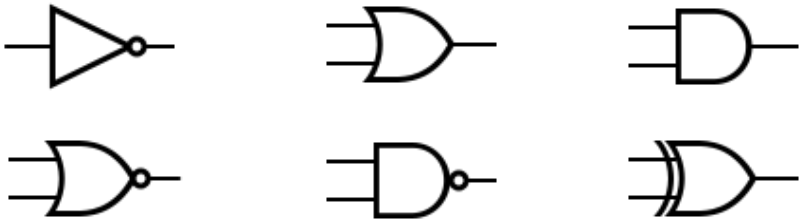
CS 261
Fall 2016

Mike Lam, Professor

Combinational Circuits

Quiz

- Match the gates with their truth tables and write the boolean function name



	0	1
0	0	1
1	1	1

0	1
1	0

	0	1
0	1	0
1	0	0

	0	1
0	0	0
1	0	1

	0	1
0	0	1
1	1	0

	0	1
0	1	1
1	1	0

Logic gates

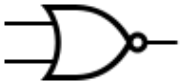
- Primary gates:

|	0	1
0	0	1
1	1	1

OR

!	
0	1
1	0

NOT



	0	1
0	1	0
1	0	0

NOR

&	0	1
0	0	0
1	0	1

AND

^	0	1
0	0	1
1	1	0

XOR

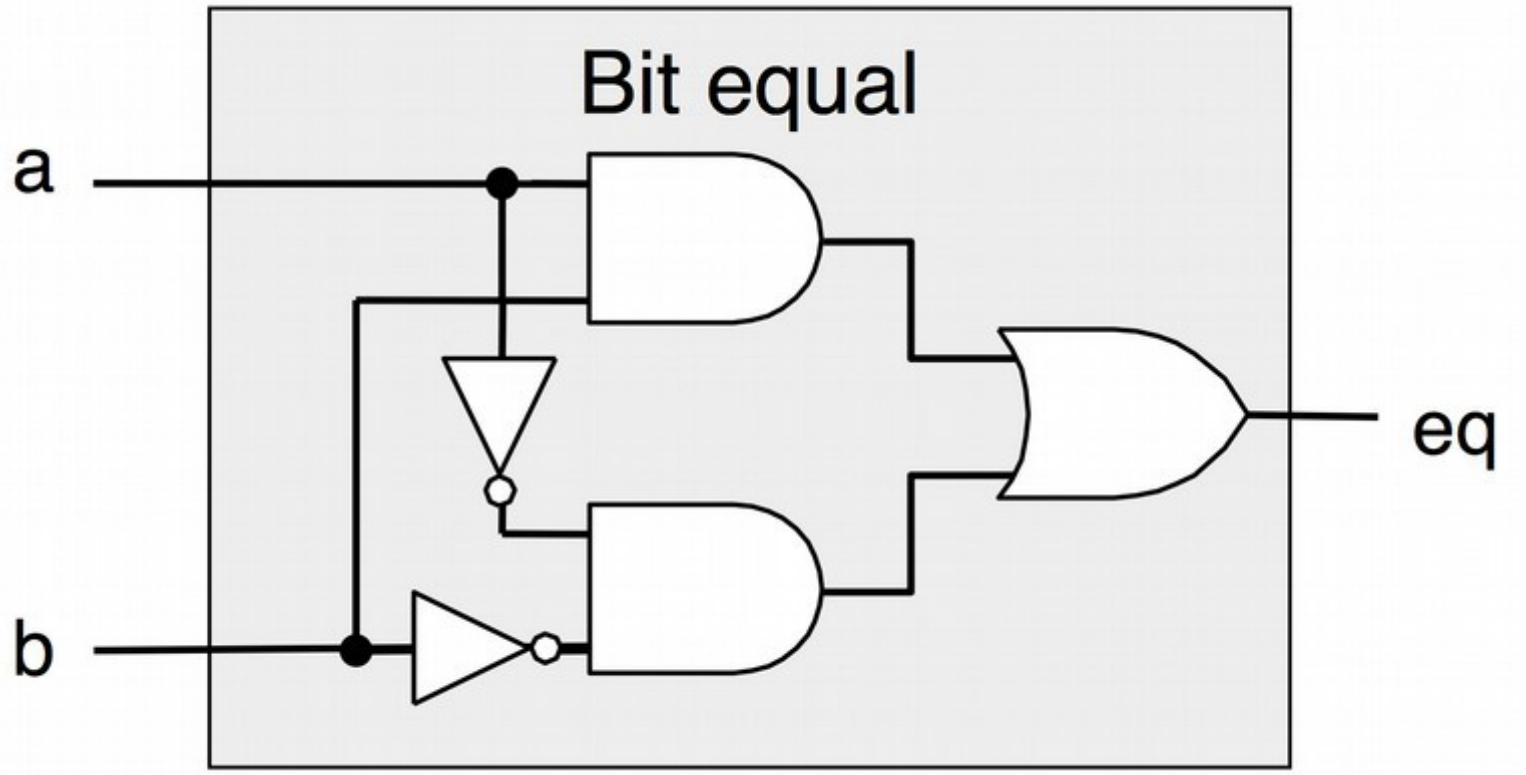
	0	1
0	1	1
1	1	0

NAND

Circuits

- Circuits are formed by linking gates together
  - Inputs and outputs
    - Link output of one gate to input of another
    - Some gates have multiple inputs and/or outputs
  - Combinational circuits: outputs are a boolean function of inputs
    - Not time-dependent
    - Used for computation
  - Sequential circuits: output is dependent on previous inputs
    - Time-dependent
    - Used for memory

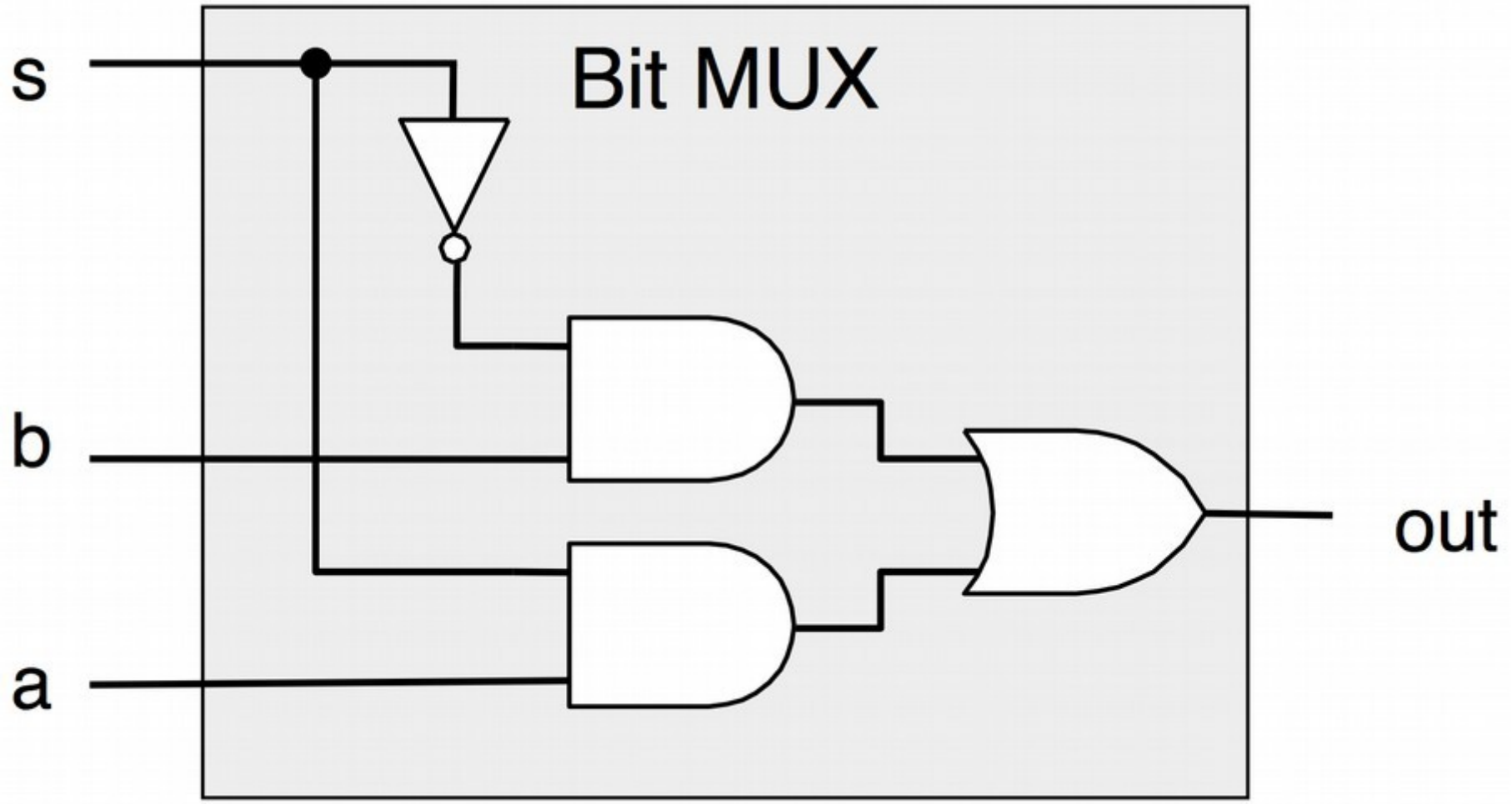
Equality



a EQ b = (a & b) | (!a & !b)

EQ(a, b) = OR(AND(a, b), AND(NOT(a), NOT(b)))

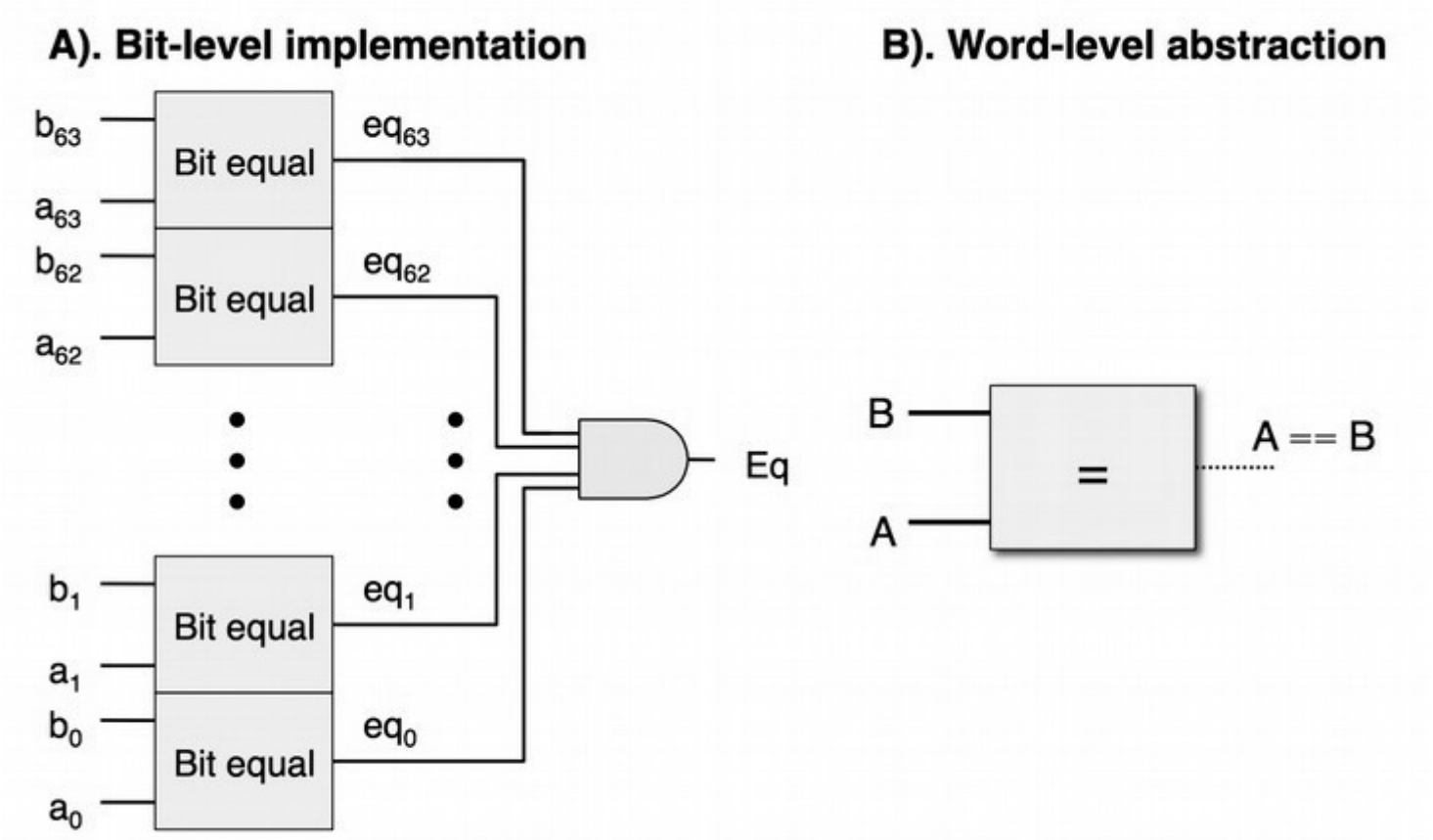
Multiplexor (“selector”)



MUX (a, b, s) = (s & a) | (!s & b)

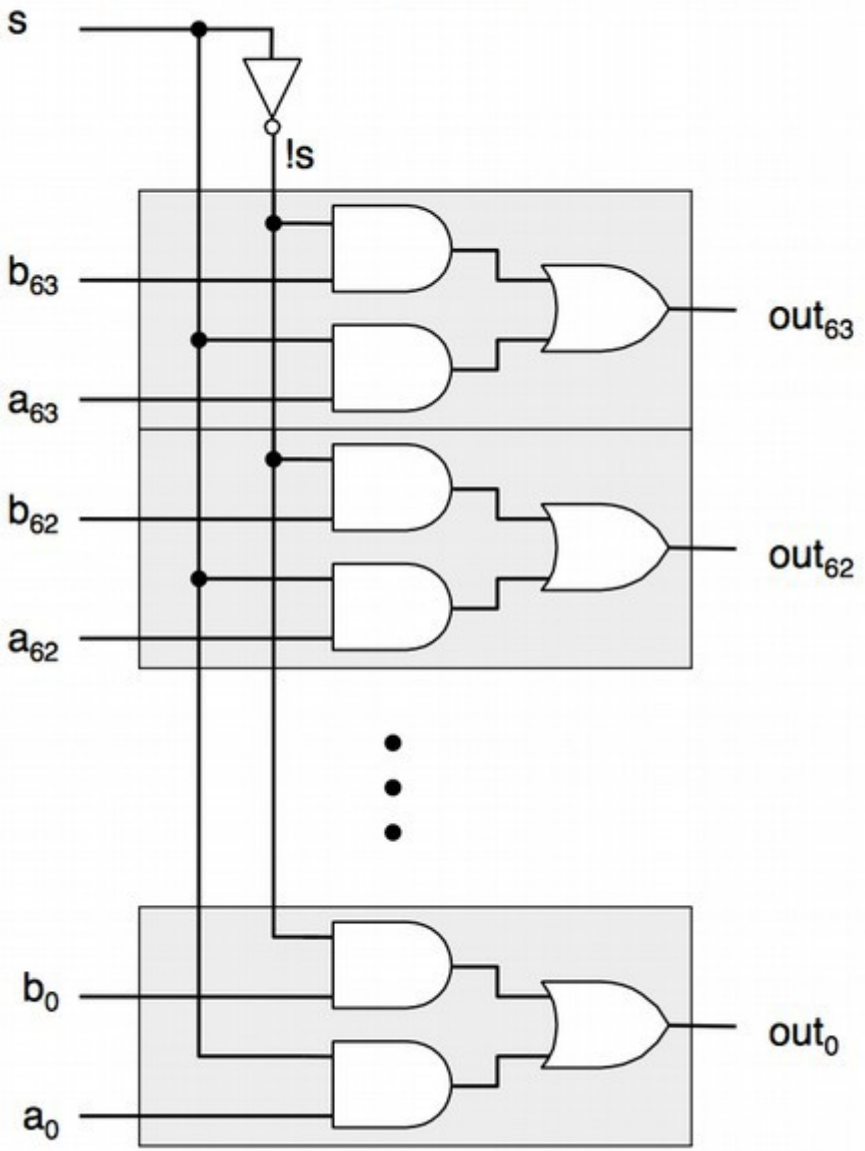
Abstraction

- Name circuits, then use them to build more complex circuits
  - E.g., use bit-level EQ to build a word-level equality circuit:

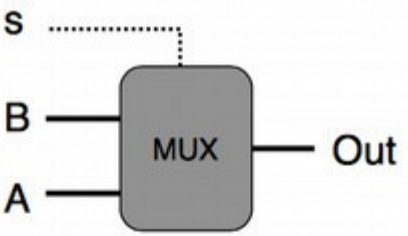


Word-level 2-way multiplexer

A). Bit-level implementation

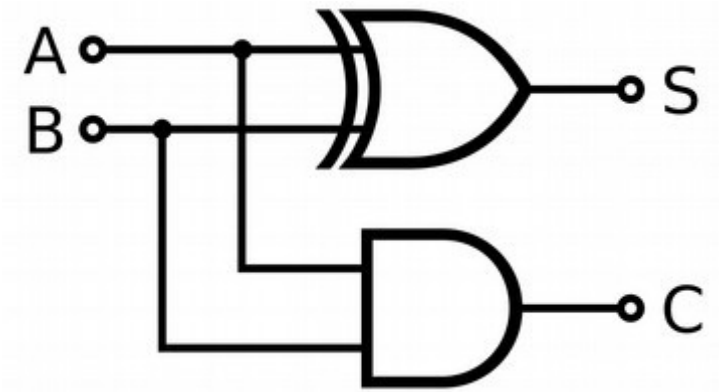


B). Word-level abstraction




```
int Out = [
  s : A;
  1 : B;
];
```

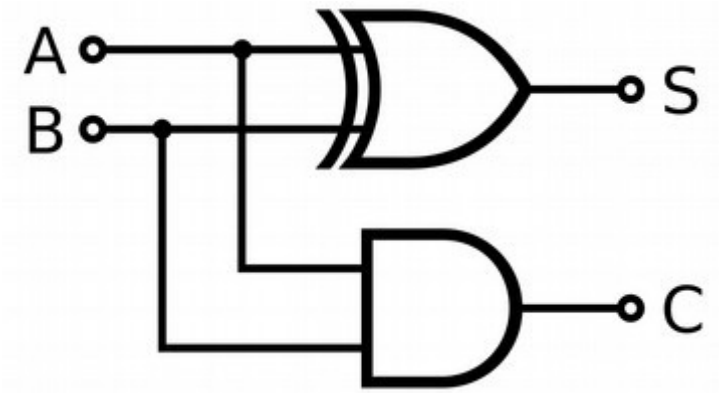
Half adders



Half Adder

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

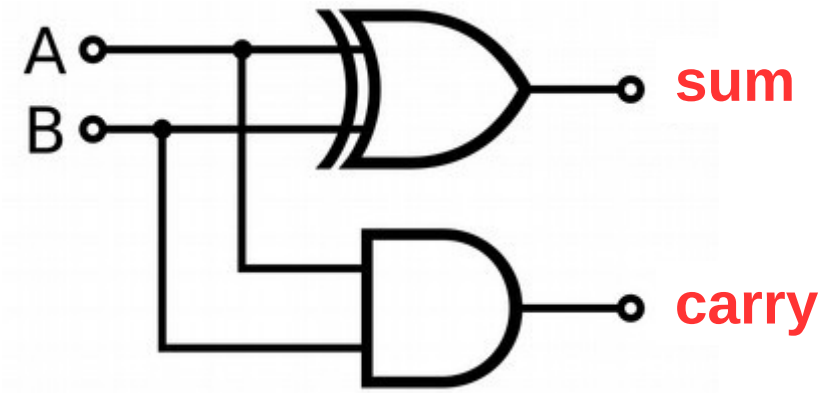
Half adders



Half Adder

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

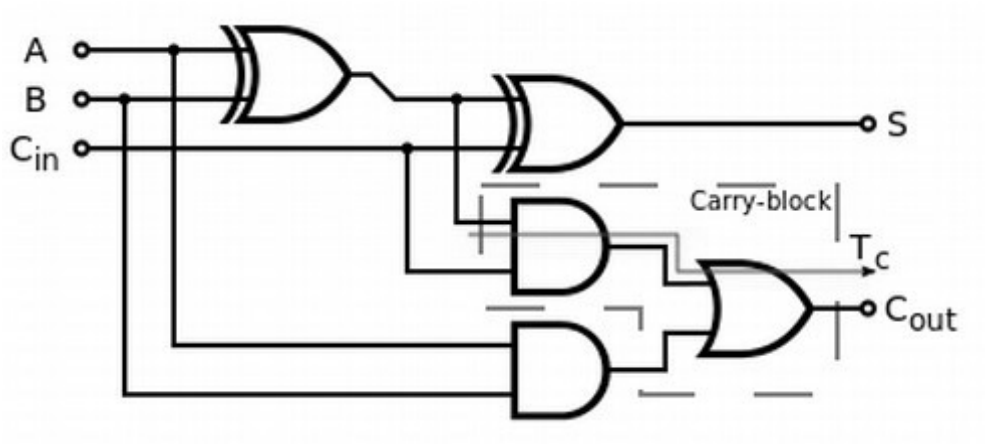
Half adders



Half Adder

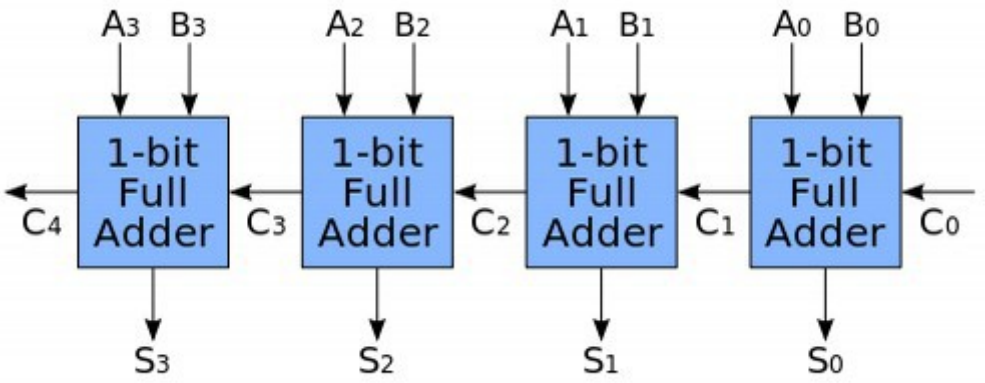
A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Full adders

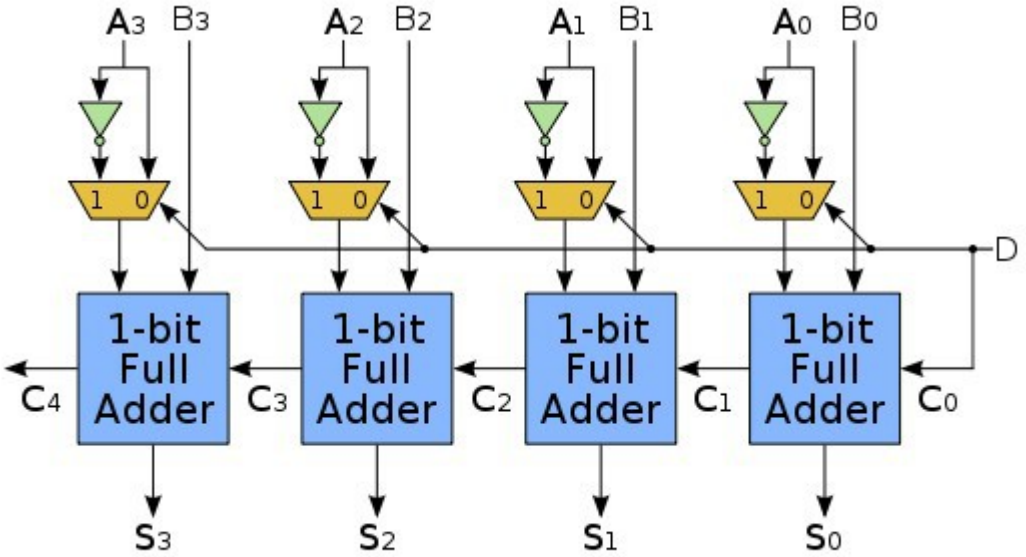


Full Adder

Connect full adders to build a ripple-carry adder
that can handle multi-bit addition:



Adder/subtractor



D=0: Addition
D=1: Subtraction

In two's complement: **B – A = B + !A + 1**