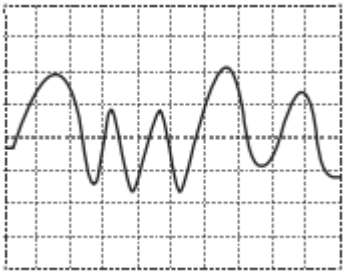
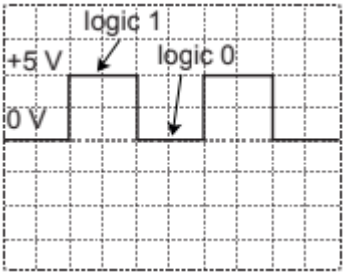
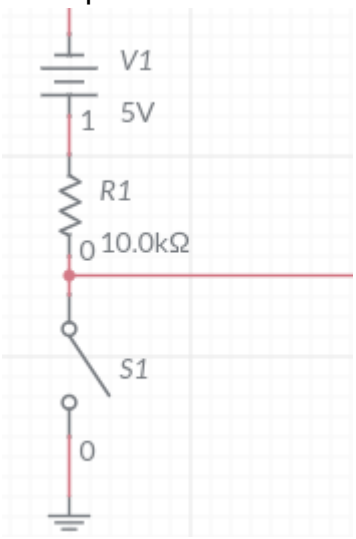
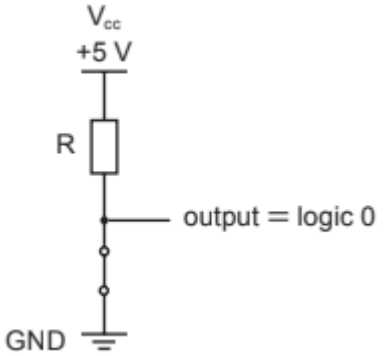
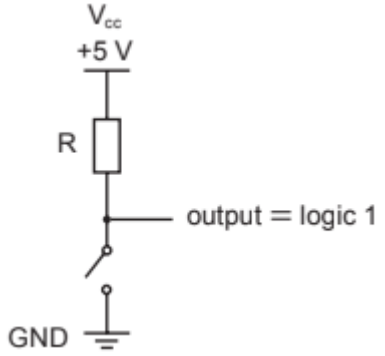
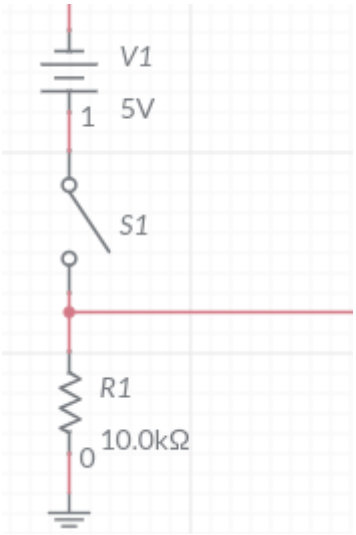
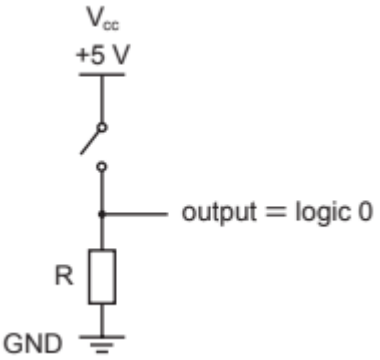
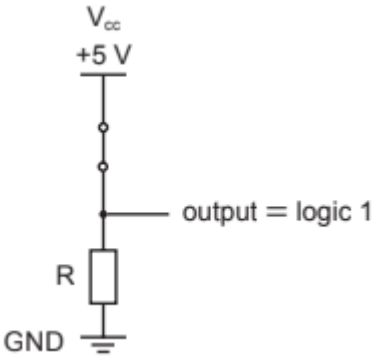
Electronics notes (Digital) (pm something3009 on discord if there are any errors)

Chapter	Content	Examples
Chapter 10: Introduction to Digital Electronics	Analogue quantities vary continuously while Digital quantities vary in steps How to differentiate between analogue and digital signals: Analogue signals: Has continuously varying Voltage levels Digital signals: Only has 2 distinct voltage levels (HIGH or LOW only) (HIGH refers to Logic 1 state (+5V), LOW refers to Logic 0 state (0V))	Analogue signals  Digital signals 
	Logic Switches Pull-Up switch:  (Pull-Up resistor with a switch below) When switch is closed, output is Logic 0 (0V) When switch is open, output is Logic 1 (+5V)	Pull-Up switch when closed:  Pull-Up switch when opened: 

	Pull-Down switch:  (Pull-Down resistor with a switch above) When switch is closed, output is Logic 0 (0V) When switch is open, output is Logic 1 (+5V) * Resistor in Pull-Up/Pull-Down switch are usually 10kΩ Resistor is needed if not the logic switch will produce an 'floating' output when open (Bad as it can pick up unwanted electrical signals and take on Logic 0 or Logic 1 in a unpredictable way)	Pull-Down switch when closed:  Pull-Down switch when opened: 
	Advantages/Disadvantages of using Digital systems over Analogue systems Advantages: - Less affected by electrical interference (due to fairly large difference between voltage levels), can tell Logic 1 apart from Logic 0 as long as it is not too badly distorted - Signals can be restored to original condition via repeaters (can travel over long distances and is more reliable) - Easier to design due to only having 2 voltage levels - Easier to store information (as it is stored as a series of 0s and 1s)	

<table>
<tr>
<td></td>
<td>
- Large number of digital components can be squeezed into small area of semiconductor material (save costs)

Disadvantages:
- Additional steps are needed to convert between Analogue and Digital signals (many quantities in our daily lives such as temperature and light intensity are analogue quantities)
</td>
<td></td>
</tr>
<tr>
<td></td>
<td>
How to represent and display values in Digital Form

Decimal System:
Made up of 10 digits (0-9)
Position of a digit determines its place value

Binary System:
Made up of 2 digits (bits) (0 and 1)
Position of a bit determines its place value
Highest place value - Most Significant Bit (MSB)
Lowest place value - Least Significant Bit (LSB)

* you can press calculator to skip the calculation parts in converting decimal to binary and Vice Versa

Binary-coded decimal (BCD):
Uses groups of 4-bit binary code to represent the digits of a decimal number (Coding system)
<table>
<tr><th>Decimal number</th><th>BCD equivalent</th></tr>
<tr><td>4</td><td>0100</td></tr>
<tr><td>98</td><td>1001 1000</td></tr>
<tr><td>163</td><td>0001 0110 0011</td></tr>
</table>
</td>
<td>
Converting between Binary, Decimal systems and BCD

Binary to Decimal:
1100_2
= $(1x2^3)+(1x2^2)+(0x2^1)+(0x2^3)$
= 8 + 4 + 0 + 0
= 12_{10}

Decimal to Binary:
12_{10}
$12 \div 2 = 6$R0 (LSB)
$6 \div 2 = 3$R0
$3 \div 2 = 1$R1
$12 \div 2 = 6$R1 (MSB)
$12_{10} = 1100_2$

Decimal to BCD:
387_{10} = 0011 1000 0111 $_{BCD}$
BCD to Decimal:
0101 0110 1001 $_{BCD}$ = 569_{10}
</td>
</tr>
</table>

<table>
<tr>
<td></td>
<td>Advantages/Disadvantages of converting between Decimal and BCD
Advantages:
- More straightforward to convert larger decimal number to BCD than to Binary
- Useful for displaying decimal numbers

Disadvantages:
- More bits are needed to store the same number in Binary

BCD to 7-Segment Display:
To display binary numbers in decimal form
Insert BCD input into 7-segment decoder to convert BCD to a 7-segment code</td>
<td>BCD to 7-Segment code Truth Table
<table><tr><th rowspan="2">BCD</th><th colspan="7">7-bit binary code</th><th rowspan="2">7-segment display (decimal digit)</th></tr><tr><th>g</th><th>f</th><th>e</th><th>d</th><th>c</th><th>b</th><th>a</th></tr><tr><td>0000</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td></tr><tr><td>0001</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td></tr><tr><td>0010</td><td>1</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td></tr><tr><td>0011</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td></tr><tr><td>0100</td><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td></tr><tr><td>0101</td><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td></tr><tr><td>0110</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td></tr><tr><td>0111</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td></tr><tr><td>1000</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr><tr><td>1001</td><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td></tr></table></td>
</tr>
<tr>
<td rowspan="2">Chapter 11: Basic Logic Gates</td>
<td>Logic Gates
Components with 1 or more inputs and 1 output of either Logic 1 or Logic 0 depending on the Logic gate

Truth Table
A way to show the outputs of a Logic gate for all possible input combinations
(Input combinations are arranged in binary from smallest to biggest)</td>
<td>Truth table of a NOT Gate
<table><tr><th>A</th><th>X</th></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table></td>
</tr>
<tr>
<td>Common Logic Gates

NOT Gate

Boolean Expression: $X = \overline{A}$
(Inverts the Input)
Truth Table
<table><tr><th>A</th><th>X</th></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table></td>
<td></td>
</tr>
</table>

AND Gate

Boolean Expression: $X = A \cdot B$
(Produces a Logic 1 output only if both inputs are Logic 1)
Truth Table

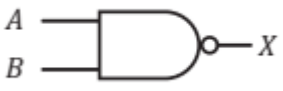
A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

OR Gate

Boolean Expression: $X = A + B$
(Produces a Logic 1 output when either or both inputs are Logic 1)
Truth Table

A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

NAND Gate (Universal Gate)



(AND + NOT gate)
Boolean Expression: $X = \overline{A \cdot B}$
(Opposite of AND gate; Produces Logic 1 if only one input is Logic 1 or both inputs are Logic 0)

Truth Table

A	B	X
0	0	1
0	1	1
1	0	1
1	1	0

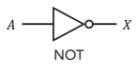
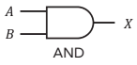
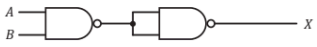
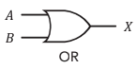
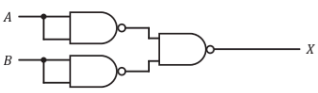
NOR Gate (Universal Gate)

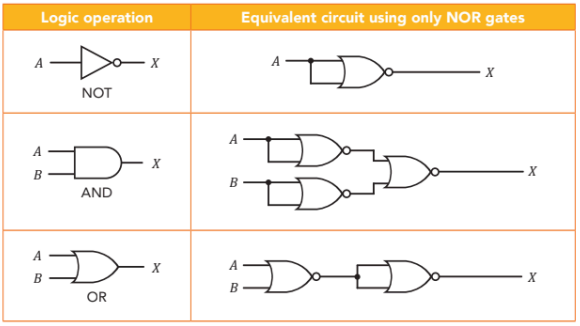
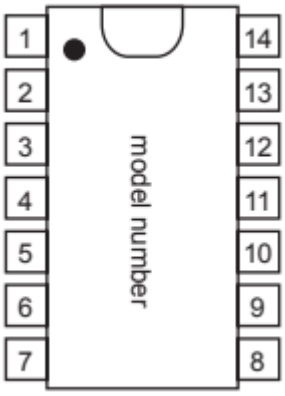
(OR + NOT gate)
Boolean Expression: $X = \overline{A + B}$
(Opposite of OR gate; Produces Logic 1 only if both inputs are Logic 0)
Truth Table:

A	B	X
0	0	1
0	1	0
1	0	0
1	1	0

Universal Gates: Gates that can be used to make any other type of gates (NAND, NOR)

NOT, AND, OR using NAND gates:

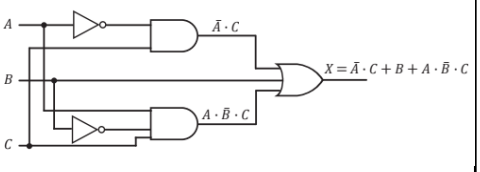
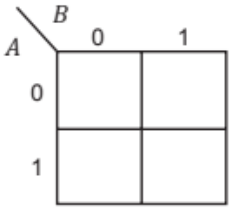
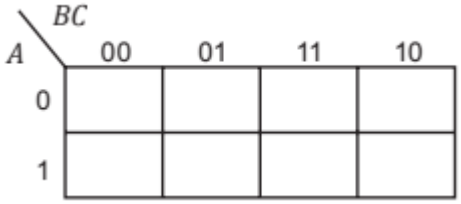
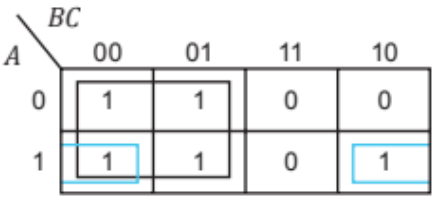
Logic operation	Equivalent circuit using only NAND gates
	
	
	

<table>
<tr><td></td><td>NOT, AND. OR using NOR gates:
 
</td><td></td></tr>
<tr><td></td><td>Dual-In-Line (DIL) ICs
Identifying pins:
 

Pin 1 - Beside Dot and Notch
* Refer to Datasheet for DIL IC connections, ratings and characteristics</td><td>Key DIL ICs:
74LS00 - 2 input NAND gate
74LS02 - 2 input NOR gate
74LS04 - NOT gate
74LS08 - 2 input AND gate
74LS11 - 3 input AND gate
74LS32 - 2 input OR gate
74LS47 - BCD to 7-Segment decoder
74LS390 - Decade Counter
NE555 - 555 Timer
LM311 - Voltage Comparator</td></tr>
<tr><td>Chapter 12: Combinational Logic Circuits</td><td>Describing a Logic Circuit

To create Truth Table:
1. Determine number of Rows (Number of possible input combinations) and Columns (Number of inputs, intermediate signals and outputs)
2. Draw and fill up the inputs of the Truth Table
3. Work out the intermediate signals and outputs

Boolean Expression
To express the outputs of an Truth Table (sum-of-product expression) (SOP)
$\cdot$ - Product (multiply)
$+$ - Sum (add up)
$\overline{X}$ - Inverse</td><td>When A = 1. B = 1, X = 1
When A = 0, C = 1, X = 1
Truth Table:
<table><tr><th>A</th><th>B</th><th>C</th><th>X</th></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>0</td><td>1</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>0</td><td>0</td></tr><tr><td>1</td><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td><td>1</td></tr></table></td></tr>
</table>

		SOP expression: $X = \overline{A} \cdot \overline{B} \cdot C + \overline{A} \cdot B \cdot C + A \cdot B \cdot \overline{C} + A \cdot B \cdot C$
	Converting Boolean expression into Logic Circuit Step 1: Draw a Logic circuit for each AND term Step 2: Connect the outputs to the inputs of an OR Gate	$X = \overline{A} \cdot C + B + A \cdot \overline{B} \cdot C$ 
	Karnaugh Map (K-map) To simplify SOP expressions to reduce number of gates required, saving costs and reducing errors Step 1. Prepare K-Map 2-Input:  3-Input:  (Note that the numbers for BC are arranged as 00 01 11 10) Step 2: Input information into K-Map Step 3: Loop the 1s in the K-Map - Loop in groups of 2 or 4 or individually - Only loop 1s that are adjacent to each other - First and Last column are considered adjacent to each other Step 4: Obtain the simplified Boolean expression by identifying the common inputs	$X = \overline{A} \cdot B \cdot C + A \cdot \overline{B} \cdot C + A \cdot B \cdot C$ A 3-input K-Map will be used  Simplified SOP expression: $X = \overline{B} + A \cdot \overline{C}$ (p.s. K-Map is easier in my opinion)

Boolean Algebra
To simplify SOP expressions to reduce number of gates required, saving costs and reducing errors

Boolean Laws
* Laws given in Datasheet

One Variable:

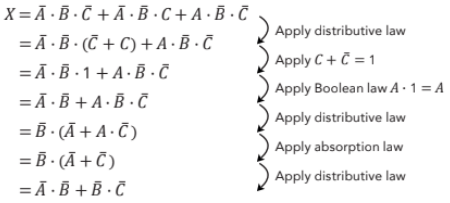
Law	Explanation
$\overline{\overline{A}} = A$	If $A = 0$, $\overline{\overline{0}} = \overline{1} = 0$ and vice versa
$A + 0 = A$	Adding a 0
$A + 1 = 1$	Answer will always become 1
$A + A = A$	If A = 1, result is 1 If A = 0, result is 0
$A + \overline{A} = 1$	Either A or $\overline{A}$ = 1
$A \cdot 0 = 0$	Multiply by 0
$A \cdot 1 = A$	Multiply by 1
$A \cdot A = A$	If A = 1, result is 1 If A = 0, result is 0
$A \cdot \overline{A} = 0$	Either A or $\overline{A}$ = 0, hence multiply by 0

Two or more variables:

Law	Type
$A + B = B + A$	OR commutative law
$A \cdot B = B \cdot A$	AND commutative law
$A + (B + C) = (A + B) + C$	OR associative law
$A \cdot (B \cdot C) = (A \cdot B) \cdot C$	AND associative law

$X = \overline{A} \cdot B \cdot C + A \cdot \overline{B} \cdot C + A \cdot B \cdot C$

Using Boolean Algebra,



(Same answer as K-Map)

<table>
<tr>
<td></td>
<td>
<table>
<tr><td>$A \cdot (B + C) = A \cdot B + A \cdot C$

$(A + B) \cdot (C + D) = A \cdot C + A \cdot D + B \cdot C + B \cdot D$</td><td>Distributive Laws</td></tr>
<tr><td>$A + A \cdot B = A$
$A \cdot (A + B) = A$

$A + \overline{A} \cdot B = A + B$

$A \cdot (\overline{A} + B) = A \cdot B$</td><td>Absorption Laws</td></tr>
<tr><td>$\overline{A \cdot B} = \overline{A} + \overline{B}$
$\overline{A + B} = \overline{A} \cdot \overline{B}$</td><td>De Morgan’s Theorem</td></tr>
</table>
just do K-Map its easier trust me
</td>
</tr>
<tr>
<td></td>
<td>Using Combinational Logic to solve real-life problems

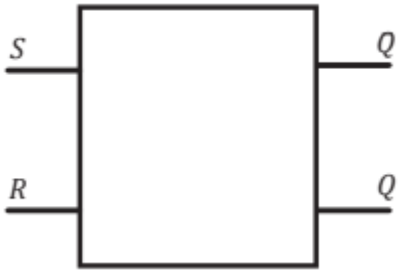
Step 1: Create Truth Table and create an unsimplified boolean expression

Step 2: Use K-Map OR Boolean Algebra to get a simplified boolean expression

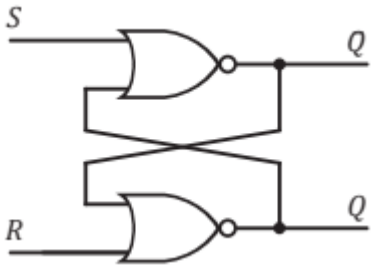
Step 3: Create the circuit using the simplified boolean expression</td>
<td>Repeat the above examples using the context given within the question

* ONLY PICK ONE METHOD WHEN SIMPLIFYING SOP</td>
</tr>
<tr>
<td>Chapter 13: Set-Reset Latches</td>
<td>An S-R Latch is a device that can store a Logic state (1 or 0)

S-R Latch symbol

 
</td>
<td></td>
</tr>
</table>

S-R Latch using NOR gates

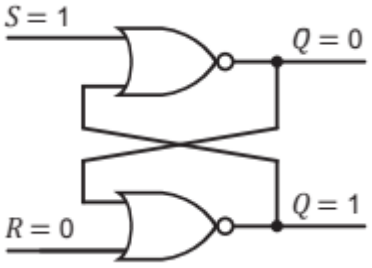


* $\overline{Q}$ is above, Q is below (Printing errors made it difficult to identify)

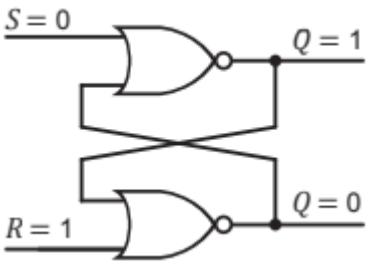
Truth Table

S	**R**	**Q**	$\overline{Q}$
0	0	No change	
0	1	0	1
1	0	1	0
1	1	Invalid	

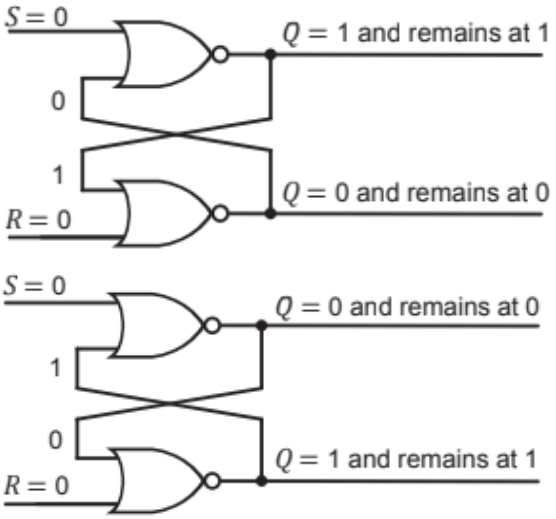
Condition 1:
When S = 1 and R = 0, the latch is set
Outputs Q = 1, $\overline{Q}$ = 0
(When S = 1, Q = 0, and when Q and R = 0, $\overline{Q}$ = 0)



Condition 2:
When S = 0 and R = 1, the latch is reset
Outputs Q = 0, $\overline{Q}$ = 1
(When R = 1, $\overline{Q}$ = 0, and when $\overline{Q}$ and S = 0, Q = 1)

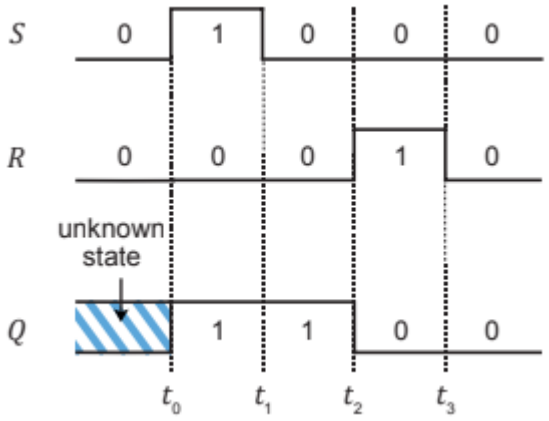


Condition 3
When S and R = 0, the latch remains at the same logic state it was before it entered this condition
(When Q = 1 and $\overline{Q}$ = 0, or Q = 0 and $\overline{Q}$ = 0, since S and R = 0, they remain the same)

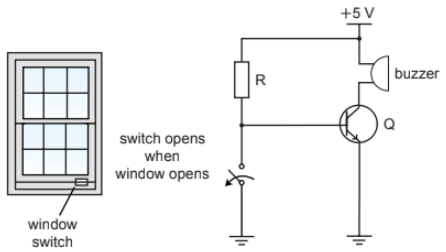


Condition 4
When S and R = 1, Q and $\overline{Q}$ = 0
(As an input of 1 into a NOR gate will result in 0, both NOR gates produce an output of 0 since S and R = 1, hence this state is invalid)

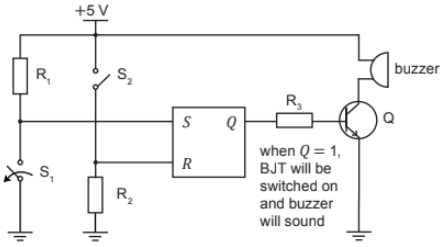
Timing Diagram
Used to show how the output of a digital system changes with time

<table>
<tr>
<td></td>
<td>


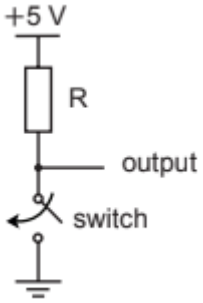
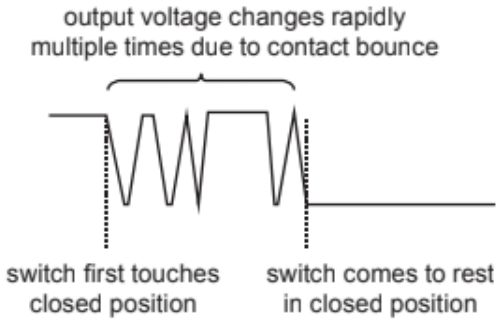
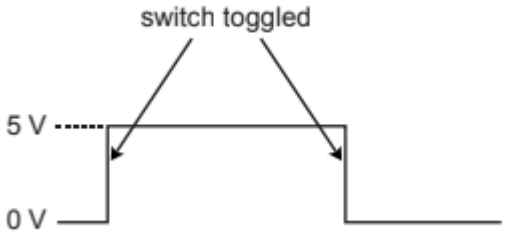
At t_0, S = 1 and R = 0, Q = 1
(Condition 1)

At t_1, S = 0 and R = 0, Q remains at 1
(Condition 3)
At t_2, S = 0 and R = 1, Q = 0
(Condition 2)
At t_3, S = 0 and R = 0, Q remains at 0
(Condition 3)
</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Applications for S-R Latch
- Systems that require an momentary occurrence to be converted into an constant output (eg Intruder Alarm System, Traffic Light, ect)</td>
<td>Intruder Alarm System without S-R Latch
 

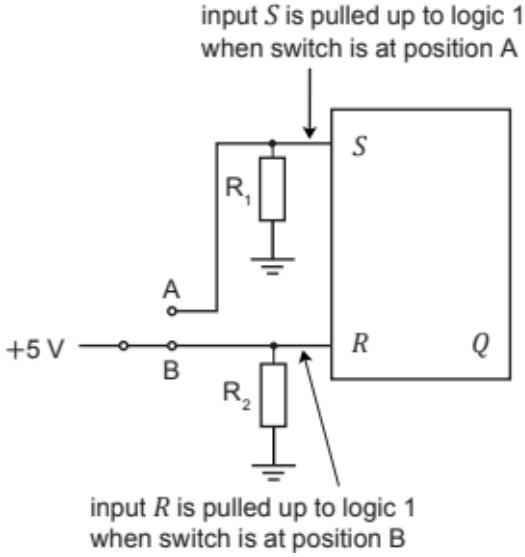
(The intruder can close the window to disable the alarm system)

Intruder Alarm System with S-R Latch
 

(The alarm system can only be reset via another switch, preferably hidden of course)</td>
</tr>
</table>

	Debounced switch using S-R Latch Contact Bounce When a Mechanical switch (SPST, SPDT, ect) is opened or closed, the contacts will connect and disconnect multiple times before having a firm connection (multiple input signals are sent, which affects the accuracy of the system)   A debounced switch can produce a single rising or falling edge signal, which is more suitable  	For rising/falling edge input digital systems, if a normal switch is used, the count value may increase by multiple times instead of once Applications: 555 Timer (Astable) input, 74LS390 Decade Counter (1CKA/2CKA) input, ect

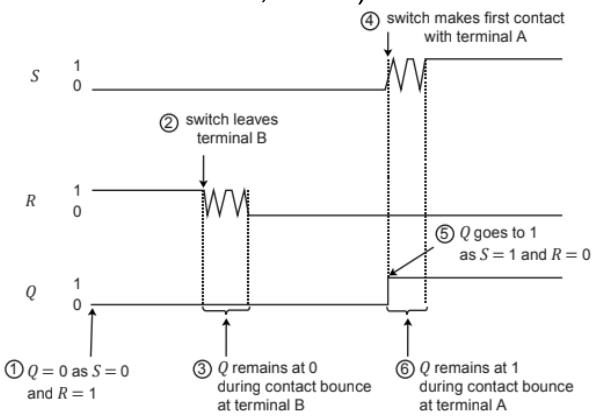
How to build a Debounced switch:



(1 Pull-Down resistor at each input S and R connected to a SPDT switch)

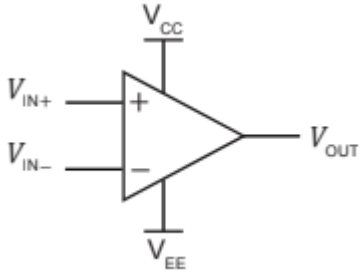
How it works:

- When the switch is at B, S is pulled down to Logic 0 while R is pulled down to Logic 1. Q = 0
- When the switch is moving from B to A, R alternates between Logic 1 and Logic 0. Q remains at 0 (As R = 0 and S = 0, there is no change in Q)
- When the switch reaches A, S alternates between Logic 1 and Logic 0. Q becomes 1 (When S = 1 and R = 0, Q = 1)



Chapter 14: Voltage Comparators, Timers and Counters

Voltage Comparator

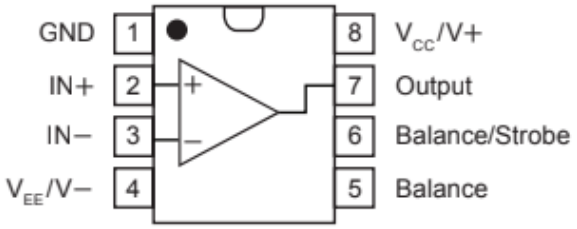


It compares between 2 analogue inputs (V_{in+} and V_{in-}) to produce a digital output (V_{out})

If $V_{in+} > V_{in-}$, V_{out} will be HIGH

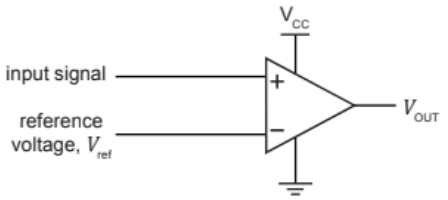
If $V_{in+} < V_{in-}$, V_{out} will be HIGH

LM311 Voltage Comparator IC

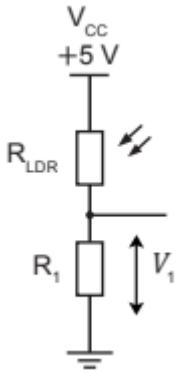


Pin	Name	How to connect
1	GND	Connect to ground (0V)
2	IN+	Connect to voltages to be compared
3	IN-	
4	V_{EE}	Connect to ground (0V)
5	Balance	Connect together (Pins not used)
6	Balance/ Strobe	
7	Output	Connect to V_{cc} using Pull-Up resistor (10kΩ)
8	V_{cc}	Connect to positive voltage

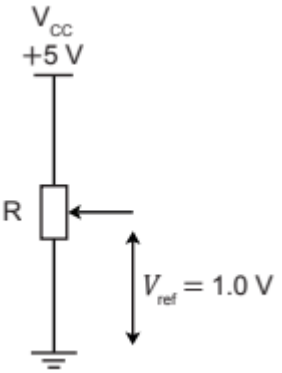
Application in a automatic home lighting system



Input Signal - LDR in Voltage Divider configuration



Reference Voltage - Voltage Divider OR Potentiometer

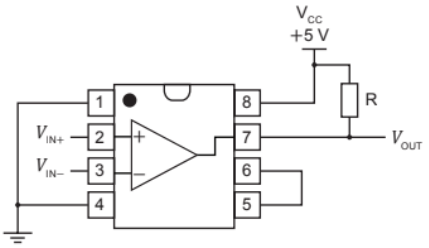


(in this example a potentiometer is used producing a reference voltage of 1V)

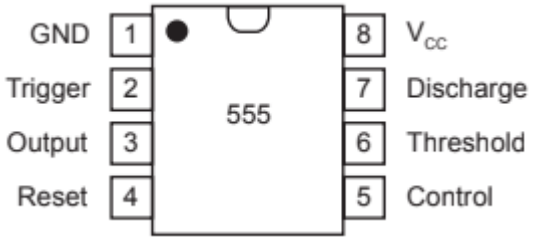
When it is dark and V_1 is less than 1V, $V_{in+} > V_{in-}$ and V_{out} will be HIGH

When it is bright and V_1 is more than 1V, $V_{in+} < V_{in-}$ and V_{out} will be LOW

How to connect LM311 Voltage Comparator IC



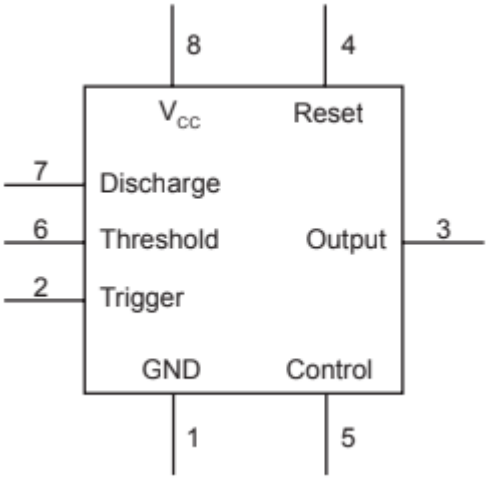
555 Timer IC



Pin	Name	Function
1	GND	Connect to ground (0V)
2	Trigger	Causes Pin 3 to go HIGH and start the timing cycle when it reaches $\frac{1}{3}V_{CC}$
3	Output	Produces a digital output
4	Reset	Resets timing interval when connected to GND
5	Control	Controls Trigger and Threshold level (Not used; connect with 10nF capacitor to GND
6	Threshold	Monitors external voltage across external capacitor When voltage reaches $\frac{2}{3}V_{CC}$,

		timing cycle ends and Pin 3 goes LOW
7	Discharge	Connects to GND and discharges external capacitor when output is LOW, acts as an open circuit when output is HIGH
8	V_{cc}	Connect to positive voltage (between 5V to 15V)

Pin layout diagram



The 555 Timer IC can function as a Monostable or an Astable Multivibrator

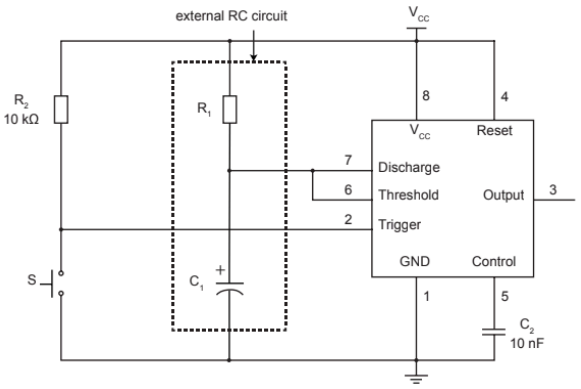
Difference between both modes

Monostable	**Astable**
When triggered, create a single rectangular pulse of predetermined width	Creates a rectangular waveform without the need of a trigger
Pulse width is determined by external capacitor and resistor	Period is determined by external capacitor and 2 resistors

* Both Monostable and Astable formulas will be given in Datasheet

Monostable Multivibrator
Timing device that produces a singular pulse of specific width (in terms of time)
The signal used to activate the multivibrator is known as a Trigger

Configuration



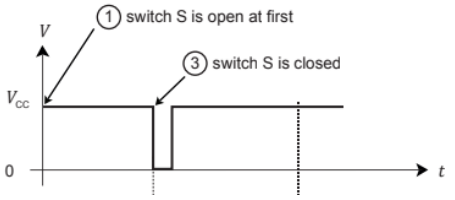
- Pin 2 is connected to a Pull-Up resistor with pushbutton switch
- Pins 6 and 7 is connected to an external RC circuit

Function

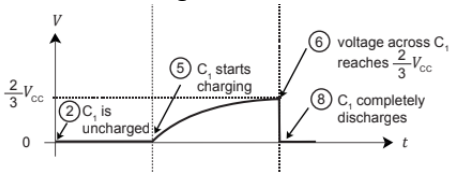
1. At first, switch S is open which causes Pin 2 to be pulled up to 5V
2. As Pin 2 is more than $\frac{1}{3}V_{CC}$, Pin 3 (Output) is LOW. Pin 7 (Discharge) is connected to GND which completely discharges C_1
3. When S is closed, Pin 2 is pulled down to 0V
4. Since Pin 2 is less than $\frac{1}{3}V_{CC}$, Pin 3 goes HIGH and the timing cycle starts. Pin 7 acts as an open circuit
5. C_1 charges towards V_{CC}
6. When C_1 reaches $\frac{2}{3}V_{CC}$, it is detected by Pin 6 (Threshold)
7. Pin 3 goes LOW and timing cycle stops

Applications: Timers, Debounced switch, ect

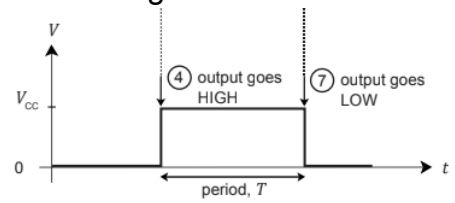
Pin 2 Voltage

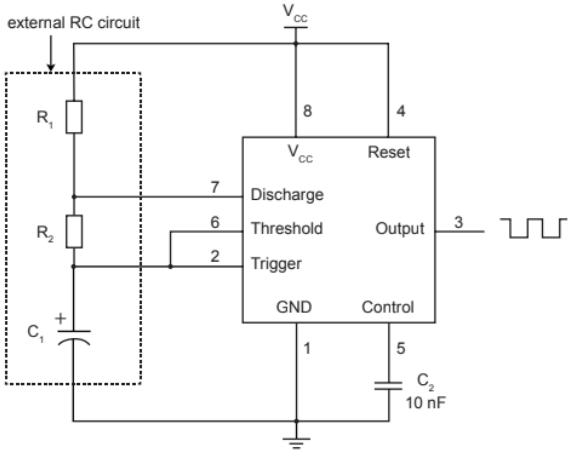
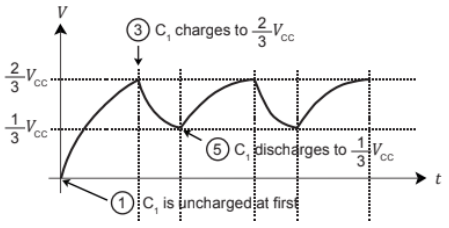
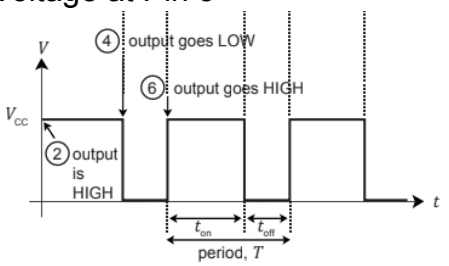


Pin 6/7 Voltage



Pin 3 Voltage



	Pulse width of output (Time Constant) $T = 1.1RC$ T - Time constant in seconds (s) R - Resistance (Ω) C - Capacitance (F)	
	Astable Multivibrator Timing device that produces a continuous stream of pulses (rectangular waveforms) Configuration  - Pin 7 connected between R_1 and R_2 , Pin 2 and 6 connected between R_2 and C_1 - R_1 , R_2 and C_1 forms an external RC circuit Function 1. Assume C_1 is not charged and Pin 2 is 0V 2. As Pin 2 is less than $\frac{1}{3}V_{CC}$, Pin 3 (Output) is HIGH and causes Pin 7 to act as an open circuit. C_1 charges to V_{CC} 3. C_1 charges till Pin 6 (Threshold) reaches $\frac{2}{3}V_{CC}$ 4. Pin 3 goes LOW and Pin 7 is connected to GND, causing C_1 to discharge 5. C_1 discharges to $\frac{1}{3}V_{CC}$	Applications: Flashing LED lights, Logic clocks, ect Voltage at Pin 2 and 6 (Assuming C_1 is uncharged)  Voltage at Pin 3 

6. Pin 3 goes HIGH and the process repeats itself (Repeat 1 to 5)

Duration of signal at HIGH state is known as t_{on}, while the duration of signal at LOW state is known as t_{off}.

Period of a complete cycle $T = t_{on} + t_{off}$

$$T = \frac{(R_1 + 2R_2)C_1}{1.44}$$

$$t_{on} = 0.7(R_1 + R_2)C_1$$

$$t_{off} = 0.7R_2C_1$$

T - Period in seconds (s)
R - Resistance (Ω)
C - Capacitance (F)
t_{on} - Duration when output is HIGH
t_{off} - Duration when output is LOW

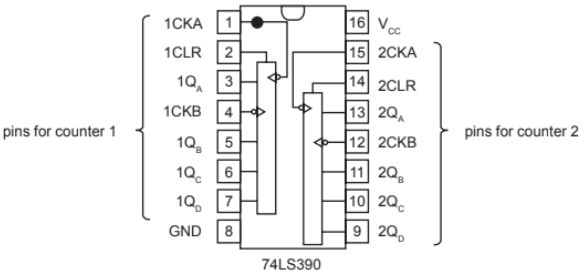
* Duty cycle and frequency of pulses:

$$\text{Duty cycle} = \frac{t_{on}}{T} \times 100\%$$

$$f = \frac{1}{T}$$

f - Frequency (Hz)
T - Period (s)
t_{on} - Duration when output is HIGH

74LS390 Decade Counter IC



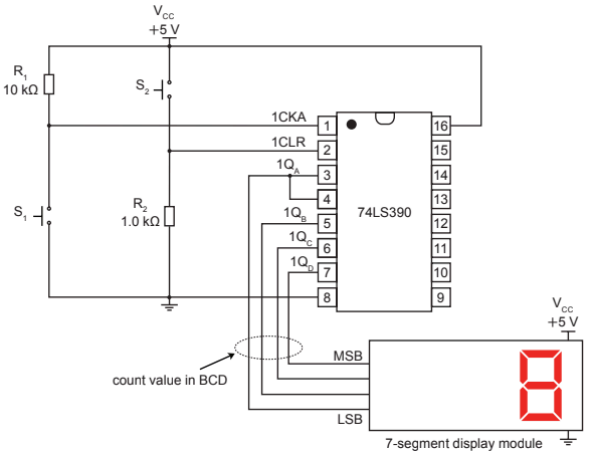
Pin	Name	Function
16	V_{cc}	Connect to 5V power source
8	GND	Connect to ground (0V)

1, 15	1CKA, 2CKA	Clock A input pins for counter 1 and 2 respectively. Increases count value by one for every falling edge
2, 14	1CLR, 2CLR	Clear input pins for counter 1 and 2 respectively. Resets the count value to zero when it is HIGH.
4, 12	1CKB, 2CKB	Clock B input pins for counter 1 and 2 respectively.
7, 6, 5, 3	$1Q_D$, $1Q_C$, $1Q_B$, $1Q_A$	Output pins for counter 1, to represent count value as BCD code where $1Q_D$ is the most significant bit (MSB) while $1Q_A$ is the least significant bit (LSB)
9, 10, 11, 13	$2Q_D$, $2Q_C$, $2Q_B$, $2Q_A$	Output pins for counter 2, to represent count value as BCD code where $2Q_D$ is the most significant bit (MSB) while $2Q_A$ is the least significant bit (LSB)

Count sequence

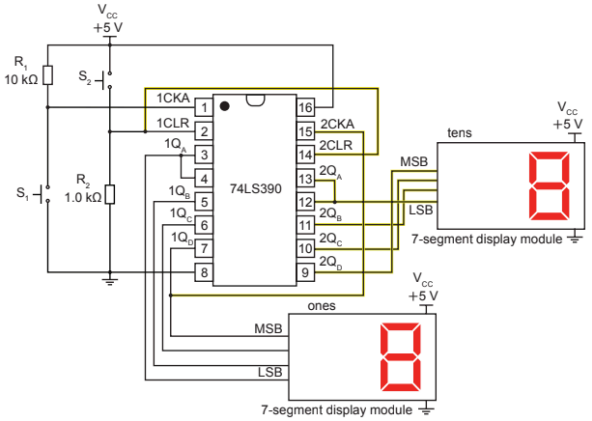
4-bit BCD				Decimal equivalent
Q_D	Q_C	Q_B	Q_A	
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9

How to connect (1 counter, count to 9):



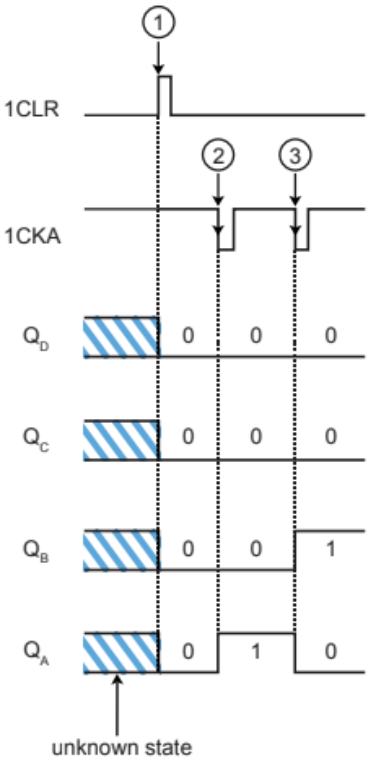
- Pin 3 ($1Q_A$) and 4 (1CKB) are connected together
- Pins 7, 6, 5, 3 are connected to a 7-segment display module that takes in the BCD input and displays the decimal equivalent
- Pin 1 (1CKA) is to be connected to a Pull-Up switch, while Pin 2 (1CLR) is to be connected to a Pull-Down switch
- S_1 must be a debounced switch to prevent contact bounce increasing the count value more than once

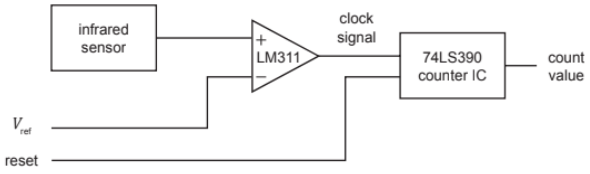
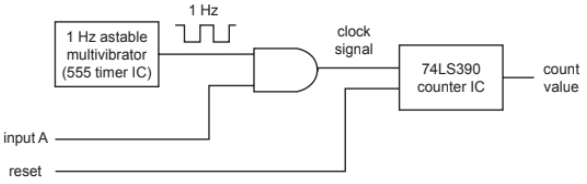
How to connect (2 counters, count to 99):



- Pin 13 ($2Q_A$) and 12 (2CKB) are connected together
- Pin 7 connected to Pin 15 so that $1Q_D$ functions as a clock for counter 2
- Pins 2 and 14 connected together

Timing diagram



	- Pins 9, 10, 11, 13 are connected to a 7-segment display module Function: - When switch S_2 is pressed, the count values for both counters is reset and the 7-segment displays will display 0 - When switch S_1 is pressed, the count value increases by 1, and the 7-segment display displays 1 - When the count value is 9 and switch S_1 is pressed, the count value is automatically reset to 0 while the count value for counter 2 increases by 1 Application: Electronic Carpark display  - When a car passes through a infrared sensor, LM311 Voltage comparator sends a falling edge input to the 74LS390 counter IC - Allows count value to increase by 1 for every car that passes through Application: Stopwatch  - When Input A is HIGH, the AND gate produces an falling edge input for the 74LS390 counter IC whenever the Astable output waveform is LOW - Allows the display to function as a stopwatch when it is activated	

(No chapter 15 as its under Coursework; refer to your teacher for help i guess)
(here are some tips tho: always look up the datasheet and pin diagrams of all the components ur using, ensure that your circuit is functioning properly before testing and ensure all necessary information have been taken down before moving on to the next section)