Bound Flasher Circuit Validation

Created by: Group 3 - VLSI Lab CC02

Bound Flasher's order of operations

At the initial state, all lamps are OFF. If flick signal is ACTIVE (set 1), the flasher start operating:
1. The lamps are turned ON gradually from lamp[0] to lamp[5].
2. The lamps are turned OFF gradually from lamp[5] (max) to lamp[0] (min).
3. The lamps are turned ON gradually from lamp[0] to lamp[10].
4. The lamps are turned OFF gradually from lamp[10] (max) to lamp[5] (min).
5. The lamps are turned ON gradually from lamp[5] to lamp[15].
6. Finally, the lamps are turned OFF gradually from lamp[15] to lamp[0], return to initial state.

Target											
1st category		2nd category		3rd category		4th category		4th category		Confirmation Method	Status
No.		No.	(Test case name)	No.	(Test case explanation)	No.	(Test steps)	No.	(Expected outcome)		
1	**Bound Flasher circuit**	1	start_program	1	Verify that the program starts only when **flick** = 1.	1	Initially set **flick** = 0 for a set number of clock cycles then set **flick** = 1.	1	**lamp[15:0]** stays at 0 when **flick** = 0 then **lamp[0]** to **lamp[5]** are turned on gradually (**line 1**) when **flick** =	Simulation	Pass
		2	normal_flow	2	Verify that the program follows the normal order of operations when **flick** = 0 at kickback points (**lamp[5]** and **lamp[10]**).	2	Set **flick** = 0 right before the kickback point at **lamp[5]** when **lamp[0]** to **lamp[10]** are being turned on (**line 3**) then leave it as is.	2	The program follows the normal order of operations and returns to the initial state.	Simulation	Pass
		3	flick_behavior_1	3	When the program is turning **lamp[0]** to **lamp[10]** on (**line 3**) and **flick** = 1 at kickback point **lamp[5]**, the program will start turning **lamp[5]** to **lamp[0]** off instead.	3	Set **flick** = 1 right before the kickback point at **lamp[5]** when **lamp[0]** to **lamp[10]** are being turned on (**line 3**).	3	**lamp[5]** to **lamp[0]** are turned off gradually then the program will attempt the operation at **line 3** again.	Simulation	Pass
		4	flick_behavior_2	4	When the program is turning **lamp[0]** to **lamp[10]** on (**line 3**) and **flick** = 1 at kickback point **lamp[10]**, the program will start turning **lamp[10]** to **lamp[0]** off instead of **line 4**.	4	Set **flick** = 0 right before the kickback point at **lamp[5]** and **flick** = 1 right before the kickback point at **lamp[10]** when **lamp[0]** to **lamp[10]** are being turned on (**line 3**).	4	**lamp[10]** to **lamp[0]** are turned off gradually then the program will attempt the operation at **line 3** again.	Simulation	Pass
		5	flick_behavior_3	5	When the program is turning **lamp[5]** to **lamp[15]** on (**line 5**) and **flick** = 1 at kickback point **lamp[5]**, the program will turn **lamp[5]** off	5	Set **flick** = 1 right before the kickback point at **lamp[5]** when **lamp[0]** to **lamp[15]** are being turned on (**line 5**).	5	**lamp[5]** will be turned off then the program will attempt the operation at **line 5** again.	Simulation	Pass
		6	flick_behavior_4	6	When the program is turning **lamp[5]** to **lamp[15]** on (**line 5**) and **flick** = 1 at kickback point **lamp[10]**, the program will turn **lamp[10]** to **lamp[5]** off instead.	6	Set **flick** = 0 right before the kickback point at **lamp[5]** and **flick** = 1 right before the kickback point at **lamp[10]** when **lamp[0]** to **lamp[15]** are turning on (**line 5**).	6	**lamp[10]** to **lamp[5]** are turned off gradually then the program will attempt the operation at **line 5** again.	Simulation	Pass