CMSC 313 Lecture 18

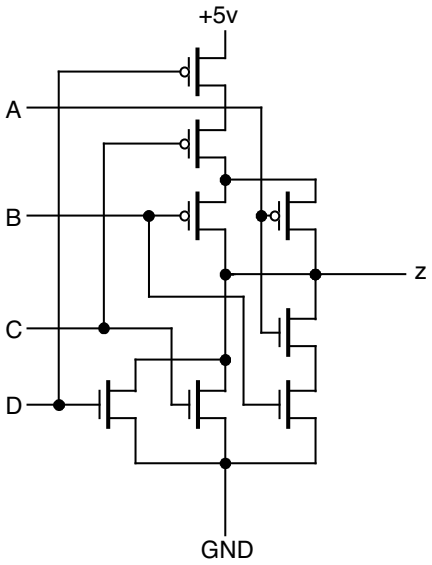
- Midterm Exam returned
- Assign Homework 3
- Circuits for Addition
- Digital Logic Components
- Programmable Logic Arrays

Due: November 11, 2004

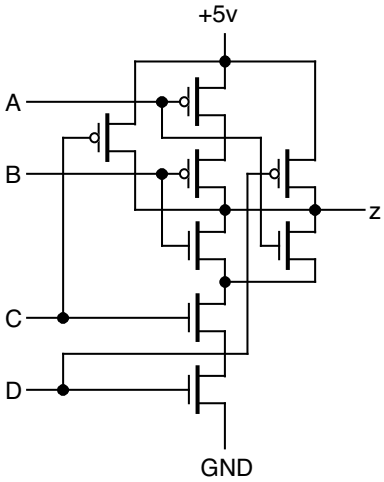
1. (30 points) Draw schematics for the following functions using AND, OR and NOT gates. (Do not simplify the formulas.)
   (a) $\overline{X}Y + XY\overline{Z} + XYZ$
   (b) $(X\overline{Y} + \overline{W}Z)(W\overline{X} + Y\overline{Z})$
   (c) $\overline{(X+Y)}\,\overline{(\overline{X}+\overline{Y})}$

2. (30 points) Using the postulates and theorems of Boolean algebra in Table A-1 (p. 451), simplify the following formulas. *Show all of your work.*
   (a) $WXYZ(WXY\overline{Z} + W\overline{X}YZ + \overline{W}XYZ + WX\overline{Y}Z)$
   (b) $AB + AB\overline{C}D + ABD\overline{E} + AB\overline{C}E + \overline{C}DE$
   (c) $MNO + \overline{QPN} + PRM + \overline{Q}OM\overline{P} + MR$

3. (40 points) For each CMOS circuit below,
   (a) Provide a truth table for the circuit's function.
   (b) For diagram (a), write down the Sum-of-Products (SOP) Boolean formula for the truth table. For diagram (b), wrtie down the Product-of-Sums (POS) Boolean formula.
   (c) Simplify the SOP or POS formula using the postulates and theorems of Boolean Algebra (p. 451). *Show all work.*
   (d) Draw the logic diagram of the simplified formula using AND, OR, NAND, NOR and NOT gates.



(a)



(b)

Last Time

- Postulates & Theorems of Boolean Algebra
- Periodic Table & Semiconductors
- P-N junction
- Field-Effect Transistors
- CMOS Logic Gates

Half Adder

- Inputs: A and B
- Outputs: S = lower bit of $A + B$, c_{out} = carry bit

A	B	S	c_{out}
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

- Using Sum-of-Products: $S = \overline{A}B + A\overline{B}$, $c_{\text{out}} = AB$.
- Alternatively, we could use XOR: $S = A \oplus B$.

Full Adder

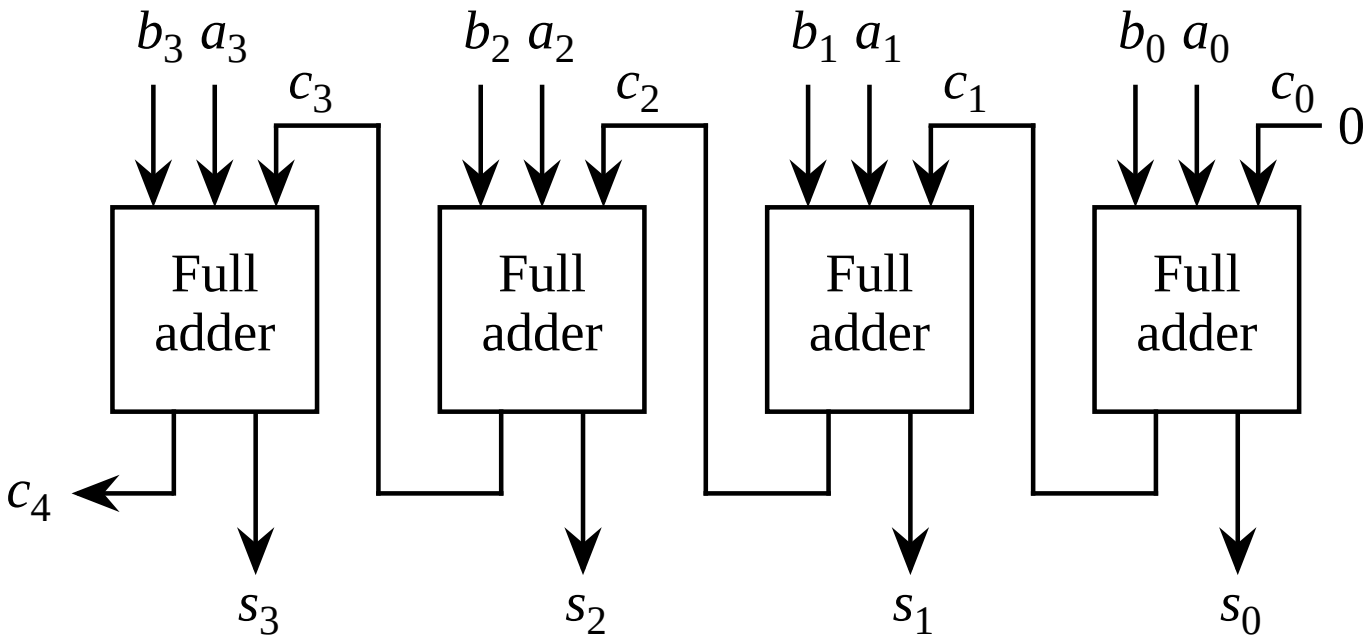
- Inputs: A, B and c_{in}
- Outputs: $S =$ lower bit of $A + B$, $c_{\text{out}} =$ carry bit

A	B	c_{in}	S	c_{out}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

- $S = \overline{A}\,\overline{B}C + \overline{A}B\overline{C} + A\overline{B}\,\overline{C} + ABC = A \oplus B \oplus C.$
- $c_{\text{out}} = \text{MAJ3} = AB + BC + AC.$

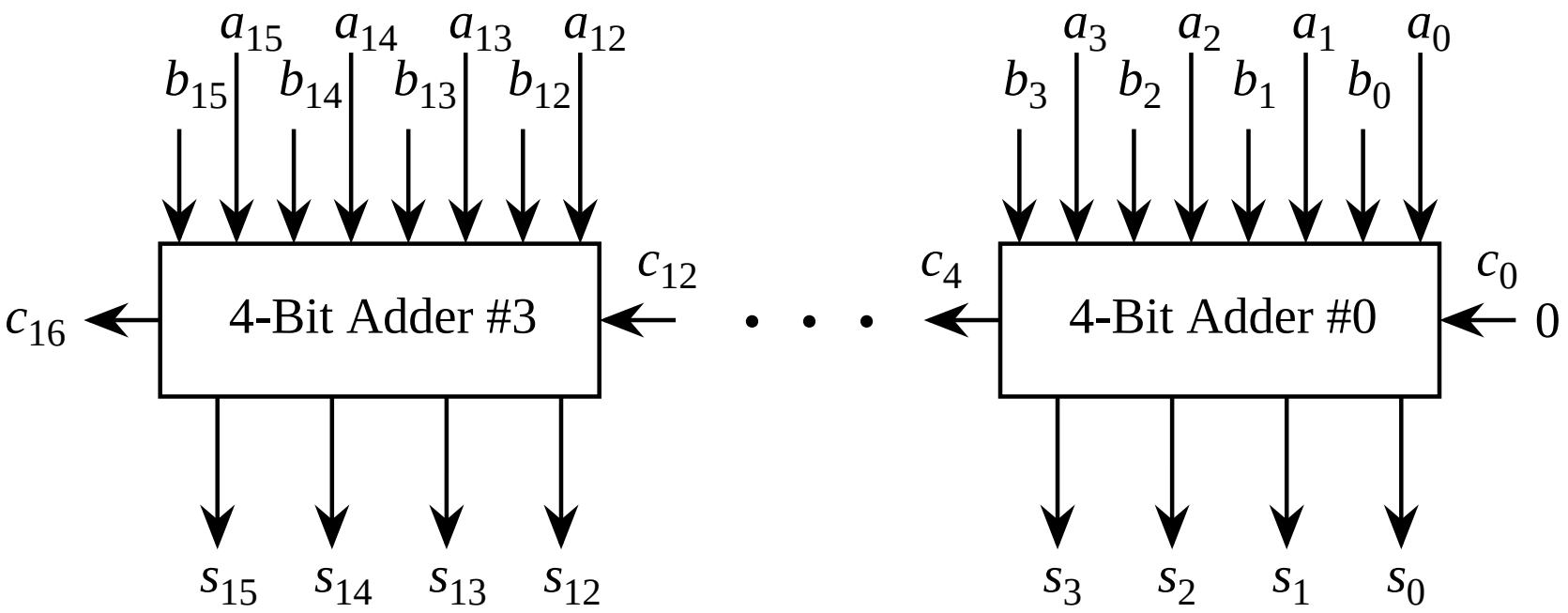
Ripple Carry Adder

- **Two binary numbers *A* and *B* are added from right to left, creating a sum and a carry at the outputs of each full adder for each bit position.**



Constructing Larger Adders

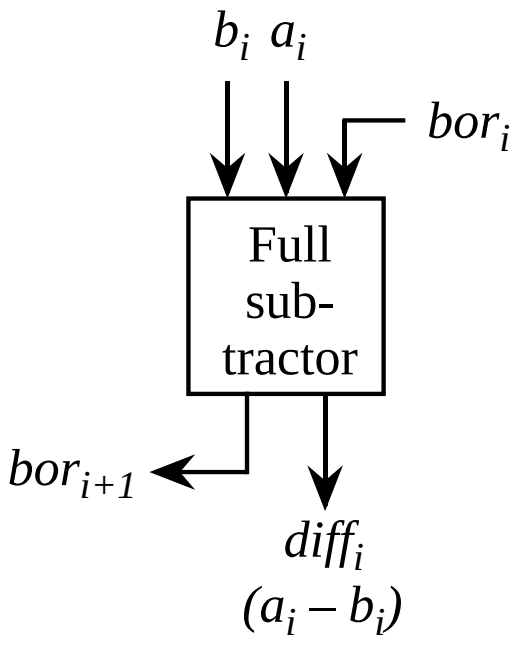
- **A 16-bit adder can be made up of a cascade of four 4-bit ripple-carry adders.**



Full Subtractor

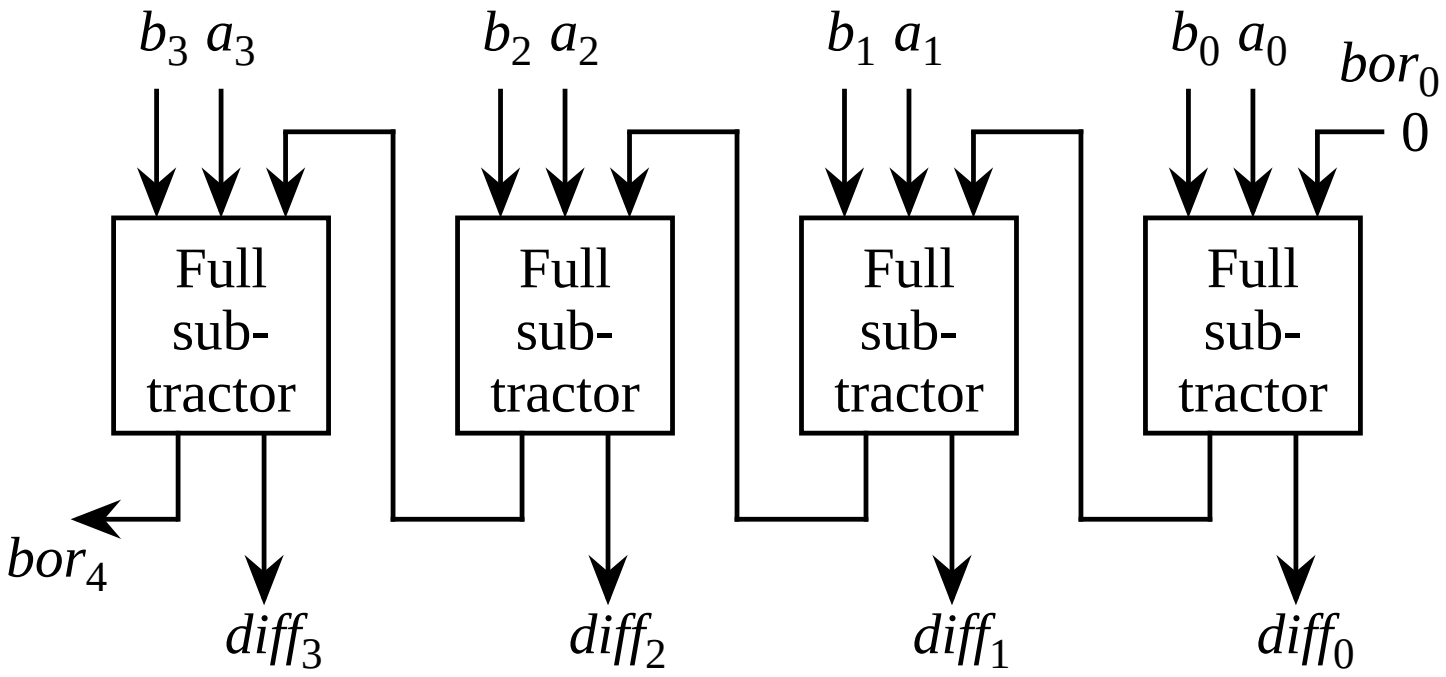
- **Truth table and schematic symbol for a ripple-borrow subtractor:**

a_i	b_i	bor_i	$diff_i$	bor_{i+1}
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1



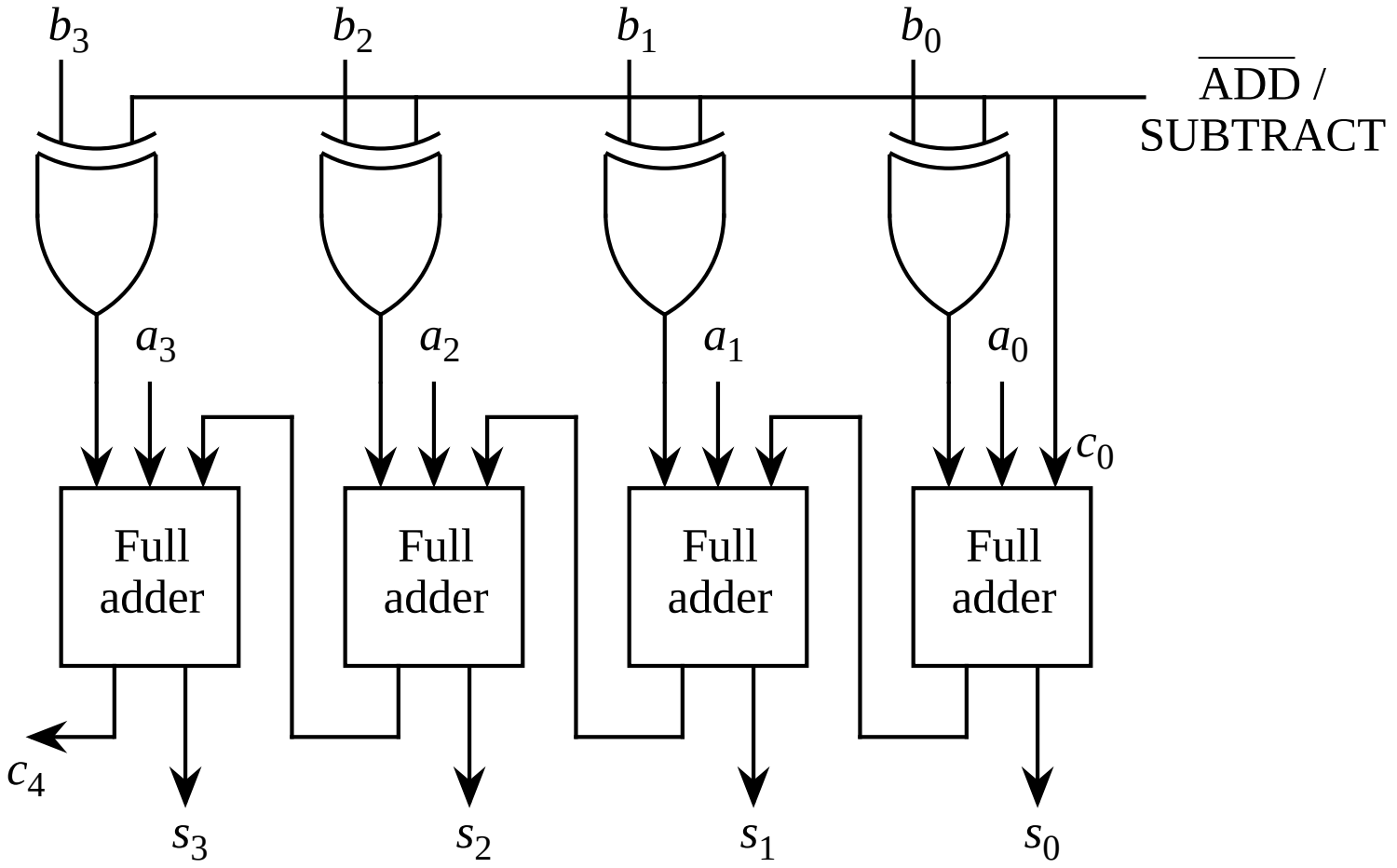
Ripple-Borrow Subtractor

- **A ripple-borrow subtractor can be composed of a cascade of full subtractors.**
- **Two binary numbers *A* and *B* are subtracted from right to left, creating a difference and a borrow at the outputs of each full subtractor for each bit position.**



Combined Adder/Subtractor

- **A single ripple-carry adder can perform both addition and subtraction, by forming the two's complement negative for *B* when subtracting. (Note that +1 is added at c_0 for two's complement.)**



Carry-Lookahead Addition

$$s_i = \bar{a}_i\bar{b}_i c_i + \bar{a}_i b_i \bar{c}_i + a_i\bar{b}_i\bar{c}_i + a_i b_i c_i$$

$$c_{i+1} = b_i c_i + a_i c_i + a_i b_i$$

$$c_{i+1} = a_i b_i + (a_i + b_i)c_i$$

$$c_{i+1} = G_i + P_i c_i$$

- **Carries are represented in terms of G_i (generate) and P_i (propagate) expressions.**

$$G_i = a_i b_i \quad \text{and} \quad P_i = a_i + b_i$$

$$c_0 = 0$$

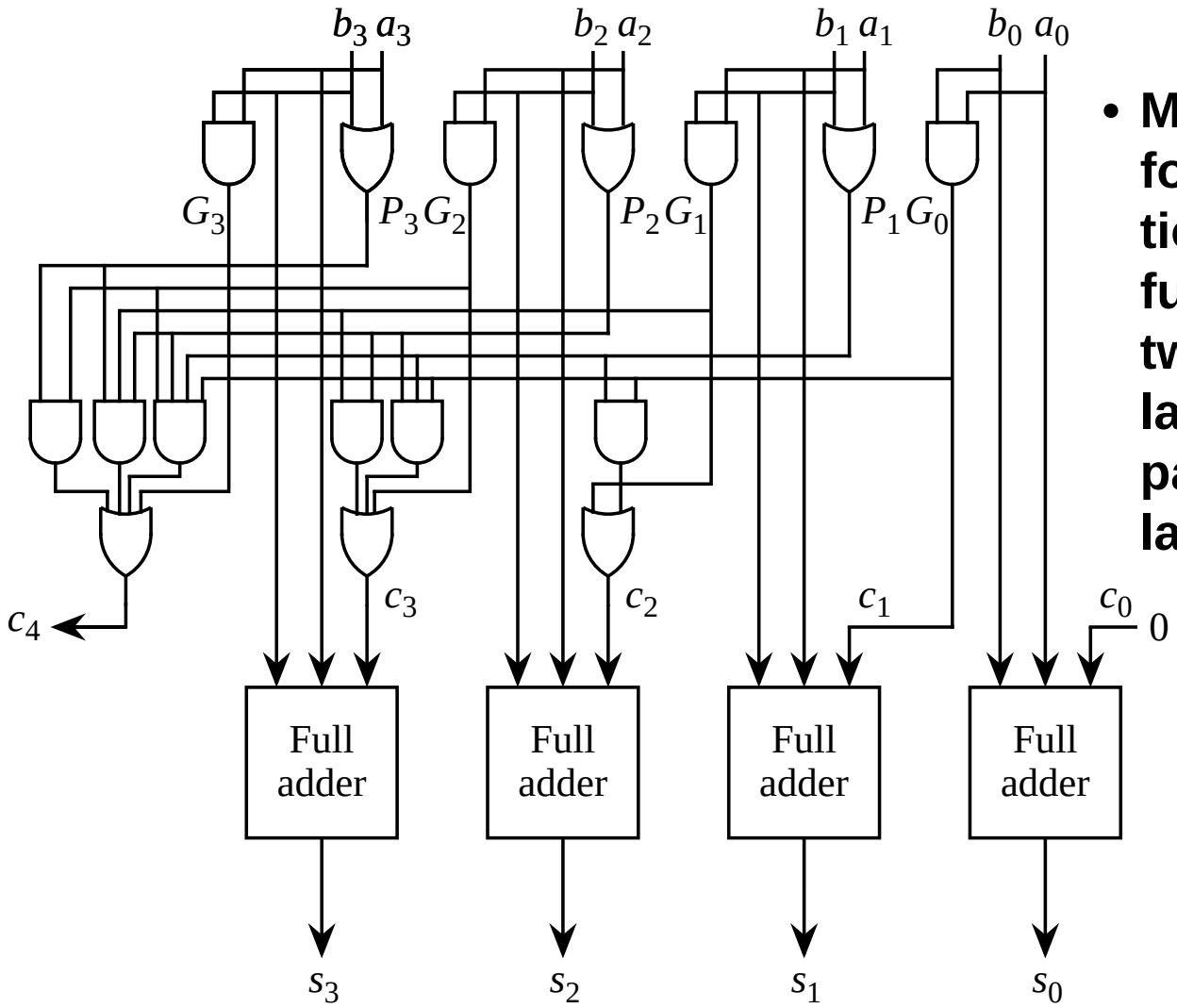
$$c_1 = G_0$$

$$c_2 = G_1 + P_1 G_0$$

$$c_3 = G_2 + P_2 G_1 + P_2 P_1 G_0$$

$$c_4 = G_3 + P_3 G_2 + P_3 P_2 G_1 + P_3 P_2 P_1 G_0$$

Carry Lookahead Adder

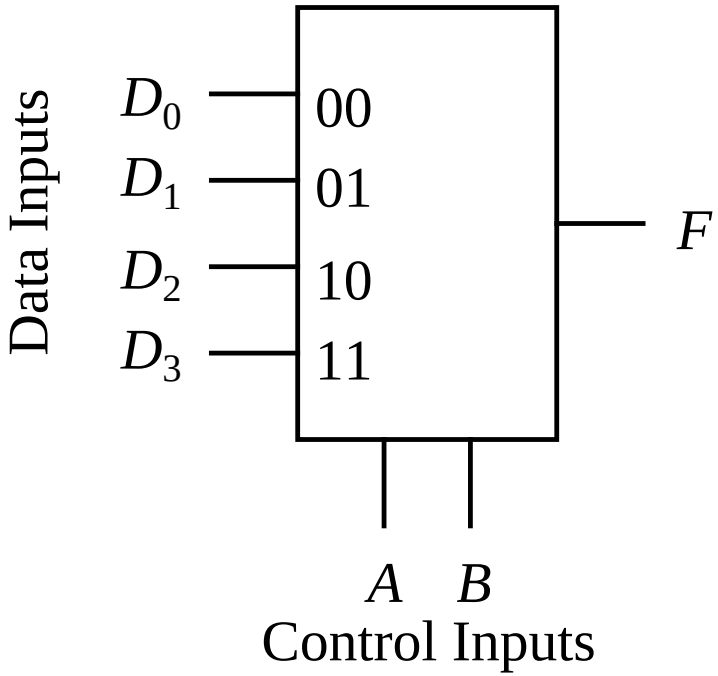


- **Maximum gate delay for the carry generation is only 3. The full adders introduce two more gate delays. Worst case path is 5 gate delays.**

Digital Components

- High level digital circuit designs are normally created using collections of logic gates referred to as *components*, rather than using individual logic gates.
- Levels of integration (numbers of gates) in an integrated circuit (IC) can roughly be considered as:
  - Small scale integration (SSI): 10-100 gates.
  - Medium scale integration (MSI): 100 to 1000 gates.
  - Large scale integration (LSI): 1000-10,000 logic gates.
  - Very large scale integration (VLSI): 10,000-upward logic gates.
  - These levels are approximate, but the distinctions are useful in comparing the relative complexity of circuits.

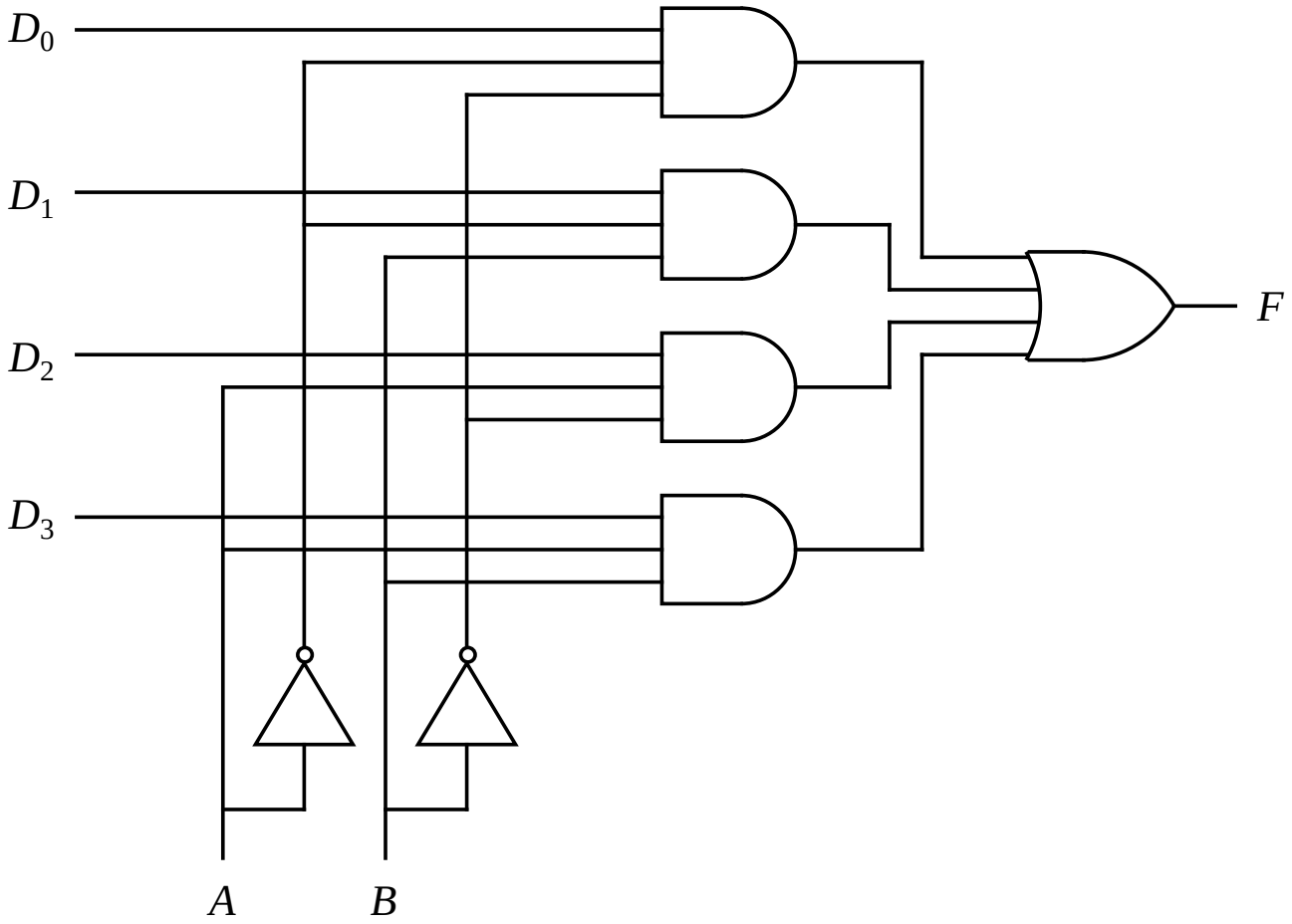
Multiplexer



A	B	F
0	0	D_0
0	1	D_1
1	0	D_2
1	1	D_3

$$F = \overline{A}\,\overline{B}\,D_0 + \overline{A}\,B\,D_1 + A\,\overline{B}\,D_2 + A\,B\,D_3$$

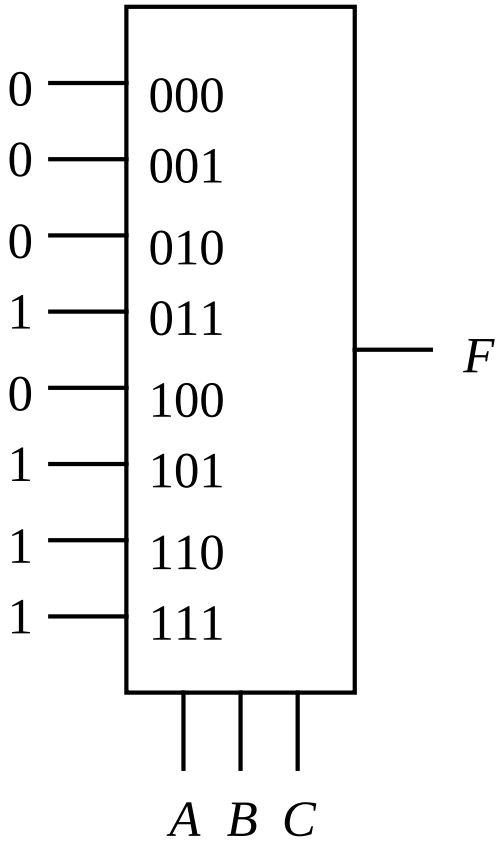
AND-OR Implementation of MUX



MUX Implementation of Majority

- **Principle: Use the 3 MUX control inputs to select (one at a time) the 8 data inputs.**

A	*B*	*C*	*M*
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1



4-to-1 MUX Implements 3-Var Function

- **Principle: Use the A and B inputs to select a pair of minterms. The value applied to the MUX data input is selected from {0, 1, C, $\overline{C}$} to achieve the desired behavior of the minterm pair.**

A	B	C	F	
0	0	0	0	0
0	0	1	0	
0	1	0	1	1
0	1	1	1	
1	0	0	0	C
1	0	1	1	
1	1	0	1	$\overline{C}$
1	1	1	0