(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
9 October 2003 (09.10.2003)

PCT

(10) International Publication Number
WO 03/083677 A2

(51) International Patent Classification[7]: G06F 13/00

(21) International Application Number: PCT/US03/09636

(22) International Filing Date: 26 March 2003 (26.03.2003)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/368,474 28 March 2002 (28.03.2002) US
10/144,333 13 May 2002 (13.05.2002) US

(71) Applicant: **ADVANCED ANALOGIC TECHNOLOGIES, INC.** [US/US]; Dangelo, Kevin, 830 East Arques Avenue, Sunnyvale, CA 94085-4519 (US).

(72) Inventors: **D'ANGELO, Kevin**; 1516 Homestead Road, Santa Clara, CA 95050-4721 (US). **BROWN, David, Alan**; 123 Wood Road, Los Gatos, CA 95030 (US). **SUNG, John, K.**; 34785 Monaco Common, Freemont, CA 94555 (US). **NILSSON, Jan**; 792 Lusterleaf Drive, Sunnyvale, CA 94086 (US). **WILLIAMS, Richard, K.**; 10292 Norwich Ave., Cupertino, CA 95014 (US).

(74) Agent: **HOLLINGER, Joseph**; 3035 Baker Street #301, San Francisco, CA 94123 (US).

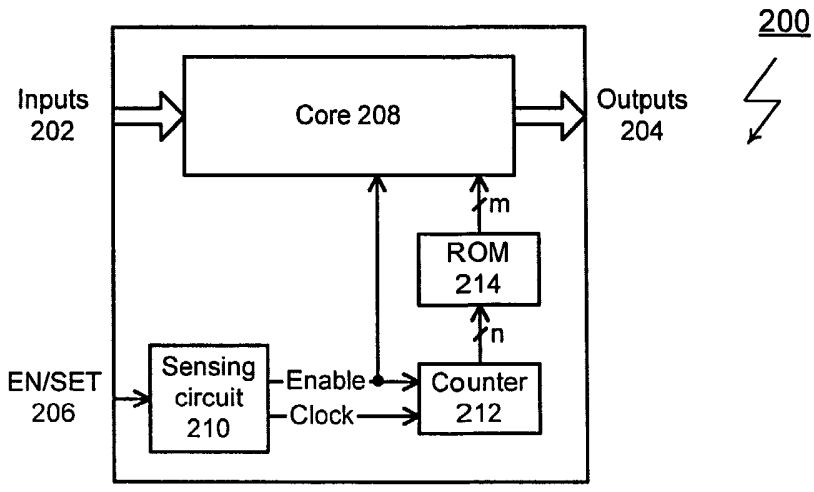
(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NI, NO, NZ, OM, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, UZ, VC, VN, YU, ZA, ZM, ZW.

(84) Designated States *(regional)*: European patent (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IT, LU, MC, NL, PT, RO, SE, SI, SK, TR).

Declarations under Rule 4.17:
— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii)) for all designations*
— *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii)) for all designations*

[Continued on next page]

(54) Title: SINGLE WIRE SERIAL INTERFACE



(57) Abstract: A single wire serial interface for power ICs and other devices is provided. To use the interface, a device is configured to include an EN/SET input pin. A counter within the device counts clock pulses sent to the EN/SET input pin. The output of the counter is passed to a ROM or other decoder circuit. The ROM selects an operational state for the device that corresponds to the value of the counter. In this way, control states may be selected for the device by sending corresponding clock pulses to the EN/SET pin. Holding the EN/SET pin high causes the device to maintain its operational state. Holding the EN/SET pin low for a predetermined timeout period resets the counter and causes the device to adopt a predetermined configuration (such as off) until new clock pulses are received at the EN/SET pin.

WO 03/083677 A2

— *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii)) for all designations*

Published:

— *without international search report and to be republished upon receipt of that report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

SINGLE WIRE SERIAL INTERFACE

Technical Field

The present invention relates generally to control interfaces for integrated circuits and other devices. More particularly, the present invention includes a single wire serial interface that may be used to control power ICs and other devices.

Background Art

In power IC applications, an interface generally serves to manage functions such as power level, or on and off switching. In the load switch power IC case, the IC either delivers power to a subsystem or not depending on the state of the on/off pin. In a more complex power supply controller, the regulated output voltage is set by a more complex interface such as an integrated 5-pin digital to analog interface. When many subsystems exist within the same system, an even more complex interface, such as the SMBUS interface may be implemented.

The complex power IC can easily afford a multi-pin control interface, since it is already in a large package, and has sufficient functional density. The stand-alone power management function cannot normally offer a complex control interface due to die size or package size constraints. Still there are cases where this type of control is desirable. For instance, it may be desirable to vary a current limit over different load scenarios. However, few pins are available for control of the simple load switch because most of the pins are used by the power function, and there is no board space or budget for a larger package. Some functionality can be added by means of an analog interface, but since most applications are controlled by a microprocessor, a digital interface is easiest to implement and most cost effective. A serial interface is efficient, but common simple serial

interfaces such as 3-wire or 2-wire require too many pins. Complex serial interfaces such as SMBUS are generally too complex and expensive to merit implementation for the stand-alone power management function.

For these reasons and others, there is a need for an interface that may be used to control stand-alone power and other IC types. Ideally, this interface would be able to accommodate a wide variety of control needs and be scaleable to many levels of complexity. Minimal pin use is also desirable, with the ideal being use of a single pin that may optionally be shared with another function.

Summary of the Invention

An aspect of the present invention provides a single wire serial interface that may be used to control stand-alone power ICs and other devices. For this aspect, an IC is configured to include a sensing circuit, a counter, and a ROM or similar decoder. The sensing circuit monitors the voltage present at one of the IC pins. Typically, this will be the on/off pin and is referred to as the EN/SET pin. The sensing circuit determines whether or not the voltage at the EN/SET pin is high, low, or toggling.

When the voltage at the EN/SET pin is toggling the counter is enabled. This causes the counter to count the rising edge of each clock pulse sent to the EN/SET pin. Holding the voltage at the EN/SET pin high causes the counter to stop counting and maintain its value. Holding the voltage at the EN/SET pin low for more than a preset timeout period causes the counter to reset to zero.

The ROM contains a total of 2^n words of m bits. Each m-bit word corresponds to one control state for the IC. The output of the counter is an address within the ROM selecting a particular m-bit word and control state. For simple

functions, the counter can be only a few bits, in which case the counter outputs can be directly decoded in logic without the complexity of a ROM.

Another aspect of the present invention is an LED current source IC incorporating the single wire serial interface. The LED current source includes at least one current output and one EN/SET input. For a representative implementation, the ROM includes a total of thirty-two (32) words. Each word corresponds to an output level for the one or more current outputs. The output levels are preferably configured as a logarithmic scale, yielding two decades of output levels and LED luminosity.

Another aspect of the present invention is a load switch IC incorporating the single wire serial interface. The load switch includes one EN/SET input and n outputs where n is greater than one. For the case of the load switch, the bits in the counter may be used to directly control the state of the individual outputs (i.e., each bit determines the state of a corresponding output). This allows the ROM to be omitted from the load switch IC, simplifying its design. The bits in the counter yield a total of 2^n different output configurations (i.e., all possible configurations).

Another aspect of the present invention is a current limited load switch IC incorporating the single wire serial interface. The current limited load switch includes one or more outputs and one EN/SET input. Each word in the ROM corresponds to a different current limit for the outputs.

Other aspects and advantages of the present invention will become apparent from the following descriptions and accompanying drawings.

Brief Description of the Drawings

For a more complete understanding of the present invention and for further features and advantages, reference is now made to the following description taken in conjunction with the accompanying drawings, in which:

Figure 1 is a timing diagram illustrating the use of the single wire protocol according to one aspect of the present invention.

Figure 2 is a block diagram showing an IC using a single wire serial interface according to one aspect of the present invention.

Figure 3 is a timing diagram illustrating the use of the single wire serial interface of the IC of Figure 2.

Figure 4 is a diagram showing a sensing circuit appropriate for use in the IC of Figure 2.

Figure 5 is a block diagram showing an IC using a latched implementation of single wire serial interface according to one aspect of the present invention.

Figure 6 is a timing diagram illustrating the use of the single wire serial interface of the IC of Figure 5.

Figure 7 is a diagram showing a latch driver circuit appropriate for use in the IC of Figure 2.

Detailed Description of the Preferred Embodiments

The preferred embodiments of the present invention and their advantages are best understood by referring to Figures 1 through 7 of the drawings. Like numerals are used for like and corresponding parts of the various drawings.

Single Wire Serial Protocol

An aspect of the present invention provides a single wire serial protocol that may be used to control ICs and other compatible devices. To use the single wire serial protocol, a device must support a series of different operational states or modes. For one example, a stand-alone power IC might be configured to support a range of different output levels. Typically, these output levels would progress in even increments from a no-power or off condition to a full power condition. Each different output level would define a particular operational state. The single wire serial protocol allows the operational states of compatible devices to be dynamically controlled. Thus, for the stand-alone power IC example, the single wire serial protocol would be used to select different operational states and associated output power levels.

Devices that support the single wire serial protocol are configured to receive an EN/SET signal. As shown by the timing diagram of Figure 1, the EN/SET signal may be characterized as having three different waveforms. The first of these is a toggling waveform where the EN/SET signal is composed of a series of clock pulses. The second waveform is where the EN/SET signal is asserted to have a constant high value. The third waveform is where the EN/SET signal is asserted to have a constant low value.

The toggling waveform causes compatible devices to select particular operational states. The total number of

clock pulses (or rising edges) determines the particular operational state that will be selected (i.e., four clock pulses selects the fourth operational state and so on). Additional clock pulses that exceed the number of operational states supported by a compatible device will generally cause the count to rollover and start again with the first operational state.

The constant high waveform causes compatible devices to maintain their previously selected operational states. As shown in Figure 1, the current operational state may be continued for an arbitrary duration in this way.

The constant low waveform causes compatible devices to power off (or otherwise adopt a predefined configuration) after a pre-defined timeout period has elapsed. The timeout period allows compatible devices to distinguish between the constant low waveform and the shorter low portions of the toggling waveform. For a typical implementation, the timeout value is 400µs with the EN/SET signal having a frequency in the range of 1Mhz to 10kHz. Higher and lower frequencies are also possible.

Single Wire Serial Interface

To use the single wire serial protocol, compatible devices must provide a single wire serial interface. For the purposes of illustration, Figure 2 shows a block diagram of an IC (generally designated 200) configured to provide this interface. IC 200 includes one or more inputs 202 and one or more outputs 204. IC 200 also includes an EN/SET input 206 and a core portion 208. Core portion 208 is intended to be generally representative of the circuits that function to create outputs 204 using inputs 202EN/SET input 206 is connected to a sensing circuit 210. Sensing circuit 210 monitors the EN/SET signal at EN/SET input 206 and determines if that voltage is constantly high, constantly low, or

toggling. Based on this determination, sensing circuit 210 produces two signals: a Clock signal and an Enable signal. The Clock and Enable signals control the operation of a counter 212 having n bits. Counter 212 counts the rising transitions of the Clock signal whenever sensing circuit 210 asserts the Enable signal. Counter 212 resets whenever the Enable signal is not asserted.

The relationship between the EN/SET signal and the Clock and Enable signals is shown in more detail in the timing diagram of Figure 3. As shown in that figure, a rising transition of the EN/SET signal causes sensing circuit 210 to assert the Enable signal. Sensing circuit 210 holds the Enable signal high until the EN/SET signal transitions to a logical low state and remains in the low state until the predetermined timeout period has elapsed. The Enable signal acts to gate the Clock signal. As long as the Enable signal remains high, sensing circuit 210 forwards the EN/SET signal as the Clock signal. Counter 212 receives both the Clock and Enable signal. The first rising transition of the EN/SET signal raises the Enable signal and causes the EN/SET signal to be forwarded as the Clock signal. Counter 212 responds by increasing its value to one. Subsequent rising transitions causes Counter 212 to increment its value to two, three and so on. Counter 212 resets to zero when sensing circuit 210 transitions the Enable signal to a low value.

The n output bits of counter 212 control a ROM 214. ROM 214 has a total of 2^n words, each having m bits. Each m-bit word corresponds to one control state for IC 200. The n-bit output of counter 212 selects a particular m-bit word within ROM 214. The selected control state and Enable signal are passed to core portion 208. Core portion 208 is configured to adjust its operation to match the selected control state.

Sensing Circuit

Figure 4 shows a representative implementation for sensing circuit 210. As shown in that figure, sensing circuit 210 produces the Enable and Clock signals by timing the logic low period of the EN/SET signal. As long as the timeout period is not exceeded, the Enable signal will remain high, and the EN/SET signal will feed through logic gate AND1 to become the Clock signal. In the described implementation, the timer consists of capacitor C1 and current source I1. Transistors MN2 and MN3 mirror current source I1. This linearly discharges capacitor C1 when the EN/SET signal is a logical low, and transistor MP1 is off. If the EN/SET signal remains in a logic low state long enough, capacitor C1 will discharge to a voltage that is less than the threshold of transistor MN1 and turn MN1 off. When MN1 is off, R1 pulls node "2" to the threshold of Schmit trigger ST1 and the Enable signal goes to a logic low state. As long as the EN/SET signal remains low for a period less than the timeout period, the Enable signal will remain in a logic high state. The timeout period is dominated by the power supply voltage, the threshold of transistor MN1 (V_{tMN1}), the value of capacitor C1, and the magnitude of current source I1, given by:

$$\text{Timeout} = C * (Vcc - V_{tMN1})/I1$$

Typical values of C1 = 10pF, Vcc = 5v, V_{tMN1} = 1v and I1 = 0.1μA yield a timeout period of 400μs. Sensing circuit 210 can respond to a 400ns signal of the EN/SET signal. As a result, it is able to differentiate between the EN/SET signal as Clock and EN/SET signal as Enable. A typical application can be designed around a range of EN/SET frequencies between 1Mhz to 10kHz, or slower if desired.

Latched Single Wire Serial Interface

Devices that implement the just described single wire serial interface select a new control state each time a rising edge of a clock pulse is received. One result is that compatible devices progressively select each control state in sequence until the desired control state is reached. So, selecting the eighth control state means that compatible devices will progressively select control states one through seven before finally selecting the eighth (desired) control state. For some devices this behavior is acceptable or even desirable. This can be true, for example where the device is a current source where progressively increasing output can be benign or even useful. In other cases, selection of intermediate control states may have unwanted side effects. This could be true for the case of the multiple load switch that is described below.

Figure 5 shows a block diagram of an IC (generally designated 500) that uses an implementation of the single wire serial interface that eliminates intermediate control states. IC 500 includes the majority of components previously described for Figure 2 and IC 200. In this case, the output of counter 212 is passed through a latch 502 before reaching ROM 214. Latch 502 is controlled by a Latch signal generated by a latch driver circuit 504.

The relationship between the EN/SET, Clock, Enable and Latch signals is shown in Figure 6. As shown, the Latch signal remains low until the EN/SET signal has been maintained in a high state for a duration that exceeds a predetermined latch timeout period. Holding the EN/SET signal high for longer than the latch timeout period causes latch driver 504 to assert the Latch signal. This, in turn causes latch 502 to forward the accumulated value of counter 212 to ROM 214. The result is that counter 212 is prevented from

forwarding intermediate control states until the EN/SET signal has been asserted high after the train of clock pulses has been completed.

Latch Driver Circuit

Figure 7 shows a representative implementation for latch driver 504. As shown in that figure, latch driver 504 produces the Latch signal by timing the logic high period of the EN/SET signal. As long as the EN/SET signal is high for less than the latch timeout period, the Latch signal remains low. In the described implementation, the timer consists of capacitor C1 and current source I1. Transistors MN2 and MN3 mirror current source I1. This linearly discharges capacitor C1 when the EN/SET signal is a logical high, and transistor MP1 is off. If the EN/SET signal remains in a logic high state long enough, capacitor C1 will discharge to a voltage that is less than the threshold of transistor MN1 and turn MN1 off. When MN1 is off, R1 pulls node "2" to the threshold of Schmit trigger ST1 and the Latch signal goes to a logic high state. As long as the EN/SET signal remains high for a period less than the latch timeout period, the Latch signal will remain in a logic low state. The latch timeout period is dominated by the power supply voltage, the threshold of transistor MN1 (V_{tMN1}), the value of capacitor C1, and the magnitude of current source I1, given by:

$$\text{Latch Timeout} = C * (Vcc - V_{tMN1})/I1$$

Typical values of C1 = 10pF, Vcc = 5v, V_{tMN1} = 1v and I1 = 0.1μA yield a latch timeout period of 400μs. Latch driver 504 can respond to a 400ns signal of the EN/SET signal. As a result, it is able to differentiate between the EN/SET signal as Clock and EN/SET signal as Latch. A typical application can be designed around a range of EN/SET frequencies between 1Mhz to 10kHz, or slower if desired.

Decoder

ROM 214 provides a mapping between the EN/SET signal and associated control states for IC 200. In some cases, there may be relatively few control states. In other cases, the mapping may be defined functionally. In these cases, it is possible to replace ROM 214 with a decoder. This allows the outputs of counter 212 to be directly decoded in logic without the complexity of a ROM.

LED Driver

The white LED has become the backlight source of choice for small displays used in products such as cell phones that typically use a lithium ion battery for power. The white LED is an excellent light source. However, it requires from 3.6 to 4.1 volts of forward bias voltage to conduct current and emit light. Since the lithium ion battery runs between 4.1 and 2.9 volts, a regulated boosted voltage must be generated to power the LED. Four LED's are typically used in a display; either in a serial or a parallel arrangement.

The lowest cost solution is to drive the four LED's in parallel with a charge pump. The higher cost solution is to drive the four LED's in series with a DC/DC boost converter capable of boosting the lithium ion battery up to four times the forward voltage of the LEDs (e.g. 4 x 4.1 = 16.4volts). The DC/DC boost converter is higher cost due to the cost and size of the required inductor, but since the LED is really a current mode device, the performance is better because all of the LED's in series will be biased with the same current and share the same luminosity.

The charge pump solution is attractive because small low cost capacitors can be used to develop a voltage of up to 1.5 or 2 times the battery voltage. The disadvantage to the charge pump solution is that the resulting voltage must be

sensed as a current for brightness control of the LED. A single voltage can drive multiple LED's, however only one LED is used as the current reference. This is achieved by adding a current setting and sensing resistor in series with it. The additional LED's have a matching resistor in series, but unless their forward voltages match that of the reference LED, they will have substantially different currents and, as such, brightness levels. A better solution would have parallel current outputs for driving the LED with a current. In this manner, all LED's would have the same bias current and luminosity. The parallel outputs however, require more pins and a larger package that is a significant disadvantage.

Another issue is brightness control. Brightness control can be performed by setting a reference current and leaving it constant, or by applying some control means to the DC/DC converter to obtain a different output voltage or current. One way to control the brightness of an LED is to simply turn it on and off at a higher frequency than the human eye can detect, and pulse width modulate (PWM) the on-time. An easier system solution would be an interface whereby a current control is input to the DC/DC converter to control the output current. This can be accomplished either by a control voltage or a digital interface. A simple solution is a digital interface, but to have enough resolution, or a large enough range, many bits of control are required. This leads again to higher undesirable pin count.

Since the human eye senses brightness logarithmically, a useful digital control would result in a logarithmic brightness scale. A logarithmic scale that adequately covers two decades of luminosity requires at least 5-bits or 32 levels.

An aspect of the present invention provides an LED driver that effectively meets all of these requirements. The

LED driver is preferably configured as a 12-pin device with four LED current source outputs. The LED driver also includes an EN/SET input that supports the single wire serial protocol described above. The EN/SET input functions as the on/off control as well as the brightness control. Internally, the LED driver includes a five-bit counter and a thirty-two word ROM. The control states included in the ROM are configured to provide logarithmically increasing levels of luminosity. The counter and ROM are scaleable to any number of levels beyond or below 32.

Multiple Load Switch

Another aspect of the present invention is a load switch IC incorporating the interface described in the preceding paragraphs. For an eight-pin package, the load switch includes one EN/SET input, five outputs, a power input and a ground input. For the case of the load switch, the bits in the counter may be used to directly control the state of the individual outputs (i.e., each bit determines the state of a corresponding output). This allows the ROM to be omitted from the load switch IC, simplifying its design. The bits in the counter yield a total of 2^5 or thirty-two different output configurations (i.e., all possible configurations). If the load switches are very slow to respond, the single wire serial interface can be operated at a much higher frequency than the switches can respond and the outputs will be well behaved. In the case where the switches are fast, an addition must be made whereby the value clocked into the single wire serial interface is not latched until the clocking has stopped.

Current Limited Load Switch with Configurable Current Limit

Another aspect of the present invention is a current limited load switch IC incorporating the interface described in the preceding paragraphs. The current limited load switch

includes one or more outputs and one EN/SET input. Each word in the ROM corresponds to a different current limit for the one or more outputs. The current limited load switch is disabled a predetermined period after the EN/SET transitions to the low state.

Although particular embodiments of the present invention have been shown and described, it will be apparent to those skilled in the art that changes and modifications may be made without departing from the present invention in its broader aspects, and therefore, the appended claims are to encompass within their scope all such changes and modifications that fall within the true scope of the present invention.

WHAT IS CLAIMED IS:

1. An interface for controlling a device, the interface comprising:

a first circuit for accumulating a count of clock pulses encoded in a received signal;

a second circuit for mapping the count of clock pulses into a corresponding control state for the device; and

a third circuit for resetting the count of clock pulses to zero if the received signal is low for a period that exceeds a predetermined timeout value.

2. An interface as recited in claim 1 that further comprises a fourth circuit for maintaining a prior control state until the count of pulses has been accumulated.

3. An interface as recited in claim 2 that further comprises a fifth circuit for discarding the prior control state if the received signal is high for a period that exceeds a predetermined latch timeout value.

4. An interface as recited in claim 1 wherein the second circuit further comprises a read only memory (ROM), the ROM including one word for each control state of the device, each word addressable by a corresponding count of clock pulses.

5. An interface as recited in claim 1 wherein the second circuit further comprises an array of logic elements, the array of logic elements configured to directly map each count of clock pulses to a corresponding control state for the device.

6. An interface as recited in claim 1 wherein the first circuit further comprises a counter, the counter configured to increment with each rising edge of each clock pulse of the received signal.

7. An interface as recited in claim 1 wherein the third circuit is configured to enable the first circuit at the rising edge of the first clock pulse of the received signal, the third circuit maintaining the first circuit in an enabled state until the received signal is low for a period that exceeds the predetermined timeout value.

8. A method for controlling a device, the method comprising:

accumulating a count of clock pulses encoded in a received signal;

mapping the count of clock pulses into a corresponding control state for the device; and

resetting the count of clock pulses to zero if the received signal is low for a period that exceeds a predetermined timeout value.

9. A method as recited in claim 8 that further comprises the step of accessing a read only memory (ROM) at an address equal to the count of clock pulses to retrieve the corresponding control state.

10. A method as recited in claim 8 that further comprises the step of decoding the count of clock pulses to generate the corresponding control state.

11. A method as recited in claim 8 that further comprises the step of incrementing a counter with each rising edge of each clock pulse of the received signal.

12. A method as recited in claim 8 that further comprises the step of maintaining a previous control state until the step of accumulating a count of clock pulses has completed.

13. A method for controlling a device, the method comprising:

toggling an input signal to the device to encode a series of clock pulses with the number of clock pulses corresponding to a desired control state for the device;

maintaining the input signal in a logical high state for a period equal to the duration in which the desired control state is to remain active shortened by a predefined timeout period; and

maintaining the input signal in a logical low state for a duration that exceeds the predefined timout period to terminate the desired state.

14. A device that comprises:

a core portion configured to support a series of control states, each control state defining a set of operational parameters for the device;

a counter for accumulating a count of pulses encoded in a received signal;

a circuit for mapping the count of clock pulses into a corresponding control state; and

a sensing circuit for resetting the count of clock pulses to zero if the received signal is low for a period that exceeds a predetermined timeout value.

15. A device as recited in claim 14 that further comprises a latch for maintaining a prior control state until the count of pulses has been accumulated.

16. A device as recited in claim 15 that further comprises a latch driver for causing the latch to discard the prior control state if the received signal is high for a period that exceeds a predetermined latch timeout value.

17. A device as recited in claim 14 wherein the circuit for mapping further comprises a read only memory (ROM), the ROM including one word for each control state of the device, each word addressable by a corresponding count of clock pulses.

18. A device as recited in claim 14 wherein the circuit for mapping further comprises an array of logic elements, the array of logic elements configured to directly map each count of clock pulses to a corresponding control state for the device.

19. A device as recited in claim 14 that further comprises one or more LED current source outputs and where each control state corresponds to a different output level for the LED current source outputs.

20. A device as recited in claim 14 that further comprises a series of load switch outputs and where each control state corresponds to a different combination of off and on states for the load switch outputs.

21. A device as recited in claim 14 that further comprises a series of current limited load switch outputs and where each control state corresponds to a current limit for the limited load switch outputs.

Fig. 1

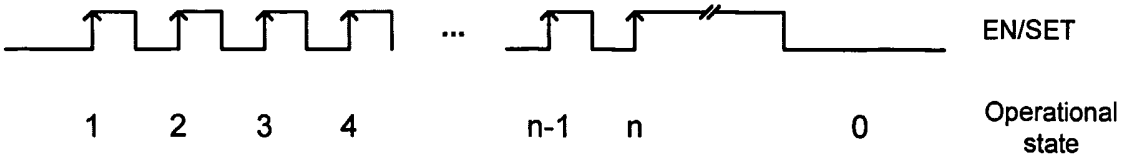

EN/SET
1
2
3
4
n-1
n
0
Operational state

Fig. 2

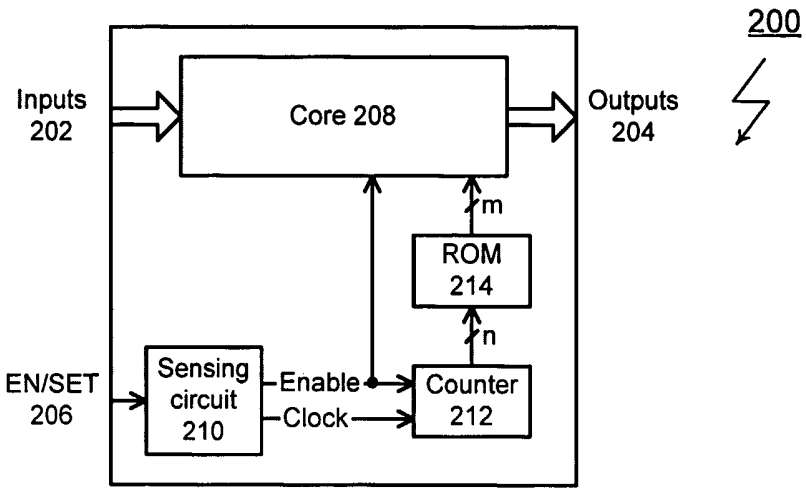

200
Inputs 202
Core 208
Outputs 204
m
ROM 214
n
EN/SET 206
Sensing circuit 210
Enable
Clock
Counter 212

Fig. 3

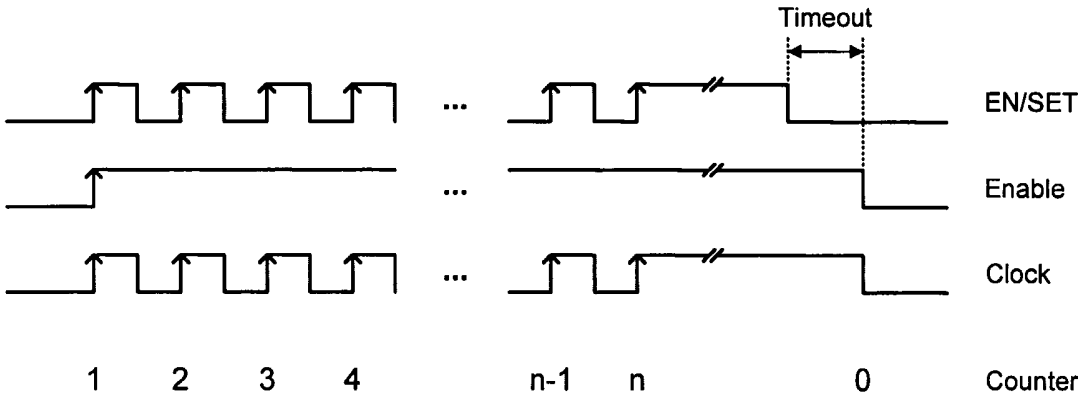

Timeout
EN/SET
Enable
Clock
1 2 3 4 n-1 n 0
Counter

Fig. 4

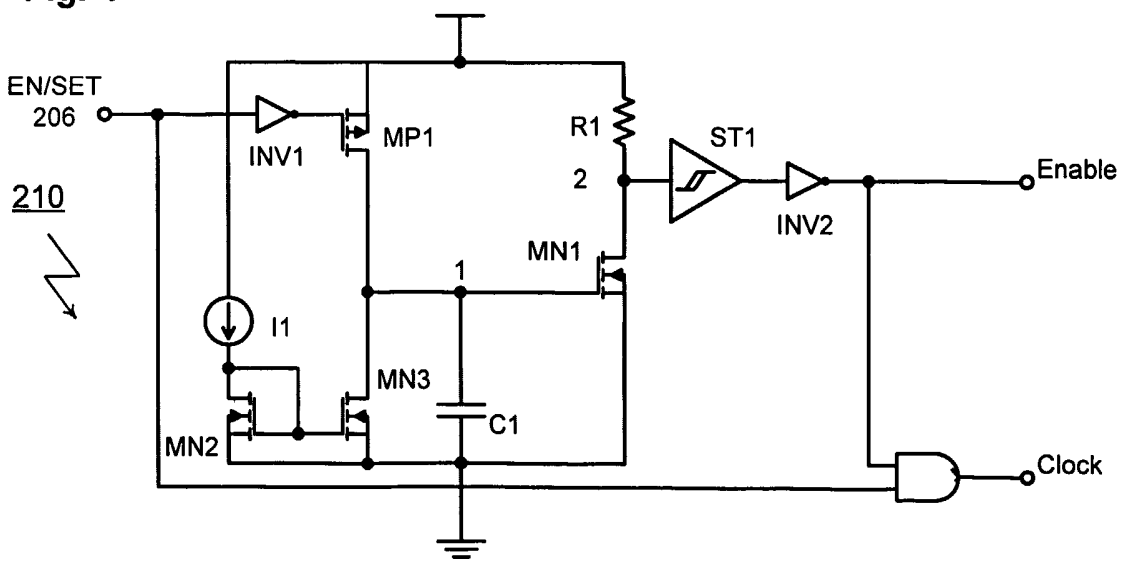

EN/SET
206
210
INV1
MP1
R1
2
ST1
INV2
Enable
MN1
1
I1
MN3
C1
MN2
Clock

Fig. 5

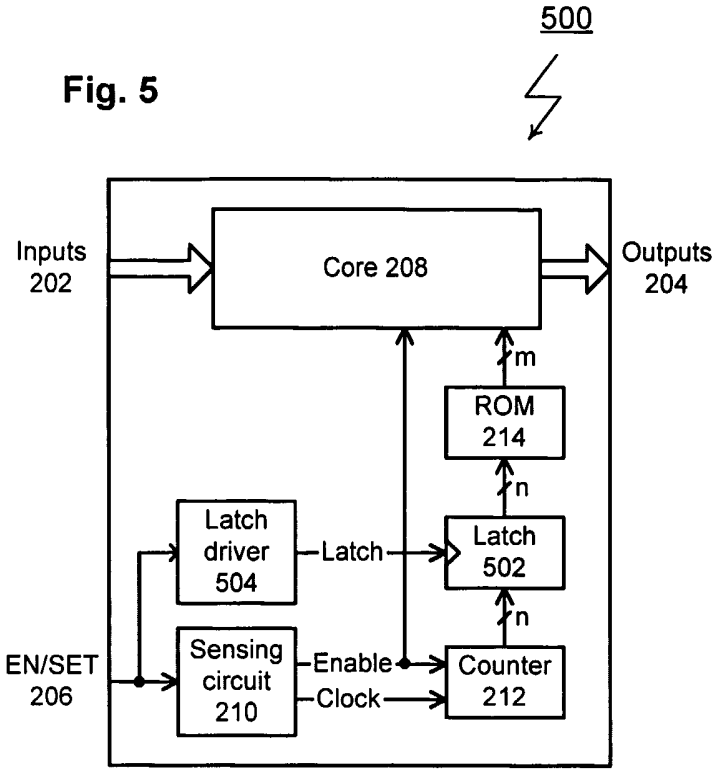

500
Inputs
202
Core 208
Outputs
204
m
ROM
214
n
Latch
driver
504
Latch
Latch
502
n
EN/SET
206
Sensing
circuit
210
Enable
Clock
Counter
212

Fig. 6

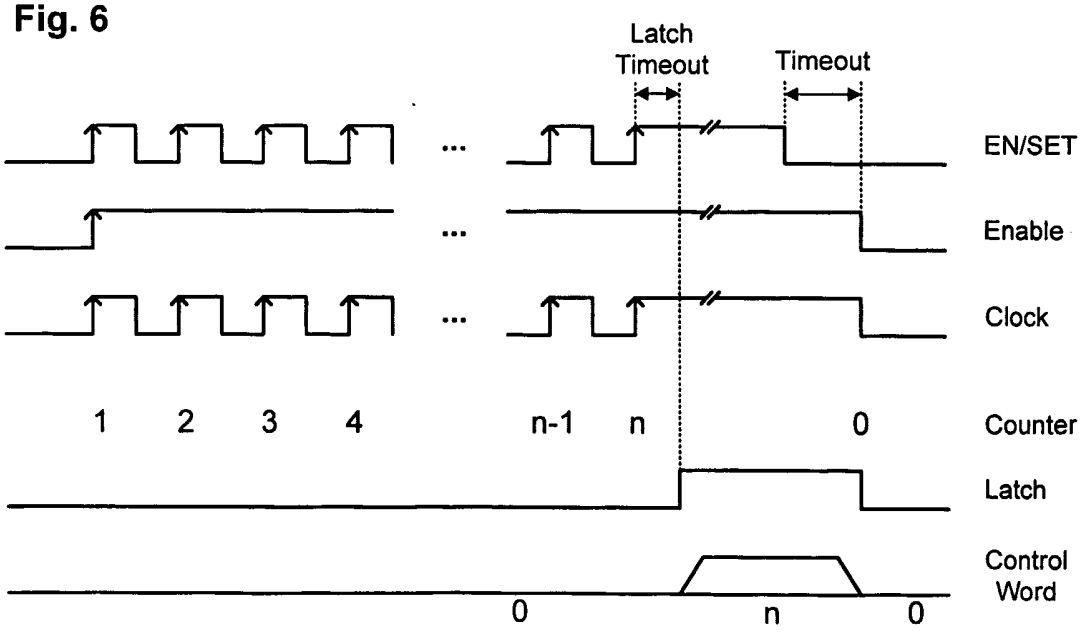

Latch
Timeout
Timeout
EN/SET
Enable
Clock
1
2
3
4
n-1
n
0
Counter
Latch
Control
Word
0
n
0

Fig. 7