RATE VERIFICATION OF AN INCOMING SERIAL ALIGNMENT SEQUENCE

Field of the Invention

[0001] The present invention pertains to the field of semiconductor devices. More particularly, this invention pertains to the field of serial transmission alignment.

Background of the Invention

[0002] In today's computer systems, some computer system component interconnect protocols provide techniques to establish and to maintain synchronization. One such protocol is the Serial ATA protocol (Serial ATA Specification rev. 1.0 released June 28, 2001). This protocol allows communication between two devices such as a disk controller and a disk drive. The Serial ATA specification provides for a serial interconnect using differential pair signaling. The Serial ATA specification further provides for periodic transmission of alignment primitives. The alignment primitive is a predetermined pattern of bits of a predetermined length that is recognized by devices coupled to the interconnect. The alignment primitive allows devices that have lost synchronization to recover bit-boundary alignment.

[0003] The alignment method provided for in the Serial ATA specification involves the transfer of ALIGN primitives across the interface. The ALIGN primitive is a four byte sequence. The first byte of the primitive is an encoded K28.5

character (rd+: 110000 0101; rd-: 001111 1010). The receiving device compares the first character of the incoming ALIGN primitive with the expected K28.5 character. If there is a match, then the receiving device is assumed to be synchronized.

[0004] The above mentioned method is susceptible to false detection of ALIGN primitives that are being sent at a different rate than expected, or false interpretation of noise on the interconnect wires as a valid ALIGN primitive.

Brief Description of the Drawings

[0005] 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.

[0006] Figure 1 is a block diagram of one embodiment of a computer system including a serial interface controller coupled to a disk drive.

[0007] Figure 2 is a block diagram of one embodiment of an apparatus for rate verification of an incoming serial alignment sequence including a shift register, a checking logic unit, and a state machine.

[0008] Figure 3a is a block diagram of a shift register.

[0009] Figure 3b is a block diagram of a checking logic unit.

[0010] Figure 4 is a diagram of a state machine.

[0011] Figure 5 is a flow diagram of one embodiment of a method for rate verification of an incoming serial alignment sequence.

Detailed Description

[0012] Figure 1 is a block diagram of one embodiment of a computer system 100 including a serial interface controller 200 coupled to a disk drive 150. The serial interface controller 200 is included in an input/output hub 140 which is also coupled to a peripheral bus 145.

[0013] The computer system 100 also includes a system logic device 120 that is coupled to the input/output hub 140 via a hub interconnect 125. The system logic device 120 is coupled to a processor 110 and is also coupled to a system memory 130.

[0014] The serial interface controller 200 is coupled to the disk drive 150 via an interconnect 147. For this example embodiment, the interconnect 147 is implemented in accordance with the Serial ATA specification. Other embodiments are possible using other interconnect implementations. For this example embodiment, the interconnect 147 includes one differential pair of signals that deliver data from the controller 200 to the disk drive 150 and another differential pair of signals that deliver data from the disk drive 150 to the controller 200.

[0015] Periodically, the serial interconnect controller 200 receives an align sequence over the interconnect 147. The align sequence includes a series of align primitives. For this embodiment, an align primitive is a four byte sequence, with the first byte including an encoded K28.5 character.

[0016] Figure 2 is a block diagram of one embodiment of an apparatus for rate verification of an incoming serial alignment sequence including a shift register, a checking logic, and a state machine. The embodiment of Figure 2 may be implemented in a serial interconnect controller such as the controller 200 shown in Figure 1. A data recovery circuit/analog front end (AFE) 210 receives an input stream over the serial ATA differential pair 147. The unit 210 asserts a non-rate verified align detect signal 215 whenever a K28.5 character is recognized coming in with the input stream. The non-rate verified align detect signal 215 is received at a shift register and checking logic unit 300 and a state machine 400. The units 300 and 400 are discussed below in connection with Figures 3a, 3b, and 4.

[0017] The units 300 and 400 together determine whether the incoming align primitives are being received at a target rate. If the rate of the incoming align primitives matches the target rate, then a rate verified align detect signal 225 is asserted and delivered to the core logic of the serial interconnect controller 200. One use of the rate verified align detect signal 225 is for speed negotiation in compliance with the Serial ATA specification for the serial interconnect 147. The serial interconnect controller 200 can use the rate verified align detect signal to know whether the incoming serial stream is being received at the target rate or some other rate.

[0018] Figure 3a is a block diagram of one embodiment of the shift register of unit 300. The non-rate verified align detect signal 215 is received at a flip-flop 310. The flip-flop 310 is clocked at a target clock rate. The output of the flip-flop 310 is denoted as Last Align Detect Signal 0 (LastAD[0]). The output of the flip-flop 310 is delivered to a flip-flop 312. The flip-flop 312 is also clocked at the target rate. The output of the flip-flop 312 is denoted as LastAD[1] and is delivered to a flip-flop 314. The flip-flop 314 is also clocked at the target rate. The output of the flip-flop 314 is denoted as LastAD[2] and is delivered to a flip-flop 316. The flip-flop 316 is also clocked at the target rate. The output of the flip-flop 316 is denoted as LastAD[3].

[0019] Figure 3b is a block diagram of the checking logic unit of unit 300. A NOR gate 320 receives at its inputs LastAD[3:0]. An OR gate 322 receives as its input LastAD[3:1]. The output of the OR gate 322 is delivered to an input of an AND gate 324 along with signal LastAD[0]. The outputs of the NOR gate 320 and the AND gate 324 are received at an OR gate 326.

[0020] The output of the NOR gate 320 becomes asserted (logical "1") if none of the LastAD[3:0] signals indicates a receipt of an align primitive during the last four samples of the non-rate verified align detect signal 215, indicating that an align sequence is not being recognized at the target frequency.

[0021] The output of the AND gate 324 becomes asserted if there are more than one K28.5 characters sampled in a 4 byte sequence. If either the output of the NOR gate 320 or the output of the AND gate 324 becomes asserted, then a nonalignddetected signal 327 becomes asserted.

[0022] Figure 4 is a diagram of the state machine 400. The state machine 400 is essentially a counter that looks for a predetermined number of align detects and resets any time one of the disqualifying events mentioned above in connection with Figure 3b occurs (nonalignddetected signal 327 asserted). The final state holds the rate verified align detect signal 225 asserted until an acknowledge signal is returned from the serial interconnect controller 200 core logic.

[0023] Upon receiving an asserted reset signal 410, the state machine starts at an idle state 410. An assertion of the non-rate verified align detect signal 215 causes the state machine to enter the align 1 state. An assertion of the nonalignddetected signal 327 causes the state machine to return to the idle state 410, and if no assertion of the nonalignddetected signal 327, then an assertion of the non-rate verification align detect signal 215 causes the state machine to enter an align 2 state. An assertion of the nonalignddetected signal 327 causes the state machine to return to the idle state 410, and if no assertion of the nonalignddetected signal 327, then an assertion of the non-rate verification align detect signal 215 causes the state machine to enter an align 3 state. An assertion of the nonalignddetected signal 327 causes the state machine to

return to the idle state 410, and if no assertion of the nonalignd etected signal 327, then an assertion of the non-rate verification align detect signal 215 causes the state machine to enter an align 4 state. An assertion of the nonalignd etected signal 327 causes the state machine to return to the idle state 410, and if no assertion of the nonalignd etected signal 327, then an assertion of the non-rate verification align detect signal 215 causes the state machine to enter an align 5 state. An assertion of the nonalignd etected signal 327 causes the state machine to return to the idle state 410, and if no assertion of the nonalignd etected signal 327, then an assertion of the non-rate verification align detect signal 215 causes the state machine to enter an align 6 state. An assertion of the nonalignd etected signal 327 causes the state machine to return to the idle state 410, and if no assertion of the nonalignd etected signal 327, then an assertion of the non-rate verification align detect signal 215 causes the state machine to enter an align 7 state. An assertion of the nonalignd etected signal 327 causes the state machine to return to the idle state 410, and if no assertion of the nonalignd etected signal 327, then an assertion of the non-rate verification align detect signal 215 causes the state machine to enter the assert align detect state 420. It is during the state 420 that the rate-verified align detect signal is asserted to the core logic. The state machine remains in state 420 until the acknowledge signal 411 is returned from the core logic. The state machine then returns to the idle state 410 and the process repeats.

[0024] Although the state machine 400 includes counting eight assertions of the non-rate verified align detect signal 215, other embodiments are possible using other number counts. The appropriate count number may be a function of the factors that may introduce errors in the signature of an align sequence at the target rate.

[0025] Figure 5 is a flow diagram of one embodiment of a method for rate verification of an incoming serial alignment sequence. At block 510, an incoming serial stream is received. Then, at block 520 a determination is made as to whether an align sequence is recognized in the incoming serial stream. When an align sequence is recognized, then processing proceeds to block 530. At block 530, a check is made to determine if an appropriate number of align primitives are received during a predetermined number of clock periods. Blocks 540 and 550 indicate that if the number of received align primitives matches the predetermined number, then a rate-verified align detect signal is asserted.

[0026] 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.

[0027] 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:

an input to receive a non-rate verified align detect signal;

a rate verification unit to determine whether an appropriate number of align primitives are received during a predetermined number of clock periods; and

an output to deliver a rate-verified align detect signal.

2. The apparatus of claim 1, the rate verification unit including a shift register that is clocked at a target clock rate.

3. The apparatus of claim 2, the rate verification unit further including a checking logic unit.

4. The apparatus of claim 3, the rate verification unit further including a state machine.

5. The apparatus of claim 2, the shift register including a first flip flop to receive a non-rate verified align detect signal and to output a first Last Align Detect signal.

6. The apparatus of claim 5, the shift register further including a second, a third, and a fourth flip-flop, the second flip-flop to receive the first Last Align Detect signal and to output a second Last Align Detect signal, the third flip-flop to receive the second Last Align Detect signal and to output a third Last Align Detect signal, and the fourth flip-flop to receive the third Last Align Detect signal and to output a fourth Last Align Detect signal.

7. The apparatus of claim 6, the checking logic unit to receive the first, second, third, and fourth Last Align Detect signals from the shift register, the checking logic to assert a nonaligndetected signal if each of the values of the first, second, third, and fourth Last Align Detect signals are zero.

8. The apparatus of claim 7, the checking logic further to assert the nonaligndetected signal if more than one K28.5 characters are sampled in a 4-byte sequence.

9. The apparatus of claim 7, the state machine to count up to n align detects, the count to increase each time the non-rate verified align detect signal is asserted and the count to reset each time the nonalignddetected signal is asserted.

10. The apparatus of claim 9, the state machine to cause the rate-verified align detect signal to be asserted.

11. The apparatus of claim 10, the state machine to keep the rate-verified align detect signal asserted until an acknowledge signal is received.

12. A system, comprising:

a serial interconnect host controller including

a data recovery circuit to output a non-rate verified align detect signal;

a rate verification unit to determine whether an appropriate number of

align primitives are received during a predetermined number

of

clock periods and to deliver a rate-verified align detect signal;

and

a system component coupled to the serial interconnect host controller via a serial interconnect.

13. The system of claim 12, wherein the serial interconnect is implemented according to a Serial ATA specification.

14. The system of claim 13, wherein the system component is a mass storage device.

15. The system of claim 14, the rate verification unit including a shift register that is clocked at a target clock rate.

16. The system of claim 15, the rate verification unit further including a checking logic unit.

17. The system of claim 16, the rate verification unit further including a state machine.

18. A method, comprising:

receiving a serial input stream;

detecting an align sequence;

determining whether an appropriate number of align primitives are received during a predetermined number of clock periods; and

generating a rate-verified align detect signal.

19. The method of claim 18, wherein determining whether an appropriate number of align primitives are received during a predetermined number of clock periods includes using clock periods that conform to a target clock rate.

20. The method of claim 18, further comprising determining whether the align primitives are being received at a target rate.