

Digital Integrated Circuits and Systems

Andrej Trost

Lab 1: Digital circuit design for FPGA devices

Tool: Vivado 2025.1

Task: design circuit models in SystemVerilog,
run synthesis and simulation

2026/27



Design tool: Vivado 2025

short path, no spaces
no special characters



1. Prepare project **folder**: c:/proj/myname
2. Run **Vivado**, select **Create Project >**
3. New Project, define **Project name** and **Project location**
4. Project Type: **RTL**, Next, Next, **Default Part**: search xc7z010clg400
or | Boards, Red Pitaya
5. Finish

Default Part
Choose a default AMD part or board for your project.

Parts | Boards

[Reset All Filters](#)

Category: Package: Temperature:
Family: Speed: Static power:

Search: (3 matches)

Part	I/O Pin Count	Available IOBs	LUT Elements	FlipFlops	Block RAMs
xc7z010clg400-3	400	100	17600	35200	60
xc7z010clg400-2	400	100	17600	35200	60
xc7z010clg400-1	400	100	17600	35200	60



Vivado

Main tasks

Active view

Status

Close

Console

Reports

ndes1 - [C:/xproj/ndes1/ndes1.xpr] - Vivado 2025.1

File Edit Flow Tools Reports Window Layout View Help Q- Quick Access

Flow Navigator

- PROJECT MANAGER
 - Settings
 - Add Sources
 - Language Templates
 - IP Catalog
- IP INTEGRATOR
 - Create Block Design
 - Open Block Design
 - Generate Block Design
- SIMULATION
 - Run Simulation
- RTL ANALYSIS
 - Run Linter
 - Open Elaborated Design
 - Report Methodology
 - Report DRC
 - Report Noise
 - Schematic
- SYNTHESIS
 - Run Synthesis

ELABORATED DESIGN - xc7z007sclg225-1

Sources x Netlist

- Design Sources (1)
 - regop (regop.sv)
- Constraints (1)
 - constrs_1 (1)
 - constr.xdc
- Simulation Sources (1)

Hierarchy Libraries Compile Order

Synthesis Run Properties

synth_1

Name: synth_1

Part: xc7z007sclg225-1 (active)

General Properties Options Log Reports Messages

Project Summary x Schematic x regop.sv x constr.xdc x

43 Cells 43 I/O Ports 85 Nets

reg_a_reg[13:0] RTL_REG

reg_b_reg[13:0] RTL_REG

reg_c_reg[13:0] RTL_REG

reg_c[13:0]

clk a[13:0] b[13:0]

c_i RTL_DIV

IO[13:0] I1[13:0] O[13:0]

Tcl Console Messages Log Reports Design Runs

Name	Constraints	Status	WNS	TNS	WHS	THS	WBSS	TPWS	Total Power	Failed Routes	Methodology	RQA Score	LUT	QoR Suggest
✓ synth_1	constrs_1	synth_design Complete!											209	
✓ impl_1	constrs_1	route_design Complete!	62.818	0.000	0.358	0.000		0.000	0.093	0	42 Warn		209	

Create Design Source

1. PROJECT MANAGER, **Add Sources**, Add or create design sources
2. Click button **Create File**, type: **SystemVerilog**, name: logic_gates, Finish
3. Define I/O Ports (optional)
4. OK
5. Open file under **Design Sources**
click > to expand, double click to open

no spaces, ...
SystemVerilog
naming convention



Define a module and specify I/O Ports to add to your source file.
For each port specified:
MSB and LSB values will be ignored unless its Bus column is checked.
Ports with blank names will not be written.

Module Definition

Module name:

I/O Port Definitions

Port Name	Direction	Bus	MSB	LSB
a, b	input	☐	0	0
y	output	☒	1	0

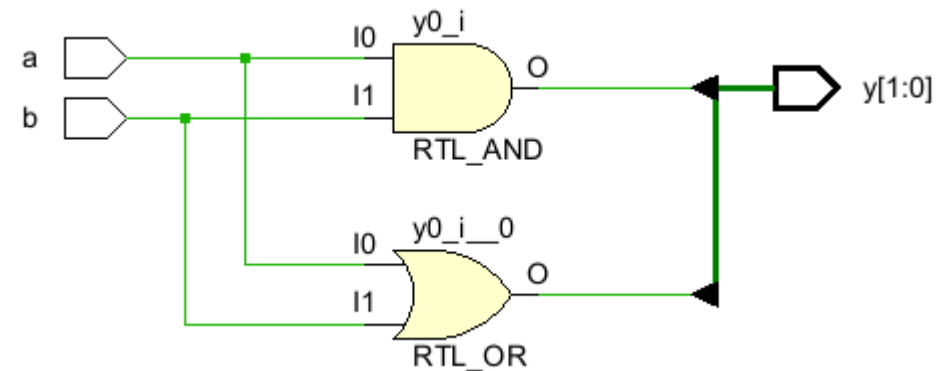
OK Cancel

Logic description & elaboration

Example: on 2-bit output y, bit 0 is result of AND gate, bit 1 is result of OR gate

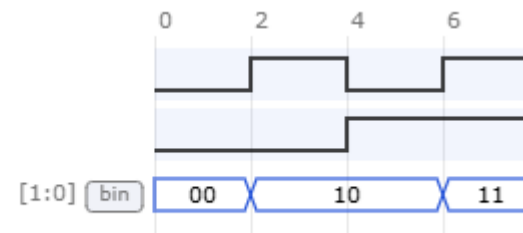
```
module logic_gates(  
    input a, b,  
    output [1:0] y  
);  
  
assign y[0] = a & b;  
y[1] = a | b;  
  
endmodule
```

← syntax error



1. RTL ANALYSIS, Open Elaborated Design
2. Simulation, set inputs and verify operation

```
1 module logic_gates(  
2     input a, b,  
3     output [1:0] y  
4 );  
5  
6 assign y[0] = a & b;  
7 assign y[1] = a | b;  
8  
9 endmodule
```

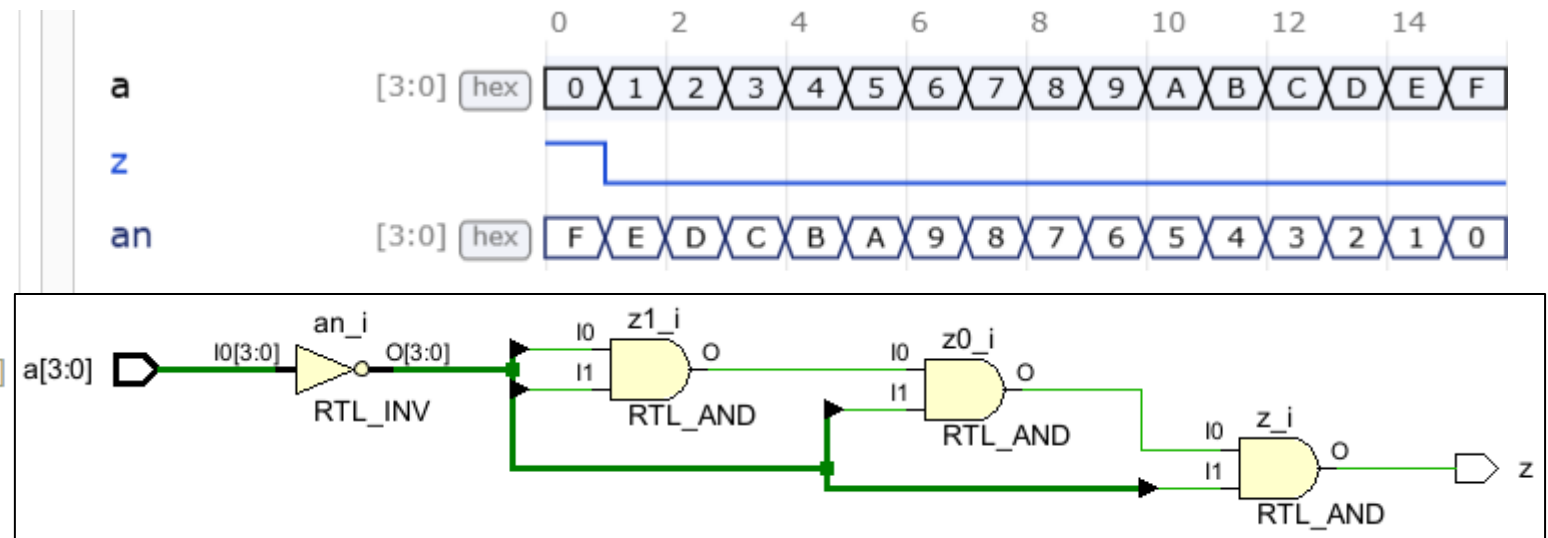


Logic comparators

- Task: describe digital comparator with logic gates, synthesize design, check circuit schematics and verify on simulation

Example: $z=1$ if all bits of input a are 0

```
module logic_cmp(  
    input [3:0] a,  
    output z  
);  
  
    logic [3:0] an;  
    logic b,c;  
  
    assign an = ~a;  
    assign z = an[0] & an[1] & an[2];  
  
endmodule
```

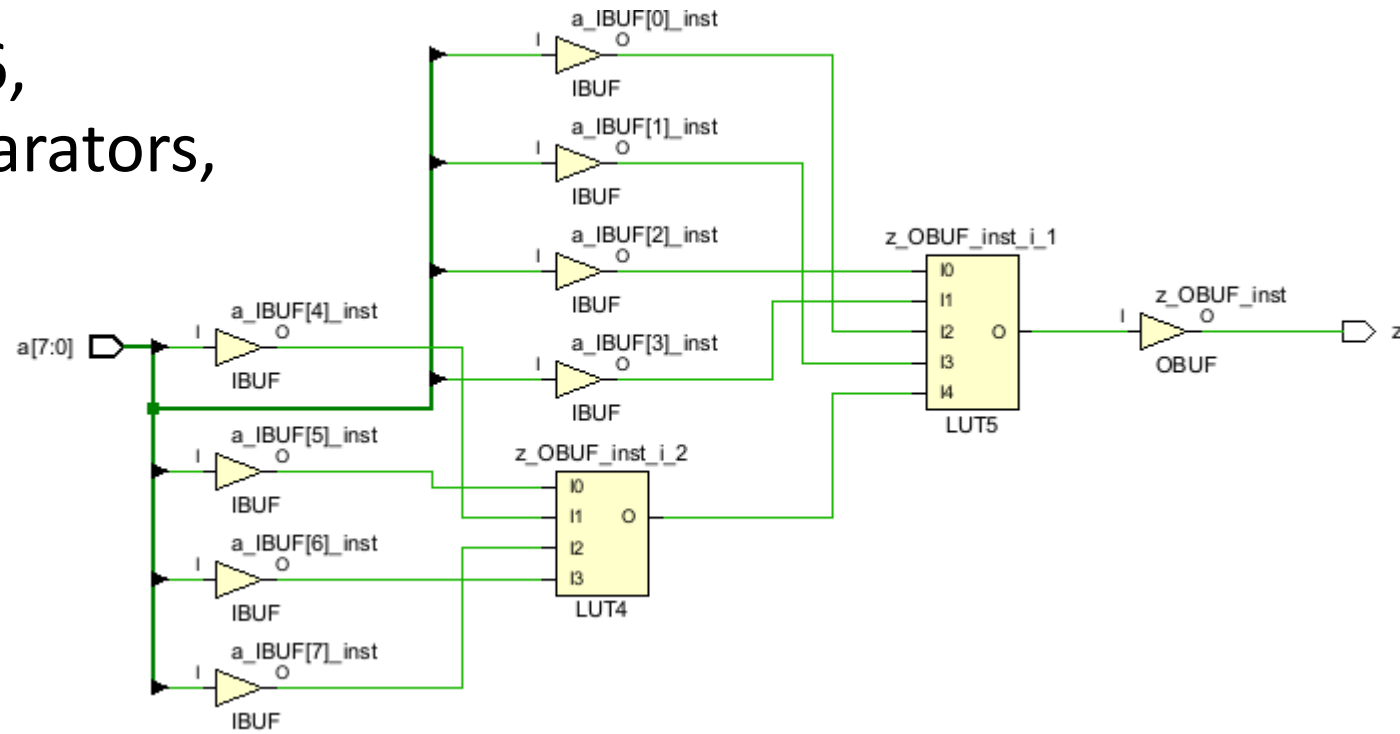


compact description of negating all bits (inversion)
try to describe z with a sequence of AND gates or
with reduction operator

Logic synthesis and FPGA utilization

1. SYNTHESIS, Run Synthesis
2. Open Synthesized Design, Schematic
3. Check FPGA utilization under Design Runs (number of LUT)

- **Task:** change input width to 6, 8, 10, 20 bits, describe comparators, synthesize and check results
- Explain utilization numbers!



SystemVerilog operators

- **Task:** Describe and evaluate equality and magnitude comparators

Inputs: N-bit vectors a and b

Output: is_equal (or is_higher)

- Write number of LUTs for all comparators with different input widths (N)

N	4	6	8	10	20
<code>a == 0</code>					
<code>a == b</code>					
<code>a > b</code>					

Background and Outcomes

- Understand SystemVerilog models with logic operators
- Use Vivado for design and synthesis
- Use online tool for simulation
- Understand implementation cost of basic logic operations

Prerequisites

- Digital circuit structures, signals and binary logic
- Basic understanding of FPGA structure and design flow