Application for a Standard Patent or
A Standard Patent of Addition

We, ITT CORPORATION, of 320 Park Avenue, New York 10022, State of New York, United States of America, hereby apply for the grant of a standard patent/standard patent of addition for an invention entitled
"ADAPTIVE INTERFACE FOR MICROCOMPUTERS"

which is described in the accompanying provisional/complete specification.
Details of basic application(s) —
Number of basic application 682,035
Name of Convention country in which basic application was filed United States of America
Date of basic application 14 December, 1984

Our address for service is:

PATENT DEPARTMENT,
STANDARD TELEPHONES AND CABLES PTY. LIMITED,
252-280 BOTANY ROAD,
ALEXANDRIA, N.S.W. 2015.
AUSTRALIA.

Dated this Sixth day of December 1985

ITT CORPORATION
FORM 8
COMMONWEALTH OF AUSTRALIA
PATENTS ACT 1952-1969

DECLARATION IN SUPPORT OF A CONVENTION
APPLICATION FOR A PATENT OR
A PATENT OF ADDITION

In support of the Convention application made for a patent or addition for an invention entitled
"ADAPTIVE INTERFACE FOR MICROCOMPUTER"

I, PATRICK MICHAEL CONRICK,
of Standard Telephones and Cables Pty. Limited, 252-280 Botany Road, Alexandria, 2015, N.S.W., Australia, do solemnly and sincerely declare as follows:

1. I am authorised by ITT CORPORATION, the applicant for the patent or addition to make this declaration on its behalf.

2. The basic application as defined by Section 141 of the Act was made in United States of America on 14th December, 1984 by IYENGAR NARAYANAN KRISHNAN and HERBERT JOSEPH TOEGEL.

3. IYENGAR NARAYANAN KRISHNAN, of 39 Benjamin Heights Drive, Milford, Connecticut 06460, United States of America and HERBERT JOSEPH TOEGEL, of 77 Cross Road, Middlebury, Connecticut 06762, United States of America are the actual inventors of the invention, and the facts upon which the Applicant is entitled to make the application are as follows:

   ITT CORPORATION is the Assignee of the said inventors.

4. The basic application referred to in paragraph 2 of this Declaration was the first application in a Convention country in respect of the invention the subject of the application.

Declared at Sydney this 6th day of December 1985

ITT CORPORATION

[Signature]
Declarant

To: The Commissioner of Patents
An interface for use with a microcomputer, said interface comprising a first means for transporting data between a first port and a second port, said first means including a data storage medium, a second means for transporting data between said first port and said second port, said second means including an address storage medium, and means for selecting either said first means or said second means for transporting data across said interface.
The following statement is a full description of this invention, including the best method of performing it known to us:

state, state J' is used to update the registers 62. The following transition to state G1 is effectively suppressed.
This invention generally relates to an adaptive interface for use with a microcomputer and, in particular, relates to such an interface having means for selecting the operating mode thereof.

The advent of the microcomputer has revolutionized many industries, not the least of which is the telecommunication industry. The telecommunication industry has been effected first by the increasing number of subscriber devices to which service must be extended; and second, by the available microcomputers that can be integrated into the present and future switching systems to provide the services demanded.

Historically, telecommunication switching systems have been centralized, i.e. a central computer is provided to establish all communication paths, execute all protocol conversions and provide all maintenance and monitoring services. However, the centralized architecture has, inter alia, the severe drawback of a total exchange failure affecting all subscribers in the event of component failure in the central computer. In addition, in order to update or modernize such a central exchange a very costly re-design of the entire network is usually necessary. Clearly, such centralized systems are generally unable to maintain pace with technological advancements such as the microcomputer.

The fear of a total exchange failure lead to the development of a distributively controlled switching network wherein a digital switching network is provided with a plurality of autonomous terminal interfaces. The autonomous terminal interfaces are then connected to a plurality of subscribers. In such a distributively controlled network the terminal interfaces are provided, usually via a microcomputer, with the command and control instructions necessary to establish communication links between subscribers of different terminal interfaces as well as among subscribers.
to the same terminal interface. Thus, the fear was dispelled since any failure at any terminal interface produced only a minimal effect with regard to the overall subscribers to the network. One such distributively controlled switching network is the ITT "System 12" Digital Exchange.

Nevertheless, as the development of microcomputers and switching systems continues there is an increased need for an interface to a microcomputer that can be adapted to permit microcomputers of different capabilities to be interconnected to, for example, a switching apparatus.

Accordingly, it is one object of the present invention to provide an adaptive interface for use with a microcomputer.

This object is accomplished, at least in part, by an adaptive interface having means for selecting the data transfer operating mode thereof.

Other objects and advantages will become apparent to those skilled in the art from the following detailed description read in conjunction with the drawings attached hereto of which:

Fig. 1 is a block diagram of an interface embodying the principles of the present invention;

Fig. 2 is a block diagram of the storage medium shown in Fig. 1;

Fig. 3 is a block diagram of an interface controller adapted for use as shown in Fig. 1;

Fig. 4 is a state diagram for a sequencer adapted for use as shown in Fig. 3.

An adaptive interface, generally indicated at 10 in Fig. 1 and embodying the principles of the present invention, includes an input buffer 12, an output buffer 14, a storage medium 16 and an interface controller 18. The interface 10 further includes means 20 for selecting the operating mode thereof.

The claims defining the invention are as follows:
In one embodiment, the interface 10, as shown in Fig. 1, is interconnected, via a first port 22 thereof, to an apparatus 24 for establishing communication paths and, via a second port 26 thereof, to a microcomputer 28. The apparatus 24 includes a plurality of serial ports 30 and an intelligent switch 32 having a memory 34 and a switch controller 36. The ports 30 and the switch 32 are interconnected via a time-division-multiplexed (TDM) bus 38. One such apparatus 24 is described and discussed in co-pending Australian Patent Application No.

As fully discussed therein, in a particular embodiment, the transport of information along the TDM bus 38 is organized by frames. Each frame is subdivided into thirty-two channels each of which is further subdivided into sixteen words. Hence, there are 512 words per frame, each word, preferably, being represented by 16 bits of information. The movement of information via a TDM bus 38 is known and need not be discussed in detail other than to point out that such an apparatus 24 includes a master clock, not shown in the drawings, that provides strobe signals to the various destinations to indicate the arrival, i.e. prepare for, of the appropriate word, i.e. time slot, having information therefor. For reasons more fully discussed below, in the preferred embodiment, the switch controller 36 is adapted to provide system specific information to the interface controller 18 via, for example, an eight bit bus 39.

As used herein the phrase "intelligent switch" or the idiomatic equivalent thereof is taken to mean a device adapted to dynamically assign, or switch, a plurality of communication paths. One particular embodiment of the intelligent switch 32 includes a scratch pad memory 34 of the type described and discussed in co-pending Australian Patent Application No.

8. An interface as claimed in claim 1, wherein said data is one seg-
controlling, shown herein as controller 36 the path assignments in the scratch pad memory 34. One switch controller 36 particularly adapted to operate with the scratch pad memory 34 shown in the immediately above mentioned patent application is fully described in co-pending Australian Patent Application No. 36

The microcomputer 28, in one embodiment, includes a microprocessor portion 40, a random access memory 42 (RAM) and a read only memory 44 (ROM) interconnected via a local bus 46. Generally, microcomputers can be classified, inter alia, by the ability thereof to accept "direct memory access" handshake signals from other intelligent devices to allow, for example, the direct access to the microcomputers random-access-memory by the other intelligent device. The microcomputers of this classification can thus be used to transfer data rapidly by using known "direct memory access" techniques. Preferably the microcomputers chosen, in this classification, are sixteen bit devices, i.e. each "word" processed thereby includes sixteen bits of information. One typical microcomputer satisfying this classification is the 8086, manufactured and marketed by INTEL Corp. of Santa Clara, California. Microcomputers that are not provided with the handshake capability are usually 8 bit devices, i.e. each "word" processed thereby includes only 8 bits of information. One such microcomputer is the 8051, manufactured and marketed by INTEL Corp. of Santa Clara, California. Usually, since direct memory access techniques are not available for such a device data is transferred by "first-in-first-out" (FIFO), or buffer, mode techniques. Such techniques are somewhat slower than direct memory access techniques since all data must pass through an intermediate buffer. Hence, the extra read and write steps necessary to pass data across the intermediate buffer extend the time required to move any data processed by such a
microcomputer. Naturally, microcomputers that operate on larger words, such as a 32 bit device, can also be used with the interface 10 without undue difficulty.

In the preferred embodiment, the input and output buffers, 12 and 14, respectively, are sixteen bit latching registers. Thus, in response to a strobe from the controller 18, a word of up to sixteen bits is read from, or written onto, a designated word slot of the TDM bus 38. The designation of word slots is effectively controlled by the apparatus 24 and in this embodiment, includes a memory interconnection bus 48 whereby data is read from, or written into, the data portion of the memory 34. With reference to the above-identified co-pending Australian Patent Application No.

The storage medium 16, shown in more detail in Fig. 2, includes an input data buffer 40, an output data buffer 52 and an address storage medium 54. As shown, and as more fully discussed below, the input data buffer 50 includes a first input data buffer 56, i.e. channel zero and a second input data buffer 58, i.e. channel one. In one embodiment, the input data buffer 50 is a 256 bit random access memory and, if each word slot on the TDM bus 38 transfers eight bit words into the input buffer 12, is, effectively, a 16 x 8 bit input channel zero data buffer 56 and a 16 x 8 bit input channel one data buffer 58. The output data buffer 52 and the address storage medium 54 are each 256 bit random access memories.

Essentially, the interface 10 is adapted to operate in either a FIFO, or buffer, mode or a direct data transfer mode. In the FIFO mode, the input and output buffers, 12 and 14, respectively, exchange data with the input data buffer 50 and the output data buffer 52, respectively. This operating mode is shown by the solid line switch positions in Fig. 1. In
the direct data transfer mode the means 20, shown via the dashed line, switch positions, connect the input and output buffers, 12 and 14, respectively, directly to the local bus 46 of the microcomputer 28.

To provide a clearer understanding of the subsequent discussion of the controller 18, a brief operational discussion of the interface 10 is provided hereinafter. In the FIFO mode of operation, it is initially assumed that each word slot on the TDM bus 38 contains an eight bit word and that the interface 10 is assigned a number of such time slots each frame. In addition, each message to be transferred across the interface, regardless of the direction, will be conveyed in segments, each segment being equal to one word and conveyed in one of the assigned number time slots. Further, each consecutive message segment will be located at the same predesignated word slot in consecutive frames. In such an arrangement, the microcomputer 28 can be either an eight or sixteen bit device. In such an operating mode, the input data buffer 50 is used as two 16 x 8 bit buffers, 56 and 58. Hence, segments from two different messages can be virtually simultaneously accepted by the input data buffer 50 and serviced by the microcomputer 28. The microcomputer 28, when providing messages to the TDM bus 38 via the output buffer 14 then provides 8 bit words into one of the assigned number of word slots per frame.

In the event that the TDM bus 38 is adapted to carry 16 bit words, the input data buffer 50 would, if a sixteen bit microcomputer is connected, accept the sixteen bit words but would then be able to only service one message at any given time. That is, the distinction between channel zero and channel one would no longer exist since each word would take a 16 bit portion of the input data buffer 50.
The operation of the interface 10 in the direct data transfer mode is essentially identical to that of the apparatus described in co-pending Australian Patent Application No. Essentially, data is transferred directly between the microcomputer 28 and the buffers, 12 and 14, without any intermediate read or write steps.

One embodiment of the interface controller 18 is shown in Fig. 3. As shown therein, the controller 18 includes an address decoder 60, a plurality of registers 62, a plurality of counters 64 and a sequencer 66. The address decoder 60 is, preferably, memory mapped to allow direct access to the registers 62, counters 64 and the storage medium 16 by the microcomputer 28. As shown, the address decoder 60 interconnects to each of the registers 62, the counters 64 and the storage medium 16 via address bus 68, which is accessed by the microcomputer 28 via the local bus 46 thereof. By memory mapped it is meant that the microcomputer 28, for example, the ROM 44 thereof, can retrieve an address from the ROM 44. The address is presented on the pins connecting to the address decoder 60. The address decoder 60 then latches the addressed destination to the microcomputer 28 whereafter the microcomputer 28 executes a read or write to that latched destination via the data bus 70 thereof. As known, the address and data buses, 68 and 70, respectively, have the signals thereon multiplexed on the local bus 46 of the microcomputer 28. The strobes are provided by the sequencer 66 via control bus 71.

Throughout the following discussion of the various registers 62 it is understood that the phrase "set" refers to any change in the logic level of a bit in a register in response to an event. The phrase "reset" refers to the return to the previous logic level of a particular bit that has been changed in response to an event, i.e., that bits starting logic level be-
fore any events. It is immaterial whether the original, or starting, logic
level is a binary one or a binary zero. Further, reference to binary ones
or zeros does not imply any particular voltage or current and a set or re-
set implies only a change thereof.

The counters 64 include counters, 72, 74, 75, and 78. The counters 64
can be designated as a current output word pointer (COWP) 72, a next output
word pointer (NOWP) 74, a next input word pointer (INWP) 76 and a current
input word pointer (CINWP) 78. The COWP 72 is incremented each frame and
points to the next word to be transferred to the TDM bus 38 via the output
buffer 14. The NOWP 74 is also incremented each frame and points to the
memory location of the word to be outputted as the "current word" during
the next frame. The INWP 76, preferably, in the sixteen bit TDM mode,
points to the next word to be read by the microcomputer 28 from the input
data buffer 50. The CINWP 78 points to the word currently being accessed by
the microcomputer 28. In the eight bit TDM mode a INWP 76 and CINWP 78 is
provided for each channel, i.e. one for each input channel, 56 and 58, in
the input data buffer 50. The counters 64, during message segment trans-
fers, change at each frame and provide continuously updated addresses to
the address decoder 60.

The registers 62 include interrupt registers, 80, 82 and 84, word
count, or address incrementing, registers 86, 88 and 90, end of message
registers, 92 and 94, a mode register 96 and a plurality of system specific
registers, 98, 100, 102, 104 and 106. The interrupt registers can be des-
ignated as an interrupt request register (IRR) 80, an Interrupt mask regis-
ter (IMR) 82 and an interrupt service register (ISR) 84. The word count
registers can be designated, for the FIFO mode, as input count register for
channel zero (ICRO) 86, an input count register for channel one (ICRH) 88,
and an output count register (OCR) 90. The end of message registers are designated as end of message on input (EDMI) 92 and end of message on output (EDMO) 94. The system particular registers can, for example, be designated as a command reply register (CRPL) 98, a first network alarm register (NALA) 100, a second network alarm register (NALB) 102, an error, or not acknowledged, register (STACK) 104 and a stack counter register (SH) 106.

The IRR 80 is an eight bit read only register. In one embodiment, the bit positions are defined as follows: (a) an alarm bit, which alarm bit is set under various predefined conditions such as a clock or synchronizing failure, (b) an end of message on output, (c) an end of message on input, these two bits are reset when the EDMO 94 and EDMI 92, respectively, are reset, (d) a stack bit which is a logic level 1 only when the stack register 104 is other than zero; (e) two alarm bits responsive to alarms registered in NALA 100 and NALB 102, (f) two bits are used to indicate the "almost overflow" on input or "almost underflow" on output conditions.

The IMR 82 is an eight bit register and can be read from, or written to, by the microcomputer 28. When a bit is set therein the setting of a corresponding bit in the IRR 80 is prevented from causing an interrupt signal to be sent to the microprocessor 28. Thus, under conditions where the microcomputer 28 is performing a program that, for example, has priority, the prioritizing can be accomplished by setting the lower priority task bits in the IMR 82 to ensure that a lower priority task will not interfere with the presently executing program.

The ISR 84 is a read only eight bit register and, when a bit therein is set, generates an interrupt signal via an interrupt line 108 to the local bus 46 of the microcomputer 28. The bits in the ISR 84 will be set in re-
response to a logical AND of the corresponding bits in the IRR 80 with the inverse of the corresponding bits in the IMR 82.

The ICRO and ICRI registers, 86 and 88 respectively, are eight bit read only registers and indicate the number of words in each of the input data buffers, 56 and 58. In the FIFO mode, the values are incremented each time a word is entered into the input data buffer 50 and, when a preselected number of words are stored, the corresponding bit in the IRR 80 is set. Similarly, the COR 90 is an eight bit read only register that is decremented each time a word is outputted. When the count reaches a preselected threshold an interrupt request bit is set.

In the direct data transfer mode one of the ICR registers, 86 or 88, and the COR 90 are, effectively, the input and output address incrementers, IAI and OAI, respectively, discussed in the aforementioned patent application no.

The EOMI 92 and EOMO 94 registers are eight bit read only registers. These registers, 92 and 94, ensure that each complete message is separated from subsequent messages. When an end of message flag enters the input buffer 12 the corresponding bit in the IRR 80 is set and no further message segments will be accepted by the input buffer 12 until that bit is reset. When an end of message flag on output is detected a new message can be started only after the EOMO bit is reset.

In the preferred embodiment, the mode register 96 is an eight bit register and can be both read from, or written to, by the microcomputer 28. The eight bits preferably correspond to the following information: (a) 2 bits are used to indicate a threshold for the "almost overflow" interrupt; (b) 2 bits are used to indicate a threshold for the "almost underflow" interrupt; (c) one bit is used to indicate to the interface 10 to operate in
either an 8 bit mode or a 16 bit mode; (d) one bit is used to select either the FIFO, or buffer, mode or the direct data transfer mode; and (e) one bit is used in the DMA mode, to hold a particular address in the address storage medium 54 until valid information is to be transferred.

The CRPL 98, NALA 100 and NALB 102 are eight bit read only registers and are set according to instructions provided in predesignated reply or alarm time slots in each frame on the TDM bus 38. In this fashion, the microcomputer 28 can be used to execute prespecified system oriented programs, for example, to minimize the consequences of malfunctions. In addition, the STACK 104 is a plurality, such as eight, eight bit read only register that effectively record each failure of a port or channel to complete an instruction. That is, if a particular source forwards an instruction and that instruction is not executed, a signal is registered in the STACK 104. The not acknowledged signal is provided to the STACK 104 via an eight bit external control bus from the switch controller 36 of the intelligent switch 32.

The SR 106 register is an eight bit read only register that counts the number of entries provided to the STACK 104 and sets a bit in the IRR 80 accordingly.

The sequencer 66 is, for all intents and purposes, a state machine. As well as known, a state machine is a controller the next state of which depends only on its present state and its present inputs. Such state machines can be implemented by use of readily available programmable logic arrays (PLAs) or by programmable read only memories (PROMs). Preferably, however, the controller 18 of the interface 10 is implemented using large scale integration (LSI) techniques and satisfies the state diagram shown in Fig. 4. In the preferred embodiment, the transition between states occurs...
upon receipt of a clock signal from an external clock, not shown in the
drawings. Usually, the clock signal is provided by a master clock associ-
ated with the system connected to the first port 22 whereby the state tran-
sitions of the interface are co-ordinated with that system.

Referring now to Fig. 4, there is shown therein a state diagram of the
various states involved in the data transfer between the TDM bus 38 and the
microcomputer 28. As means 20 is set by the above mentioned bit in the
mode register 96 to either the FIFO, buffer, mode or the direct data
transfer mode whereupon the sequencer 66 executes one or the other state
sequences shown in Fig. 4.

In the buffer mode indicated by the solid line switch position, the
state A is representative of an idle state. From the idle state, when
strobed, the state is either shifted to state B or state C. As shown,
state C is an updating of all registers 62 and occurs after each transfer
of data. State B is a buffer read state during which the microcomputer 28
read the information from the buffer 50. On the next strobe the microcom-
puter 28 is set to no operation state D and, on the next strobe, shifts to
either state C, which is a register update state, or state F, a buffer
write. Hence, if information is available, it is first read in from the
TDM bus 38, if information is to be sent out, it occurs on the next set of
strobes. After the buffer write state occurs, the end of message regis-
ters, at state G, are updated or interrupts are generated. Thereafter, the
sequencer moves to a state E, which is also a no operation state. The no
operation states, D and E, are designed to ensure stepwise synchronization
with both the internal interface operation and the TDM bus 38. Thereafter,
the sequencer 66 moves to state C where the updating of registers 62
occurs and thereafter returns to the idle state A.
Basically, the above state diagram is adapted to provide the following method for operation during the FIFO, or buffer mode. First, data is extracted from the TDM bus 38 on either channel 0 or 1 and transferred into the input data buffer 50. The input data buffer 50 is emptied by the microcomputer 28 at either an end of message interrupt or an "almost full" interrupt. Data is outputted from the microcomputer 28 by a write step from the microcomputer 28 into the output data buffer 52 and a subsequent write step from the output data buffer 52 into the output buffer 14 whereafter it is injected onto the TDM bus 38 in the appropriate time slot corresponding to the destination channel.

In the direct data transfer mode state A' is an idle state which signifies that the interface 10 is ready to interact with the TDM bus 38. When strobed the sequencer 66 moves to state B' whereupon the address of the next message segment of a predetermined word slot is acquired by the microcomputer 28. At state C' the address is secured by the microcomputer 28 and the message segment is, at state F', directly transferred from the RAM 42 to the output buffer 14. At state E' the message segment in the output buffer 14 is injected, or read into, the appropriate word slot or the TDM bus 38.

On the next strobe, if there is no incoming message segment, the registers 62 are updated, during state D'. If, however, a message segment is arriving in the next designated word slot the state machine shifts to state H'. In state H' the microcomputer 28 acquires the address associated with the word slot having an incoming message segment and, in state I' presents that address to the RAM 42.

In states L' and K' the message segment is written into the input buffer 15, then, at K', directly written into the RAM 42. The next
state, state $J'$ is used to update the registers $62$. The following transition, to state $G'$, is, effectively, a nop operation state but is used to maintain the state machine in sequence with the clock pulse.

As shown, when at either state $E'$ surging a "fetch" or state $K'$ during a "put" the sequencer $66$ is able to shift to states $B'$ or $H'$ respectively. Such state transitions can be implemented to allow the interface $10$ having a $16$ bit microcomputer $28$ to be readily used with an eight bit system. In such an eight bit system it would thus be desirable to read or write two eight bit words to maximize the efficiency of the microcomputer $28$.

Accordingly, the interface $10$ can interface either an eight bit or sixteen bit microcomputer, with or without "direct memory access" capability, to an eight or sixteen bit intelligent device.

The present invention has been described herein with regard to a particular exemplary embodiment and is not deemed limited thereto as other configurations and uses may be developed which do not depart from the spirit and scope of this invention. Consequently, the present invention is deemed limited only by the appended claims and the reasonable interpretation thereof.
Claims
The claims defining the invention are as follows:

1. An interface for use with a microcomputer, said interface comprising a first means for transporting data between a first port and a second port, said first means including a data storage medium, a second means for transporting data between said first port and said second port, said second means including an address storage medium, and means for selecting either said first means or said second means for transporting data across said interface.

2. An interface as claimed in claim 1, wherein said first means includes an input data buffer and an output data buffer.

3. An interface as claimed in claim 2, wherein said input data buffer includes first and second input data buffer portion.

4. An interface as claimed in claim 1, further comprising means for accepting data from said first port, and means for presenting data to said first port.

5. An interface as claimed in claim 4, wherein said means for accepting data is an input buffer and said means for presenting data is an output buffer.

6. An interface as claimed in claim 5, wherein said data is a segment of a multi-segmented message and said input and output buffers include a plurality of storage bits, said storage bits being at least equal to the number of bits in said segment.

7. An interface as claimed in claim 6, wherein each said segment, when said first data transporting means is selected, is temporarily stored in said data storage medium and, when said second data transporting means is selected, is exchanged directly between said microcomputer and said input or output buffers.
8. An interface as claimed in claim 1, wherein said data is one segment of a segmented message, said interface further comprising means for registering the end of a message.

9. An interface as claimed in claim 8, wherein said end of message registering means includes separate means for registering the end of a message to said microcomputer and the end of a message from said microcomputer.

10. An interface as claimed in claim 8, further comprising means, responsive to said end of message registering means, for recording a request to interrupt said microcomputer when an end of message is registered.

11. An interface as claimed in claim 10, wherein said interrupt request recording means includes a first bit and a second bit, said first bit being associated with said end of message to said microcomputer registering means and said second bit being associated with said end of message from said microcomputer registering means.

12. An interface as claimed in claim 10, further comprising means for selectively servicing said interrupt requests.

13. An interface as claimed in claim 12, wherein said selective servicing means includes an interrupt mask register and an interrupt service register, the bit positions in said interrupt service register being set as a logical AND of the setting of the corresponding bit in said interrupt request register and the inverse of the setting of the corresponding bit in said interrupt mask register.

14. An interface as claimed in claim 1, wherein said data is a segment of a segmented message, said interface further comprising means, when said second data transporting means is selected, for providing a different ad-
An interface as claimed in claim 14, wherein said different address providing means includes an address incrementer.

16. An interface as claimed in claim 14, wherein said different address providing means includes an input message segment address incrementer and an output message segment address incrementer.

17. An interface as claimed in claim 1, further comprising means for virtually simultaneously transferring message segments of a plurality of messages between said first port and said second port.

18. An interface as claimed in claim 17, wherein said address storage medium is adapted to store a plurality of addresses.

19. An interface as claimed in claim 18, wherein said plurality of addresses includes a first group of addresses for association with data from a TDM bus interconnected to said first port and a second group of addresses for association with data to said TDM bus.

20. An interface as claimed in claim 19, wherein each one of said first group of addresses is associated with a different message from said TDM bus and each one of said second group of addresses is associated with a different message to said TDM bus, and means for registering the end of a message.

21. An interface as claimed in claim 20, wherein said registering means includes a first register having a plurality of bits, each said different message from said TDM bus having a bit associated therewith, which bit being set in response to the end of said message associated therewith, and a second register having a plurality of bits, each said different mes-
sage to said TDM bus having a bit associated therewith, which bit being set in response to the end of said message associated therewith.

22. An interface as claimed in claim 21, further comprising means, responsive to the setting of bit $s$ in said first or said second register, for requesting an interrupt signal to be sent to said microcomputer, and means for selectively servicing said interrupt requests.

23. An interface as claimed in claim 19, further comprising means for incrementing each one of said first group of addresses for each message segment of the message associated therewith, and means for incrementing each one of said second group of addresses for each message segment of the message associated therewith.

24. An interface as claimed in claim 17, wherein when said first data transportation means is selected, said data storage medium includes a first input data portion and a second input data portion, said first and said second input data portions simultaneously storing segments from first and second messages.

25. An interface as claimed in claim 24, further comprising means for generating an interrupt signal when the number of segments stored in either said first or second input data portion exceeds preselected thresholds associated therewith whereby said microcomputer can empty either said first or said second input data portion to avoid message segment overflow of either said first or second input data portion and wherein said interrupt signal generating means includes means for independently counting the number of message segments stored in said first and second data portion.

26. An interface as claimed in claim 17, wherein each message segment is conveyed in a word slot of a time-division multiplexed bus associated with an intelligent switch, said word slot having a unique position as-
An interface for use with a microprocessor substantially as herein described with reference to Figs. 1 to 4 of the accompanying drawings.

DATED THIS SECOND DAY OF DECEMBER, 1985
ITT CORPORATION.
the return to the previous logic level of a particular bit that has been changed in response to an event, i.e., that bits starting logic level be-
registers can be designated, for the FIFO mode, as input count register for channel zero (ICHO) 86, an input count register for channel one (ICHI) 88,
FIG. 2
(b) 2 bits are used to indicate a threshold for the "almost underflow" interrupt; (c) one bit is used to indicate to the interface 10 to operate in
scale integration (LSI) techniques and satisfies the state diagram shown in Fig. 4. In the preferred embodiment, the transition between states occurs...
with both the internal interface operation and the TEM bus 38. Thereafter, the sequencer 66 moves to state C where the updating of registers 62 occurs and thereafter returns to the idle state A.
FIG. 4
FIG. 4