LINKÖPING UNIVERSITY

Department of Electrical Engineering

TMEL53, DIGITALTEKNIK

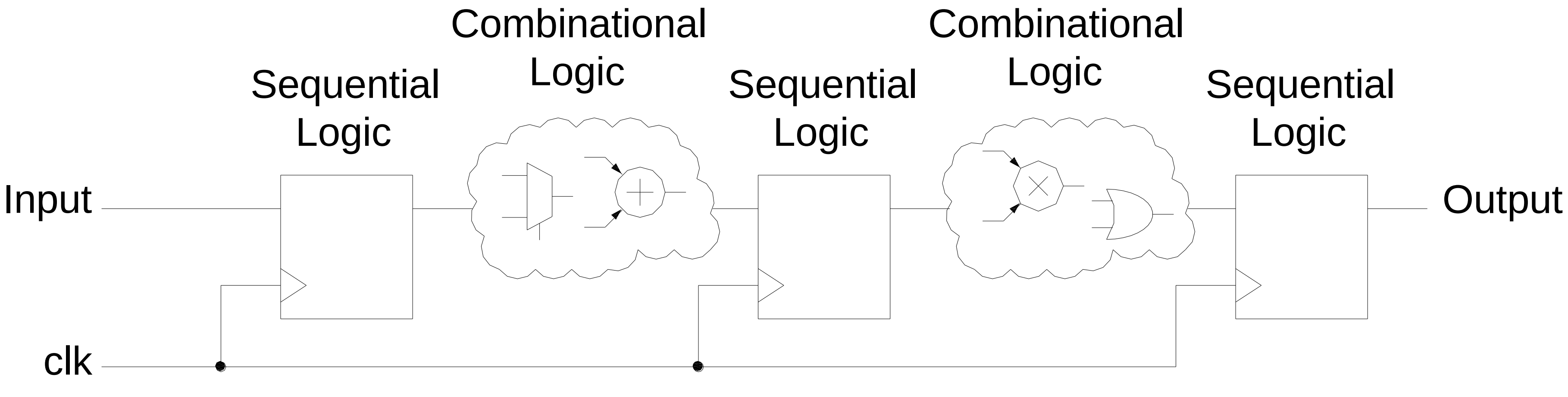
INTRODUCTION TO SYNCHRONOUS CIRCUITS, FLIP-FLOPS and COUNTERS

Mario Garrido Gálvez
mario.garrido.galvez@liu.se

Linköping, 2017

STRUCTURE OF A HW SYSTEM

- **Combinational logic**: Its output only depends on the current input.
- **Sequential logic:** Its output depends on present and previous values. They are memory elements.
- At each clock event, the output of the sequential logic is updated. This makes the combinational logic have new inputs, and the combinational logic makes the calculations for the new inputs.



(DIGITAL) MUSIC

- The **clock** is like a conductor in an ochestra. There is only one and assures that everything is synchronized. With two conductors the orchestra cannot be synchronized. **Only one clock signal in the system!!**

- The conductor defines the tempo of the music. In a digital system, this tempo is given by the clock frequency (or the clock period).

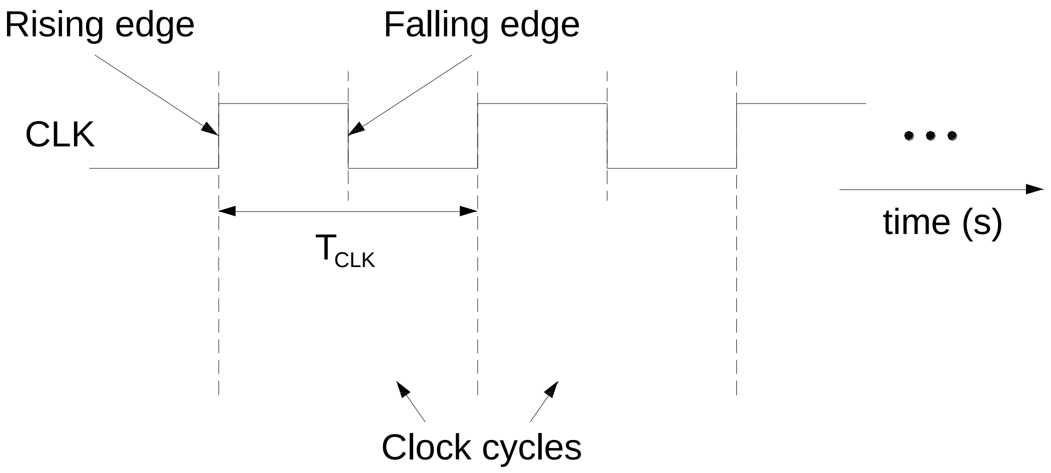
- And the instruments? -> Sequential logic. The clock is only connected to the sequential logic and only connected as a clock, not as a normal input.

CLOCK

- The **clock** of a digital system is a periodical signal that changes alternatively between one and zero. The time in a digital system in measured in **clock cycles.**

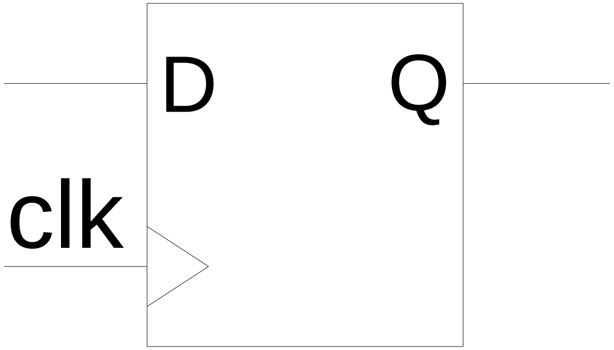
- The **clock period (T_{CLK})** is the time in seconds between two consecutive clock cycles. The **clock frequency (f_{CLK})** is the inverse of the clock period and indicates the number of clock cycles per second, measured in Hz. Higher clock frequency means faster processing.

- All the digital system must be synchronized with the clock and updates every clock cycle. The sequential logic is activated only in the **rising edge** (alternatively the **falling edge**) of the clock.



$$f_{CLK} = \frac{1}{T_{CLK}}$$

D FLIP-FLOP (D-VIPPA) OR REGISTER



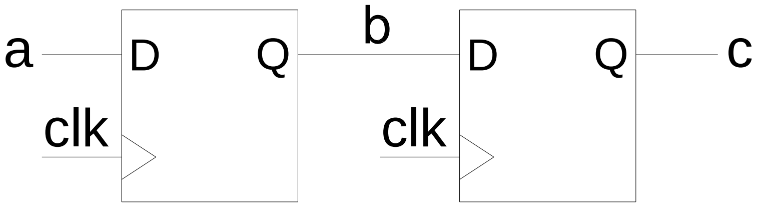
D(n-1)	Q(n-1)	Q(n)
0	0	0
0	1	0
1	0	1
1	1	1

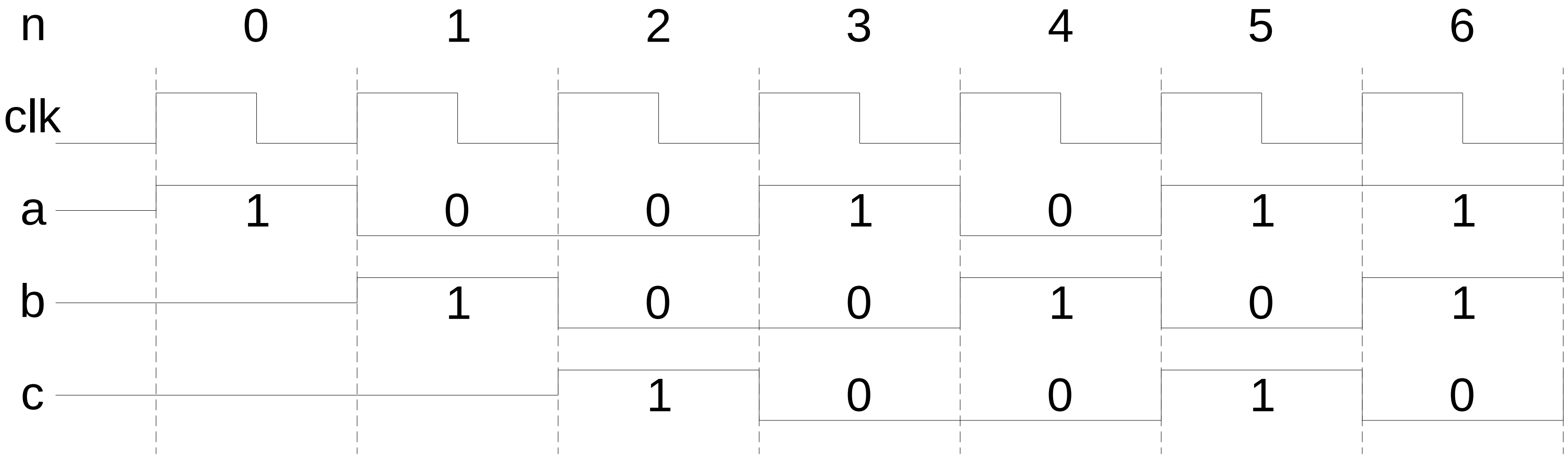
- Behavior: every time that there is a transition in clk (from “0” to “1”), the value at the input is stored in the flip-flop. The value stored in the flip-flop is always provided at the output Q.

- The output at the n-th clock cycle is equal to the input at the (n-1)-th clock cycle, i.e., Q(n) = D(n-1) => the D flip-flop delays data 1 clock cycle. “n” is the number of clock cycle.

- Can be used to store data.

TIMING DIAGRAM (TIDSDIAGRAM)

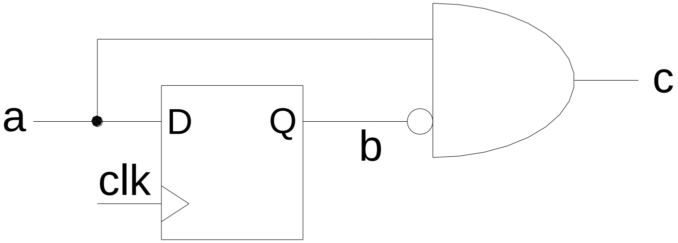




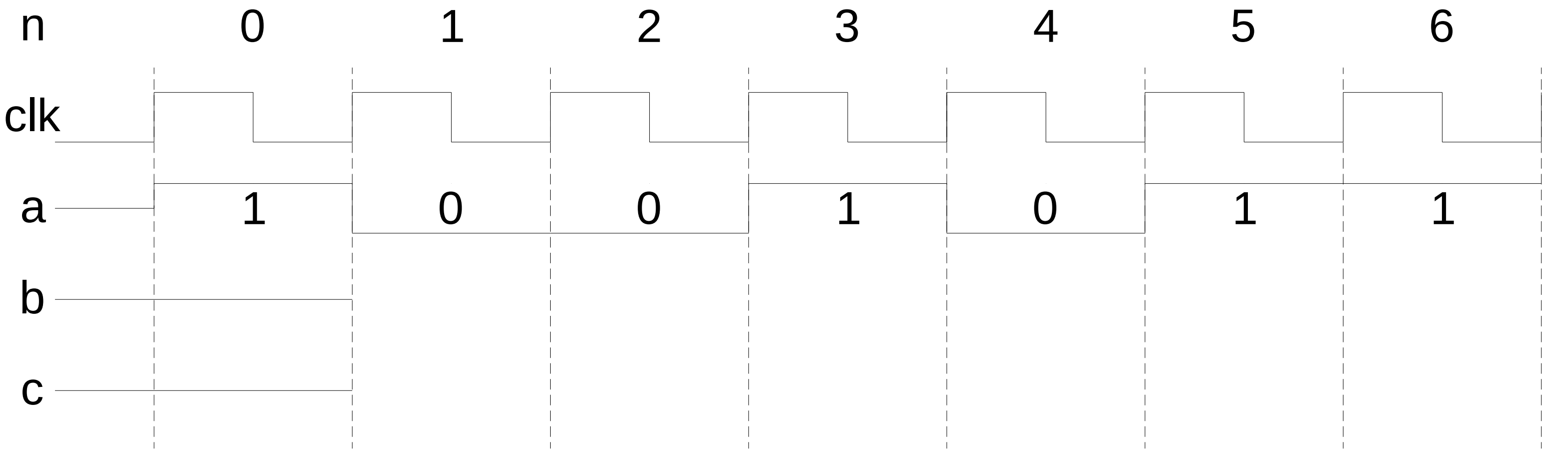
n	0	1	2	3	4	5	6
a	1	0	0	1	0	1	1
b		1	0	0	1	0	1
c			1	0	0	1	0

- n: number of clock cycle.
- Signals only change in the rising edge of the clock.
- Each flip-flop delays the signal one clock cycle.

WHAT DOES THIS CIRCUIT DO?

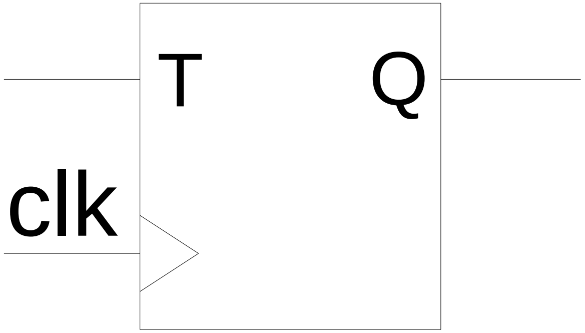


1. Complete the timing diagram:



2. What is the function of the circuit?

T FLIP-FLOP (T-VIPPA)

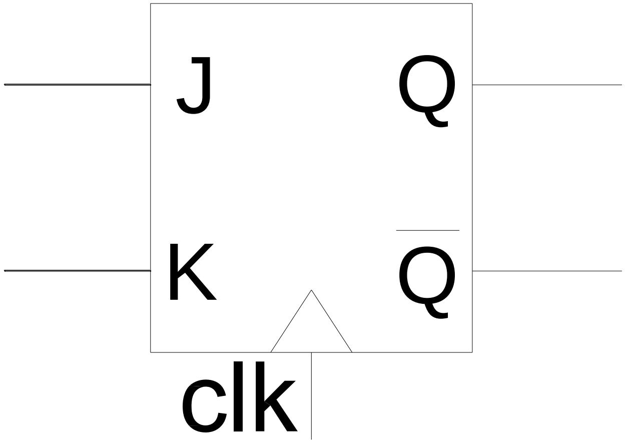


T(n-1)	Q(n-1)	Q(n)
0	0	0
0	1	1
1	0	1
1	1	0

- Behavior: every time that there is a transition in clk (from “0” to “1”), the value stored in the flip-flop toggles if T=1, and keeps the same value if T=0. The value stored in the flip-flop is always provided at the output Q.

- The output at the n-th clock cycle is $Q(n) = T(n-1) \oplus Q(n-1)$.

JK FLIP-FLOP (JK-VIPPA)



J(n-1)	K(n-1)	Q(n)
0	0	Q(n-1)
0	1	0
1	0	1
1	1	$\overline{\text{Q(n-1)}}$

- Question 1: How can we create a D flip-flop by using a JK flip-flop and logic gates?

- Question 2: How can we create a T flip-flop by using a JK flip-flop and logic gates?

COUNTERS (RÄKNARE)

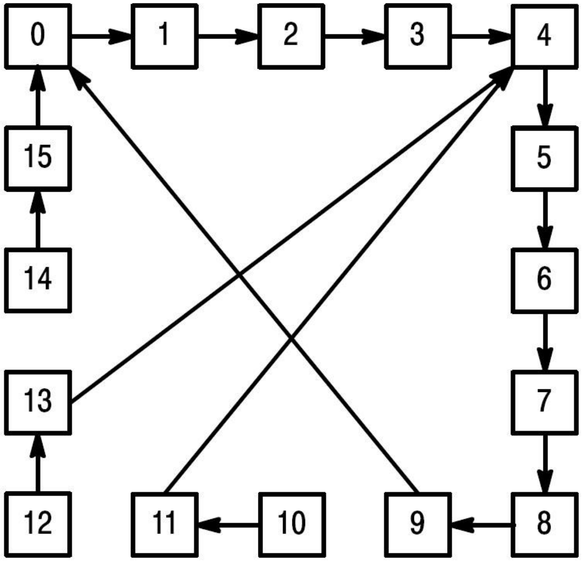
- Each clock cycle when the count is enabled, the counter counts.
- Generally the counters count up, but some counters can also count down.
- There are binary counters (0…2^(b-1)) and decade counters (0…9).

4-bit decade count (up)	5-bit binary count (down)
0000	11111
0001	11110
0010	…
…	00001
1001	00000
0000	11111
…	…

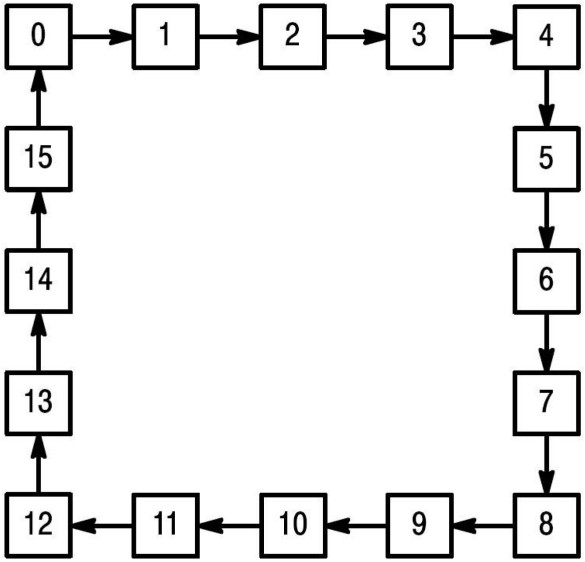
COUNTER CHIPS

- 74LS160: 4-bit decade counter.
- 74LS163: 4-bit binary counter.
- 74LS669: 4-bit up/down binary counter.

74LS160



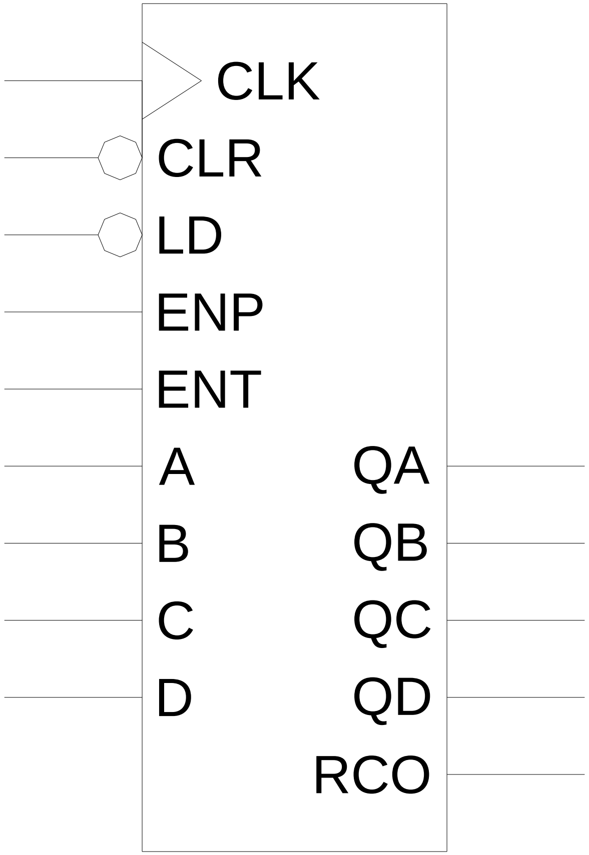
74LS163



- How would the 74LS669 count?

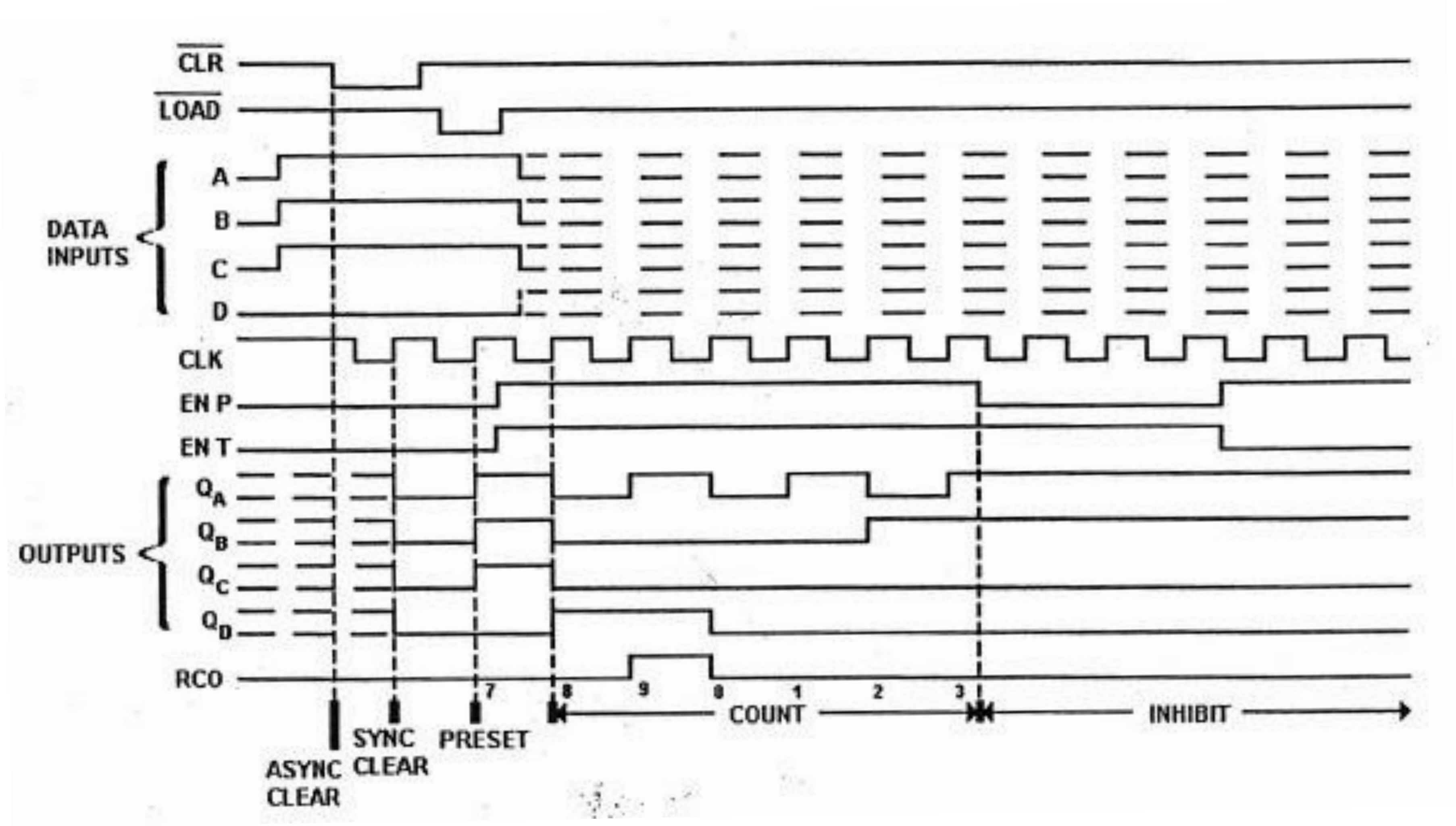
COUNTERS 74LS160 and 74LS163

74LS160 and 74LS163



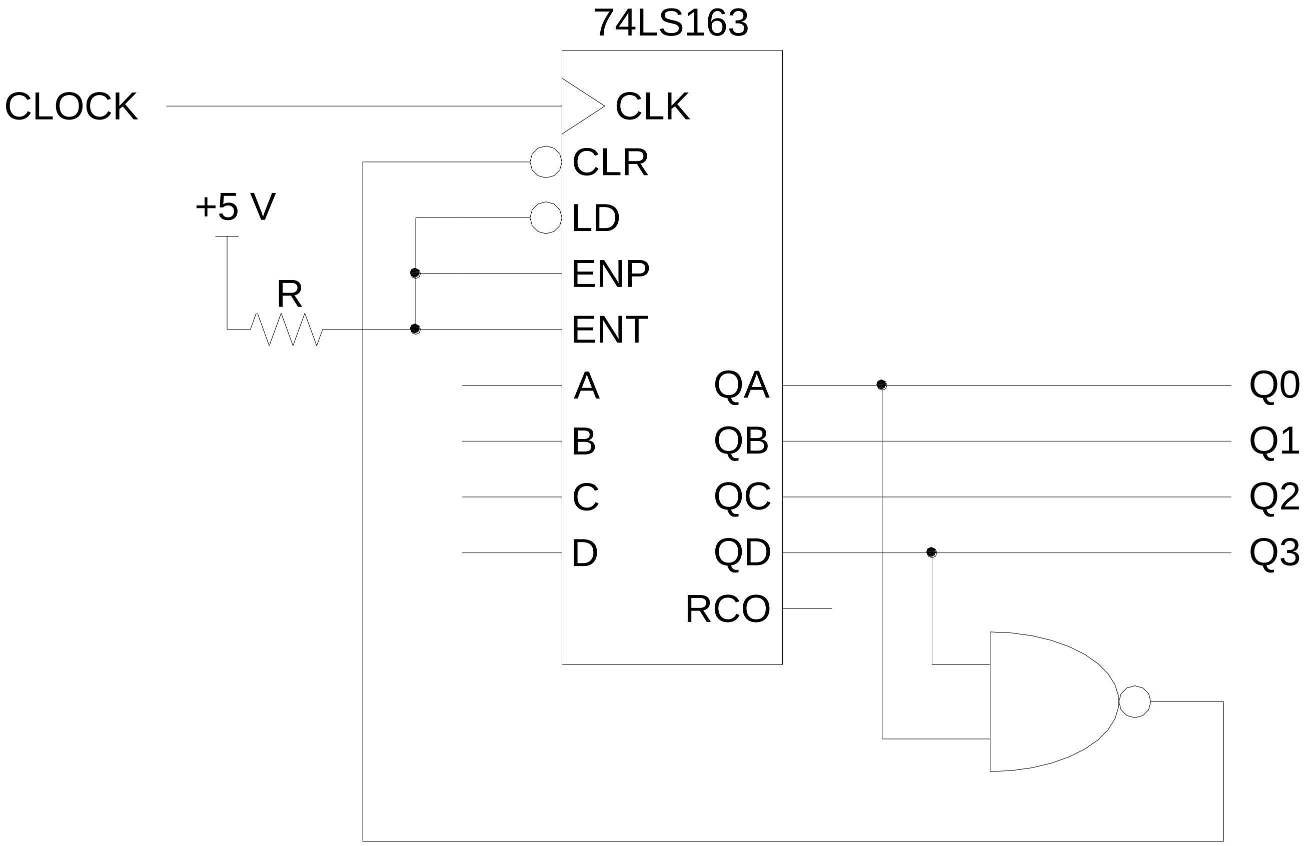
- By default, the counter counts up. After it reaches 9 (74LS160) or 15 (74LS163), it continues with 0. The count is shown in the bits [QD, DC, QB, QA].
- CLK: clock input.
- CLR (clear): sets the count to 0. Active LOW.
- LD (load): loads the value in [D,C,B,A] to the count. Active LOW. D is the MSB and A the LSB.
- ENP & ENT: enable signals. They have to be active to enable the count.
- RCO (Ripple-carry output): It is active when the counter reaches 9 (74LS160) or 15 (74LS163). Allows for interconnecting several counters and make a bigger one.

TIMING DIAGRAM 74LS160



COUNT FROM 0 TO 9 WITH A 74LS163

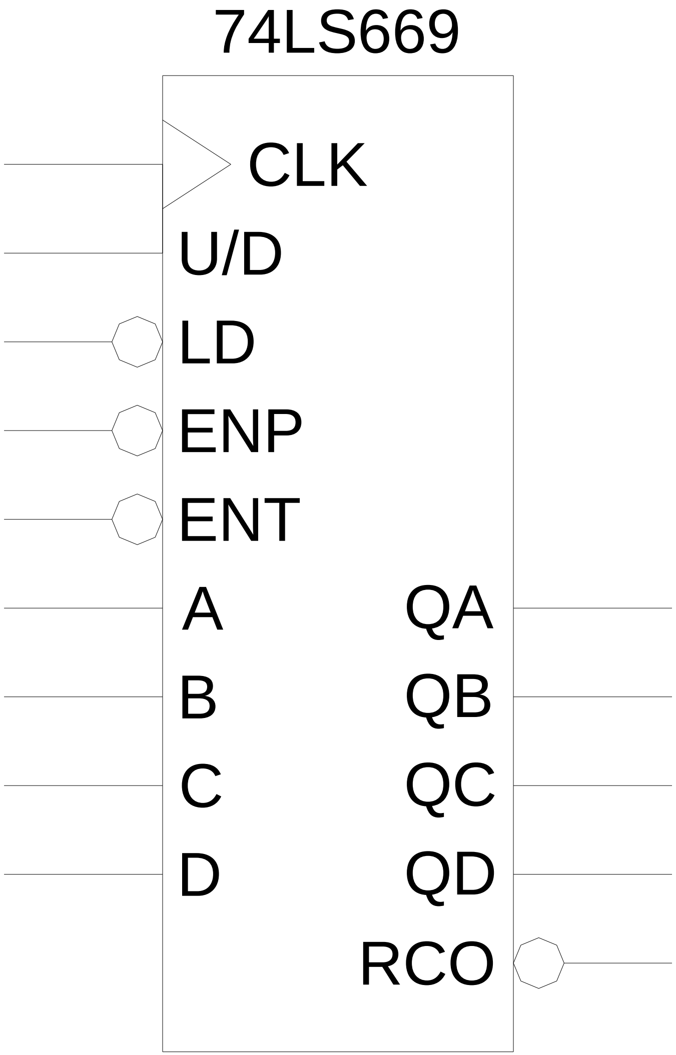
- The 74LS163 is a 4-bit binary counter. How can we make it count from 0 to 9?



EXERCISE

- Use a 74LS163 and logic gates to count cyclically from 5 to 10, i.e., 5,6,7,8,9,10,5,6,…

COUNTER: 74LS669



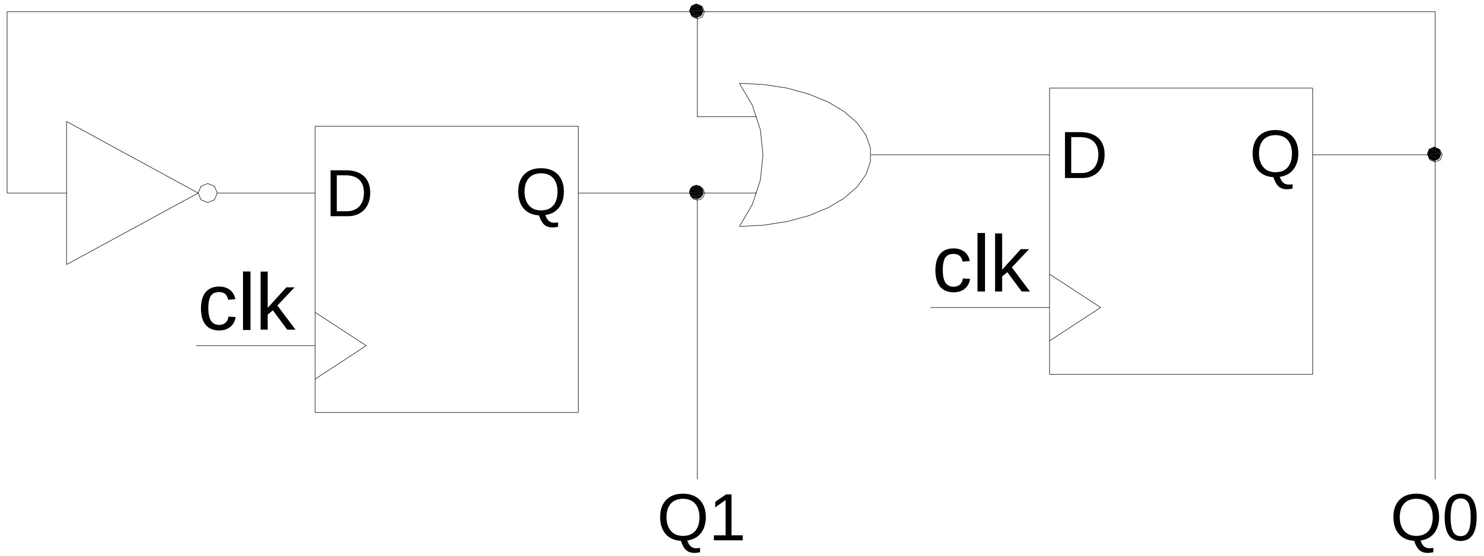
- Similar to the 74x163, but can count up and down using the signal UP/DN.
- Note that some signals are active LOW.

LAB 1

Time to finish the preparations of lab 1

EXERCISES (1)

- Draw the first 10 clock cycles of the timing diagram of the following circuit. Consider that the values in the registers are Q1 = 0 and Q0 = 0 at n = 0.

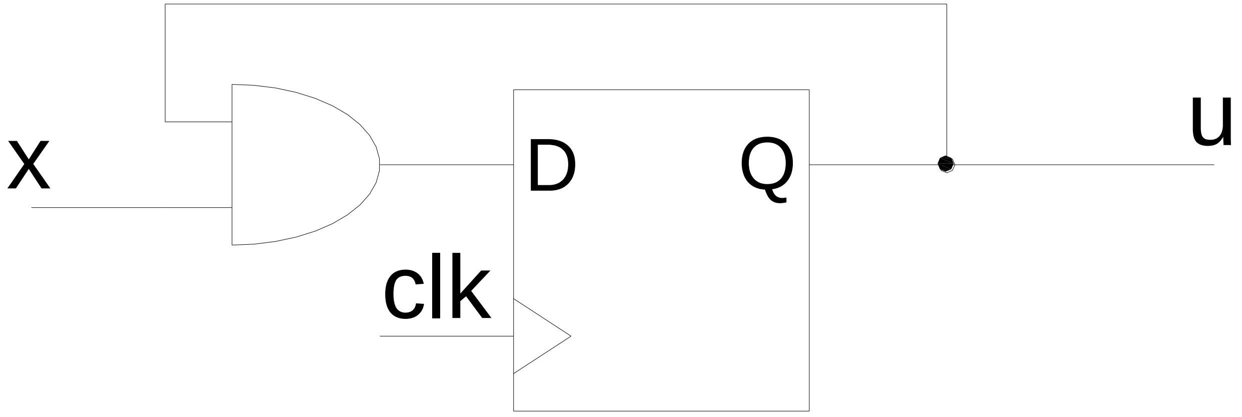


EXERCISES (2)

- Describe the behavior of this circuit. If you need it, you can make use of a timing diagram.

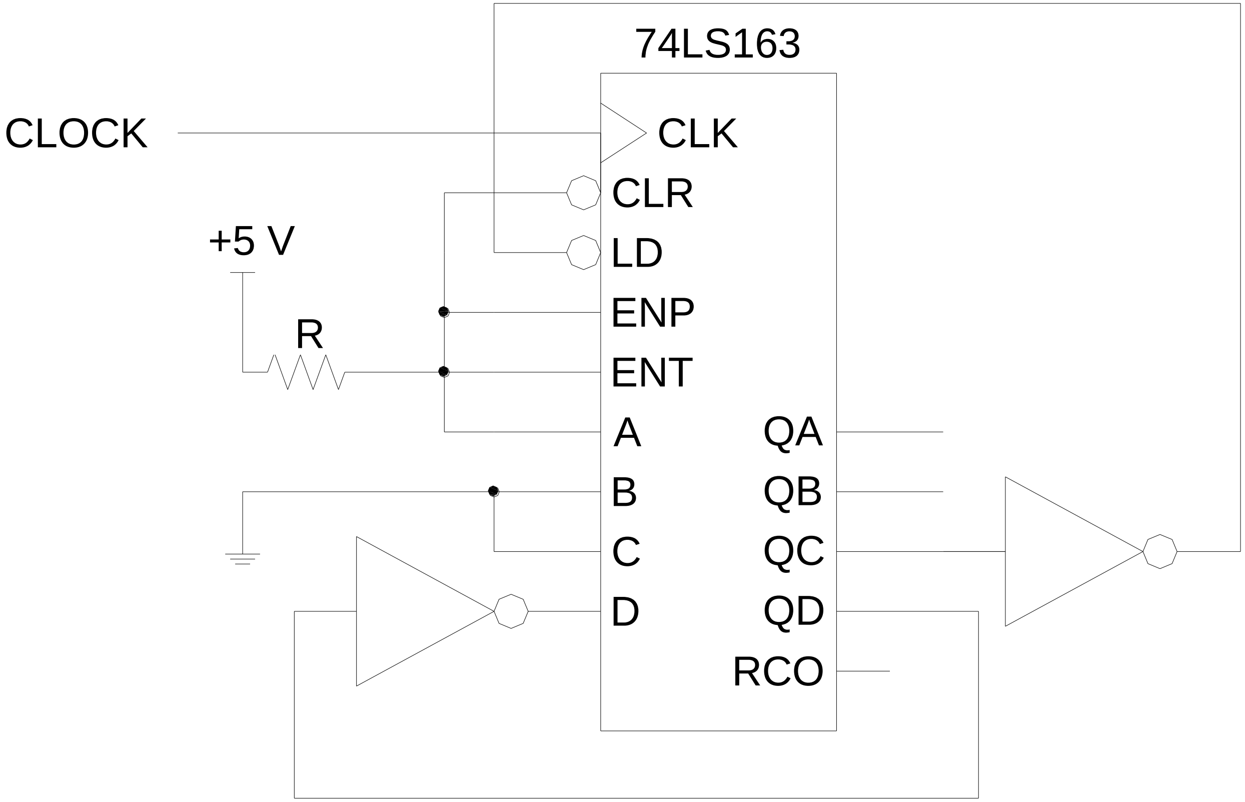
  a) When Q[0] = 0.

  b) When Q[0] = 1.



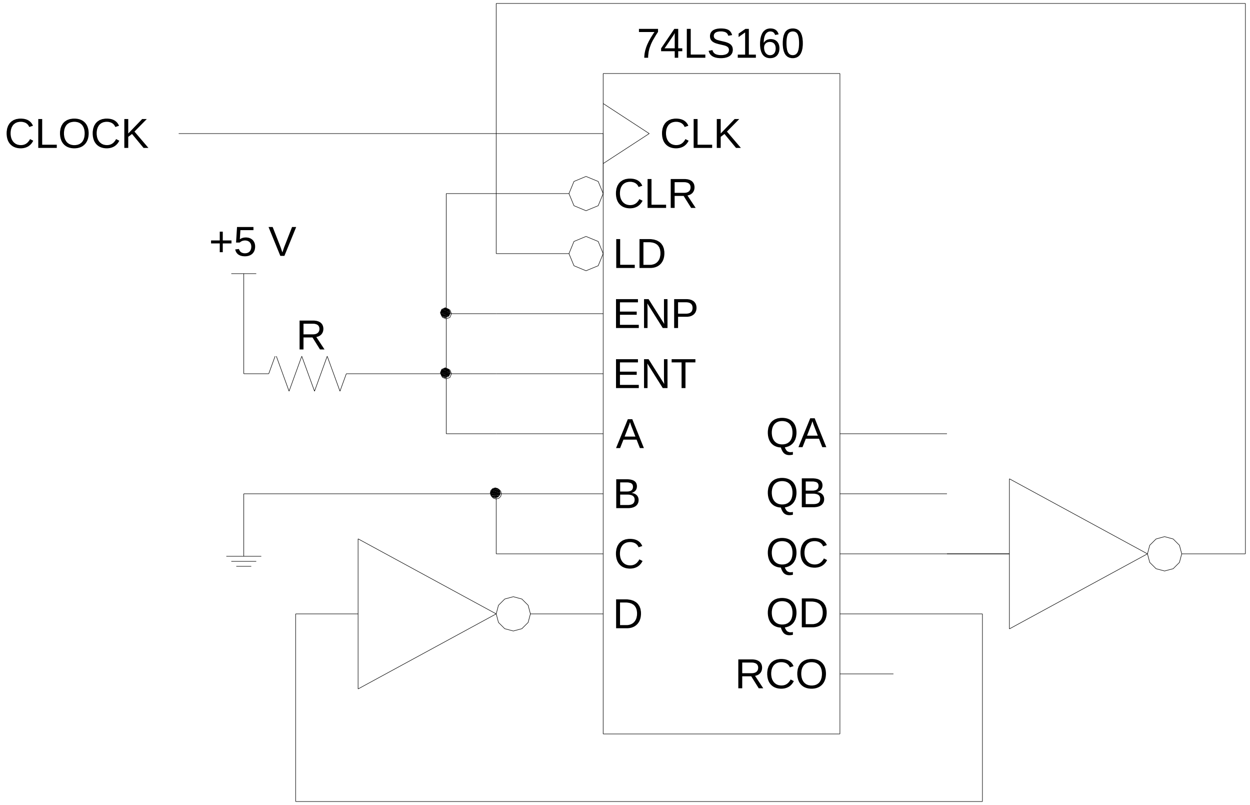
EXERCISES (3)

- What does this circuit do? If you need it, you can make use of a timing diagram.



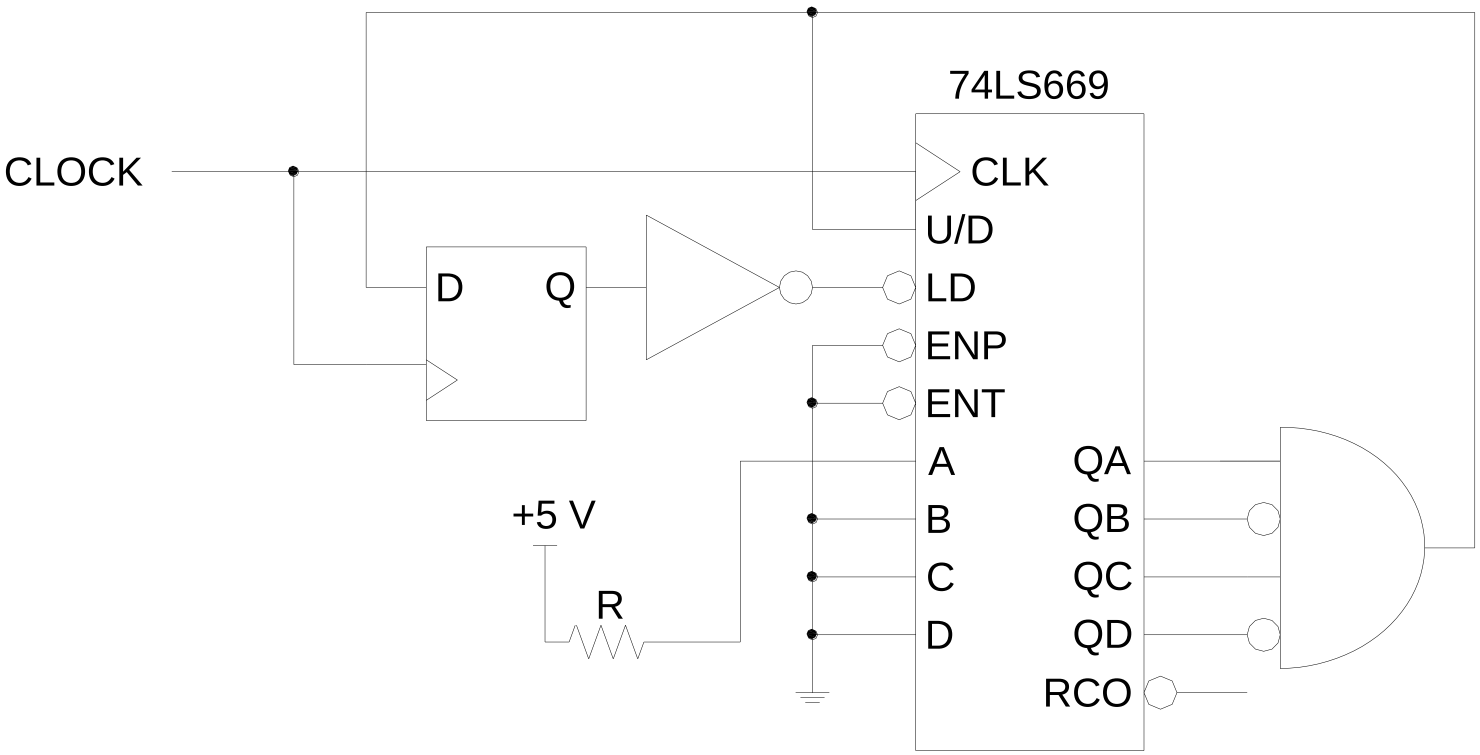
EXERCISES (4)

- What does this circuit do? If you need it, you can make use of a timing diagram. The only difference with respect to the previous exercise is that we are using a 74LS160 instead of a 74LS163.



EXERCISES (5)

- What does this circuit do? If you need it, you can make use of a timing diagram.



LINKÖPING UNIVERSITY

Department of Electrical Engineering

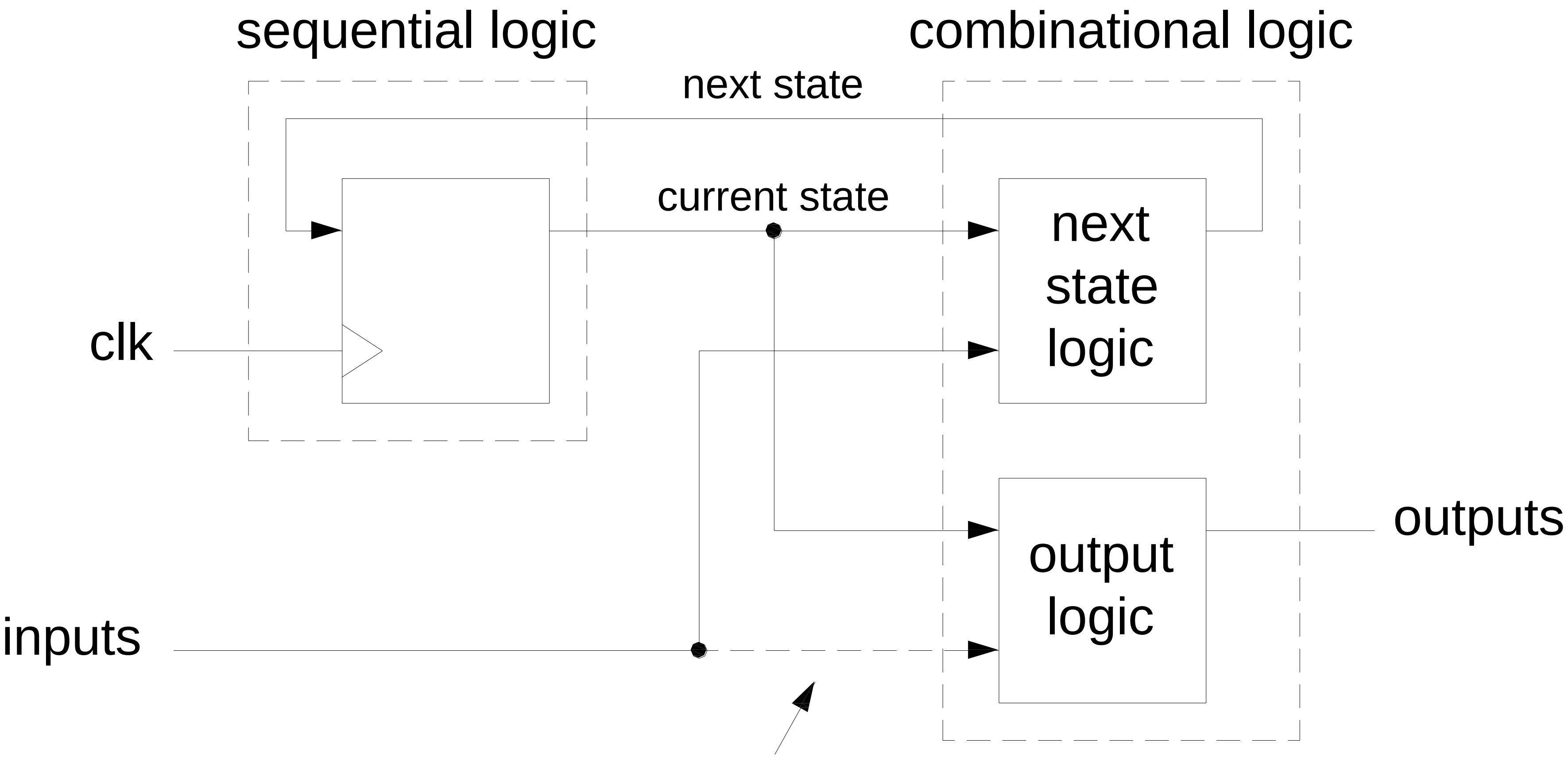
TMEL53, DIGITALTEKNIK

STATE MACHINES

Mario Garrido Gálvez
mario.garrido.galvez@liu.se

Linköping, 2017

FINITE-STATE MACHINES (TILLSTÅNDSMASKIN)



- Moore: The output is obtained from the current state => it only changes when the clock changes.
- Mealy: The output is obtained from the current state and from the inputs=> it may change at any time due to a change in the inputs.

FINITE-STATE MACHINES

- A “finite-state machine” or just “state machine” is a circuit that evolves through different states (tillstånd).
- It can only be at one state at a time.
- Every clock cycle the state is updated. The transitions among states and the output values depend on the current state and on the input values.
- It combines sequential and combinational logic. The sequential logic consists in one or several registers that store the current state. The combinational logic is used to calculate the next state and the outputs.

HOW TO DESIGN A STATE MACHINE

1) Define the states of the state machine:

-Moore: One state for each combination of internal variables.

- Mealy: One state for each combination of internal variables and input values.

2) Draw the state diagram.

3) Do the coding of the states.

4) Draw the transition table. The transition table is a truth table with the input signals and current states as inputs. From them we calculate the values for the next state and for the output signals.

5) Obtain the logic function of each of the variables that define the next state as a function of the inputs and the current state. Obtain the logic function of each output as a function of the current state (and also the inputs in case of Mealy).

6) Implement the circuit.

HOW TO ANALYZE A STATE MACHINE

The analysis is the inverse process to the design of a state machine: It goes from the circuit to the state diagram. The steps are:

1) Obtain the logic functions: the next states and outputs as a function of the current states and inputs.
2) Draw the transition table. Put input signals and current states as inputs. From them, calculate the values for the next state and for the output signals.
3) Draw the state diagram.