Computers Fundamentals

Subject 4. Sequential circuits

Index

- Introduction to sequential systems (SS)
  - Classification of Sequential systems
    - **Asynchronous** sequential systems (SSA)
    - **Synchronous** sequential systems(SSS)
- Basic sequential systems
  - Storage register
  - Registers bank, Memory
  - Shift registers
  - Counters

Basic sequential systems

- The most useful sequential circuits are:
  - Storage registers
    - They can store *N* bits of data
  - Register bank
    - Set of storage registers
  - Memory
    - Its functioning is similar to a register bank.
    - Main differences: capacity (number of storage registers, speed, design)

Basic sequential systems

- More useful sequential circuits are:
  - Shift register
    - It stores N bits.
    - It can *shift* data to the left or to the right
  - Counter
    - The circuit changes the output value every active edge of the clock signal.
    - The ouput values are a sequence of predefined values
      - In general the count is binary (up-count or down-count)

Storage register

FCO

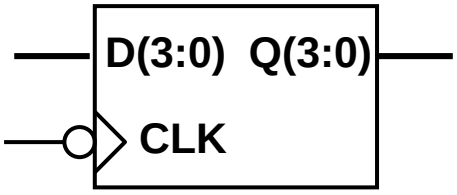
- Set of D flip-flops
  - The number of D flips flops is given by the number of bits that we want to store
    - The D flip-flop can store one bit
  - The inputs and outputs of the register are the same as the inputs and outputs of the D flip-flops
  - The input clock of all the D flip-flops is connected to the same signal. The signal is named clock
    - All the d flip-flops are identical. They are triggered by the same edge of the clock signal..

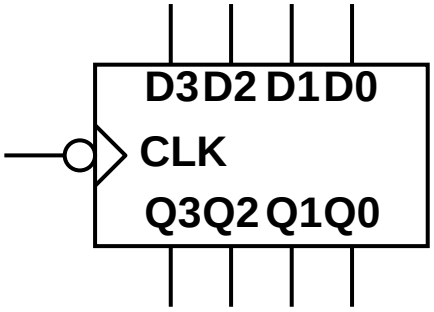
DISCA

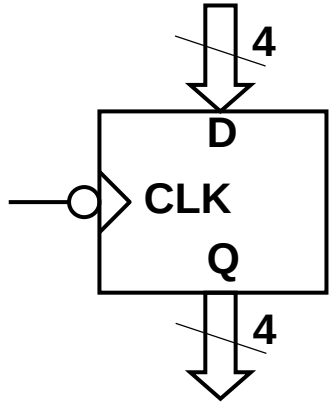
Storage register

FCO

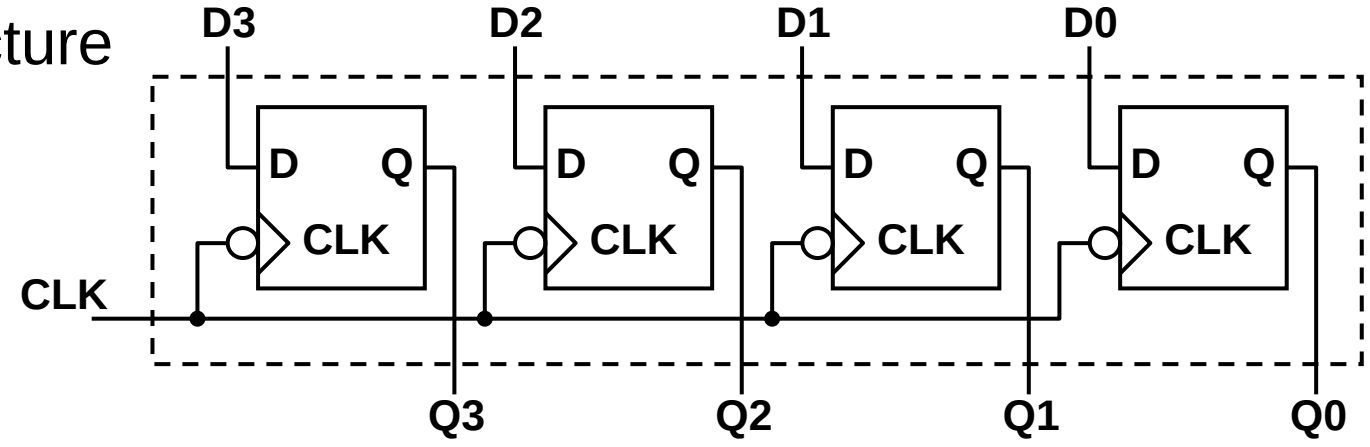
- Example: 4-bit storage register falling-edge triggered.
  - Equivalent logic symbols







  - Internal structure



Storage register

FCO

- Writing operation
  - Destructive operation
    - The data stored into the register is updated with the new data at the inputs of the register. The writing is done whit the clock edge.
  - How writing operation is performed:
    1) Set the input-value of each D flip-flop.
    2) Change clock signal in order to obtain an edge
- Reading operation
  - Non destructive operation
    - The data stored into the register is not affected
  - How reading-operation is performed
    - Examining the outputs of the D Flip-flops

DISCA

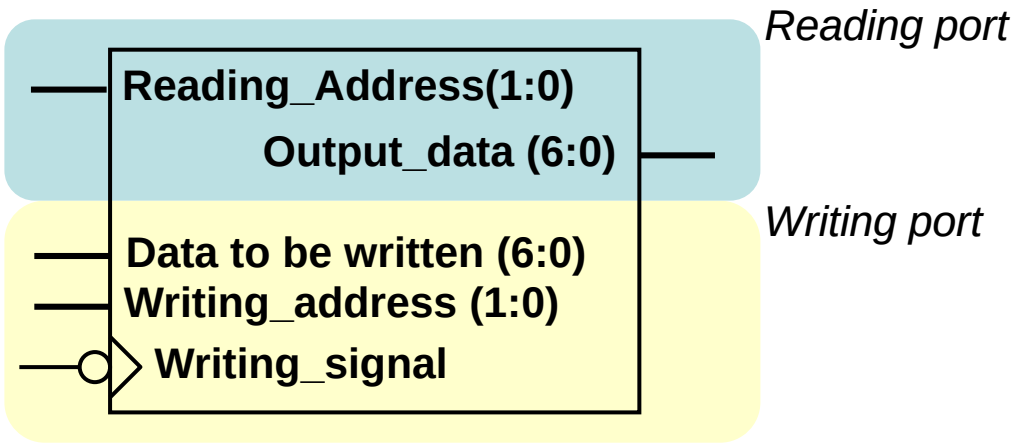
Registers Bank

- Set of registers which allow us to:
  - Write data into one register
    - It is mandatory to select what register we want to write
      - **Writing address**
  - Read the data stored in a register
    - It is mandatory to select what register we want to read
      - **Reading address**
    - We will design a register bank which accepts ***two reading address*** in order to obtain (at the same time) the value of two registers in the data outputs

Registers Bank

FCO

- ***Example***: Register bank of 4 registers. Each register can store 7 bits. The register bank has reading address bus and one writing address bus. The reading and the writing bus are composed by 2 bits each one.
  - 4 Registers => 2 bits used to select the 4 different address
  - Input/Output data. 7 bits each signal
  - Logic symbol

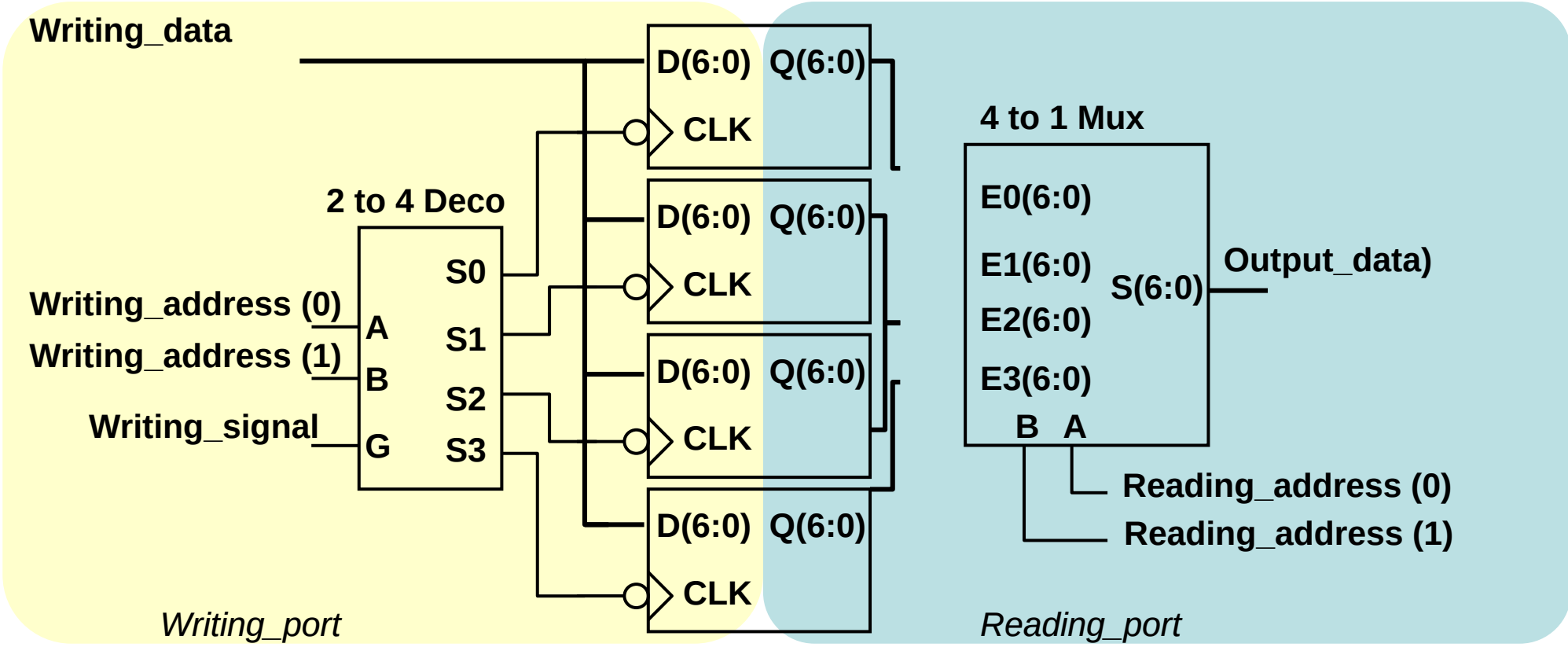


DISCA

Registers Bank

FCO

- Example(cont):
  - Internal structure



Registers Bank

FCO

- Writing operation
  - Steps
    1) Set the data that will be written into the register
    2) Set the writing address
    3) Modify the writing_signal to obtain an edge
       - Enabling and disabling the decoder
- Reading operation
  - Steps
    1) Set the reading address
    2) Examining the value of the output_value bus

DISCA

Memory

- Its functionality is like the functionality of a *register bank* with the following differences:
  - Much more capacity (Kbytes, Mbytes, Gbytes, …)
  - Much more slower
  - The technologies used in their implementations is different
- At any time only can be performed one operation (reading or writing)
  - It has one input-address bus and one input-data bus
  - Allowed operations: reading, writing or none

Shift register

FCO

- Set of D flip-flops
  - It is mandatory to have as many D flip-flops as bits to store
  - Several clock-cycles are necessary to insert or pull-out data
    - In this case it is said that the shift register has one serial input and one serial output
    - When all the input data bits are inserted or are pulled-out using only one clock cycle, it is said that the shift-register has parallel-input and parallel-output
  - Structure with serial-input
    - The first flip-flop input is connected with the input data
    - The rest of the flip-flops connects their input with the output of the preceding flip-flop
  - Structure with serial output
    - There is only one output and is the output of the last flip-flop

DISCA

Shift register

- Is a cascade of flip flops, sharing the same clock, which has the output of any one but the last flip-flop connected to the "data" input of the next one in the chain
- The shift register circuit shifts by one position the one-dimensional "bit array" stored in it, *shifting in* the data present at its input and *shifting out* the last bit in the array, when enabled to do so by a transition of the clock input.

Shift registers

FCO

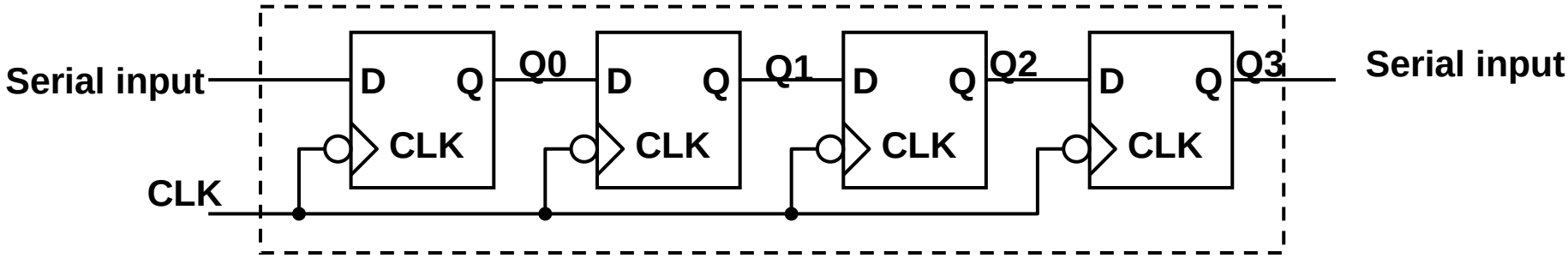
- Shift registers can have both parallel and serial inputs and outputs.
  - These are often configured as **serial-in, parallel-out** (SIPO) or as **parallel-in, serial-out** (PISO).
- There are also types that have both serial and parallel input and types with serial and parallel output.
- There are also **bi-directional** shift registers which allow shifting in both directions: L→R or R→L. The serial input and last output of a shift register can also be connected together to create a **circular shift register**.

DISCA

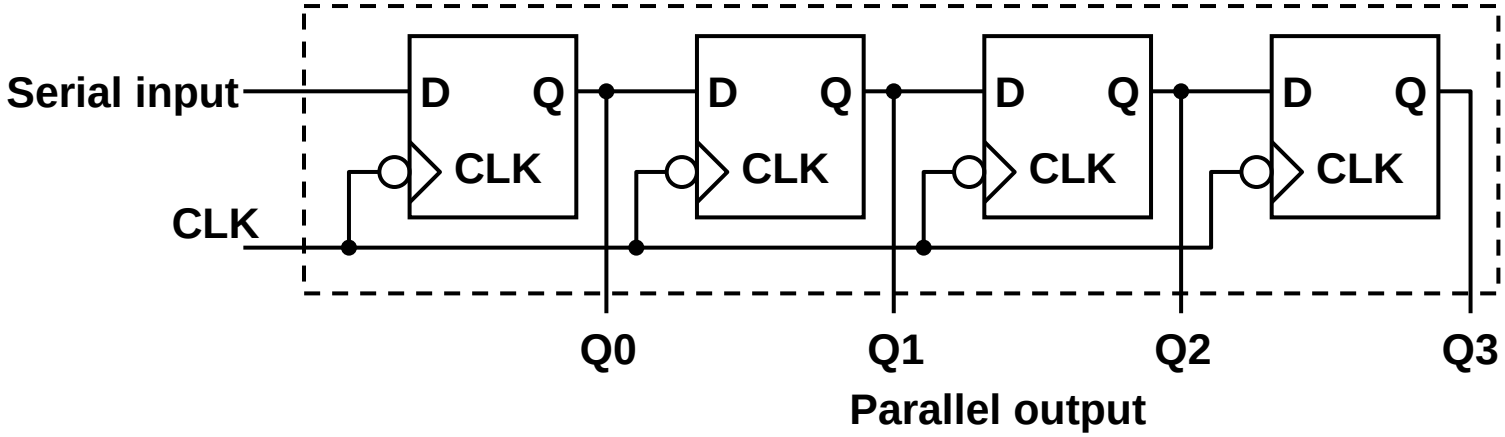
Shift-registers:

FCO

- Serial input, serial output (SISO) shift register:

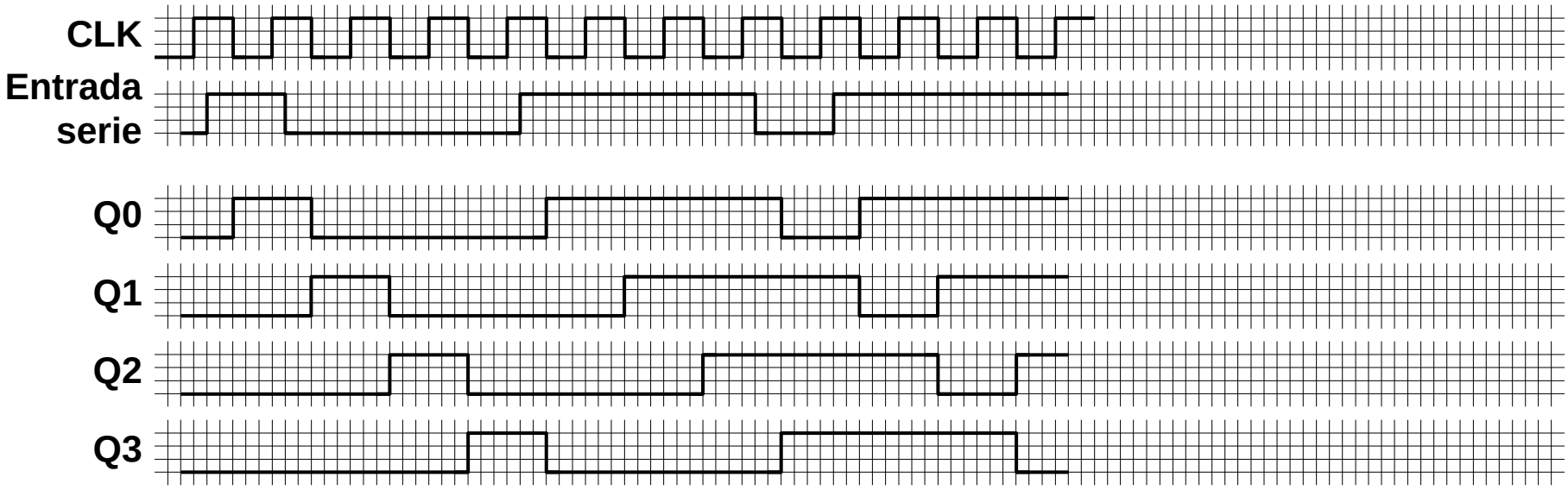


- Serial input, parallel output (SIPO) shift Register



Shift register

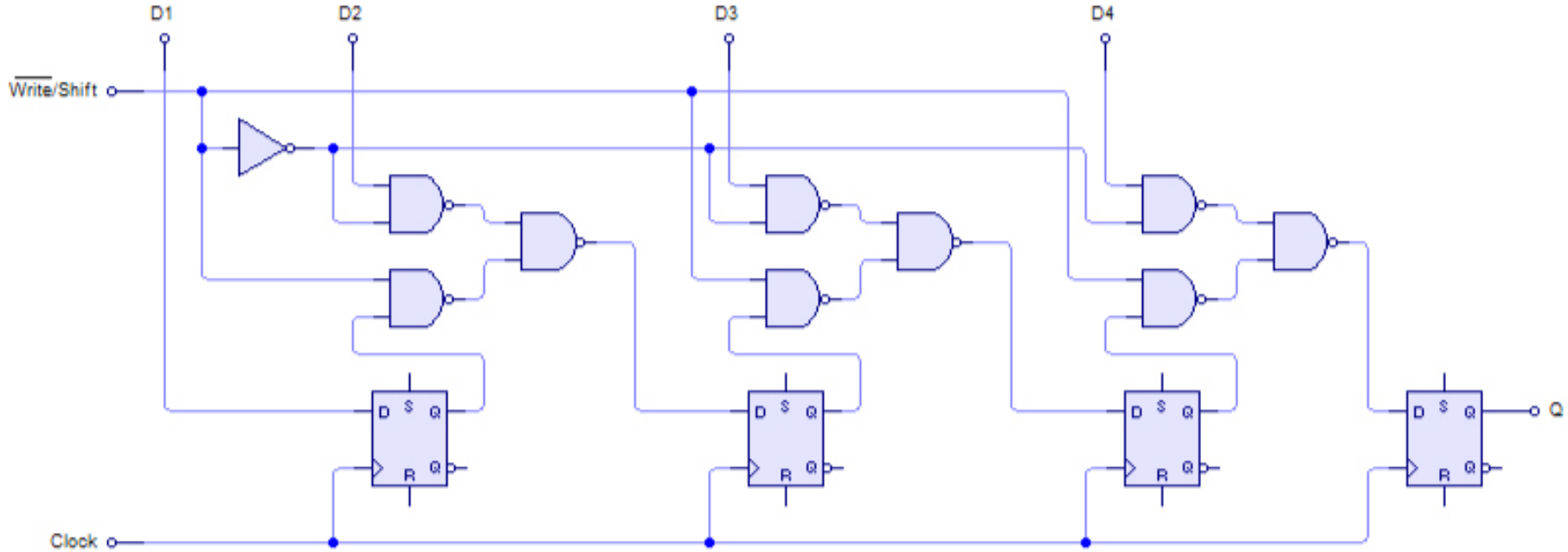
- SIPO Shift Register
  - Example
    - Initial state Q3Q2Q1Q0 = 0000
    - The serial-input takes the values indicated in the timing diagram



Shift register

FCO

- Parallel Input/serial output



Shift register (Summary)

- Can shift data to the left or can shift data to the right
  - There is no unanimity in how to define what means ***shift to the left*** or ***shift to the right***. The direction must be defined explicitly
- Can shift in both directions
  - Not at the same time
  - A direction-input is added plus two serial-inputs used to put the bit that will be shifted to the left and the bit which will be shifted to the right
- Data can be inserted into the shift-register in two ways:
  - Serial (In each clock-cycle only one bit can be shifted into the register)
  - Parallel (all the data bits can be inserted in just one clock-cycle)
- The output can be pulled-out of the shift register in two ways:
  - Serial (One bit can be read from the output, each clock cycle the output changes)
  - Parallel (the whole data stored in the register can be read)

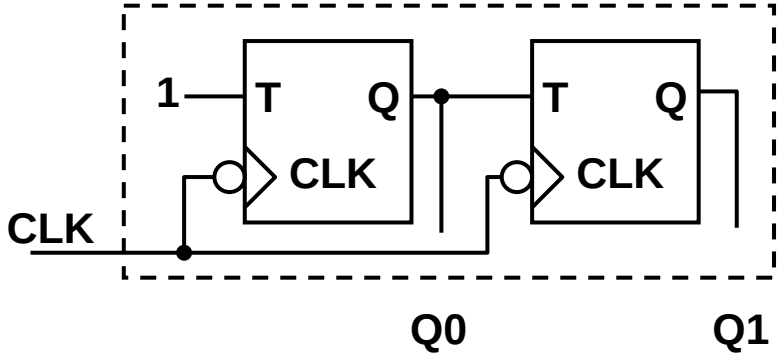
Counter

FCO

- Digital circuit whose output changes with each active edge of the clock.
- The output is a binary sequence that in decimal is a count.
- The count can be only up count or only down count
- The count can be up count or down count
  - A control-input is added to control if the count must be up count or down count
  - Never can be up count and down count at the same time!

Synchronous Counters FCO

- T flip-flops are used
  - Synchronouus circuits do not generate glitches
  - Example: 2 bits synchronous up-counter
    - Output count: 00_2,01_2,10_2,11_2



Counters

FCO

- Classification
  - Binary counter
    - Generates all the possible binary sequences using N bits.
  - N-module counters
    - Generates N different output binary sequences, where $N < 2^{\text{number of flip-flops}}-1$
      - Normally the output sequence includes the binary sequence equivalent to 0_{10}. The count obtained (in decimal) is into the range [0,N-1]
    - Examples
      - A ten-module counter outputs the binary sequence: 0000_2, 0001_2, 0010_2, 0011_2, 0100_2, 0101_2, 0011_2, 0111_2, 1000_2, 1001_2, 1100, 0000

DISCA