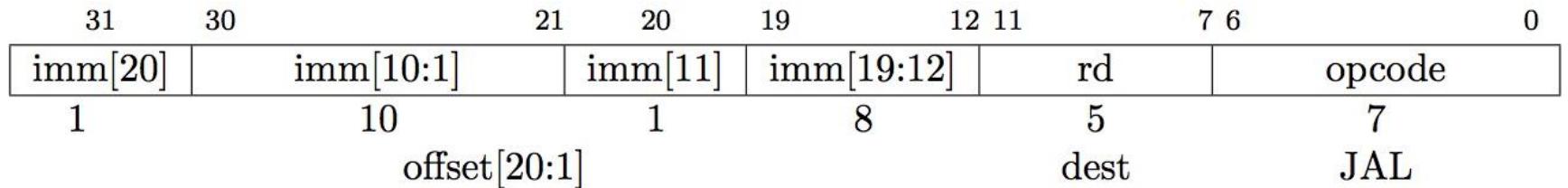


How to Add a New Instruction to a Given Datapath

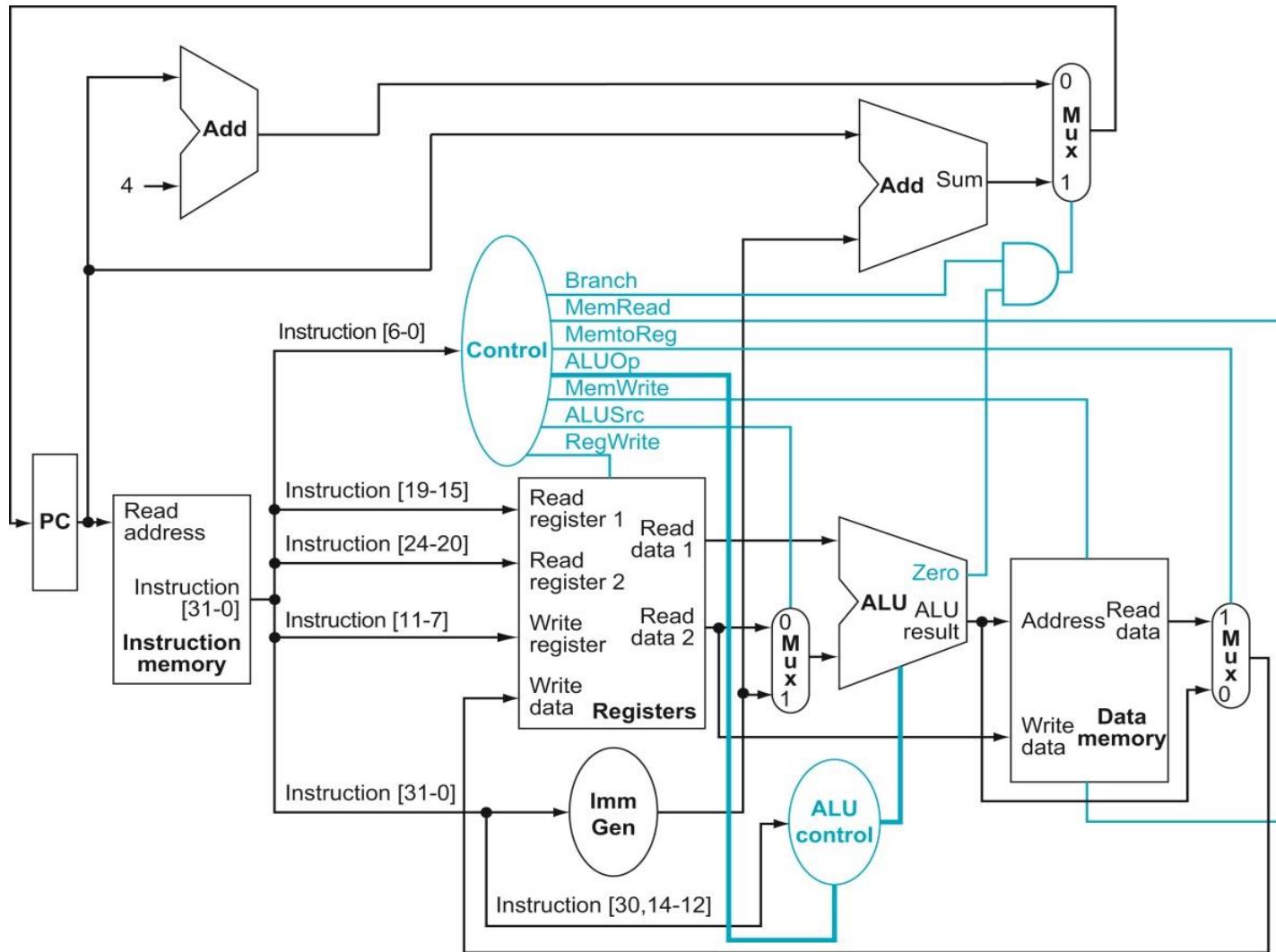
- 1) what is the new instruction supposed to accomplish?
Describe it using Register Transfer Language.
- 2) for any new task for the instruction, do we have all the components needed in the current processor? If not, what shall we add?
- 3) if any new component is added, how do we assemble it into the existing datapath?
- 4) do we need to add any new control signal? if yes, how to set its value for the new instruction?
- 5) do we need to add a new multiplexor for the support of the new instruction? if yes, where do we add it?

Implementing jal

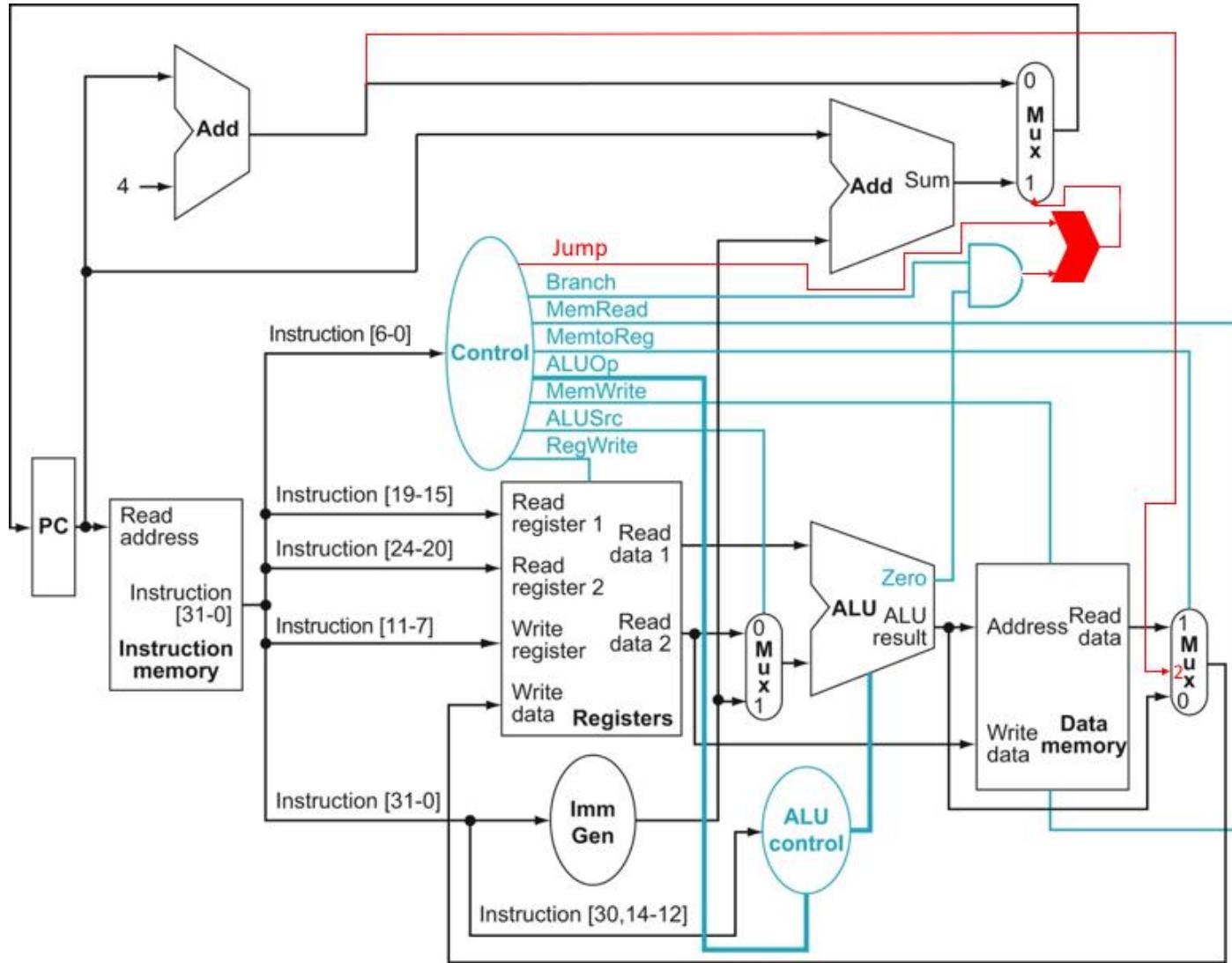


- **R[rd] = PC+4**
- **PC = PC + immed (untangled and shifted left 1 bit)**

Implementing jal



Single-Cycle Datapath Updated with Jal support



Performance of a Single-Cycle Datapath

- **Time needed by functional units:**
 - Memory unit (r/w): 200 ps
 - ALU or adder: 100 ps
 - Register file (r/w): 50 ps
 - No delay for other units
- **Instruction mix: 25% loads, 10% stores, 45% R-type instructions, 20% branches**
- **Compare the performance of the following 2 single cycle implementations**
 - Fixed clock cycle time (the same for all instructions)
 - Variable clock cycle time per instruction

Measuring Performance of a Single Cycle Professor

- Determine pathway of each instruction
- Determine time of each pathway
- Max cost for fixed cycle length
- Actual cost (weighted average) for variable cycle length

Functional Units used by each Instr Class

Instruction class	Instruction Fetch	Register Access	ALU	Memory Access	Register Access
R-type	X	X	X		X
Load	X	X	X	X	X
Store	X	X	X	X	
Branch	X	X	X		

Required Length for each Instr Class

Instruction class	Instruction Memory	Register Read	ALU operation	Data Memory	Register Write	Total (ps)
R-type	200	50	100		50	400
Load	200	50	100	200	50	600
Store	200	50	100	200		550
Branch	200	50	100			350

The clock cycle for a machine with a single clock for all instructions will be determined by the longest instruction, which is 600 ps here

Variable Length Cycles

Instruction class	Instruction Mix	Instruction Memory	Register Read	ALU operation	Data Memory	Register Write	Total (ps)
R-type	45%	200	50	100		50	400
Load	25%	200	50	100	200	50	600
Store	10%	200	50	100	200		550
Branch	20%	200	50	100			350

The average time per instruction with a variable clock is:

$$v = 455\text{ps}$$

Comparing Fixed and Variable Cycle

- $\frac{\text{CPU performance}_{\text{variable clock}}}{\text{CPU performance}_{\text{single clock}}} = \frac{\text{CPU time}_{\text{single clock}}}{\text{CPU time}_{\text{variable clock}}}$
- $\text{CPU time} = \text{IC} \times \text{CPU clock cycle} \times \text{CPI}$
- IC's are the same
- CPI = 1
- $\frac{\text{CPU time}_{\text{single clock}}}{\text{CPU time}_{\text{variable clock}}} = \frac{600}{455} = 1.32$
- **Variable Clock Cycle is 1.32 times of Fixed Clock Cycle**

Single-Cycle Use of Components

add	addi	beq	lw	sw
20%	20%	25%	25%	10%

- In what fraction of all cycles is the data memory used?
- In what fraction of all cycles is the input of the immediate-gen circuit needed (addi uses immediate-gen, what other ops use it)?
- What is the immediate-gen circuit doing in the cycles in which its input is not needed?