Notification
for KS32C50100 Design Bugs

2 February, 2000
(Ver 4.0)

Samsung Electronics Co., Ltd.

CONTENTS

1. HDLC frame with 2 flags

1.1 Symptom

A HDLC frame received after 2 flags (figure) has always CRC error.

Frame 1	Flag	Flag	Frame 2

normal receive CRC Error

1.2 Cause

In state machine, two flags case was not considered. There is no work-around.

2. HDLC Rx DMA stops after HDLC underrun

2.1 Symptom

During the HDLC Tx mode (especially for more than 10Mbps data rates), when Tx underrun occurs, very rarely Tx mode hangs and data transmit will be stopped. HDLC DMA hang can be released by resetting the DMA Tx.

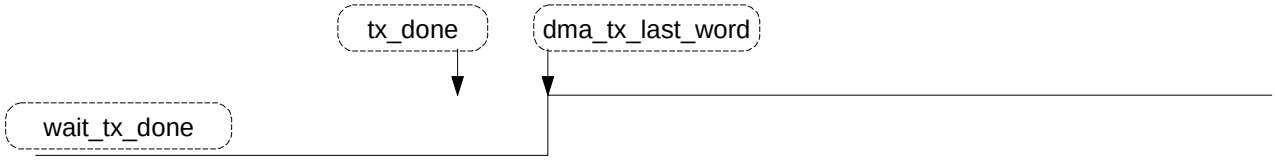
2.2 Mechanism causing the bug

The wait_tx_done signal in DMA Tx module will be set to "1" when DMA completes moving data into TxFIFO. It will be cleared to "0" when tx_done signal, which tells the completion of data transfer into TxFIFO, is asserted from XMTR. DMA will check the level of wait_tx_done and when it is "0", DMA will assert BUS Request to move data into TxFIFO.

Generally, if a frame has transmitted successfully, dma_tx_last_word signal is asserted earlier than tx_done signal. And then wait_tx_done is set to "1" and returns to "0".

But if Underrun happens, tx_done signal is asserted. When it happens, tx_done is asserted at 1 cycle earlier than dma_tx_last_word. As a result, wait_tx_done isn't set to "0", at all. Therefore, DMA Tx does not ask Bus Request for data transmission and system is hanged eventually.

Normal Operation



Underrun

2.3 Remedy

KS32C50300 will be revised to resolve the problem. For the current devices, there are two software turn around solutions.

1) Software Turn Around 1

In the interrupt service routine, whenever Tx Underrun happens, turn OFF and ON MFF in HMODE. It will not degrade system performance and can fix the bug.

2) Software Turn Around 2

In the interrupt service routine, whenever Tx Underrun happens, turn ON TxABT in HCON. It will not degrade system performance.

3. HDLC DMA operation error with Rx alignment widget

3.1 Symptom

When RxWA value of HCON register is one of 1, 2 or 3 (not 0) and the receiving frame length is larger than Rx Data Buffer size, it looks like one word missing on the boundary between two Data Buffers.

3.2 Mechanism causing the bug

When RxWA value of HCON register is one of 1, 2 or 3 (not 0), if the receiving frame length is larger than Rx Data Buffer size, one word is stored to the first location of the next Data Buffer and then it is overwritten by the first data of the next Data Buffer.

3.3 Remedy

KS32C50300 will be revised to fix this design bug.

1) Software Turn Around

When Rx Alignment Widget (RxWA) is not 0, 1 word space between Rx Data Buffer will fix this problem.

4. HMLFR of HDLC mis-operates in non-multiple of 4

4.1 Symptom

When HMLFR(HDLC Maximum Length Frame Register) is set to a multiple of 4 only it operates normally. That is, as HMLFR=60, the FLV bit is set receiving over 61 bytes, which is normal. As HMLFR=61, 62, or 63 which are not a multiple of 4, the FLV bit is set receiving over 65 bytes. In normal operation, the FLV bit should be set if the number of bytes received is larger than the HMLFR value.

5. Ethernet CAM memory read wrong

5.1 Symptom

As increasing the operating current or the temperature, the read data from CAM may be wrong sometimes. This occurs by a lack of timing margin.

5.2 Work-around

Write operation is good, but read may be errored. Instead of reading the CAM you can use some global parameter for CAM in software.

6. Ethernet Tx BDMA could be stopped by BTxMSL and BTxBRST

6.1 Symptom

In case of BTxMSL=111 or 110, the Ethernet Tx BDMA could be stopped by the BTxBRST value.

6.2 Work-around

Not use 22 to 28 as BTxBRST value.

7. Ethernet Rx BDMA could be stopped by overflow

7.1 Symptom

Overflowing Rx BDMA FIFO, the Ethernet Rx BDMA could be hang.

7.2 Mechanism causing the bug

If Rx BDMA FIFO overflows two EOFs can exist in neighbouring location of the BDMA FIFO. If EOF is wrote in the buffer memory the Rx BDMA controller updates the buffer descriptor. The concatenated two EOFs have the Rx BDMA controller stay at the buffer descriptor state. That is, the buffer descriptor state is not completed. This case occurs rarely.

8. Ethernet BDMA one word missing

8.1 Symptom

When the Ethernet BDMA writes a received data into the external memory area, very rarely BDMA skips writing one word of information and jumps to the next address area. As a result, the stored data in the external memory looks like have a hole in memory area.

8.2 Mechanism causing the bug

If the Ethernet BDMA requests the system bus in order to transfer the received data, then the bus arbitor gives the corresponding Acknowledge to the Ethernet BDMA. This symptom occurs if EOF (End Of Frame) is transferred from MAC FIFO to the Ethernet BDMA FIFO before the Ethernet BDMA receives the Acknowledge.

In this situation, the status of the current frame is transferred to the BDMA FIFO, while the system manager gives the BDMA the corresponding Acknowledge signal. However, the BDMA does not recognize the acknowledge by the EOF signal from the MAC FIFO.

As a result, the BDMA receives the status from the MAC FIFO, while the memory controller operates in read mode.

9. Ethernet BDMA stores frame mixed

9.1 Symptom

The Ethernet BDMA has designed to write the only one received frame into one frame memory area that was defined by the Buffer Descriptor. But, sometimes a frame memory shows that it has two or more frames rather than one.

9.2 Mechanism causing the bug

This situation can be occurred rarely while KS32C50300 receives heavy traffic. The BDMA updates the current Buffer Descriptor after writing EOF. Before starting the update of the Buffer Descriptor, if the BDMA requests the bus to write the next frame data and then receives the acknowledge signal the current Buffer Descriptor is not updated. Moreover, the BDMA writes the next frame data to the current frame memory.

10. BDMA Tx hangs after Ethernet underrun (Tx_COUNT Error)

10.1 Symptom

There will be a possibility of Ethernet BDMA Tx hang when Underrun occurs during the data transmission with enabled EnUnder bit of MACTXCON register.

10.2 Mechanism causing the bug

Tx BDMA in KS32C50300 has two counters, TX_BYTE and Tx_COUNT. TX_BYTE counts the number of bytes moved from memory to BDMA. Tx_COUNT counts the number of bytes moved from BDMA to MAC.

If Underrun happens while data is being transmitted by Tx BDMA with enabled EnUnder bit of MACTXCON, Ethernet MAC generates interrupt and sends Tx_done signal to BDMA. Tx_done signal clear Tx_COUNT in BDMA.

When Underrun happens, Tx_done signal is generated before than Tx_EOF signal which clears TX_BYTE. The Tx_EOF signal was designed to be asserted only when frame length and TX_COUNT value are same. So, if Tx_done has asserted earlier than Tx_EOF because of Underrun, Tx_COUNT will be cleared by Tx_done, and Tx_EOF can not be asserted, at all.

The MAC Tx waits for the Tx_EOF in abort mode and BDMA will not request another data transfer because it already has transmitted the data from memory.

11. Ethernet BDMA writes the receive rata into frame buffer with OwnerBit=0

11.1 Symptom

The BDMA writes data into memory area where has designated by Buffer Descriptor. To do this, Owner bit in the Buffer Descriptor is needed to be "0", which means Not Owner. If the Owner bit in a Buffer Descriptor is "1", BDMA should not write data into the memory pointed by the Buffer Descriptor.

Rarely, it occurs that BDMA writes data into the memory pointed by the Buffer Descriptor with Owner bit = "1".

11.2 Mechanism causing the bug

Rx BDMA updates the Buffer Descriptor (UPDATE_BD) whenever it completes the current frame receiving.

It takes 3 clocks to set NOTOWNER signal after Buffer Descriptor is updated. During this period, if BDMA Rx FIFO has next frame, BDMA will assert nREQ_Rx and write data into NOTOWNER area after it gets nACK_Tx signal.

12. When buffer descriptor in Ethernet BDMA is NULL, it still writes receive data.

12.1 Symptom

Even though Rx BDMA Buffer Descriptor is empty or NULL, the current KS32C50300 BDMA still regards there is data in buffer descriptor and try to move data into memory area. The memory area will be designated by data pointer in NULL address.

12.2 Mechanism causing the bug

BRxEn can disable Rx BDMA operation. BRxEn will be disabled 3 clocks later when Buffer Descriptor update was completed and it meets NULL list. During this period, if BDMA Rx FIFO has data, Rx BDMA will assert nREQ_Rx and regards null address area as data pointer and writes data in it right after Rx BDMA has nACK_Rx.

13. Ethernet BDMARxLSZ counter error

13.1 Symptom

The Maximum Size Over error occurs when BDMARxLSZ value is bigger than the maximum received frame size even though Maximum Size Over error was designed to happen only when frame length count (WCOUNT) and BDMARxLSZ have same value.

13.2 Mechanism causing the bug

The WCOUNT value is based on multiple of 4 (Word = 4 Bytes) so that frame length such as 4*B+1, 4*B+2, 4*B+3 will be regarded as 4*B+4 (Round Off). In contrast, BDMARxLSZ is based on multiple of 4, but it will be truncated.

Therefore, Maximum Size Over error can be happened even though frame length is 10 bytes and less than BDMARxLSZ.

13.3 Remedy

KS32C50300 WCOUNT will be revised to count correct received frame length.

1) Software Turn Around

For the current device, make BDMARxLSZ to have bigger than 10 than the affordable maximum frame size.

14. Errors when Ethernet BDMA is disabled by software

14.1 Symptom

1) When BDMA transmits frames continuously increasing the frame length by one byte, we observed that BDMA transmits twice the last word of frame whose frame length are 137, 138, 139, 140, 1265, 266, 267, 268, 649, 650, 651, 652, 1033, 1034, 1035, and 1036.

   The frame length can be calculated by Destination Address (6) + Source Address (6) + LengthOrType (20) + LLC data + CRC (4).
2) During the BDMA operation, if CPU disables and then enables BTxEn or BRxEn signals, BDMA operation might be halt or system will be hanged.
3) While Cache is enabled and BDMA transmits data continuously, if GDMA operates together, BDMA Tx mode hangs at a special timing.

14.2 Results

KS32C50300 BDMA will be revised to operate continuously and complete the data transmission until it transmits whole data even though BTxEN or BRxEn has deactivated

1) Software Turn Around

   For the current device, some of instructions which deactivates BTxEn should be removed for stable operation.

15. One word length frame by Ethernet OverMax error

15.1 Symptom

If the length of the previous frame is same as BDMARxLSZ value, the next frame has one word length and good status.

15.2 Mechanism causing the bug

If the length of frame is over BDMARxLSZ the write address is not incremented for protection the next frame memory. If the frame length is exactly same as BDMARxLSZ the address of the enxt frame is not incremented. This is bug.

16. Ethernet buffer descriptor status update error

16.1 Symptom

The status field of the buffer descriptor has incorrect value that comes from the data of the next frame.

16.2 Mechanism causing the bug

When Read and Write operation of the BDMA FIFO occur concurrently, where Read means storing the status data of FIFO to the buffer descriptor and Write means transferring the receive data of MAC to FIFO. The Write operation is executed first

by priority. During Write execution, Read Pointer of the FIFO is incremented which is unwanted operation. As a result, the first data of the next frame is stored to the status field of the buffer descriptor.