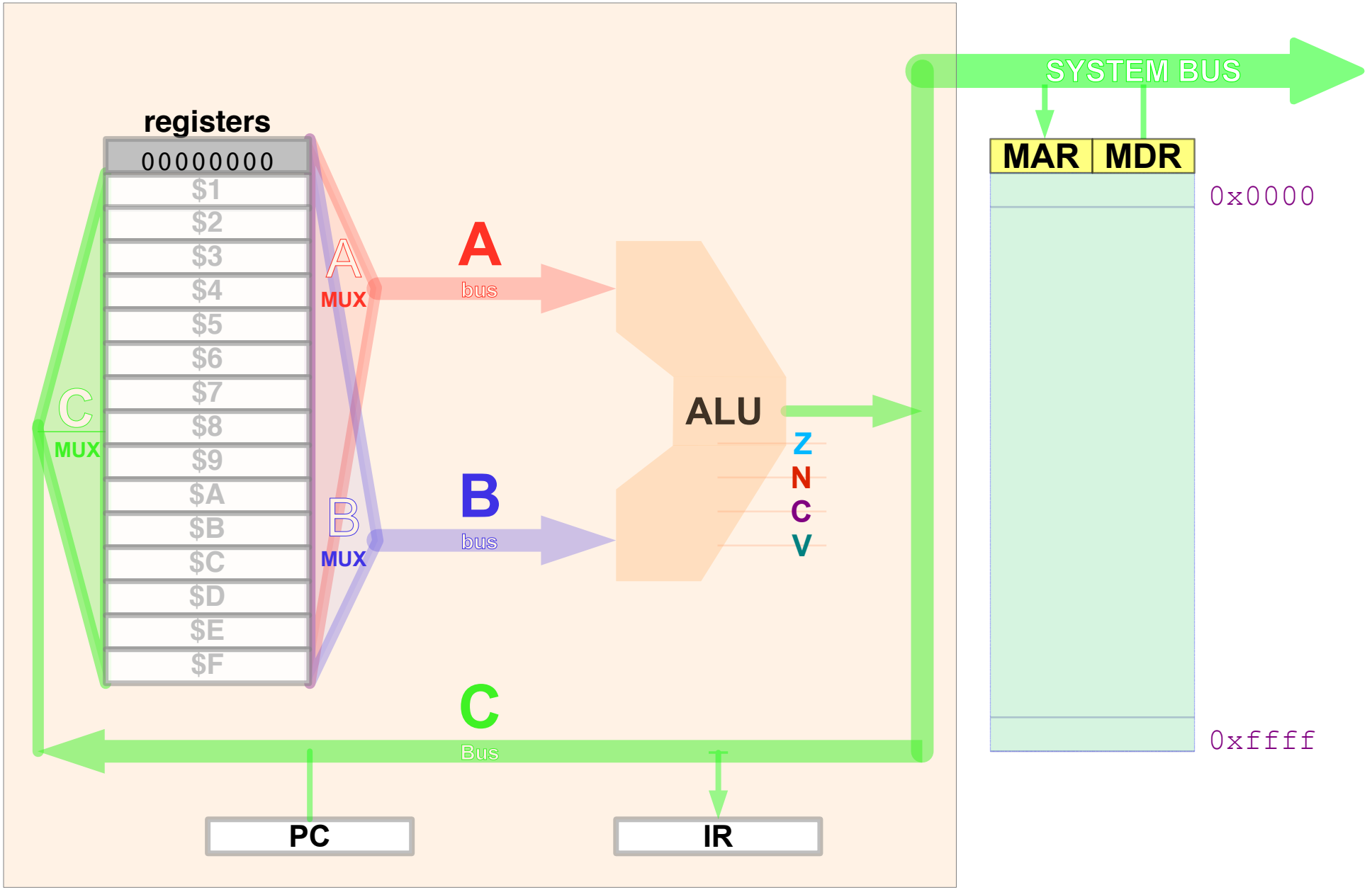
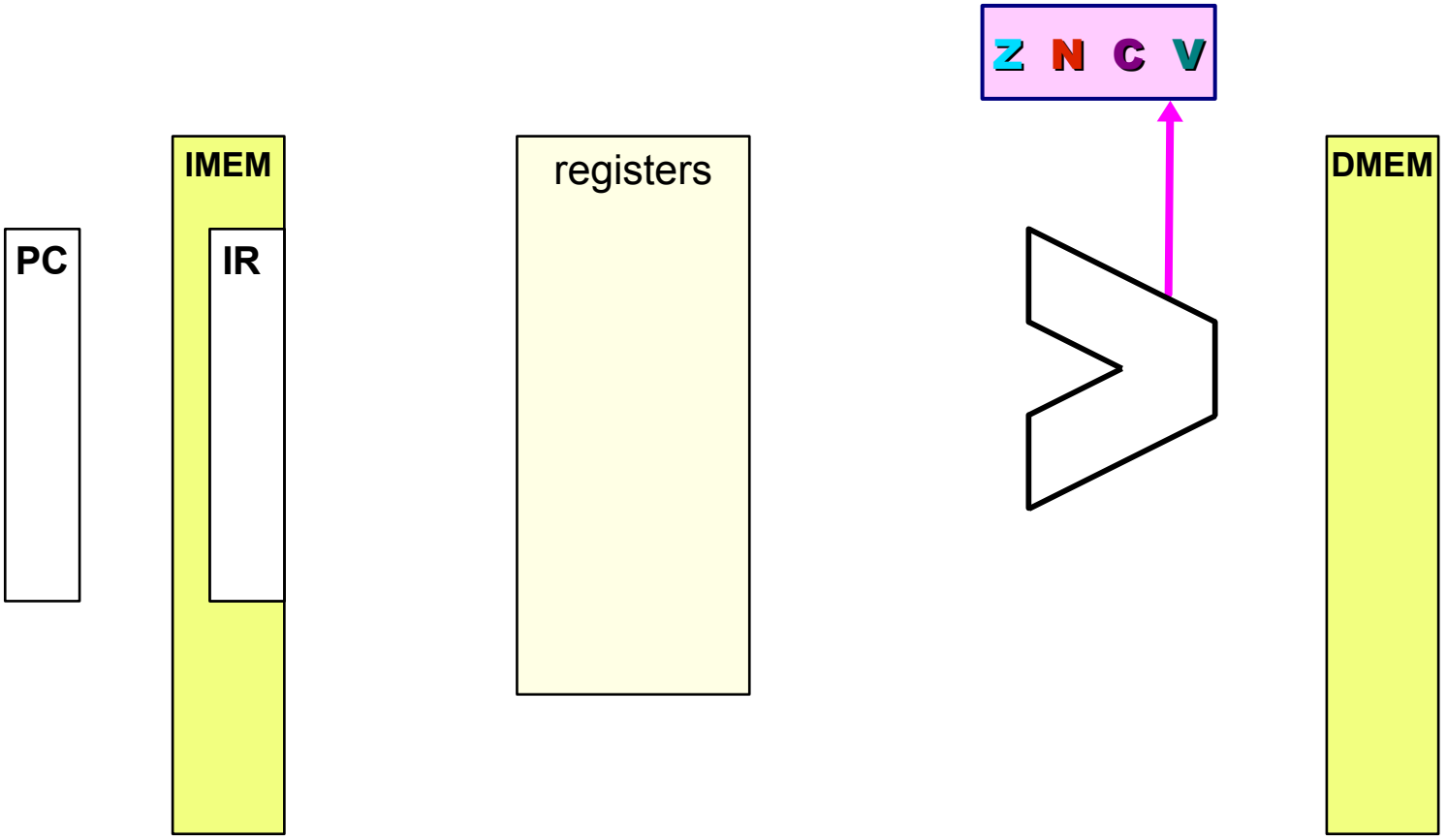
CMP 334 (4/10/19)

- **TOY** processor redesign
  - Single cycle instruction execution (review)
    - Datapath (separate instruction & data memory)
    - Control signals
  - Pipeline instruction execution implementations
- Processor performance
  - Performance equations (review)
  - HW 13, HW 14
- Pipeline hazards
  - Structural hazards
  - Data hazards
  - Control hazards

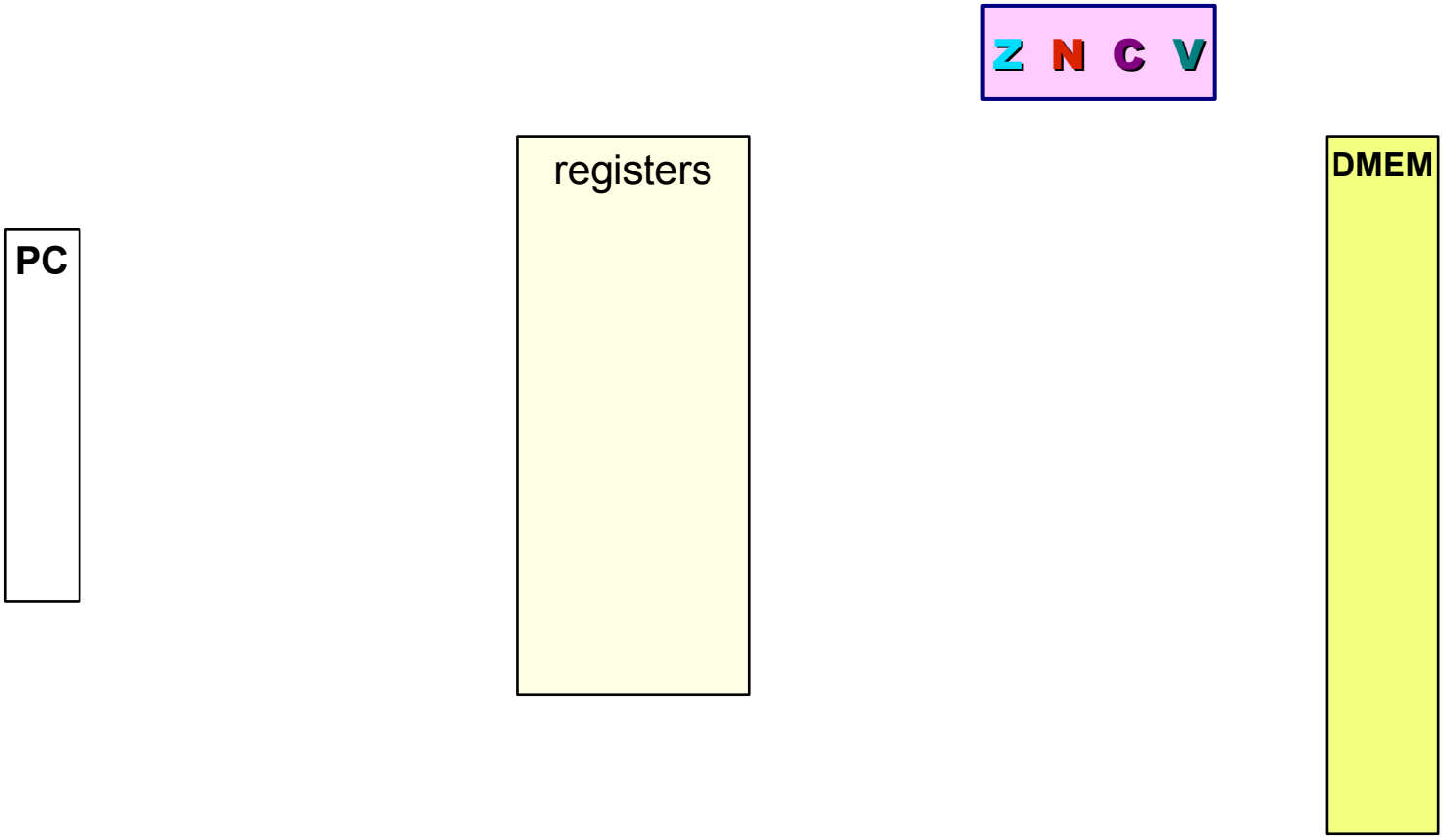
TOY ISA Simple Implementation



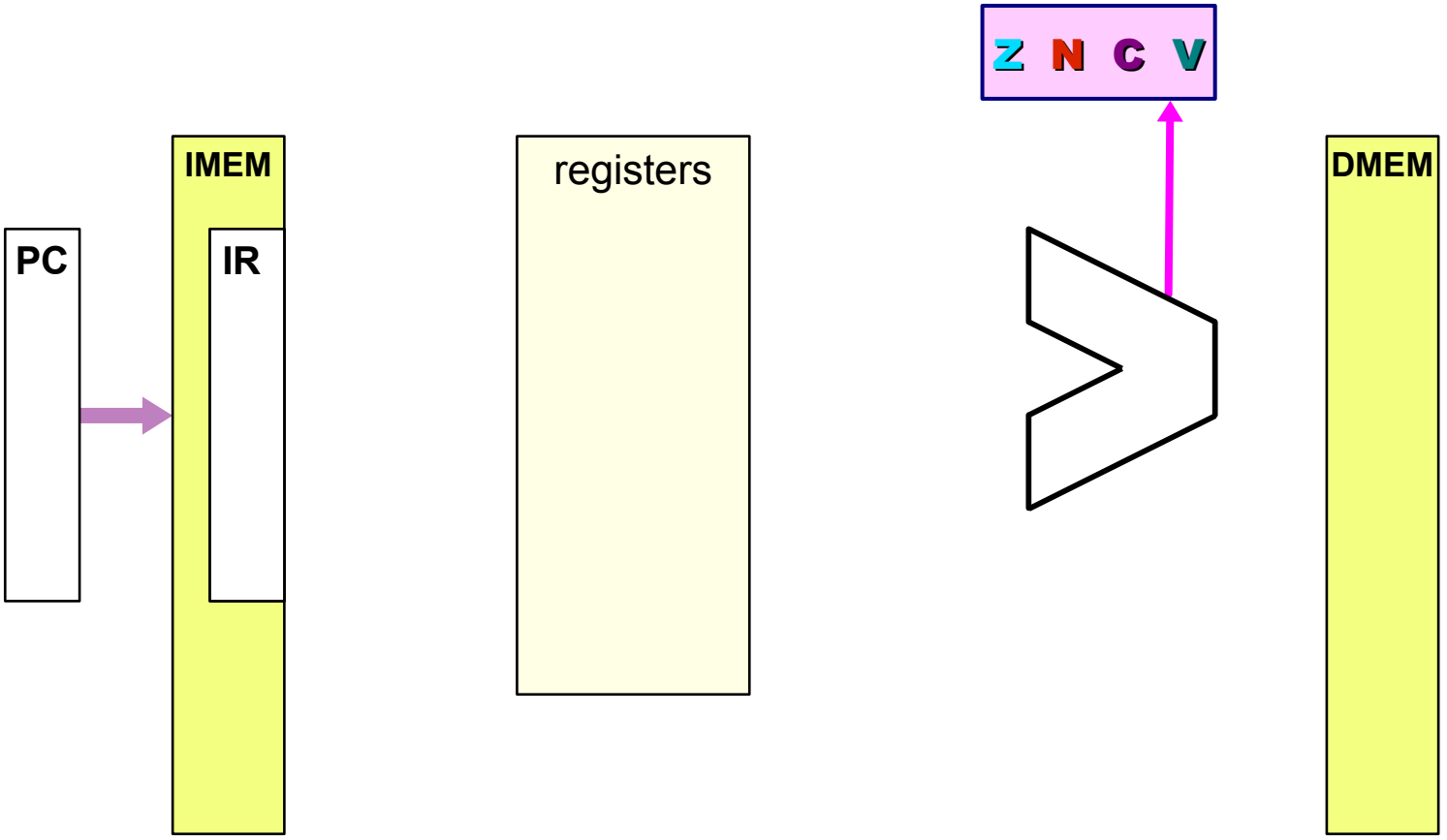
TOY Processor Redesign



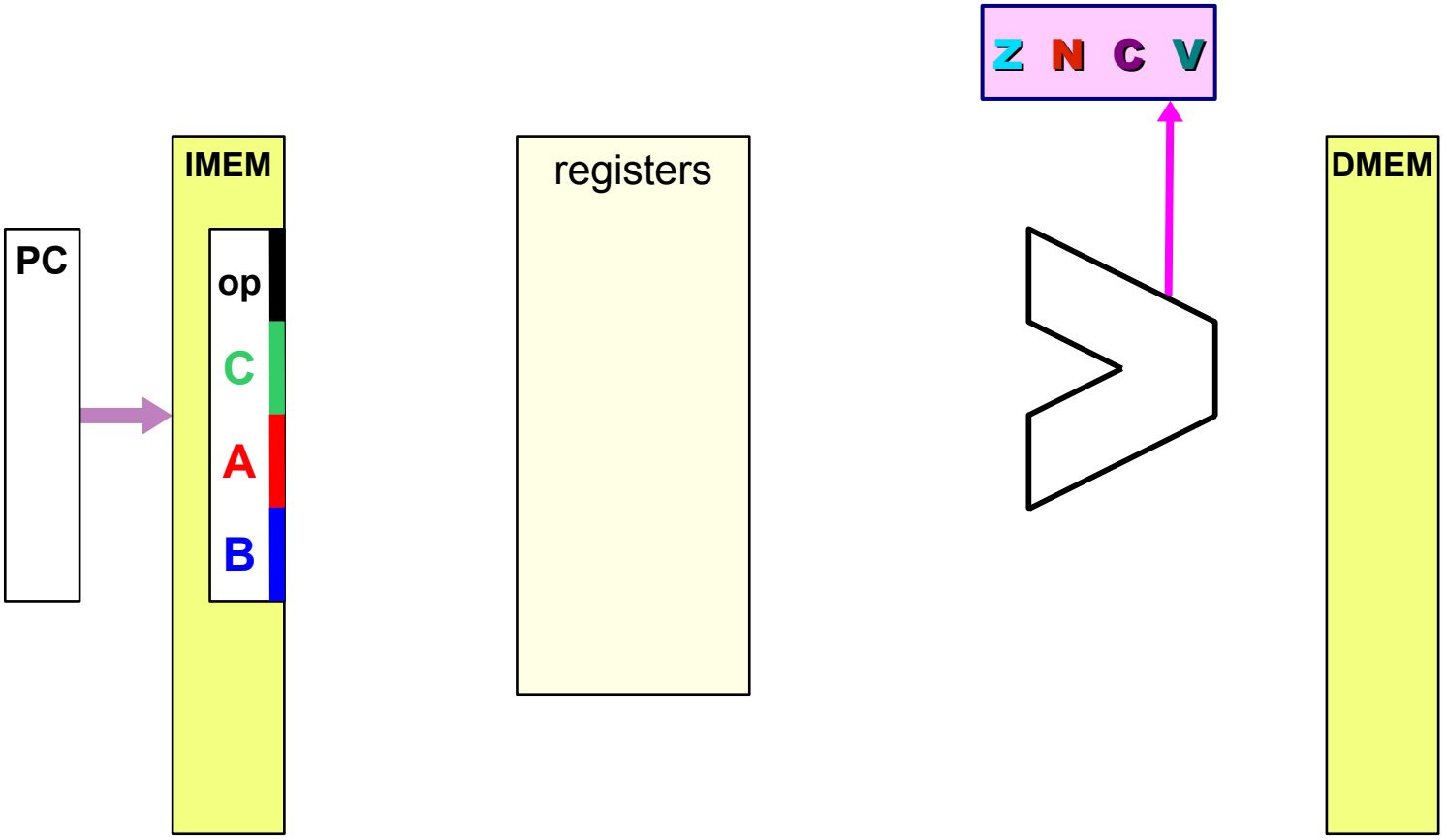
Redesign Memory Elements



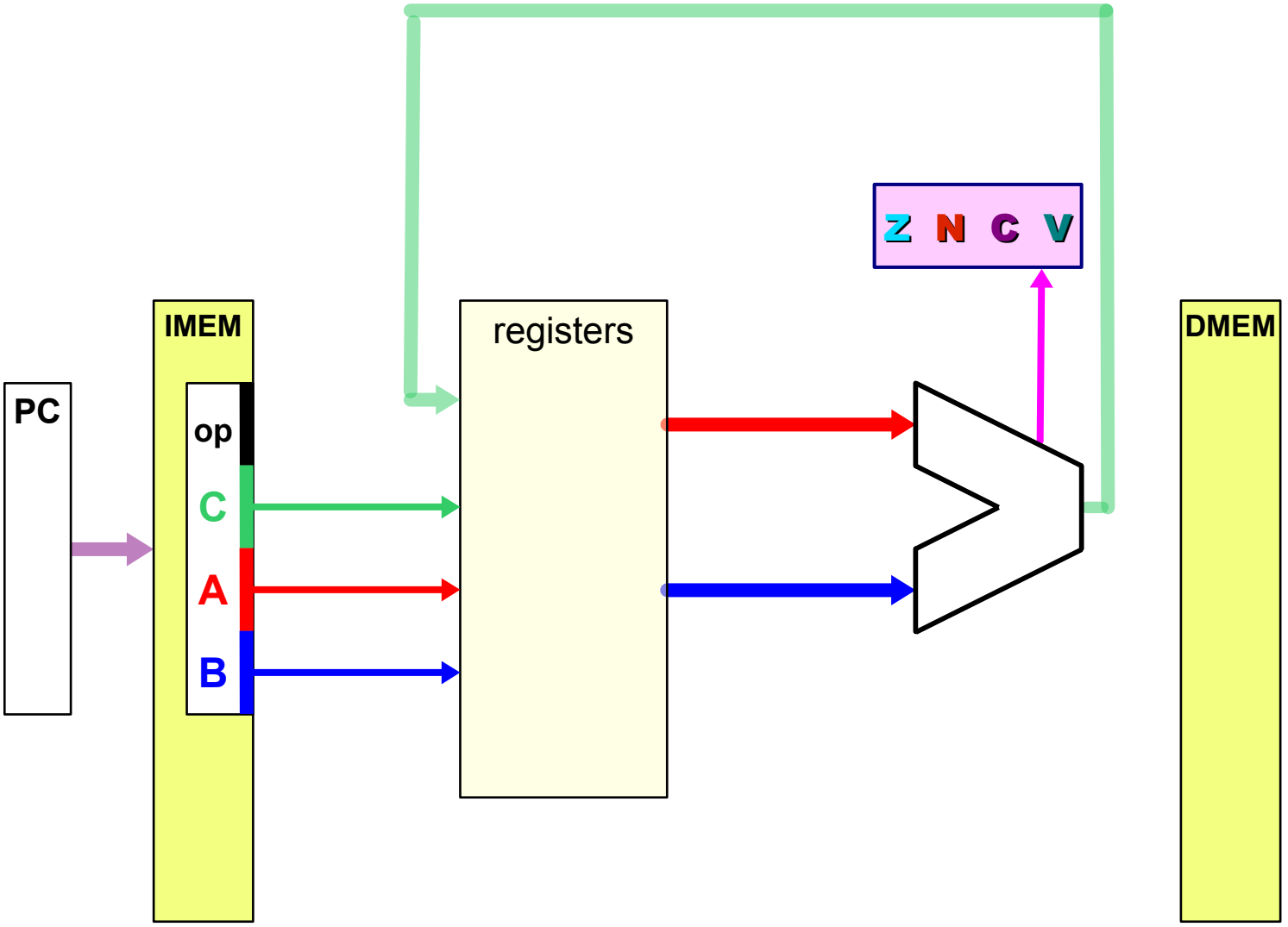
Support for Instruction `fetch`



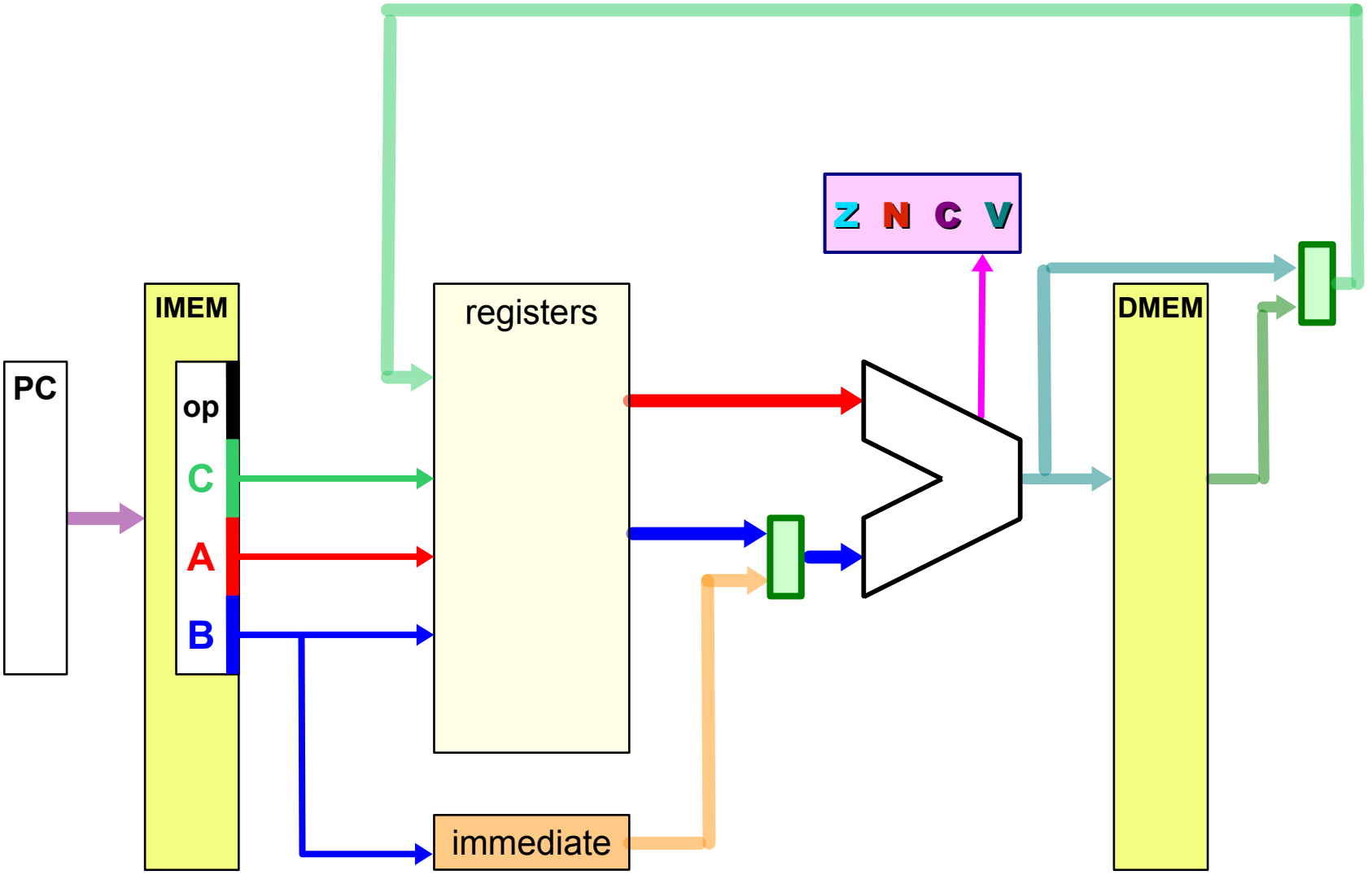
TOY Redesign **IR**



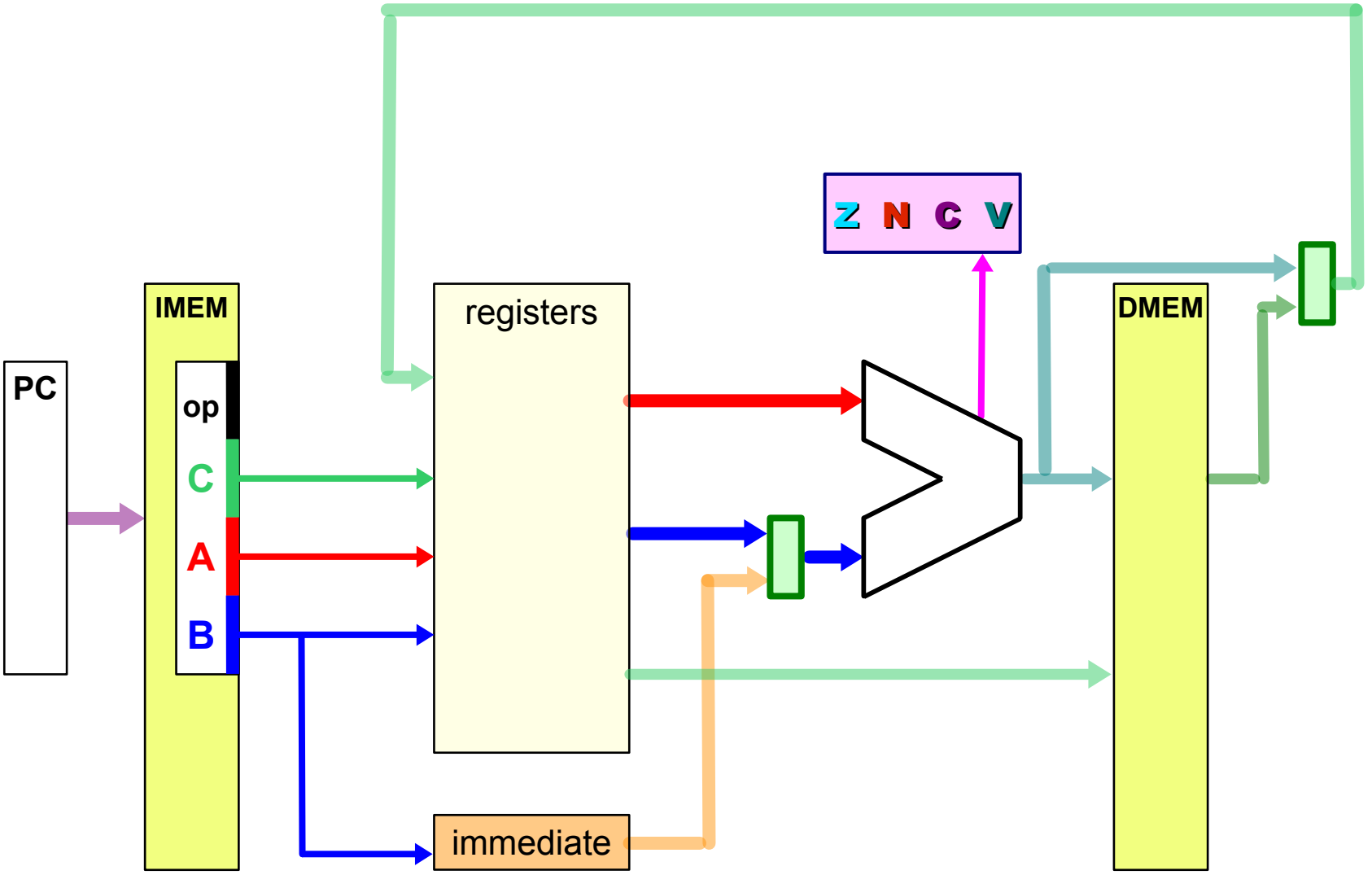
Support for ALU Instructions



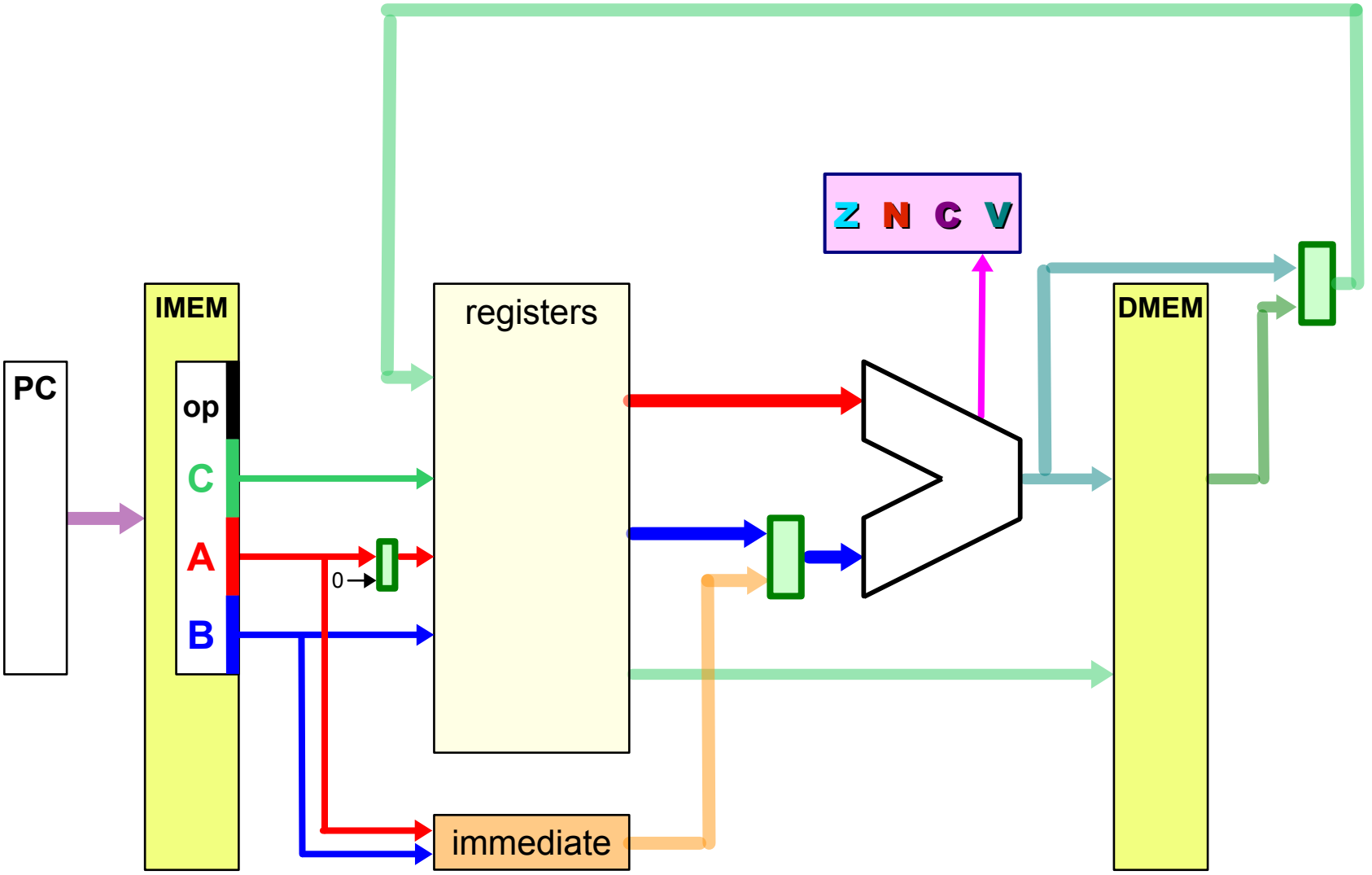
Support for `load` Instruction



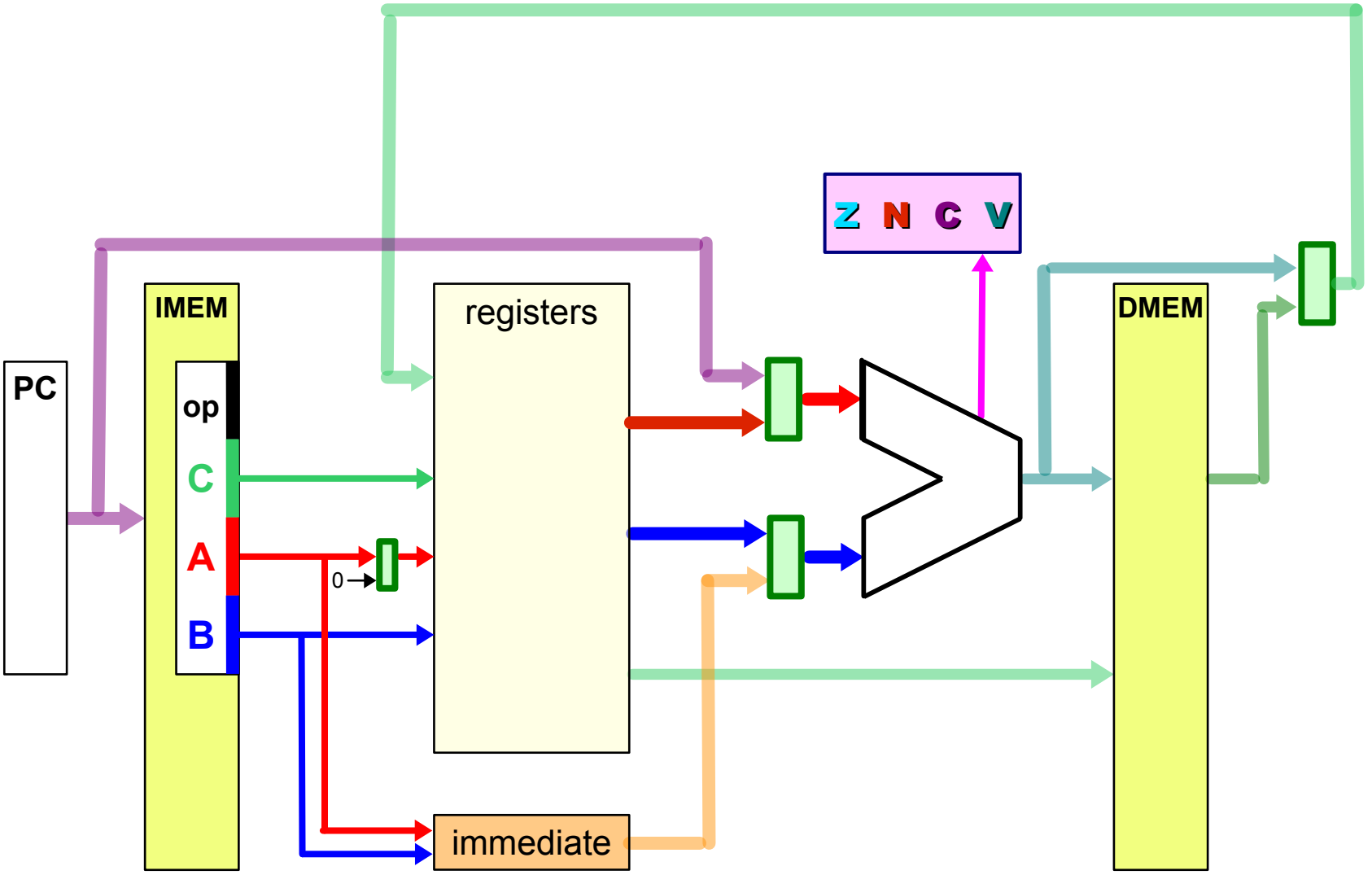
Support for `store` Instruction



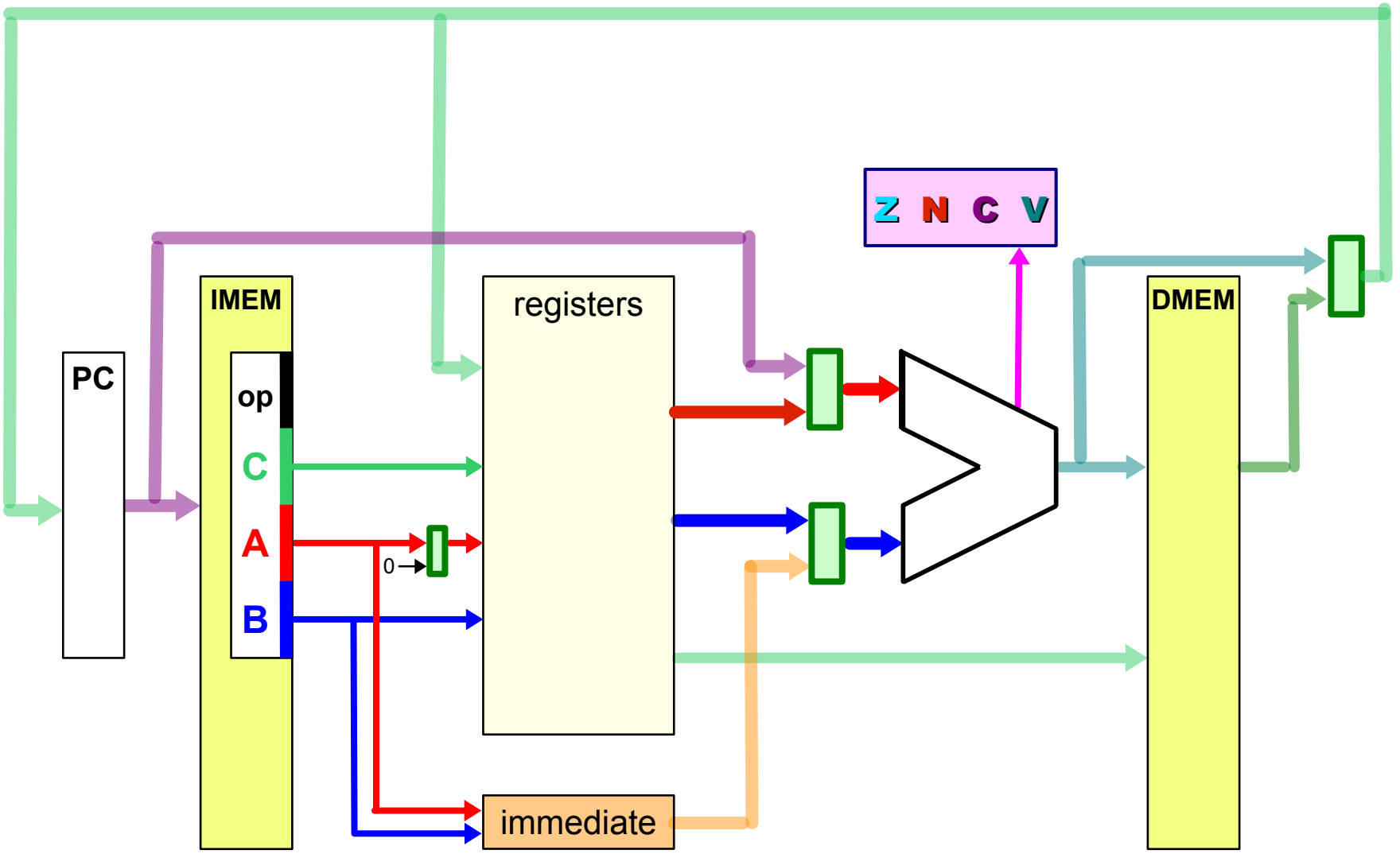
Support for `lis` Instruction



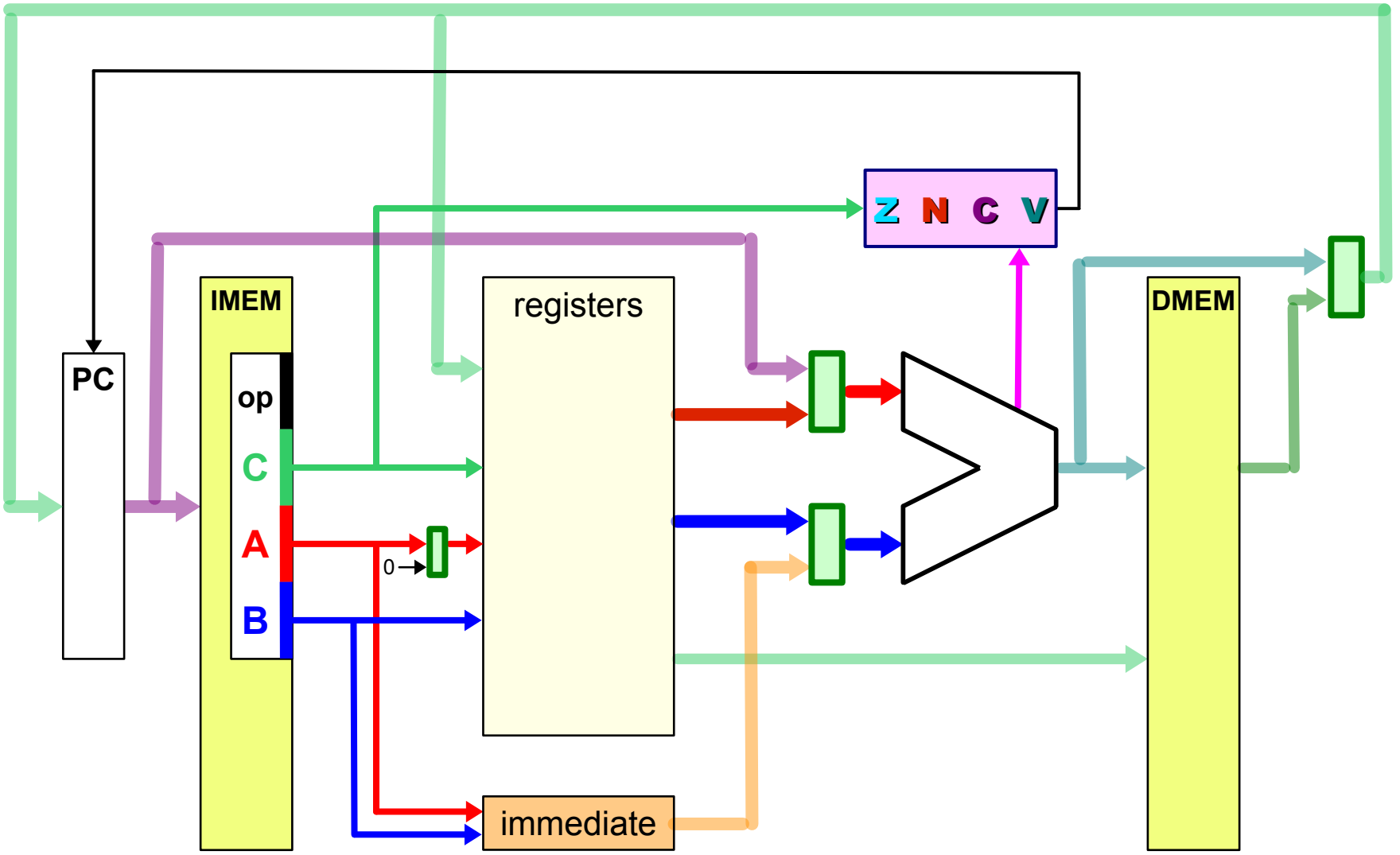
Support for `bc` Instruction (1)



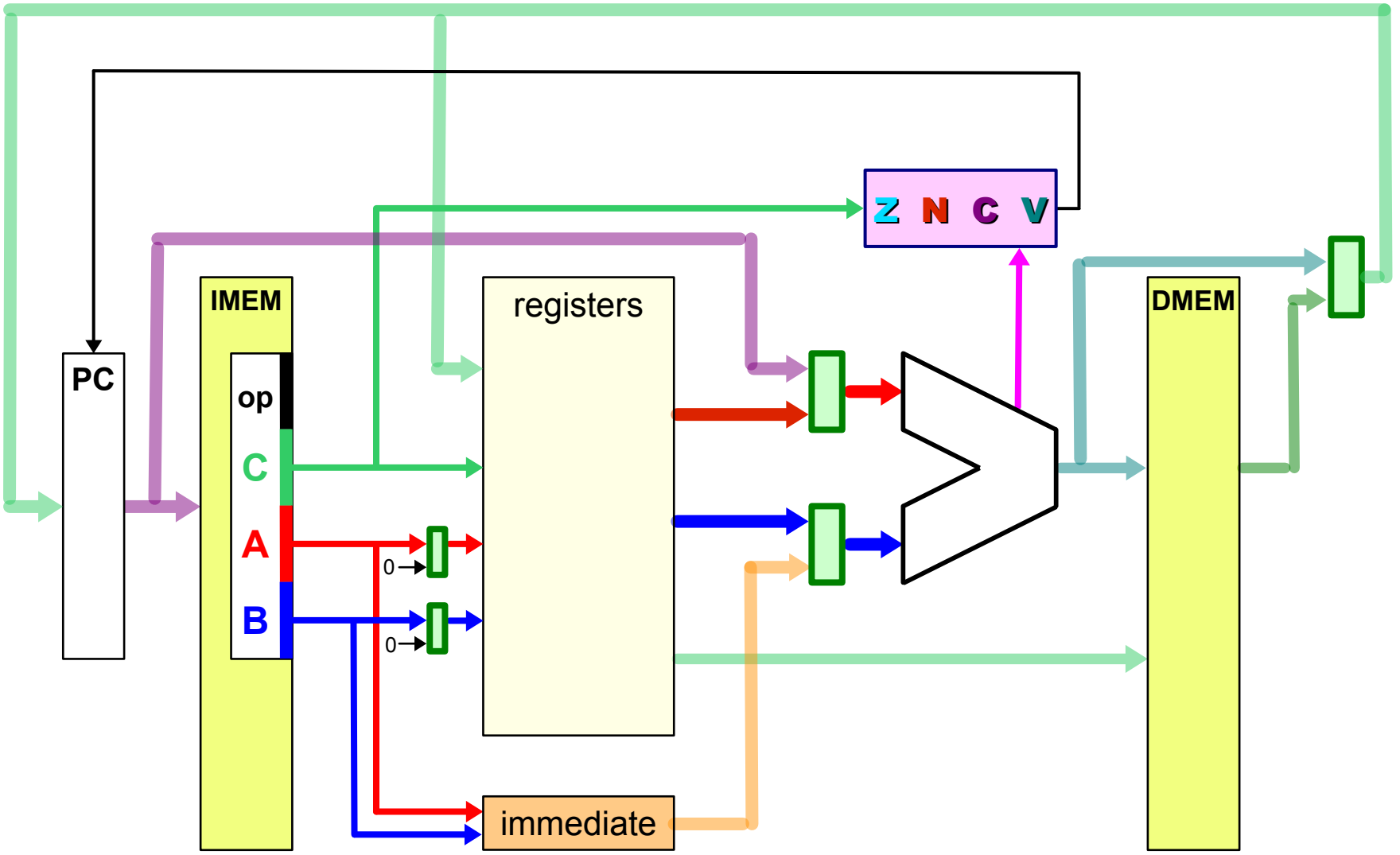
Support for `bc` Instruction (2)



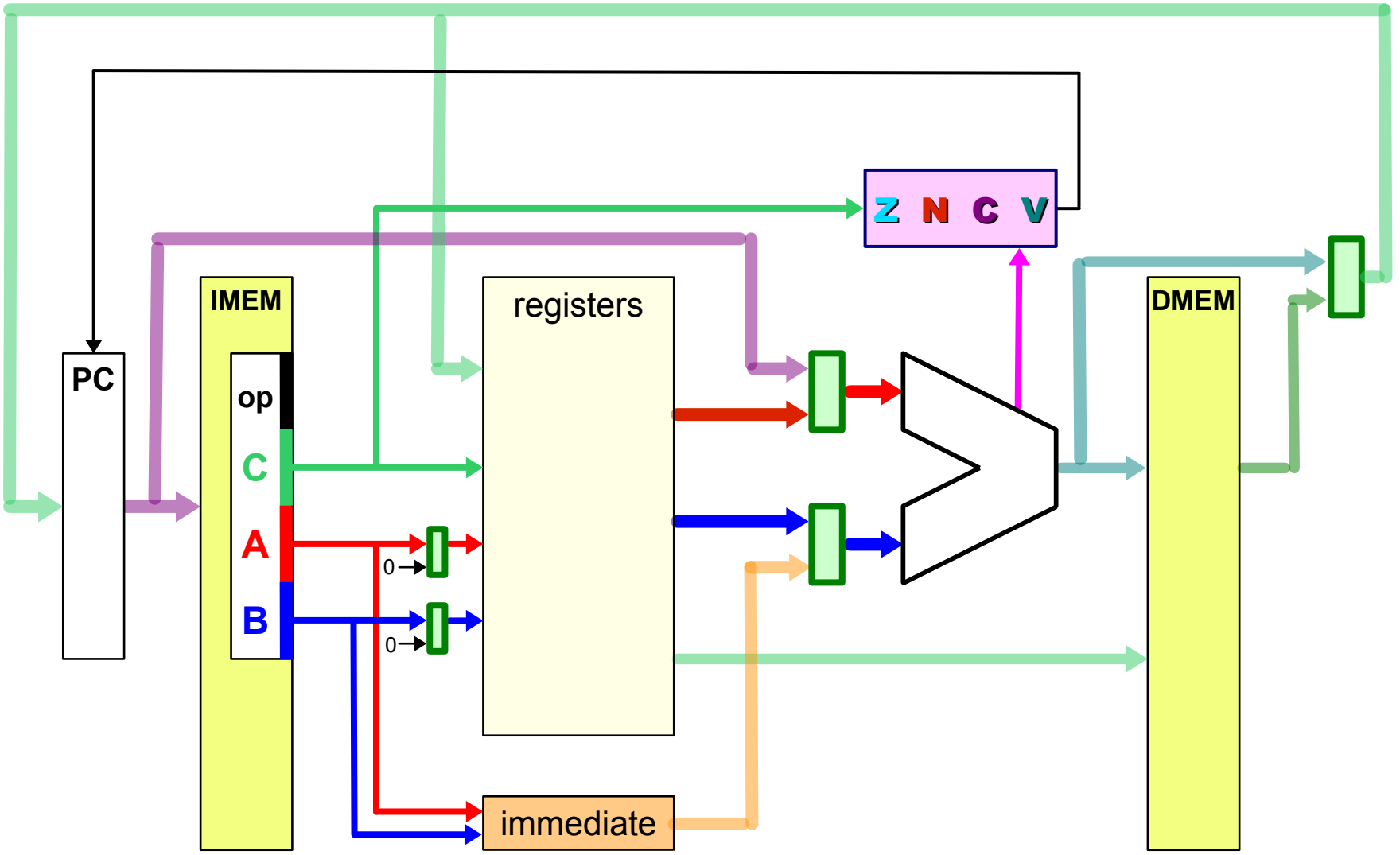
Support for `bc` Instruction



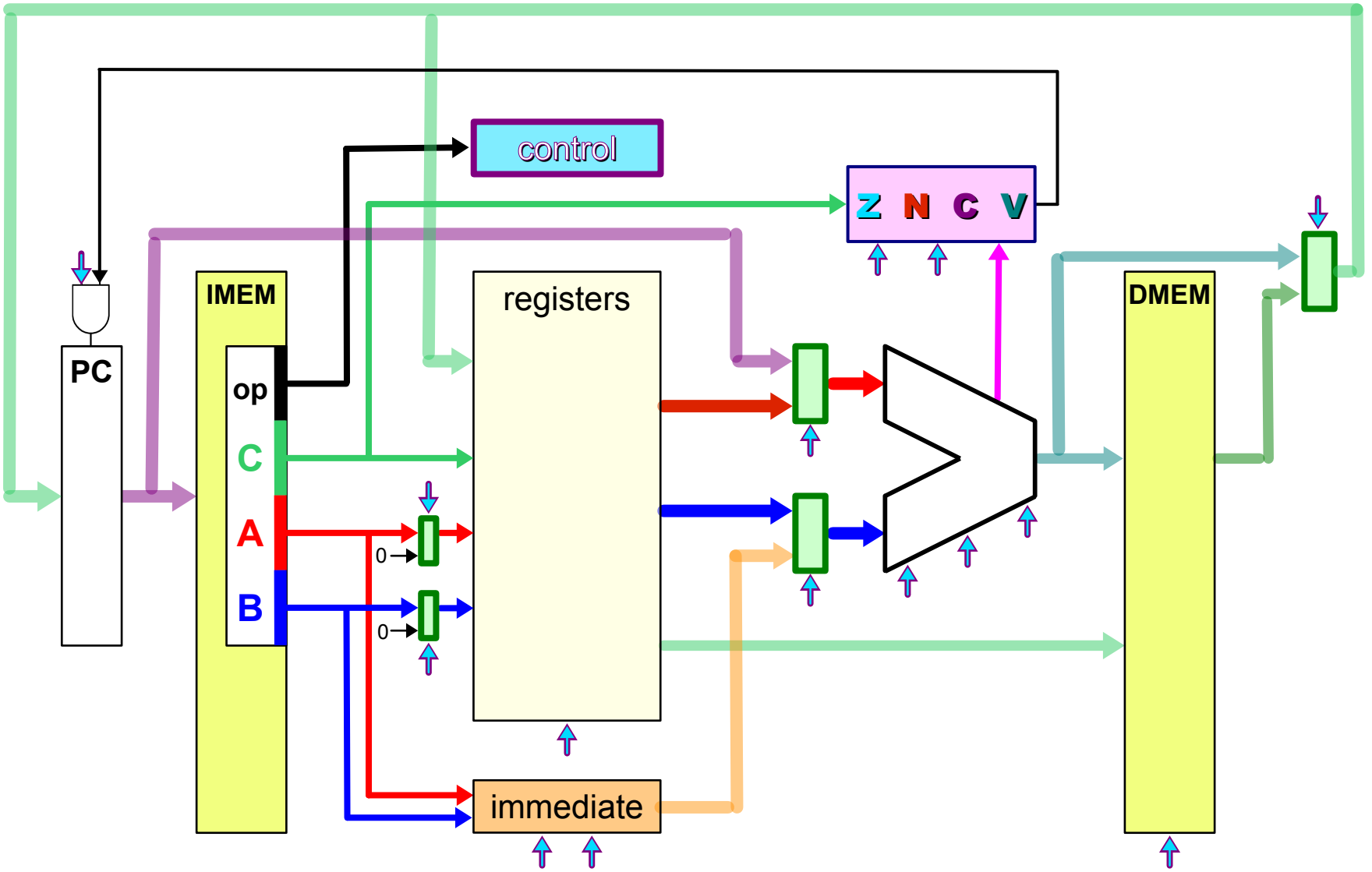
Partial Support for `blc` Instruction



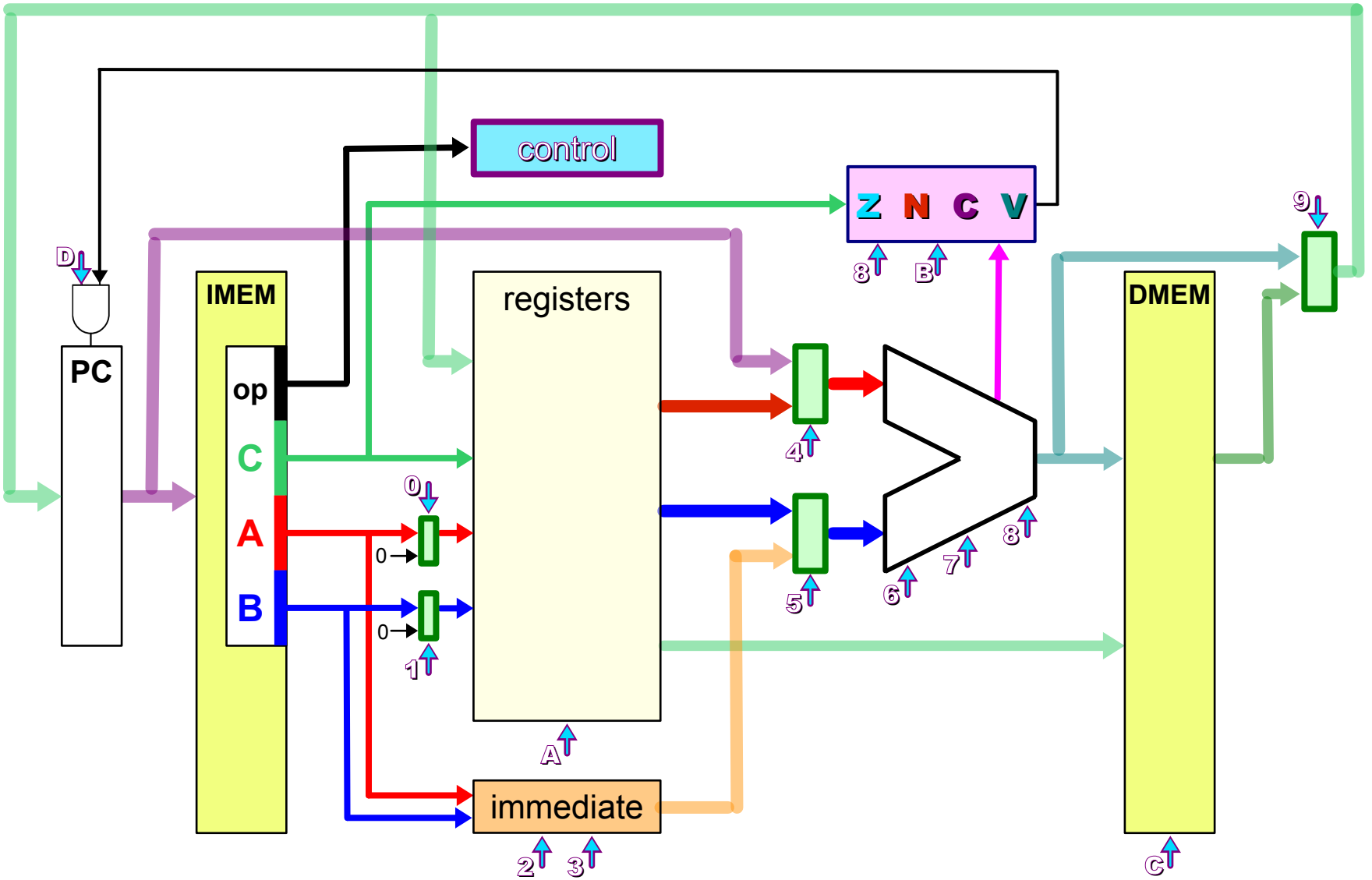
TOY Redesign *Datapath*



TOY Redesign with Control



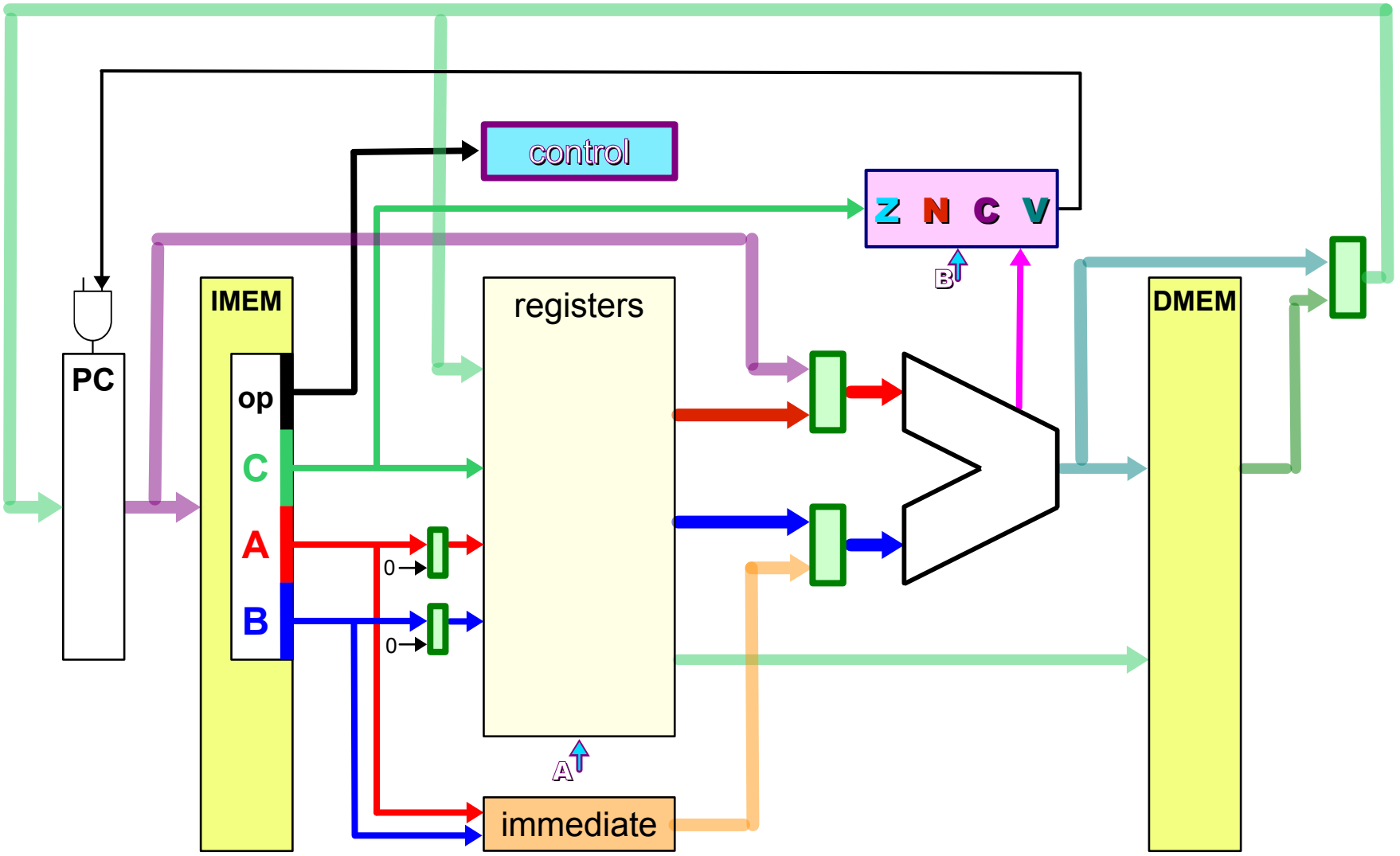
TOY Redesign with Control



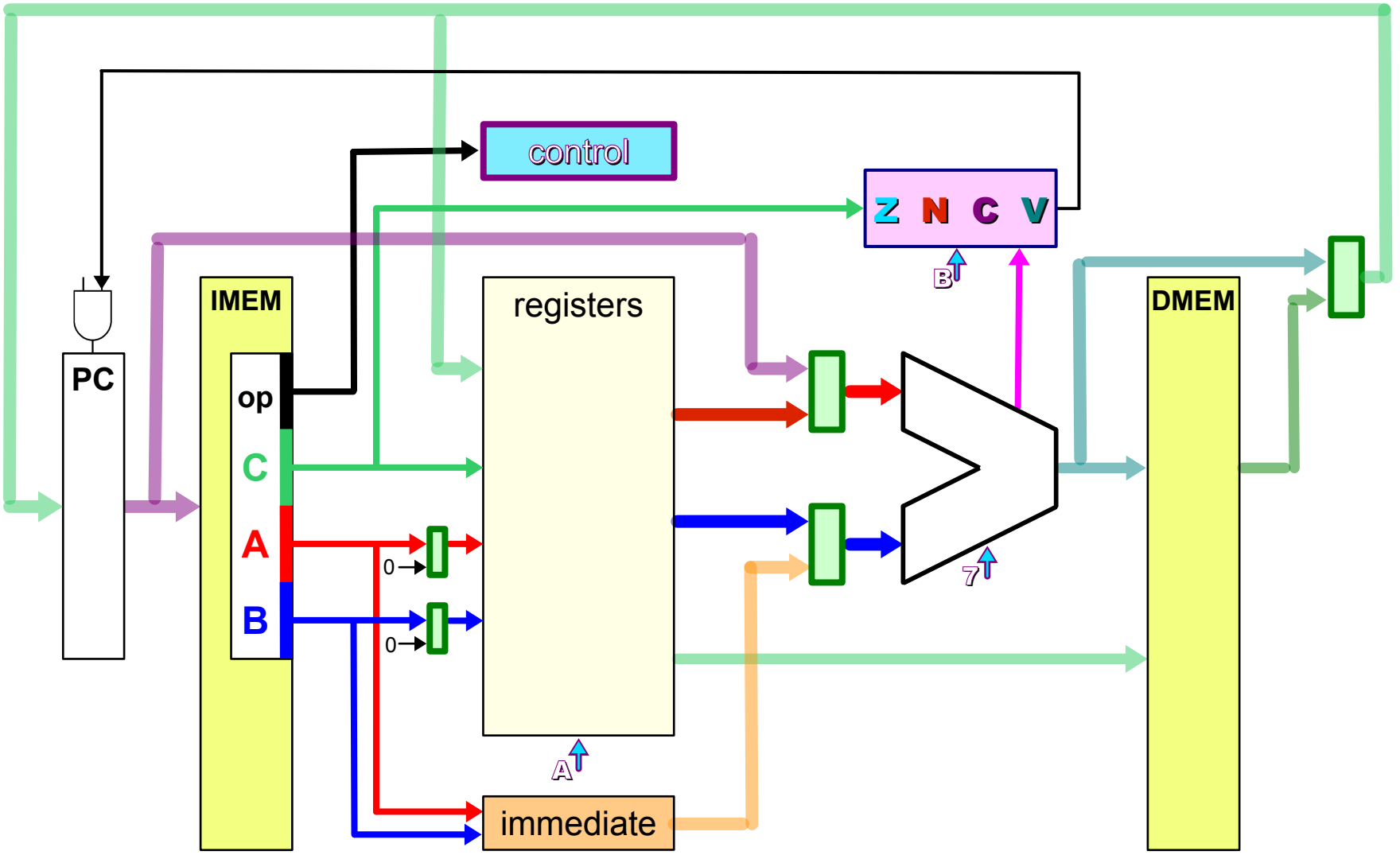
TOY Redesign Control Signals

	?	0	1
0	**A** register address	**IR**[7..4]	0000
1	**B** register address	**IR**[3..0]	0000
2	Immediate input	**IR**[7..0]	0000 **IR**[3..0]
3	Immediate output	(sign extended) [7..0]	[F..8] 11111111
4	**B** bus source	**B** register	Immediate value
5	**A** bus source	**A** register	**PC**
6	Invert **A** bus	no	yes
7	Invert **B** bus	no	yes
8	ALU output	Arithmetic (full adder)	Logical (AND gate)
9	**C** bus source	**ALU**	Data memory
A	Register file write	no	yes
B	Condition write	no	**Z N C V** or **Z N**
C	Data memory write	no	yes
D	Possible branch?	no	yes

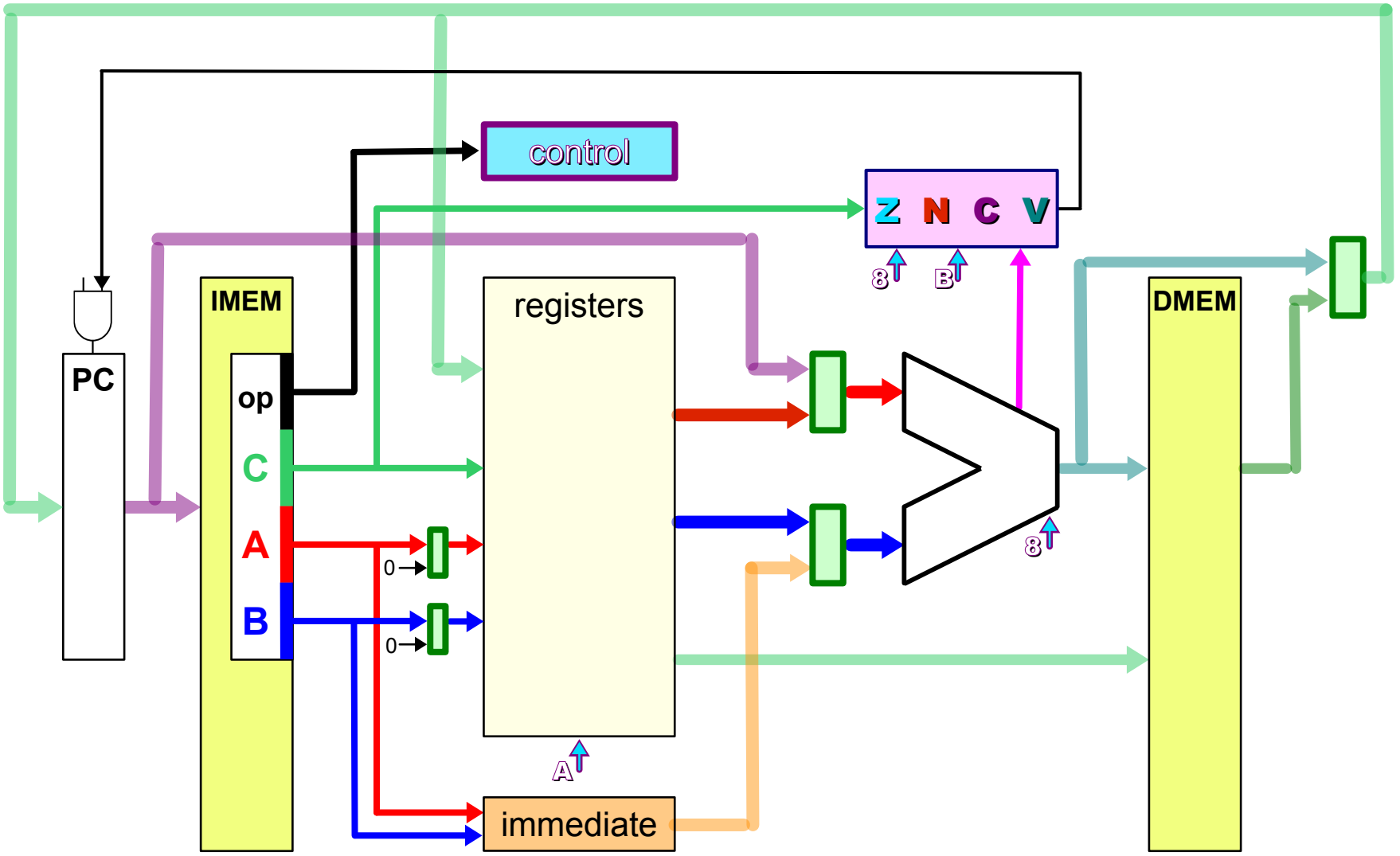
Control = 1: add instruction



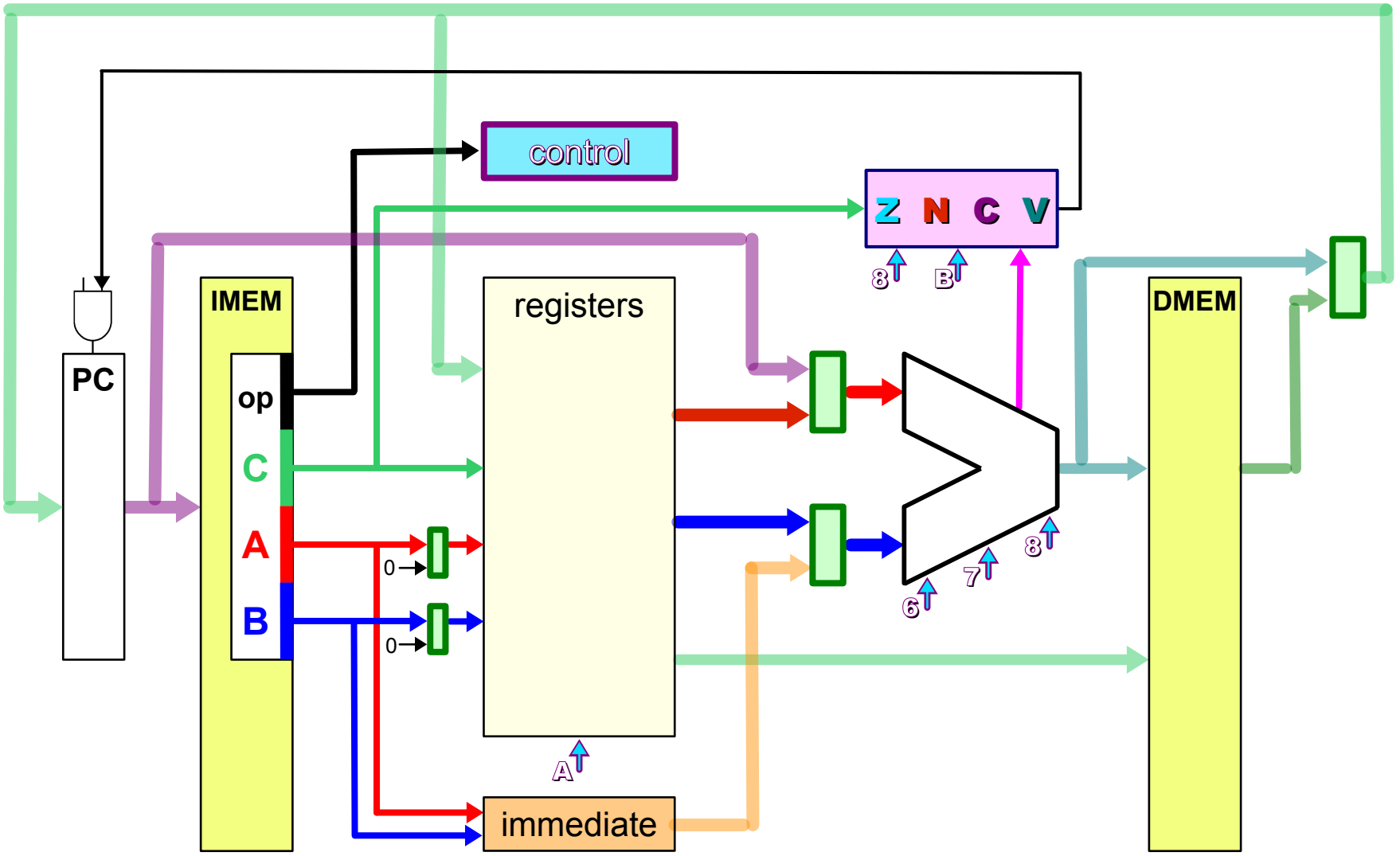
Control = 1: `sub` instruction



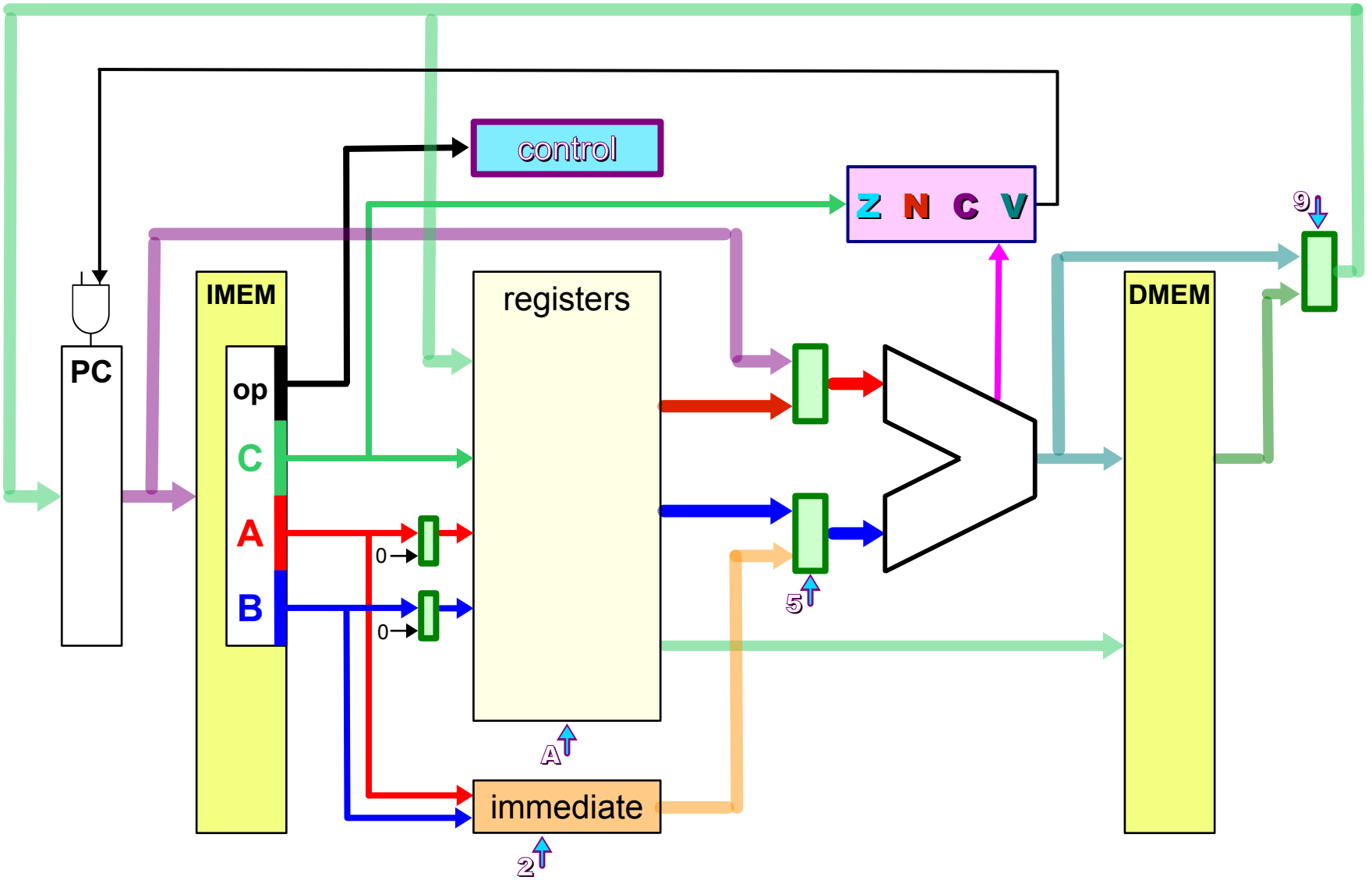
Control = 1: `and` instruction



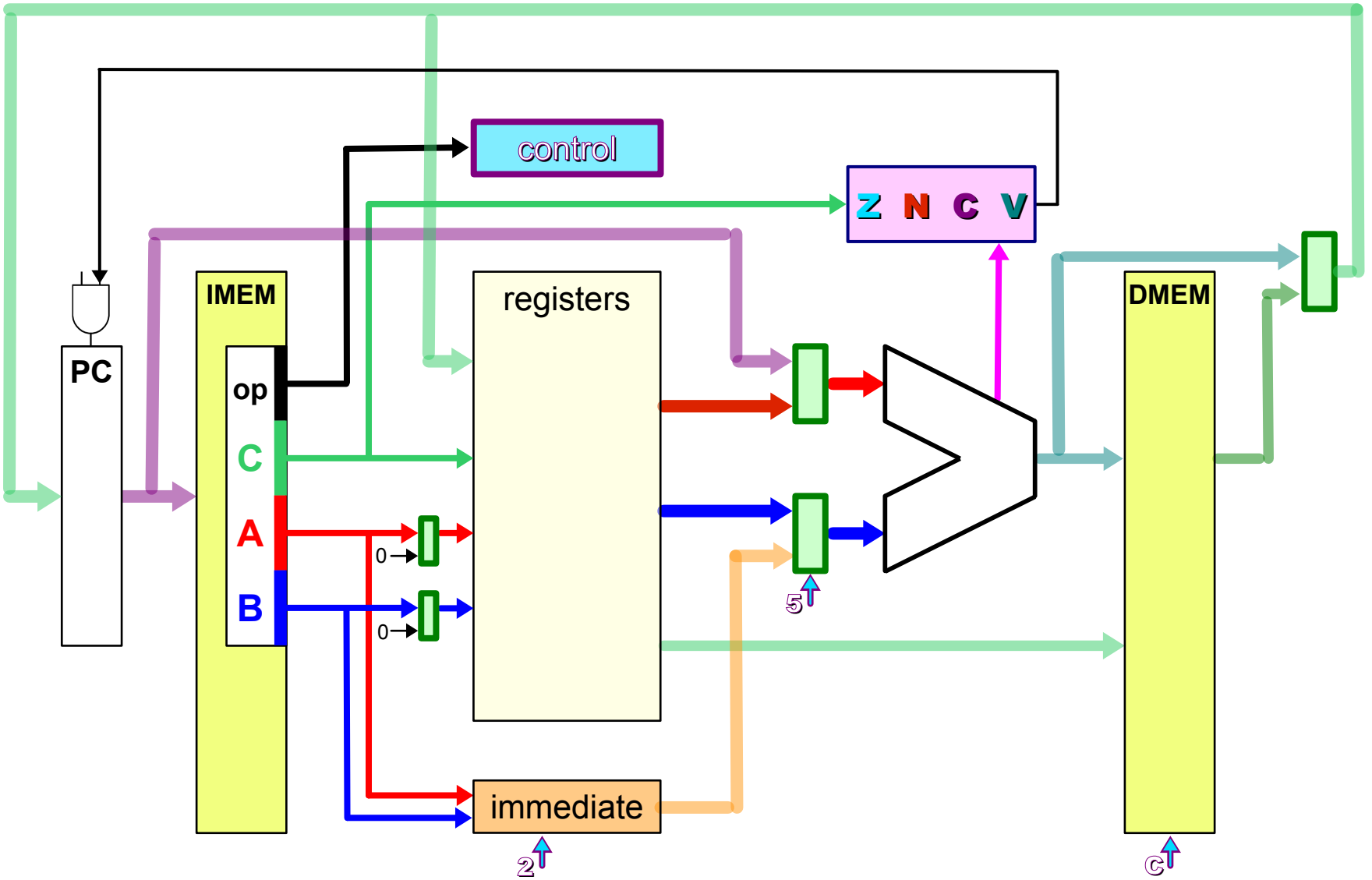
Control = 1: `nor` instruction



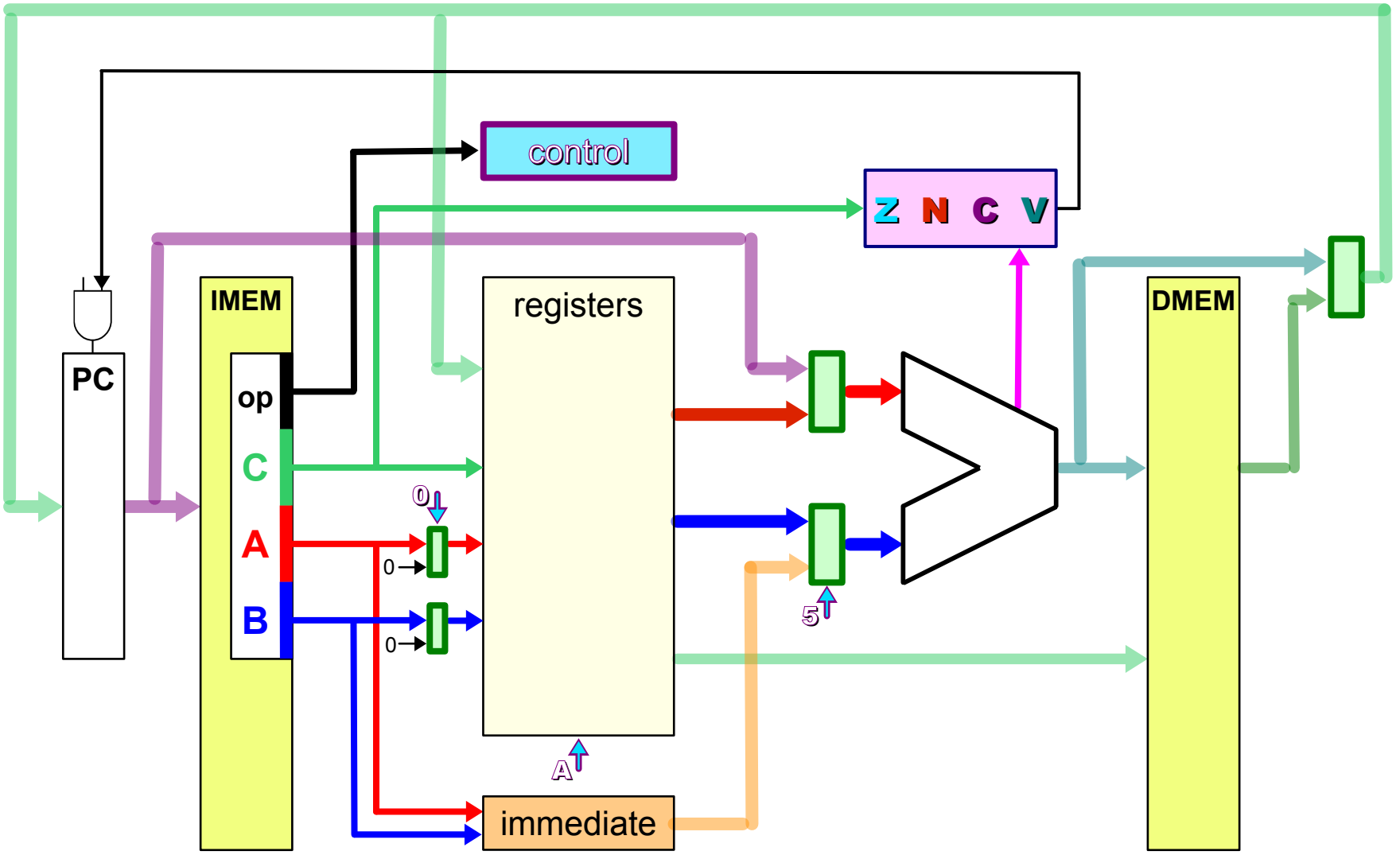
Control = 1: load instruction



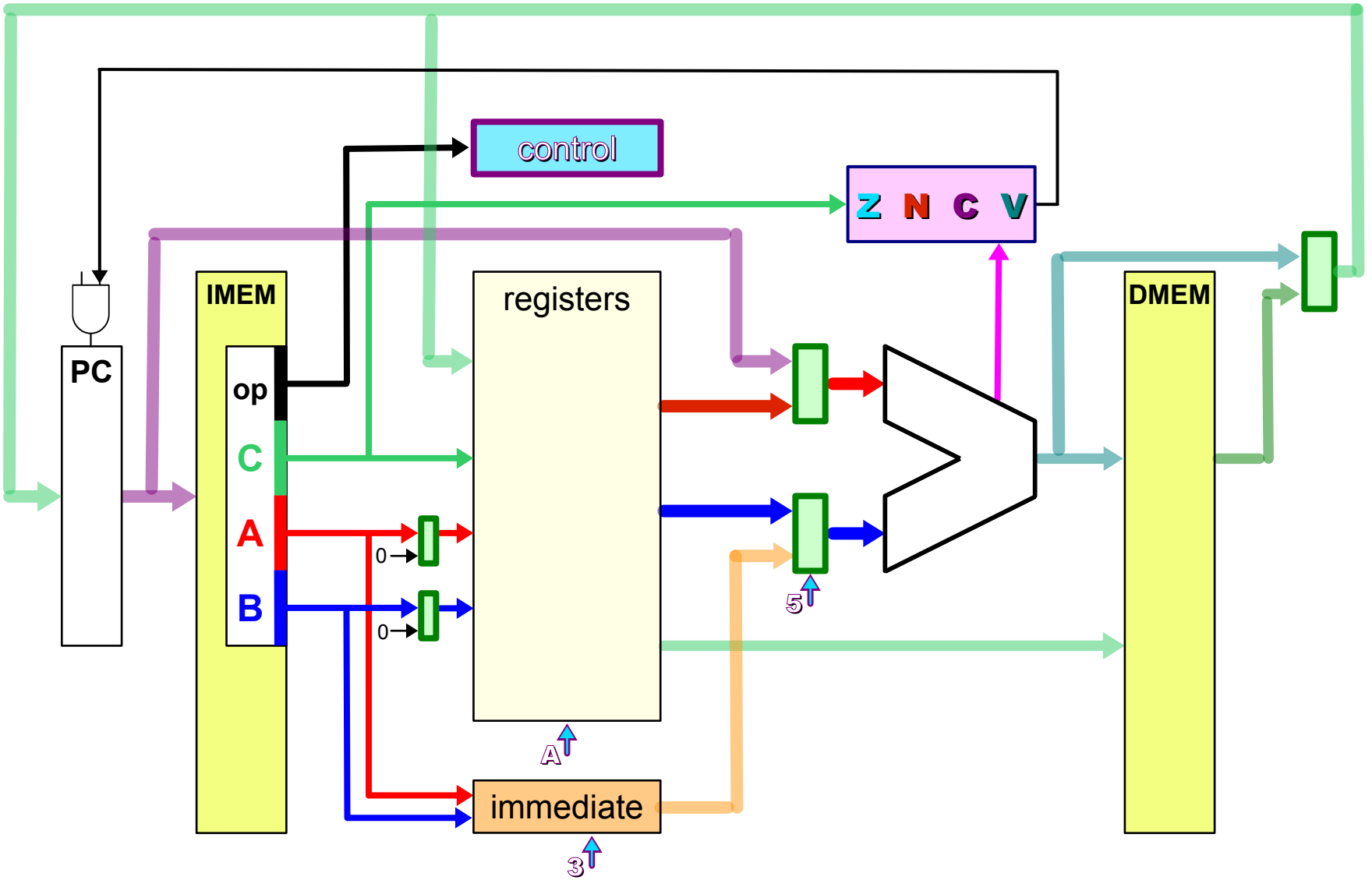
Control = 1: store instruction



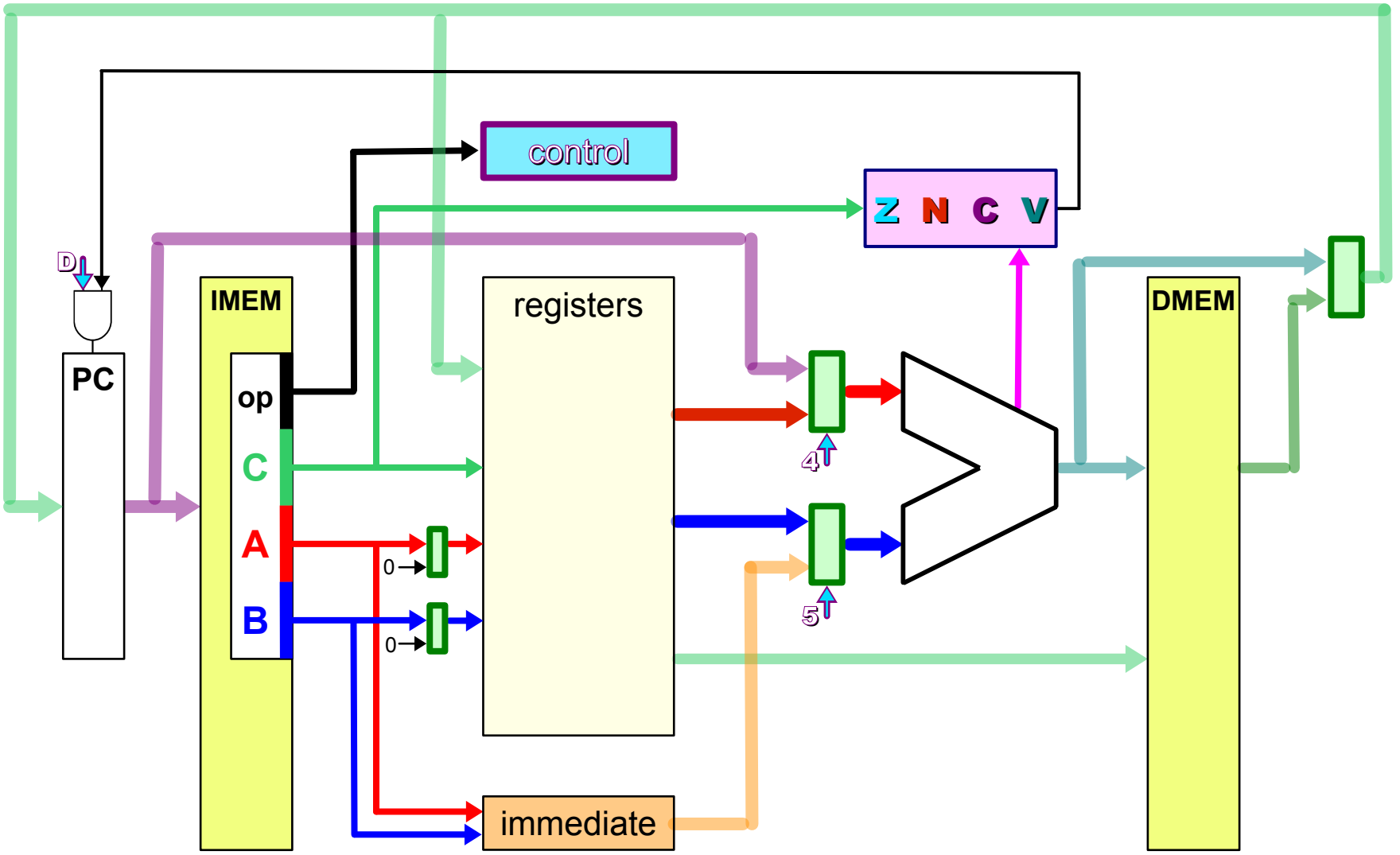
Control = 1: `lis` instruction



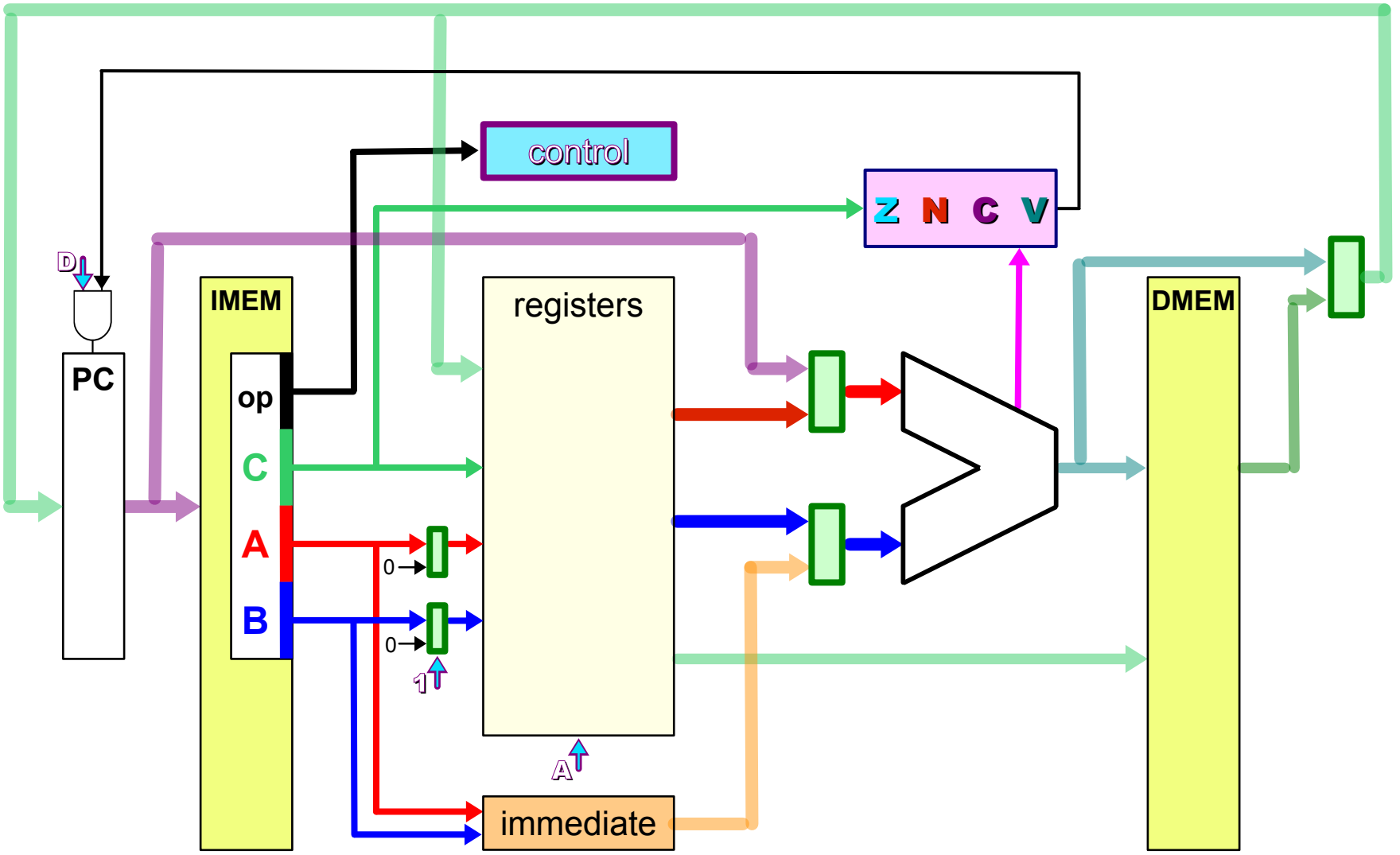
Control = 1: `lih` instruction



Control = 1: `bc` instruction



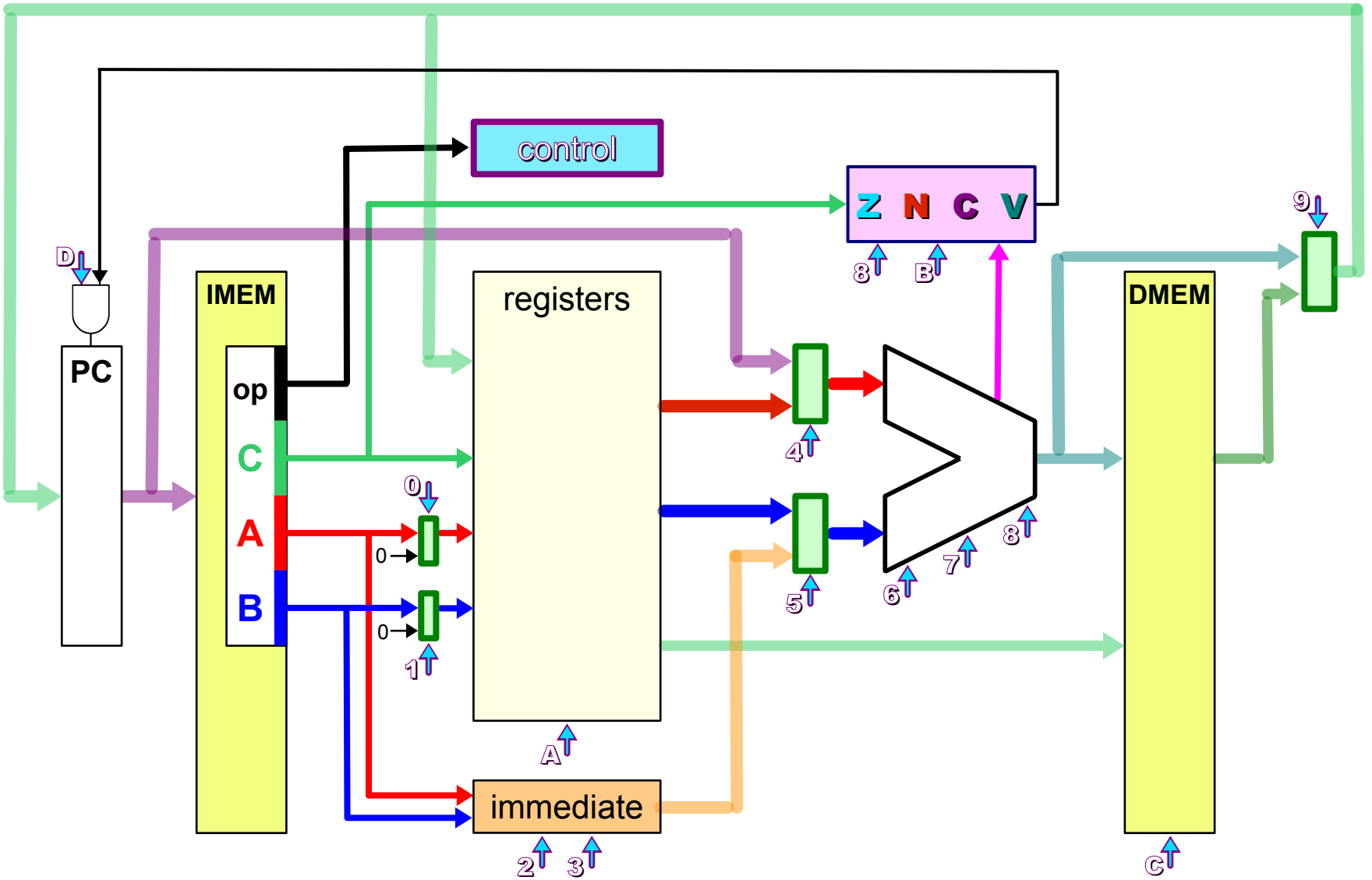
Control = 1: `bcl` instruction



TOY Redesign Control Signals

	?	0	1
0	**A** register address	**IR**[7..4]	0000
1	**B** register address	**IR**[3..0]	0000
2	Immediate input	**IR**[7..0]	0000 **IR**[3..0]
3	Immediate output	(sign extended) [7..0]	[F..8] 11111111
4	**B** bus source	**B** register	Immediate value
5	**A** bus source	**A** register	**PC**
6	Invert **A** bus	no	yes
7	Invert **B** bus	no	yes
8	ALU output	Arithmetic (full adder)	Logical (AND gate)
9	**C** bus source	**ALU**	Data memory
A	Register file write	no	yes
B	Condition write	no	**Z N C V** or **Z N**
C	Data memory write	no	yes
D	Possible branch?	no	yes

TOY Single Cycle Implementation



TOY Single Cycle Implementation

Single cycle instruction execution

- Combinational circuit with
- Memory elements constant til the end of a cycle
  - **PC**, register file, **Z N C V**, **DMEM**

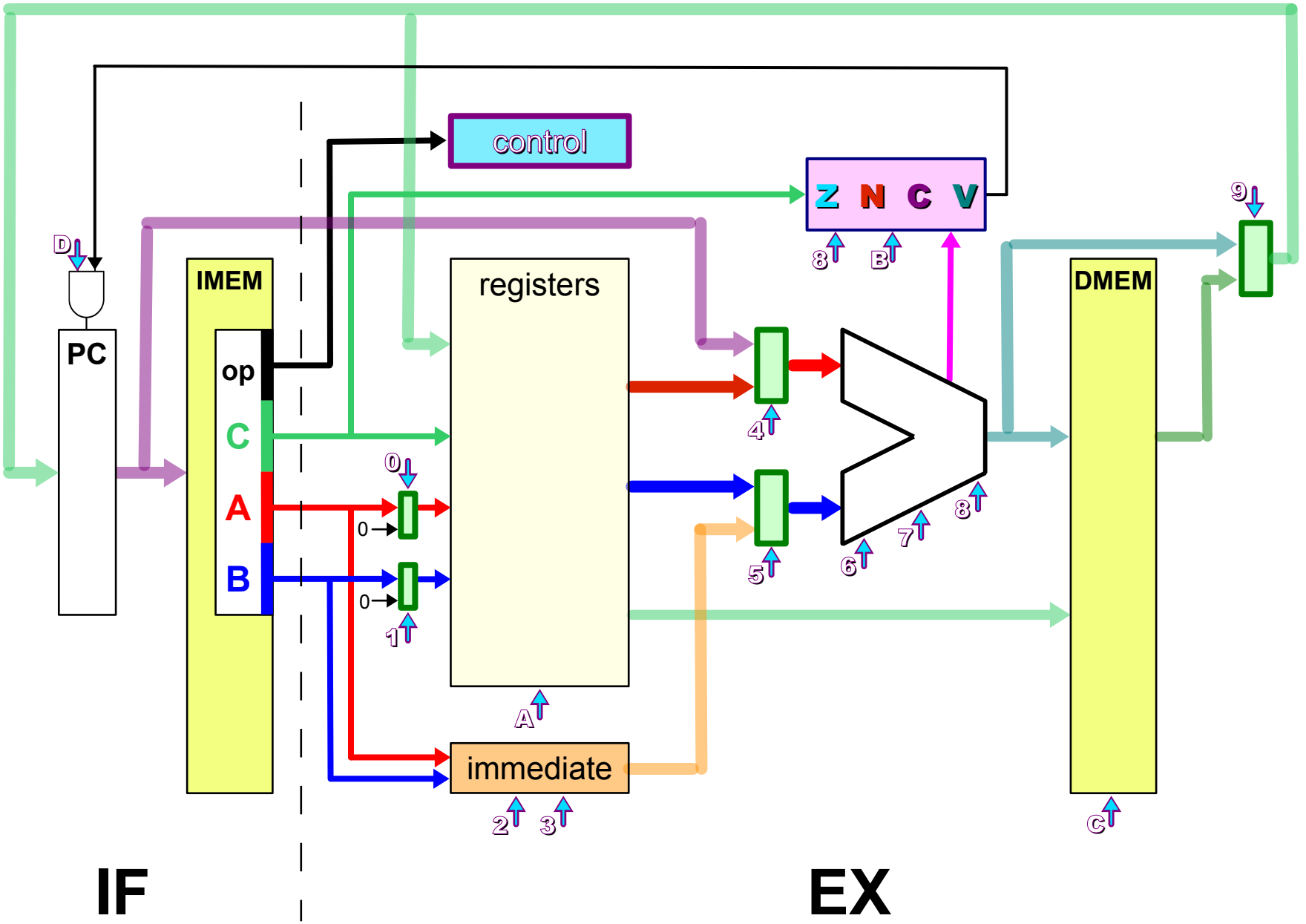
`lih` instruction not fully implemented

- Need to channel **C** address to **A MUX**
- Need 3rd input to **4** **MUX** (top 8 bits only)
  - **1111 1111** → **A** bus [F..8]

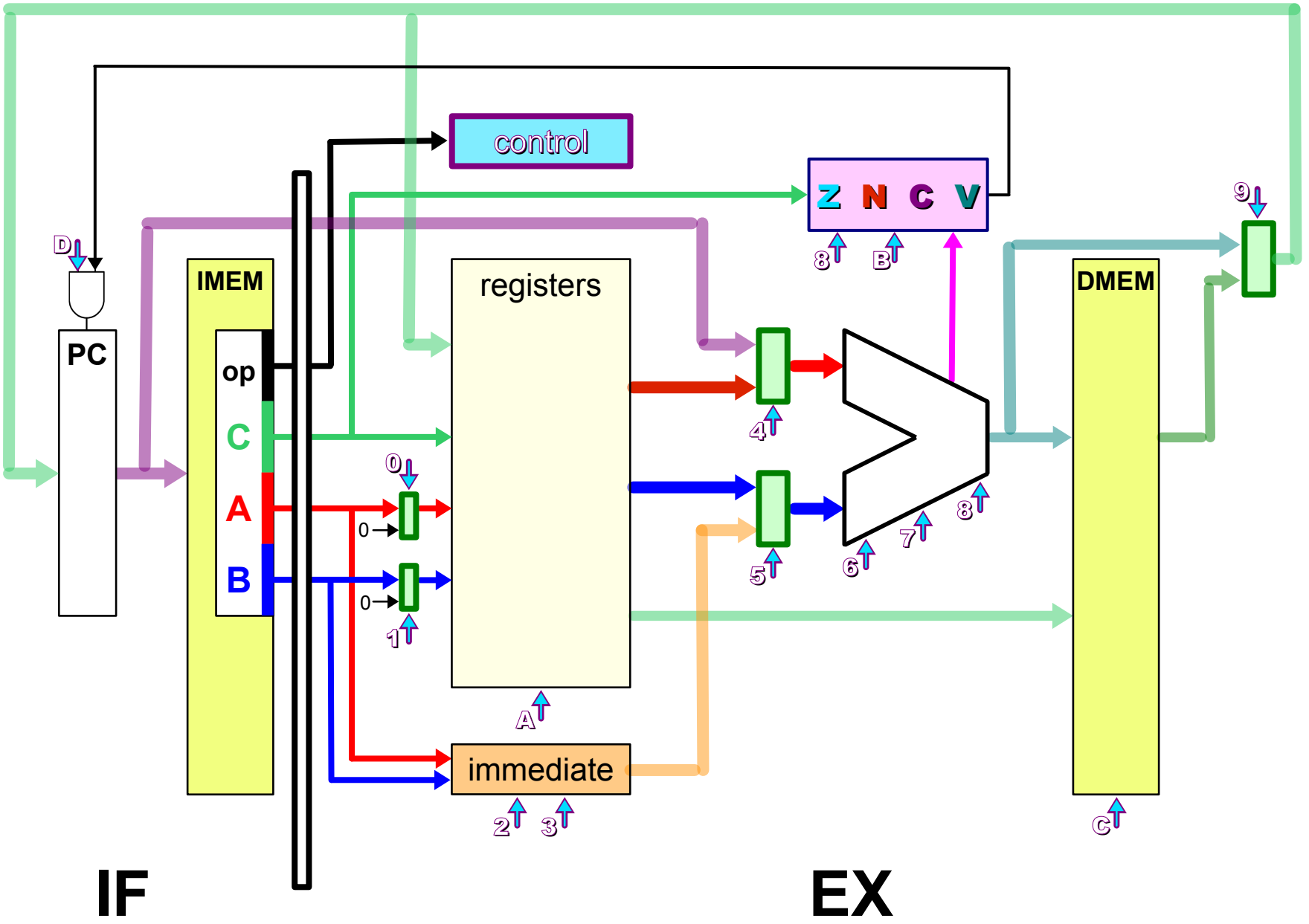
`bcl` instruction not fully implemented

- Need to channel **B** address to **C** decoder
- Need to resolve structural hazard for the **C** bus
  - **PC** → register file write data input
  - Register **A** + 0 → **PC**

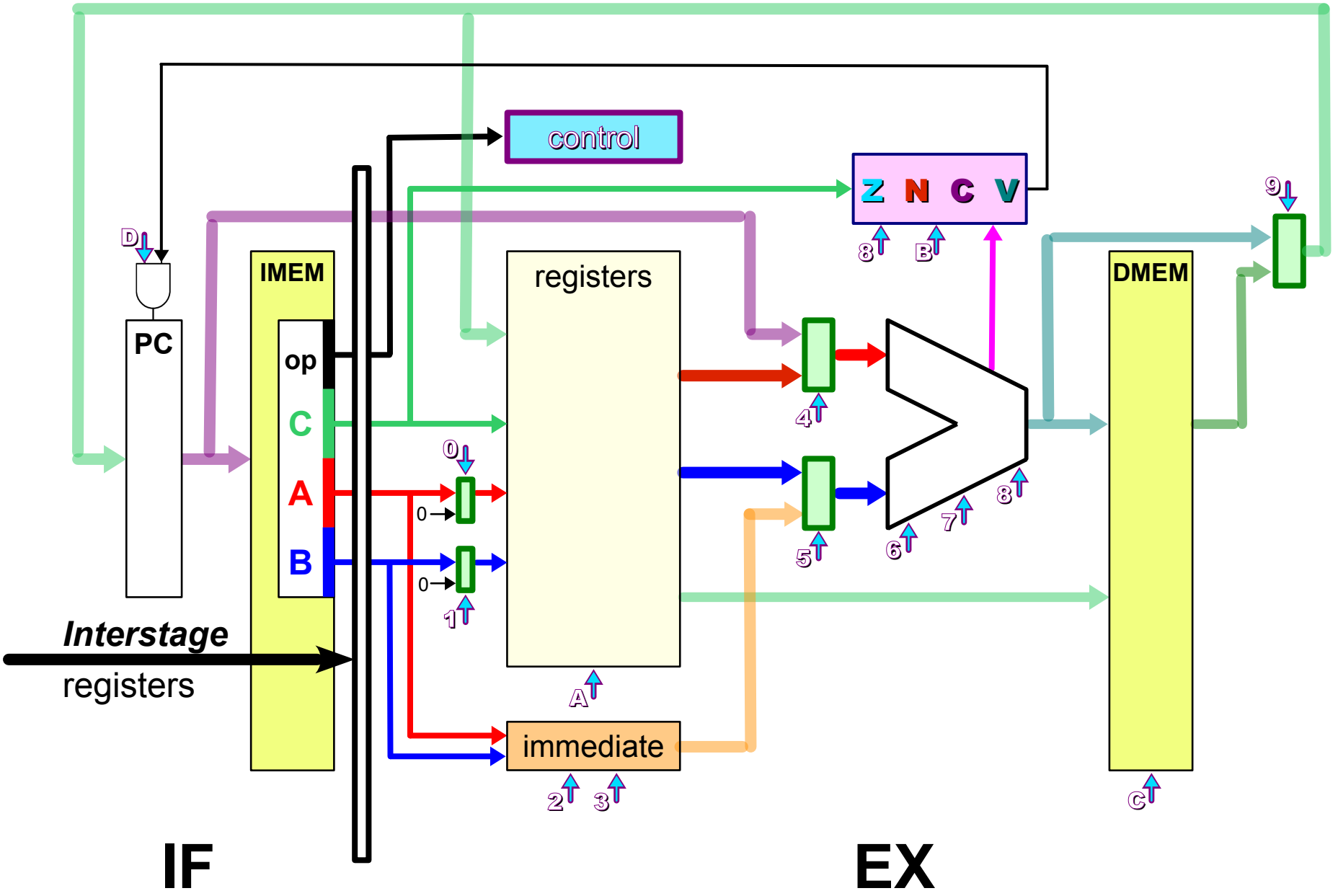
TOY Pipeline Implementation?



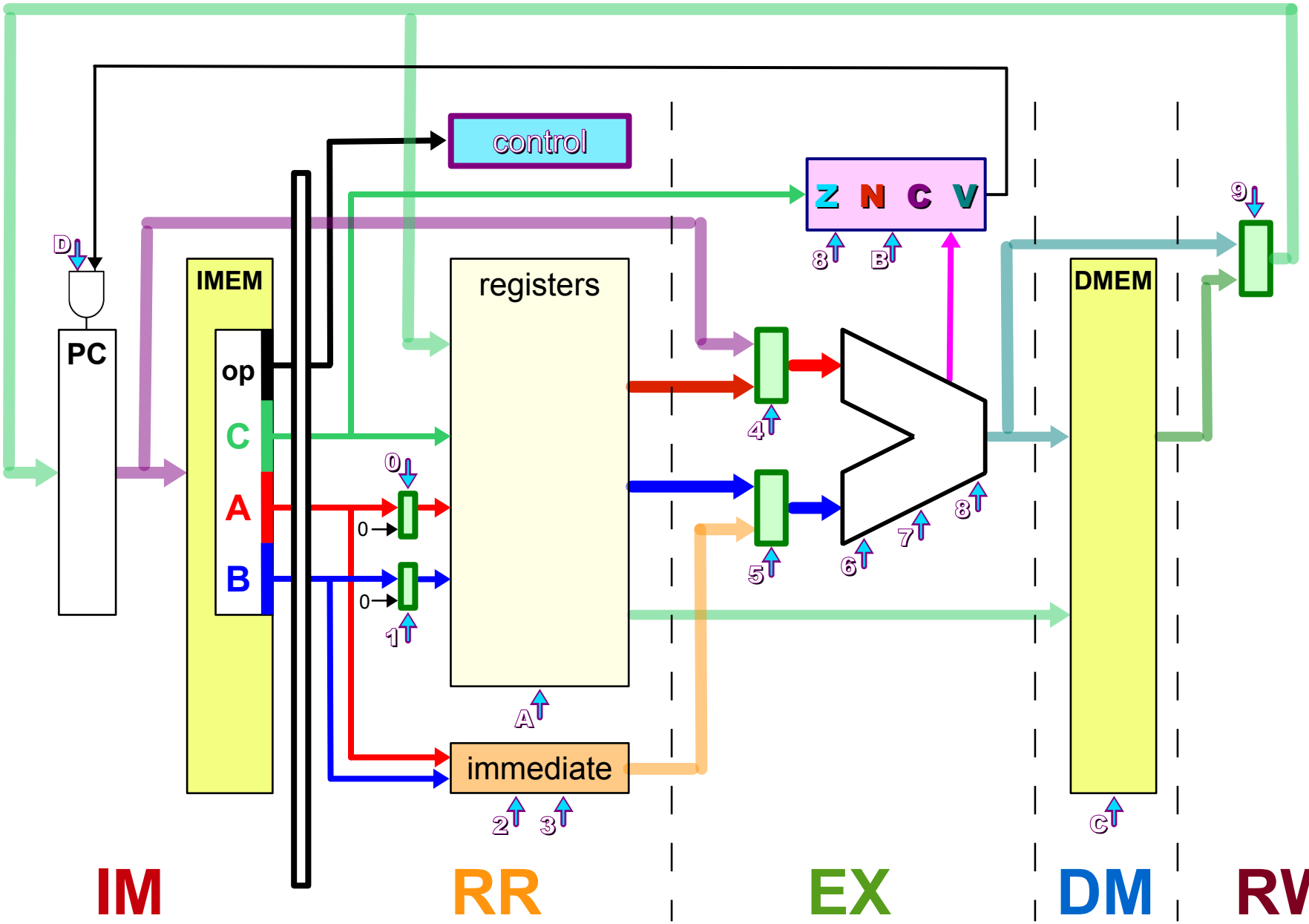
TOY 2 Stage Pipeline



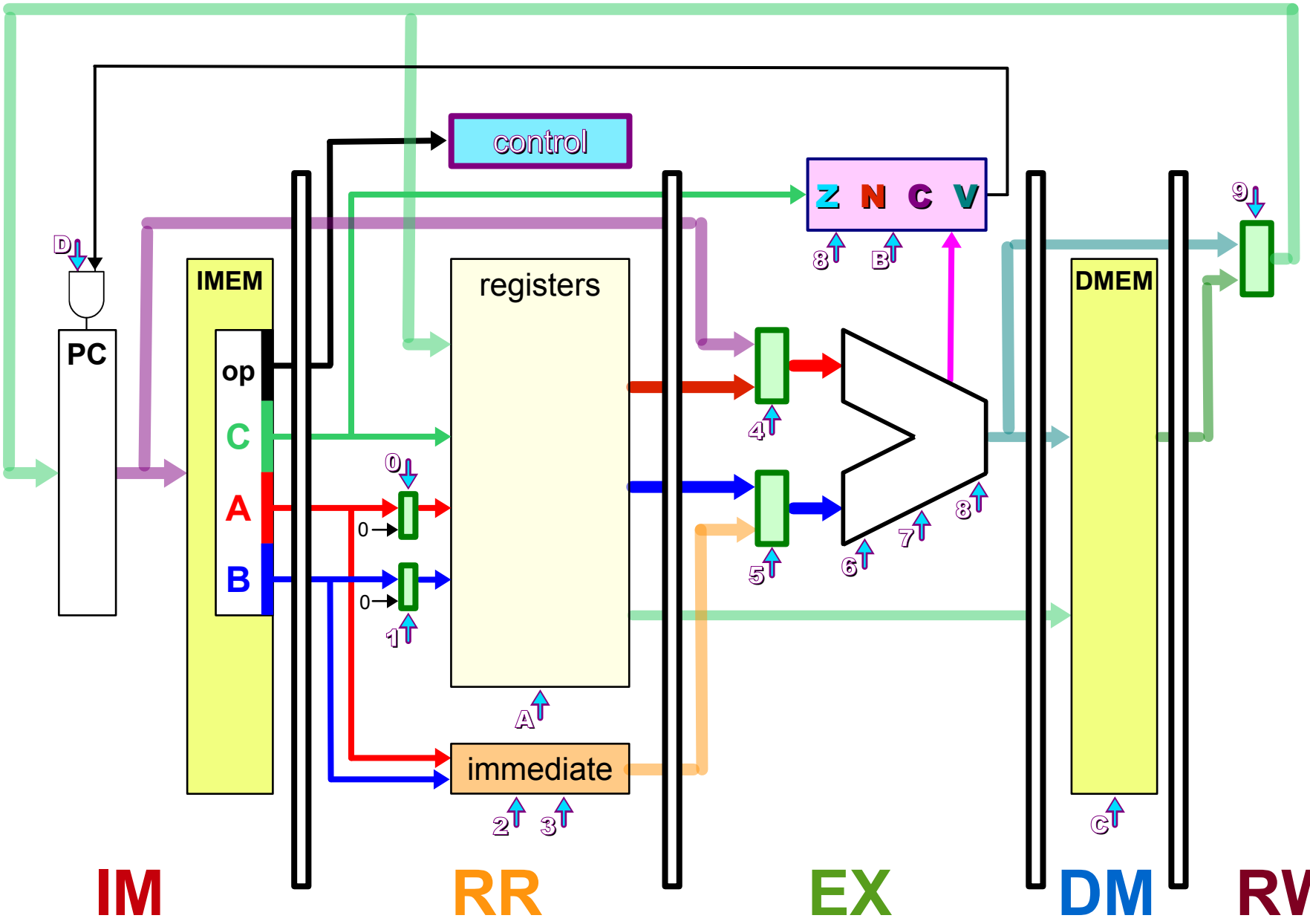
TOY 2 Stage Pipeline



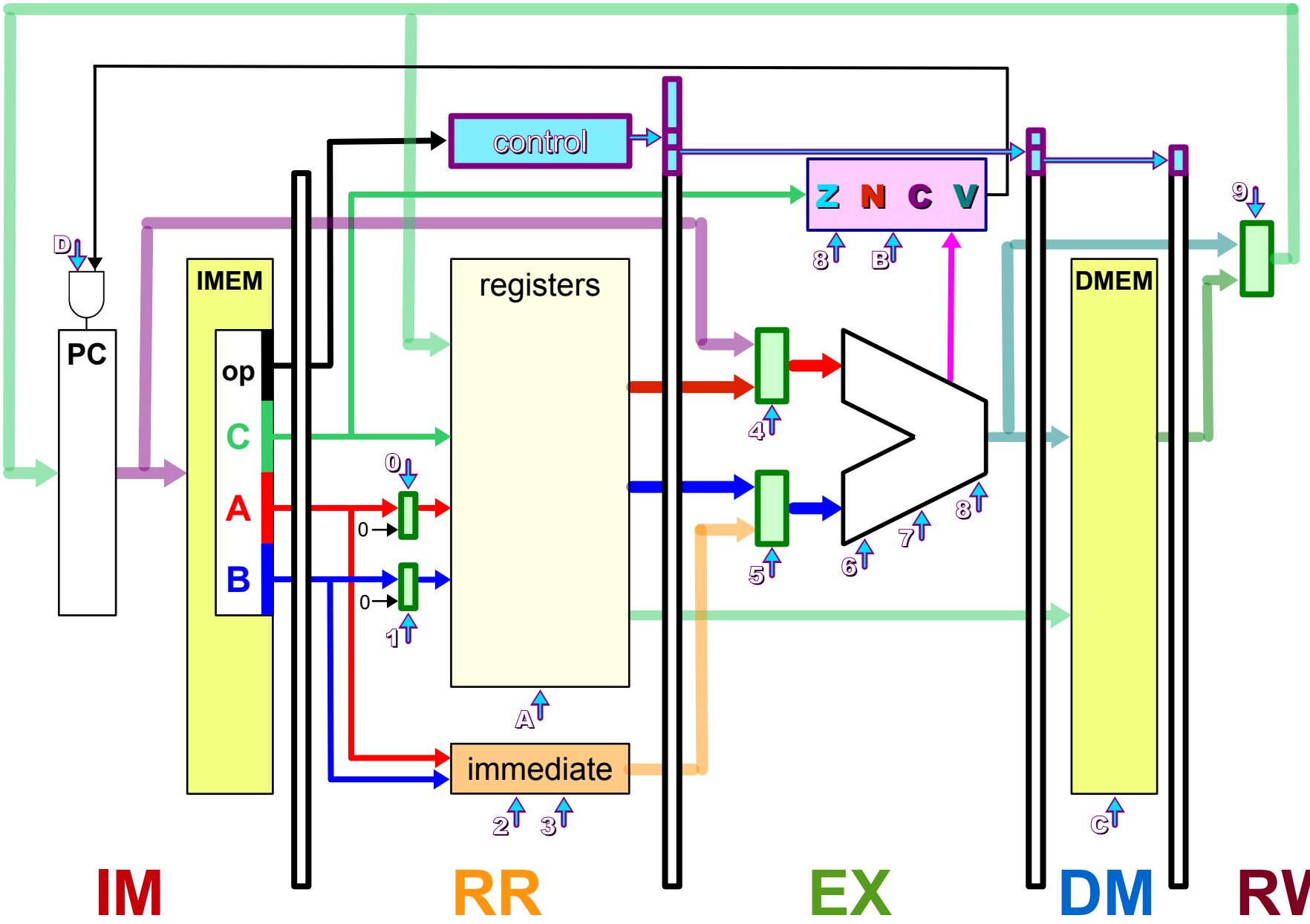
TOY 5 Stage Pipeline ??



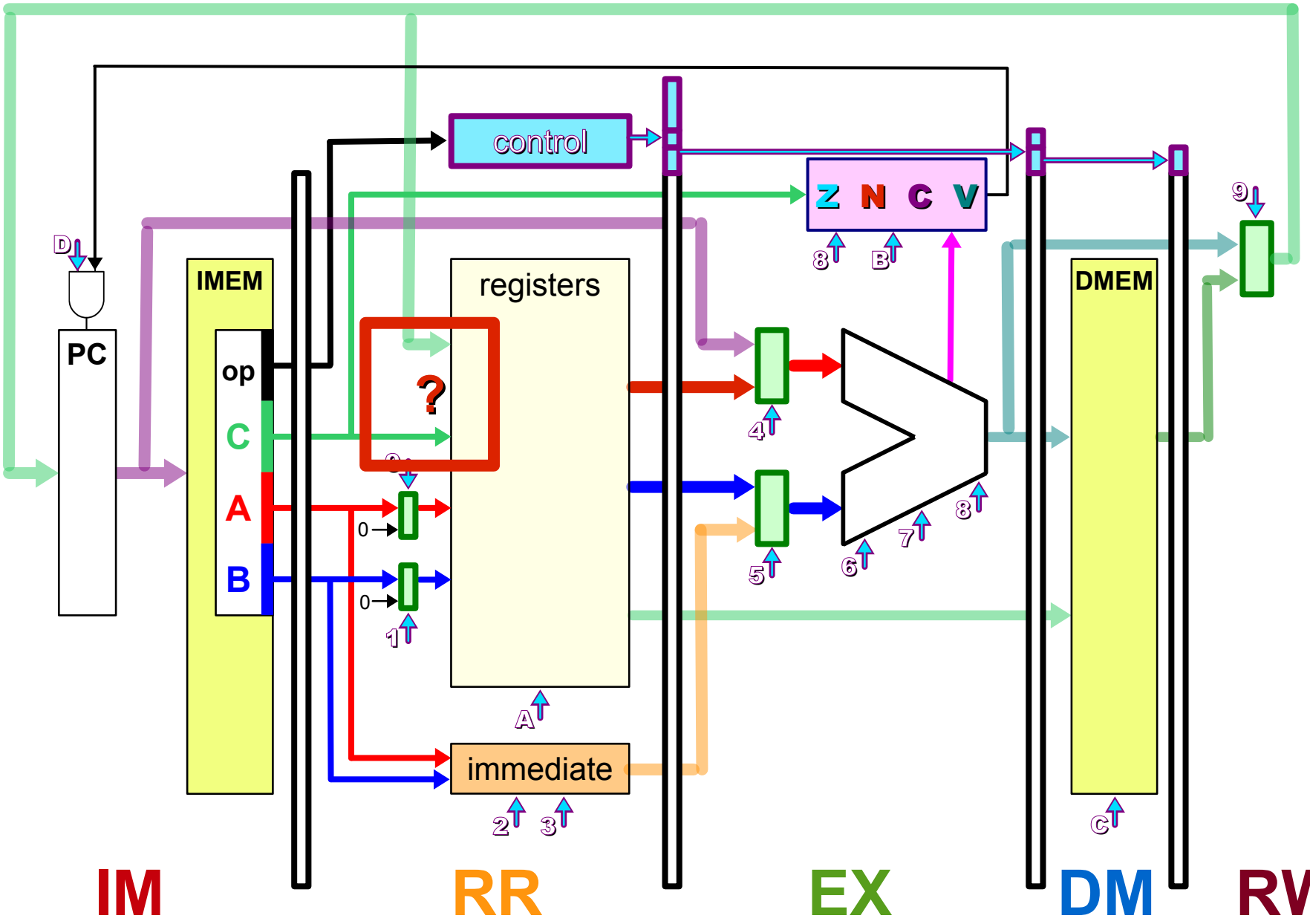
TOY 5 Stage Pipeline ?



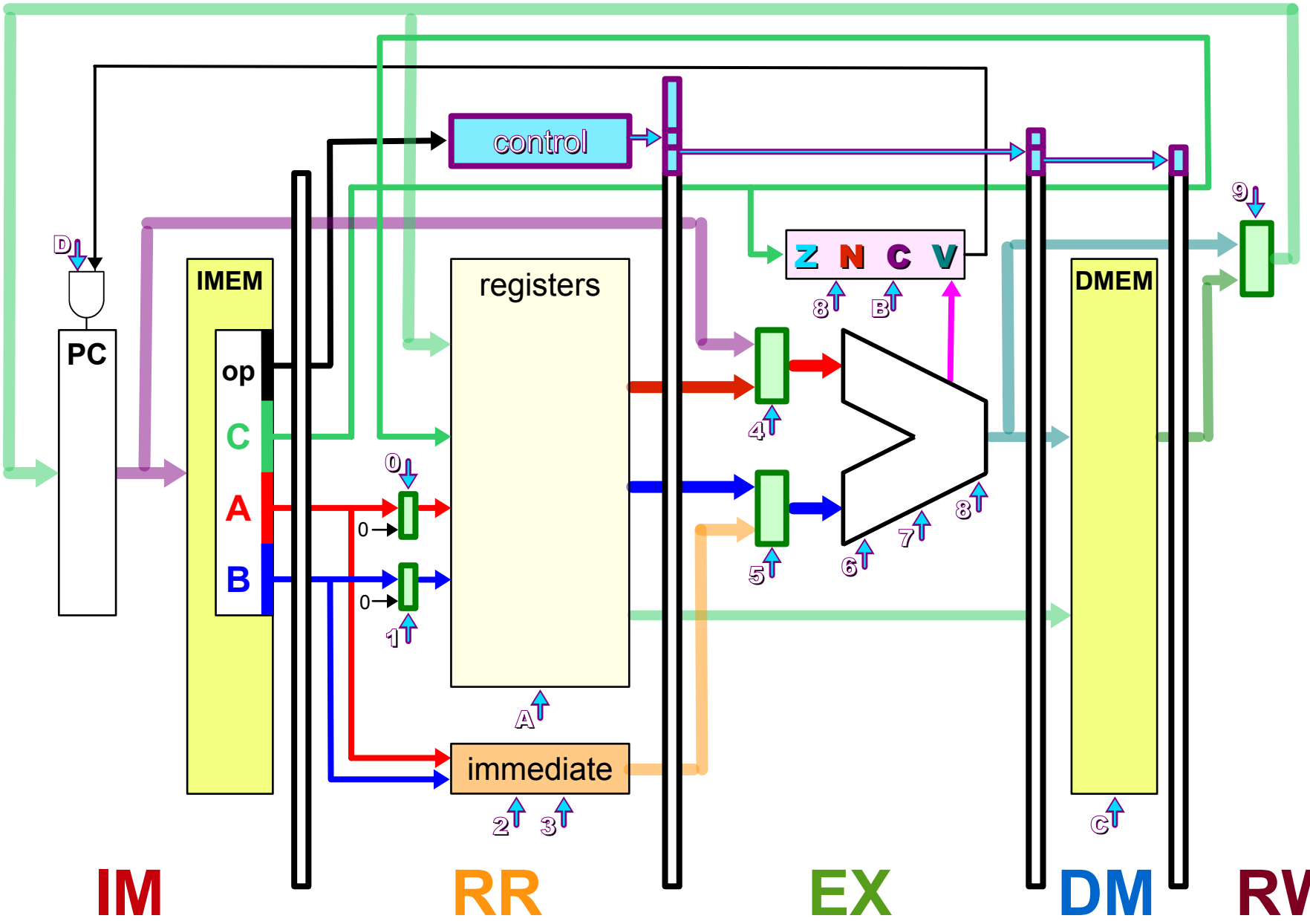
TOY 5 Stage Pipeline ?



TOY 5 Stage Pipeline ?



TOY 5 Stage Pipeline !



Performance Equations

Definition of performance

performance: $P_x \equiv \frac{1}{T_x}$ ***relative performance:*** $\frac{P_x}{P_y} = \frac{T_y}{T_x}$

CPU time equation

$$T_{CPU}(\text{execution}) = \frac{\#\,\text{instructions}}{\text{execution}} \cdot \frac{\#\,\text{cycles}}{\text{instruction}} \cdot \frac{\#\,\text{seconds}}{\text{cycle}}$$

Amdahl's law

$$T_{new} = \frac{\text{fraction effected} \cdot T_{old}}{\text{improvement}} + \text{fraction not effected} \cdot T_{old}$$

Processor Performance Equations

$$T_X = \#\,\text{instructions}_X \cdot \text{CPI}_X \cdot \text{cycleTime}_X$$

$$T_X = \frac{\#\,\text{instructions}_X \cdot \text{CPI}_X}{\text{clockRate}_X}$$

$$\frac{P_X}{P_Y} = \frac{T_Y}{T_X} = \frac{\#\,\text{instructions}_Y \cdot \text{CPI}_Y \cdot \text{cycleTime}_Y}{\#\,\text{instructions}_X \cdot \text{CPI}_X \cdot \text{cycleTime}_X}$$

$$\frac{P_X}{P_Y} = \frac{T_Y}{T_X} = \frac{\#\,\text{instructions}_Y \cdot \text{CPI}_Y \cdot \text{clockRate}_X}{\#\,\text{instructions}_X \cdot \text{CPI}_X \cdot \text{clockRate}_Y}$$

HW 13: Processor Performance

1) Processor A runs program P in 30 seconds executing $12 \cdot 10^9$ instructions. Half of these are ALU instructions, a quarter are data transfer (load or store) instructions, and a quarter are branch instructions. Execution of the ALU and branch instructions require 4 cycles each. Execution of load and store instructions require 6 cycles each. What is the clock speed of processor A in GHz (billions of cycles per second)?

2) On processor B every instruction takes 3 cycles to execute. The clock for processor B is twice as fast as processor A's clock. How long does it take processor B to execute program P?

3) What is the relative performance of processor B to processor A?

HW 13: Processor Performance

1) Processor A runs program P in 30 seconds executing $12 \cdot 10^9$ instructions. Half of these are ALU instructions, a quarter are data transfer (load or store) instructions, and a quarter are branch instructions. Execution of the ALU and branch instructions require 4 cycles each. Execution of load and store instructions require 6 cycles each. What is the clock speed of processor A in GHz (billions of cycles per second)?

$$T_X = \frac{\#\,\text{instructions}_X \cdot \text{CPI}_X}{\text{clockRate}_X}$$

$$30 = \frac{12 \cdot 10^9 \cdot (0.5 \cdot 4 + 0.25 \cdot 6 + 0.25 \cdot 4)}{X \text{ Hz}} = \frac{12 \cdot 4.5}{X \text{ Hz}} \cdot 10^9$$

$$X \text{ Hz} = \frac{54}{30} \cdot 10^9 \text{ Hz} = 1.8 \cdot 10^9 \text{ Hz} = 1.8 \text{ GHz}$$

HW 13: Processor Performance

2) On processor B every instruction takes 3 cycles to execute. The clock for processor B is twice as fast as processor A's clock. How long does it take processor B to execute program P?

$$\frac{T_B}{T_A} = \frac{\#\,\text{instructions}_B \cdot CPI_B \cdot \text{clockRate}_A}{\#\,\text{instructions}_A \cdot CPI_A \cdot \text{clockRate}_B}$$

$$\frac{T_B}{30} = \frac{\cancel{12 \cdot 10^9} \cdot 3 \cdot \cancel{\text{clockRate}_A}}{\cancel{12 \cdot 10^9} \cdot (0.5 \cdot 4 + 0.25 \cdot 6 + 0.25 \cdot 4) \cdot 2 \cdot \cancel{\text{clockRate}_A}}$$

$$T_B = 30 \cdot \frac{3}{4.5 \cdot 2} = \frac{90}{9} = \mathbf{10}$$

3) What is the relative performance of processor B to processor A?

$$\frac{P_B}{P_A} = \frac{T_A}{T_B} = \frac{30}{10} = \mathbf{3}$$

B is 3 times as fast as A

Eras in Processor Architecture Evolution

Multi-cycle instructions — **CPI > 1**

- Antiquated (ex. simple **TOY** processor)

Single-cycle instructions — **CPI = 1**

- Long cycles

Pipelined instructions[a] — **CPI = 1 + σ**

- Short cycles
- σ is **S**tall **C**ycles **P**er **I**nstruction (SCPI)

Multiple issue instructions — **CPI < 1**

- **V**ery **L**ong **I**nstruction **W**ord (VLIW)
- Superscalar

[a] assumes CPI for the perfect pipeline = 1

Performance Problems

Conversions between “eras”

- Multi-cycle to single-cycle instruction execution
- Single-cycle to pipeline instruction execution
  - Cycle time: sum(stage times) to max(stage times)+Δ
    - Δ is “pipeline overhead” (e.g. interstage register access)
- Pipeline to pipeline
- Multiple issue to pipeline

CPU time = #instructions · CPI · cycle time

- Actual pipeline CPI = CPI_{pp} + **SCPI**

Stall **C**ycles **P**er **I**nstruction

- Sum over all types of hazard of
  - Frequency of occurrence · # cycles per occurrence

HW 14 Pipeline Performance

This problem considers the performance of several implementations of the same instruction set architecture on a suite of programs with the following characteristics:

ALU instructions	45%	
Load instructions	20%	
Store instructions	10%	
Branch instructions	25%	
Unconditional	5%	
Conditional	20%	
Backward	16%	(taken 6%, not taken 10%)
Forward	4%	(taken 2%, not taken 2%)

HW 14 part A

A multiple-cycle serial implementation has the following cycle counts for the various instructions:

ALU instructions	3 cycles
Load instructions	8 cycles
Store instructions	6 cycles
Unconditional branches	3 cycles
Taken conditional branches	5 cycles
Not-taken conditional branches	4 cycles

What is the CPI of the program suite on this implementation?

$$\begin{aligned} \text{CPI} &= 0.45 \cdot 3 + 0.20 \cdot 8 + 0.10 \cdot 6 + 0.05 \cdot 3 + (0.06 + 0.02) \cdot 5 + (0.10 + 0.02) \cdot 4 \\ &= 1.35 + 1.6 + 0.6 + 0.15 + 0.4 + 0.48 \\ &= 4.58 \end{aligned}$$

HW 14 part B

A second serial implementation executes every instruction in a single cycle. Instruction execution is performed in the following phases having the indicated execution times:

IF	instruction fetch	150 ps
RR	register read	100 ps
EX	ALU operation	400 ps
MW	memory write	300 ps
MR	memory read	150 ps
RW	register write	100 ps

How long is the cycle time for this implementation?

$$150 + 100 + 400 + 300 + 150 + 100 = 1200\ ps$$

HW 14 part C

A third implementation executes instructions in a pipeline using the stages given in part B.

How long is the cycle time for this implementation?

$$max(150, 100, 400, 300, 150, 100) = 400\ ps$$

What is the relative performance of this implementation to that of part B?

$$\frac{P_C}{P_B} = \frac{T_B}{T_C} = \frac{cycleTime_B}{cycleTime_C} = \frac{1200}{400} = 3.0$$

HW 14 part D

A final implementation is also a pipeline of the stages given in part B except that the execute phase, EX, is partitioned into two phases, EX1 and EX2 each requiring 200 ps.

How long is the cycle time for this implementation?

$$max(150, 100, 200, 200, 300, 150, 100) = 300\ ps$$

What is the relative performance of this implementation to that of part C?

$$\frac{P_D}{P_C} = \frac{T_C}{T_D} = \frac{cycleTime_C}{cycleTime_D} = \frac{400}{300} = 1.333\ldots \approx 1.33$$

Actual Pipeline CPI

Perfect pipeline (pp) ignores hazards

- If each pipeline stage takes 1 cycle **$CPI_{pp} = 1$**

Actual pipeline (ap) performance

- Actual pipeline sometimes *stalls* due to hazards
- A stall is a cycle without useful work

$$\mathbf{CPI_{ap} = CPI_{pp} + SCPI_{ap}}$$

$SCPI_{ap}$ is **S**tall **C**ycles **P**er **I**nstruction

First calculate CPI_{pp} ignoring stalls

Then add in the average number of stalled cycles

Memory Access Cost

Pipeline cycles per instruction:

$$\mathrm{CPI}_{\text{actual}} = \mathrm{CPI}_{\text{perfect pipeline}} + \sum_{h \in \{\text{Hazards}\}} \mathrm{SCPI}_h$$

Stall cycles due to *memory access* hazards

$$\mathrm{SCPI}_{\text{ma}} = \mathrm{MAPI} \cdot \mathrm{SC}_{\text{ma}}$$

MAPI – memory accesses per instruction

SC_{ma} – stall cycles per memory access

memory access cost (in cycles) **– 1**

Memory **A**ccess **C**ost

$$\mathbf{MAC} = \mathbf{cost}_{\mathbf{hit}} + \mathbf{miss}_{\mathbf{rate}} \cdot \mathbf{miss}_{\mathbf{penalty}}$$

$$\mathbf{hit}_{\mathbf{rate}} = 1 - \mathbf{miss}_{\mathbf{rate}}$$

Cost due to Memory Hazards

$$\mathbf{SCPI}_{\mathbf{ma}} = \mathbf{MAPI} \cdot \mathbf{SC}_{\mathbf{ma}}$$

MAPI **M**emory **A**ccess **P**er **I**nstruction

(*usually*, ~ 1.3)

Instruction fetch (***always***, 1 per instruction)

Data **l**oad (*usually*, ~ 0.2 per instruction)

Data **st**ore (*usually*, ~ 0.1 per instruction)

$\mathbf{SC}_{\mathbf{ma}}$ **S**tall **C**ycles per **m**emory **a**ccess

$$\mathtt{memory\ access\ cost} - 1$$

$$\mathtt{cost}_{\mathtt{hit}} + \mathtt{miss}_{\mathtt{rate}} \cdot \mathtt{cost}_{\mathtt{miss}} - 1$$

Example Instruction Distribution

50%	ALU	700 ps	(IF, ID, EX, WB)
20%	Load	1000 ps	(IF, ID, EX, MEM, WB)
05%	Store	850 ps	(IF, ID, EX, MEM)
05%	uncond	550 ps	(IF, ID, EX)
20%	cond	700 ps	(IF, ID, EX, M1)

Conditional branches

14%	backwards	(**taken**)
01%	backwards	(**not taken**)
02%	forwards	(**taken**)
03%	forwards	(**not taken**)

Stall **C**ycles **P**er **I**nstruction

1 in 20 *loads* causes load-use data hazard

Frequency:

stall cycles:

SCPI:

2 cycle stall per ***taken*** *conditional branch*

Frequency:

stall cycles:

SCPI:

1 in 100 *memory accesses* takes 17 cycles

Frequency:

stall cycles:

SCPI:

Total SCPI:

Stall **C**ycles **P**er **I**nstruction

1 in 20 *loads* causes load-use data hazard

Frequency: 0.01 = 0.2 · 0.05

stall cycles: 1

SCPI: 0.01 = 0.01 · 1

2 cycle stall per ***taken*** *conditional branch*

Frequency: 0.16 = 0.14 + 0.02

stall cycles: 2

SCPI: 0.32 = 0.16 · 2

1 in 100 *memory accesses* takes 17 cycles

Frequency: 0.0125 = (1 + 0.2 + 0.05) · 0.01

stall cycles: 16 = 17 – 1 (last cycle is ***not*** a stall)

SCPI: 0.20 = 0.0125 · 16

Total SCPI: 0.53 = 0.01 + 0.32 + 0.20

Example SCPI Problem

This assignment concerns the cost of hazards to a pipeline processor with a perfect pipeline CPI of 1.0. Compute the Stall Cycles Per Instruction (SCPI) for each hazard on a program with the following instruction distribution:

50%	ALU	
20%	Load	
05%	Store	
05%	Unconditional branches	
20%	Conditional branches	(fraction)
	Backward–taken	0.75
	Backward–not–taken	0.10
	Forward–taken	0.05
	Forward–not–taken	0.10

Example SCPI Hazards

A) **Def-use data hazard**: One out of evert 40 ALU instructions is delayed by 3 cycles.

B) **Cache miss hazard**: One out of every 200 memory accesses takes 25 cycles.

C) **Conditional branch hazard**: Every taken conditional branch is delayed 1 cycle.

D) **Exception hazard**: 5 of every million instruction executions encounters an exception that requires 15 thousand cycles to handle.

Finally, what is the CPI of the actual pipeline that experiences each of the above hazards?

$$\mathrm{CPI}_{\mathbf{ap}} = \mathrm{CPI}_{\mathbf{pp}} + \sum_{h \in \{\text{Hazards}\}} \mathrm{f}_h \cdot \mathrm{c}_h$$

A) **Def-use data hazard**: One out of evert 40 ALU instructions is delayed by 3 cycles.

$$\begin{aligned}
\mathrm{SCPI}_{\mathrm{A}} &= \mathrm{f}_{\mathrm{A}} \cdot \mathrm{c}_{\mathrm{A}} \\
&= 0.5 \cdot \frac{1}{40} \cdot 3 \\
&= \frac{1}{80} \cdot 3 \\
&= \frac{3}{80} \\
&= 0.0375 \\
&\approx 0.038
\end{aligned}$$

$$\mathrm{CPI}_{\mathbf{ap}} = \mathrm{CPI}_{\mathbf{pp}} + \sum_{h \in \{\text{Hazards}\}} \mathrm{f}_h \cdot \mathrm{c}_h$$

B) **Cache miss hazard**: One out of every 200 memory accesses takes 25 cycles.

$$\begin{aligned}
\mathrm{SCPI}_{\mathrm{B}} &= \mathrm{f}_{\mathrm{B}} \cdot \mathrm{c}_{\mathrm{B}} \\
&= MAPI \cdot \frac{1}{200} \cdot (25 - 1) \\
&= \frac{1 + 0.2 + 0.5}{200} \cdot 24 \\
&= \frac{1.25}{25} \cdot 3 \\
&= 0.05 \cdot 3 \\
&= 0.15
\end{aligned}$$

$$\mathrm{CPI}_{\mathbf{ap}} = \mathrm{CPI}_{\mathbf{pp}} + \sum_{h \in \{\mathrm{Hazards}\}} \mathbf{f}_h \cdot \mathbf{c}_h$$

C) **Conditional branch hazard**: Every taken conditional branch is delayed 1 cycle.

$$\begin{aligned} \mathrm{SCPI}_{\mathrm{C}} &= \mathbf{f}_{\mathrm{C}} \cdot \mathbf{c}_{\mathrm{C}} \\ &= 0.2 \cdot (0.75 + 0.05) \cdot 1 \\ &= 0.2 \cdot 0.8 \cdot 1 \\ &= 0.16 \end{aligned}$$

$$\mathrm{CPI}_{\mathbf{ap}} = \mathrm{CPI}_{\mathbf{pp}} + \sum_{h \in \{\mathrm{Hazards}\}} \mathbf{f}_h \cdot \mathbf{c}_h$$

D) **Exception hazard**: 5 of every million instruction executions encounters an exception that requires 15 thousand cycles to handle.

$$\begin{aligned} \mathrm{SCPI}_D &= \mathbf{f}_D \cdot \mathbf{c}_D \\ &= \frac{5}{1000000} \cdot 15000 \\ &= \frac{5 \cdot 15}{1000} \\ &= 0.075 \end{aligned}$$

$$\mathrm{CPI}_{\mathbf{ap}} = \mathrm{CPI}_{\mathbf{pp}} + \sum_{h \in \{\mathrm{Hazards}\}} \mathbf{f}_h \cdot \mathbf{c}_h$$

Finally, what is the CPI of the actual pipeline that experiences each of the above hazards?

$$\begin{aligned}
\mathrm{CPI}_{\mathbf{ap}} &= \mathrm{CPI}_{\mathbf{pp}} + \sum_{h \in \{\mathrm{Hazards}\}} \mathrm{SCPI}_h \\
&= 1 + \mathrm{SCPI}_{\mathrm{A}} + \mathrm{SCPI}_{\mathrm{B}} + \mathrm{SCPI}_{\mathrm{C}} + \mathrm{SCPI}_{\mathrm{D}} \\
&= 1 + 0.0375 + 0.15 + 0.16 + 0.075 \\
&= 1.4225 \\
&\approx 1.42
\end{aligned}$$