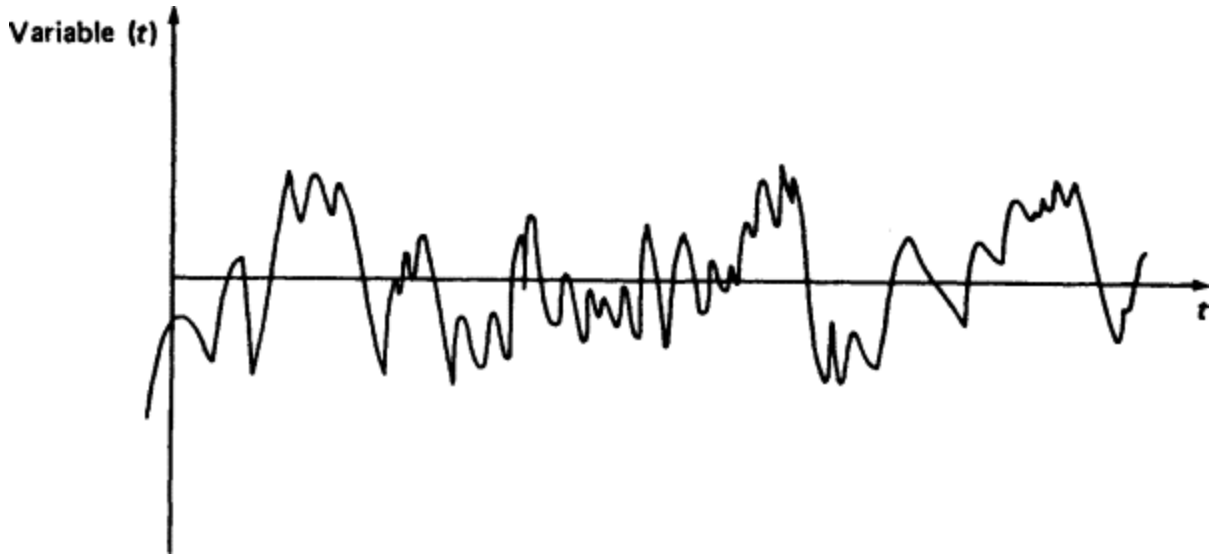
Introduction to digital (circuit) design (with FPGAs)

A. Paramonov (Argonne National Laboratory)

June of 2025

What is digital?

- **Is the signal below digital?**
- Raise your hand if you think it is a digital signal.
- Do not do that if you think it is analog.



What is digital?

- **What's the correct answer?**

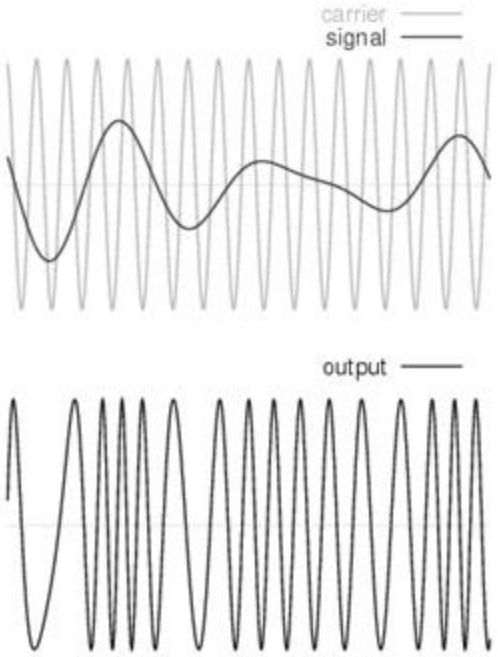
What is digital?

- **The answer is “It depends who you ask.”.**
- So everybody is right.

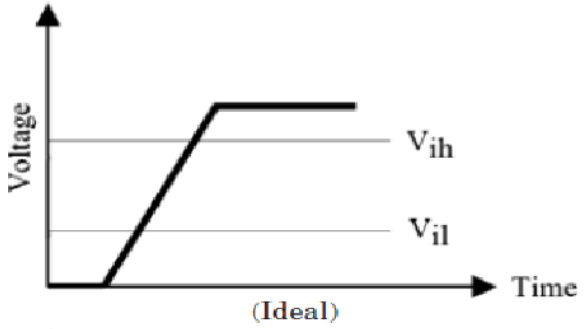
What is digital?

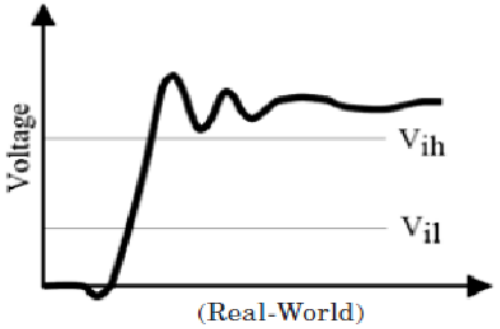
- It is how you interpret an electrical signal's parameters; voltages, currents.

- For analog signals there is a continuous spectrum of amplitudes that are meaningful.
- These signals can be modulated with amplitude, frequency, phase, or some other way.



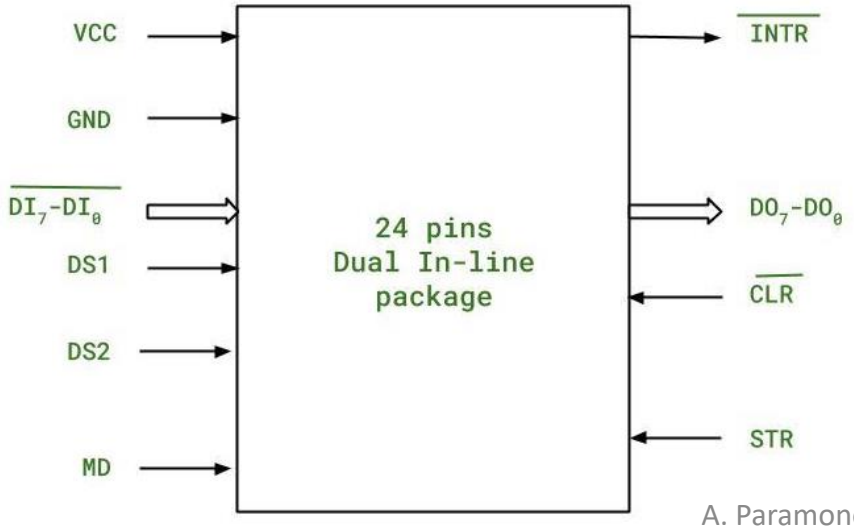
- For digital signals only selected ranges of the amplitudes are meaningful and interpreted as discrete values (e.g. 0 & 1).
- The observable of interest can be frequency, for example.
- FPGAs and modern CMOS circuits use voltage levels to signal binary data.





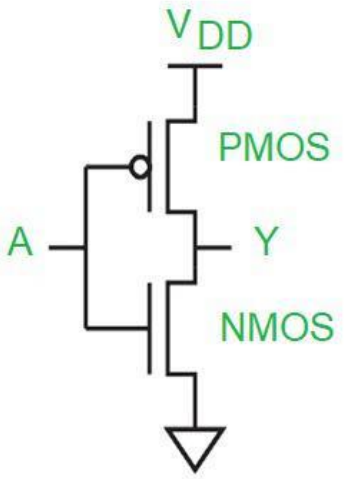
Inputs and Outputs Ports

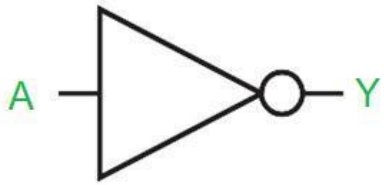
- Digital circuits have two kinds of connections
  - Power (VCC and GND)
  - Input ports: wires that need to driven externally
  - Output ports: wires driven by the digital circuit
  - Bi-directional ports (they can be controlled to acts as input or output)
- The block represents some circuit
- Thin lines represent wires
- Thick lines are data busses (groups of wires)



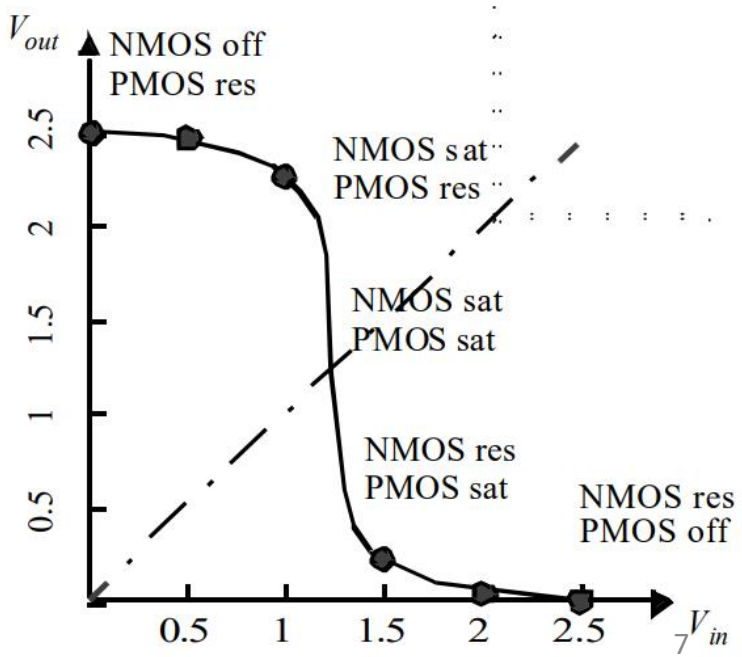
CMOS Inverter

- The majority of modern digital circuits are Complementary metal–oxide–semiconductor(CMOS).
- They use p- and n- Field-Effect Transistors.
- The static power consumption is low.
- The current is flowing from Vdd to Gnd (i.e. both transistors are open) when the inverter is switching (the input voltage is between Vdd and Gnd).



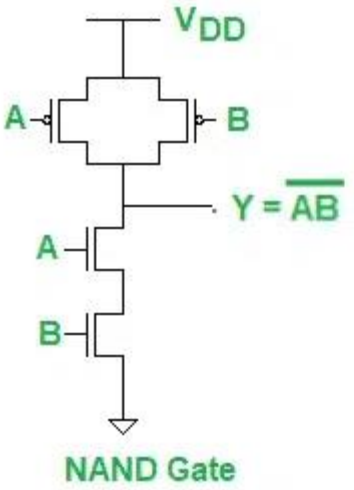


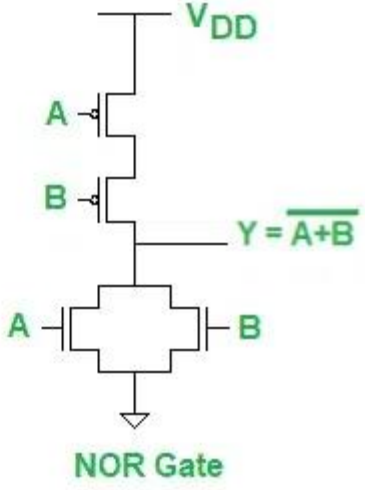
A	Y
0	1
1	0

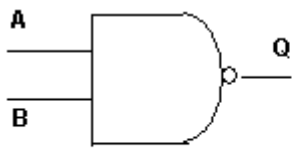


Combinatorial Logic

- Combinatorial logic circuits do not have memory; the output voltages change according to the input voltages.
- These circuits can be designed with only p- and n- transistors.
- These circuits have similar switching characteristics as the inverter (the transistors are ON or OFF when the inputs are stable).
- The gates below have two inputs but that can be larger.

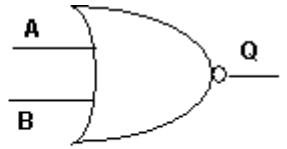






NAND

A	B	Q
0	0	1
0	1	1
1	0	1
1	1	0



NOR

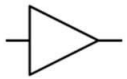
A	B	Q
0	0	1
0	1	0
1	0	0
1	1	0

Exercise → Design a 3-input NAND gate

(Boolean) Logic gates

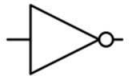
- And just like that we can design various universal logic gates to perform Boolean logic operations
- This allows us to design circuits using the logic gates abstraction instead of transistors.

Buffer



Input	Output
0	0
1	1

Inverter



Input	Output
0	1
1	0

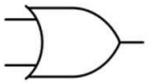
AND

A	B	Output
0	0	0
1	0	0
0	1	0
1	1	1

NAND

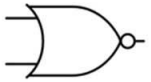
A	B	Output
0	0	1
1	0	1
0	1	1
1	1	0

OR



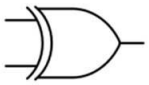
A	B	Output
0	0	0
1	0	1
0	1	1
1	1	1

NOR



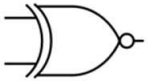
A	B	Output
0	0	1
1	0	0
0	1	0
1	1	0

XOR



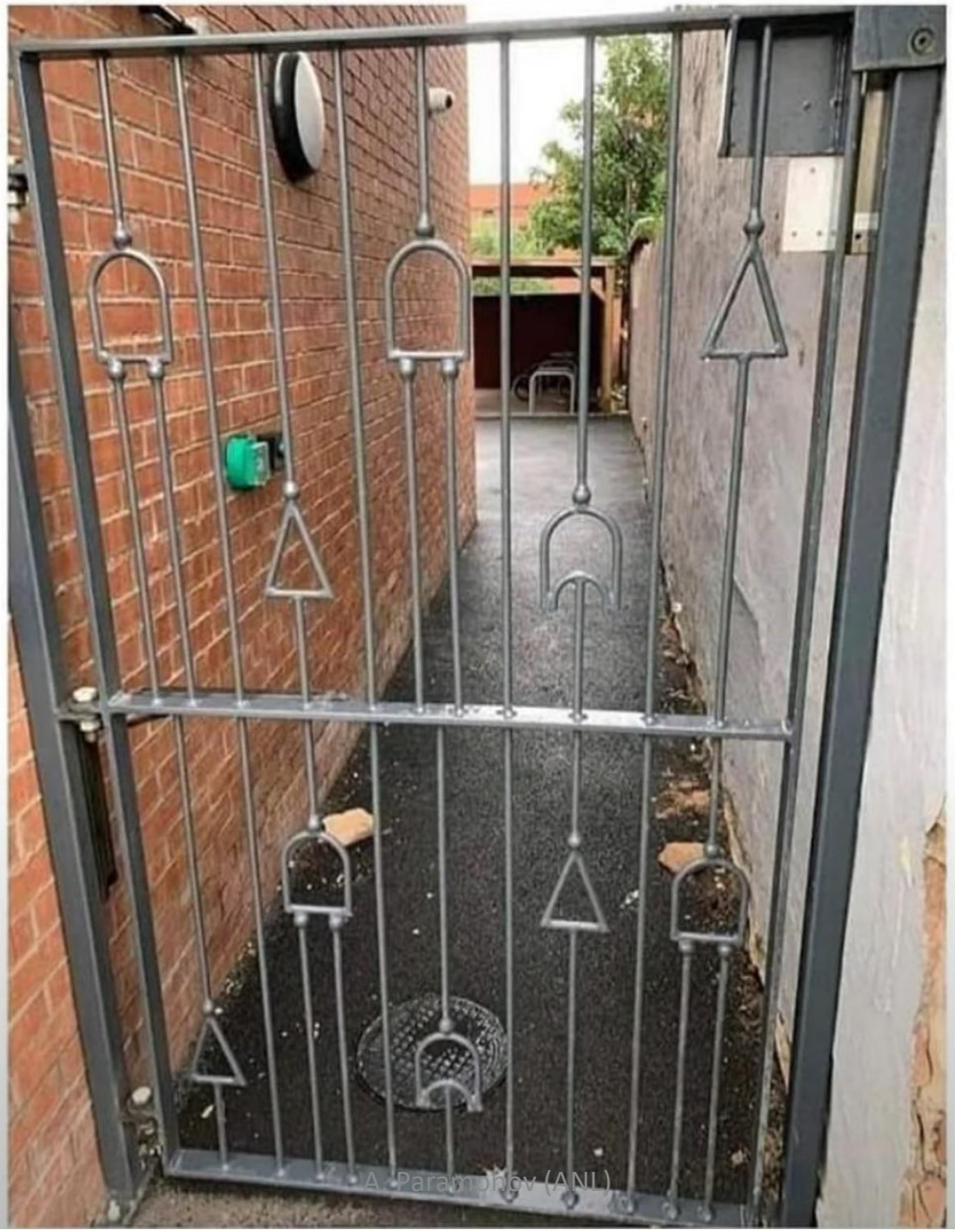
A	B	Output
0	0	0
1	0	1
0	1	1
1	1	0

XNOR



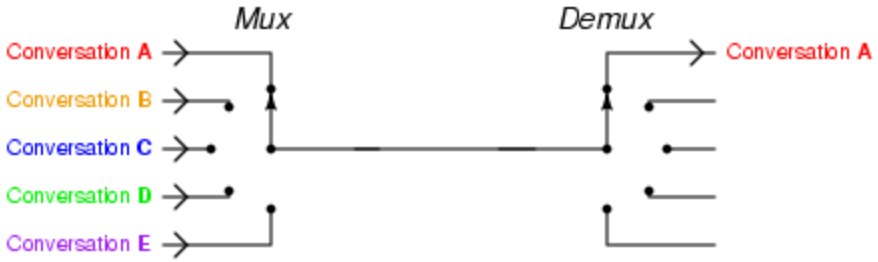
A	B	Output
0	0	1
1	0	0
0	1	0
1	1	1

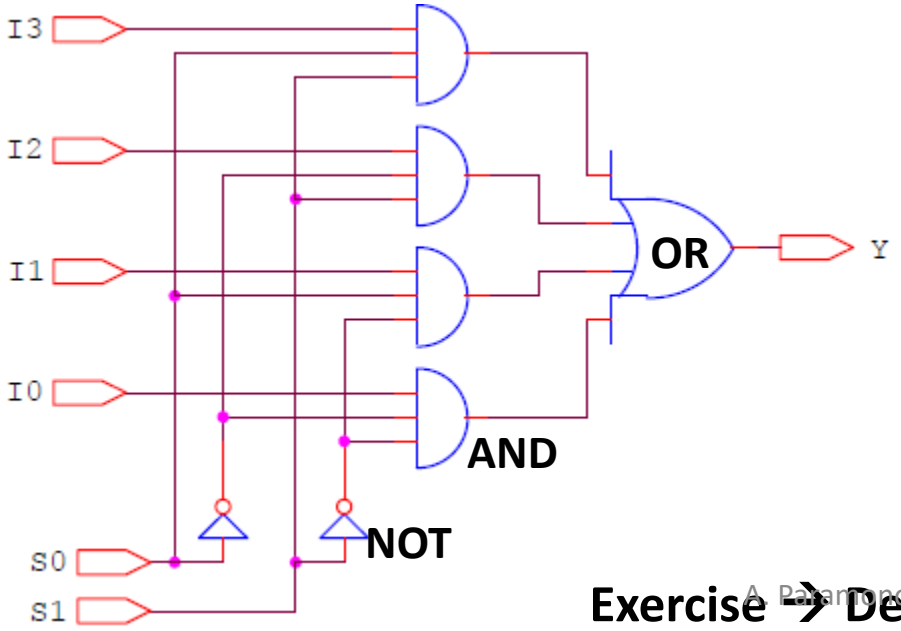
Logic Gate:

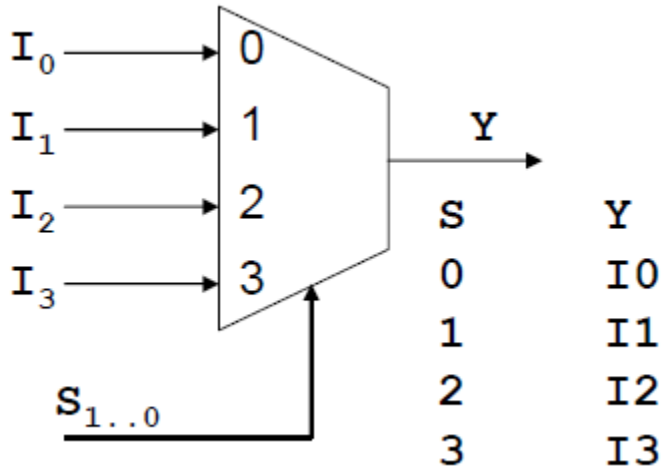


Multiplexer (aka MUX)

- A circuit that connects one of the inputs to the output. The choice of input is controllable with the select pins/ports.



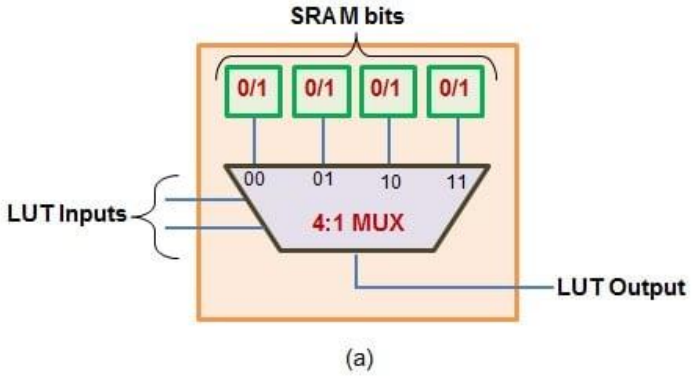




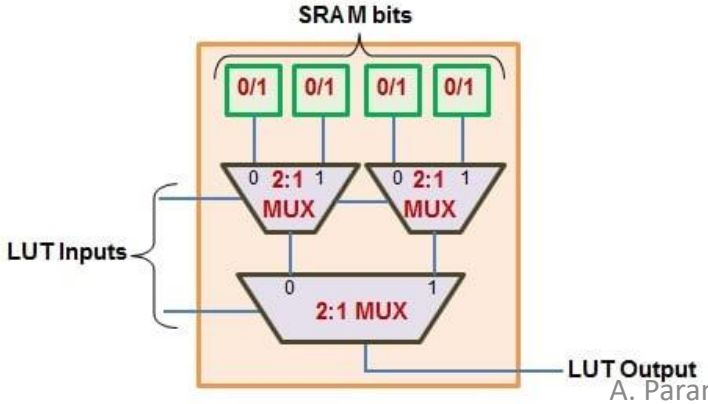
Exercise → Design a 4-input DEMUX with logic gates

Lookup Tables

- FPGAs use LookUp Tables (LUTs) to perform logical operations.
- A lookup table is a memory (input → Address & Data → output).
- SRAM = Static Random Access Memory



(a)



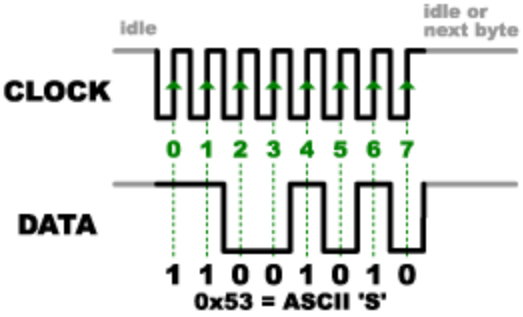
(b)

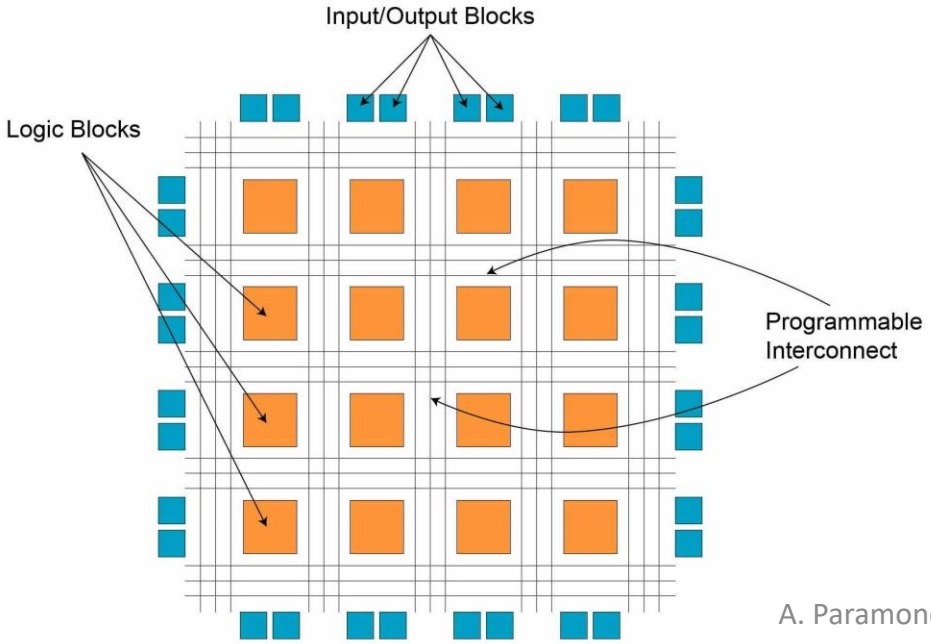
Truth Table

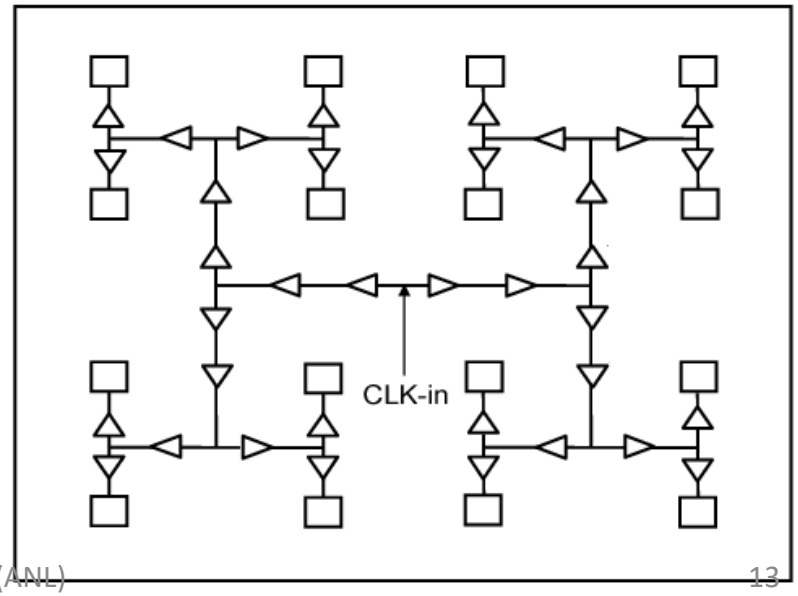
Inputs				Output
A	B	C	D	Y
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

Types of digital signals

- FPGAs have two types of signals: data and clock.
- The clock signal tell us when the sample the data signals.
- The data signals are distributed in an FPGA though the interconnect layer. The interconnect layer is flexibly and supports a large number of signals but suffers from routing delays.
- The clock signals are distributed through the clock distribution networks. There are few of them but they offer low propagation delays (skew).

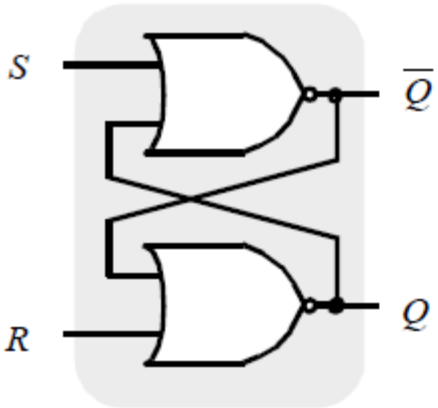




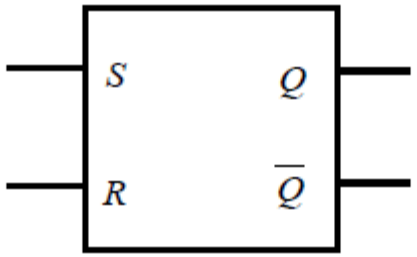


Sequential (Registered) Logic

- Outputs depend on the current AND previous input values
- i.e. these circuits have memory.
- SR flip-flop is an example of a sequential logic circuit
- They typically have internal feedback to have stable states.



(a) Schematic diagram



(b) Logic symbol

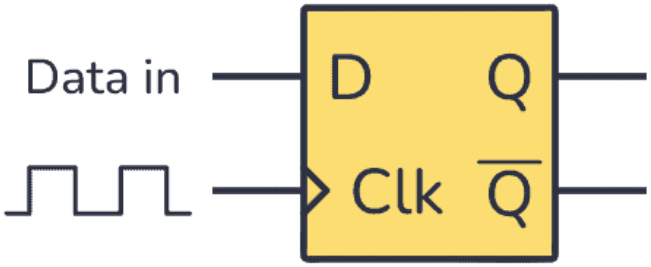
S	R	Q	$\overline{Q}$
0	0	Q	$\overline{Q}$
1	0	1	0
0	1	0	1
1	1	0	0

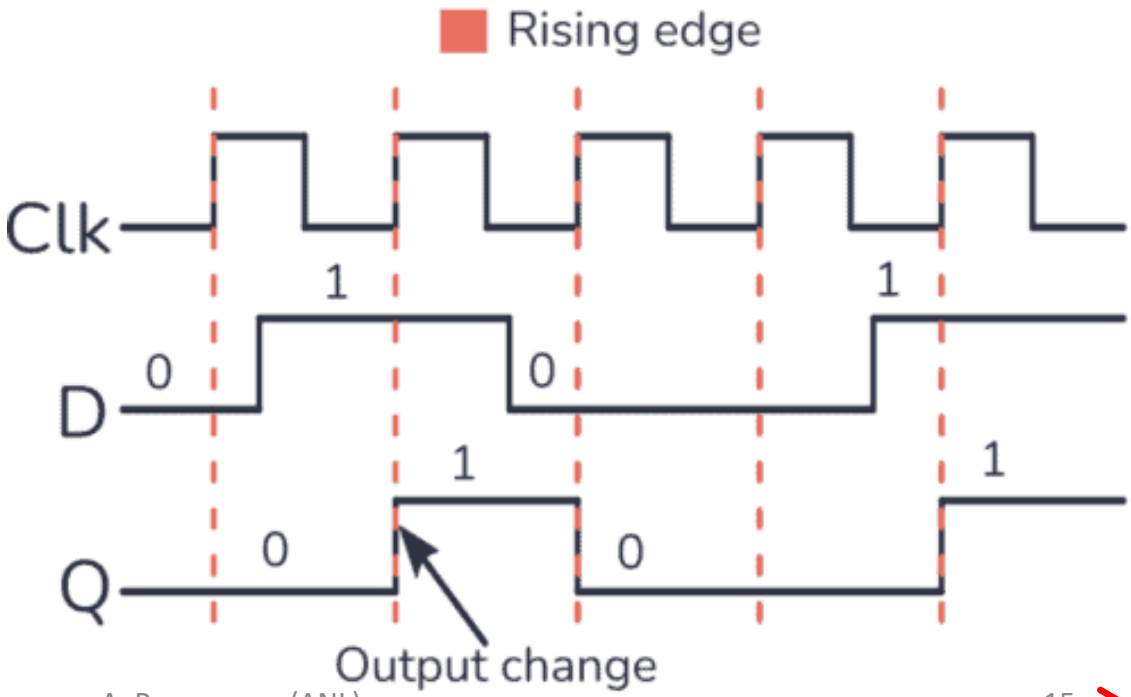
Forbidden State

(c) Characteristic table

DFF

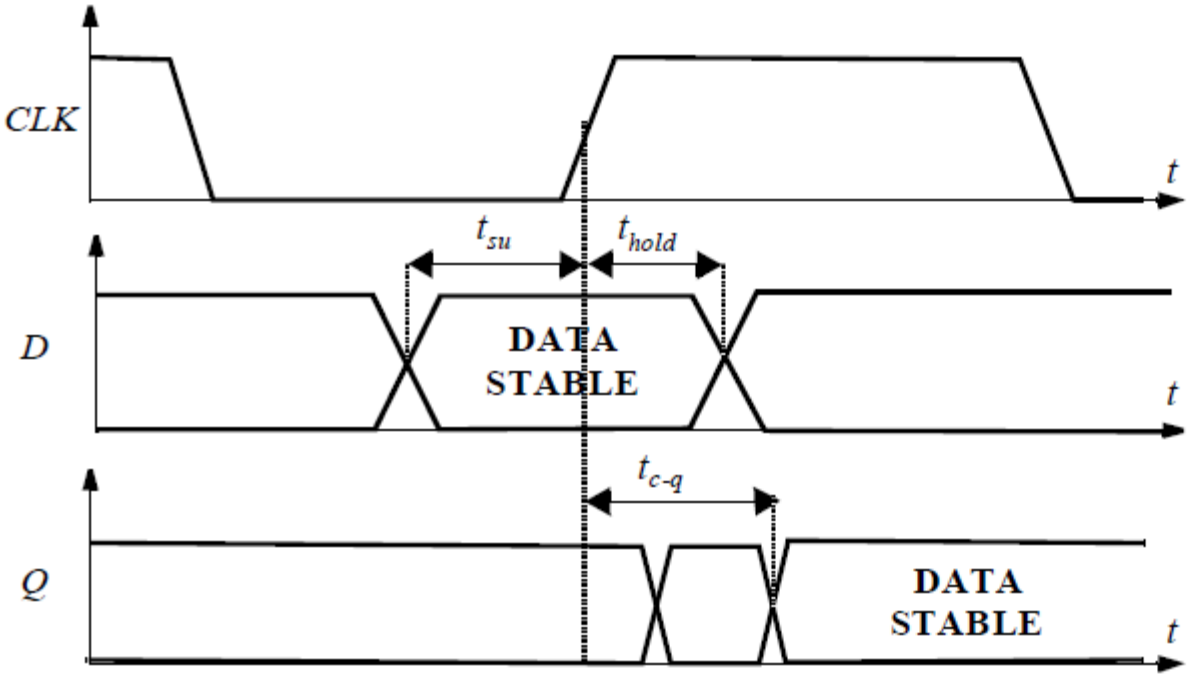
- D flip-flop is a register triggered by the rising edge of the clock signal.
- A DFF may have a synchronous or asynchronous reset





DFF Timing characteristics

- The data signal needs to be stable before and after the clock rising edge
- The DFF may be in a meta-stable state if the data is unstable during the clock transition.
- DFF output changes some time after the clock transition. That's propagation delay.



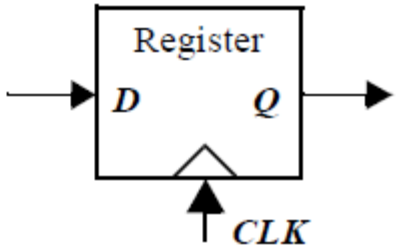
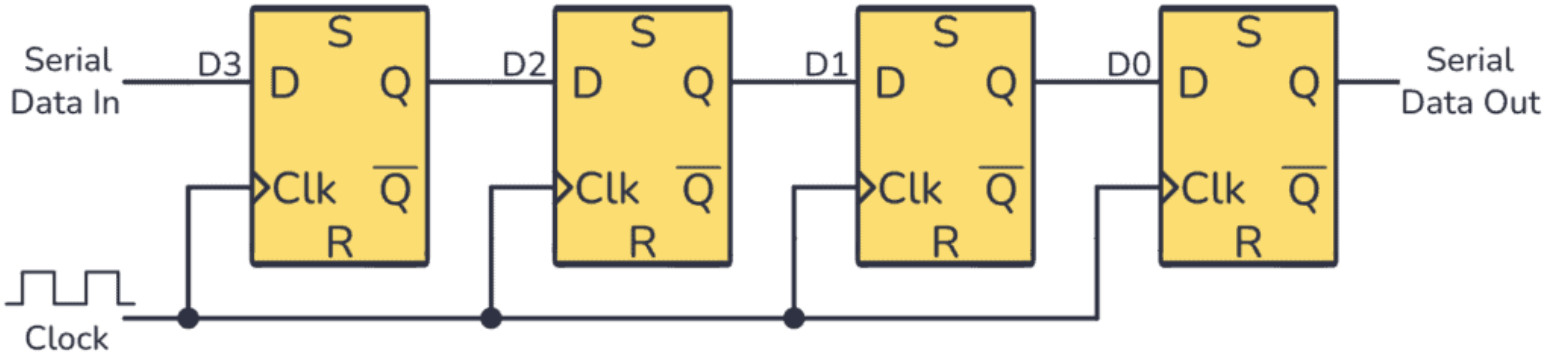
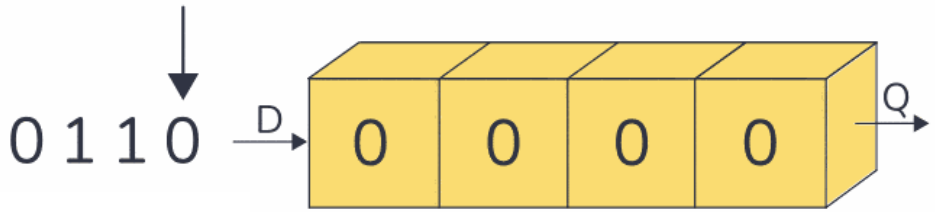


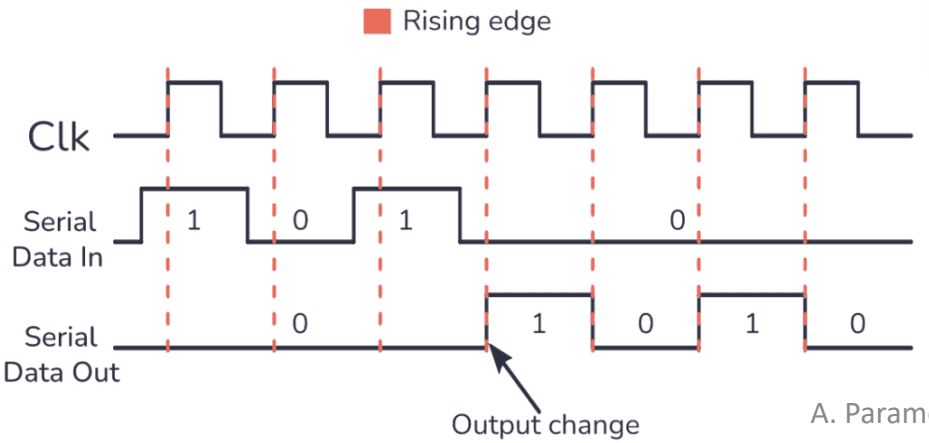
Figure 7.2 Definition of *set-up time*, *hold time*, and *propagation delay* of a synchronous register.

Shift register

- It can be used to delay a signal by a given number of clock cycles.



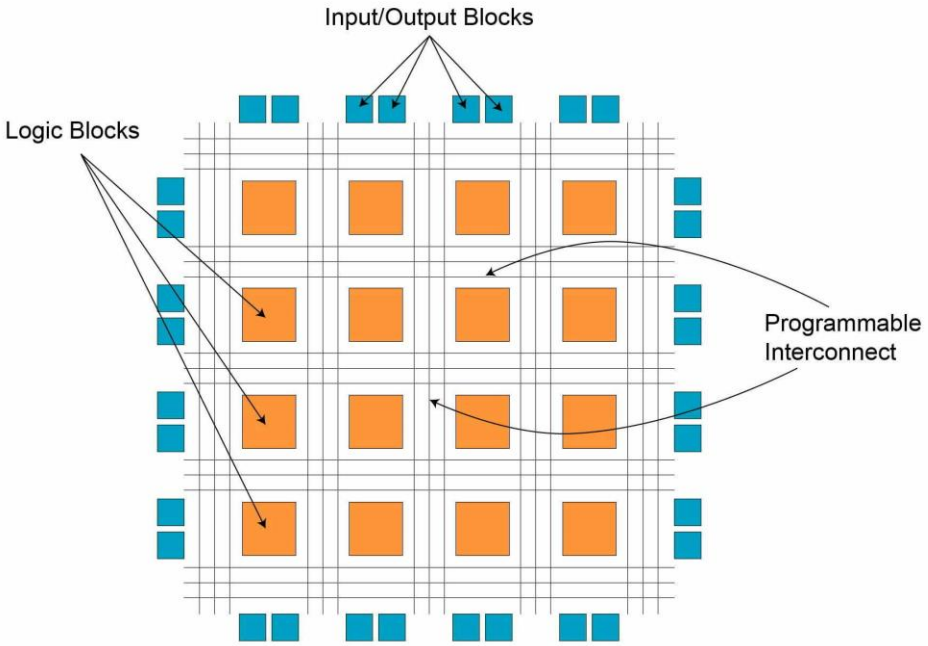


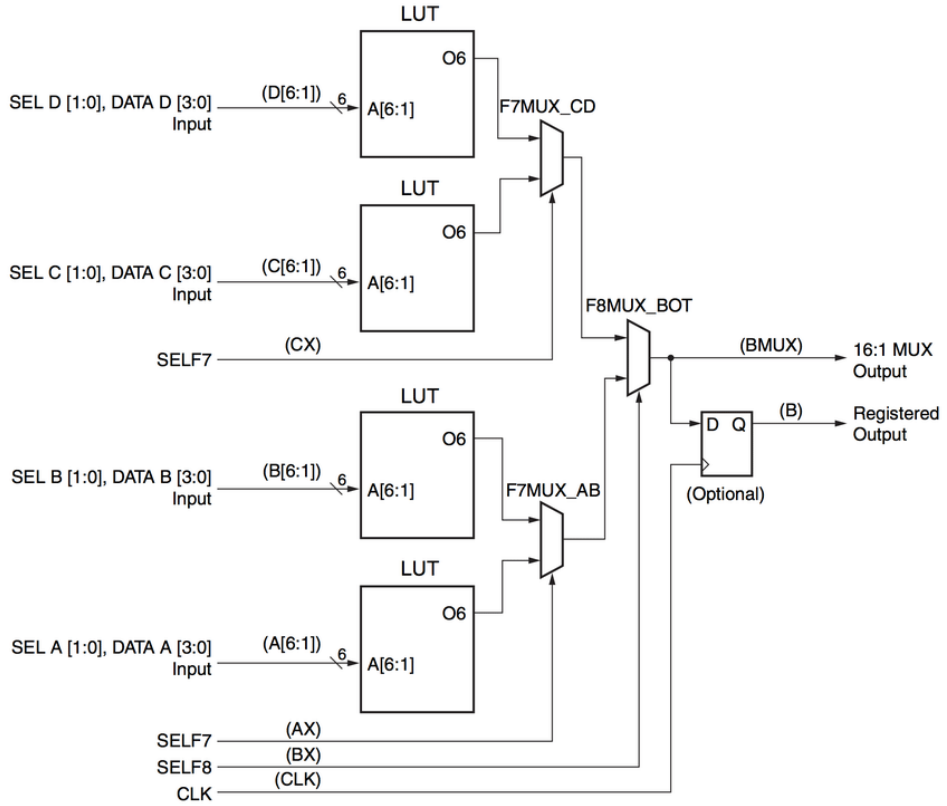


Exercise → Design a 4-bit synchronous counter

FPGA structure

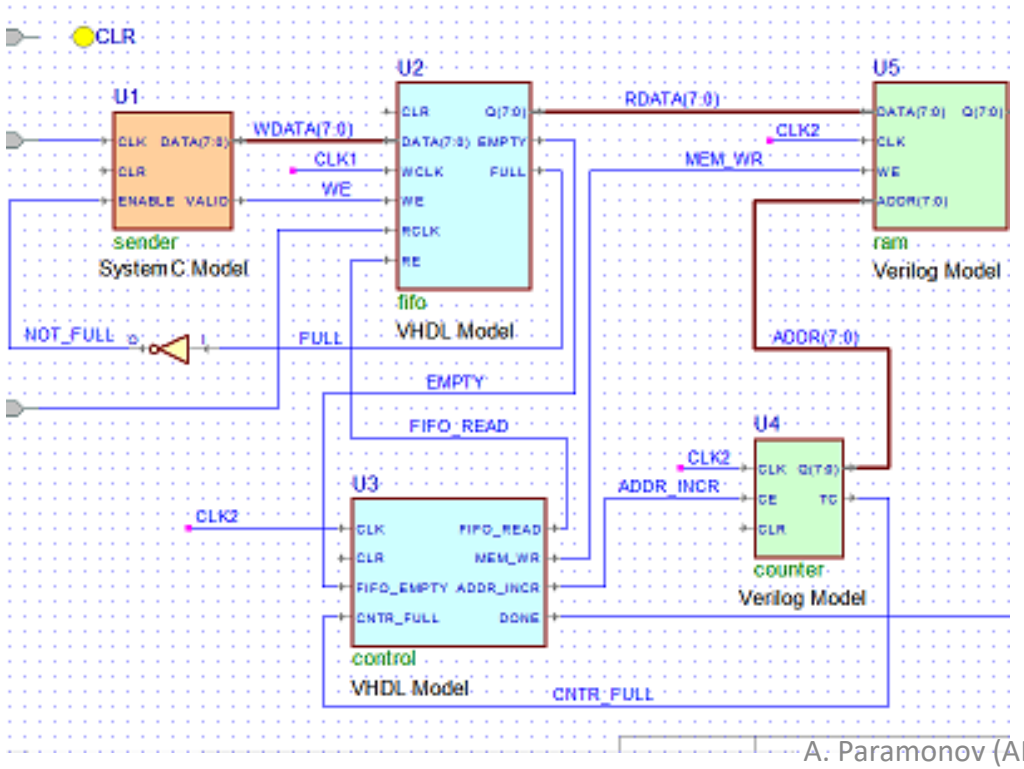
- An FPGA contains Configurable Logic Blocks (Xilinx) or Logic Cells (Altera)
- A CLB contains LUTs and DFFs. This allows flexible implementations of combinatorial and sequential circuits.





Approaches to digital circuit design

- Block entry (schematics)
- Text entry (Verilog, VHDL, etc). Verilog and VHDL are Hardware Description Languages (HDLs).
- Approaches like HLS allow to generate Verilog and VHDL from C-like code




```vhdl
-- (this is a VHDL comment)
/*
    this is a block comment (VHDL-2008)
*/
-- import std_logic from the IEEE library
library IEEE;
use IEEE.std_logic_1164.all;

-- this is the entity
entity ANDGATE is
  port (
    I1 : in std_logic;
    I2 : in std_logic;
    O  : out std_logic);
end entity ANDGATE;

-- this is the architecture
architecture RTL of ANDGATE is
begin
  O <= I1 and I2;
end architecture RTL;
```

Basics of VHDL

- “VHDL (VHSIC Hardware Description Language) is a hardware description language that can model the behavior and structure of digital systems at multiple levels of abstraction, ranging from the system level down to that of logic gates, for design entry, documentation, and verification purposes.”

- VHDL allows to describe the behavior of the required circuit and to verify/simulate it.
- Synthesis tools can translate the design into a IC (transistors and wires or gates and wires) or a bit file to program an FPGA.
- “VHDL is a dataflow language in which every statement is considered for execution simultaneously, unlike procedural computing languages such as BASIC, C, and assembly code, where a sequence of statements is run sequentially one instruction at a time. “

VHSIC = Very High Speed Integrated Circuit

Example VHLD code

- “Register-transfer level (RTL) is a design abstraction which models a synchronous digital circuit in terms of the flow of digital signals (data) between hardware registers, and the logical operations performed on those signals.”
- The code includes interface and actual implementation.
- The input and output ports are discussed in the “entity” block.
- Behavior description of the circuit is given in the “architecture” block.

```
-- (this is a VHDL comment)
/*
    this is a block comment (VHDL-2008)
*/
-- import std_logic from the IEEE library
library IEEE;
use IEEE.std_logic_1164.all;

-- this is the entity
entity ANDGATE is
  port (
    I1 : in std_logic;
    I2 : in std_logic;
    O  : out std_logic);
end entity ANDGATE;

-- this is the architecture
architecture RTL of ANDGATE is
begin
  O <= I1 and I2;
end architecture RTL;
```

IEEE 1164

- The IEEE 1164 standard describes the definitions of logic values.
- The library also defines Boolean logic operations.

Character	Value
'U'	uninitialized
'X'	strong drive, unknown logic value
'0'	strong drive, logic zero
'1'	strong drive, logic one
'Z'	high impedance
'W'	weak drive, unknown logic value
'L'	weak drive, logic zero
'H'	weak drive, logic one
'-'	don't care

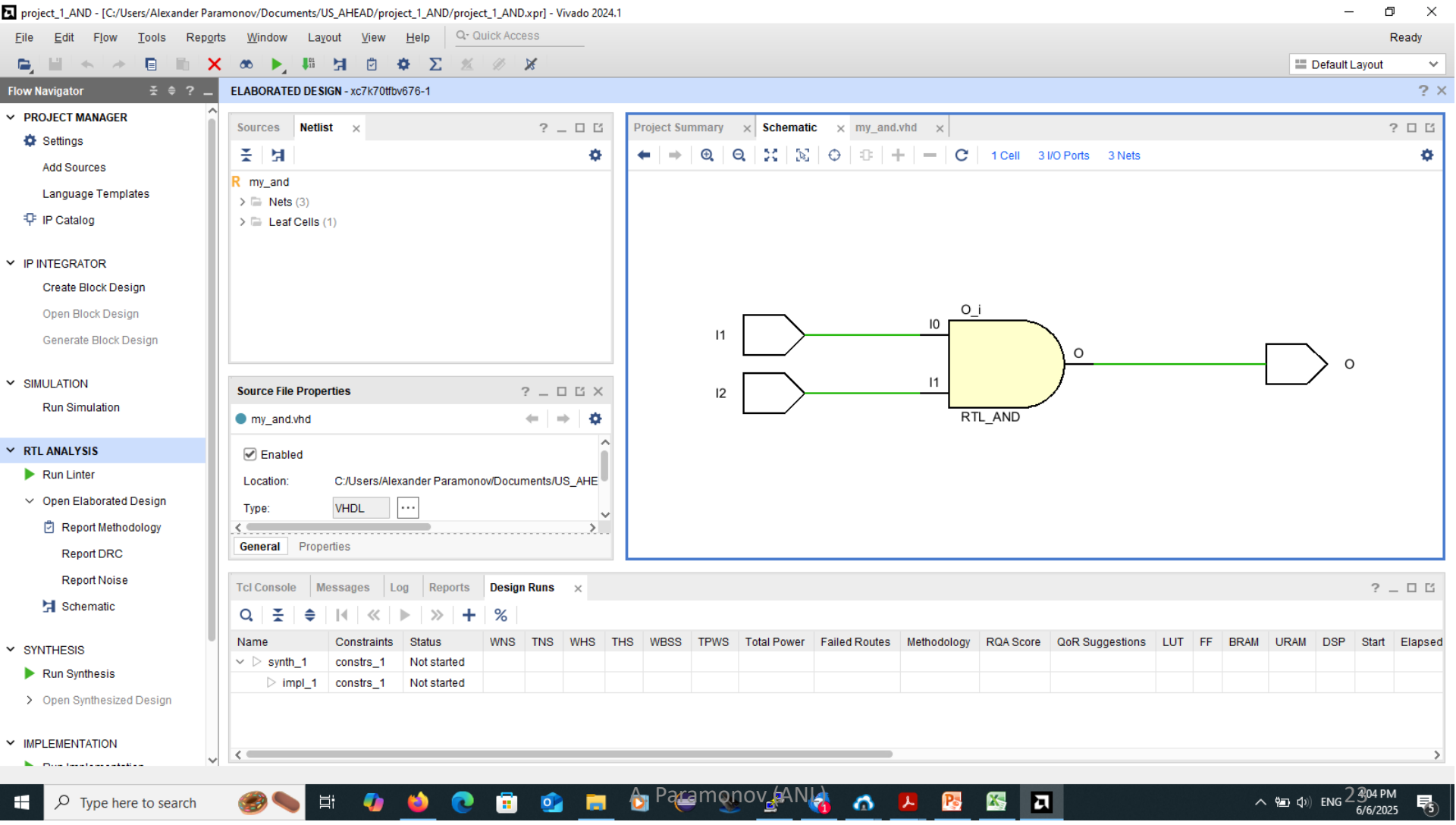
```
-- (this is a VHDL comment)
/*
    this is a block comment (VHDL-2008)
*/
-- import std_logic from the IEEE library
library IEEE;
use IEEE.std_logic_1164.all;

-- this is the entity
entity ANDGATE is
  port (
    I1 : in std_logic;
    I2 : in std_logic;
    O  : out std_logic);
end entity ANDGATE;

-- this is the architecture
architecture RTL of ANDGATE is
begin
  O <= I1 and I2;
end architecture RTL;
```

Elaborated design

- Synthesis turns the VHDL text into a circuit.

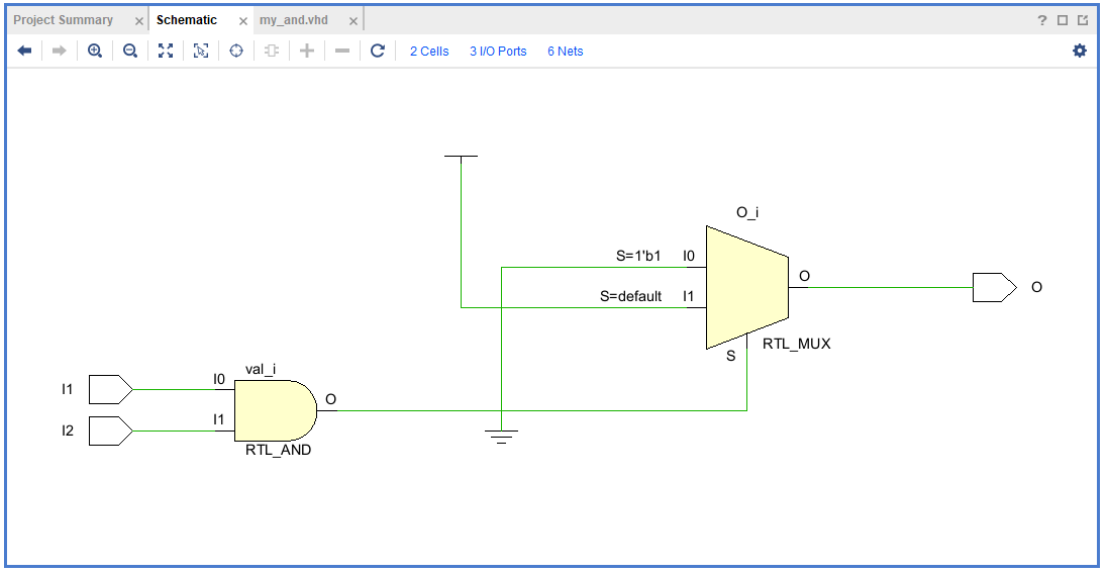


when else

- In the previous example we used so called “concurrent signal assignment”: O <= I1 AND I2;
- There are other concurrent code structures we can use such as “when … else”.
- Order of the assignments does not matter. It will be the same circuit.
- This is “dataflow style”

```
entity my_and is
    Port ( I1 : in STD_LOGIC;
           I2 : in STD_LOGIC;
           O : out STD_LOGIC);
end my_and;

architecture Behavioral of my_and is
  signal val : std_logic;
begin
  val <= I1 and I2;
  O <= '0' when val = '1' else '1';
end Behavioral;
```



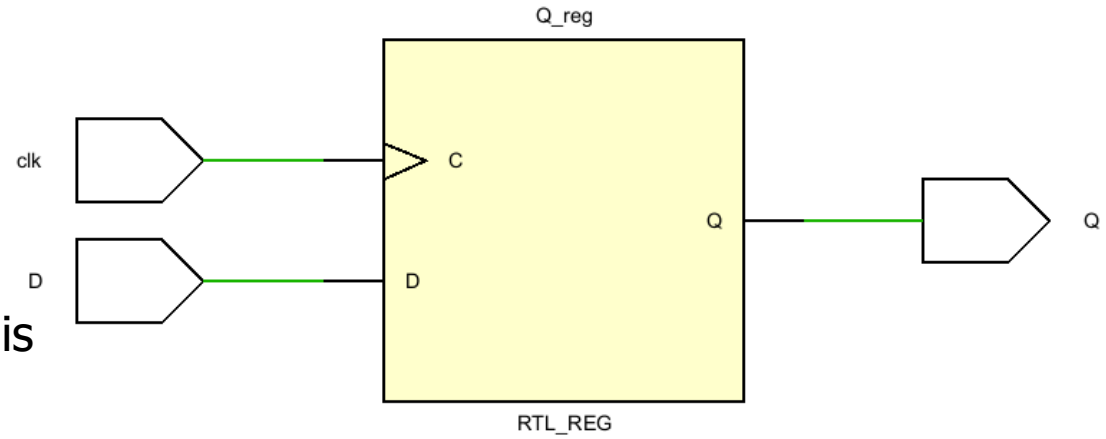
DFF

- Here is how we can describe a D Flip-flop with the data flow style.

```
entity my_DFF is
    Port ( D : in STD_LOGIC;
           clk : in STD_LOGIC;
           Q : out STD_LOGIC);
end my_DFF;

architecture Behavioral of my_DFF is

begin
  Q <= D when rising_edge(clk);
end Behavioral;
```



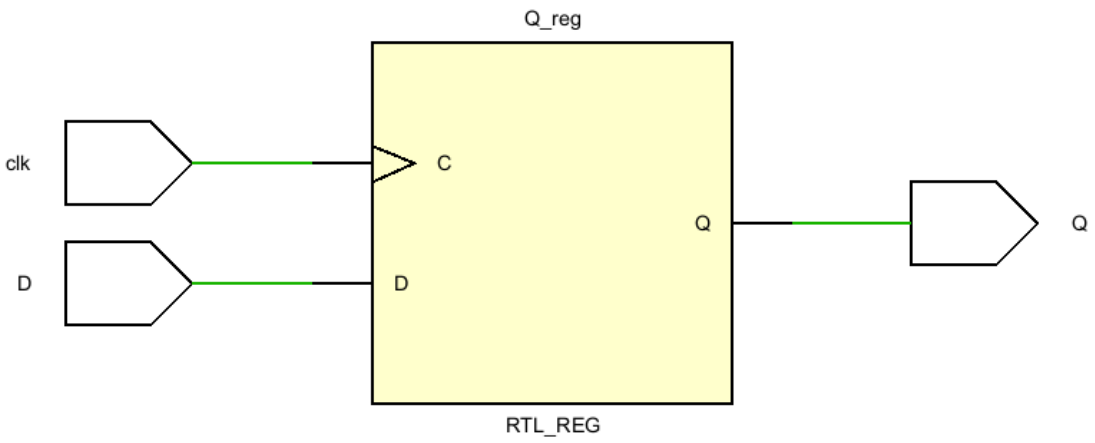
DFF

- Here is a DFF coded behavioral style
- Assignments inside “process” statement is interpreted sequentially.

```
entity my_DFF is
    Port ( D : in STD_LOGIC;
           clk : in STD_LOGIC;
           Q : out STD_LOGIC);
end my_DFF;

architecture Behavioral of my_DFF is

begin
  process(clk)
  begin
    if( rising_edge(clk)) then
       Q <= D;
    end if;
  end process;
end Behavioral;
```

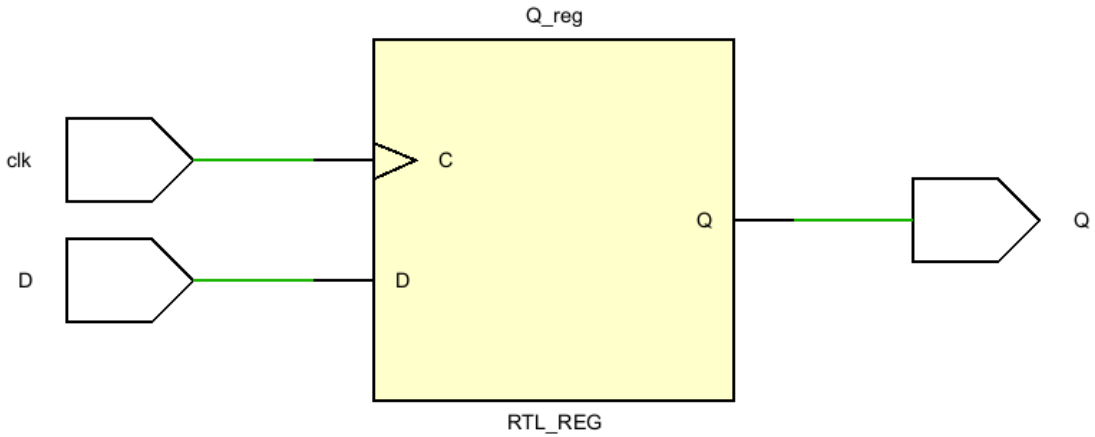


Is this a DFF?

```
process(clk)
 begin
   if( rising_edge(clk)) then
       Q <= '0';
       Q <= D;
   end if;
 end process;
```

Is this a DFF?

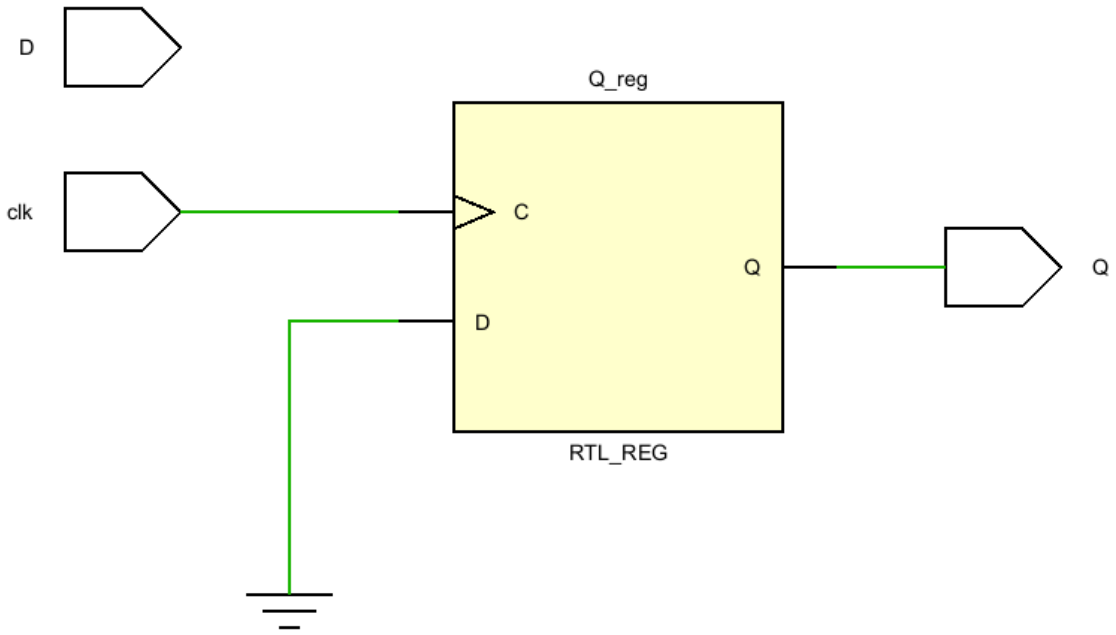
Yes!



Is this a DFF?

```
process(clk)
 begin
   if( rising_edge(clk)) then
       Q <= D;
       Q <= '0';
   end if;
 end process;
```

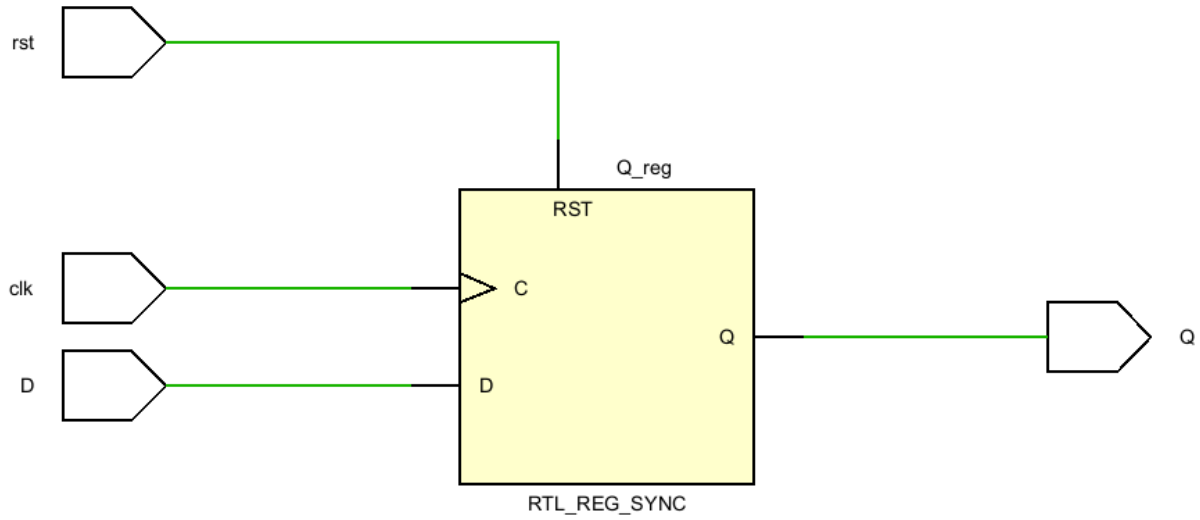
Is this a DFF?



DFF with a synchronous reset

- The reset is active only during the clock rising edge.
- The output Q changes synchronously with the clock signal.

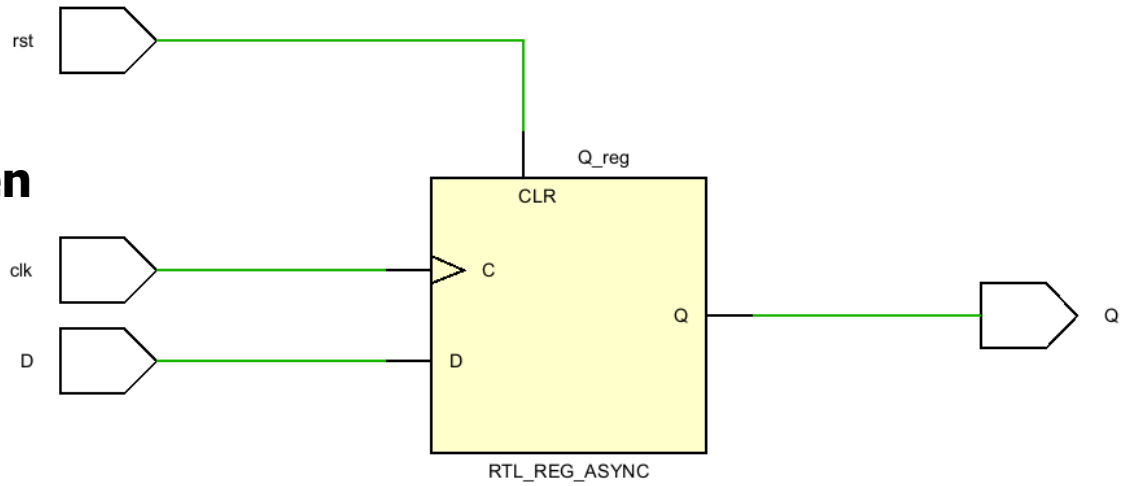
```
process(clk)
 begin
   if( rising_edge(clk)) then
      if (rst = '1') then
         Q <= '0';
      else
         Q <= D;
      end if;
   end if;
 end process;
```



DFF with an asynchronous reset

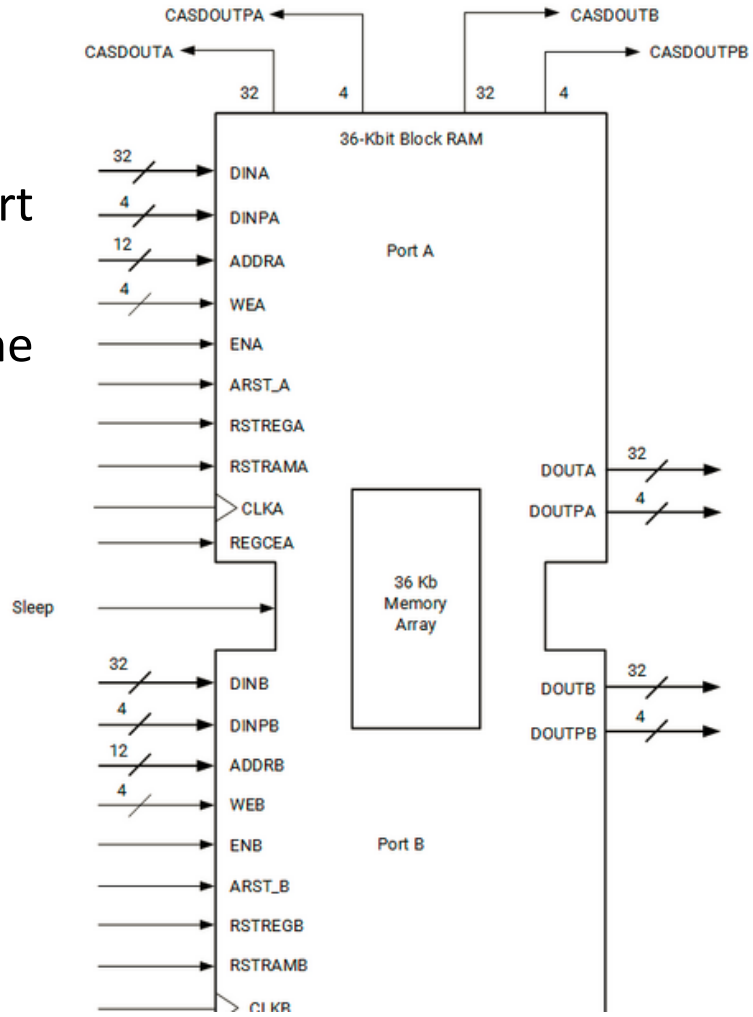
- The reset is active all the time, not only during the clock rising edge.
- The output Q can change any time.
- This makes the timing analysis of the circuit hard. Timing analysis is needed to verify the signal integrity of the circuit.
- Do not use asynchronous resets unless you absolutely need them.

```
process(clk, rst)
 begin
      if (rst = '1') then
         Q <= '0';
      elsif(rising_edge(clk)) then
         Q <= D;
      end if;
 end process;
```



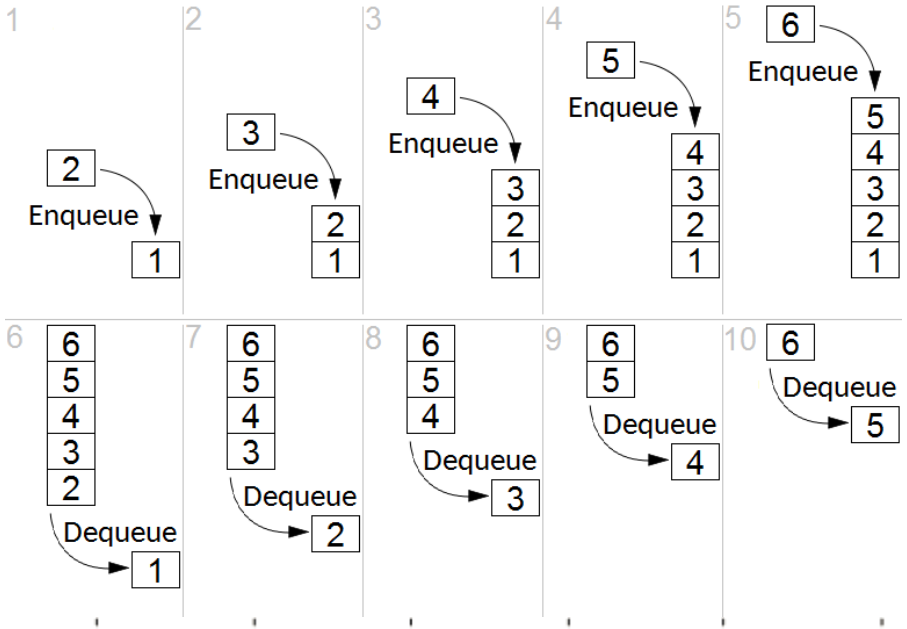
Block Random Access Memory (RAM)

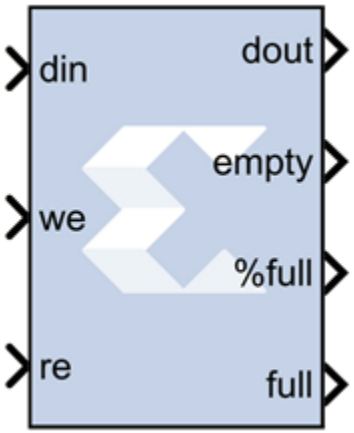
- A random access memory is basically a table.
- A RAM needs to have an input address port and an output data port.
- Data stored in a given address shown in the data port when the address bus is set to the desired value.
- Modern FPGAs have build in RAMs; so called Block RAM (BRAM) and Ultra RAM (URAM).
- The RAMs are clocked. There is a delay to access the data.
- There are read and write ports.
- Dual-port means that the same memory can be access (read or written) from two ports independently.

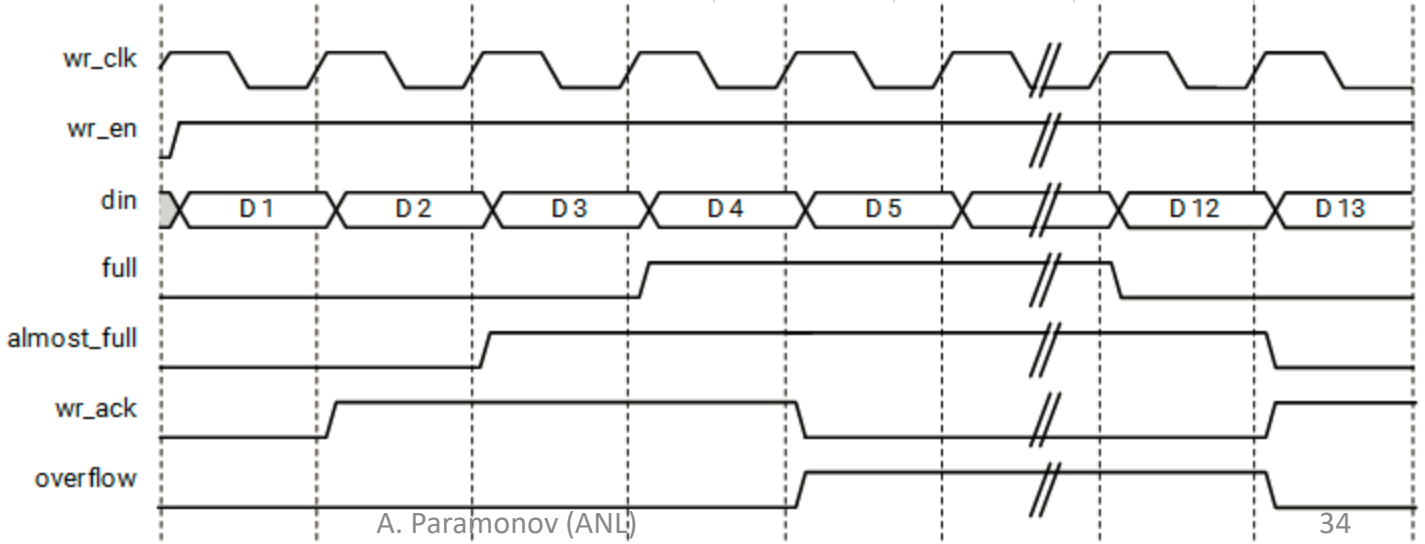


First In, First Out (FIFO)

- FIFO is a memory. It is not random access.
- It is used to pass data between circuits that may be asynchronous.
- There are FWFT (first word fall through) and “standard” FIFOs.
- Their read operations are different.



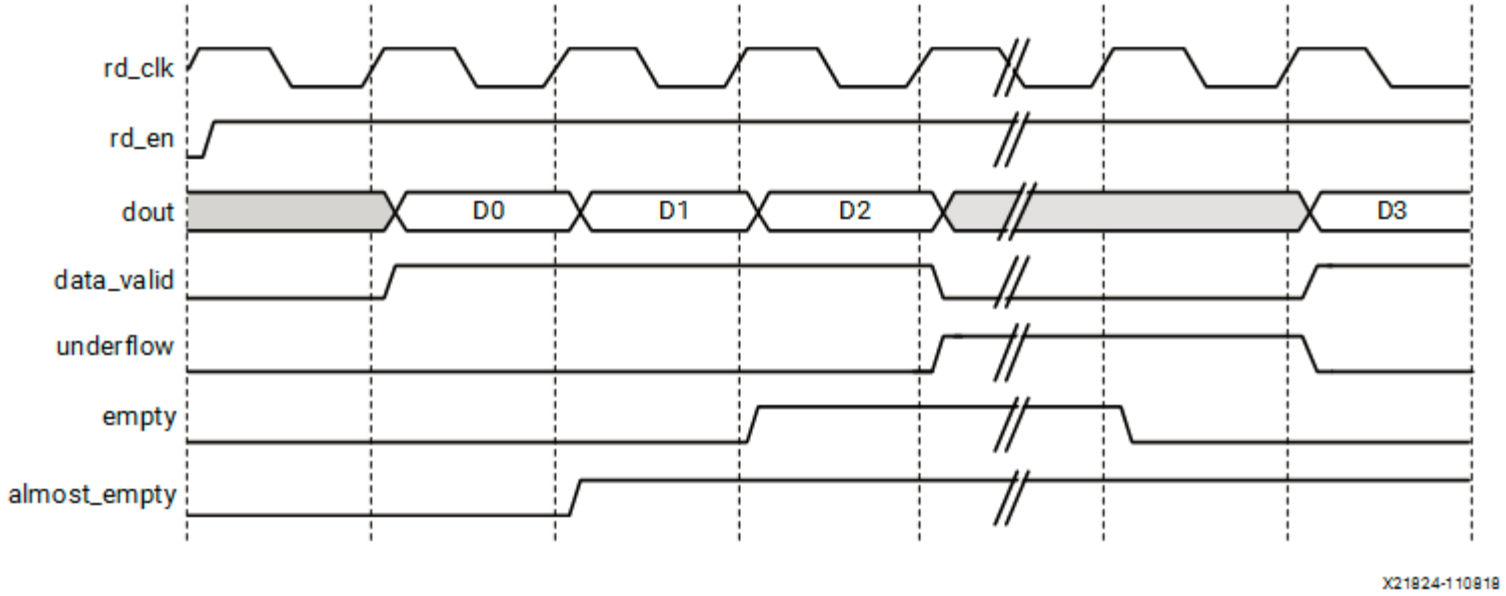




X21827-111318

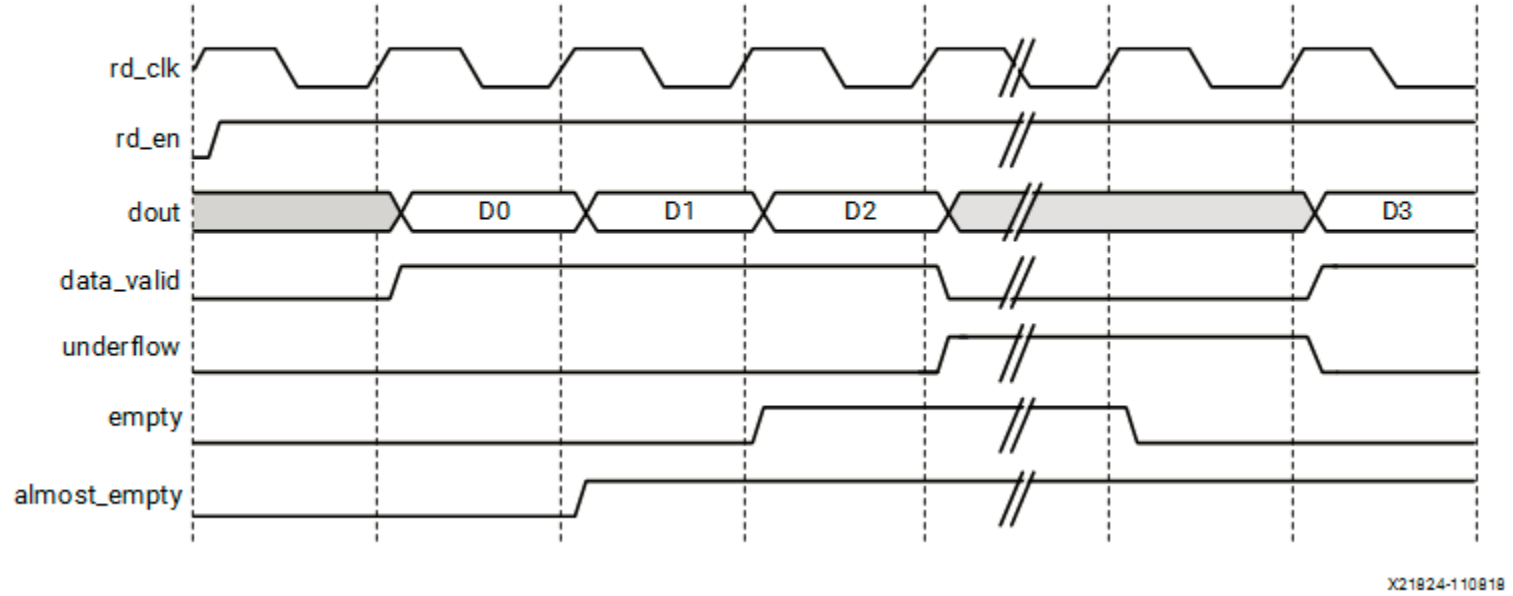
First In, First Out (FIFO)

Figure: Standard Read Operation for a FIFO with Independent Clocks



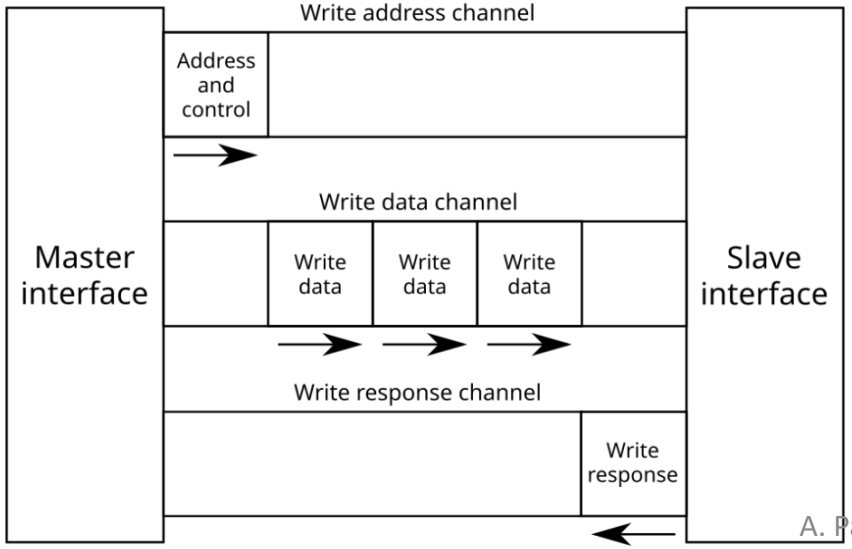
First In, First Out (FIFO)

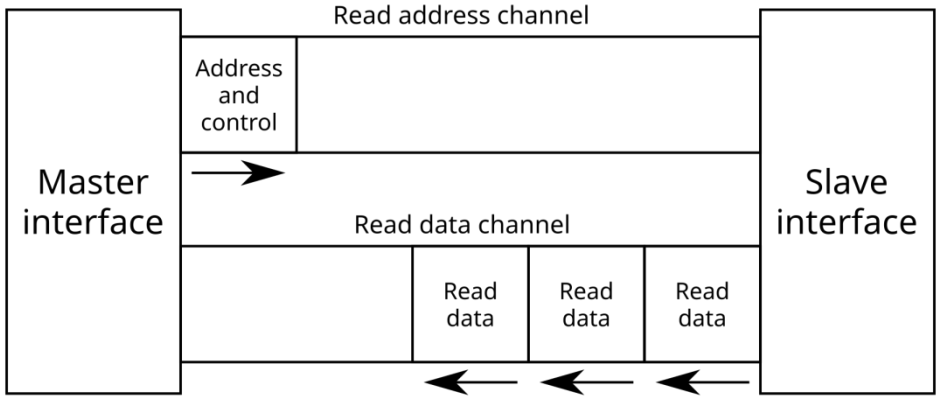
Figure: Standard Read Operation for a FIFO with Independent Clocks



Advanced eXtensible Interface (AXI) Bus

- On-chip communication bus protocol.
- Two circuits can pass data between each other (in one direction or two directions).
- **AXI4-Stream** is a simplified, lightweight bus protocol designed specifically for high-speed streaming data applications



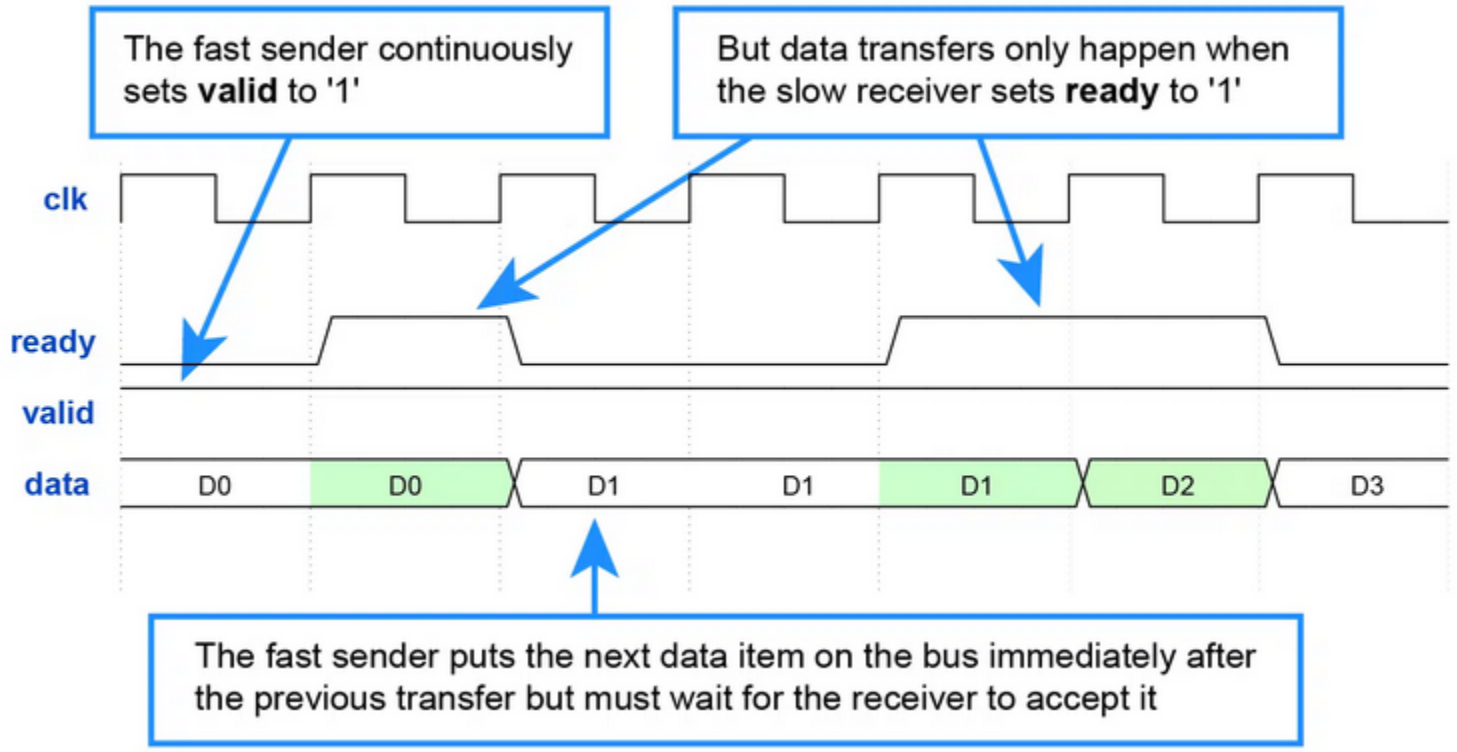


AXI4 Signals

- This is a subset of signals. Basically, it is very similar to writing into a FIFO

Signal	Source	
ACLK	Clock	ACLK is a global clock signal. All signals are sampled on the rising edge of ACLK.
TVALID	Transmitter	TVALID indicates the Transmitter is driving a valid transfer. A transfer takes place when both TVALID and TREADY are asserted.
TDATA	Transmitter	TDATA is the primary payload used to provide the data that is passing across the interface.
TLAST	Transmitter	TLAST indicates the boundary of a packet.
TREADY	Receiver	TREADY indicates that a Receiver can accept a transfer.

AXI4 waveform

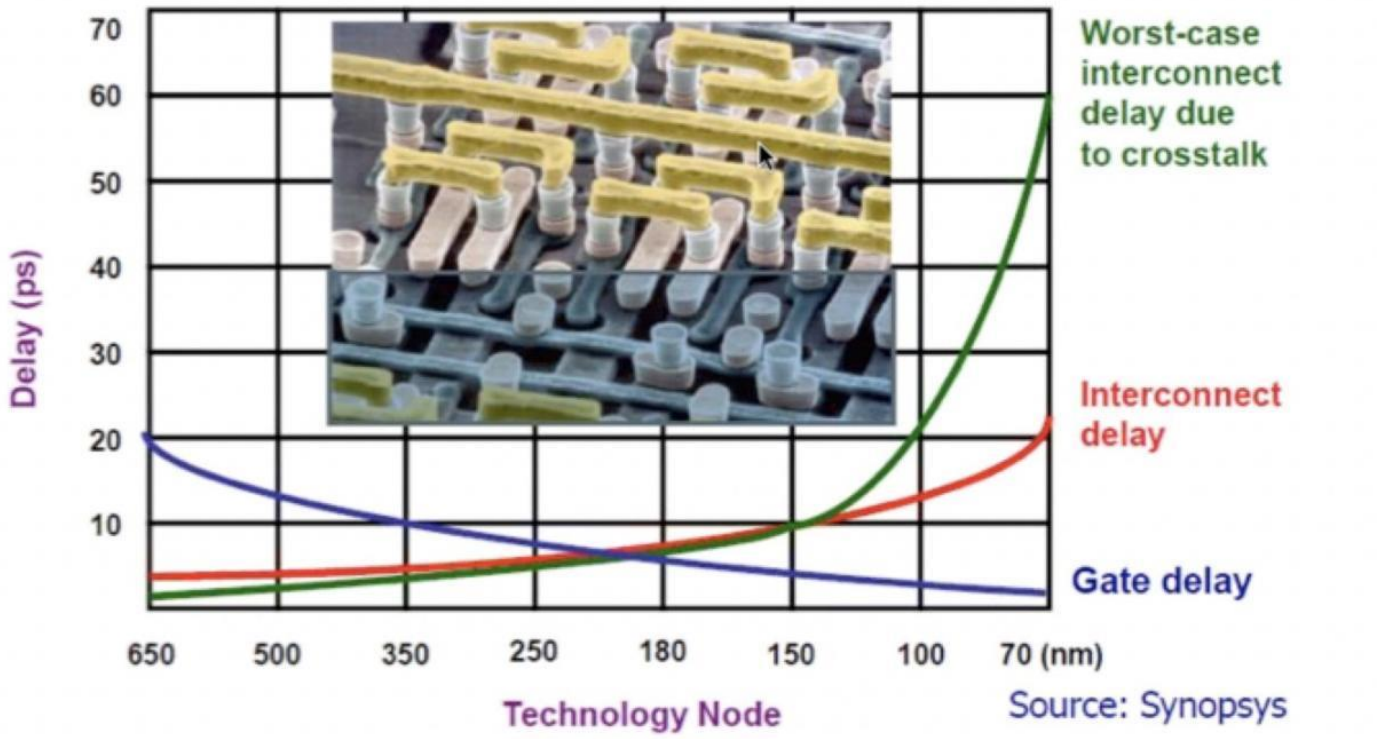


Parasitics

- What limits the clock rate of digital circuits?
- The clock rate of CPUs is in GHz range. For FPGAs it is hard to go beyond 100’s of MHz.

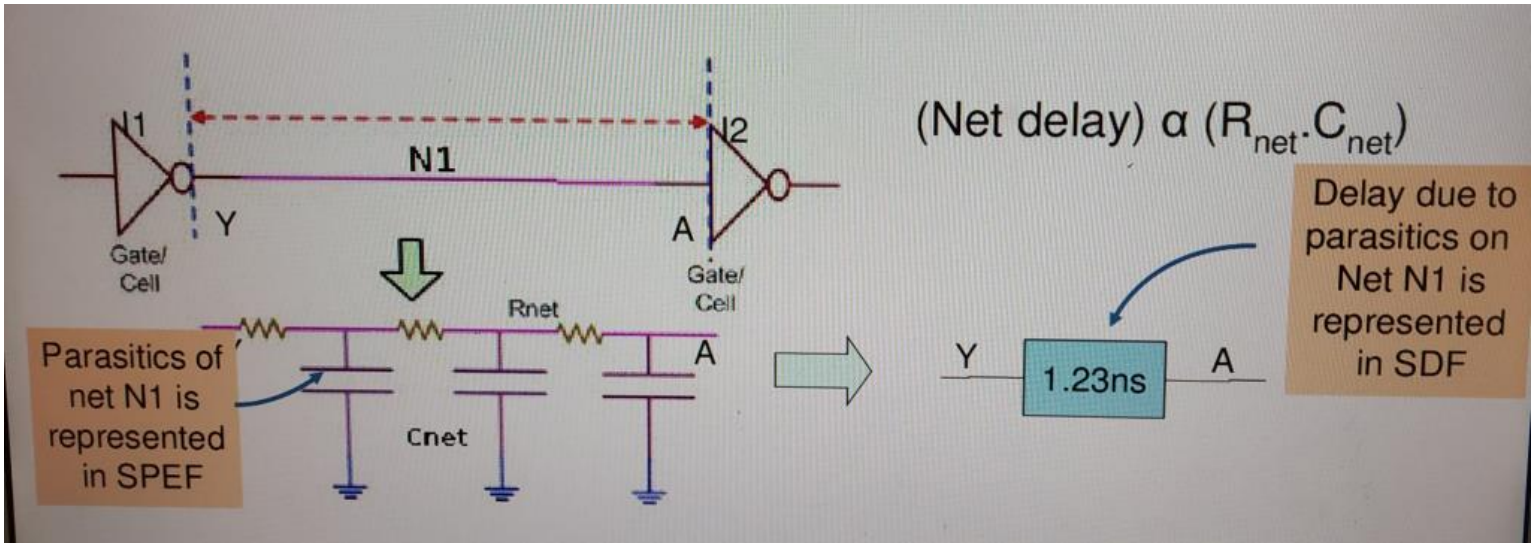
Parasitics

- High resistance of interconnects (wires) limits performance of Ics.
- In FPGAs the interconnect layer has higher parasitics than regular “wires”



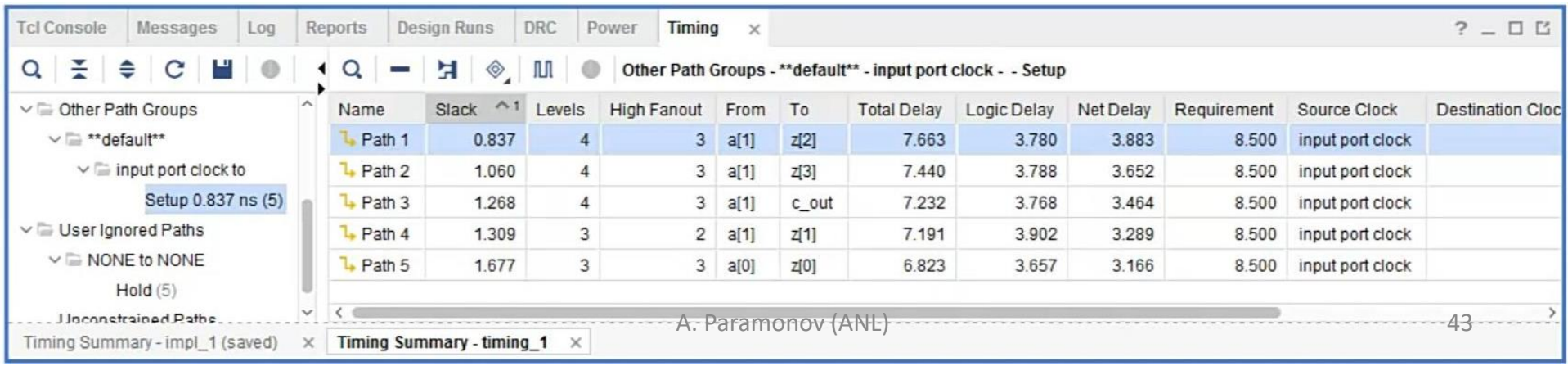
Parasitics

- T = R*C



FPGA parasitics

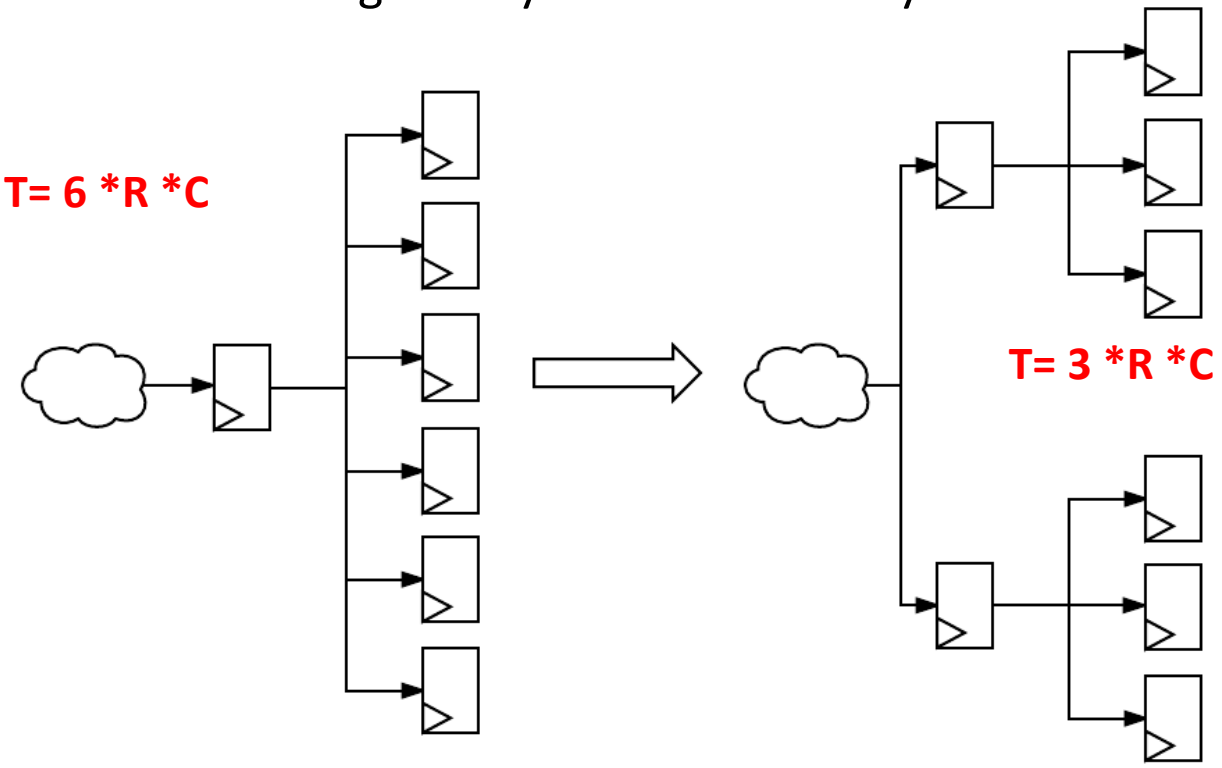
- HLS/HDL first goes through synthesis to become a circuit.
- Then it goes through “place and route” when a synthesized circuit is mapped to an FPGAs (CLBs are connected).
- Place and Route tool also analyzed performance of the circuit and it gives you a timing report.
- The timing report tells is you if the firmware can operate at the desired clock frequency.
- “Fanout” (how many sources are driven).
- “Levels” is how many look up tables are chained between flip-flops.



Name	Slack	Levels	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source Clock	Destination Cloc
Path 1	0.837	4	3	a[1]	z[2]	7.663	3.780	3.883	8.500	input port clock	
Path 2	1.060	4	3	a[1]	z[3]	7.440	3.788	3.652	8.500	input port clock	
Path 3	1.268	4	3	a[1]	c_out	7.232	3.768	3.464	8.500	input port clock	
Path 4	1.309	3	2	a[1]	z[1]	7.191	3.902	3.289	8.500	input port clock	
Path 5	1.677	3	3	a[0]	z[0]	6.823	3.657	3.166	8.500	input port clock	

Fanout

- Clock is not shown.
- Here is an example how the place and route tool may optimize a circuit.
- Sometimes the tool is not smart enough and you need to do it yourself.



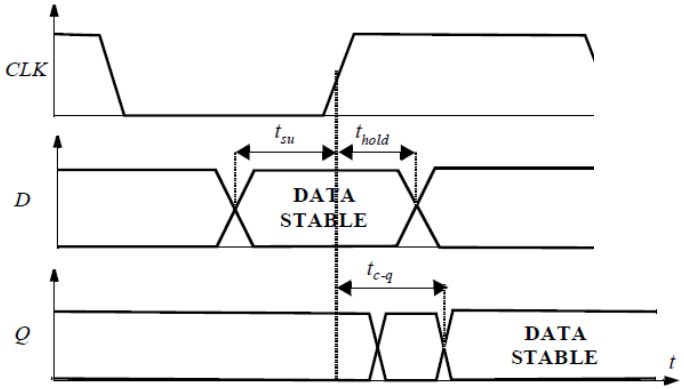
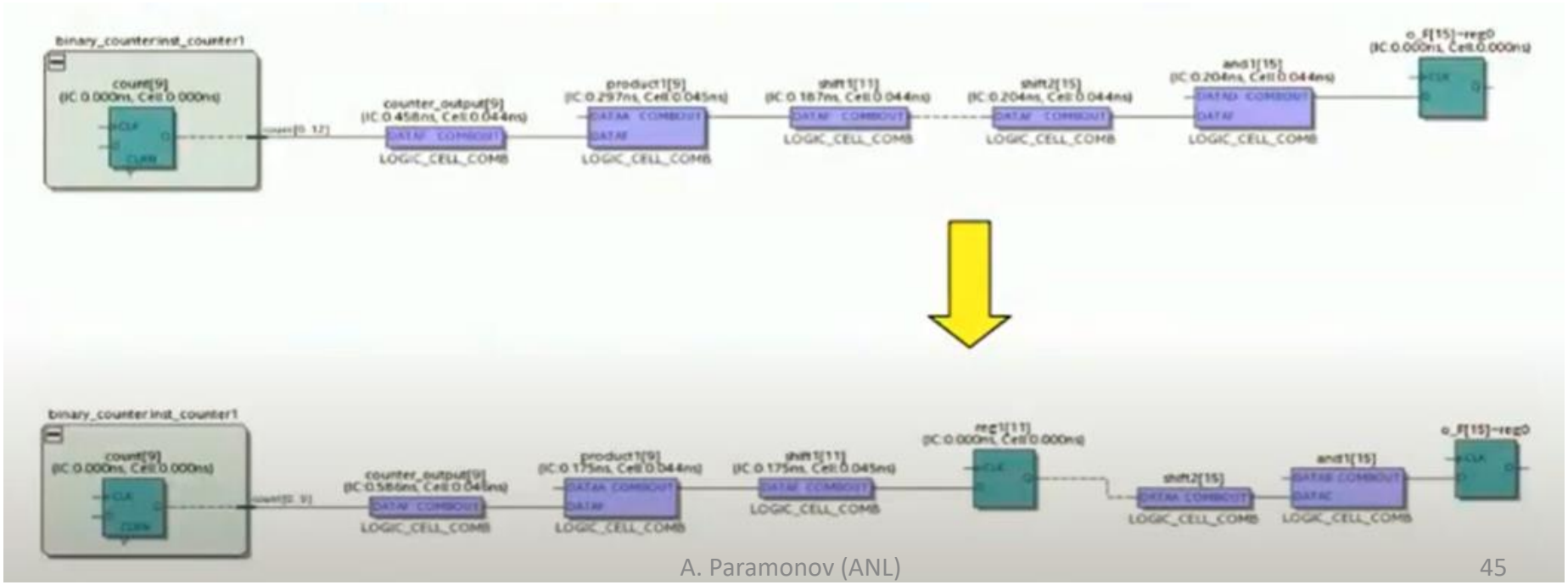


Figure 7.2 Definition of *set-up time*, *hold time*, and *propagation delay* of a synchronous register.

Levels

- Each look up table adds a delay.
- The cumulative delay of the entire operation needs to be smaller than period of the clock.
- Putting a register in the middle of the calculation results in two quicker calculations.
- Place and route tool does not do it for you.



Outlook

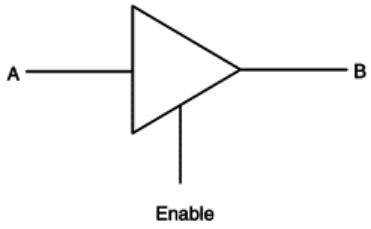
- Digital electronics is a foundation of the modern computing and data processing.
- It is not as hard as Jahred described it. You may grow to love it.
- This presentation highlighted a few vital concepts that you may find useful later as you progress with HLS.

BACKUP

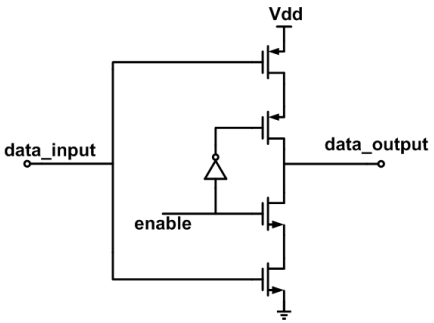
More on Std_logic states

- Sometimes an output driver/buffer may need to be pulled-up weakly (with a resistor) or disabled (set to high impedance).

The output buffer can be set to high-impedance state.



Enable	A	B
0	0	Z
0	1	Z
1	0	0
1	1	1



Here we can use a common buss with a weak pull-up.