SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with 'A' Grade by NAAC

Lecture session 3_ UNIT-3

Unit-3-COMBINATIONAL LOGIC

CARRY LOOK A HEAD ADDER,BCD ADDER

By

V.GEETHA

ASSISTANT PROFESSOR/EEE

SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY

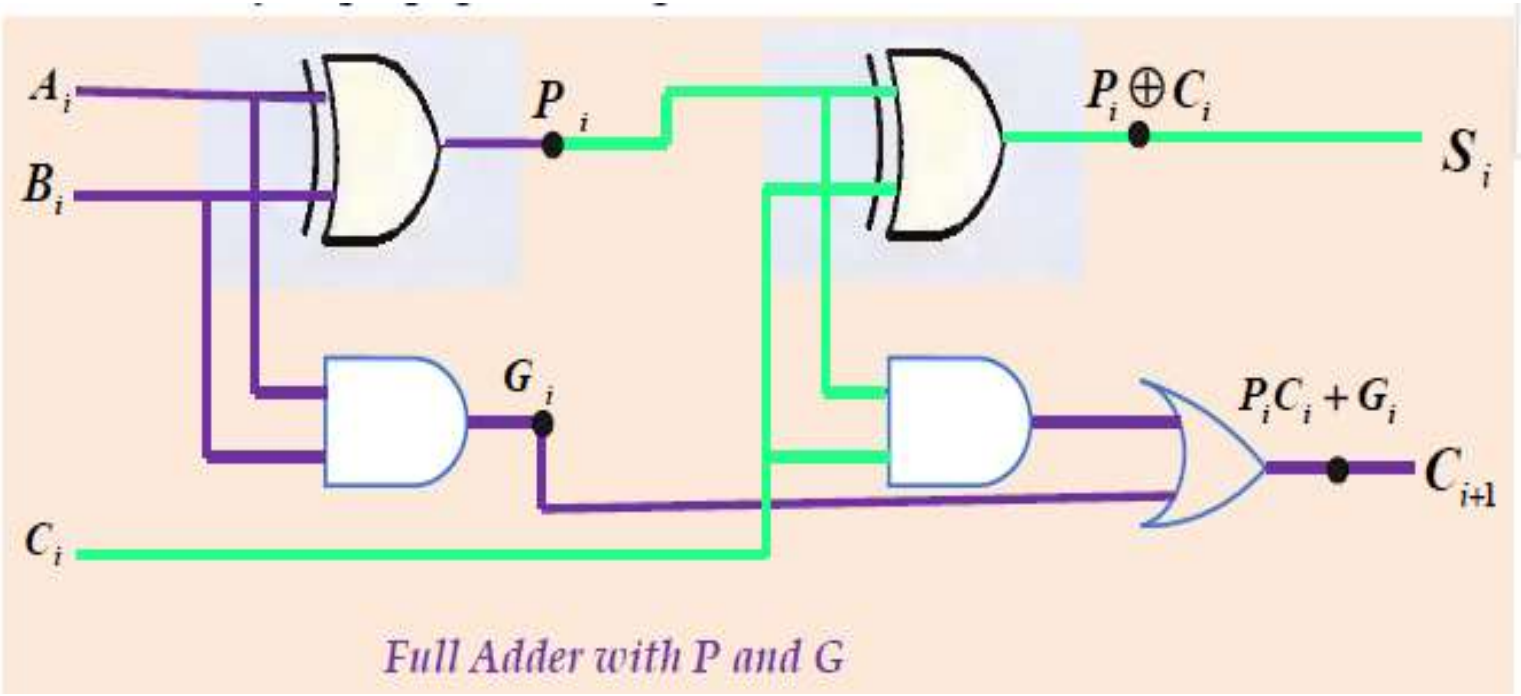
CHENNAI-119

CARRY LOOK A HEAD ADDER

The carry propagation time is an important attribute of the adder because it limits the speed with which two numbers are added.

To reduce the carry propagation delay time:

1) Employ faster gates with reduced delays.
2) Employ the principle of Carry Look a head Logic.



Full Adder with P and G

***Proof**: (using carry lookahead logic)*

$$P_i = A_i \oplus B_i$$

$$G_i = A_i B_i$$

The output sum and carry are:

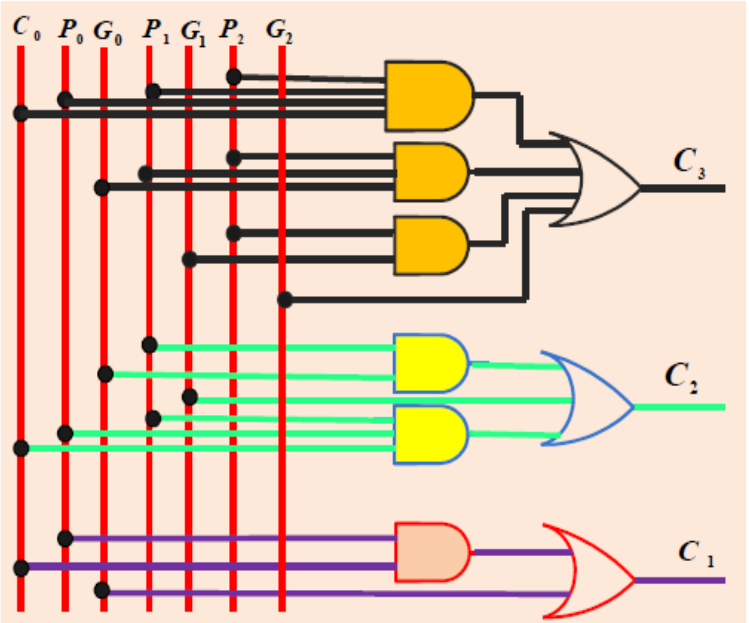
$$S_i = P_i \oplus C_i$$

$$C_{i+1} = G_i + P_i C_i$$

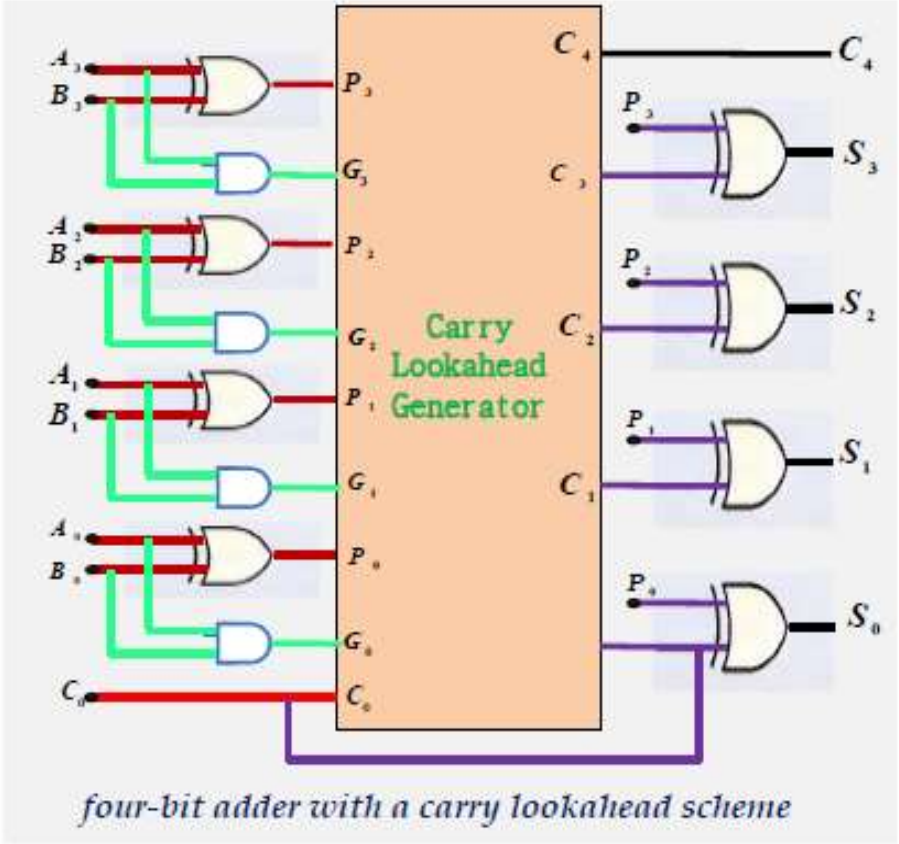
- ✓ G_i-called a ***carry generate***, and it produces a carry of 1 when both A_i and B_i are 1.
- ✓ P_i-called a ***carry propagate***, it determines whether a carry into stage i will propagate into stage $i + 1$.
- ✓ The ***Boolean function*** for the carry outputs of each stage and substitute the value of each C_i form the previous equations:

$$\begin{cases} C_0 = input\ carry \\ C_1 = G_0 + P_0 C_0 \\ C_2 = G_1 + P_1 C_1 = G_1 + P_1 (G_0 + P_0 C_0) \\ \quad = G_1 + P_1 G_0 + P_1 P_0 C_0 \\ C_3 = G_2 + P_2 C_2 = G_2 + P_2 (G_1 + P_1 G_0 + P_1 P_0 C_0) \\ \quad = G_2 + P_2 G_1 + P_2 P_1 G_0 + P_2 P_1 P_0 C_0 \end{cases}$$

- ➢ The three Boolean functions C_1, C_2 and C_3 are implemented in the *carry lookahead generator.*



Logic Diagram for Carry Lookahead Generator



four-bit adder with a carry lookahead scheme

The two level-circuit for the output carry C_4 is not shown, it can be easily derived by the equation.

- C_3 does not have to wait for C_2 and C_1 to propagate, in fact C_3 is propagated at the same time as C_1 and C_2.

BCD ADDER:

A BCD adder should perform the following

- Add two 4-bit BCD number using straight Binary Addition.

- If the sum of two numbers is equal to or less than 9, then the sum is valid BCD number and no correction is required.

- If the sum of two numbers is greater than 9 or carry is generated from the sum, then the sum is not valid BCD number. Then add 0110 (6) to the sum, the result will be valid BCD number. If further a carry is generated then add 0110 to the result.

Assume that two 4-bit BCD numbers are being added. In BCD addition, there are three different cases. Let us consider one by one.

CASE 1: Sum equal to or less than 9 with carry 0

	DECIMAL	BCD
	2	0 0 1 0
	+ 3	0 0 1 1
Carry		1
SUM	5	0 0 1 0 1

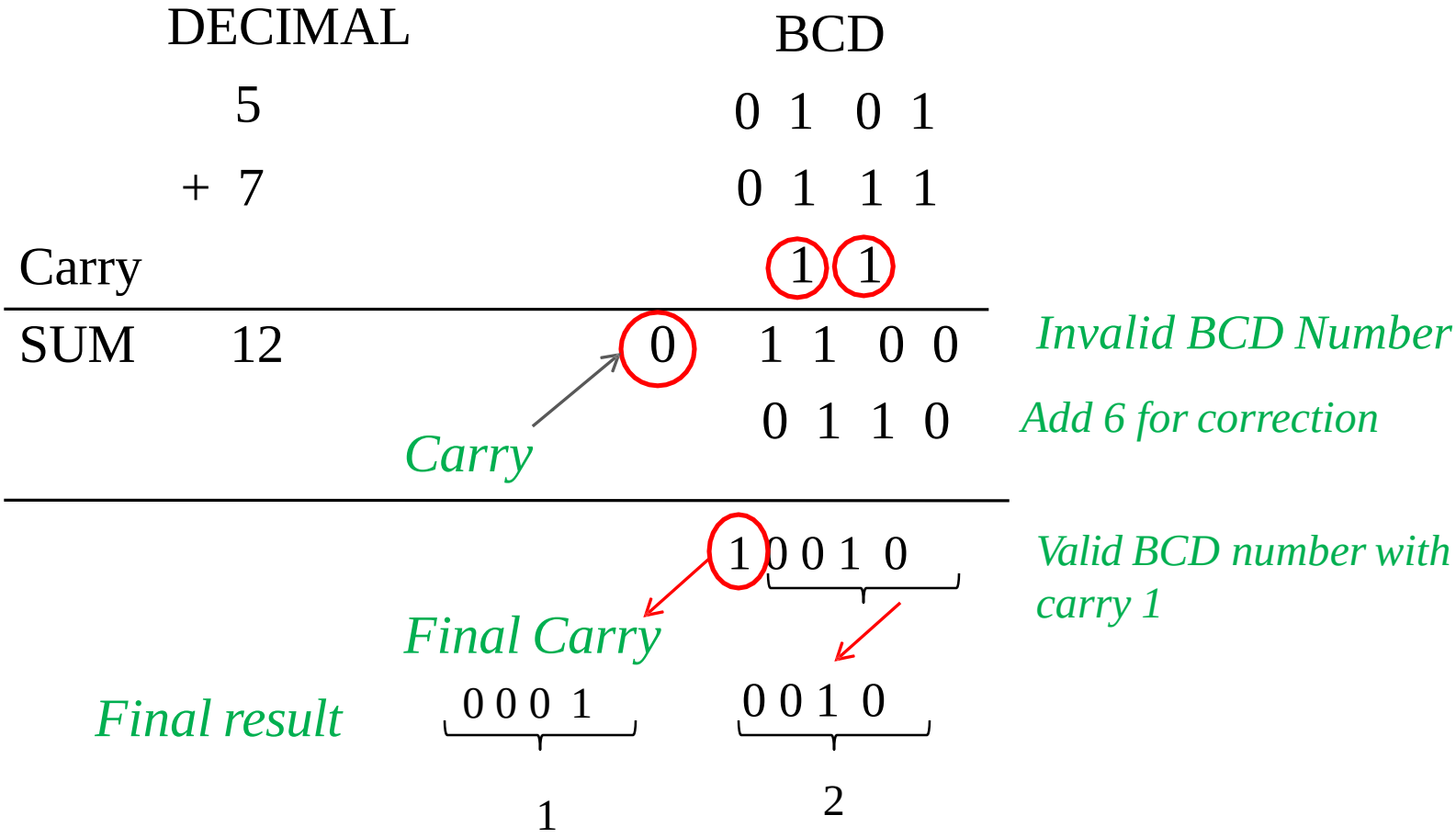
Final Carry

Valid BCD Number

If the sum of two numbers is less than or equal to 9 with final carry zero, then the sum is valid BCD number and no correction is required.

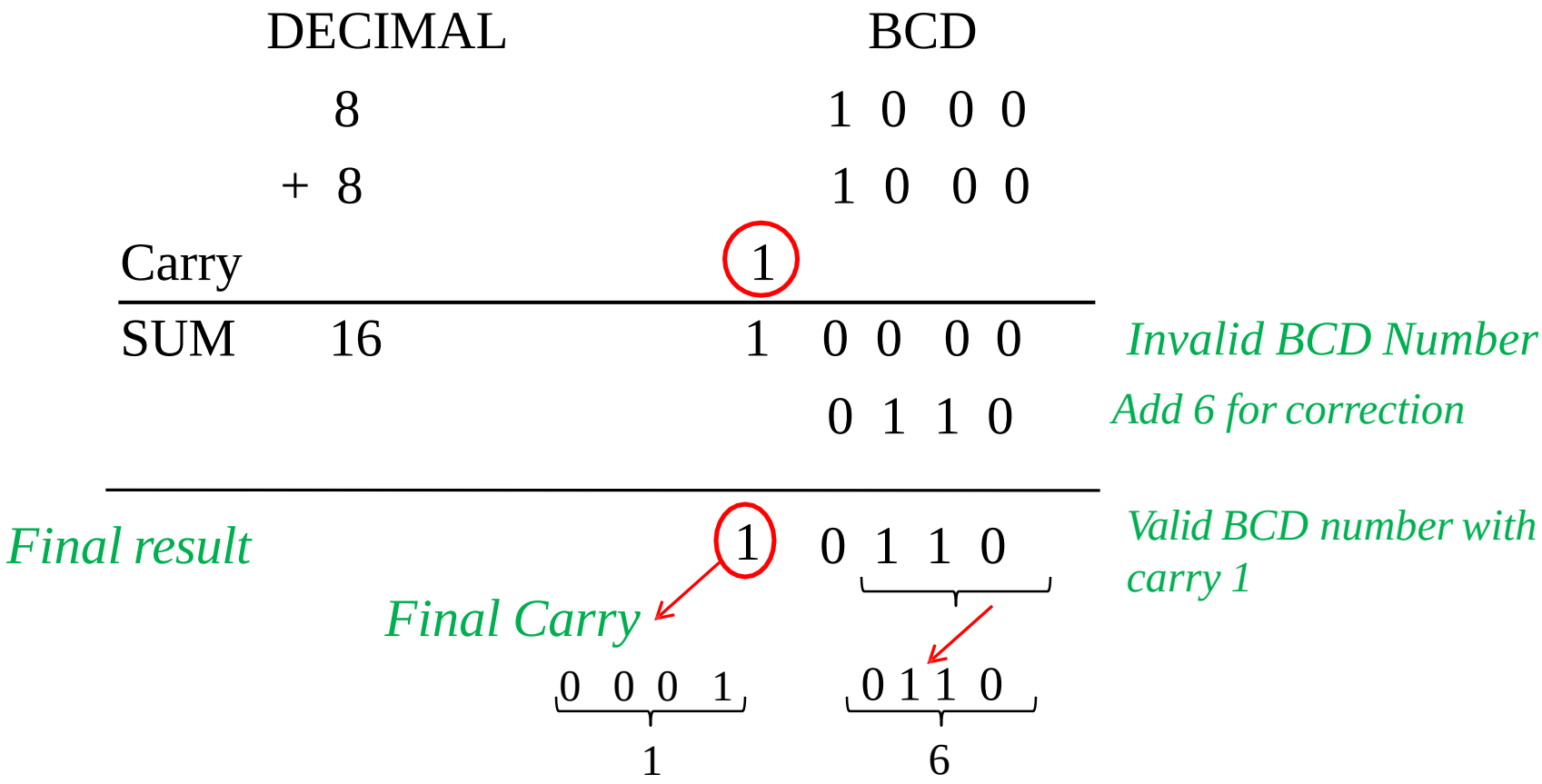
Consider second case

CASE 2: Sum greater than 9 with carry 0



If the sum of two numbers is greater than 9 with carry zero , and the sum is invalid BCD number, then correction is required. Add 0 1 1 0 (6) for correction. Final result will be the valid BCD number.

CASE 3: Sum is equal to or less than 9 with carry 1



If the sum of two numbers is equal to or less than than 9 with carry 1 , and the sum is invalid BCD number, then correction is required. Add 0 1 1 0 (6) for correction. Final result will be the valid BCD number.

DESIGN:

- In fig. 1 $B_3B_2B_1B_0$ and $A_3A_2A_1A_0$ are the BCD inputs. $S_3S_2S_1S_0$ and C_{out} is the output of Adder 1.
- Checked the output of Adder 1, whether it is greater than or less than 9.
- If the sum of Adder 1 is greater than 9 then the output of combinational circuit should be 1 (i.e C_{out} should be high) and correction is required.
- Write the truth table and K-Map for combinational circuit.
- The Boolean Expression from K-Map $Y= S_3S_2+S_3S_1$
- The output of combinational circuit Y_2 is connected to B_2B_1 of Adder 2 and B_3B_0 are connected to the ground. Therefore $B_3=B_0=0$
- The output sum of Adder 1 is connected to Adder 2. the output of Adder 2 is the final result of BCD addition with Carry which can be ignored if any.

TRUTH TABLE

INPUTS				OUTPUTS
S3	S2	S1	S0	Y_1
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

Sum is valid BCD number. Therefore Y=0

Sum is valid BCD number. Therefore Y=1

K-Map

TRUTH TABLE

INPUTS				OUTPUTS
S3	S2	S1	S0	Y_1
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

Sum is valid BCD number. Therefore Y=0

Sum is valid BCD number. Therefore Y=1

K-Map

S_3S_2 \ S_1S_0	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	1	1	1	1
10	0	0	1	1

TRUTH TABLE

INPUTS				OUTPUTS
S3	S2	S1	S0	Y_1
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

Sum is valid BCD number. Therefore Y=0

Sum is valid BCD number. Therefore Y=1

K-Map

S_3S_2 \ S_1S_0	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	1	1	1	1
10	0	0	1	1

$Y_1=S_3S_2+S_3S_1$

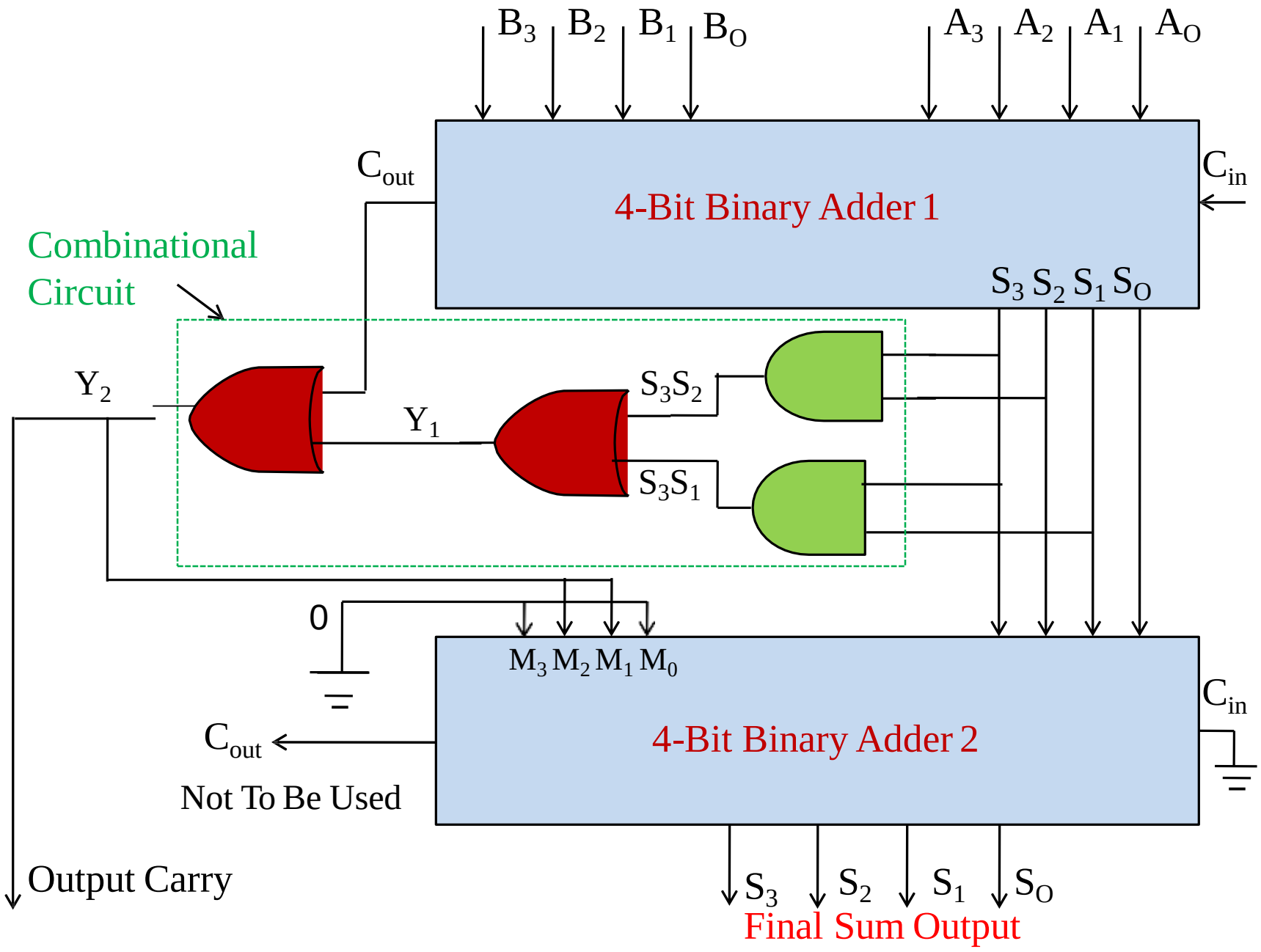


Fig. 1: BLOCK DIAGRAM OF BCD ADDER

CASE 1: Sum equal to or less than 9 with carry 0

DECIMAL		BCD
	2	0 0 1 0
+	1	0 0 0 1
Carry		0
SUM	3	0 0 0 1 1

Final Carry

Valid BCD Number

Output of combinational circuit $Y_2 = 0$, $M_3 M_2 M_1 M_0 = 0$

Output of Adder 2 is same as the output of Adder 1

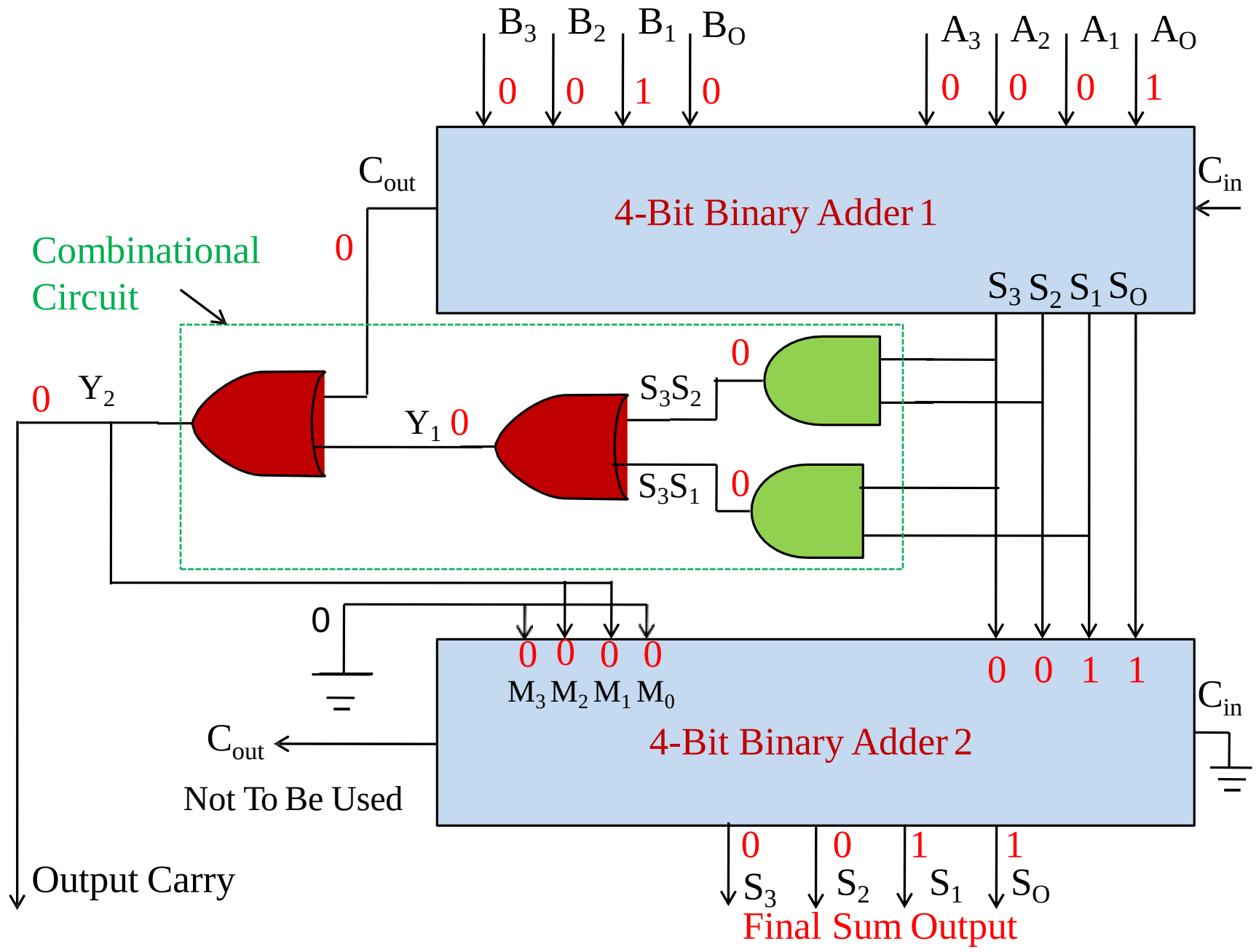
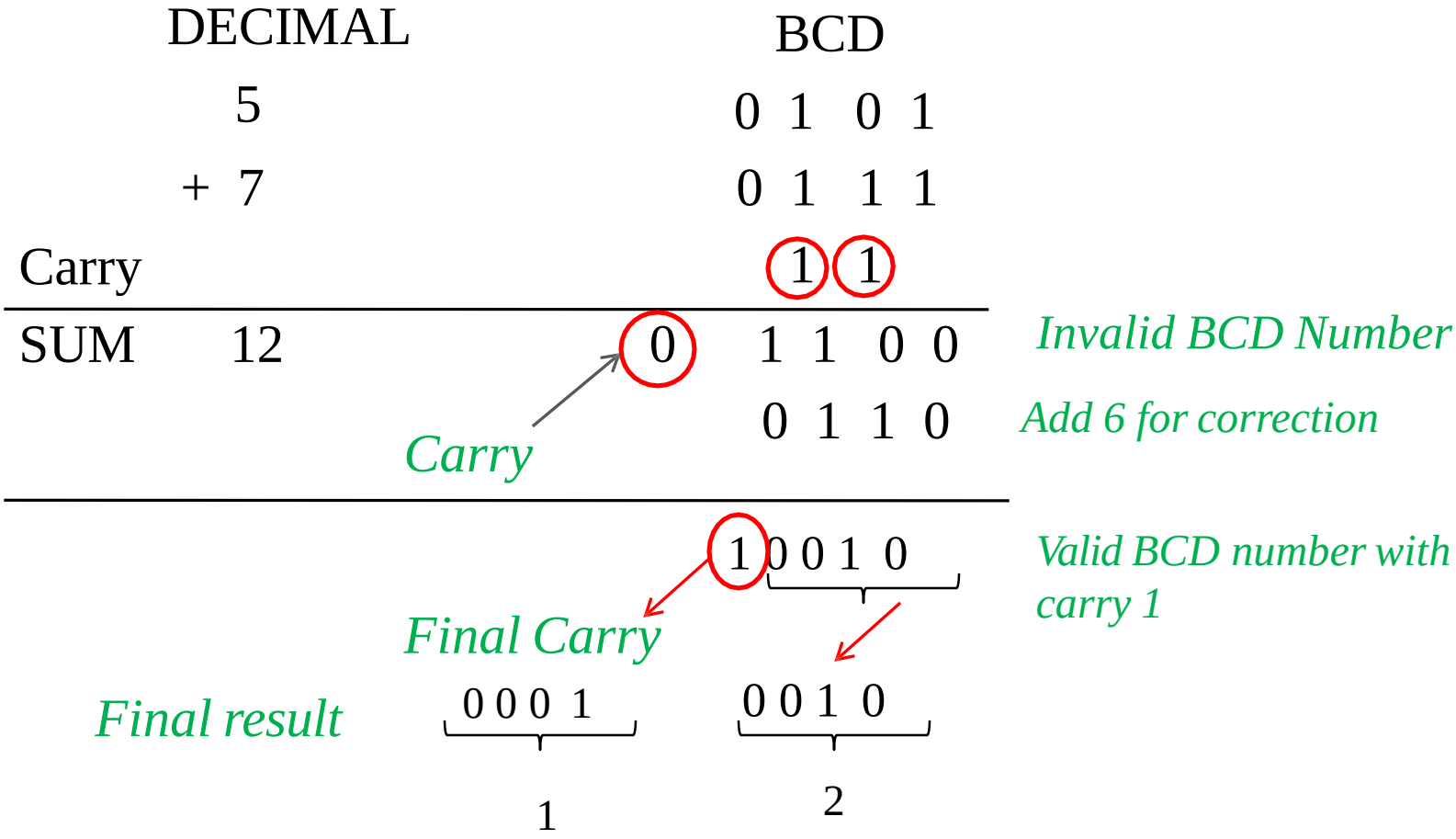


Fig. 1: BLOCK DIAGRAM OF BCD ADDER

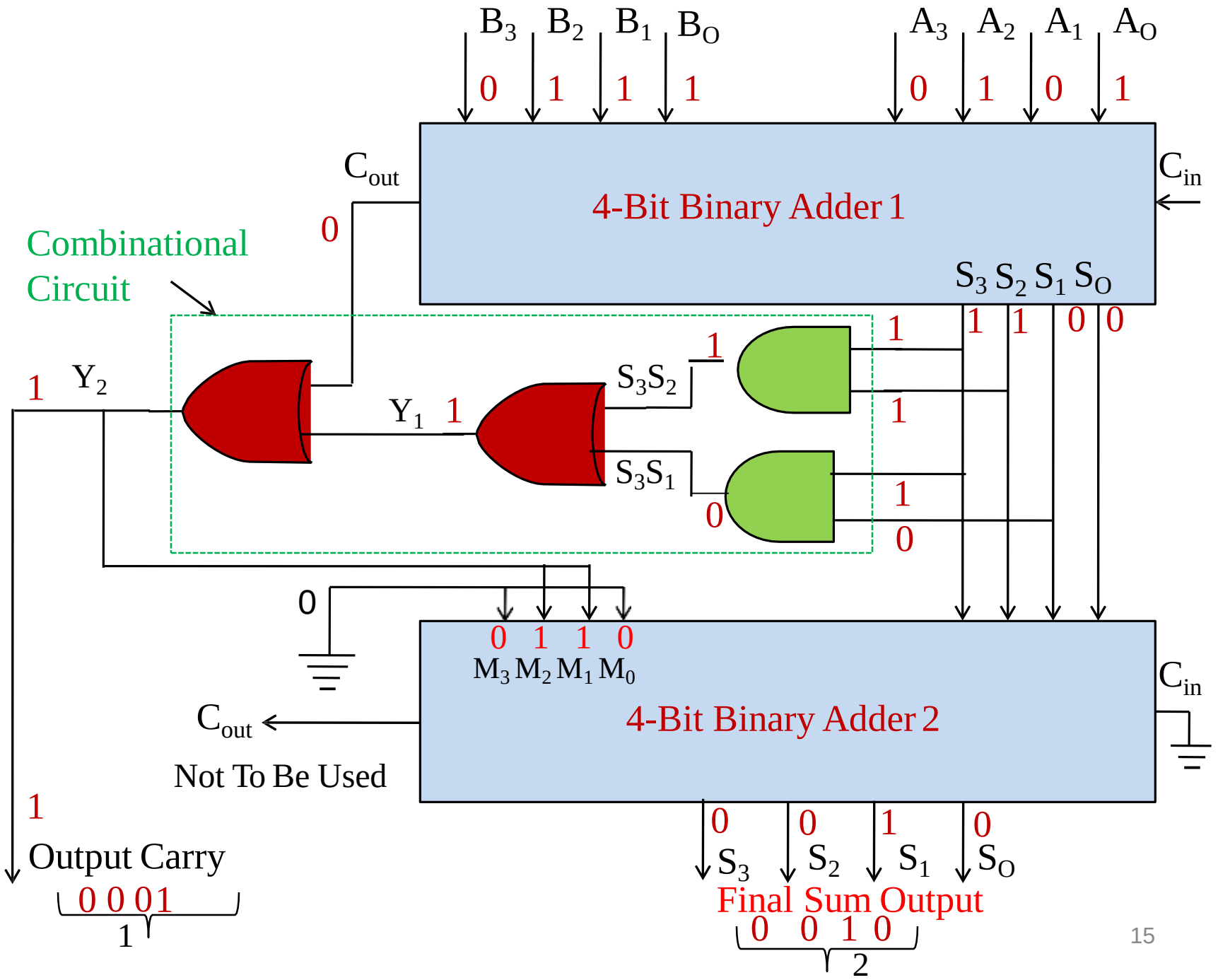
Consider second case

CASE 2: Sum greater than 9 with carry 0



Output of combinational circuit $Y_2 = 0$, $M_3\, M_2\, M_1\, M_0 = 0\ 1\ 1\ 0$

Fig. 1: BLOCK DIAGRAM OF BCD ADDER



CASE 3: Sum is equal to or less than 9 with carry 1

	DECIMAL	BCD	
	8	1 0 0 0	
	+ 8	1 0 0 0	
Carry		1	
SUM	16	1 0 0 0 0	*Invalid BCD Number*
		0 1 1 0	*Add 6 for correction*
Final result		1 0 1 1 0	*Valid BCD number with carry 1*

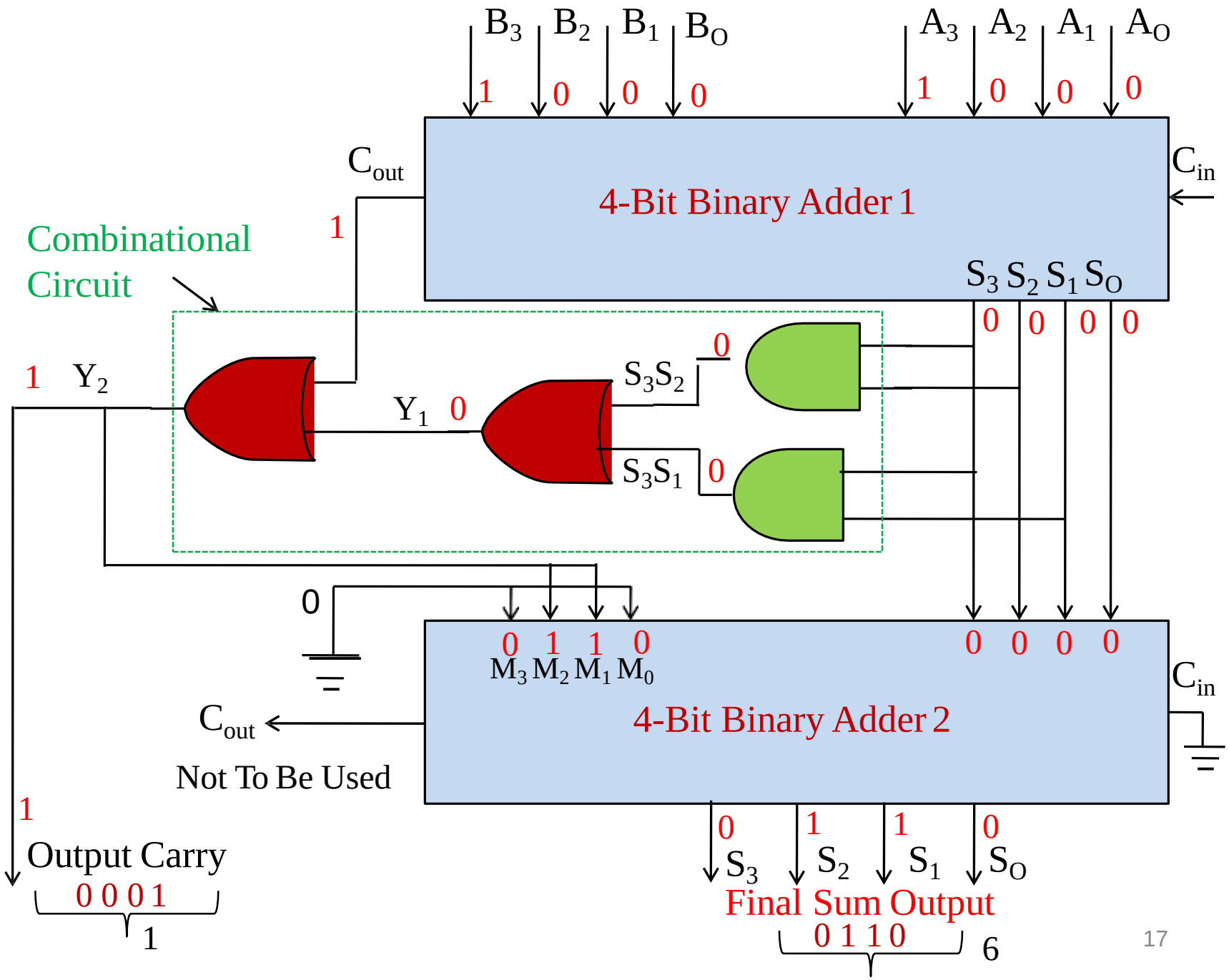
Final Carry

0 0 0 1 — 1

0 1 1 0 — 6

Output of combinational circuit $Y_2 = 1$, $M_3\, M_2\, M_1\, M_0 = 0\ 1\ 1\ 0$

Fig. 1: BLOCK DIAGRAM OF BCD ADDER



Thank you