

Travaux Pratiques
Informatique Industrielle

Logique
Séquentielle

Table des Matières

Quelques conseils	5
1 - Bascules et Monostables	7
2 - Compteurs Asynchrones	15
3 - Compteurs Synchrones.....	21
4 - Registres à décalage	27
Annexes.....	31
Annexe I - Brochage Circuits Intégrés TTL	33
Annexe II - Compléments Circuits Intégrés	35

Quelques conseils ...

La préparation du TP

La préparation du TP est importante. Vous devez, avant de venir en TP, avoir préparé votre travail, c'est à dire avoir rédigé la partie théorique du TP. De plus, *la préparation est individuelle* et ne doit pas se faire en prenant exemple sur le travail d'un collègue.

La mise au point du montage

Pour éviter les erreurs de câblage, il est très important de dessiner le montage en indiquant les numéros de broches. D'ailleurs, la notation prendra en compte ce point.

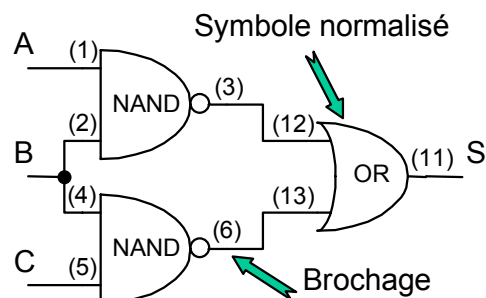
Une autre panne courante est d'avoir un ou des circuits non alimentés. C'est à dire, le 5 volts ou la masse ne sont pas connectés aux circuits.

Votre *comportement en TP*, c'est à dire votre autonomie, la façon dont vous résolvez les problèmes, *votre efficacité est prise en compte dans la note.*

Le compte-rendu

Le compte-rendu doit comporter pour chaque exercice:

- une partie théorique,
- un ou des schémas avec des symboles normalisés et le brochage comme l'exemple de la figure ci-dessous,



- une analyse des résultats obtenus.

Enfin, la présentation en elle-même de tout travail est importante.

Le contrôle

Le contrôle a lieu à la fin des TPs. Il dure 1h20 et sera individuel. Le contrôle rentre pour $\frac{3}{4}$ dans la note finale des Travaux Pratiques.

1 - Bascules et Monostables

Matériels fournis

2 × 7400, 7404, 7410, 7473, 7474, 74121, 74123; 74279, Générateur TTL, R, C

1.1 Bascules

La bascule (ou flip-flop en anglais) est le circuit séquentiel le plus simple remplissant la fonction de mémorisation. Son rôle consiste à enregistrer une information fugitive et à conserver cet état lorsque l'information disparaît. Dans son fonctionnement, le temps joue un rôle essentiel.

1.1.1 La Bascule RS

1.1.1.a Principe

La bascule RS dispose de deux entrées S (Set) et R (Reset) et deux sorties Q et \bar{Q} . La bascule a deux états stables $Q=1, \bar{Q}=0$ et $Q=0, \bar{Q}=1$. Son état dépend du signal appliqué sur les entrées S et R et de son état Q^- antérieur à l'application du signal d'entrée. Le fonctionnement de la bascule est résumé ci-dessous.

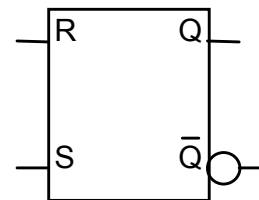


Figure 1-1: Bascule D.

- Si $R = S = 0$, la bascule conserve son état antérieur, c'est à dire $Q = Q^-$,
- Si $R = 0$ et $S = 1$, $Q = 1$ quel que soit Q^- ,
- Si $R=1$ et $S = 0$, $Q = 0$ quel que soit Q^- ,
- Enfin, $R = S = 1$ conduit à un état interdit.

Travail demandé

- Construire la table de vérité donnant l'état de sortie Q en fonction des variables d'entrée R et S et de l'état initial Q^- .

1.1.1.b Schémas internes d'une bascule RS

On peut constituer une bascule RS soit avec des portes NOR, soit avec des portes NAND, comme l'indiquent la Figure 1-2.

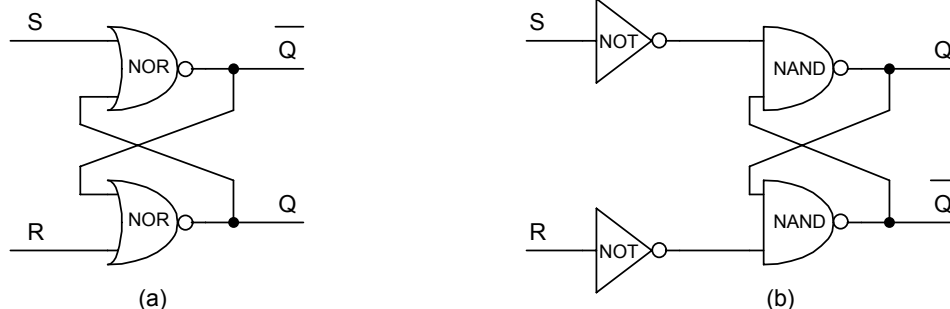


Figure 1-2: Schémas internes d'une bascule RS en NOR ou en NAND.

Travail demandé

a) Vérifier que ces deux schémas sont équivalents de façon théorique.

1.1.1.c Etude dynamique d'une bascule réalisée avec des portes NAND

Nous appelons θ_1 et θ_2 les retards apportés par les deux portes NAND constituant la bascule RS. Nous allons étudier le comportement dynamique de la bascule RS lorsqu'on lui applique les impulsions de commande représentées sur le figure suivante.

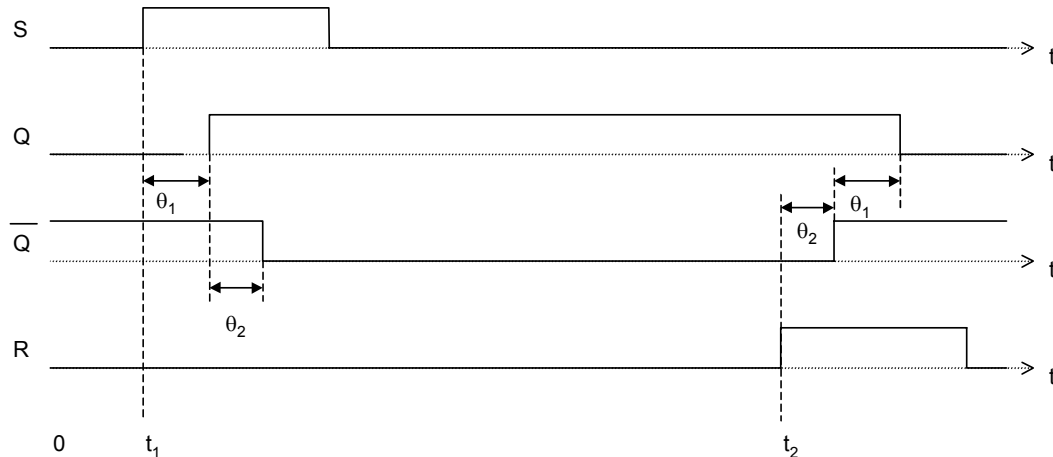


Figure 1-3: Comportement dynamique d'une bascule RS.

Pour $0 < t < t_1$: $R = S = 0$, la bascule conserve son état (dans notre exemple $Q = 0$).

Pour $t = t_1$: S devient égal à 1.

Pour $t_1 < t < t_1 + \theta_1$: S reste égal à 1.

Pour $t = t_1 + \theta_1$: Q devient égal à 1.

Pour $t_1 + \theta_1 < t < t_1 + \theta_1 + \theta_2$: S reste égale à 1.

Pour $t = t_1 + \theta_1 + \theta_2$: \overline{Q} passe de 1 à 0.

Ainsi si S reste égal à 1 pendant un intervalle de temps supérieur ou égal à $\theta_1 + \theta_2$ la bascule passe de l'état $Q=0$ à l'état $Q=1$. (θ_1 et θ_2 sont typiquement de l'ordre de 10ns).

Lorsque S revient à 0, on a alors $S = R = 0$ et la bascule conserve l'état $Q = 1$ c'est à dire remplit sa fonction de mémorisation.

Notons enfin que si $R = S = 1$, on a $Q = \overline{Q} = 1$. Lorsque R et S reviendront à 0, la bascule se stabilisera aléatoirement soit dans l'état $Q = 1$, soit dans l'état $Q = 0$. Son état final est indéterminé. C'est pourquoi ce type de fonctionnement doit être évité dans les bascules RS.

Travail demandé

a) Expliquer ce qui se passe pour $t > t_2$.

b) Construire une bascule RS avec des portes NAND. Tester son fonctionnement en le comparant à la table de vérité.

1.1.1.d Bascule RS synchrone

La plupart du temps, lorsqu'on utilise une bascule, il est préférable de mettre les entrées à leurs niveaux respectifs et de déclencher le fonctionnement de la bascule par une impulsion d'horloge. Le schéma d'une bascule RS synchrone est représenté ci-dessous.

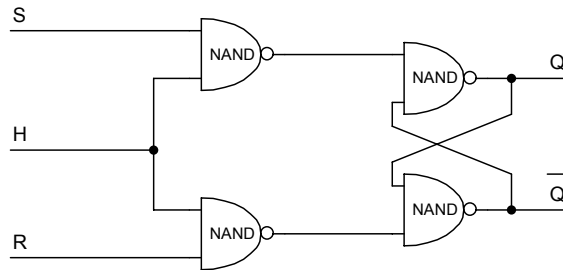


Figure 1-4: Schéma d'une bascule RS synchrone.

Travail demandé

- Construire cette bascule. Expliquer et vérifier son fonctionnement.
- Tester le fonctionnement du circuit SN74279.

1.1.2 Bascule D synchrone

On a vu que le type de fonctionnement $R = S = 1$ devait être évité pour une bascule RS car il conduisait à un état indéterminé.

Un moyen de s'assurer qu'il ne peut pas se produire ce type de situation est de n'utiliser qu'une entrée. C'est le cas de la bascule D appelé aussi bascule T dont le schéma est représenté Figure 1-5.

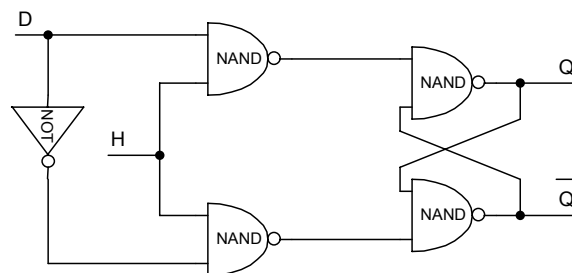


Figure 1-5: Schéma d'une bascule T.

Travail demandé

- Etablir la table de vérité de cette bascule.
- Construire le circuit et le tester. Existe-t-il des états indéterminés ?
- Tester le fonctionnement du circuit SN7474.

1.1.3 Bascule JK

C'est une bascule à deux entrées qui ne présente pas d'état indéterminé. Son schéma est représenté ci-dessous.

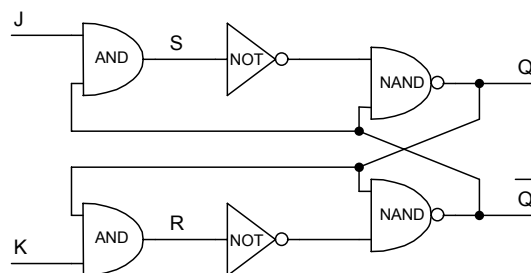


Figure 1-6: Schéma d'une bascule JK.

On appelle Q^- l'état initial de la bascule. La table de vérité s'écrit comme suit:

J	K	Q^-	$S = J/Q^-$	$R = K.Q^-$	Q	Etat
0	0	0	0	0	0	Mémoire
0	0	1	0	0	1	
0	1	0	0	0	0	
0	1	1	0	1	0	
1	0	0	1	0	1	
1	0	1	0	0	1	
1	1	0	1	0	1	Commutation
1	1	1	0	1	0	

Tableau 1-1: Table de vérité de la Figure 1-6.

On constate qu'il n'y a plus d'état indéterminé même lorsque $J = K = 1$.

On distingue en fait deux fonctions:

- une fonction de mémorisation,
- une fonction de commutation ($J = K = 1$). La bascule prend l'état inverse de l'état précédent.

On peut résumer son fonctionnement comme suit:

J	K	Q
0	0	Q^-
0	1	0
1	0	1
1	1	$\overline{Q^-}$

Tableau 1-2: Table de vérité de la bascule JK.

Mettons tout de suite en évidence sur un exemple la difficulté qui existe dans le mode de fonctionnement en commutation. Supposons qu'à $t=0$, $Q=0$ et que l'on applique aux entrées $J=K=1$.

Au temps θ , on va observer $Q=1$. Mais si l'on a toujours à l'entrée $J=K=1$, au temps 2θ on observera $Q=0$ et ainsi de suite ...

La sortie Q va commuter plusieurs fois, sauf si $J=K=1$ n'est appliqué à l'entrée que durant un intervalle de temps τ inférieur à θ . Nous allons voir dans la suite que le fonctionnement en mode maître/esclave permet de s'affranchir de cette difficulté.

Travail demandé

- Réaliser une bascule JK.
- Tester son fonctionnement et le comparer à la table de vérité.

1.1.4 Bascule RS maître/esclave

1.1.4.a Fonction de décalage

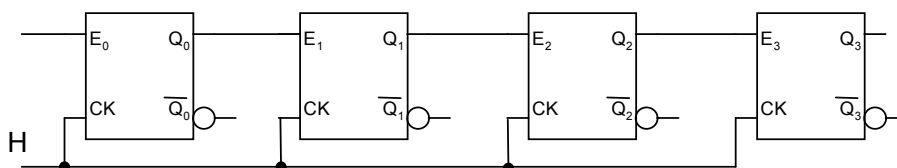


Figure 1-7: Fonction à décalage.

Considérons un ensemble de n cellules. Chaque cellule possède une entrée E_i , une entrée d'horloge H et une sortie Q_i .

La fonction de décalage consiste à faire glisser l'information d'une cellule à la suivante.

Si l'on applique à l'instant t une impulsion de commande sur l'entrée d'horloge H , l'information présente à l'entrée E_i de la i ème cellule est transférée à la sortie Q_i de cette cellule au temps $t+\theta$. Si la durée ΔT de l'impulsion d'horloge est inférieure à θ , l'information se décale d'une cellule. Mais si par exemple $2\theta < \Delta T < 3\theta$ l'information se décalerait de 3 cellules ce qui ne convient pas.

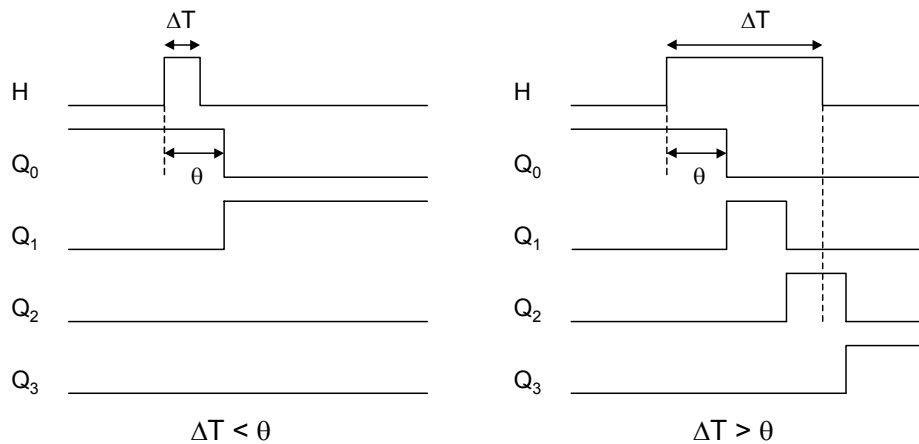


Figure 1-8: Propagation de la fonction à décalage.

Pour pallier à cet inconvénient, on utilise la bascule RS maître/esclave.

1.1.4.b Bascule RS maître/esclave

Le schéma d'une bascule RS maître/esclave est représenté ci-dessous.

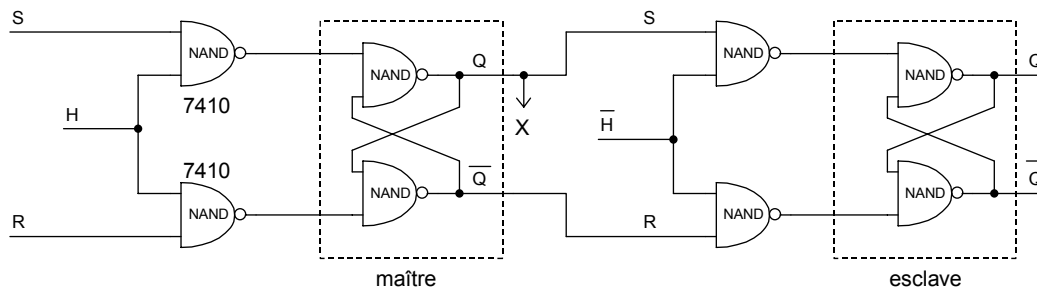


Figure 1-9: Schéma d'une bascule maître/esclave.

Si l'on appelle respectivement Q^- et Q l'état de sortie avant et après l'impulsion d'horloge, on a la table de vérité suivante.

R	S	Q
0	0	Q^-
0	1	1
1	0	0
1	1	?

Tableau 1-3: Table de vérité d'une bascule RS

Ici encore, on a un état indéterminé lorsque $S = R = 1$.

Travail demandé

- Réaliser une bascule RS maître/esclave. Utiliser pour les entrées S et R des circuits SN7410 qui seront utiles pour la suite du TP. Tester et expliquer son fonctionnement. Observer la sortie X de l'étage maître.
- Expliquer comment on peut construire un registre à décalage avec des bascules RS maître/esclave. Quel est l'inconvénient par rapport à l'utilisation de bascules D ?

1.1.5 Bascule JK maître/esclave

Nous avons vu que la bascule JK ne présentait pas d'état indéterminé mais qu'une difficulté apparaissait dans sa fonction de commutation. La configuration maître/esclave représentée ci-dessous remédie à cette difficulté.

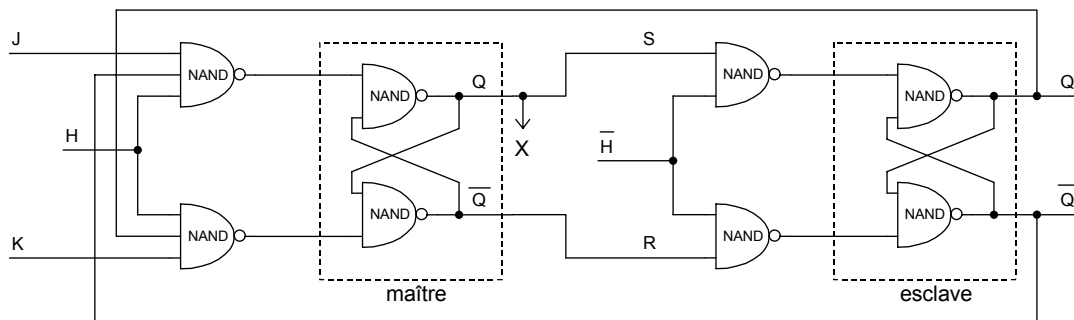


Figure 1-10: Schéma d'une bascule JK maître/esclave.

Supposons que l'état initial soit 1. Donc $Q = X = 1$.

On applique $J=K=1$. Lorsque le front montant de l'horloge arrive, le maître passe dans l'état 0, donc $X=0$. Tant que l'impulsion d'horloge est au niveau 1, l'étage esclave est bloqué. Lorsque le front de descente de l'horloge arrive, l'étage esclave passera au niveau 0, donc $Q=0$.

Ainsi indépendamment de la durée du signal d'horloge, la bascule aura commuté une seule fois, passant de l'état 0 à l'état 1.

Le table de vérité de la bascule JK maître/esclave est la même que celle établie dans le Tableau 1-2.

Travail demandé

- Construire une bascule JK maître/esclave et tester son fonctionnement. Observer la sortie X de l'étage maître.
- Tester le fonctionnement du circuit SN7473 (notamment la fonction *Clear*).

1.2 Monostables

Un monostable génère, à partir d'un front montant ou descendant, une impulsion dont la durée variable est déterminée par une résistance et une capacité extérieures.

Travail demandé

- Réaliser les schémas pour le fonctionnement des circuits SN74121 et SN74123 dont les implantations, les tables de vérité ainsi que les abaques permettant de choisir les composants extérieurs donnés en annexe.

- b) Tester les schémas en appliquant des fronts qui sont obtenus à partir d'un signal TTL délivré par un générateur. Ce signal d'entrée sera observé à l'oscilloscope simultanément au niveau de la sortie Q du monostable étudié.
- c) Comparer les caractéristiques des deux circuits.
- d) S'intéresser en particulier aux possibilités présentées par le circuit SN74123.

2 - Compteurs Asynchrones

Matériels fournis

7400, 7404, 7408, 7410, 7420, 2×7473 , 7486, 7493, 74123, 74155, composants R et C

2.1 Principe

Considérons un compteur qui compte en binaire pur sur trois bits. Ce compteur doit être constitué de trois cellules dont les sorties Q_0 , Q_1 et Q_2 correspondent respectivement au bit de rang 2^0 , à celui de rang 2^1 et à celui de rang 2^2 . Le Tableau 2-1 donne la succession des états binaires observés. On constate qu'un étage commute lorsque le précédent passe de 1 à 0.

décimal	Q_2	Q_1	Q_0
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1
0	0	0	0

Tableau 2-1: Principe du compteur asynchrone.

Nous avons vu que la fonction commutation était assurée par la bascule JK maître/esclave dans les conditions $J=K=1$.

Il suffit donc d'appliquer l'impulsion d'horloge sur le premier étage correspondant au rang 2^0 du compteur, et de relier ensuite la sortie Q_{n-1} du $(n-1)^{\text{ème}}$ étage à l'horloge du $n^{\text{ème}}$ étage, de telle sorte que la transition $1 \rightarrow 0$ de Q_{n-1} fasse commuter Q_n .

2.2 Compteur binaire pur à 4 étages

Le schéma ci-dessous représente un compteur asynchrone à 4 étages.

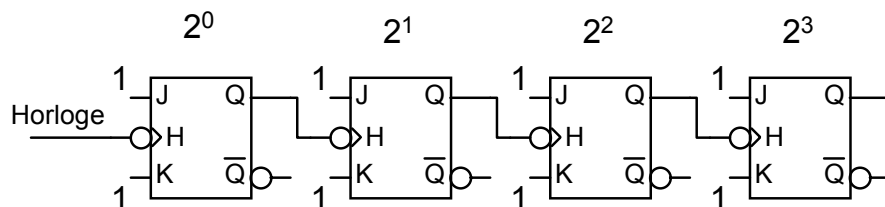


Figure 2-1: Compteur binaire pur 4 étages.

Examinons le diagramme des temps de ce compteur. Appelons θ le temps de basculement d'une cellule. Les commutations se produisant en cascade, les retards s'accroissent pour atteindre la valeur 4θ pour la quatrième cellule comme le montre les chronogrammes de la Figure 2-2.

On constate que le compteur asynchrone présente des états transitoires au moment des commutations. Par exemple, pour aller de la valeur 001 à la valeur 010, le compteur passe pendant un temps θ par la valeur 000. Ces états transitoires sont gênants si l'on veut exploiter le contenu du compteur pendant sa progression. Il est clair par exemple, que pour un compteur à 4 étages, aucun prélèvement de l'état du compteur ne sera possible si la période T de l'horloge n'est pas supérieure à 4θ .

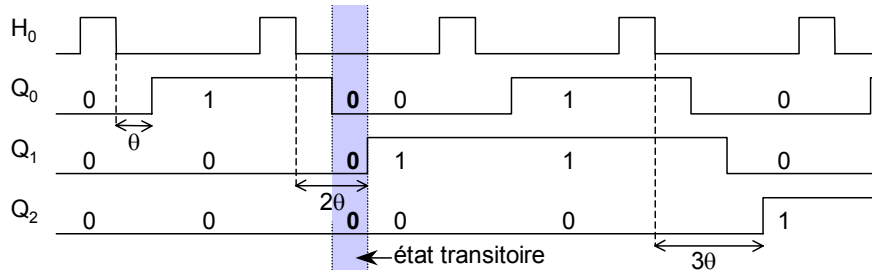


Figure 2-2: Chronogrammes du compteur binaire pur à 4 étages.

Travail demandé

- Construire un compteur asynchrone à 4 étages, à l'aide de 2 boîtiers SN 7473. Chaque sortie Q_i sera connectée à une LED témoin afin de visualiser les états successifs de chaque étage du compteur. Les impulsions d'horloge seront obtenues à partir d'un générateur manuel (interrupteur). En observant la LED de la sortie Q_3 , on mettra en évidence la fonction diviseur de fréquence du compteur.
- En utilisant le démultiplexeur SN74LS155, effectuer le décodage des états des trois premiers étages du compteur.
- Modifier ensuite le compteur en connectant les entrées d'horloge H_i ($i \geq 1$) de chaque étage à la sortie $\overline{Q_{i-1}}$ du précédent. Observer et expliquer le fonctionnement du compteur.

2.3 Compteur asynchrone à arrêt automatique à module variable

Nous venons de voir un compteur binaire pur à 4 étages qui revient à 0 au bout de 16 impulsions d'horloge. Dans certains cas, on peut avoir besoin d'un compteur qui s'arrête après un nombre n d'impulsions et attendre un signal extérieur de remise à zéro. Pour ce faire, il faut adjoindre au compteur un circuit qui détectera le nombre n et qui provoquera alors l'arrêt du compteur. Pour bloquer le compteur, il suffit de mettre au niveau 0 les entrées J_0 et K_0 de la première bascule. Dès lors Q_0 ne sera plus modifié lors des coups d'horloge suivants, et comme Q_0 fournit le signal d'horloge H_1 de la bascule suivante, cette dernière ne changera plus d'état non plus, et ainsi de suite pour les bascules successives.

Le circuit utilisé est représenté à la Figure 2-3. Il s'agit (pour un compteur à 4 étages) d'un NAND à 4 entrées, dont la sortie est reliée à J_0 et K_0 . Il est clair que lorsque les 4 entrées de ce NAND seront au niveau 1, J_0 et K_0 seront au niveau 0 et le compteur sera bloqué.

Si l'on veut, par exemple, que le compteur s'arrête au nombre 1100, il suffit de porter respectivement les points I_0 , I_1 , I_2 et I_3 à des interrupteurs et il suffit d'écrire avec ces interrupteurs le nombre qui doit provoquer l'arrêt du compteur.

Travail demandé

- Construire le compteur représenté à la Figure 2-3 et tester son fonctionnement.

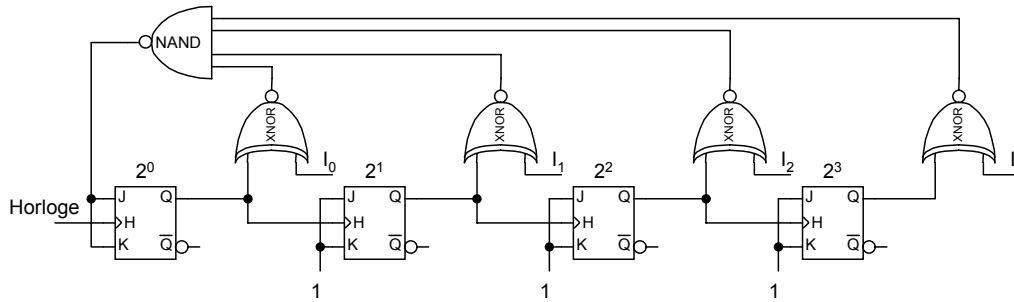


Figure 2-3: Compteur asynchrone à arrêt automatique à module variable.

2.4 Compteur asynchrone DCB 8421

Le code DCB 8421 suit l'ordre des nombres binaires jusqu'au nombre 1001 puis retourne à 0. La séquence des états du compteur est représentée dans le tableau ci-après. De tels compteurs sont appelés compteurs à cycle incomplet ou raccourci.

Q ₃	Q ₂	Q ₁	Q ₀	état
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
0	0	0	0	0

Tableau 2-2: Représentation des états du compteur asynchrone DCB 8421.

Pour réaliser ce compteur, on peut songer à détecter le nombre 1001 et à activer alors la commande *clear* de chacune des bascules ce qui aurait pour effet de ramener le compteur dans l'état 0000.

Cette méthode, à priori séduisante, est en fait déconseillée car peu fiable. D'ailleurs, voici les inconvénients d'une telle utilisation:

- Si le compteur à n étages, et si la période T de l'horloge est inférieure à $n\theta$, pas de prélèvement possible de l'état du compteur, donc impossibilité de détecter un nombre.
- Le commande *clear* est activée par la sortie d'un NAND à x entrées, si le nombre qui doit provoquer l'arrêt du compteur a x bits. Prenons pour origine des temps l'instant où toutes les entrées du NAND à x entrées sont au niveau 1 comme le présente la Figure 2-4.

A l'instant $t_{pHL}(P)$ la sortie du NAND passe de 1 à 0. Si l'on appelle $t_{pHL}(Q)$ le temps de basculement de 1 à 0, sous l'action de la commande *clear*, de la bascule la plus rapide, au temps $t_{pHP}(P) + t_{pHL}(Q)$ les x entrées ne sont plus toutes à 1. Donc au temps $t_{pHL}(P) +$

$t_{pHL}(Q) + t_{pLH}(Q)$ la sortie du NAND est au niveau 1 et la commande *clear* n'est plus activée.

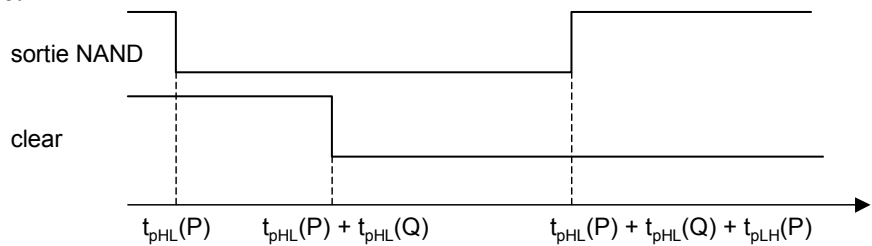


Figure 2-4: Utilisation de la commande *clear*.

Il faut que la durée du signal de la commande *clear* soit supérieure à la durée minimum exigée par la bascule la plus lente. Or des différences importantes peuvent exister à cause des dissymétries (charge capacitive) ou du vieillissement.

- D'autres part, si le compteur utilise un code qui comprend un saut (par exemple dans les 2421, on saute de 0100 à 1011), les commandes *clear* ne sont d'aucune utilité.

Il faut donc utiliser une autre méthode.

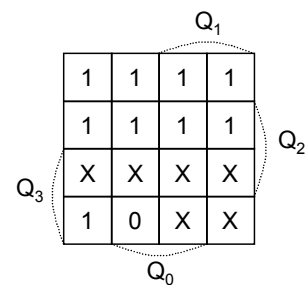
On constate que la sortie Q_0 commute à chaque état du compteur. Il faut donc que, chaque fois que le compteur passe d'un état au suivant, le signal reçu sur l'entrée H_1 passe de 1 à 0. On est dès lors tenté d'utiliser Q_0 pour commander l'entrée H_1 . Mais si l'on réalisait directement cette connexion, Q_1 subirait une cinquième commutation (case grisée) qui est indésirable. Par conséquent, on ne peut faire $H_1=Q_0$.

Par contre, si l'on fabrique une fonction f qui est égale à 1 pour les 9 premiers états (de 0 à 8), et égale à 0 pour l'état 9, on réussit à inhiber la commutation indésirable et $H_1=f.Q_0$ convient parfaitement.

Pour trouver f , il suffit d'écrire sa définition:

$$f = \bar{Q}_0 + \bar{Q}_3$$

$$\text{donc } H_1 = (\bar{Q}_0 + \bar{Q}_3)Q_0 = Q_0.\bar{Q}_3$$



Travail demandé

- De la même façon, montrer que
 - $H_2 = Q_1$
 - $H_3 = Q_2 + Q_0.Q_3.H$ ou encore $H_3 = \overline{\overline{Q_2.Q_0.Q_3.H}}$
- Dessiner le schéma de ce compteur.
- Réaliser ce compteur et tester son fonctionnement.

2.5 Compteur asynchrone DCB 2421

Le code 2421 est constitué par les cinq premières et les cinq dernières combinaisons du binaire pur à 4 bits.

Travail demandé

- En procédant comme précédemment, montrer que l'on doit avoir:

$$H_0 = H$$

$$H_1 = Q_0 + Q_2 \cdot \overline{Q_3} \cdot H$$

$$H_2 = Q_1 + Q_2 \cdot \overline{Q_3} \cdot H$$

$$H_3 = Q_2$$

- b) Dessiner le schéma de ce compteur et tester son fonctionnement.
- c) Vous constatez que l'on observe pas la succession des états prévus. Expliquer pourquoi et résoudre le problème en utilisant un monostable SN74123.

Vous pouvez vous aider en complétant la Figure 2-5.

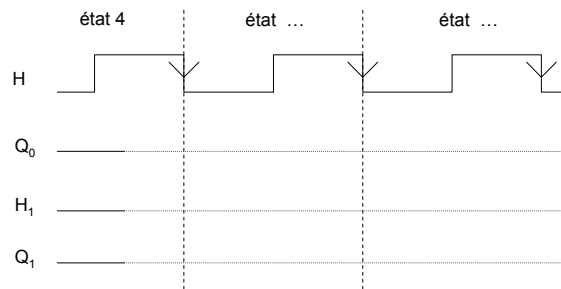


Figure 2-5 : Chronogrammes pour la résolution du problème.

2.6 Etude du circuit SN7493

L'objectif est d'étudier un circuit à partir d'une documentation fournie en annexe.

Travail demandé

- a) Etudier les possibilités de ce circuit,
- b) Câbler le circuit et vérifier les possibilités précédemment analysées.

3 - Compteurs Synchrones

Matériels fournis

2×7404 , 7408, 7410, 7420, 7432, 2×7473 , 7474, 7486, 74123, 74191, R, C

3.1 Principe

Le Tableau 3-1 donne la succession des états binaires pour un compteur à 3 étages.

Dans la manipulation précédente, nous avons remarqué qu'une bascule commutait chaque fois que la sortie de la bascule précédente passait de 1 à 0. Ceci nous avait conduit aux compteurs asynchrones.

Q_2	Q_1	Q_0	état
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7
0	0	0	0

Tableau 3-1: Principe du compteur synchrone.

On peut faire une autre analyse du Tableau 3-1. En effet, on remarque qu'une bascule commute chaque fois que toutes les sorties des bascules précédentes sont au niveau 1. Cette remarque nous conduit:

- à appliquer le même signal d'horloge sur tous les étages du compteur,
- à appliquer aux entrées $J_n=K_n$ de l'étage n un niveau qui sera égal à 1 si toutes les sorties Q_i ($0 < i < n-1$) sont au niveau 1, et égal à 0 dans le cas contraire.

3.2 Compteur synchrone binaire pur

Le schéma (Figure 3-1) représente un compteur synchrone binaire pur à 4 étages.

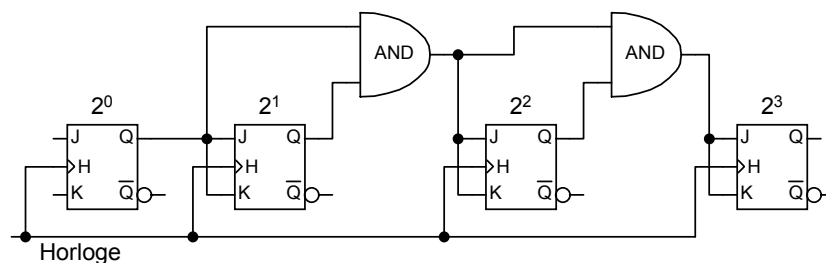


Figure 3-1: Compteur synchrone binaire pur.

Le diagramme des temps de ce compteur est représenté Figure 3-2.

Apparemment le compteur synchrone ne connaît pas d'états transitoires puisque tous les étages ont leurs horloges activées simultanément. En fait, ces états transitoires existent quand

même (θ n'est pas rigoureusement identique pour toutes les bascules), mais leur durée est beaucoup plus brève que dans un compteur asynchrone et le prélèvement des états du compteur est plus simple.

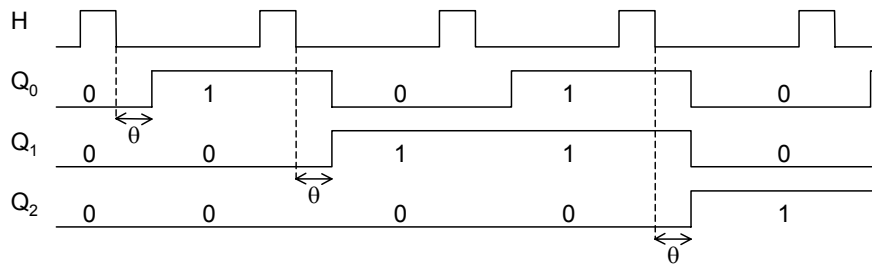


Figure 3-2: Chronogrammes du compteur synchrone binaire pur.

Travail demandé

- Construire un compteur synchrone binaire pur à 4 étages et tester son fonctionnement.
- Concevoir et construire un décompteur synchrone binaire pur à 4 étages. Tester son fonctionnement.

3.3 Compteur synchrone à recyclage

On demande souvent à un compteur de compter jusqu'à un certain nombre puis de retourner à zéro et de recommencer. Considérons par exemple un compteur à recyclage modulo 6 dans le code 8421. La succession des états du compteur est représentée dans le Tableau 3-2.

La synthèse des états du compteur peut s'effectuer par 2 méthodes: l'une utilise la table d'excitation, l'autre est basée sur l'identification. Nous allons illustrer ici la méthode utilisant la table d'excitation.

Q ₂	Q ₁	Q ₀	état
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
0	0	0	0

Tableau 3-2: Etats du compteur synchrone à recyclage.

La succession des états de la bascule 2^0 conduit à la table de Karnaugh ci-dessous, où F_0 , F_1 , M_0 et M_1 désignent respectivement le forçage à 0, le forçage à 1, le maintien à 0 et le maintien à 1. De plus, on rappelle la table d'excitation d'une bascule JK.

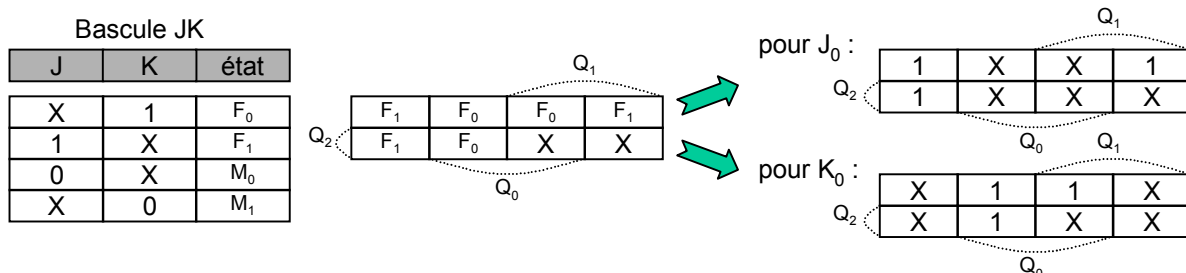


Figure 3-3: Tableaux de Karnaugh pour la bascule 2^0 .

On en déduit bien sûr : $J_0 = K_0 = 1$.

Pour la bascule 2^1 , on a une succession d'états qui conduit à la table de Karnaugh. On obtient le résultat suivant:

$$J_1 = Q_0 \cdot \overline{Q_2}$$

$$K_1 = Q_0 \cdot \overline{Q_2}$$

	Q ₁			
	M ₀	F ₁	F ₀	M ₁
Q ₂	M ₀	M ₀	X	X
	Q ₀			

pour J₁ :

	Q ₁			
	0	1	X	X
Q ₂	0	0	X	X
	Q ₀			

pour K₁ :

	Q ₁			
	X	X	1	0
Q ₂	X	X	X	X
	Q ₀			

Figure 3-4: : Tableaux de Karnaugh pour la bascule 2^1 .

Travail demandé

- On montrera de même que $J_2 = Q_0 \cdot Q_1$ et $K_2 = Q_0$.
- Dessiner le schéma de ce compteur. Le construire et tester son fonctionnement.

3.4 Compteur synchrone à excédent 3

Un nombre à excédent 3 est formé en ajoutant 3 à la représentation binaire du nombre à coder. La succession des états du compteur est représentée dans le Tableau 3-3.

Q ₃	Q ₂	Q ₁	Q ₀	état
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	10
1	0	1	1	11
1	1	0	0	12

Tableau 3-3: Etats du compteur synchrone à excédent 3.

Sur cet exemple, nous allons illustrer la méthode d'identification qui exprime le nouvel état du compteur par:

$$Q^+ = J \cdot \overline{Q} + \overline{K} \cdot Q.$$

Pour le bit 2^2 on a ici la table de Q_2^+ présentée ci-contre. Le résultat est donc le suivant:

$$Q_2^+ = Q_2 (\overline{Q_0} \cdot \overline{Q_3} + Q_0 \cdot \overline{Q_1}) + \overline{Q_2} \cdot Q_0 \cdot Q_1$$

d'où l'on tire:

$$J_2 = Q_0 \cdot Q_1 \text{ et } K_2 = \overline{\overline{Q_0} \cdot \overline{Q_3} + Q_0 \cdot \overline{Q_1}}$$

	Q ₁			
	X	X	1	X
	1	1	0	1
Q ₃	0	X	X	X
	0	0	1	0
	Q ₀			

Figure 3-5: Tableau de Karnaugh du bit 2^2 .

Travail demandé

- Montrer que l'on obtient de même:

 $J_0=1$ et $K_0=1$,

 $J_1=Q_0+Q_2.Q_3$ et $K_1=Q_0$,

 $J_3=Q_0.Q_1.Q_2$ et $K_3=Q_2$.
- En utilisant la table d'excitation, montrer que l'on obtient le résultat plus simple:

 $J_2 = K_2 = Q_0.Q_1+Q_2.Q_3$.

 La méthode d'identification est en effet plus rapide mais ne permet pas autant de simplifications que l'utilisation de la table d'excitation.
- Dessiner le schéma de ce compteur. Le construire et tester son fonctionnement.

3.5 Compteur synchrone à recyclage à module variable

Ce compteur doit compter jusqu'à un nombre maximum (pré-affiché par des interrupteurs) puis retourner à 0.

Il faut donc d'abord détecter le nombre maximum et provoquer le blocage du compteur. Le circuit de détection est identique à celui déjà vu pour les compteurs asynchrones. C'est un *Nand* à 4 entrées. Lorsque ses 4 entrées sont au niveau 1, la sortie du *Nand* est au niveau 0, ce qui a pour effet de mettre toutes les entrées des bascules au niveau 0. Le compteur est alors bloqué. Il faut alors faire passer le compteur dans l'état 0. Pour cela il suffit d'appliquer un niveau 1 sur toutes les entrées K.

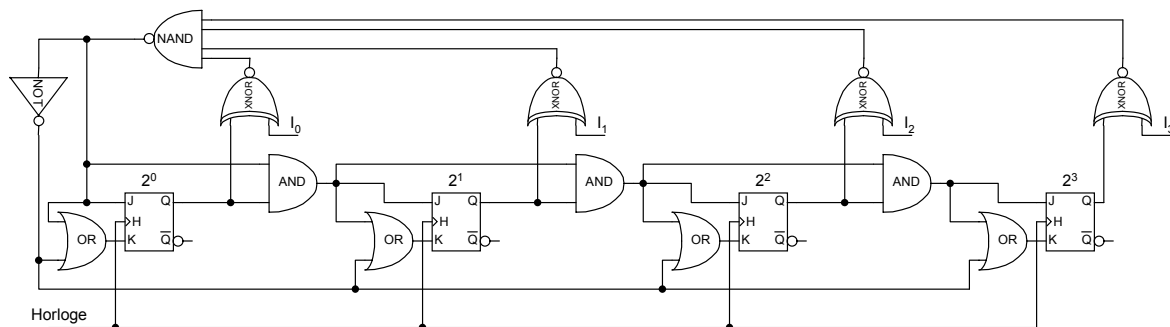


Figure 3-6: Schéma du compteur synchrone à recyclage à module variable.

Travail demandé

- Construire ce compteur et tester son fonctionnement.

3.6 Le circuit SN74191

L'objectif est de mettre en œuvre le circuit SN74191 dans les conditions présentées ci-après.

Travail demandé

- En utilisant le commande *Load* activée par un circuit SN7420, et les entrées *Data A, B, C, D* on peut faire fonctionner le compteur entre deux nombres quelconques $a_3a_2a_1a_0$ et $b_3b_2b_1b_0$. Expliquer le principe de ce fonctionnement.

- b) On veut faire fonctionner le compteur entre 3 et 12. Dessiner le chronogramme à partir de l'état 11 dans les 2 cas:
- Lorsque la sortie du circuit SN7420 est reliée directement à la commande *Load*,
 - Lorsqu'une bascule D est intercalée entre le 7420 et la commande *Load*. Vous constatez alors qu'une fois que l'état 3 est apparu, le coup d'horloge suivant est inhibé. Expliquez pourquoi.
 - Comment remédier à ceci en utilisant un circuit 74123 ?

4 - Registres à décalage

Matériels fournis

4 × 7400, 7408, 7404, 2 × 7410, 7420, 7432, 2 × 7473, 7486, 74194, 74195, Génér. TTL

4.1 Etude de circuits

Lorsque n cellules sont câblées en cascade, la fonction de décalage est celle qui permet, à chaque impulsion d'horloge, le glissement de l'information d'une cellule à la suivante.

En général, on dispose (ou éventuellement on construit) de(s) registres dans lesquels on peut effectuer des décalages de la gauche vers la droite ou de la droite vers la gauche, et dans lesquels on peut, de plus, charger des données en parallèle dans les différentes cellules.

Travail demandé

- Etudier dans le détail les circuits SN74194 et SN74195,
- Expliquer et tester leur fonctionnement. Retrouver en particulier les différentes fonctions décrites ci-dessus.

4.2 Diviseur de fréquence

Soit un compteur synchrone dont la succession des états est donnée dans Tableau 4-1. Il est possible de réaliser ce compteur à partir du circuit SN74195, pourvu que l'on remarque que ce registre est construit de telle façon que pour la bascule de rang n on ait $J_n = \overline{K_n}$. De plus $J_n = Q_{n-1}$ sauf pour la première bascule.

Travail demandé

- Faire la synthèse de ce compteur de 2 façons:
 - d'une part, en utilisant la table d'excitation de la bascule,
 - d'autre part, en effectuant une identification (ne pas faire pour J_a et $\overline{K_a}$)

Un exemple de chaque type de synthèse est donné dans la manipulation du *compteurs synchrones*.

Remarque: pour N=16, il faut effectuer un chargement parallèle.

Montrer alors que l'une des solutions à la synthèse de ce compteur est:

$$J_a = \overline{K_a} = \overline{Q_c} \oplus Q_d + Q_a \cdot Q_b \cdot Q_c \cdot Q_d; J_b = \overline{K_b} = Q_a, J_c = \overline{K_c} = Q_b, J_d = \overline{K_d} = Q_c$$

N	états			
	A	B	C	D
16	1	1	1	1
15	0	1	1	1
14	1	0	1	1
13	1	1	0	1
12	0	1	1	0
11	0	0	1	1
10	1	0	0	1
9	0	1	0	0
8	1	0	1	0
7	0	1	0	1
6	0	0	1	0
5	0	0	0	1
4	0	0	0	0
3	1	0	0	0
2	1	1	0	0
1	1	1	1	0

Tableau 4-1: Table de transition.

Cette solution a l'avantage d'utiliser le circuit SN7195.

- b) La Figure 4-1 montre la réalisation d'un tel compteur utilisé en diviseur de fréquence. C'est à dire que si l'on applique sur l'entrée des données (DATA) un nombre de 4 bits, la fréquence du signal de sortie est égale à la fréquence de l'horloge divisée par N, N étant la valeur qui correspond au nombre de 4 bits introduit.

Expliquer comment fonctionne ce diviseur de fréquence.

- c) Réaliser ce diviseur de fréquence et tester son fonctionnement.

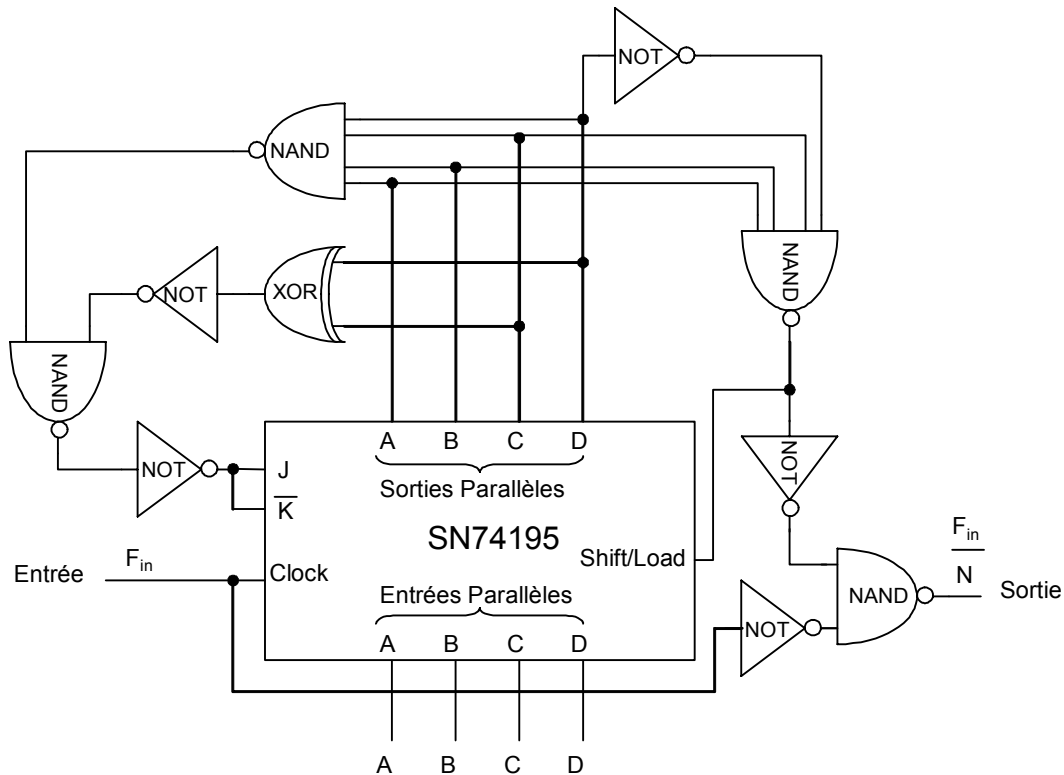


Figure 4-1: Diviseur de fréquence.

4.3 Réalisation d'un registre à décalage

Le but que l'on se fixe maintenant est de construire, à l'aide de boîtier SN7473, un registre 4 bits qui remplit les fonctions suivantes:

- décalage à droite,
- décalage à gauche,
- chargement parallèle.

Le choix de 3 fonctions implique l'utilisation de 2 entrées de sélection S_1 et S_0 qui seront reliées à des interrupteurs. On pourra convenir, par exemple, que les 3 fonctions précédentes seront codées comme dans le Tableau 4-2.

$\overline{S_1}$	$\overline{S_0}$	Fonction
0	0	décalage à droite
0	1	décalage à gauche
1	0	chargement parallèle

Tableau 4-2: Fonctions du registre à décalage.

Travail demandé

- a) Ecrire les fonctions logiques que l'on doit appliquer respectivement sur les entrées J_i et K_i de la bascule de rang i .
- b) A quel fonctionnement correspond la combinaison $\overline{S_0} = \overline{S_1} = 1$?

Pour être complet, le registre dans sa fonction de décalage, doit pouvoir soit fonctionner en anneau, soit rentrer en série 0 ou 1 (fonction *serial input* dans les registres intégrés). Ceci conduit à introduire une nouvelle entrée de sélection $\overline{S_2}$ qui sera reliée à un interrupteur. On pourra convenir que $\overline{S_2} = 0$ correspond à un fonctionnement en anneau, alors que $\overline{S_2} = 1$ permet d'introduire des 0 ou 1 suivant la position d'un interrupteur.

- c) Ecrire les nouvelles fonctions logiques J_1 et J_4 qu'il faut appliquer sur les entrées respectives de la première et de la dernière bascule.
- d) Réaliser ce registre et tester son fonctionnement en visualisant les sorties Q_i de chacune des bascules.

Remarque: Dans vos équations, des termes se répètent. Il est très recommandé de les détecter pour le câblage du schéma. Ceci vous permettra de diminuer sensiblement le nombre de circuits, et surtout la quantité de liaisons, donc de fils.

Annexes

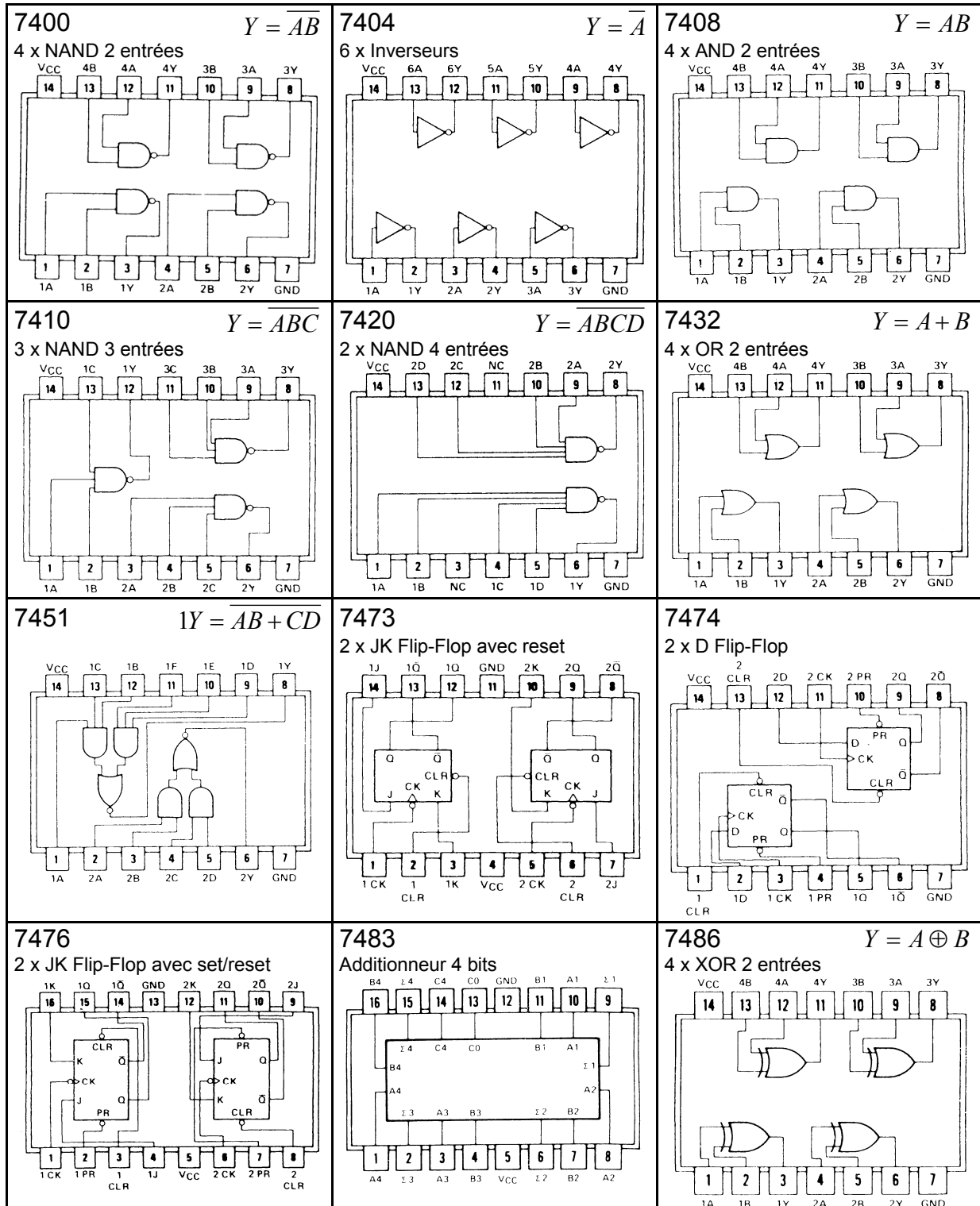
~~~~~

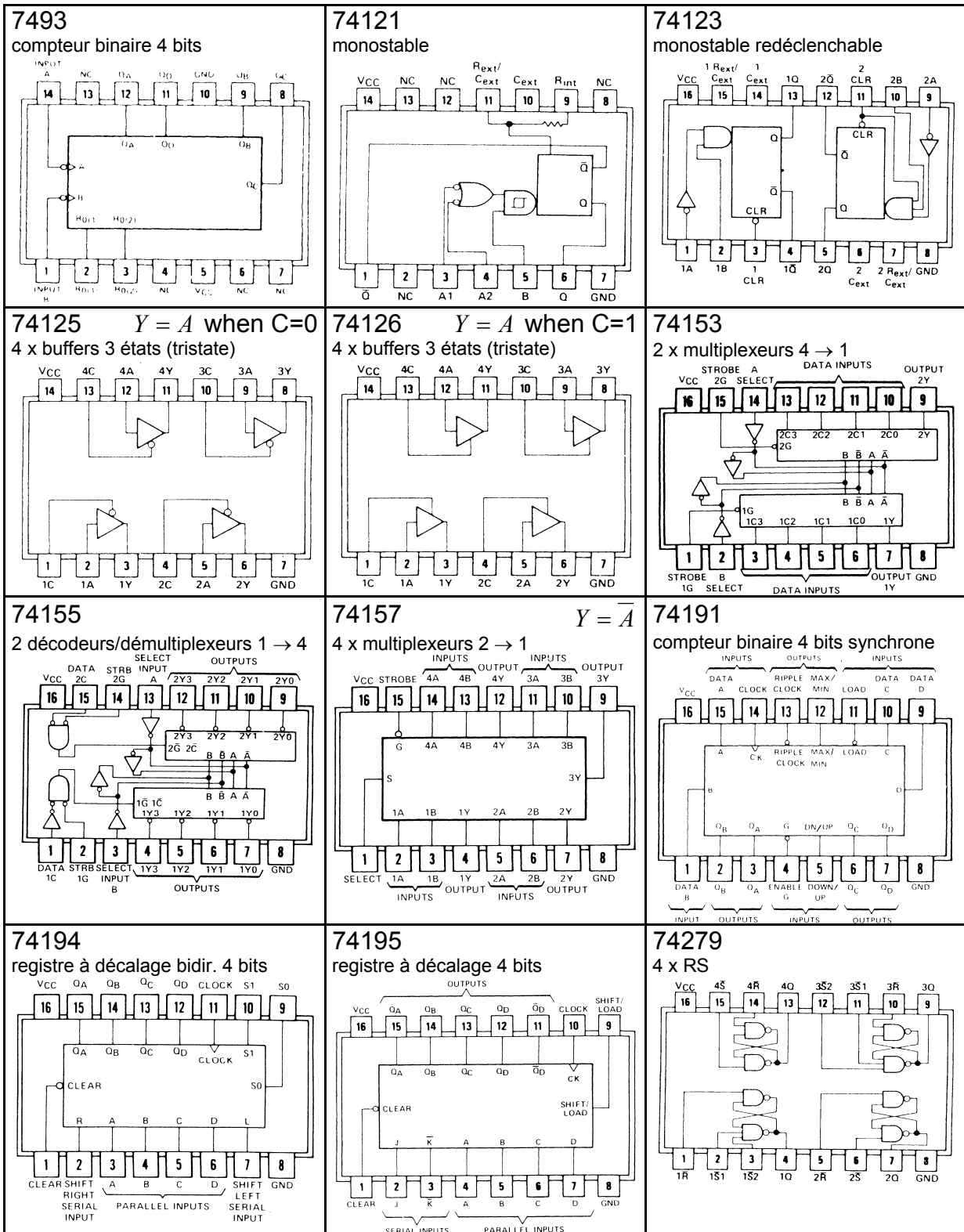
|                                                        |           |
|--------------------------------------------------------|-----------|
| <b>Annexe I - Brochage Circuits Intégrés TTL .....</b> | <b>33</b> |
| <b>Annexe II - Compléments Circuits Intégrés .....</b> | <b>35</b> |
| II.1 Circuit 7493 .....                                | 35        |
| II.2 Circuit 74121 .....                               | 36        |
| II.3 Circuit 74123 .....                               | 38        |
| II.4 Circuit 74155 .....                               | 39        |
| II.5 Circuit 74191 .....                               | 40        |
| II.6 Circuit 74194 .....                               | 42        |
| II.7 Circuit 74195 .....                               | 44        |





## Annexe I - Brochage Circuits Intégrés TTL





## Annexe II - Compléments Circuits Intégrés

### II.1 Circuit 7493

'90A, 'L90 . . . DECADE COUNTERS

'92A . . . DIVIDE-BY-TWELVE  
COUNTER

'93A, 'L93 . . . 4-BIT BINARY  
COUNTERS

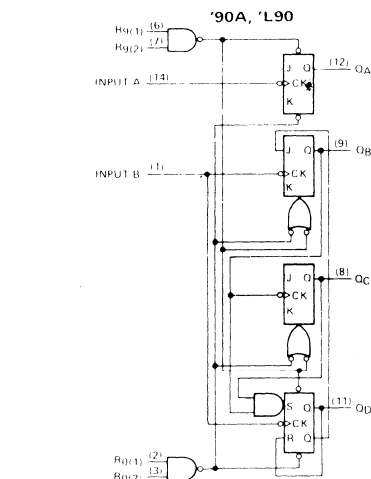
#### description

Each of these monolithic counters contains four master-slave flip-flops and additional gating to provide a divide-by-two counter and a three-stage binary counter for which the count cycle length is divide-by-five for the '90A and 'L90, divide-by-six for the '92A, and divide-by-eight for the '93A and 'L93.

All of these counters have a gated zero reset and the '90A and 'L90 also have gated set-to-nine inputs for use in BCD nine's complement applications.

To use their maximum count length (decade, divide-by-twelve, or four-bit binary) of these counters, the B input is connected to the  $Q_A$  output. The input count pulses are applied to input A and the outputs are as described in the appropriate function table. A symmetrical divide-by-ten count can be obtained from the '90A or 'L90 counters by connecting the  $Q_D$  output to the A input and applying the input count to the B input which gives a divide-by-ten square wave at output  $Q_A$ .

#### functional block diagrams

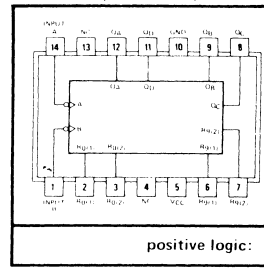


dynamic input activated by transition from a high level to a low level.

The J and K inputs shown without connection are for reference only and are functionally at a high level.

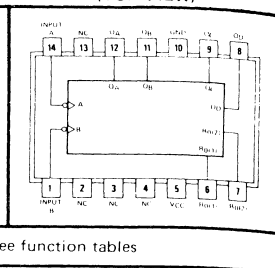
'90A . . . J, N, OR W PACKAGE

'L90 . . . J, N, OR T PACKAGE  
(TOP VIEW)



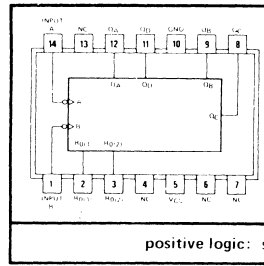
positive logic: see function tables

'92A . . . J, N, OR W PACKAGE  
(TOP VIEW)



'93A . . . J, N, OR W PACKAGE  
(TOP VIEW)

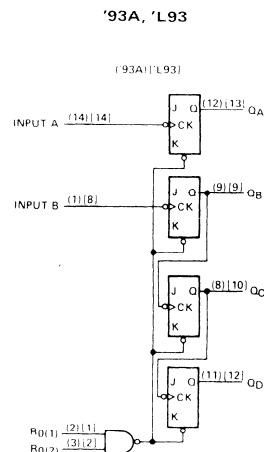
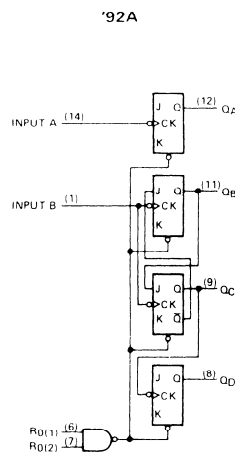
'L93 . . . J, N, OR T PACKAGE  
(TOP VIEW)



positive logic: see function tables

NC - No internal connection

| TYPES      | TYPICAL<br>POWER DISSIPATION |
|------------|------------------------------|
| '90A       | 145 mW                       |
| 'L90       | 20 mW                        |
| '92A, '93A | 130 mW                       |
| 'L93       | 16 mW                        |





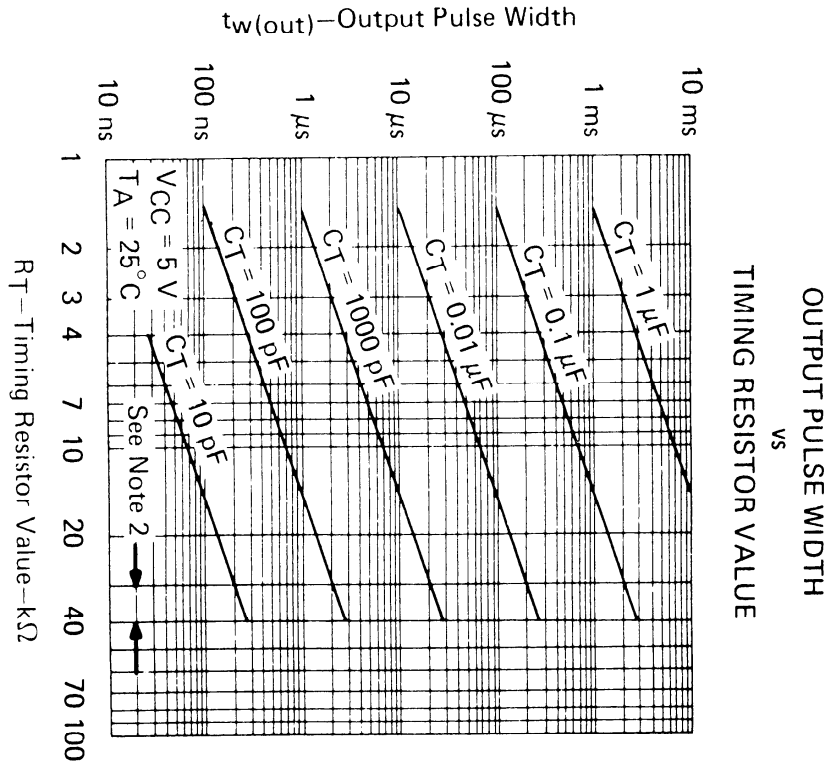


FIGURE 6

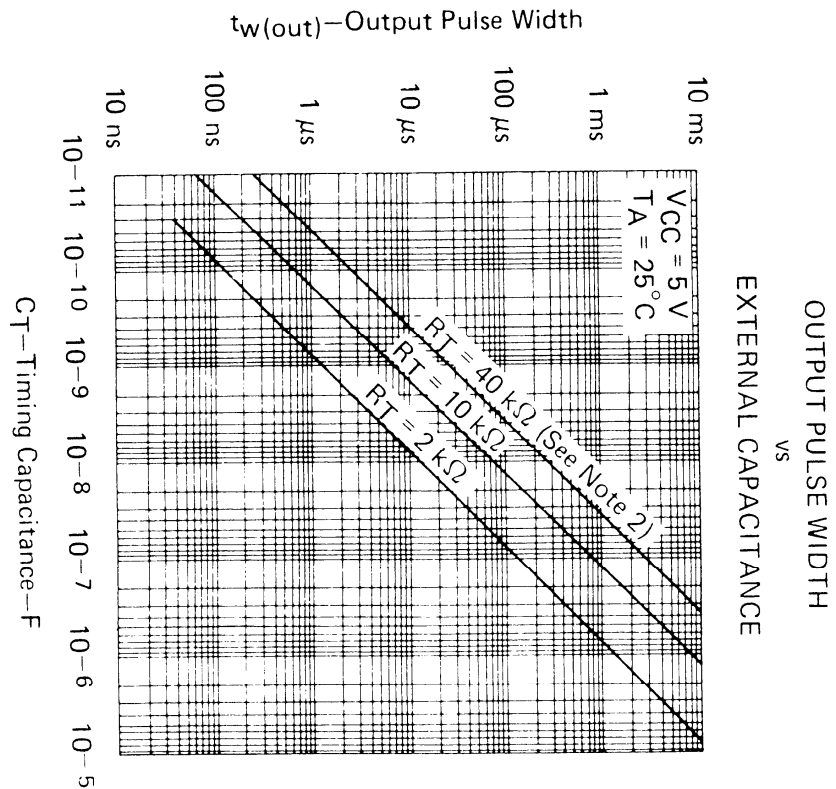


FIGURE 7

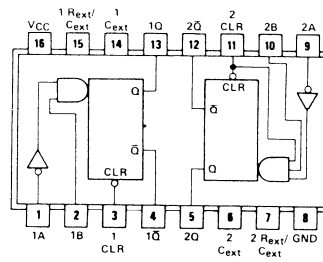
§ Data for temperatures below  $0^\circ \text{ C}$  and above  $70^\circ \text{ C}$  are applicable for SN54121 and SN54L121 only.  
 NOTE 2: These values of resistance exceed the maximum recommended for use over the full temperature range of the SN54121 and SN54L121.

## II.3 Circuit 74123

**FUNCTION TABLE**

| INPUTS |   |   | OUTPUTS |           |
|--------|---|---|---------|-----------|
| CLEAR  | A | B | Q       | $\bar{Q}$ |
| L      | X | X | L       | H         |
| X      | H | X | L       | H         |
| X      | X | L | L       | H         |
| H      | L | ↑ | ⌋       | ⌋         |
| H      | ↓ | H | ⌋       | ⌋         |
| ↑      | L | H | ⌋       | ⌋         |

See page 138



SN54123/SN74123(J, N, W)  
 SN54L123/SN74L123(J, N)

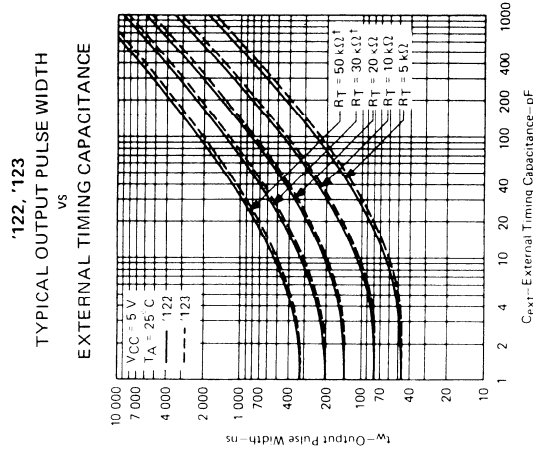


FIGURE B

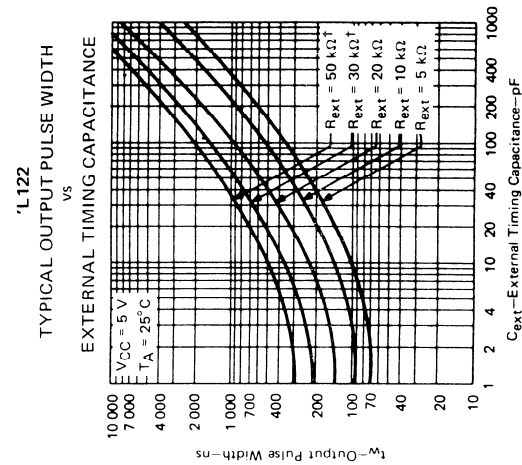
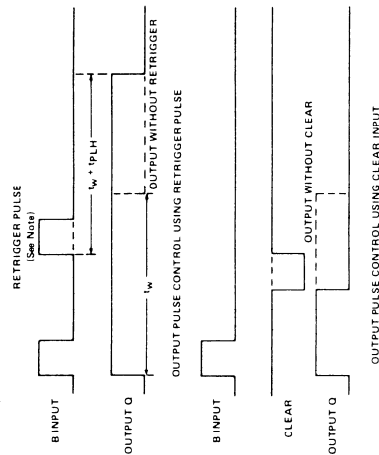


FIGURE C

† These values of resistance exceed the maximum recommended for use over the full temperature range of the SN54 and SN74L circuits.

### description

The '122, '123, 'L122, and 'L123 multivibrators feature d-c triggering from gated low-level-active (A) and high-level-active (B) inputs, and also provide overriding direct clear inputs. Complementary outputs are provided. The retrigger capability simplifies the generation of output pulses of extremely long duration. By triggering the input before the output pulse is terminated, the output pulse may be extended. The overriding clear capability permits any output pulse to be terminated at a predetermined time independently of the timing components R and C. Figure A below illustrates triggering the one-shot with the high-level-active (B) inputs.



NOTE: Retrigger pulse must not start before  $0.22 C_{ext}$  (in picofarads) nanoseconds after previous trigger pulse.

FIGURE A—TYPICAL INPUT/OUTPUT PULSES

These monostables are designed to provide the system designer with complete flexibility in controlling the pulse width, either to lengthen the pulse by retriggering, or to shorten by clearing. The '122 and 'L122 each has an internal timing resistor which allows the circuit to be operated with only an external capacitor, if so desired. Applications requiring more precise pulse widths (up to 28 seconds) and not requiring the clear feature can best be satisfied with '121 or 'L121.

The output pulse is primarily a function of the external capacitor and resistor. For  $C_{ext} > 1000 \text{ pF}$ , the output pulse width ( $t_w$ ) is defined as:

$$t_w = K \cdot RT \cdot C_{ext} \left( 1 + \frac{0.7}{RT} \right)$$

where

RT is in  $k\Omega$  (either internal or external timing resistor),

$C_{ext}$  is in pF,

$t_w$  is in ns,

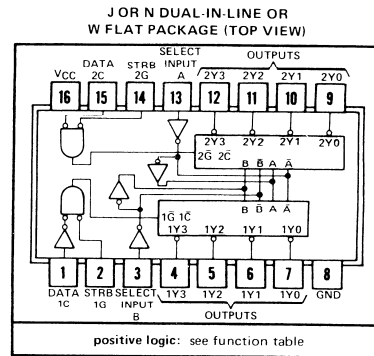
K is 0.32 for '122, 0.28 for '123, 0.37 for 'L122, 0.33 for 'L123.

For pulse widths when  $C_{ext} \leq 1000 \text{ pF}$ , see Figures B and C.

## II.4 Circuit 74155

- Applications:
  - Dual 2-to-4-Line Decoder
  - Dual 1-to-4-Line Demultiplexer
  - 3-to-8-Line Decoder
  - 1-to-8-Line Demultiplexer
- Individual Strobes Simplify Cascading for Decoding or Demultiplexing Larger Words
- Input Clamping Diodes Simplify System Design
- Choice of Outputs:
  - Totem Pole ('155, 'LS155)
  - Open-Collector ('156)

| TYPES      | TYPICAL AVERAGE PROPAGATION DELAY<br>3 GATE LEVELS | TYPICAL POWER DISSIPATION |
|------------|----------------------------------------------------|---------------------------|
| '155, '156 | 21 ns                                              | 125 mW                    |
| 'LS155     | 18 ns                                              | 31 mW                     |

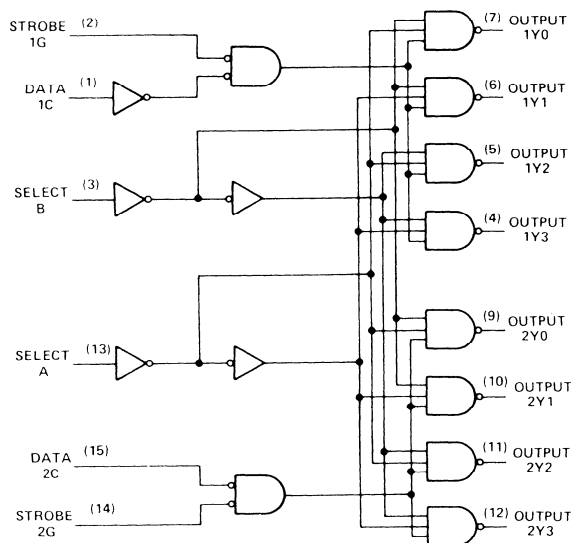


### description

These monolithic transistor-transistor-logic (TTL) circuits feature dual 1-line-to-4-line demultiplexers with individual strobes and common binary-address inputs in a single 16-pin package. When both sections are enabled by the strobes, the common binary-address inputs sequentially select and route associated input data to the appropriate output of each section. The individual strobes permit activating or inhibiting each of the 4-bit sections as desired. Data applied to input 1C is inverted at its outputs and data applied at 2C is not inverted through its outputs. The inverter following the 1C data input permits use as a 3-to-8-line decoder or 1-to-8-line demultiplexer without external gating. Input clamping diodes are provided on all of these circuits to minimize transmission-line effects and simplify system design.

Series 54 and 54LS are characterized for operation over the full military temperature range of  $-55^{\circ}\text{C}$  to  $125^{\circ}\text{C}$ ; Series 74 and 74LS are characterized for operation from  $0^{\circ}\text{C}$  to  $70^{\circ}\text{C}$ .

### functional block diagram and logic



**FUNCTION TABLES**  
**2-LINE-TO-4-LINE DECODER**  
**OR 1-LINE-TO-4-LINE DEMULTIPLEXER**

| INPUTS |        |      |    | OUTPUTS |     |     |     |
|--------|--------|------|----|---------|-----|-----|-----|
| SELECT | STROBE | DATA |    |         |     |     |     |
| B      | A      | 1G   | 1C | 1Y0     | 1Y1 | 1Y2 | 1Y3 |
| X      | X      | H    | X  | H       | H   | H   | H   |
| L      | L      | L    | H  | L       | H   | H   | H   |
| L      | H      | L    | H  | H       | L   | H   | H   |
| H      | L      | L    | H  | H       | H   | L   | H   |
| H      | H      | L    | H  | H       | H   | H   | L   |
| X      | X      | X    | L  | H       | H   | H   | H   |

| INPUTS |        |      |    | OUTPUTS |     |     |     |
|--------|--------|------|----|---------|-----|-----|-----|
| SELECT | STROBE | DATA |    |         |     |     |     |
| B      | A      | 2G   | 2C | 2Y0     | 2Y1 | 2Y2 | 2Y3 |
| X      | X      | H    | X  | H       | H   | H   | H   |
| L      | L      | L    | L  | L       | H   | H   | H   |
| L      | H      | L    | L  | H       | L   | H   | H   |
| H      | L      | L    | L  | H       | H   | L   | H   |
| H      | H      | L    | L  | H       | H   | H   | L   |
| X      | X      | X    | H  | H       | H   | H   | H   |

**FUNCTION TABLE**  
**3-LINE-TO-8-LINE DECODER**  
**OR 1-LINE-TO-8-LINE DEMULTIPLEXER**

| INPUTS |                |   |    | OUTPUTS |     |     |     |     |     |     |     |
|--------|----------------|---|----|---------|-----|-----|-----|-----|-----|-----|-----|
| SELECT | STROBE OR DATA |   |    |         |     |     |     |     |     |     |     |
| C†     | B              | A | G‡ | 2Y0     | 2Y1 | 2Y2 | 2Y3 | 1Y0 | 1Y1 | 1Y2 | 1Y3 |
| X      | X              | X | H  | H       | H   | H   | H   | H   | H   | H   | H   |
| L      | L              | L | L  | L       | H   | H   | H   | H   | H   | H   | H   |
| L      | L              | H | L  | H       | L   | H   | H   | H   | H   | H   | H   |
| L      | H              | L | L  | H       | H   | L   | H   | H   | H   | H   | H   |
| L      | H              | H | L  | H       | H   | H   | L   | H   | H   | H   | H   |
| H      | L              | L | L  | H       | H   | H   | H   | L   | H   | H   | H   |
| H      | L              | H | L  | H       | H   | H   | H   | H   | L   | H   | H   |
| H      | H              | L | L  | H       | H   | H   | H   | H   | H   | L   | H   |
| H      | H              | H | L  | H       | H   | H   | H   | H   | H   | H   | L   |

†C – inputs 1C and 2C connected together  
 ‡G – inputs 1G and 2G connected together  
 H = high level, L = low level, X = irrelevant



## II.5 Circuit 74191

- Counts 8-4-2-1 BCD or Binary
- Single Down/Up Count Control Line
- Count Enable Control Input
- Ripple Clock Output for Cascading
- Asynchronously Presetable with Load Control
- Parallel Outputs
- Cascadable for n-Bit Applications

| TYPE           | AVERAGE<br>PROPAGATION<br>DELAY | TYPICAL<br>MAXIMUM<br>CLOCK<br>FREQUENCY | TYPICAL<br>POWER<br>DISSIPATION |
|----------------|---------------------------------|------------------------------------------|---------------------------------|
| '190, '191     | 20 ns                           | 25 MHz                                   | 325 mW                          |
| 'LS190, 'LS191 | 20 ns                           | 25 MHz                                   | 90 mW                           |

### description

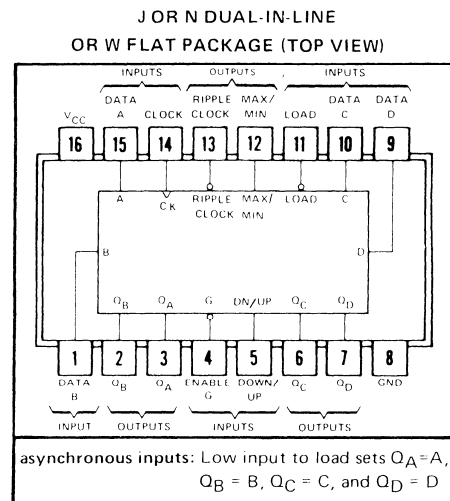
The '190, 'LS190, '191, and 'LS191 are synchronous, reversible up/down counters having a complexity of 58 equivalent gates. The '191 and 'LS191 are 4-bit binary counters and the '190 and 'LS190 are BCD counters. Synchronous operation is provided by having all flip-flops clocked simultaneously so that the outputs change coincident with each other when so instructed by the steering logic. This mode of operation eliminates the output counting spikes normally associated with asynchronous (ripple clock) counters.

The outputs of the four master-slave flip-flops are triggered on a low-to-high-level transition of the clock input if the enable input is low. A high at the enable input inhibits counting. Level changes at the enable input should be made only when the clock input is high. The direction of the count is determined by the level of the down/up input. When low, the counter counts up and when high, it counts down.

These counters are fully programmable; that is, the outputs may be preset to either level by placing a low on the load input and entering the desired data at the data inputs. The output will change to agree with the data inputs independently of the level of the clock input. This feature allows the counters to be used as modulo-N dividers by simply modifying the count length with the preset inputs.

The clock, down/up, and load inputs are buffered to lower the drive requirement which significantly reduces the number of clock drivers, etc., required for long parallel words.

Two outputs have been made available to perform the cascading function: ripple clock and maximum/minimum count. The latter output produces a high-level output pulse with a duration approximately equal to one complete cycle of the clock when the counter overflows or underflows. The ripple clock output produces a low-level output pulse equal in width to the low-level portion of the clock input when an overflow or underflow condition exists. The counters can be easily cascaded by feeding the ripple clock output to the enable input of the succeeding counter if parallel clocking is used, or to the clock input if parallel enabling is used. The maximum/minimum count output can be used to accomplish look-ahead for high-speed operation.

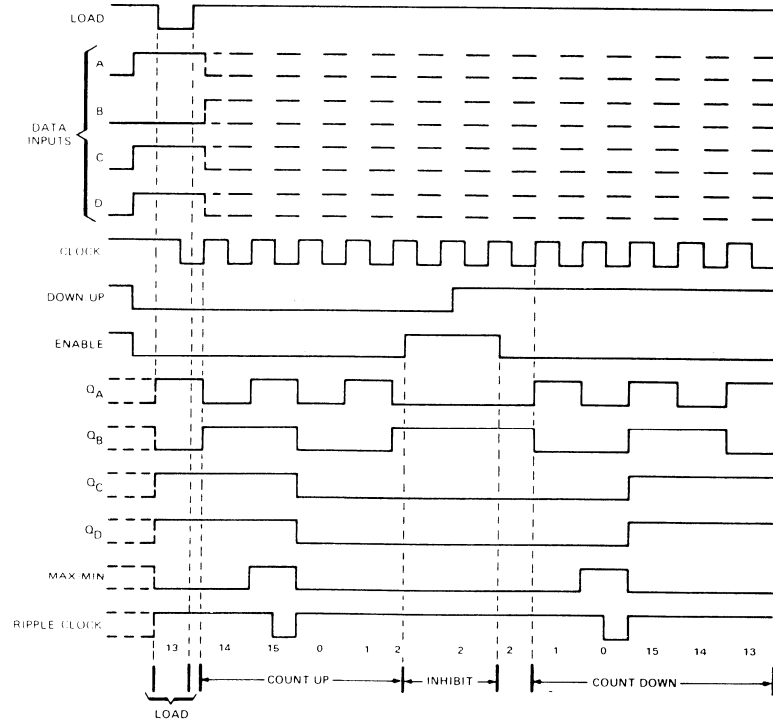




typical load, count, and inhibit sequences

Illustrated below is the following sequence:

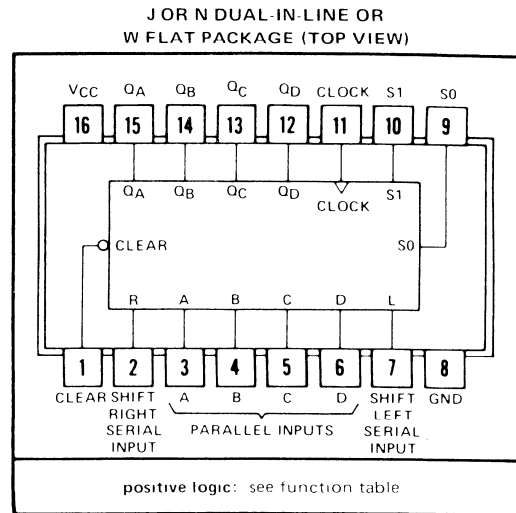
1. Load (preset) to binary thirteen.
2. Count up to fourteen, fifteen (maximum), zero, one, and two.
3. Inhibit.
4. Count down to one, zero (minimum), fifteen, fourteen, and thirteen.



## II.6 Circuit 74194

- Parallel Inputs and Outputs
- Four Operating Modes:
  - Synchronous Parallel Load
  - Right Shift
  - Left Shift
  - Do Nothing
- Positive Edge-Triggered Clocking
- Direct Overriding Clear

| TYPE   | TYPICAL<br>MAXIMUM<br>CLOCK<br>FREQUENCY | TYPICAL<br>POWER<br>DISSIPATION |
|--------|------------------------------------------|---------------------------------|
| '194   | 36 MHz                                   | 195 mW                          |
| 'LS194 | 28 MHz                                   | 60 mW                           |
| 'S194  | 105 MHz                                  | 425 mW                          |



### description

These bidirectional shift registers are designed to incorporate virtually all of the features a system designer may want in a shift register. The circuit contains 46 equivalent gates and features parallel inputs, parallel outputs, right-shift and left-shift serial inputs, operating-mode-control inputs, and a direct overriding clear line. The register has four distinct modes of operation, namely:

- Parallel (Broadside) Load
- Shift Right (In the direction  $Q_A$  toward  $Q_D$ )
- Shift Left (In the direction  $Q_D$  toward  $Q_A$ )
- Inhibit Clock (Do nothing)

Synchronous parallel loading is accomplished by applying the four bits of data and taking both mode control inputs,  $S_0$  and  $S_1$ , high. The data is loaded into the associated flip-flop and appears at the outputs after the positive transition of the clock input. During loading, serial data flow is inhibited.

Shift right is accomplished synchronously with the rising edge of the clock pulse when  $S_0$  is high and  $S_1$  is low. Serial data for this mode is entered at the shift-right data input. When  $S_0$  is low and  $S_1$  is high, data shifts left synchronously and new data is entered at the shift-left serial input.

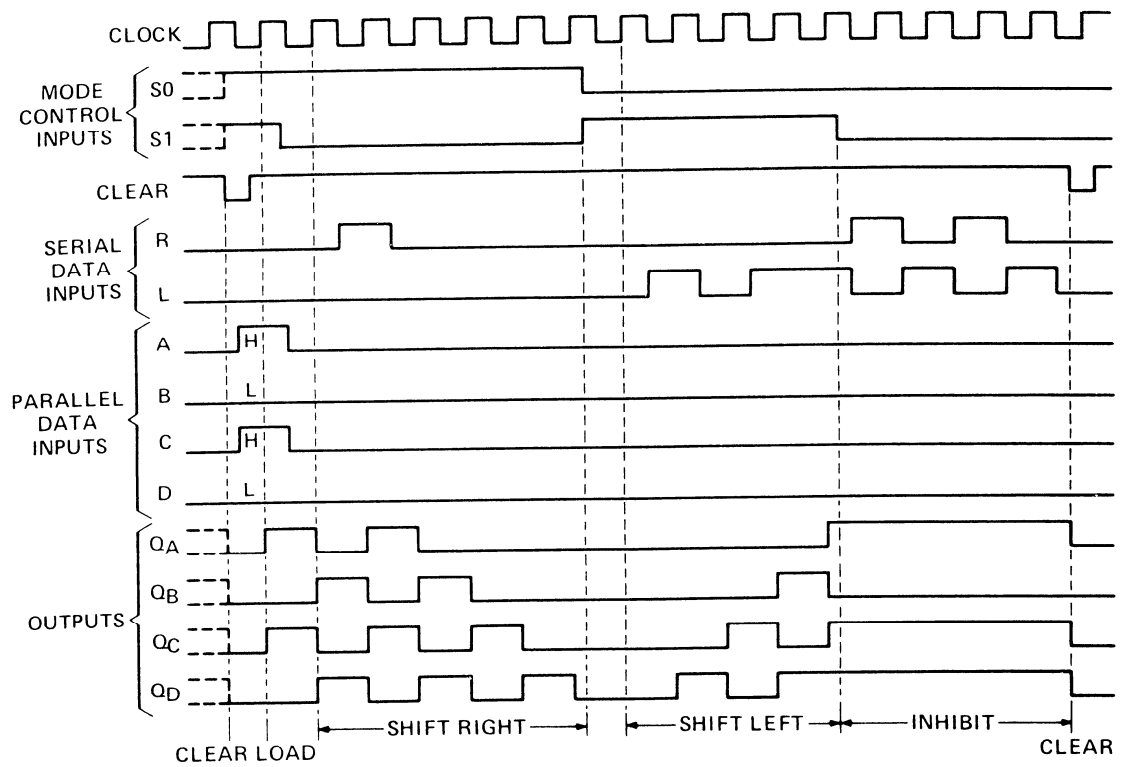
Clocking of the flip-flop is inhibited when both mode control inputs are low. The mode controls of the SN54194/SN74194 should be changed only while the clock input is high.

**FUNCTION TABLE**

| CLEAR | MODE  |       | CLOCK | INPUTS |       |          |   | OUTPUTS |       |          |          |          |          |          |          |
|-------|-------|-------|-------|--------|-------|----------|---|---------|-------|----------|----------|----------|----------|----------|----------|
|       | $S_1$ | $S_0$ |       | SERIAL |       | PARALLEL |   | $Q_A$   | $Q_B$ | $Q_C$    | $Q_D$    |          |          |          |          |
|       |       |       |       | LEFT   | RIGHT | A        | B |         |       |          |          | C        | D        |          |          |
| L     | X     | X     | X     | X      | X     | X        | X | X       | X     | X        | X        | L        | L        | L        | L        |
| H     | X     | X     | L     | X      | X     | X        | X | X       | X     | X        | X        | $Q_{A0}$ | $Q_{B0}$ | $Q_{C0}$ | $Q_{D0}$ |
| H     | H     | H     | ↑     | X      | X     | a        | b | c       | d     | a        | b        | c        | d        |          |          |
| H     | L     | H     | ↑     | X      | H     | X        | X | X       | X     | H        | $Q_{An}$ | $Q_{Bn}$ | $Q_{Cn}$ |          |          |
| H     | L     | H     | ↑     | X      | L     | X        | X | X       | X     | L        | $Q_{An}$ | $Q_{Bn}$ | $Q_{Cn}$ |          |          |
| H     | H     | L     | ↑     | H      | X     | X        | X | X       | X     | $Q_{Bn}$ | $Q_{Cn}$ | $Q_{Dn}$ | H        |          |          |
| H     | H     | L     | ↑     | L      | X     | X        | X | X       | X     | $Q_{Bn}$ | $Q_{Cn}$ | $Q_{Dn}$ | L        |          |          |
| H     | L     | L     | X     | X      | X     | X        | X | X       | X     | $Q_{A0}$ | $Q_{B0}$ | $Q_{C0}$ | $Q_{D0}$ |          |          |

H = high level (steady state)  
 L = low level (steady state)  
 X = irrelevant (any input, including transitions)  
 ↑ = transition from low to high level  
 a, b, c, d = the level of steady-state input at inputs A, B, C, or D, respectively  
 $Q_{A0}, Q_{B0}, Q_{C0}, Q_{D0}$  = the level of  $Q_A, Q_B, Q_C,$  or  $Q_D,$  respectively, before the indicated steady-state input conditions were established  
 $Q_{An}, Q_{Bn}, Q_{Cn}, Q_{Dn}$  = the level of  $Q_A, Q_B, Q_C, Q_D,$  respectively, before the most recent ↑ transition of the clock.

typical clear, load, right-shift, left-shift, inhibit, and clear sequences



## II.7 Circuit 74195

- Synchronous Parallel Load
- Positive-Edge-Triggered Clocking
- Parallel Inputs and Outputs from Each Flip-Flop
- Direct Overriding Clear
- J and  $\bar{K}$  Inputs to First Stage
- Complementary Outputs from Last Stage
- For Use in High-Performance: Accumulators/Processors  
Serial-to-Parallel, Parallel-to-Serial Converters

### description

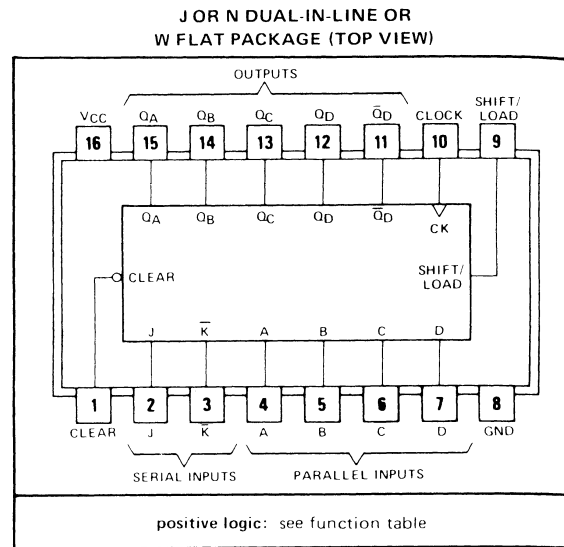
These 4-bit registers feature parallel inputs, parallel outputs, J-K serial inputs, shift/load control input, and a direct overriding clear. All inputs are buffered to lower the input drive requirements. The registers have two modes of operation:

- Parallel (Broadside) Load
- Shift (In direction  $Q_A$  toward  $Q_D$ )

Parallel loading is accomplished by applying the four bits of data and taking the shift/load control input low. The data is loaded into the associated flip-flop and appears at the outputs after the positive transition of the clock input. During loading, serial data flow is inhibited.

Shifting is accomplished synchronously when the shift/load control input is high. Serial data for this mode is entered at the J-K inputs. These inputs permit the first stage to perform as a J- $\bar{K}$ , D-, or T-type flip-flop as shown in the function table.

The high-performance 'S195, with a 105-megahertz typical maximum shift-frequency, is particularly attractive for very-high-speed data processing systems. In most cases existing systems can be upgraded merely by using this Schottky-clamped shift register.



| TYPE   | TYPICAL<br>MAXIMUM CLOCK<br>FREQUENCY | TYPICAL<br>POWER<br>DISSIPATION |
|--------|---------------------------------------|---------------------------------|
| '195   | 39 MHz                                | 195 mW                          |
| 'LS195 | 28 MHz                                | 50 mW                           |
| 'S195  | 105 MHz                               | 375 mW                          |

FUNCTION TABLE

| CLEAR | SHIFT/<br>LOAD | CLOCK | INPUTS |           |          |   | OUTPUTS |       |                |          |             |          |                |
|-------|----------------|-------|--------|-----------|----------|---|---------|-------|----------------|----------|-------------|----------|----------------|
|       |                |       | SERIAL |           | PARALLEL |   | $Q_A$   | $Q_B$ | $Q_C$          | $Q_D$    | $\bar{Q}_D$ |          |                |
|       |                |       | J      | $\bar{K}$ | A        | B | C       | D     |                |          |             |          |                |
| L     | X              | X     | X      | X         | X        | X | X       | X     | L              | L        | L           | L        | H              |
| H     | L              | ↑     | X      | X         | a        | b | c       | d     | a              | b        | c           | d        | $\bar{d}$      |
| H     | H              | L     | X      | X         | X        | X | X       | X     | $Q_{A0}$       | $Q_{B0}$ | $Q_{C0}$    | $Q_{D0}$ | $\bar{Q}_{D0}$ |
| H     | H              | ↑     | L      | H         | X        | X | X       | X     | $Q_{A0}$       | $Q_{A0}$ | $Q_{Bn}$    | $Q_{Cn}$ | $\bar{Q}_{Cn}$ |
| H     | H              | ↑     | L      | L         | X        | X | X       | X     | L              | $Q_{An}$ | $Q_{Bn}$    | $Q_{Cn}$ | $\bar{Q}_{Cn}$ |
| H     | H              | ↑     | H      | H         | X        | X | X       | X     | H              | $Q_{An}$ | $Q_{Bn}$    | $Q_{Cn}$ | $\bar{Q}_{Cn}$ |
| H     | H              | ↑     | H      | L         | X        | X | X       | X     | $\bar{Q}_{An}$ | $Q_{An}$ | $Q_{Bn}$    | $Q_{Cn}$ | $\bar{Q}_{Cn}$ |

H = high level (steady state)

L = low level (steady state)

X = irrelevant (any input, including transitions)

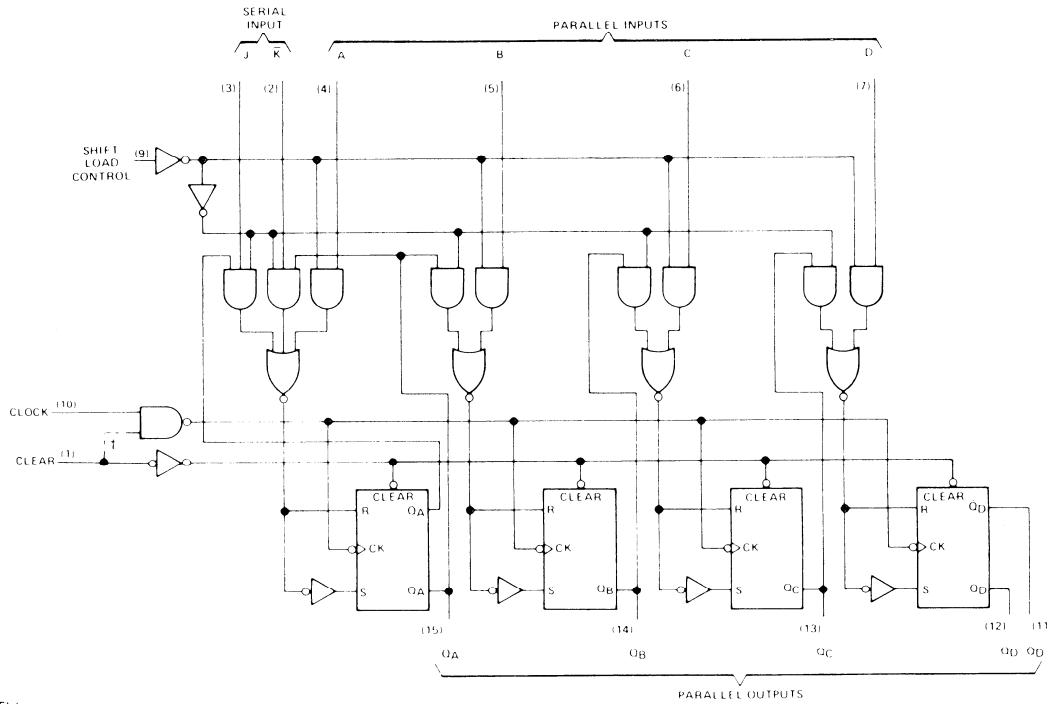
↑ = transition from low to high level

a, b, c, d = the level of steady-state input at A, B, C, or D, respectively

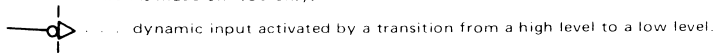
$Q_{A0}$ ,  $Q_{B0}$ ,  $Q_{C0}$ ,  $Q_{D0}$  = the level of  $Q_A$ ,  $Q_B$ ,  $Q_C$ , or  $Q_D$ , respectively, before the indicated steady-state input conditions were established

$Q_{An}$ ,  $Q_{Bn}$ ,  $Q_{Cn}$  = the level of  $Q_A$ ,  $Q_B$ , or  $Q_C$ , respectively, before the most-recent transition of the clock

functional block diagram



† This connection is made on '195 only.



typical clear, shift, and load sequences

