

Les systèmes logiques séquentiels

I. Introduction

On appelle système séquentiel asynchrone un système correspondant à la structure de la *Figure 1*.

Les variables indépendantes x_i ($i \in [1, n]$) sont appelées **variables d'entrée** du système séquentiel. Les variables z_i ($i \in [1, p]$) sont les **variables de sortie**. Les sorties rebouclées y_i ($i \in [1, m]$) et retardées sont les **variables internes**.

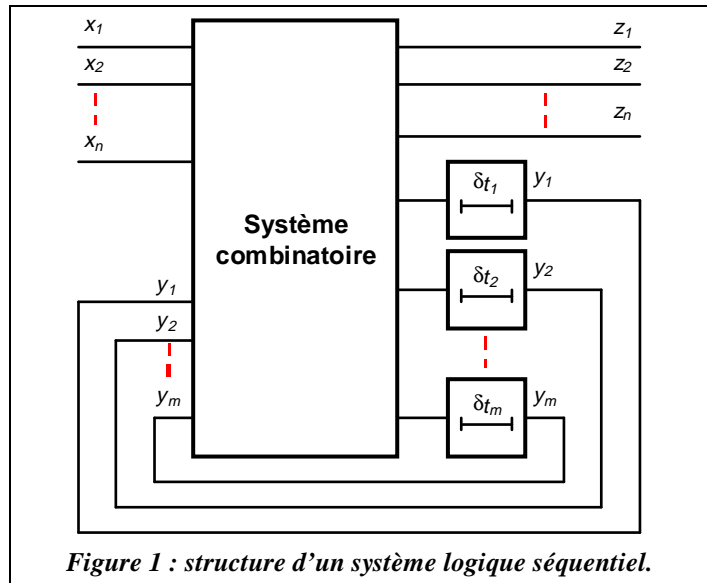


Figure 1 : structure d'un système logique séquentiel.

II. Les bascules

À l'instar des opérateurs logiques élémentaires en logique combinatoire, les bascules (*flip-flop*) sont les éléments de base de la logique séquentielle.

II.1. Mise en place de la fonction

Le montage de la *Figure 2* fonctionne comme l'indique le *Tableau 1*. Par commodité, on contracte l'écriture dans le *Tableau 2*. L'examen de cette table permet d'indiquer que S et S^* sont complémentaires sauf dans le cas où a et b sont simultanément à 0. Pour utiliser cette complémentarité des sorties, on estime que les entrées ne doivent jamais être simultanément à zéro. L'emploi de cette combinaison est prohibé. Ce montage prend le nom de bascule $\overline{R} \overline{S}$ avec $a = \overline{S}$, $b = \overline{R}$, $S = Q$ et $S^* = \overline{Q}$.

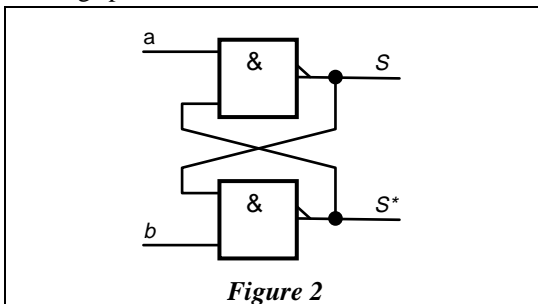


Figure 2

a	b	S	S^*
0	0	1	1
0	1	1	0
1	0	0	1
1	1	0	1
1	1	1	0

Tableau 1

\overline{S}	\overline{R}	Q_{n+1}
0	X	1
1	0	0
1	1	Q_n

Tableau 2

II.2. Bascule RS asynchrone

À partir de l'exemple précédent, on définit la bascule RS asynchrone (*Figure 3*). Son fonctionnement est donné dans la *table des états* (*Tableau 3*). Q_n est l'état de la bascule avant le changement de l'une des entrées et Q_{n+1} est le nouvel état pris par la sortie. S place la sortie à « 1 » (set) et R à « 0 » (reset).

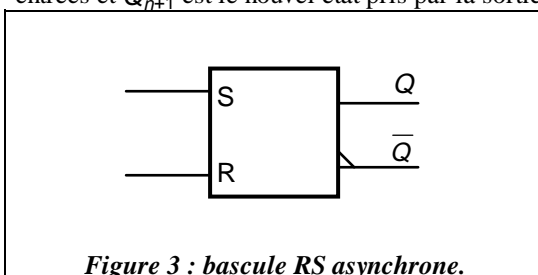


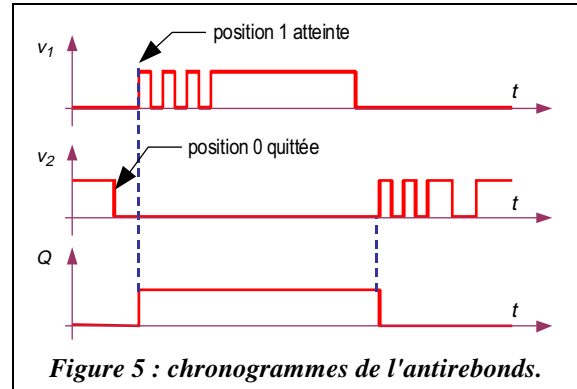
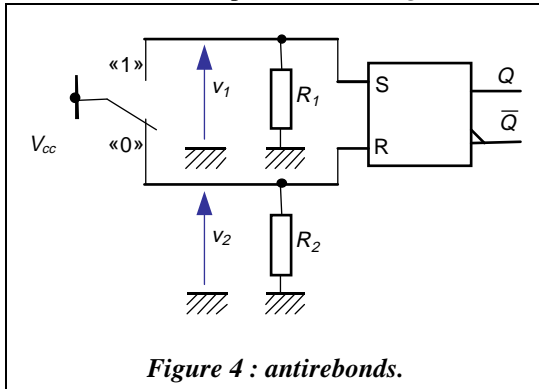
Figure 3 : bascule RS asynchrone.

S	R	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	X

Tableau 3

Application de la bascule RS : antirebonds pour contact électrique

Le schéma de la bascule antirebond est donné à la **Figure 4** tandis que les chronogrammes de fonctionnement sont présentés à la **Figure 5**.

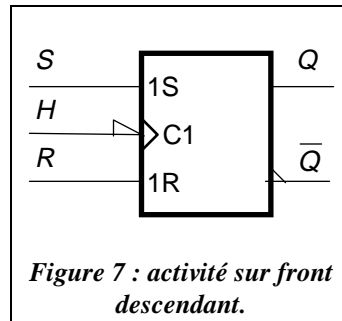
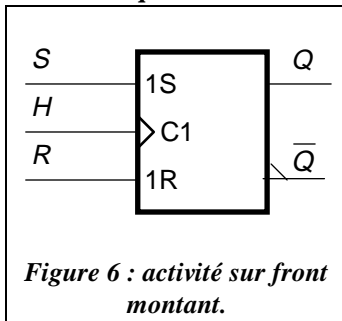


II.3. Bascule RS synchrone (ou bascule RSH)

Dans une bascule synchrone, un signal complémentaire, l'horloge, autorise le changement des sorties de manière statique (sur niveau logique 0 ou 1) ou dynamique (front montant ou descendant). La bascule RS synchrone de la **Figure 6** est active sur les fronts montants de l'horloge H . Son fonctionnement est donné dans le **Tableau 4**.

On peut retrouver ce fonctionnement à l'aide des indications du symbole : le triangle face à l'entrée indique qu'elle est dynamique (front). Elle n'est pas complétement, c'est un front montant. Le repère « 1 » indique les entrées contrôlées (lettre C) par l'horloge.

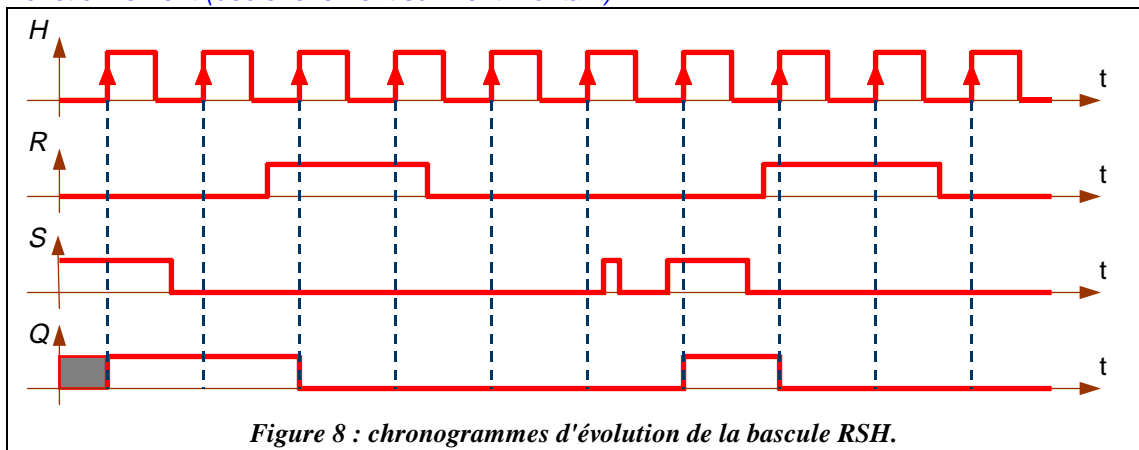
Remarque : on rencontre aussi des bascules déclenchables sur front descendant (**Figure 7**).



S	R	H	Q_{n+1}
0	0	↑	Q_n
0	1	↑	0
1	0	↑	1
1	1	↑	×
×	×	×	Q_n

Tableau 4

Fonctionnement (déclenchement sur front montant)



Remarque : les fronts actifs sont indiqués par une flèche sur le chronogramme de H .

II.4. Evolution vers la bascule JK

II.4.1. Introduction

Malgré l'avantage du synchronisme, l'inconvénient de l'état $RS = 11$ demeure. La solution à ce problème conduit à la bascule JK. Sa structure interne et son symbole sont indiqués à la **Figure 9**.

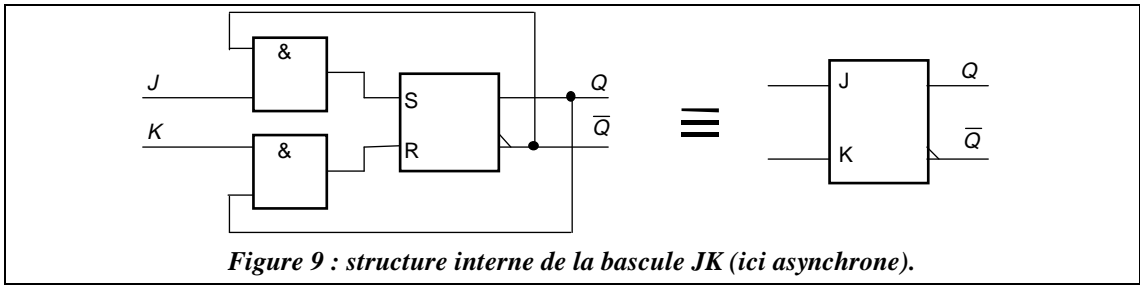


Figure 9 : structure interne de la bascule JK (ici asynchrone).

II.4.2. Caractéristiques de la bascule JK synchrone

Table des états

J	K	H	Q_{n-1}	Q_n
0	0	↑	0	0
0	0	↑	1	1
0	1	↑	0	0
0	1	↑	1	0
1	0	↑	0	1
1	0	↑	1	1
1	1	↑	0	1
1	1	↑	1	0

Equation : synthétisée à partir de la table des états, on obtient $Q_n = J\overline{Q_{n-1}} + \overline{K}Q_{n-1}$

Table de fonctionnement réduite (contraction de la table précédente)

J	K	H	Q_n	Commentaires
0	0	↑	Q_{n-1}	Conservation de l'état précédent (mémorisation)
0	1	↑	0	Mise à zéro (<i>reset</i>)
1	0	↑	1	Mise à un (<i>set</i>)
1	1	↑	$\overline{Q_{n-1}}$	Inversion de l'état (<i>toggle</i>)
×	×	×	Q_{n-1}	Pas de changement

II.5. Bascule dérivée de la JK : bascule D

La bascule D (Figure 10) est dérivée de la bascule JK avec la condition $J = \overline{K} = D$. On a alors : $Q_n = D$.

Ceci justifie son appellation D (D pour *Delay*, retard) : elle décale dans le temps l'état de la sortie.

Remarque : une bascule D ne disposant pas d'entrée d'horloge est appelée verrou (*latch*).

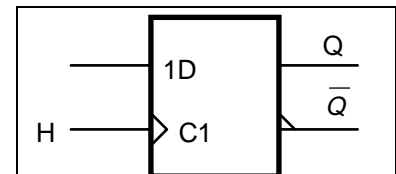


Figure 10 : bascule D déclenchée sur front.

II.6. Fonctionnement forcé des bascules (forçage)

Il est parfois nécessaire d'affecter le niveau de sortie d'une bascule de manière non synchrone : c'est le rôle des entrées de forçage. On distingue des forçages **synchrone**s (sous le contrôle de l'horloge) et d'autres **asynchrone**s qui s'imposent **immédiatement** aux sorties (symboles aux Figure 11 et Figure 12).

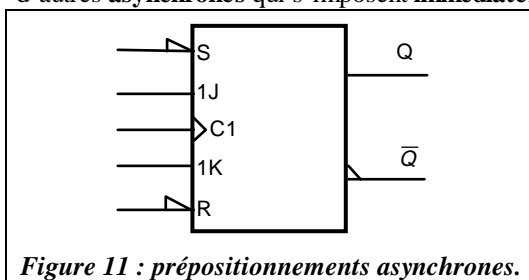


Figure 11 : prépositionnements asynchrones.

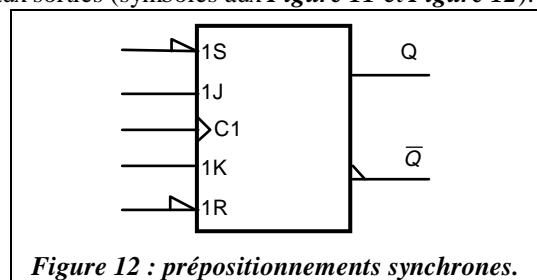


Figure 12 : prépositionnements synchrones.

III. Les registres

III.1. Introduction

Le registre (*register*) résulte de l'assemblage d'un ensemble combinatoire et séquentiel permettant le stockage d'informations binaires en vue d'une mémorisation temporaire avec ou sans traitement (représentation générale à la **Figure 13**).

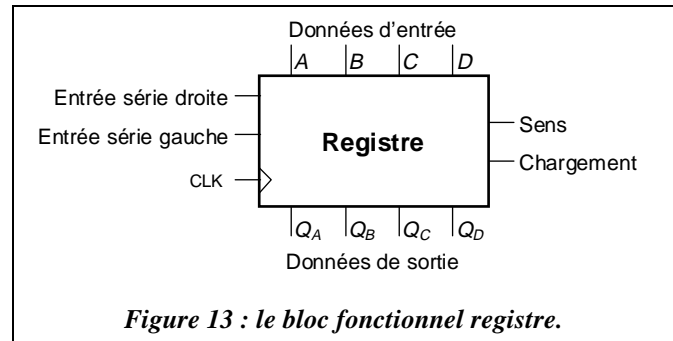


Figure 13 : le bloc fonctionnel registre.

Les registres peuvent réaliser différentes manipulations comme l'indique la **Figure 14**. L'écriture constitue un chargement du registre. La lecture (a) est immédiatement possible car les sorties représentent le contenu du registre. En conséquence la lecture demeure possible (b) tant que le contenu n'est pas modifié (mémorisation). Le décalage à droite (c) ou à gauche (d) rend libre une « case » du registre pour recevoir un bit série au travers de l'entrée correspondante. L'autre bit extrême est perdu. A titre indicatif, on remarquera que l'entrée série est dénommée de la même manière que le sens de décalage, alors qu'elle est positionnée de l'autre côté.

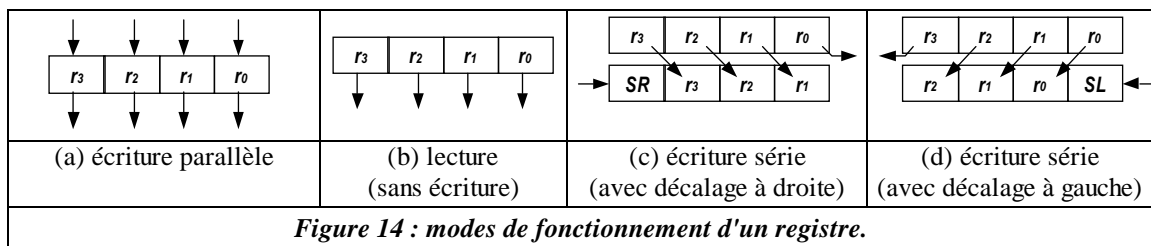


Figure 14 : modes de fonctionnement d'un registre.

III.2. Analyse du registre universel de type 194

Il n'est plus nécessaire aujourd'hui d'effectuer la synthèse des registres que l'on utilise : un choix très vaste est offert par les constructeurs. A titre d'exemple, le registre universel du type 194 est présenté.

C'est un registre à chargement parallèle ou série, avec la possibilité d'un déplacement de l'information vers la droite (Q_A vers Q_D) ou la gauche (Q_D vers Q_A).

III.2.1. Symbole IEC et description des entrées et sorties

Le symbole normalisé IEC du registre de type 194 est donné à la **Figure 15**.

- CLK, l'entrée d'horloge, synchronise le registre sur fronts montants.
- CLR, entrée asynchrone de remise à zéro des sorties.
- A, B, C et D, entrées de chargement parallèle.
- SR (*shift right*), entrée de chargement série côté droit, SL (*shift left*), entrée de chargement série côté gauche.
- S0 et S1, les entrées de contrôle synchrones. Les modes de fonctionnement sont donnés dans le **Tableau 5**.
- Q_A , Q_B , Q_C et Q_D , sorties du registre et représente son contenu. Utilisées simultanément, la lecture est parallèle. Si seule la dernière est utilisée, la lecture est série.

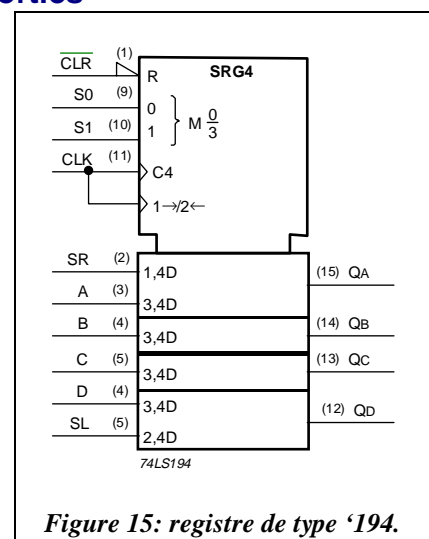


Figure 15: registre de type '194.

S1	S0	Mode de fonctionnement
0	0	Inhibition (registre figé malgré l'horloge)
0	1	Chargement série par l'entrée SR, déplacement de l'information à droite (Q_A vers Q_D)
1	0	Chargement série par l'entrée SL, déplacement de l'information à gauche (Q_D vers Q_A)
1	1	Chargement parallèle par les entrées A, B, C et D.

Tableau 5 : les modes de fonctionnement du '194.

III.2.2. Chronogrammes de fonctionnement

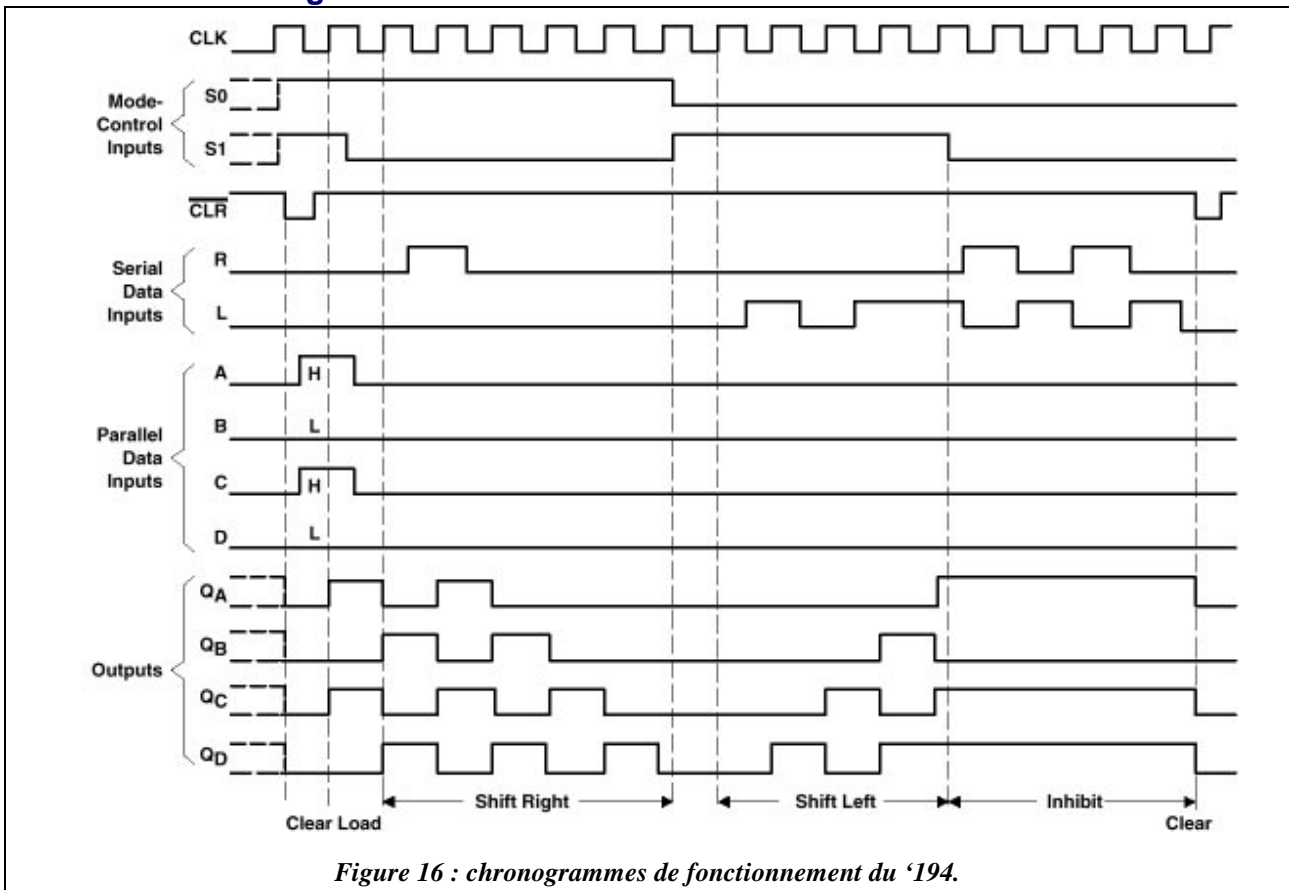


Figure 16 : chronogrammes de fonctionnement du '194.

IV. Les compteurs

IV.1. Introduction

La fonction qui s'impose pour dénombrer des événements numériques est le **compteur**.

Un compteur (*counter*) résulte de l'assemblage d'un ensemble combinatoire et séquentiel (bascules JK le plus souvent) cadencé par un signal d'horloge H . La combinaison des n bits de sortie forme un mot binaire qui évolue en croissant ou décroissant au rythme de l'horloge.

IV.2. Les compteurs asynchrones

IV.2.1. Introduction

La réalisation d'un compteur asynchrone (*asynchronous counter*) consiste à mettre en cascade des bascules (assurant la fonction de diviseur par deux), détecter la combinaison de remise à zéro puis l'appliquer aux entrées de remise à zéro de chaque bascule.

Cette méthode n'assure pas un fonctionnement « fluide » du système : la rupture du séquençement montre le comportement asynchrone qui génère des aléas (défauts de fonctionnement).

IV.2.2. Exemple de compteur asynchrone

La structure est donnée à la *Figure 17*. Les chronogrammes illustrant le fonctionnement sont tracés à la *Figure 18*.

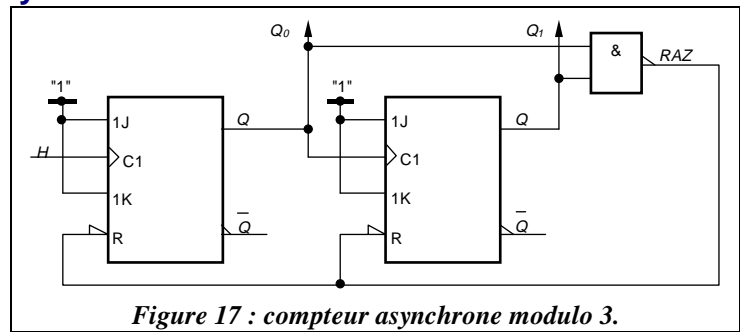


Figure 17 : compteur asynchrone modulo 3.

Si la rupture n'existe pas, la longueur de la séquence (modulo du compteur) est un poids binaire (2^n). Pour un modulo différent, le redémarrage à 0 est assurée après la dernière combinaison souhaitée en détectant la combinaison qui succède à celle de rupture à l'aide d'un système combinatoire (une fonction qui prend 1 pour un seul minterme). La sortie pilote alors la remise à zéro de toutes les bascules. La combinaison détectée apparaît furtivement (sa présence dépend des retards des composants).

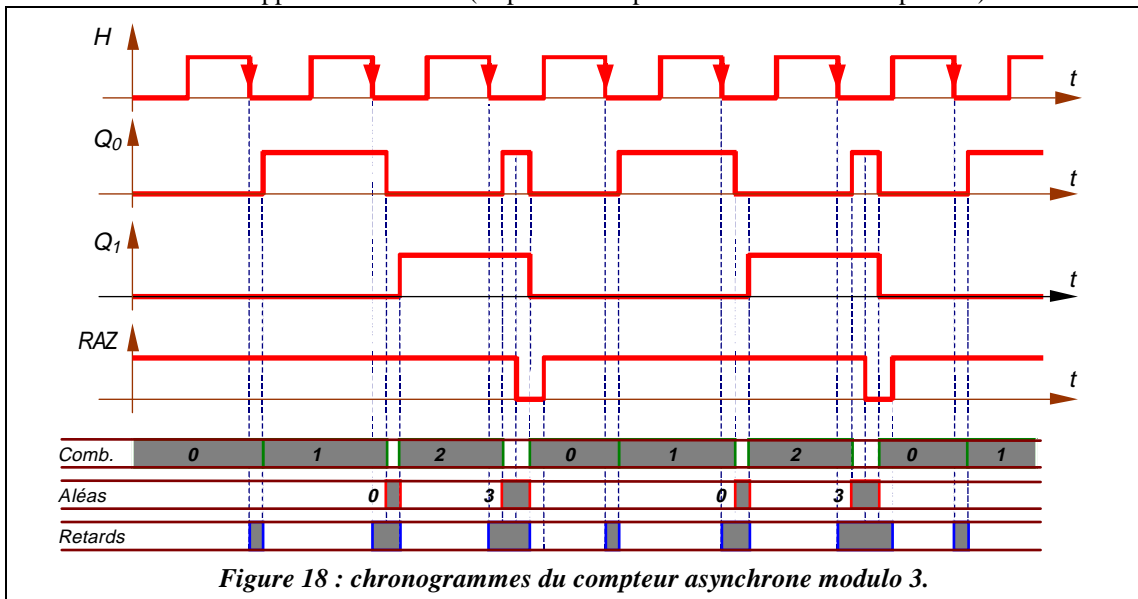


Figure 18 : chronogrammes du compteur asynchrone modulo 3.

IV.3. Les compteurs synchrones

Le comptage synchrone, utilise le signal de référence (l'horloge) pour éviter le problème d'aléas liés au décodage de la combinaison de remise à zéro. Ceci implique que le système évolue naturellement vers sa combinaison de bouclage. Pour cela, les entrée J et K des bascules sont pilotées par un ensemble combinatoire. L'entrée *reset* des bascules n'est plus nécessaire.

Les méthodes de synthèse de ce genre de compteur n'est pas présenté dans ce document.

IV.4. Mise en oeuvre des compteurs intégrés

IV.4.1. Compteurs intégrés : le prêt à porter numérique

Dans la pratique, on ne synthétise que très rarement des compteurs. On préfère utiliser des fonctions précâblées. Ces composants possèdent différentes broches permettant d'adapter le fonctionnement à un maximum de situations (standardisation).

IV.4.2. Analyse des compteurs de type 190 et 191

La référence 190 correspond à un compteur/décompteur par 10 tandis que le 191 est modulo 16. Mis à part ces différences, les fonctionnements sont identiques. C'est la raison pour laquelle nous ne nous attacherons qu'au 190.

IV.4.2.1. Les entrées

Clock est l'entrée d'**horloge**. Cette dernière synchronise le compteur sur ses fronts montants.

Le système peut **compter ou décompter** suivant l'état de D/\bar{U} (*Down/Up*).

Le fonctionnement du compteur peut être bloqué en plaçant l'entrée \overline{CTEN} (*Count Enable*) au niveau bas. Dans cette éventualité, les sorties du compteur sont figées.

A, B, C et D sont les entrées de préchargement. La combinaison binaire de quatre bits devient l'état interne du compteur (et apparaît donc aussi en sortie) dès que l'entrée \overline{LOAD} (*To load = charger*) passe à 0. Il s'agit donc d'une opération de **préchargement asynchrone**.

IV.4.2.2. Les sorties

MAX/MIN reste à 1 pendant que la combinaison en sortie est 1001 (= 9) en mode comptage et 0000 en mode décomptage. Elle permet d'**indiquer un débordement prochain du compteur**.

\overline{RCO} (*Ripple Clock Output*) permet de synchroniser le fonctionnement d'un autre compteur réaliser une mise en cascade. Reliée à l'entrée \overline{CTEN} d'un boîtier suivant, ce dernier serait bloqué durant toute la phase de comptage du premier compteur (unités par exemple) car \overline{RCO} est à 1. Au moment où le compteur des unités va boucler « un tour », \overline{RCO} se positionne à 0, validant ainsi l'étage suivant et permettant donc la prise en compte du front d'horloge pour s'incrémenter (d'une dizaine par exemple).

IV.4.2.3. Modulos différents de 10

Pour réaliser un compteur bouclant avant 9, on décode le modulo (grâce à une fonction ET). Le résultat, synchronisé avec l'horloge, assure un niveau 0 pour le chargement de la combinaison 0000 préalablement fixée sur ABCD.

IV.4.3. Analyse du symbole IEC

Les symboles normalisé IEC des compteurs 190 et 191 aux *Figure 19* et *Figure 20*. Les différentes notations permettent de **décrire complètement le fonctionnement de ce composant numérique**.

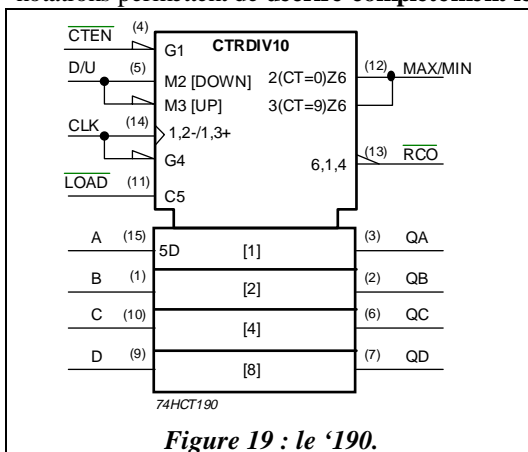


Figure 19 : le '190.

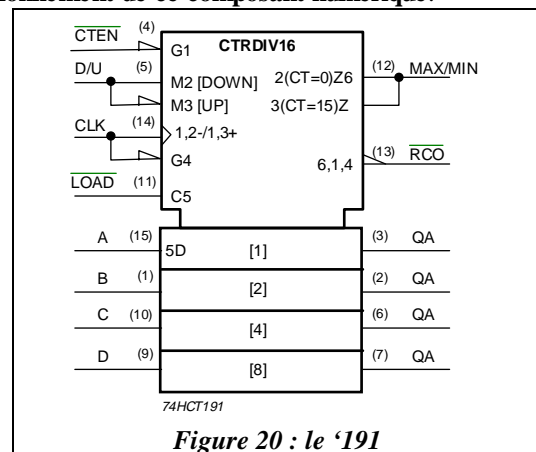


Figure 20 : le '191

Le symbole est formé de deux blocs essentiels. Le **cadre de contrôle** ou de commande (en haut) caractérisé par les deux échancrures. Il regroupe toutes les indications concernant **l'ensemble de la fonction**. Dans sa partie supérieure, on remarque la fonction réalisée (compteur diviseur par 10 ou 16). Le cadre du bas est subdivisé en quatre parties à l'image des quatre bits de données. Les numéros de broche sont toujours indiqués entre parenthèses. Toutes les indications à l'intérieur des cadres sont normalisées, c'est à dire **invariantes et immuables**. Ce qui est à l'extérieur est la prérogative de l'utilisateur (mais il est conseillé de conserver les indications données par le constructeur).

On retrouve les notations classiques de la norme IEC : les numéros repères qui renvoient aux broches (à ne pas confondre avec le numéro des broches), les notations de dépendance des broches (G pour une entrée d'opérateur ET, Z pour une liaison interne directe), M pour les modes de fonctionnement, C pour un contrôle et D pour une fonction mémoire (entrée de bascule D). Les annotations entre crochets sont indicatives (ici ce sont les poids binaires des entrées et sorties).

On peut dire que ce compteur agit sur quatre groupes de 1 bit. Les états des entrées A, B, C et D sont mémorisées si la broche correspondant au repère 5 est active, c'est à dire une demande de préchargement par un état 0 sur la broche \overline{LOAD} (on remarque bien qu'un contrôle est assuré par cette entrée). La notation 5D n'est indiquée que dans la première case et doit être dupliquée dans celles du dessous.

Deux modes (lettre M) de fonctionnement sont possibles. La broche $\overline{D/U}$ assure les modes comptage ou décomptage : les repères 2 et 3 renvoient à l'effet de l'horloge (signes + et -). D'ailleurs, des indications entre crochets finissent de nous renseigner quant à l'effet observé. On remarque aussi que

l'horloge est active sur front montant (> sans complémentation) et que, sur niveau bas, par le repère 4, elle participe à l'élaboration du signal \overline{RCO} (on peut dire que celui-ci est issu du produit logique de CLK, \overline{CTEN} et MAX/MIN).

L'effet de l'horloge H est autorisé par la broche \overline{CTEN} puisque ces entrées sont liées par un ET (lettre G, repère 1). On voit aussi très bien l'incidence du mode de fonctionnement (repères 2 et 3).

La sortie MAX/MIN est issue d'un OU câblé entre les états 0 (dans le sens du décomptage, donc mode 2) et 9 (ou 15, dans le sens du comptage, donc mode 3) du compteur.

IV.4.4. Chronogrammes de fonctionnement (cas du '191)

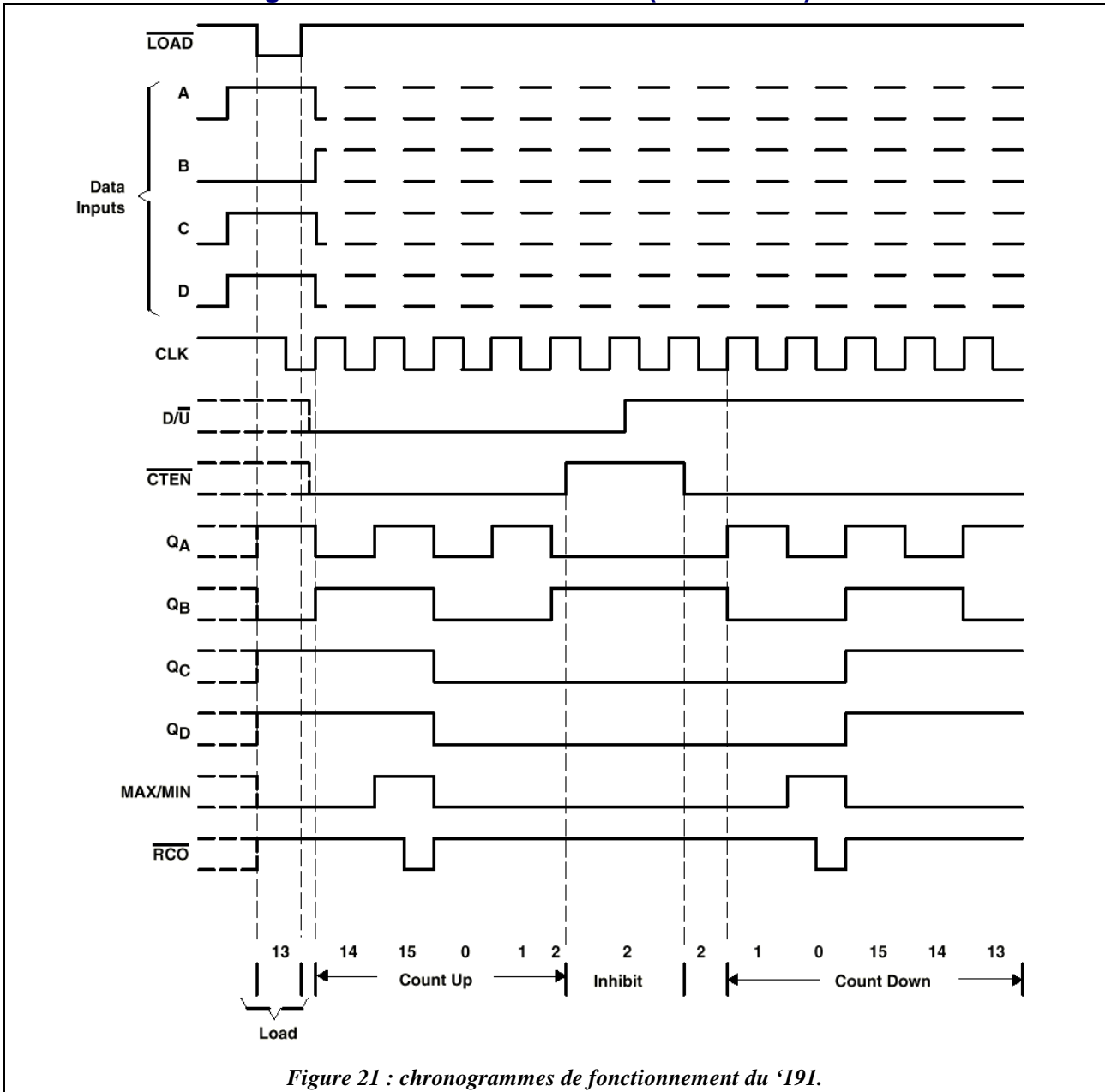
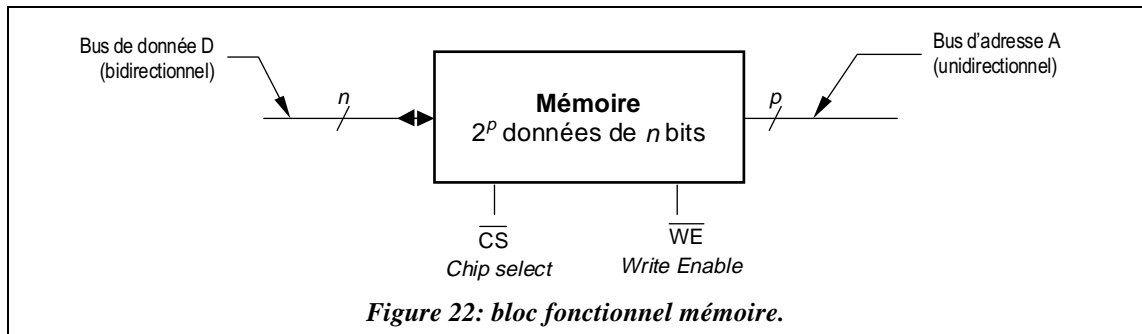


Figure 21 : chronogrammes de fonctionnement du '191.

V. Les mémoires

V.1. Introduction

Les mémoires (*memory-ies*) sont les éléments permettant la sauvegarde de données en quantité variable. Le symbole fonctionnel est présenté à la **Figure 22**. L'adresse est le mot binaire qui permet de localiser la donnée qui entre et sort par un canal unique appelé bus de données qui est bidirectionnel (deux sens possibles).



Dans le cas général, le fonctionnement d'une mémoire est découpé en deux cycles.

Mécanisme de **lecture** :

- appliquer le mot adresse sur le bus d'adresse,
- sélectionner le boîtier mémoire en appliquant un niveau logique bas sur la ligne \overline{CS} ,
- sélectionner le mode « lecture » en appliquant un niveau logique haut sur la ligne \overline{WE} ,
 - ◆ dès cet instant, l'adresse est décodée et la donnée correspondante se présente sur les sorties de données (D_i),
 - ◆ le retour de \overline{CS} à l'état haut fait passer le bus de données à l'état haute impédance.

Mécanisme d'**écriture** :

- appliquer le mot d'adresse sur le bus d'adresse,
- appliquer le mot de donnée sur le bus de données,
- sélectionner le boîtier mémoire en appliquant un niveau logique bas sur la ligne \overline{CS} ,
- sélectionner le mode « écriture » en appliquant un niveau logique bas sur la ligne \overline{WE} ,
- dès cet instant, l'adresse est décodée et les données présentes en entrées sont prises en compte si elles demeurent pendant une durée suffisante.

V.2. Les différents types de mémoires

Les mémoires sont classées suivant deux familles :

- mémoires mortes (en anglais, ROM pour *Read Only Memory*, mémoire à lecture seule) ;
- mémoires vives (en anglais, RAM pour *Random Access Memory*, mémoire à lecture aléatoire).

V.2.1. Les mémoires mortes

Le contenu est fixé à la construction ou par l'utilisateur. La disparition de l'alimentation électrique n'altère pas le contenu.

Les ROM sont utilisées pour stocker des informations figées telles que les générateurs de caractères matriciels, les programmes fixes dans des machines programmées ou les tables de conversion de données.

V.2.2. Les mémoires vives

On distingue deux types de RAM : statiques (le contenu est modifié par un nouveau) ou dynamiques (l'information est conservée par un condensateur et il faut la régénérer). Toute disparition de l'alimentation électrique efface le contenu.

Dès qu'un système doit conserver temporairement des informations, la RAM trouve sa place. En informatique, elles sont largement mises en œuvre en quantités importantes (plus de 16 Mo en micro-informatique et plusieurs centaines de méga octets en mini-informatique).

V.2.3. Les mémoires programmables et effaçables par l'utilisateur

Les mémoires programmables sont intermédiaires entre les RAM et les ROM. Leur contenu peut être défini par l'utilisateur et subsister sans alimentation électrique. Leur mise en œuvre est aussi aisée que celle des RAM sans leur volatil inconvénient.

On en rencontre de différentes familles :

- les PROM (*Programmable ROM*) sont composées de liaisons que l'on peut détruire une seule fois ;
- les réseaux logiques programmables (PLA, *Programmable Logic Array*), une sorte de PROM adaptée à la confection de circuits combinatoires ;
- les composants logiques programmables (PLD, *Programmable Logic Device*) offrent un ensemble de structures séquentielles et combinatoires préintégréées. Des connexions fusibles permettent de programmer le composant suivant les besoins ;
- les mémoires effaçables (EPROM, *Erasable PROM*).
- les mémoires effaçables électriquement (EEPROM, *Electrical Erasable PROM*).

Les mémoires PROM sont bien adaptées aux ensembles fabriqués en séries restreintes ou souvent renouvelées. Les EPROM trouvent emploi dans les petites séries et les prototypes. Quant aux mémoires EEPROM, elles sont bien adaptées à la sauvegarde hors alimentation des paramètres de configuration.