Week 11

CS2100 - L19 - Combinational Circuits

19.1 - Design methods

19.2 - Gate-level design

- Half-adder
- Full-adder

19.3 - Block-level design

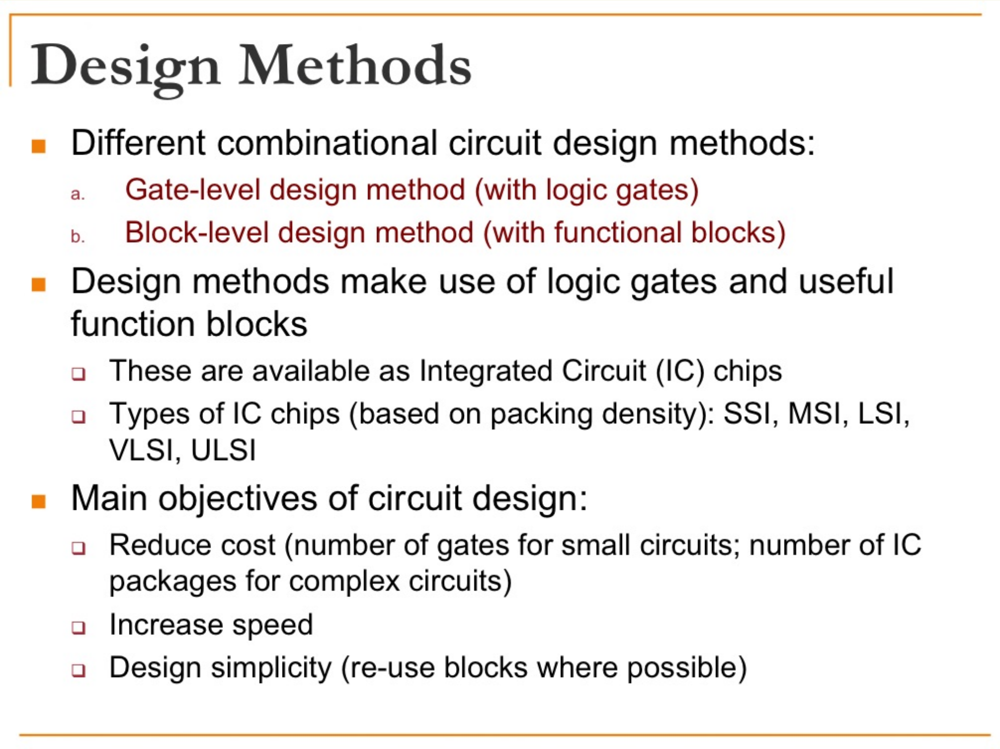
- Parallel adders
- Adder cum subtractor

19.4 - Arithmetic circuits

- Magnitude comparator

19.5 - Circuit delays

19.1 - Design methods

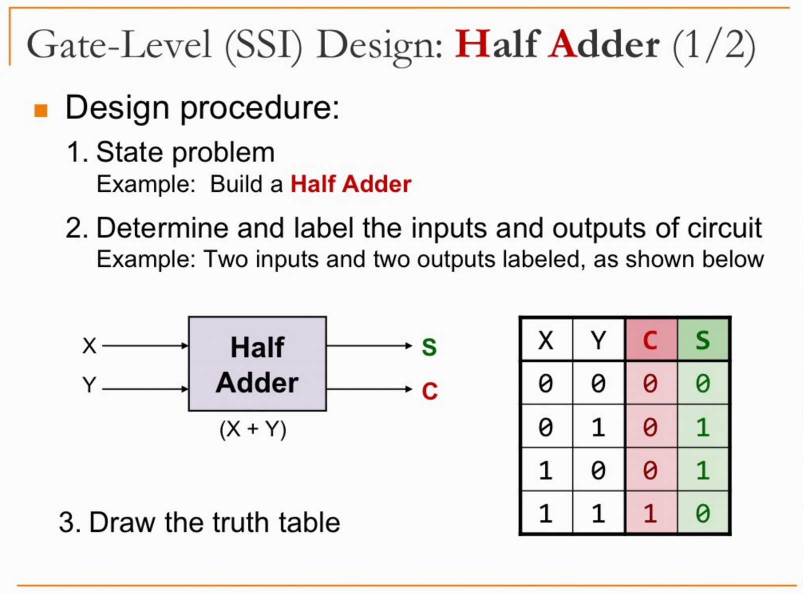


Design Methods

- Different combinational circuit design methods:
  a. Gate-level design method (with logic gates)
  b. Block-level design method (with functional blocks)
- Design methods make use of logic gates and useful function blocks
  - These are available as Integrated Circuit (IC) chips
  - Types of IC chips (based on packing density): SSI, MSI, LSI, VLSI, ULSI
- Main objectives of circuit design:
  - Reduce cost (number of gates for small circuits; number of IC packages for complex circuits)
  - Increase speed
  - Design simplicity (re-use blocks where possible)

19.2 - Gate-level design

Half-adder



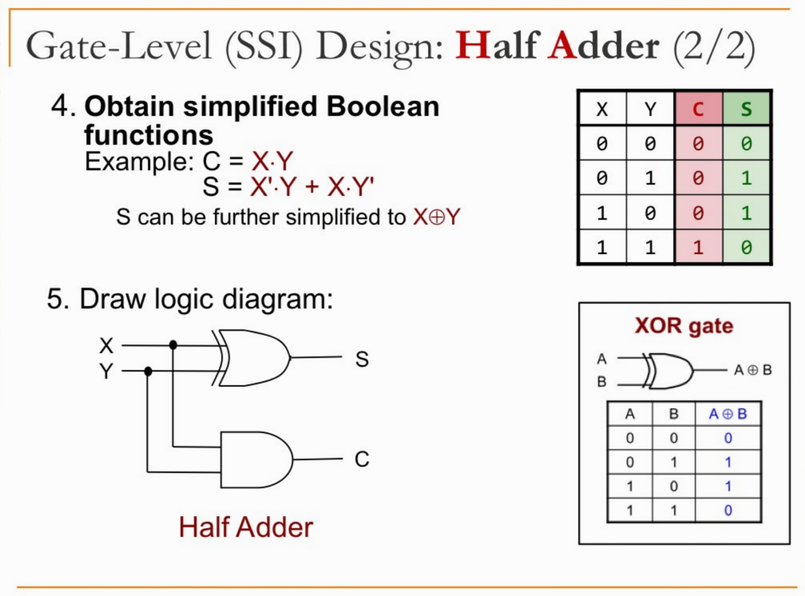
Gate-Level (SSI) Design: **Half Adder** (1/2)

- Design procedure:
  1. State problem
     Example: Build a **Half Adder**
  2. Determine and label the inputs and outputs of circuit
     Example: Two inputs and two outputs labeled, as shown below

X → Half Adder → S
Y → (X + Y) → C

3. Draw the truth table

X	Y	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0



Gate-Level (SSI) Design: **Half Adder** (2/2)

4. **Obtain simplified Boolean functions**
   Example: C = $X \cdot Y$
   S = $X' \cdot Y + X \cdot Y'$
   S can be further simplified to $X \oplus Y$

X	Y	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

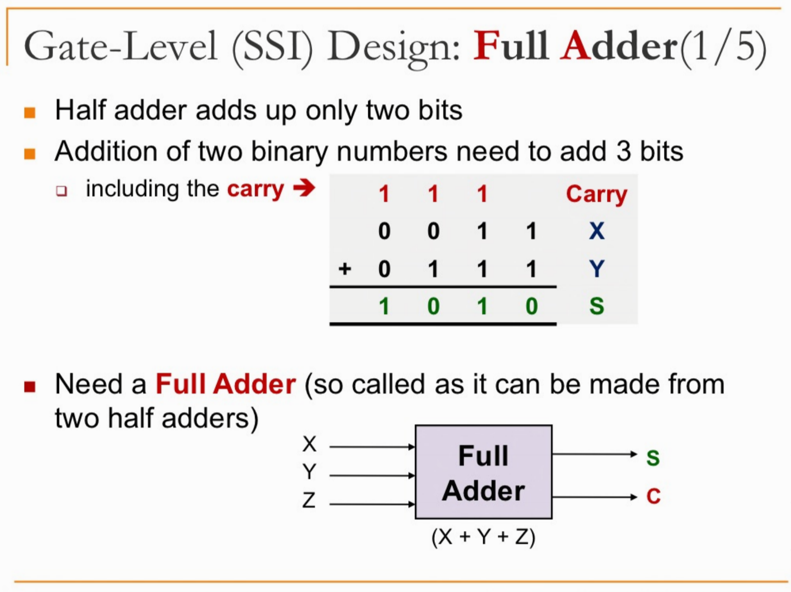
5. Draw logic diagram:

Half Adder

XOR gate

A	B	A ⊕ B
0	0	0
0	1	1
1	0	1
1	1	0

Full -adder



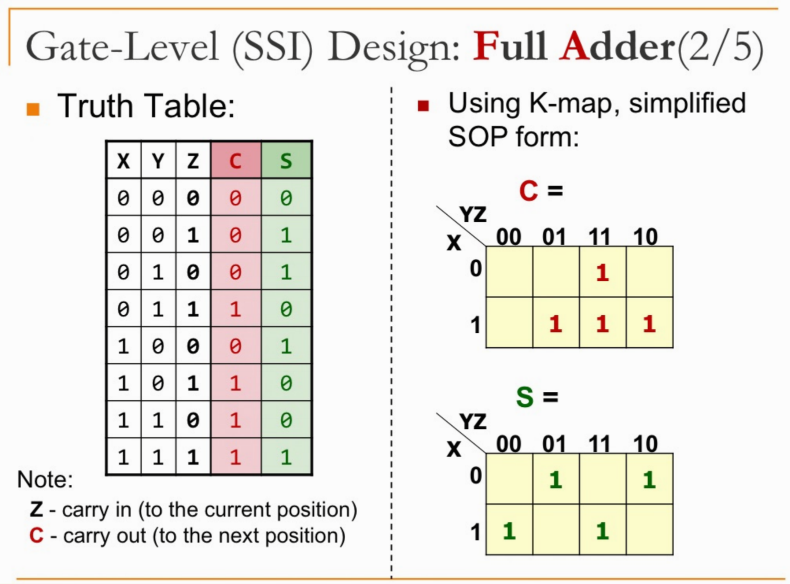
Gate-Level (SSI) Design: **Full Adder**(1/5)

- Half adder adds up only two bits
- Addition of two binary numbers need to add 3 bits
  - including the **carry** ➔

	1	1	1		Carry
	0	0	1	1	X
+	0	1	1	1	Y
	1	0	1	0	S

- Need a **Full Adder** (so called as it can be made from two half adders)

X
Y
Z
Full Adder
S
C
(X + Y + Z)



Gate-Level (SSI) Design: **Full Adder**(2/5)

- Truth Table:

X	Y	Z	C	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

Note:
Z - carry in (to the current position)
C - carry out (to the next position)

- Using K-map, simplified SOP form:

C =

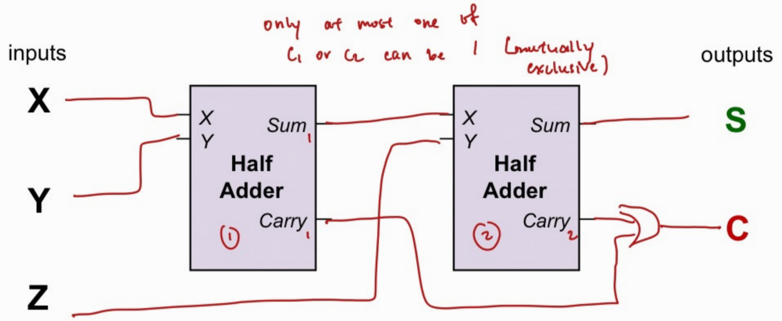
X \ YZ	00	01	11	10
0			1	
1		1	1	1

S =

X \ YZ	00	01	11	10
0		1		1
1	1		1	

Gate-Level (SSI) Design: **Full Adder**(3/5)

- Alternatively, can you construct the full-adder using **two half adders** and a suitable logic gate?


only at most one of
C_1 or C_2 can be 1 (mutually exclusive)

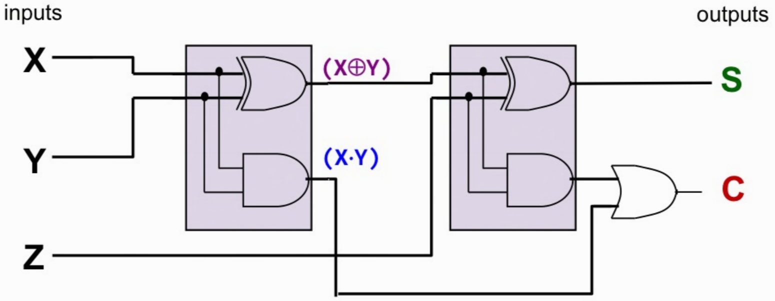
- This is an example of **block level design**

Gate-Level (SSI) Design: **Full Adder**(4/5)

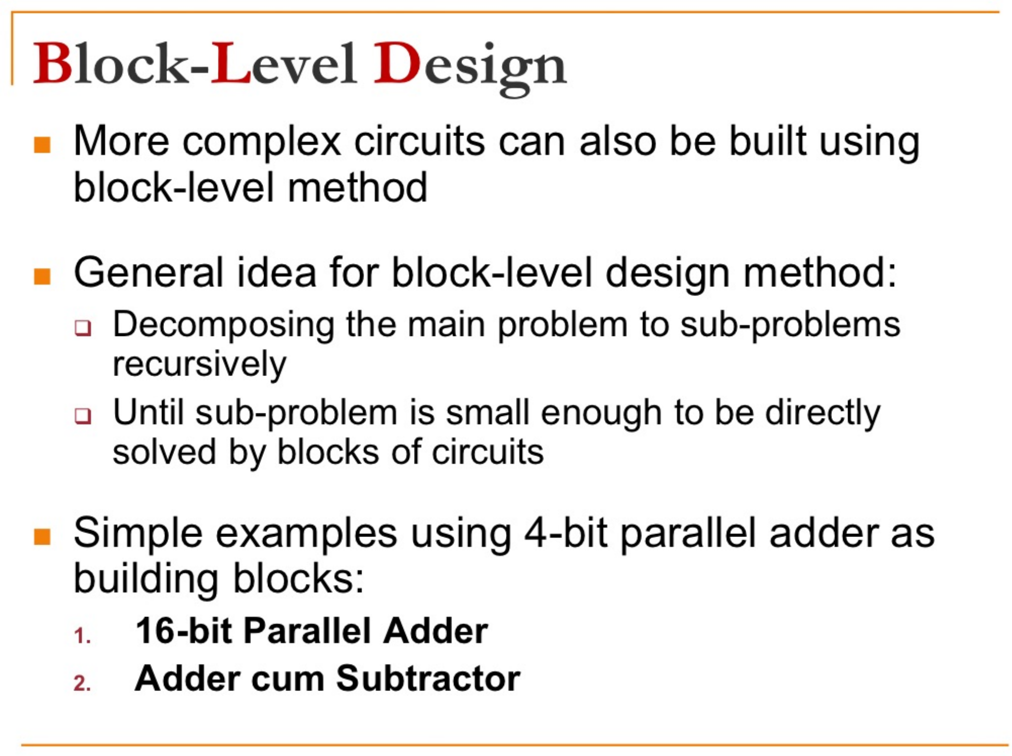
- If we trace the internal circuit, we get:

  C = X·Y + (X⊕Y)·Z

  S = X⊕(Y⊕Z) = (X⊕Y)⊕Z (XOR is associative)



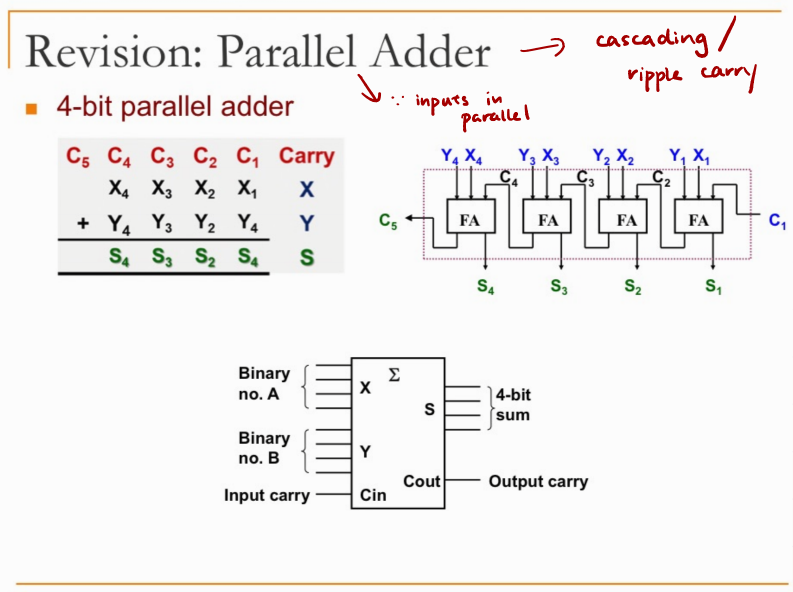
19.3 - Block-level design



Block-Level Design

- More complex circuits can also be built using block-level method
- General idea for block-level design method:
  - Decomposing the main problem to sub-problems recursively
  - Until sub-problem is small enough to be directly solved by blocks of circuits
- Simple examples using 4-bit parallel adder as building blocks:
  1. **16-bit Parallel Adder**
  2. **Adder cum Subtractor**

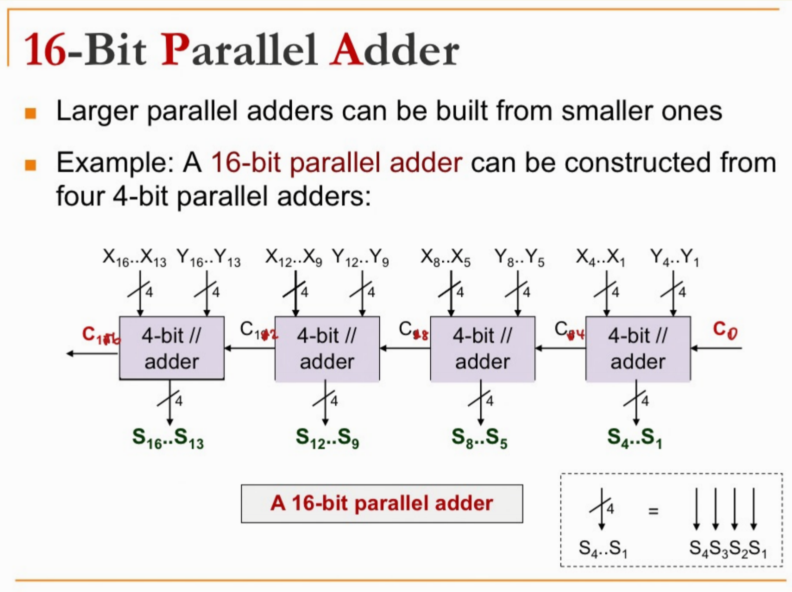
Parallel adders



Revision: Parallel Adder → cascading / ripple carry

- 4-bit parallel adder ↓ ∵ inputs in parallel

C_5	C_4	C_3	C_2	C_1	Carry
	X_4	X_3	X_2	X_1	X
+	Y_4	Y_3	Y_2	Y_4	Y
	S_4	S_3	S_2	S_4	S

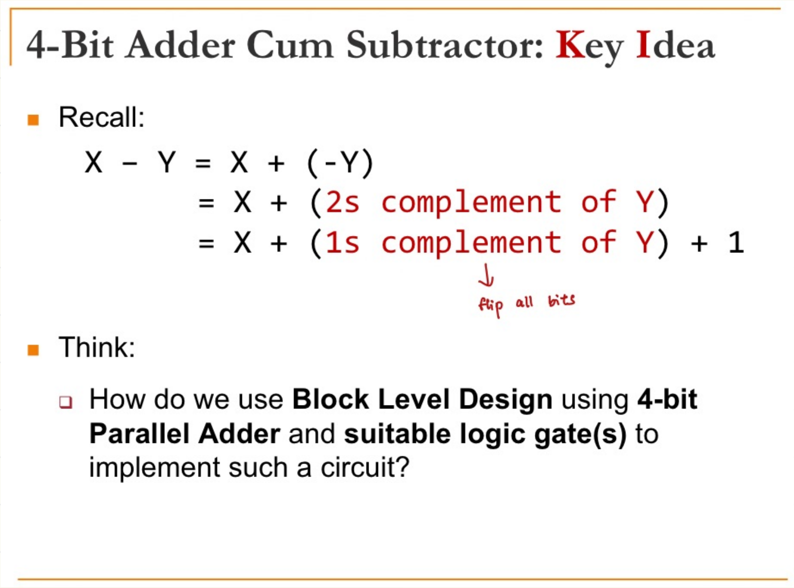


16-Bit Parallel Adder

- Larger parallel adders can be built from smaller ones
- Example: A 16-bit parallel adder can be constructed from four 4-bit parallel adders:

A 16-bit parallel adder

Adder cum subtractor



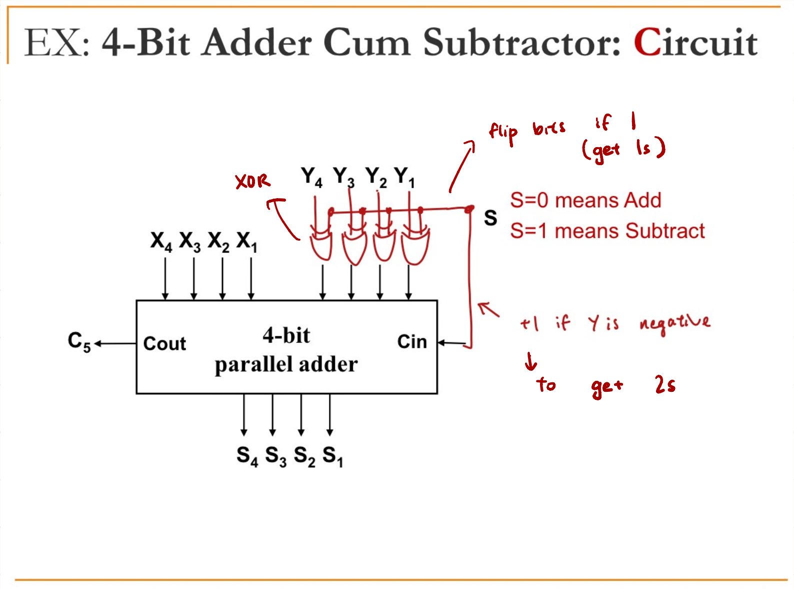
4-Bit Adder Cum Subtractor: Key Idea

- Recall:

$$
\begin{aligned}
X - Y &= X + (-Y) \\
&= X + (\text{2s complement of } Y) \\
&= X + (\text{1s complement of } Y) + 1
\end{aligned}
$$

↓ flip all bits

- Think:
  - How do we use **Block Level Design** using **4-bit Parallel Adder** and **suitable logic gate(s)** to implement such a circuit?



EX: 4-Bit Adder Cum Subtractor: Circuit

S=0 means Add
S=1 means Subtract

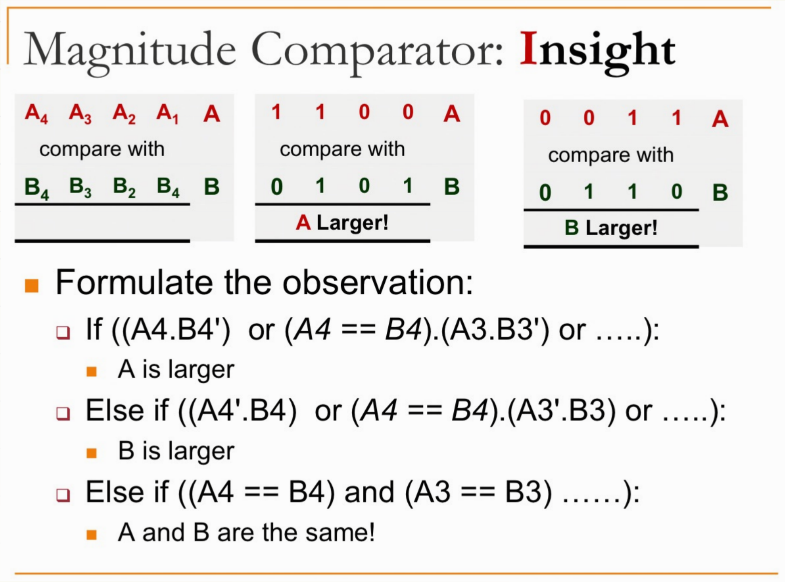
XOR

flip bits if 1 (get 1s)

+1 if Y is negative
↓
to get 2s

19.4 - Arithmetic circuits

Magnitude comparator



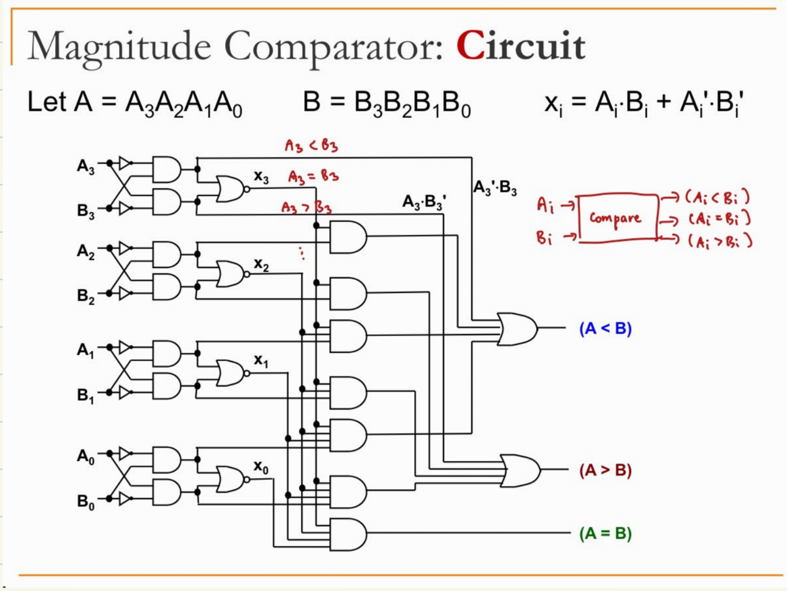
Magnitude Comparator: Insight

A_4	A_3	A_2	A_1	A
compare with				
B_4	B_3	B_2	B_4	B

1	1	0	0	A
compare with				
0	1	0	1	B
A Larger!				

0	0	1	1	A
compare with				
0	1	1	0	B
B Larger!				

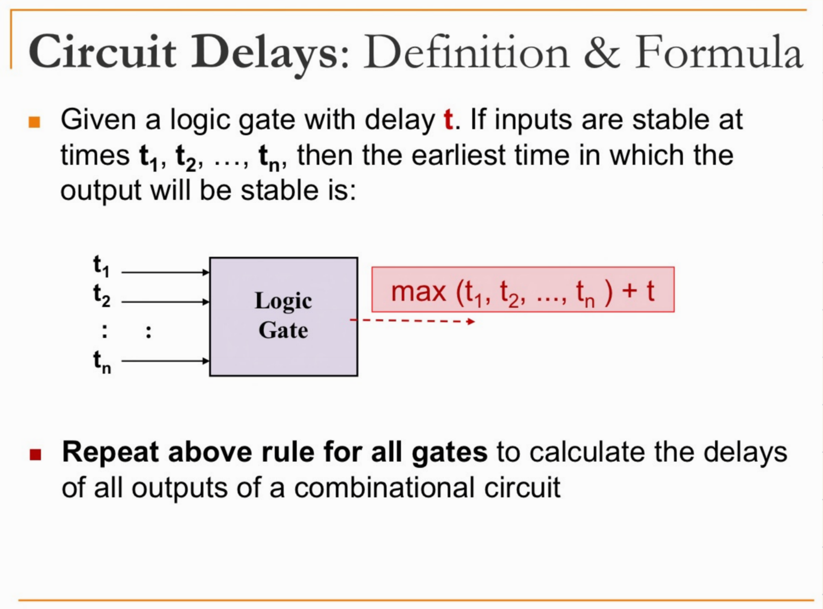
- Formulate the observation:
  - If ((A4.B4') or (*A4* == *B4*).(A3.B3') or …..):
    - A is larger
  - Else if ((A4'.B4) or (*A4* == *B4*).(A3'.B3) or …..):
    - B is larger
  - Else if ((A4 == B4) and (A3 == B3) ……):
    - A and B are the same!



Magnitude Comparator: Circuit

Let $A = A_3A_2A_1A_0$ $B = B_3B_2B_1B_0$ $x_i = A_i \cdot B_i + A_i' \cdot B_i'$

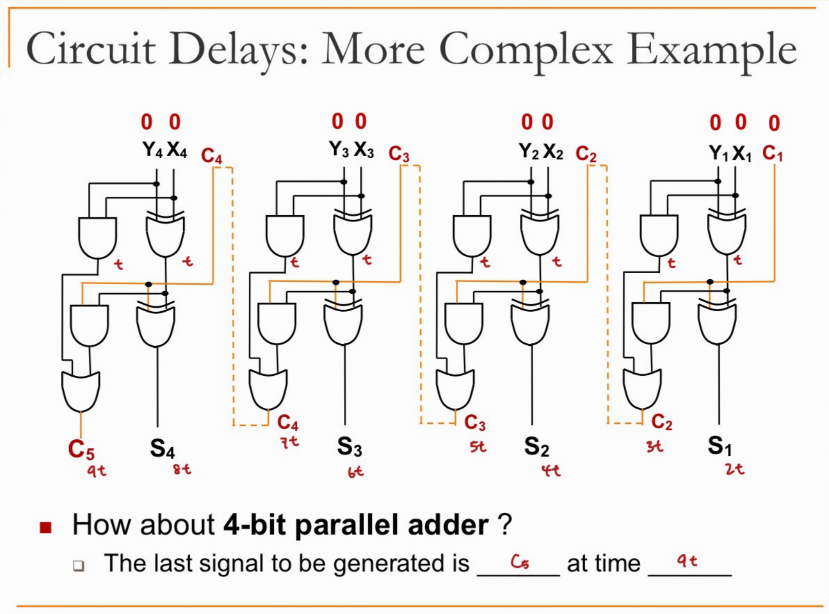
19.5 - Circuit delays



Circuit Delays: Definition & Formula

- Given a logic gate with delay **t**. If inputs are stable at times $\mathbf{t_1}, \mathbf{t_2}, \ldots, \mathbf{t_n}$, then the earliest time in which the output will be stable is:

- **Repeat above rule for all gates** to calculate the delays of all outputs of a combinational circuit



Circuit Delays: More Complex Example

- How about **4-bit parallel adder** ?
  - The last signal to be generated is C_5 at time 9t