



CVSD HW2

Speaker: 施敬彥

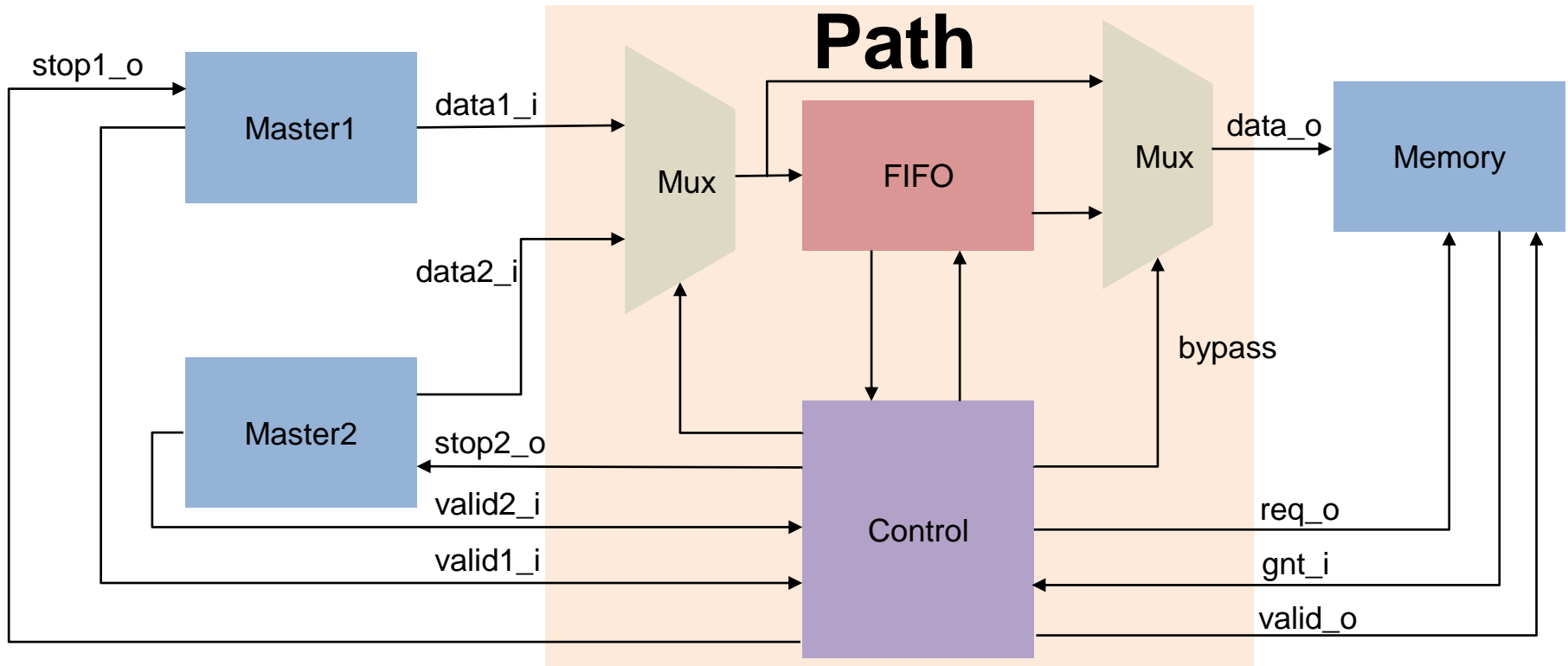
E-mail: cys@access.ee.ntu.edu.tw

Date: 2018/10/9



作業說明

- ❖ 一般的系統當中，可能會有許多個裝置想要存取Memory，此時，需要有一個溝通協定來處理此狀況，避免data collision發生。
- ❖ 這次作業中，用來處理此狀況的module叫做"path"，他會交替的給各個master存取權限，讓他們可以將資料送出。若沒有path允許，master不能送出valid的資料。且master也未必會"連續"送valid資料 (p.8)
- ❖ 而由於memory並不是隨時都處於可以接受資料的狀態，因此，path裡面會有fifo暫存資料，避免資料丟失。
- ❖ 這次作業需要自己實作path module以及testbench的部分
 - ❖ Testbench: 用來模擬兩個Master及memory的行為，請自行設計，若設計不完整，可能會使你的path電路無法通過助教的testbench
 - Master電路: 當stop不為1時，才能送出valid訊號，也就是說可以在stop為0的任何時候發送valid訊號。
 - Memory電路: 當收到req後，可能馬上或隔一段時間才會發出gnt。並且當valid_o為1時，會抓取data_o資料存入
 - Master和Memory(用二維陣列模擬，不須addr)
- ❖ 中間電路要怎麼設計(要不要檔flip-flop)都可以，只要你tb符合規範(p.8)，助教在valid_o拉起來時能收到正確的值即可





Path.v

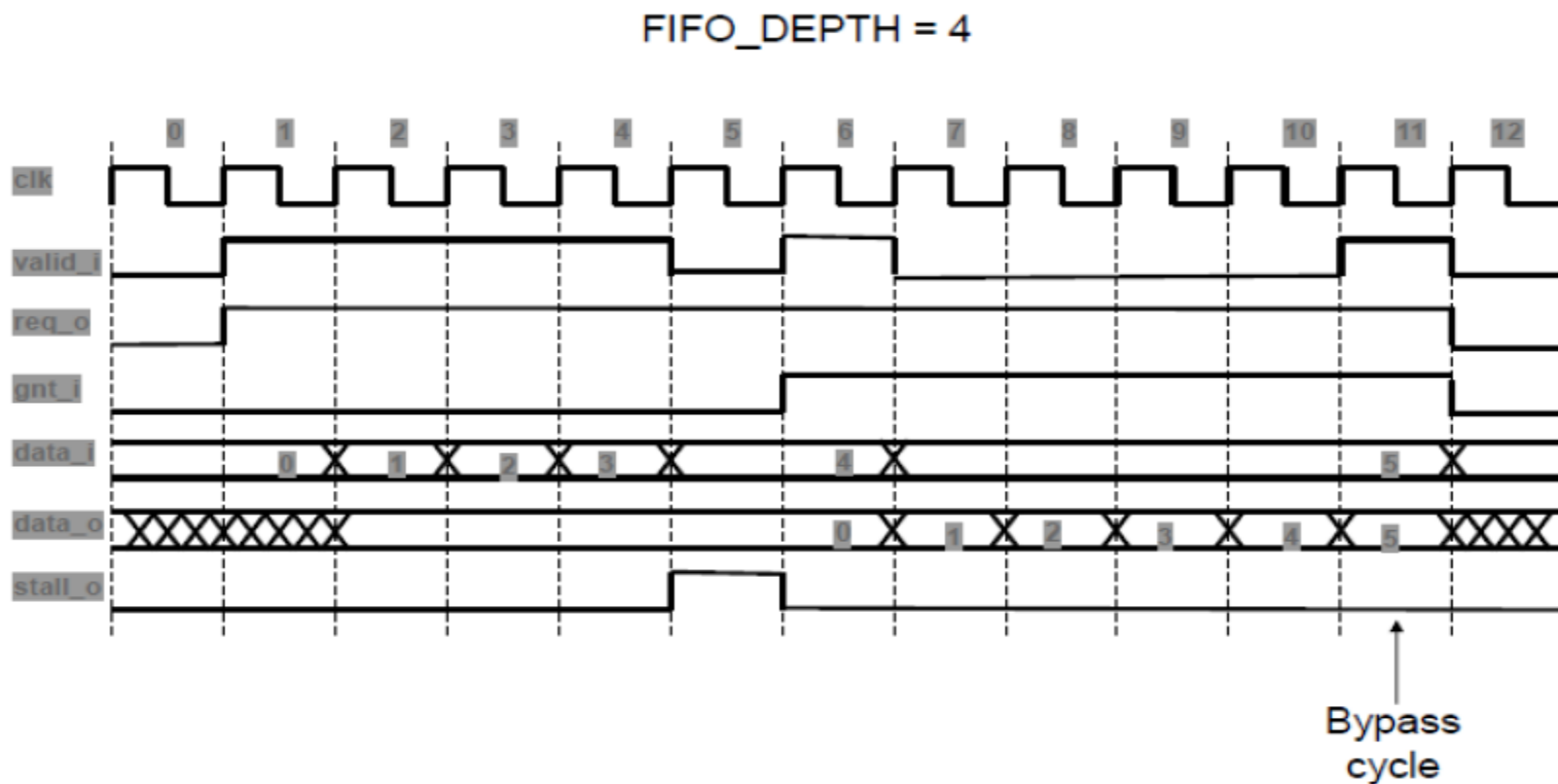
- ❖ 收到10筆master1有效資料後，會將通道存取權轉交給master2
- ❖ 收到10筆master2有效資料後，會將通道存取權轉交給master1
- ❖ Master1 先有存取權
- ❖ 當fifo滿了且memory又不允許送東西給他，則stop1, stop2可能同時為1

信號名稱	位元寬度	說明
clk	1	時脈訊號
rst_n	1	低位準非同步之系統重置訊號
data1_i	8	Master1給Path的資料
data2_i	8	Master2給Path的資料
valid1_i	1	當valid1_i拉起時，表示data1_i資料有效
valid2_i	1	當valid2_i拉起時，表示data2_i資料有效
req_o	1	path要求傳送訊號給Memory
gnt_i	1	Memory允許path輸出訊號
stop1_o	1	path告訴Master1停止輸入訊號，若為0，則Master1可以輸入訊號
stop2_o	1	path告訴Master2停止輸入訊號，若為0，則Master2可以輸入訊號
data_o	8	Path之output訊號
valid_o	1	表示此時path輸出訊號為有效



Spec – Path Block

- ❖ $\text{FIFO_DEPTH} = \text{FIFO最大儲存量}$
- ❖ 此為波形說明檔，實際作業 $\text{FIFO_DEPTH} = 5$





Spec - Breaking Down the PATH Design

❖ Protocol

- ❖ gnt_i 一定會在req_o後發生，不會平白無故就有gnt_i
 - 你要我才給你，我不會自己給你存取權
- ❖ 如果stall_o == 1，valid_i就不可能會是1
- ❖ 一個valid_i會對應到req_o
 - valid_i代表有有效資料進入path，所以應該也要有req_o，否則資料會卡在path
- ❖ data_i會對應到data_o
 - 資料有進去就會出來，並且值不會變



Fifo.v

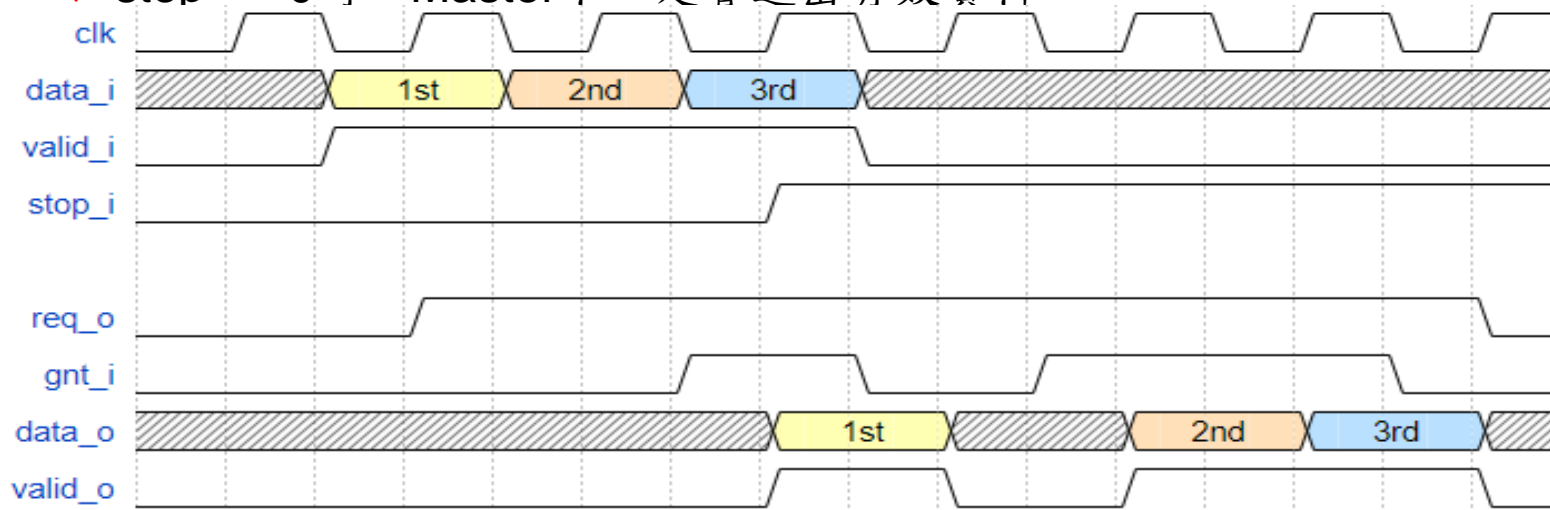
❖ 最大儲存量: 5

信號名稱	位元寬度	說明
clk	1	時脈訊號
rst_n	1	低位準非同步之系統重置訊號
data_i	8	輸入fifo之data
write_i	1	表示此時要將值寫進fifo中
read_i	1	表示此時fifo要輸出值
full_o	1	表示fifo此時為滿的
empty_o	1	表示fifo此時為空的
data_o	8	fifo輸出之data



tb.v (資料皆在負緣給or取值)

- ❖ 請將Master1讀入之檔案名稱設為pattern1.dat (自己設計pattern)
- ❖ 請將Master2讀入之檔案名稱設為pattern2.dat (自己設計pattern)
- ❖ 請將Memory讀入之檔案名稱設為golden1.dat
 - ❖ 此檔案為ground truth，用來驗證你的design是否正確
- ❖ 請思考各種可能狀況，例如
 - ❖ memory不會馬上回gnt，導致fifo可能會滿
 - ❖ stop == 0時，Master不一定會送出有效資料





備註

- ❖ Data會進入path module當 $\text{valid1_i} \ \&\& \ \text{!stop1_o}$ 或是 $\text{valid2_i} \ \&\& \ \text{!stop2_o}$
- ❖ 如果path沒有馬上把資料往外送(bypass)，則data會存在fifo中
 - ❖ $\text{bypass} = \text{empty} \ \&\& \ \text{gnt_i} \ \&\& \ \text{valid_i}$
 - ❖ $\text{valid_i} = \text{valid1_i} \ || \ \text{valid2_i}$
- ❖ Data離開path藉由req/gnt的機制
 - ❖ Path向memory發出請求(req)，memory同意後(gnt)，才能傳送出去
- ❖ 當fifo滿了而且memory不同意接收資料時，則stop拉起來(不准master送有效東西進來)



繳交檔案

- ❖ 使用ftp上傳到140.112.174.174中的hw2資料夾
 - ❖ account: 1071cvsd
 - ❖ password: cvsd2018
- ❖ r07943xxx.zip
 - ❖ r07943xxx/
 - fifo.v
 - path.v
 - tb.v
- ❖ 注意
 - ❖ r07943xxx.zip在Linux系統下執行unzip指令後，需要產生一個r07943xxx的資料夾，裡面含有fifo.v, path.v, tb.v這三個檔案
 - ❖ 修改檔案請上傳r07043xxx_vk.zip, k為1,2,3.....，解壓縮後產生的資料夾為r07943xxx（沒有'_vk'）
 - ❖ 如果繳交的檔案沒有符合上述規定，將會扣15分