

CLAIMS

1 1. A method of communicating with a master over a shared bus having a data line,
2 comprising:
3 receiving a request signal from a master requesting a slave address from each slave
4 device coupled to the data line be sent to the master;
5 causing the data line to be sequentially placed in logic states corresponding to bit
6 values in a first slave address; and
7 upon the data line being placed in a logic state that is different from a corresponding
8 bit value of the first slave address, temporarily entering an idle state until another slave
9 device has completed sending a slave address thereof to the master.

1 2. The method of claim 1, further comprising entering the idle state when all bits of
2 the first slave address have been placed on the data line.

1 3. The method of claim 1, wherein the causing comprises driving the data line to a
2 first logic state when the corresponding bit value of the first slave address is the first logic
3 state, and releasing the data line when the corresponding bit value of the first slave address is
4 a second logic state.

1 4. The method of claim 3, wherein the first logic state is a logic zero state and the
2 second logic state is the logic one state.

1 5. The method of claim 1, further comprising monitoring the logic state of the data
2 line and determining whether the monitored logic state of the data line is the same as the
3 corresponding bit value of the first slave address, wherein entering the idle state is based
4 upon the determination.

1 6. The method of claim 1, wherein the causing is performed in a serial manner from
2 most significant bit of the first slave address to least significant bit thereof.

1 7. The method of claim 1 further comprising counting a first number of clock cycles
2 from a time when the causing began, wherein entering the idle state comprises entering the
3 idle state for a second number of clock cycles, the second number of clock cycles
4 corresponding to a number of bits in the first address less the first number of clock cycles.

1 8. The method of claim 7, further comprising exiting the idle state upon completion
2 of the second number of clock cycles, repeating the act of causing unless the data line is again
3 placed in a logic state that is different from a corresponding bit value of the first slave address
4 and reentering the idle state in response thereto.

1 9. The method of claim 8, further comprising, upon completion of causing the data
2 line to be sequentially placed in the logic state corresponding to each bit value of the first
3 slave address, entering the idle state until an indication from the master is received that all
4 slave addresses have been received thereby.

1 10. A slave device, comprising:
2 an interface port for coupling to a shared bus having a clock line and a data line;
3 nonvolatile memory for storing a first slave address corresponding to the slave device;
4 a controller communicatively coupled to the interface port and to the nonvolatile
5 memory, the controller configured to:
6 upon the interface port receiving a request signal from a master requesting that
7 a slave address of each slave device that is coupled to the shared bus be sent to the master,
8 controlling the interface port to cause, in a serial manner, the data line to be placed in logic
9 states corresponding to bit values in the first slave address; and
10 upon the data line being placed in a logic state that is different from a
11 corresponding bit value of the first slave address, controlling the interface port to temporarily
12 enter an idle state until another slave device has completed sending a slave address thereof to
13 the master.

1 11. The slave device of claim 10, wherein the interface port drives the data line to a
2 first logic state when a corresponding bit value of the first slave address is the first logic state,
3 and releases the data line when the corresponding bit value of the first slave address is a
4 second logic state.

1 12. The slave device of claim 10, wherein the controller is configured to determine
2 whether another slave device caused the data line to be placed in a logic state that is different
3 from the corresponding bit value of the first slave address, and to enter the idle state in
4 response.

1 13. The slave device of claim 10, wherein the interface port enters the idle state
2 following a completion of the first slave address being placed on the data line.

1 14. The slave device of claim 10, wherein following the another slave device sending
2 the slave address thereof to the master, the controller controls the interface port to cause the
3 data line to be placed in logic states corresponding to bit values in the first slave address.

- 1 15. The slave device of claim 10, wherein the interface port causes the first address to
2 be serially placed on the data line from most significant bit to least significant bit.

1 16. A method of communicating with a master over a shared bus having a data line,
2 comprising:

3 receiving a request signal from a master requesting that a slave address from each
4 slave device that is coupled to the data line be sent to the master;

5 causing the data line to be sequentially placed in logic states corresponding to bit
6 values in a first slave address;

7 monitoring the data line;

8 based upon the monitoring, determining that another slave device coupled to the data
9 line has a slave address that is less than a value of the first slave address; and

10 entering an idle state based upon the determining.

CLAIMS

1. A method of communicating with a master over a shared bus having a data line, comprising:

5 receiving, by slave device circuitry, a request signal from a master requesting a slave address from each slave device coupled to the data line be sent to the master;

controlling, by the slave device circuitry, the data line for the data line to be sequentially placed in logic states corresponding to bit values in a first slave address, the first slave address being the slave address for the slave device to which the slave device circuitry is associated;

10 and

upon the data line being placed in a logic state that is different from a corresponding bit value of the first slave address, the slave device circuitry temporarily entering an idle state until another slave device has completed sending a slave address thereof to the master.

15 2. The method of claim 1, further comprising entering, by the slave device circuitry, the idle state when all bits of the first slave address have been placed on the data line.

3. The method of claim 1, wherein the controlling comprises driving the data line to a first logic state when the corresponding bit value of the first slave address is the first logic state, and releasing the data line when the corresponding bit value of the first slave address is a second logic state.

4. The method of claim 3, wherein the first logic state is a logic zero state and the second logic state is the logic one state.

25 5. The method of claim 1, further comprising monitoring the logic state of the data line during the controlling and determining whether the monitored logic state of the data line is the same as the corresponding bit value of the first slave address, wherein entering the idle state is based upon the determination.

30 6. The method of claim 1, wherein the controlling is performed in a serial manner from most significant bit of the first slave address to least significant bit thereof.

7. The method of claim 1 further comprising counting a first number of clock cycles from a time when the controlling began, wherein entering the idle state comprises entering the idle state for a second number of clock cycles, the second number of clock cycles corresponding to a number of bits in the first address less the first number of clock cycles.

5

8. The method of claim 7, further comprising exiting the idle state upon completion of the second number of clock cycles, repeating the act of controlling unless the data line is again placed in a logic state that is different from a corresponding bit value of the first slave address and reentering the idle state in response thereto.

10

9. The method of claim 8, further comprising, upon completion of controlling the data line to be sequentially placed in the logic state corresponding to each bit value of the first slave address, entering the idle state until an indication from the master is received that all slave addresses have been received thereby.

15

10. A slave device, comprising:

an interface port for coupling to a shared bus having a clock line and a data line;

nonvolatile memory for storing a first slave address corresponding to the slave device;

a controller communicatively coupled to the interface port and to the nonvolatile memory,

20 the controller configured to:

upon the interface port receiving a request signal from a master requesting that a slave address of each slave device that is coupled to the shared bus be sent to the master, controlling the interface port to cause, in a serial manner, the data line to be placed in logic states corresponding to bit values in the first slave address; and

25

upon the data line being placed in a logic state that is different from a corresponding bit value of the first slave address, controlling the interface port to temporarily enter an idle state until another slave device has completed sending a slave address thereof to the master.

30

11. The slave device of claim 10, wherein the interface port drives the data line to a first logic state when a corresponding bit value of the first slave address is the first logic state, and releases the data line when the corresponding bit value of the first slave address is a second logic state.

12. The slave device of claim 10, wherein the controller is configured to determine whether another slave device caused the data line to be placed in a logic state that is different from the corresponding bit value of the first slave address, and to enter the idle state in response.

5

13. The slave device of claim 10, wherein the interface port enters the idle state following a completion of the first slave address being placed on the data line.

14. The slave device of claim 10, wherein following the another slave device sending the slave address thereof to the master, the controller controls the interface port to cause the data line to be placed in logic states corresponding to bit values in the first slave address.

10

15. The slave device of claim 10, wherein the interface port causes the first address to be serially placed on the data line from most significant bit to least significant bit.

15

16. A method of communicating with a master over a shared bus having a data line, comprising:

receiving a request signal from a master requesting that a slave address from each slave device that is coupled to the data line be sent to the master;

causing the data line to be sequentially placed in logic states corresponding to bit values in a first slave address;

20

monitoring the data line;

based upon the monitoring, determining that another slave device coupled to the data line has a slave address that is less than a value of the first slave address; and

25

entering an idle state based upon the determining.