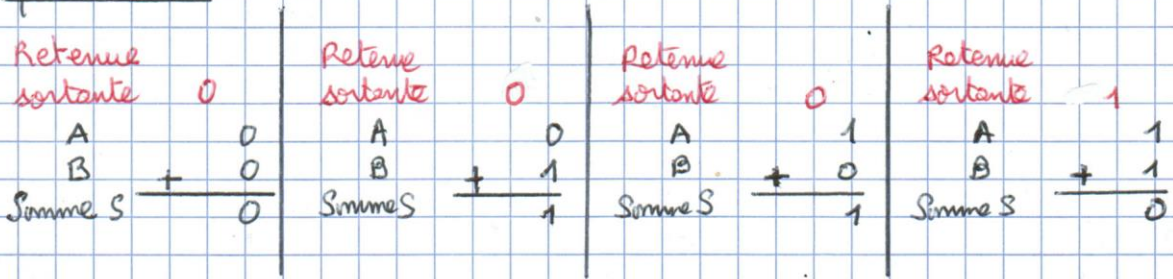


22 PCSI DS2 Exercice 1

1) Lorsqu'on additionne deux nombres binaires A et B de 1 bit, il y a quatre cas:



Donc la table de vérité:

A	B	S	RS
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

2) la retenue entrante est connectée à la retenue sortante de la colonne située juste à droite lorsqu'on fait des additions de nombres A et B comportant plusieurs bits.

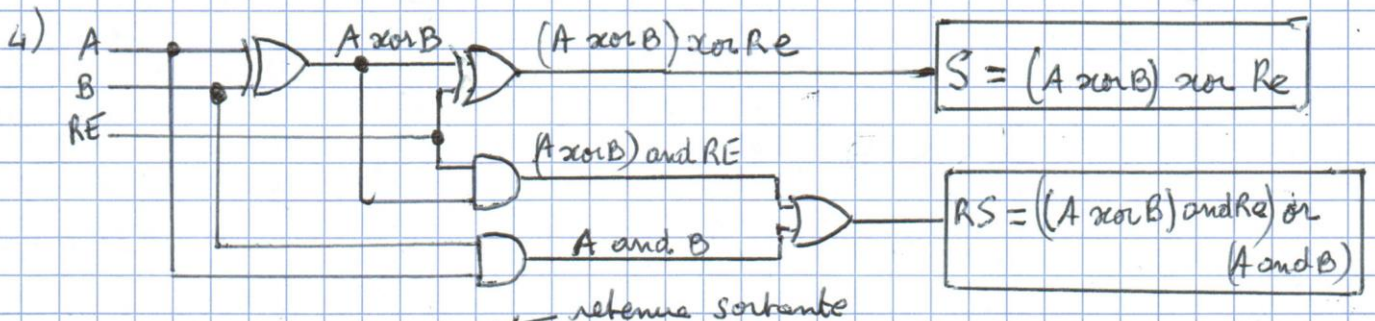
3) les tables de vérité sont faites en observant les huit cas donnés dans l'énoncé à la page 2.

a)

A	B	RE	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	0	0
0	0	1	1
0	1	1	0
1	0	1	0
1	1	1	1

b)

A	B	RE	RS
0	0	0	0
0	1	0	0
1	0	0	0
1	1	0	1
0	0	1	0
0	1	1	1
1	0	1	1
1	1	1	1



5) Si on additionne $A = 1011$ et $B = 0111$

$$\begin{array}{r} 1111 \\ 1011 \\ + 0111 \\ \hline 0010 \end{array}$$

Somme. c'est à dire $(11)_2 + (7)_2 = (18)_{10}$

le resultat de l'addition est $\overset{RS}{1} \overset{S}{0010}$

(retenue sortante)

6) a) $A = (1100)_2$ donc $A = (C)_{16}$ L'officier A affiche C

b) $B = (1111)_2$ donc $B = (F)_{16}$ L'officier B affiche F

c) $(1100)_2 + (1111)_2 = (11011)_2$ donc $S = (1011)_2$
L'officier affiche B

d) On a vu que la somme $(1100)_2 + (1111)_2$ donne un cinquième bit de poids fort qui vaut 1 donc RS vaut 1

7) $B = 0010$ donc pour avoir $-B$, Leo doit prendre le complément à 2 de 0010 .

Il y a deux étapes:

• Inverser tous les bits: 1101

• Ajouter 1

$$\begin{array}{r} 1101 \\ + 1 \\ \hline 1110 \end{array}$$

Ainsi -2 est codé par 1110 .

Si Leo ne fait que mettre quatre portes not alors il ne fait que la première étape (inverser tous les bits).

Pour obtenir 1, il doit entrer 1 en mettant la retenue entrante à 1.

Vérification:

$$\begin{array}{r} (7)_{10} \\ + (-2)_{10} \\ \hline \end{array} \quad \begin{array}{r} 1111 \\ 0111 \\ + 1110 \\ \hline 10101 \end{array}$$

bit "perdu"

le résultat est $(0101)_2 = (5)_{10}$

$$7 - 2 = 5$$

• le problème de Leo, c'est qu'il a oublié la 2^e étape (ajouter 1).
Il a seulement inversé tous les bits de $(2)_{10} = (0010)_2$
ce qui donne:

$$\begin{array}{r} (7)_{10} \\ + (-2)_{10} \\ \hline \end{array} \quad \begin{array}{r} 1111 \\ 0111 \\ + 1101 \\ \hline 10100 \end{array}$$

perdu

le résultat est $(0100)_2 = (4)_{10}$
ce qui est faux.

Exercice 2

1) Elle peut remarquer que "ë" s'est transformé en "Ã©" et que "ä" s'est transformé en "Ã".

2) a) l'octet 20 est une écriture hexadécimale.

$(2)_{16}$ s'écrit en binaire $(0010)_2$

$(0)_{16}$ s'écrit en binaire $(0000)_2$

Donc les huit bits de l'octet qui servent à coder l'espace sont 0010 0000

b) "blanc à droite" est codé

62 6C 61 6E 63 20 C3 A0 20 64 72 6F 69 74 65 29

c) le caractère 'à' est codé C3 A0

3) L'octet C3 s'affiche par Ã si c'est la table de caractères ISO 8859-9 qui est utilisée et l'octet A0 correspond à un espace.

4) le fichier Python n'a pas été encodé selon la table ISO 8859-9 car le caractère 'à' n'est pas codé sur un octet, mais sur deux.

5) le fichier Python qui a été téléchargé est encodé en UTF-8.

6) Conseil à Jeanne:

Écrire en première ligne de son script Python
-*- coding: utf-8 -*-

Ainsi son environnement de développement Python saura que le fichier est encodé selon la table de caractères UTF-8 et affichera donc correctement les accents.