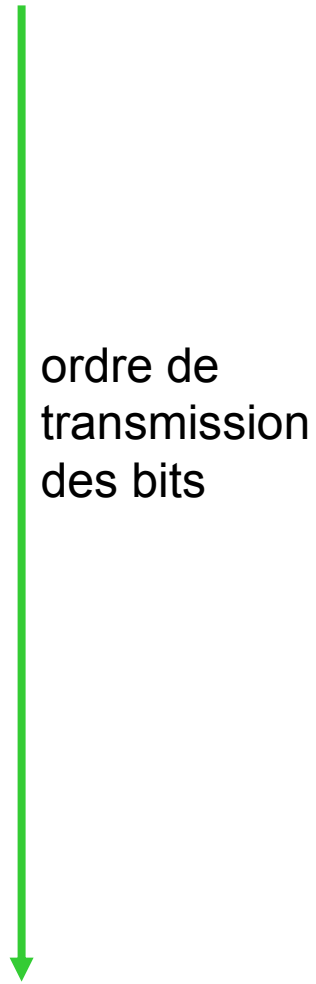
Utilisation d'un code de Hamming pour corriger des erreurs en rafale

Caractères	Code ASCII	En rouge: les bits de contrôle
H	1001000	00110010000
a	1100001	10111001001
m	1101101	11101010101
i	1101001	01101011001
n	1101110	01101010110
g	1100111	11111001111
	0100000	10011000000
c	1100011	11111000011
o	1101111	00101011111
d	1100100	11111001100
e	1100101	00111000101



Codes correcteurs d'erreurs distance de Hamming: un exemple

Codes avec un bit de parité

0000000 0
0000001 1
0000010 1
0000011 0
...

Distance de Hamming de 2: chaque code valide diffère d'au moins 2 bits

→ ne peut pas détecter les erreurs de 2 bits: le code résultant corrompu correspond à un code valide

Codes correcteurs d'erreurs: définitions

Erreur la plus simple à détecter: erreur sur un bit isolé.

Distance de Hamming permet d'identifier le bit erronné.

Code: *k* bits de données, *n* bits au total $\rightarrow$ code *(n,k)*

Taux de codage = efficacité du code = ratio *k / n*

Redondance = *1 – k / n*

Codes de Hamming: surtout utilisées dans les applications qui ont des erreurs isolées sur un seul bit, e.g., les systèmes de mémoire des semi-conducteurs

Codes les plus utilisées dans les transmissions de données: **les codes convolutionnels**

Codes de Hamming: un exemple

11	10	9	8	7	6	5	4	3	2	1
1	0	0	x	1	1	0	x	1	x	x

- Considérer les bits de données égaux à 1, et considérer la représentation en base deux de leurs indices

11	=	1	0	1	1
7	=	0	1	1	1
6	=	0	1	1	0
3	=	0	0	1	1
	=	1	0	0	1

$11 = 2^3 + 2^1 + 2^0$

$7 = 2^2 + 2^1 + 2^0$

$6 = 2^2 + 2^1$

$3 = 2^1 + 2^0$

- Le code transmis est

11	10	9	8	7	6	5	4	3	2	1
1	0	0	1	1	1	0	0	1	0	1

Codes de Hamming: un exemple (suite 1)

- Code transmis

11	10	9	8	7	6	5	4	3	2	1
1	0	0	1	1	1	0	0	1	0	1

- Vérification :
  considérer les bits de données et de contrôle qui sont égaux à 1 : calculer leur représentation en base 2

11	=	1	0	1	1	$11 = 2^3 + 2^1 + 2^0$
8	=	1	0	0	0	$8 = 2^3$
7	=	0	1	1	1	$7 = 2^2 + 2^1 + 2^0$
6	=	0	1	1	0	$6 = 2^2 + 2^1$
3	=	0	0	1	1	$3 = 2^1 + 2^0$
1	=	0	0	0	1	$1 = 2^0$
	=	0	0	0	0	

la somme est égale à 0 → pas d'erreur de transmission

Codes de Hamming: un exemple (suite 2)

- Code transmis : 1 erreur de transmission sur le bit # 11

11	10	9	8	7	6	5	4	3	2	1
0	0	0	1	1	1	0	0	1	0	1

- Vérification :
  considérer les bits de données et de contrôle qui sont égaux à 1 : calculer leur représentation en base 2

8	=	1	0	0	0
7	=	0	1	1	1
6	=	0	1	1	0
3	=	0	0	1	1
1	=	0	0	0	1
	=	**1**	**0**	**1**	**1**

$8 = 2^3$

$7 = 2^2 + 2^1 + 2^0$

$6 = 2^2 + 2^1$

$3 = 2^1 + 2^0$

$1 = 2^0$

- Identification du bit erroné $\rightarrow$ $11 = 2^3 + 2^1 + 2^0$ soit 1011 en base 2 !

Codes de Hamming: un exemple (suite 3)

- Code transmis : 2 erreurs de transmission sur les bits # 11 et # 7

11	10	9	8	7	6	5	4	3	2	1
0	0	0	1	0	1	0	0	1	0	1

- Vérification :
  considérer les bits de données et de contrôle qui sont égaux à 1 : calculer leur représentation en base 2

8	=	1	0	0	0
6	=	0	1	1	0
3	=	0	0	1	1
1	=	0	0	0	1
	=	**1**	**1**	**0**	**0**

$8 = 2^3$

$6 = 2^2 + 2^1$

$3 = 2^1 + 2^0$

$1 = 2^0$

→ identification d'au moins 1 erreur de transmission
→ pas de possibilité d'identifier les bits erronés
→ pas possible d'identifier les erreurs de transmission sur plus que 2 bits

Contrôle de redondance cyclique (CRC)
Cyclic Redundancy Code

- Traite le bloc de données comme un énorme nombre binaire
- Division du nombre binaire par un autre nombre fixé à l'avance
- Reste de la division $\rightarrow$ valeur du CRC
- Récepteur refait le calcul avec le même diviseur: comparaison de sa valeur avec le CRC reçu
- Effet de chaque bit du bloc de données est différent sur le résultat final à cause de sa position dans le dividende

Contrôle de redondance cyclique (CRC)
Cyclic Redundancy Code

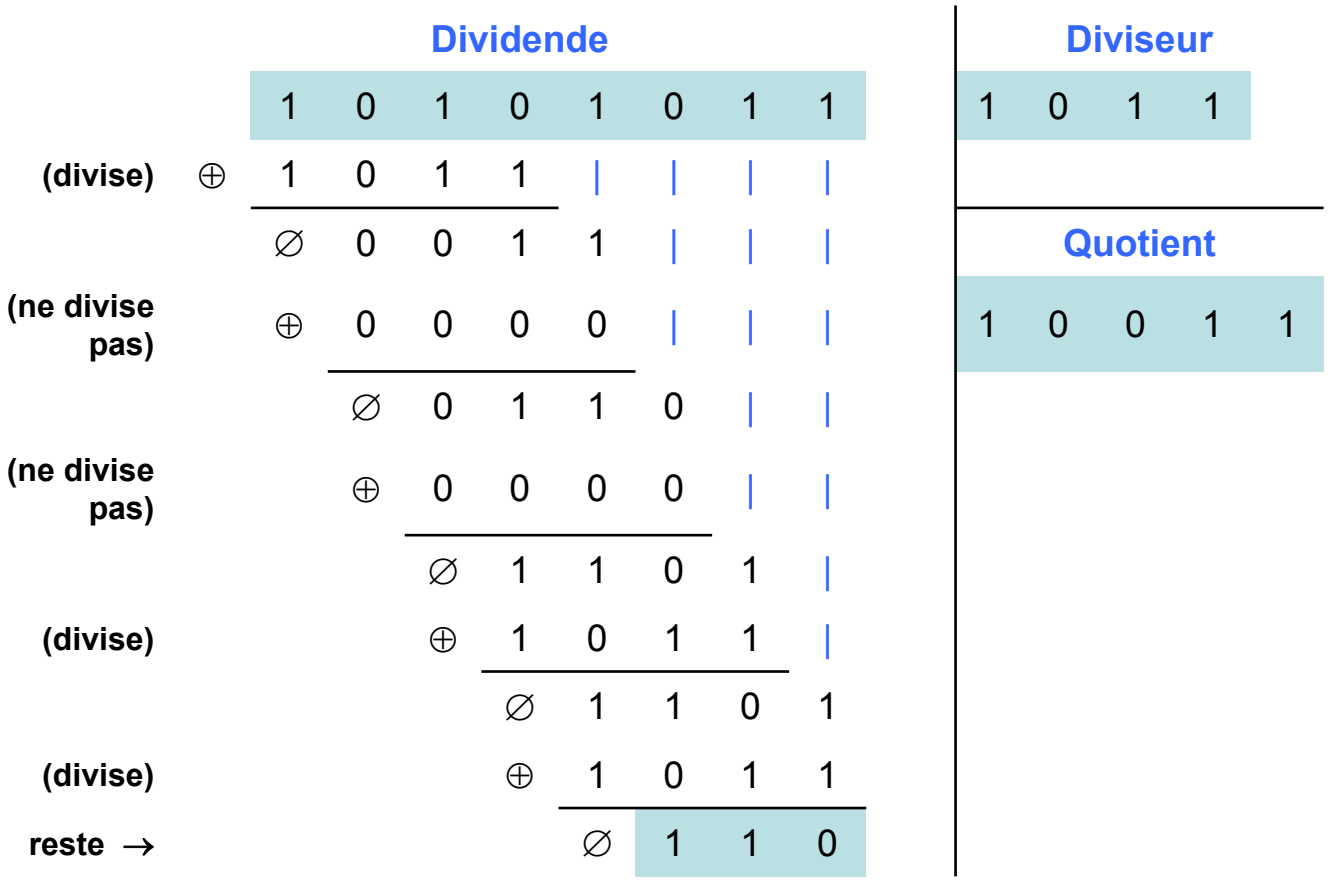
Division appliquée sur un bloc de données considéré comme un nombre

Format du bloc de données	Dividende	Diviseur	Quotient	Reste
Hexadécimal	<42><6C><6F><63>	<D3>	<5096F9>	<28>
	<426C6F63>			
Binaire	<01000010><01101100><01101111><01100011>	<11010101>	<010100001001011011111001>	<00101000>
	<01000010011011000110111101100011>			

Contrôle de redondance cyclique (CRC)
Cyclic Redundancy Code

- **Arithmétique modulo 2** : exécuter les calculs (addition et soustraction) en négligeant la retenue ou l'emprunt → OU exclusif ⊕
- **Opération réversible** : si on applique le deuxième opérande sur le résultat de la première opération
  10011 ⊕ 01001 = 11010
  11010 ⊕ 01001 = 10011
- **Négliger la notion de grandeur après le bit le plus significatif**
  1010 > 0010
  mais 1010 n'est pas plus grand que 1001 :
  1010 + 0011 = 1001 (on oublie les retenues)
  1010 – 0011 = 1001 (on oublie les emprunts)
  Observation : 1010 ⊕ 0011 = 1001
- Effectuer la division lorsque le nombre divisé et le diviseur sont de la **même grandeur**

Contrôle de redondance cyclique (CRC)
Cyclic Redundancy Code
Division en arithmétique modulo-2



Méthode de calcul du CRC

- **Émetteur**
  - M(x) polynôme associé au bloc de données. Bits sont les coefficients du polynôme. Dividende $\leftarrow$ M(x).
  - Diviseur : polynôme de contrôle G(x).
  - Bloc M(x) est augmenté de r = degré(G(x)) zéros $\rightarrow$ M'(x) = 2^r M(x)
  - Division : 2^r M(x) / G(x) arithmétique modulo 2

    $\rightarrow$ 2^r M(x) = Q(x) G(x) + R(x) $\rightarrow$ 2^r M(x) / G(x) = Q(x) + R(x) / G(x)
  - T(x) = bloc transmis = R(x) $\oplus$ 2^r M(x)
  - T(x) est divisible par G(x) :

    T(x)/G(x) = (R(x) $\oplus$ 2^r M(x)) / G(x) = Q(x) $\oplus$ R(x)/G(x) $\oplus$ R(x)/G(x) = Q(x)
- **Récepteur**
  - Calculer le CRC du bloc transmis T(x) / G(x)
  - Si le reste est égal à 0 : pas d'erreur de transmission

Calcul de la valeur du CRC pour un bloc de données

ÉMETTEUR

```
1 0 1 1 1 0 1 0 0 1 0 0 0 0  |  1 0 0 1 1
1 0 0 1 1                    |-----------------------------
- 0 1 0 0 0                  |  1 0 1 0 0 1 0 0 1 0
  0 0 0 0 0                  |
  - 1 0 0 0 1
    1 0 0 1 1
   - 0 0 1 0 0
     0 0 0 0 0
    - 0 1 0 0 0
      0 0 0 0 0
    - 1 0 0 0 1
      1 0 0 1 1
     - 0 0 1 0 0
       0 0 0 0 0
     - 0 1 0 0 0
       0 0 0 0 0
     - 1 0 0 0 0
       1 0 0 1 1
     - 0 0 1 1 0
       0 0 0 0 0
     - 0 1 1 0
```

RÉCEPTEUR

```
1 0 1 1 1 0 1 0 0 1 0 1 1 0  |  1 0 0 1 1
1 0 0 1 1                    |-----------------------------
- 0 1 0 0 0                  |  1 0 1 0 0 1 0 0 1 0
  0 0 0 0 0                  |
  - 1 0 0 0 1
    1 0 0 1 1
   - 0 0 1 0 0
     0 0 0 0 0
    - 0 1 0 0 0
      0 0 0 0 0
    - 1 0 0 0 1
      1 0 0 1 1
     - 0 0 1 0 0
       0 0 0 0 0
     - 0 1 0 0 1
       0 0 0 0 0
     - 1 0 0 1 1
       1 0 0 1 1
     - 0 0 0 0 0
       0 0 0 0 0
     - 0 0 0 0    →  pas d'erreur !
```

$M(x) = 1\,0\,1\,1\,1\,0\,1\,0\,0\,1$ $\quad C(x) = x^4 + x + 1$ $\quad 2^4\,M(x) = 1\,0\,1\,1\,1\,0\,1\,0\,0\,1\,0\,0\,0\,0$

$R(x) = 0\,1\,1\,0$

$2^4\,M(x) = 1\,0\,1\,1\,1\,0\,1\,0\,0\,1\,0\,0\,0\,0$ $\quad D(x) = 1\,0\,0\,1\,1$ $\quad T(x) = 1\,0\,1\,1\,1\,0\,1\,0\,0\,1\,0\,1\,1\,0$

Calcul de la valeur du CRC pour un bloc de données

ÉMETTEUR

```
1 0 1 1 1 0 1 0 0 1 0 0 0 0  |  1 0 0 1 1
1 0 0 1 1                    |-----------------------------
- 0 1 0 0 0                  |  1 0 1 0 0 1 0 0 1 0
  0 0 0 0 0                  |
  - 1 0 0 0 1
    1 0 0 1 1
   - 0 0 1 0 0
     0 0 0 0 0
    - 0 1 0 0 0
      0 0 0 0 0
    - 1 0 0 0 1
      1 0 0 1 1
     - 0 0 1 0 0
       0 0 0 0 0
     - 0 1 0 0 0
       0 0 0 0 0
     - 1 0 0 0 0
       1 0 0 1 1
     - 0 0 1 1 0
       0 0 0 0 0
     - 0 1 1 0
```

RÉCEPTEUR

```
1 0 1 1 1 0 1 0 0 1 0 1 1 1  |  1 0 0 1 1
1 0 0 1 1                    |-----------------------------
- 0 1 0 0 0                  |  1 0 1 0 0 1 0 0 1 0
  0 0 0 0 0                  |
  - 1 0 0 0 1
    1 0 0 1 1
   - 0 0 1 0 0
     0 0 0 0 0
    - 0 1 0 0 0
      0 0 0 0 0
    - 1 0 0 0 1
      1 0 0 1 1
     - 0 0 1 0 0
       0 0 0 0 0
     - 0 1 0 0 1
       0 0 0 0 0
     - 1 0 0 1 1
       1 0 0 1 1
     - 0 0 0 0 1
       0 0 0 0 0
     - 0 0 0 1
```

Question
Que se passe t-il si
E(x) = 00000100000000 ?

→ erreur de transmission

$M(x) = 1\,0\,1\,1\,1\,0\,1\,0\,0\,1$

$2^4\,M(x) = 1\,0\,1\,1\,1\,0\,1\,0\,0\,1\,0\,0\,0\,0$

$C(x) = x^4 + x + 1$

$D(x) = 1\,0\,0\,1\,1$

$2^4\,M(x) = 1\,0\,1\,1\,1\,0\,1\,0\,0\,1\,0\,0\,0\,0$

$R(x) = 0\,1\,1\,0$

$T(x) = 1\,0\,1\,1\,1\,0\,1\,0\,0\,1\,0\,1\,1\,0$

Polynômes de contrôle et CRC

CRC	Polynôme de contrôle
CRC-6	$x^{16} + x^{15} + x^2 + 1$
CRC-12	$x^{12} + x^{11} + x^3 + x^2 + x + 1$
CRC-16	$x^{16} + x^{15} + x^2 + 1$
CRC-CCITT	$x^{16} + x^{12} + x^5 + 1$
CRC-32	$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$

Taux de détection des erreurs avec un CRC sur 16 bits

Type d'erreur	Taux de détection
Sur 1 bit	100 %
Sur 2 bits	100 %
Nombre impair de bits	100 %
Suite de 16 bits ou moins	100 %
Suite de 17 bits	99,997 %
Suite de 18 bits ou plus	99,998 %