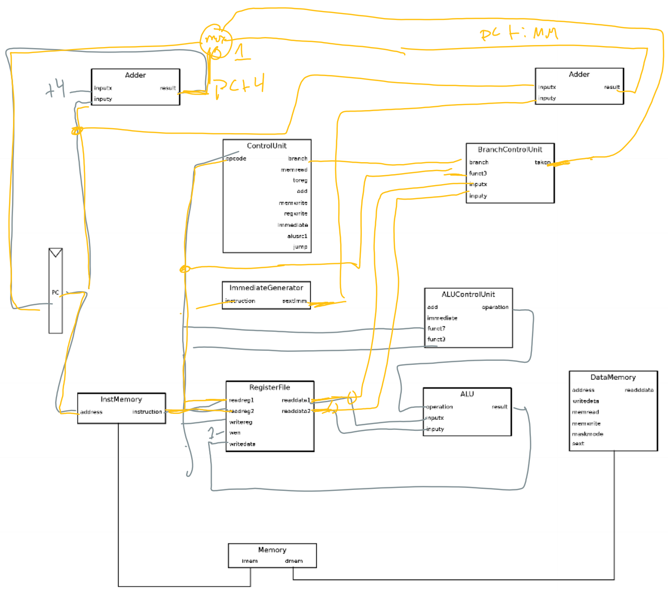
| imm | rs1 | funct3 | rd | 0000011 |
|---|---|---|---|---|

↳ mask and signextend

lb byte
lh halfword
lbu unsigned
lbu

00000 - - - - - - 10000000 lbu
1111111111111110000000 lb

## branch



r type

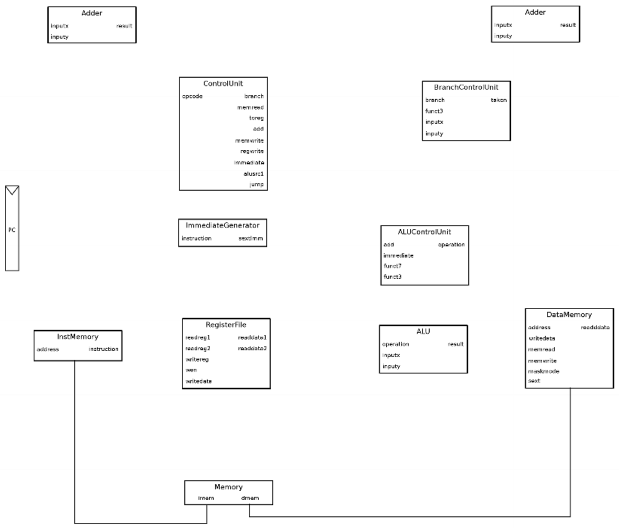
branch

beq if (R[rs1] == R[rs2])

PC = PC + imm<<1

imm gen does shift for you

## jumps


Adder
inputx
inputy
result
Adder
inputx
inputy
result
ControlUnit
opcode
branch
memread
toreg
add
memwrite
regwrite
immediate
alusrc1
jump
BranchControlUnit
branch
funct3
inputx
inputy
taken
PC
ImmediateGenerator
instruction
sextimm
ALUControlUnit
add
immediate
funct7
funct3
operation
InstMemory
address
instruction
RegisterFile
readreg1
readreg2
writereg
wen
writedata
readdata1
readdata2
ALU
operation
inputx
inputy
result
DataMemory
address
writedata
memread
memwrite
maskmode
sext
readdata
Memory
imem
dmem