EW-2 Project

Processor Design

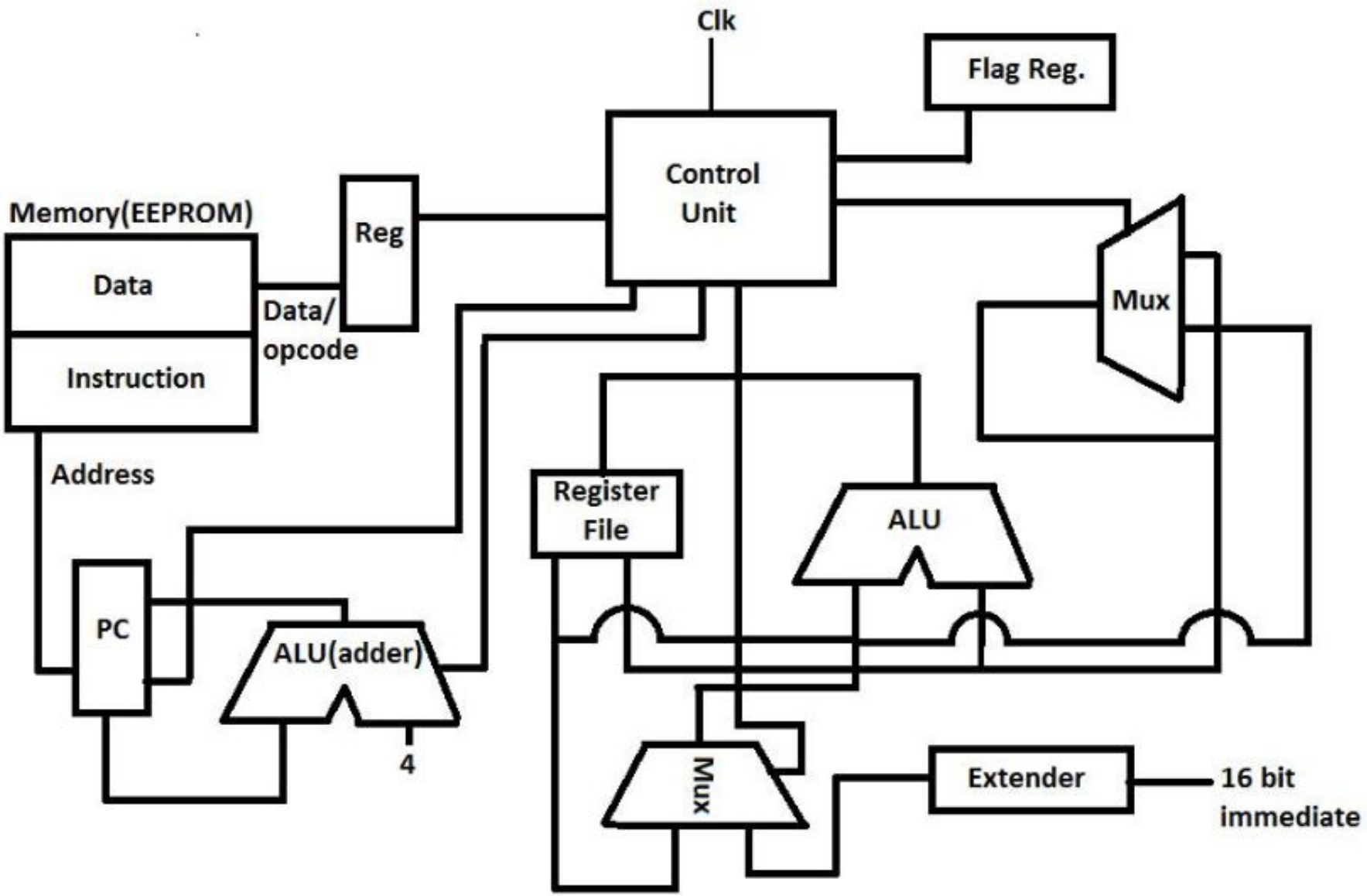
- Dr. Rahul Shrestha

Abhinav Moudgil
Vinamra Benara
Roopal Nahar
Nikita Kad

Aim and Goal

- 8-bit Processor
- Implemented as MCU
- FSM Control Unit
- Single cycle datapath (one instruction executed in one cycle)
- External EEPROM as Instruction Memory
- Output displayed on LCD module
- Will be able to run programs
- Bubble sort and Fibonacci series

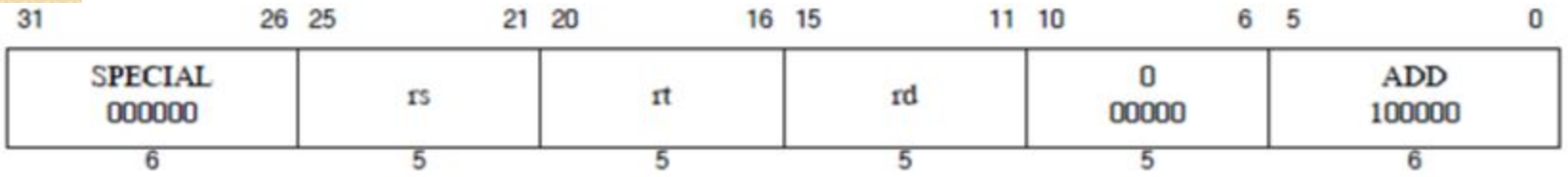
Architecture



Instruction Set

- Add
- Sub
- Load
- Store
- Branch
- Jump
- OR

ADD/SUB Instruction



ADD rd, rs, rt

=>R[rd] = R[rs] + R[rt]

It is a 32-bit 2's Complement Addition

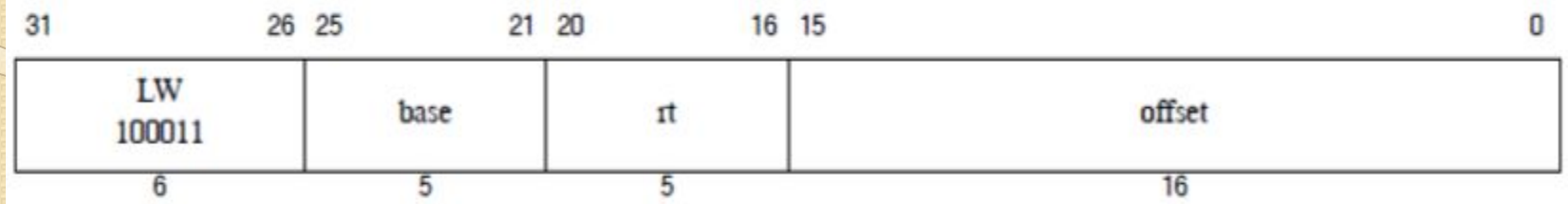
Similarly, for subtraction : (function is 100010)

SUB rd, rs, rt

=>R[rd] = R[rs] - R[rt]

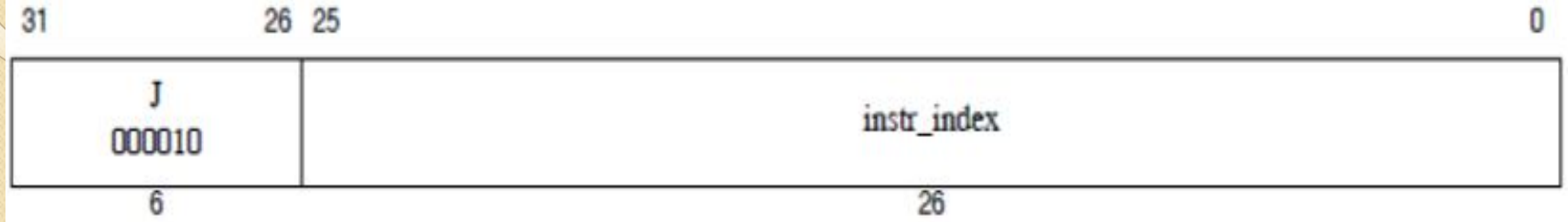
32-bit signed subtraction

Load/Store Instruction



LW rt, offset(base)
=>vdddr = sign_extend(offset) + R[base]
R[rt] = Mem[vaddr]
Similarly for store: (function is 101011)
SW rt, offset(base)
=>vdddr = sign_extend(offset) + R[base]
Mem[vaddr] = R[rt]

J instruction

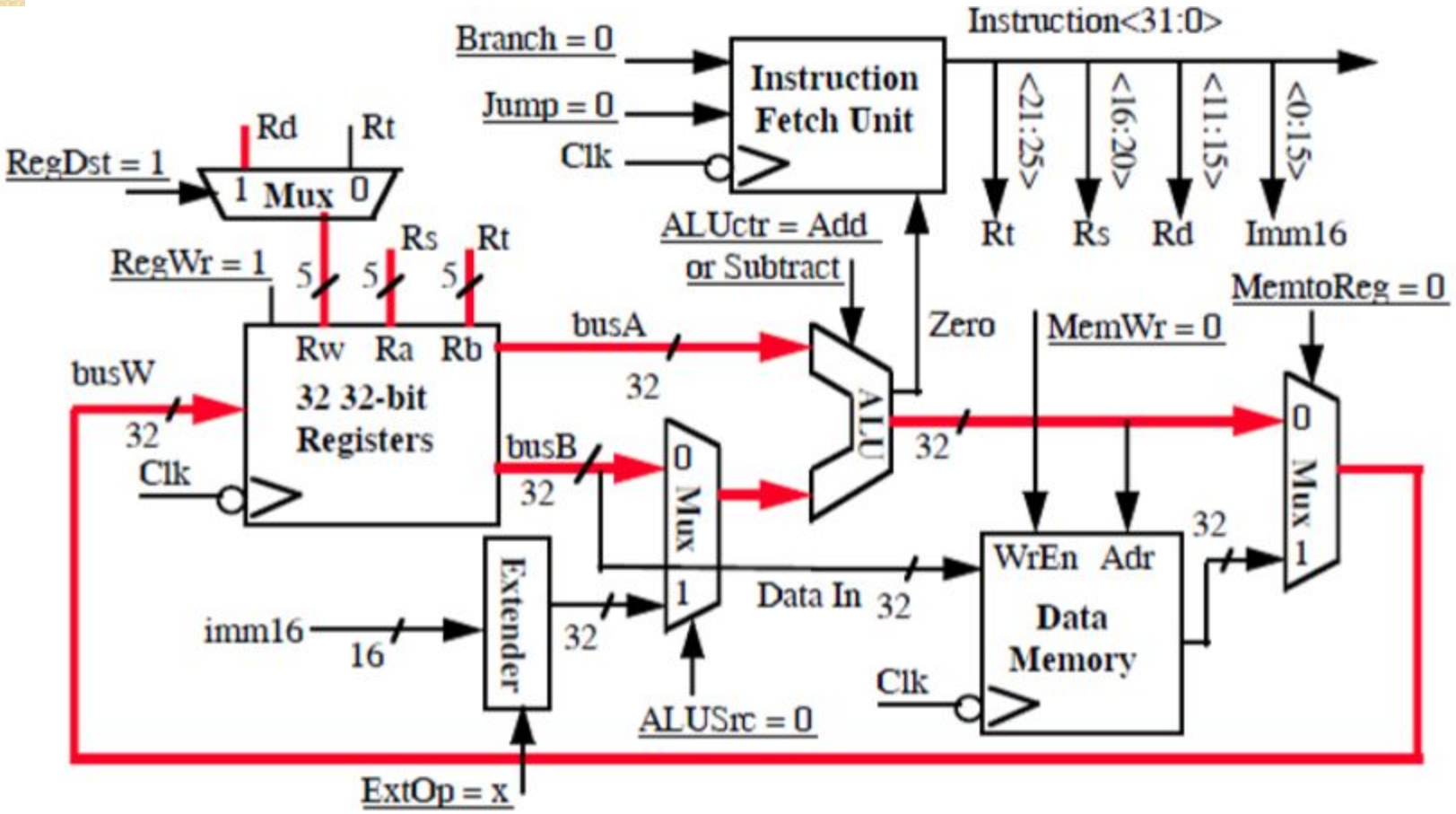


J target

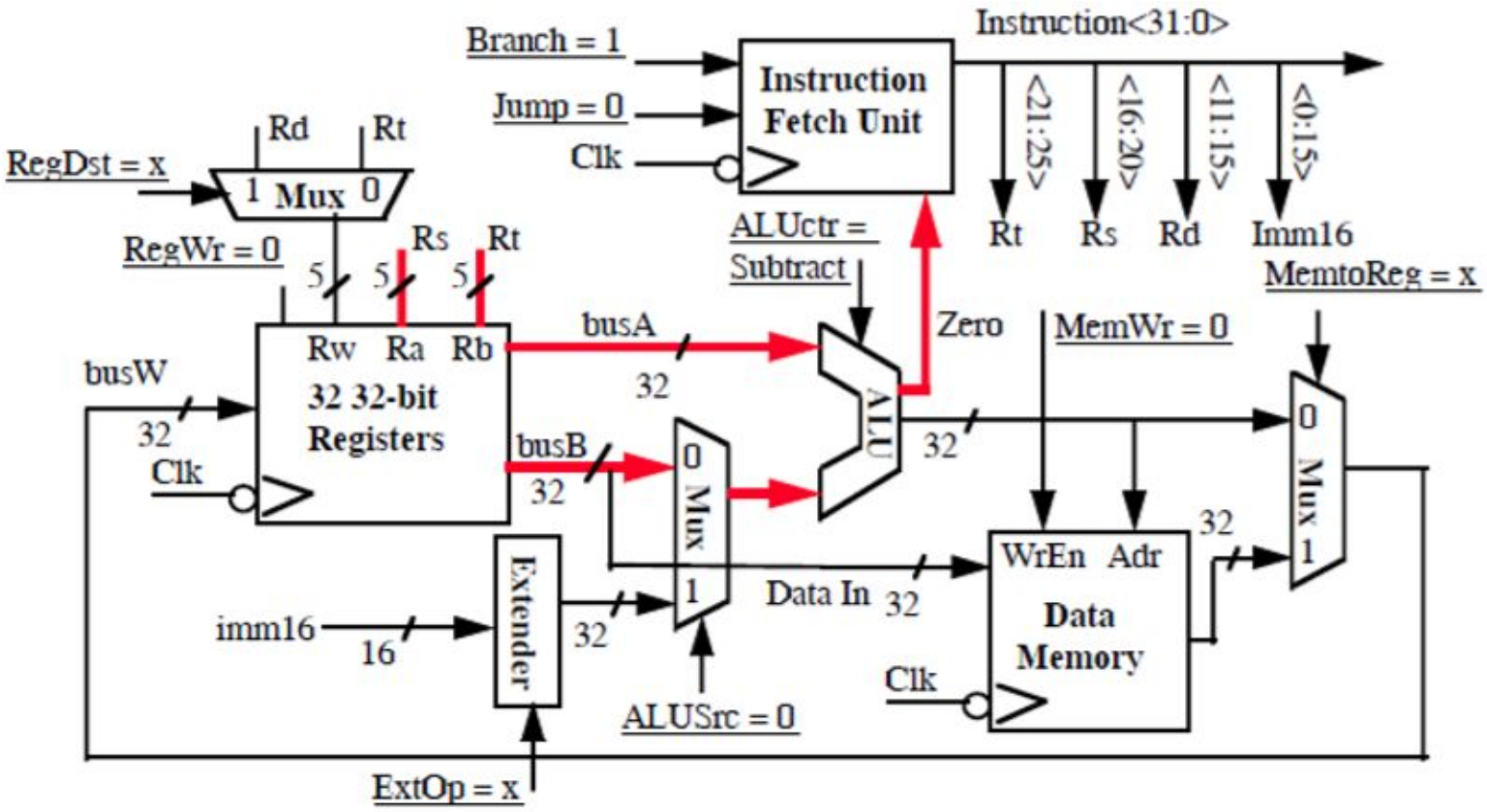
Target Address

- Lower 28 bits: instr_index||00
- Upper Four Bits: Bits 31, 30, 29, 28 of the address of the Jump Instruction.

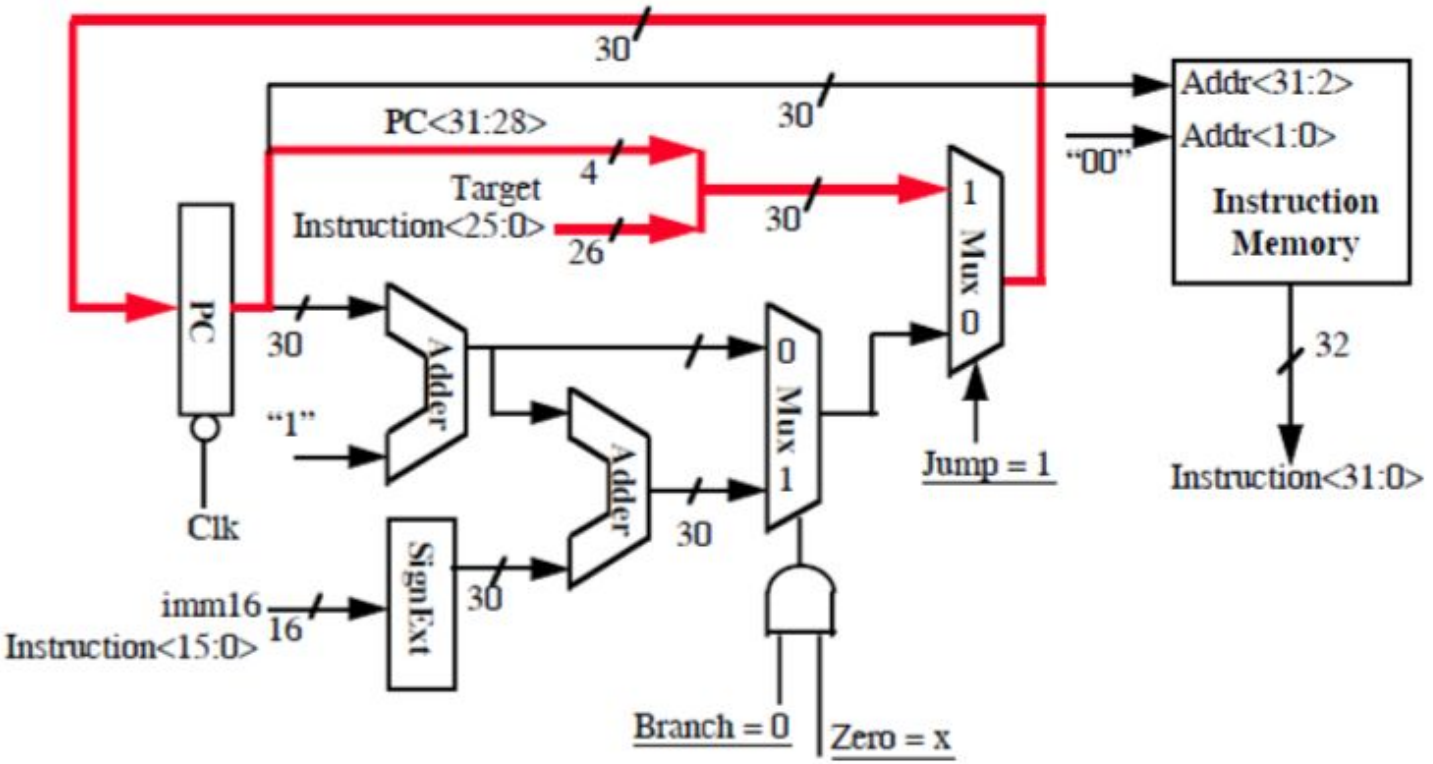
The Single Cycle Data path during Add and Subtract



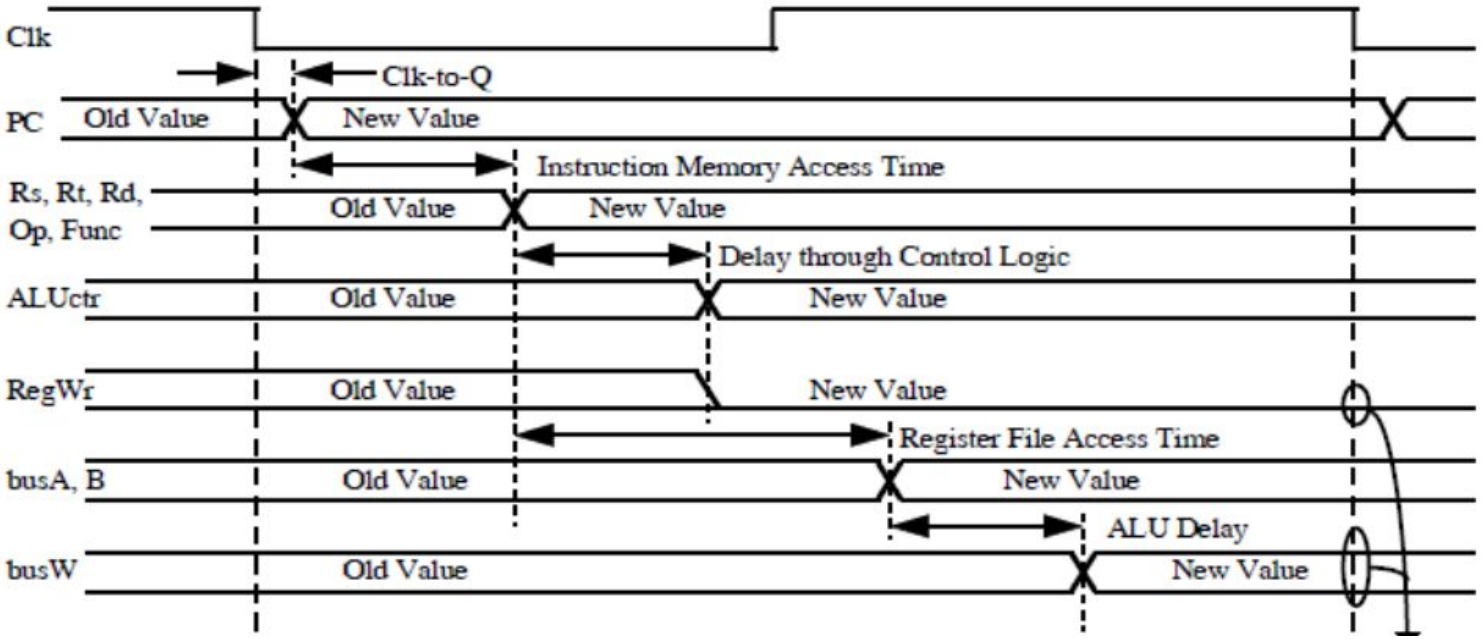
Single Cycle Datapath during Branch



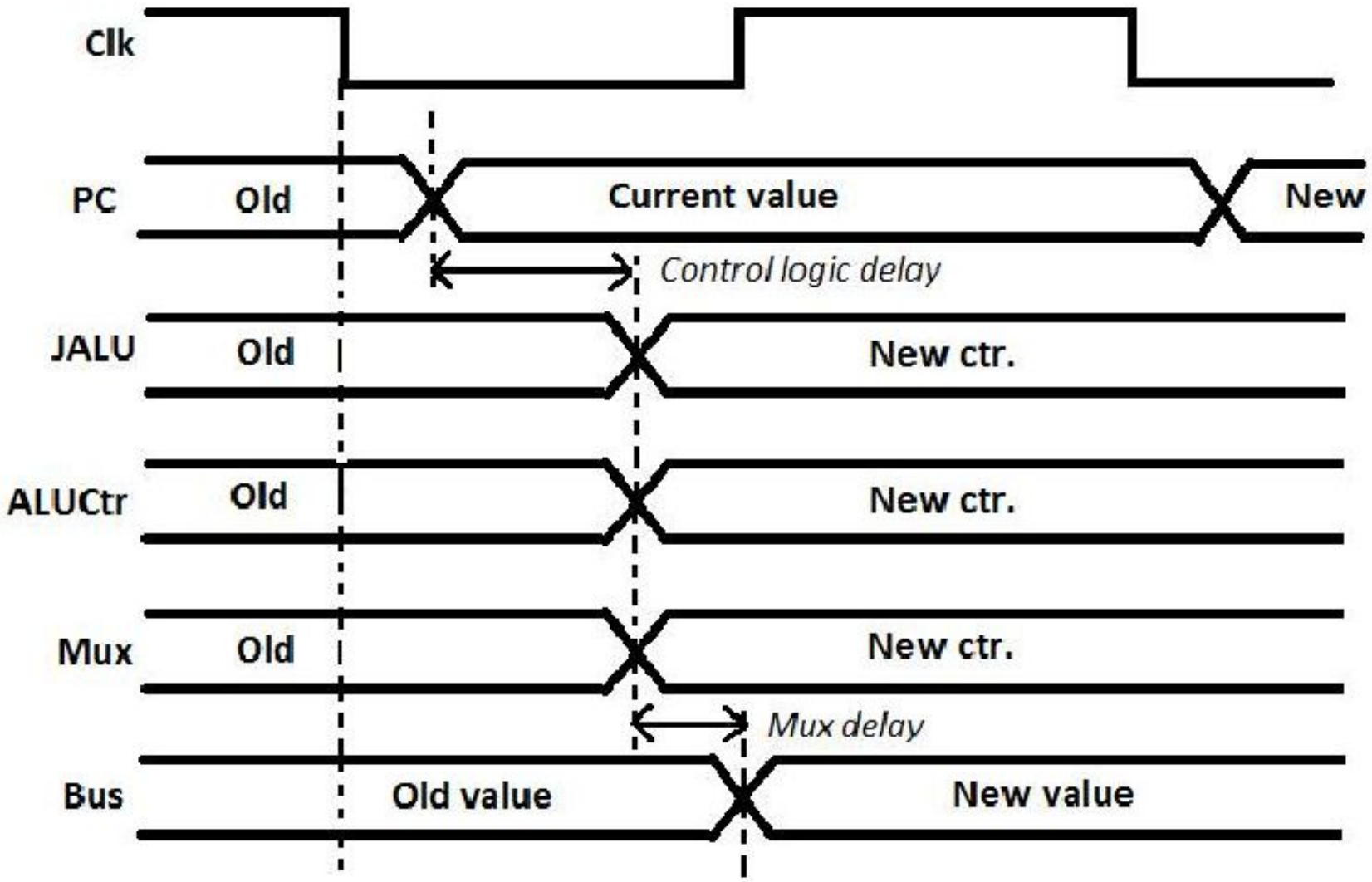
Instruction Fetch Unit at the End of Jump



Timing Diagram: R-Type



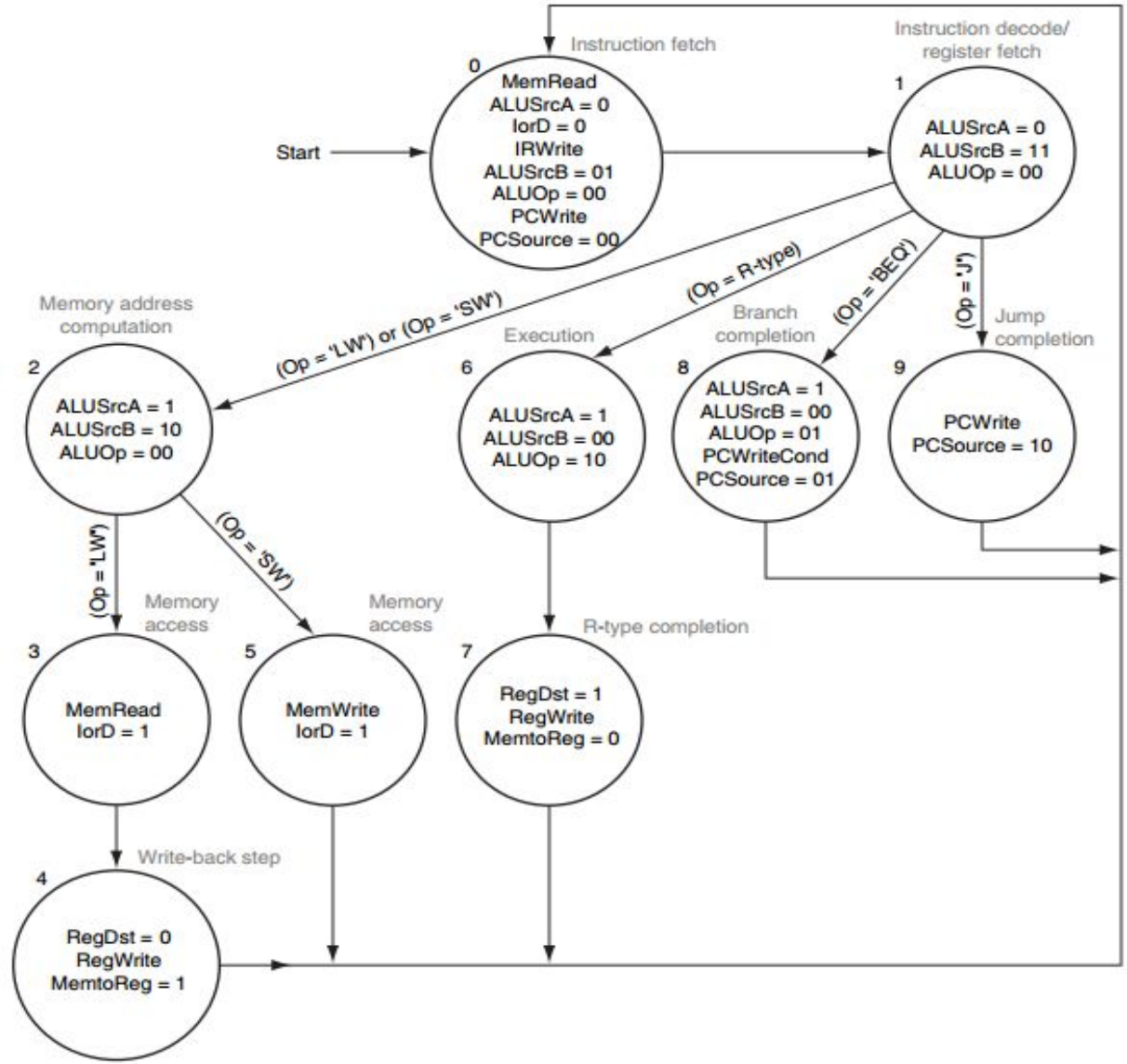
Timing Diagram: Jump Type



Control Unit: Single Cycle

op	00 0000	00 1101	10 0011	10 1011	00 0100	00 0010
	R-type	**ori**	**lw**	**sw**	**beq**	**jump**
RegDst	1	0	0	x	x	x
ALUSrc	0	1	1	1	0	x
MemtoReg	0	0	1	x	x	x
RegWrite	1	1	1	0	0	0
MemWrite	0	0	0	1	0	0
Branch	0	0	0	0	1	0
Jump	0	0	0	0	0	1
ExtOp	x	0	1	1	x	x
ALUop (Symbolic)	"R-type"	Or	Add	Add	Subtract	xxx
ALUop <2>	1	0	0	0	0	x
ALUop <1>	0	1	0	0	0	x
ALUop <0>	0	0	0	0	1	x

Control Unit: Multi-Cycle



Thank you