Step 1: Create a High-Level State Machine

Let's consider each step of the RTL design process in more detail

Step 1

- Soda dispenser example
- Not an FSM because:
  - Multi-bit (data) inputs a and s
  - Local register tot
  - Data operations tot=0, tot<s, tot=tot+a

- Useful high-level state machine:
  - Data types beyond just bits
  - Local registers
  - Arithmetic
  - Equations/expressions

Example of how to create a high-level state machine to describe desired processor behavior
Laser-based distance measurement - pulse laser, measure time T to sense reflection
Laser light travels at speed of light, 3*10^8 m/sec
Distance is thus D = T sec * 3*10^8 m/sec / 2

Laser-based Distance Measurer

- Used in adaptive cruise control systems for automobiles
- Maintain particular "safe" distance between car in front
  - May need to decrease cruise speed
  - Need to measure the distance to the car in front

Inputs/outputs

- B: bit input, from button to begin measurement
- L: bit output, activates laser
- S: bit input, senses laser reflection
- L: 16-bit output, displays computed distance
Step 1 Example: Laser-Based Distance Measurer

Step 1: Create high-level state machine
- Begin by declaring inputs and outputs
- Create initial state, name it S0
  - Initialize laser to off (L=0)
  - Initialize displayed distance to 0 (D=0)

Add another state, call S1, that waits for a button press
- B' – stay in S1, keep waiting
- B – go to a new state S2

Add a state S2 that turns on the laser (L=1)
Then turn off laser (L=0) in a state S3

Stay in S3 until sense reflection (S)
To measure time, count cycles for which we are in S3
- To count, declare local register Dctr
- Increment Dctr each cycle in S3
- Initialize Dctr to 0 in S1. S2 would have been O.K. too

Once reflection detected (S), go to new state S4
- Calculate distance
  - Assuming clock frequency is 3x10^8, Dctr holds number of meters, so D=Dctr/2
- After S4, go back to S1 to wait for button again

Step 2: Create a Datapath

Datapath must
- Implement data storage
- Implement data computations
- Look at high-level state machine, do three substeps
  - (a) Make data inputs/outputs be datapath inputs/outputs
  - (b) Instantiate declared registers into the datapath (also instantiate a register for each data output)
  - (c) Examine every state and transition, and instantiate datapath components and connections to implement any data computations

Instantiate: to introduce a new component into a design.
Step 2 Example: Laser-Based Distance Measurer

(a) Make data inputs/outputs be datapath inputs/outputs
(b) Instantiate declared registers into the datapath (also instantiate a register for each data output)
(c) Examine every state and transition, and instantiate datapath components and connections to implement any data computations

Step 2 Example: Laser-Based Distance Measurer

(c) (continued) Examine every state and transition, and instantiate datapath components and connections to implement any data computations

Step 2 Example Showing Mux Use

(a) Introduce mux when one component input can come from more than one source

Step 3: Connecting the Datapath to a Controller

- Laser-based distance measurer example
- Easy - just connect all control signals between controller and datapath

Step 4: Deriving the Controller’s FSM

- FSM has same structure as high-level state machine
- Inputs/outputs all bits now
- Replace data operations by bit operations using datapath

Step 4: Deriving the Controller’s FSM

- Using shorthand of outputs not assigned implicitly assigned 0
Step 4

Implement FSM as state register and logic (Ch3) to complete the design.

Inputs: B, S
Outputs: L, Dreg_clr, Dreg_d, Dctr_clr, Dctr_cnt

S0
S1
S2
S3
S4

L = 0 (lower 0)
Dreg_clr = 1 (laser off)
Dctr_cnt = 1 (laser off)
Dctr_clr = 1 (clear count)
Dreg_ld = 1 (load Dreg with Dctr/2)

L = 1 (lower 1)
Dreg_clr = 1 (laser on)
Dctr_cnt = 0 (stop counting)

Dreg_d = 1 (clear Dreg)
Dctr_d = 1 (count up)

Design Challenge

- Design a 4-bit up-counter using the RTL design process.
- Design a high-level state machine for a 4-bit up-counter with count control input cnt, clear input clr, and a terminal count output tc.
- Use the RTL design method to convert the high-level state machine to a controller and a datapath.
- Use a register and adder in the datapath, not a counter itself.
- Design the datapath and controller to structure.

Due:
- Next Lecture (Monday, October 17)
- Extra Credit (Homework)
  - 2 points