

TD 1 : Bases, Conversions et Arithmétique

Exercice 1 Ordres de grandeur

- $2^0 =$
- $2^1 =$
- $2^2 =$
- $2^3 =$
- $2^4 =$
- $2^5 =$
- $2^6 =$
- $2^7 =$
- $2^8 =$
- $2^9 =$
- $2^{10} =$
- 1 Ko = ...o
- 1 Mo =o
- 1 Go =o

Exercice 2 Pour une base quelconque $b \geq 2$ de chiffres c_i :

1. comment s'écrivent les nombres 0 et 1 ?
2. quel est l'intervalle pour i ?
3. comment s'écrit le nombre b ? Et ses puissances b^n ?
4. quel est le plus petit nombre à 3 chiffres ? Et le plus grand ?

Exercice 3 Conversions entre base 2 et 16

1. Convertir $(101011110000)_2$ en base 16.
2. Convertir $(BAFFE)_{16}$ en base 2.
3. Expliquez le mécanisme de conversion d'une base b à une base b^n et réciproquement.

Exercice 4 Conversions

1. $(0)_2 = (\quad)_{10}$
2. $(1)_2 = (\quad)_{10}$
3. $(10)_2 = (\quad)_{10}$
4. $(0)_{16} = (\quad)_{10}$
5. $(1)_{16} = (\quad)_{10}$
6. $(12)_2 = (\quad)_{10}$
7. $(BAFFE)_{16} = (\quad)_{10}$
8. $(BAG)_{16} = (\quad)_{10}$
9. $(1346)_{10} = (\quad)_2$
10. $(11011011)_2 = (\quad)_{10}$
11. $(3982)_{10} = (\quad)_{16}$
12. $(A4EF)_{16} = (\quad)_{10}$
13. $(DE3C)_{16} = (\quad)_2$
14. $(10111010110)_2 = (\quad)_{16}$

Exercice 5 Représentation des entiers naturels

1. Donner les intervalles de codage d'un entier naturel sur :
 - (a) 8 bits
 - (b) 16 bits
 - (c) 32 bits
2. Dans la suite de l'exercice, on code sur 8 bits. Effectuer les opérations suivantes.
 - (a) $10010110 + 00101101$
 - (b) $11001110 - 10100111$
 - (c) 00001011×00010110
 - (d) $00111011 / 00000101$
 - (e) $10010110 + 01101101$
 - (f) 01001001×00000110
 - (g) $00001001 \times (00000011 / 00000101)$
 - (h) $(00001001 \times 00000011) / 00000101$

TD 2 : Représentation d'un nombre entier relatif

Exercice 1 Représentation des entiers relatifs : signe/module

1. Donner les intervalles de codage sur 8 bits et sur 16 bits. Dans la suite de l'exercice, on travaille sur 8 bits.
2. Coder les entiers $(+97)_{10}$ et $(-34)_{10}$.
3. Décoder $(00110101)_2$ et $(10110101)_2$.
4. Effectuer les additions :
 - (a) $0110\ 1011 + 1011\ 1101$
 - (b) $0110\ 1011 + 1111\ 0000$
 - (c) $1001\ 0110 + 1011\ 1011$.

Exercice 2 Représentation des entiers relatifs : complément à 2

1. Donner les intervalles de codage sur 8 bits et sur 16 bits. Dans la suite de l'exercice, on travaille sur 8 bits.
2. Coder les entiers $(+97)_{10}$ et $(-34)_{10}$.
3. Décoder $(00110101)_2$ et $(10110101)_2$.
4. Calculer le codage en complément à deux de $(10000000)_2 = (-128)_{10}$. Que se passe-t-il ?
5. Effectuer les additions :
 - (a) $0110\ 1011 + 1011\ 1101$
 - (b) $1001\ 0110 + 1111\ 1011$
 - (c) $0110\ 1111 + 0001\ 1001$
 - (d) $1000\ 0010 + 1010\ 1011$.
6. En distinguant trois cas, montrer que l'addition binaire fonctionne avec des nombres représentés en complément à deux (sauf dépassement de capacité!).

Exercice 3 Récapitulatif

En mémoire, on trouve la séquence suivante 01101010 10010101. Indiquez la valeur de cette mémoire dans les cas où cette suite est constituée de :

1. deux nombres entiers naturels codés en Décimal Codé Binaire (BCD) ;
2. deux nombres entiers naturels codés chacun sur 8 bits ;
3. deux nombres entiers relatifs en représentation signe/module sur 8 bits ;
4. deux nombres entiers relatifs représentés sur 8 bits en complément à 2.

TD 3 : Représentation des réels et des caractères

Exercice 1 Représentation d'une partie fractionnaire

1. Coder sur 8 bits les parties fractionnaires suivantes :
 - $(0.578125)_{10}$
 - $(0.85)_{10}$
2. Décoder :
 - $(0.10110000)_2$
 - $(0.11011001)_2$

Exercice 2 Représentation des réels en norme IEEE 754

Rappeler l'étendue des valeurs des nombres normalisés en simple et double précision.

Exercice 3 Codage en IEEE 754 Coder les réels suivants (représentés en base 10) en simple précision :

- 40
- -0.078125
- 13.625
- -87.375
- 0
- NaN
- $+\infty$

Exercice 4 Décodage de réels exprimés en norme IEEE 754 Décoder les réels suivants, donnés en simple précision :

- $41FE8000_{16}$
- $3EA80000_{16}$
- $C5E00000_{16}$
- 00380000_{16}

Exercice 5 Additions Effectuer les additions suivantes :

- $C0880000_{16} + 41680000_{16}$
- $815A6000_{16} + 015B1000_{16}$

ASCII								
(.) ₁₀	(.) ₁₆	char	(.) ₁₀	(.) ₁₆	char	(.) ₁₀	(.) ₁₆	char
32	0x20	<SPACE>	64	0x40	@	96	0x60	`
33	0x21	!	65	0x41	A	97	0x61	a
34	0x22	"	66	0x42	B	98	0x62	b
35	0x23	#	67	0x43	C	99	0x63	c
36	0x24	\$	68	0x44	D	100	0x64	d
37	0x25	%	69	0x45	E	101	0x65	e
38	0x26	&	70	0x46	F	102	0x66	f
39	0x27	'	71	0x47	G	103	0x67	g
40	0x28	(72	0x48	H	104	0x68	h
41	0x29)	73	0x49	I	105	0x69	i
42	0x2A	*	74	0x4A	J	106	0x6A	j
43	0x2B	+	75	0x4B	K	107	0x6B	k
44	0x2C	,	76	0x4C	L	108	0x6C	l
45	0x2D	-	77	0x4D	M	109	0x6D	m
46	0x2E	.	78	0x4E	N	110	0x6E	n
47	0x2F	/	79	0x4F	O	111	0x6F	o
48	0x30	0	80	0x50	P	112	0x70	p
49	0x31	1	81	0x51	Q	113	0x71	q
50	0x32	2	82	0x52	R	114	0x72	r
51	0x33	3	83	0x53	S	115	0x73	s
52	0x34	4	84	0x54	T	116	0x74	t
53	0x35	5	85	0x55	U	117	0x75	u
54	0x36	6	86	0x56	V	118	0x76	v
55	0x37	7	87	0x57	W	119	0x77	w
56	0x38	8	88	0x58	X	120	0x78	x
57	0x39	9	89	0x59	Y	121	0x79	y
58	0x3A	:	90	0x5A	Z	122	0x7A	z
59	0x3B	;	91	0x5B	[123	0x7B	}
60	0x3C	<	92	0x5C	\	124	0x7C	
61	0x3D	=	93	0x5D]	125	0x7D	{
62	0x3E	>	94	0x5E	^	126	0x7E	~
63	0x3F	?	95	0x5F	_	127	0x7F	

Exercice 6 Représentation des caractères

1. En utilisant le code ASCII, écrire votre nom et votre prénom sans oublier de mettre les initiales en majuscule.
2. En utilisant le code ASCII, décoder la phrase suivante : 76 39 97 114 99 104 105 44 32 99 39 101 115 116 32 102 97 99 105 108 101 46
3. En utilisant le code ASCII, décoder la phrase suivante donnée en hexadécimal : 4A 27 41 49 20 54 52 4F 55 56 45 20 21

TD 4 : Circuits logiques

Exercice 1 Circuit mystérieux

Le circuit de la Figure 1, réalisé en technologie CMOS, possède 3 entrées, x_1 , x_2 et s , et une sortie y .

1. Quels sont les transistors n MOS et les p MOS dans ce circuit ?
2. Déterminer la fonction de transfert du circuit, c'est-à-dire l'expression de y en fonction des entrées. Pour cela, déterminer si les transistors sont passants ou bloquants en fonction de la valeur de s .

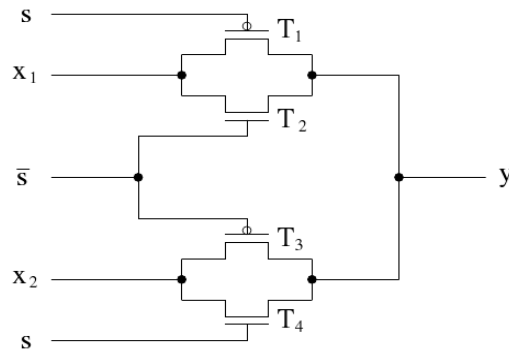


FIG. 1 – Circuit mystérieux en technologie CMOS.

3. Quel est l'intérêt de connecter les entrées à la sortie à travers à la fois des transistors n MOS et p MOS qui ont le même état (bloquant ou passant suivant s) au même moment ?

Exercice 2 Conception d'un additionneur-soustracteur n bits

L'objectif de cet exercice est la réalisation d'un additionneur-soustracteur n bits pour des entiers naturels.

1. La première étape consiste à réaliser un additionneur 1 bit prenant en entrée les deux bits a et b à additionner. La seconde est d'utiliser ces circuits pour former un additionneur n bits.
Quelles doivent être les entrées/sorties de l'additionneur 1 bit pour pouvoir l'utiliser dans la construction de l'additionneur n bits ? Représenter l'additionneur 1 bit sous forme de boîte en indiquant ses entrées/sorties.
2. Donner un schéma logique pour l'additionneur n bits en utilisant l'additionneur 1 bit comme circuit de base.
3. Quelles informations peut-on obtenir en sortie de l'additionneur n bits ?
4. En utilisant les propriétés du codage des entiers relatifs en complément à deux et le mode de calcul du complément à deux, proposer une opération alternative à la soustraction.
5. Proposer un circuit pouvant réaliser la soustraction sur n bits à partir d'un additionneur n bits.
6. Proposer un circuit utilisant un unique additionneur n bits qui peut réaliser soit l'addition soit la soustraction sur n bits suivant la valeur d'une nouvelle entrée : OP . Par exemple si OP vaut 0 le circuit réalise l'addition de A et B et si OP vaut 1, c'est la soustraction $A - B$ qui est effectuée.

7. Une fois que les valeurs d'entrée a , b et c de l'additionneur 1 bit sont toutes établies, un délai de t_{a_1} unités de temps est nécessaire avant l'établissement des valeurs de sorties r et c correctes. Quel délai t_{a_n} est nécessaire avant que toutes les sorties de l'additionneur n bits soient établies? (NB : on considère que toutes les entrées a_i et b_i sont établies simultanément)
8. Même question pour le délai maximum t_{as} de l'additionneur-soustracteur.

Exercice 3 Portes NAND

Toutes les expressions booléennes peuvent se représenter à l'aide des seuls opérateurs NOT, AND et OR. Cependant dans certaines technologies, comme la technologie CMOS, il est impossible de réaliser directement les portes AND et OR. Il est par contre possible d'obtenir des portes NAND et NOT.

1. Montrer comment on peut obtenir les trois opérateurs NOT, AND et OR uniquement à partir de l'opérateur NAND.

x	y	AND	OR	NAND
0	0	0	0	1
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0

TAB. 1 – Table de valeur des opérateurs binaires AND, OR et NAND.

2. Donner le schéma logique des trois opérateurs de base en utilisant uniquement des portes NAND.