Numération Codage

Q1 Ecrire dans le format virgule flottante simple précision le nombre 14,75

Q2 Donner le résultat en hexadécimal.

Algèbre de Boole

Q3 Calculer

0.1

$1+1$

1.1

$1\oplus\overline{1}$

$1\oplus1$

$\overline{\overline{0}\oplus\overline{1}}$

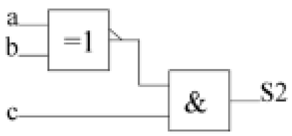
$\overline{1\oplus1}$

Fonction logique

Q4 Déterminer l'équation de la sortie S_1 sous une forme simplifiée

S_1	S_2	S_3	M
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Q5 Déterminer l'équation de la sortie S_2



Problème : le contrôle d'erreur

CRC (Code de redondance cyclique)

Le CRC est fréquemment utilisé pour contrôler l'intégrité d'un message ou d'u fichier. C'est une valeur calculée sur un morceau de données avant compression. Quand l'archiver déballe ce fichier, il lit la valeur du CRC précédemment écrit et contrôle la validité du fichier non décompressé en refaisant le même calcul. Quand les deux résultats correspondent, il y a de bonne chance que les fichiers soient identiques.

Principe :

L'idée fondamentale des algorithmes de CRC est simplement de traiter le message comme une grande valeur numérique (sous forme polynomiale) , de le diviser par une valeur fixe (polynôme générateur) , et **de prendre le reste de la division en tant que somme de contrôle.**

La division binaire ressemble beaucoup à une division euclidienne, le résultat donne un quotient et un reste. Le calcul bit à bit se fait par un "ou exclusif" (encore appelé XOR).

Message à transmettre : 1 1 0 1 0 1 1 0 1 1 0 0 0 0

Soit le polynôme (Dividende) : $X^{13}+X^{12}+X^{10}+X^{8}+X^{7}+X^{5}+X^{4}$

Polynôme générateur (Diviseur) X^4+X+1 soit 1 0 0 1 1

	1	1	0	1	0	1	1	0	1	1	0	0	0	0	1 0 0 1 1
xor	1	0	0	1	1	↓	↓	↓	↓	↓	↓	↓	↓	↓	
=		1	0	0	1	1	↓	↓	↓	↓	↓	↓	↓	↓	1 1 0 0 0 0 1 0 1 0
	xor	1	0	0	1	1	↓	↓	↓	↓	↓	↓	↓	↓	
	=		0	0	0	0	1	↓	↓	↓	↓	↓	↓	↓	
		xor	0	0	0	0	0	↓	↓	↓	↓	↓	↓	↓	
		=		0	0	0	1	0	↓	↓	↓	↓	↓	↓	
			xor	0	0	0	0	0	↓	↓	↓	↓	↓	↓	
			=		0	0	1	0	1	↓	↓	↓	↓	↓	
				xor	0	0	0	0	0	↓	↓	↓	↓	↓	
				=		0	1	0	1	1	↓	↓	↓	↓	
					xor	0	0	0	0	0	↓	↓	↓	↓	
					=		1	0	1	1	0	↓	↓	↓	
						xor	1	0	0	1	1	↓	↓	↓	
						=		0	1	0	1	0	↓	↓	
							xor	0	0	0	0	0	↓	↓	
							=		1	0	1	0	0	↓	
								xor	1	0	0	1	1	↓	
								=		0	1	1	1	0	
									xor	0	0	0	0	0	
									=		1	1	1	0	

On sélectionne le bit de poids fort du dividende, ici (1), il sera le bit de poids fort du quotient.
Maintenant, on peut faire (1 x diviseur) = 10011. On fait ensuite un XOR entre le dividende et ce résultat qui est décalé de suffisamment de bits pour commencer au début du dividende.
Ensuite, on ajoute au résultat de l'opération le bit suivant correspondant au dividende.
L'opération s'enchaîne ainsi de suite. On sélectionne le bit de poids fort du résultat précédemment trouvé, ici, c'est un 1. On lui applique un XOR avec 10011. Le résultat sera 0000, auquel on fera descendre 1, ce qui donne 00001. Le bit de poids fort est un 0, donc (0x diviseur)=00000. On refait l'opération avec le XOR. Ces opérations seront réalisées jusqu'à ce qu'on ne puisse plus " descendre " de bit depuis le dividende, donc on réalise (Taille du dividende-Taille du diviseur+1) opérations (en fait, la taille correspondant au nombre de bits).

Le Code CRC est | 1 1 1 0 | et sera transmis avec le message | 1 1 0 1 0 1 1 0 1 1 0 0 0 0 |

Application:

- **Q6** En utilisant le polynôme générateur G(x) = $X^4 + X + 1$, calculer le CRC ajouté pour la transmission des 10 bits de données 1101011011.
- **Q7** Donner le résultat sous un format 16 bits justifié à gauche en hexadécimal.
- **Q8** Soit la trame reçue suivante : $2B90. G(x) est toujours $X^4 + X + 1$. Déterminer le CRC calculé à la réception.
- **Q9** Quelle opération logique peut donner le résultat de la comparaison des CRC (reçu et calculé à la réception) pour déterminer s'il y a une erreur de transmission E ? (E=0 si aucune erreur de transmission).
- **Q10** L'appliquer à la trame reçue $2B90 en calculant la valeur de E.