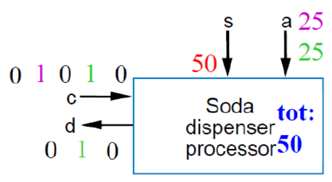
ECE 4/530 Activity 03 – Soda Machine Design

Design Task

Design a Verilog implementation on FPGA for a soda-dispensing machine as specified below using finite state machines. Elaborate the design using one hot and observe the resource utilization. You might try other encoding styles and compare the resource utilization.

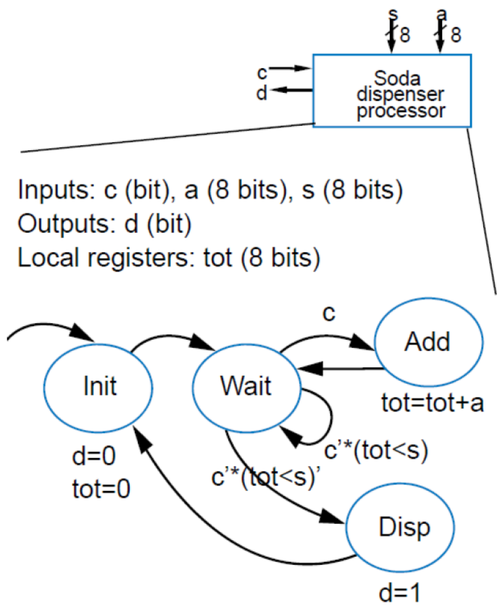
Step1: Specifications

- *c*: bit input, 1 when coin is deposited
- *a*: 8-bit input having value of deposited coin
- *s*: 8-bit input having cost of a soda
- *d*: bit output is set to 1 when total value of deposited coins equals or exceeds cost of a soda



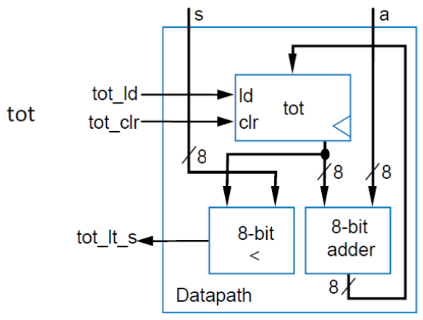
Step 2: High-level State Machine

- Declare local register *tot*
- **Init** state: Set d=0, tot=0
- **Wait** state: wait for coin
  - If see coin, go to **Add** state
- Add State: update total value
  - Tot = tot + a
    - Remember a is present coni's value
  - Go back to Wait state
- In **Wait** state, if tot >= s, go to Disp State
- In **Disp** state – Set d-1 (dispense soda)
  - Return to **Init** state



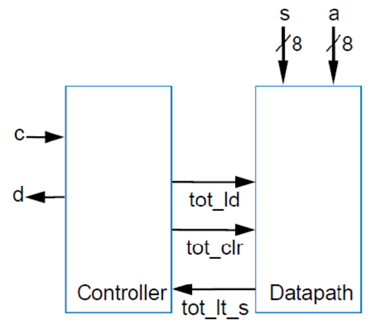
Step 3: Datapath Design and Operation

- Need *tot* register
- Need 8-bit comparator to compare s and a
- Need 8-bit adder to perform tot = tot + a
- Connect everything
- Create control input/output



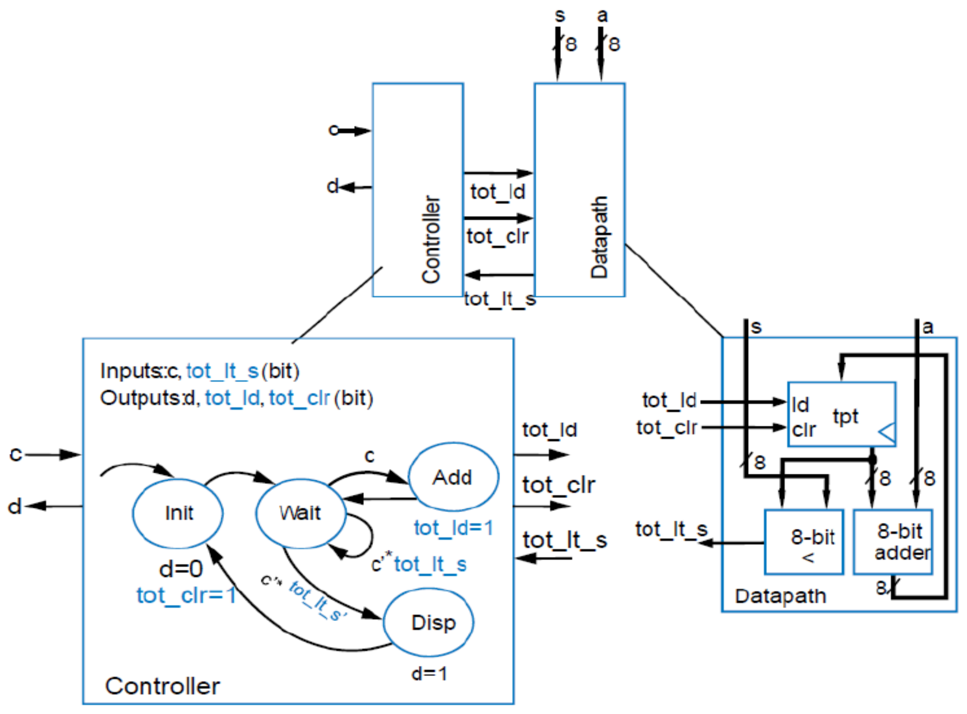
Step 4: Connect Datapath to Controller

- ➢ Controller's inputs
  - External input c (coin detected)
  - Input from datapath comparator's output, which we named tot_lt_s
- ➢ Controller's outputs
  - External output d (dispense soda)
  - Outputs to datapath to load and clear tot register



Step5: Drive the controller's FSM

- ➢ Same states and transitions as high-level FSM
- ➢ Set/read data path control signals for all datapath operations and conditions



Procedure

- *Design* the Soda machine in Verilog to match the description given using one-hot encoding and a low-active reset. The design should have the following modules:
  1. soda_machine_fsm module – This module should contain the FSM design in three procedural blocks: fsm_register, next_state_logic, and output_logic.
  2. soda_machine_datapath module – This module is a clocked module that contains the supporting components needed for the FSM operation as shown in the block diagram.
  3. soda_machine_top module – This module is the top module of the design that contains instantiations for the two other modules and the connection between them.

- *Synthesis* the design and examine the resource utilization. Make sure the design contains no latches. Look at the number of FFs and make sure they match what was intended for the design – Explain the utilization in your report

- *Simulation*: write a simple testbench to check the functionality of your design according to the following specifications:
  1. Declare a constant for the clock period, and others for the value of nickel, dime, and quarter.
  2. Generate a clock with a 10ns period.
  3. Set the cost to $1.50
  4. Generate the reset. Offset it from clock edges. Start as not-active, activate it for a clock period then deactivate it.
  5. Generate the stimulus by adding 4 quarters, 5 dimes, and 3 nickels.
  6. Reset the system and Change the price of a soda to $2.00. Enter amount of $2.50 using a different combination of the three coins and evaluate the behavior of the design.

- *Wrapper:* design the wrapper to include the soda machine module and all other components that interface the switches, pushbuttons, OLED display and synthesizers, edge detectors, and frequency divider according to the information provided in the implementation section

- *Implementation*:
  1. Connect the OLED display to the board as it will be explained later. It is important to make sure that the display works fine before you proceed with the implementation. The display runs at the board clock frequency while the FSM should run on 10KHz.
  2. Use the following Zybo board resources to emulate the Soda machine functionality
     - ➢ PB0 – (low-active) Reset
     - ➢ PB1 – accumulate the value of the cost of soda as indicated by [SW2 – SW0], each push will add these values.
     - ➢ PB2 – set deposited coin value, one at a time, using [SW2 – SW0], each push will pass this value to the corresponding soda machine inputs.
     - ➢ PB3 – Accept total value for the cost of the soda and deliver it to the soda machine corresponding input.
     - ➢ SW0 – indicates 5c value.
     - ➢ SW1 – indicates 10c value.
     - ➢ SW2 – indicates 25c value.
     - ➢ SW3 – select price/coin mode (1: price mode, 0: coin entry mode)
     - ➢ Use the LEDS at your leisure to help you debug.

  OLED Operation:
  - ❖ After reset, the OLED should display "*Ready*"
  - ❖ After you set the soda price, the OLED should display "*Cost $xx.xx*"
  - ❖ After each coin is deposited, the OLED should display the amount deposited as "*xxxxC*"
  - ❖ After depositing coins is done, the OLED should display the total deposit as "tot $xx.xx"
  - ❖ After the soda is dispensed, the display should displaythe word "*DISPENSED*"

  Operational Example
  a. Press PB0 for a low-active reset
  b. Set SW3 = 1 to initiate price mode to set the soda price.
  c. Use [SW2 – SW0] to set the Soda price, each push of PB1 will add the values.
  d. After finishing the price setting, press PB3.
  e. After price of soda is set, go the coin entry mode by switching SW3 = 0
  f. Select any of [SW2 – SW0] one at a time to deposit coins. Press PB2 to deposit the coin into the machine

- Submission: Prepare a report according to the format posted on Blackboard. The report should include annotated simulation waveforms, elaborated schematic results, resource utilization extracted from the synthesis report with explanation on how they are related to your design. Include the following with your submission: all Verilog files for design and testbench(s) – these files should be commented explain the results and the test cases along with expected results, check off video. Zip your files to a file called "*Lastname_Firstname_A03.zip"*

- Check off (separate from submission): You can check off during a class period or request a private session. Here is the procedure:

    1. Reset – Display ready
    2. Set Soda price (3Q – 4D – 5N) – Cost Display
    3. Enter coins in sequence (2N – 3D – 4Q)
    4. Display each coin value
    5. Display total
    6. Final Display dispensed