Option Informatique en Spé MP et MP*
L'additionneur diviser pour régner : le corrigé

Question 1 • Dans $\mathbb{N}$, nous avons $a_0 + b_0 + r = s_0 + 2s_1$. Mais $0 \leqslant s_0 < 2$, donc s_1 et s_0 sont le quotient et le reste dans la division euclidienne de $a_0 + b_0 + r$ par 2. Nous en déduisons le tableau ci-dessous :

a_0	b_0	r	$a_0 + b_0 + r$	s_0	s_1
0	0	0	0	0	0
0	0	1	1	1	0
0	1	0	1	1	0
0	1	1	2	0	1
1	0	0	1	1	0
1	0	1	2	0	1
1	1	0	2	0	1
1	1	1	3	1	1

Dans l'algèbre $\mathcal{B}$, ceci nous donne les relations :

$$\begin{aligned} s_0 &= \overline{a_0}\,\overline{b_0} r + \overline{a_0} b_0 \overline{r} + a_0 \overline{b_0}\,\overline{r} + a_0 b_0 r \\ s_1 &= \overline{a_0} b_0 r + a_0 \overline{b_0} r + a_0 b_0 \overline{r} + a_0 b_0 r \end{aligned}$$

Commençons par construire les sept produits. Il nous faut trois portes NON pour calculer $\overline{a_0}$, $\overline{b_0}$ et $\overline{r}$; puis quatre portes ET pour calculer $a_0 b_0$, $a_0 \overline{b_0}$, $\overline{a_0}\,\overline{b_0}$ et $\overline{a_0} b_0$. Sept autres portes ET sont alors nécessaires pour calculer les sept produits. La figure 1 décrit une implantation possible de cette construction.

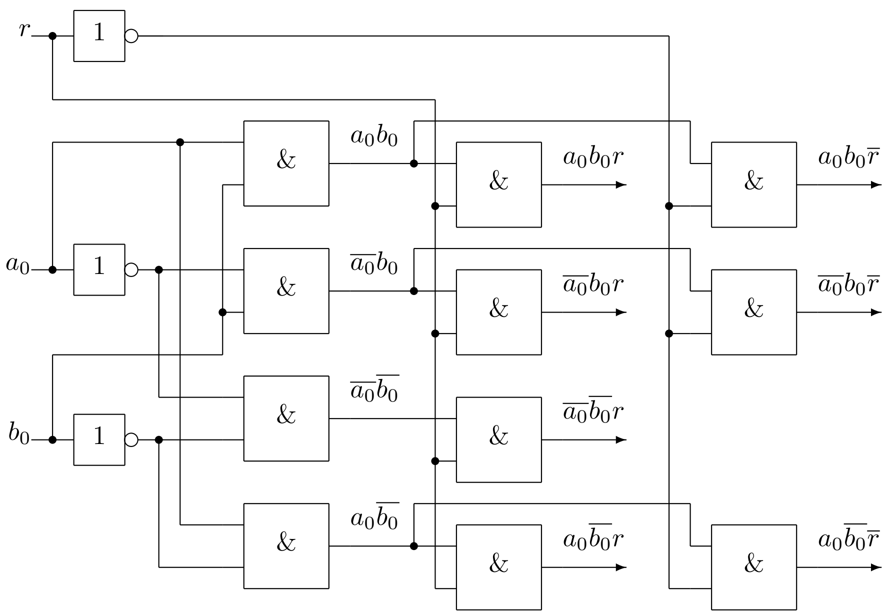

Figure 1: le calcul des sept produits (question 1)

Chacune des deux sommes requiert trois portes OU, pour un total de 6 ; cette partie du circuit est banale, nous ne la représenterons pas. Nous utilisons ainsi 20 portes en tout ; c'est cette valeur qui sera utilisée dans les questions 3 et 10. Bien entendu, il est possible d'économiser quelques portes ; par exemple, avec la méthode de Karnaugh, nous constatons aisément que $s_1 = a_0 b_0 + a_0 r + b_0 r$, et le nombre de portes descend à 18.

Question 2 • Il suffit de relier la sortie s_1 du i-ième additionneur à l'entrée r du $(i+1)$-ième, et ce pour tout $i \in [\![0, n-1]\!]$. L'entrée r de l'additionneur 0 sera forcée à 0 ; la sortie s_1 de l'additionneur $n-1$ sera la retenue issue de la somme des deux nombres.

Question 3 • $s(n) = 20n$ évidemment : les connexions ne coûtent rien.

Question 4 • Le plus long chemin (dans l'additionneur 1-bit) requiert la traversée d'une porte NON, de deux portes ET en cascade et de deux portes OU en cascade ; donc :

$$\boxed{t(n) = 5n\tau}$$

Question 5 • Le gain sera nul, car le temps de propagation t' du circuit proposé par Jean-Marcel vérifie $t'(2n) = 2t'(n)$ et donc $t'(2^n) = 2^n t'(1)$. Notre ami n'a pas volé son nom...

Question 6 • Les définitions de l'énoncé se traduisent par les deux formules $a_0 + b_0 = s_0 + 2g$ et $a_0 + b_0 + 1 = t_0 + 2p$, avec $0 \leqslant s_0 < 2$ et $0 \leqslant t_0 < 2$. Donc g et s_0 (resp. p et t_0) sont le quotient et le reste dans la division euclidienne de $a_0 + b_0$ (resp. $a_0 + b_0 + 1$) par 2. Ceci permet de construire la table ci-dessous :

a_0	b_0	$a_0 + b_0 + 0$	s_0	g	$a_0 + b_0 + 1$	t_0	p
0	0	0	0	0	1	1	0
0	1	1	1	0	2	0	1
1	0	1	1	0	2	0	1
1	1	2	0	1	3	1	1

Dans l'algèbre $\mathcal{B}$, ceci se traduit par $s_0 = a_0\overline{b_0} + \overline{a_0}b_0$, $t_0 = a_0b_0 + \overline{a_0}\,\overline{b_0}$, $g = a_0b_0$ et $p = a_0 + b_0$. Le dessin du circuit correspondant est immédiat : le lecteur voudra bien nous en faire grâce.

Question 7 • Notons :

$$a^D = \sum_{0 \leqslant k < n} 2^k a_k \qquad a^G = \sum_{0 \leqslant k < n} 2^k a_{n+k}$$

Définissons de façon analogue b^D et b^G, et notons :

$$a = a^D + 2^n a_G = \sum_{0 \leqslant k < 2n} 2^k a_k \qquad b = b^D + 2^n b_G = \sum_{0 \leqslant k < 2n} 2^k b_k$$

• $g = 1$ ssi le calcul de $a + b$ génère une retenue, ce qui peut se produire dans deux cas de figure :

1. le calcul de $a^D + b^D$ génère une retenue ($g^D = 1$), qui est propagée par l'étage de gauche, ce qui revient à dire que le calcul de $a^G + b^G + 1$ génère une retenue ($p^G = 1$) ;
2. le calcul de $a^G + b^G$ génère une retenue ($g^G = 1$) car, alors, le calcul de $a^G + b^G + 1$ en génèrera une lui aussi.

Nous résumons ceci par la formule :

$$\boxed{g = g^G + p^G g^D}$$

• $p = 1$ ssi le calcul de $a + b + 1$ génère une retenue, ce qui peut se produire dans deux cas de figure :

1. le calcul de $a^D + b^D + 1$ génère une retenue ($p^D = 1$), qui est propagée par l'étage de gauche, ce qui revient à dire que le calcul de $a^G + b^G + 1$ génère une retenue ($p^G = 1$) ;
2. le calcul de $a^G + b^G$ génère une retenue ($g^G = 1$) car, alors, le calcul de $a^G + b^G + 1$ en génèrera une lui aussi.

Nous résumons ceci par la formule :

$$\boxed{p = g^G + p^G p^D}$$

Question 8 • Étudions d'abord le calcul de $a + b$: l'étage de droite va calculer $a^D + b^D$. S'il ne génère pas de retenue ($g^D = 0$), l'étage de gauche va calculer $a^G + b^G$, si bien que $s_i = s_i^G$; sinon, $s_i = t_i^G$. Résumons ceci par la formule :

$$\boxed{s_i = \overline{g^D} s_i^G + g^D t_i^G}$$

• Voyons maintenant le calcul de $a + b + 1$: l'étage de droite va calculer $a^D + b^D + 1$. S'il ne propage pas la retenue ($p^D = 0$), l'étage de gauche va calculer $a^G + b^G$, si bien que nous aurons $t_i = s_i^G$; sinon, $t_i = t_i^G$. Résumons ceci par la formule :

$$\boxed{t_i = \overline{p^D} s_i^G + p^D t_i^G}$$

Question 9 • Les deux additionneurs n-bits fonctionnent en parallèle, et le sélecteur a un temps de propagation de 3τ ; nous en déduisons $T(2n) = T(n)+3\tau$. L'additionneur 1-bit a lui aussi un temps de propagation $T(1) = 3\tau$, donc avec un télescopage :

$$\boxed{T(2^n) = 3(n+1)\tau}$$

Question 10 • Le tableau ci-dessous montre que, pour une architecture 64 bits ($n = 6$) l'additionneur *diviser pour régner* est quinze fois plus rapide que l'additionneur à propagation de retenue.

n	0	1	2	3	4	5	6
2^n	1	2	4	8	16	32	64
$t(2^n)$	5	10	20	40	80	160	320
$T(2^n)$	3	6	9	12	15	18	21

Question 11 • La question 6 montre que $S(1) = 9$ (nous ne calculons qu'une seule fois a_0b_0). Les questions 7 et 8 montrent que $S(2n) = 2S(n) + 6n + 6$. En effet, le calcul de p et g requiert quatre portes ; celui de $\overline{p^D}$ et $\overline{g^D}$ requiert deux portes (ici encore, le calcul n'est fait qu'une seule fois) ; celui de s_i et p_i requiert alors six portes pour chaque i.

• Nous avons donc $S(2^{n+1}) = 2S(2^n) + 6 \cdot 2^n + 6$ que nous écrivons :

$$\frac{S(2^{n+1})}{2^{n+1}} = \frac{S(2^n)}{2^n} + 3 + 3 \cdot 2^{-n}$$

Avec un télescopage, il vient :

$$\begin{aligned}
\frac{S(2^n)}{2^n} &= S(1) + \sum_{0 \leqslant k < n} \left(\frac{S(2^{k+1})}{2^{k+1}} - \frac{S(2^k)}{2^k} \right) \\
&= 9 + \sum_{0 \leqslant k < n} \left(3 + 3 \cdot 2^{-k}\right) = 9 + 3n + 3(2 - 2^{-n+1}) \\
&= 3n + 15 - 6 \cdot 2^{-n}
\end{aligned}$$

Nous en déduisons :

$$\boxed{S(2^n) = (3n+15)2^n - 6}$$

Question 12 • Le tableau ci-dessous montre que, pour une architecture 64 bits ($n = 6$) l'additionneur *diviser pour régner* requiert environ 50% de portes élémentaires de plus que l'additionneur à propagation de retenue.

n	0	1	2	3	4	5	6
2^n	1	2	4	8	16	32	64
$s(2^n)$	20	40	80	160	320	640	1280
$S(2^n)$	9	30	78	186	426	954	2106

Question 13 • Le signal obtenu en sortie à l'instant 0 dépend de deux signaux au plus (qui doivent donc être établis à l'instant $-\tau$), qui dépendent à leur tour de quatre signaux au plus (qui doivent donc être établis à l'instant -2τ), et ainsi de suite... Les n entrées du circuit doivent donc être établies à l'instant $-k\tau$, avec $2^k \geqslant n$ soit $k \geqslant \lg n$ ou $k \geqslant \lceil \lg n \rceil$.

Question 14 • Notons $(x_1, \ldots, x_{2n}) = (a_0, \ldots, a_{n-1}, b_0, b_{n-1})$. Examinons le cas $1 \leqslant i \leqslant n$, le cas $n < i \leqslant 2n$ se traitant de façon symétrique. Il suffit de prendre $x_j = 0$ pour $1 \leqslant j \leqslant n$ et $x_j = 1$ pour $n < j \leqslant 2n$. Alors :

$$\begin{aligned}
f(x_1, \ldots, x_{i-1}, x_i, x_{i+1}, \ldots, x_{2n}) &= 2^n - 1 \\
f(x_1, \ldots, x_{i-1}, \overline{x_i}, x_{i+1}, \ldots, x_{2n}) &= 2^n
\end{aligned}$$

Le bit de poids 2^n est nul dans le premier cas, égal à 1 dans le deuxième cas.

Question 15 • Un tel circuit compte $n+1$ sorties. Nous venons de voir que l'une de ces sorties est totalement dépendante des $2n$ entrées, et ne peut donc être établie qu'avec un délai au moins égal à $\lceil \lg n \rceil \tau$.

FIN

Cette figure provient du site Web du *Microsystems Prototyping Laboratory* de la *Mississippi State University*.