DIGITAL STOPWATCH

Derrick McElwee
Cherith-Eden Clements

February 24, 2012

EECE494 Computer Bus and SoC Interfacing
Electrical and Computer Engineering
Howard University

Instructor: Dr. Charles Kim

Overview

- Getting an initial idea
- Pseudo-coding
- Actual Coding/Debugging
- Pin Planning
- Verifying Design
- Finalizing Design

So many pieces, which to choose??

- LEDs
- LCD
- Seven-Segment Display
- Push pins
- Switches
- RAM

Narrowing Down Parts

- Just choose 1-2 parts to be the “flashy part” of your design.
- Ex: using switch to control speed of LED flashing.
- Using LEDs to count in binary while Seven Segment counts in decimal.
- Displaying words on LCD

What to do with parts you have?

- Seven Segment – can display words or numbers.
  - Numbers: digital clock/stop watch/etc
  - Words: Spell out a word across the displays.

- LEDs – can use green or red LEDs
  - Display a set pattern
  - Flash lights randomly etc

Choosing Which Project To Go With

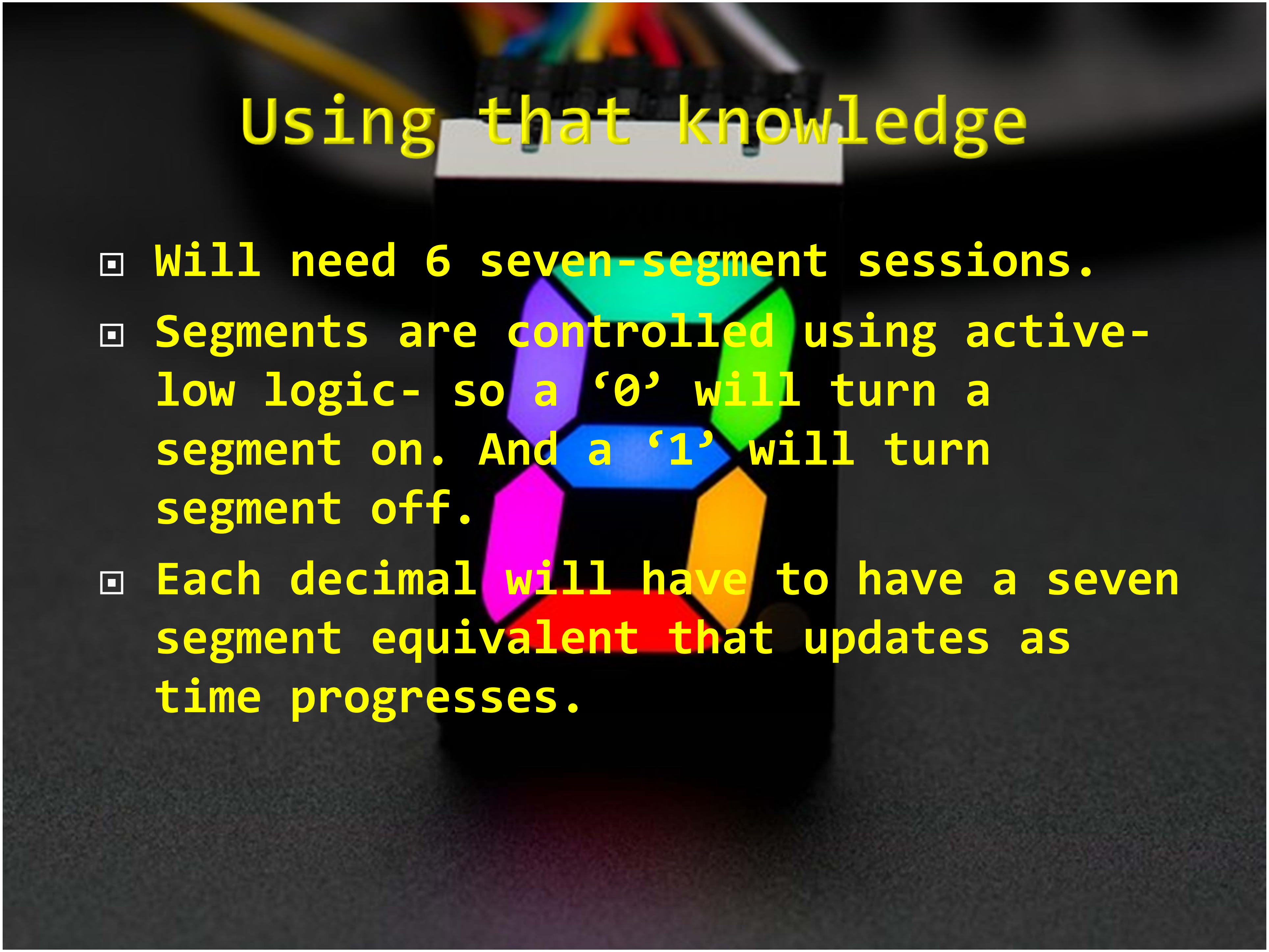
- Know your coding/Google capabilities
- Will it be feasible without tearing your hair out.
- Don’t do excessive amounts of extra.
- Is easier to add onto a basic project than to shell out an extravagant one from the get go.

I Choose You!

- Digital Stopwatch
- Let’s get to work

What does a stopwatch do?

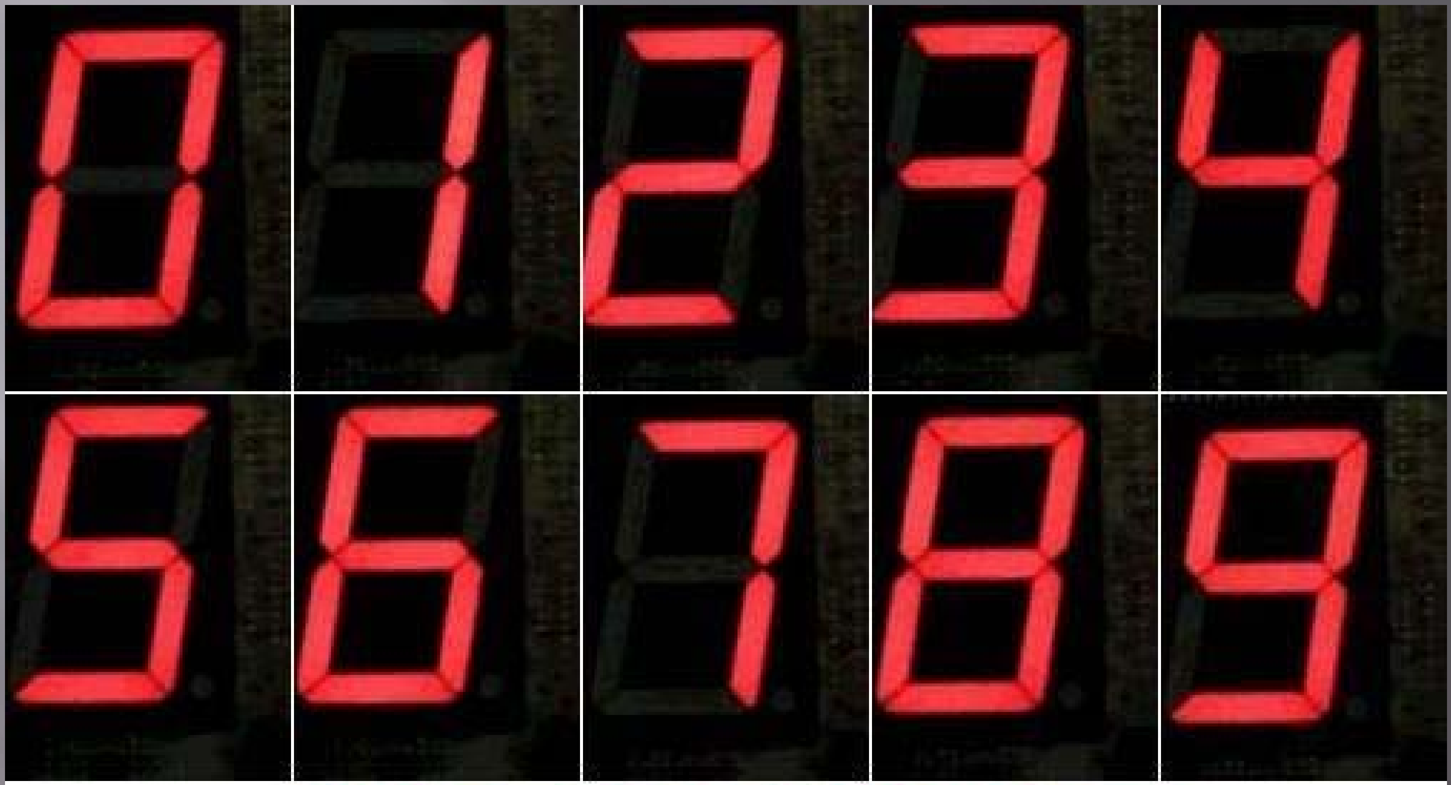
- Seconds turn into minutes
- Minutes turn into hours
- Designer issue- should implementation be to make 12 hours and restart at 1 for hours, or go to 23 then go to 00.



Using that knowledge

- Will need 6 seven-segment sessions.
- Segments are controlled using active-low logic- so a ‘0’ will turn a segment on. And a ‘1’ will turn segment off.
- Each decimal will have to have a seven segment equivalent that updates as time progresses.

Closer look at Seven Segment



http://program-plc.blogspot.com/

Pseudocoding- Timing Control

```
Increment seconds using a clock control.
If (seconds = 59)
    Reset seconds to 0
    Increment minutes by 1
    If (minutes = 59)
        -Reset minutes to 0.
        -Increment hours by 1.
        If (hours = 23)
            - reset hours to 00.
```

Pseudocoding – Control segments

- If (digit = 0) – display 0 on seven seg
- If (digit = 1) – display 1 on seven seg
- Etc Etc…

Time to Code

- Inputs: Just need a clock
- Altera board has a 50Mhz internal clock
- $50x10^6$ Hz = 1 second for this clock
- Need a counter to keep track of each hertz
- Once counter reaches 50 mil, it should increment seconds and then reset value of counter to 1.

Implementing the counter

- process(clk1)
- begin
- if(clk1'event and clk1='1') then – increment on rising edge
- count <= count+1;
- if(count = 50000000) then --50,000,000 Hz clock cycle
- clock <= not clock; --resets clock to falling edge
- count <=1; --resets counter
- end if;
- end if;
- end process;

Implementing “Clock” Functionality

```
process(clock)   --period of clk is 1 second.
begin
if(clock'event and clock='1') then
   sec_ones <= sec_ones+ 1; --seconds go up
if(sec_ones = 9) then -- 09 goes to 10, 19 to 20, etc
sec_ones <=0; -- reset ones_place
sec_tens <= sec_tens + 1; --increment tens_place
```

Linking digits to the Seven Segment

```
process(sec_ones)
  begin
if sec_ones = 0 then ones_second <=
  "1000000"; -- displays 0
elsif sec_ones = 1 then ones_second <=
       "1111001"; -- displays 1
elsif sec_ones = 2 then ones_second <=
       "0100100"; -- displays 2
```

- Etc Etc…

Recall Though when implementing

- Vectors are listed with MSB first.
- 0 is used to turn part “on”.
- Every case must be accounted for if using a case statement, but can circumvent this using if-else statement.
- Comment code using “--” in VHDL to keep track of what you’re doing.

Once Code is Done

- Compile just the code,not a full project, to see if any syntax rule have been violated.

Processing – start – start analysis & elaboration.

Once code is done open pin planner and start to map pins to inputs/outputs.

Pin Planner

- Each of the vector segments corresponds to a segment of the display.
- Hours controlled by two leftmost displays. Minutes by middle two. And seconds by the right most.

Verify the Design

- *Direct your attention to the board please*

Once Design is Implemented

- If it works correctly…Congratulations!!!
- If there is a bug, the fun process of debugging begins.
- Follow through logic of code on paper to see if train of thought is correct.

Now demonstration