|  |
| --- |
|  |
| **RTL\_EXERCISE\_1 BOUND FLASHER** |
|  |
| |  |  | | --- | --- | | Author | Tan Hai. Nguyen | | Date | 2017/09/27 | | Version | 1.0 | |
|  |
|  |

|  |
| --- |
|  |

Contents

[1. Interface 2](#_Toc494282141)

[2. Functional implementation. 3](#_Toc494282142)

[3. Internal implementation. 5](#_Toc494282143)

[3.1. Overall. 5](#_Toc494282144)

[3.2. State Machine 7](#_Toc494282145)

[4. History 10](#_Toc494282146)

# 1. Interface

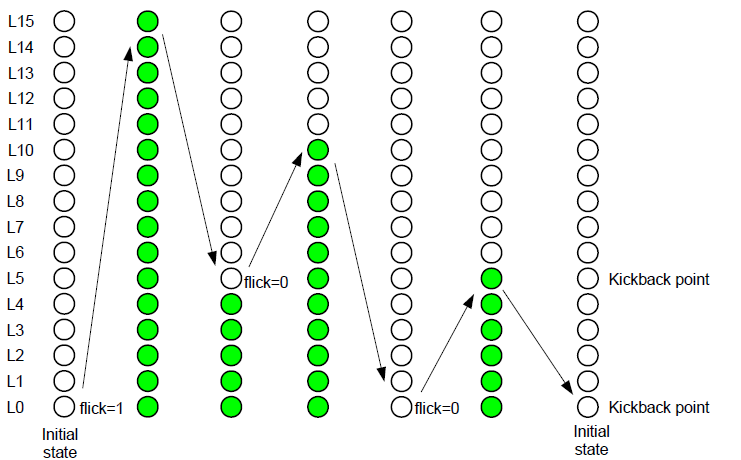
|  |
| --- |
| **bound\_flasher**  clk  16  flick  rst\_n |
| a\_lamp  Figure 1: The figure of Bound Flasher System |

|  |  |  |  |
| --- | --- | --- | --- |
| Signal | Width | In/Out | Description |
| clk | 1 | In | The function operate state’s transition at positive edge of the clock signal. |
| rst\_n | 1 | In | When rst\_n =0, system will be reset with initial state. |
| flick | 1 | In | Special input for controlling state transfer. |
| a\_lamp | 16 | Out | Output of module. |

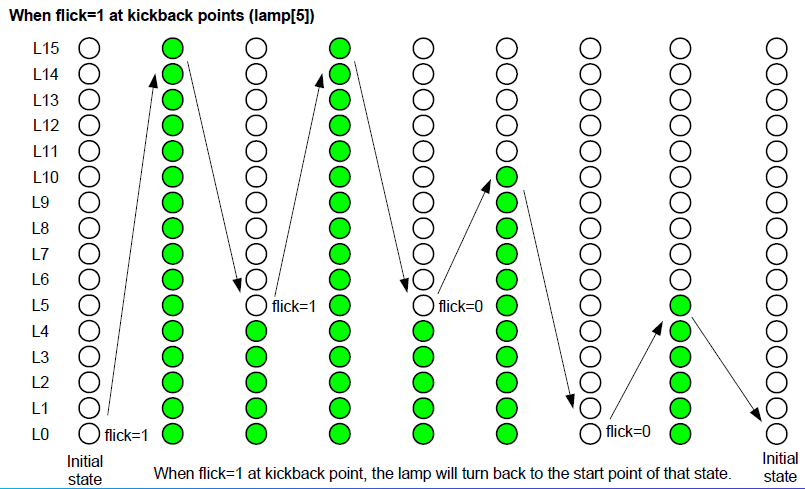
Table 1: Description of signals in Bound Flasher

# 2. Functional implementation.

* Implement a 16-bits lamps 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 lamp[0] to lamp[15].
* The lamps are turned OFF gradually from lamp[15] to lamp[5].
* The lamps are turned ON gradually from lamp[5] to lamp[10].
* The lamps are turned OFF gradually from lamp[10] to lamp[0].
* The lamps are turned ON gradually from lamp[0] to lamp[5].
* Finally, the lamps s are turned OFF gradually from lamps [5] to lamps [0], return to initial state.
* Additional condition: At each kickback point (lamp[5] and lamp[0]), if flick signal is ACTIVE, the lamps will go back and repeat that STATE. For simple, kickback point is considered only when the lamps 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.

**bound\_flasher**

next\_f\_state

state control logic

f\_state

4

clk

f\_state

4

16

flick

lamp control logic

a\_lamp

next\_lamp

a\_lamp

temp

16

16

rst\_n

Figure 3.1: Block diagram of Bound Flasher

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | Items | Description | | State control logic | This block is used to control logic between lots of states.  Ex: Begining at INTL state. According to the condition, the state will move to the other states.  +Input: flick, f\_state, temp.  +Output: next\_f\_state  This is a combinational logic block. | | f\_state | This block is the D-flipflop. When rst\_n =0 and clk signal changes to 1, f\_state is placed to next\_f\_state.  +Input: clk, rst\_n  +Output: next\_f\_state  This is a sequential logic block. | | Lamp control logic | This block is used to control logic between a lots of lamps.  Ex: Begining at INTL state. According to the condition, the state will move to the other states.  +Input: f\_state, flick, temp  +Output: next\_lamp  This is a combinational logic block. | | a\_lamp | This block is the D-flipflop. When rst\_n =0 and clk signal changes to 1, a\_lamp is placed to next\_a\_lamp.  +Input: clk, rst\_n  +Output: next\_lamp  This is a sequential logic block. | | Combinational logic | This means that no memory element is needed to realize the logic. | | Sequential logic | The output of sequential logic is not determined by the current inputs only. It may depending on what condition, state, it is in. | |

Table 3.1: Block diagram of Bound Flasher Description

## 3.2. State Machine

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| state  event | | | INTL  (1) | UP\_  HIGH  (2) | DOWN\_  TOP\_  TO\_  FIVE  (3) | UP\_  FIVE\_  TO\_  TEN  (4) | DOWN\_  TEN\_  TO\_  ZERO  (5) | UP\_  ZERO\_  TO\_  TEN  (6) | UP\_  ZERO\_  TO\_  FIVE  (7) | DOWN\_  FIVE\_  TO\_  ZERO  (8) |
| rst\_n=0 | | | a\_lamp=0  INTL | a\_lamp=0  INTL | a\_lamp=0  INTL | a\_lamp=0  INTL | a\_lamp=0  INTL | a\_lamp=0  INTL | a\_lamp=0  INTL | a\_lamp=0  INTL |
| rst\_n = 1 | flick = 0 | a\_lamp  [14] = 1 |  | {a\_lamp  [14:0],  1’b1}  (3) | {1’b0,  a\_lamp  [15:1]} |  |  |  |  |  |
| a\_lamp  [10] = 1 |  | {a\_lamp  [14:0],  1’b1} | {1’b0,a\_lamp  [15:1]} | {1’b0,  a\_lamp  [15:1]}  (5) | {1’b0,  a\_lamp  [15:1]} |  |  |  |
| a\_lamp  [5] = 1 |  | {a\_lamp  [14:0],  1’b1} | {a\_lamp  [14:0],  1’b1}  (4) | {a\_lamp  [14:0],  1’b1} | {1’b0,  a\_lamp  [15:1]} |  | {a\_lamp[14:0],  1’b1}  (8) | {1’b0,  a\_lamp  [15:1]} |
| a\_lamp  [0] = 1 |  | {a\_lamp  [14:0],  1’b1} |  |  | {a\_lamp  [14:0],  1’b1}  (7) |  | {a\_lamp  [14:0],  1’b1} | {1’b0,  a\_lamp  [15:1]}  INTL |
| Other than above | No operation | {a\_lamp  [14:0],  1’b1} | {1’b0,  a\_lamp  [15:1]} | {a\_lamp  [14:0],  1’b1} | {1’b0,  a\_lamp  [15:1]} | No operation | {a\_lamp  [14:0],  1’b1} | {1’b0,  a\_lamp  [15:1]} |
| flick = 1 | a\_lamp[14] = 1 |  | {a\_lamp  [14:0],1’b1}  (2) | {1’b0,  a\_lamp  [15: 1]} |  |  |  |  |  |
| a\_lamp[10] = 1 |  | {a\_lamp  [14:0],  1’b1} | {1’b0,  a\_lamp  [15:1]} | {1’b0,  a\_lamp  [15:1]}  (5) | {1’b0,  a\_lamp  [15:1]} | {1’b0,  a\_lamp  [15:1]}  (5) |  |  |
| a\_lamp[5] = 1 |  | {a\_lamp  [14:0],  1’b1} | {a\_lamp  [14:0],  1’b1}  (2) | {a\_lamp  [14:0],  1’b1} | {a\_lamp  [14:0],  1’b1}  (4) | {a\_lamp  [14:0],  1’b1} | {1’b0,  a\_lamp  [15:1]}  (8) | {1’b0,  a\_lamp  [15:1]} |
| a\_lamp[0] = 1 |  | {a\_lamp  [14:0],  1’b1} | {a\_lamp  [14:0],  1’b1} |  | {a\_lamp  [14:0],  1’b1}  (6) | {a\_lamp  [14:0],  1’b1} | {a\_lamp  [14:0],  1’b1} | {1’b0,  a\_lamp  [15:1]}  INTL |
| Other than above | a\_lamp= 1  (2) | {a\_lamp  [14:0],1’b1} | {1’b0,  a\_lamp  [15:1]} | {a\_lamp  [14:0],  1’b1} | {1’b0,  a\_lamp  [15:1]} | {a\_lamp  [14:0],  1’b1} | {a\_lamp  [14:0],  1’b1} | {1’b0,  a\_lamp  [15:1]} |

Figure 3.2: State Machine of Bound Flasher

: shows that the same operation in “other than above” row is applied.

|  |  |
| --- | --- |
| Variable name | Description |
| rst\_n | Reset signal |
| clk | Clock signal |
| flick | Special input for controlling state transfer |
| a\_lamp | Output |

Table 3.2: Variable name of State machine

|  |  |
| --- | --- |
| State name | Description |
| INTL | Initial state |
| UP\_HIGH | Turn on from a\_lamp[0] to a\_lamp[15] |
| DOWN\_TOP\_TO\_FIVE | Turn off from a\_lamp[15] to a\_lamp[5] |
| UP\_FIVE\_TO\_TEN | Turn on from a\_lamp[5] to a\_lamp[10] |
| DOWN\_TEN\_TO\_ZERO | Turn off from a\_lamp[10] to a\_lamp[0] |
| UP\_ZERO\_TO\_TEN | Turn on from a\_lamp[0] to a\_lamp[10] |
| UP\_ZERO\_TO\_FIVE | Turn on from a\_lamp[0] to a\_lamp[5] |
| DOWN\_FIVE\_TO\_ZERO | Turn off from a\_lamp[5] to a\_lamp[0] |

Table 3.3: State name of State machine

# 4. History

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Author | Modified part | Description |
| 2017/09/27 | Nguyen Hai. Tan | All | New creation |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |