Indian Institute of Technology (IIT-Kharagpur)

AUTUMN Semester, 2024
COMPUTER SCIENCE AND ENGINEERING

Computer Organization and Architecture Laboratory

Verilog 1

INSTRUCTIONS: Please answer all the questions.
Special credit would be given for answers which are to-the-point.
Please ensure that your handwriting is legible.
If you prefer to submit printouts of the typed solutions, please typeset your work in LaTeX.

Introduction to Verilog Programming

In this assignment, we shall be designing and comparing two adder circuits using the Verilog Programming Language.

1. *Write the Verilog Code for an n-bit adder. Proceed step by step as follows:*

- *(a) A* Half Adder *is a combinational circuit, which takes in two input bits, a and b, and produces the sum bit, s and the carry-out bit, c. Write the truth-table for the assignments of s and c. Write the verilog code for the* Half Adder.
- *(b) A* Full Adder *is a combinational circuit, which takes in three input bits, a, b, and in addition a carry-in bit c_0, and produces the sum bit, s and the carry-out bit, c. Write the truth-table for the assignments of s and c in the Full-Adder. Write the verilog code for the* Full Adder.
- *(c) Implement two separate designs using behavioral and structural coding styles respectively.*
- *(d) Cascade 8 Full adders and create an 8-bit adder. These type of adders are called* Ripple Carry Adders. *Like-wise create,* 16, 32, *and* 64 *bit adders, and observe the longest delays in the circuits.*
- *(e) How can you use the above circuit, to compute the difference between two n-bit numbers?*

2. *In the previous assignment, we have designed a Ripple Carry Adder (RCA). One of the weaknesses in the adder is the large delay because of the rippling of the carry, through what we call as the carry-chain. In this assignment, we endeavour to design a high-speed adder using the technique of what we call as the Carry Look-ahead Adder (CLA). In this exercise, we start with designing a 4-bit CLA and then use it to design a 16-bit CLA using a hierarchical structure.*

Proceed as follows:

- *(a) First, design a 4-bit CLA. Clearly state, the Boolean equations of the Look-ahead carry generation for the 4 carry bits, C1, C2, C3, and C4 in terms of the generate and propagate signals, denoted as $G0, \ldots, G3$ and $P0 \ldots, P3$ respectively. Also state the equations for the corresponding generate and propagate signals.*
- *(b) Design the 4-bit CLA using verilog and compare its speed with a 4-bit RCA (ensure that the KEEP_HIERARCHY option in your synthesizer is set TRUE). Check the correctness of the code by a test-bench in verilog and compare the speeds from the synthesis reports generated by the Xilinx tool.*
- *(c) Download the design onto the FPGA and check for the correctness of the design.*

(d) Now let us try to make a 16-bit adder in a hierarchical *fashion. For this, we observe that* $C4 = G3 + P3G2 + P3P2G1 + P3P2P1G0 + P3P2P1P0C0$. *One can create the 16-bit adder in two ways. In both the ways, we reuse the 4-bit CLA just designed by you. However, in one of the ways we ripple in the carry out from the 4-bit CLA to the second stage, while in the other we compute the carry in of the second 4-bit CLA by using a separate level of* Lookahead Carry Unit. *This lookahead carry unit works on block propagates and block generates, denoted as P, G in the circuit shown in* **Fig 1**. *The definitions of the block propagate and block generate are as follows:*

$$P = P3P2P1P0 \quad (1)$$

$$G = G3 + P3G2 + P3P2G1 + P3P2P1G0 \quad (2)$$

Thus, we can write $C4 = G + PC0$. *Now if you consider wrt. the Lookahead Carry Unit, this is the logic for the carry bit C1. Likewise, one can compute the logic for the carry bits, C2, C3, and C4 using the block propagates and generates. We also compute P3-0 and G3-0 which are block propagates and generates for the next level of the hierarchy (if one wishes to design 32 or 64 bit adders using this structure as a component).* Note that the labels for the propagates and generates can be confusing, but since the labels are written internal to the blocks (or verilog modules) they dont conflict with each other.

In this part you are supposed to:

(i) Augment the 4-bit adder to compute the block propagate and generate signals. Test the circuit using a testbench in verilog.

(ii) Design the Lookahead Carry Unit and integrate as shown in **Fig 1**.

(iii) Compare this circuit by annotating the delays for obtaining the sum and final carry out bits for the 16-bit adder, with that if the carry was rippled in (without using the second layer of lookahead).

(iv) Synthesize the results, and see how the speed and the LUT (lookup-table cost of the FPGA) compares with a 16-bit RCA.

(v) Prototype and see that the design works on the FPGA Board.

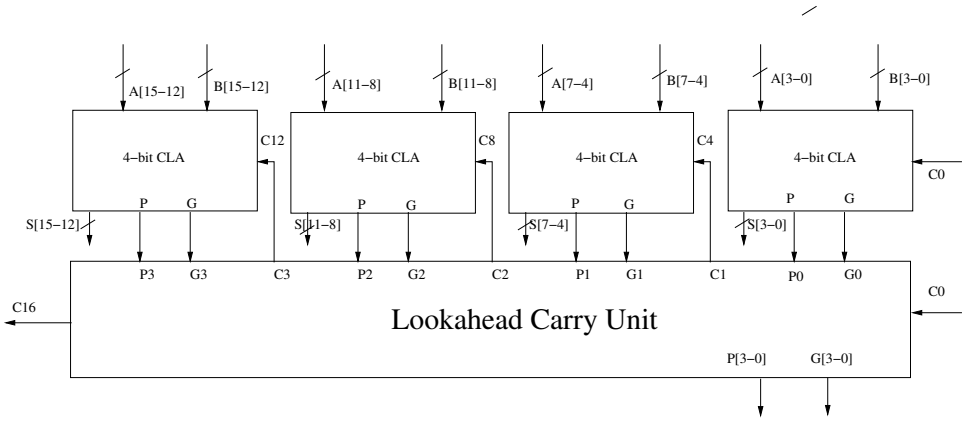

Figure 1: The Hierarchical Structure of a 16-bit CLA