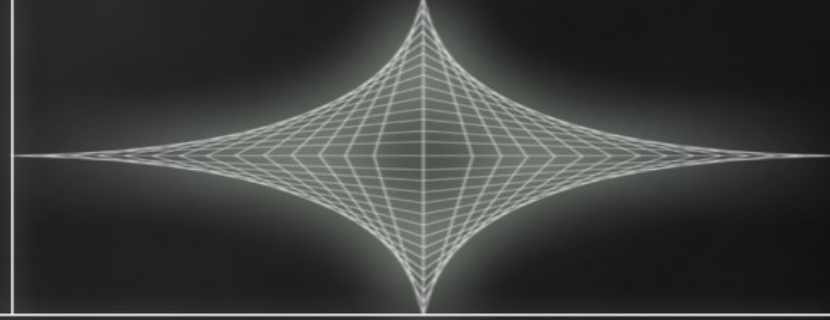
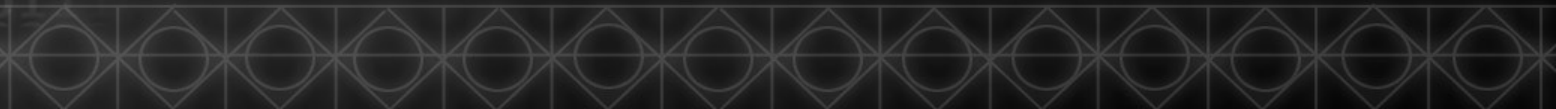


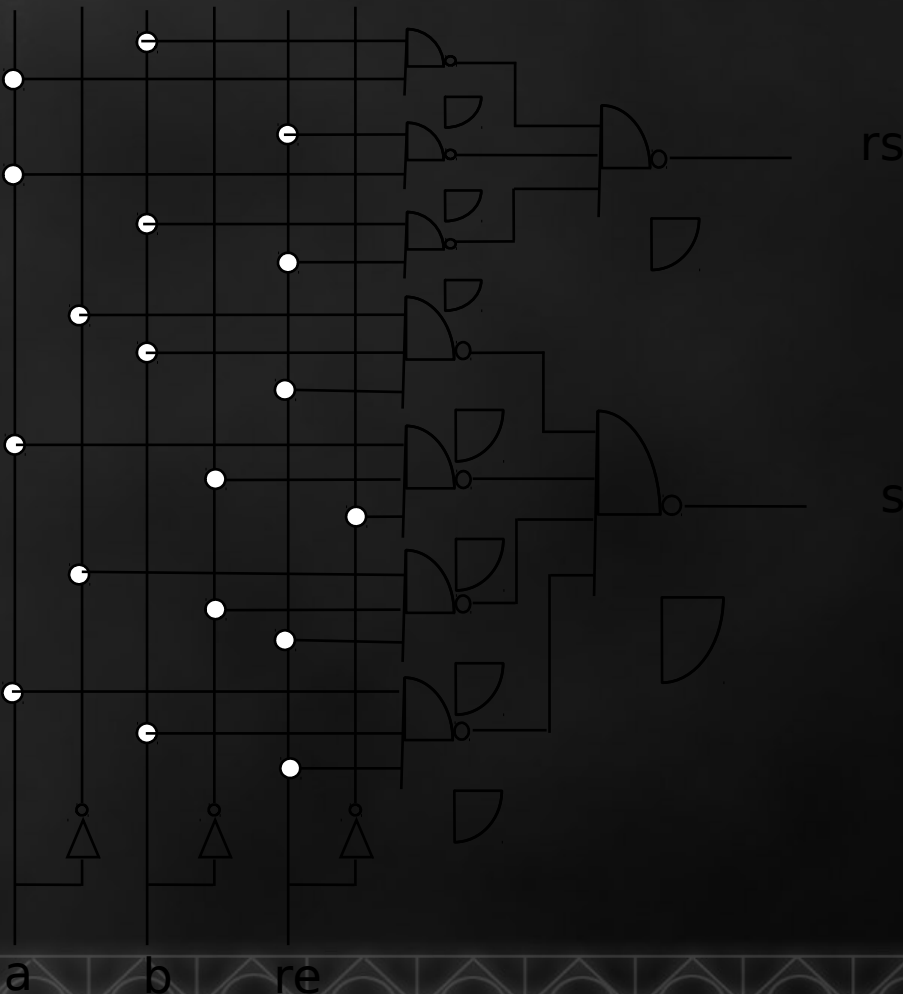
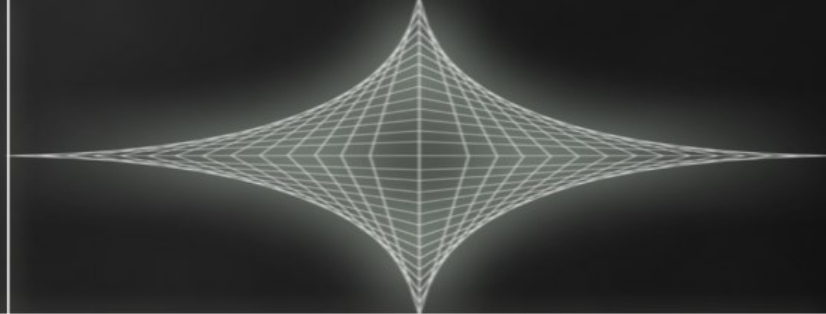
AE 5



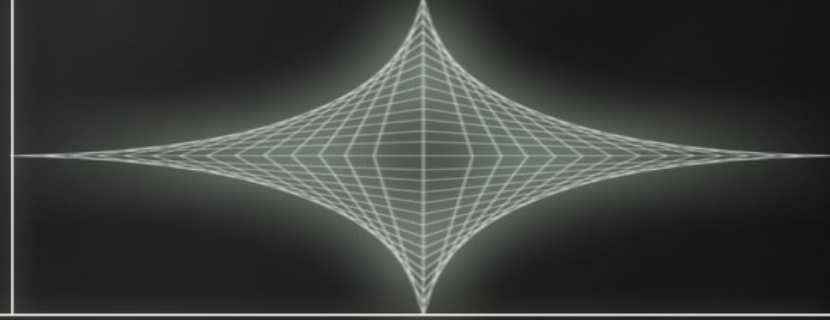
# Synthèse d'un circuit séquentiel



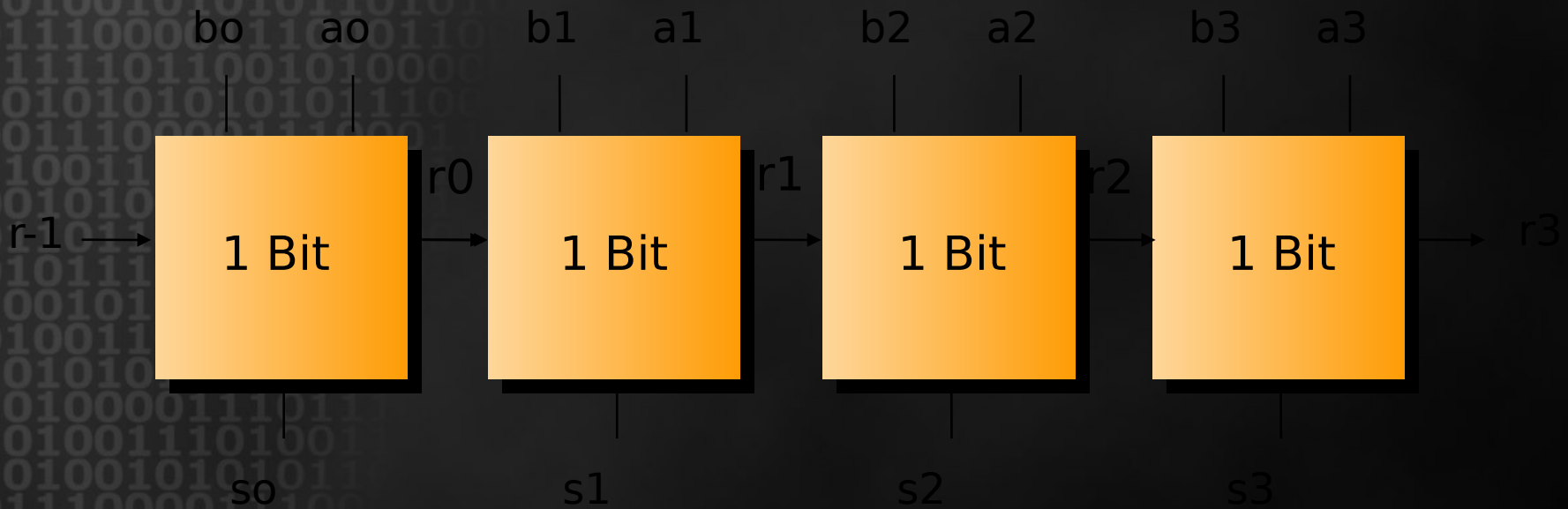
# L'additionneur 1 bit



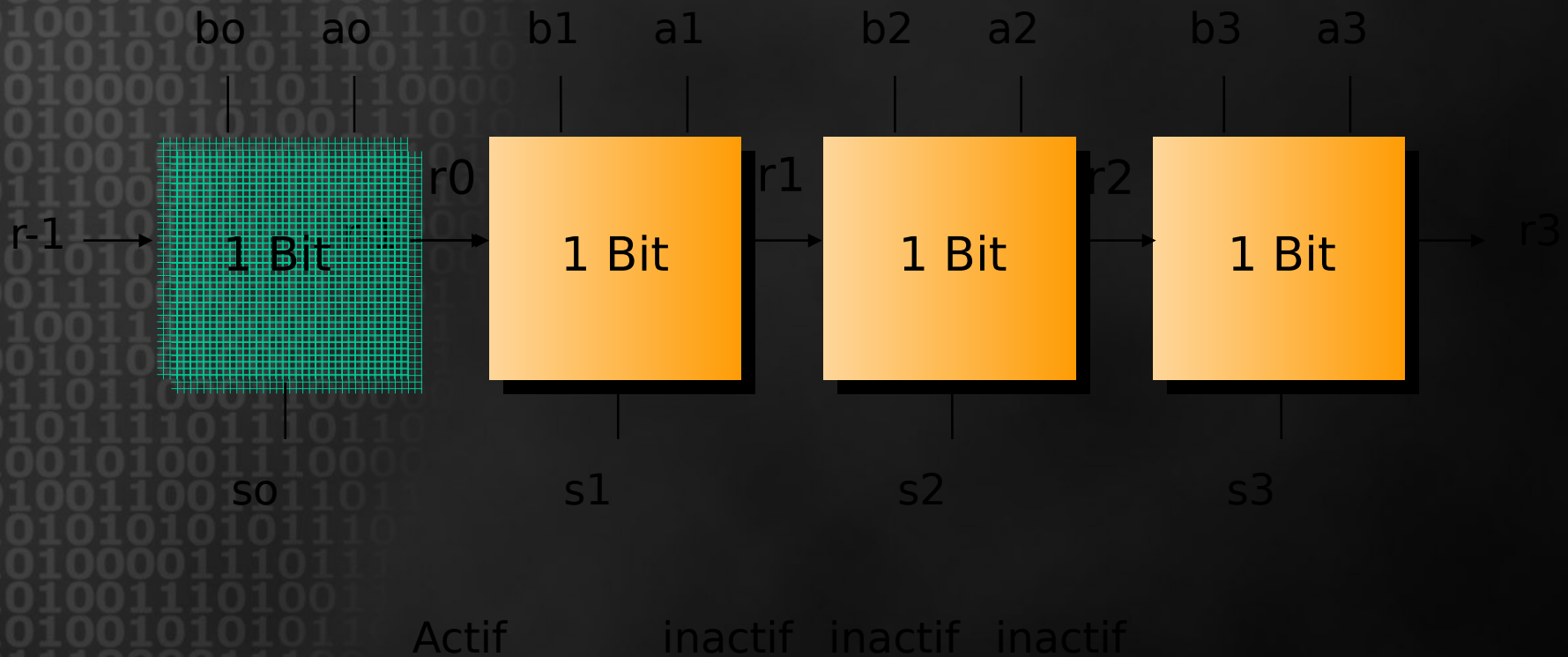
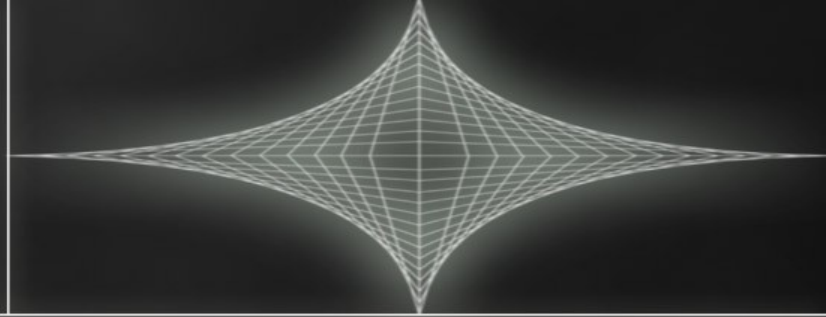
# L'additionneur 4 bits à propagation



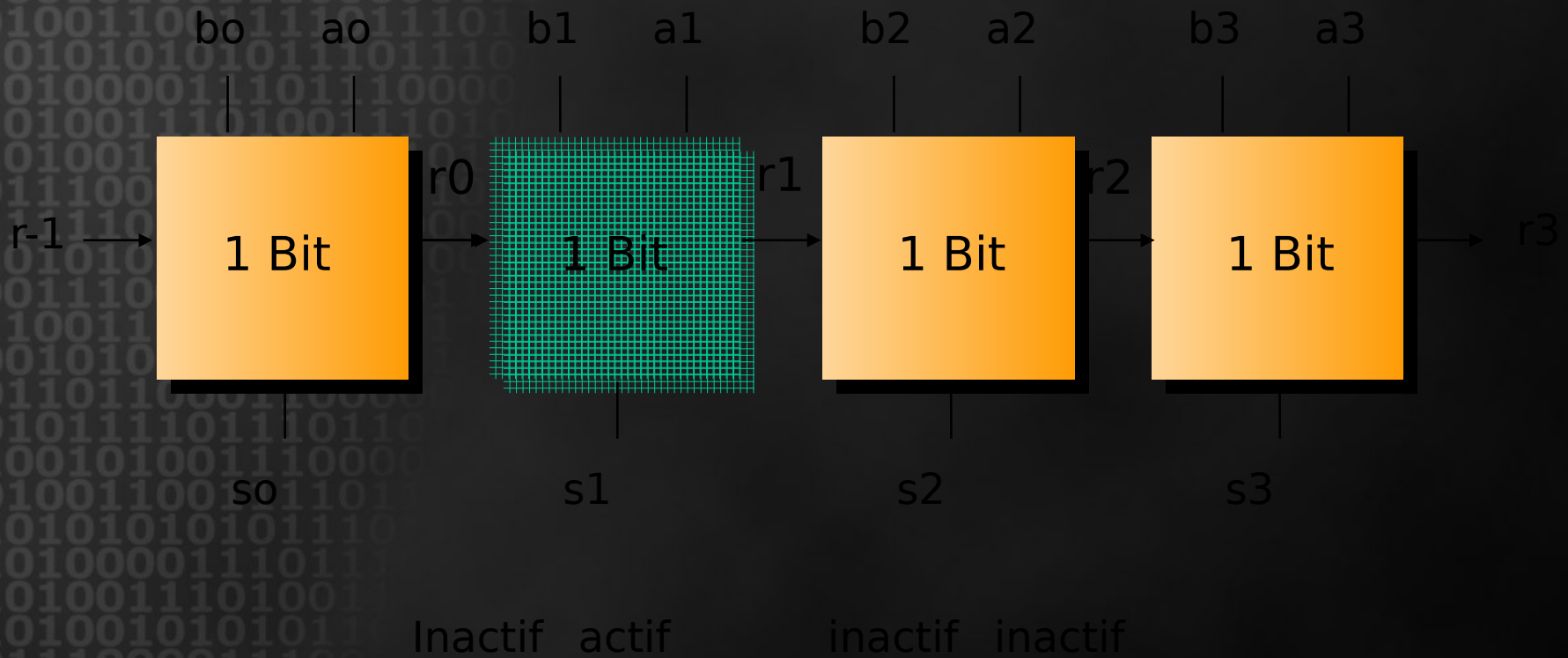
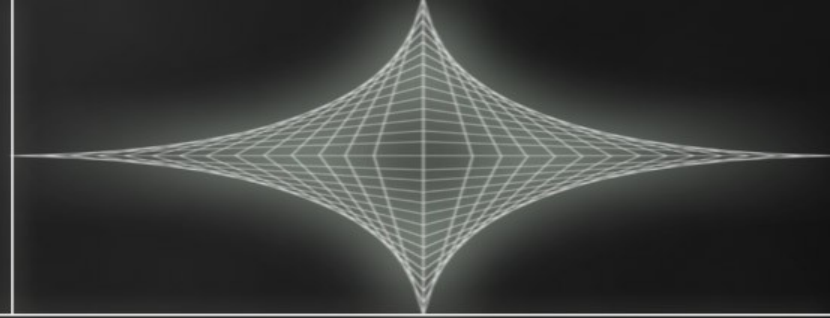
Les activités des additionneurs 1 bit en fonction du temps.



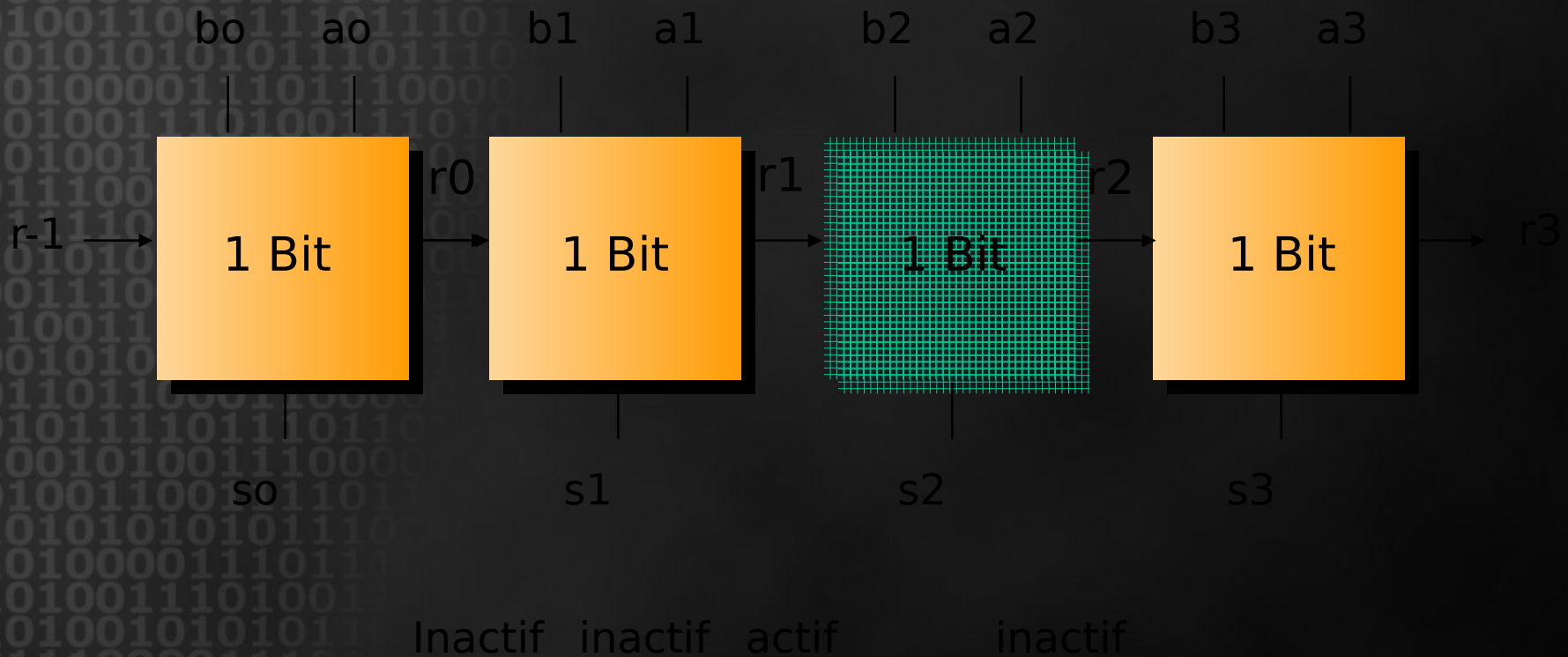
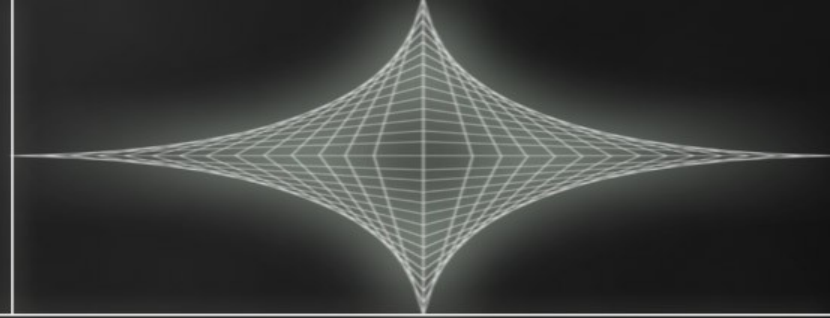
# L'additionneur 4 bits à propagation



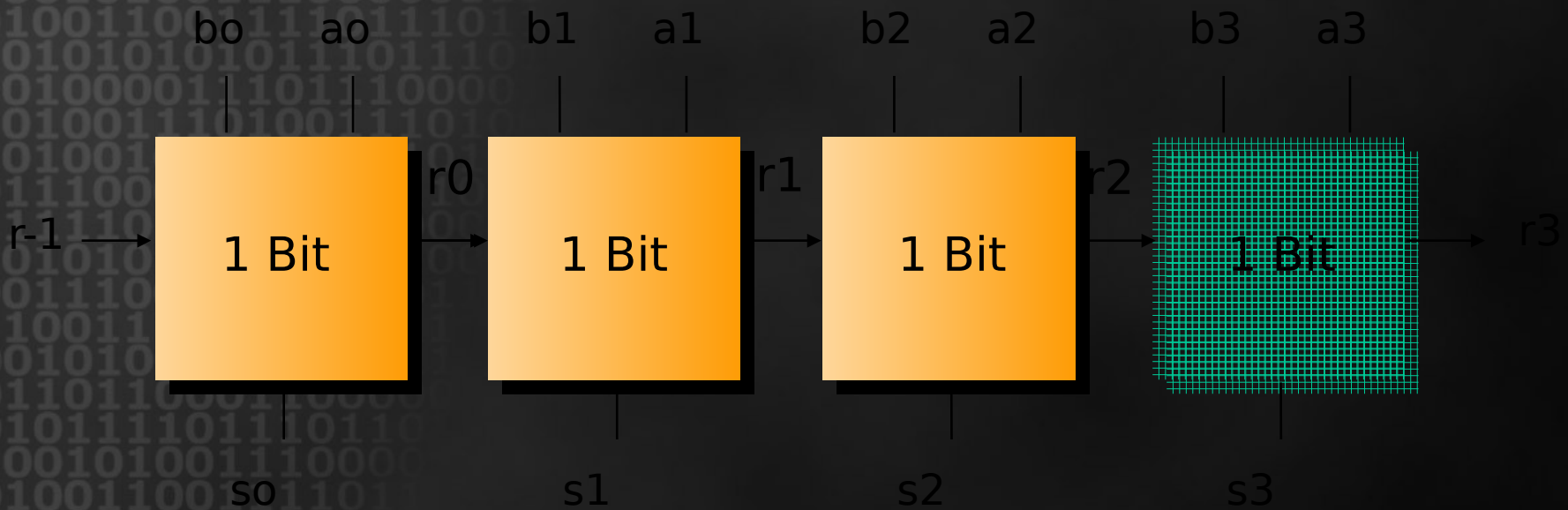
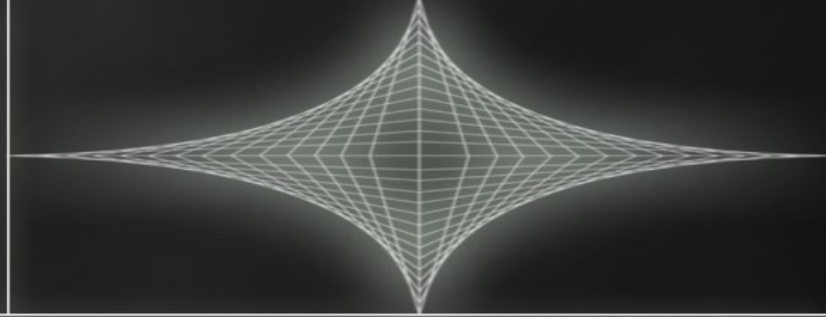
# L'additionneur 4 bits à propagation



# L'additionneur 4 bits à propagation

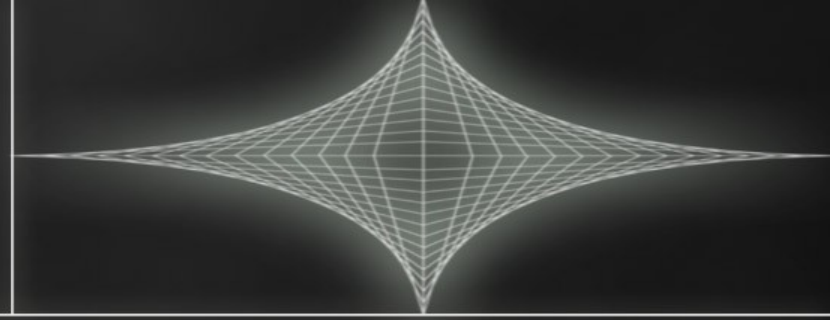


# L'additionneur 4 bits à propagation



Inactif inactif inactif actif

# L'additionneur



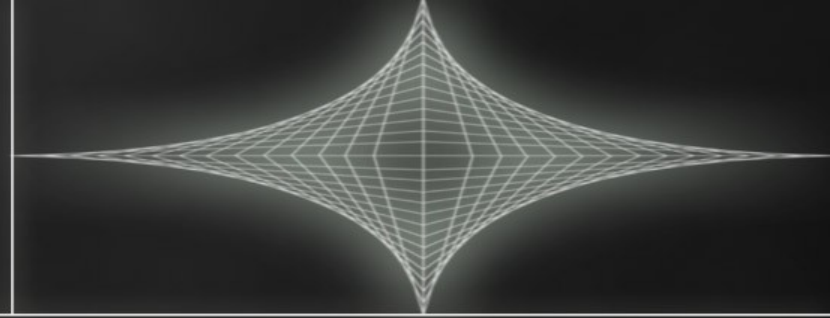
A chaque intervalle de temps un seul additionneur est actif!

Comment réaliser un circuit avec une seul additionneur?

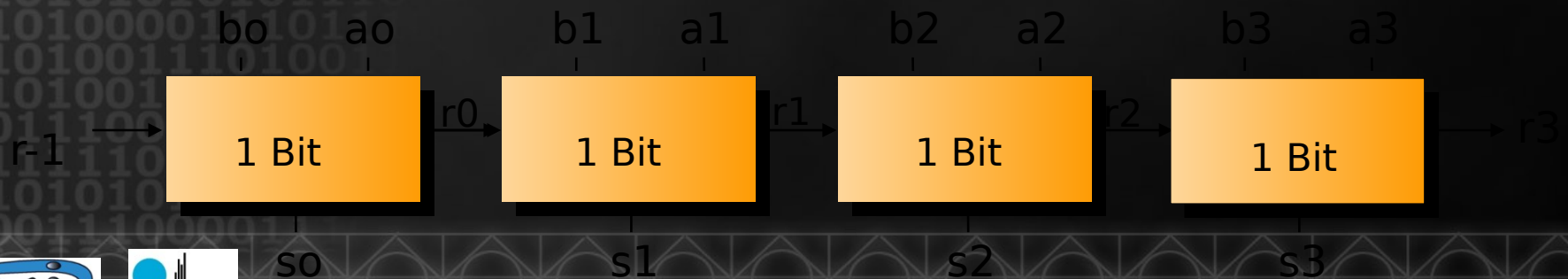




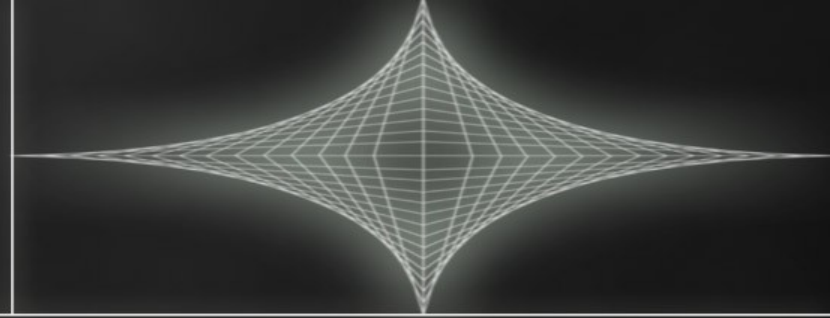
# L'additionneur



fonctionnement en fonction du temps.



# L'additionneur



t0

a0

b0

r-1



1 Bit

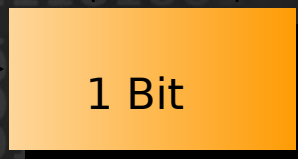
bo ao

b1 a1

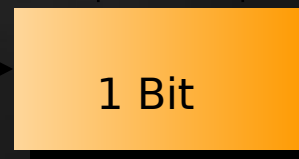
b2 a2

b3 a3

r-1



1 Bit



1 Bit



1 Bit



1 Bit

t0

s0

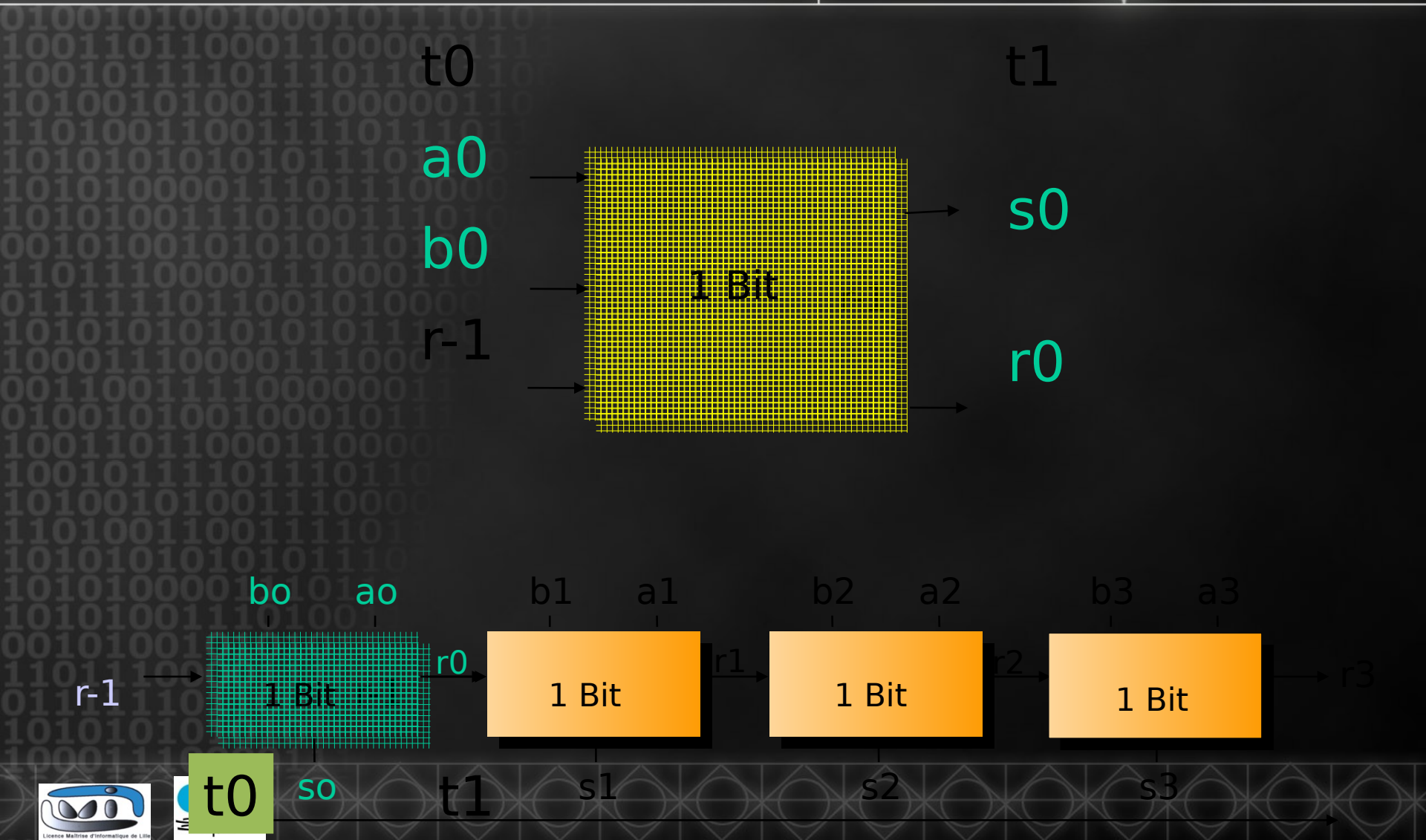
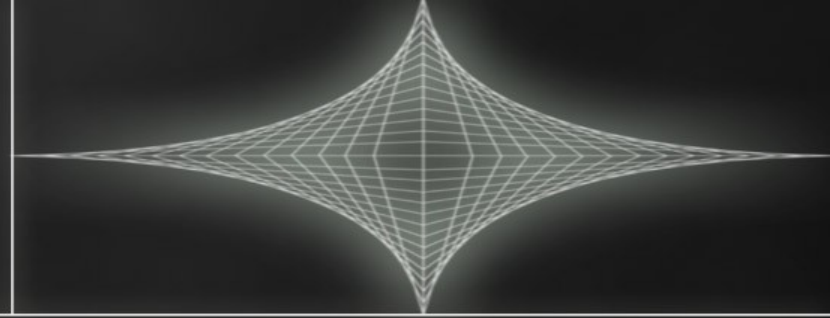
s1

s2

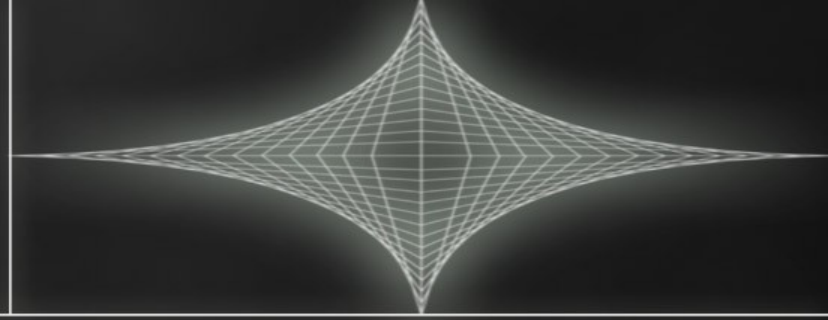
s3



# L'additionneur



# L'additionneur

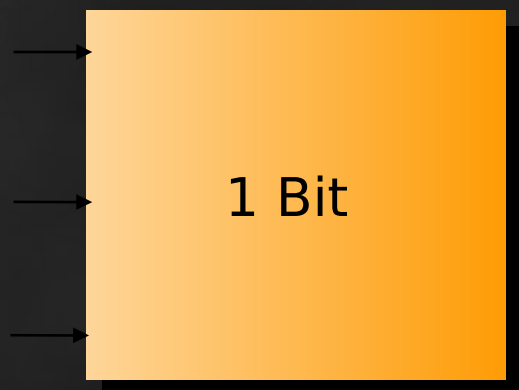


t1

a1

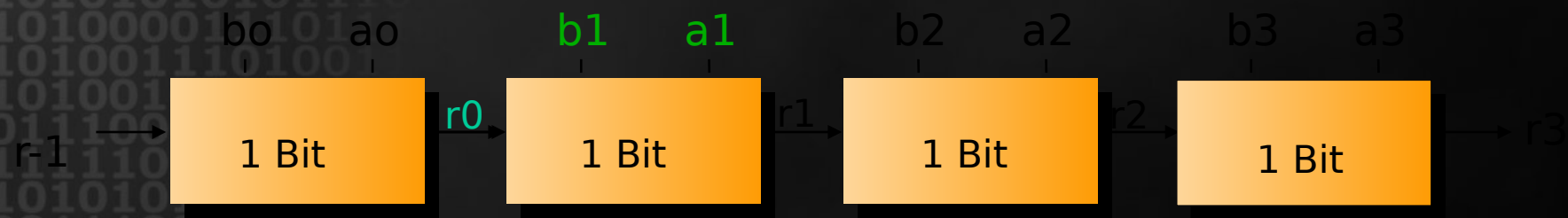
b1

r0



s0

r0



t0

s0

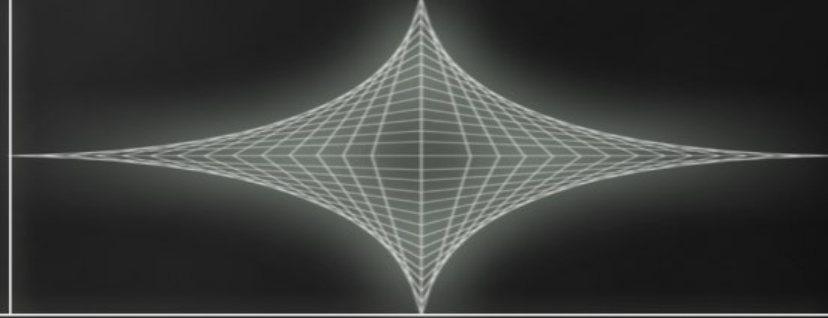
t1

s1

s2

s3

# L'additionneur



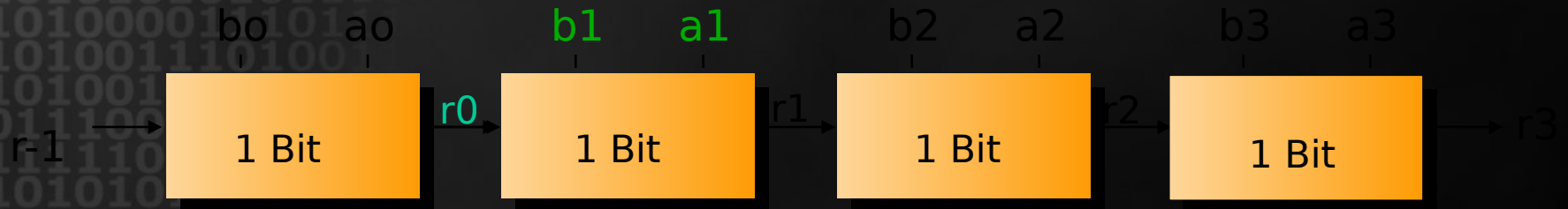
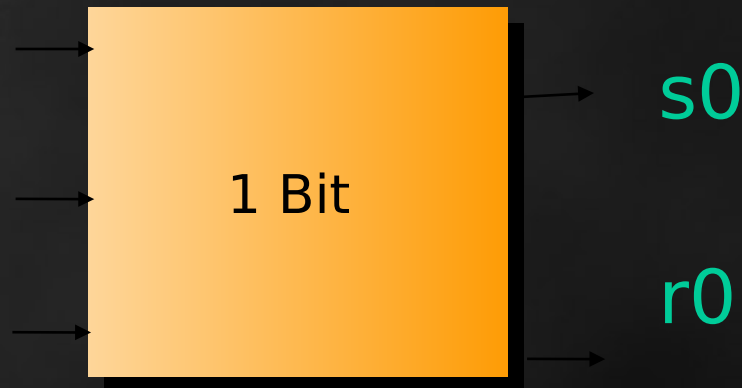
La retenue doit être disponible en entrée.

t1

a1

b1

r0



t0

s0

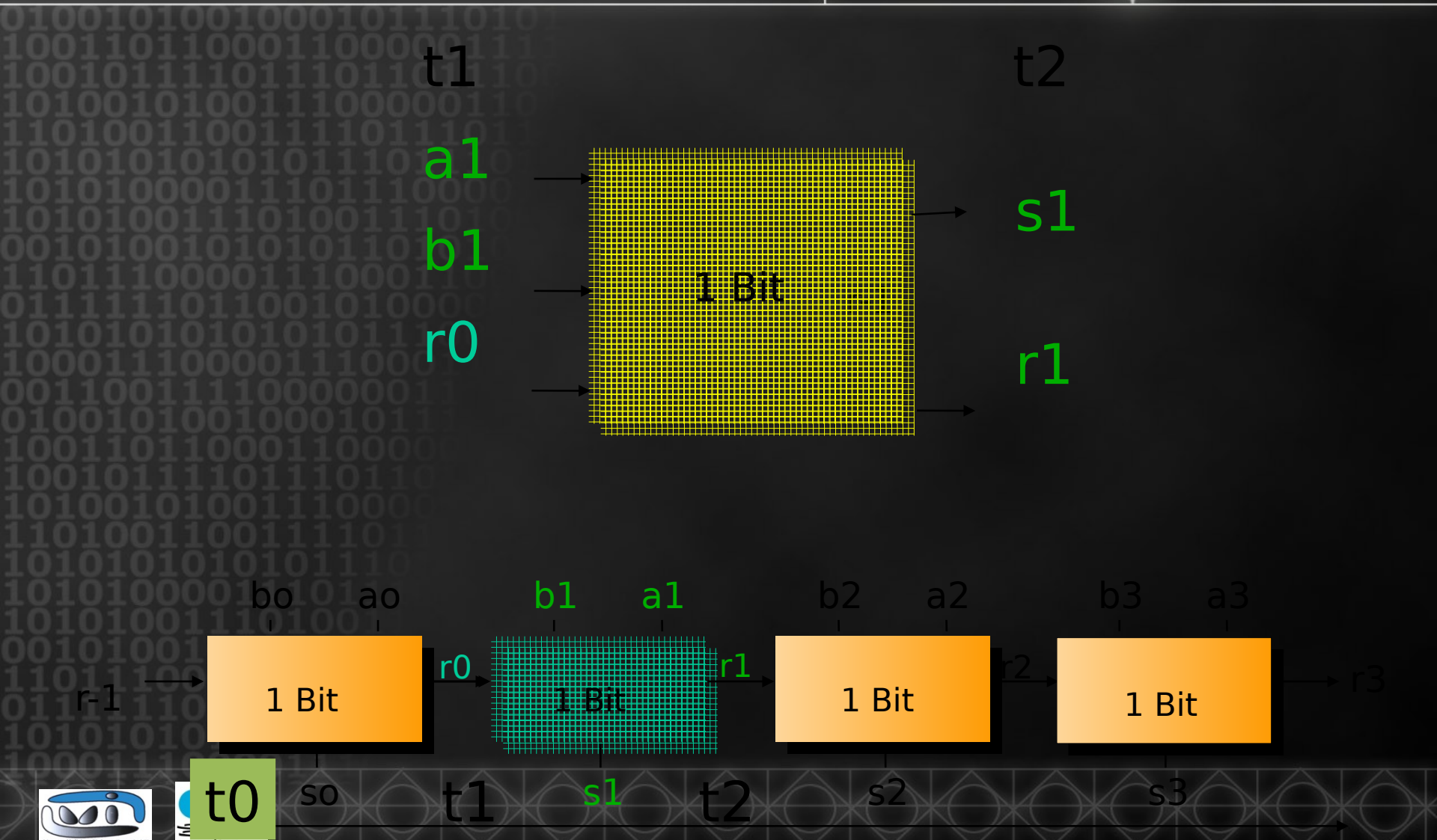
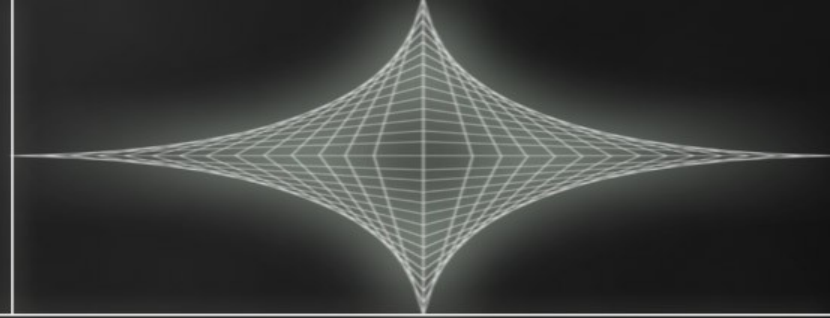
t1

s1

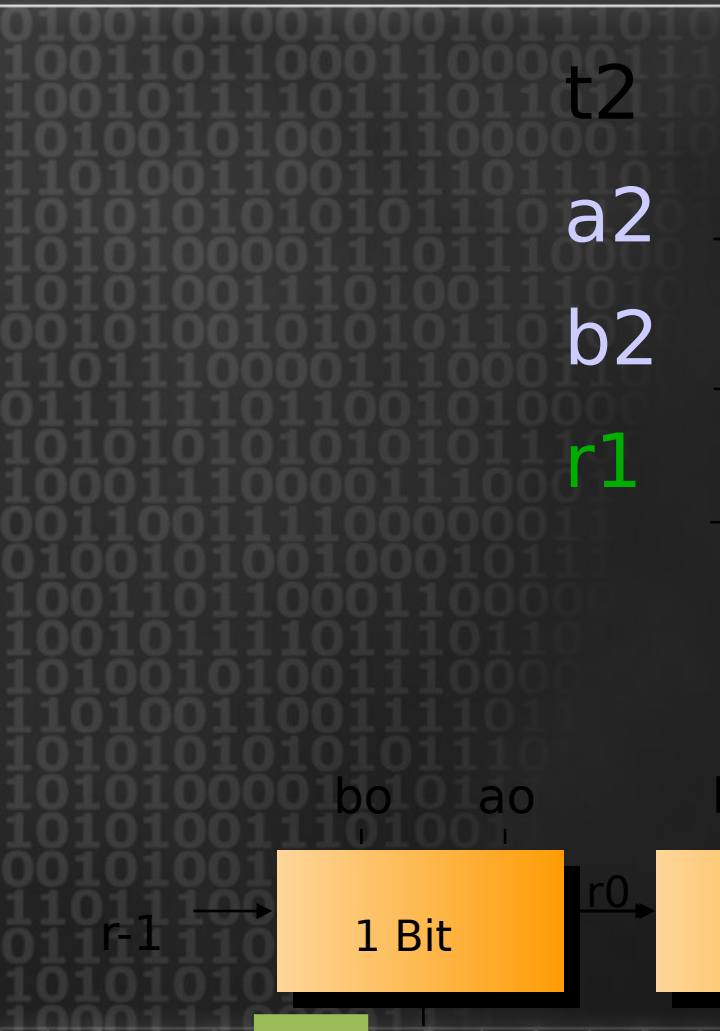
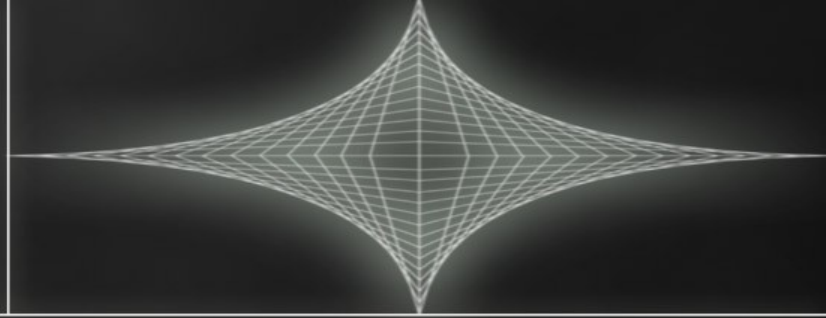
s2

s3

# L'additionneur



# L'additionneur



t2

a2

b2

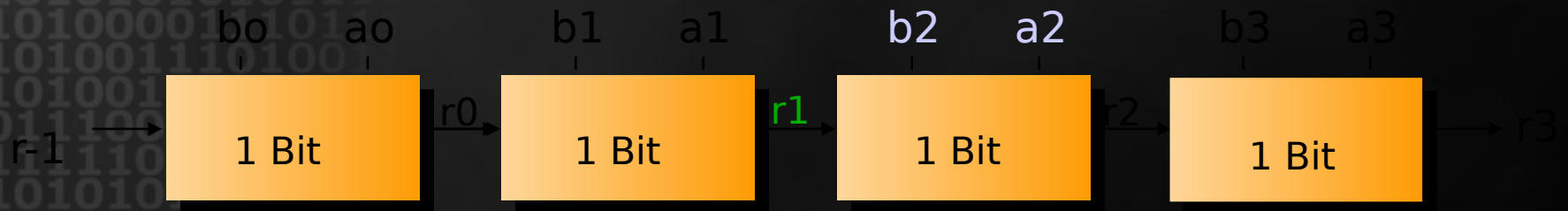
r1



1 Bit

s1

r1



t0

s0

t1

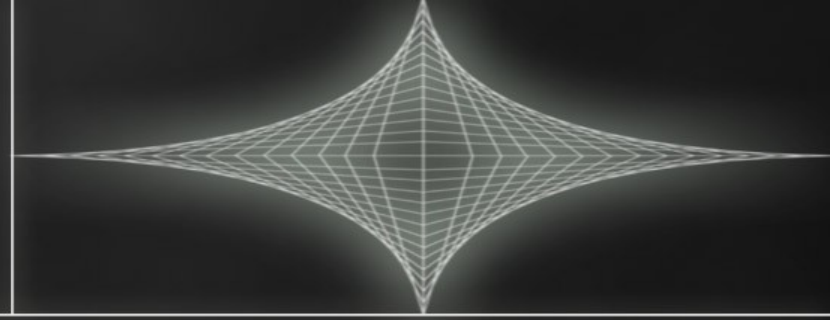
s1

t2

s2

s3

# L'additionneur

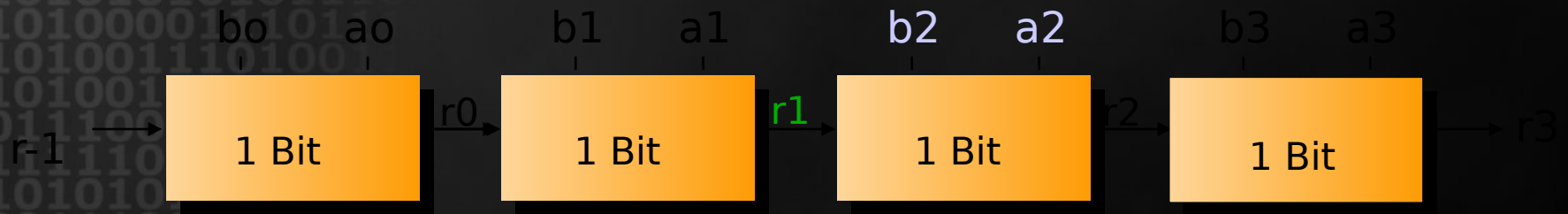


t2

a2

b2

r1



s0

t1

s1

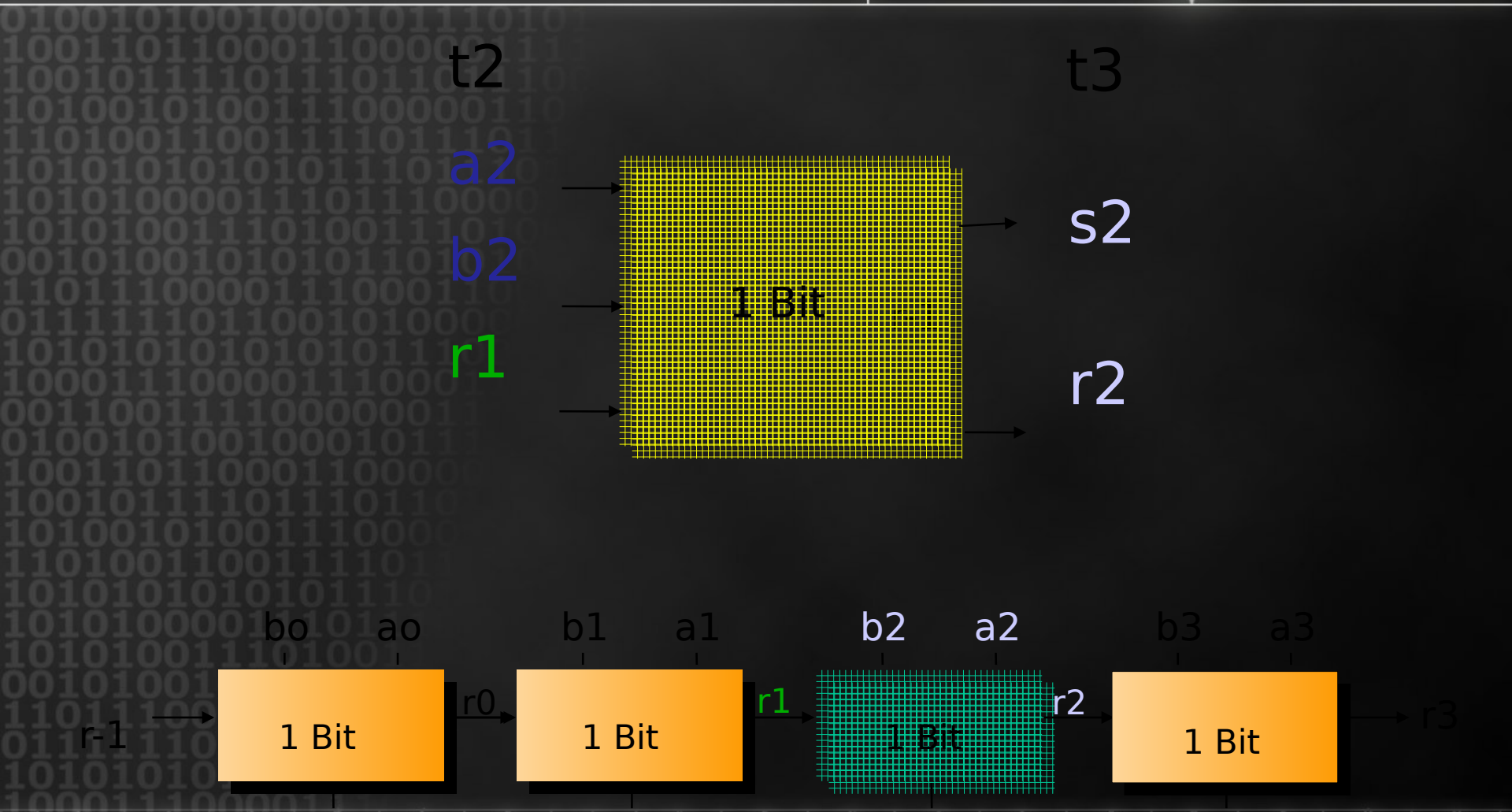
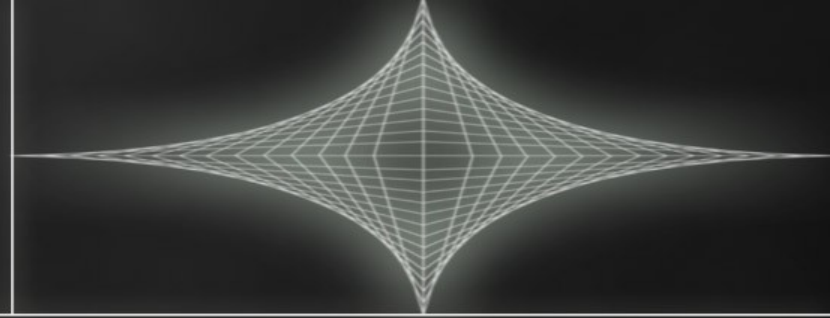
t2

s2

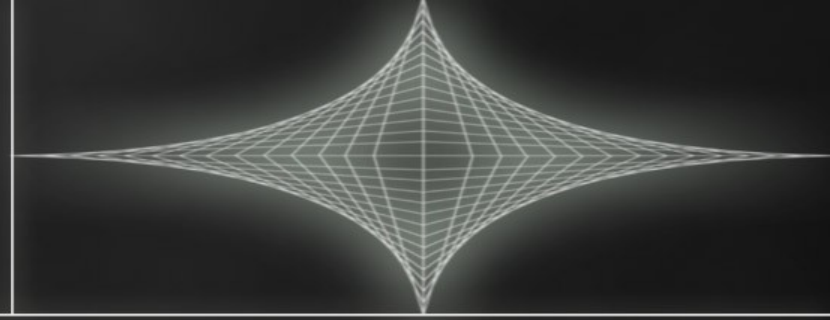
s3



# L'additionneur



# L'additionneur



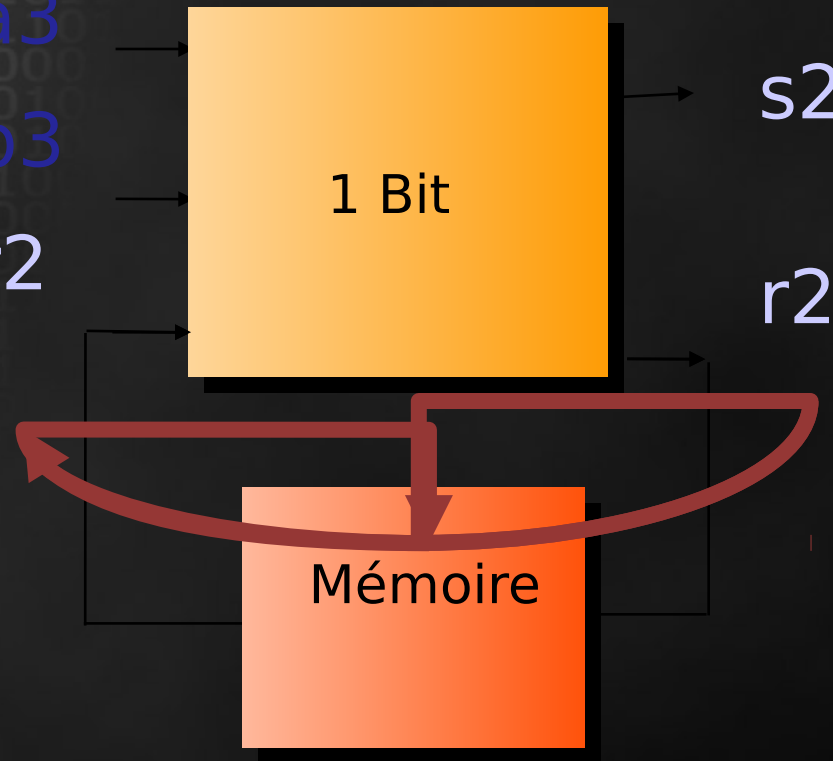
L'utilisation ici  
d'un registre  
permet de  
mémoriser la  
retenue t-1.

t3

a3

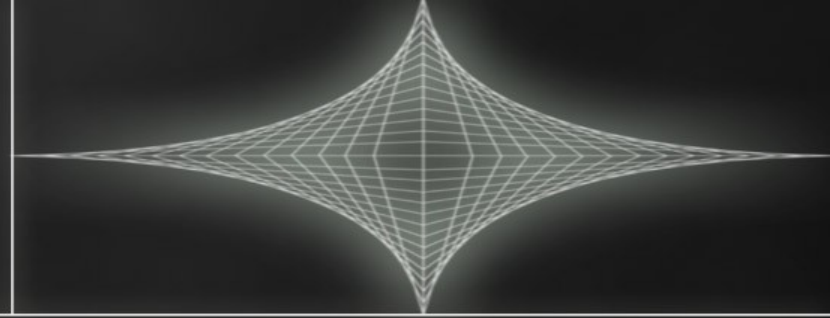
b3

r2

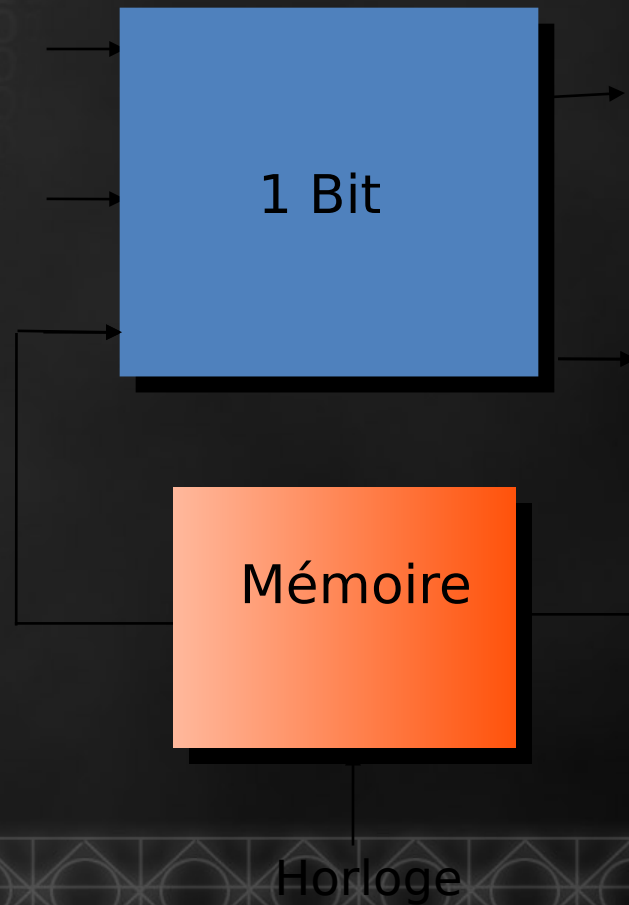


Horloge

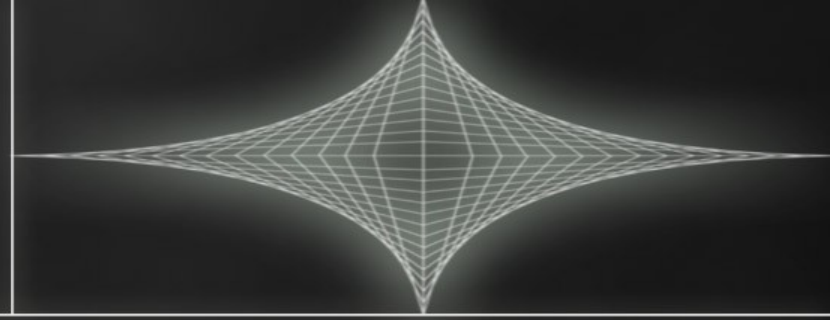
# L'additionneur



Réalisation d'un additionneur à partir d'un circuit séquentiel.

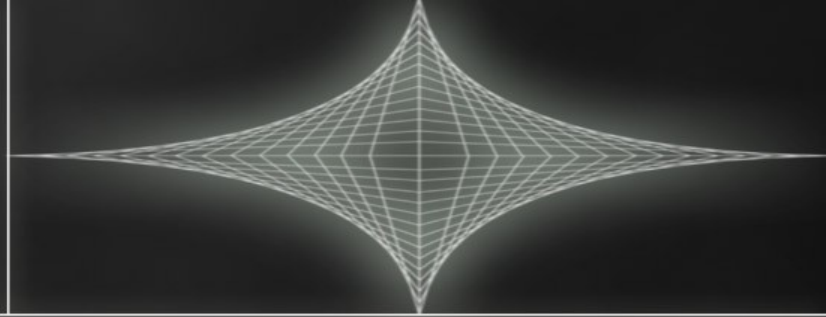


# Les circuits séquentiels



- Un circuit séquentiel est un circuit dont les sorties dépendent des entrées et de l'état du système.
- **Etat : ce qu'il faut mémoriser de l'histoire du passé, c-a-d jusqu'à l'instant  $t+1$ , pour pouvoir déterminer les sorties présentes  $S(t)$ .**

# Les automates d'états finis



$e(t_i)$

$q(t_i)$

Logique combinatoire

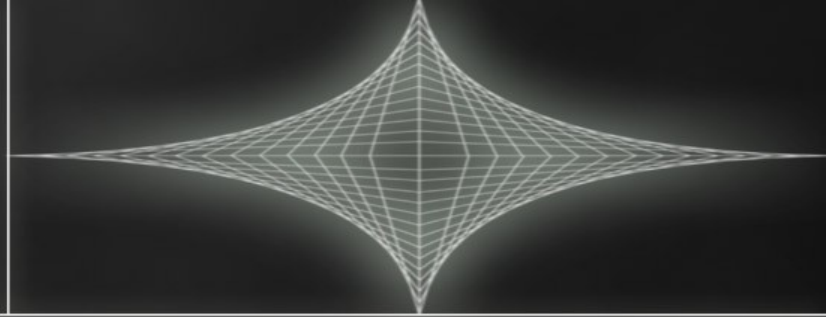
$s(t_i+1)$

$q(t_i+1)$

état de l'automate

Horloge

# Les automates d'états finis



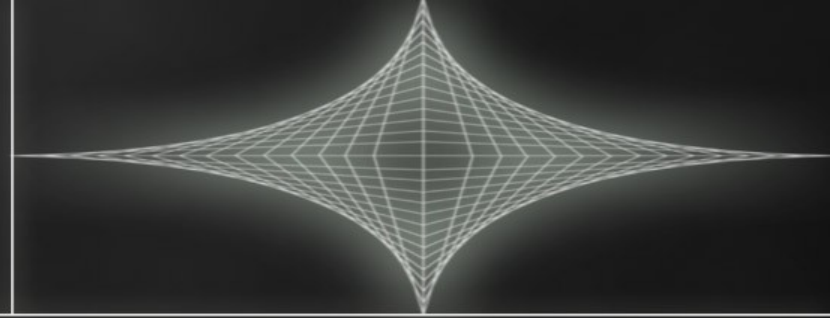
$e(t_i)$

Automate d'états finis

$s(t_i + 1)$

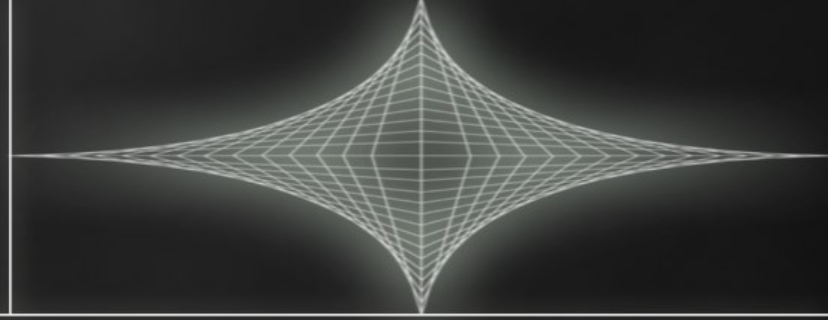
Horloge

# Les automates d'états finis



- Un automate est un être mathématique dont la réponse à un stimulus extérieur dépend de ce stimulus et de l'état interne de l'automate.
- Un automate fini a un nombre fini d'états internes. Les stimulus sont susceptibles de faire passer l'automate d'un état à un autre état.
- L'automate est entièrement déterminé par la donnée de ses fonctions de transition qui fournissent le nouvel état et la réponse en fonction de l'ancien état et du stimuli.

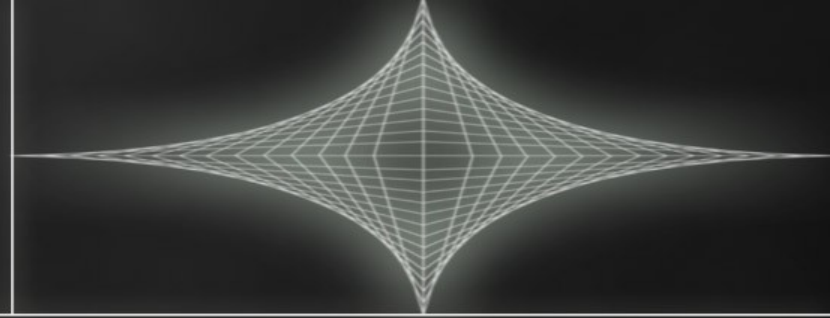
# Synthèse d'un circuit séquentiel



- Pour réaliser la synthèse d'un circuit séquentiel il faut :
  - 1 déterminer le graphe des états (diagramme de transitions) ;
  - 2 déterminer le nombre de bascules ;
  - 3 construire la table d'états ;
  - 4 réaliser les circuits combinatoires associés aux entrées des bascules et aux sorties



# Synthèse d'un additionneur

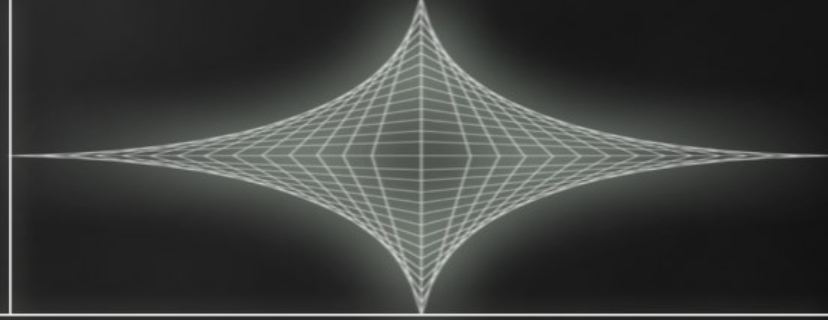


Nous allons tenter de réaliser la synthèse d'un additionneur à l'aide d'un circuit séquentiel.

Vue externe



# Diagramme de transition : graphe d'état

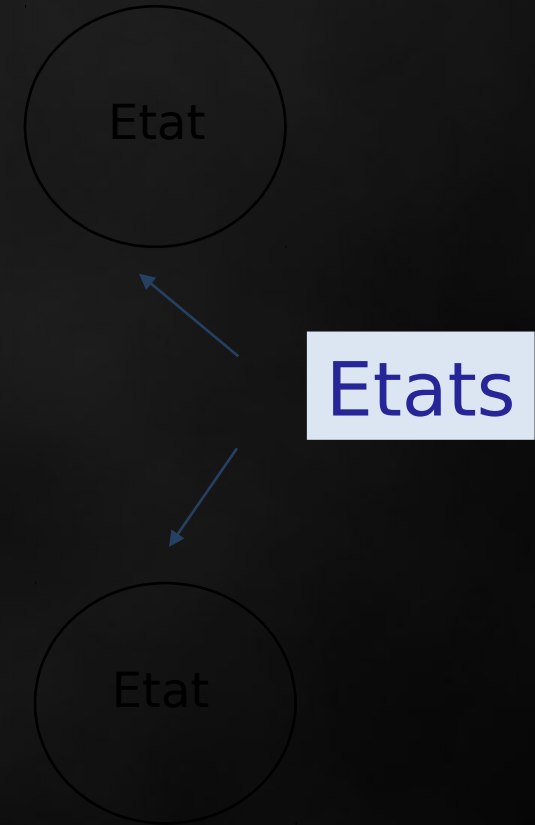


**État** : ce qu'il faut mémoriser de l'histoire du passé, c-a-d jusqu'à l'instant  $t+1$ , pour pouvoir déterminer les sorties présentes  $S(t)$

Dans notre exemple , il y a deux états internes

Etat1 = Retenue ;

Etat2 = Pasretenue.



# Synthèse d'un additionneur (graphe d'état)

Après avoir défini les états, il faut compléter le graphe par les **transitions** du systèmes.

Une fonction de transition définit l'évolution d'un automate sous l'effet d'un stimulus externe.



Entrées/sortie



Transition

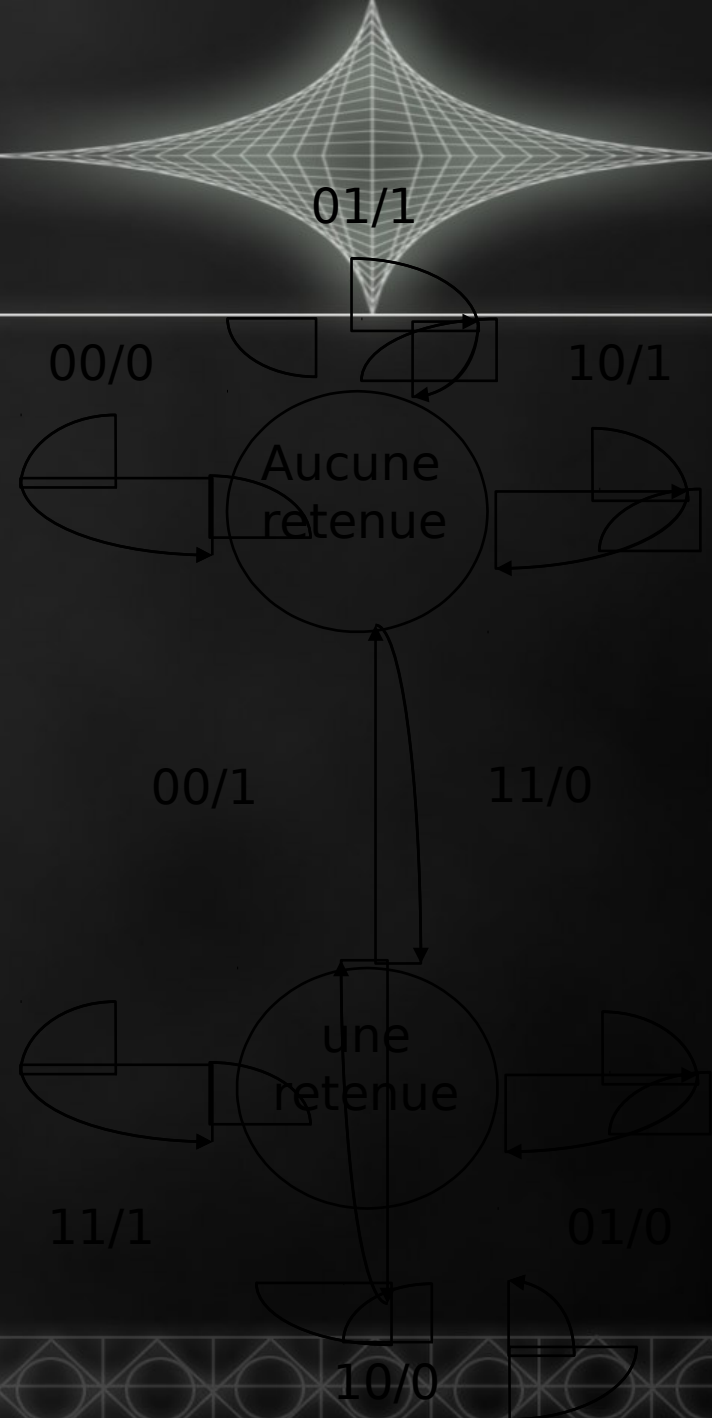
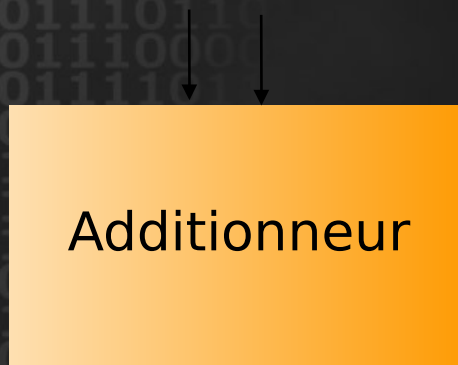
Etats



# Synthèse d'un additionneur (graphe d'état)

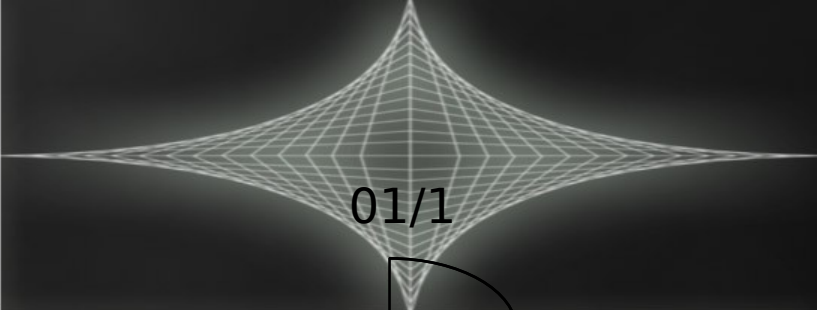
Voici le graphe de transition complet de l'additionneur.

Nous allons vérifier son comportement à partir d'un certains nombres de stimulus d'entrées.

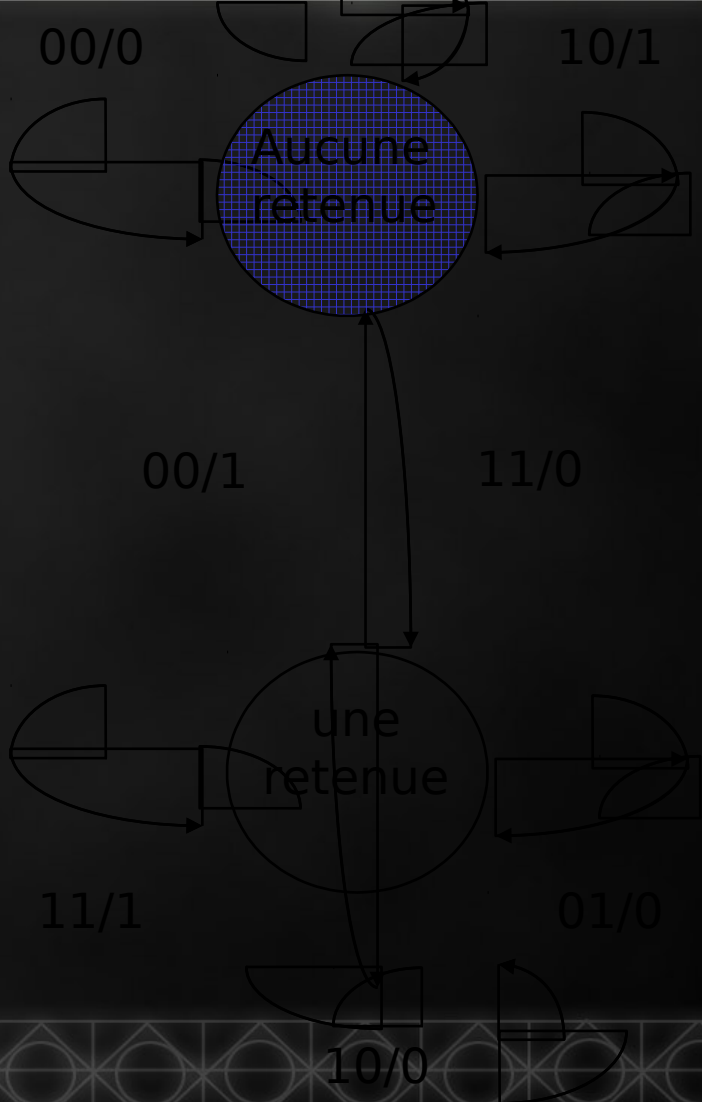
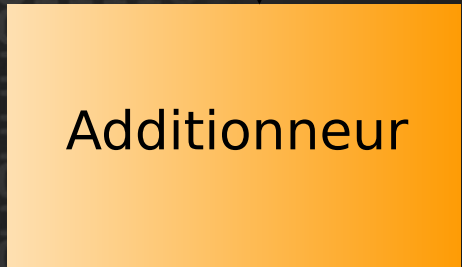


# Synthèse d'un additionneur

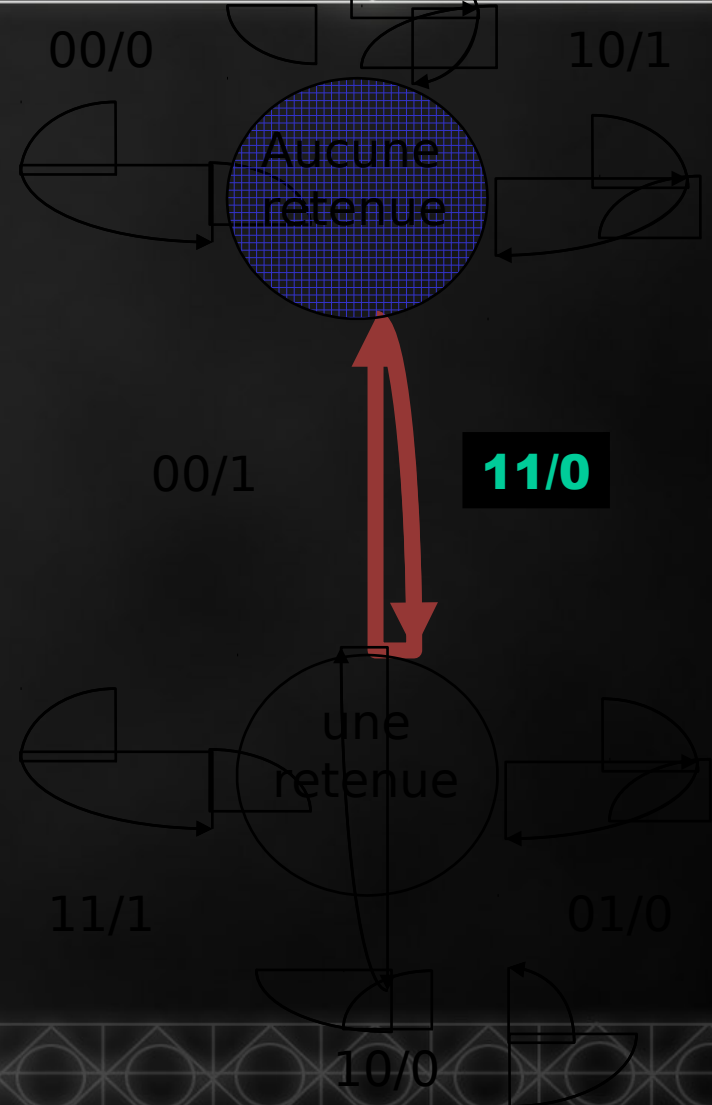
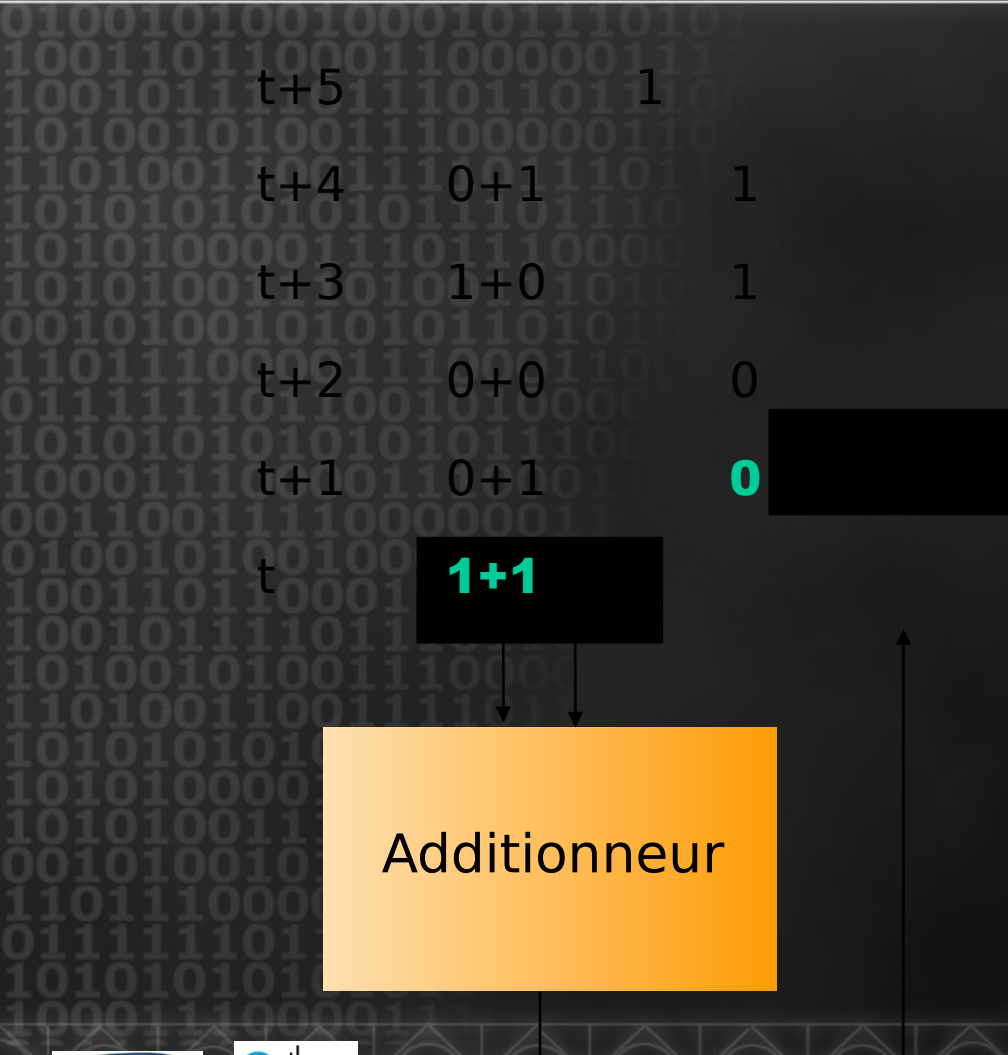
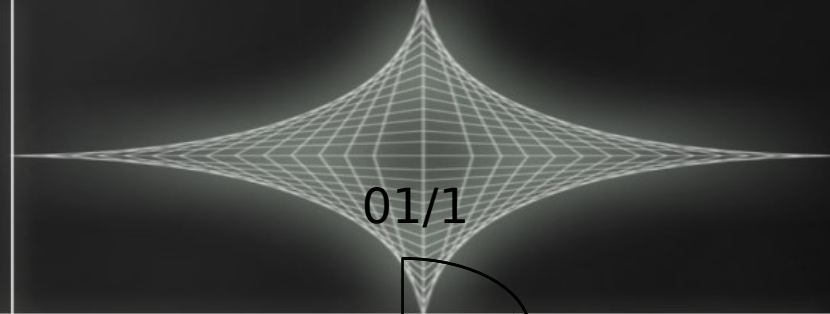
1001 + 10011 = 11100



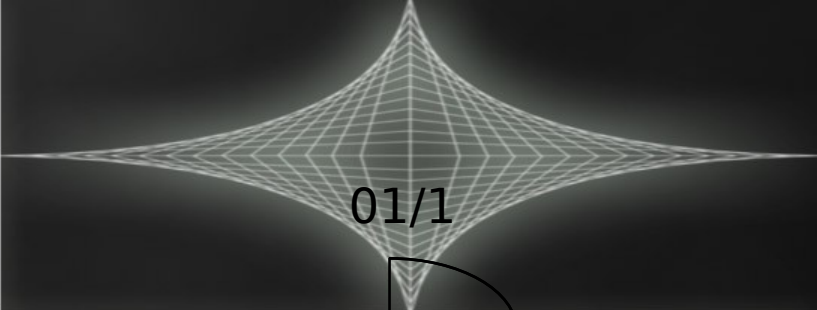
t+5		1
t+4	0+1	1
t+3	1+0	1
t+2	0+0	0
t+1	0+1	0
t	1+1	



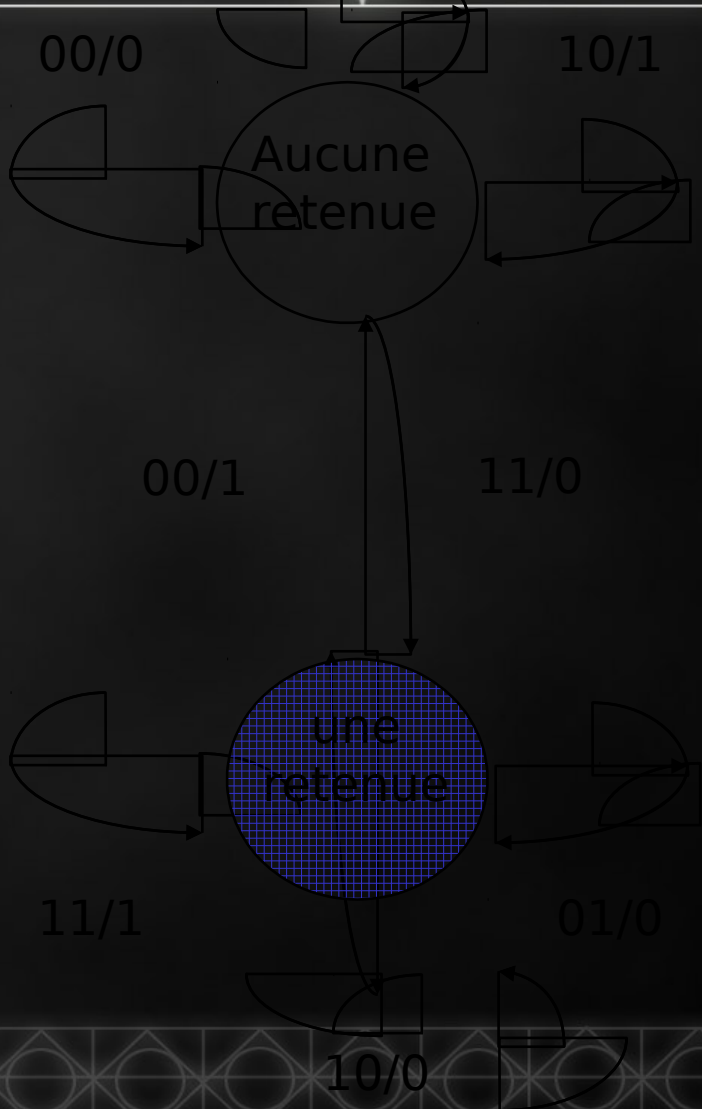
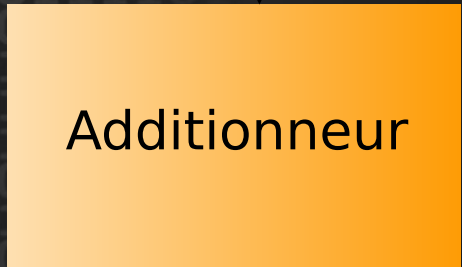
# Synthèse d'un additionneur (graphe d'état)



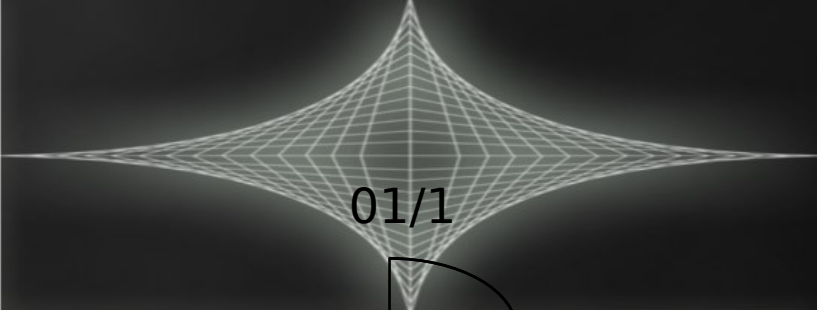
# Synthèse d'un additionneur



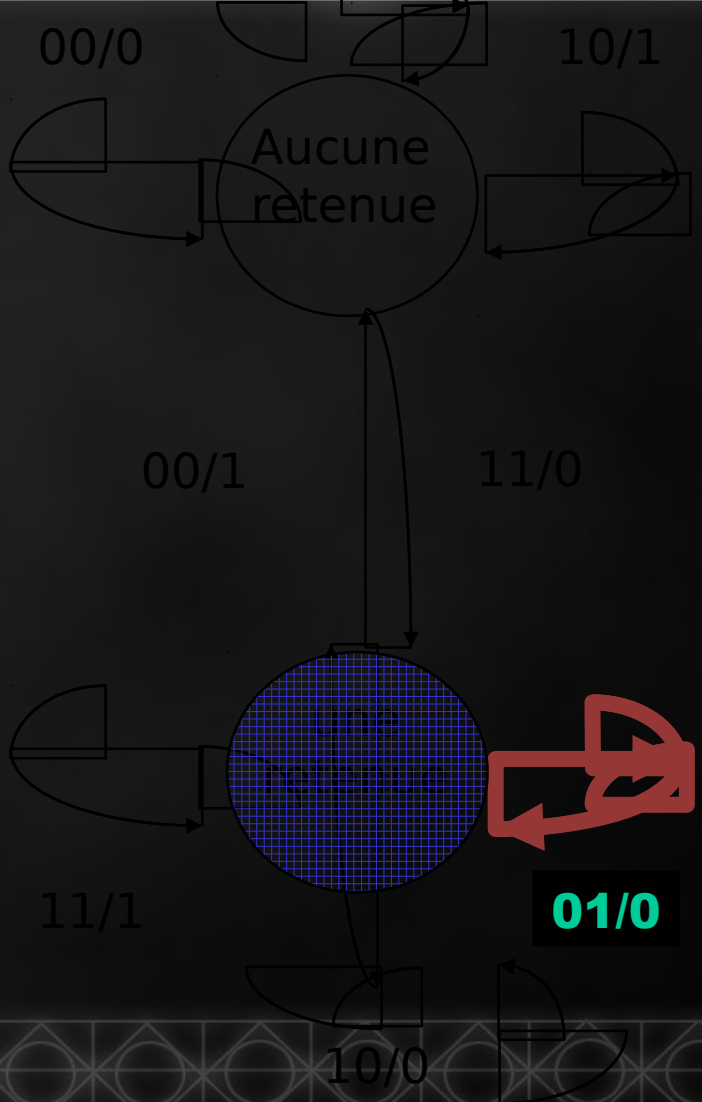
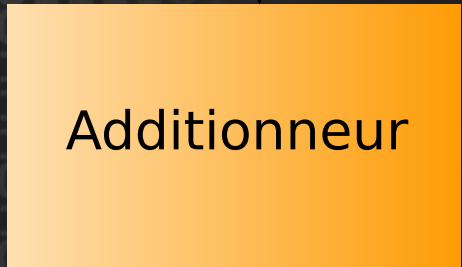
$t+5$       1  
 $t+4$     0+1      1  
 $t+3$     1+0      1  
 $t+2$     0+0      0  
 $t+1$     0+1      0  
 $t$        1+1



# Synthèse d'un additionneur

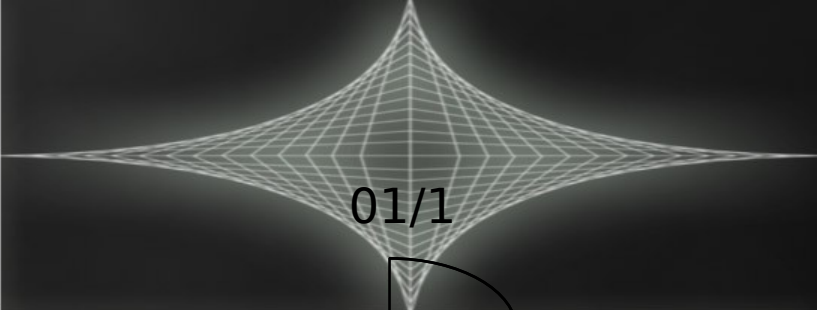


t+5		1
t+4	0+1	1
t+3	1+0	1
t+2	0+0	0
t+1	<b>0+1</b>	0
t	1+1	



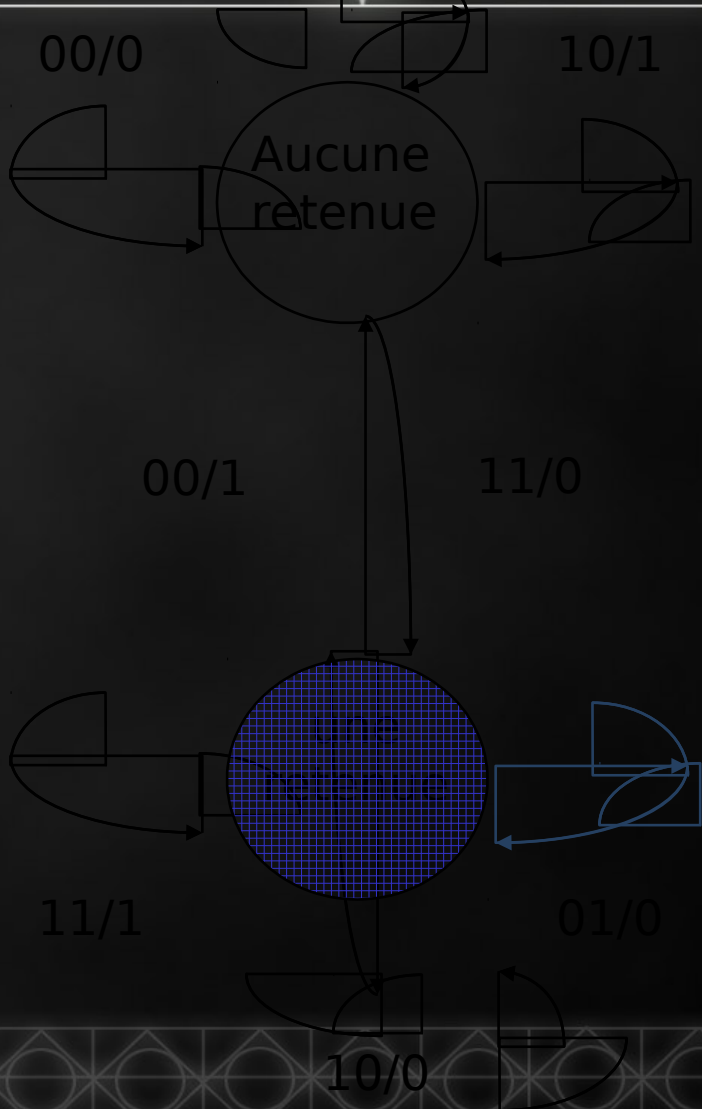
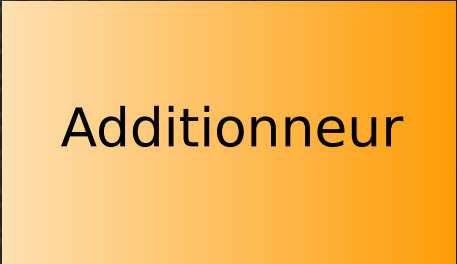


# Synthèse d'un additionneur

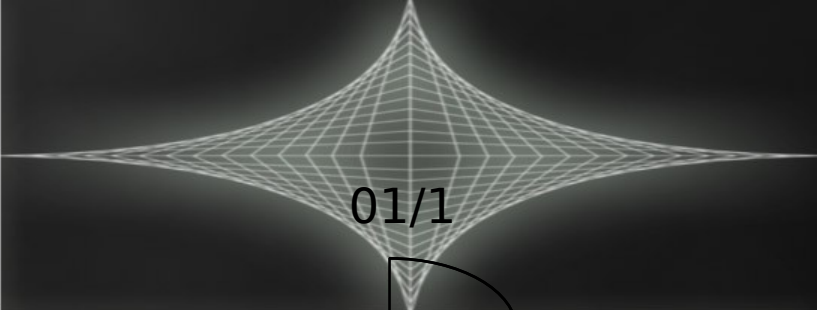


t+5	1	1000000	1
t+4	0+1	1	1
t+3	1+0	1	1
t+2	0+0	0	0
t+1	0+1	0	0
t	1+1		

Two arrows point from the right side of the table to the "Additionneur" block below.



# Synthèse d'un additionneur

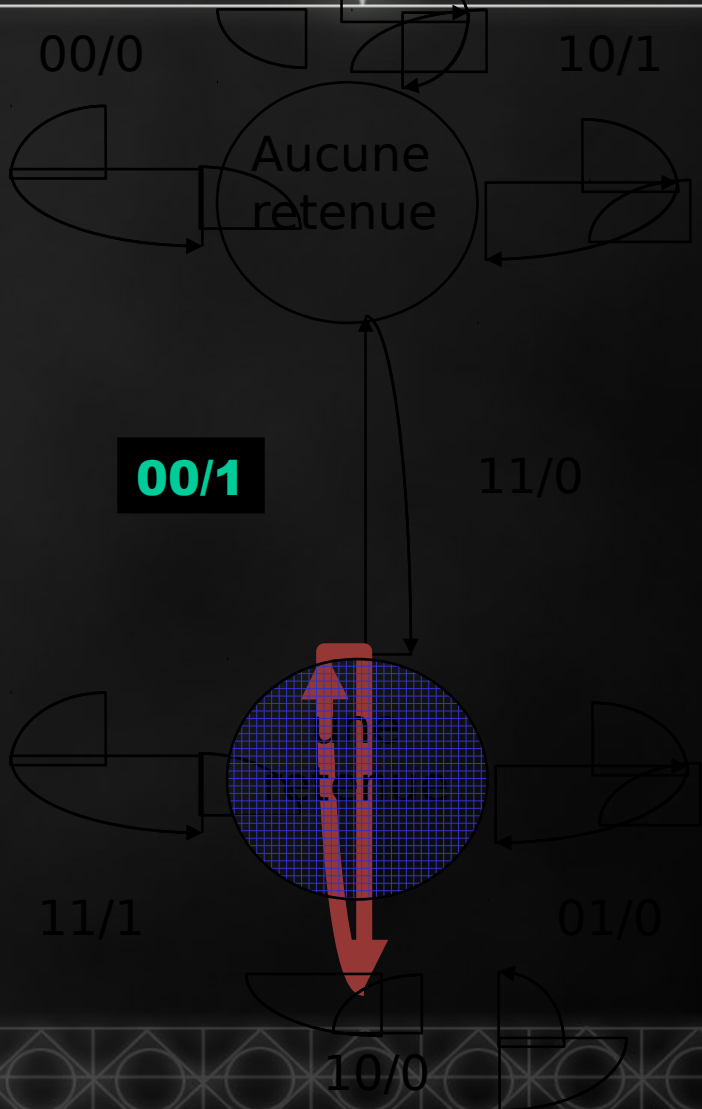
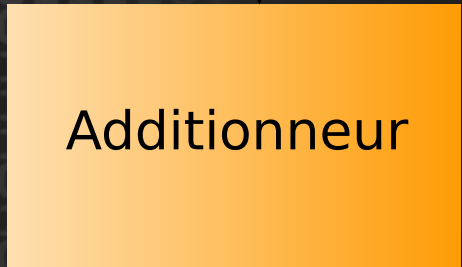


t+5		1	
t+4	0+1	1	
t+3	1+0	1	
t+2	0+0	0	
t+1	0+1	0	
t	1+1		

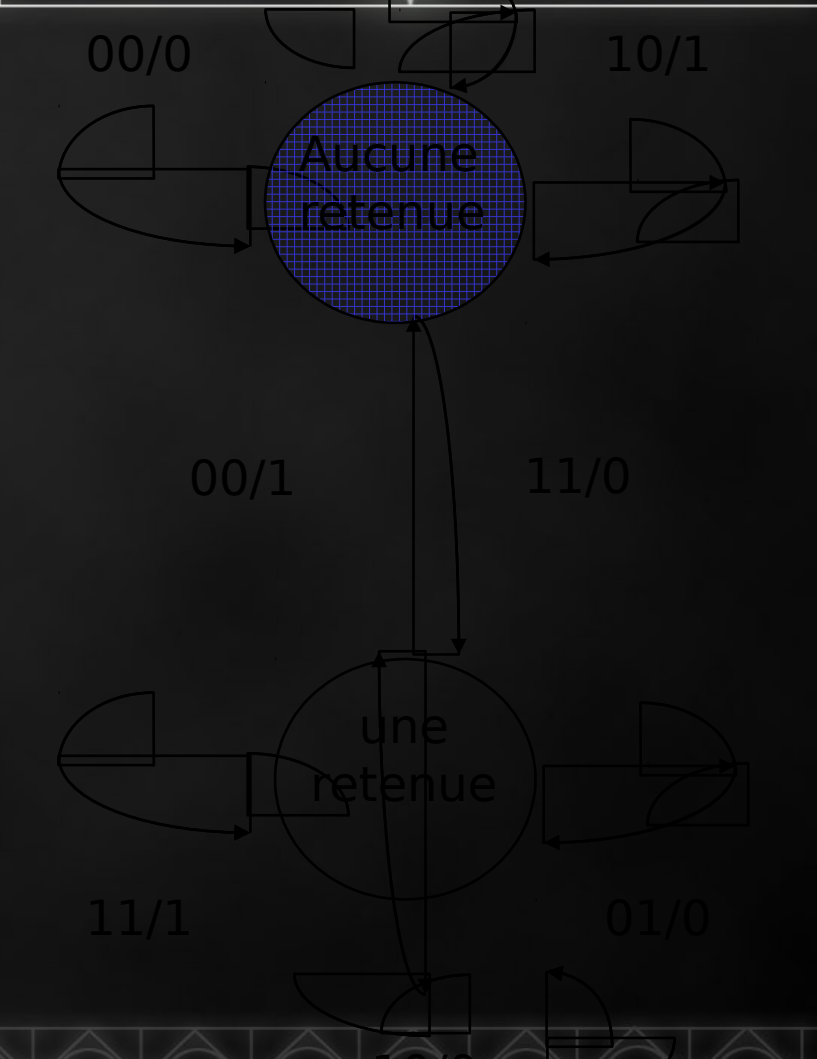
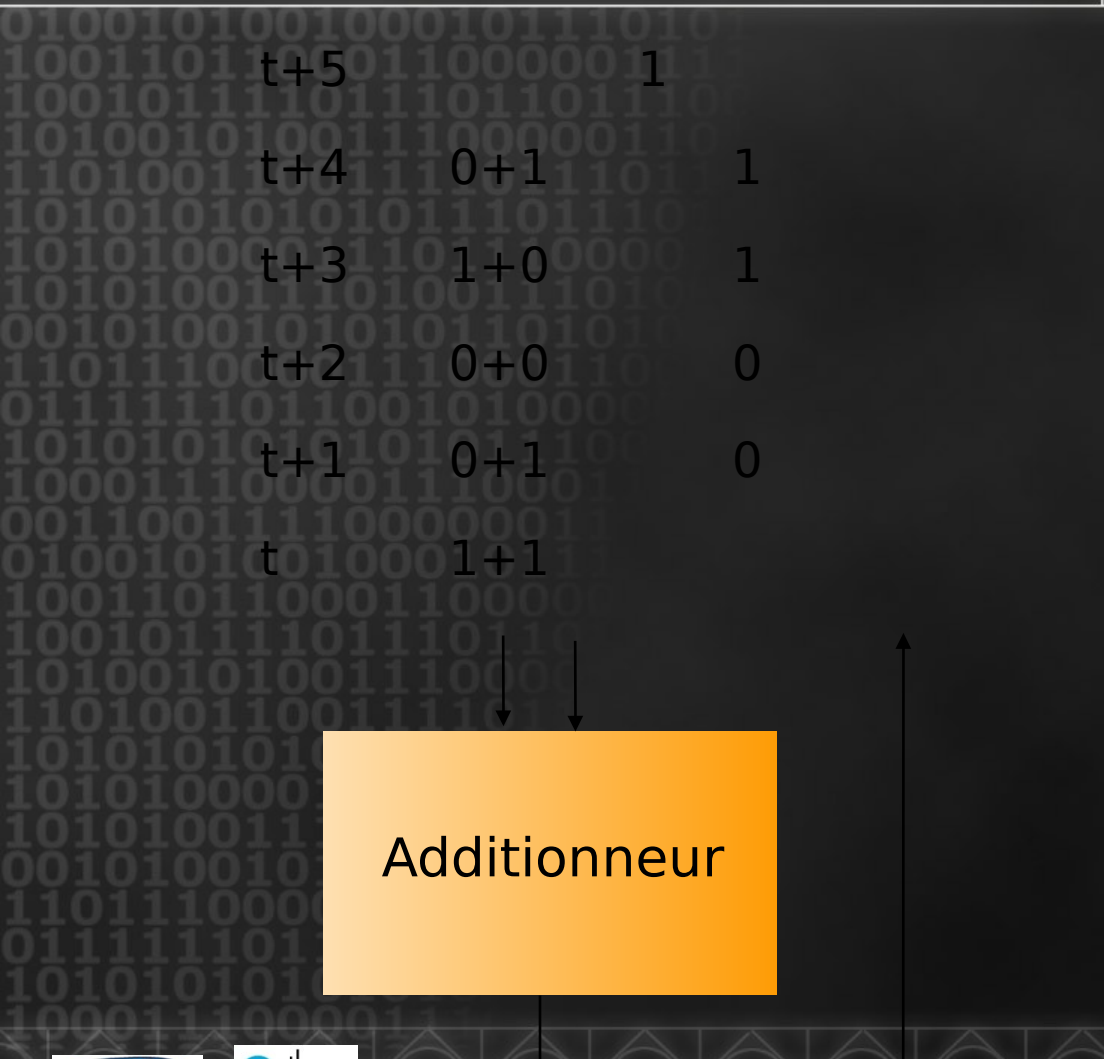
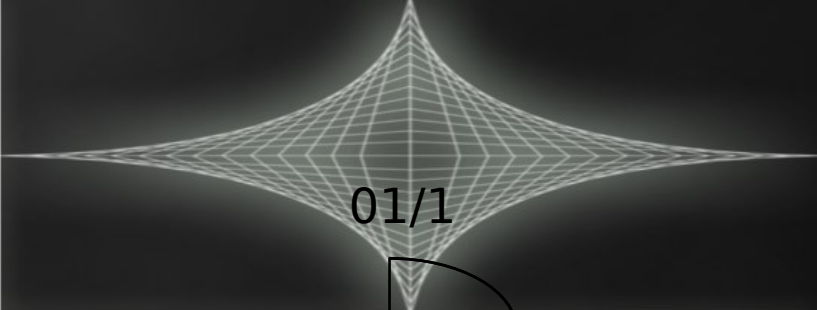
1

0+0

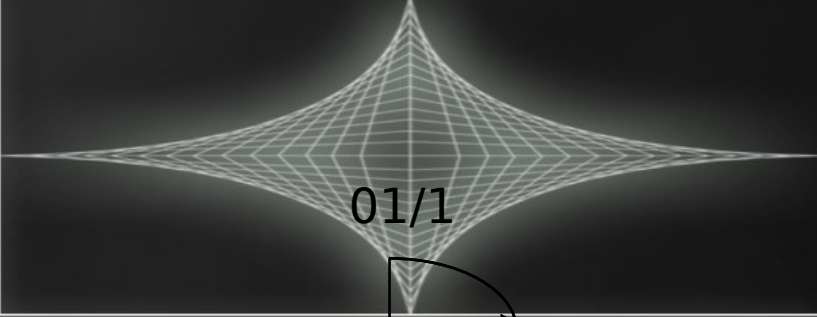
00/1



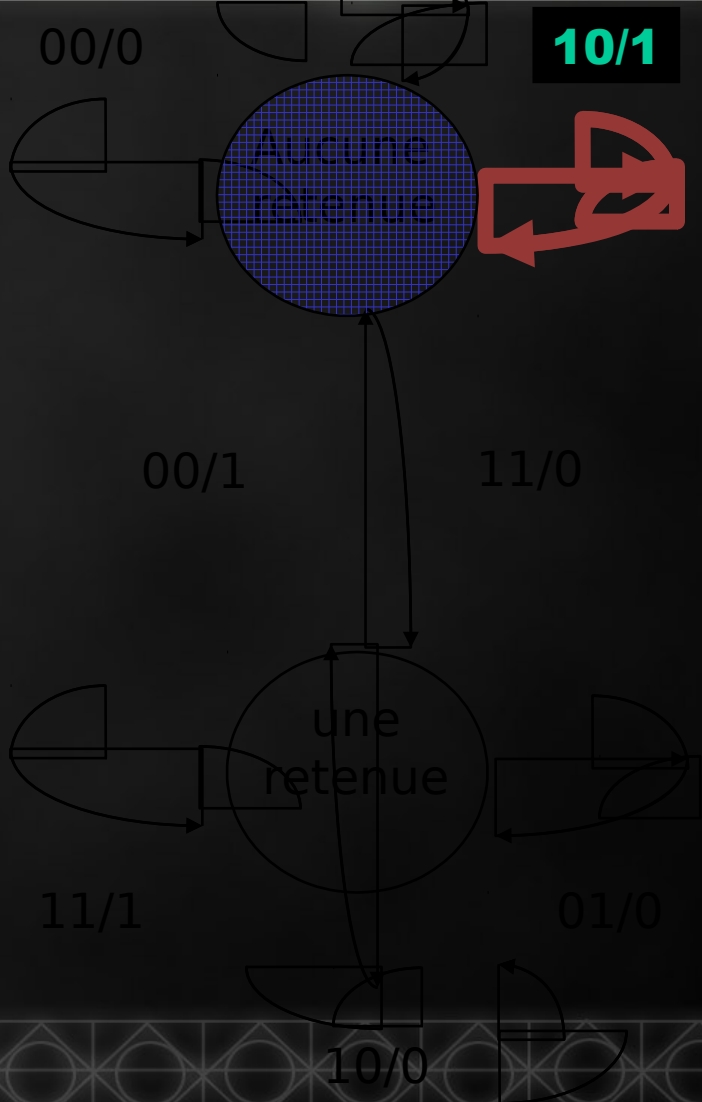
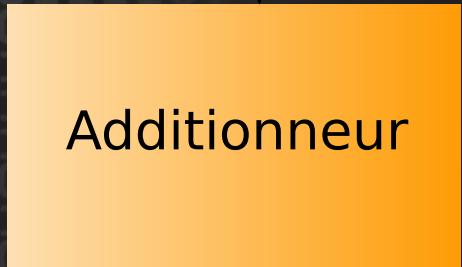
# Synthèse d'un additionneur



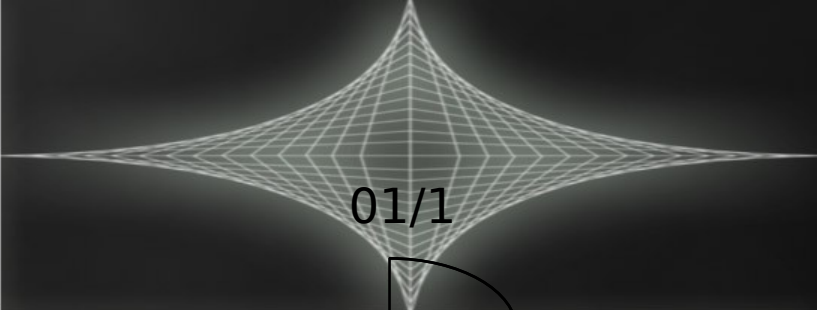
# Synthèse d'un additionneur



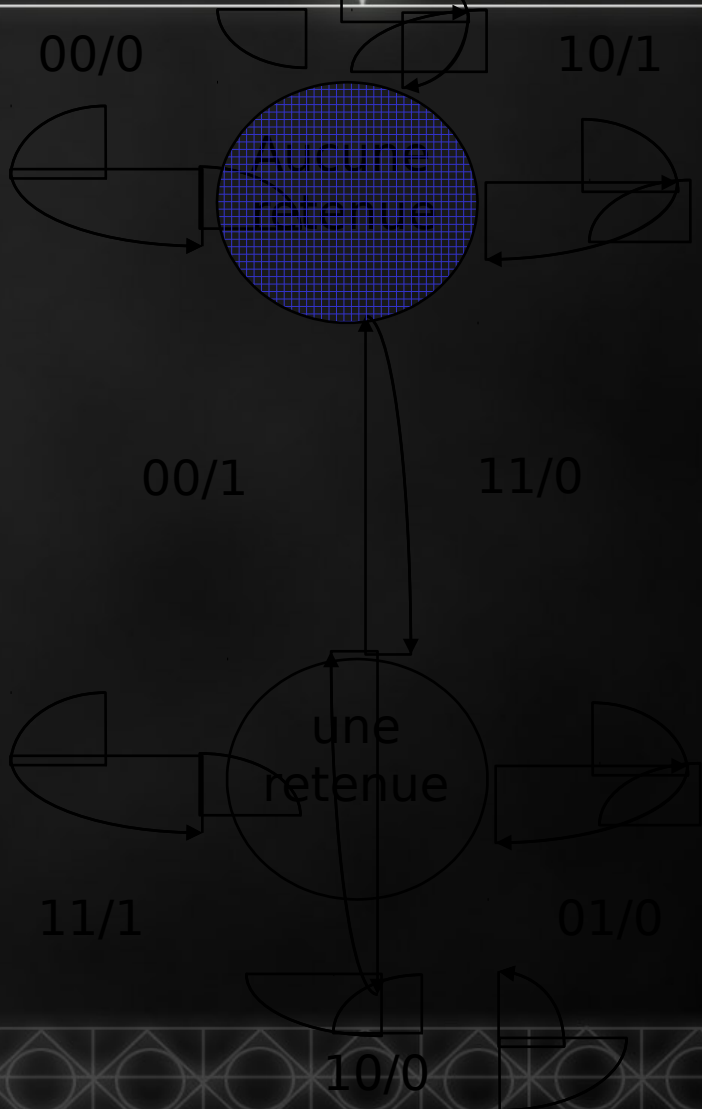
t+5		1
t+4	0+1	1
t+3	1+0	1
t+2	0+0	0
t+1	0+1	0
t	1+1	



# Synthèse d'un additionneur



	$t+5$		1
	$t+4$	$0+1$	1
	$t+3$	$1+0$	1
	$t+2$	$0+0$	0
	$t+1$	$0+1$	0
	$t$	$1+1$	

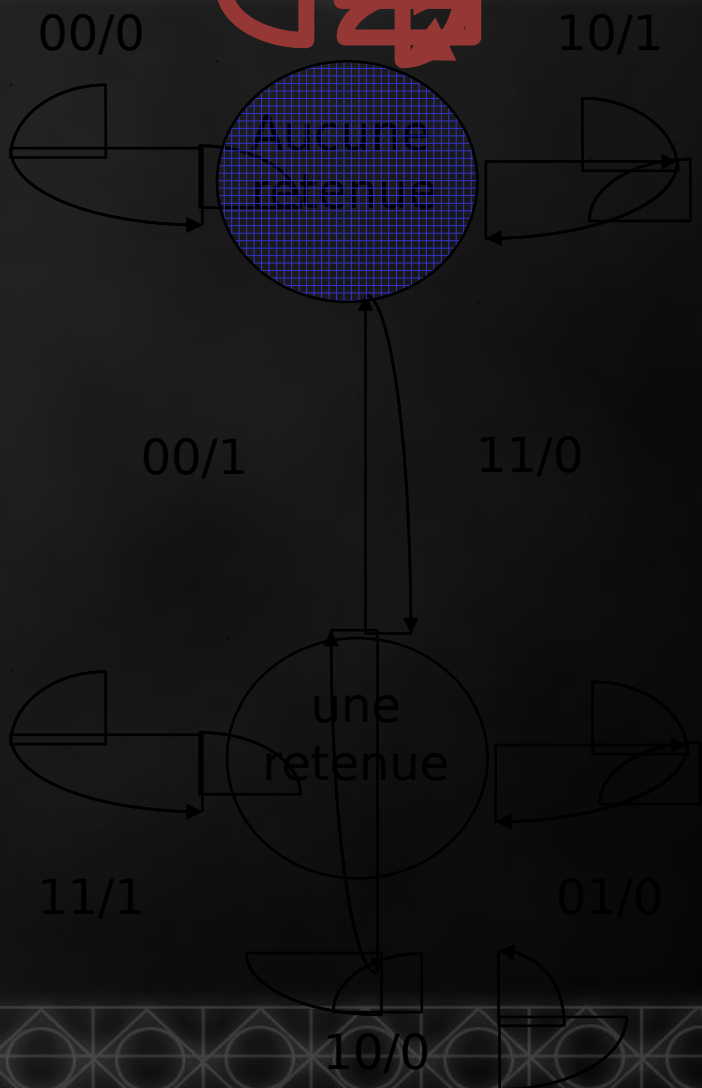
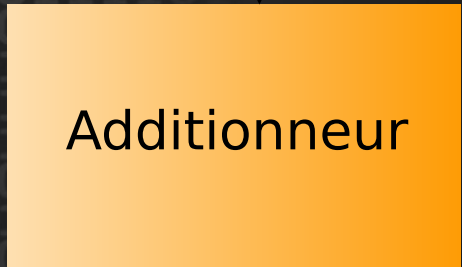


# Synthèse d'un additionneur

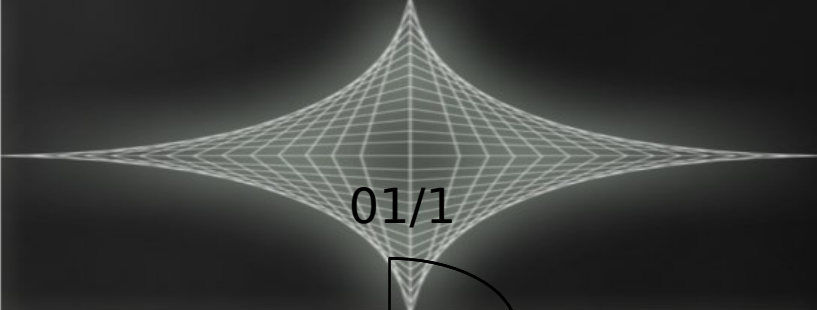


t+5                    1                    [Redacted]

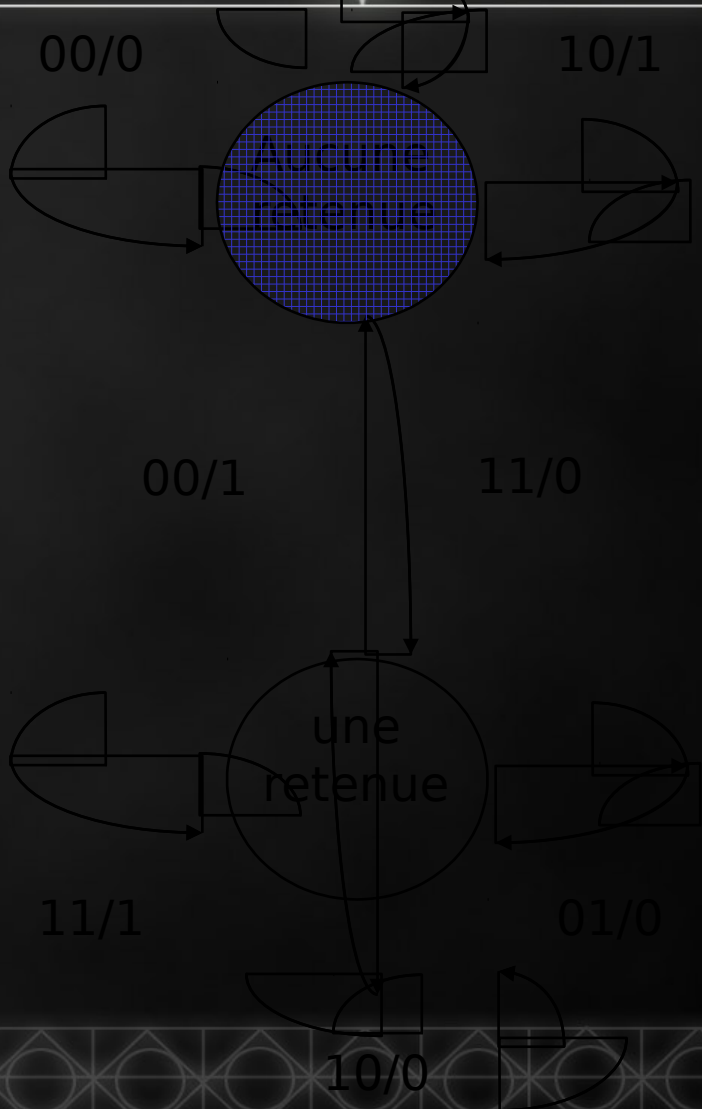
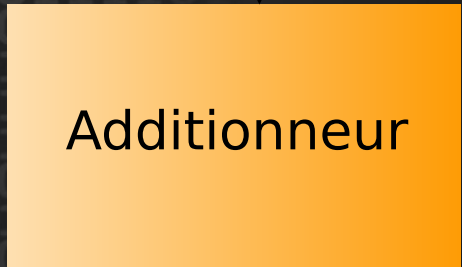
t+4	<b>0+1</b>	1
t+3	1+0	1
t+2	0+0	0
t+1	0+1	0
t	1+1	



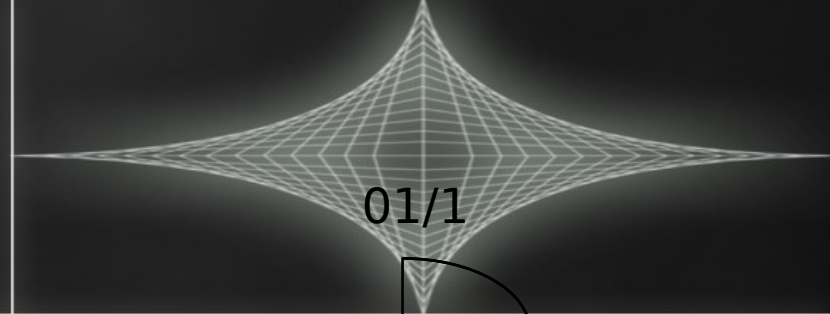
# Synthèse d'un additionneur



t+5		1
t+4	0+1	1
t+3	1+0	1
t+2	0+0	0
t+1	0+1	0
t	1+1	



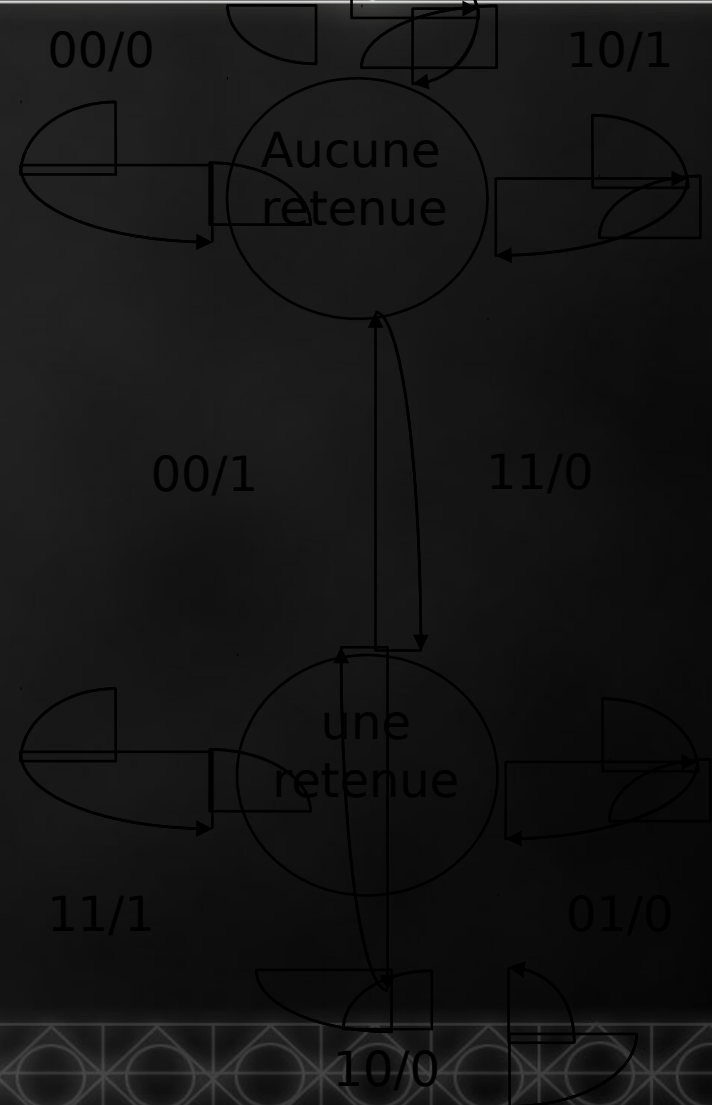
# Synthèse d'un additionneur



10010100100010111010  
10011011000110000011  
10010111101110110110  
10100101001110000010  
11010011001111011101  
10100101010101010101  
100011100001110000  
1010111101110110  
1010010100111000  
101001010101110  
101001001110100  
00101001010101  
11011100001110  
0111110110010  
10101010101010  
10001110000111000

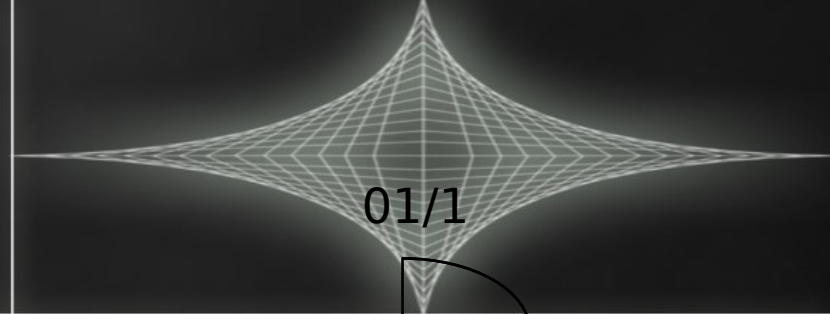
J'ai vérifié sur un jeu d'entrées **non exhaustif** que le graphe semble bien correspondre au comportement attendu de mon système.

La construction du graphe est l'étape la plus délicate.

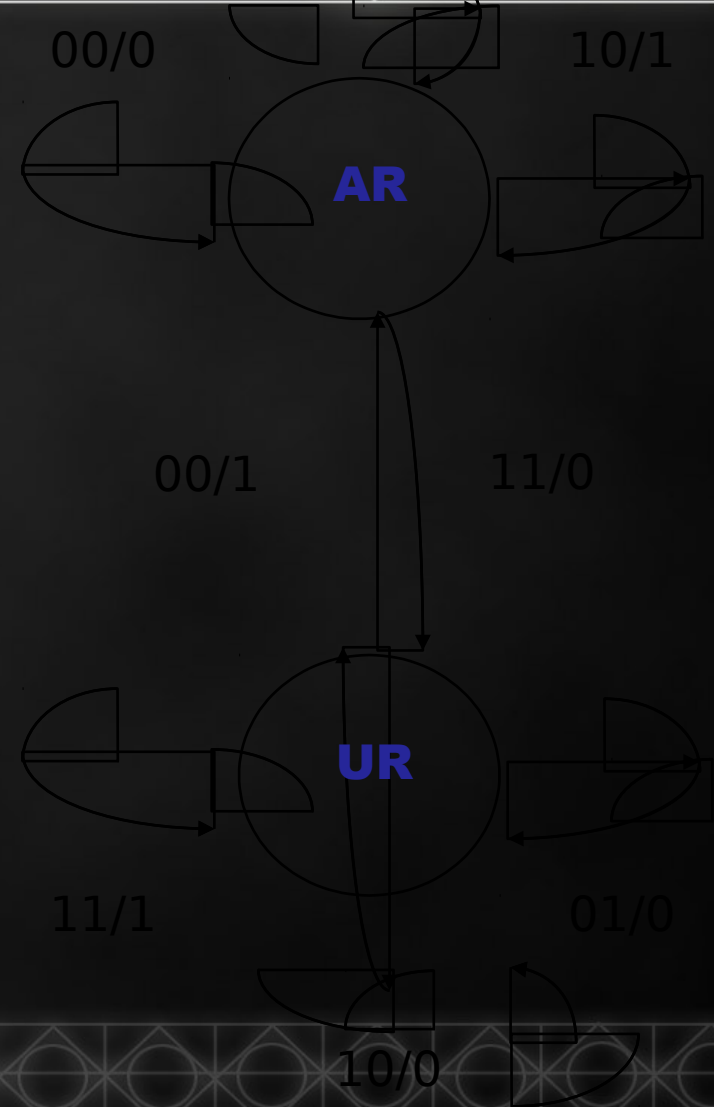




# Représentation sous forme de table



a	b	$E_{\text{Present}}$	S	$E_{\text{Futur}}$
0	0	AR 0	AR	
0	0	UR 1	AR	
0	1	AR 1	AR	
0	1	UR 0	UR	
1	0	AR 1	AR	
1	0	UR 0	UR	
1	1	AR 0	UR	
1	1	UR 1	UR	



# Codage des états

Codage des états = Nombre de bascules

Il y a deux états :

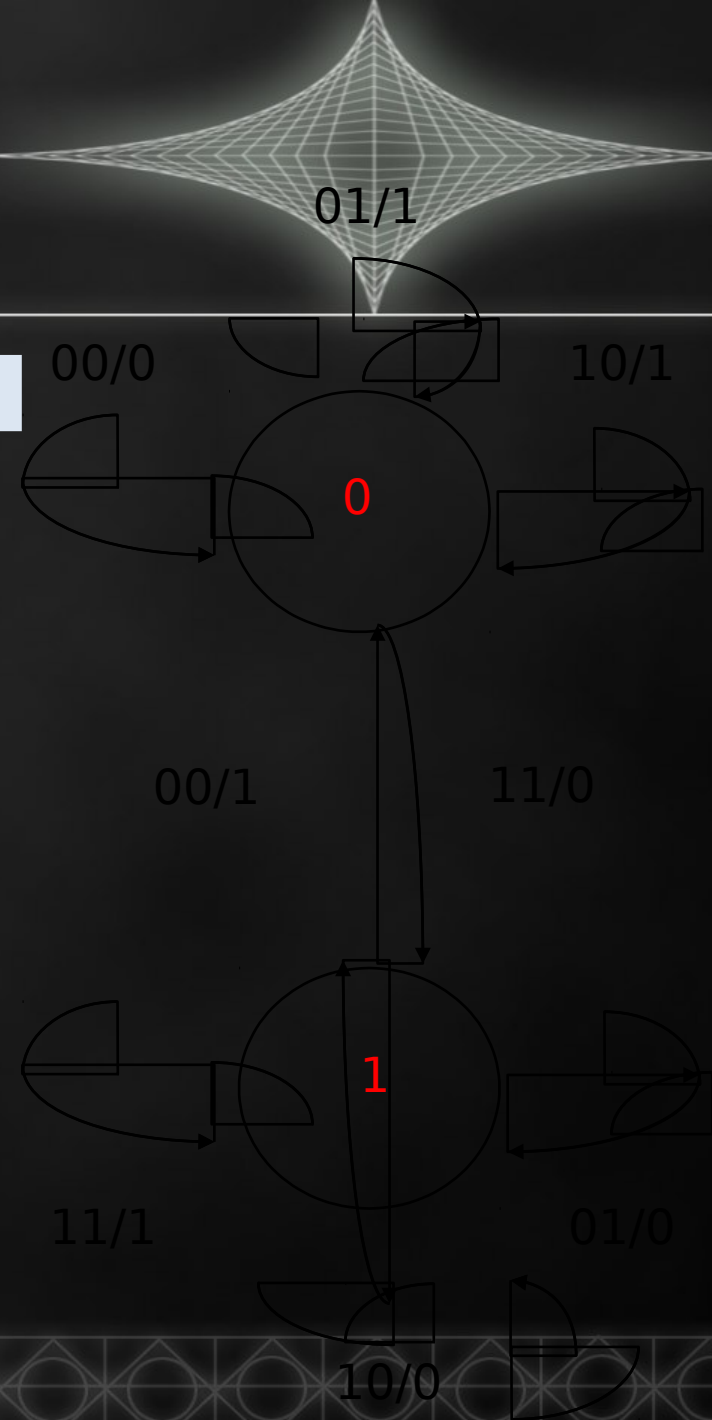
Etat *aucune retenue* est codé **0**

Etat *une retenue* est codé **1**

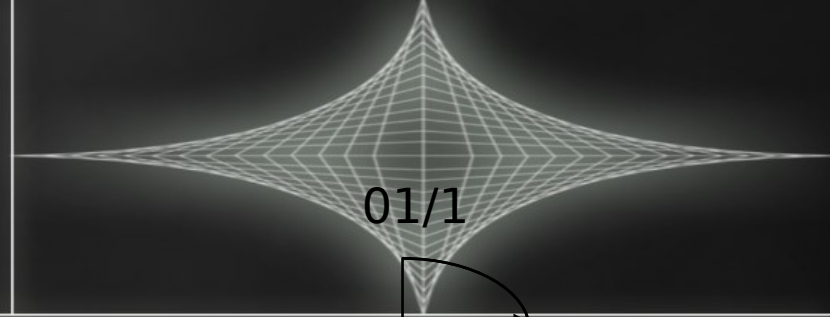
Le nombre de bascules est donnée par :

$$2^{nbB} \geq nb \text{ Etats}$$

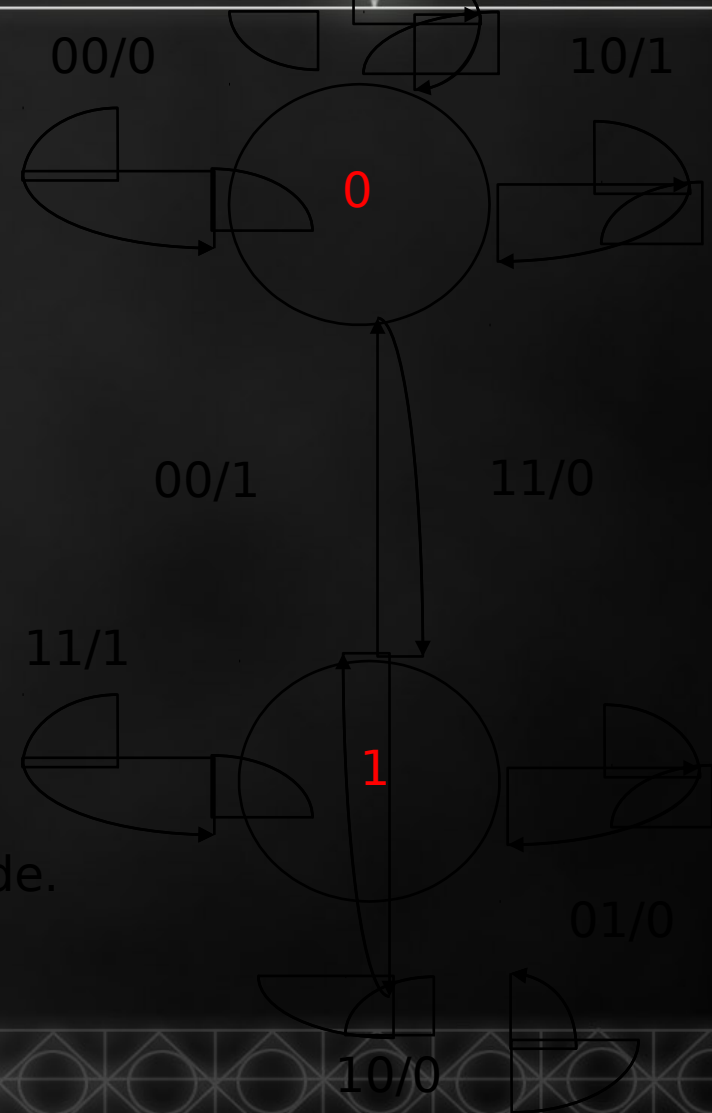
$$nb \text{ B} = 1$$



# La table des états



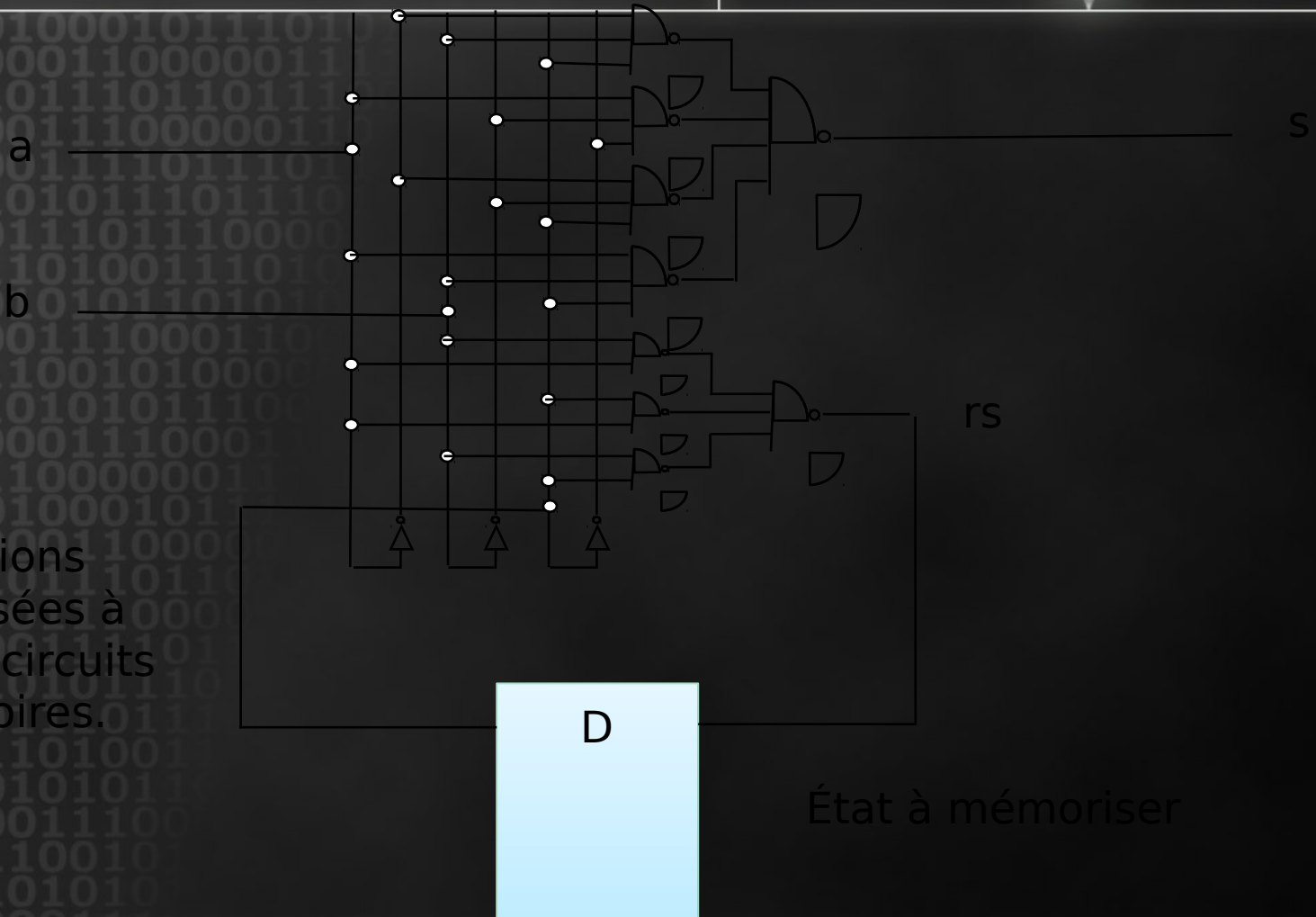
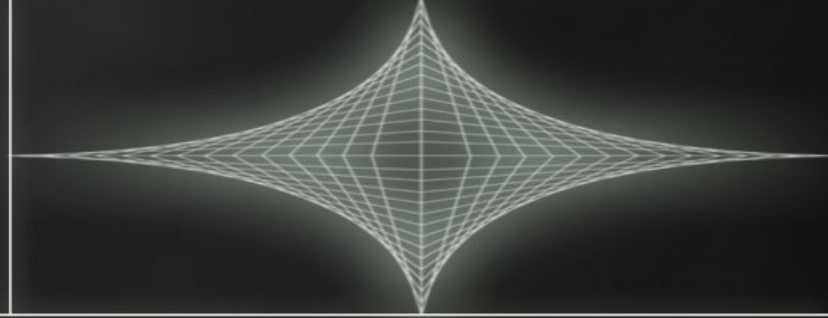
a	b	$E_{\text{Present}}$	S	$E_{\text{Futur}}$
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1



On remplace le nom de l'état par son code.

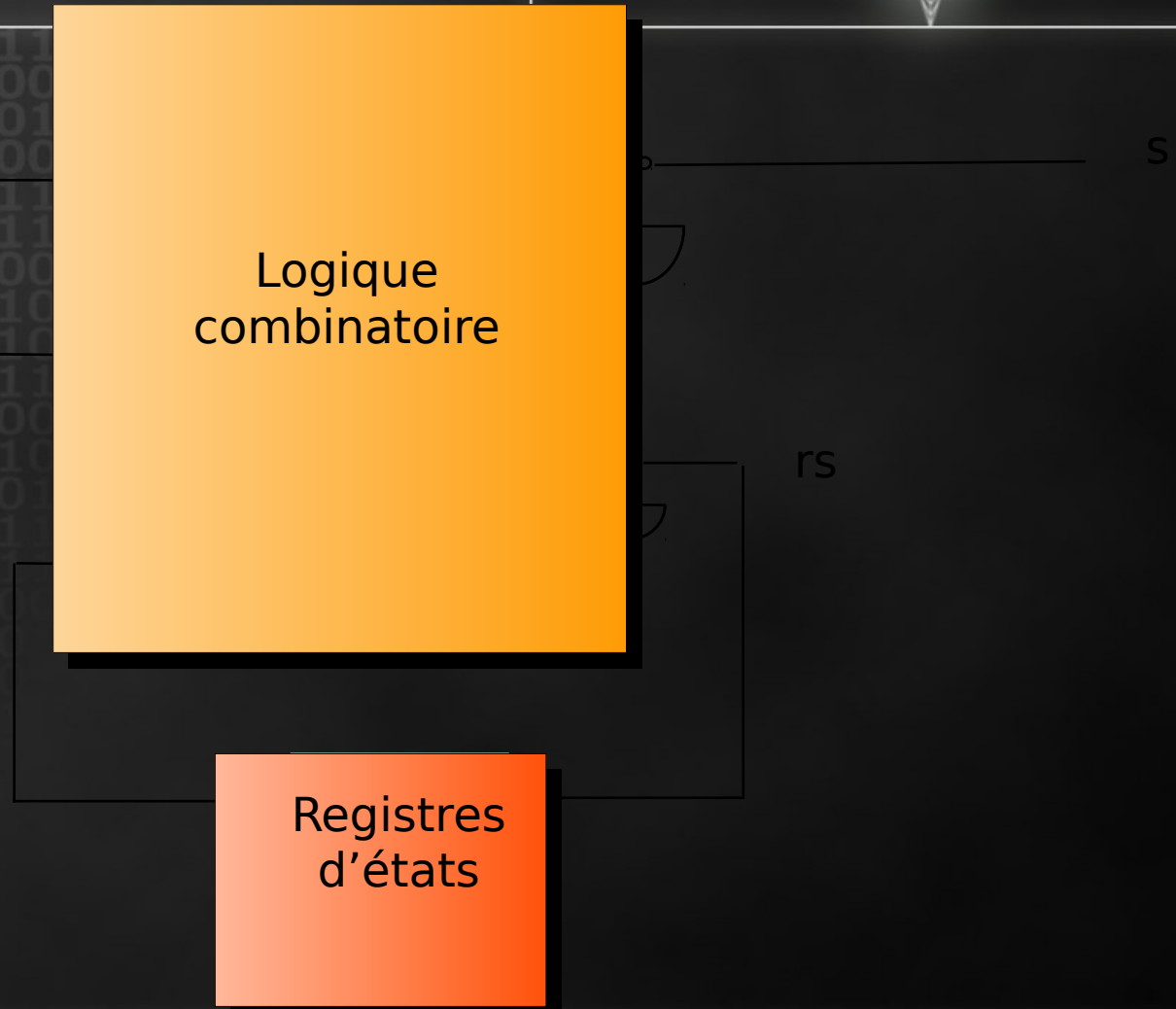
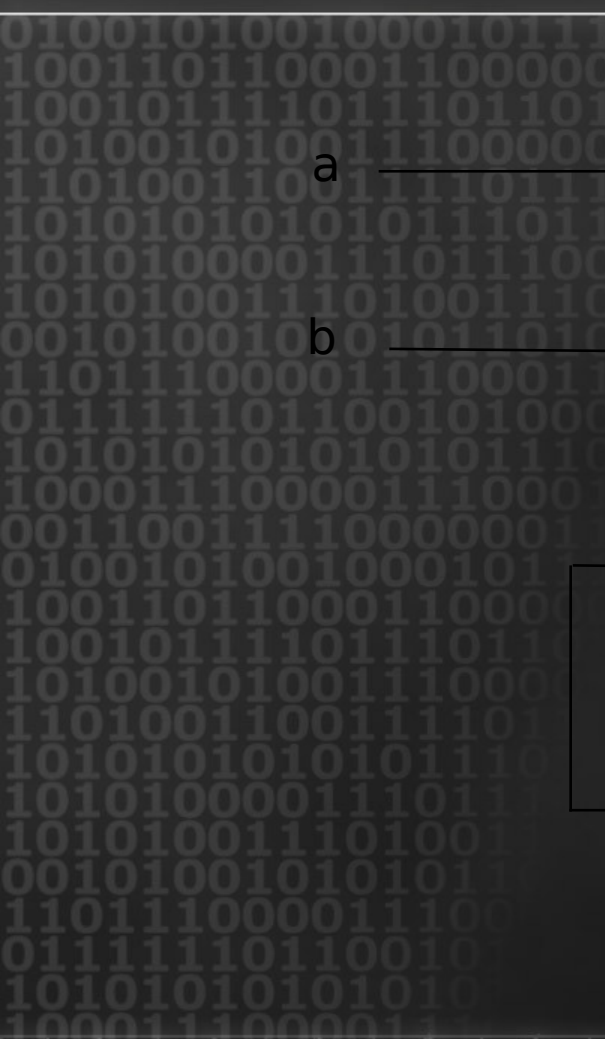
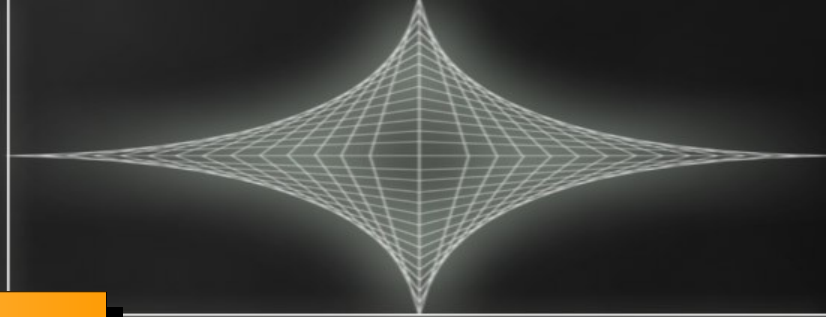
Fonctions logiques pour la sortie et l'état

# Synthèse d'un additionneur (schéma)

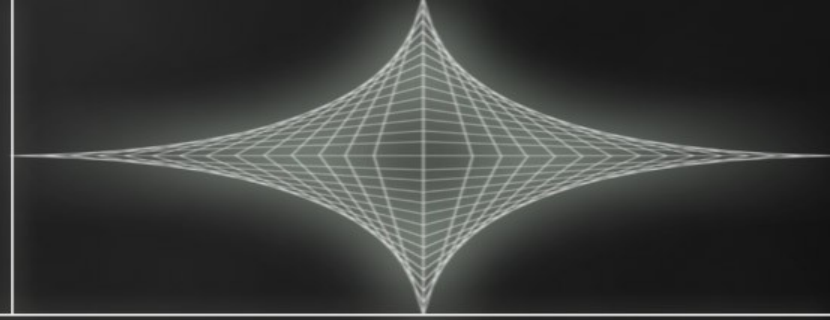


Les équations  
sont réalisées à  
l'aide de circuits  
combinatoires.

# Synthèse d'un additionneur (schéma)

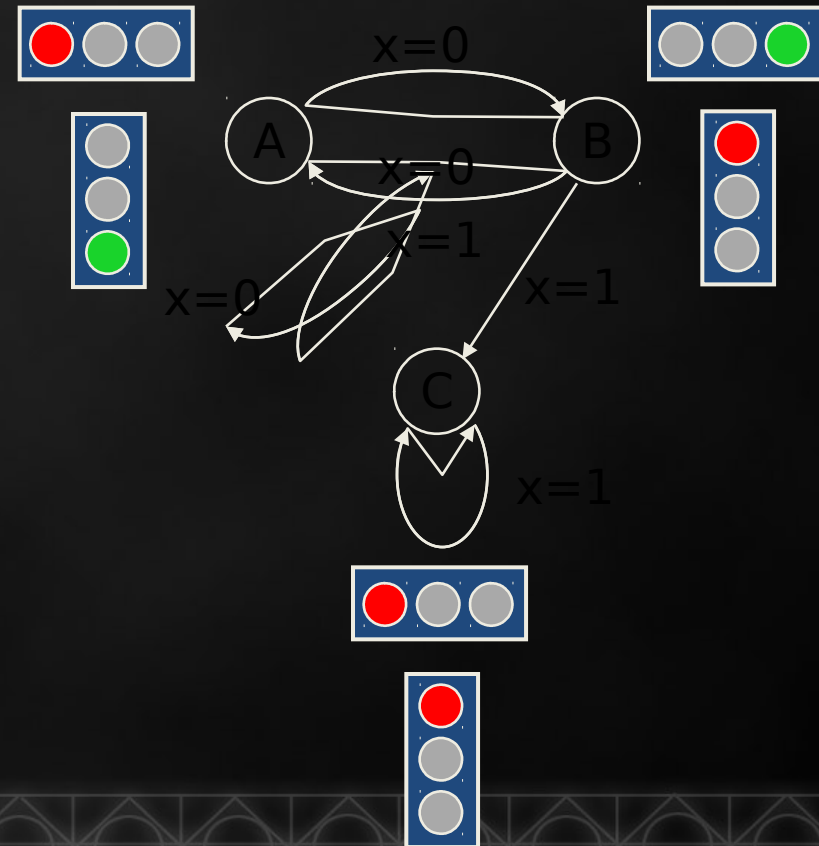


# Système de contrôle de feux de circulation

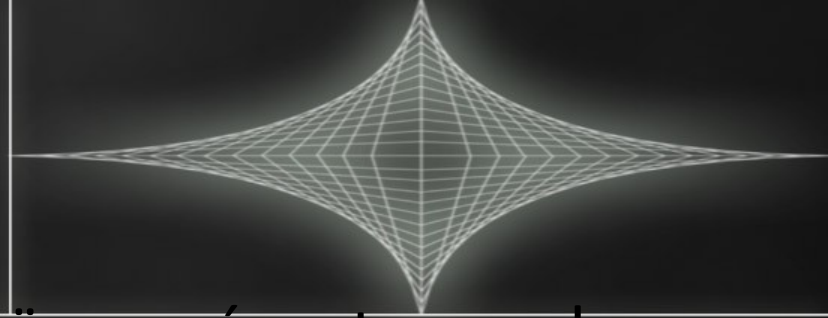


Les feux alternent de A à B à chaque coup d'horloge quand  $x = 0$ .

- Dans l'état A, la circulation se fait dans la direction NS,
  - Dans l'état B, dans la direction EO.
- Un piéton peut traverser après avoir appuyé sur le bouton ( $x = 1$ )
- Quand  $x = 1$ , on passe à l'état C dans lequel les feux sont sous deux rouges pour la durée d'une horloge ou tant que le bouton est enfoncé.

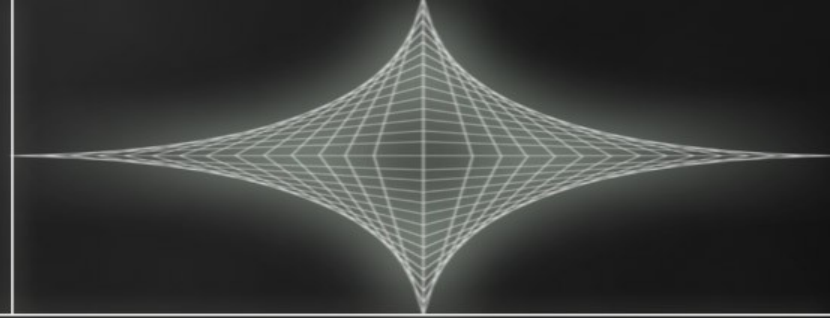


## feux de circulation pas top!

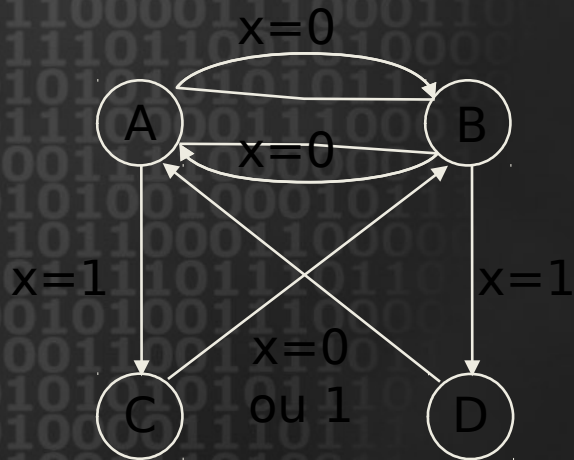


- Cette réalisation un peu naïve présente quelques problèmes :
- Si un malin appuie sans cesse sur le bouton, la circulation automobile est complètement paralysée.
- D'autre part, comme le système une fois dans l'état C retourne toujours dans l'état A, il se pourrait qu'on n'arrive presque jamais dans l'état B s'il y a trop de piétons.
- **Bien définir son automate!!!!**

# Une variante



- Une meilleure réalisation serait la suivante :
- Z1 et Z2 commandes des deux feux

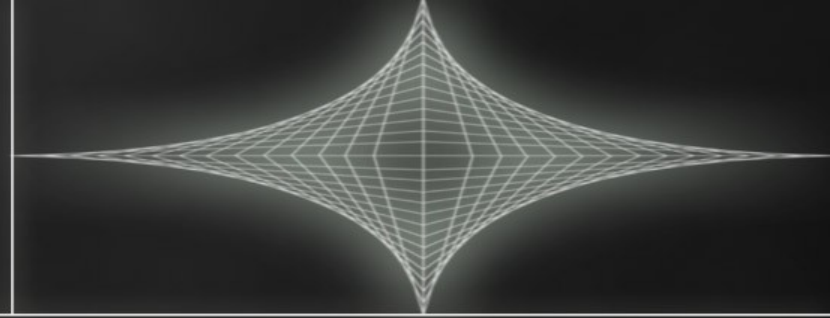


État présent	Entrée présente x		Sortie présente $z_1 z_2$
	0	1	
A	B	C	0 1
B	A	D	1 0
C	B	B	0 0
D	A	A	0 0

État  
suivant



# Codage feux de circulation



➤ Codage des états : on attribue arbitrairement

$Q_1 Q_2 = 00$  représente A

$Q_1 Q_2 = 01$  représente B

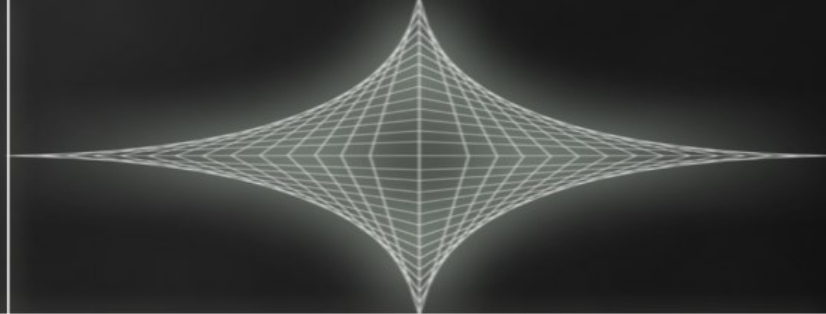
$Q_1 Q_2 = 10$  représente C

$Q_1 Q_2 = 11$  représente D

État présent	Entrée présente x		Sortie présente $z_1 z_2$
	0	1	
00	10	10	0 1
01	00	11	1 0
10	01	01	0 0
11	00	00	0 0

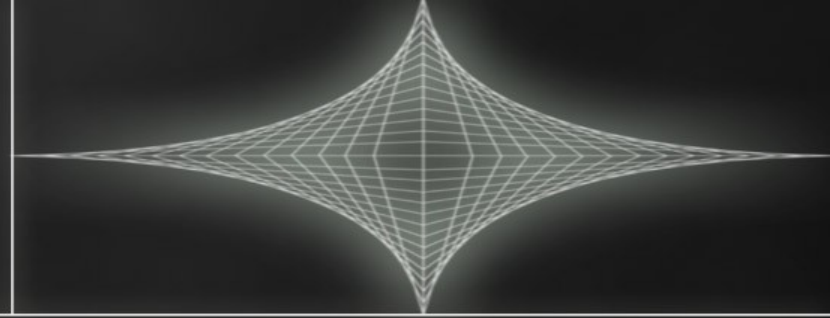
État  
suivant

# Table de transition



Entrée x	État présent $Q_1 Q_2$	État suivant $Q_1^+ Q_2^+$	Sorties Bistables présentes $z_1 z_2$	$D_1 D_2$
0	00	01	01	
0	01	00	10	
0	10	01	00	
0	11	00	00	
1	00	10	01	
1	01	11	10	
1	10	01	00	
1	11	00	00	

# Simplification



➤ Tables de Karnaugh pour les entrées des bistables :

	$Q_1Q_2$			
$x$	00	01	11	10
0	0	0	0	0
1	1	1	0	0

$D_1$

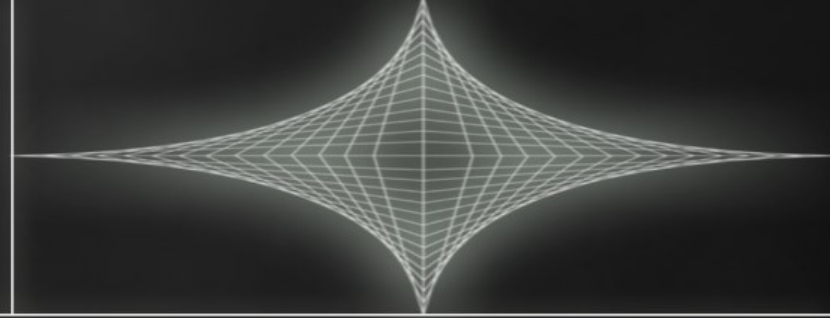
$$D_1 = x \cdot \overline{Q_1}$$

	$Q_1Q_2$			
$x$	00	01	11	10
0	1	0	0	1
1	0	1	0	1

$D_2$

$$D_2 = Q_1 \cdot \overline{Q_2} + \overline{x} \cdot \overline{Q_2} + x \cdot \overline{Q_1} \cdot Q_2$$

# Simplification



➤ Tables de Karnaugh pour les sorties :

	$Q_1Q_2$			
$x$	00	01	11	10
0	0	1	0	0
1	0	1	0	0

$z_1$

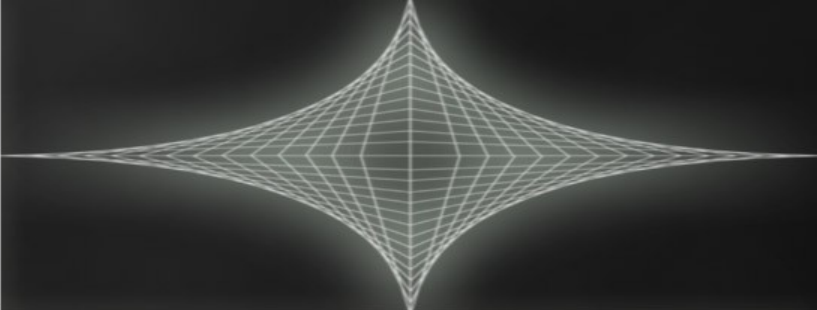
$$z_1 = \overline{Q_1} \cdot Q_2$$

	$Q_1Q_2$			
$x$	00	01	11	10
0	1	0	0	0
1	1	0	0	0

$z_2$

$$z_2 = \overline{Q_1} \cdot \overline{Q_2}$$

# Circuit contrôle de feux



## ➤ Circuit :

