CSc 106 Fall 2012

Lab 3: Logic and Circuits, Sorting and Running times

Boolean Algebra review

- Boolean Operators:
  - AND $\wedge$
  - OR $\vee$
  - NOT $\neg$
  - XOR $\oplus$
  - NAND
  - NOR

Order of Operations

BNAO

1. Brackets
2. Not
3. And
4. Or

AND, OR, XOR, NOT Truth Tables

AND

A	B	A∧B
0	0	0
0	1	0
1	0	0
1	1	1

OR

A	B	A∨B
0	0	0
0	1	1
1	0	1
1	1	1

XOR

A	B	A⊕B
0	0	0
0	1	1
1	0	1
1	1	0

NOT

A	¬A
0	1
1	0

NAND, NOR Truth Tables

AND

A	B	A∧B
0	0	0
0	1	0
1	0	0
1	1	1

OR

A	B	A∨B
0	0	0
0	1	1
1	0	1
1	1	1

NAND

A	B	¬(A∧B)
0	0	1
0	1	1
1	0	1
1	1	0

NOR

A	B	¬(A∨B)
0	0	1
0	1	0
1	0	0
1	1	0

¬(A OR B)

What is the answer if: A=1, B=0?

A. 0

B. 1

Quick Quiz
¬(A OR B)

What is the answer if: A=1, B=0?

A. 0

B. 1

ANSWER: 0

(¬ A XOR B)

What is the answer if: A=1, B=1?

A. 0

B. 1

Quick Quiz
(¬ A XOR B)

What is the answer if: A=1, B=1?

A. 0

B. 1

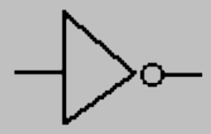
ANSWER: 1

Boolean Operators and Gates

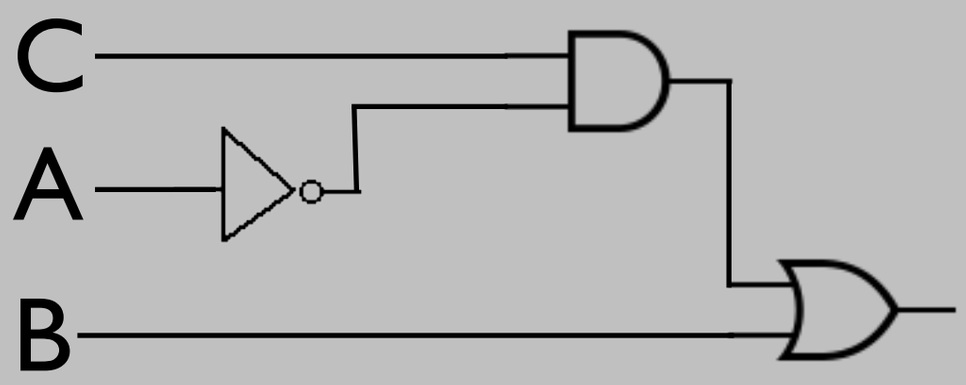
- Each Operator can be physically built as a gate to build more complex circuits
- We can represent these gates as symbols to draw circuits

Logic Gates

AND 

NOT 

OR 

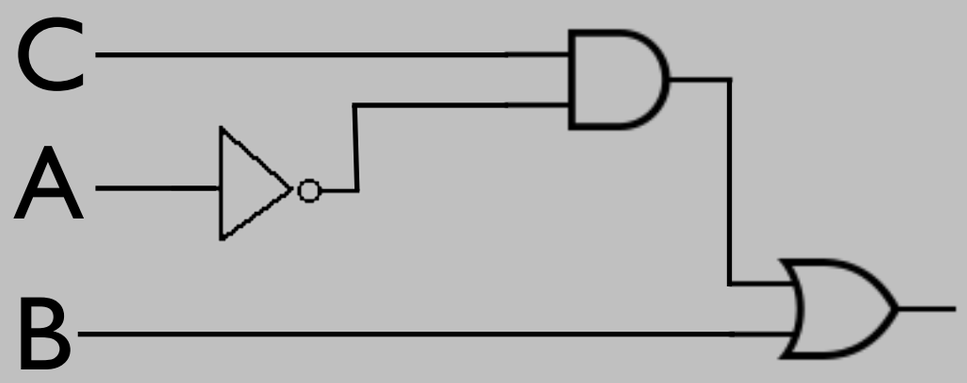


Boolean Formulas and Circuits

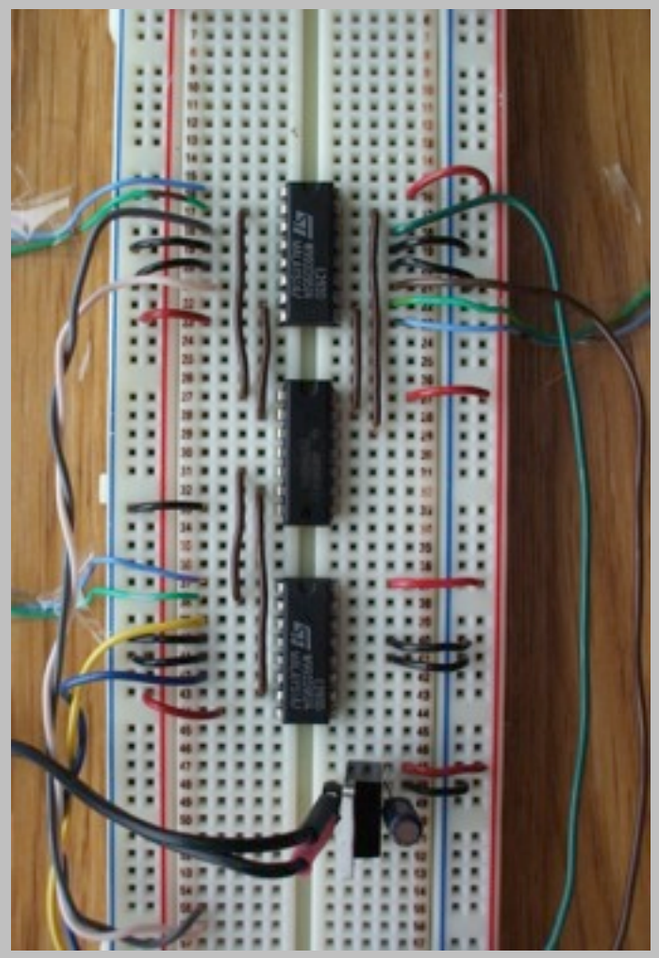
Boolean Formula

$B \vee (C \wedge \neg A)$

Circuit: Representation



Circuit: Breadboard*



* http://webhome.csc.uvic.ca/~mcheng/samples/hoole/wiring.html

Translating from Formula to Circuit

- $(A \wedge B) \vee (C \wedge \neg A)$
  - each operator is a gate

Translating from Formula to Circuit

- $(A \wedge B) \vee (C \wedge \neg A)$
  - each operator is a gate
  - each variable is an input

Translating from Formula to Circuit

- $(A \wedge B) \vee (C \wedge \neg A)$
  - each operator is a gate
  - each variable is an input
- lines represent wires that connect them

$$(A \wedge B) \vee (C \wedge \neg A)$$

Let's draw the circuit representation of this boolean formula

$$(A \wedge B) \vee (C \wedge \neg A)$$

First: inputs

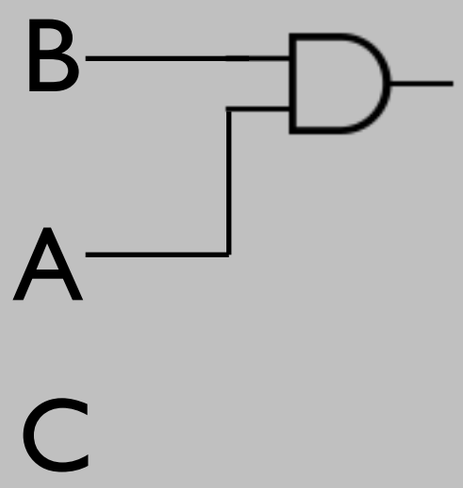
B

A

C

$$(A \wedge B) \vee (C \wedge \neg A)$$

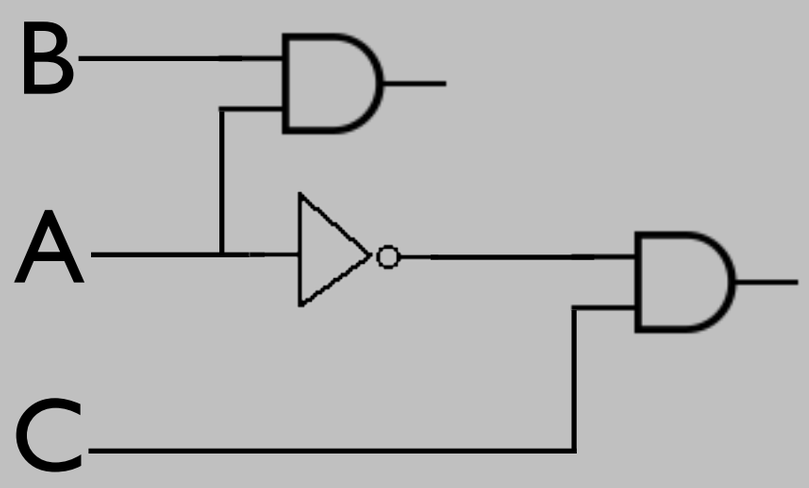
$(A \wedge B)$



$$(A \wedge B) \vee (C \wedge \neg A)$$

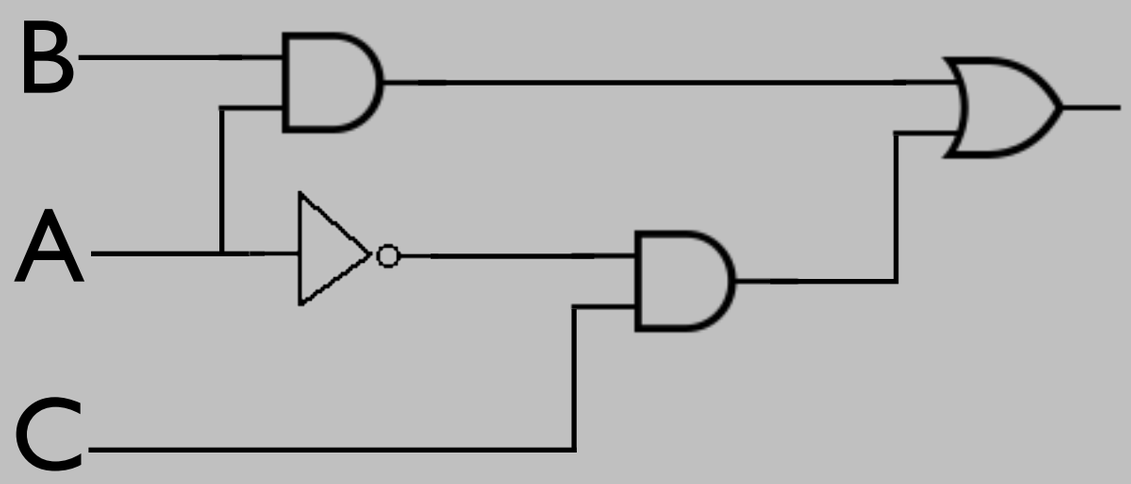
$(A \wedge B)$

$(C \wedge \neg A)$

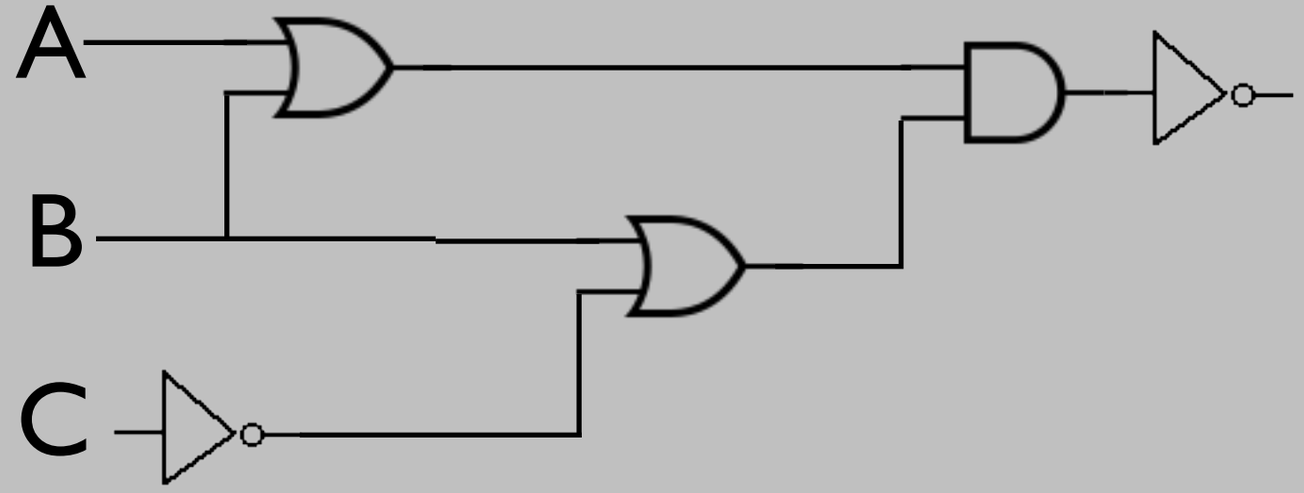


$$(A \wedge B) \vee (C \wedge \neg A)$$

$$(A \wedge B) \vee (C \wedge \neg A)$$

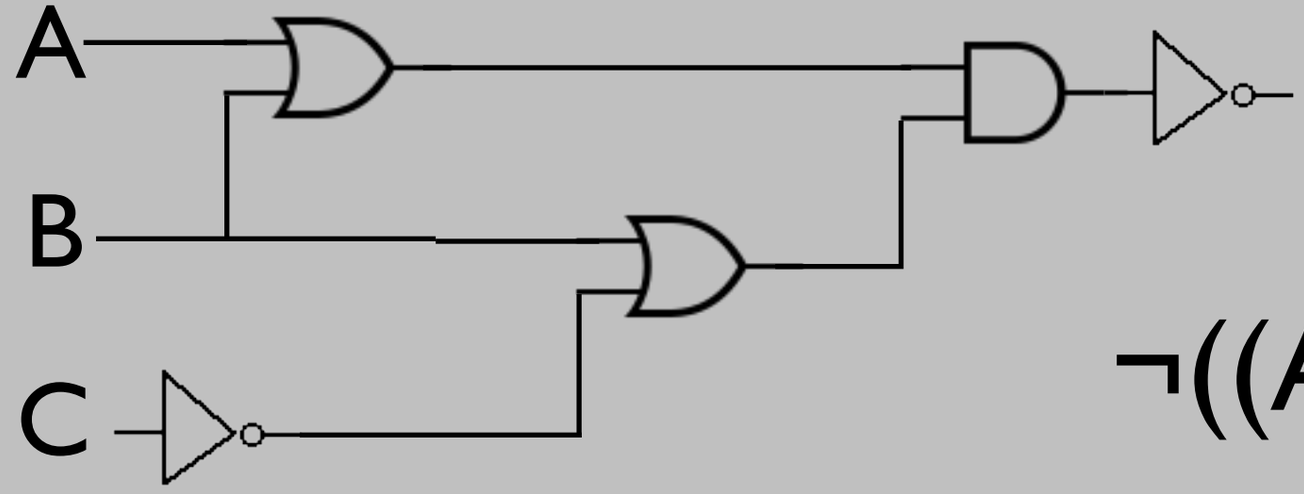


Translating from Circuit to Formula



Work it out...

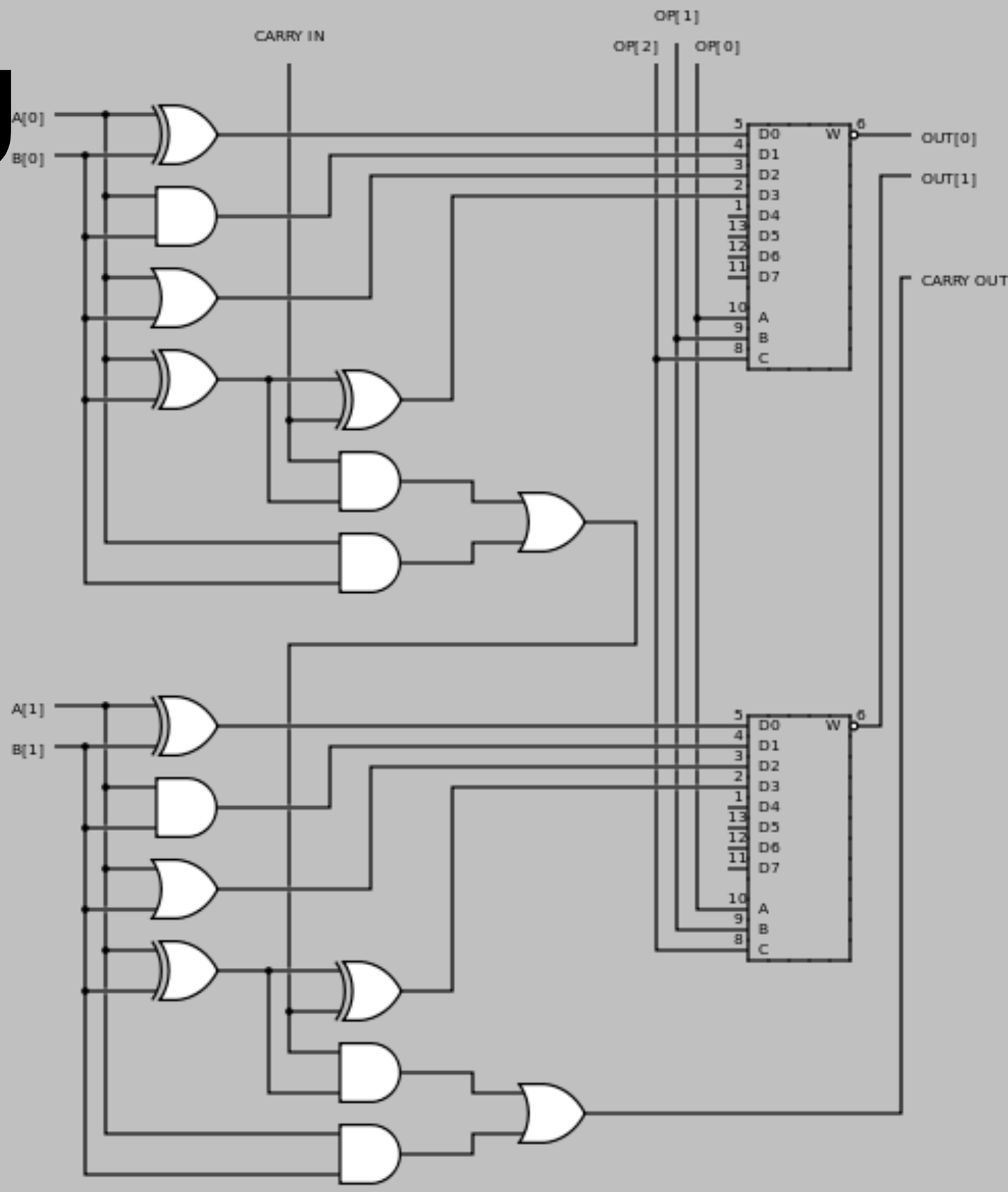
Translating from Circuit to Formula



$\neg((A \vee B) \wedge (\neg C \vee B))$

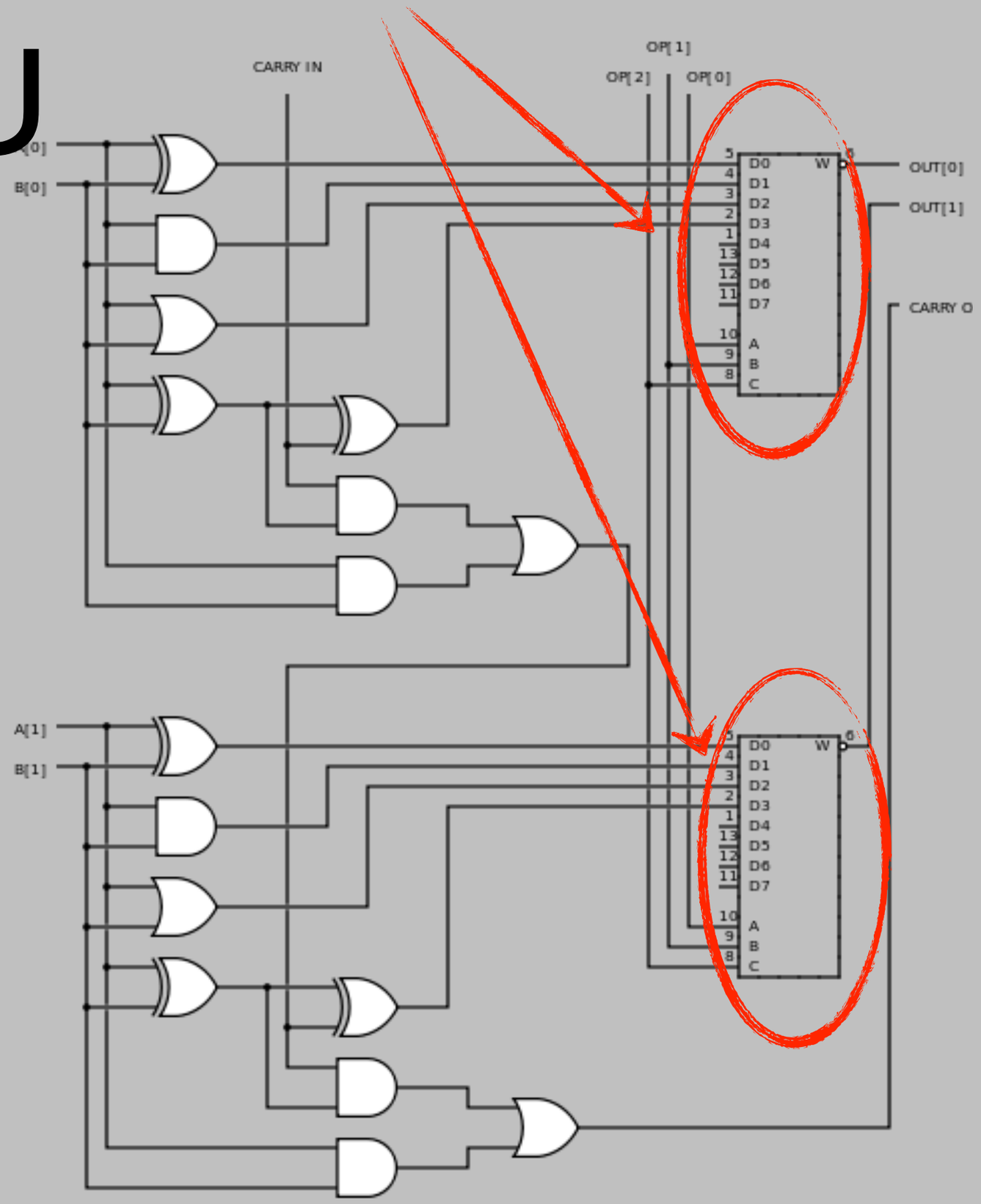
Simple ALU

- 2 2-bit inputs + carry
- control input
- 2-bit output + carry
- can perform AND, OR, XOR and addition



Simple ALU

Multiplexors



- 2 2-bit inputs + carry
- control input
- 2-bit output + carry
- can perform AND, OR, XOR and addition