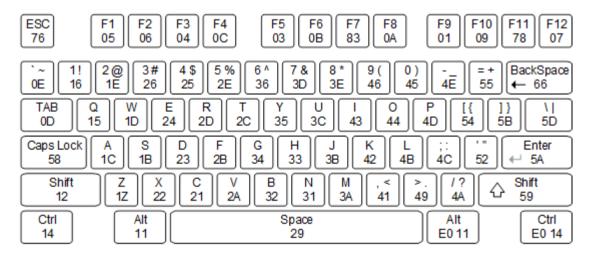
FPGA PS2

- 1. PS/2:負責 FGPA 和鍵盤的溝通。這邊只介紹 Device to Host,也就是鍵盤到 FGPA 的方向。而其實這套協定是雙向的,只是這邊只需要一個方向即可。
- 2. 在沒有發生任何事情時,時脈線和資料線都會保持在 1,在按下鍵盤按鍵之後,鍵盤會傳出一組時脈,以及傳出一系列的資料,每一個 bit 的 資料都是在負緣的時候被偵測,一個按鍵總共會有 11bits 的資料會被傳出來。第一個 bit 是 start,永遠是 0,再來的 8bits 則是代表按鍵的編號,最先開始傳的是最末位的 bit,接著是 parity bit,用來檢查傳輸的資料是否正確,最後一個 bit 則是 stop,永遠為 1,在一組資料傳完後兩條線都會回復到 1。若是按鍵被壓下一段



時間,則鍵盤會不斷的傳輸很多組的11bits,直 到按鍵被放開為止,另外,但按鍵被鬆開時,會傳出另一組特定的11bits,也可以用來判斷是否已經鬆開按鍵。

3. 下表是按鍵以及對應的編碼。

(鬆開: F0, ex. 按下T鍵 (收到2C) -> 鬆開T鍵 (依序收到 F0, 2C)

- 4. 本次實作採用比較簡單的方法,直接利用鍵盤傳輸的時脈來做運算,也就是完全相信鍵盤不會出錯,只要時脈一來就開始進行資料的抓取,也不利用 parity bit 來判斷是否有誤,大致上來看會是正確的,但只要時間一久,或是在鍵盤上按得太快太密集,整個正確性就會消失,所以還是需要一些方法來避免上述狀況。解決辦法則是將板子上的時脈也注入到同一個 module 裡面,因為其大約在50 MHz 的數量級,而鍵盤產生的時脈則大約在 10kHz~16.7kHz,兩者之間有個落差,所以可以透過一個計數器來判斷,若此計數器的數字到達一定程度的話就表示已經 很久沒有鍵盤時脈的到來,一方面可以省去要用 F0 來判斷按鍵鬆開的麻煩,一方面若中間有出錯的話,也可以透過這個機制來將內部儲存資料的地方歸零,以便在下一筆正確的資料來臨時不會一直出錯下去。
- 5. further information and example: https://embeddedthoughts.com/2016/07/05/fpga-keyboard-interface/