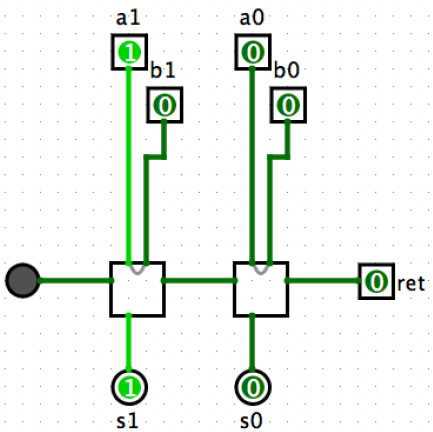
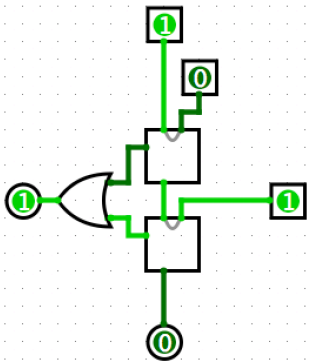
Principes de fonctionnement des machines binaires

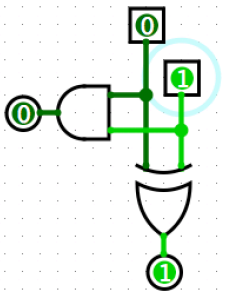
2022–2023

Matthieu Picantin

- numération et arithmétique
- numération et arithmétique en machine
- codes, codages, compression,
- contrôle d'erreur (détection, correction)
- crypto (confidentialité, authenticité, intégrité)
- logique et calcul propositionnel
- circuits numériques







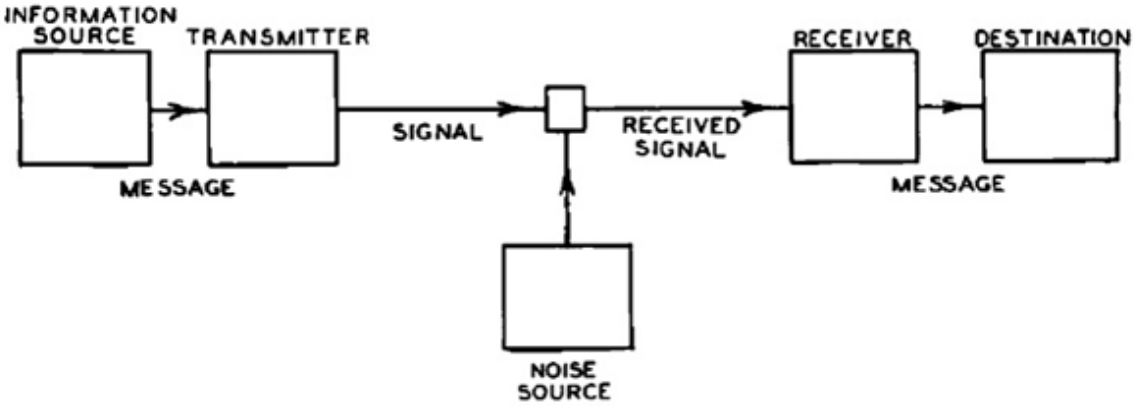


Fig. 1—Schematic diagram of a general communication system.

Détection (+retransmission)

- juste assez de redondance pour que le récepteur puisse détecter d'éventuelles erreurs & demander une retransmission
- adapté sur canal fiable

Correction

- suffisamment de redondance pour que le récepteur puisse corriger d'éventuelles erreurs
- adapté sur canal bruité

Redondance et mots de code

- mot de code (n bits) = données (m bits) + contrôle (r bits)
- 2^m mots de code (légaux) parmi 2^n mots possibles

A Mathematical Theory of Communication by C. E. Shannon (1948)

Distance de Hamming

- nombre de bits différents entre 2 mots
- somme des **1** du XOR des 2 mots
- nombre minimum d'erreurs simples pour passer d'un mot à l'autre

Distance de Hamming d'un code

distance minimale entre deux mots du code:

$$d_H(\mathcal{C}) = \min\{d_H(u,v) : u \neq v \in \mathcal{C}\}$$

Qualité d'un code

- un code $\mathcal{C}$ vérifiant $d_H(\mathcal{C}) \geq k+1$ permet de détecter k erreurs
- un code $\mathcal{C}$ vérifiant $d_H(\mathcal{C}) \geq 2k+1$ permet de corriger k erreurs

Un code 1-détecteur

L'ajout d'un bit de parité produit un code de distance de Hamming 2

Un code 2-correcteur

Le code $\{$**0000000000**, **0000011111**, **1111100000**, **1111111111**$\}$ est de distance de Hamming 5

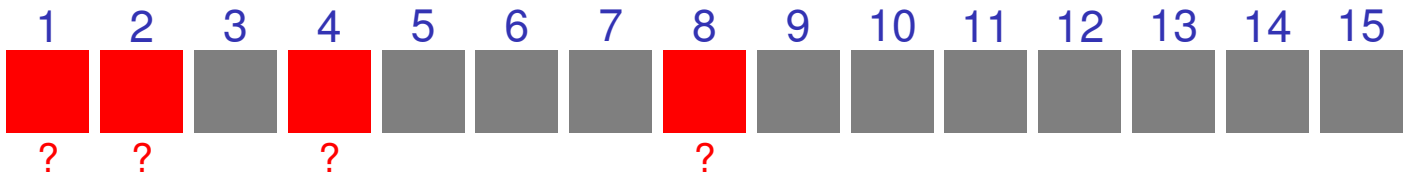
Error detecting and error correcting codes by R. W. Hamming (1950)

En théorie: de l'espace pour les boules (condition nécessaire)

Pouvoir corriger toute erreur simple pour m bits de données demande r bits de contrôle avec $m + r < 2^r$

En pratique: le code de Hamming (condition suffisante)

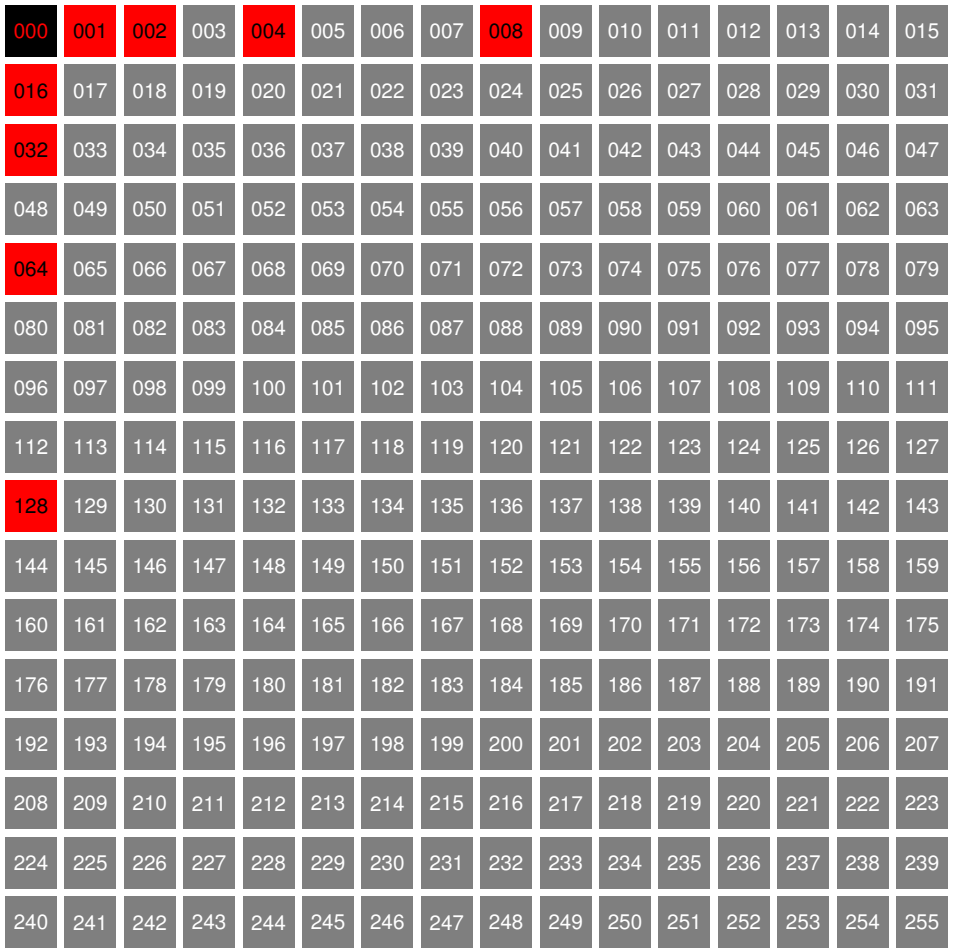
- bits numérotés de 1 à $n = m + r$ de gauche à droite
- r bits de contrôle aux positions puissances de 2 $(1, 2, 4, 8, 16, ...)$
- m bits de données aux autres positions $(3, 5-7, 9-15, 17, 18...)$
- bits de contrôle = calcul de parité sur les bits de données aux positions dont la décomposition en somme de puissances de 2 fait intervenir la position du bit de contrôle concerné
- détection du bit erroné (et correction) par somme des positions des bits de contrôle non-conformes à la parité



Error detecting and error correcting codes by R. W. Hamming (1950)

Mieux encore: le code de Hamming étendu

- des bits numérotés de 1 à $n = m + r$ comme pour le code original
- un bit de parité supplémentaire (parité globale) numéroté 0



Error detecting and error correcting codes by R. W. Hamming (1950)

Mieux encore: le code de Hamming étendu

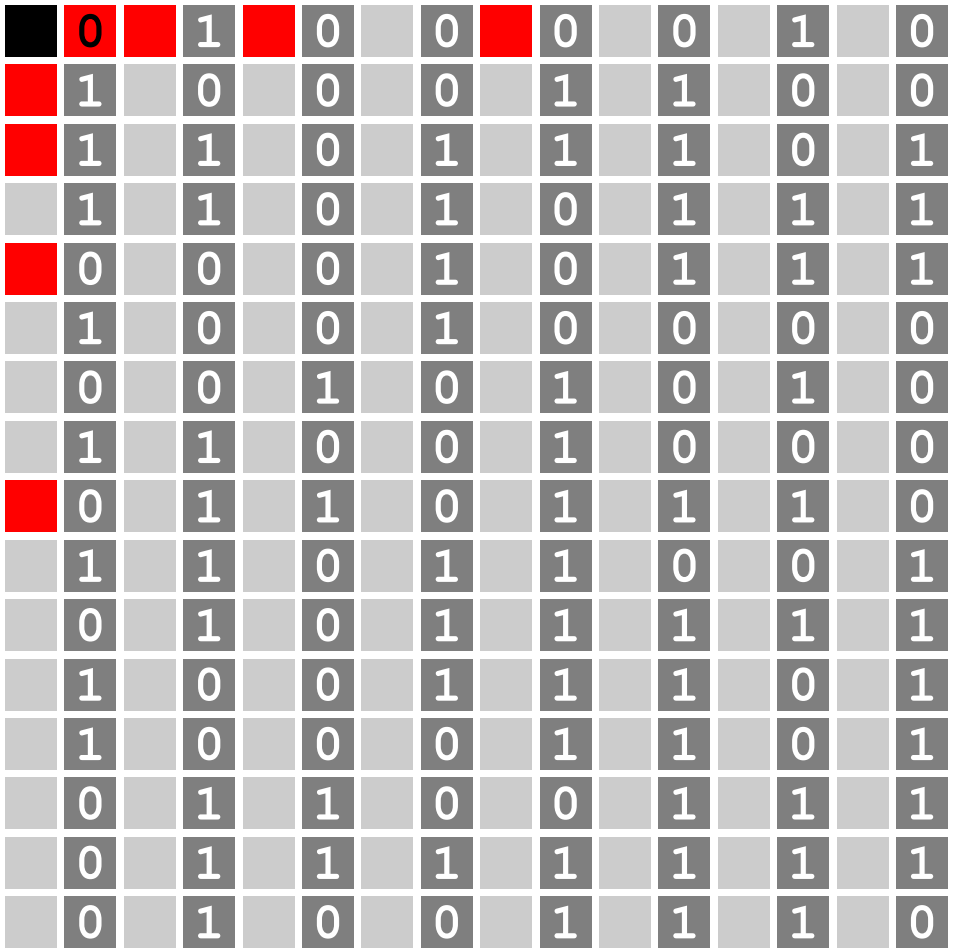
- des bits numérotés de 1 à $n = m + r$ comme pour le code original
- un bit de parité supplémentaire (parité globale) numéroté 0

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

Error detecting and error correcting codes by R. W. Hamming (1950)

Mieux encore: le code de Hamming étendu

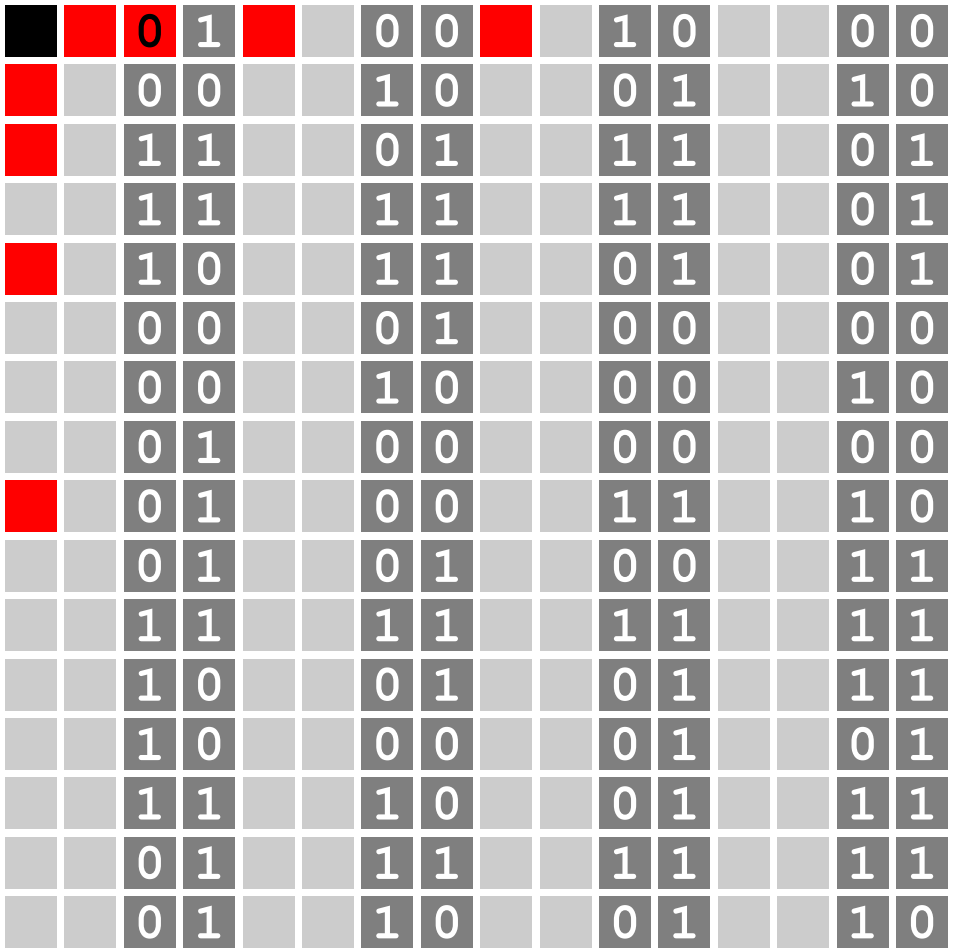
- des bits numérotés de 1 à $n = m + r$ comme pour le code original
- un bit de parité supplémentaire (parité globale) numéroté 0



Error detecting and error correcting codes by R. W. Hamming (1950)

Mieux encore: le code de Hamming étendu

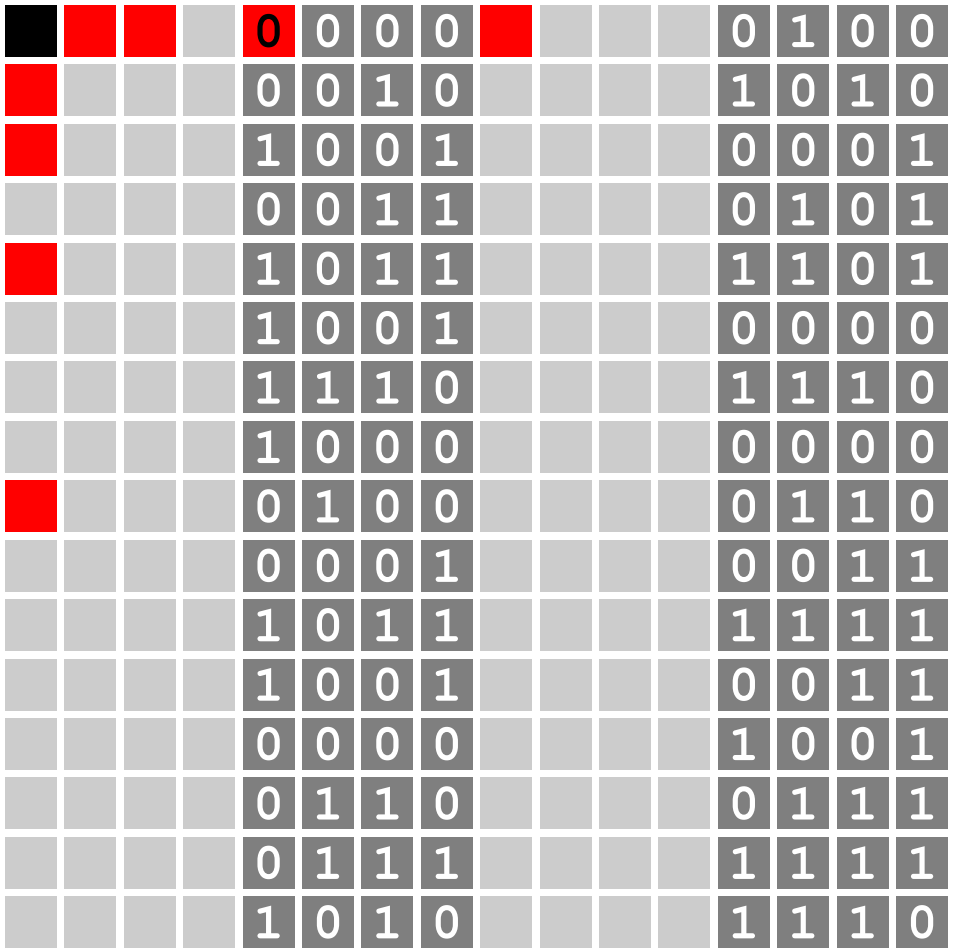
- des bits numérotés de 1 à $n = m + r$ comme pour le code original
- un bit de parité supplémentaire (parité globale) numéroté 0



Error detecting and error correcting codes by R. W. Hamming (1950)

Mieux encore: le code de Hamming étendu

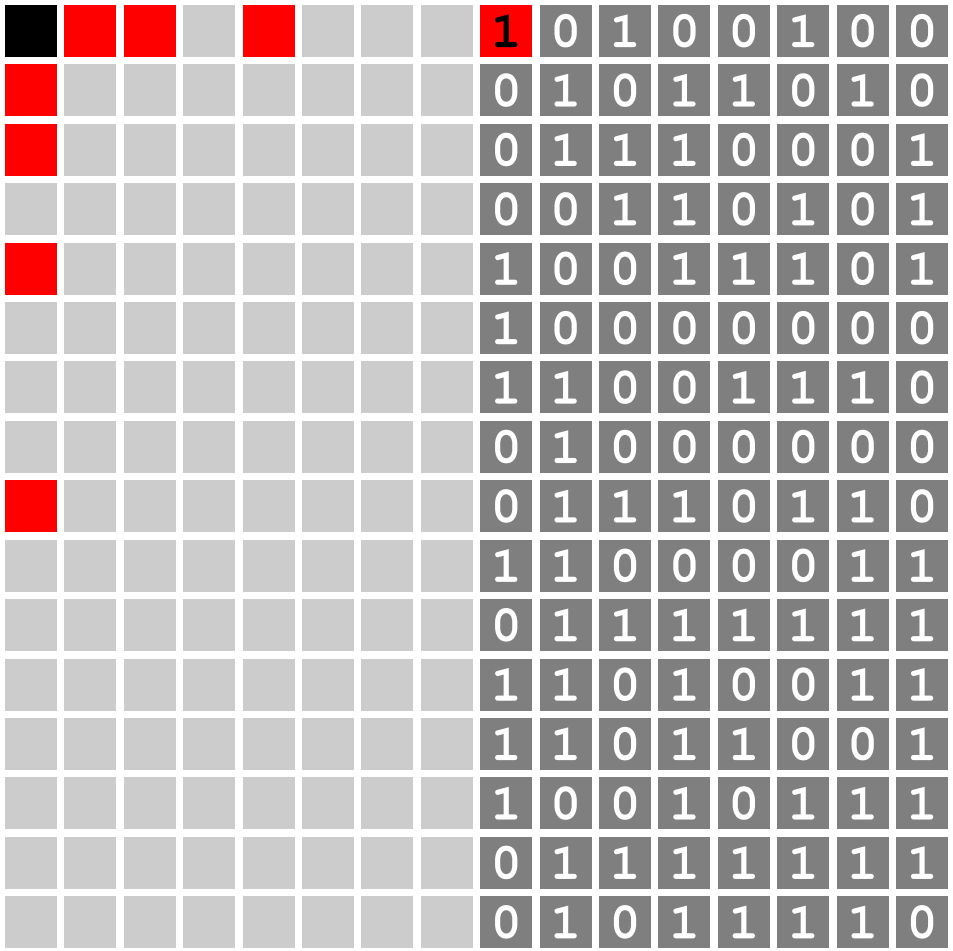
- des bits numérotés de 1 à $n = m + r$ comme pour le code original
- un bit de parité supplémentaire (parité globale) numéroté 0



Error detecting and error correcting codes by R. W. Hamming (1950)

Mieux encore: le code de Hamming étendu

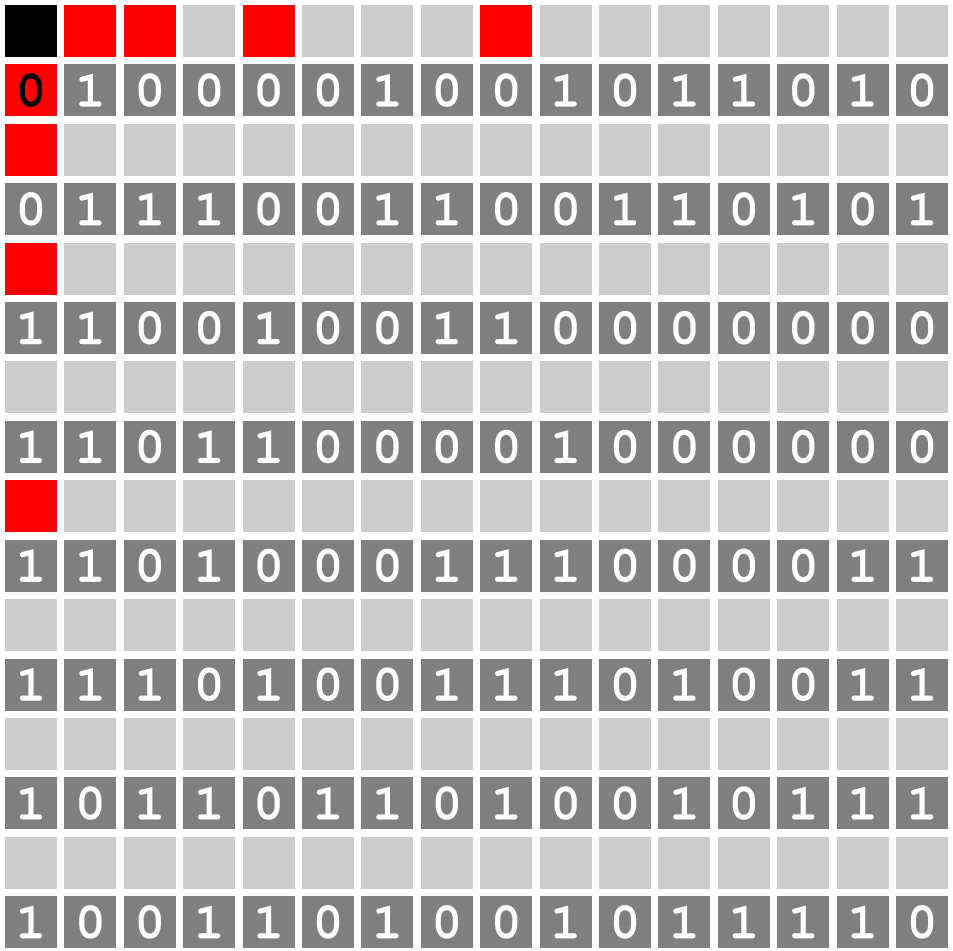
- des bits numérotés de 1 à $n = m + r$ comme pour le code original
- un bit de parité supplémentaire (parité globale) numéroté 0



Error detecting and error correcting codes by R. W. Hamming (1950)

Mieux encore: le code de Hamming étendu

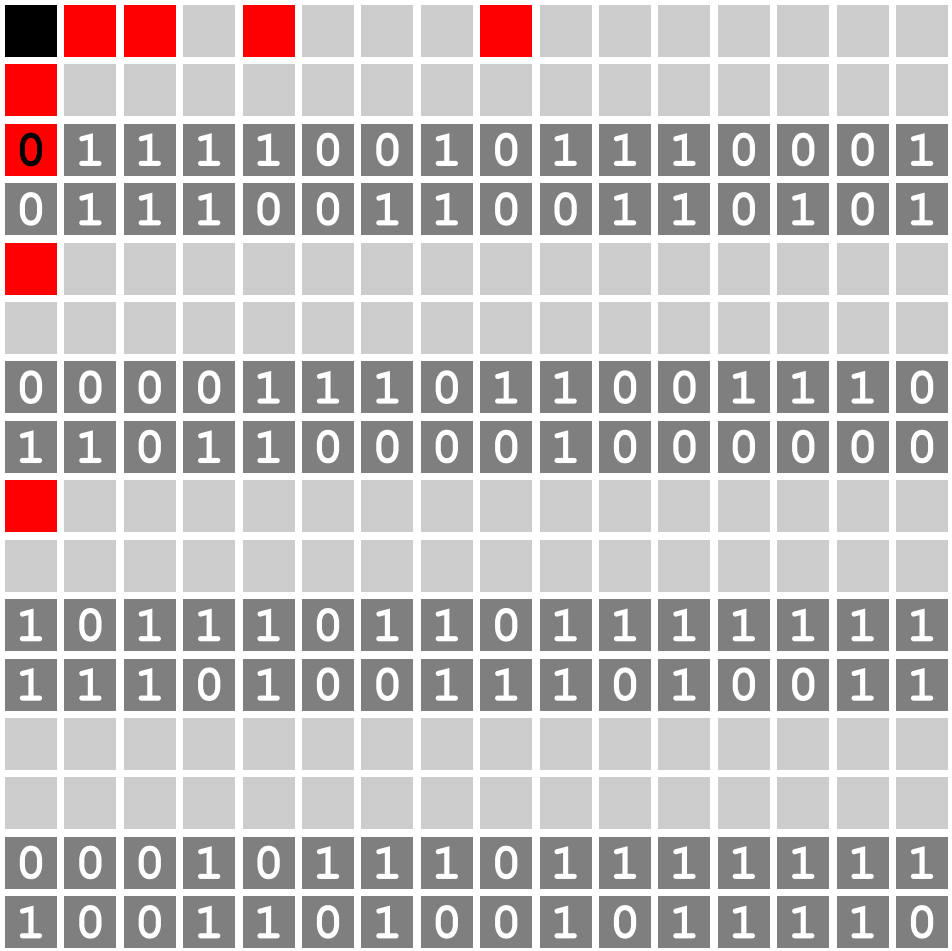
- des bits numérotés de 1 à $n = m + r$ comme pour le code original
- un bit de parité supplémentaire (parité globale) numéroté 0



Error detecting and error correcting codes by R. W. Hamming (1950)

Mieux encore: le code de Hamming étendu

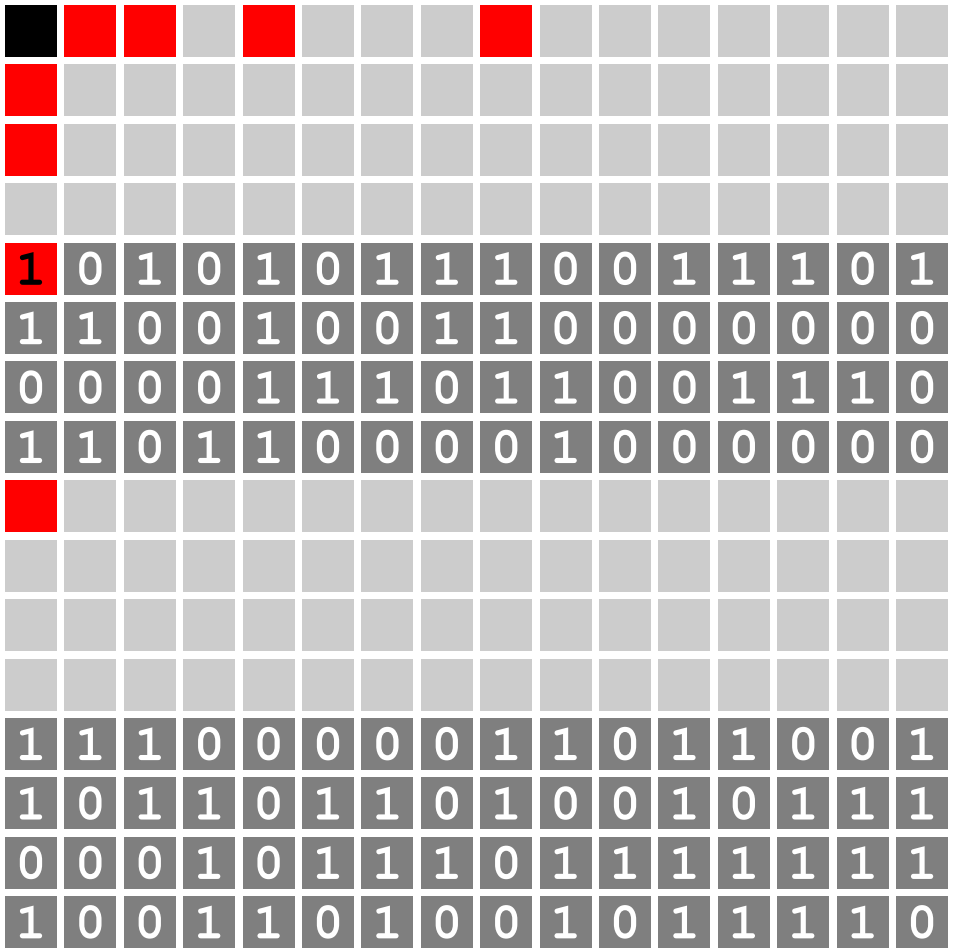
- des bits numérotés de 1 à $n = m + r$ comme pour le code original
- un bit de parité supplémentaire (parité globale) numéroté 0



Error detecting and error correcting codes by R. W. Hamming (1950)

Mieux encore: le code de Hamming étendu

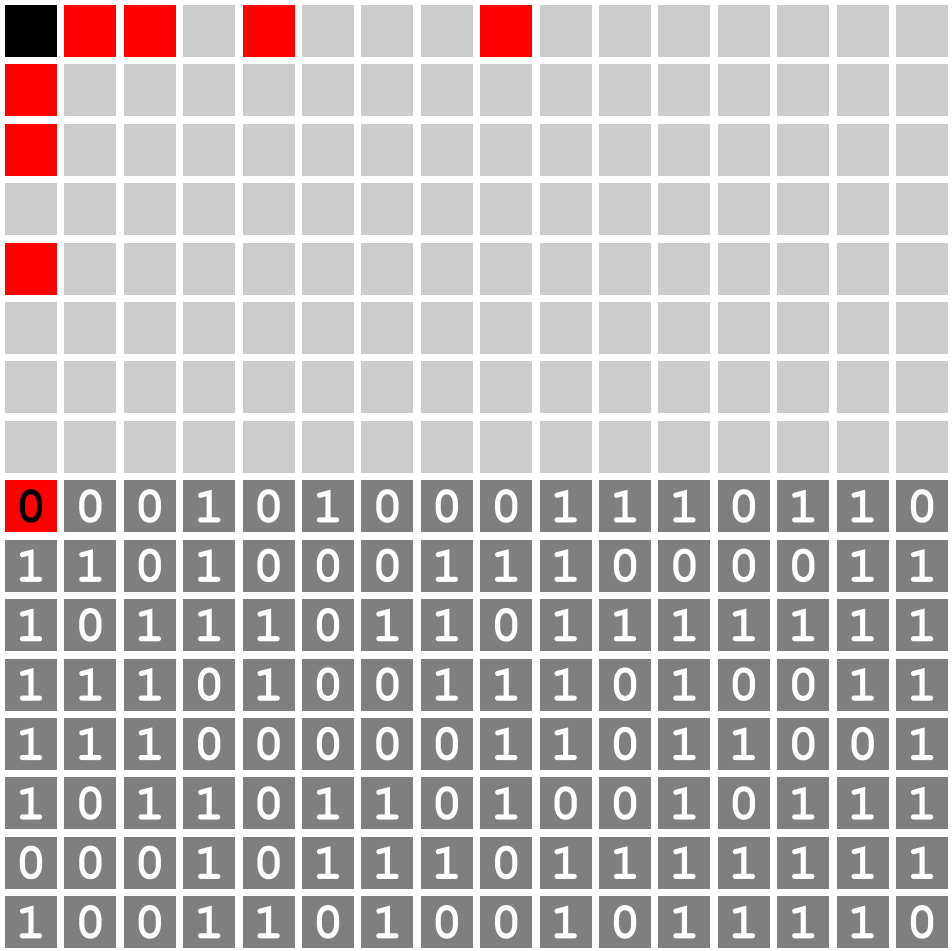
- des bits numérotés de 1 à $n = m + r$ comme pour le code original
- un bit de parité supplémentaire (parité globale) numéroté 0



Error detecting and error correcting codes by R. W. Hamming (1950)

Mieux encore: le code de Hamming étendu

- des bits numérotés de 1 à $n = m + r$ comme pour le code original
- un bit de parité supplémentaire (parité globale) numéroté 0



Error detecting and error correcting codes by R. W. Hamming (1950)

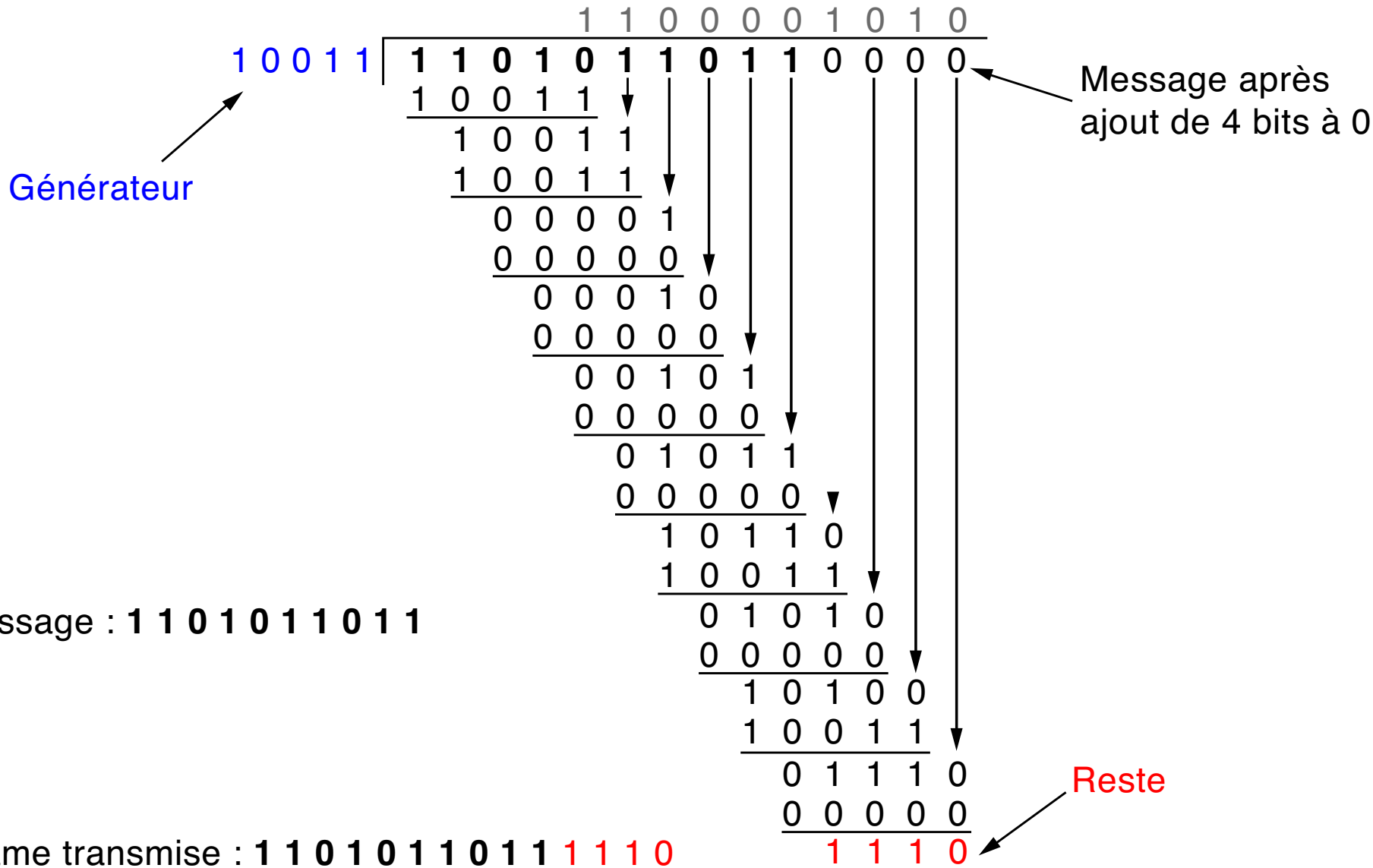
Codes CRC (Cyclic Redundancy Check)

- correspondance entre rang des bits et degré des monômes
  - le mot **10011** code le polynôme $x^4 + x + 1$
- arithmétique polynomiale (soustraction modulo 2 et division euclidienne)

Utilisation d'un polynôme générateur $G(x)$

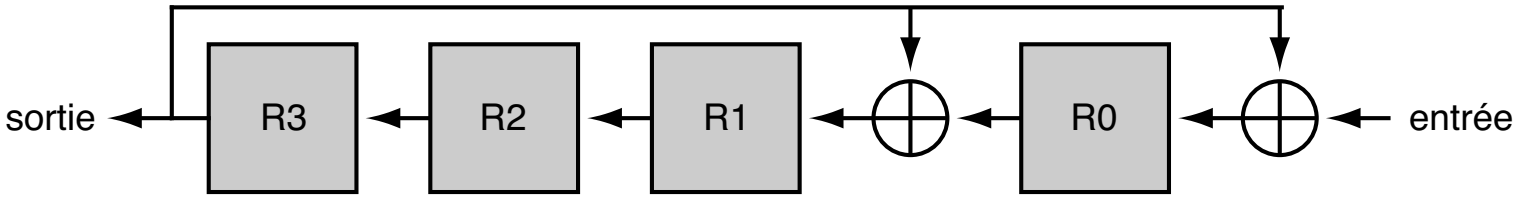
- $G(x)$ de degré r et message $M(x)$
- ajout de r bits à **0** après le bit de poids faible de $M(x)$
- division de $x^r M(x)$ par $G(x)$: reste $R(x)$
- envoi de $T(x) = x^r M(x) - R(x)$
- $T(x)$ est divisible par $G(x)$ (à vérifier par le récepteur !)

CRC16 : $x^{16} + x^{15} + x^2 + 1$
CRC32 : $x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$



Message : **1 1 0 1 0 1 1 0 1 1**

Trame transmise : **1 1 0 1 0 1 1 0 1 1** 1 1 1 0



	0	0	0	0	1101101011**0000**
0	0	0	0	1	101101011**0000**
00	0	0	1	1	01101011**0000**
000	0	1	1	0	1101011**0000**
0000	1	1	0	1	101011**0000**
00001	1	0	0	0	01011**0000**
000011	0	0	1	1	1011**0000**
0000110	0	1	1	1	011**0000**
00001100	1	1	1	0	11**0000**
000011001	1	1	1	0	1**0000**
0000110011	1	1	1	0	**0000**
00001100111	1	1	1	**1**	**000**
000011001111	1	1	**0**	**1**	**00**
0000110011111	1	**0**	**0**	**1**	**0**
00001100111111	**0**	**0**	**0**	**1**	

$\oplus$	**0**	**1**
0	**0**	**1**
1	**1**	**0**