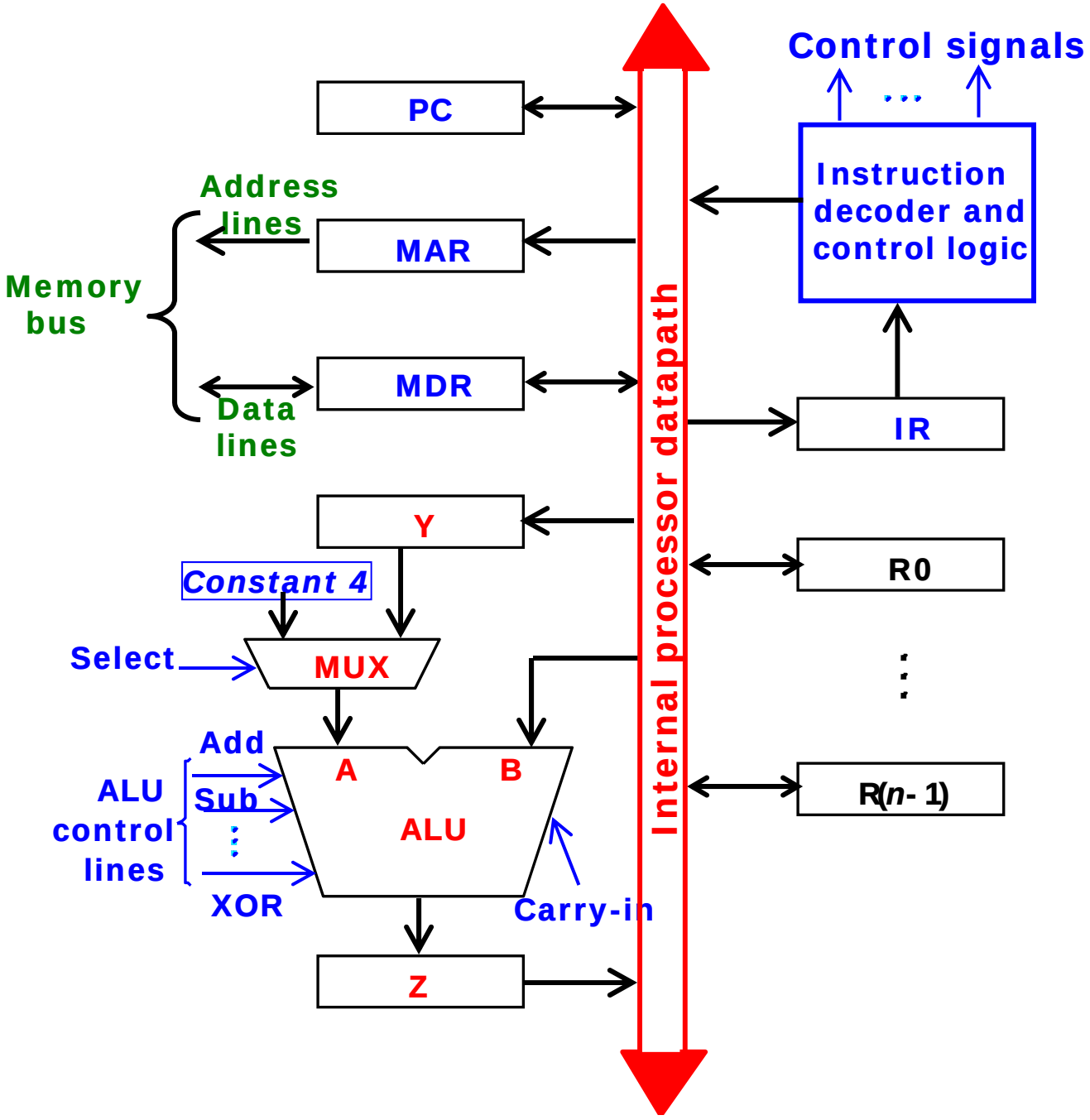
07 CPU Organization and Datapath V.1

CSC 230

M&H: 4.1; 5.1, 5.2 (translated from ARC)
Stallings: 2; 3.1; 3.2 (skip interrupts and I/O; 14.1 to 14.3

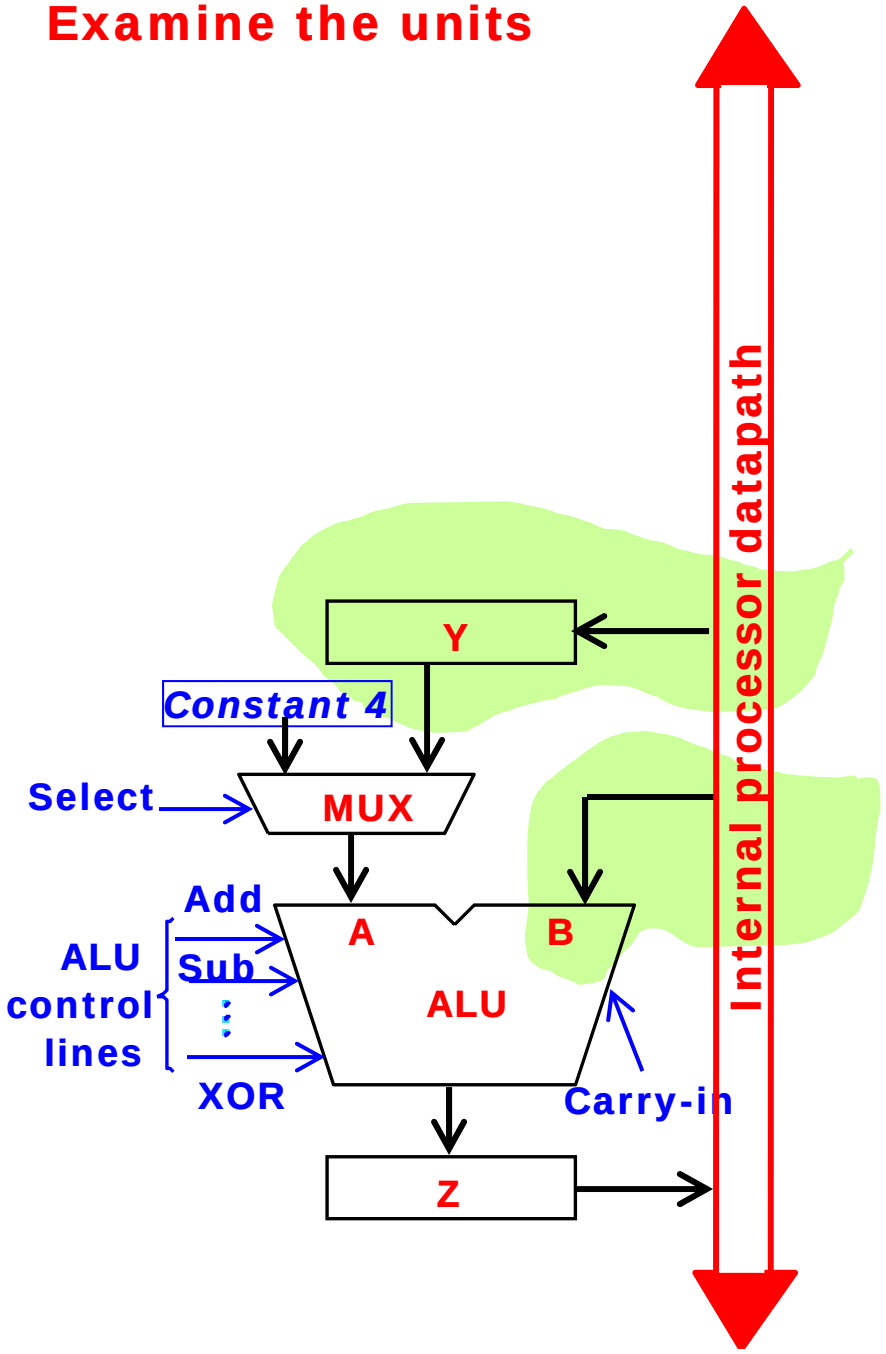
Refresher: Detailed CPU activity

1. **fetch instruction from memory**
   - **address in PC → MAR**
   - **MAR → Address Bus**
   - **read signal → Control Bus**
   - **Wait for memory**
   - **content of location in memory → Data Bus**
   - **Data Bus → MDR**
   - **MDR → IR**
2. **update current memory address**
   - **increment PC (ready for next instruction)**
3. **decode instruction**
   - **decode phase looks at opcode and operands**
4. **fetch further data, if required**
   - **(MAR, MDR and PC are possibly in use)**
5. **execute instruction**
   - **(may involve ALU, reads and writes to memory)**
6. **go to step (1)**



Learn how a datapath works by examining in details the Fetch – Decode – Execute Cycle for a general instruction (not ARM or RISC)

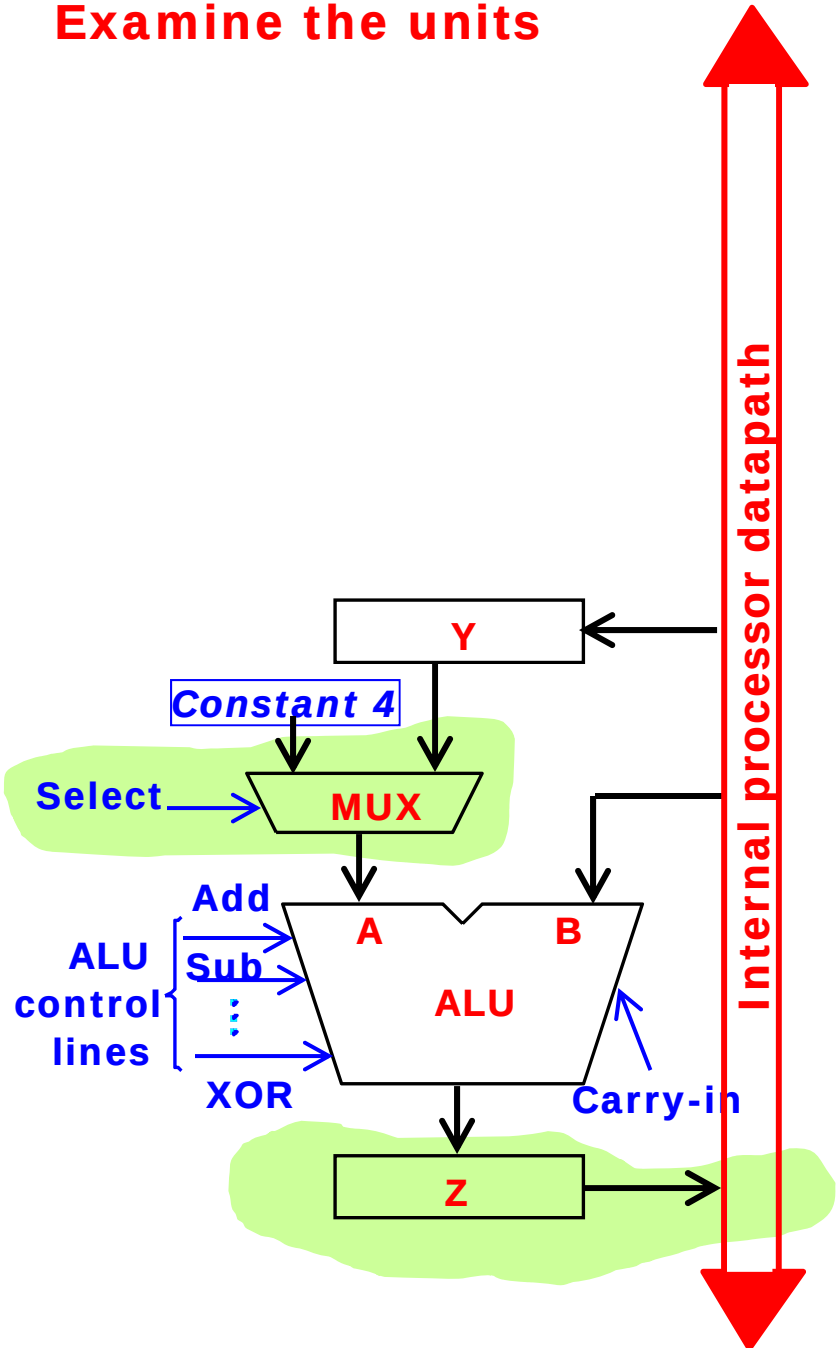
Examine the units



1. ALU

- combinational device, i.e. it has no memory or internal storage.
- must have one input connection from the datapath.
- other input must be stored in a holding register: Y register.

Examine the units



2. MUX = multiplexer

- A multiplexer selects between register Y and a constant (here = 4) depending upon the Select line.

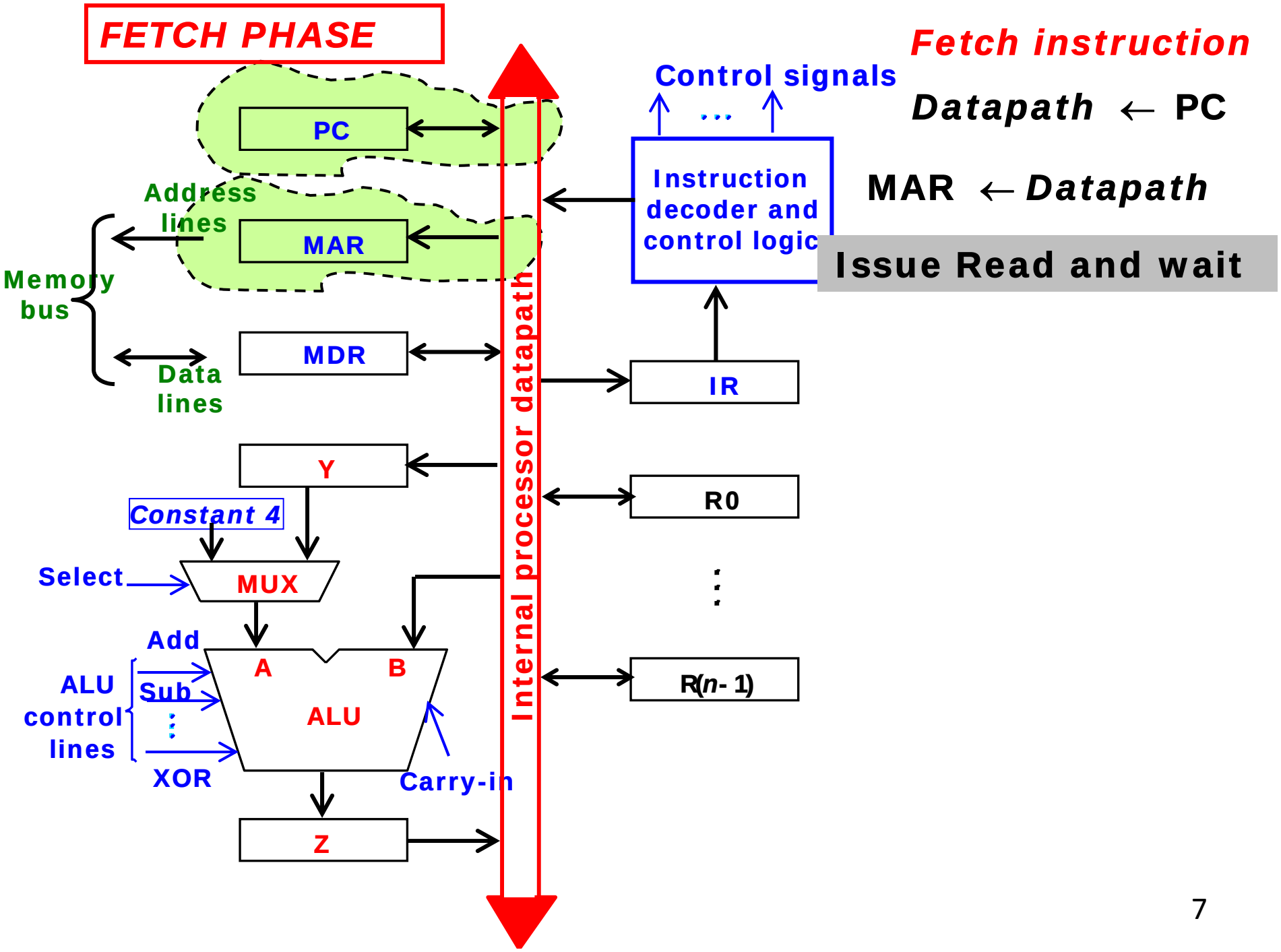
3. Register Z:

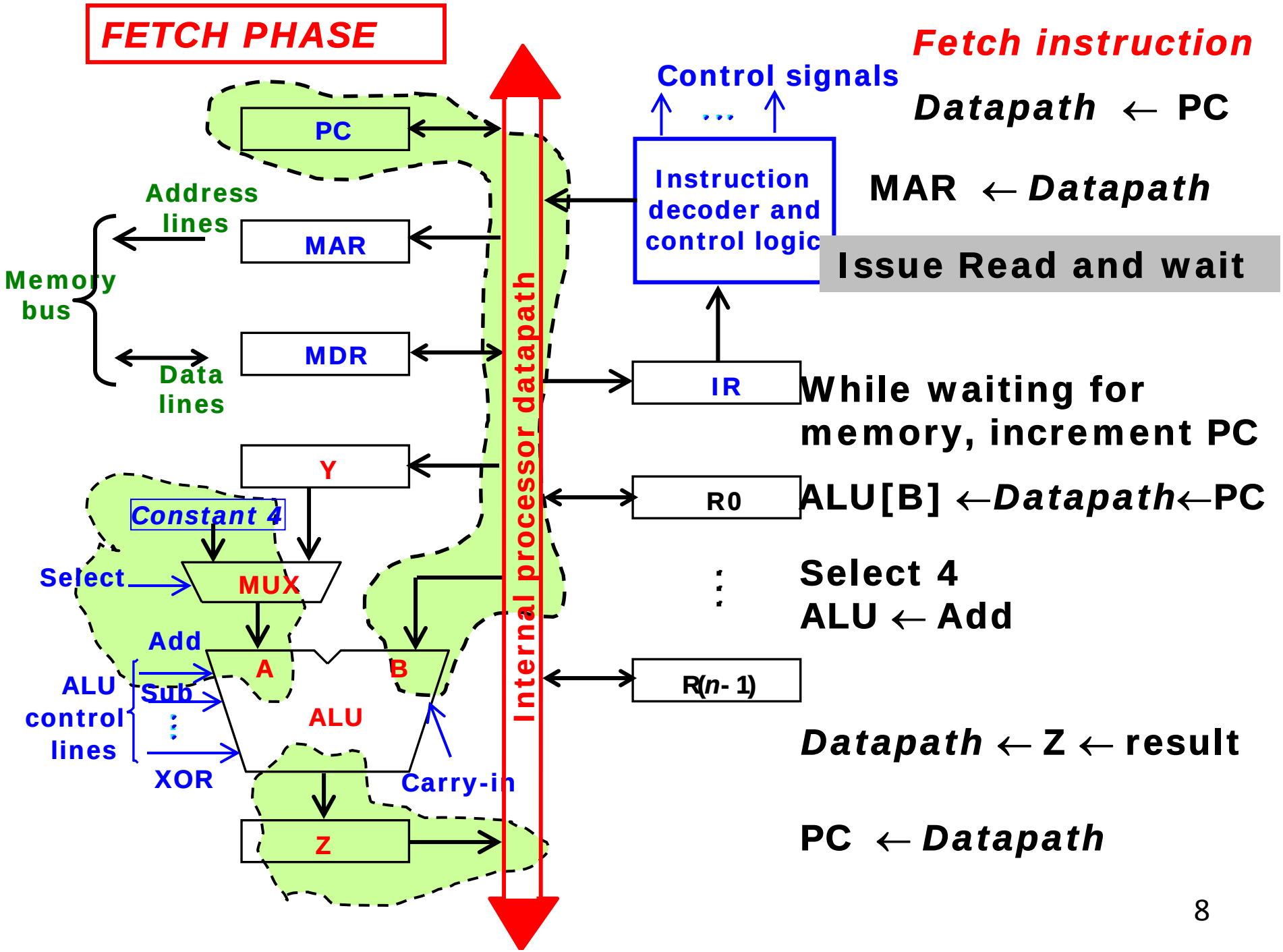
- The output register Z collects the result.
- Z is connected to the datapath.

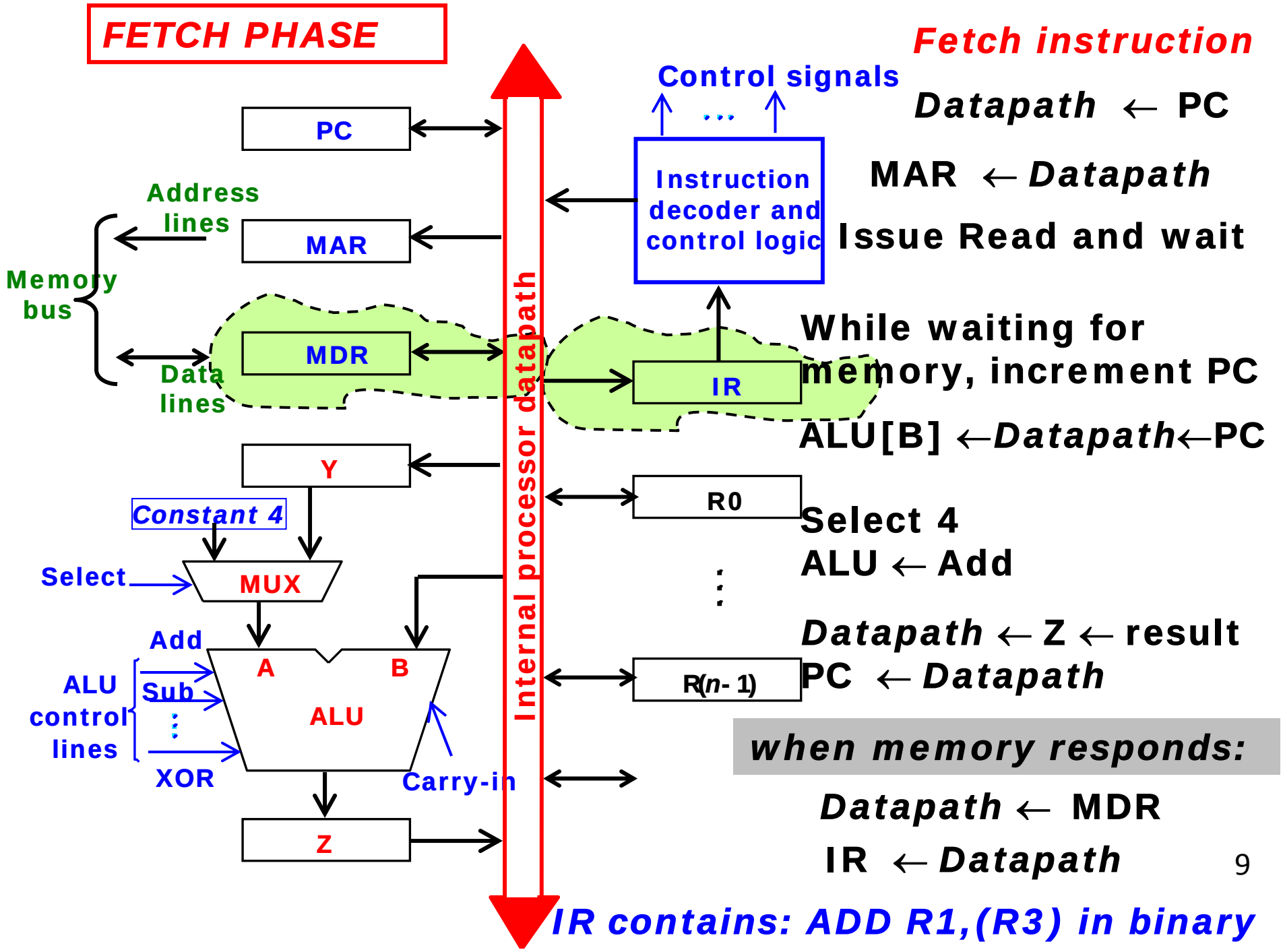
Example: ADD R1,[R3] *(not ARM)*

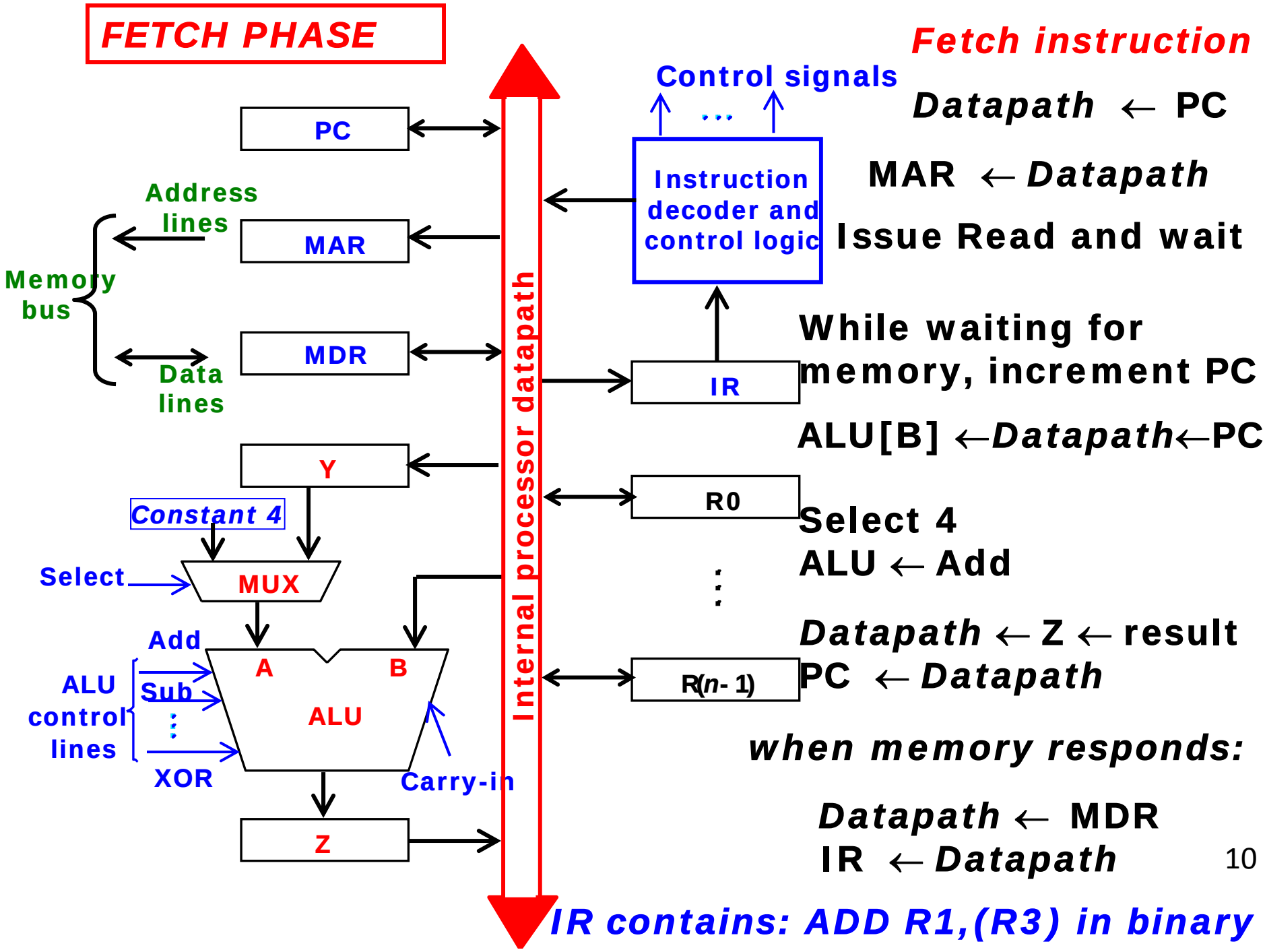
ADD R1,[R3] R1 ← R1 + Memory at [R3]

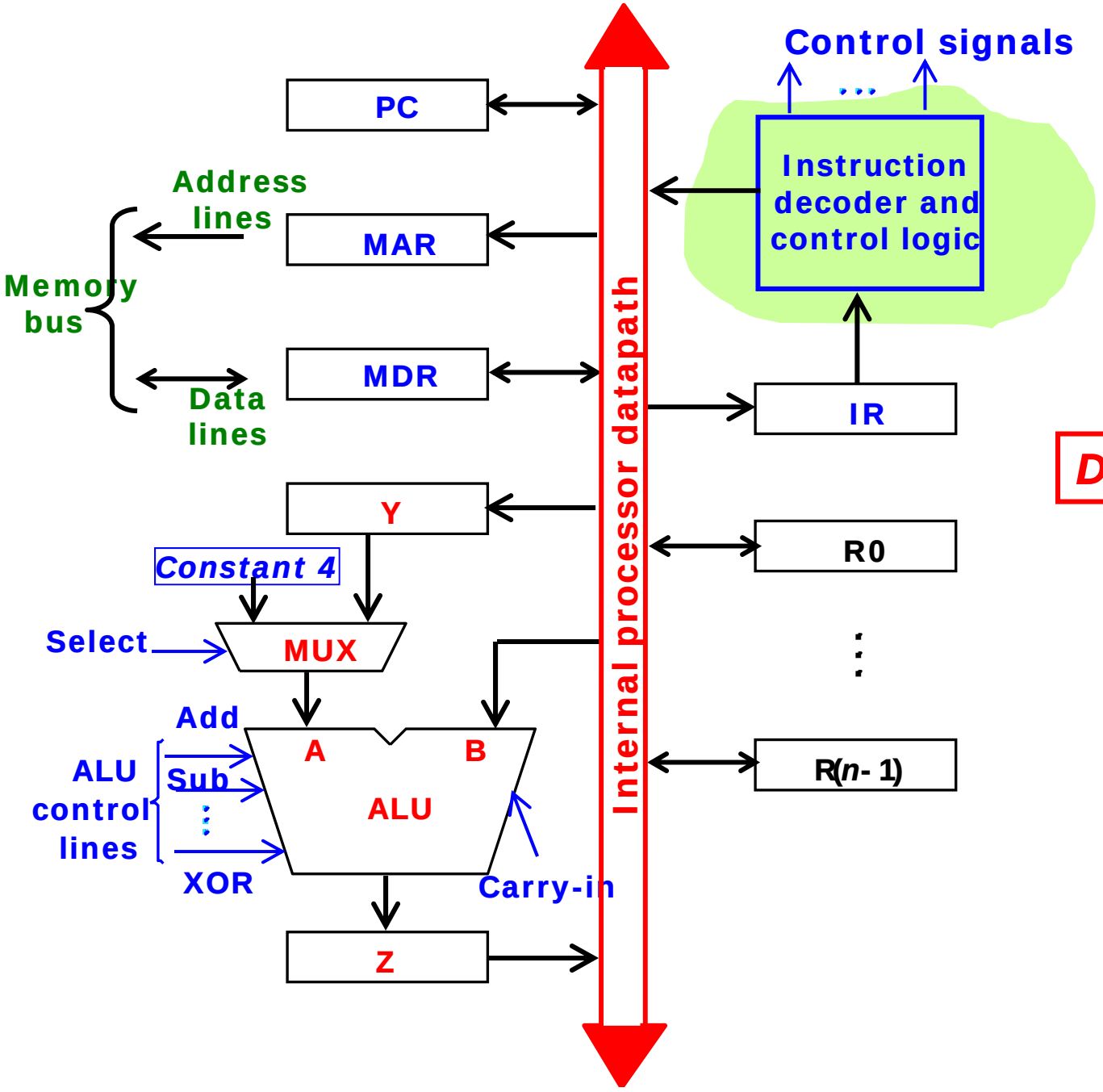
1. **Fetch instruction**
2. **Fetch memory operand**
   (content of memory at address in R3)
3. **Add**
4. **Result in R1**


FETCH PHASE
Fetch instruction
Control signals
Datapath ← PC
MAR ← Datapath
Issue Read and wait
PC
Address lines
MAR
Memory bus
MDR
Data lines
Instruction decoder and control logic
IR
Y
R0
Constant 4
Select
MUX
Add
A
B
ALU control lines
Sub
ALU
XOR
Carry-in
R(n- 1)
Z
Internal processor datapath


FETCH PHASE
Fetch instruction
Control signals
PC
Address lines
MAR
Memory bus
MDR
Data lines
Y
Constant 4
Select
MUX
Add
A
B
ALU control lines
Sub
ALU
XOR
Carry-in
Z
Internal processor datapath
Instruction decoder and control logic
IR
R0
R(n-1)
Datapath ← PC
MAR ← Datapath
Issue Read and wait
While waiting for memory, increment PC
ALU[B] ←Datapath←PC
Select 4
ALU ← Add
Datapath ← Z ← result
PC ← Datapath

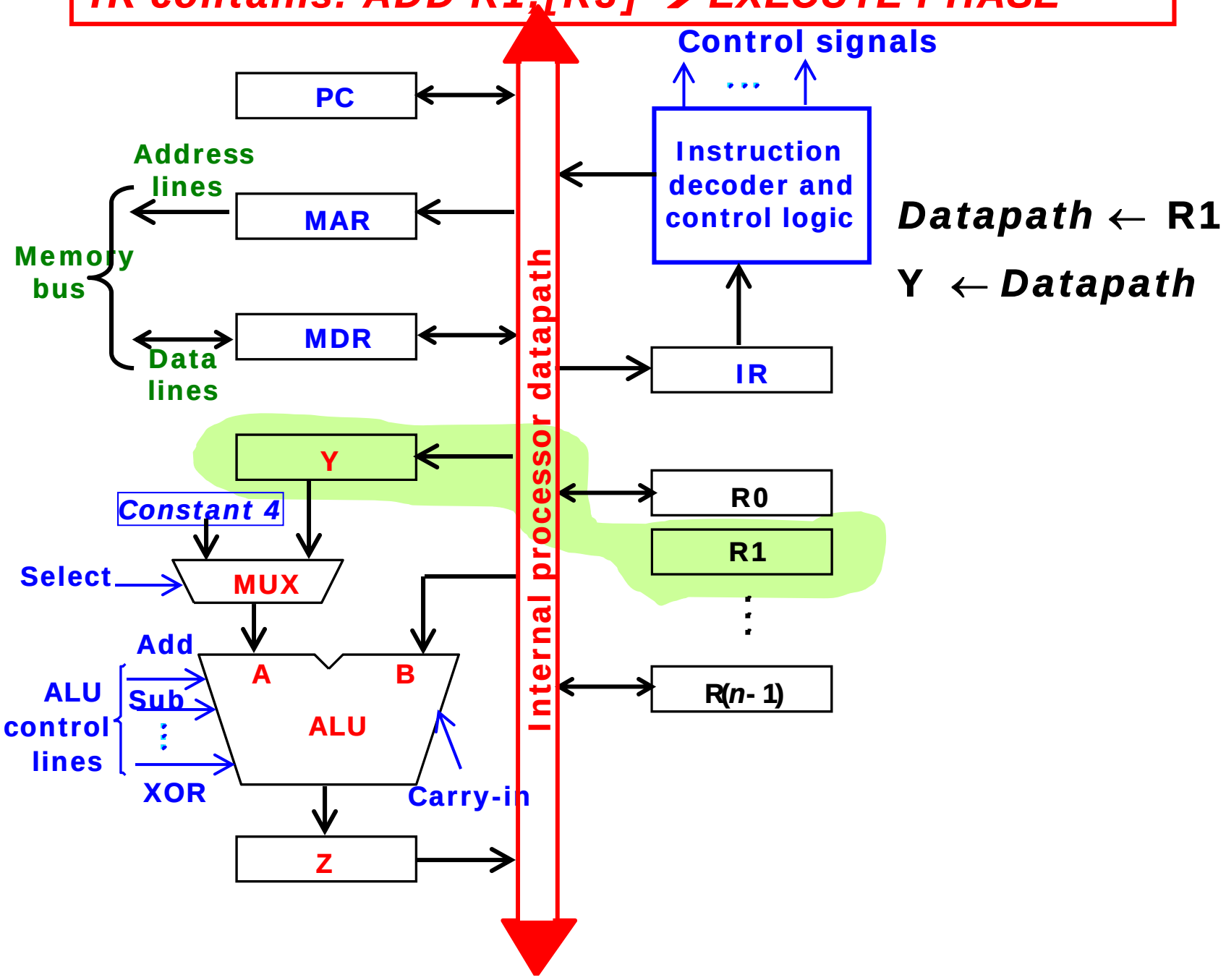

FETCH PHASE
Fetch instruction
Control signals
PC
Instruction decoder and control logic
Address lines
MAR
Memory bus
MDR
Data lines
IR
Internal processor datapath
Y
Constant 4
Select
MUX
Add
A
B
ALU control lines
Sub
ALU
XOR
Carry-in
Z
R0
R(n-1)
Datapath ← PC
MAR ← Datapath
Issue Read and wait
While waiting for memory, increment PC
ALU[B] ←Datapath←PC
Select 4
ALU ← Add
Datapath ← Z ← result
PC ← Datapath
when memory responds:
Datapath ← MDR
IR ← Datapath
IR contains: ADD R1,(R3) in binary


FETCH PHASE
Fetch instruction
Control signals
PC
Instruction decoder and control logic
Datapath ← PC
MAR ← Datapath
Issue Read and wait
Address lines
MAR
Memory bus
MDR
Data lines
While waiting for memory, increment PC
IR
ALU[B] ←Datapath←PC
Y
R0
Constant 4
Select 4
Select
MUX
ALU ← Add
Add
A
B
ALU control lines
Sub
ALU
Datapath ← Z ← result
PC ← Datapath
R(n- 1)
XOR
Carry-in
when memory responds:
Z
Datapath ← MDR
IR ← Datapath
Internal processor datapath
IR contains: ADD R1,(R3) in binary

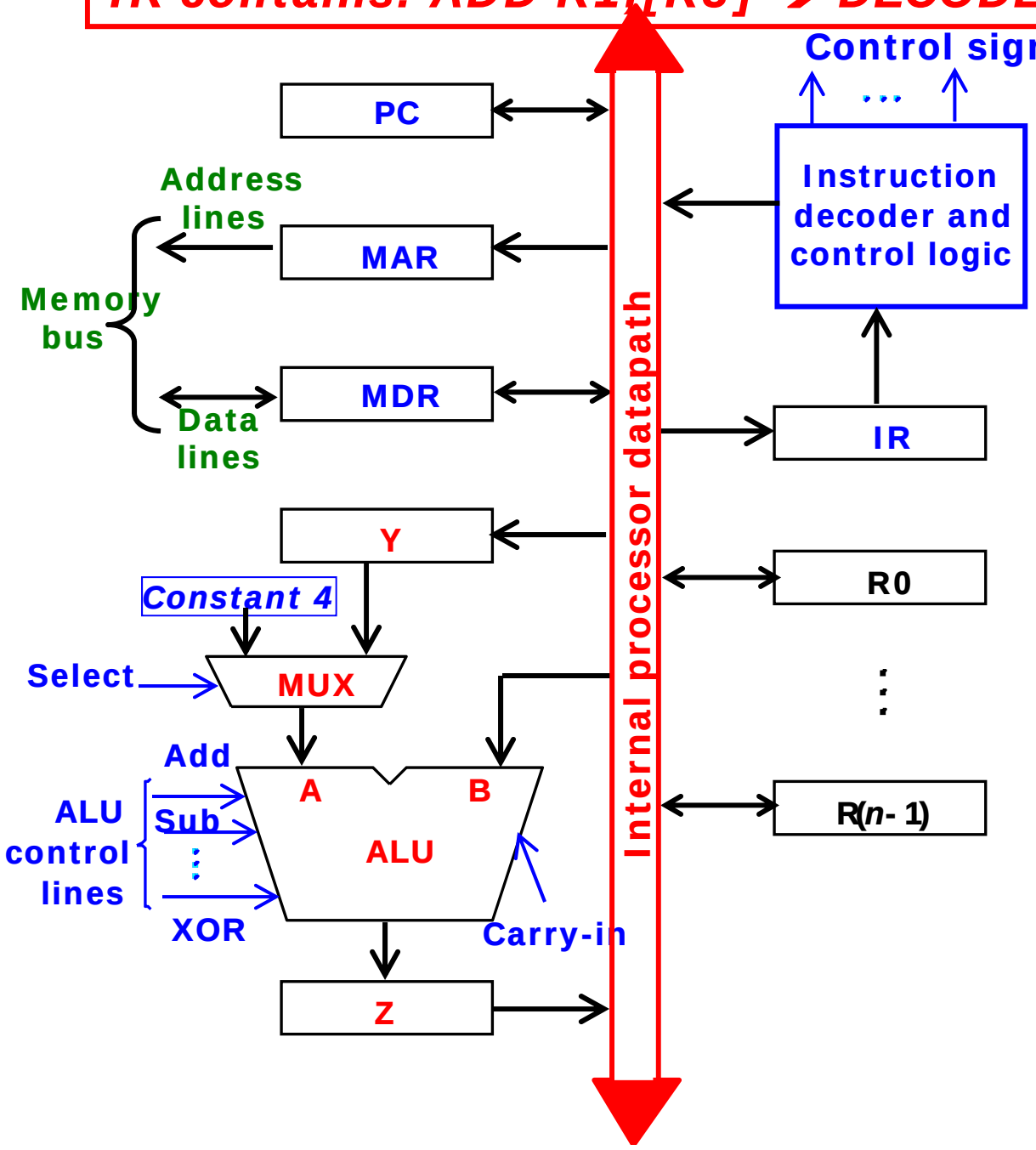


DECODE PHASE

IR contains: ADD R1, [R3] ➔ EXECUTE PHASE



IR contains: ADD R1, [R3] ➔ DECODE PHASE

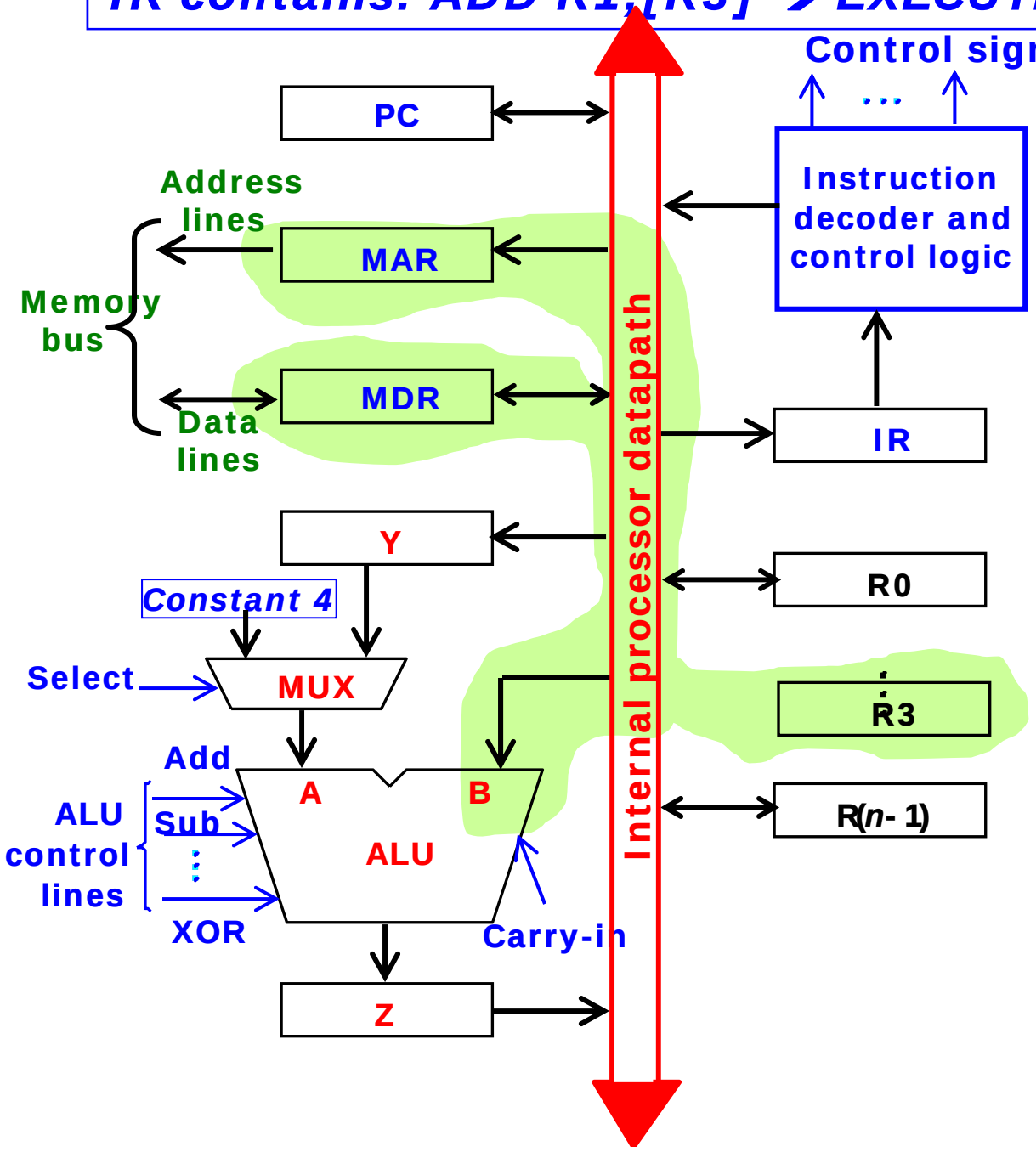


Get memory operand:

➔ Content of memory at address in R3

➔ need to do a READ

IR contains: ADD R1,[R3] ➔ EXECUTE PHASE



Datapath ← **R3**

MAR ← ***Datapath***

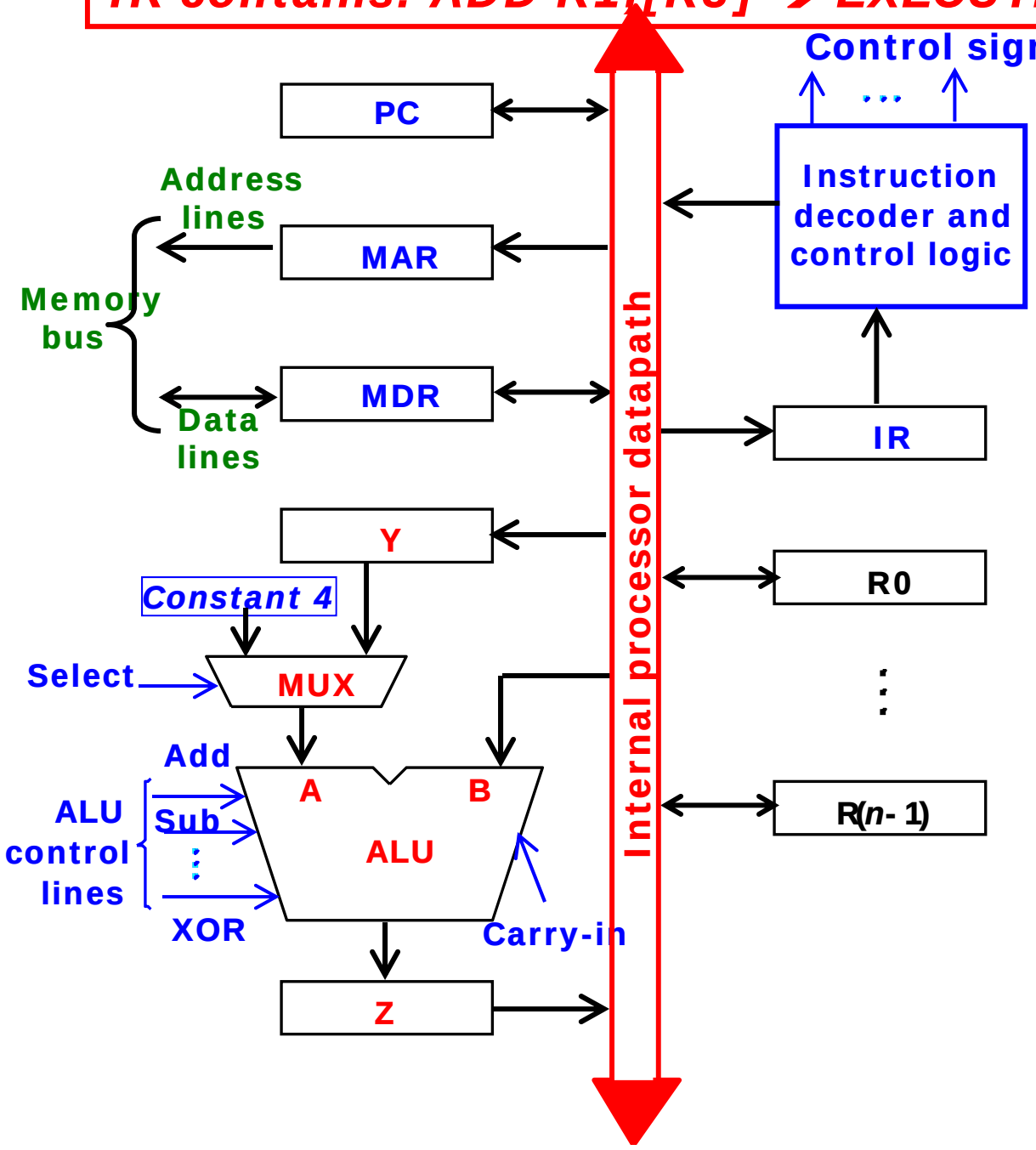
Read signal
Wait for memory

When memory answers:

Datapath ← **MDR**

ALU[B] ← ***Datapath***

IR contains: ADD R1, [R3] ➔ EXECUTE PHASE



Get memory operand:

➔ **Content of memory at address in R3**

➔ ***need to do a READ***

Datapath ← **R3**

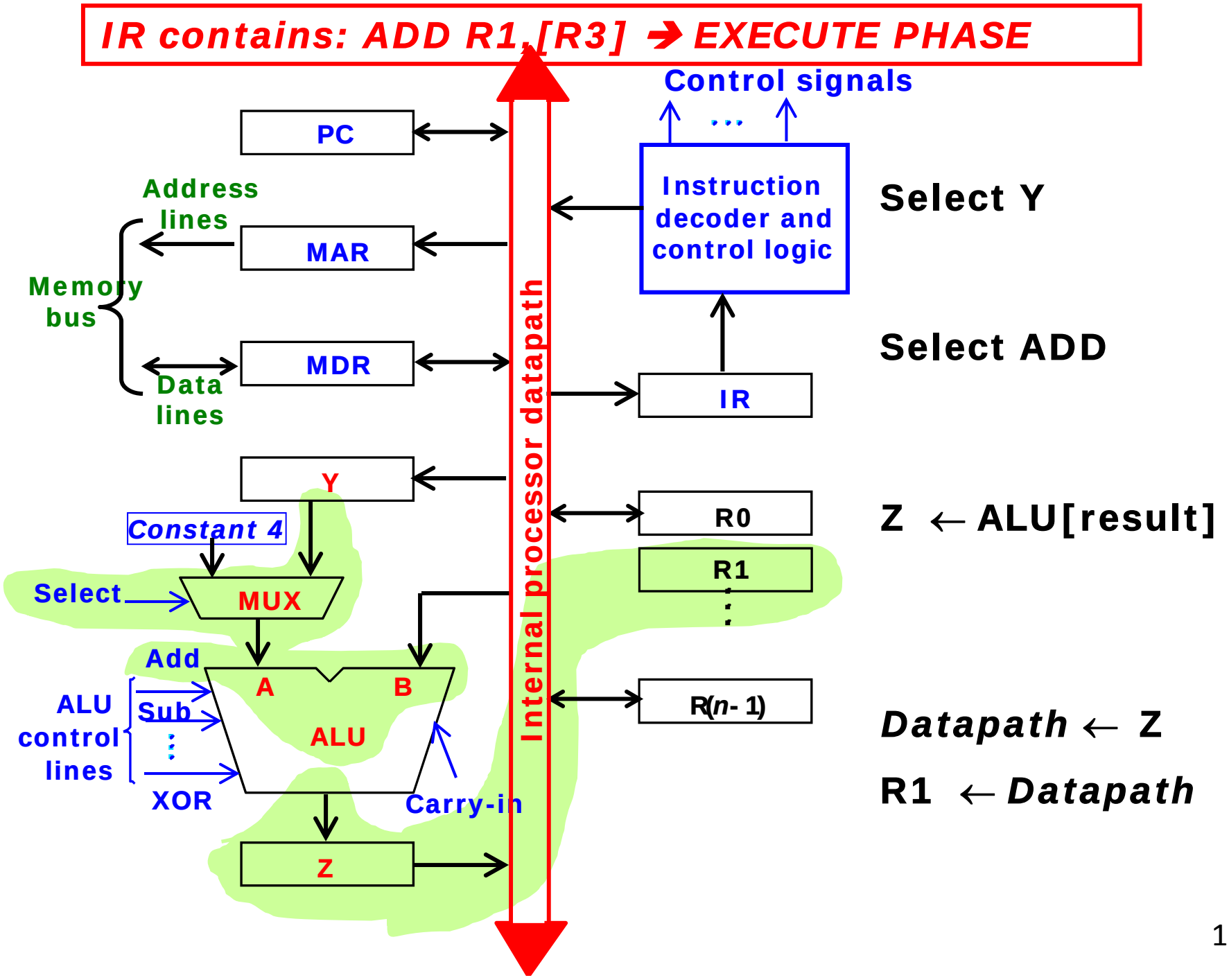
MAR ← ***Datapath***

Read and Wait for memory

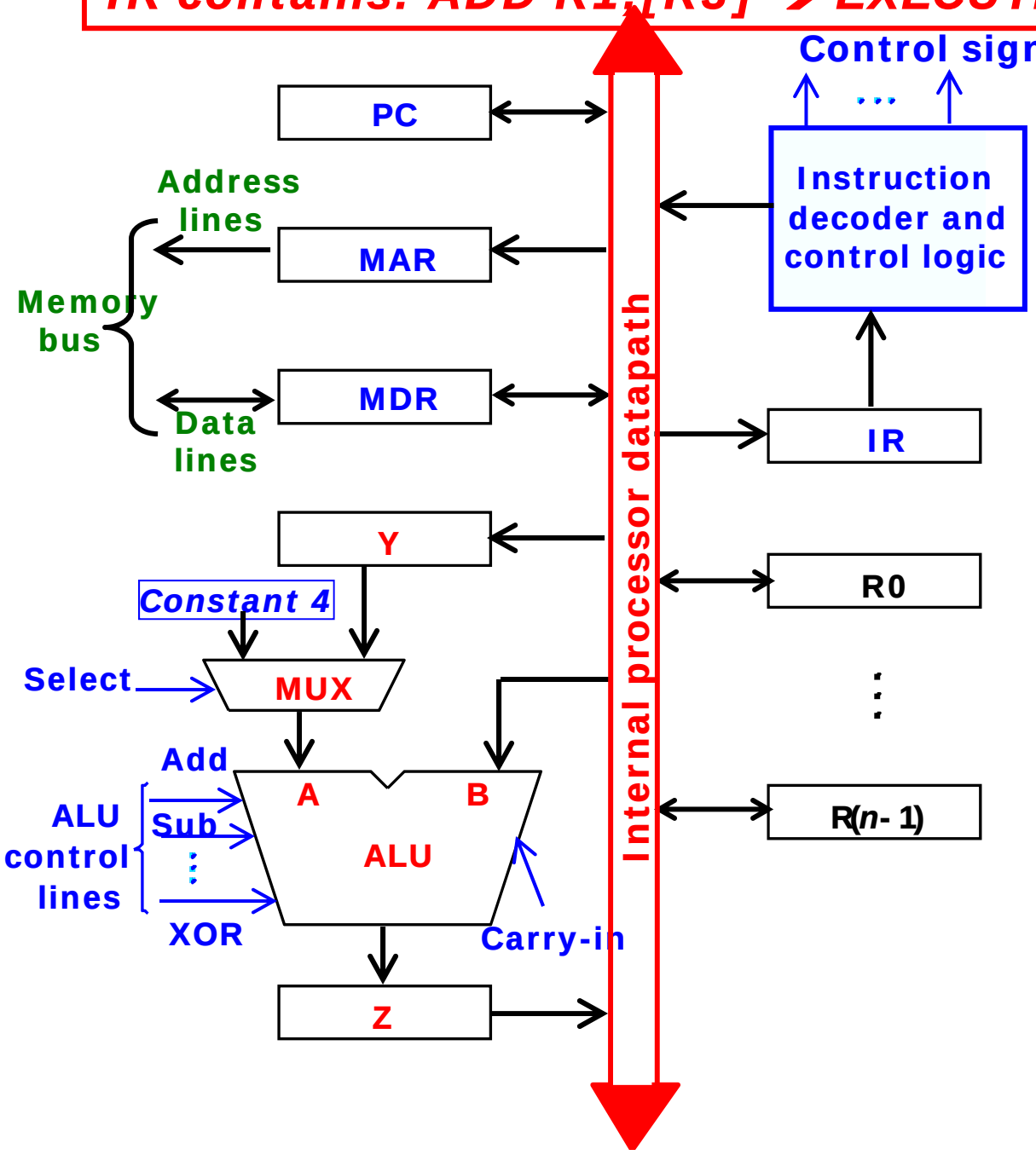
When memory answers:

Datapath ← **MDR**

ALU[B] ← ***Datapath***


IR contains: ADD R1, [R3] ➔ EXECUTE PHASE
Control signals
PC
Address lines
MAR
Memory bus
MDR
Data lines
Instruction decoder and control logic
IR
Y
Constant 4
Select
MUX
Add
A
B
ALU
ALU control lines
Sub
XOR
Carry-in
Z
Internal processor datapath
R0
R1
R(n- 1)
Select Y
Select ADD
Z ← ALU[result]
Datapath ← Z
R1 ← Datapath

IR contains: ADD R1, [R3] ➔ EXECUTE PHASE



Datapath ← R1

Y ← *Datapath*

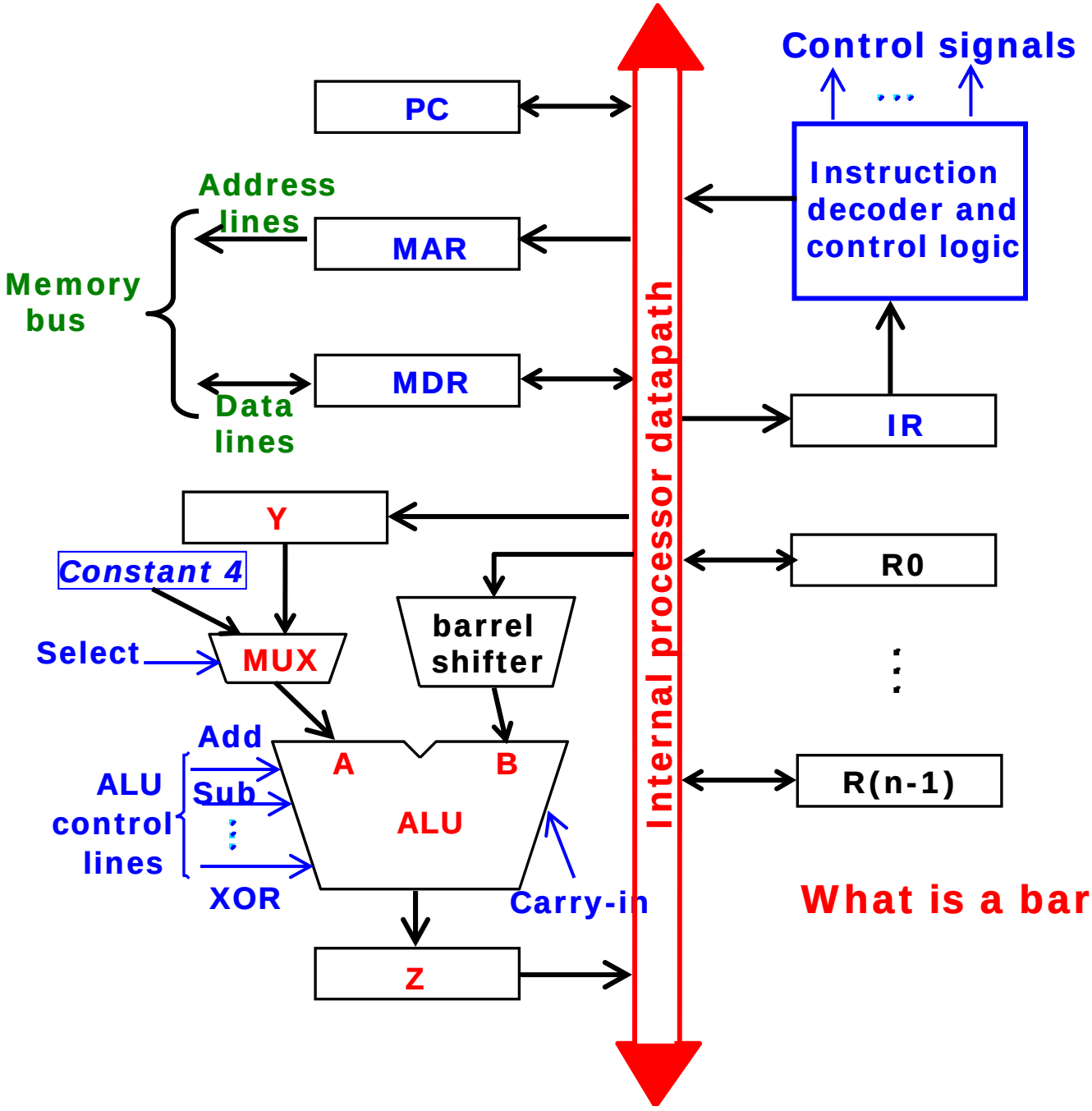
Select Y

Select ADD

Z ← ALU[result]

Datapath ← Z

R1 ← *Datapath*



What is a barrel shifter?