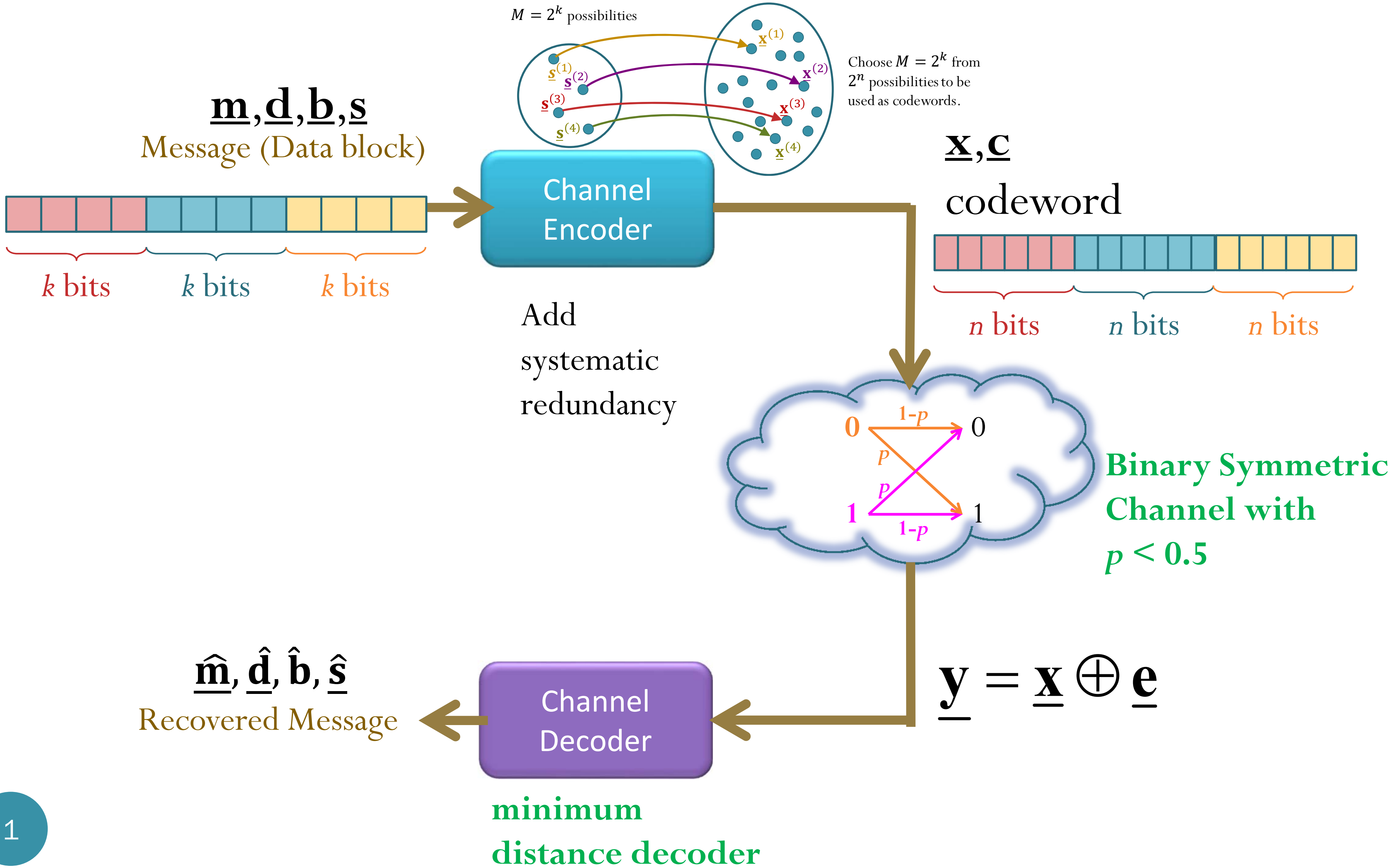
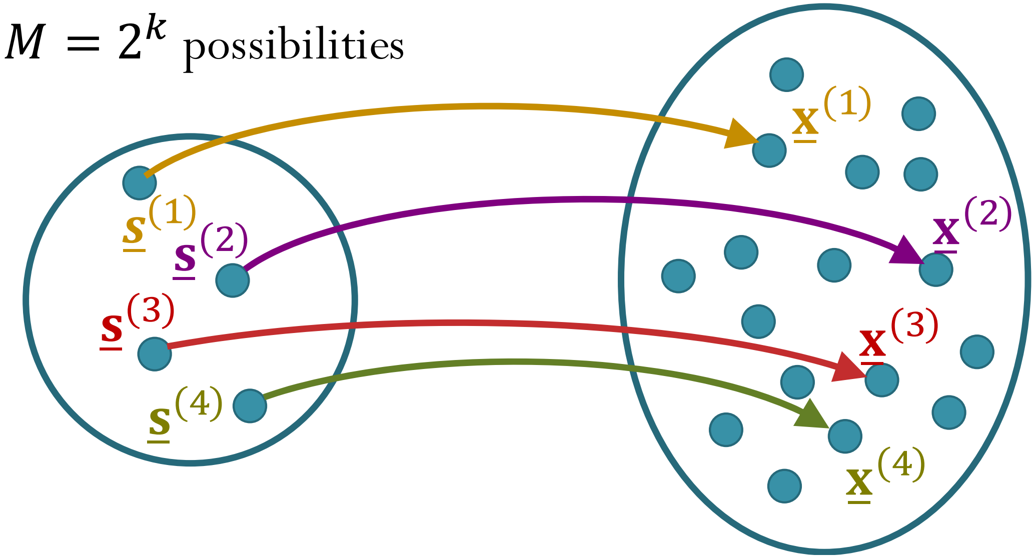
Review: Channel Encoder and Decoder



Review: Linear Block Codes

- k = number of bits in each data block.
  - There are 2^k possibilities for the k-bit data block.
- n = number of bits in each codeword.
  - There are 2^k valid codewords.
    - One for each possible data block.

$M = 2^k$ possibilities



Choose $M = 2^k$ from 2^n possibilities to be used as codewords.

Review: Linear Block Codes

- Given a list of codewords for a code $\mathcal{C}$, we can determine whether $\mathcal{C}$ is linear by
  - Definition: if $\underline{\mathbf{x}}^{(1)}$ and $\underline{\mathbf{x}}^{(2)} \in \mathcal{C}$, then $\underline{\mathbf{x}}^{(1)} \oplus \underline{\mathbf{x}}^{(2)} \in \mathcal{C}$
  - Shortcut:
    - First check that $\mathcal{C}$ must contain $\underline{\mathbf{0}}$.
    - Check the definition but only check the non-zero codewords.
- Codewords can be generated by a **generator matrix**
  - $\underline{\mathbf{x}} = \underline{\mathbf{b}}\mathbf{G} = \sum_{i=1}^{k} b_i \underline{\mathbf{g}}^{(i)}$ where $\underline{\mathbf{g}}^{(i)}$ is the i^{th} row of $\mathbf{G}$

Review: Single-parity-check code

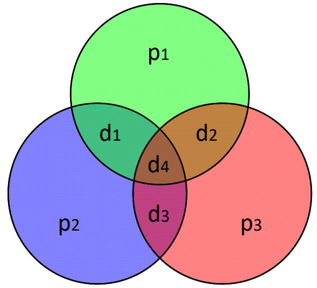
- $\underline{\mathbf{x}} = \left[\boxed{\quad \underline{\mathbf{b}} \quad} ; \underbrace{\sum_{j=1}^{k} b_j}_{\text{parity bit}} \right]$
- An example of linear block code.
- **Use even parity**
- $\mathbf{G} = [\mathbf{I}_{k \times k}; \underline{\mathbf{1}}^T]$
- Can ***detect*** any odd number of bit error.

Review: Linear Block Codes

Code structure

$$\underline{\mathbf{x}} = [p_1 \quad d_1 \quad p_2 \quad d_2 \quad p_3 \quad d_3 \quad d_4]$$

where
$$p_1 = d_1 \oplus d_2 \oplus d_4$$
$$p_2 = d_1 \oplus d_3 \oplus d_4$$
$$p_3 = d_2 \oplus d_3 \oplus d_4$$



Codebook

$\underline{\mathbf{d}}$				$\underline{\mathbf{x}}$						
0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	0	1	0	1	0	1
0	0	1	0	0	0	1	0	1	1	0
0	0	1	1	1	0	0	0	0	1	1
0	1	0	0	1	0	0	1	1	0	0
0	1	0	1	0	0	1	1	0	0	1
0	1	1	0	1	0	1	1	0	1	0
0	1	1	1	0	0	0	1	1	1	1
1	0	0	0	1	1	1	0	0	0	0
1	0	0	1	0	1	0	0	1	0	1
1	0	1	0	1	1	0	0	1	1	0
1	0	1	1	0	1	1	0	0	1	1
1	1	0	0	0	1	1	1	1	0	0
1	1	0	1	1	1	0	1	0	0	1
1	1	1	0	0	1	0	1	0	1	0
1	1	1	1	1	1	1	1	1	1	1

$$\begin{array}{c} \\ \\ \mathbf{G} = \end{array} \begin{array}{c} \begin{array}{ccccccc} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 \\ p_1 & d_1 & p_2 & d_2 & p_3 & d_3 & d_4 \end{array} \\ \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} \end{array}$$

Generator Matrix

$$\begin{array}{c} \\ \\ \mathbf{H} = \end{array} \begin{array}{c} \begin{array}{ccccccc} y_1 & y_2 & y_3 & y_4 & y_5 & y_6 & y_7 \\ x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 \\ p_1 & d_1 & p_2 & d_2 & p_3 & d_3 & d_4 \end{array} \\ \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} \end{array}$$

Parity Check Matrix

Review: Even Parity

- A binary vector (or a collection of 1s and 0s) has **even parity** if and only if the number of 1s in there is even.
  - Suppose we are given the values of all the bits except one bit.
    - We can force the vector to have even parity by setting the value of the remaining bit to be the sum of the other bits.

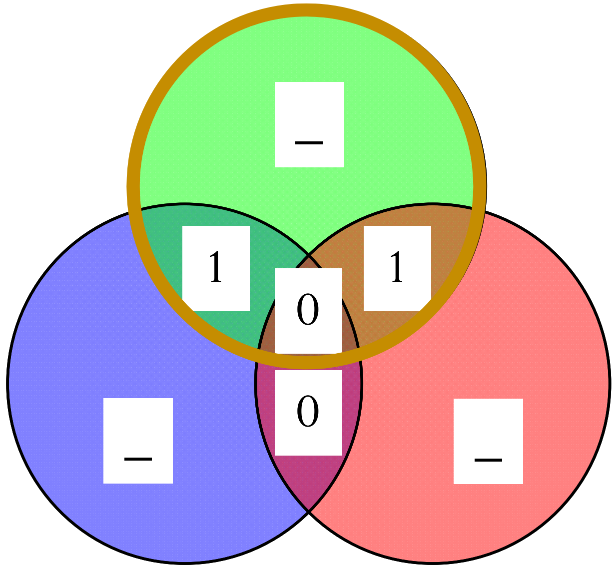
Single-parity-check code

[1 0 1 1 0 _]

Square array

1	0	1	_
0	1	1	_
0	0	1	_
_	_	_	

Hamming code



Review: Linear Block Codes

- The code structure is **built** into each codeword at the encoder (transmitter) via the generator matrix
  - Each codeword is created by $\underline{\mathbf{x}} = \underline{\mathbf{d}}\mathbf{G}$.
- The code structure is **checked** at the decoder (receiver) via the parity check matrix.
  - A valid codeword must satisfy $\underline{\mathbf{x}}\mathbf{H}^T = \underline{\mathbf{0}}$.

$$\begin{array}{cc} & \begin{array}{ccccccc} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 \\ p_1 & d_1 & p_2 & d_2 & p_3 & d_3 & d_4 \end{array} \\ \mathbf{G} = & \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} \end{array} \Longleftrightarrow \begin{array}{cc} & \begin{array}{ccccccc} y_1 & y_2 & y_3 & y_4 & y_5 & y_6 & y_7 \\ x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 \\ p_1 & d_1 & p_2 & d_2 & p_3 & d_3 & d_4 \end{array} \\ \mathbf{H} = & \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} \end{array}$$

Review: Linear Block Codes

$$\mathbf{G} = \begin{array}{c} \begin{array}{ccccccc} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 \\ p_1 & d_1 & p_2 & d_2 & p_3 & d_3 & d_4 \end{array} \\ \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} \end{array} \quad \Longleftrightarrow \quad \mathbf{H} = \begin{array}{c} \begin{array}{ccccccc} y_1 & y_2 & y_3 & y_4 & y_5 & y_6 & y_7 \\ x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 \\ p_1 & d_1 & p_2 & d_2 & p_3 & d_3 & d_4 \end{array} \\ \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} \end{array}$$

- The "identity-matrix" columns in **G** corresponds to positions of the message (data) bits in each codeword.
  - Ex. For this code, codeword $\underline{\mathbf{x}} = [1\ 1\ 0\ 0\ 1\ 1\ 0]$ corresponds to message $\underline{\mathbf{b}} = [1\ 0\ 1\ 0]$.
- The "identity-matrix" columns in **H** corresponds to positions of the parity (check) bits in each codeword.

Review: Hamming Code Recipe

Here,
$m = 3$
$n = 2^3 - 1$
$= 7$
$k = 4$

$$\begin{array}{cc} & \begin{array}{cc} \mathbf{I}_m & \mathbf{P}^T \end{array} \\ \mathbf{H} = & \left[\begin{array}{ccc|cccc} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{array}\right] \end{array}$$

$$\begin{array}{cc} \mathbf{G} = & \left[\begin{array}{ccc|cccc} 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{array}\right] \\ & \begin{array}{cc} \mathbf{P} & \mathbf{I}_k \end{array} \end{array}$$

- Start with the parity-check matrix
- m rows
  - $m = n - k$
- Columns are all possible $\underbrace{\text{nonzero}}$ m-bit vectors
  - $n = 2^m \underbrace{- 1}$ columns
  - Arranged to have $\mathbf{I}_m$ on the left (or on the right).
    - This simplifies conversion to **G.**
- Get **G** from **H**.

$$\mathbf{G} = \left[\mathbf{P}_{k\times(n-k)} \;\vdots\; \mathbf{I}_k\right] \Leftrightarrow \mathbf{H} = \left[\mathbf{I}_{n-k} \;\vdots\; -\mathbf{P}^T\right]$$

- Note that the size of the identity matrices in **G** and **H** can be different.

Review: Linear Block Code