Block Diagram

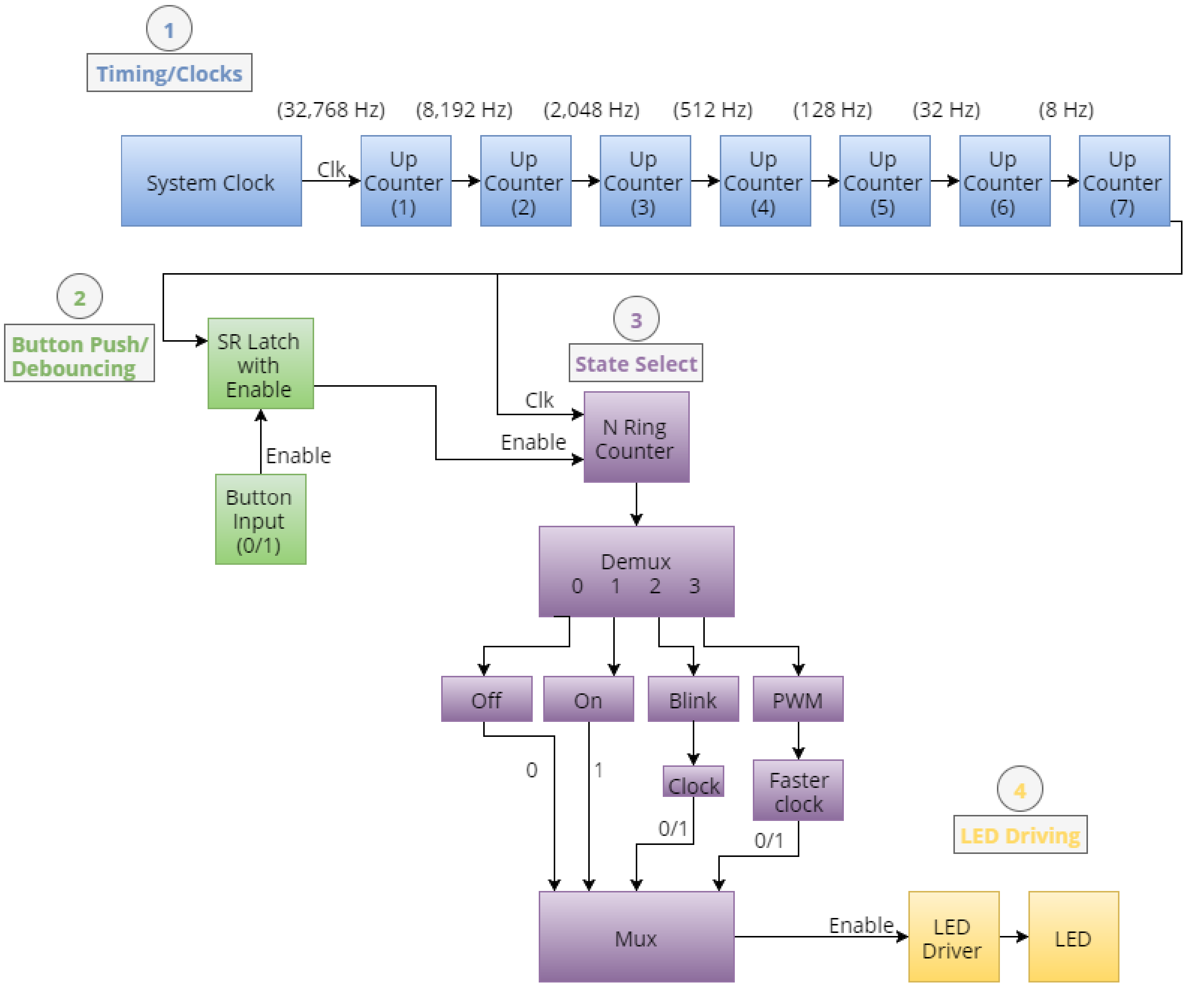
Meg McCauley

<u>High Level Walk Through</u>

Button checking occurs every clock cycle
User pushes button, debouncing occurs, system says button = on
Set state counter to state counter + 1
Set LED state to corresponding state counter
Repeat

<u>Four Important Pieces</u>

1.	Timing/Clocks	(blue)
2.	Button Push/Debouncing	(green)
3.	State Select	(purple)
4.	LED Driving	(yellow)



Explanation

1. *Timing/Clocks*

The block diagram starts with a system clock, which begins oscillating at 32,768 Hz with a square wave when is it powered. In order to get this timing on the same order of magnitude as the timing of the blinking lights, a series of seven up counters (see schematic 1) were used. Each up counter divides the frequency by 4, as can be seen on the block diagram. After all seven have been taken into account, the frequency of the main clock is 8 Hz.

2. *Button Push/Debouncing*

Next, the button input is taken from the user and debounced using a set/reset latch with an enable (aka gated SR latch, see schematic 2).

3. *State Select*

After the button state has changed to high, the N stage ring counter is activated and begins keeping track of the state. Because each stage is attached to the output of the previous stage, a ripple effect occurs, causing each stage to add one as it is triggered. The N stage ring counter is a one-hot counter, which is perfect for keeping track of states. Since we have only 4 states, a counter where N = 4 is used.

4. *LED Driving*

The output from the N stage ring counter is the state number. This output is fed into the input of a decoder (aka 'demux,' see schematic 3), which is able to change its output behavior based on the inputted state. If the state is zero, this indicates that the LED should be off, and the demux sends a 0 to the LED driver. If the state is one, this indicates that the LED should be on, and the demux sends a 1 to the LED driver. If the state is 2, this indicates that the LED should be blinking, so the demux uses a clock to switch the input from 0 to 1 though oscillations. If the state is 3, this indicates that the LED should be at a mid-brightness level. To do this, pulse width modulation (aka PWM) is used. With PWM, a certain duty cycle is used (indicated by the "faster clock" on the block diagram) which flashes the light on and off at a certain speed. This tricks the human eye into seeing a dimmer light because the speed is faster than the eye can process. Instead of seeing the light on and off, the human eye averages the values and sees a lower brightness level. The 1s and 0s that are passed from the demux are then passed into a multiplexor (aka ‘mux,’ see schematic 4) which turns the multiple values into a single output. This output is then fed into the LED driver, which in turn controls the LED turning on and off.

Area Cost Estimation

Element	**Cost (in area)**
System Clock	2
Up Counter (x7)	58 x 7 = 406
Gated Set/Reset Latch	10
N Stage Ring Counter (where N = 4)	(21 x 4) – 1 = 83
1 to 4 Decoder (aka ‘1:4 Demux’)	18
4 to 1 Multiplexer (aka ‘4:1 Mux’)	23
LED Driver	211
	Total: 753

External Resources Used

1. Up Counters: http://www.allaboutcircuits.com/textbook/digital/chpt-11/synchronous-counters/
2. Gated Set/Reset Latches: http://www.allaboutcircuits.com/textbook/digital/chpt-10/the-gated-s-r-latch/
3. Decoders and Multiplexers: http://www.edwardbosworth.com/CPSC2105/Lectures/Slides_05/Chapter_03/DecodersAndMux.htm
4. Input Conditioning: https://www.acromag.com/page/signal-conditioning
5. Debouncing: http://www.ganssle.com/debouncing-pt2.htm

Schematic 1: Up Counter

Specification

An up counter is a synchronous counter that counts up in binary. When the **clk** is on the positive edge, the first flip-flop will toggle no matter what. The second flip-flop toggles only if the output of the first was high. The third will only toggle if both the first and the second output high. The fourth will only toggle if the outputs of all three previous flip-flops are high.

Inputs

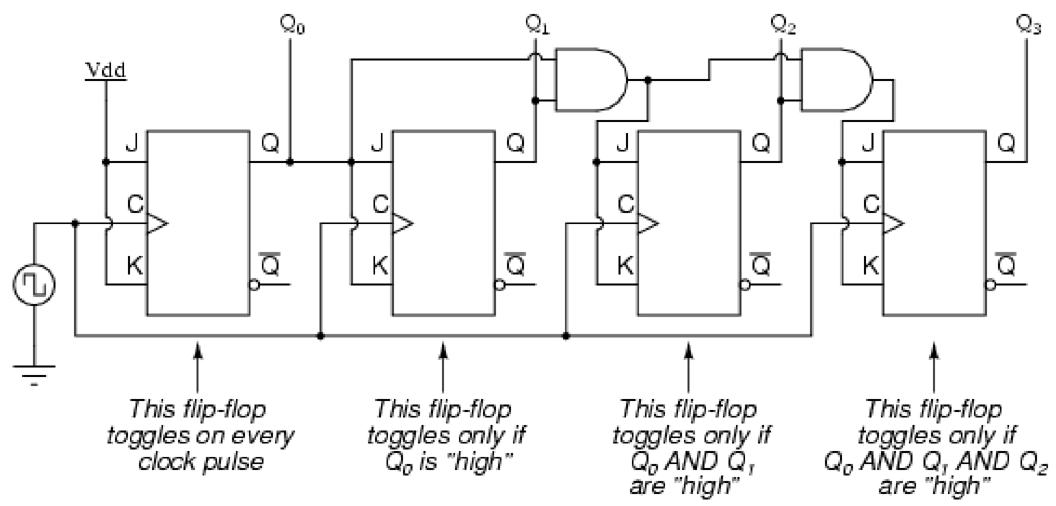
clk is the clock input, **Vdd** is the power input

Outputs

Q_0, Q_1, Q_2, Q_3 are the outputs of each flip flop respectively
(Each flip flop also has a ~Q output, but these are not used)

Schematic (from resource 1)

A four-bit synchronous "up" counter



Q_0
Q_1
Q_2
Q_3
Vdd
J
Q
C
K
$\overline{Q}$
This flip-flop toggles on every clock pulse
This flip-flop toggles only if Q_0 is "high"
This flip-flop toggles only if Q_0 AND Q_1 are "high"
This flip-flop toggles only if Q_0 AND Q_1 AND Q_2 are "high"

Cost

Subcomponent	**Cost per**	**# Used**	**Total**
JK Flip Flop	13	4	52
AND gate	3	2	6
			Total: 58

Schematic 2: Gated Set/Reset Latch

Specification

The gated SR latch behaves as a normal SR latch when the enable is on. When the enable is off, the SR latch holds its previous state.

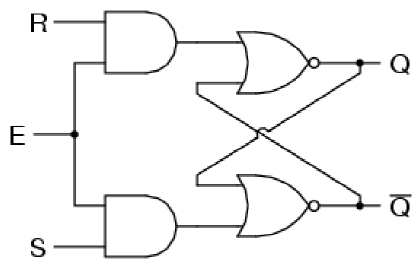
Inputs

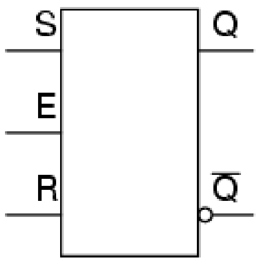
S is the first input, **R** is the second input, **E** is the enable input

Outputs

Q is the positive output, **~Q** is the negative output

Schematic (both from resource 2)





Cost

Subcomponent	Cost per	# Used	Total
AND gate	3	2	6
NOR gate	2	2	4
			Total: 10

Schematic 3: 1 to 4 Decoder (aka '1:4 Demux')

Specification

A one to four demux takes in a single value and maps it to four values based on the value of another input (or two).

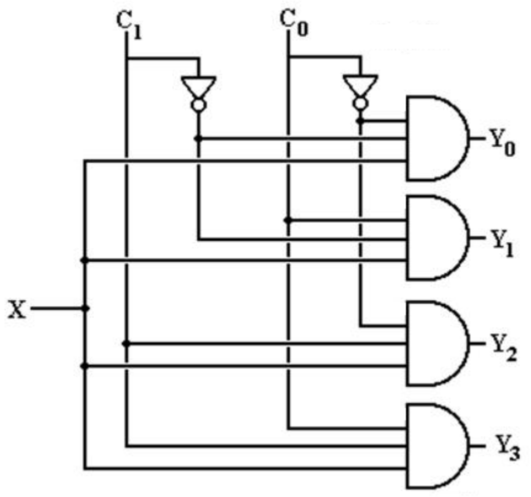
Inputs

C_1, C_0 are the other inputs that direct the output, **X** is the single input that will map to the many output values

Outputs

Y_0, Y_1, Y_2, Y_3 are the four values that the demux outputs

Schematic (from resource 3, edited to remove the enable signal)



Cost

Subcomponent	Cost per	# Used	Total
NOT gate	1	2	2
3 input AND gate	4	4	16
			Total: 18

Schematic 4: 4 to 1 Multiplexer (aka '4:1 Mux)

Specification

A four to one mux takes in four values and outputs a single value based on the value of another input (or two).

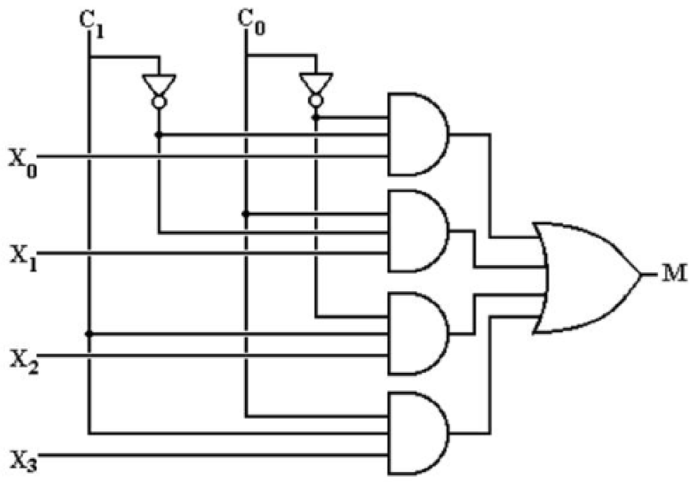
Inputs

C_1, C_0 are the other inputs that direct the output, **X_0, X_1, X_2, X_3** are the four values that the mux chooses between

Outputs

M is the final and only output

Schematic (from resource 3)



Cost

Subcomponent	**Cost per**	**# Used**	**Total**
NOT gate	1	2	2
3 input AND gate	4	4	16
4 input OR gate	5	1	5
			Total: 23