|  |
| --- |
|  |
| **RTL\_EXERCISE\_1 BOUND FLASHER** |
|  |
| |  |  | | --- | --- | | Author |  | | Date | 2022/03/04 | | Version | 1.1 | |
|  |
|  |

|  |
| --- |
|  |

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

|  |
| --- |
| 16  1  1  16  flick  rst  clk  **bound\_flasher\_sys**  LED |
| Figure 1: the figure of Bound Flasher System |

|  |  |  |  |
| --- | --- | --- | --- |
| Signal | Width | In/Out | Description |
| flick | 1 | In | Determine module’s operation, depend on current state |
| clk | 1 | In | Master clock signal, for apply pulse for module purpose |
| rst | 1 | In | To reset the module |
| LED | 16 | Out | 16 single LEDs in 1 column |

Table 1: Description of signals in Bound Flasher

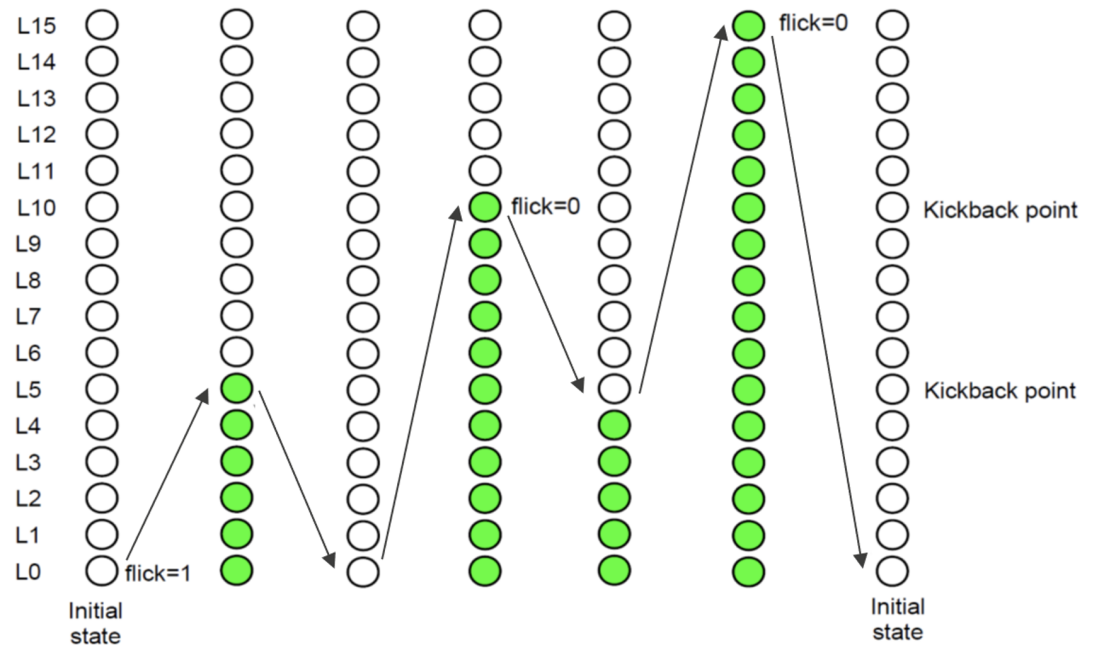
# 2. Functional implementation.

* Implement a 16-bits LEDs system
* System’s Operation base on three input signals
  + 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 lamp[0]to lamp[5]**.**
* The lamps are turned OFF gradually from lamp[5] **(max)** to lamp[0] **(min)**.
* The lamps are turned ON gradually from lamp[0]to lamp[10].
* The lamps are turned OFF gradually from lamp[10] **(max)** to lamp[5] **(min)**.
* The lamps are turned ON gradually from lamp[5] to lamp[15].
* The lamps are turned OFF gradually from lamp[15] to lamp[0], return to initial state.

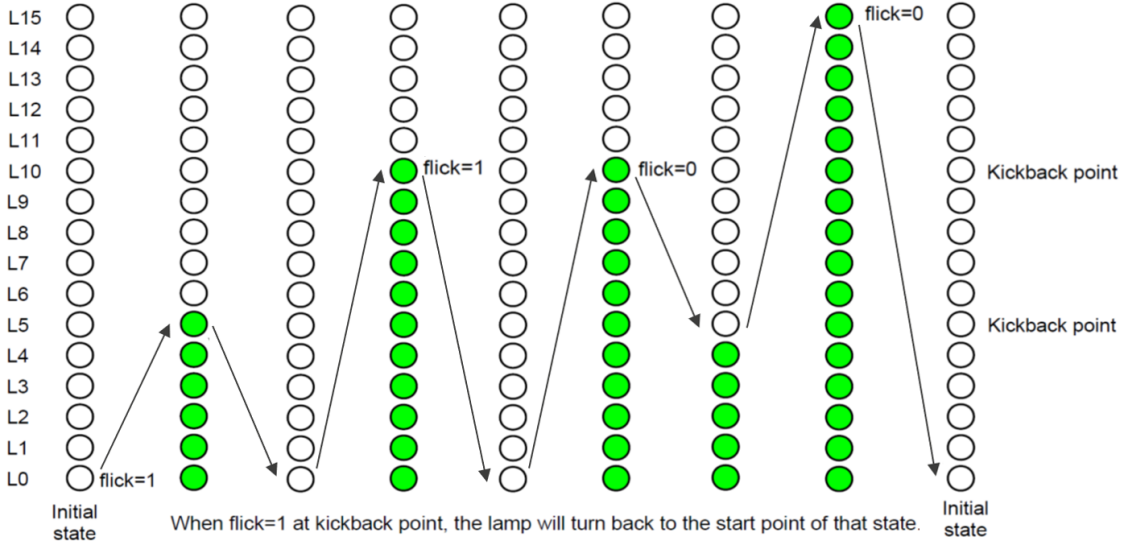
At each kickback point (lamp[5] and lamp[10]), if flick signal is ACTIVE, the lamps will turn OFF gradually again to the min lamp of the previous state, then continue operation as above description.

For simple, kickback point is considered only when the lamps are turned ON gradually, except the first state.

* Some insulations:
* When flick = 0 at kickback points



* When flick = 1 at kickback points (lamp[10])



# 3. Internal implementation.

## 3.1. Overall.

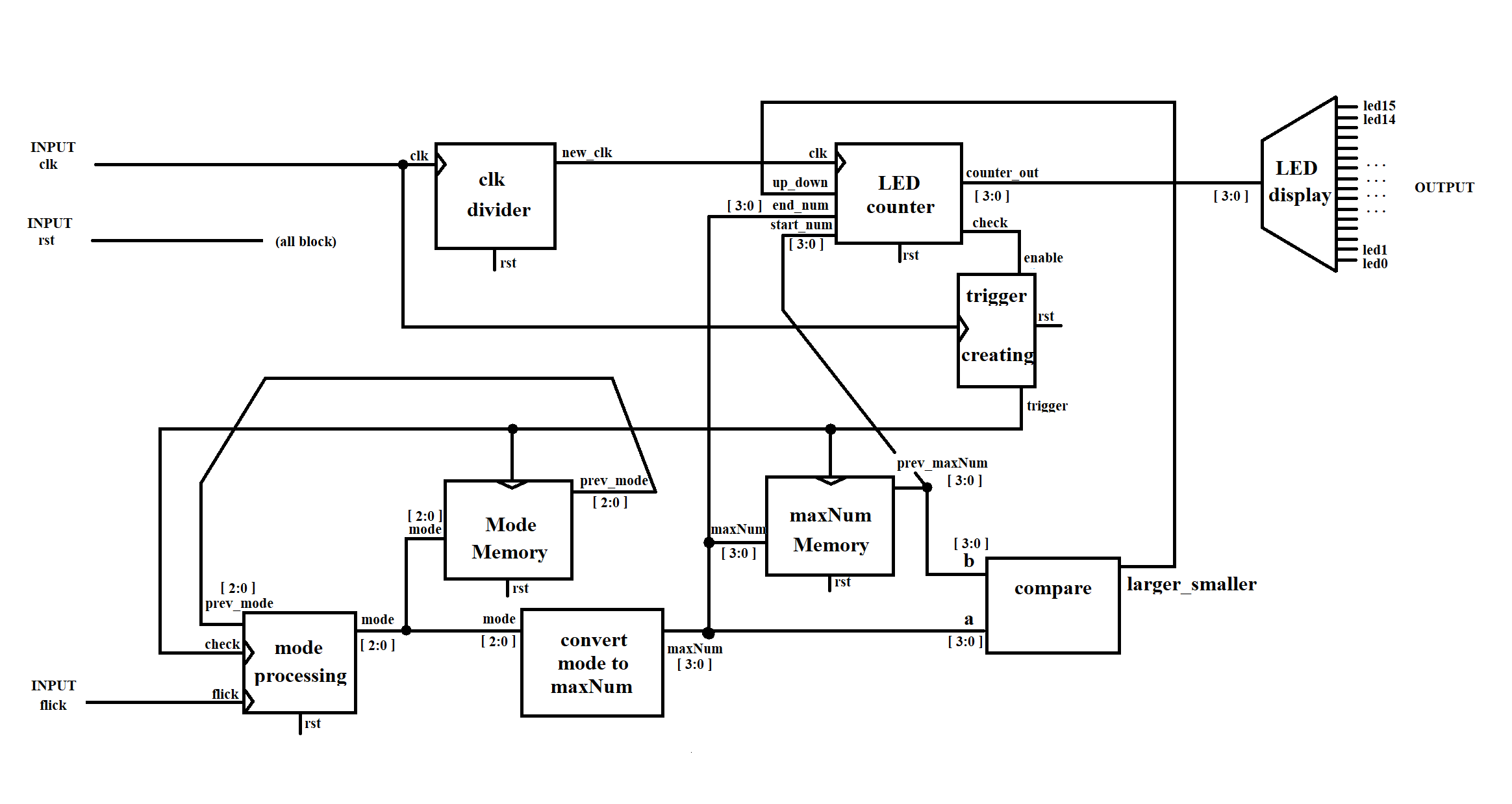


Figure 3.1: Block diagram of Bound Flasher

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | **Module** | **Signal** | **Description** | | clk\_divider | **input** clk  **input** rst  **output** new\_clk | Reduce the **clk** to **new\_clk** for LED\_counter block | | LED\_counter | **input** clk  **input** up\_down  **input** rst  **[3:0] input** start\_num  **[3:0] input** end\_num **[3:0] output** counter\_out  **output** check |  | | LED\_display | **[3:0] input** counter\_out  **[15:0] output** led\_out | Converting 4 bits number (represent for on-light LEDs) to 16 bits displayed on single LED, start from bit 0. | | trigger\_creating | **input** enable  **input** clk  **input** rst  **output** trigger | To create trigger for Mode\_memory, Mode\_memory, and Mode\_processing. | | Mode\_memory | **[2:0] input** mode  **input** trigger  **input** rst  **[2:0] output** prev\_mode | To store the previous mode value. | | maxNum\_Memory | **input** trigger  **input** rst  **[3:0] input** maxNum  **[3:0]** **output** prev\_maxNum | To store the previous maxNum value. | | mode\_processing | **[2:0] input** prev\_mode  **input** check  **input** flick  **input** rst  **[2:0]** **output** mode | If **check** pulse up, data from **flick** & **prev\_mode** will be combined to caculate **mode** (new mode). | | convert\_mode\_to\_maxNum | **[2:0] input** mode  **[3:0] output** maxNum | **maxNum** represent for number of on-light corresponding to **mode**. | | compare | **[3:0] input** a  **[3:0] input** b  **output** larger\_smaller | Compare **a** (new maxNum) to **b** (prev maxNum) to trigger count up or down mode | |

Table 3.1: Block diagram of Bound Flasher Description

## 3.2. State Machine

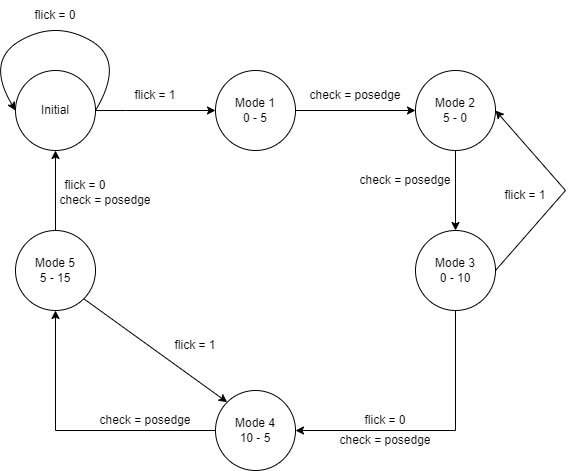


Figure 3.2: State Machine of Bound Flasher

|  |  |
| --- | --- |
| Variable Name | Description |
| flick | - Special input for controlling state transfer |
| check | - Checking condition to change mode |

Table 3.2: Variable name of State machine

|  |  |
| --- | --- |
| State name of State machine | Description |
| Initial | All lamps are OFF |
| Mode 1 | Turn ON gradually from lamp [0]to lamp [5]**.** |
| Mode 2 | Turn OFF gradually from lamp [5] **(max)** to lamp [0] **(min)**. |
| Mode 3 | Turn ON gradually from lamp [0]to lamp [10]. |
| Mode 4 | Turn OFF gradually from lamp [10] **(max)** to lamp [5] **(min)** |
| Mode 5 | Turned ON gradually from lamp [5] to lamp [15] |

Table 3.3: State name of State machine

# 4. History

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Author | Modified part | Description |
| 06/03/2022 | Lê Hoàng Minh Tú  Nguyễn Trọng Nhân | Interface | - New creation & complete interface & block diagram.  - Block diagram of Bound Flasher Description (without detailed decripstion) |
| 07/03/2022 | Lê Hoàng Minh Tú | block diagram,  block diagram description | - Add **rst** signal in block diagram  - Fill **Mode\_memory** & **maxNum\_Memory** description |
| 8/3/2022 | Nguyễn Trọng Nhân | block diagram,  block diagram description | - Fill **LED\_display** & **convert\_mode\_to\_maxNum** & **mode\_processing** |
| 8/3/2022 | Nguyễn Thanh Sang | FSM + Explaination | -Fill **Trigger\_creating**, **State name of State machine** description |
| 8/3/2022 | Hồ Hữu Trọng | FSM + Explaination | - Fill **clk\_divider, compare** description  - New creation and conplete FSM, fill FSM variable name and description |
|  |  |  |  |