VLSI DESIGN LAB. ASSIGNMENT -7

SAURAV GUPTA

ROLL No. – 153070046

Dept. - Electrical Engineering

Specialisation – Microelectronics (TA)

Batch - 2015-17

Teacher concerned – VIRENDRA SINGH

Problem Statement:

Design a router using *synthesizable Verilog* to be used with network-on-chip (NoC) that is organized in an array of 4x4. A router consists of 4 routing ports (*N, S, E, W*) and a local port for the connected core. Keeping it a low cost router, it should use dimension order routing (DOR) so that it does not requires to store routing table. Data is transferred as flit. It uses virtual channels (2 virtual channel per input) to prevent deadlock condition.

Flit Format: The flit format is presented in Table 1 and 2. The 32-bit packet consists of a header flit followed by payload flits. Two additional 2-bit heads are type and (Identity ID) bits. The flit type can be as header (Header flit), message body, and the end of message body or the last flit of the message (Tail flit).

Table 1: Flit format

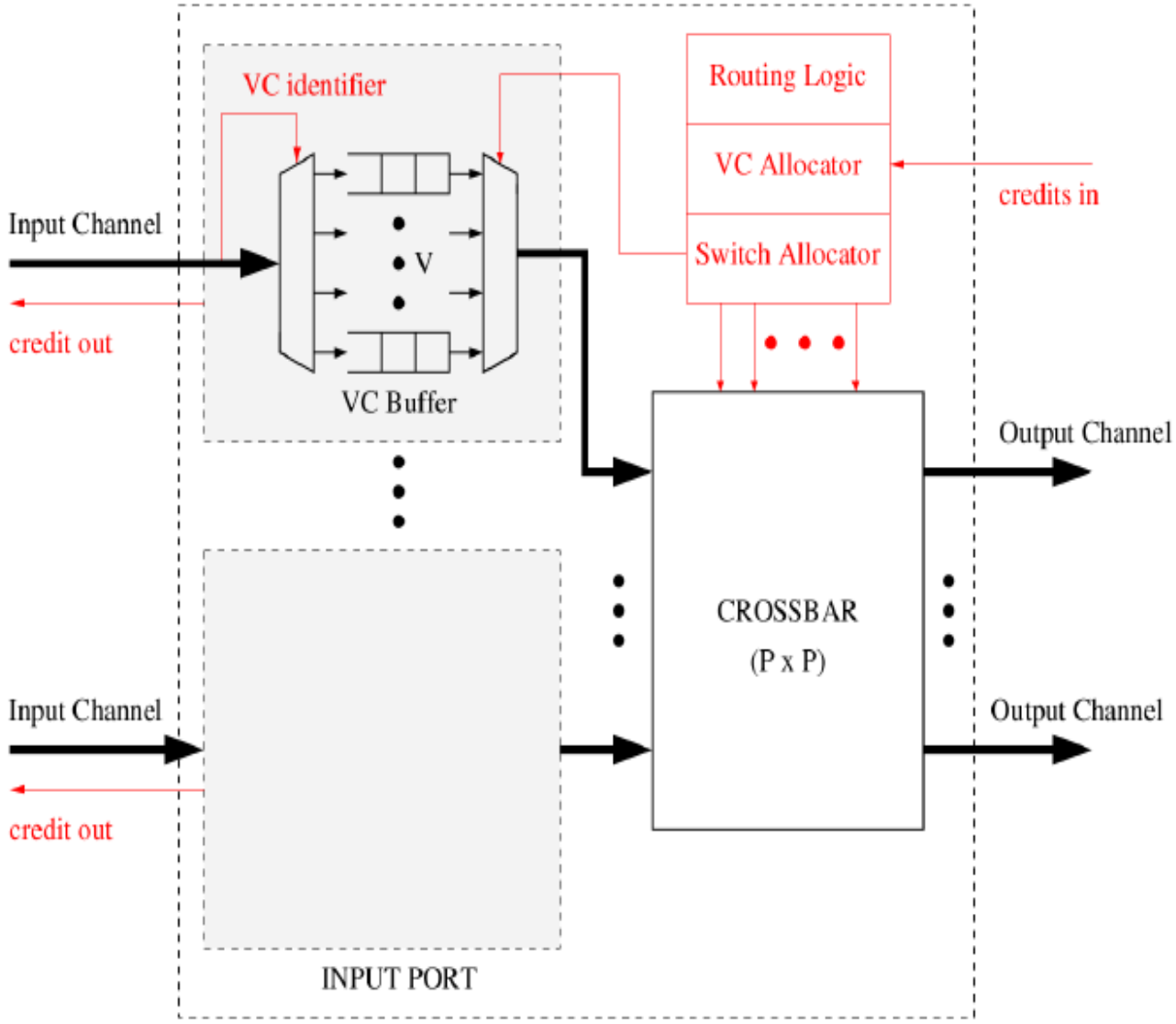
Type header (2 bits)	ID (2 bits)	Source X (2 bits)	Source Y (2 bits)	Dest. X (2 bits)	Dest. Y (2 bits)	Payload (20 bits)
Type header	ID	Payload (28 bits)				
		. . . . .				
Type header	ID	Payload (28 bits)				

Table 2: Flit types

Type of flits	Binary (Decode) code
No flit	00 (0)
Header Flit	01 (1)
Message/Data body	10 (2)
Tail flit	11 (3)

In order to control the flow every router sends a credit (availability of one flit space or not). Based on the credit it transfers the flit to the neighbor.

ROUTER ARCHITECTURE



Sol:

Components:

1. **Input port –** five – Local, North, South, East, West
2. **Routing Logic –** DOR-XY
3. **Switch Allocator –** Credit Based flow control
4. **Crossbar Switch –** 5X5 crossbar switch

1. Input Port

Each input port contains FIFO Virtual Channel Buffer of 4-word length, where each word is of 32-bits.
Input flit comes at every positive edge of the clock and output flit leaves at positive edge of the clock. However, the decisions are made at negative edge of the clock to avoid any ambiguity in data.

Note that there is a data_request signal input coming from switch allocator which indicates that output has been taken. Whenever this signal comes VC Buffer is right shifted by 32 bits/1 word and Credit_out signal is made high indicating that one word of the buffer has been cleared and there is more room for new data.

The Block diagram of Input Port is shown in fig. 1.1

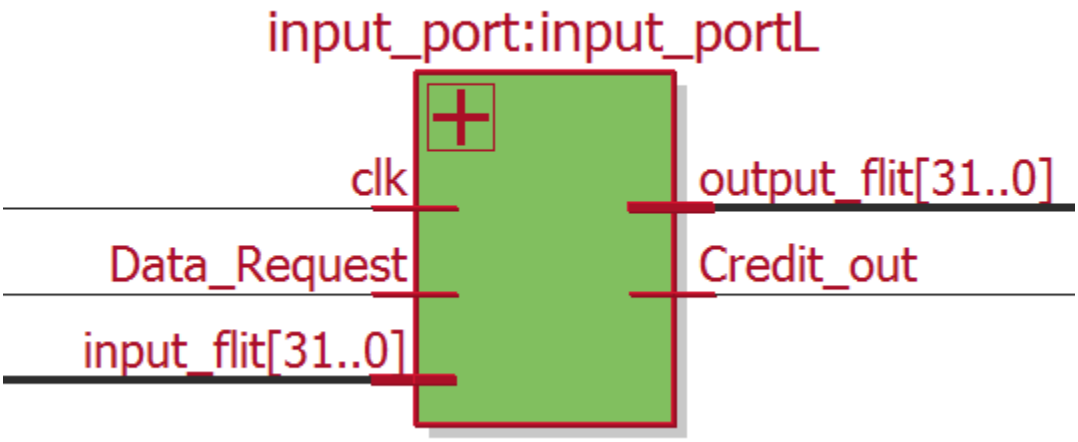


Fig. 1.1 Block Diagram of Input Port

2. Routing Logic

Routing Logic has 5 FSM for each input port.
FSM of Local input port is shown in fig.2.1

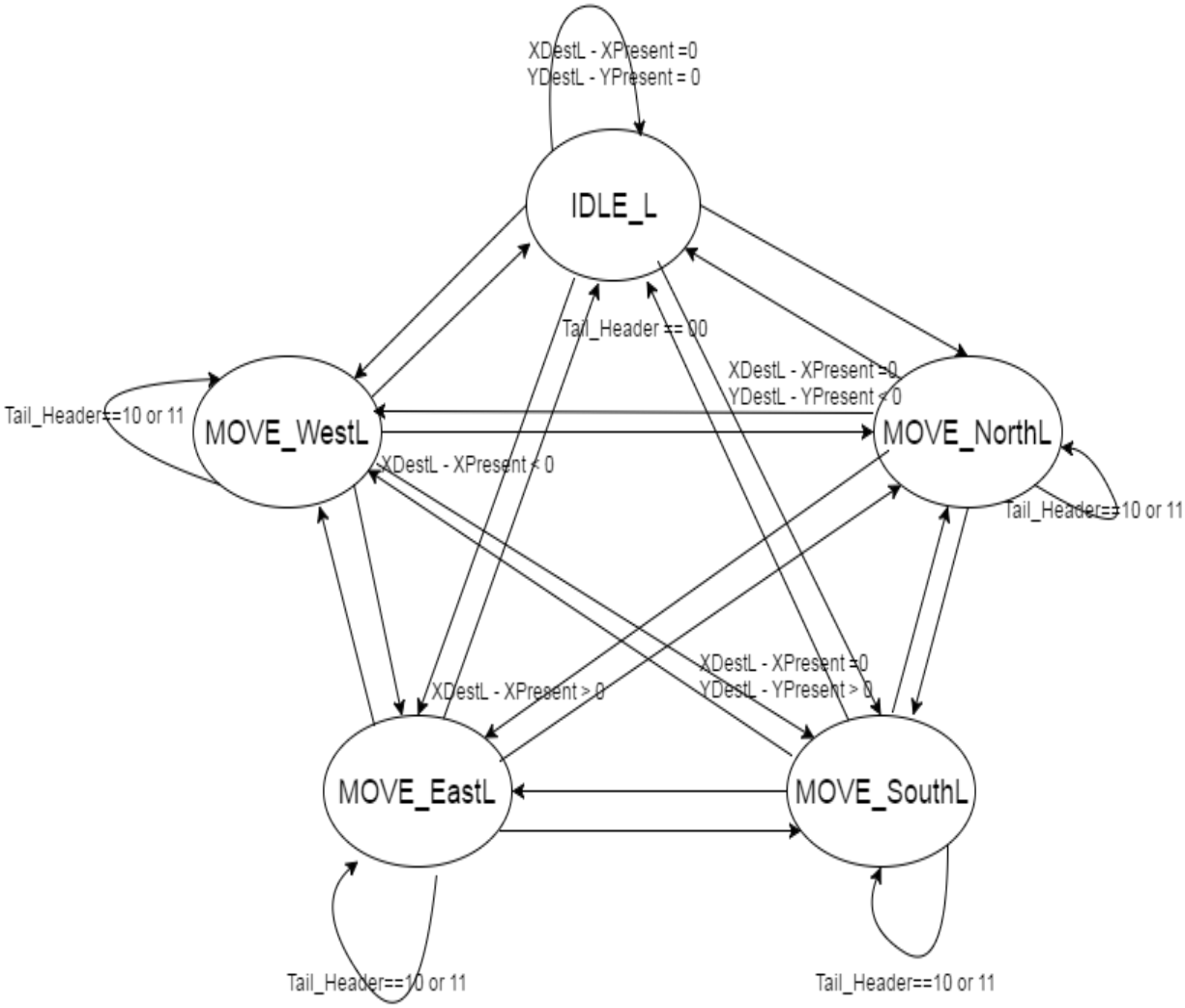


Fig. 2.1 FSM of Routing Logic for Local input port

FSM for North, South, East and West input port is similar to the above FSM. Here we have used Dimension order routing (DOR) , XY Routing to be precise.

Based on the XY coordinates of the Destination and the present address we determine where the data should be forwarded.

The Block diagram of Routing logic is given in fig. 2.2

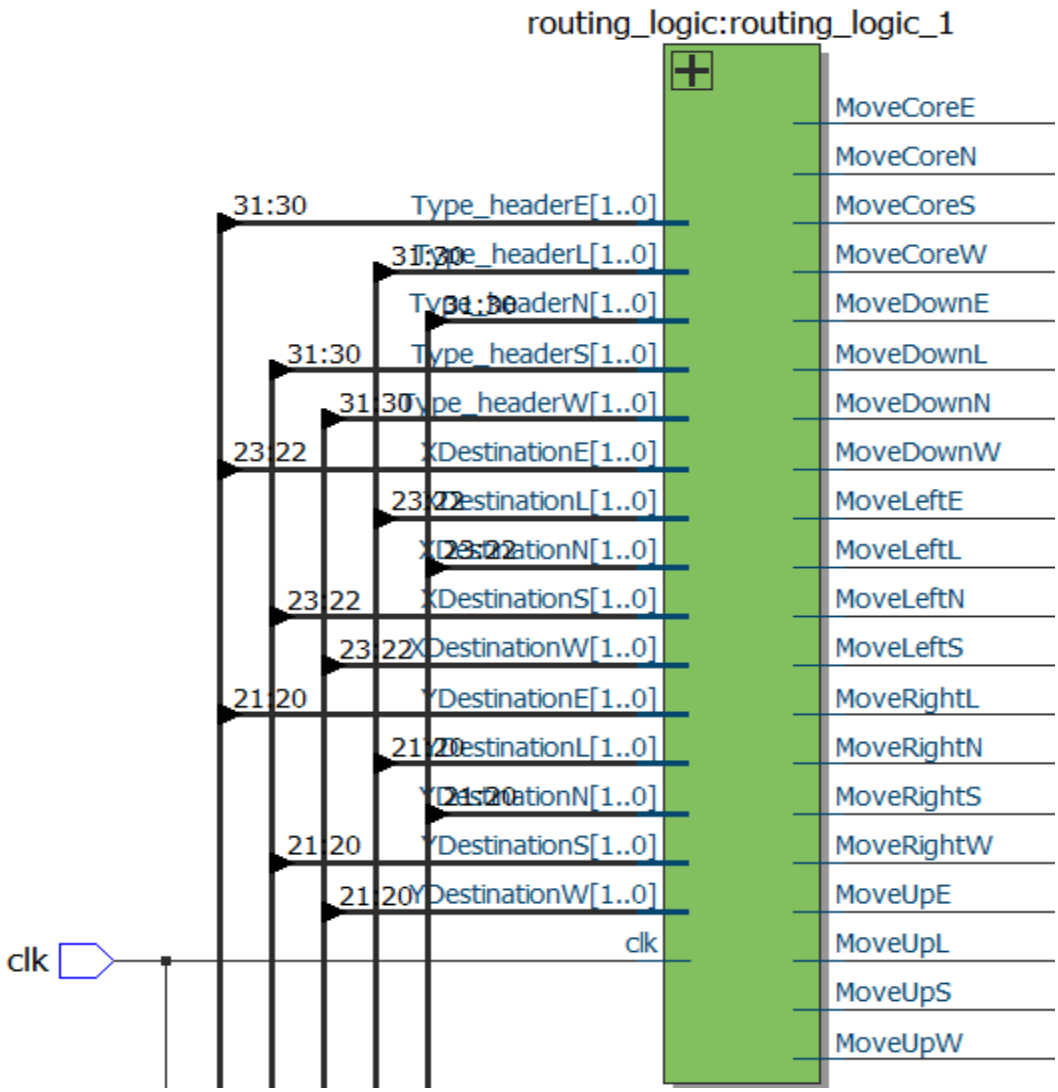


Fig. 2.2 Block Diagram of Routing logic

3. Switch Allocator

When there are input data streams from various input port for the same output port, there is a collision. To avoid collision Arbiter or switching allocator is used. It arbitrates between various inputs. There are various Arbitration techniques of which Round Robin is one of them. Here we have used Round Robin arbitration between the input ports at which data is available.

Credit based Flow control is implemented here, In this scheme Upstream router stores the count of each downstream router. It decrements the

credit count when flit is forwarded and increments the counter when the buffer of downstream router is freed in which case it receives a credit in signal.

Note: When count =0 the buffer is full hence the data is not forwarded.

There are total 5 FSMs implemented for each output port. The FSM for Local output port is shown in fig. 3.1.

The block diagram for Switching Allocator is shown in fig. 3.2

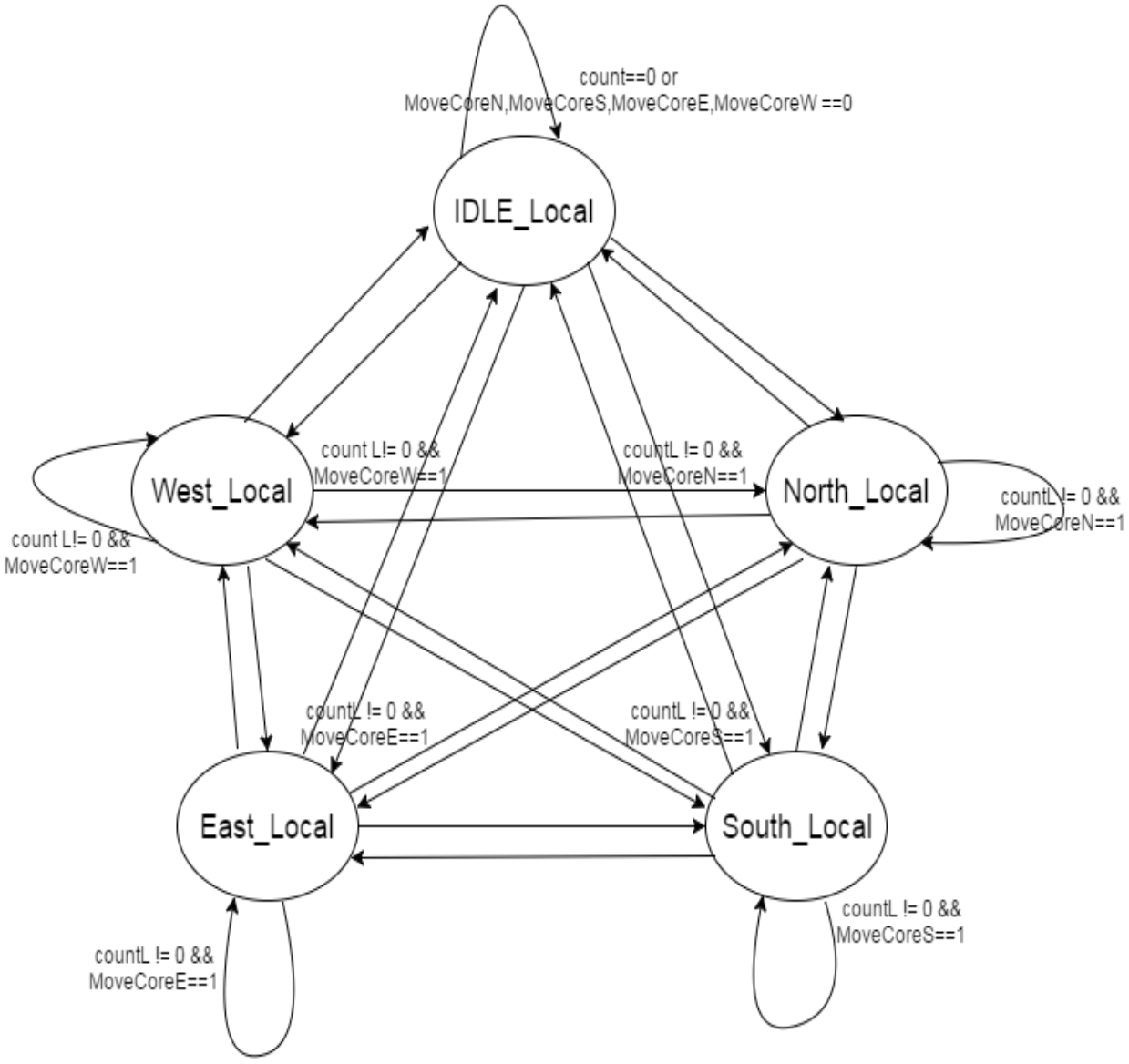


Fig. 3.1 FSM of Switching Allocator for Local output port

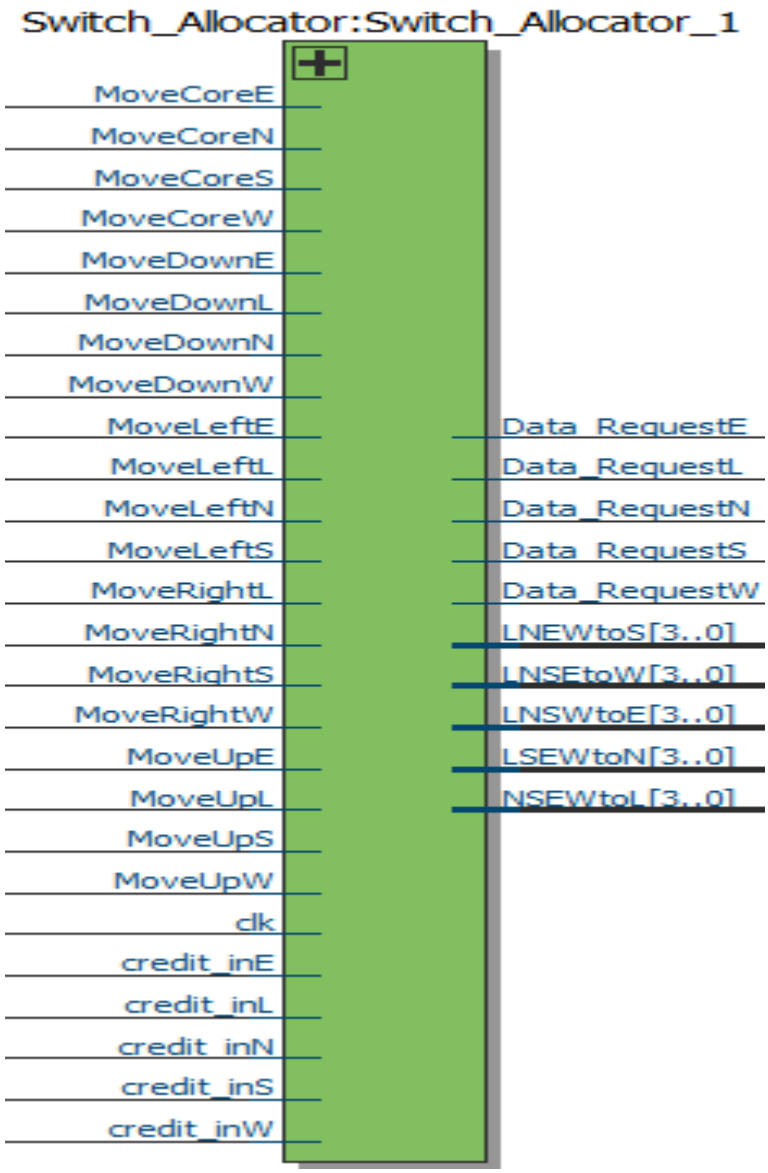


Fig. 3.2 Block Diagram of Switching Allocator

4. Crossbar Switch

The implementation of crossbar switch is pretty straight forward. It receives a total of 20 input pins from the Switching Allocator. Local_to_North, Local_to_South… and all the possible combination.
Based on this input signal it forwards the data from input to output port.

The Block diagram of crossbar switch is shown in fig. 4.1

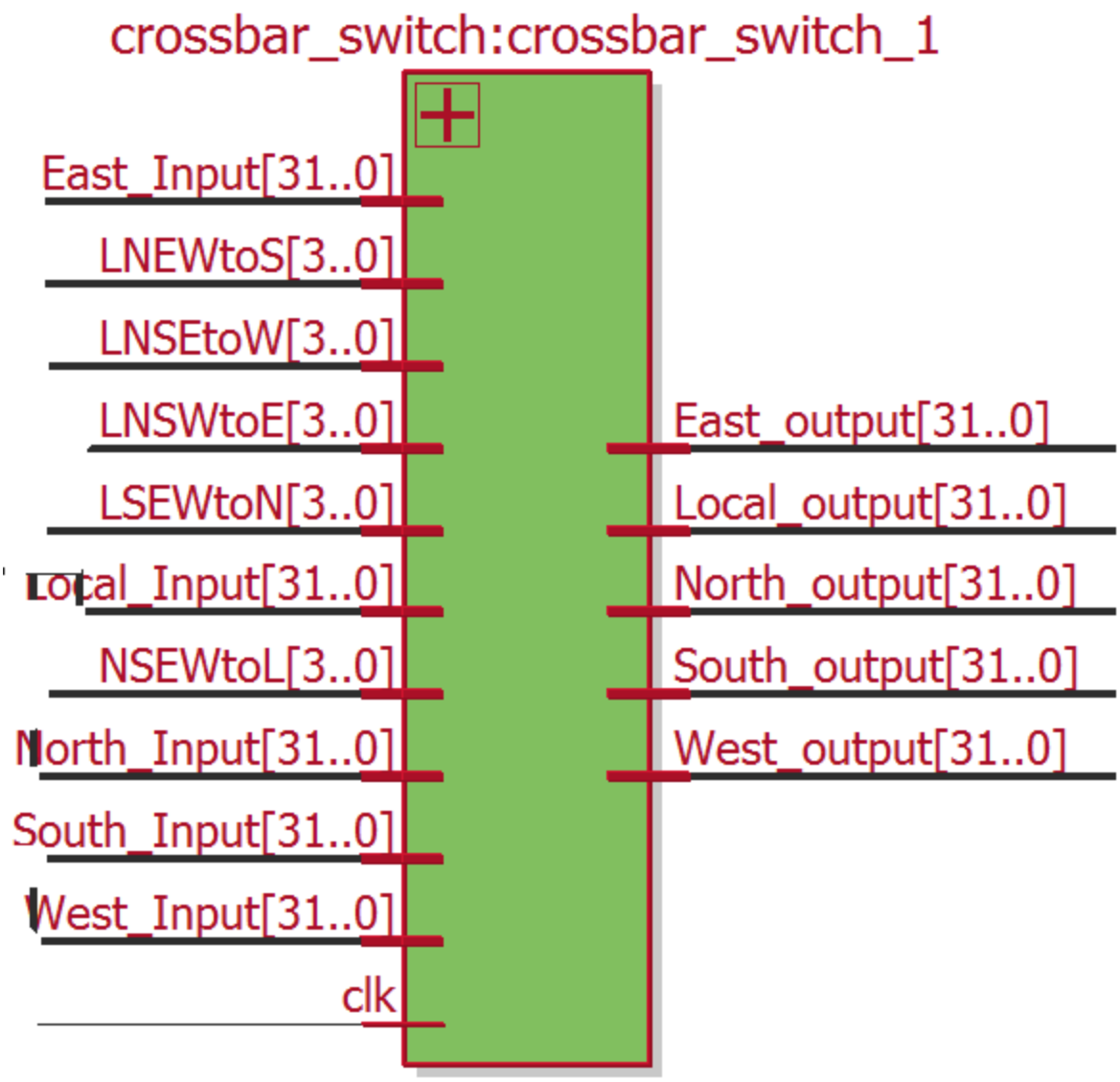
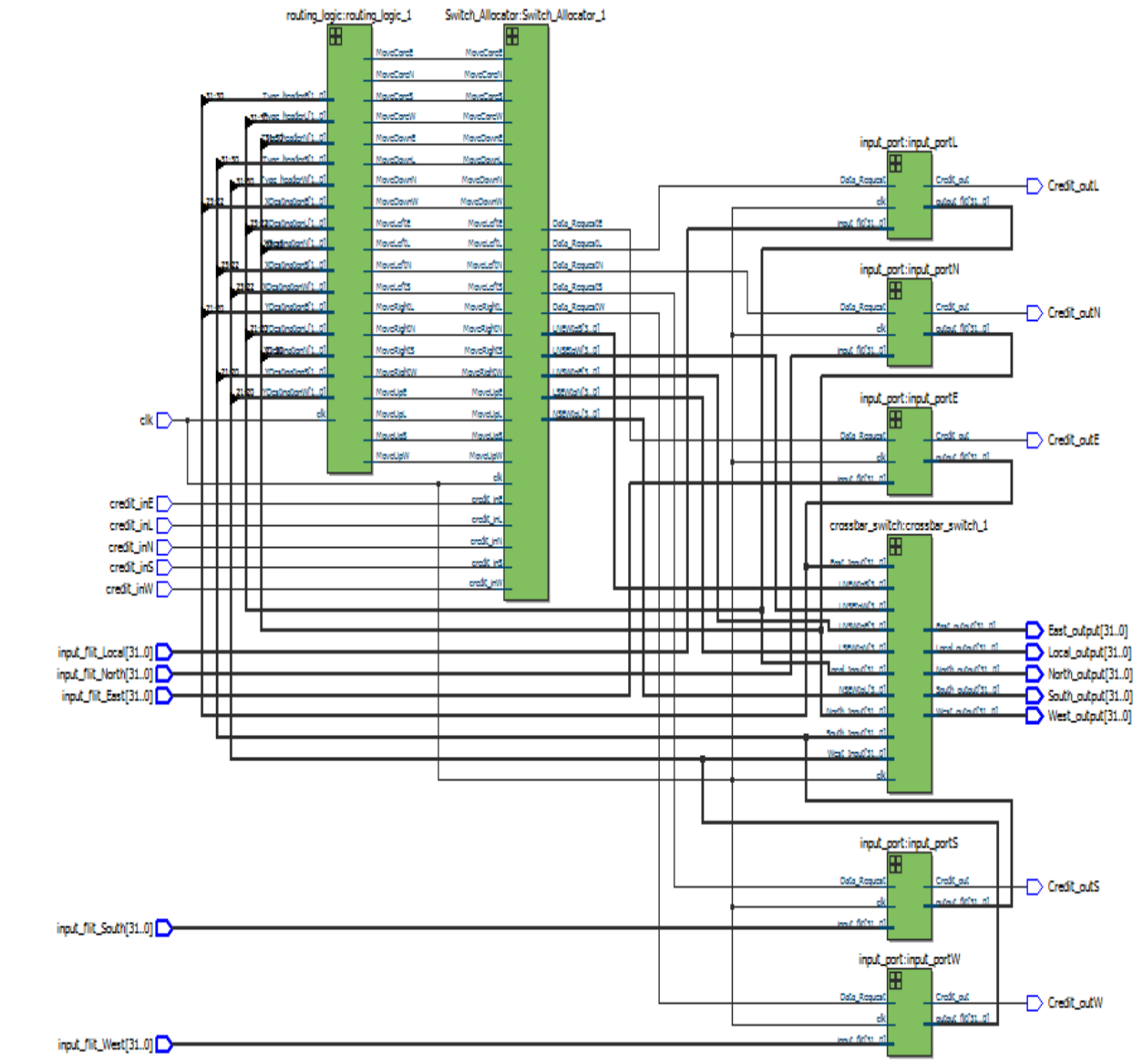


Fig. 4.1 Block Diagram of Crossbar Switch

Overall Circuit Diagram:


routing_logic:routing_logic_1
Switch_Allocator:Switch_Allocator_1
input_port:input_portL
input_port:input_portN
input_port:input_portE
crossbar_switch:crossbar_switch_1
input_port:input_portS
input_port:input_portW
clk
credit_inE
credit_inL
credit_inN
credit_inS
credit_inW
input_flit_Local[31..0]
input_flit_North[31..0]
input_flit_East[31..0]
input_flit_South[31..0]
input_flit_West[31..0]
Credit_outL
Credit_outN
Credit_outE
Credit_outS
Credit_outW
East_output[31..0]
Local_output[31..0]
North_output[31..0]
South_output[31..0]
West_output[31..0]