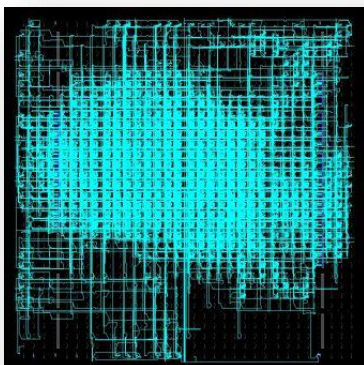
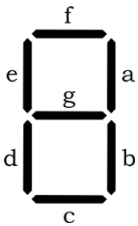
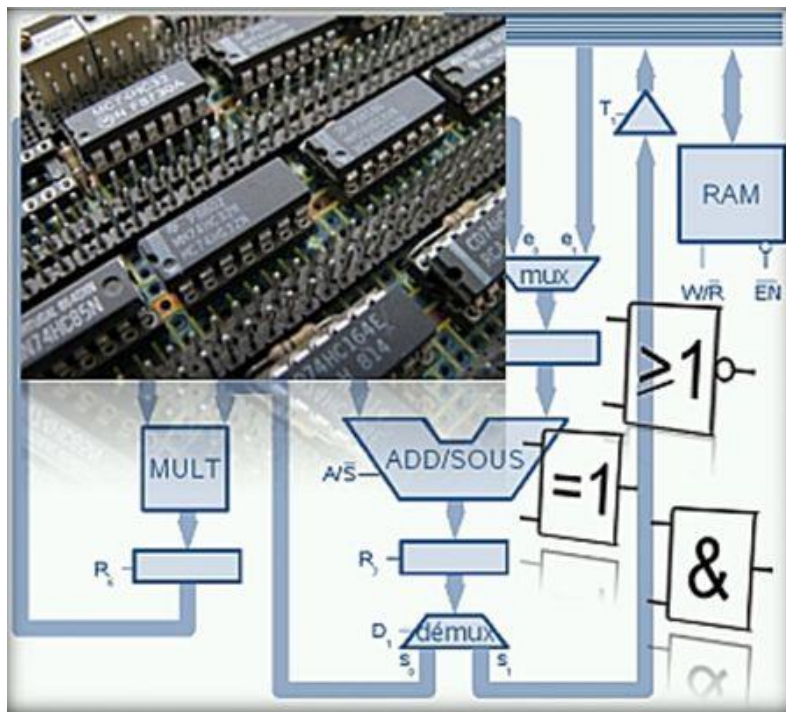


Electronique numérique

Circuits et architectures logiques

2023-2024 – 1A ELEC GPSE





ENSICAEN
Digital Electronic

I. Introduction

Contexte de l'électronique numérique

M. Denoual 1

1

ENSICAEN
Digital Electronic

I. Introduction

Objectif de ce cours

s'adresse aux élèves en GPSE 1A et INFO 1A

en lien direct avec les cours d'architecture et technologie des ordinateurs et de **processeurs pour l'embarqué** (Hugo Descoubes (GPSE) et Bogdan Cretu (INFO)) et de **VHDL et synthèse logique sur FPGA** (GPSE SATE 2A)

Connaître la structure et le fonctionnement interne d'un cœur de calculateur

Compétences à acquérir :

- 1- codage et représentation des nombres,
- 2- synthèse de circuits logiques combinatoires,
- 3- connaissance des structures de logique séquentielle.

2

2



ENSICAEN Digital Electronic

I. Introduction

Contexte de l'électronique numérique

Distinction électronique analogique/numérique

En électronique analogique, une information est par une grandeur électrique -tension ou courant- qui est connue de manière . Cette information peut prendre une , dont la connaissance est limitée par le des composants et des fonctions analogiques utilisées.

Signal analogique

Signal analogique

3

ENSICAEN Digital Electronic

I. Introduction

Contexte de l'électronique numérique

Distinction électronique analogique/numérique

- Les domaines d'application de l'électronique analogique ont tendance à se réduire au profit de l'électronique numérique.
- Les 3 domaines restants d'applications de l'électronique analogique sont:
 - 1. le **conditionnement** de signaux issus de **capteurs** (amplification et filtrage). L'objectif est alors de dégager un signal électrique souvent faible du bruit de façon à pouvoir l'utiliser ou le traiter.

Remarques :

- *Toutes les grandeurs physiques sont analogiques*
- *Dans certains capteurs récents, on peut trouver des électroniques de numérisation directe sur puce qui repoussent encore le domaine de l'analogique.*

4



ENSICAEN Digital Electronic

I. Introduction

Contexte de l'électronique numérique

Distinction électronique analogique/numérique

- 2. communication radio-fréquence (RF) ou électronique **haute-fréquence**
- 3. électronique de **puissance**

Remarques :

- On trouve également de l'électronique analogique dans les systèmes automatiques pour l'asservissement ou la régulation. Mais la tendance générale en automatique est le développement à partir d'électronique numérique. Cette tendance est soutenue par les capacités croissant sans cesse des composants numériques (vitesse, capacité de calcul) et par la diminution de leur coût.
- L'électronique analogique peut présenter un intérêt pour des dispositifs peu complexes et faible coût.

5

5

ENSICAEN Digital Electronic

I. Introduction

Contexte de l'électronique numérique

Distinction électronique analogique/numérique

- En électronique numérique ou électronique digitale (français), les signaux présentent des **sauts** dans le temps et
- L'information est représentée au moyen d'un nombre limité de valeurs grâce à une ou plusieurs grandeurs binaires appelées bits.

Amplitude vs Temps

Legend: signal (red line), instant d'échantillonnage (blue dots), signal discrétisé (blue step function)

T_e : période d'échantillonnage

Amplitude vs Temps

Legend: signal (red line), signal discrétisé (blue dotted line), signal quantifié (magenta step function)

q : quantum, pas de quantification

6



Contexte de l'électronique numérique
Distinction électronique analogique/numérique

continue dans le temps \Rightarrow instant d'échantillonnage
infinité de valeur \Rightarrow nb de valeurs fini

discretisation
quantification

Req 1 Remarque :
contrainte sur les instants d'échantillonnages (théorème de Shannon-Nyquist)
Il faut que la **fréquence d'échantillonnage** (inverse de l'intervalle entre deux instants d'échantillonnage) soit supérieure à 2 fois la fréquence maximum du signal analogique.

7

Contexte de l'électronique numérique
Distinction électronique analogique/numérique

représentation linéaire par défaut

code binaire 2^2-1
11
10
01
00

code pour 1V
01

valeur analogique [V]

1,7V

erreur de quantification +q

Exemple : signal à 1,7V Erreur 0,7V

Représentation utilisée

représentation linéaire centrée

code binaire 2^2-1
11
10
01
00

code pour 2V
10

valeur analogique [V]

1,7V

erreur de quantification +q/2

Erreur -0,3V

$q/2 = 0,5V$
 $3q/2$
 q
 $q/2$

8



ENSICAEN Digital Electronic

I. Introduction

Contexte de l'électronique numérique

Signaux numériques

bits : **binary digit** (chiffre binaire)

Ces bits prennent les valeurs :

TRUTH TABLE

I ₁	I ₂	O ₁
L	L	L
H	L	H
L	H	H
H	H	L

Notes

1. H = HIGH state (the more positive voltage)
L = LOW state (the less positive voltage)

table de vérité dans une datasheet de composant logique discret

1/4 of device shown
 $J = \overline{A+B}$
 Logical "1" = HIGH
 Logical "0" = LOW
 All inputs protected by standard CMOS protection circuit.

relation entre valeur logique et niveau électrique dans une datasheet de composant logique discret

valeur	contexte
vrai/faux	logique
0/1	logique booléenne
0V/5V	aspect électrique
L/H	datasheet
5V/0V	logique négative
12V/-12V	transmission série
...	

9

9

ENSICAEN Digital Electronic


I. Introduction

Contexte de l'électronique numérique

Signaux numériques

Exemples de signaux numériques

- bouton poussoir (touche d'un clavier):
BP=1, bouton enfoncé, interrupteur fermé
BP=0, bouton relâché, interrupteur ouvert
- lampe : L=1, lampe allumée / L=0, lampe éteinte
- pixel d'une image (télévision numérique)
 - diffusion : bits



Smart Watch et autres petits écran format image RGB565 16 bits

? **Question :** combien de couleur peut-on coder en format diffusion ?

10

10



ENSI CAEN
Digital Electronic

I. Introduction

Contexte de l'électronique numérique

Distinction électronique analogique/numérique

? **Questions :**

- Combien de valeurs peut-on représenter avec N de bits ?
- Quelle est l'erreur de quantification maximum en fonction du quantum ?
- On suppose une gamme de tension de [0-4V] et 2 bits de représentation, combien valent le quantum (pas de quantification) et l'erreur maximum de représentation ?
- Application numérique pour N=10 bits et une gamme de tension [0-5V]

11


11

ENSI CAEN
Digital Electronic

I. Introduction

Contexte de l'électronique numérique

Signaux numériques



Exemples de signaux numériques (suite)

- CD audio, MP3 :
 - bits
 - fréquence échantillonnage:

? **Question :** taille d'une image FHD 1920x1080 pixels format BMP ?

? **Question :** quelle taille mémoire pour 1 heure de musique, stéréo non compressée ?

- Signal numérisé sortant d'un capteur : 8 à 12 bits. Exemple, capteur de température DS18P20 : 10 bits
- Code ASCII : bits (caractères)
(American Standard Code for Information Interchange)

12

12



Contexte de l'électronique numérique

ENSICAEN Digital Electronic

I. Introduction

Signaux numériques : code ASCII

Decimal	Hex	ASCII	Decimal	Hex	ASCII	Decimal	Hex	ASCII	Decimal	Hex	ASCII
0	0	NUL	32	20		64	40	@	96	60	`
1	1	SOH	33	21	!	65	41	A	97	61	a
2	2	STX	34	22	"	66	42	B	98	62	b
3	3	ETX	35	23	#	67	43	C	99	63	c
4	4	EOT	36	24	\$	68	44	D	100	64	d
5	5	ENO	37	25	%	69	45	E	101	65	e
6	6	ACK	38	26	&	70	46	F	102	66	f
7	7	BEL	39	27	'	71	47	G	103	67	g
8	8	BS	40	28	(72	48	H	104	68	h
9	9	HT	41	29)	73	49	I	105	69	i
10	A	LF	42	2A	^	74	4A	J	106	6A	j
11	B	VT	43	2B	+	75	4B	K	107	6B	k
12	C	FF	44	2C	,	76	4C	L	108	6C	l
13	D	CR	45	2D	-	77	4D	M	109	6D	m
14	E	SOH	46	2E	.	78	4E	N	110	6E	n
15	F	SI	47	2F	/	79	4F	O	111	6F	o
16	10	DLE	48	30	0	80	50	P	112	70	p
17	11	DC1	49	31	1	81	51	Q	113	71	q
18	12	DC2	50	32	2	82	52	R	114	72	r
19	13	DC3	51	33	3	83	53	S	115	73	s
20	14	DC4	52	34	4	84	54	T	116	74	t
21	15	NAK	53	35	5	85	55	U	117	75	u
22	16	SYN	54	36	6	86	56	V	118	76	v
23	17	ETB	55	37	7	87	57	W	119	77	w
24	18	CAN	56	38	8	88	58	X	120	78	x
25	19	EM	57	39	9	89	59	Y	121	79	y
26	1A	SUB	58	3A	:	90	5A	Z	122	7A	z
27	1B	ESC	59	3B	;	91	5B	[123	7B	{
28	1C	FS	60	3C	<	92	5C	\	124	7C	
29	1D	GS	61	3D	=	93	5D]	125	7D	}
30	1E	RS	62	3E	>	94	5E	^	126	7E	~
31	1F	US	63	3F	?	95	5F	_	127	7F	□

Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
128	80	Ç	160	A0	à	192	C0	À	224	E0	α
129	81	ü	161	A1	á	193	C1	Á	225	E1	β
130	82	ë	162	A2	â	194	C2	Â	226	E2	γ
131	83	ä	163	A3	ã	195	C3	Ã	227	E3	δ
132	84	å	164	A4	ä	196	C4	Ä	228	E4	ε
133	85	å	165	A5	å	197	C5	Å	229	E5	σ
134	86	æ	166	A6	æ	198	C6	Æ	230	E6	μ
135	87	ç	167	A7	ç	199	C7	Ç	231	E7	Υ
136	88	è	168	A8	è	200	C8	È	232	E8	ϕ
137	89	é	169	A9	é	201	C9	É	233	E9	ω
138	8A	ê	170	AA	ê	202	CA	Ê	234	EA	α
139	8B	ë	171	AB	ë	203	CB	Ë	235	EB	β
140	8C	ì	172	AC	ì	204	CC	Ì	236	EC	γ
141	8D	í	173	AD	í	205	CD	Í	237	ED	δ
142	8E	î	174	AE	î	206	CE	Î	238	EE	ε
143	8F	ï	175	AF	ï	207	CF	Ï	239	EF	σ
144	90	ÿ	176	BO	ÿ	208	DO	ÿ	240	FO	μ
145	91	æ	177	B1	æ	209	D1	æ	241	F1	Υ
146	92	ç	178	B2	ç	210	D2	ç	242	F2	ϕ
147	93	è	179	B3	è	211	D3	è	243	F3	ω
148	94	é	180	B4	é	212	D4	é	244	F4	α
149	95	ê	181	B5	ê	213	D5	ê	245	F5	β
150	96	ë	182	B6	ë	214	D6	ë	246	F6	γ
151	97	ì	183	B7	ì	215	D7	ì	247	F7	δ
152	98	í	184	B8	í	216	D8	í	248	F8	ε
153	99	î	185	B9	î	217	D9	î	249	F9	σ
154	9A	ï	186	BA	ï	218	DA	ï	250	FA	μ
155	9B	ÿ	187	BB	ÿ	219	DB	ÿ	251	FB	Υ
156	9C	æ	188	BC	æ	220	DC	æ	252	FC	ϕ
157	9D	ç	189	BD	ç	221	DD	ç	253	FD	ω
158	9E	è	190	BE	è	222	DE	è	254	FE	α
159	9F	é	191	BF	é	223	DF	é	255	FF	β

représenté sur 8 bits (256 caractères). Exemple : 6B_h représente le caractère « k »

13

Contexte de l'électronique numérique

ENSICAEN Digital Electronic

I. Introduction

électronique numérique

- L'électronique numérique est particulièrement **adaptée aux calculs** et aux tests.
- Ces calculs et tests sont la base des algorithmes de traitement par exemple en **traitement numérique du signal (TNS)** pour le filtrage, pour le **cryptage** ou la commande de procédé.

Remq 1 *Remarque : en électronique analogique, la seule façon de faire des calculs est d'utiliser des amplificateurs opérationnels (d'où le nom).*

- Les signaux numériques présentent en outre la possibilité d'être **compressés** (codage MP3, zip) et **stockés** longtemps.
- D'un point de vue transmission et également stockage et traitement, les informations numériques présentent une meilleure **immunité au bruit**. Cette caractéristique est très utile pour les communications pour réduire le taux d'erreur et peut être accrue par un **codage** (code correcteur d'erreur).

14



ENSICAEN
Digital Electronic

I. Introduction


Contexte de l'électronique numérique

électronique numérique

Remarque : moins d'information ici qu'à gauche Rmq !

Signal analogique

- La confidentialité des transmissions peut être obtenue par **cryptage** (on peut également crypter en analogique mais il y a moins de possibilités).
- Champs des informations créées par l'homme** : image virtuelle, jeux, musique numérique, films 3D...



15


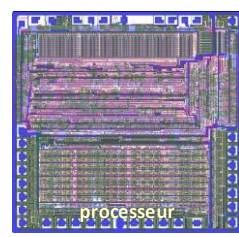
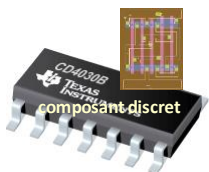
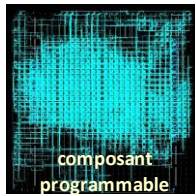
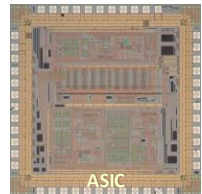
15

ENSICAEN
Digital Electronic

I. Introduction

Contexte de l'électronique numérique

électronique numérique

- Mémoire** : stockage 
- Microprocesseurs** : exécution d'instruction de programme 
- Composants logiques** : fonctions spécifiques (ASIC, FPGA, PLD)
 - 
 - 
 - 

16



ENSICAEN Digital Electronic

I. Introduction

Contexte de l'électronique numérique

électronique numérique

- **Mémoire** : stockage
- **Microprocesseurs** : exécution d'instruction de programme
- **Composants logiques** : fonctions spécifiques

17

ENSICAEN Digital Electronic

I. Introduction

A l'issue de cette première partie

- Qu'est-ce qu'un signal discrétisé ?
- Qu'est-ce qu'un signal quantifié ?
- Quels intérêts à utiliser des signaux numériques plutôt qu'analogiques ?
- Quelles limites à l'utilisation des signaux numériques ?

18



ENSICAEN
Digital Electronic

II. Numération et Codage

Représentation des grandeurs au cœur des architectures

M. Denoual

1

1

ENSICAEN
Digital Electronic

II. Numération et codage

1. Numération, bases classiques

Les bases utilisées en électronique numérique sont:

- base 2 (binaire)
- base 8 (octal)
- base 10 (décimal)
- base 16 (hexadécimal)

Remq **Remarques:**
-D'un point de vue physique (électrique), les signaux sont toujours représentés par des niveaux haut ou bas (2 états). Les autres bases permettent de **compacter l'écriture** (base 8 et 16) pour faciliter la lecture et l'écriture des valeurs et ainsi limiter les erreurs.

$1101011100110111 \rightarrow D737$

- La base 10 facilite l'interfaçage homme-machine (affichage-saisie).

2

2



ENSICAEN
Digital Electronic

II. Numération et codage

1. Numération, bases classiques

Remq **Remarques:**

- Toute base s'écrit dans son propre système de numération
- Un décalage à gauche : un nombre par sa base
- Un décalage à droite : un nombre par sa base

? **Question :** combien faut-il de chiffres n dans une base B pour représenter un nombre décimal de N chiffre ?

Application numérique: base 2 et $N=2$.

3

3

ENSICAEN
Digital Electronic

II. Numération et codage

2. Conversions, passage d'une base à l'autre

a. D'une base B quelconque à la base décimale

On attribue le poids de chaque chiffre.

Soit un nombre $N = (a_n a_{n-1} \dots a_1 a_0, a_{-1} a_{-2} \dots)_B$
alors la valeur en base 10 correspondant est :

$$(N)_{10} = a_n B^n + a_{n-1} B^{n-1} + \dots + a_1 B^1 + a_0 B^0 + a_{-1} B^{-1} + a_{-2} B^{-2} + \dots$$

Exemples :

4

4



ENSICAEN Digital Electronic

II. Numération et codage

2. Conversions, passage d'une base à l'autre

b. De la base décimale vers une base quelconque

Divisions successives par la base

On traite différemment les parties entières et fractionnaires.
Soit un nombre N et sa représentation en base B sous la forme :

$$(a_n a_{n-1} \dots a_1 a_0, a_{-1} a_{-2} \dots)_B$$

$\underbrace{\hspace{100px}}_{E[N]}$
 $\underbrace{\hspace{100px}}_{\text{Frac}[N]}$

La partie entière, E[N], peut se mettre sous la forme :

$$E[N] = a_n B^n + a_{n-1} B^{n-1} + \dots + a_2 B^2 + a_1 B^1 + a_0 B^0$$

en factorisant par la base B, on obtient :

$$E[N] = B(a_n B^{n-1} + a_{n-1} B^{n-2} + \dots + a_2 B^1 + a_1 B^0) + a_0$$

en répétant l'opération :

$$E[N] = B(B(a_n B^{n-1} + a_{n-1} B^{n-2} + \dots + a_2 B^0) + a_1) + a_0$$

1^{ère} division donne a_0

2^è division donne a_1

3^è division donne a_2

...

On fait ainsi apparaître que les a_k peuvent être obtenus par division successive de E[N] par B.

5

5

ENSICAEN Digital Electronic

II. Numération et codage

2. Conversions, passage d'une base à l'autre

b. De la base décimale vers une base quelconque

Multiplications successives par la base

(cont.)

La partie fractionnaire, Frac[N], peut se mettre sous la forme :

$$\text{Frac}[N] = a_{-1} B^{-1} + a_{-2} B^{-2} + \dots + a_{-m} B^{-m}$$

en multipliant par la base, on obtient :

$$B \cdot \text{Frac}[N] = a_{-1} + a_{-2} B^{-1} + \dots + a_{-m} B^{-m+1}$$

en répétant l'opération :

$$B^2 \cdot \text{Frac}[N] = B a_{-1} + (a_{-2} + \dots + a_{-m} B^{-m+2})$$

1^{ère} multiplication donne a_{-1}

2^è division donne a_{-2}

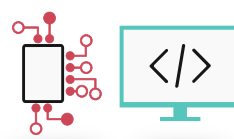
3^è division donne a_{-3}

...

On fait ainsi apparaître que les a_k peuvent être obtenus par multiplications successives par B.

6

6



ENSICAEN
Digital Electronic

II. Numération et codage

2. Conversions, passage d'une base à l'autre

c. Conversion entre base 2^k -aire

- de la base 2 vers base 2^k avec $k>1$
regroupement des bits par k
base 2 vers base 8 : regroupement par 3 des bits
base 2 vers base 16 : regroupement par 4 des bits
- des bases 2^k avec $k>1$ vers la base 2
remplacement des chiffres de la base 2^k par leurs expressions dans la base 2
- d'une base 2^k ($k>1$) vers une autre base 2^k ($k>1$)
le plus simple est de passer par l'intermédiaire de la base 2 (développement puis regroupement)

exemples

Base 2 : 010011100011

Base 8 : 2 3 4 3

Base 2 : 010011100011

Base 16 : 4 E 3

7

7

ENSICAEN
Digital Electronic

II. Numération et codage

3. Codages à partir des deux chiffres 0 et 1

1010 ?

D'un point de vue de l'électronique numérique, les données réellement manipulées sont des **niveaux haut ou bas**, c'est-à-dire des 1 ou des 0 binaires.

A partir de ces deux chiffres, il existe plusieurs possibilités pour le codage des grandeurs ou signaux. Ces possibilités sont choisies en fonction de leur adéquation à une fonction (affichage, capteur position), de leur précision ou dynamique de représentation ou bien de leur simplicité d'implémentation (réduction de l'électronique associée).

Par la suite, on distinguera les codes pondérés des codes non pondérés.

Définitions :

- un code est **pondéré** si chaque chiffre est affecté d'un poids.
- un code est **continu** si les mots binaires consécutifs sont adjacents, c'est-à-dire qu'ils ne diffèrent que d'un bit.
- un code est **cyclique** s'il est continu et si le premier et le dernier mots sont adjacents.

8

8



ENSICAEN Digital Electronic

II. Numération et codage

3. Codages à partir des deux chiffres 0 et 1

a. Codes classiques couramment utilisés

Code Gray

Le code Gray ou binaire réfléchi est cyclique par construction.

Valeur	code Gray
0	0 0 0
1	0 0 1
2	0 1 1
3	0 1 0
4	1 1 0
5	1 1 1
6	1 0 1
7	1 0 0

binaire classique

code Gray

Utilisé dans les codeurs de position optique
(une erreur de lecture -bit flou- ne conduit à une incertitude que sur deux valeurs successives)
Utilisé dans certaines machines d'état pour réduire la taille des blocs logiques

9

9

ENSICAEN Digital Electronic

II. Numération et codage

3. Codages à partir des deux chiffres 0 et 1

a. Codes classiques couramment utilisés

Application code Gray

binaire

Gray

10

10



ENSICAEN Digital Electronic

II. Numération et codage

3. Codages à partir des deux chiffres 0 et 1

a. Codes classiques couramment utilisés (cont.)

Code 7 segments

Codage utilisé pour les afficheurs 7 segments

chiffre à afficher	code 7 segments						
	a	b	c	d	e	f	g
0	1	1	1	1	1	1	0
1	0	1	1	0	0	0	0
2
3
4
5
6
7	1	1	1	0	0	0	0
8
9

11

ENSICAEN Digital Electronic

II. Numération et codage

3. Codages à partir des deux chiffres 0 et 1

a. Codes classiques couramment utilisés (cont.)

Code BCD (Binary Coded Decimal)

Aussi appelé code DCB (Décimal codé binaire)
 Dans ce codage, chaque chiffre d'un nombre décimal est codé individuellement en binaire.

Exemple : 2 3 1

Remarque : on ne peut pas faire de calcul facilement avec ce code.

12



3. Codages à partir des deux chiffres 0 et 1

II. Numération et codage

ENSICAEN Digital Electronic

b. Autres codes

N ₁₀	Codes DCB pondérés								Codes DCB non pondérés			
	DCBN 8421	biquinaire 10 86420	biquinaire 50 43210	7421	MBQ 5 421	5211	Aiken 2421	753 -6	X3S -3	décimal Johnson	Gray DCB	2 parmi 5
0	0000	01 00001	01 00001	0000	0 000	0000	0000	000 0	0011	00000	0000	11000
1	0001	10 00001	01 00010	0001	0 001	0001	0001	100 1	0100	00001	0010	00011
2	0010	01 00010	01 00100	0010	0 010	0100	0010	011 1	0101	00011	0110	00101
3	0011	10 00010	01 01000	0011	0 011	0101	0011	001 0	0110	00111	0111	00110
4	0100	01 00100	01 10000	0100	0 100	0111	1010	101 1	0111	01111	0101	01001
5	0101	10 00100	10 00001	0101	1 000	1000	0101	010 0	1000	11111	1101	01010
6	0110	01 01000	10 00010	0110	1 001	1010	1100	110 1	1001	11110	1111	01100
7	0111	10 01000	10 00100	1000	1 010	1011	1101	100 0	1010	11100	1110	10001
8	1000	01 10000	10 01000	1001	1 011	1110	1110	011 0	1011	11000	1010	10010
9	1001	10 10000	10 10000	1010	1 100	1111	1111	111 1	1100	10000	1000	10100
auto complémentaires						oui	oui	oui	oui			
cyclique										oui	oui	
détection d'erreur		oui	oui									oui

13

13

3. Codages à partir des deux chiffres 0 et 1

II. Numération et codage

ENSICAEN Digital Electronic

c. Codage pour le calcul

Problématique du codage pour le calcul

Caractéristiques requises de la représentation des nombres pour le calcul

- **bijectif** (chaque valeur n'a qu'une représentation)
- **extensible** (CAN 8 bits -> processeur 16 bits par instruction élémentaire)
- **simplicité des opérateurs de calculs (ALU)**
- **opérateurs cascables** (2 additionneurs 8 bits pour former un additionneur 16 bits)
- **même représentation des nombres positifs en signé ou non**

14

14



ENSI CAEN
Digital Electronic

II. Numération et codage

3. Codages à partir des deux chiffres 0 et 1

c. Codage pour le calcul
Représentations classiques

binaire naturel (base 2)

sur N bits

$$x = b_{N-1} 2^{N-1} + b_{N-2} 2^{N-2} + \dots + b_2 2^2 + b_1 2^1 + b_0 2^0$$

valeurs
nombre entre [0;]

Propriétés : bijectif, extensible, opérateurs cascadables

15

15

ENSI CAEN
Digital Electronic

II. Numération et codage

3. Codages à partir des deux chiffres 0 et 1

c. Codage pour le calcul
Représentations classiques

Ce code n'est pas utilisé en pratique.

binaire signé

sur N bits

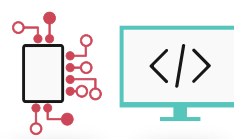
$$x = (-1)^{b_{N-1}} (b_{N-2} 2^{N-2} + \dots + b_2 2^2 + b_1 2^1 + b_0 2^0)$$

valeurs
nombre entre [;]

Propriétés :
Le bit de signe n'a pas de pondération. Ce code est difficilement extensible, pas cascable et on a deux représentations pour 0.
Ce code n'est pas utilisé en pratique.

16

16



3. Codages à partir des deux chiffres 0 et 1

II. Numération et codage

c. Codage pour le calcul
Binaire décalé

binaire décalé
Ce type de codage vient des convertisseurs analogique/numérique (CAN), à cause de l'alimentation dissymétrique des CANs.

Amplitude

Temps

Uniquement positif

V_{DD}

CAN

Code binaire

Temps

code binaire $2^N - 1$

valeur

représentation binaire

code binaire $2^N - 1$

valeur

$-V_{max}/2$ 0 $V_{max}/2$

représentation binaire décalée

Représente une valeur négative

17

3. Codages à partir des deux chiffres 0 et 1

II. Numération et codage

c. Codage pour le calcul
Binaire décalé (cont.)

sur N bits

Décalage : moitié de l'espace de représentation **pour info**

$$x = (b_{N-1} 2^{N-1} + \dots + b_2 2^2 + b_1 2^1 + b_0 2^0) - 2^{N-1}$$

$$x = (b_{N-1} - 1) 2^{N-1} + \dots + b_2 2^2 + b_1 2^1 + b_0 2^0$$

$$x = -b_{N-1} 2^{N-1} + b_{N-2} 2^{N-2} + \dots + b_2 2^2 + b_1 2^1 + b_0 2^0$$

valeurs nombre entre [;]

Exemples:

code binaire $2^N - 1$

valeur

$-V_{max}/2$ 0 $V_{max}/2$

représentation binaire décalée

18



ENSICAEN Digital Electronic

II. Numération et codage

3. Codages à partir des deux chiffres 0 et 1

c. Codage pour le calcul
Binaire décalé (cont.)

Règle d'extension (pour passer d'une représentation sur N bits à une représentation sur N+1 bits) :
remarque préalable : $2^N - 2^{N-1} = 2^{N-1}$

$$b_{N-1} \ b_{N-2} \ \dots \ b_2 \ b_1 \ b_0$$

$$\downarrow$$

$$b_{N-1} \ b_{N-1} \ b_{N-2} \ \dots \ b_2 \ b_1 \ b_0$$

pour info

Propriétés :
code bijectif,
extensible
➡ mais pas la même représentation des nombres positifs en signé ou non.

19

19

ENSICAEN Digital Electronic

II. Numération et codage

3. Codages à partir des deux chiffres 0 et 1

c. Codage pour le calcul
Complément à 2

c'est **LE** codage utilisé dans les circuits et architectures logiques pour le calcul avec des nombres signés

Complément à 2
sur N bits

$$x = -b_{N-1} 2^{N-1} + b_{N-2} 2^{N-2} + \dots + b_2 2^2 + b_1 2^1 + b_0 2^0$$

valeurs
nombre entre [;]

Exemples :

Propriétés : code bijectif, extensible, même représentation des nombres positifs en signé ou non, conduit à des structures d'opérateurs simples et cascadables.

important !

Règle d'extension (pour passer d'une représentation sur N bits à une représentation sur N+1 bits) :
simplement par **recopie du bit de poids fort**.

nombre positif commence par : nombre négatif commence par :

à retenir !

20

20



ENSICAEN Digital Electronic

II. Numération et codage

3. Codages à partir des deux chiffres 0 et 1

c. Codage pour le calcul
Complément à 2 (cont.)

Rmn **Remarque** : tant que les bits de poids forts successifs sont identiques, on peut les supprimer pour ne garder de le dernier.

Pour obtenir l'expression de la valeur négative d'un nombre, on procède en trois étapes comme suit :

à retenir !

- 1- expression de la valeur absolue du nombre (binaire naturel)
- 2- complément à 1 de chacun des bits
- 3- ajout d'un bit à '1' tout à droite

21

ENSICAEN Digital Electronic

II. Numération et codage

3. Codages à partir des deux chiffres 0 et 1

c. Codage pour le calcul
Complément à 2 (cont.)

Exemples (1/2)

22



ENSI CAEN Digital Electronic

II. Numération et codage

3. Codages à partir des deux chiffres 0 et 1

c. Codage pour le calcul Complément à 2 (cont.)

Exemples (2/2) -1563

1- expression de la valeur absolue du nombre (binaire naturel)

$$\begin{array}{r}
 1563 \quad | \quad 16 \\
 11 \quad | \quad 97 \quad | \quad 16 \\
 \quad \quad | \quad 1 \quad | \quad 6
 \end{array}$$

1563 → 61B
 ↓
 0110 0001 1011

2- complément à 1 de chacun des bits

$$\begin{array}{r}
 0110 \ 0001 \ 1011 \\
 1001 \ 1110 \ 0100
 \end{array}$$

3- ajout d'un bit à '1' tout à droite

$$\begin{array}{r}
 1001 \ 1110 \ 0100 \\
 1001 \ 1110 \ 0101
 \end{array}$$

Calcul de l'erreur de quantification. Vérification qu'elle est inférieure à l'erreur max (q/2)

23

23

ENSI CAEN Digital Electronic

II. Numération et codage

3. Codages à partir des deux chiffres 0 et 1

c. Codage pour le calcul Complément à 2 (cont.)

Démonstration de cette construction en trois étapes

24

24



ENSICAEN
Digital Electronic

II. Numération et codage

3. Codages à partir des deux chiffres 0 et 1

Résumé représentation entiers non-signés/signés

Exemple d'un composant convertisseur numérique analogique (CNA, DAC Digital Analog Converter)

Data Sheet AD5724/AD5734/AD5754

Ideal Output Voltage to Input Code Relationship—AD5754

Table 7. Bipolar Output, Offset Binary Coding

Digital Input				Analog Output		
MSB	LSB	±5 V Output Range	±10 V Output Range	±10.8 V Output Range		
1111	1111	+2 × REFIN × (32,767/32,768)	+4 × REFIN × (32,767/32,768)	+4.32 × REFIN × (32,767/32,768)		
1111	1111	+2 × REFIN × (32,766/32,768)	+4 × REFIN × (32,766/32,768)	+4.32 × REFIN × (32,766/32,768)		
...		
1000	0000	+2 × REFIN × (1/32,768)	+4 × REFIN × (1/32,768)	+4.32 × REFIN × (1/32,768)		
1000	0000	0 V	0 V	0 V		
0111	1111	-2 × REFIN × (1/32,768)	-4 × REFIN × (1/32,768)	-4.32 × REFIN × (1/32,768)		
...		
0000	0000	-2 × REFIN × (32,767/32,768)	-4 × REFIN × (32,767/32,768)	-4.32 × REFIN × (32,767/32,768)		
0000	0000	-2 × REFIN × (32,768/32,768)	-4 × REFIN × (32,768/32,768)	-4.32 × REFIN × (32,768/32,768)		

Table 8. Bipolar Output, Twos Complement Coding

Digital Input				Analog Output		
MSB	LSB	±5 V Output Range	±10 V Output Range	±10.8 V Output Range		
0111	1111	+2 × REFIN × (32,767/32,768)	+4 × REFIN × (32,767/32,768)	+4.32 × REFIN × (32,767/32,768)		
0111	1111	+2 × REFIN × (32,766/32,768)	+4 × REFIN × (32,766/32,768)	+4.32 × REFIN × (32,766/32,768)		
...		
0000	0000	+2 × REFIN × (1/32,768)	+4 × REFIN × (1/32,768)	+4.32 × REFIN × (1/32,768)		
0000	0000	0 V	0 V	0 V		
1111	1111	-2 × REFIN × (1/32,768)	-4 × REFIN × (1/32,768)	-4.32 × REFIN × (1/32,768)		
...		
1000	0000	-2 × REFIN × (32,767/32,768)	-4 × REFIN × (32,767/32,768)	-4.32 × REFIN × (32,767/32,768)		
1000	0000	-2 × REFIN × (32,768/32,768)	-4 × REFIN × (32,768/32,768)	-4.32 × REFIN × (32,768/32,768)		

Table 9. Unipolar Output, Straight Binary Coding

Digital Input				Analog Output		
MSB	LSB	+5 V Output Range	+10 V Output Range	+10.8 V Output Range		
1111	1111	+2 × REFIN × (65,535/65,536)	+4 × REFIN × (65,535/65,536)	+4.32 × REFIN × (65,535/65,536)		
1111	1111	+2 × REFIN × (65,534/65,536)	+4 × REFIN × (65,534/65,536)	+4.32 × REFIN × (65,534/65,536)		
...		
1000	0000	+2 × REFIN × (32,769/65,536)	+4 × REFIN × (32,769/65,536)	+4.32 × REFIN × (32,769/65,536)		
1000	0000	+2 × REFIN × (32,768/65,536)	+4 × REFIN × (32,768/65,536)	+4.32 × REFIN × (32,768/65,536)		
0111	1111	+2 × REFIN × (32,767/65,536)	+4 × REFIN × (32,767/65,536)	+4.32 × REFIN × (32,767/65,536)		
...		
0000	0000	+2 × REFIN × (1/65,536)	+4 × REFIN × (1/65,536)	+4.32 × REFIN × (1/65,536)		
0000	0000	0 V	0 V	0 V		

25

ENSICAEN
Digital Electronic

II. Numération et codage

3. Codages à partir des deux chiffres 0 et 1

c. Codage pour le calcul

Remarques générales

Req **Remarques : décalages arithmétiques et logiques**

- un décalage à droite correspond à une division par 2
exemple :
- un décalage à gauche correspond à une multiplication par 2
exemple :

⚠ Attention au débordement

- Différence entre LSR (Logic Shift Right) décalage à droite logique (introduction d'un 0) et ASR (Arithmetic Shift Right) (introduction du bit de signe)
- Décalages spéciaux sur certains processeurs/microcontrôleurs par exemple PIC18 : décalage circulaire (RRNCF (rotate right f))

26

26



ENSICAEN Digital Electronic

II. Numération et codage

3. Codages à partir des deux chiffres 0 et 1

d. Représentations des nombres non entiers

Représentation à virgule fixe en $Q_{m,k}$

$$b_{N-1} b_{N-2} \dots b_k , b_{k-1} \dots b_2 b_1 b_0$$

$$(2^{N-1} 2^{N-2} \dots 2^k 2^{k-1} \dots 2^2 2^1 2^0) \cdot 2^{-k} \text{ k décalages}$$

$$2^{N-1-k} 2^{N-2-k} \dots 2^0 2^{-1} \dots 2^{-k} 2^{1-k} 2^{-k}$$

Alors la valeur représentée appartient à l'intervalle :

Le quantum (pas de quantification) est :

Remarque : la virgule n'a pas d'existence matérielle. Les opérateurs arithmétiques traitent les nombres à virgule fixe comme des entiers (suite de 0 et de 1).

27

27

ENSICAEN Digital Electronic

II. Numération et codage

3. Codages à partir des deux chiffres 0 et 1

d. Représentations des nombres non entiers

Représentation à virgule fixe en $Q_{m,k}$ et complément à 2 nombres non entiers signés

On n'ajoute pas 1

Exemple : -12,25

12,25 : 0000 1100 0100

Complément à 1

1111 0011 1011

+1

1111 0100 1011

1 0100 1011

-16 +4 +0,5 +0,125 +0,0625

-11,3125

FAUX

10 0 1 1 0 1

1 1 11 10011 1 0 1

0 1111 111 11110 0

111 0001 111 1011 01

0 1 1110 01110110

0110000 10011111

01111111 0 110001

100 11 000111

1 0111 111 110 1

10 11 11 11 01

00 11 11 11 11

10 11 0 10 10 10 10

1 0 100 11 10 10 11 11

0 11 10 11 10 11 11 10

000000000000

001011 01011 0001 0000

11001 011 0 111 0111

1100 001 1 000

000 10 0 100

1 1 001

On ajoute un bit à 1 à droite de la représentation

Représentation $Q_{8,4}$

12,25 : 0000 1100 0100

Complément à 1

1111 0011 1011

+1 à droite

1111 0011 1100

1 0011 1100

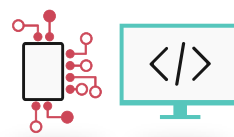
-16 +2 +1 +0,5 +0,25

-12,25 ✓

La virgule n'a pas d'existence matérielle. Dans la machine, il n'y a que des 0 et des 1.

28

28



ENSICAEN
Digital Electronic

II. Numération et codage

3. Codages à partir des deux chiffres 0 et 1

d. Représentations des nombres non entiers

Obtenir la représentation à virgule fixe en $Q_{m,k}$

<u>Méthode 1 :</u>		<u>Méthode 2 :</u>
Si négatif : représentation de la valeur absolue puis complément à 2.		Multiplication du nombre à représenter par 2^k .
Pour les nbs positifs et la valeur absolue, obtention de la partie entière par divisions successives, obtention de la partie fractionnaire par multiplications successives.		Représentation de l'entier obtenu sur $m+k$ bits.
		k décalages à droite de la représentation.

29

29

ENSICAEN
Digital Electronic

II. Numération et codage

3. Codages à partir des deux chiffres 0 et 1

d. Représentations des nombres non entiers

Exemples

30

30



ENSI CAEN Digital Electronic
II. Numération et codage
3. Codages à partir des deux chiffres 0 et 1
d. Représentations des nombres non entiers

Exemples conversion en base 2 (**16 bits** virgule fixe **Q12,4**) de -621,17.

nombre négatif, conversion en 3 étapes

Méthode 1

- 1- On représente la valeur absolue du nombre soit 621,17

partie entière div. succ.	partie fractionnaire mult. succ.
621	0,17 0,72
13 38	x 16 x 16
6 2	2,72 11,52
	0,17 → 0,2B _h
	arrondi 0,17 → 0,3 _h
	0,17 → 0,0011
- 2- Complément à « 1 » de chacun des bits

0010 0110 1101 0011
1101 1001 0010 1100
- 3- On ajoute un bit à « 1 » tout à droite

1101 1001 0010 1101	+1
---------------------	----

L'erreur est introduite lors de l'arrondi : 0,0011 → 0,1875

$erreur = |0,1875 - 0,17| = 0,0175 < erreur_{max} = q/2 = 2^{-5} = 0,03125$

31

31

ENSI CAEN Digital Electronic
II. Numération et codage
3. Codages à partir des deux chiffres 0 et 1
d. Représentations des nombres non entiers

Exemples conversion en base 2 (**16 bits** virgule fixe **Q12,4**) de -621,17.

Méthode 2

Multiplication du nombre à représenter par 2^k ici 2^4

$-621,17 * 2^4 = -9938,72$

Représentation de l'entier obtenu sur $m+k$ bits, ici 16 bits. L'entier est négatif, on commence par représenter la valeur absolue du nombre.

9938,72 → arrondi 9939	9939	16	9939 → 26D3
Erreur de conversion introduite ici (arrondi)	3	621	0010 0110 1101 0011
		13 38	1101 1001 0010 1100
		6 2	1101 1001 0010 1101
			Compl à 1
			+1

k décalages à droite de la représentation.

1101 1001 0010 1101

Partie entière ↑ Partie fractionnaire

32

32



ENSICAEN Digital Electronic

II. Numération et codage

3. Codages à partir des deux chiffres 0 et 1

d. Représentations des nombres non entiers

Représentation à virgule flottante

Une valeur est représentée par une mantisse et un exposant.

$$x = \pm m.b^e$$

b : base; m : mantisse, e : exposant.

? *Question : pourquoi virgule flottante ?*

Par la suite, on ne considère que l'étude en base 2 (b=2).

$$x = \pm m.2^e$$

34

34

ENSICAEN Digital Electronic

II. Numération et codage

3. Codages à partir des deux chiffres 0 et 1

d. Représentations des nombres non entiers

On adapte le zoom

Représentation à virgule flottante (cont.)

Avec cette écriture, on peut avoir plusieurs solutions pour représenter une même valeur. Alors, on fixe un **contrainte sur la mantisse** de façon à avoir une écriture unique pour la valeur.

$$1 \leq |m| < 2$$

La mantisse est représentée en virgule fixe.
On utilise un bit pour le signe.

Finalement, le nombre en virgule flottante s'écrit :

$$x = (-1)^S . 1, F . 2^E$$

Le 1 de la partie entière n'a pas besoin d'être stocké. Il s'agit d'un 1 implicite.

35

35



ENSI CAEN
 Digital Electronic
 II. Numération et codage
 3. Codages à partir des deux chiffres 0 et 1
 d. Représentations des nombres non entiers

Représentation à virgule flottante (cont.)

Norme IEEE 754 (1985)

C'est la représentation standard pour l'arithmétique en virgule flottante, utilisée par la plupart des systèmes pour améliorer la compatibilité entre les architectures et la portabilité des applications.

$$X = (-1)^S \cdot 1, F \cdot 2^{E-B}$$

(B biais ou décalage
1+mantisse
(E≠0))

Exposant biaisé : l'exposant est représenté par un entier auquel on soustrait un biais B.

Exemple de la représentation IEEE 754 sur 32 bits (*simple precision*) :

exposant E sur 8 bits, F sur 23 bits : $X = (-1)^S \cdot 1, F \cdot 2^{E-127}$

S E₇E₆E₅E₄E₃E₂E₁E₀ F₂₂F₂₁F₂₀...F₂F₁F₀

Représentation IEEE 754 sur 64 bits (*double precision*), exposant E sur 11 bits.

37

37

ENSI CAEN
 Digital Electronic
 II. Numération et codage
 3. Codages à partir des deux chiffres 0 et 1
 d. Représentations des nombres non entiers

Représentation à virgule flottante (cont.)

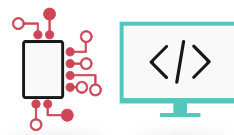
Norme IEEE 754 (1985) exemple

-65324892,4562

1- mise en forme : signe puissance de 2 et mantisse

38

38



ENSICAEN Digital Electronic | II. Numération et codage | 3. Codages à partir des deux chiffres 0 et 1 | d. Représentations des nombres non entiers

Représentation à virgule flottante (cont.)

Norme IEEE 754 (1985) exemple

A = -40 468,48

1- mise en forme : signe puissance de 2 et mantisse $A = (-1)^1 \times 1,235 \times 2^{15}$

E-127 = 15 → E = 142 → 1000 1110

0,235	0,76	0,16	0,56	0,96	0,36
x 16	x 16	x 16	x 16	x 16	x 16
3,76	12,16	2,56	8,96	15,36	5,76
0011	1100	0010	1000	1111	0101

S E7 E0 F22 F0

A 1 1000 1110 0011 1100 0010 1000 1111 011

39

39

ENSICAEN Digital Electronic | II. Numération et codage | 3. Codages à partir des deux chiffres 0 et 1 | d. Représentations des nombres non entiers

Représentation à virgule flottante (cont.)

Norme IEEE 754 (cont.)

Nombres dénormalisés : en plus de la représentation normalisée avec le 1 implicite, la norme définit une représentation alternative lorsque l'exposant vaut 0. Pour représenter les valeurs très proches de 0.

$$x = (-1)^S \cdot 0, F \cdot 2^{-126}$$

Exposant = 0
Mantisse ≠ 0

Cette dénormalisation permet de représenter le 0 lorsque F=0.

Valeur la plus petite dénormalisée : $0,0000\dots 01 \cdot 2^{1-8}$

Continuité avec normalisé :
 dénormalisé : $0,11111\dots 111 \cdot 2^{1-8}$
 normalisé : $1,00000\dots 0000 \cdot 2^{1-8}$

$$\left(\begin{array}{l} (-1)^S \cdot 0,11\dots 11 \cdot 2^{-126} \\ (-1)^S \cdot 1,00\dots 00 \cdot 2^{1-127} \end{array} \right)$$

Représentations spéciales :

infini : représenté par un exposant maximum et F=0. Le bit de signe distingue +∞ et -∞.

NaN : Not a Number, correspondant au résultat de calcul incorrect. Représenté par un exposant maximal et F non nul.

40

40



ENSICAEN Digital Electronic

II. Numération et codage

3. Codages à partir des deux chiffres 0 et 1

d. Représentations des nombres non entiers

Représentation à virgule flottante (cont.)

Norme IEEE 754 (cont.)

Exemples: sur 32 bits valeur	S	E ₇ E ₆ E ₅ E ₄ E ₃ E ₂ E ₁ E ₀	F ₂₂ F ₂₁ F ₂₀ ...F ₂ F ₁ F ₀
0	0	0000 0000	000.....000
-0	1	0000 0000	000.....000
∞	0	1111 1111	000.....000
-∞	1	1111 1111	000.....000
NaN	0	1111 1111	00010....0
6,5	0	1000 0001	1010....00
7,347.10 ⁻³⁹	0	0000 0000	1010....00

41

41

ENSICAEN Digital Electronic

II. Numération et codage

3. Codages à partir des deux chiffres 0 et 1

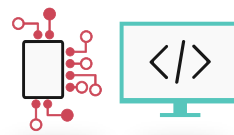
d. Représentations des nombres non entiers

Représentation à virgule flottante (cont.), nombre de décimales significatives avec 32 bits pour π ?

$a[]$: arrondi à l'unité

42

42



ENSICAEN Digital Electronic
 II. Numération et codage
 3. Codages à partir des deux chiffres 0 et 1
 d. Représentations des nombres non entiers

Représentation à virgule flottante (cont.)

Représentation IEEE 754 sur 64 bits (*double precision*), exposant E sur 11 bits, correspond à combien de chiffres significatifs derrière la virgule en décimal ?

43

ENSICAEN Digital Electronic
 II. Numération et codage
 3. Codages à partir des deux chiffres 0 et 1
 d. Représentations des nombres non entiers

Comparaison virgule fixe/ virgule flottante

Des critères de comparaisons sont la **dynamique de représentation**, le **bruit de quantification** et la **complexité des opérateurs arithmétiques**.

La dynamique se définit comme le rapport des valeurs maximum et minimum de la représentation.

$$\text{dynamique} = \frac{\text{valeur max}}{\text{valeur min}}$$

capacité à représenter des valeurs très petites (résolution) et très grandes (étendue)

On l'utilise souvent en dB par : $D_N(\text{dB}) = 20 \log \left(\frac{\text{valeur max}}{\text{valeur min}} \right)$

44



3. Codages à partir des deux chiffres 0 et 1

II. Numération et codage

d. Représentations des nombres non entiers

Comparaison virgule fixe/ virgule flottante (cont.)

The graph plots dynamic range in dB against the number of bits (Nb bits). The x-axis ranges from 0 to 30 bits, and the y-axis ranges from 0 to 1600 dB. A blue line represents fixed-point representation, showing a linear increase. A red line represents floating-point representation, showing a much steeper increase that reaches approximately 1500 dB at 30 bits.

virgule fixe 16 bits : $D_N(\text{dB}) = 6.02 \cdot (16-1) = 90.3 \text{ dB}$

virgule flottante 16 bits, E sur 4 bits : $D_N(\text{dB}) = 6.02 \cdot (2^4-1) = 90.3 \text{ dB}$

45

45

3. Codages à partir des deux chiffres 0 et 1

II. Numération et codage

d. Représentations des nombres non entiers

Comparaison virgule fixe/ virgule flottante (cont.)

Critère : bruit de quantification (erreur de représentation relative).

virgule fixe

The top graph shows absolute error (e) as a high-frequency sawtooth wave between 0 and 0.1. The bottom graph shows relative error (e/x) as a curve that starts at 0.1 and decays towards 0 as x increases.

virgule flottante

The top graph shows absolute error (e) as a sawtooth wave with a constant amplitude of approximately 0.05. The bottom graph shows relative error (e/x) as a sawtooth wave with a constant amplitude of approximately 0.05.

46

46



ENSI CAEN
Digital Electronic

II. Numération et codage

3. Codages à partir des deux chiffres 0 et 1

e. Remarque déclaration de format de nombre en C

Remarques : déclaration de format de nombre en C

Une déclaration de type de nombre sert à définir :

- la taille (multiple de 8 bits).
- le domaine qui peut varier suivant la nature du type (signé, non signé).

type	taille	domaine
unsigned char	8 bits	0 à 255
char	8 bits	-128 à 127
unsigned int (OS 32 bits)	32 bits	0 à 4 294 967 295
short int	16 bits	-32768 à 32767
int (OS 32 bits)	32 bits	-2 147 483 648 à 2 147 483 647
long	32 bits	-2 147 483 648 à 2 147 483 647
float	32 bits	$3,4 \cdot 10^{-38}$ à $3,4 \cdot 10^{38}$
double	64 bits	$1,7 \cdot 10^{-308}$ à $1,7 \cdot 10^{308}$

47



ENSI CAEN
Digital Electronic

II. Numération et codage

3. Codages à partir des deux chiffres 0 et 1

e. Remarque déclaration de format de nombre en C

Remarques : déclaration de format de nombre en C

<pre>uint_8 nb = 10001111; itoa(nb, nbText, 10); PutString(nbText);</pre>		<pre>char nb = 10001111; itoa(nb, nbText, 10); PutString(nbText);</pre>
 143		 -113

Même contenu mémoire, interprétation différente

48



ENSICAEN Digital Electronic

II. Numération et codage

4. De l'importance du choix de codage

En fonction de la vulnérabilité de l'application : le code doit permettre la représentation des grandeurs manipulées sans débordement (ou avec signalisation du débordement) et avoir une résolution suffisante pour que le bruit de quantification (dit autrement l'arrondi de représentation) n'est pas d'impact sur l'application.

Le compromis est à trouver entre le nombre de bits utilisés pour la représentation et la dynamique associée et la taille/consommation/coût de l'unité de calcul associée.

$3+2,19 =$

49

49

ENSICAEN Digital Electronic

II. Numération et codage

4. De l'importance du choix de codage

Exemple d'effet d'arrondi :
système réel : incrément de 1/10 toutes 100 ms.

en théorie :

temps [s]	valeur sur 8 bit	valeur sur 24 bits
0	0	0
0,1	0,09375	0,1
0,2	0,1875	0,2
...
100h	360000	337500
...
1 semaine	604800	567000
		604799,42

en réalité :
 $1/10 \Rightarrow$ base 2 : 0.000110011001100110011...
sur 8 bits : 0.0001100 soit 0,09375
 erreur de $6,25 \times 10^{-3}$ à chaque ajout
 au bout de 100h : 6h15 de décalage
sur 24 bits :
 erreur de $9,5367 \times 10^{-8}$ à chaque ajout
 au bout de 100h : 0,34s de décalage
 au bout d'une semaine : 0,576782 s

```

>>> 0.1+0.1 == 0.2
True
>>> 0.1+0.1+0.1 == 0.3
False
>>>
    
```

50

50



ENSICAEN Digital Electronic

II. Numération et codage


4. De l'importance du choix de codage

Appendix II

Effect of Extended Run Time on Patriot Operation

Hours	Seconds	Calculated Time (Seconds)	Inaccuracy (Seconds)	Approximate Shift In Range Gate (Meters)
0	0	0	0	0
1	3600	3599.9966	.0034	7
8	28800	28799.9725	.0275	55
20 (a)	72000	71999.9313	.0687	137
48	172800	172799.8352	.1648	330
72	259200	259199.7528	.2472	494
100 (b)	360000	359999.6667	.3333	687

a. Continuous operation exceeding about 20 hours--target outside range gate
b. Alpha Battery ran continuously for about 100 hours



Page 15 -- GAO/IMTEC-92-26 Patriot Missile Software Problem

erreur de distance = erreur de temps × vitesse
~2000 m/s

51

51

ENSICAEN Digital Electronic

II. Numération et codage

4. De l'importance du choix de codage

Exemple de sous capacité de représentation :

Ariane 501 Inquiry Board report

- The internal SRI software exception was caused during execution of a data conversion from 64-bit floating point to 16-bit signed integer value. The floating point number which was converted had a value greater than what could be represented by a 16-bit signed integer. This resulted in an Operand Error. The data conversion instructions (in Ada code) were not protected from causing an Operand Error, although other conversions of comparable variables in the same place in the code were protected.
- The error occurred in a part of the software that only performs alignment of the strap-down inertial platform. This software module computes meaningful results only before lift-off. As soon as the launcher lifts off, this function serves no purpose.
- The alignment function is operative for 50 seconds after starting of the Flight Mode of the SRIs which occurs at $H_0 - 3$ seconds for Ariane 5. Consequently, when lift-off occurs, the function continues for approx. 40 seconds of flight. This time sequence is based on a requirement of Ariane 4 and is not required for Ariane 5.
- The Operand Error occurred due to an unexpected high value of an internal alignment function result called BH, Horizontal Bias, related to the horizontal velocity sensed by the platform. This value is calculated as an indicator for alignment precision over time.
- The value of BH was much higher than expected because the early part of the trajectory of Ariane 5 differs from that of Ariane 4 and results in considerably higher horizontal velocity values.

Ariane 501 Inquiry Board report page 4

coût 500 millions !

1996

ARIANE 5

Flight 501 Failure

valeur plus grande que la représentation possible sur 16 bits

accélération Ariane 5 5 fois plus grande que Ariane 4



52

52



ENSICAEN
Digital Electronic

II. Numération et codage

4. De l'importance du choix de codage

Quand va-t-on avoir un problème avec le format du temps sur ordinateur ?

Représentation du temps sur les ordinateurs (heure POSIX ou heure UNIX) :

32bits entier signé pour représenter le nombre de secondes depuis le 1^{er} janvier 1970 à minuit.


Maximum :

54

ENSICAEN
Digital Electronic

II. Numération et codage

A l'issue de cette deuxième partie (1/3)



- Comment sont représentées d'un point de vue électrique les valeurs numériques dans les composants numériques (circuits logiques, microprocesseurs, microcontrôleurs) ?
- Dans quels cas utilise-t-on la base 10 ?
- Pourquoi utilise-t-on la base 16 ?
- Qu'est-ce qu'un code pondéré ?

55

55



ENSI CAEN
Digital Electronic

II. Numération et codage

A l'issue de cette deuxième partie (2/3)

- Quelles caractéristiques doit présenter un codage utilisé pour le calcul ?
- Quel codage est utilisé en pratique dans les circuits numériques pour représenter les nombres positifs et négatifs ?
- Comment représente-t-on un nombre positif en complément à 2 ?
- Comment représente-t-on un nombre négatif en complément à 2 ?

56

56

ENSI CAEN
Digital Electronic

II. Numération et codage

A l'issue de cette deuxième partie (3/3)

- Comment est représentée physiquement la virgule du codage à virgule fixe au cœur des circuits numériques ?
- Pourquoi impose-t-on une contrainte à la valeur de la mantisse dans le cas du codage en virgule flottante ?
- Quels sont les avantages/inconvénients relatifs des codages en virgule fixe et virgule flottante ?

57

57



ENSI CAEN
Digital Electronic

III. Logique combinatoire

Algèbre de Boole et fonctions Booléennes, implémentation

M. Denoual

1

1

ENSI CAEN
Digital Electronic


III. Logique combinatoire

Back to basics


développée de longue date

de λόγος / λόγος signifiant
raison, langage et raisonnement

- Conceptuellement :
 - En philosophie (le vrai, le faux) : étude des règles formelles que doit respecter toute argumentation correcte. III^e siècle av. J.-C.
 - En mathématique : algèbre de Boole. XIX^e siècle.
- En pratique :
 - Hydraulique
 - Mécanique
 - Électrique (électricité/relais-interrupteurs diodes → transistors)



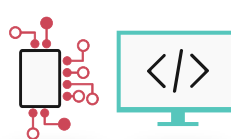
Steve Mould, Water computing



Pascaline, 1642

2

2



ENSICAEN Digital Electronic

III. Logique combinatoire

Back to basics

logique d'interrupteur

- Électricité : logique d'interrupteur

3

3

ENSICAEN Digital Electronic

III. Logique combinatoire

Algèbre de Boole : Opérations de base

- Algèbre de Boole définie sur E_2 constitué des éléments $\{0,1\}$
- Trois opérations de base :

complémentation		union + max OU			intersection · min ET		
a	\bar{a}	a	b	s	a	b	s
0	1	0	0	0	0	0	0
1	0	0	1	1	0	1	0
		1	0	1	1	0	0
		1	1	1	1	1	1

- 0 : élément neutre de l'union, élément minimum.
- 1 : élément neutre de l'intersection, élément maximum.

4

4



ENSI CAEN Digital Electronic

III. Logique combinatoire

Algèbre de Boole : Propriétés des opérations de base

$\forall a, b, c \in E_2$

- complémentation : $a.\bar{a} = 0$
 $a + \bar{a} = 1$
- commutativité : $a.b = b.a$
 $a + b = b + a$
- associativité : $(a.b).c = a.(b.c)$
 $(a + b) + c = a + (b + c)$
- distributivité : $a.(b + c) = a.b + a.c$
 $a + (b.c) = (a + b).(a + c)$

5

5

ENSI CAEN Digital Electronic

III. Logique combinatoire

Algèbre de Boole : Théorèmes associés à l'algèbre de Boole

$\forall a, b, c \in E_2$

- idempotence : $a.a = a$
 $a + a = a$
- absorption : $a + a.b = a$
 $a.(a + b) = a$
- involution : $\bar{\bar{a}} = a$
- Théorème de Morgan : $\overline{a + b} = \bar{a}.\bar{b}$; $\overline{a.b} = \bar{a} + \bar{b}$
- Théorème de Shannon : $a + b = a.b + a.\bar{b} + \bar{a}.b$
- Consensus de 1^{ière} espèce : $a + \bar{a}.b = a + b$
 $a.(a + b) = a.b$
- Consensus de 2nd espèce : $a.b + \bar{b}.c = a.b + \bar{b}.c + a.c$
 $(a + b).(b + c) = (a + b).(b + c).(a + c)$

à voir par vous même

Termes de Consensus

6

6



ENSICAEN Digital Electronic

III. Logique combinatoire

Algèbre de Boole : Méthode de réduction des expressions algébriques

à voir par vous même

- repérer les variables biformes,
- introduire le terme de Consensus de 2nd espèce s'il permet de réduire l'expression par absorption,
- absorption,
- retirer le terme de consensus.

Exemple :

$$F = f\bar{e}b + hgf\bar{e}dca + hg\bar{b}$$

ajout terme Consensus

$$F = f\bar{e}b + hgf\bar{e}dca + hg\bar{b} + hgf\bar{e}$$

Absorption
retrait Consensus

$$F = f\bar{e}b + hg\bar{b}$$

7

7

ENSICAEN Digital Electronic

III. Logique combinatoire

Algèbre de Boole : Fonctions booléennes

$E_2 \times E_2 \times \dots \times E_2$ dans E_2 .

entrées → fonction logique combinatoire → sortie(s)

entrées		sortie(s)
a	b	s
0	0	0
0	1	1
1	0	1
1	1	0

Table de vérité

Objectif :
Exprimer la ou les sorties en fonction des entrées (puis y associer des portes logiques – support matériel : synthèse logique)

Deux façons d'exprimer une fonction booléenne :

- la forme **somme de produit** ou disjonctive normale $\Sigma\Pi$
- la forme **produit de somme** ou conjonctive normale $\Pi\Sigma$

Idée : avoir l'expression la plus compacte possible

Moins de support physique, moins de consommation, plus rapide

8

8



ENSICAEN Digital Electronic

III. Logique combinatoire

Algèbre de Boole : Forme somme de produit $\Sigma\Pi$

- Première forme canonique
 - On associe une variable binaire m_i à chaque entrée de la table
 - *minterm*, terme produit

m_0 est associé à la ligne 0, actif/vrai (à 1) pour la ligne 0

$m_0 = 1$ si $a = 0$ et $b=0$ c'est-à-dire si $\bar{a} = 1$ et $\bar{b} = 1$

$m_0 = \bar{a} \cdot \bar{b}$

m_i	a	b	s
m_0	0	0	s_0
m_1	0	1	s_1
m_2	1	0	s_2
m_3	1	1	s_3

$m_0 = \bar{a} \cdot \bar{b}$; $m_1 = \bar{a} \cdot b$; $m_2 = a \cdot \bar{b}$; $m_3 = a \cdot b$

$$s = s_0 m_0 + s_1 m_1 + s_2 m_2 + s_3 m_3$$

9

9

ENSICAEN Digital Electronic

III. Logique combinatoire

Algèbre de Boole : Forme somme de produit $\Sigma\Pi$ (cont.)

- Simplification de l'expression
 - minterm pour lesquels la fonction vaut 1
- Dans le cas de la fonction XOR

m_i	a	b	s
m_0	0	0	0 (s_0)
m_1	0	1	1 (s_1)
m_2	1	0	1 (s_2)
m_3	1	1	0 (s_3)

$$s = m_1 + m_2$$

$$s = \bar{a} \cdot b + a \cdot \bar{b}$$

Remarque : réalisation simple avec des portes NAND

10

10



ENSICAEN Digital Electronic

III. Logique combinatoire

Algèbre de Boole : Forme produit de somme $\Pi\Sigma$

- Seconde forme canonique
 - On associe une variable binaire M_i à chaque entrée de la table
 - *maxterm*, terme somme
- Exemple pour la fonction XOR

a	b	M_0	M_1	M_2	M_3
0	0	0	1	1	1
0	1	1	0	1	1
1	0	1	1	0	1
1	1	1	1	1	0

M_i	a	b	s
M_0	0	0	0 (s_0)
M_1	0	1	1 (s_1)
M_2	1	0	1 (s_2)
M_3	1	1	0 (s_3)

$M_0 = a + b$; $M_1 = a + \bar{b}$; $M_2 = \bar{a} + b$; $M_3 = \bar{a} + \bar{b}$

$$s = (s_0 + M_0).(s_1 + M_1).(s_2 + M_2).(s_3 + M_3)$$

Maxterm constitué de l'union (OU) de toutes les variables d'entrée, non complémentées si leur valeur est 0, complémentées si leur valeur est 1.

à voir par vous même

11

11

ENSICAEN Digital Electronic

III. Logique combinatoire

Algèbre de Boole : Forme produit de somme $\Pi\Sigma$ (cont.)

- Simplification de l'expression
 - On conserve les maxterm pour lesquels la fonction vaut 0

a	b	M_0	M_1	M_2	M_3	M_0M_3	s
0	0	0	1	1	1	0	0
0	1	1	0	1	1	1	1
1	0	1	1	0	1	1	1
1	1	1	1	1	0	0	0

$s = 0$ si $M_0 = 0$ ou si $M_3 = 0$ c'est-à-dire si $M_0.M_3 = 0$
 $s = M_0.M_3$ $s = (a + b).(\bar{a} + \bar{b})$

Rmq **Remarque :** réalisation simple avec des portes NOR

à voir par vous même

12

12



ENSI CAEN
Digital Electronic

III. Logique combinatoire

Fonctions logiques

↓

Portes logiques

Aux fonctions logiques booléennes élémentaires sont associées des portes logiques.

Ces portes logiques ont une existence matérielle qui permet de réaliser concrètement les fonctions logiques.

Elles peuvent se présenter sous différentes formes:

- discrètes,
- intégrées,
- assemblées dans des blocs élémentaires,
- sous forme de matrice connectables,
- ...

Idée : passer d'une fonction sur papier ou écran de PC à un support physique

fonction	symbole CEI 60617 EN 60617:1999	(ancien symbole)	table de vérité															
NON, NOT, inversion			<table border="1"> <tr><td>a</td><td>s</td></tr> <tr><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td></tr> </table>	a	s	0	1	1	0									
a	s																	
0	1																	
1	0																	
ET, AND, intersection, min			<table border="1"> <tr><td>a</td><td>b</td><td>s</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </table>	a	b	s	0	0	0	0	1	0	1	0	0	1	1	1
a	b	s																
0	0	0																
0	1	0																
1	0	0																
1	1	1																
OU, OR, union, max			<table border="1"> <tr><td>a</td><td>b</td><td>s</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </table>	a	b	s	0	0	0	0	1	1	1	0	1	1	1	1
a	b	s																
0	0	0																
0	1	1																
1	0	1																
1	1	1																
NAND, non-ET			<table border="1"> <tr><td>a</td><td>b</td><td>s</td></tr> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </table>	a	b	s	0	0	1	0	1	1	1	0	1	1	1	0
a	b	s																
0	0	1																
0	1	1																
1	0	1																
1	1	0																
NOR, non-OU			<table border="1"> <tr><td>a</td><td>b</td><td>s</td></tr> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </table>	a	b	s	0	0	1	0	1	0	1	0	0	1	1	0
a	b	s																
0	0	1																
0	1	0																
1	0	0																
1	1	0																
XOR, ou-exclusif			<table border="1"> <tr><td>a</td><td>b</td><td>s</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </table>	a	b	s	0	0	0	0	1	1	1	0	1	1	1	0
a	b	s																
0	0	0																
0	1	1																
1	0	1																
1	1	0																

13

ENSI CAEN
Digital Electronic

III. Logique combinatoire

Fonctions logiques

↓

Portes logiques

Remarque 1 : en pratique, les équations de l'algèbre de Boole ne sont pas toujours vérifiées.

$s = \bar{e}$

Idée : différence entre idéal (fonction logique) et implémentation (réalisation physique)

Physiquement :

- Temps de propagation
- Temps d'établissement (montée, descente)
- Limites en courant (sortance)
- Consommation

Philips Semiconductors Product specification

Quadruple exclusive-OR gate HEF4030B gates

	V _{DD} V	SYMBOL	TYP.	MAX.	TYPICAL EXTRAPOLATION FORMULA
Propagation delays	5	t _{PVH}	85	175	57 ns + (0,55 ns/pF) C _L
	10		30	75	24 ns + (0,23 ns/pF) C _L
	15		20	55	22 ns + (0,16 ns/pF) C _L
LOW to HIGH	5	t _{PLH}	75	150	47 ns + (0,55 ns/pF) C _L
	10		30	65	19 ns + (0,23 ns/pF) C _L
	15		25	50	17 ns + (0,16 ns/pF) C _L
Output transition times	5	t _{rL}	60	120	10 ns + (1,0 ns/pF) C _L
	10		30	60	9 ns + (0,42 ns/pF) C _L
	15		20	40	6 ns + (0,28 ns/pF) C _L
LOW to HIGH	5	t _{rUH}	60	120	10 ns + (1,0 ns/pF) C _L
	10		30	60	9 ns + (0,42 ns/pF) C _L
	15		20	40	6 ns + (0,28 ns/pF) C _L

14



ENSI CAEN Digital Electronic

III. Logique combinatoire

Fonctions logiques → Synthèse combinatoire

Portes logiques

Synthèse combinatoire

fonction logique à réaliser sous forme de table ou d'équation

choix d'une cible technologique

réalisation matérielle de la fonction

- logique anarchique
- opérateurs élémentaires
- logique structurée (mémoire ROM, PAL, CPLD, FPGA)

16

16

ENSI CAEN Digital Electronic

III. Logique combinatoire

Fonctions logiques → Synthèse combinatoire

Portes logiques

Synthèse combinatoire

fonction logique à réaliser sous forme de table ou d'équation

choix d'une cible technologique

réalisation matérielle de la fonction

Note

le choix de la cible technologique dépend

- de la complexité de la fonction à réaliser
- du coût
- de la vitesse de traitement nécessaire
- du savoir-faire du concepteur
- ...

- logique anarchique
- opérateurs élémentaires
- logique structurée (mémoire ROM, PAL, CPLD, FPGA)

17

17



ENSICAEN Digital Electronic


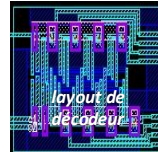
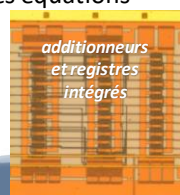
III. Logique combinatoire

Synthèse combinatoire – 1- Logique anarchique

- à base de porte logique AND, OR, NAND, NOR, XOR, INV.
 - méthode historique
 - adaptée à de très faibles complexités dans le cas de composants discrets
 - utilisation de composants discrets en voie d'obsolescence
- base de pour les circuits intégrés full-custom
 - (ASIC: Application Specific Integrated Circuit)
- création de bloc élémentaire pour des structures plus complexes
 - cas des opérateur arithmétiques

première étape de la synthèse, simplification des équations

- algèbre de Boole
- table de Karnaugh
- logiciel de synthèse

18

18

ENSICAEN Digital Electronic

III. Logique combinatoire

Synthèse combinatoire - 1- Logique anarchique - Table de Karnaugh

- outil graphique pour la simplification des équations
- fait apparaître la symétrie sur les variables

	ba	0 0	0 1	1 1	1 0
c		s_0	s_1	s_3	s_2
0		s_4	s_5	s_7	s_6
1					

ATTENTION !

Si l'on fait un groupement autour de cet axe de symétrie, cela élimine la variable c

- Le groupement d'éléments se traduit par la suppression de variables dans l'expression de la fonction.

Groupe de 2 éléments : suppression d'une variable
 Groupe de 4 éléments : suppression de deux variables

Idée : outil graphique pour simplifier l'expression algébrique

19

19



ENSICAEN Digital Electronic
III. Logique combinatoire

Synthèse combinatoire - 1- Logique anarchique - Table de Karnaugh

- Règles de simplification**
 - rechercher les « 1 » isolés,
 - faire croître par adjacence les regroupements,
 - faire des regroupements de taille maximale,
 - ne prendre que les regroupements ou termes produits nécessaires pour prendre au moins une fois chaque 1 sans redondance.

Note quand la valeur de sortie n'est pas définie, on associe à la case de la table le caractère \emptyset (don't care) ou X.

- Limité aux fonctions ayant au plus 4 entrées.**
- Permet de créer des sous-structures élémentaires simples ensuite cascadées. C'est le cas pour les opérateurs arithmétiques

Idee : méthode historique utile en TD, TP, rapidement limitée

20

20

ENSICAEN Digital Electronic
III. Logique combinatoire

Synthèse combinatoire - 1- Logique anarchique - Table de Karnaugh

Table de vérité

c	b	a	f
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

Table de Karnaugh

c \ ba	00	01	11	10
0	0	0	0	1
1	0	1	0	1

Table de Karnaugh

c \ ba	00	01	11	10
0	0	0	0	1
1	0	1	0	1

Table de Karnaugh

d \ c	00	01	11	10
00				
01				
11				
10				

Table de Karnaugh

e \ d	000	001	011	010	110	111	101	100
00	0	0	0	0	0	0	1	0
01	0	0	0	0	1	1	1	0
11	0	0	1	0	0	1	1	1
10	1	0	1	1	1	1	0	1

21

21



ENSI CAEN
Digital Electronic

III. Logique combinatoire

Synthèse combinatoire - 1- Logique anarchique - Table de Karnaugh

	b a	00	01	11	10
d c	00				
	01				
	11				
	10				

	b a	00	01	11	10
d c	00				
	01				
	11				
	10				

	b a	00	01	11	10
d c	00				
	01				
	11				
	10				

	c b a	000	001	011	010	110	111	101	100
e d	00	0	0	0	0	0	0	1	0
	01	0	0	0	0	1	1	1	0
	11	0	0	1	0	0	1	1	1
	10	1	0	1	1	1	1	0	1

$F = e\bar{d}\bar{a} + eba + \bar{e}dcb + \bar{e}c\bar{b}a + edc\bar{b}$

22

22

ENSI CAEN
Digital Electronic

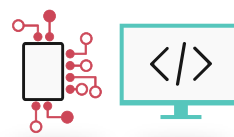
III. Logique combinatoire

Synthèse combinatoire - 1- Logique anarchique - Table de Karnaugh

Écrivez les équations Booléennes d'une fonction active pour les nombres premiers entre 0 et 7

23

23



ENSICAEN Digital Electronic

III. Logique combinatoire

Synthèse combinatoire - 1- Logique anarchique - Table de Karnaugh

Exemple du transcodeur 4B6B

Le transcodage 4B6B étend un code de 4 bits en un code de 6 bits contenant le même nombre de "1" et de "0".

	Code binaire 4B	Code 6B
	$b_3 b_2 b_1 b_0$	$g_5 g_4 g_3 g_2 g_1 g_0$
0	0 0 0 0	0 0 1 1 1 0
1	0 0 0 1	0 0 1 1 0 1
2	0 0 1 0	0 1 0 0 1 1
3	0 0 1 1	0 1 0 1 1 0
4	0 1 0 0	0 1 0 1 0 1
5	0 1 0 1	1 0 0 0 1 1
6	0 1 1 0	1 0 0 1 1 0
7	0 1 1 1	1 0 0 1 0 1
8	1 0 0 0	0 1 1 0 0 1
9	1 0 0 1	0 1 1 0 1 0
A	1 0 1 0	0 1 1 1 0 0
B	1 0 1 1	1 1 0 0 0 1
C	1 1 0 0	1 1 0 0 1 0
D	1 1 0 1	1 0 1 0 0 1
E	1 1 1 0	1 0 1 0 1 0
F	1 1 1 1	1 0 1 1 0 0

Table de vérité de la fonction logique de transcodage 4B6B

Entrées de la fonction: b_3, b_2, b_1, b_0

Sorties de la fonction: $g_5, g_4, g_3, g_2, g_1, g_0$

Fonction logique combinatoire de transcodage 4B6B

Fonction logique combinatoire à 4 entrées et 6 sorties.

Exemple de résolution pour la sortie g_5 Il faut faire de même pour les 5 autres sorties (g_0 à g_4).

g_5	$b_3 b_2$	$b_1 b_0$	00	01	11	10
00	0	0	0	0	0	0
01	0	1	1	1	1	1
11	1	1	1	1	1	1
10	0	0	1	0	0	0

$$g_5 = b_2 b_1 + b_2 b_0 + b_3 b_2 + b_3 b_1 b_0$$

24

ENSICAEN Digital Electronic

III. Logique combinatoire

Synthèse combinatoire - 1- Logique anarchique – Logiciel de synthèse logique

- algorithme de réduction algébrique.

à partir de la table de vérité de la fonction et du choix du type de portes logiques

Exemple avec le logiciel gratuit Logisim

[Logisim: Download \(cburch.com\)](http://Logisim: Download (cburch.com))

25



ENSI CAEN Digital Electronic

III. Logique combinatoire

Synthèse combinatoire - 1- Logique anarchique – Logiciel de synthèse logique

Exemple de l'algorithme de Quine-Mac Cluskey (1956) utilise la 1^{ère} forme canonique ($\Sigma\Pi$). Les combinaisons sont rassemblées en groupes en fonction du nombre de 1 des minterms pour lesquels la fonction est vraie. Des simplifications sont faites entre les minterms adjacents de groupes consécutifs.

Sur un exemple :

d	c	b	a	f
0	0	0	0	1
0	0	0	1	1
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1

rassemblement des minterms

grp	N°	d	c	b	a
0	0	0	0	0	0
1	1	0	0	0	1
	2	0	1	0	0
2	3	0	1	0	1
	4	1	0	0	1
	5	1	0	1	0
	6	0	1	1	1
3	7	1	1	0	1
	8	1	1	1	1

26

26

ENSI CAEN Digital Electronic

III. Logique combinatoire

Synthèse combinatoire - Logique anarchique – Logiciel de synthèse logique

grp	N°	d	c	b	a
0	0	0	0	0	0
1	1	0	0	0	1
	2	0	1	0	0
2	3	0	1	0	1
	4	1	0	0	1
	5	1	0	1	0
	6	0	1	1	1
3	7	1	1	0	1
	8	1	1	1	1

simplifications

grp	N°	d	c	b	a
0-1	0-1	0	0	0	X
	0-2	0	X	0	0
1-2	1-4	X	0	0	1
	2-3	0	1	0	X
2-3	3-6	0	1	X	1
	3-7	X	1	0	1
	4-7	1	X	0	1
	6-8	X	1	1	1
3-4	7-8	1	1	X	1

le N°5 n'est pas représenté

comparaison du minterm du groupe 0 avec les minterms du groupe 1

0-1	0	0	0	X
0-2	0	X	0	0

comparaison des minterms du groupe 1 avec les minterms du groupe 2

1-4	X	0	0	1
2-3	0	1	0	X

comparaison des minterms du groupe 2 avec les minterms du groupe 3

3-6	0	1	X	1
3-7	X	1	0	1
4-7	1	X	0	1

comparaison des minterms du groupe 3 avec le minterm du groupe 4

6-8	X	1	1	1
7-8	1	1	0	1

27

27



ENSICAEN Digital Electronic
III. Logique combinatoire
Synthèse combinatoire - Logique anarchique – Logiciel de synthèse logique

grp	N°	d	c	b	a
0-1	0-1	0	0	0	X
	0-2	0	X	0	0
1-2	1-4	X	0	0	1
	2-3	0	1	0	X
2-3	3-6	0	1	X	1
	3-7	X	1	0	1
	4-7	1	X	0	1
3-4	6-8	X	1	1	1
	7-8	1	1	X	1

le N°5 n'est pas représenté

comparaison du minterm du groupe 0-1 avec les minterms du groupe 1-2

0-1 ; 2-3	0	X	0	X
-----------	---	---	---	---

comparaison des minterms du groupe 1-2 avec les minterms du groupe 2-3

1-4 ; 3-7	X	X	0	1
-----------	---	---	---	---

comparaison des minterms du groupe 2-3 avec les minterms du groupe 3-4

3-6 ; 7-8	X	1	X	1
3-7 ; 6-8	X	1	X	1

le N°5 n'est pas représenté

simplicifications

grp	N°	d	c	b	a
0-1 ; 1-2	0-1 ; 2-3	0	X	0	X
1-2 ; 2-3	1-4 ; 3-7	X	X	0	1
2-3 ; 3-4	3-6 ; 7-8	X	1	X	1
	3-7 ; 6-8	X	1	X	1

le N°5 n'est pas représenté

28

ENSICAEN Digital Electronic
III. Logique combinatoire
Synthèse combinatoire - 1- Logique anarchique – Logiciel de synthèse logique

grp	N°	d	c	b	a
0-1 ; 1-2	0-1 ; 2-3	0	X	0	X
1-2 ; 2-3	1-4 ; 3-7	X	X	0	1
2-3 ; 3-4	3-6 ; 7-8	X	1	X	1
	3-7 ; 6-8	X	1	X	1

le N°5 n'est pas représenté

1^{er} réarrangement

grp	N°	d	c	b	a
0	0	0	0	0	0
1	1	0	0	0	1
	2	0	1	0	0
2	3	0	1	0	1
	4	1	0	0	1
	5	1	0	1	0
3	6	0	1	1	1
	7	1	1	0	1
4	8	1	1	1	1

le N°5 n'est pas représenté

Finalement, la fonction est représentée par: (5), (0-1 ; 2-3), (1-4 ; 3-7), (3-6 ; 7-8)

ou encore : $d\bar{c}b\bar{a} + \bar{d}b + \bar{b}a + ca$

29



ENSI CAEN Digital Electronic

III. Logique combinatoire

Synthèse combinatoire - 1- Logique anarchique – Point sur les portes logiques

Point sur les portes logiques discrètes

2 séries et 2 familles technologiques (principalement)

série 4000
Nom commençant par 40 __

technologie CMOS
Complementary-Metal-Oxide-Semiconductor

Alimentation 3V à 15V

Caractéristiques
vitesse dépend de la tension d'alimentation,
fréquence typiquement < 1MHz
plus faible consommation de puissance que TTL

série 74
Nom commençant par 74

TTL (Transistor-Transistor-Logic)
ou **CMOS** ou mixte **BiCMOS**

Alimentation **5V** (toujours)

Caractéristiques
plus rapides, mais plus de
consommation de puissance

Rmq en intégré 3.3V, 1.8V, 0,7V

à retenir !

30

30

ENSI CAEN Digital Electronic

III. Logique combinatoire

Synthèse combinatoire - 1- Logique anarchique – Point sur les portes logiques

Noms des portes logiques discrètes

H	High speed
LV	Low voltage
C	CMOS
HC	High speed CMOS
LS	Low power Schottky

série

fonction logique

XX74LS11

spécification technologie, vitesse, consommation

40 __	fonction
00	NOT
01	NOR2
11	NAND2
23	NAND3
25	NOR3
30	XOR2
71	OR2
73	AND3
81	AND2

FAIRCHILD SEMICONDUCTOR™

DM74LS11
Triple 3-Input AND Gate

General Description
This device contains three independent gates each of which performs the logic AND function.

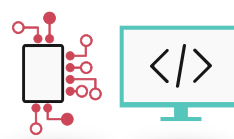
Philips Semiconductors Product specification

Quadruple exclusive-OR gate	CD4030 0,57 € Farnell 2013	HEF4030B
	1,09 € Farnell 2023	gates

fabricants : NXP, Fairchild, Intersil, Texas Instruments, ...

31

31



ENSI CAEN Digital Electronic

III. Logique combinatoire

Synthèse combinatoire - 1- Logique anarchique – Point sur les portes logiques

Chemin critique

- Le parcours qui va générer le délai de propagation le plus long entre les entrées et la sortie du circuit. C'est a priori le chemin qui traverse le plus de portes logiques.

porte	Délai (ns)
NOT	30
AND	60
OR	70

Rmq Le chemin critique détermine le temps de calcul des opérateurs arithmétiques pour le calcul.

Rmq Le chemin critique limite la fréquence maximum de fonctionnement des systèmes synchrones.

32

32

ENSI CAEN Digital Electronic

III. Logique combinatoire

Synthèse combinatoire - 1- Logique anarchique – Point sur les portes logiques

Glitch : variations transitoires des niveaux logiques avant d'atteindre un état

$A=0$

$B=1 \rightarrow 0$

$C=1$

$Y = \overline{A}B + BC$

N_1

N_2

Chemin critique

Sortie Y Y

Glitch

33

33



ENSI CAEN Digital Electronic

III. Logique combinatoire

Synthèse combinatoire - 1- Logique anarchique – Point sur les portes logiques

Glitch : solutions

1. Les éliminer en modifiant le circuit (par exemple termes redondants de Consensus)
2. Ne pas s'en préoccuper en utilisant une **conception synchrone** avec un cadencement d'horloge adapté (on ne prend en compte le signal que périodiquement, une fois stabilisé).

	BA	00	01	11	10
C	0	1	0	0	0
	1	1	0	1	1

34

ENSI CAEN Digital Electronic

III. Logique combinatoire

Synthèse combinatoire - 1- Logique anarchique – Point sur les portes logiques

Sortie d'un circuit logique

Types de sortie (étage de sortie): la sortie d'un circuit logique peut être fournie soit par 2 transistors complémentaires « totem-pole » ou par un seul transistor « collecteur ouvert ».

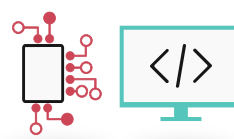
La sortie totem-pole est pratique pour cascader des circuits de même technologie, même alimentation.

La sortie collecteur ouvert est pratique pour changer de niveau de tension ou la réalisation d'un ET câblé (bus I2C).

Sortance : capacité d'une porte logique à servir de source à d'autres portes logiques. Courant de sortie > somme de courant d'entrée des portes en aval. En CMOS limite de vitesse de fonctionnement.

Court-circuit (état indéterminé)

35



ENSI CAEN Digital Electronic

III. Logique combinatoire

Synthèse combinatoire – 2- Opérateurs élémentaires

Opérateurs élémentaires

- multiplexeurs,
- démultiplexeur,
- décodeur,
- encodeur

Rmq Pédagogique, en pratique plus utilisé

Rmq opérateurs élémentaires eux-mêmes constitués de portes logiques ET, NON, OU...

Rmq comme la logique anarchique avec des portes ET, OU, NON, ... discrètes, ce type d'implémentation est en voie d'obsolescence et réservé à des fonctions logiques simples

Idée : opérateurs ne sont plus utilisés pour la synthèse logique mais sont des éléments de base des systèmes numériques

36

36

ENSI CAEN Digital Electronic

III. Logique combinatoire

Synthèse combinatoire – 2- Opérateurs élémentaires - Multiplexeur

Multiplexeur

- **fonction d'aiguillage**
- exemple d'un multiplexeur 4 vers 1

c ₁	c ₀	s
0	0	e ₀
0	1	e ₁
1	0	e ₂
1	1	e ₃

$$s = e_0 \cdot \bar{c}_1 \cdot \bar{c}_0 + e_1 \cdot \bar{c}_1 \cdot c_0 + e_2 \cdot c_1 \cdot \bar{c}_0 + e_3 \cdot c_1 \cdot c_0$$

37

37



ENSI CAEN Digital Electronic

III. Logique combinatoire

Synthèse combinatoire - 2- Opérateurs élémentaires – Multiplexeur (cont.)

- entrée de sélection : entrée de la table
- exemple de la fonction XOR

Rmq Pédagogique, en pratique plus utilisé

c ₁	c ₀	s
0	0	0 e ₀
0	1	1 e ₁
1	0	1 e ₂
1	1	0 e ₃

Fonction XOR

38

ENSI CAEN Digital Electronic

III. Logique combinatoire

Synthèse combinatoire - 2- Opérateurs élémentaires - Démultiplexeur

Démultiplexeur

- opérateur dual

$$S_0 = \bar{c}_1 \cdot \bar{c}_0 \cdot e$$

$$S_1 = \bar{c}_1 \cdot c_0 \cdot e$$

$$S_2 = c_1 \cdot \bar{c}_0 \cdot e$$

$$S_3 = c_1 \cdot c_0 \cdot e$$

c ₁	c ₀	s ₀	s ₁	s ₂	s ₃
0	0	e	0	0	0
0	1	0	e	0	0
1	0	0	0	e	0
1	1	0	0	0	e

39



ENSI CAEN Digital Electronic

III. Logique combinatoire

Synthèse combinatoire - 2- Opérateurs élémentaires - Décodeur

Décodeur

- active une sortie parmi 2^N grâce à N signaux de commande.
- utilisé pour « adresser ».**
- équivalent du démultiplexeur avec une entrée de donnée à « 1 »
- exemple de décodeur 4 sorties

$$S_0 = \overline{C_1} \cdot \overline{C_0}$$

$$S_1 = C_1 \cdot \overline{C_0}$$

$$S_2 = \overline{C_1} \cdot C_0$$

$$S_3 = C_1 \cdot C_0$$

c_1	c_0	s_0	s_1	s_2	s_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

Rmq Dans certains décodeurs discrets, la sortie active peut être au niveau bas et les inactives au niveau haut.

40

40

ENSI CAEN Digital Electronic

III. Logique combinatoire

Synthèse combinatoire - 3- Logique structurée

Logique structurée

- structures universelles prédéfinies sous forme de matrice
- forte densité d'intégration → VLSI
- structure dépend du composant :
 - ROM, espace mémoire et système d'adressage → **Note** structure la plus universelle
 - PAL, matrices de ET et de OU pouvant être interconnectées
 - CPLD, macrocellules logiques interconnectables
 - FPGA

Note toutes les fonctions logiques ne sont pas réalisables, mais il y a de la marge

41

41



ENSICAEN Digital Electronic
III. Logique combinatoire

Synthèse combinatoire - 3- Logique structurée - Structure ROM

- mémoire Mxn : sauvegarde de M mots de n bits.
- Exemple :
MC28C64C : 64 kbit ; 8kx8bits, parallel EEPROM
nombre de bits d'adresse ?

Idee : prétexte pour parler des mémoires, n'est plus utilisé en pratique pour la synthèse logique, mais reste un élément fondamental de l'électronique numérique

42

42

ENSICAEN Digital Electronic
III. Logique combinatoire

Synthèse combinatoire - 3- Logique structurée - Structure ROM (cont.)

- La taille du mot mémorisé permet de définir autant de fonctions logiques que son nombre de bits.
- Le nombre de bits d'adresse correspond au nombre de variables des fonctions logiques.

Remarque :
plus petites mémoires
2kx8 : 16k RAM 1,6€ (2021)
2k EEPROM : 0,26€ 24C02 (2021)

Rmq

- Exemple : implémentation de 5 fonctions logiques à 3 entrées

c	b	a	F ₄	F ₃	F ₂	F ₁	F ₀
0	0	0	0	1	1	0	0
0	0	1	1	1	0	1	0
0	1	0	1	0	0	0	1
0	1	1	0	0	1	0	0
1	0	0	0	1	0	1	0
1	0	1	1	1	1	1	0
1	1	0	0	1	0	1	1
1	1	1	0	0	1	1	1

43

43



ENSI CAEN Digital Electronic

III. Logique combinatoire

Synthèse combinatoire - 3- Logique structurée - Structure ROM (cont.)

c	b	a	F ₄	F ₃	F ₂	F ₁	F ₀
0	0	0	0	1	1	0	0
0	0	1	1	1	0	1	0
0	1	0	1	0	0	0	1
0	1	1	0	0	1	0	0
1	0	0	0	1	0	1	0
1	0	1	1	1	1	1	0
1	1	0	0	1	0	1	1
1	1	1	0	0	1	1	1

44

44

ENSI CAEN Digital Electronic

III. Logique combinatoire

Synthèse combinatoire - 3- Logique structurée - Structure PAL

historique (années 80)

Programmable Logic Array

- constituée de deux sous-ensembles :
 - une matrice de ET,
 - une matrice de OU.

PAL 16V8

nb variables

nb sorties

fabricants : Lattice Semiconductor, AMD (Advanced Micro Devices), Texas Instrument

Alimentation +5 V

Matrice produit

Matrice somme

$S_0 = A \cdot \bar{B} \cdot \bar{D} + \bar{A} \cdot B \cdot C$

T1C PAL22V10Z-25C 17,26 \$ Texas Instrument 2022

45

45



ENSI CAEN Digital Electronic

III. Logique combinatoire

Synthèse combinatoire - 3- Logique structurée - Structure PAL (cont.)

- implémentation de la fonction XOR

Alimentation +5 V

Matrice produit

Matrice somme

$$S_0 = \bar{A}.B + A.\bar{B}$$

46

46

ENSI CAEN Digital Electronic

III. Logique combinatoire

Synthèse combinatoire - 3- Logique structurée – Structures CPLD, FPGA

CPLD : Complex Programmable Logic Device
FPGA : Field Programmable Gate Array

- réseaux logiques programmables complexes formés de blocs logiques disposés sous forme matricielle

Donnée entrée

Variables logiques

Activation d'horloge

Horloge

RAZ Inactif au 0

RAZ global

Mux : multiplexeur

bloc logique de FPGA Look-Up Table (LUT)

Réseau du bloc logique

Expansion logique parallèle (depuis autres macrocellules)

Matrice sélection termes produit

Expansion logique partagée

Signaux depuis matrice interconnexion

EN Enable Activation (activation)

PR Prezet (mise à 1)

Horloge globale

RAZ global

Selection horloge Enable

Bascule programmable

Vers matrice interconnexion

Vers bloc E/S

macrocellule de CPLD

Idee : composants de l'électronique numérique programmée actuelle

Figure 24 – Exemple de macrocellule d'un CPLD (série MAX3000 d'Altera)

47

47



ENSI CAEN Digital Electronic

III. Logique combinatoire

Synthèse combinatoire - 3- Logique structurée – Structures CPLD, FPGA (cont.)

structure CPLD

structure FPGA

FPGA plus interconnecté ⇒ plus flexible
 ⇒ plus difficile à implémenter
 ⇒ délais plus difficiles à estimer
 FPGA plus de ressources intégrées

Note implémentation avec un logiciel et un langage de programmation

48

48

ENSI CAEN Digital Electronic

III. Logique combinatoire

Synthèse combinatoire - 3- Logique structurée – Structures CPLD, FPGA (cont.)

Fabricant FPGA : Xilinx, Altera, Lattice

Fabricant CPLD : Altera, Lattice, Atmel

Altera 5M40ZE64 3,44 €
Farnell 2022
32 macrocellules

Lattice ICE40 4,26 €
Radiospire 2022
1280 blocs logiques,
16kbits RAM mémoire

Artix 7 Xilinx 433,86 €
Farnell 2022
215360 macrocellules,
676 broches

28nm ARTIX 7G
6G Transceivers

49

49



III. Logique combinatoire Synthèse combinatoire - 3- Logique structurée – Structures CPLD, FPGA (cont.)

ENSICAEN Digital Electronic

VHDL, verilog

Conception

- Capture
 - schématique
 - VHDL/Verilog
- Implémentation
 - transformation/synthèse
 - vérification des règles
 - partitionnement
 - placement des blocs
 - routage
 - création du fichier de programmation
- Programmation du composant

Vérification

- Simulation fonctionnelle
 - vérification logique
 - estimation temporelle
- Simulation temporelle
 - après placement
- Analyse temporelle statique
 - après placement
- Débogage en fonctionnement
 - pour partie programmable
 - utilise avec autre système matériel ou logiciel

```

1 LIBRARY ieee, work;
2 USE ieee_std_logic_1164.all;
3 USE ieee_math_real.all;
4 USE work.types.all;
5
6
7 ENTITY PFD IS
8 PORT
9 (
10     dn      : OUT_CLOCK;
11     up      : OUT_CLOCK;
12     div     : IN_CLOCK;
13     ref     : IN_CLOCK
14 );
15 END PFD;
16
17 ARCHITECTURE behaviour OF PFD IS
18     SIGNAL reset_s : STD_LOGIC := '0';
19     SIGNAL up_s    : STD_LOGIC := '0';
20     SIGNAL dn_s    : STD_LOGIC := '0';
21     SIGNAL upn_s   : STD_LOGIC := '0';
22     SIGNAL upn_s   : STD_LOGIC := '0';
23
24 BEGIN
25
26 Reference : PROCESS (ref, reset_s)
27 BEGIN
28     IF (reset_s = '0') THEN
29         up_s <= '0';
30     ELSIF (ref.dig'EVENT and ref.dig = '1') THEN
31         dn_s <= '1' AFTER 10 PS;
32     END IF;
33 END PROCESS;
34
35 Divider : PROCESS (div, reset_s)
36 BEGIN
37     IF (reset_s = '0') THEN
38         dn_s <= '0';
39     ELSIF (div.dig'EVENT and div.dig = '1') THEN
40         up_s <= '1' AFTER 10 PS;
41     END IF;
42 END PROCESS;
43
44 upn_s <= NOT(up_s);
45 dnn_s <= NOT(dn_s);
46 reset_s <= (NOT(up_s)) OR (NOT(dn_s)) AFTER 10 PS;
47 up.dig <= up_s;
48 up.freq <= ref.freq;
49 up.duty <= ref.duty;
50
51 dn.dig <= dn_s;
52 dn.freq <= ref.freq;
53 dn.duty <= ref.duty;
54
55 END behaviour;
                
```

phase/freq detector (PFD)

50

III. Logique combinatoire Synthèse combinatoire – 4- Circuit intégrée (full custom, ASIC)

ENSICAEN Digital Electronic

ASIC (Application Specific Integrated Circuit)

circuit dédié ⇒ plus performant 😊

⇒ coût réduit pour des grandes quantités 😊

⇒ temps de conception plus long 😞

⇒ pas flexible 😞

FPGA

```

Functional Specification
↓
HDL
↓
Synthesis → Behavioral Simulation
↓
Place & Route → Static Timing Analysis
↓
Download and Verify in Circuit
                
```

ASIC

```

Functional Specification
↓
HDL
↓
Synthesis → Behavioral Simulation
           → Static Timing Analysis
           → Equivalency Checking
↓
Place & Route → Static Timing Analysis
              → Equivalency Checking
              → Verification of 2nd & 3rd Order Effects
↓
Verify in Circuit
                
```

Hand-Off to Foundry: Wait 1-3 Months

52




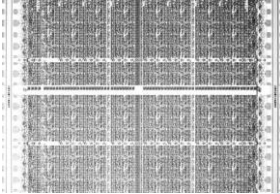
ENSI CAEN
 Digital Electronic
 III. Logique combinatoire

Choix de la cible pour l'implémentation

spécifique versus flexible

spécifique versus flexible

- Permanent, ne peut plus être modifié
- Temps de conception long (plrs mois à > d'un an)
- *Non-recurring engineering (NRE) costs* (10^5 - 10^6 \$)
- Optimisé pour l'application (meilleures performances -vitesse, consommation-)
- Coût plus faible pour de grande production
- peut être modifié (adaptation à un cahier des charges changeant)
- Temps de conception plus court (programmation rapide, test direct, moins de corps de métiers)
- Logiciels de développement peu onéreux

53

53

ENSI CAEN
 Digital Electronic
 III. Logique combinatoire



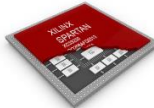





Choix de la cible pour l'implémentation

Onde de Makimoto

Spécifique versus flexible

standard
 1957 composants discrets
 1967 mémoires μ P
 1977
 1987
 1997 FPGA
 2007
 2017 P-SoC/P-SiP

spécifique
 composants spécifiques pour montre, TV, calculatrices (LSI)
 circuits spécifiques (ASIC)
 SoC/SiP

54

54



ENSI CAEN
Digital Electronic

III. Logique combinatoire

Complément : d'un point de vue transistor

D'un point de vue physique, les portes logiques sont réalisées à partir de transistors. Plusieurs types de transistors peuvent être employés, mais le plus couramment utilisé est celui des transistors Métal Oxyde Isolant plus connu sous le sigle **transistor MOS**. La fabrication des portes logiques s'appuie alors sur l'utilisation de deux types de transistors : les transistors **NMOS** et les transistors **PMOS**. Ces transistors sont utilisés comme des interrupteurs et chacun de ces transistors est caractérisé par les niveaux logiques qui le rendent assimilable à un **interrupteur ouvert ou bien fermé**.

<p>NMOS</p> <p>interrupteur ouvert</p> <p>interrupteur fermé</p>	<p>substrat P</p>
<p>PMOS</p> <p>interrupteur fermé</p> <p>interrupteur ouvert</p>	<p>substrat N</p>

55

ENSI CAEN
Digital Electronic

III. Logique combinatoire

Complément : d'un point de vue transistor (cont.)

Exemples de portes logiques

56



ENSICAEN Digital Electronic

III. Logique combinatoire

Complément : d'un point de vue transistor (cont.)

Exemples de portes logiques

57

57

ENSICAEN Digital Electronic

III. Logique combinatoire

Complément : d'un point de vue transistor (cont.)

derrière le transistor le silicium...

Exemples de layout de portes logiques

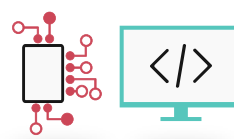
schéma logique

schéma électrique

layout

58

58



ENSICAEN Digital Electronic

III. Logique combinatoire

Complément : d'un point de vue transistor (cont.)

derrière le transistor le silicium...

Exemples de layout de portes logiques (cont.)

59

59

ENSICAEN Digital Electronic

III. Logique combinatoire

Performances des circuits numériques

Temps de commutation et puissance dissipée

- Temps de commutation (delay)
- Puissance dissipée
 - Deux aspects différents :
 - Puissance dynamique : activité
 - Puissance de fuite (leakage power)

$$delay \propto \frac{V_{dd}}{(V_{dd} - V_{th})^2}$$

$$P = C \cdot V_{dd}^2 \cdot F$$

I_{sub} : courant de fuite sous le seuil (subthreshold off state leakage current)
I_g : courant de fuite par effet tunnel (gate tunnelling leakage current)
I_d : courant de fuite de jonction inverse (reverse junction leakage current)
I_{gidl} : courant de fuite induit par la grille (Gate Induced Drain Leakage)
I_{pt} : courant de fuite transistor à canal court (Drain Source Punch Through)

Technologie sub-micron → 50/50

60



ENSICAEN Digital Electronic

III. Logique combinatoire

A l'issue de cette troisième partie

?

- Quel outil graphique permet de faciliter la synthèse de petits circuits logiques combinatoires ? Quelle est sa limitation ?
- Comment fait-on la synthèse de circuit logique combinatoire sur les composants de type logique structurée ?
- Quels sont les avantages/inconvénients comparés des solutions programmables de type FPGA et fixe ASIC ?

61

61

ENSICAEN Digital Electronic

III. Logique combinatoire

Complément : choix des tensions d'alimentation

pourquoi 5V, 3.3V, 1.8V, 0.7V ?

JEDEC* committee

JC-16 Interface Technology

The activities within JC-16's scope include the specification of power supply voltage levels for digital integrated circuits and the definition of electrical interfaces between the components of a system. The committee's scope further encompasses interface protocols, modeling, simulation, testing environments, and verification.

EIAJ ED-5001A

2.2 Recommended operating conditions

Table 2 Recommended operating conditions

Parameter	Symbol	Normal range (3.3V nominal)	Wide range (3V nominal)	Unit
Power supply voltage	V_{DD}	3.0 ~ 3.6	2.7 ~ 3.6	V
Operating temperature range	T_a	Note 3	Note 3	°C

Note 3: Specified by manufacturer for various purposes, respectively.

5-V LOGIC LEVELS

5-V CMOS $V_{CC} = 4.5 - 5.5 V$

TTL $V_{CC} = 4.5 - 5.5 V$

LVTTTL $V_{CC} = 2.7 - 3.6 V$
EIA/JESD 8A

LOW-VOLTAGE LEVELS

2.5 V $V_{CC} = 2.3 - 2.7 V$
EIA/JESD 8-5

1.8 V $V_{CC} = 1.65 - 1.95 V$
EIA/JESD 8-5

$V_{OH} = 4.44 V$

$V_{IH} = 0.7 \times V_{CC}$

$V_{TH} = 0.5 \times V_{CC}$

$V_{IL} = 0.3 \times V_{CC}$

$V_{OL} = 0.5 V$

$V_{OH} = 2.4 V$

$V_{IH} = 2.0 V$

$V_{TH} = 1.5 V$

$V_{IL} = 0.8 V$

$V_{OL} = 0.4 V$

$V_{OH} = 2 V$

$V_{IH} = 1.7 V$

$V_{IL} = 0.7 V$

$V_{OL} = 0.4 V$

$V_{OH} = V_{CC} - 0.45 V$

$V_{IH} = 0.65 V_{CC}$

$V_{IL} = 0.35 V_{CC}$

$V_{OL} = 0.45 V$

3,3V CMOS compatible avec TTL

$\sim 1/\sqrt{2}$

*Joint Electron Device Engineering Council

62

62



ENSI CAEN
Digital Electronic

III. Logique combinatoire

Complément : choix des tensions d'alimentation

pourquoi réduit-on la tension d'alimentation ?

- Cela réduit la puissance active consommée
- Pour conserver un champ électrique constant lors de la miniaturisation

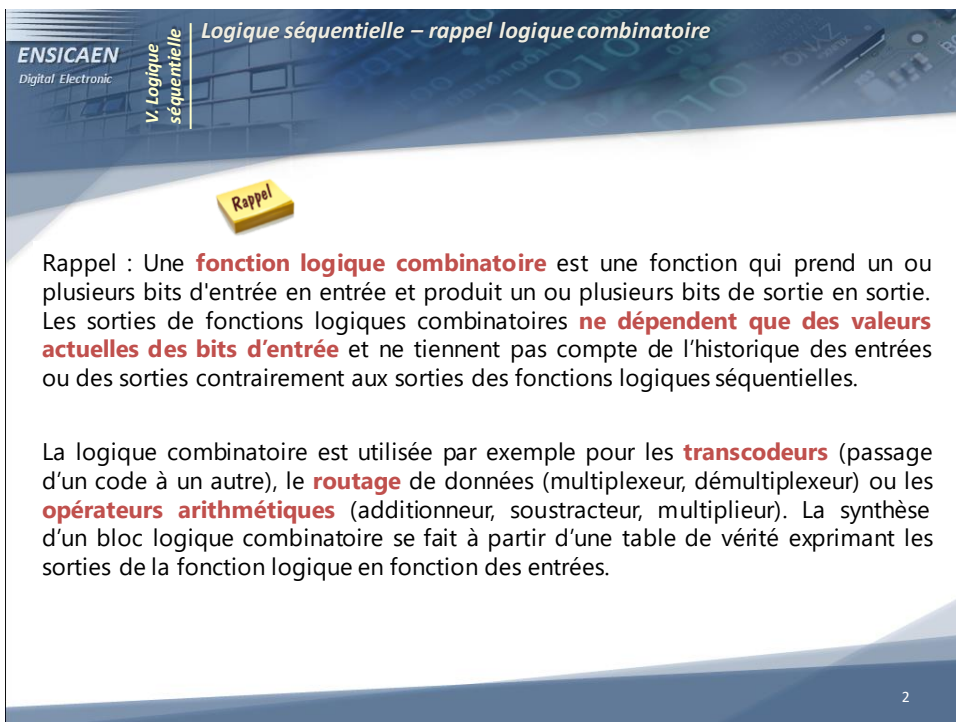
$$P = C \cdot V_{dd}^2 \cdot F$$

$$E_{ox} = \frac{V_{dd}}{t_{ox}}$$

Évolution de la tension d'alimentation V_{dd} , de la tension de seuil V_{th} et de l'épaisseur d'oxyde en fonction de la technologie.



1



2

Logique séquentielle – Généralités

ENSICAEN Digital Electronic
V. Logique séquentielle

Généralités

$A+B+C+D+\dots$

important ! calcul les uns après les autres
Notion de séquentialité

3

3

Logique séquentielle – Généralités (cont.)

ENSICAEN Digital Electronic
V. Logique séquentielle

compteur : état $N \Rightarrow$ état $N+1$

Incrémenter d'un état

L'état suivant dépend de l'état présent.
Il faut connaître l'état présent.

4

4

ENSICAEN Digital Electronic | V. Logique séquentielle | Logique séquentielle – Généralités (cont.)

Les portes logiques combinatoires n'ont pas de « **mémoire** », la valeur de leurs sorties ne dépend que de l'état des entrées.

Ces portes adaptées au décodage ou aux opérations arithmétiques ne permettent pas seule de décrire un **système** dont l'**état** évolue.

Un état : une configuration possible du système.

Exemples :	lampe	allumée, éteinte, cassée
	feu tricolore	rouge, vert, orange clignotant, hors service
	ordinateur	allumé, en veille, en veille prolongé, éteint
	machine à laver	prélavage, lavage, rinçage, essorage, arrêt
	compteur	0, 1, 2, 3, ..., N
	smartphone	verrouillé, déverrouillé

des **événements** font évoluer le système (passage d'un état à un autre).
Exemples : signaux logiques activés (capteurs, boutons), durée écoulée (Timer)

5

5

ENSICAEN Digital Electronic | V. Logique séquentielle | Logique séquentielle – Généralités (cont.)

Fonction logique combinatoire :



entrée interrupteur / sortie ampoule
La valeur de la sortie ne dépend que de la valeur de l'entrée.

Fonction logique séquentielle :



entrée interrupteur / sortie feu piéton
La valeur de la sortie dépend de l'état du système et de l'entrée.

Idée : en logique séquentielle, le système se souvient que l'on a appuyé et sait dans quel état il est.

6

ENSICAEN Digital Electronic | V. Logique séquentielle | Logique séquentielle – Généralités (cont.)


Besoin d'un autre type de structures logiques :

- portes logiques séquentielles → bascules
- points mémoires

1. Bascules

Réalisées à partir de portes logiques élémentaires (ensembles de transistors).
Éléments fondamentaux permettant de **mémoriser** et de **modifier l'information**.

Utilisées pour le stockage temporaire de données et la réalisation d'automates pour le contrôle de l'évolution des systèmes.

 **Remarque** : les compteurs sont un cas particulier d'automate.

7

7

ENSICAEN Digital Electronic | V. Logique séquentielle | Logique séquentielle – 1. Bascules

1. Bascules

Opérateurs élémentaires de mémorisation.

Leur état dépend de celui de leurs entrées mais également de leur état précédent.

$$\text{état}(i+1) = f(\text{état}(i), \text{entrées})$$

(verrouillé + touche écran => déverrouillé)

On distingue les bascules **asynchrones** sensibles au niveau des entrées et les bascules **synchrones** sensibles au front d'un signal spécial cadencant le système (typiquement un signal d'horloge).

Historiquement : bascules RS, bascules JK. **A l'heure actuelle uniquement des bascules synchrones, les bascules D** à front constituées de 2 bascules *latch* en série.

8

8

ENSICAEN Digital Electronic | V. Logique séquentielle | Logique séquentielle – 1. Bascules – Bascule latch

Bascule latch

Entrée de donnée D (état) et entrée de commande (LE *latch enable*).

Q : état présent
Q+ : état suivant

Table de vérité de la bascule latch

LE	D	Q+
0	0	Q
0	1	Q
1	0	0
1	1	1

mode verrouillée, mémorisation (LE=0)
mode transparent (LE=1)

Idée : soit on laisse entrer une nouvelle valeur soit on verrouille le contenu.

11

ENSICAEN Digital Electronic | V. Logique séquentielle | Logique séquentielle – 1. Bascules – Bascule latch (cont.)

Bascule latch : structures internes

Entrée de donnée D (état) et entrée de commande (LE).

Q : état présent , Q+ : état suivant

comment ça marche ?

Structure à portes NAND

Structure compacte à portes de transmission et inverseurs

Idée : c'est le rebouclage interne qui permet de mémoriser un état.

12

ENSICAEN Digital Electronic | V. Logique séquentielle | Logique séquentielle – 1. Bascules – Bascule latch (cont.)

Bascule latch : structures internes
 Entrée de donnée D (état) et entrée de commande (LE).
 Q : état présent, Q+ : état suivant

comment ça marche ?

Structure à portes NAND

Structure compacte à portes de transmission et inverseurs

Idee : c'est le rebouclage interne qui permet de mémoriser un état.

13

ENSICAEN Digital Electronic | V. Logique séquentielle | Logique séquentielle – 1. Bascules – Bascule latch (cont.)

pas possible avec ce type de bascule

Compteur :

0	1	2	3	4	5
0	0	0	0	1	1
0	0	1	1	0	0
0	➔ 1	➔ 0	➔ 1	➔ 0	➔ 1

➔ C'est un rebouclage utilisé dans beaucoup de systèmes séquentiels (compteurs par exemple)

⬇️

Besoin d'une structure pas entièrement transparente

14

Logique séquentielle – 1. Bascules – Bascule D à front

ENSICAEN Digital Electronic
V. Logique séquentielle

c'est LA porte logique séquentielle dont on se sert en pratique

à retenir !

Bascule D à front

La bascule D est **non transparente**. Pas de liaison directe entre l'entrée et la sortie. Elle résulte de l'association en série de deux bascules *latch*.

Une bascule maître et une bascule esclave. Lorsque l'une est transparente, l'autre est verrouillée.

Pour garantir un fonctionnement correct, il faut des signaux de commandes LE_1 et LE_2 sans recouvrement.

15

15

Logique séquentielle – 1. Bascules – Bascule D (cont.)

ENSICAEN Digital Electronic
V. Logique séquentielle

comment ça marche ?

Structure interne de la bascule D à front.

D	C	Q+
0	0	0
1	1	1

$C: 0 \Rightarrow 1$

17

17

Logique séquentielle – 1. Bascules – Remarques générales

Remarques générales en pratique :

Rmq Les bascules comprennent des entrées supplémentaires.
 Il faut respecter des temps de mise en place et de maintien des niveaux logiques pour garantir un bon fonctionnement des bascules.

Fig. 5 Logic diagram.

Fig. 1 Pin configuration. Fig. 2 Logic symbol.

Fig. 7 Waveforms showing the data set-up and hold times for the data input (D_n).

The shaded areas indicate when the input is permitted to change for predictable output performance.

(1) HC : $V_M = 50\%$; $V_I = \text{GND to } V_{CC}$
 HCT : $V_M = 1.3 \text{ V}$; $V_I = \text{GND to } 3 \text{ V}$.

18

Logique séquentielle – 2. Registres

Registres
registre : ensemble de bascules
 pour le **stockage temporaire d'information**

donnée ← état

registre de donnée registre d'état

horloge

horloge

horloge

registre

registre

19

ENSICAEN Digital Electronic | V. Logique séquentielle | Logique séquentielle – 2. Registres (cont.)

Autres types de registres

- registre à chargement parallèle
- registre à décalage
- conversion série-parallèle
- conversion parallèle-série
- génération SBPA

Rmq les données sont traitées sous forme parallèle par les opérateurs arithmétiques. Les données lors de communications entre équipements peuvent être sous forme série

? quelles communications série connaissez-vous ?

20

20

ENSICAEN Digital Electronic | V. Logique séquentielle | Logique séquentielle – 3. Machines à états finis

3. Machines à états finis (MAEF)

Finite State Machine (FSM) ou automates synchrones.

Une machine d'état est un **opérateur séquentiel** dont la sortie est fonction des entrées et de l'état précédent de la machine d'état. Elle utilise des bascules pour mémoriser l'état présent et des **blocs de logique combinatoire** pour générer les sorties et l'état futur.

Il existe 2 types de structure de machine d'état:

1. Machine de **Mealy** : les sorties dépendent de l'état et des entrées
2. Machine de **Moore** : les sorties dépendent exclusivement de l'état

Rmq **Remarque** : les compteurs sont un cas particulier de machine à états finis.

21

21

ENSICAEN Digital Electronic | V. Logique séquentielle | Logique séquentielle – 3. Machines à états finis – 3.1. Type Mealy

3.1 Machines à états finis (MAEF) type Mealy

les sorties peuvent changer de manière asynchrone.

À éviter si non indispensable

22

22

ENSICAEN Digital Electronic | V. Logique séquentielle | Logique séquentielle – 3. Machines à états finis – 3.2. Type Moore

c'est la structure dont on se sert en pratique

3.2 Machines à états finis (MAEF) type Moore

Le type Moore garantit des **signaux logiques synchrones**. Il est privilégié dans les applications intégrées et cœur de calculateur pour le séquenceur.

23

23

ENSICAEN Digital Electronic
V. Logique séquentielle

Logique séquentielle – 3. Machines à états finis – 3.2. Type Moore

c'est la structure dont on se sert en pratique

3.2 Machines à états finis (MAEF) type Moore

Bloc logique d'évolution des états prépare l'état suivant (celui qui sera chargé dans le registre au prochain front d'horloge)

Registre d'état mémorise l'état du système

Bloc logique de décodage des sorties met à jour les sorties du système en fonction de l'état présent

24

24

ENSICAEN Digital Electronic
V. Logique séquentielle

Logique séquentielle – 3. Machines à états finis – 3.3. Exemple d'un compteur

3.3 Machines à états finis (MAEF) type Moore : exemple d'un compteur [1/3]

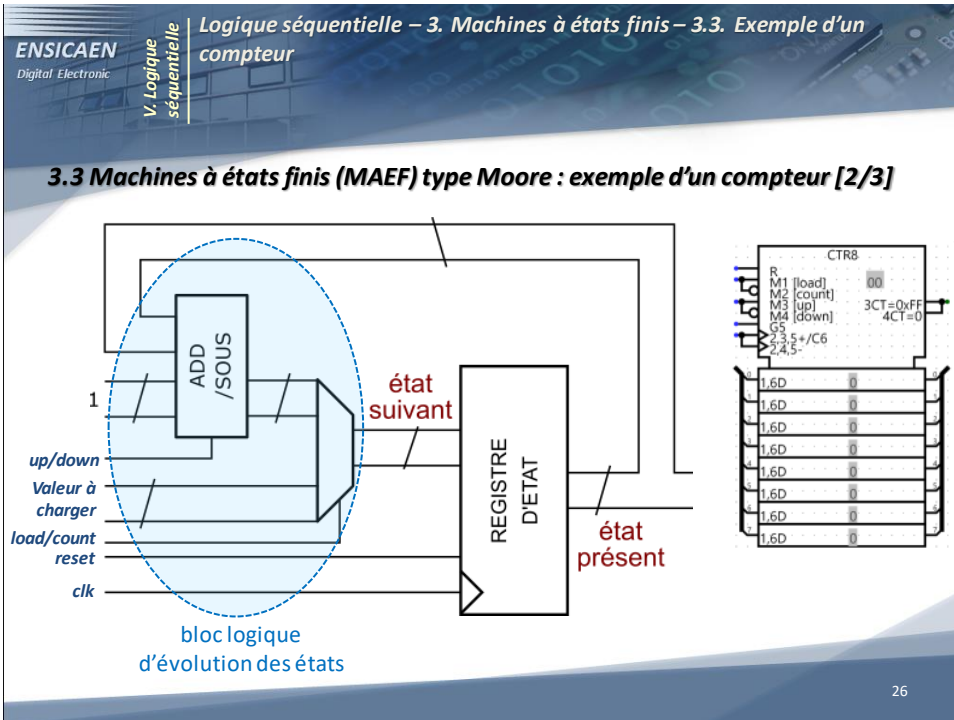
Entrées : commande de chargement ou de comptage (load/count). Sens de comptage (up/down). Valeur à charger. Mise à zéro (Reset).

Registre d'état mémorise l'état (valeur) du compteur

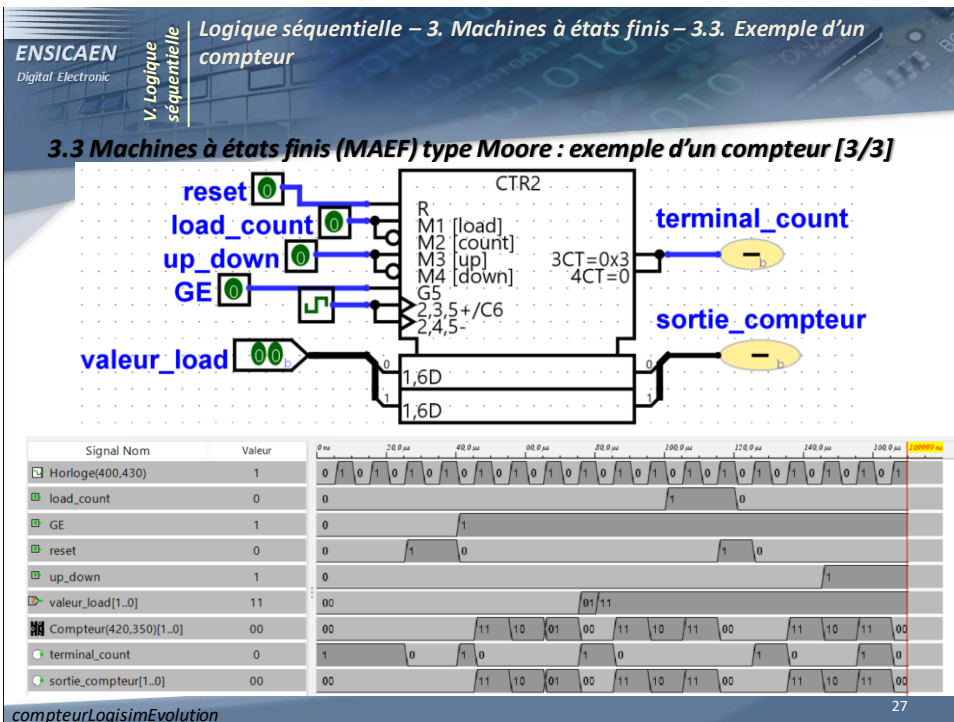
Valeur du compteur

25

25



26



27

ENSICAEN Digital Electronic | V. Logique séquentielle | Logique séquentielle – 3. Machines à états finis – 3.5. Compteurs

Remarque : Les compteurs sont un cas particulier (simple) de machine à états finis

Rmq

On peut dans certains cas utiliser un autre type de bascule pour les réaliser : des bascules T (*Toogle*)

Table de vérité de la bascule T

T	C	Q+
1	↓	\bar{Q}
0	-	Q

? Comment faire une bascule T à partir d'une bascule D à front ?

42

42

ENSICAEN Digital Electronic | V. Logique séquentielle | Logique séquentielle – 3. Machines à états finis – 3.5. Compteurs (cont.)

Exemple de compteur à partir de bascule T : composant discret 4029

actif à niveau bas

TRUTH TABLE

CLOCK	TE	PE	J	Q	\bar{Q}
X	X	0	0	0	1
X	X	0	1	X	\bar{Q}
X	X	0	1	1	0
—	1	1	X	Q	\bar{Q}
—	X	1	X	Q	\bar{Q}

X = Don't Care

CD4029BMS TOP VIEW

PRESET ENABLE	1	10	VDD
Q4	2	19	CLOCK
JAM 4	3	14	Q3
JAM 1	4	13	JAM 3
CARRY IN	5	12	JAM 2
Q1	6	11	Q2
CARRY OUT	7	10	UP/DOWN
VSS	8	9	BINARY/DECADE

Logic Diagram

43

43

ENSICAEN Digital Electronic | V. Logique séquentielle | Logique séquentielle – 3. Machines à états finis – 3.5. Compteurs (cont.)

Structure à éviter

Les signaux de sortie des bascules de cette structure ne sont pas synchrones. On ne peut pas s'en servir pour générer des signaux à cause des aléas (états transitoires non souhaités).

Exemple : on souhaite générer une impulsion toutes les 64 périodes (fin de comptage (cf Timer 0 du PIC18)).

44

44

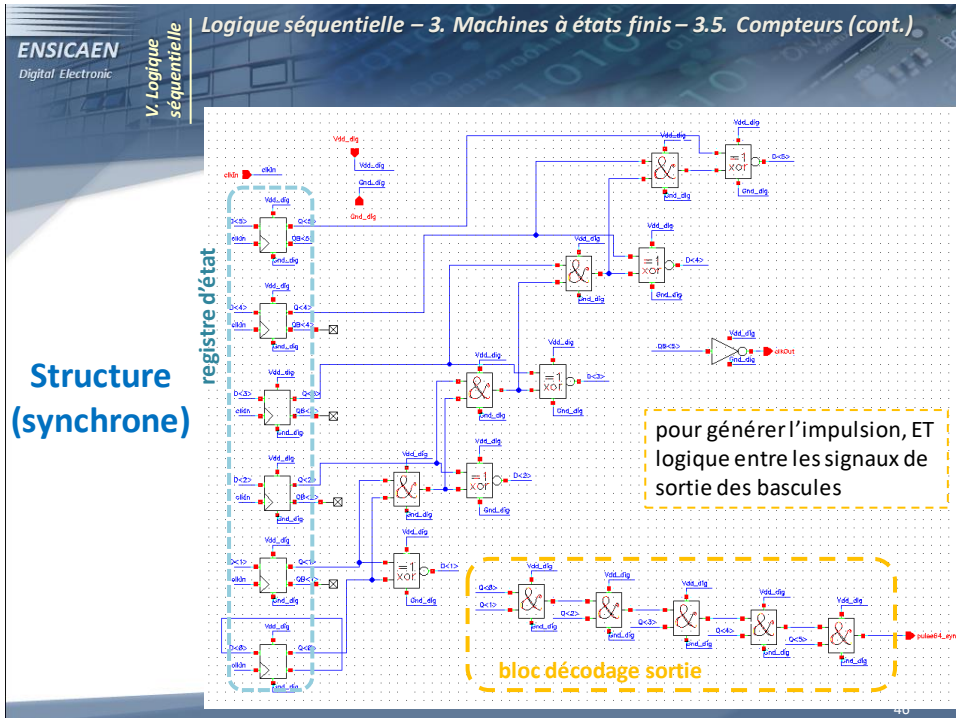
ENSICAEN Digital Electronic | V. Logique séquentielle | Logique séquentielle – 3. Machines à états finis – 3.5. Compteurs (cont.)

structure à éviter (asynchrone)

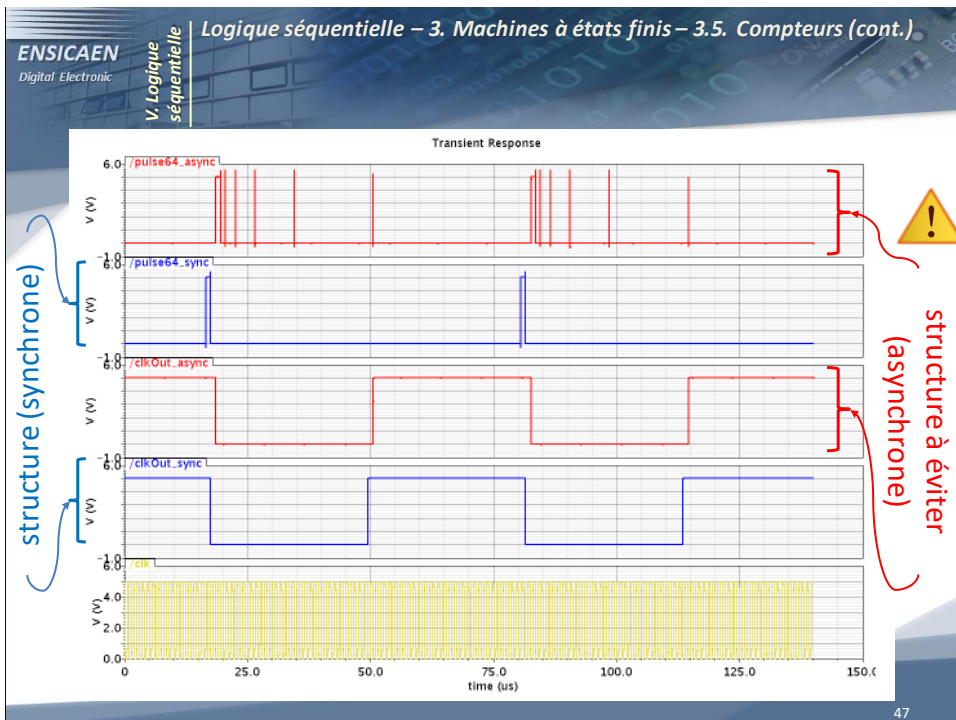
bloc décodage sortie

45

45



46



47

ENSICAEN Digital Electronic | V. Logique séquentielle | Logique séquentielle – 3. Machines à états finis

Diagramme d'état : notion plus large que l'électronique numérique, adaptée aux systèmes séquentiels, à la programmation.

48

48

ENSICAEN Digital Electronic | V. Logique séquentielle | Logique séquentielle – 3. Machines à états finis – 3.6. Dans les CPU/MCU

Dans les CPU/MCU:

- Compteurs/décompteurs à préchargement**
 - Compteur programme
 - Timer (signal de fin de comptage/décomptage, IT)
- Séquenceur**

Les machines à états finis sont une des structures pour réaliser des **séquenceurs (automates) dans les cœurs de calculateur**. L'autre structure est le séquenceur micro-programmé.

49

49

ENSICAEN Digital Electronic | V. Logique séquentielle | Logique séquentielle – 4. Séquenceur micro-programmé

4. Séquenceur micro-programmé

possibilités d'évolution restreintes, moins flexible, moins grande efficacité
branchement conditionnel, incrémentation implicite

si la condition est vraie, on charge un nouvel état
n : si (condition) aller à m
n+1 : si.....

50

ENSICAEN Digital Electronic | V. Logique séquentielle | Logique séquentielle – 4. Séquenceur micro-programmé (cont.)

Exemple simple : compteur-décompteur, fonctionnement naturel compteur

a => CO

état	Instruction	sorties		micro-instruction		
		S_1S_0	C_0	A_1A_0	S_1S_0	
0	-----	00	0	XX	00	
1	Si a aller en 0	00	1	00	00	
2	Si a aller en 1	11	1	01	11	
3	Si a aller en 2	10	1	10	10	

51

ENSICAEN Digital Electronic | V. Logique séquentielle | Logique séquentielle – 4. Séquenceur micro-programmé (cont.)

Exemple : (cont.)

52

52

ENSICAEN Digital Electronic | V. Logique séquentielle | Logique séquentielle – 4. Séquenceur micro-programmé (cont.)

PROSE : PROgrammable SEquencer

53

53

ENSICAEN Digital Electronic | V. Logique séquentielle | Logique séquentielle – 5. Séquenceur

Séquenceur

Le séquenceur coordonne les sous-opérations nécessaires pour l'exécution des instructions.

solution câblée :
l'automate du séquenceur est une machine à états finis câblée.
privilegiée pour les processeurs RISC (*Reduced Instruction Set Core/Computer*);
exemple PIC

solution micro-programmée :
l'automate est un séquenceur micro-programmé (microprogramme contenu dans une ROM)
privilegiée pour les processeurs CISC (*Complex Instruction Set Computer*)

54

54

ENSICAEN Digital Electronic | V. Logique séquentielle | Logique séquentielle. 6. Point mémoire

Point mémoire statique (bistable)

Point mémoire dynamique

Mémoire flash

grille de commande G grille flottante S

Note

transistor NMOS

55

55

Logique séquentielle. 6. Point mémoire

ENSICAEN Digital Electronic

V. Logique séquentielle

Inventée par Toshiba en 1980

État de la grille flottante	État logique
Chargée (programmée)	« 0 » - bas
Non chargée (effacée)	« 1 » - haut

Mémoire flash

State 1 - No Charge

State 2 - Lightly Charged

State 3 - Medium Charge

State 4 - Highly Charged

V_{Total}

SLC NAND	MLC NAND	TLC NAND
1	11	111
	10	110
	01	101
	00	100
0	011	010
	001	000

SLC 1-Bit/Cell MLC 2-Bits/Cell TLC 3-Bits/Cell

Voltage Allocated to each State based on NAND Flash Technology

<https://www.cactus-tech.com/resources/blog/details/solid-state-drive-primer-2-slc-mlc-and-tlc-nand-flash/>

MLC (Multi Level Cell) TLC (Tri Level Cell)

56

Logique séquentielle. 6. Point mémoire

ENSICAEN Digital Electronic

V. Logique séquentielle

Mémoire flash

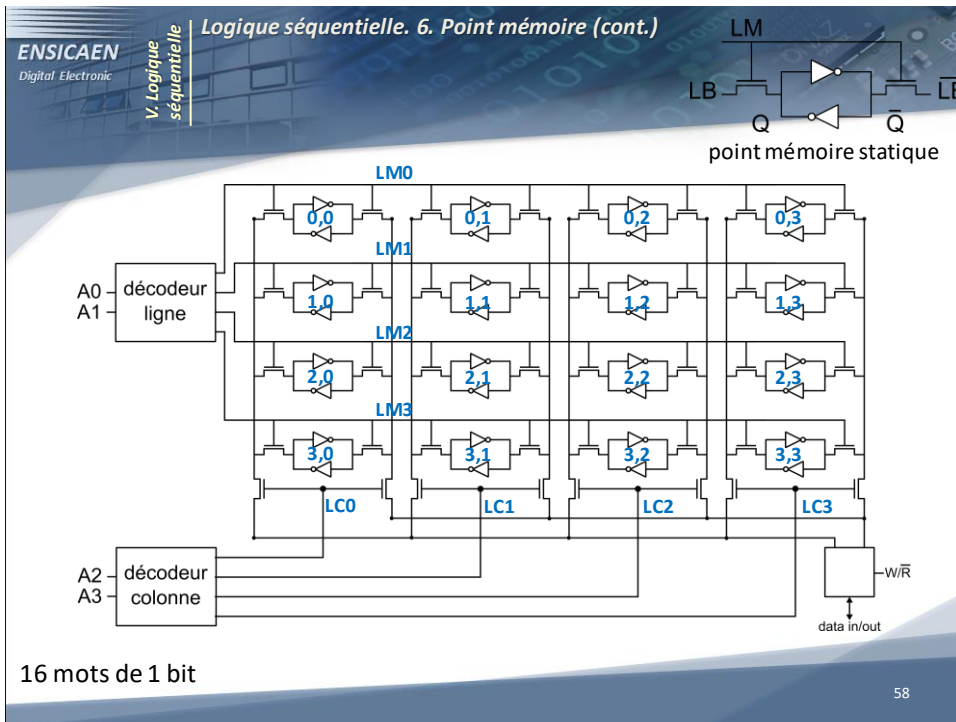
Flash NAND : écriture, lecture, effacement par blocs. Dans les clefs USB, disques durs, smartphones.

Transistors en série. Bit line à «0» quand toutes les Word lines à «1»

Flash NOR : écriture, lecture, effacement par mots. Stockage du code exécutable dans les microcontrôleurs, mémoire de démarrage (Boot ROM), firmware et BIOS (Basic Input/Output System).

«1» sur Word line programme un «0»

57



58

A l'issue de cette cinquième partie (1/2)

ENSICAEN Digital Electronic | V. Logique séquentielle

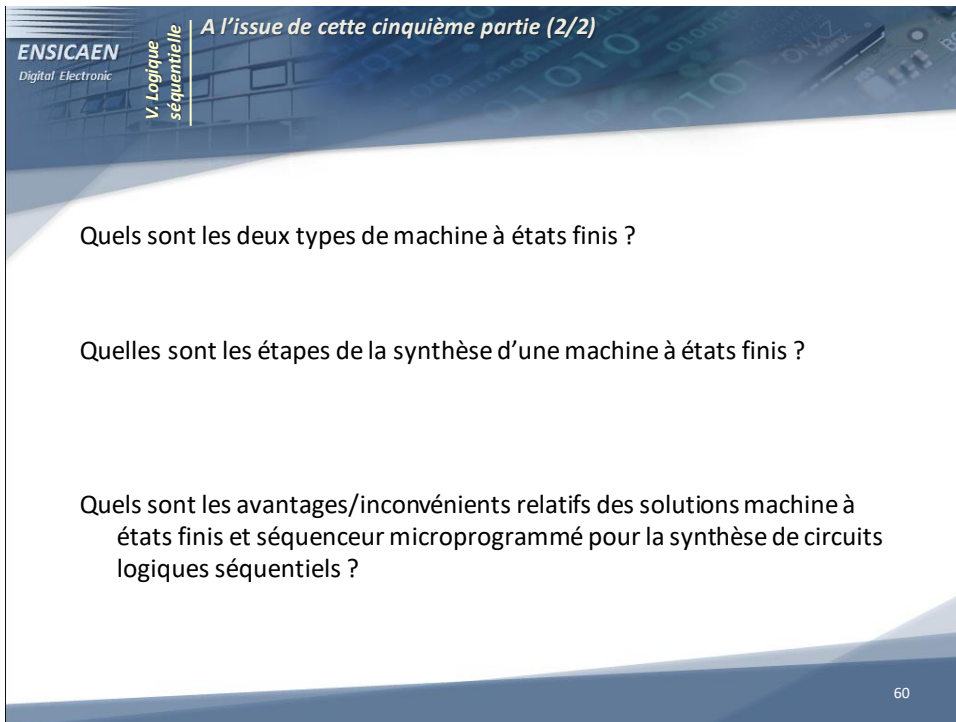
?

Quelle différence essentielle existe-t-il entre un système asynchrone et un système synchrone ?

Quels sont les avantages/inconvénients respectifs des systèmes asynchrones et synchrones ?

De quel type sont la grande majorité des microprocesseurs et microcontrôleurs (synchrone ou asynchrone) ?

59



ENSICAEN
Digital Electronic

V. Logique séquentielle

A l'issue de cette cinquième partie (2/2)

Quels sont les deux types de machine à états finis ?

Quelles sont les étapes de la synthèse d'une machine à états finis ?

Quels sont les avantages/inconvénients relatifs des solutions machine à états finis et séquenceur microprogrammé pour la synthèse de circuits logiques séquentiels ?

60

60



ENSICAEN
Digital Electronic

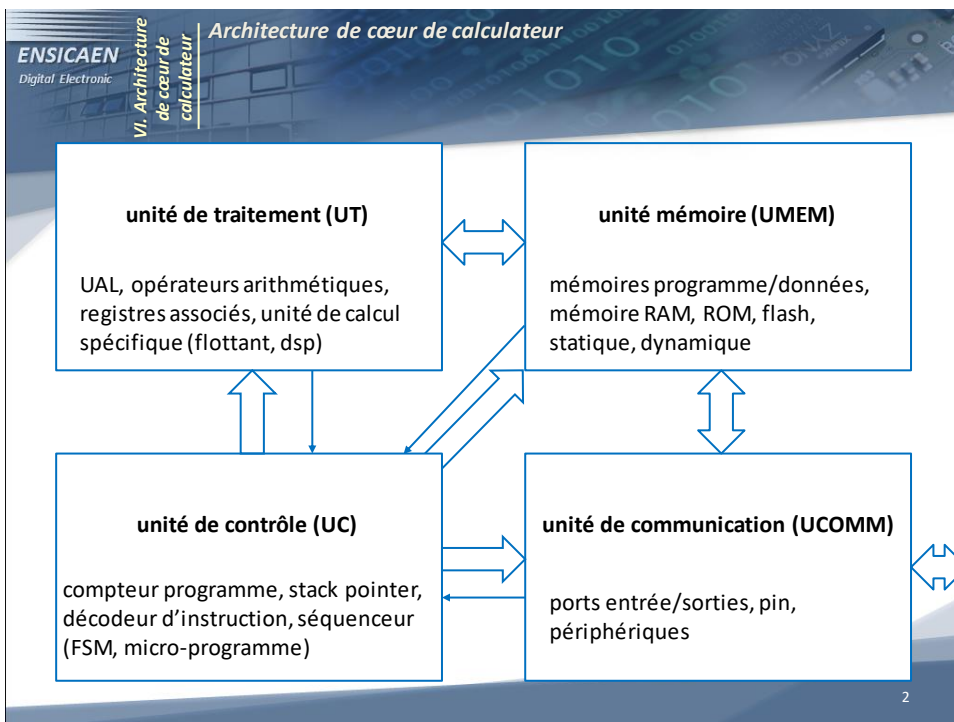
VI. Architecture de cœur de calculateur

UT (UAL), UC, Umem, Ucomm

M. Denoual

1

1



2



ENSICAEN
Digital Electronic

VI. Architecture de cœur de calculateur

Architecture PIC18

Vue générale

compteur programme

multiplexeur

séquenceur (FSM)

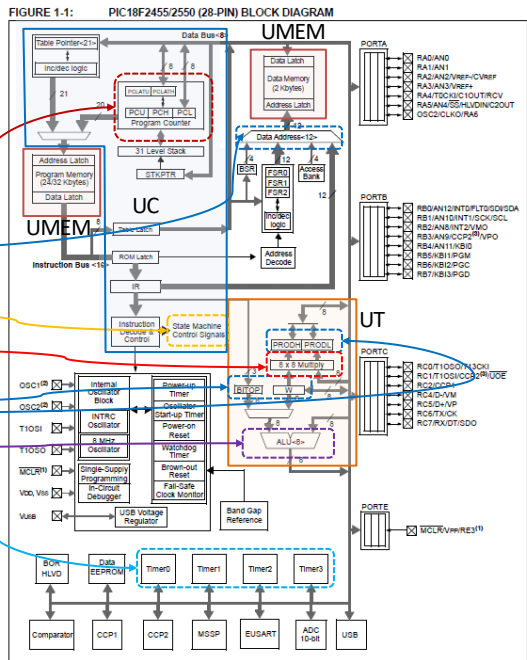
multiplieur

registres

ALU

compteurs

PIC18F2455/2550/4455/4550



3

ENSICAEN
Digital Electronic

VI. Architecture de cœur de calculateur

Architecture PIC18

Timer 0

fréquence d'horloge configurable

registres de préchargement

4



ENSICAEN
Digital Electronic

VI. Architecture de cœur de calculateur

Architecture PIC18

Multiplication

PIC18F2455/2550/4455/4550

8.0 8 x 8 HARDWARE MULTIPLIER

8.1 Introduction

All PIC18 devices include an 8 x 8 hardware multiplier as part of the ALU. The multiplier performs an unsigned operation and yields a 16-bit result that is stored in the product register pair, PRODH:PRODL. The multiplier's operation does not affect any flags in the Status register.

Making multiplication a hardware operation allows it to be completed in a single instruction cycle. This has the advantages of higher computational throughput and reduced code size for multiplication algorithms and allows the PIC18 devices to be used in many applications previously reserved for digital signal processors. A comparison of various hardware and software multiply operations, along with the savings in memory and execution time, is shown in Table 8-1.

8.2 Operation

Example 8-1 shows the instruction sequence for an 8 x 8 unsigned multiplication. Only one instruction is required when one of the arguments is already loaded in the WREG register.

Example 8-2 shows the sequence to do an 8 x 8 signed multiplication. To account for the sign bits of the arguments, each argument's Most Significant bit (MSB) is tested and the appropriate subtractions are done.

EXAMPLE 8-1: 8 x 8 UNSIGNED MULTIPLY ROUTINE

```

MOVWF ARG1, W      ; ARG1 * ARG2 ->
MULWF ARG2         ; PRODH:PRODL
                    ; PRODH:PRODL
                    ;
                    ;

```

EXAMPLE 8-2: 8 x 8 SIGNED MULTIPLY ROUTINE

```

MOVWF ARG1, W      ; ARG1 * ARG2 ->
MULWF ARG2         ; PRODH:PRODL
                    ; PRODH:PRODL
BTFSRC ARG2, SB    ; Test Sign Bit
SUBWF PRODH, F     ; PRODH = PRODH
                    ;
                    ;
MOVWF ARG2, W      ; ARG1 * ARG2 ->
BTFSRC ARG1, SB    ; Test Sign Bit
SUBWF PRODL, F     ; PRODL = PRODL
                    ;
                    ;

```

TABLE 8-1: PERFORMANCE COMPARISON FOR VARIOUS MULTIPLY OPERATIONS

Routine	Multiply Method	Program Memory (Words)	Cycles (Max)	Time		
				@ 40 MHz	@ 10 MHz	@ 4 MHz
8 x 8 unsigned	Without hardware multiply	13	69	6.9 μs	27.6 μs	69 μs
	Hardware multiply	1	1	100 ns	400 ns	1 μs
8 x 8 signed	Without hardware multiply	33	91	9.1 μs	36.4 μs	91 μs
	Hardware multiply	6	6	600 ns	2.4 μs	6 μs
16 x 16 unsigned	Without hardware multiply	21	242	24.2 μs	96.8 μs	242 μs
	Hardware multiply	28	28	2.8 μs	11.2 μs	28 μs
16 x 16 signed	Without hardware multiply	52	254	25.4 μs	102.6 μs	254 μs
	Hardware multiply	35	40	4.0 μs	16.0 μs	40 μs

5

ENSICAEN
Digital Electronic

VI. Architecture de cœur de calculateur

Architecture dsPIC30

Cœur

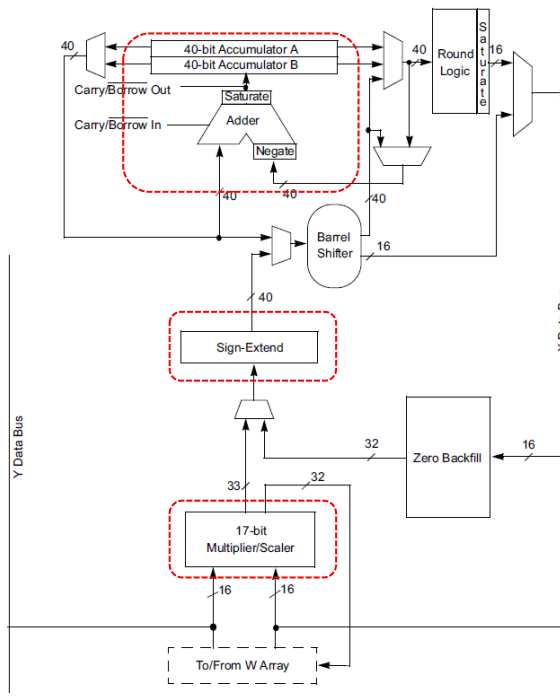
A DSP engine has been included to significantly enhance the core arithmetic capability and throughput. It features a high speed 17-bit by 17-bit multiplier, a 40-bit ALU, two 40-bit saturating accumulators and a 40-bit bidirectional barrel shifter. Data in the accumulator or any working register can be shifted up to 15 bits right, or 16 bits left in a single cycle. The DSP instructions operate seamlessly with all other instructions and have been designed for optimal real-time performance. The MAC class of instructions can concurrently fetch two data operands from memory while multiplying two W registers. To enable this concurrent fetching of data operands, the data space has been split for these instructions and linear for all others. This has been achieved in a transparent and flexible manner, by dedicating certain working registers to each address space for the MAC class of instructions.

6



ENSICAEN Digital Electronic VI. Architecture de cœur de calculateur Architecture dsPIC

DSP engine



7

ENSICAEN Digital Electronic VI. Architecture de cœur de calculateur Architecture dsPIC30

Multiplication

dsPIC30F

2.5.1 MULTIPLIER

The 17 x 17-bit multiplier is capable of signed or unsigned operation and can multiplex its output using a scaler to support either 1.31 fractional (Q31) or 32-bit integer results. The respective number representation formats are shown in Figure 2-10. Unsigned operands are zero-extended into the 17th bit of the multiplier input value. Signed operands are sign-extended into the 17th bit of the multiplier input value. The output of the 17 x 17-bit multiplier/scaler is a 33-bit value which is sign-extended to 40 bits. Integer data is inherently represented as a signed two's complement value, where the MSB is defined as a sign bit. Generally speaking, the range of an N-bit two's complement integer is -2^{N-1} to $2^{N-1} - 1$. For a 16-bit integer, the data range is -32768 (0x8000) to 32767 (0x7FFF) including '0' (see Figure 2-10). For a 32-bit integer, the data range is -2,147,483,648 (0x8000 0000) to 2,147,483,645 (0x7FFF FFFF).

When the multiplier is configured for fractional multiplication, the data is represented as a two's complement fraction, where the MSB is defined as a sign bit and the radix point is implied to lie just after the sign bit (QX format). The range of an N-bit two's complement fraction with this implied radix point is -1.0 to $(1 - 2^{1-N})$. For a 16-bit fraction, the Q15 data range is -1.0 (0x8000) to 0.999969482 (0x7FFF) including '0' and has a precision of 3.01518×10^{-5} . In Fractional mode, the 16x16 multiply operation generates a 1.31 product which has a precision of 4.65661×10^{-10} .

FIGURE 2-10: 16-BIT INTEGER AND FRACTIONAL MODES

Different Representations of 0x4001																
Integer:																
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
2^0	2^{14}	2^{13}	2^{12}	2^{11}	...											2^0
$0x4001 = 2^{14} + 2^0 = 16385$																
1.15 Fractional:																
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
-2^0	2^1	2^2	2^3	...												2^{15}
$0x4001 = 2^1 + 2^{15} = 0.500030518$																

8



ENSICAEN Digital Electronic VI. Architecture de cœur de calculateur Architecture dsPIC30

Multiplication

Assembly Syntax	Description	# of Words	# of Cycles	Status Flags Affected
MPY Wm*Wn,Acc,Wx,Wxd,Wy,Wyd	Multiply Wm by Wn to Accumulator	1	1	OA,OB,OAB,SA,SB,SAB
MPY Wm*Wm,Acc,Wx,Wxd,Wy,Wyd	Square Wm to Accumulator	1	1	OA,OB,OAB,SA,SB,SAB
MPY.N Wm*Wn,Acc,Wx,Wxd,Wy,Wyd	-(Multiply Wm by Wn) to Accumulator	1	1	None
MSC Wm*Wm,Acc,Wx,Wxd,Wy,Wyd,AWB	Multiply and Subtract from Accumulator	1	1	OA,OB,OAB,SA,SB,SAB
MUL.SS Wb,Ws,Wnd	{Wnd+1, Wnd} = signed(Wb) * signed(Ws)	1	1	None
MUL.SU Wb,Ws,Wnd	{Wnd+1, Wnd} = signed(Wb) * unsigned(Ws)	1	1	None
MUL.US Wb,Ws,Wnd	{Wnd+1, Wnd} = unsigned(Wb) * signed(Ws)	1	1	None
MUL.UU Wb,Ws,Wnd	{Wnd+1, Wnd} = unsigned(Wb) * unsigned(Ws)	1	1	None
MUL.SU Wb,#lit5,Wnd	{Wnd+1, Wnd} = signed(Wb) * unsigned(lit5)	1	1	None
MUL.UU Wb,#lit5,Wnd	{Wnd+1, Wnd} = unsigned(Wb) * unsigned(lit5)	1	1	None
MUL f	W3:W2 = f * WREG	1	1	None

9

9

ENSICAEN Digital Electronic VI. Architecture de cœur de calculateur Conclusion, électronique numérique

solutions spécifiques/standards

circuit intégré dédié

- ASIC : Application Specific Integrated Circuit
- SoC : System-on-Chip
- Sip : System-in-Package

architecture à CPU

- microprocesseur
- microcontrôleur
- DSP

composants discrets

très petites applications

composants à logique programmable

PLD, FPGA

System-on-chip programmable

PSoC

à l'intégrateur de tout cela, il y a des circuits logiques et vous avez vu comment c'est fait!

standard (1957), composants discrets (1957), mémoires μP (1977), circuits spécifiques (ASIC) (1987), FPGA (1997), P-SoC/P-SiP (2017)

spécifique (onde Makimoto), composants spécifiques pour montre, TV, calculatrices (LSI), SoC/SiP

10

10