Performance-based assessments for digital circuit competencies

---

The purpose of these assessments is for instructors to accurately measure the learning of their electronics students, in a way that melds theoretical knowledge with hands-on application. In each assessment, students are asked to predict the behavior of a circuit from a schematic diagram and component values, then they build that circuit and measure its real behavior. If the behavior matches the predictions, the student then simulates the circuit on computer and presents the three sets of values to the instructor. If not, then the student then must correct the error(s) and once again compare measurements to predictions. Grades are based on the number of attempts required before all predictions match their respective measurements.

You will notice that no component values are given in this worksheet. The *instructor* chooses component values suitable for the students' parts collections, and ideally chooses different values for each student so that no two students are analyzing and building the exact same circuit. These component values may be hand-written on the assessment sheet, printed on a separate page, or incorporated into the document by editing the graphic image.

This is the procedure I envision for managing such assessments:

1. The instructor hands out individualized assessment sheets to each student.
2. Each student predicts their circuit's behavior at their desks using pencil, paper, and calculator (if appropriate).
3. Each student builds their circuit at their desk, under such conditions that it is impossible for them to verify their predictions using test equipment. Usually this will mean the use of a multimeter only (for measuring component values), but in some cases even the use of a multimeter would not be appropriate.
4. When ready, each student brings their predictions and completed circuit up to the instructor's desk, where any necessary test equipment is already set up to operate and test the circuit. There, the student sets up their circuit and takes measurements to compare with predictions.
5. If any measurement fails to match its corresponding prediction, the student goes back to their own desk with their circuit and their predictions in hand. There, the student tries to figure out where the error is and how to correct it.
6. Students repeat these steps as many times as necessary to achieve correlation between all predictions and measurements. The instructor's task is to count the number of attempts necessary to achieve this, which will become the basis for a percentage grade.
7. (OPTIONAL) As a final verification, each student simulates the same circuit on computer, using circuit simulation software (Spice, Multisim, etc.) and presenting the results to the instructor as a final pass/fail check.

These assessments more closely mimic real-world work conditions than traditional written exams:

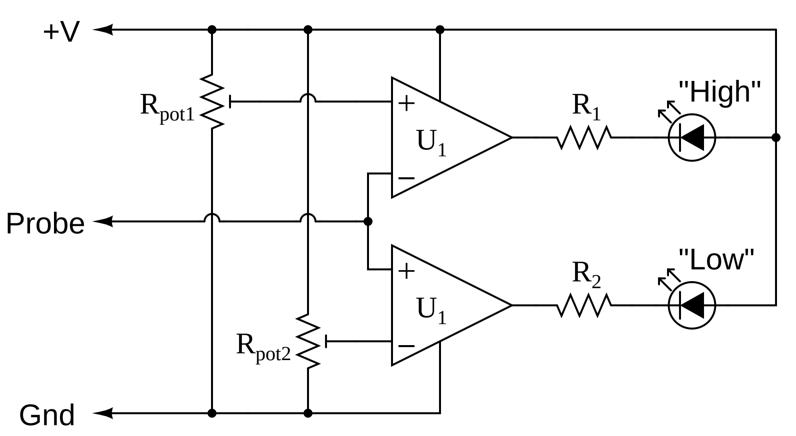
- Students cannot pass such assessments only knowing circuit theory or only having hands-on construction and testing skills – they must be proficient at both.
- Students do not receive the "authoritative answers" from the instructor. Rather, they learn to validate their answers through real circuit measurements.
- Just as on the job, the work isn't complete until *all errors* are corrected.
- Students must recognize and correct their own errors, rather than having someone else do it for them.
- Students must be fully prepared on exam days, bringing not only their calculator and notes, but also their tools, breadboard, and circuit components.

Instructors may elect to reveal the assessments before test day, and even use them as preparatory labwork and/or discussion questions. Remember that there is absolutely nothing wrong with "teaching to the test" *so long as the test is valid*. Normally, it is bad to reveal test material in detail prior to test day, lest students merely memorize responses in advance. With performance-based assessments, however, there is no way to pass without truly understanding the subject(s).

Question 1

Competency: **Logic probe circuit** Version:

Schematic



Given conditions

V_{High} =

$R_{pot1} = R_{pot2}$ =

V_{Low} =

$R_1 = R_2$ =

Parameters

	Predicted	Tested	
LED status			$V_{probe} \geq V_{High}$
LED status			$V_{probe} \leq V_{Low}$

Fault analysis

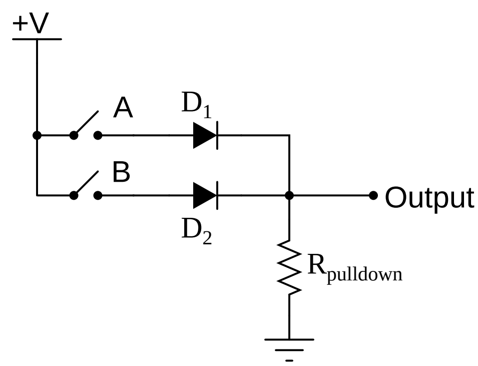
Suppose component ______ fails ☐ open ☐ shorted ☐ other ______

What will happen in the circuit?

file 02851

Competency: **OR gate, diode-resistor logic** Version:

Schematic



Given conditions

V_{DD} = $R_{pulldown}$ =

Truth table

Predicted

A	B	Output
0	0	
0	1	
1	0	
1	1	

Actual

A	B	Output
0	0	
0	1	
1	0	
1	1	

Fault analysis

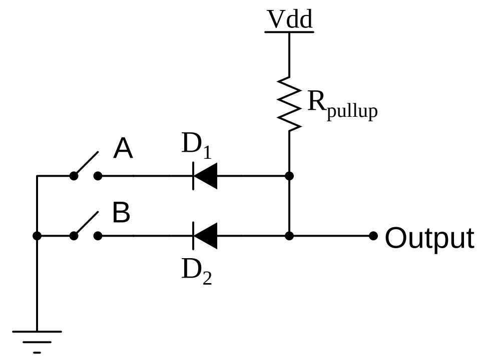
Suppose component ☐ fails ☐ open ☐ other ______ ☐ shorted

What will happen in the circuit?

file 02787

Competency: **AND gate, diode-resistor logic** Version:

Schematic



Given conditions

V_{DD} = R_{pullup} =

Truth table

Predicted

A	B	Output
0	0	
0	1	
1	0	
1	1	

Actual

A	B	Output
0	0	
0	1	
1	0	
1	1	

Fault analysis

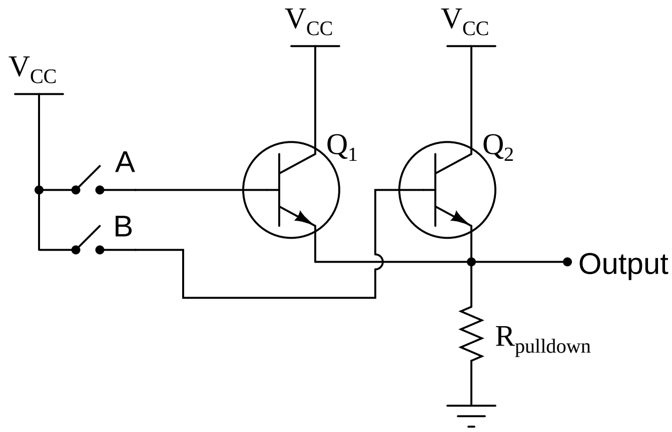
Suppose component ☐ fails ☐ open ☐ shorted ☐ other ________

What will happen in the circuit?

file 02788

Competency: **OR gate, simple BJT logic** Version:

Schematic



Given conditions

V_{CC} = $R_{pulldown}$ =

Truth table

Predicted

A	B	Output
0	0	
0	1	
1	0	
1	1	

Actual

A	B	Output
0	0	
0	1	
1	0	
1	1	

Fault analysis

Suppose component ☐ fails ☐ open ☐ other ______ ☐ shorted

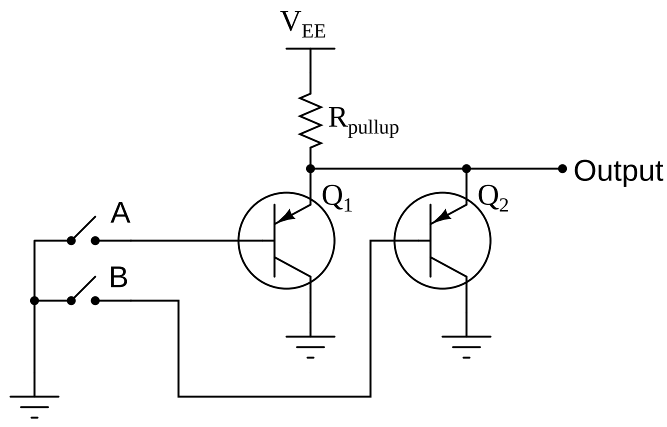
What will happen in the circuit?

file 02794

Question 5

Competency: **AND gate, simple BJT logic** Version:

Schematic



Given conditions

V_{EE} = R_{pullup} =

Truth table

Predicted

A	B	Output
0	0	
0	1	
1	0	
1	1	

Actual

A	B	Output
0	0	
0	1	
1	0	
1	1	

Fault analysis

Suppose component ☐ fails ☐ open ☐ other ______ ☐ shorted

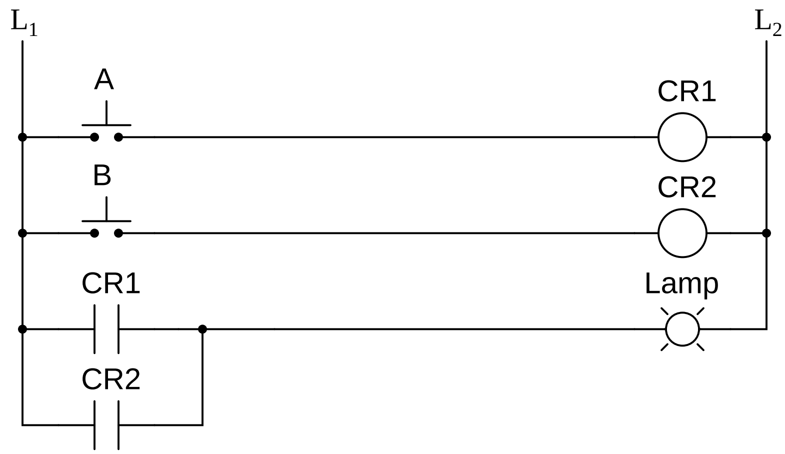
What will happen in the circuit?

file 02793

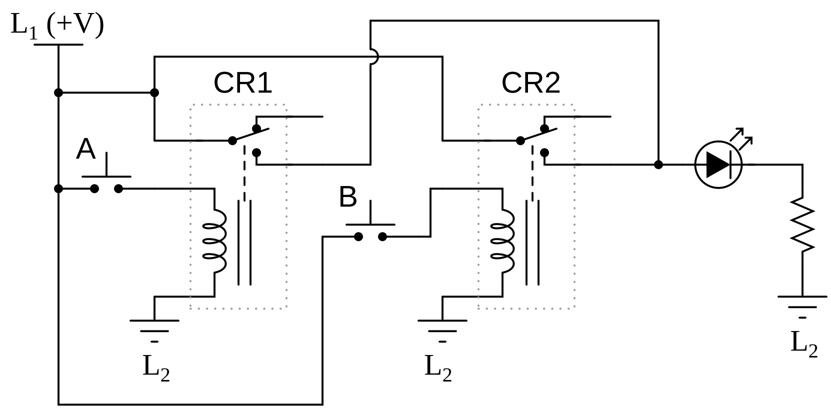
Question 6

Competency: **OR gate, relay logic** Version:

Ladder logic diagram



Schematic



Truth table

Predicted

A	B	Output
0	0	
0	1	
1	0	
1	1	

Actual

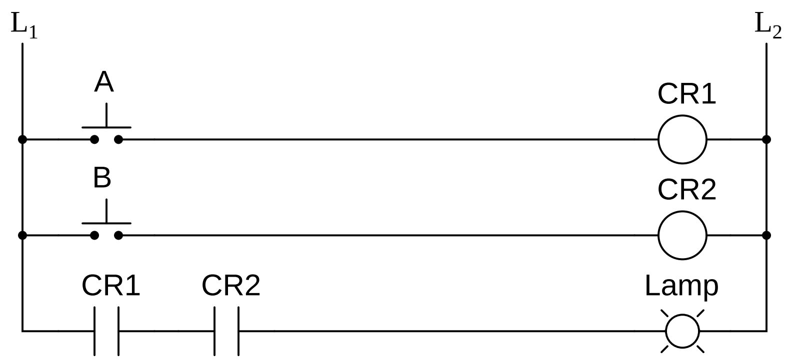
A	B	Output
0	0	
0	1	
1	0	
1	1	

file 02792

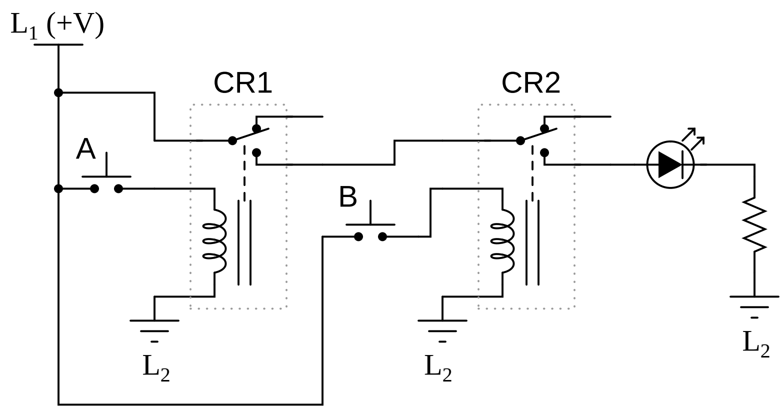
Question 7

Competency: **AND gate, relay logic** Version:

Ladder logic diagram



Schematic



Truth table

Predicted

A	B	Output
0	0	
0	1	
1	0	
1	1	

Actual

A	B	Output
0	0	
0	1	
1	0	
1	1	

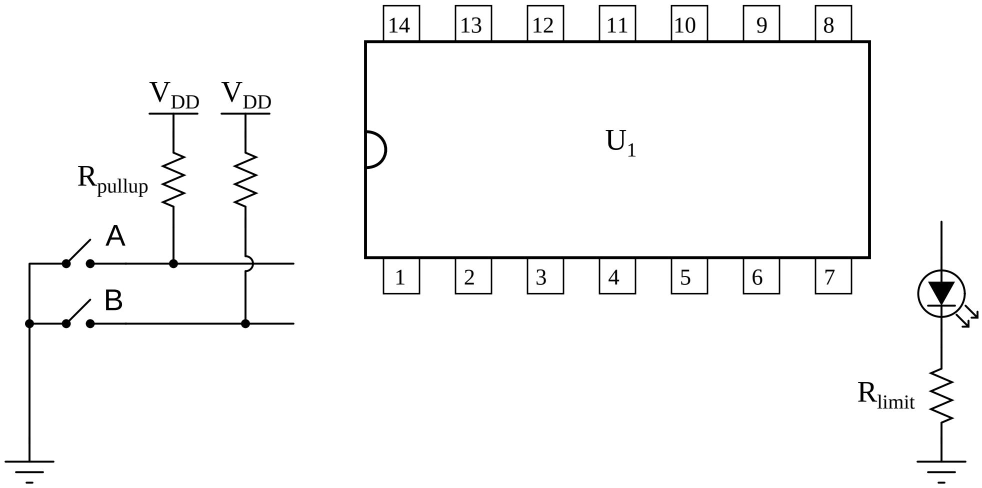
file 02791

Competency: **IC logic gate usage** Version:

Description

Connect one of the IC's two-input gates to the switches and to the LED, then prove your prediction for that gate's truth table.

Diagram



Given conditions *(instructor chooses IC part number)*

V_{DD} = R_{pullup} = R_{limit} = U_1 =

Truth table

Predicted

A	B	Output
0	0	
0	1	
1	0	
1	1	

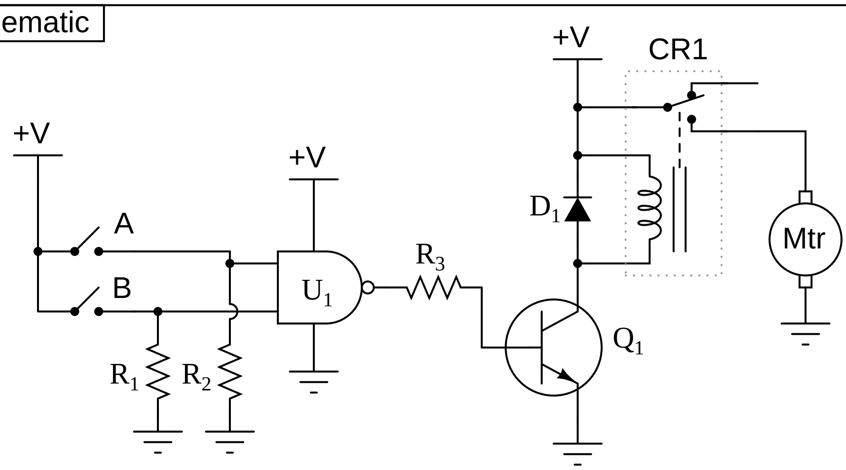
Actual

A	B	Output
0	0	
0	1	
1	0	
1	1	

file 02789

Competency: **Gate-relay interposing** Version:

Schematic



Given conditions

+V = $R_1 = R_2 =$ $R_3 =$

Truth table

Predicted

A	B	Output
0	0	
0	1	
1	0	
1	1	

Actual

A	B	Output
0	0	
0	1	
1	0	
1	1	

Fault analysis

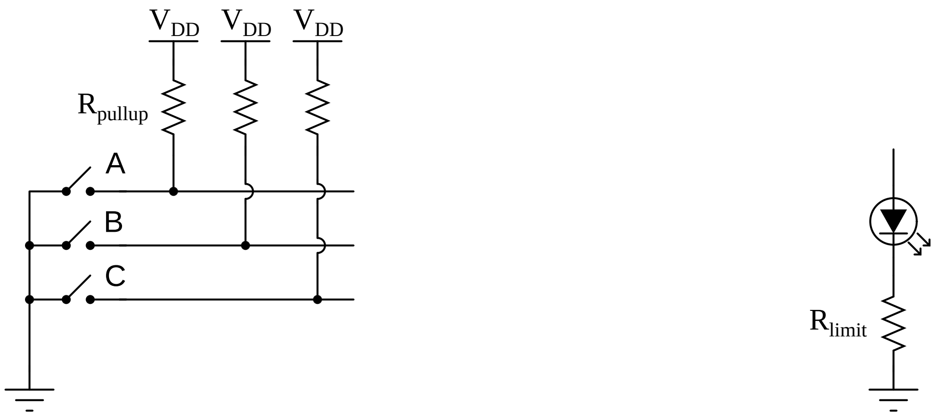
Suppose component ☐ fails ☐ open ☐ other ________ ☐ shorted

What will happen in the circuit?

file 02795

Competency: **Combinational logic circuit** Version:

Schematic



Given conditions

V_{DD} = R_{pullup} = R_{limit} =

Truth table

Predicted

A	B	C	Output
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

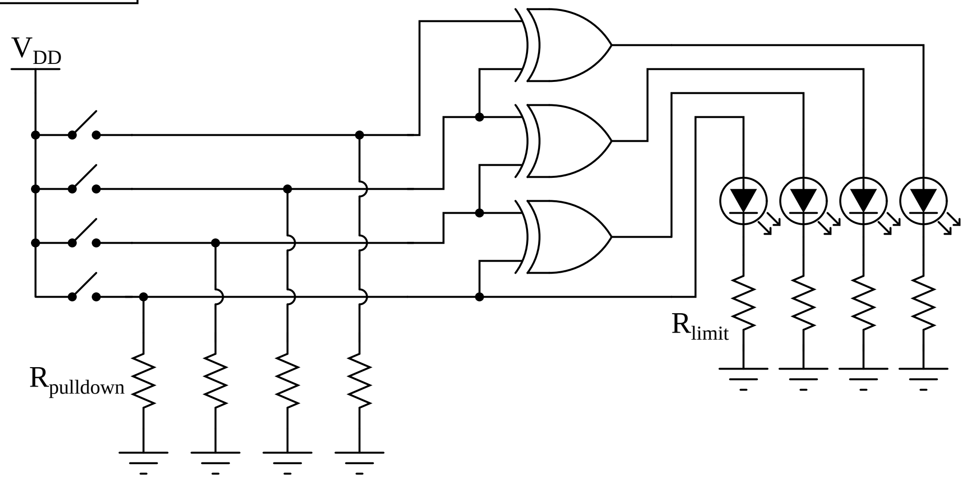
Actual

A	B	C	Output
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

file 01620

Competency: **Binary to Gray code converter** Version:

Schematic



Given conditions

V_{DD} = $R_{pulldown}$ = R_{limit} =

Truth table

Predicted

Binary	*Gray*
0 0 0 0	
0 0 0 1	
0 0 1 0	
0 0 1 1	
0 1 0 0	
0 1 0 1	
0 1 1 0	
0 1 1 1	
1 0 0 0	
1 0 0 1	
1 0 1 0	
1 0 1 1	
1 1 0 0	
1 1 0 1	
1 1 1 0	
1 1 1 1	

Actual

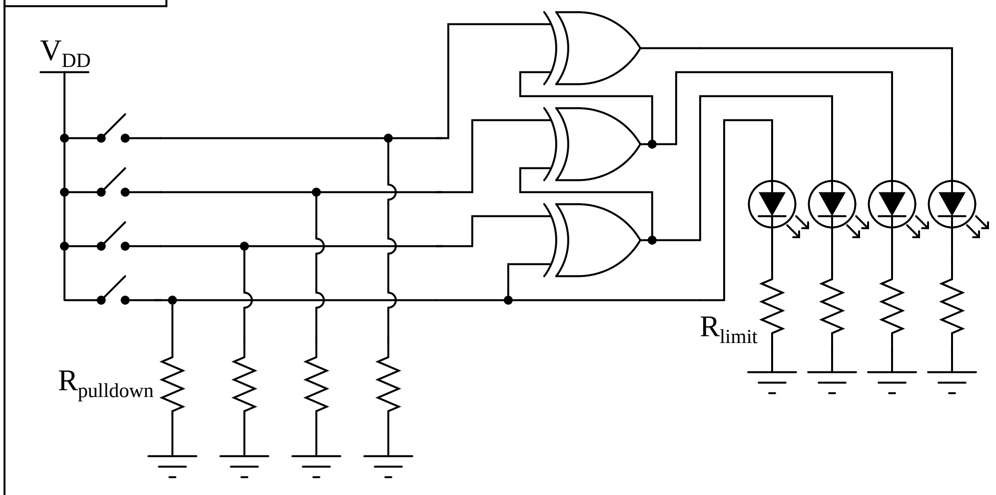
Binary	*Gray*
0 0 0 0	
0 0 0 1	
0 0 1 0	
0 0 1 1	
0 1 0 0	
0 1 0 1	
0 1 1 0	
0 1 1 1	
1 0 0 0	
1 0 0 1	
1 0 1 0	
1 0 1 1	
1 1 0 0	
1 1 0 1	
1 1 1 0	
1 1 1 1	

file 02855

Question 12

Competency: **Gray to Binary code converter** Version:

Schematic



Given conditions

V_{DD} = $R_{pulldown}$ = R_{limit} =

Truth table

Predicted

Gray	*Binary*
0 0 0 0	
0 0 0 1	
0 0 1 1	
0 0 1 0	
0 1 1 0	
0 1 1 1	
0 1 0 1	
0 1 0 0	
1 1 0 0	
1 1 0 1	
1 1 1 1	
1 1 1 0	
1 0 1 0	
1 0 1 1	
1 0 0 1	
1 0 0 0	

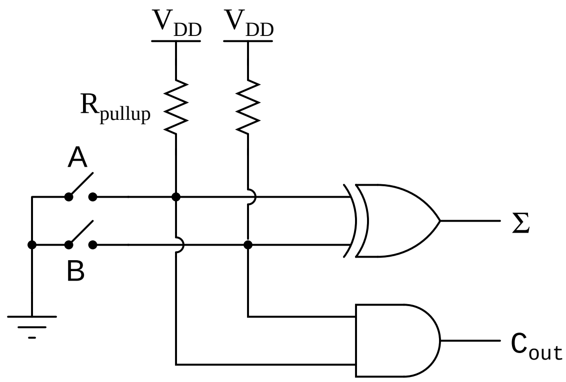
Actual

Gray	*Binary*
0 0 0 0	
0 0 0 1	
0 0 1 1	
0 0 1 0	
0 1 1 0	
0 1 1 1	
0 1 0 1	
0 1 0 0	
1 1 0 0	
1 1 0 1	
1 1 1 1	
1 1 1 0	
1 0 1 0	
1 0 1 1	
1 0 0 1	
1 0 0 0	

file 02856

Competency: **Half adder circuit** Version:

Schematic



Given conditions

V_{DD} = R_{pullup} = R_{limit} =

Truth table

Predicted

A	B	C_{out}	Σ
0	0		
0	1		
1	0		
1	1		

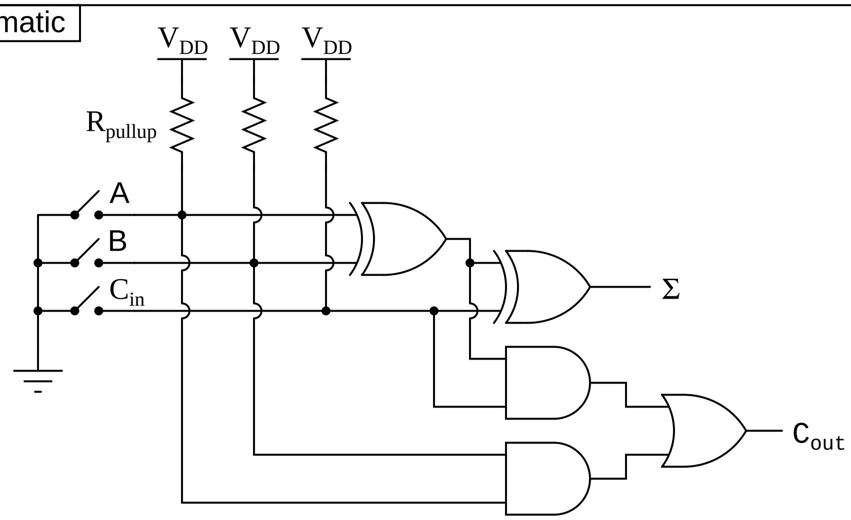
Actual

A	B	C_{out}	Σ
0	0		
0	1		
1	0		
1	1		

file 02857

Competency: **Full adder circuit** Version:

Schematic



Given conditions

V_{DD} = R_{pullup} = R_{limit} =

Truth table

Predicted

A	B	C_{in}	C_{out}	Σ
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

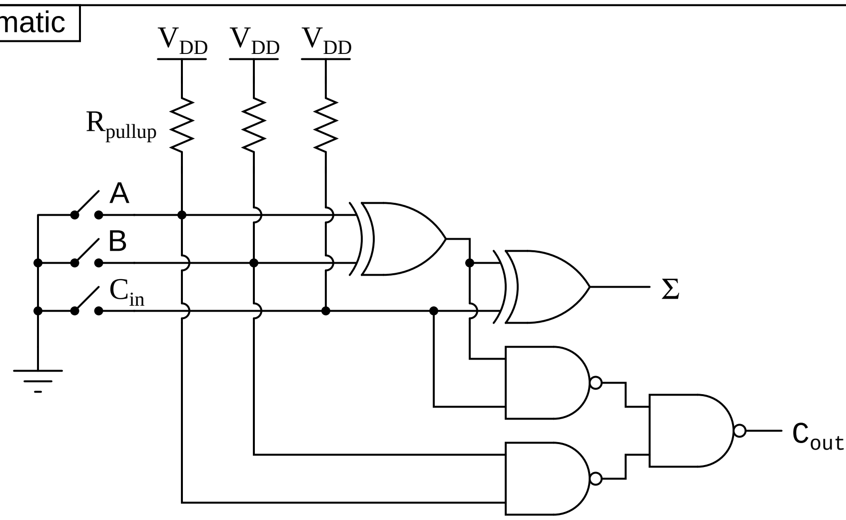
Actual

A	B	C_{in}	C_{out}	Σ
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

file 02858

Competency: **Full adder circuit w/ NAND gates**	Version:

Schematic



Given conditions

V_{DD} = R_{pullup} = R_{limit} =

Truth table

Predicted

A	B	C_{in}	C_{out}	Σ
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

Actual

A	B	C_{in}	C_{out}	Σ
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

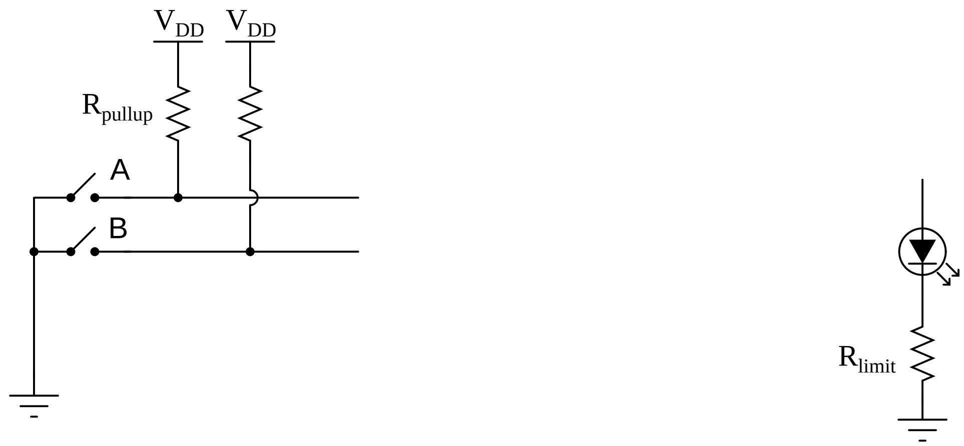
file 02859

Competency: **Gate circuit from Boolean expression**	Version:

Boolean expression

Output =

Schematic



Truth table

Predicted

A	B	Output
0	0	
0	1	
1	0	
1	1	

Actual

A	B	Output
0	0	
0	1	
1	0	
1	1	

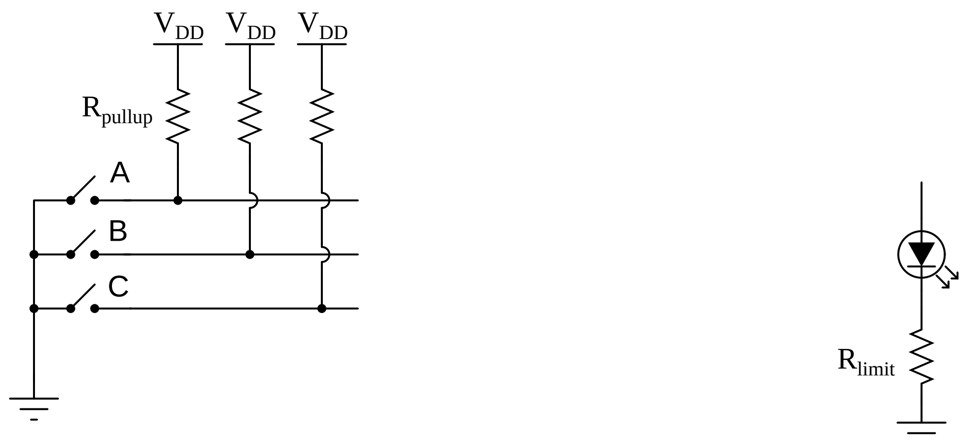
file 02809

Competency: **Gate circuit from Boolean expression** Version:

Boolean expression

Output =

Schematic



Truth table

Predicted

A	B	C	Output
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

Actual

A	B	C	Output
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

file 02810

Competency: **Gate circuit from truth table** Version:

Truth table

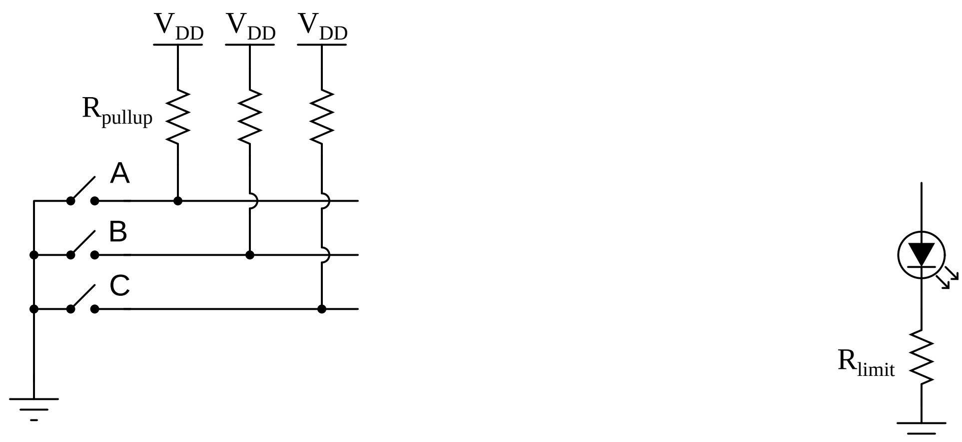
Given

A	B	C	Output
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

Actual

A	B	C	Output
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

Schematic



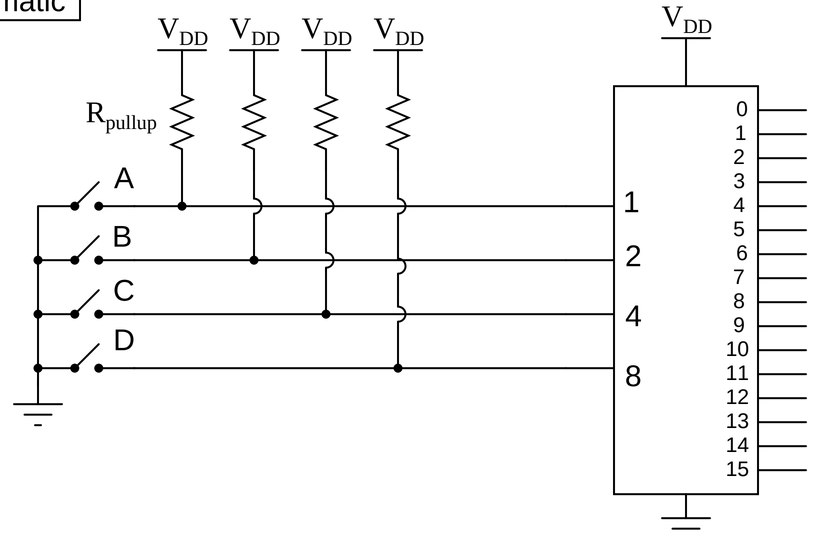
file 02134

Competency: **4-line to 16-line decoder IC** Version:

Truth table

D	C	B	A	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	0																
0	0	0	1																
0	0	1	0																
0	0	1	1																
0	1	0	0																
0	1	0	1																
0	1	1	0																
0	1	1	1																
1	0	0	0																
1	0	0	1																
1	0	1	0																
1	0	1	1																
1	1	0	0																
1	1	0	1																
1	1	1	0																
1	1	1	1																

Schematic



file 03009

Competency: **Arbitrary logic function using mux** Version:

Truth table

Given

C	B	A	Output
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

Actual

C	B	A	Output
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

Schematic

You draw it, in its entirety!

file 03008

Competency: **NAND gate universality** Version:

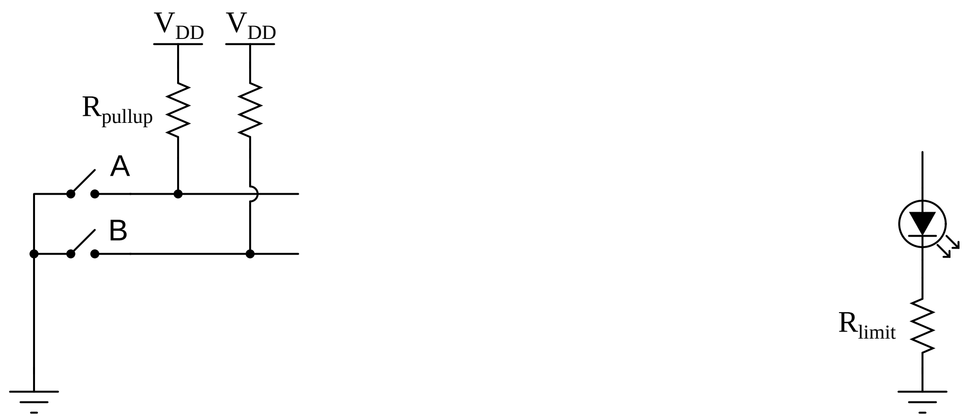
Description

Emulate the specified logic function using nothing but interconnected NAND gates.

Emulated function *(instructor checks one box)*

☐ AND ☐ OR ☐ NOR

Diagram



Truth table

Predicted

A	B	Output
0	0	
0	1	
1	0	
1	1	

Actual

A	B	Output
0	0	
0	1	
1	0	
1	1	

file 02807

Competency: **NOR gate universality** Version:

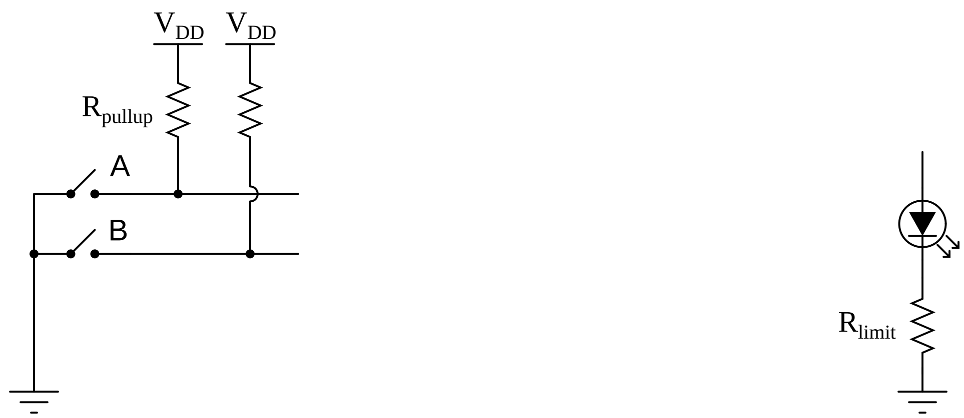
Description

Emulate the specified logic function using nothing but interconnected NOR gates.

Emulated function *(instructor checks one box)*

☐ AND ☐ OR ☐ NAND

Diagram



Truth table

Predicted

A	B	Output
0	0	
0	1	
1	0	
1	1	

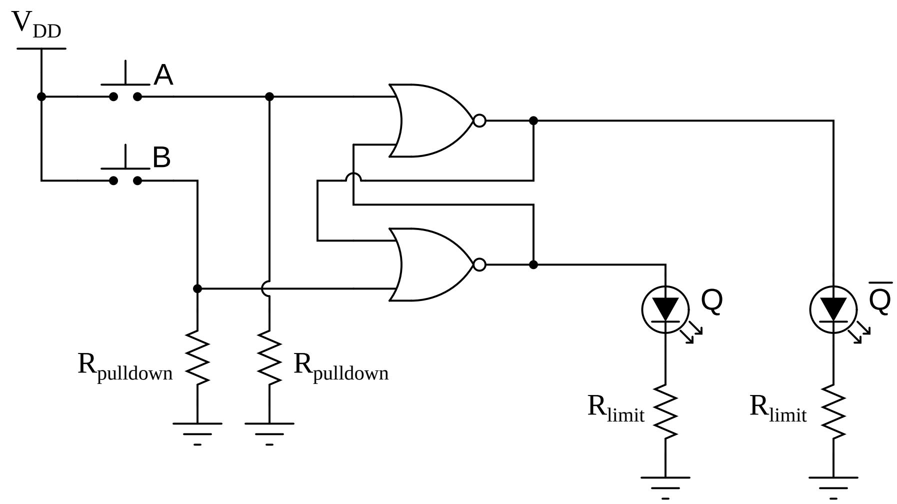
Actual

A	B	Output
0	0	
0	1	
1	0	
1	1	

file 02808

Competency: **S-R latch using NOR gates** Version:

Schematic



Given conditions

V_{DD} = $R_{pulldown}$ = R_{limit} =

Truth table

Predicted

A	B	Q	$\overline{Q}$
0	0		
0	1		
1	0		
1	1		

Actual

A	B	Q	$\overline{Q}$
0	0		
0	1		
1	0		
1	1		

Experiment

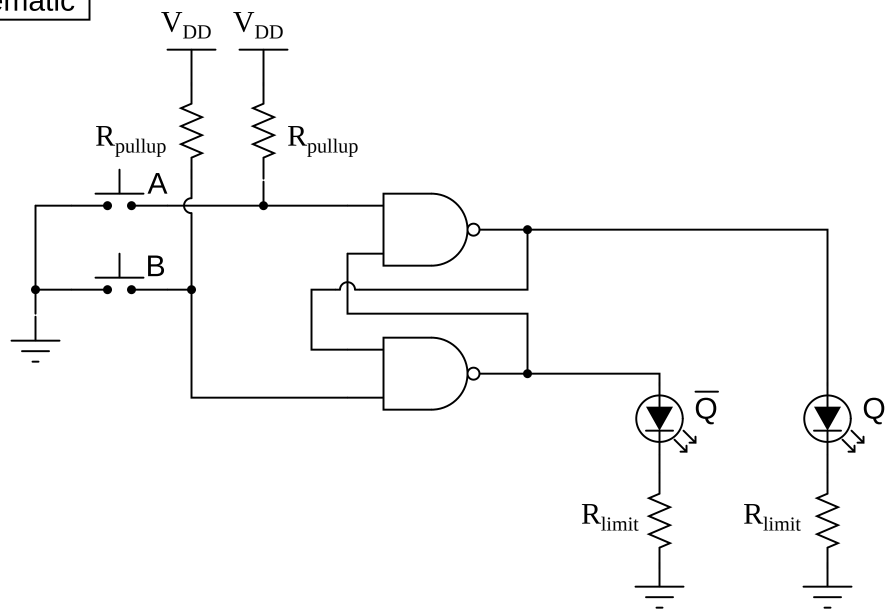
What happens when you use different values of R_{limit} resistance?

Hint: try 10 Ω versus 1000 Ω to see the difference it makes!

file 03988

Competency: **S-R latch using NAND gates** Version:

Schematic



Given conditions

V_{DD} = R_{pullup} = R_{limit} =

Truth table

Predicted

A	B	Q	$\overline{Q}$
0	0		
0	1		
1	0		
1	1		

Actual

A	B	Q	$\overline{Q}$
0	0		
0	1		
1	0		
1	1		

Experiment

What happens when you use different values of R_{limit} resistance?

Hint: try 10 Ω versus 1000 Ω to see the difference it makes!

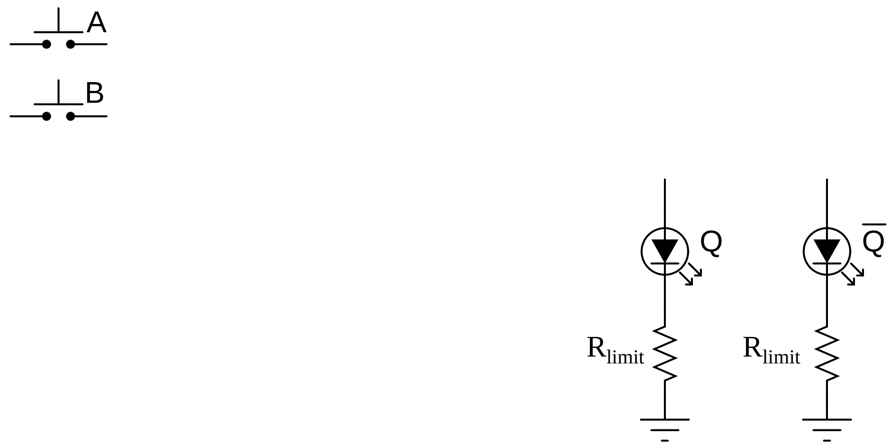
file 03989

Competency: **S-R latch circuit** Version:

Description

Build an S-R latch circuit using either NAND or NOR gates

Schematic



Truth table

Predicted

A	B	Q	$\overline{Q}$
0	0		
0	1		
1	0		
1	1		

Actual

A	B	Q	$\overline{Q}$
0	0		
0	1		
1	0		
1	1		

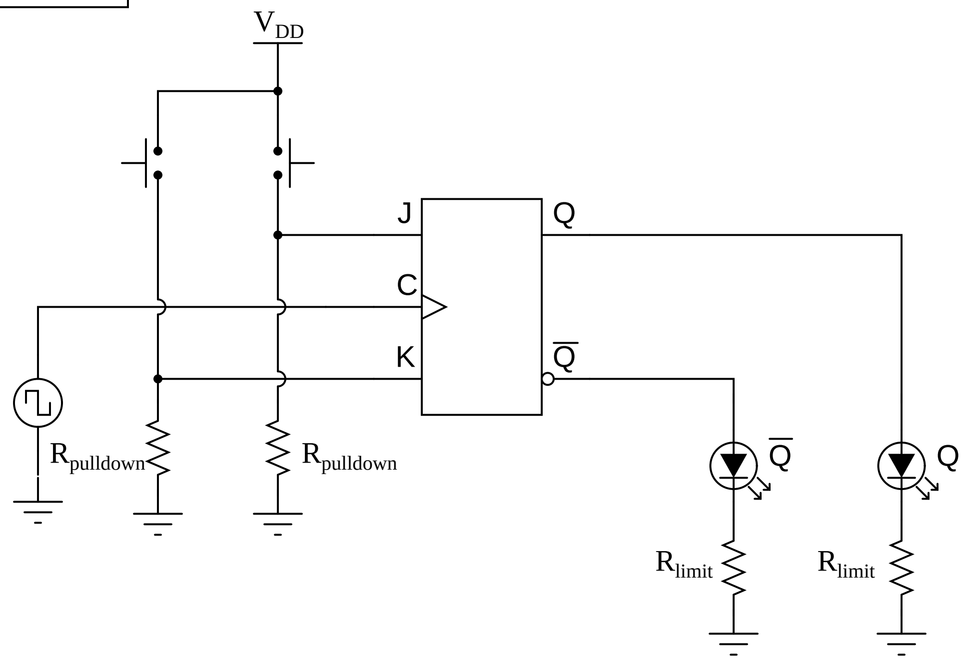
file 01621

Competency: **J-K flip-flop IC** Version:

Description

Demonstrate the "set," "reset," and "toggle" modes of a J-K flip-flop integrated circuit.

Schematic



Parameters

"Set" mode demonstrated ☐

"Reset" mode demonstrated ☐

"Toggle" mode demonstrated ☐

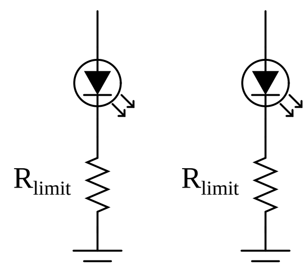
file 02900

Competency: **2-bit flip-flop counter circuit** Version:

Description

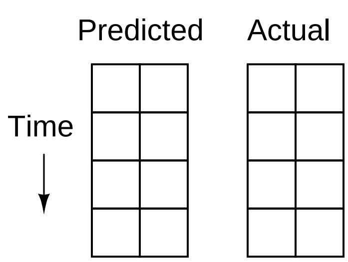
Build a 2-bit counter circuit using individual J-K flip-flops.

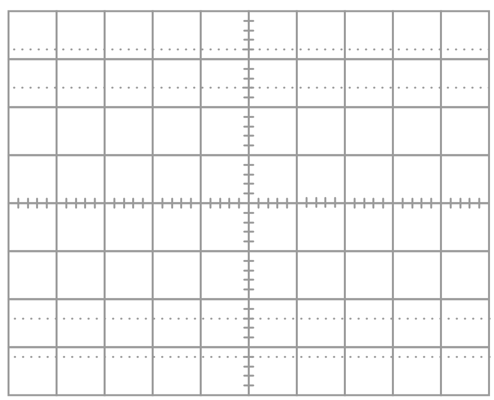
Schematic



Count sequence

Output timing diagram as shown by oscilloscope





Identify the counting states on the oscilloscope display (00, 01, 10, and 11).

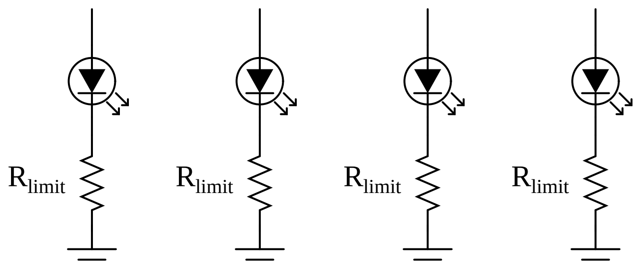
file 02947

Competency: **4-bit flip-flop counter circuit** Version:

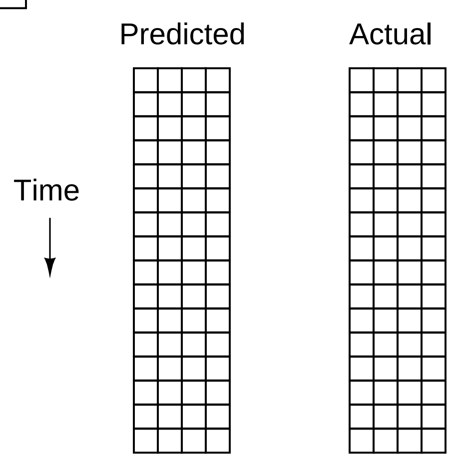
Description

Build a 4-bit counter circuit using individual J-K flip-flops

Schematic



Count sequence



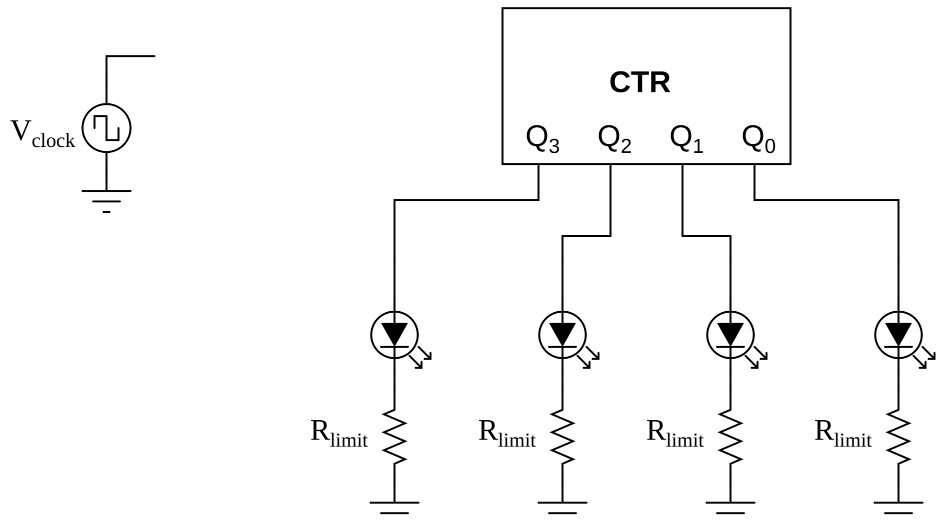
file 02135

Competency: **4-bit up/down counter IC**	Version:

Description

Configure a 4-bit counter IC to count either up or down depending on the position of a selector switch. Complete the schematic diagram to show the switch and all other necessary components/connections.

Schematic



Count sequence

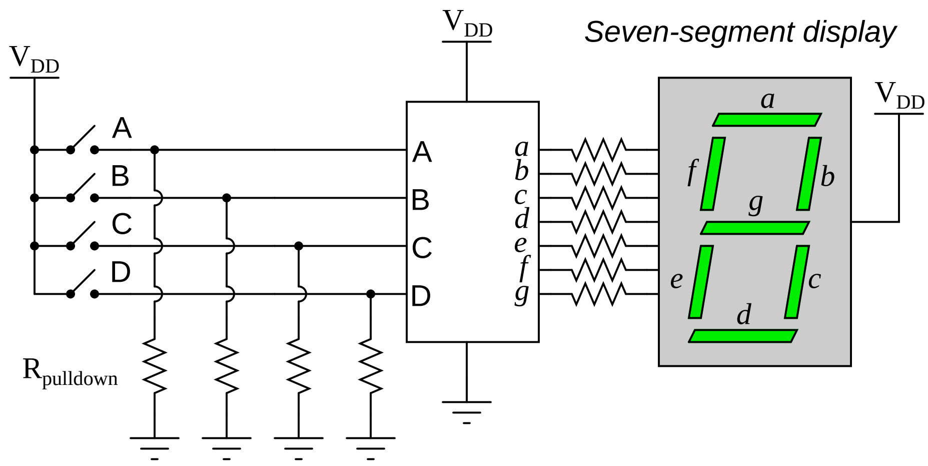
Counts in the "up" direction ☐

Counts in the "down" direction ☐

file 02957

Competency: **BCD to 7-segment decoder/driver IC** Version:

Schematic



Parameters

All numerals (0 through 9) demonstrated ☐

Fault analysis

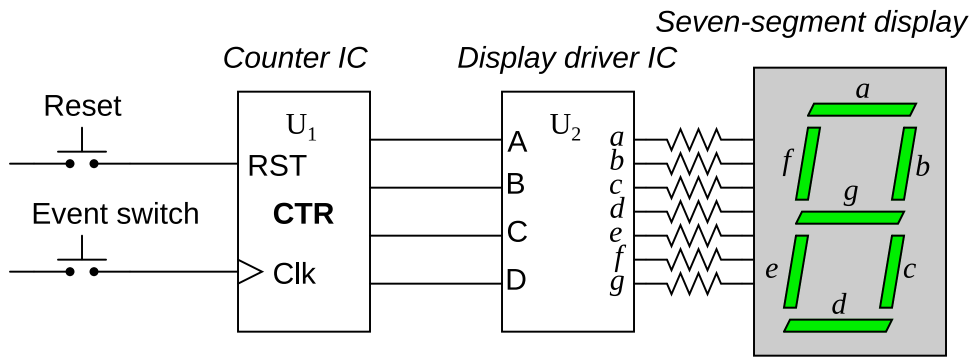
Suppose component ☐ fails ☐ open ☐ other ______ ☐ shorted

What will happen in the circuit?

file 03010

Competency: **Decade counter circuit** Version:

Schematic



Details purposely omitted from schematic diagram

Given conditions

U_1 =

U_2 =

Parameters

Counter increments with each physical event, counting from 0 to 9 and then resetting back to 0 again. Count sequence exhibits no skipped counts and no missed events.

☐ YES

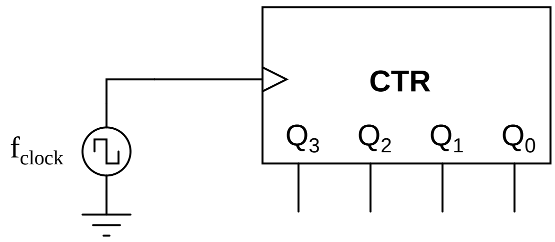
☐ NO

file 03851

Question 32

Competency: **Binary counter as frequency divider** Version:

Schematic



Given conditions

f_{clock} =

Parameters

	Predicted	Measured
f_{Q0}		
f_{Q1}		
f_{Q2}		
f_{Q3}		

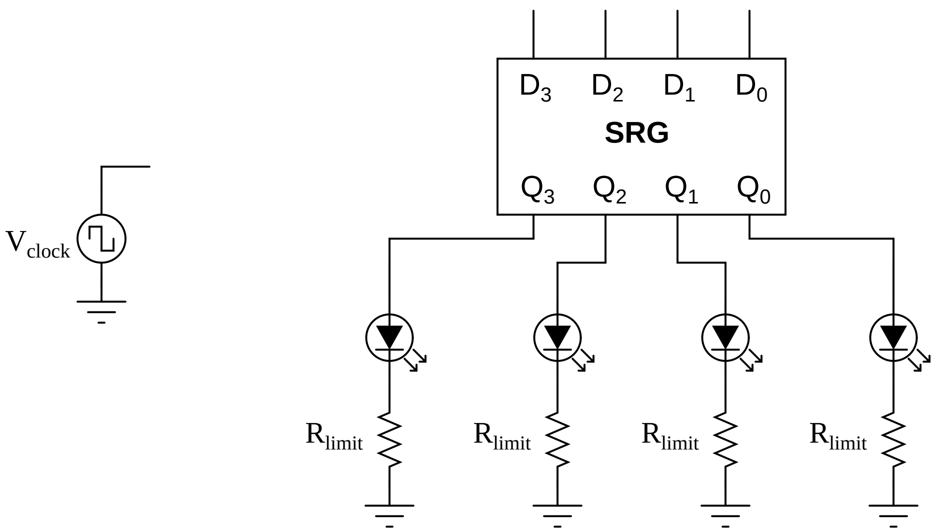
file 02959

Competency: **4-bit universal shift register IC** Version:

Description

Configure a 4-bit universal shift register IC to load parallel data, then shift in both directions. Complete the schematic diagram to show all switches and other necessary components/connections.

Schematic



Shift sequence

Loads parallel data ☐

Shifts right ☐

Shifts left ☐

file 02958

Competency: **Stepper motor drive circuit**	Version:
Description	
Design a simple circuit for driving a stepper motor using four pushbutton switches (pushing the switches in the correct sequence will cause the motor to turn).	
Schematic	

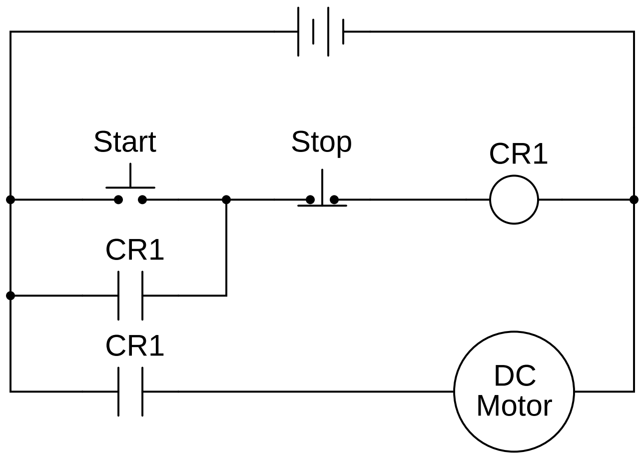
file 01619

Competency: **Relay start/stop motor control circuit** Version:

Description

Build a start/stop motor control circuit using
an electromechanical relay and two pushbutton switches

Schematic



Truth table

Predicted

Start	Stop	Motor
0	0	
0	1	
1	0	
1	1	

Actual

Start	Stop	Motor
0	0	
0	1	
1	0	
1	1	

Fault analysis

Suppose component ☐ fails ☐ open ☐ other ______
☐ shorted

What will happen in the circuit?

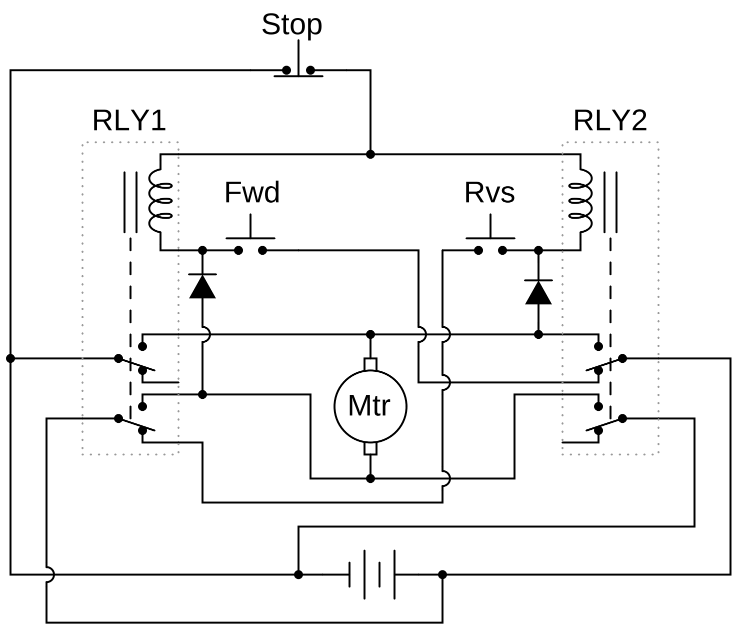
file 02362

Competency: **Relay start/stop motor control circuit** Version:
Description
Build a start/stop motor control circuit using an electro-mechanical relay and two (momentary) pushbutton switches
Schematic
Parameters
Motor latches in the energized state when "Start" switch is pressed and released ☐
Motor latches in the de-energized state when "Stop" switch is pressed and released ☐

file 02390

Competency: **Reversing start/stop motor control circuit** Version:

Schematic



Parameters

Motor starts and latches in "Forward" ☐

Motor starts and latches in "Reverse" ☐

file 03151

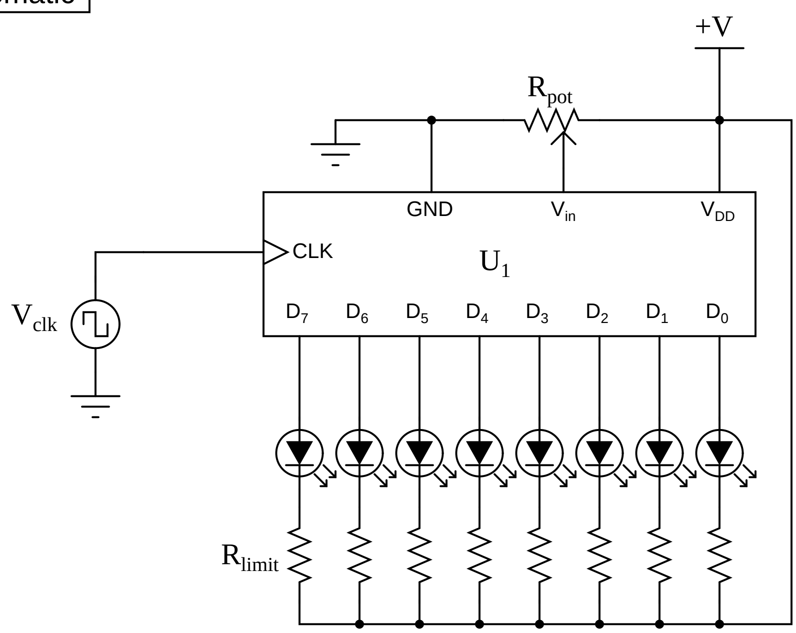
Question 38

Competency: **Analog-digital converter IC** Version:

Description

Demonstrate the operation of an analog-to-digital converter integrated circuit, using a potentiometer as the variable input signal source.

Schematic



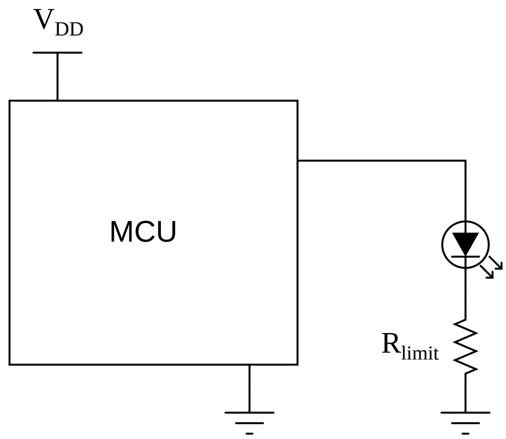
Parameters

	Predicted	Measured
$V_{resolution}$		

file 02950

Competency: **Light flasher using MCU** Version:

Schematic



Given conditions

t_{on} = t_{off} =

Program

file 03152

Competency: **Logic function using MCU**	Version:

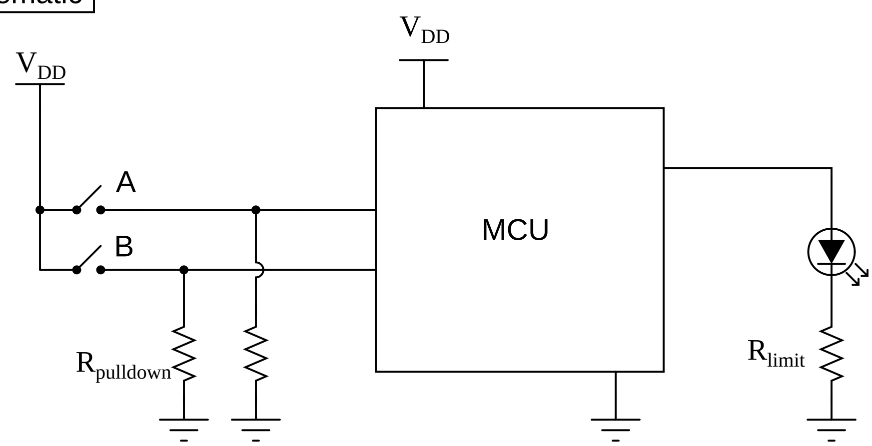
Description

Program a microcontroller (MCU) to emulate
a two-input logic gate of the instructor's choosing.

Emulated function *(instructor checks one box)*

☐ AND ☐ OR ☐ NAND ☐ NOR

Schematic



Program

file 03150

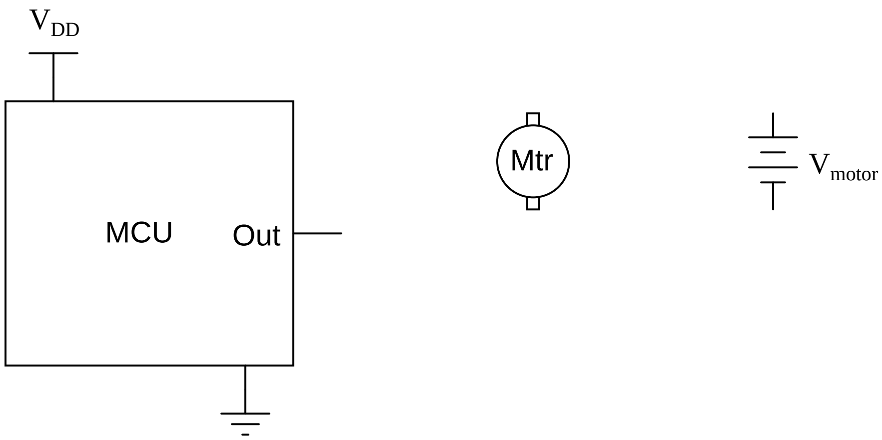
Question 41

Competency: **Motor speed control using MCU** Version:

Description

Use the Pulse-Width Modulation (PWM) capability of a microcontroller to set the speed of a DC motor.

Schematic



Given conditions

V_{motor} =

D (duty cycle) =

Program

file 04018

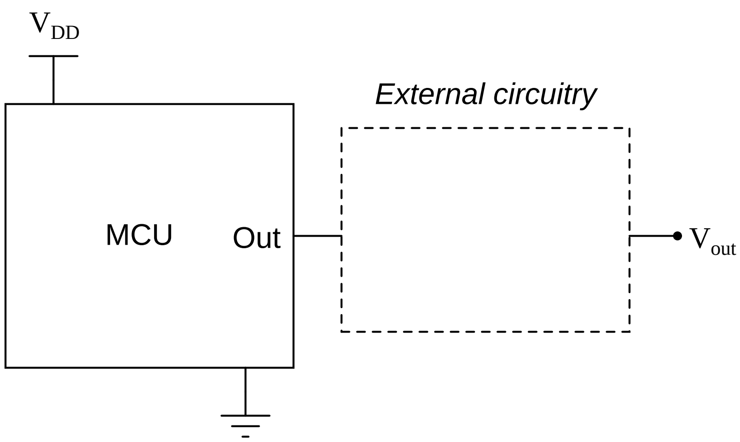
Question 42

Competency: **MCU analog voltage output** Version:

Description

Generate an analog output voltage using an MCU.

Schematic



Given conditions

V_{out} =

Program

file 04019

Question 43

(Template)

Competency: Version:

Schematic

Given conditions

Parameters

Predicted	Measured

file 01602

Answer 1

Use circuit simulation software to verify your predicted and measured parameter values.

Answer 2

Use circuit simulation software to verify your predicted and actual truth tables.

Answer 3

Use circuit simulation software to verify your predicted and actual truth tables.

Answer 4

Use circuit simulation software to verify your predicted and actual truth tables.

Answer 5

Use circuit simulation software to verify your predicted and actual truth tables.

Answer 6

Use circuit simulation software to verify your predicted and actual truth tables.

Answer 7

Use circuit simulation software to verify your predicted and actual truth tables.

Answer 8

Use circuit simulation software to verify your predicted and actual truth tables.

Answer 9

Use circuit simulation software to verify your predicted and actual truth tables.

Answer 10

Use circuit simulation software to verify your predicted and actual truth tables.

Answer 11

Use circuit simulation software to verify your predicted and actual truth tables.

Answer 12

Use circuit simulation software to verify your predicted and actual truth tables.

Answer 13

Use circuit simulation software to verify your predicted and actual truth tables.

Answer 14

Use circuit simulation software to verify your predicted and actual truth tables.

Answer 15

Use circuit simulation software to verify your predicted and actual truth tables.

Answer 16

Use circuit simulation software to verify your predicted and actual truth tables.

Answer 17

Use circuit simulation software to verify your predicted and actual truth tables.

Answer 18

Use circuit simulation software to verify your predicted and actual truth tables.

Answer 19

Use circuit simulation software to verify your predicted and actual truth tables.

Answer 20

Use circuit simulation software to verify your predicted and actual truth tables.

Answer 21

Use circuit simulation software to verify your predicted and actual truth tables.

Answer 22

Use circuit simulation software to verify your predicted and actual truth tables.

Answer 23

Use circuit simulation software to verify your predicted and actual truth tables.

Answer 24

Use circuit simulation software to verify your predicted and actual truth tables.

Answer 25

Use circuit simulation software to verify your predicted and actual truth tables.

Answer 26

Use circuit simulation software to verify your predicted and actual truth tables.

Answer 27

Use circuit simulation software to verify your predicted and actual truth tables.

Answer 28

Use circuit simulation software to verify your predicted and actual truth tables.

Answer 29

Use circuit simulation software to verify your predicted and actual truth tables.

Answer 30

Use circuit simulation software to verify your predicted and actual truth tables.

Answer 31

Use circuit simulation software to verify your predicted and measured parameter values.

Answer 32

Use circuit simulation software to verify your predicted and actual truth tables.

Answer 33

Use circuit simulation software to verify your predicted and actual truth tables.

Answer 34

The real circuit you build will validate your circuit design.

Answer 35

The real circuit you build will validate your circuit design.

Answer 36

The real circuit you build will validate your circuit design.

Answer 37

The real circuit you build will validate your circuit design.

Answer 38

Use circuit simulation software to verify your predicted and actual truth tables.

Answer 39

Use circuit simulation software to verify your predicted and actual truth tables.

Answer 40

Use circuit simulation software to verify your predicted and actual truth tables.

Answer 41

Use circuit simulation software to verify your predicted and actual truth tables.

Answer 42

Use circuit simulation software to verify your predicted and actual truth tables.

Answer 43

Here, you would indicate where or how to obtain answers for the requested parameters, but not actually give the figures. My stock answer here is "use circuit simulation software" (Spice, Multisim, etc.).

Notes 1

I recommend the use of the LM339 comparator for this circuit.

An extension of this exercise is to incorporate troubleshooting questions. Whether using this exercise as a performance assessment or simply as a concept-building lab, you might want to follow up your students' results by asking them to predict the consequences of certain circuit faults.

Notes 2

The choice of diodes is not critical in a circuit where the inputs come from manually actuated switches. Neither is the choice of resistor sizes (10 kΩ) works just fine, especially when the only thing loading the output is the student's logic probe or voltmeter.

Notes 3

It should be noted that the input states in this circuit are defined by the voltage levels, not by the contact status. In other words, a closed contact equals a "low" (0) logic state.

The choice of diodes is not critical in a circuit where the inputs come from manually actuated switches. Neither is the choice of resistor sizes (10 kΩ) works just fine, especially when the only thing loading the output is the student's logic probe or voltmeter.

Notes 4

Nothing special to note here!

Notes 5

It needs to be understood that this is an AND gate only if you consider the "0" and "1" states as defined by voltage levels with respect to ground, and not by switch actuation. Many students assume an actuated (pushed) switch is a "1" input and a de-actuated (unpushed) switch is a "0" input. Not necessarily so! In this circuit, the switches are connecting inputs to *ground*. This means a closed (actuated) switch provides a low (0) input state, while an open (unactuated) switch provides a high (1) input state.

Notes 6

The transition from ladder logic diagram to actual relay wiring is a confusing one for many students. This is what I consider to be the most significant learning objective of this exercise: figuring out how to build the circuit, not necessarily understanding the logical function of it.

Notes 7

The transition from ladder logic diagram to actual relay wiring is a confusing one for many students. This is what I consider to be the most significant learning objective of this exercise: figuring out how to build the circuit, not necessarily understanding the logical function of it.

Notes 8

The purpose of this exercise is for students to research what type of IC this is (from the given part number for U_1), its pinout, and then predict and prove its operation using truth tables to document the results. You, as the instructor, may select any 14-pin CMOS or TTL logic IC that you wish. Students are to draw the logic gate symbol within the rectangle of U_1, then connect that symbol to the input switches and output LED.

It needs to be understood that the "0" and "1" states are defined by voltage levels with respect to ground, and not by switch actuation. Many students assume an actuated (pushed) switch is a "1" input and a de-actuated (unpushed) switch is a "0" input. Not necessarily so! In this circuit, the switches are connecting inputs to *ground*. This means a closed (actuated) switch provides a low (0) input state, while an open (unactuated) switch provides a high (1) input state.

Notes 9

Something omitted from too many basic digital electronics texts is a thorough discussion on interfacing IC logic gates with high-power devices, usually using relays. This is a very important subject, however, because many devices we wish to control with digital logic circuits are too power-hungry to directly drive with the logic gate outputs! Here, students get the opportunity to experiment with how to make a logic gate (CMOS, preferably) drive an electric motor.

One component value you may wish to have your students size themselves is resistor R_3, being the base current limiting resistor for transistor Q_1. It must be sized such that the transistor is saturated with the gate output in the HIGH state, yet not allowing so much base current that the transistor becomes damaged. Figuring out an appropriate size for this resistor is a very practical exercise, forcing students to review transistor theory (calculations with β) as well as consider characteristics of the load.

It may be advisable (especially if the logic gate is TTL and requires a precise 5.0 volt power supply) to have a separate source of power for the electric motor.

Notes 10

It should be noted that the input states in this circuit are defined by the voltage levels, not by the contact status. In other words, a closed contact equals a "low" (0) logic state.

I strongly recommend having students build their logic circuits with CMOS chips rather than TTL, because of the less stringent power supply requirements of CMOS. I also recommend drawing a combinational circuit using four gates, because this is the common number of two-input gates found on 14-pin DIP logic chips.

Notes 11

I strongly recommend having students build their logic circuits with CMOS chips rather than TTL, because of the less stringent power supply requirements of CMOS. I also recommend drawing a combinational circuit using four gates, because this is the common number of two-input gates found on 14-pin DIP logic chips.

Notes 12

I strongly recommend having students build their logic circuits with CMOS chips rather than TTL, because of the less stringent power supply requirements of CMOS. I also recommend drawing a combinational circuit using four gates, because this is the common number of two-input gates found on 14-pin DIP logic chips.

Notes 13

It should be noted that the input states in this circuit are defined by the voltage levels, not by the contact status. In other words, a closed contact equals a "low" (0) logic state.

I strongly recommend having students build their logic circuits with CMOS chips rather than TTL, because of the less stringent power supply requirements of CMOS. I also recommend drawing a combinational circuit using four gates, because this is the common number of two-input gates found on 14-pin DIP logic chips.

Notes 14

It should be noted that the input states in this circuit are defined by the voltage levels, not by the contact status. In other words, a closed contact equals a "low" (0) logic state.

I strongly recommend having students build their logic circuits with CMOS chips rather than TTL, because of the less stringent power supply requirements of CMOS. I also recommend drawing a combinational circuit using four gates, because this is the common number of two-input gates found on 14-pin DIP logic chips.

Notes 15

It should be noted that the input states in this circuit are defined by the voltage levels, not by the contact status. In other words, a closed contact equals a "low" (0) logic state.

I strongly recommend having students build their logic circuits with CMOS chips rather than TTL, because of the less stringent power supply requirements of CMOS. I also recommend drawing a combinational circuit using four gates, because this is the common number of two-input gates found on 14-pin DIP logic chips.

Notes 16

It should be noted that the input states in this circuit are defined by the voltage levels, not by the contact status. In other words, a closed contact equals a "low" (0) logic state.

Here are some suggested Boolean expressions for your students to build gate circuits from:

- Output $= AB + A$
- Output $= \overline{A}B + A$
- Output $= (A + B)A$
- Output $= (A + B)B$
- Output $= \overline{A} + B$
- Output $= A + \overline{B}$
- Output $= \overline{A}B$
- Output $= A\overline{B}$

Notes 17

It should be noted that the input states in this circuit are defined by the voltage levels, not by the contact status. In other words, a closed contact equals a "low" (0) logic state.

Here are some suggested Boolean expressions for your students to build gate circuits from:

- Output $= AB + C$
- Output $= (A + B)C$
- Output $= \overline{A} + BC$
- Output $= \overline{A}B + C$

Notes 18

It should be noted that the input states in this circuit are defined by the voltage levels, not by the contact status. In other words, a closed contact equals a "low" (0) logic state.

Suggested truth tables include the following (encoded as Boolean SOP statements):

- $AB\overline{C} + ABC$
- $\overline{A}B\overline{C} + \overline{A}BC$
- $\overline{A}B\overline{C} + \overline{A}BC + \overline{A}\,\overline{B}\,\overline{C}$
- $A\overline{B}\,\overline{C} + A\,\overline{B}\,C$
- $AB\overline{C} + A\overline{B}\,\overline{C} + \overline{A}\,\overline{B}\,\overline{C}$
- $\overline{A}BC + \overline{A}\,\overline{B}C + \overline{A}\,\overline{B}\,\overline{C}$
- $ABC + \overline{A}BC + AB\overline{C}$
- $A\overline{B}C + \overline{A}\,\overline{B}C + \overline{A}\,\overline{B}\,\overline{C}$
- $ABC + A\overline{B}C + \overline{A}\,\overline{B}C$

I strongly recommend having students build their logic circuits with CMOS chips rather than TTL, because of the less stringent power supply requirements of CMOS. I also recommend drawing a combinational circuit using four gates, because this is the common number of two-input gates found on 14-pin DIP logic chips.

Notes 19

It should be noted that the input states in this circuit are defined by the voltage levels, not by the contact status. In other words, a closed contact equals a "low" (0) logic state.

Here, the arrangement of the input letters D, C, B, and A is purposeful: D represents the most significant bit, while A represents the least significant bit, just like the IC datasheets typically order the input lines.

Normally, I draw LEDs in the schematic to give visual indication of output states. Here, due to the sheer number of required LEDs (16), I decided not to. However, students with access to lots of LEDs may choose to add them to their circuits, because visual indicators do make the circuit's function easier to understand.

If the decoder IC has enable inputs, the students must figure out what to do with them to make the circuit function!

Notes 20

The purpose of this exercise is for students to connect a multiplexer to fulfill an arbitrary logic function specified by the instructor, thus showing the flexibility of the technique.

Here, the arrangement of the input letters C, B, and A is purposeful: C represents the most significant bit, while A represents the least significant bit, just like the IC datasheets typically order the input lines.

Notes 21

Here, I let students choose appropriate values for R_{pullup} and R_{limit}, rather than specify them as given conditions.

Notes 22

Here, I let students choose appropriate values for R_{pullup} and R_{limit}, rather than specify them as given conditions.

Notes 23

Pushbutton switches are recommended over toggle switches for demonstrating a latch circuit. When the contacts are momentary, the latching characteristics of the circuit becomes more evident to see.

I have intentionally requested students try different resistance values for R_{limit} so they may see the effects of gate output *loading*, and the importance of proper logic level voltages. Students should try undersized resistors (10 Ω, perhaps) on both LEDs to generate this problem, and then use oversized resistors (1000 Ω, perhaps) to make the problem go away. Large-valued limiting resistors will cause the LEDs to be dim, but will also restore workable voltage levels so that "high" output states are actually interpreted as "high" when fed back to the gate inputs.

Notes 24

Pushbutton switches are recommended over toggle switches for demonstrating a latch circuit. When the contacts are momentary, the latching characteristics of the circuit becomes more evident to see.

I have intentionally requested students try different resistance values for R_{limit} so they may see the effects of gate output *loading*, and the importance of proper logic level voltages. Students should try undersized resistors (10 Ω, perhaps) on both LEDs to generate this problem, and then use oversized resistors (1000 Ω, perhaps) to make the problem go away. Large-valued limiting resistors will cause the LEDs to be dim, but will also restore workable voltage levels so that "high" output states are actually interpreted as "high" when fed back to the gate inputs.

Notes 25

Pushbutton switches are recommended over toggle switches for demonstrating a latch circuit. When the contacts are momentary, the latching characteristics of the circuit becomes more evident to see.

If students use LEDs to indicate the Q and $\overline{Q}$ output states, they may experience trouble with the circuit not latching as it should. This is an excellent example of gate output *loading*, and the importance of proper logic level voltages. If such problems are encountered, advise the student(s) to use over-sized (too large) LED dropping resistors. This will cause the LEDs to be dim, but restore workable voltage levels so that "high" output states are actually interpreted as "high" when fed back to the gate inputs.

Notes 26

In this activity, students are asked to figure out how to wire the inputs of the J-K flip-flop circuit, and also how to demonstrate the three modes (Set, Reset, and Toggle). Students will have to properly set up their square-wave signal generators to create a workable clock pulse. This not only means a clock pulse at the correct voltage levels, but also one that is slow enough to allow them to clearly see the toggling of the flip-flop.

A great thing to do here is have students use a logic probe to sense the clock pulse and compare that frequency with the blinking of the Q and $\overline{Q}$ LEDs.

Notes 27

One lesson of digital circuits many students learn the hard way is the importance of not leaving CMOS inputs floating. In this case, the lesson is often learned in the form of leaving asynchronous inputs of the J-K flip-flops floating (preset, clear, or both). Be sure to check to see that all chip inputs are accounted for before passing students on this competency. If you see an input floating, touch the chip pin with a pen or pencil and let your students see the effect static has on their circuit!

Notes 28

I strongly recommend having students build their logic circuits with CMOS chips rather than TTL, because of the less stringent power supply requirements of CMOS.

Notes 29

Students are to research the datasheet for their particular IC and figure out from that what connections and timing sequences they need to make the circuit perform as requested. It is very important for students to learn to interpret manufacturers' datasheets!

I recommend a slow clock frequency (1 Hz or so) to allow for easy viewing of the count sequence.

Notes 30

Students are left on their own to figure out what they must do with the other inputs (lamp test, BI, RBI, etc.) to make the decoder/driver chip function properly.

Notes 31

I have purposely left the details of the schematic diagram vague, so that students must do a lot of datasheet research on their own to figure out how to make an event counter circuit. You may choose to give your students part numbers for the integrated circuits, or choose not to, depending on how capable your students are. The point is, they must figure out how to make the ICs work based on what they read from the manufacturer.

Something else students will probably have to do is de-bounce the event switch. Some event switches are inherently bounceless, while others are definitely not. Switch debouncing is something your students need to learn about and integrate into this circuit.

An extension of this exercise is to incorporate troubleshooting questions. Whether using this exercise as a performance assessment or simply as a concept-building lab, you might want to follow up your students' results by asking them to predict the consequences of certain circuit faults.

Notes 32

Students are to research the datasheet for their particular IC and figure out from that what connections and timing sequences they need to make the circuit perform as requested. It is very important for students to learn to interpret manufacturers' datasheets!

I recommend a slow clock frequency (1 Hz or so) to allow for easy viewing of the shift patterns. To conserve the number of necessary input switches, I allow students to hard-wire the data inputs (D_0 through D_3). This means they only need switches to control the mode of the shift register (parallel load, shift right, shift left, and shift inhibit).

Notes 33

Students are to research the datasheet for their particular IC and figure out from that what connections and timing sequences they need to make the circuit perform as requested. It is very important for students to learn to interpret manufacturers' datasheets!

I recommend a slow clock frequency (1 Hz or so) to allow for easy viewing of the shift patterns. To conserve the number of necessary input switches, I allow students to hard-wire the data inputs (D_0 through D_3). This means they only need switches to control the mode of the shift register (parallel load, shift right, shift left, and shift inhibit).

Notes 34

Use a four-pole, unipolar stepper motor for this assessment, with a power supply capable of sourcing the required current.

An extension of this exercise is to incorporate troubleshooting questions. Whether using this exercise as a performance assessment or simply as a concept-building lab, you might want to follow up your students' results by asking them to predict the consequences of certain circuit faults.

Notes 35

An extension of this exercise is to incorporate troubleshooting questions. Whether using this exercise as a performance assessment or simply as a concept-building lab, you might want to follow up your students' results by asking them to predict the consequences of certain circuit faults.

Notes 36

An extension of this exercise is to incorporate troubleshooting questions. Whether using this exercise as a performance assessment or simply as a concept-building lab, you might want to follow up your students' results by asking them to predict the consequences of certain circuit faults.

Notes 37

An extension of this exercise is to incorporate troubleshooting questions. Whether using this exercise as a performance assessment or simply as a concept-building lab, you might want to follow up your students' results by asking them to predict the consequences of certain circuit faults.

The two diodes in this circuit are a matter of necessity: getting the circuit to work with only two sets of switch contacts per relay. Ideally, each relay would be 3PDT with separate contact sets for latching, interlocking, and motor power. To use a DPDT relay requires that one of these contact sets do double-duty. In this case, one of the contact sets on each relay handling power to the motor must also handle the job of seal-in (latching). Without the diodes in place, both relays chatter when either motion button is pressed. This is because both relay coils receive power: one coil directly through the switch; the other through the same switch, back through the motor, and then through the seal-in (latching) connection. The diodes prevent this "feed-through" to the other relay coil from happening, without interfering with the normal latching function.

Notes 38

In this activity, students are asked to figure out the details of configuring the ADC: what power supply voltage to use, what resistor values, etc. The best source for this information is the ADC's datasheet!

For those students who have trouble figuring out how to calculate resolution, I recommend the following formula:

$$V_{resolution} = \frac{V_{range}}{2^n - 1}$$

Where,

V_{range} = "Span" of analog voltage input (how many volts of range it has from 00000000 to 11111111)

n = Number of output bits for the ADC

Notes 39

Here, I let students choose appropriate values for $R_{pulldown}$ and R_{limit}, rather than specify them as given conditions.

Notes 40

Here, I let students choose appropriate values for $R_{pulldown}$ and R_{limit}, rather than specify them as given conditions.

Notes 41

Here, I let students design and build their own transistor drive circuit to interpose between the MCU and the DC motor.

Notes 42

One method that is convenient for generating an analog output voltage with many microcontrollers is to program the MCU to generate a PWM output, then build an analog filter circuit to capture just the average DC value of that PWM waveform.

Notes 43

Any relevant notes for the assessment activity go here.