

Représentation des entiers



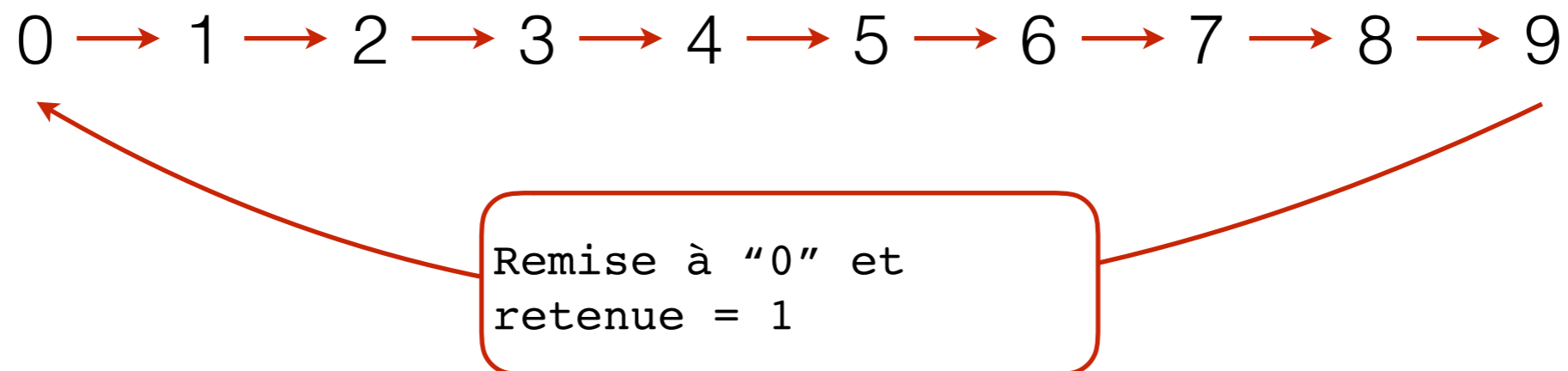
GIF-1001 Ordinateurs : Structure et Applications, H2017
Jean-François Lalonde

Représentation des entiers

- Chaque nombre doit avoir une représentation différente
 - 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14...
 - I, II, III, IV, V, VI, VII, VIII, IX, X, XI, XII, XIII, XIV...
- Pour se simplifier la vie, on utilise une représentation avec laquelle il est facile de compter
 - $\text{CMXCIX} + \text{I} = ?$

Compter en décimal

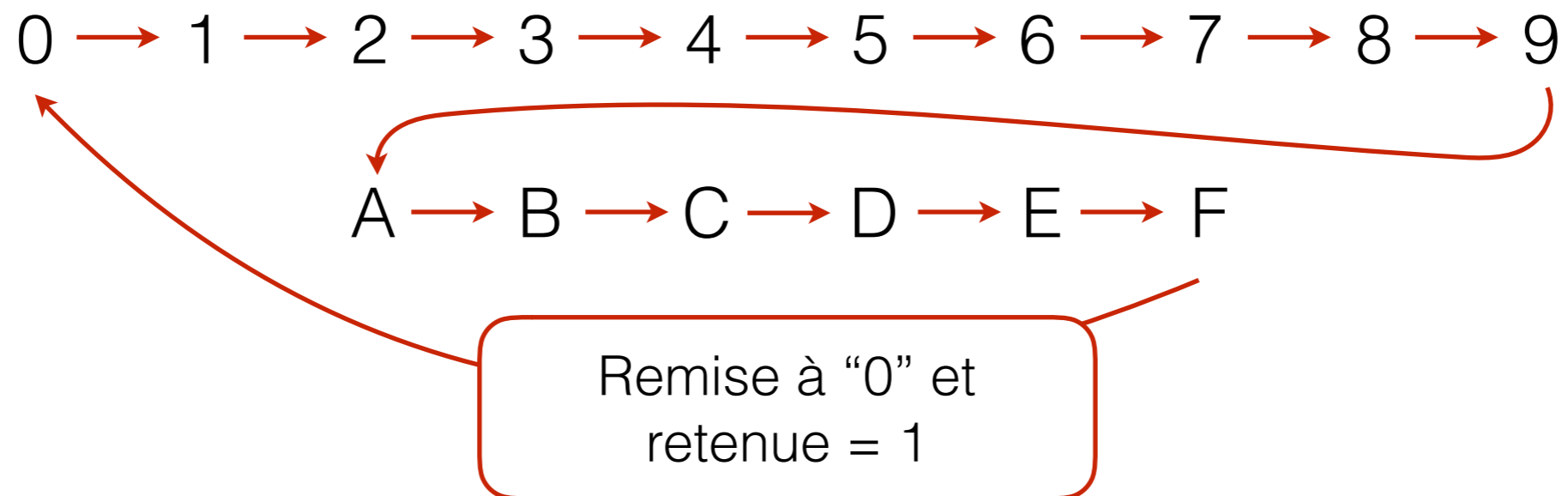
- Ordre des symboles: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
- Que faire quand on arrive au bout?
 - on recommence au début en ajoutant une retenue de 1 au prochain symbole



Représentation décimale	Position 3	Position 2	Position 1	Position 0					
	1	4	2	3					
Valeur décimale	1×10^3	+	4×10^2	+	2×10^1	+	3×10^0	=	1423

Compter en *hexadécimal*

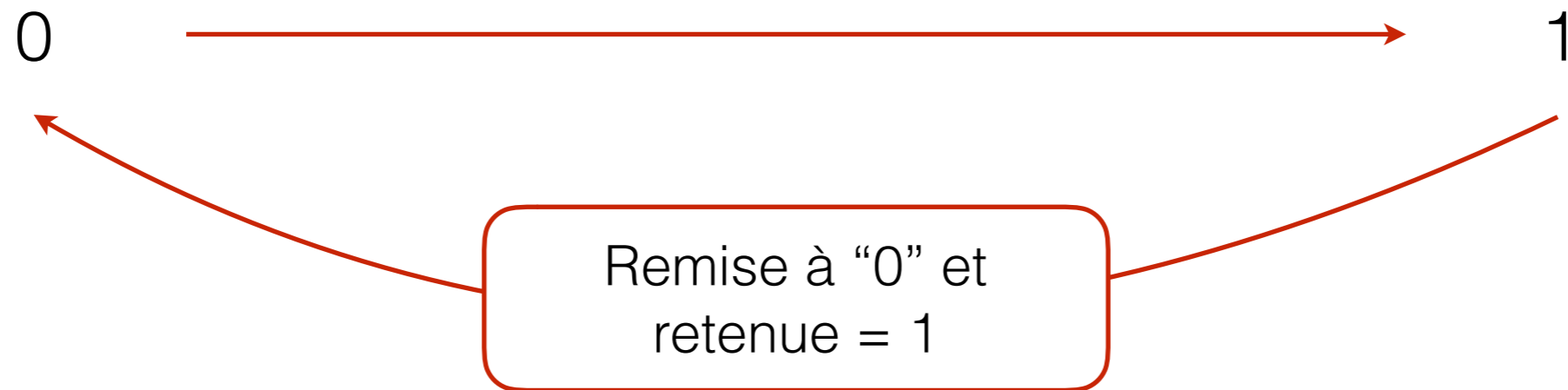
- Et si on utilisait 16 symboles au lieu de 10?



Représentation hexadécimale	Position 3	Position 2	Position 1	Position 0					
	1	4	2	3					
Valeur décimale	1×16^3	+	4×16^2	+	2×16^1	+	3×16^0	=	5155

Compter en binaire

- Et si on utilisait 2 symboles au lieu de 10?



- En binaire, un symbole se nomme « bit »

Représentation binaire	Position 3	Position 2	Position 1	Position 0					
	1	0	1	1					
Valeur décimale	1×2^3	+	0×2^2	+	1×2^1	+	1×2^0	=	11

Aujourd'hui

- 4 « points hyper importants à retenir™ »



Point hyper important à retenir™ #1

- Dans un ordinateur, *tout, absolument tout*, est stocké en format binaire
 - nombres entiers positifs, négatifs, nombre à virgule flottante, caractères, symboles, instructions, programmes, adresses, images, sons, fichiers pdf, vidéos, facebook, etc. Bref, *tout!*
 - La plus petite unité d'information est nommée "bit".
 - Un bit peut avoir la valeur 0 ou 1.



Nombres différents

- Combien de nombres différents peut-on générer avec...
 - 2 symboles en format décimal?
 - 3 symboles en format décimal?
 - 1 symbole en format hexadécimal?
 - 2 symboles en format hexadécimal?
 - 3 symboles en format hexadécimal?
 - 4 bits en format binaire?
 - 8 bits en format binaire?
 - 16 bits en format binaire?

Nombres différents

- Combien de nombres différents peut-on générer avec...
 - 2 symboles en format décimal? $\rightarrow 0 \text{ à } 99 = 10^2 = 100$ nombres
 - 3 symboles en format décimal? $\rightarrow 0 \text{ à } 999 = 10^3 = 1000$ nombres
 - 1 symbole en format hexadécimal? $\rightarrow 0 \text{ à } F = 16^1 = 16$ nombres
 - 2 symboles en format hexadécimal? $\rightarrow 0 \text{ à } FF = 16^2 = 256$ nombres
 - 3 symboles en format hexadécimal? $\rightarrow 0 \text{ à } FFF = 16^3 = 4096$ nombres
 - 4 bits en format binaire? $\rightarrow 0 \text{ à } 1111 = 2^4 = 16$ nombres
 - 8 bits en format binaire? $\rightarrow 0 \text{ à } 11111111 = 2^8 = 256$ nombres
 - 16 bits en format binaire? $\rightarrow 0 \text{ à } 1111111111111111 = 2^{16} = 65536$ nombres

Nombre de bits

- 1 bit : 2 valeurs (0 et 1)
- 4 bits: 16 valeurs (0 à 15) (aussi appelé *nibble*)
- 8 bits: 256 valeurs (0 à 255) (aussi appelé *octet*, ou, en anglais, « *byte* »)
- 10 bits: 1024 valeurs
- 16 bits: 65536 valeurs
- 20 bits: 1 048 576 valeurs
- 32 bits: 4 294 967 296 valeurs

Point hyper important à retenir™ #2

- On utilise un nombre *fini* et *pré-déterminé* de bits pour représenter de l'information.
 - 8 bits forment un octet (byte ou char) pouvant représenter $2^8 = 256$ valeurs.
 - 1 Kilo-octet (1 ko) = 1024 octets ou 8192 bits.
 - 1 Mega-octet (1 Mo) = 1024 Ko, 1 048 576 octets, ou ...
- Ce nombre doit toujours être pré-déterminé
 - Exemple (en programmation C/C++)

```
int toto = 2;
```

 - Le « `int` » indique, la plupart du temps, un entier (signé) sur 4 octets
 - Exemple (pour le TP0)
 - Convertissez 9 en binaire sur 8 bits → 00001001



Conventions d'écriture

- Comment différencier
 - 1111 (hexadécimal),
 - 1111 (binaire),
 - et 1111 (décimal)?
- Hexadécimal: on utilise le préfixe « 0x » ou l'indice « h ». Ex:
 - $0x1111 = 1111_h = 4369$
- Binaire: on utilise le préfixe « 0b » ou l'indice « b ». Ex:
 - $0b1111 = 1111_b = 0xF = 15$
- Décimal: aucune notation particulière.

Conversions

- Vers décimal
 - Binaire vers décimal
 - Hexadécimal vers décimal
- Binaire / hexadécimal
 - Binaire vers hexadécimal
 - Hexadécimal vers binaire
- Décimal vers binaire
- Décimal vers hexadécimal

Vers décimal

- De binaire vers décimal
 - 0b10010101

Position	7	6	5	4	3	2	1	0
Bit	1	0	0	1	0	1	0	1
Valeur	128	64	32	16	8	4	2	1
=	128	0	0	16	0	4	0	1

= 149

- De hexadécimal vers décimal
 - 0xCAFE

Position	3	2	1	0
Chiffre	C (12)	A (10)	F (15)	E (14)
Valeur	4096	256	16	1
=	49152	2560	240	14

= 51966

Petite pratique...

- De binaire vers décimal
 - 0b11001011

Position	7	6	5	4	3	2	1	0
Bit	1	1	0	0	1	0	1	1
Valeur	128	64	32	16	8	4	2	1
=								

- De hexadécimal vers décimal
 - 0xFACE

Position	3	2	1	0
Chiffre	F	A	C	E
Valeur	4096	256	16	1
=				

Petite pratique...

- De binaire vers décimal

- 0b11001011

Position	7	6	5	4	3	2	1	0
Bit	1	1	0	0	1	0	1	1
Valeur	128	64	32	16	8	4	2	1
=	128	64	0	0	8	0	2	1

= 203

- De hexadécimal vers décimal

- 0xFACE

Position	3	2	1	0
Chiffre	F	A	C	E
Valeur	4096	256	16	1
=	61440	2560	192	14

= 64206

Binaire / hexadécimal

Hexadécimal	Binaire
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
A	1010
B	1011
C	1100
D	1101
E	1110
F	1111

0xDEADBEEF

0b 1101 1110 1010 1101 1011 1110 1110 1111

Petite pratique...

Hexadécimal	Binaire
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
A	1010
B	1011
C	1100
D	1101
E	1110
F	1111

0xCAFE

0b ?

0x12AB

0b ?

Petite pratique...

Hexadécimal	Binaire
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
A	1010
B	1011
C	1100
D	1101
E	1110
F	1111

0xCAFE

0b 1100 1010 1111 1110

0x12AB

0b 0001 0010 1010 1011

Petite pratique...

Hexadécimal	Binaire
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
A	1010
B	1011
C	1100
D	1101
E	1110
F	1111

0b 1010 0011 1101 1000

0x ?

0b 1100 0011 1001 0000

0x ?

Petite pratique...

Hexadécimal	Binaire
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
A	1010
B	1011
C	1100
D	1101
E	1110
F	1111

0b 1010 0011 1101 1000

0xA3D8

0b 1100 0011 1001 0000

0xC390

Point hyper important à retenir™ #3

- L'hexadécimal est une *façon plus compacte* de représenter du binaire.
- 1 “symbole” en hexadécimal = 4 bits.

Hexadécimal	Binaire
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
A	1010
B	1011
C	1100
D	1101
E	1110
F	1111



De décimal vers binaire

- 10 = 0b?

10	2		
-10	5	2	
0	-4	2	2
	1	-2	1
		0	

- 10 = 0b1010

De décimal vers binaire

Position	7	6	5	4	3	2	1	0
Valeur	128	64	32	16	8	4	2	1

- 10 en binaire?
 - commencer de la gauche: 0×128 , 0×64 , ...

Bits	0	0	0	0				
------	---	---	---	---	--	--	--	--

- 1×8 . On met un "1" à la position 3. Reste 2

Bits	0	0	0	0	1			
------	---	---	---	---	---	--	--	--

- 0×4

Bits	0	0	0	0	1	0		
------	---	---	---	---	---	---	--	--

- 1×2 . On met un "1" à la position 1. Reste 0. Terminé!

Bits	0	0	0	0	1	0	1	0
------	---	---	---	---	---	---	---	---

De décimal vers binaire

Position	7	6	5	4	3	2	1	0
Valeur	128	64	32	16	8	4	2	1

- 147 en binaire?
- commencer de la gauche: 1 x 128. Reste 19

Bits	1							
------	---	--	--	--	--	--	--	--

- 0 x 64, 0 x 32, 1 x 16. Reste 3

Bits	1	0	0	1				
------	---	---	---	---	--	--	--	--

- 0 x 8, 0 x 4, 1 x 2. Reste 1

Bits	1	0	0	1	0	0	1	
------	---	---	---	---	---	---	---	--

- 1 x 1. Terminé!

Bits	1	0	0	1	0	0	1	1
------	---	---	---	---	---	---	---	---

De décimal vers binaire

- 147 = 0b?

147	2						
-146	73	2					
1	-72	36	2				
	1	-36	18	2			
		0	-18	9	2		
			0	-8	4	2	
				1	-4	2	2
					0	-2	1
						0	

- 147 = 0b10010011

De décimal vers hexadécimal

- $23147 = 0x?$

23147	16		
-23136	1446	16	
11 (B)	-1440	90	16
	6	-80	5
		10 (A)	

- $23147 = 0x5A6B$

Addition en binaire

Cas simples

$$\begin{array}{r} 0 \\ + 0 \\ \hline 0 \end{array} \quad \begin{array}{r} 0 \\ + 1 \\ \hline 1 \end{array} \quad \begin{array}{r} 1 \\ + 0 \\ \hline 1 \end{array} \quad \begin{array}{r} 1 \\ + 1 \\ \hline 1 \ 0 \end{array}$$

dépassements

Cas à plusieurs bits

$$\begin{array}{r} 1 \ 1 \ 0 \\ + 0 \ 0 \ 1 \\ \hline 1 \ 1 \ 1 \end{array} \quad \begin{array}{r} 1 \ 0 \ 1 \ 0 \\ + 0 \ 0 \ 1 \ 1 \\ \hline 1 \ 1 \ 0 \ 1 \end{array} \quad \begin{array}{r} 1 \ 0 \ 1 \ 0 \ 0 \ 0 \ 1 \\ + 1 \ 0 \ 1 \ 0 \ 1 \ 1 \ 0 \\ \hline 1 \ 0 \ 1 \ 0 \ 0 \ 1 \ 1 \ 1 \end{array}$$

retenues

Nombres entiers négatifs

- Représentation « signe et magnitude »
 - Le premier bit est le signe: 0 = positif, 1 = négatif
 - Le reste est la magnitude
 - Exemple: $1101_b = (-1) \times (4 + 1) = -5$
 - Combien de nombres peut-on représenter?
- Problèmes?
 - Représentation de 0?
 - $0 = 0000_b = 1000_b$
 - Opérations arithmétiques:
 - $-3 + 4 = 1011_b + 0100_b = 1111_b = -7!$

Représentation “complément 2”

- Le premier bit est $-2^{(\text{nombre de bits}-1)}$, le reste est additionné
- Exemple: $1101_b = ?$
 - $1101_b = -2^3 + 5 = -3.$
- Pour changer le signe d'un nombre, il faut:
 - le soustraire de 2^N (d'où le nom “complément 2”)
 - plus direct (et plus facile): inverser tous les bits et ajouter 1.
- Exemple: $3 = 0011_b$. $-3 = ?_b$
 - Si on prend le complément + 1, on obtient: $1100_b + 1_b = 1101_b.$
- Combien de nombres peut-on représenter?

Nombres entiers négatifs

- Problèmes de tout à l'heure?
 - Représentation de 0?
 - 0000_b seulement (maintenant, $1000_b = -8_d$)
 - Opérations arithmétiques:
 - $-3 + 4 = 1101_b + 0100_b = 0001_b = 1$ (beaucoup mieux)
- En pratique, soustraction = addition avec complément 2

Débordement (“overflow”)

- Quels nombres peut-on représenter avec N bits en complément 2?
 - -2^{N-1} à $2^{N-1}-1$
- Qu’arrive-t-il si on additionne
 - $5 + 4 = ?$
 - on obtient -7!
- De façon similaire, $-5 + -4 = 7$
- Comment faire pour détecter un débordement?
 - le bit de signe change

Condition spéciale: « carry »

- Utile pour l'arithmétique non-signée
- 2 conditions:
 - S'il y a une retenue qui sort des bits les plus significatifs
 - Si un emprunt doit être fait sur le bit le plus significatif
- N'est pas important pour l'arithmétique signée

Point hyper important à retenir™ #4

- A priori, nous ne pouvons pas savoir ce qu'une chaîne binaire signifie.
 - Ex: quelle est la valeur décimale de 0b1010 (ou 0xA)?
 - La bonne réponse est: ça dépend!

entier non-signé	10
entier signé	-6

- Il nous *faut* donc savoir quel format utiliser pour bien interpréter les données

