The Multiplexers-E Project 5- Classify

Diwan Anuj Jitendra 170070005 Soumya Chatterjee 170070010
Vabilisetti Mohan Abhyas 170260032 Arnab Jana 170100082

1 Input Output Contracts

- The valid_in bit during input from a port is continuously 1 externally, as long as the packet does not end.
- When the oData_rd becomes 1, the packet at the head of the FIFO (if any) is output over as many clock cycles as required by the length of the packet. Each packet segment given as input and output of 144 bits.
- The maximum number of hops is assumed to be 60, so path-vector has been allocated a maximum size of 480 bits.

2 Implementation

- A **32-port round-robin arbiter** chooses one of the available input ports and reads data from chosen port to registers. Arbiter is released only after we return to idle after finishing a packet. We use signal "valid" which says the duration during which input data is valid and can be read.
- We used a **FIFO** with din and dout of size 145 bits (including valid bits) and read and write clocks synchronous with our input clock.
- We have 2 state machines, where the second state machine is a sub part of one of the state of main state machine. These states are explained in greater detail in the longer report.
- We stay in **idle** state if input is unavailable. Once input is available, it enters the **iav** state and asks the arbiter to give a grant. After one clock cycle through **oav** state, it uses the received grant to enter the **reading** state. In this state the entire packet is read and processed.
- It stays in the **rinter** state as long as valid is 0 even after grant. When valid is high, next state is r0,r1,r2,r3,r4,rcont. These successively extract the ethertype,value,opcode,no. of hops, hop pointer, MAC addr and path vector.
- After processing the header, if valid path-vector present, we **increment hop-pointer** and **mark opcode as forward** in the packet. If target MAC matches the packet is accepted else it is dropped.
- At the **rcont** stage, packet0 is being pushed in FIFO and remains in this state as long as valid_in is high. If valid become 0, next state is r0out, and from here the remaining 4 buffered packets are put into the FIFO.
- A parallel process outputs the data after oData_rd is made 1. We use valid bit for each chunk, and use an invalid bit to denote a separator in the FIFO to distinguish between chunks of two different packets.
- We maintain **counters for each of 32 ports**, which are reset to count a tick of 1sec, once the first inter-switch packet arrives and set the port as core-port.
- For subsequent packets, if the timer times out, we mark the port as edge port and if we receive an inter-switch packet, we mark it as core-port and reset the timer.
- FSM and timing diagrams are in the longer report.