三下專題

Design of Distributed Shared-Buffer Network-on-Chip(NoC) Router

目錄

1. 專題實作

在 Router 的設計上，目前主流 router 架構可分為三類：Output-buffered router(OBR)、Input-buffered router(IBR)、Distributed shared-buffer(DSB) router。其中 OBR 和 IBR 為分別將緩衝 memory 放在輸入端和輸出端，以承受網絡的資料吞吐，下面是 OBR 和 IBR 的簡單架構：

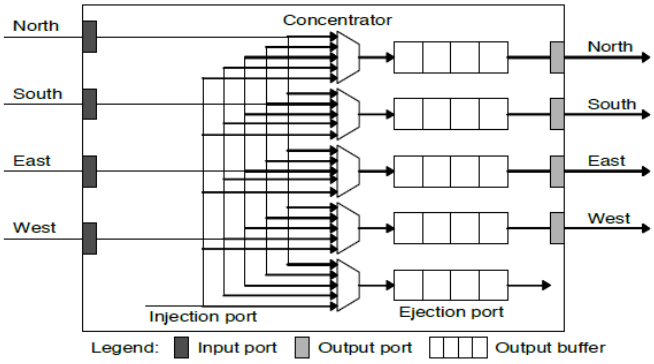


Fig. 2. OBR architecture model

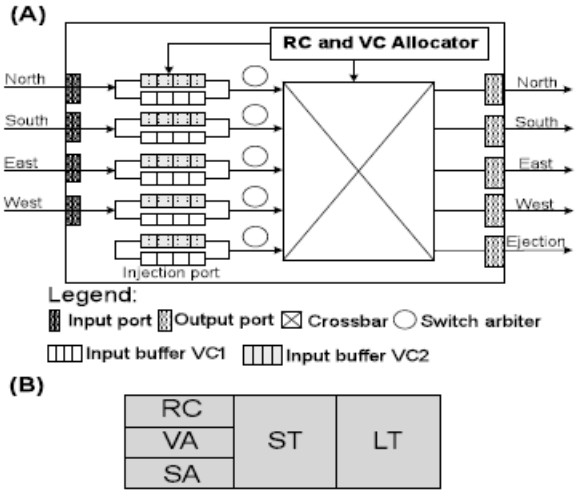


Fig. 3. IBR architecture model with 3-stage pipeline

DSB router 結合了兩者架構，它不直接把 buffer 連接到輸入或輸出的一端，而是以 Middle memory 的方式作為 router 內部的動態分配緩衝，透過兩部分 switching network 和控制電路連接到輸入和輸出端。DSB 的好處是其提供了更靈活的設計方式，如 Packet 中有 timestamp 的話 DSB router 能夠兼容 OBR 的功能和性質，另外其也像 IBR 一樣從內部動態決定資料流向，能做到 pipeline 架構和低功耗表現。

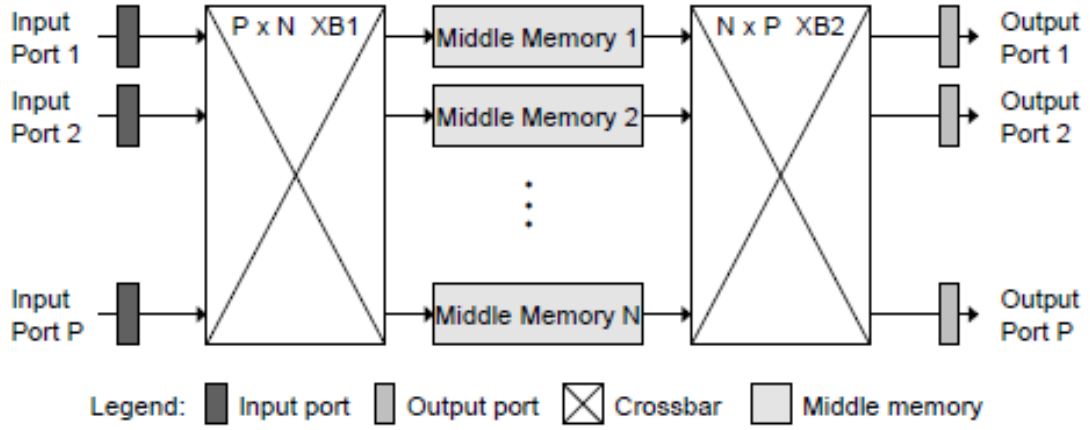


Fig. 4. DSB router architecture model

Arrival conflict and departure conflict in DSB router:

在 DSB router 中，Middle memory 的加入雖然能帶來更多的資料分配決策，但在硬體實現上會出現一些新的問題，如 arrival conflict 和 departure conflict。

Arrival conflict 指的是同一 cycle 多個 input port 同時申請同一 middle memory；而 departure conflict 指的是同一 cycle 多個 middle memory 同時申請同一 output port。隨着分配算法的複雜度，處理 conflict 的難度將越來越高。

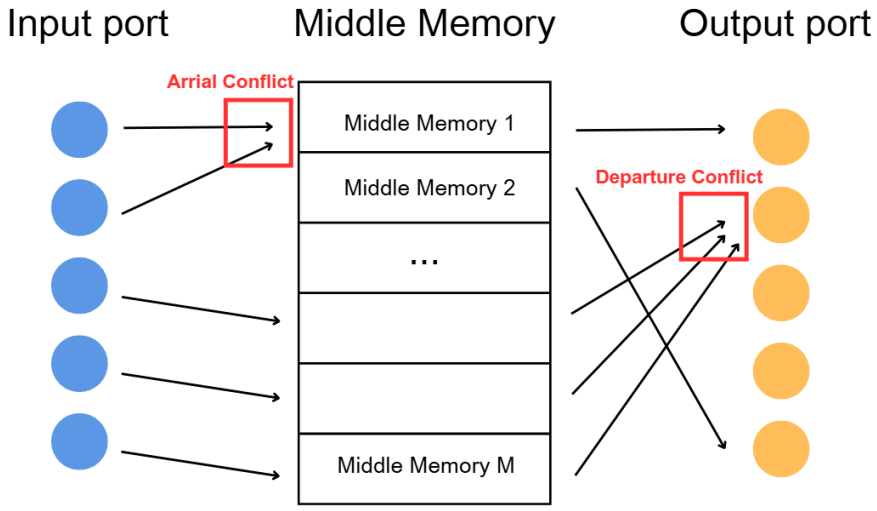


Fig. 5. Arrival conflict and departure conflict in DSB router

在專題中我實作的 DSB router 資訊如下：

- 3-stage pipeline in natural
- X-Y routing in 2D mesh
- Input buffer + Distributed shared buffering
- Middle memory 0 highest priority, memory N lowest priority.
- First-In-First-Out memory
- 5 input buffer, 5 middle memory
- Size of 8 data in buffer/memory

專題實作的參考架構如下圖[1]：

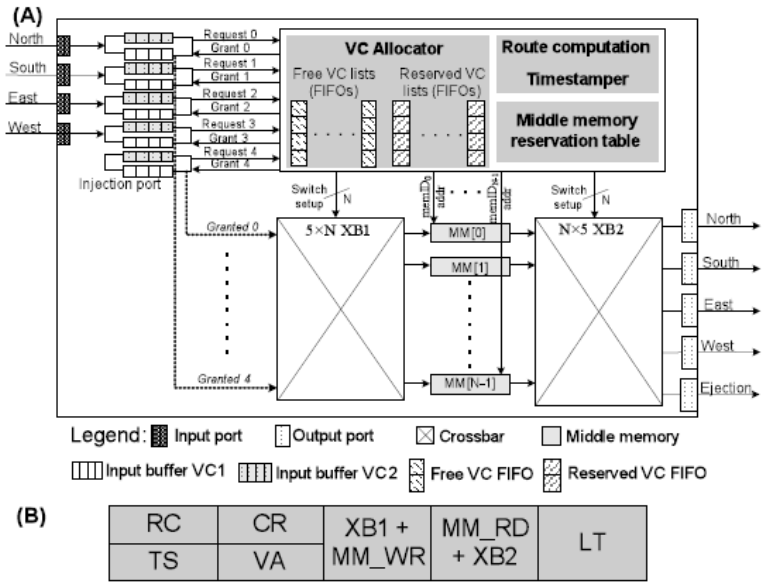


Fig. 6. Distributed shared-buffer (A) router microarchitecture with *N* middle memories and (B) its 5-stage pipeline: (1) Route computation (RC) + timestamping (TS), (2) conflict resolution (CR) and virtual channel allocation (VA), (3) first crossbar traversal (XB1) + middle memory write (MM_WR), (4) middle memory read (MM_RD) and second crossbar traversal (XB2), and (5) link traversal (LT).

參考論文的架構為先接上 input buffer 後，再分配到 middle memory(MM.)進行中間的動態分配。分配的思路為，(RC + TS)先給予每一筆進來的 data 一個預計離開的初始 timestamp，之後(CR + VA) 計算 Arrival Conflict 並重新分配正確的 timestamp，(XB1 + MM_WR)查詢 memory 是否有位置後放入 MM.，(MM_RD + XB2)將合適 memory data 放到 output bus，(LT)決定是否將資料輸出。資料在 NoC 上的傳播路徑演算法為 X-Y routing。

[1] R. S. Ramanujam, V. Soteriou, B. Lin, and L.-S. Peh, "Design of a High-Throughput Distributed Shared-Buffer NoC Router," in Proc. 4th ACM/IEEE Int. Symp. Networks-on-Chip (NoCs), Grenoble, France, May 3–6, 2010, pp. 69–78, doi: 10.1109/NOCS.2010.17.

在實作上，為了簡化設計架構，我將參考論文中的 pipeline stage (CR + VA)和(XB1 + MM_WR)，以及(MM_RD + XB2)和(LT)分別整合在一起。Memory 結構為 FIFO，在 memory 的調用下為 3 個 pipeline stage。另外在 data 的 destination type 的存放上，論文方法中對於 Output Port K，每個 cycle 會有來自 M 個 middle memory 的 M 個輸出候選中找到唯一符合的(由先前 timestamping 決定) destination type 的資料作為輸出；而我把相同的 dest. type data 放在同一 MM. 中，並對所有 MM 做優先排序，循環調整每一種 dest. type 對於 MM.的優先度。我的方法在網絡未接近完全飽和時有良好的效果和適應能力，用 middle memory 的小部分利用率，換取 arrival conflict 和 departure conflict 分在了兩個階段處理，有效平衡了各個階段的電路延遲。

下面是 DSB router 的總體流程圖：

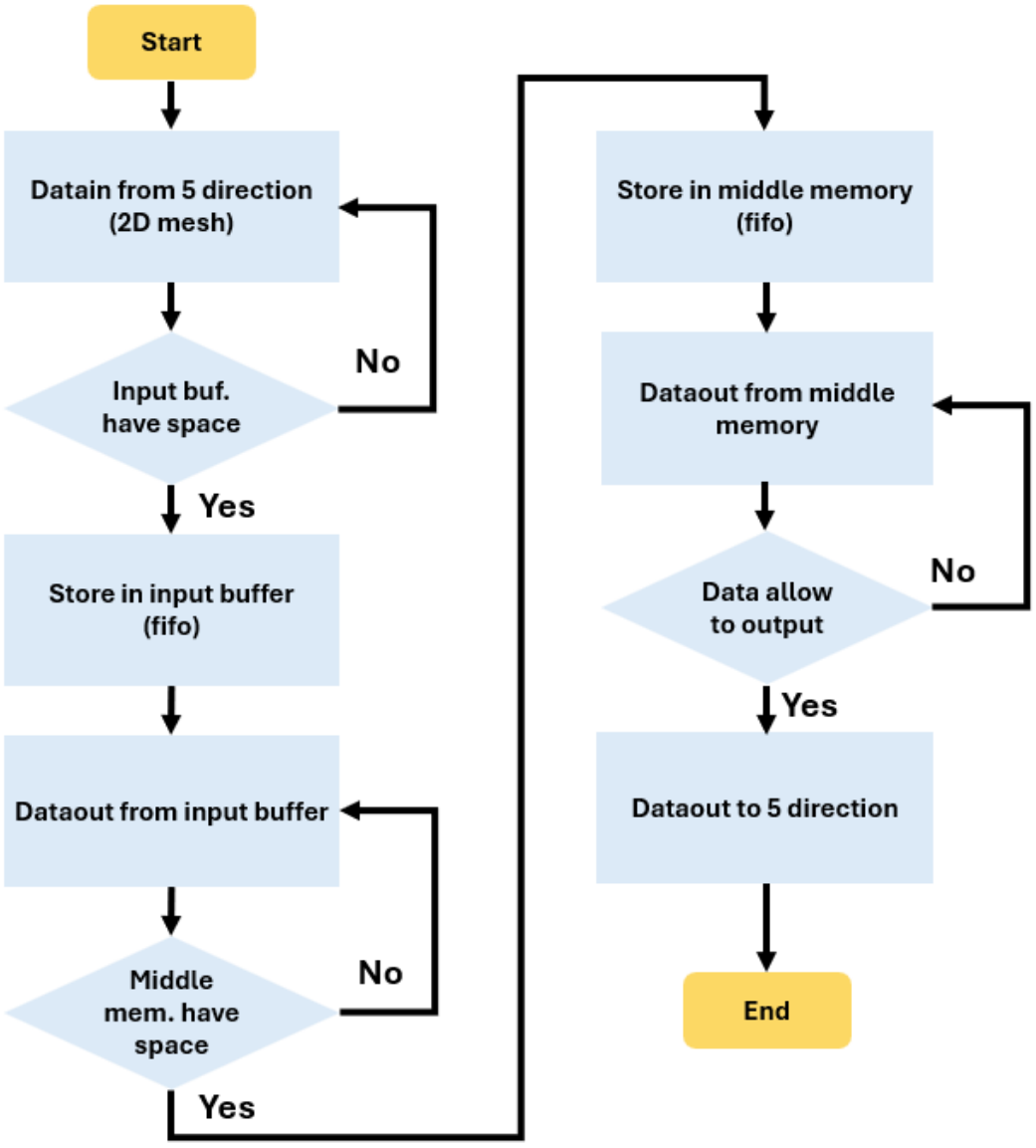


Fig. 7. Overall Dataflow in DSB router

下面是對於實作的 3-stage pipeline DSB router 的具體介紹：

1. First stage: Input data and input buffer storing + Decoder (timestamp)

```
module switch_router_dsb (
    clk, rst, lx, ly,
    P_datain, P_dataout, P_reqin, P_ackout, P_ackin, P_reqout,
    S_datain, S_dataout, S_reqin, S_ackout, S_ackin, S_reqout,
    W_datain, W_dataout, W_reqin, W_ackout, W_ackin, W_reqout,
    N_datain, N_dataout, N_reqin, N_ackout, N_ackin, N_reqout,
    E_datain, E_dataout, E_reqin, E_ackout, E_ackin, E_reqout
);
```

Fig. 8. Input/Output Port of DSB router

這一階段的作用為：一、回傳 ack 訊息給外部網絡，二、收集輸入訊號並放到 Input buffer memory，三、對輸入訊號進行解碼得到其輸出方向和對應輸出時間，解碼後的資訊會附加到 data 中繼續處理。

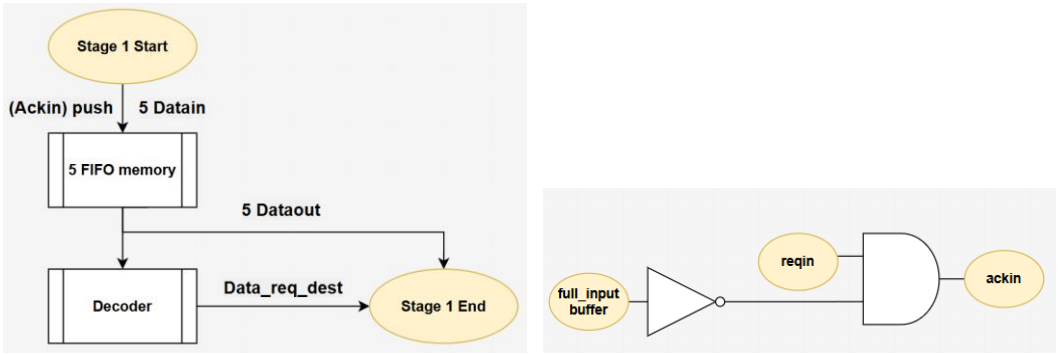


Fig. 9. Stage 1 Flow diagram

2. Second stage: Arbiter + Input buffer output to middle memory by XB1

中間階段為分配輸入訊號到合適的 Middle memory，處理 Arrival Conflict 的問題。仲裁器(Arbiter)決定每個 input buffer 的輸出放到哪一個 MM.，和產生 grant 訊號利用 Switch5 分配到 MM.中。另外使用與參考論文一致的記錄 MM.狀態架構。

```
// Conflict resolution signals
wire [4:0] mm_compatible [0:4];  // mm_compatible[i][j] = 1 if input i is compatible with MM j
wire [4:0] final_mm_grant [0:4]; // final_mm_grant[i][j] = 1 if input i gets MM j
```

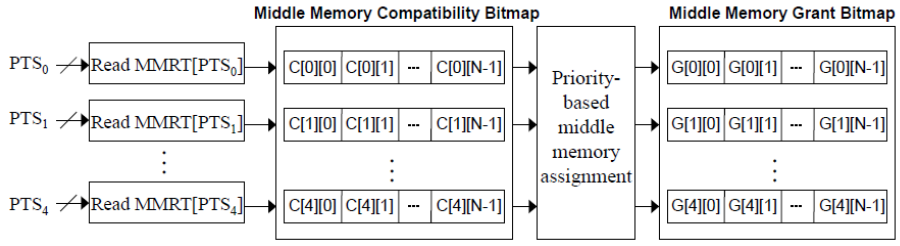


Fig. 10. Middle Memory Bitmaps in DSB router and wire declarations

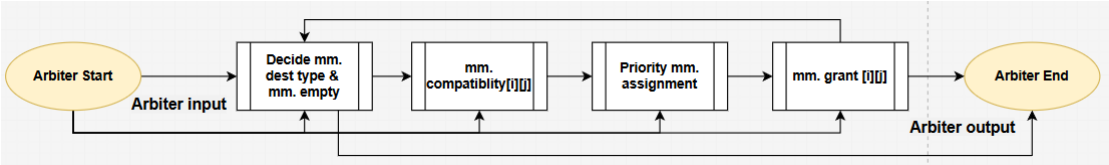
下圖是為 Middle Memory 定義固定優先度的部分(MM0 >MM1> … > MM 4)

```
assign final_mm_grant[0] = gnt_to_input0;

assign final_mm_grant[1][0] = gnt_to_input1[0] & ~final_mm_grant[0][0];
assign final_mm_grant[1][1] = gnt_to_input1[1] & ~final_mm_grant[0][1];
assign final_mm_grant[1][2] = gnt_to_input1[2] & ~final_mm_grant[0][2];
assign final_mm_grant[1][3] = gnt_to_input1[3] & ~final_mm_grant[0][3];
assign final_mm_grant[1][4] = gnt_to_input1[4] & ~final_mm_grant[0][4];

assign final_mm_grant[2][0] = gnt_to_input2[0] & ~final_mm_grant[0][0] & ~final_mm_grant[1][0];
assign final_mm_grant[2][1] = gnt_to_input2[1] & ~final_mm_grant[0][1] & ~final_mm_grant[1][1];
assign final_mm_grant[2][2] = gnt_to_input2[2] & ~final_mm_grant[0][2] & ~final_mm_grant[1][2];
assign final_mm_grant[2][3] = gnt_to_input2[3] & ~final_mm_grant[0][3] & ~final_mm_grant[1][3];
assign final_mm_grant[2][4] = gnt_to_input2[4] & ~final_mm_grant[0][4] & ~final_mm_grant[1][4];
```

● ● ● **Fig. 11. Fix Priority of Middle Memory logic**



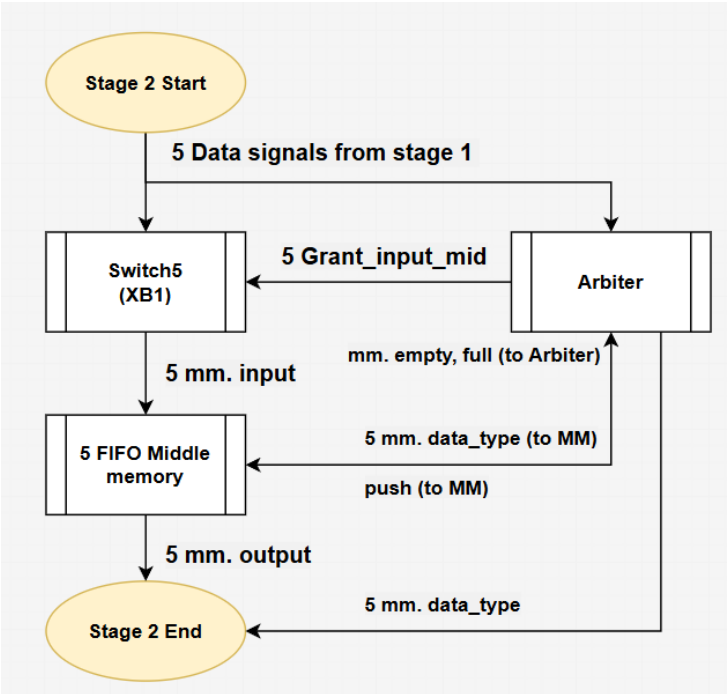


Fig. 12. Stage 2 Flow diagram

3. Third stage: Middle memory to output bus by XB2 + Output link

最後階段將 Middle memory 要輸出的訊號，根據外部網絡決定是否輸出，以及利用相同結構的 Switch5 模塊處理 Departure Conflict 的問題。

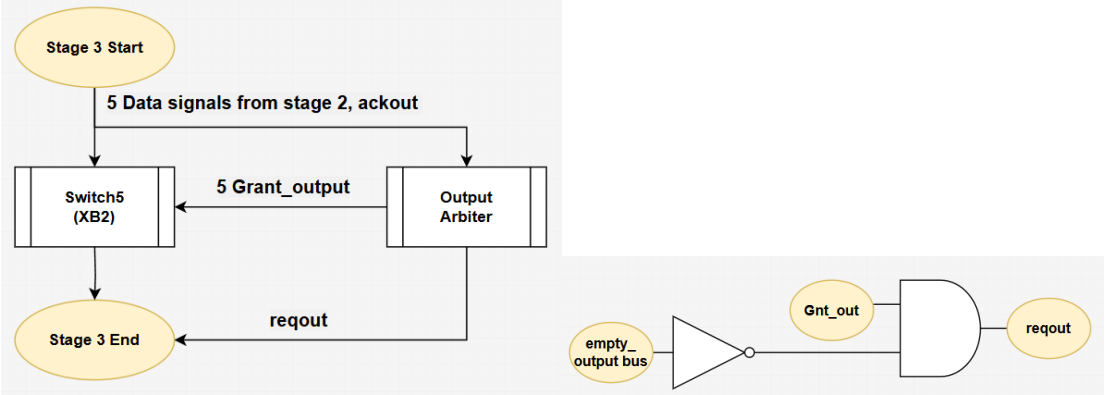


Fig. 13. Stage 3 Flow diagram

2. 實驗結果與分析

實驗設計分成兩個部分：第一部分測試 router 是否能夠傳輸來自不同方向的 flow control unit 到不同的輸出方向(包括 router 本身，定義方向為 P)；

在第一部分的輸入數據處理上，首先利用 C++程序生成所需要的 data 格式，我把要模擬的 data flow 情況先用 txt 文檔記錄如下：

```
1   .flit_bit
2   72
3   .local_place
4   1 1
5   .request
6   1 0 1 0 1 2
7   2 0 1 1 2 1
8   3 4 0 1 1 1
9   4 4 2 1 0 1
10  5 8 0 1 2 1
11  6 8 1 2 1 0
12  7 10 1 1 1 0
13  8 10 0 1 1 1
```

Fig. 14. Data generate txt file

在這個文檔中，每一筆 data unit size 是 72 bit，要模擬的 router 的位置在(x,y)=(1,1)，之後的 request 則由 6 個 int 組成，其格式為[計數，發送時間，發送起點(x,y)，發送終點(x,y)]。之後透過 data_preprocessing.cpp 轉成 5 個方向的 router 輸入文檔，再接着用 Verilog 寫上 pattern 和 testbench 調用.dat 文檔：

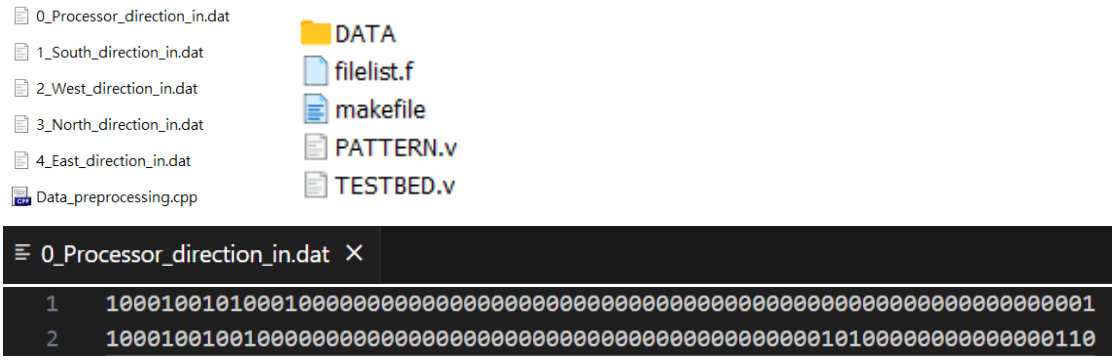


Fig. 15. Data generate preprocessing files

在處理好輸入數據後，我製造了兩個 testcase 來模擬比較傳統 IBR 與我的 DSB router 的單獨 throughput 情況。(Left: IBR，Right: DSB router)

Testcase 1 :

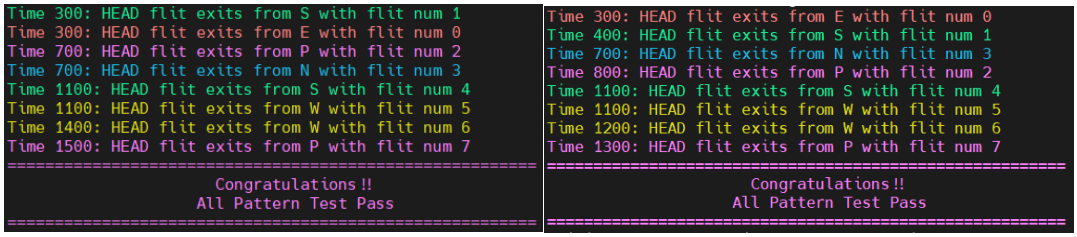


Fig. 16. Part 1 testcase 1

Testcase 2:

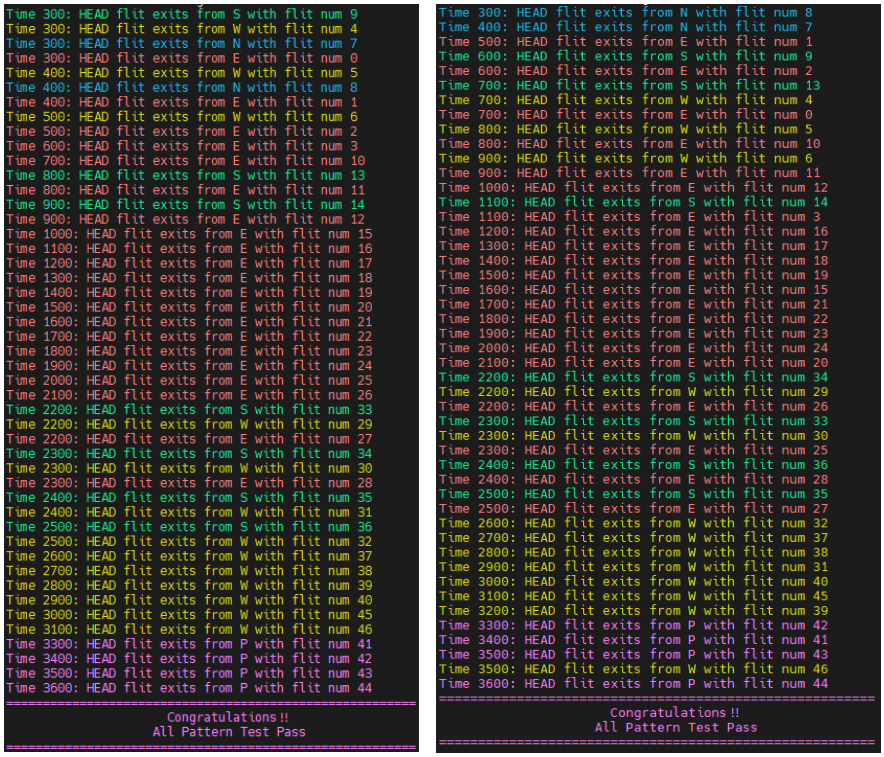


Fig. 17. Part 1 testcase 2

可以看到兩個 Router 都能正常運作，且沒有明顯差距。

第二部分是模擬 router 在 2 * 2 的 2D mesh network 中作訊號的傳遞。輸入數據的處理基本跟第一部分相同，做法為把轉換過後的.dat 文檔對應到 Mesh 0-3 的 P 方向輸入端即可。實驗比較對象為傳統的 3-stage pipeline IBR，IBR 的 FIFO memory 一樣為 8 個 data 的大小。2*2 的 mesh network 表示如下圖：

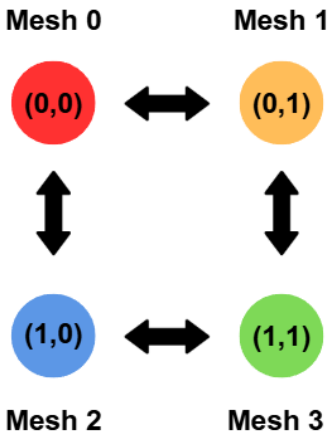


Fig. 18. Simulated 2*2 mesh network

以下是 testcase 測試兩種 Router 組成的 network 的效率。

(First: IBR，Second: DSB router)

Testcase:

```
Time 700: HEAD flit exits from M0 with flit num 3, generate time: 0, delay: 7
Time 700: HEAD flit exits from M1 with flit num 2, generate time: 0, delay: 7
Time 700: HEAD flit exits from M2 with flit num 1, generate time: 0, delay: 7
Time 800: HEAD flit exits from M0 with flit num 7, generate time: 1, delay: 7
Time 800: HEAD flit exits from M1 with flit num 6, generate time: 1, delay: 7
Time 800: HEAD flit exits from M2 with flit num 5, generate time: 1, delay: 7
Time 900: HEAD flit exits from M0 with flit num 11, generate time: 2, delay: 7
Time 900: HEAD flit exits from M1 with flit num 10, generate time: 2, delay: 7
Time 900: HEAD flit exits from M2 with flit num 9, generate time: 2, delay: 7
Time 900: HEAD flit exits from M3 with flit num 0, generate time: 0, delay: 9
Time 1000: HEAD flit exits from M3 with flit num 4, generate time: 1, delay: 9
Time 1100: HEAD flit exits from M3 with flit num 8, generate time: 2, delay: 9
Time 1200: HEAD flit exits from M3 with flit num 12, generate time: 3, delay: 9
Time 1300: HEAD flit exits from M3 with flit num 13, generate time: 4, delay: 9
Time 1400: HEAD flit exits from M3 with flit num 14, generate time: 5, delay: 9
Time 1500: HEAD flit exits from M3 with flit num 15, generate time: 6, delay: 9
==================================================
                Congratulations!!
             All Pattern Test Pass
==================================================
```

```
Time 500: HEAD flit exits from M0 with flit num 3, generate time: 0, delay: 5
Time 500: HEAD flit exits from M1 with flit num 6, generate time: 1, delay: 4
Time 500: HEAD flit exits from M2 with flit num 5, generate time: 1, delay: 4
Time 500: HEAD flit exits from M3 with flit num 0, generate time: 0, delay: 5
Time 600: HEAD flit exits from M0 with flit num 7, generate time: 1, delay: 5
Time 600: HEAD flit exits from M1 with flit num 10, generate time: 2, delay: 4
Time 600: HEAD flit exits from M2 with flit num 9, generate time: 2, delay: 4
Time 600: HEAD flit exits from M3 with flit num 8, generate time: 2, delay: 4
Time 700: HEAD flit exits from M1 with flit num 2, generate time: 0, delay: 7
Time 700: HEAD flit exits from M2 with flit num 1, generate time: 0, delay: 7
Time 700: HEAD flit exits from M3 with flit num 4, generate time: 1, delay: 6
Time 800: HEAD flit exits from M3 with flit num 12, generate time: 3, delay: 5
Time 900: HEAD flit exits from M3 with flit num 13, generate time: 4, delay: 5
Time 1000: HEAD flit exits from M0 with flit num 11, generate time: 2, delay: 8
Time 1000: HEAD flit exits from M3 with flit num 15, generate time: 6, delay: 4
Time 1100: HEAD flit exits from M3 with flit num 14, generate time: 5, delay: 6
============================================================
                    Congratulations !!
                 All Pattern Test Pass
============================================================
```

Fig. 19. Part 2 testcase

可以看到實作的 DSB router 的平均 delay 會比傳統 IBR 來得更低，這是因為當 router 要同時處理每個方向進入的 data 時，DSB router 會有更好的分配方式，而傳統 IBR 中每一筆 data 平均需要做較長的等待，delay 值會差不多。可以說明實作 DSB router 在 Network-on-chip 的應用上會是比傳統 IBR 更好的選擇。

最後是實作的 DSB router 在使用 tsmc 130nm 製程成功合成的數據，clock cycle 為 10ns：

```
Number of ports:                          748
Number of nets:                         20212
Number of cells:                        19741
Number of combinational cells:          13100
Number of sequential cells:              6640
Number of macros/black boxes:               0
Number of buf/inv:                        611
Number of references:                     135

Combinational area:            134359.390894
Buf/Inv area:                    6523.108280
Noncombinational area:         166587.930728
Macro/Black Box area:               0.000000
Net Interconnect area:        2957002.104309

Total cell area:               300947.321622
Total area:                   3257949.425931
```

```
clock clk (rise edge)                                    10.00      10.00
clock network delay (ideal)                               0.50      10.50
clock uncertainty                                        -0.10      10.40
mm4_fifo/buffer_reg[5][3]/CK (DFFQX1)                     0.00      10.40 r
library setup time                                       -0.27      10.13
data required time                                                  10.13
--------------------------------------------------------------------------
data required time                                                  10.13
data arrival time                                                  -10.12
--------------------------------------------------------------------------
slack (MET)                                                         0.01
```

Fig. 20. Synthesis Result