JOINT INSTITUTE
交大密西根学院

Ve270 Introduction to Logic Design

Homework 10

Assigned: July 25, 2019

Due: August 1, 2019, 2:00pm.

The homework should be submitted in hard copies.

1. Problem 5.19 (30 points)

5.19 Using a timer, design a system with single-bit inputs U and D corresponding to two buttons, and a 16-bit output Q which is initially 0. Pressing the button for U causes Q to increment, while D causes a decrement; pressing both buttons causes Q to stay the same. If a single button is held down, Q should then continue to increment or decrement at a rate of once per second as long as the button is held. Assume the buttons are already debounced. Assume Q simply rolls over if its upper or lower value is reached.

2. Problem 5.27 (20 points)

5.27 Convert the following C-like code, which calculates the greatest common divisor (GCD) of the two 8-bit numbers a and b, into a high-level state machine.

```
Inputs: byte a, byte b, bit go
Outputs: byte gcd, bit done
GCD:
while(1) {
  while(!go);
  done = 0;
  while ( a != b ) {
    if( a > b ) {
      a = a - b;
    }
    else {
      b = b - a;
    }
  }
  gcd = a;
  done = 1;
}
```

3. Problem 5.28 (20 points)

5.28 Use the RTL design process to convert the high-level state machine created in Exercise 5.27 to a controller and a datapath. Design the datapath to structure, but design the controller to an FSM and then stop.

4. Problem 6.27 (10 points)

6.27 Trace the execution of the 16-bit carry-lookahead adder built from 4-bit adders as shown in Figure 6.60 when a = 43690 and b = 21845. Do not trace internal behavior of the individual 4-bit carry-lookahead adders.

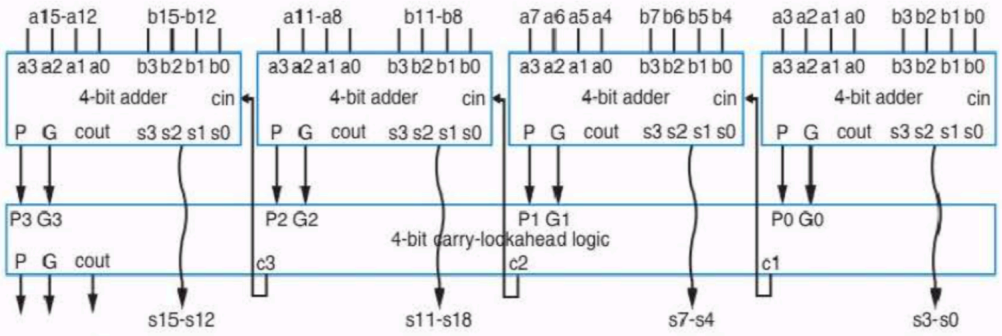

Figure 6.60 16-bit adder implemented using four CLA 4-bit adders and a second level of lookahead.

5. Problem 6.28 (10+5+5 points)

6.28 (a) Design a 64-bit hierarchical carry-lookahead adder using 4-bit carry-lookahead adders. (b) What is the total delay through the 64-bit adder? (c) What is the speedup of the carry-lookahead adder compared to a 64-bit carry-ripple adder; compute speedup as (slower time)/(faster time).