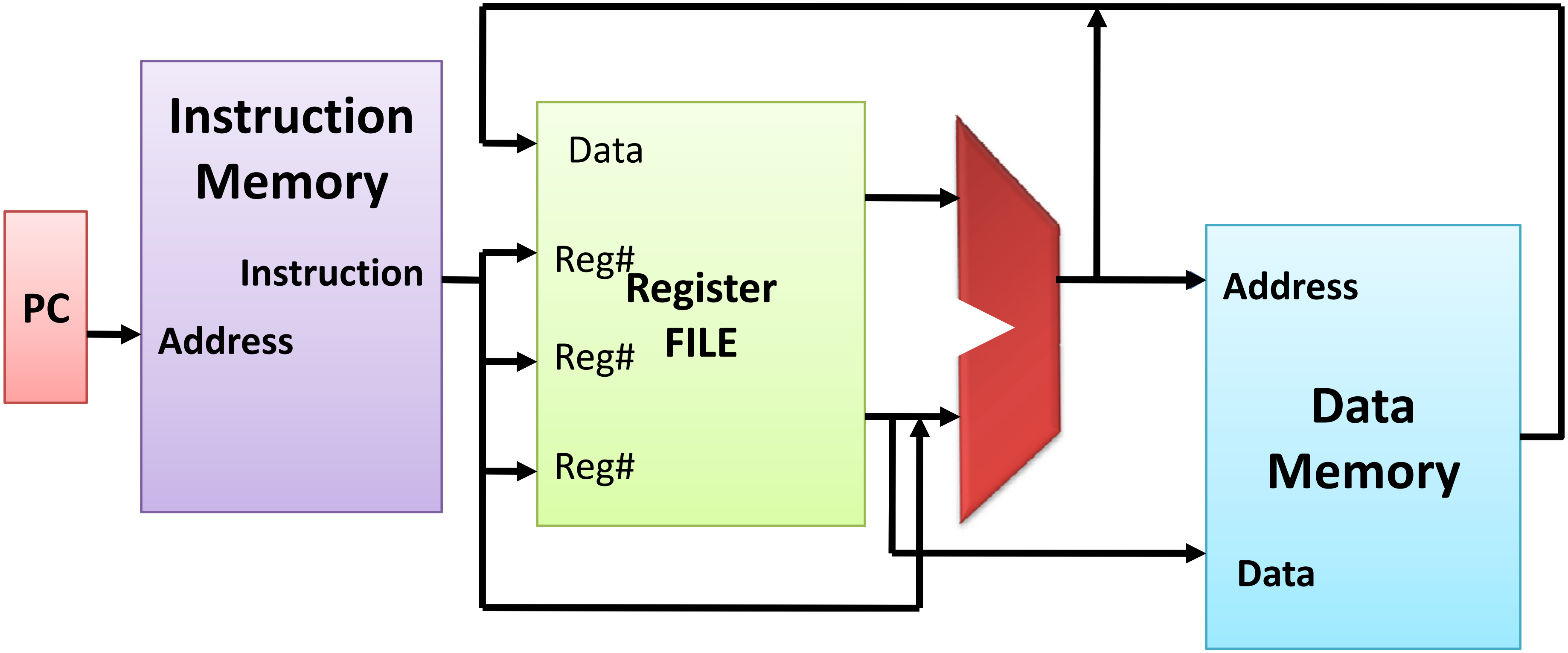
Outline

- A simple implementation: Single Cycle
  - Data path and control
- Design Strategy
- Control Path Implementation
- Performance considerations

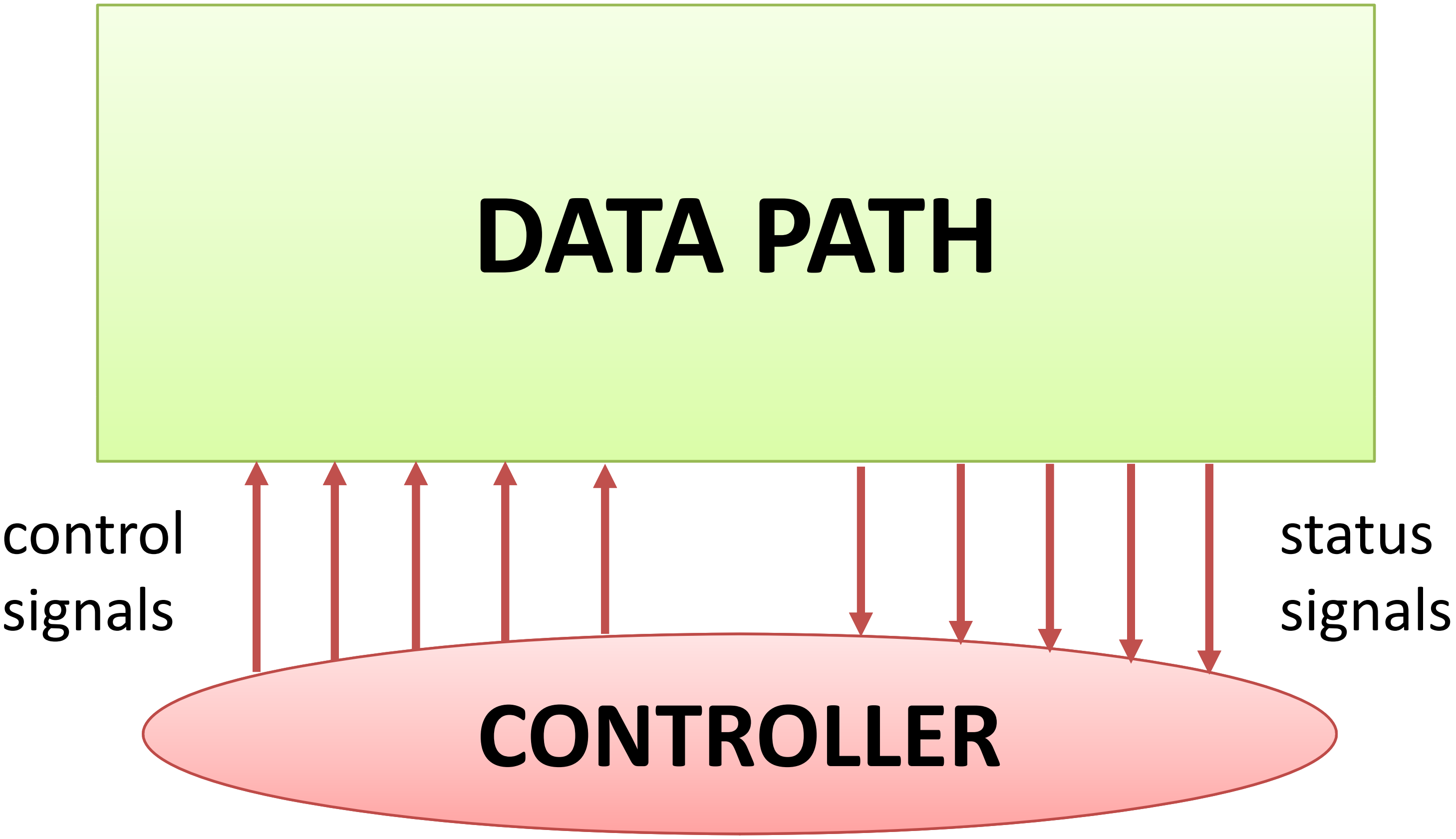
MIPS subset for implementation

- Arithmetic - logic instructions
  - **add, sub, and, or, slt**
- Memory reference instructions
  - **lw, sw**
- Control flow instructions
  - **beq, j**

Design overview



Division into data path and control



Datapath for add,sub,and,or,slt

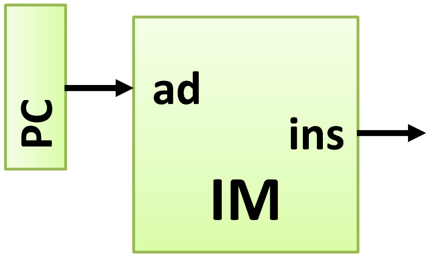
- fetch instruction
- address the register file
- pass operands to ALU
- pass result to register file
- increment PC

actions required

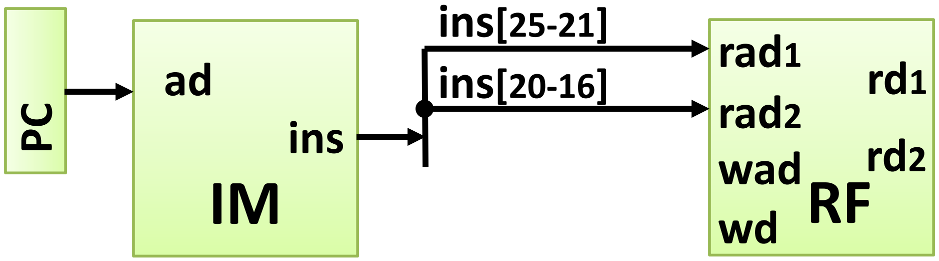
Format: add $t0, $s1, $s2

000000	10001	10010	01000	00000	100000
op	rs	rt	rd	shamt	funct

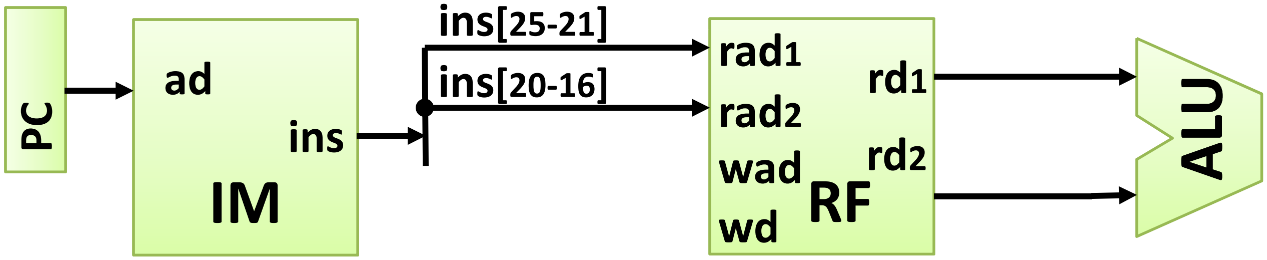
Fetching instruction



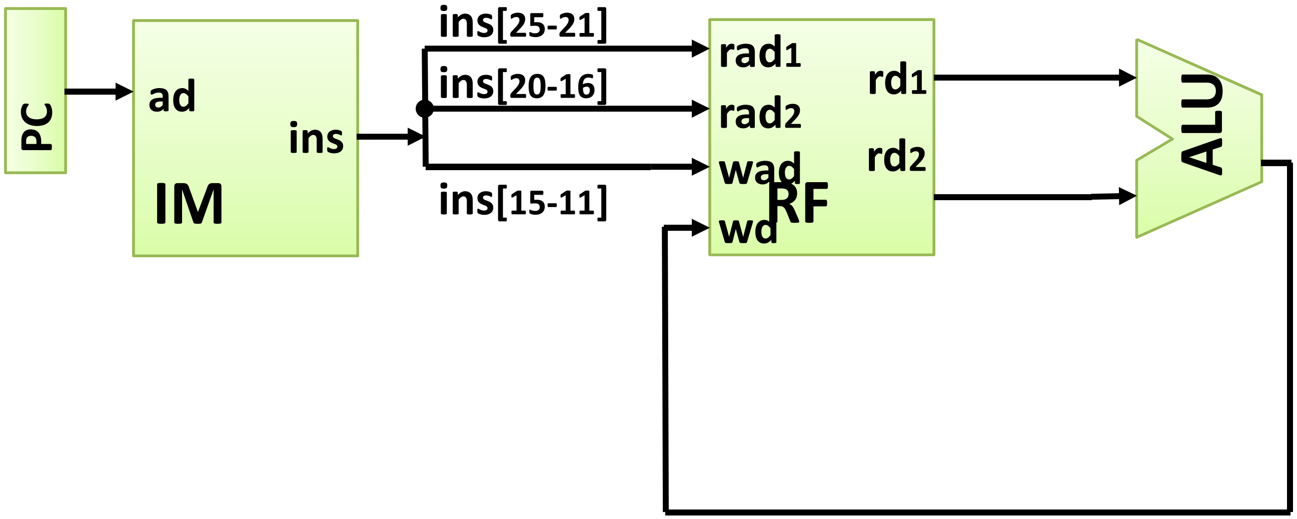
Addressing RF



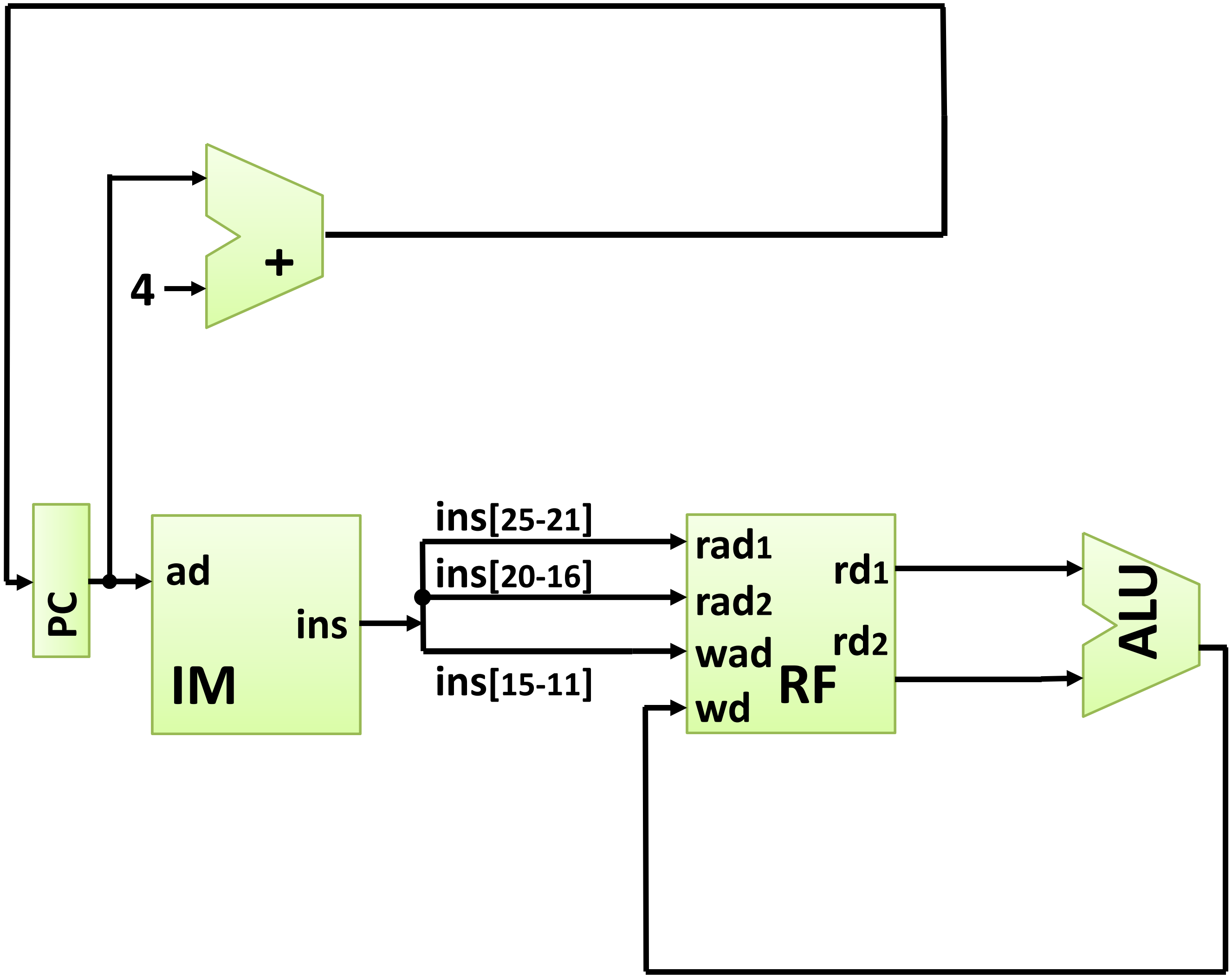
Passing operands to ALU



Passing the result to RF



Incrementing PC



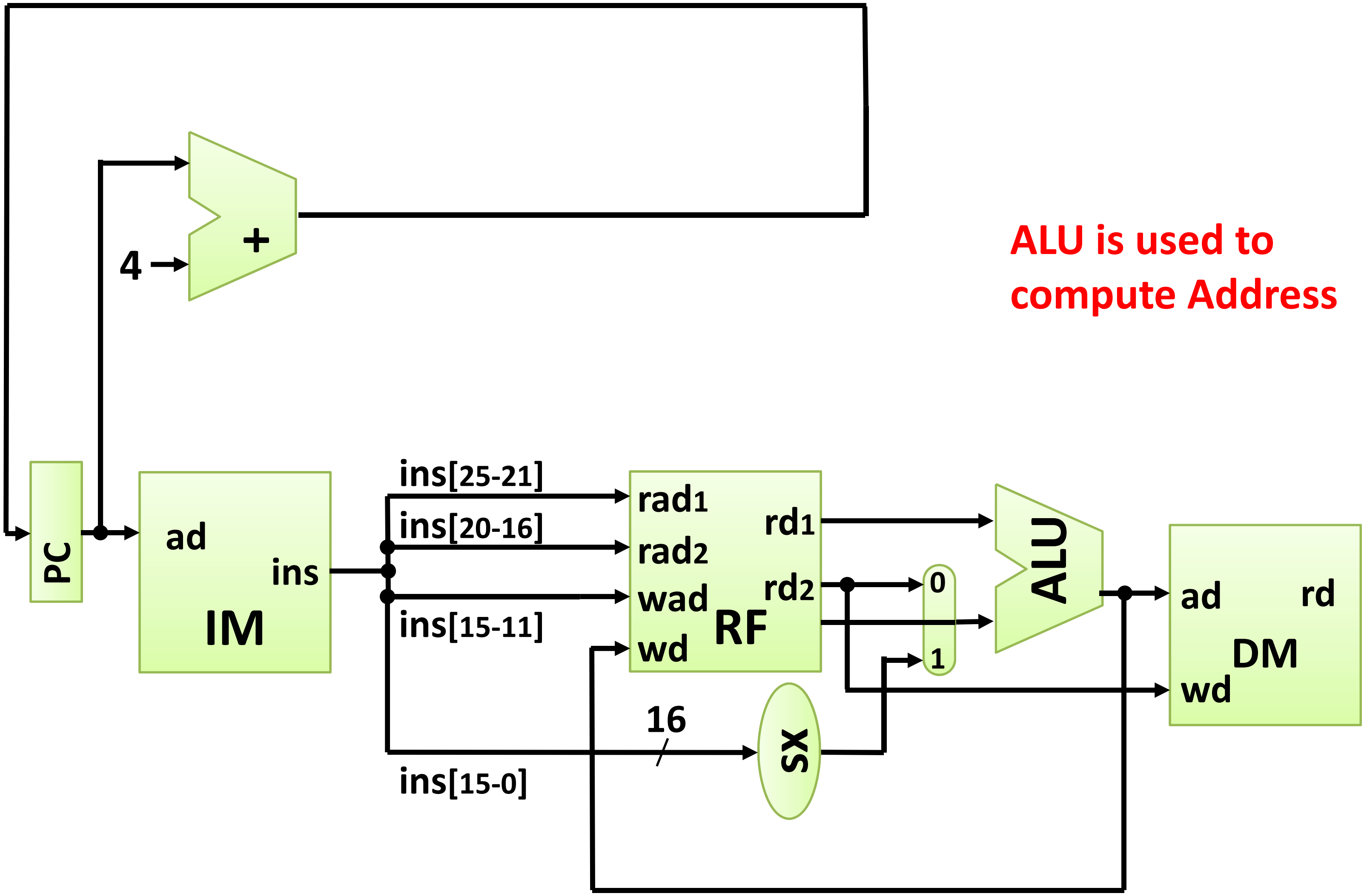
Load and Store instructions

- format : I

- Example: lw $t0, 32($s2)

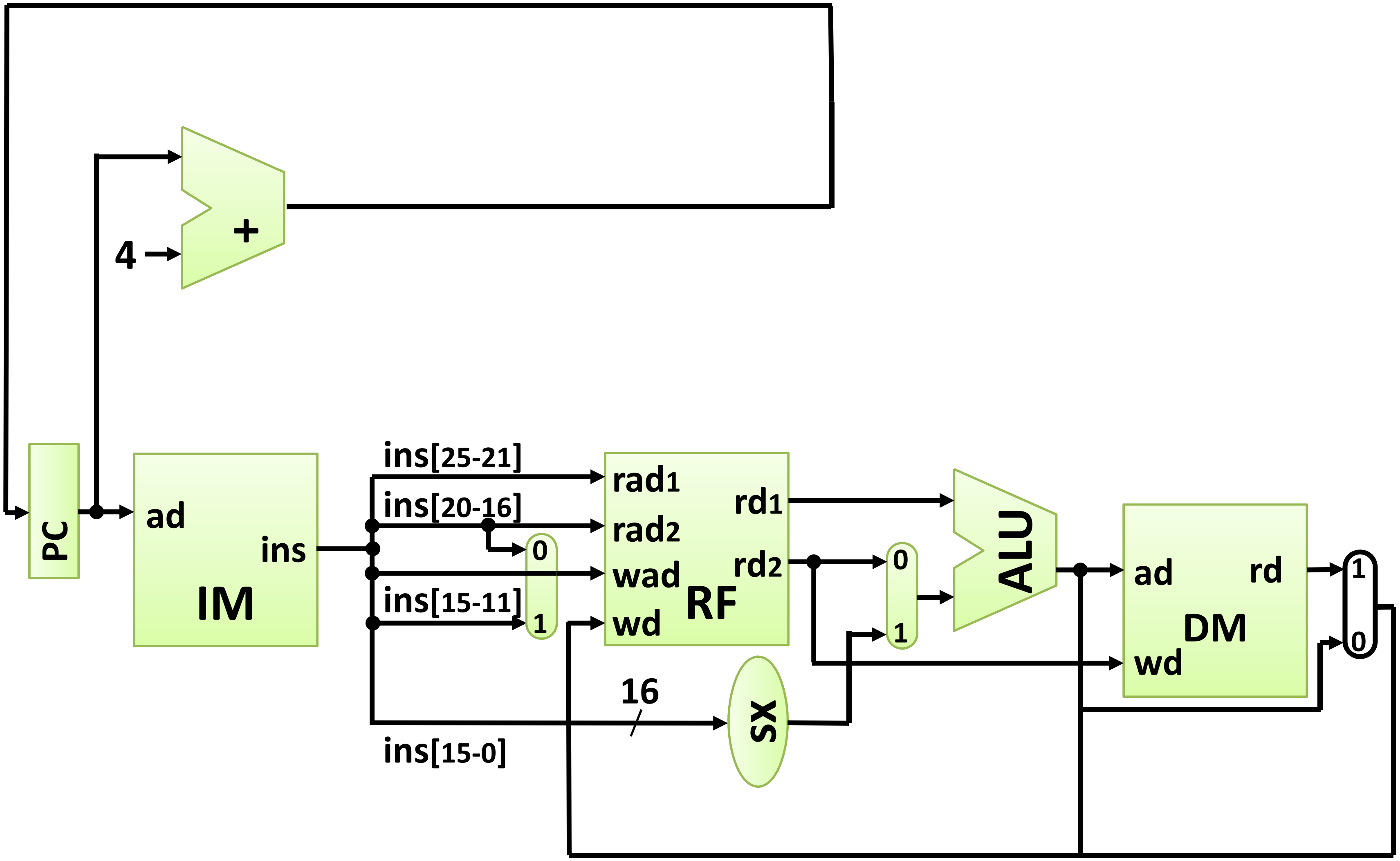
35	18	9	32
op	rs	rt	16 bit number

Adding "sw" instruction



ALU is used to compute Address

Adding “lw” instruction

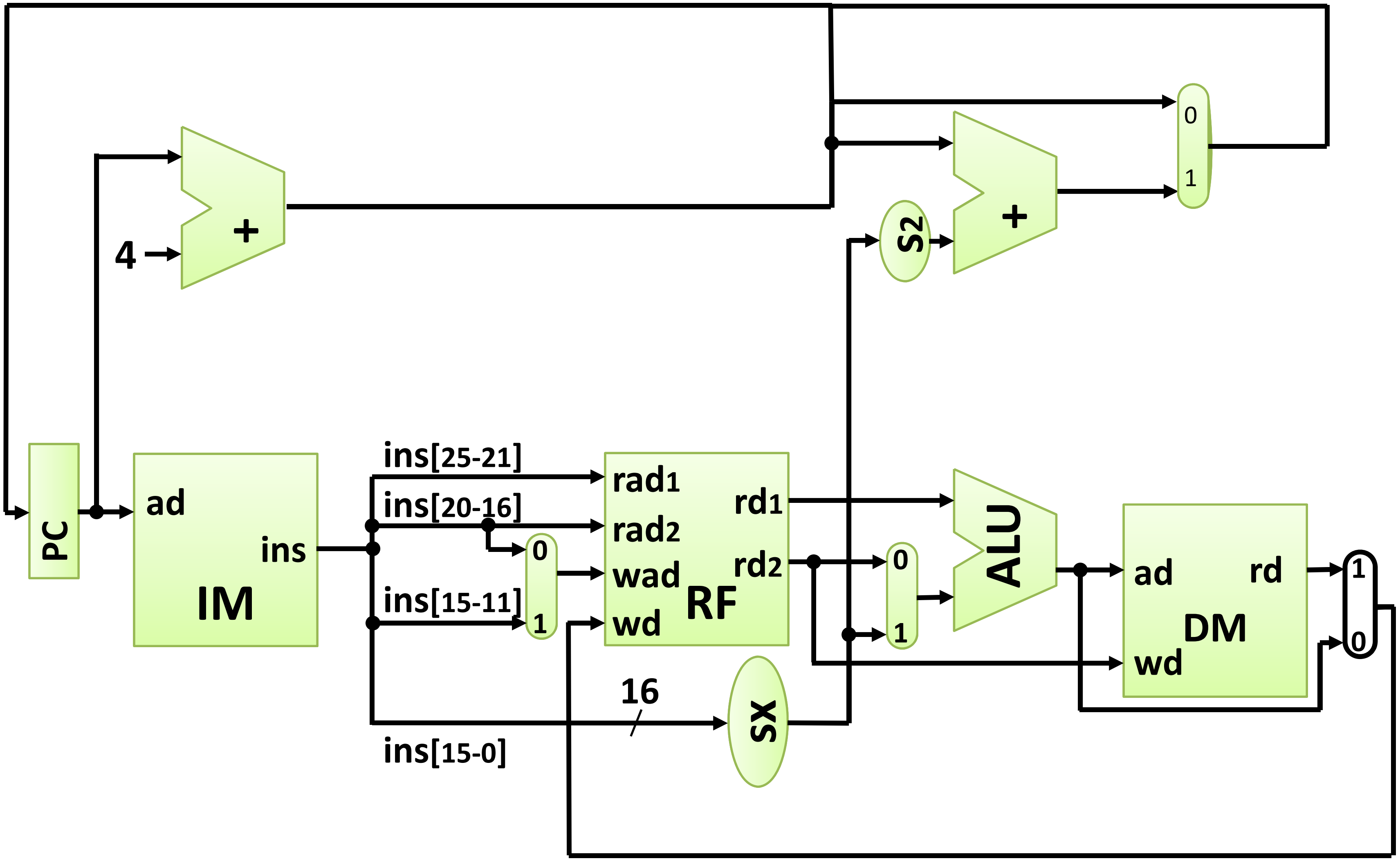


Format of beq instruction

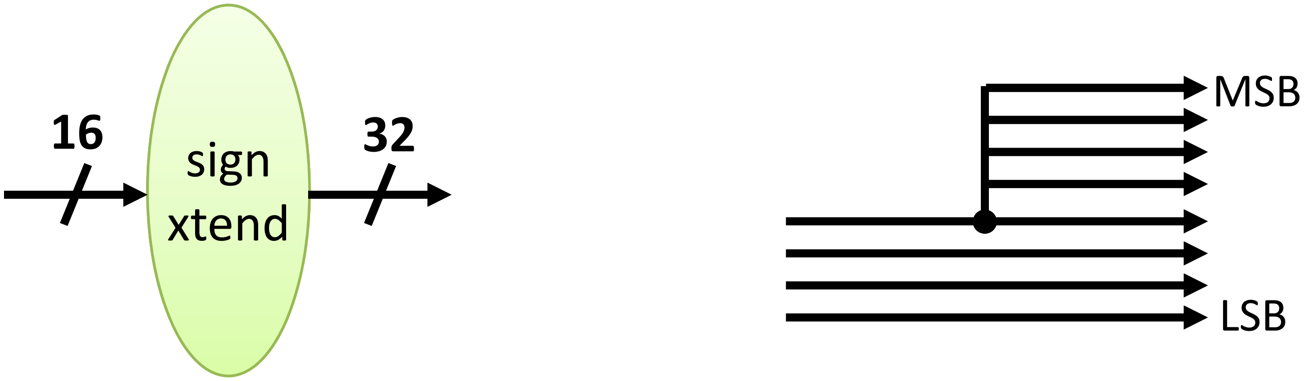
- beq I - format

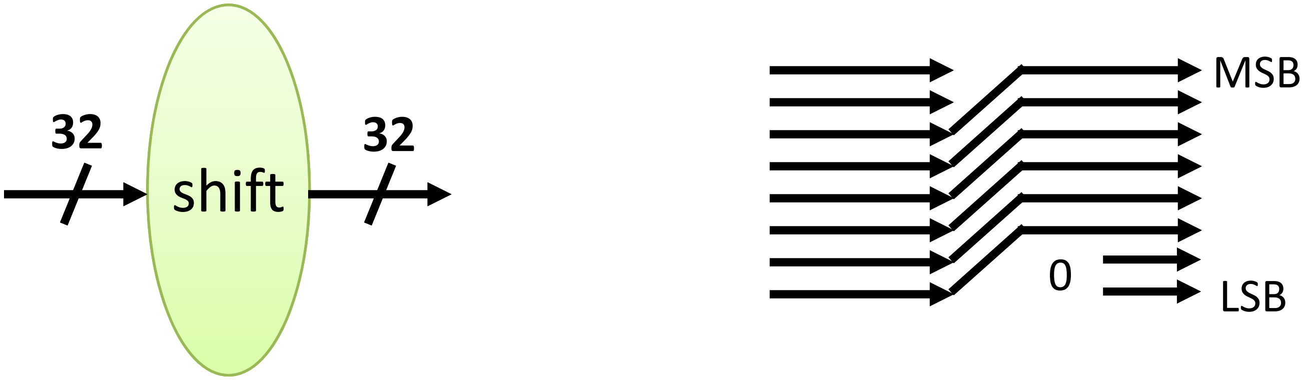
op	rs	rt	16 bit number

Adding "beq" instruction



MIPS components - bit manipulation circuits



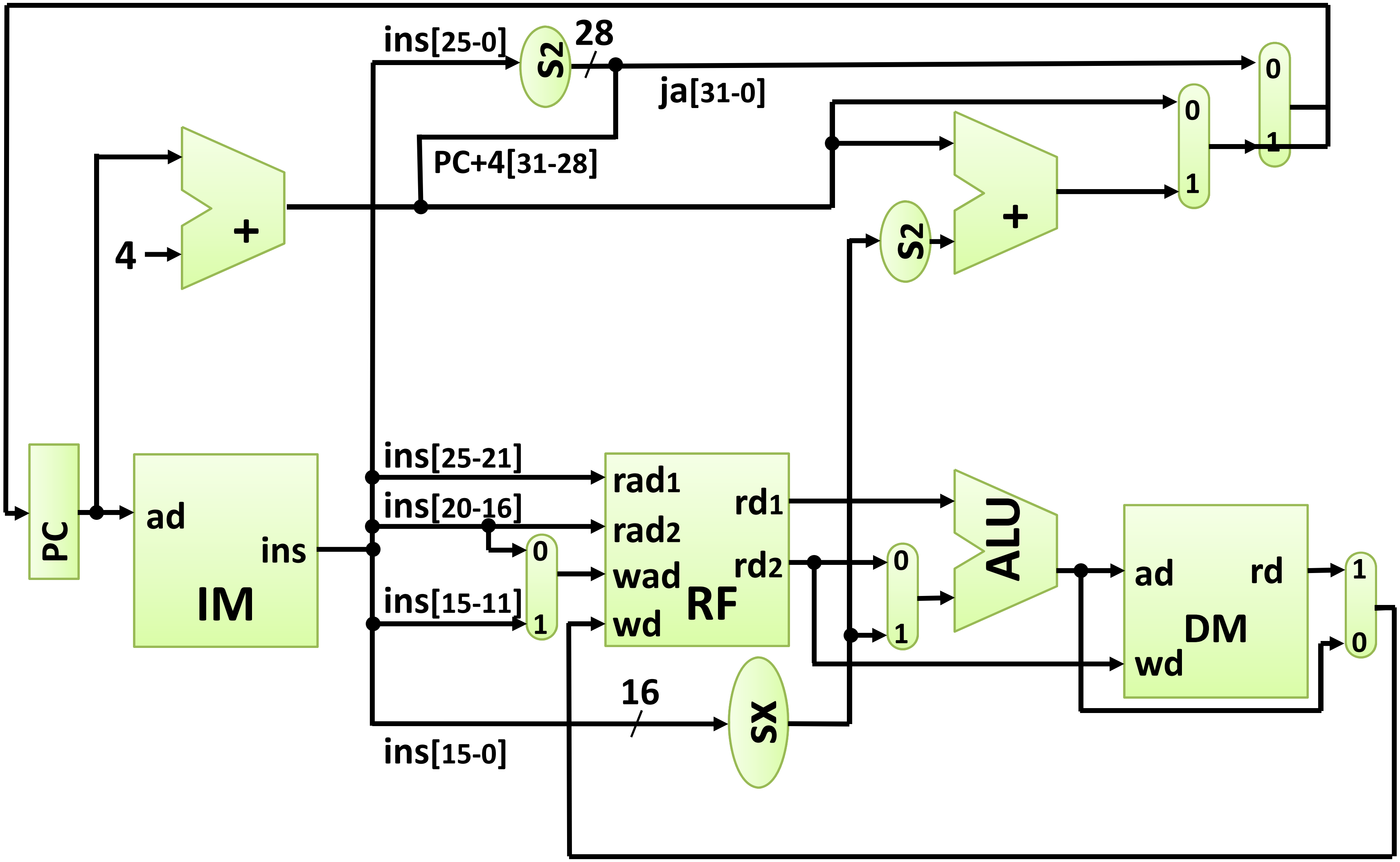


Format of jump instruction

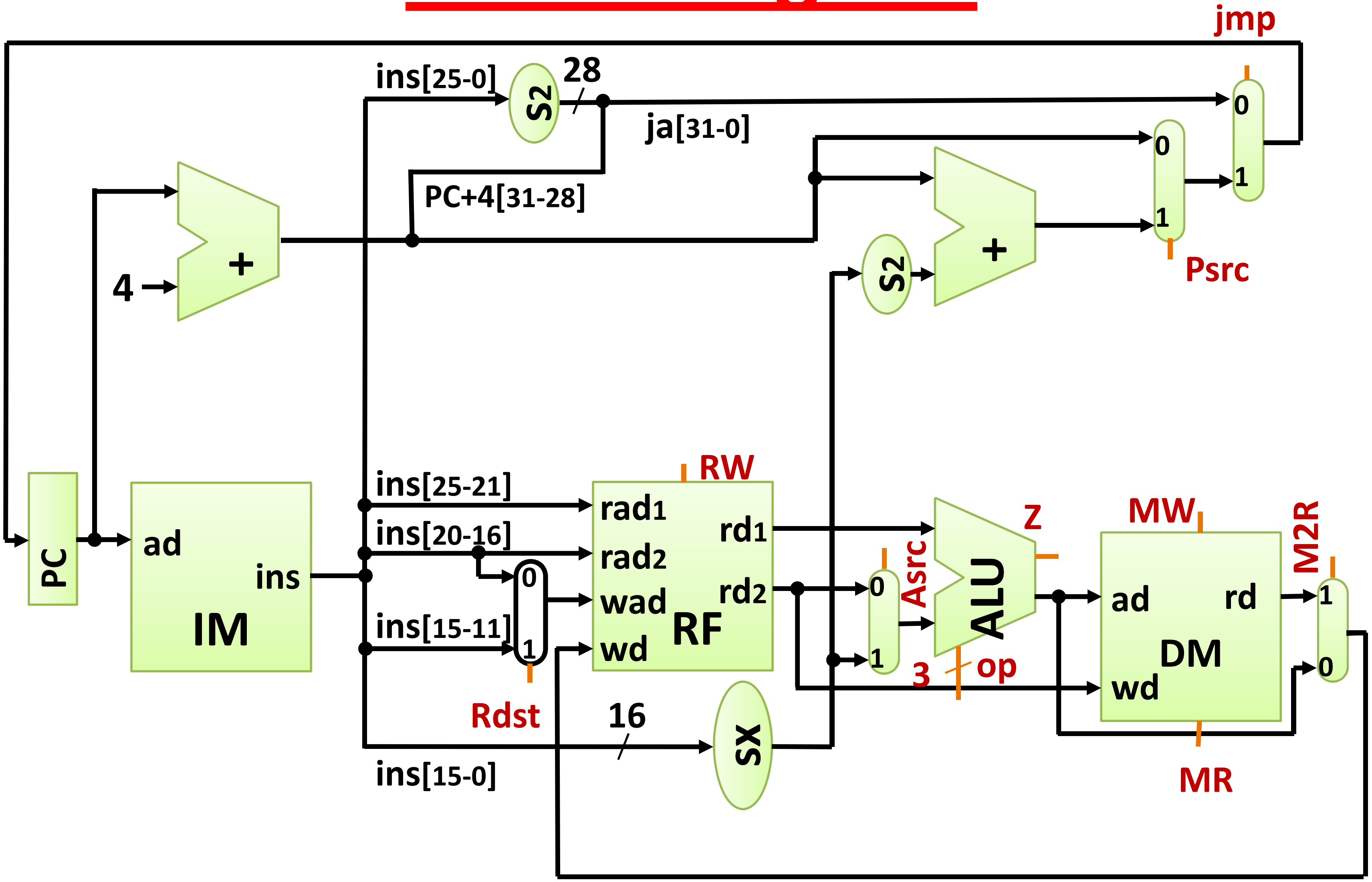
- j J - format

op	26 bit number

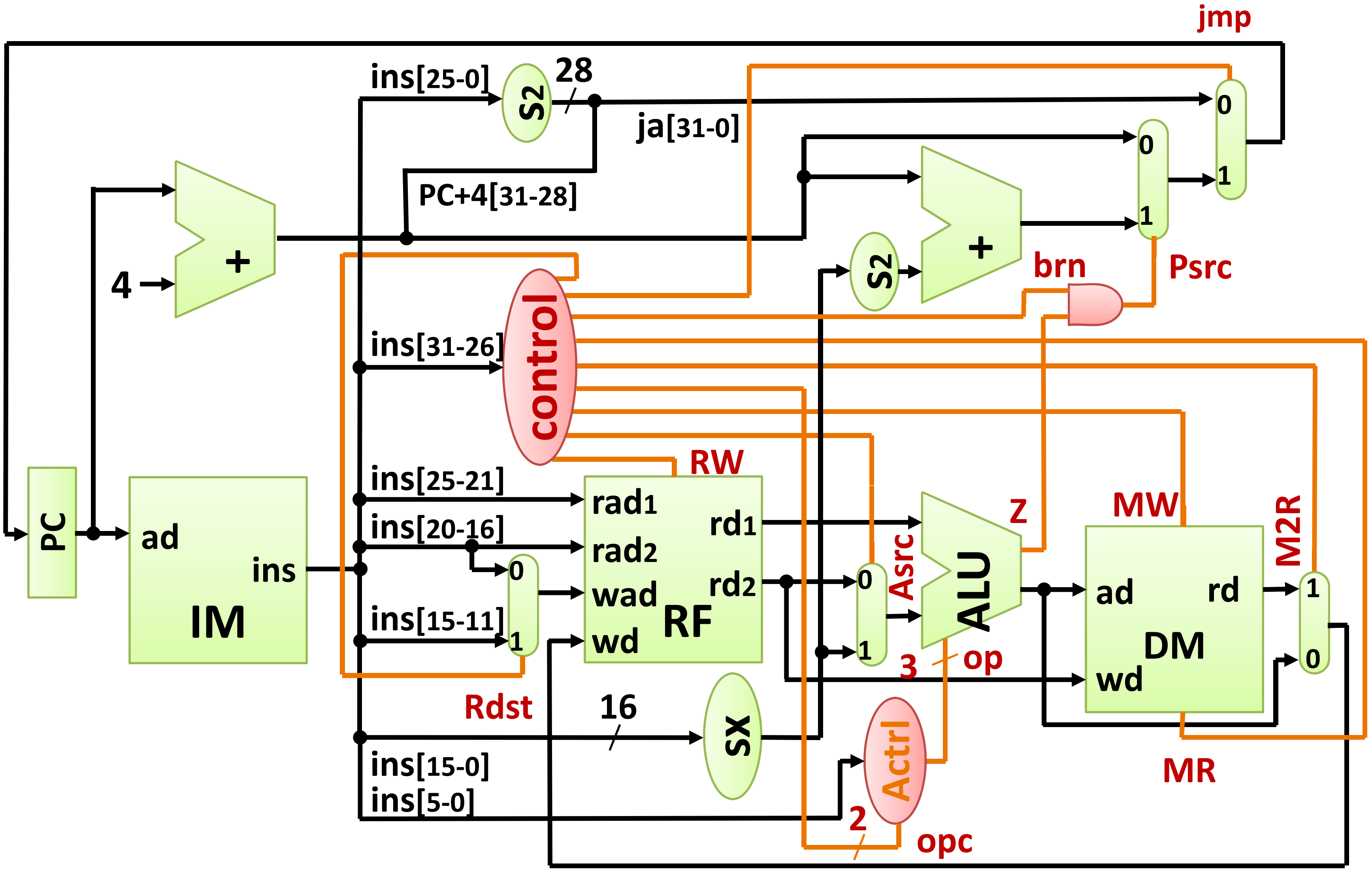
Adding “j” instruction



Control signals



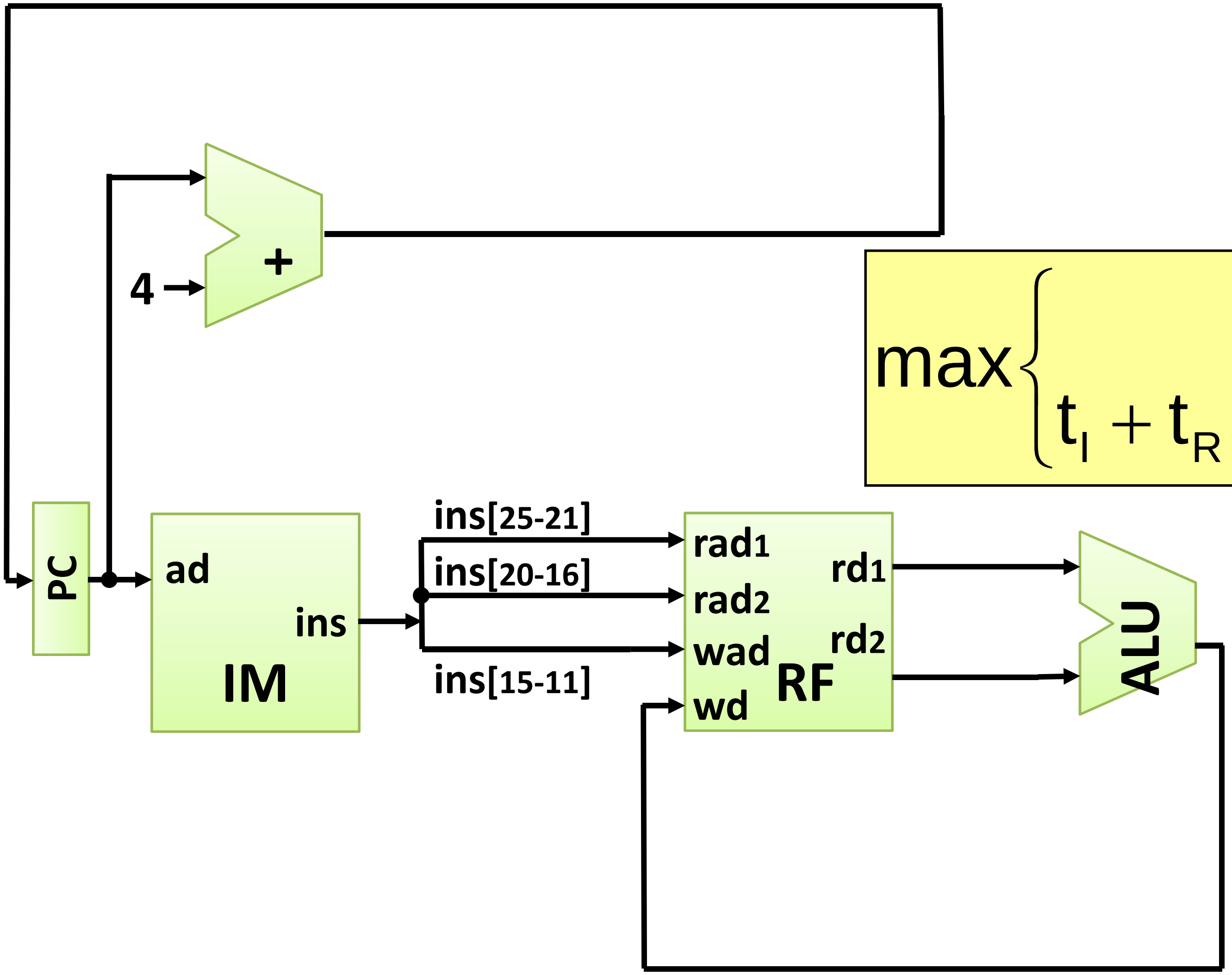
Datapath + Control



Analyzing performance

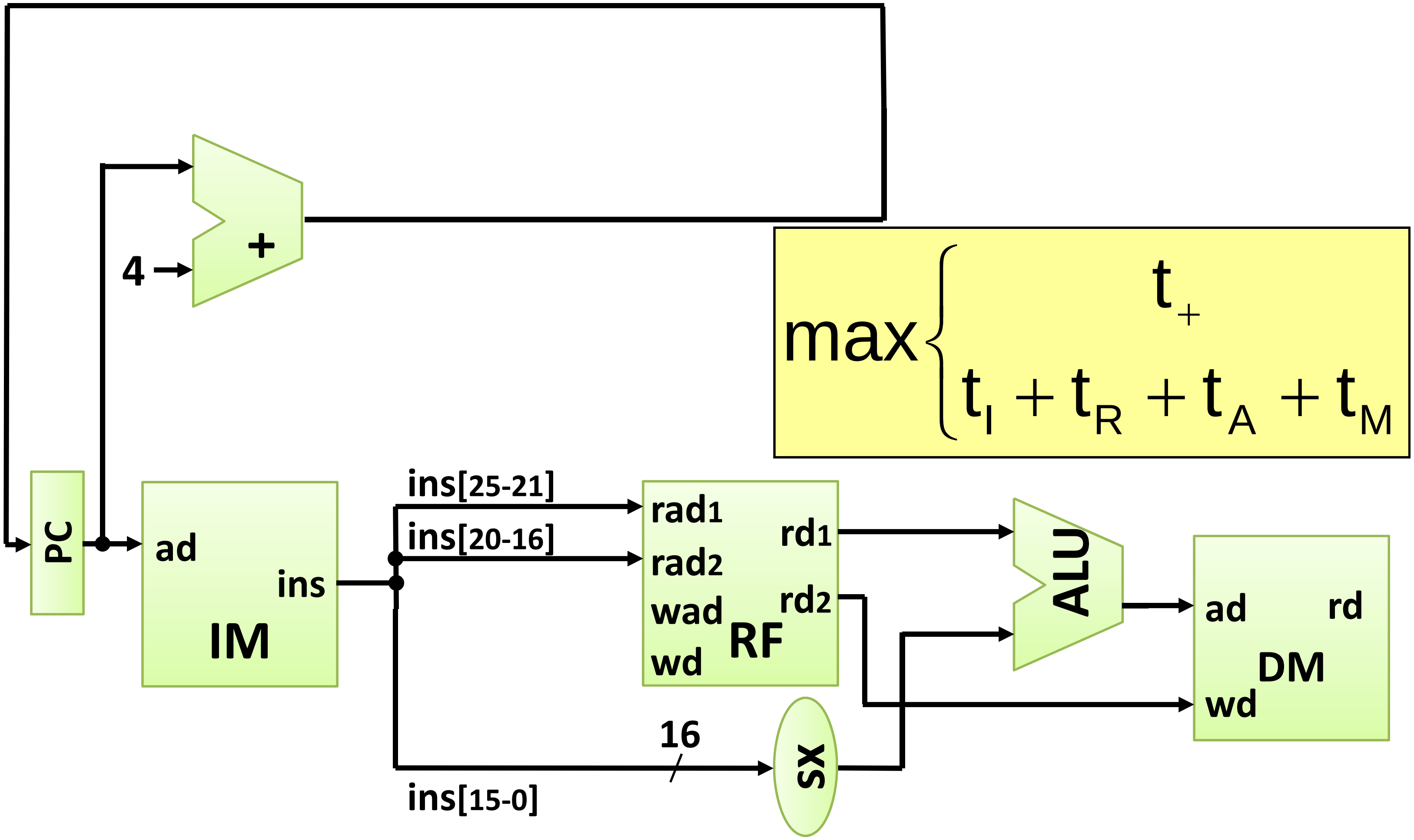
Components	Delay
Register	0
Adder	t_+
ALU	t_A
MUX	0
RF	t_R
Instruction Memory	t_I
Data Memory	t_m
Bit manipulation	0

Delay for {add, sub, and, or, slt}



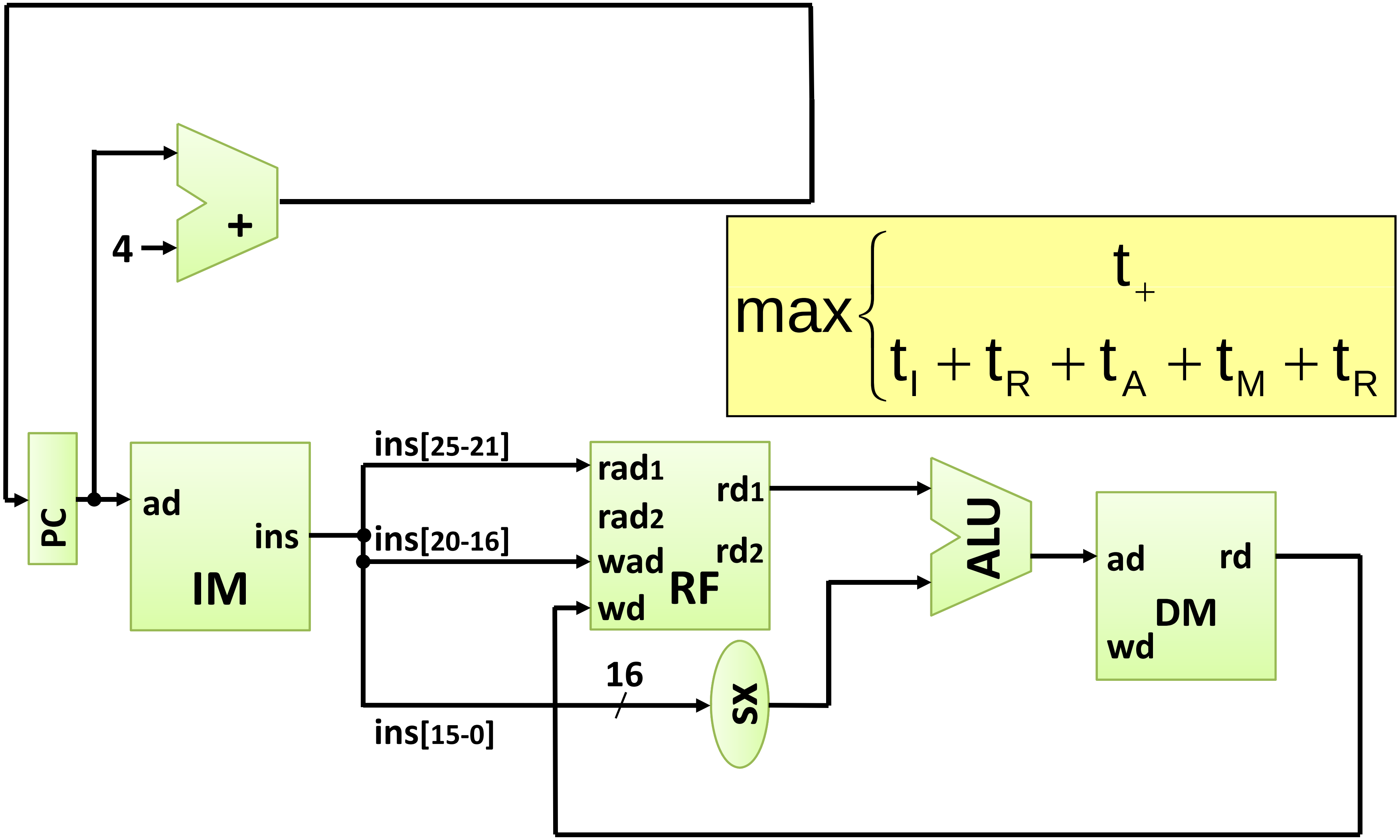
$$\max\begin{cases} t_+ \\ t_I + t_R + t_A + t_R \end{cases}$$

Delay for {sw}



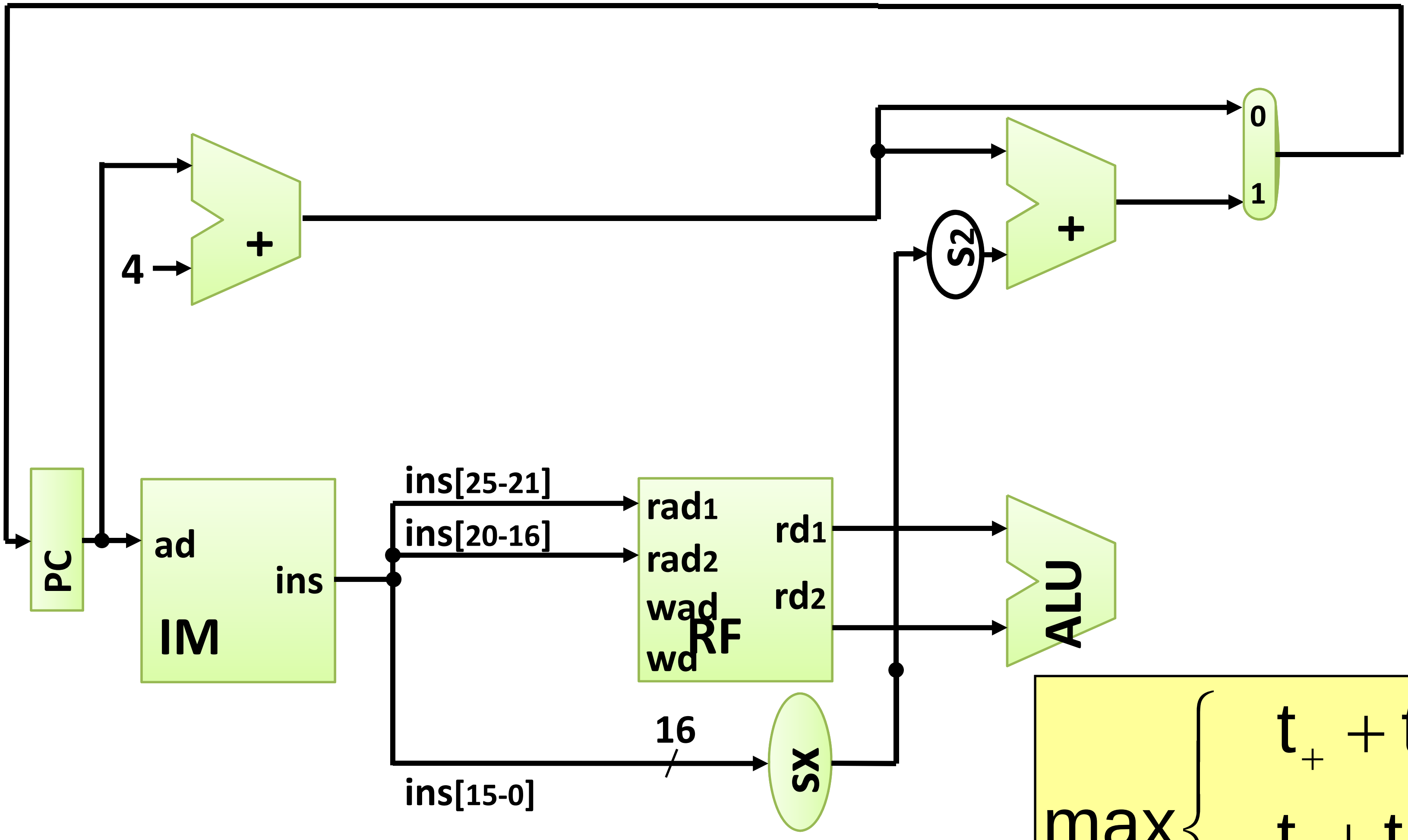
$$\max\begin{cases} t_+ \\ t_I + t_R + t_A + t_M \end{cases}$$

Delay for {lw}



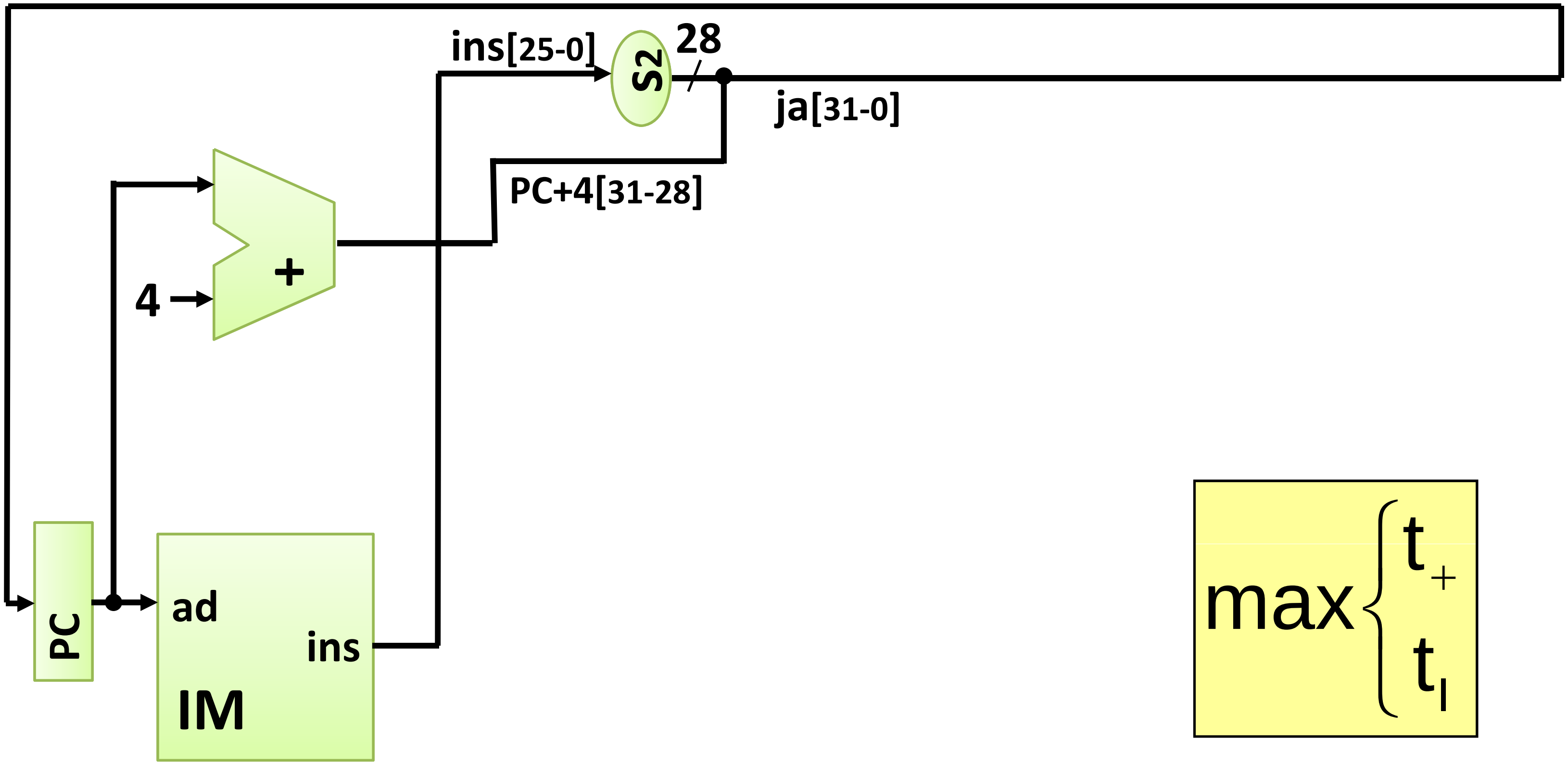
$$\max\begin{cases} t_+ \\ t_I + t_R + t_A + t_M + t_R \end{cases}$$

Delay for {beq}



$$\max\begin{cases} t_{+} + t_{+} \\ t_{I} + t_{+} \\ t_{I} + t_{R} + t_{A} \end{cases}$$

Delay for { j}



$$\max\begin{cases} t_+ \\ t_I \end{cases}$$

Overall clock period

$$\max\begin{cases} R: t_+, t_I + t_R + t_A + t_R \\ SW\ : t_+, t_I + t_R + t_A + t_M \\ LW\ : t_+, t_I + t_R + t_A + t_M + t_R \\ Beq\ : t_+ + t_+, t_I + t_+, t_I + t_R + t_A \\ J\ : t_+, t_I \end{cases}$$

or

$$\max\begin{cases} \mathrm{t}_\mathrm{I} + \mathrm{t}_\mathrm{R} + \mathrm{t}_\mathrm{A} + \mathrm{t}_\mathrm{M} + \mathrm{t}_\mathrm{R} \\ \mathrm{t}_+ + \mathrm{t}_+ \\ \mathrm{t}_\mathrm{I} + \mathrm{t}_+ \end{cases}$$

Analyzing performance

Components	Delay	Example
Register	0	**0ns**
Adder	t_+	**4ns**
ALU	t_A	**5ns**
MUX	0	**0ns**
RF	t_R	**3ns**
Instruction Memory	t_I	**6ns**
Data Memory	t_m	**6ns**
Bit manipulation	0	**0ns**

INS	**Delay**
R	17ns
SW	20ns
LW	23ns
Beq	14ns
J	6ns

Thanks