EE332 Lab2: Simulation of Full Adder on Nexys 4 DDR

1st Qiu Kunyuan
EEE. Southern University of Science and Technology
Shenzhen, PRC
11913019@mail.sustech.edu.cn

***Abstract*—This report focuses on the phenomenons encountered in the simulation of a simple full adder, to reveal some critical features of FPGA programming comparing to ordinary programmable devices. Among these phenomenons, jitters and the race-hazard conditions are of the most concerns.**

***Index Terms*—FPGA, programmable logic, full adder, race-hazard condition, jitter, delay**

I. Introduction

Half adder and full adder are basic elements in digital circuits used to perform addition operations on binary numbers.

A *half adder*

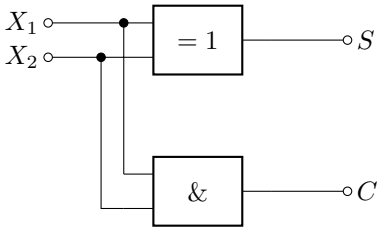

Fig. 1: Gate Level Description of Half Adder

is a combinational logic circuit shown in Figure.(1) that takes two binary inputs and produces two binary outputs: sum and carry.

$$\mathrm{HA} : (x_1, x_2) \mapsto (Q, C) := \begin{cases} Q & = \mathrm{xor}(x_1, x_2) \\ C & = \mathrm{and}(x_1, x_2) \end{cases} \tag{1}$$

The main limitation of the half adder is that it cannot handle carry inputs and therefore can only be used for 1-bit addition.

A *full adder* is an extension of the half adder that accepts three binary inputs: two additions and a rounding input, and produces two binary outputs: sum and carry. Full adders can be connected in series to achieve binary addition of any number of bits.

Implementation of full adder can be derived directly from 3-digit addition, where the final carry output is toggled if any of $X_1 + X_2$ or $C_i + Q_1$ produces a carry:

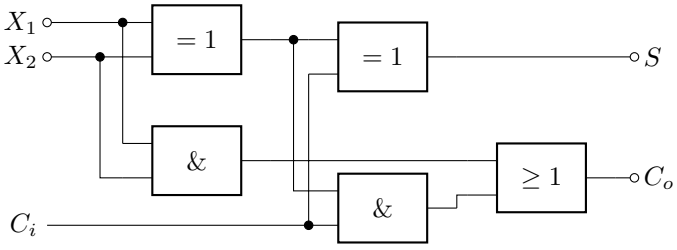

Fig. 2: Gate Level Description of Full Adder

$$(q_1, c_1) = \mathrm{HA}(x_1, x_2) \tag{2}$$
$$(q_2, c_2) = \mathrm{HA}(q_1, c_i) \tag{3}$$
$$c_o = \mathrm{or}(c_1, c_2) \tag{4}$$

Therefore the gate-level circuit of the full adder can be easily captured, as shown of Figure.(2)

II. Verilog Modeling

There are two levels of HDL modeling for full adder, one for gate level description and one for behavioral level description.

A. Gate-Level Implementation

```
`timescale 1ns / 1ps



module full_adder_g (
    input  wire c0,
    input  wire a,
    input  wire b,
    output wire s,
    output wire c1
);

  assign s  = ((~c0) & (~a) & b) | ((~c0) & a & (~b
     )) | (c0 & (~a) & (~b)) | (c0 & a & b);
  assign c1 = (c0 & a) | (a & b) | (c0 & b);

endmodule
```

Listing 1: Gate Level Modeling of Full Adder in Verilog

For the gate level modeling, the DNF (OR-AND) of the output ports are required:

$$
\begin{aligned}
Q_2 &= HA(HA(x_1, x_2)[0], c_i)[0] \\
&= \text{xor}(\text{xor}(x_1, x_2), c_i) \\
&= \overline{A}\overline{B}C + \overline{A}B\overline{C} + A\overline{B}\overline{C} + ABC \\
C_{\text{out}} &= AB + BC + AC
\end{aligned} \tag{5}
$$

Therefore, the Verilog code is pretty straight forward, as the Code.(2). For the Vivado elaboration and synthesis process, the logic functions are expanded to its DNF, meaning only AND/OR/NOT gates remains in the elaborated schematic:

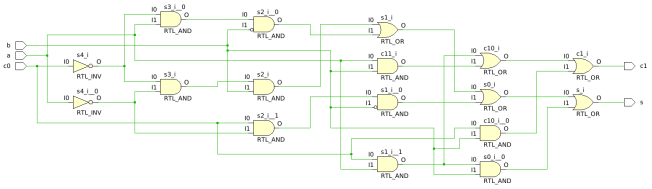

Fig. 3: Elaborated Gate Level Description

B. LUT Implementation

```
`timescale 1ns / 1ps



module full_adder_lut (
  input  wire [2:0] in_1,
  output wire [1:0] out_1
);

  assign out_1 = in_1[2] + in_1[1] + in_1[0];

endmodule
```

Listing 2: Gate Level Modeling of Full Adder in Verilog

The combine logic functions are implemented by LUT(Look-Up Table)s inside the FPGA components in real world, while the half adder is a common LUT element inside the CLBs. By simply configuring the switchable connections between the CLBs and the controlling MUXs inside the CLBs, a cascaded design can be easily obtained.

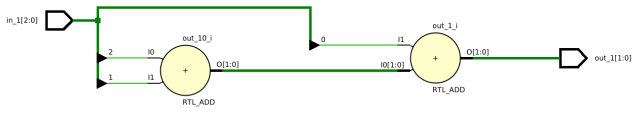

Fig. 4: Elaborated RTL Level Description

C. Synthesized Implementation

Either the gate level design (3) or the RTL level design (4) are synthesized into the same schematic (5) after the synthesis and implementation process, since the FPGA device only provide configurable LUT as the measure to combinational logic inside the CLB.

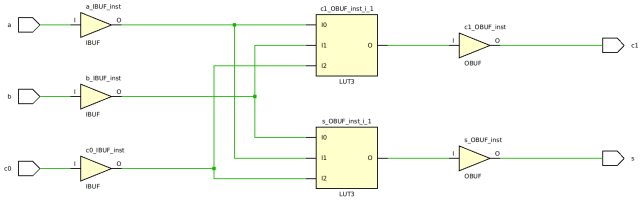

Fig. 5: Synthesized Design

X_1	X_0	C_i	dec($in[2:0]$)	C_o	S	dec($out[1:0]$)
0	0	0	0	0	0	0
0	0	1	1	0	1	1
0	1	0	2	0	1	1
0	1	1	3	1	0	2
1	0	0	4	0	1	1
1	0	1	5	1	0	2
1	1	0	6	1	0	2
1	1	1	7	1	1	3

TABLE I: Truth Table of Full Adder

In the synthesized schematic (5), the two LUT cells are combinational logic functions of $S = \text{LUT}_1(X_0, X_1, C_i)$ and $C_o = \text{LUT}_2(X_0, X_1, C_i)$ respectively.

III. Simulation

The truth table of the full adder is Table.(I).