

# *Error Correction*

It can be handled in two ways:

- 1) receiver can have the sender retransmit the entire data unit.
- 2) The receiver can use an error-correcting code, which automatically corrects certain errors.

# *Single-bit error correction*

To correct an error, the receiver reverses the value of the altered bit. To do so, it must know which bit is in error.

Number of redundancy bits needed

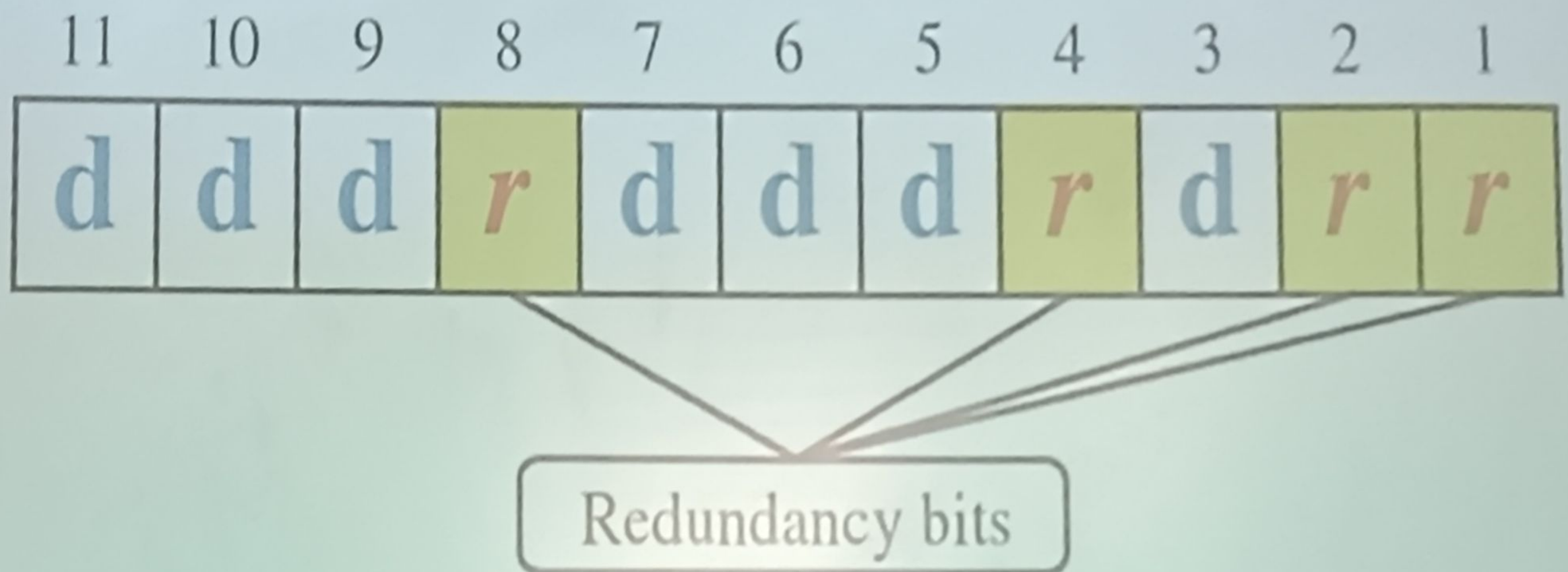
- Let data bits =  $m$
- Redundancy bits =  $r$

$\therefore$  Total message sent =  $m+r$

The value of  $r$  must satisfy the following relation:

$$2^r \geq m+r+1$$

# Hamming Code

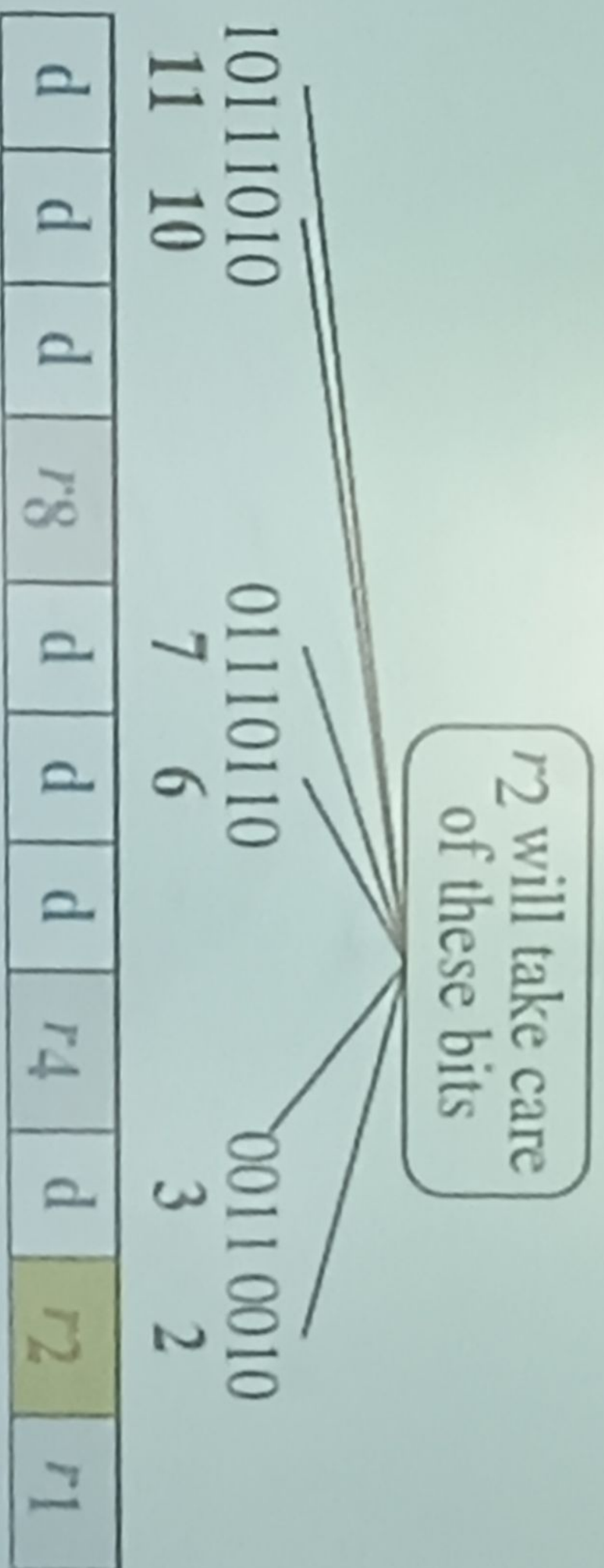
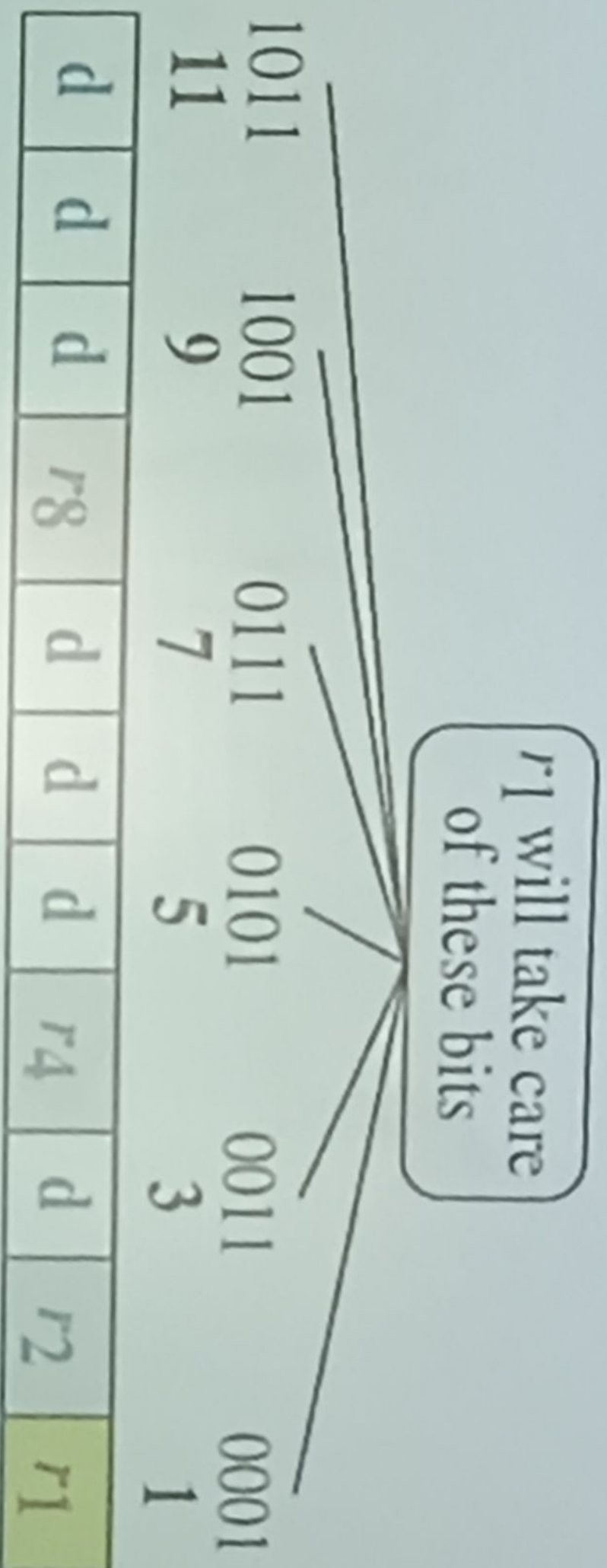


Redundancy bits will be at position  $2^n$  where  $n=0,1,2,3,\dots$

So the Redundancy bits 'r' will be at position 1,2,4,8,16,.....



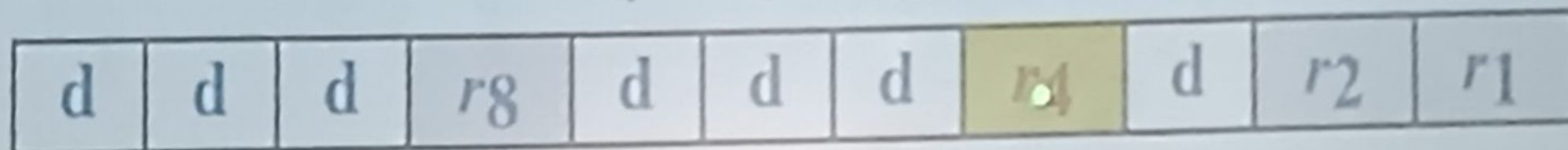
# Hamming Code



# Hamming Code

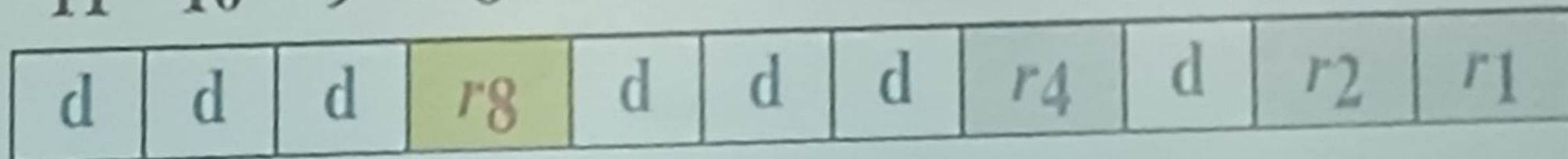
$r_4$  will take care of these bits

0111011001010100  
7 6 5 4



$r_8$  will take care of these bits

1011101010011000  
11 10 9 8





# Example of Hamming Code

Data: 1 0 0 1 1 0 1

11	10	9	8	7	6	5	4	3	2	1
----	----	---	---	---	---	---	---	---	---	---

Data

1	0	0	r4	1	1	0	r3	1	r2	r1
---	---	---	----	---	---	---	----	---	----	----

Adding r1

1	0	0	r4	1	1	0	r3	1	r2	1
---	---	---	----	---	---	---	----	---	----	---

Adding r2

1	0	0	r4	1	1	0	r3	1	0	1
---	---	---	----	---	---	---	----	---	---	---

Adding r4

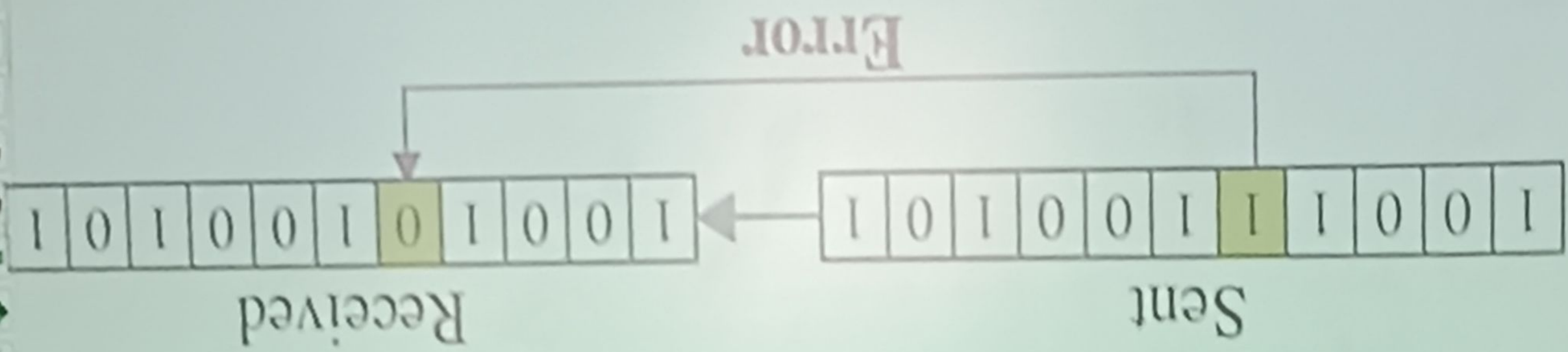
1	0	0	r4	1	1	0	0	1	0	1
---	---	---	----	---	---	---	---	---	---	---

Adding r8

1	0	0	1	1	1	0	0	1	0	1
---	---	---	---	---	---	---	---	---	---	---

Code: 1 0 0 1 1 1 0 0 1 0 1

# Single-bit error





# Error Detection

