Pre-Lab 9: An Introduction to High-Speed Addition

ECEN 248 - 505

TA: Younggyun Cho

Date: October 27, 2020

1. With dataflow Verilog, describe the Generate/Propagate Unit, the Carry-Lookahead Unit, and the Summation Unit in Figure 1 as separate modules. Do not include delays in your models. We will add them later in the lab experiments. Use the module interfaces below as a guide. Gate-level schematics can be hard to read so you may find expressions (3) through (11) easier to follow.

```
// This module describes the generate propagate unit
module generate_propagate_unit(G,P,X,Y);
// ports are wires as we will use dataflow
        output wire [3:0] G,P;
        input wire [3:0] X,Y;
        wire X3, X2, X1, X0;
        wire Y3, Y2, Y1, Y0;
        wire G3, G2, G1, G0;
        wire P3, P2, P1, P0;
        and and0(G3,X3,Y3);
        and and1(G2,X2,Y2);
        and and2(G1,X1,Y1);
        and and3(G0,X0,Y0);
        xor xor3(P3,X3,Y3);
        xor xor2(P2,X2,Y2);
        xor xor1(P1,X1,Y1);
        xor xor0(P0,X0,Y0);
endmodule

// This module describes the carry look ahead unit
module carry_lookahead_unit(C,G,P,C0);
        output wire [4:1] C; // C4, C3, C2, C1
        input wire [3:0] G,P; // generates and propagates
        input wire C0; // input carry in
        wire and00, and01, and02, and03, and04, and05, and06, and07, and08; //outputs of and
gates
        and and0(and00,P3,P2,P1,P0,C0)
         and and1(and01,P3,P2,P1,P0,G0)
        and and2(and02,P3,P2,P1,P0,G1)
        and and3(and03,P3,G2)
        and and4(and04,P2,P1,P0,P0,C0)
        and and5(and05,P2,P1,G0)
        and and6(and06,P2,G1)
        and and7(and07,P1,P0,C0)
        and and8(and08,P1,G0)
```

```
        and and9(and09,P0,C0)
        xor xor0(C4,and00,and01,and02,and03,and04,G3)
        or or0(C3,and05,and06,and07,G2)
        or or1(C2,and08,and09,G1)
        or or2(C1,and09,G0)
endmodule
// This module describes the summation unit
module summation_unit(S,P,C);
        output wire [3:0] S; // sum vector
        input wire [3:0] P,C; // propagate and carry vectors
         xor xor0(S3,P3,C3)
         xor xor1(S2,P2,C2)
         xor xor2(S1,P1,C1)
         xor xor3(S0,P0,C0)
```

2. Now, use structural Verilog along with the modules you have just created to wire up a 4-bit Carry-Lookahead adder. The module interface you should use if provided below.

```
// This is the top-level module
module carry_lookahead_4bit(Cout,S,X,Y,Cin);
        output wire Cout; // C_4 for a 4-bit adder
        output wire [3:0] S; // final 4-bit sum vector
        input wire [3:0] X,Y; // 4-bit addends
        input wire Cin; // input carry in
        summation_unit sum0(S,P,C)
        carry_lookahead_unit carry0(C,G,P,C0)
         generate_propagate_unit gen0(G,P,X,Y)
```

3. What is the gate-count of your 4-bit carry-lookahead adder?

The gate-count for the 4-bit carry-lookahead adder is 26 gates.

4. The previous problems were concerned with a single-level 4-bit carry-lookahead adder. In one of the lab experiments, we will construct a 16-bit, 2-level carry-lookahead adder. The following questions will prepare you for this exercise. What is the propagation delay of the 16-bit, 2-level carry-lookahead adder in Figure 2? Likewise, what is the gate-count?

With 82 gates and the prop delay is 2ns per gate, the total delay is 164ns.