

## RTL\_EXERCISE\_1 BOUND FLASHER

---

<b>Author</b>	Nhóm 09 – L01
<b>Date</b>	06/03/2023
<b>Version</b>	1.1

## Contents

1. Interface.....	2
2. Functional implementation.....	3
3. Internal implementation. ....	5
3.1. Overall. ....	5
3.2. State Machine .....	7
4. History	9

1. Interface

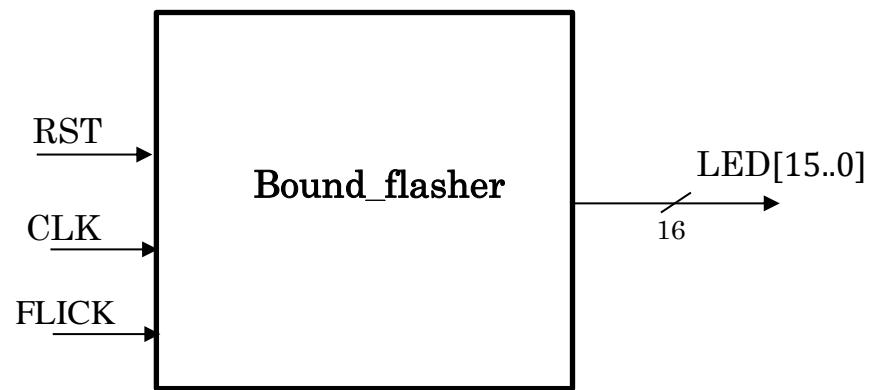


Figure 1: The figure of Bound Flasher System

Signal	Width	In/Out	Description
RST	1	In	Tín hiệu reset, tích cực mức <b>thấp</b>
CLK	1	In	Tín hiệu clock, kích cạnh dương
FLICK	1	In	Tín hiệu đặc biệt để điều khiển trạng thái
LED	16	Out	Output LED[15..0] có 16 đèn

Table 1: Description of signals in Bound Flasher

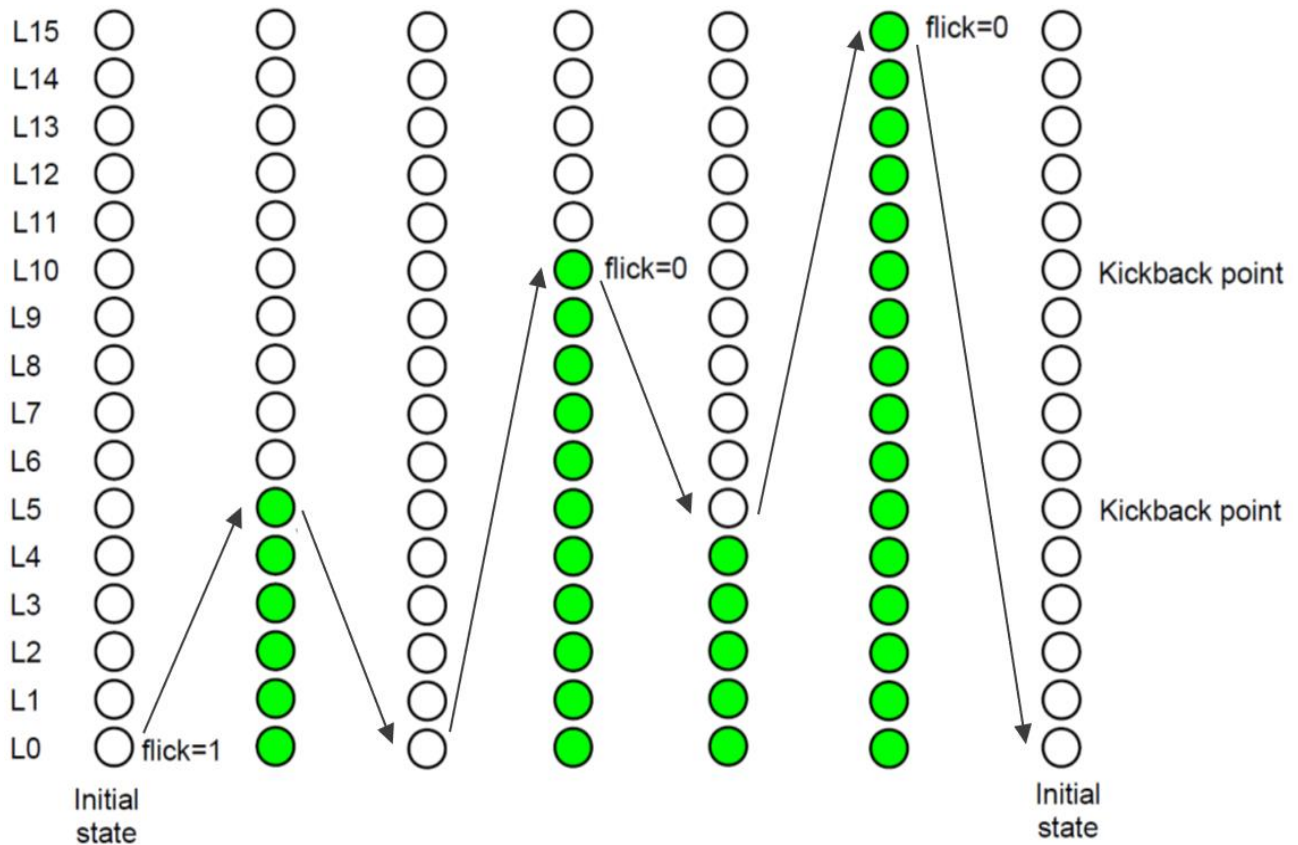
## 2. Functional implementation.

- Implement a 16-bits LEDs system
- System's Operation base on three input signal.
  - Clock
  - Reset
  - 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: An ACTIVE-LOW asynchronous input signal used to restart the system to the initial state.
  - Flick signal: A special input signal used in controlling state transitions.
  - Output – 16 bit LEDs[15..0]: A signal representing the state of 16 lamps of the bound flasher.
- At the initial state, all lamps are OFF. If flick signal is ACTIVE (set 1), 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]
  - Finally, 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.

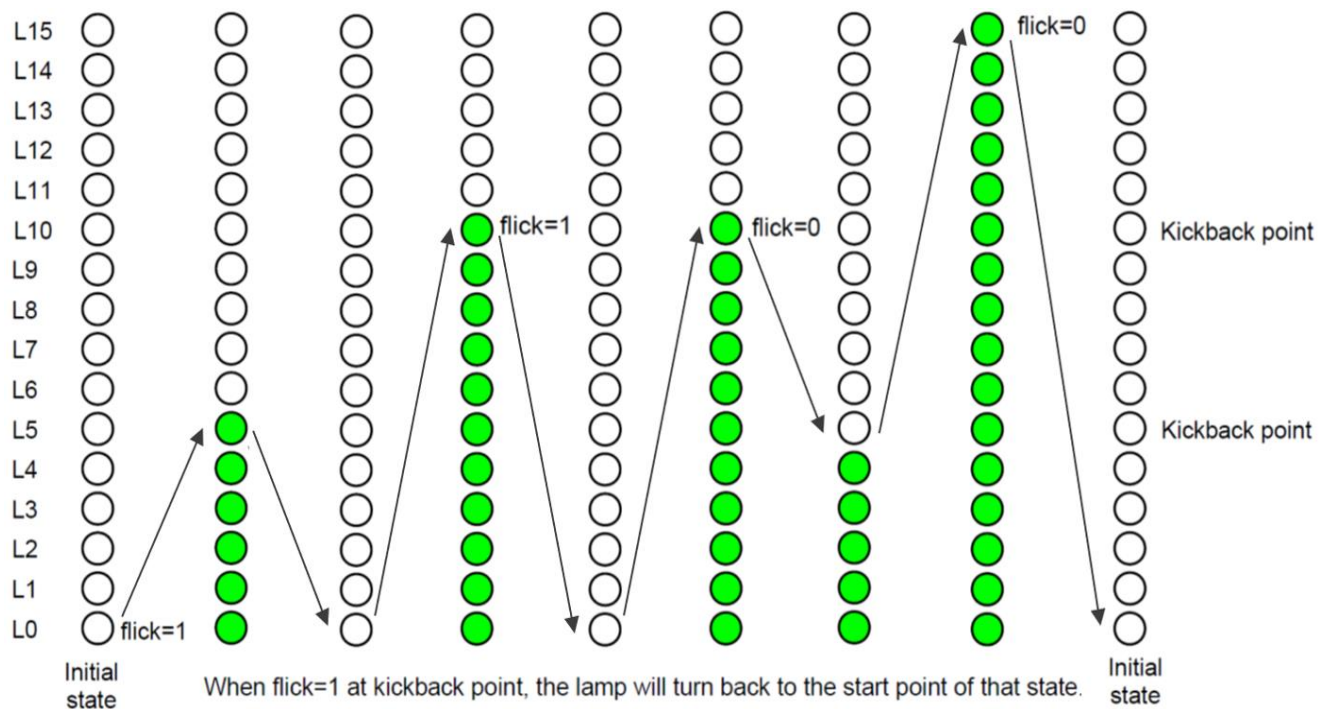
## RTL\_Exercise1 Bound Flasher

– Some insulations:

- When flick = 0 at kickback points



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



3. Internal implementation.

3.1. Overall.

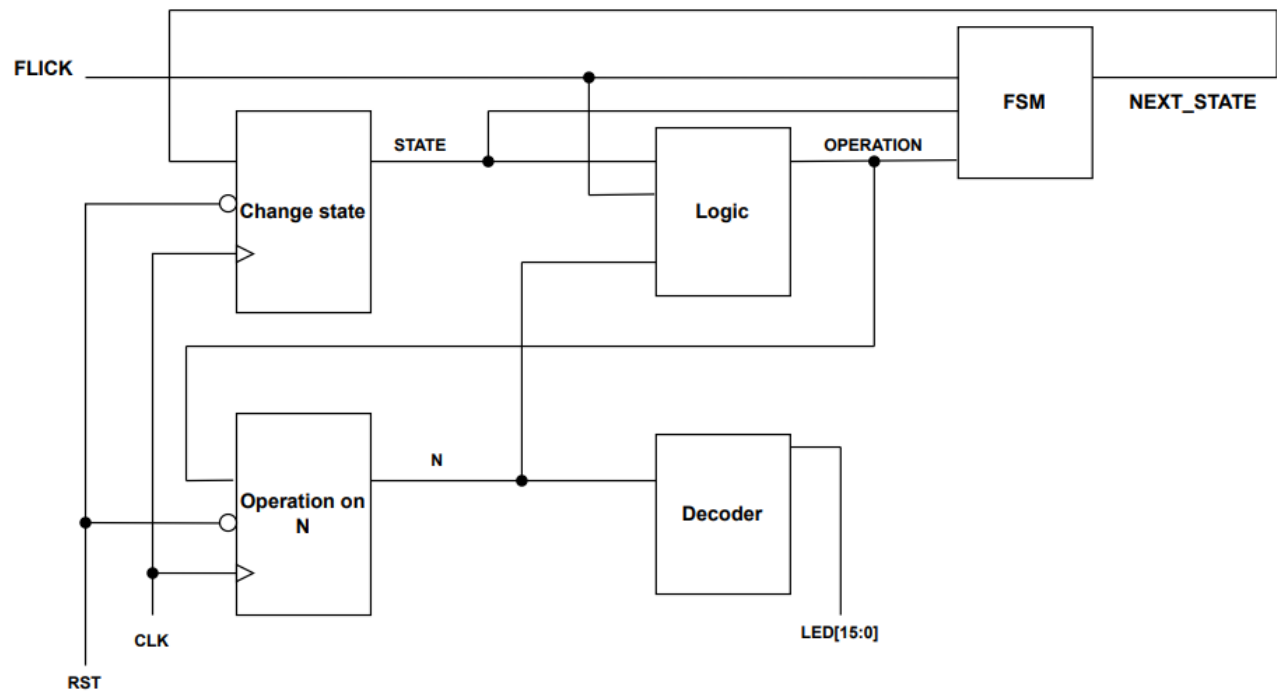


Figure 3.1: Block diagram of Bound Flasher

Name	Type	Description
RST	input	Tín hiệu reset
CLK	input	Xung clock
FLICK	input	Tín hiệu đặc biệt
LED	output	Trạng thái đèn
STATE	reg	Chứa thông tin của trạng thái hiện tại
NEXT_STATE	reg	Chứa thông tin của trạng thái tiếp theo
N	integer	Dùng để bật tắt đèn, quyết định đến trạng thái tiếp theo
OPERATION	reg	Điều khiển N trong xung nhịp tiếp theo
CHANGE STATE	DFF	Thay đổi trạng thái hiện tại đến trạng thái kế

DECODER	decoder block	Chuyển đổi tín hiệu N thành đèn LED
OPERATION on N	DFF	Xác định việc tăng, giảm tín hiệu N dựa trên đầu vào
LOGIC	control block	Xử lý các điều kiện và chuyển tín hiệu xuống khối OPERATION on N
FSM	control block	Xác định trạng thái tiếp theo dựa trên 3 trạng thái là FLICK, STATE và OPERATION

Table 3.1: Block diagram of Bound Flasher Description

## 3.2. State Machine

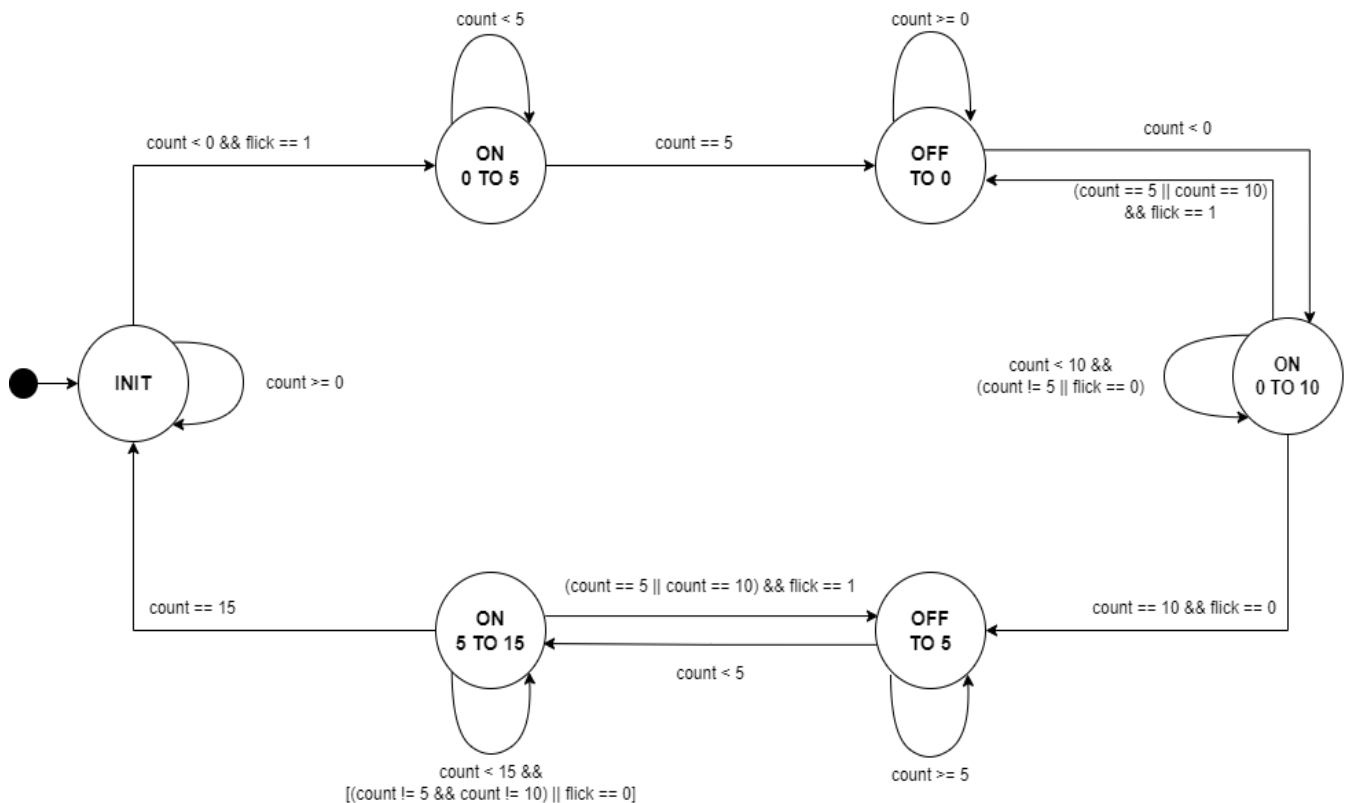


Figure 3.2: State Machine of Bound Flasher

Variable name	Description
count	Được sử dụng để nhằm theo dõi vị trí đèn LED nào được bật gần đây nhất và dùng như là điều kiện so sánh để chuyển đổi trạng thái trong FSM nếu count đang ở “kickback point” (LED[5] và LED[10]). Qua mỗi chu kỳ xung clock, biến count luôn được cộng dồn 1 đơn vị dù ở bất kỳ state nào.
LED	Dùng như một mảng chứa 16 phần tử, tương đương quản lý 16 led đơn. Mỗi phần tử có 2 giá trị là “1” và “0”, quy ước cho 2 trạng thái của 1 led đơn lần lượt là “bật” và “tắt”
flick	Dùng như là một tín hiệu đầu vào để xác định trạng thái tiếp theo, nếu trạng thái hiện tại của FSM rơi vào trong 2 trường hợp: “turn_on_0_to_10” và “turn_on_5_to_15” và biến count đang ở “kickback point”.

Table 3.2: Variable name of State machine



State name	Description
INIT	<p>Đây là trạng thái đầu tiên của FSM, dùng để làm mới toàn bộ các trạng thái của 16 led đơn về mức “0”, bất kể 16 led đơn đang ở trạng thái nào đi nữa:</p> <p>Led[count] = 0 count--</p> <p>Kiểm tra điều kiện:</p> <ul style="list-style-type: none"> <li>+ Nếu count &lt; 0 và flick == 1: nhận tín hiệu flick để bắt đầu sáng dần về Led[5] (chuyển sang state “ON 0 TO 5”).</li> <li>+ Nếu count &gt;= 0: state init đóng vai trò tắt dần dần 16 led nếu state trước đó là “ON 5 TO 15” (ở lại state INIT).</li> </ul>
ON 0 TO 5	<p>Đây là trạng thái các led đơn từ vị trí 0 tới 5 của mảng 16 led (LED[0] tới LED[5]) được bật sáng dần dần qua từng chu kỳ xung clock:</p> <p>Led[count] = 1 count++</p> <p>Kiểm tra điều kiện:</p> <ul style="list-style-type: none"> <li>+ Nếu count == 5: sáng đủ đến Led[5], chuyển sang tắt dần về Led[0] (chuyển sang state “OFF TO 0”).</li> <li>+ Nếu count &lt; 5: sáng chưa đủ đến Led[5], tiếp tục bật sáng dần (ở lại state “ON 0 TO 5”).</li> </ul>
OFF TO 0	<p>Trạng thái có chức năng thực hiện hiệu ứng tắt dần dần các đèn ở vị trí count đến 0 theo chu kỳ xung clock bằng việc thực hiện tắt đèn mỗi ở vị trí count:</p> <p>Led[count] = 0 count--</p> <p>Kiểm tra điều kiện:</p> <ul style="list-style-type: none"> <li>+ Nếu count &gt;= 0, quay lại trạng thái OFF TO 0</li> <li>+ Nếu count &lt; 0, chuyển sang trạng thái ON 0 TO 10</li> </ul>
ON 0 TO 10	<p>Trạng thái có chức năng thực hiện hiệu ứng bật đèn dần dần các đèn ở vị trí 0 đến 10 theo chu kỳ xung clock bằng việc thực hiện bật đèn ở mỗi vị trí count:</p> <p>count++ Led[count] = 1</p> <p>Kiểm tra điều kiện:</p> <ul style="list-style-type: none"> <li>+ Nếu count &lt; 10 và count không phải là Kickback point hoặc plick = 0, quay lại trạng thái ON 0 TO 10</li> <li>+ Nếu count là Kickback point và flick = 1, chuyển về lại trạng thái OFF TO 0</li> <li>+ Nếu count = 10 và flick = 0, chuyển sang trạng thái OFF TO 5</li> </ul>
OFF TO 5	<p>Điều kiện so sánh trước khi vào trạng thái này sẽ có hai trường hợp:</p> <ul style="list-style-type: none"> <li>- count == 10 &amp;&amp; flick == 0 (từ trạng thái ON_0_TO_10 chuyển sang)</li> <li>- (count == 5    count == 10) &amp;&amp; flick == 1 (từ trạng thái ON_5_TO_15 chuyển sang)</li> </ul> <p>Trạng thái này sẽ lần lượt tắt từ LED[count] cho đến đèn LED[5], tắt mỗi đèn</p>

	<p>trong một chu kì xung clock:</p> <ul style="list-style-type: none"> <li>- <math>LED[count] = 0</math>: tắt đèn thứ count</li> <li>- <math>count--</math> : giảm giá trị biến count để có thể tắt đèn tiếp theo nếu có)</li> </ul> <p>Chỉ đến khi nào <math>count &lt; 5</math> (tức là đã tắt xong đèn LED[5]) thì mới chuyển sang trạng thái tiếp theo ON_5_TO_15.</p>
ON 5 TO 15	<p>Điều kiện so sánh trước khi vào trạng thái này sẽ là <math>count &lt; 5</math>.</p> <p>Trạng thái này sẽ tiến hành bật lần lượt từ đèn LED[5] đến đèn LED[15], bật mỗi đèn trong một chu kì xung clock:</p> <ul style="list-style-type: none"> <li>- <math>count++</math> : tăng giá trị biến count</li> <li>- <math>LED[count] = 0</math>: bật đèn thứ count</li> </ul> <p>Trong quá trình bật đèn nếu ở các điểm kickpoint mà biến flick được kích giá trị bằng 1 thì ngay lập tức dừng lại và chuyển sang trạng thái OFF_TO_5.</p> <p>Cuối cùng, nếu đã hoàn thành bật đến LED[15] thì sẽ tiến hành chuyển sang trạng thái INIT để reset lại các đèn.</p>

Table 3.3: State name of State machine

#### 4. History

Date	Author	Modified part	Description
28/02/2023	All		Xác định yêu cầu, chia các phần đặc tả thiết kế
28/02/2023	All	Interface	Xác định và mô tả input & output
01/03/2023	All	Functional Implementation	Xác định các chức năng của hệ thống
05/03/2023	Bùi Thanh Duy Vũ Quang Huy	Internal Implementation	Hiện thực phần 3.1 - Overall
05/03/2023	Lê Duy Hào Đỗ Thành Minh Nguyễn Ngọc Trí	Internal Implementation	Hiện thực phần 3.2 - State Machine