1.0 Overview

This application note will give an overview of how packet traffic flows through an LS based switch. It will show how unicast and multicast traffic flows between input and output ports and how the MAC Address mapping logic affects the dataflow. This document assumes that the reader is familiar with Ethernet, the IEEE 802.1d bridging specification and LAN switching technology.

2.0 Introduction

The LS architecture is based on a distributed queuing structure with a central crossbar switching fabric. Each LS100 port controller contains four MAC interfaces, input and output queue management, and switch fabric interface logic. The LS101 cross bar switching fabric includes a non-blocking crossbar switching matrix, arbitration logic and link interface logic, used to communicate with the port controllers. Figure 1 shows the dataflow architecture of an LS based switch.

A key innovation in the LS architecture is the way in which multicast traffic is handled. From the MAC input blocks to the switch fabric unicast and multicast traffic follow the same datapath. After leaving the switching fabric unicast and multicast traffic are forwarded over separate datapaths. Unicast packets are forwarded to their destination port via dedicated point to point datalinks that connect the switch fabric output ports to one of the four port interfaces on the port controller. Multicast traffic on the other hand, is switched to a special multicast output port on the switch fabric. This port is connected to dedicated multicast input ports on each port controller in the system. The unicast and multicast traffic streams are combined just before they exit through the MAC blocks. This approach decouples the scheduling of multicast traffic from that of unicast traffic. The benefit is higher aggregate switch bandwidth and lower latency.

The remainder of this document describes the behavior of the individual blocks that make up Figure 1. The circled numbers next to each block reference the section of this document that describes it.
Figure 1: LS Packet Dataflow

Confidential
3.0 MAC Input processing

The MAC input block processes the data stream that enters the LS100 from the physical layer. Functions this block performs include:

- Packet delineation
- CRC checking
- Packet Classification
- Packet Filtering

While the first two functions are the same as those performed by standard Ethernet MAC controllers the packet decoding functions are switch specific.

3.1 Packet Classification

The packet decode functions check the headers of incoming packets for several characteristics. The results of the decode function are stored in a switch header that is prepended to the packet by the MAC input block. This information is used in several locations as packets move through the switch.

3.1.1 Unicast/Multicast/Broadcast Decode

The destination address field of each incoming packet is checked to determine if it is a unicast or multicast (which includes broadcast) address. This is done by examining the first bit of the address, if it is ‘1’ the packet is flagged as a multicast, if it is ‘0’ the packet is flagged as a unicast. If all bits in the destination address are set to ‘1’ the packet is flagged as a broadcast.

3.1.2 802.1d Reserved Multicast Address Decode

The destination address field of each incoming packet is also checked to determine if it is one of the seventeen address values reserved by the IEEE 802.1d bridge specification. Two of these values are defined as Bridge Protocol Data Units (BPDUs) with the other fifteen being reserved. The LS100 treats all seventeen values as BPDUs and for the remainder of this document we will refer to all packets that contain any of these seventeen addresses as (somewhat inaccurately) BPDUs.

3.1.3 Typefield/Length Decode

The typefield/length field of each incoming packet is examined to determine if the field contains typefield or length information. If the value of typefield/length field is greater than 1536, the packet is flagged as typefield encoded, otherwise it is flagged as length encoded.

3.2 Packet Filtering

Packet filtering is performed by the MAC input block for either of two functions. The first is to implement 802.1d port states. The second is size filtering based on packet length.
3.2.1 802.1d Port State Filtering

The information decoded by the packet classification logic is used by both the MAC input and output blocks to filter packets in a manner compliant with the port states defined in the 802.1D specification. The filtering behaviors supported by the MAC input block are:

- Pass received non-BPDU frames
- Pass received BPDU frames

These filtering behaviors can be independently configured on a per-port basis. Table 1 shows how these filtering behaviors correspond to 802.1d port states.

<table>
<thead>
<tr>
<th>802.1D Port State</th>
<th>Pass Received Non-BPDU Frames</th>
<th>Pass Received BPDU Frames</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disabled</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Blocking</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Listening</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Forwarding</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Table 1: LS Packet Dataflow

3.2.2 Overlength and Underlength Packet Filtering

The MAC input block truncates all packets that exceed the space available in packet buffers. In the LS100 this is 2032 Bytes. Packets that exceed the IEEE 802.3 limit of 1518 bytes are not discarded.

The MAC input block can also be programmed to filter well formed packets that do not meet the IEEE 802.3 minimum length requirement of 64 bytes.

4.0 Input ATC Mapping

The ATC (Address Translation Cache) examines all traffic coming from the MAC interface in order to:

- Map the packet’s destination address to a destination port number.
- Check to see if the source address is known.

The ATC mapping operation occurs while the incoming packet is being stored in its input buffer. The address mapping system, which includes the primary cache, the secondary cache (if present) and the address mapping database in the CPU’s memory space, attempts to resolve these address queries.

Addresses must be resolved before the end of the packet arrives at the port. The handling of packets whose addresses cannot be resolved within this time frame is independently programmable for both source and destination address cases. The handling of packets in these cases is referred to as Default Mapping Behavior. The address mapping system may fail to resolve an address in time if an address mapping does not exist or if misses have overwhelmed the address mapping system.
Details of the address mapping system, including the interaction of the primary and optional secondary caches with the address mapping database can be found in the application note MAC Address Translation in the LS Architecture. Figure 2 gives an overview of the ATC’s input mapping logic.

![Figure 2: ATC Input Mapping Logic](image)

### 4.1 Destination Address Mapping

Destination address mapping resolves a MAC address to a destination port number. The resulting port number and other control information is added to the switch packet header. The packet is then ready to be sent to the destination port and is flagged as ready.

Destination address mapping is handled differently for unicast packets, multicast packets and 802.1d reserved multicast packets (BPDUs).

#### 4.1.1 Unicast Packet Handling

Unicast destination addresses map to a single, physical port ID. This address is derived from a hit on any one of the entries located in the ATC. The four usage bits that are included in each ATC entry can be used to qualify whether each of the four port controllers are allowed to use an address mapping located in the cache.

The default mapping behavior for unicast destination addresses can be programmed, on a per-port basis, to any of the following behaviors:

- Flood the packet, the address is mapped to the multicast port ID. This is the traditional handling of such cases. Unfortunately this also compromises security by giving stations access to traffic that was not intended for them.
- Forward the packet to the uplink port, the address is mapped to the uplink port ID. The intention of this is to allow a routing agent to examine the packet.
- Discard the packet.
4.1.2 Multicast Packet Handling

Multicast and broadcast addresses are always mapped to the multicast port ID (port 254). Mapping the address to this port ID will result in the packet being switched to the multicast output port on the LS101. This port is connected to the multicast input port on all port controllers in the switch.

4.1.3 802.1d Reserved Multicast (BPDU) Packet Handling

The LS100 can be configured to map BPDU addresses in one of two ways. The method used is determined by whether the BPDU arrived from an external switch port or a port connected to the 802.1d management entity.

If the BPDU arrived from an external switch port then the address is treated as a unicast and mapped to the management port ID. In this way the packet is forwarded to the 802.1d management entity.

If the BPDU arrived from a port connected to the management entity, i.e. this is the management port, the address is mapped to the multicast port ID (port 254).

4.2 Source Address Mapping

Source addresses are checked to see if they are mapped to a destination port in the ATC. If a mapping is not found by the address mapping system it is forwarded to system software to determine if it should be added to the address mapping database, i.e., learned. Note that any address mapping in the system can be used for either source or destination address mapping.

When a source address mapping operation finds a matching entry in the ATC the port controller can also perform an optional check to determine if the port that the address maps to is the same as the port ID that the packet came in on. This source port checking feature can be used to accelerate the detection of a station move or as a security feature.

5.0 Input Queuing

Each port in an LS100 port controller has a separate input queue. The purpose of the input queues is to buffer traffic until it can be forwarded through the crossbar to an output port. The input queues, like all queues in the LS architecture, are implemented as linked lists of fixed size packet buffers. In spite of the underlying linked list structure; the input queues in the LS100 are logically structured as FIFOs. Input queue management is handled by two processes. The queue filling process and queue forwarding process.

5.1 Input Queue Filling Process

The input queue filling process performs two functions: It allocates packet buffers for packets arriving from MAC and marks them as being ready for forwarding when the forwarding policy conditions are met. The queue fill-
ing process sets a bit in the packet’s switching header to indicate when a packet is forwardable. This information is used by the queue forwarding process to determine if a packet is ready for forwarding.

![Queue Forwarding Policy Diagram](image)

**Figure 3: Queue Forwarding Policy**

The only time that the queue forwarding process may encounter a packet not marked forwardable is when there is only one packet in the queue.

### 5.2 Input Queue Forwarding Process

The queue forwarding process examines the destination port ID of the packet at the head of the queue, attempts to establish a connection with that port through the crossbar, and transfers the packet to the destination port when the connection is made.

The forwarding process will not be able to establish a connection if another port is forwarding a packet to that port or arbitration for that port has been stalled due to output queue congestion. The forwarding process will not be able to clear the packet from the head of the queue until the output port is available for forwarding. This situation is commonly referred to as head of queue or head of line blocking and is an issue with simple FIFO queuing structures.
A Day in the Life of an LS Packet

The LS100 implements a simple but effective scheme to limit the impact of head of line blocking. It is accomplished by performing a one level look ahead into the queue when it occurs. The overall forwarding algorithm with look ahead is as follows:

1) When a packet reaches the head of the queue the forwarding process attempts to establish a connection through the switch to the destination port. The connection request is made to the LS101 crossbar at normal priority. If the connection is successful the packet is forwarded.

2) If the connection request made in ‘1’ above was denied then the queue forwarding process looks ahead to the next packet in the queue. If that packet is destined for a port that is different from the destination port of the head of line packet, the forwarding process attempts to establish a connection through the switch to the destination port of the next packet. The connection request is made to the LS101 crossbar at normal priority. If the connection is successful the packet is forwarded.

3) If the connection request made in ‘2’ above was denied then the forwarding process returns to the head of line packet and again attempts to establish a connection through the switch to the destination port. This time the connection request is made at a higher priority and the forwarding process waits for the connection to be established, and when it is, the packet is forwarded.

6.0 Switch Link TX Port

The Switch Link TX port is the transmission and control gate into the switching fabric. Each port in the switch has its own independent connection to the crossbar switching fabric that performs all data and signaling communication between the quad port LS100s and the LS101 crossbar switch. In-band signaling is used to send connection requests to the LS101 crossbar. The LS101 answers the connection requests on a global response bus where each LS100 listens for its own answers. The following sections cover this interface in more details.

To save device pins and to maximize the number of possible connections in-band signaling is used to send commands to the LS101 crossbar switch. Each octet of a packet is encoded using a 4b/5b encoding scheme that is compatible with the ISO 9314 FDDI specification. This encoding is performed in the quad port LS100’s Switch Link TX port prior to transmitting the packet to the LS101 crossbar.

6.1 Signaling Overview

The 4b/5b scheme expands the code space and produces 16 symbols more than are required for coding the input data. The FDDI specification defines these extra symbols as either control symbols or violations. Violation symbols are not used in FDDI since they can result in failure of the clock recovery mechanisms used.

The LS architecture uses a subset of the control symbols defined by the FDDI specification for demarcating the beginning and end of data packets.
In addition it uses two of the violation symbols to prepend or flag signaling packets. The LS100 uses these signaling packets to make connection requests to the LS101 crossbar. Whenever the LS101 sees a signaling packet it passes the port connection request to its arbitration logic. Depending upon the signaling packet sent, the LS101 crossbar will appropriately respond back to the quad port LS100s by using the global response bus. In this way, each port knows the result of its connection request. The LS100 supports two different kinds of connection requests at four priority levels. The first kind of connection request is CONC - connect and notify on completion. The second kind is CONI - connect and notify result immediately. These two types of connection request are used by the input queue forwarding process to implement the look ahead scheme described in section 5.

6.2 Priority Usage

The LS101 crossbar supports four levels of priority on each connection request. These levels are numbered 0 through 3 with 3 being the highest priority level. The LS100 uses two of these levels to implement the queue look ahead scheme. Levels 0 and 2 may used for normal priority signaling and levels 1 and 3 may used for high priority. Which set of levels is used can be selected on a per-port basis.

6.3 Packet Forwarding

All packets are forwarded through the switch at a 100 Mb data rate with one exception. Cut-through forwarding from a 10 Mb port to a 10 Mb port. In such cases the packets are transferred through the switch by sending each data symbol ten times. This results in an effective 10 Mb transfer rate. The following table summarizes how the source and destination port data rates effect cut through forwarding support and the effective transfer rate through the switch fabric.

<table>
<thead>
<tr>
<th>Source Port Speed</th>
<th>Destination Port Speed</th>
<th>Cut Through Support</th>
<th>Transfer Rate Through Crossbar</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>10</td>
<td>Y</td>
<td>10</td>
</tr>
<tr>
<td>10</td>
<td>100</td>
<td>N</td>
<td>100</td>
</tr>
<tr>
<td>100</td>
<td>10</td>
<td>Y</td>
<td>100</td>
</tr>
<tr>
<td>100</td>
<td>100</td>
<td>Y</td>
<td>100</td>
</tr>
</tbody>
</table>

Table 2: Packet Forwarding Modes

7.0 Switch Link Receiver

The switch link receivers accept the 4b/5b data stream from the switching fabric and recover packets from it. It order to accomplish this they perform several functions:

Symbol Delineation Determine where the symbol boundaries are within the bit stream.
A Day in the Life of an LS Packet

Packet Delineation
Determine where incoming packets start and where they end.

Signaling Packet Discard
Removing switch signaling packets from the data stream.

5b to 4b Decoding
Convert the data stream into its original format.

After performing these functions the port Switch Link Receivers transfer their data to the unicast and multicast output queues.

8.0 Output ATC Mapping

The ATC (Address Translation Cache) examines all traffic entering the multicast output queue in order to filter outgoing traffic. This traffic may be either multicast/broadcast packets or unicast packets that have been flooded. Filtering is accomplished by mapping either the destination address or the type field to a four-bit port vector that determines which of the four ports in the port controller may transmit the packet. The port vector is stored in the multicast output queue in the switch header for each packet.

Multicast filtering is typically used to prevent the flow of traffic between VLAN domains. Details of the use of this feature are contained in the application note “Virtual LAN and Security Support in the LS Architecture”.

The ATC mapping operation occurs while the incoming packet is being stored in the multicast output queue. The address mapping system, which includes the primary cache, the secondary cache (if present) and the address mapping database in the CPU’s memory space, attempts to resolve these address queries. Figure 4 gives an overview of the ATC’s Output mapping logic.

Figure 4: ATC Output Mapping Logic

Confidential
8.1 Multicast Packet Filtering

When enabled, multicast and broadcast packet information is mapped to port vectors by the ATC. The port vector information is contained in the USAGE bit field in the ATC entries. What information is used for the ATC lookup is determined by whether the packet uses the typefield or length format. A flag contained in the switch header, set by the MAC in block (see section 3.1.3), is used to determine the format.

If the packet uses the length format the entire 48 bit destination address is loaded. This mode of operation supports filtering based only on the multicast address and is compatible with both the IEEE 802.3 and DIX Ethernet frame formats. Figure 5 shows the ATC address field format when using this mode.

![Figure 5: Multicast Destination Filtering Mode Address Field Format](image)

If the packet uses the typefield format the sixteen bit typefield is loaded from octets 21 and 22 of the Ethernet frame and the eight source port ID from the switch attached header are loaded into the capture register. The I/G and U/L bits are copied from the destination address field, while bits 2 to 23 are set to zero. This mode of operation supports protocol based filtering, but is compatible only with the DIX Ethernet frame format. Figure 6 shows the ATC address field format when using this mode.

![Figure 6: Multicast Typefield Filtering Mode Address Field Format](image)

8.2 Unicast Packet Filtering

Flooded unicast traffic is identified by way of control information contained in the packet header. Mapping of unicast packets to the port vector does not result in an ATC access and is forwarded to all ports.

9.0 Output Queues

Each port in an LS100 port controller has a separate output queue. In addition each port controller has a single multicast output queue. The purpose of these queues is twofold:
Rate Adaptation

When traffic is delivered to 10 Mb ports it arrives at a 100 Mb data rate. The only exception to this is cut-through traffic from a 10 Mb port to a 10 Mb port (see Table 2 Packet Forwarding Modes). The output queues decouple the port transmission rate from the fabric data rate.

Collision Handling

If the destination port is using CSMA/CD, there is always the possibility that a collision on the media will necessitate a retransmission of the packet. The output queues eliminate the need to retransmit the packet through the switching fabric.

The output queues, like all queues in the LS architecture, are implemented as linked lists of fixed size packet buffers. In spite of the underlying linked list structure; the output queues in the LS100 are logically structured as FIFOs. The benefit of the dynamic buffer allocation is that it allows the output queue depth to grow as needed without having to pre-allocate large quantities of memory.

Merging traffic from the multicast output queue and the unicast output queues is managed on a per port basis by the Output Port Muxes.

10.0 Output Port Muxes

The output port muxes are used to merge single destination unicast packets queued for a particular output port with packets queued for multicast in the Multicast Queue.

Each of the four LS100 output ports contain two sets of buffer pointers. One set of pointers is used to control the port’s access to its own output queue and the other set is used to control the port’s access to the common multicast queue. The pointers not only give independent access to both queues for each output but also give each output independent access to the multicast queue. The per-port multicast queue pointers allow the packets stored in the multicast queue to be output at the full rate independently for each port.

Keep in mind that each port’s output queue can contain only unicast packets destined for the output. While the LS100’s multicast queue can contain true multicast packets or it can contain unicast packets that have been flooded (see section 4.1.1 Unicast Packet Handling).

10.1 Resolving Packet Ordering

When an output port is ready to send out its next packet it checks its two sets of pointers to determine if either of the two queues has a packet ready. If only one queue is ready that queue’s packet is sent out. If both queues are ready the Switch In timestamp stored in both packet’s headers are examined. The packet with the earlier timestamp is sent out.
10.2 Freeing Queue Entries

Once a unicast packet stored in the port’s output queue is sent out the port, the entry in the queue is no longer needed. These queue entries are freed up immediately making space for more packets in the buffer memory. The multicast queue cannot work this way, however.

Packets stored in the multicast queue can be discarded only after they have been sent out all of the output ports they are supposed to go to. This can range from all four of LS100 ports to none of them. Which ports the packet must go out is controlled by the four USAGE bits (one for each port). If the USAGE bit for a port is set to a one the packet must be sent out that port. The USAGE bits are set to all ones in the case of a flooded unicast packet. In the case of multicast packets the USAGE bits are variable and are returned from the ATC entry (see section 8 - Output ATC Mapping). When a packet stored in the multicast queue is sent out a port the USAGE bit for that port is cleared to a zero. Only after all four USAGE bits are zero can the entry in the queue be freed up.

11.0 MAC Output Processing

The MAC output block handles the transfer of PDUs across the MII interface to the physical layer device. Functions this block performs include:

- CSMA/CD support
- Packet Filtering
- Carrier Assertion Flow Control

11.1 Packet Filtering

Packet filtering is performed by the MAC output block for either of two functions. The first is to implement 802.1d port states. The second is packet loopback filtering.

11.1.1 802.1d Port State Filtering

Like the MAC input block, the MAC output block contains logic to filter packets in a manner compliant with the with port states defined in the 802.1D specification. The filtering behaviors supported by the MAC output block are:

- Pass transmitted non-BPDU frames
- Pass transmitted BPDU frames
These filtering behaviors can be independently configured on a per-port basis. Table 3 shows how these filtering behaviors correspond to 802.1d port states.

<table>
<thead>
<tr>
<th>802.1D Port State</th>
<th>Pass Transmitted Non-BPDU Frames</th>
<th>Pass Transmitted BPDU Frames</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disabled</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Blocking</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Listening</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Forwarding</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Table 3: MAC Output Packet Filtering

11.1.2 Packet Loopback Filtering

The purpose of this function is to prevent multicast packets or flooded unicast packets from being transmitted out the port that they arrived from. This is accomplished by comparing the source port ID field in the switch packet header with the ID of the output port. If they are the same the packet is not transmitted.

11.2 Carrier Assertion Flow Control

The LS100 supports a link flow control technique for half duplex environments that we refer to as Carrier Assertion Flow Control. Support for IEEE 802.3 full duplex flow control will be included in future members of the LS family.

The Carrier Assertion flow control is implemented with the LS chipset as follows:

- When the number of frames in the input queue exceeds some programmed amount (the “high water mark”), the LS100 will generate a management exception.
- In response to this exception the management CPU will enable the Carrier Assertion logic in MAC controller for the port that is experiencing input queue congestion.
- The MAC logic for the port will complete reception of any frame-in-progress on that port. As a result the high-water mark should leave a buffer available for a frame-in-progress.
- Following the end of receiving a frame-in-progress, the MAC block waits a shortened interframe gap time of 48 bit times (instead of the normal 96) and begins sending preamble. The MAC block will transmit the preamble sequence for 14400 bit times, pause i.e. stop sending preamble for 48 bit times, and begin send preamble again. The MAC logic will continue to send this preamble/pause sequence until disabled by system software.
- System software must monitor (via polling) the transmit queue depth and will disable the Carrier Assertion feature when it drops below a programmed “low water mark”. Normally the low water mark will be set lower than the high water mark in order to implement hysteresis.
12.0 References

1. I-Cube [1996]. LS100 Data Sheet.
4. I-Cube [1996]. Application Note: Implementing 802.1D Port States and BPDU Handling with the LS Architecture.