

Computer Organization

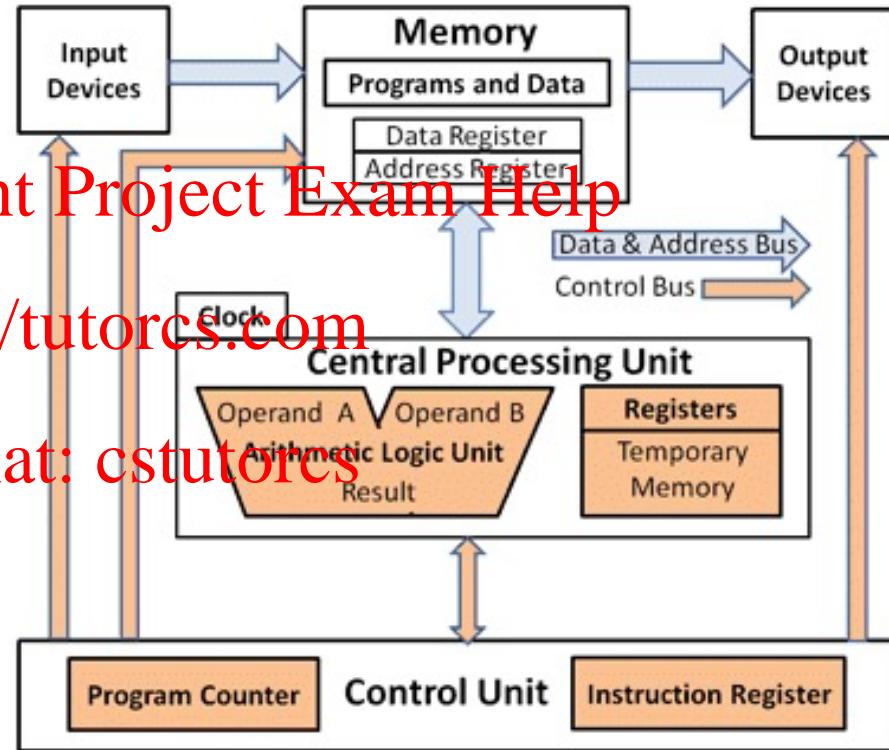
Brent C. Munsell

ALU Circuit Design Add,
Subtract, Shift, and
Logic Operations

Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs



Agenda

- Announcements (~ 5 mins)
- No Poll Everywhere
- Quickly finish common circuit design (~25 mins)
- Add/Subtract
- Shift
- Logic
- ALU

Assignment Project Exam Help

<https://tutorcs.com>

(~45 mins)

WeChat: cstutorcs

Announcements

- Release Lab 1 tomorrow (9/8)
 - 7 segment display circuit design
 - think you'll really like **Assignment Project Exam Help**
- Great job on **https://tutorcs.com**
- Quiz 1 (see schedule, Canvas announcement)
WeChat: estutorcs

Assignment Project Exam Help

<https://tutorcs.com>

Add and Subtract
WeChat: cstutores

Combinational Logic Circuit Design

Binary Addition

Not a single operation

- $A + B = \text{Sum and Carry-out}$
- $A, B, \text{Sum} (S)$ and Carry-out (C_o) are one-bit binary values

$$\begin{array}{r} C_o \\ A \\ + B \\ \hline S \end{array}$$

<https://tutorcs.com>

Four possibilities (A and B):

$$\begin{array}{r} 0 \\ + 0 \\ \hline 0 \end{array}$$

$$\begin{array}{r} 0 \\ + 1 \\ \hline 1 \end{array}$$

$$\begin{array}{r} 1 \\ + 0 \\ \hline 1 \end{array}$$

$$\begin{array}{r} 1 \\ + 1 \\ \hline 0 \end{array}$$

Binary Half Adder Circuit Design

Two input bits: 

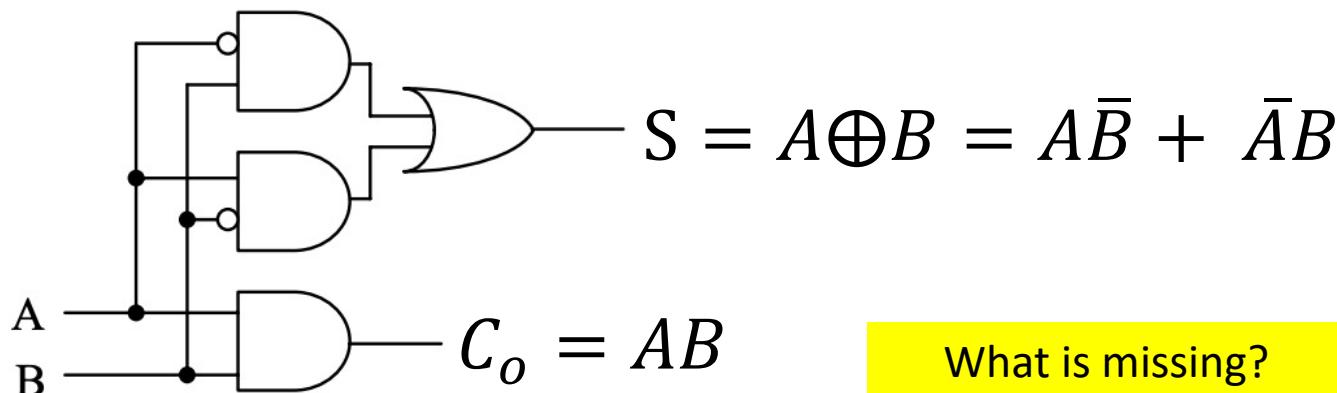
- A and B: 1 bit each

Assignment Project Exam Help
Two output bits: 

- C and S: 1 bit each

A	B	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

WeChat: cstutorcs



What is missing?

Binary Full Adder Circuit Design

Three input bits: 

- C_{in} , A, B: 1 bit each

Two output bits: 

- C_o and S: 1 bit each

$$\begin{array}{r} C_o \quad C_i \\ A \quad \quad \quad \\ + \quad B \\ \hline S \end{array}$$

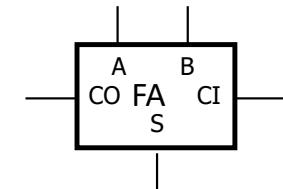
C_i	A	B	C_o	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

$$C_o = C_i(A + B) + AB = C_i(A \oplus B) + AB$$

$$S = C_i \oplus A \oplus B$$

Full Adder Circuit

Schematic
Diag.

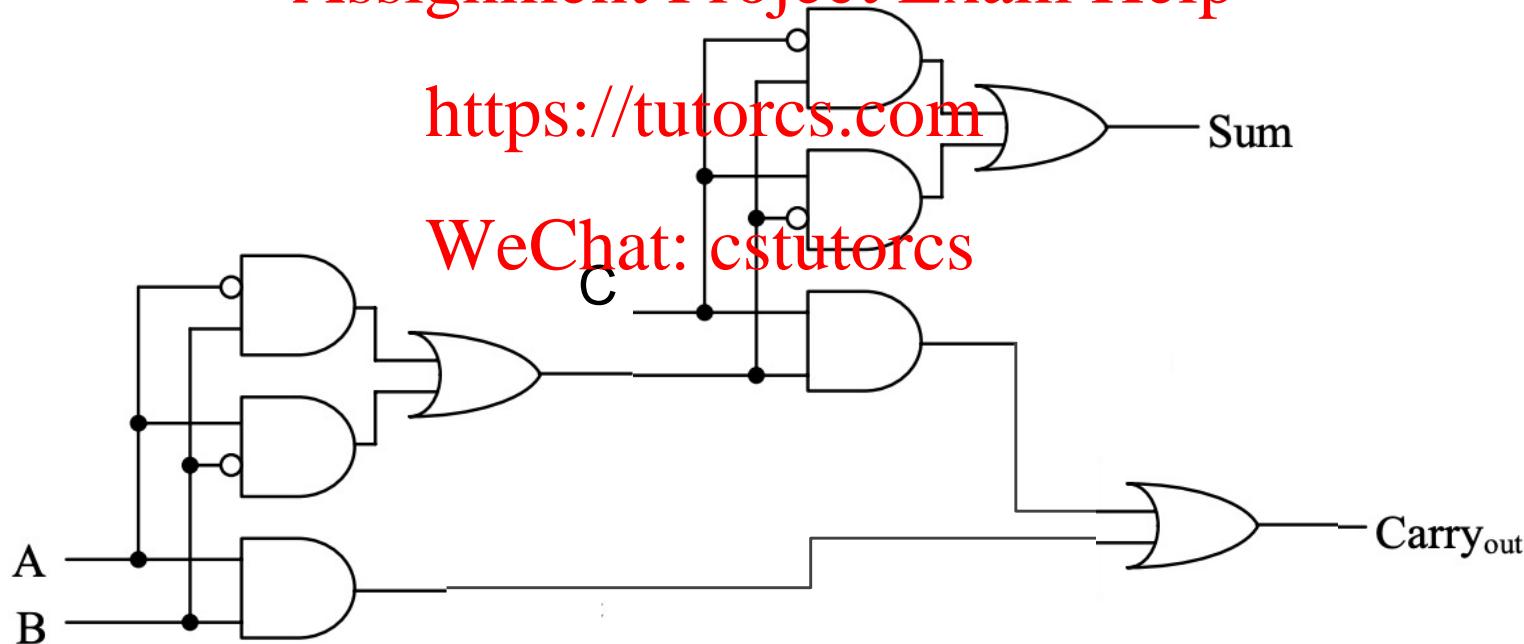


Two HA circuits plus one or gate

Assignment Project Exam Help

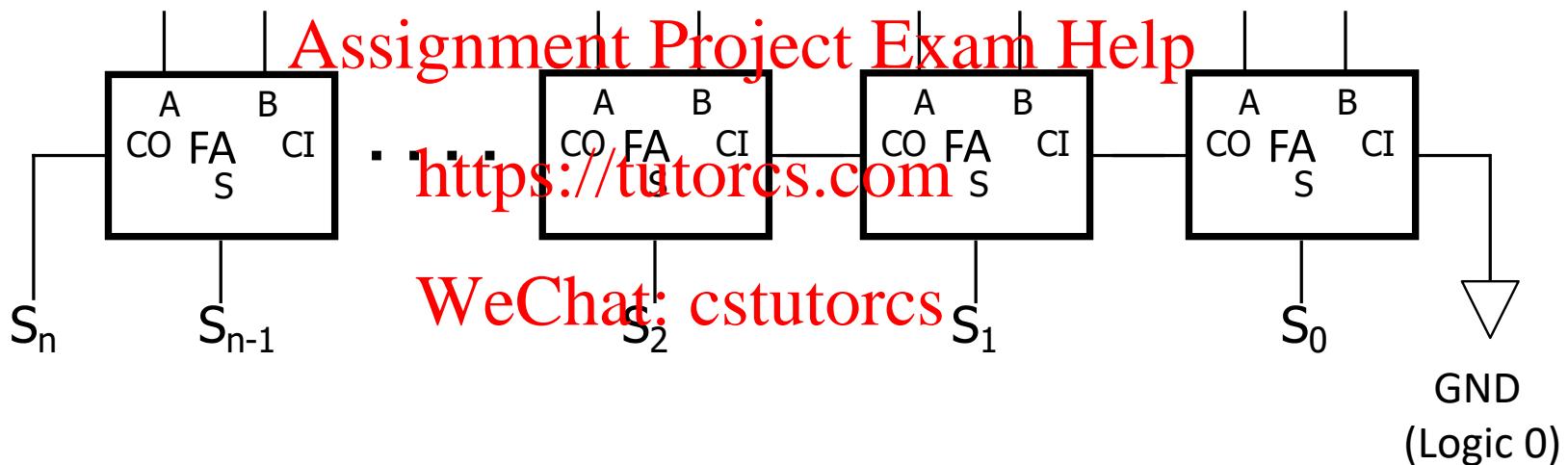
<https://tutorcs.com>

WeChat: cstutorcs



Binary Addition Component

Using FA circuits, extend to arbitrary # of bits



“Ripple-Carry Adder”

- carries ripple through from right to left
- longest chain of carries has length n

Not the most efficient design, why?

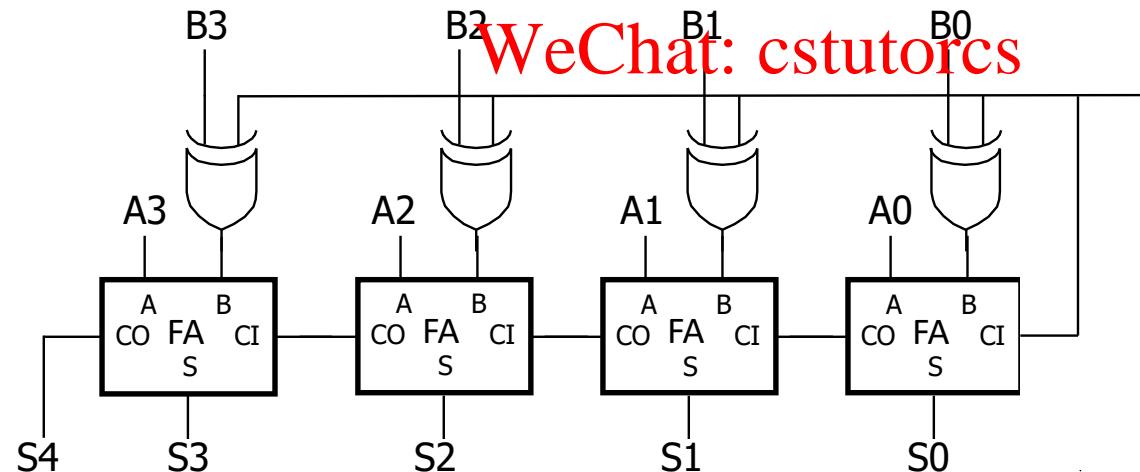


Binary Addition/Subtract Component (4-bit Example)

Subtract A-B: 2's complement Operation

Assignment Project Exam Help bit-wise complement
2's complement: $-B = \sim B + 1$

<https://tutoros.com>



control bit

control bit = 0: Add
control bit = 1: Subtract

Carry-out bit

We'll see how this is used very soon!

Assignment Project Exam Help

<https://tutorcs.com>

Bit Shift Circuit

Shift right and left

Shift Operations: Overview

Left Shift: shifts in a 0 from the right end

- $(X \ll 1) = 00101000_2 = 40_{10}$

$$X = 20_{10} = 00010100_2$$

Assignment Project Exam Help

"Logic" Right Shift: shifts in a 0 from the left end

- $(X >> 1) = 00001010_2 = 10_{10}$

WeChat: cstutorcs

"Arithmetic" Right Shift: maintains the sign bit

- $-X = -20_{10} = 2\text{'s complement of } X = 11101100_2$
- $(-20_{10} >>> 1) = (11101100_2 >>> 1) = 11110110_2 = -10_{10}$

Note:

- shift right arithmetic notation ($>>>$)
- shift right logic notation ($>>$)

Bit Shift Component Design

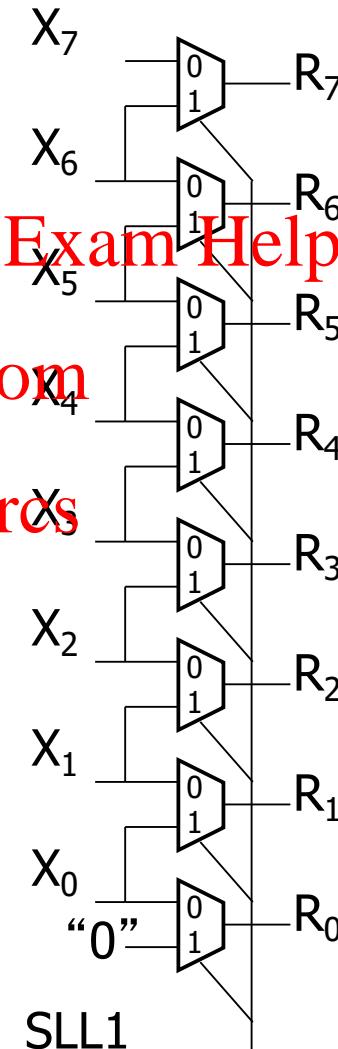
Example Shift Left Circuit

If SLL1 is true (logic 1)

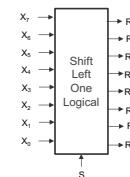
- Shifts the input X one bit to the left
- $R \leftarrow X \ll 1$

If SLL1 is false (logic 0)

- Do not shift X
- $R \leftarrow X$



Schematic
Diag.



Shift by other amounts

Shift left by 2

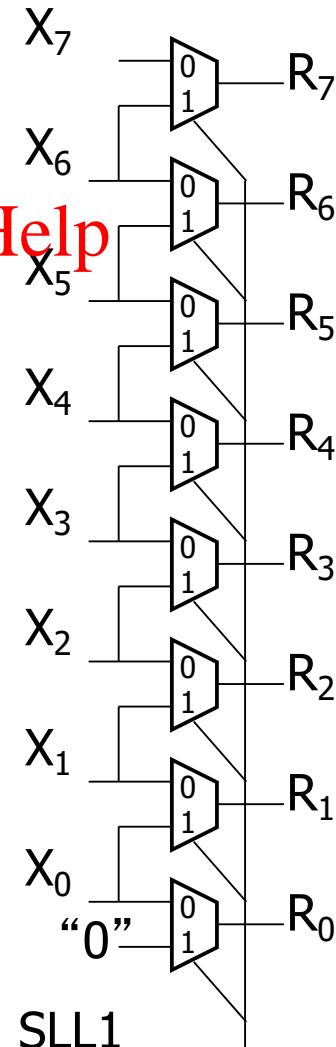
- Rewire the multiplexors so each X_i feeds into R_{i+2}
- Similarly: shift left by 4, etc.

<https://tutorcs.com>

Shift right circuits have similar circuitry

WeChat: cstutorcs

- shift right logical: each X_i feeds into a lower numbered R_j
- shift right arithmetic: sign bit stays the same



5-bit Shift Left Circuit

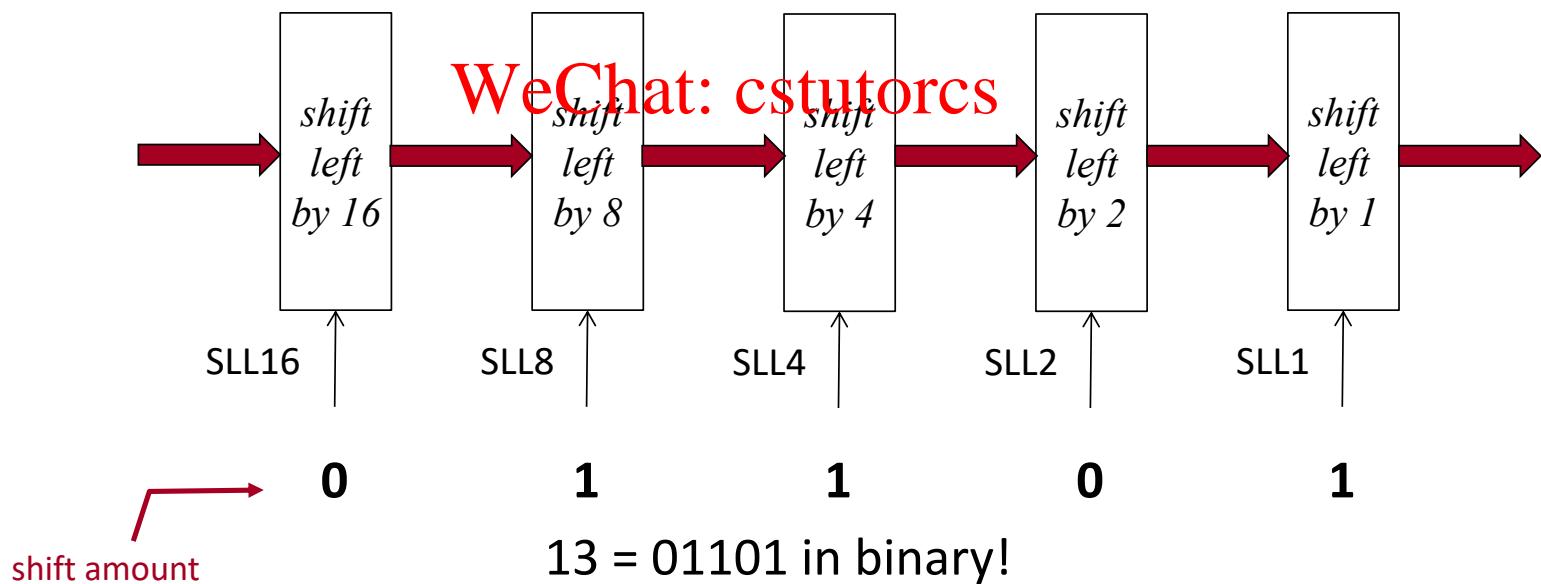
Make five shift components: SLL1, SLL2, SLL4, SLL8, SLL16

- Any arbitrary shift amount can be made by combining these shifts

Assignment Project Exam Help

Example: $SLL\ 13 = SLL\ (8 + 4 + 1)$

<https://tutorcs.com>



Assignment Project Exam Help

<https://tutorcs.com>

Logic Circuit

Bitwise AND, OR, XOR, NOR

Boolean Logic

Boolean component to perform bit-wise logic operations

- AND ($Y = AB$)
- OR ($Y = A \oplus B$)
- XOR ($Y = A \oplus B$)
- NOR ($Y = \overline{A + B}$)

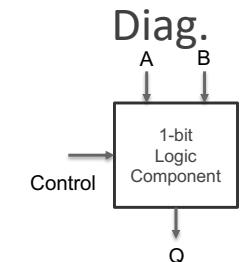
Remember

WeChat: cstutorcs

- inputs A, B are only one bit!
- output Y is also one bit!

Logic Component Design

Schematic



Four gates and MUX to select the logic operation

- 2-bit MUX (2 Bool Bits = Select Bits)
- Supports 4 logic operations

Assignment Project Exam Help
<https://tutorcs.com>

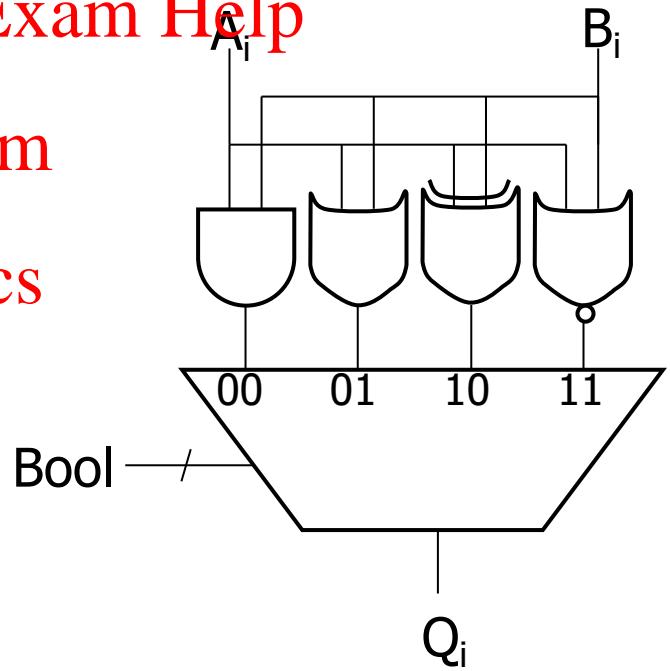
Input bits:

- A and B: 1 bit each
- Bool: 2 bits

WeChat: cstutorcs

Output bits:

- Q: 1 bit



How many of logic components are needed for a n-bit design?



Control and Datapath

AND operation

- Bool bits: 00

Assignment Project Exam Help

OR operation

- Bool bits: 01

<https://tutorcs.com>

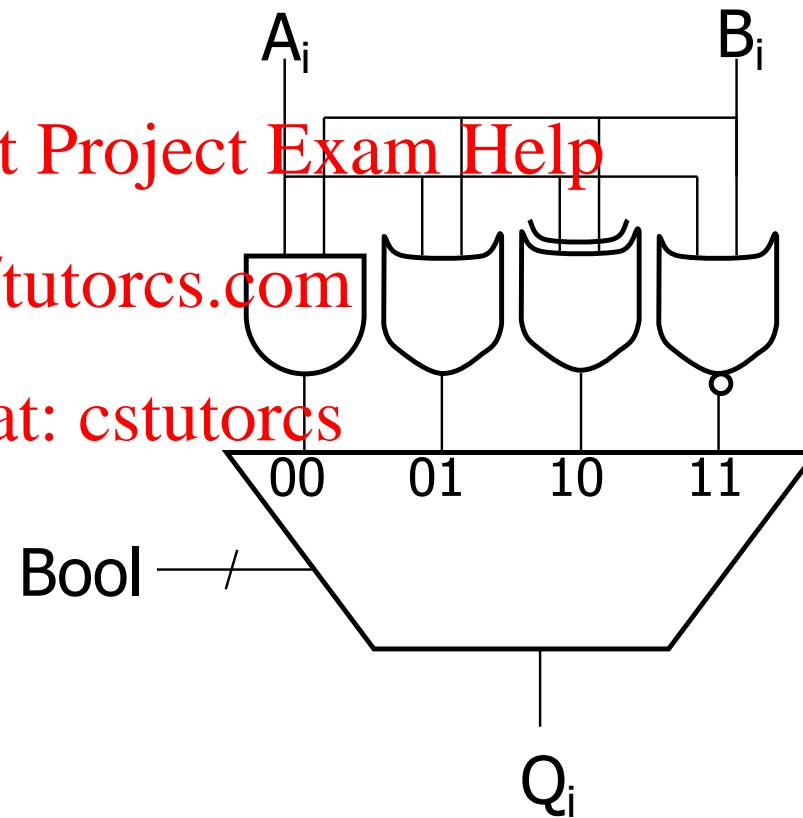
XOR operation

- Bool bits: 10

WeChat: cstutorcs

NOR operation

- Bool bits: 11



Assignment Project Exam Help

Arithmetic and Logic Unit

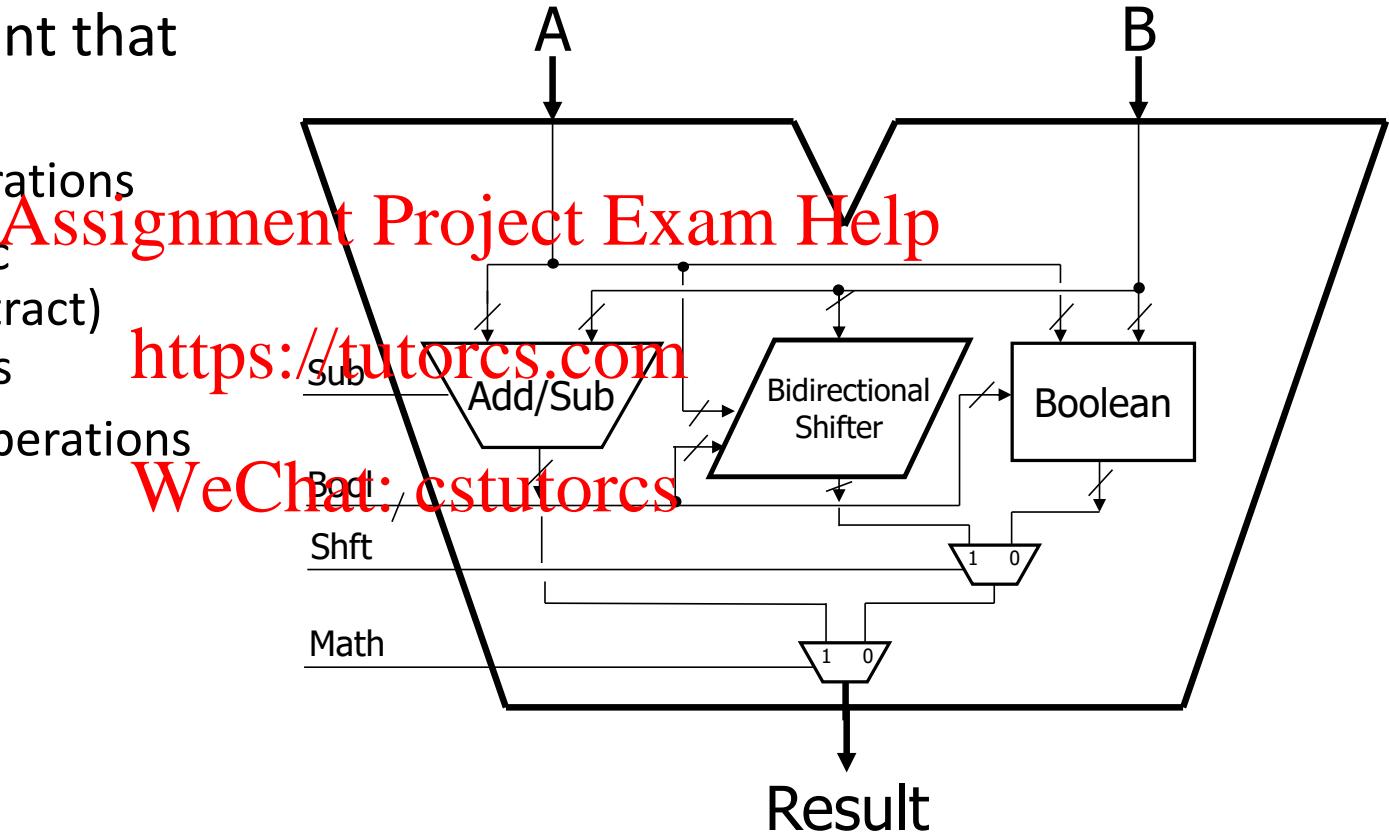
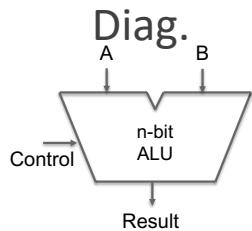
ALU

Overview

One component that performs

- Logic operations
- Arithmetic (Add/Subtract) operations
- Bit shift operations

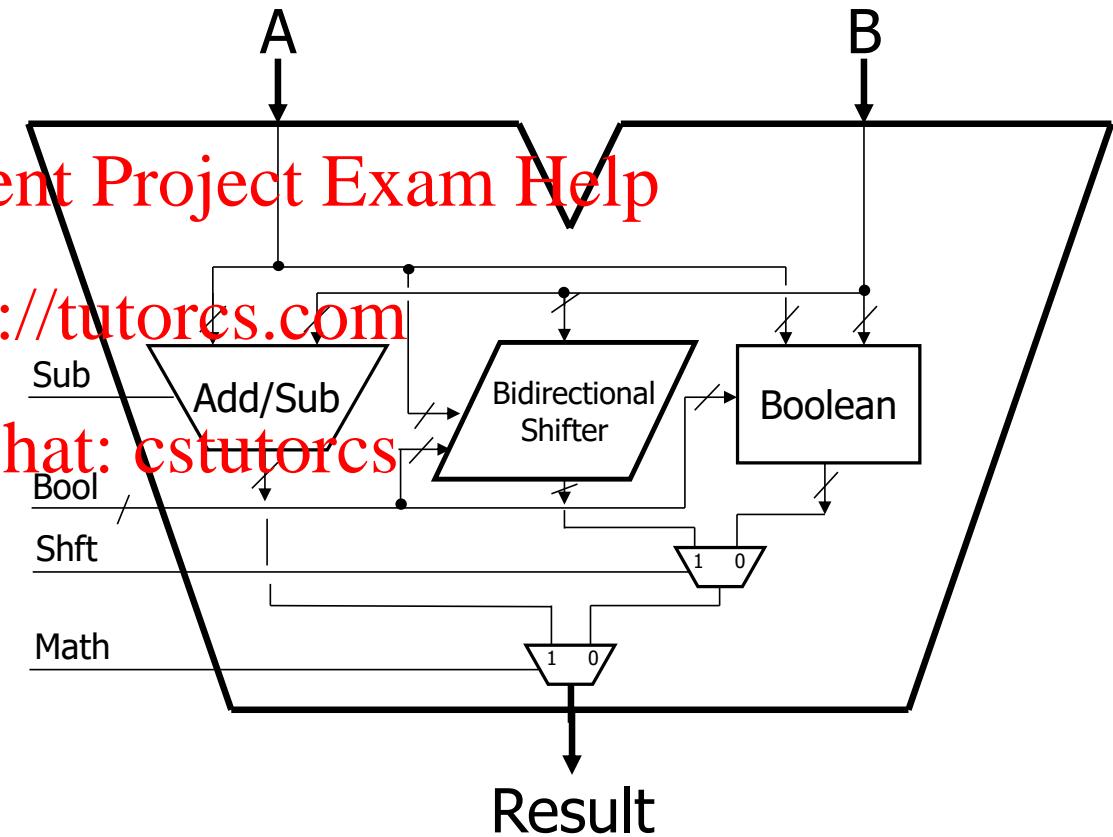
Schematic



Inputs and Output

Inputs to ALU

- A, B: n-bits each
- 5 functions bits
 - Sub: 1 bit
 - Bool: 2 bits
 - Shft: 1 bit
 - Math: 1 bit

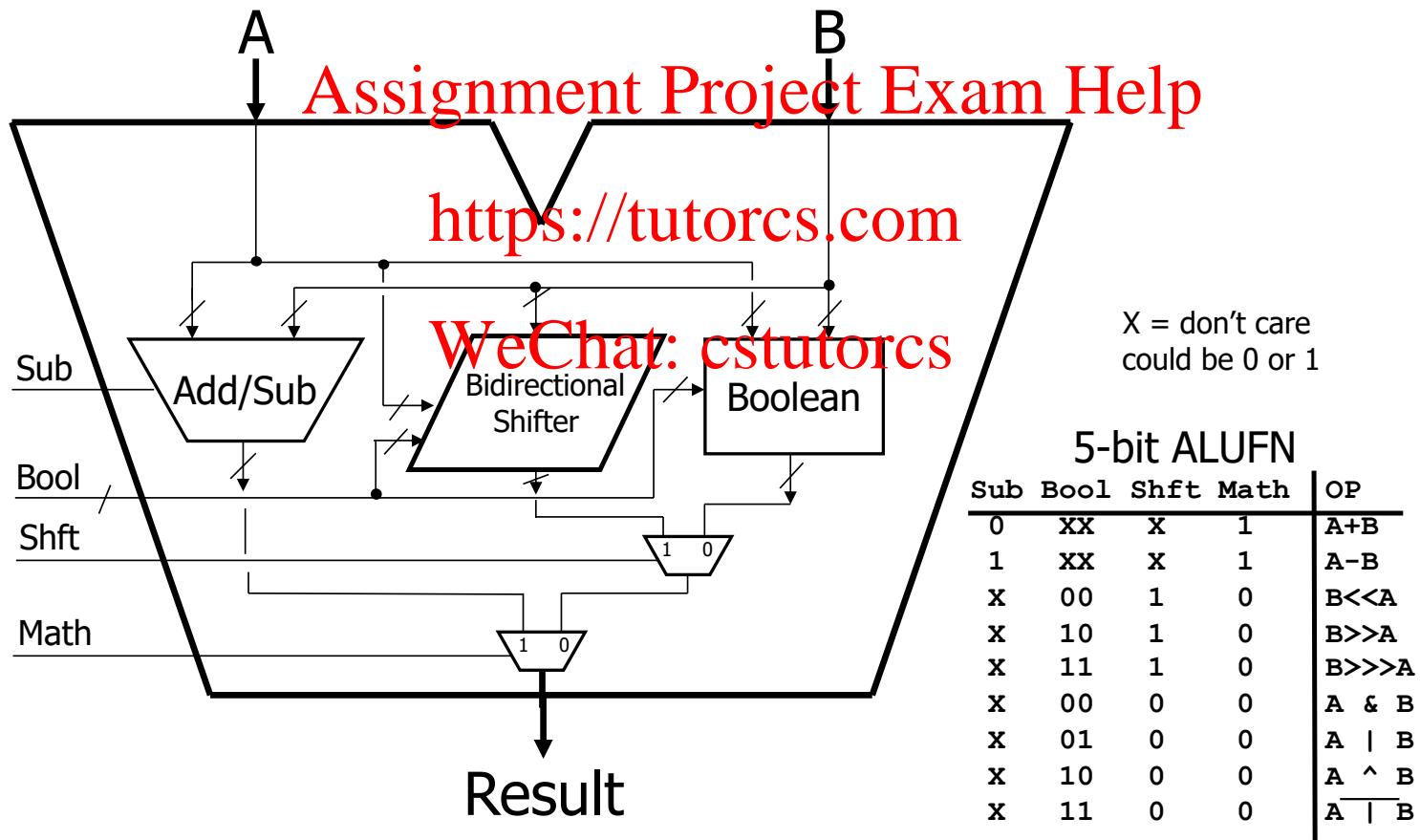


Output of ALU

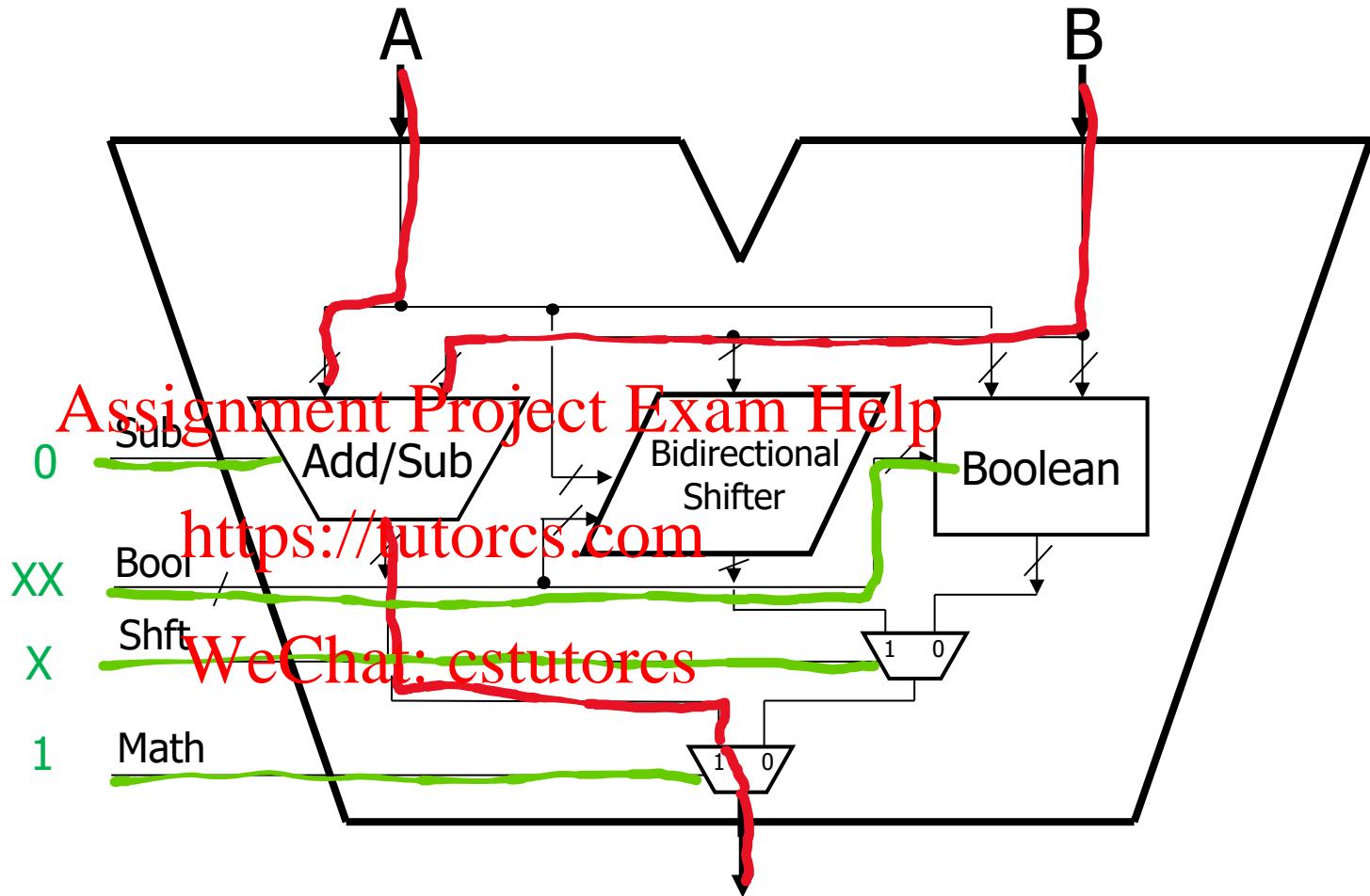
- Result: n-bits

ALU Function Table

Configure ALU: 5 function bits



$A+B$

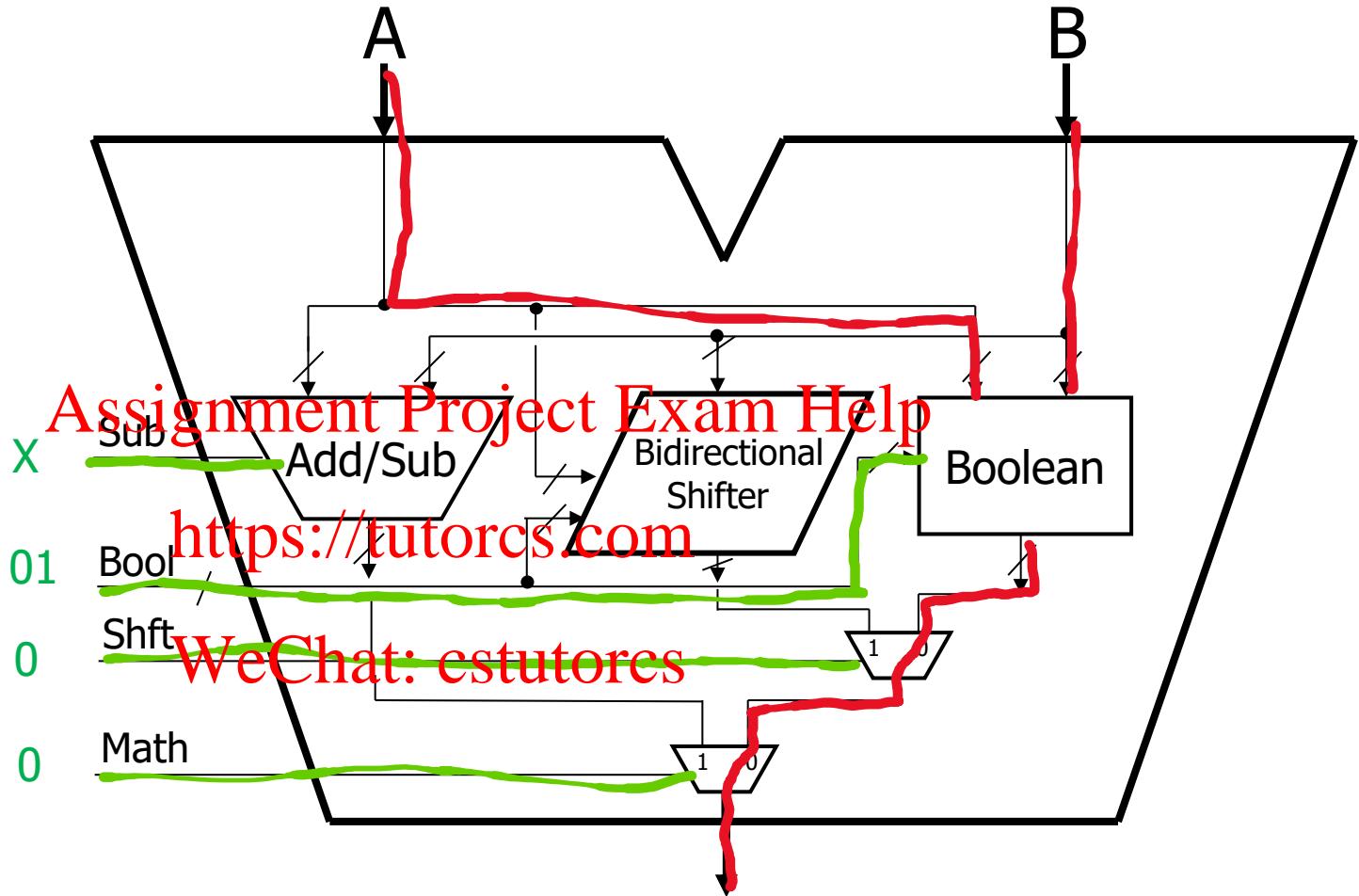


5-bit ALUFN				OP
Sub	Bool	Shft	Math	
0	XX	X	1	A+B
1	XX	X	1	A-B
x	00	1	0	B<<A
x	10	1	0	B>>A
x	11	1	0	B>>>A
x	00	0	0	A & B
x	01	0	0	A B
x	10	0	0	A ^ B
x	11	0	0	A B

Red = Data Path
Green = Configuration

Result

A | B



5-bit ALUFN

Sub	Bool	Shft	Math	OP
0	XX	X	1	A+B
1	XX	X	1	A-B
x	00	1	0	B<<A
x	10	1	0	B>>A
x	11	1	0	B>>>A
x	00	0	0	A & B
x	01	0	0	A B
x	10	0	0	A ^ B
x	11	0	0	A B

Red = Data Path
Green = Configuration

Result