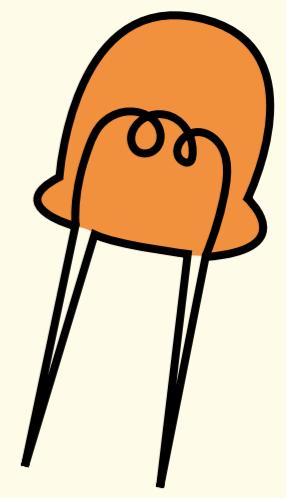
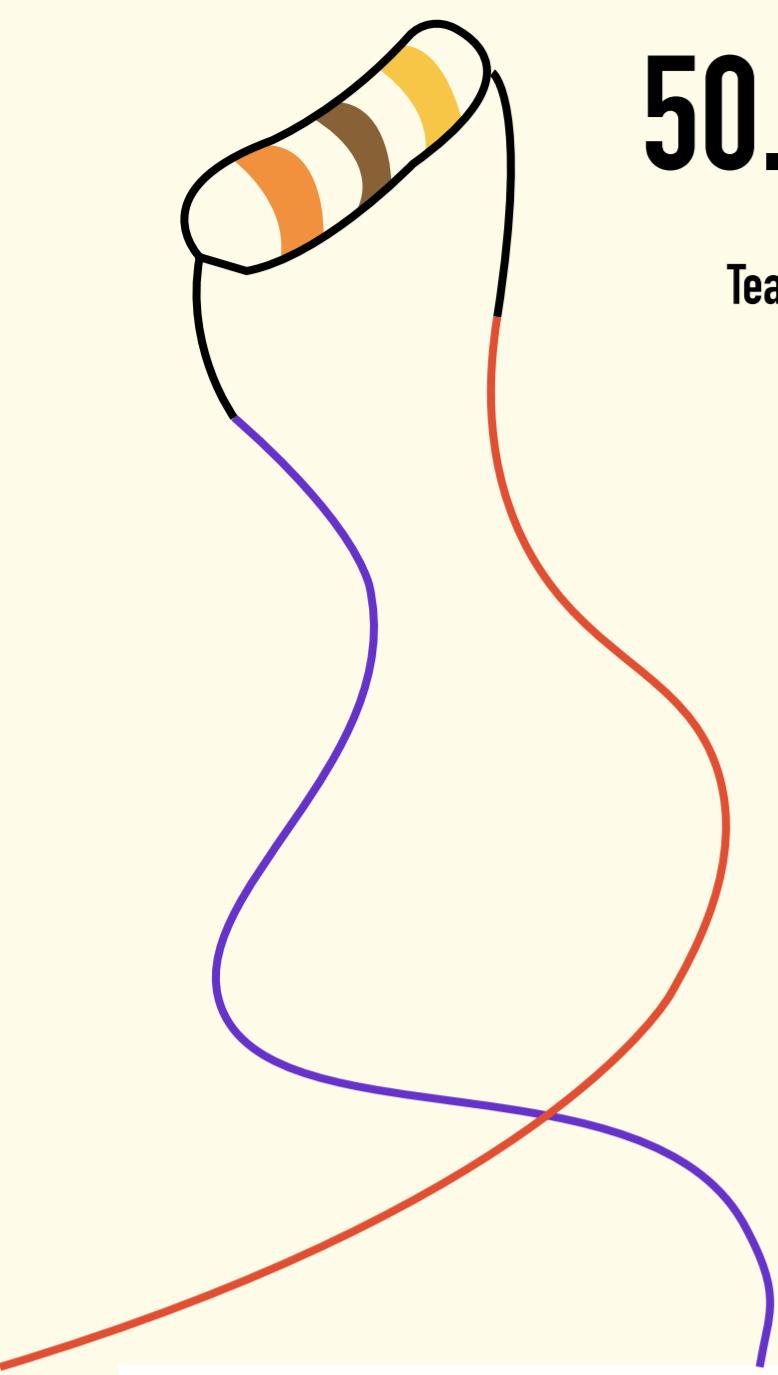


50.002 1D Part 1: Mini Hardware Project

Team 03-2: James Raphael Tiovalen, Ng Yu Yan, Jodi Tan Kai Yu, Cheow Wei Da Nicholas, Sun Kairan



Truth Table for Logic Gates

		74HC00 2-input NAND	74HC02 2-input NOR	74HC08 2-input AND	74HC32 2-input OR	74HC86 2-input XOR
A	B	Y	Y	Y	Y	Y
0	0	1	1	0	0	0
0	1	1	0	0	1	1
1	0	1	0	0	1	1
1	1	0	0	1	1	0

Truth Table of 1-Bit Half Adder

A	B	C _o (Carry-out)	S (Sum)
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

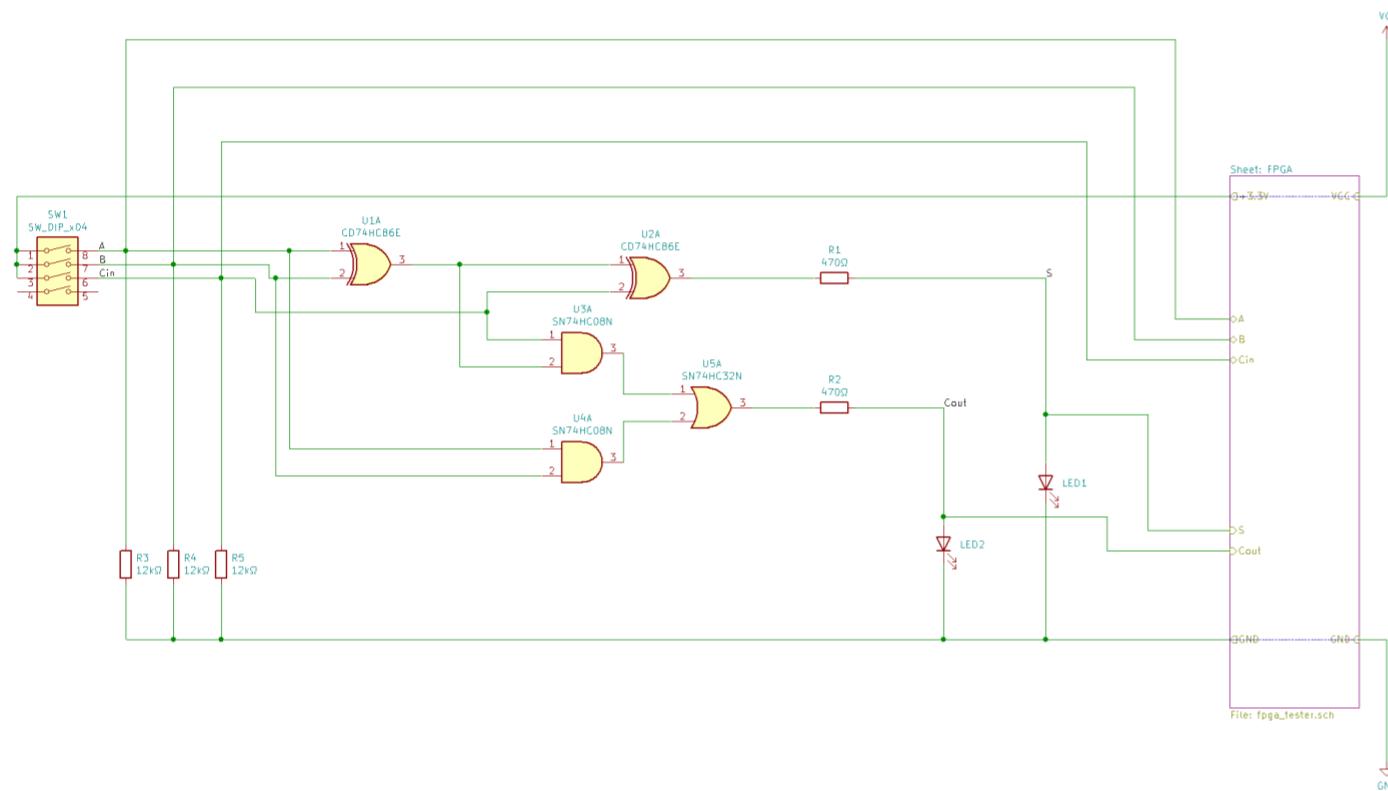
Test Result of 2-Input NAND

DI01 (nA)	DI00 (nB)	Output LED0 (nY)
0	0	1
0	1	1
1	0	1
1	1	0

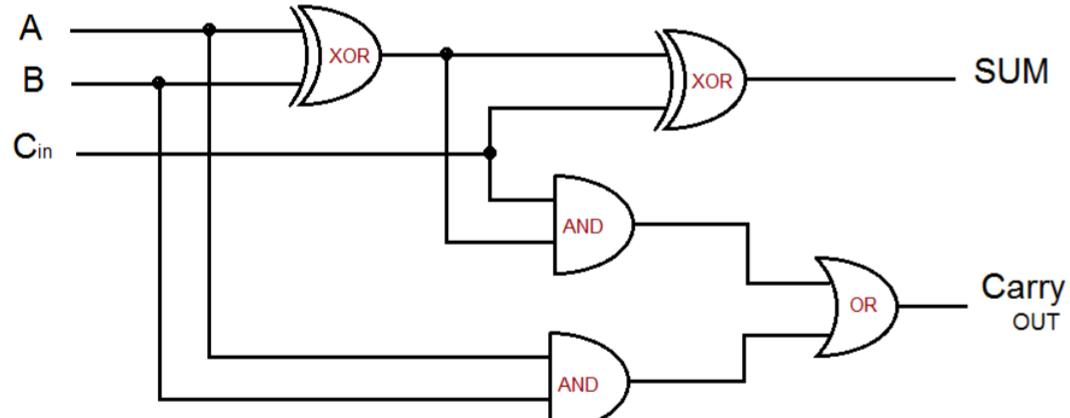
Truth Table for Inverter

74HC04 / 74H04 Inverter	
A	Y
0	1
1	0

1-Bit Full Adder Schematic Diagram



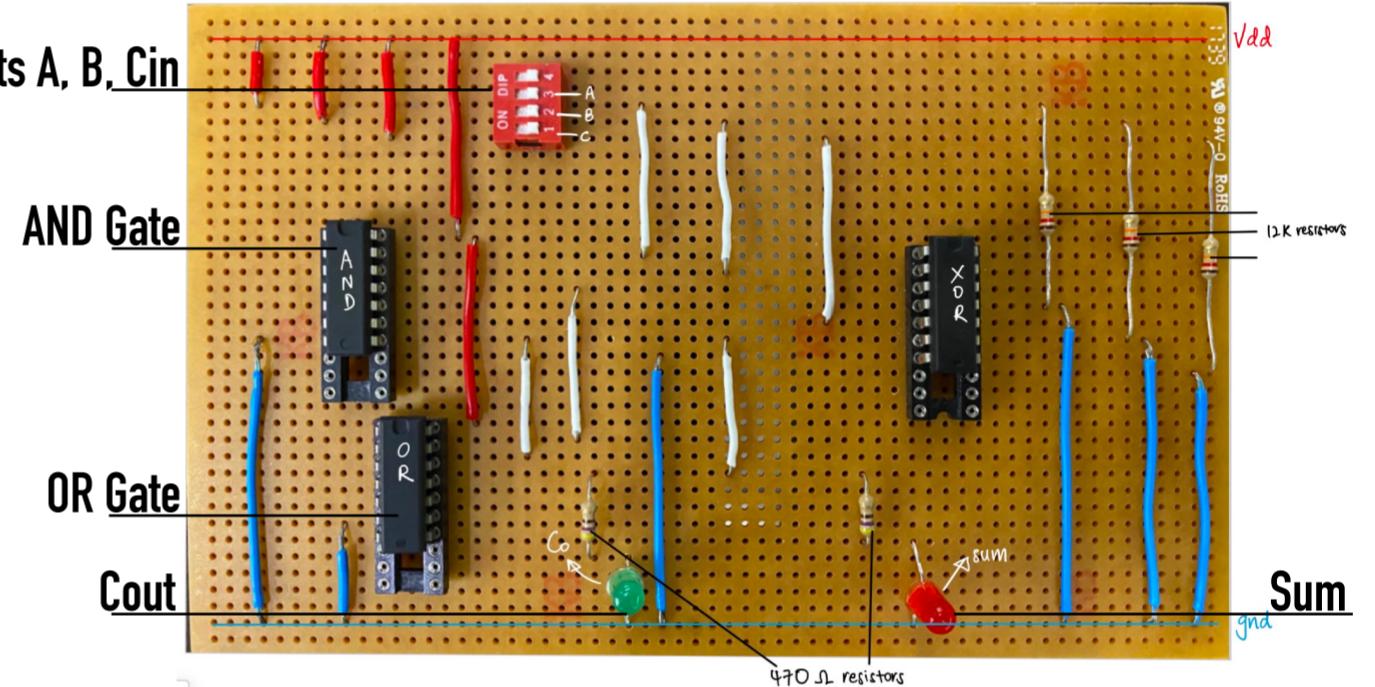
1-Bit Full Adder Logic Diagram



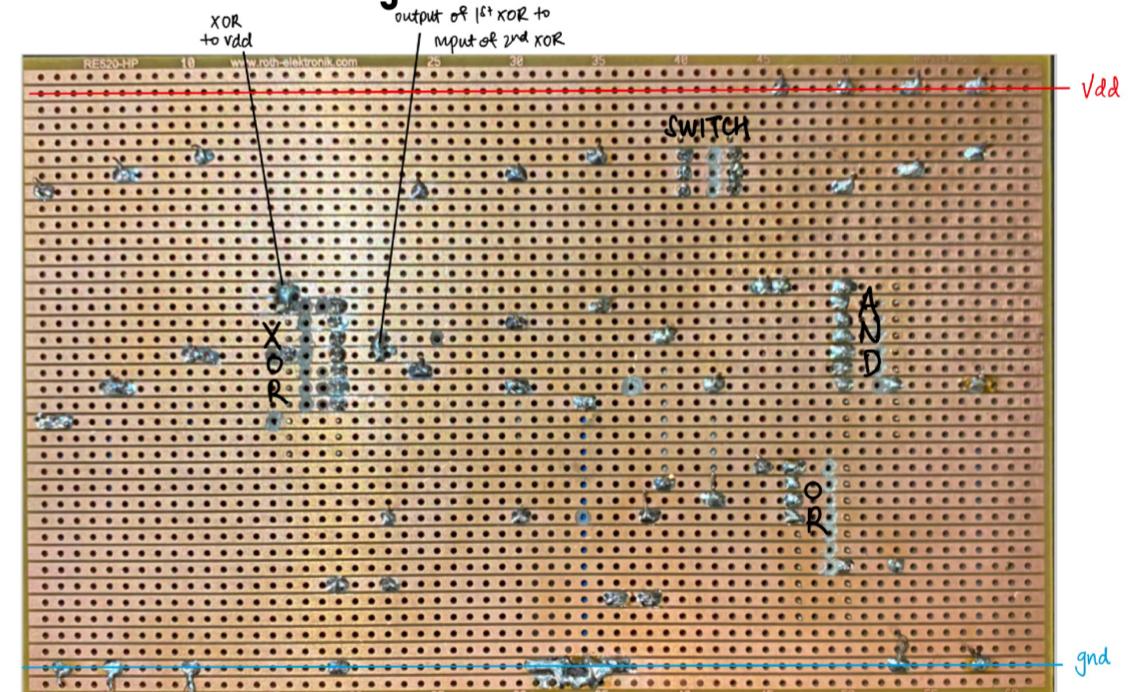
Components used:

- 1 XOR Gate
- 1 AND Gate
- 1 OR Gate
- 3 IC Socket
- 1 Alchitry Au + IO Shield
- 1 4-Position DIP Switch
- 1 Stripboard
- 2 470 Ω Resistors
- 3 12k Ω Resistors

1-Bit Full Adder Design Front



1-Bit Full Adder Design Back



FPGA Testing

1. Connect circuit to the provided VDD (+3.3 Volts) and GND on the custom-made SUTD ISTD Alchitry Br PCB board (any bank)
2. Connect the output Sum Bit wire to B42 and output Carry-Out Bit wire to B43
3. Turn off the switches on the Alchitry IO board for io_dip[2][2:0] and io_dip[1][1:0]
4. io_led[2][2:0] will serve as input LEDs (thus indicating the values of A, B and C_in)
5. io_led[1][1:0] will serve as output LEDs (thus indicating the values of S and C_out)

```
io_led[2][2:0] = adder_inputs; // show adder's input using left LEDs
adder_outputs = io_dip[1][1:0]; // read adder's output
io_led[1][1:0] = adder_outputs; // show adder's output using the middle LEDs
```

Manual Testing

- utilize the 1-Bit Full Adder's own DIP switch
- io_dip[0][0] is turned off (switched to LOW)

```
// switch between manual or auto
auto_test = io_dip[0][0];
```

Automatic Testing

- utilize the FPGA's internal circuitry to send clock governed signals to the external 1-Bit Full Adder circuit
- io_dip[0][0] is turned on (switched to HIGH)

```
if (!auto_test) {
    state.d = state.MANUAL;
}

end_test = test_gen.value[5];
adder_inputs = test_gen.value[4:2];
do_test = (test_gen.value[1:0] == 2b10);

state.MANUAL:
    if (auto_test) {
        state.d = state.AUTO;
    }

end_test = io_button[0];
adder_inputs = io_dip[2][2:0];
do_test = io_button[1];

// MANUAL as default
default:
    if (auto_test) {
        state.d = state.AUTO;
    }

else if (!auto_test) {
    state.d = state.MANUAL;
}

end_test = io_button[0]; // press UP button to stop the test
adder_inputs = io_dip[2][2:0];
do_test = io_button[1]; // press CENTER button to start the test
```

