NCKU-ES
Introduction to Digital IC Design
Fall 2017

Final Project

System On Chip

Professor:Wen-Long Chin

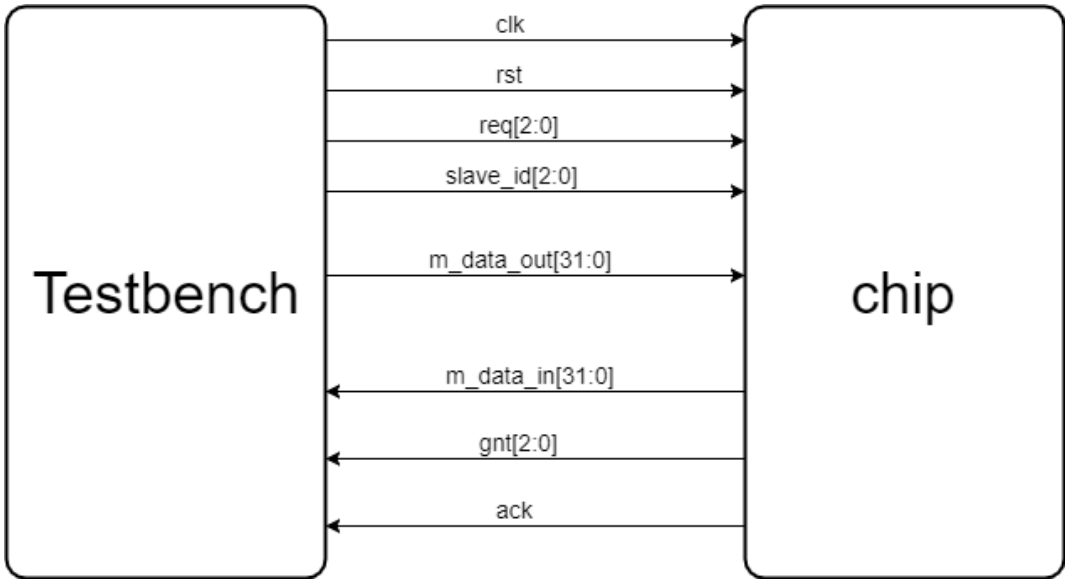
TA: Cheng-Chieh Chang、

Chieh-Sheng Cheng

VLSI signal processing LAB

Design Description

- The block diagram is shown below. Please finish the design named as “chip”. The masters are behavior models, which will be provided together with the testbench. There are three masters (testbench), two slaves (MAC and FFT accelerators), and one arbiter in the system-on-a-chip (SoC). The data bus, arbitrated by the arbiter, is shared among all masters and slaves.



Specifications

- Top module name : chip (File name : chip.v)
- Input pins : clk, rst, req[2:0], slave_id[2:0], m_data_out31:0]
- Output pins : gnt[2:0], ack, m_data_in[31:0]

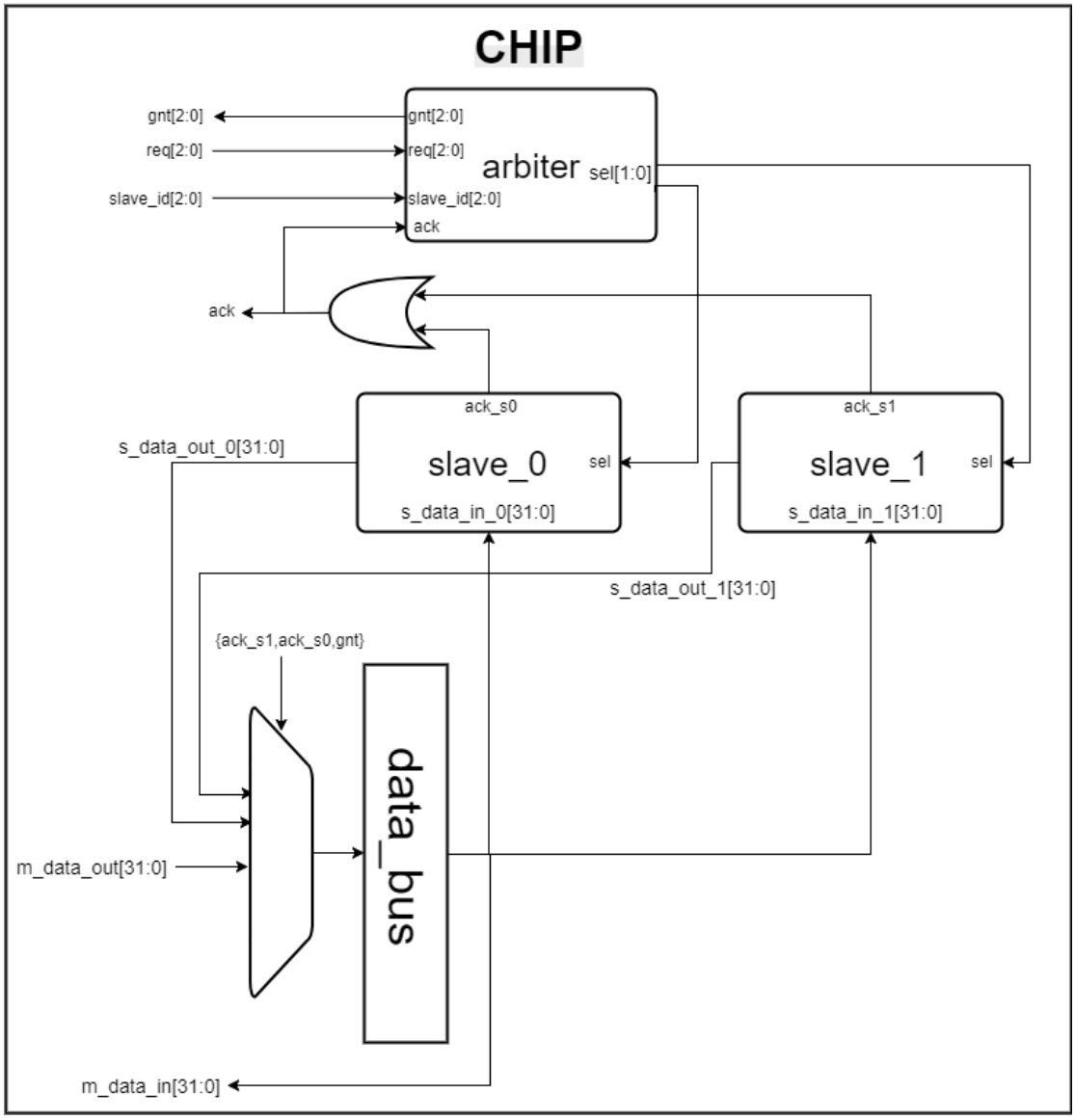
- Module name : arbiter (File name : arbiter_pri.v/arbiter_RR.v)
- Input pins : clk, rst, req[2:0], slave_id[2:0], ack
- Output pins : gnt[2:0], sel[1:0]

- Module name : slave_0 (File name : slave_0.v)
- Input pins : clk, rst, sel, s_data_in_0[31:0]
- Output pins : ack, s_data_out_0[31:0]

- Module name : slave_1 (File name : slave_1.v)
- Input pins : clk, rst, sel, s_data_in_1[31:0]
- Output pins : ack, s_data_out_1[31:0]

信號名稱	輸入/輸出	位元寬度	說明
clk	input	1	時脈訊號。
rst	input	1	高準位非同步(active high asynchronous)之系統重置信號。 說明:此信號於系統啟動時送出。
req	input	3	主控端 3 個 Master(M0, M1, M2)請求使用系統訊號。 說明:當主控端欲使用系統進行資料傳輸及計算時，會將此信號相對應之位元設為 high。 呈現 3'b001，表示 M0 發出使用請求。 呈現 3'b110，表示 M2 及 M1 同時發出使用請求。
slave_id	input	3	主控端 3 個 Master(M0, M1, M2)請求使用系統之指定 slave 訊號。 說明: slave_id[i]=0 時，表示 Mi 請求使用 slave_0(MAC)。 slave_id[i]=1 時，表示 Mi 請求使用 slave_1(FFT)。
m_data_out	input	32	Master 發出的串列輸入資料。 MAC 資料輸入形式:每 cycle 輸入一對 a、b 分別為 m_data_out[15:8]、m_data_out[7:0] 共 8 cycle 為有效輸入。 FFT 資料輸入形式:每 cycle 輸入兩筆 16 位元時域訊號 m_data_out={x(t) , x(t+1)}，t 為時間 共 4 cycle 為有效輸入。
gnt	output	3	主控端 3 個 Master(M0, M1, M2)接收之授權訊號 說明 : gnt[i]=1 時，Mi 即輸出 m_data_out_i。 (注意:系統同時間只授權一 Master 送出有效輸入資料)
ack	output	1	串列輸出致能訊號。 說明:當此訊號為 1 時，表示回傳資料 m_data_in 為有效的。
m_data_in	output	32	主控端接收之回傳資料。 MAC 資料回傳形式:共 8 對 a、b 經乘加後回傳資料 FFT 資料回傳形式:每 cycle 序列回傳一筆 32 位元資料 m_data_in ={X(t)}，t 為時間。

Functionality



- The master i, i=0, 1, 2, requests the data bus and slave_j, j=0, 1, by the signals, req[i] and slave_id[i], respectively, to the arbiter.
- If slave_id[i]=j, once granted by the arbiter via the signal, gnt[i], the master can send data via the signal, m_data_out, to slave_j via the data bus. Meanwhile, the slave j will be selected by the signal, sel[j], to receive data.
- When the slave receives data, the operation will start immediately. After the operation is done, the signal ack_sj will be asserted until the data has been transferred to the master i via the signal, s_data_out_j.
- After the de-assertion of ack[j], req[i], gnt[i] and sel[j] will de-assert, too. Then, the next master will be granted and the above handshake

protocol will repeat again. Note that, there is only one set data in a transaction.

- Arbiter priority rule

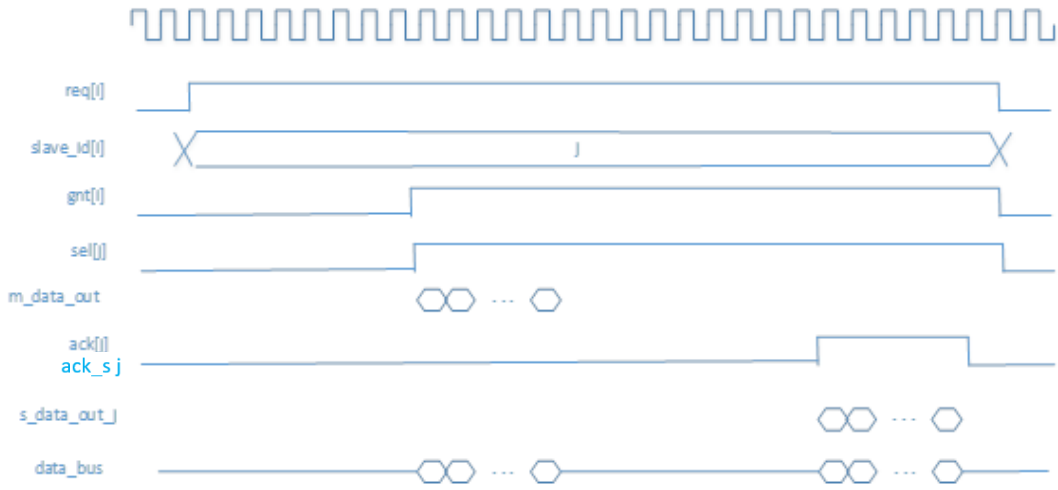
1. Round robin arbiter (arbiter_RR.v) :

```
always@(*)
begin
    state_ns=state_cs;
case(state_cs)
    IDLE: if(req[0])
            state_ns=M0;
          else  if(req[1])
            state_ns=M1;
          else  if(req[2])
            state_ns=M2;

    M0:   if(cmd_done & req[1])
            state_ns=M1;
          else  if(cmd_done & req[2])
            state_ns=M2;
          else  if(cmd_done)
            state_ns=IDLE;

    M1:   if(cmd_done & req[2])
            state_ns=M2;
          else  if(cmd_done & req[0])
            state_ns=M0;
          else  if(cmd_done)
            state_ns=IDLE;

    M2:   if(cmd_done & req[0])
            state_ns=M0;
          else  if(cmd_done & req[1])
            state_ns=M1;
          else  if(cmd_done)
            state_ns=IDLE;
endcase
end
```

```
assign cmd_done=~ack & ack_r;
always@(posedge clk)
  ack_r <= ack;
```

2. Prioritized arbitration with master 0 (M0), master 1 (M1), and master 2 (M2) exhibiting the highest, second highest, and lowest priorities (arbiter_pri.v)

- Waveform



- Testbench check

```
# ------------------------------------------------------
# Oops!!! Your code can not work...!!
# ------------------------FAIL------------------------
```

If transcript show the message, it means that your code never get the right answer.

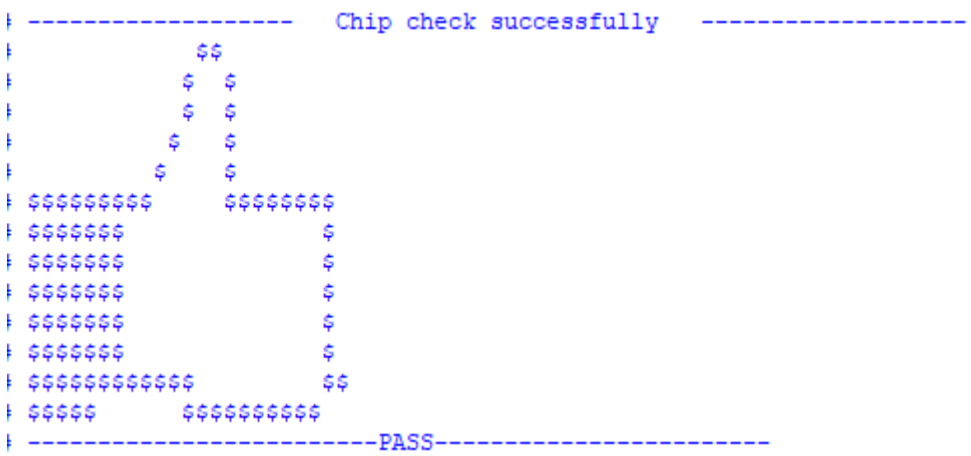
```
#
# ------------------------------------------------------
#
# Fail!! There are          49 error with your code!
#
# ------------------------FAIL------------------------
```

If transcript show the message, it means that sometimes your code get the right answer but your code still goes wrong.

- Note
- 本測驗提供測試程式(testbench)，各位同學之設計需能通過 testbench 的驗證
  - 書面報告(report.doc)需包含：

    (請詳細描述原理及分析波型)

    1. 設計原理(Design principle)
    2. 架構設計(Architecture)
    3. 波型(Waveform)分析
    4. Pass 截圖
    5. 工作分配比重(組員須簽名，兩人一組，與實驗課分組相同)



  - 書面報告請勿手寫
  - 評分標準依是否達到題目要求各事項，未達到要求依項扣分
    1. 功能驗證:60%
    2. 書面報告:40%
  - 繳交的作業資料夾組織與命名請與下頁圖示相同
  - Design code 上傳繳交至 Moodle
  - 書面報告請繳交紙本給助教
  - Due date: 2018/1/19（五）23:55 (上述兩者相同)

Directory Organization