

Name : Binary Arithmetic.

A. Objectives :

- + Understand the concept of binary addition and subtraction.
- + Learn about half and full binary adders.
- + Perform binary addition and subtraction using IC 7483.
- + Understand the concept of BCD addition and implement a BCD adder using IC 7483.

B. Theory:

Digital computers perform a variety of information-processing tasks. Among the functions encountered are the various arithmetic operations. The most basic arithmetic operation is the addition of

two binary digits. This simple addition consists of four possible elementary operations: $0+0=0$, $0+1=1$, $1+0=1$ and $1+1=10$. The first three operations produce a sum of one digit, but when both augend and addend bits are equal to 1, the binary sum consists of two digits. The higher significant bit of this result is called a carry. When the augend and addend numbers contain more significant digits, the carry obtained from the addition of two bits is added to the next higher order pair of significant bits. A combinational circuit that performs the addition of two bits is called a half adder. One that performs the addition of three bits (two significant bits and a previous carry) is a

full adder. The names of the circuits stem from the fact that two half adders can be employed to implement a full adder.

Full Adder

Half Adder

x	y	c	s
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

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

Experiment 1: Binary Adder-Subtractor

C.1 Apparatus

- + Trainer board
- + 1x IC 7483 4bit binary adder.
- + 1x IC 7486 quadruple 2-Input XOR gates.

E.1 Report

(1) In 4bit adder-subtractor, we have used the XOR gate and data B (B_1, B_2, B_3, B_4) is the input. We have to take 2 inputs for per XOR gate. So, we have used "M" for another input to get the right output. In no. 13 pin is used for M bit. And M is connected by all the XOR gate. M is known as control bit. It controls the whole output of XOR gate. Here,
 $M=0/1$. $M=0$ (for add) and $M=1$ (for subtract).

(2)

A	B	Cin	Sum	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

Table : 4-bit adder Truth table.

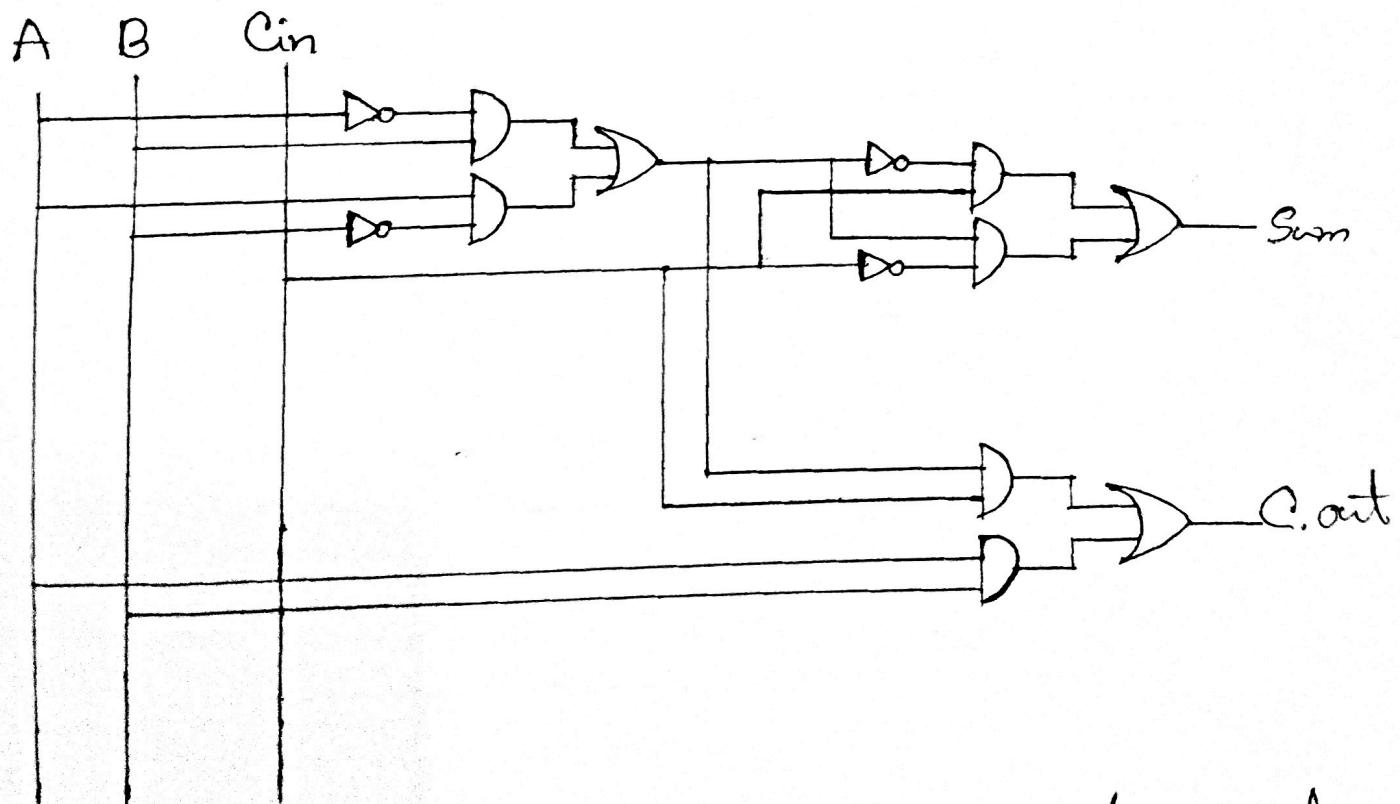


Figure : 4-bit adder using basic logic gates.

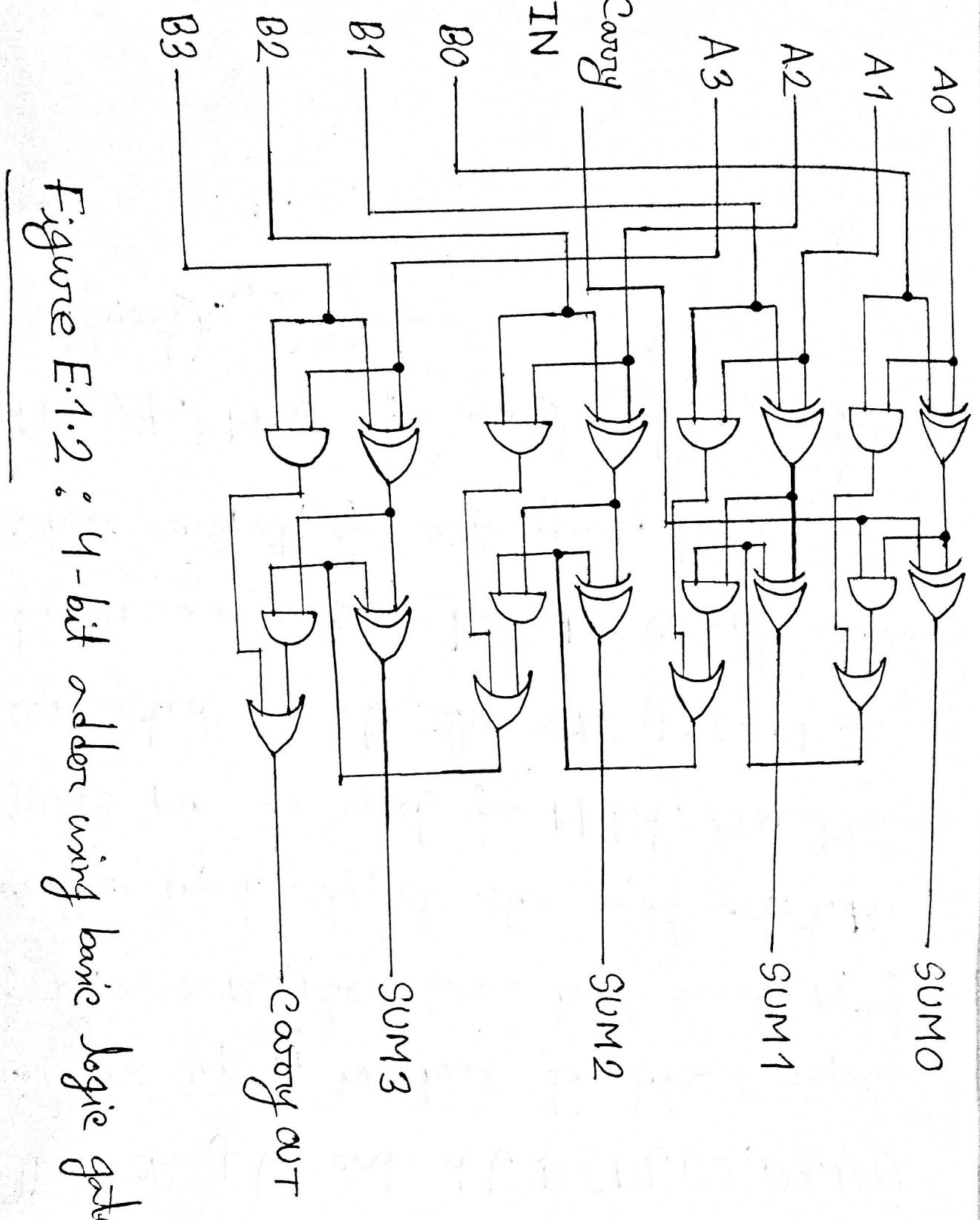


Figure E.1.2 : 4-bit adder using basic logic gates.

Experiment 2 : Ripple - Through - Carry Adder

C.2 Apparatus

- Trainer board
- 2x IC 7483 4 bit-binary adder.

E2 : Report :

(1)

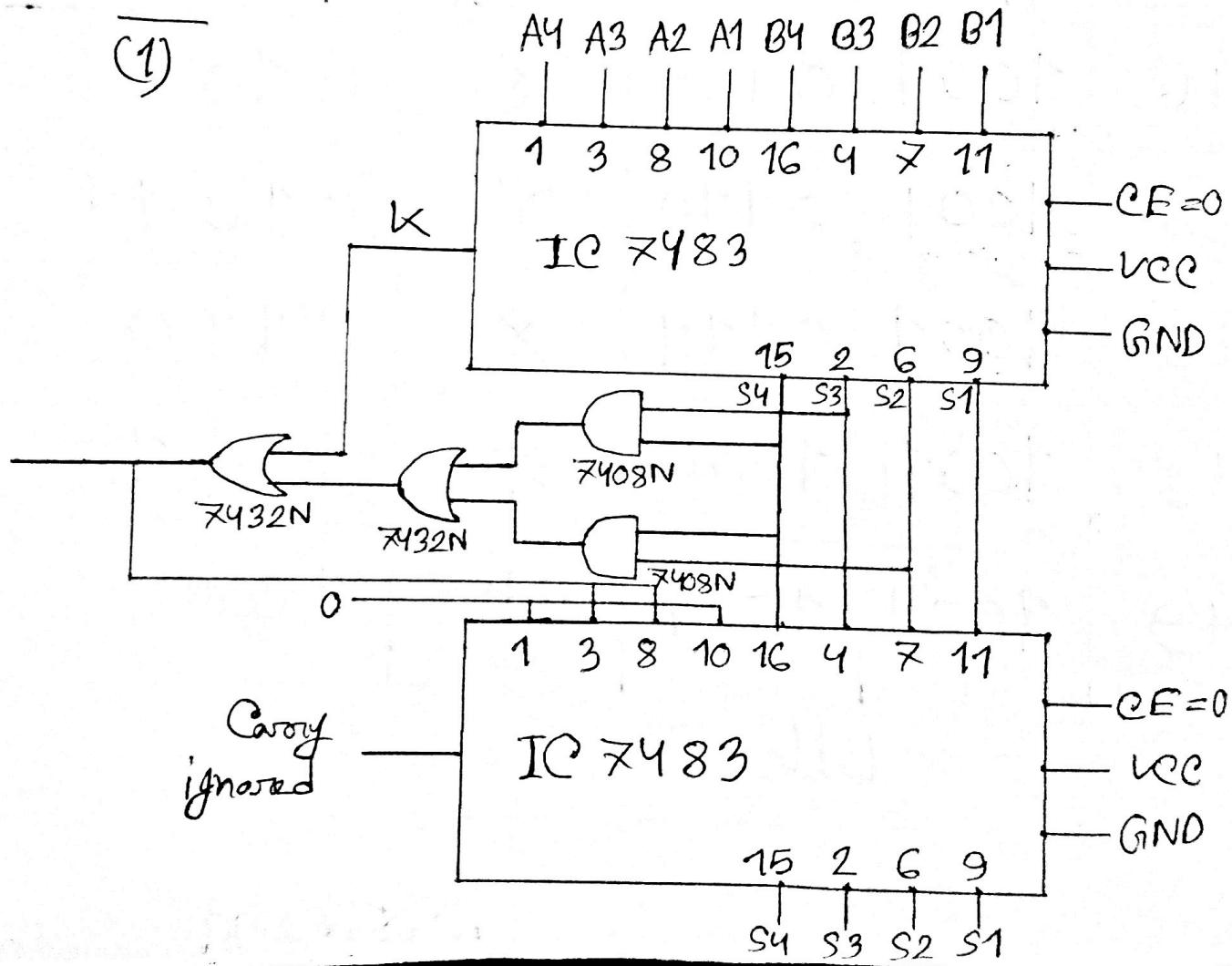


Figure: IC diagram for the 8-bit ripple-through carry adder.

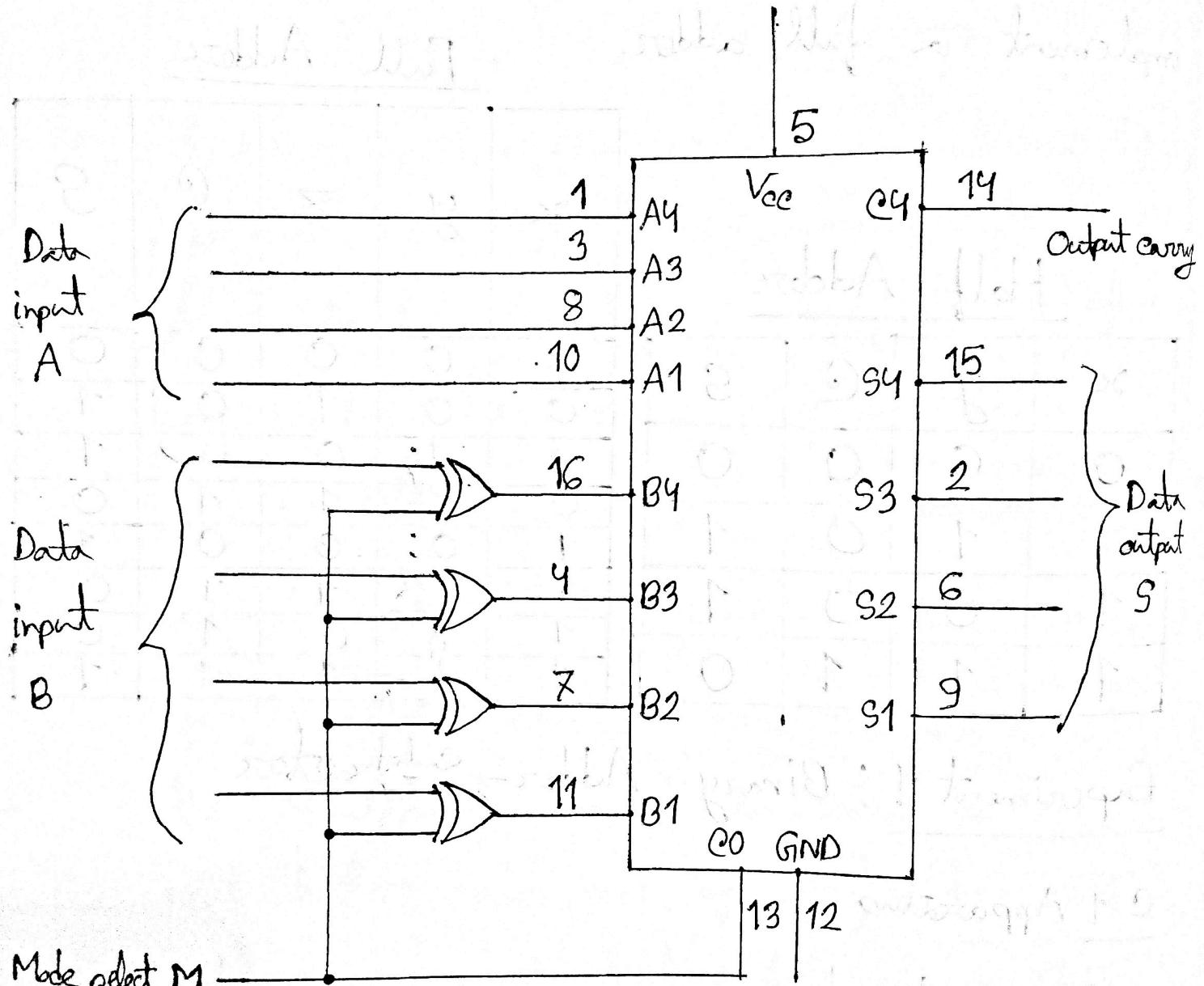
Experiment 3 : BCD Adder

C.3 Apparatus

- * Trainer board
- * $2 \times$ IC 7483 4-bit binary adder
- * $1 \times$ IC 7408 quadruple 2-Input AND gates
- * $1 \times$ IC 7432 quadruple 2-Input OR gates.

E.3 Report:

(1)



Mode select M

M = 0 for add

M = 1 for subtract

Figure D.1.1: 4-bit adder-subtractor

F.1: Experimental Data (Binary Adder-Subtractor)

Operation	M	A	B	C4	S4	S3	S2	S1
$7+5$	0	0111	0101	0	1	1	0	0
$4+6$	0	0100	0110	0	1	0	1	0
$9+11$	0	1001	1011	1	0	1	0	0
$15+15$	0	1111	1111	1	1	1	1	0
$7-5$	1	0111	0101	0	0	0	1	0
$4-6$	1	0100	0110	1	1	1	1	0
$11-2$	1	1011	0010	0	1	0	0	1
$15-15$	1	1111	1111	0	0	0	0	0

Table F.1.1

F.2 : Experimental Data (Ripple-Through-Carry Adder).

Operation	A	B	Overflow Carry	Sum
$7+5$	0000 0111	0000 0101	0	0000 1100
$18+19$	0001 0010	0001 0011	0	0010 0101
$72+83$	0100 1000	0101 0011	0	1001 1011
$129+255$	1000 0001	1111 1111	1	1000 0000

Table F.2.1

F.3 : Experimental Data (BCD Adder).

Decimal Value	Binary Sum					BCD Sum				
	K	Z_3	Z_2	Z_1	Z_0	C	S_3	S_2	S_1	S_0
0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	1	0	0	0	0	1
2	0	0	0	1	0	0	0	0	1	0
3	0	0	0	1	1	0	0	0	1	1
4	0	0	1	0	0	0	0	0	1	0
5	0	0	1	0	1	0	0	1	0	1
6	0	0	1	1	0	0	0	1	1	0
7	0	0	1	1	1	0	0	1	1	1
8	0	1	0	0	0	0	0	1	0	0
9	0	1	0	0	1	0	1	0	0	0
10	0	1	0	1	0	1	0	0	0	1
11	0	1	0	1	1	1	0	0	0	0
12	0	1	1	0	0	1	0	0	0	1
13	0	1	1	0	1	1	0	0	1	0
14	0	1	1	1	0	1	0	0	1	1
15	0	1	1	1	1	1	0	1	0	0
16	1	0	0	0	0	1	0	1	0	1
17	1	0	0	0	1	1	0	1	1	0
18	1	0	0	1	0	1	1	1	1	1
19	1	0	0	1	1	1	1	0	0	1

Table F.3-1

Operation	A	B	Overflow Carry	Sum
9+0	1001	0000	0	1001
9+1	1001	0001	1	0000
9+2	1001	0010	1	0001
9+3	1001	0011	1	0010
9+4	1001	0100	1	0011
9+5	1001	0101	1	0100
9+6	1001	0110	1	0101
9+7	1001	0111	1	0110
9+8	1001	1000	1	0111
9+9	1001	1001	1	1000

Table F.3.2