

SIGNAL PROCESSING CIRCUIT

The present invention relates to time division multiplexed (TDM) communications systems in general and the processing of the signalling, in particular, but not limited to, Signalling System No.7 (SS7) Layer 2 signalling.

The present invention is directed to processing SS7 Layer 2 signalling associated with time division multiplexed communications protocols such as those defined in International Telecommunication Union recommendation ITU-T G.704. SS7 Layer 2 signalling is defined in recommendation ITU-T Q.703. In particular the "T1" frame structure based on a data rate of 1544Kbit/s as defined in the ANSI series of standards T1.111 and the "E1" frame structure based on a data rate of 2048Kbit/s. The T1 frame structure comprises a frame of 24 time slots each time slot comprising 8 bits plus a frame synchronisation "F" bit outside of the time slots at the start of each frame indicating frame synchronisation. In addition a multi-frame synchronisation signal is present for identifying the start of a multi-frame comprising 24 frames (as defined in ITU-T G.704). The E1 frame structure defines a frame of 32 time slots, each comprising 8 bits with time slot zero used for frame synchronisation. In addition spare bits in time slot zero may be used for data signalling over a multi-frame sequence.

Conventional microprocessors and digital signal processors (DSP) have been used to process SS7 Layer 2 signalling under software control. However, the highly iterative nature of SS7 Layer 2 signal processing places a great demand on the resources of such processors. In a multi-data stream communications interface, one microprocessor or DSP has been required per data stream to process SS7 Layer 2 signalling with resulting cost and power consumption

penalties.

The present invention provides a circuit for processing signalling associated with one or more time division multiplexed (TDM) data streams in which each data stream is shared between a plurality of data channels, in which each data channel is allocated a sequence of time slots in the data stream; and in which each time slot comprises a plurality of bits; in which operation of the circuit is synchronised to a clock signal and the circuit comprises means for processing one bit in one cycle of the clock signal.

The present invention also provides a circuit for processing signalling associated with one or more TDM data streams in which each data stream is shared between a plurality of data channels in which each data channel is allocated a sequence of time slots in the data stream; and in which each time slot comprises a plurality of bits; in which the circuit comprises means for sequentially processing the bits of a data channel; in which the circuit comprises processing means for processing a plurality of parameter bits simultaneously in a single step; in which the plurality of parameter bits comprise the data channel bit to be processed, the most recently processed bits from that data channel and a plurality of status bits.

In a preferred embodiment the signalling comprises signalling system No. 7 (SS7) Layer 2.

In a further preferred embodiment the invention provides processing means for processing SS7 Layer 2 messages, the processing means comprising a functional block for processing transmit and receive bits in successive operations.

This Sheet Has Been Replaced



The invention will now be described by way of example with reference to the drawings in which:

Figure 1 shows in block diagrammatic form a communications system incorporating the circuit of the invention;

Figure 2 shows the main functional blocks of the circuit of the invention;

Figure 3 shows the timing of the oversampling operation according to the invention;

Figure 4 shows the transceiver block of the circuit of the invention in more detail;

Figure 5 shows the message store buffer of the circuit of the invention in more detail;

Figure 6 shows the organisation of the message store according to the invention.

The term "deformatted" is used here applied to an SS7 signalling channel to denote that the channel does not include the layer 2 byte alignment features, nor the check bits i.e. flags, stuffed zeros, Cyclic Redundancy Check (CRC) bits. The deformatted channel appears on the host side of the Signalling Handler. The term "formatted" is used here applied to an SS7 signalling channel to denote that the layer 2 byte alignment features, and the check bits i.e. flags, stuffed zeros, CRC bits. The formatted channel appears on the TDM Interface side of the Signalling Handler.

The term receive is used here to denote a direction from the TDM Interface side to the host side of the Signalling Handler and the term transmit is used to denote the opposite direction.

According to a first embodiment of the invention, the Signalling Handler is able to handle both

Media and Signalling data.

Fig. 1 shows a communications system comprising the Signalling Handler of the present invention for processing signalling and media as part of a TDM/Internet Protocol (IP) interface.

The TDM/Internet protocol interface of Figure 1 functions to connect an IP environment, as represented by the dual Ethernet links at the left of the figure with a TDM environment as represented by the PSTN connections at the right of the figure. The interface is controlled by a host processor HOST which may comprise a plurality of dedicated processors working in cooperation, suitable parts being commercially available. The host processor is provided with two serial controllers FCC for handling communication with the Ethernet communications system and with memory in the form of synchronous dynamic ram (SDRAM) and field erasable/programmable read only memory (FEPR0M) with which the host communicates via a memory bus. The host communicates with the Signalling Handler circuit by means of a local bus which carries address, control and data signals: the data signals including de-formatted data received from, or for transmission to, the TDM communications system. The Signalling Handler is connected to the TDM communications system via four separate TDM datastreams which are routed from the TDM communications system (e.g. the public switched telephone network (PSTN)) via a quad PCM termination circuit which terminates the four PSTN PCM connections. In addition, the host is also connected to the four TDM datastreams to provide flexibility should it be desired to allow the host to process TDM data directly in future. Connections to the four TDM data streams (one per PSTN PCM connection) is via a TDM switch implemented as a PLD, with the exception of the four receive TDM datastreams to the

Signalling Handler which by-pass the switch and pass direct from the quad-PCM termination to the Signalling Handler.

The TDM/IP interface circuit comprises, in a preferred embodiment for handling media data in addition to signalling, a plurality of DSP arrays. For the media functionality, the Signalling Handler provides the interface between an array of DSPs and the host processor. The DSPs perform media transformation and monitoring functions and part of the IP media protocol stack. As shown in Figure 1, four groups of six DSPs each are arranged in communication via four media buses with four media interfaces on the signalling handler.

The TDM switch PLD performs a mapping function for distributing TDM time slots among the DSPs. Each DSP comprises two TDM input ports resulting in twelve TDM inputs being provided to the TDM switch PLD.

The Signalling Handler also has a connection to its own local memory (Packet Memory/SS7 Message Store) which it uses for temporary storage of media packets and/or SS7 layer 2 messages en route between the IP and TDM communications systems. SS7 Layer 2 messages are also known as signalling units (SU).

The host incorporates an interface to an IP based communications system (see the dual 100Mbit/s Ethernet controllers (FCC)). It deposits IP packets received from the IP system for transmission to the TDM system into the Packet Memory from where they can be passed by the Signalling Handler to the appropriate DSP. The Signalling Handler transfers media and control packets received from the TDM system via the DSPs into packet Memory from where

they can be transferred by the host over the Ethernet links or interrogated or copied by the host program.

When a DSP is ready to send Media data (i.e. an IP packet or a control message) it raises a receiver request (HRRQ) and the Signalling Handler transfers the data from the DSP to an address in Packet Memory previously programmed by the host. The packet or control message contains a header which indicates the length of the data to be transferred. The header also contains a channel identifier.

When programmed by the host, the Signalling Handler transfers a block of Media data (e.g. an IP packet) from packet memory to a selected DSP. This transfer has a maximum block size of 512 Kbytes. This allows this type of transfer to be used for DSP program code enhancement.

For the SS7 Layer 2 functionality, the host software writes deformatted messages, via the host local bus interface, to the attached Message Store RAM. These are retrieved and processed by the Signalling Handler, and then transmitted as formatted messages to four TDM interfaces (as described below).

Similarly, formatted messages received from the TDM system on the four TDM interfaces are processed by the Signalling Handler and deposited into the Message Store RAM as deformatted messages. The host software then reads the deformatted messages from the Message Store RAM via the local bus.

The Packet and Message Store memories consists of commercially available 4M x 16 Synchronous DRAMs.

For access to 56 kilobit/s and 64 kilobit/s common channel signalling channels, multiframe synchronisation is not required. However, spare multifunction ports are connected for future design enhancement; for example multiframe synchronisation, and signalling data.

Figure 2 shows the internal structure of the Signalling Handler of Figure 1.

The Signalling Handler circuit comprises a host interface for communication with the host processor in addition to an interrupt-controller for notifying the host of significant events in the Signalling Handler. The host interface communicates with the host over the local bus which includes a bi-directional 32 bit host databus. Internally the host interface communicates with the interrupt controller and a transceiver block also via a 32 bit databus. The transceiver is in communication with four bi-directional TDM data stream interfaces (not shown) via TDM oversamplers 1-4, each TDM oversampler having a bi-directional serial data link with one of the TDM interfaces and a bi-directional 8 bit parallel data link with the transceiver. The transceiver is also in communication with the external SS7 Message Store via a message store buffer and memory controller. The message store buffer is connected to the transceiver and to the memory controller via separate 32 bit bi-directional data buses. The memory controller is in communication with the host interface and the external memory via separate bi-directional 32 bit data buses. The TDM Oversamplers (described below), transceiver and message store buffer are collectively referred to as the SS7 logic.

The host interface divides a 66.7 MHz clock signal received from the host by two to produce a 33.3 MHz clock signal. In general, the signalling processing data paths use the 33.3MHz and the host interface and the Packet Memory/Message Store Interface operate at 66.7 MHz. In addition the host interface performs bidirectional data buffering, and buffering and decoding of address signals received from the host.

Resources addressable by Signalling Handler include the SS7 Message Store, the Packet Memory (if present), the SS7 registers and the Media registers (if present). These resources are located relative to a 'base address' and a suitable decode scheme is shown in Table 1, in which the addresses are relative to the Signalling Handler base address.

<u>Block Size</u>	<u>Block Start</u>	<u>Block End</u>	<u>Block Usage</u>
<u>(bytes)</u>	<u>Byte Address</u>	<u>Byte Address</u>	
	<u>(hex.)</u>	<u>(hex.)</u>	
4 Mbytes	01C0 0000	01FF FFFF	Media Registers
4 Mbytes	0180 0000	01BF FFFF	SS7 Registers
16 Mbytes	0080 0000	017F FFFF	Not Available (reserved)
7936 Kbytes	0004 0000	007F FFFF	RAM for Media and SS7
256 Kbytes	0000 0000	0003 FFFF	SS7 Message Store

Table 1

TDM frames are defined as comprising 32 TDM time slots (E1) and 24 TDM time slots plus the frame synchronisation "F" bit (T1). The duration of both types of frames is 125us

If 'n' TDM data channels are active, then the average number of requests from the SS7 Message Store Buffer Resource is $n/2$ per 125 us frame, with an absolute maximum of

$n \times 2$ per 125 us frame. Message Store Buffer resource requests are equal in number of reads and writes. There is no danger of the Message Store Buffer 'hogging' the RAM bandwidth as the number of Message Store Buffer requests per frame are limited as stated above.

For signalling bandwidth calculations, consider that 62 TDM data channels are active and the Signalling Handler is only handling signalling, not media. The Signalling Handler will need to perform 31 accesses on average, and 124 accesses maximum in 125 us, and it is assumed that the host software will require an equivalent bandwidth. Assuming 7 clock cycles per access and 66.7 MHZ operation, then this equates to 5.2% of the bandwidth on average, and 21% absolute maximum. If only 4 SS7 data channels are terminated (Media plus Signalling combined), the average bandwidth utilisation is 0.34% which is negligible.

The Signalling Handler Interrupt Controller generates the interrupt output signal to the host in response to interrupt events occurring within the Signalling Handler transceiver. The Interrupt Controller receives Interrupt Descriptor signals from the transceiver, which indicate the nature of the interrupt event and the relevant TDM interface and channel number. The Interrupt Controller writes this information into the back end of a single 32 word deep by 16 bit wide FIFO. The presence of a valid interrupt descriptor at the front of the FIFO causes the interrupt output to be asserted. The host software has read only access to the front of the FIFO. The completion of a host software read of the FIFO

results in the interrupt output being negated, and the FIFO being clocked on so that as soon as the next Interrupt Descriptor becomes available, it travels to the front of the FIFO.

One TDM Oversampler is provided for each of the four TDM interfaces. The TDM Oversampler oversamples the TDM receive data stream, bit rate 'clock', and frame marker inputs using the 33.3 MHZ clock signal derived from the host interface. This results in a common clock domain for the processing by the Signalling Handler of signalling messages

from all four TDM interfaces irrespective of the phase or accuracy of the incoming clock. Hence the Signalling Handler can cope with TDM data streams from a plurality of different independent networks. Similarly, the TDM transmit data stream is synchronised to the 33.3 MHZ clock signal such that the PCM termination set up and hold margins are met. The TDM Oversampler also performs a serial to parallel conversion on the receive data and vice-versa on the transmit data.

Figure 3 illustrates the intra-time slot count maintained by the Signalling Handler and based on the 33.3MHz clock signal. There are 130.21 periods of the 33.3 MHZ clock in one E1 time slot, and 172.71 periods in one T1 time slot. The TDM Oversampler generates an intra-time slot count from 0 to 129/130 (E1) or from 0 to 171/172 (T1) with the count value 0 starting on the first logic '1' value of the TDM clock received with the TDM data stream corresponding to the first bit of each TDM time slot. Thus realignment is performed at the start of every time slot. The TDM Oversampler samples the received

TDM data, and transmits the TDM transmit data at defined counts of the 'intra-time slot count'. Because the time slot period is not an exact multiple of the 33.3 MHz clock period, then the count will vary between two extreme values as shown in Figures 3(a) and 3(b).

The upper line of Figures 3(a) and 3(b) shows the 33.3MHz clock signal CLK used internally to the Signalling Handler. The second line shows the slower, TDM clock signal received via the Signalling Handler TDM interface from the TDM communications system, each cycle of the TDM clock signal corresponding to one bit of a TDM time slot. At the bottom of the Figure is shown the intra-time slot count maintained by the Signalling

Handler. This count relates to the number of 33.3MHz Signalling Handler clock periods from the start of the current time slot.

Figure 3(a) shows one extreme for an E1 data stream with a total count of 129, whilst Figure 3(b) shows the alternative extreme with a count of 130. This timing mechanism will result in a variance of +/- 30 ns (i.e. one 33.3 MHz period) for the Signalling Handler sample time and transmit time relative to the TDM clock. This will not be significant. A similar timing mechanism is used for T1 interfaces, except for the different terminal count, and the 'F' bit at the start of each frame.

The Signalling Handler transfers TDM data to and from the TDM Interface according to Table 2. In Table 2 B1, B2, etc. denote bits 1, 2, etc. of a particular time slot. The Transmit Frame Sync Signal is output by the Signalling Handler to indicate TDM transmit frame timing.

<u>Transfer</u>	<u>Intra-time slot count value</u>	
	<u>E1</u>	<u>T1</u>
Transmit Frame Sync = '1'	121	160
transmit B1	121	160
receive B1	7	9
transmit B2	7	9
receive B2	23	31
transmit B3	23	31
receive B3	39	52
transmit B4	39	52
receive B4	55	74
transmit B5	55	74
receive B5	72	96
transmit B6	72	96
receive B6	88	117
transmit B7	88	117
receive B7	104	139
Transmit Frame Sync = '0'	104	139
transmit B8	104	139
receive B8	121	160

Table 2

The TDM Oversampler buffers the byte received in serial form from the TDM interface in each time slot, and makes it available in parallel form to the Transceiver soon after reception i.e. at or just before the start of the next time slot. A signal (not shown) is pulsed asserted by the TDM Oversampler to the Transceiver when the buffer is full and

simultaneously a Time Slot Count signal from the TDM Oversampler (not shown) indicates to the Transceiver the time slot number of the data in the buffer.

In the transmit direction, the TDM Oversampler waits until receipt of an asserted signal (TxBA) from the Transceiver indicating that a byte is available. The byte supplied to the TDM Oversampler is to be transmitted to the TDM interface in the following time slot.

The detailed operation of the Signalling Handler transceiver will now be described with reference to Figure 4. As can be seen from Figure 4, the Signalling Handler Transceiver comprises three main logic blocks: SS7 layer 2 state logic, transceiver state logic and the state register which process the SS7 layer 2 messages together with a parameter RAM, a time slot RAM and a set of global registers. Deformatted data, i.e. on the host (Internet Protocol) side of the transceiver is exchanged with the message store buffer (see Figure 2) via a bi-directional 32 bit data bus. The SS7 layer 2 state logic communicates with the TDM oversamplers (see Figure 2) via a bi-directional 8 bit data bus. In the receive direction four 8 bit data buses, one from each TDM over-sampler are multiplexed in a 4-1 multiplexer under control of the receive select (RXSEL) signal onto a single 8 bit bus connected to the SS7 layer 2 state logic. In the transmit direction a single 8 bit data bus is connected from the SS7 layer 2 state logic to a set of four 8 bit registers each register having a separate 8 bit data bus connection to one of the TDM oversamplers. The SS7 layer 2 state logic is also connected to message received/interrupt logic, itself connected to a register via two unidirectional 20 bit data buses. The message received interrupt logic reacts to events occurring in the Transceiver, and particularly to the complete reception of a message. The interrupt logic counts the received messages per TDM data

stream and stores this number in the register.

The SS7 global registers, parameter RAM and time slot map RAM are all in communication with the host via a bi-directional 32 bit data bus and a 9 bit address bus. The parameter RAM is connected to the SS7 layer 2 state logic via a 64 bit output data bus for sending data to an assembly register. The assembly register is connected by a 320 bit data bus to a 2:1 multiplexer which selects between the 320 bit input from the assembly register and a separate 320 bit input from the state register and outputs the selected data via a third 320 bit data bus to the SS7 layer 2 state logic. The SS7 layer 2 state logic is connected to a further, output 320 bit data bus for sending data to a dis-assembly register. The dis-assembly register is connected via a 320 bit data bus to a 5:1 multiplexer for dividing the 320 bit word received from the SS7 layer 2 state logic into five parts, each 64 bits wide. The 5:1 multiplexer is connected via a 64 bit data bus to the parameter RAM. The output 320 bit data bus from the SS7 layer 2 state logic to the disassembly register also connects to the state register for transferring data from the SS7 layer 2 state logic to the state register. The state register functions to temporarily store the 320 bit data words received from the SS7 layer 2 state logic and to output them on the 320 bit data bus leading to the 2:1 multiplexer, as described above.

In operation, at the beginning of the processing of a new TDM time slot, the relevant parameters (i.e. those generated the last time the corresponding time slot was processed) are read by the SS7 layer 2 state logic, via the 2:1 multiplexer from the assembly register

where they have been pre-fetched from the parameter RAM in a plurality of reads. The channel is processed one time slot at a time (i.e. in 8 bit chunks - or 7 bit chunks in the case of 56 kilobit/s T1. During the processing of the time slot, the parameters are circulated through the state register once for every bit of that time slot, each bit of the time slot being processed by the SS7 layer 2 state logic in a single 33.3 MHz clock period. On completion of the processing of the time slot the resultant, new parameters are written to the dis-assembly register from where they are written in a plurality of writes via the 5:1 multiplexer to the parameter RAM for storage until the next corresponding time slot is received for processing. Advantageously, accesses to the parameter RAM, both the reading into the assembly register and the writing from the dis-assembly register can take place independently of the operation of the SS7 layer 2 state logic thus allowing a "parallel" operation.

The transceiver state logic receives an Assert Receive Byte Available (RxBA) signal from each of the four TDM oversamplers for indicating which of the TDM interfaces have a new TDM time slot ready for processing. The transceiver state logic also receives four 5 bit time slot counts, one count being received from each of the four TDM oversamplers to indicate the number of each of the four current time slots on the four TDM interfaces and outputs an Assert Transmit Byte Available (TxBA) signal to each of the four TDM oversamplers to inform them when a new valid transmit time slot is available. Each time slot number corresponds to one of the actual channel numbers input to the time slot map RAM. The receive select (RxSEL) and transmit select (TxSEL) signals are generated by the transceiver state logic to control selection of TDM

oversampler/TDM interface for transmit and receive messages. The SS7 layer 2 state logic operates both as transmitter and receiver using the same circuit in successive time intervals for transmit and receive functions, advantageously reducing the number of logic gates required by virtue of the Signalling Handler logic functions. The 33MHz clock signal ensures that there is sufficient time to do this, as described below, with reference to Table 4.

The state of a TDM data channel with respect to SS7 layer 2 can be wholly described by 377 state bits, or parameters: 271 for the receiver and 106 for the transmitter. A complete set of these parameters for all 62 potential channels may be held on-chip in a Parameter RAM. Considering the receiver first, for each time slot, the Assembly Register assembles a set of receive parameters obtained from the Parameter RAM by successive reads therefrom. The parameters comprise status information plus data comprising the current formatted TDM data byte from the TDM Oversampler, plus data bits received during the previous time slot relating to the current data channel as required, such that at least the seven data bits immediately preceding the data bit next to be processed are available. This is necessary to allow the identification of flags, inserted zeros, etc.

A partially deformatted word is stored temporarily as additional parameter bits in the Parameter RAM. Upon completion of the processing activity for each time slot (i.e. for each 8 bit or 7 bit chunk of data), a new set of parameters, representing the current status of the processing circuit (comprising the SS7 state logic, the transceiver state logic and the state register), together with processed and yet-to-be-processed data are transferred

en bloc to the Dis-Assembly Register, from where the parameters are written back to the Parameter RAM in a plurality of successive writes as dictated by the limited width of the parameter RAM data bus. The deformatted data resulting from the processing is collected as additional parameters. Once a 32 bit word has been collected it is written to the Message Store Buffer of Figure 2 and subsequently transferred by the memory controller to the Message Store of Figure 1. Hence the Message Store Buffer acts as a temporary store (e.g. FIFO memory) to permit the transceiver to write deformatted data without incurring any wait states.

When a complete message has been received, the SS7 Layer 2 State Logic signals the Message Received Interrupt Logic which maintains a separate count for each TDM Interface. When any one of these counts reaches a predetermined limit, an interrupt event is signalled to the Interrupt Controller to inform the host.

A similar procedure applies to the transmit process, with a set of transmit parameters read prior to processing from the parameter RAM via the Assembly Register, and written back to the parameter RAM afterwards via the Disassembly Register. Deformatted data is read from the Message Store in words of 32 bits. The Transceiver generates a read request for a 32 bit formatted word (or for the Buffer Descriptor) one TDM frame in advance of the data being required. The Message Store Buffer and Memory Controller guarantee to retrieve the data within one TDM frame period. The Transfer of a deformatted 32 bit word from the Message Store to the Transceiver for processing results in a sequence of formatted bytes being output to the appropriate

TDM Interface. As with receive, the data relating to a particular data channel is processed 8 bits at a time with the residual deformatted bits held temporarily as parameter bits in the Parameter RAM.

The scheduled transfers of 32 bit words to and from the Message Store occur approximately once every four TDM frames in each direction; it is not an exact number because the formatted bandwidth is greater than the de-formatted bandwidth due to flags, check bits and stuffed zeros which occur in the formatted data but are absent from the deformatted data.

The Transceiver operates using 'logical channel' numbers 0 to 61; these are used for efficiently addressing the Parameter RAM and Message Store RAM. The 'logical' channels are mapped to actual TDM data channels (each constituted as a sequence of time slots on a particular TDM data stream passed by a particular TDM Interface) in the Time slot Map RAM. The Time slot Map RAM is preloaded at configuration with all locations set to zero which corresponds to all channels inactive. The host software has read and write access to the Time slot Map RAM and sets up the mapping between logical and actual channel numbers. The Signalling Handler has read only access, the address supplied by the Signalling Handler being the actual channel number. The format of the data byte in this RAM is shown in Table 3.

RAM bit	7	6	5	4	3	2	1	0
	active	reserve	<-----			logical_channel		----->

Table 3

A logic '1' in the "Active" field activates the addressed time slot for receive and transmit SS7 processing, and the Logical_channel value allocates a logical channel number to the addressed time slot.

In a preferred embodiment, the transceiver processes one time slot from each of the receive and transmit directions of one TDM data stream as one inseparable sequence of operations synchronised to the oversampling (33.3 MHZ) clock as shown in Table 4. Advantageously, considerable speed increase is gained from the "parallel" nature of the Signalling Handler operation according to this embodiment, as is evident from the Table. This parallel operation allows SS7 State Logic processing to take place at the same time as accesses to the Parameter Ram (via the Assembly or Dis-assembly Registers), byte transfers to or from the TDM Interface and 32 bit word transfers to or from the Message Store. The table shows processing steps during time slot N: in the receive direction the data received during time slot N-1 (i.e. the previous time slot) is processed; in the transmit direction, the data for transmission in time slot N+1 (i.e. the next time slot) is processed.

<u>Trans-</u> <u>ceiver</u> <u>State</u>	<u>SS7 State Logic</u> <u>Activity</u>	<u>Access to</u> <u>Parameter</u> <u>RAM</u>	<u>Access to</u> <u>Time slot</u> <u>MAP RAM</u>	<u>B y t e</u> <u>Transfer,</u> <u>TDM</u> <u>interface</u>	<u>32 bit</u> <u>W o r d</u> <u>Transfer,</u> <u>Message</u> <u>Store</u>
<u>select</u>	Select which TDM data channel to process				
test rx active			read rx active and logical channel	rx byte avail	
test tx active			read tx active and logical channel	rx byte avail	
1		Read RPA, TS N-1		rx byte avail	
2		Read RPB, TS N-1		rx byte avail	Receiver retrieve descriptor
3		Read RPC, TS N-1		rx byte avail	
4		Read RPD, TS N-1		rx byte avail	
5		Read RPE, TS N-1		rx byte avail	
6	Receiver processes TS N-1 B1			rx byte avail	Transmitter

7	Receiver processes TS N-1 B2			rx byte avail	retrieve descriptor
8	Receiver processes TS N-1 B3			rx byte avail	
9	Receiver processes TS N-1 B4			rx byte avail	
10	Receiver processes TS N-1 B5			rx byte avail	Transmitt er
11	Receiver processes TS N-1 B6	Read TPA, TS N+1		rx byte avail	retrieve
12	Receiver processes TS N-1 B7	Read TPB, TS N+1		rx byte avail	data
13*	Receiver processes TS N-1 B8	Read TPC, TS N+1		rx byte avail	
14	Transmitter processes TS N+1 B1	Write RPA, TS N-1			Receiver
15	Transmitter processes TS N+1 B2	Write RPB, TS N-1			write data
16	Transmitter processes TS N+1 B3	Write RPC, TS N-1			or
17	Transmitter processes TS N+1 B4	Write RPD, TS N-1			r e a d descriptor
18	Transmitter processes TS N+1 B5	Write RPE, TS N-1			Receiver
19	Transmitter processes TS N+1 B6				write
20	Transmitter processes TS N+1 B7				descriptor
21*	Transmitter processes				

22			Write TPA, TS			tx byte	Transmitt
			N+1			avail	er
23			Write TPB, TS			tx byte	r e a d
			N+1			avail	descriptor
24			Write TPC, TS			tx byte	or write
			N+1			avail	descr.
25							or read
							data

* receiver ignores bit 8, and the transmitter sets bit 8 to '1' for 56 kilobit/s channel rate

Table 4

In Table 4, the states indicated for accessing the Message Store (right hand column) are only utilised as required. As eight bits will be collected per frame for a particular data channel (less if inserted zeros are deleted) a 32 bit word access will be required approximately once per four TDM frames.

These accesses include both the transfer of "data" (i.e. 32 bit deformatted data words), and also the Message Store Buffer descriptors. Neither the receive nor the transmit processes are expected to utilise both read and write data states within the same 125 us frame, so that the maximum number of data accesses for both receiver and transmitter for one time slot in one frame period is two. The receive process reads data only when reading a descriptor and writes data only when a new 32 bit deformatted data word is available or a descriptor value needs to be updated.

The reading and writing of descriptors is performed infrequently the Transceiver reading only at the beginning and writing only at the end of each message. The receive process writes data during message processing, the transmit process reads data during message processing.

The Transceiver is able to service a time slot from all four TDM data streams in $4 \times 28 = 112$ clock periods. As there are 130 33.3 MHZ clock periods within one E1 time slot, and more in a T1 time slot, then all four TDM data streams can be serviced within a time slot period. The 'bandwidth' of the Transceiver is therefore capable of processing all 128 time slots carried by the four TDM data streams. This ability is exploited in a further preferred embodiment of the invention according to which the Signalling Handler comprises the necessary, additional Parameter RAM and Message Store RAM to support the enhanced operation.

The availability of a TDM time slot for processing is indicated by a Receive Byte Available (RxBA) flag, one of which is provided for each TDM data stream. This flag is asserted when the Assert RxBA pulse is received from the TDM Oversampler, and negated when the Transceiver has completed processing the TDM time slot, e.g. on transceiver state 25 (see Table 4). The Transceiver stores the processed transmit TDM byte in an internal register between the end of transmit processing and the end of the time slot (i.e. in one of transceiver states 22 to 25), and generates an Assert Transmit Byte Available (TxBA) pulse to the appropriate TDM Oversampler. The RxBA signals from each TDM oversampler remain active until cleared by the transceiver. This may result in

more than one RxBA signal being active at a time however, as explained above, the transceiver has a sufficient number of clock periods within a single time slot period to enable it to process a transmit and receive byte relating to a transmit and receive time slot on each of the four TDM data streams.

The Transceiver State Logic determines which TDM data stream to service next on a 'round-robin' basis and according to whether the RxBA flag is asserted, and whether the current time slot for that data stream has been enabled, i.e. the 'active' bit in the Time slot Map RAM is asserted, as illustrated by Table 5.

<u>Current State</u>	<u>Previous TDM</u>	<u>Time slot 'active' and RxBA asserted on :-</u>				<u>Next TDM</u>
		<u>TDM 1</u>	<u>TDM 2</u>	<u>TDM 3</u>	<u>TDM 4</u>	
(any state other than select)	X	X	X	X	X	n/a
select	X	0	0	0	0	n/a (note)
select	4	1	X	X	X	1
select	4	0	1	X	X	2
select	4	0	0	1	X	3
select	4	0	0	0	1	4
select	1	X	1	X	X	2
select	1	X	0	1	X	3
select	1	X	0	0	1	4
select	1	1	0	0	0	1
select	2	X	X	1	X	3
select	2	X	X	0	1	4
select	2	1	X	0	0	1
select	2	0	1	0	0	2
select	3	X	X	X	1	4
select	3	1	X	X	0	1
select	3	0	1	X	0	2
select	3	0	0	1	0	3

(note : if Receive Byte Available is negated on all four TDM data streams, then the Transceiver remains in the select state).

Table 5

As stated above, the state of a data channel with respect to SS7 layer 2 and the other functions specified earlier can be wholly described by 377 state bits, or parameters; 271 for the receiver and 106 for the transmitter. A complete set of these parameters for all 62 potential channels is held on-chip in a Parameter RAM organised as 512 words of 64 bits. These parameter bits are listed below, together with an indication of their organisation within the Parameter RAM.

Parameters For Standard HDLC Receiver Features

<u>Mnemonic</u>	<u>Description</u>	<u>No.</u> <u>bits</u>
FB7-1	Formatted Bits. The last seven received bits for flag, abort and stuffed bit detection.	7
C2-0	Contiguous ones count. A three bit count of the number of contiguous ones passing through the receiver. This, together with the abort detector, is used to detect 15 contiguous ones which is the 'idle' condition.	3
DFB31-1	De-formatted bits. Up to 31 formatted bits i.e. with inserted zeros removed for assembling a thirty two bit word to the Message Store.	31

BT11-0	Bit Count. A bit count of the number of bits received since the end of the last flag. This is used to indicate byte boundaries, assemble the 32 bit words, and count the number of bytes in each message	12
SFL	Stored Flag. One bit to indicate that the last byte received was a flag.	1
SVC	Stored Validity Check. One bit to indicate that the calculated check bits were valid at the last byte boundary.	1
SAB	Stored Abort. Indicates that an 'abort' condition has been detected.	1
CRC15-0	Current check bits.	16
EH	Enter Hunt. Set by the receiver when it is looking for a start flag (byte alignment)	1
MR_IEN	Message Received Interrupt Enable. A bit to enable/disable the raising of an interrupt for each received SU.	1
RXOR_IEN	Receiver Overrun Interrupt Enable. A bit to enable/disable the raising of an interrupt when the receiver has filled all buffers.	1
CUR_BUF1-0	Current Buffer. Indicates the current message buffer being used. This is a 2 bit count which is incremented whenever a complete SU is written to the Message Store.	2
<i>(Total bits for standard HDLC Receiver Features = 77)</i>		

Table 6

Parameters For SS7 Layer 2 Receiver Features

<u>Mnemonic</u>	<u>Description</u>	<u>No.</u> <u>bits</u>
OCM	Octet Counting Mode status. This bit is set and reset by the receiver in response to SU error checks. May be set by the Processor to force entry into octet counting mode.	1

OC7-0	Octet count. Used in octet counting mode to determine SUERM/AERM increment events.	8
SUC15-0	Signal Unit Count. A sixteen bit count of all received messages, good or bad, and used to generate the SUERM decrement event (D).	16
AERM_EN	Alignment Error Rate Monitor Enable. A bit to enable/disable the operation of the AERM	1
SUERM_EN	Signal Unit Error Rate Monitor Enable. A bit to enable/disable the operation of the SUERM	1
SUERM7-0	Signal Unit Error Rate Monitor value.	8
SUERM_IEN	SUERM Interrupt Enable. A bit to enable/disable the raising of an interrupt when the SUERM reaches its threshold.	1
AERM7-0	Alignment Error Rate Monitor value. Incremented for consecutive SU errors or octet counting mode events. Reset to zero when a good SU is received, or if the AERM threshold is reached.	8
AERM_IEN	AERM Interrupt Enable. A bit to enable/disable the raising of an interrupt when the AERM proving count has been aborted M times.	1
PFB39-0	Previous Five Bytes. The first five octets of the previous SU are stored. These are used for FISU and LSSU filtering.	40
RCF	Running Compare Flag. Used for FISU and LSSU filtering to indicate a positive compare result on all previous bits.	1
FILTER_EN	Message Filter Enable. A single bit to enable or disable FISU and LSSU filtering.	1
FILTER_RES	FISU/LSSU Filter Reset. A single bit which when toggled in value causes the filter comparator to be reset.	1

PREV_F_RES	Previous value of the F_RES bit. Required to detect changes to the F_RES bit.	1
GSUC15-0	Good SU Count. A sixteen bit counter to count reception of SUs which have passed the acceptance procedure.	16
BSUC15-0	Bad SU Count. A sixteen bit counter to count reception of SUs which have failed the acceptance procedure.	16
BSUC_OV	Bad SU Count Overflow Flag. This bit will be set when the BSUC count overflows, and must be cleared by the host software.	1
PROV_COM	AERM Proving Complete Flag. This bit is set when the AERM timer T4 matures. An interrupt may be raised at the same time.	1
L PROV	AERM Proving Aborted Flag. This bit is set when the AERM reaches its threshold.	1
_ABORT SHORT3-0	Signalling Message limit (in bytes), counting the opening flag and up to but not including the closing flag. This value sets the lower limit for message length for the purpose of message acceptance. Messages shorter than this limit will be discarded and registered as a 'bad' SU by the error rate monitors.	4
LONG7-0	Long Message limit (in bytes), counting the opening flag and up to but not including the closing flag (if any). This value sets the upper limit for message length for the purpose of message acceptance. Messages longer than this limit will be discarded and registered as a 'bad' SU by the error rate monitors.	8
OCM_N7-0	Octet Counting Mode N Parameter. Number of octets per OCM event. Set by host software.	8

AERM_T7-0	AERM T Parameter. AERM threshold value. Set by host software.	8
SUERM_T7-0	SUERM T Parameter. SUERM threshold value. Set by host software.	8
SUERM_D15-0	SUERM D Parameter. SUERM leak rate value. Set by host software.	16
T4_DURATION	AERM Timer T4 Duration. Indicates the T4 duration as either 12 bits or 16 bits. Set by host software.	1
T4_TIMER15-0	AERM Timer T4 current count. Counter counts down, starting at either 0FFF (hex) or FFFF (hex) depending on the value of the T4_DURATION bit, and counts received bytes. When timer matures, the PROV_COMPL bit is set.	16

(Total bits for SS7 Layer 2 Receiver Features = 193)

Table 7

Parameters For Standard HDLC Transmitter Features

<u>Mnemonic</u>	<u>Description</u>	<u>No. bits</u>
TST1-0	Transmitter State. Two bits to encode the transmitter state (Sending Flags, Sending Data, Sending Check Bits, Sending Abort or Idle).	2
C2-0	Contiguous ones count. A three bit count of the number of contiguous ones transmitted by the receiver. Used for zero bit insertion.	3

SBF	Second Byte Flag. Indicates that the second byte of the check bits	1
	or idle pattern is being sent.	
BT11-0	Bit Count. A bit count of the number of bits transmitted since the	12
	end of the opening flag. This is used to indicate byte boundaries,	
	disassemble the 32 bit words, and count the number of bytes	
	transmitted for each message	
TDA0	Transmitter Default Activity. Indicates what should be transmitted	1
	when not transmitting from the Message Store i.e. the 'busy' flag	
	is cleared (Send Flags, Send Idle).	
BUSY	Busy Flag. Indicates that the transmitter is sending a message.	1
RT	Re-transmit Flag. Indicates that the message in the current buffer	1
	should be re-transmitted continuously.	
CUR_BUF1-	Current Buffer. Indicates the current message buffer being used.	2
0	This is a 2 bit count which is incremented whenever a complete	
	SU has been read from the Message Store.	
CRC15-0	Current check bits.	16
DFB31-1	De-formatted Bits. Because of zero bit insertion, it is possible to	31
	have up to seven bits that cannot be transmitted in the current	
	processing period. Additionally, up to three bytes from the thirty	
	two bit word read from the Message Store may be outstanding.	
MEO8-0	Message End Offset. Indicates the position of the end of the	9
	message in the buffer relative to the buffer start address.	
NFBM7-0	Number of Flags Between Messages. It is possible to have	8
	between 1 and 256 flags between transmitted messages (1 sets	
	identical closing/opening flag)	
<i>(Total bits for standard HDLC Transmitter Features =</i>		<i>87)</i>

Table 8

Parameters For SS7 Layer 2 Transmitter Features

<u>Mnemonic</u> <u>bits</u>	<u>Description</u>	<u>No.</u>
TST2	Transmitter State. Augments the TST1-0 bits to add the 'Sending FISUs' state.	1
TDA1	Transmitter Default Activity. Augments the TDA0 bit to indicate what should be transmitted when not transmitting from the Message Store i.e. the 'busy' flag is reset; adds a 'Send continuous FISUs' option.	1
PBSN7-0	BSN/BIB octet of the previous SU. This is required for auto-transmission of FISUs.	8
PFSN7-0	FSN/FIB octet of the previous SU. This is required for auto-transmission of FISUs.	8
<i>(Total bits for SS7 Layer 2 Transmitter Features =</i>		<i>18)</i>

Table 9

Some of these parameters are accessible by both the transceiver logic and the host software, and these occupy the lower 32 bits of the parameter RAM data width. The remaining bits are needed by the transceiver logic only. The lower 32 bits of TPB and TPC are not specified.

The Message Store Buffer buffers the Transceiver requests for accesses to the external

Message Store. This is required so that the Transceiver is not subject to any wait states due to contention between the Signalling Handler and the host for accesses to the Message Store.

The Message Store Buffer completes an access request from the Transceiver within 1 frame period (125 us). In the case of the Transceiver requesting a data word from the Message Store (i.e. for transmission), then the Transceiver generates the request during the frame prior to that in which the data is to be processed so that the data is immediately available when needed.

Operation of the Message Store Buffer will now be described with reference to Figure 5.

In Figure 5 signals at the left hand side of the drawing with the prefix "TR" are received from or sent to the transceiver. These include transceiver request signals from the transceiver to the message store buffer controller and transceiver access address, write data and write control signal output from the transceiver to the access request RAM. The read data RAM outputs data to the transceiver on the 32 bit TR read data bus. Request and acknowledge signals are exchanged between the message buffer store controller and the memory controller. The controller functions to generate address signals for both the transceiver and the message store sides of the access request RAM and also for the message store side of the read data RAM. The address for the transceiver side of the read data RAM is provided by the transceiver on the retrieve address signal.

The Transceiver enters an access request into the Message Store Buffer by providing the

Memory Store address on signal TR_ACCESS_ADDRESS, the data to be written (if a write) on signal TR_WRITE_DATA, indicating the access direction on signal TR_WRITE, and then generating a pulse on signal TR_REQUEST. There is no handshake for this transfer; the Message Store Buffer is expected to transfer the information in a fixed sequence.

The access address comprises:

Logical Channel Number	6 bits
Receive / Transmit Area	1 bit
Buffer Number	2 bits
Buffer Offset (in 32 bit words)	7 bits
TOTAL	16 bits

The address (and data if a write) access information is stored in Access Request dual port (DP) RAM which acts as a FIFO memory with the access information being written and read to/from consecutive addresses; the message store buffer Controller maintains the read and write pointers for this FIFO memory.

When the Access Request DPRAM contains access information, then this is forwarded to the Memory Controller, and the MESSAGE STORE BUFFER_REQUEST signal asserted. When the Memory Controller has completed the access i.e. if a write, then the data has been written to the Message Store, or if a read, then the required data has been output by the message store and made available on signal MESSAGE STORE BUFFER_READ_DATA, then the MESSAGE STORE BUFFER_ACKNOWLEDGE signal is asserted by the Controller. When

a read access is being processed, then the data is stored in the Read Data DPRAM at an address comprising 6 bits corresponding to the logical channel number together with a single further bit whose state depends on whether a receive or transmit buffer was accessed.

When the Transceiver wishes to read the data it had previously requested one frame beforehand, then it simply accesses the Read Data DPRAM without asserting the TR_REQUEST signal.

As described above, the Signalling Handler interfaces to an attached RAM which is used as the Message Store. The organisation of the Message Store is shown in Figures 6(a) and 6(b). The Message Store is composed of a plurality of receive buffers and transmit buffers which have a fixed size of 512 bytes each. There are 2 or 4 buffers allocated as transmit buffers plus 2 or 4 buffers allocated as receive buffers per logical channel; a maximum total of 248 kilobytes for 62 logical channels. Note that the total memory space allocated to the transmit or receive buffers per channel is 2 kilobytes regardless of the number of buffers allocated per data. As shown in Figure 6(b), within each 512 byte buffer, the position of the message is defined by a Message Start Offset (MSO), which is 32 bit aligned, and a Message End Offset (MEO), which is byte aligned. The first 32 bit word at the start of the buffer is the 'Descriptor Word' (DES) which holds information relating to the buffer. The Message Start Offset has a single value for all buffers and channels, receive and transmit, and is programmed by the host software in the MSO register (MSOR) that forms part of the SS7 Global Registers. This facility allows space for the host software to include a header before (or after) the message to minimise copying.

There are specific Descriptor formats appropriate for receive and transmit buffers, but the bit allocation is consistent so that one or more receive buffers could be copied in their entirety to the equivalent number of transmit buffers. Descriptor formats are defined as follows:

Receive Buffer Descriptor.

Address Located at the start of each buffer.
Access Read and Write.
Reset Value -
MEO Bits 8-0 Message End Offset. The offset from the start of the buffer of the last byte of the message. This value is written by the Signalling Handler receiver.

MA Bit 16 Message Available. This flag indicates that a message is available in the buffer.

MA is set by the Signalling Handler receiver after it has finished writing to the buffer. MA is cleared by the host when it has finished with the buffer, thus signalling to the Signalling Handler that the receive buffer is available for re-use.

Transmit Buffer Descriptor.

Address Located at the start of each buffer.
Access Read and Write.
Reset Value -
MEO Bits 8-0 Message End Offset. The offset from the start of the buffer of the last byte of the message. This value shall be written by the host software.

MA Bit 16 Message Available. This flag indicates that a message is available in the buffer.

MA is set by the host software *after* it has finished writing the message to be transmitted to the buffer.

MA is cleared by the Signalling Handler when it has finished with the buffer; (i.e. the message has been transmitted) thus signalling to the host that the transmit buffer is available for re-use.

RT Bit 17 Re-transmit Flag. The host software shall set this flag to cause the Signalling Handler transmitter to continuously re-transmit the message in this buffer. The host software shall clear this bit to end the re-transmission (the Signalling Handler completes the current transmission before stopping).

On initialisation, e.g. at power up, it is necessary to configure the Signalling Handler for E1 or T1 operation on a per TDM data stream basis and to set the required channel bit rate to 56 kilobit/s or 64 kilobit/s accordingly. In addition, unique logical channel numbers are allocated to actual data channels (identified by TDM interface and time slot number) by programming the TIME SLOT_MAP registers; the number of buffers required per channel per direction is set to either 2 or 4 ; the MSOR register is programmed with the offset required for the start of a message in the Message Store buffers; the thresholds which determine the allowed range of message lengths are set by writing to the SHORT and LONG bit fields in the Receive Parameter (RP) A registers (of which there is one per logical data channel); error rate monitor operating parameters are set by writing to bit fields in the RPC and RPD registers (of which there is one each per logical data channel). If interrupts are to be used to inform the host about the presence of received messages, then the required number of messages to be received per TDM interface between interrupts is set; for each channel, the required interrupts are enabled by writing to the appropriate RPA register. The possible interrupt sources are: Messages Received, Receive Buffers Full (overflow condition), Alignment Error Rate Monitor (AERM) event, Signal Unit Error Rate Monitor (SUERM) count reached. Threshold Message filtering is enabled as required by writing to the appropriate RPA register and the required number of flags to be inserted between transmitted messages is set by writing to the appropriate Transmit Parameter (TP) A register.

The transmit buffers allocated to a particular logical channel are used in sequence, buffer 0 first, then 1, 2 etc. until the last buffer, N-1 (N=2 or 4), followed by buffer 0 once more. Buffers are mapped to fixed addresses which are determined by the Message Store Base Address (MSBA), the logical channel number, the direction (transmit or receive), and the buffer number. The procedure for sending (transmitting) a message will now be described

The host reads the MA bit in the Descriptor word of the transmit buffer (or buffers in the case of multiple messages) allocated to the appropriate logical channel to be used and, only if the MA bit is cleared, will write the message or messages, one per buffer, with each message starting at the MSO offset relative to the start of the buffer. If the MA bit is not clear, then the message in that buffer has not yet been transmitted, and a wait state is entered.

In each of the buffers written to, the host writes the new MEO value to the Descriptor. The host sets Transmitter default activity by writing to the appropriate Transmitter Default Activity (TDA) bits. Continuous re-transmission of a message may be achieved by setting a Retransmit (RT) bit in the descriptor for that buffer. The continuous re-transmission may be halted by clearing the RT bit. The RT bit must be set either at the same time as or before the MA bit in order to guarantee that the message will be continuously re-transmitted. Message transmission is started by setting the MA bits in the Descriptors of each of the buffers used.

The Signalling Handler will transmit messages from buffers which have the MA bit set in

numerical sequence and will clear each MA bit when the associated message has been transmitted. For continuous re-transmission the MA bit will remain asserted until the RT bit has been cleared and the last message of the continuous retransmission has been sent. The Signalling Handler will stop when the MA bit in the next sequential buffer is clear even if there are un-transmitted messages further on.

The progress of the Signalling Handler transmitter may be monitored in two ways: (i) by reading the MA bit in the Descriptor of each buffer and (ii) by reading the buffer number currently being transmitted.

As with the transmit procedure, described above, receive buffers are used in sequence, buffer 0 first, then 1, 2 etc. until the last buffer N-1 ($N=2$ or 4) followed by buffer 0 once more. Again, buffers are mapped to fixed addresses which are determined by the MSBA, the logical channel number, the direction (transmit or receive), and the buffer number. The procedure for receiving a message will now be described.

The Signalling Handler receiver will process incoming messages, and deposit them in sequential receive buffers allocated to the appropriate logical channel which have the MA bit cleared. The Signalling Handler will store all bytes except the flags and check bits and will set the MA bit and write the MEO offset value to the Descriptor when the complete message has been transferred to the buffer. The Signalling Handler will not store messages with incorrect check bits, or which do not pass the acceptance criteria (too short, too long, not an integer number of bytes). The Signalling Handler receiver will stop when it detects the next sequential buffer has its MA bit set even if there are empty buffers further on. If the Signalling Handler receiver starts to process an incoming message, but finds that the next

sequential buffer is not available, then an overrun condition is detected and the incoming message will be discarded. A Receiver Overrun interrupt may be asserted to notify the host, if enabled. When one or more messages have been received by the host, these are read from the buffers starting at the MSO and ending at the MEO offset as indicated in the Descriptor. When the complete message has been read, the MA bit is cleared by the host to allow its re-use by the Signalling Handler receiver.

The progress of the Signalling Handler receiver may be monitored in three ways: (i) by reading the MA bits in the Descriptors of each buffer, (ii) by reading the current buffer number being used for message reception and (iii) by waiting for a pre-programmed interrupt. The Messages Received interrupt may be set to interrupt for every m messages per TDM data stream, where $m = 1$ to 32.

The operation of the Signalling Handler in filtering incoming (received) messages will now be described. When enabled, the Signalling Handler filters FISU and/or LSSU messages by comparing the first five bytes of consecutive received messages (after zero bit deletion). When all bits compare, then the Signalling Handler will not store the message in the Message Store and will not count the message for the purpose of generating the Messages Received interrupt. The filter facility may be enabled and disabled under host software control by writing to the appropriate bit in the appropriate RPA register. When the filter facility is operational (enabled), then the host software may reset the filter comparator by toggling the appropriate bit in the appropriate RPA register forcing the next comparison to fail and allowing one FISU or LSSU to be received and stored.

The circuit of the invention could be implemented using commercially available

programmable logic device (PLD) components such as the Xilinx Virtex range, details of which are contained in the manufacturer's data sheet for this family of devices. Suitable devices in this range include the XCV200-4 and XCV300-4. Virtex is a Trade Mark of the Xilinx corporation. In particular, the Xilinx Virtex family includes DP RAMs. These are utilised as in Table 10.

<u>RAM Block</u>	<u>Requirement</u>	<u>Xilinx Virtex</u>
		<u>Implementation</u>
Transceiver Parameter RAM	512 x 64 D-RAM	8 off 512 x 8 Block RAMs
Transceiver Time slot Map RAM	128 x 8 D-RAM	1 off 256 x 16 Block RAM
Message Store Buffer Access Request RAM	124 x 49 D-RAM	2 off 256 x 16 Block RAMs
Message Store Buffer Read Data RAM	124 x 32 D-RAM	1 off 256 x 16 Block RAMs
		*
		*

* Note: the data to be transferred as 2 beats

Table 10

In a further preferred embodiment, the functionality of the Signalling Handler may be extended, in particular by enlarging the parameter RAM and providing for a larger message store capable of storing 128 messages per logical channel.

Although the system of Figure 1 has been described interfacing to an IP communications system (by way of example via dual Ethernet connections), the circuit of the present invention

has application to other communications systems such those implementing asynchronous transfer mode (ATM).

The frequency of the Signalling Handler clock signal does not need to be set at 33.3MHz and could be set to another frequency although a minimum of around 30MHz will be needed if the Signalling Handler is to be able to process the maximum number of data channels given above. The maximum value for this clock signal frequency is in theory set by the limitations of the technology used to implement the circuit. With current PLD technology this is of the order of 50MHz although it is expected that this figure could be increased as technology develops.

The direct connection of the host to the TDM data streams is not essential to the present invention and may be omitted. Similarly, the connection of media data via the Signalling Handler is not essential to the invention and the DSPs, media interfaces, packet memory and associated media features may be omitted.

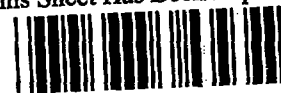


Claims

1. A circuit for processing signalling associated with one or more time division multiplexed (TDM) data streams in which each data stream is shared between a plurality of data channels, in which each data channel is allocated a sequence of time slots in the data stream; and in which each time slot comprises a plurality of bits;

in which operation of the circuit is synchronised to a clock signal and the circuit comprises means for processing one bit in one cycle of the clock signal.
2. The circuit of Claim 1 in which the signalling comprises signalling system No. 7 (SS7) Layer 2 .
3. The circuit of any above claim comprising processing means for processing SS7 Layer 2 messages, the processing means comprising a functional block for processing transmit and receive bits in successive operations.
4. The circuit of any above claim comprising a store divided into a number of buffers, each buffer for successively storing status information and data relating to successive time slots of the sequence of time slots allocated to a particular data channel.

This Sheet Has Been Replaced





5. The circuit of Claim 4 in which the number of buffers is less than the number of data channels comprised in the TDM data streams processed by the circuit; in which the circuit comprises channel allocation means for allocating a unique number to each data channel;

in which the circuit comprises means for translating the data channel numbers and for using the translated numbers for addressing the buffers.

6. The circuit of any one of Claims 4 and 5 comprising assembly means for accessing by means of a plurality of read accesses the status information and data associated with one of the time slots stored in one of the buffers;

in which the assembly means comprises means for assembling the status information and data for the processing thereof in one cycle of the clock signal.

7. The circuit of any one of Claims 4 to 6 comprising disassembly means for receiving from the processing means in one piece the current status information and data associated with one of the time slots;

the disassembly means comprising means for dividing the status information and data into parts and writing each part separately to one of the buffers of the store.

This Sheet Has Been Replaced





8. A circuit for processing SS7 Layer 2 signalling associated with one or more TDM data streams in which each data stream is shared between a plurality of data channels in which each data channel is allocated a sequence of time slots in the data stream; and in which each time slot comprises a plurality of bits;
- in which the circuit comprises means for sequentially processing the bits of a data channel;
- in which the circuit comprises processing means for processing a plurality of parameter bits simultaneously in a single step; in which the plurality of parameter bits comprise the data channel bit to be processed, the most recently processed bits from that data channel and a plurality of status bits.
9. The circuit of Claim 8 in which the signalling comprises signalling system No. 7 (SS7) Layer 2 .
10. The circuit of of any one of Claims 8 and 9 comprising processing means for processing SS7 Layer 2 messages, the processing means comprising a functional block for processing transmit and receive bits in successive operations.
11. The circuit of any one of Claims 7 to 10 comprising storing means for temporarily storing the plurality of parameter bits between processing steps.





12. The circuit of Claim 11 in which the storing means comprises a store divided into a number of buffers, each buffer for successively storing the plurality of parameter bits relating to successive time slots of the sequence of time slots allocated to a particular data channel.
13. The circuit of Claim 12 in which the number of buffers is less than the number of data channels comprised in the TDM data streams processed by the circuit; in which the circuit comprises channel allocation means for allocating a unique number to each data channel;
- in which the circuit comprises means for translating the data channel numbers and for using the translated numbers for addressing the buffers.
14. The circuit of any one of Claims 12 and 13 comprising assembly means for accessing by means of a plurality of read accesses the plurality of parameter bits associated with one of the time slots stored in one of the buffers;
- in which the assembly means comprises means for assembling the plurality of parameter bits for the processing thereof simultaneously in a single step.
15. The circuit of any one of Claims 12 to 14 comprising disassembly means for receiving from the processing means in one piece the plurality of current parameter bits





associated with one of the time slots;

the disassembly means comprising means for dividing the plurality of parameter bits into parts and writing each part separately to one of the buffers of the store.

16. The circuit of any above claim comprising a message store comprising a plurality of message buffers each for storage of a message associated with a particular data channel;

in which the number of message buffers is less than the number of data channels comprised in the TDM data streams processed by the circuit; in which each data channel is allocated a unique number;

in which the circuit comprises means for translating the data channel numbers and for using the translated numbers for addressing the message buffers.

17. The circuit of any above claim comprising one or more TDM oversampling means for oversampling the received TDM data streams.

18. The circuit of any above claim comprising media means for transmitting and receiving TDM media data.

This Sheet Has Been Replaced



19. A communications interface comprising the circuit of any above claim for transferring information between a TDM communications network and an Internet protocol communications network.
20. A communications interface comprising the circuit of any one of claims 1 to 18 for transferring information between a TDM communications network and an asynchronous transfer mode communications network.
21. The circuit of any above claim comprising a programmable logic device (PLD).
22. A communications system comprising the circuit of any above claim.



This Sheet Has Been Replaced

