2017 Digital IC Design
Homework 1: 4-bit Ripple Carry Adder

1. Introduction

The ripple carry adder (RCA) circuit can be built using a cascade architecture of 1-bit full-adders. As a result, a 4-bit RCA comprises four 1-bit full-adders, which is shown in Fig. 1. In addition, a full-adder can also be constructed by two half-adders and one *OR* gate. Assume that *x* and *y* are 1-bit input signals and *s* and *c* are outputs standing for *sum* and *carry*. The computation of the half adder can be represented as (1). Fig. 2 illustrates the architecture of a half adder and a full adder. In this homework, please design a 4-bit ripple carry adder according to Fig. 1 in **Gate Level**. You are suggested to design the circuit using hierarchy modules including series of *half adders* and *full adders*. The values are considered as unsigned integers.

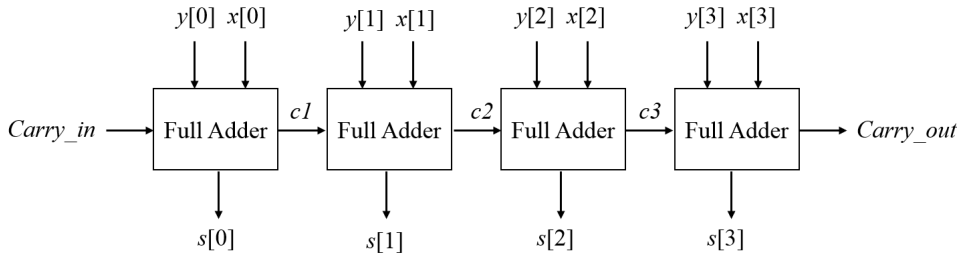


Fig. 1. The architecture of a 4-bit ripple carry adder.

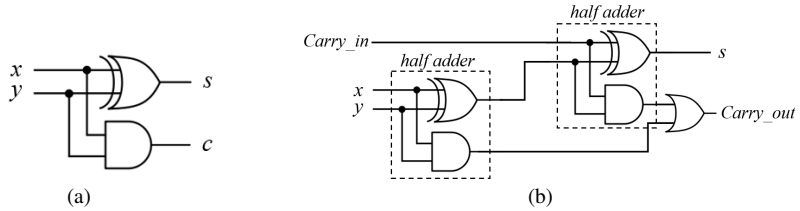


Fig. 2. (a) The architecture of a half-adder and (b) a full adder.

$$s = x \oplus y$$
$$c = x \cdot y \tag{1}$$

2. Design Specifications

2.1 Block Overview

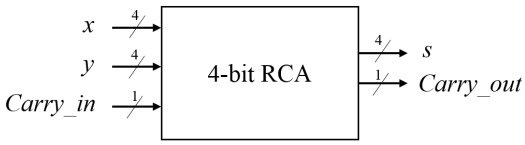


Fig. 3. The Block overview.

2.2 I/O Interface

Signal Name	I/O	width	Description
x	I	4	augend
y	I	4	summand
Carry_in	I	1	carry in
s	O	4	Sum
Carry_out	O	1	carry out

2.3 File Description

File Name	Description
HA.v	The module of half adder.
FA.v	The module of full adder.
RCA.v	The module of ripple carry adder, which is the top module in this design.
RCA_tb.v	The testbench file. You are **not allowed** to modify the content in this file.

3. Scoring

3.1 Functional Simulation (pre-sim) **[70%]**

All of the result should be generated correctly, and you will get the following message in ModleSim simulation.

```
#         508 data is correct
#         509 data is correct
#         510 data is correct
#         511 data is correct
# --------------------PASS--------------------
# All data have been generated successfully!
```

3.2 Gate-Level Simulation (post-sim) **[30%]**

3.2.1 Synthesis

Your code should be synthesizable. After synthesizing in Quartus, a file named *RCA.vo* will be obtained.

3.2.2 Simulation

All of the result should be generated correctly using *RCA.vo*, and you will get the following message in ModleSim simulation.

```
#         508 data is correct
#         509 data is correct
#         510 data is correct
#         511 data is correct
# --------------------PASS--------------------
# All data have been generated successfully!
```

4. Submission

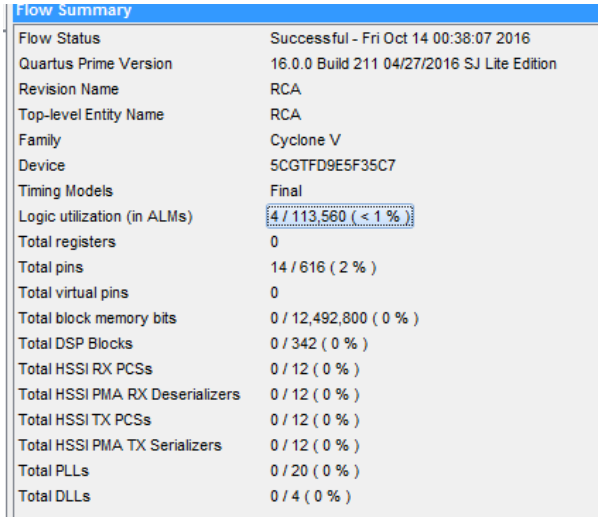
4.1 Submitted files

You should classified your files into three directories and compressed to .zip format. The naming rule is **HW1_*studentID_name_version*.zip**. The vision is v1 for the first submission, and v2, v3… for the revisions.

RTL category	
*.v	All of your verilog RTL code
Gate-Level category	
*.vo	Gate-Level netlist generated by Quartus
Documentary category	
*.pdf	The report file of your design (in pdf).

4.2 Report file

You have to describe how the circuit is designed as detailed as possible, and the Flow Summary result after synthesis is necessary in the report.



Flow Summary	
Flow Status	Successful - Fri Oct 14 00:38:07 2016
Quartus Prime Version	16.0.0 Build 211 04/27/2016 SJ Lite Edition
Revision Name	RCA
Top-level Entity Name	RCA
Family	Cyclone V
Device	5CGTFD9E5F35C7
Timing Models	Final
Logic utilization (in ALMs)	4 / 113,560 (< 1 %)
Total registers	0
Total pins	14 / 616 (2 %)
Total virtual pins	0
Total block memory bits	0 / 12,492,800 (0 %)
Total DSP Blocks	0 / 342 (0 %)
Total HSSI RX PCSs	0 / 12 (0 %)
Total HSSI PMA RX Deserializers	0 / 12 (0 %)
Total HSSI TX PCSs	0 / 12 (0 %)
Total HSSI PMA TX Serializers	0 / 12 (0 %)
Total PLLs	0 / 20 (0 %)
Total DLLs	0 / 4 (0 %)

Fig. 4. The Flow Summary.

4.3 Please submit your .zip file to folder HW1 in the ftp site.

Deadline: 2017-11-06 23:59

ftp : 140.116.245.92

Usermame : ic_design

Password : icdesign

5. If you have any problem, please contact the TA by email : p78031175@mail.ncku.edu.tw