
3) Logique séquentielle

1. Définitions.....	2
1.1. Présentation	2
1.2. Chronogramme	2
1.3. Niveau logique vs front.....	2
2. Les bascules RS (ou SR).....	2
3. Temporisateur.....	3
4. Monostables	3
4.1. Monostable non redéclenchable.....	3
4.2. Monostable redéclenchable	4
5. Chronogrammes et équations	4
6. Opérations numériques sur des mots	4
6.1. Définition	4
6.2. Décalage	4
6.3. Rotation	5
6.4. Opérations logiques sur des mots.....	5
7. Le compteur	5
8. CodeSys et la logique séquentielle.....	6
8.1. Généralités	6
8.2. Fronts	6
8.3. Bascules RS et SR.....	6
8.4. Temporisations	6
8.5. Compteurs	7
8.6. Générateur de créneaux.....	7
8.7. Blocs de comparaison.....	7

1. Définitions

1.1. Présentation

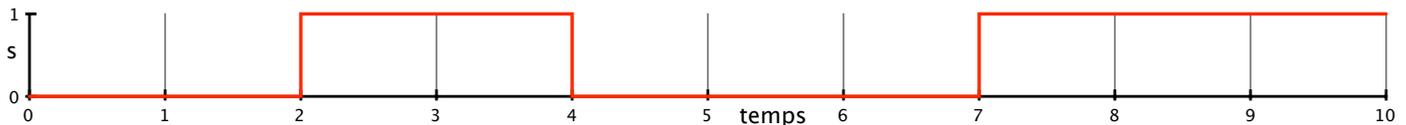
La logique séquentielle, contrairement à la logique combinatoire fait intervenir le temps. La valeur d'une variable logique à l'instant $t+dt$ dépend de celles qu'avaient les variables logiques à l'instant t .

Exemple : $S(t+dt)=a(t)+b(t).S(t)$ que l'on notera : $S = a + b.S$

Remarque : On voit alors que S peut dépendre de $S...$

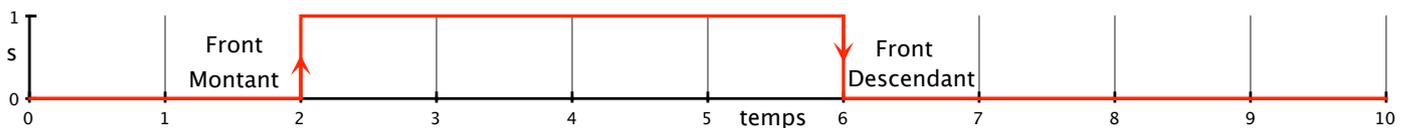
1.2. Chronogramme

Le chronogramme est une représentation graphique de l'évolution temporelle d'un signal électrique ou logique. On trouve sur l'axe des abscisses le temps, sur l'axe des ordonnées l'état (0 ou 1) des variables étudiées.



1.3. Niveau logique vs front

Une variable logique s peut avoir deux niveaux ; le niveau logique haut (vrai) et le niveau logique bas (faux). Quand elle passe du niveau bas vers le niveau haut, elle définit le front montant. Dans le cas contraire, elle définit le front descendant.



Remarque : Le front montant $\uparrow a$ correspond au front descendant de $\downarrow \bar{a}$ et vis-versa.

$$\uparrow a = \downarrow \bar{a} \text{ et } \downarrow a = \uparrow \bar{a}$$

2. Les bascules RS (ou SR)

Une bascule est un circuit logique doté d'une sortie et d'une ou plusieurs entrées. Les changements d'état de la sortie sont déterminés par les signaux appliqués aux entrées. Ce qui différencie les bascules des circuits logiques combinatoires, c'est que la sortie maintient son état même après disparition du signal de commande, la bascule a une 'mémoire'. La bascule est l'élément de base de la logique séquentielle. En effet, en assemblant des bascules, on peut réaliser des compteurs, des registres, des registres à décalage, des mémoires.

La bascule RS à deux entrées, une pour mettre sa sortie Q à 1 ; S (set) et une pour mettre sa sortie Q à 0 ; R (reset).

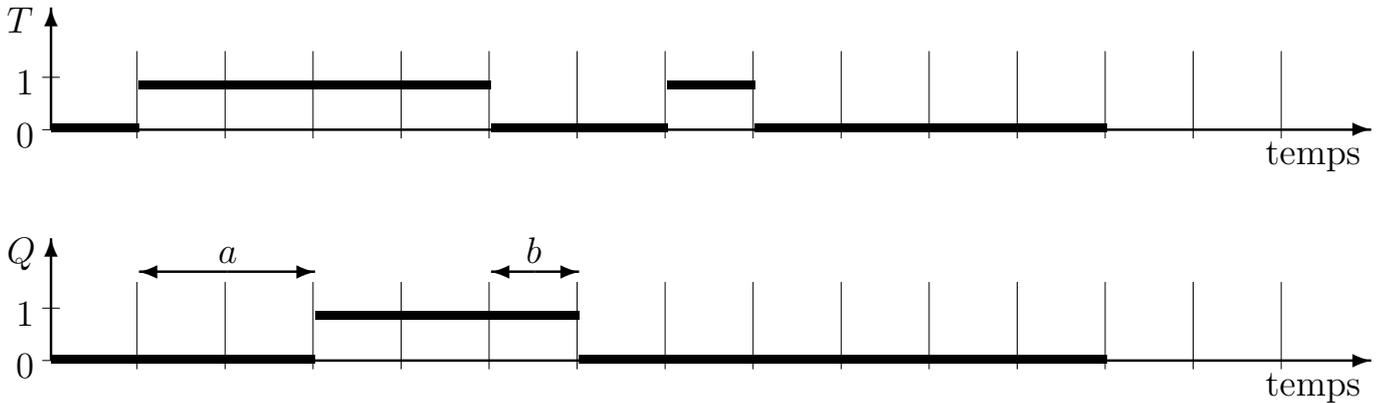
Schématisation	Table de Karnaugh			Equation de sortie
	SR/Q	0	1	Set prioritaire : $Q = S + Q.\bar{R}$
	00	0	1	
	01	0	0	Reset prioritaire : $Q = (S + Q).\bar{R}$
	11	Selon priorité		
	10	1	1	

3. Temporisateur

Le temporisateur retarde le front montant d'un signal T d'un temps a et le front descendant du même signal d'un temps b.

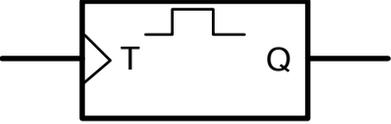
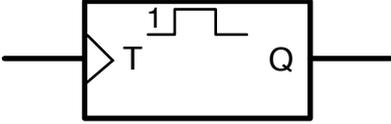
Schématisation	Equation de sortie	Remarque
	$Q = a/T/b$	$a/T/b = \overline{b/\overline{T}/a}$

Le chronogramme sera alors le suivant :



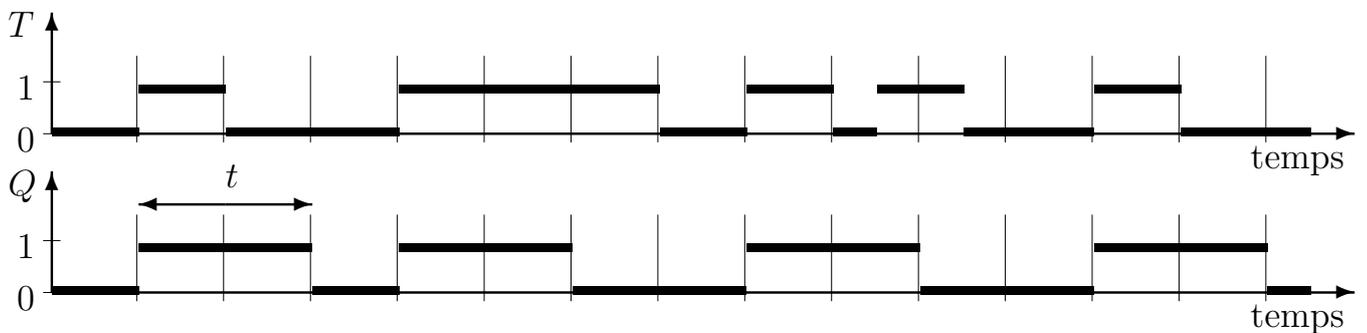
4. Monostables

Comme son nom l'indique ce composant ne connaît qu'un seul état stable. C'est l'état pour lequel sa sortie Q est à 0. Un front positif sur son entrée T provoque l'état instable qui dure un temps déterminé t. Il existe des monostables redéclençables et des non redéclençables.

Redéclençable	Equation	Non redéclençable	Equation
	$Q = \uparrow T/t$		$Q = (Q + T) \cdot \overline{t/Q}$ $\uparrow Q = T \text{ et } \downarrow Q = t/Q$

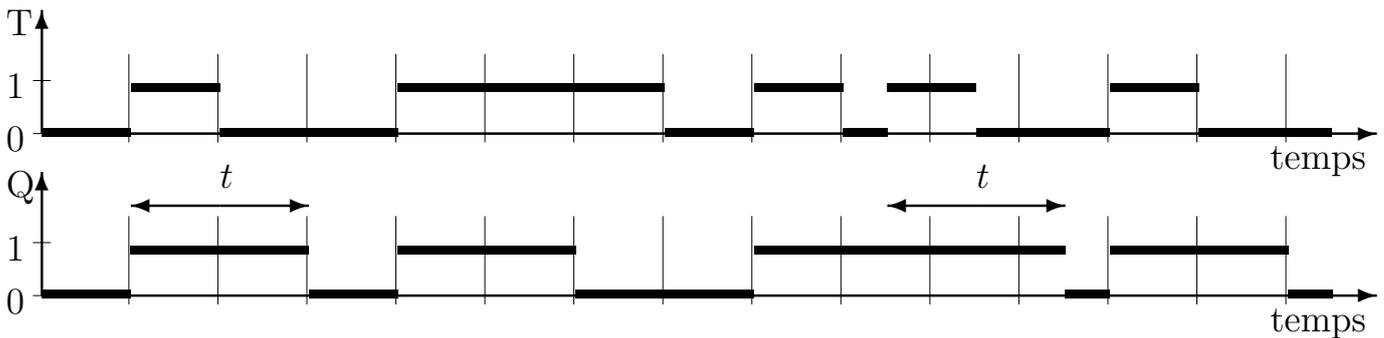
4.1. Monostable non redéclençable

Le front de T déclenche le monostable et la sortie Q passe à 1 pendant une durée t même si l'impulsion est plus longue. Les fronts de T n'ont aucune influence quand Q est à 1. Le monostable doit revenir dans son état stable pour être redéclenché.



4.2. Monostable redéclenchable

Le front de T déclenche le monostable quel que soit la valeur de la sortie Q qui passe ou reste à 1 pendant une durée t même si l'impulsion est plus longue.



5. Chronogrammes et équations

Retard travail	Retard repos	Monostable	Front retardée
$Q=t/l$	$Q=l/t$	$Q=\uparrow I/t$	$Q=\downarrow(\uparrow I/t)$
Temporisation		Monostable redéclenchable	Monostable + Front descendant

6. Opérations numériques sur des mots

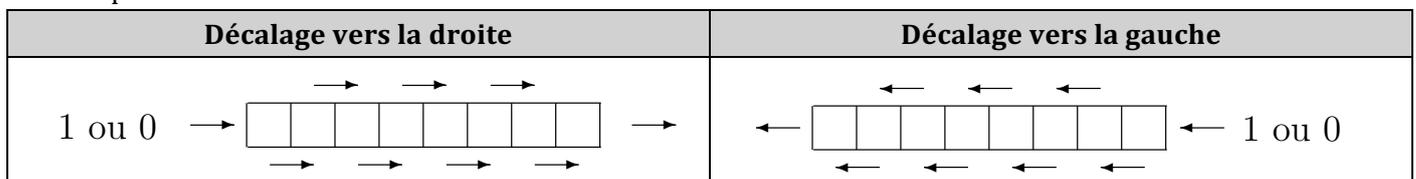
6.1. Définition

Un mot, en informatique, est l'unité de base manipulée par un microprocesseur. La taille d'un mot s'exprime en bits ou en octets et est souvent utilisée pour classer les microprocesseurs (8 bits, 16 bits...). Toutes choses égales par ailleurs, un microprocesseur est d'autant plus rapide que ses mots sont longs, car les données qu'il traite à chaque opération sont plus longues. (D'après Wikipédia)

1	1	0	0	0	1	0	1
Mot de 8 bits							

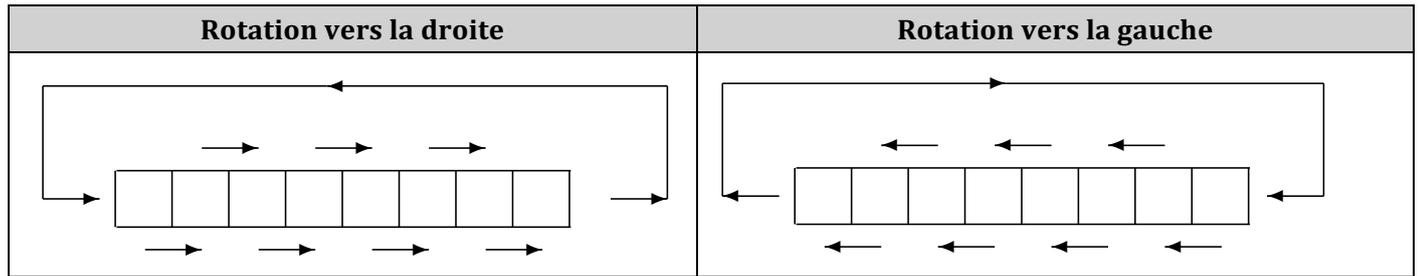
6.2. Décalage

Le décalage consiste à décaler les bits d'un mot soit vers la droite, soit vers la gauche et de remplacer la place vacante par un bit à 0 ou 1.



6.3. Rotation

La rotation est une permutation circulaire bit à bit d'un mot. Cette rotation se fait soit vers la droite, soit vers la gauche.



6.4. Opérations logiques sur des mots

Les opérateurs logiques (NON, ET, OU, OU exclusif) que nous avons vu précédemment, peuvent être appliqué sur un mot, bit par bit.

Exemple :

1	1	0	0	0	1	0	1
ET							
0	0	0	0	1	1	1	1
=							
0	0	0	0	0	1	0	1

Ce type de fonction est très utilisé pour masquer une partie d'un mot. Dans l'exemple ci-dessus, les 4 bits de poids forts sont masqués.

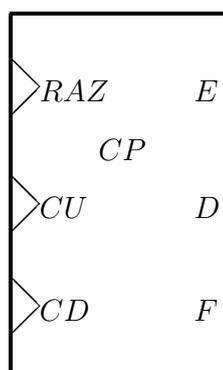
Remarque : $(M \oplus K) \oplus K = M$, cette égalité permet de faire du codage de mot avec une clef privé K.

7. Le compteur

Un compteur est un ensemble de n bascules interconnectées par des portes logiques. Il peut donc mémoriser des entiers codés en binaire naturel sur n bits. Il est généralement muni :

- D'une entrée RAZ pour le mettre à 0 ;
- D'une entrée CU, pour incrémenter le compteur ;
- D'une entrée CD pour le décrémenter ;
- D'une mémoire pour mémoriser la présélection CP ;
- D'une sortie débordement décomptage E ;
- D'une sortie présélection atteinte D ;
- D'une sortie débordement comptage F.

Exemple de schématisation :



8. CodeSys et la logique séquentielle

8.1. Généralités

Vous devez fournir un nom (placé au-dessus) pour tous les composants de logique combinatoire que vous utilisez. Vous pourrez utiliser toutes les entrées/sorties des composants en utilisant leur nom, soit `NOM_DU_COMPOSANT.NOM_DE_LA_SORTIE`.

8.2. Fronts

Front montant - R_TRIG			Front descendant - F_TRIG		
Entrée			Sortie		
CLK	BOOL		Q	BOOL	

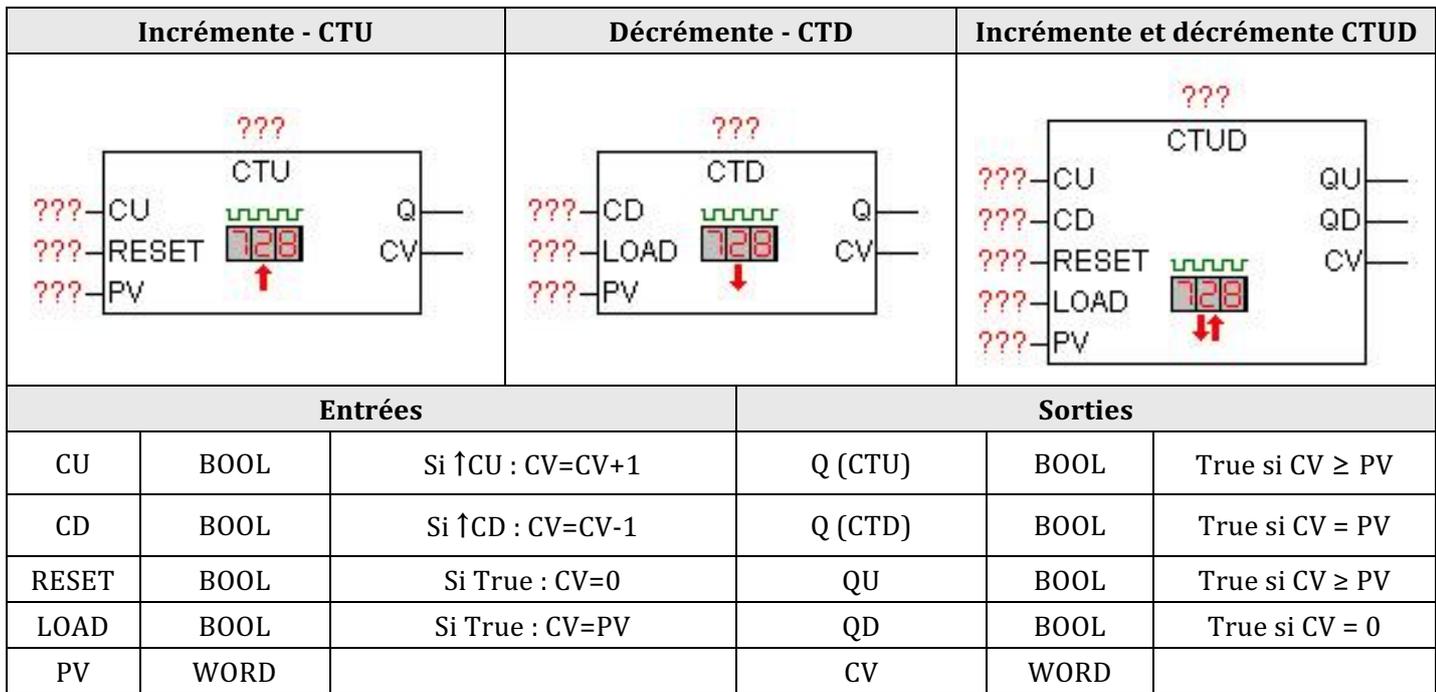
8.3. Bascules RS et SR

Bascule SR - Set prioritaire			Bascule RS - Reset prioritaire		
Entrées			Sortie		
SET1	BOOL		Q1	BOOL	
RESET	BOOL				
SET	BOOL				

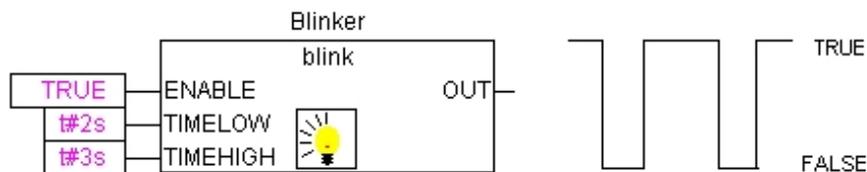
8.4. Temporisations

Travail - TON		Repos - TOF		Monostable - TP	
Entrées			Sorties		
IN	BOOL		Q	BOOL	
PT	TIME	T#5s	ET	TIME	

8.5. Compteurs



8.6. Générateur de créneaux



Le bloc fonctionnel BLINK (dans la bibliothèque IO_IPC/util.lib) génère un signal créneaux. Si la valeur TRUE est affectée à l'entrée ENABLE, alors BLINK est activé, ce qui signifie que la valeur TRUE sera affectée à la sortie pendant la durée TIMEHIGH et ensuite la valeur FALSE sera affectée à la sortie pendant la durée TIMELOW.

Entrées			Sorties		
ENABLE	BOOL	Active le bloc	OUT	BOOL	
TIMELOW	TIME	Durée de la partie à 0			
TIMEHIGH	TIME	Durée de la partie à 1			

8.7. Blocs de comparaison

On peut être amené à comparer deux nombres pour satisfaire au cahier des charges. Il suffit alors d'écrire une condition dans un bloc **Entrée**.

Exemple :

<div style="border: 1px solid black; padding: 2px; display: inline-block;">C1.CV=2</div>	<ul style="list-style-type: none"> Vrai si la valeur (CV) du compteur C1 est égale à 2 ; Faux si la valeur (CV) du compteur C1 n'est pas égale à 2.
<div style="border: 1px solid black; padding: 2px; display: inline-block;">C1.CV>2</div>	<ul style="list-style-type: none"> Vrai si la valeur (CV) du compteur C1 est supérieure à 2 ; Faux si la valeur (CV) du compteur C1 n'est pas supérieure à 2.