2E200 : Electronique Numérique, Combinatoire et Séquentielle

Bertrand Granado

Licence E^{2}A

Hiver 2019

Plan

1. Introduction : L'électronique numérique à l'aube de 2020 / Méthodes et outils de Conception des systèmes numériques
2. Algèbre de Boole
3. Codage
4. Les composants combinatoire simples
5. Les composants combinatoires complexes
6. Les composants séquentiels : les bascules
7. Les composants séquentiels : les registres
8. Les composants séquentiels : les compteurs / Le traitement Pipeliné
9. Interface avec l'environnement continu : Conversion Analogique vers Numérique et Numérique vers Analogique

Plan

1. Introduction : L'électronique numérique à l'aube de 2020 / Méthodes et outils de Conception des systèmes numériques
2. Algèbre de Boole
3. Codage
4. Les composants combinatoire simples
5. Les composants combinatoires complexes
6. Les composants séquentiels : les bascules
7. Les composants séquentiels : les registres
8. Les composants séquentiels : les compteurs / Le traitement Pipeliné

SCIENCES SORBONNE UNIVERSITÉ

Plan

1. Introduction : L'électronique numérique à l'aube de 2020 / Méthodes et outils de Conception des systèmes numériques
2. Algèbre de Boole
3. Codage
4. Les composants combinatoire simples
5. Les composants combinatoires complexes
6. Les composants séquentiels : les bascules
7. Les composants séquentiels : les registres
8. Les composants séquentiels : les compteurs / Le traitement Pipeliné

Plan

1. Introduction : L'électronique numérique à l'aube de 2020 / Méthodes et outils de Conception des systèmes numériques

2. Algèbre de Boole

3. Codage
   - Codage Binaire - Codage Décimal
   - Codage Héxadécimal
   - Codage Décimal Codé Binaire
   - Codage de Gray
   - Codage ASCII
   - Codage des nombres signés : le complément à 2
   - Codage Virgule Fixe
   - Codage Flottant
   - Analyse de l'erreur

4. Les composants combinatoire simples

Codage

- Système de base : codage décimal

Codage

- Système de base : codage décimal
- Conversion décimal-binaire et binaire-décimal

Codage

- Système de base : codage décimal
- Conversion décimal-binaire et binaire-décimal
- $\nexists n \Rightarrow 2^n = 10$, nécessité codage octal ou héxadécimal

Codage

- Système de base : codage décimal
- Conversion décimal-binaire et binaire-décimal
- $\nexists n \Rightarrow 2^n = 10$, nécessité codage octal ou héxadécimal
- Codage DCB : Décimal Codé Binaire

Codage

- Système de base : codage décimal
- Conversion décimal-binaire et binaire-décimal
- $\nexists n \Rightarrow 2^n = 10$, nécessité codage octal ou héxadécimal
- Codage DCB : Décimal Codé Binaire
- Code de Gray ou binaire réfléchi

Codage

- Système de base : codage décimal
- Conversion décimal-binaire et binaire-décimal
- $\nexists n \Rightarrow 2^n = 10$, nécessité codage octal ou héxadécimal
- Codage DCB : Décimal Codé Binaire
- Code de Gray ou binaire réfléchi
- Code ASCII

Conversion binaire-décimal

- La conversion binaire-décimal s'effectue simplement en réalisant la somme des bits pondérés par leur position

Conversion binaire-décimal

- La conversion binaire-décimal s'effectue simplement en réalisant la somme des bits pondérés par leur position
- $\sum_{i=0}^{n-1} b_i * 2^i$ où b_i est la valeur du bit de position i

Conversion binaire-décimal

- La conversion binaire-décimal s'effectue simplement en réalisant la somme des bits pondérés par leur position
- $\sum_{i=0}^{n-1} b_i * 2^i$ où b_i est la valeur du bit de position i

Nombre Binaire	1	0	1	1	0
Position du bit					
Nombre Décimal					

Conversion binaire-décimal

- La conversion binaire-décimal s'effectue simplement en réalisant la somme des bits pondérés par leur position
- $\sum_{i=0}^{n-1} b_i * 2^i$ où b_i est la valeur du bit de position i

Nombre Binaire	1	0	1	1	0
Position du bit	4				
Nombre Décimal					

Conversion binaire-décimal

- La conversion binaire-décimal s'effectue simplement en réalisant la somme des bits pondérés par leur position
- $\sum_{i=0}^{n-1} b_i * 2^i$ où b_i est la valeur du bit de position i

Nombre Binaire	1	0	1	1	0
Position du bit	4	3			
Nombre Décimal					

Conversion binaire-décimal

- La conversion binaire-décimal s'effectue simplement en réalisant la somme des bits pondérés par leur position
- $\sum_{i=0}^{n-1} b_i * 2^i$ où b_i est la valeur du bit de position i

Nombre Binaire	1	0	1	1	0
Position du bit	4	3	2		
Nombre Décimal					

Conversion binaire-décimal

- La conversion binaire-décimal s'effectue simplement en réalisant la somme des bits pondérés par leur position
- $\sum_{i=0}^{n-1} b_i * 2^i$ où b_i est la valeur du bit de position i

Nombre Binaire	1	0	1	1	0
Position du bit	4	3	2	1	
Nombre Décimal					

Conversion binaire-décimal

- La conversion binaire-décimal s'effectue simplement en réalisant la somme des bits pondérés par leur position
- $\sum_{i=0}^{n-1} b_i * 2^i$ où b_i est la valeur du bit de position i

Nombre Binaire	1	0	1	1	0
Position du bit	4	3	2	1	0
Nombre Décimal					

Conversion binaire-décimal

- La conversion binaire-décimal s'effectue simplement en réalisant la somme des bits pondérés par leur position
- $\sum_{i=0}^{n-1} b_i * 2^i$ où b_i est la valeur du bit de position i

Nombre Binaire	1	0	1	1	0
Position du bit	4	3	2	1	0
Nombre Décimal	2^4				

Conversion binaire-décimal

- La conversion binaire-décimal s'effectue simplement en réalisant la somme des bits pondérés par leur position
- $\sum_{i=0}^{n-1} b_i * 2^i$ où b_i est la valeur du bit de position i

Nombre Binaire	1	0	1	1	0
Position du bit	4	3	2	1	0
Nombre Décimal	2^4	0			

Conversion binaire-décimal

- La conversion binaire-décimal s'effectue simplement en réalisant la somme des bits pondérés par leur position
- $\sum_{i=0}^{n-1} b_i * 2^i$ où b_i est la valeur du bit de position i

Nombre Binaire	1	0	1	1	0
Position du bit	4	3	2	1	0
Nombre Décimal	2^4	0	2^2		

Conversion binaire-décimal

- La conversion binaire-décimal s'effectue simplement en réalisant la somme des bits pondérés par leur position
- $\sum_{i=0}^{n-1} b_i * 2^i$ où b_i est la valeur du bit de position i

Nombre Binaire	1	0	1	1	0
Position du bit	4	3	2	1	0
Nombre Décimal	2^4	0	2^2	2^1	

Conversion binaire-décimal

- La conversion binaire-décimal s'effectue simplement en réalisant la somme des bits pondérés par leur position
- $\sum_{i=0}^{n-1} b_i * 2^i$ où b_i est la valeur du bit de position i

Nombre Binaire	1	0	1	1	0
Position du bit	4	3	2	1	0
Nombre Décimal	2^4	0	2^2	2^1	0

Conversion binaire-décimal

- La conversion binaire-décimal s'effectue simplement en réalisant la somme des bits pondérés par leur position
- $\sum_{i=0}^{n-1} b_i * 2^i$ où b_i est la valeur du bit de position i

Nombre Binaire	1	0	1	1	0
Position du bit	4	3	2	1	0
Nombre Décimal	2^4	0	2^2	2^1	0

$= 16 + 0 + 4 + 2 + 0 = 22$

Conversion décimal-binaire

- La conversion décimal-binaire peut s'effectuer en utilisant la méthode inverse de celle énoncée précédemment. *Exemples*. Fastidieux pour de grand nombre.

Conversion décimal-binaire

- La conversion décimal-binaire peut s'effectuer en utilisant la méthode inverse de celle énoncée précédemment. *Exemples*. Fastidieux pour de grand nombre.
- Réalise un division par 2

Conversion décimal-binaire

- La conversion décimal-binaire peut s'effectuer en utilisant la méthode inverse de celle énoncée précédemment. *Exemples*. Fastidieux pour de grand nombre.
- Réalise un division par 2

29

Conversion décimal-binaire

- La conversion décimal-binaire peut s'effectuer en utilisant la méthode inverse de celle énoncée précédemment. *Exemples*. Fastidieux pour de grand nombre.
- Réalise un division par 2

$$\begin{array}{r|l} 29 & 2 \\ \hline \end{array}$$

Conversion décimal-binaire

- La conversion décimal-binaire peut s'effectuer en utilisant la méthode inverse de celle énoncée précédemment. *Exemples*. Fastidieux pour de grand nombre.
- Réalise un division par 2

$$\begin{array}{r|l} 29 & 2 \\ \hline 1 & 14 \end{array}$$

Conversion décimal-binaire

- La conversion décimal-binaire peut s'effectuer en utilisant la méthode inverse de celle énoncée précédemment. *Exemples*. Fastidieux pour de grand nombre.
- Réalise un division par 2

$$\begin{array}{r|l|l} 29 & 2 & \\ \cline{2-2} 1 & 14 & 2 \\ \cline{3-3} \end{array}$$

Conversion décimal-binaire

- La conversion décimal-binaire peut s'effectuer en utilisant la méthode inverse de celle énoncée précédemment. *Exemples*. Fastidieux pour de grand nombre.
- Réalise un division par 2

$$\begin{array}{r|l}
29 & 2 \\ \hline
1 & 14 \;\big|\; 2 \\
 & 0 \quad\;\; 7
\end{array}$$

Conversion décimal-binaire

- La conversion décimal-binaire peut s'effectuer en utilisant la méthode inverse de celle énoncée précédemment. *Exemples*. Fastidieux pour de grand nombre.
- Réalise un division par 2

$$\begin{array}{cc|cc|c}
29 & 2 & & & \\
\hline
1 & 14 & 2 & & \\
\cline{2-4}
 & 0 & 7 & 2 & \\
\cline{4-4}
\end{array}$$

Conversion décimal-binaire

- La conversion décimal-binaire peut s'effectuer en utilisant la méthode inverse de celle énoncée précédemment. *Exemples*. Fastidieux pour de grand nombre.
- Réalise un division par 2

$$
\begin{array}{cccc}
29 & \underline{\ 2\ } & & \\
1 & 14 & \underline{\ 2\ } & \\
 & 0 & 7 & \underline{\ 2\ } \\
 & & 1 & 3
\end{array}
$$

Conversion décimal-binaire

- La conversion décimal-binaire peut s'effectuer en utilisant la méthode inverse de celle énoncée précédemment. *Exemples*. Fastidieux pour de grand nombre.
- Réalise un division par 2

29	2			
1	14	2		
	0	7	2	
		1	3	2

Conversion décimal-binaire

- La conversion décimal-binaire peut s'effectuer en utilisant la méthode inverse de celle énoncée précédemment. *Exemples*. Fastidieux pour de grand nombre.
- Réalise un division par 2

29	2			
1	14	2		
	0	7	2	
		1	3	2
			1	1

Conversion décimal-binaire

- La conversion décimal-binaire peut s'effectuer en utilisant la méthode inverse de celle énoncée précédemment. *Exemples*. Fastidieux pour de grand nombre.
- Réalise un division par 2

29	2				
1	14	2			
	0	7	2		
		1	3	2	
			1	1	2

Conversion décimal-binaire

- La conversion décimal-binaire peut s'effectuer en utilisant la méthode inverse de celle énoncée précédemment. *Exemples*. Fastidieux pour de grand nombre.
- Réalise un division par 2

$$
\begin{array}{cccccc}
29 & \underline{\;2\;} & & & & \\
1 & 14 & \underline{\;2\;} & & & \\
 & 0 & 7 & \underline{\;2\;} & & \\
 & & 1 & 3 & \underline{\;2\;} & \\
 & & & 1 & 1 & \underline{\;2\;} \\
 & & & & 1 & 0
\end{array}
$$

Conversion décimal-binaire

- La conversion décimal-binaire peut s'effectuer en utilisant la méthode inverse de celle énoncée précédemment. *Exemples*. Fastidieux pour de grand nombre.
- Réalise un division par 2

29	2				
1	14	2			
	0	7	2		
		1	3	2	
			1	1	2
				1	0

Nombre binaire = 11101

Codage Hexadécimal

- Travaille avec des quartets binaires : 1010

Codage Hexadécimal

- Travaille avec des quartets binaires : 1010
- Intéressant la taille du mot binaire de base est l'octet

Codage Hexadécimal

- Travaille avec des quartets binaires : 1010
- Intéressant la taille du mot binaire de base est l'octet
- Un octet = Deux Quartets

Codage Hexadécimal

- La base du système Héxadécimal est la base 16

Codage Hexadécimal

- La base du système Héxadécimal est la base 16
- Il faut donc 16 symboles

Codage Hexadécimal

- La base du système Héxadécimal est la base 16
- Il faut donc 16 symboles
- 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F

Codage Hexadécimal

Hexa	Décimal	Binaire	Hexa	Décimal	Binaire
0	0	0000	8	8	1000
1	1	0001	9	9	1001
2	2	0010	A	10	1010
3	3	0011	B	11	1011
4	4	0100	C	12	1100
5	5	0101	D	13	1101
6	6	0110	E	14	1110
7	7	0111	F	15	1111

Conversion Hexadécimal-Décimal

- De même que pour la conversion binaire-décimal, il s’agit ici de faire une sommation pondéré des symboles

Conversion Hexadécimal-Décimal

- De même que pour la conversion binaire-décimal, il s'agit ici de faire une sommation pondéré des symboles

Nombre Hexadécimal	A	2	E
Position du symbole	2		
Puissance associée			
Nombre Décimal			

Conversion Hexadécimal-Décimal

- De même que pour la conversion binaire-décimal, il s'agit ici de faire une sommation pondéré des symboles

Nombre Hexadécimal	A	2	E
Position du symbole	2	1	
Puissance associée			
Nombre Décimal			

Conversion Hexadécimal-Décimal

- De même que pour la conversion binaire-décimal, il s'agit ici de faire une sommation pondéré des symboles

Nombre Hexadécimal	A	2	E
Position du symbole	2	1	0
Puissance associée			
Nombre Décimal			

Conversion Hexadécimal-Décimal

- De même que pour la conversion binaire-décimal, il s'agit ici de faire une sommation pondéré des symboles

Nombre Hexadécimal	A	2	E
Position du symbole	2	1	0
Puissance associée	16^2		
Nombre Décimal			

Conversion Hexadécimal-Décimal

- De même que pour la conversion binaire-décimal, il s'agit ici de faire une sommation pondéré des symboles

Nombre Hexadécimal	A	2	E
Position du symbole	2	1	0
Puissance associée	16^2	16^1	
Nombre Décimal			

Conversion Hexadécimal-Décimal

- De même que pour la conversion binaire-décimal, il s'agit ici de faire une sommation pondéré des symboles

Nombre Hexadécimal	A	2	E
Position du symbole	2	1	0
Puissance associée	16^2	16^1	16^0
Nombre Décimal			

Conversion Hexadécimal-Décimal

- De même que pour la conversion binaire-décimal, il s'agit ici de faire une sommation pondéré des symboles

Nombre Hexadécimal	A	2	E
Position du symbole	2	1	0
Puissance associée	16^2	16^1	16^0
Nombre Décimal	$10 * 16^2$		

Conversion Hexadécimal-Décimal

- De même que pour la conversion binaire-décimal, il s'agit ici de faire une sommation pondéré des symboles

Nombre Hexadécimal	A	2	E
Position du symbole	2	1	0
Puissance associée	16^2	16^1	16^0
Nombre Décimal	$10 * 16^2$	$+2 * 16^1$	

Conversion Hexadécimal-Décimal

- De même que pour la conversion binaire-décimal, il s'agit ici de faire une sommation pondéré des symboles

Nombre Hexadécimal	A	2	E
Position du symbole	2	1	0
Puissance associée	16^2	16^1	16^0
Nombre Décimal	$10 * 16^2$	$+2 * 16^1$	$+14 * 16^0$

Conversion Hexadécimal-Décimal

- De même que pour la conversion binaire-décimal, il s'agit ici de faire une sommation pondéré des symboles

Nombre Hexadécimal	A	2	E
Position du symbole	2	1	0
Puissance associée	16^2	16^1	16^0
Nombre Décimal	$10 * 16^2$	$+2 * 16^1$	$+14 * 16^0$
	$= 2606$		

Conversion Décimal-Hexadécimal

- De même que pour la conversion décimal-binaire on a recourt à la division

Conversion Décimal-Hexadécimal

- De même que pour la conversion décimal-binaire on a recourt à la division

311

Conversion Décimal-Hexadécimal

- De même que pour la conversion décimal-binaire on a recourt à la division

$$311 \quad \underline{\;16\;}$$

Conversion Décimal-Hexadécimal

- De même que pour la conversion décimal-binaire on a recourt à la division

$$\begin{array}{r|l} 311 & 16 \\ \cline{2-2} 7 & 19 \end{array}$$

Conversion Décimal-Hexadécimal

- De même que pour la conversion décimal-binaire on a recourt à la division

311	16	
7	19	16

Conversion Décimal-Hexadécimal

- De même que pour la conversion décimal-binaire on a recourt à la division

311	16	
7	19	16
	3	1

Conversion Décimal-Hexadécimal

- De même que pour la conversion décimal-binaire on a recourt à la division

$$\begin{array}{cccc} 311 & \underline{\;16\;} & & \\ 7 & 19 & \underline{\;16\;} & \\ & 3 & 1 & \underline{\;16\;} \end{array}$$

Conversion Décimal-Hexadécimal

- De même que pour la conversion décimal-binaire on a recourt à la division

$$\begin{array}{cccc} 311 & 16 & & \\ \cline{2-2} 7 & 19 & 16 & \\ \cline{3-3} & 3 & 1 & 16 \\ \cline{4-4} & & 1 & 0 \end{array}$$

Conversion Décimal-Hexadécimal

- De même que pour la conversion décimal-binaire on a recourt à la division

$$
\begin{array}{cc|cc}
311 & \underline{16} & & \\
7 & 19 & \underline{16} & \\
 & 3 & 1 & \underline{16} \\
 & & 1 & 0
\end{array}
$$

Conversion Décimal-Hexadécimal

- De même que pour la conversion décimal-binaire on a recourt à la division

$$\begin{array}{r|l}
311 & 16 \\ \hline
7 & 19 \quad \begin{array}{|l} 16 \\ \hline 1 \quad \begin{array}{|l} 16 \\ \hline 0 \end{array} \\ 1 \end{array} \\
 & 3
\end{array}$$

Nombre Héxadécimal = 137

Conversion Hexadécimal-Binaire

- Le nombre binaire est déduit en remplaçant chaque chiffre hexadécimal par son quartet binaire

Conversion Hexadécimal-Binaire

- Le nombre binaire est déduit en remplaçant chaque chiffre hexadécimal par son quartet binaire

Nombre Hexadécimal	E	3	B	1
Nombre Binaire				

Conversion Hexadécimal-Binaire

- Le nombre binaire est déduit en remplaçant chaque chiffre hexadécimal par son quartet binaire

Nombre Hexadécimal	E	3	B	1
Nombre Binaire	1110			

Conversion Hexadécimal-Binaire

- Le nombre binaire est déduit en remplaçant chaque chiffre hexadécimal par son quartet binaire

Nombre Hexadécimal	E	3	B	1
Nombre Binaire	1110	0011		

Conversion Hexadécimal-Binaire

- Le nombre binaire est déduit en remplaçant chaque chiffre hexadécimal par son quartet binaire

Nombre Hexadécimal	E	3	B	1
Nombre Binaire	1110	0011	1011	

Conversion Hexadécimal-Binaire

- Le nombre binaire est déduit en remplaçant chaque chiffre hexadécimal par son quartet binaire

Nombre Hexadécimal	E	3	B	1
Nombre Binaire	1110	0011	1011	0001

Conversion Binaire-Hexadécimal

- La méthode est l'inverse de la précédente

Conversion Binaire-Hexadécimal

- La méthode est l'inverse de la précédente
- on regroupe les bits par quartet et on remplace les quartets par leur équivalent hexadécimal.

Conversion Binaire-Hexadécimal

- La méthode est l'inverse de la précédente
- on regroupe les bits par quartet et on remplace les quartets par leur équivalent hexadécimal.

Nombre Binaire	0101	1010	1100	1011

Conversion Binaire-Hexadécimal

- La méthode est l'inverse de la précédente
- on regroupe les bits par quartet et on remplace les quartets par leur équivalent hexadécimal.

Nombre Binaire	0101	1010	1100	1011
Nombre Hexadécimal				

Conversion Binaire-Hexadécimal

- La méthode est l'inverse de la précédente
- on regroupe les bits par quartet et on remplace les quartets par leur équivalent hexadécimal.

Nombre Binaire	0101	1010	1100	1011
Nombre Hexadécimal	5			

Conversion Binaire-Hexadécimal

- La méthode est l'inverse de la précédente
- on regroupe les bits par quartet et on remplace les quartets par leur équivalent hexadécimal.

Nombre Binaire	0101	1010	1100	1011
Nombre Hexadécimal	5	A		

Conversion Binaire-Hexadécimal

- La méthode est l'inverse de la précédente
- on regroupe les bits par quartet et on remplace les quartets par leur équivalent hexadécimal.

Nombre Binaire	0101	1010	1100	1011
Nombre Hexadécimal	5	A	C	

Conversion Binaire-Hexadécimal

- La méthode est l'inverse de la précédente
- on regroupe les bits par quartet et on remplace les quartets par leur équivalent hexadécimal.

Nombre Binaire	0101	1010	1100	1011
Nombre Hexadécimal	5	A	C	B

Notations

- Les symboles 0,1 appartiennent au code binaire, décimal et hexadécimal

Notations

- Les symboles 0,1 appartiennent au code binaire, décimal et hexadécimal
- les symboles 0,1,2,3,4,5,6,7,8 et 9 appartiennent au code décimal et hexadécimal

Notations

- Les symboles 0,1 appartiennent au code binaire, décimal et hexadécimal
- les symboles 0,1,2,3,4,5,6,7,8 et 9 appartiennent au code décimal et hexadécimal
- Nécessité d'une convention d'écriture pour différencier

Notations

- Les symboles 0,1 appartiennent au code binaire, décimal et hexadécimal
- les symboles 0,1,2,3,4,5,6,7,8 et 9 appartiennent au code décimal et hexadécimal
- Nécessité d'une convention d'écriture pour différencier

Binaire	100
Décimal	100
Hexadécimal	100

Notations

- Les symboles 0,1 appartiennent au code binaire, décimal et hexadécimal
- les symboles 0,1,2,3,4,5,6,7,8 et 9 appartiennent au code décimal et hexadécimal
- Nécessité d'une convention d'écriture pour différencier

Binaire	100_B
Décimal	100
Hexadécimal	100

Notations

- Les symboles 0,1 appartiennent au code binaire, décimal et hexadécimal
- les symboles 0,1,2,3,4,5,6,7,8 et 9 appartiennent au code décimal et hexadécimal
- Nécessité d'une convention d'écriture pour différencier

Binaire	100_B
Décimal	100
Hexadécimal	100_H

Décimal Codé Binaire : DCB

- Remplacer chaque chiffre d'un nombre décimal par son équivalent binaire

Décimal Codé Binaire : DCB

- Remplacer chaque chiffre d'un nombre décimal par son équivalent binaire
- Faire une correspondance directe entre binaire et décimal

Décimal Codé Binaire : DCB

- Remplacer chaque chiffre d'un nombre décimal par son équivalent binaire
- Faire une correspondance directe entre binaire et décimal

Nombre Décimal	5	3	7	1
Nombre Binaire				

Décimal Codé Binaire : DCB

- Remplacer chaque chiffre d'un nombre décimal par son équivalent binaire
- Faire une correspondance directe entre binaire et décimal

Nombre Décimal	5	3	7	1
Nombre Binaire	0101			

Décimal Codé Binaire : DCB

- Remplacer chaque chiffre d'un nombre décimal par son équivalent binaire
- Faire une correspondance directe entre binaire et décimal

Nombre Décimal	5	3	7	1
Nombre Binaire	0101	0011		

Décimal Codé Binaire : DCB

- Remplacer chaque chiffre d'un nombre décimal par son équivalent binaire
- Faire une correspondance directe entre binaire et décimal

Nombre Décimal	5	3	7	1
Nombre Binaire	0101	0011	0111	

Décimal Codé Binaire : DCB

- Remplacer chaque chiffre d'un nombre décimal par son équivalent binaire
- Faire une correspondance directe entre binaire et décimal

Nombre Décimal	5	3	7	1
Nombre Binaire	0101	0011	0111	0001

Décimal Codé Binaire : DCB

- Remplacer chaque chiffre d'un nombre décimal par son équivalent binaire
- Faire une correspondance directe entre binaire et décimal

Nombre Décimal	5	3	7	1
Nombre Binaire	0101	0011	0111	0001
Nombre Binaire	0101	1001	1000	0011
Nombre Décimal				

Décimal Codé Binaire : DCB

- Remplacer chaque chiffre d'un nombre décimal par son équivalent binaire
- Faire une correspondance directe entre binaire et décimal

Nombre Décimal	5	3	7	1
Nombre Binaire	0101	0011	0111	0001
Nombre Binaire	0101	1001	1000	0011
Nombre Décimal	5			

Décimal Codé Binaire : DCB

- Remplacer chaque chiffre d'un nombre décimal par son équivalent binaire
- Faire une correspondance directe entre binaire et décimal

Nombre Décimal	5	3	7	1
Nombre Binaire	0101	0011	0111	0001
Nombre Binaire	0101	1001	1000	0011
Nombre Décimal	5	9		

Décimal Codé Binaire : DCB

- Remplacer chaque chiffre d'un nombre décimal par son équivalent binaire
- Faire une correspondance directe entre binaire et décimal

Nombre Décimal	5	3	7	1
Nombre Binaire	0101	0011	0111	0001
Nombre Binaire	0101	1001	1000	0011
Nombre Décimal	5	9	8	

Décimal Codé Binaire : DCB

- Remplacer chaque chiffre d'un nombre décimal par son équivalent binaire
- Faire une correspondance directe entre binaire et décimal

Nombre Décimal	5	3	7	1
Nombre Binaire	0101	0011	0111	0001
Nombre Binaire	0101	1001	1000	0011
Nombre Décimal	5	9	8	3

Décimal Codé Binaire

- Sous-Utilisation de l'espace de représentation binaire

Décimal Codé Binaire

- Sous-Utilisation de l'espace de représentation binaire
- 6 représentations interdites

Décimal Codé Binaire

- Sous-Utilisation de l'espace de représentation binaire
- 6 représentations interdites
- 1010_B,1011_B,1100_B,1101_B,1110_B,1111_B

Décimal Codé Binaire

- Sous-Utilisation de l'espace de représentation binaire
- 6 représentations interdites
- 1010_B,1011_B,1100_B,1101_B,1110_B,1111_B
- Différence entre codage binaire et DCB

Décimal Codé Binaire

- Sous-Utilisation de l'espace de représentation binaire
- 6 représentations interdites
- 1010_B,1011_B,1100_B,1101_B,1110_B,1111_B
- Différence entre codage binaire et DCB
- 231 en binaire
- 231 en DCB

Décimal Codé Binaire

- Sous-Utilisation de l'espace de représentation binaire
- 6 représentations interdites
- 1010_B,1011_B,1100_B,1101_B,1110_B,1111_B
- Différence entre codage binaire et DCB
- 231 =11100111_B en binaire
- 231 en DCB

Décimal Codé Binaire

- Sous-Utilisation de l'espace de représentation binaire
- 6 représentations interdites
- 1010_B,1011_B,1100_B,1101_B,1110_B,1111_B
- Différence entre codage binaire et DCB
- 231 =11100111_B en binaire
- 231 =001000110001_B en DCB

Code de Gray

- Une représentation ne diffère de la précédente que d'un bit

Code de Gray

- Une représentation ne diffère de la précédente que d'un bit

Décimal	Binaire	Gray	Décimal	Binaire	Gray
0	0000	0000	8	1000	1100
1	0001	0001	9	1001	1101
2	0010	0011	10	1010	1111
3	0011	0010	11	1011	1110
4	0100	0110	12	1100	1010
5	0101	0111	13	1101	1011
6	0110	0101	14	1110	1001
7	0111	0100	15	1111	1000

Code ASCII

- Besoin de traiter de l'information non numérique

Code ASCII

- Besoin de traiter de l'information non numérique
- Information Alphanumérique : , ? R t j

Code ASCII

- Besoin de traiter de l'information non numérique
- Information Alphanumérique : , ? R t j
- Mise en place d'un codage sur 7 bits :

Code ASCII

- Besoin de traiter de l'information non numérique
- Information Alphanumérique : , ? R t j
- Mise en place d'un codage sur 7 bits : l'ASCII

Code ASCII

- Besoin de traiter de l'information non numérique
- Information Alphanumérique : , ? R t j
- Mise en place d'un codage sur 7 bits : l'ASCII
- American Standard Code for Information Interchange

Code ASCII

- Besoin de traiter de l'information non numérique
- Information Alphanumérique : , ? R t j
- Mise en place d'un codage sur 7 bits : l'ASCII
- American Standard Code for Information Interchange
- 7 bits : 26 lettres minuscules, 26 lettres majuscules, 10 chiffres, 7 signes de ponctuation soit 69 signes à coder. Le reste sert pour des caractères spéciaux

Code ASCII

- Besoin de traiter de l'information non numérique
- Information Alphanumérique : , ? R t j
- Mise en place d'un codage sur 7 bits : l'ASCII
- American Standard Code for Information Interchange
- 7 bits : 26 lettres minuscules, 26 lettres majuscules, 10 chiffres, 7 signes de ponctuation soit 69 signes à coder. Le reste sert pour des caractères spéciaux
- ASCII étendu : 8 bits

ASCII

Caractère	Code Hexadécimal
A	41_H
E	45_H
I	49_H
M	$4D_H$
N	$4E_H$
Q	51_H
R	52_H
U	55_H

ASCII

Caractère	Code Hexadécimal
A	41_H
E	45_H
I	49_H
M	$4D_H$
N	$4E_H$
Q	51_H
R	52_H
U	55_H

- $4E_H 55_H 4D_H 45_H 52_H 49_H 51_H 55_H 45_H$

ASCII

Caractère	Code Hexadécimal
A	41_H
E	45_H
I	49_H
M	$4D_H$
N	$4E_H$
Q	51_H
R	52_H
U	55_H

- $4E_H 55_H 4D_H 45_H 52_H 49_H 51_H 55_H 45_H$
- NUMERIQUE

Nombres Signés

- Comment Coder les Nombres Signés en Binaire ?

Nombres Signés

- Comment Coder les Nombres Signés en Binaire ?
- Introduire un bit de signe :

Nombres Signés

- Comment Coder les Nombres Signés en Binaire ?
- Introduire un bit de signe : bit de poids fort

Nombres Signés

- Comment Coder les Nombres Signés en Binaire ?
- Introduire un bit de signe : bit de poids fort
- Nombre sur 4 bits

b_3	b_2	b_1	b_0	signe	valeur décimale
0	1	0	0		
1	1	0	0		

Nombres Signés

- Comment Coder les Nombres Signés en Binaire ?
- Introduire un bit de signe : bit de poids fort
- Nombre sur 4 bits

b_3	b_2	b_1	b_0	signe	valeur décimale
0	1	0	0	+	
1	1	0	0		

Nombres Signés

- Comment Coder les Nombres Signés en Binaire ?
- Introduire un bit de signe : bit de poids fort
- Nombre sur 4 bits

b_3	b_2	b_1	b_0	signe	valeur décimale
0	1	0	0	+	4
1	1	0	0		

Nombres Signés

- Comment Coder les Nombres Signés en Binaire ?
- Introduire un bit de signe : bit de poids fort
- Nombre sur 4 bits

b_3	b_2	b_1	b_0	signe	valeur décimale
0	1	0	0	+	4
1	1	0	0	-	

Nombres Signés

- Comment Coder les Nombres Signés en Binaire ?
- Introduire un bit de signe : bit de poids fort
- Nombre sur 4 bits

b_3	b_2	b_1	b_0	signe	valeur décimale
0	1	0	0	+	4
1	1	0	0	-	-4

Nombres Signés

- Comment Coder les Nombres Signés en Binaire ?
- Introduire un bit de signe : bit de poids fort
- Nombre sur 4 bits

b_3	b_2	b_1	b_0	signe	valeur décimale
0	1	0	0	+	4
1	1	0	0	-	-4

- Codage Signe + Valeur Absolue

Nombres Signés

- Comment Coder les Nombres Signés en Binaire ?
- Introduire un bit de signe : bit de poids fort
- Nombre sur 4 bits

b_3	b_2	b_1	b_0	signe	valeur décimale
0	1	0	0	+	4
1	1	0	0	-	-4

- Codage Signe + Valeur Absolue
- Nécessite trop de logique pour réaliser des opérateurs arithmétiques

Complément à 2

- Utilisation d'un codage qui permet de limiter les opérateurs

Complément à 2

- Utilisation d'un codage qui permet de limiter les opérateurs
- Complément à 2 :

Complément à 2

- Utilisation d'un codage qui permet de limiter les opérateurs
- Complément à 2 :
  - Bit de signe : bit de poids fort

Complément à 2

- Utilisation d'un codage qui permet de limiter les opérateurs
- Complément à 2 :
  - Bit de signe : bit de poids fort
  - Si bit de signe = 0 : Le nombre est codé

Complément à 2

- Utilisation d'un codage qui permet de limiter les opérateurs
- Complément à 2 :
  - Bit de signe : bit de poids fort
  - Si bit de signe = 0 : Le nombre est codé
  - Si bit de signe = 1 : Complément à 2 pour avoir la valeur

Complément à 2

- Utilisation d'un codage qui permet de limiter les opérateurs
- Complément à 2 :
  - Bit de signe : bit de poids fort
  - Si bit de signe = 0 : Le nombre est codé
  - Si bit de signe = 1 : Complément à 2 pour avoir la valeur
- Principe : Pour un nombre de n bits complémenter le nombre pour arriver à 2^n

Complément à 2

- Codage de 7 :

Complément à 2

- Codage de 7 :

b_3	b_2	b_1	b_0	signe	valeur décimale

Complément à 2

- Codage de 7 :

b_3	b_2	b_1	b_0	signe	valeur décimale
0					

Complément à 2

- Codage de 7 :

b_3	b_2	b_1	b_0	signe	valeur décimale
0	1	1	1		

Complément à 2

- Codage de 7 :

b_3	b_2	b_1	b_0	signe	valeur décimale
0	1	1	1	+	

Complément à 2

- Codage de 7 :

b_3	b_2	b_1	b_0	signe	valeur décimale
0	1	1	1	+	7

Complément à 2

- Codage de 7 :

b_3	b_2	b_1	b_0	signe	valeur décimale
0	1	1	1	+	7

- Codage de -7 :

Complément à 2

- Codage de 7 :

b_3	b_2	b_1	b_0	signe	valeur décimale
0	1	1	1	+	7

- Codage de -7 :

b_3	b_2	b_1	b_0	signe	valeur décimale

Complément à 2

- Codage de 7 :

b_3	b_2	b_1	b_0	signe	valeur décimale
0	1	1	1	+	7

- Codage de -7 :

b_3	b_2	b_1	b_0	signe	valeur décimale
1					

Complément à 2

- Codage de 7 :

b_3	b_2	b_1	b_0	signe	valeur décimale
0	1	1	1	+	7

- Codage de -7 :

b_3	b_2	b_1	b_0	signe	valeur décimale
1	0	0	1		

Complément à 2

- Codage de 7 :

b_3	b_2	b_1	b_0	signe	valeur décimale
0	1	1	1	+	7

- Codage de -7 :

b_3	b_2	b_1	b_0	signe	valeur décimale
1	0	0	1	-	

Complément à 2

- Codage de 7 :

b_3	b_2	b_1	b_0	signe	valeur décimale
0	1	1	1	+	7

- Codage de -7 :

b_3	b_2	b_1	b_0	signe	valeur décimale
1	0	0	1	-	-7

Complément à 2

- Etapes pour complémenter à 2

Complément à 2

- Etapes pour complémenter à 2
- Faire le complément à 1 du nombre : complémentation bit à bit

Complément à 2

- Etapes pour complémenter à 2
- Faire le complément à 1 du nombre : complémentation bit à bit
- Ajouter 1 au nombre

Complément à 2

- Etapes pour complémenter à 2
- Faire le complément à 1 du nombre : complémentation bit à bit
- Ajouter 1 au nombre
- Exemple : codage de -5

Complément à 2

- Etapes pour complémenter à 2
- Faire le complément à 1 du nombre : complémentation bit à bit
- Ajouter 1 au nombre
- Exemple : codage de -5

b_3	b_2	b_1	b_0	Commentaires

Complément à 2

- Etapes pour complémenter à 2
- Faire le complément à 1 du nombre : complémentation bit à bit
- Ajouter 1 au nombre
- Exemple : codage de -5

b_3	b_2	b_1	b_0	Commentaires
0	1	0	1	Valeur Absolue

Complément à 2

- Etapes pour complémenter à 2
- Faire le complément à 1 du nombre : complémentation bit à bit
- Ajouter 1 au nombre
- Exemple : codage de -5

b_3	b_2	b_1	b_0	Commentaires
0	1	0	1	Valeur Absolue
				Complément à 1

Complément à 2

- Etapes pour complémenter à 2
- Faire le complément à 1 du nombre : complémentation bit à bit
- Ajouter 1 au nombre
- Exemple : codage de -5

b_3	b_2	b_1	b_0	Commentaires
0	1	0	1	Valeur Absolue
1				Complément à 1

Complément à 2

- Etapes pour complémenter à 2
- Faire le complément à 1 du nombre : complémentation bit à bit
- Ajouter 1 au nombre
- Exemple : codage de -5

b_3	b_2	b_1	b_0	Commentaires
0	1	0	1	Valeur Absolue
1	0			Complément à 1

Complément à 2

- Etapes pour complémenter à 2
- Faire le complément à 1 du nombre : complémentation bit à bit
- Ajouter 1 au nombre
- Exemple : codage de -5

b_3	b_2	b_1	b_0	Commentaires
0	1	0	1	Valeur Absolue
1	0	1		Complément à 1

Complément à 2

- Etapes pour complémenter à 2
- Faire le complément à 1 du nombre : complémentation bit à bit
- Ajouter 1 au nombre
- Exemple : codage de -5

b_3	b_2	b_1	b_0	Commentaires
0	1	0	1	Valeur Absolue
1	0	1	0	Complément à 1

Complément à 2

- Etapes pour complémenter à 2
- Faire le complément à 1 du nombre : complémentation bit à bit
- Ajouter 1 au nombre
- Exemple : codage de -5

b_3	b_2	b_1	b_0	Commentaires
0	1	0	1	Valeur Absolue
1	0	1	0	Complément à 1
+			1	Ajout de 1

Complément à 2

- Etapes pour complémenter à 2
- Faire le complément à 1 du nombre : complémentation bit à bit
- Ajouter 1 au nombre
- Exemple : codage de -5

b_3	b_2	b_1	b_0	Commentaires
0	1	0	1	Valeur Absolue
1	0	1	0	Complément à 1
+			1	Ajout de 1
			1	

Complément à 2

- Etapes pour complémenter à 2
- Faire le complément à 1 du nombre : complémentation bit à bit
- Ajouter 1 au nombre
- Exemple : codage de -5

b_3	b_2	b_1	b_0	Commentaires
0	1	0	1	Valeur Absolue
1	0	1	0	Complément à 1
+			1	Ajout de 1
		1	1	

Complément à 2

- Etapes pour complémenter à 2
- Faire le complément à 1 du nombre : complémentation bit à bit
- Ajouter 1 au nombre
- Exemple : codage de -5

b_3	b_2	b_1	b_0	Commentaires
0	1	0	1	Valeur Absolue
1	0	1	0	Complément à 1
+			1	Ajout de 1
	0	1	1	

Complément à 2

- Etapes pour complémenter à 2
- Faire le complément à 1 du nombre : complémentation bit à bit
- Ajouter 1 au nombre
- Exemple : codage de -5

b_3	b_2	b_1	b_0	Commentaires
0	1	0	1	Valeur Absolue
1	0	1	0	Complément à 1
+			1	Ajout de 1
1	0	1	1	

Complément à 2

- Etapes pour complémenter à 2
- Faire le complément à 1 du nombre : complémentation bit à bit
- Ajouter 1 au nombre
- Exemple : codage de -5

b_3	b_2	b_1	b_0	Commentaires
0	1	0	1	Valeur Absolue
1	0	1	0	Complément à 1
+			1	Ajout de 1
1	0	1	1	Complément à 2

Complément à 2

- Avantage :

Complément à 2

- Avantage :
- Unicité du 0

Complément à 2

- Avantage :
- Unicité du 0
- Utilisation du même opérateur pour l'addition et la soustraction

Complément à 2

- Avantage :
- Unicité du 0
- Utilisation du même opérateur pour l'addition et la soustraction
- Modulo : 7_H - 4_H = $(7_H + C_H) modulo(10_H)$ = 3_H

Complément à 2

- Avantage :
- Unicité du 0
- Utilisation du même opérateur pour l'addition et la soustraction
- Modulo : 7_H - 4_H = $(7_H + C_H) modulo (10_H)$ = 3_H
- Exemples en binaire.

Complément à 2

- Codage sur N bits, N fini
- On veut coder un nombre négatif $-P$ sur N bits, $P \in [0, 2^N]$
- On sait que $2^N = CP + P$
- On pose $-P = CPmod2^N$
- Ce qui donne $-P = (2^N - P)mod2^N$
- On sait que $P \in [0, 2^N]$ donc on a bien $(2^N - P)mod2^N = -P$

Complément à 2

Si P positif on le code

$P = \sum_{i=0}^{i=N-1} b_i * 2^i$
$P = b_{N-1} * 2^{N-1} + \sum_{i=0}^{i=N-2} b_i * 2^i$ avec $b_{N-1} = 0$

Si P négatif on le code

$P = -(2^N - \sum_{i=0}^{i=N-1} b_i * 2^i)$
$P = -(2^N - b_{N-1} * 2^{N-1} - \sum_{i=0}^{i=N-2} b_i * 2^i)$ avec $b_{N-1} = 1$
$P = -(2^N - 2^{N-1} - \sum_{i=0}^{i=N-2} b_i * 2^i)$
$P = -(2^{N-1}(2-1) - \sum_{i=0}^{i=N-2} b_i * 2^i)$
$P = -(2^{N-1} - \sum_{i=0}^{i=N-2} b_i * 2^i)$
$P = -b_{N-1} * 2^{N-1} + \sum_{i=0}^{i=N-2} b_i * 2^i$ avec $b_{N-1} = 1$

Nombre en complément à 2

$P = -b_{N-1} * 2^{N-1} + \sum_{i=0}^{i=N-2} b_i * 2^i$

Codage Virgule Fixe

Codage $Q_{N.M}$: une partie entière, une partie décimale

$$N = \sum_{i=-M}^{i=N-1} b_i * 2^i$$

Codage $Q_{N.M}$: N bits en partie entière, M bits en partie décimale.

Exemples

$Q_{1.7}$: 10010111_B $=>$ partie entière 1, partie décimale 0010111
représente le nombre décimal 1,171875 soit $2^0 + 2^{-3} + 2^{-5} + 2^{-6} + 2^{-7}$

Codage Flottant

Format IEEE 754 simple précision 32 bits

$$N = (-1)^S.2^{(exposant-127)}.(1, mantisse)$$

- s sur 1 bit : signe, S=0 positif, S=1 négatif
- exposant sur 8 bits : codé en excès 127
- mantisse sur 23 bits : 1,mantisse forme normalisée de l'écriture du nombre

Exemples

$N = 11000000011000000000000000000000_B =>$

- Signe : 1
- Exposant : 10000000
- Mantisse : 11000000000000000000000
- Valeur décimale $N = -2^{(128-127)}.(1,75) = -3,5$

Codage Flottant

l'infini et le zéro

- $+\infty$: 0 11111111 $00000000000000000000000_B$
- $-\infty$: 1 11111111 $00000000000000000000000_B$
- zéro : 0 00000000 $00000000000000000000000_B$

Dynamique Codage simple précision

- Plus grand nombre positif : 0 11111110 $11111111111111111111111_B$ $\approx 3,4^{38}$
- Plus grand nombre négatif : 1 11111110 $11111111111111111111111_B$ $\approx -3,4^{38}$
- Plus petit nombre positif : 0 00000001 $00000000000000000000000_B$ $\approx 1,17^{-38}$
- Plus petit nombre négatif : 1 00000001 $00000000000000000000000_B$ $\approx -1,17^{-38}$

Codage Flottant

Distance entre deux nombres consécutifs

- La distance entre deux nombres consécutifs dépend de l'exposant et est égale à $2^{Exposant-127}.2^{-23}$

Représentation Dénormalisée

- Si Exposant = 00000000 et que la mantisse est non nulle
- Dans ce cas $N = (-1)^{S}.2^{-127}.(0, mantisse)$

Not a Number

- si l'exposant vaut 11111111 et que la mantisse vaut 11111111111111111111111 alors le résultat est NaN
- correspond au cas d'un nombre non initialisé

Codage Flottant

Format IEEE 754 double précision 64 bits

$$N = (-1)^S.2^{(exposant-1023)}.(1, mantisse)$$

- s sur 1 bit : signe, S=0 positif, S=1 négatif
- exposant sur 11 bits : codé en excès 1023
- mantisse sur 52 bits : 1,mantisse forme normalisée de l'écriture du nombre

Format IEEE 754 demi précision 16 bits

$$N = (-1)^S.2^{(exposant-15)}.(1, mantisse)$$

- s sur 1 bit : signe, S=0 positif, S=1 négatif
- exposant sur 5 bits : codé en excès 15
- mantisse sur 10 bits : 1,mantisse forme normalisée de l'écriture du nombre

Erreur des codages entiers, fixe et flottant

Erreur maximale

- la moitié de la valeur du bit de poids faible ou $1/2LSB$ (Least Significant Bit)
  - Codage entier : $0,5 = 2^{-1}$
  - Codage fixe : $0,5 * 2^{-M} = 2^{-(M+1)}$
  - Codage flottant simple précision : $0,5 * 2^{Exposant-127} * 2^{-23} = 2^{Exposant-127} * 2^{-24}$

Plan

1 Introduction : L'électronique numérique à l'aube de 2020 / Méthodes et outils de Conception des systèmes numériques

2 Algèbre de Boole

3 Codage

4 Les composants combinatoire simples

5 Les composants combinatoires complexes

6 Les composants séquentiels : les bascules

7 Les composants séquentiels : les registres

8 Les composants séquentiels : les compteurs / Le traitement Pipeliné

Plan

1 Introduction : L'électronique numérique à l'aube de 2020 / Méthodes et outils de Conception des systèmes numériques

2 Algèbre de Boole

3 Codage

4 Les composants combinatoire simples

5 Les composants combinatoires complexes

6 Les composants séquentiels : les bascules

7 Les composants séquentiels : les registres

8 Les composants séquentiels : les compteurs / Le traitement Pipeliné

Plan

1. Introduction : L'électronique numérique à l'aube de 2020 / Méthodes et outils de Conception des systèmes numériques
2. Algèbre de Boole
3. Codage
4. Les composants combinatoire simples
5. Les composants combinatoires complexes
6. **Les composants séquentiels : les bascules**
7. Les composants séquentiels : les registres
8. Les composants séquentiels : les compteurs / Le traitement Pipeliné

Plan

1. Introduction : L'électronique numérique à l'aube de 2020 / Méthodes et outils de Conception des systèmes numériques
2. Algèbre de Boole
3. Codage
4. Les composants combinatoire simples
5. Les composants combinatoires complexes
6. Les composants séquentiels : les bascules
7. Les composants séquentiels : les registres
8. Les composants séquentiels : les compteurs / Le traitement Pipeliné

Plan

1 Introduction : L'électronique numérique à l'aube de 2020 / Méthodes et outils de Conception des systèmes numériques

2 Algèbre de Boole

3 Codage

4 Les composants combinatoire simples

5 Les composants combinatoires complexes

6 Les composants séquentiels : les bascules

7 Les composants séquentiels : les registres

8 Les composants séquentiels : les compteurs / Le traitement Pipeliné

Plan

1. Introduction : L'électronique numérique à l'aube de 2020 / Méthodes et outils de Conception des systèmes numériques
2. Algèbre de Boole
3. Codage
4. Les composants combinatoire simples
5. Les composants combinatoires complexes
6. Les composants séquentiels : les bascules
7. Les composants séquentiels : les registres
8. Les composants séquentiels : les compteurs / Le traitement Pipeliné