

# Midterm

## Bicycle Light

### Specifications

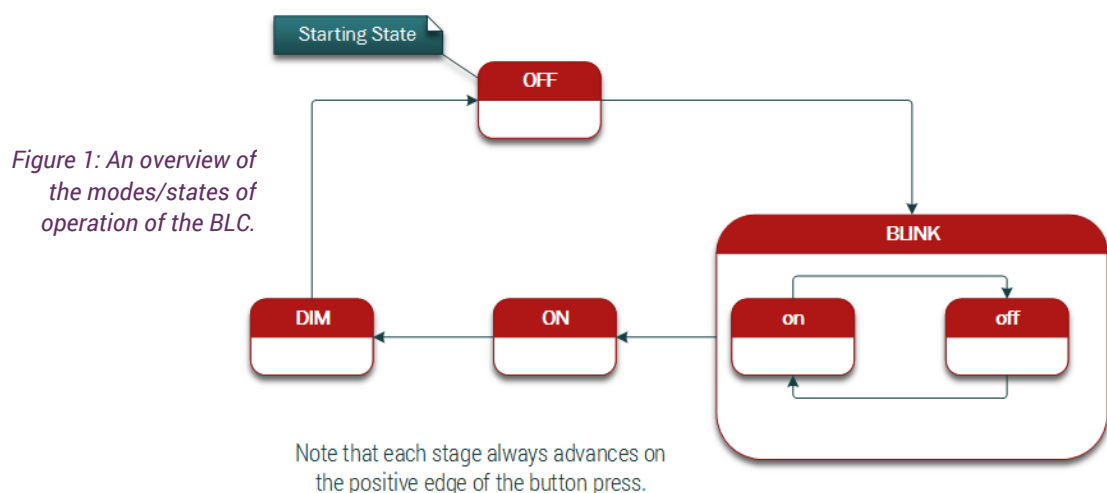
The bicycle light controller [BLC] has one output and one input.

An “LED” output leads to an LED controller which drives/powers an LED from a logical pulse-width modulated [PWM] signal. When the BLC outputs a logical high, the LED will be on at 100% of its brightness. When the BLC outputs a logical low, the LED will be off (0% brightness). The LED controller is an external component which uses the BLC “LED” output to deliver the proper voltage across and current through the LED.

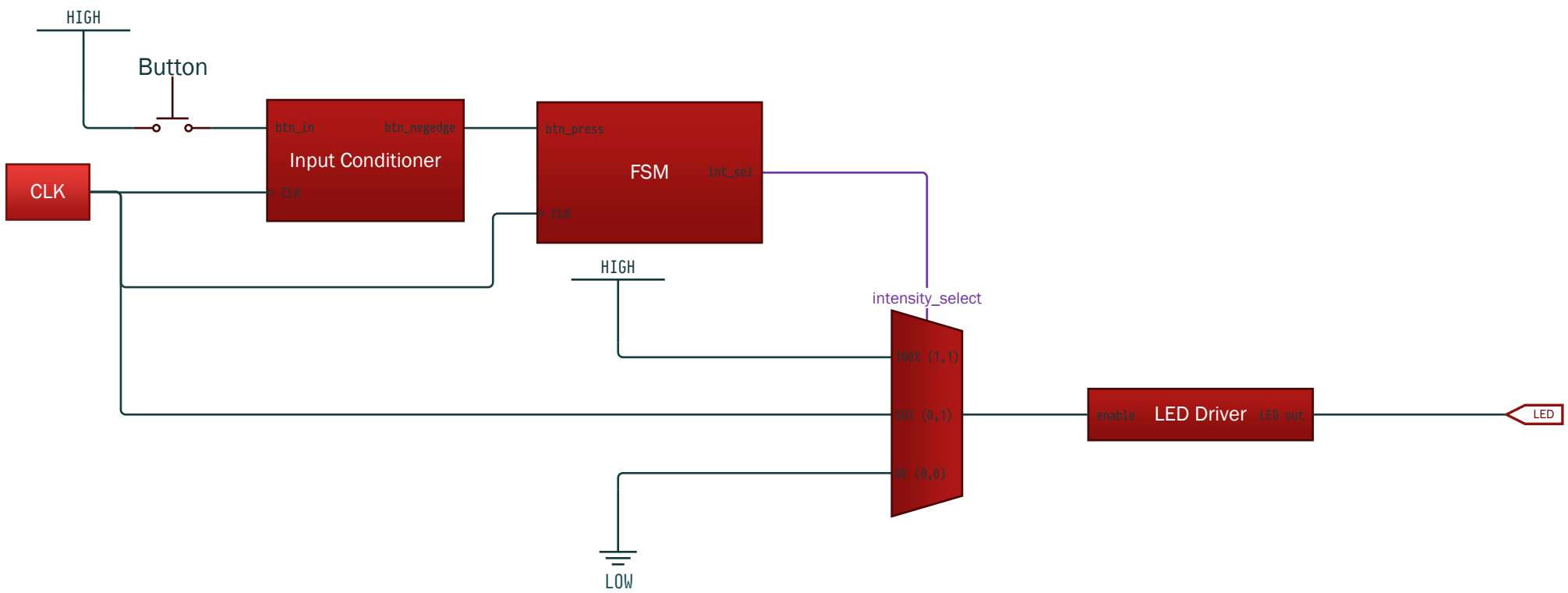
A push button provides a one-bit input. The button is spring-loaded so it is naturally open (logical 0). When the switch is depressed, the button is closed and outputs a high signal (logical 1). When the button is released, the BLC should change operational modes.

The BLC has four operational modes. The first mode is “OFF”, in which the LED is off. The second mode is “BLINK”, in which the LED alternates sequentially between being fully on and fully off at 2Hz. The “off” and “on” sub-states of this blinking mode have equivalent durations (i.e. the LED is on for a quarter of a second and then off for a quarter of a second, etc.). The third mode is “ON”, during which the LED is fully on. The fourth and final mode is “DIM”, during which the LED is lit at 50% brightness. The states proceed sequentially—when the button is released, OFF transitions to BLINK, BLINK transitions to ON, ON transitions to DIM, or DIM transitions to OFF. The states and associated system controls are managed by a finite state machine [FSM] whose general architecture is shown in Figure 1.

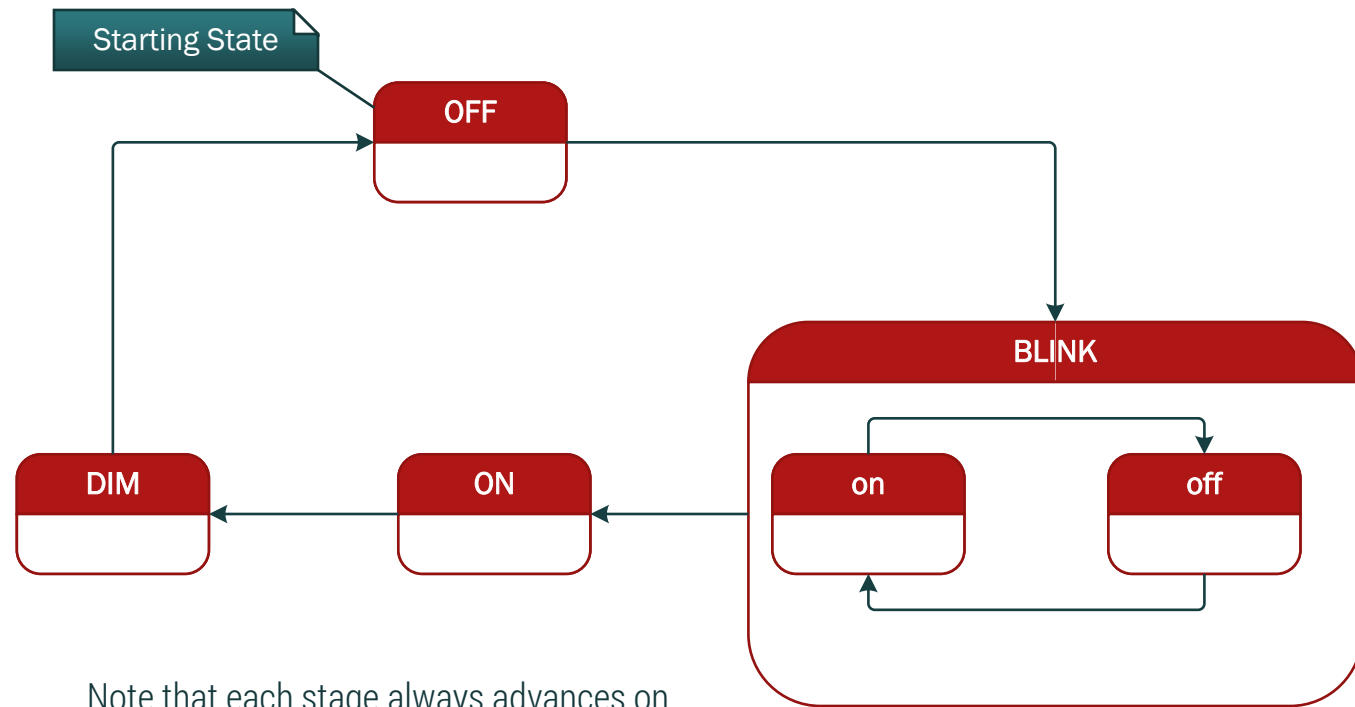
### BLC FSM Overview



TITLE				DESCRIPTION	INPUTS
Bicycle Light Controller Block Diagram					OUTPUTS
ITEM	QTY.	COST PER	TOTAL COST	From each display state, the circuit can determine the appropriate LED intensity output. The driver converts this into a LED-drivable signal and outputs it to the LED. COST: 657	LED [signal which directly drives LED.]
Input Cond	1	207	207		
FSM	1	219	219		
4d-MUX	1	18	18		
LED Driv.	1	211	211		
Button	1	0	0		
Clock	1	2	2		



# BLC FSM Overview

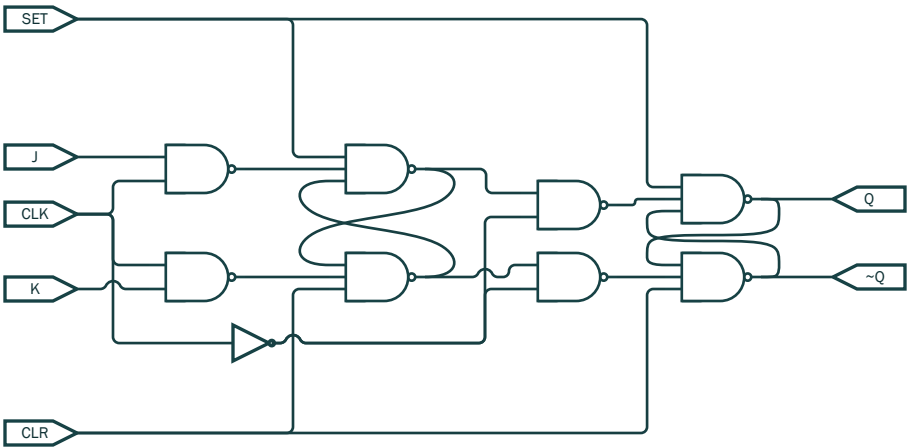


Note that each stage always advances on the positive edge of the button press.



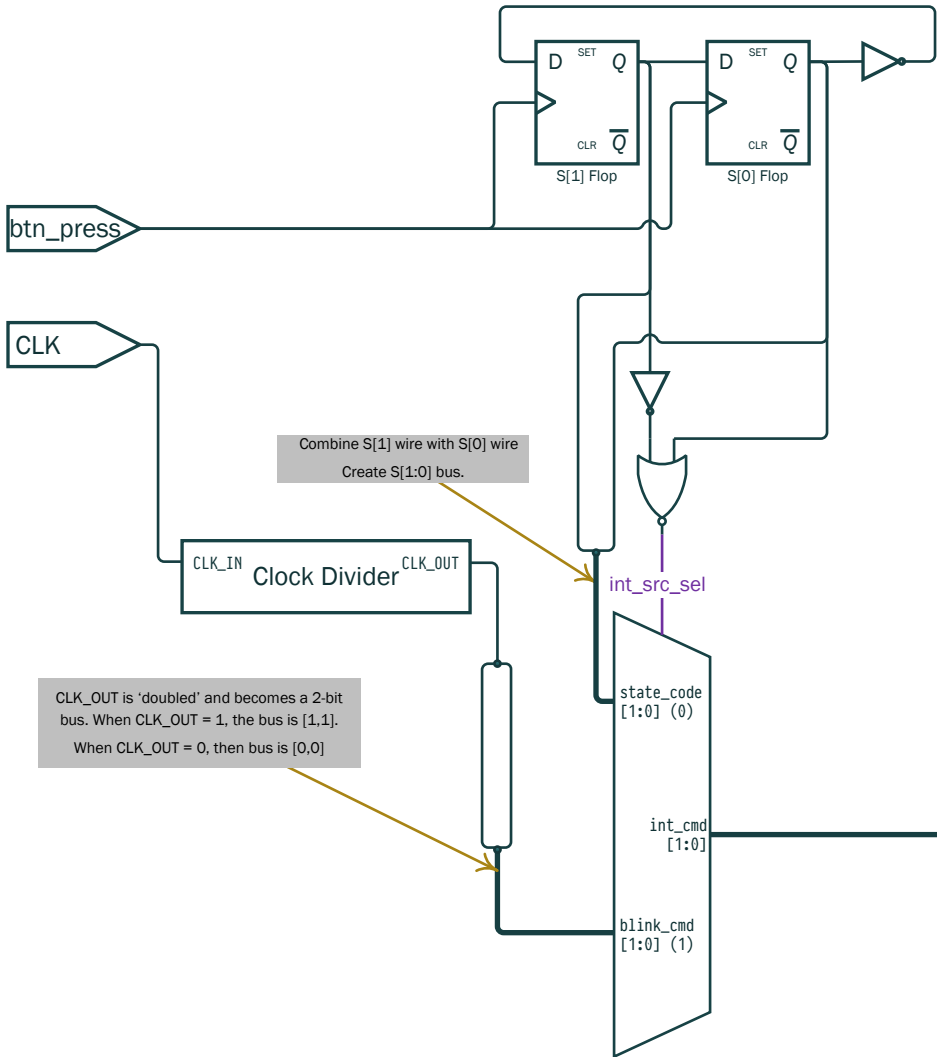
TITLE	JK Flip Flop	DESCRIPTION	INPUTS
		NAND implementation of JK flip-flop with Set and Reset [as seen at <a href="http://userpages.umbc.edu/~squire/download/jkff.gif">http://userpages.umbc.edu/~squire/download/jkff.gif</a> ] COST: 21	J, K, CLK [Clock input], SET [Async. resets Q to 1, active low], CLR [Async. resets Q to 0, active low]
			OUTPUTS
			Q [output], ~Q [inverse of Q]

ITEM	QTY.	COST PER	TOTAL COST
2-in NAND	4	2	8
3-in NAND	4	3	12
Inverter	1	1	1



TITLE				DESCRIPTION		INPUTS	
Finite State Machine Schematic				Implementation of state machine. States advance on the positive edge of btn_press. The CLK is divided to a 2 Hz. signal for BLINK state. COST: 219		CLK [input clock signal], btn_press [signal asserted at negedge of button press]	
						OUTPUTS	
						Int_sel [control signal used to select which signal is used to drive LED. Corresponds to intensity]	

ITEM	QTY.	COST PER	TOTAL COST
D Flip-Flop	2	13	26
Inverter	2	1	2
Clock Divider	1	182	182
2-in NOR	1	2	2
2d MUX	1	7	7



Control Table

S[1]	S[0]	State	LED Intensity	int_sel	
0	0	OFF	0%	0	0
1	0	BLINK	0%/100%	0/1	0/1
1	1	ON	100%	1	1
0	1	DIM	50%	0	1

Clock Divider Schematic				DESCRIPTION	INPUTS
				Divides an input clock by a factor of 2^14. For example, a 32768 Hz. clock input will be output as a 2 Hz. signal. For frequency division purposes, this asynchronous counter of D-flip-flops will work; we are not concerned with effects of a rippling clock edge across the counter. COST: 182	CLK_IN [input clock signal]
					OUTPUTS
ITEM	QTY.	COST PER	TOTAL COST		CLK_OUT [divided clock signal]
D Flip-Flop	14	13	182		

