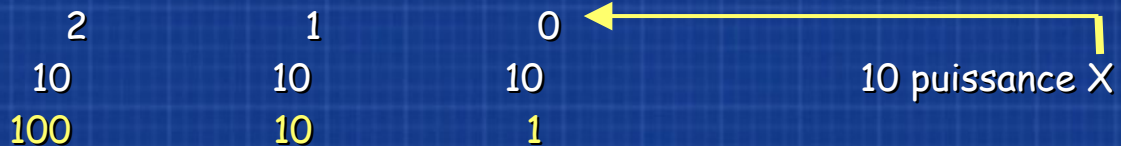


LE DECIMALE

0 1 2 3 4 5 6 7 8 9

le décimale comprends 10 chiffres de 0 à 9 **BASE 10**



$$4 \cdot 10^0 + 8 \cdot 10^1 + 2 \cdot 10^2 = 4 + 80 + 200 = 284$$

BINAIRE

0 à 1 correspondant à l'état électrique de 0 = 0V et 1 = 5V **BASE 2**

2	1	0	
2	2	2	2 puissance X
4	2	1	

1 | **0** | **1** (de 0 à 1)

$$101 = 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 1 \cdot 4 + 0 \cdot 2 + 1 \cdot 1 = 4 + 0 + 1 = 5$$

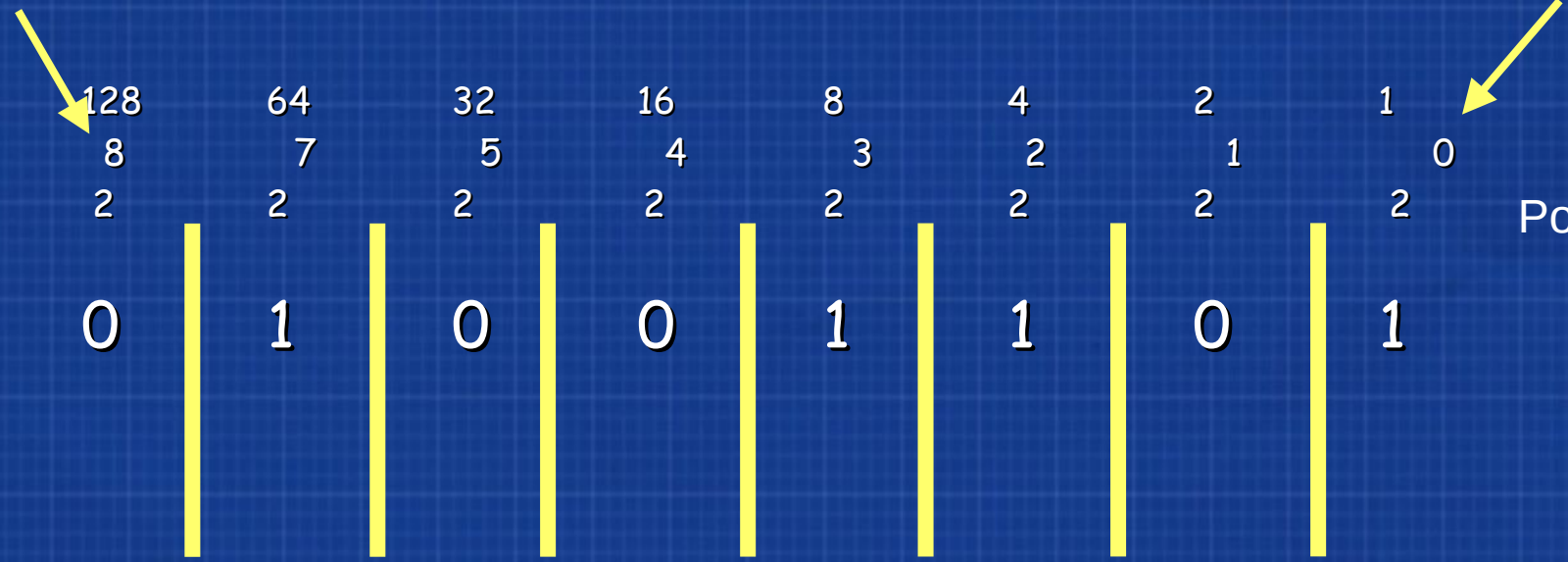
Base 2

Base 10

BINAIRE

MSB Most significant bit

LSB last significant bit



Poids binaire

UN OCTET (8 bit)

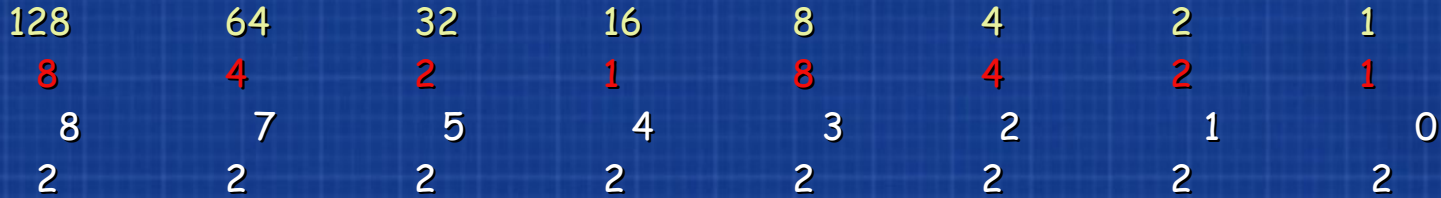
Bit = Binary digit

HEXADECIMAL

HEXADECIMAL de 0 à F

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

(16 formes)



hexadécimal	Décimal	binaire
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
A	10	1010
B	11	1011
C	12	1100
D	13	1101
E	14	1110
F	15	1111
10	16	1 0000
11	17	1 0001
....

0

1

0

0

1

1

0

1

4 en base 10

13 en base 10

4 en base 16

D en base 16

Dixième chiffre hexa
de 0 à F

Premier chiffre en hexa
de 0 à F

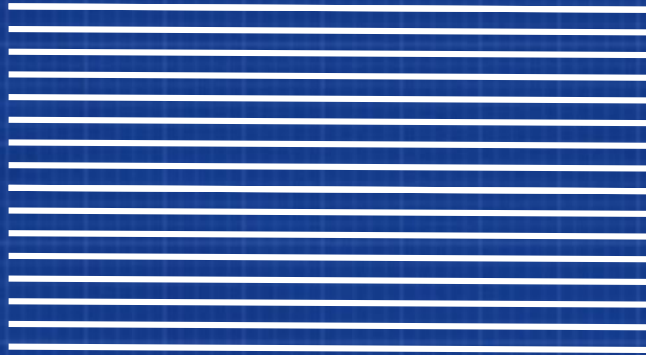
TABLE ASCII

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	\$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[END OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

BUS DE DONNEES ET BUS ADRESSE



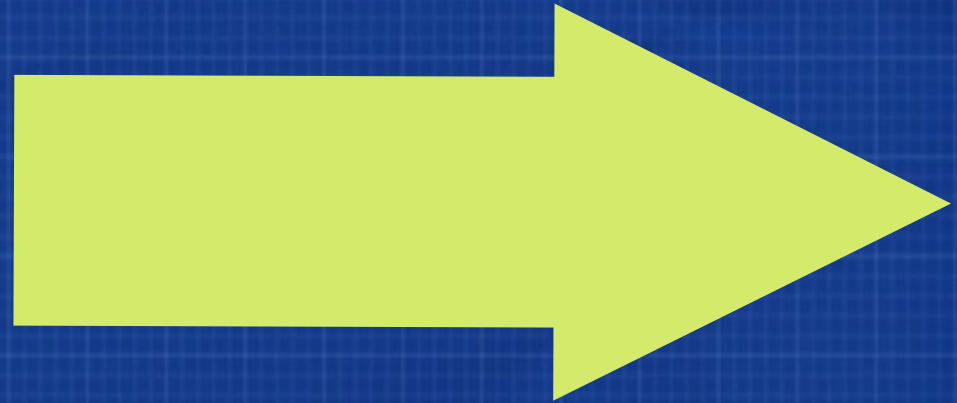
BUS DE DONNEES



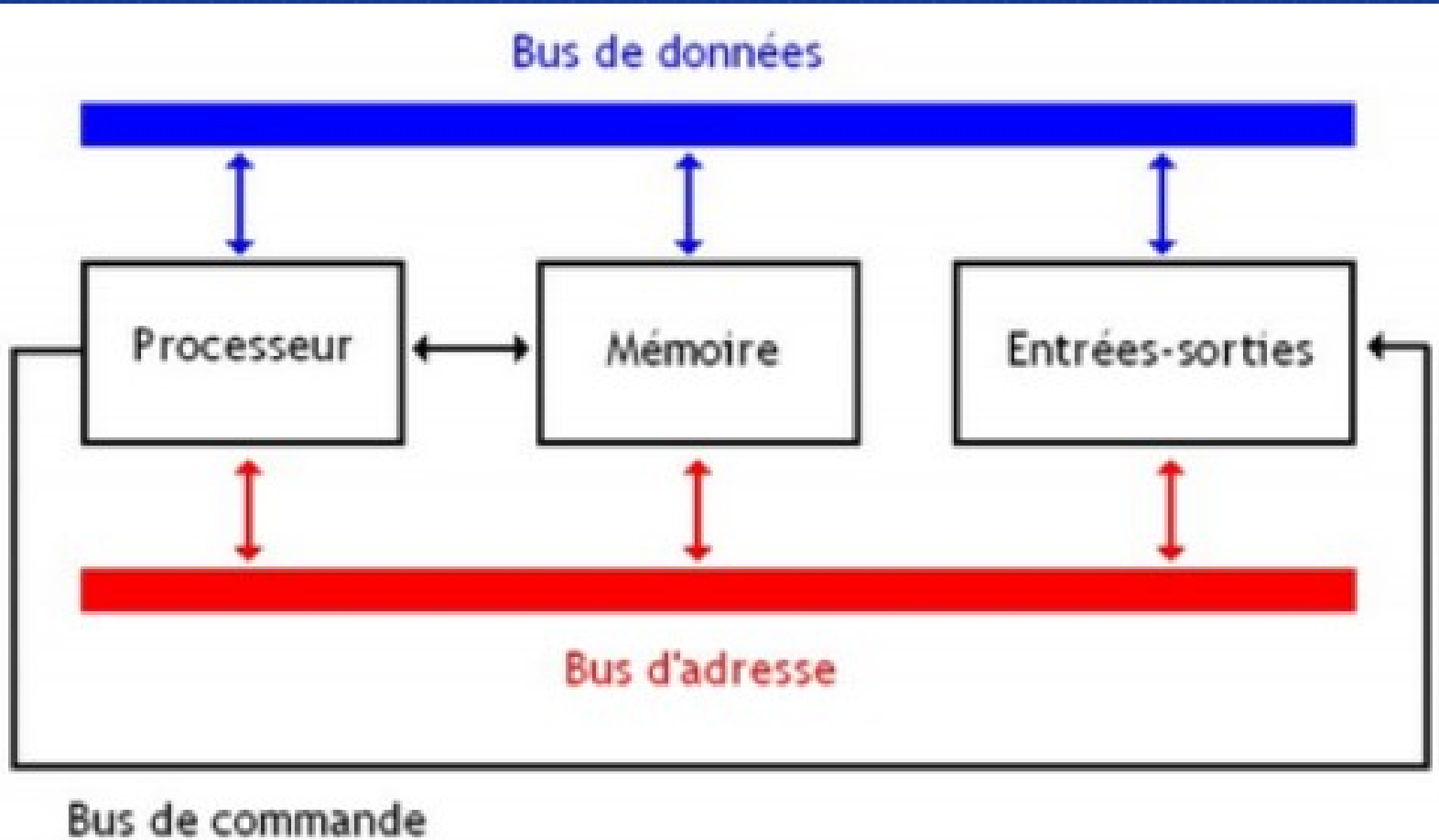
BUS D'ADRESSE



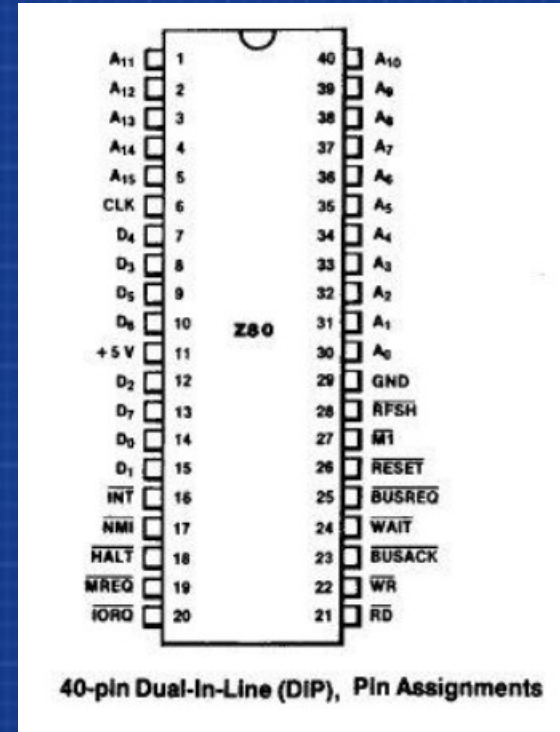
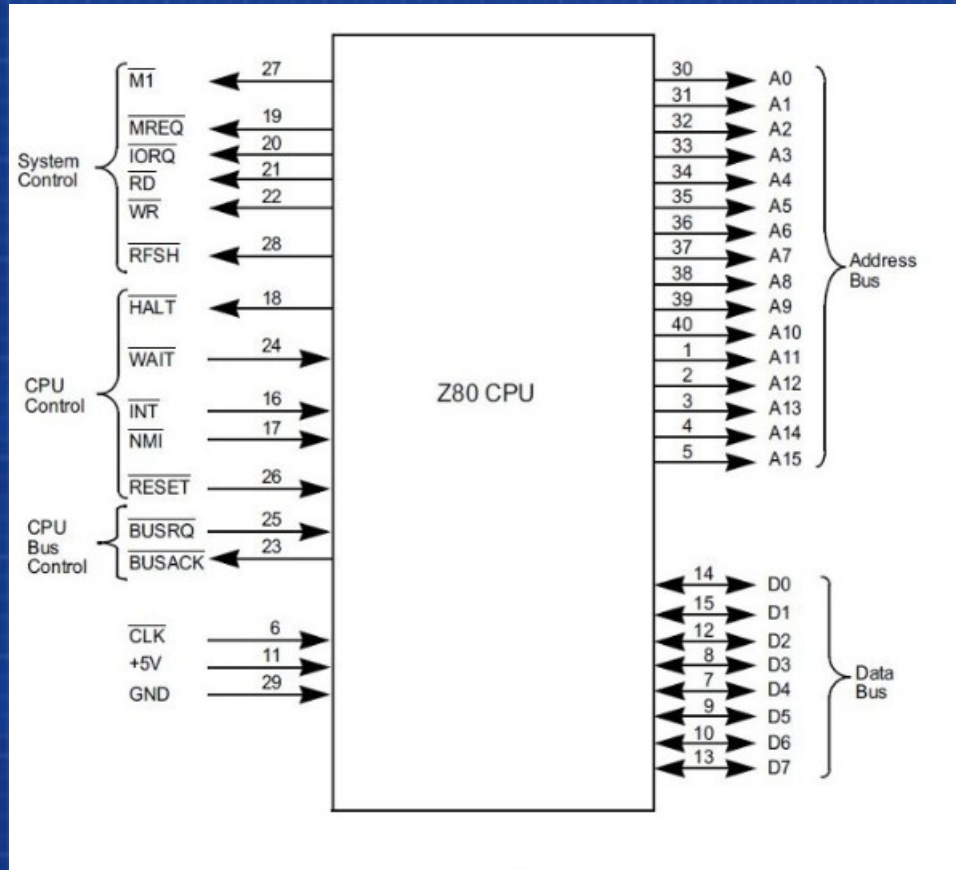
BUS DE CONTROL



BUS DE DONNEES ET BUS ADRESSE



BUS DE DONNEES ET BUS ADRESSE



INSTRUCTIONS

EXEMPLE

COMMANDE

DONNEE 1

DONNEE n

Instruction

valeur

valeur

commande

STORE

Adresse

Donnée

Hexa

I1A

I2C

1F

Binaire

10001 1010

0010 1011

0001 1111

INSTRUCTIONS EXEMPLE

Adresse
mémoire

Instruction
HEXA

Instruction
En clair

```
.data:00000000 55          push ebp
.data:00000001 31d2       xor edx,edx
.data:00000003 89e5       mov ebp,esp
.data:00000005 8b4508     mov eax,DWORD PTR [ebp+0x8]
.data:00000008 56         push esi
.data:00000009 8b750c     mov esi,DWORD PTR [ebp+0xc]
.data:0000000c 53         push ebx
.data:0000000d 8d58ff     lea ebx,[eax-0x1]
.data:00000010
.data:00000010          loop:
.data:00000010 0fb60c16  movzx ecx,BYTE PTR [esi+edx*1]
.data:00000014 884c1301  mov BYTE PTR [ebx+edx*1+0x1],cl
.data:00000018 83c201     add edx,0x1
.data:0000001b 84c9       test cl,cl
.data:0000001d 75f1       jne loop
.data:0000001f 5b         pop ebx
.data:00000020 5e         pop esi
.data:00000021 5d         pop ebp
.data:00000022 c3         ret
```

STRUCTURE DES COUCHES DE PROGRAMATION

