→ Complements :-

→ (r-1)'s
→ r's

1) Binary → 1's, 2's
2) Decimal → 9's, 10's
3) Octal → 7's, 8's
4) Hexadecimal → F's, 16's

→ (r-1)'s Complement
↳ Subtract max. No to the given no

Prob-1 → 1010 → (r-1)'s

```
 1111
-1010
------
 0101  →
```

Prob → 10110 ↳ (r-1)'s Comp. ?

```
 11111
-10110
------
 01001 0  → (r-1) comp.
```

↳ 9's (r-1)'s → 2679 → (r-1)'s Comp. ?

```
max. 9999
   - 2679
   ------
   7320  → 9's complement
```

* r's Complement :-

↳ (r-1)'s Complement

↳ Add +1

prob1 → 10100 → 2's Complement ?

I → 1's complement

```
  11111
- 10100
-------
  01011   → 1's complement
      +1
-------
  01100   → r's or 2's complement
```

prob-2 → 5690 → Calculate the 10's complement ?

I → 9995 ← max

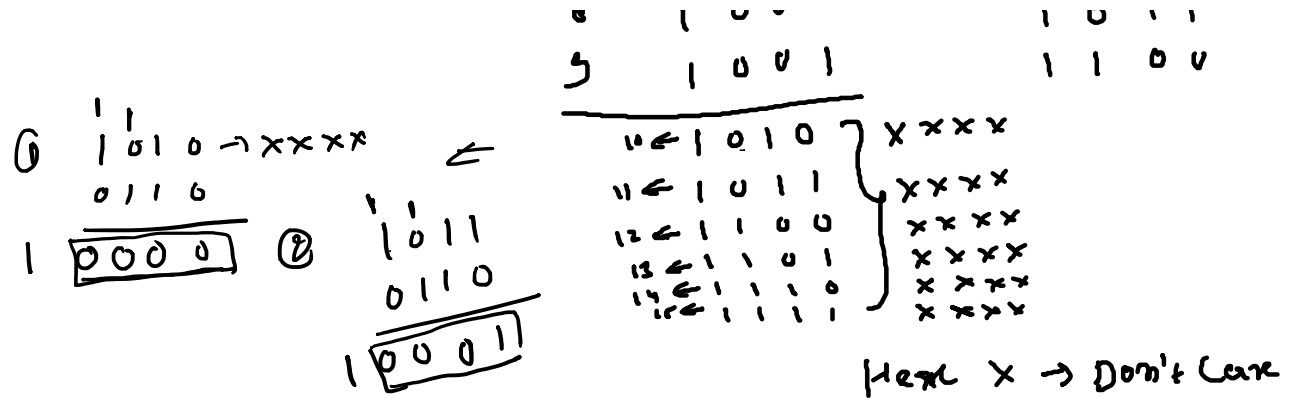
```
  9995
- 5690
-------
  4309   → 9's complement
     +1
-------
  4310   → 10's complement
```

* Codes → BCD

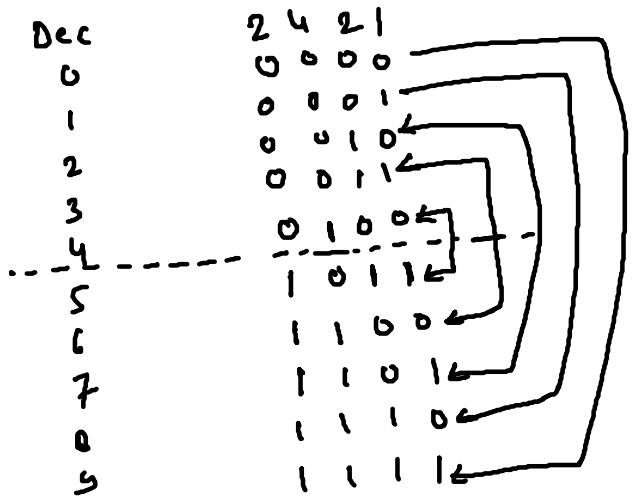
Excess-3 code

8421 code

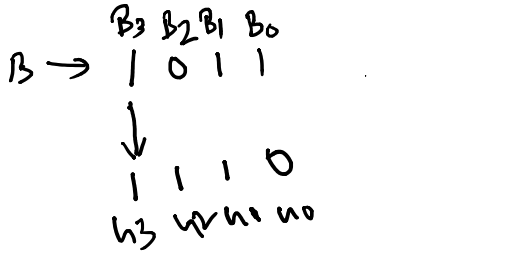
Dec	BCD	Excess-3-Code
0	0000	0011
1	0001	0100
2	0010	0101
3	0011	0110
4	0100	0111
5	0101	1000
6	0110	1001
7	0111	1010
8	1000	1011
9	1001	1100

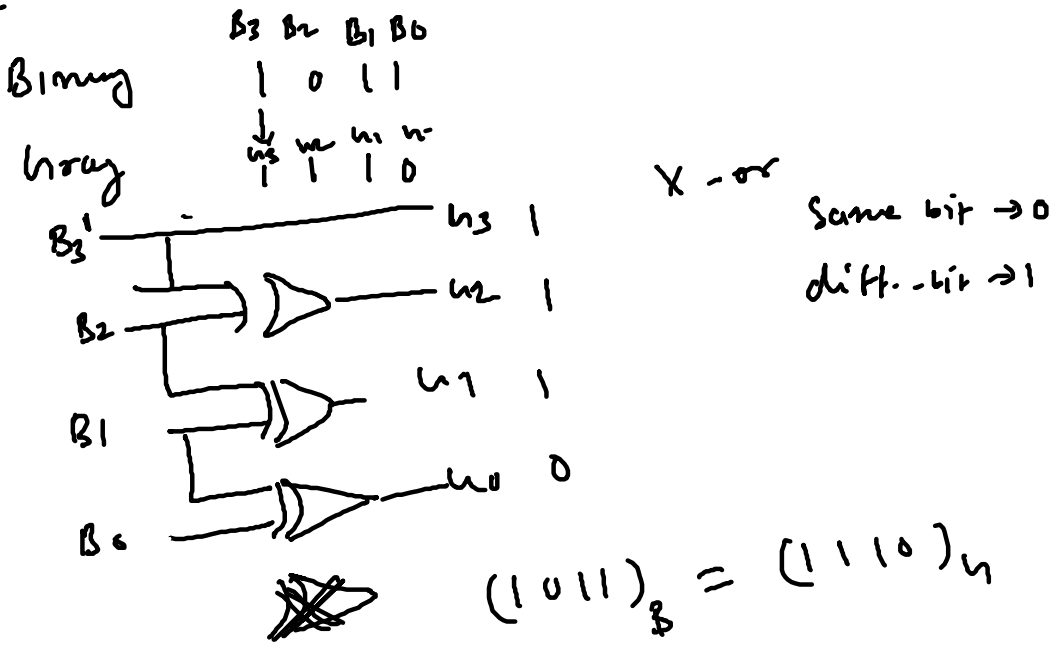


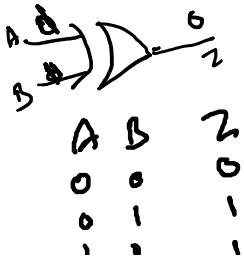
→ 2-4-2-1 Code



→ Binary to gray Code :-







0	0	0
0	1	1
1	0	1
1	1	0

$(1011)_B =$

Gray to Binary

$g_3\ g_2\ g_1\ g_0$

1 1 1 0

↓

1 0 1 1

B_3

Exp – $g_3\ g_2\ g_1\ g_0$: 1 1 1 0

↓

1 0 1 1

$B_3\ B_2\ B_1$

$(1110)_G = (1011)_B$