Magistère de Mathématiques Fondamentales & Appliquées et d'Informatique

Corrigé du contrôle partiel du cours : Systèmes Digitaux.

Le 13/11/01 de 9h30 à 12h30.

Un *code Gray* sur n bits est une *permutation* g_n des entiers $k < 2^n$, telle que :

1. $g_n(0) = 0$;
2. les codes $g_n(k)$ et $g_n(k+1)$ de deux entiers consécutifs sont différents sur *exactement un* bit :

$$g_n(k) \oplus g_n(k+1) = 2^{p_n(k)},$$

avec $0 \leq p_n(k) < n$ pour $0 < k+1 < 2^n$ et pour $k+1 = 2^n = 0 \pmod{2^n}$.

Example : $g_2\{0\,1\,2\,3\} = \{0\,1\,3\,2\}$ est un code Gray sur 2 bits ; sa table binaire, poids faibles à gauche : $_2[00\,10\,11\,01]$

Question 1 (Code Gray) *1. Montrer qu'il existe un code Gray sur n bits, pour tout $n > 0$.*

2. Combien y a t-il de codes Gray : sur 1, 2, 3 bits ? N'envisager les valeurs 4 . . . n qu'à la fin des autres questions.

Réponse 1 *1. L'unique code Gray sur 1 bit est l'identité $g_1 = [0\,1]$.*

Sur 2 bits, il y en a deux : le code standard g_2 *ci-dessus, et son* miroir $\tilde{g}_2 = [0\,2\,3\,1]$.

On construit le code standard g_{n+1} *sur $n+1$ bits à partir du code standard g_n sur n bits :*

$$g_{n+1}(k) = \begin{cases} g_n(k) \text{ si } k < 2^n, \\ 2^n + g_n(2^{n+1} - k - 1) \text{ si } k \geq 2^n. \end{cases} \tag{1}$$

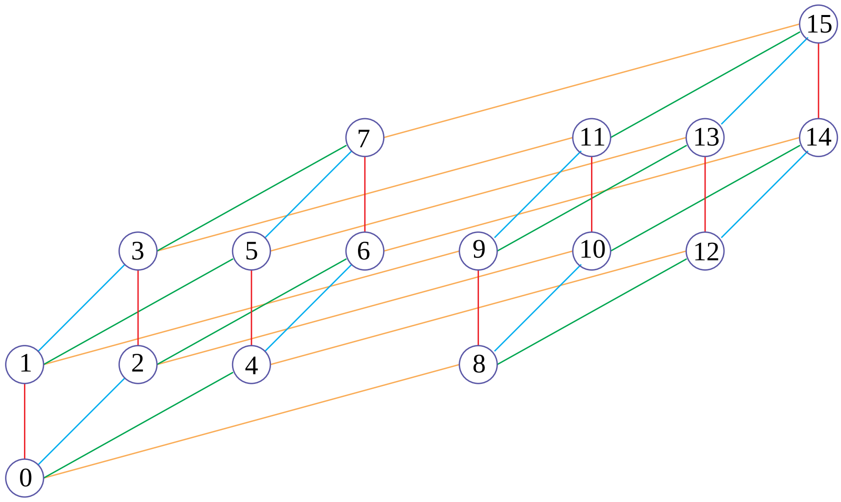
Le miroir *du code standard est donné par $\tilde{g}_n(k) = g_n(2^n - k)$, pour $0 < k < 2^n$, et $\tilde{g}_n(0) = 0$.*

Pour 3 bits, on a :

(a) Décimal : $g_3 = [0\,1\,3\,2\,6\,7\,5\,4]$ et $\tilde{g}_3 = [0\,4\,5\,7\,6\,2\,3\,1]$.

(b) Binaire gauche : $g_3 = [000\,100\,110\,010\,011\,111\,101\,001]$.

2. *On arrive à compter les codes sur trois bits : il y en a 12. Ce sont les images du code standard g_3 et de son miroir $\tilde{g}_3$, par les 6 permutations des 3 bits de sortie.*



A 4 bits, le temps se gâte car il faut compter les cycles hamiltoniens *dans l'hypercube H_4 ci-dessus. Ces cycles partent de l'origine 0 et y reviennent après avoir traversé exactement une fois tous les autres points de l'hypercube.*
*Le calcul par ordinateur indique l'existence de 2688 codes Gray sur 4 bits. La réponse, pour un nombre arbitraire de bits, ne semble pas connue. L'*Encyclopedie Electronique des Suites Entieres*, rubrique 3042 en*
http://www.research.att.com/ njas/sequences/indexfr.html
donne seulement les valeurs : 1 2 12 2688 1813091520.

Question 2 (Compteur Gray sur 3 bits) 1. *Donner explicitement la table binaire d'un code Gray g_3 sur* 3 *bits.*

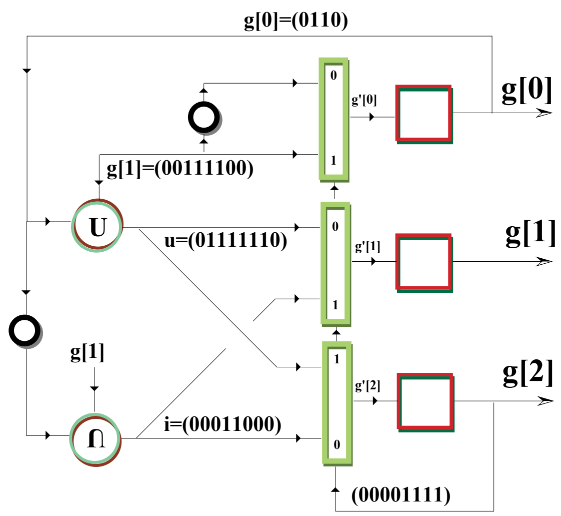
2. *Construire et dessiner* proprement *un circuit digital synchrone G_3, sans entrée, dont la sortie au cycle $t \in \mathbf{N}$ soit la représentation sur* 3 *bits du code de Gray $g_3(t \cdot|\cdot 8)$.*
3. *Analyser le nombre de portes logiques et de registres dans G_3, ainsi que son chemin critique.*

Réponse 2 1. *La table de g_3 est en page 1.*

2. *La table de transition d'une configuration $g = g_3(k) = g_0 + 2g_1 + 4g_2$ vers la suivante $g' = g_3(k+1) = g'_0 + 2g'_1 + 4g'_2$ est donnée par :*

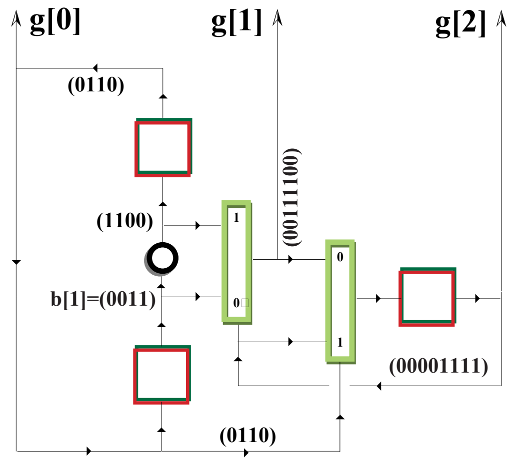
k	$g(k)$			$g(k+1)$		
	g_0	g_1	g_2	g'_0	g'_1	g'_2
0	0	0	0	1	0	0
1	1	0	0	1	1	0
3	0	1	0	0	1	1
2	1	1	0	0	1	0
7	0	0	1	0	0	0
6	1	0	1	0	0	1
4	0	1	1	1	1	1
5	1	1	1	1	0	1

En appliquant la synthèse BDD à cette table, on trouve le circuit G_3 suivant :



3. *G_3 comprend 7 multiplexeurs et 3 registres ; profondeur combinatoire 3.*
4. *Le circuit g_3 qui suit, en 3* **mux** *(dont un inverseur) et 3* **reg** *est le meilleur trouvé.*

Peut-on, ou non, le réaliser avec seulement 2 **mux** *?*



Question 3 (Code inverse) *Montrer que l'on peut toujours transformer un compteur Gray en compteur binaire sur n bits.*

1. *On construira pour cela un circuit combinatoire dont les n entrées proviennent d'un compteur Gray sur n bits, et dont les n sorties sont identiques à celles d'un compteur binaire, pour tout cycle $t \in \mathbf{N}$.*
2. *Optimiser sa taille.*
3. *Optimiser sa profondeur.*

Réponse 3 *L'énoncé ne devrait faire référence qu'aux $n!$ codes dérivés par permutation des bits du code Gray standard. Pour les autres, c'est moins clair, cf. la fin de la question 1.*

1. *Soit $k < 2^n$ un entier présenté sur n bits*

$$k = {}_2 b_0 b_1 \cdots b_{n-1}.$$

La représentation binaire du successeur $k+1$ de k est

$$k+1 = {}_2 b'_0 b'_1 \cdots b'_{n-1},$$

avec $b'_i = 1 - b_i$ pour $i \leq v$ et $b'_i = b_i$ pour $i > v$. Ici $v = v_2(k+1)$ est la valuation *binaire de l'entier $k+1$; c'est la plus grande puissance de deux qui divise $k+1$, soit : $k+1 = 2^v(1+2k')$, et $k = 2^v - 1 + 2^{v+1}k$; il vient $k \oplus (k+1) = 2^{v+1} - 1$.*
On voit que $r(k) = k \oplus (k+1) = 2^{v+1} - 1$ est le vecteur des retenues, *dans la transition $k+1 = k \oplus r(k)$ d'un compteur binaire. En écrivant $j+1 = j \oplus r(j)$ pour $j < k$, et en sommant bit à bit modulo 2, on trouve :*

$$k = \bigoplus_{j<k} r(j). \tag{2}$$

La valuation binaire $v_2(k+1)$ *donne la position de la* retenue la plus significative*, dans la transition d'un compteur binaire à partir de k. C'est aussi la position* $p_n(k) = min\{v_2(k+1), n-1\}$ *de l'*unique *bit à changer pour faire la transition de* $g_n(k)$ *à* $g_n(k+1) = g_n(k) \oplus 2^{p_n(k)}$*, dans le code Gray standard sur n bits.*

k_{10}	${}_2k$			$r(k)$			$2^{p_3(k)}$			$g_3(k)$		
	b_0	b_1	b_2	r_0	r_1	r_2	p_0	p_1	p_2	g_0	g_1	g_2
0	0	0	0	1	0	0	1	0	0	0	0	0
1	1	0	0	1	1	0	0	1	0	1	0	0
2	0	1	0	1	0	0	1	0	0	1	1	0
3	1	1	0	1	1	1	0	0	1	0	1	0
4	0	0	1	1	0	0	1	0	0	0	1	1
5	1	0	1	1	1	0	0	1	0	1	1	1
6	0	1	1	1	0	0	1	0	0	1	0	1
7	1	1	1	1	1	1	0	0	1	0	0	1

Calculons $r(k) \oplus (r(k) \div 2) = (2^{v+1}-1) \oplus (2^v - 1) = 2^v$ *ce qui donne* $2^{p(k)} = r(k) \oplus (r(k) \div 2)$*. Modulo* 2^n*, cette relation devient* $2^{p_n(k)} = r(k) \oplus (r(k) \div 2)$*. En sommant bit à bit et modulo 2 pour* $j < k$*, on trouve :*

$$\begin{aligned} g_n(k) &= \bigoplus_{j<k} 2^{p_n(j)} \\ &= \bigoplus_{j<k} r(j) \oplus \bigoplus_{j<k} (r(j) \div 2). \end{aligned}$$

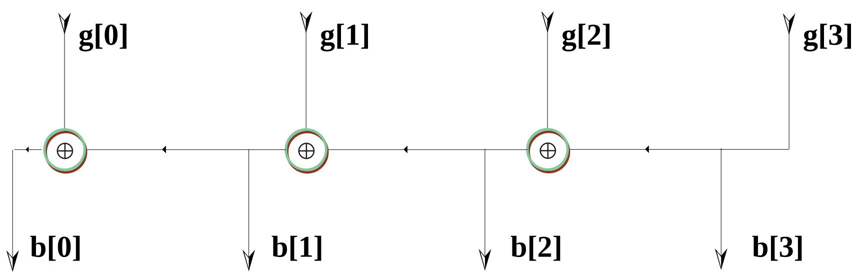
En utilisant (2) deux fois, il vient :

$$g_n(k) = k \oplus (k \div 2). \tag{3}$$

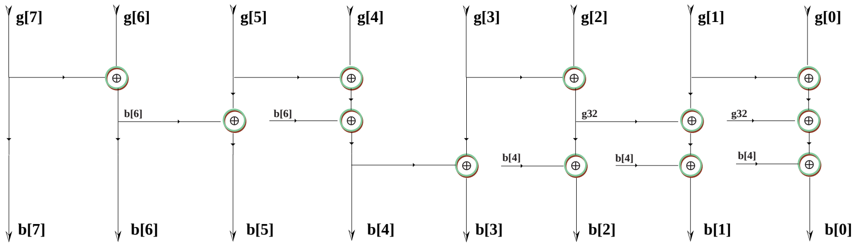
Divisons par deux et ajoutons : $g_n(k) \oplus g_n(k \div 2) = k \oplus (k \div 4)$*. En continuant ainsi, on inverse la formule (3) par :*

$$k = \bigoplus_{0 \le j < n} g_n(k \div 2^j). \tag{4}$$

2. *Traduisons la formule (4) en circuit combinatoire :*



3. *En procédant maintenant par dichotomie :*

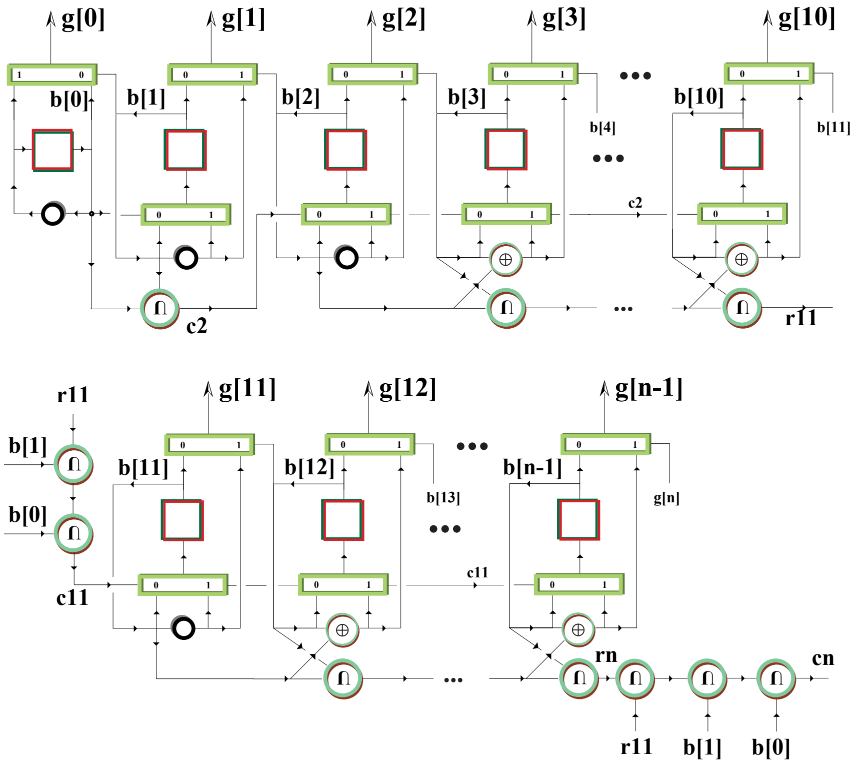


Question 4 (Compteur Gray haute fréquence)

1. *Concevoir un compteur Gray sur n bits, pour n grand ; on veut néanmoins opérer à très haute fréquence.*
2. *Caractériser, en fonction de n et des* délais combinatoires *des portes la fréquence* maximale *d'horloge à laquelle une réalisation cMOS de votre circuit g_n calcule de manière* fiable.

Réponse 4

1. *Partons du compteur* haute fréquence *du cours, et de la formule (4) :*



2. *Le compteur Gray ci-dessus est conçu pour pouvoir opérer jusqu'au* toggle rate, *soit une période d'horloge supérieure à* $\tau = \tau_{reg} + 2\tau_{mux}$, *le délai de registre à registre au travers de deux multiplexeurs.*
   Le chemin combinatoire qui va du registre $b[2]$ *à l'entrée e du registre, (disons)* $g[10]$, *comprend 10* **mux** *en série. La commande* $c2 = {}_2(0001)$ *laisse quatre cycles pour s'établir au signal e, qui est lié au registre source à travers neuf* $\cap$ *et un* **mux**. *Supposons, pour simplifier, que* $\tau_{\cap} = \tau_{mux}$. *A partir de l'instant ou* $c2$ *tombe de 1 à 0, le signal e se* stabilise *au temps* $\tau_{reg} + 10\tau_{mux}$. *Ceci arrive avant le quatrième coup d'horloge (et donc la prochaine commande* $c2 = 1$*) si* $\tau_{reg} + 10\tau_{mux} < 4\tau_{reg} + 8\tau_{mux}$, *soit* $2\tau_{mux} < 3\tau_{reg}$. *C'est une hypothèse technologique raisonnable (sinon, réduire la taille de cette tranche du compteur).*
   La commande $c11$ *de cette tranche a une période de 2K* $=2^{11}$ *cycles. Ceci nous permet d'opérer, à la fréquence* toggle rate, *tous les compteurs - Gray comme binaire - jusqu'à plus de 4K bits.*
   C'est bien trop, en pratique !
   Que ceci n'empêche pourtant pas les curieu(x/ses) de montrer qu'on peut pousser la construction arbitrairement loin, du moins en théorie. Faites-le, c'est amusant.