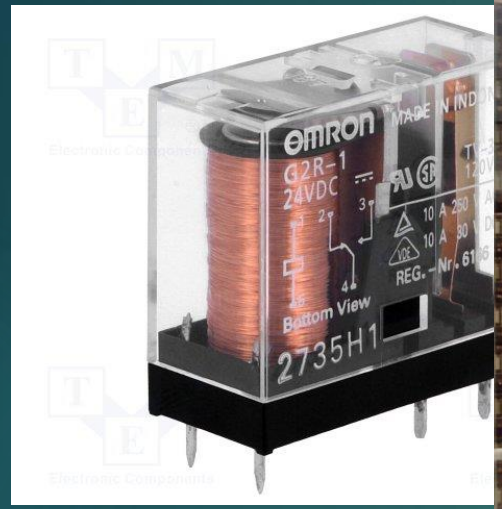
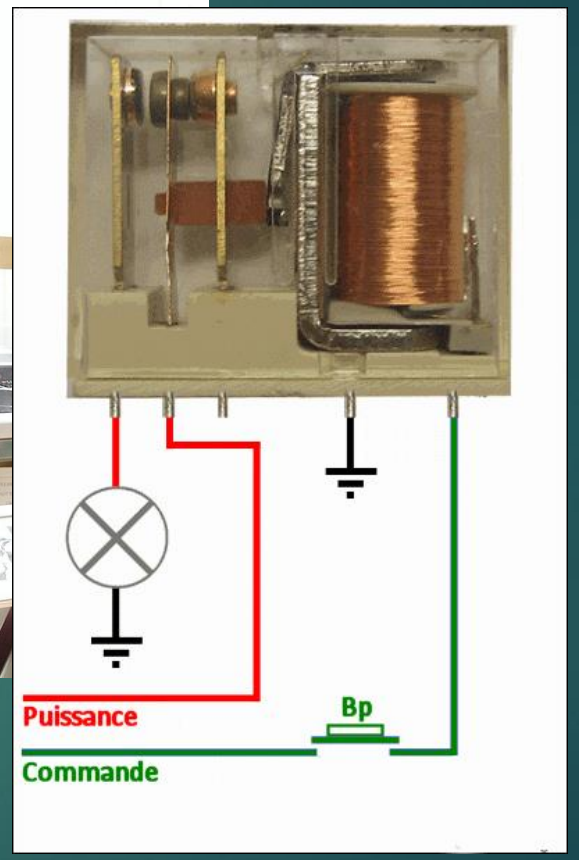
Quelques « *pourquois* ?» de l'informatique

BINAIRE, BITS, OCTETS, BYTE…DE QUOI PARLE T'ON ?

1-Pourquoi le binaire ?

Les composants utilisés dans les premiers ordinateurs furent des « relais » : (IBM Mark 1 en 1944)

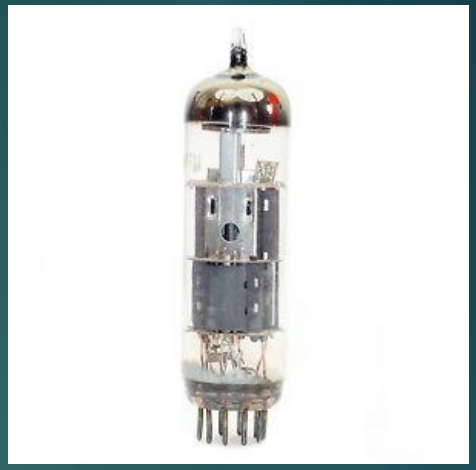




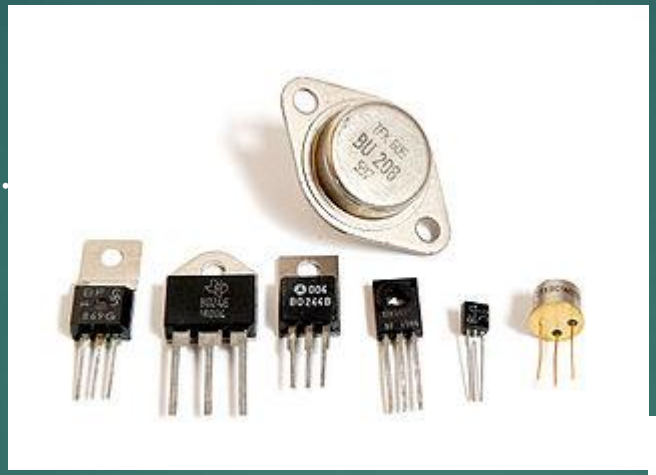
Un relais est soit « ouvert », soit « fermé »

il a donc deux états possibles…d'où une arithmétique **binaire** pour décrire son état

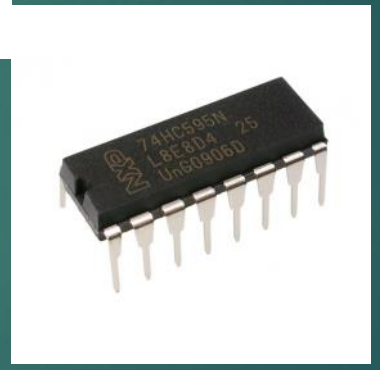
Plus tard (1940) remplacé par le « tube électronique » ou la « lampe électronique »…elle-même possédant deux états stables

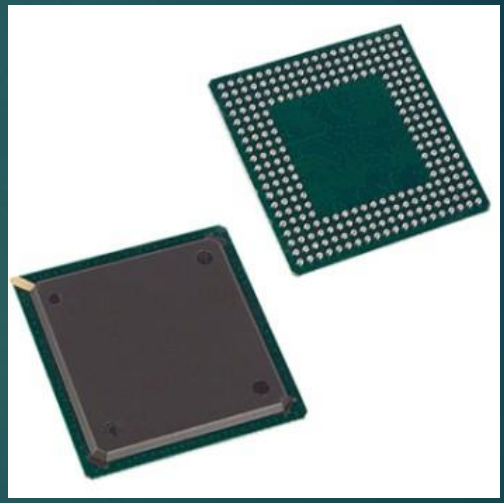


Puis par le « transistor » (1950)….



Puis enfin les « circuits intégrés » (1960) utilisant des centaines voire des millions (1980) de transistors intégrés.





Tous ces composants ayant un fonctionnement binaire l'arithmétique **binaire** s'est naturellement imposée

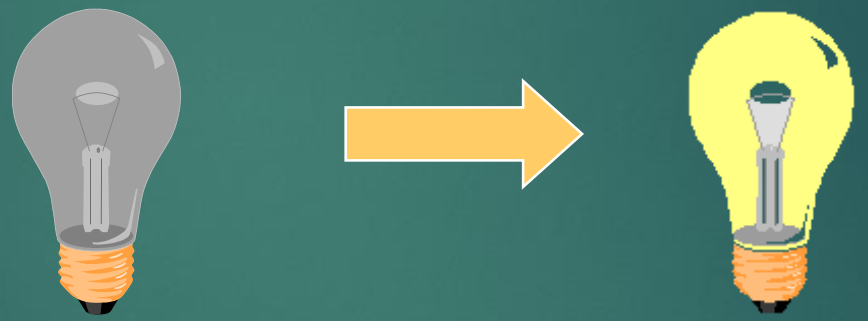
Le binaire des circuits simples

Hors tension

0

Sous tension

1



Commutateur

On peut décrire l'état du commutateur avec un nombre binaire (0 ou 1) :
c'est un « Bit » (*BInary digiT*)

La représentation binaire

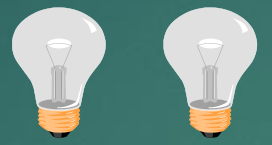
2 ampoules…quatre combinaisons

Si

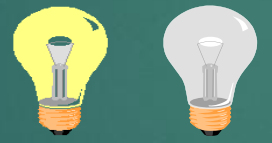
 = 0

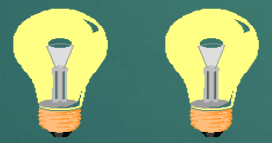
et

 = 1

 = 0

 = 1

 = 2

 = 3

4 valeurs
ou 2^2

Avec 2 bits…on peut coder quatre combinaisons différentes

La conversion en binaire

Rappel sur le système de numération décimal

…	10^4	10^3	10^2	10^1	10^0
…	10000	1000	100	10	1
Exemple :			**2**	**2**	**4**

$2x100 + 2x10 + 4x1 = 224_{10}$

La représentation binaire

…de la même façon, en binaire, on a :

…	2^{10}	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
…	1024	512	256	128	64	32	16	8	4	2	1
Exemple				**1**	**1**	**0**	**1**	**1**	**1**	**0**	**0**

128 + 64 +16 +8 +4 =224

$$1101\ 1100_2 = 224_{10}$$

Conversion base 2 vers base 10

(binaire vers décimal)

Que vaut 1101 1100_2 en décimal ?

...	2^{10}	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
...	1024	512	256	128	64	32	16	8	4	2	1

Exemple de conversion décimal/binaire

- **Comment écrire le nombre décimal 56 en binaire (base 2) ?**
  - On cherche la plus grande puissance de 2 comprise dans 56.
    - C'est 2^5 soit 32
  - On soustrait cette valeur 32 à 56, il reste 56-32= 24
  - On cherche la plus grande puissance de 2 comprise dans 24
    - C'est 2^4 soit 16
  - On soustrait cette valeur 16 à 24, il reste 24-16=8
  - On cherche la plus grande puissance de 2 comprise dans 8
    - C'est 2^3 soit 8
  - On soustrait cette valeur 8 à 8, il reste 8-8=0
  - La conversion est terminée et 56 peut s'écrire comme 56=32+16+8 qui sont des puissances de 2

    Ainsi, **56 (en base 10) s'écrit donc 11100 en base 2** et exige 5 bits pour pouvoir être transcrit

- **Convertir 98 en base 2**

- **Que vaut $1101\ 1100_2$ en base 10 ?**

2 -*Pourquoi* on parle d' « *octets* » ?

- Octet vient du grec « octo » (8)…1 octet = 8 bits
- Avec 8 bits, on peut coder 2^8 nombres binaires différents soit 256 caractères différents
- Le « code ASCII » (sur 8 bits) permettait de coder toutes les lettres et symboles courant. Exemple « a »=11000001

DEC	ASCII	BINAIRE	DEC	ASCII	BINAIRE	DEC	ASCII	BINAIRE
33	!	0010 0001	64	@	0100 0000	95	_	0101 1111
34	"	0010 0010	65	A	0100 0001	96	`	0110 0000
35	#	0010 0011	66	B	0100 0010	97	a	0110 0001
36	$	0010 0100	67	C	0100 0011	98	b	0110 0010
37	%	0010 0101	68	D	0100 0100	99	c	0110 0011
38	&	0010 0110	69	E	0100 0101	100	d	0110 0100
39		0010 0111	70	F	0100 0110	101	e	0110 0101
40	(	0010 1000	71	G	0100 0111	102	f	0110 0110
41	)	0010 1001	72	H	0100 1000	103	g	0110 0111
42	*	0010 1010	73	I	0100 1001	104	h	0110 1000

Cela suffisait

…en 1960.

D'autres codages ont été inventés pour la couleur (VESA,RGB,...), le son (mp3, FLAC...), l'image (gif, jpg,png...), la vidéo (avi,mpeg...)

Exemple, codage couleur d'un pixel sur 24 bits (png) :

Blanc = FFFFFF ou 1111 1111 1111 1111 1111 1111

Rouge = FF0000 ou 1111 1111 0000 0000 0000 0000

Multiples : Octets, kilo-octets, méga-octets, giga-octets….

Taille	Désignation	Nombre de valeurs possibles
8 bits	Octet	256
10 bits	Kilo-octet(kO)	1 024
20 bits	Méga-octet(MO)	1 048 576
30 bits	Giga-octet (GO)	1 073 741 824

3- *Pourquoi* 1kO (kilo-octet) fait 1024 octets (…et pas 1000) ?

2 bits pour adresser une mémoire ⇨ 4 emplacements (adresses) possibles

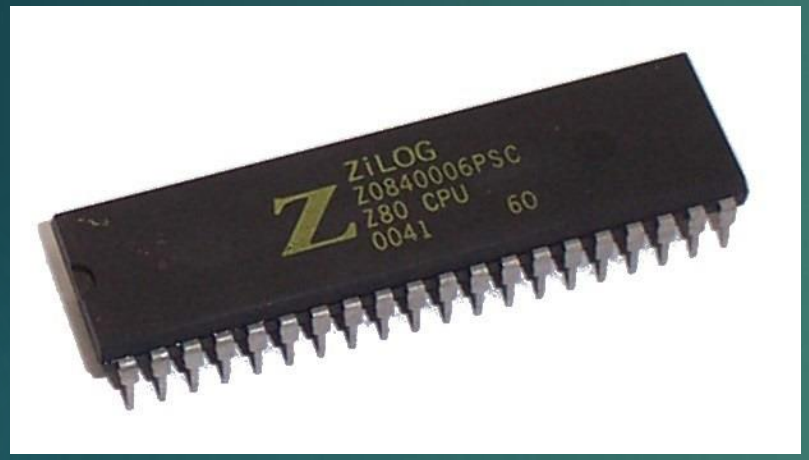
4 bits pour adresser une mémoire ⇨ 16 emplacements (adresses) possibles

.
.
.

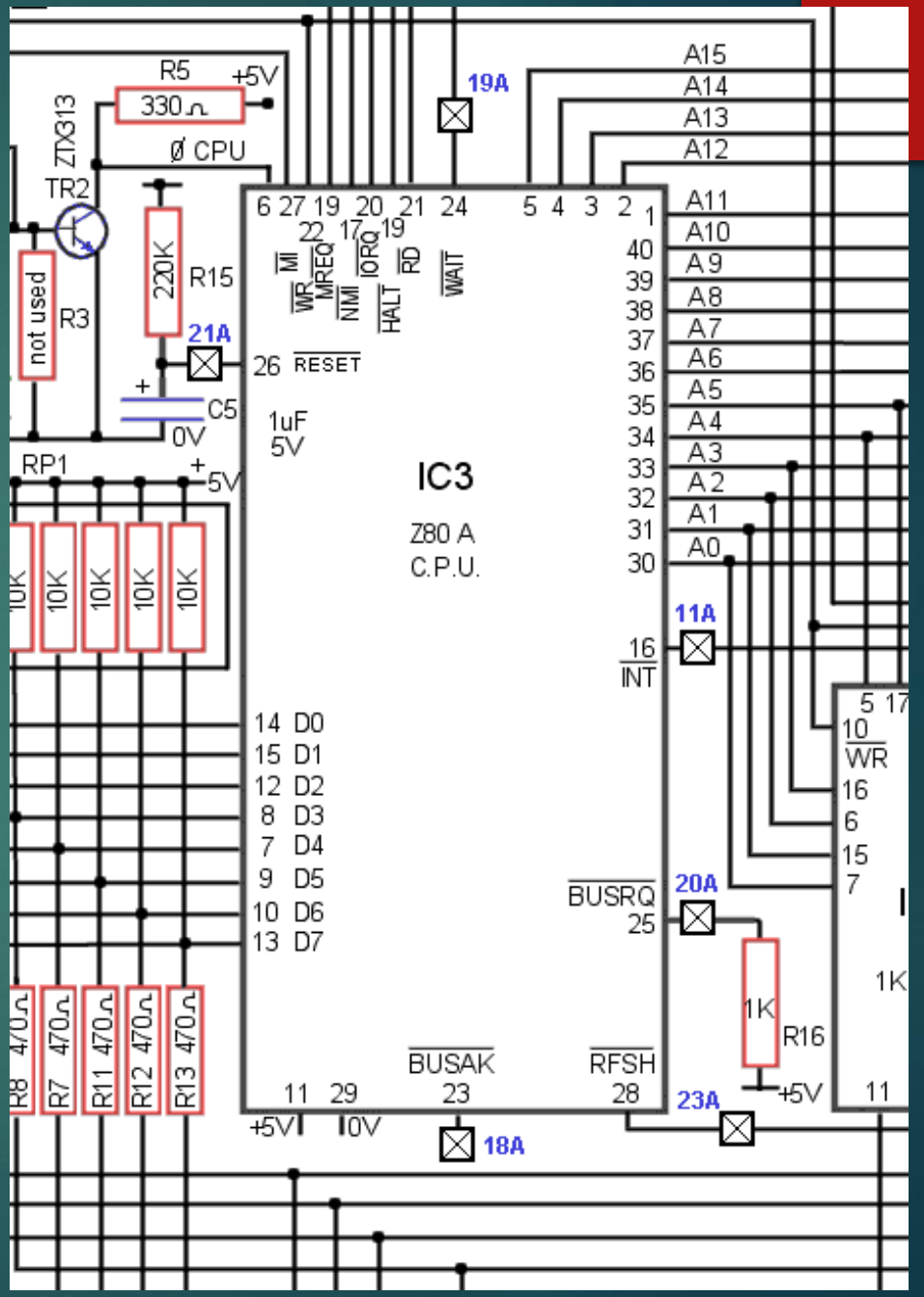
n bits pour adresser une mémoire ⇨ 2^n emplacements (adresses) possibles

Si le boitier à 10 bits d'adresse, on peut adresser 2^{10} octets soit 1024 octet, c'est ainsi que l'on a baptisé « kilo Octets » un volume mémoire de 1024 octet…même s'il ne correspond pas exactement à 1000 octets

Exemple de Mémoire adressable



Microprocesseur ZILOG Z80
Microprocesseur « 8 bits »
15 bits d'adresse, donc … emplacements adressables



Exemple de structure d'un ordinateur

Micro-ordinateur Sinclair ZX81 (1982)