CSE 31

Computer Organization

Lecture 24 – CPU Design (3)

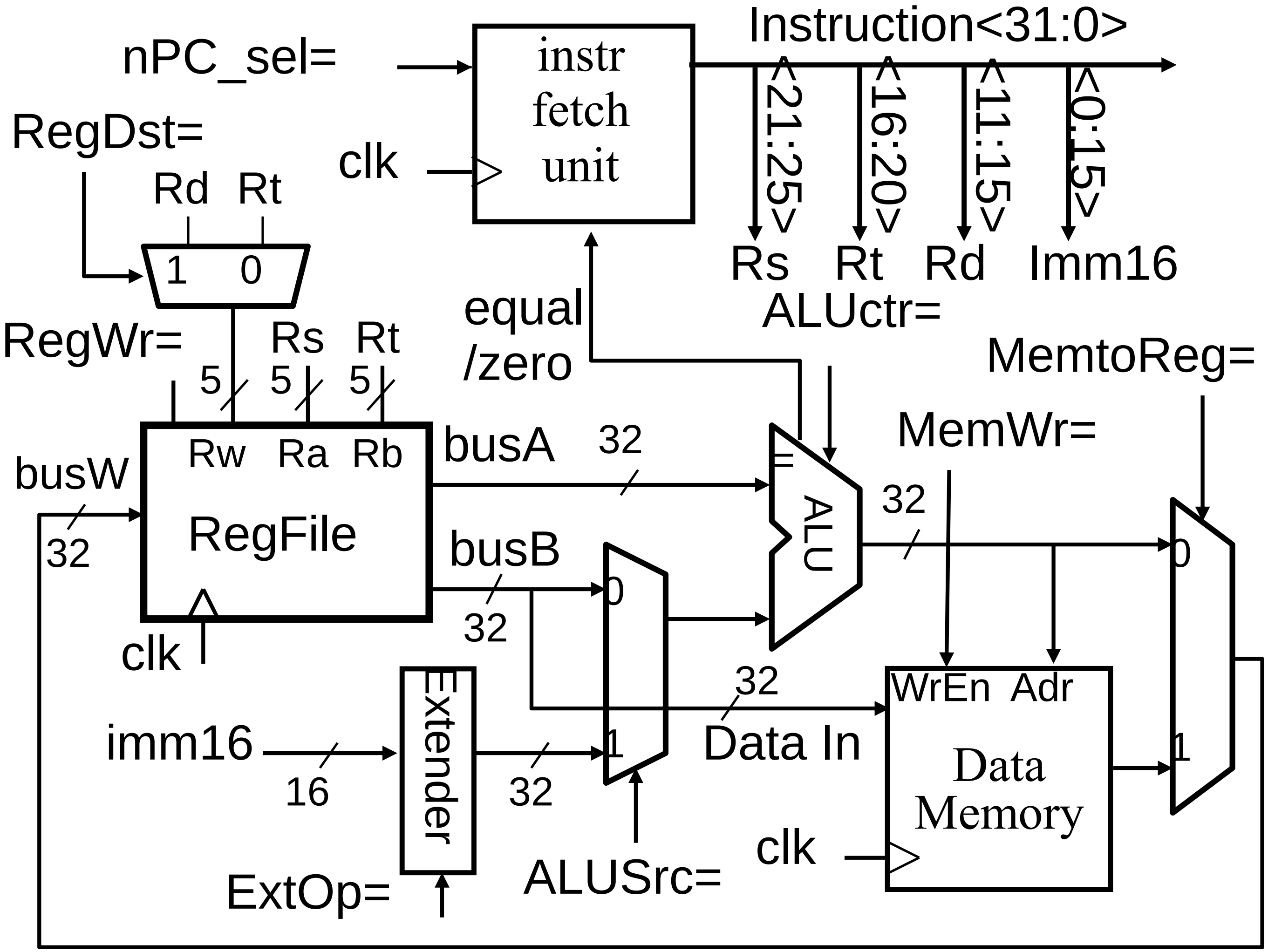
Announcement

- No lab this week
  - Project #1/2 grading during lab
- HW #7 at zyBooks
  - Due Monday(12/10) at 11:59pm
- Project #2
  - Due Monday (12/3)
    - Don’t start late, you won’t have time!
- Course evaluation online
  - Fill out by 12/6 (Thursday)
- Reading assignment
  - Chapter 5.7 – 5.11 of zyBooks (Reading Assignment #6)
    - Make sure to do the Participation Activities
    - Due Wednesday (12/5)

Announcement

- zyBooks assignment Re-dos
  - Re-submit at most 5 reading assignments or HW (zyBooks only)
  - Email to me (not your TAs)
    - Include your name, assignment numbers
    - (Monday) 12/10 at 11:59pm, no extension
  - Fill out online evaluation by 12/6, tomorrow (70% of class)
- Final Exam
  - 12/11 (Tuesday), 11:30 – 2:30pm
  - Cover all
  - Practice exam in CatCourses
  - Closed book
  - 2 sheet of note (8.5” x 11”)
  - MIPS reference sheet will be provided
  - Review: 12/10 (Monday) 1-3pm, COB 113

Single Cycle Datapath



The Add Instruction

31 26 21 16 11 6 0

op	rs	rt	rd	shamt	funct
6 bits	5 bits	5 bits	5 bits	5 bits	6 bits

`add rd, rs, rt`

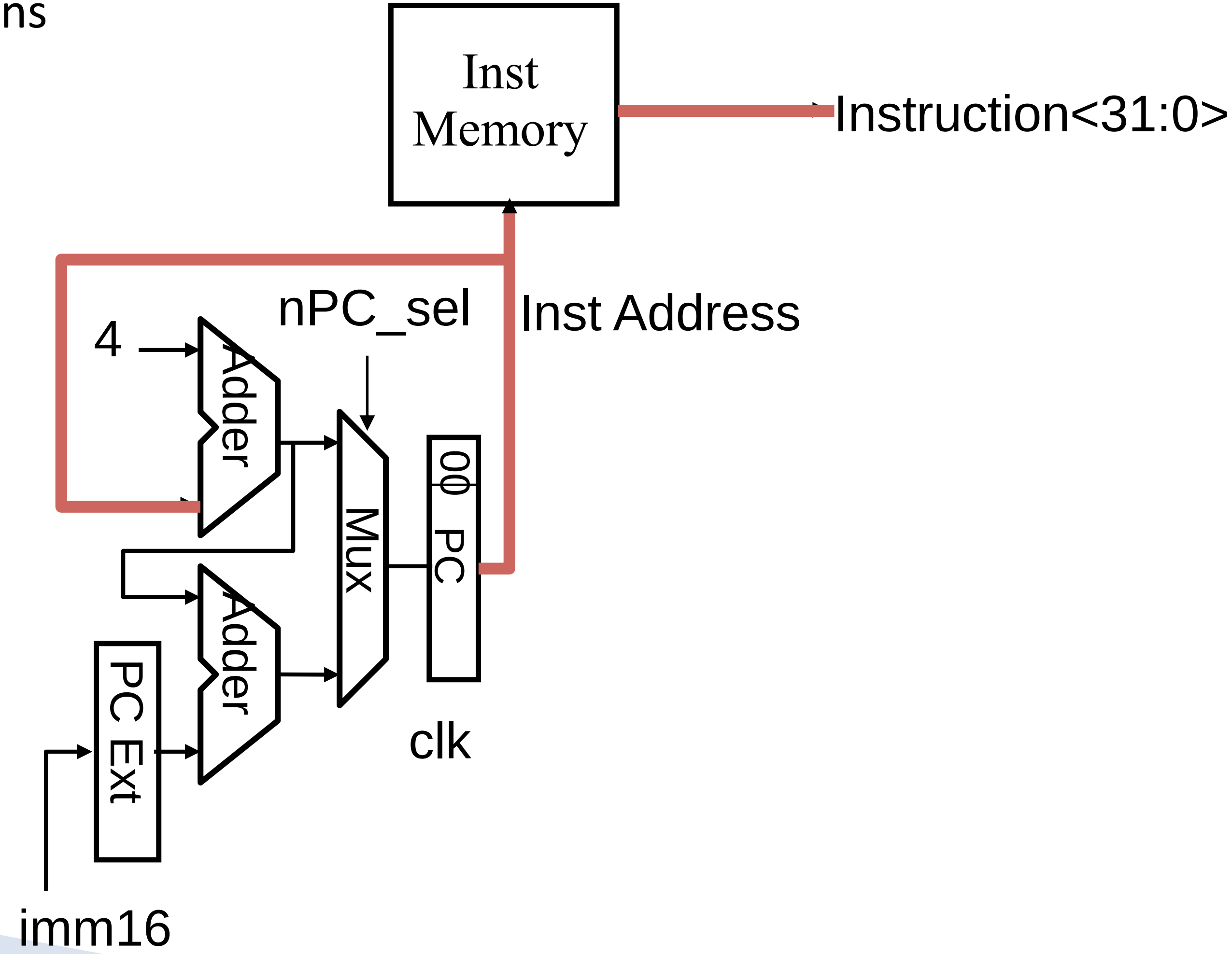
- MEM[PC] Fetch the instruction from memory
- R[rd] = R[rs] + R[rt] The actual operation
- PC = PC + 4 Calculate the next instruction’s address

Instruction Fetch Unit start of Add

- Fetch the instruction from Instruction memory:

  Instruction = MEM[PC]

  - same for all instructions

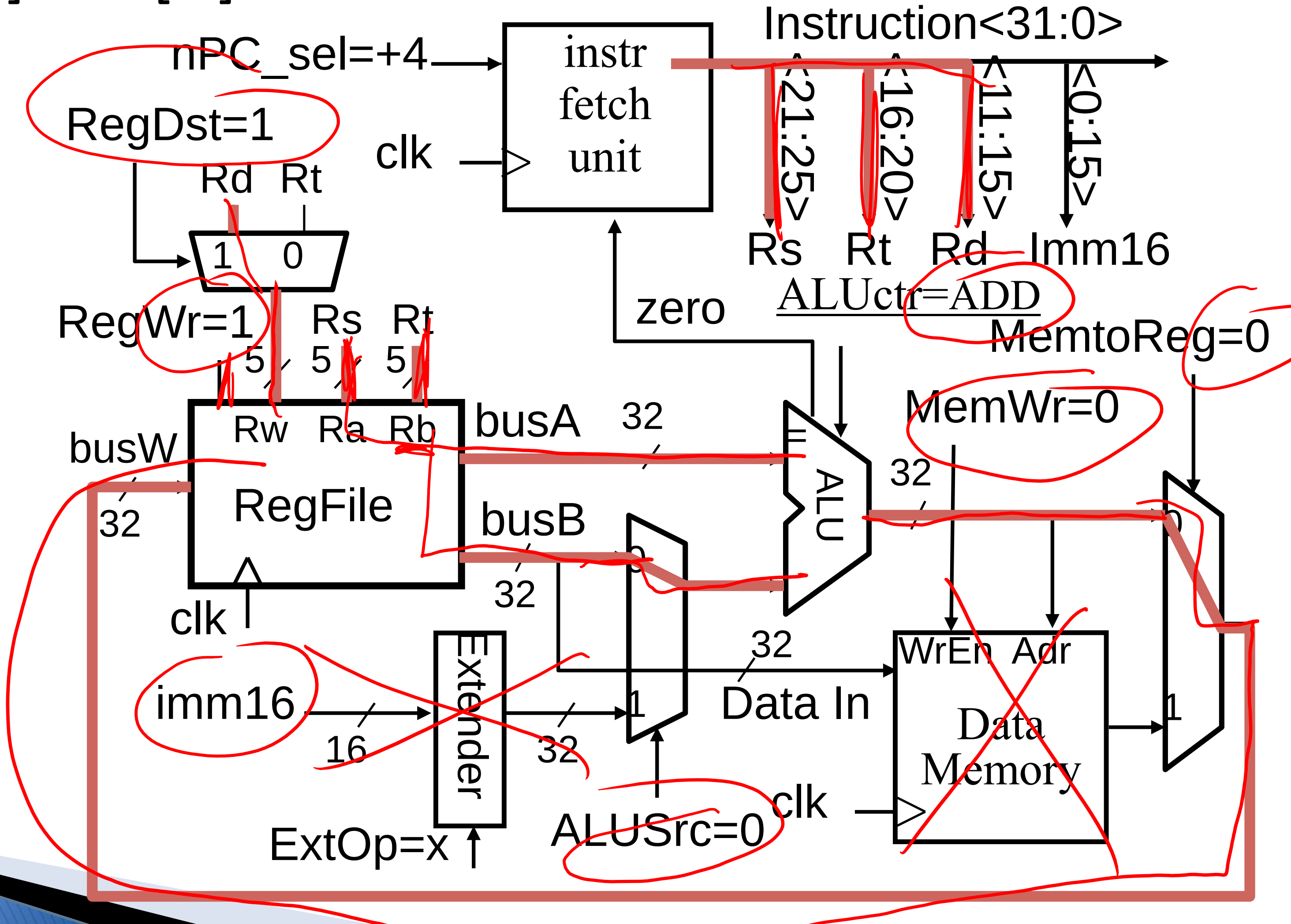


The Single Cycle Datapath during Add

31 26 21 16 11 6 0

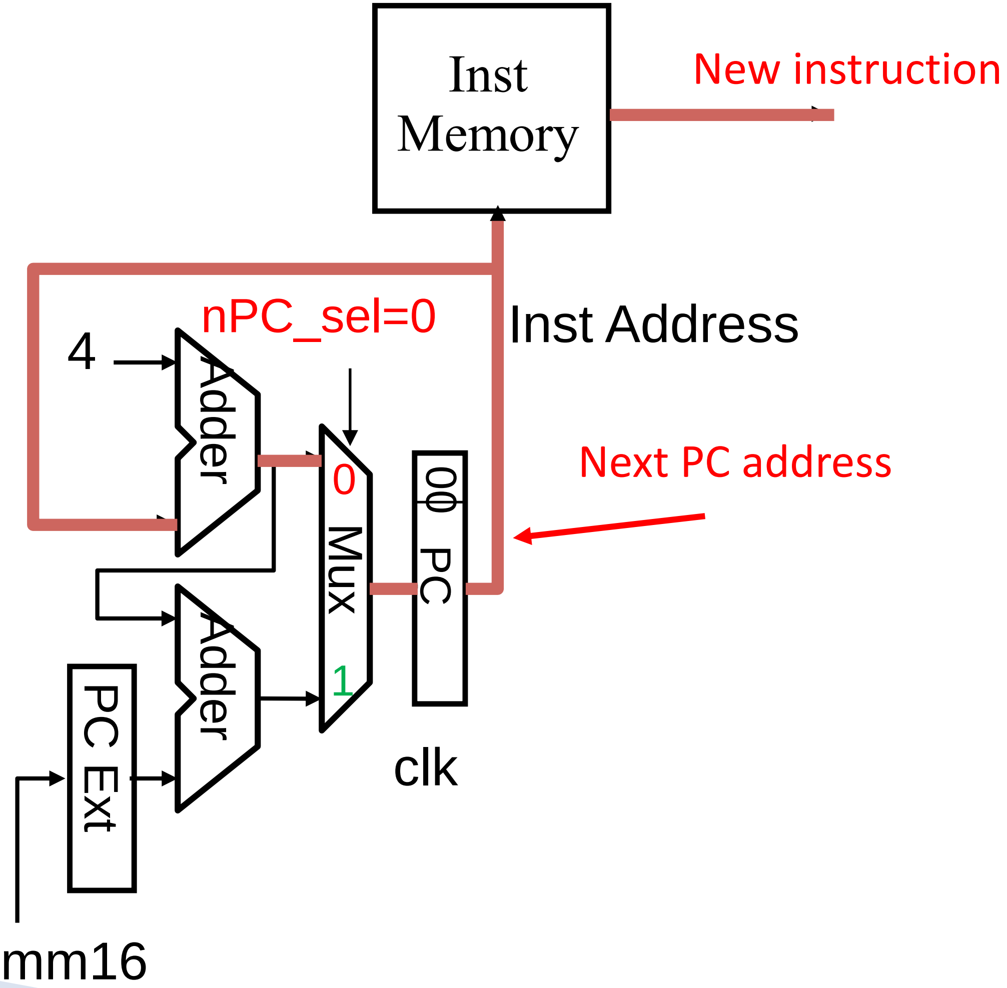
op	rs	rt	rd	shamt	funct

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



Instruction Fetch Unit end of Add

- PC = PC + 4
  - This is the same for all instructions except: Branch and Jump

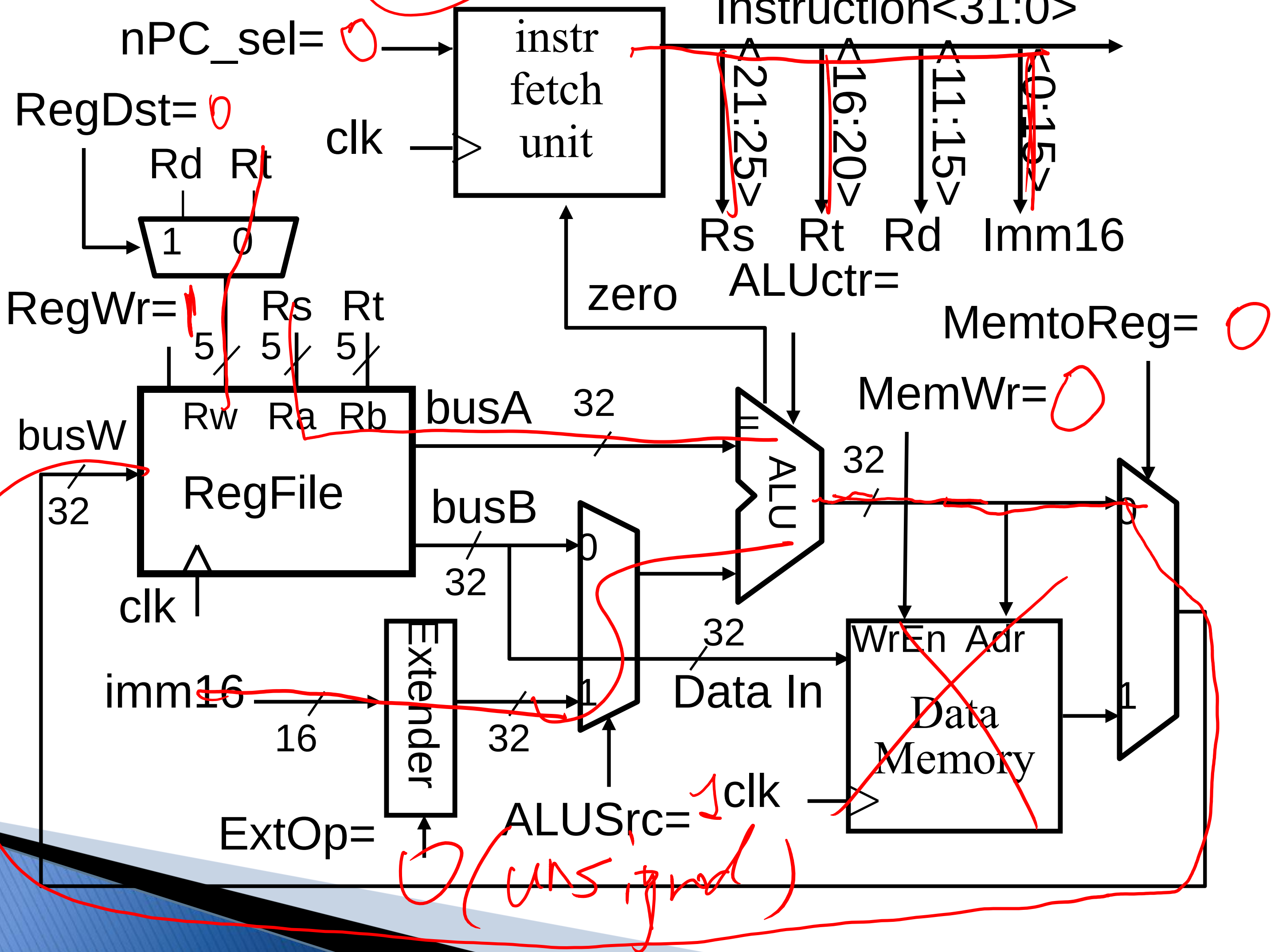


Single Cycle Datapath for Ori

31 26 21 16 0

op	rs	rt	immediate

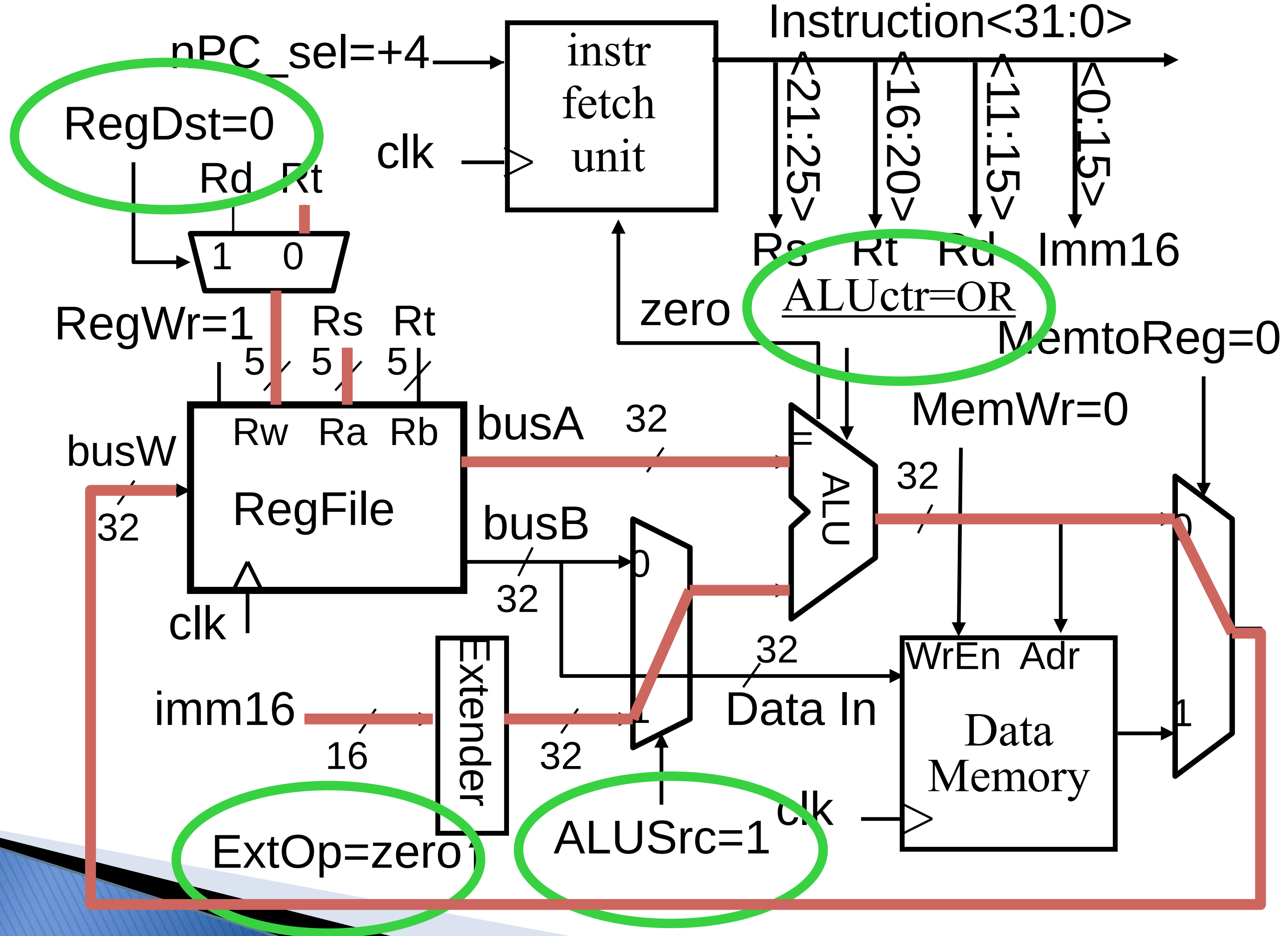
- R[rt] = R[rs] OR ZeroExt[Imm16]



Single Cycle Datapath for Ori

31	26	21	16	0
op	rs	rt	immediate	

- R[rt] = R[rs] OR ZeroExt[Imm16]

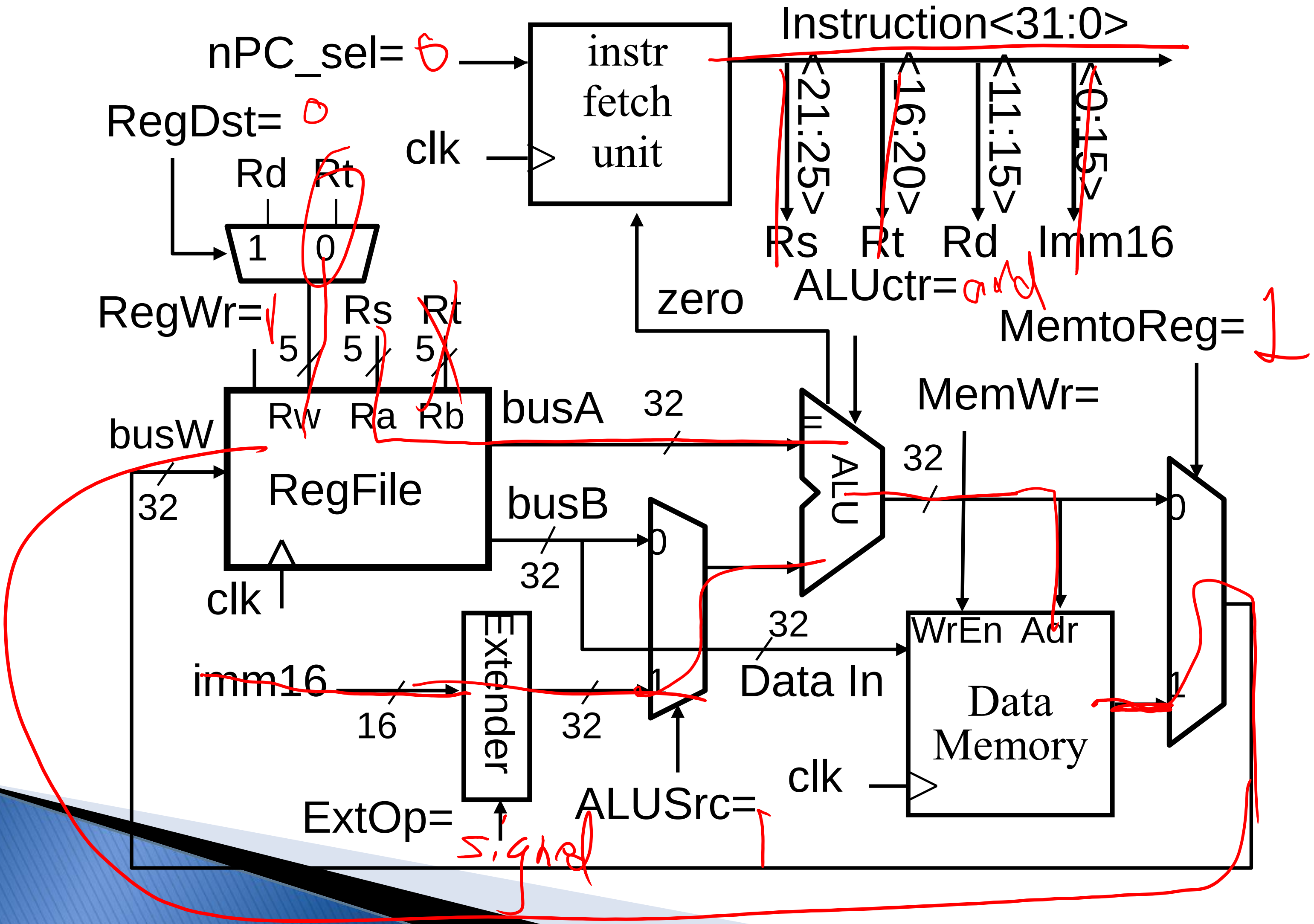


Single Cycle Datapath for LW

31 26 21 16 0

op	rs	rt	immediate

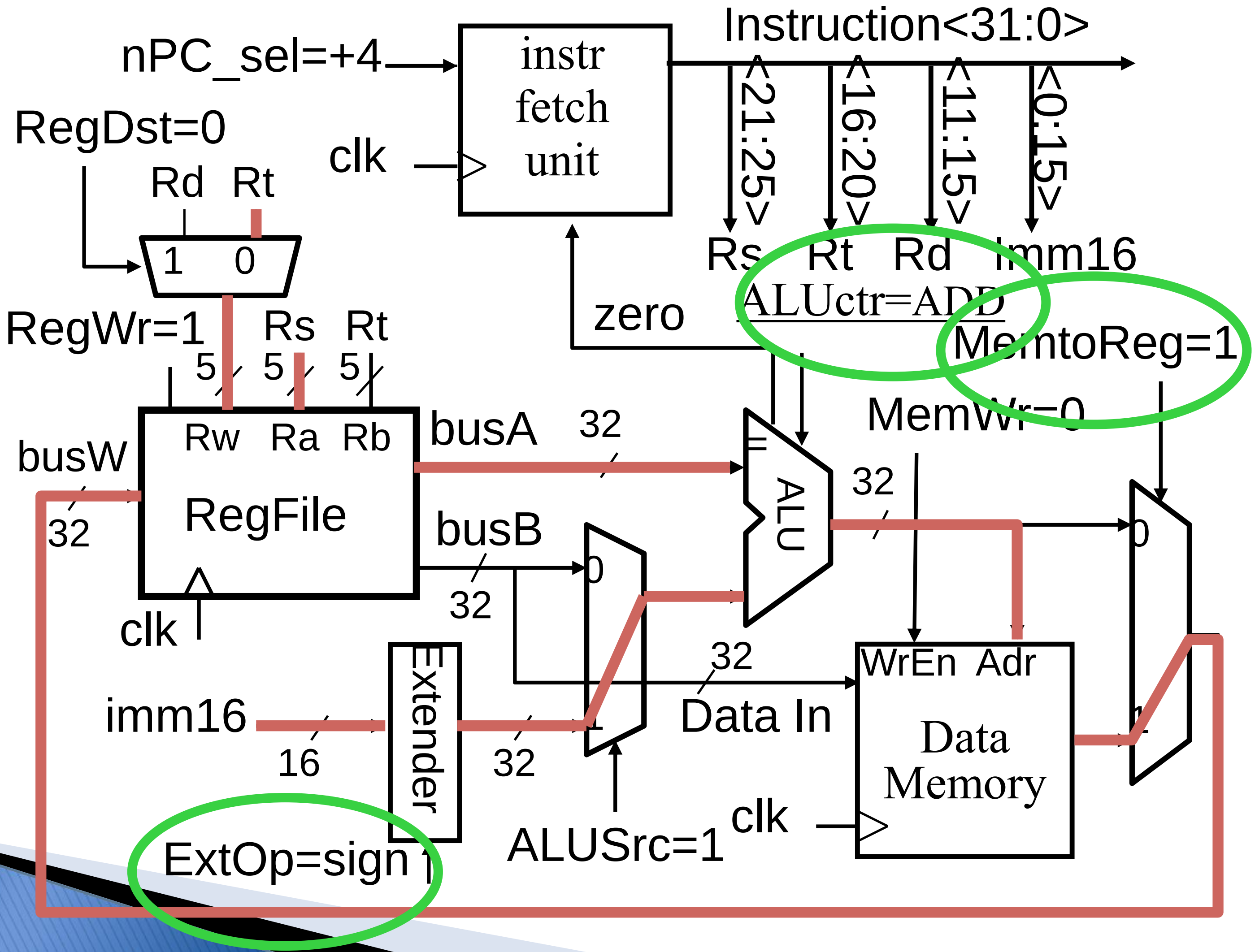
- R[rt] = Data Memory {R[rs] + SignExt[imm16]}



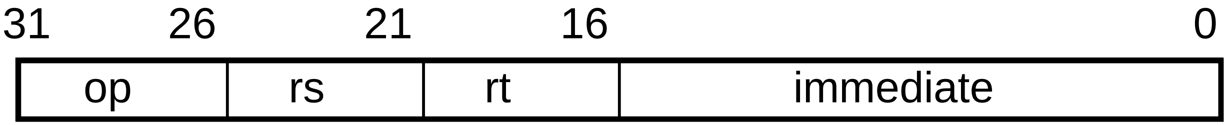
Single Cycle Datapath for LW

31	26	21	16 0
op	rs	rt	immediate

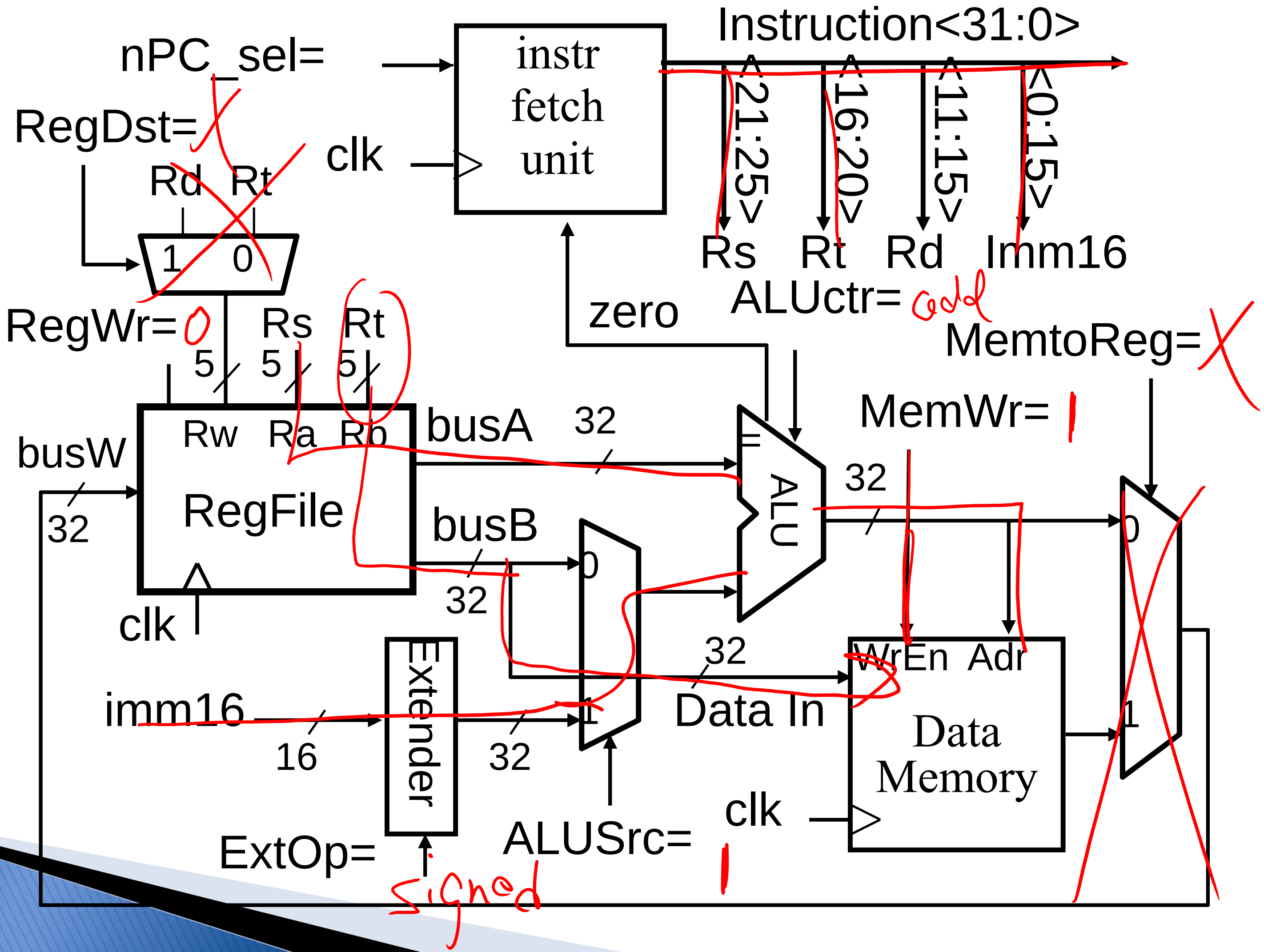
- R[rt] = Data Memory {R[rs] + SignExt[imm16]}



Single Cycle Datapath for SW



- Data Memory {R[rs] + SignExt[imm16]} = R[rt]



nPC_sel=

RegDst= X

RegWr= 0

ALUctr= add

MemtoReg= X

MemWr= 1

ALUSrc= 1

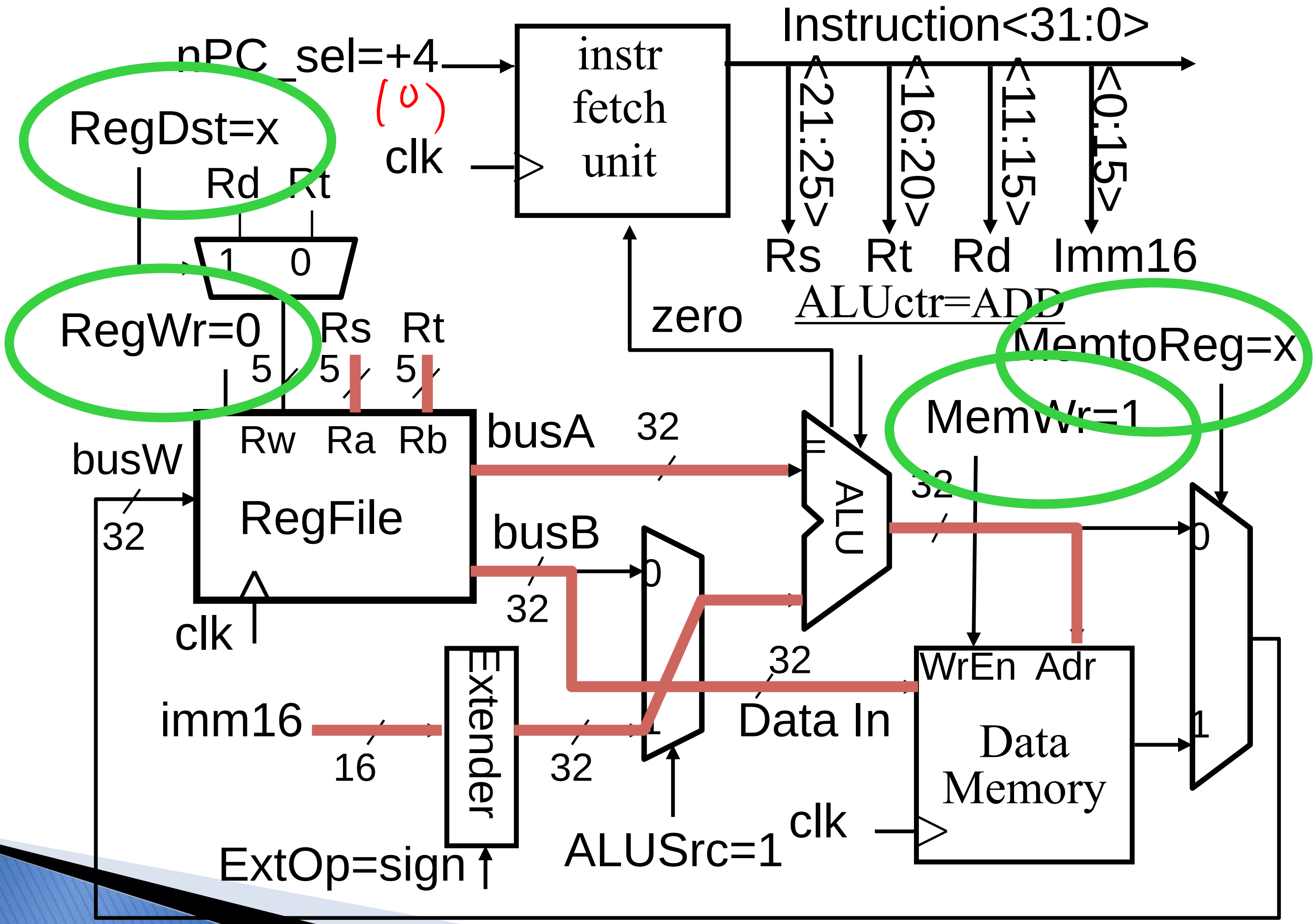
ExtOp= signed

Single Cycle Datapath for SW

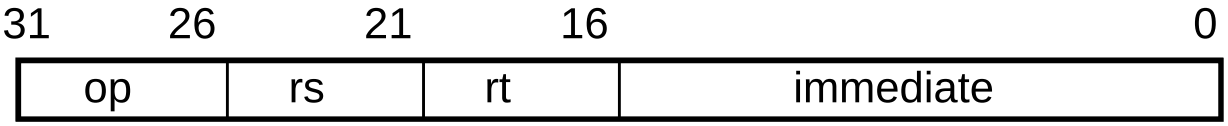
31 26 21 16 0

op	rs	rt	immediate

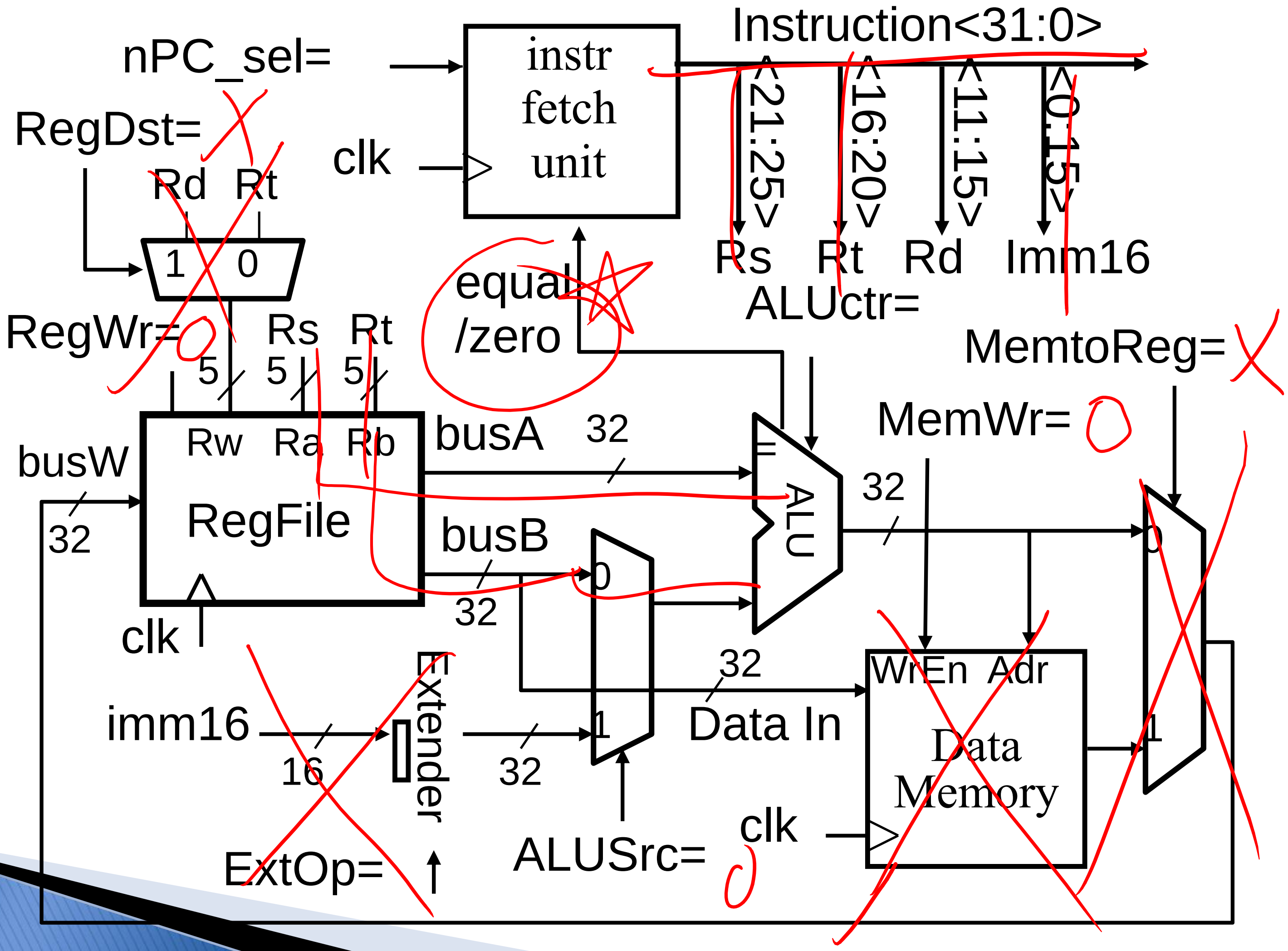
- Data Memory {R[rs] + SignExt[imm16]} = R[rt]



Single Cycle Datapath for Branch



- if (R[rs] - R[rt] == 0) then Zero = 1 ; else Zero = 0

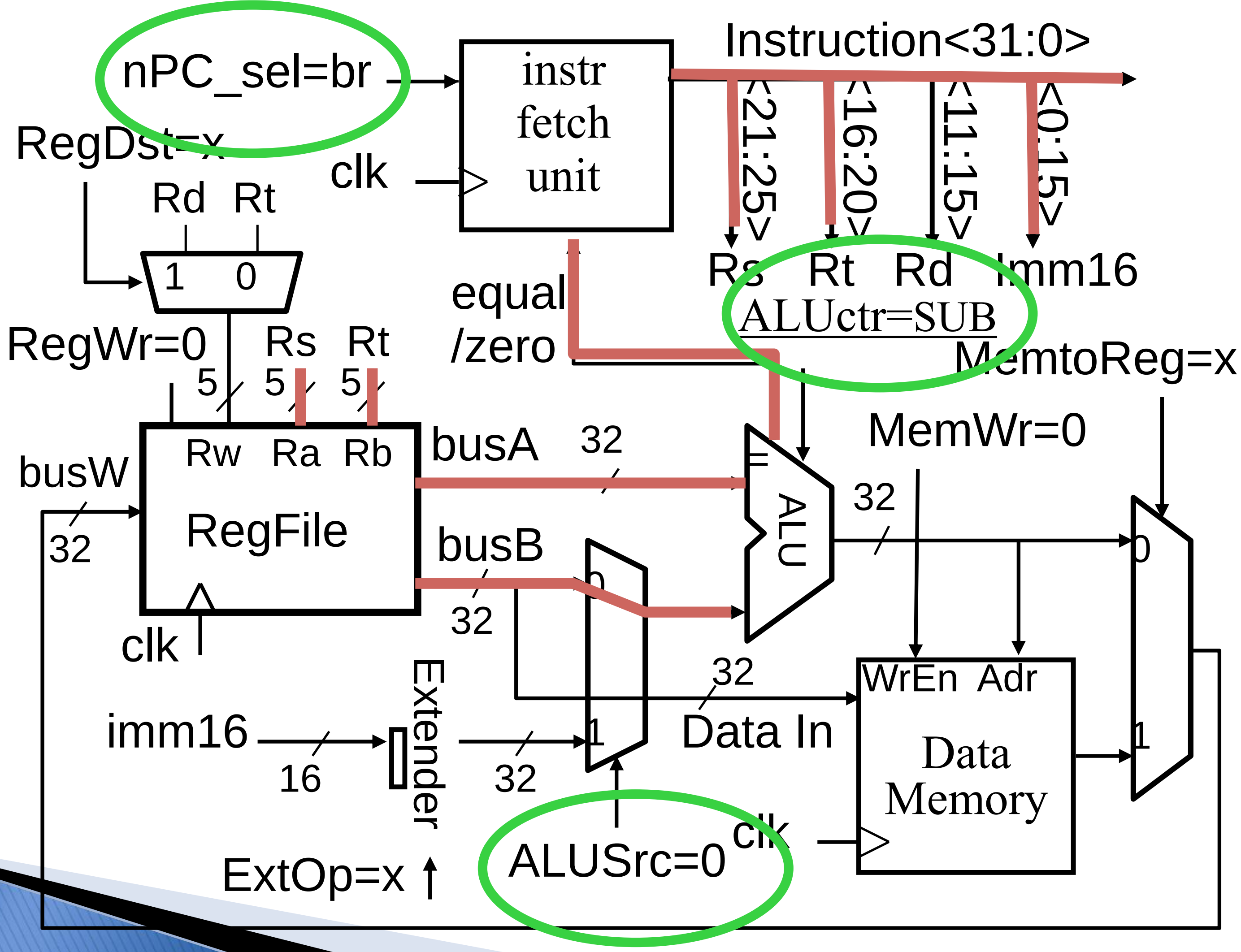


Single Cycle Datapath for Branch

31 26 21 16 0

op	rs	rt	immediate

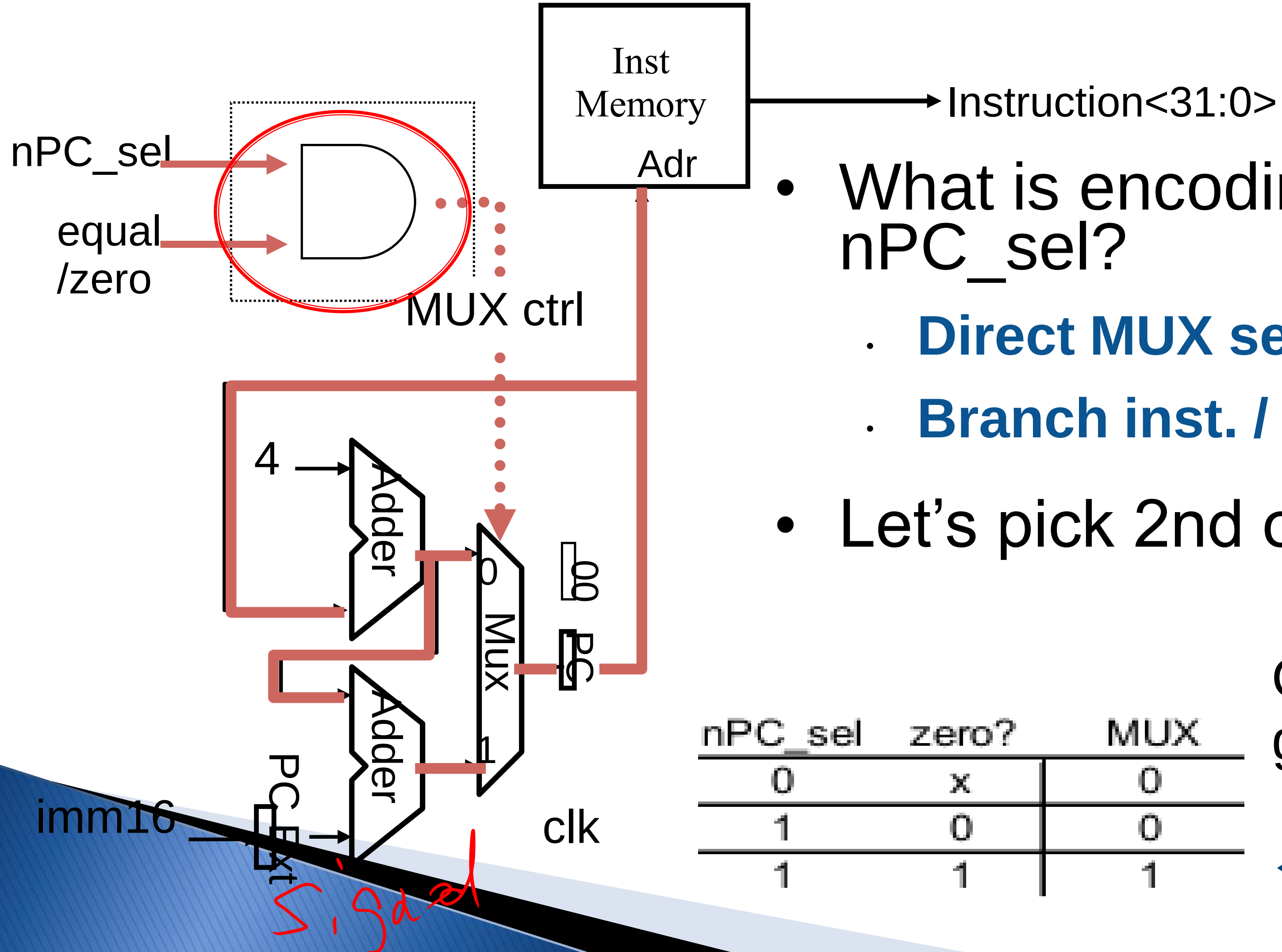
- if (R[rs] - R[rt] == 0) then Zero = 1 ; else Zero = 0



Instruction Fetch Unit end of Branch

31	26	21	16	0
op	rs	rt	immediate	

- if (Zero == 1) then PC = PC + 4 + SignExt[imm16]*4 ; else PC = PC + 4



- What is encoding of nPC_sel?
  - **Direct MUX select?**
  - **Branch inst. / not branch**
- Let’s pick 2nd option

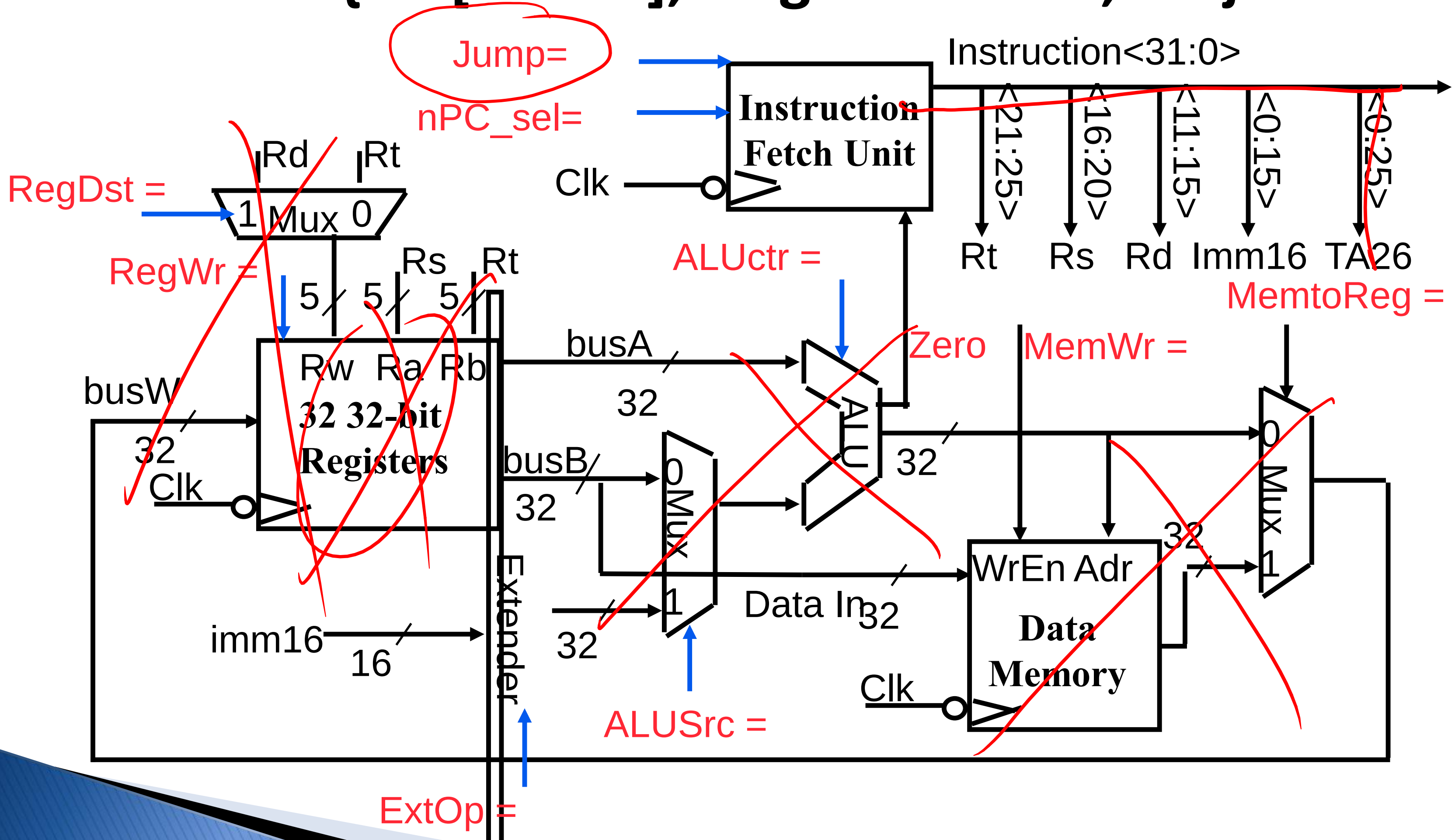
nPC_sel	zero?	MUX
0	x	0
1	0	0
1	1	1

Q: What logic gate?

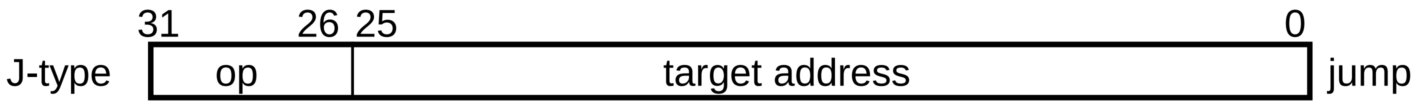
The Single Cycle Datapath during Jump

	31 — 26	25 — 0	
J-type	op	target address	jump

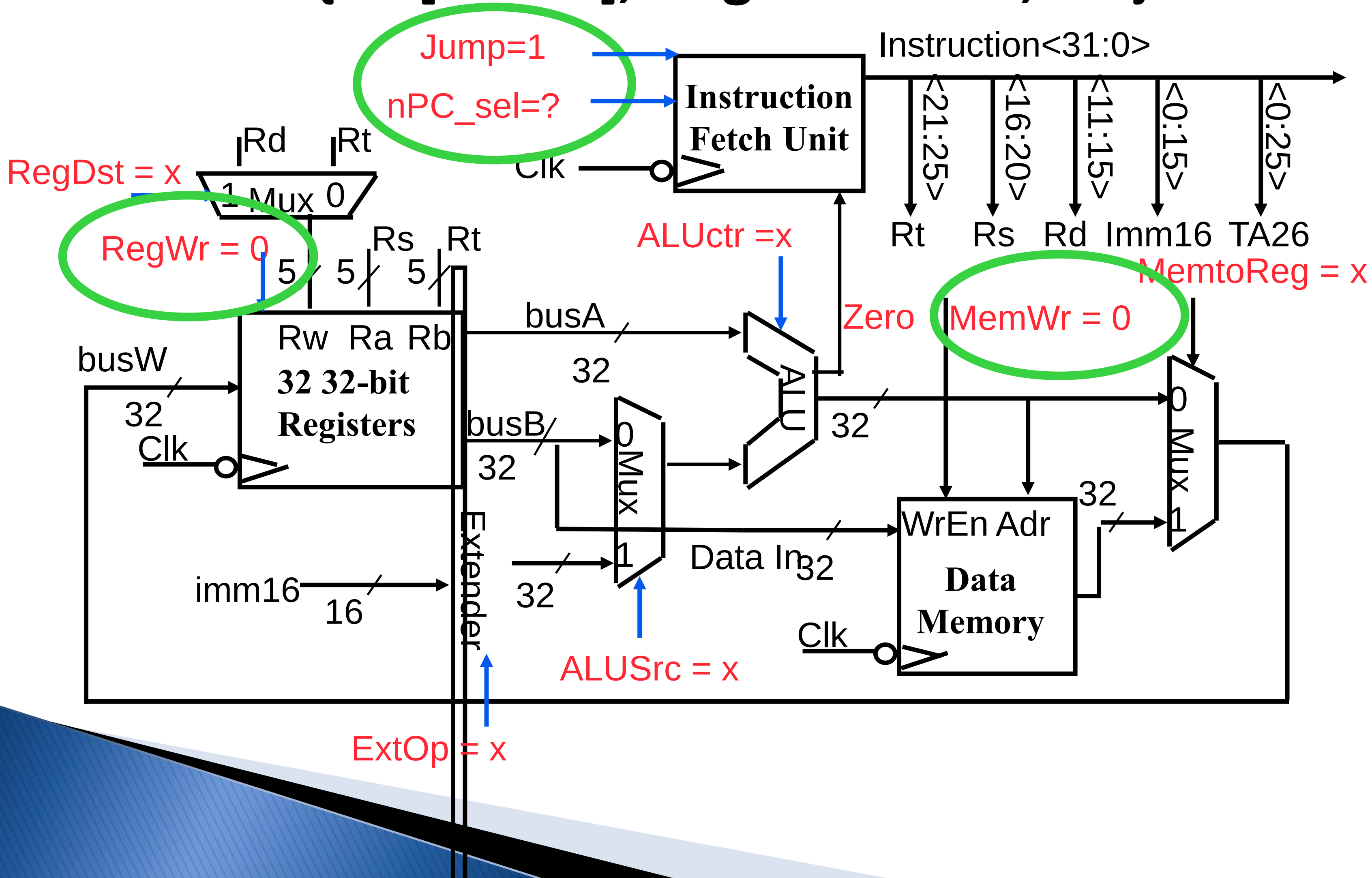
- **New PC = { PC[31..28], target address, 00 }**



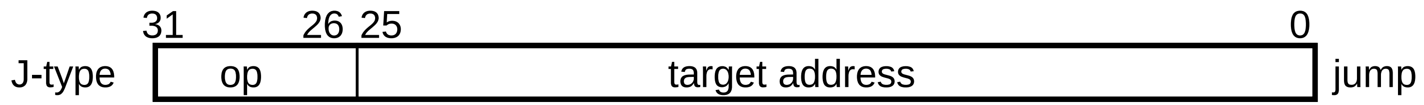
The Single Cycle Datapath during Jump



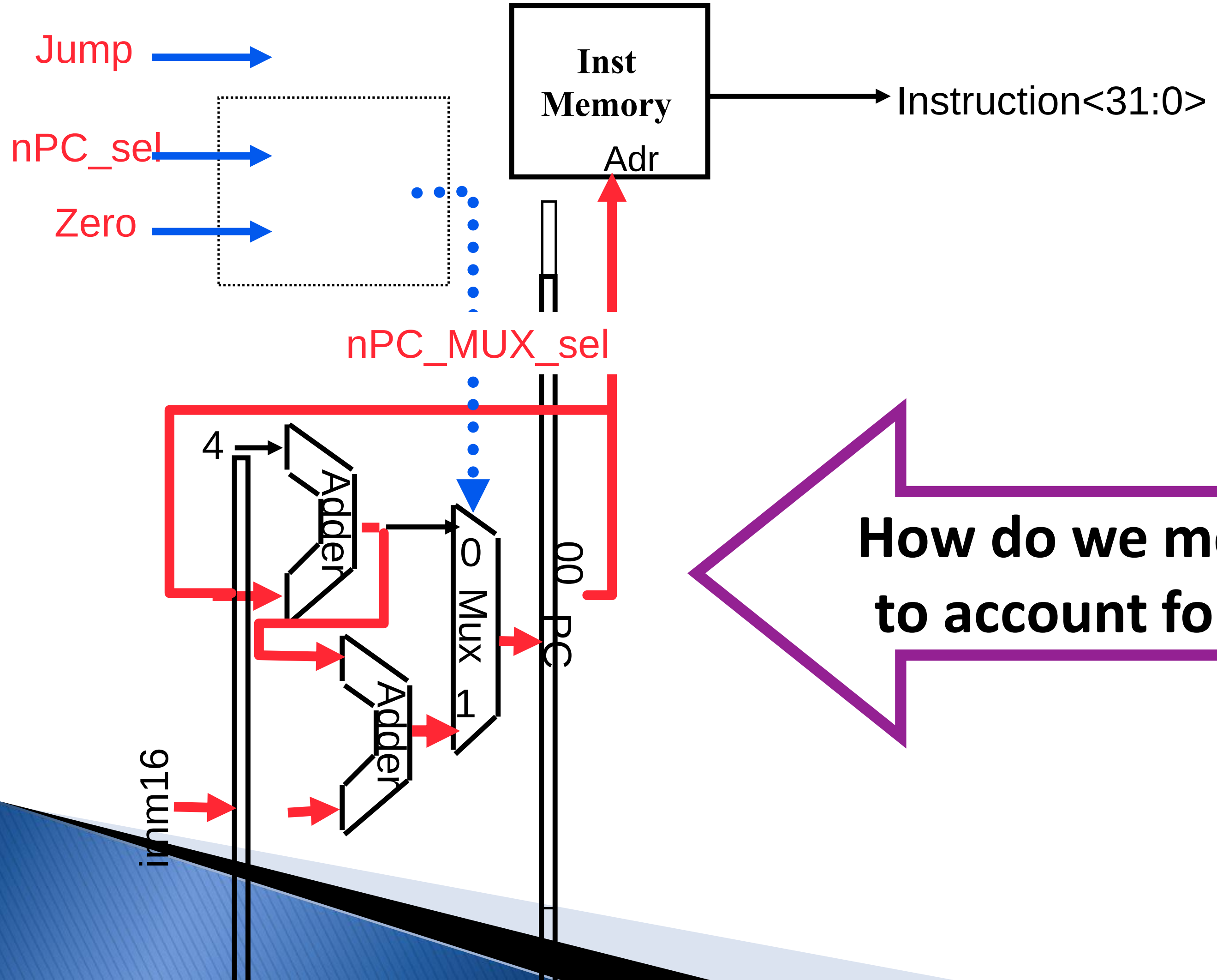
- **New PC = { PC[31..28], target address, 00 }**



Instruction Fetch Unit at the End of Jump



- **New PC = { PC[31..28], target address, 00 }**

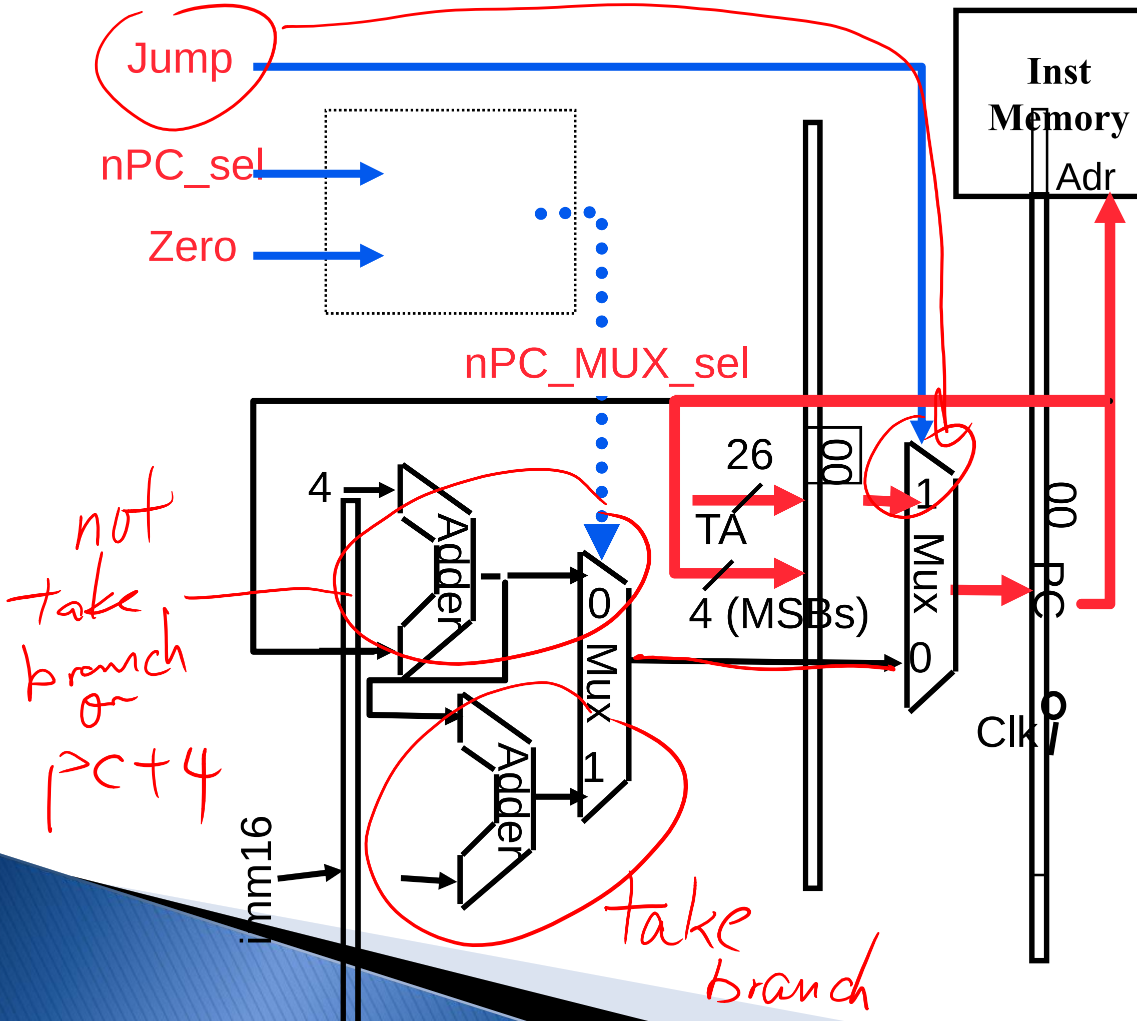


How do we modify this to account for jumps?

Instruction Fetch Unit at the End of Jump

	31 … 26	25 … 0	
J-type	op	target address	jump

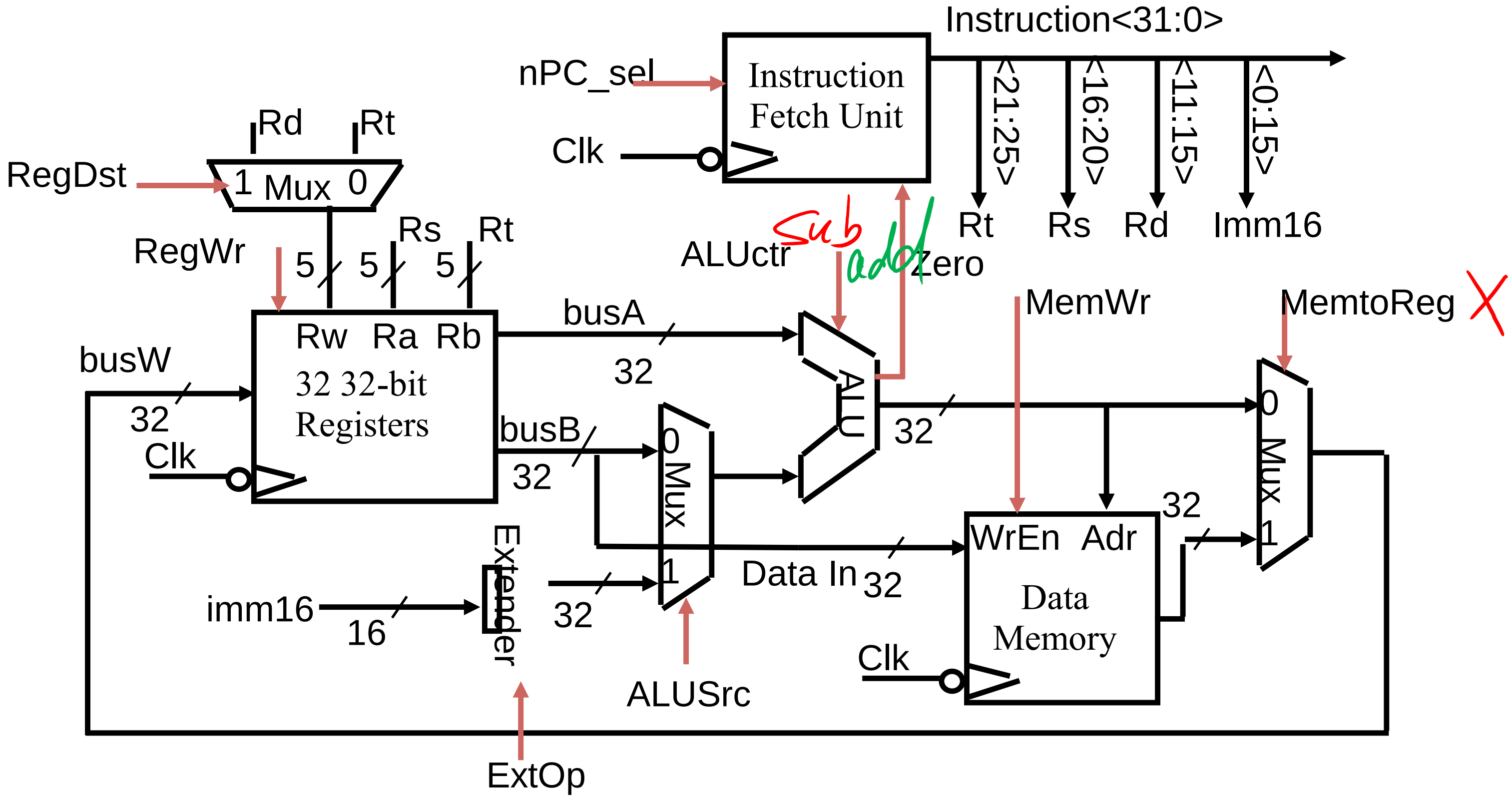
- **New PC = { PC[31..28], target address, 00 }**



Query

- **Can Zero still get asserted?**
- **Does nPC_sel need to be 0?**
  - **If not, what?**

Quiz



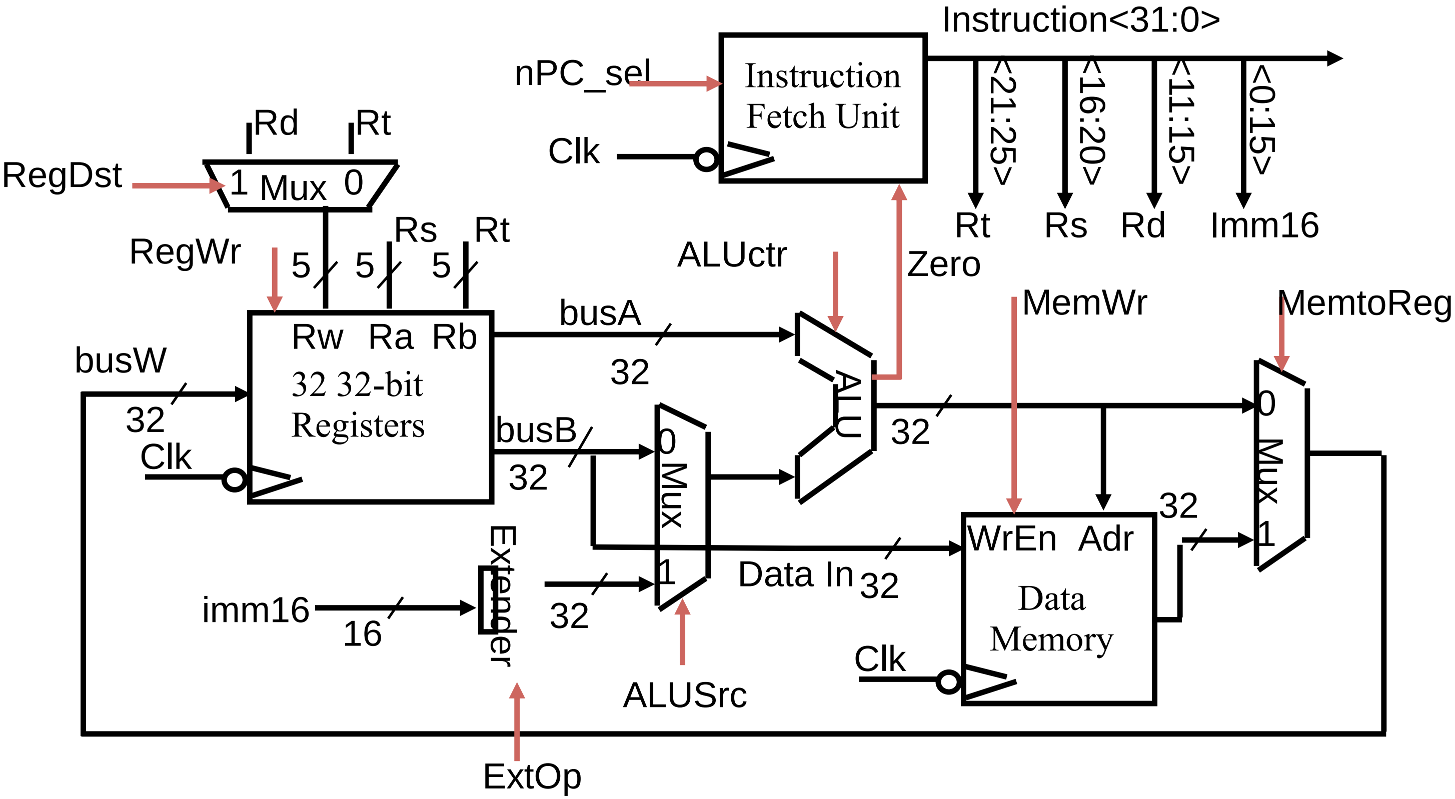
1) MemToReg='x' & ALUctr='sub'. SUB or BEQ?

2) ALUctr='add'. Which 1 signal is different for all 3 of: ADD, LW, & SW? RegDst or ExtOp?

1, 0, X   X, signed, signed

	12
a)	SR
b)	SE
c)	BR
d)	BE

Quiz



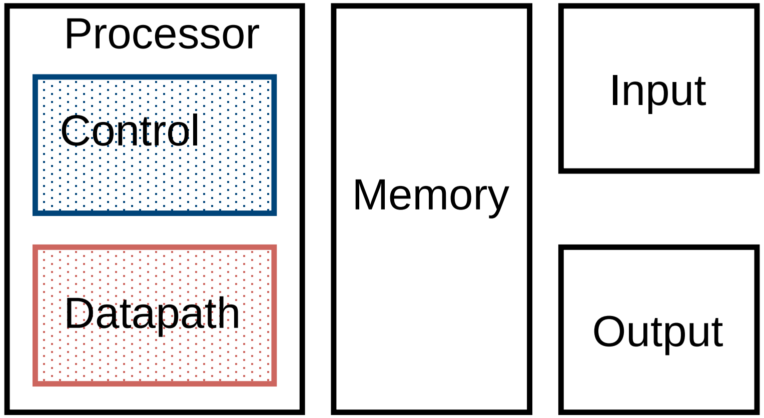
1) MemToReg='x' & ALUctr='sub'.
SUB or BEQ?

2) ALUctr='add'. Which 1 signal is different for all 3 of:
ADD, LW, & SW? RegDst or ExtOp?

	12
a)	SR
b)	SE
c)	**BR**
d)	BE

Summary: Single-cycle Processor

- 5 steps to design a processor
  - **1. Analyze instruction set → datapath requirements**
  - **2. Select set of datapath components & establish clock methodology**
  - **3. Assemble datapath meeting the requirements**
  - **4. Analyze implementation of each instruction to determine setting of control points that effects the register transfer.**
  - **5. Assemble the control logic**
    - **Formulate Logic Equations**
    - **Design Circuits**


Processor
Control
Datapath
Memory
Input
Output