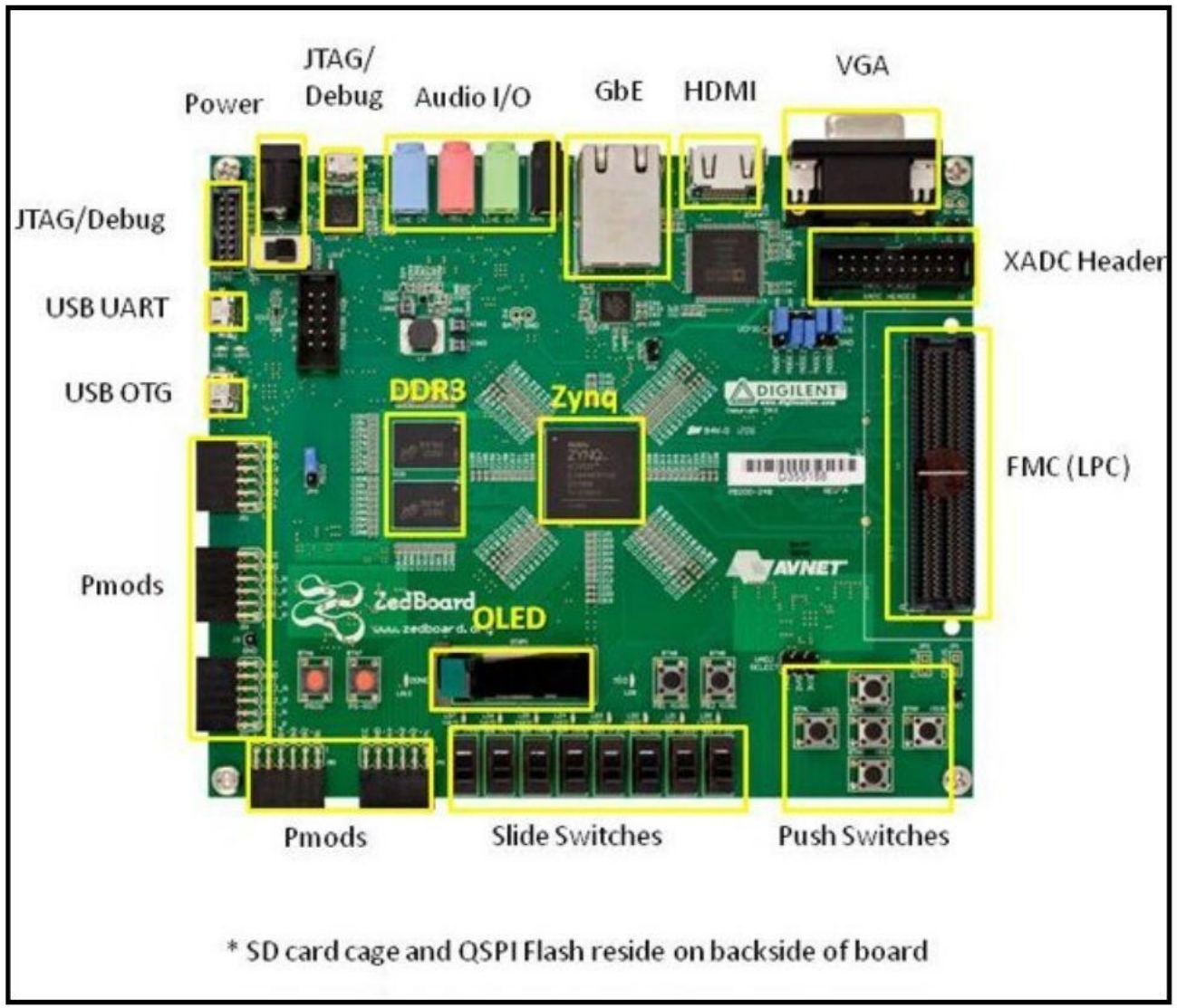
DCRUST - ECED

DIGITAL CIRCUIT IMPLEMENTATION ON ZEDBOARD

Aarti Kumari

Zedboard



FPGA:

- Xilinx Zynq-7000 SoC (Z-7020)
- Dual-core ARM Cortex-A9 processor
- Programmable logic (85K logic cells)

Memory:

- 512MB DDR3 RAM
- 256Mb Quad-SPI Flash
- 4GB SD card included (supports boot from SD)

Connectivity:

- Gigabit Ethernet
- USB OTG (micro-USB)
- USB UART
- HDMI output
- VGA output
- Audio In/Out (line-in, headphone out, mic)

Expansion:

- FMC (FPGA Mezzanine Card) connector
- 2 x Pmod connectors
- 1 x XADC header (analog input)

User Interface:

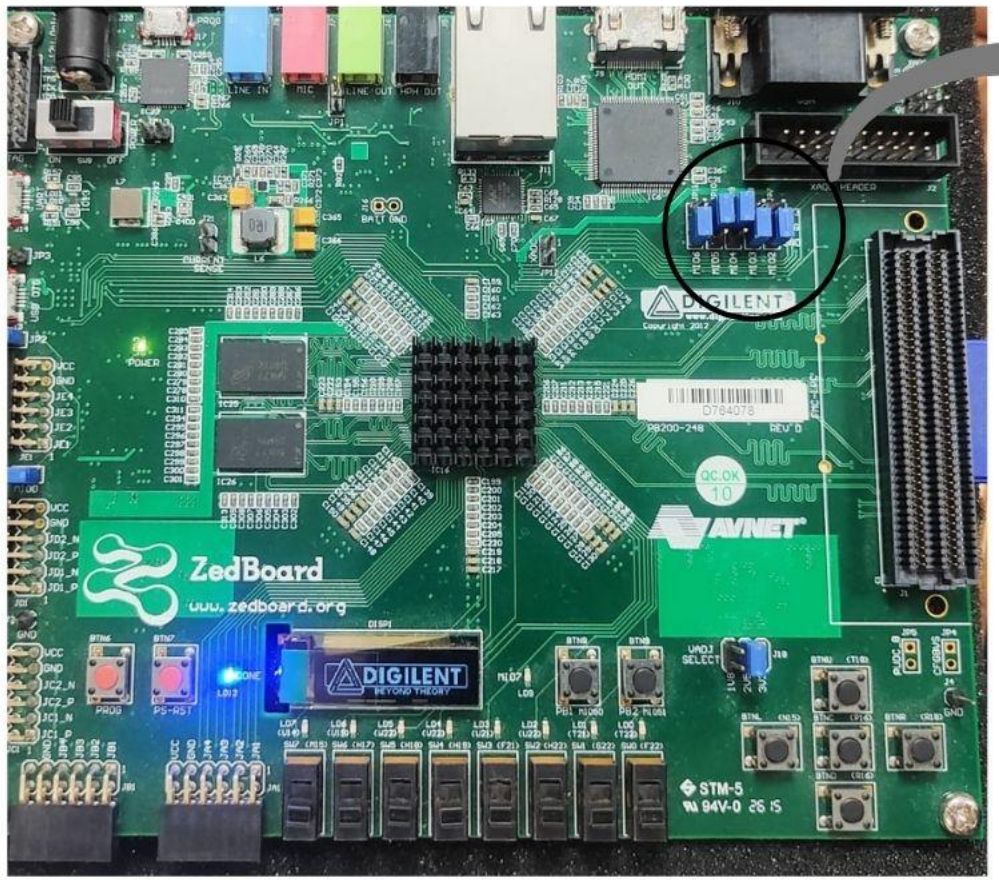
- Push buttons, slide switches
- LEDs (user and power)
- OLED display

Power:

- 12V input (AC adapter included)
- On-board power management

PIN Configuration

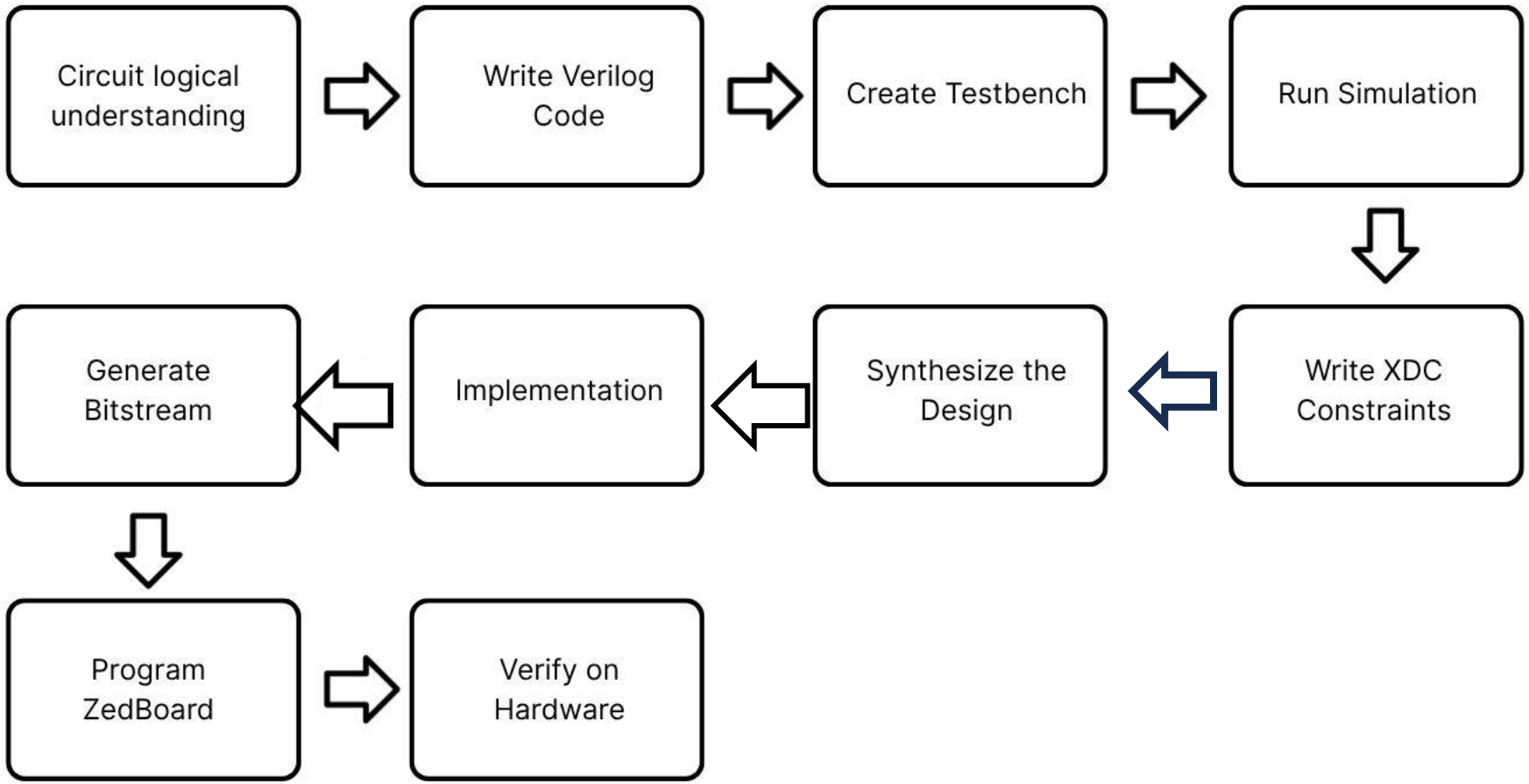
- Power on the ZedBoard with jumpers set and SD card inserted.
- Zynq's BootROM reads the SD card, loads BOOT.bin.
- FSBL runs, programs PL (FPGA bitstream), then hands off to U-Boot or Linux.
- System boots into Linux or your custom application.



SD CARD Boot mode:

Set JP 10 and JP 9 to 1, rest 0

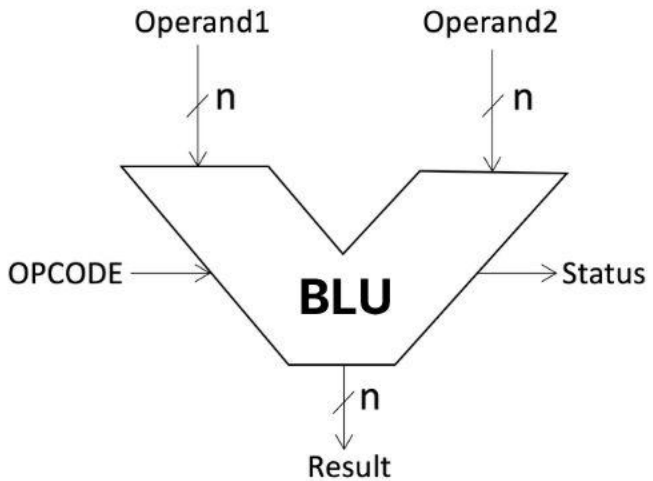
General Steps for FPGA-Based Circuit Implementation



BITWISE LOGIC UNIT(BLU):

An BLU takes in:

- Two binary inputs: A and B
- One control input: OP (operation selector)
- And produces a single output: RESULT




```
module blu_zed (
    input  [3:0] A,
    input  [3:0] B,
    input  [1:0] opcode,
    output reg [3:0] Y
);

always @(*) begin
    case (opcode)
        2'b00: Y = A & B;     // AND
        2'b01: Y = A | B;     // OR
        2'b10: Y = A ^ B;     // XOR
        2'b11: Y = ~A;        // NOT A
        default: Y = 4'b0000;
    endcase
end

endmodule
```

Opcode	Operation
0	A AND B
1	A OR B
10	A XOR B
11	NOT A

OUTPUT AND CONSTRAINT FILE

SW7–SW0	BTN (Center, Left)	Inputs	Opcode	Operation	Expected LED (Y)
0000 1111	00 (no press)	A = 0000 B = 1111	0	A AND B	0
1010 1100	0	A = 1010 B = 1100	0	A AND B	1000
1010 1100	01 (Left pressed)	A = 1010 B = 1100	1	A OR B	1110
1010 1100	10 (Center pressed)	A = 1010 B = 1100	10	A XOR B	110
1010 1100	11 (Both pressed)	A = 1010 B = 1100	11	NOT A	101
1111 0000	10 (Center pressed)	A = 1111 B = 0000	10	A XOR B	1111
0101 0101	01 (Left pressed)	A = 0101 B = 0101	1	A OR B	101
1111 1111	0	A = 1111 B = 1111	0	A AND B	1111
0000 0000	11 (Both pressed)	A = 0000 B = 0000	11	NOT A	1111

```
# INPUT SWITCHES A[3:0] - Connected to SW0-SW3
set_property PACKAGE_PIN F22 [get_ports {A[0]}]
set_property IOSTANDARD LVCMOS33 [get_ports {A[0]}]
set_property PACKAGE_PIN G22 [get_ports {A[1]}]
set_property IOSTANDARD LVCMOS33 [get_ports {A[1]}]
set_property PACKAGE_PIN H22 [get_ports {A[2]}]
set_property IOSTANDARD LVCMOS33 [get_ports {A[2]}]
set_property PACKAGE_PIN F21 [get_ports {A[3]}]
set_property IOSTANDARD LVCMOS33 [get_ports {A[3]}]

# INPUT SWITCHES B[3:0] - Connected to SW4-SW7
set_property PACKAGE_PIN H19 [get_ports {B[0]}]
set_property IOSTANDARD LVCMOS33 [get_ports {B[0]}]
set_property PACKAGE_PIN H18 [get_ports {B[1]}]
set_property IOSTANDARD LVCMOS33 [get_ports {B[1]}]
set_property PACKAGE_PIN H17 [get_ports {B[2]}]
set_property IOSTANDARD LVCMOS33 [get_ports {B[2]}]
set_property PACKAGE_PIN M15 [get_ports {B[3]}]
set_property IOSTANDARD LVCMOS33 [get_ports {B[3]}]

# BUTTONS for Operation Selection opcode[1:0]
set_property PACKAGE_PIN N15 [get_ports {opcode[0]}]
set_property IOSTANDARD LVCMOS33 [get_ports {opcode[0]}]
set_property PACKAGE_PIN P16 [get_ports {opcode[1]}]
set_property IOSTANDARD LVCMOS33 [get_ports {opcode[1]}]

# LED OUTPUT Y[3:0] → LD0-LD3
set_property PACKAGE_PIN T22 [get_ports {Y[0]}]
set_property IOSTANDARD LVCMOS33 [get_ports {Y[0]}]
set_property PACKAGE_PIN T21 [get_ports {Y[1]}]
set_property IOSTANDARD LVCMOS33 [get_ports {Y[1]}]
set_property PACKAGE_PIN U22 [get_ports {Y[2]}]
set_property IOSTANDARD LVCMOS33 [get_ports {Y[2]}]
set_property PACKAGE_PIN U21 [get_ports {Y[3]}]
set_property IOSTANDARD LVCMOS33 [get_ports {Y[3]}]
```

VIVADO WAVEFORM ANALYSIS

Inputs Used:

- A = 1010 (hex a)
- B = 1100 (hex c)

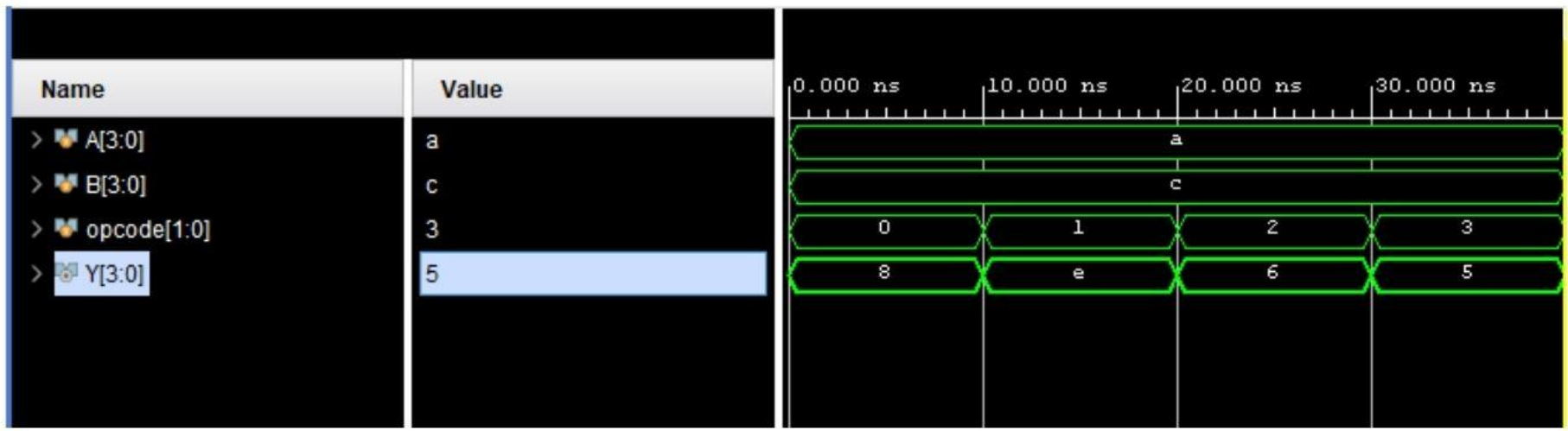
Opcode Changes Over Time:

- 00 → AND
- 01 → OR
- 10 → XOR
- 11 → NOT A

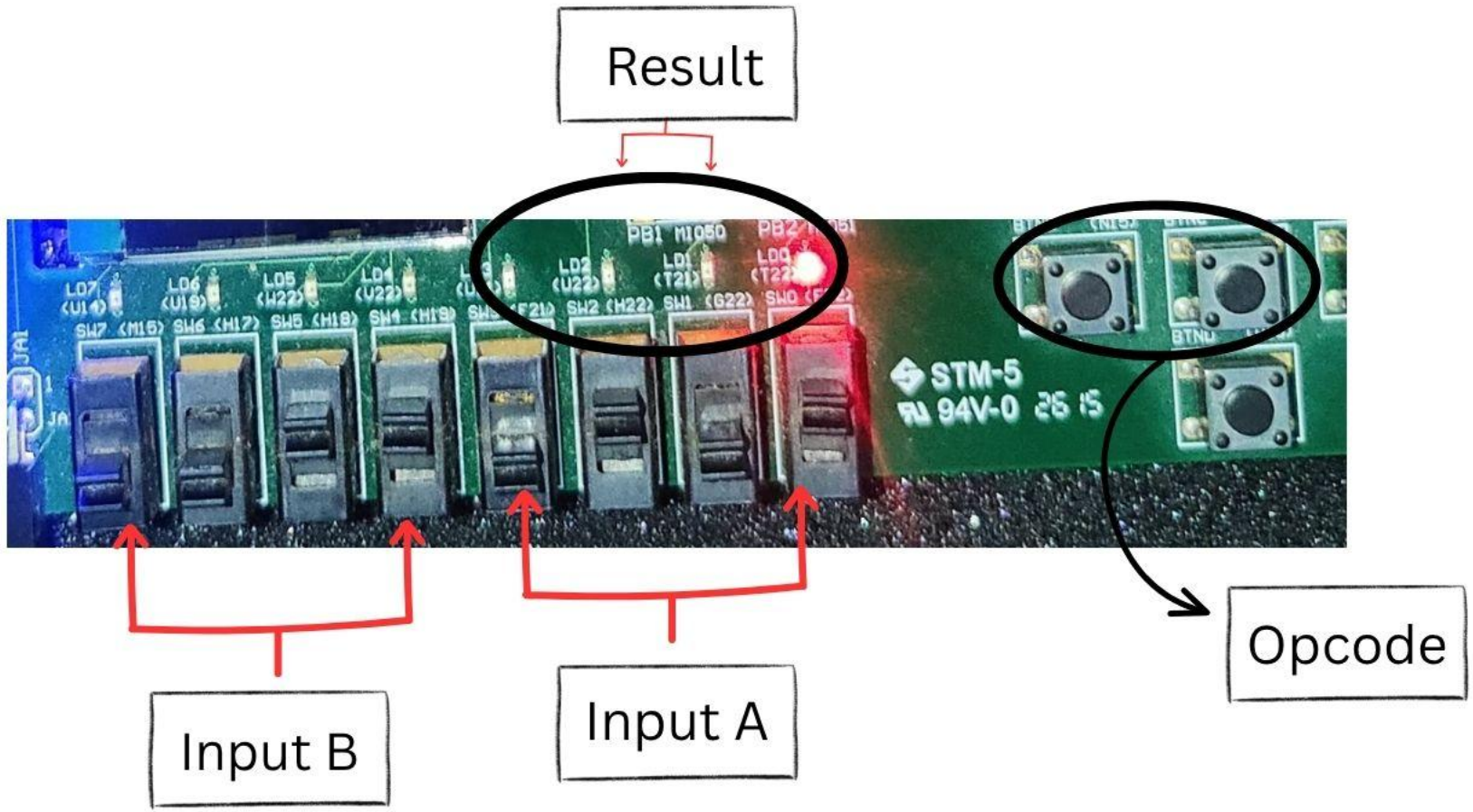
Simulation Cycles (Each 10 ns):

- 0 ns: opcode = 00 → Y = A & B = 1000 → Hex 8
- 10 ns: opcode = 01 → Y = A | B = 1110 → Hex e
- 20 ns: opcode = 10 → Y = A ^ B = 0110 → Hex 6
- 30 ns: opcode = 11 → Y = ~A = 0101 → Hex 5

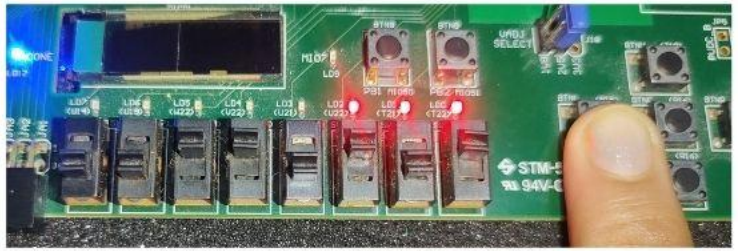
Opcode	Operation
0	A AND B
1	A OR B
10	A XOR B
11	NOT A



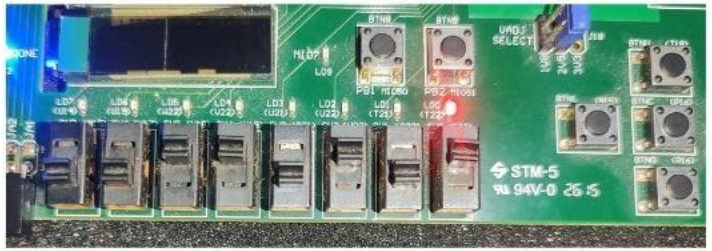
SW7–SW0	BTN (Center, Left)	Inputs	Opcode	Operation	Expected LED (Y)
1010 1100	0	A = 1010, B = 1100	0	A AND B	1000



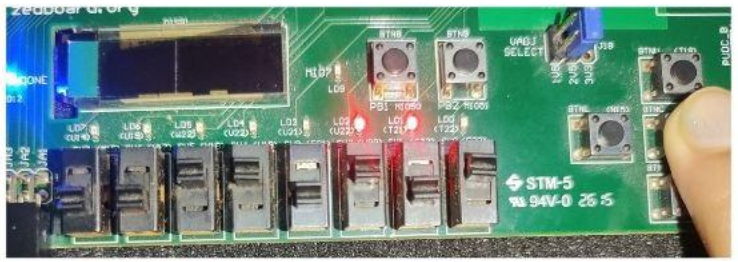
Verification on Zedboard



Test Case 1

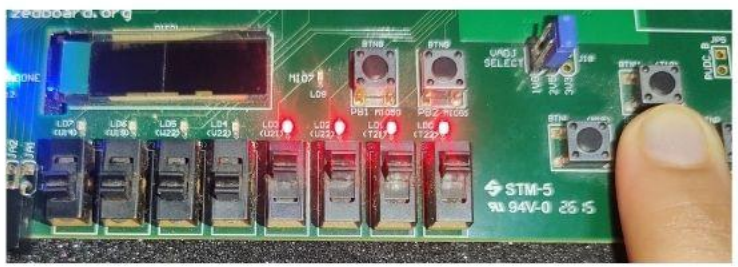


Test Case 2

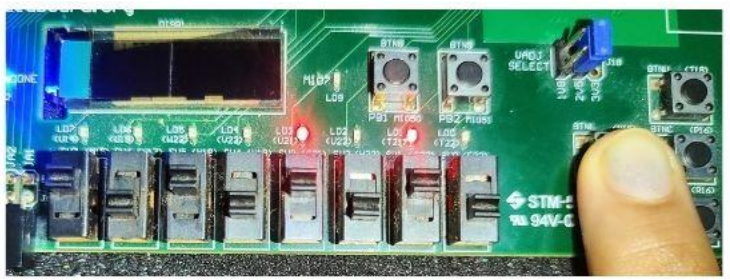


Test Case 3

Test Case 4



Test Case 5



Test Case 6