

Enseignes et afficheurs à LED

Nombres et champs de bits



Dr. Yves Tiecoura

Nombres et champs de bits



Dr. Yves Tiecoura

Nombres et champs de bits



- Symboles binaires
- Numération binaire
- Arithmétique modulaire
- Nombres signés
- Types en C
- Hexadécimal
- Codage des caractères



Nombres et champs de bits

- LED éteinte ou allumée
- Faux ou Vrai
- 0 ou 1
- 0 V ou + 5 V
- Ex : 1 0 0 1 0 0 1 0

Numération binaire



Codage binaire

Binaire	Décimal
0	0
1	1
1 0	2
1 1	3
1 0 0	4
1 0 1	5
1 1 0	6
1 1 1	7
1 0 0 0	8
1 0 0 1	9
1 0 1 0	10
1 0 1 1	11
1 1 0 0	12
1 1 0 1	13
1 1 1 0	14
1 1 1 1	15
1 0 0 0 0	16

Poids des bits

Rang	Valeur	Décimal
0	1	$1 = 2^0$
1	10	$2 = 2^1$
2	100	$4 = 2^2$
3	1000	$8 = 2^3$
4	10000	$16 = 2^4$
5	100000	$32 = 2^5$
6	1000000	$64 = 2^6$
7	10000000	$128 = 2^7$
8	100000000	$256 = 2^8$
9	1000000000	$512 = 2^9$
10	10000000000	$1024 = 2^{10}$

2
16
64
128
1024

$$10011010010b = 1234$$

Nombres de taille limitée



- Les mathématiciens travaillent avec des nombre arbitrairement grands.
- Les circuits électroniques ont toujours une taille limitée !
- Quelle sont les contraintes liées à cette limite de taille ?

Nombres entiers modulaires



Codage binaire

Binaire	Décimal
0	0
1	1
1 0	2
1 1	3
1 0 0	4
1 0 1	5
1 1 0	6
1 1 1	7
1 0 0 0	8
1 0 0 1	9
1 0 1 0	10
1 0 1 1	11
1 1 0 0	12
1 1 0 1	13
1 1 1 0	14
1 1 1 1	15
1 0 0 0 0	16

Table de l'addition :

0 + 0 = 0, retenue 0
 0 + 1 = 1, retenue 0
 1 + 0 = 1, retenue 0
 1 + 1 = 0, retenue 1

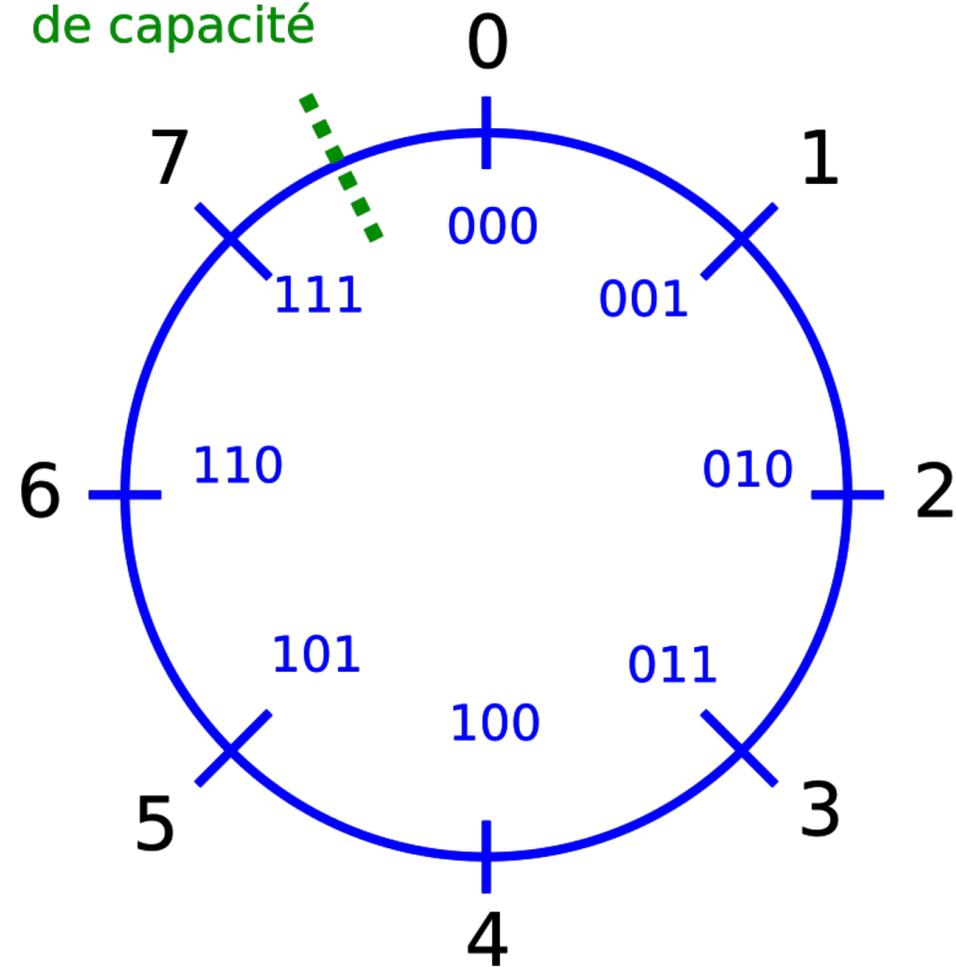
$$\begin{array}{r}
 \overset{1}{0} \ \overset{1}{1} \ \overset{1}{1} \\
 \hline
 0 \ 1 \ 1 \\
 + \ 2 \\
 \hline
 1 \ 0 \ 1 \\
 = 5
 \end{array}
 \qquad
 \begin{array}{r}
 \overset{1}{1} \ \overset{1}{1} \ 0 \\
 \hline
 1 \ 1 \ 0 \\
 + \ 3 \\
 \hline
 0 \ 0 \ 1 \\
 = 1
 \end{array}$$

Dépassement de capacité

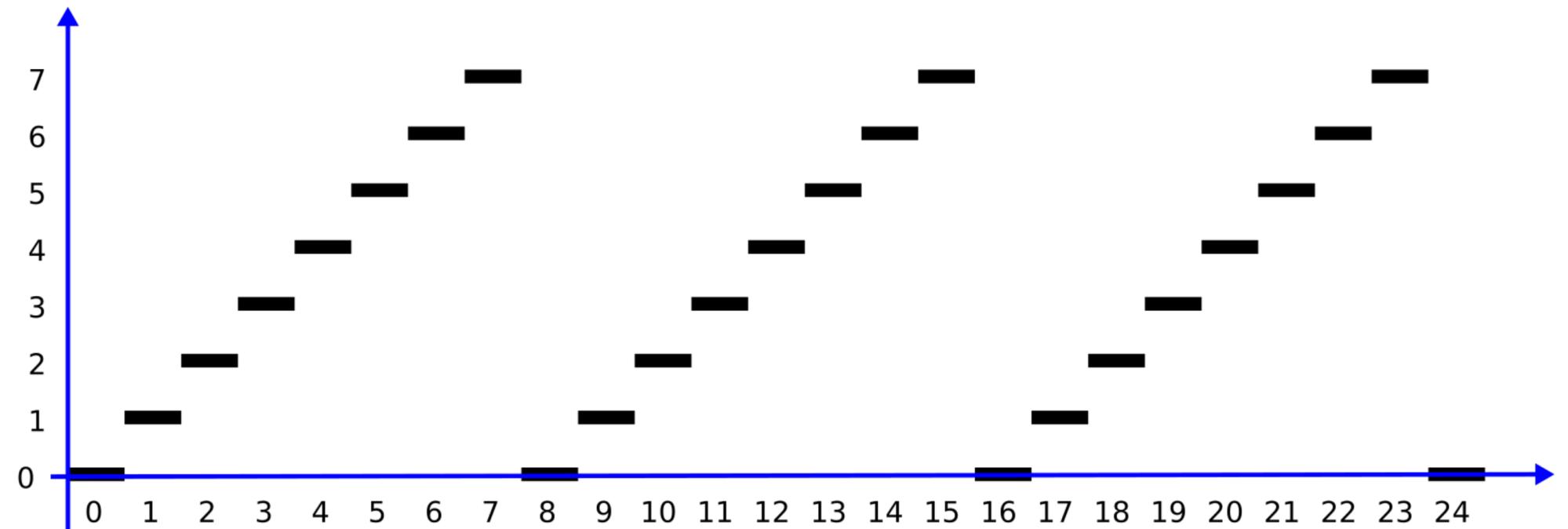
Nombres entiers modulaires



Dépassement
de capacité



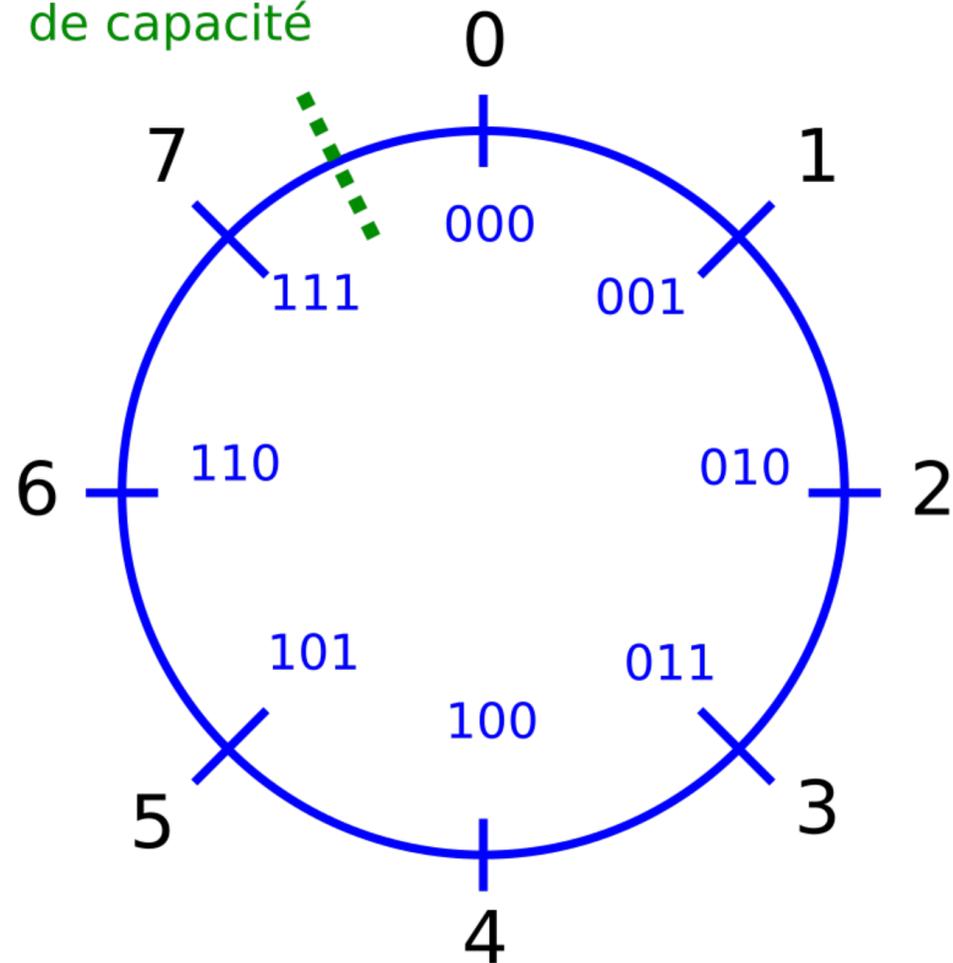
Nombres positifs :



Nombres entiers modulaires



Dépassement de capacité



$$\begin{array}{r}
 \overset{1}{0} \ 1 \ 1 \\
 \hline
 0 \ 1 \ 0 \\
 \hline
 1 \ 0 \ 1 \\
 = 5
 \end{array}$$

$$\begin{array}{r}
 1 \ 1 \ 0 \\
 \hline
 0 \ 1 \ 1 \\
 \hline
 0 \ 1 \ 1 \\
 = 3
 \end{array}$$

$$\begin{array}{r}
 \overset{1}{1} \ 1 \ 0 \\
 \hline
 0 \ 1 \ 1 \\
 \hline
 0 \ 0 \ 1 \\
 = 1
 \end{array}$$

$$\begin{array}{r}
 0 \ 1 \ 1 \\
 \hline
 1 \ 0 \ 1 \\
 \hline
 1 \ 1 \ 0 \\
 = 6
 \end{array}$$

Nombres de taille limitée

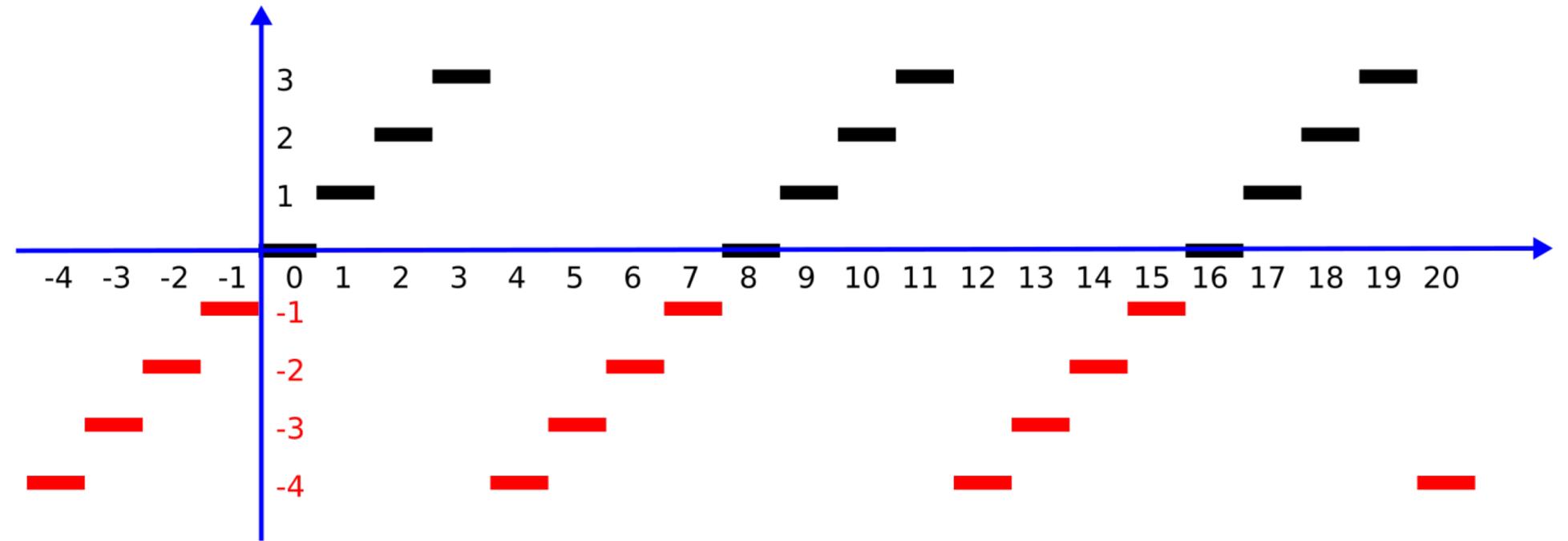
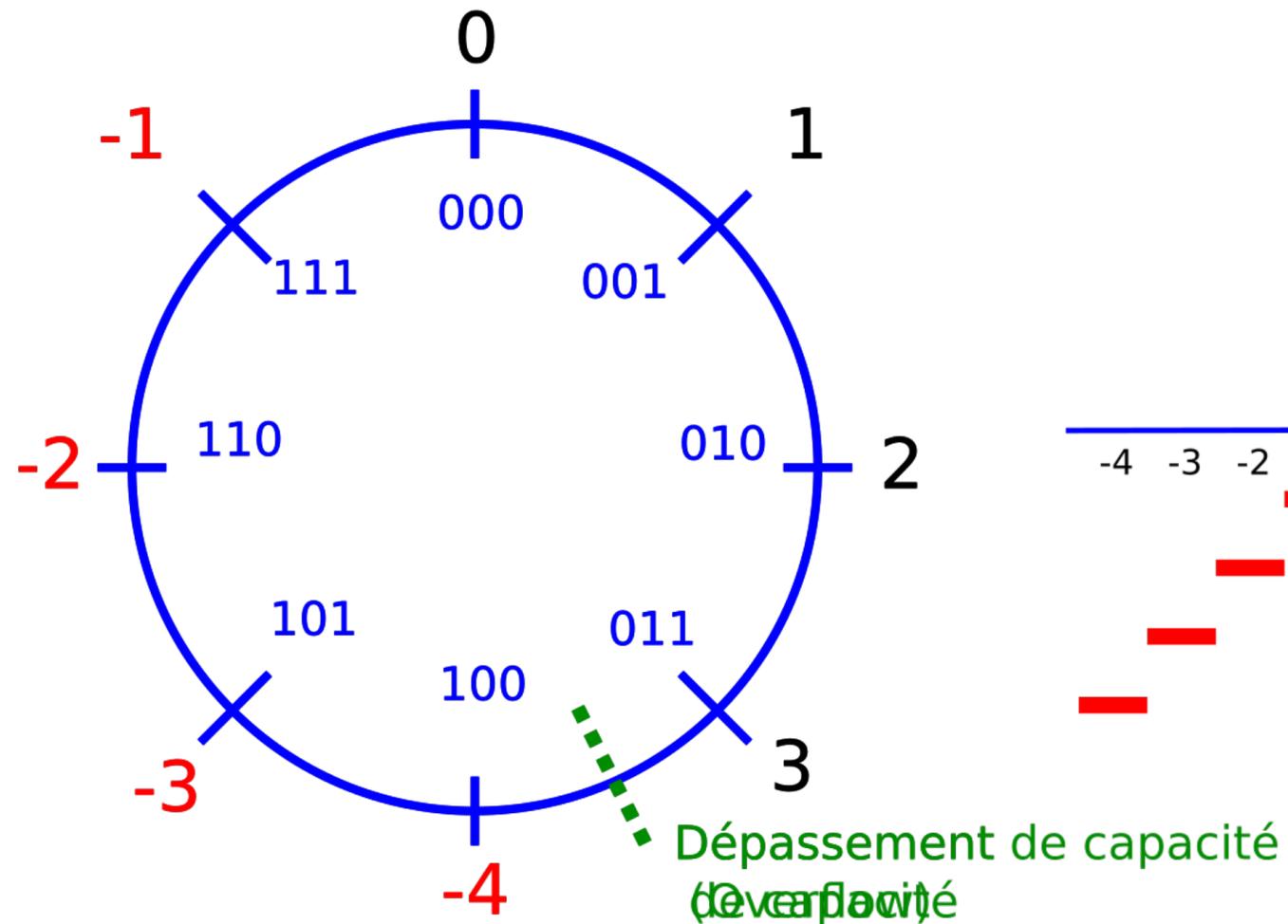


- Peut-on représenter des nombres négatifs ?

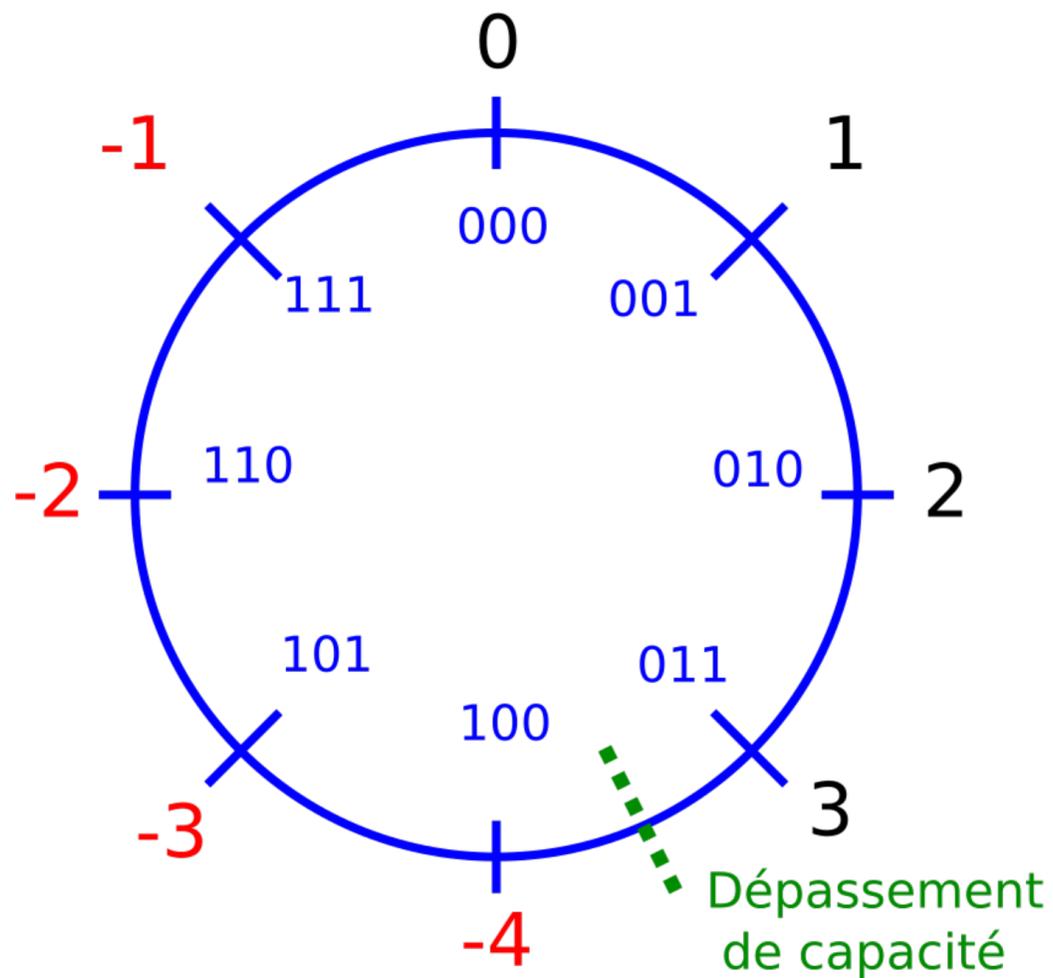
Nombres entiers modulaires



Nombres signés (complément à deux) :



Nombres entiers modulaires



$$\begin{array}{r}
 \overset{1}{1} \ 1 \ 1 \\
 0 \ 1 \ 1 \\
 \hline
 0 \ 1 \ 0 \\
 \end{array}
 \begin{array}{l}
 + -1 \\
 + 3 \\
 = 2
 \end{array}$$

$$\begin{array}{r}
 \overset{1}{0} \ \overset{1}{1} \ 0 \\
 0 \ 1 \ 1 \\
 \hline
 1 \ 0 \ 1 \\
 \end{array}
 \begin{array}{l}
 + 2 \\
 + 3 \\
 = -3 \\
 \dots\dots
 \end{array}$$

$$\begin{array}{r}
 1 \ 1 \ 0 \\
 0 \ 0 \ 1 \\
 \hline
 1 \ 0 \ 1 \\
 \end{array}
 \begin{array}{l}
 - 2 \\
 - 1 \\
 = -3
 \end{array}$$

$$\begin{array}{r}
 1 \ 1 \ 0 \\
 0 \ 1 \ 1 \\
 \hline
 0 \ 1 \ 1 \\
 \end{array}
 \begin{array}{l}
 - 2 \\
 - 3 \\
 = 3 \\
 \dots\dots
 \end{array}$$

Types en C



Type :

char :	nombre de 8 bits (signé ou non, selon les réglages du compilateur)
signed char :	nombre de 8 bits signé
unsigned char :	nombre de 8 bits positif
int :	nombre généralement de 16 bits (signé ou non)
signed int :	nombre de 16 bits signé
unsigned int :	nombre de 16 bits positif
long int :	nombre généralement de 32 bits (signé ou non)
signed long int :	nombre de 32 bits signé
unsigned long int :	nombre de 32 bits positif

Types en C, version C99



Type :

uint8_t : nombre de 8 bits positifs

int8_t : nombre de 8 bits signé

uint16_t : nombre de 16 bits positif

int16_t : nombre de 16 bits signé

uint32_t : nombre de 32 bits positif

int32_t : nombre de 32 bits signé

Magnitude :

0 ... 255

-128 ... +127

0 ... 65'635

-32'768 ... +32'767

0 ... 4'294'967'295

-2'147'483'648 ... +2'147'483'647

Hexadécimal

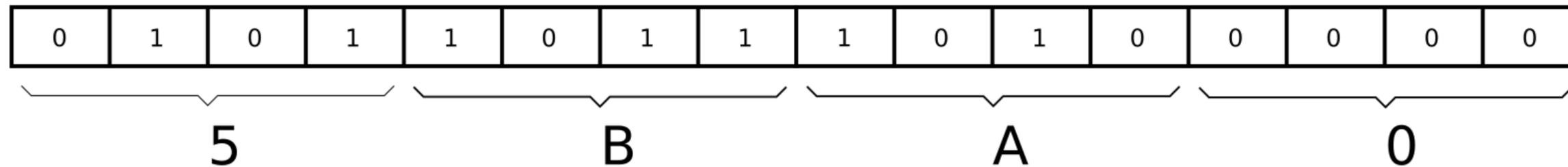


- Le binaire est parfait pour les machines
- ...mais malcommode pour les humains !
- L'hexadécimal est plus pratique.

Nombres entiers modulaires



Le nombre décimal 23456 :



Notation du langage C : **0x5BA0**

0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	A
1	0	1	1	B
1	1	0	0	C
1	1	0	1	D
1	1	1	0	E
1	1	1	1	F

Codage des caractères



Table de codage ASCII

	00 _h	01 _h	02 _h	03 _h	04 _h	05 _h	06 _h	07 _h	08 _h	09 _h	0A _h	0B _h	0C _h	0E _h	0E _h	0F _h
	0.	1.	2.	3.	4.	5.	6.	7.	8	9.	10.	11.	12.	13.	14.	15.
00 _h 0.	NUL	SOH	STX	ETX	EOT	ENQ	ENQ	BEL	BS	HT	LF	VT	FF	CR	SO	SI
10 _h 16.	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
20 _h 32.		!	"	#	\$	%	&	'	()	*	+	,	-	.	/
30 _h 48.	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40 _h 64.	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50 _h 80.	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
60 _h 96.	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
60 _h 112.	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL

Pour trouver le code d'un caractère, ajouter la valeur de la ligne et celle de la colonne **en décimal** ou **en hexadécimal**

Nombres et champs de bits



- Symboles binaires
- Numération binaire
- Arithmétique modulaire
- Nombres signés
- Types en C
- Hexadécimal
- Codage des caractères