Title
Implementation of a station/network interface for a CAMB tree network

Authors
Godbille, Kenneth A.
Huang, Hung Khei
Suda, Tatsuya

Publication Date
1991

Peer reviewed

Implementation of a Station/Network Interface for a CAMB Tree Network [1]

Technical Report 91-26

Kenneth A. Godbille, Hung Khei Huang, Tatsuya Suda
Department of Information and Computer Science
University of California, Irvine
Irvine, CA 92717

[1]This material is based upon work supported by the National Science Foundation under Grant No. NCR-8907909. This research is also in part supported by University of California MICRO program.

ABSTRACT.

Packet collisions and their resolution create a performance bottleneck in random-access LANs. A hardware solution to this problem is to use collision avoidance switches. These switches allow the implementation of random access protocols without the penalty of collisions among packets. An architecture based on collision avoidance is the CAMB (Collision Avoidance Multiple Broadcast) tree network, where concurrent broadcasts are possible.

This paper is a companion to an earlier report, "TTL Implementations of a CAMB Tree Switch," where a tree network architecture was described for two different implementations of a CAMB tree switch. In the pages that follow, a hardware implementation of the interface between the network stations and the packet switches is proposed. This implementation is based on the first switch design in the companion paper.

1. A GENERAL OVERVIEW.

The purpose of this section is to provide a simple model for the implementation of a CAMB interface board. Towards this end, the board circuitry has been partitioned into four distinct modules (see figure 1):

- the *Station Interface Unit* (SIU), which facilitates data transfer between the network station and the interface board;

- the *Microprocessor Control Unit* (MCU), which manages data movement, data buffering, and imposes MAC layer Tree Net protocol on transmission and reception;

- the *Transmission Unit* (TU), which accepts data from the MCU, queues the data, serializes it, generates *Cyclic Redundancy Code* (CRC) error detection bits for it, and transmits it as a packet to the tree switches;

- and the *Reception Unit* (RU), which receives incoming packets, determines their correctness, re-parallelizes them, and queues them for transfer to the MCU.

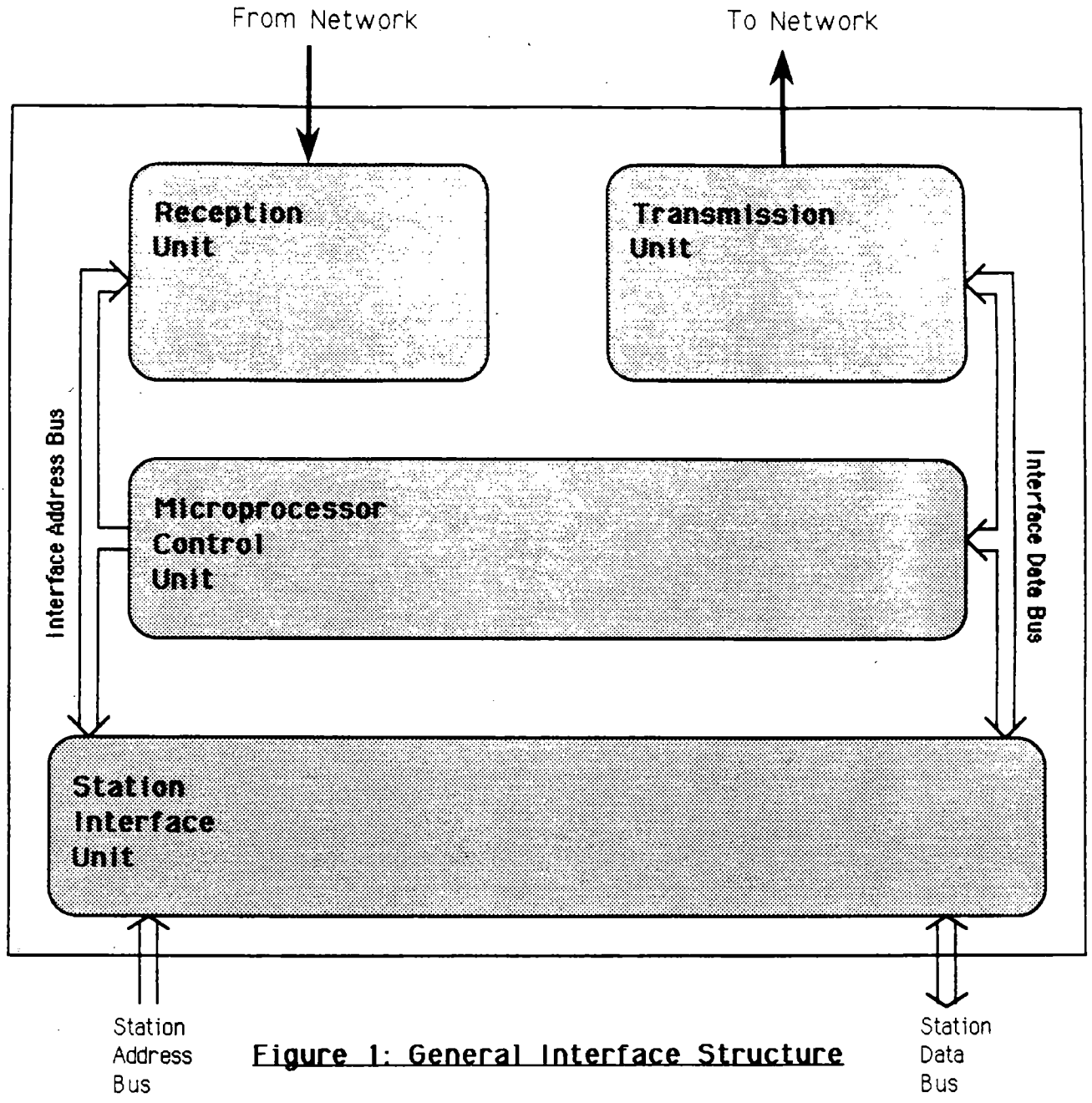


Figure 1: General Interface Structure

The four units which comprise the network interface work together to perform two concurrent functions: packet transmission, and packet reception. Following is a general description of these two operations.

1.1 General Data Flow: Transmission.

When a station wishes to send a message over the network, the station partitions and formats the message into one or more data packets. Then, the station sends each data packet to the interface board via the SIU.

The SIU may be thought of as two independent mailboxes. One mailbox belongs to the station, and the other mailbox belongs to the interface board. The station places a packet for transmission into the interface mailbox, and raises the electronic equivalent of a red mailbox flag. This flag signals the MCU that the interface mailbox is full.

The MCU responds to the raised interface mailbox flag by:

- re-lowering the interface mailbox flag;
- removing the packet from its box;
- and either queueing the packet if the TU is busy, or forwarding the packet directly to the TU for transmission otherwise.

Then, the MCU lets the station know it is ready for another packet by dropping a note in the *station* mailbox which says "Interface mailbox emptyed, ready for more mail!", and raising the station mailbox's flag.

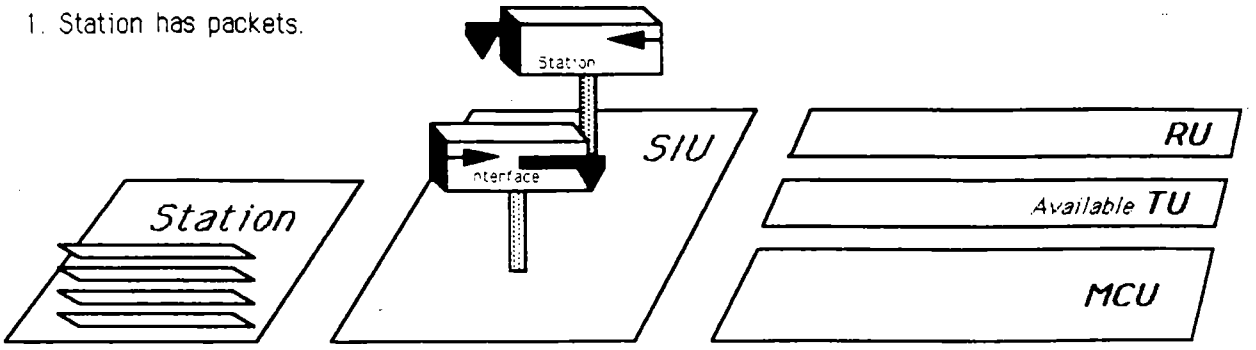
The station responds to its raised mailbox flag by echoing the mail fetching steps performed by the MCU:

- re-lowering the station mailbox flag,
- removing and processing the note in its box,
- and putting another packet into the interface mailbox if there is one available.

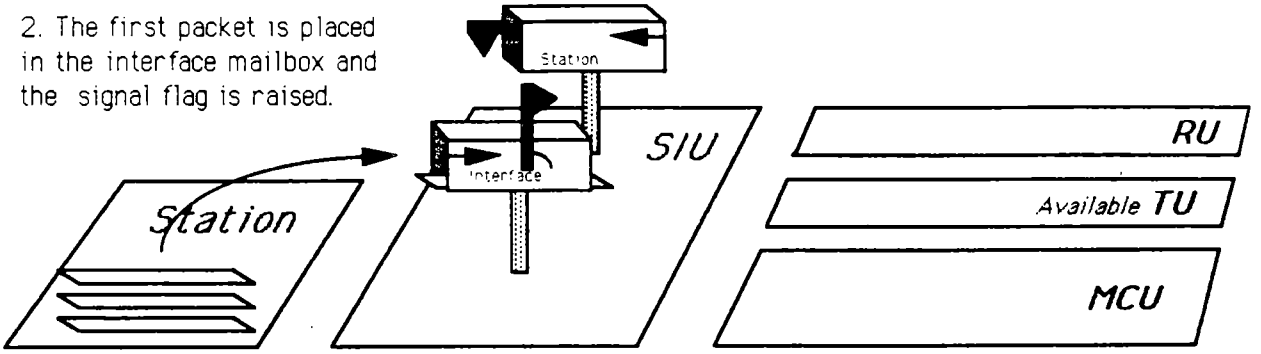
Thus, the station transfers a packet to the SIU, the MCU removes the packet from the SIU for transmission, and then signals the station it is ready for more packets.

The entire process is show below in figure 2.

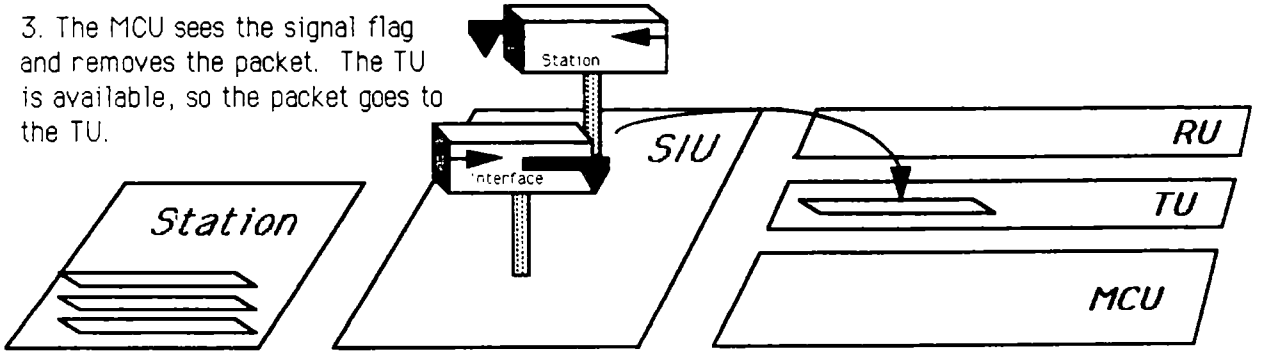
1. Station has packets.



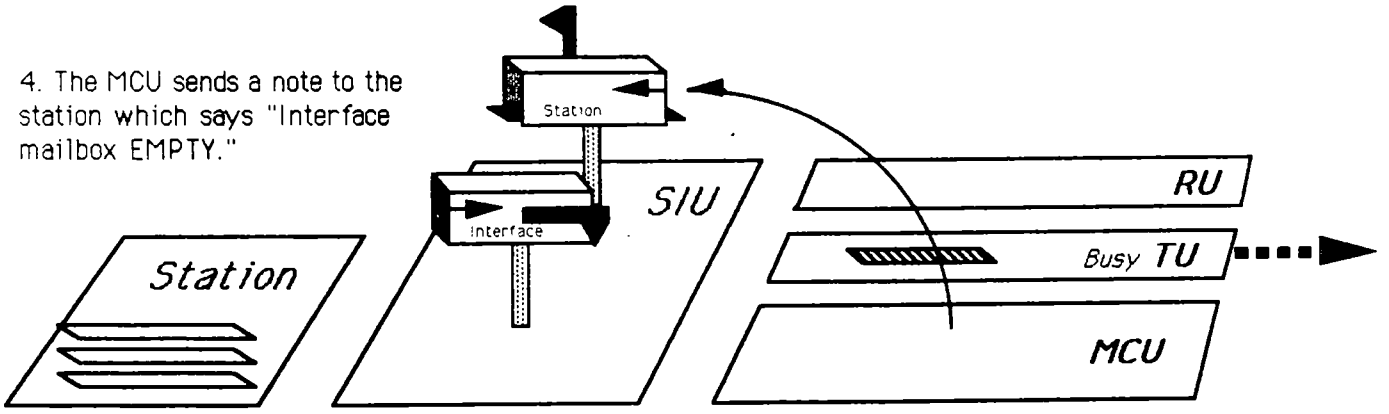
2. The first packet is placed in the interface mailbox and the signal flag is raised.



3. The MCU sees the signal flag and removes the packet. The TU is available, so the packet goes to the TU.



4. The MCU sends a note to the station which says "Interface mailbox EMPTY."



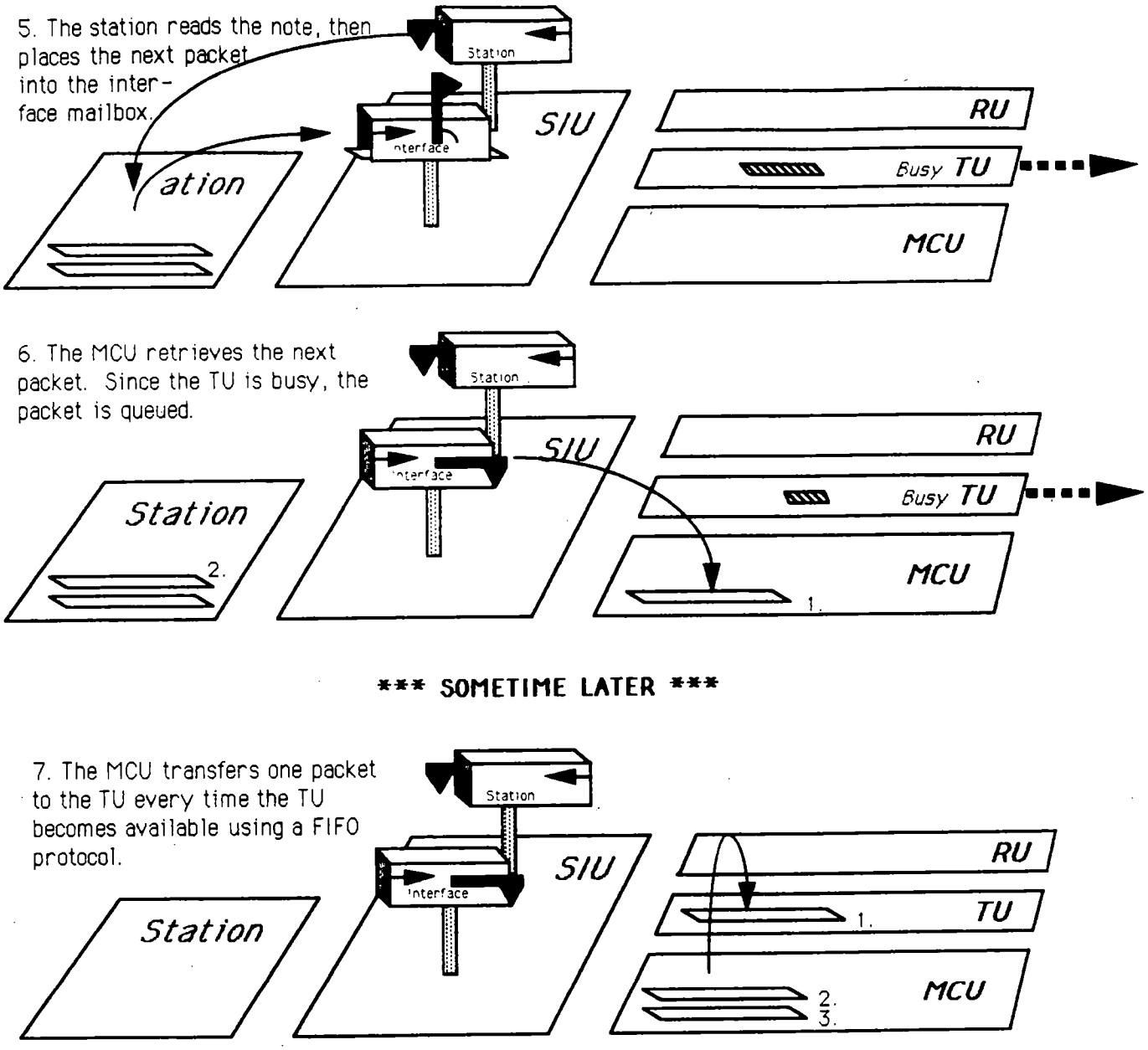


Figure 2: Transmission.

When a packet is fetched from the SIU by the MCU, the MCU transfers this packet into a FIFO transmission queue. The front of this queue is located within the TU, and is one packet in size. The rest of the queue is maintained within the MCU. The packet which resides in the TU is transmitted to the tree network. After a successful transmission, the front position becomes vacant and the next packet in line is moved from the MCU into the TU.

1.2 General Data Flow: Reception.

Packet reception is basically packet transmission in reverse. The only difference is that the MCU provides one additional function. The MCU gives to the station only those packets which are error free and correctly addressed.

When the RU starts to receive a packet form the network, it immediately signals the MCU to begin queueing and processing its output. The MCU is programmed to recognize which of four possible RU outputs apply:

• In the first case, the packet which arrives is the same packet the station just transmitted via the TU. In the CAMB tree network protocol, when a packet is broadcast, it is received by both the source *and* the destination stations. The MCU identifies a successful transmission when a broadcasted packet is echoed back by the network without error.

The MCU responds to receiving a complete and correctly echoed packet by incrementing the FIFO transmission queue and resetting the TU. If there are no more packets in this queue, the MCU places the next packet from the station into the queue for transmission.

If no echoed packet is forthcoming after the maximum round trip propagation time through the network, tree network protocol calls for the MCU to signal the TU to retransmit the packet in the TU until its echo is detected by the RU.

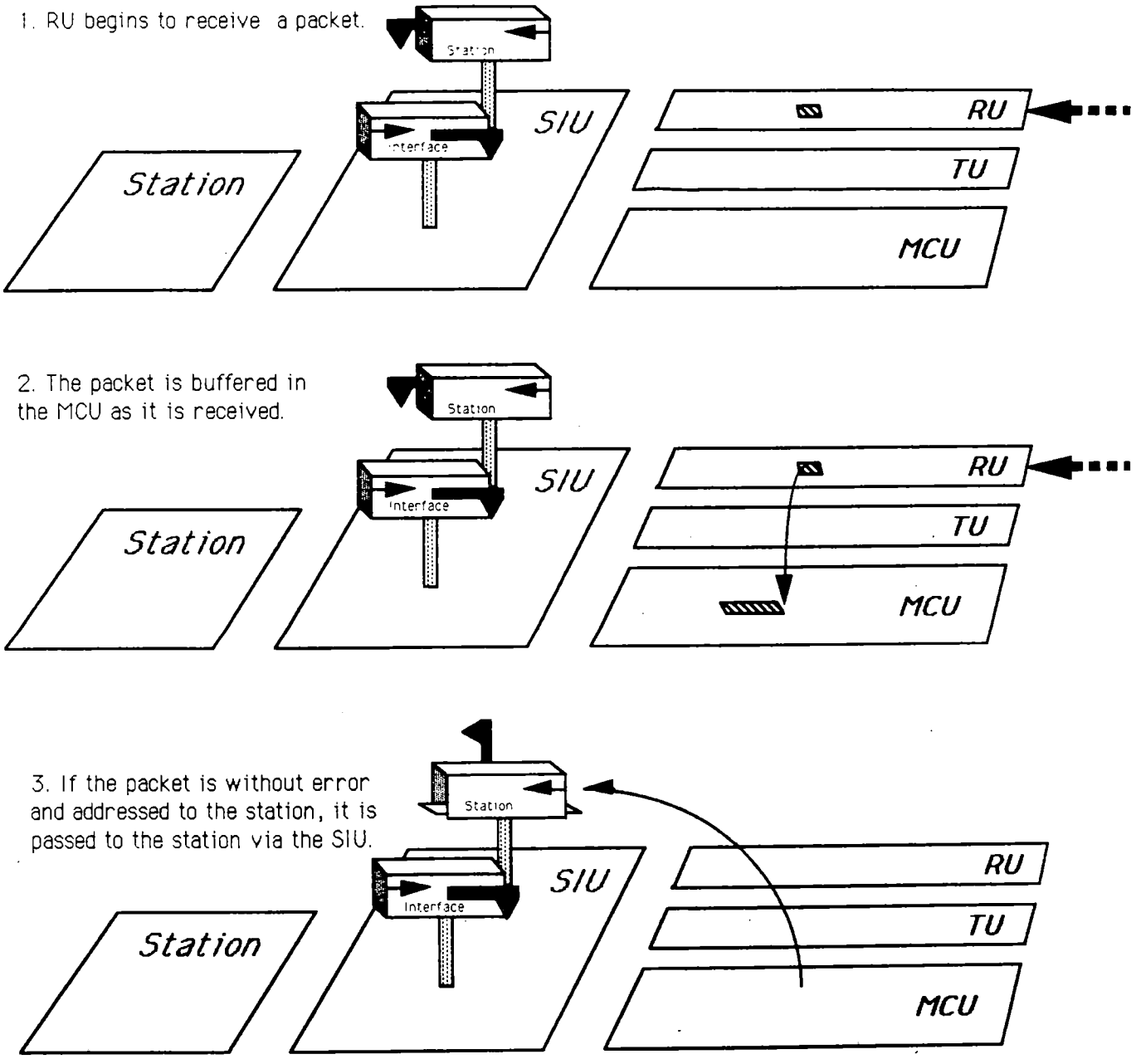
• In the second case, the MCU receives a complete and correct packet which is addressed to the station it serves. The MCU responds by placing this packet into a FIFO reception queue, located within the MCU. When the station mailbox becomes empty, the packet at the front of the reception queue is removed and transferred to the SIU. Then, the station mailbox signal flag is raised.

The station responds to this raised signal flag by re-lowering the flag and removing the arriving packet for processing. Then, the station places a "Station mailbox EMPTY" note in the interface mailbox and raises the interface mailbox flag. This note, when read by the MCU, signals that the station mailbox is ready to receive another packet.

• In the third case, the MCU receives a packet which is not addressed to the station it serves. In this case, the packet is discarded by the MCU.

• And in the fourth and final case, the MCU receives a packet which contains errors. In this case, as in the third case, the packet is discarded.

Figure 3 illustrates the above processes.



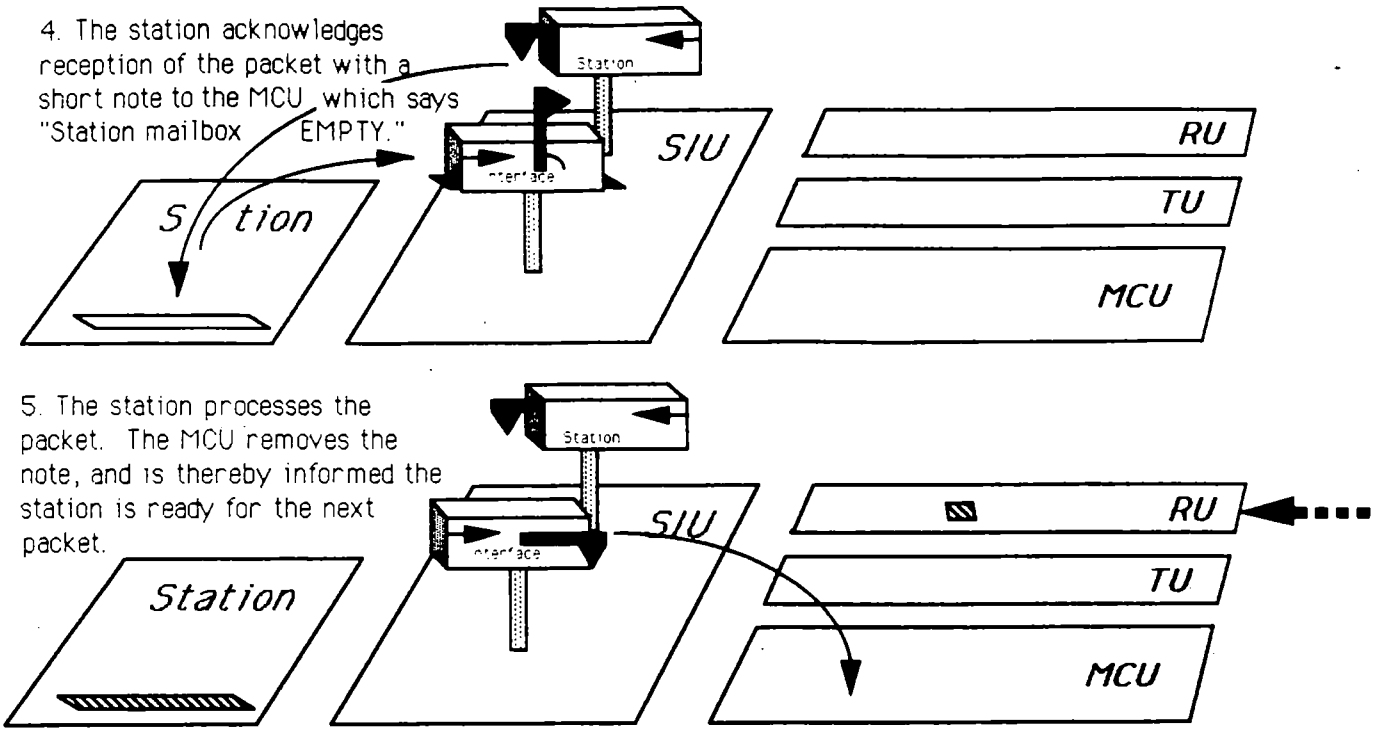


Figure 3: Reception.

Thus, packets are received from the network by the RU and stored in the MCU for processing. When an echoed packet from the TU is received by the MCU, the transmission queue is incremented and the packet is discarded. Likewise, if the MCU determines that the received packet is either erroneous or not addressed to the station the MCU serves, the packet is discarded. Packets which are both error free and addressed to the station the MCU serves are placed in an internal reception queue for transfer to the SIU and, consequently, to the host station.

2. The Interface Hardware: A Reference.

The following describes the implementation of the four modules defined in section 1. This description includes information specific to the second prototype board which is now under construction at the University of California at Irvine.

2.1 The Physical Connection.

The current prototype board is designed to interface with a PC-9801 edge connector (Figure 4). Table 1 shows the relevant signals defined for each of the 100

contacts of the connector. Signals which are used by the interface board are in bold type. (Note: the purpose of this table is not to give a complete definition of the PC-9801 standard, but rather to described those signals used by the interface board.)

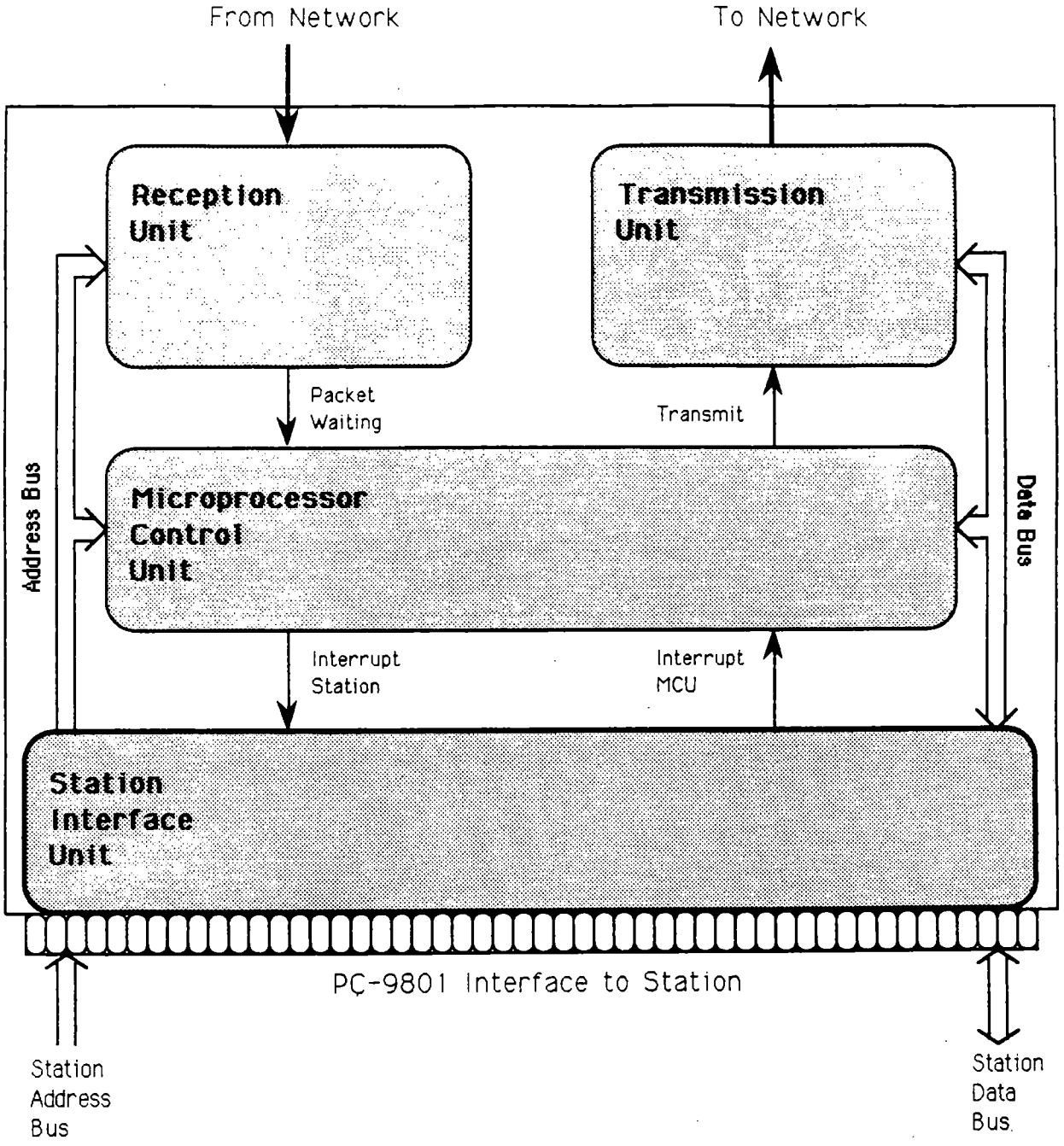


Figure 4: PC-9801 Interface.

Front Side	Signal Name	Signal Description	Back Side	Signal Name	Signal Description
A1	GND	- 5V	B1	GND	- 5V
A2	V1		B2	V1	
A3	V2		B3	V2	
A4	AB001	Station Address Bus, Bit 0 (LSB)	B4	DB001	Station Data Bus, Bit 0 (LSB)
A5	AB011	" " " " 1	B5	DB011	" " " " 1
A6	AB021	" " " " 2	B6	DB021	" " " " 2
A7	AB031	" " " " 3	B7	DB031	" " " " 3
A8	AB041	" " " " 4	B8	DB041	" " " " 4
A9	AB051	" " " " 5	B9	DB051	" " " " 5
A10	AB061	" " " " 6	B10	DB061	" " " " 6
A11	GND	- 5V	B11	GND	- 5V
A12	AB071	Station Address Bus, Bit 7	B12	DB071	Station Data Bus, Bit 7
A13	AB081	" " " " 8	B13	DB081	" " " " 8
A14	AB091	" " " " 9	B14	DB091	" " " " 9
A15	AB101	" " " " 10	B15	DB101	" " " " 10
A16	AB111	" " " " 11	B16	DB111	" " " " 11
A17	AB121	" " " " 12	B17	DB121	" " " " 12
A18	AB131	" " " " 13	B18	DB131	" " " " 13
A19	AB141	" " " " 14	B19	DB141	" " " " 14
A20	AB151	" " " " 15	B20	DB151	" " " " 15 (MSB)
A21	GND	- 5V	B21	GND	- 5V
A22	AB161	Station Address Bus, Bit 16	B22	+ 12V	
A23	AB171	" " " " 17	B23	+ 12V	
A24	AB181	" " " " 18	B24	IR31	
A25	AB191	" " " " 19 (MSB)	B25	IR51	
:	:	:	:	:	:
A29	AB231		B29	IR121	Interrupt #5 to the station's CPU.
A30	INT0		B30	IR131	
A31	GND	- 5V	B31	GND	- 5V
A32	IOCHK		B32	- 12V	
A33	IOR0		B33	- 12V	
A34	IOW0		B34	RESET0	This signal resets the interface board.
A35	MRC0	Station's READ Signal.	B35	DACK00	
A36	MWC0	Station's WRITE Signal.	B36	DACK30/20	
:	:	:	:	:	:
A41	GND	- 5V	B41	GND	- 5V
A42	CPUENB10		B42	RQGT0	
A43	RFSH0		B43	DMATC0	
A44	BHE0	Is LOW when station reads and writes 16-bit words at a time (Word Access), and BHE0=(AB001)' when the station reads and writes 8-bit bytes at a time (Byte Access).	B44	NMI0	
A45	IORDY1	I/O ready: signals the interface board is ready for a WRITE operation.	B45	MWE0	
:	:	:	:	:	:
A49	+ 5V	+ 5V	B49	+ 5V	+ 5V
A50	+ 5V	+ 5V	B50	+ 5V	+ 5V

Table 1: Relevant PC-9801 signals.

2.2 The Station Interface Unit (SIU).

The Station Interface Unit links directly with the PC-9801 connector. The circuit consists of the following components:

QTY	Device	Description	
1/6	74HC14	Hex inverters (NOT gates)	(CMOS)
1	74LS32	Quad 2-input OR gates	(TTL)
2	74LS244	Octal Buffer and Line Driver	(TTL)
2	74LS245	Octal Bus Transceiver with 3-state outputs	(TTL)
1	74LS520	8-bit Comparator	(TTL)
1	8421	Dual Port RAM Master	(CMOS)
1	8431	Dual Port RAM Slave	(CMOS)
5	Pullup Registers	(For 35 pullup resistors)	

Table 2: SIU Components.

Since the function of the DPRAM is critical to understanding the SIU, the next section is devoted entirely to explaining its operation.

2.2.1 The Dual Port RAM (DPRAM).

The dual port RAM is a specialized memory chip which has two separate sets of address and data lines, called *ports*. Each port, designated left and right, can access a common area of memory (2K bytes) contained within the DPRAM. Moreover, as long as both ports do not try a WRITE operation to the same memory cell, each port can access this memory simultaneously and asynchronously:

Right Port		Left Port		
Address	Operation	Address	Operation	Result
Any	READ	Any	READ	Allowed
Any	READ	Any	WRITE	Allowed
Any	WRITE	Any	READ	Allowed
address A	WRITE	address B (B≠A)	WRITE	Allowed
address A	WRITE	address A	WRITE	Not Allowed

Table 3: DPRAM Operation.

In other words, two READ operations on any two cells in the DPRAM will work, and one READ operation and one WRITE operation from either port will work. Two WRITE operations will also work *as long as the memory cells written to by each port are different.* In the case where two WRITEs are requested of the same memory cell, arbitration circuitry in the DPRAM signals one port to wait via the BUSY' signal. This unique architecture makes the DPRAM ideal for efficient communication between two different microprocessors.

A second feature of the DPRAM is an on chip interrupt unit. The last two memory locations in each DPRAM are very special. The first is located physically at DPRAM address $7FE. A WRITE operation to $7FE on the *left* port causes the interrupt line on the *right* port to go active (LOW). And, a READ operation on the *right* port of the same address, $7FE, deactivates the *right* port interrupt line (HIGH).

Likewise, a WRITE on the *right* port to the second special location, $7FF, forces the *left* port's interrupt line active (LOW). And, a READ on the *left* port at the address $7FF makes the *left* port interrupt line inactive (HIGH). This interrupt process is the mechanism behind the mailbox signal flag illustrated in section 1.

2.2.2 The SIU.

As shown in figure 5, the SIU has two sections.

In one section, the 74LS244s and 74LS245s serve to isolate the interface board's inputs from physical contact with the station's circuitry. This has the beneficial effect of reducing noise across the PC-9801 connection.

The second section consists of the Dual Port RAM (DPRAM) chips, the 74LS520 Comparator, and the 74LS32 Quad OR gates.

The DPRAMs and the logic implemented by these other two chips act as the mailbox described in section 1. The left port of the DPRAM is logically connected to the address and data lines belonging to the network station, while the right port is connected to the address and data lines belonging to the MCU. The 74LS520 associated with the left port is an 8-bit comparator used to map the 11 bit (2K) physical address of the DPRAM into a 20 bit address space on the station's bus. The

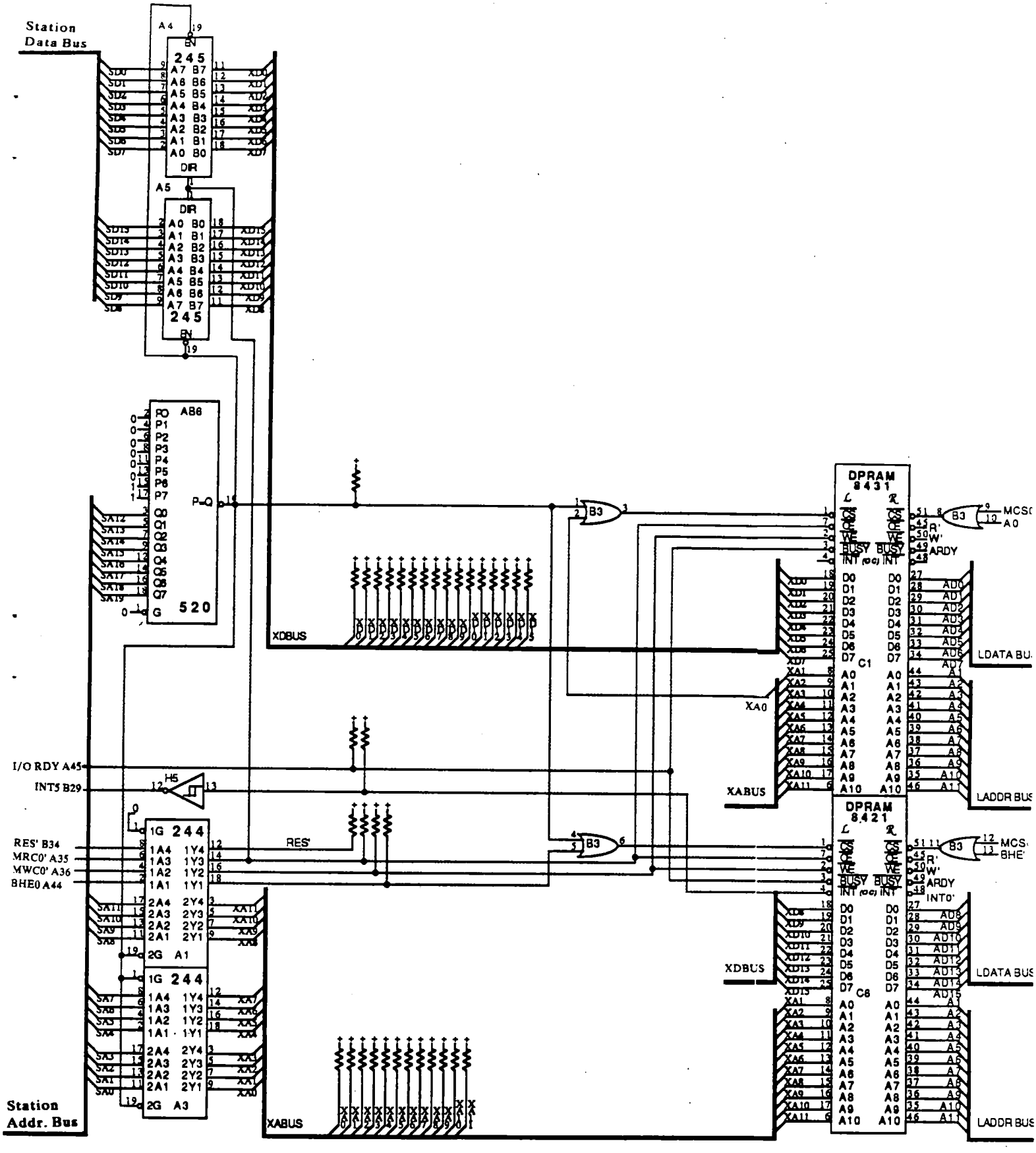


Figure 5: The Station Interface Unit.

two OR gates connected to the Chip Select line on the DPRAM are used to facilitate either byte or word access on the DPRAM.

The DPRAMs form a memory area 2K x 16 bits in size which is a part of both the station's memory map, and the interface board's memory map. By virtue of the DPRAM design, this 2K x 16 memory area is simultaneously and asynchronously accessible to both the station and the microprocessor on the interface card.

2.3 The Microprocessor Control Unit (MCU).

The Microprocessor Control Unit is the driving force behind the interface board. Shown in Figure 6, it consists of the following components:

QTY	Device	Description	
2	27C512	EPROM memories	(CMOS)
1/4	74HC02	Quad 2-input NOR gates	(CMOS)
4/6	74HC14	Hex inverters	(CMOS)
1	74HC32	Quad 2-input OR gates	(CMOS)
2	74HC573	Octal D Latches	(CMOS)
1	80C186	16-bit Microprocessor	(CMOS)
2	5256	Static RAM memories	(CMOS)
3	Capacitors	20p*f*, 30p*f*, and 10µ*f*	
1	Crystal	20MHz	
1	Diode		
7	Pullup Registers	(For 41 pullup resistors)	
1	Resistor	4.7KΩ	
1	SPST switch		

Table 4: MCU Components.

The heart of the MCU is the 80C186 microprocessor. By executing a small firmware code contained in the two 27C512 EPROMs, the 80C186 implements the control logic which drives all the other units. The 80C186 is a 16-bit microprocessor, operating at 10MHz, which contains the following features pertinent to the operation of the CAMB interface board:

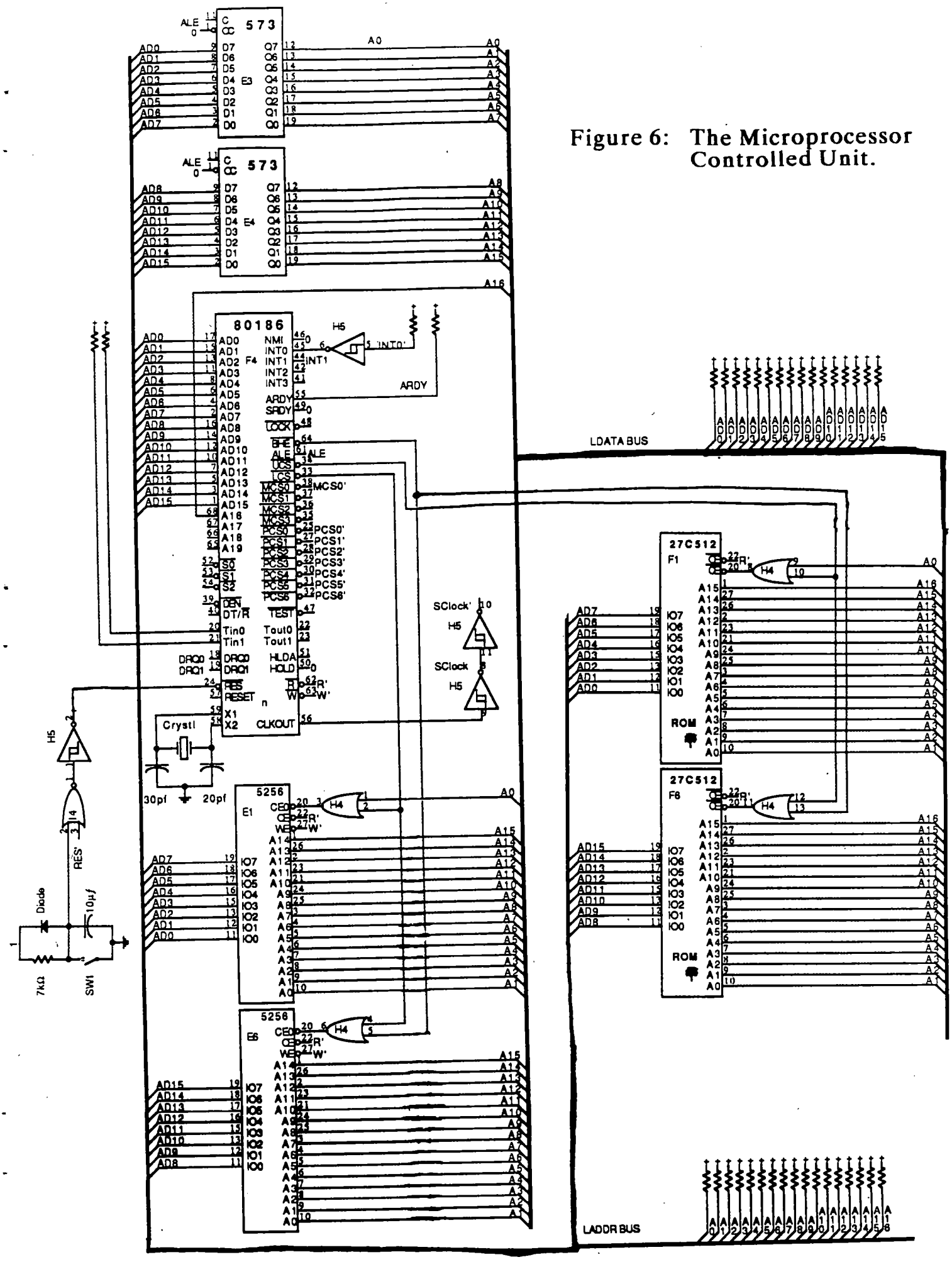


Figure 6: The Microprocessor Controlled Unit.

- two on-board DMA (Direct memory access) modules,
- a clock generator module,
- a programmable timer module,
- and, a programmable chip selection module.

The control signals generated by the 80C186 are summarized in the following table: (Note: entries suffixed by a single quote (') are indicative of an active LOW signal.)

CPU Pin	Signal Name	Type {*In,Out*}	Purpose
See Fig. 5	AD0-AD15, A16	I/O	Multiplexed address and data bus. Pin 61, Address Latch Enable (ALE) is used with the two 573's to form a separate address bus for the 80C186 microprocessor. AD1-AD15 and A16 generate the 16-bit address. AD0 distinguishes between two different data transfer modes: byte transfer mode and word transfer mode. See BHE' for a description of these two modes.
55	ARDY	I	Asynchronous Ready. When this pin is pulled LOW, the 80C186 bus enters a wait state until this signal returns HIGH.
18	DRQ0	I	Data ReQuest input for DMA unit 0. DMA unit 0 enters a wait state when this line goes LOW, and remains in this state until DRQ0 returns HIGH. This signal is generated when the TU's FIFO memory overflows.
19	DRQ1	I	Data ReQuest input for DMA unit 1. A LOW state on this line places DMA unit 1 in a wait state until DRQ1 returns HIGH. This signal is generated by the RU and indicates that the RU's FIFO memory is currently empty, or that one full packet has been transferred to the MCU.
45	INT0	I	Interrupt 0. This is an interrupt signal which tells the 80C186 the network station has left information in the DPRAM which needs to be processed.
44	INT1	I	Interrupt 1. This interrupt signal informs the 80C186 that a packet has just been transferred out of the Reception Unit by DMA unit 1 and is awaiting processing in the 80C186's RAM memory.
24	RES'	I	When this signal is LOW, the 80C186 is reset.
59, 58	X1, X2	I	These inputs are part of the CPU's internal clock generation circuit.
61	ALE	O	Address Latch Enable. The 80C186 uses AD0-AD15 as a multiplexed address and data bus. The address for every READ and WRITE cycle is latched by the two 573's in the MCU when the ALE signal goes LOW.

Table 5A: MCU Control Signals.

CPU Pin	Signal Name	Type {*In,Out*}	Purpose
64	BHE'	O	Bus High Enable. The 80C186 is capable of moving data in either 8 bits at a time, or 16 bits at a time. In 16 bit mode BHE'=A0=LOW. In 8 bit mode, 16 bit data is transferred in two 8 bit bytes distinguished as the *even byte* and the *odd byte*. BHE' and A0 are used as control signals to indicate which byte is to be transferred: BHE'=HIGH and A0= LOW for an even byte access, and BHE'=LOW and A0=HIGH for an odd byte access.
33	LCS'	O	Lower memory Chip Select. Enables the MCU's RAM chips for data transfer on the multiplexed address and data (AD) bus.
34	UCS'	O	Upper memory Chip Select. This signal is used to select the MCU's ROM chips for the multiplexed address and data (AD) bus.
38	MCS0'	O	Mid-range memory Chip Selection #0. This signal enables the DPRAM chips for data transfer.
25	PCS0'	O	Programmable Chip Select #0. This signal is activated under firmware control. It is used to reset the Transmission Unit and its packet buffer.
27	PCS1'	O	Programmable Chip Select #1. This signal is activated under firmware control. Once a packet is loaded into the TU's buffer, this signal starts the transmission of the packet to the network.
28	PCS2'	O	Programmable Chip Select #2. This signal is activated under firmware control. If a transmission is unsuccessful, i.e., an echo of the transmitted packet is not received, this signal is activated. This signal causes retransmission of the packet in the TU's buffer to the network.
29	PCS3'	O	Programmable Chip Select #3. This signal is activated under firmware control. It is used by the 80C186 to select the FIFO memory in the TU for data transfer.
30	PCS4'	O	Programmable Chip Select #4. This signal is activated under firmware control. It resets the Reception Unit and its packet buffer.
31	PCS5'	O	Programmable Chip Select #5. This signal is activated under firmware control. It is used to clear the interrupt (INT1) which occurs after a packet has been transferred out of the RU.
32	PCS6'	O	Programmable Chip Select #6. This signal is activated under firmware control. It is used by the 80C186 to select the packet reception FIFO memory chip for data transfer out of the RU.
62	R'	O	80C186 READ control signal.
63	W'	O	80C186 WRITE control signal.

Table 5B: MCU Control Signals.

In addition to the 80C186 CPU, the Microprocessor Control Unit contains two ROM chips, two RAM chips, and a few miscellaneous logic chips. The ROM chips contain the firmware algorithms which direct packet reception and packet transmission. The RAM chips provide temporary storage for the firmware algorithms, a transmission queue for the TU, and a reception queue for the RU.

2.4 The Transmission Unit (TU).

The Transmission Unit, shown in figure 7, accepts data from the MCU, queues the data, serializes it, generates CRC (cyclic redundancy code) error detection bits for it, and transmits it as packet to the tree switches. These functions are governed by roughly three components: a FIFO queue, a CRC generation circuit, and an output circuit. Table 6 below summarizes the parts of the TU circuit:

QTY	Device	Description	
2/4	74HC08	Quad 2-in AND gates	(CMOS)
1/6	74HC14	Hex inverters	(CMOS)
3/4	74HC32	Quad 2-input OR gates	(CMOS)
1/2	74HC74	Dual Flip-Flops w/ CLR, PRE	(CMOS)
1	74HC86	Quad 2-in XOR gates	(CMOS)
1	74HC157	Quad 2-in Mux	(CMOS)
1	74HC175	Octal D Latches	(CMOS)
2	74HC273	Octal Flip Flops w/ CLR	(CMOS)
1	75172	Line Driver	
1	IDT72103	Serial to Parallel FIFO memory	(CMOS)
1	Pulldown Resistor		
1/7	Pullup Register	(For 1 pullup resistor)	

Table 6: TU Components.

As the FIFO chip used in the TU is important to the understanding of how the TU functions, the next section briefly describes its operation.

2.4.1 The IDT72103 FIFO Chip.

The 72103 FIFO chip can be thought of as a RAM memory chip with two data ports, one port for input and one port for output. There are, however, two differences between the 72103 and this model. The first difference is there are no external address lines on the 72103. Addressing is taken care of by control logic located inside the 72103, and is implemented such that the first 9 bits written to the input port of the 72103 are the first 9 bits read from the output port of the 72103. The second difference is in the way bits can be stored in and read out of the 72103. The 72103 is capable of reading and/or writing data in both serial format and parallel format. I.e., the 72103 contains a both a serial-to-parallel converter, and a parallel-to-serial converter.

Tree Net Interface Transmission Circuit 9/3/90
D/B Ken Godbille and Hung Huang

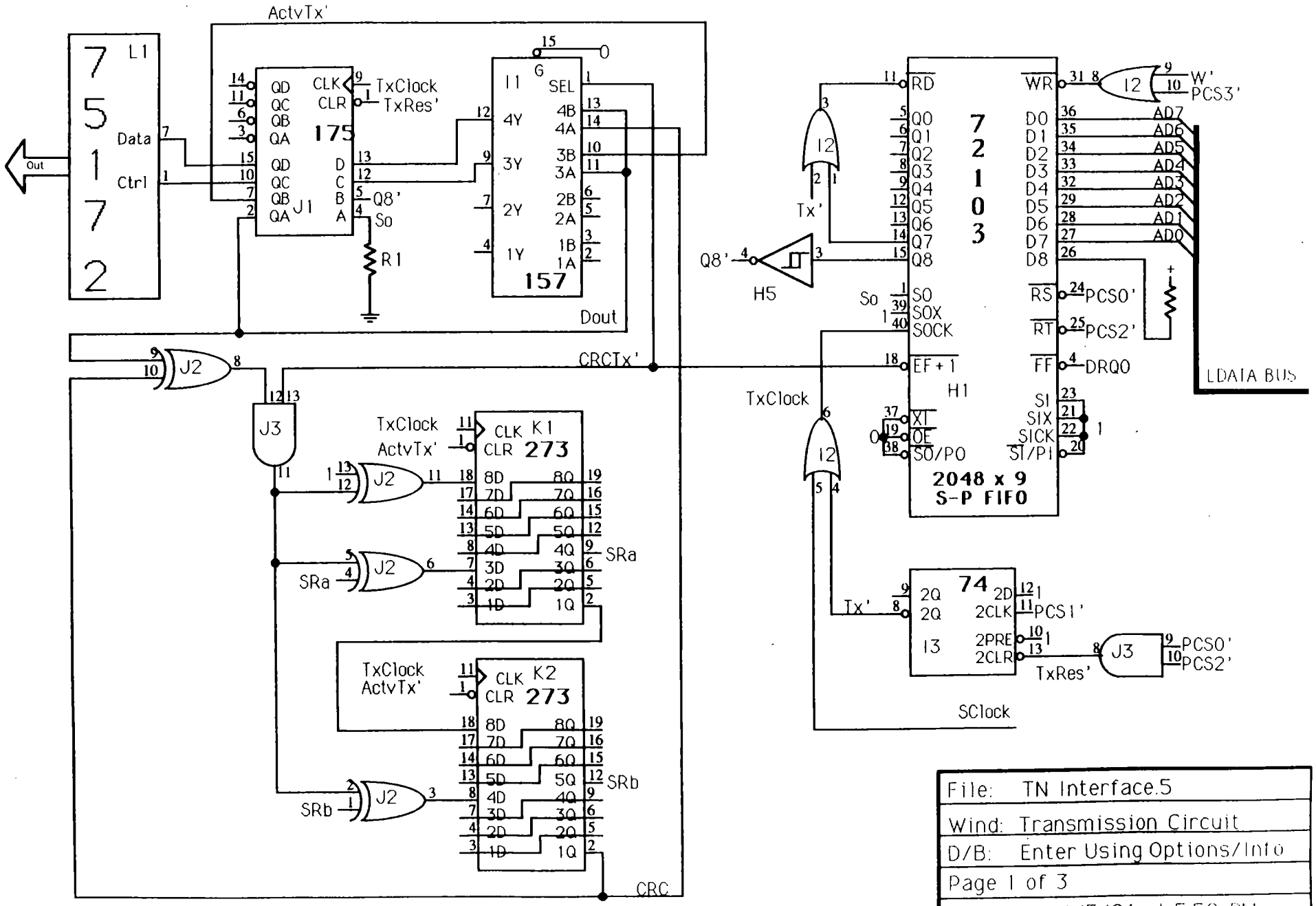


Figure 7: The Transmission Unit.

File:	TN Interface.5
Wind:	Transmission Circuit
D/B:	Enter Using Options/Info
Page 1 of 3	
Last Mod:	1/5/91 at 5:58 PM

Thus, in the case of the tree network interface, each 9 bit set of parallel data written to the input port of the 72103 can be converted to a serial data stream for output to the tree network. Table 6 describes the inputs and outputs of the 72103 pertinent to the operation of the TU. Definitions given under the heading "purpose" assume a parallel input, serial output configuration for the FIFO chip. (Note: entries suffixed by a single quote (') are indicative of an active LOW signal.)

FIFO Pin	Signal Name	Type {*In,Out*}	Purpose
See Fig.7	D0-D8	I	FIFO Data lines. These lines are connected to the low order bits of the multiplexed address and data (AD) bus of the 80C186.
11	RD'	I	FIFO memory READ line. When this line is LOW, the item at the top of the FIFO queue is transferred from the internal FIFO queue to an internal serial output register.
24	RS'	I	FIFO ReSet. Clears the internal FIFO queue.
25	RT'	I	FIFO ReTransmit. When this line goes LOW, it restores the FIFO memory queue to the state it was in immediately after the last packet was loaded. In this way, a packet can be re-read to the network.
38	S0'/P0	I	This signal selects the operating mode of the 72103. When this line is LOW, the chip generates a serial output on pin 1. Otherwise, the FIFO uses Q0-Q8 for a 9-bit parallel output.
20	SI'/PI	I	This signal selects the operating mode of the 72103. When this line is LOW, the chip expects input to be serial, using pin 23. In this case, however, this input is HIGH, indicating that the parallel input lines D0-D8 are to be used.
40	SOCK	I	Serial Output ClocK. Every cycle a new bit is shifted out of the Serial Output (Pin 1).
31	WR'	I	FIFO memory WRITE line. When this line goes from LOW to HIGH, 9 bits of data on D0-D8 are placed at the rear of the FIFO memory queue.
18	EF+1'	O	When there are two or less items in the 9 bit queue, this line goes low. This signal is used to determine when the CRC code is to be appended to the end of the packet.
4	FF'	O	FIFO Full Flag. This signal becomes LOW when the FIFO memory is full.
See Fig.7	Q0-Q8	O	In Serial Output mode, these becomes outputs from a digital delay line governed by SOCK.
1	SO	O	Serial Output. This output is connected to a shift register internal to the 72103. The shift register shifts out one bit of 9-bit parallel data every serial output clock cycle (pin 40).

Table 7: Pertainent FIFO Chip Signals

2.5 The Reception Unit (RU).

The Reception Unit (Figure 8) receives incoming packets, determines their correctness, re-parallelizes them, and queues them for transfer to the MCU. These functions are governed by roughly three components: an input circuit, a CRC error detector, a serial to parallel data converter, and a FIFO queue with control logic which notifies the MCU of a packet arrival. The components which implement these functions are listed below in table 8.

QTY	Device	Description	
3/4	74HC02	Quad 2-in NOR gates	(CMOS)
2/4	74HC08	Quad 2-in AND gates	(CMOS)
2	74HC30	8-input NAND gate	
5/4	74HC32	Quad 2-input OR gates	(CMOS)
1/2	74HC74	Dual Flip-Flops w/ CLR, PRE	(CMOS)
1	74HC86	Quad 2-in XOR gates	(CMOS)
1	74HC164	8 bit serial in/parallel out shift register	(CMOS)
1	74HC166	8 bit parallel or serial in/serial out shift register	(CMOS)
1	74HC175	Octal D Latches	(CMOS)
2	74HC273	Octal Flip Flops w/ CLR	(CMOS)
1	75173	Line Receiver	
1	IDT7200	FIFO memory	(CMOS)
3/7	Pullup Register	(For 3 pullup resistors)	

Table 8: RU Components.

Tree Net Interface Reception Circuit 11/1/90
D/B Ken Godbille and Hung Huang

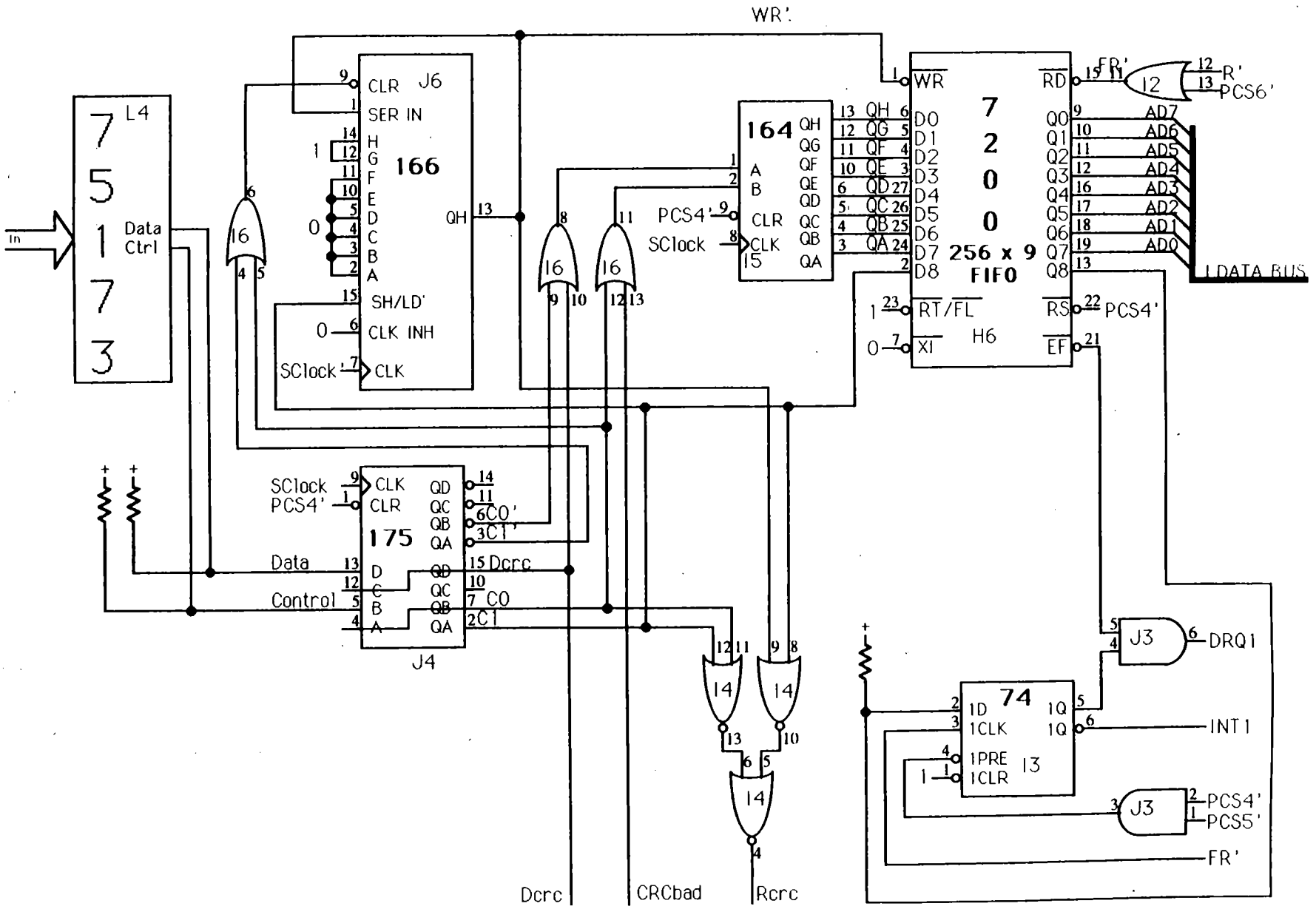


Figure 8A: The Reception Unit.

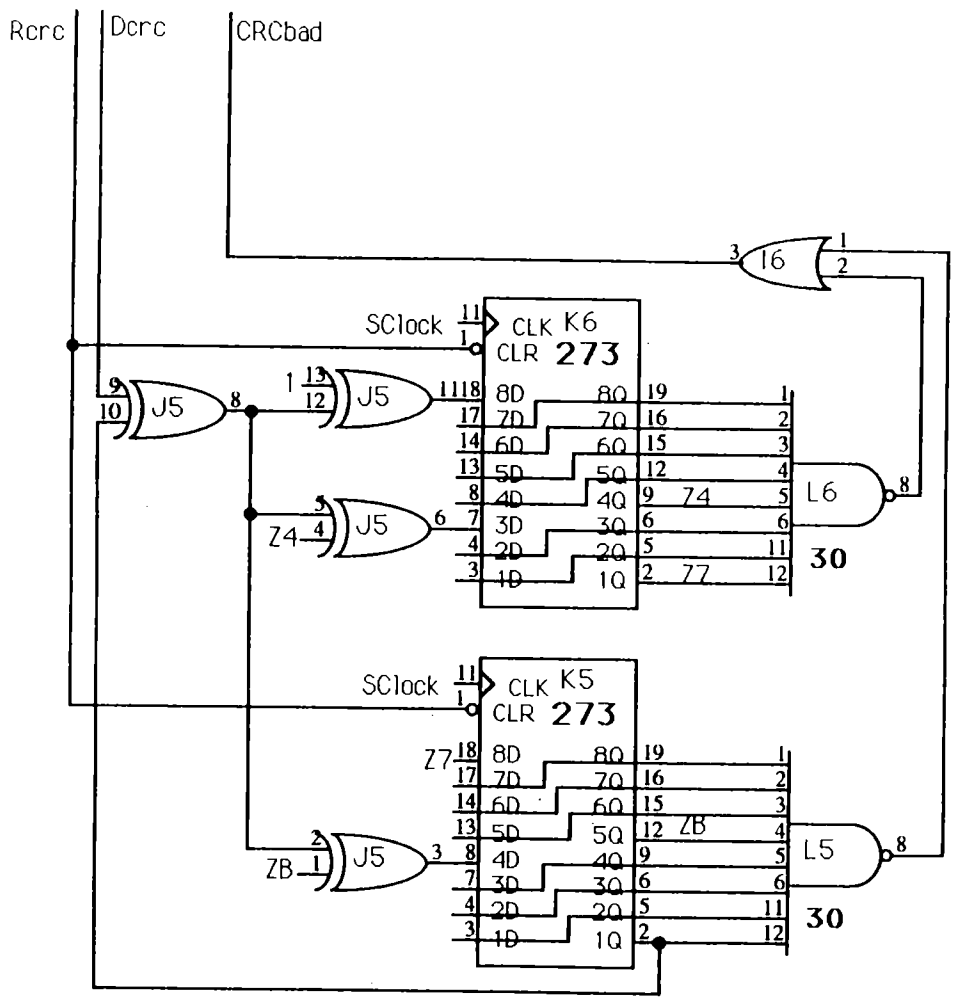


Figure 8B: The Reception Unit.

File: TN Interface.5
Wind: RU CRC Circuit
D/B: Enter Using Options/Info
Page 2 of 3
Last Mod: 1/5/91 at 5:58 PM

3. A TALE OF TWO PROCESSES.

As stated in the general model of the CAMB tree network interface, the interface hardware implements two concurrent processes: packet transmission, and packet reception. These processes are both interrupt driven. In other words, the appropriate process is triggered when either a packet is left in the DPRAM and the right port interrupt goes active, or when the RU signals that a packet is being received.

The goal of the next two sections is to describe how these two processes operate at the hardware level. This is accomplished by example. The information is presented in a outline format to enhance comprehension of interface board function. For reference, figure 10 gives the complete hardware circuit.

3.1. The Transmission Process.

I. The transmission process starts when the network station writes a packet of data into the dual port RAM chip in the SIU.

II. The software processes running at the station and on the interface board partition the DPRAM into two separate mailboxes of 1K x 16 bits each: one for the station, and one for the CAMB network interface. Each mailbox is designed to hold one packet and one word of status information. Note that this means that each packet can be at most 1023 words in size.

In the case of sending a packet from the station to the interface board, the network station performs the following steps:

A. The packet is written to the DPRAM partition corresponding to the interface mailbox.

B. A word of status information is written to the DPRAM address $7FF. This action forces the raising of the interface "mailbox flag": the right port's interrupt line, INT0', goes LOW.

C. The LOW on INT0' interrupts the MCU's microprocessor. This interrupt starts the transmission algorithm. This algorithm causes the following to occur:

1. Using MCS0' to address the DPRAM, the 80C186 reads the status word at $7FF, thereby clearing the interrupt signal INT0'.

2. The status word tells the transmission process that there is a packet in the "interface mailbox." The software process acts on this information by transferring the packet to either the TU if the TU is not busy, or to the MCU's internal RAM for queueing otherwise.

3. Finally, it writes a status byte in the station mailbox area of the DPRAM, saying that the interface mailbox is available. The status byte is written to DPRAM address $7FE.

4. This WRITE forces the left port DPRAM interrupt line, INT5, active.

D. The station responds to INT5 active by starting the reception software process at the station. This process reads the status byte left by the MCU.

E. This read at DPRAM address $7FE deactivates INT5.

F. After a brief analysis of the status byte just read, the station reception process informs the station transmission process that the DPRAM memory is available to receive another packet.

IIIa.

- Assume for purpose of illustration that the TU was busy transmitting a previously received packet, and consequently, the packet just received from the station was queued in the MCU's internal RAM. This 32K x 16 bit RAM, which is formed by the two 5256 chips, is partitioned into three regions. The first region is small and contains variables used by the firmware program running on the 80C186. The second and third regions are equal in size and form two separate queues: one for packets waiting to be transmitted, and one for packets waiting to be passed to the station.

- If a packet is queued in the MCU's internal RAM, this means the TU was busy transmitting another packet. In this case, the 80C186 is programmed to continue to queue subsequent packets given to it by the station in the transmission partition of its 32K x 16 bit RAM until this RAM becomes full. If this RAM should become full, the 80C186 waits until there is space in the RAM before sending the "interface mailbox EMPTY" message to the station.

- When the TU becomes available, the 80C186 resets the TU using one of its programmable chip select lines, PCS0'. The PCS0' signal is connected to the reset line of the 72103 to clear the last packet transmitted form the 72103's FIFO memory. In addition, this signal is connected to a 74LS74 flip flop which is used to stop the transmission clock-- when the transmission clock is restarted later, the contents of the 72103 FIFO are processed and sent over the network.

- After clearing the 72103, the 80C186 programs DMA unit 0 to promote the next packet in the transmission queue to the TU. PCS3' is used to address the FIFO chip.

IIIb.

- Assume for the purpose of illustration that the TU was available. Here, the 80C186 uses one of its programmable chip select lines, PCS0', to reset the 72103 FIFO queue.

- Then, the 80C186 employs the AD bus to transfer the packet in the DPRAM directly to the 72103 FIFO queue. PCS3' is used by the 80C186 to address the 72103.

IV. Thusly, the 80C186 uses DMA unit 0 to transfer the packet to the 72103 FIFO chip either directly from the DPRAM or from the queue maintained in the 80C186's RAM.

V. Two bytes with all bits set are added to the end of the packet to act as a frame into which a CRC error detection code will later be placed.

VI. Next, the 80C186 pulses another programmable chip select line, PCS1'. This line starts the transmission clock, represented by the signal TxClock. When TxClock starts oscillating, the data stored in the 72103 FIFO is output from SO (pin 1) as a serial stream.

The data in this stream generates two signals which are transmitted over the network. These two signals are used to convey both data and switching information to the network. The first signal, called the data signal, is a serial representation of the bytes of packet data stored in the 72103 plus a two byte CRC error detection code added to the end. The second signal, called the control signal, is always HIGH when the data signal is active, and LOW when no packet is being transmitted. Figure 9 illustrates these two signals.

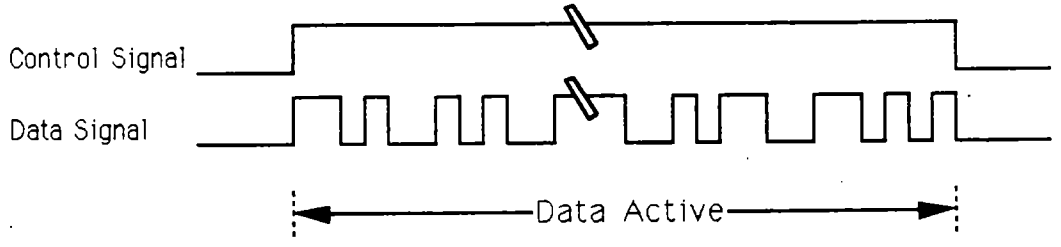


Figure 9: Tree Network Signals.

VII. When a packet is transmitted by the TU, the TU moves between two distinct states. These two states are controlled by the EF+1' signal on the 72103. This signal is LOW when there are less than two bytes in the 72103's queue, and HIGH when there are more than two bytes in the 72103's queue. After a packet is loaded, EF+1' is initially HIGH.

VIII. As transmission takes place, the CRC generation circuit, which consists of 16 flip-flops, four XOR gates, and an AND gate, calculates a 16 bit CRC error code on the fly as the data is moved out of the TU.

IX. After a time, all but the last two bytes of the packet have been transmitted to the network: EF+1' goes LOW. This causes the TU to switch states for the last two bytes of the transmission. In this new state, the 74HC157 allows 16 bits of error detection code to be shifted out of the CRC generation circuit, replacing the two bytes of data in the FIFO. Also in this new state, the two bytes in the FIFO become the control signal. The transmission ends when the last bit of these two bytes is shifted out to the network.

X. Meanwhile, while the TU is transmitting, the firmware transmission process is using one of the 80C186's three programmable timer circuits. This timer is set to (the maximum propagation time of a packet through the tree network) + (the maximum processing time of the MCU to identify a newly received packet). When the timer runs out, the firmware transmission process checks with the reception process to see if the packet just transmitted is being received. If it is, the transmission process will wait until the entire packet is received correctly. Then, it will look for a new packet to transmit. Otherwise, PCS2' from the 80C186 is activated. This resets the packet read pointer in the 72103 FIFO to the beginning of the packet. Then, the PCS1' signal is used to restart the packet transmission from the beginning.

3.2 The Reception Process.

I. The reception of packets starts at the RU. When the control signal is LOW, all the parts of the reception circuit are held in an initial state. In particular, INT1 and DRQ1 are inactive.

II. In the MCU, the reception process waits, having previously initialized DMA unit 1 in the 80C186 to fetch three bytes from the reception FIFO as soon as DRQ1 becomes active. Until DRQ1 becomes active, the DMA unit is held in a wait state.

III. When the control signal goes HIGH, the RU is set into motion. The CRC checking circuit starts processing the incoming data signal for errors, the 74HC164 starts shifting in the data packet for 8 bit parallel transfer to the 7200 FIFO, and 74HC166 starts generating a WRITE signal (WR') every 8 cycles to transfer this data from the 74HC164 to the 7200 FIFO.

IV. Every time 8 bits of data are transferred to the 7200 FIFO, an extra 9th bit is written as well. As long as the 8 bits transferred to the FIFO are not the last byte from the data signal, this 9th bit is HIGH.

V. When the first byte of the incoming packet is written to the FIFO, the Empty Flag (EF') signal on the FIFO chip goes HIGH. This forces DRQ1 HIGH, which allows DMA unit 1 to transfer a byte to the MCU's RAM.

VI. After three such transfers, the DMA unit wakes the reception process to examine these first three bytes of the incoming packet. The three bytes represent the destination address of the packet, the source address of the packet, and a 7-bit identity code. The MCU analyzes these bits to determine what to do with the packet arriving.

   A. If the destination address of the packet matches that of the station the MCU serves, the packet is queued in the MCU's RAM for transfer to the station.

   B. If the source address matches that of the station the MCU serves, the reception process tells the transmission process its packet is being received.

   C. If neither the source address or destination address of the packet matches that of the station the MCU serves, then the incoming packet is marked to be deleted from the reception buffer after transmission is complete.

VII. Having established how the incoming packet is to be handled, the reception process reprograms DMA unit 1 on the 80C186 to transfer an infinite number of bytes to the reception queue maintained in the MCU's RAM.

VIII. When the control signal drops to LOW, the RU writes the last 8 bits of the packet to the 7200. However, on this write, the last data bit does not come from the data signal, but rather from the CRC checking circuit. When this last bit is LOW, it indicates there was an error in the packet. In addition, the 9th bit is written as LOW. The LOW 9th bit indicates that the byte just written is the last byte in the packet.

IX. The reading of the last byte of data from the RU FIFO causes the 74HC74 flip-flop to change states. This state change causes INT1 to go active, and forces DRQ1 LOW. Consequently, the 80C186 is interrupted, and DMA unit 1 on the 80C186 enters a wait state.

X. The reception process responds to the INT1 signal by examining the last bit of the transmission: the bit from the CRC checking circuit.

   A. If the bit is HIGH, i.e. the transmission was error free the reception process performs the appropriate action based on the information gathered earlier from the first three bytes of the transmission.

   B. Otherwise, the newly-arrived packet is discarded.

XI. In the case of sending a packet form the interface board to the station, the MCU writes the packet from its internal buffer into the DPRAM partition corresponding to the station mailbox.

XII. Then, a word of status information is written to DPRAM address $7FE. This action implements the raising of the station "mailbox flag." The write forces INT5 active, interrupting the station reception software process.

XIII. In response, the software process reads the status word in the DPRAM at $7FE, thereby clearing the interrupt, and removes the packet from the DPRAM.

XIV. Finally, the station writes a status byte of its own in location $7FF to tell the interface board that it has emptied its mailbox.

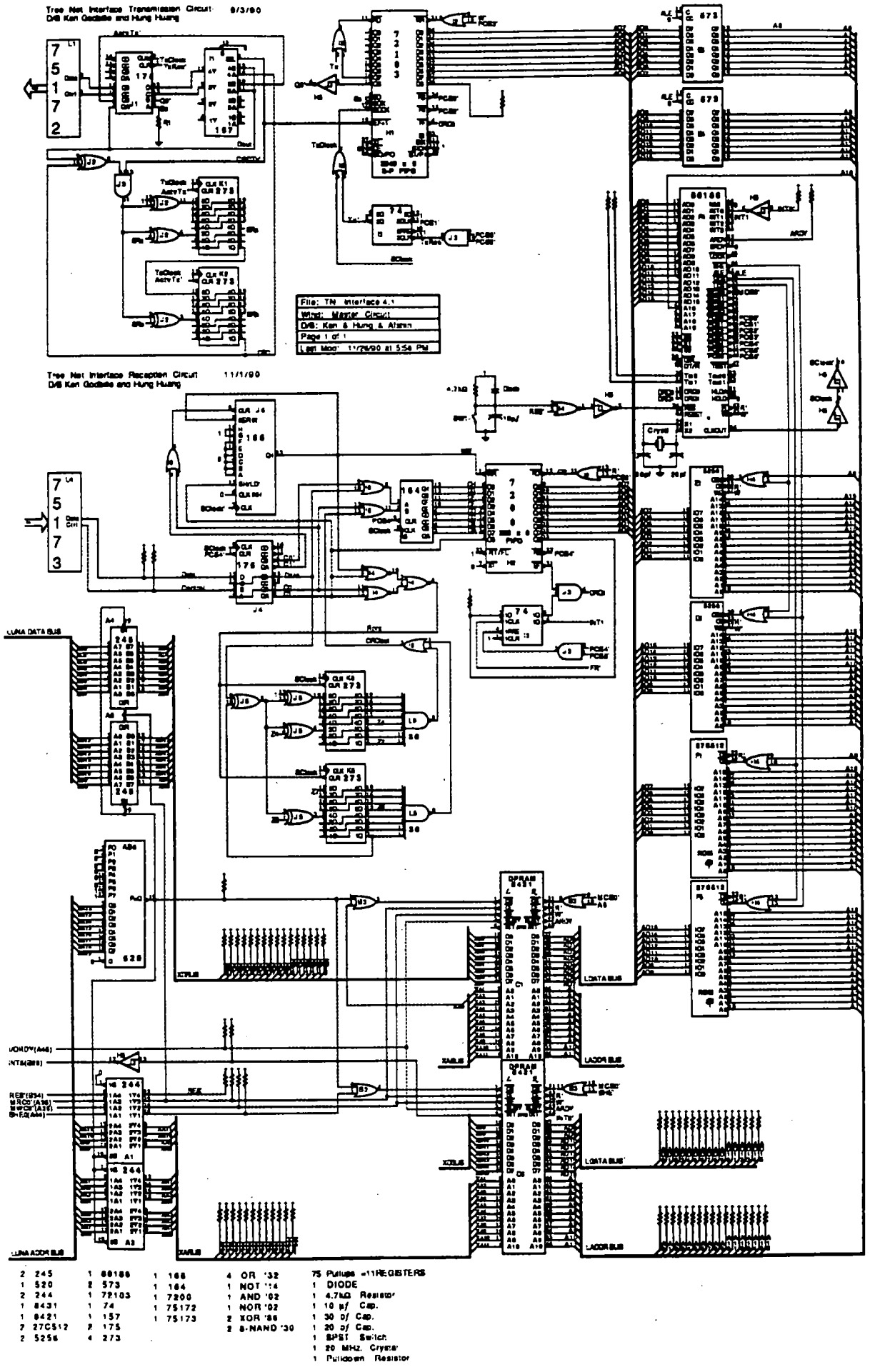


Figure 10: The Complete Interface.