Support More Than 100Mbps Ethernet on MT92210/220

Application Note

Contents

Issue 1 November 2002

1.0 Background

2.0 Adding GMII on MT92210/220

2.1 POS-PHY to GMII

2.1.1 FPGA with MAC Function

2.1.2 FPGA without MAC Function

2.2 UTOPIA to GMII

2.3 MII to GMII

3.0 Conclusion

1.0 Background

Both the MT92210 and the MT92220 voice over IP processors can covert 1023 full-duplex PCM or ADPCM voice channels to RTP/UDP/IP[1] packets, and vice versa. As shown in Figure 1[2], the MT92210/220 have an H.110 TDM bus carrying PCM or ADPCM data and two network interfaces connecting to IP and/or ATM network. The primary network interface can be configured to one of the following three modes.

1. A UTOPIA interface for IP over AAL5 or LANE (LAN Emulation) applications;
2. A POS-PHY interface for PPP over SONET application;
3. An MII interface for 10/100Mbps Ethernet applications.

The secondary network interface is always a UTOPIA port.

To send IP packets over Ethernet or Fast Ethernet, the primary port should be configured as an MII interface connecting to a 10/100Mbps PHY device. However, due to the 100Mbps bandwidth limitation of Fast Ethernet, sometimes it is impossible to put all 1023 voice channels through the MII interface even though the MT92210/220 has such capacity.

1. RTP header is optional
2. AAL2 assembly and disassembly engines are available on MT92220 only.

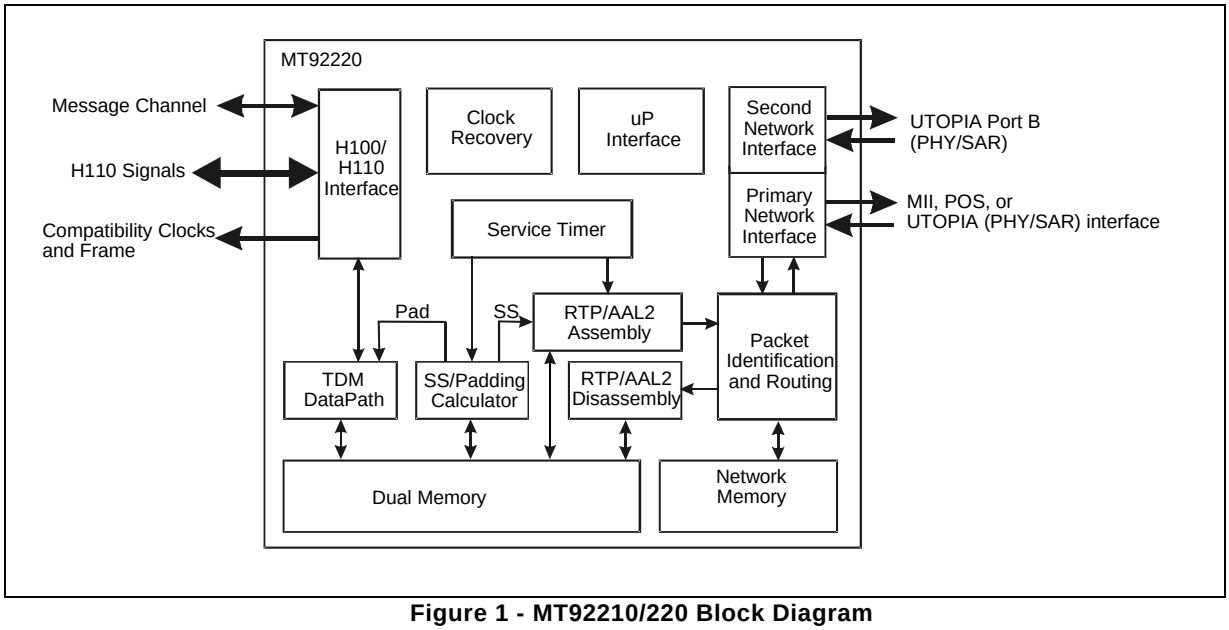

Figure 1 - MT92210/220 Block Diagram

The actual number of voice channels supported by the MII interface depends on several factors, including the packet header size, the number of channels per packet, and the number of samples per channel. Assuming that T is the total number of voice channels, N is the average number of voice channels per connection, M is the average number of frames per channel, and H is the average size of all protocol overheads, we have the following constraint:

$$\frac{T}{N} \times \frac{8000}{M} \times \left(N \times M + H\right) \times 8 \leq 10^8$$

The left side of the equation is the total required bandwidth, which must be less than 100Mbps. Some examples are given in Figure 1 where N is set to 1 (only one channel per packet).

H - overhead (bytes)	M - frames/channel	T - maximum channels	Total bandwidth (Mbps)
78[1]	1	19	96.06
78	40	529	99.88
78	160	1050	99.86
98[2]	40	452	99.80

Table 1 - Maximum Channel Number in Various Configurations

1. Assuming 8 bytes (pre-amble) + 14 bytes (MAC header) + 20 bytes (IPv4 header) + 8 bytes (UDP header) + 12 bytes (RTP header) + 4 bytes (CRC) + 12 bytes (Interframe gap).
2. Assuming 8 bytes (pre-amble) + 14 bytes (MAC header) + 40 bytes (IPv6 header) + 8 bytes (UDP header) + 12 bytes (RTP header) + 4 bytes (CRC) + 12 bytes (Interframe gap).

From Table 1, we can see that for a generic voice over RTP/UDP/IP/MAC application, the payload size must be at least 160 frames or 20ms in order to get all 1023 channels through a single MII interface. For 40 frame or 5ms payload, only 529 channels (half of the MT92210/220 capacity) are supported by the MII interface.

The following is a list of practices that can increase the channel capacity on the MII interface.

1. Implement trunking. Put more TDM channels into a single RTP/IP connection.
2. Reduce overhead by eliminating RTP or UDP headers.
3. Use silence suppression technology to reduce traffic.

2.0 Adding GMII on MT92210/220

In spite of the bandwidth limitation of the MII interface, the primary port on MT92210/220 still has sufficient bandwidth for 1023 or more voice channels when programmed in POS-PHY or UTOPIA mode. In this section we will present some ideas of converting a POS-PHY or UTOPIA port to a GMII interface in order to support 1023 voice channels over Gigabit Ethernet.

2.1 POS-PHY to GMII

It is expected that an external device, e.g. a FPGA or microprocessor, will perform the convention between POS-PHY and GMII. Two scenarios will be discussed: FPGA with MAC function and FPGA without MAC function.

2.1.1 FPGA with MAC Function

Figure 2 shows an implementation where the MAC function is included in the FPGA. Essentially, three functional blocks are required in FPGA. They are illustrated below.

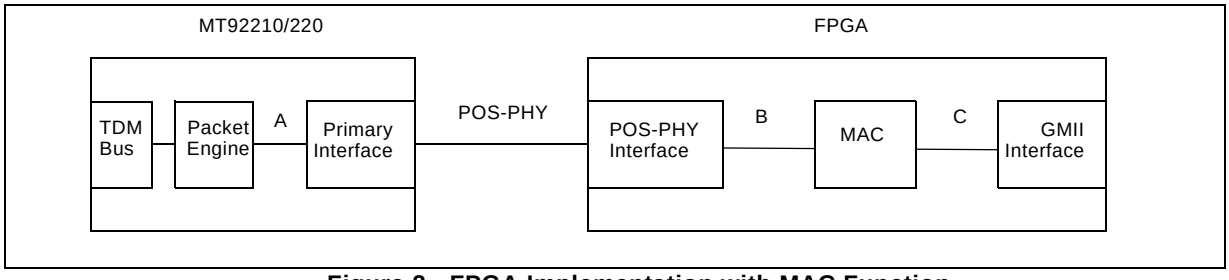

Figure 2 - FPGA Implementation with MAC Function

POS-PHY Interface

This block is the interface to MT92210/220 primary port which is configured in the same POS-PHY mode. POS-PHY is a handshaking bus used for packet transportation. The POS-PHY interface on MT92210/220 is a simplified version of POS-PHY Level 2 standard. It doesn't have an address bus, therefore it is a point-to-point connection only. All necessary pin connections are shown in Figure 3.

Although POS-PHY is meant for PPP packets, the POS-PHY interface on the MT92210/220 is transparent to the link layer protocol. It can transmit and receive any packet that starts with one of the following headers:

- PPP
- IP
- LANEv1
- LLC
- MPLS
- MPOA
- RTP

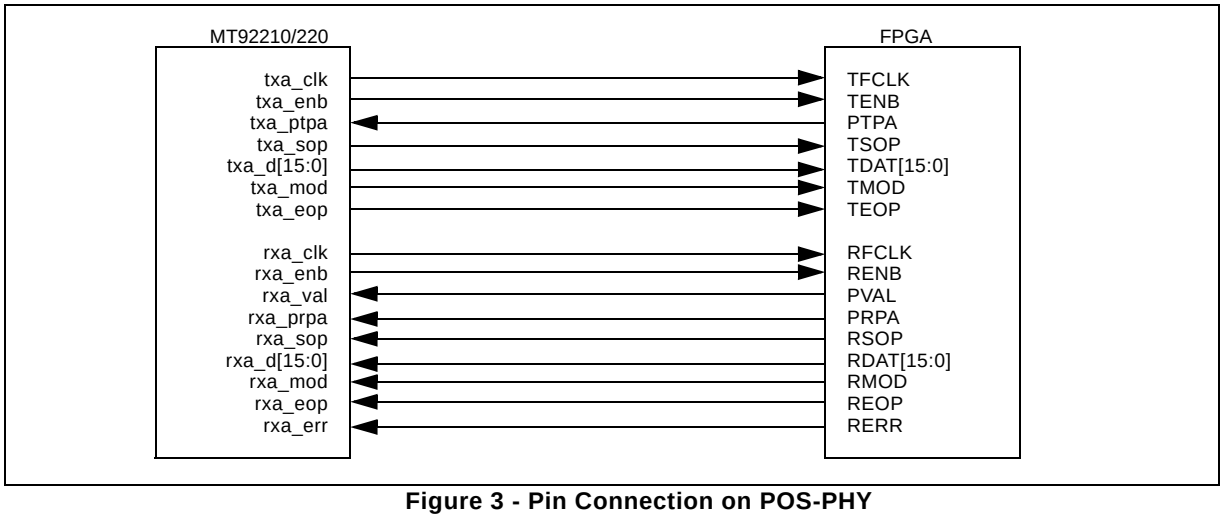

Figure 3 - Pin Connection on POS-PHY

MAC

This block performs general MAC layer functions. It takes IP packets from POS-PHY Interface, adds the MAC header and the FCS, and passes the MAC frame to the GMII Interface. On the reverse path it extracts IP packets from the MAC frame.

GMII Interface

This connects to a Gigabit Ethernet PHY. Many FPGA vendors offer a GMII module. It is also available on many microprocessors.

As shown in Figure 4, IP packets are transported between the POS-PHY interfaces. The MAC layer is inserted in the FPGA.

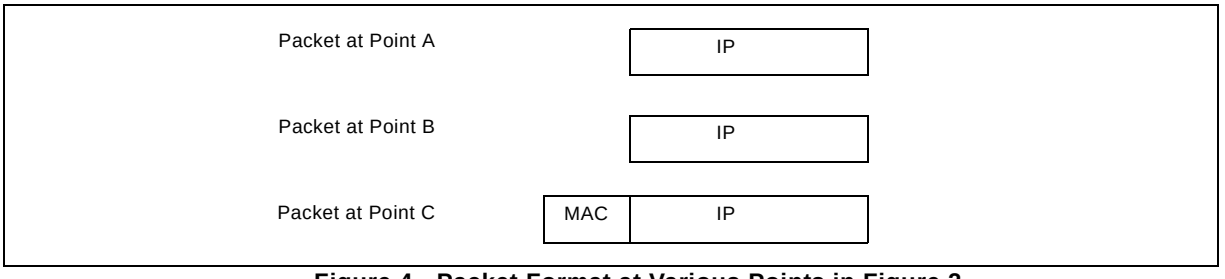

Figure 4 - Packet Format at Various Points in Figure 2

2.1.2 FPGA without MAC Function

The FPGA implementation in Figure 2 can be simplified by removing the MAC block. We can make use of the MAC function that is available in MT92210/220. However, MT92210/220 won't recognize the MAC frame. Any MAC frame must be preceded by a 2-byte LANEv1 header. Therefore, a Header Modification block in the FPGA is required to simply add or remove 2 bytes[1] in front of MAC frames. Figure 5 and Figure 6 show the block diagram and packet formats, respectively.

Both the POS-PHY Interface and the GMII Interface in Figure 5 are the same implementation as in Figure 2.

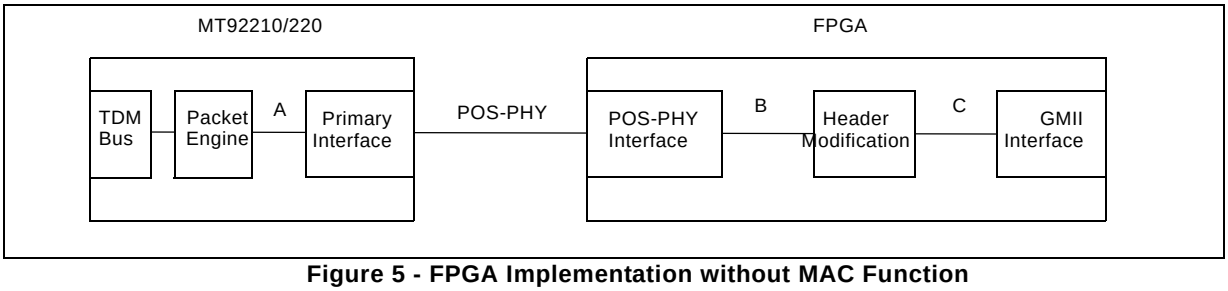

Figure 5 - FPGA Implementation without MAC Function

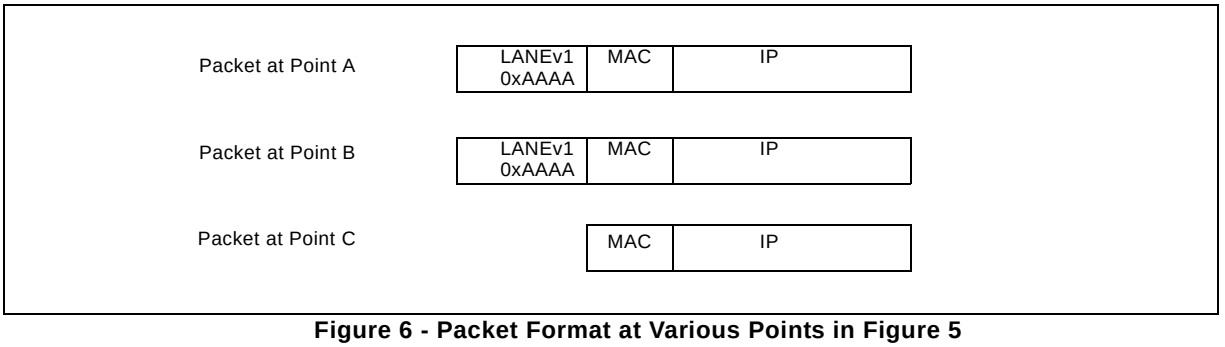

Figure 6 - Packet Format at Various Points in Figure 5

1. The value of these two bytes can be anything other than 0xFFXX.

2.2 UTOPIA to GMII

It is also possible to use the UTOPIA bus instead of the POS-PHY for GMII expansion, if it is easier to implement in the external device. For instance, some microprocessors may have an embedded UTOPIA bus and a SAR function that can be reused. In such a case, some sort of AAL5 SAR function is required to convert between cell and packet as shown in Figure 7 and Figure 8.

Again, the MAC frame must be preceded with a LANEv1 header before being assembled into AAL5 cells. Header Modification serves the same function as in Figure 5.

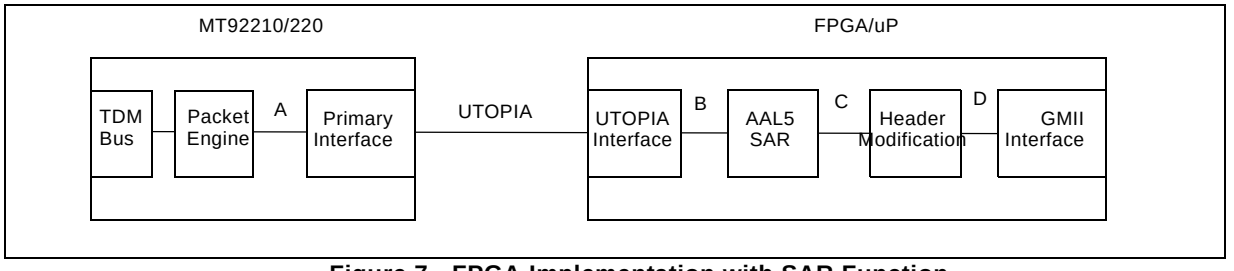

Figure 7 - FPGA Implementation with SAR Function

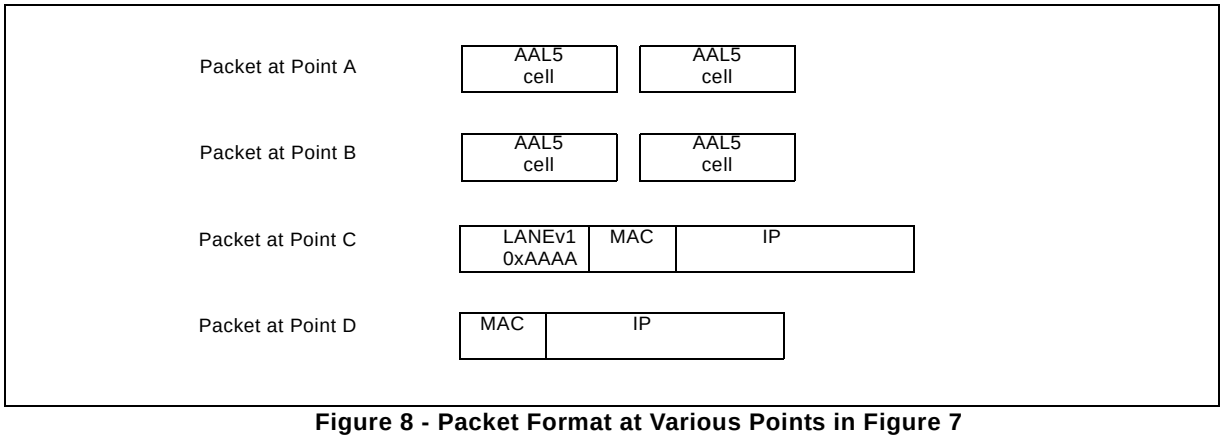

Figure 8 - Packet Format at Various Points in Figure 7

2.3 MII to GMII

If the MII interface can only support half of the channel capacity offered by MT92210/220, as in the example shown in Table 1, using two devices will allow the full 1023 channel capacity. An Ethernet switch such as the Zarlink ZL50407 can multiplex two or more MII ports onto a GMII port.

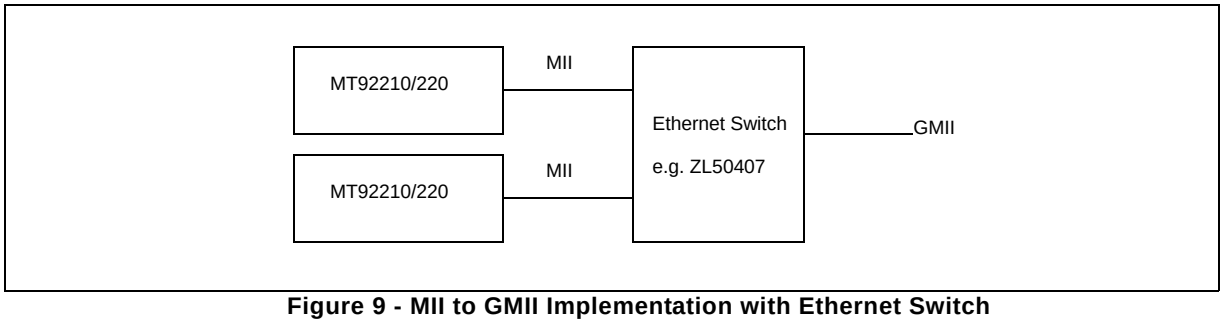

Figure 9 - MII to GMII Implementation with Ethernet Switch

3.0 Conclusion

Although the MT92210/220 can support 1023 voice over IP connections, the MII interface may become a bottleneck for some configurations. To run the MT92210/220 at full capacity in these configurations, a GMII interface may be used. This application note discussed several ideas that can be implemented by an FPGA or a microprocessor.

For more information about all Zarlink products
visit our Web Site at

www.zarlink.com

TECHNICAL DOCUMENTATION - NOT FOR RESALE