DS31256
256-Channel, High-Throughput HDLC Controller

REVISION B2 ERRATA

The errata listed below describe situations where DS31256 revision B2 components perform differently than expected or differently than described in the data sheet. Dallas Semiconductor intends to correct these errata in subsequent die revisions.

This errata sheet only applies to DS31256 revision B2 components. Revision B2 components are branded on the top side of the package with a six-digit code in the form yywwB2, where yy and ww are two-digit numbers representing the year and work-week of manufacture, respectively. To obtain an errata sheet on another DS31256 die revision, visit our website at www.maxim-ic.com/errata.

1. HIGH-SPEED Rx FIFO READ/WRITE POINTER SYNC

Description:
A bug has been discovered in the Rx FIFO block of the DS31256. The odds of encountering it are extremely low since the system is designed to prevent overflows, but it can occur under the following conditions:

- Only in applications running at a port speed higher than 28.87MHz, which corresponds to high-speed ports 0, 1, and 2.
- When an overflow occurs on a packet that has a size equal to (4n + 1) bytes, where n is an integer. The overflow must occur on byte 4n of the packet.

The bug does not affect the Tx side of the DS31256.

Symptoms:
When this bug occurs, the write-side state machine passes the read-side state machine. Because of this the chip can exhibit a number of symptoms, including:

- It can continue to operate normally if no other overflows occur.
- It can get into a state where certain packets sit in the FIFO and every packet received causes an older packet to come out. Thus, latency is exhibited between packet reception and packet being written to memory.
- The receive side of the chip can lock up. In this condition, the overflow bit would be continuously set whenever there is receive traffic.
- If the FIFO is smaller than the packet size, the FIFO can repeatedly write data out including packets bigger than the FIFO size until an end of a packet is reached.

Work Around:
The bug is detected only by the ROVFL bit (bit 5) in the SDMA register. This status may not make it out to the done-queue descriptor for the channel. This status is detected by either polling the SDMA register or through an interrupt. On detection, one of the following work arounds is recommended:

Work Around #1:
If it is acceptable, a full chip reset should be performed including the DMA data structures. Transmit traffic can be held in a queue while this occurs if the latency is tolerable so that transmit traffic is not lost.

Work Around #2:
Otherwise, a channel specific reset can be performed on the high-speed channels as follows. These work arounds assume that all RCHEN bits in the CPnRD indirect registers are set to 0 during the initialization stage.

1. Disable Layer 1 by taking the chip out of high-speed mode. RPnCR.RPnHS = 0
2. Wait for the Rx FIFO to empty in case it is in a state where the FIFO needs to just empty out the present packet. The time recommended would be (the size of the Rx FIFO in blocks x 4 x 1.5 x 30) nanoseconds.
3. In the Rx FIFO, reinitialize the Rx FIFO starting block pointer for all high-speed channels. This has the effect of resyncing the read and write sides of each channel's Rx FIFO. This write to the starting block pointer may have to be performed multiple times to be successful. Since the probability of successfully performing this step varies based on the FIFO size, the number of packets in the FIFO at that time, PCI bus bandwidth, etc., it is recommended that this be performed 50 times initially.
4. Enable Layer 1 by putting the chip back in high-speed mode. RPnCR.RPnHS = 1
5. Put the chip in local loopback. RPnCR.LLB = 1
6. Transmit a test packet and check to make sure the packet is received with an acceptable latency. This confirms that the reset was successful. If the reset was unsuccessful, repeat the 6 steps again.
7. Remove the channel from local loopback mode. RPnCR.LLB = 0

While this reinitialization process is being performed, the host software must continue to service the Rx done queue and replenish the Rx free queue buffers. If the Rx free queue ever runs out of buffers, the DS31256 Rx FIFO and Rx DMA reach a deadlock state and remain there until more buffers are supplied. While in this deadlock state, the channel on which the bug occurred cannot be reset through the re-initialization sequence.

The re-initialization procedure is only valid if the read-side state machine is idle. When the bug occurs, the affected channel could be in the process of writing data out. This results in a low probability per attempt of the Rx FIFO starting block pointer write being successful, so multiple sequential writes to the starting block pointer are employed to improve the success rate.

Work Around #3:
If the DMA caches are not enabled, a third work around is available.

1. Stall the process of queuing up more Tx packets and build these packets in the main host memory queue. They can be added to the Tx pending queue once the reset is complete.
2. Make sure that all the entries in the Tx pending queue have been processed and the transmit pending-queue write pointer (TPQWP) and transmit done-queue read pointer (TPQRP) are equal.
3. Wait for the data present in the Tx FIFO to be transmitted out. This time would be equivalent to (size of Tx FIFO in blocks x 16 x clock period in ns of lowest speed port in high-speed mode x 1.5) nanoseconds.
4. Place the chip in test mode. TEST.FT = 1
5. Disable Layer 1 by taking the chip out of high-speed mode. RPnCR.RPnHS = 0
6. Wait for the Rx FIFO to empty in case it is in a state where the Rx FIFO needs to just empty out the present packet. The time recommended would be (the size of the Rx FIFO in blocks x 4 x 1.5 x 30) nanoseconds.
7. In the Rx FIFO, reinitialize the Rx FIFO starting block pointer for all high-speed channels. This has the effect of resyncing the read and write sides of each channel's Rx FIFO.

8. Read the internal write and read configuration RAM values for the high-speed channels being reset. Make sure the two values are equal for the three high-speed ports. If the values are not equal, repeat step 7. Loop through steps 7 and 8 until the channels are reset.
9. Remove the chip from test mode. TEST.FT = 1
10. Enable Layer 1 by putting the chip back in high-speed mode. RPnCR.RPnHS = 1
11. Re-enable the queuing of packets on the Tx pending queue.

The reason for stalling the queueing of Tx packets is as follows. When the chip is placed in test mode, the transmit side of the chip is also stalled and any transmit packets in the chip can be transmitted erroneously. (They will be received on the remote end as packets with bad CRCs and lengths).

The advantage of this work around over work around #2 is that the reset of the receive side can be guaranteed by reading the two RAMs through the test mode.

Work Around #4:
If only one high-speed port is in use, a hardware work around might be used to prevent this bug from occurring. One of the unused high-speed ports should be wired to the same receive data stream and clock as the port that needs to be protected. The redundant received traffic would also have to be processed as done-queue entries are generated, but could then be discarded. This work around is still being investigated, but can be validated in simulation by Dallas if we are requested to do so.

2. OVERFLOW—SMALL PACKETS WITH CRC STRIPPING

Description:

A bug has been discovered in the receive (Rx) FIFO block of the DS31256 that occurs when all of the following conditions are simultaneously met:

- CRC stripping bit is enabled,
  i.e., RHCD[6] (RCS) = 1.
- The chip is set to either a 16 bit CRC or a 32 bit CRC,
  i.e., RHCD[3:2] (RCRC1, RCRC0) is either 01 or 10.
- The total number of bytes received on a given channel is (32,768 x n - 4) since the chip was initialized, where n is an integer. For this calculation, packets that are nonmultiples of 4 should be rounded to the nearest multiple of 4.
- A valid packet (no errors) of the following size is received. The packet size does not include the CRC:

CRC MODE	PACKET SIZES FOR BUG TO OCCUR
16-bit CRC	11, 12 bytes
32-bit CRC	9, 10, 11, 12 bytes
No CRC	Bug does not occur

- During receipt of the offending packet, the DMA must receive access to the PCI bus for this specific channel. If PCI access is not granted in that window of time, the channel will recover (read- and write-side state machines will resynchronize without incident).

NOTE: This bug does not affect the transmit side of the DS31256.

Symptoms:
When this bug occurs, the read-side state machine shoots past the write-side state machine. Due to this, the chip can exhibit several symptoms:

- Continued normal operation if no other instances of the bug occur.
- A certain number of older packets/data can come out of the chip twice. Furthermore, there may be latency between the time the packet is written into the chip, and when it is read out.
- An overflow can occur. After the initial bug-induced overflow, the likelihood subsequent overflows on that channel increases significantly.

Work Around:

Work Around #1:
If the system will not be exposed to **valid** packets of the sizes that can cause this error, no action is required. Invalid packets of these sizes do not pose a problem.

Work Around #2:
If the packets sizes that can cause the error can occur in your system, do not use the CRC stripping function,

i.e., RHCD[6] (RCS) = 0.

Work Around #3:
It is possible for the CPU to actively monitor (poll) various memories in the DS31256 to detect when this condition has occurred, or possibly detect it as being imminent and rectify the problem before it occurs. This would add considerable overhead to both the processor and the PCI bus, however, so it is not a favored approach. Should still this be of interest, the please contact the factory for assistance in implementation.

3. V.54 LOOP-UP/DOWN DETECTOR DOES NOT FUNCTION PROPERLY

Description:
The V.54 loop-up/down detector will correctly identify the 2E7-1 loop-up pattern but will mistake the all-ones pattern that follows the loop-up pattern for the V.54 loop-down pattern. The loop-down pattern is an inverted 2E7-1 pattern. In the device, after a loop-up detection, the device inverts data into the detector and searches for the 2E7-1 pattern. The all-ones pattern that follows the loop-up pattern is inverted and presented to the detector as an all-zeros pattern. An all-zeros pattern satisfies the requirement for the 2E7-1 pattern because of the errata, and is therefore mistaken for the loop-down pattern.

Work Around:
None.

Reference: *Application Note 3818* describes how to use the receive BERT function in the DS31256 to perform Fractional-T1 (FT1) loop-up or loop-down detection (V.54) as described in the Fractional T1.403 Annex B spec.