PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : G06F 13/42

A2

(11) International Publication Number: **WO 00/02134**

(43) International Publication Date: 13 January 2000 (13.01.00)

(21) International Application Number: PCT/US99/14696

(22) International Filing Date: 30 June 1999 (30.06.99)

(30) Priority Data:

60/091,383	1 July 1998 (01.07.98)	US
09/248,939	11 February 1999 (11.02.99)	US

(71) Applicant: QUALCOMM INCORPORATED [US/US]; 6455 Lusk Boulevard, San Diego, Ca 92121 (US).

(72) Inventor: JHA, Sanjay, K.; 7415 Sean Taylor Lane, San Diego, CA 92126 (US).

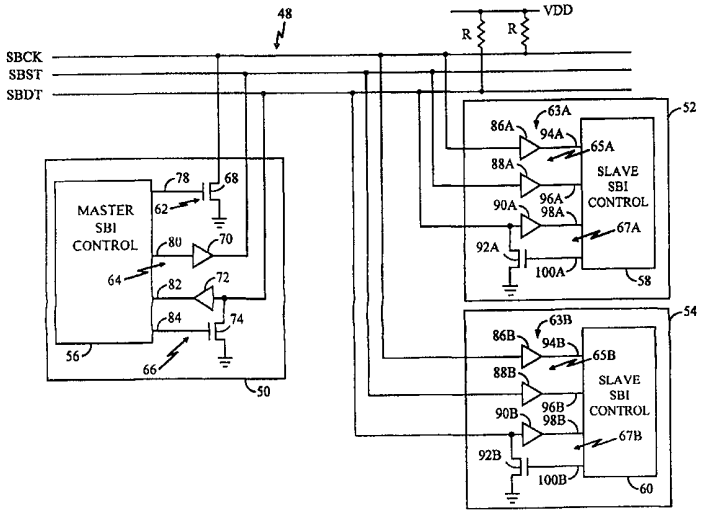
(74) Agents: OGROD, Gregory, D. et al.; Qualcomm Incorporated, 6455 Lusk Boulevard, San Diego, CA 92121 (US).

(81) Designated States: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, UZ, VN, YU, ZA, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published

Without international search report and to be republished upon receipt of that report.

(54) Title: IMPROVED INTER-DEVICE SERIAL BUS PROTOCOL



(57) Abstract

An inter-device serial bus protocol facilitates the interconnection and communication among various devices via a serial bus. The bus (48) comprises a clock wire, a data wire, and a start/stop wire. A master serial bus interface couples a master device to the serial bus. A slave serial bus interface couples a slave device to the serial bus. The master serial bus interface (180) may comprise a transaction initiator, a data write mechanism, a data read mechanism, and a clock driver. The transaction initiator initiates a transaction by pulling the signal level of the start/stop wire low. The data write mechanism controls the signal level on the data wire in accordance with the data to be written to the slave device. The data read mechanism reads data by monitoring the signal level on the data wire. The clock driver controls the signal level on the clock wire in accordance with a desired clock signal.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	**ES**	Spain	**LS**	Lesotho	**SI**	Slovenia
AM	Armenia	**FI**	Finland	**LT**	Lithuania	**SK**	Slovakia
AT	Austria	**FR**	France	**LU**	Luxembourg	**SN**	Senegal
AU	Australia	**GA**	Gabon	**LV**	Latvia	**SZ**	Swaziland
AZ	Azerbaijan	**GB**	United Kingdom	**MC**	Monaco	**TD**	Chad
BA	Bosnia and Herzegovina	**GE**	Georgia	**MD**	Republic of Moldova	**TG**	Togo
BB	Barbados	**GH**	Ghana	**MG**	Madagascar	**TJ**	Tajikistan
BE	Belgium	**GN**	Guinea	**MK**	The former Yugoslav Republic of Macedonia	**TM**	Turkmenistan
BF	Burkina Faso	**GR**	Greece			**TR**	Turkey
BG	Bulgaria	**HU**	Hungary	**ML**	Mali	**TT**	Trinidad and Tobago
BJ	Benin	**IE**	Ireland	**MN**	Mongolia	**UA**	Ukraine
BR	Brazil	**IL**	Israel	**MR**	Mauritania	**UG**	Uganda
BY	Belarus	**IS**	Iceland	**MW**	Malawi	**US**	United States of America
CA	Canada	**IT**	Italy	**MX**	Mexico	**UZ**	Uzbekistan
CF	Central African Republic	**JP**	Japan	**NE**	Niger	**VN**	Viet Nam
CG	Congo	**KE**	Kenya	**NL**	Netherlands	**YU**	Yugoslavia
CH	Switzerland	**KG**	Kyrgyzstan	**NO**	Norway	**ZW**	Zimbabwe
CI	Côte d'Ivoire	**KP**	Democratic People's Republic of Korea	**NZ**	New Zealand		
CM	Cameroon			**PL**	Poland		
CN	China	**KR**	Republic of Korea	**PT**	Portugal		
CU	Cuba	**KZ**	Kazakstan	**RO**	Romania		
CZ	Czech Republic	**LC**	Saint Lucia	**RU**	Russian Federation		
DE	Germany	**LI**	Liechtenstein	**SD**	Sudan		
DK	Denmark	**LK**	Sri Lanka	**SE**	Sweden		
EE	Estonia	**LR**	Liberia	**SG**	Singapore		

IMPROVED INTER-DEVICE SERIAL BUS PROTOCOL

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates to inter-device serial buses. In certain implementations, the invention relates to an inter-device serial bus having a small number of lines and using a protocol for facilitating simple master/slave relationships between devices connected to the bus, some of which may utilize a different protocol.

2. Description of Related Art

Electronic devices -- hand-held cellular phones, electronic calculators, CD players, camcorders, to name a few -- include various internal components (ICs or chips) controlled by a single-chip microprocessor. Frequently, these devices incorporate an inter-device serial bus to link their internal components together. The chips communicate with an internal microprocessor, transferring data to and from the microprocessor via the inter-device serial bus.

The microprocessor has certain control over the chip. For example, it can configure the chip so that it uses no power, change its functionality, and otherwise interact with the chip to change its performance. Absent a serialization process, i.e., the use of an inter-device serial bus, each communication from the microprocessor to a particular chip would need to be routed separately to the chip's appropriate register using more pins and potentially including greater power consumption. However, with the use of an inter-device serial bus, a small number of pins can be used to facilitate the control of hundreds of functions in various chips by a microprocessor.

The Phillips I^2C-bus is a commonly used inter-device serial bus protocol. In accordance with the I^2C-bus protocol, slave chips listen to the bus, and when addressed by the processor serving as a master, respond by grabbing the appropriate messages, internally decoding address information, and directing the data accordingly.

There is a desire to make these multi-chip devices more compact in size, and to reduce the costs associated with their design and manufacture. Accordingly, mechanisms are needed to facilitate the interchangeability of pre-made chips in different multi-chip devices, and to provide alternate yet inter-

compatible serial bus protocols. This increases options available to IC suppliers and producers of end product multiple chip devices. When chips are build with common serial interfaces their configuration and use becomes standardized and economical at the board or system level.

Accordingly, there is a need for an improved inter-device serial bus protocol, which is operable over a wide range of clock rates, and can coexist with the two-wire Phillips I^2C bus protocol. Such a protocol would need to accommodate the transferring of clock information, data information, and information controlling the starting and stopping of transactions among inter-connected chips. In some instances, such a protocol would need to include mechanisms for arbitrating among plural masters accessing the serial bus, for addressing slave devices, and for indicating when data should be read from or written to a slave device. Communication between devices via such an improved serial bus protocol would preferably involve a synchronous operation.

3. Definitions of Terms

The following term definitions are provided to assist in conveying an understanding of the embodiments and features disclosed herein.

Control Information:

Information for controlling the flow of information over a bus. Such information may include a start transaction indication, and a stop transaction indication.

Master:

A device connected to a bus capable of initiating a transaction with other devices connected to the same bus.

Parity:

This term denotes a quality of sameness or equivalence between devices communicating with each other. It may comprise an error-checking procedure in which transfer data is checked, for example, to determine if the information being transmitted is transmitted without error.

Receiver:

Any device receiving data from a data line provided on the bus.

Serial Communication:

The transfer of information between computers or other devices one bit at a time over a single line. Serial communication can be synchronous (controlled by a time standard such as a clock) or asynchronous (managed by the exchange of control signals that govern the flow of information). In serial communication, both the sender and receiver use the same parity and control information.

Slave:

A device that can be written to and read from, but which does not initiate a transaction.

Synchronous Operation:

An operation controlled by a clock or timing mechanism. In the case of a synchronous bus operation, data is transferred in accordance with clock pulses either embedded in the data stream or provided simultaneously on a separate line.

Transaction:

A transaction comprises the transfer of data between a master and a slave for a period of time extending between the time at which the data transfer is initiated by the master until the time at which the data transfer is terminated by the master or another device.

Transmitter:

Any device connected to a bus which is transmitting information over a data line provided on the bus.

SUMMARY OF THE INVENTION

The present invention is provided to improve upon existing serial interface protocols for linking master and slave devices within small multi-chip devices. The present invention generally provides mechanisms for implementing a synchronous protocol between devices via a bus interface. The mechanisms facilitate efficient communication between various devices using only a small number of lines and otherwise facilitate the design and implementation of multiple chip devices and systems. The protocol mechanisms presented herein further should further help with fault diagnosis and debugging of various components of the multi-chip system.

The present invention, therefore, is directed to an inter-device serial bus protocol, or one or more parts thereof, for facilitating the interconnection and the

communication among various devices via a serial bus. The serial bus may comprise a clock wire, a data wire, and a start/stop wire. A master serial bus interface may be provided which couples a master device to the serial bus. A slave serial bus interface may be provided which couples a slave device to the serial bus. In certain aspects of the invention, the master serial bus interface comprises a transaction initiator, a data write mechanism, a data read mechanism, and a clock driver. The transaction initiator initiates a transaction by pulling the signal level of the start/stop wire low. The data write mechanism controls the signal level on the data wire in accordance with the data to be written to the slave device. The data read mechanism reads data by monitoring the signal level on the data wire. The clock driver controls the signal level on the clock wire in accordance with a desired clock signal.

BRIEF DESCRIPTION OF THE DRAWINGS

The above and other objects, features, and advantages of the present invention are further described in the detailed description which follows with reference to the drawings by way of non-limiting exemplary embodiments of the present invention, wherein like reference numerals represent similar parts of the present invention throughout the several views, and wherein:

Fig. 1. illustrates the connections of devices to an I^2C-bus;

Fig. 2 is a waveform diagram depicting a bit transfer on an I^2C-bus;

Fig. 3 is a waveform diagram depicting start and stop indications on an I^2C-bus;

Fig. 4 is a schematic representation of the basic data format of the I^2C-bus protocol;

Fig. 5 is a schematic diagram of a serial bus configuration in accordance with the illustrated embodiment of the present invention;

Fig. 6 shows a serial bus connected to devices utilizing the new protocol as well as devices utilizing the I^2C-bus protocol;

Fig. 7 illustrates an interrupt transfer mode (ITM) message format;

Fig. 8 illustrates a fast transfer mode (FTM) message format;

Fig. 9 illustrates a bulk transfer mode (BTM) message format;

Fig. 10 is a block diagram of a master device;

Fig. 11 is a more detailed block diagram of the master SBI controller illustrated in Fig. 10;

Fig. 12 is a block diagram of a data path block;

Fig. 13 is a flow chart showing the operation of the master SBI controller of Fig. 10 performing a transaction; and

Fig. 14 is a block diagram of a slave device.

DETAILED DESCRIPTION OF THE ILLUSTRATED EMBODIMENT

Figs. 1-6 illustrates various aspects of the I^2C-bus protocol and some example implementations thereof. Figs. 7-24 illustrate in various respects an exemplary embodiment of the present invention, directed to a new serial bus interface (SBI) protocol. The illustrated SBI protocol utilizes a bus having three wires. Devices can be coupled to the bus allowing them to transfer information to and from each other. Such devices include master devices and slave devices. An arbitration procedure may be incorporated in the SBI protocol for allowing more than one master to initiate a data transfer at the same time. The illustrated SBI protocol is compatible not only with devices accommodating the new protocol, but also accommodates I^2C-bus devices.

Referring now to the drawings in greater detail, the conventional I^2C-bus protocol is first described with reference to Figs. 1-7. Fig. 1 shows a pair of devices connected to an I^2C-bus, otherwise referred to as an I^2C-bus configuration 10. The illustrated I^2C-bus configuration 10 comprises a bus 11 having a pair of wires. The pair includes a first wire which is a serial data line SDA, and a second wire which is a serial clock line SCL. Pull-up resistors 12, 13 are connected respectively at first ends thereof to serial clock line SCL and serial data line SDA, and at their second end to a common DC voltage source $+V_{DD}$. A pair of devices including a first device 14 and a second device 16 are coupled to the illustrated I^2C-bus 11. First device 14 comprises, among other elements (not shown), a clock interface circuit 18 and a data interface circuit 19. Second device 16 comprises, among other elements (not shown), a clock interface circuit 18' and a data interface circuit 19'. Each of the illustrated clock interface circuits 18, 18' comprises an amplifier and a transistor. When each wire SDA and SCL is free, both lines are high or in a one state. The transistors of the respective data interface circuits 18, 18', 19, and 19' are coupled in an open-drain or open-collector fashion in order to perform a wired-AND function, so that when activated, they pull their respective line SDA or SCL down to a low state, thereby indicating a logical "zero". A variety of different technologies (CMOS, NMOS, BIPOLAR, etc.) can be connected to the I^2C-bus.

Fig. 2 is a waveform diagram depicting a bit transfer on an I^2C-bus. The top waveform is the signal and data line SDA, and the bottom line is the signal and clock line SCL. During a stability check period 20, during which the clock signal is high, the level of the data signal on data line SDA cannot change.

During a data line change period 22, the period during which the clock signal is low, the high or low state of the data line SDA can change. One clock pulse is generated for each data bit transferred on data line SDA.

Fig. 3 shows a waveform diagram depicting the stop and start indications on an I^2C-bus. The two waveforms on data line SDA and clock line SCL, respectively, represent a start condition 24 and a stop condition 26 as shown. Start condition 24 is indicated when there is a high to low transition on data line SDA while the signal on clock line SCL is high. Stop condition 26 is indicated when there is a transition from a low to a high on data line SDA while the signal level on clock line SCL is high. According to the I^2C-bus specification, start and stop conditions are always generated by the master.

Fig. 4 is a schematic block diagram representing the general protocol format of the I^2C-bus protocol. A single transaction is shown, the start of which is indicated by start condition 24 and the end of which is indicated by stop condition 26. The first byte of information transmitted includes a seven-bit slave address 28 followed by a one-bit read/write bit (R/W) 30. An acknowledge bit A 32 then follows. Following the first acknowledge bit 32 is a byte of data and a second acknowledge bit 36. An additional byte of data 38 follows, together with another acknowledge bit 40.

As shown, each byte is followed by an acknowledge bit as indicated by the "A" blocks in the sequence. If the R/W bit is zero, data is written from the master to the slave device, in which case an acknowledge or failure to acknowledge only comes from the slave device to the master. If the R/W bit is a one, data is read from the slave to the master. If the R/W bit is a one, data is read from the slave device to the master device, and the acknowledgement bits are sent from the master device to the slave device.

Fig. 5 shows a serial bus configuration in accordance with an illustrated embodiment of the present invention. As shown in Fig. 5, a master device 50 is connected via a serial bus 48 to first and second slave devices 52, 54. Serial bus 48 comprises a clock wire SBCK, a start/stop wire SBST and a data wire SBDT. Clock wire SBCK and data wire SBDT are each connected to a pull-up DC voltage V_{DD} via pull-up resistors R. The pull-up resistor values may be adjusted to maintain a certain RC constant, as more devices are added on serial bus 48. The start/stop wire SBST is not connected in a pull-up fashion. Each of master device 50, first slave device 52, and second slave device 54 comprises a respective SBI controller and a set of bus interaction circuits which collectively serve as a serial bus interface for coupling the respective device to serial bus 48.

Master SBI controller 56 is connected to clock wire SBCK, start/stop wire SBST, and data wire SBDT via respective interaction circuits comprising a clock wire interaction circuit 62, a start/stop wire interaction circuit 64, and a data wire interaction circuit 66. The illustrated clock wire interaction 62 comprises a field effect transistor 68 comprising source, drain, and gate electrodes. Its source electrode is connected to clock wire SBCK. Its drain electrode is connected to ground, and its gate electrode is connected to a clock wire operation terminal 78 of master SBI controller 56.

Start/stop interaction circuit 64 comprises an amplifier 70 having input and output terminals. Its output terminal is connected to start/stop wire SBST, and its input terminal is connected to a start/stop wire operation terminal 80.

Data/wire interaction circuit 66 comprises an amplifier 72 and a field effect transistor 74. Amplifier 72 comprises an input terminal and an output terminal, the input terminal being connected to data wire SBDT, and the output terminal being connected to a data monitoring terminal 82 of master SBI controller 56. Field effect transistor 74 comprises source, drain, and gate electrodes. Its source electrode is connected to data wire SBDT in common with the input terminal of amplifier 72. Its drain electrode is connected to ground. Its gate electrode is connected to a data operation terminal 84 of master SBI controller 56.

Each of slave devices 52 and 54 comprises a plurality of interaction circuits respectively coupled to clock wire SBCK, start/stop wire SBST, and data wire SBDT. More specifically, slave devices 52 and 54 comprise clock wire interaction circuits 63a and 63b connected to clock wire SBCK, start/stop wire interaction circuits 65a and 65b connected to start/stop wire SBST, and data wire interaction circuits 67a and 67b connected to data wire SBDT.

Clock interaction circuits 63a and 63b comprise respective amplifiers 86a and 86b. Start/stop wire interaction circuits 65a and 65b comprise respective amplifiers 88a and 88b. Data wire interaction circuits 67a and 67b comprise respective first and second sets of circuit elements. The first set of circuit elements forming data wire interaction circuit 67a comprises an amplifier 90a and a field effect transistor 92a. The second set of circuit elements forming data wire interaction circuit 67b comprises an amplifier 90b and a field effect transistor 92b.

Amplifier 86a comprises an input terminal and an output terminal, the input terminal being connected to clock wire SBCK, and the output terminal being connected to a clock monitoring terminal 94a. Amplifier 88a comprises input and output terminals, the input terminal being connected to start/stop

wire SBST, and the output terminal being connected to a start/stop monitoring terminal 96a. Amplifier 90a comprises input and output terminals, the input terminal being connected to data wire SBDT, and its output terminal being connected to a data monitoring terminal 98a. Field effect transistor 92a comprises source, drain and gate electrodes. Its source electrode is connected in common with the input terminal of amplifier 90a to data wire SBDT. Its drain electrode is connected to ground. Its gate electrode is connected to a data operation terminal of slave controller 58.

In operation, a zero is represented on any one of the wires SBCK, SBST, SBDT by a low voltage level. Particularly, on clock wire SBCK and data wire SBDT, a zero is indicated by pulling down the line, and a one is indicated by tri-stating the driver and letting an external pull-up take the voltage level high. On start/stop wire SBST, a zero or a one is indicated under the exclusive control of the appropriate master controller 56 via an amplifier 70. When the signal at the output terminal of amplifier 70 is low, a zero is indicated on the start/stop wire SBST, and when the signal is high a "1" is indicated. The clock wire interaction circuit 62 of master device 50 serves to allow a master SBI controller 56 to operate on clock wire SBCK via a clock wire operation terminal 78, thereby pulling the voltage level on clock wire SBCK low. Start/stop wire detection circuit 64 of master device 50 facilitates the operation of master SBI controller 56 on start/stop wire SBST via a start/stop operation terminal 80, so as to control the signal level present on start/stop wire SBST. Data wire interaction circuit 66 of master device 50 allows master SBI controller 56 to operate on data wire SBDT and to monitor the signal level on data wire SBDT via data monitoring terminal 82 and data operation terminal 84, respectively.

More specifically, clock wire and master controller 56 will operate on the gate electrode of field effect transistor 68 via clock wire operation terminal 78, thus allowing the current to flow between the source and drain of field effect transistor 68 and bringing the voltage level present on clock wire SBCK low. Master controller 56 operates on data wire SBDT by triggering the gate electrode of field effect transistor 74 via data operation terminal 84, thus causing the current to flow from the source to the drain of field effect transistor 74 and accordingly bringing the voltage level present on data wire SBDT low.

Each slave device 52 and 54 (and possibly others, not illustrated) may monitor the signal level at each of clock wire SBCK, start/stop wire SBST, and data wire SBDT via a monitoring terminal which receives an output terminal of a corresponding amplifier. More specifically, slave SBI controller 58 of slave device 52 includes a clock monitoring terminal 94a which receive the output

terminal of amplifier 86a, the input terminal of which is connected to clock wire SBCK. Slave controller 58 also includes a start/stop monitoring terminal 96a which receives the output terminal of amplifier 88a, the input terminal of which is connected to start/stop wire SBST. Data monitoring terminal 98a of slave controller 58 receives the output terminal of amplifier 90a, the input terminal of which is connected to data wire SBDT. Data wire operation terminal 100a of slave controller 58 is connected to the gate electrode of field effect transistor 92a. This allows slave controller 58 to switch field effect transistor 92a, thus causing current to flow from the source to the drain of field effect transistor 92a and bringing the voltage level on data wire SBDT low.

Master device 50 is allowed to communicate with slave devices 52 and 54 (and optionally other devices, not specifically shown) via serial bus 48. The illustrated serial bus 48 includes a set of bus wires (i.e., clock wire SBCK, start/stop SBST, and data wire SBDT). A transaction between master device 50 and one or more of slave devices 52 and 54 can be initiated by master device 50 providing a start indication on start/stop wire SBST. Accordingly, master SBI controller 56 of master device 50 will set the voltage level on start/stop wire SBST by outputting a voltage level at start/stop operation terminal 80 which will cause amplifier 70 to output the desired voltage level. More specifically, in the illustrated embodiment, master SBI controller 56 initiates a transaction by outputting a signal at start/stop operation terminal 80 resulting in pulling the signal level on start/stop wire SBST low.

A clock signal is placed on clock wire SBCK by master device 50. Master controller 56 includes a clock driver (not shown in Fig. 5) which causes the appropriate operation signal to be output at clock wire operation terminal 78 in order to switch field effect transistor 68. In this fashion, master SBI controller 56 controls the signal level on clock wire SBCK in accordance with a desire clock signal. The resulting desired clock signal is used in common by master device 50 and slave devices 52 and 54 to facilitate synchronous operation of master device 50 and any one or more slave devices 52, 54 participating in the transaction.

Master SBI controller 56 of master device 50 comprises a data write mechanism, utilizing data operation terminal 84 to switch field effect transistor 74, thus either tri-stating field effect transistor 74 and leaving the voltage level on data wire SBDT high or pulling the voltage level down by causing a current flow from the source to the drain of field effect transistor 74. Accordingly, the signal level on data wire SBDT is controlled in accordance with data to be written to the slave device 52 or 54 participating in the particular transaction. This operation will be further described below with reference to subsequent figures.

Fig. 6 shows a serial bus configuration, wherein a master device and slave device of a first type (using the new protocol described herein) are connected to a serial bus 48 in common with a master and slave device of a second type (using the I^2C-bus protocol). Specifically, as shown, a serial bus 48 is provided comprising a clock wire SBCK, a start/stop wire SBST, and a data wire SBDT. Clock wire SBCK and data wire SBDT are each connected to a DC voltage V_{DD} via a pull-up resistor R. A first type master 100, comprising a master device utilizing a first serial bus interface protocol, is connected via separate terminals to each of clock wire SBCK, start/stop wire SBST, and data wire SBDT. A first type slave 102, comprising a slave device utilizing the first serial bus interface protocol, is connected via respective terminals to clock wire SBCK, start/stop wire SBST and data wire SBDT. First type master 100 and first type slave 102 may comprise master and slave devices as described above in reference to Fig. 5. Accordingly, first type master 100 may comprise master SBI controller 56 and interaction circuits including clock wire interaction circuit 62, start/stop wire interaction circuit 64, and data wire interaction circuit 66. First type slave 102 may comprise slave controller 58 and interaction circuits including clock wire interaction circuit 63a, start/stop wire interaction circuit 65a, and data wire interaction circuit 67a.

I^2C master 104 and I^2C slave 106 may comprise master and slave devices as described, for example, in the Philips Semiconductors document entitled "The I^2C-bus and How to Use It (Including Specifications)" -- 1995 update, pages 1-24 (April 1995), the content of which is hereby incorporated by reference herein in its entirety. In any event, I^2C master 104 and I^2C slave 106 may be constructed in any well known manner. Each of the I^2C devices 104 and 106 comprises two terminals respectively connected to clock wire SBCK and data wire SBDT in conformance with the I^2C protocol. Accordingly, I^2C master 104 comprises a first terminal connected to clock wire SBCK, and comprises a mechanism for controlling the signal level on clock wire SBCK in accordance with a desired clock signal. The desired clock signal is used in common by I^2C master 104 and I^2C slave 106 to facilitate the synchronous operation of I^2C master 104 and I^2C slave 106 when a transaction is occurring between those two devices.

I^2C master 104 further includes a second terminal connected to data wire SBDT and a transaction initiator (not shown) for initiating a transaction by providing a start indication on data wire SBDT in accordance with the I^2C-bus protocol, which is described above with reference to Fig. 3. I^2C master 104 is further provided with a master data write mechanism (not shown) for controlling the signal level on data wire SBDT in accordance with data to be

written to I^2C slave 106. That data may include payload data and overhead data including information addressing I^2C slave 106, in accordance with the I^2C protocol.

In operation, first type master device 100 initiates a transaction by providing a start indication on a designated bus wire designated for the transmission of control information -- including transaction start and stop indications. In the illustrated embodiment, the designed bus wire comprises a start/stop wire SBST, which is separate and distinct from clock wire SBCK and data wire SBDT. First type master device 100 controls the signal level on data bus wire SDBT in accordance with data to be transferred to first type slave device 102. That data may include payload data and overhead data including information addressing the first type slave device 102 as well as one or more specific registers therein.

The signal level on clock wire SBCK is controlled by first type master device 100 in accordance with a desired clock signal being used in common by first type master device 100 and first type slave device 102 to facilitate synchronous operation of first type master device 100 and first type slave device 102.

First type master device 100 terminates a transaction by providing a stop indication on the designated bus wire (start/stop wire SBST) and maintaining that stop indication irrespective of the existence of any indications present on data and clock wires SBDT and SBCK. In the illustrated embodiment, this is done by maintaining the level of the start/stop wire SBST in a high state. Accordingly, first type master device 100 inhibits the control by first type master device 100 of the signal level on data wire SBDT to transfer data and inhibits the control by first type master device 100 of the signal level on clock wire SBCK for as long as the stop indication is provided and maintained.

I^2C master device 104 initiates a transaction by providing an I^2C start indication in accordance with the I^2C protocol on data wire SBDT. I^2C master device 104 controls the signal level on data wire SBDT in accordance with data to be transferred to I^2C slave device 106, and that data may include payload and overhead data including information addressing I^2C slave 106, in accordance with the I^2C-bus protocol. I^2C master device 104 controls the signal level on clock wire SBCK in accordance with a desired clock signal. The desired clock signal is used in common by I^2C master device 104 and I^2C slave device 106 to facilitate synchronous operation of the two devices during the transaction. The transaction is terminated by I^2C master 104 by providing a stop indication on data wire SBDT. The I^2C master 104 will inhibit any transaction between itself

and an I^2C slave 106 when it detects a low signal level on data wire SBDT that does not coincide with the intended signal level as instructed by I^2C master device 104. The ability of the new bus to operate in conjunction with the older I^2C bus is a beneficial aspect of the one embodiment of the invention.

Fig. 7 shows a timing diagram illustrating an interrupt transfer mode message (ITM) format of the illustrated serial bus interface protocol. A schematic presentation of an ITM transaction 110 is provided together with waveforms representing the signals on clock wire SBCK, start/stop wire SBST, and data wire SBDT. The illustrated ITM transaction 110 includes a start indication 112, a transfer mode identifier 114, a slave address 116, an encoded message 118, a clock rest period 120, and a stop indication 122. Start indication 112 triggers a first bit transmitted on data line SBDT, the first bit being latched on the second falling edge of the internal clock signal of the master device (such signal is not shown explicitly in Fig. 7) after the start/stop signal has gone low. Accordingly, the first bit of data is shown in Fig. 7 as starting at the start time 124. This is the transaction start time.

Following transaction start time 24, a transfer mode is transmitted from the master device to all slave devices utilizing the same protocol and connected to the same serial bus. In the illustrated embodiment, transfer mode identifier 114 comprises a pair of bits which are 00 indicating that the master device is initiating a transaction in an interrupt mode. Accordingly, the signal level on data wire SBDT is low for two clock cycles until a slave address start time 126 is reached which corresponds to a falling edge of clock signal SBCK. A one bit slave address 116 is then transmitted. Accordingly, only two receivers (which may comprise masters or slaves) can be addressed in the interrupt mode. The interrupt transfer mode (ITM) is used to transfer only one byte of encoded information. An ITM message may be used as a write-only message used by one master to signal an interrupt to another master. The encoded message 118 comprises five transmitted data bits. Following encoded message 118, a clock rest period 120 is shown, after which a stop indication 122 is signaled on start-stop wire SBST.

As described previously, when more than one master device is connected to a serial bus, a master device will yield to another master device which is the first to transmit a zero on the data wire. Accordingly, an ITM transaction will take priority over other transactions, since the first two bits of information (transfer mode identifier 114) transmitted on data wire SBDT are zeros -- i.e., low voltage levels, which preclude other master devices which are not transmitting

two consecutive zeros as their initial bits fail to maintain the signal level on data wire SBDT in their intended state.

Referring back to Fig. 5, a master device 50 will control a signal level on data wire SBDT via a data operation terminal 84, and will simultaneously monitor the signal level on data wire SBDT via a data monitoring terminal 82. When the signal level on data wire SBDT does not correspond to the intended signal level per the operation of data operation terminal 84 in connection with field effect transistor 74, master controller 56 of master device 50 will release the bus. Accordingly, a master transmitting a zero always wins an arbitration over a master transmitting a one. For this reason, an ITM transaction is appropriate for one master device to interrupt another master device or to otherwise supersede another master device's control over the serial bus.

Fig. 8 shows a transaction and corresponding waveforms corresponding to a fast transfer mode (FTM) transaction. An FTM transaction 134 is illustrated which comprises five bytes of transmitted information. Specifically, the illustrated FTM transaction 134 starts upon the occurrence of a start indication 136 which is followed by an initial two bits of information ("01") transmitted on data wire SBDT, i.e., transfer mode identifier 138, followed by a six bit slave address 140 which completes the first byte of transmitted information. A first clock rest period 141a then follows. The second transmitted byte includes an initial R/W (read/write) bit 142 followed by a seven bit register address 144. A second clock rest period 141b then separates another byte of information transmitted via data wire SBDT. During each clock rest period 141a - 141e, the level of the signal on data wire SBDT is high.

The first set of information transmitted between the transmitter and the receiver comprises a first byte of data 146. This data may be sent from the master device (serving as transmitter) to the slave device (serving as receiver) or transferred from the slave device (serving as transmitter) to the master device (serving as receiver) depending upon the state of the previously transmitted read/write bit 142. A second read/write bit 147 is then transmitted by the master device as an initial bit followed by a register address 148 to which data is to be forwarded or from which data is to be retrieved. If data is being read from the identified register, the slave device will drive the data wire SBDT during the following byte of data 150, which will commence upon completion of another clock rest period 141d. Near the end of the transmission, a final clock rest period 141e will occur, and the transaction will end upon a stop indication being signaled by the master device on start/stop wire SBST.

FTM transactions are intended for data transfer to and from slave devices which may not support the other transfer modes. Transmissions to such slave devices are of moderate priority, and are thus preceded by a transfer mode identifier of "01". In this mode, data may be both read to and written from the slave device within the same transaction.

Fig. 9 shows a bulk transfer mode (BTM) transaction together with the signals on each of the clock, start/stop and data wires. The illustrated BTM transaction 160 commences with a start indication 162 followed by a 2 bit transfer mode identifier 164 and a slave address 166. The information transmitted in the illustrated BTM transaction 160 includes many bytes. The second byte, transmitted following a first clock rest period 169a, includes a read/write bit 167 followed by a seven-bit register address 168. The third byte of data 170 is then transmitted following a second clock rest period 169b. All remaining bytes are transferred in a similar fashion until the master asserts the SBST line, indicating the end of the transaction. The number of data bytes transferred only depends on the time limit of the protocol. A final clock rest period 169d then occurs followed by a stop indication placed on start/stop wire SBST. The first and second bytes (and subsequent bytes) of data 170, 172 may be either written to a particular register of a slave device, or may be read from a particular register of a slave device, in accordance with the value placed within the read/write bit 167. More specifically, in the illustrated embodiment, when the read/write bit 167 is clear, i.e., the level of the signal on data wire SBDT is low, the master device is writing data to the slave device. When the read/write bit 167 is set, i.e., the signal level on data wire SBDT is high, the master device is reading data from the slave device.

When the master device wants to read data following a particular register address byte, it accordingly releases the data wire SBDT after the edge of the signal on clock wire SBCK which immediately follows the last bit transferred within the register address 168. Following each of the second and third clock rest periods 169b and 169c, the slave device then transmits the data. The master device continues to control the signal level on clock wire SBCK. The signal level on start/stop wire SBST remains low throughout the transaction.

In each of the above modes, the register address is a 7-bit field. This allows the addressing of up to 128 registers in a slave. In a bulk transfer mode the slave generates its own register addresses and therefore unlimited number of register addresses can be generated internal to the slave.

The master SBI controller of each master device may be configured to have two operating modes, including a functional mode and a test mode.

During the functional mode, the master device will only keep control of the bus for a limited time. In such a mode, the master device provides a stop indication on the designated clock wire SBCK when a threshold indicative of the length of the transaction has been reached. In the illustrated embodiment, that threshold is an amount of data transferred, i.e., when thirty two bytes have been transferred. In the test mode, the master device will be permitted unlimited data transfer. By limiting the amount of bytes transferable within a given transaction governed by a single master, this will allow other master devices to share the use of the serial bus.

Some general characteristics of the illustrated exemplary embodiments of the protocol disclosed herein may be described as follows:

(1) All changes in state on the data wire SBDT occur while the signal level on the clock wire SBCK is low.

(2) All transactions are initiated by pulling the signal on the start/stop wire SBST low, and are completed by taking the signal level on the start/stop wire SBST high.

(3) SBST, SBDT, and SBCK must all be high for at least one clock period (which would be 600ns should the clock rate be 1.53MHz) before a new transaction can begin. If any one of the three wires is low, a master cannot initiate a new transaction and must wait for all three signal levels to remain high for one clock period.

(4) Data transmission is always most significant bit (MSB) first to last significant bit (LSB) last.

(5) In the functional mode, no master device will keep control of the bus for more than 32 bytes of data transfer. While in the test mode, this constraint does not apply.

(6) A master releases its transmission on the serial bus if the data it transmits does not appear on the bus. The master transmitting a zero will always win an arbitration over a master transmitting a one. As described herein, this rule facilitates inter-master arbitration, and also facilitates the assignment of priorities to certain receivers and to certain types of modes of transfer, for example, as described above with respect to the interrupt mode, fast transfer mode, and bulk transfer mode. The interrupt transfer mode starts a transaction by transmitting two zeros on data wire SBDT, while the fast transfer mode starts a transaction by transmitting a zero followed by a one on data wire SBDT. A bulk transfer mode starts the transmission by transmitting a one followed by a zero on data wire SBDT. Accordingly, they are given priority in precisely that

order by the master devices which may be concurrently attempting to gain control of the serial data bus.

(7) Data transmission can be terminated at any time by taking a signal level at the start/stop wire SBST high. Slave devices and master devices should be provided with mechanisms for recovering from this condition.

(8) No master device or slave device can drive the data wire SBDT during a clock spacer time.

Fig. 10 is a block diagram further illustrating an exemplary master device 178 which may be coupled to the illustrated serial bus 48, for example, as shown in Fig. 5. The master device 178 as shown in Fig. 10 comprises a master serial bus interface (SBI) 180 which comprises a master SBI controller 181, a parallel processor interface 183, and a serial interface 185. Parallel processor interface 183 links master SBI controller 181 to a master device processor 182. Serial interface 185 links master SBI controller 181 to a serial bus -- i.e., serial bus 48 as shown in Fig. 5.

In the embodiment illustrated in Fig. 10, serial interface 185 comprises a clock wire interaction circuit 188, a start/stop wire interaction circuit 190, and a data wire interaction circuit 192. Each of these interaction circuits may be constructed as described above with respect to the master device 50 as shown in Fig. 5.

Parallel processor interface 183 comprises a primary interface 184 which in the illustrated embodiment includes a sixteen bit bi-directional data bus, and a secondary interface 186 provided for other connections between master device processor 182 and master SBI controller 181.

Master SBI controller 181 implements a serial link interface between master device processor 182 and the various slave devices which may be connected to the serial bus. Master SBI controller 181 may accordingly allow master device processor 182 to interact with the slave devices, for example, initializing, configuring, and selectively powering up specific functions of such slave devices. Master SBI controller 181 will further facilitate the ability of master device 182 to monitor the operation of the slave devices.

Parallel processor interface 183 may comprise any known microprocessor parallel interface that is either commercially available or constructed using techniques well known in the art. Master SBI controller 181 receives data from a parallel processor 183, and serializes addresses and data using a serial interface 185 comprising three pins, thereby implementing the serial bus interface protocol as described herein.

As noted above, parallel processor interface 183 comprises a primary interface 184 and a secondary interface 186. Primary interface 184 comprises a 16-bit bi-directional data bus which allows the transfer of pairs of assembled 8-bit SBI addresses and 8-bit data for transmission to and from slave devices connected to the serial bus. Master device processor 182 will have to address master SBI controller 181 periodically e.g., once every NSBI clock cycles to replenish a buffer provided within master SBI controller 181 with new address pairs for transmission to the appropriate slave devices.

Fig. 11 illustrates master SBI controller 181 in further detail. The illustrated master SBI controller 181 may comprise (among other elements, not specifically shown to simplify the explanation herein) a processor interface 194 coupled to a divider 196 and a data path block 198. Processor interface 194 receives at one side parallel processor 183. Data path block 198 interfaces with the serial bus via serial bus I/O pins 200. As shown in Fig. 11, parallel processor interface 183 comprises a plurality of parallel connections and control pins, among others, a micro_ reset input pin 202 and a clock CLK input pin 204. Micro_reset pin 202 is an input pin of processor interface 194 which allows the master device processor to reset master SBI controller 181. Clock pin CLK 204 comprises an input pin receiving a two-phase clock input from master device processor 182. The remaining data pin connections 206 comprise remaining pins of secondary processor connection 186 as well as of the primary interface 184 as shown in Fig. 10.

Processor interface 194 comprises several internal connections including a write enable connection WR_EN 208, a write data connection WR_DATA 210, a write address connection WR_ADDR 212, and a read data connection RD_DATA 214. Read data connection 214 is an input to processor interface 194, and the other connections 208, 210, and 212 are outputs. They are coupled, via internal buses of master SBI controller 181, to divider 196 and data path block 198.

The illustrated divider 196 comprises an internal bus connection 208 and a clock "CLK" input pin 210. It further comprises a first output 210 for outputting a serial bus clock signal MSBI_SBCK, and a second output 212 for outputting a master SBI controller enable signal MSBI_EN.

Data path device 198 has an internal bus connection 215 at one side thereof, and comprises a plurality of serial bus I/O interfacing pins 200 at an opposite side. Serial bus I/O interfacing pins 200 may comprise (among other pins not specifically shown for the sake of simplicity), a clock wire operation terminal 218, a start/stop operation terminal 220, a data monitoring terminal 222, and a data operation terminal 224.

Processor interface 194 asynchronously interfaces master SBI controller 181 to a processor bus while retaining synchronous operation for register reads and writes in master SBI controller 181. Divider block 196 subdivides a main CLK clock input, received at clock input 210, to produce a serial shift clock and appropriate enables for the operation of master SBI controller 181. Divider 196 may be provided with a mechanism for allowing serial operation with clock rates in a range of, for example, 1.5MHz down to 100 KHz. The illustrated divider 196 would facilitate such a range of clock rates and would enable master SBI controller 181 to function at clock rates defined with a resolution of 1/M sub-multiples of the main clock input CLK, with duty cycles of 40-50%.

Fig. 12 is a block diagram of a data path block 228 in further detail. Various details and specific elements that may form part of data path block 228 are omitted for purposes of simplifying the description herein. The illustrated data path block 228 comprises a plurality of write registers 238, a plurality of read registers 240, a multiplexer 242, and a central shift register 244. Data path block 228 further includes an SBI control register 246, a start control register 248, and an output portion 250. Plural write registers 238 comprise a write register 230 and a working buffer 232. Write register 230 comprises a one-byte register address portion 234a and a one-byte register data portion 234b. Register address and register data portions 234a and 234b comprise outputs respectively connected to a register address portion 236a and a register data portion 236b of working buffer 232. Working buffer 232 comprises a working buffer enable input 233 for receiving an enable signal. Each of the register address and register data portions 236a and 236b of working buffer 232 comprises an output which is input to a multiplexer 242. Multiplexer 242 further comprises a serial register selection input 243 receiving a serial register selection signal. Multiplexer 242 also receives a slave ID from SBI control register 246.

SBI start control register 248 comprises an output which is coupled to a control word input of SBI control register 246.

Plural read registers 240 comprises a register address portion 252 and an SBI read data portion 254. The information provided within read registers 240 is output via a read bus output 255. Read registers 240 further comprises a read register enable input 256 receiving a read register enable signal.

Shift register 244 comprises a data input terminal 258 which corresponds to data monitoring terminal 222 of data path device 198 as shown in Fig. 11. Shift register 244 further comprises a shift register enable input 260 for receiving a shift register enable signal. Shift register 244 comprises an output directed to the input side of an output multiplexer 262 which forms part of output portion

250. Output multiplexer 262 comprises a clock output terminal 264 which corresponds to clock wire operation terminal 218 as shown in Fig. 11, a start/stop output terminal 266 which corresponds to start/stop operation terminal 220, and a data output terminal 268 which corresponds to data operation terminal 224, as shown in Fig. 11.

Fig. 13 is a flowchart which illustrates certain operations performed by data path block 228 and other components of master SBI controller 181 in performing a multi-word transaction with a slave device through a serial bus as disclosed in the illustrated embodiment. In an initial step S2, the controller will write to a clock control register provided in divider 196 in order to pick a desired divide ratio. In step S4, the controller will write to SBI control register 246. In this write operation, the appropriate mode of operation of the controller will be chosen by writing the desired slave ID (SLV_ID) and serial bus protocol mode bits in SBI control register 246. In step S6, the controller will write to write register 230 (SBI_WR) the address and data to be transferred to a particular slave device. Thereafter, in step S8, the controller, when ready to start the transaction, will write to bit zero of the SBI start control register 248 to set the start_flag to the number 1. In step S10, the controller will first transfer the serial transfer mode bits and slave ID bits, and then transfer the contents of write register 230 to its working buffer 232. The controller will then assert its interrupt to let master device processor 182 know that the write register 230 is empty.

In step S12, the controller will serialize bits [15:8] and then bits [7:0] of working register 232.

In the next step S14, a determination is made as to whether write register 230 has been re-written. If write register 230 has been re-written, the process will return to step S10, at which point the data will be transferred to working buffer 232 and an interrupt will be sent to master device processor 182. If a determination is made at step S14 that write register 230 has not been written to, the transaction is terminated at step S16.

Controller 181 may be provided with a status register which gives a complete picture of the state of the controller with one read operation. Such a register would be readable by master device processor 182.

Fig 11 is a block diagram illustrating a slave device 270. The illustrated device 270 comprises a slave SBI controller 272 coupled to a read/write register 274 and a read register 276 by a bus structure 278. A serial interface is provided at the front end of slave SBI controller 272 which comprises a data wire interaction circuit 280, a clock wire interaction circuit 282, and a start/stop interaction circuit 284.

The illustrated slave 272 more specifically comprises a bi-directional 7-bit parallel databus connection 286 coupled to a 7-bit parallel bus 287, a 5-bit parallel address bus connection 288 coupled to an address bus 289, a write clock pin 290 coupled to a write clock bus (wire) 291, and a read enable pin 292 coupled to a read enable bus (wire) 293.

In operation, when a master device, e.g., the master device 178 shown in Fig. 10, addresses slave device 270 in order to write data to read/write register 274, master device processor 182 will instruct master SBI controller 181 to perform a write transaction, and will transfer the data to be written via primary interface 184 to master SBI controller 181. Master SBI controller 181 will then control the serialization process, first signaling a start indication on start/stop wire SBST via start/stop interaction circuit 190. Slave SBI controller 272 of slave device 270 will be in a ready state listening for such a start indication via start/stop interaction circuit 284. Slave SBI controller 272 will receive the clock signal via its clock interaction circuit 282. The transmitted data is received through data interaction circuit 280.

While the invention has been described by way of an exemplary embodiment, it is understood that the words which have been used herein are words of description, rather than words of limitation. Changes may be made, within the purview of the appended claims, without departing from the scope and spirit of the invention its broader aspects. Although the invention has been described herein with reference to particular structures, materials, and embodiments, it is understood that the invention is not limited to the particulars disclosed. The invention extends to all appropriate equivalent structures, mechanisms, implementations, and uses.

WHAT IS CLAIMED IS:

CLAIMS

1. An inter-device control link, comprising:
a serial bus comprising a clock wire, a data wire, and a start/stop wire;
a master serial bus interface coupling a master device to said serial bus;
a slave serial bus interface coupling a slave device to said serial bus;
said master serial bus interface comprising:
(a) a transaction initiator for initiating a transaction by providing a start indication on said start/stop wire;
(b) a master data write mechanism for controlling the signal level on said data wire in accordance with data to be written to said slave device;
(c) a master data read mechanism for reading data by monitoring the signal level on said data wire; and
(d) a clock driver for controlling the signal level on said clock wire in accordance with a desired clock signal;
said slave serial bus interface comprising:
(e) a slave data write mechanism for controlling the signal level on said data wire in accordance with slave-originating data to be written from said slave device to another device;
(f) a slave data read mechanism for reading data by monitoring the signal level on said data wire; and
(g) an internal clocking mechanism for clocking the controlling of the signal level on said data wire in accordance with a clock signal said slave serial bus interface receives from said master serial bus interface.

2. The inter-device control link according to claim 1, wherein said transaction initiator comprising a mechanism for pulling the signal level on said start/stop wire low.

3. The inter-device control link according to claim 1, wherein said master data write mechanism comprises a mechanism for controlling the signal level on said data wire in accordance with data including payload data and overhead data, said overhead data including information addressing said slave device.

4. The inter-device control link according to claim 3, wherein said information addressing said slave device further specifies a register within said slave device.

5. The inter-device control link according to claim 1, wherein said desired clock signal is used in common by said master device and said slave device to facilitate synchronous operation of said master device and said slave device.

6. The inter-device control link according to claim 1, further comprising a pull-up circuit connected to said data wire and said clock wire, wherein said master data write mechanism comprises a data write driver which indicates on said data wire a zero by pulling down a voltage level on said data wire and indicates on said data wire a one by tri-stating said data write driver thereby letting said pull-up circuit take the voltage level of said data wire high, and wherein said clock driver comprises a clock signal driver which indicates on said clock wire a zero by pulling down a voltage level of said clock wire and indicates a one by tri-stating said clock signal driver thereby letting said pull-up circuit take the voltage level of said clock wire high.

7. The inter-device control link according to claim 1, comprising at least one additional master device, said additional master device and said master device competing for use of said bus, said additional master device comprising:

(a) a transaction initiator for initiating a transaction by providing a start indication on said start/stop wire;

(b) a master data write mechanism for controlling the signal level on said data wire in accordance with data to be written to said slave device;

(c) a master data read mechanism for reading data by monitoring the signal level on said data wire; and

(d) a clock driver for controlling the signal level on said clock wire in accordance with a desired clock signal;

said master device and said additional master device each comprising a data wire monitor for monitoring the signal level on said data wire and for inhibiting operation of said master device or said additional master device when the monitored signal level does not match on intended signal level;

said data write mechanism of each of said master device and said additional master device comprising a slave addressing mechanism for preceding a transmitted slave address with a sequence of ones and zeros in accordance with the priority of the transaction to take place.

8. The inter-device control link according to claim 7, wherein said transaction initiator of said master serial bus interface of said master device and said transaction initiator of said master serial bus interface of said additional

master device each comprises a mechanism for providing the start indication on said start/stop wire by pulling the signal level on said start/stop wire low.

9. The inter-device control link according to claim 7, wherein said master data write mechanism of said master serial bus interface of each of said master device and said additional master device comprises a mechanism for writing data including payload data and overhead data, said over-head data including information addressing said slave device and specifying a register within said slave device.

10. The inter-device control link according to claim 7, wherein said desired clock signal of said clock driver of said master serial bus interface of each of said master device and said additional master device comprises a signal being used in common by said master device or said additional master device and said slave device to facilitate synchronous operation of said master device or said additional master device and said slave device.

11. The inter-device control link according to claim 7, wherein said sequence of ones and zeros comprises at least two zeros for transactions involving a broadcast to all slaves connected to said bus.

12. A method for communicating among devices each connected to a serial bus, said devices comprising a first type master device, a first type slave device, a second type master device, and a second type slave device, said first type master and slave devices utilizing a first serial bus protocol, said second type master and slave devices utilizing a second serial bus protocol, said serial bus comprising a set of bus wires including a data bus wire and a clock bus wire, said method comprising:

(a) said first type master device initiating a transaction by providing a start indication on a designated bus wire designated for the transmission of transaction start and stop indications;

(b) said first type of master device controlling the signal level on said data bus wire in accordance with data to be transferred to said first type slave device, said data including payload data and overhead data including information addressing said first type slave device;

(c) controlling the signal level on said clock bus wire in accordance with a desired clock signal, said desired clock signal being used in common by

said master device and said first type slave device to facilitate synchronous operation of said master device and said slave device;

(d) said first type master device terminating a transaction by providing a stop indication on said designated bus wire and maintaining said stop indication, said first type master device inhibiting the control by said first type master device of the signal on said data bus wire and inhibiting the control by said first type master device of the signal level on said clock bus wire for as long as said stop indication is provided and maintained by said master device;

(e) said first type slave device refraining from operating for as long as said stop indication is provided and maintained by said first type master device:

(f) said second type master device initiating a transaction by providing a start indication on said data bus wire;

(g) said second type master device controlling the signal level on said data bus wire in accordance with data to be transferred to said second type slave device, said data including payload data and overhead data including information addressing said second type slave device;

(h) said second type master controlling the signal level on said clock bus wire in accordance with a desired clock signal, said desired clock signal being used in common by said master device and said slave device to facilitate synchronous operation of said master device and said slave device;

(i) said second type master device terminating a transaction by providing a stop indication on said data bus wire;

(j) said second type master device inhibiting a transaction between said second type master device and said second type slave device.

13. The method according to claim 12, wherein said second serial bus protocol comprises the I^2C-bus protocol.

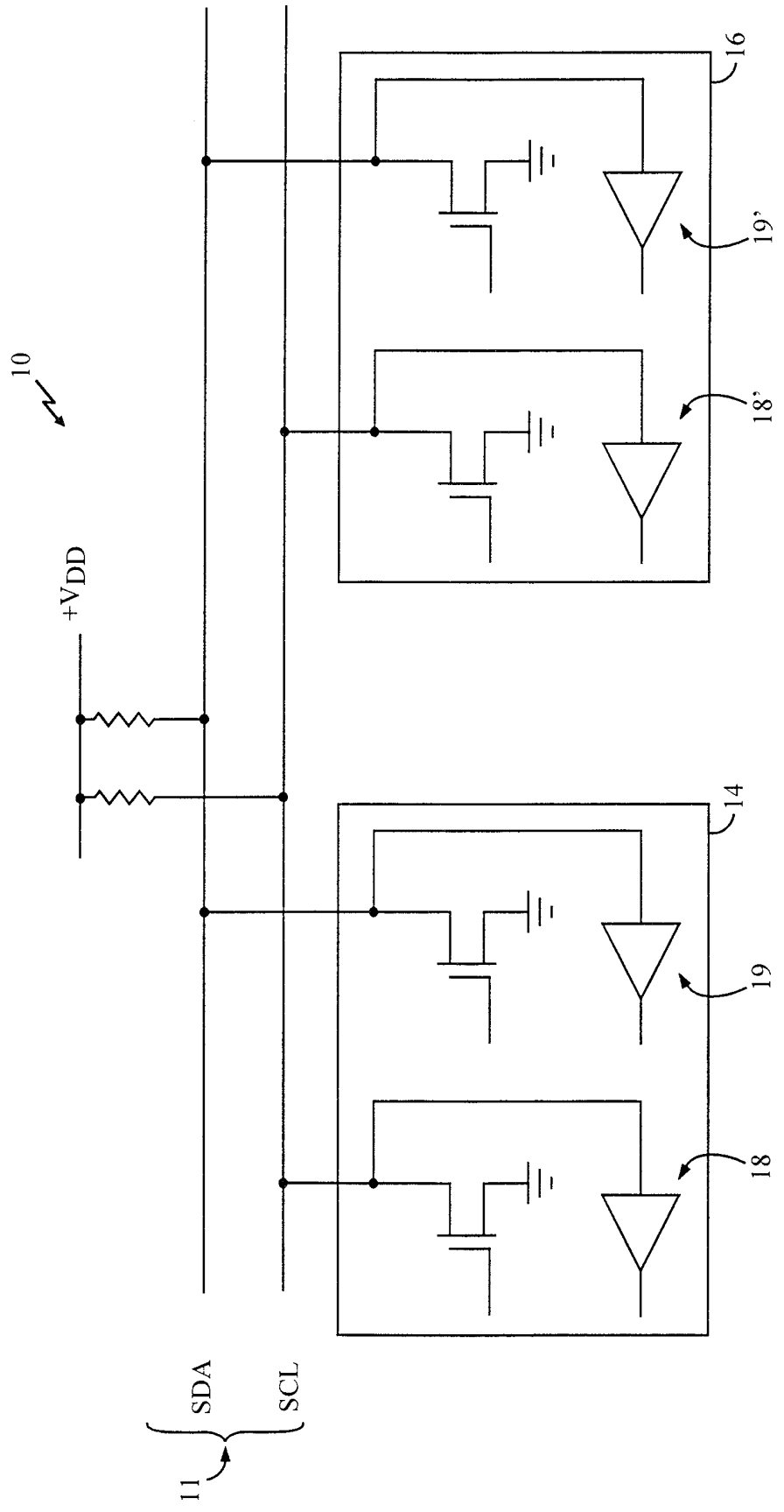
14. The method according to claim 12, wherein said designated bus wire comprises a bus wire separate and distinct from said data and clock bus wires.

15. The method according to claim 12, wherein said first type master device maintains said stop indication irrespective of the existence of any indications present on said data or clock buses.

16. The method according to claim 12, wherein said first type master device provides a stop indication on said designated bus wire when a threshold

indicative of the length of the transaction between said first type master and slave devices has been reached.

17. The method according to claim 16, wherein said threshold comprises an amount of data transferred in said transaction.



BACKGROUND ART

FIG. 1

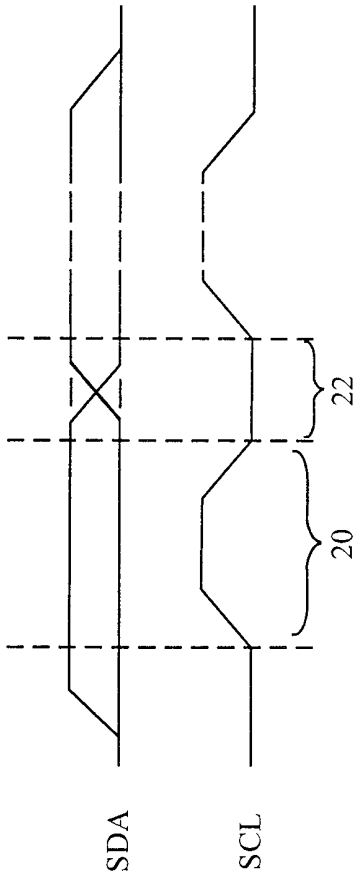


FIG. 2
BACKGROUND ART

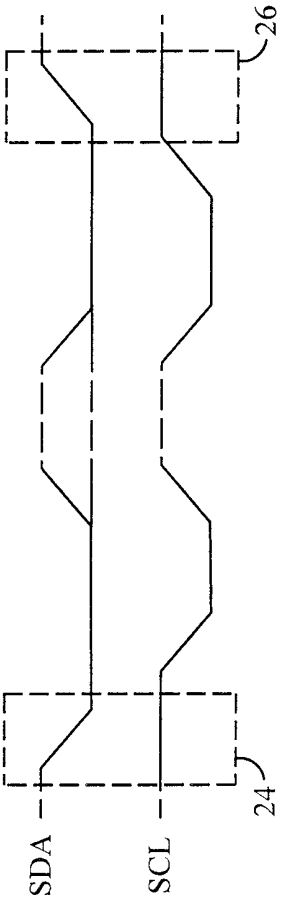


FIG. 3
BACKGROUND ART

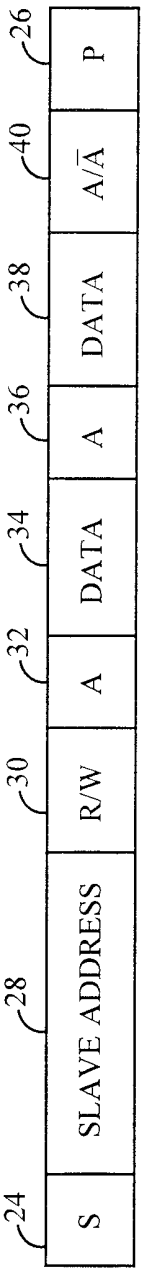


FIG. 4
BACKGROUND ART

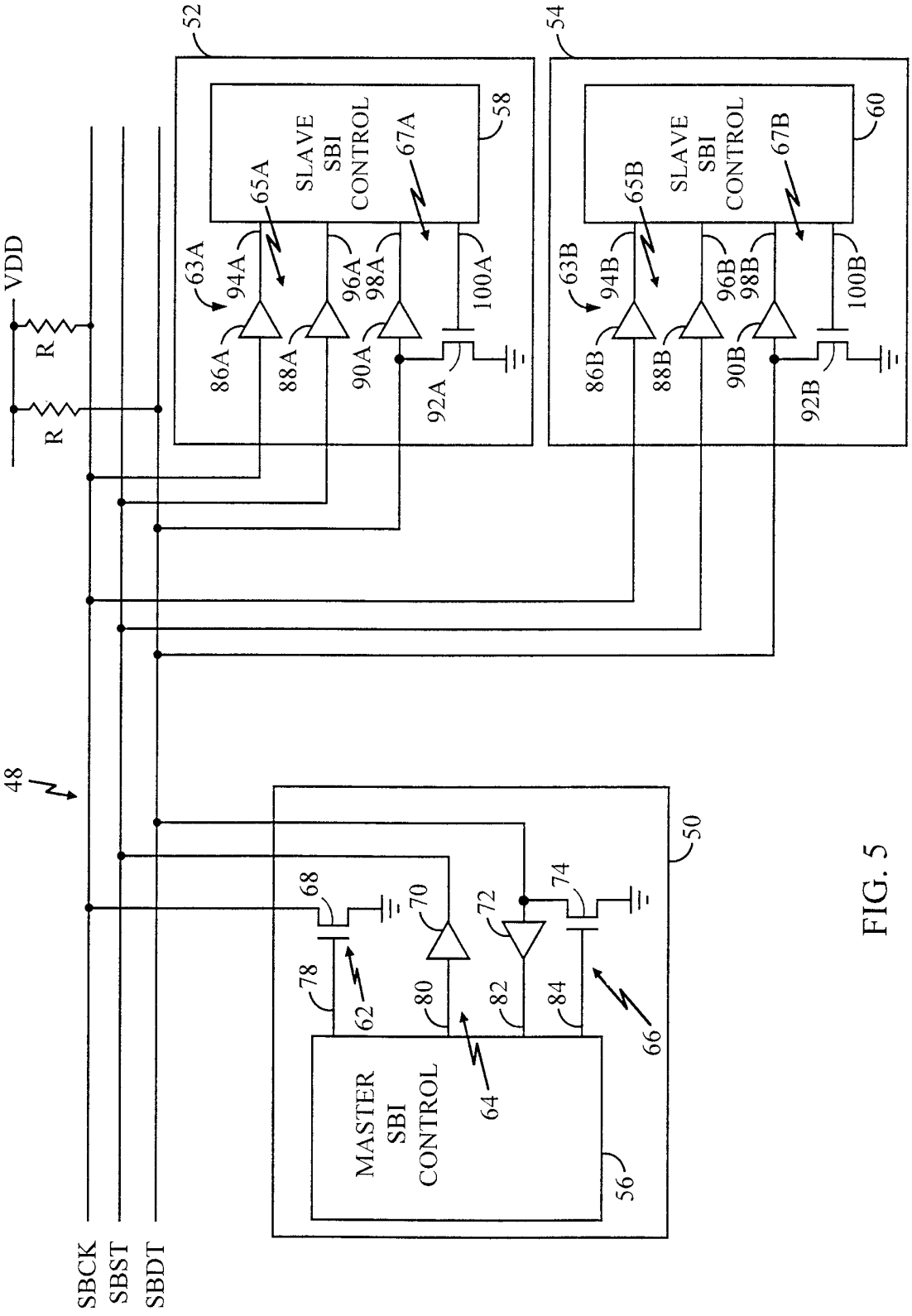

52
54
58
60
SLAVE SBI CONTROL
65A
67A
65B
67B
63A
63B
94A
94B
96A
96B
98A
98B
100A
100B
86A
86B
88A
88B
90A
90B
92A
92B
VDD
R
R
48
50
MASTER SBI CONTROL
56
62
64
66
68
70
72
74
78
80
82
84
SBCK
SBST
SBDT

FIG. 5

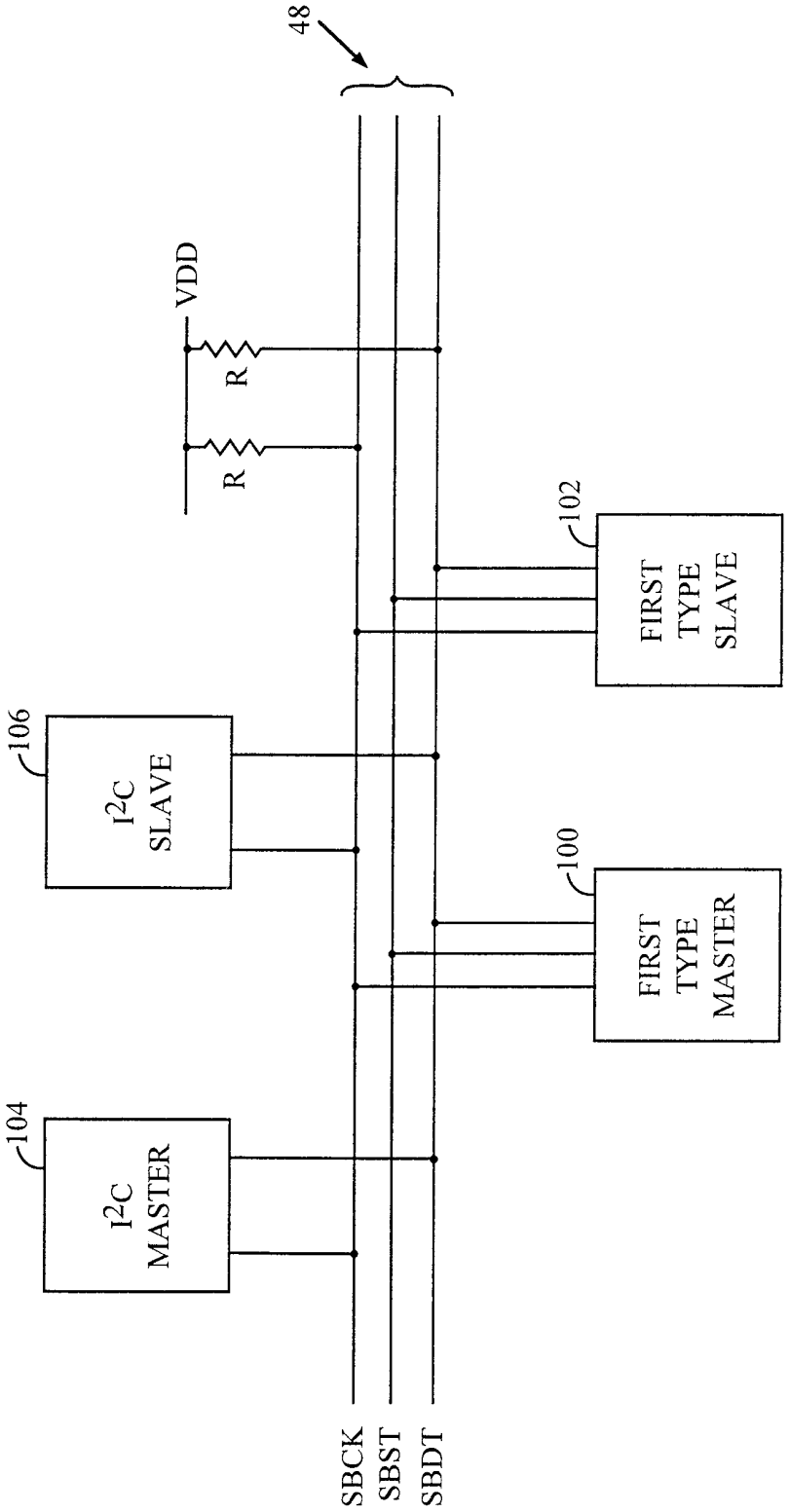

48
VDD
R
R
102
FIRST TYPE SLAVE
106
I2C SLAVE
100
FIRST TYPE MASTER
104
I2C MASTER
SBCK
SBST
SBDT

FIG. 6

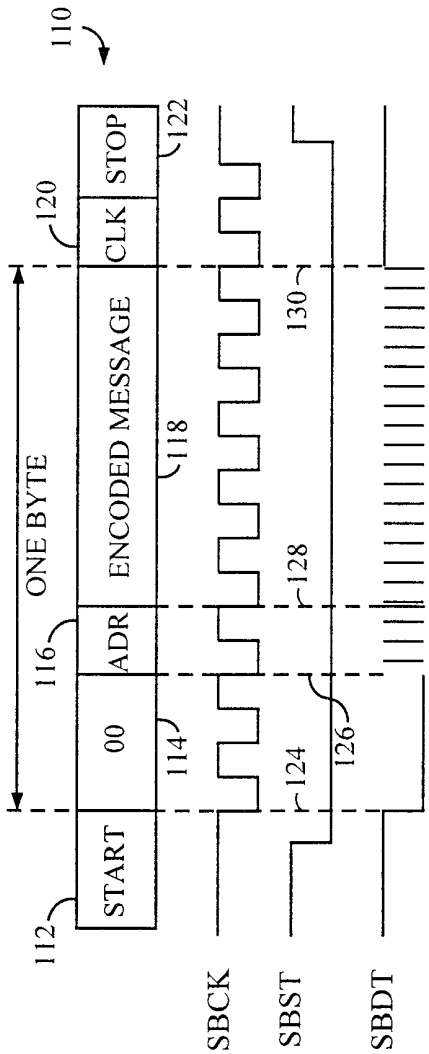

110
ONE BYTE
START
00
ADR
ENCODED MESSAGE
CLK
STOP
112
114
116
118
120
122
SBCK
SBST
SBDT
124
126
128
130

FIG. 7

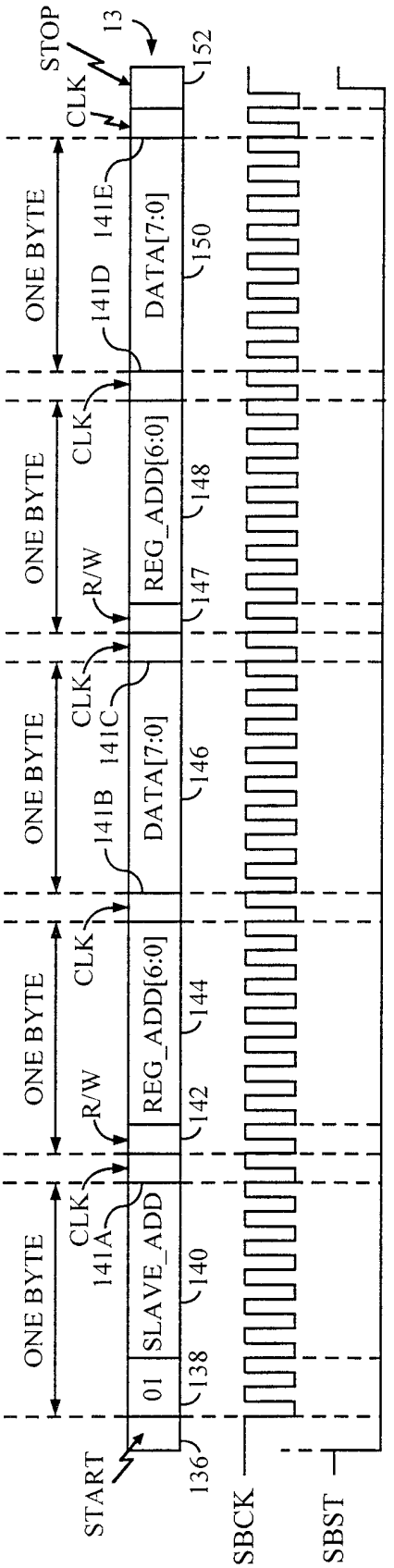

START
01
SLAVE_ADD
R/W
REG_ADD[6:0]
DATA[7:0]
R/W
REG_ADD[6:0]
DATA[7:0]
STOP
CLK
ONE BYTE
136
138
140
141A
142
144
141B
141C
146
147
148
141D
141E
150
152
SBCK
SBST

FIG. 8

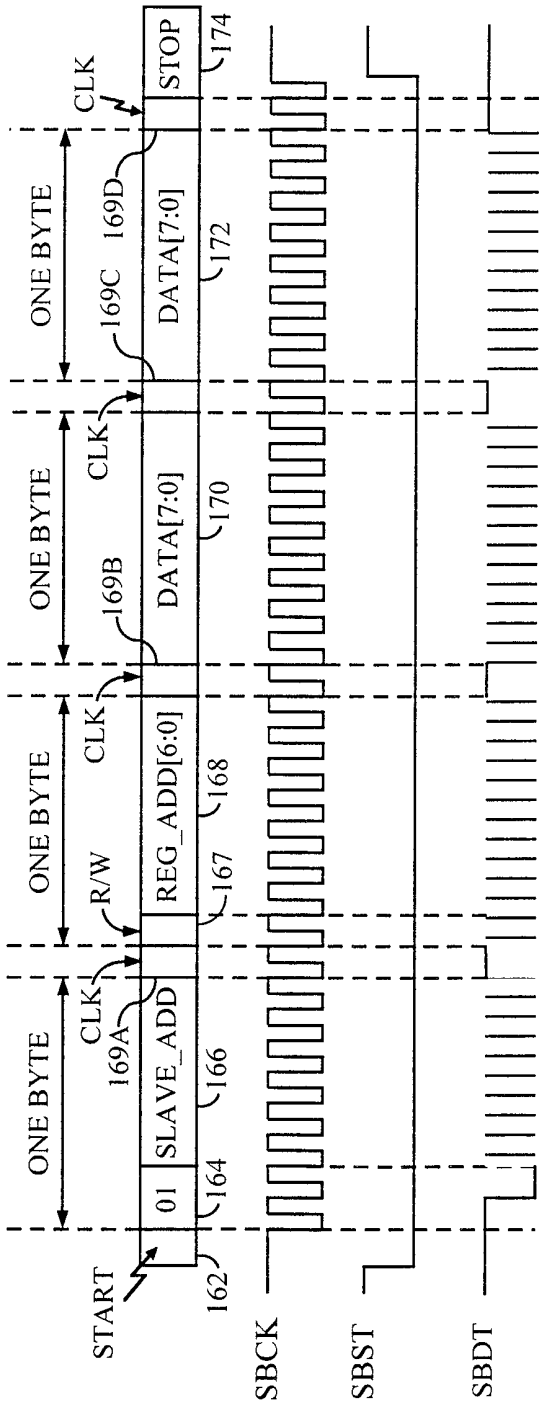

START
01
SLAVE_ADD
R/W
REG_ADD[6:0]
DATA[7:0]
DATA[7:0]
STOP
CLK
CLK
CLK
CLK
ONE BYTE
ONE BYTE
ONE BYTE
ONE BYTE
162
164
166
167
168
169A
169B
169C
169D
170
172
174
SBCK
SBST
SBDT

FIG. 9

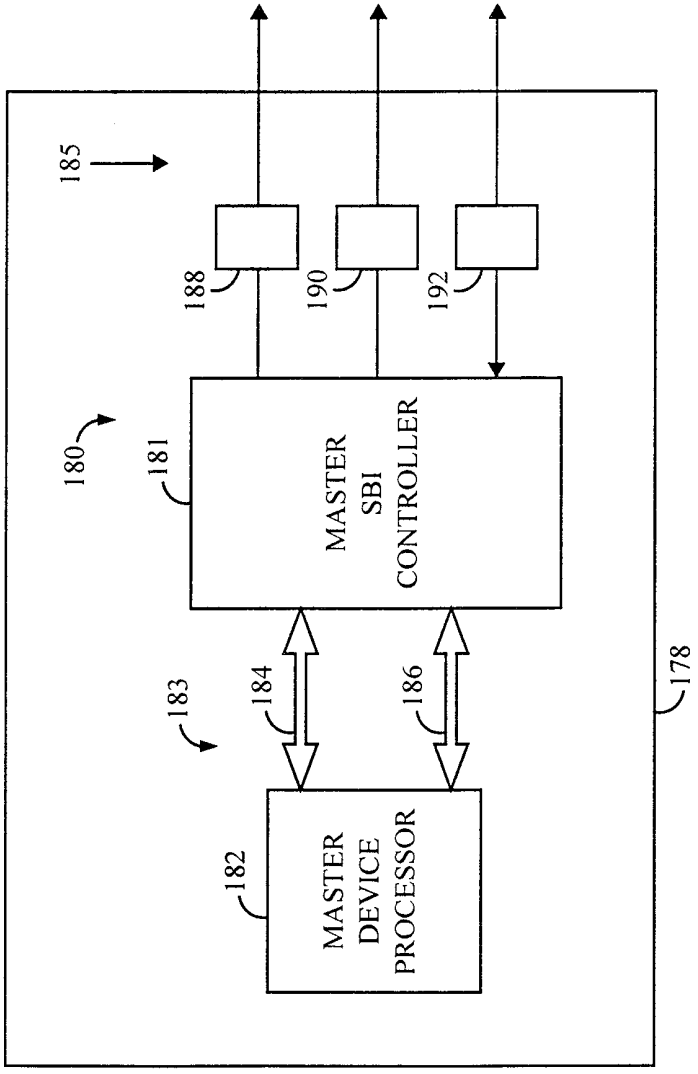

185
188
190
192
180
181
MASTER SBI CONTROLLER
183
184
186
178
182
MASTER DEVICE PROCESSOR

FIG. 10

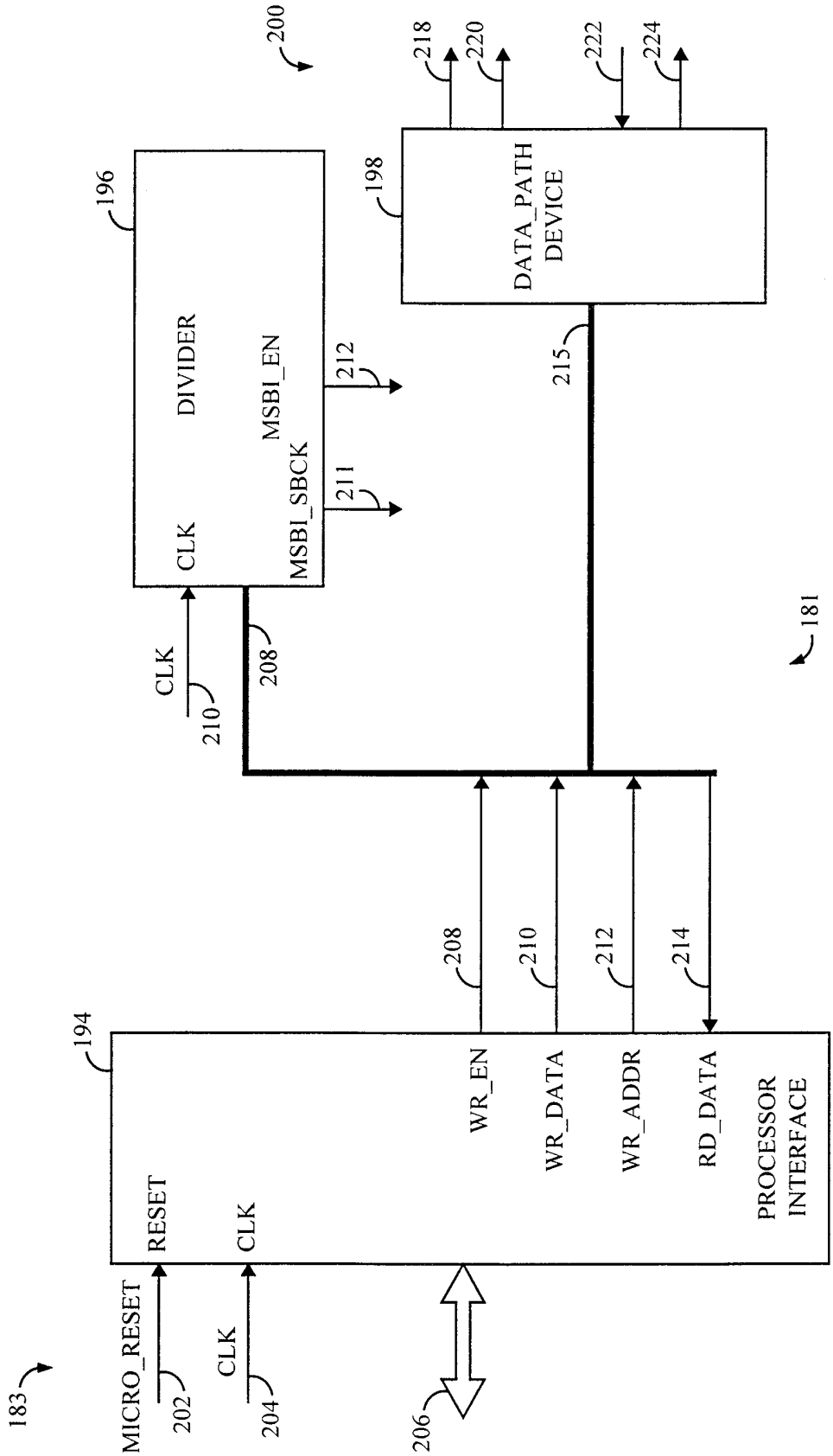

183
MICRO_RESET
202
CLK
204
206
194
RESET
CLK
WR_EN
WR_DATA
WR_ADDR
RD_DATA
PROCESSOR INTERFACE
208
210
212
214
CLK
210
208
196
CLK
DIVIDER
MSBI_EN
MSBI_SBCK
212
211
215
198
DATA_PATH DEVICE
218
220
222
224
200
181

FIG. 11

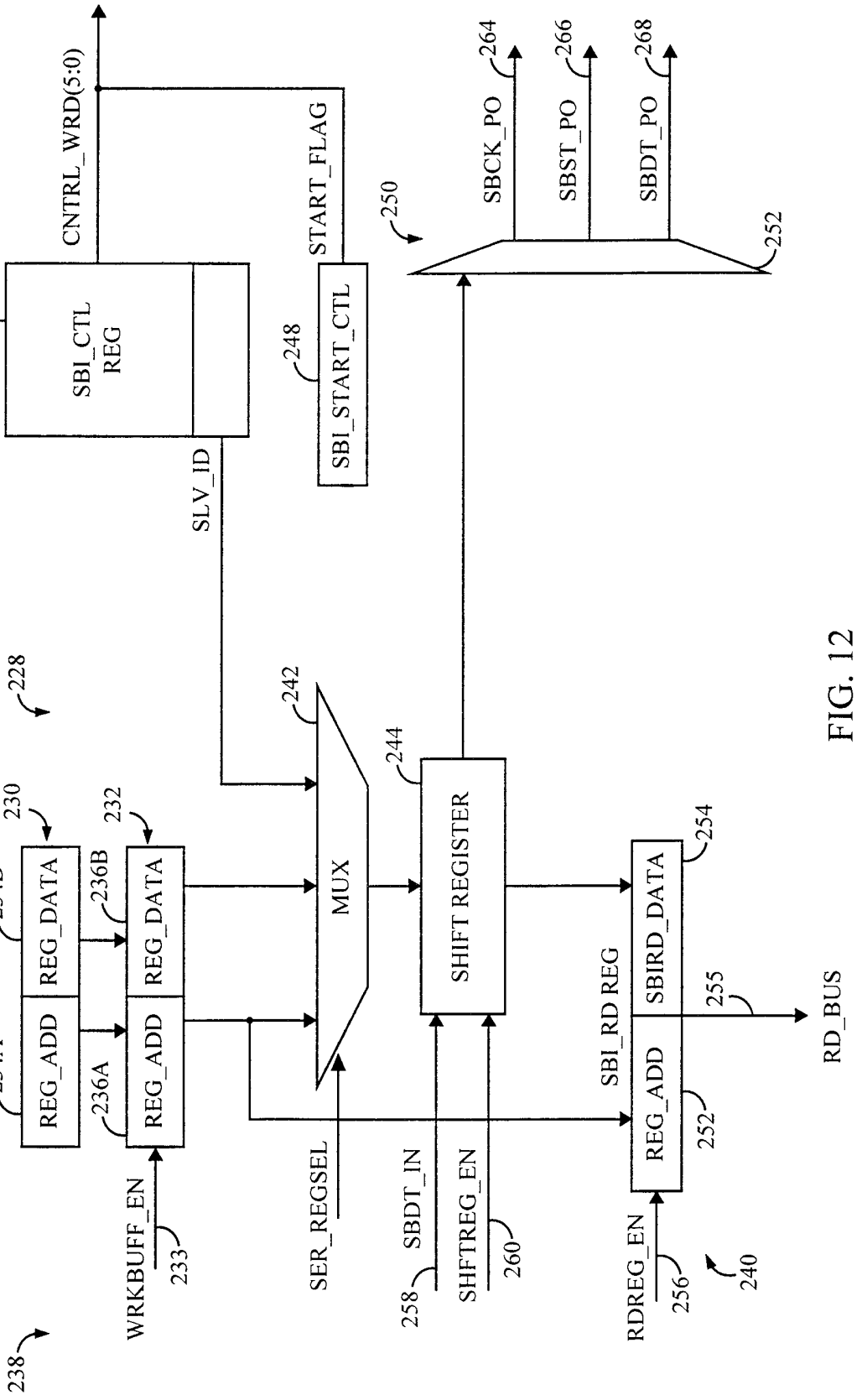

CNTRL_WRD(5:0)
START_FLAG
SBCK_PO
SBST_PO
SBDT_PO
246
SBI_CTL REG
248
SBI_START_CTL
250
252
SLV_ID
228
242
MUX
244
SHIFT REGISTER
230
232
234B
REG_DATA
236B
REG_DATA
234A
REG_ADD
236A
REG_ADD
254
SBIRD_DATA
255
RD_BUS
SBI_RD REG
REG_ADD
252
WRKBUFF_EN
233
SER_REGSEL
SBDT_IN
258
SHFTREG_EN
260
RDREG_EN
256
240
238

FIG. 12

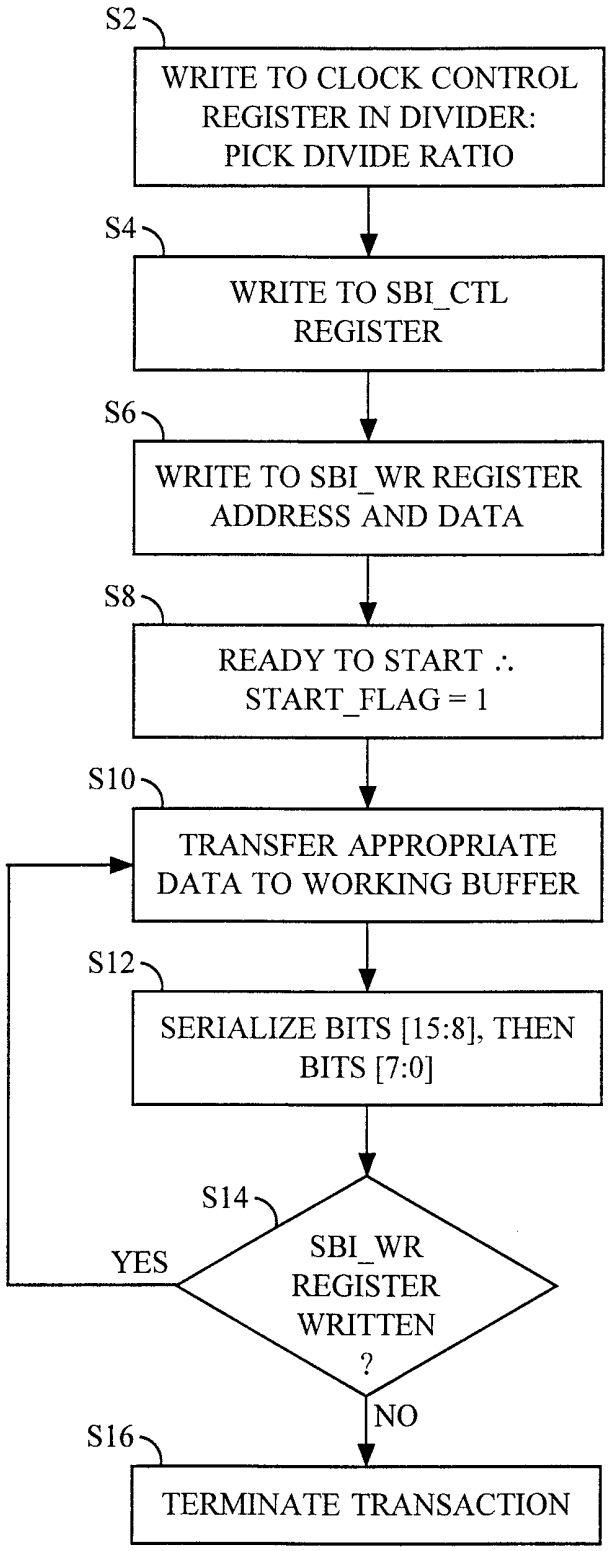

S2
WRITE TO CLOCK CONTROL REGISTER IN DIVIDER: PICK DIVIDE RATIO
S4
WRITE TO SBI_CTL REGISTER
S6
WRITE TO SBI_WR REGISTER ADDRESS AND DATA
S8
READY TO START ∴ START_FLAG = 1
S10
TRANSFER APPROPRIATE DATA TO WORKING BUFFER
S12
SERIALIZE BITS [15:8], THEN BITS [7:0]
S14
SBI_WR REGISTER WRITTEN ?
YES
NO
S16
TERMINATE TRANSACTION

FIG. 13

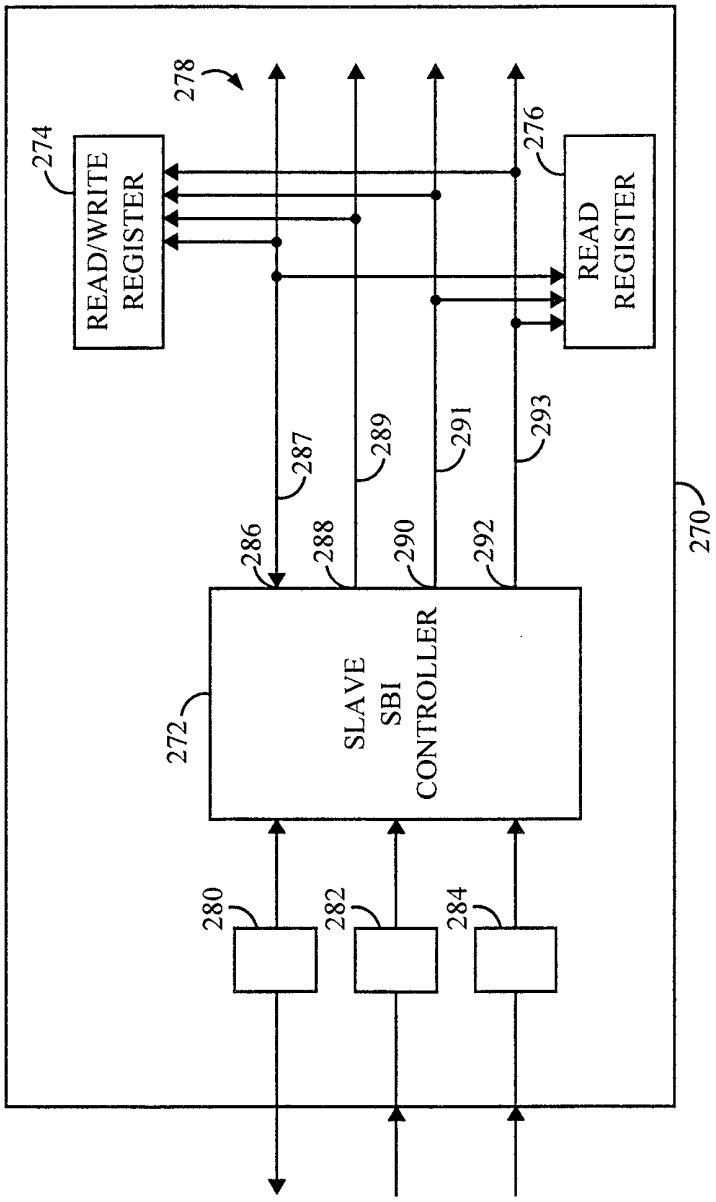

278
274
READ/WRITE REGISTER
276
READ REGISTER
287
289
291
293
286
288
290
292
270
272
SLAVE SBI CONTROLLER
280
282
284

FIG. 14