



澳門城市大學
Universidade da Cidade de Macau
City University of Macau

設計計算工作室I



主講人 |

姓名 張琪

Name Zhang Qi

澳門城市大學

City University of Macau



考核要求

- 課後書面作業（30%）
- 出勤（10%）
- 書面報告（60%）
 - 書面報告應不超過十頁A4紙，參考給定的實驗報告格式和模板。
 - 單次的實驗書面報告不超過十頁，如兩次實驗合并的書面報告不應超過二十頁。





參考教材

- 周舸. 計算機導論 (第2版) . 人民郵電出版社 , 2023
- 黃仙山. 大學物理 (上冊) . 人民郵電出版社 , 2020
- 歐陽星明. 數字電路邏輯設計 (第3版) (微課版) , 2021
- 大學物理實驗虛擬仿真在綫資料
- 中國大學MOOC在綫資料
- CircuitJS1在綫電子電路模擬器





實驗目的

本實驗是瞭解Circuitjs使用方法和邏輯的輸入輸出

1. 使用 Circuitjs 創建一個包含LED燈的簡單電路
2. circuitjs 使用邏輯輸入簡化電源與開關
3. circuitjs 七段 LED 數碼管使用
4. circuitjs 使用邏輯輸入輸出在七段 LED 數碼管上顯示 1 或 0



實驗原理

- Circuitjs 在綫電子電路模擬器概述
- Circuitjs 電子電路模擬器是一款 web 在綫電路模擬器，可以在瀏覽器上仿真地模擬各種 模擬 或 數字 的電路，用戶無需安裝各種軟件，生成的電路也可以保存下來，以及便捷地通過在綫方式分享給其它用戶
- This program is free software;
- <https://www.falstad.com/circuit/circuitjs.html>

實驗原理

- 1 使用 Circuitjs 創建一個包含LED燈的簡單電路
- 創建新的空白電路
- 默認給出的示例電路只是起一個演示作用，如果你要創建自己的電路，可以點擊 菜單--電路--空白的電路 從而打開一個新的空白電路



實驗原理

- 1 使用 Circuitjs 創建一個包含LED燈的簡單電路
- 添加組件
- 清空默認的示例電路後, 可以通過 菜單--繪製 菜單添加自己想要的組件
- 可以參考視頻 :
<https://www.xiaogd.net/book/spcp/other/video/circuitjs-create-a-simple-circuit-video.html>

實驗原理

- 1 使用 Circuitjs 創建一個包含LED燈的簡單電路
- 添加組件
- 下面以添加一個繼電器為例，通過點擊 菜單--繪製--無源元件--添加繼電器 後便處於添加繼電器狀態，此時的鼠標光標變為十字光標 "+" 形狀：



實驗原理

- 1 使用 Circuitjs 創建一個包含LED燈的簡單電路
- 連續添加組件
- 選擇添加組件後，系統默認即是連續添加，因此添加一個組件後，繼續處在添加該組件狀態，用戶可以繼續拖動以添加更多的類似組件。
- 退出添加狀態
- 如不再需要繼續添加組件，可以通過按 Esc 鍵退出添加狀態。

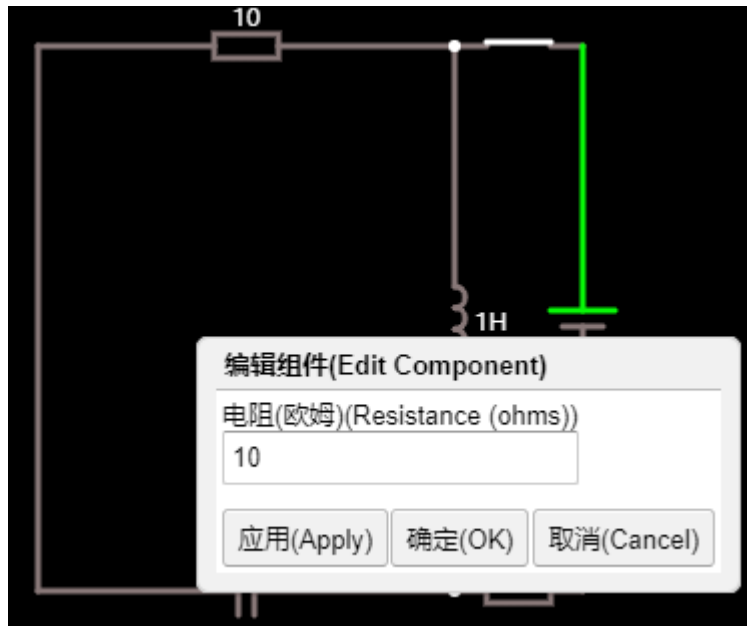
實驗原理

- 1 使用 Circuitjs 創建一個包含LED燈的簡單電路
- 編輯屬性
- 在組件上單擊鼠標右鍵，在彈出的快捷菜單上選擇 編輯：



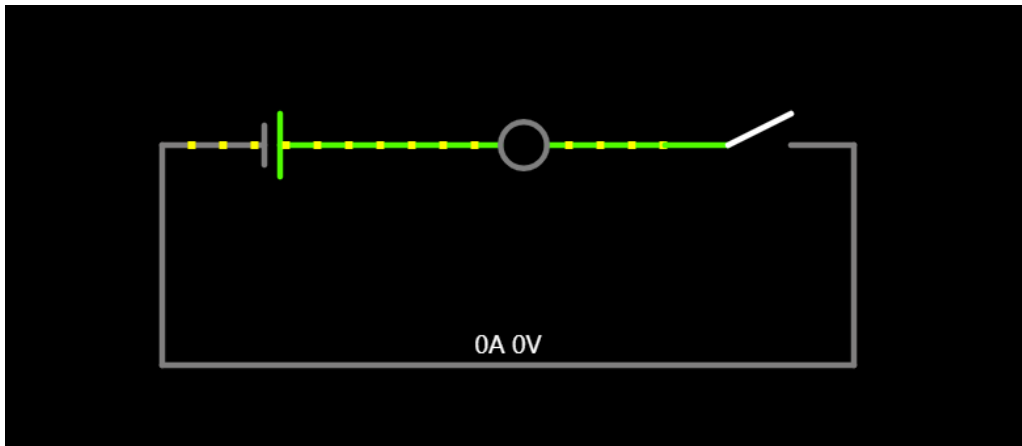
實驗原理

- 1 使用 Circuitjs 創建一個包含LED燈的簡單電路
- 編輯屬性
- 在彈出的對應組件的屬性窗口上，顯示了那些可以編輯的屬性值，調整相關值，然後點擊 確定 即可改變相應屬性值：



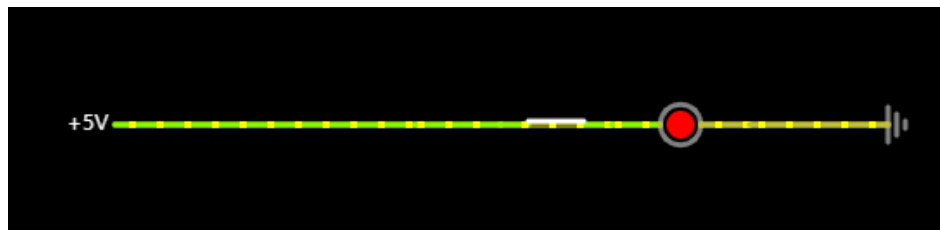
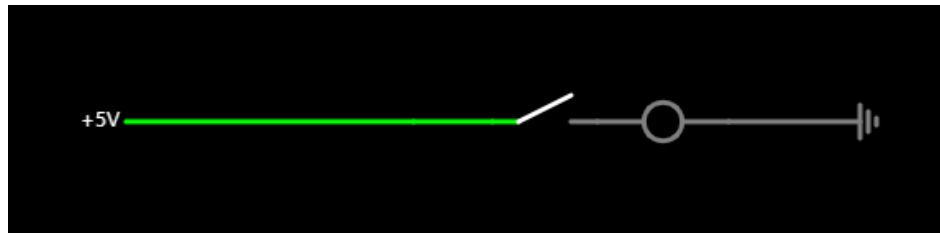
實驗原理

- 1 使用 Circuitjs 創建一個包含LED燈的簡單電路
- 包含LED燈的簡單電路例子如下所示



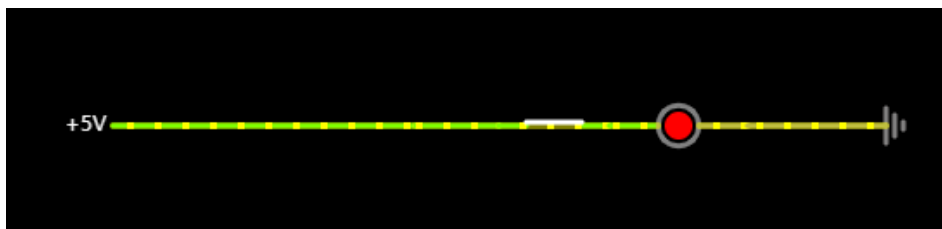
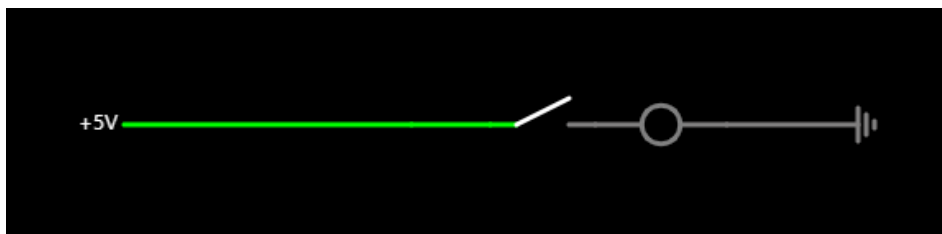
實驗原理

- 1 使用 Circuitjs 創建一個包含LED燈的簡單電路
- 包含LED燈的簡單電路例子如下所示



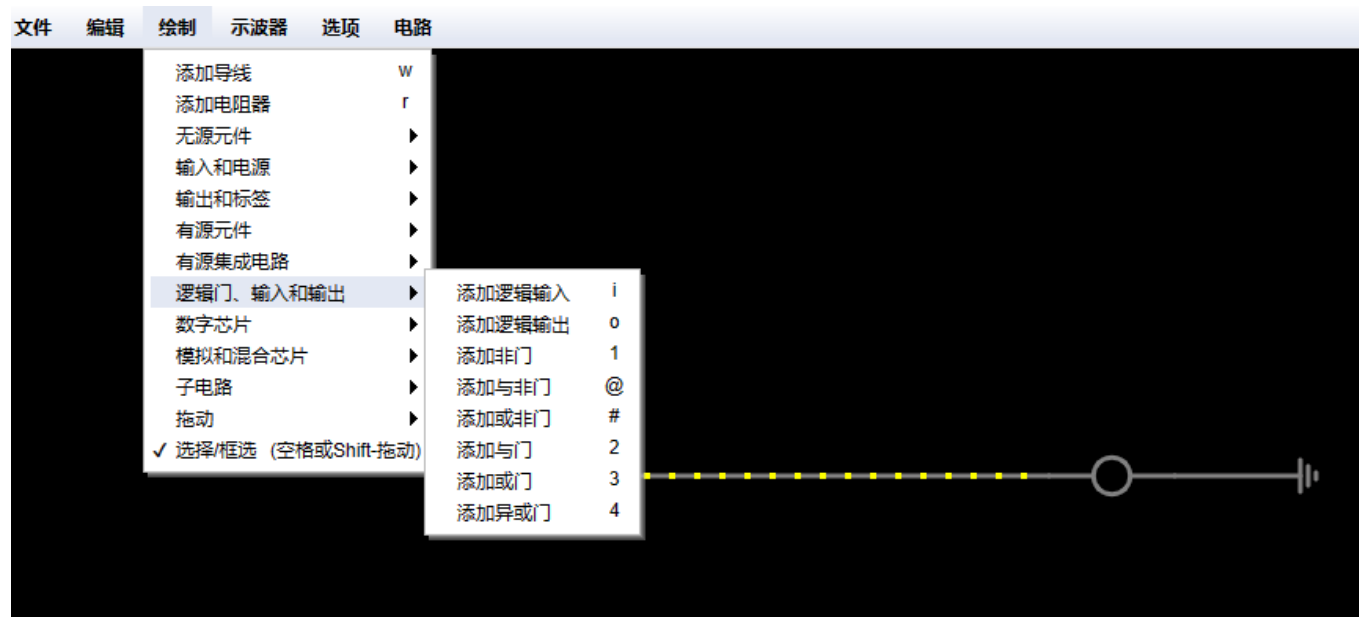
實驗原理

- 2 circuitjs 使用邏輯輸入簡化電源與開關
- 電源和開關合二為一，利用邏輯輸入進行簡化
- <https://www.xiaogd.net/book/spcp/other/video/circuitjs-simplify-source-and-switch-by-logic-input-video.html>
- 原始的輸入電源和開關



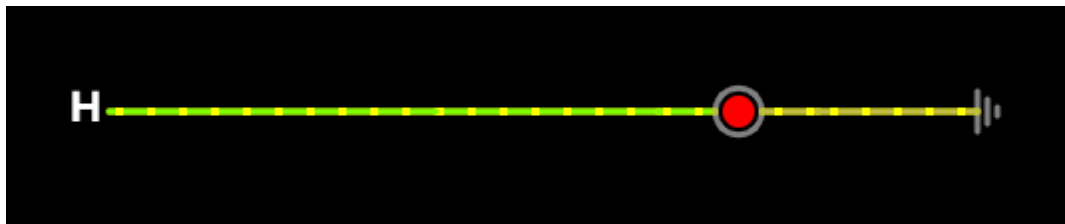
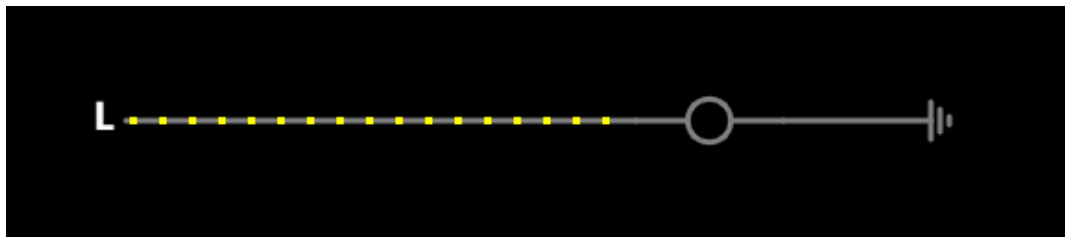
實驗原理

- 2 circuitjs 使用邏輯輸入簡化電源與開關
- 電源和開關合二為一，利用邏輯輸入進行簡化
- <https://www.xiaogd.net/book/spcp/other/video/circuitjs-simplify-source-and-switch-by-logic-input-video.html>
- 刪除原始的輸入電源和開關
- 添加邏輯輸入如圖所示



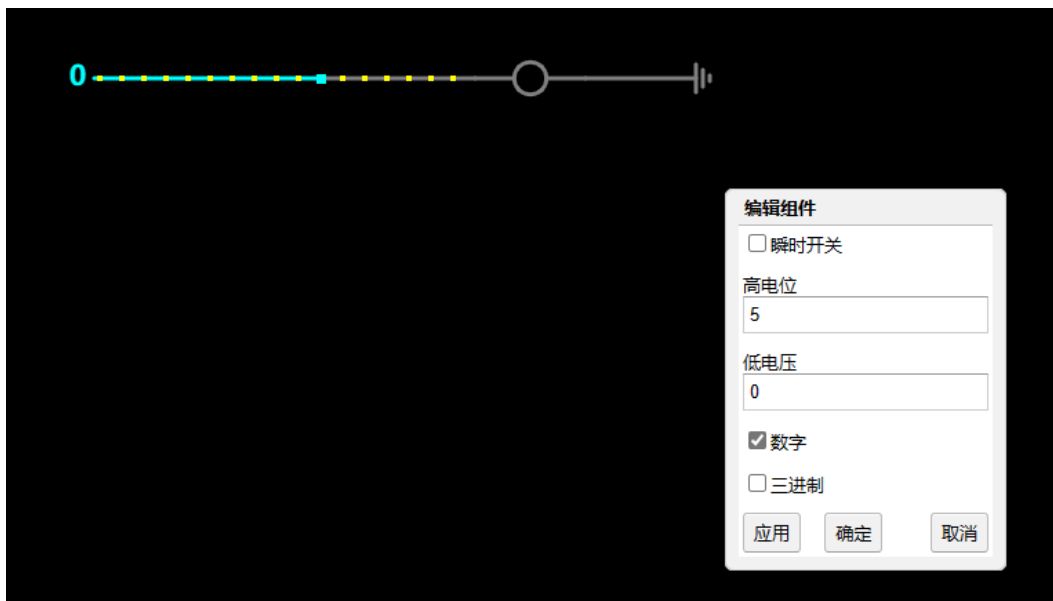
實驗原理

- 2 circuitjs 使用邏輯輸入簡化電源與開關
- 電源和開關合二為一，利用邏輯輸入進行簡化
- <https://www.xiaogd.net/book/spcp/other/video/circuitjs-simplify-source-and-switch-by-logic-input-video.html>
- 刪除原始的輸入電源和開關
- 構成簡化的電路如圖所示



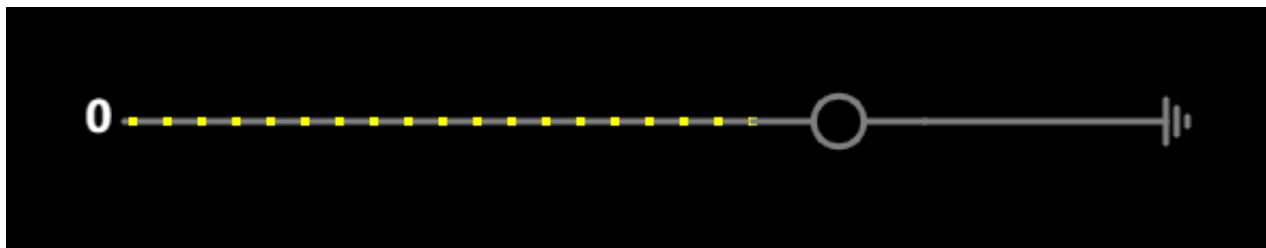
實驗原理

- 2 circuitjs 使用邏輯輸入簡化電源與開關
- 電源和開關合二為一，利用邏輯輸入進行簡化
- <https://www.xiaogd.net/book/spcp/other/video/circuitjs-simplify-source-and-switch-by-logic-input-video.html>
- 刪除原始的輸入電源和開關
- 也可以編輯邏輯輸入以數字的方式進行顯示



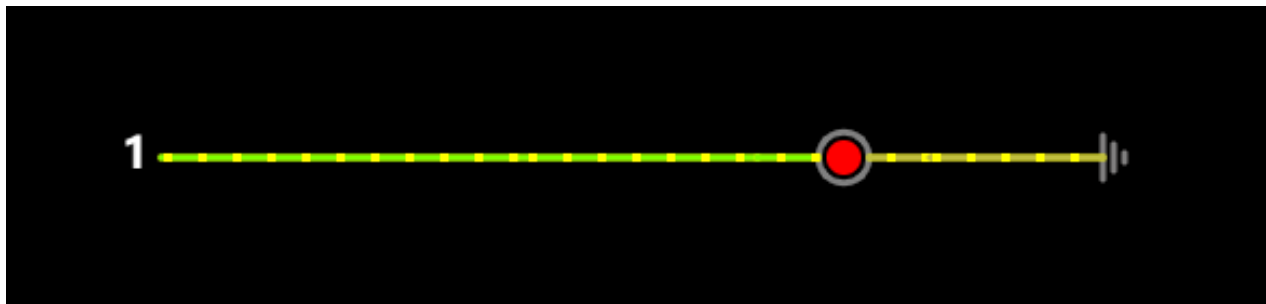
實驗原理

- 2 circuitjs 使用邏輯輸入簡化電源與開關
- 電源和開關合二為一，利用邏輯輸入進行簡化
- <https://www.xiaogd.net/book/spcp/other/video/circuitjs-simplify-source-and-switch-by-logic-input-video.html>
- 刪除原始的輸入電源和開關
- 也可以編輯邏輯輸入以數字的方式進行顯示



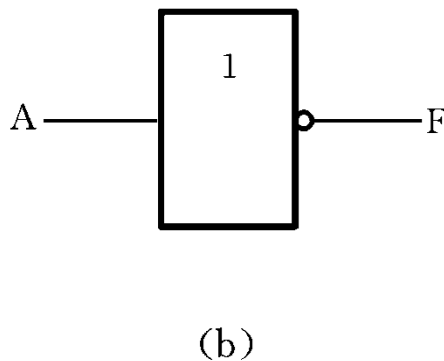
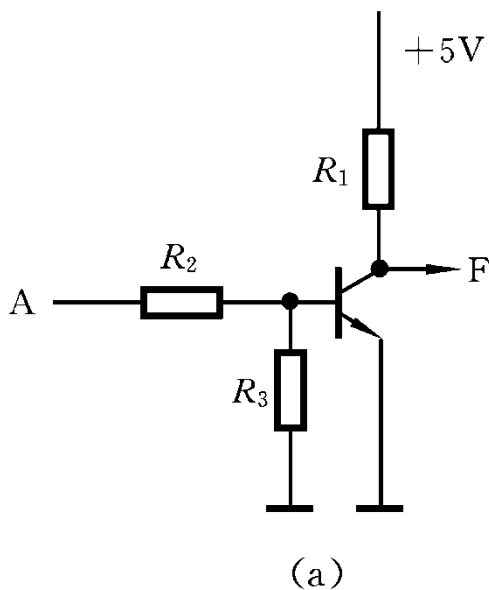
實驗原理

- 2 circuitjs 使用邏輯輸入簡化電源與開關
- 電源和開關合二為一，利用邏輯輸入進行簡化
- <https://www.xiaogd.net/book/spcp/other/video/circuitjs-simplify-source-and-switch-by-logic-input-video.html>
- 刪除原始的輸入電源和開關
- 也可以編輯邏輯輸入以數字的方式進行顯示



12.3 邏輯門電路

- 簡單邏輯門電路
- 一. 非門
- 非門又稱“反相器”。晶體三極管反相器的電路圖和邏輯符號如圖 (a)和圖(b)所示

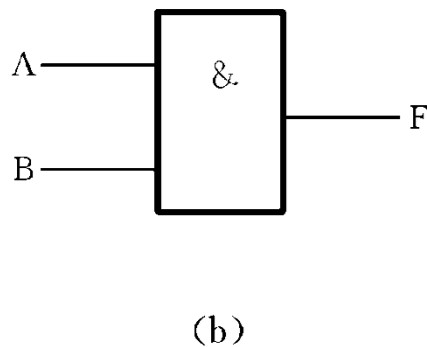
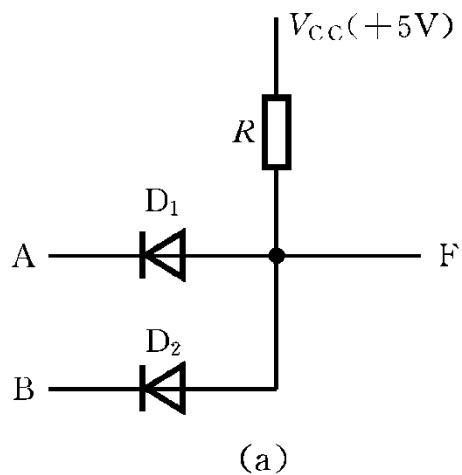


A/V	F/V
0	+5
+5	0

A	F
0	1
1	0

12.3 邏輯門電路

- 簡單邏輯門電路
- 二． 與門
- 一個由二極管構成的2輸入與門電路如下圖所示

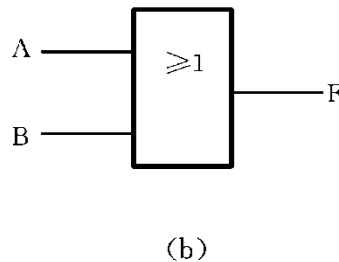
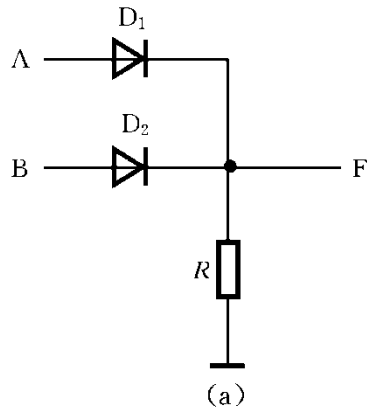


A/V	B/V	F/V
0	0	0
0	+5	0
+5	0	0
+5	+5	+5

A	B	F
0	0	0
0	1	0
1	0	0
1	1	1

12.3 邏輯門電路

- 簡單邏輯門電路
- 三． 或門
- 一個由二極管構成的2輸入或門電路如下圖所示

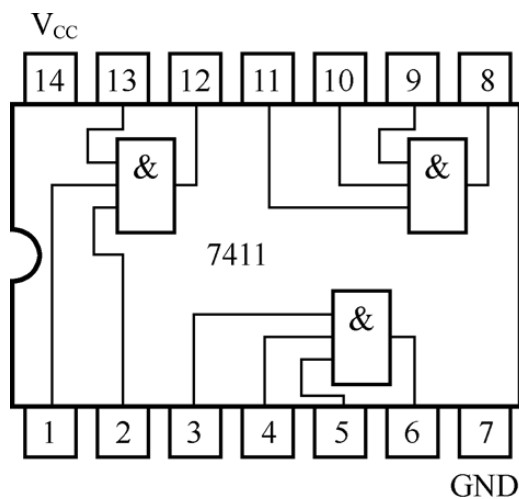


A/V	B/V	F/V
0	0	0
0	+5	+5
+5	0	+5
+5	+5	+5

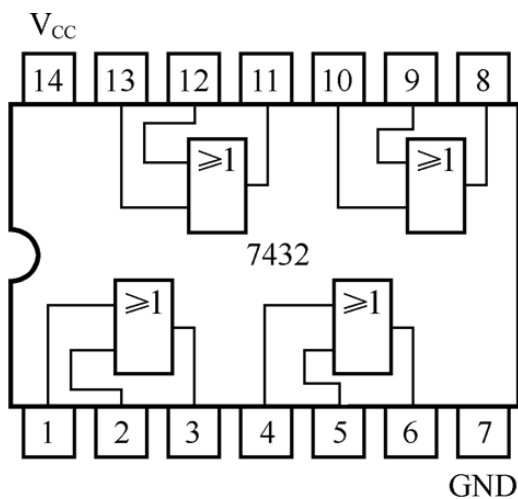
A	B	F
0	0	0
0	1	1
1	0	1
1	1	1

12.3 邏輯門電路

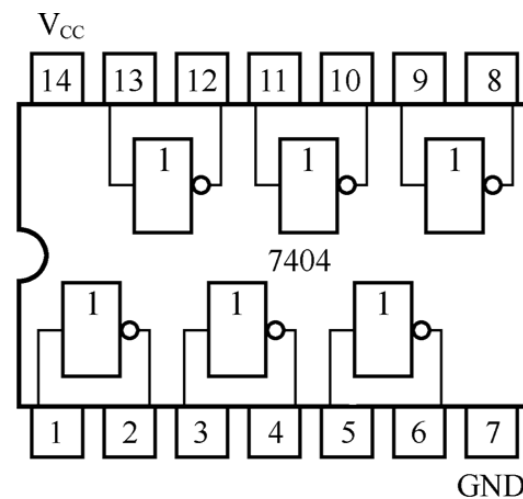
- 常用TTL集成邏輯門
- 1. 基本邏輯門
- 基本邏輯門是指實現3種基本邏輯運算的與門、或門和非門。常用的TTL與門集成電路芯片有四2輸入與門7408，三3輸入與門7411等



(a)



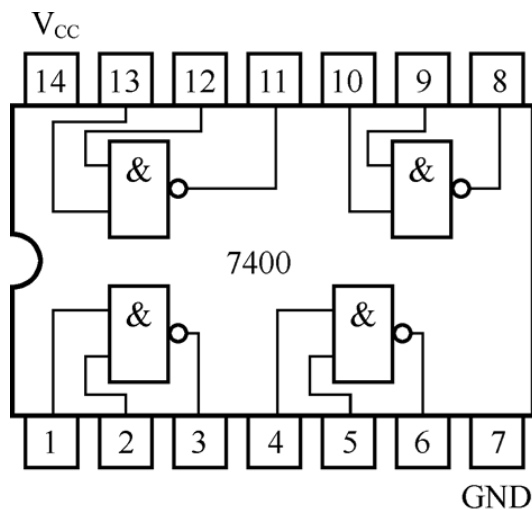
(b)



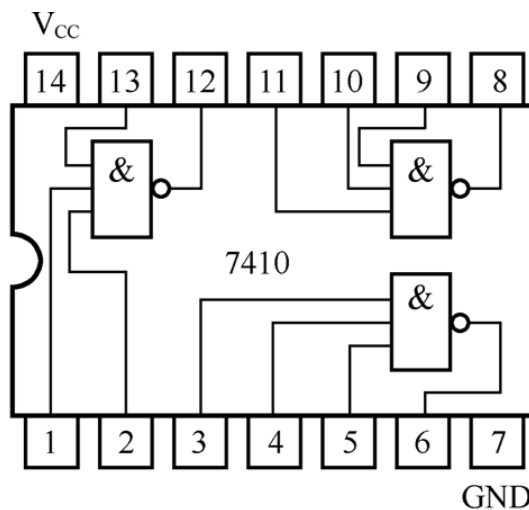
(c)

12.3 邏輯門電路

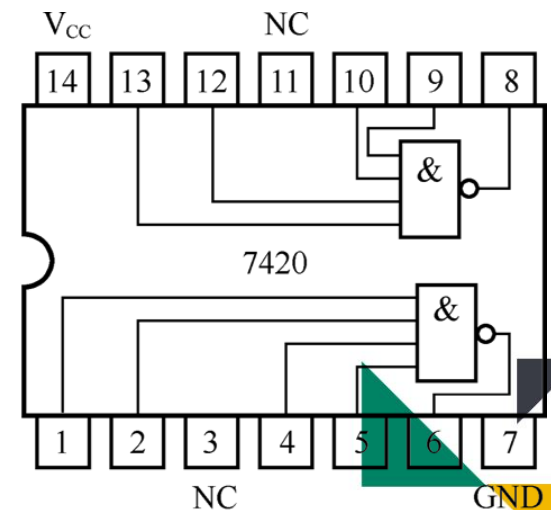
- 常用TTL集成邏輯門
- 2、複合邏輯門
- 複合邏輯門是指實現複合邏輯運算的與非門、或非門、與或非門、異或門等
- 與非門
- 常用的TTL與非門集成電路芯片有四2輸入與非門7400，三3輸入與非門7410，二4輸入與非門7420等



(a)



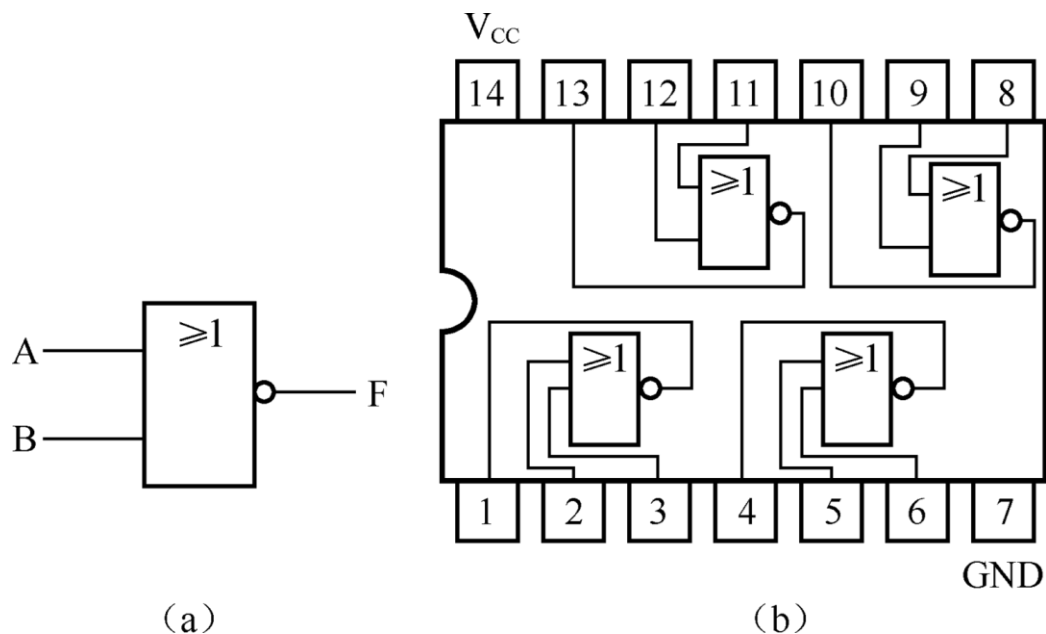
(b)



(c)

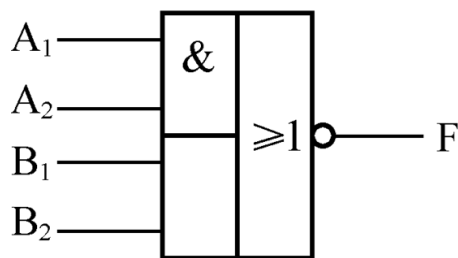
12.3 邏輯門電路

- 常用TTL集成邏輯門
- 2、複合邏輯門
- 或非門
- 常用的TTL或非門集成電路芯片有四2輸入或非門7402，三3輸入或非門7427等。例如：7402

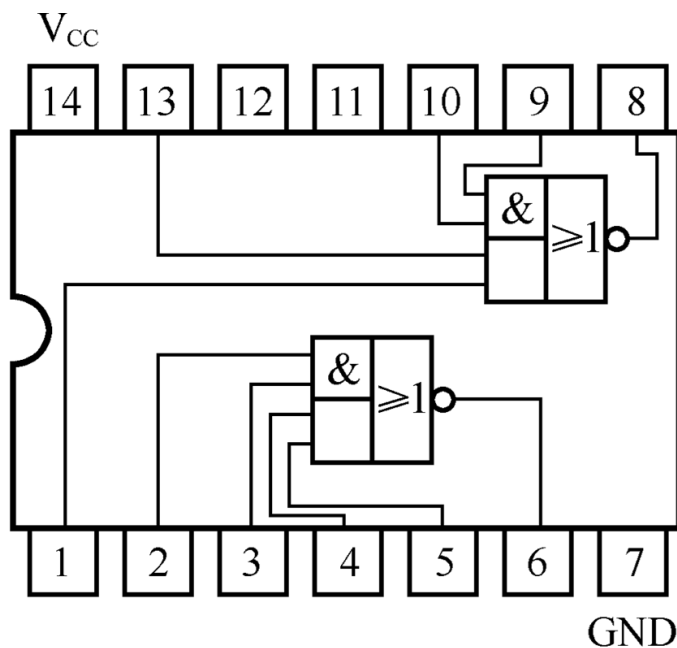


12.3 邏輯門電路

- 常用TTL集成邏輯門
- 2、複合邏輯門
- 與或非門
- 常用的TTL與或非門集成電路芯片有雙2-2與或非門7451、3-2-2-3與或非門7454等。例如：7451



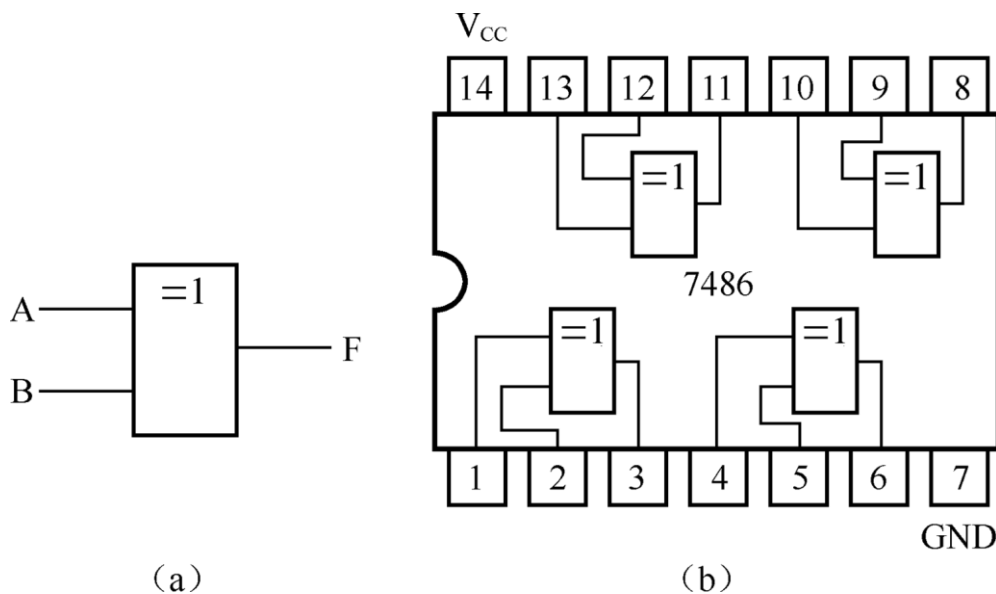
(a)



(b)

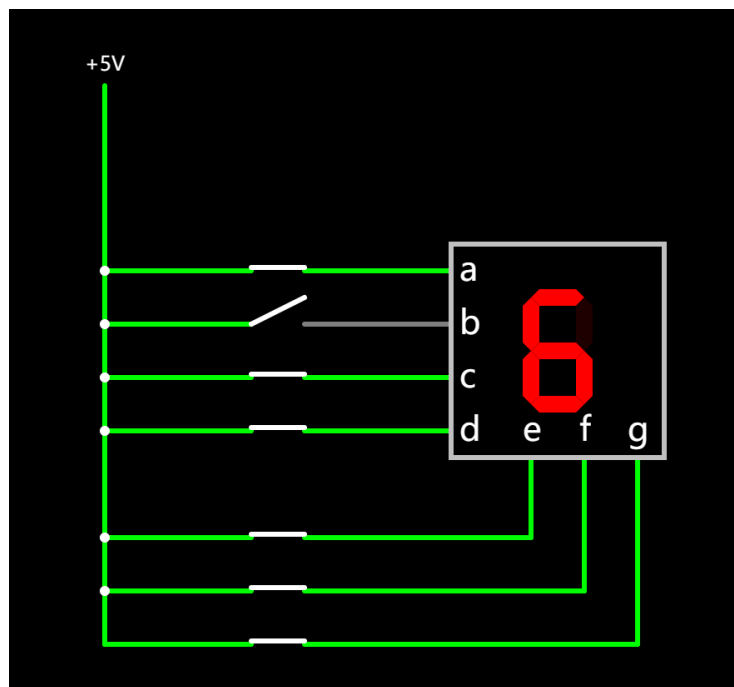
12.3 邏輯門電路

- 常用TTL集成邏輯門
- 2、複合邏輯門
- 異或門
- 異或門只有兩個輸入端，常用的TTL異或門集成電路芯片有7486等。下圖所示為異或門的邏輯符號和7486的引腳排列圖



實驗原理

- 3 circuitjs 七段 LED 數碼管使用
- 可以參考如下資料：
<https://www.xiaogd.net/book/spcp/other/video/circuitjs-seven-segment-led-video.html>
- 七段 LED 數碼管使用電路構成如圖所示



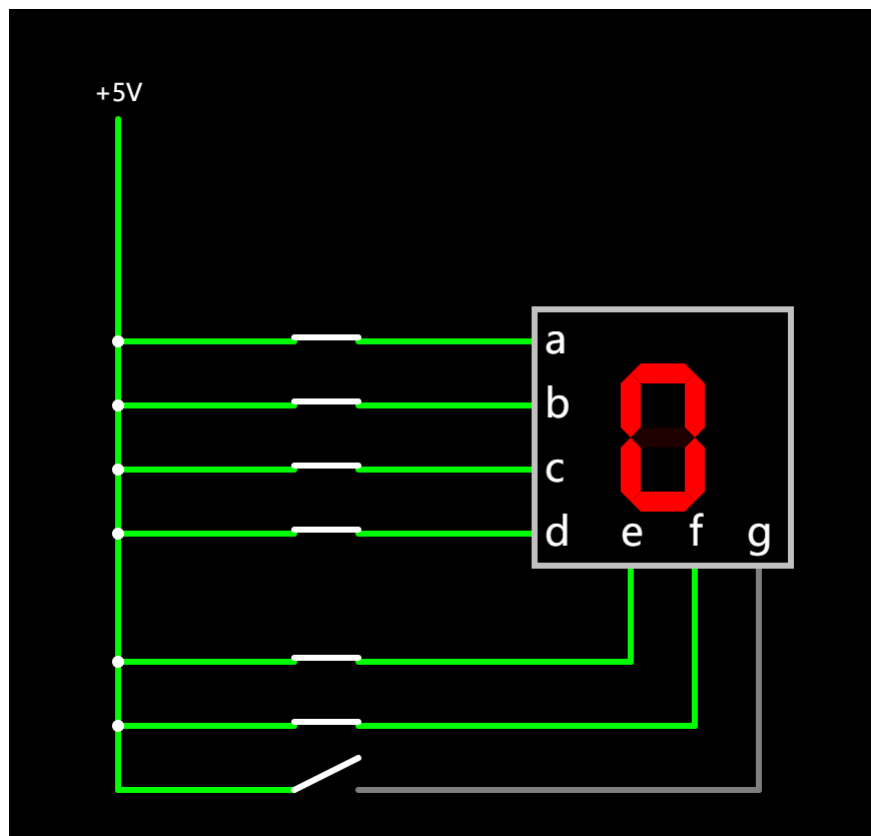
實驗原理

- 3 circuitjs 七段 LED 數碼管使用
- 需要利用導綫，電源，開關，和七段 LED 顯示器繪製



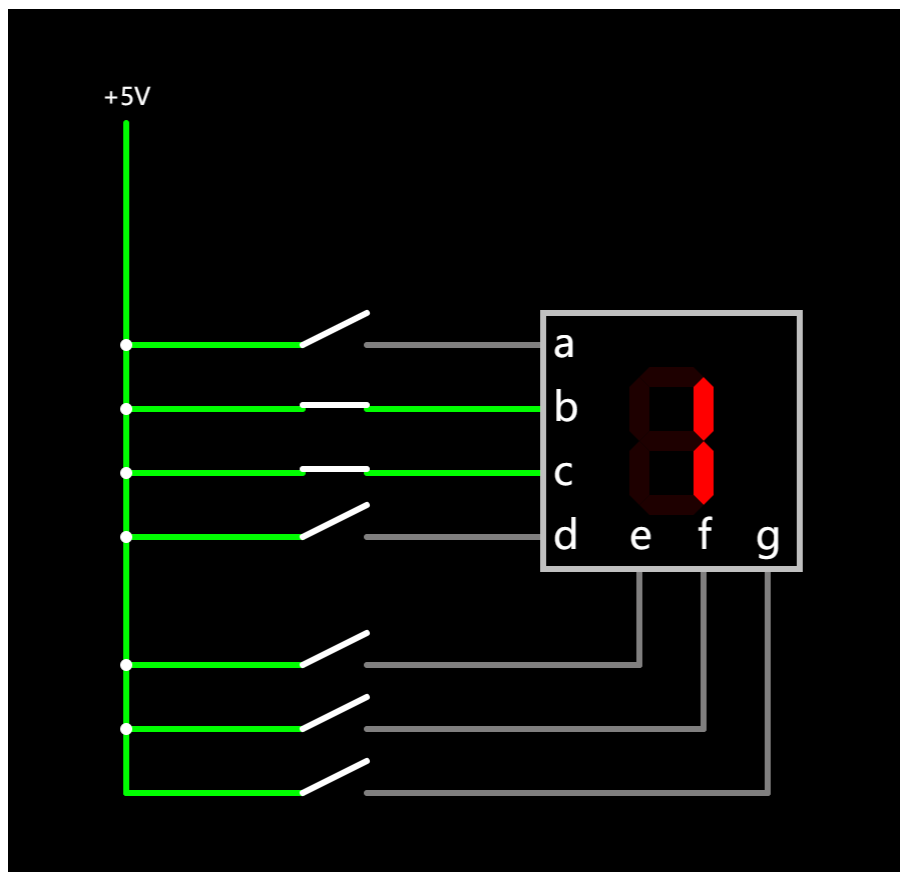
實驗原理

- 4 circuitjs 使用邏輯輸入輸出在七段 LED 數碼管上顯示 1 或 0
- 正常的開關顯示 0



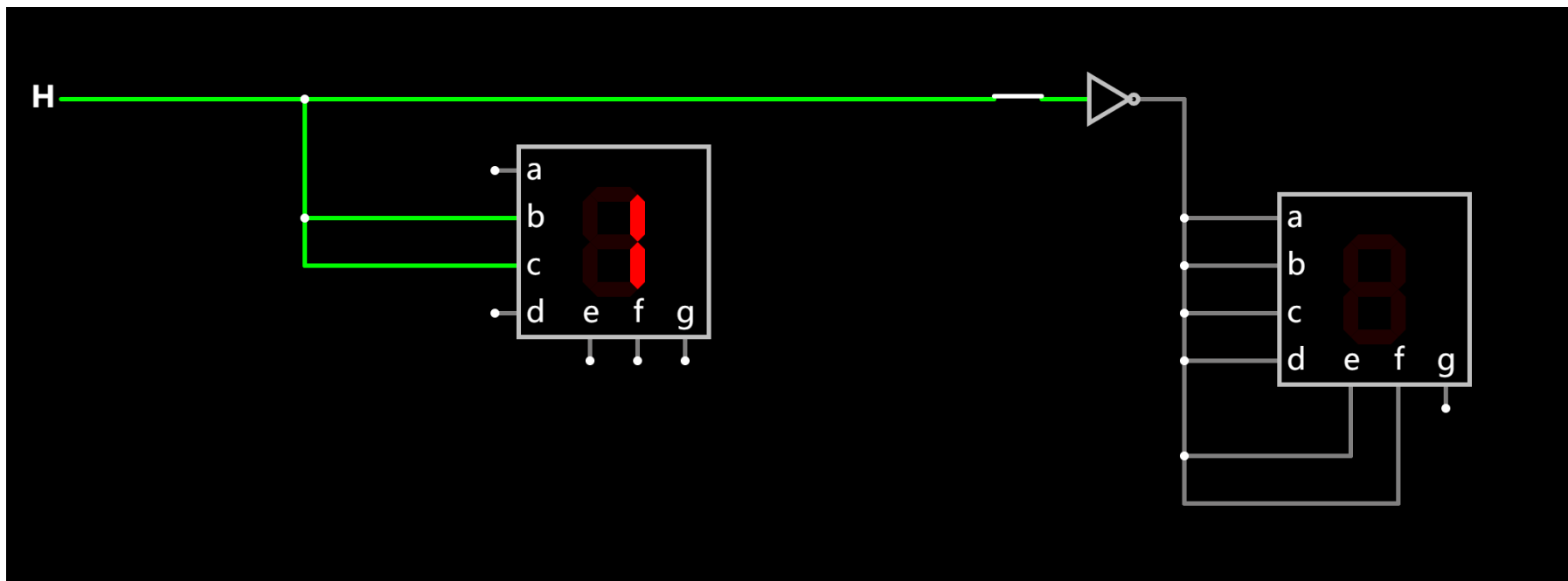
實驗原理

- 4 circuitjs 使用邏輯輸入輸出在七段 LED 數碼管上顯示 1 或 0
- 正常的開關顯示 1



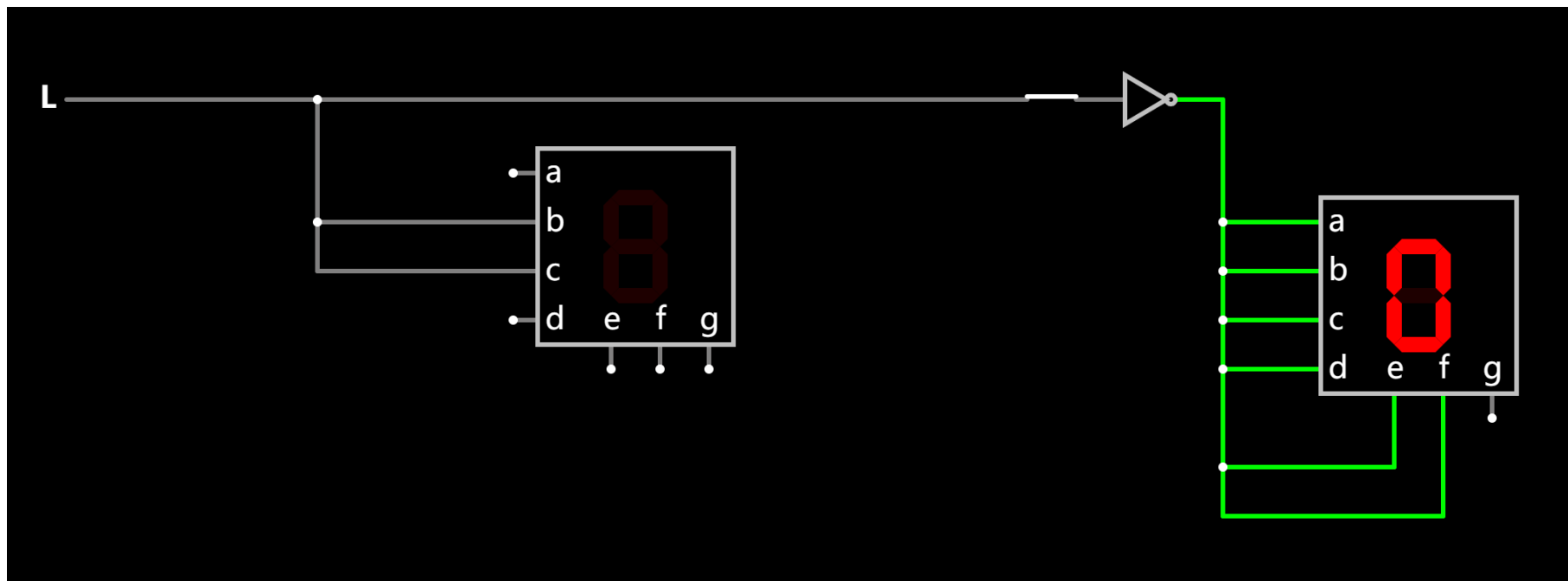
實驗原理

- 4 circuitjs 使用邏輯輸入輸出在七段 LED 數碼管上顯示 1 或 0
- 邏輯輸入輸出顯示 1



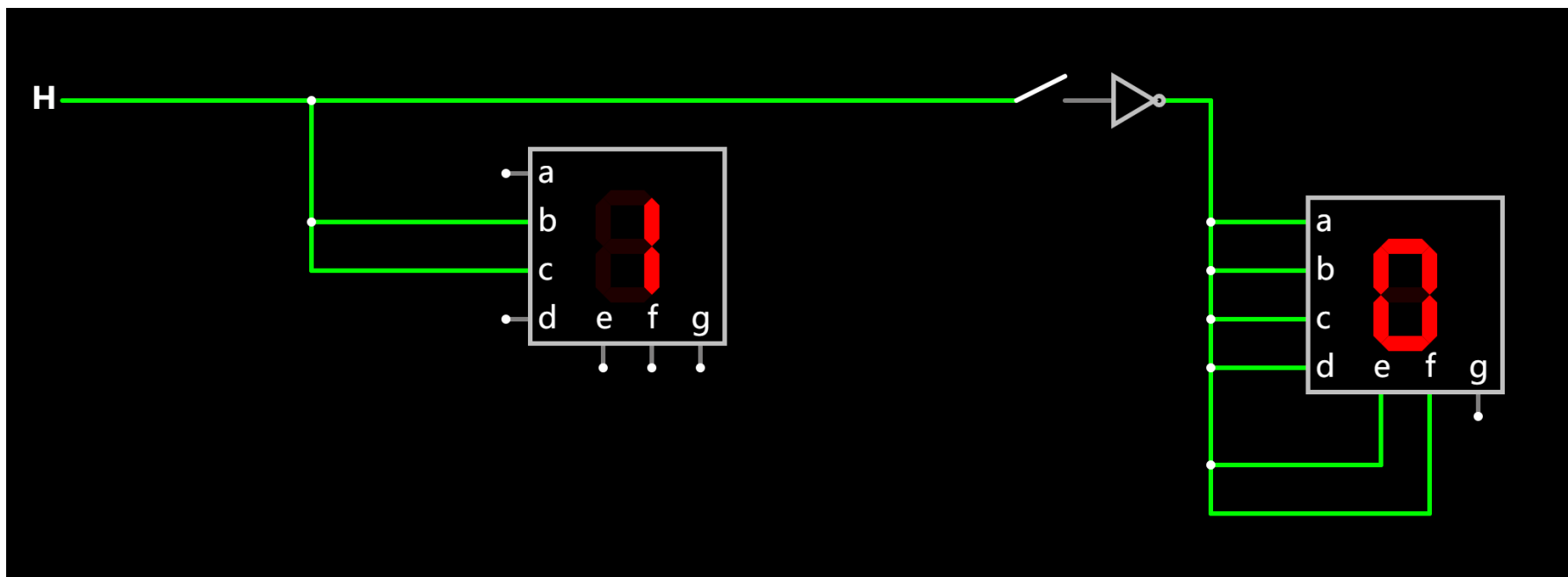
實驗原理

- 4 circuitjs 使用邏輯輸入輸出在七段 LED 數碼管上顯示 1 或 0
- 邏輯輸入輸出顯示 0



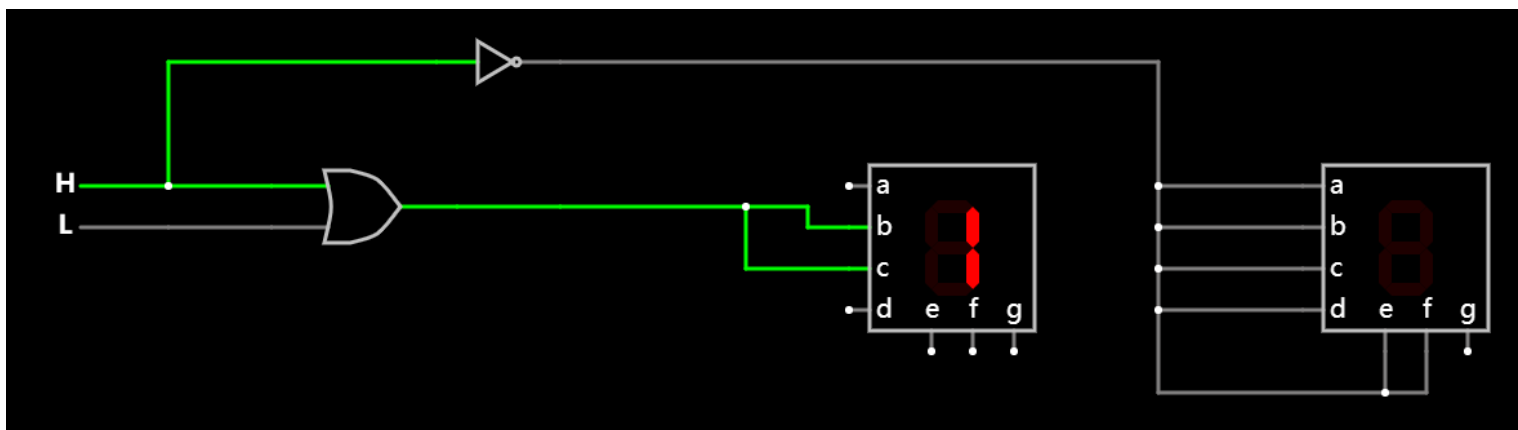
實驗原理

- 4 circuitjs 使用邏輯輸入輸出在七段 LED 數碼管上顯示 1 或 0
- 也可同時顯示 0 和 1



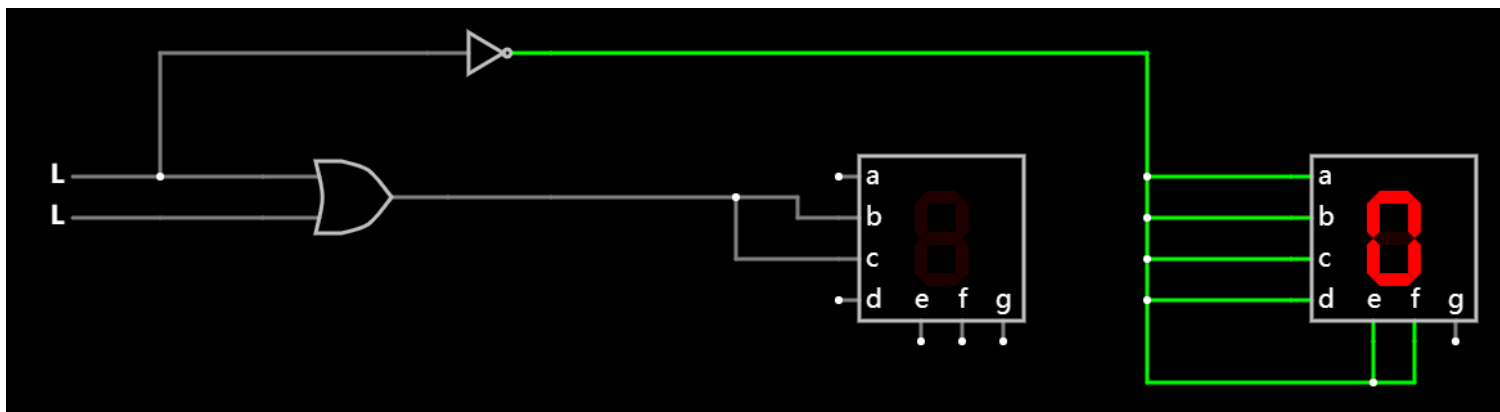
實驗原理

- 4 circuitjs 使用邏輯輸入輸出在七段 LED 數碼管上顯示 1 或 0
- 也可利用邏輯或門，非門控制輸入輸出結果進行顯示
- 如圖所示



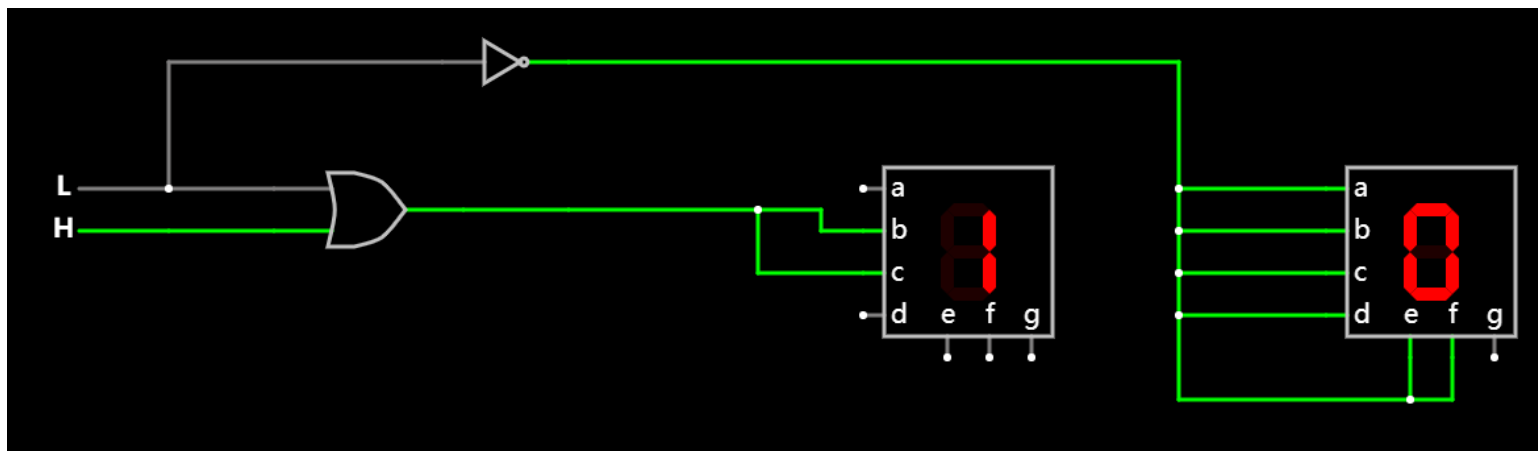
實驗原理

- 4 circuitjs 使用邏輯輸入輸出在七段 LED 數碼管上顯示 1 或 0
- 也可利用邏輯或門，非門控制輸入輸出結果進行顯示
- 如圖所示



實驗原理

- 4 circuitjs 使用邏輯輸入輸出在七段 LED 數碼管上顯示 1 或 0
- 也可利用邏輯或門，非門控制輸入輸出結果進行顯示
- 如圖所示





實驗內容

1. 使用 Circuitjs 創建一個包含LED燈的簡單電路
 - 請參考實驗原理，創建自己的一個包含LED燈的簡單電路
(最好和實驗原理裏的不一樣)
 - 截圖自己創建的電路，并簡要說明





實驗內容

2. circuitjs 使用邏輯輸入簡化電源與開關

- 請參考實驗原理，根據自己已經創建的電路，進行進一步化簡，使用邏輯輸入簡化電源與開關
- 截圖自己化簡的電路，并給出簡要說明





實驗內容

3. circuitjs 七段 LED 數碼管使用

- 請參考實驗原理，創建自己的七段 LED 數碼管
- 請使用自己的七段 LED 數碼管，顯示出數字 2，3，4，5，6，7
- 截圖點亮數字 2，3，4，5，6，7 并給出簡要說明





實驗內容

4. circuitjs 使用邏輯輸入輸出在七段 LED 數碼管上顯示數字
- 請參考實驗原理，在自己的創建的七段 LED 數碼管上，使用邏輯輸入輸出（建議使用邏輯門電路，或門，非門等）顯示出0和8
 - 請參考實驗原理，在自己的創建的七段 LED 數碼管上，使用邏輯輸入輸出（建議使用邏輯門電路，或門，非門等）顯示出0和3
 - 截圖上述顯示的電路圖，并給出簡要說明





思考題

思考題請回答如下問題

1. 請思考一下，七段 LED 數碼管在實際的生產工作生活中有什麼作用，并給出具體的說明。
2. 請思考一下，如果想顯示26個字母，應該選用什麼樣的顯示設備，請舉例說明。





注意事項

- #請完成思考題
- #請儘快完成實驗13-1,并在TronClass系統內按時提交作業。
- #請儘快完成實驗14-1,并在TronClass系統內按時提交作業。
- #請儘快完成實驗14-1,并在TronClass系統內按時提交作業。
- #請儘快完成實驗14-1,并在TronClass系統內按時提交作業。





感謝觀賞

Thank you for listening.