

## DS 1 : Détection d'erreurs par le codage CRC

• Bit de parité

Question 1.

```
let ou_exclusif = fun
  0 0 -> 0
  | 0 1 -> 1
  | 1 0 -> 1
  | 1 1 -> 0
  | _ _ -> failwith "ou_exclusif" ;;
```

La dernière ligne de la fonction correspond simplement au souhait d'avoir un filtrage exhaustif; elle n'est pas indispensable ici.

Question 2.

- 2.a On peut calculer  $b_n$  de plusieurs manières, par exemple à l'aide de la formule :  $b_n = b_0 \oplus b_1 \oplus \dots \oplus b_{n-1}$ . On peut donc calculer  $b_n$  par l'intermédiaire de l'itération :

$$u_0 = b_0 \quad \text{et} \quad \forall k \in \llbracket 1, n-1 \rrbracket, \quad u_k = u_{k-1} \oplus b_k.$$

```
let bit_parité b =
  let n = vect_length b in
  let rec u = function
    0 -> b.(0)
    | k -> ou_exclusif (u (k-1)) b.(k)
  in u (n-1) ;;
```

- 2.b Si le message reçu contient un nombre impair de bits égaux à 1, il y a donc eu un nombre impair d'erreurs de transmissions, donc au moins une! En revanche, un nombre pair d'erreurs de transmissions ne sera pas détecté.

• Le codage CRC

Question 3.

Posons  $P_k(X) = b_{n-1-k}X^k + b_{n-k}X^{k-1} + \dots + b_{n-2}X + b_{n-1}$  pour  $k \in \llbracket 0, n-1 \rrbracket$ , et  $P_{-1}(X) = 0$ .

Alors pour tout  $k \geq 0$ ,  $\deg P_k = \begin{cases} k & \text{si } b_{n-k-1} = 1 \\ \deg P_{k-1} & \text{sinon} \end{cases}$ . D'où la fonction :

```
let degré b =
  let n = vect_length b in
  let rec deg = function
    -1 -> -1
    | k when b.(n-1-k) = 1 -> k
    | k -> deg (k-1)
  in deg (n-1) ;;
```

Question 4.

```
let plus b c i j l =
  for k = 0 to l-1 do
    b.(i+k) <- ou_exclusif b.(i+k) c.(j+k)
  done ;;
```

**Question 5.**

- 5.a** Si le mot a été transmis sans erreur, il est associé au polynôme  $T(X) = X^k P(X) + R(X)$ ; or celui-ci est par définition divisible par  $G(X)$ ; ainsi,  $(X^k P(X) + R(X)) \bmod G(X) = 0$ .
- 5.b** Réciproquement, si on note  $T'(X)$  le polynôme associé au mot reçu, posons  $E(X) = T(X) \oplus T'(X)$ ; le message est donc transmis sans erreur si et seulement si  $E(X) = 0$ . Or il est tout à fait possible d'avoir  $E(X) \neq 0$  sans que l'erreur soit détectée; il suffit que  $E(X)$  soit divisible par  $G(X)$ . On verra néanmoins qu'un choix judicieux de  $G(X)$  rend cette situation très improbable.

**Question 6.**

- 6.a** Puisque  $G(X)$  divise  $T(X)$ , si  $G(X)$  ne divise pas  $E(X)$ , il ne divise pas non plus  $T'(X)$ , et donc  $T'(X) \bmod G(X) \neq 0$ ; l'erreur est détectée.
- 6.b** Une erreur sur un seul bit correspond à  $E(X) = X^i$  avec  $i \in \llbracket 0, n+k \rrbracket$ ; si  $G(X)$  n'est pas un monôme,  $E(X)$  n'est pas divisible par  $G(X)$  et l'erreur est détectée.
- 6.c** Supposons que  $G(X)$  soit divisible par  $(X+1)$ , et soit  $E(X)$  une erreur non détectée par le CRC. Alors  $G(X)$  divise  $E(X)$  et donc  $(X+1)$  aussi. On en déduit que 1 est racine de  $E(X)$ :  $E(1) = 0$ . Mais ceci ne peut avoir lieu que si l'erreur contient un nombre pair de 1.  
Ainsi, toute erreur portant sur un nombre impair de bits est détectée.

**Question 7.**

- 7.a** Un paquet d'erreurs de longueur  $\ell$  correspond à un polynôme  $E(X) = X^{i+\ell-1} + \dots + X^i = X^i F(X)$  avec  $\deg F = \ell - 1$ . Supposons que  $G(X)$  divise  $E(X)$ . Si le coefficient constant de  $G(X)$  n'est pas nul,  $G(X)$  est premier avec  $X^i$  donc  $G(X)$  divise  $F(X)$ . Puisque  $\deg G(X) = k$ , on a :  $\ell - 1 \geq k$ , soit  $\ell > k$ . En contraposant, on en déduit que tout paquet d'erreur de longueur  $\ell \leq k$  est détecté.
- 7.b** Un paquet d'erreurs de longueur  $k+1$  correspond à un polynôme  $E(X) = X^{i+k} + \dots + X^i = X^i F(X)$  avec  $\deg F = k$ . Si cette erreur n'est pas détectée,  $G(X)$  divise  $F(X)$ , et puisqu'ils ont même degré,  $F(X) = G(X)$ . Il y a donc un seul paquet d'erreurs non détecté, parmi les  $2^{k-1}$  possibles (correspondants au choix des coefficients de  $X^{i+1}, X^{i+2}, \dots, X^{i+k-1}$  dans  $E(X)$ ), donc une probabilité égale à  $\frac{1}{2^{k-1}}$ .
- 7.c** Un paquet d'erreurs de longueur  $k+p$ , avec  $p \geq 2$ , correspond à un polynôme  $E(X) = X^{i+k+p-1} + \dots + X^i = X^i F(X)$ , avec  $\deg F = k+p-1$ . Si cette erreur n'est pas détectée,  $F(X) = G(X)Q(X)$ , avec  $\deg Q = p-1$ . De plus,  $X$  ne divise pas  $F(X)$ , donc le coefficient de  $Q(X)$  n'est pas nul. Ainsi,  $Q(X) = X^{p-1} + \dots + 1$ ; ce qui donne  $2^{p-2}$  polynômes possibles. La probabilité que cette erreur ne soit pas détectée est donc égale à :  $\frac{2^{p-2}}{2^{k+p-2}} = \frac{1}{2^k}$ ; la probabilité qu'elle le soit est donc égale à  $1 - \frac{1}{2^k}$ .
- 7.d** Nous avons  $G(X) = (X+1)(X^{15} + X + 1)$ , donc d'après la question 6.c, les erreurs en nombre impair sont détectées.  
Le coefficient constant de  $G$  n'est pas nul, donc d'après les questions précédentes, tous les paquets d'erreurs de longueur inférieure ou égale à 16 sont détectés; la probabilité de détecter un paquet d'erreurs de longueur 17 est égale à  $1 - \frac{1}{2^{15}} \approx 99,997\%$ ; la probabilité de détecter un paquet d'erreurs de longueur supérieure ou égale à 18 est égale à  $1 - \frac{1}{2^{16}} \approx 99,998\%$ .

**Question 8.**

- 8.a** Nous allons effectuer les calculs dans un tableau auxiliaire  $c$  correspondant à un polynôme  $C(X)$  initialement égal au polynôme  $X^k P(X)$ , et tant que  $\deg C > k$ , on remplace  $C(X)$  par  $C(X) \oplus X^{\deg C - k} G(X)$ .

```

let crc b g =
  let n = vect_length b and p = vect_length g in
  let k = degré g in
  let c = make_vect (n+k) 0 in
  plus c b 0 0 n ;      (* recopie bX^k dans c *)
  while degré c >=k do
    plus c g (n+k-1-degré c) (p-1-k) (k+1)
  done ;
  sub_vect c n k ;;
    
```

**8.b** Le coût spatial de cette fonction est lié à la création du tableau  $c$  ; c'est donc un  $\Theta(n+k)$ .  
 Le nombre d'itération est majoré par  $n$ , et la fonction `plus` a un coût proportionnel à son dernier argument, donc le coût temporel est un  $O(nk)$ .

**Question 9.**

Pour tout  $i \in \llbracket 0, n+k-2 \rrbracket$ ,

$$R_{i+1}(X) = (X(b_0X^i + b_1X^{i-1} + \dots + b_{i-1}X + b_i) + b_{i+1}) \bmod G(X) = (XR_i(X) + b_{i+1}) \bmod G(X).$$

Posons  $R_i(X) = \alpha_0X^{k-1} + \alpha_1X^{k-2} + \dots + \alpha_{k-1}$ .

Alors  $XR_i(X) + b_{i+1} = \alpha_0X^k + \alpha_1X^{k-1} + \dots + \alpha_{k-1}X + b_{i+1}$  donc  $\deg(XR_i(X) + b_{i+1}) \oplus \alpha_0G(X) \leq k-1$  et par conséquent :

$$(XR_i(X) + b_{i+1}) \oplus \alpha_0G(X) = (XR_i(X) + b_{i+1}) \bmod G(X) = R_{i+1}(X).$$

On notera en particulier que :  $R_{n-1+k}(X) = X^kP(X) \bmod G(X)$  ; cette formule permet le calcul par récurrence du CRC.

**Question 10.**

**10.a** Les valeurs successives prises par le tableau  $\llbracket r_0; r_1; r_2; r_3; r_4 \rrbracket$  définissent une suite de polynômes  $(\tilde{R}_i(X))_{0 \leq i \leq n+4}$  débutant ainsi :

$$\begin{aligned} \tilde{R}_0(X) &= b_0 \\ \tilde{R}_1(X) &= b_0X + b_1 \\ \tilde{R}_2(X) &= b_0X^2 + b_1X + b_2 \\ \tilde{R}_3(X) &= b_0X^3 + b_1X^2 + b_2X + b_3 \\ \tilde{R}_4(X) &= b_0X^4 + b_1X^3 + b_2X^2 + b_3X + b_4 \end{aligned}$$

et enfin :  $\tilde{R}_5(X) = (b_1X^4 + b_2X^3 + b_3X^2 + b_4X + b_5) \oplus (b_0X^4 + b_0X^2 + b_0)$

Or, puisque  $b_0 \oplus b_0 = 0$ , on peut aussi écrire :

$$\tilde{R}_5(X) = (b_0X^5 + b_1X^4 + b_2X^3 + b_3X^2 + b_4X + b_5) \oplus (b_0X^5 + b_0X^4 + b_0X^2 + b_0),$$

soit :

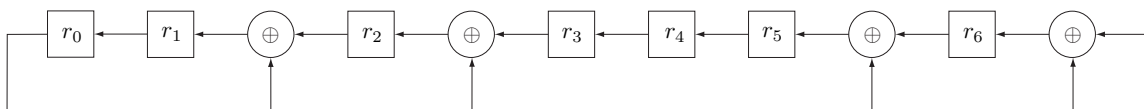
$$\tilde{R}_5(X) = (b_0X^5 + b_1X^4 + b_2X^3 + b_3X^2 + b_4X + b_5) \oplus (b_0G(X)) = (b_0X^5 + b_1X^4 + b_2X^3 + b_3X^2 + b_4X + b_5) \bmod G(X).$$

Plus généralement, si on note  $\tilde{R}_i(X) = \alpha_0X^4 + \alpha_1X^3 + \alpha_2X^2 + \alpha_3X + \alpha_4$ , alors :

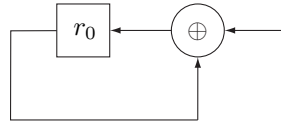
$$\tilde{R}_{i+1}(X) = (X\tilde{R}_i(X) + b_{i+1}) \oplus (\alpha_0G(X))$$

donc  $\tilde{R}_i(X) = R_i(X)$  et en particulier,  $\tilde{R}_{n+4}(X)$  est le polynôme associé au CRC.

Ainsi, le circuit associé au polynôme générateur  $G(X) = X^7 + X^5 + X^4 + X + 1$  est :



10.b Considérons le polynôme générateur  $G(X) = X + 1$ . Il correspond au circuit suivant :



Autrement dit, le CRC est ici égal à :  $b_0 \oplus b_1 \oplus \dots \oplus b_{n-1}$  ; c'est le bit de parité.

