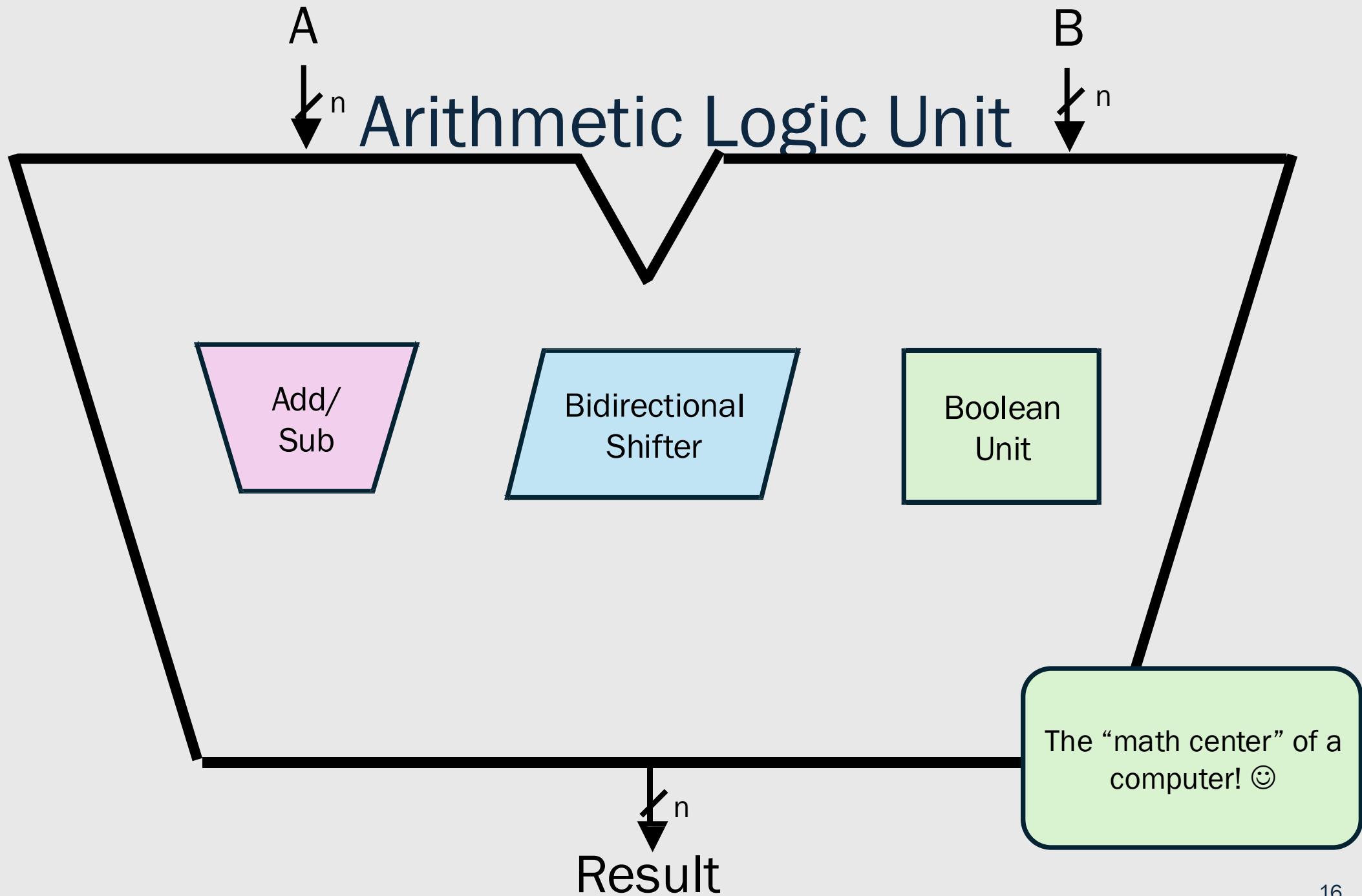


COMP311: *COMPUTER ORGANIZATION!*

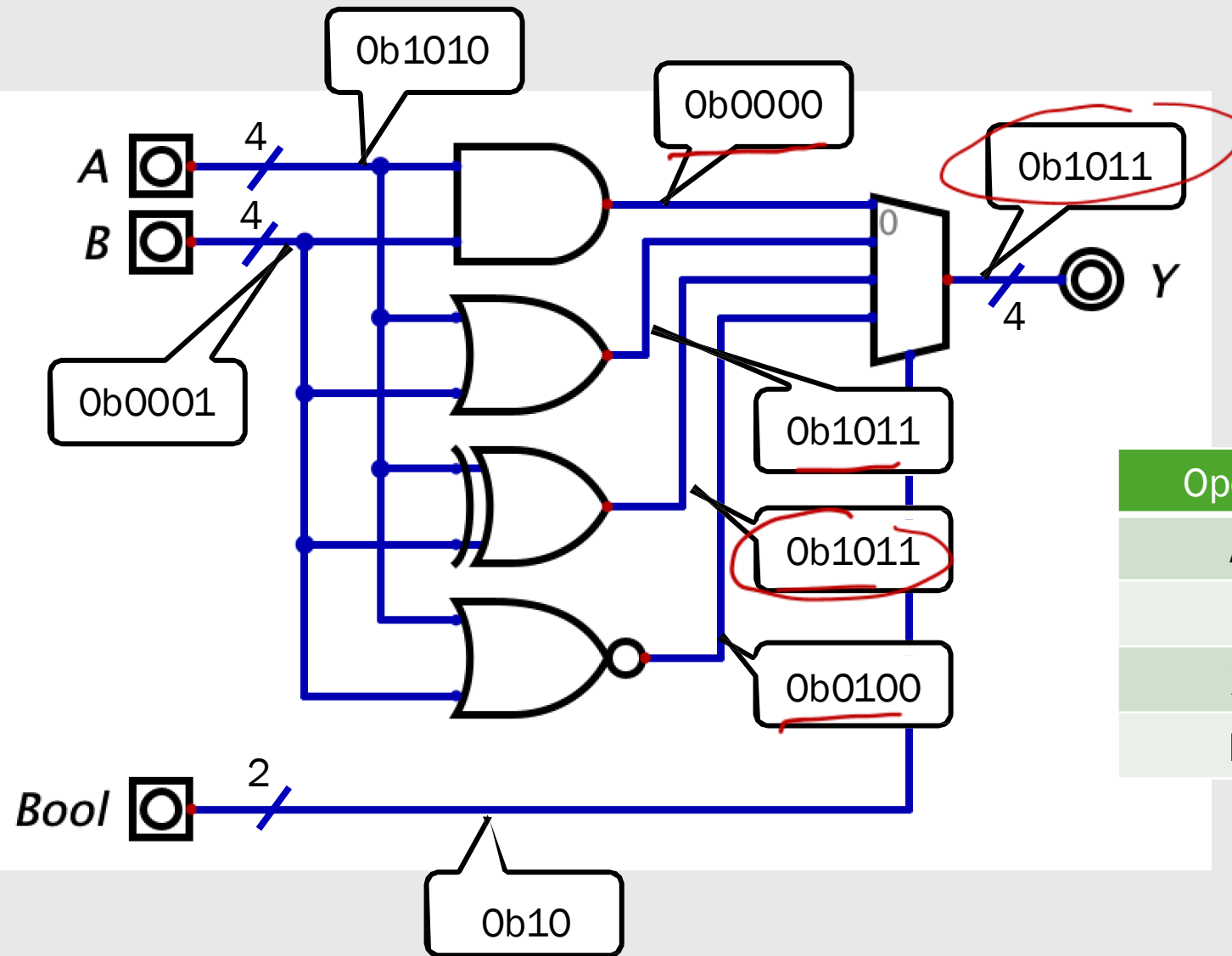
Lecture 12: The ALU & Register File

tinyurl.com/comp311-sp26

ARITHMETIC LOGIC UNIT (ALU) 😊



What are the values on each signal when we want to perform $Y = A \text{ XOR } B$ where A is $0b1010$ and B is $0b0001$?



Operation	Expression	Bool
AND	$A \times B$	$0b00$
OR	$A + B$	$0b01$
XOR	$A \oplus B$	$0b10$
NOR	$A + B$	$0b11$

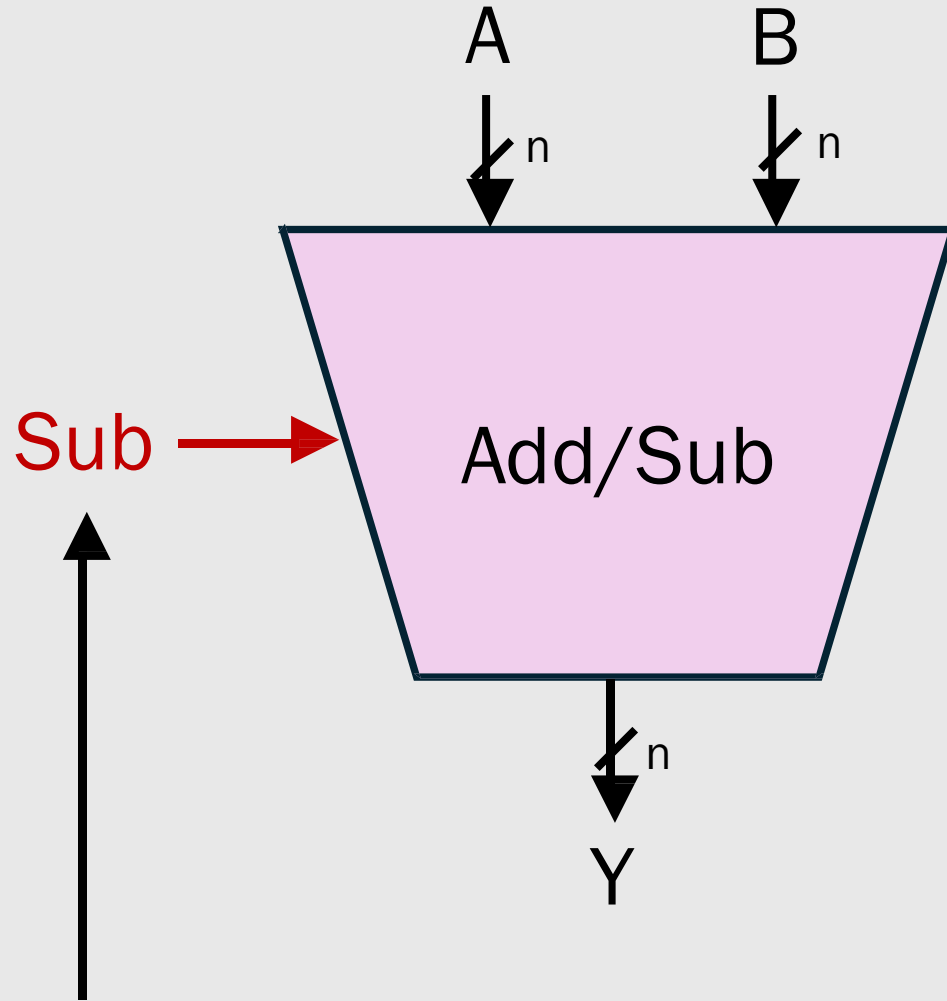
Bidirectional Shifter

- The left bit tells you the direction of the shift (left or right)
- The right bit tells you the type of shift (arithmetic or logical)
- Logical and arithmetic left shift are the same

Operation	Expression	Bool
Left Shift	$B \ll A$	0 0
None	N/A	0 1
Logical Right Shift	$B \gg A$	1 0
Arithmetic Right Shift	$B \ggg A$	1 1

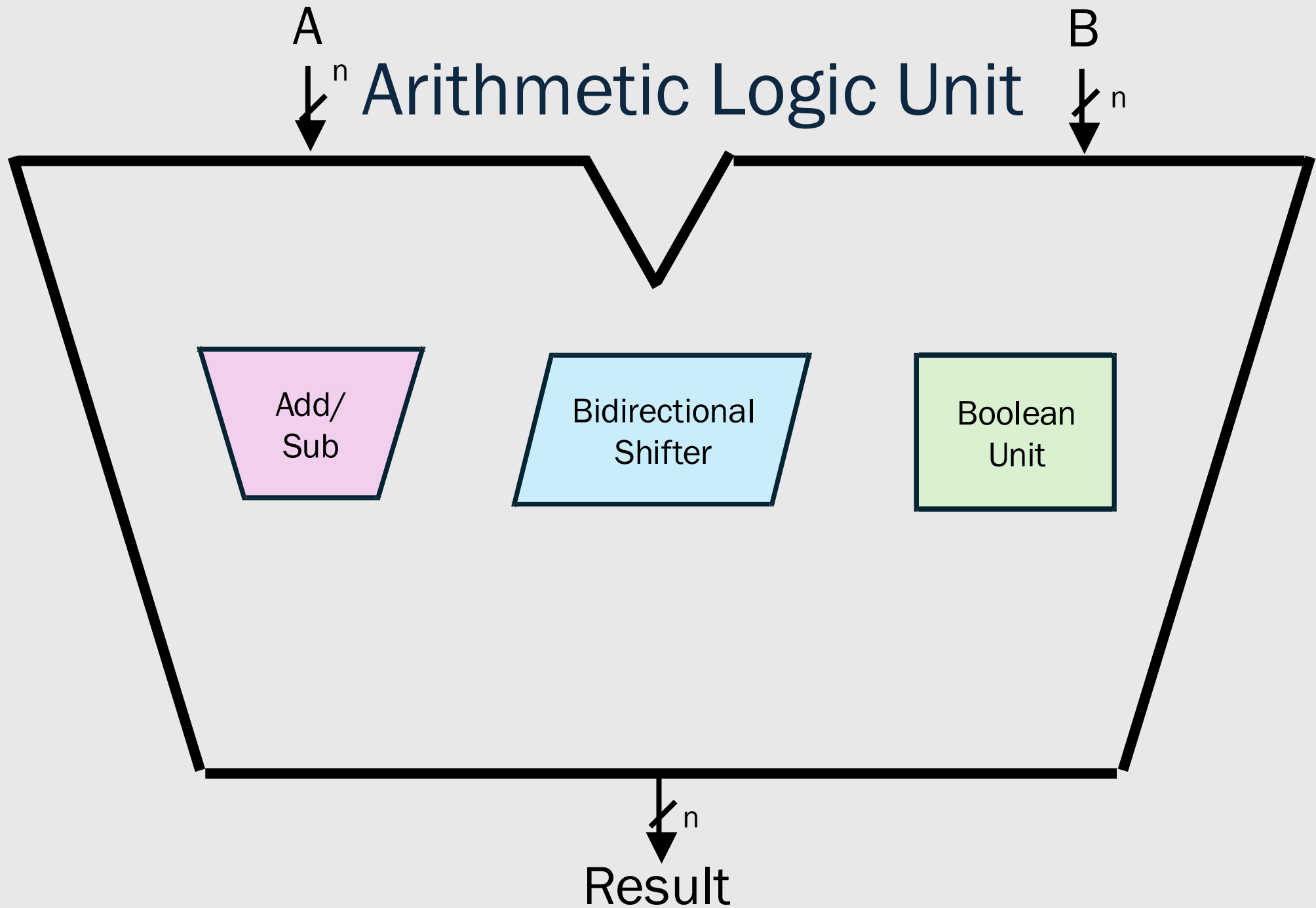
Keep the sign bit

Add/Sub

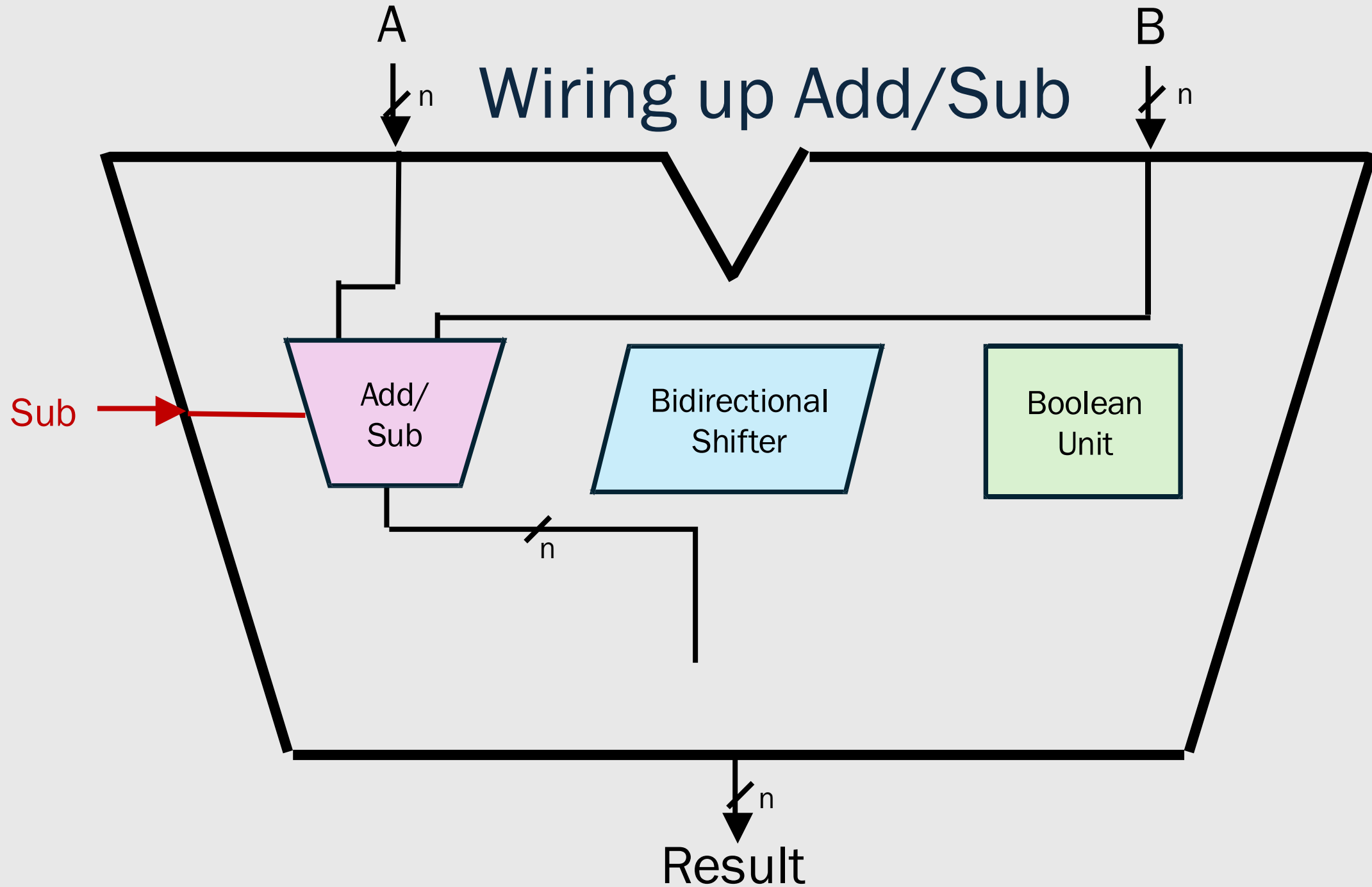


Control signal: Tells the circuit
which operation to perform

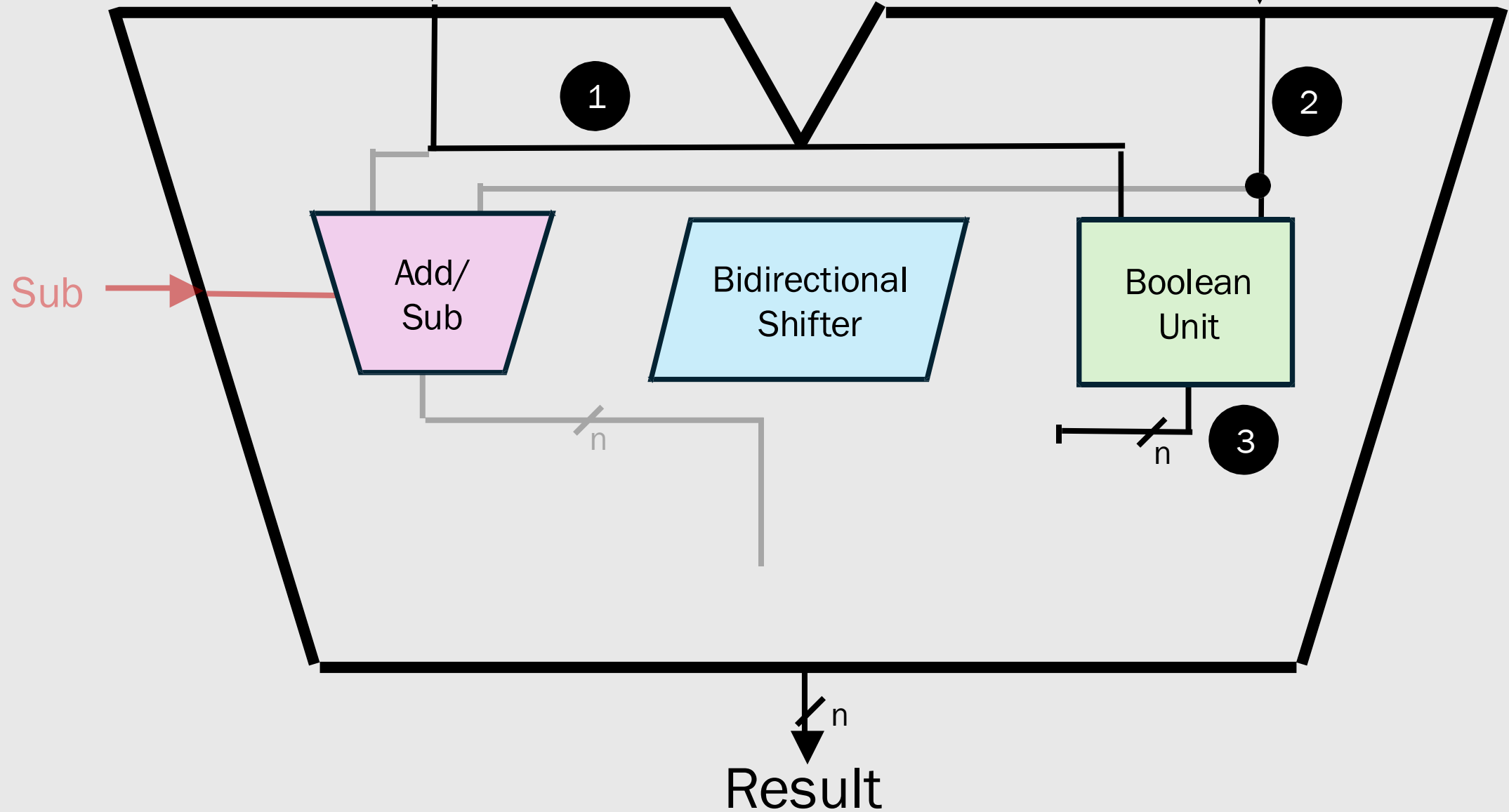
Operation	Expression	Sub
Add	$A + B$	0
Sub	$A - B$	1



Wiring up Add/Sub



Wiring up Boolean Unit

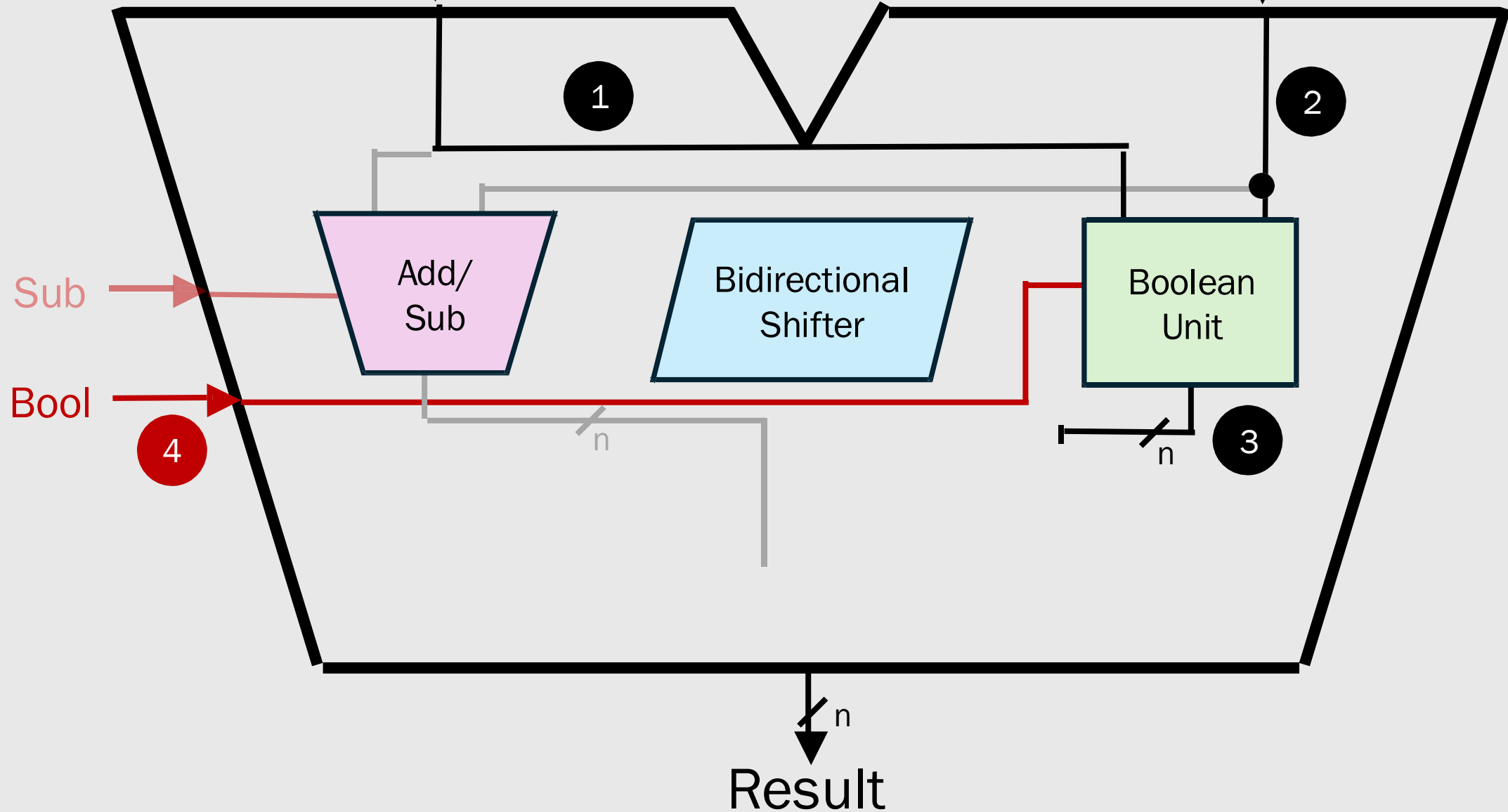


A

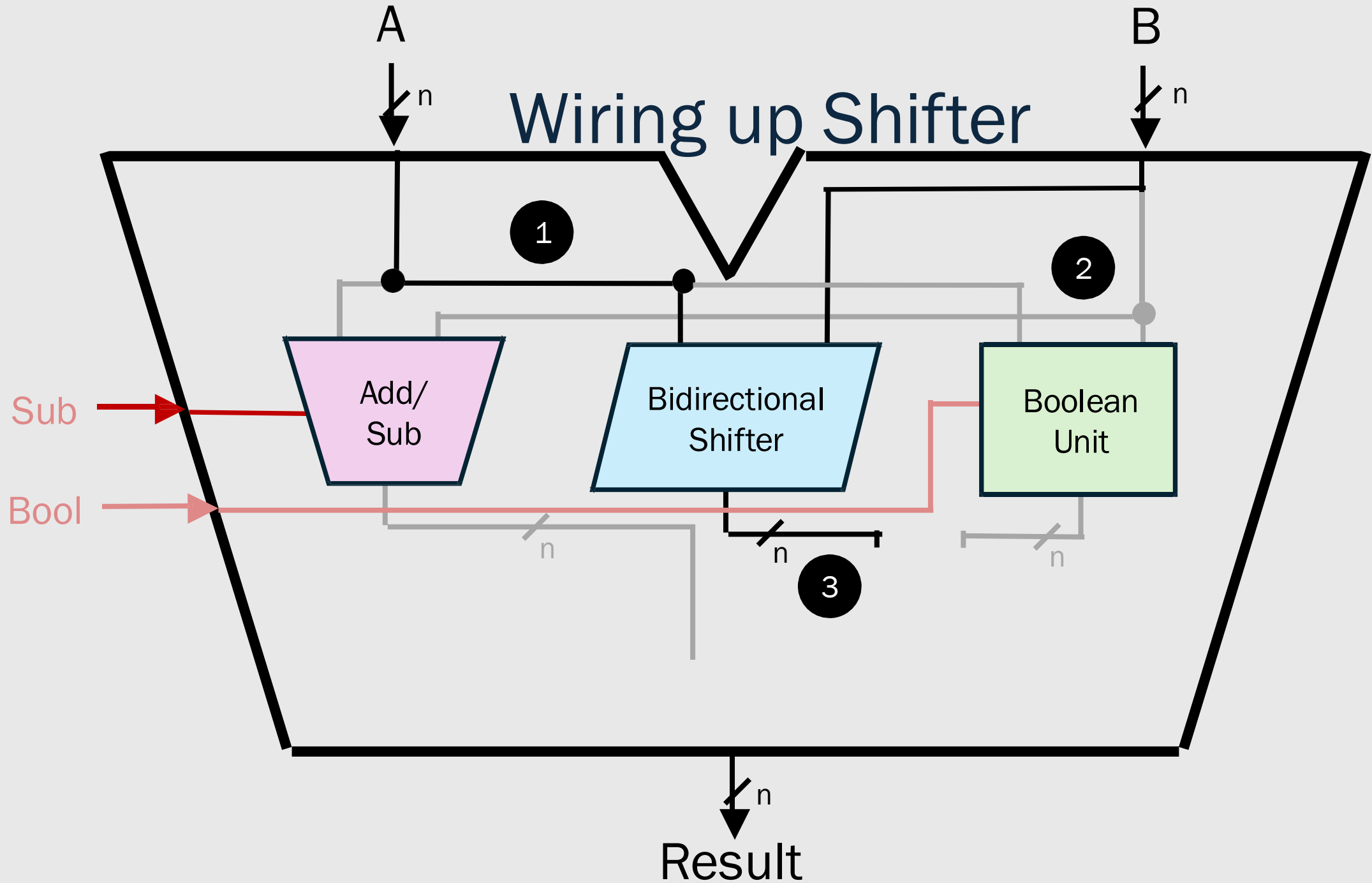
 n

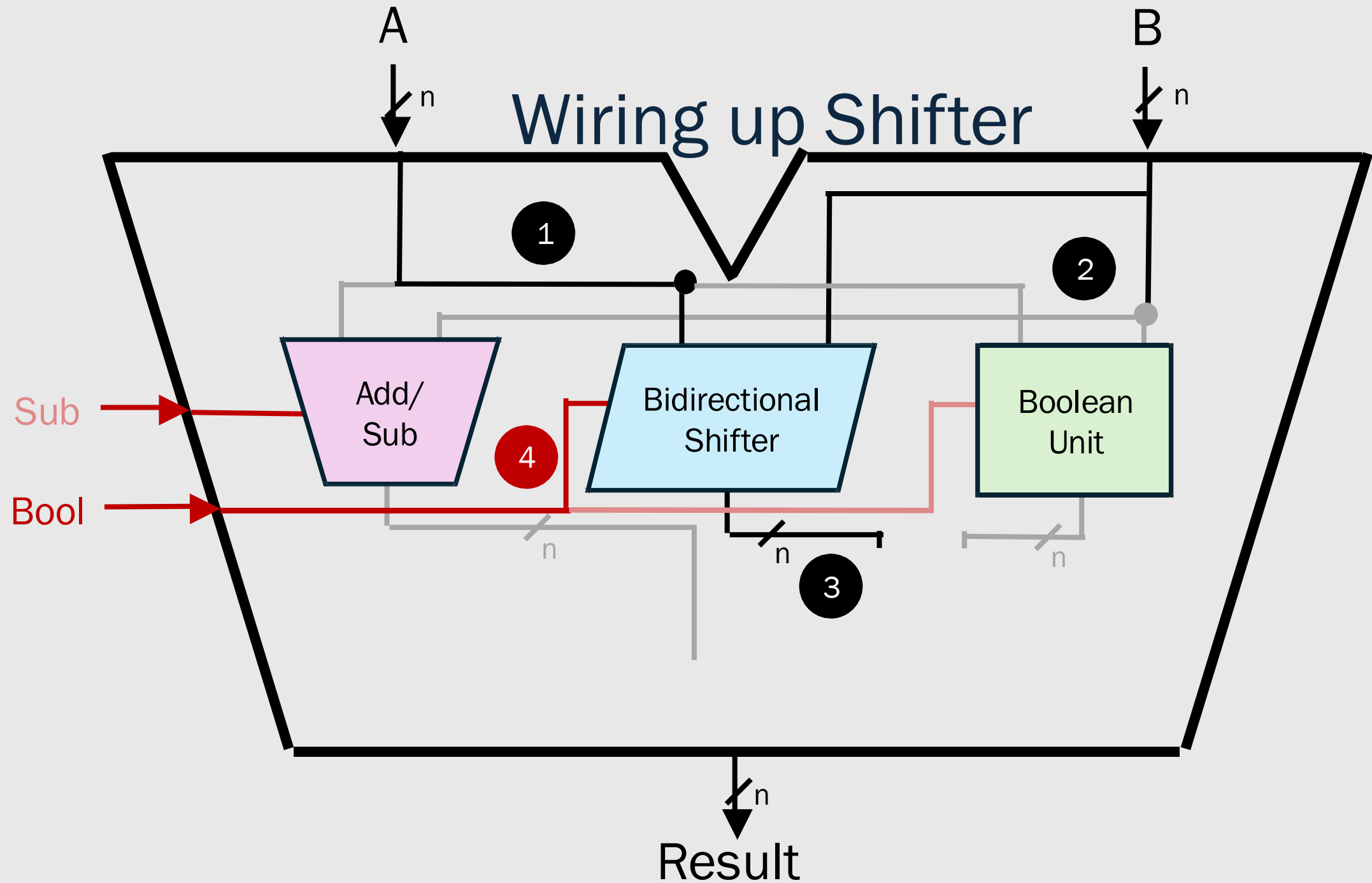
Wiring up Boolean Unit

B

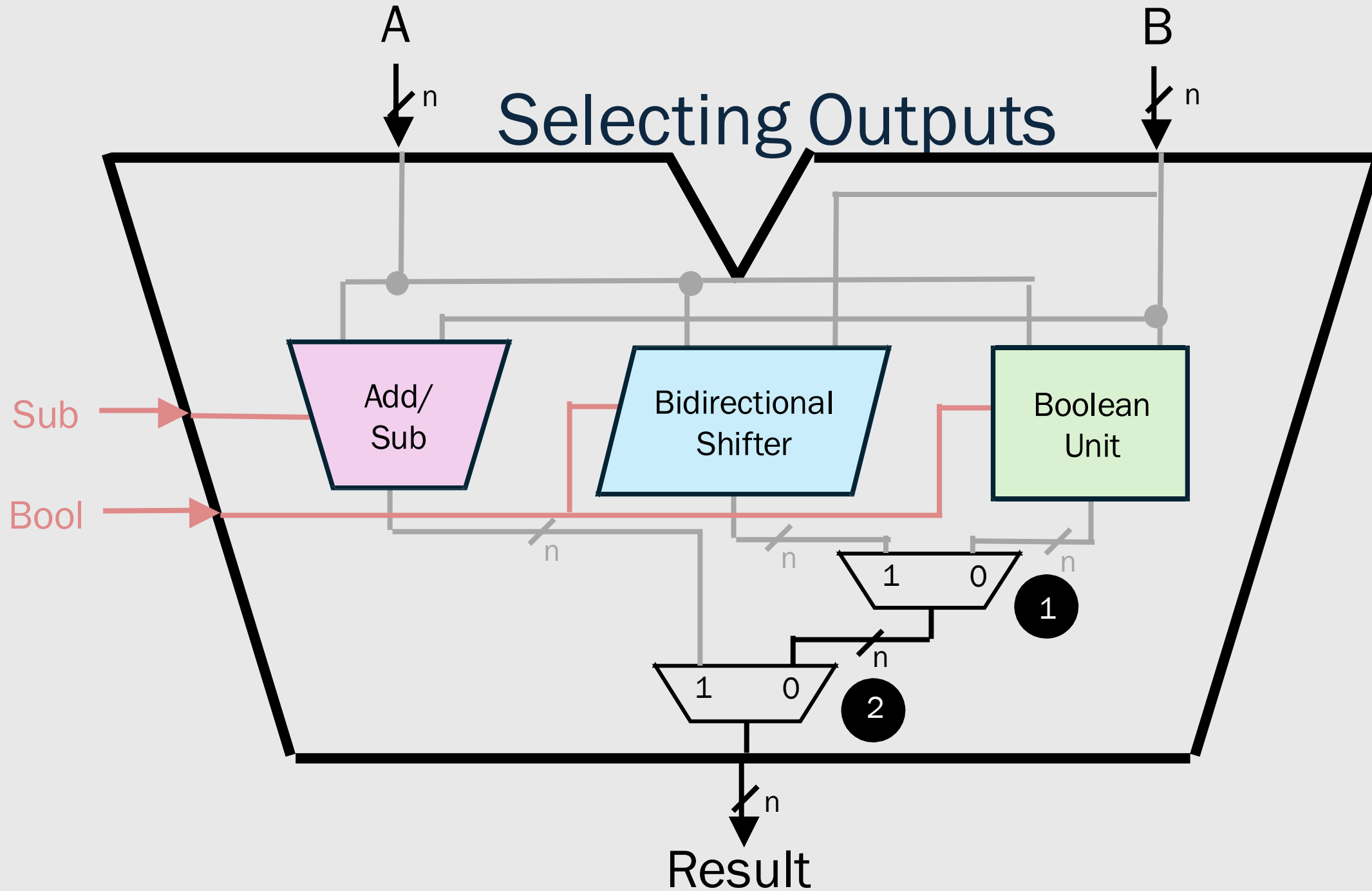
 n 

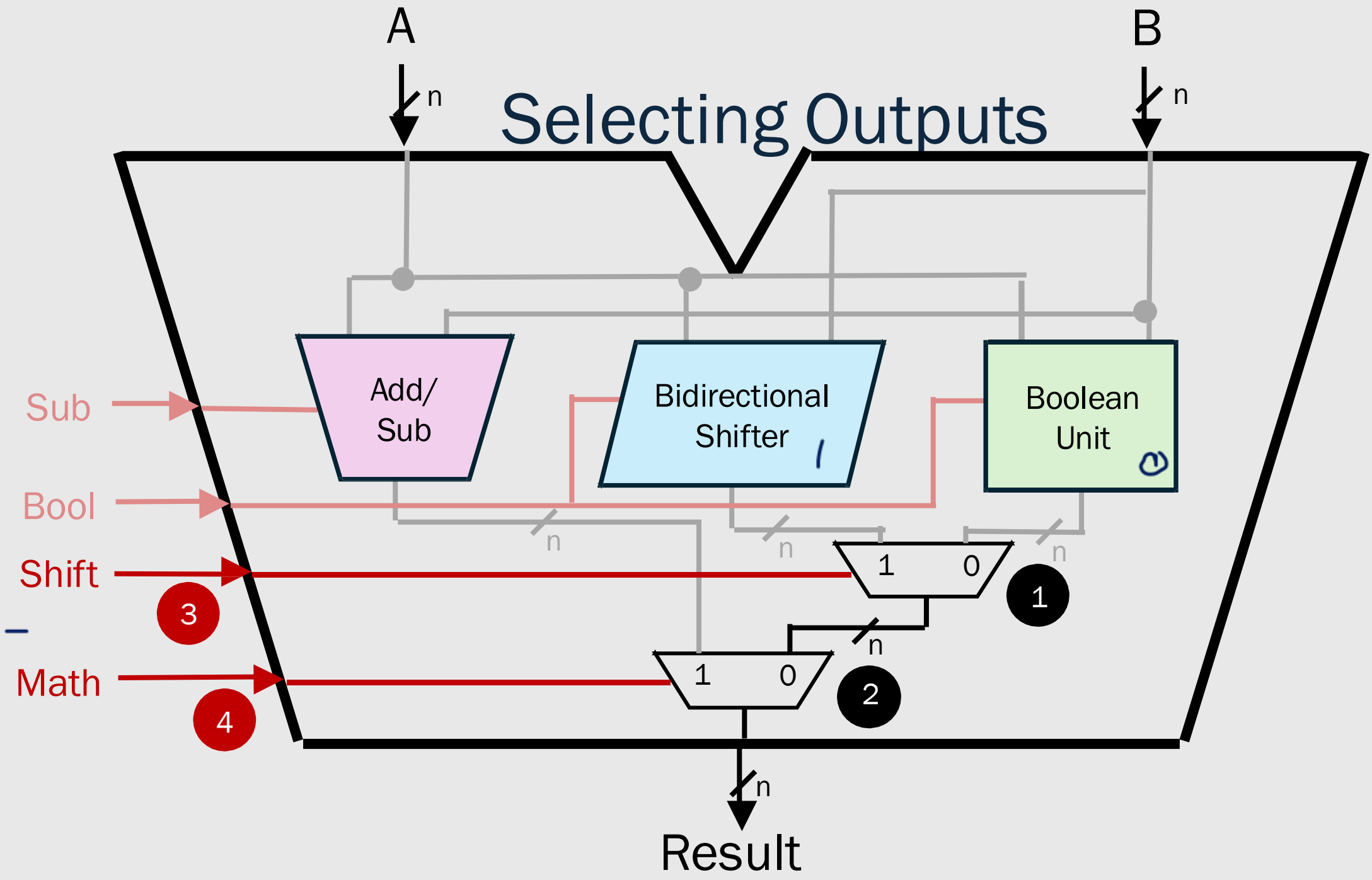
Wiring up Shifter

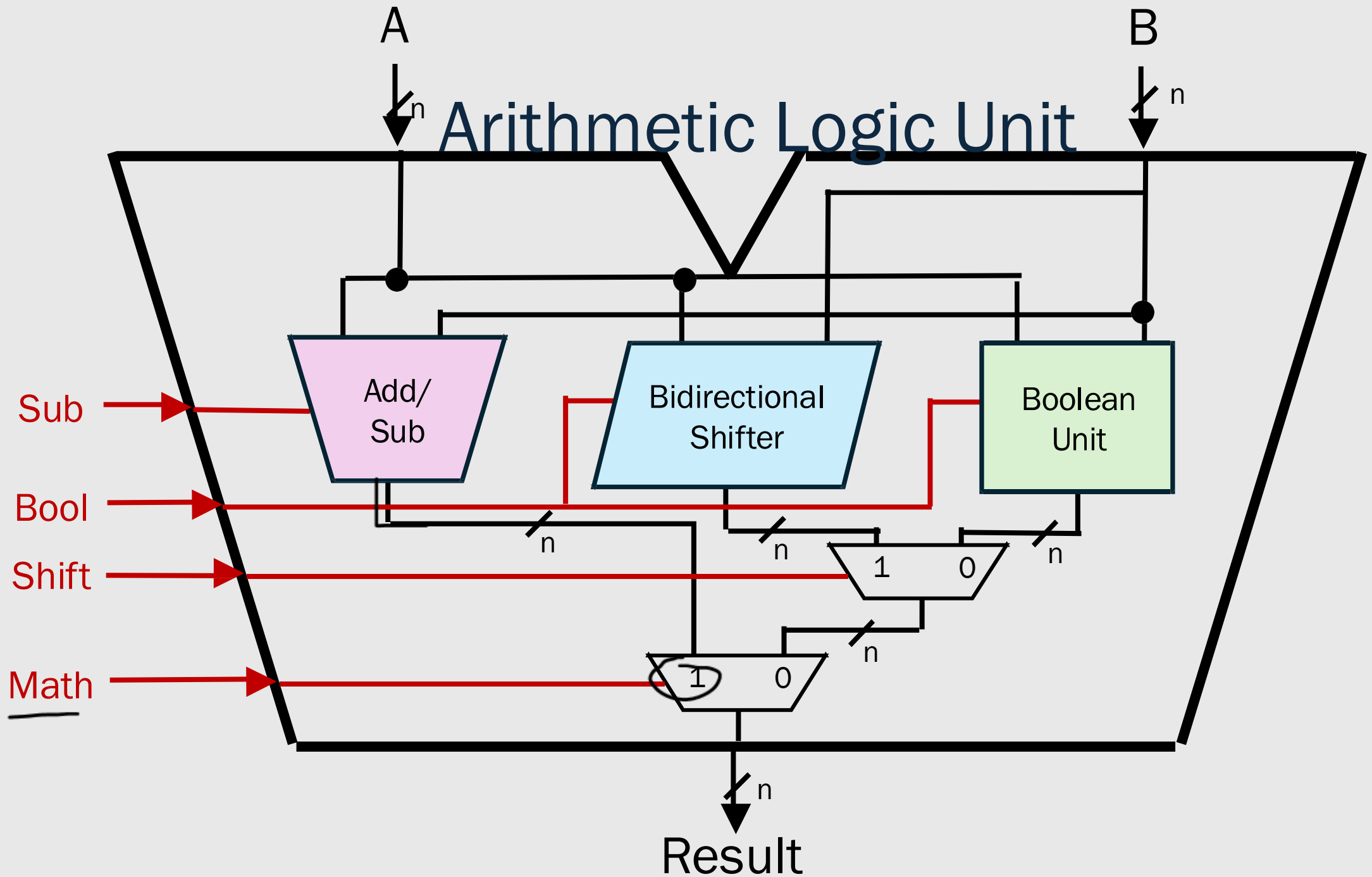




Selecting Outputs







ALU Control Bits

Operation	Expression	Sub
Add	$A + B$	0
Sub	$A - B$	1

Operation	Expression	Bool
Left Shift	<u>$B \ll A$</u>	<u>00</u>
Logical Right Shift	$B \gg A$	10
Arithmetic Right Shift	$B \ggg A$	11

Operation	Expression	Bool
AND	$A \times B$	<u>00</u>
OR	$A + B$	01
XOR	$A \oplus B$	10
NOR	$\overline{A + B}$	11

Operation	Sub	Bool	Shift	Math
$A + B$	0	XX	X	1
$A - B$				
$B \ll A$				
$B \gg A$				
$B \ggg A$				
$A \text{ AND } B$				
$A \text{ OR } B$				
$A \text{ XOR } B$				
$A \text{ NOR } B$				

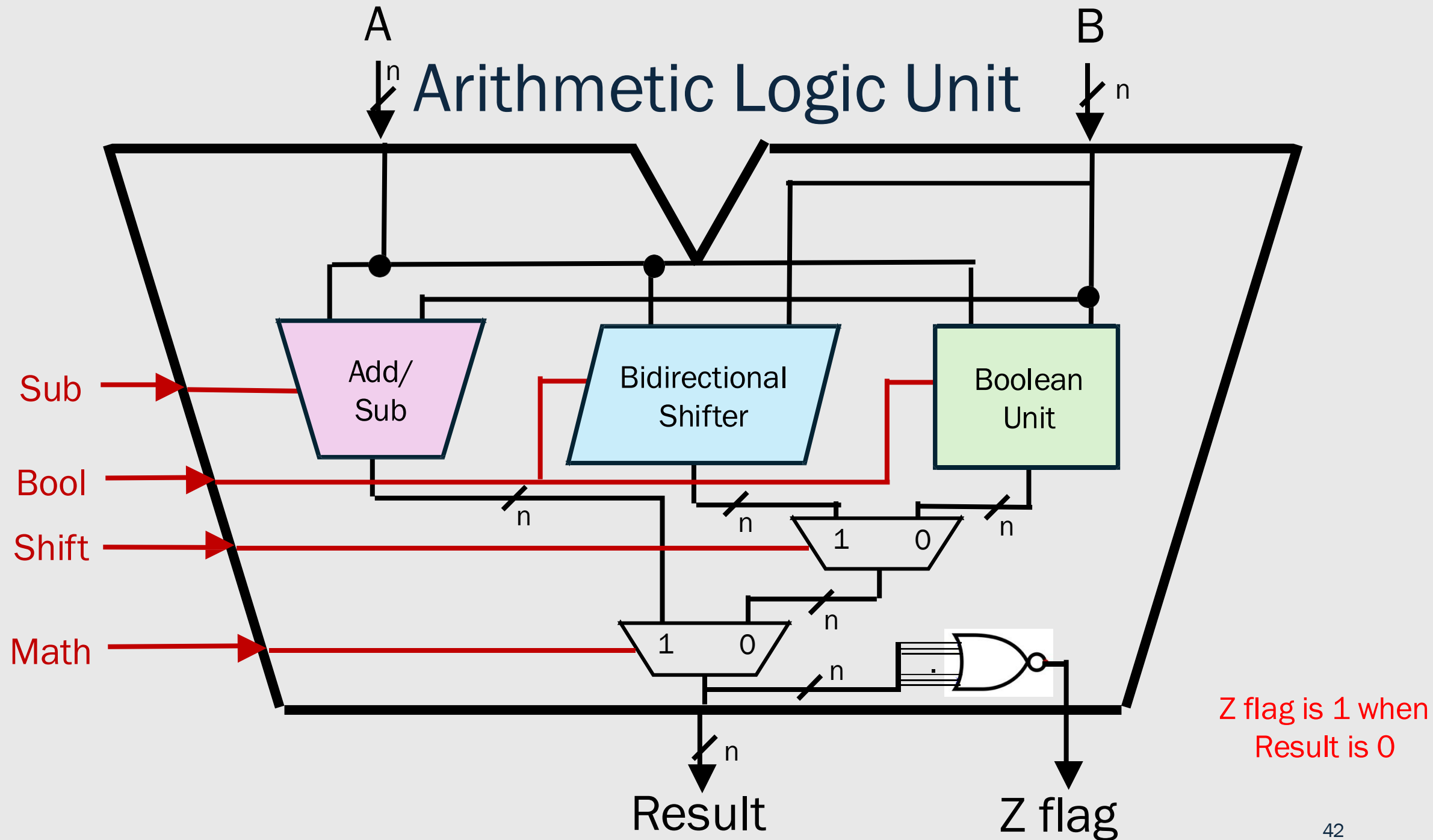
ALU Control Bits

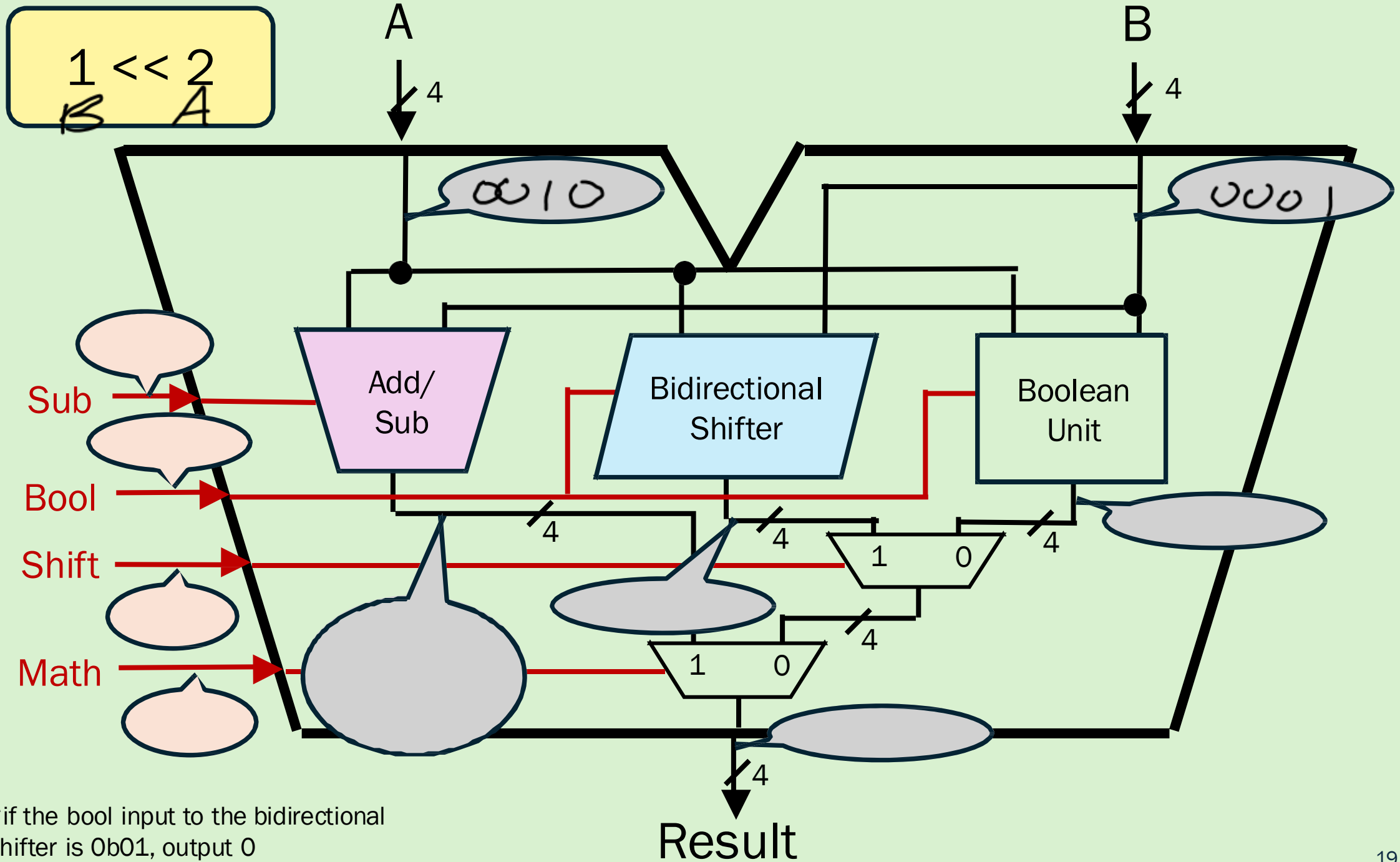
Operation	Expression	Sub
Add	$A + B$	0
Sub	$A - B$	1

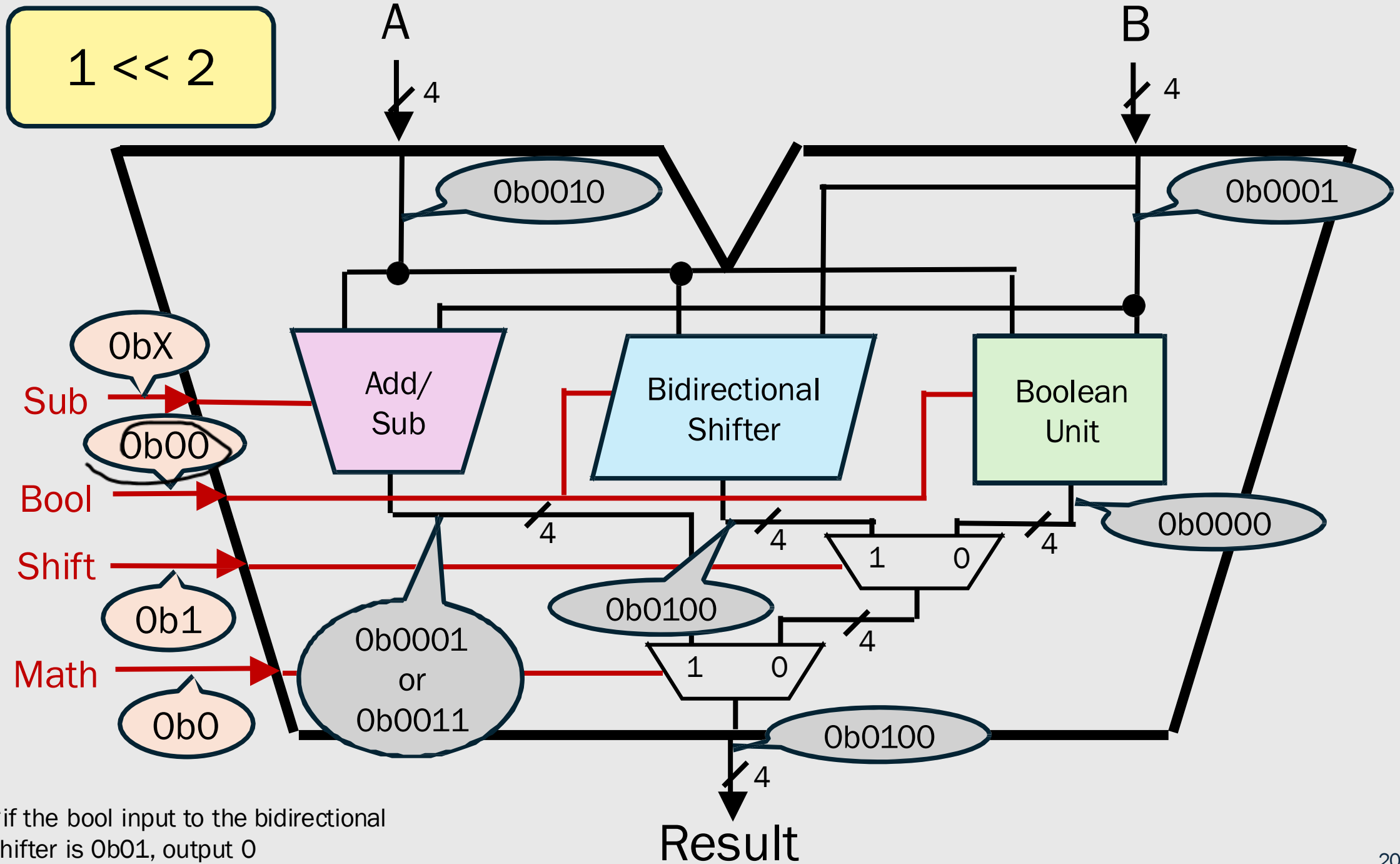
Operation	Expression	Bool
Left Shift	$B \ll A$	<u>00</u>
Logical Right Shift	$B \gg A$	10
Arithmetic Right Shift	$B \ggg A$	<u>11</u>

Operation	Expression	Bool
AND	$A \times B$	00
OR	$A + B$	01
XOR	$A \oplus B$	<u>10</u>
NOR	$\overline{A + B}$	11

Operation	Sub	Bool	Shift	Math
$A + B$	0b0	0bXX	0bX	0b1
$A - B$	0b1	0bXX	0bX	0b1
$B \ll A$	0bX	<u>0b00</u>	0b1	0b0
$B \gg A$	0bX	<u>0b10</u>	0b1	0b0
$B \ggg A$	0bX	<u>0b11</u>	0b1	0b0
$A \text{ AND } B$	0bX	0b00	0b0	0b0
$A \text{ OR } B$	0bX	0b01	0b0	0b0
<u>$A \text{ XOR } B$</u>	0bX	<u>0b10</u>	0b0	0b0
$A \text{ NOR } B$	0bX	0b11	0b0	0b0

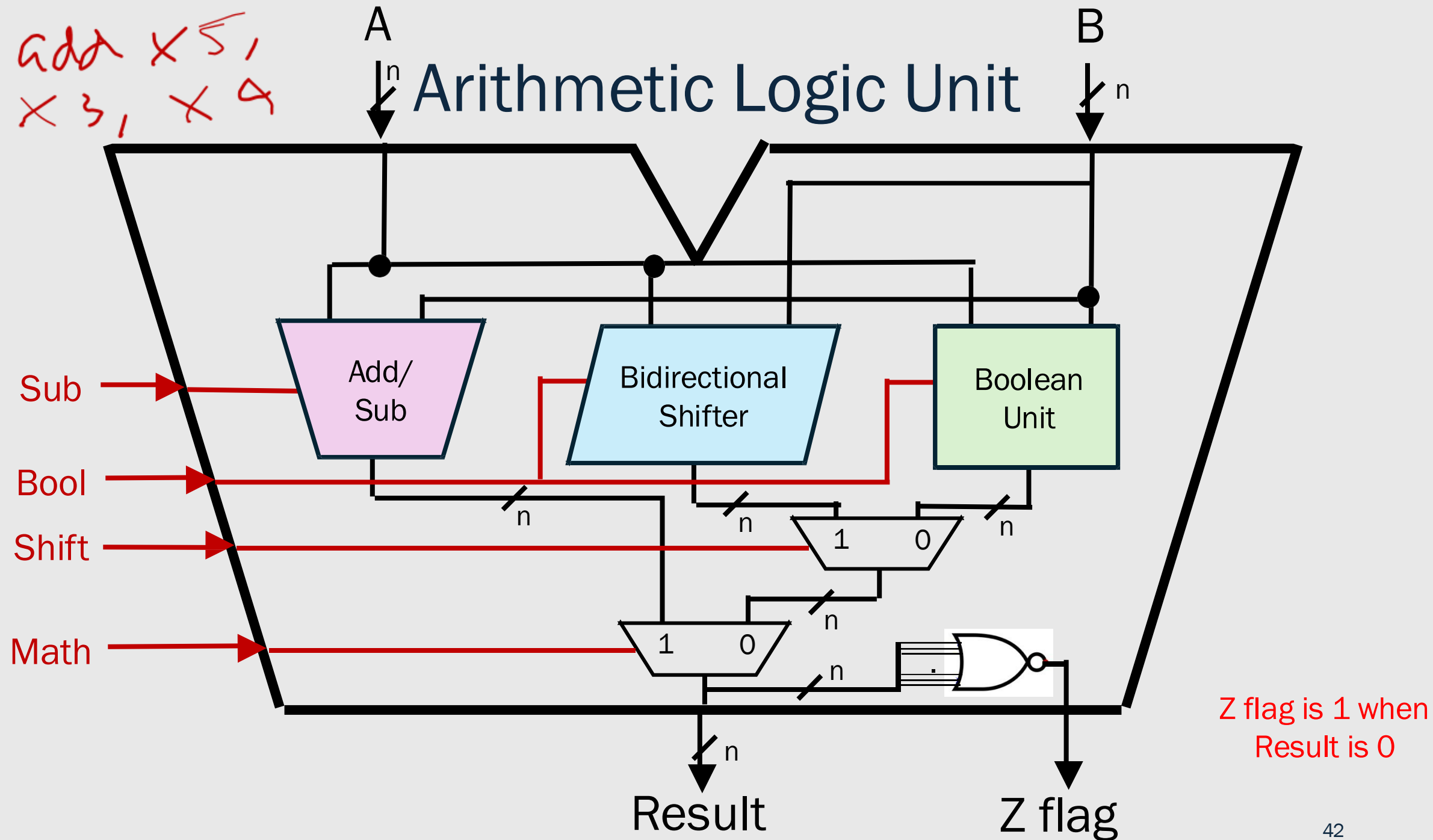






*if the bool input to the bidirectional shifter is 0b01, output 0

add x5,
x3, x4





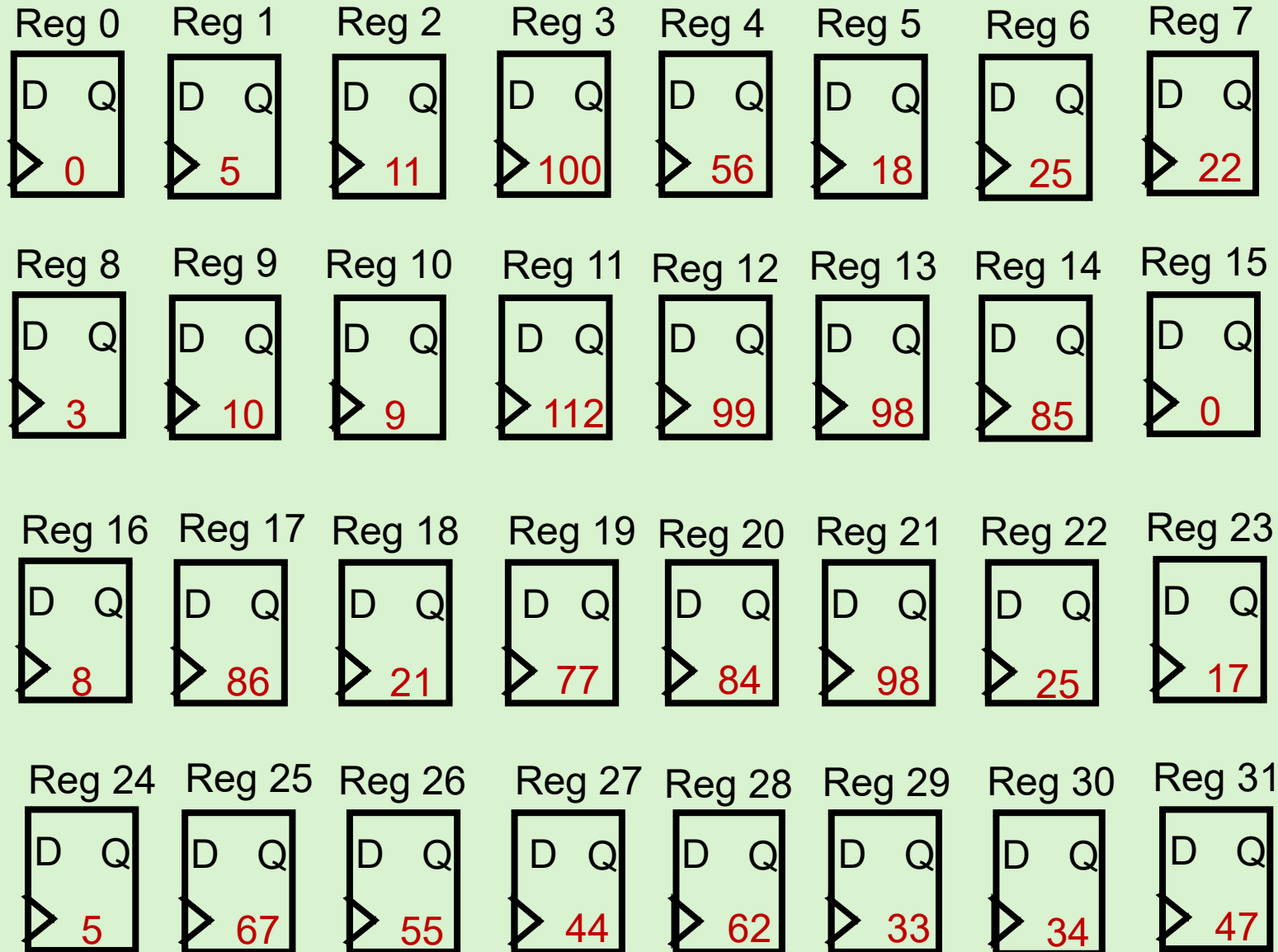
THE REGISTER FILE



Reminder: What is a register?

- Registers: special locations built directly into the hardware to store data!
- These serve as the operands of many assembly instructions
- These primitives are also visible to the assembly programmer once the computer is built.
- The size of a register is determined by the architecture
 - *We are assuming 32-bit registers in this class!*
- Registers are also synchronized with a clock!
 - *We will get to this next week. 😊*
- The processor's 32 general-purpose registers are stored in a structure called a register file

Register File



- Contains the register state of the computer
- Any register can be read or written, just specify the number of the register

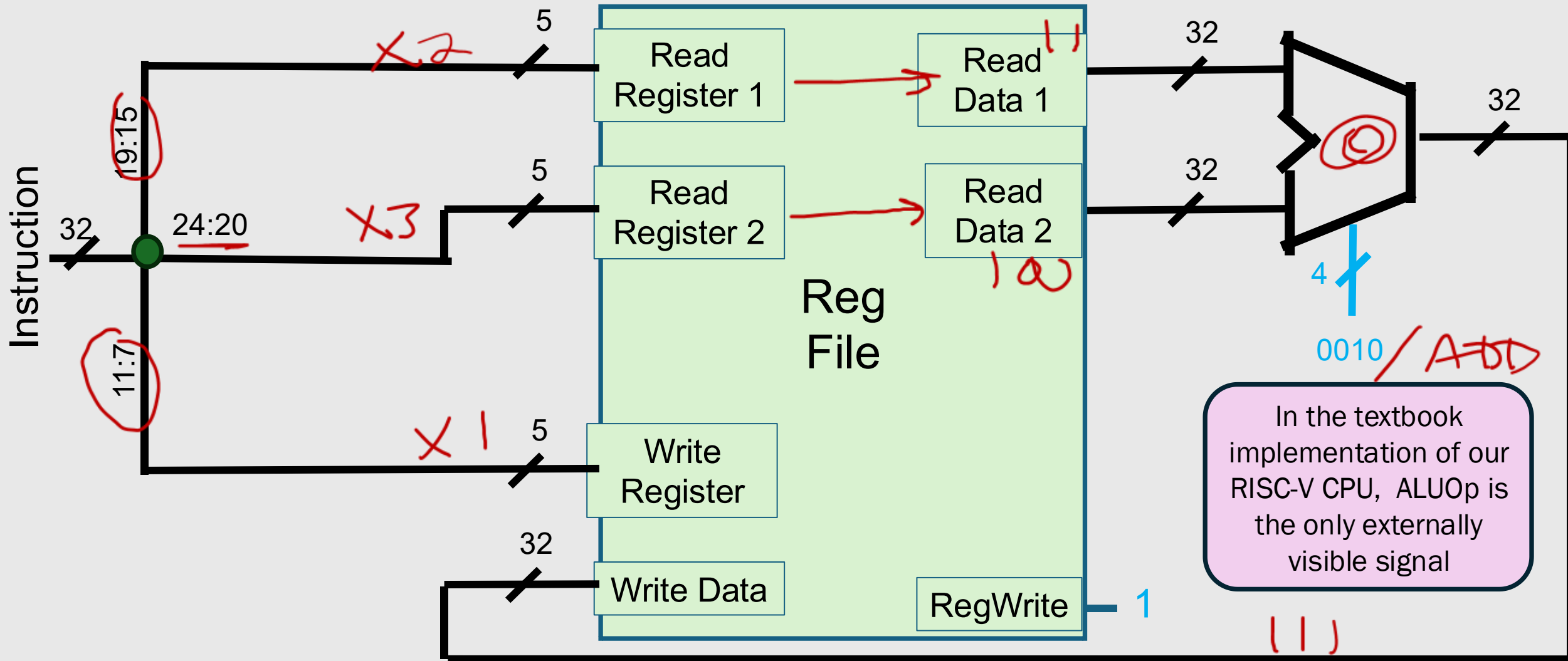
$$x1 = x3 + x2$$

funct7	source register 2	source register 1	funct3	destination register	opcode
0000000	00011	00010	000	00001	0110011
31	25 24	20 19	15 14	12 11	7 6
	x3	x2		x1	

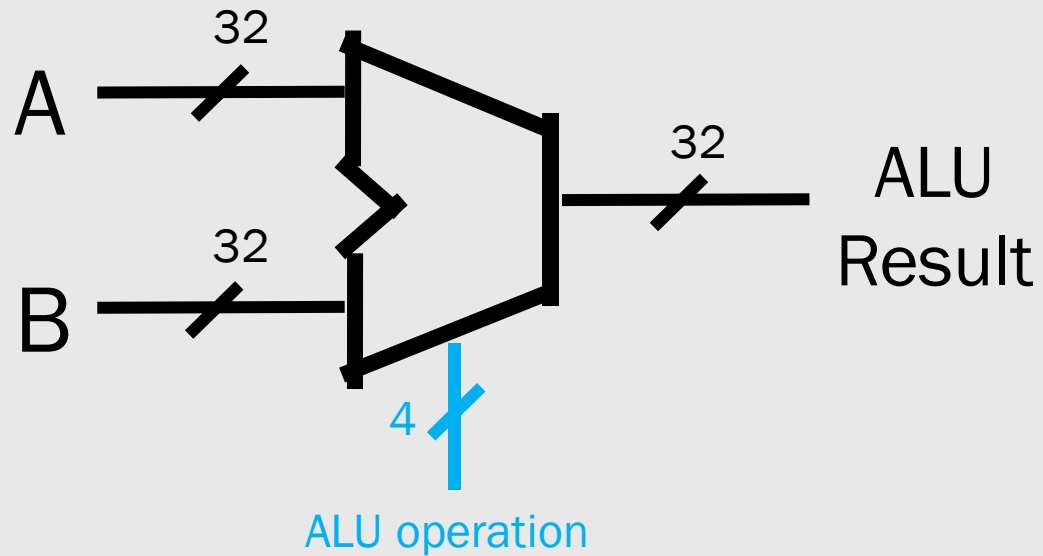
000000000011000100000000010110011

Opcode is always 0110011 for integer R-type ALU instructions
 funct3 is always the primary operation code group (e.g. 000 = add/sub. 100 = xor, 111 = and)
 funct7 is used to further specify the operation (e.g. distinguishes add vs. sub. srl vs. sra)

funct7	rs2	rs1	funct3	rd	opcode	
0000000	00011	00010	000	00001	0110011	
31	25 24	20 19	15 14	12 11	7 6	0



RISC-V 32I ALU



Function	ALU Operation
AND	0b0000
OR	0b0001
ADD	0b0010
SUB	0b0110
set on less than (SLT)	0b0111

Control Signal Input

Set on Less Than

- If $A < B$
 - *ALU result* = 1
- If $A \geq B$
 - *ALU result* = 0

What are the contents of the register file after the following instructions are executed?



0x003202b3
0x00528533
0x00EA6020

Contents of RegFile

Reg 0	Reg 1	Reg 2	Reg 3	Reg 4	Reg 5	Reg 6	Reg 7
Reg 8	Reg 9	Reg 10	Reg 11	Reg 12	Reg 13	Reg 14	Reg 15
Reg 16	Reg 17	Reg 18	Reg 19	Reg 20	Reg 21	Reg 22	Reg 23
Reg 24	Reg 25	Reg 26	Reg 27	Reg 28	Reg 29	Reg 30	Reg 31

What are the contents of the register file after the following instructions are executed? $\rightarrow$ **0x003202b3**

funct7	source register 2	source register 1	funct3	destination register	opcode
0000 000	000 1 1	00 100	000	00 10 1	0110 011
31	25 24	20 19	15 14	12 11	7 6 0

X3

X9

X5

R-Type

$$X5 = 56 + 100$$

$$X5 = 156$$

add X5, X4, X3

What are the contents of the register file
after the following instructions are
executed? $\longrightarrow$ 0x003202b3

$\longrightarrow$ 0b000000000001100100000001010110011

funct7	source register 2	source register 1	funct3	destination register	opcode
0000000	00011	00100	000	00101	0110011
31	25 24	20 19	15 14	12 11	7 6 0

$$x5 = x4 + x3$$

$$x5 = 56 + 100$$

$$x5 = 156$$

add x5, x4, x3

What are the contents of the register file
after the following instructions are
executed? **0x00528533**

funct7							source register 2							source register 1							funct3							destination register							opcode						
31							25							20							15						12							7							0

What are the contents of the register file after the following instructions are executed?

0x00528533

0b000000000010100101000010100110011

funct7							source register 2							source register 1							funct3							destination register							opcode													
0000000							00101							00101							000							01010							0110011													
31							25 24							20 19							15 14							12 11							7 6							0						

$x10 = x5 + x5$
 $x10 = 156 + 156$
 $x10 = 312$

add x10, x5, x5

What are the contents of the register file
after the following instructions are
executed? **0x00a38633**

funct7							source register 2							source register 1							funct3							destination register							opcode						
31							25							20							15						12							7							0

What are the contents of the register file after the following instructions are executed?

0x00a38633

0b00000000101000111000011000110011

funct7								source register 2								source register 1								funct3				destination register				opcode					
0000000								01010								00111								000				01100				0110011					
31								25 24								20 19								15 14				12 11				7 6					

$$x12 = x7 + x10$$

$$x12 = 22 + 312$$

$$x12 = 334$$

add x12, x7, x10

Final Contents of RegFile

Reg 0	Reg 1	Reg 2	Reg 3	Reg 4	Reg 5	Reg 6	Reg 7
Reg 8	Reg 9	Reg 10	Reg 11	Reg 12	Reg 13	Reg 14	Reg 15
Reg 16	Reg 17	Reg 18	Reg 19	Reg 20	Reg 21	Reg 22	Reg 23
Reg 24	Reg 25	Reg 26	Reg 27	Reg 28	Reg 29	Reg 30	Reg 31

func17: 00000000 func3: 111

What is the value of each signal when executing: and x4, x2, x5

opcode: 0110011

