

The background features a dark, textured surface with numerous out-of-focus, warm-toned circular lights (bokeh) scattered across the upper half. A large, solid green shape, resembling a stylized leaf or a speech bubble, is positioned on the right side, containing the title and author information.

# Digital Circuit Design

---

Li Bai

# Module 5

---

- Subtraction
- Two's complement
- Binary subtraction as addition
- <https://www.edaplayground.com/x/bDf>

# Numer system rules

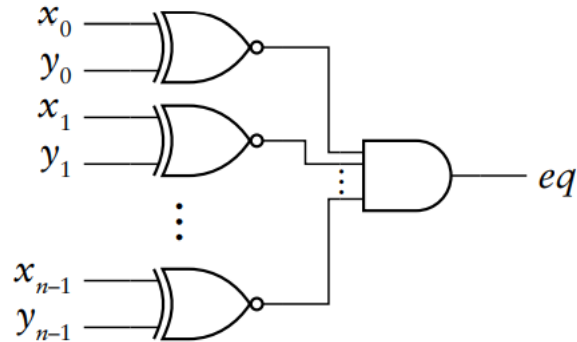
---

- In a base  $n$  system,  $n$  basic symbols or  $n$  digits are used.
- The value of a number is determined by the symbols and their positions within the description.
- For integers, the starting number (symbol) is zero (0).
- This class will initially focus on integer representations and not fractions.

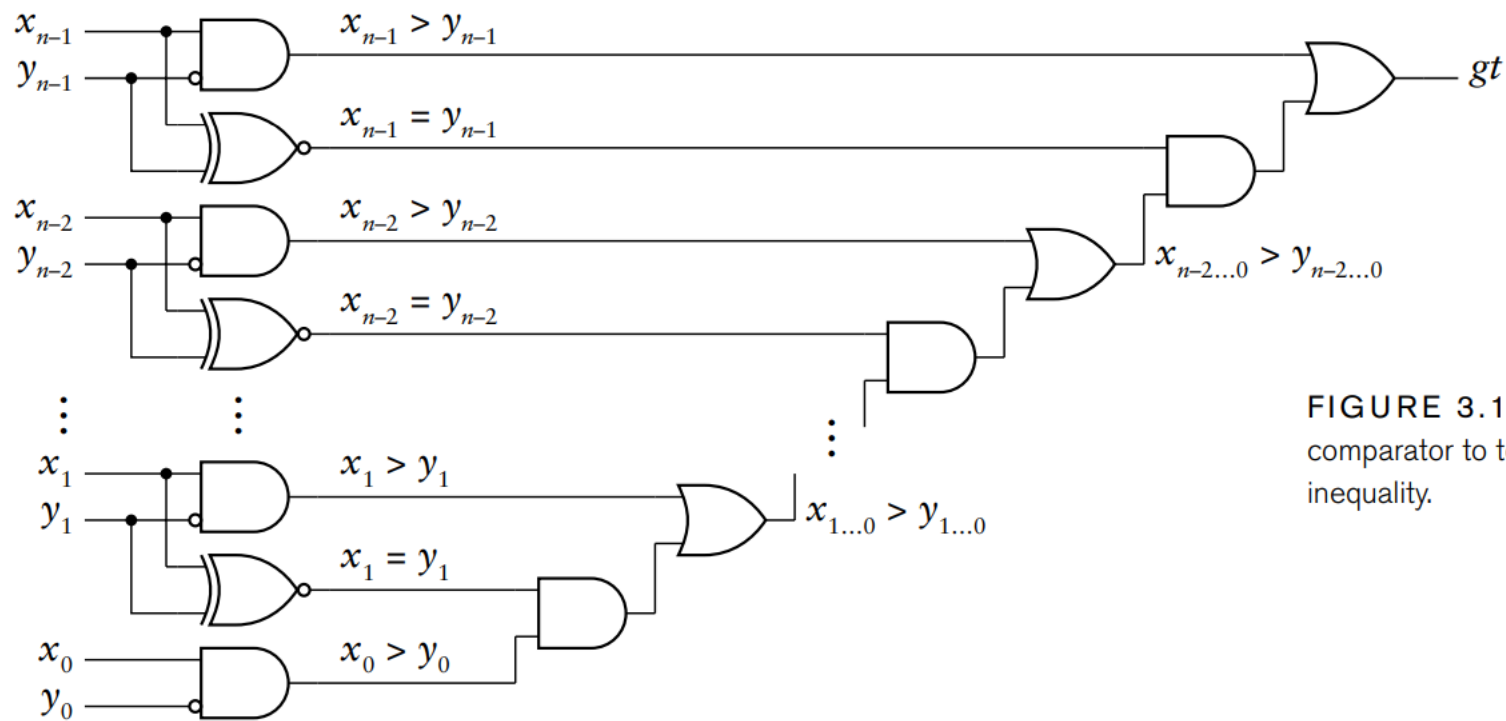
# Number comparison

---

- Equality  $x=y$



# $x > y$



**FIGURE 3.12** A magnitude comparator to test for greater than inequality.

# Unsigned Number vs Signed number

---

- A n-bit Unsigned Number range is  $(0, 2^n - 1)$
- A n-bit signed Number range is  $(-2^{(n-1)}, 2^{(n-1)} - 1)$
- 2's complement:  $-x$  can be represented as  $\sim x + 1$

# Exercises

---

72:  $\boxed{0\ 0} \begin{array}{cccccccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}$   
49:  $\begin{array}{cccccccc} 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \end{array}$   
 $\hline$   
121:  $\begin{array}{cccccccc} 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \end{array}$   
 $\hline$   
 $\begin{array}{cccccccc} 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 \end{array}$   
  

no overflow

72:  $\boxed{0\ 1} \begin{array}{cccccccc} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{array}$   
105:  $\begin{array}{cccccccc} 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \end{array}$   
 $\hline$   
 $\begin{array}{cccccccc} 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{array}$   
 $\hline$   
 $\begin{array}{cccccccc} 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \end{array}$   
  

positive overflow

-63:  $\boxed{1\ 1} \begin{array}{cccccccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}$   
-32:  $\begin{array}{cccccccc} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}$   
 $\hline$   
-95:  $\begin{array}{cccccccc} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \end{array}$   
 $\hline$   
 $\begin{array}{cccccccc} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \end{array}$   
  

no overflow

-63:  $\boxed{1\ 0} \begin{array}{cccccccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}$   
-96:  $\begin{array}{cccccccc} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}$   
 $\hline$   
 $\begin{array}{cccccccc} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{array}$   
 $\hline$   
 $\begin{array}{cccccccc} 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \end{array}$   
  

negative overflow

-42:  $\boxed{0\ 0} \begin{array}{cccccccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}$   
8:  $\begin{array}{cccccccc} 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \end{array}$   
 $\hline$   
-34:  $\begin{array}{cccccccc} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{array}$   
 $\hline$   
 $\begin{array}{cccccccc} 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \end{array}$   
  

no overflow

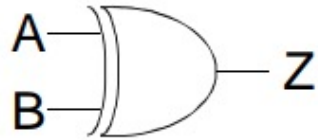
42:  $\boxed{1\ 1} \begin{array}{cccccccc} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \end{array}$   
-8:  $\begin{array}{cccccccc} 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{array}$   
 $\hline$   
34:  $\begin{array}{cccccccc} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \end{array}$   
 $\hline$   
 $\begin{array}{cccccccc} 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \end{array}$   
  

no overflow

# Signed Subtraction

$$x - y = x + (-y) = x + \bar{y} + 1$$

- Use a 2s-complement adder
  - Complement  $y$  and set  $c_0 = 1$
- How to effectively complement  $y$ ?



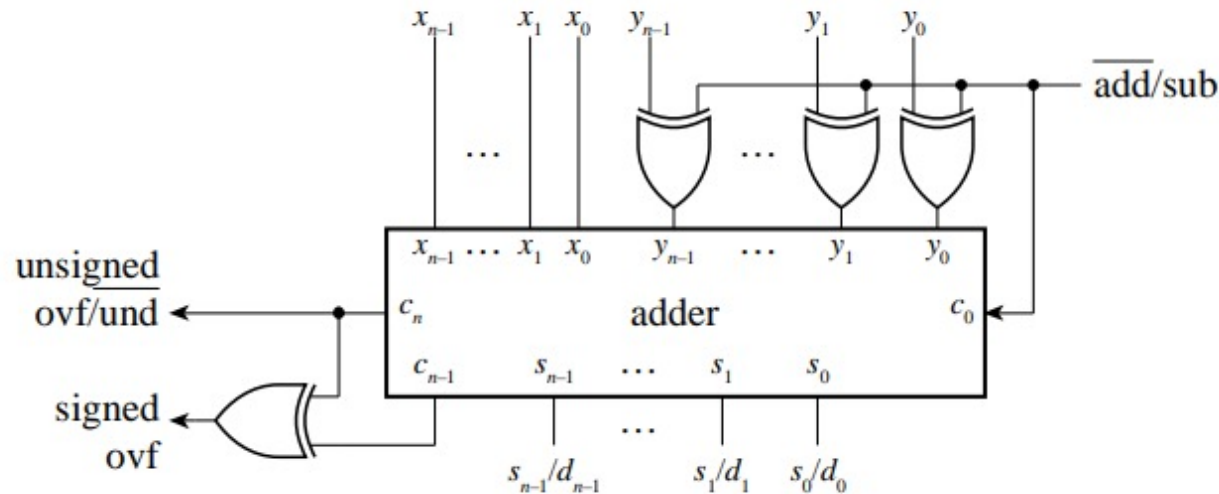
A	B	Z
0	0	0
1	0	1
0	1	1
1	1	0



# Add/Sub circuits

$$x - y = x + (-y) = x + \overline{y} + 1$$

- Use a 2s-complement adder
  - Complement  $y$  and set  $c_0 = 1$



# 2's complement sign extension

---

- To extend a signed number from  $n$  bits to  $m$  bits:
  - add  $m - n$  bits to the msb side
  - fill the extra bits with the value of the original msb
- Examples 4 bits to 8 bits:

Number	4 bit				8 bit							
2	0	0	1	0	0	0	0	0	0	1	0	
-2	1	1	1	0	1	1	1	1	1	1	0	

# Verilog using

---

- <https://edaplayground.com/x/uuR5>
- logic signed [7:0] test\_sign\_8