Problem solving :-

if The ASCII Value for 'A' is 65 convert this ASCII value to binary and store them as 8 dit data.

Answer :-

The ASCII value for 'A' is '65'

The Binary value for '65' = 11000001

Sender

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
0	1	0	0	0	0	0	1

→ 8 data bits,

Compute Parity bit

The total no. of 1's in the data unit is 2 so '0' is add to the data unit as even parity.

0	0	1	0	0	0	0	0	1	

Parity bit

0	0	1	0	0	0	0	0	1

↓

| transmitted |

↓

By wrongly the 'D_3' location bit is changed to '1'

0	0	1	0	0	1	0	0	1

↓

| Compute Parity bit |

So the "error is detect" Becaus of inappropriate Parity Bit".

Now the no. of bit's '1' is 3 But is odd but it has even Parity

reject ←No— ⟨even⟩ —Yes→ | Accept |

Hence the error detected :-

Error - Correcting:

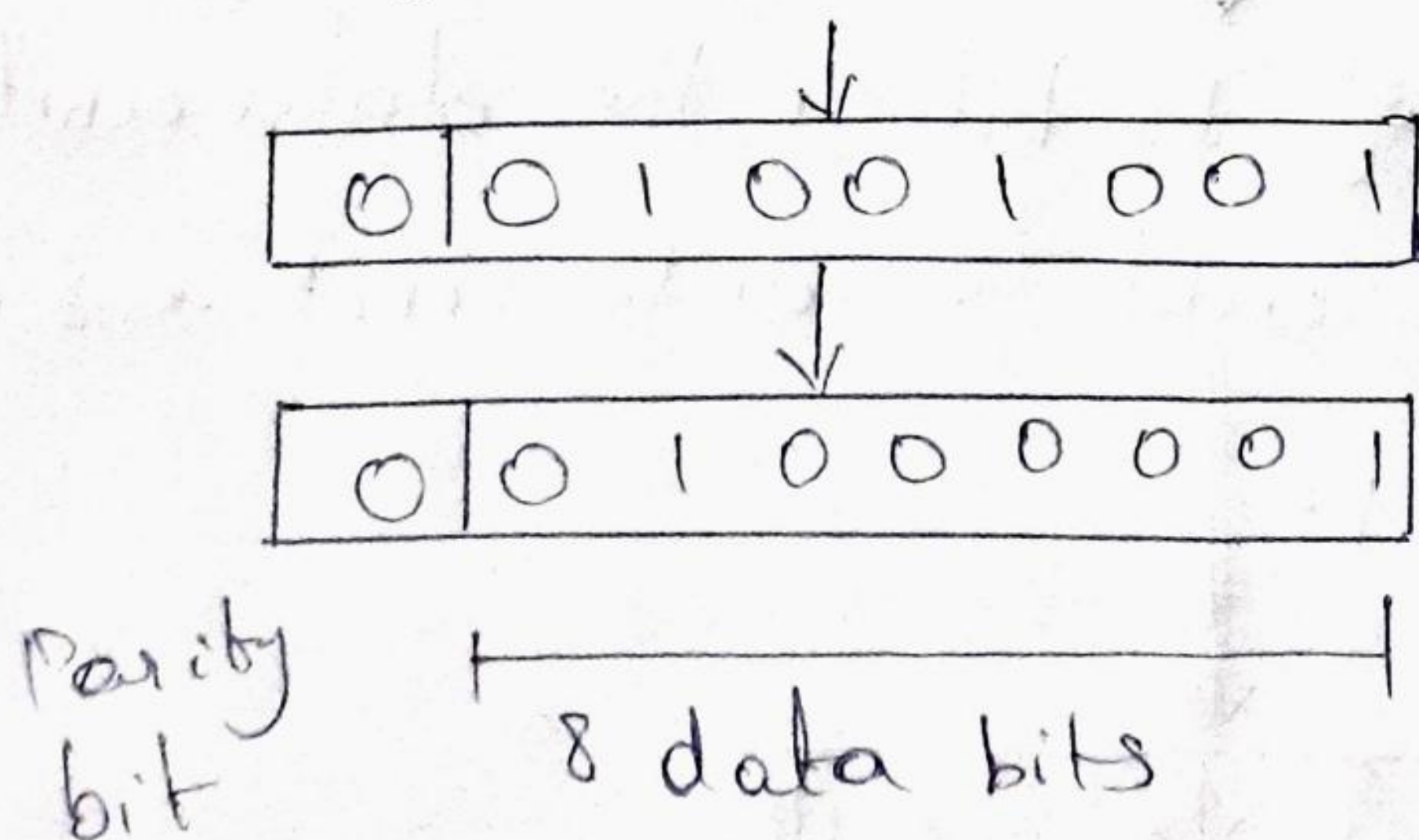* Along with error-detecting code, we can also Pass some data to figure out the original message

from the corrupt message that we received

A) In error-correcting, Parity Ceck has a simple way to detect errors along with a sophisticated mechanism to determin the corrupt bit location.

*) Once the corrupt bit es located. it's value is reverted (from 0 to 1 or 1 too) to get the original message.

→ By the use of error-correcting code it detect the exact location of the corrupt bits here as "D_3" Chang it '1' to '0'
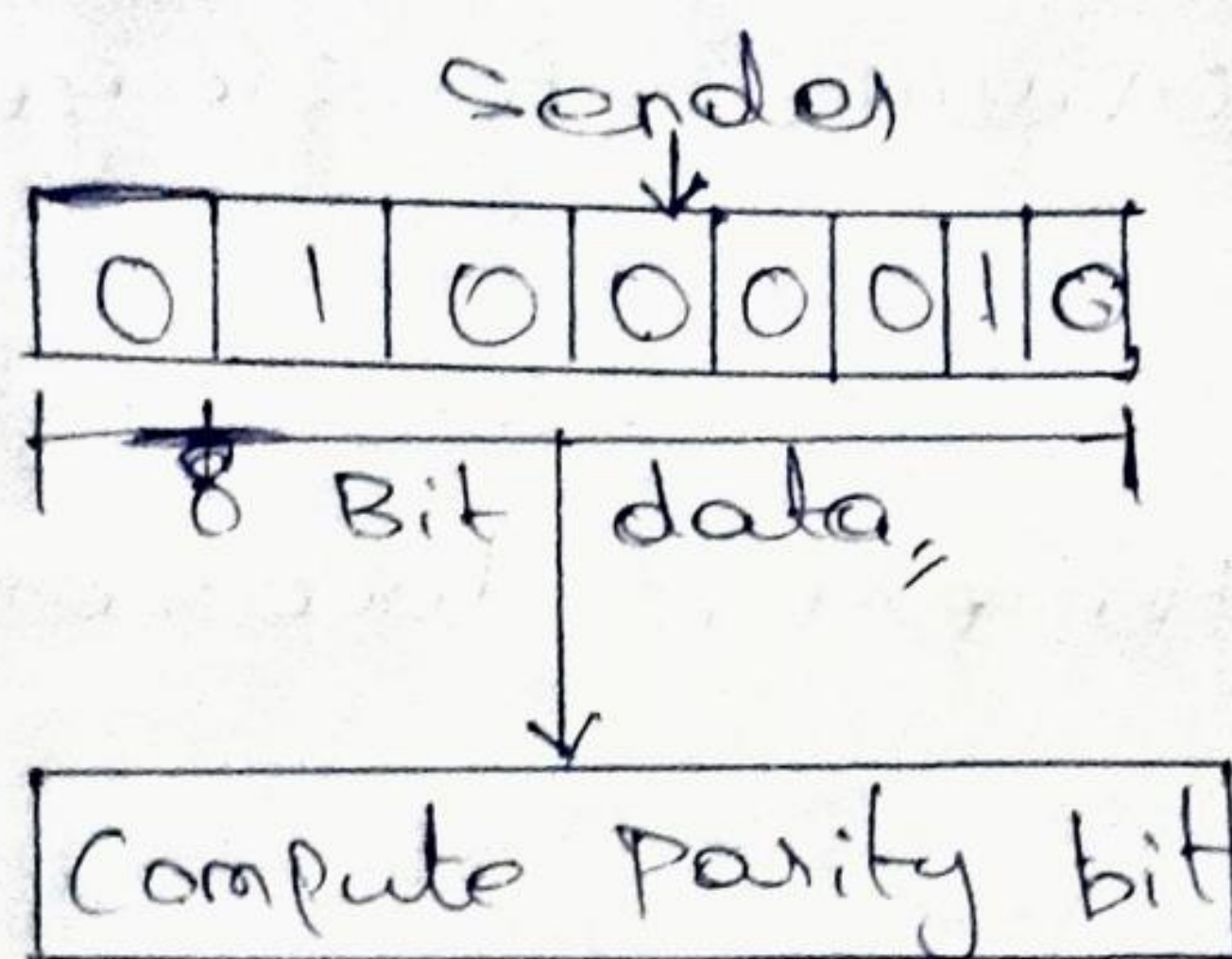
```
  ↓
┌─┬─────────────────┐
│0│0 1 0 0 1 0 0 1  │
└─┴─────────────────┘
        ↓
┌─┬─────────────────┐
│0│0 1 0 0 0 0 0 1  │
└─┴─────────────────┘
```

Parity |←———————————————→|
bit   8 data bits

② ASCII value for b is 66 convert this ASCII value to binary and store them as 8 bit data assume D_1 is wrongly entered as '0'
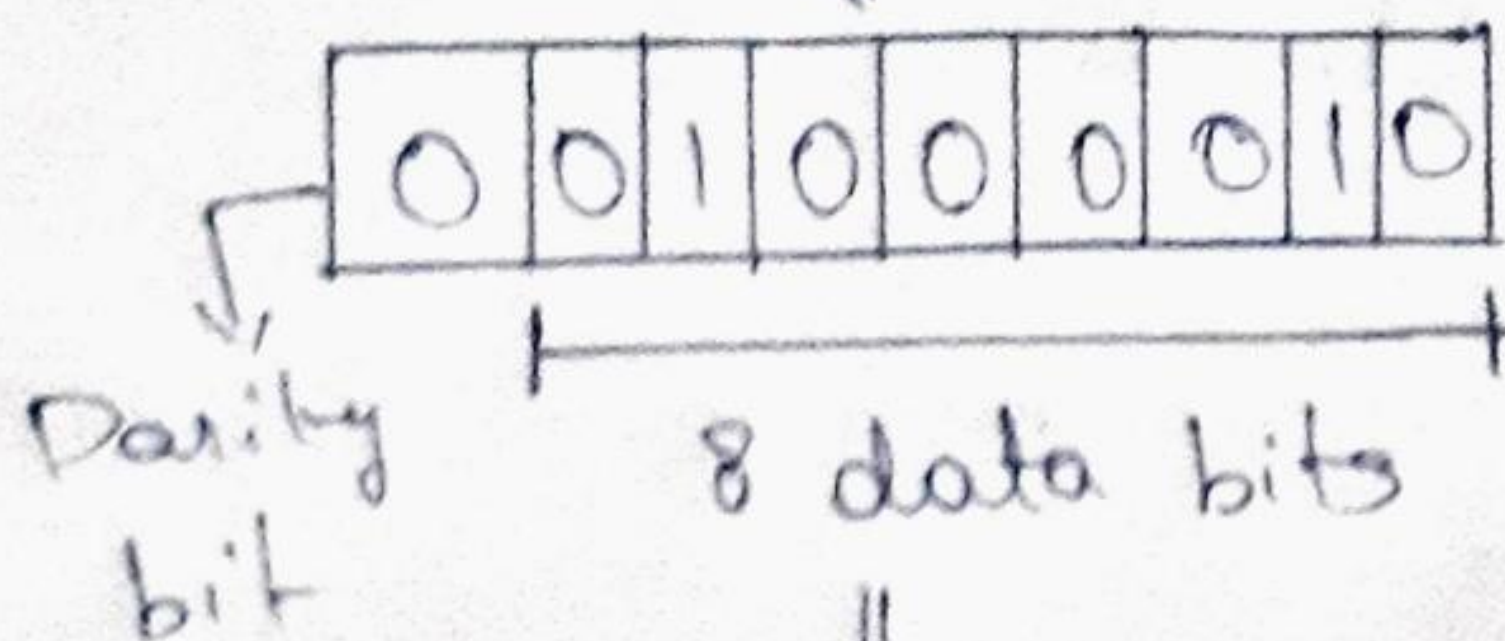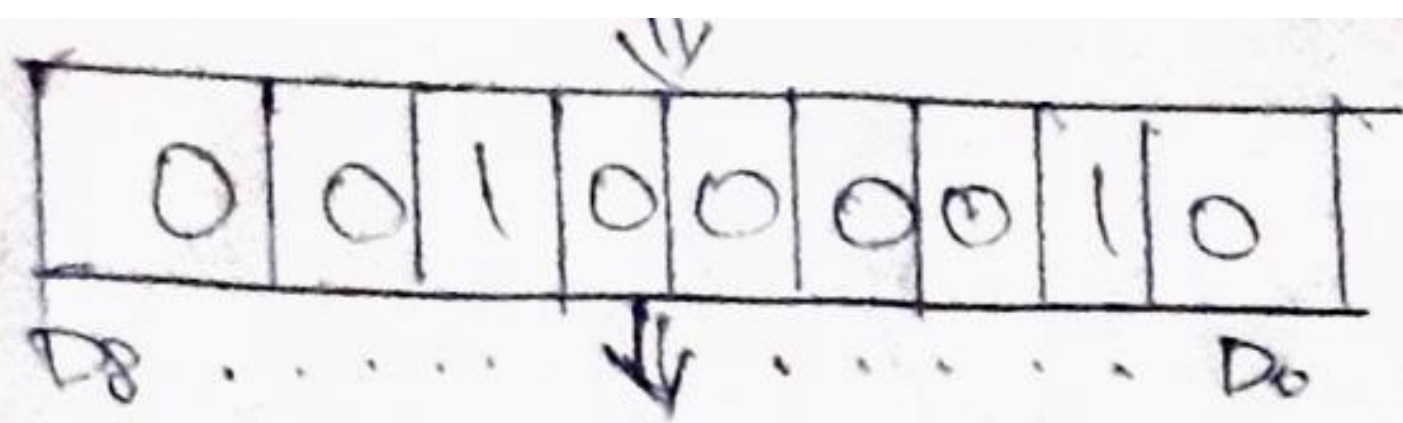
Answer:

The ASCII value 'b' = 66

The Binary value of "66" = 1000010

Sender

0	1	0	0	0	0	1	0

⟵ 8 Bit data, ⟶

↓

Compute Parity bit

↓

The total no. of '1's bit in the data unit is '2' So '0' is add to data unit as a Even Parity.

↓

0	0	1	0	0	0	0	1	0

Parity bit | ⟵ 8 data bits ⟶

↓

0	0	1	0	0	0	1	0

D_8 ..... D_0

```
┌─────────────┐
│ Transmitted │
└─────────────┘
```

In this 8 bit data D_1 is wronly entered as '0'

D_8 D_7 P_6 D_5 D_4 D_3 D_2 D_1 D_0

0	0	1	0	0	0	0	0	0

```
┌──────────────────────┐
│ Compute Parity bit    │
└──────────────────────┘
```
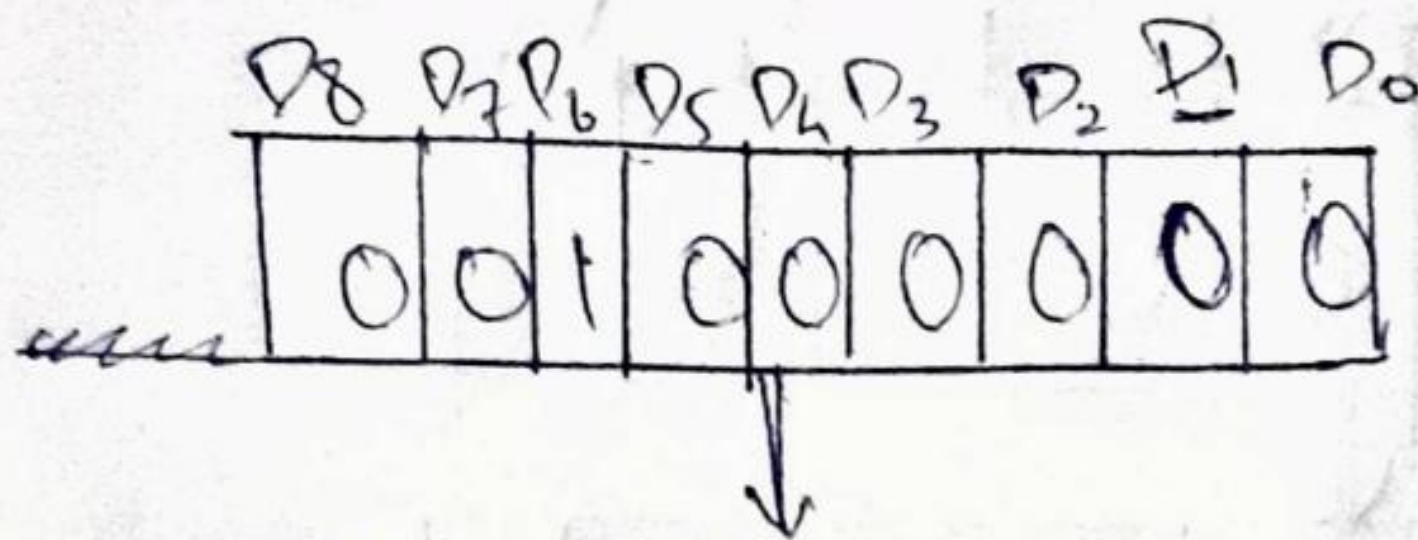
So the error is detected

Now the no. of '1' bit is 1 only it's odd but it has even parity '0'

```
         ╱‾‾‾‾‾╲
┌────────┤ even ├─────────┐
│ reject │╲    ╱│  Yes    │ Accept │
└────────┘ No╲ ╱          └────────┘
```

reject ← No   even   Yes → Accept
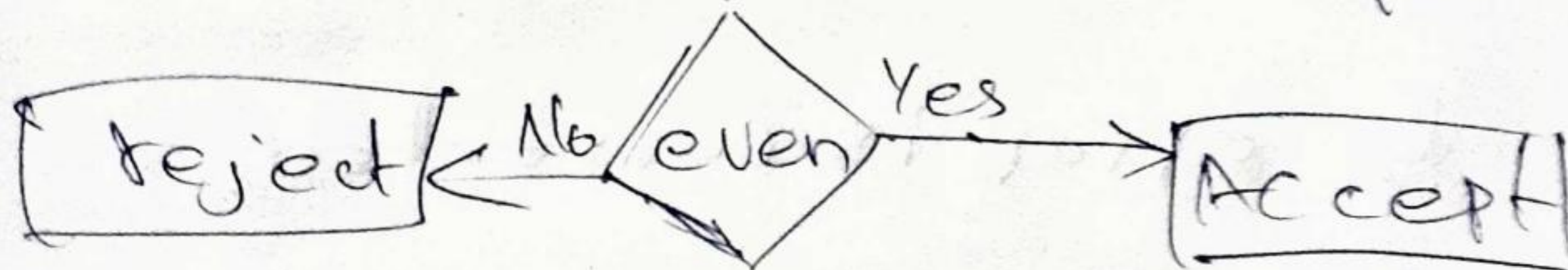
Error Correcting:

*) The Error Correcting codes also deploy the same strategy as error-detecting codes but additionally Such codes also dect the exact location of the corrupt bit.

→ Once the corrupt bit is located, it's value is reverted from 0 to 1 & 1 to 0. So here the "D_1 is wrongly entered as '0' now it is changed to '1".

B_8	D_7	D_6	D_5	D_4	B_3	D_2	D_1	D_0
0 0	1	0	0	0	0	0	0	

0 0	1	0	0	0	0	1	0

Parity bit   8 data bit
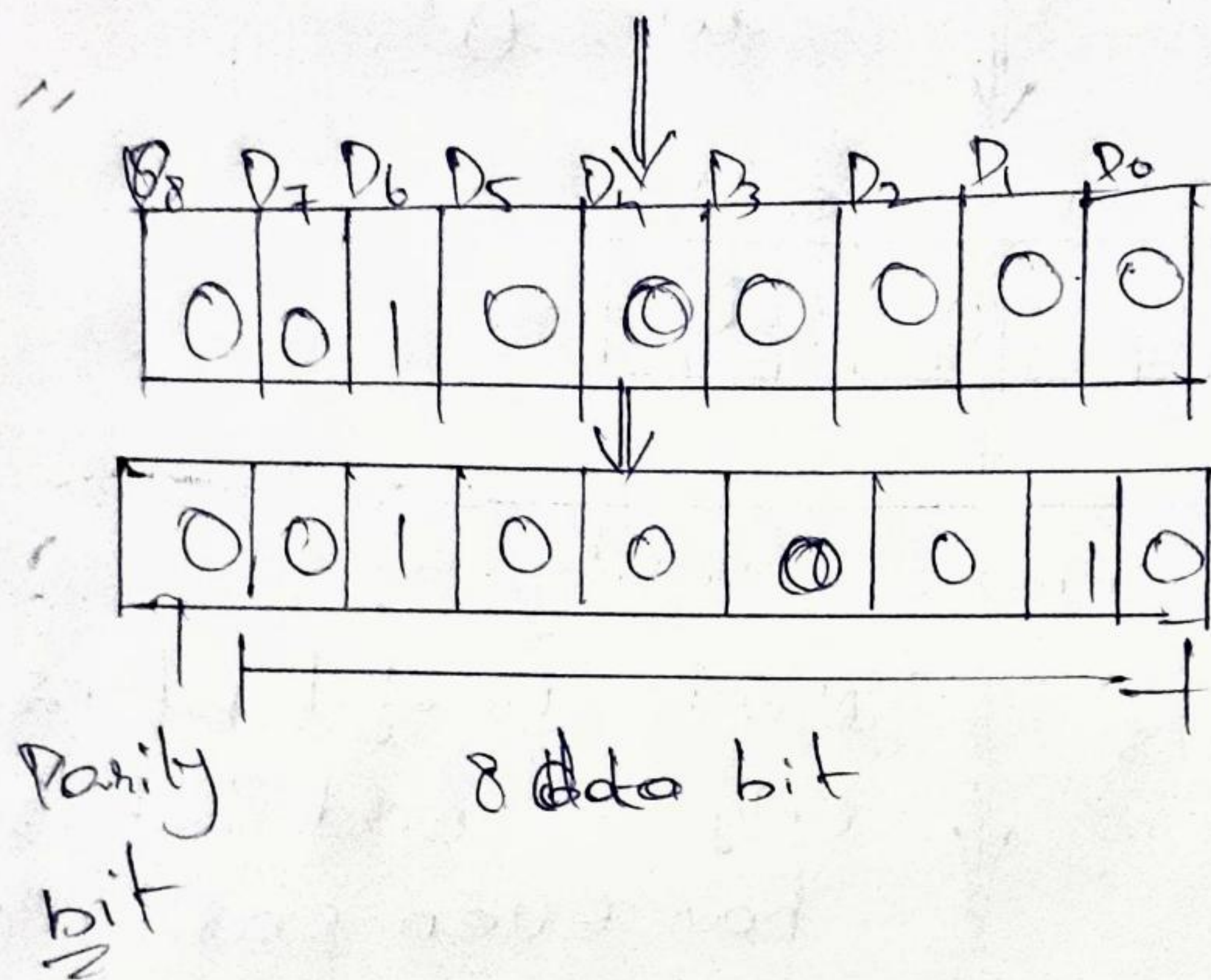
Hence the error is corrected.