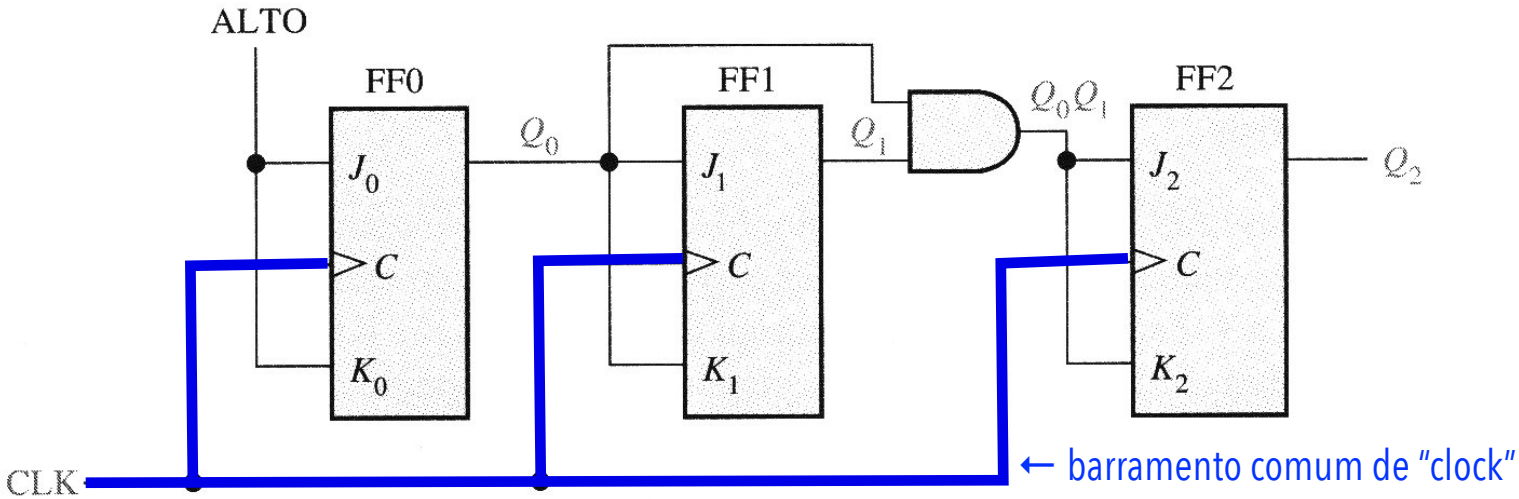
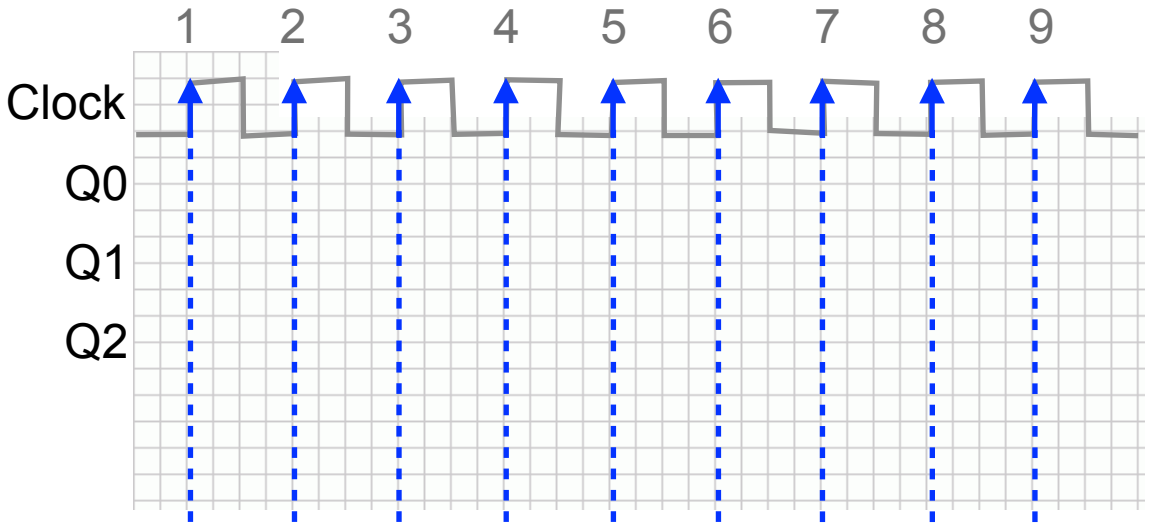
Contadores Síncronos

Prof. Fernando Passold

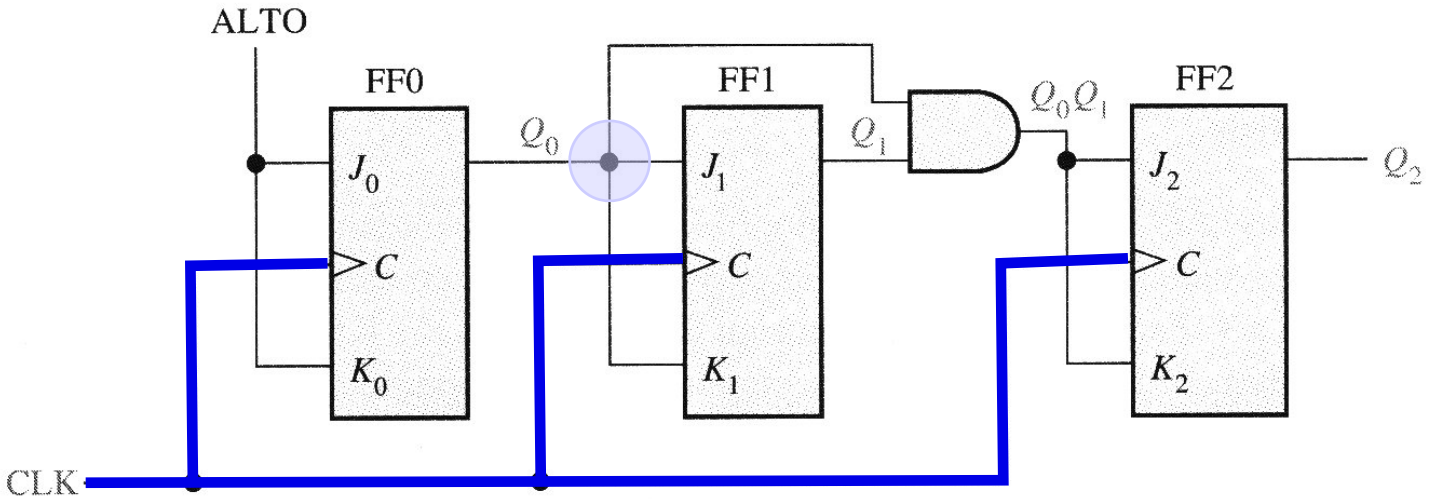
Oct/2008, Abr/2020, 2022.

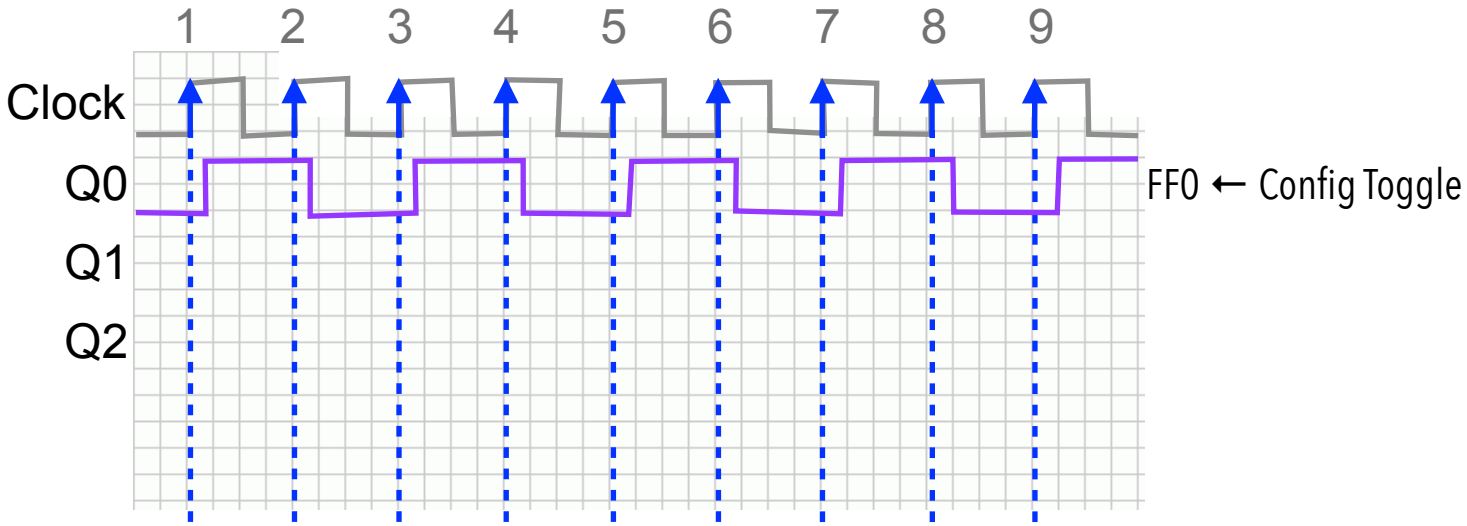
Contador **síncrono** binario de 3 bits:



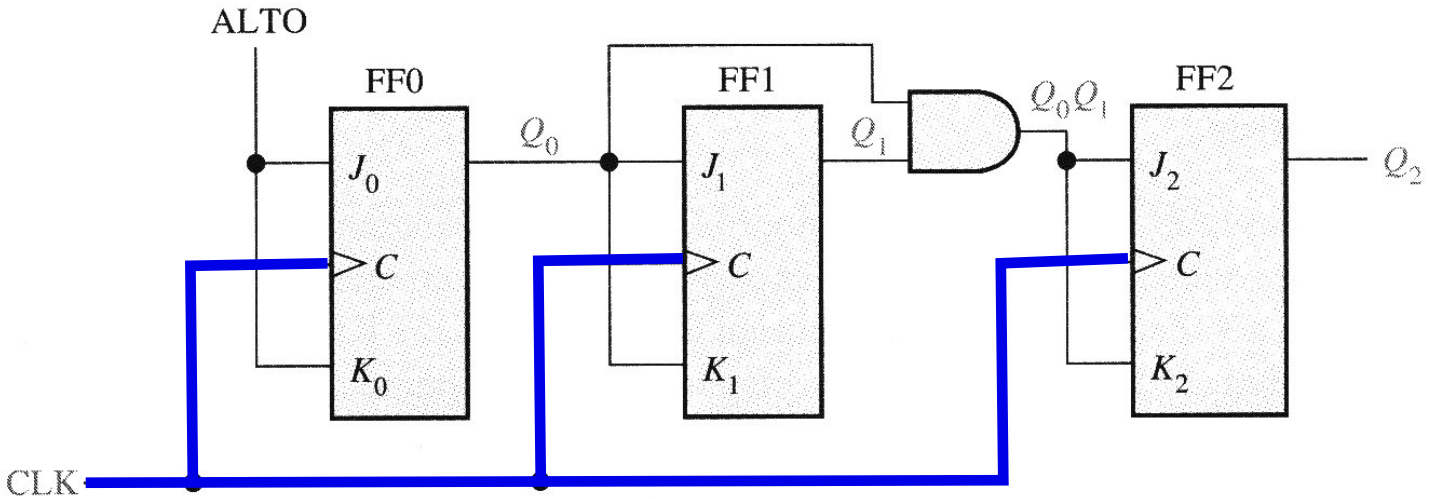


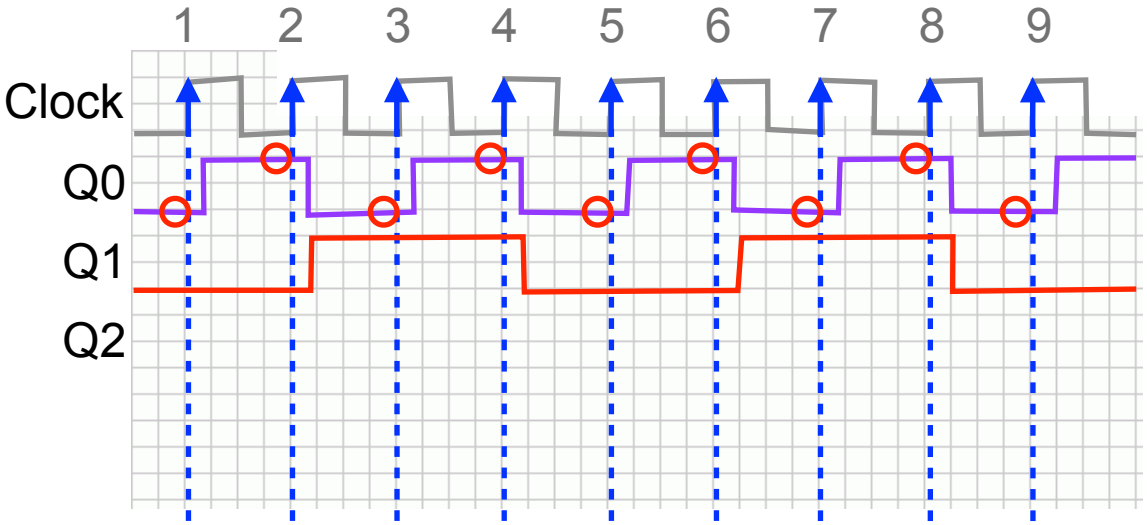
Contador **síncrono** binario de 3 bits:





Contador **síncrono** binario de 3 bits:

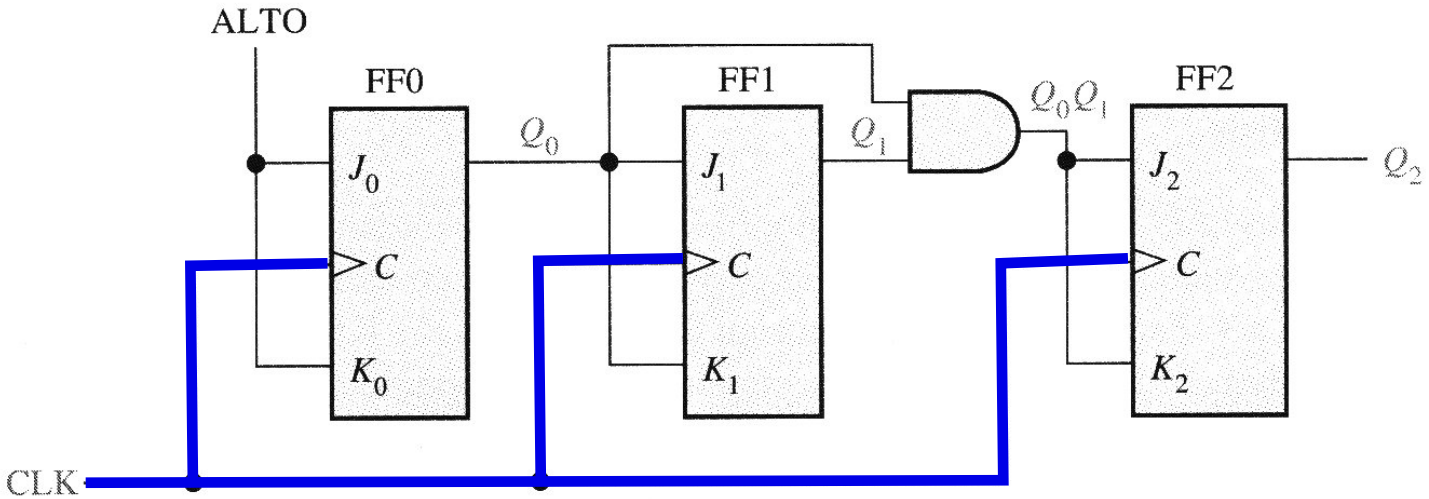


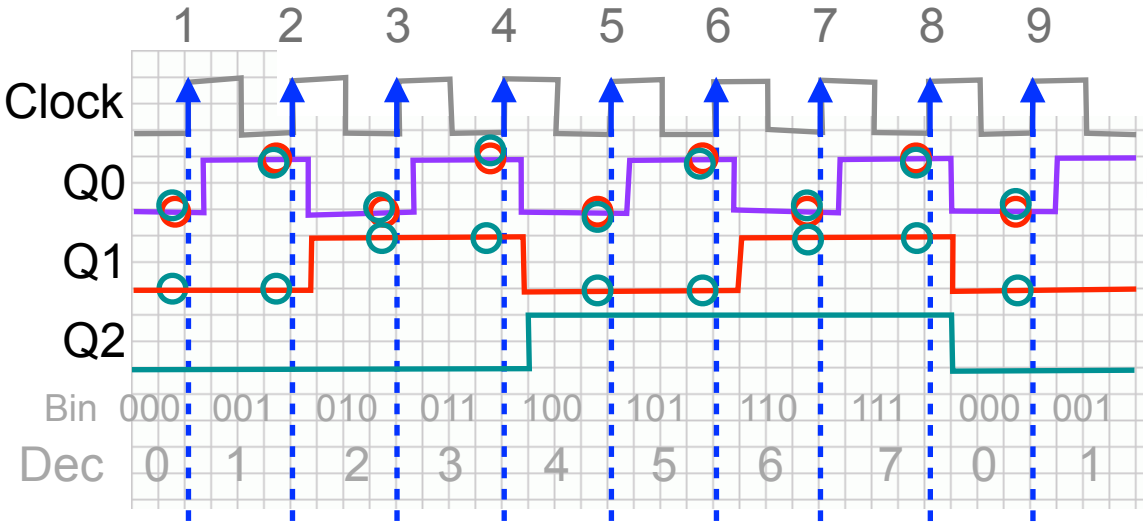


FF0 ← Config Toggle

FF1 ← "toggles" apenas qdo $Q_0 = 1$

Contador **síncrono** binario de 3 bits:



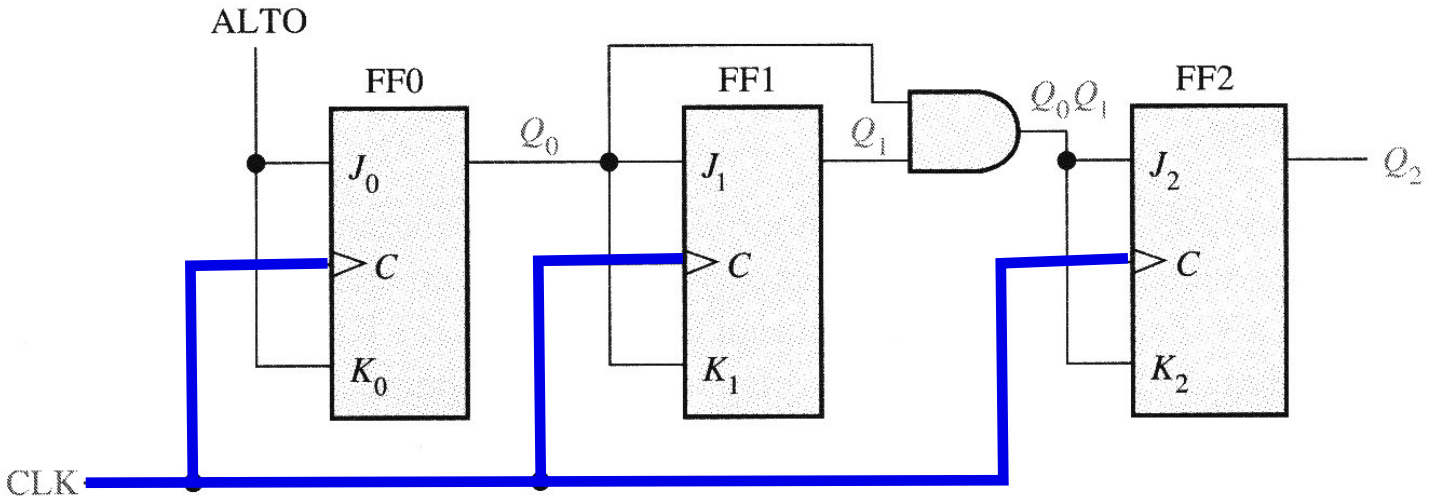


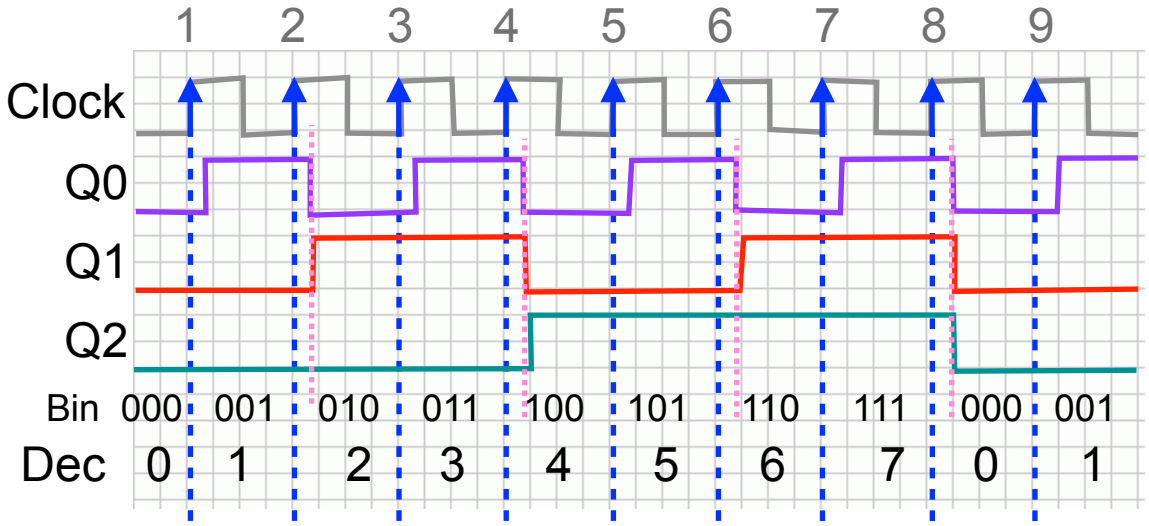
FF0 ← Config Toggle

FF1 ← "toggles" apenas qdo $Q_0 = 1$

FF2 ← "toggles" apenas qdo $Q_1 \cdot Q_0 = 1$

Contador **síncrono** binario de 3 bits:



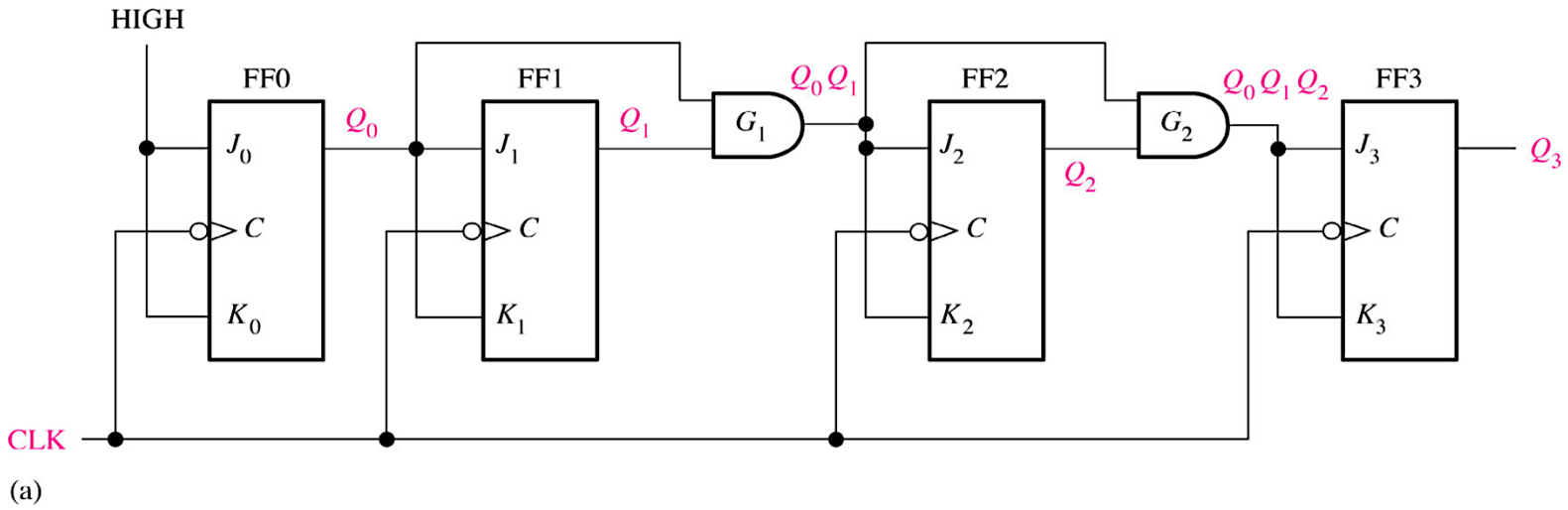


FF0 ← Config Toggle

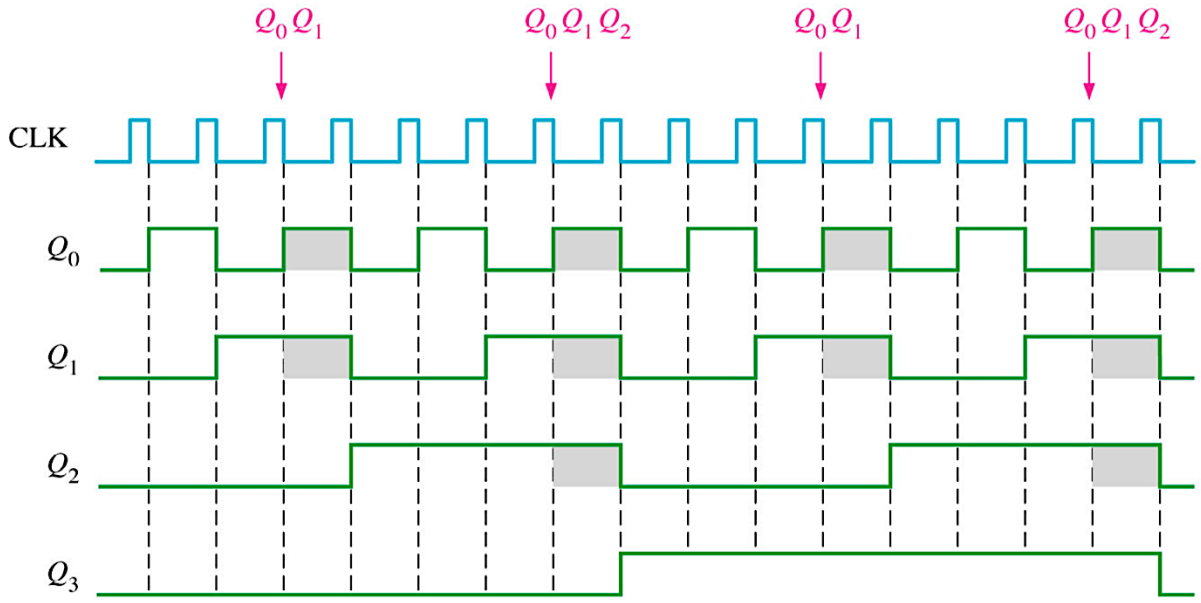
FF1 ← "toggles" apenas qdo $Q_0 = 1$

FF2 ← "toggles" apenas qdo $Q_1 \cdot Q_0 = 1$

Contador síncrono binário de 4 bits:



(a)



Contador síncrono × assíncrono – Detalhes:

Contador **síncrono** 3 bits (módulo 8):

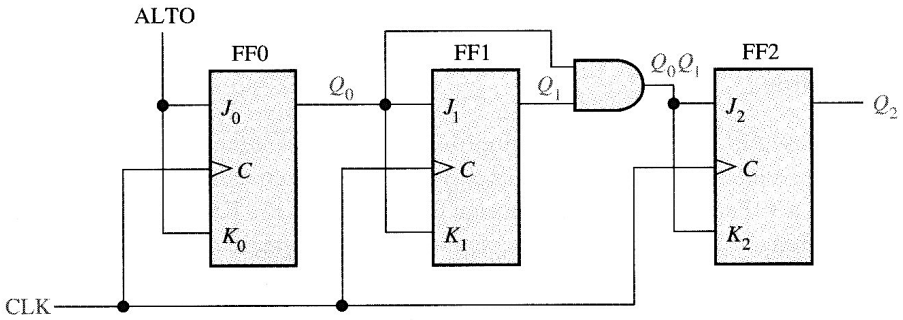
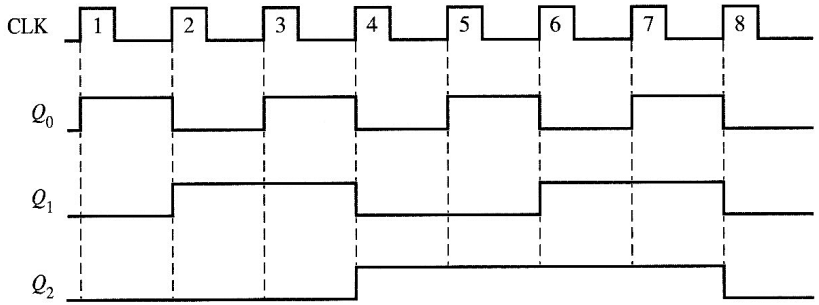


FIGURA 8.14 Contador binario síncrono de 3 bits.



Contador **assíncrono** 3 bits (módulo 8):

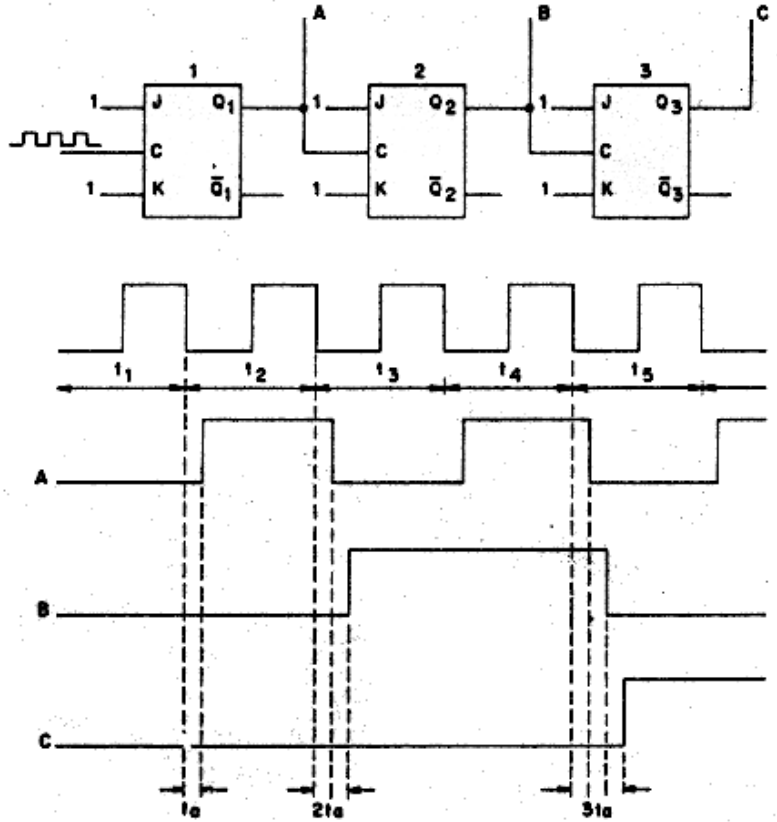
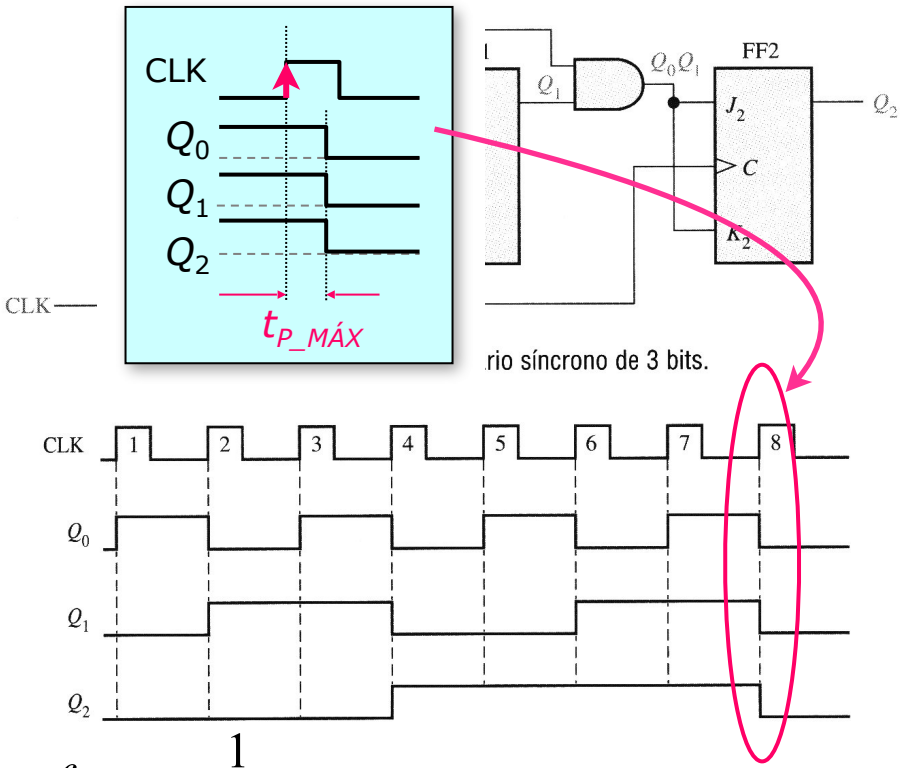


Figura 4.18

Contador síncrono × assíncrono – Detalhes:

Contador **síncrono** 3 bits (módulo 8):



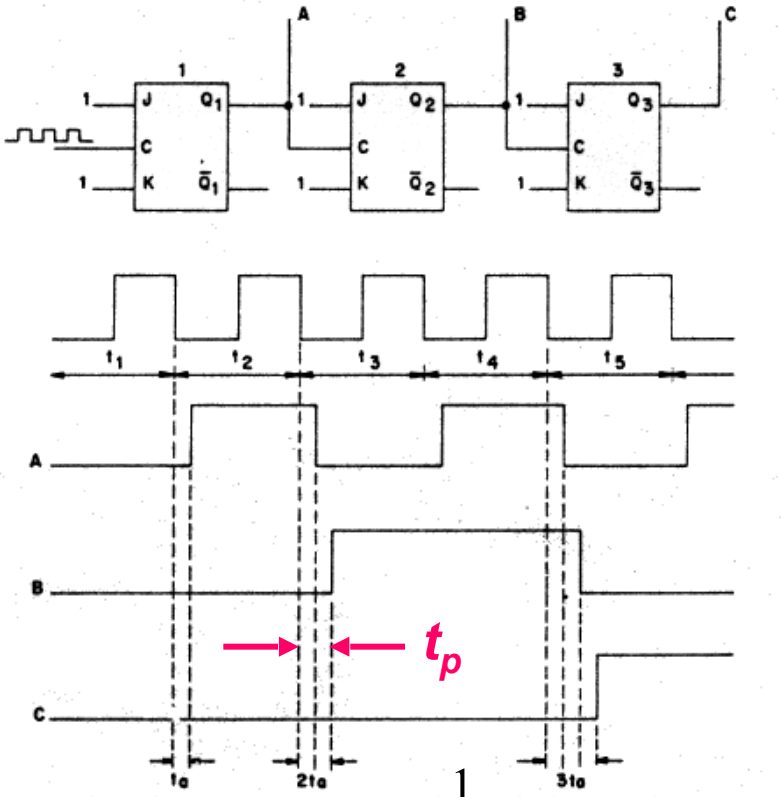
$$f_{MÁX} = \frac{1}{t_{P_{MÁX}}}$$

$$f_{MÁX} = \frac{1}{20\times10^{-9}}$$

$$f_{MÁX} = 50\ \text{MHz}$$

		Typ	Max	Min	Unit
t_{PLH} t_{PHL}	Clock, Clear, Set to Output	15	20		ns
t_s	Setup Time			20	ns
t_h	Hold Time			0	ns
f_{max}			45	30	MHz

Contador **assíncrono** 3 bits (módulo 8):



$$f_{MÁX} = \frac{1}{2\times t_{P_{MÁX}}}$$

$$f_{MÁX} = \frac{1}{40\times10^{-9}}$$

$$f_{MÁX} = 25\ \text{MHz}$$

Contador síncrono × assíncrono – Detalhes:

- E sem problemas com *glitches*...

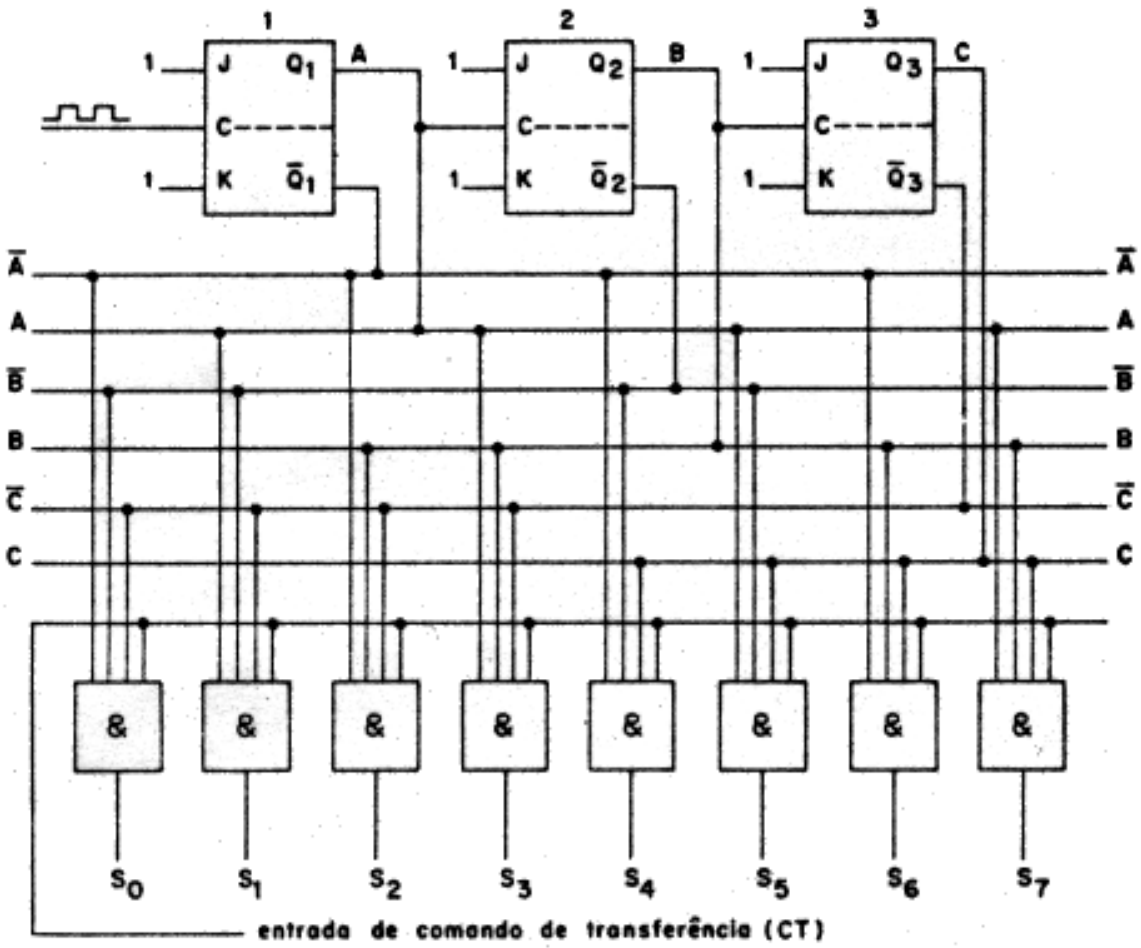


Figura acima: circuito de contador assíncrono con saídas decodificadas.

Contador síncrono × assíncrono – Detalhes:

- E sem problemas com *glitches*...

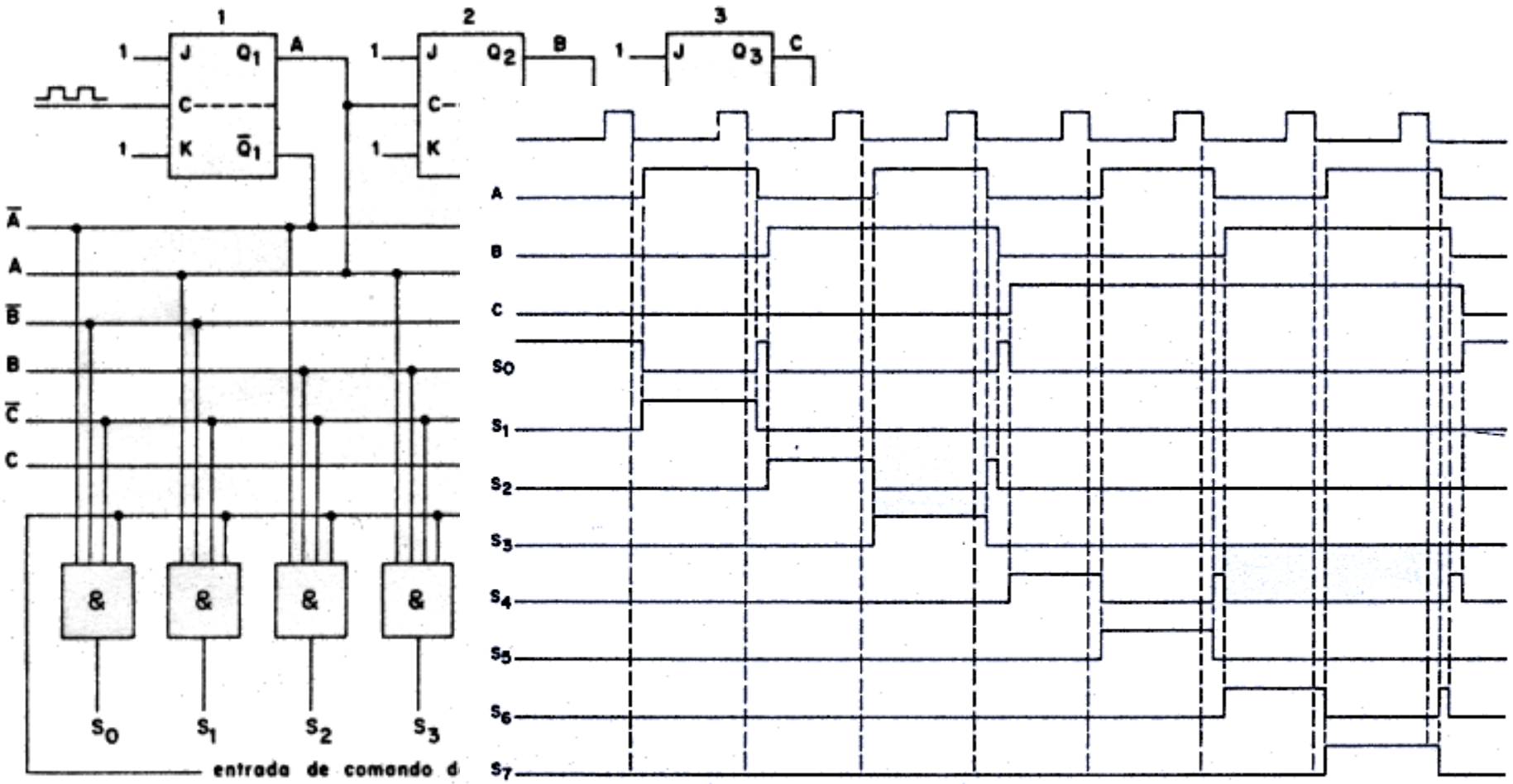


Figura acima: circuito de contador assíncrono con saídas decodificadas.

Contador síncrono × assíncrono – Detalhes:

- E sem problemas com *glitches*...

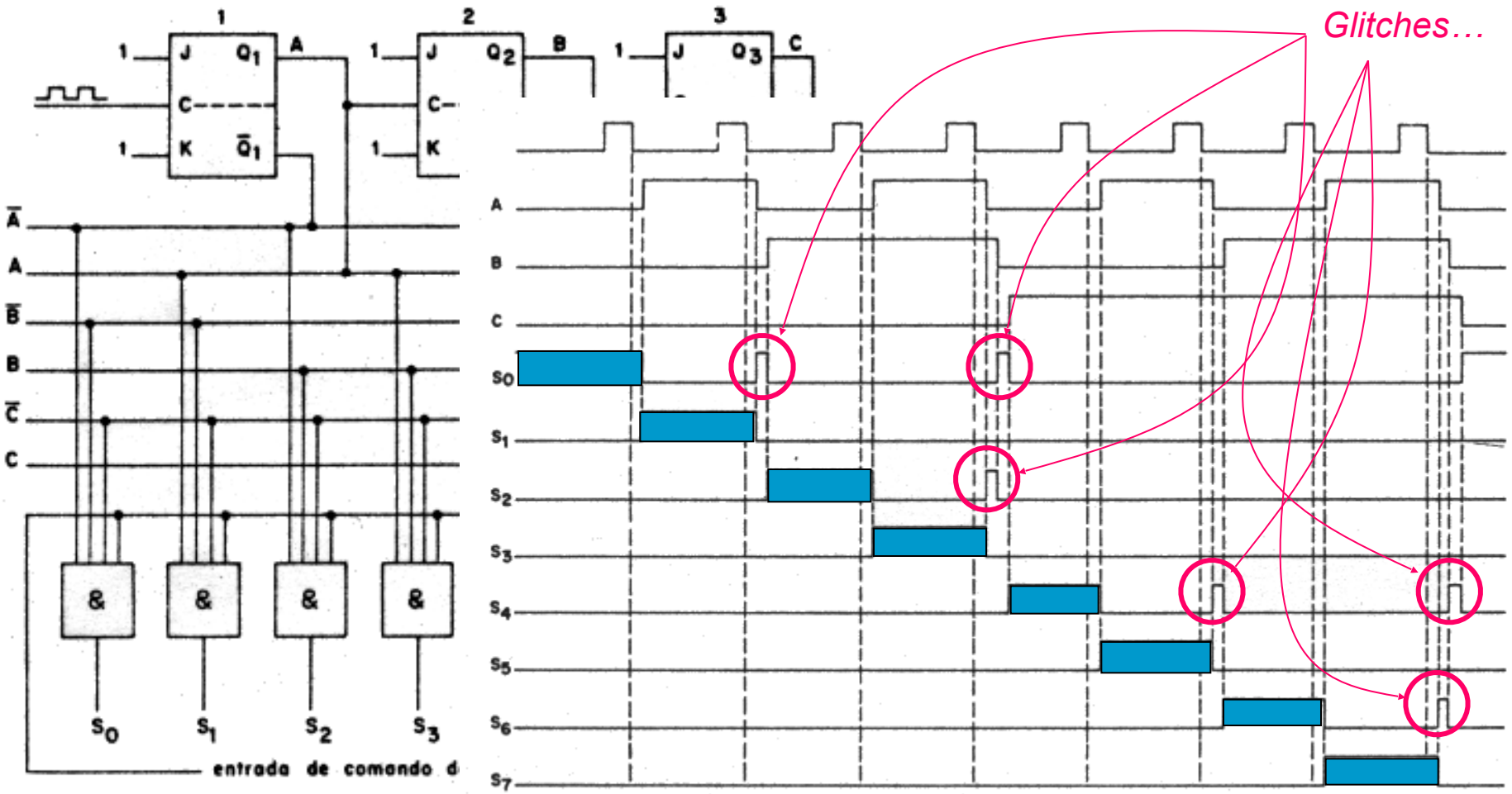
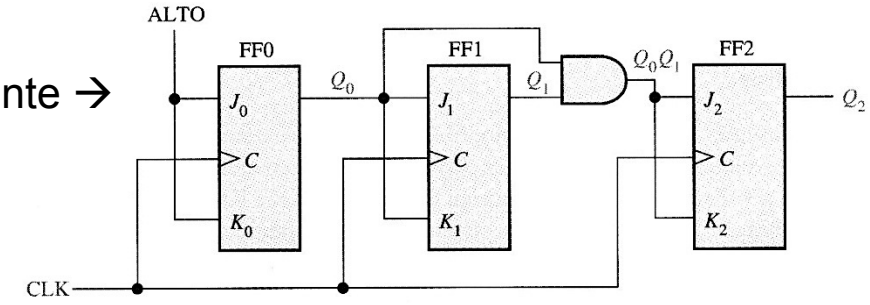


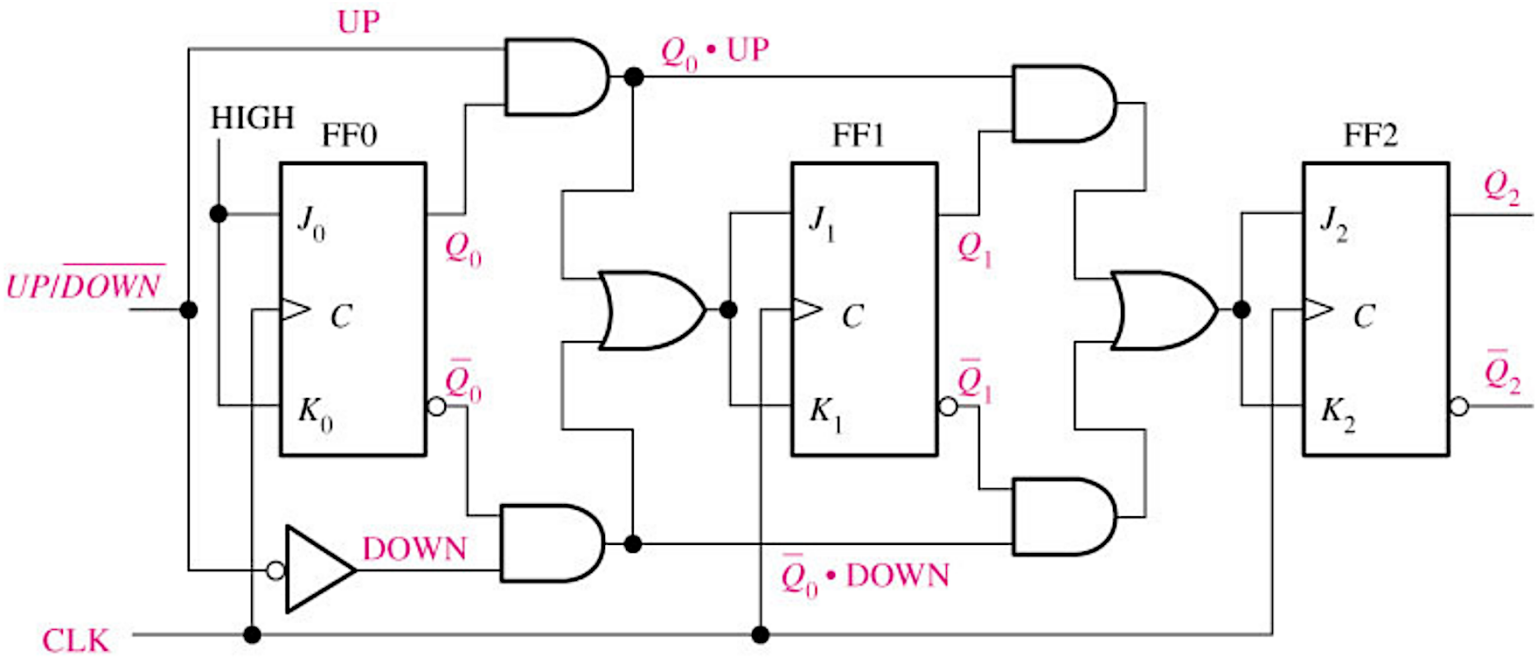
Figura acima: circuito de con

Contador síncrono ascendente/descendente:

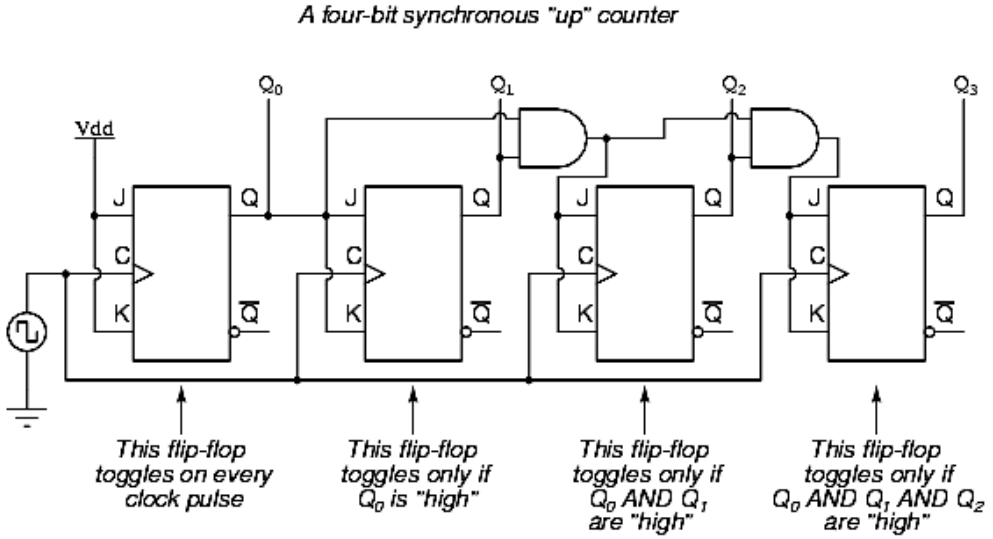
Contador síncrono crescente →

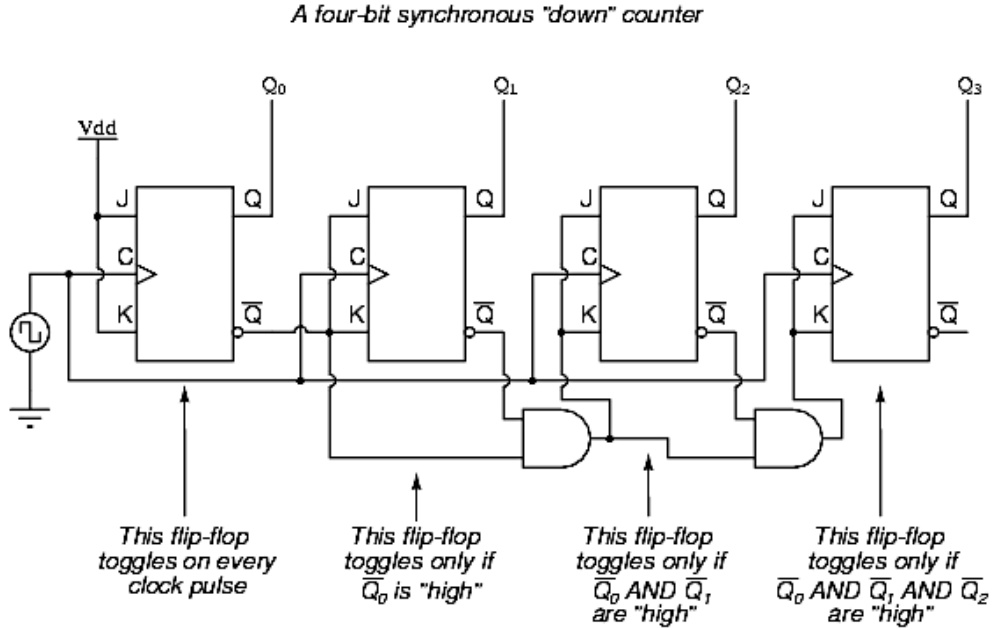


Comando bidireccional

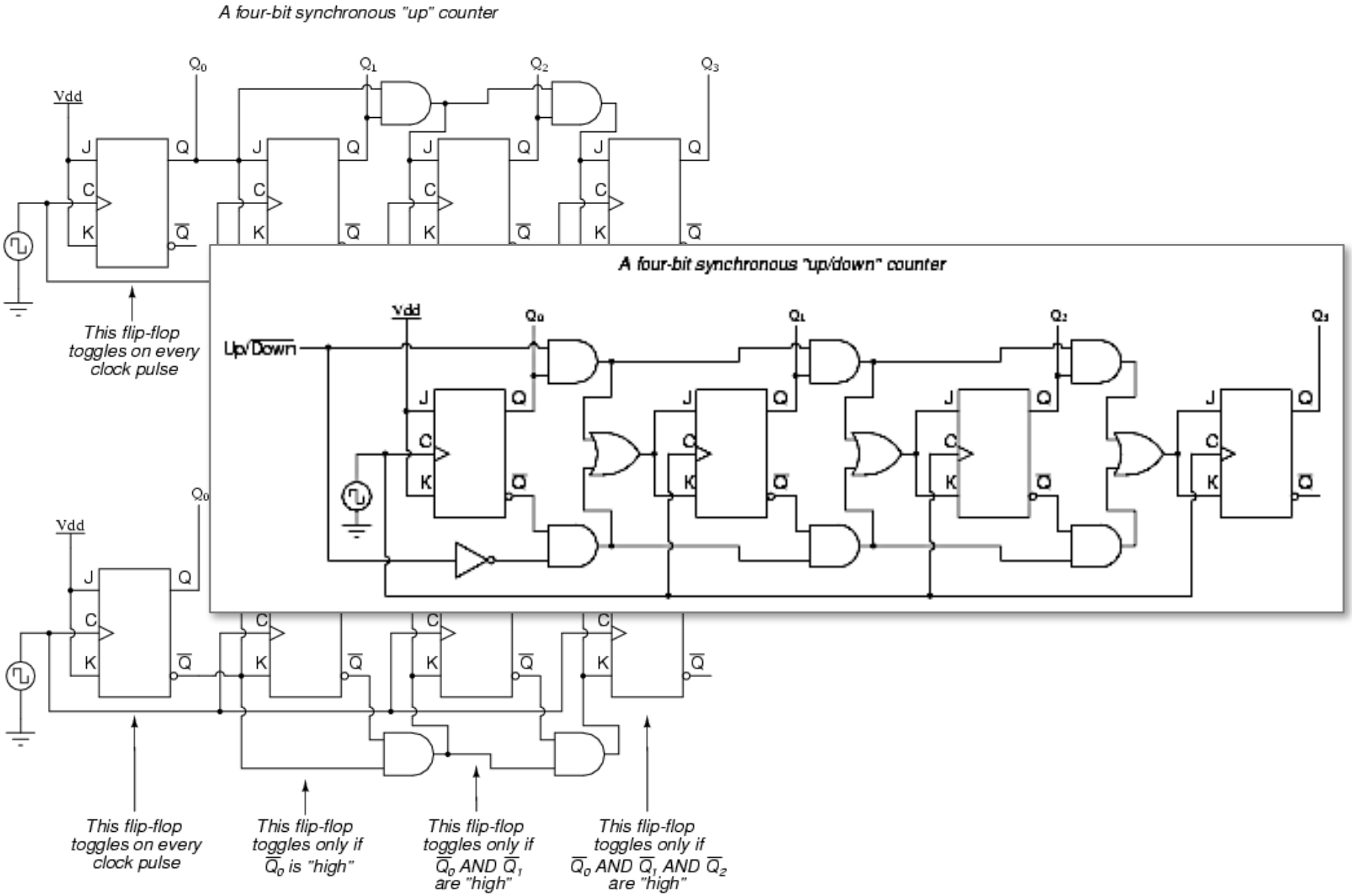


Contador síncrono ascendente/descendente (2):





Contador síncrono ascendente/descendente (2):



CIs comercias 74190/74191

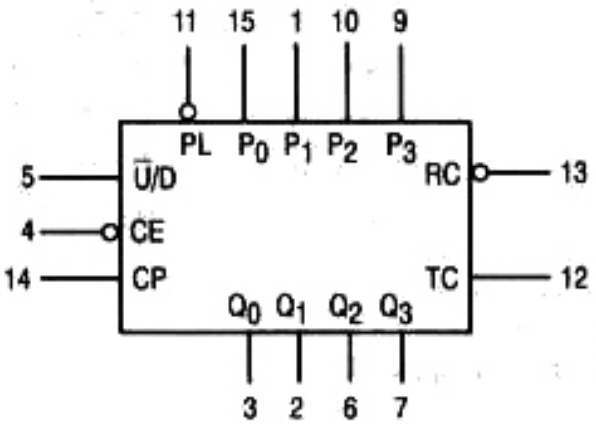
- 74190: **PRESETTABLE BCD/DECADE UP/DOWN COUNTERS**
- 74191: **PRESETTABLE 4-BIT BINARY UP/DOWN COUNTERS**

MOTOROLA

The SN54/74LS190 is a synchronous UP/DOWN BCD Decade (8421) Counter and the SN54/74LS191 is a synchronous UP/DOWN Modulo-16 Binary Counter. State changes of the counters are synchronous with the LOW-to-HIGH transition of the Clock Pulse input.

An asynchronous Parallel Load ($\overline{PL}$) input overrides counting and loads the data present on the P_n inputs into the flip-flops, which makes it possible to use the circuits as programmable counters. A Count Enable ($\overline{CE}$) input serves as the carry/borrow input in multi-stage counters. An Up/Down Count Control ($\overline{U}/D$) input determines whether a circuit counts up or down. A Terminal Count (TC) output and a Ripple Clock ($\overline{RC}$) output provide overflow/underflow indication and make possible a variety of methods for generating carry/borrow signals in multistage counter applications.

- Low Power . . . 90 mW Typical Dissipation
- High Speed . . . 25 MHz Typical Count Frequency
- Synchronous Counting
- Asynchronous Parallel Load
- Individual Preset Inputs
- Count Enable and Up/Down Control Inputs
- Cascadable
- Input Clamp Diodes Limit High Speed Termination Effects



V_{CC} = PIN 16
GND = PIN 8

$\overline{CE}$	Count Enable (Active LOW) Input
CP	Clock Pulse (Active HIGH going edge) Input
$\overline{U}/D$	Up/Down Count Control Input
$\overline{PL}$	Parallel Load Control (Active LOW) Input
P_n	Parallel Data Inputs
Q_n	Flip-Flop Outputs (Note b)
$\overline{RC}$	Ripple Clock Output (Note b)
TC	Terminal Count Output (Note b)

CIs comercias 74190/74191

FUNCTIONAL DESCRIPTION

The LS190 is a synchronous Up/Down BCD Decade Counter and the LS191 is a synchronous Up/Down 4-Bit Binary Counter. The operating modes of the LS190 decade counter and the LS191 binary counter are identical, with the only difference being the count sequences as noted in the state diagrams. Each circuit contains four master/slave flip-flops, with internal gating and steering logic to provide individual preset, count-up and count-down operations.

Each circuit has an asynchronous parallel load capability permitting the counter to be preset to any desired number. When the Parallel Load ($\overline{PL}$) input is LOW, information present on the Parallel Data inputs (P_0–P_3) is loaded into the counter and appears on the Q outputs. This operation overrides the counting functions, as indicated in the Mode Select Table.

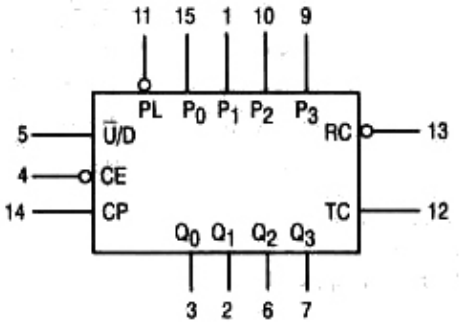
A HIGH signal on the $\overline{CE}$ input inhibits counting. When $\overline{CE}$ is LOW, internal state change are initiated synchronously by the LOW-to-HIGH transition of the clock input. The direction of counting is determined by the $\overline{U}$/D input signal, as indicated in the Mode Select Table. When counting is to be enabled, the $\overline{CE}$ signal can be made LOW when the clock is in either state. However, when counting is to be inhibited, the LOW-to-HIGH $\overline{CE}$ transition must occur only while the clock is HIGH. Similarly, the $\overline{U}$/D signal should only be changed when either $\overline{CE}$ or the clock is HIGH.

Two types of outputs are provided as overflow/underflow indicators. The Terminal Count (TC) output is normally LOW and goes HIGH when a circuit reaches zero in the count-down mode or reaches maximum (9 for the LS190, 15 for the LS191) in the count-up mode. The TC output will then remain HIGH until a state change occurs, whether by counting or presetting or until $\overline{U}$/D is changed. The TC output should not be used as a clock signal because it is subject to decoding spikes.

The TC signal is also used internally to enable the Ripple Clock ($\overline{RC}$) output. The $\overline{RC}$ output is normally HIGH. When $\overline{CE}$ is LOW and TC is HIGH, the $\overline{RC}$ output will go LOW when the clock next goes LOW and will stay LOW until the clock goes HIGH again. This feature simplifies the design of multi-stage counters, as indicated in Figures a and b. In Figure a, each $\overline{RC}$ output is used as the clock input for the next higher stage. This configuration is particularly advantageous when the clock source has a limited drive capability, since it drives only the first stage. To prevent counting in all stages it is only necessary to inhibit the first stage, since a HIGH signal on $\overline{CE}$ inhibits the $\overline{RC}$ output pulse, as indicated in the $\overline{RC}$ Truth Table. A disadvantage of this configuration, in some applications, is the timing skew between state changes in the first and last stages. This represents the cumulative delay of the clock as it ripples through the preceding stages.

A method of causing state changes to occur simultaneously in all stages is shown in Figure b. All clock inputs are driven in parallel and the $\overline{RC}$ outputs propagate the carry/borrow signals in ripple fashion. In this configuration the LOW state duration of the clock must be long enough to allow the negative-going edge of the carry/borrow signal to ripple through to the last stop before the clock goes HIGH. There is no such restriction on the HIGH state duration of the clock, since the $\overline{RC}$ output of any package goes HIGH shortly after its CP input goes HIGH.

The configuration shown in Figure c avoids ripple delays and their associated restrictions. The $\overline{CE}$ input signal for a given stage is formed by combining the TC signals from all the preceding stages. Note that in order to inhibit counting an enable signal must be included in each carry gate. The simple inhibit scheme of Figures a and b doesn't apply, because the TC output of a given stage is not affected by its own $\overline{CE}$.



V_{CC} = PIN 16
GND = PIN 8

MODE SELECT TABLE

INPUTS				MODE
PL	CE	U/D	CP	
H	L	L	⌐	Count Up
H	L	H	⌐	Count Down
L	X	X	X	Preset (Asyn.)
H	H	X	X	No Change (Hold)

RC TRUTH TABLE

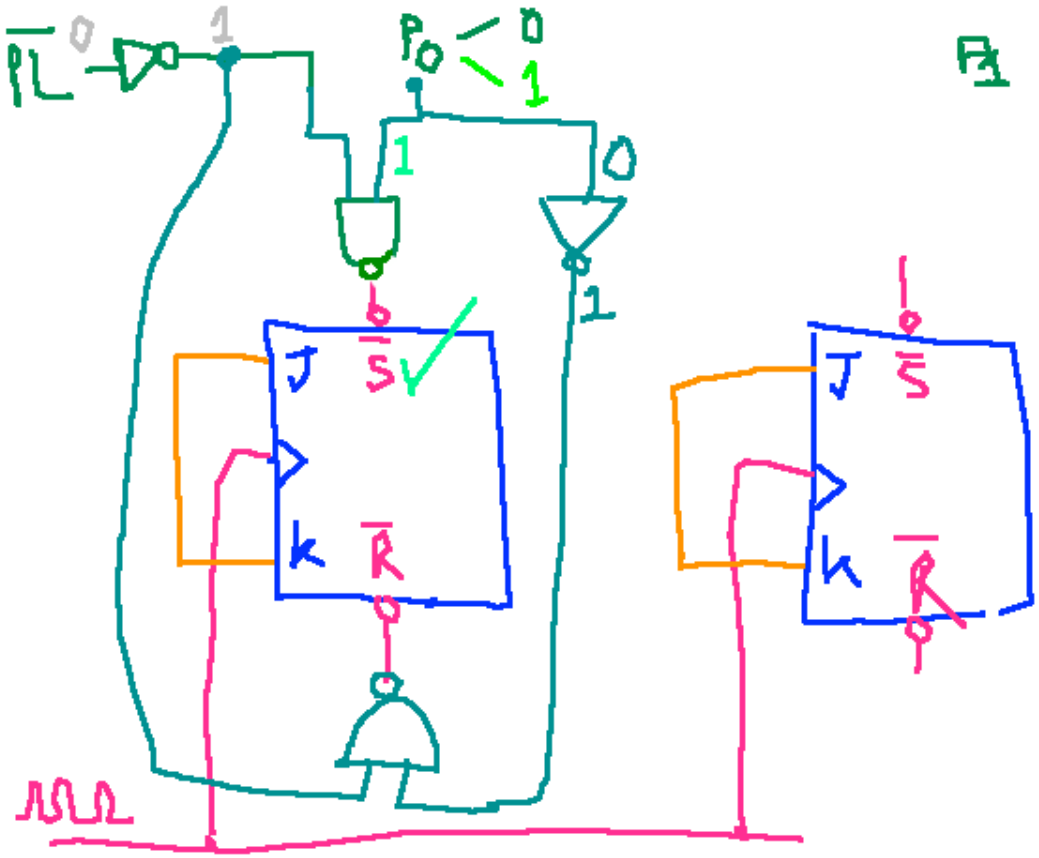
INPUTS			RC OUTPUT
$\overline{CE}$	TC*	CP	
L	H	⌐¬	⌐¬
H	X	X	H
X	L	X	H

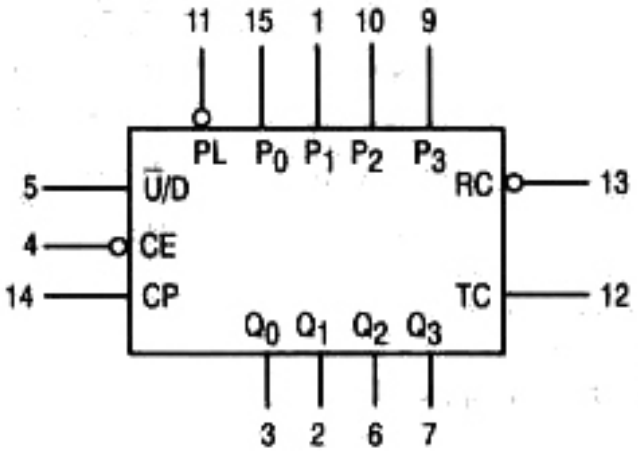
* TC is generated internally

L = LOW Voltage Level
H = HIGH Voltage Level
X = Don't Care
⌐ = LOW-to-HIGH Clock Transition
⌐¬ = LOW Pulse

CIs comercias 74190/74191

Detalhe do “Parallel Load” ($\overline{PL}$):





V_{CC} = PIN 16
GND = PIN 8

MODE SELECT TABLE

INPUTS				MODE
PL	CE	$\overline{U}$/D	CP	
H	L	L	↓	Count Up
H	L	H	↓	Count Down
L	X	X	X	Preset (Asyn.)
H	H	X	X	No Change (Hold)

RC TRUTH TABLE

INPUTS			RC OUTPUT
$\overline{CE}$	TC*	CP	
L	H	⊓	⊓
H	X	X	H
X	L	X	H

* TC is generated internally

L = LOW Voltage Level
H = HIGH Voltage Level
X = Don't Care
↓ = LOW-to-HIGH Clock Transition
⊓ = LOW Pulse

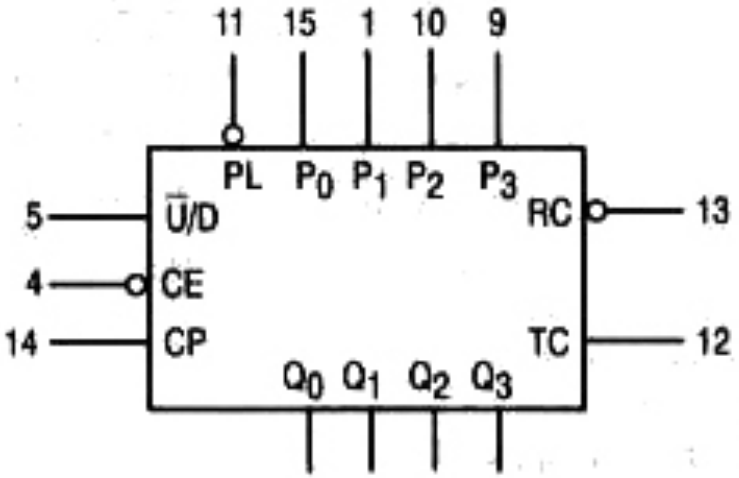
CIs comercias 74190/74191

Tabelas funcionais…

- 74190: PRESETTABLE BCD/DECADE UP/DOWN COUNTERS
- 74191: PRESETTABLE 4-BIT BINARY UP/DOWN COUNTERS

MOTOROLA

The SN54/74LS190 is a synchronous UP/DOWN BCD Decade (8421) Counter and the SN54/74LS191 is a synchronous UP/DOWN Modulo-16 Binary Counter. State changes of the counters are synchronous with the



$\overline{CE}$	Count Enable (Active LOW) Input
CP	Clock Pulse (Active HIGH going edge) Input
$\overline{U}/D$	Up/Down Count Control Input
$\overline{PL}$	Parallel Load Control (Active LOW) Input
P_n	Parallel Data Inputs
Q_n	Flip-Flop Outputs (Note b)
$\overline{RC}$	Ripple Clock Output (Note b)
TC	Terminal Count Output (Note b)

$\overline{RC}$ Truth Table

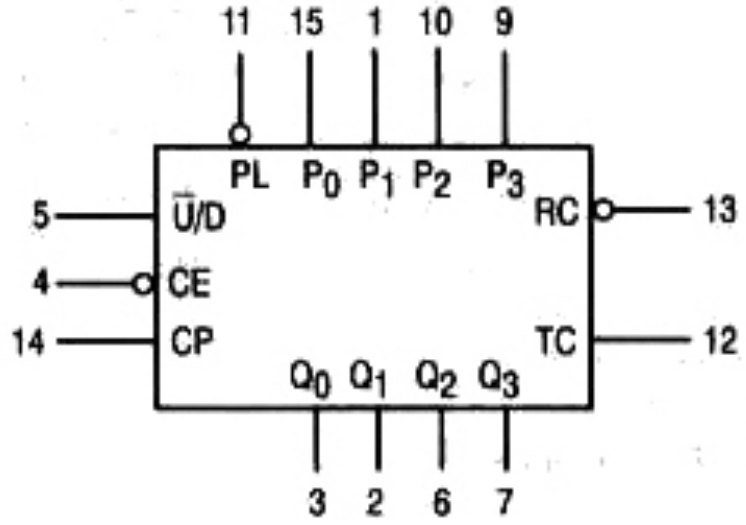
Inputs			Output
$\overline{CE}$	TC*	CP	$\overline{RC}$
L	H	⊓	⊓
H	X	X	H
X	L	X	H

Mode Select Table

Inputs				Mode
$\overline{PL}$	$\overline{CE}$	$\overline{U}/D$	CP	
H	L	L	⌐	Count Up
H	L	H	⌐	Count Down
L	X	X	X	Preset (Asyn.)
H	H	X	X	No Change (Hold)

*TC is generated internally
H = HIGH Voltage Level
L = LOW Voltage Level
X = Immaterial
⌐ = LOW-to-HIGH Clock Transition
⊓ = LOW Pulse

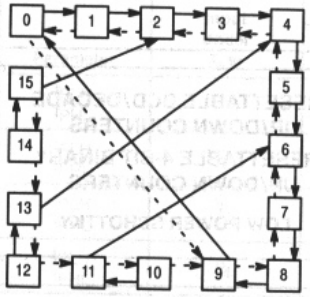
CIs comercias 74190/74191



V_{CC} = PIN 16
GND = PIN 8

$\overline{CE}$	Count Enable (Active LOW) Input
CP	Clock Pulse (Active HIGH going edge) Input
$\overline{U}/D$	Up/Down Count Control Input
$\overline{PL}$	Parallel Load Control (Active LOW) Input
P_n	Parallel Data Inputs
Q_n	Flip-Flop Outputs (Note b)
$\overline{RC}$	Ripple Clock Output (Note b)
TC	Terminal Count Output (Note b)

STATE DIAGRAMS

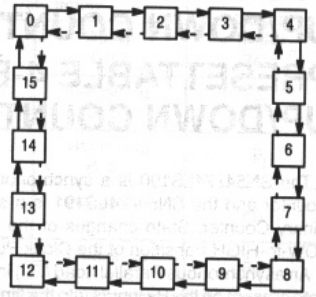


LS190

LS190
UP: $TC = Q_0 \cdot \overline{Q_3} \cdot (\overline{U/D})$
DOWN: $TC = \overline{Q_0} \cdot \overline{Q_1} \cdot \overline{Q_2} \cdot \overline{Q_3} \cdot (\overline{U/D})$

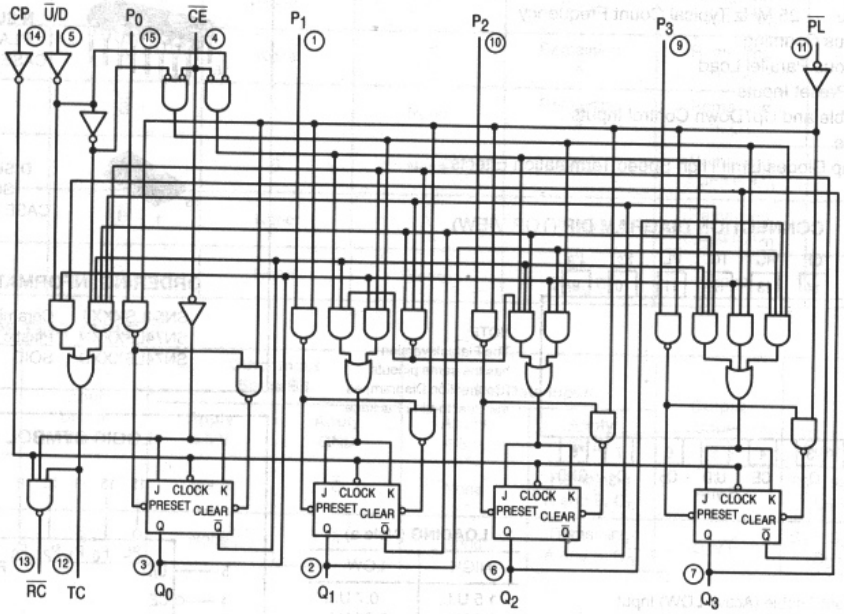
LS191
UP: $TC = Q_0 \cdot Q_1 \cdot Q_2 \cdot Q_3 \cdot (\overline{U/D})$
DOWN: $TC = \overline{Q_0} \cdot \overline{Q_1} \cdot \overline{Q_2} \cdot \overline{Q_3} \cdot (\overline{U/D})$

COUNT UP ——
COUNT DOWN - - - -



LS191

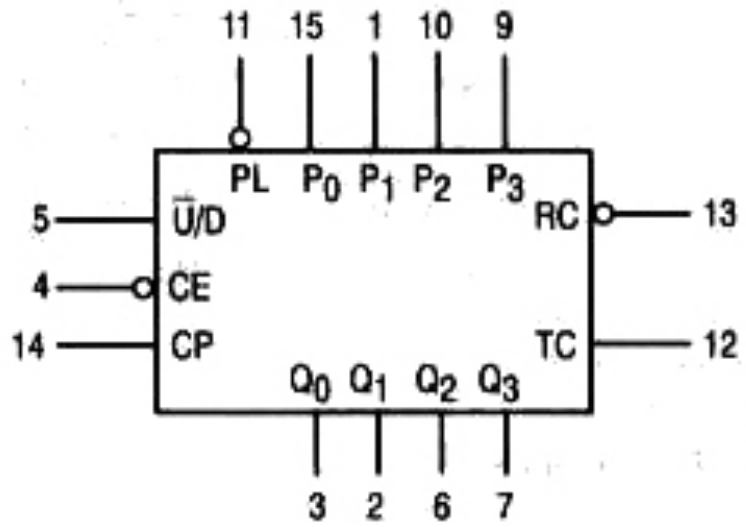
LOGIC DIAGRAMS



V_{CC} = PIN 16
GND = PIN 8
○ = PIN NUMBERS

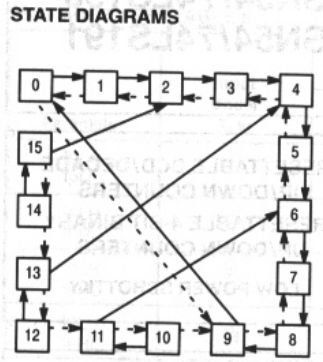
DECADE COUNTER
LS190

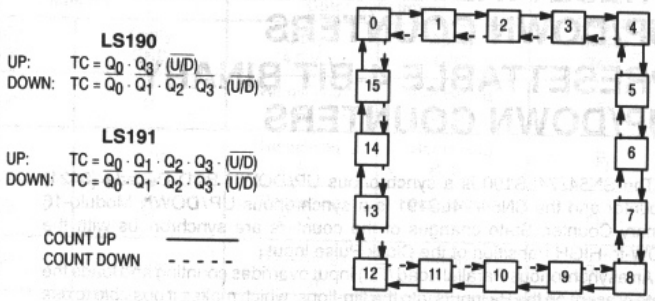
CIs comercias 74190/74191



VCC
GND

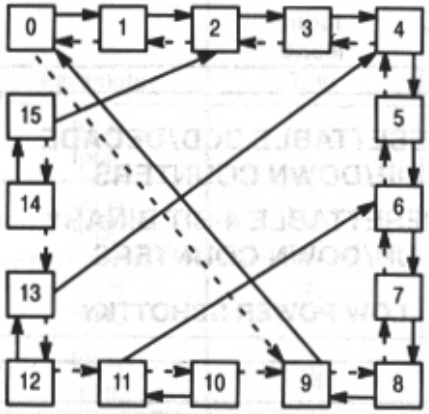
CE Count Enable
CP Clock Pulse (
U/D Up/Down Cou
PL Parallel Load
Pn Parallel Data
Qn Flip-Flop Outp
RC Ripple Clock
TC Terminal Cou

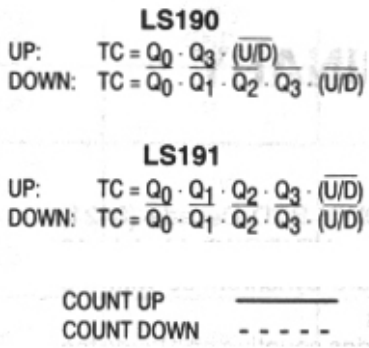


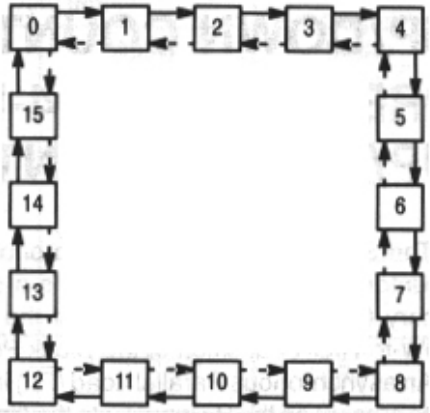


LOGIC DIAGRAMS

Diagrama de Estados



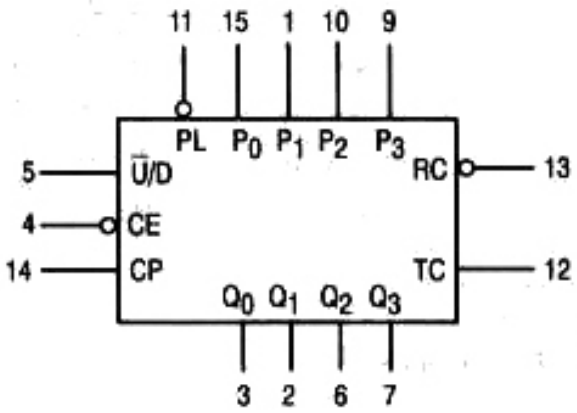




VCC = PIN 16
GND = PIN 8
○ = PIN NUMBERS

DECADE COUNTER
LS190

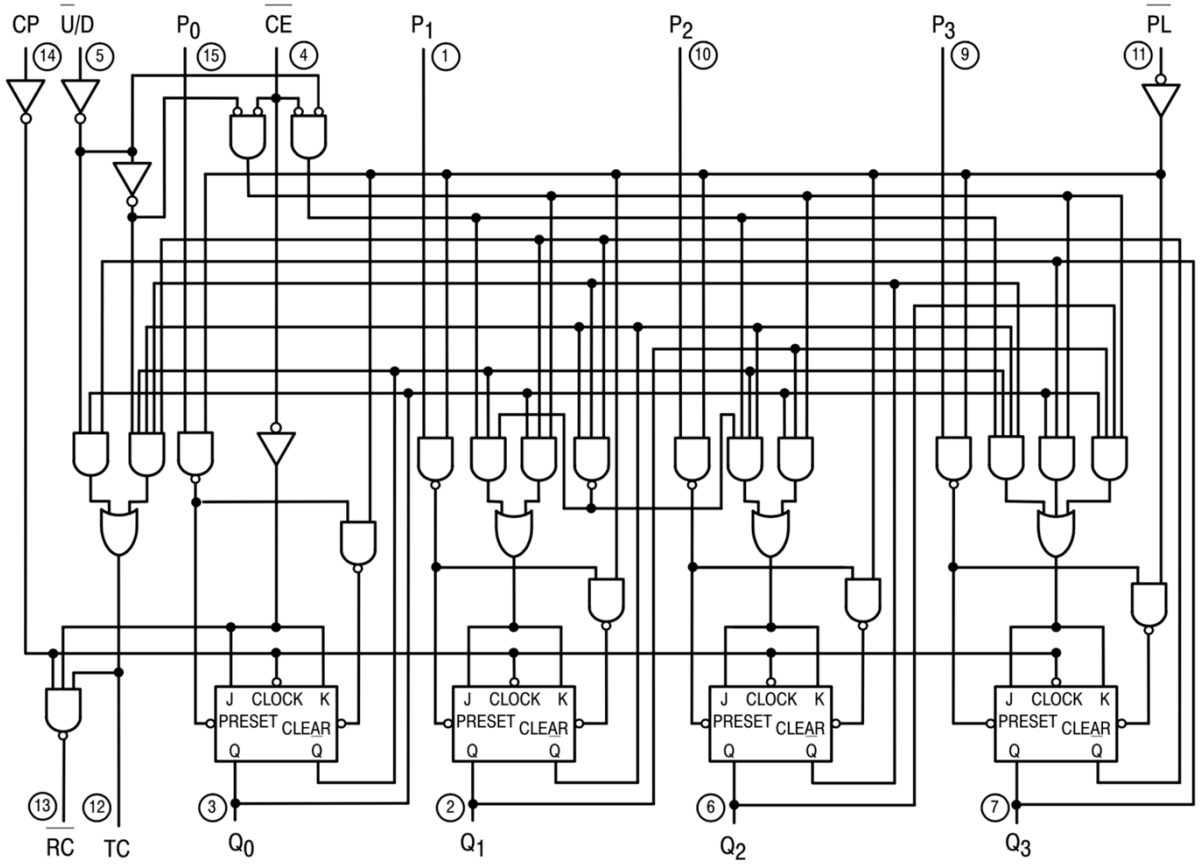
CIs comercias 74190/74191



V_{CC} = PIN 16
GND = PIN 8

$\overline{CE}$	Count Enable (Active LOW) Input
CP	Clock Pulse (Active HIGH going edge) Input
$\overline{U}/D$	Up/Down Count Control Input
$\overline{PL}$	Parallel Load Control (Active LOW) Input
P_n	Parallel Data Inputs
Q_n	Flip-Flop Outputs (Note b)
$\overline{RC}$	Ripple Clock Output (Note b)
TC	Terminal Count Output (Note b)

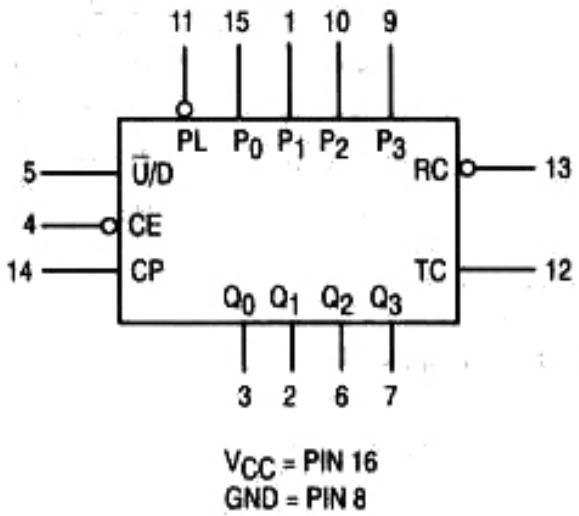
Diagrama lógico



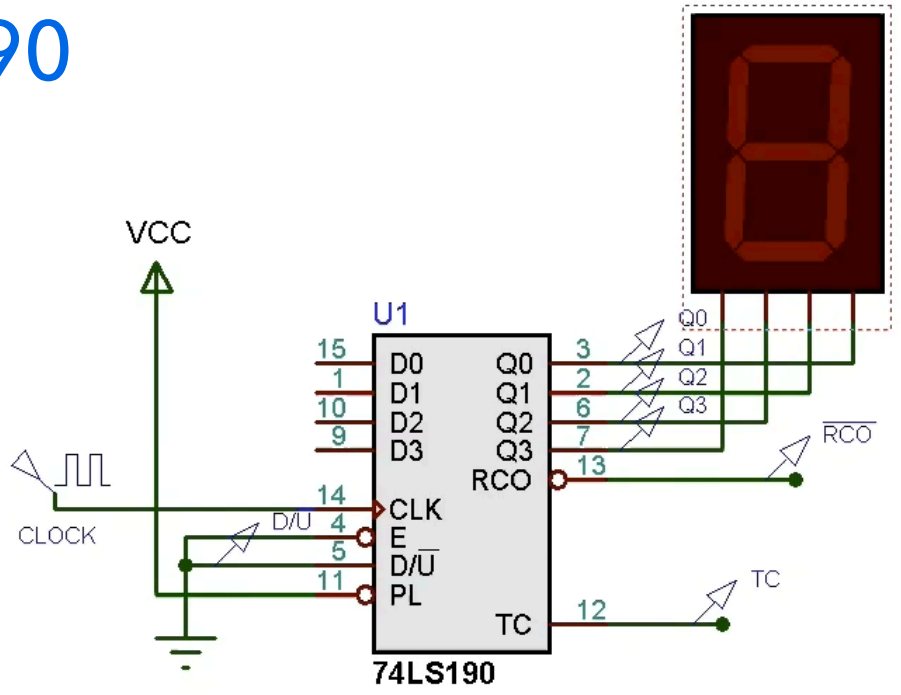
V_{CC} = PIN 16
GND = PIN 8
◯ = PIN NUMBERS

DECADE COUNTER
LS190

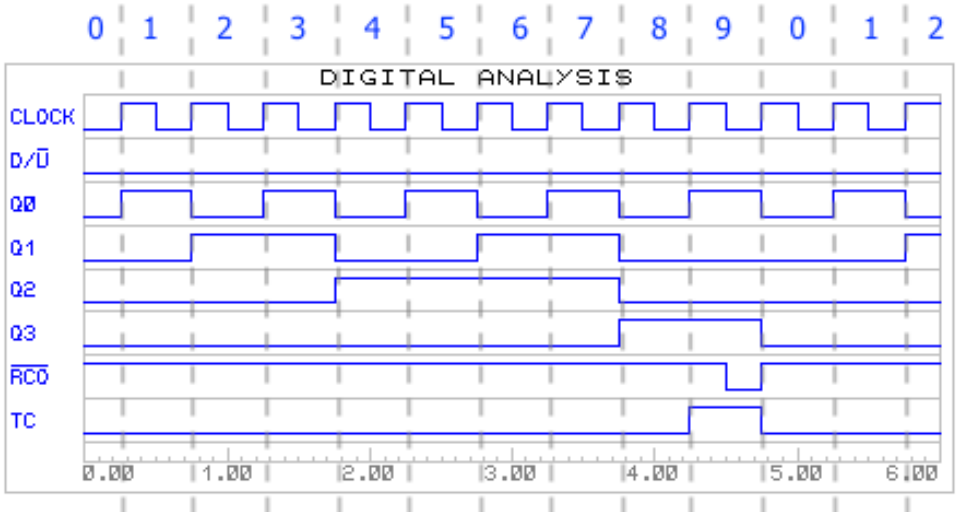
Exemplo de uso de 74190



Crescente:
[0..9]

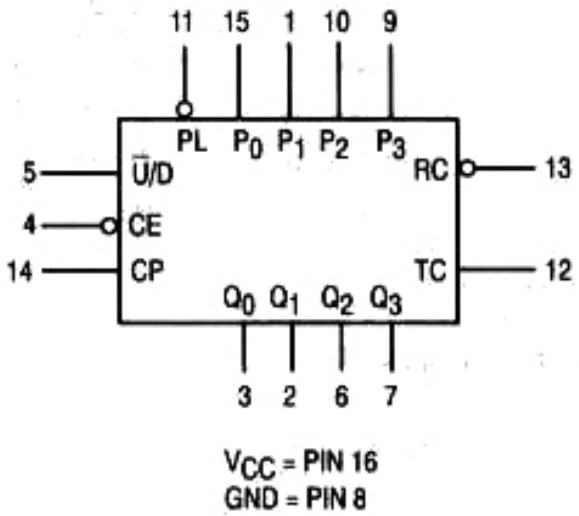


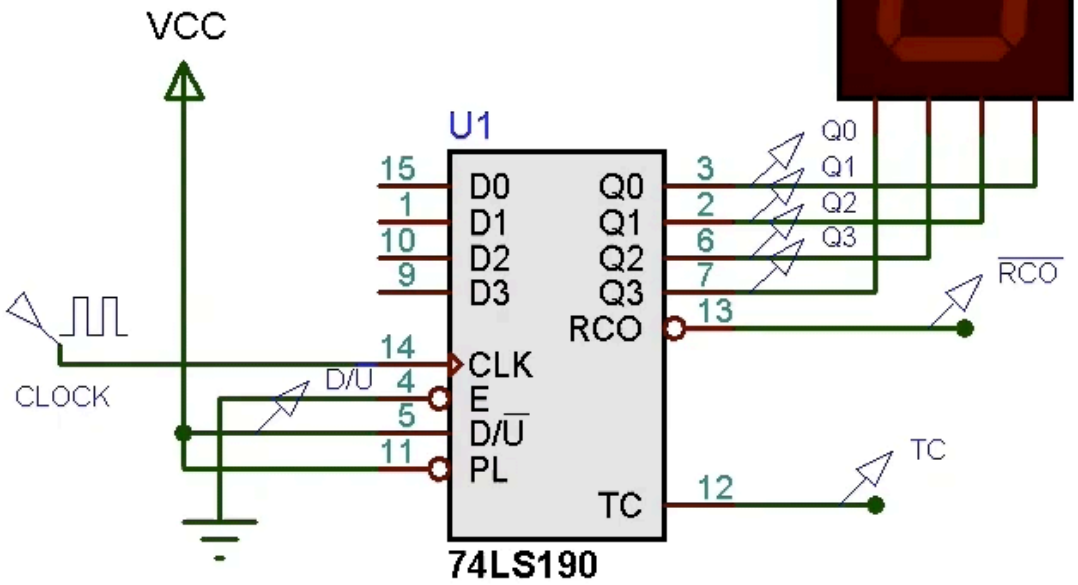
$\overline{CE}$	Count Enable (Active LOW) Input
CP	Clock Pulse (Active HIGH going edge) Input
$\overline{U}/D$	Up/Down Count Control Input
$\overline{PL}$	Parallel Load Control (Active LOW) Input
P_n	Parallel Data Inputs
Q_n	Flip-Flop Outputs (Note b)
$\overline{RC}$	Ripple Clock Output (Note b)
TC	Terminal Count Output (Note b)



Exemplo de uso de 74190

Decrescente:
[9..0]

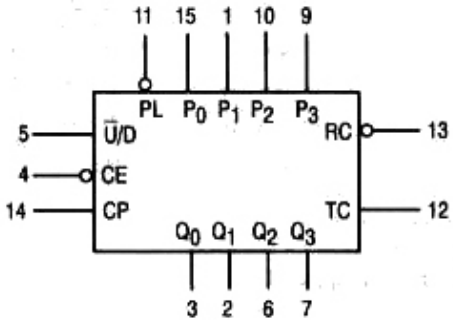




$\overline{CE}$	Count Enable (Active LOW) Input
CP	Clock Pulse (Active HIGH going edge) Input
$\overline{U}/D$	Up/Down Count Control Input
$\overline{PL}$	Parallel Load Control (Active LOW) Input
P_n	Parallel Data Inputs
$\overline{Q_n}$	Flip-Flop Outputs (Note b)
$\overline{RC}$	Ripple Clock Output (Note b)
TC	Terminal Count Output (Note b)

CIs comercias 74190/74191

$\overline{CE}$	Count Enable (Active LOW) Input
CP	Clock Pulse (Active HIGH going edge)
$\overline{U}/D$	Up/Down Count Control Input
$\overline{PL}$	Parallel Load Control (Active LOW) Inp
P_n	Parallel Data Inputs
Q_n	Flip-Flop Outputs (Note b)
$\overline{RC}$	Ripple Clock Output (Note b)
TC	Terminal Count Output (Note b)



V_{CC} = PIN 16
GND = PIN 8

Características AC:

AC CHARACTERISTICS (T_A = 25°C)

Symbol	Parameter	Limits Min	Limits Typ	Limits Max	Unit	Test Conditions
f_{MAX}	Maximum Clock Frequency	20	25		MHz	V_{CC} = 5.0 V, C_L = 15 pF
t_{PLH} / t_{PHL}	Propagation Delay, $\overline{PL}$ to Output Q		22 / 33	33 / 50	ns	
t_{PLH} / t_{PHL}	Data to Output Q		20 / 27	32 / 40	ns	
t_{PLH} / t_{PHL}	Clock to $\overline{RC}$		13 / 16	20 / 24	ns	
t_{PLH} / t_{PHL}	Clock to Output Q		16 / 24	24 / 36	ns	
t_{PLH} / t_{PHL}	Clock to TC		28 / 37	42 / 52	ns	
t_{PLH} / t_{PHL}	$\overline{U}/D$ to $\overline{RC}$		30 / 30	45 / 45	ns	
t_{PLH} / t_{PHL}	$\overline{U}/D$ to TC		21 / 22	33 / 33	ns	
t_{PLH} / t_{PHL}	$\overline{CE}$ to $\overline{RC}$		21 / 22	33 / 33	ns	

AC SETUP REQUIREMENTS (T_A = 25°C)

Symbol	Parameter	Limits Min	Limits Typ	Limits Max	Unit	Test Conditions
t_W	CP Pulse Width	25			ns	V_{CC} = 5.0 V
t_W	$\overline{PL}$ Pulse Width	35			ns	
t_s	Data Setup Time	20			ns	
t_h	Data Hold Time	5.0			ns	
t_{rec}	Recovery Time	40			ns	

DEFINITIONS OF TERMS

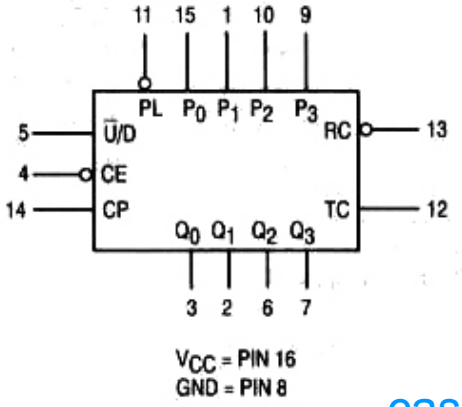
SETUP TIME (t_s) is defined as the minimum time required for the correct logic level to be present at the logic input prior to the clock transition from LOW-to-HIGH in order to be recognized and transferred to the outputs.

HOLD TIME (t_h) is defined as the minimum time following the clock transition from LOW-to-HIGH that the logic level must be maintained at the input in order to ensure continued recognition. A negative HOLD TIME indicates that the correct logic level may be released prior to the clock transition from LOW-to-HIGH and still be recognized.

RECOVERY TIME (t_{rec}) is defined as the minimum time required between the end of the reset pulse and the clock transition from LOW-to-HIGH in order to recognize and transfer HIGH data to the Q outputs.

CIs comercias 74190/74191

$\overline{CE}$	Count Enable (Active LOW) Input
CP	Clock Pulse (Active HIGH going edge) Input
$\overline{U}/D$	Up/Down Count Control Input
$\overline{PL}$	Parallel Load Control (Active LOW) Input
P_n	Parallel Data Inputs
Q_n	Flip-Flop Outputs (Note b)
$\overline{RC}$	Ripple Clock Output (Note b)
TC	Terminal Count Output (Note b)



LS190

UP: $TC = \overline{Q_0} \cdot \overline{Q_3} \cdot \overline{(U/D)}$

DOWN: $TC = Q_0 \cdot Q_1 \cdot Q_2 \cdot Q_3 \cdot (U/D)$

Opções de cascateamento:

SN54/74LS190 • SN54/74LS191

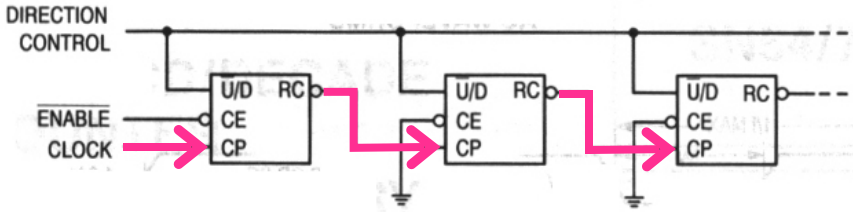


Figure a. n-Stage Counter Using Ripple Clock

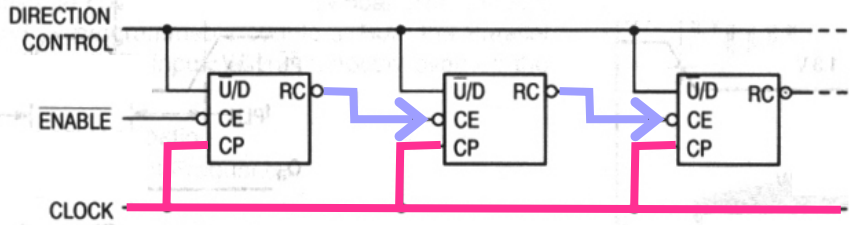


Figure b. Synchronous n-Stage Counter Using Ripple Carry/Borrow

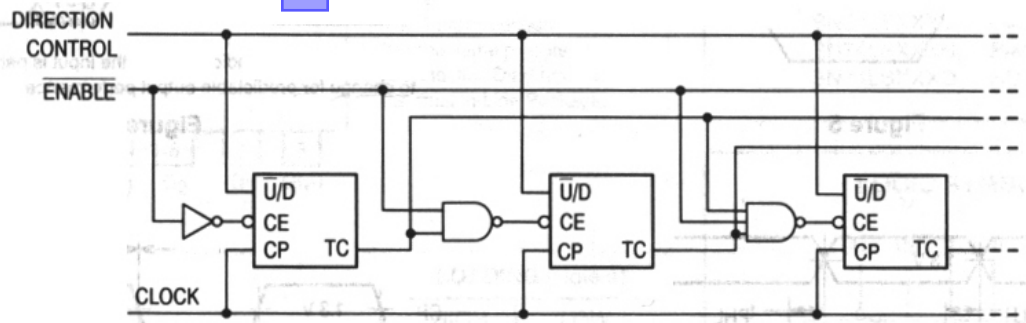
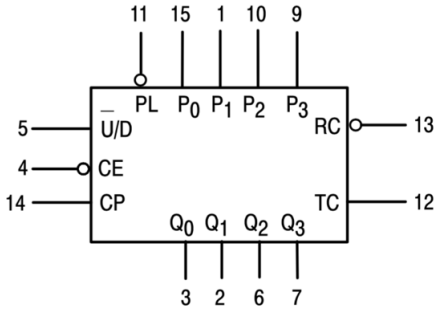
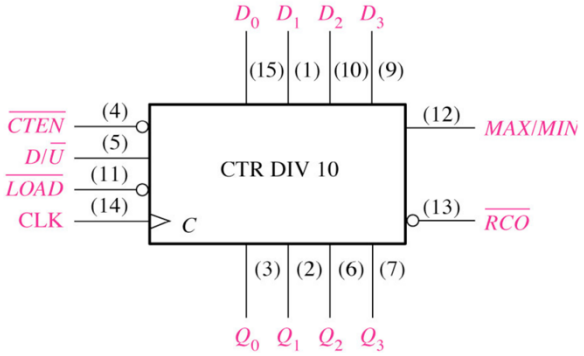


Figure c. Synchronous n-Stage Counter with Parallel Gated Carry/Borrow

74190:





Opções de cascateamento:

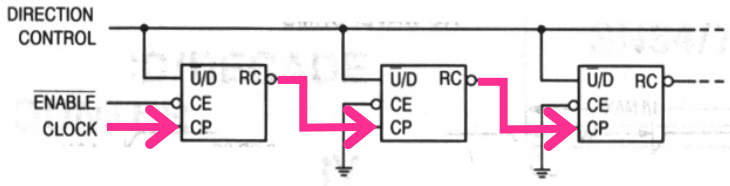


Figure a. n-Stage Counter Using Ripple Clock

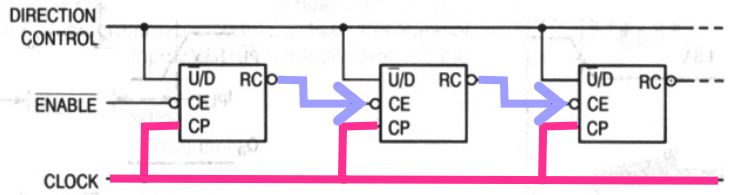


Figure b. Synchronous n-Stage Counter Using Ripple Carry/Borrow

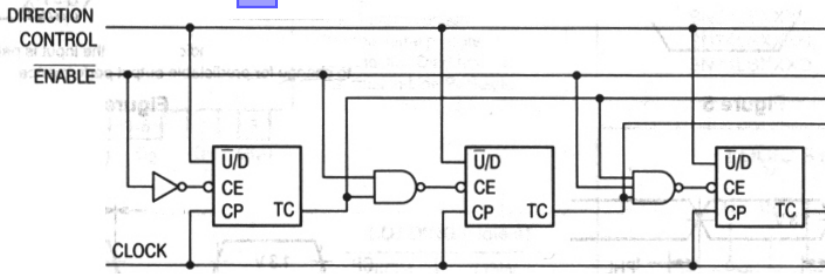
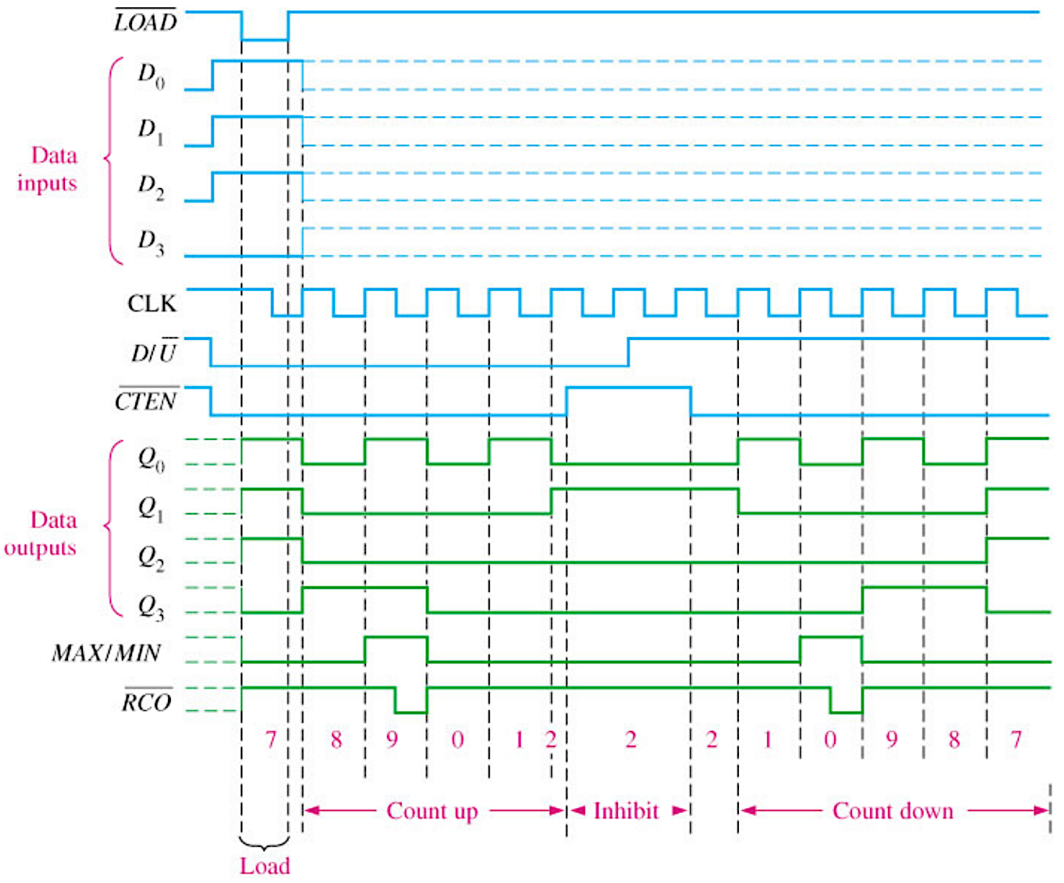
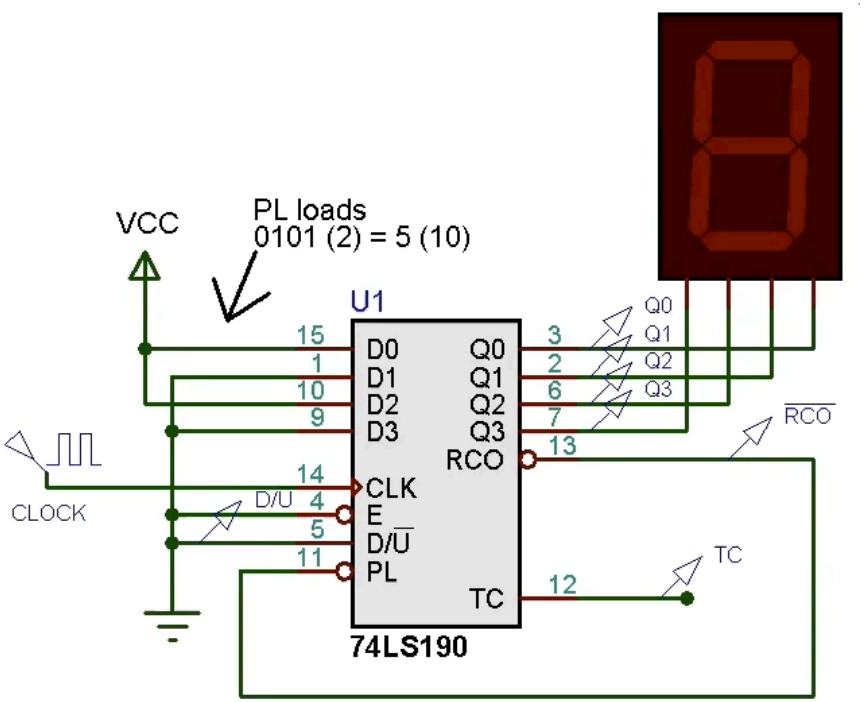
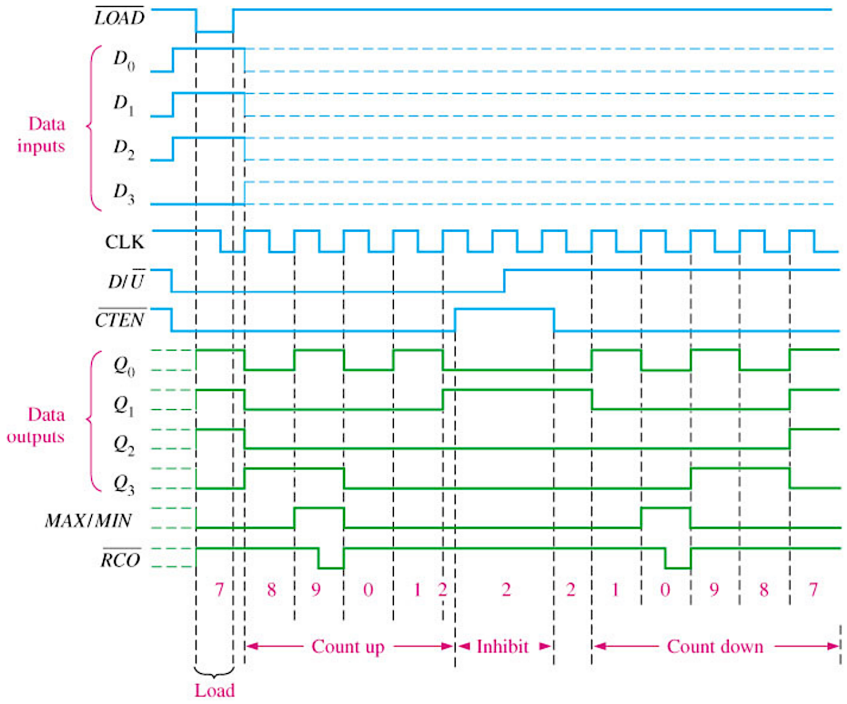


Figure c. Synchronous n-Stage Counter with Parallel Gated Carry/Borrow



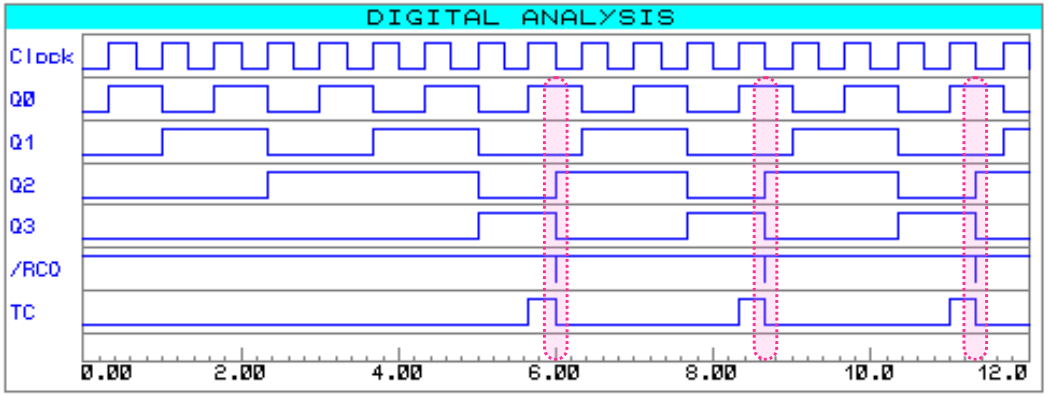
Usando entradas PL num 74190



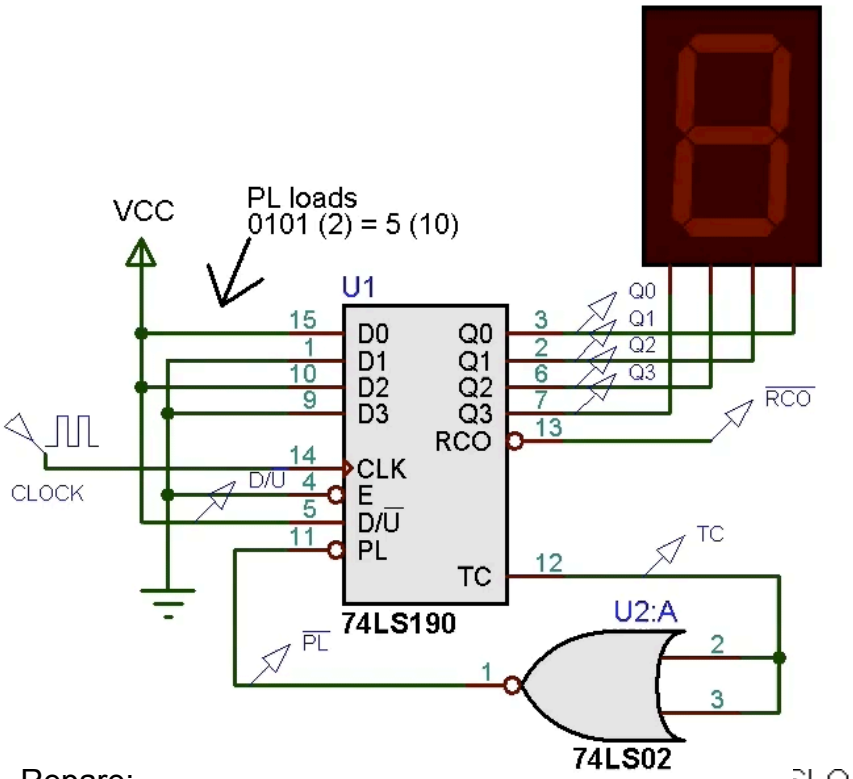


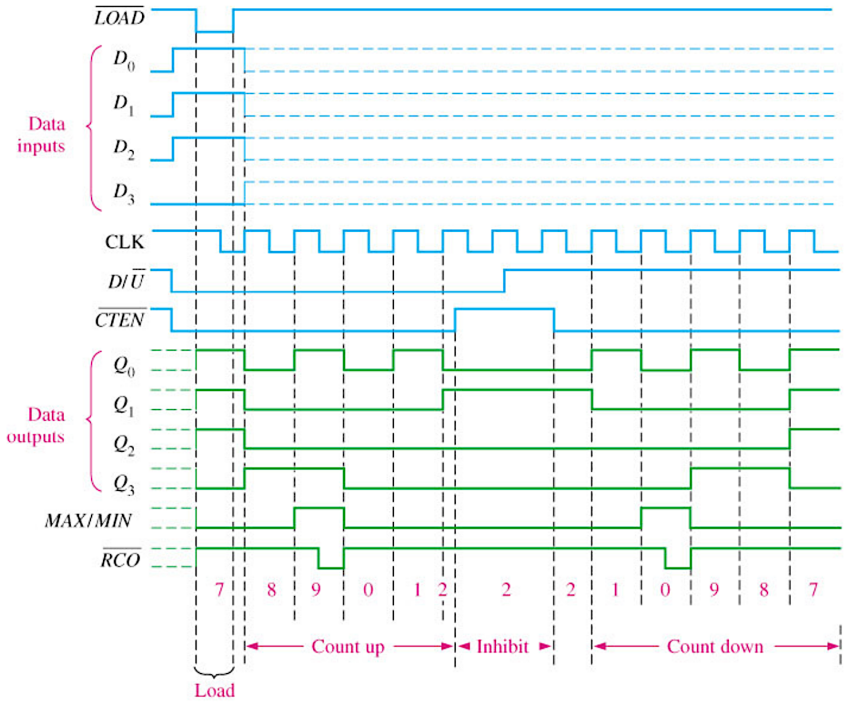
Repare:
Pin $\overline{RCO}$ → pin $\overline{PL}$

Sequência assumida:
0 → 1 → 2 → 3 → 4 → 5 → 6 → 8 → 9/5 →
→ 6 → 7 → 8 → 9/5 → 6 → 7 → 8 → …



Ex.2: Contador decrescente de 5 → 1:





Repare:

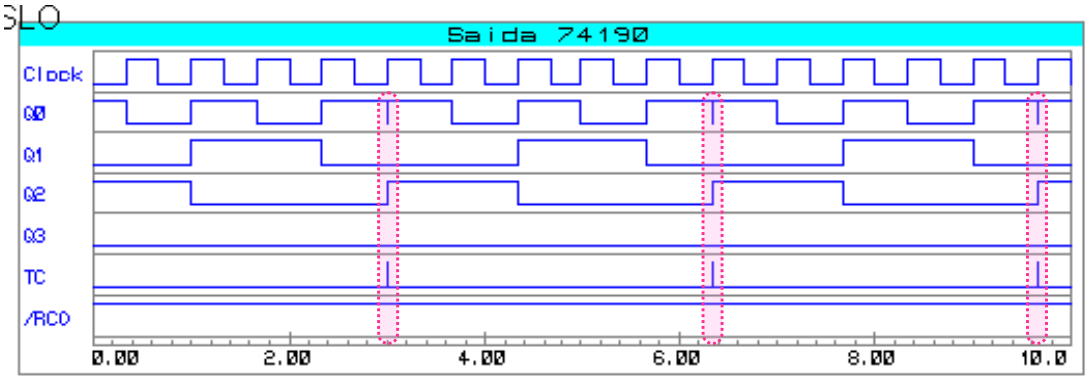
$\overline{PL} = \overline{TC}$

$P_3P_2P_1P_0 = 0101_{(2)} = 5_{(10)}$

$D/\overline{U} = 1$

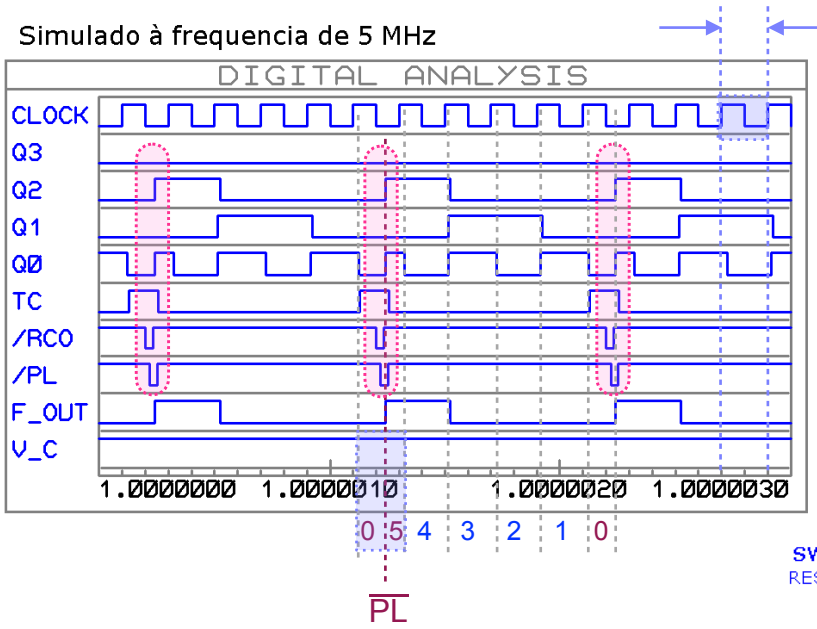
Sequência assumida:

5 → 4 → 3 → 2 → 1 → 0/5 → 4 → 3 → 2 →
→ 1 → 0/5 → 4 → 3 → 2 → 1 → …

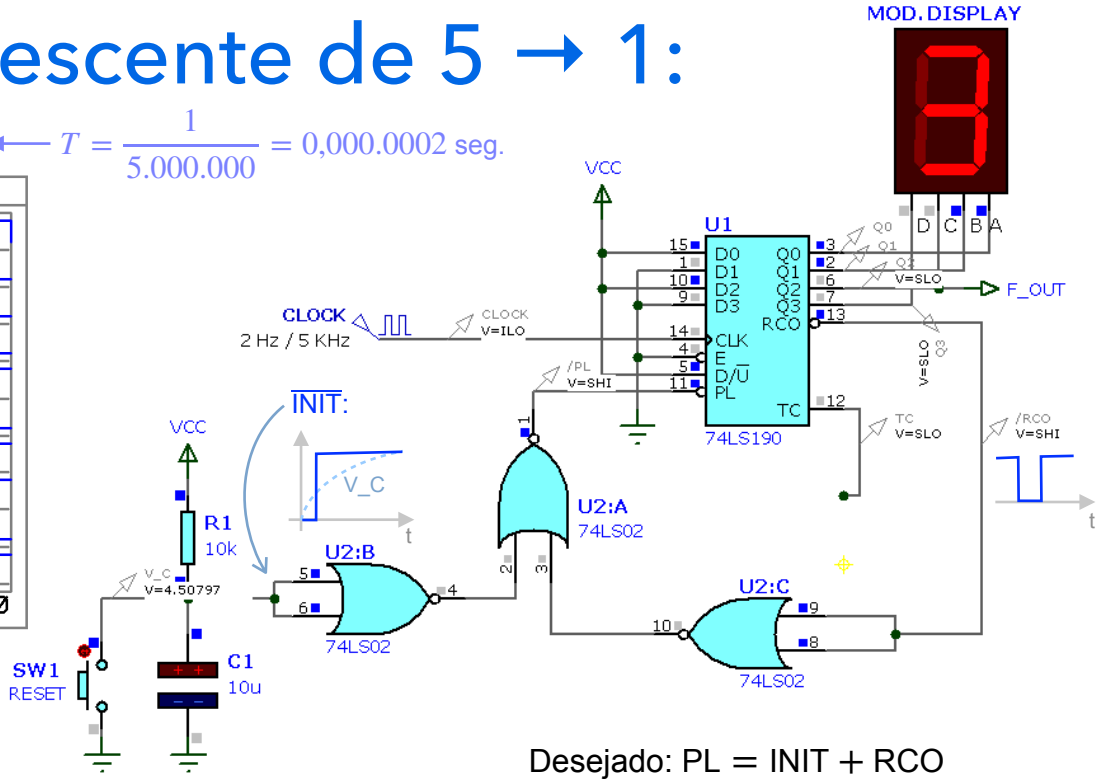


Ex.3: Contador decrescente de 5 → 1:

Simulado à frequencia de 5 MHz



$$T = \frac{1}{5.000.000} = 0{,}000.0002 \text{ seg.}$$



Repare:

$\overline{PL} \leftarrow \overline{RCO}$

$P_3P_2P_1P_0 = 0101_{(2)} = 5_{(10)}$

$D/\overline{U} = 1$

Desejado: $PL = INIT + RCO$

Deve-se gerar $\overline{PL}$:

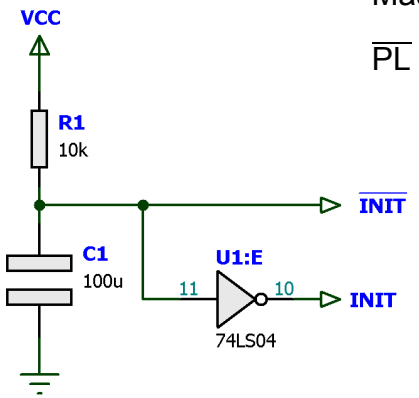
$\overline{PL} = \overline{INIT + RCO}$

Mas disponíveis: $\overline{INIT}$, $\overline{RCO}$:

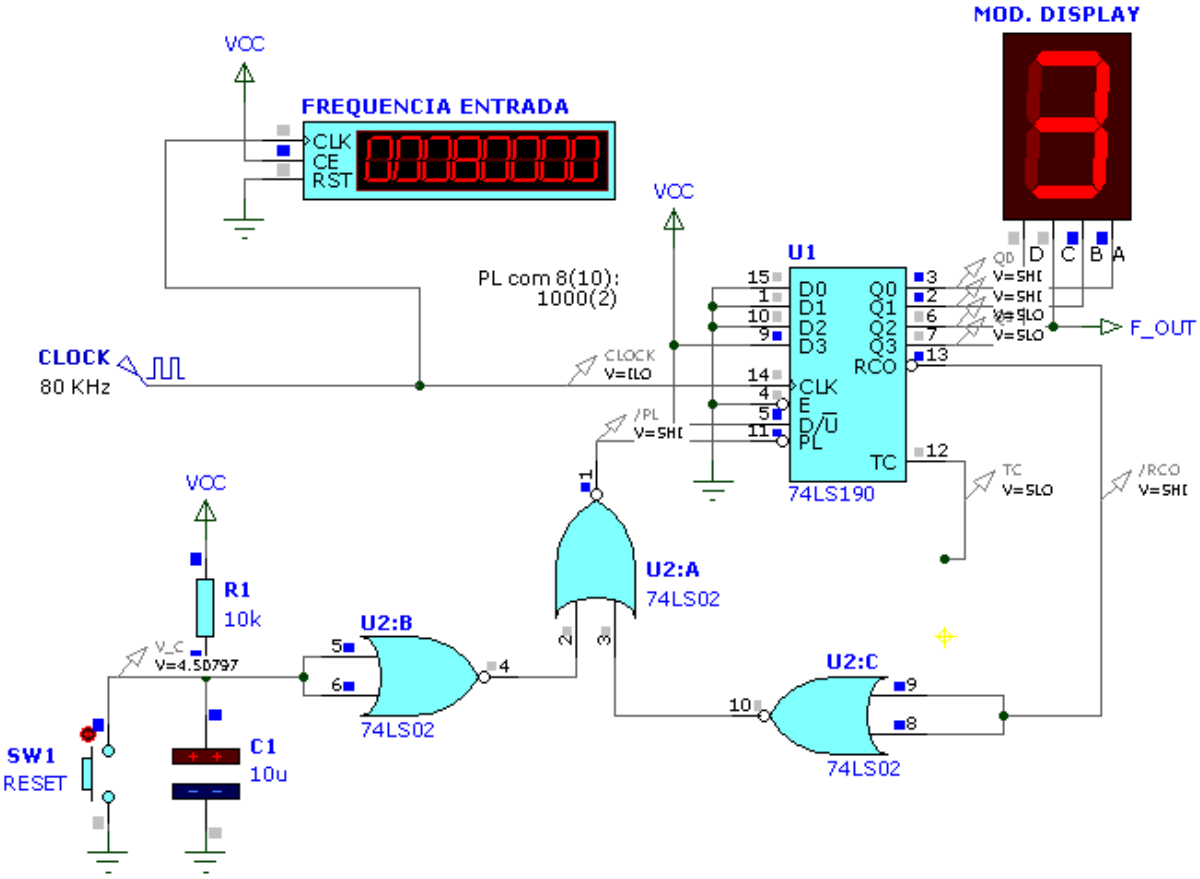
$\overline{PL} = \overline{\overline{INIT} + \overline{RCO}}$

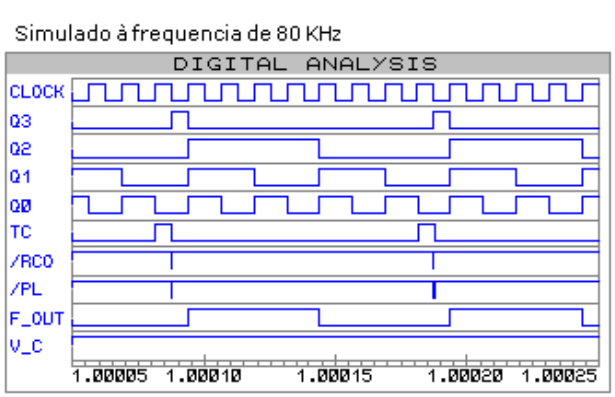
Sequência assumida:

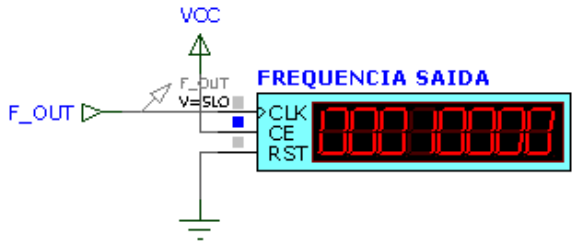
5 → 4 → 3 → 2 → 1 → 0/5 → 4 → 3 → 2 →
→ 1 → 0/5 → 4 → 3 → 2 → 1 → …



Ex.4: Divisor de frequência, *f* ÷ 8:







Pastilhas 74192/74193:

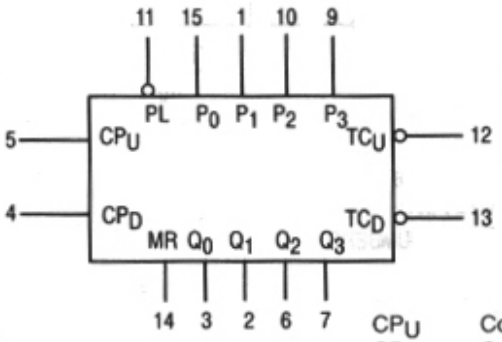
MOTOROLA

PRESETTABLE BCD/DECADE UP/DOWN COUNTER
PRESETTABLE 4-BIT BINARY UP/DOWN COUNTER

The SN54/74LS192 is an UP/DOWN BCD Decade (8421) Counter and the SN54/74LS193 is an UP/DOWN MODULO-16 Binary Counter. Separate Count Up and Count Down Clocks are used and in either counting mode the circuits operate synchronously. The outputs change state synchronous with the LOW-to-HIGH transitions on the clock inputs.

Separate Terminal Count Up and Terminal Count Down outputs are provided which are used as the clocks for a subsequent stages without extra logic, thus simplifying multistage counter designs. Individual preset inputs allow the circuits to be used as programmable counters. Both the Parallel Load ($\overline{PL}$) and the Master Reset (MR) inputs asynchronously override the clocks.

- Low Power . . . 95 mW Typical Dissipation
- High Speed . . . 40 MHz Typical Count Frequency
- Synchronous Counting
- Asynchronous Master Reset and Parallel Load
- Individual Preset Inputs
- Cascading Circuitry Internally Provided
- Input Clamp Diodes Limit High Speed Termination Effects



V_{CC} = PIN 16
GND = PIN 8

CP_U	Count Up Clock Pulse Input
CP_D	Count Down Clock Pulse Input
MR	Asynchronous Master Reset (Clear) Input
$\overline{PL}$	Asynchronous Parallel Load (Active LOW) Input
P_n	Parallel Data Inputs
Q_n	Flip-Flop Outputs (Note b)
$\overline{TC}_D$	Terminal Count Down (Borrow) Output (Note b)
$\overline{TC}_U$	Terminal Count Up (Carry) Output (Note b)

NOTES:
a. 1 TTL Unit Load (U.L.) = 40 µA HIGH/1.6 mA LOW.
b. The Output LOW drive

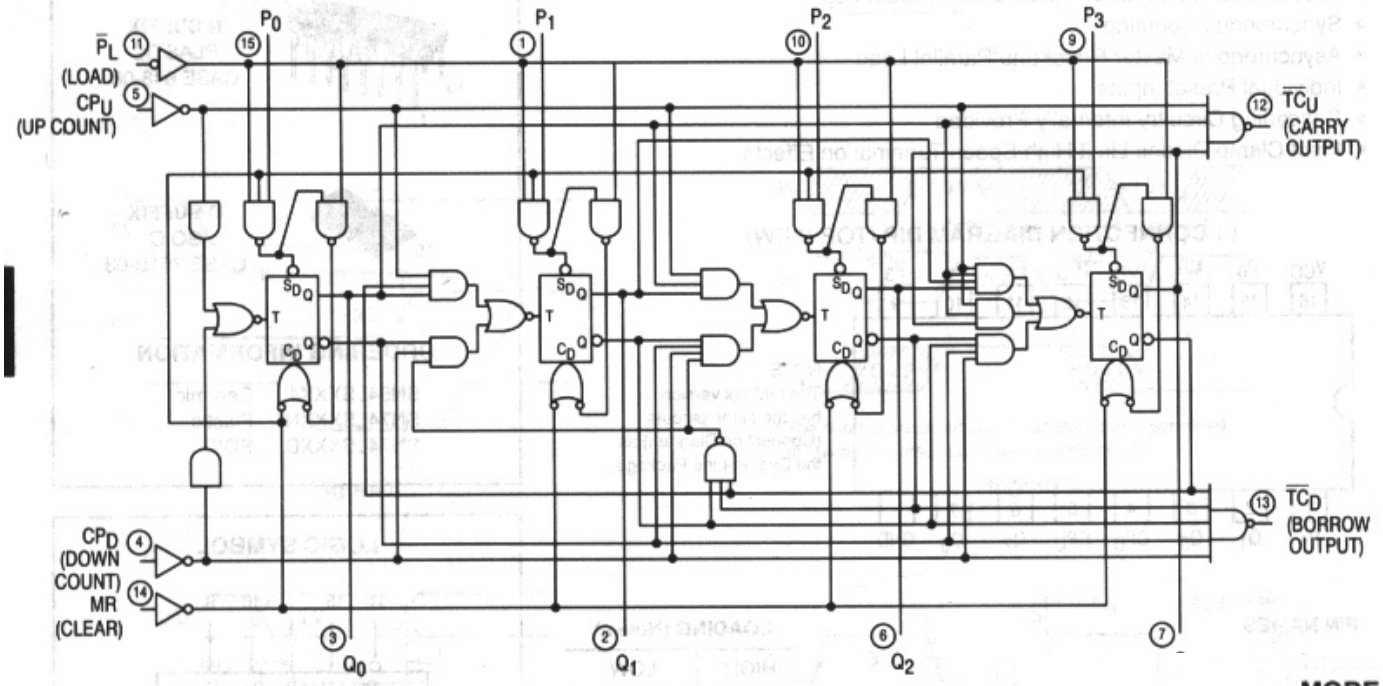
MODE SELECT TABLE

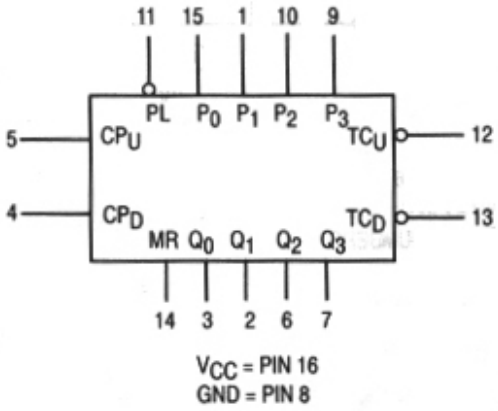
MR	$\overline{PL}$	CP_U	CP_D	MODE
H	X	X	X	Reset (Asyn.)
L	L	X	X	Preset (Asyn.)
L	H	H	H	No Change
L	H	⌐	H	Count Up
L	H	H	⌐	Count Down

L = LOW Voltage Level
H = HIGH Voltage Level
X = Don't Care
⌐ = LOW-to-HIGH Clock Transition

Pastilhas 74192/74193:

LOGIC DIAGRAMS





V_{CC} = PIN 16
GND = PIN 8

V_{CC} = PIN 16
GND = PIN 8
◯ = PIN NUMBERS

CP_U	Count Up Clock Pulse Input
CP_D	Count Down Clock Pulse Input
MR	Asynchronous Master Reset (Clear) Input
$\overline{PL}$	Asynchronous Parallel Load (Active LOW) Input
P_n	Parallel Data Inputs
Q_n	Flip-Flop Outputs (Note b)
$\overline{TC}_D$	Terminal Count Down (Borrow) Output (Note b)
$\overline{TC}_U$	Terminal Count Up (Carry) Output (Note b)

NOTES:
a. 1 TTL Unit Load (U.L.) = 40 µA HIGH/1.6 mA LOW.
b. The Output LOW drive

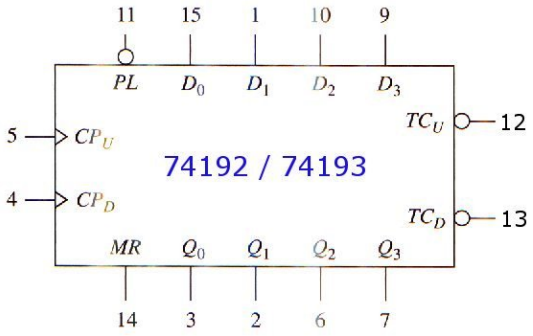
MODE SELECT TABLE

MR	$\overline{PL}$	CP_U	CP_D	MODE
H	X	X	X	Reset (Asyn.)
L	L	X	X	Preset (Asyn.)
L	H	H	H	No Change
L	H	⎍	H	Count Up
L	H	H	⎍	Count Down

L = LOW Voltage Level
H = HIGH Voltage Level
X = Don't Care
⎍ = LOW-to-HIGH Clock Transition

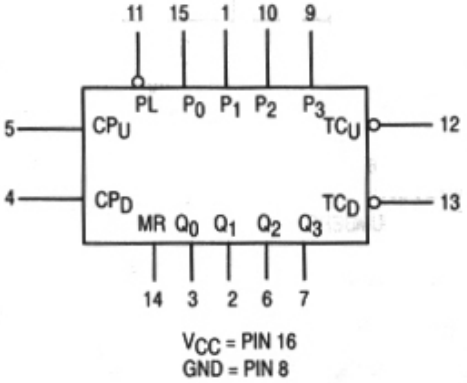
Pastilhas comerciais:

- Contador síncrono up/down: 74HC192/74HC193

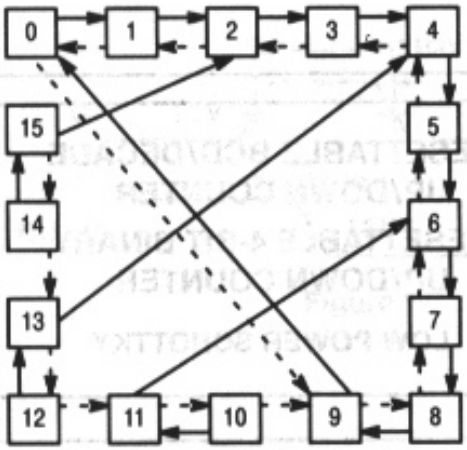


CP_U	Count Up Clock Pulse Input
CP_D	Count Down Clock Pulse Input
MR	Asynchronous Master Reset (Clear) Input
$\overline{PL}$	Asynchronous Parallel Load (Active LOW) Input
P_n	Parallel Data Inputs
Q_n	Flip-Flop Outputs (Note b)
$\overline{TC_D}$	Terminal Count Down (Borrow) Output (Note b)
$\overline{TC_U}$	Terminal Count Up (Carry) Output (Note b)

NOTES:
a. 1 TTL Unit Load (U.L.) = 40 µA HIGH/1.6 mA LOW.
b. The Output LOW drive



STATE DIAGRAMS



LS192

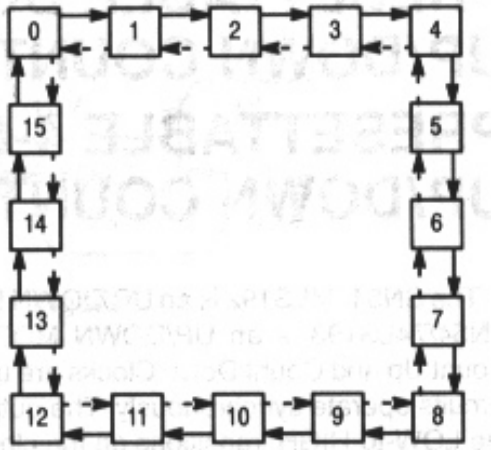
LS192 LOGIC EQUATIONS FOR TERMINAL COUNT

$$\overline{TC_U} = Q_0 \cdot Q_3 \cdot \overline{CP_U}$$
$$\overline{TC_D} = Q_0 \cdot Q_1 \cdot Q_2 \cdot Q_3 \cdot \overline{CP_D}$$

LS193 LOGIC EQUATIONS FOR TERMINAL COUNT

$$\overline{TC_U} = Q_0 \cdot Q_1 \cdot Q_2 \cdot Q_3 \cdot \overline{CP_U}$$
$$\overline{TC_D} = Q_0 \cdot Q_1 \cdot Q_2 \cdot Q_3 \cdot \overline{CP_D}$$

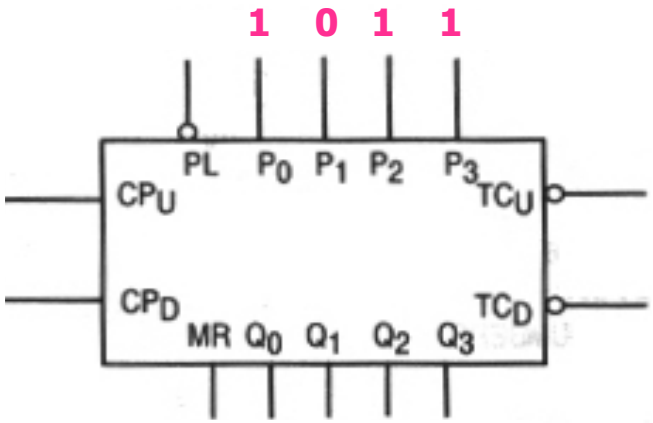
COUNT UP ———
COUNT DOWN - - - - -

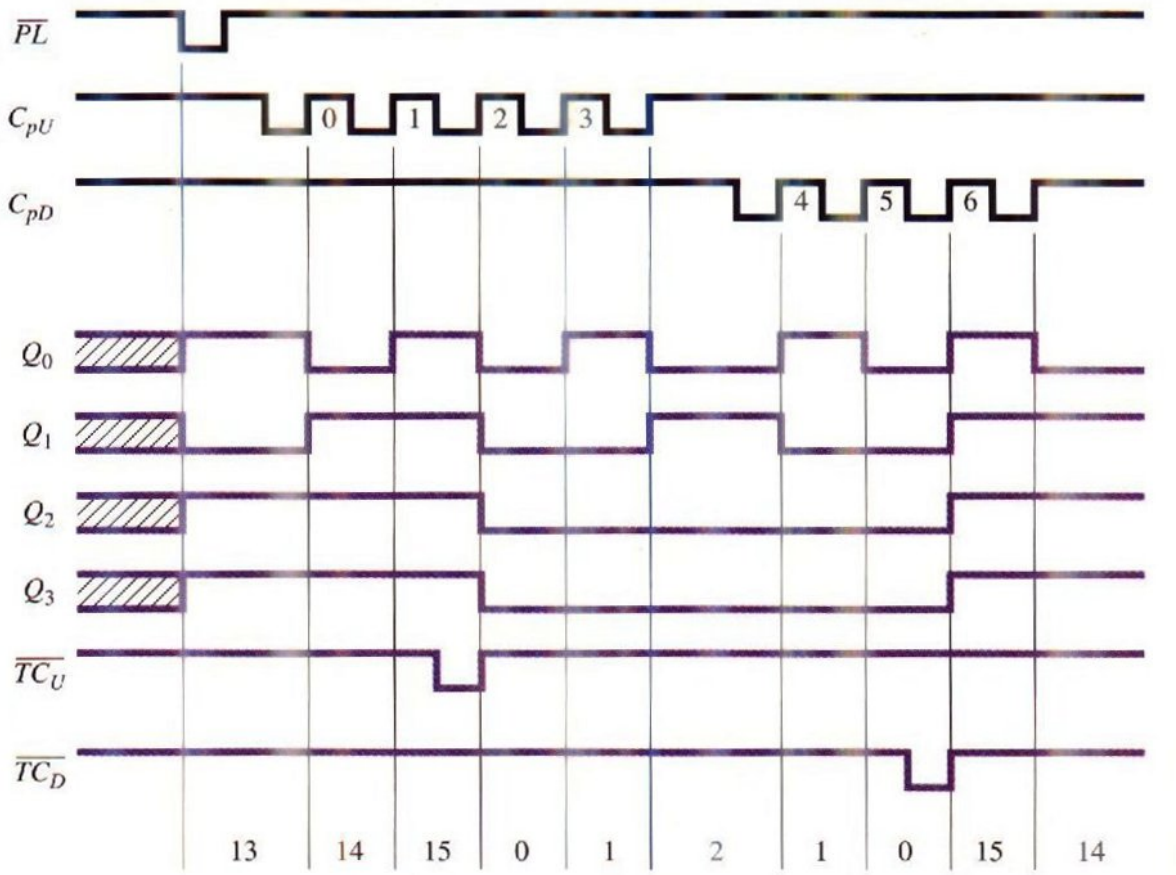


LS193

Usando o CI 74193:

Complete as formas de onda nas saídas do 74193:

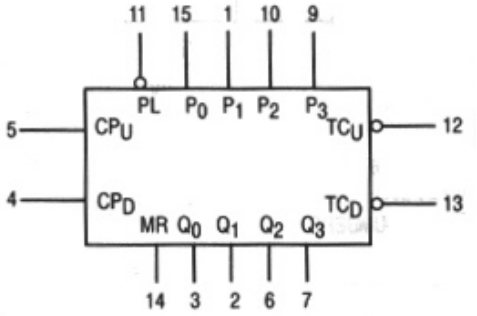




Usando o CI 74192:

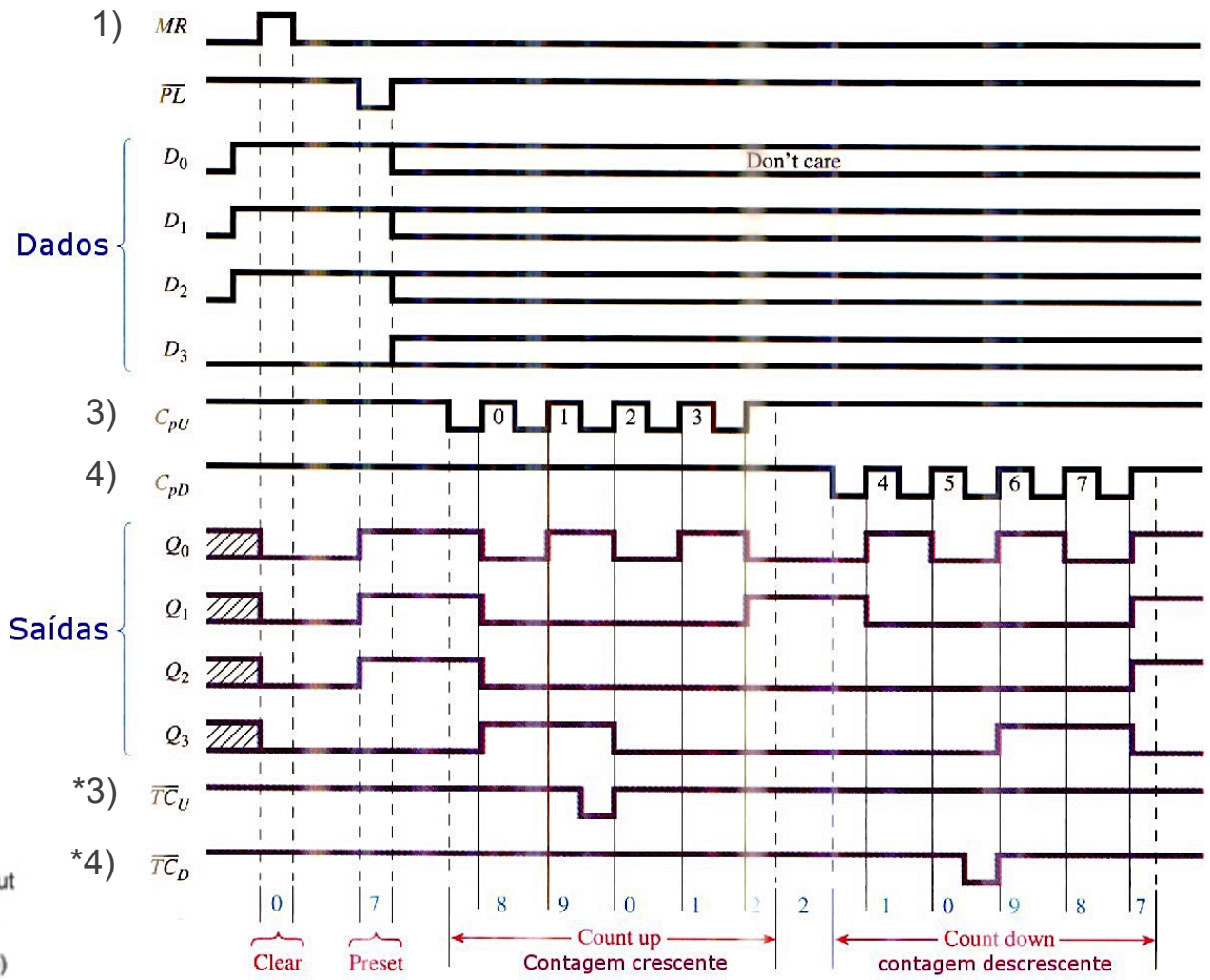
Que acontece quando ?

1) Resseto todas as suas saídas;
2) Realizo carga paralela com 7(10)=0111(2);
3) Ajusto para contar crescente +5 pulsos de Clock;
4) Ajusto para contar decrescente + 5 pulsos de



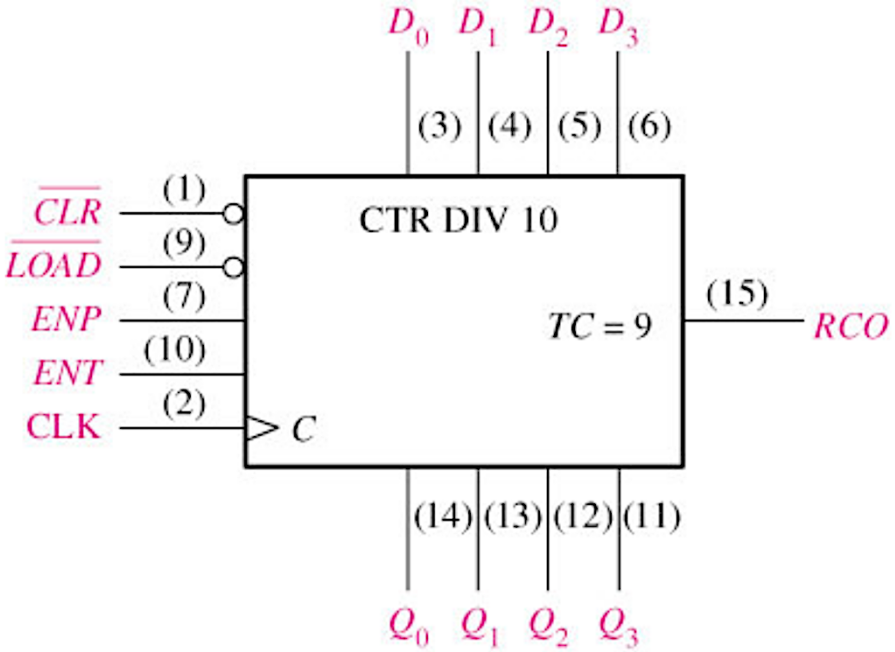
CP_U	Count Up Clock Pulse Input
CP_D	Count Down Clock Pulse Input
MR	Asynchronous Master Reset (Clear) Input
$\overline{PL}$	Asynchronous Parallel Load (Active LOW) Input
P_n	Parallel Data Inputs
Q_n	Flip-Flop Outputs (Note b)
$\overline{TC}_D$	Terminal Count Down (Borrow) Output (Note b)
$\overline{TC}_U$	Terminal Count Up (Carry) Output (Note b)

NOTES:
a. 1 TTL Unit Load (U.L.) = 40 µA HIGH/1.6 mA LOW.
b. The Output LOW drive



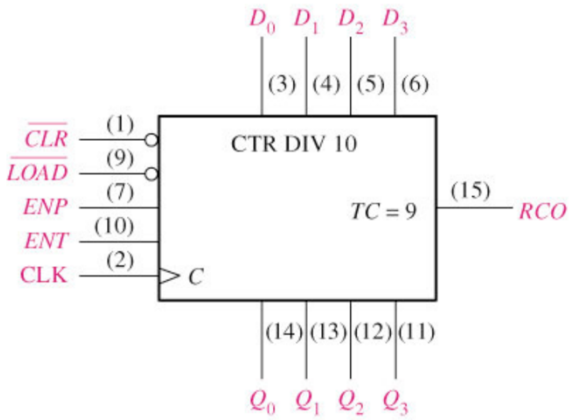
Pastilhas comerciais:

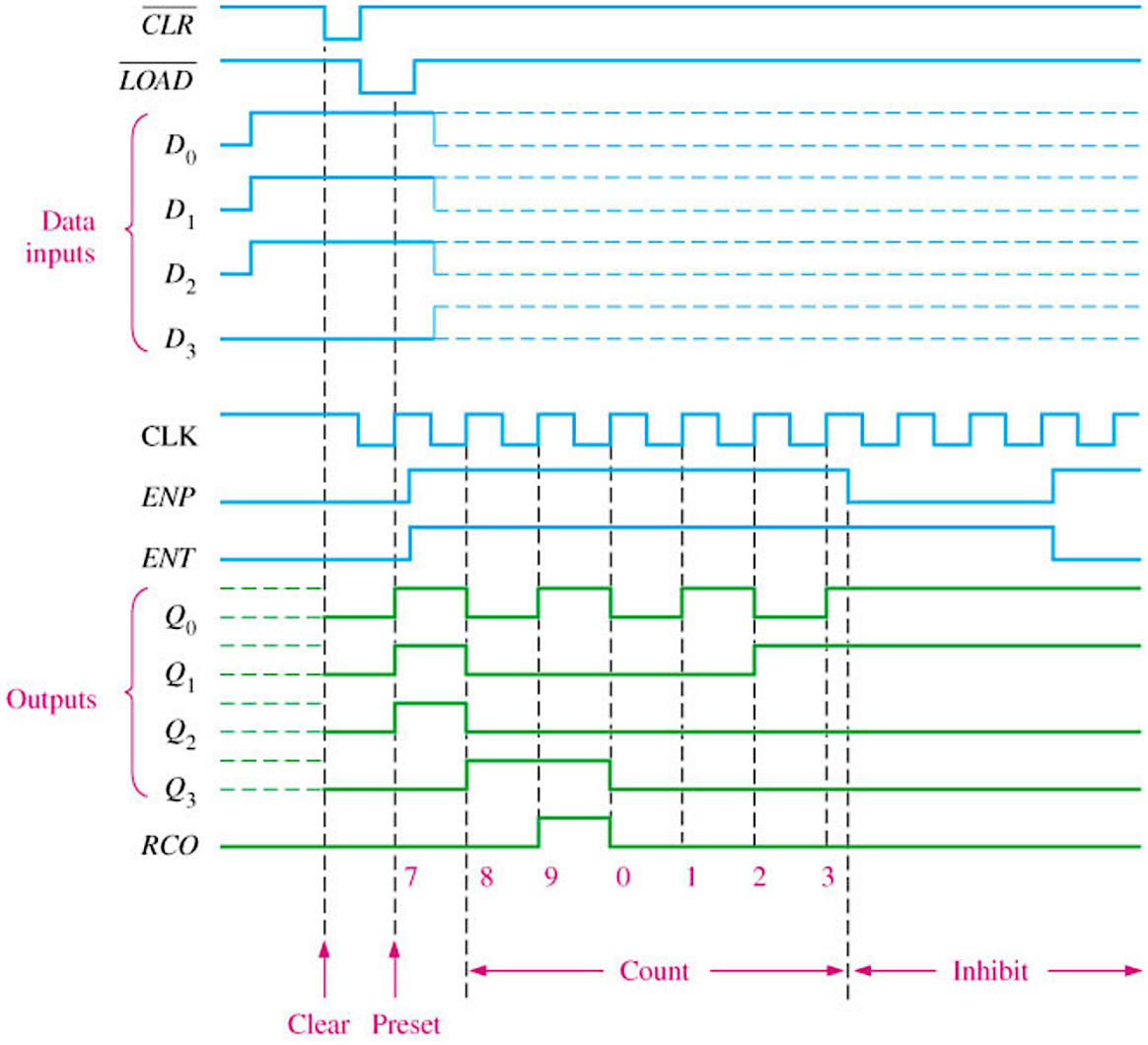
- Contador síncrono decimal up/down: 74HC160



Pastilhas comerciais:

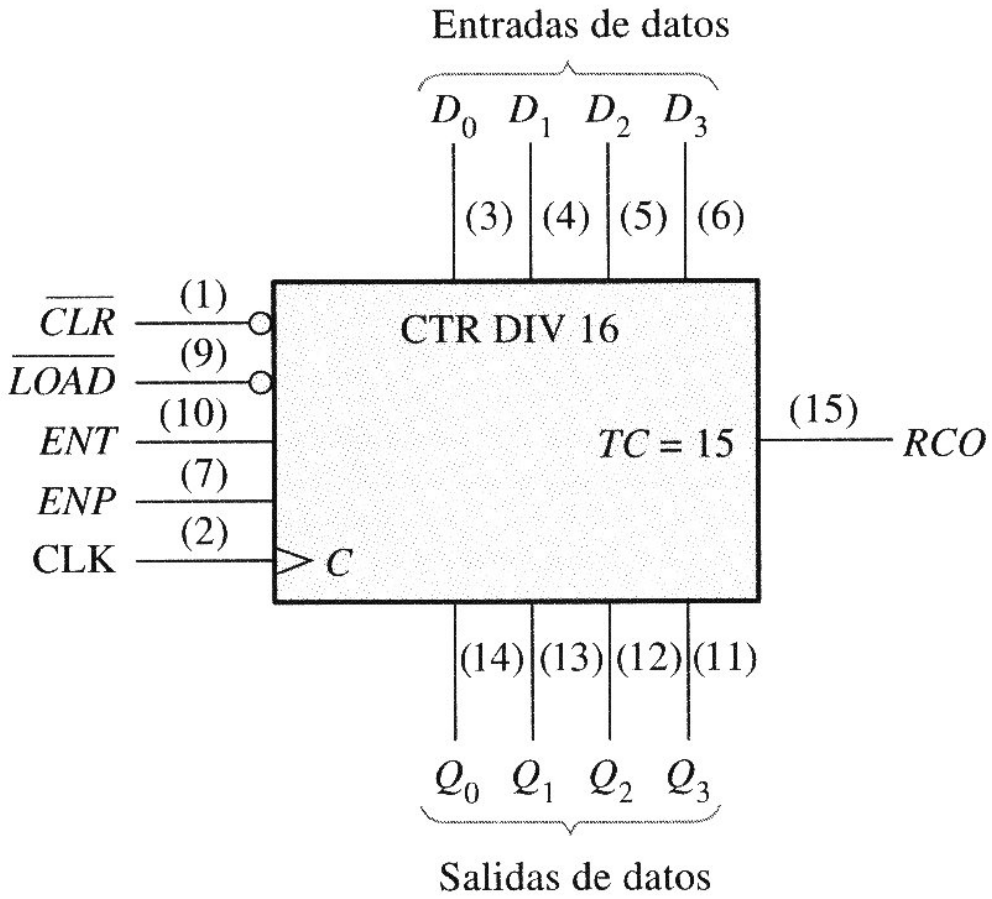
- Contador síncrono decimal up/down: 74HC160





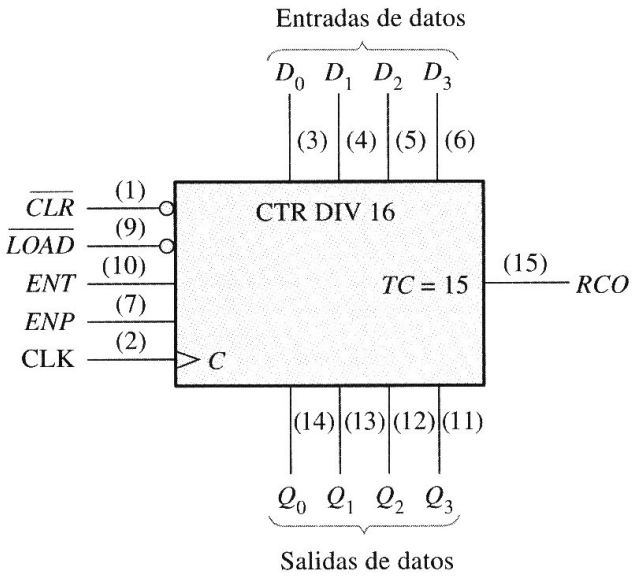
Pastilhas comerciais:

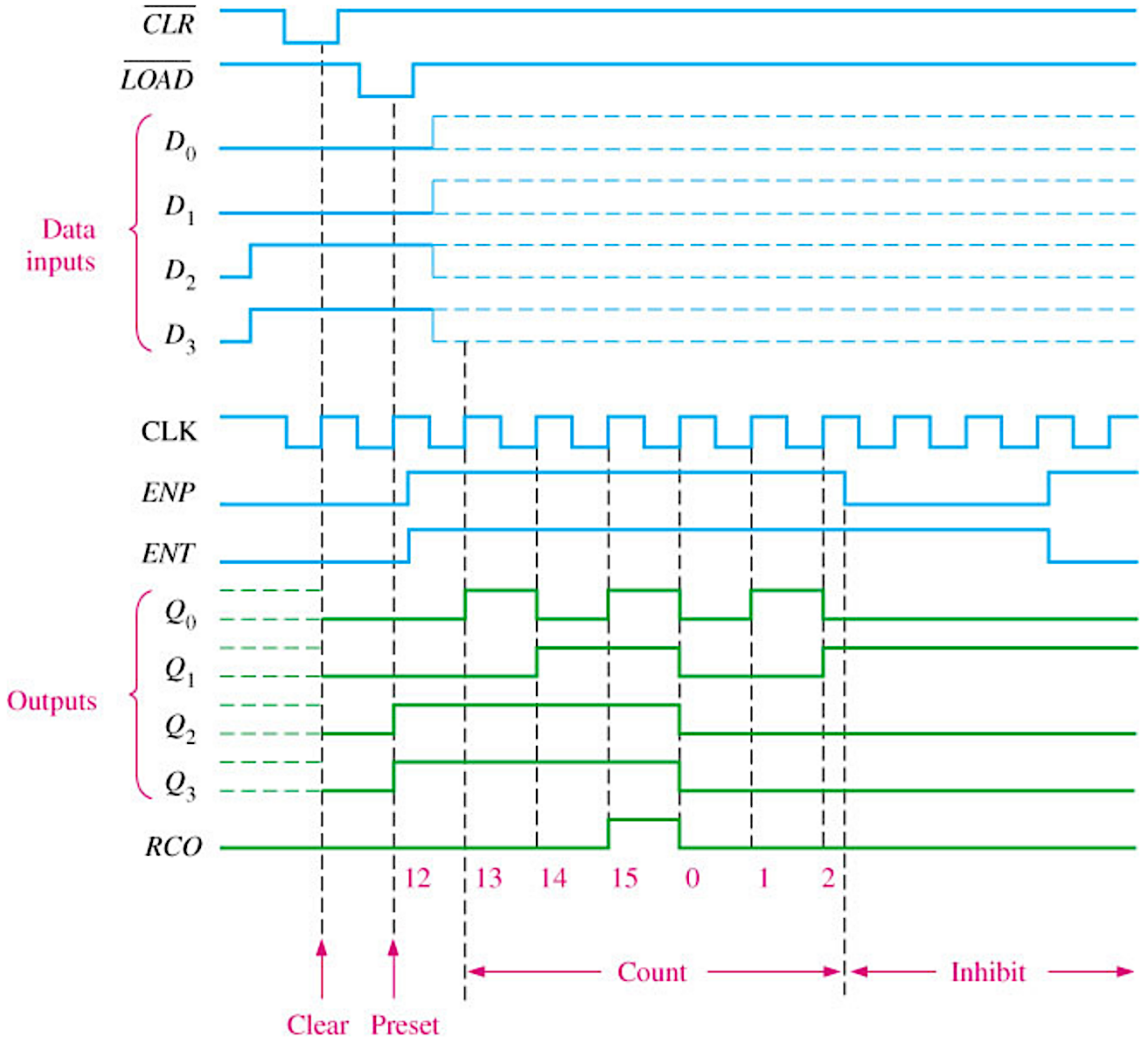
- Contador síncrono binario up/down: 74HC163



Pastilhas comerciais:

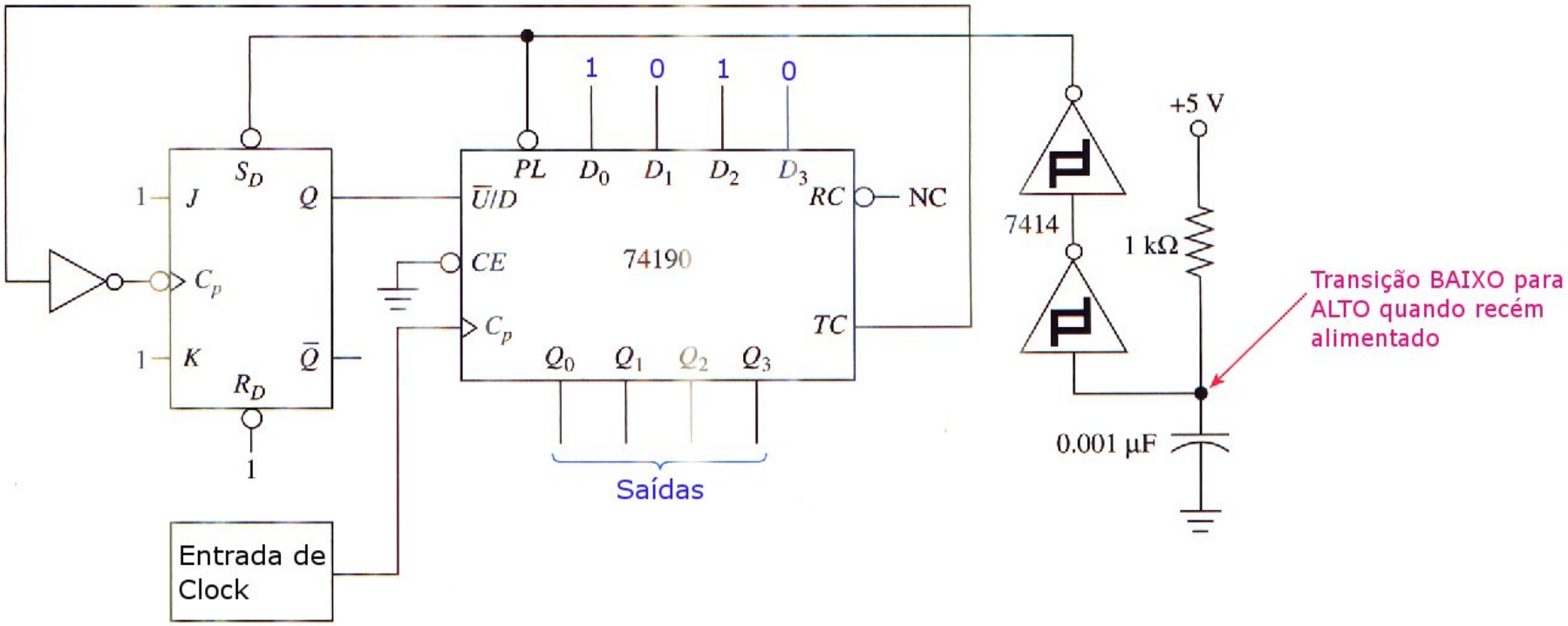
- Contador síncrono binario up/down: 74HC163



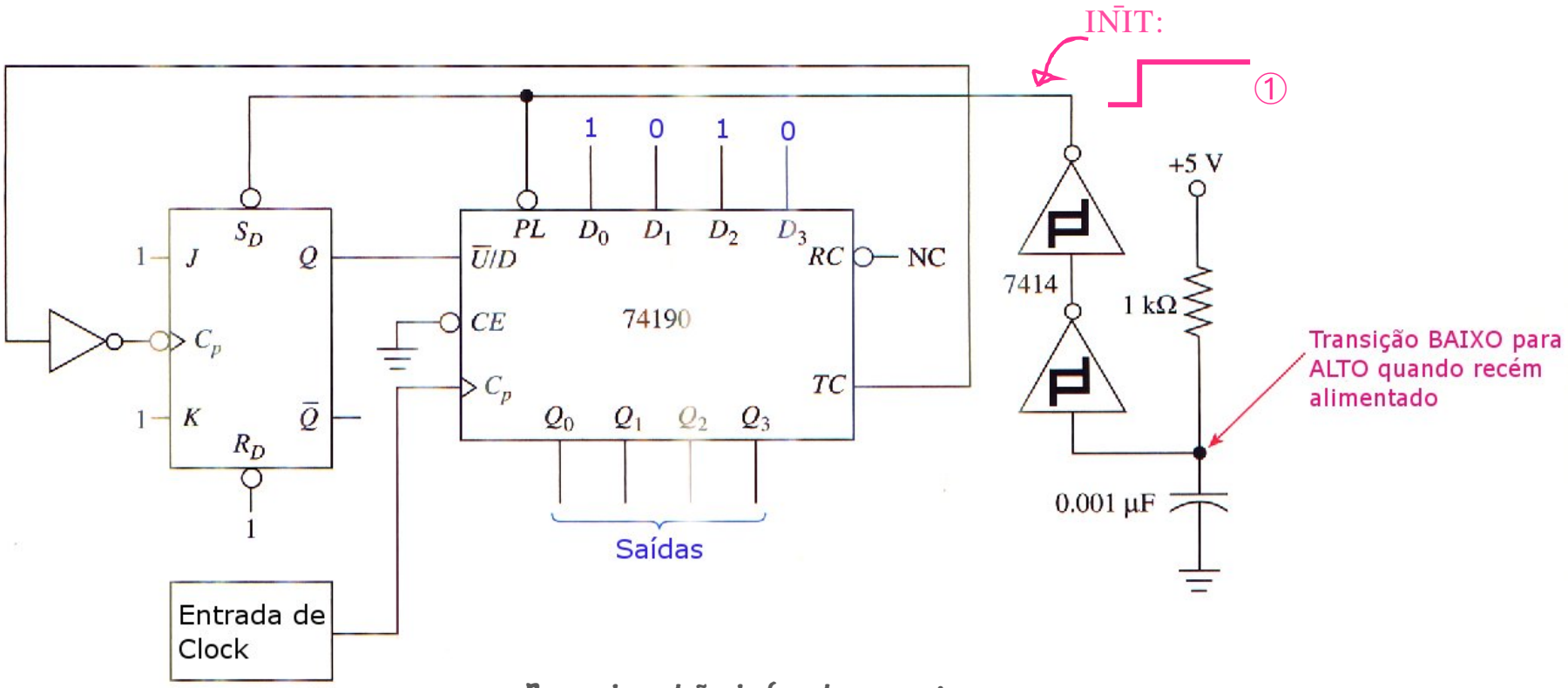


Problema

O que faz o circuito abaixo?

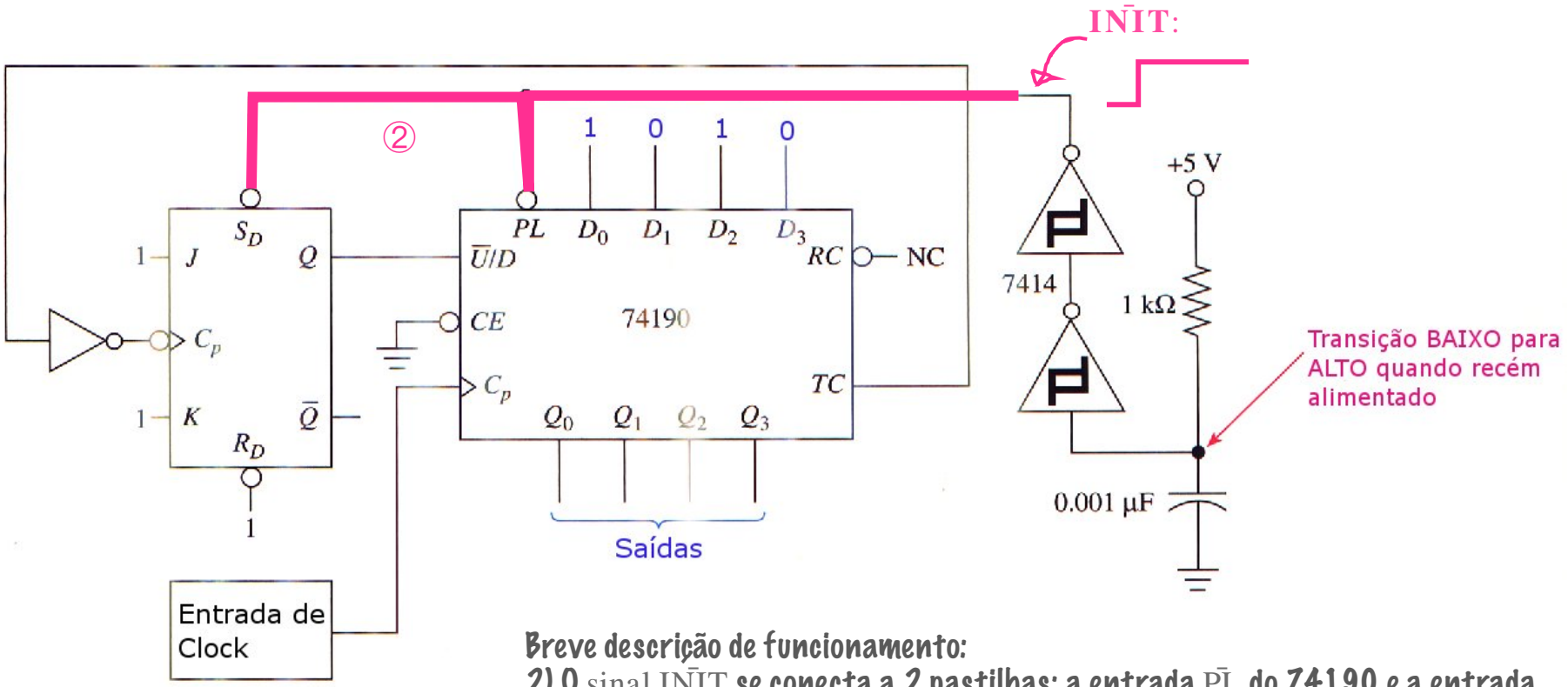


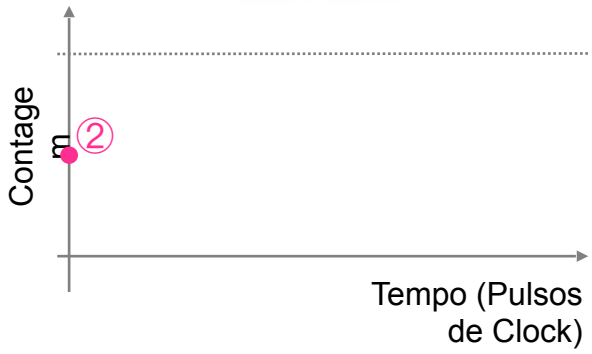
Problema



Breve descrição de funcionamento:
1) O circuito formado pelo Resistor e Capacitor gera um pulso que inicia e continua em nível lógico baixo na saída das portas Schmidt trigger enquanto o capacitor carrega. Chamaremos a este pulso de INĪT. Notar que capacitor carrega rápido, então o sinal INĪT é curto. Uma vez carregado, INĪT permanece em nível lógico alto.

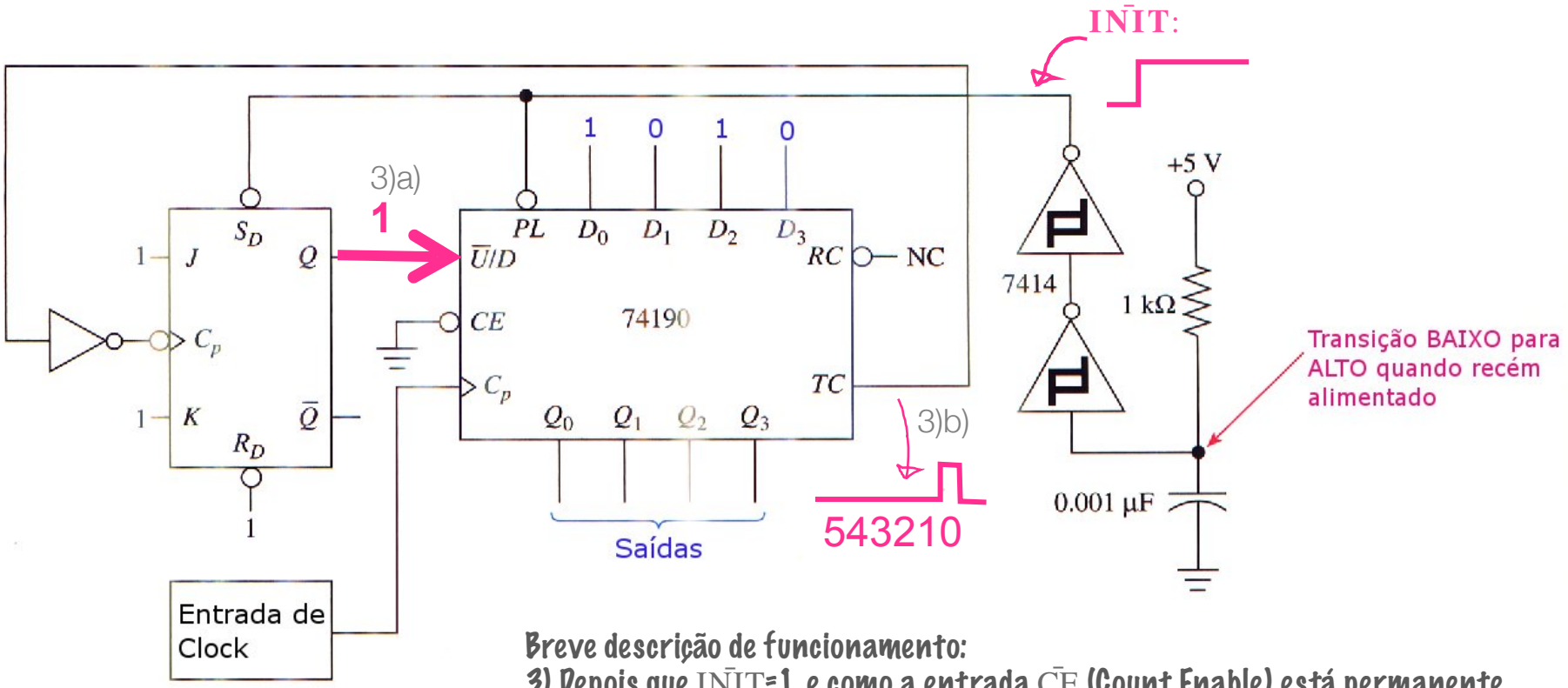
Problema

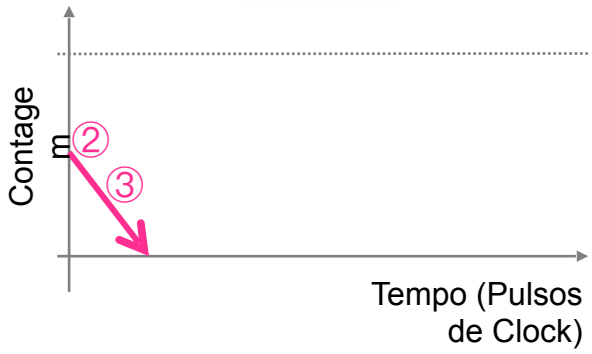




Breve descrição de funcionamento:
2) O sinal INIT **se conecta a 2 pastilhas: a entrada** PL **do 74190 e a entrada** $\bar{S}_d$ **do FF-JK. Ambas são assíncronas (não dependem do sinal do clock). Isto força respectivamente o "Parallel Load" do contador com o número** $D_3D_2D_1D_0=0101_{(2)}=5_{(10)}$ **e o "Set" do FF-JK. Desta forma, isto garante que o contador inicie a contagem no número 5. Note que saída do FF-JK controla o sentido de contagem do contador. Como neste caso o FF é setado, Q=1, a entrada** Ū/D=1, **forçando o contado num contagem descrescente (Down), tão logo o pulso** INIT **cesse.**

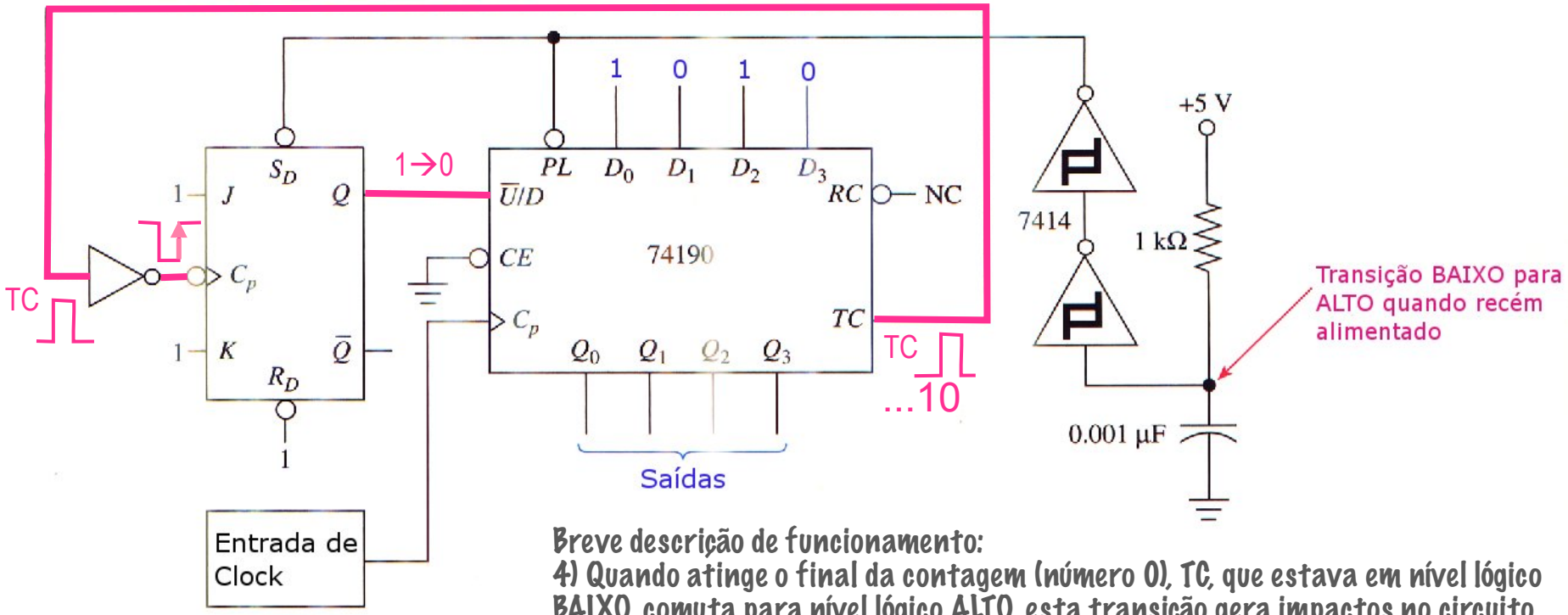
Problema

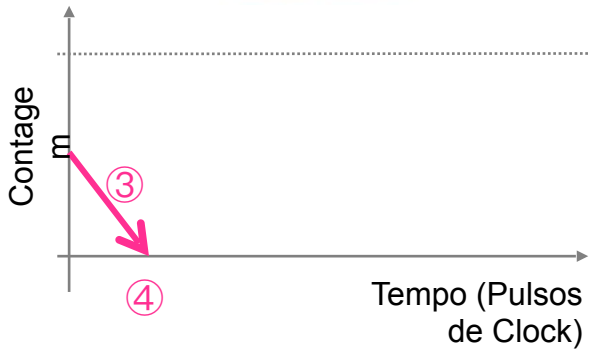




Breve descrição de funcionamento:
3) Depois que INIT=1, e como a entrada CE (Count Enable) está permanente ativada (=0), o contador começa a contar, de maneira decrescente (a) já que U/D =1 (o FF-JK está setado). Esta condição de contagem continua até o contador atingir o final de contagem, indicado por um novo pulso na saída TC (b).

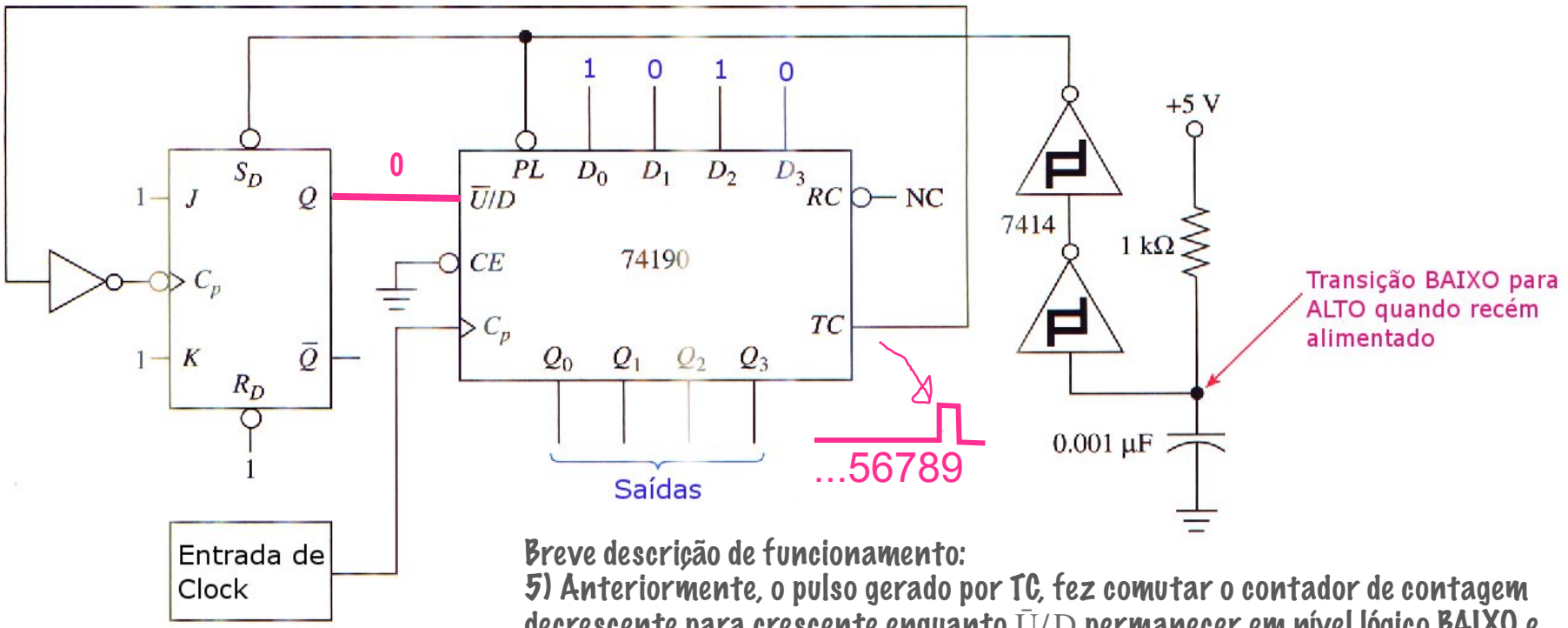
Problema

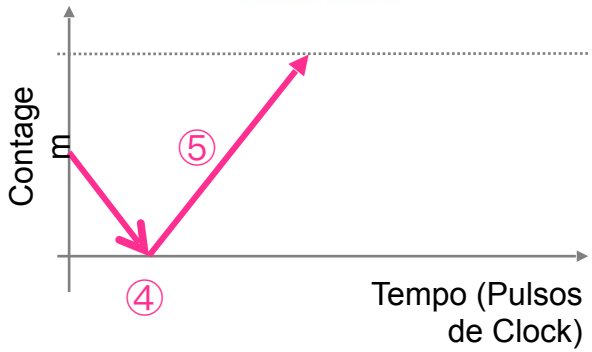




Breve descrição de funcionamento:
4) Quando atinge o final da contagem (número 0), TC, que estava em nível lógico BAIXO, comuta para nível lógico ALTO, esta transição gera impactos no circuito. Note que TC está conectado à entrada de Clock (borda de subida) do FF-JK (que está na configuração toggle). Quando o contador vai reciclar (baixa TC de nível lógico ALTO para BAIXO), a porta NOT na entrada de Clock do JK, faz gerar uma borda de subida no JK, o que por sua vez (dada a configuração toggle do JK), faz alternar sua saída Q de 1 para 0, e assim a contagem passar agora à ser crescente, já que $\bar{U}/D$ passa para nível lógico BAIXO ($\bar{U}p$).

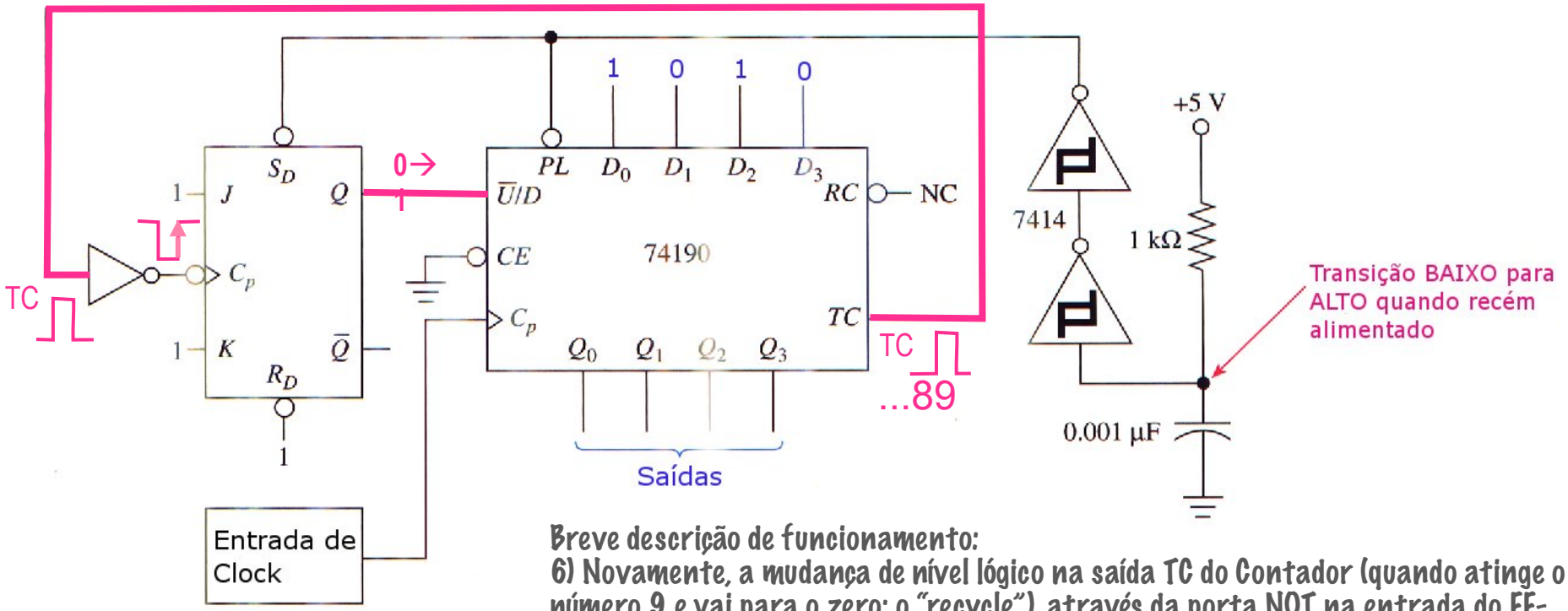
Problema

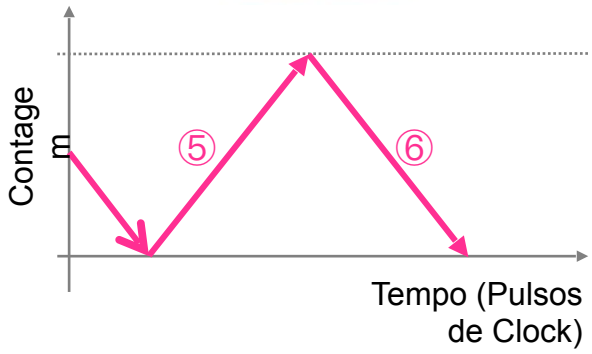




Breve descrição de funcionamento:
5) Anteriormente, o pulso gerado por TC, fez comutar o contador de contagem decrescente para crescente enquanto $\bar{U}/D$ permanecer em nível lógico BAIXO e assim o contador passa a contar agora até 0 até o número 9 (crescente). Quando atingir o número ((final da contagem no modo crescente), note que um novo pulso é gerada na saída TC.

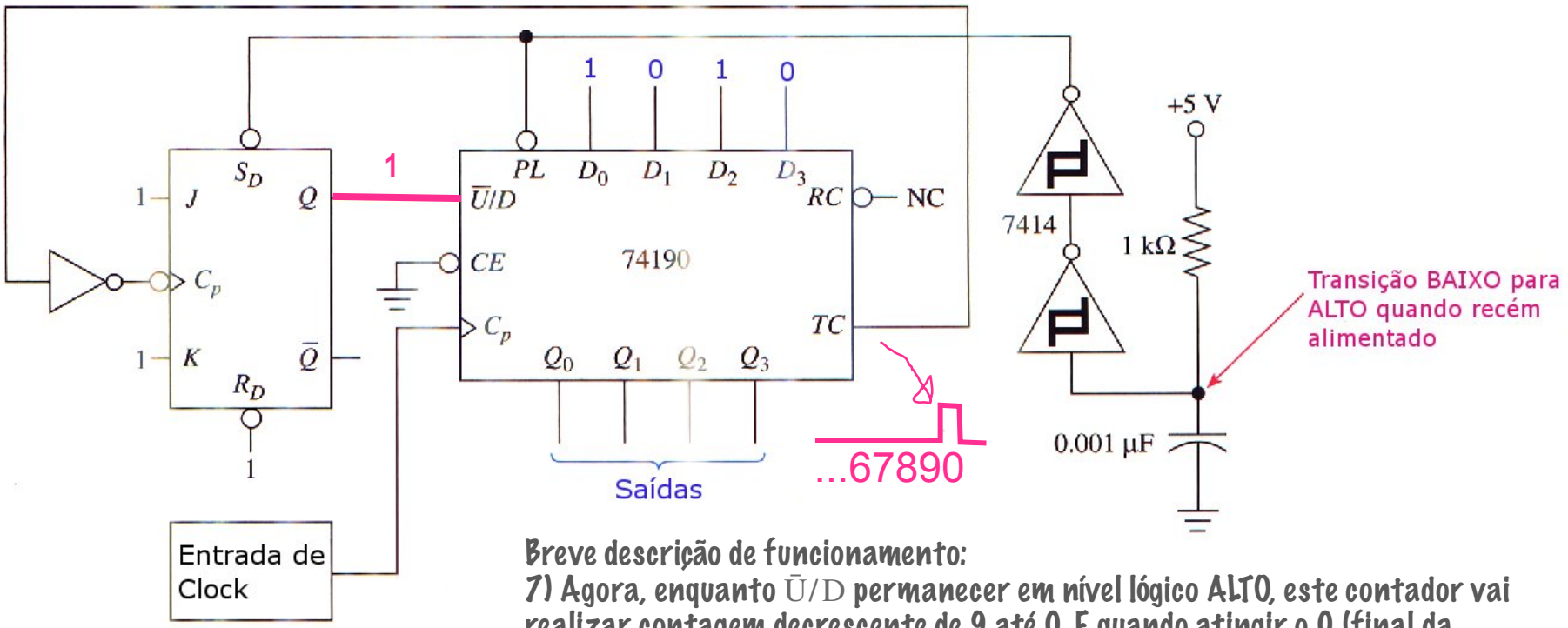
Problema

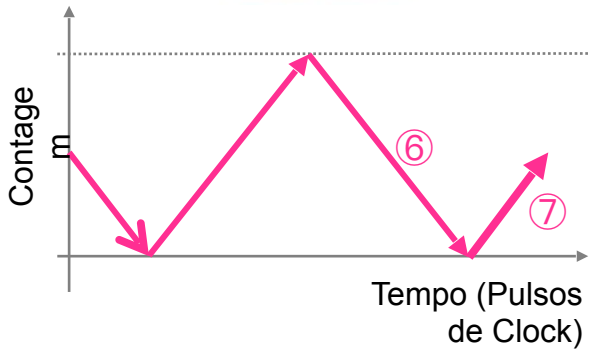




Breve descrição de funcionamento:
6) Novamente, a mudança de nível lógico na saída TC do Contador (quando atinge o número 9 e vai para o zero; o "recycle"), através da porta NOT na entrada do FF-JK (que está em configuração toggle) faz alternar a saída do mesmo de 0 para 1, e assim, na próxima borda de subida do sinal de Clock, o Contador vai realizar contagem decrescente (já que agora Ū/D está em 1).

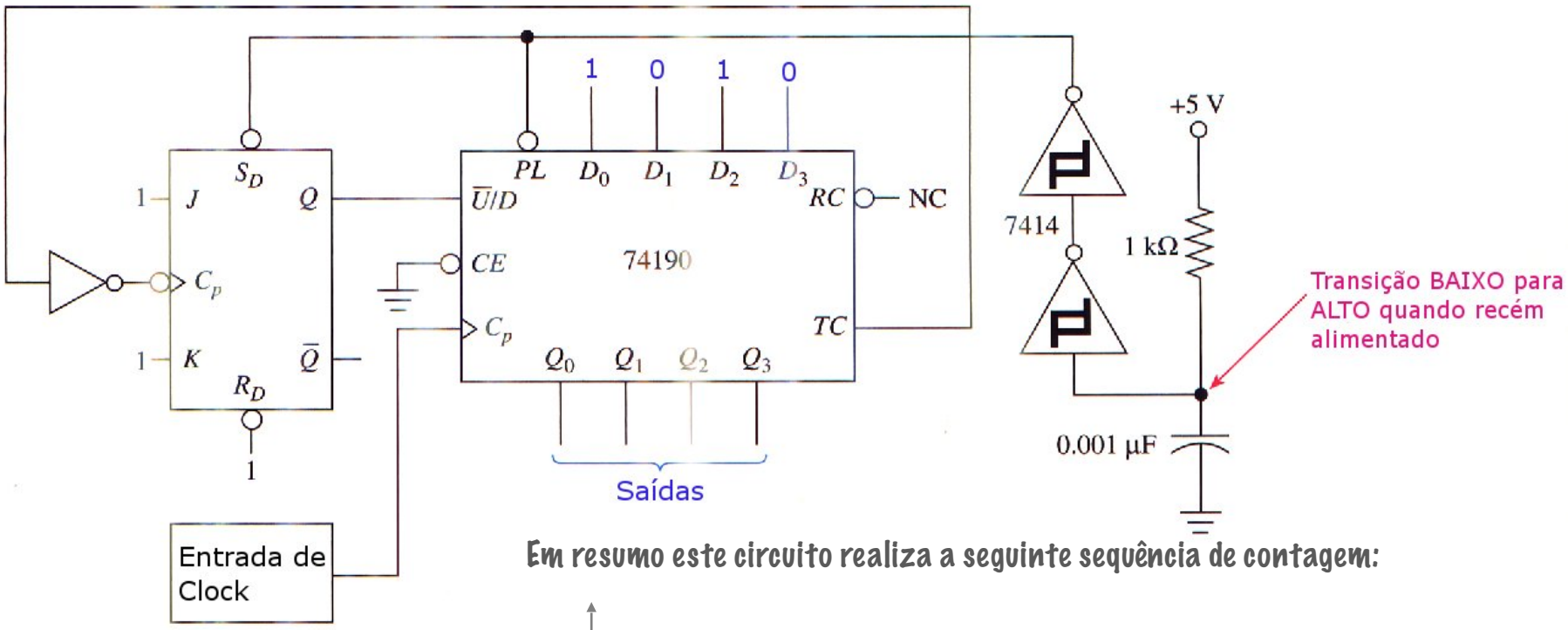
Problema



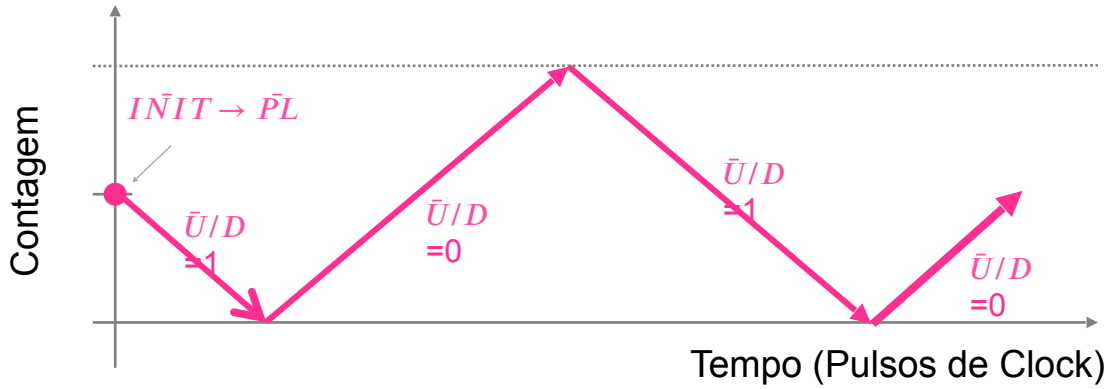


Breve descrição de funcionamento:
7) Agora, enquanto $\bar{U}/D$ permanecer em nível lógico ALTO, este contador vai realizar contagem decrescente de 9 até 0. E quando atingir o 0 (final da contagem), o novo pulso gerado na saída TC vai fazer com que o Contador alterne novamente a contagem de decrescente à crescente. E assim o ciclo de contagem segue indefinidamente alternando contagens decrescentes com crescentes enquanto o circuito permanecer alimentado.

Problema

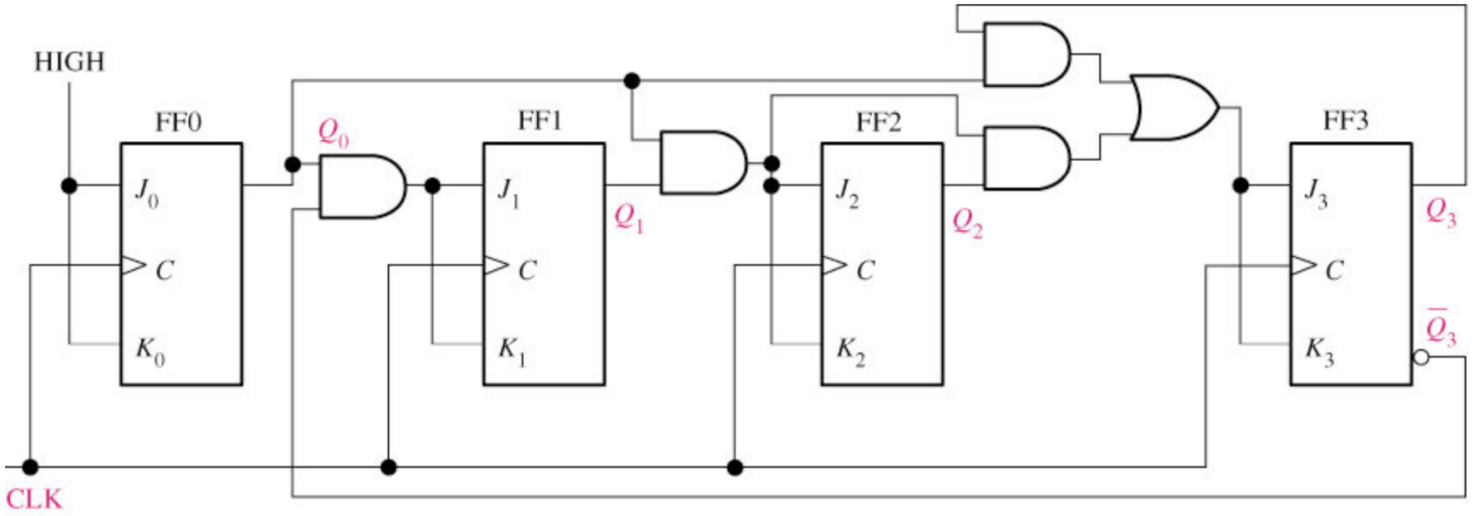


Em resumo este circuito realiza a seguinte sequência de contagem:



Projetos:

- Exemplo: contador síncrono decimal:



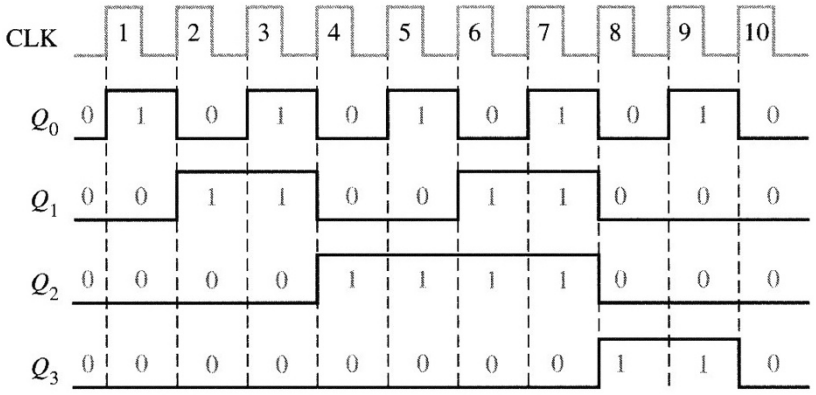


Tabela de transições do FF-JK:

Q(t)	Q(t+1)	J(t)	K(t)
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Projetos:

- Exemplo: contador síncrono decimal:

2. Mapas de Karnaugh → Circuito final

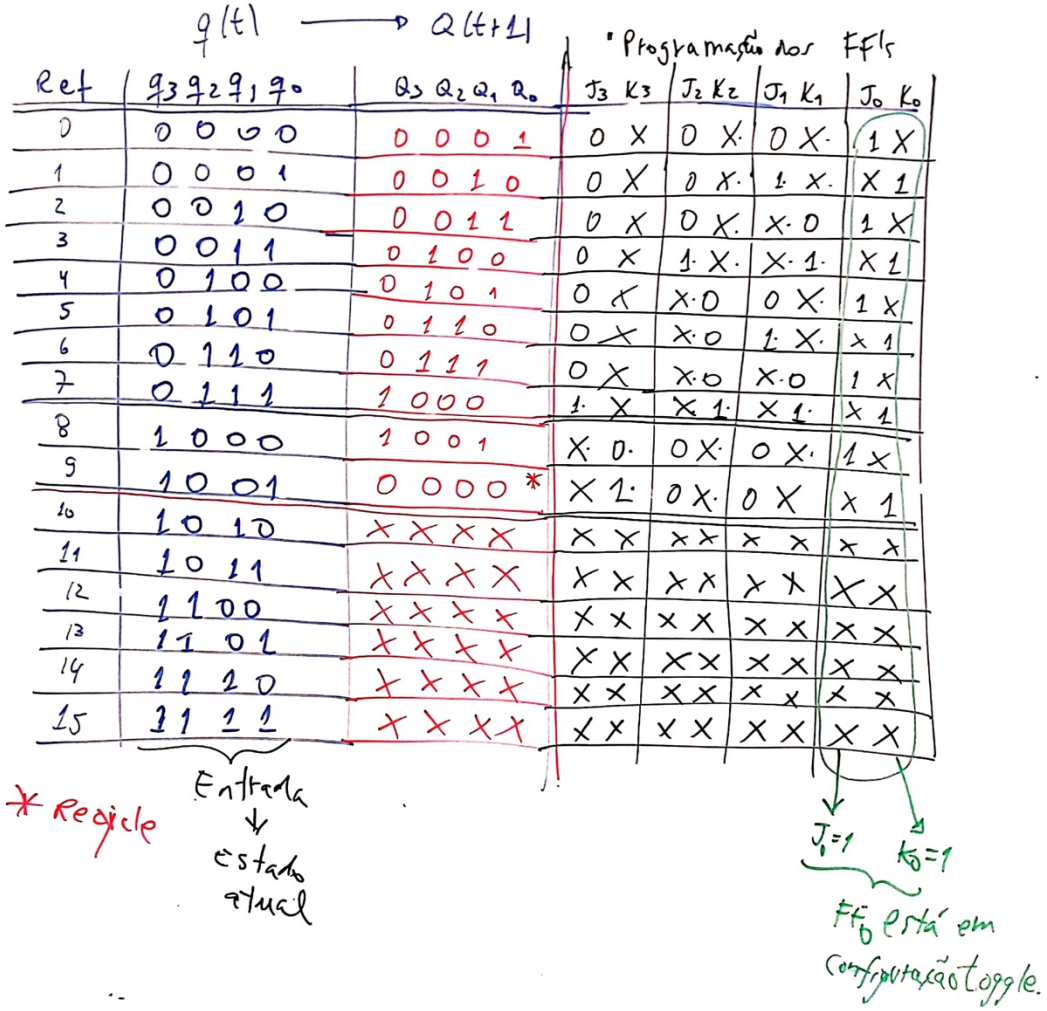


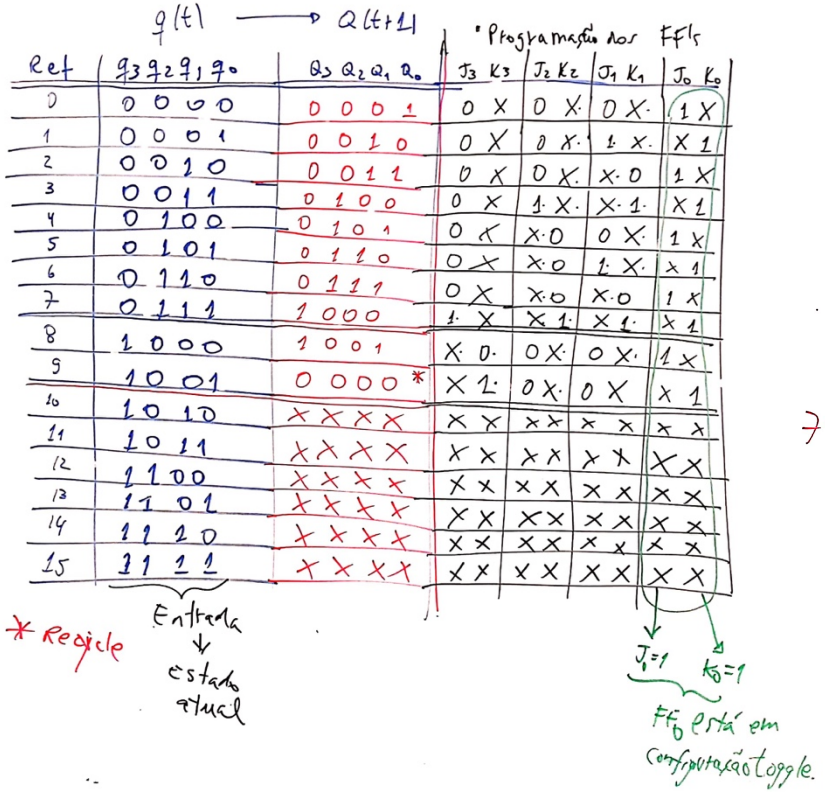
Tabela de transições do FF-JK:

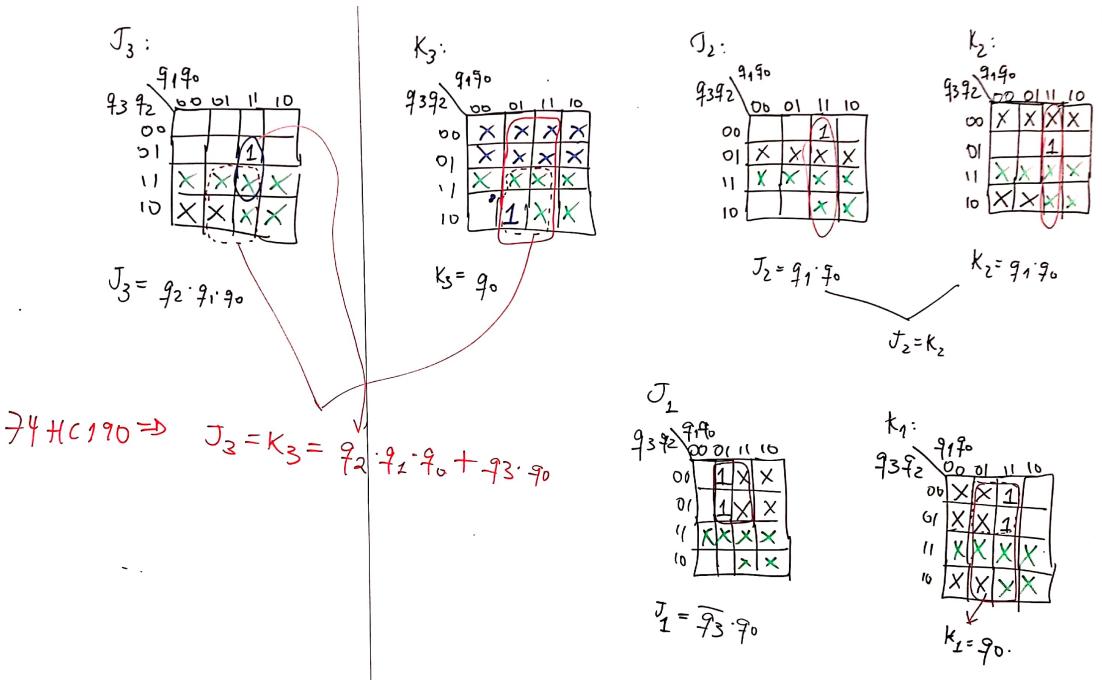
Q(t)	Q(t+1)	J(t)	K(t)
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Projetos:

- Exemplo: contador síncrono decimal:

1. Tabela de transição de contagem (circuito completo)

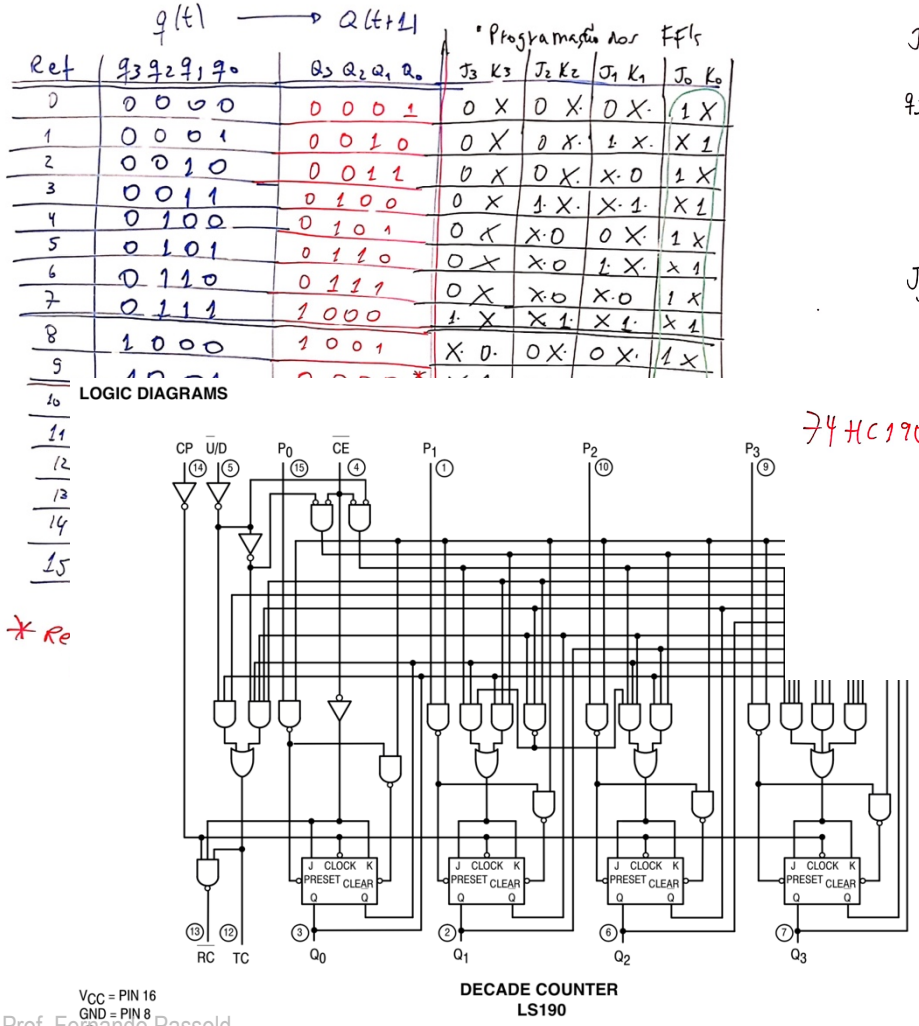




Projetos:

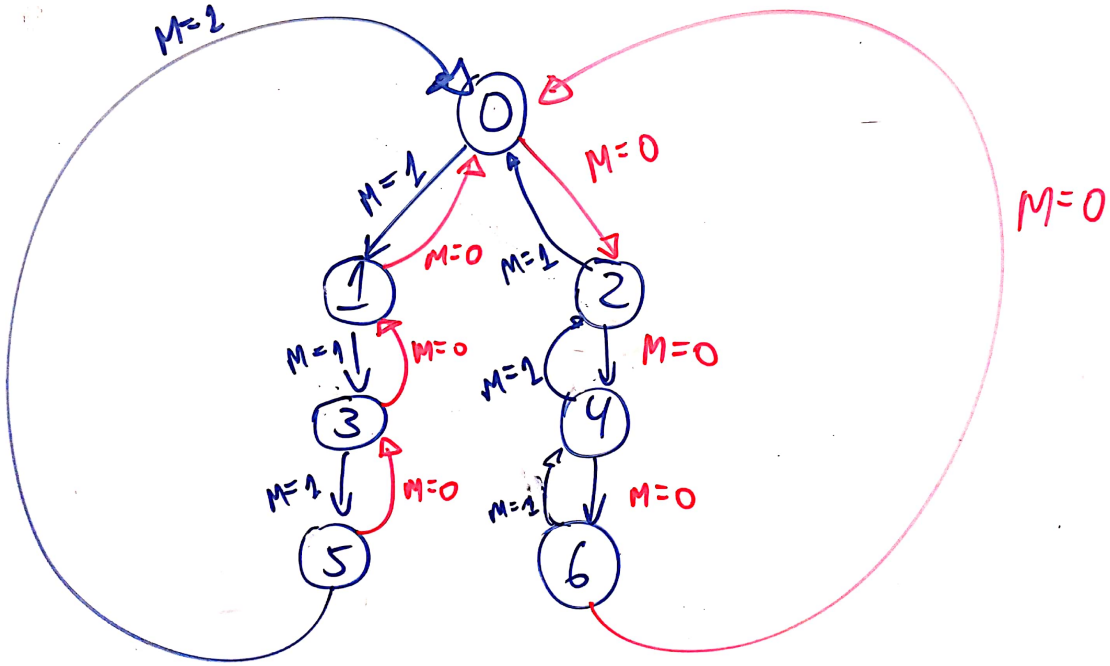
- Exemplo: contador síncrono decimal:

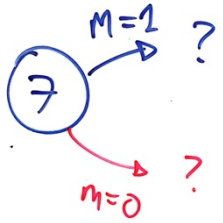
1. Tabela de transição de contagem (circuito completo)



Outros projetos:

- Exemplo$_2$:





opções → FF's-JK ⇒ 3 x FF's ⇒ 3 x 2 entradas (J, K) ⇒ 6 Mapas K

↘ FF's-D ⇒ 3 x FF's ⇒ 3 x 1 entrada (D) ⇒ 3 Mapas K ←

Outros projetos:

- Exemplo$_2$:

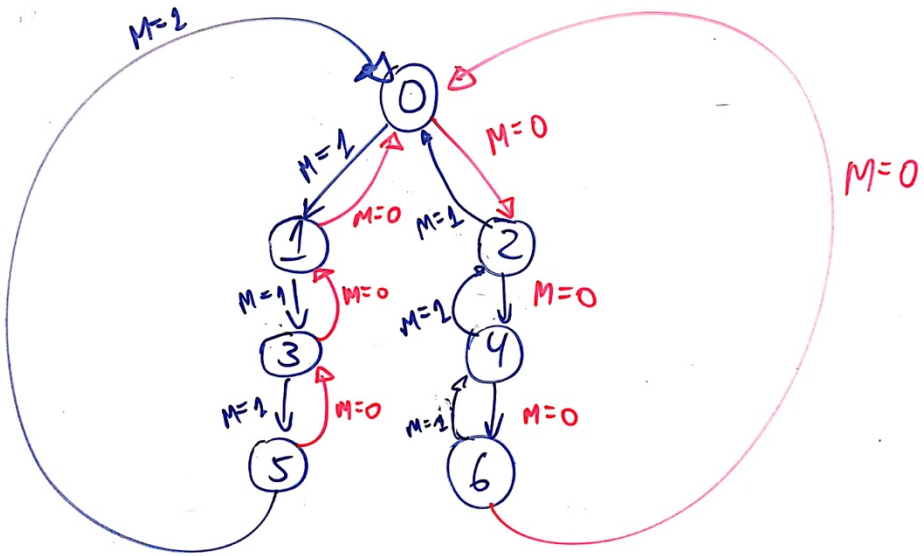


Tabela de transição do circuito completo:

Ref	M	q_2 q_1 q_0	Q_2 Q_1 Q_0	D_2 D_1 D_0
0	0	0 0 0	0 1 0	0 1 0
1	0	0 0 1	0 0 0	0 0 0
2	0	0 1 0	1 0 0	1 0 0
3	0	0 1 1	0 0 1	0 0 1
4	0	1 0 0	1 1 0	1 1 0
5	0	1 0 1	0 1 1	0 1 1
6	0	1 1 0	0 0 0	0 0 0
7	0	1 1 1	X X X	X X X
8	1	0 0 0	0 0 1	0 0 1
9	1	0 0 1	0 1 1	0 1 1
10	1	0 1 0	0 0 0	0 0 0
11	1	0 1 1	1 0 1	1 0 1
12	1	1 0 0	0 1 0	0 1 0
13	1	1 0 1	0 0 0	0 0 0
14	1	1 1 0	1 0 0	1 0 0
15	1	1 1 1	X X X	X X X

opções → FF's-JK ⇒ 3 x FF's ⇒ 3 x 2 entradas (J, K) ⇒ 6 Mapas K

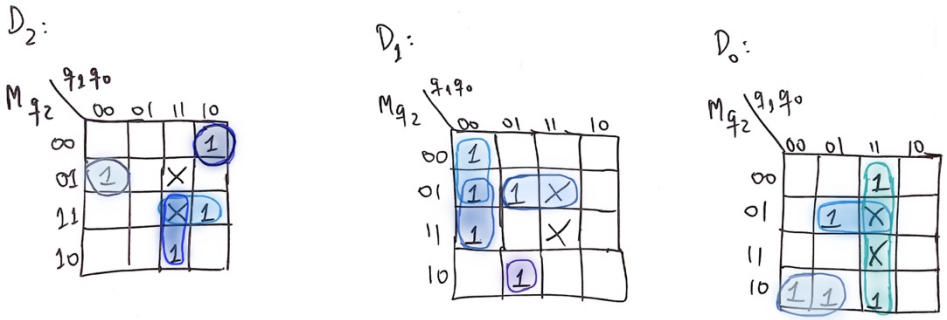
↘ FF's-D ⇒ 3 x FF's ⇒ 3 x 1 entrada (D) ⇒ 3 Mapas K ←

Outros projetos:

- Exemplo$_2$:

Tabela de transição do circuito completo:

Ref	M	q_2 q_1 q_0 (t)	Q_2 Q_1 Q_0 (t+1)	D_2 D_1 D_0
0	0	0 0 0	0 1 0	0 1 0
1	0	0 0 1	0 0 0	0 0 0
2	0	0 1 0	1 0 0	1 0 0
3	0	0 1 1	0 0 1	0 0 1
4	0	1 0 0	1 1 0	1 1 0
5	0	1 0 1	0 1 1	0 1 1
6	0	1 1 0	0 0 0	0 0 0
7	0	1 1 1	X X X	X X X
8	1	0 0 0	0 0 1	0 0 1
9	1	0 0 1	0 1 1	0 1 1
10	1	0 1 0	0 0 0	0 0 0
11	1	0 1 1	1 0 1	1 0 1
12	1	1 0 0	0 1 0	0 1 0
13	1	1 0 1	0 0 0	0 0 0
14	1	1 1 0	1 0 0	1 0 0
15	1	1 1 1	X X X	X X X



$$D_2 = \overline{M}\cdot q_2\cdot\overline{q_1}\cdot\overline{q_0} + \overline{M}\cdot\overline{q_2}\cdot q_2\cdot\overline{q_0} + M\cdot q_2\cdot q_1 + M\cdot q_1\cdot q_0$$
$$= \overline{M}\cdot\overline{q_0}\,(q_2\cdot\overline{q_1} + \overline{q_2}\cdot q_1) + M\cdot q_1\,(q_2 + q_0)$$
$$= \overline{M}\cdot\overline{q_0}\,(q_2 \odot q_1) + M\cdot q_1\cdot(q_2 + q_0)$$

$$D_1 = \overline{M}\cdot\overline{q_1}\cdot\overline{q_0} + q_2\cdot\overline{q_1}\cdot\overline{q_0} + \overline{M}\cdot q_2\cdot q_0 + M\cdot\overline{q_2}\cdot\overline{q_1}\cdot q_0$$
$$= \overline{q_1}\cdot\overline{q_0}\left(\overline{M} + q_2\right) + q_0\left(\overline{M}\cdot q_2 + M\cdot\overline{q_2}\cdot\overline{q_1}\right)$$

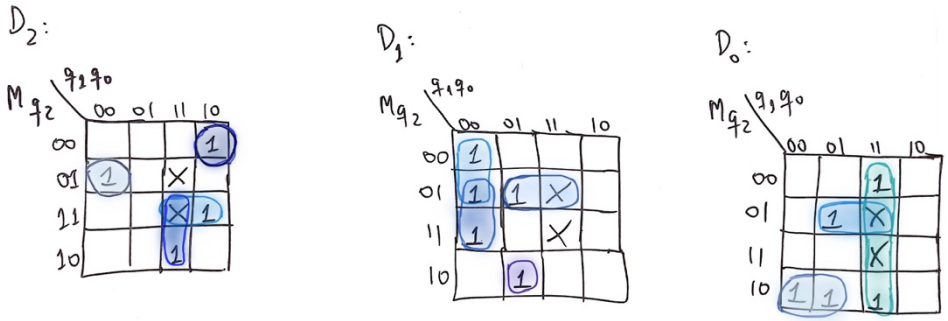
$$D_0 = q_1\cdot q_0 + \overline{M}\cdot q_2\cdot q_0 + M\cdot\overline{q_2}\cdot\overline{q_1}$$

Outros projetos:

- Exemplo$_2$:

Tabela de transição do circuito completo:

Ref	M	q_2 q_1 q_0 (t)	Q_2 Q_1 Q_0 (t+1)	D_2 D_1 D_0
0	0	0 0 0	0 1 0	0 1 0
1	0	0 0 1	0 0 0	0 0 0
2	0	0 1 0	1 0 0	1 0 0
3	0	0 1 1	0 0 1	0 0 1
4	0	1 0 0	1 1 0	1 1 0
5	0	1 0 1	0 1 1	0 1 1
6	0	1 1 0	0 0 0	0 0 0
7	0	1 1 1	X X X	X X X
8	1	0 0 0	0 0 1	0 0 1
9	1	0 0 1	0 1 1	0 1 1
10	1	0 1 0	0 0 0	0 0 0
11	1	0 1 1	1 0 1	1 0 1
12	1	1 0 0	0 1 0	0 1 0
13	1	1 0 1	0 0 0	0 0 0
14	1	1 1 0	1 0 0	1 0 0
15	1	1 1 1	X X X	X X X



$$
\begin{aligned}
D_2 &= \overline{M}\cdot q_2\cdot\overline{q_1}\cdot\overline{q_0}+\overline{M}\cdot\overline{q_2}\cdot q_2\cdot\overline{q_0}+ \\
&\quad +M\cdot q_2\cdot q_1+M\cdot q_1\cdot q_0 \\
&= \overline{M}\cdot\overline{q_0}\,(q_2\cdot\overline{q_1}+\overline{q_2}\cdot q_1)+M\cdot q_1\,(q_2+q_0) \\
&= \overline{M}\cdot\overline{q_0}\,(q_2\odot q_1)+M\cdot q_1\cdot(q_2+q_0)
\end{aligned}
$$

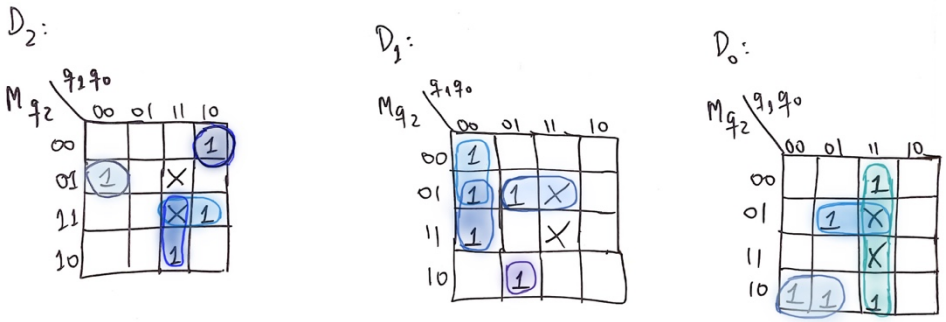
$$
\begin{aligned}
D_1 &= \overline{M}\cdot\overline{q_1}\cdot\overline{q_0}+q_2\cdot\overline{q_1}\cdot\overline{q_0}+\overline{M}\cdot q_2\cdot q_0+ \\
&\quad +M\cdot\overline{q_2}\cdot\overline{q_1}\cdot q_0 \\
&= \overline{q_1}\cdot\overline{q_0}\left(\overline{M}+q_2\right)+q_0\left(\overline{M}\cdot q_2+M\cdot\overline{q_2}\cdot\overline{q_1}\right)
\end{aligned}
$$

$$
D_0 = q_1\cdot q_0+\overline{M}\cdot q_2\cdot q_0+M\cdot\overline{q_2}\cdot\overline{q_1}
$$

Outra solução + simples p/circuito?

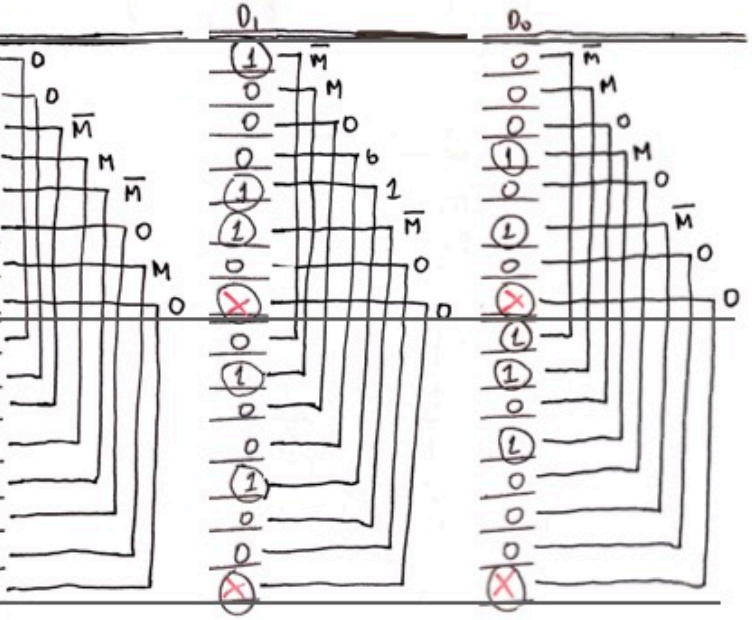
Outros projetos:

- Exemplo$_2$:



Outra solução + simples p/ circuito? Uso de MUXes

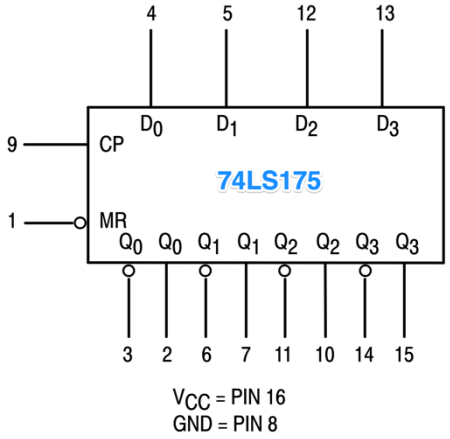
Ref	M	q_2	q_1	q_0	Q_2	Q_1	Q_0	D_2
0	0	0	0	0	0	1	0	0
1	0	0	0	1	0	0	0	0
2	0	0	1	0	1	0	0	1
3	0	0	1	1	0	0	1	0
4	0	1	0	0	1	1	0	1
5	0	1	0	1	0	1	1	0
6	0	1	1	0	0	0	0	0
7	0	1	1	1	X	X	X	X
8	1	0	0	0	0	0	1	0
9	1	0	0	1	0	1	1	0
10	1	0	1	0	0	0	0	0
11	1	0	1	1	1	0	1	1
12	1	1	0	0	0	1	0	0
13	1	1	0	1	0	0	0	0
14	1	1	1	0	1	0	0	1
15	1	1	1	1	X	X	X	X

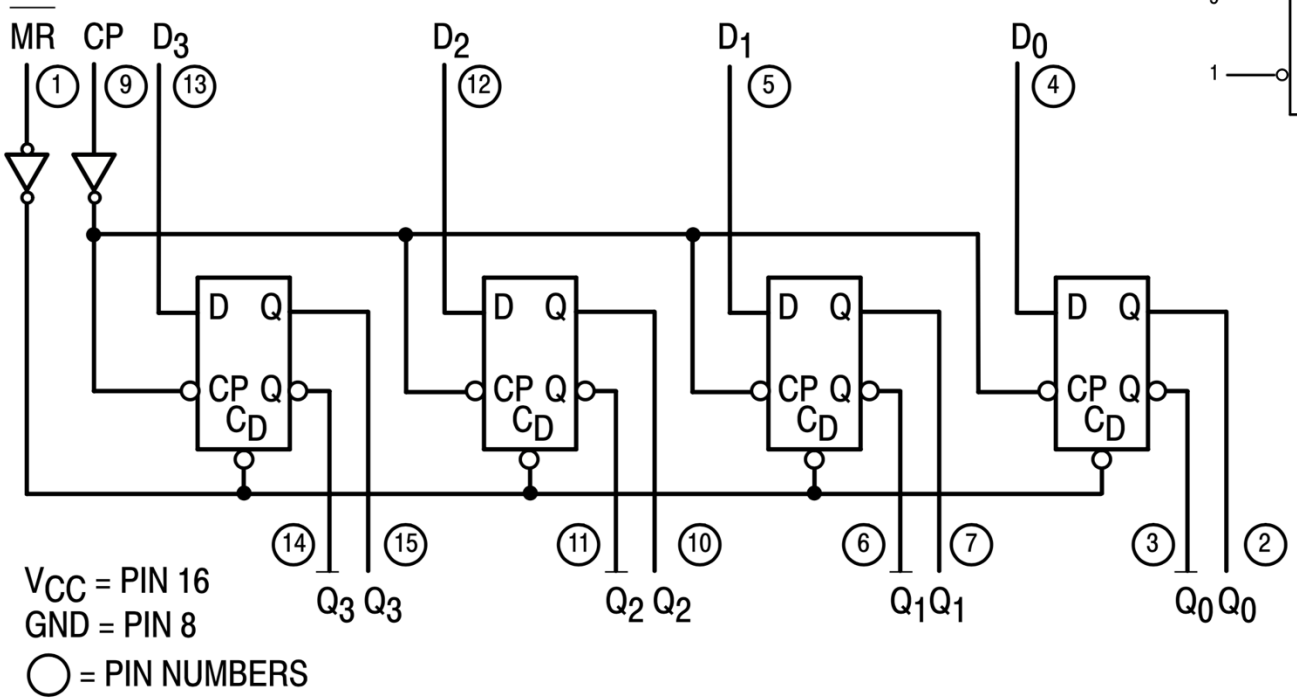


Outros projetos:

- Exemplo$_2$:

Outra solução + simples p/ circuito? Uso de MUXes e CI com 4 FF's-D: 74LS175

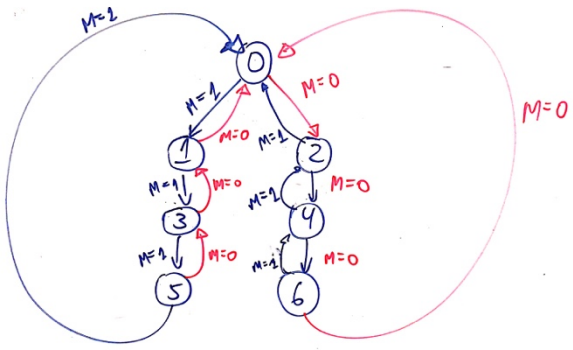


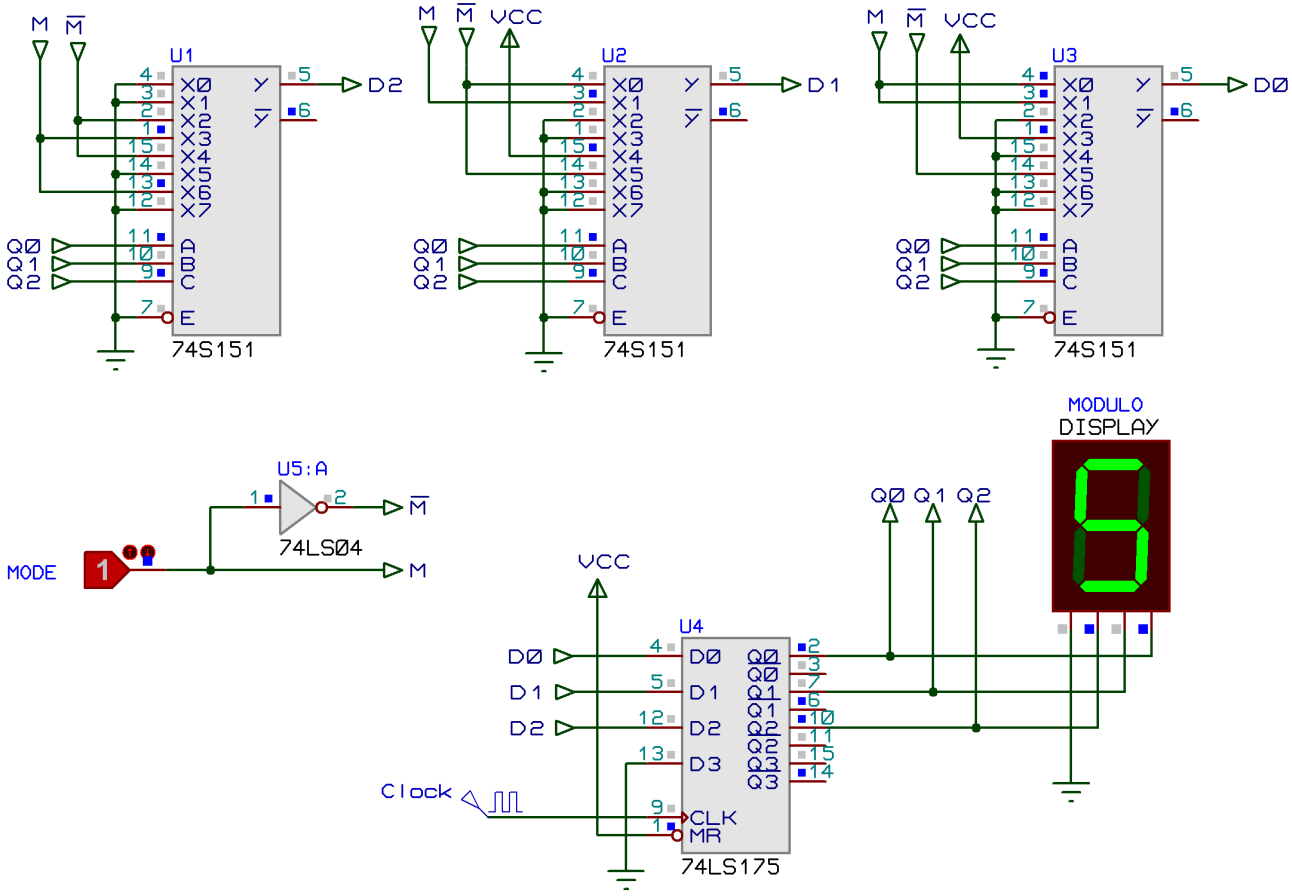


Outros projetos:

- Exemplo$_2$:

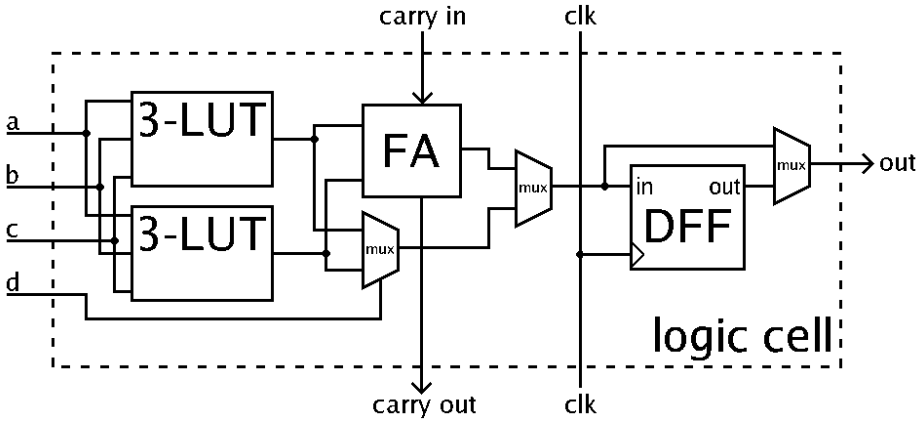
Solução + simples p/circuito: Uso de MUXes e CI com 4 FF's-D: 74LS175





Detalhe última solução ➠ FPGAs

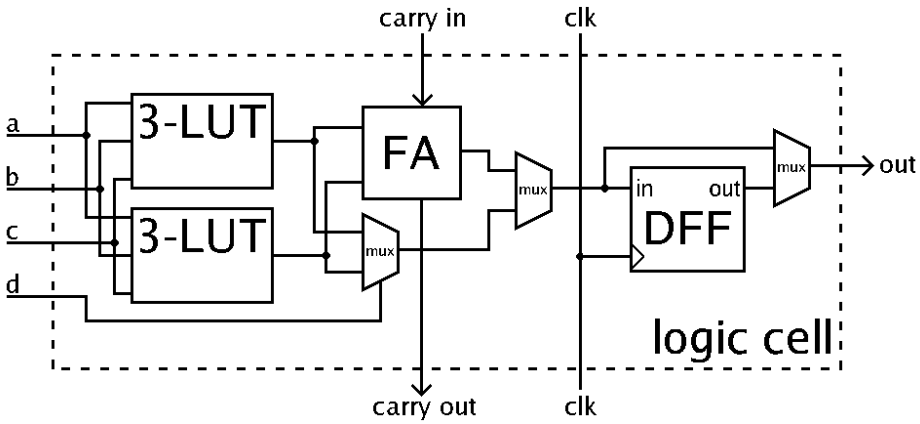
- Estrutura interna (genérica/simplificada) de um FPGA:



- Este tipo de pastilha permite programar o acionamento de Flipf-Flop's através de "lookup tables". A logica combinacional definida para acinar a(s) entrada(s) de controle do FF fica programada dentro de uma "lookup table", ou "LUT".
- Note a presença de um FF tipo D, um somador completo (FA = Full Adder) e 2 conjuntos de LUT para até 3 variáveis de entrada.
- Note que é comum que o circuito lógico correspondente a cada LUT seja formado usando um Multiplexador, no caso da figura mostrada, seria um MUX para até 8 linhas de entrada, os pontos a, b e c correspondem aos pinos de Select do MUX.

Detalhe última solução ➠ FPGAs

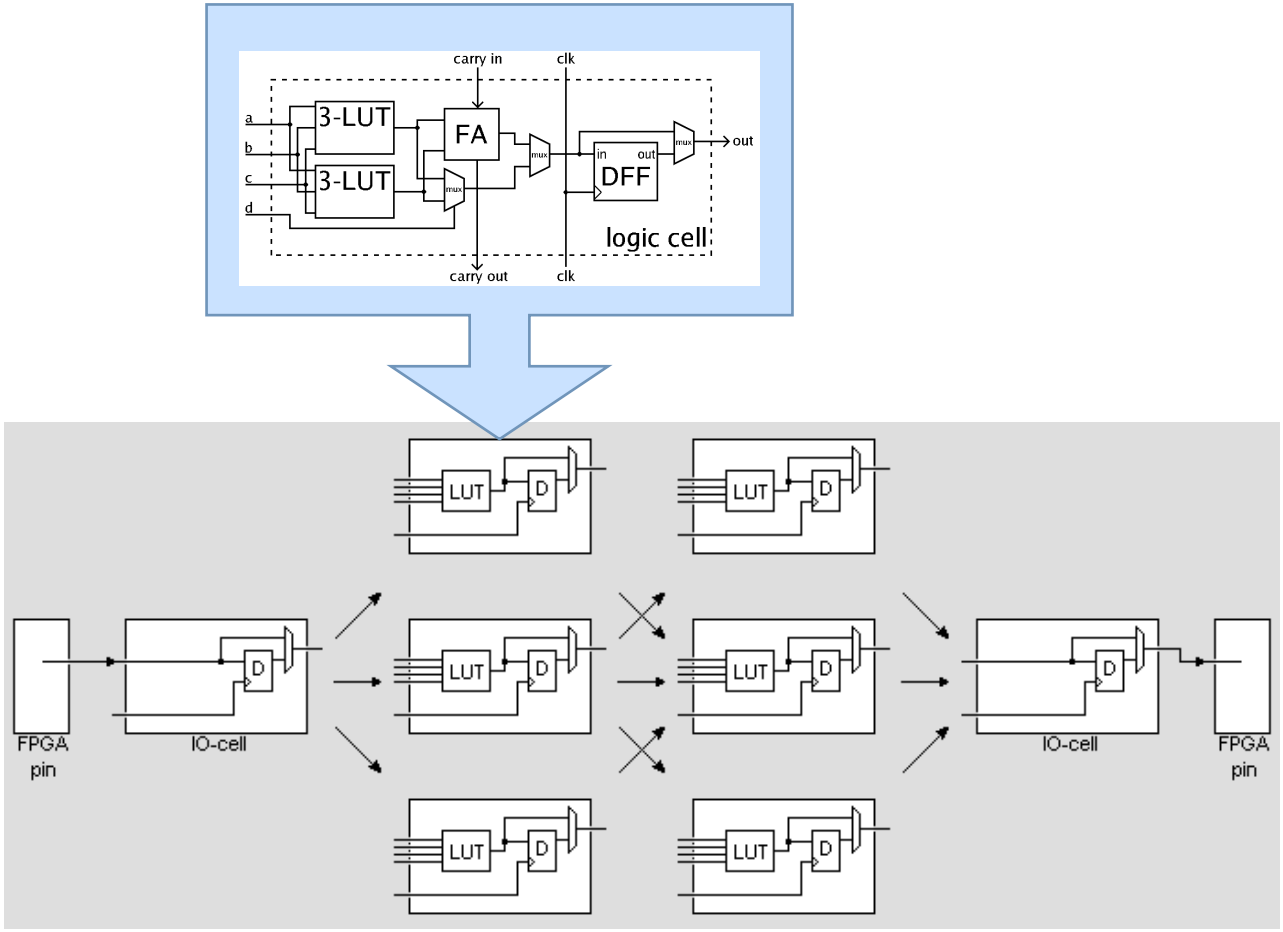
- Estrutura interna (genérica/simplificada) de um FPGA:



- Alguns FPGA's não incluem na sua célula básica um soma- dor completo.
- Neste caso, a saída do MUX é normalmente conectada à entrada de uma porta AND de 2 entradas, onde a segunda entrada da porta AND ficaria conectada ao ponto d (da figura). Note que este ponto d permite que o mesmo atue como um sinal de Enable.
- Uma pastilha FPGA é formada por centenas ou milhares destas células lógicas básicas. Cada célula lógica pode ser conectada a outras células através de recursos de interconexão ("fusíveis") programados por software. Cada célula isoladamente pode fazer pouco, mas com muitas delas conectados juntas, funções lógicas complexas podem ser criadas através de celulas de Entrada/ Saída (IO-cells).

Detalhe última solução ➠ FPGAs

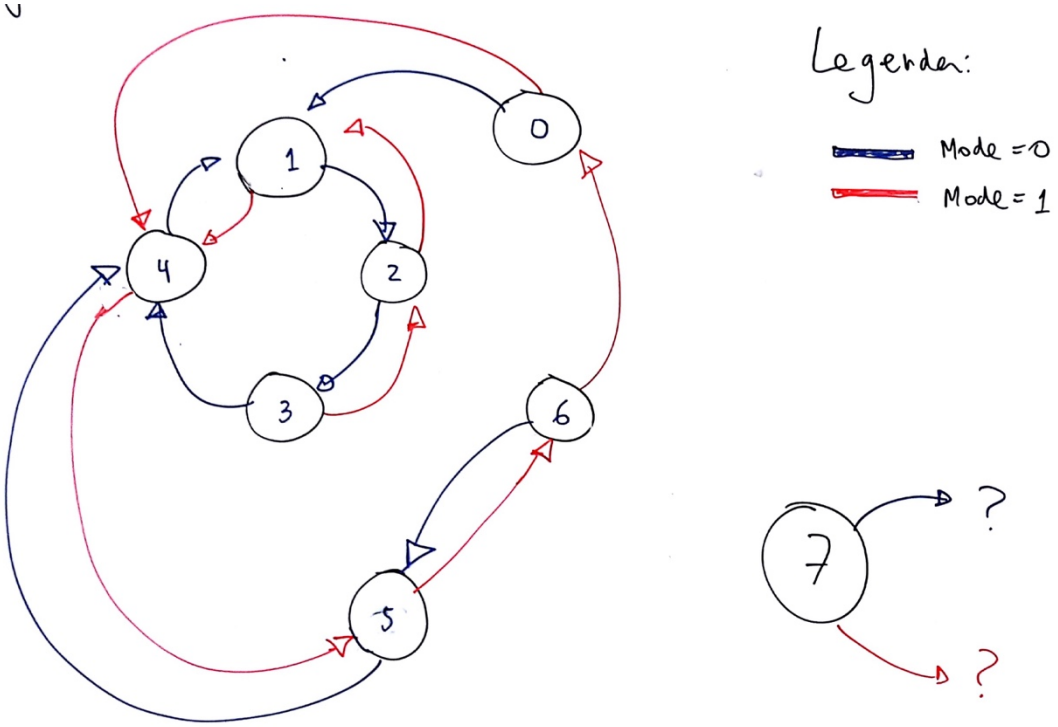
- Estrutura interna (genérica/simplificada) de um FPGA:



Uma pastilha FPGA é formada por centenas ou milhares destas células lógicas básicas. Cada célula lógica pode ser co- nectada a outras células através de recursos de interconexão ("fusíveis") programados por software. Cada célula isoladamente pode fazer pouco, mas com muitas delas conectados juntas, funções lógicas complexas podem ser criadas através de células de Entrada/ Saída (IO-cells).

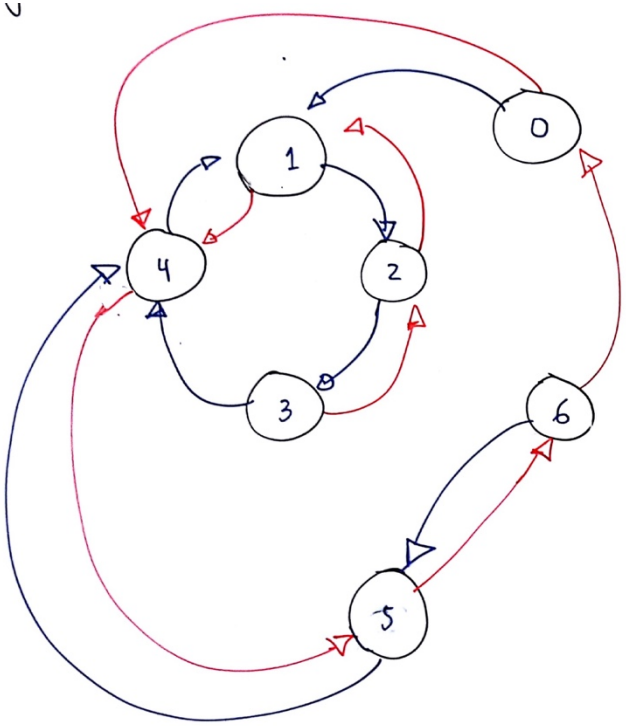
Outros projetos:

- Exemplo$_3$:



Outros projetos:

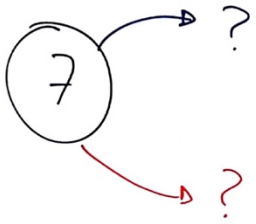
- Exemplo$_3$:



Legenda:

Mode = 0

Mode = 1



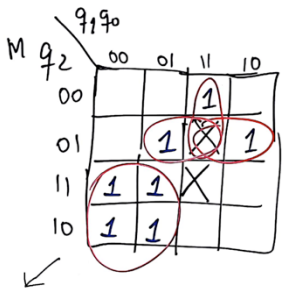
Ref	(M) Mode	q(t) → Q(t+1) $q_2\ q_1\ q_0$	$Q_2\ Q_1\ Q_0$	$D_2\ D_1\ D_0$
0	0	0 0 0	0 0 1	0 0 1
1	0	0 0 1	0 1 0	0 1 0
2	0	0 1 0	0 1 1	0 1 1
3	0	0 1 1	1 0 0	1 0 0
4	0	1 0 0	0 0 1	0 0 1
5	0	1 0 1	1 0 0	1 0 0
6	0	1 1 0	1 0 1	1 0 1
7	0	1 1 1	X X X	X X X
8	1	0 0 0	1 0 0	1 0 0
9	1	0 0 1	1 0 0	1 0 0
10	1	0 1 0	0 0 1	0 0 1
11	1	0 1 1	0 1 0	0 1 0
12	1	1 0 0	1 0 1	1 0 1
13	1	1 0 1	1 1 0	1 1 0
14	1	1 1 0	0 0 0	0 0 0
15	1	1 1 1	X X X	X X X

Outros projetos:

- Exemplo$_3$:

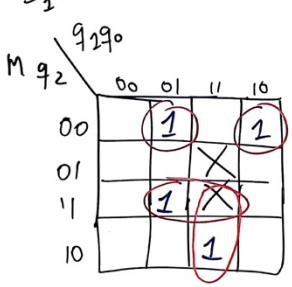
Ref	(M) Mode	$q_2\ q_1\ q_0$	$Q_2\ Q_1\ Q_0$	$D_2\ D_1\ D_0$
0	0	0 0 0	0 0 1	0 0 1
1	0	0 0 1	0 1 0	0 1 0
2	0	0 1 0	0 1 1	0 1 1
3	0	0 1 1	1 0 0	1 0 0
4	0	1 0 0	0 0 1	0 0 1
5	0	1 0 1	1 0 0	1 0 0
6	0	1 1 0	1 0 1	1 0 1
7	0	1 1 1	X X X	X X X
8	1	0 0 0	1 0 0	1 0 0
9	1	0 0 1	1 0 0	1 0 0
10	1	0 1 0	0 0 1	0 0 1
11	1	0 1 1	0 1 0	0 1 0
12	1	1 0 0	1 0 1	1 0 1
13	1	1 0 1	1 1 0	1 1 0
14	1	1 1 0	0 0 0	0 0 0
15	1	1 1 1	X X X	X X X

D_2:

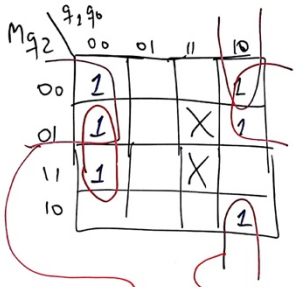


$$D_2 = M \cdot \overline{q_1} + \overline{M} \cdot q_2 \cdot q_0 + \overline{M} \cdot q_1 \cdot q_0 + \overline{M} \cdot q_2 \cdot q_1$$

D_1:



D_0:

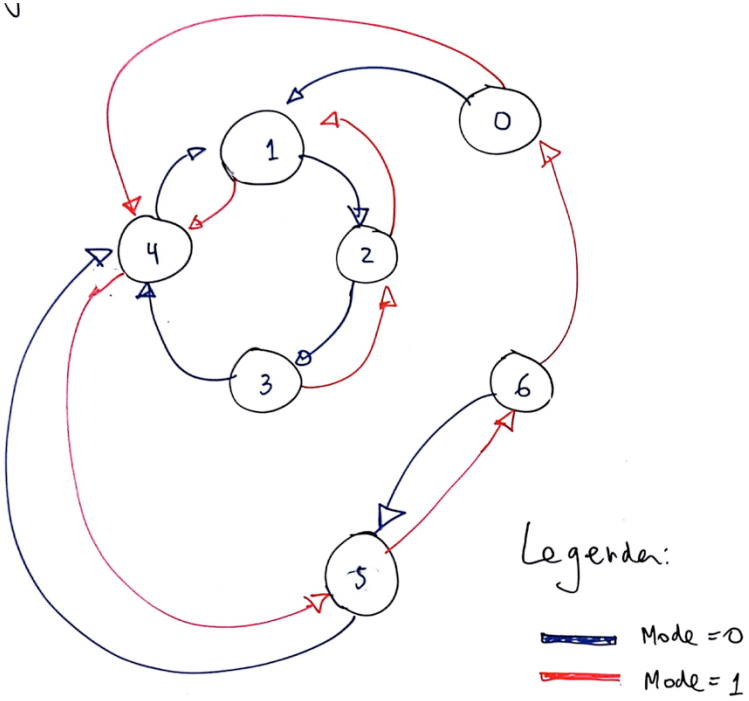


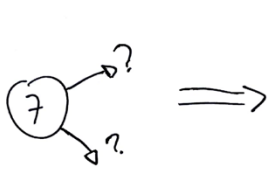
$$D_0 = \overline{M} \cdot \overline{q_0} + \overline{q_2} \cdot q_1 \cdot \overline{q_0} + q_2 \cdot \overline{q_1} \cdot \overline{q_0}$$

$$D_1 = \underbrace{\overline{M} \cdot \overline{q_2} \cdot \overline{q_1} \cdot q_0 + \overline{M} \cdot \overline{q_2} \cdot q_1 \cdot \overline{q_0}}_{\overline{M} \cdot \overline{q_2}\,(q_1 \oplus q_0)} + M \cdot q_2 \cdot q_0 + M \cdot q_1 \cdot q_0$$

Outros projetos:

- Exemplo$_3$:





M	$q_2 q_1 q_0$		$Q_2 Q_1 Q_0$	$Q_2 Q_1 Q_0$
0	1 1 1	→	?	1 0 0
1	1 1 1	→	?	0 1 0

$$D_2 = M \cdot \bar{q}_1 + \bar{M} \cdot q_2 \cdot q_0 + \bar{M} \cdot q_1 \cdot q_0 + \bar{M} \cdot q_2 \cdot q_1$$

Mode=0 → $D_2 = 0 \cdot \bar{q}_1 + 1 \cdot 1 \cdot 1 + \ldots = 1$

Mode=1 → $D_2 = 1 \cdot 0 + 0 \cdot q_2 \cdot q_0 + 0 \cdot q_1 \cdot q_0 + 0 \cdot q_2 \cdot q_1 = 0$

$$D_1 = \bar{M} \cdot \bar{q}_2 \cdot \bar{q}_1 \cdot q_0 + \bar{M} \cdot \bar{q}_2 \cdot q_1 \cdot \bar{q}_0 + M \cdot q_2 \cdot q_0 + M \cdot q_1 \cdot q_0$$

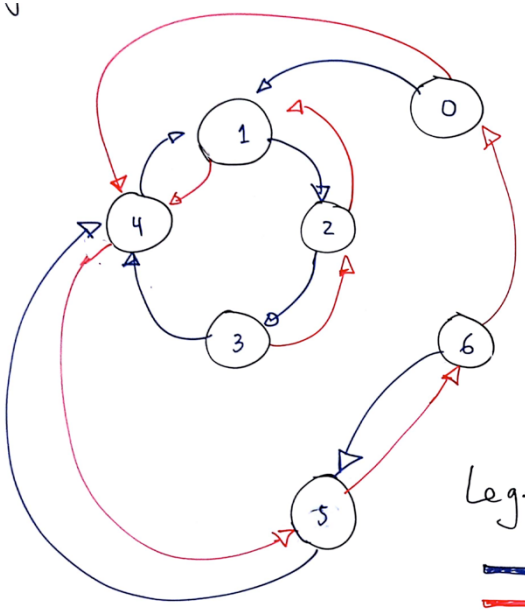
Mode=0 → $D_1 = \underbrace{1 \cdot \bar{1} \cdot \bar{1} \cdot 1}_{0} + \underbrace{1 \cdot \bar{1} \cdot 1 \cdot \bar{1}}_{0} + \underbrace{0 \cdot q_2 \cdot q_0}_{0} + \underbrace{0 \cdot q_1 \cdot q_0}_{0} = 0$

Mode=1 → $D_1 = \underbrace{0 \cdot \bar{q}_2 \cdot \bar{q}_1 \cdot q_0}_{0} + \underbrace{0 \cdot \bar{q}_2 \cdot q_1 \cdot \bar{q}_0}_{0} + \underbrace{1 \cdot 1 \cdot 1}_{1} + \ldots = 1$

$$D_0 = \bar{M} \cdot \bar{q}_0 + \bar{q}_2 \cdot q_1 \cdot \bar{q}_0 + q_2 \cdot q_1 \cdot \bar{q}_0$$

M=0 → $D_0 = 1 \cdot 0 + \underbrace{0 \cdot q_1 \cdot \bar{q}_0}_{0} + \underbrace{q_2 \cdot q_1 \cdot 0}_{0} = 0$

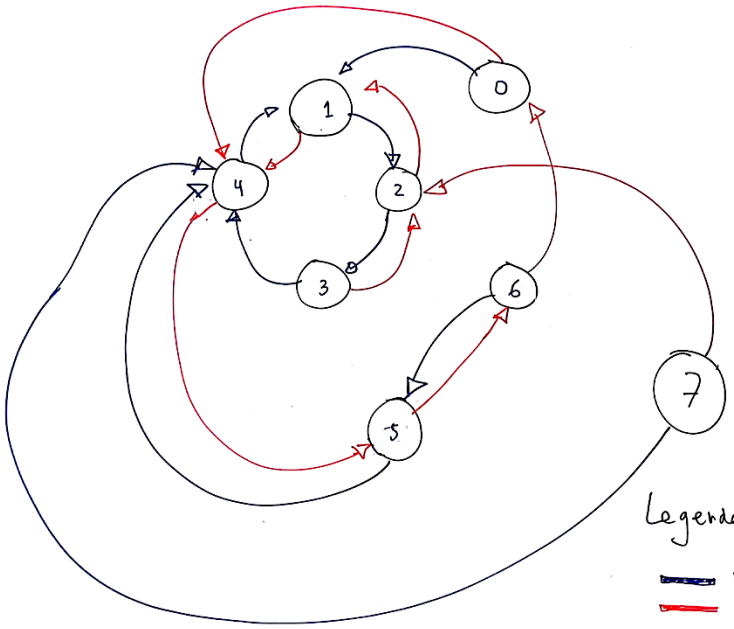
M=1 → $D_0 = 0 \cdot \bar{q}_0 + \underbrace{0 \cdot q_1 \cdot \bar{q}_0}_{0} + \underbrace{q_2 \cdot q_1 \cdot 0}_{0} = 0$



Legenda:

Mode = 0

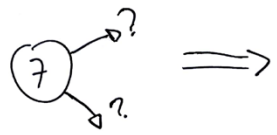
Mode = 1



Legenda:

Mode = 0

Mode = 1



M	$q_2 q_1 q_0$		$Q_2 Q_1 Q_0$	$Q_2 Q_1 Q_0$
0	1 1 1	→	?	1 0 0
1	1 1 1	→	?	0 1 0

7 → Mode 0 → $100_{(2)}$ = $4_{(10)}$

7 → Mode 1 → $010_{(2)}$ = $2_{(10)}$

$$D_2 = M \cdot \bar{q}_1 + \bar{M} \cdot q_2 \cdot q_0 + \bar{M} \cdot q_1 \cdot q_0 + \bar{M} \cdot q_2 \cdot q_1$$

Mode=0 → $D_2 = 0 \cdot \bar{q}_1 + 1 \cdot 1 \cdot 1 + \ldots = 1$

Mode=1 → $D_2 = 1 \cdot 0 + 0 \cdot q_2 \cdot q_0 + 0 \cdot q_1 \cdot q_0 + 0 \cdot q_2 \cdot q_1 = 0$

$$D_1 = \bar{M} \cdot \bar{q}_2 \cdot \bar{q}_1 \cdot q_0 + \bar{M} \cdot \bar{q}_2 \cdot q_1 \cdot \bar{q}_0 + M \cdot q_2 \cdot q_0 + M \cdot q_1 \cdot q_0$$

Mode=0 → $D_1 = \underbrace{1 \cdot \bar{1} \cdot \bar{1} \cdot 1}_{0} + \underbrace{1 \cdot \bar{1} \cdot 1 \cdot \bar{1}}_{0} + \underbrace{0 \cdot q_2 \cdot q_0}_{0} + \underbrace{0 \cdot q_1 \cdot q_0}_{0} = 0$

Mode=1 → $D_1 = \underbrace{0 \cdot \bar{q}_2 \cdot \bar{q}_1 \cdot q_0}_{0} + \underbrace{0 \cdot \bar{q}_2 \cdot q_1 \cdot \bar{q}_0}_{0} + \underbrace{1 \cdot 1 \; 1}_{1} + \ldots = 1$

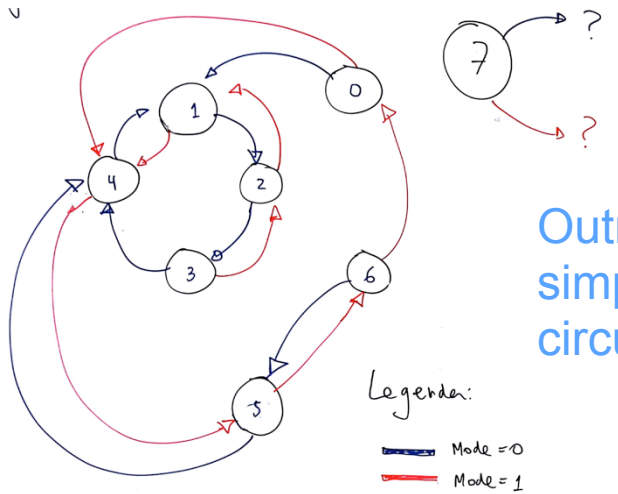
$$D_0 = \bar{M} \cdot \bar{q}_0 + \bar{q}_2 \cdot q_1 \cdot \bar{q}_0 + q_2 \cdot q_1 \cdot \bar{q}_0$$

M = 0 → $D_0 = 1 \cdot 0 + \underbrace{0 \cdot q_1 \cdot \bar{q}_0}_{0} + \underbrace{q_2 \cdot q_1 \cdot 0}_{0} = 0$

M = 1 → $D_0 = 0 \cdot \bar{q}_0 + \underbrace{0 \cdot q_1 \cdot \bar{q}_0}_{0} + \underbrace{q_2 \cdot q_1 \cdot 0}_{0} = 0$

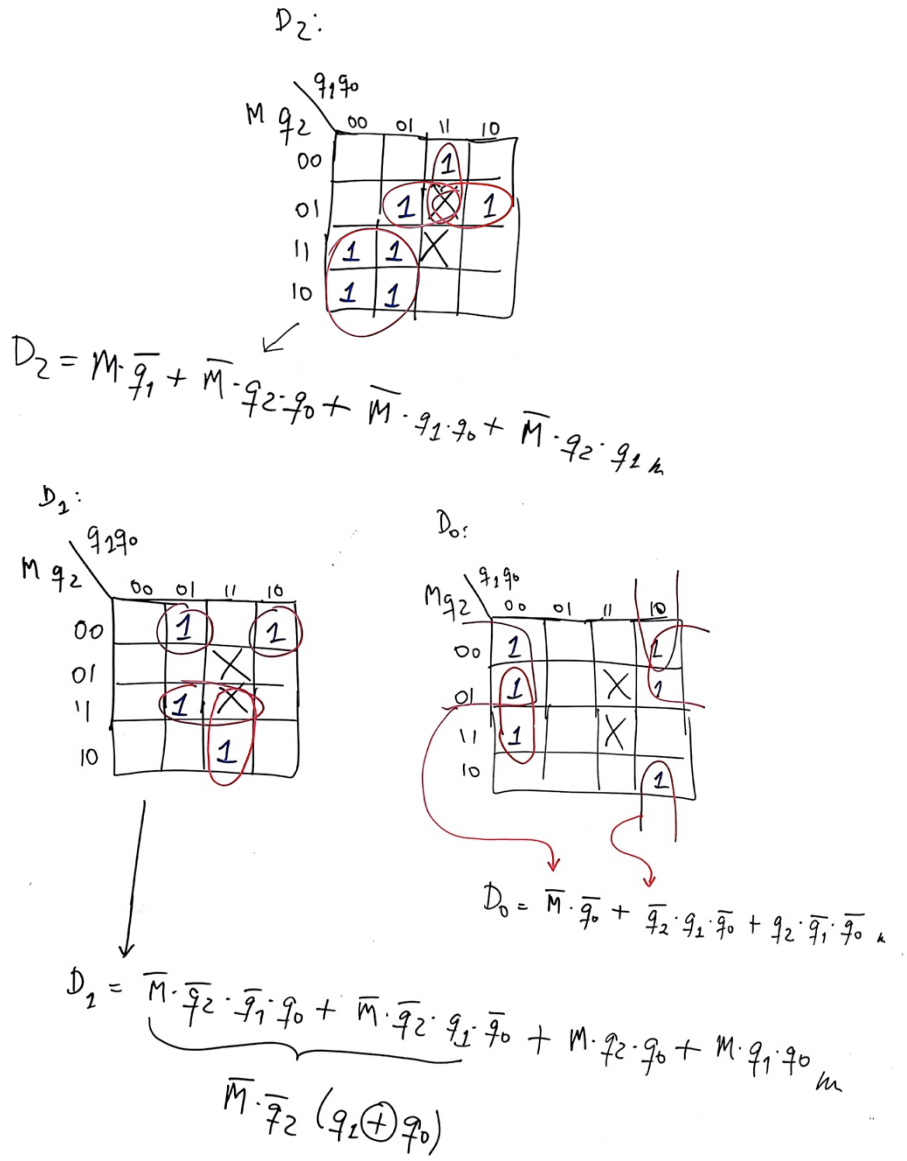
Outros projetos:

- Exemplo$_3$:



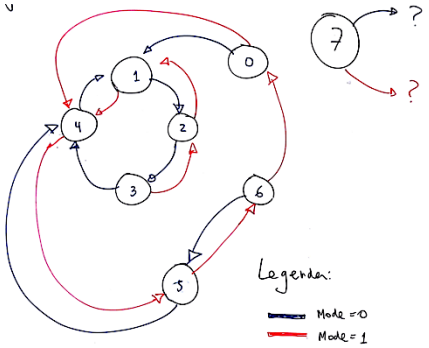
Ref	(M) Mode	$q_2\ q_1\ q_0$	$Q_2\ Q_1\ Q_0$	$D_2\ D_1\ D_0$
0	0	0 0 0	0 0 1	0 0 1
1	0	0 0 1	0 1 0	0 1 0
2	0	0 1 0	0 1 1	0 1 1
3	0	0 1 1	1 0 0	1 0 0
4	0	1 0 0	0 0 1	0 0 1
5	0	1 0 1	1 0 0	1 0 0
6	0	1 1 0	1 0 1	1 0 1
7	0	1 1 1	X X X	X X X
8	1	0 0 0	1 0 0	1 0 0
9	1	0 0 1	1 0 0	1 0 0
10	1	0 1 0	0 0 1	0 0 1
11	1	0 1 1	0 1 0	0 1 0
12	1	1 0 0	1 0 1	1 0 1
13	1	1 0 1	1 1 0	1 1 0
14	1	1 1 0	0 0 0	0 0 0
15	1	1 1 1	XXX	XXX

Outra solução + simples p/ circuito?



Outros projetos:

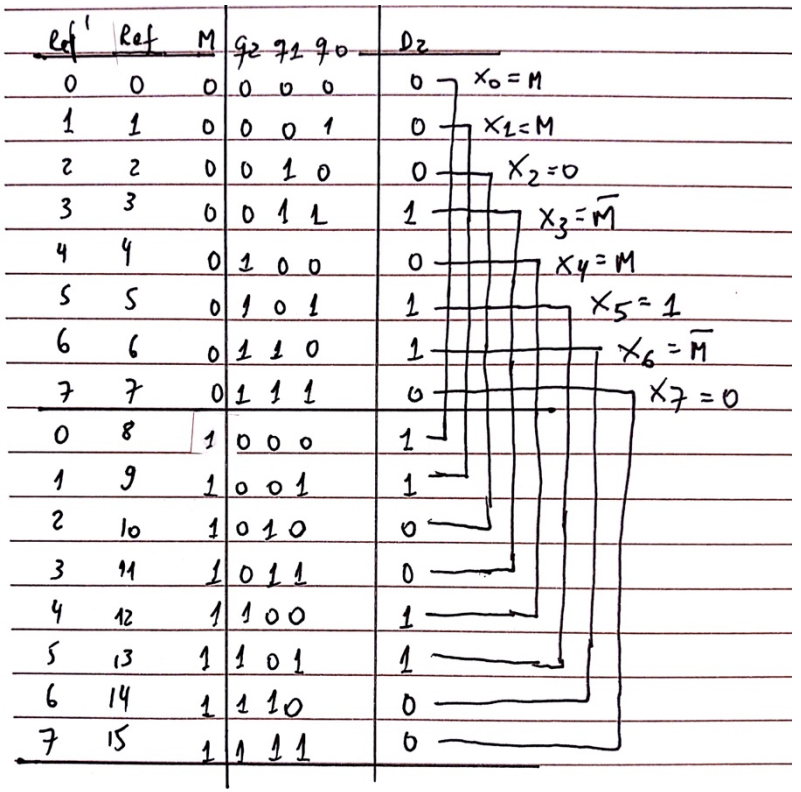
- Exemplo$_3$:



Ref	(M) Mode	$q_2\ q_1\ q_0$	$Q_2\ Q_1\ Q_0$	$D_2\ D_1\ D_0$
0	0	0 0 0	0 0 1	0 0 1
1	0	0 0 1	0 1 0	0 1 0
2	0	0 1 0	0 1 1	0 1 1
3	0	0 1 1	1 0 0	1 0 0
4	0	1 0 0	0 0 1	0 0 1
5	0	1 0 1	1 0 0	1 0 0
6	0	1 1 0	1 0 1	1 0 1
7	0	1 1 1	X X X	X X X
8	1	0 0 0	1 0 0	1 0 0
9	1	0 0 1	1 0 0	1 0 0
10	1	0 1 0	0 0 1	0 0 1
11	1	0 1 1	0 1 0	0 1 0
12	1	1 0 0	1 0 1	1 0 1
13	1	1 0 1	1 1 0	1 1 0
14	1	1 1 0	0 0 0	0 0 0
15	1	1 1 1	X X X	X X X

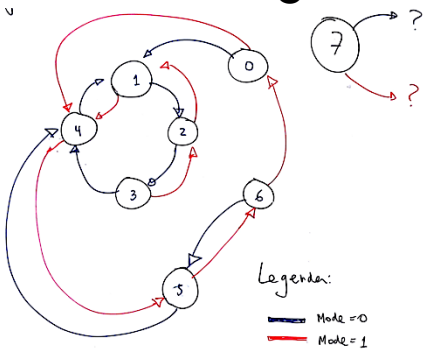
Outra solução + simples: Uso de MUXes:

Ref'	Ref	M	$q_2\ q_1\ q_0$	D_2
0	0	0	0 0 0	0
1	1	0	0 0 1	0
2	2	0	0 1 0	0
3	3	0	0 1 1	1
4	4	0	1 0 0	0
5	5	0	1 0 1	1
6	6	0	1 1 0	1
7	7	0	1 1 1	0
0	8	1	0 0 0	1
1	9	1	0 0 1	1
2	10	1	0 1 0	0
3	11	1	0 1 1	0
4	12	1	1 0 0	1
5	13	1	1 0 1	1
6	14	1	1 1 0	0
7	15	1	1 1 1	0



Outros projetos:

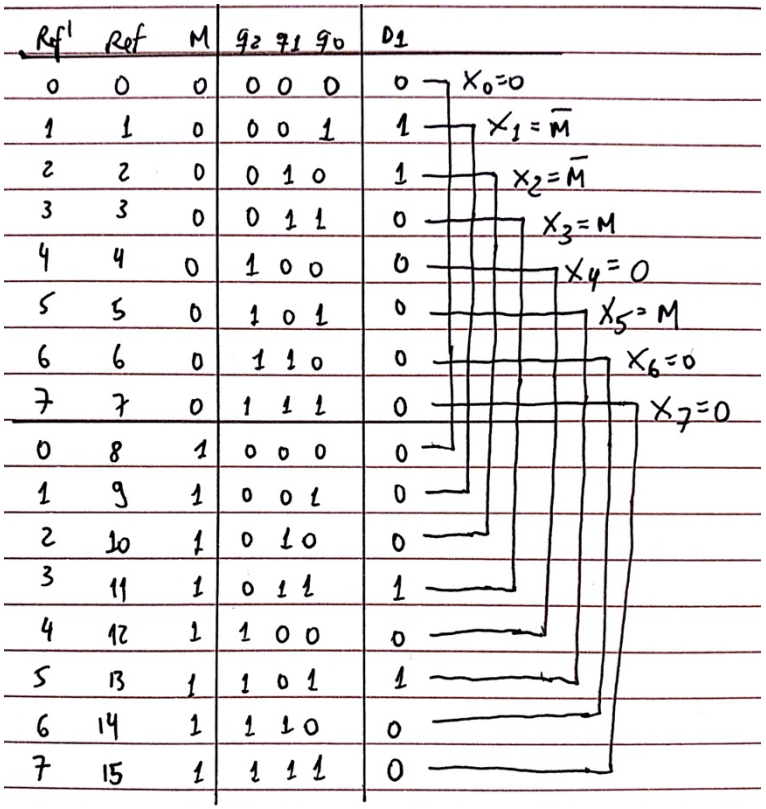
- Exemplo$_3$:



Ref	(M) Mode	$q_2\ q_1\ q_0$	$Q_2\ Q_1\ Q_0$	$D_2\ D_1\ D_0$
0	0	0 0 0	0 0 1	0 0 1
1	0	0 0 1	0 1 0	0 1 0
2	0	0 1 0	0 1 1	0 1 1
3	0	0 1 1	1 0 0	1 0 0
4	0	1 0 0	0 0 1	0 0 1
5	0	1 0 1	1 0 0	1 0 0
6	0	1 1 0	1 0 1	1 0 1
7	0	1 1 1	X X X	X X X
8	1	0 0 0	1 0 0	1 0 0
9	1	0 0 1	1 0 0	1 0 0
10	1	0 1 0	0 0 1	0 0 1
11	1	0 1 1	0 1 0	0 1 0
12	1	1 0 0	1 0 1	1 0 1
13	1	1 0 1	1 1 0	1 1 0
14	1	1 1 0	0 0 0	0 0 0
15	1	1 1 1	X X X	X X X

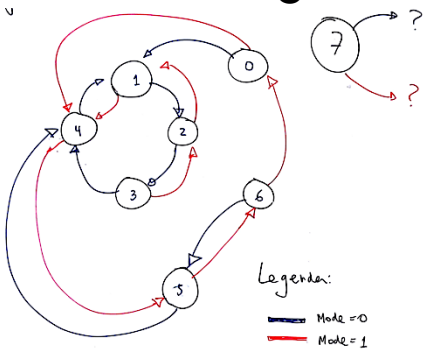
Outra solução + simples: Uso de MUXes:

Ref	Ref	M	$q_2\ q_1\ q_0$	D_1
0	0	0	0 0 0	0
1	1	0	0 0 1	1
2	2	0	0 1 0	1
3	3	0	0 1 1	0
4	4	0	1 0 0	0
5	5	0	1 0 1	0
6	6	0	1 1 0	0
7	7	0	1 1 1	0
0	8	1	0 0 0	0
1	9	1	0 0 1	0
2	10	1	0 1 0	0
3	11	1	0 1 1	1
4	12	1	1 0 0	0
5	13	1	1 0 1	1
6	14	1	1 1 0	0
7	15	1	1 1 1	0



Outros projetos:

- Exemplo$_3$:



Ref	(M) Mode	$q_2\ q_1\ q_0$	$Q_2\ Q_1\ Q_0$	$D_2\ D_1\ D_0$
0	0	0 0 0	0 0 1	0 0 1
1	0	0 0 1	0 1 0	0 1 0
2	0	0 1 0	0 1 1	0 1 1
3	0	0 1 1	1 0 0	1 0 0
4	0	1 0 0	0 0 1	0 0 1
5	0	1 0 1	1 0 0	1 0 0
6	0	1 1 0	1 0 1	1 0 1
7	0	1 1 1	X X X	X X X
8	1	0 0 0	1 0 0	1 0 0
9	1	0 0 1	1 0 0	1 0 0
10	1	0 1 0	0 0 1	0 0 1
11	1	0 1 1	0 1 0	0 1 0
12	1	1 0 0	1 0 1	1 0 1
13	1	1 0 1	1 1 0	1 1 0
14	1	1 1 0	0 0 0	0 0 0
15	1	1 1 1	X X X	X X X

Outra solução + simples: Uso de MUXes:

Ref'	Ref	M	$q_2\ q_1\ q_0$	D_0
0	0	0	0 0 0	1
1	1	0	0 0 1	0
2	2	0	0 1 0	1
3	3	0	0 1 1	0
4	4	0	1 0 0	1
5	5	0	1 0 1	0
6	6	0	1 1 0	1
7	7	0	1 1 1	0
0	8	1	0 0 0	0
1	9	1	0 0 1	0
2	10	1	0 1 0	1
3	11	1	0 1 1	0
4	12	1	1 0 0	1
5	13	1	1 0 1	0
6	14	1	1 1 0	0
7	15	1	1 1 1	0