RTL_EXERCISE_1 BOUND FLASHER

Author	Nguyễn Hữu Trung Nhân – Student ID: 1752392
Date	06 August 2020
Version	3.0

Contents

1. Interface

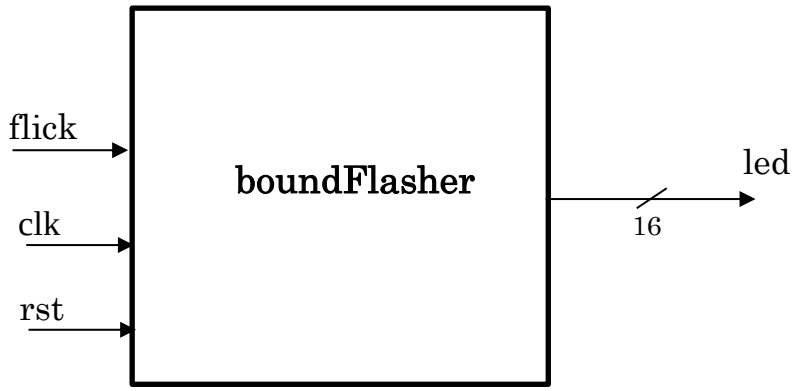


Figure 1: The figure of Bound Flasher System

Signal	Width	In/Out	Description
flick	1	In	Asynchronous input signal; When the output (led) are turned OFF gradually, at led[5] or led[0], if flick = 1, then the output will turn on gradually again to the max led of the previous state, except the final state.
clk	1	In	Clock signal; The function operates state's transition at the rising edge of the clock signal.
rst	1	In	Reset signal; LOW-ACTIVE Reset = 0: System restarts to Initial State; "rst" is asynchronous signal (does not depend on "clk" signal).
led	16	Out	16-bit led from led[0] to led[15] ; led[0] is the Least Significant Bit ; led[15] is the Most Significant Bit.

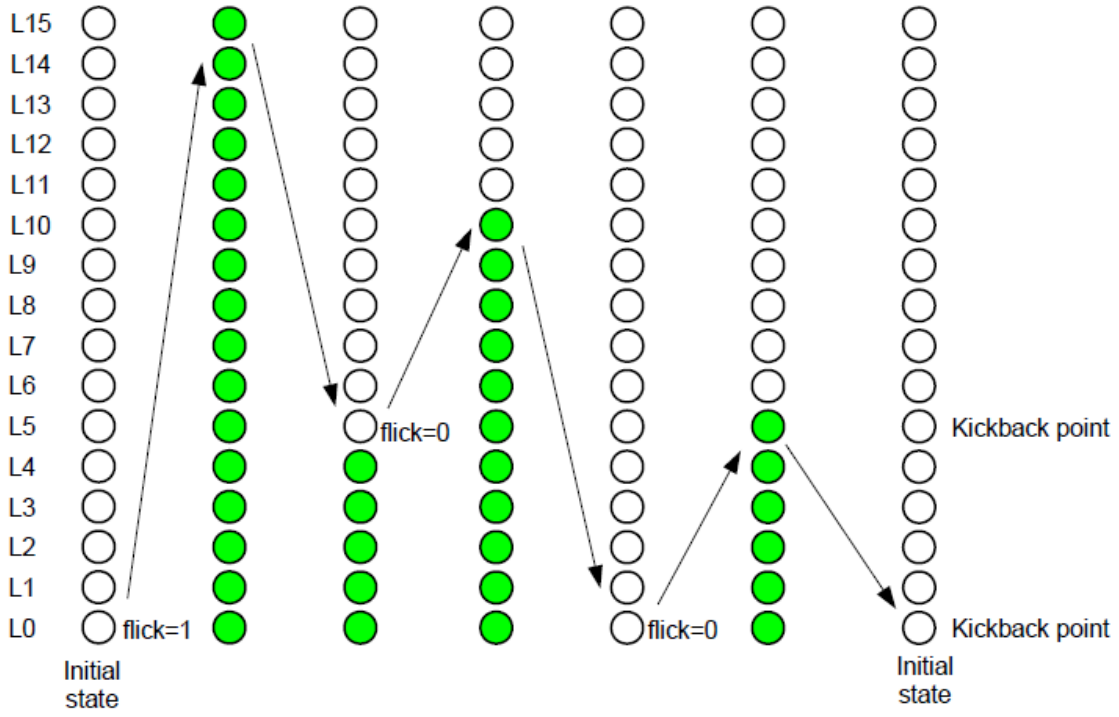
Table 1: Description of signals in Bound Flasher

2. Functional implementation.

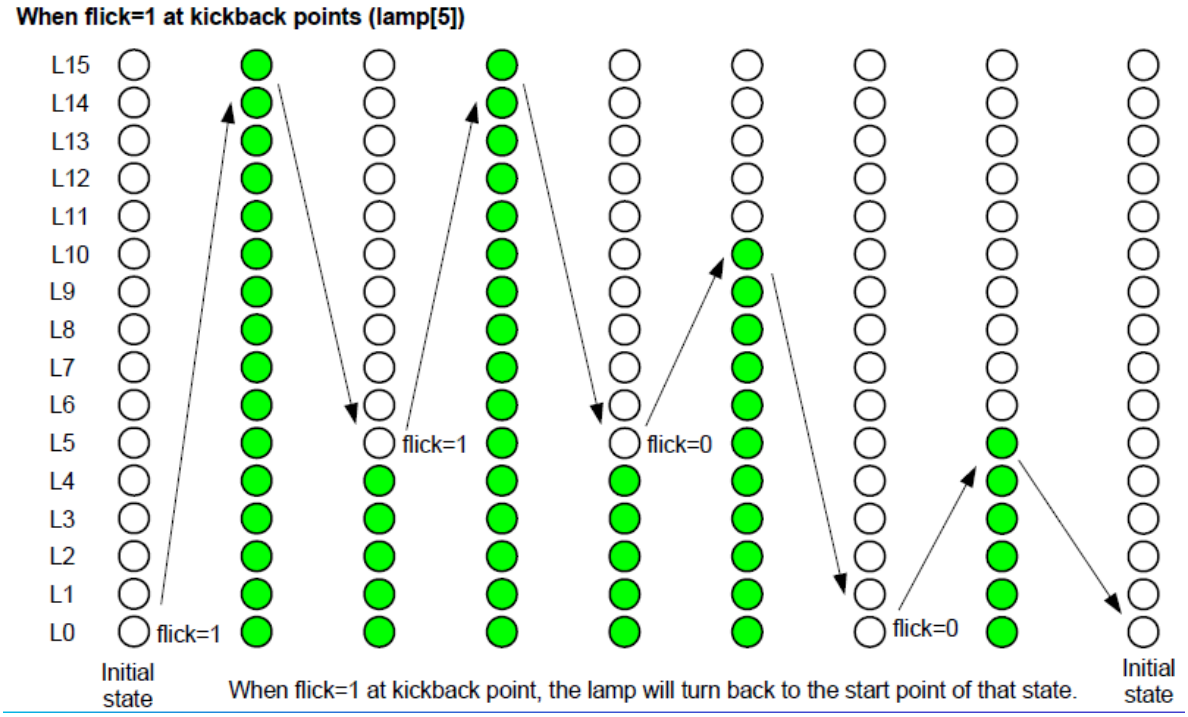
- Implement a 16-bits LEDs system
- System's Operation base on three input signal
  - Reset
  - Clock
  - Flick
- The system specification
  - Clock signal is provided for system inspire of function status. The function operate state's transition at positive edge of the clock signal.
  - Reset signal:
    - LOW-ACTIVE Reset = 0: System is restarted to Initial State.
    - HIGH-ACTIVE Reset = 1: System is started with initial state.
- Flick signal: special input for controlling state transfer.
- At the initial state, all lamps are OFF. If flick signal is ACTIVE, the flasher start operating:
  - The lamps are turned ON gradually from LEDs [0] to LEDs [15].
  - The LEDSs are turned OFF gradually from LEDs [15] to LEDs [5].
  - The LEDSs are turned ON gradually from LEDs [5] to LEDs [10].
  - The LEDSs are turned OFF gradually from LEDs [10] to LEDs [0].
  - The LEDSs are turned ON gradually from LEDs [0] to LEDs [5].
  - Finally, the LEDs s are turned OFF gradually from LEDSs [5] to LEDSs [0], return to initial state.
- Additional condition: At each kickback point (LEDs [5] and LEDs [0]), if flick signal is ACTIVE, the LEDs will go back and repeat that STATE. For simple, kickback point is considered only when the LEDs s are turned OFF gradually, except final state.

– Some insulations:

- When flick = 0 at kickback points



- When flick = 1 at kickback points (lamp[5])



3. Internal implementation.

3.1. Overall.

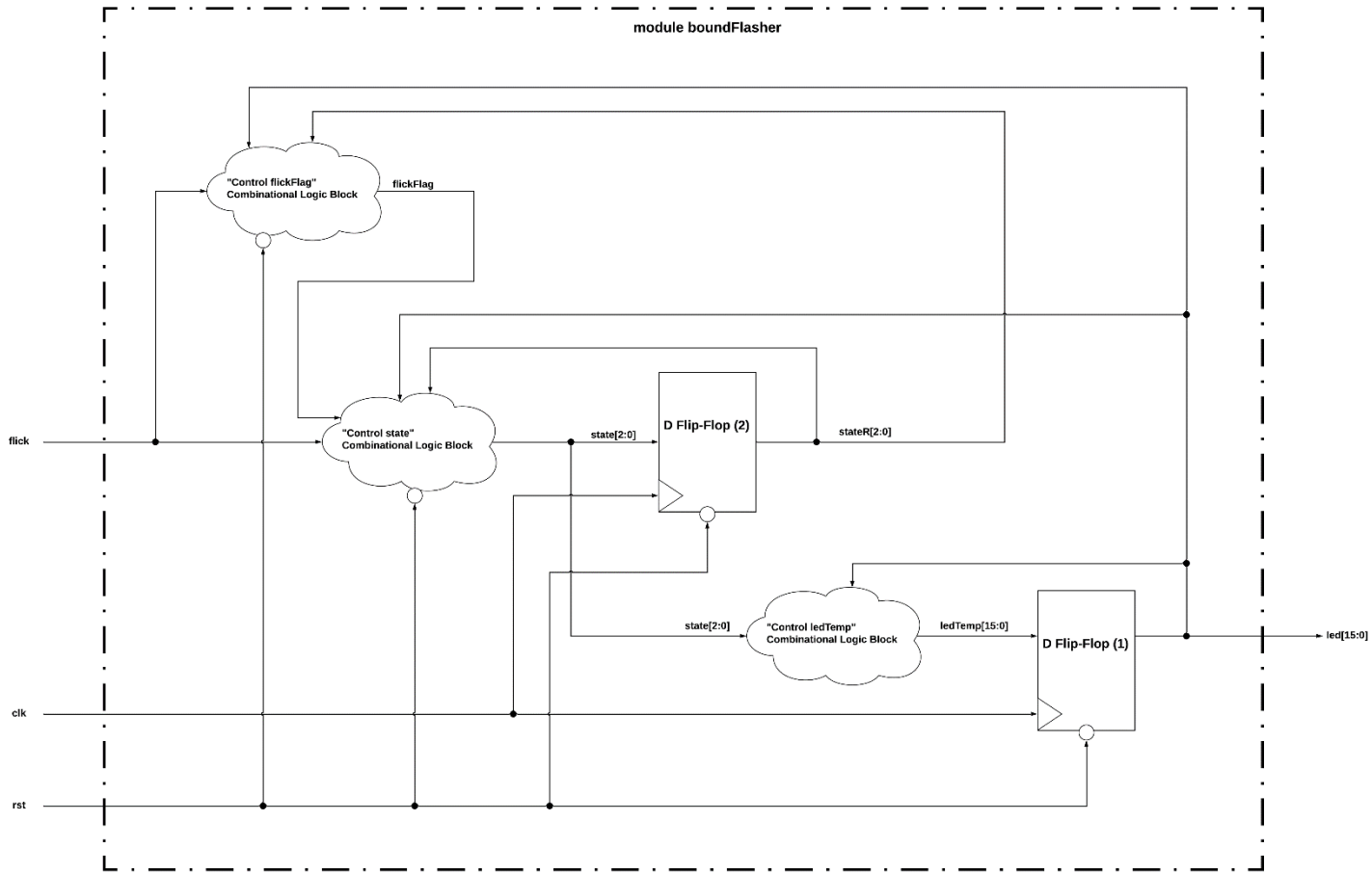


Figure 3.1: Block diagram of Bound Flasher

Block	Description
D Flip-Flop (1)	Synchronize the input signal (ledTemp[15:0]) with the rising edge clock. Using the clock signal to increase or decrease the 16-bit led. The rst (Reset) signal is the low-active asynchronous signal (whenever rst == 0, all the led will be off immediately).
"Control ledTemp" Combinational Logic block	Using the input signals (state[2:0] and led[15:0]) to control the ledTemp[15:0] signal.
D Flip-Flop (2)	Synchronize the input signal (state[2:0]) with the rising edge clock. Using the clock signal to change the output signal (stateR[2:0]) (stateR means "state Real"). The rst (Reset) signal is the low-active asynchronous signal (whenever rst == 0, then the "Real state" will be reset to Initial State immediately).
"Control state" Combinational Logic block	If "flick signal" is 1 at "kick-back points", "state" will be changed to previous "state"; If "flick signal" is 1 at "Initial State", "state" will be changed to "State 1". The rst (Reset) signal is the low-active asynchronous signal (whenever rst == 0, state will be reset to Initial State immediately).
"Control flickFlag" Combinational Logic block	Using a flag (called flickFlag) to check if there is a flick signal (flick == 1) at "kick-back points". If there is a flick signal (flick == 1) at any point of the "kick-back points", this 1-bit flag will be 1 (flickFlag = 1). The changing of "stateR" signal (state Real) and the led will help to set this flag back to 0. The rst (Reset) signal is the low-active asynchronous signal (whenever rst == 0, flickFlag = 0 immediately).

Table 3.1: Block diagram of Bound Flasher Description

3.2. State Machine

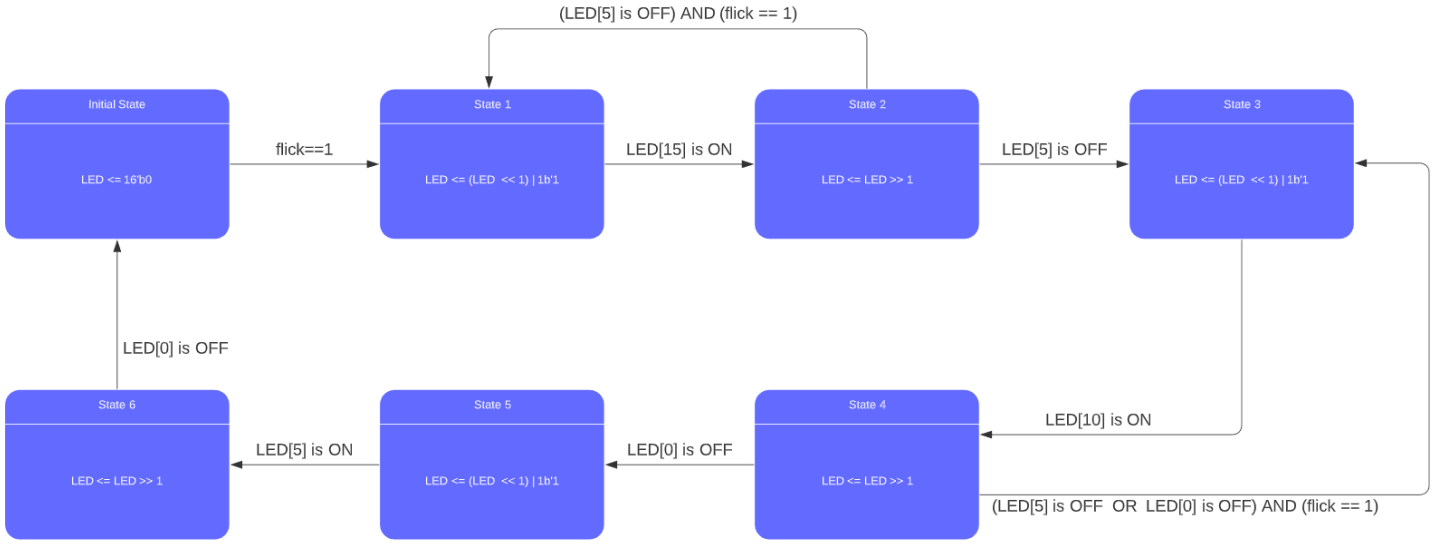


Figure 3.2: State Machine of Bound Flasher

flick	Asynchronous input signal; When the output (LED) are turned OFF gradually, at LED[5] or LED[0], if Flick = 1, then the output will turn on gradually again to the max LED of the previous state, except the final state.
LED	16-bit output. LED[0] is the Least Significant Bit. LED[15] is the Most Significant Bit.

Table 3.2: Variable name of State machine

Initial State	All 16 bits of the LED are zero (LED[0] to LED[15] are OFF). At the very first beginning, state = Initial State. If signal flick = 1, then state = state 1 (moving to state 1).
State 1	LEDs are turned ON from LED[0] to LED[15] gradually, from LED[0] = 1 (the Least Significant Bit) to LED[15] =1 (the Most Significant Bit) by using this below method: "Shift all the bit to the left and then OR with 1" If LED[15] is ON, then state = state 2 (moving to state 2).
State 2	LEDs are turned OFF from LED[15] to LED[5] gradually. From LED[15]=0 (the Most Significant Bit) to LED[5]=0 by using this below method: "Shift all the bit to the right" If LED[5] is OFF and signal flick = 1, then state = state 1 (return to the previous state). Else if only LED[5] is OFF, then state = state 3.
State 3	LEDs are turned ON from LED[5] to LED[10] by using this below method: "Shift all the bit to the left and then OR with 1" If LED[10] is ON, then state = state 4 (moving to state 4).
State 4	LEDs are turned OFF from LED[10] to LED[0] by using this below method: "Shift all the bit to the right" If (LED[5] is OFF or LED[0] is OFF) and signal flick = 1, then state = state 3 (return to the previous state) Else if only LED[0] is OFF, then state = state 5.
State 5	LEDs are turned ON from LED[0] to LED[5] by using this below method: "Shift all the bit to the left and then OR with 1" If LED[5] is ON, then state = state 6.
State 6	LEDs are turned OFF from LED[5] to LED[0] by using this below method: "Shift all the bit to the right" If LED[0] is OFF, then state = Initial State.

Table 3.3: state name of State machine

4. History

Date	Author	Modified part	Description
2020/July/15	Nguyễn Hữu Trung Nhân	All	New creation
2020/July/26	Nguyễn Hữu Trung Nhân	Table 1 and Figure 1; Figure 3.1	Page 2, Table 1 and Figure 1: re-check module name and signal name to match with the RTL code. Page 5, Figure 3.1: draw again the block diagram to match with the RTL code.
2020/August/06	Nguyễn Hữu Trung Nhân	Figure 3.1; Table 3.1	Page 5, Figure 3.1: design again the block diagram to match with the RTL code. Page 6, Table 3.1: re-write the description for the block diagram.