

Complex Programmable Logic Device (CPLD)

Objectives

- Digital Design through Programmable Logic Devices in Verilog
- Frequency Divider (ex. tone generator for an audio device)

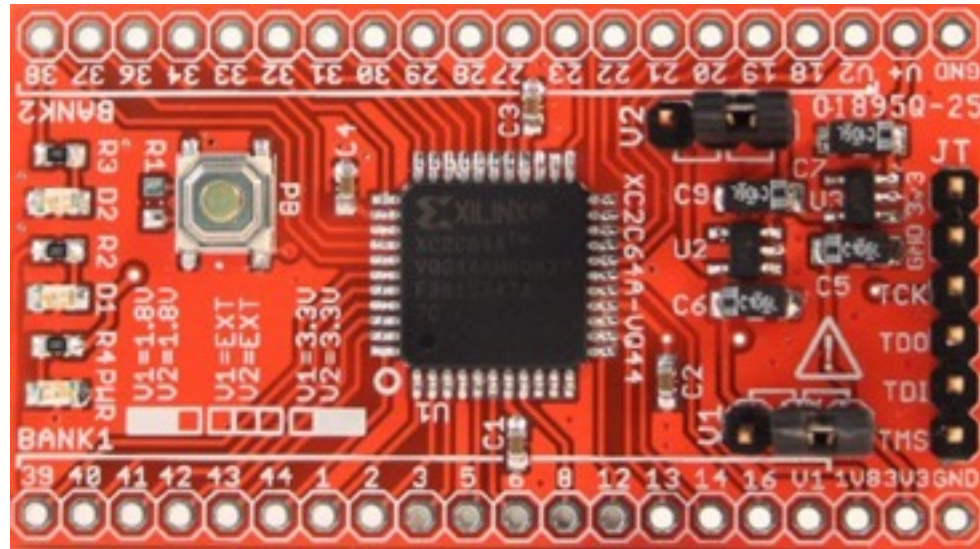


Table of Contents

1. Introduction

- a. Boolean Functions
- b. Logic Families

2. Sequential logic

- a. Registers
- b. Counters

3. CPLD features

- a. CoolRunner II Family
- b. Verilog Language

1. Introduction

Boolean functions and Logic families

Boolean functions: Gates

2-input OR gate



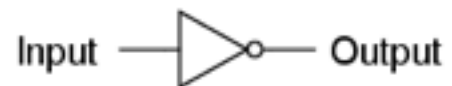
A	B	Output
0	0	0
0	1	1
1	0	1
1	1	1

2-input AND gate



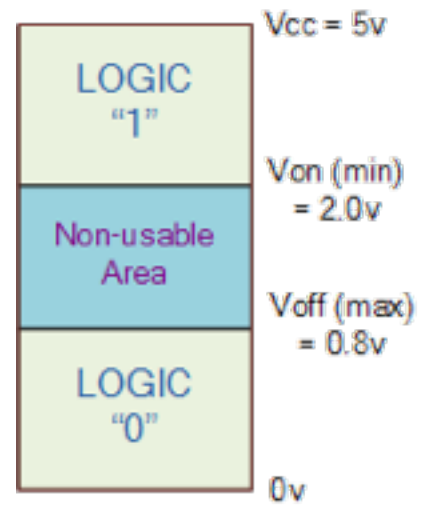
A	B	Output
0	0	0
0	1	0
1	0	0
1	1	1

Inverter, or NOT gate

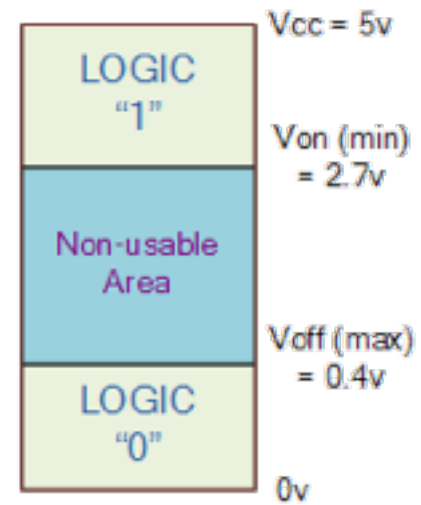


Logic Families

- TTL, CMOS, etc.
- Logic levels: MULTIMETER



LS - TTL Input
Voltage levels



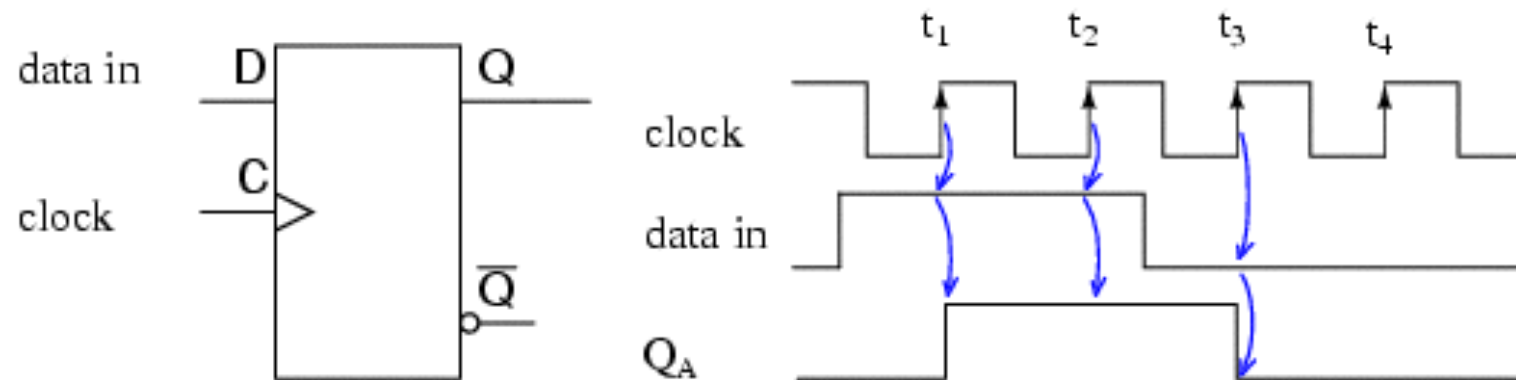
LS - TTL Output
Voltage levels

2. Sequential Logic

Registers and Counters

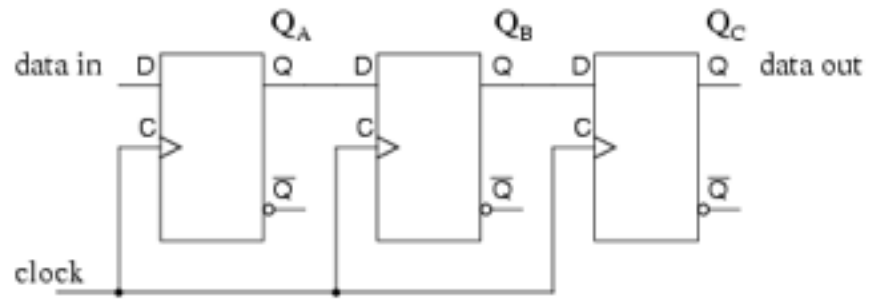
Registers

- Apart from the input, the output depends on the state (memory)
- Receiving data is not synchronized to the register clock



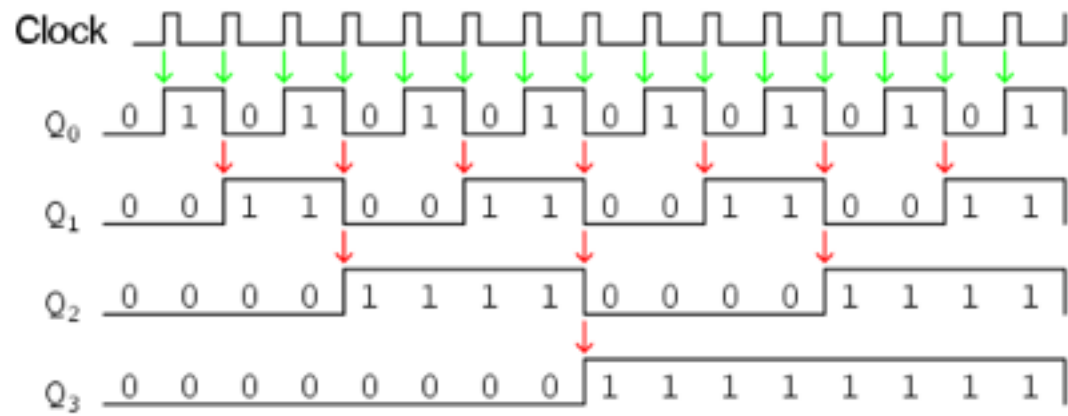
Data present at clock time is transferred from D to Q.

Shift register



Serial-in, serial-out shift register using type "D" storage elements

Binary counter



3. CPLD Features

CoolRunner II family and the Verilog language

CoolRunner II CPLD breakout board

- http://dangerousprototypes.com/docs/CoolRunner-II_CPLD_breakout_board
- CPLDs can give you the logic you need, with the pinout you want, while saving board space and board revisions.
- The platform has two separate banks of pins that can operate at different voltages (1V8 to 3V3) with internal pull-up resistors
- Disadvantages:
 - No banks at 5V (although the +V power supply)
 - ISE WebPack and Xilinx USB Cable (JTAG) is required

Verilog test-bench

The screenshot displays the Harmony IDE interface for a Verilog testbench. The top window shows the Verilog code for a testbench module named `testbench_d`. The code includes a timescale of 10ps/1ps, a module definition with registers for `clk` and `data`, and a wire for `out`. The initial conditions set `clk=0` and `data=0`, with a simulation time of 10000 ns. The testbench includes two always blocks: one for toggling the clock every 100 ns and another for toggling the data every 333 ns. The DFF0 component is instantiated with `out`, `data`, and `clk` signals.

The bottom window shows a timing diagram for the testbench. The diagram displays the signals `data`, `clk`, and `out` over time. The signals are sampled at 10ps intervals. The diagram shows the clock signal toggling every 100 ns, and the data signal toggling every 333 ns. The output signal `out` is shown to be a delayed version of the data signal.

Name	Scope	Value
New ...		
data	testbench_d	St1
clk	testbench_d	St1
out	testbench_d	St1

Item is a wire

5.261ns 17.936ns Tdelta 12.675ns Time 100.000ns Harmony 4.10.91 © SILVACO 2010

Verilog designs

```
// Flip flop register
```

```
reg out;
```

```
always@(posedge clk)
```

```
begin
```

```
    out <= data;
```

```
end
```

```
// Binary counter
```

```
reg [3:0] counter;
```

```
always@(posedge clk)
```

```
begin
```

```
    counter <= counter + 1;
```

```
end
```

Questions

juangagobenitez@gmail.com