MILESTONE 3

Accumulator Based Vending Machine

Objectives

- Interfacing CPLD with simple input/output devices
- Design of high-level modules based on specifications
- Building a circuit on a strip board

3.1 Overall Description

Implement a vending machine simulator illustrated in Fig. 3.1 on CPLD and circuit board. The major components are:

Table 3.1: List of modules for vending machine.

Module#	Module	Source
1	Adder	Milestone 1
2	Accumulator	Milestone 1 & Chapter 8
3	Knight Rider lights	Milestone 2
4	Prescaler	Milestone 2 & Subsection 9.4.3
5	Comparator	Section 6.4
6	Binary-to-BCD converter	Section 6.2
7	Input filter	Section 9.3
8	Selector	New

The vending machine is based on a 4-bit accumulator which keeps tracks of how much money has been deposited. The accumulator value equals multiples of 10 sens. It has two button inputs: button 1 for 20 sen, button 2 for 50 sen. Pressing a button adds 2 or 5 to the accumulator.

The 7 segment display shows the current value of accumulator in BCD up to 15. When the accumulator contains 12 or more, the Knight Rider LEDs (from Milestone 2) will display one round of lights (as opposed to unlimited rounds in Milestone 2).

Implementation steps:

1. Prepare all modules in Table 3.1. Simulate all independently, except the prescaler (module 4).
2. Integrate/combine all modules, except the prescaler, and simulate.
3. Prepare an *I/O board* consisting of two seven-segment displays, and two switches and the necessary resistors. It is highly recommended to use a stripboard for the 7-segment display to avoid connectivity problems. The switch can be on the same stripboard or on a breadboard; it is less critical due to fewer wires required.
4. Add the prescaler, connect the I/O board with the CPLD and program the CPLD.

The following sections describes the module functions, grouped by level of reuse.

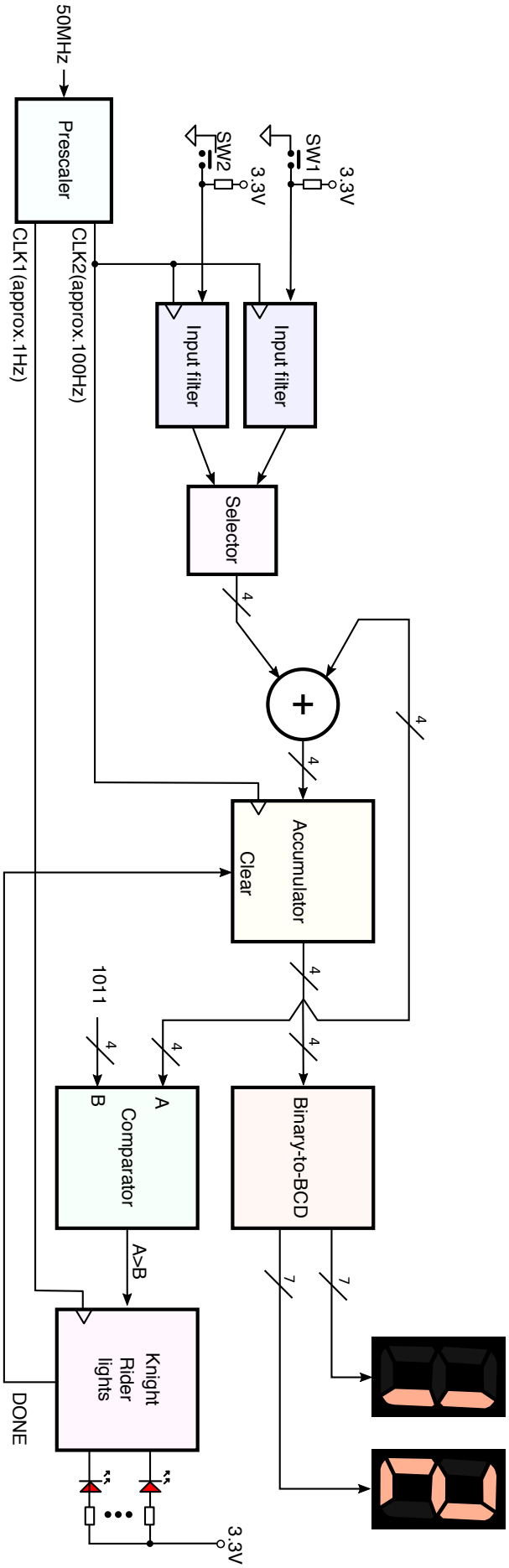

Figure 3.1: Block diagram of vending machine

3.2 Modules from Previous Milestones

3.2.1 Adder

The 4-bit ripple carry adder from Milestone 1 can be used directly without modification. No simulation is necessary for this circuit as it is completely reusable.

3.2.2 Accumulator

The procedure to build the accumulator from Milestone 1 can be followed. However, the function is different. In this accumulator, the control line is called *Clear*.

Clear	Function
0	Loads data at D0-D3
1	Register is cleared (Q←0) on next clock edge

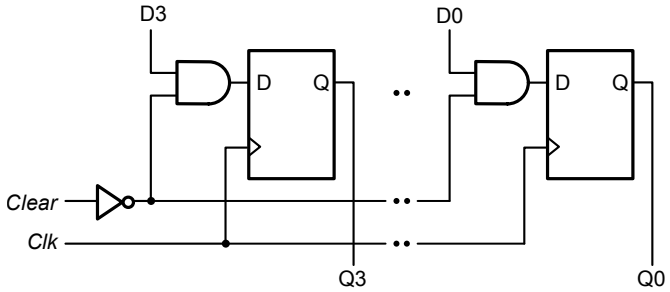

Figure 3.2: First and last cells of the register with synchronous clear.

Build and simulate one cell first. Then combine 4 cells to build the accumulator. Show the compilation report, schematic and simulation of the 4-cell accumulator only.

3.2.3 Knight Rider Lights

The Knight Rider lights from Milestone 2 runs forever. Make two modificaitons:

- Add a control signal circuitry to enable/disable the lights, so that the LEDs runs only when this input is high.
- Add a status signal (DONE) that indicates when the lights have run a complete round.

Simulate to check if the circuit can be enabled/disabled on demand, and the DONE is activated at the correct time.

3.2.4 Prescaler

This prescaler generates two outputs:

- CLK1: a signal approximately 1 Hz. This speed is required so that the Knight Rider light can run slowly and visible to humans.
- CLK2: a signal approximately 100Hz. This speed is required to eliminate contact bounce that happens when a button is pressed.

Implement the prescaler using a 26-bit up counter from LPM_COUNTER. Use bit 25 as CLK1 and another bit as CLK2.

Do not simulate the prescaler. Do not integrate it with the main system yet. We will add it at the final construction stage.

3.3 Modules based on Circuit in Textbook

3.3.1 Comparator

The comparator gives a high output when the accumulator contains a value more than 11. Choose a combinational or iterative comparator from Section 6.4. For the comparator, show the compilation reports, schematics and simulation.

3.3.2 Binary-to-BCD Converter

The input to the converter is a binary input 0000 to 1111. The outputs are 7 lines to the tens digits and 7 lines to the ones digit. Inside the converter, the binary input is first converted to BCD to produce a two digit BCD result. Use one unit of the conditional add by 3 adder (cadd3) module in Section 6.2 to convert the 4-bit binary number to two BCD digits. The two BCD digits are then connected to two units of 7447 modules before connecting to 7-segment displays.

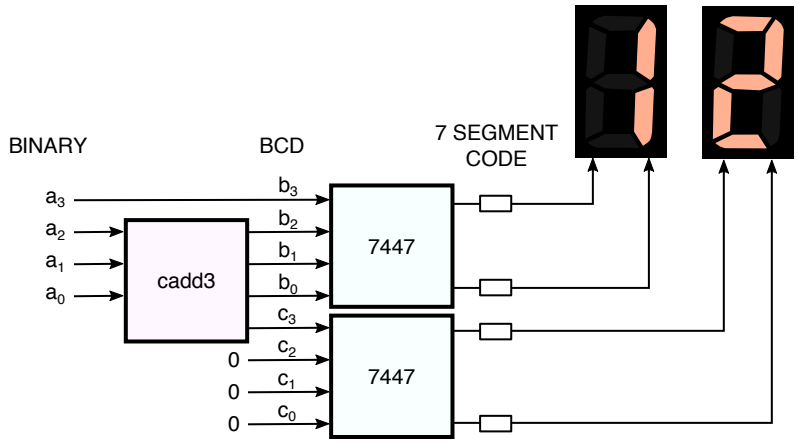

Figure 3.3: Binary-to-BCD converter with BCD-to-7-segment decoder combination.

3.3.3 Input Filter

Button presses produce glitches lasting 1-10 milliseconds which are detected as multiple presses if a fast clock is used. A slow clock at the filter removes the glitches. The function of the input filter is to convert a button press into a single clean pulse.

Note that if the clock is too slow, the user must press the button for an unreasonable amount of time before the button press is detected. The frequency of clock input is not fixed. Change it to a suitable frequency depending on the switches used.

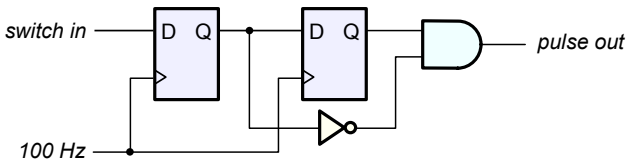

Figure 3.4: Input filter.

You will need two copies of the filter, one for each button. The buttons are active low, meaning that it is high when not pressed, and low when pressed. The circuit is Fig. 3.4 produces a single pulse when a high-to-low transition is detected at its input.

3.4 Selector

The adder add the current value of the accumulator (first input) and the value chosen by the selector circuit (second input). The selector chooses a value to place at the second adder input based on this table:

Switch Pressed	Second adder input	Accumulator function
None	0000	Hold
SW1	0010	ACC ← ACC + 2
SW2	0101	ACC ← ACC + 5

The selected value is presented for exactly one clock cycle for every button press. This is a combinational circuit which your team must design.

3.5 Obtaining Maximum Credit

For each new module:

- Compile until there are no errors.
- Test/simulate using carefully selected data inputs.
- Convert to symbol
- Record a very short video of the entering the design (while using schematic editor or Verilog editor).