METHOD AND APPARATUS FOR REDUCING SIMULTANEOUS SWITCHING OUTPUT NOISE USING DYNAMIC BUS INVERSION

Field Of The Invention

The present invention pertains to the field of computer systems. More particularly, this invention pertains to the field of reducing simultaneous switching output noise on a data bus.

Background of the Invention

In an effort to increase performance in today's computer systems, system designers seek to increase clock frequencies on various system data busses. As bus frequencies increase, noise on the data lines becomes an increasingly important issue. An increase in clock frequency, and its corresponding reduction in clock period, allows for less time for noise present on data lines to settle before data is latched at the receiving end. Noise present on data lines at the time data is latched at the receiving end may result in the latching of invalid data.

One source of noise on data lines may be referred to as simultaneous switching output noise. This noise results from several to many output drivers on a particular device switching states at the same time. One example of this may include a system logic device driving 32 bits of data at once to a graphics device over a graphics bus. The system logic device may drive 32 bits of "1"s during one clock period and then drive 32 bits of "0"s during the next clock period. In this example all of the graphics bus bits change state from one clock period to the next. Such transitions may induce simultaneous switching output noise on some or all of the 32 data lines and limit the possible clock frequency for the graphics bus, thereby limiting the potential performance of the graphics subsystem.

Brief Description of the Drawings

The invention will be understood more fully from the detailed description given below and from the accompanying drawings of embodiments of the invention which, however, should not be taken to limit the invention to the specific embodiments described, but are for explanation and understanding only.

Figure 1 is a block diagram of an embodiment of a system including a system logic device having a dynamic bus inversion encoder and a graphics device having a dynamic bus inversion decoder.

Figure 2 is a block diagram of an embodiment of a dynamic bus inversion encoder.

Figure 3 is a flow diagram of an embodiment of a method for reducing simultaneous switching output noise using dynamic bus inversion.

Detailed Description

Figure 1 is a block diagram of an embodiment of a system 100 including a system logic device 120 having a dynamic bus inversion (DBI) encoder 200 and a graphics device 130 having a DBI decoder 124. The system 100 also includes a processor 110 coupled to the system logic device 120. The system logic device is further coupled to a system memory 140 and an input/output hub 150.

The system logic device 120 is coupled to the graphics device 130 by way of an output data bus 203 and an inversion signal 205. The DBI encoder 200 compares a set of data bits previously placed on the output data bus 203 with a next set of data bits to be placed on the data bus 203. The DBI encoder 200 determines how many data bit transitions would occur on the data bus 203 as a result of the next set of data bits being driven on the data bus 203 following the previous set of data bits. If the number of potential data bit transitions is greater than a predetermined number, then the DBI encoder 200 inverts each data bit in the next set of data bits and drives the inverted data onto the output data bus 203. The DBI encoder 200 asserts the inversion signal 205 to indicate to

the DBI decoder 124 that the data bits have been inverted. The DBI decoder 124 then inverts the inverted data bits in order to restore the original data.

The predetermined number is selected to produce the fewest bit transitions. For some embodiments, the predetermined number is selected to equal one half of the data width. For example, for a group of 16 bits of data, the predetermined number would be 8. Therefore, if more than 8 data bits would change state, the data bits are inverted and the inversion signal is asserted.

The DBI embodiment described above reduces the amount of simultaneous switching output noise by limiting the number of bit transitions occurring on the output data bus.

An embodiment of the decoder 124 includes an XOR circuit that performs an XOR function on the incoming data and the inversion signal. In this manner, if no inversion is indicated (the inversion signal 205 is a "0") then the incoming data is not inverted by the decoder 124. If an inversion is indicated (the inversion signal 205 is a "1") then each bit of incoming data is inverted by the decoder 124 in order to restore the original data.

Although system 100 shows a single data bus and a single inversion signal, the data bus 203 may be divided into two or more groups. For example, a 32 bit bus may be divided into two groups of 16 bits each. Other configurations are possible. Inversion decisions are then made by the encoder 200 on a group-by-group basis. Each group utilizes its own inversion signal.

Figure 2 is a block diagram of an embodiment of the dynamic bus inversion encoder 200. Data to be delivered over the output data bus 203 is delivered to the encoder 200 via an internal data bus 201. For this embodiment, the internal data bus 201 and the output data bus 203 are 16 bits wide. Other embodiments are possible with a data width of 32 bits organized into two groups of 16 bits each. Still other embodiments are possible with other data organizations and groupings.

The internal data 201 is delivered to both an XOR circuit 210 and an inverter 220/multiplexor 230 combination. The XOR circuit 210 also receives output data 203.

The output data 203 represents a current set of data bits. An XOR operation is performed on the input data 201 and the output data 203. An inversion determination circuit 240 receives the output of the XOR operation and determines whether the number of data bit transitions found by the XOR operation exceeds a predetermined number. For this example embodiment, the predetermined number is eight. Other embodiments are possible using other predetermined numbers. It is also possible to implement the encoder 200 using a programmable predetermined number. If the inversion determination circuit 240 determines that the number of data bit transitions exceeds the predetermined number, an internal inversion signal 209 is asserted.

The internal inversion signal 209 is delivered to the multiplexor 230 and a latch 260. If the internal inversion signal 209 is asserted, indicating that the predetermined number of data bit transitions is exceeded, then the multiplexor 230 delivers the output of the inverter circuit 220 to a latch 250. If the internal inversion signal is not asserted, then the multiplexor 230 delivers the non-inverted internal data to the latch 250. The latch 250 then latches the output of the multiplexor 230 onto the output data bus 203. The internal inversion signal 209 is latched onto the inversion signal 205.

The inverter 220/multiplexor 230 combination may be implemented as an XOR circuit where the internal data bus bits 201 are each XORed with the internal inversion signal 209. When the internal inversion signal 209 is asserted, then each of the internal data bits 201 are inverted and delivered to the input of the latch 250. If the internal inversion signal 209 is not asserted, then the XOR operation will leave the internal data bits 201 unchanged.

Figure 3 is a flow diagram of an embodiment of a method for reducing simultaneous switching output noise using dynamic bus inversion. At block 310, a first n bits of data are delivered over a data bus. The number of bit transitions between the first n bits of data and a second n bits of data are counted at block 320. As indicated at block 330, if the counted bit transitions exceeds a predetermined number, then the next n bits of data are inverted at block 340. The inverted next n bits of data are delivered over the data

bus at block 360. An inversion signal is also asserted at block 360. If the counted bit transitions do not exceed the predetermined number, then the next n bits of data are delivered over the data bus at block 350. The preceding embodiment of a method may be repeated for every subsequent n bits of data to be delivered over the data bus. For this embodiment, n equals 16 and the predetermined number is eight, although other embodiments are possible using other data widths and predetermined numbers.

Although some of the previously discussed embodiments mention a system logic device transmitting data to a graphics device over a graphics bus, other embodiments are possible using any system device as a transmitter having a DBI encoder and using any other device as a receiver having a DBI decoder.

In the foregoing specification the invention has been described with reference to specific exemplary embodiments thereof. It will, however, be evident that various modifications and changes may be made thereto without departing from the broader spirit and scope of the invention as set forth in the appended claims. The specification and drawings are, accordingly, to be regarded in an illustrative rather than in a restrictive sense.

Reference in the specification to "an embodiment," "one embodiment," "some embodiments," or "other embodiments" means that a particular feature, structure, or characteristic described in connection with the embodiments is included in at least some embodiments, but not necessarily all embodiments, of the invention. The various appearances of "an embodiment," "one embodiment," or "some embodiments" are not necessarily all referring to the same embodiments.

CLAIMS

What is claimed is:

1. An apparatus, comprising:

a bit transition detection circuit to determine whether a next bus transaction would result in greater than a predetermined number of bit transitions when compared with a current bus transaction, the bit transition detection circuit to assert an inversion signal if the next bus transaction would result in greater than the predetermined number of bit transitions when compared with the current bus transaction; and

an inversion circuit to invert the bits of the next bus transaction in response to an assertion of the inversion signal.

2. The apparatus of claim 1, wherein the bit detection circuit includes a first XOR circuit to sense a number of bit transitions between the current bus transaction and the next bus transaction.

3. The apparatus of claim 2, wherein the bit detection circuitry includes a circuit to determine whether the number of bit transitions sensed by the first XOR circuit exceeds the predetermined number.

4. The apparatus of claim 3, wherein the inversion circuit includes a second XOR circuit to perform an XOR function between the next bus transaction and the inversion signal.

5. The apparatus of claim 4, further comprising a first latch to latch the output of the second XOR circuit, the first latch to provide the latched output to an external data bus and further to provide the latched output to the first XOR circuit.

6. The apparatus of claim 5, further comprising a second latch to latch the inversion signal and to provide the latched inversion signal to an external inversion signal.

7. The apparatus of claim 6, wherein the current and next bus transactions are sixteen bits wide.

8. The apparatus of claim 7, wherein the predetermined number of bit transitions is eight.

9. A method, comprising:

determining whether a next bus transaction would result in greater than a predetermined number of bit transactions when compared with a current bus transaction; and

providing an inverted version of the next bus transaction.

10. The method of claim 9, further comprising providing an inversion signal to indicate that the next bus transaction is inverted.

11. A method, comprising:

delivering a first n bits of data over a data bus;

counting bit transitions between the first n bits of data and a second n bits of data;

determining whether the counted bit transitions exceed a predetermined number;

inverting the next n bits of data and asserting an inversion signal if the counted bit transitions exceed the predetermined number; and

delivering the next n bits over the data bus.

12. The method of claim 11, wherein inverting the next n bits includes performing an XOR operation between the next n bits and the inversion signal.

13. The method of claim 12, wherein n is sixteen.

14. The method of claim 13, wherein the predetermined number is eight.

15. A system, comprising:

a processor;

a first logic device coupled to the processor, the system logic device including a dynamic bus inversion encoder, the dynamic bus inversion encoder including

a bit transition detection circuit to determine whether a next bus transaction would result in greater than a predetermined number of bit transitions when compared with a current bus transaction, the bit transition detection circuit to assert an inversion signal if the

next bus transaction would result in greater than the predetermined number of bit transitions when compared with the current bus transaction, and

an inversion circuit to invert the bits of the next bus transaction in response to an assertion of the inversion signal; and

a second logic device coupled to the first logic device via a bus.

16. The system of claim 15, wherein the bit detection circuit includes a first XOR circuit to sense a number of bit transitions between the current bus transaction and the next bus transaction.

17. The system of claim 16, wherein the bit detection circuitry includes a circuit to determine whether the number of bit transitions sensed by the first XOR circuit exceeds the predetermined number.

18. The system of claim 17, wherein the inversion circuit includes a second XOR circuit to perform an XOR function between the next bus transaction and the inversion signal.

19. The system of claim 17, further comprising a first latch to latch the output of the second XOR circuit, the first latch to provide the latched output to the bus and further to provide the latched output to the first XOR circuit.

20. The system of claim 19, further comprising a second latch to latch the inversion signal and to provide the latched inversion signal to an external inversion signal, the external inversion signal coupled to the second logic device.

21. The system of claim 20, wherein the second logic device includes a dynamic bus inversion decoder.

21. An apparatus, comprising:

a data bus input to receive n bits of data;

an inversion signal input to receive an inversion signal; and

an inversion circuit to invert the n bits of data received at the data bus input if the inversion signal is asserted.

22. The apparatus of claim 21, wherein the inversion circuit includes an XOR circuit to perform an XOR function between the n bits of data received at the data bus input and the inversion signal.

23. The apparatus of claim 22 wherein n is 16.

24. A method, comprising:

receiving n bits of data at a receiving device; and

inverting the n bits of data in response to an assertion of an inversion signal, the inversion signal asserted by a transmitting device.

25. The method of claim 24, wherein inverting the n bits of data includes performing an XOR function between the n bits of data and the inversion signal.