

Un canal bruité, un CD rayé, ou bien un trou lu ou non lu sur une carte perforée dans les années 50, à l'époque où Richard Hamming travaillait sur un ordinateur de faible fiabilité produisent des erreurs qui se manifestent par le changement d'un état «0» en un «1» ou d'un «1» en un «0» lors d'une transmission de données numériques. Il existe des méthodes de codage qui permettent de détecter la présence d'une ou plusieurs erreurs dans un mot : il s'agit des codes détecteurs d'erreurs. Le codage étudié ici permet la détection mais aussi la correction d'une erreur se produisant lors de la transmission de messages de 4 bits. Il s'agit du code détecteur et correcteur de Hamming dans sa version (7,4), fondé sur un contrôle de parité.

D'une manière générale on associe à un message de n bits, k bits de contrôle, ce qui totalise $n+k$ bits à transmettre. Dans l'hypothèse d'une seule erreur par mot de $(n+k)$ bits il faut distinguer $(n+k)$ possibilités ainsi que l'absence d'erreur. Par conséquent, il faut choisir k de sorte que :

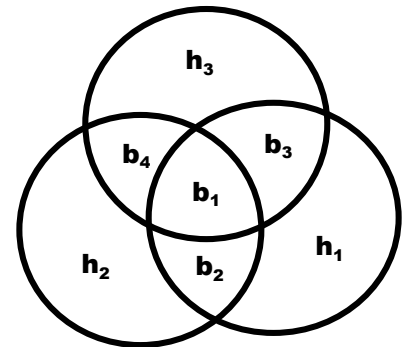
$$2^k \geq n+k+1.$$

Dans le cas particulier du code (7,4) de Hamming le message est donc codé dans le format ci-contre :

h3	h2	b4	h1	b3	b2	b1
----	----	----	----	----	----	----

Chaque bit de hamming h_i contrôle la parité d'un groupe de 3 bits comme cela est schématisé ci-contre. Ainsi :

- $h1$ examine la parité de $(b1, b2, b3)$
- $h2$ examine la parité de $(b1, b2, b4)$
- $h3$ examine la parité de $(b1, b3, b4)$



Liens intéressants :

http://en.wikipedia.org/wiki/Hamming_code
http://www2.rad.com/networks/1994/err_con/hamming.htm
http://fr.wikipedia.org/wiki/Code_correcteur

I. CODAGE A L'EMISSION

CONTROLE DE PARITE

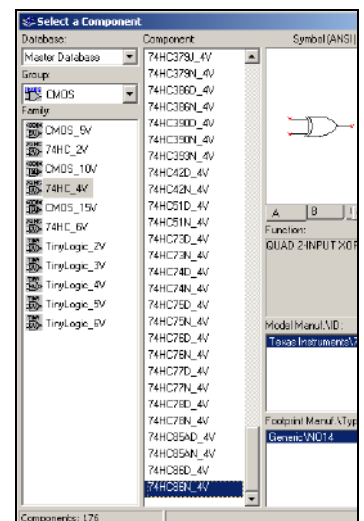
Q1*. Dresser la table de vérité d'une fonction détectant la parité du nombre de 1 dans un mot de 3 bits $b_1b_2b_3$ telle que :
 $p(b_1b_2b_3)=0$, pour un nombre de 1 paire.
 $p(b_1b_2b_3)=1$, pour un nombre de 1 impaire.
 En déduire l'expression simplifiée de $p(b_1b_2b_3)$.

Q2. Réaliser cette fonction avec les portes logiques élémentaires de la famille 74HCxx_4V.

- Tester son fonctionnement en simulant en entrées une séquence de mots judicieuse au moyen du WORD GENERATOR.
- Sauvegarder ce circuit en tant qu'un sous-circuit que nous désignons par SUBDET3. (suivre le mode opératoire fourni en annexe).

Q3. Réaliser un contrôle de parité dans un mot de 4 bits en utilisant SUBDET3 et des portes logiques de la famille 74HCxx_4V.

- Suivre le mode opératoire résumé en annexe pour sauvegarder ce circuit comme un deuxième sous-circuit que nous désignerons par SUBDET4.
- Comment peut-on utiliser ces circuits (en émission ou en réception) pour le codage d'un message et la détection (sans correction) d'une erreur lors de la transmission de données de 3 bits.



CODAGE (7,4) DE HAMMING

Q4. Saisir le montage d'un circuit pour la synthèse des bits de Hamming (h_1 , h_2 et h_3). Tests et simulations.

II. DECODAGE A LA RECEPTION

DETECTION D'ERREUR

Q4. A la réception, nous synthétisons trois signaux S_i tels que :

- $S_1 = (h1 \oplus b1 \oplus b2 \oplus b3).$
- $S_2 = (h2 \oplus b1 \oplus b2 \oplus b4).$
- $S_3 = (h3 \oplus b1 \oplus b3 \oplus b4).$

Dresser la table de ces signaux en considérant l'occurrence des erreurs ci-contre.

h3	h2	b4	h1	b3	b2	b1	S1	S2	S3
x									
	x								
		x							
			x						
				x					
					x				
						x			

Q5. Quelle est l'indication fournie par le mot (S1,S2,S3) ?

Q6. Réaliser le circuit de décodage et de correction en employant un décodeur 3→8 (voir TP2), et des portes logiques élémentaires.

III. ANNEXES

1 Après avoir réalisé un circuit : Clic droit/Place Schematics HB/SC Connector.

2 Définir le sous-circuit.

3 Lui donner un nom.

4 Sélection puis copie.

5 Double clic sur le sous-circuit pour passer en mode d'édition avec **Edit HB/SC**.

6 Coller le circuit sélectionné dans le fichier apparaissant sous un nouvel onglet. Sauvegarde.

7 Apparition du sous-circuit avec ses entrées/sorties.

8 Reproduction par copier/coller autant que nécessaire.