BCD adder (or) Decimal adder (0 to 9)

Tuesday, October 13, 2020 8:40 AM

case 1 : Sum equal to 9 (or) less with carry = 0

```
   1 1
   6 → 0110 → BCD for 6
 + 3 → 0011 → BCD for 3
 ---   ----
  +9   1001 → BCD for 9 → No Correction
```

normal addition

case 2 :- Sum greater than 9 with carry = 0

```
   7 → 0111 → BCD for 7
 + 8 → 1000 → BCD for 8
 ---   ----
  15   1111 → Invalid BCD
```

$(15)_{10}$ = 0001 0101 = BCD for 15

```
   1 1
Invalid BCD = 1111
```

to correct it add (+6) = 0110

```
000 1 0101 → Correct BCD
 1    5
```

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

```
 8 → 1000
+9 → 1001
---   ----
17  1 0001 → Invalid BCD
```

17 → 1 0001 → Invalid BCD

carry = 1, Sum ≤ 9

$(17)_{10} = (0001\ 0111)_{BCD}$

(0001 = 1, 0111 = 7)

Invalid BCD = 1 0001

To correct it add (+6) = + 0110

0001 0111 → correct BCD

(0001 = 1, 0111 = 7)

Summary of BCD addition

1. Add two BCD numbers using ordinary binary addition
2. If four bit sum is equal to or less than 9 no correction is needed. Sum is in BCD form
3. If four bit sum greater than 9 (or)
   If carry generated from the four bit sum. Then sum is invalid BCD.
   To correct it add +6 (0110) to the Invalid BCD number.

To implement BCD adder

1. 4 bit parallel adder for initial addition
2. Logic ckt to detect sum greater 9
3. one more 4 bit parallel adder

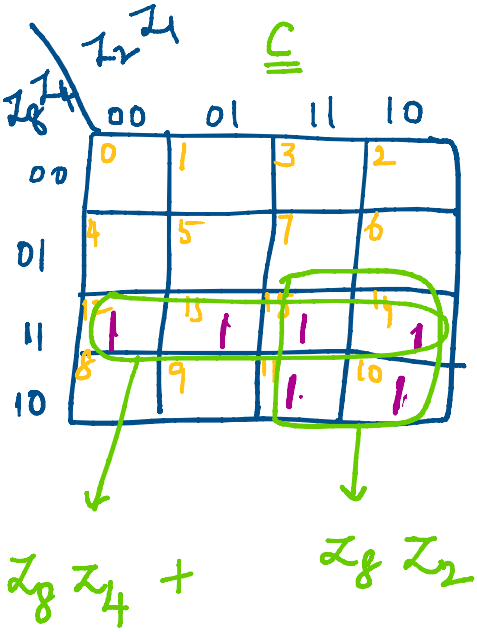
BCD numbers (0 to 9) ✓

$(10)_{10} = (0001\ 0000)_{BCD}$

$(15)_{10} = (0001\ 0101)_{BCD}$

Binary Sum						BCD Sum						Decimal
K	Z_8	Z_4	Z_2	Z_1		C	S_8	S_4	S_2	S_1		
0	0	0	0	0	S	0	0	0	0	0		0
0	0	0	0	1	A	0	0	0	0	1		1
0	0	0	1	0	M	0	0	0	1	0		2
.	.	.	.	.	E	.	.	.	.	.		.
.	.	.	.	.		.	.	.	.	.		.
.	.	.	.	.	C	.	.	.	.	.		.
.	.	.	.	.	O	.	.	.	.	.		.
0	1	0	0	0	D	0	1	0	0	0		8
0	1	0	0	1	E	0	1	0	0	1		9
10 to 19 Binary and BCD codes are not the same												
0	1	0	1	0		1	0	0	0	0		10
0	1	0	1	1		1	0	0	0	1		11
0	1	1	0	0		1	0	0	1	0		12
0	1	1	0	1		1	0	0	1	1		13
0	1	1	1	0		1	0	1	0	0		14
0	1	1	1	1		1	0	1	0	1		15
1	0	0	0	0		1	0	1	1	0		16
1	0	0	0	1		1	0	1	1	1		17
1	0	0	1	0		1	1	0	0	0		18
1	0	0	1	1		1	1	0	0	1		19

carry, sum > 9, carry=0, carry=1, logic ckt, sum < 9



$$C = K + Z_8 Z_4 + Z_8 Z_2$$

1. It is obvious from the table that a correction is needed when the **'Binary Sum'** has an output carry K=1.
2. The other six combinations from 10 to 15 need correction in which the bit on the Z8 position is 1.

3. In the Binary sum of 8 and 9, the bit on the Z_8 position is also 1. So, the second step fails, and we need to modify it.
4. To distinguish these two numbers, we specify that the bit on the Z_4 or Z_2 position also needs to be 1 with the bit of Z_8
5. The condition for a correction and an output carry can be expressed by the Boolean function:

$C = K + Z_8.Z_4 + Z_8.Z_2$