

Lab 8: Speaker

106061218 李丞恩

1. Audio-data parallel-to-serial module

Design Specification

(1) Input:

Clk: 100MHz global clock ◦

rst: 開關，接在一個 dip switch 上 ◦

(2) Output:

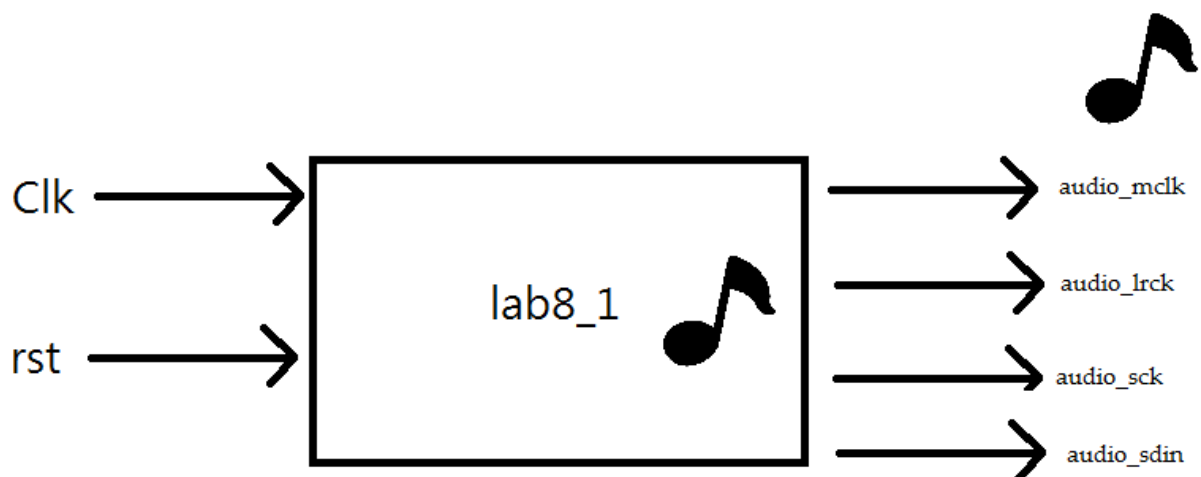
audio_mclk: Master clock ◦

audio_lrck: Left_right clock ◦

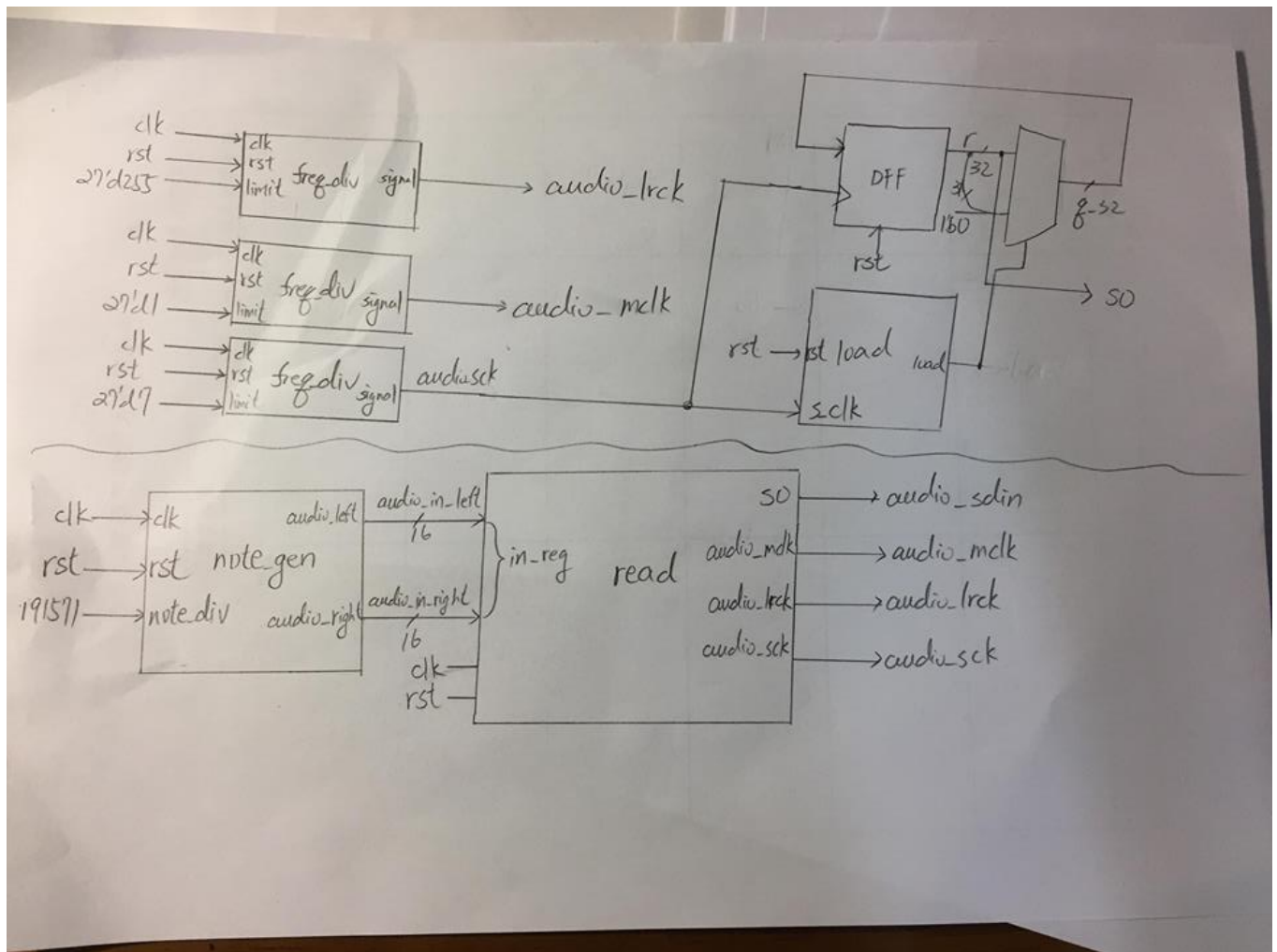
audio_sck: serial clock ◦

audio_sdin: serial data input ◦

(3) Block diagram:



Design Implementation



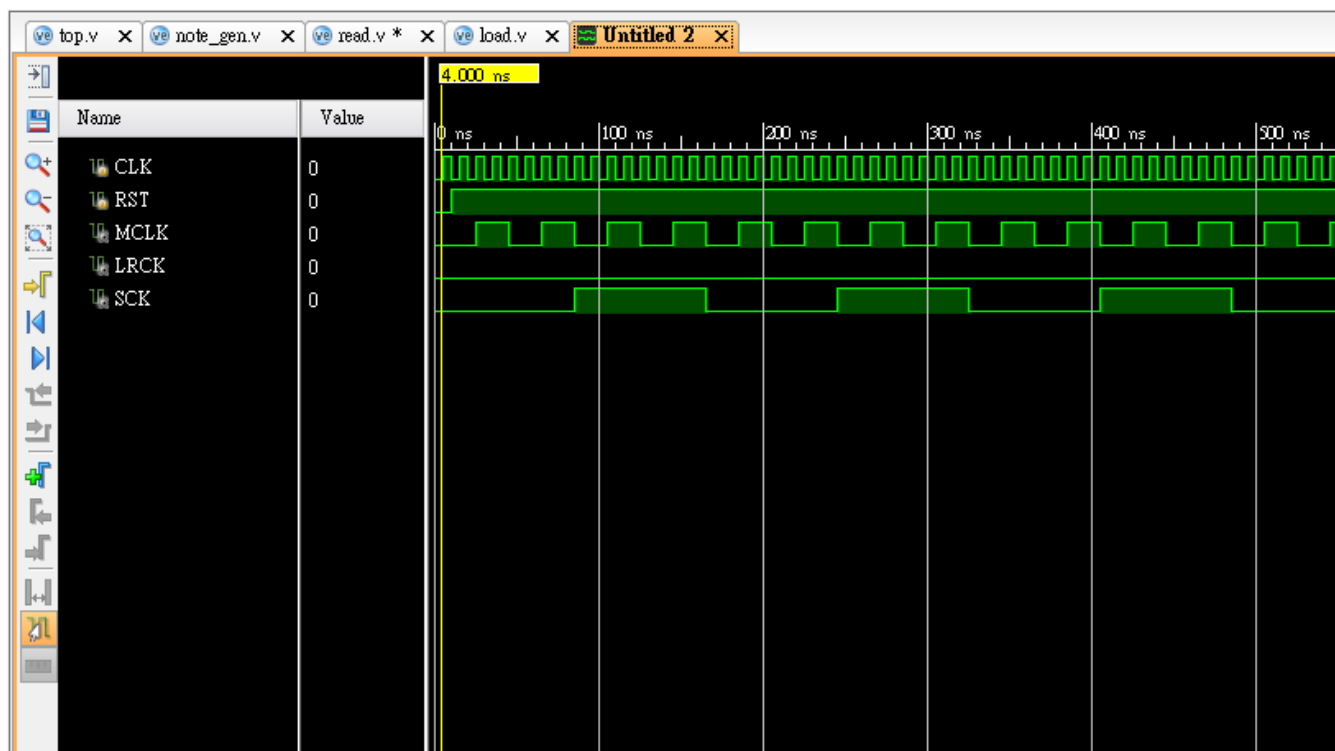
圖片的下半部是整個 lab8_1 的接線圖，上半部是 read 的接線圖。其中每個 module 的功能如下：

Note_gen：產生 audio_right 跟 audio_left，照講義上面打就差不多了。

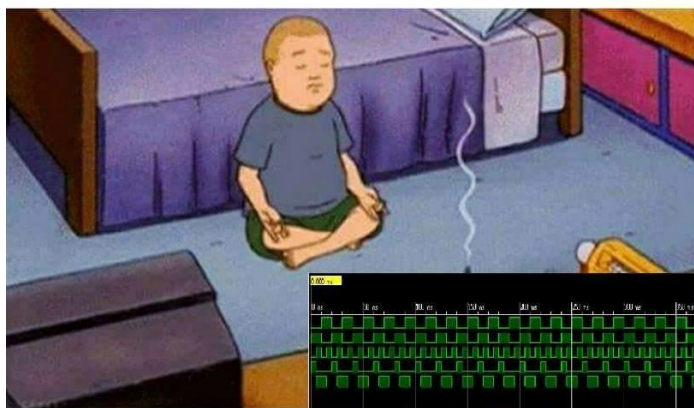
Read：負責產出三大 clock 與 serial input。藉由同一個除頻器的 module 但餵給不同的上數極限 limit，即可產出三大 clock。再把 sampling clock 接到一個 DFF 即可產出 serial output。

Discussion

波形模擬的結果如下：



佛系邏輯設計



不寫Testbench 不做邏輯化簡
不用管電路有沒有接通
不用管位元數量
緣分到了 signal和wave會自己出現

(博君一笑，轉自 FB 粉專「彥斌的程式小教室」)

2. Speaker control

Design Specification

(1) Input:

clk: Global 100MHz clock。

rst: 開關，接在 dip switch 上。

Do: 接在 push button 上，按下去時產生 Do。

Re: 接在 push button 上，按下去時產生 Re。

Mi: 接在 push button 上，按下去時產生 Mi。

up_in: 接在 push button 上，負責調高音量

down_in: 接在 push button 上，負責調低音量。

(2) Output:

audio_mclk: Master clock。

audio_lrck: Left_right clock。

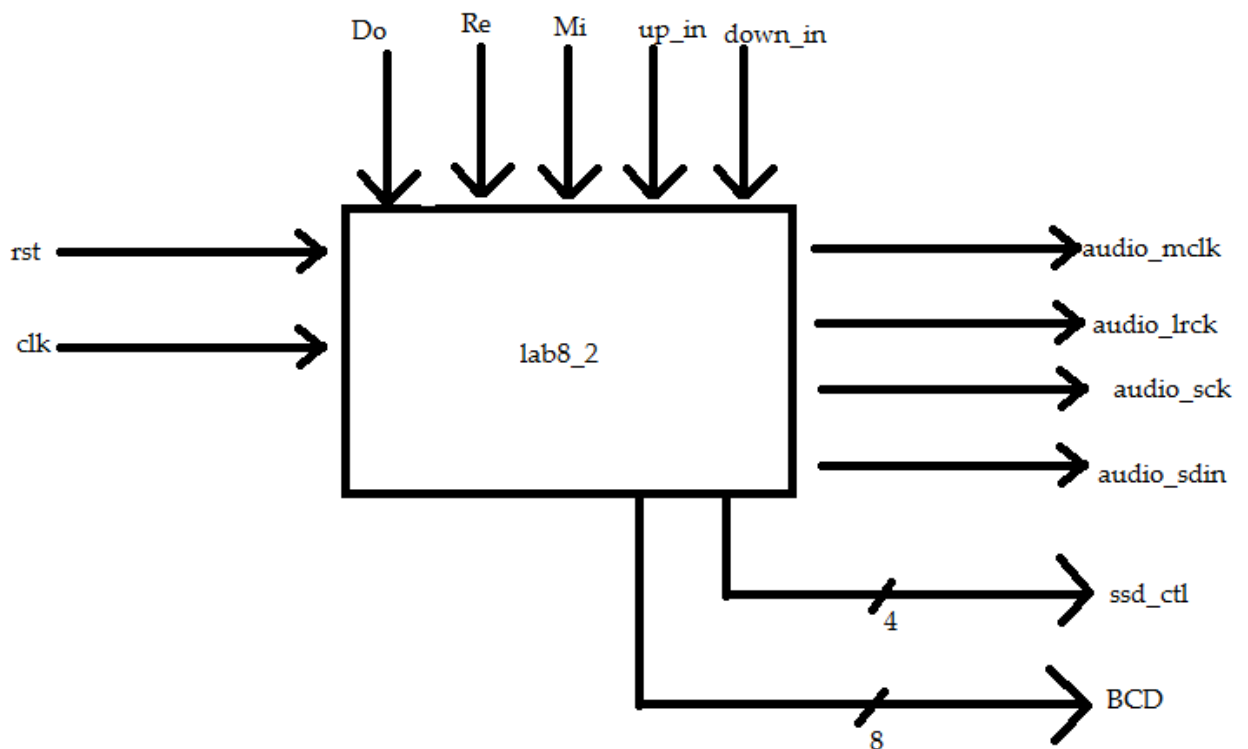
audio_sck: serial clock。

audio_sdin: serial data input。

[3:0] ssd_ctl: 控制七段顯示器。

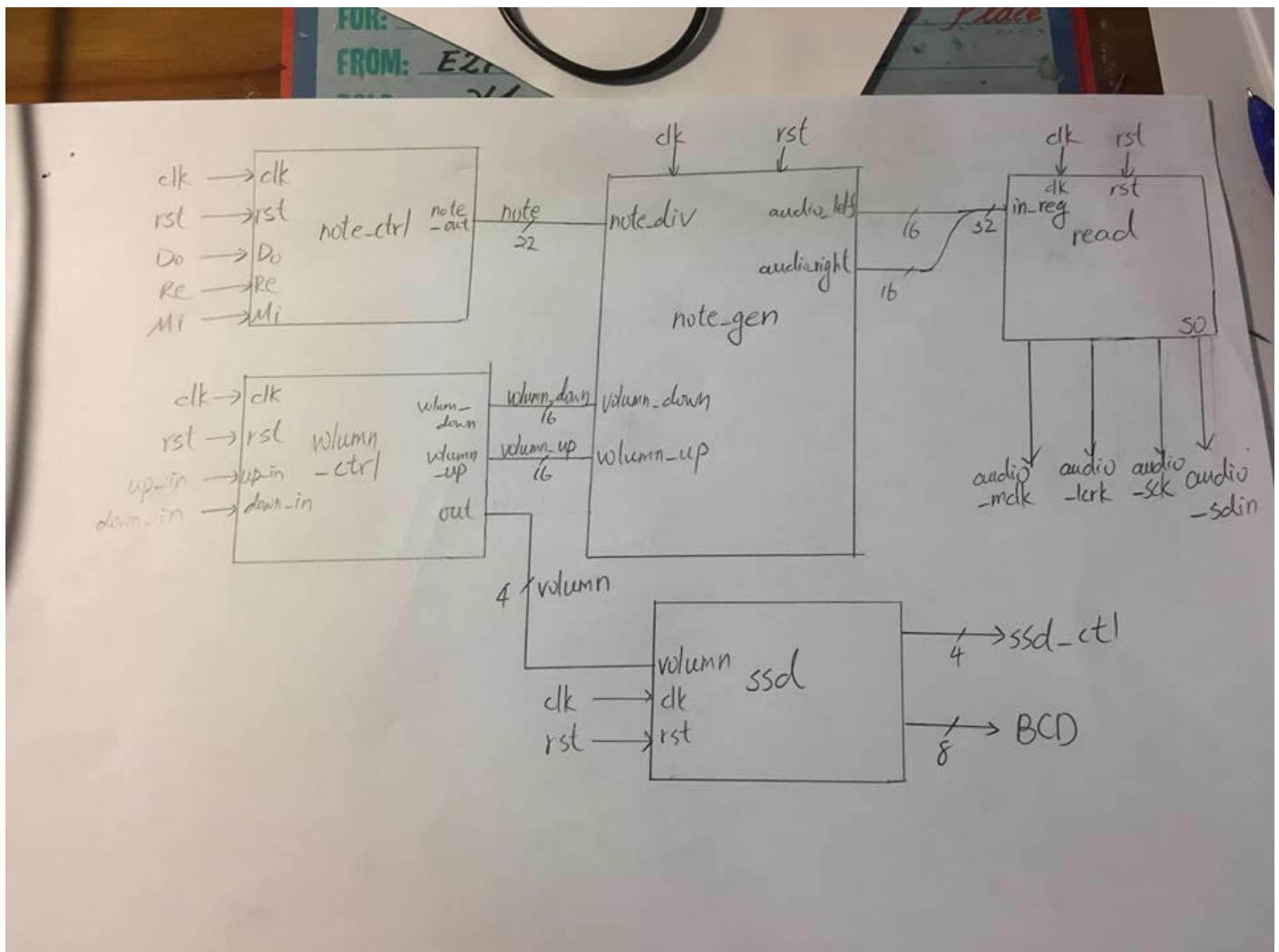
[7:0] BCD: 七段顯示器的各段。

(3) Block diagram:

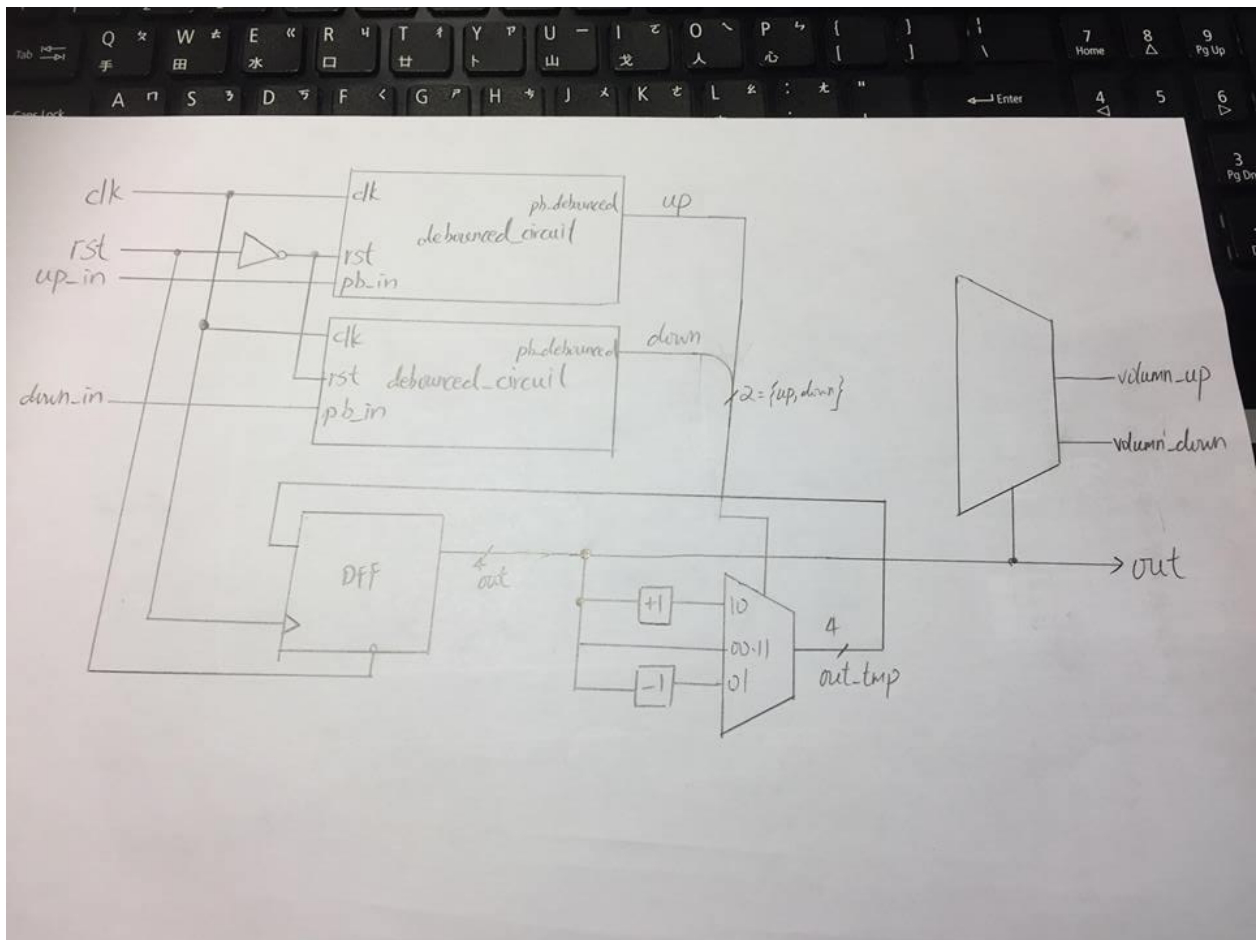


Design Implementation

下圖是整個 lab 的 block diagram :



下圖是 `volumn_ctrl` 的 block diagram :



各個 module 的功能如下：

`Note_ctrl`: 產生 Do、Re、Mi 三個聲音的頻率。

`Volumn_ctrl`: 輸入兩個控制音量的按鈕，控制播放的音量 `volumn_up` 跟 `volumn_down`。

`Note_gen`: 產生 `audio_right` 跟 `audio_left`，照講義上面打就差不多了。

`Read`: 與第一題一模一樣，負責產出三大 clock 與 serial input。藉由同一個除頻器的 module 但餵給不同的上數極限 `limit`，即可產出三大 clock。再把 `sampling clock` 接到一個 DFF 即可產出 serial output。

`Ssd`: 顯示音量(1~16)。

Discussion

我自己對這一題有一些改進的想法。當我暗下控制音量的按鈕時，音量並不會一格一格往上升，而是隨機往上跳一個數字。放一個 `one_pulse` 進去可能就可以了。

Conclusion

控制 speaker 的奧義就在 `read` 的那三個除頻器和名為 `load` 的 `shift register` 之中。其他的部分大體都是一些簡單的組合邏輯。



References

《上課的講義》，應該是馬席彬教授所著，應該是 2018 年

《黃元豪教授的講義》，應該是黃元豪教授所著，應該是 2018 年

《從漫畫了解 FGO！》，リヲ著，Type-Moon，2015 年，