|  |
| --- |
|  |
| **RTL\_EXERCISE\_1 BOUND FLASHER** |
|  |
| |  |  | | --- | --- | | Author | 1. Dương Gia An  2. Thiều Quang Trung  3. Lê Nguyễn Gia Nghi  4. Đoàn Việt Tú | | Date | 2022/04/22 | | Version | 1.3 | |
|  |
|  |

|  |
| --- |
|  |

Contents

[1. Interface 2](#_Toc478478267)

[2. Functional implementation. 3](#_Toc478478268)

[3. Internal implementation. 5](#_Toc478478269)

[3.1. Overall. 5](#_Toc478478270)

[3.2. State Machine 6](#_Toc478478271)

[4. History 7](#_Toc478478272)

# 1. Interface

|  |
| --- |
| **Module bounder\_flasher**  reset  flick  led[15:0]  clk  16 |
| Figure 1: the figure of Bound Flasher System |

|  |  |  |  |
| --- | --- | --- | --- |
| Signal | Width | In/Out | Description |
| flick | 1 | In | Button, Active High signal  When the output led turns on gradually, at LED[5] or LED[10] if the flick=1 the output will turn back to the previous state. |
| clock | 1 | In | Operate state’s transition at the rising edge of the clock signal. |
| reset | 1 | In | Button, active low  When reset is low, return to initial state. |
| led | 16 | Out | Led display depends on the state. |

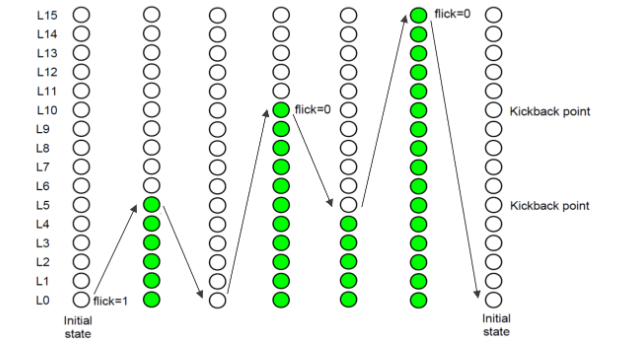
Table 1: Description of signals in Bound Flasher

# 2. Functional implementation.

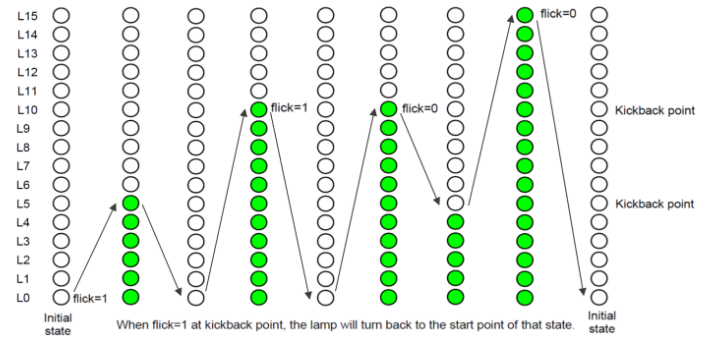
* Implement the bound flasher system with 16-bit LEDs.
* The system’s operation is based on three input signals:
  + Reset
  + Clock
  + Flick
* The system’s specifications:
* A clock signal is provided for system inspire of function status. The function operates state’s transition at positive edge of 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 LEDs are OFF. If flick signal is ACTIVE (set 1), the flasher starts operating:

1. The LEDs are turned ON gradually from LEDs [0] to LEDs [5].
2. The LEDs are turned OFF gradually from LEDs [5] (**max**) to LEDs [0] (**min**).
3. The LEDs are turned ON gradually from LEDs [0] to LEDs [10].
4. The LEDs are turned OFF gradually from LEDs [10] (**max**) to LEDs [5] (**min**).
5. The LEDs are turned ON gradually from LEDs [5] to LEDs [15].
6. Finally, the LEDs are turned OFF gradually from LEDs [15] to LEDs [0], return to initial state.

* **Additional condition:**
* At each kickback point (LEDs [5] and LEDs [10]), if flick signal is ACTIVE, the LEDs will turn OFF gradually again to the min LEDs of the previous state, then continue operation as above description.
* For simple, kickback point is considered only when the LEDs are turned ON gradually, except the first state.
* Some insulation:
* When flick = 0 at kickback points

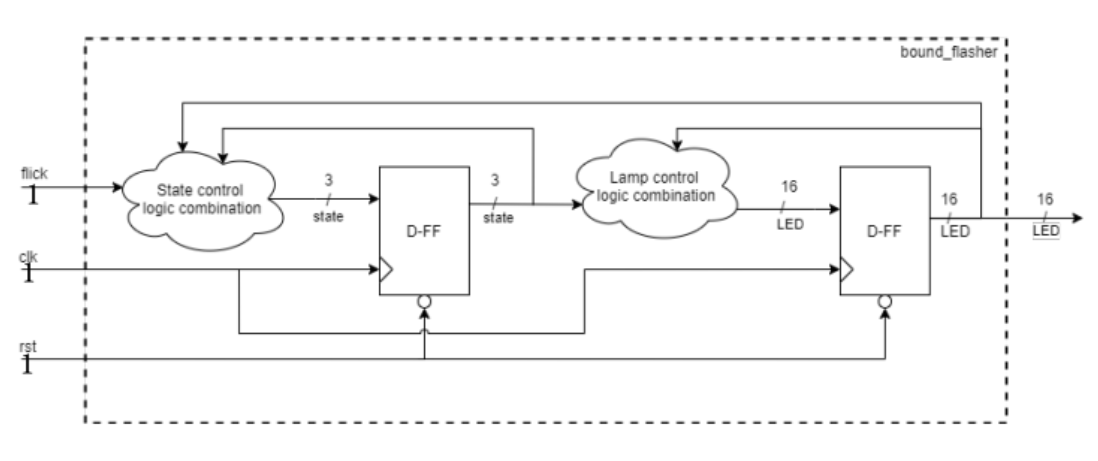


* When flick = 1 at kickback points (LEDs [5])



# 3. Internal implementation.

## 3.1. Overall.

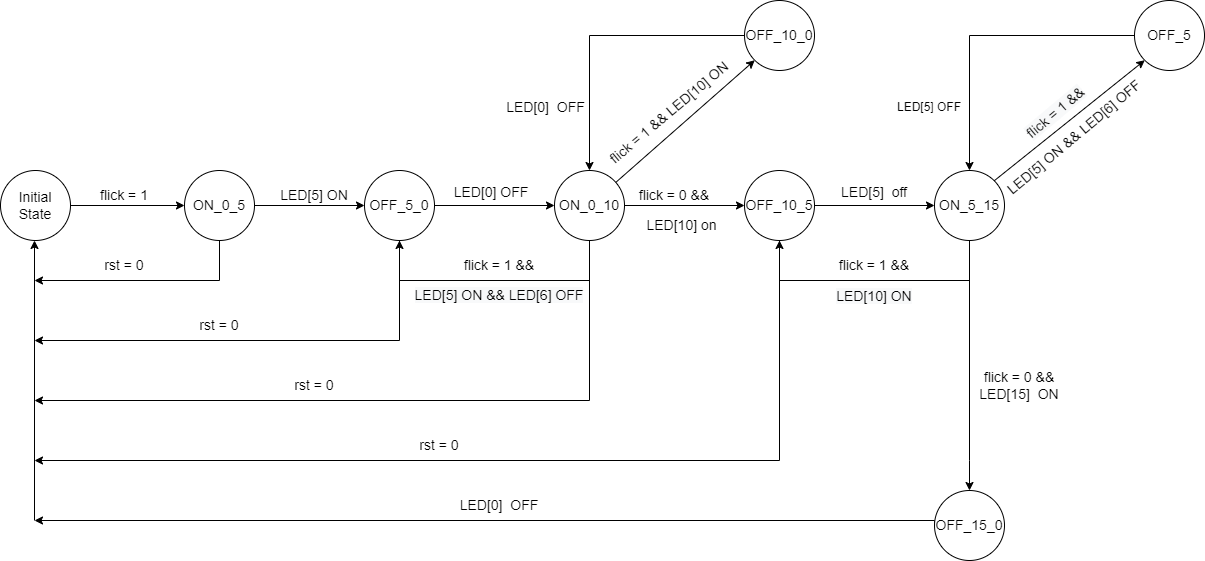


**Figure 3.1: Block diagram of Bound Flasher**

|  |  |
| --- | --- |
|  | |
| **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 lowactive asynchronous signal (whenever rst == 0, all the led will be off immediately). | |
| “Control led” 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 Flick” 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**

|  |  |
| --- | --- |
| **Variable Name** | **Explanation** |
| flick | Asynchronous input signal. At kickback points (LED[5] & LED[10]), if flick signal = 1, the LEDs will turn OFF gradually again to the **min** LED of the previous state. |
| LED | 16-bit output, LED[0] is the Least Significant Bit, LED[15] is the Most Significant Bit |
| rst | Active low, whenever reset signal = 0, the system immediately returns to **Initial State.** |

**Table 3.2: Variable Name of State machine**

|  |  |
| --- | --- |
| **State Name** | **Explanation** |
| Initial State | All LEDs are OFF (from LED[0] to LED[15]). If flick signal = 1 then moving to **State ON\_0\_5**. |
| ON\_0\_5 | The LEDs are turned ON gradually from LED[0] to LED[5]. If LED[5] is ON then moving to **State OFF\_5\_0.** |
| OFF\_5\_0 | The LEDs are turned OFF gradually from LED[5] to LED[0]. If LED[0] is OFF then moving to **State ON\_0\_10.** |
| ON\_0\_10 | The LEDs are turned ON gradually from LED[0] to LED[10]. When LED[5] is ON, flick signal =1 but LED[6] is OFF, then moving back to **State OFF\_5\_0**. If (LED[10] is ON && flick signal = 0) then moving to **State OFF\_10\_5**. Otherwise, if (LED[10] is ON && flick signal = 1) then moving to **State OFF\_10\_0.** |
| OFF\_10\_0 | The LEDs are turned OFF gradually from LED[10] to LED[0]. If LED[0] is OFF then moving to **State ON\_0\_10**. |
| OFF\_10\_5 | The LEDs are turned OFF gradually from LED[10] to LED[5]. If LED[5] is OFF, then moving to **State ON\_5\_15**. |
| ON\_5\_15 | The LEDs are turned ON gradually from LED[5] to LED[15]. If (LED[5] is ON && LED[6] is OFF) and flick signal = 1 then moving to **State OFF\_5**. If (LED[15] is ON && flick signal = 0) then moving to **State OFF\_15\_0**. |
| OFF\_5 | LED[5] is turned OFF. If LED[5] is OFF the moving back to **State ON\_5\_15** |
| OFF\_15\_0 | The LEDs are turned OFF gradually from LED[15] to LED[0]. If LED[0] is off, then moving to **Initial State**. |

**Table 3.3: State Name of State machine**

# 4. History

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Author | Modified part | Description |
| 16/4/2022 | An | All | New creation, share link for all members to edit report |
| 20/4/2022 | Nghi, Tu | Diagram | Finish diagram |
| 22/4/2022 | Trung, An | FSM | Finish report |