Computer Architecture

A Generic CPU-2
Control Unit and ALU

Dr. Masood Mir

Week 4

Control Unit

- The control unit for the datapath directs the information flow through the buses, the ALU, the shifter, and the registers by applying signals to the select inputs.
- For example, to perform the microoperation

  R1 <- R2 + R3

- the control unit must provide binary selection values to the following sets of control inputs:

  1. A select, to place the contents of R2 onto A data and, hence, Bus A.
  2. B select, to place the contents of R3 onto the 0 input of MUX B; and MB select, to put the 0 input of MUX B onto Bus B.
  3. G select, to provide the arithmetic operation A + B.
  4. MF select, to place the ALU output on the MUX F output.
  5. MD select, to place the MUX F output onto Bus D.
  6. Destination select, to select R1 as the destination of the data on Bus D.
  7. Load enable, to enable a register—in this case, R1—to be loaded.

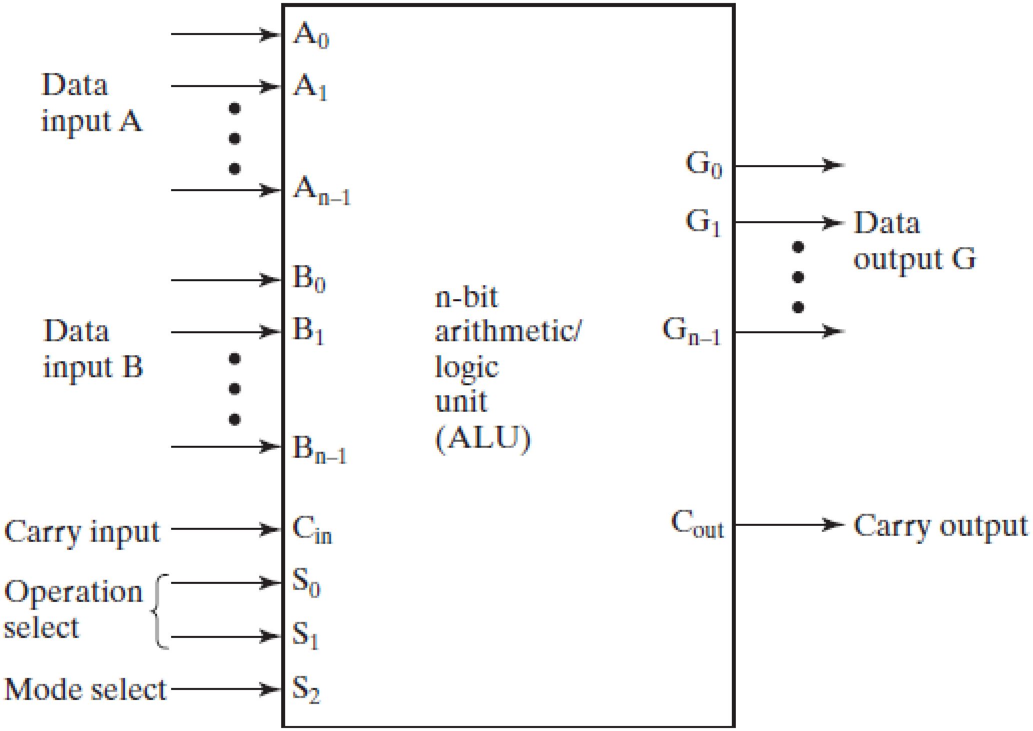
Time Delay

- The sets of input values must be generated and must become available on the corresponding control lines early in the clock cycle.
- The binary data from the two source registers must propagate through the multiplexers and the ALU and on into the inputs of the destination register, all during the remainder of the same clock cycle.
- Then, when the next positive clock edge arrives, the binary data on Bus D is loaded into the destination register.
- To achieve fast operation, the ALU and shifter are constructed with combinational logic having a limited number of levels.

Arithmetic Logic Unit

- The ALU is a combinational circuit that performs a set of basic arithmetic and logic microoperations.
- It has a number of selection lines used to determine the operation to be performed.
- The selection lines are decoded within the ALU, so that k selection lines can specify up to 2k distinct operations.
- Figure shows the external connections for a typical n-bit ALU.
- The n data inputs from A and the n data inputs from B provide the two operands to generate the result of an operation at the G outputs.
- The mode-select input S2 distinguishes between arithmetic and logic operations.
- The two Operation select inputs S1 and S0 and the Carry input Cin specify the 8 (eight) arithmetic operations with S2 at 0.
- Operand select input S0 and Cin specify the four logic operations with S2 at 1.
- We will discuss the design of this ALU in three stages. First, we design the arithmetic section. Then we design the logic section, and finally, we combine the two sections to form the ALU.

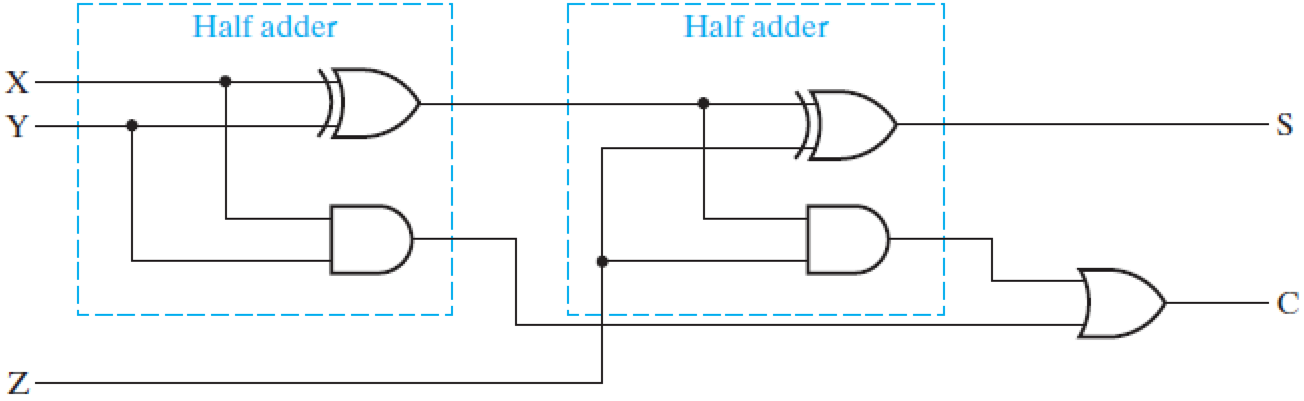
Arithmetic Logic Unit

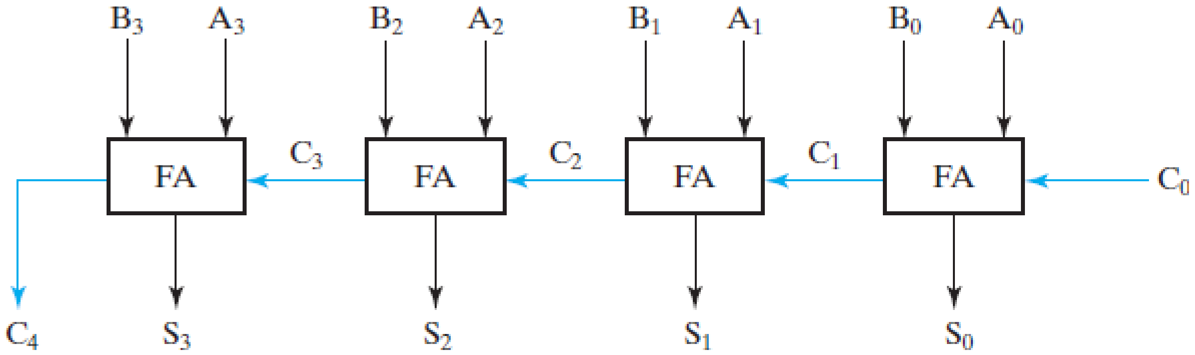


□ **FIGURE 8-2**
Symbol for an *n*-Bit ALU

Arithmetic Circuit

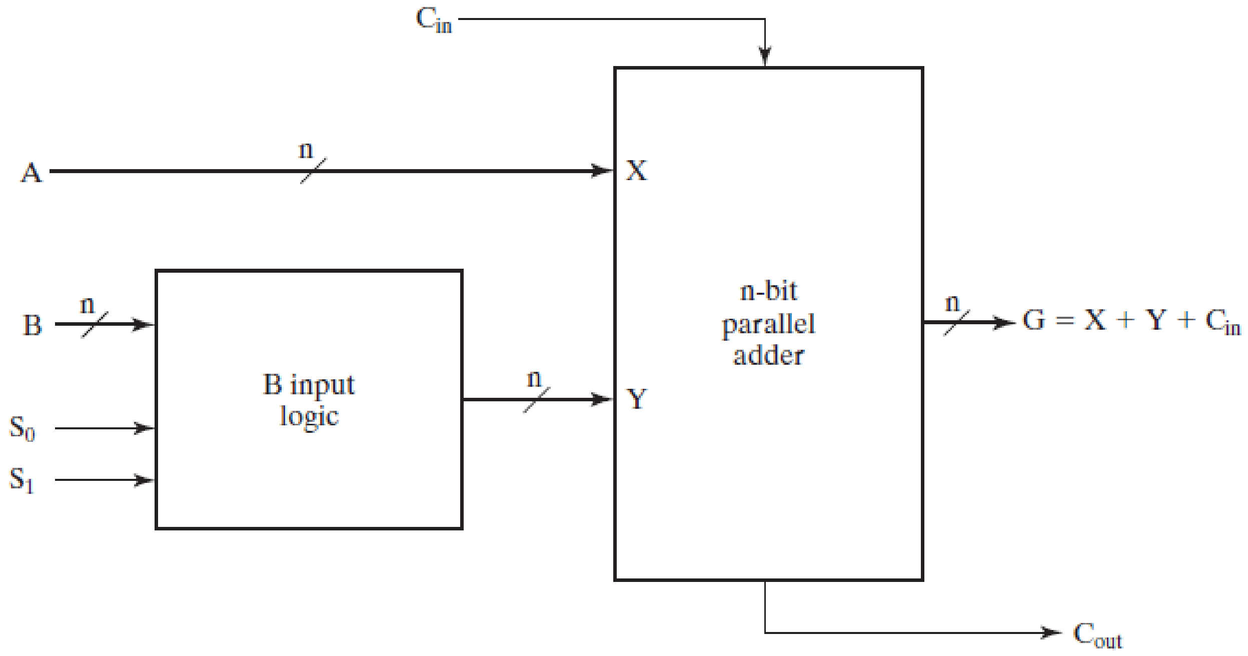
- The basic component of an arithmetic circuit is a parallel adder, which is constructed with a number of full-adder circuits connected in cascade, as discussed in DLD.





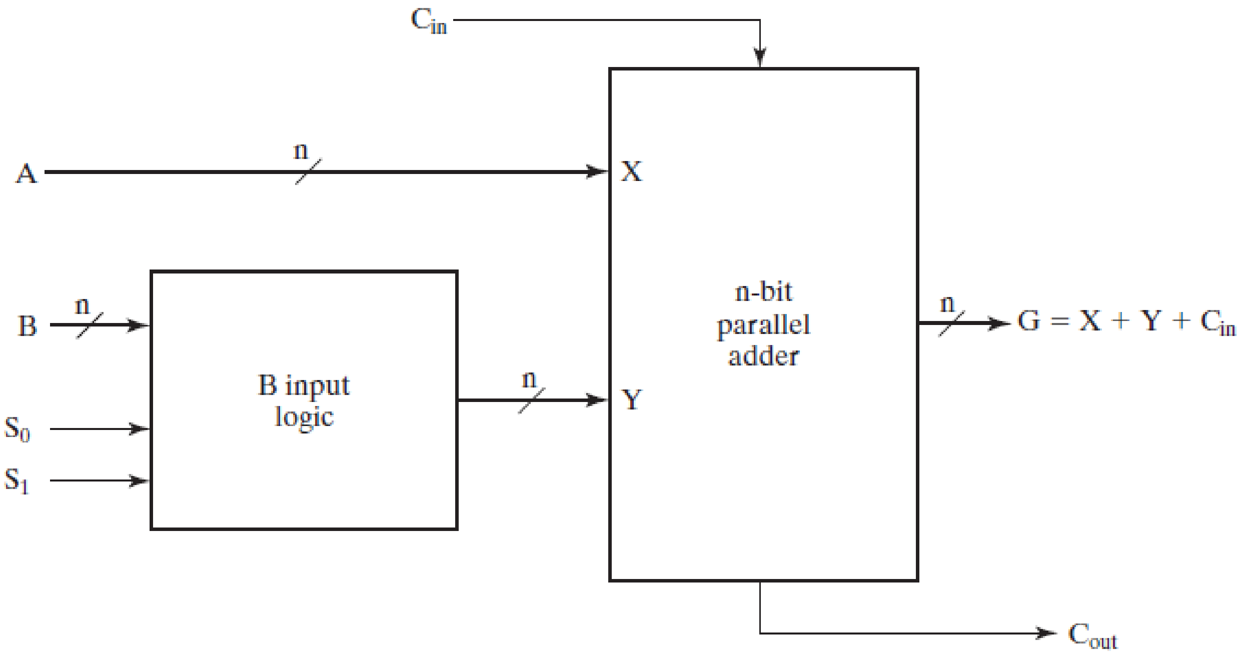
Arithmetic Circuit

- By controlling the data inputs to the parallel adder, it is possible to obtain different types of arithmetic operations.
- The block diagram in Figure demonstrates a configuration in which one set of inputs to the parallel adder is controlled by the select lines S1 and S0.
- There are n bits in the arithmetic circuit, with two inputs A and B and output G. The n inputs from B go through the B input logic to the Y inputs of the parallel adder.
- The output of the parallel adder is calculated from the arithmetic sum as G = X + Y + Cin



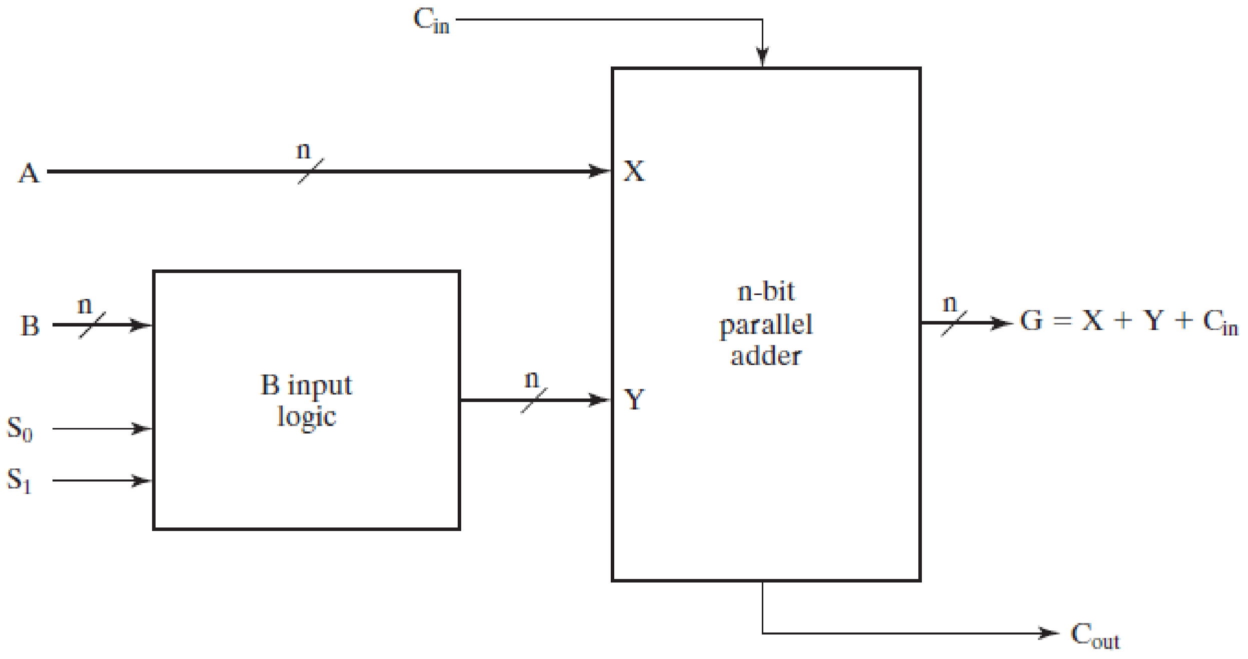
Arithmetic Circuit

- By controlling the data inputs to the parallel adder, it is possible to obtain different types of arithmetic operations.
- The block diagram in Figure demonstrates a configuration in which one set of inputs to the parallel adder is controlled by the select lines S1 and S0.
- There are n bits in the arithmetic circuit, with two inputs A and B and output G. The n inputs from B go through the B input logic to the Y inputs of the parallel adder.



Arithmetic Circuit

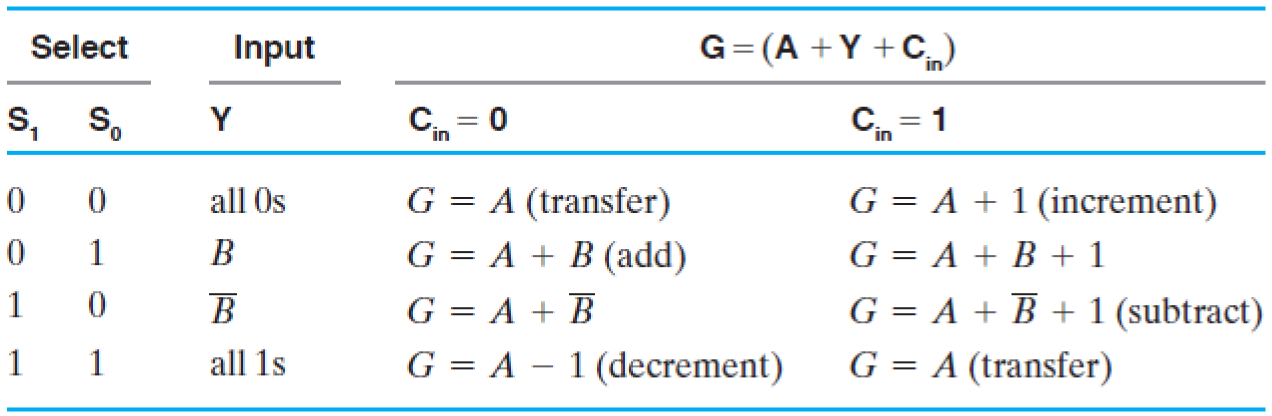
- The output of the parallel adder is calculated from the arithmetic sum as G = X + Y + Cin
- where X is the n-bit binary number from the inputs and Y is the n-bit binary number from the B input logic. Cin is the input carry, which equals 0 or 1.
- Note that the symbol + in the equation denotes arithmetic addition and not OR logic.



Arithmetic Circuit

- Table shows the arithmetic operations that are obtainable by controlling the value of Y with the two selection inputs S1 and S0.
- If the inputs from B are ignored and we insert all 0s at the Y inputs, the output sum becomes G = A + 0 + Cin. This gives G = A when Cin = 0 and G = A + 1 when Cin = 1. In the first case, we have
- a direct transfer from input A to output G. In the second case, the value of A is incremented
- by 1.
- For a straight arithmetic addition, it is necessary to apply the B inputs to the Y inputs of the parallel adder. This gives G = A + B when Cin = 0.

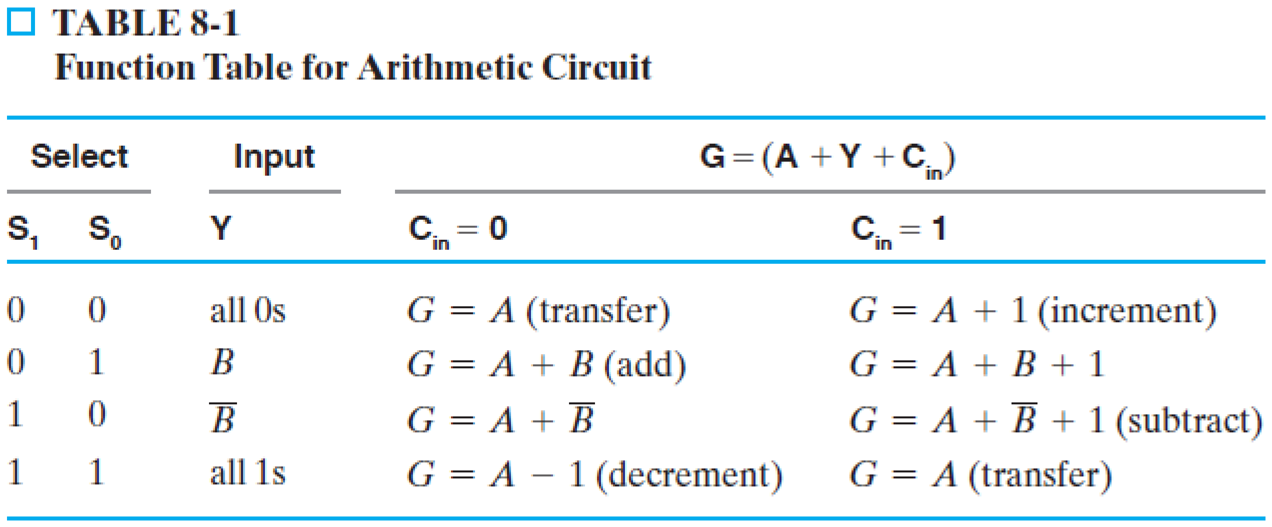
□ **TABLE 8-1**
Function Table for Arithmetic Circuit



Select		Input	$G = (A + Y + C_{in})$	
S_1	S_0	Y	$C_{in} = 0$	$C_{in} = 1$
0	0	all 0s	$G = A$ (transfer)	$G = A + 1$ (increment)
0	1	B	$G = A + B$ (add)	$G = A + B + 1$
1	0	$\overline{B}$	$G = A + \overline{B}$	$G = A + \overline{B} + 1$ (subtract)
1	1	all 1s	$G = A - 1$ (decrement)	$G = A$ (transfer)

Arithmetic Circuit

- Arithmetic subtraction is achieved by applying the complement of inputs B to the Y inputs of the parallel adder, to obtain G = A + B + 1 when Cin = 1. This gives A plus the 2s complement of B, which is equivalent to 2s complement subtraction.
- All 1s is the 2s complement representation for -1. Thus, applying all 1s to the Y inputs with Cin = 0 produces the decrement operation G = A - 1.

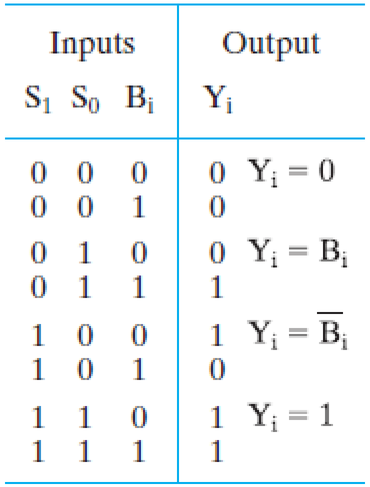


□ **TABLE 8-1**
Function Table for Arithmetic Circuit

Select		Input	$G = (A + Y + C_{in})$	
S_1	S_0	Y	$C_{in} = 0$	$C_{in} = 1$
0	0	all 0s	$G = A$ (transfer)	$G = A + 1$ (increment)
0	1	B	$G = A + B$ (add)	$G = A + B + 1$
1	0	$\overline{B}$	$G = A + \overline{B}$	$G = A + \overline{B} + 1$ (subtract)
1	1	all 1s	$G = A - 1$ (decrement)	$G = A$ (transfer)

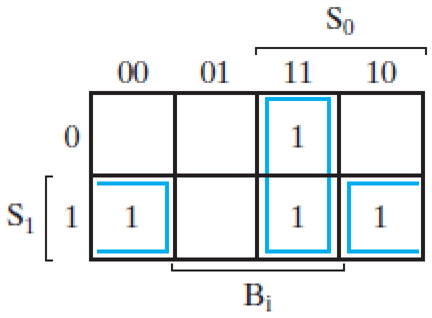
Arithmetic Circuit

- The B input logic in Figure can be implemented with n multiplexers. The data inputs to each multiplexer in stage i for i = 0, 1, …… n - 1 are 0, B_i, B_i, and 1, corresponding to selection values S1S0: 00, 01, 10, and 11, respectively.
- Thus, the arithmetic circuit can be constructed with n full adders and n 4-to-1 multiplexers.
- The number of gates in the B input logic can be reduced if, instead of using 4-to-1 multiplexers, we go through the logic design of one stage (one bit) of the B input logic.
- This can be done by K-map as shown in Figure



Inputs			Output	
S_1	S_0	B_i	Y_i	
0	0	0	0	$Y_i = 0$
0	0	1	0	
0	1	0	0	$Y_i = B_i$
0	1	1	1	
1	0	0	1	$Y_i = \overline{B_i}$
1	0	1	0	
1	1	0	1	$Y_i = 1$
1	1	1	1	

(a) Truth table



	00	01	11	10
0			1	
1	1		1	1

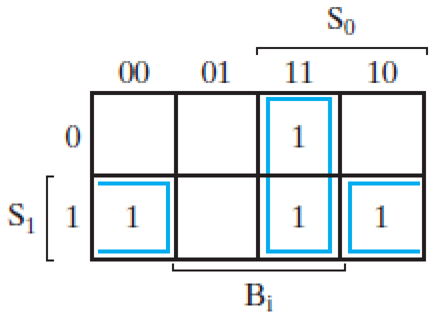
(b) Map simplification:
$Y_i = B_iS_0 + \overline{B_i}S_1$

Arithmetic Circuit

- The truth table for one typical stage i of the logic is given in Figure (a).
- The inputs are S1, S0, and Bi, and the output is Yi.
- Following the requirements specified in Table 8-1, we let Yi = 0 when S1S0 = 00, and similarly assign the other three values of Yi for each of the combinations of the selection variables.
- Output Yi is simplified in the map in Figure 8-4(b) to give Yi = BiS0 + Bi’S1.
- where S1 and S0 are common to all n stages. Each stage i is associated with input Bi and output Yi for i = 0, 1, 2, …., n -1.

Inputs			Output	
S_1	S_0	B_i	Y_i	
0	0	0	0	$Y_i = 0$
0	0	1	0	
0	1	0	0	$Y_i = B_i$
0	1	1	1	
1	0	0	1	$Y_i = \overline{B}_i$
1	0	1	0	
1	1	0	1	$Y_i = 1$
1	1	1	1	

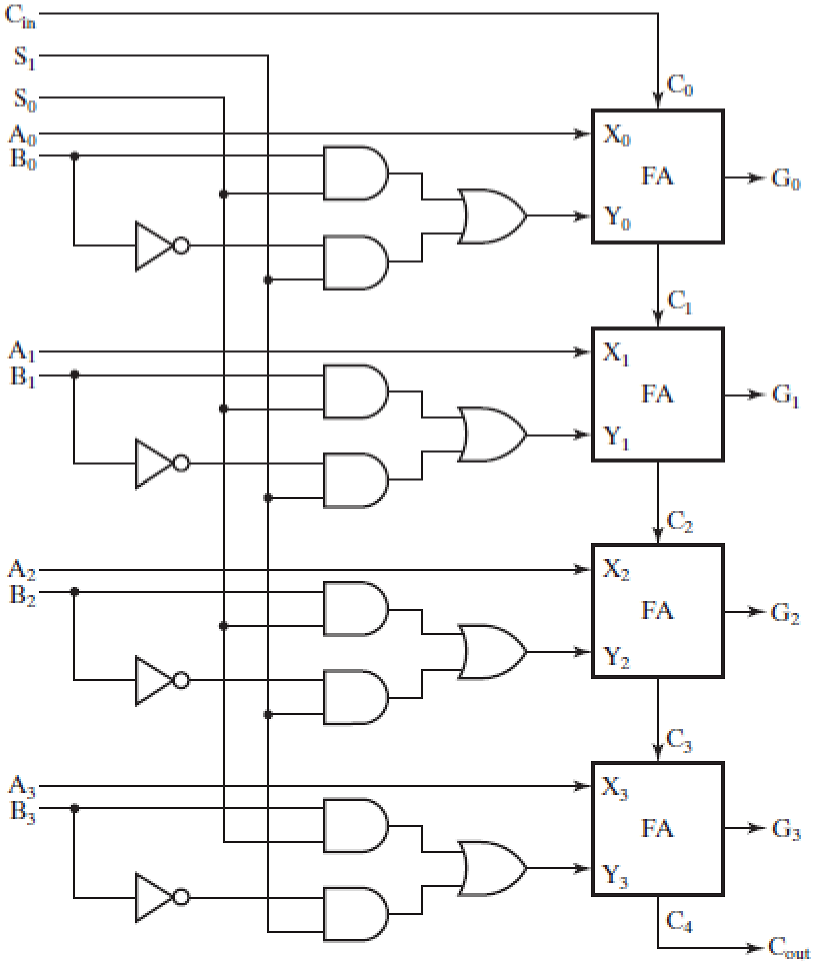
(a) Truth table



(b) Map simplification:
$Y_i = B_iS_0 + \overline{B}_iS_1$

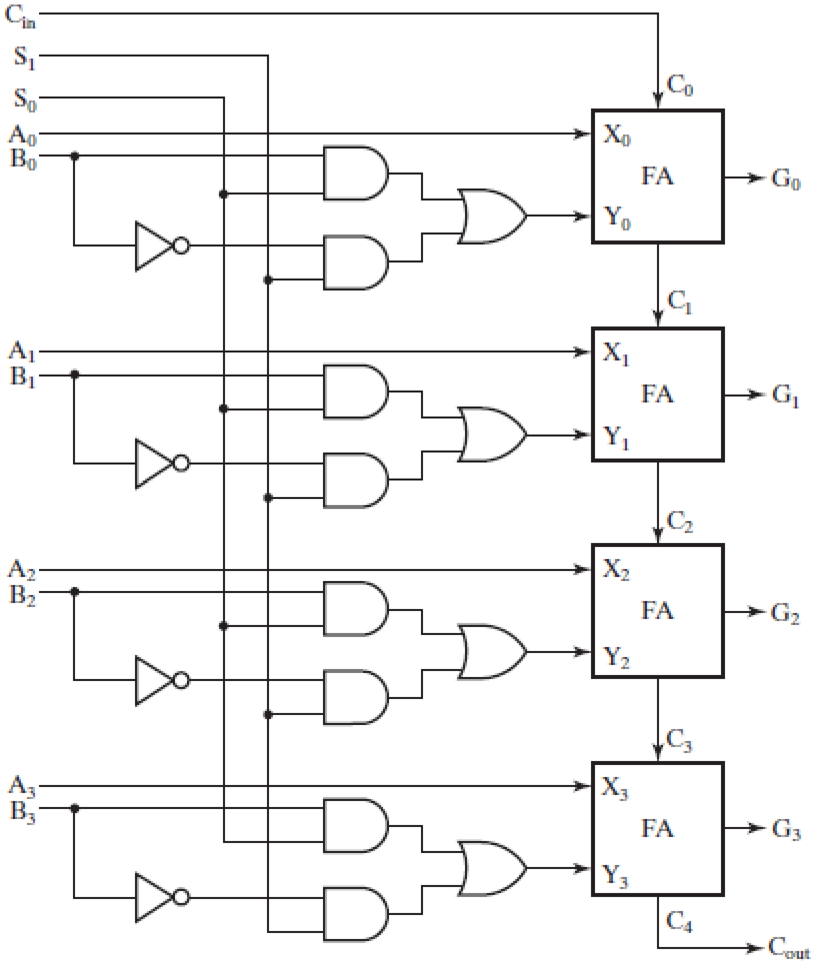
Arithmetic Circuit

- Next Figure shows the logic diagram of an arithmetic circuit for n = 4. The four full-adder (FA) circuits constitute the parallel adder. The carry into the first stage is the input carry Cin. All other carries are connected internally from one stage to the next.



Arithmetic Circuit

- Next Figure shows the logic diagram of an arithmetic circuit for n = 4. The four full-adder (FA) circuits constitute the parallel adder. The carry into the first stage is the input carry Cin. All other carries are connected internally from one stage to the next.


C_{in}
S_1
S_0
A_0
B_0
C_0
X_0
FA
Y_0
G_0
C_1
A_1
B_1
X_1
FA
Y_1
G_1
C_2
A_2
B_2
X_2
FA
Y_2
G_2
C_3
A_3
B_3
X_3
FA
Y_3
G_3
C_4
C_{out}