CSE140: Sample RTL Problems

Tajana Simunic Rosing
We are getting close to the end!!!

• Last time:
  – Completed combinational and sequential design, HLSMs, memory, CPU
• Plan for today:
  – Midterm #3 sample problems
• Deadlines coming up:
  – HW#7 due today; bonus HW grade
• Reminders
  – Exam #3 Friday, March 15th
    • 8 ½ x 11” sheet of paper with handwritten notes; ID, pencil and eraser
    • 4 problems, focus on recent material, but exam is cumulative
  – Bonus final exam worth 1% of course grade
    • Released on Monday of final’s week
    • Due on Friday of final’s week, turn it into gradescope
• Office hours this week MW 1-2pm, no office hrs on Friday
• Please fill out CAPEs at http://cape.ucsd.edu, I take your feedback very seriously and use it to improve the course. Please provide written comments as well.
  – If 95% of the class fill out CAPEs -> I will drop the lowest HW and Zybook grades!
Simple data encryption/decryption device

- $B = 1$, set offset $O = I[0:31]$
- $B = 0$, $e = 1$: encrypt mode: output $J = I + O$
- $B = 0$, $e = 0$: decrypt mode: get $I = J - O$
Hot Water Detector

Create an alarm system that sets alarm=1 when the average temperature of four consecutive samples CT meets or exceeds a threshold WT. Signal clr=1 disables the alarm.
Hot Water Detector cont.
RTL Examples: Reaction Timer

On reset \((rst)\) reaction timer waits for 10 sec before turning on light \((len=1)\). It measures the length of time \(rtime\) (ms) until a user presses the button \(B\)

- If reaction slower than 2sec, output \(slow=1\) and \(rtime=2000\)
Fast sum of 16 32-bit registers
Design from “C” code

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;
    }
Fibonacci Lookup Table

• Design a lookup table 256 x 256 bit that stores Fibonacci #s:
  – $F_n = 0$ if $n = 0$, 1 if $n = 1$, $F_{n-1} + F_{n-2}$ otherwise ($n<256$)
Finish HLSM Design

- Design an 8-bit counter using RTL:
  - When input $E = 1$, it counts even numbers (0, 2, 4, 6, ..) and when $E = 0$, it counts odd numbers (1, 3, 5, 7, ..).
  - When input $CLR = 1$ and $E=1$, then it clears the output to 0; if $CLR=1$ and $E=0$, it sets output to “00000001”.
  - If you were initially counting even(odd) numbers, and $E$ flips, then the output changes to the nearest greater odd (even) value.