A. Affected Silicon Revision

Errata details published in this document refer to the following silicon:

netX500, Revision A (Step A, ROM Rev. 0, Boot loader major vers. 0x41)

B. Document Revision History

Revision	Date	Description	Author
1.0	March, 2006	Created, added Errata 1 - 3	J. Lipfert
1.1 (2.0)	June, 2006	Added Erratum 4	A. Jacob
1.2	July, 2006	Added Errata 5 and 6, changed Document Revision numbering (previous Revision 2.0 is now Revision 1.1)	J. Lipfert
1.3	August, 2006	Revised Errata 1 – 6, added Errata 7 - 10	J. Lipfert
1.4	March, 2007	Added Erratum 11	J. Lipfert
1.5	June, 2007	Added Errata 12 and 13	J. Lipfert
1.6	August, 2008	Added Errata 14 and 15, extended Silicon Rev. information	J. Lipfert
1.7	Nov., 4th, 2009	Added Erratum 16	J. Lipfert
1.8	April, 28th, 2010	Added Errata 17,18, confined workaround for Erratum 6	J. Lipfert
1.9	Dec, 8th, 2010	Added Erratum 19	J. Lipfert M. Melzer

© Hilscher GmbH, 2010

C. Errata Summary

Nr.	Description
1	SDRAM Controller: SDRAM collides with parallel Flash after Reset
2	Extension Bus boot mode: Boot mode does not work as intended
3	DPM mode: Global DPM ARM side interrupt status flag will never be set
4	I^2C Core: No Multi-Master-Support; no ACK / NAK detection
5	PCI Arbiter: Internal PCI Master collides with external master 0
6	PCI Target: Read Errors (incorrect data) when reading handshake cells
7	PCI Arbiter: Error in arbiter control register (DPMAS_ARB_CTRL)
8	Hostinterface: Error in PIO configuration register (DPMAS_IO_MODE_0)_
9	SPI: FIFO Access during SPI transfer, leads to data corruption or malfunction
10	DPM boot mode: Host system must always provide 64K address space
11	SPI boot mode: Booting from some SD-Cards does not work
12	SPI: Slave mode does not work properly
13	Reset Control Register: Register not protected by netX locking mechanism
14	CLKOUT Signal: Signal can not be reactivated after internal reset or reset from RSTINn
15	SDRAM Controller: Access conflict between ARM CPU and LCD-Controller
17	Hostinterface: Error in configuration register regarding DPM ISA mode_
18	Internal PHYs: Error in 10 Mbit half duplex mode
19	Host Interface: DPM access time with Hilscher standard DPM layout is unpredictable

© Hilscher GmbH, 2010

D. Errata Details:

1 SDRAM Controller: SDRAM collides with parallel FLASH after Reset

Issue:

In netX systems equipped with SDRAM and parallel FLASH, connected to the netX memory bus that use the parallel FLASH as boot medium, the following problem may occur:

Upon a reset (regardless if issued by hardware or software), all internal netX clock sources are stopped immediately, which also includes the SDRAM clock signal. If the reset is issued while there is a read access to the SDRAM in progress, the access will not be completed and the output drivers of the SDRAM will remain activated and will hence continue to drive the memory bus.

In this scenario, any attempt to boot from the parallel flash (which shares the memory bus with the SDRAM) will fail due to a signal collision between SDRAM and FLASH.

Solution / Workaround:

a) Use a different boot medium (SPI Flash, DPM) if applicable.

b) Use an additional boot memory (SPI Flash/EEPROM, I2C EEPROM) containing a second stage boot loader that initializes the SDRAM interface (and hence allows the SDRAM to deactivate its drivers) before continuing to boot from the parallel FLASH.

c) Use a high speed bus switch to isolate the SDRAM from the memory bus during accesses to the parallel FLASH.

This issue will be fixed in the next silicon revision (the ROM Boot loader will initialize the SDRAM Interface before continuing to check for firmware in a parallel FLASH).

2 Extension Bus boot mode: Boot mode does not work as intended

Issue:

Due to the error described in Erratum 8, booting from a parallel FLASH connected to the Extension Bus of the netX does not work as intended, as the ROM Boot loader does currently not configure the RDY pin for Hostinterface mode, rendering address line A13 unusable while booting.

Solution / Workaround:

- Use a pulldown resistor on address line A13 to provide a low level on this otherwise floating signal.
- Either use a firmware with a size, that does not exceed 8K, or use a small (less than 8K) second stage loader firmware (which can also reside in the Extension Bus FLASH), that reconfigures the DPMAS_IO_MODE_0 (0x00103620) register (sets Bit 14, enabling address line A13) and then loads the main firmware (please contact netX Support if you require an appropriate solution).

Although initially planned, this issue will not be fixed in future silicon revisions (modified ROM boot loader) for compatibility reasons. This erratum will however disappear when Erratum 8 will be fixed in a future silicon revision.

© Hilscher GmbH, 2010

3 DPM mode: Global DPM ARM side interrupt status flag will never be set

Issue:

When working with DPM interrupts, the flag for global IRQ signaling (Bit 31 in register DPMAS_INT_STA1 (0x001034E0)) will never be set.

Solution / Workaround:

Use the 8 Bit IRQ Vector located in the same register (Bits [23:16]) instead. Any vector value other than 0x00 indicates a pending DPM interrupt.

4 I^2C Core

4.1 Missing Multi-Master-Support

Issue:

The I^2C Core of the netX does not provide bus arbitration functionality.
Hence it is not possible to build a Multi-Master-System with the netX.

Solution / Workaround:

Currently none.

4.2 ACK / NAK detection:

Issue:

ACK / NAK conditions on the bus can not be detected by the netX
(the internal ACK counter does not represent the correct value).

Solution / Workaround:

a) FLASH or EEPROM devices which require ACK polling after a write or erase command should be polled for their defined maximum write cycle time.

b) Special function devices like the ATMEL™ Secure EEPROM AT88SC0104C provide a status register which can be used instead of ACK polling to determine success or failure of an operation.

© Hilscher GmbH, 2010

5 PCI Arbiter: Internal PCI Master collides with external master 0

Issue:

When using the internal PCI Arbiter of the netX (netX operating as PCI Host), the internal master and an external master connected to REQ/GNT signal pair 0 can not be used concurrently, as this will lead to malfunction of the arbitration state machine, resulting in a system lockup.

Solution / Workaround:

a) Do not use the REQ/GNT signal pair 0 (pin E16 and D17). In that case only 3 external PCI masters can be used.

OR

b) Do not use the internal PCI master (do not set up DMA transfers in the netX).

6 PCI Target: Read Errors (incorrect data) when reading handshake cells

Issue:

When handshake cells mapped into the DPM area are read from the host, the target FIFO is filled with incorrect data, also affecting consecutive target reads.

Solution / Workaround:

- Do not map and enable Handshake cells (→ Registers DPMAS_HSCR0 – 15, however Bit 0 of the appropriate DPMAS_HSC Registers, configuring 8 or 16 Bit handshake cell mode has to be programmed accordingly (set for 16 Bit, cleared for 8 Bit wide handshake cell).
- Instead, map the host controlled handshake register block (0x00103200 – 0x001032FF) into a desired DPM area and access the handshake cells directly through this memory window.
- When responding to a handshake interrupt (generated from netX side), read ALL handshake cells consecutively (0 to 15) and compare them to the previous data image of the cells, as intermediate handshake interrupts (generated while reading the cells) may have been lost due to the read-ahead functionality of the target FIFO.

Addendum from April 28th, 2010:

Use and read only handshake cells 0 to 7 instead of 0 to 15, since accessing cells 8 to 15 would (due to the read-ahead) again access cells 0 to 7 (address wrap around). This would result in the need of constantly polling the handshake cells and render the handshake interrupt functionality useless.

© Hilscher GmbH, 2010

7 PCI Arbiter: Error in arbiter control register (DPMAS_ARB_CTRL)

Background:

When the internal arbiter of the netX is not activated (Bit 7 of the arbiter control register not set), REQ and GNT signals REQ(3:1) and GNT(3:1) can be used as PIOs, controlled by the DPMAS_ARB_CTRL (0x001034A8) register.

Issue:

The appropriate bits ([29:24] and [21:16]) for controlling output enable and signal level of these pins can not be written at their intended bit positions.

Solution / Workaround:

In order to correctly write the upper bits of the arbiter control register, bits (21:16) must be written at bit positions (13:8) and bits (29:24) must be written at bit positions (21:16).

8 Hostinterface: Error in PIO configuration register (DPMAS_IO_MODE_0)

Issue:

Bit 14 of the DPMAS_IO_MODE_0 (0x00103620) register, intended to configure netX Pin D14 (DPM_RDY/EXT_RDY/PIO46 for Hostinterface mode or PIO mode, also affects Pin A13 (DPM_A13/EXT_A13/PIO48). As a result, address line A13 of the Hostinterface will never work, when the RDY pin is configured for PIO mode. This also affects booting from a parallel FLASH connected to the Extension Bus (see Erratum 2).

Solution / Workaround:

Do not configure the RDY pin as PIO whenever address line A13 of the Hostinterface is to be used (DPM mode or Extension bus mode).

© Hilscher GmbH, 2010

9 SPI: FIFO access while SPI transfer is active, leads to data corruption or malfunction

Issue:

When accessing the SPI FIFOs, while an SPI transfer is active, data in the FIFOs may get corrupted or the SPI core may lock up.

Solution / Workaround:

Disable the read/write flags in the SPI_CONTROL (0x00100C08) register before accessing the FIFOs.

Example:

```
POKE(Adr_spi_control_register,
     PEEK(Adr_spi_control_register) &~
(MSK_spi_control_register_CR_write|MSK_spi_control_register_CR_read));

/* Now access the FIFOs */

POKE(Adr_spi_control_register,
     PEEK(Adr_spi_control_register) |
MSK_spi_control_register_CR_write|MSK_spi_control_register_CR_read);
```

10 DPM boot mode: Host system must always provide 64K address space

Issue:

Unlike stated in the netX Boot loader specification (mentioning an 8K window), the current boot loader always requires an external host system to be able to address the complete 64K range, when the netX is supposed to boot via DPM (host system loads netX firmware to DPM).

Solution / Workaround:

Use an external boot memory (SPI FLASH/EEPROM, I^2C EEPROM) containing a second stage boot loader that configures the DPM according to your host system capabilities.

Although initially planned, this issue will not be fixed in future silicon revisions (modified ROM boot loader) for compatibility reasons.

© Hilscher GmbH, 2010

11 SPI boot mode: Booting from some SD / MMC - Cards does not work

Issue:

Some SD- or MMC-Cards do not respond fast enough to let the netX Boot loader detect and load a valid boot image residing on the card. Hence some cards will only boot after a second (manual) reset, while other cards will not boot at all.

Solution / Workaround:

Use an external boot memory (SPI FLASH/EEPROM, I^2C EEPROM) containing a second stage boot loader that allows higher card response times (available from Hilscher).

This issue will be fixed in the next silicon revision (ROM boot loader will be modified).

12 SPI: Slave mode does not work properly

Issue:

Due to an error in the SPI module, the slave mode of the netX SPI interface can not be used.

Solution / Workaround:

If required, use netX GPIOs or PIOs to realize a slave mode interface in software.

13 Reset Control Register: Register not protected by netX locking mechanism

Issue:

The netX Reset Control Register, RESET_CTRL (0x0010000C), meant to be protected against undesired accesses by the netX locking mechanism, can be read and written directly.

Solution / Workaround:

As the register and the functions it controls are working properly, no workaround is necessary. However, the following notes should be regarded:

a) Though currently not required, the unlocking procedure as described in the Program Reference Manual should still be implemented, in order to ensure software compatibility with future chip revisions that will implement the locking feature correctly.

b) Since step a) will leave the ASIC_CTRL Register area unprotected till the next access, the access to the Reset Control Register should be followed by a dummy read access to any of the other registers in that area (e.g. IOC_CR at 0x00100004).

© Hilscher GmbH, 2010

14 CLKOUT Signal: Signal can not be reactivated after internal reset or reset from RSTINn

Issue:

When initiating an internal reset (Firmware Reset, Watchdog Reset, Host Reset, XPEC Reset) or when a reset through the RSTINn signal occurs while the CLKOUT signal was running, the CLKOUT signal can not be activated again, until a Power On Reset was performed.

Solution / Workaround:

a) For predictable resets (Firmware Reset and Host Reset) use the following procedure BEFORE initiating the reset:

- Clear Bit 31 of DPM_ARM_CLKOUT_CFG register (address 0x00103604)
- Set Bit 30 of DPM_ARM_CLKOUT_CFG register (address 0x00103604)

  (Two separate accesses must be done, they may not be combined in one write access (check your compiler output!))

b) For a solution that solves the problem for all resets, use a system reset generator with an integrated watchdog that is automatically enabled when the watchdog input signal is switched from Hi-Z to a defined level (e.g. MAX6822/6823) and connect the watchdog input to the CLKOUT signal. As soon as the CLKOUT signal is activated, the MAX watchdog becomes active and is toggled by the CLKOUT signal. When the CLKOUT signal stops due to a reset, the MAX causes a Power On Reset of the netX, which allows reactivating the CLKOUT signal.

As this always extends any other reset to a Power On Reset, the RESET_CTRL register can no longer be used to determine the initial source of the last reset, since the following Power On Reset always clears the source of the previous reset. Hence a lockup situation causing a watchdog reset, can no longer be detected after restart of the system!

© Hilscher GmbH, 2010

15 SDRAM Controller: Access conflict between ARM CPU and LCD-Controller

Issue:

Under certain conditions an access conflict between ARM and LCD Controller may occur, that results in a flickering and unreadable screen, during a screen refresh. This was noticed when running Windows CE images including Internet Explorer, while rendering an image. During the screen refresh the image starts flickering.

This could only be seen when all of the following conditions apply:

- Displays with a resolution of 640*480 or higher are used
- SDRAMs with a write recovery time > 1 clk are used
- ARM CPU writes to SDRAM

Solution / Workaround:

Although the situation improves, when the ARM CPU writes to SDRAM only uncached and unbuffered, the only way to completely avoid this conflict here, is to limit the time, the ARM may continuously access the Data Channel by setting the following registers:

Register	Address	Default Value	Workarround value
MEM_PRIO_TIMESLOT_CTRL	0x0010 0180	0x0007 7777	0x0000 7777
MEM_PRIO_ACCESS_CTRL	0x0010 0184	0x3FFF FFFF	0x1FFF FFFF

16 moved

© Hilscher GmbH, 2010

17 Host interface: Error in Configuration Register regarding DPM ISA mode

Issue:

When using the netX500 in 16 Bit ISA Mode, the BE1_MODE Bits (23:21) of the DPM_ARM_IF_CFG0 (at address 0x00103608) register will usually be set to 3'b101 in order to activate the generation of the MEM_CS16n signal on the DPM_WRHn / PIO44 pin (B14). This setting however makes the internal Byte Enable 1 (High Byte) signal active high, while the appropriate ISA Bus Signal (SBHE, System Byte High Enable) connected to the DPM_BHEn / PIO43 pin (A14) is actually an active low signal.

As a result, the corresponding high byte will always be written inadvertently when the Host performs a single byte write access to the low byte, while the high byte will remain untouched when the host performs 16 Bit or single high byte write accesses.

Solution / Workaround:

- For new hardware designs or redesigns:

  Insert an inverter gate with 5V tolerant input (e.g.NC7SZ14, single gate) between the ISA Bus SBHE# signal and the DPM_BHEn pin.

- For existing hardware designs:

  - Set the BE1_MODE Bits to 3'b001 (High Byte is selected, when address line A0=1) and the HIF_MODE Bits to 3'b010 (8 Bit DPM mode). This will configure the netX500 DPM for 8 Bit mode and disable the MEM_CS16n signal generation, which causes the Host to make 8 Bit accesses only.

Alternatively:

  - If the ISA Bus of the system is exclusively equipped with true 16 Bit adapter cards, set the BE1_MODE Bits to 3'b000 (High Byte is selected when BHEn has low level). Further clear Bit 12 of the DPM_ARM_IO_MODE0 register (sets the DPM_WRHn / PIO44 pin to PIO mode), set Bit 12 of the DPM_ARM_IO_DRV_EN0 register (enables the output driver of PIO44) and clear Bit 12 of the DPM_ARM_IO_DATA0 register (drives PIO44 low).

    The MEM_CS16n signal (connected to DPM_WRHn / PIO44) is now constantly driven low, causing the ISA host to access in 16 Bit mode only. Of course this is not applicable if the ISA bus is shared with any 8 Bit devices and is rather a workaround than a clean solution.

© Hilscher GmbH, 2010

18 Internal PHYs: Error in 10 MBit half duplex mode

Issue:

When using standard Ethernet Communication with 10 MBit half duplex mode (used with legacy hubs only), the complete PHY gets stuck in case of a network collision.

As far as Real-Time-Ethernet protocols are concerned, this problem can only occur with EtherNet/IP or Modbus TCP when using hubs instead of switches (recommended), as all other protocols do not allow the use of hubs or 10 MBit mode.

For error details see attached problem description from Renesas at the end of this document.

Solution / Workaround:

Do not use legacy 10 MBit-only hubs. Use either switches or 10/100 MBit Dual Speed hubs, to make sure the netX Ethernet ports are connected with 100 Mbit or in full duplex mode only.

This erratum is fixed with all components of the 'Y' charge (9 digit charge number shows 'Y' at position 5 (nnnn**Y**nnnn).

19 Host Interface: DPM access time with Hilscher standard DPM layout is unpredictable

Issue:

When using the host interface in DPM mode along with the Hilscher standard DPM layout (or any other DPM layout providing access to the Handshake cell area), DPM access time may increase unpredictably due to a possible race condition between the host interface and other internal controllers (xPEC, ARM). As a result, all hardware designs that work with host CPUs which do not use or support the DPM RDY/BUSY signal but use DPM layouts that allow access to handshake cells, are subject to DPM access errors, resulting in wrong data read from the DPM or DPM write accesses being ignored. Such access errors occur, whenever the actual DPM access time exceeds the fixed access time used by the host CPU. Increasing the fixed access time on the host CPU is not a solution, since a maximum DPM access time can not be specified here.

Solution / Workaround:

a) Use a host CPU that supports the RDY/BUSY signal provided by the DPM to allow extending DPM access cycles if necessary.

OR

b) Do not use DPM layouts that allow access to the handshake cells or any other memory area except the internal RAM segments (maximum DPM access time to internal RAM areas is always deterministic).
   → Using standard loadable firmware provided by Hilscher along with a host CPU that does not support the RDY/BUSY signal is not possible!

© Hilscher GmbH, 2010

Summary of 10BT problem on EthernetPHY

Renesas Electronics Europe

April 27, 2010

© 2010 Renesas Electronics Europe. All rights reserved.

10BaseT Problem

- Illegal behaviour of COL signal during 10baseT half duplex collision
  - RX immediately follows to start TX
- PHY get stuck while in 10BaseT
  - Statistical behaviour
  - Depending on IPG

© 2010 Renesas Electronics Europe. All rights reserved.

Summary of 10BT problem on EthernetPHY

		cause of problem		possibility of problem	
		received data	async on preamble filter	data_valid (crs of receiver)	clk20
10BT HD	Jam RX->TX (collision)	normal preamble	OK	normal	normal
		normal preamble	NG	normal / no detect / frequent detection	normal /frequent absence
		corrupted preamble	OK	normal	normal
		corrupted preamble	NG	normal / frequent detecttion	normal / frequent absence
	Jam TX->RX (collision)	normal preamble	OK	normal	normal
		normal preamble	NG	normal / no detect / frequent detection / temporary stuck	normal / frequent absence / temporary stuck
		corrupted preamble	OK	normal / complete stuck	normal / complete stuck
		corrupted preamble	NG	normal / complete stuck	normal / complete stuck
	Packet	normal preamble	OK	normal	normal
		normal preamble	NG	normal / no detection	normal
		corrupted preamble	OK	normal	normal
		corrupted preamble	NG	normal / no detection	normal
10BT FD	Packet	normal preamble	OK	normal	normal
		normal preamble	NG	normal / no detection ※1	normal
		corrupted preamble	OK	normal	normal
		corrupted preamble	NG	normal / no detection ※2	normal

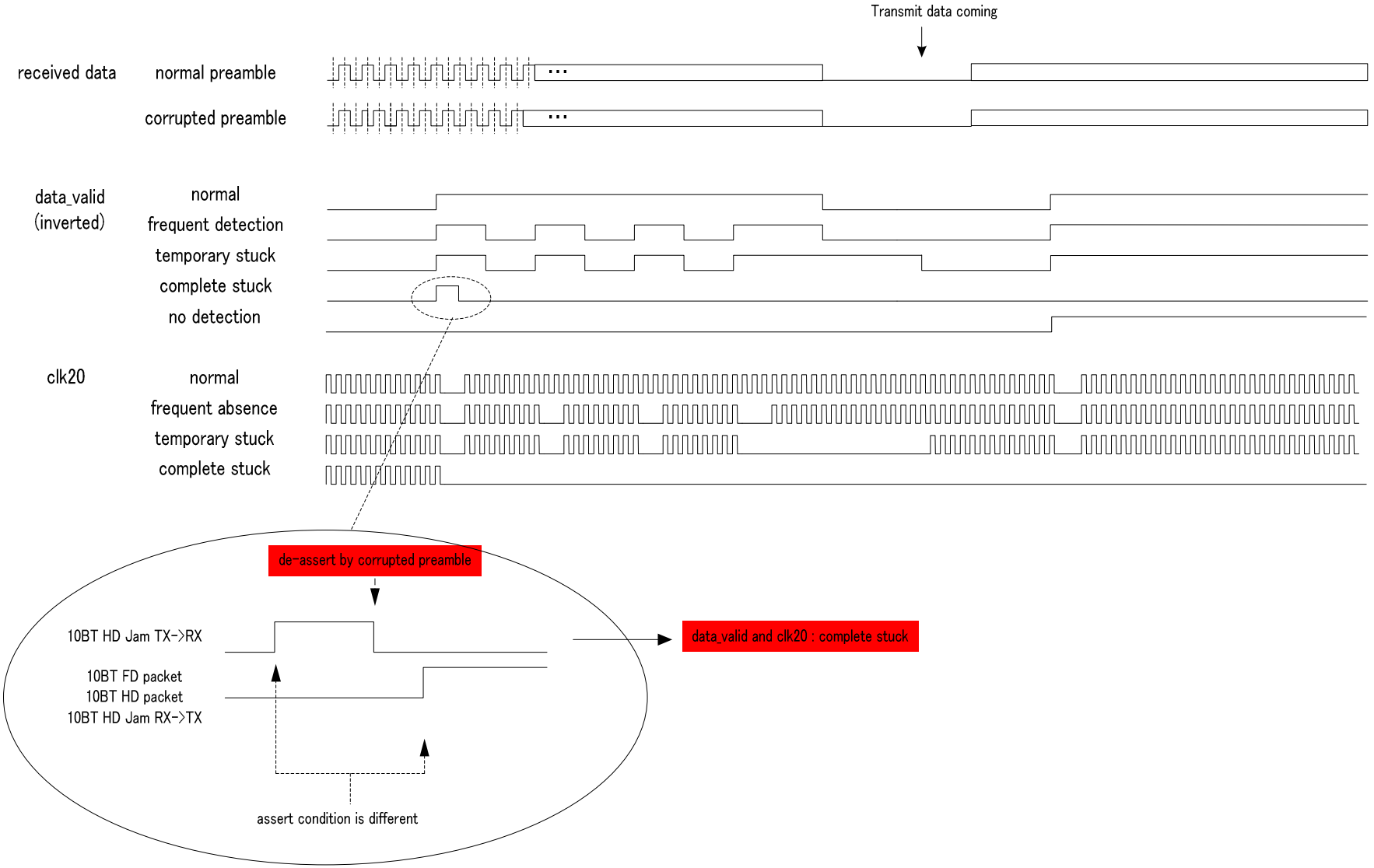
※1 There is no discarded packet when we observed in Labo. (163.8Mpackets)
The problem doesn't occur as long as one preamble is detected during all preamble pattern(56BT).

※2 Currently we found no equipment to generate corrupted preamble in 10BT FD mode.

severe level 1: problem cleared by only HW,S/W reset
severe level 2: problem cleared by next Tx data
severe level 3: only the problem data is discarded

© 2010 Renesas Electronics Europe. All rights reserved.

term and condition on previous page



© 2010 Renesas Electronics Europe. All rights reserved.

Renesas Electronics Europe

© 2010 Renesas Electronics Europe. All rights reserved.