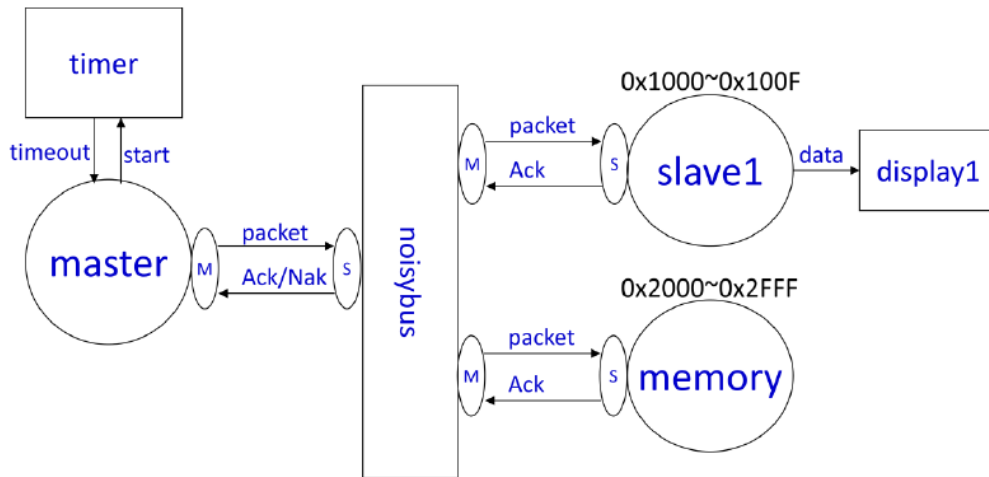


Assignment 1: A Simple Example

105062600 Yi-cheng,Chao 07:10, March 19, 2017

Design Concepts – Basic Architecture



Design Description – Packet

1. 更改 data member 至符合 spec，對於原本既有的 data member 中，我僅保留 seq 這個 int 型別變數作為判斷 packet order 以及是否有 packet dropped 發生的依據(若有 packet dropped 發生則 seq 會被重製回 0)，packet 的 data member 如下所示

```
sc_uint<32> addr; // The address field
sc_uint<8> data; // The data field
bool rw; // 1: Write; 0: Read
bool ack; // 1: Ack; 0: Nak
int seq;
```

Design Description –Master

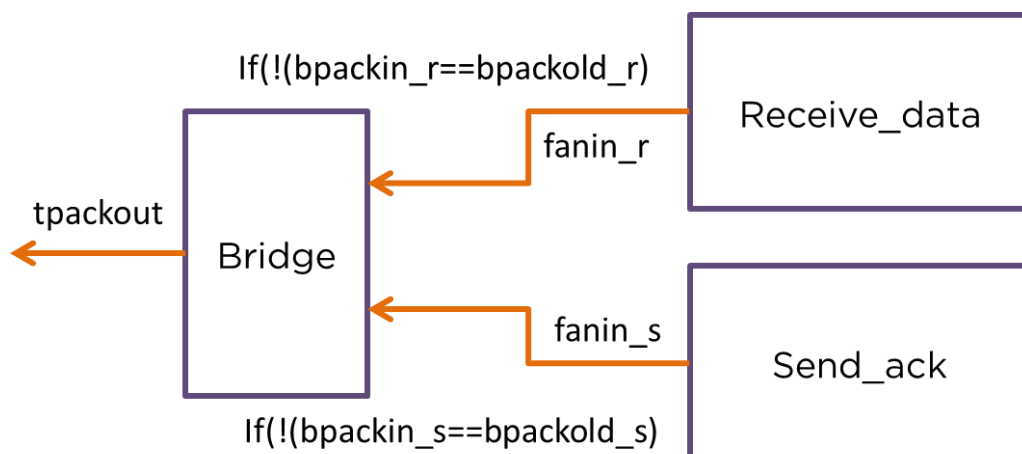
1. 資料擷取實作方法我使用 file stream 開啟檔案並使用 getline 逐行抓取資料到 string 內，再使用 string stream 解析該 string，從 string 長度上來做 input validation，對於較長的 write 指令，{instruction, address, data}字串長度至多 12，因此只要超過便為 invalid input，並如以下輸出：

```
-----
Master Read File = 0 0x11001 0x110F 73
Invalid input was found in MasterLog.txt, it is 0 0x11001 0x110F 73
-----
```

2. 相較於原本的 transmit module，額外再增加一個叫做 receive_data 的 function member 來進行 packet 的接收與寫檔，寫檔部分則同樣使用 fstream 來操作，timeout 部分亦同。

Design Description –Noisybus

1. 額外增加兩個 ports 連接 memory，在 receive data 的部分使用兩個 packet_type 型別變數來記錄上一個 packet(分別名為 mpackold 與 rpackold)，由是否與上一個 rpackin 或 mpackin 相異來判斷該 packet 來自於 slave1 還是 memory。
2. 此外在資料處理上為了解決 multiple driver 的問題，新增 function member bridge 來進行連接，在 function member 間的連接使用 sc_signal 溝通，判斷信號來源概念同上。
3. 以上兩點設計概念如下表示(以第二點為例)，因為不會同時發生兩者 packet 皆有更新的狀況，因此便能解決 multiple driver 的問題。



Design Description –Slave1 & Memory

1. 宣告各自需要的容器(型別使用 sc_uint<8>)在 class 內，便可以完成暫存器的設計，之後只要由 address 判斷資料的位置，便可進行寫入/讀出，片段程式碼如下所示：

```
sc_uint<32> tmp_addr = packin.addr;

cout << "Slave1: got packet no. = " << packin.seq << endl;

int pos = tmp_addr % 16;

if(packin.rw == true){
    slave1_reg[pos] = packin.data;
}
else {
    packin.data = slave1_reg[pos];
}
```

Result Display

```
-----
Master Read File = 0 0x1004
Parsing data: [Instr] 0[Addr] 4100[Data] 0
Master: Sending packet no. 20
Timer: timer start detected
Noisybus: Received packet seq no. = 20
Noisybus: Random number = 1131176229
Slave1: got packet no. = 20
Address 0x1004, Read 55 successfully
Display: Data value received, Data = 55
Packet is in noisybus again
Noisybus: Received Ack for packet = 20
Master: Receive packet no. 20
Read 0x1004 55, Aak
-----
```

1. 每個封包資訊皆以虛線切割，對於封包資訊除了各自 Log 內的寫檔外我亦有 cout 至顯示器上，方便追蹤 packet order 以及整體 packet 在 design 中的流動狀況。

Difficulty Encounter & Experience

由於之前沒有使用過 SystemC 的經驗，在 API 的使用上花費了我很多時間，看了很多資料以及經過了不少的 Trial and Error，在 debug 時 trace packet 還是使用基本的 printf 大法，但其實最後發現都是自己考慮不夠周全所致，例如以下 Bug:

```
Read 0x1004 55, Aak
-----
Transmit Read File = 0 0x1003
Parsing data: [instr] 0[addr] 4099[data] 0
Transmit: Sending packet no. 21
Timer: timer start detected
Noisybus: Received packet seq no. = 21
Noisybus: Random number = 859484421
Receiver: got packet no. = 21
Address 0x1003, Read 4 successfully
Display: Data value received, Data = 4
Hello! You are in noisybus again
Noisybus: Received Ack for packet = 21
Transmit: Receive packet no. 21
Read 0x1003 4, Aak
-----
Transmit Read File = 0 0x1100
Parsing data: [instr] 0[addr] 4352[data] 0
Transmit: Sending packet no. 22
Receiver: got packet no. = 21
Address 0x1003, Read 4 successfully
Timer: timer start detected
Noisybus: Received packet seq no. = 22
Noisybus: Random number = 608413784
Invalid address 0x1100, read failed
Hello! You are in noisybus again
Noisybus: Received Ack for packet = 21
Transmit: Receive packet no. 21
Read 0x1003 4, Aak
Receiver: got packet no. = 21
Address 0x1003, Read 4 successfully
Hello! You are in noisybus again
Noisybus: Received Ack for packet = 21
```

上一筆packet動作是 read,因此改變過 packet的data

Invalid address

導致在receiver端下個pos.edge在 if(packin == packold)不會return, 重複送出packet形成Deadlock

最後發現只要在讀寫資料改變前先 assign packet 到 packold 便可以輕鬆解決，若能在整體寫程式前先周全的想過，應該便不會走向遲交一途，不過想跟教授建議不知道作業是否能有至少兩個禮拜的期限，因為還有其餘課堂科目，若能有兩個假日的時間將會有更多的時間分配可以利用，謝謝！