

Université des Sciences et de la Technologie d'Oran
Mohamed BOUDIAF
USTO-MB



Faculté de Génie Electrique
Département d'Electronique

TRAVAUX PRATIQUES
Electronique Numérique
(2^{ème} année LMD)

LOGIQUE COMBINATOIRE
&
LOGIQUE SEQUENTIELLE

Mr B. BOUCHIBA

TABLE DES TRAVAUX PRATIQUES

INTRODUCTION.....	3
TP1: LES FONCTIONS DE BASE.....	5
TP2: CODE et CONVERSION (décimale - binaire)	8
TP3: CODE et CONVERSION (binaire - décimale)	10
TP 4: GENERATEUR DE BIT DE PARITE DETECTEUR D'EGALITE.....	11
TP 5: COMPAREUR.....	13
TP6: TRANSCODEURS, ROUE CODEUSE, DECODEUR BCD / 7 SEGMENTS.....	14
TP7: MULTIPLEXEUR et DEMULTIPLEXEUR / DECODEUR.....	17
TP8: CIRCUITS SEQUENTIELS (LES BASCULES).....	19
TP9: COMPTEURS ASYNCHRONES.....	23
TP10: COMPTEURS SYNCHRONES.....	25
TP11: LES REGISTRES.....	27
ANNEXE 1: FONCTIONS DE BASE ET LEURS SYMBOLES.....	30
ANNEXE 2: BROCHAGE DES CIRCUITS INTEGRES TTL.....	31

Introduction:

L'objectif des travaux pratiques est de compléter les notions théoriques qui sont vues dans les cours et travaux dirigés des matières enseignées. Ce complément permet d'enrichir pratiquement les connaissances acquises des matières étudiées.

Ce polycopié de travaux pratiques est destiné aux étudiants du semestre 4 du parcours LMD, filières Electronique, Génie Biomédicale et Télécommunication, de la matière Logique Numérique (Logique Combinatoire et Séquentielle). Selon le programme établi par le Ministère de l'Enseignement Supérieur et de la Recherche Scientifique, le volume horaire hebdomadaire est de 1H30 et réparti sur 15 semaines soit un volume horaire semestriel de 22H30.

Ce polycopié regroupe 11 TP qui permettent aux étudiants d'illustrer leurs connaissances théoriques et d'acquérir des bases solides en électronique numérique par la réalisation pratique de montages électroniques sur des simulateurs.

Les simulateurs utilisés sont de type "De Lorenzo" regroupant trois panneaux, un panneau qui regroupe l'alimentation et les variables d'entrées / sorties, un deuxième panneau pour la logique combinatoire et un troisième pour la logique séquentielle. D'autres modules supplémentaires de même type existent pour la simulation des autres circuits numériques.



Manipulation et compte-rendu:

Chaque séance de travaux pratique est prévue pour faire un TP en relation avec une partie du cours de logique combinatoire et séquentielle. L'étudiant doit préparer à l'avance le TP et la séance doit être entièrement consacrée à la réalisation pratique des montages et leurs simulations.

A la fin de chaque séance de TP, l'étudiant doit remettre un compte-rendu qui regroupe trois parties: une partie théorique, une partie pratique et une conclusion.

Absences:

La présence aux travaux pratiques est obligatoire (contrôle continu).

Toute absence non justifiée ou un compte-rendu non remis seront sanctionné par un zéro.

Le TP devra obligatoirement être rattrapé dans le cas où l'absence a été justifiée.

L'absence non justifiée à trois séances de TP entrainera un zéro pour la note finale de TP.

Les retards doivent être minimisés; si le retard dépasse 15mn l'accès au laboratoire sera refusé et les conséquences seront identiques à celles d'une absence non justifiée.

La note finale de TP sera calculé de la manière suivante:

$$TP = [\text{moyenne (compte-rendu + travail en laboratoire)}] * 70\% + [\text{test de TP}] * 30\%$$

TP1 LES FONCTIONS DE BASE

1 : Rappels théoriques :

En algèbre de BOOLE, une variable ne peut prendre que deux valeurs possibles, 0 ou 1.

1-1 : Fonction NON (NOT): On la définit comme étant le complément de la variable d'entrée E notée /E (E barre).

Table de vérité

E	S
0	1
1	0

Symbole



Fonction réalisée

$$S = \overline{E}$$

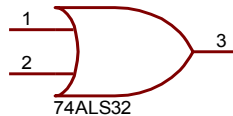
$$S = \text{NON } E$$

1-2 : Fonction OU (OR): Cette fonction notée $(E1 + E2)$ est la réunion de deux variables booléennes E1 et E2. Elle prend la valeur 1 si l'une des deux entrées est à 1.

Table de vérité

E1	E2	S
0	0	0
0	1	1
1	0	1
1	1	1

Symbole



Fonction réalisée

$$S = E1 + E2$$

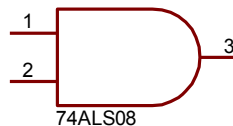
$$S = E1 \text{ OU } E2$$

1-3 : Fonction ET (AND): Cette fonction notée $(E1 \cdot E2)$ est l'intersection de deux variables booléennes E1 et E2. Cette fonction prend la valeur 1 que si les deux variables d'entrées sont égales à 1.

Table de vérité

E1	E2	S
0	0	0
0	1	0
1	0	0
1	1	1

Symbole



Fonction réalisée

$$S = E1 \cdot E2$$

$$S = E1 \text{ AND } E2$$

NB: Ces trois fonctions sont la base des réalisations de toutes les autres fonctions.

Théorème de MORGAN : Il définit les relations entre l'opération complément et les opérations de base.

$\overline{A + B} = \overline{A} \cdot \overline{B}$ le complément d'une somme = produit des compléments des variables.

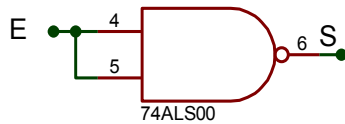
$\overline{A \cdot B} = \overline{A} + \overline{B}$ le complément d'un produit = somme des compléments des variables.

2 : Manipulation :

Dans chacun des cas, il est demandé de :

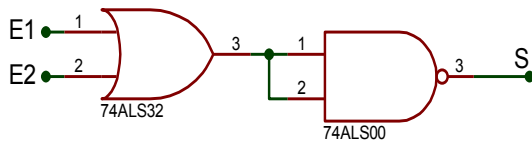
- simuler le circuit,
- déduire la table de vérité ainsi que la fonction réalisée.

2-1



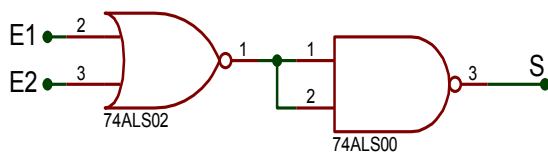
E		S
0		
1		

2-2



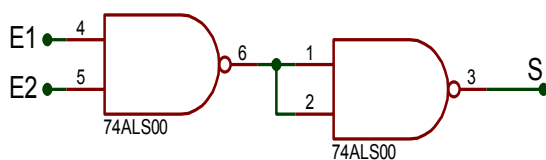
E1	E2		S
0	0		
0	1		
1	0		
1	1		

2-3



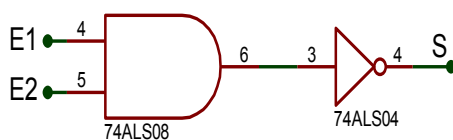
E1	E2		S
0	0		
0	1		
1	0		
1	1		

2-4



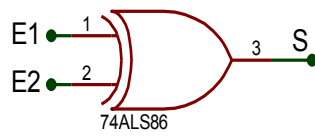
E1	E2		S
0	0		
0	1		
1	0		
1	1		

2-5

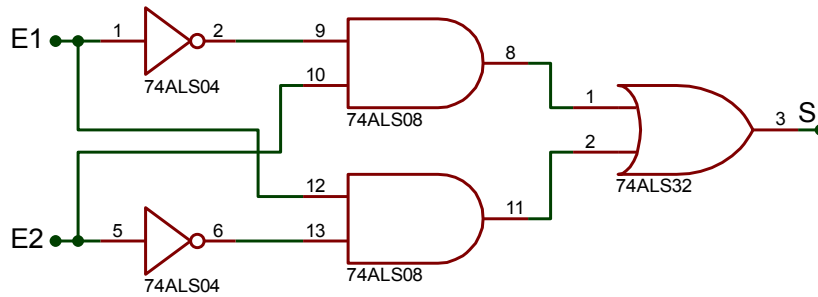


E1	E2		S
0	0		
0	1		
1	0		
1	1		

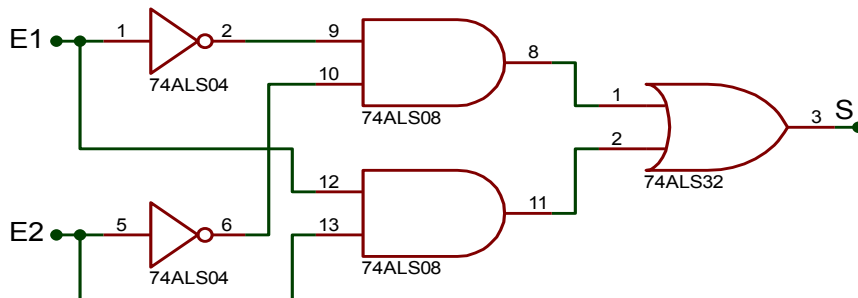
2-6



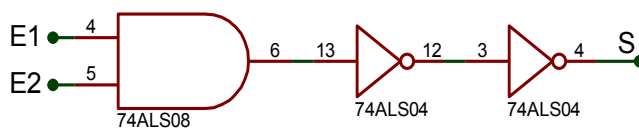
2-7



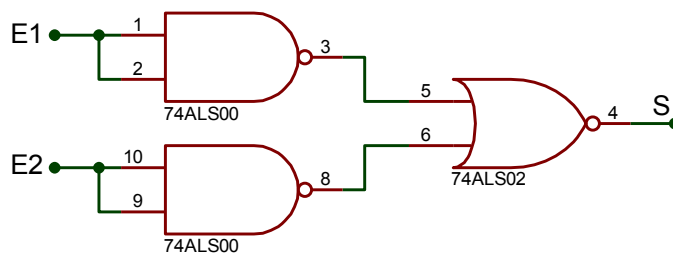
2-8



2-9



2-10



- Comparer les expressions 2-9 et 2-10.
- Que pouvez-vous vérifier à l'aide de ces deux circuits.

TP 2

CODE et CONVERSION

Décimale - binaire

1- Rappel sur le codage :

Il existe plusieurs systèmes de numération, systèmes décimal, octal, hexadécimal, binaire, ...

Le système binaire qui est utilisé dans la technique logique possède deux symboles, 0 et 1, appelés bits. Plusieurs algorithmes permettent le passage d'un système de numération à un autre. Dans le processus de saisie, de transmission et de traitement de l'information, on utilise des informations codées en prenant pour chaque symbole des combinaisons de bits qui forment des codes.

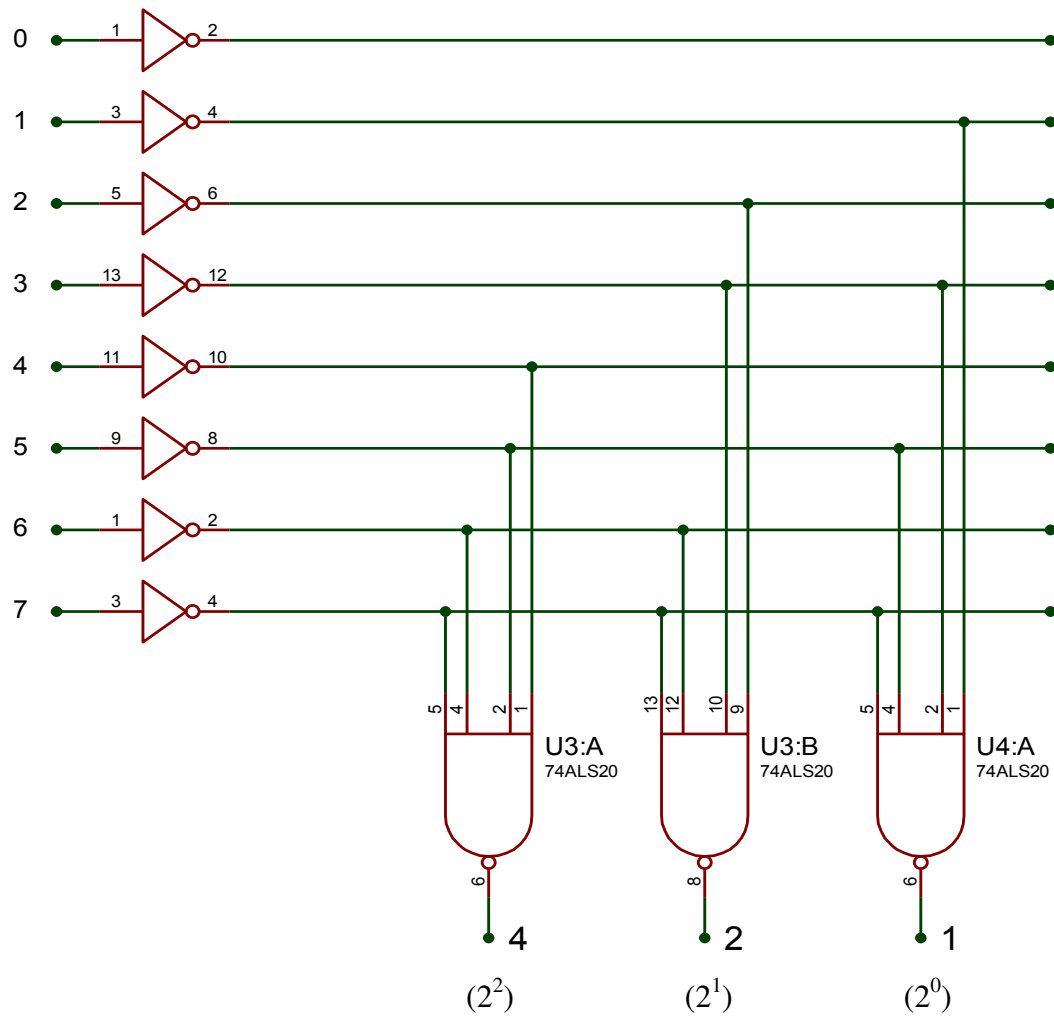
Ainsi, pour coder les chiffres décimaux (de 0 à 9), on utilise couramment 4 bits, ce qui donne un grand choix possible pour le codage. On obtient les codes décimaux codés binaires parmi lesquels on peut citer : BCD, binaire réfléchi, AIKEN, Excédent 3, GRAY, ... ; ces codes possèdent des propriétés intéressantes telle que pondération, symétrie, l'adjacence, complémentarité ...

	BCD	AIKEN	Excédent 3	Réfléchi	GRAY
	8 4 2 1	2 4 2 1			
0	0 0 0 0	0 0 0 0	0 0 1 1	0 0 0 0	0 0 0 0
1	0 0 0 1	0 0 0 1	0 1 0 0	0 0 0 1	0 0 0 1
2	0 0 1 0	0 0 1 0	0 1 0 1	0 0 1 1	0 0 1 1
3	0 0 1 1	0 0 1 1	0 1 1 0	0 0 1 0	0 0 1 0
4	0 1 0 0	0 1 0 0	0 1 1 1	0 1 1 0	0 1 1 0
5	0 1 0 1	1 0 1 1	1 0 0 0	0 1 1 1	0 1 1 1
6	0 1 1 0	1 1 0 0	1 0 0 1	0 1 0 1	0 1 0 1
7	0 1 1 1	1 1 0 1	1 0 1 0	0 1 0 0	0 1 0 0
8	1 0 0 0	1 1 1 0	1 0 1 1	1 1 0 0	1 1 0 0
9	1 0 0 1	1 1 1 1	1 1 0 0	1 1 0 1	1 0 0 0

2- Manipulation :

2-1. Conversion décimale – BCD :

Soit le montage suivant qui réalise une conversion décimale – binaire pour les chiffres 1, 2, ..., 6 et 7.



- Expliquer le fonctionnement et donner la table de vérité.
- Réaliser le montage et vérifier son fonctionnement.
- Réaliser l'extension du circuit pour les 10 chiffres décimaux.

2-2. Conversion décimale – code excédent 3 :

Donner le circuit qui permet de coder les 10 caractères du système décimal en code excédent 3. Réaliser ce montage et vérifier son fonctionnement.

TP 3

CODE et CONVERSION

Binaire - décimale

Soit une table de Karnaugh dans laquelle on écrit dans chaque case l'équivalent décimal pour les 10 symboles du système décimal. Les cases restantes contiennent des termes indifférents (\emptyset) qui peuvent éventuellement contribuer aux simplifications des équations booléennes.

CD \ AB	00	01	11	10
00	0	1	3	2
01	4	5	7	6
11	\emptyset	\emptyset	\emptyset	\emptyset
10	8	9	\emptyset	\emptyset

- Ecrire les équations des fonctions de sorties (nombres décimaux) en fonction des variables binaires A, B, C et D.

- a) avec réjection des données fausses.
- b) sans réjection des données fausses.

Exemple : l'équation de sortie du 5 s'écrit :

CD \ AB	00	01	11	10
00	0	0	0	0
01	0	1	0	0
11	\emptyset	\emptyset	\emptyset	\emptyset
10	0	0	\emptyset	\emptyset

avec rejection des données fausses
 $5 = /AB/CD$

CD \ AB	00	01	11	10
00	0	0	0	0
01	0	1	0	0
11	\emptyset	\emptyset	\emptyset	\emptyset
10	0	0	\emptyset	\emptyset

sans réjection des données fausses
 $5 = B/CD$

- Donner les schémas logiques des points a et b en utilisant les portes AND, OR et NOT.
- Simuler le fonctionnement des schémas trouvés et donner les tables de vérités.
- Donner une autre conception des schémas trouvés en utilisant uniquement des portes NOR ou NAND.

TP 4

GENERATEUR DE BIT DE PARITE DETECTEUR D'EGALITE

1- Rappel :

Pour assurer une transmission et un traitement correcte de l'information codée notamment en présence de bruit, on utilise des codes spéciaux comportant un ou plusieurs bits supplémentaires permettant la détection des erreurs.

La plus simple méthode de détection des erreurs dans le traitement de l'information est le contrôle de parité. Cette méthode consiste à ajouter à chaque combinaison binaire un bit de parité, 0 ou 1, tel que le nombre total des 1 contenus dans la combinaison est pair ou impair.

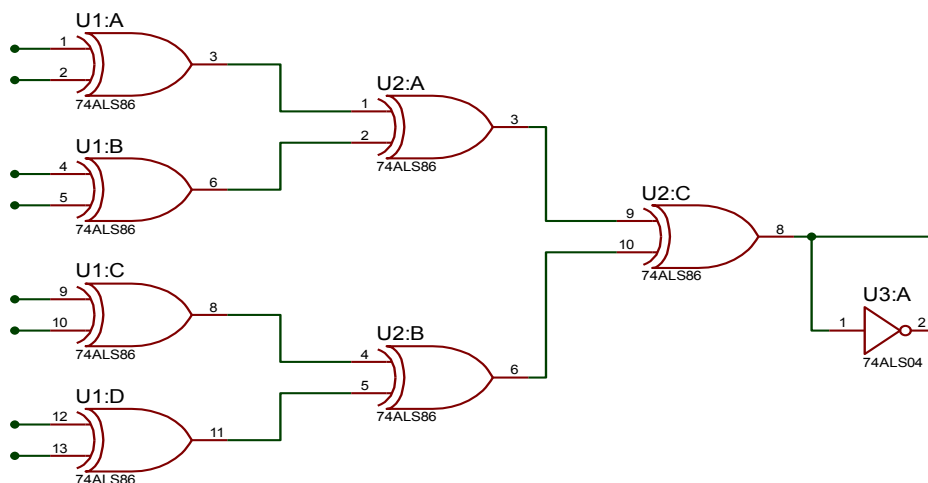
Exemple : Soit un nombre binaire $a_3a_2a_1a_0$, on lui ajoute un bit de parité p tel que $a_3a_2a_1a_0p$ est impair.

a_3	a_2	a_1	a_0	P
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
.
.
1	1	1	1	1

2- Manipulation:

2-1: Générateur de bit de parité :

Soit le schéma d'un contrôleur de parité pour 8 bits réalisé à l'aide des portes « OU exclusif ».



- Suivant cet exemple, réaliser un contrôleur de parité à 4 bits.
- Indiquer l'intérêt de prévoir deux sorties complémentaires.
- Simuler le fonctionnement et donner la table de vérité.

2-2 : Détecteur d'égalité :

A l'aide des circuits identités, on peut réaliser la comparaison des nombres binaires et la détection d'égalité. La table de vérité suivante montre l'égalité de deux nombres binaires (1 bit chacun):

a	b	E
0	0	1
0	1	0
1	0	0
1	1	1

$$E = 1 \quad \text{si} \quad a = b$$

$$a = b \longrightarrow E = \neg a/b + ab$$

Soit à comparer deux nombres A et B tel que :

$$A = a_2a_1a_0 \quad \text{et} \quad B = b_2b_1b_0$$

$$A = B \longrightarrow \text{si } a_2 = b_2 \quad \text{et} \quad a_1 = b_1 \quad \text{et} \quad a_0 = b_0$$

- Donner l'équation de sortie du circuit qui détecte l'égalité des deux nombres A et B.
- Exprimer cette équation à l'aide des circuits identités.
- Simuler le fonctionnement et donner sa table de vérité.

TP 5 COMPARATEUR

Le comparateur est un circuit combinatoire qui permet de comparer deux mots binaires constitués de plusieurs bits et de donner le résultat de la comparaison sur trois sorties.

1- Comparateur à un bit :

Soit la table de vérité suivante :

A	B	I	E	S
0	0	0	1	0
0	1	1	0	0
1	0	0	0	1
1	1	0	1	0

I : A < B

E : A = B

S : A > B

- Donner les équations de sorties I, E et S.
- Donner le schéma et simuler son fonctionnement.

2- Comparateur 7485 TTL :

Soit le circuit intégré 7485 TTL (comparateur à 4 bits), qui réalise la comparaison de deux mots binaires de 4 bits chacun; le résultat de la comparaison apparaît sur 3 sorties, **S** si A>B, **E** si A=B ou **I** si A<B.

En utilisant ce circuit donner le schéma qui réalise une comparaison de deux nombres binaires tel que :

$$A = a_2a_1a_0 \quad \text{et} \quad B = b_2b_1b_0$$

$$A > B \longrightarrow \text{si} \begin{cases} (a_2 > b_2) \text{ ou} \\ (a_2 = b_2 \text{ et } a_1 > b_1) \text{ ou} \\ (a_2 = b_2 \text{ et } a_1 = b_1 \text{ et } a_0 > b_0) \end{cases}$$

$$A = B \longrightarrow \text{si} \quad (a_2 = b_2 \text{ et } a_1 = b_1 \text{ et } a_0 = b_0)$$

$$A < B \longrightarrow \text{si} \begin{cases} (a_2 < b_2) \text{ ou} \\ (a_2 = b_2 \text{ et } a_1 < b_1) \text{ ou} \\ (a_2 = b_2 \text{ et } a_1 = b_1 \text{ et } a_0 < b_0) \end{cases}$$

- Donner la table de vérité ainsi que les équations de sorties **S** (A>B), **E** (A=B) et **I** (A<B).
- Simuler le circuit et vérifier son fonctionnement.
- A quoi servent les entrées supplémentaires A<B, A=B, et A>B.

TP 6

TRANSCODEURS, ROUE CODEUSE, DECODEUR BCD / 7 SEGMENTS

L'opération de passage d'un code à un autre code est réalisée par des circuits appelés transcodeurs. Les équations des sorties qui sont fonction des entrées peuvent être simplifiées en appliquant les méthodes classiques de synthèse des multiplexes combinatoires.

1- Manipulation :

1-1- Transcodeur BCD / XS3 :

	BCD E ₃ E ₂ E ₁ E ₀				Excédent 3 S ₃ S ₂ S ₁ S ₀			
0	0	0	0	0	0	0	1	1
1	0	0	0	1	0	1	0	0
2	0	0	1	0	0	1	0	1
3	0	0	1	1	0	1	1	0
4	0	1	0	0	0	1	1	1
5	0	1	0	1	1	0	0	0
6	0	1	1	0	1	0	0	1
7	0	1	1	1	1	0	1	0
8	1	0	0	0	1	0	1	1
9	1	0	0	1	1	1	0	0

Table de vérité d'un transcodeur
BCD/XS3

- Donner les équations de sorties du transcodeur BCD / excédent 3.
- Donner le schéma et Simuler le fonctionnement.

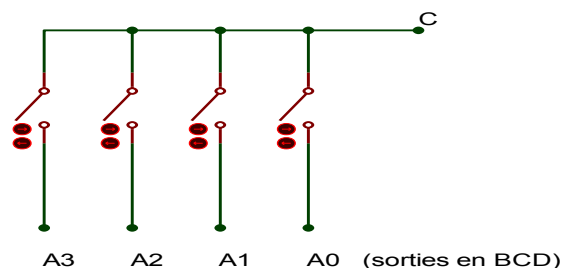
1-2- Roue codeuse en code BCD :

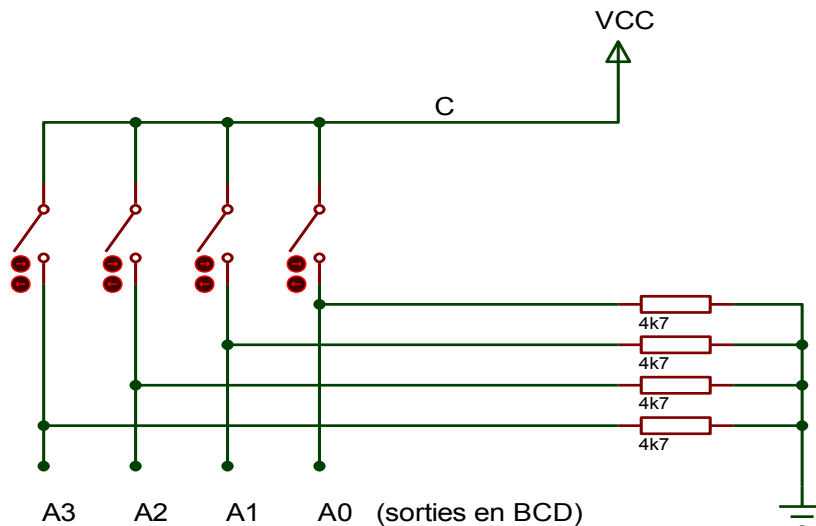
Une roue codeuse est intérieurement constituée d'un système mécanique rotatif, qui est gravé avec les dix caractères du système décimal ; grâce à sa rotation il est possible de sélectionner le caractère décimal dessiné (de 0 à 9) qui sera visualisé sur la face avant de la roue. En sortie, un code en BCD 8421 est associé à chaque chiffre sélectionné sur la face avant de la roue codeuse.

Cette roue est constituée de 4 interrupteurs indépendants, un pour chaque sortie, qui sont commandés mécaniquement à l'aide du système mécanique rotatif. Ces interrupteurs seront fermés ou ouverts selon le code adopté pour le chiffre affiché sur la roue codeuse.

La roue codeuse se présente de la manière suivante :

Soit le montage :

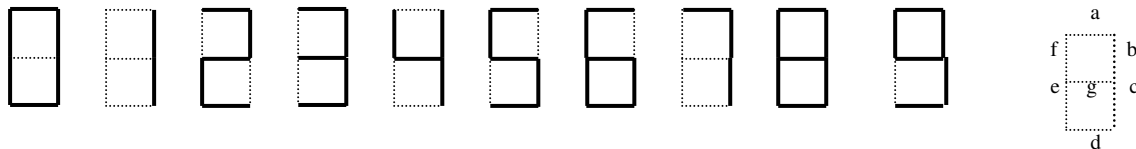




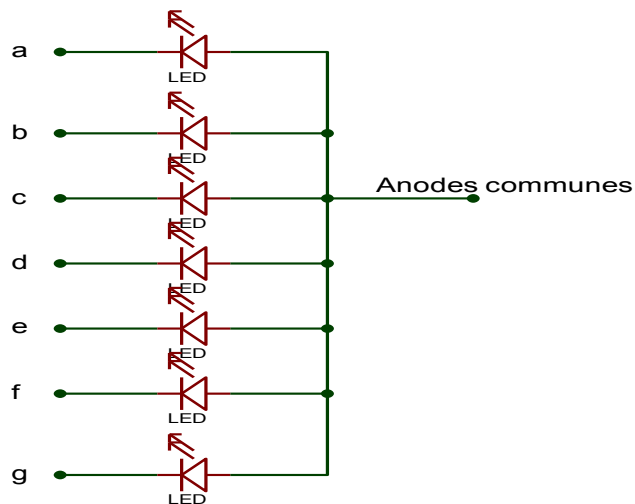
- Réaliser le montage et dresser la table de vérité.
- Expliquer le rôle des résistances.

1-3- Décodeur BCD / 7 segments :

Dans beaucoup d'applications, on a besoin d'afficher le résultat en utilisant des afficheurs qui sont à 7 segments disposés de façon à composer, par une sélection adéquate, les dix symboles du système décimal.



Chaque segment est constitué d'une ou plusieurs LEDs selon la luminosité demandée, et toutes les LEDs sont reliées entre elles par une borne commune, l'anode ou la cathode, d'où l'existence d'afficheurs à anode commune et d'afficheurs à cathode commune.

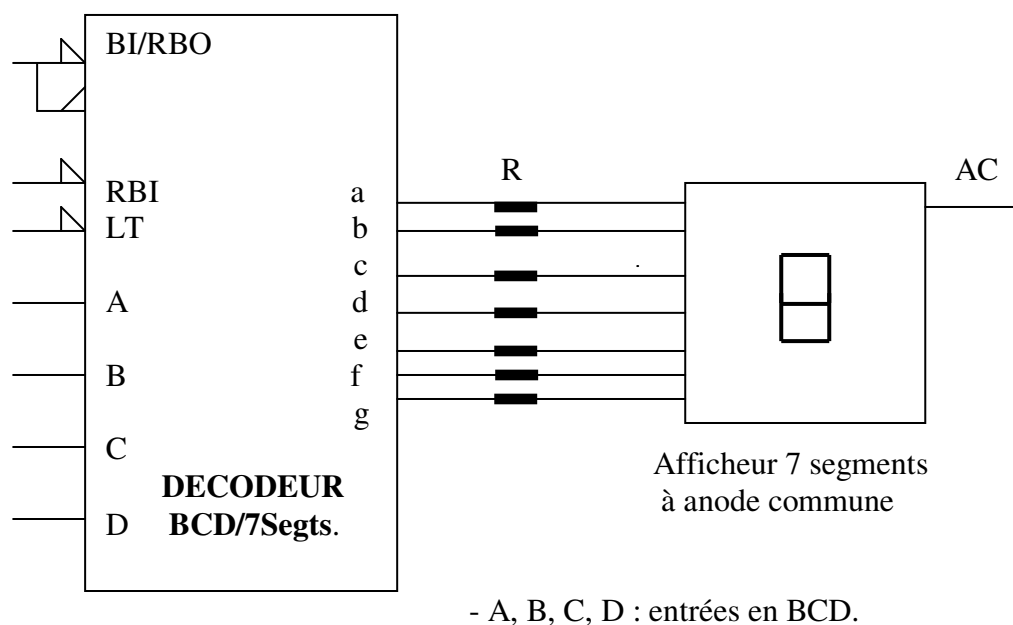


Pour commander ces afficheurs, il existe des circuits appelés décodeur BCD / 7segments tel que 7446, 7447, 7448, 7449. Par exemple pour commander un afficheur à cathode commune, on utilise le circuit 7448.

A ₃	A ₂	A ₁	A ₀		a	b	c	d	e	f	g
0	0	0	0		1	1	1	1	1	1	0
0	0	0	1		0	1	1	0	0	0	0
0	0	1	0		1	1	0	1	1	0	1
0	0	1	1		1	1	1	1	0	0	1
0	1	0	0		0	1	1	0	0	1	1
0	1	0	1		1	0	1	1	0	1	1
0	1	1	0		1	0	1	1	1	1	1
0	1	1	1		1	1	1	0	0	0	0
1	0	0	0		1	1	1	1	1	1	1
1	0	0	1		1	1	1	1	0	1	1

Table de vérité du décodeur BCD / 7segments 7448.

Réaliser le montage suivant,



* Les résistances $R = 270 \, \Omega$

- Réaliser le circuit en reliant les entrées du décodeur aux sorties de la roue codeuse.
- Simuler le fonctionnement et donner la table de vérité.
- Expliquer le rôle de LT, RBI et RBO.

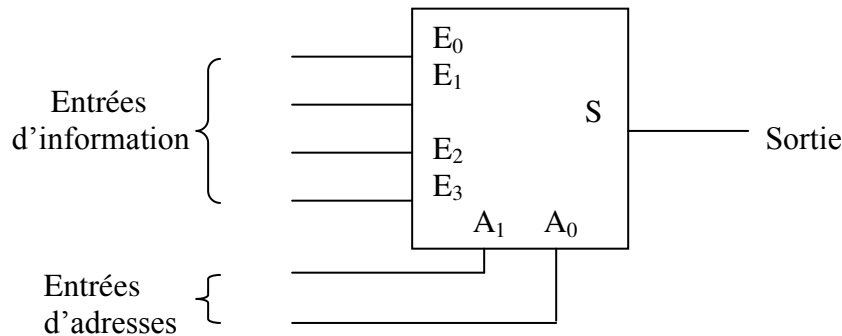
TP 7

MULTIPLEXEUR ET DEMULTIPLEXEUR / DECODEUR

1- Multiplexeur:

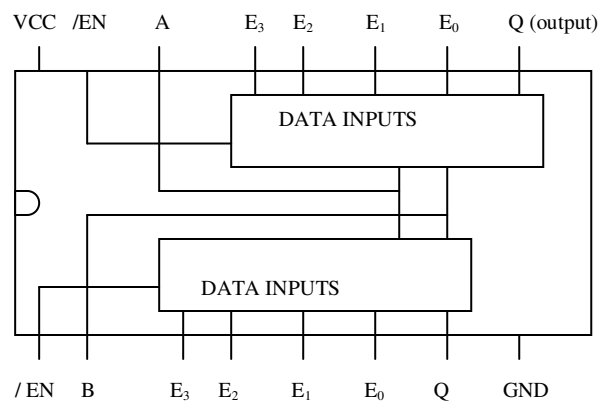
Un multiplexeur est un circuit qui est constitué de n entrées d'adresses, 2^n entrées d'information et d'une sortie. La sortie recopie l'entrée 2^i pour une adresse dont l'équivalent en décimal est i .

Soit un multiplexeur à deux entrées d'adresses



$$S = \neg A_1 \neg A_0 E_0 + \neg A_1 A_0 E_1 + A_1 \neg A_0 E_2 + A_1 A_0 E_3$$

Soit le circuit intégré TTL 74153 qui réalise la fonction “ multiplexeur “ et qui est schématisé comme suit :



Dual 4-line to 1-line data selectors / multiplexers

$\neg EN$: entrée de validation.

A, B: entrées d'adresses.

E_i : entrées d'information.

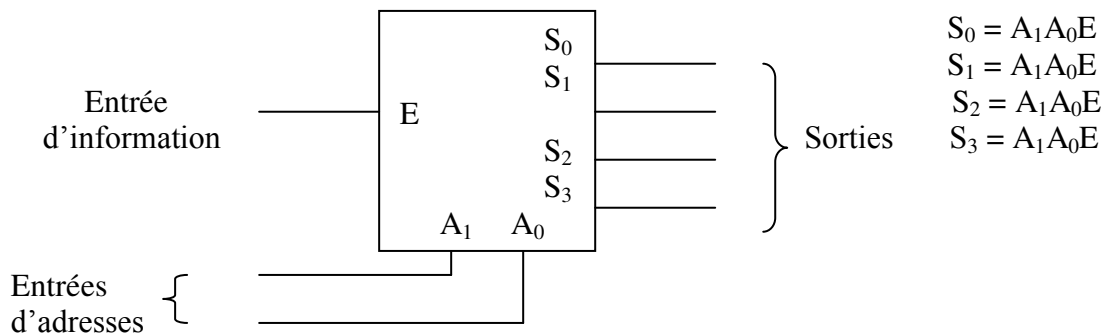
Q: sortie.

- Réaliser un multiplexeur à 2 entrées d'adresses.
- Vérifier son fonctionnement et donner la table de vérité.
- Modifier le câblage pour avoir un multiplexeur à 3 entrées d'adresses.

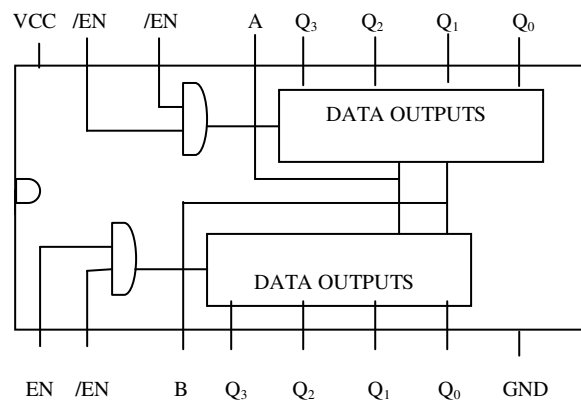
2- Démultiplexeur / Décodeur:

Un démultiplexeur est un multipole combinatoire qui comporte une entrée d'information, n entrées d'adresses et 2^n sorties. L'information présente à l'entrée est dirigée vers la sortie S_i pour une adresse dont l'équivalent en décimal est i.

Soit un démultiplexeur à deux entrées d'adresses



Soit le circuit intégré TTL 74155 qui réalise la fonction démultiplexeur / décodeur.



Dual 1 of 4 Decoder / Demultiplexer

EN, /EN: entrées de validation.

A, B: entrées d'adresses.

Q_i : sorties.

- Réaliser un décodeur 1 parmi 4, et donner sa table de vérité.
- Combiner les deux décodeurs 1 parmi 4 pour avoir un décodeur à 3 entrées d'adresses; déduire sa table de vérité.

TP 8

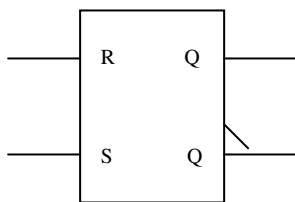
CIRCUITS SEQUENTIELS

LES BASCULES

1- Rappel théorique :

Une bascule appelée aussi bistable ou Flip-Flop est un circuit séquentiel qui est constitué d'une ou de plusieurs entrées de commandes, synchrones ou asynchrones, et de deux sorties complémentaires Q et \bar{Q} . La sortie Q qui représente l'état de la bascule est une combinaison des entrées de commande, du temps et de l'état de la sortie à l'instant t-1. L'état de la bascule reste mémorisé tant que les signaux de commandes restent inchangés. Il existe plusieurs types de bascules.

1-1- Bascule RS :



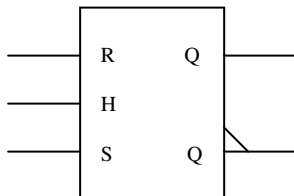
R (Reset) : entrée de remise à zéro.
S (Set) : entrée de mise à un.
Q et \bar{Q} : sorties complémentaires.

R	S		Q_{t+1}
0	0		Q_t
0	1		1
1	0		0
1	1		X

R = S = 1 cas interdit.

1-2- Bascule RSH :

Cette bascule possède le même fonctionnement que la bascule RS sauf que les entrées de commandes R et S sont synchronisées par un signal d'horloge.

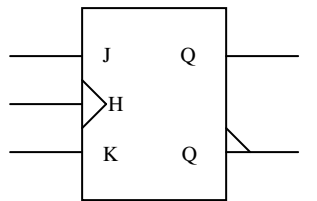


H	R	S		Q_{t+1}
0	X	X		Q_t
1	0	0		Q_t
1	0	1		1
1	1	0		0
1	1	1		X

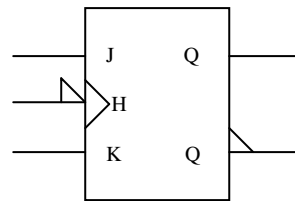
R = S = 1 cas interdit.

1-3- Bascule JK :

La bascule JK est une bascule synchrone à deux entrées de commandes J et K, et deux sorties Q et Q.



active sur front montant



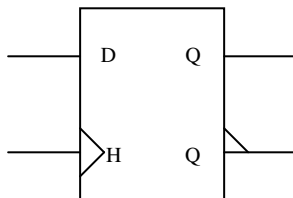
active sur front descendant

J	K		Q_{t+1}
0	0		Q_t
0	1		0
1	0		1
1	1		$\neg Q_t$

le changement d'état de la bascule se fait sur front actif (montant ou descendant).

1-4- Bascule D :

- Bascule D élémentaire :



La sortie recopie l'état de l'entrée D sur le front actif montant de H.

H	D		Q_{t+1}
0	X		Q_t
↑	0		0
↑	1		1

- Bascule D LATCH :

Le fonctionnement est le même sauf que l'entrée D est recopiée tant que H est au niveau 1.

1-5- Bascules avec des entrées asynchrones supplémentaires :

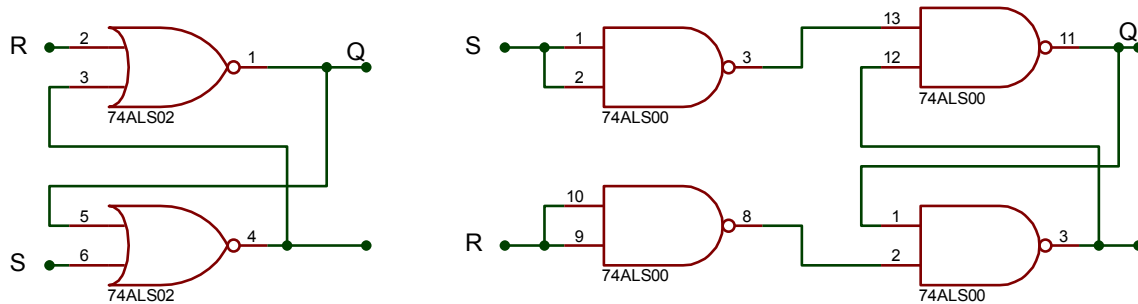
Les entrées des bascules RSH, JK et D sont des entrées synchrones; parfois il est nécessaire d'initialiser les bascules indépendamment du signal d'horloge, d'où l'utilisation de deux entrées supplémentaires asynchrones (actives à l'état bas ou l'état haut).

Reset ou Clear : remise à 0.

Set ou Preset : mise à 1.

2- Manipulation:**2-1- Bascule RS :**

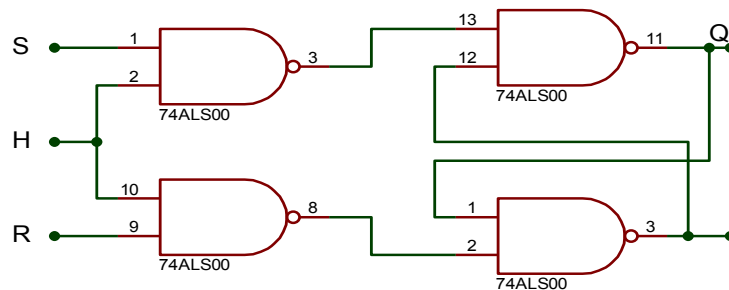
Une bascule RS peut être réalisée en utilisant des opérateurs NOR ou NAND.



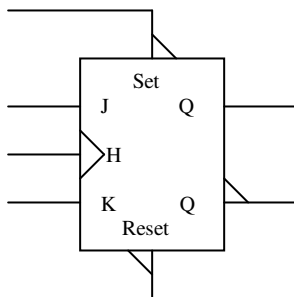
- Simuler ces deux montages et donner leurs tables de vérités.

2-2- Bascule RSH :

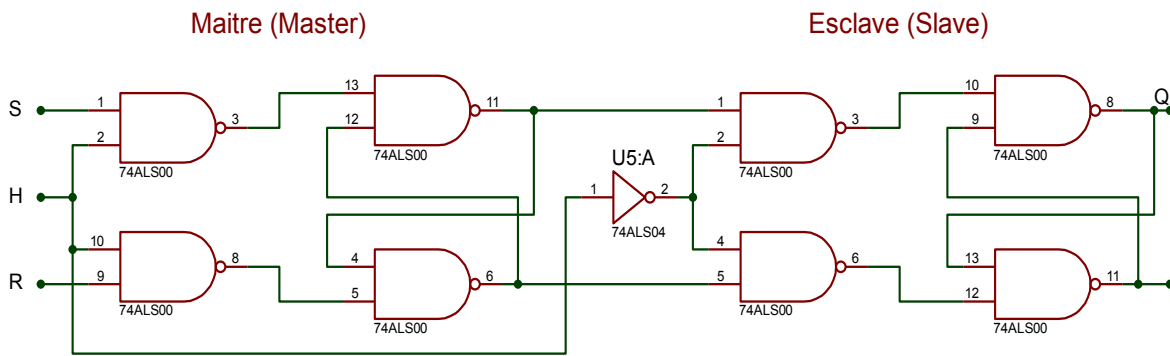
Soit le circuit de la bascule RS synchrone; réaliser le montage et vérifier le fonctionnement.

**2-3- Bascule JK :**

Soit le circuit de la bascule JK avec des entrées asynchrones set et reset.



Simuler cette bascule et donner sa table de vérité.

2-4- Bascule maître - esclave MS (Master-Slave) :

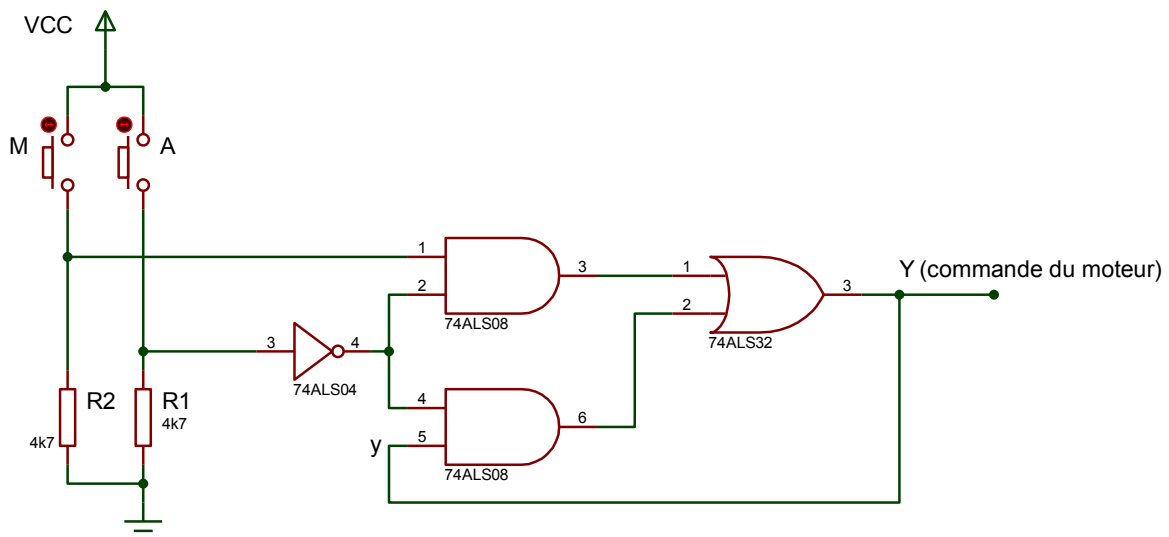
- Simuler le montage et dresser sa table de vérité en faisant figurer Q_M (sortie maître) et Q_S (sortie esclave).
- En déduire la bascule D.

2-5- Circuit séquentiel asynchrone:

Exemple d'un circuit séquentiel asynchrone :

Un moteur électrique est mis sous tension par l'intermédiaire de deux boutons poussoirs M et A commandant respectivement la marche et l'arrêt. Si l'on actionne le bouton M, le moteur commence à tourner et reste en marche lorsqu'on relâche ce bouton. Si on appuie sur A, le moteur s'arrête et reste à l'arrêt après avoir relâcher A. L'action simultanée sur les deux boutons est interdite.

Réaliser le circuit suivant sur le simulateur puis vérifier son fonctionnement.



Retrouver le circuit simulé en utilisant la synthèse des circuits séquentiels asynchrones par la méthode d'Huffman .

TP 9

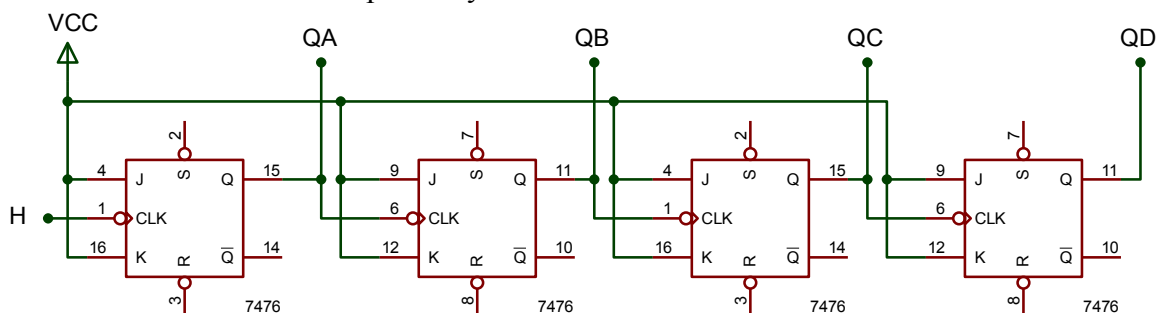
COMPTEURS ASYNCHRONES

Un compteur est un circuit qui, commandé à l'entrée par une succession d'impulsions, fournit à ses sorties une séquence donnée de configurations binaires qui se répète dans le temps. Le nombre M de configurations binaires différentes qu'on peut avoir en sortie s'appelle Modulo, et chaque configuration (état) du compteur doit rester stable jusqu'à l'application à l'entrée de l'impulsion suivante. Un compteur réalisé avec n bascules présente 2^n configurations distinctes.

Si les impulsions à compter sont appliquées seulement sur l'entrée de la première bascule et l'état de chaque bascule est fonction des états des bascules précédentes, on dit qu'il s'agit de compteurs asynchrones. Par contre dans les compteurs synchrones, les impulsions à compter sont appliquées simultanément sur les entrées de toutes les bascules.

1- Compteurs asynchrones :

Soit le circuit d'un compteur asynchrone avec des bascules JK.



- Réaliser le montage et vérifier le fonctionnement.
- Donner le diagramme des états du compteur.

Compteur asynchrone modulo M :

A l'aide d'un compteur asynchrone on peut réaliser un compteur diviseur par une puissance de 2. Si on veut réaliser une division par un nombre quelconque M (c'est à dire que la remise à zéro de toutes les bascules aura lieu après M impulsions comptées), on procède de la manière suivante :

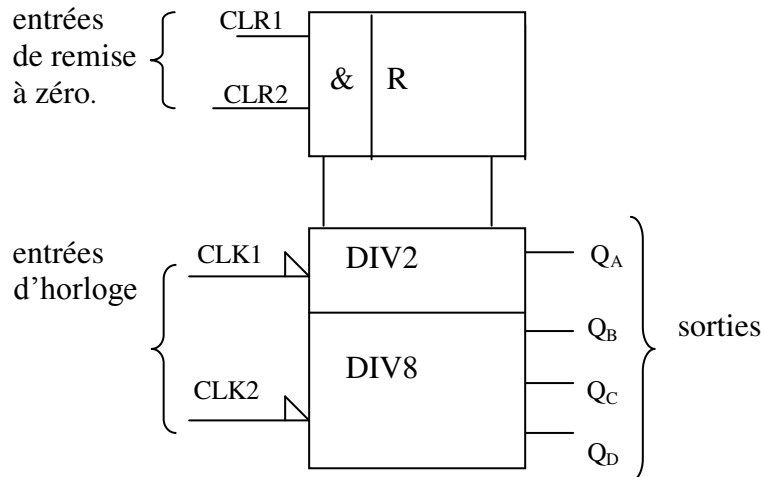
- On cherche le nombre de bascules nécessaires pour le comptage (N)
 $2^{N-1} < M < 2^N$ avec N = nombre de bascules et M = impulsions à compter.
- On réalise le compteur asynchrone à N bascules.
- On cherche l'état de comptage M (en binaire).
- On connecte toutes les sorties des bascules qui se trouvent à l'état 1 dans l'état M aux entrées d'un opérateur NAND dont la sortie sera branchée sur les entrées de commande asynchrone (/Reset ou /Clear) de toutes les bascules.

Quand le compteur arrive à l'état M , la sortie de la porte NAND initialise toutes les bascules. Pour éviter les difficultés d'initialisations provoquées par les différences entre le temps de commutation des divers étages on peut utiliser une cellule de stockage pour la commande d'effacement.

- Réaliser un compteur modulo 10, expliquer le fonctionnement.
- Simuler le circuit et donner le diagramme de comptage.

2- Compteur asynchrone intégré TTL 7493 :

Le but de cette manipulation est de vérifier le fonctionnement logique d'un compteur binaire asynchrone à 4 bits réalisé en forme intégré dans la puce 7493 TTL.



CLR1	CLR2		Q _D	Q _C	Q _B	Q _A
1	1		0	0	0	0
0	X		comptage			
X	0		comptage			

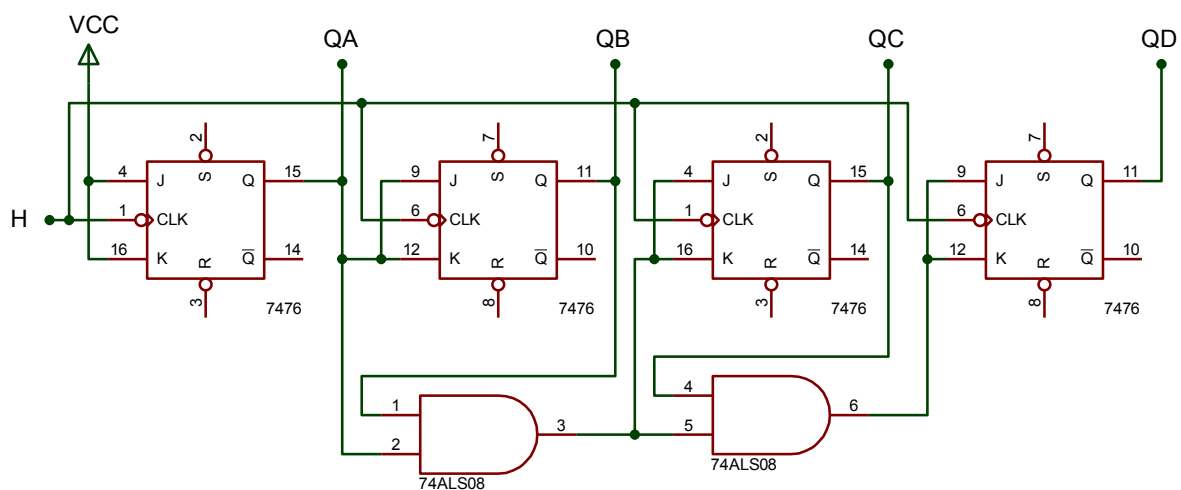
- Réaliser le circuit et vérifier son fonctionnement (relier la sortie Q_A à l'entrée CLK2).
- Quelles sont les fréquences des signaux des différentes sorties pour une fréquence du signal d'entrée F₀.
- Compléter le circuit pour avoir un compteur Mod 10.

TP 10 COMPTEURS SYNCHRONES

Dans ce type de compteur, toutes les bascules sont commandées par la même impulsion d'horloge. Les sorties de ces bascules commutent avec le seul délai causé par le temps de commutation typique d'une seule bascule; cela représente un avantage remarquable par rapport au compteur asynchrone, pour lequel le temps de commutation peut être de n fois supérieur, avec n représentant le nombre des bascules qui réalisent le compteur. Ce mode de comptage permet d'appliquer en entrée des signaux à fréquence très élevée.

1- Compteurs synchrones à bascules JK:

Soit le compteur synchrone à 4 bits, réalisé à l'aide des bascules JK.



- Retrouver le circuit du compteur en utilisant les tables de Karnaugh.

$Q_B Q_A$	00	01	11	10
$Q_D Q_C$				
00	0	1	3	2
01	4	5	7	6
11	12	13	15	14
10	8	9	11	10

Matrice de référence pour la
Succession des états de comptage

Q_t	Q_{t+1}	JK
0	0	0X
0	1	1X
1	0	X1
1	1	X0

Matrice de changement des états
pour une bascule JK

On doit établir les matrices de commande pour chaque bascule en indiquant les états correspondant à chaque changement d'état pendant le comptage.

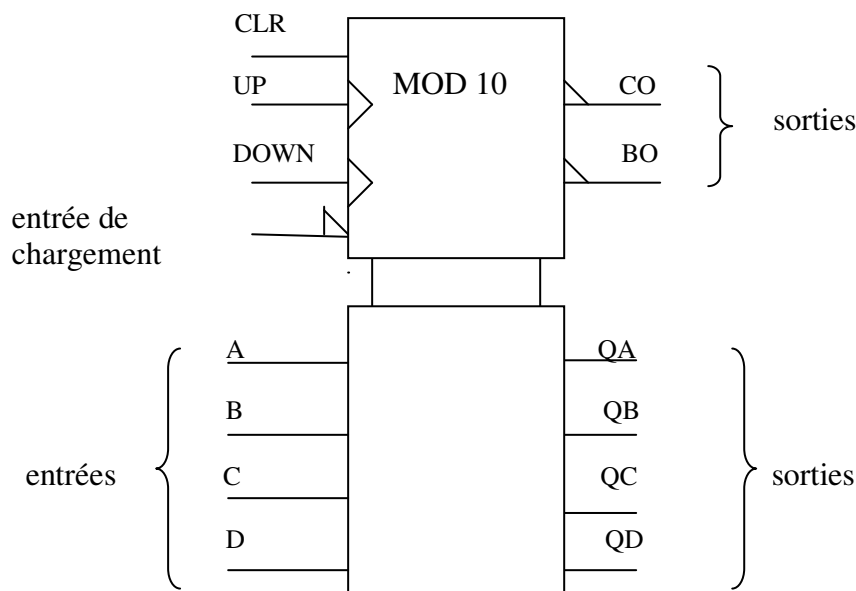
- Expliquer le fonctionnement du compteur.
- Simuler son fonctionnement.

Compteur synchrone modulo M :

En utilisant la même méthode de synthèse comme auparavant, on peut imposer une séquence de comptage avec remise à zéro du compteur à la M^{ème} impulsion.

2- Compteur synchrone intégré TTL 74192 :

Le but de cette application est de vérifier le fonctionnement du compteur intégré TTL 74192. Ce compteur Mod 10 est caractérisé par une capacité de comptage bidirectionnel ; il s'agit d'un compteur avant/arrière (UP / DOWN counter) et il est doté de quatre entrées qui permettent d'initialiser le compteur à une valeur binaire.

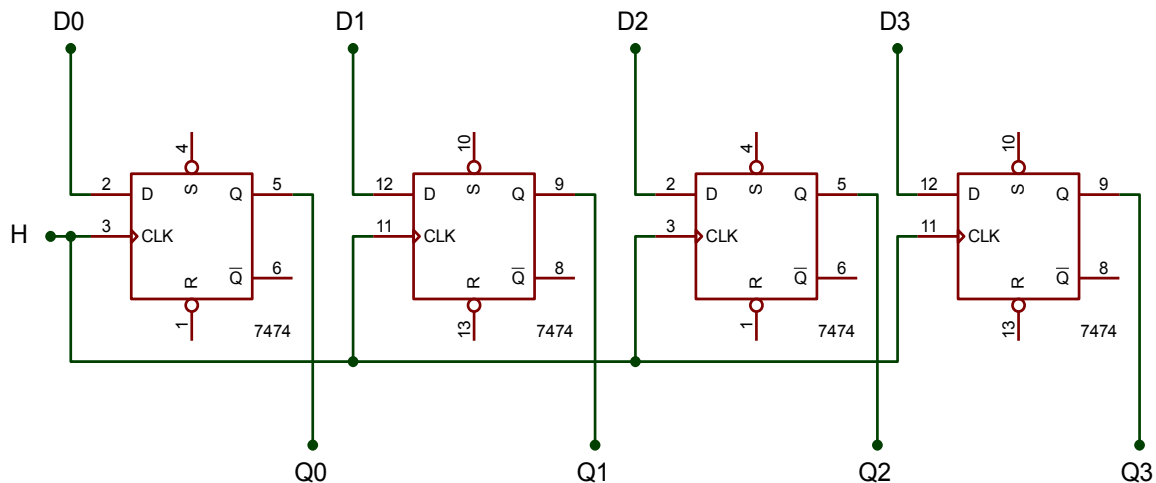


Entrées				Sorties			
CLR	LOAD	UP	DOWN	QD	QC	QB	QA
1	X	X	X	0	0	0	0
0	0	X	X	D	C	B	A
0	1	1	1	pas de commutation comptage décomptage			
0	1	↑	1				
0	1	1	↑				

- Réaliser le circuit et donner sa table de vérité.
- Que représentent les sorties /CO et /BO.

TP 11 LES REGISTRES

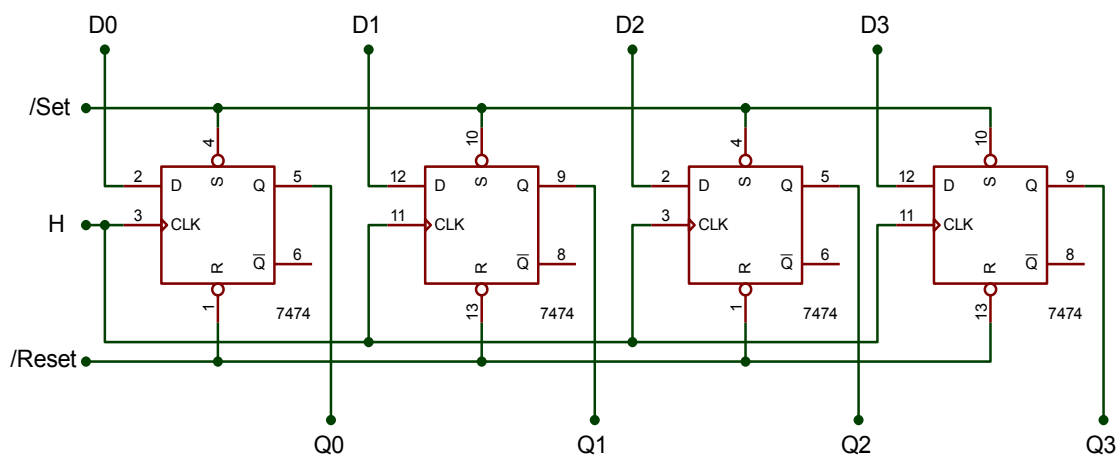
Dans les applications numériques un registre est un circuit logique qui est constitué d'un certain nombre d'éléments de mémoire (bascules) capables de mémoriser des informations binaires. La figure suivante montre un registre à 4 bits réalisé à l'aide des bascules D.



Les entrées sont transférées en sorties après une impulsion d'horloge.

1- Registre à chargement parallèle :

Pour ce type de registre l'information à mémoriser est présente sur n entrées.



C'est un registre à chargement parallèle réalisé avec des bascules D.

Les entrées R et S, actives à l'état bas, servent respectivement à la remise à 0 et la mise à 1 des bascules.

2- Registre à décalage (Shift register) :

Un registre à décalage est un registre dans lequel les informations binaires envoyées aux entrées sont transférées en sortie par des déplacements successifs et synchronisés d'une bascule

à une autre. La synchronisation des déplacements est assurée par les impulsions d'horloge ; et le déplacement s'opère de plusieurs manières différentes :

- de la droite vers la gauche (décalage à gauche / left shift register).
- de la gauche vers la droite (décalage à droite / right shift register).
- Réversible (bidirectionnel) utilisant les deux manières précédentes (bidirectionnal shift register).

Dans un registre à décalage, l'information peut être chargée de deux manières différentes soit en parallèle soit en série.

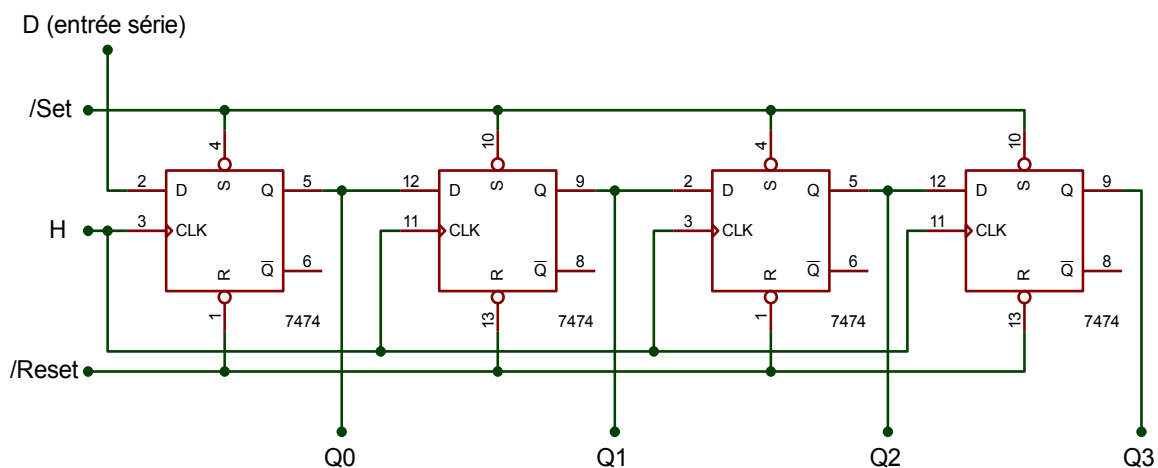
Entrée parallèle : dans ce cas les entrées sont appliquées en meme temps.

Entrée série : l'information est chargée séquentiellement bit après bit à l'entrée de la première bascule.

De même en sortie, l'information peut être lue en parallèle ou en série. On peut donc réaliser quatre modes différents d'entrée et de sortie des bits :

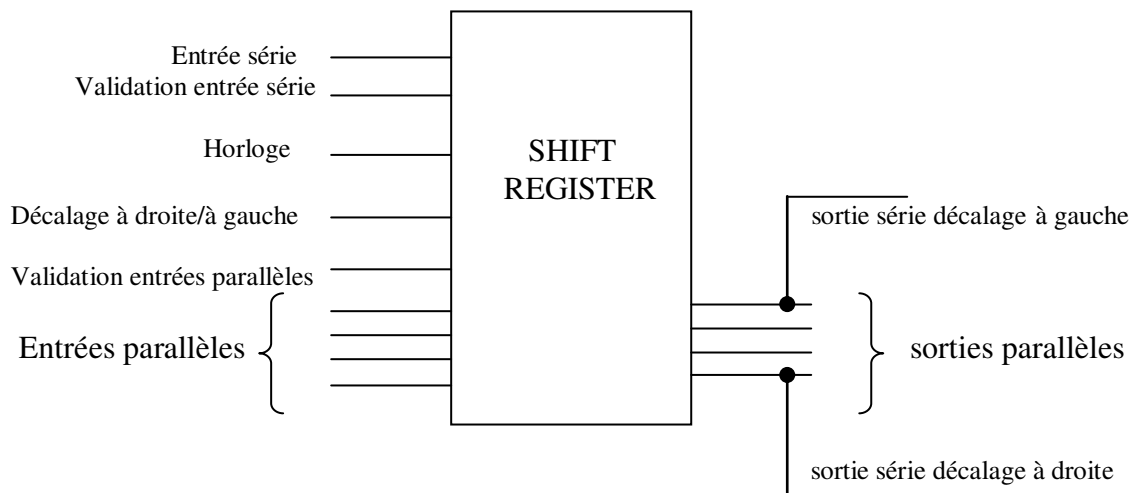
- entrée en série (serial-in).
- Entrée en parallèle (parallel-in).
- Sortie en série (serial-out).
- Sortie en parallele (parallel-out).

3- Registre à décalage à chargement série :



Q ₀	Q ₁	Q ₂	Q ₃	
0	0	0	0	état initial
D _t	0	0	0	1 ^{ère} impulsion d'horloge
D _{t+1}	D _t	0	0	2 ^{ème} impulsion d'horloge
D _{t+2}	D _{t+1}	D _t	0	3 ^{ème} impulsion d'horloge
D _{t+3}	D _{t+2}	D _{t+1}	D _t	4 ^{ème} impulsion d'horloge

Pour les registres à décalage bidirectionnel, des entrées supplémentaires sont nécessaires comme par exemple la commande pour le décalage à gauche ou à droite.

**4- Manipulation :**

- Réaliser un registre de 4 bits à entrées parallèles en utilisant des bascules D.
- Réaliser un registre à décalage à droite de 4 bits à entrée série.
- Donner le circuit d'un registre à décalage à gauche.

Annexe 1

Fonctions de base et leurs symboles

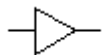
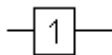
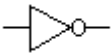
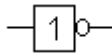

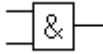

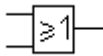



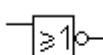

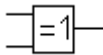

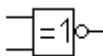
SYMBOLE (Norme MILSTD 086B)	NOM ET ÉQUATION	SYMBOLE (notation française)
	OUI (non inverseur) $a = a$	
	INVERSEUR $a = \bar{a}$	
	ET $s = a.b$	
	OU $s = a+b$	
	NON-ET $s = \overline{a.b} = \bar{a}.\bar{b}$	
	NON-OU $s = \overline{a+b} = \bar{a}.\bar{b}$	
	OU EXCLUSIF $s = \overline{a.b} + \bar{a}b$ ou $s = a \oplus b$	
	NON-OU EXCLUSIF $s = \overline{a \oplus b}$	

Tableau regroupant les fonctions de base et leurs symboles

Annexe 2

Brochage des circuits intégrés TTL

Issued July 1984 5459



74 Series logic families

This data sheet gives the pin connections and availability for the four 74 series logic families offered.

Standard 74 Series TTL

A range of popular transistor – transistor logic integrated circuits for use in basic circuits. 5Vdc supply, 0°C to +70°C operating temperature range.

Low power Schottky 74 LS Series TTL

A Schottky process using shallower diffusions yields devices with a five fold decrease in power consumption and an increase in speed compared to standard 74 TTL. 5Vdc supply, 0°C to +70°C operating temperature range.

Advance low power Schottky 74 ALS TTL

Advanced low power Schottky TTL devices are directly compatible with LS and standard TTL devices. An advanced oxide isolated construction uses small geometries giving approximately twice the speed with half the power consumption of 74 LS TTL. 5Vdc supply, 0°C to +70°C operating temperature range.

High speed CMOS 74 HCT

High speed CMOS devices are fabricated using the latest technology oxide isolated CMOS process giving good speed performance comparable with LS TTL but with much lower power consumption of the same order as 4000 CMOS.

74 HCT is offered with directly TTL compatible inputs, outputs are compatible with TTL and CMOS.

74 HCT types function with a 4.5 to 5.5Vdc supply, operating temperature range being a wide –40°C to +85°C.

Connections shown are top view. A 'negation' circle at any output or input within the schematic indicates that the terminal is active LOW or at clocking inputs the device is negative edge triggered.

Unused inputs should be connected to the appropriate defined logic level in order to achieve output conditions in line with the device truth table. As with standard CMOS high speed CMOS unused inputs must be connected to defined logic levels. For active HIGH inputs with standard TTL and ALS TTL a pull up resistor to V_{CC} should be used; up to 25 unused inputs can be connected to each resistor. With LS TTL and HCT unused active HIGH inputs can be directly tied to +ve supply so long as the connecting leads are short and the supply is adequately decoupled. Unused active LOW inputs can be directly connected to ground with all families.

Abbreviations used throughout this data sheet

A, B, C, D and E	Data inputs binary weight (where applicable) A = 1; B = 2; C = 4; D = 8; E = 16
a, b, c, d etc	Segment outputs on 7-segment decoder driver
BCD	Binary Coded Decimal
BI	Blanking Input
C_{in, out}	Carry in or out
CEP	Count Enable Parallel Input
CER	Count Enable Ripple Input
CK	Clock
CS	Chip select
D, JK	Data input to flip-flops
EN	Enable
GND	Ground 0V terminal
I/O	Input/Output
LT	Lamp test
MR	Master Test
OEN	Output enable
PE	Parallel Enable (active low) Input
Q	Output, may have a letter indicating weighing
RBI	Ripple Blanking Input
RBO	Ripple Blanking Output
RC, C, R	Capacitor and Resistor timing on monostables
RCO	Ripple carry output
S	Sum Output
SDL	Serial data in Left Shift
SDR	Serial data in Right Shift
SQ	Serial Output
SR	Synchronous reset
TC	Terminal Count Output
V_{cc}	+ Supply terminal
\mathcal{J}	Schmitt Device or function

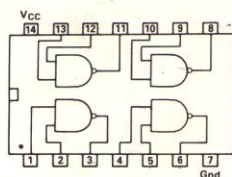


CAUTION: HCT devices

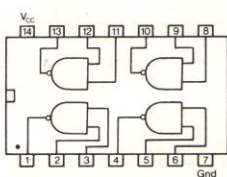
ESD (Electro-Static-Discharge) sensitive device. The digital inputs are diode protected; however, permanent damage may occur on unconnected devices subject to high energy electrostatic fields. Unused devices must be stored in conductive foam or shunts. The protective foam should be discharged to the destination socket before devices are removed.

5459

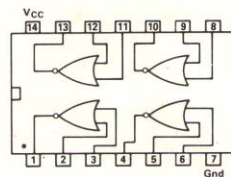
00 Quadruple 2-input NAND gate



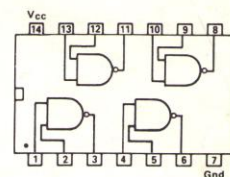
01 Quadruple 2-input NAND gate with open collector output



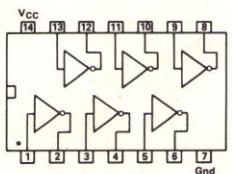
02 Quadruple 2-input NOR gate



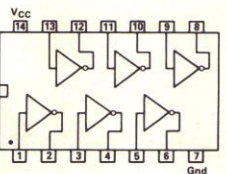
03 Quadruple 2-input NAND gate — open collector inputs



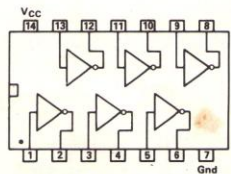
04 Hex inverter



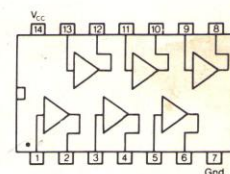
05 Hex inverter-open collector outputs



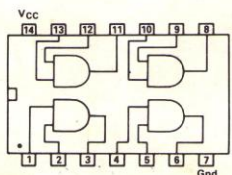
06 Hex inverter with high voltage open collector output



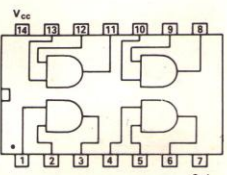
07 Hex driver with open collector output



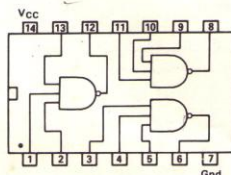
08 Quadruple 2-input AND gate



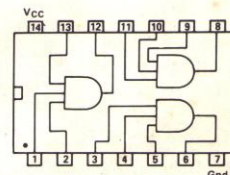
09 Quad 2-input AND gate-open collector outputs



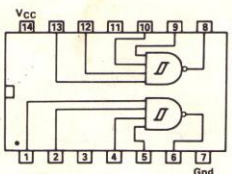
10 Triple 3-input NAND gate



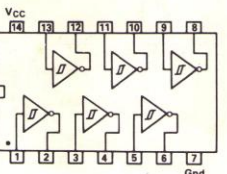
11 Triple 3-input AND gate



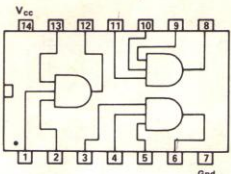
13 Dual 4-input NAND gate Schmitt trigger



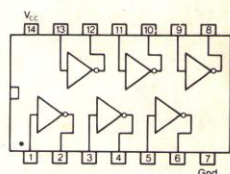
14 Hex Schmitt Trigger



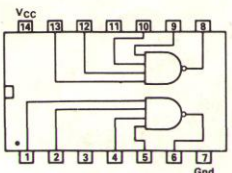
15 Triple 3-input AND gate — open collector outputs



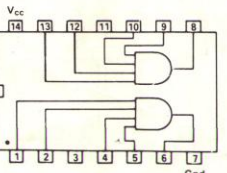
16 Hex Inverter with open collector output



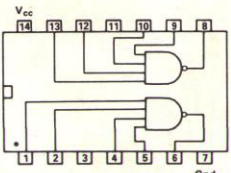
20 Dual 4-input NAND gate



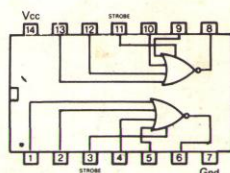
21 Dual 4-input AND gate



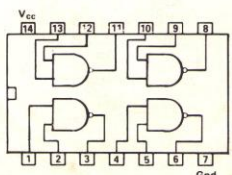
22 Dual 4-input NAND gate — open collector outputs



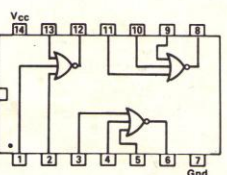
25 Dual 4-input NOR gate with strobe



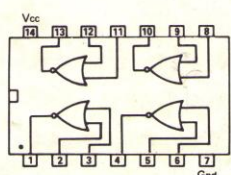
26 Quad 2-input NAND buffer-open collector outputs



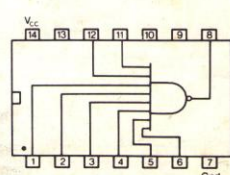
27 Triple 3-input NOR gate



28 Quad 2-input NOR buffer

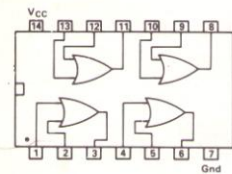


30 8-input NAND gate

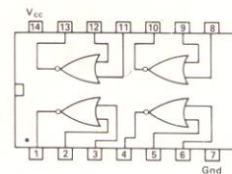


5459

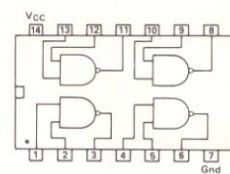
32 Quaduple 2-input OR gate



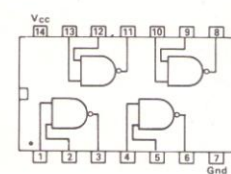
33 Quad 2-input NOR buffer-open collector outputs



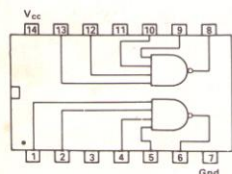
37 Quaduple 2-input NAND buffer



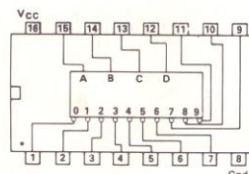
38 Quaduple 2-input NAND buffer — open collector outputs



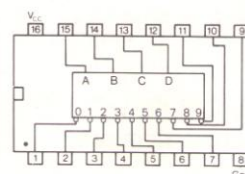
40 Dual 4-input NAND buffer



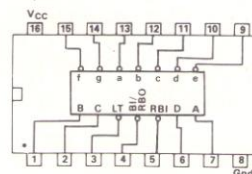
42 BCD-to-decimal decoder



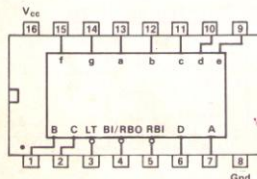
45 BCD-to decimal decoder/driver



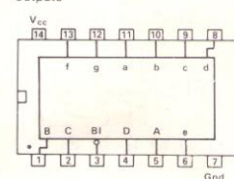
47 BCD-to-7 segment decoder/driver — open collector outputs



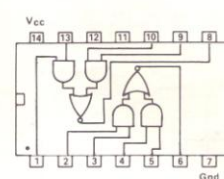
48 BCD-to-7 segment decoder/driver



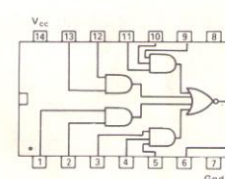
49 BCD-to-7-segment decoder/driver — open collector outputs



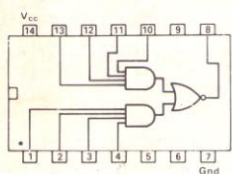
51 Dual 2-wide 2-input/3-input AND-OR-INVERT gate



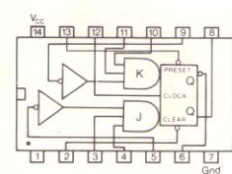
54 3-2-2-3 input AND-OR-INVERT gate



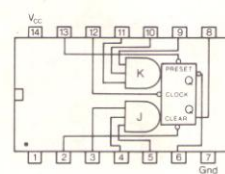
55 2-wide 4-input AND-OR-INVERT gate



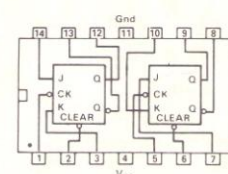
70 J-K flip-flop



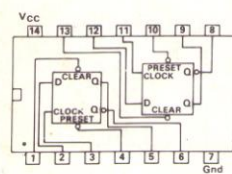
72 J-K master-slave flip-flop



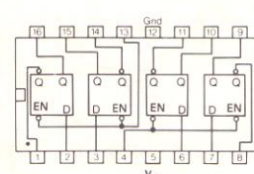
73 Dual JK negative edge-triggered Flip-Flop



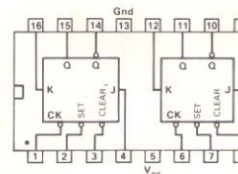
74 Dual D-type edge-triggered Flip-Flop



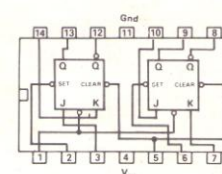
75 4-bit D Latch



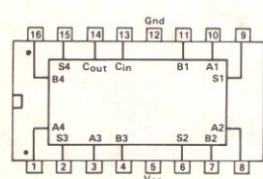
76 Dual JK Flip-Flop with set and clear



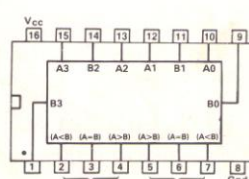
78 Dual JK Flip-Flop



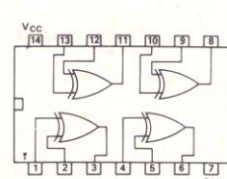
83A 4-bit Binary full adder



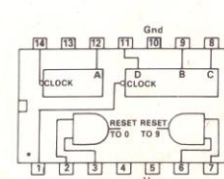
85 4-bit magnitude comparator



86 Quaduple 2-input exclusive OR gate

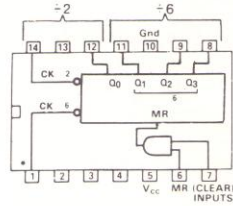


90 Decade counter

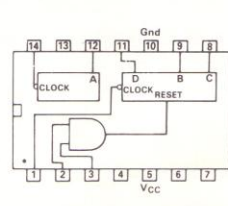


5459

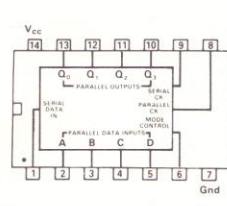
92 Divide-by-twelve counter



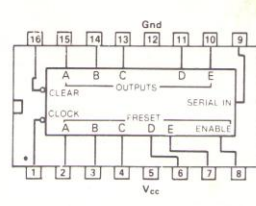
93 4-bit binary counter



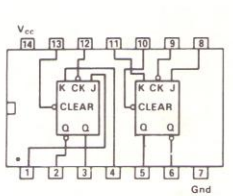
95B 4-bit shift register



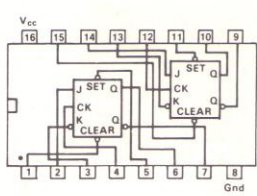
96 5-bit shift register



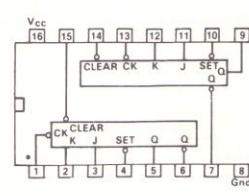
107 Dual JK Flip-Flop



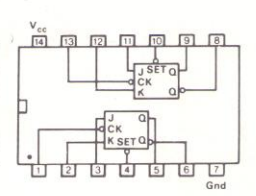
109 Dual JK positive edge-triggered Flip-Flop



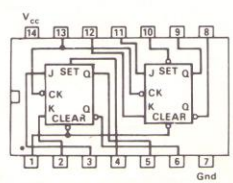
112 Dual JK edge triggered flip-flop



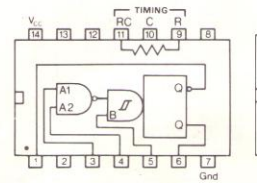
113 Dual JK negative edge-triggered Flip-Flop



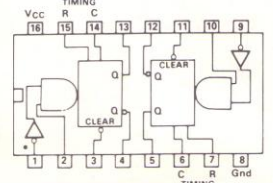
114 Dual JK negative edge-triggered Flip-Flop



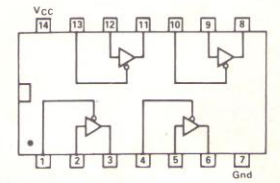
121 Monostable multivibrator



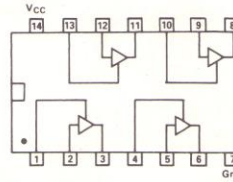
123 Dual monostable - retriggerable



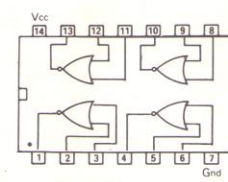
125 Quad 3-state buffer (active low enable)



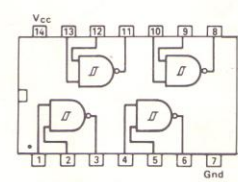
126 Quad 3-state buffer (active high enable)



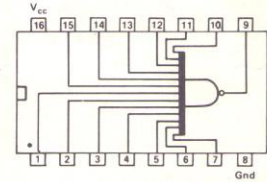
128 Quad line driver



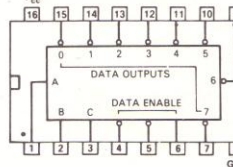
132 Quadruple 2-input NAND Schmitt gate



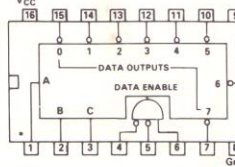
133 13-input NAND gate



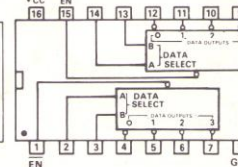
137 3-line to 8-line Decoder/Demultiplexer with address latches



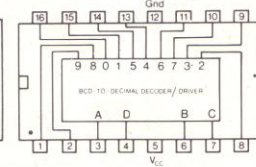
138 3 to 8 line Decoder/Multiplexer



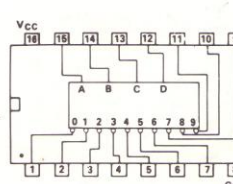
139 Dual 1 of 4 Decoder



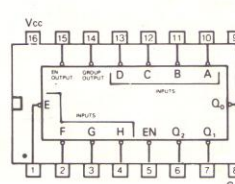
141 BCD-to-decimal decoder driver



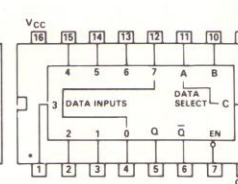
145 BCD-to-decimal decoder/driver



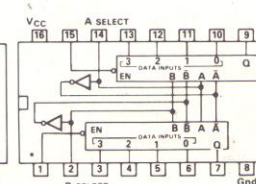
148 Octal priority encoder 8 line to 3 line

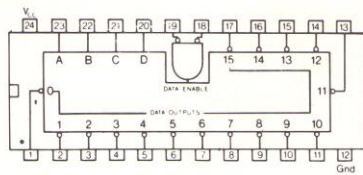
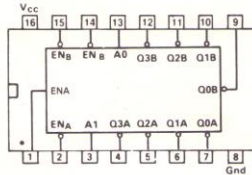
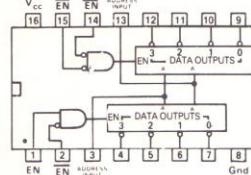
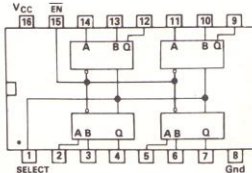
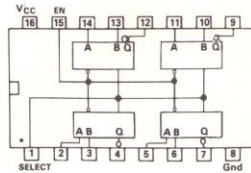
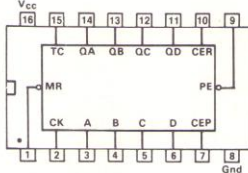
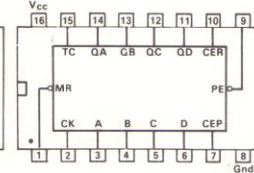
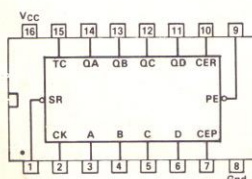
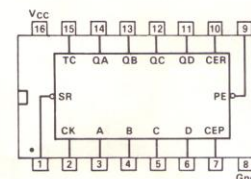
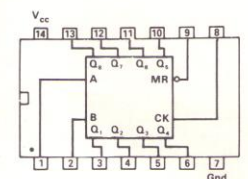
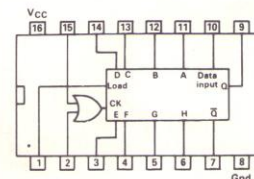
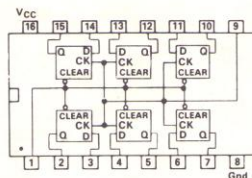
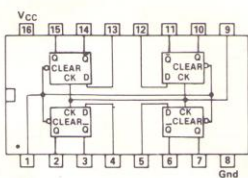
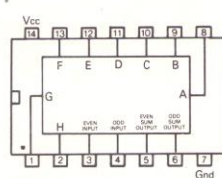
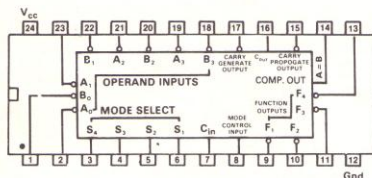
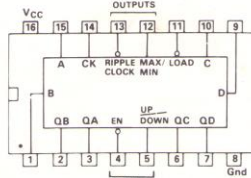
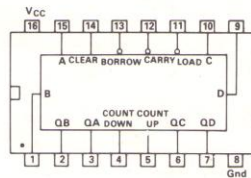
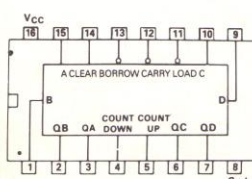
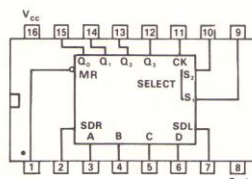
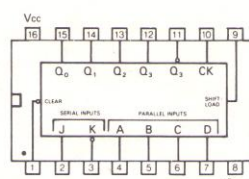
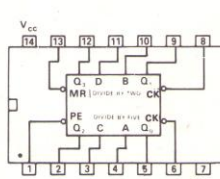


151 1 of 8 Data Selector/Multiplexer



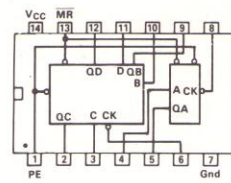
153 Dual 4-line to 1-line Data Selectors/Multiplexers



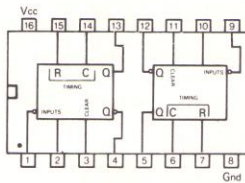
5459**154** 4 to 16 line Decoder**155** Dual 1 of 4 Decoder/Demultiplexer**156** Dual 1-of-4 Decoder/Demultiplexer with open collector outputs**157** Quad 2 to 1-line Data Selectors / Multiplexers**158** Quad 2 to 1-line Data selectors/Multiplexers with Inverted outputs**160** BCD decade counter — asynchronous reset**161** Binary counter — asynchronous reset**162** BCD counter — synchronous reset**163** Binary counter — synchronous reset**164** Serial-in parallel-out shift register**165** 8-bit parallel to serial converter register**174** Hex D-type Flip-Flops**175** Quad D-type Flip-Flops**180** Parity generator/checker 9-bit odd/even**181** 4-bit arithmetic logic unit**191** Binary synchronous up/down counter**192** Up/Down decade counter — with preset inputs**193** Up/Down binary counter-with preset inputs**194** A 4-bit bidirectional universal shift register**195** 4-bit parallel-access shift register**196** 4-stage presettable ripple counter

5459

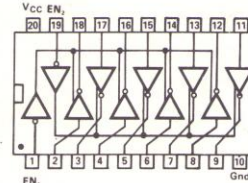
197 Presettable binary ripple counter



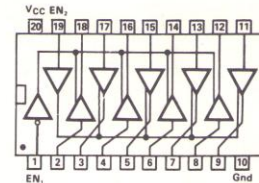
221 Dual monostable multivibrator



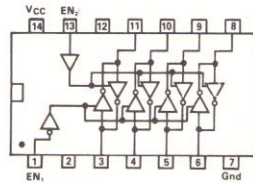
240 Octal buffer — three state inverting



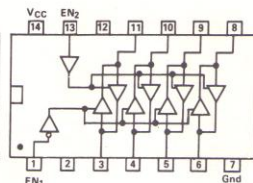
241 Octal buffer — three state non-inverting



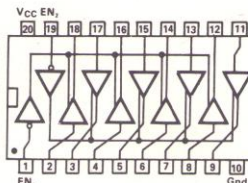
242 Quad bus transceiver — inverting



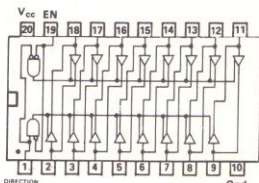
243 Quad bus transceiver — non-inverting



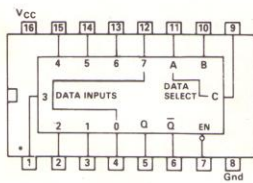
244 Octal buffer — three state non-inverting



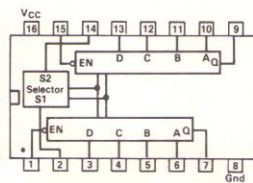
245 Octal bus transceiver with 3 state outputs



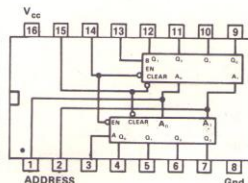
251 1 of 8 Data selector/Multiplexer with 3 state outputs



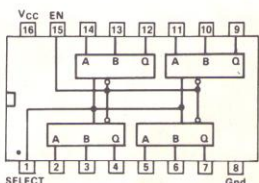
253 Dual 4 - input multiplexer with 3 state outputs



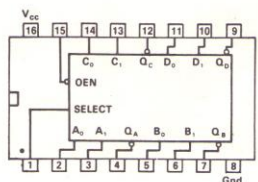
256 Dual 4-bit addressable latch



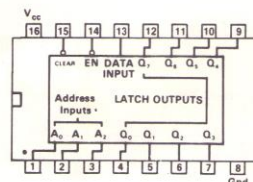
257 Quad 2-input multiplexer with 3 state outputs



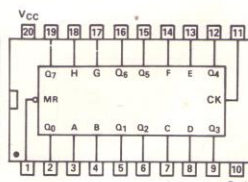
258 Quad 2-input multiplexer with 3 state outputs



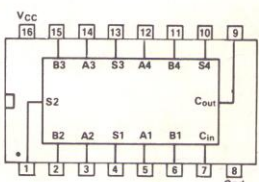
259 8-bit addressable latch



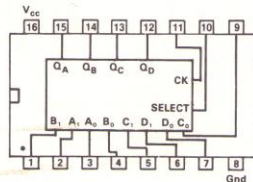
273 8-bit register with clear



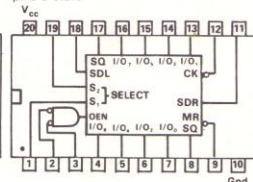
283 4-bit binary full adder



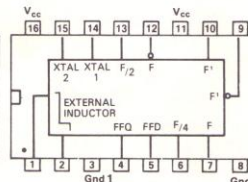
298 Quad 2-port register (Quad 2-input multiplexer with storage)



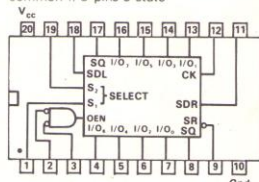
299 8-bit universal shift /storage register with common parallel I/O pins 3 state



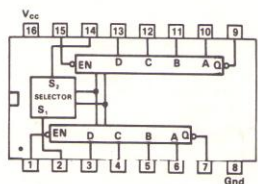
321 Crystal controlled oscillator



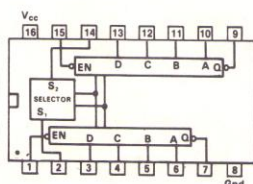
323 8-bit universal shift/storage register with synchronous reset and common I/O pins 3 state



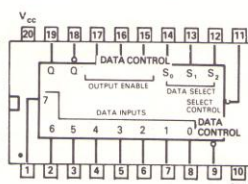
352 Dual 4-input multiplexer inverting



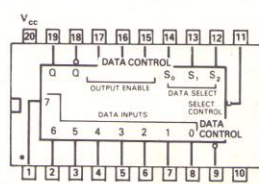
353 Dual 4-input multiplexer with 3 state outputs inverting



354 8-line to 1-line data selector/multiplexer/register

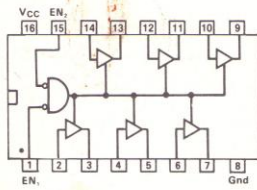


356 8-line to 1-line data selector/multiplexer/register

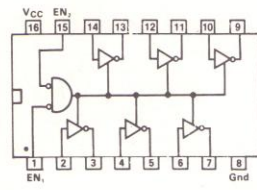


5459

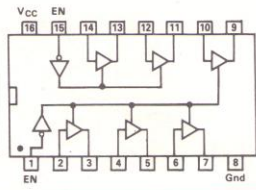
365 Hex 3-state buffer non-inverting



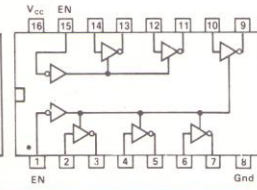
366 Hex 3-state buffer inverting



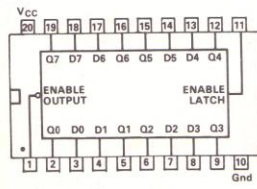
367 Hex 3-state buffer



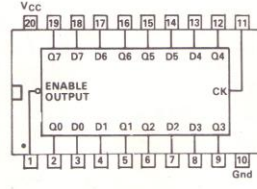
368 Hex 3-state inverter buffer (separate 2-bit & 4-bit sections)



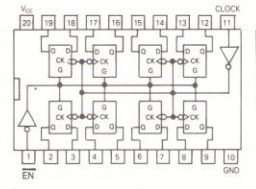
373 Octal transparent latch with 3 state outputs



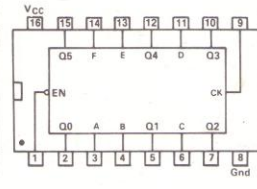
374 Octal D-type flip-flop with 3 state outputs



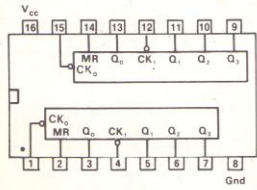
377 Octal D-type flip-flop with enable



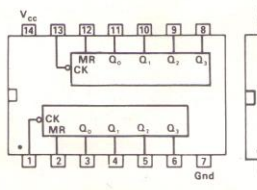
378 Hex D register



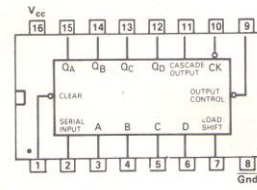
390 Dual decade counter



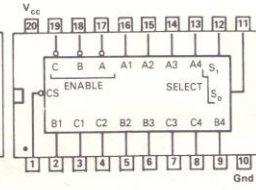
393 Dual 4 stage binary counter



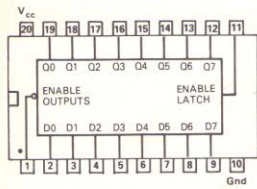
395 4-bit cascable shift register 3 state



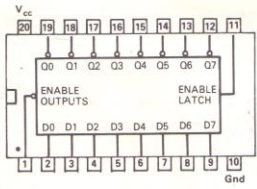
442, 443, 444 Quad tridirectional bus transceivers 3 state



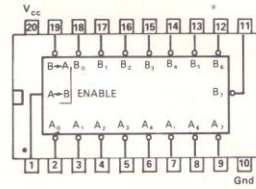
573 Octal D-type transparent latch



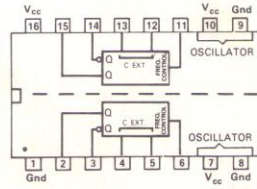
580 Octal D-type transparent latch inverted outputs



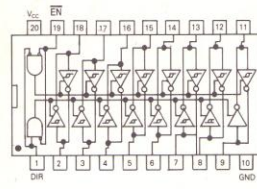
620 Octal bus transceiver



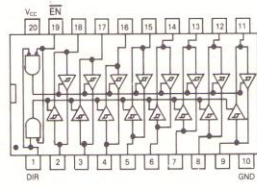
625 Voltage controlled oscillator



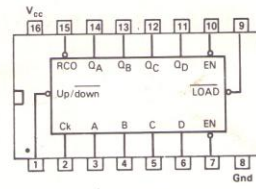
640 Tri-state, inverting octal bus transceiver



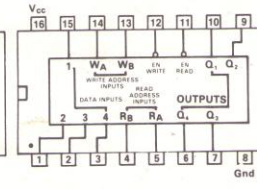
643 Tri-state, true and inverting octal bus transceiver



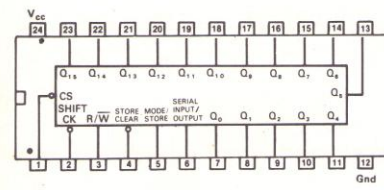
669 Up/down binary counter synchronous



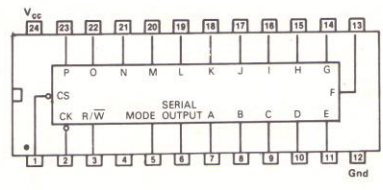
670 4 x 4 Register file with 3-state outputs



673 16-bit shift register, serial to parallel



674 16-bit shift register, parallel to serial



682 8-bit magnitude comparator

