

Mlchip Third Report

(Implement NoC by SystemC)

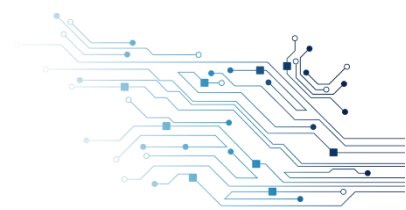
學生：燕新城 學號：311591023

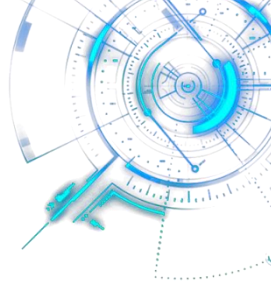
國立陽明交通大學

國際半導體產業學院碩二

E-Mail : samuelyenyen@gmail.com 電話：0966071187

完成日期 (May 16, 2024)





一、Simulation results

成功送出並接收所有 packet。

```
14:17 mlchip081@ee21[~/hw3]$ make
g++ -I. -I /RAID2/COURSE/mlchip/mlchipTA01/systemc-2.3.3/include -L
WL,-rpath,/RAID2/COURSE/mlchip/mlchipTA01/systemc-2.3.3/lib -l linux64
./run

SystemC 2.3.3-Accellera --- Mar  2 2024 23:27:20
Copyright (c) 1996-2018 by all Contributors,
ALL RIGHTS RESERVED

=====
+                                     +
+  Congratulations !!                 +
+                                     +
+  Simulation completed               +
+  at 797 th cycle                   +
+                                     +
=====

      | \_ \_ \_ \
      / 0.0
    /-----\
   / ^ ^ ^ ^ \
  /  |  |  |  \ w
   \ m m m m /
    |-----|

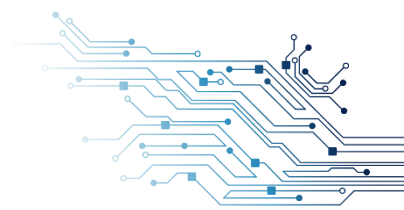
Info: /OSCI/SystemC: Simulation stopped by user.
14:18 mlchip081@ee21[~/hw3]$
```

二、程式細節

1. 整體程式的運作流程為，先從 core 讀第一筆 packet，將 packet 從 core 送往 router，將 router 送往目的地 router，目的地 router 接收後回傳回 core，core 進行 check packet。
2. Router and NI

Router 採用先傳上再傳下再傳左再傳右的優先級順序。

```
if (row_1>row_23) {
    out_req[0].write(false);
    tx_busy_03 = false;
}
else if (row_1<row_23) {
    out_req[1].write(false);
    tx_busy_13 = false;
}
else if (col_1>col_23) {
    out_req[2].write(false);
    tx_busy_23 = false;
}
else if (col_1<col_23) {
    out_req[3].write(false);
    tx_busy_33 = false;
}
else if (col_1==col_23&&row_1==row_23) {
    out_req[4].write(false);
}
```



NI 的設計由發出端先提出要求，有回應後就開始傳資料，接受端要等到 flit 出現起始的 10bit 才會開始接收，到 01bit 時就會停止。

```
if (in_req[4].read() == true){
    out_ack[4].write(true); // No data or not acknowledged, do not transmit
    sc_lv<34> receive_data = in_flit[4].read();
    sc_lv<2> first_two_bits = receive_data.range(33, 32);

    if (first_two_bits == 2) {
        store_begin4 = 1;
        addr_14 = receive_data.range(31, 28);
        addr_24 = receive_data.range(27, 24);
        lv_vector4.push_back(receive_data);
        if (router_ID==13){
            // cout << "Received Data: " << receive_data << endl;
        }
    } else if (first_two_bits == 1 && store_begin4 == 1) {
        store_begin4 = 0;
        flag_from_core = 1;
        lv_vector4.push_back(receive_data);
        if (router_ID==13){
            // cout << "Received Data: " << receive_data << endl;
            // cout << lv_vector4.size() << endl;
        }
        // cout << "RouterID " << router_ID << endl;
    } else if (store_begin4==1){
        lv_vector4.push_back(receive_data);
        if (router_ID==13){
            // cout << "Received Data: " << receive_data << endl;
        }
        // cout << "addr_1 : " << addr_1 << endl;
    }
}

req_tx.write(true); // Indicate a new request to transmit
// Convert packet data back to sc_lv and transmit
// wait(ack_tx.posedge_event());
// cout << "rst event" << endl;
// if (ack_tx.read()) {
//     for (size_t i = 0; i < lv_vector.size(); ++i) {
//         if (counter < lv_vector.size()) {
//             // cout << counter << endl;
//             // cout << lv_vector[counter] << endl;
//             if (ack_tx.read()==true) {
//                 flit_tx.write(lv_vector[counter]);
//                 counter = counter+1;
//             }
//         }
//     }
// } else if (counter==lv_vector.size()){
//     counter = 0;
//     init_done = false;
//     req_tx.write(false);
//     lv_vector.clear();
// }
```

3. Challenge face

當同一個 router 的同一個通道同時要送資料時，需要等前一筆資料送完時，才能開始送下一筆資料，因此在每個資料在傳資料的時間點都需要拉起一個 busy 的訊號，其他資料看到 busy 就需要等待他傳完，再繼續傳送。