Annexe I - Utilisation de l'analyseur logique

1.1 Fonctionnalités

L'analyseur logique permet d'envoyer une séquence sur les sorties numériques de la carte Arduino et de capturer les entrées numériques Arduino.

Il est également possible de visualiser la séquence/capture sous forme de table de transition ou de chronogrammes.

Par défaut, l'analyseur logique supporte :

- 8 entrées maximum
- 8 sorties maximum
- 24 échantillons pour une séquence ou capture

Ces valeurs sont modifiables dans le fichier « analyseur_logique.h ».

1.2 Les fonctions d'initialisation

Les prototypes des fonctions d'initialisation sont les suivantes :

```
void initEnregistrement(TypeEnregistrement &enr);
void initEnregistrement(TypeEnregistrement &enr, int periodeMilliSeconde);
void initEnregistrement(TypeEnregistrement &enr, int periodeMilliSeconde,
int echelle);
void initEnregistrement(TypeEnregistrement &enr, int periodeMilliSeconde,
int echelle, boolean jusquaLaFin);

void changePeriode(TypeEnregistrement &enr, int periodeMilliSeconde);
void changeEchelle(TypeEnregistrement &enr, int echelle);
void changeAffichageEchantillons(TypeEnregistrement &enr, boolean
jusquaLaFin);
```

Par défaut, les paramètres de l'analyseur logique sont configurés comme suit :

- *echelle* : Echelle ou zoom pour l'affichage des chronogrammes par défaut : 4 (valeurs possible entre 1 et 6)
- *periodeMilliSeconde* : Période par défaut : 10 ms
- *jusquaLaFin* : Par défaut, le nombre d'échantillon de sorties = nombre échantillon d'entrée. Si

Ceux-ci sont modifiables à l'initialisation (fonction *initEnregistrement()*) et à l'aide des fonctions *changePeriode()*, *changeEchelle()* et *changeAffichageEchantillons().*

```
void ajouteEntree(TypeEnregistrement &enr, int patte, char* nom, byte*
valeurs);
void ajouteEntree(TypeEnregistrement &enr, int patte, char* nom, byte*
valeurs, int repeter);
void ajouteSortie(TypeEnregistrement &enr, int patte, char* nom);
```

Exemple de code d'initialisation :

```
#include "analyseur_logique.h"

/* Définition des E/S */
const int PIN_B=2; // sortie Arduino MAIS entrée du circuit à tester
const int PIN_Q=4; // entrée Arduino MAIS sortie du circuit à tester
```

```
/* Déclaration de l'analyseur Logique */
TypeEnregistrement analyseur;

void setup() {

     pinMode(PIN_B, OUTPUT);
     pinMode(PIN_Q, INPUT);

     initEnregistrement(analyseur, 60, 4, true);

     /* Ajoute une entrée dont la séquence est répétée 3 fois
     * Le premier nombre correspond au nombre d'échantillons, ici 2
     */
     byte seqB[5] = {2, 0,1};
     ajouteEntree(analyseur, PIN_B, "B", seqB, 3);

     /* Ajoute une sortie */
     ajouteSortie(analyseur, PIN_Q, "Q");
}
```

1.3 Les fonctions de séquence et capture

L'exemple suivant illustre l'utilisation des fonctions permettant de jouer la séquence et de capturer les échantillons :

```
void loop() {

     /* Envoi un à un les échantillons et capture les échantillons
      * du scenario.
      * La fonction renvoi « vrai » dès que le scénario est terminé
      */
     while (executeScenario(analyseur) == false);

     /* Affichage de la Table */
     Serial.println("Table de transitions");
     affichageTable(analyseur);

     /* Affichage des chronogrammes */
     Serial.println("Chronogrammes");
     affichageChronogrammes(analyseur);

     /* On recommence le scenario */
     Serial.println("Recommence le scenario");
     recommenceScenario(analyseur);
}
```

Pour visualiser la table et les chronogrammes, il faut ouvrir la console série. Pour ce faire, aller dans le menu « Outils », puis « Moniteur série ».

Les prototypes des fonctions

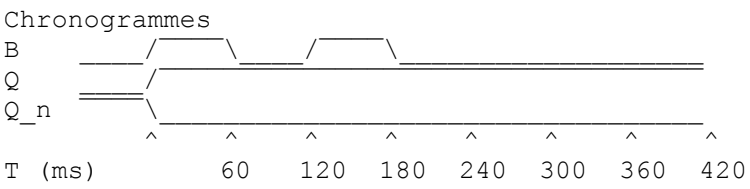
```
boolean executeScenario(TypeEnregistrement &enr);
void recommenceScenario(TypeEnregistrement &enr);

void affichageTable(TypeEnregistrement enr);
void affichageChronogrammes(TypeEnregistrement enr);
```

Exemple de table de transition

Table de transitions

B	Q	Q_n
0	0	1
1	1	0
0	1	0
1	1	0
0	1	0

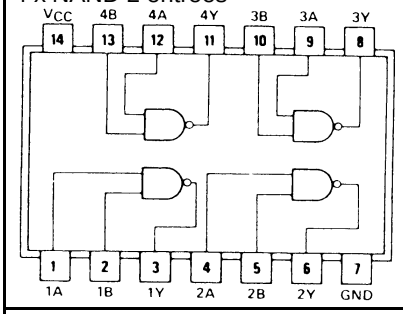
Exemple de chronogrammes :



Annexe II - Brochage Circuits Intégrés TTL

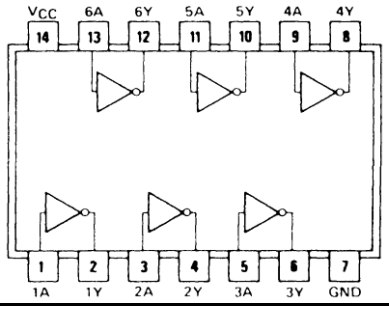
7400 — $Y = \overline{AB}$

4 x NAND 2 entrées



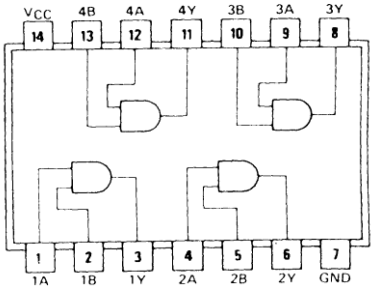
7404 — $Y = \overline{A}$

6 x Inverseurs



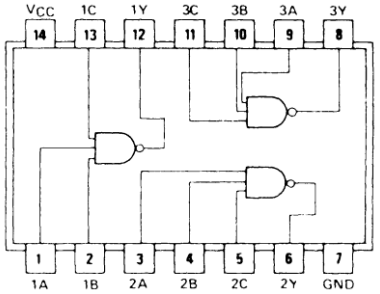
7408 — $Y = AB$

4 x AND 2 entrées



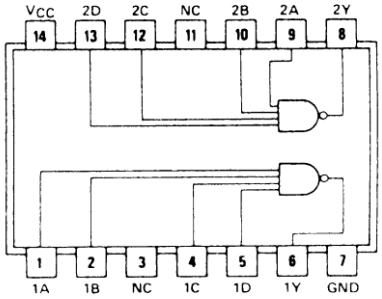
7410 — $Y = \overline{ABC}$

3 x NAND 3 entrées



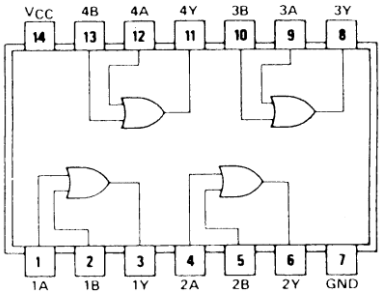
7420 — $Y = \overline{ABCD}$

2 x NAND 4 entrées

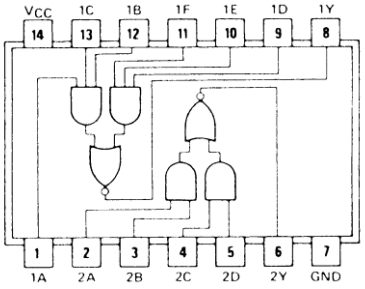


7432 — $Y = A + B$

4 x OR 2 entrées

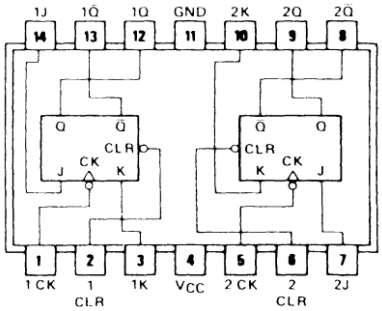


7451 — $1Y = \overline{AB + CD}$



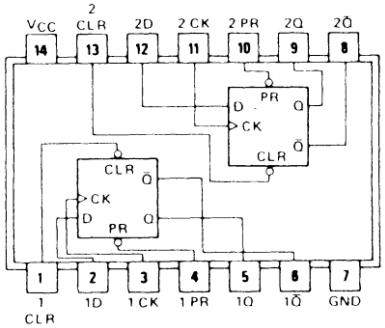
7473

2 x JK Flip-Flop avec reset



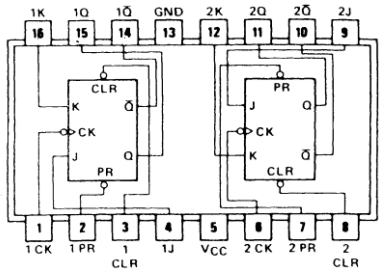
7474

2 x D Flip-Flop



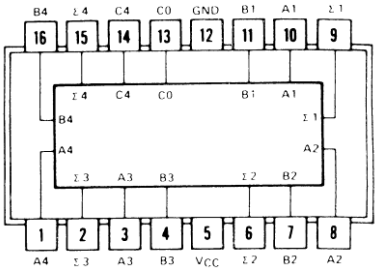
7476

2 x JK Flip-Flop avec set/reset



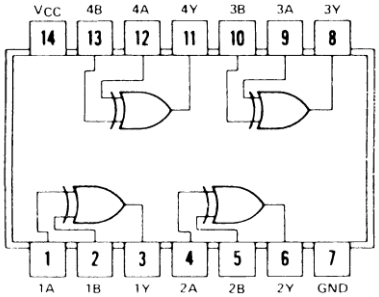
7483

Additionneur 4 bits

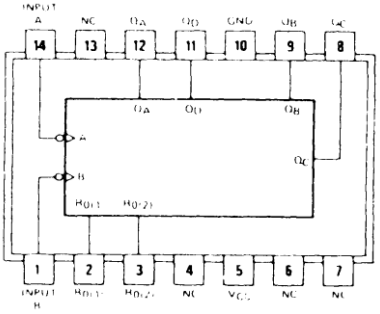


7486 — $Y = A \oplus B$

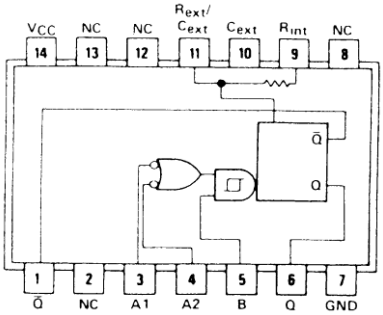
4 x XOR 2 entrées



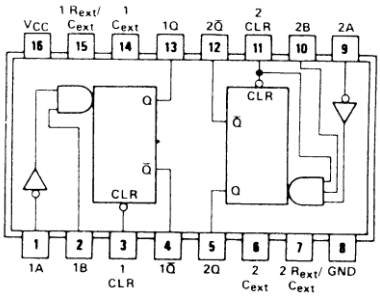
7493
compteur binaire 4 bits



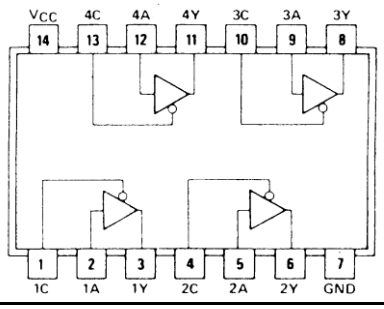
74121
monostable



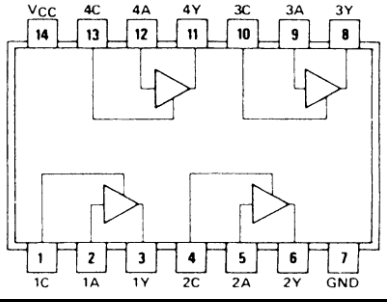
74123
monostable redéclenchable



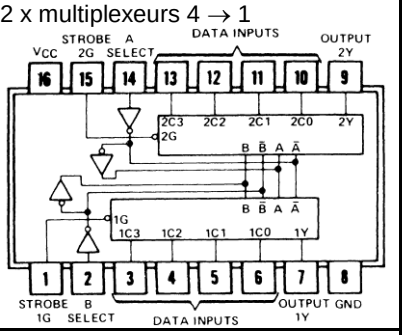
74125 $Y = A$ when C=0
4 x buffers 3 états (tristate)



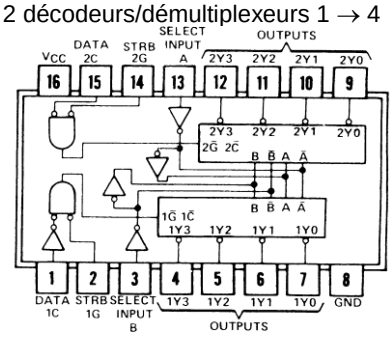
74126 $Y = A$ when C=1
4 x buffers 3 états (tristate)



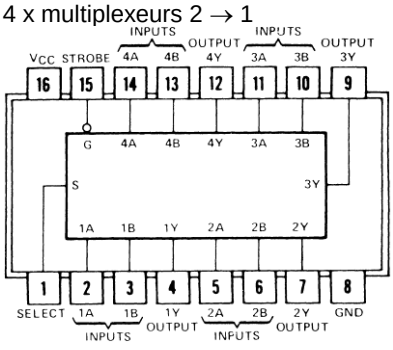
74153
2 x multiplexeurs 4 → 1



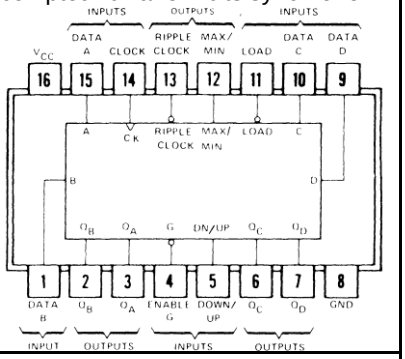
74155
2 décodeurs/démultiplexeurs 1 → 4



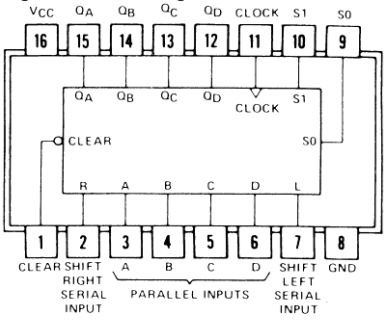
74157 $Y = \overline{A}$
4 x multiplexeurs 2 → 1



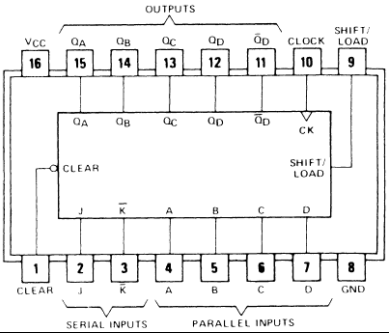
74191
compteur binaire 4 bits synchrone



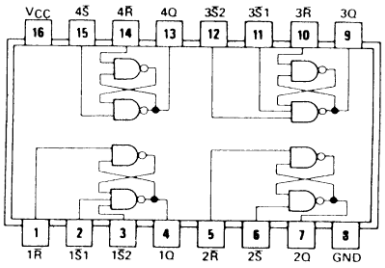
74194
registre à décalage bidir. 4 bits



74195
registre à décalage 4 bits



74279
4 x RS



Annexe III - Compléments Circuits Intégrés

III.1 Circuit 74123

Le Schéma

Figure 1 is a functional block diagram of the 'LS123. Each one-shot has two inputs, one active-low and one active-high, which allow both leading or trailing edge triggering. When triggered, the basic pulse duration can be extended by retriggering the gated low-level active A or high-level active B inputs, or the pulse duration can be reduced by use of the overriding clear. Therefore, an input cycle time shorter than the output cycle time will retrigger the 'LS123 and result in a continuously high Q output.

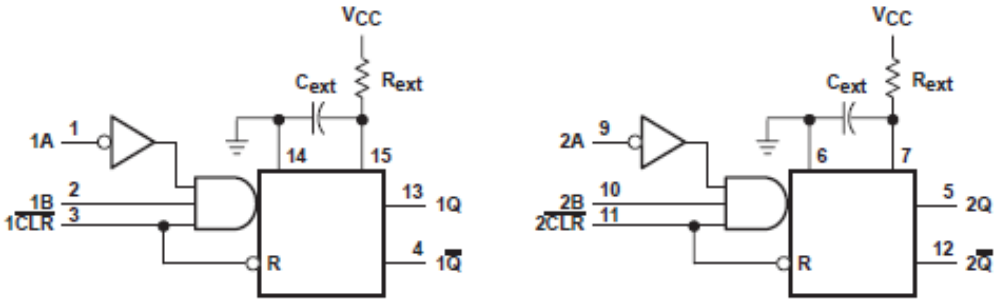


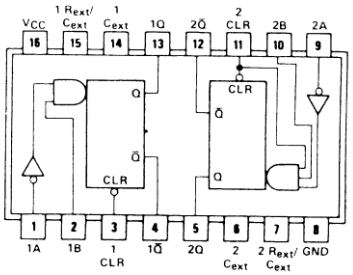
Figure 1. 'LS123 Logic Diagram

La table de vérité

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



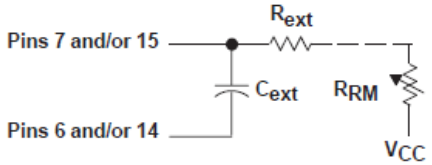
SN54123/SN74123(J, N, W)
SN54L123/SN74L123(J, N)

Les règles à respecter

1. An external resistor (R_{ext}) and an external capacitor (C_{ext}) are required, as shown in Figure 1, for proper circuit operation.

NOTE:
For best results, system ground should be applied to the C_{ext} terminals.

2. This value of R_{ext} may vary from 5 kΩ to 180 kΩ between −55°C and 125°C.
3. C_{ext} may vary from 0 pF to any necessary value.
4. The input may have a minimum amplitude of −0.5 V and a maximum of 5.5 V.
5. When an electrolytic capacitor is used as C_{ext}, the switching diode required by most one-shots is *not needed* for 'LS123 operation.
6. For remote trimming, the circuit shown in Figure 2 is recommended.



NOTE: R_{RM} is placed as close as possible to the 'LS123.

Figure 2. Remote Trimming Circuit

7. The retrigger pulse duration is calculated as shown in Figure 3.

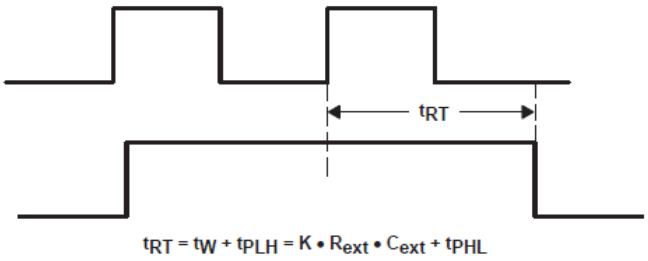


Figure 3. Retrigger Pulse-Duration Calculation

Output Pulse Duration

The basic output pulse duration is essentially determined by the values of external capacitance and timing resistance. For pulse durations when C_{ext} is < 1 µF, use the following formula:

$$t_w = K \cdot R_t \cdot C_{ext} \text{ (also see Figure 5)} \quad (1)$$

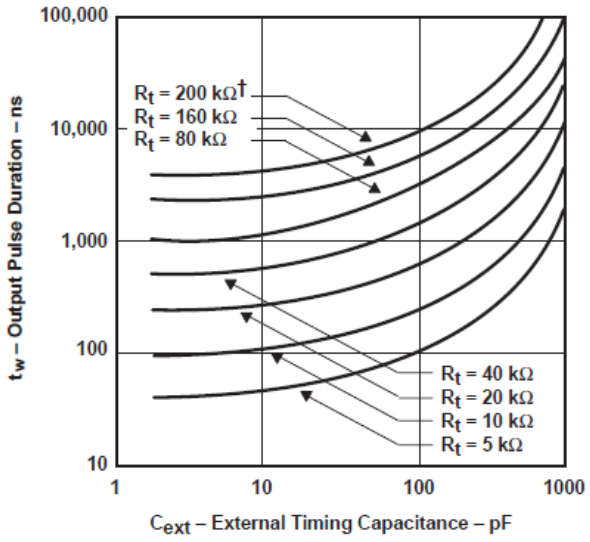
When C_{ext} is > 1 µF, the output pulse duration is defined as:

$$t_w = 0.33 \cdot R_t \cdot C_{ext} \quad (2)$$

Where, for the two previous equations, as applicable:

K = multiplier factor
R_t = given in kΩ (Internal or External Timing Resistance)
C_{ext} = in pF
t_w = in ns

For capacitor values of less than 1000 pF, the typical curves in Figure 5 can be used.



† This value of resistance exceeds the maximum recommended for use over the full temperature range of the SN54LS circuits.

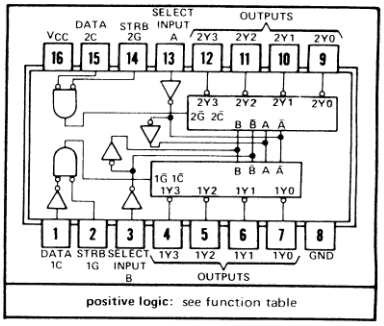
Figure 5. Output Pulse Duration Versus External Timing Capacitance

III.2 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 3 GATE LEVELS	TYPICAL POWER DISSIPATION
'155, '156	21 ns	125 mW
'LS155	18 ns	31 mW

J OR N DUAL-IN-LINE OR
W FLAT PACKAGE (TOP VIEW)

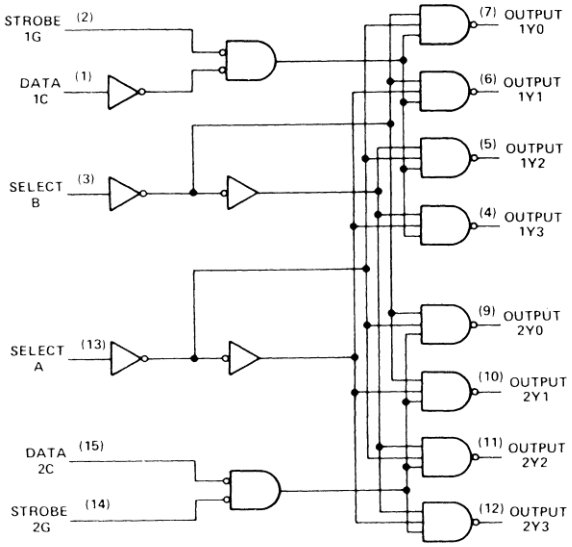


description

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°C to 125°C; Series 74 and 74LS are characterized for operation from 0°C to 70°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	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
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

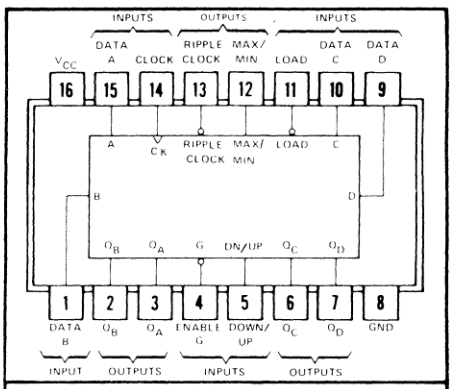
†C = inputs 1C and 2C connected together
‡G = inputs 1G and 2G connected together
H = high level, L = low level, X = irrelevant

III.3 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 Presettable with Load Control
- Parallel Outputs
- Cascadable for n-Bit Applications

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

J OR N DUAL-IN-LINE
OR W FLAT PACKAGE (TOP VIEW)



asynchronous inputs: Low input to load sets Q_A=A, Q_B = B, Q_C = C, and Q_D = D

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.

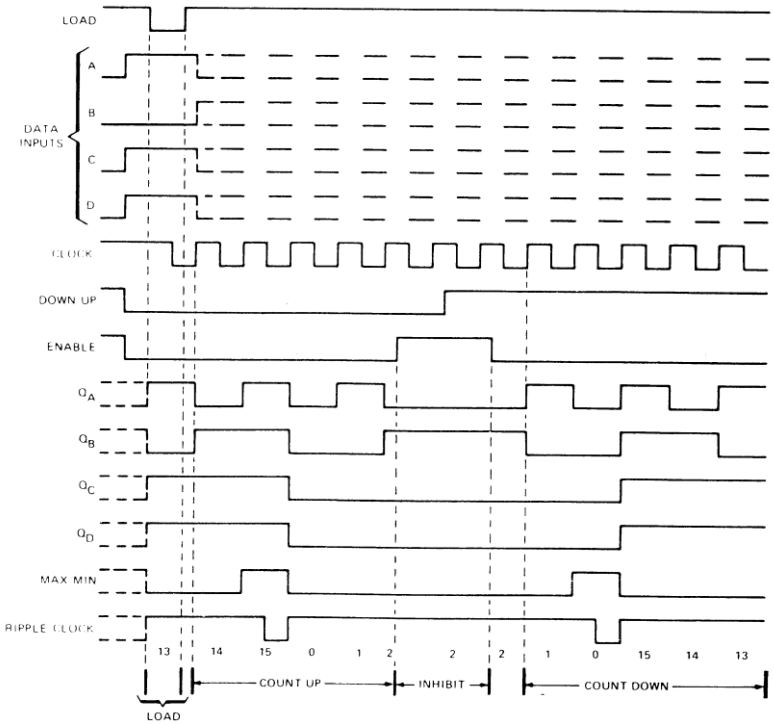
The clock, down/up, and load inputs are buffered to lower the drive requirement which significantly reduces the number of clock drivers, etc., required for long parallel words.

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.

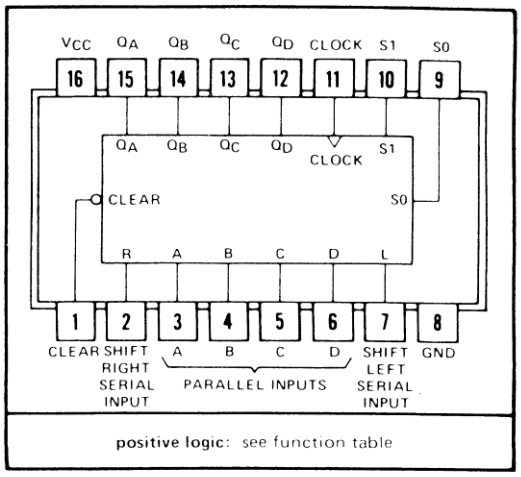


III.4 Circuit 74194

- 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 MAXIMUM CLOCK FREQUENCY	TYPICAL POWER DISSIPATION
'194	36 MHz	195 mW
'LS194	28 MHz	60 mW
'S194	105 MHz	425 mW

J OR N DUAL-IN-LINE OR
W FLAT PACKAGE (TOP VIEW)



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

INPUTS										OUTPUTS			
CLEAR	MODE		CLOCK	SERIAL		PARALLEL				Q_A	Q_B	Q_C	Q_D
	S_1	S_0		LEFT	RIGHT	A	B	C	D				
L	X	X	X	X	X	X	X	X	X	L	L	L	L
H	X	X	L	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