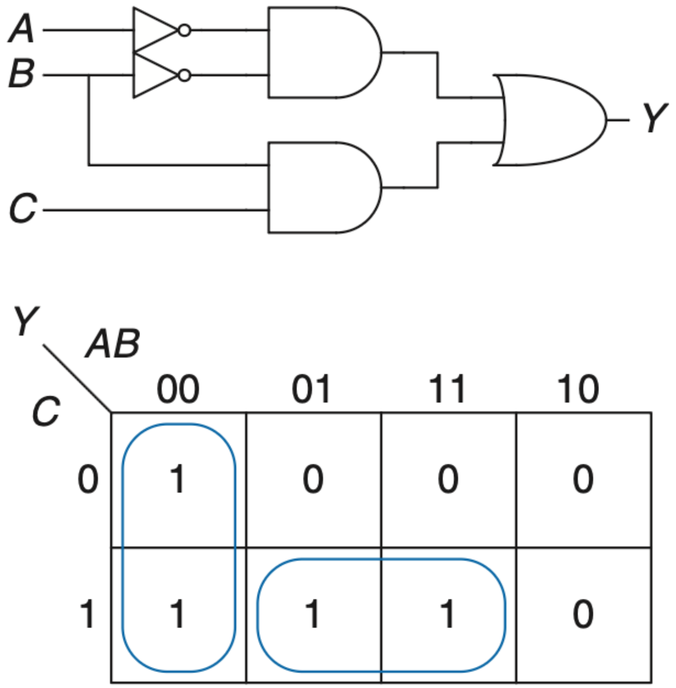
ENR-325/325L Principles of Digital Electronics and Laboratory

Xiang Li

Fall 2025

Coe College

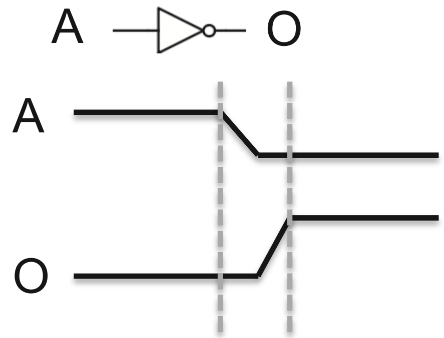
K-map is for Boolean logics, not engineering



$Y = \overline{A}\,\overline{B} + BC$

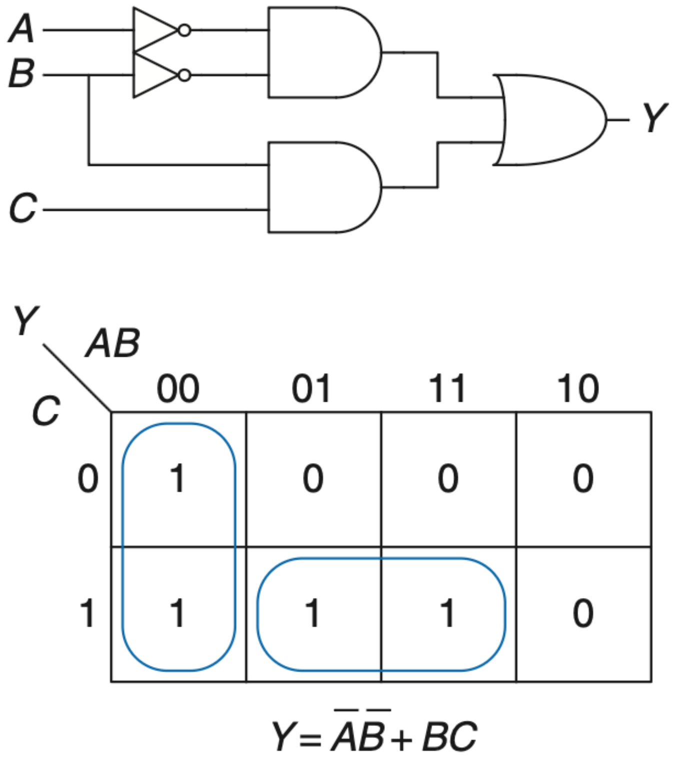
Figure 2.75 Circuit with a glitch

Why glitch? Because in real-lift every flip takes time:



Example from: Haris, D., and S. Haris. "Digital Design and Computer Architecture ARM Edition." (2016).

K-map is for Boolean logics, not engineering



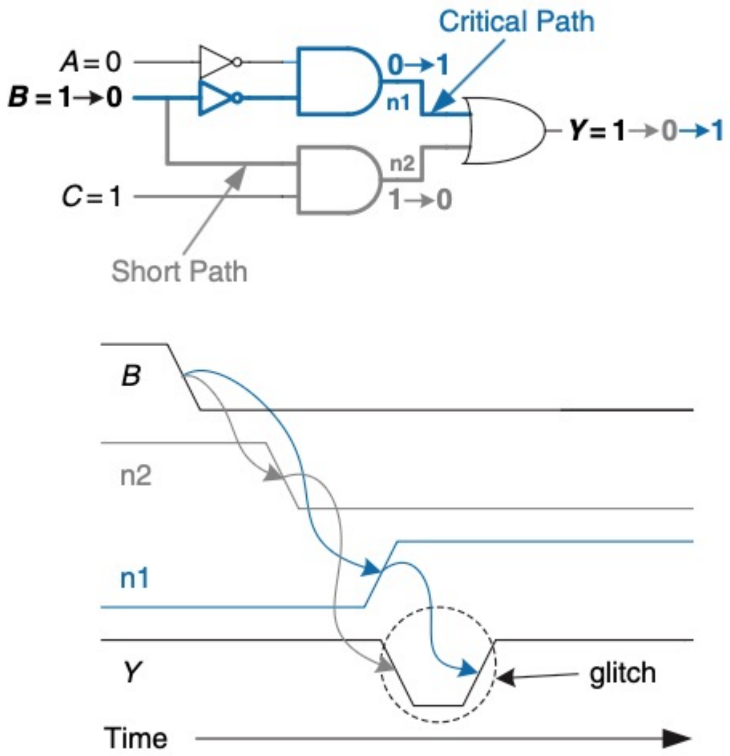


Figure 2.75 **Circuit with a glitch**

Example from: Haris, D., and S. Haris. "Digital Design and Computer Architecture ARM Edition." (2016).

Hamming codes can be done in the EE way

- Before that, we need to acquire some basic skillsets.

Pre-step: Data forms

Step 1: Data manipulation

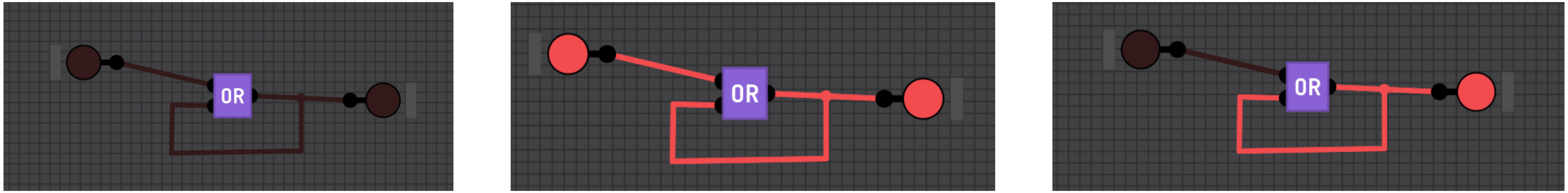
Step 2: Information storage

Step 3: Interface

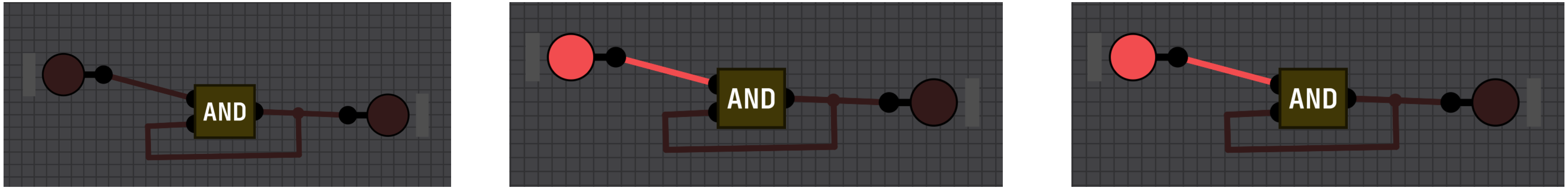
Step 1: data manipulation, continued

- Self-looped gates -> latches

Once on, always on!

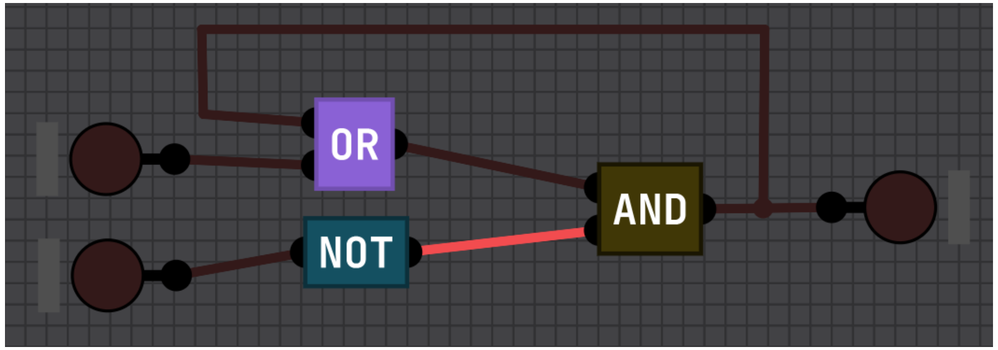


Always off, meh.

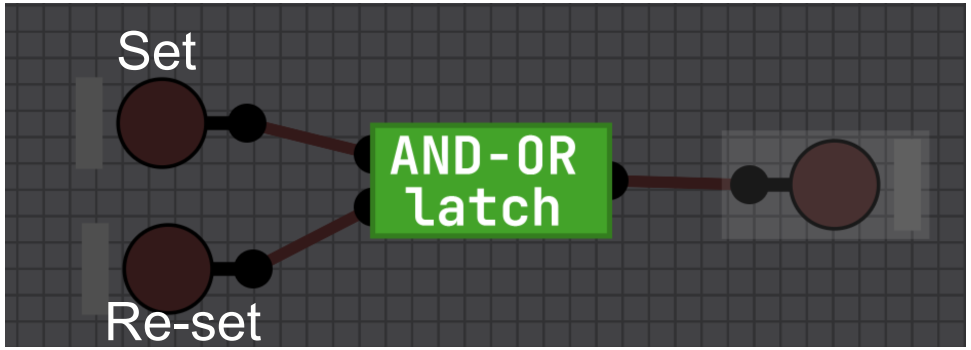


Step 1: data manipulation, continued

- Make the AND-OR latch a bit useful: it stores(writes/clears) 1 bit, kind a.

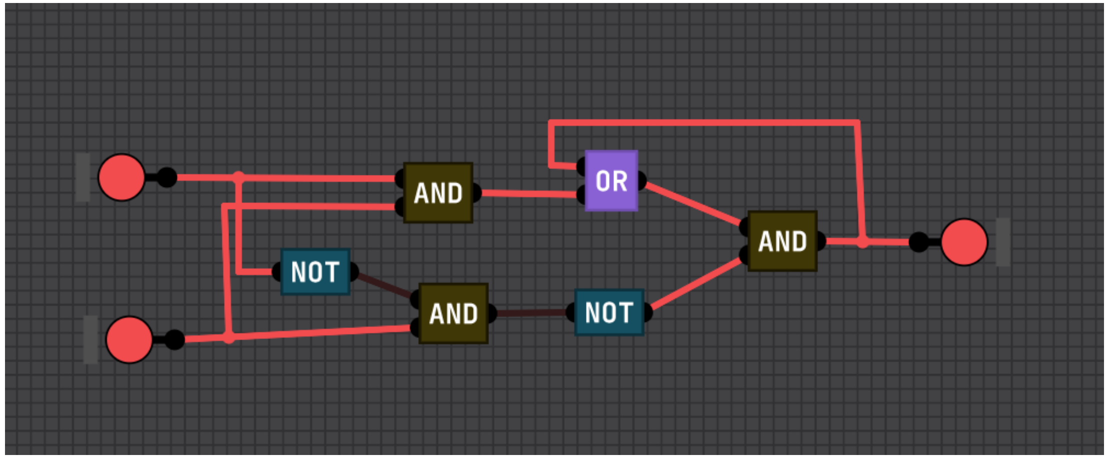


An abstraction of AND-OR latch

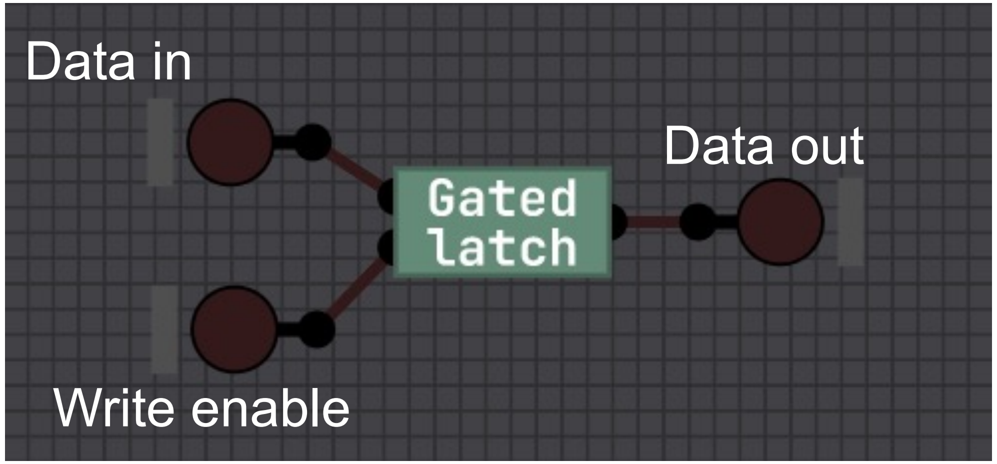


Step 2: Detour to information storage

This is a Gated latch, it stores(writes/clears) 1 bit slightly better.

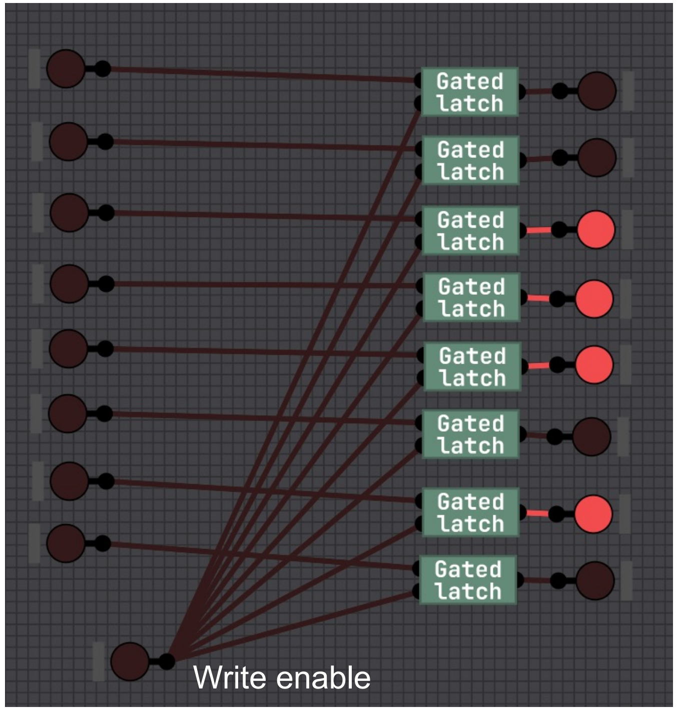


An abstraction of gated latch.



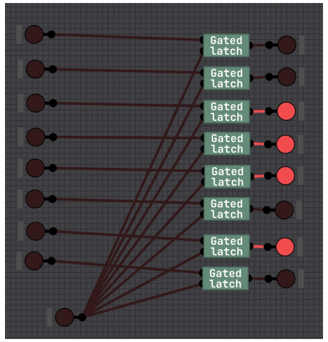
Step 2: Information storage, all done!

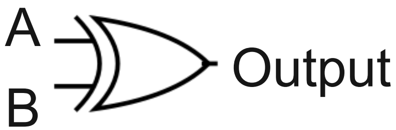
Scaling up to 8, we got an 8 bit storage (register).



So how about Hamming decoder (15,11)?

Some digits → Parity check, still basically XOR → Identify which bit is flipped based on a n digit number → A flip function, which is another XOR



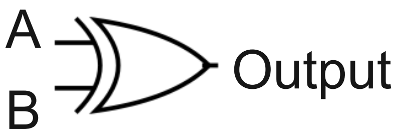


XOR gate truth table

Input		Output
A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

XOR gate can do so much more: addition

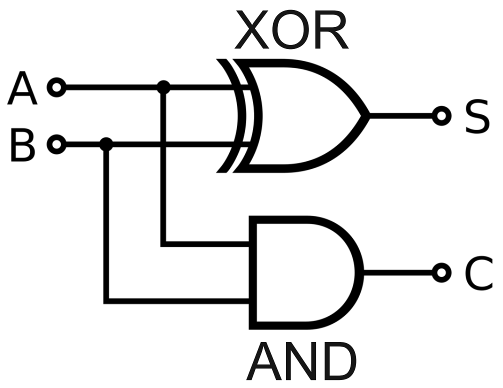
- XOR gate is almost what we need:



XOR gate truth table

Input		Output
A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

Half adder



Inputs		Outputs	
A	**B**	**C_{out}**	**S**
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

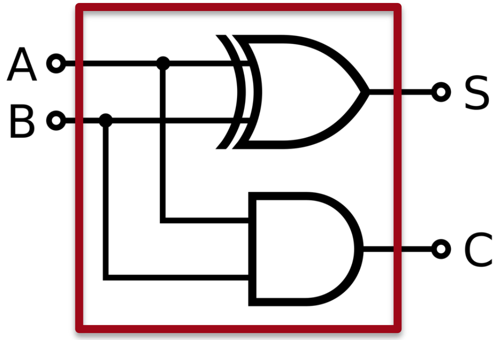
Coe College

https://en.wikipedia.org/wiki/Adder_(electronics)

Step 1: back to data manipulation

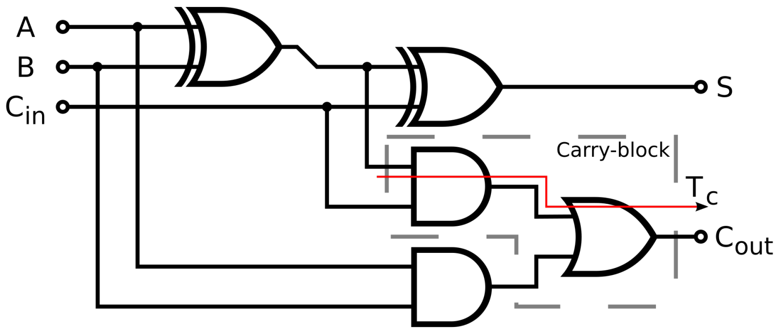
- Boolean operations: 3rd gen:

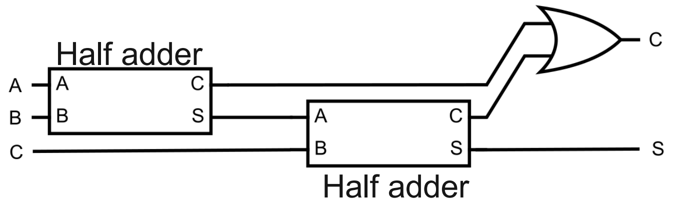
Half adder



Inputs		Outputs	
A	B	C_{out}	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Full adder



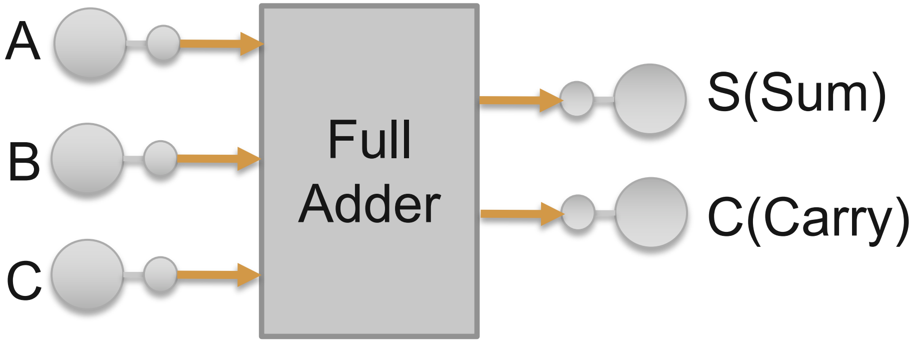


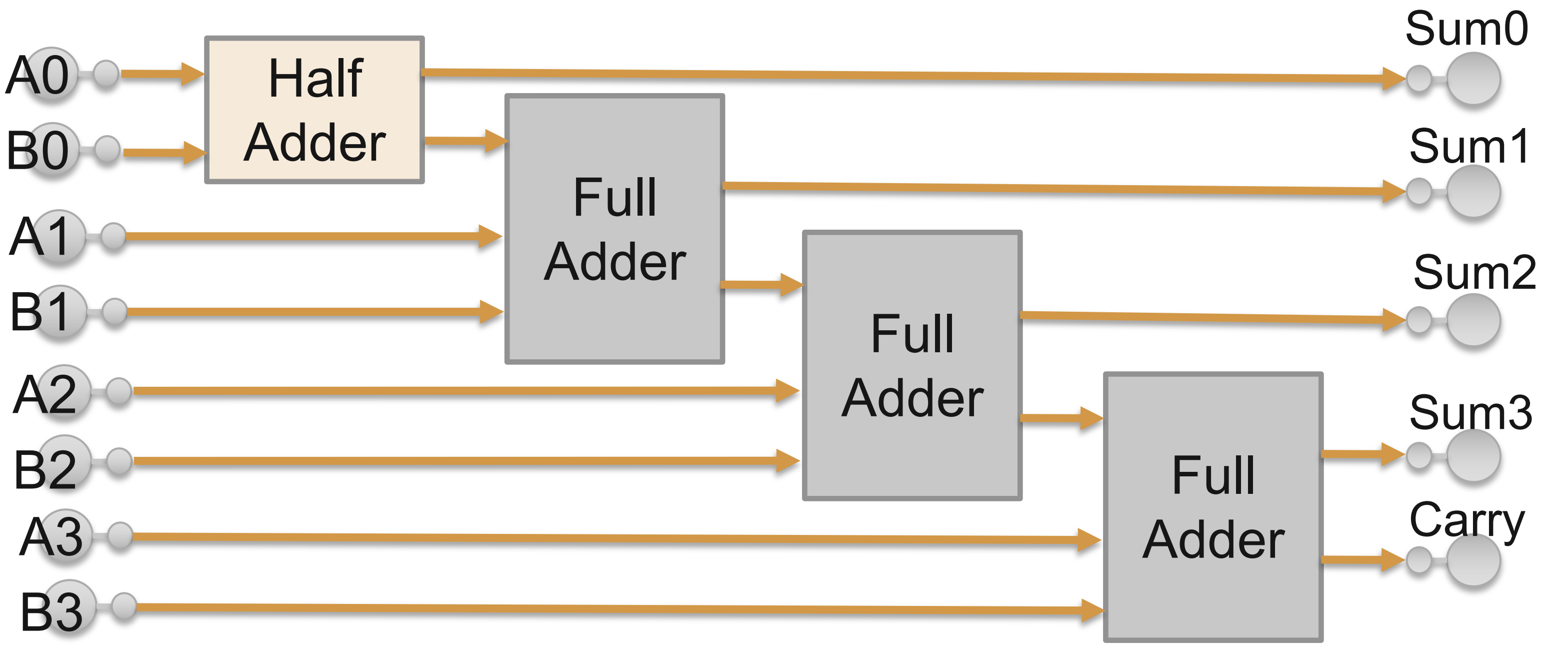
Inputs			Outputs	
A	B	C_{in}	C_{out}	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

COE COLLEGE

https://en.wikipedia.org/wiki/Adder_(electronics)

More abstraction: full adder circuits





Let's realize subtraction:

Step 1: create "negative number"

Step 2: add negative number

	1's compliment
-5	1010
-4	1011
-3	1100
-2	1101
-1	1110
-0	1111
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101

Just invert every 1 and 0, and it almost works.

1st bit no longer carry value, but signs

COE COLLEGE

Let's realize subtraction:

Step 1: create "negative number"

Step 2: add negative number

	1's compliment		2's compliment
-5	1010	+1	1011
-4	1011	+1	1100
-3	1100	+1	1101
-2	1101	+1	1110
-1	1110	+1	1111
-0	1111	+1	~~1~~0000
0	0000		0000
1	0001		0001
2	0010		0010
3	0011		0011
4	0100		0100
5	0101		0101

Sign bits

- **It works.**
- **Instead of subtraction, we do addition.**

A bit more about 2's compliment: Computing hardware defines data type

Table 3-1: Integer Types in Rust

Length	Signed	Unsigned
8-bit	`i8`	`u8`
16-bit	`i16`	`u16`
32-bit	`i32`	`u32`
64-bit	`i64`	`u64`
128-bit	`i128`	`u128`
architecture dependent	`isize`	`usize`

Three-bit integers

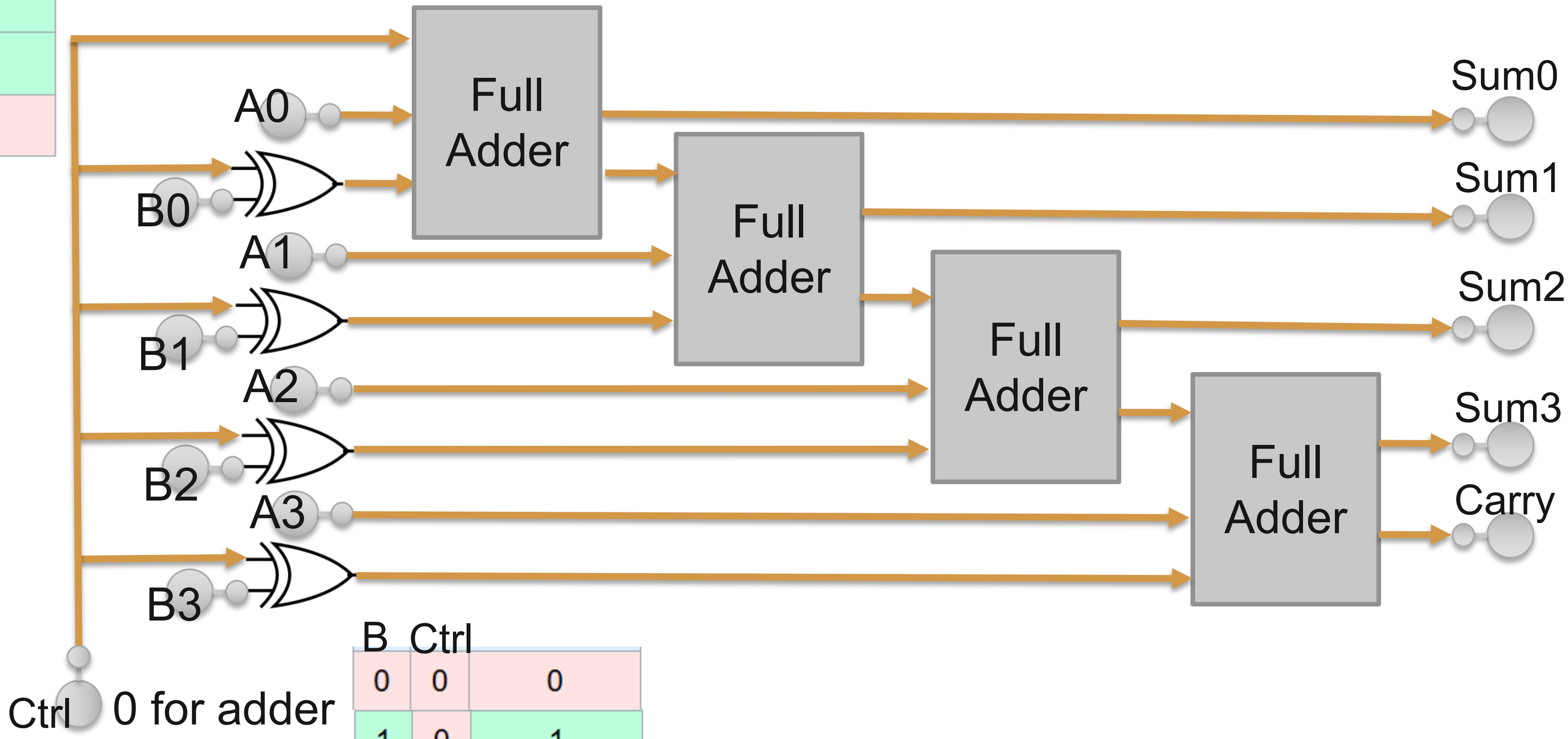
Bits	Unsigned value	Signed value (Two's complement)
000	0	0
001	1	1
010	2	2
011	3	3
100	4	−4
101	5	−3
110	6	−2
111	7	−1

https://en.wikipedia.org/wiki/Two%27s_complement

XOR gate truth table

Input		Output
A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

4-bit adder and subtractor



0 for adder

B	Ctrl	
0	0	0
1	0	1

1 for subtractor (also ignore Carry)

B	Ctrl	
0	1	1
1	1	0

COE COLLEGE

Step 3: Interface

- "Most of time, digital device failed at the interface." ---- one of my CSEE professor back in the old days.

Next week all the engineering stuff we are going to talk about is about interface:

- Between function and time
- Between logic function and hardware
- Even between hardware connections…

Step 3: The great interface theory

Massachusetts Institute of Technology

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Fundamentals of Systems Engineering

Prof. Olivier L. de Weck

Session 8
Systems Integration
Interface Management

Examples of Interfaces

Valve	mass flow $\dot{m}$	Tank
Rocket	momentum $F\Delta t$	Payload
Heat Exchanger	Heat flux $\dot{Q}$	Air
Solar Cell	Electrical power flux UI	Battery
web site (URL)	html file *data*	Browser
Motion Sensor	trigger *command*	Alarm
NPR Radio	News *cognitive*	Listener
Patient	feelings *affective*	Psycho-therapist

11

4 Canonical Types of Interfaces

- Physical Connection (always symmetric)
  - If A connects to B, B must also connect to A.
- Energy Flow
- Mass Flow
- Information Flow

12

COE COLLEGE