The Ethernet Interface of the NetBurner MOD54415

References:

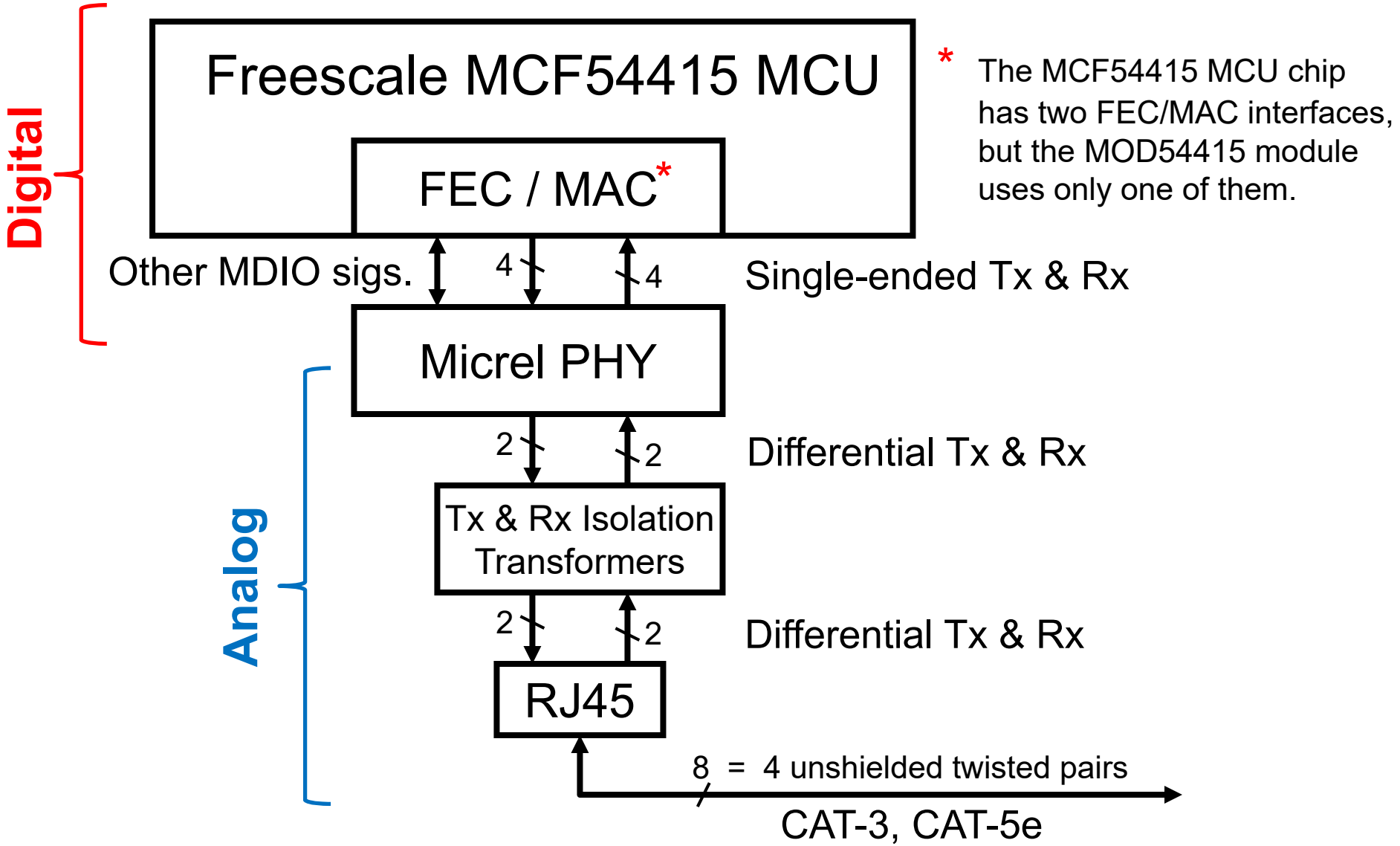
- Micrel Inc., KS8721B.BT Datasheet, Rev. 2.3, March 2006.
- Freescale Semiconductor, Inc., “ColdFire Family Programmer’s Reference Manual”, Doc. No. CFRM, Rev. 3, July 2005.
- Freescale Semiconductor, Inc., “MCF5441x Reference Manual”, Doc. No. MCF54418RM, Rev. 4, January 2012.

Figures and tables from the above documents have been included in these course notes for educational purposes in ECE 315 only. The original documentation should be consulted to ensure accuracy.

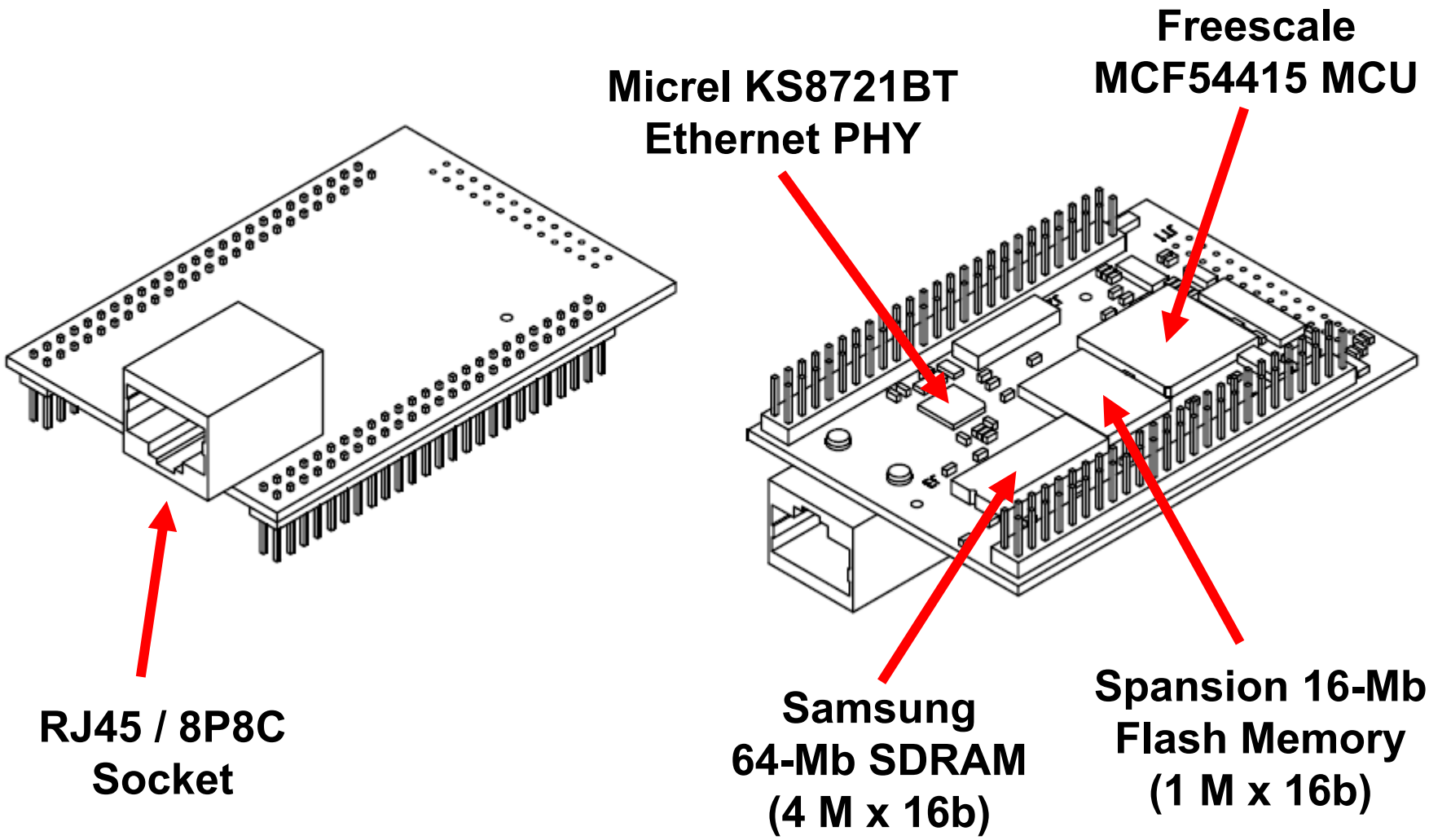
Ethernet Data Rates and Transmission Media

- To simplify the design and reduce the cost, Ethernet interface hardware is usually partitioned into two layers:
  - ***Media Access Control*** (MAC) layer, which implements the data link protocol. This is a purely digital function.
  - ***Physical Layer Transceiver*** (PHY), which provides a connection from the digital MAC to the analog communication medium (electrical, optical, or wireless).
- The MAC-PHY interface has been standardized under ***IEEE Std 802.3 MII Management Interface***, also called the Management Data Input / Output (MDIO) Interface.
- The MAC and PHY blocks can be designed by different teams or companies, and can be implemented in different integrated circuits (ICs) made with different processes.

MAC-PHY Architecture in the MOD54415

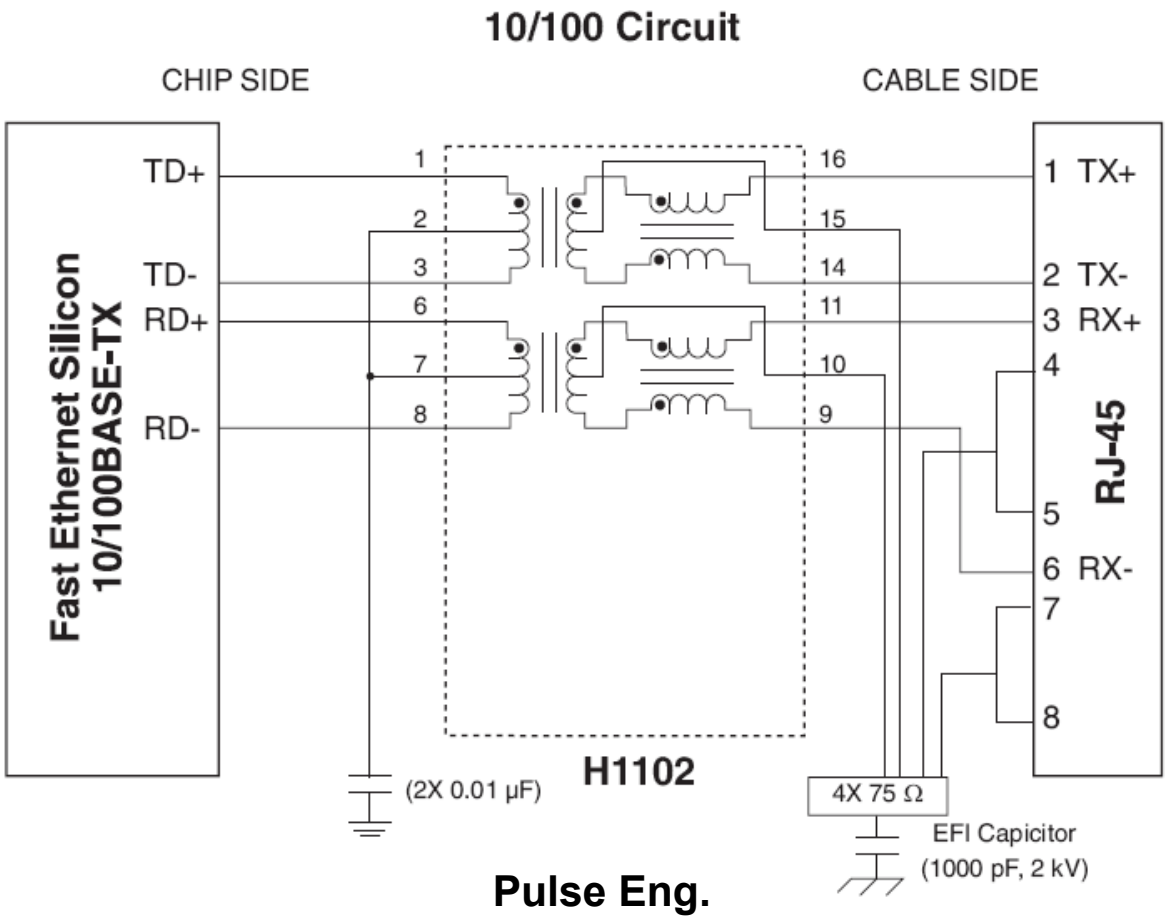


Physical Appearance of the NetBurner MOD54415



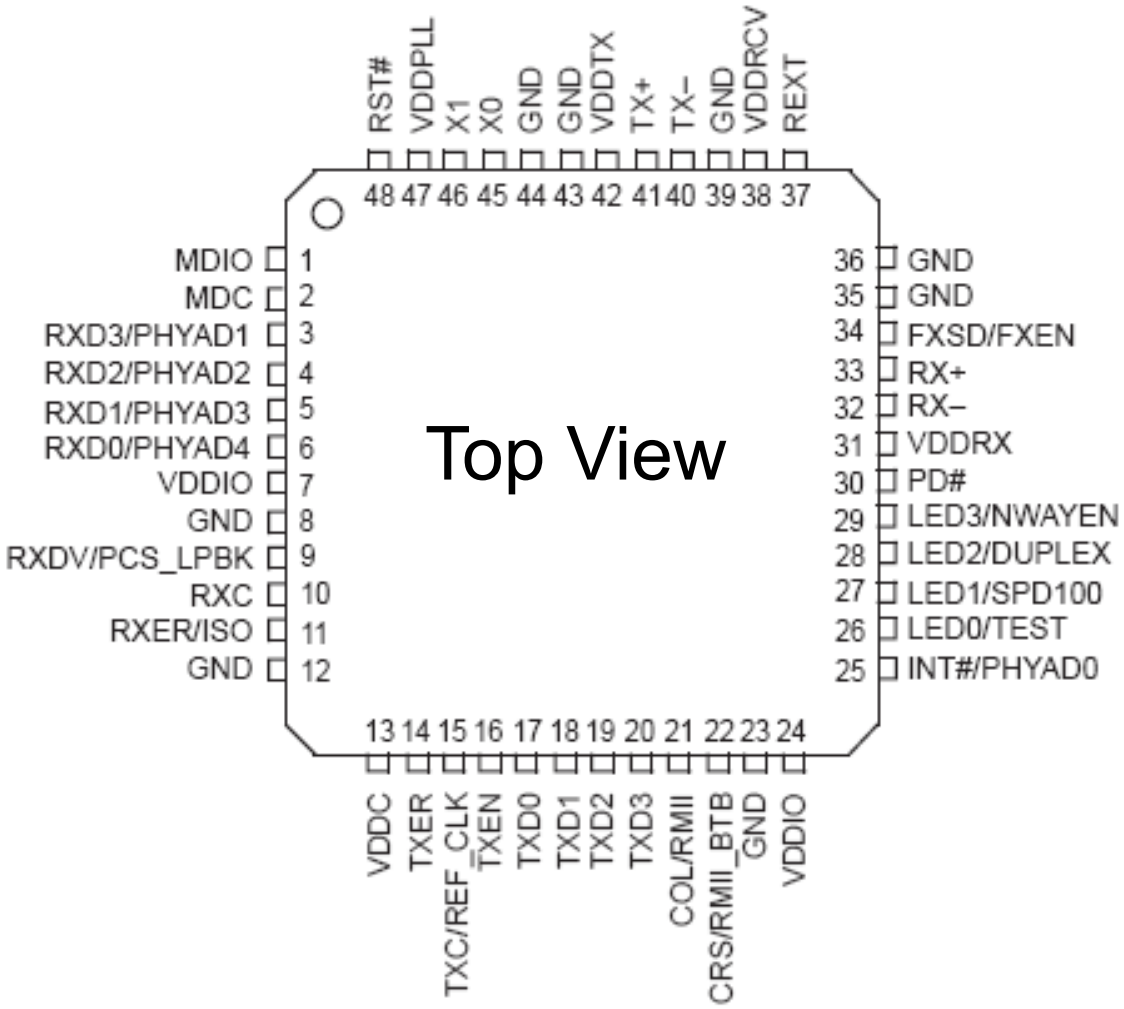
Isolation Transformer Connection Details

An isolation transformer is required by the Ethernet standard so that any offset between the ground potentials at the ends of the connection will have no effect.



Pulse Eng.

Package Outline of the Micrel KS8721BT PHY

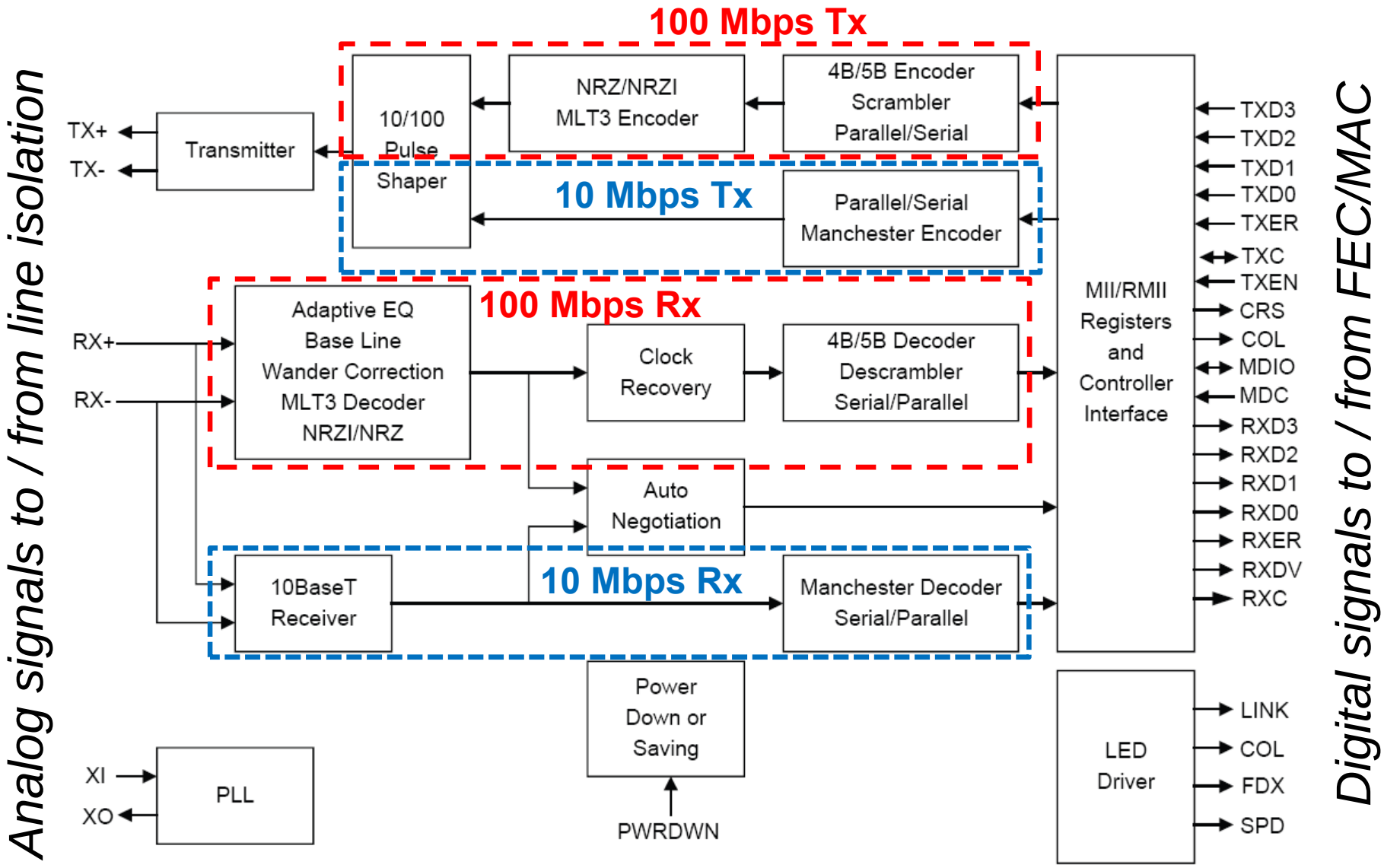


48-Pin TQFP (TQ) = Thin Quad Flat Package

Functional Overview of the Micrel KS8721BT PHY

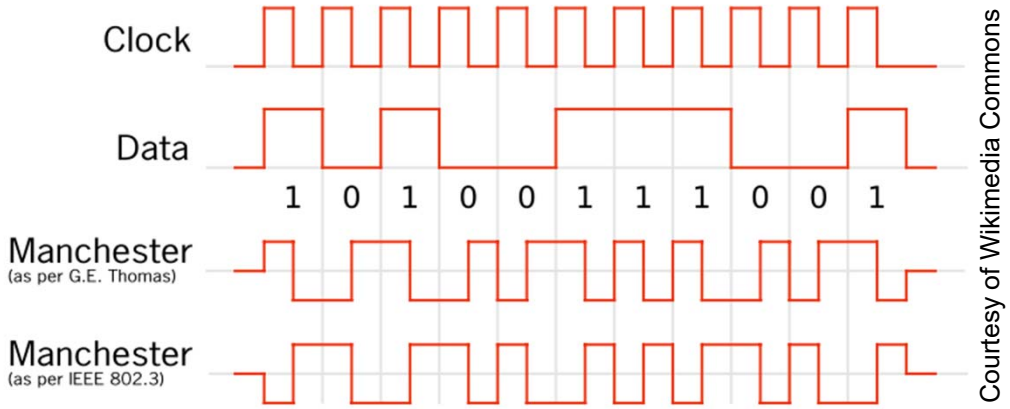
- Provides a single-chip PHY for both **10** and **100-Mbps Ethernet**:
  - 10BaseT (10 Mbps over CAT-3 cable)
  - 100BaseTX (100 Mbps over CAT-5e cable)
  - 100BaseFX (100 Mbps over optical fibre, with added transceiver)
- Compliant with the IEEE Std 802.3u MAC-PHY interface standard.
- Operation is configurable through a standard MII / MDIO serial management port as well as through external control pins. Twelve 16-bit registers are serially accessible through the MII interface.
- Provides automatic speed negotiation between the 10BaseT and 100BaseTX modes.
- Automatically negotiates full and half-duplex modes.
- Supports a power down mode and a power saving mode.
- Provides on-chip analog filtering to avoid the need for external filters. The connection to the cable medium can be made through a simple isolation transformer.

Functional Blocks inside the Micrel Ethernet PHY



The Manchester Code

- The Manchester Code provides two analog waveforms for encoding bits when they are stored or communicated on a physical medium. The code was developed in the late 1940s at the University of Manchester as part of the influential Mark 1 electronic computer project.
- A '0' is represented using a rising edge, while a '1' is represented using a falling edge. IEEE Std 802.3 reversed the waveform mapping to bits.



- Assuming that the bit sequence is reasonably random, the waveform will include a tone at the same frequency as the transmitter bit clock. This makes it easier for the receiver circuit to recover the bit clock.
- Also, the waveform for a bit sequence will not have a DC component, which allows the signal to be passed through capacitors & transformers.

The 4B5B Code

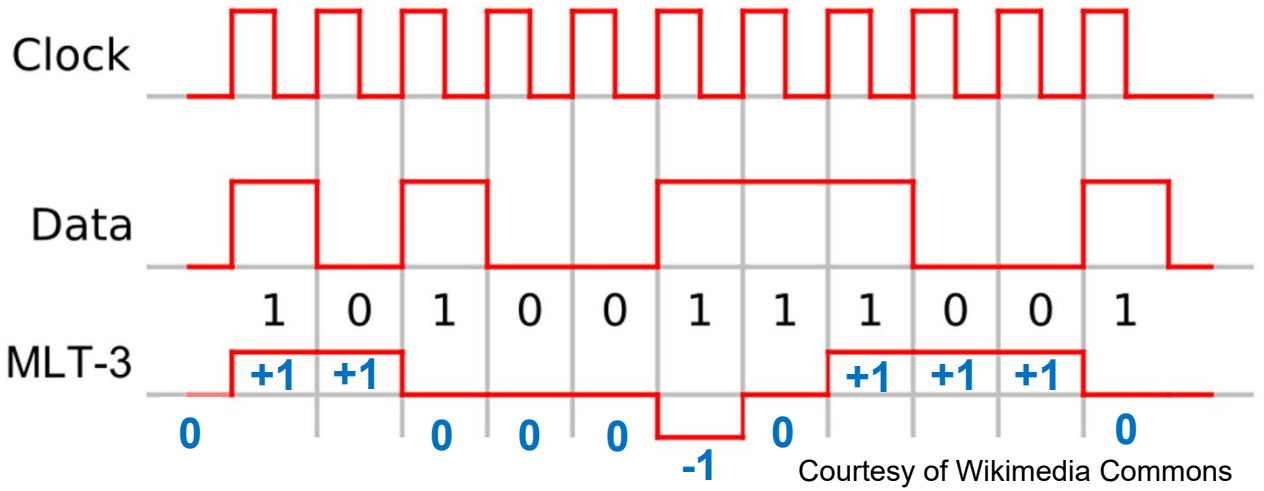
Data		4B5B code
(Hex)	(Binary)	
0	0000	11110
1	0001	01001
2	0010	10100
3	0011	10101
4	0100	01010
5	0101	01011
6	0110	01110
7	0111	01111
8	1000	10010
9	1001	10011
A	1010	10110
B	1011	10111
C	1100	11010
D	1101	11011
E	1110	11100
F	1111	11101

Courtesy of Wikimedia Commons

- The 4B5B line code partitions a given bit sequence into 4-bit groups and then replaces each group with a corresponding 5-bit codeword.
- The codewords in a 4B5B code are chosen to ensure that there is at least one 0-to-1 or 1-to-0 transition in every codeword. Thus, at the cost of increasing the bit rate by 25%, the resulting signal is guaranteed to contain a tone that is related to the original bit clock.
- The 4B5B code shown to the left was adopted in the 100-Mbit/s 100Base-TX Ethernet standard, as defined in IEEE Std 802.3u in 1995.
- Only 16 of the $2^5 = 32$ possible 5-bit codewords are used to encode data. The 16 remaining code-words can be used to encode control signals.

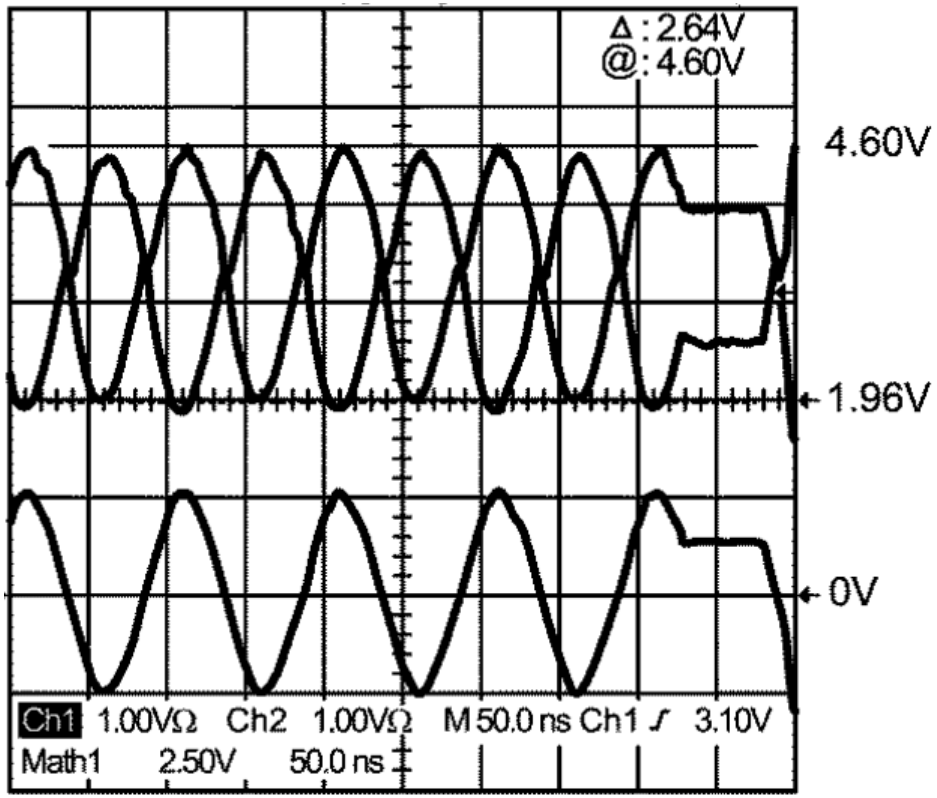
The MLT-3 Line Code

- The MLT-3 line code is a three-level code for signalling binary data on a communications medium. The three levels, which could be voltages, are represented symbolically as -1, 0 and +1.
- When the input bit is a ‘0’, the output signal level does not change.
- When the input bit is a ‘1’, output signal level changes to the next level in the following 4-step staircase sequence: 0, +1, 0, -1, 0, +1, 0, etc.
- An MLT-3 encoder should be preceded by a binary encoder or scrambler to ensure that long input strings containing only 0s do not occur.



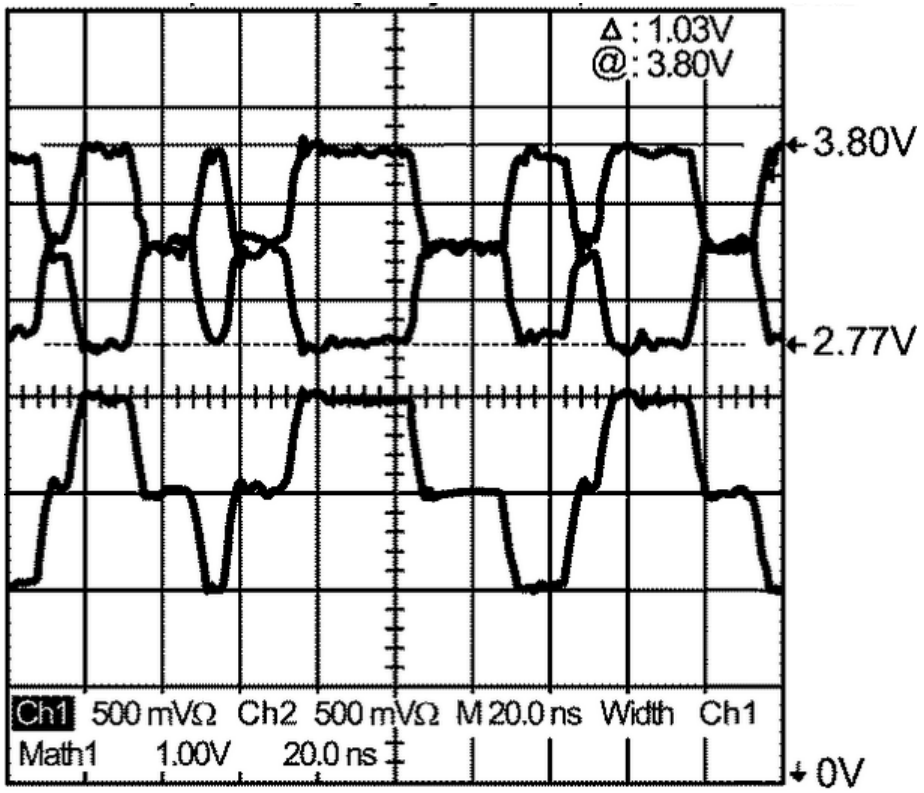
Courtesy of Wikimedia Commons

10BaseT and 100BaseTX Waveforms



10BaseT (10 Mbps)

Manchester-coded, 2-level



100BaseTX (100 Mbps)

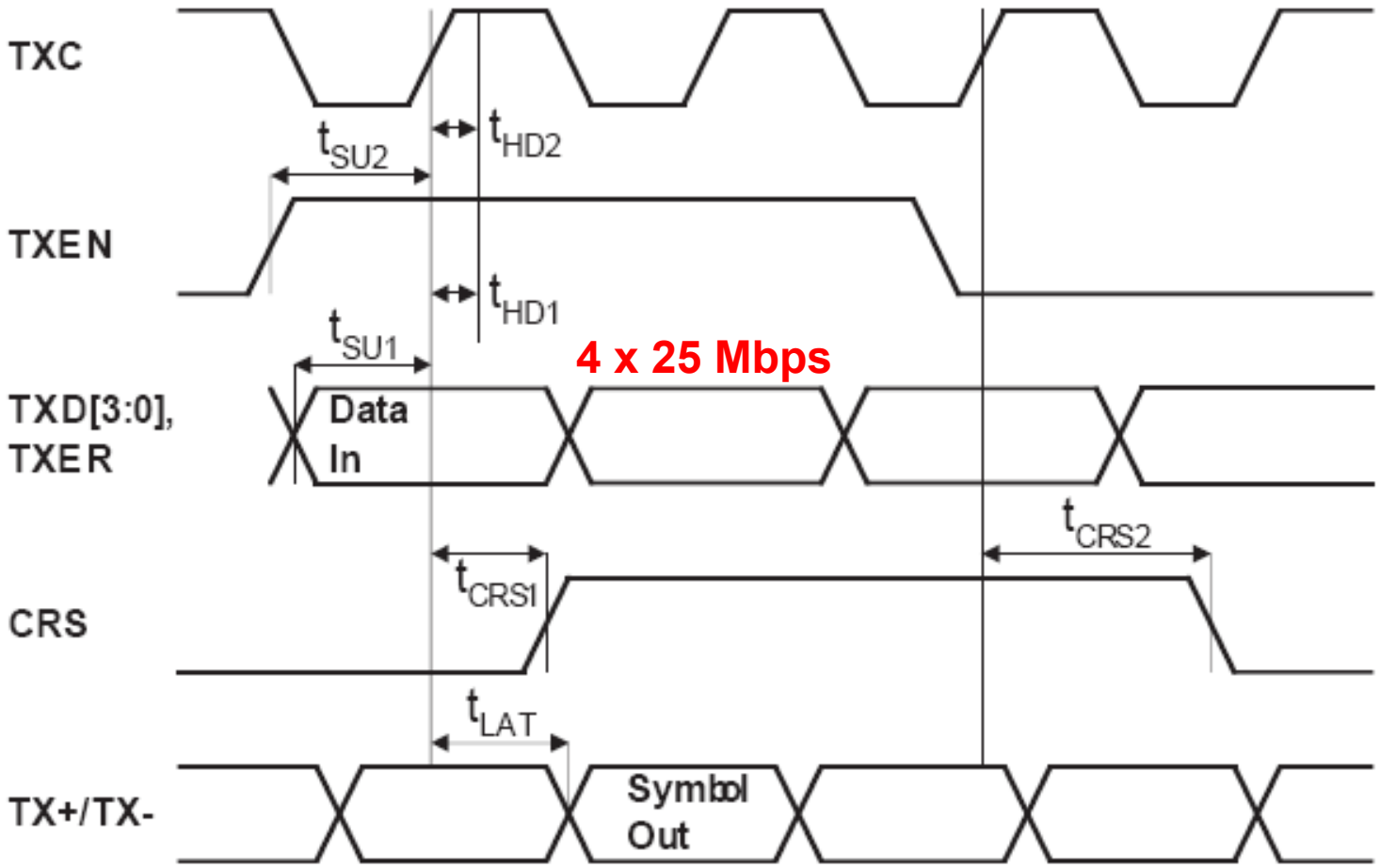
4B/5B-coded, 3-level (MLT-3)

From the Texas Instruments datasheet for DP83848C/I/VYB/YB PHYTER™ QFP Single Port 10/100 Mb/s Ethernet Physical Layer Transceiver, SNLS266E, May 2007.

PHY 100-Mbps Transmit Signals

TXC	MII Transmit Clock (input to PHY from MAC) Derived from 25-MHz clock input at X1 pin.
TXEN	MII Transmit Enable (input)
TXD[3:0]	MII Transmit Data (parallel inputs) at 25 MHz. Raw serial data rate is thus 100 Mbps.
TXER	MII Transmit Error (input)
CRS	MII Carrier Sense (output from PHY to MAC)
TX+/TX-	Differential shaped analog output signals (output from PHY to isolation transformer). Analog bit signals (called “symbols”) transmitted at 125 MHz because of the 4B/5B line code. A 3-level MLT3 waveform is used.

PHY 100-Mbps Transmit Signal Waveforms



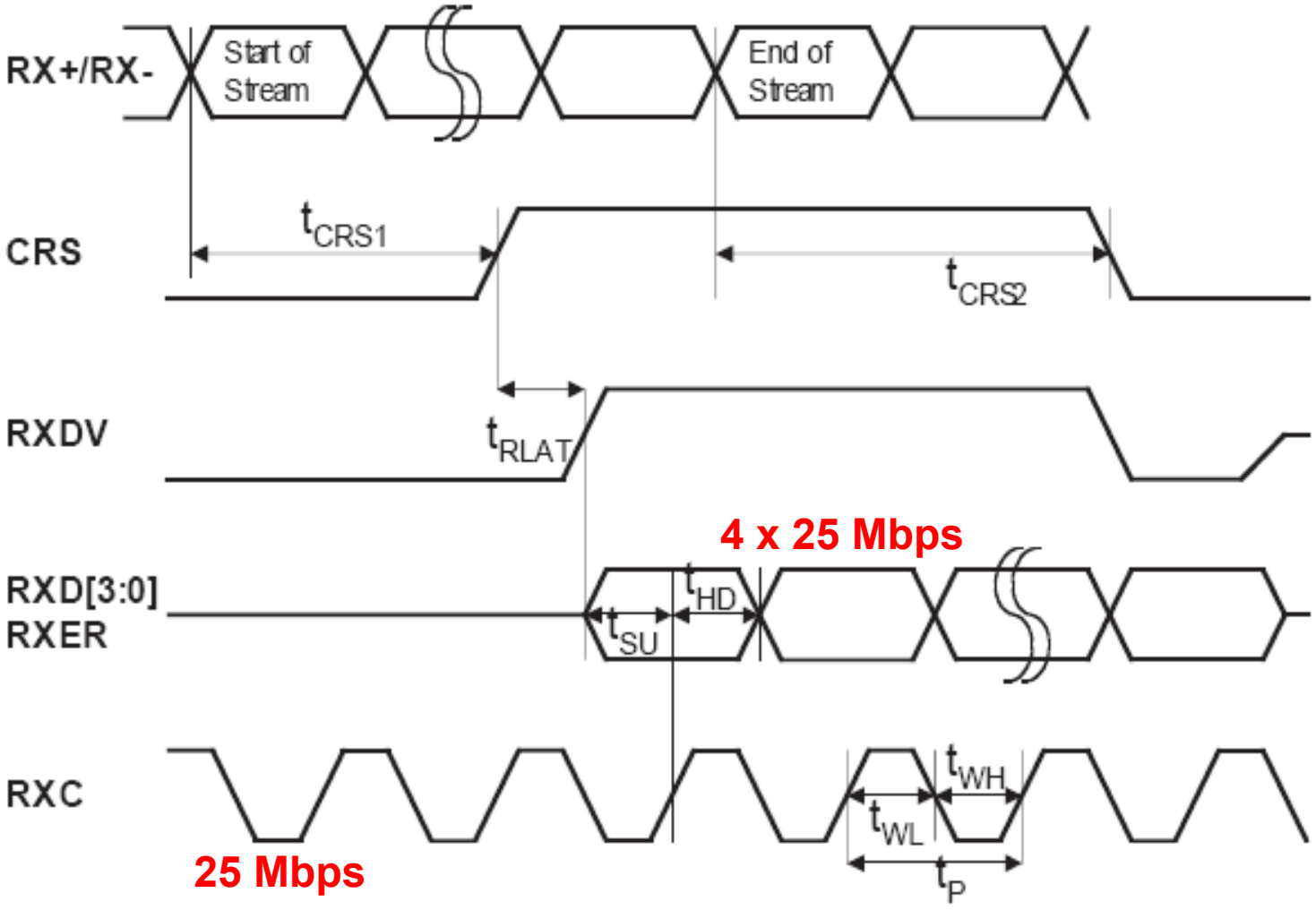
25 million 5-bit 4B/5B MLT-3-encoded symbols per second

PHY 100-Mbps Receive Signals

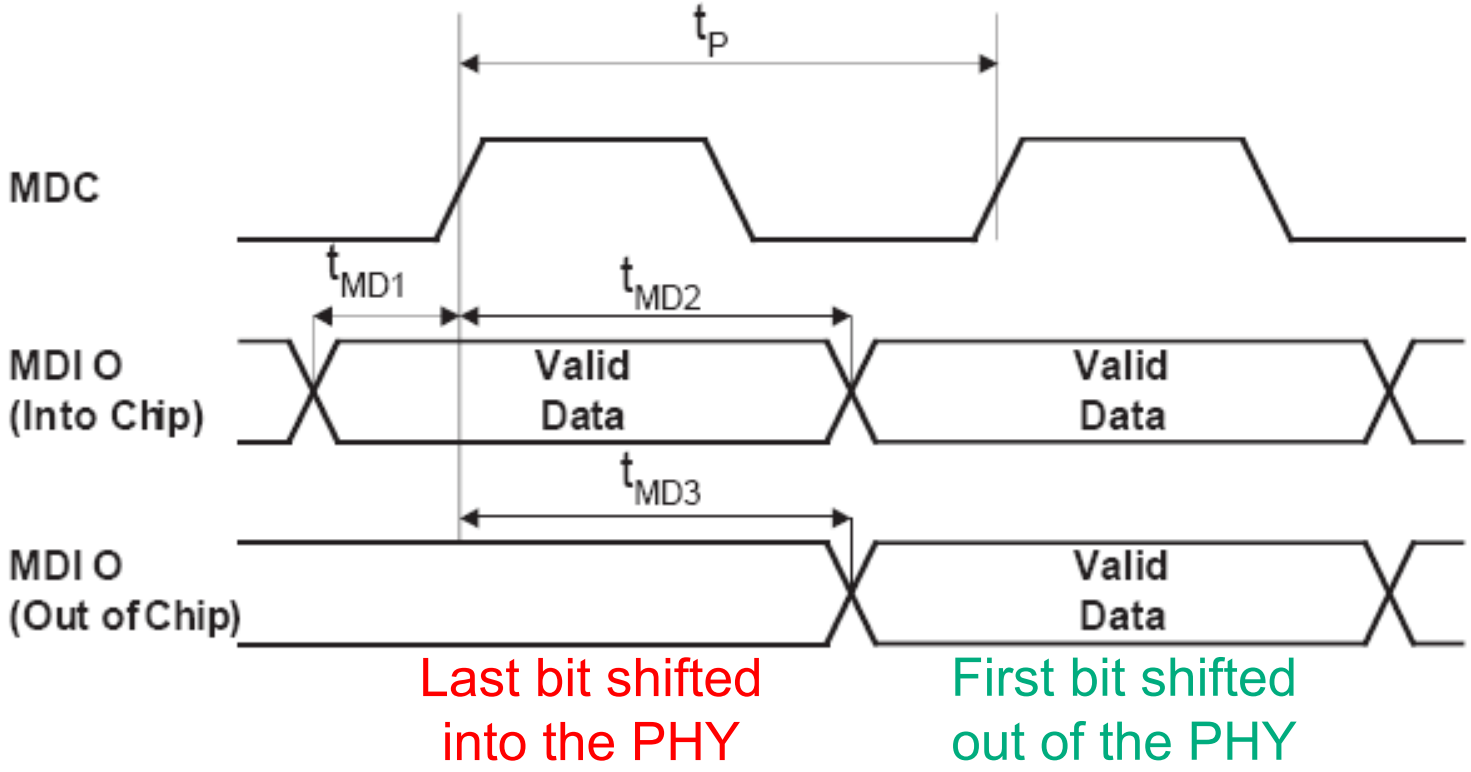
RX+/RX-	Distorted received differential analog signals (signals from isolation transformer to PHY)
CRS	MII Carrier Sense (output from PHY to MAC)
RXDV	MII Receive Data Valid (output to MAC)
RXD[3:0]	MII Receive Data (output to MAC)
RXER	MII Receiver Error (output to MAC)
RXC	MII Receive Clock (output to MAC) 100-Mbps => 25-MHz recovered clock 10-Mbps => 2.5-MHz recovered clock

PHY 100-Mbps Receive Signal Waveforms

25 million 5-bit 4B/5B MLT-3-encoded symbols per second



PHY MII Serial Management Waveforms



Shift-in phase: The MDC clock is used to shift in one 32-bit control frame, via the MDIO pin used as an input. 16 bits are loaded into one of twelve 16-bit PHY registers. An addressing scheme selects the desired register. ***Shift-out phase***: Contents of addressed register are shifted out through the MDIO pin, used as an output from the PHY.

Serially-Accessible 16-bit MDIO Registers

Reg. No.	Register Name
0x00	Basic Control Register
0x01	Basic Status Register
0x02	PHY Identifier I
0x03	PHY Identifier II
0x04	Auto-Negotiation Advertisement Register
0x05	Auto-Negotiation Link Partner Ability Register
0x06	Auto-Negotiation Expansion Register
0x07	Auto-Negotiation Next Page Register
0x08	Link Partner Next Page Ability
0x15	RXER Counter Register
0x1b	Interrupt Control/Status Register
0x1f	100BaseTX PHY Control Register

Ex: “Basic Control Register” in the PHY

Bit 15: 1 => reset the PHY (self-clearing bit)

Bit 14: 1 => enable loopback mode

Bit 13: 1 => 100 Mbps mode; 0 => 10 Mbps mode

Bit 12: 1 => enable auto-negotiation mode

Bit 11: 1 => enable power down mode

Bit 10: 1 => isolate the PHY from the MII data and TX+/TX-

Bit 9: 1 => restart auto-negotiation process (self-clearing bit)

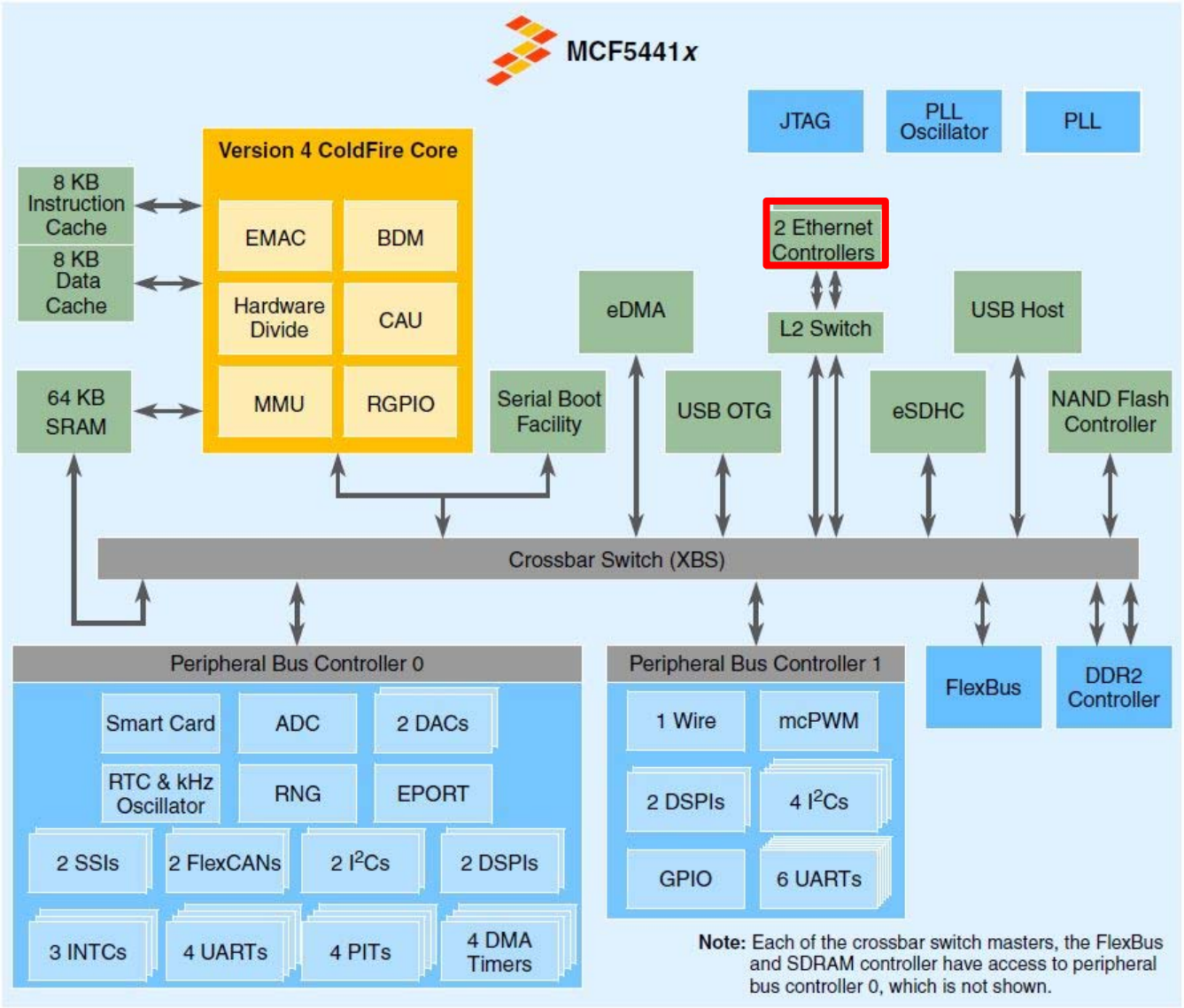
Bit 8: 1 => full duplex mode; 0 => half duplex mode

Bit 7: 1 => enable collision test

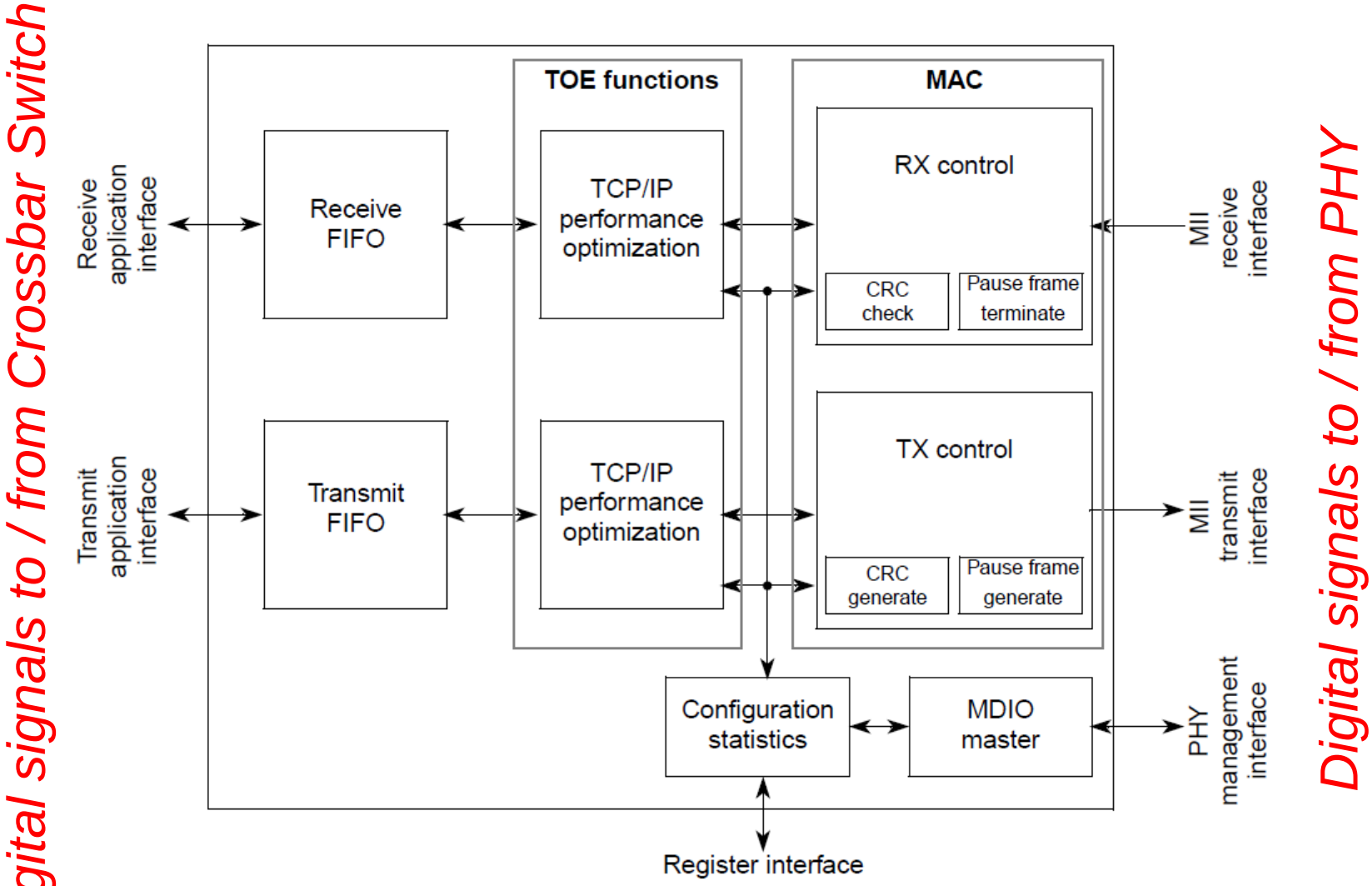
Bits 7-1: Reserved bits (for proprietary test modes?)

Bit 0: 1 => disable transmitter; 0 => enable transmitter

Architecture of the Freescale MCF54415 µC



MAC-NET Ethernet Interface in the MCF54415 MCU

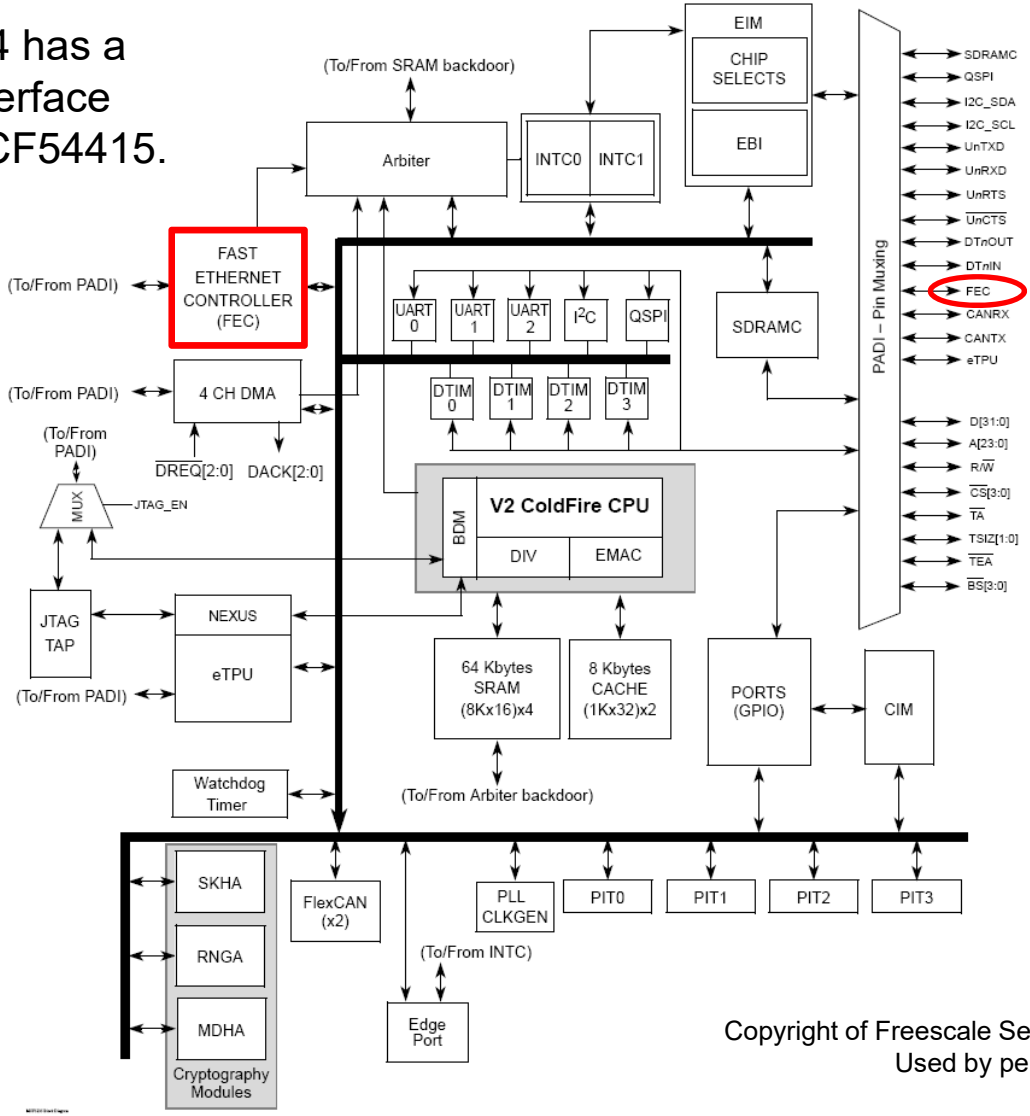


MII and RMII Signals at the MAC-PHY Interface

Signal		Description
MII	RMII	
MII_COL	—	Asserted upon detection of a collision and remains asserted while the collision persists. This signal is not defined for full-duplex mode.
MII_CRS	—	Carrier sense. When asserted, indicates transmit or receive medium is not idle. In RMII mode, this signal is present on the RMII_CRS_DV pin.
MII_MDC	RMII_MDC	Output clock provides a timing reference to the PHY for data transfers on the MDIO signal.
MII_MDIO	RMII_MDIO	Transfers control information between the external PHY and the media-access controller. Data is synchronous to MDC. This signal is an input after reset.
MII_RXCLK	—	Provides a timing reference for RXDV, RXD[3:0], and RXER.
MII_RXDV	RMII_CRS_DV	Asserting this input indicates the PHY has valid nibbles present on the MII. RXDV must remain asserted from the first recovered nibble of the frame through to the last nibble. Asserting RXDV must start no later than the SFD and exclude any EOF. In RMII mode, this pin also generates the CRS signal.
MII_RXD0	RMII_RXD0	Contains the Ethernet input data transferred from PHY to the media-access controller when RXDV is asserted.
MII_RXD1	RMII_RXD1	
MII_RXD[3:2]	—	
MII_RXER	RMII_RXER	When asserted with RXDV, indicates the PHY detects an error in the current frame. When RXDV is negated, RXER has no effect.
MII_TXCLK	—	Input clock which provides a timing reference for TXEN, TXD[3:0], and TXER.
MII_TXD0	RMII_TXD0	The serial output Ethernet data and only valid during the assertion of TXEN.
MII_TXD1	RMII_TXD1	
MII_TXD[3:2]	—	
MII_TXEN	RMII_TXEN	Indicates when valid nibbles are present on the MII. This signal is asserted with the first nibble of a preamble and is negated before the first TXCLK following the final nibble of the frame.
MII_TXER	—	When asserted for one or more clock cycles while TXEN is also asserted, PHY sends one or more illegal symbols. TXER has no effect at 10 Mbps or when TXEN is negated.
—	RMII_REF_CLK	In RMII mode, this signal is the reference clock for receive, transmit, and the control interface.

Architecture of the Freescale MCF5234 MCU

Note: The MCF5234 has a simpler Ethernet interface compared to the MCF54415.



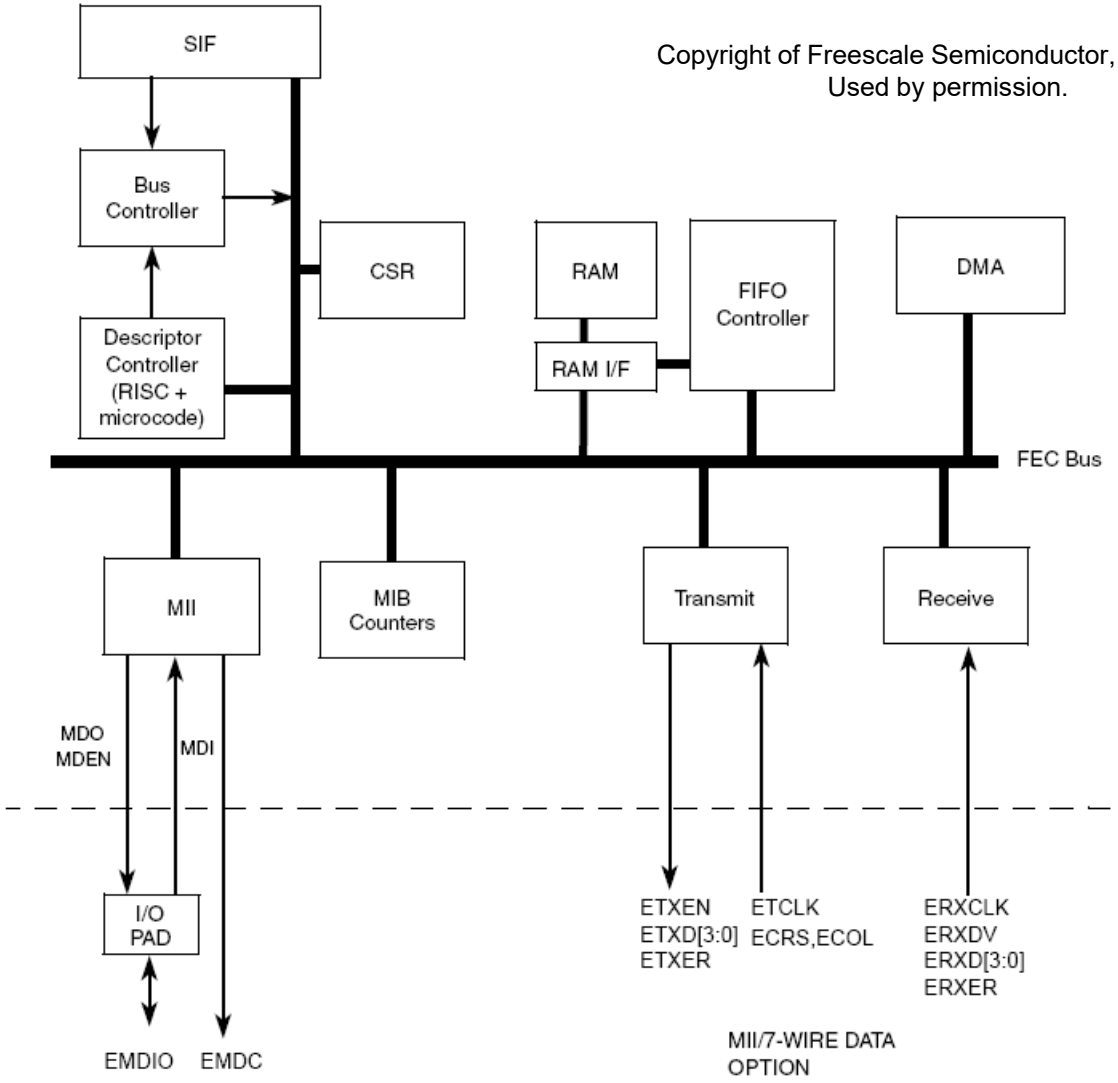
18 FEC pins

FEC = Fast Ethernet (100Base-TX) Controller

Functional Overview of the FEC

- Provides most ***Media Access Control*** (MAC) functionality
- Supports three kinds of MAC-PHY interfaces:
  - 10Base-T under the full MII standard
  - 10Base-T under a simplified 7-wire RM standard
  - 100Base-TX under the full MII standard
- Data transmission can be full-duplex or half-duplex
- Off-loads most packet transmission chores from the CPU
- Memory-mapped registers in the CPU's memory map
- On-chip transmit and receiver first-in, first-out (FIFO) buffers
- Supports flexible memory-based data structures

Fast Ethernet Controller (FEC) Architecture



Functional Overview of the FEC (cont'd)

- FEC sub-systems communicate with each other over a local ***FEC bus***.
- ***Descriptor Controller*** is a special-purpose FEC controller:
  - Microcode program is fixed during chip manufacture.
  - Initializes private registers (independent of the CPU).
  - Interprets Rx and Tx ***data buffer descriptors***.
  - Processes addresses in received data frames.
  - Generates random numbers for ***collision back-off timer***.
  - Starts autonomous ***Direct Memory Access*** (DMA) operations to transfer Rx and Tx data at high speed to and from other locations in the CPU's memory map.

FEC Memory Map (viewed from the ColdFire CPU)

Address	Function
IPSBAR + 0x1000–11FF	Control/Status Registers
IPSBAR + 0x1200–13FF	MIB Block Counters

- FEC registers are mapped to 1024 contiguous bytes of memory, starting at the address loaded in the ***Internal Peripheral System Base Address Register*** (IPSBAR) in the System Control Module (SCM) + 0x1000. The default IPSBAR contents are 0x4000_0000.
- Twenty-three 32-bit ***Control/Status Registers*** are used by the CPU to configure the FEC and to monitor FEC status.
- Data is exchanged via data structures, called ***buffer descriptors***, in a buffer region in the CPU's data RAM space.
- ***MIB Block Counters*** automatically count the occurrences of events of likely interest.

FEC Control & Status Registers

IPSBAR Offset	Name	Description	Size (bits)
0x1004	EIR	Interrupt Event Register	32
0x1008	EIMR	Interrupt Mask Register	32
0x1010	RDAR	Receive Descriptor Active Register	32
0x1014	TDAR	Transmit Descriptor Active Register	32
0x1024	ECR	Ethernet Control Register	32
0x1040	MDATA	MII Data Register	32
0x1044	MSCR	MII Speed Control Register	32
0x1064	MIBC	MIB Control/Status Register	32
0x1084	RCR	Receive Control Register	32
0x10C4	TCR	Transmit Control Register	32
0x10E4	PALR	Physical Address Low Register	32
0x10E8	PAUR	Physical Address High+ Type Field	32
0x10EC	OPD	Opcode + Pause Duration	32
0x1118	IAUR	Upper 32 bits of Individual Hash Table	32
0x111C	IALR	Lower 32 Bits of Individual Hash Table	32
0x1120	GAUR	Upper 32 bits of Group Hash Table	32
0x1124	GALR	Lower 32 bits of Group Hash Table	32
0x1144	TFWR	Transmit FIFO Watermark	32
0x114C	FRBR	FIFO Receive Bound Register	32
0x1150	FRSR	FIFO Receive FIFO Start Registers	32
0x1180	ERDSR	Pointer to Receive Descriptor Ring	32
0x1184	ETDSR	Pointer to Transmit Descriptor Ring	32
0x1188	EMRBR	Maximum Receive Buffer Size	32

FEC Ethernet Control Register (ECR)

Only the two least significant bits are used:

- Bit #0 controls an ***FEC-level reset***
- Bit #1 controls an ***FEC-level enable***

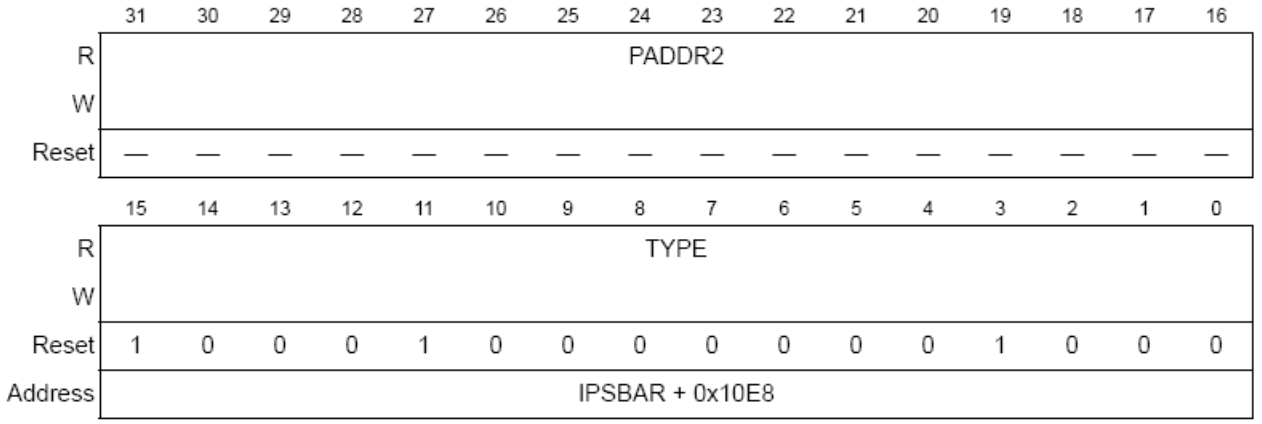
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W																
Reset	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	ETHER_EN	RESET
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Address	IPSBAR + 0x1024															

FEC ECR Bit Definitions

Bits	Name	Description
1	ETHER_EN	When this bit is set, the FEC is enabled, and reception and transmission are possible. When this bit is cleared, reception is immediately stopped and transmission is stopped after a bad CRC is appended to any currently transmitted frame. The buffer descriptor(s) for an aborted transmit frame are not updated after clearing this bit. When ETHER_EN is deasserted, the DMA, buffer descriptor, and FIFO control logic are reset, including the buffer descriptor and FIFO pointers. The ETHER_EN bit is altered by hardware under the following conditions: • ECR[RESET] is set by software, in which case ETHER_EN will be cleared • An error condition causes the EIR[EBERR] bit to set, in which case ETHER_EN will be cleared
0	RESET	When this bit is set, the equivalent of a hardware reset is performed but it is local to the FEC. ETHER_EN is cleared and all other FEC registers take their reset values. Also, any transmission/reception currently in progress is abruptly aborted. This bit is automatically cleared by hardware during the reset sequence. The reset sequence takes approximately 8 system clock cycles after RESET is written with a 1.

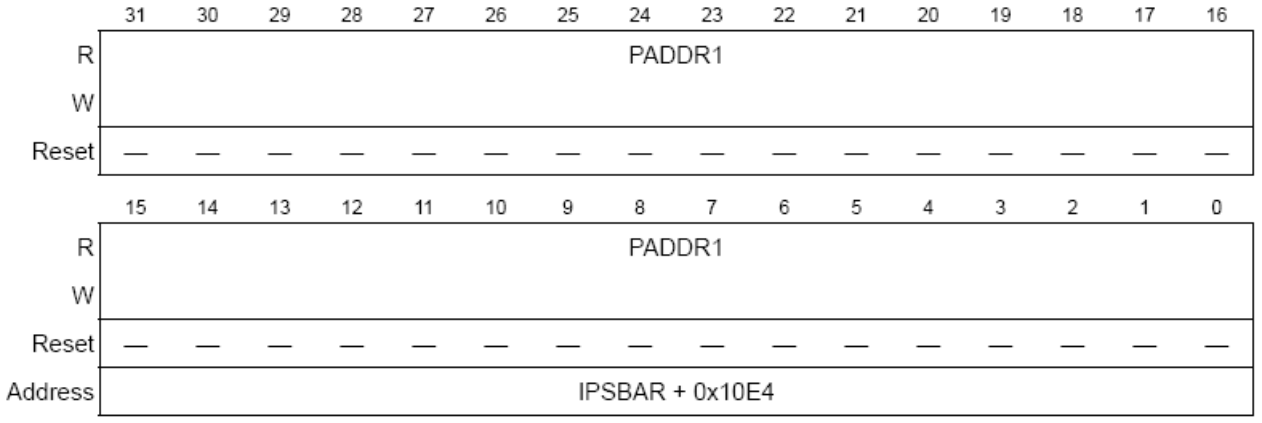
FEC Physical Address Upper Register (PAUR)



	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PADDR2															
Reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TYPE															
Reset	1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0
Address	IPSBAR + 0x10E8															

FEC Physical Address Lower Register (PALR)



	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PADDR1															
Reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PADDR1															
Reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Address	IPSBAR + 0x10E4															

FEC Transmit Control Register (TCR)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	0	0	0	RFC_ PAUSE	TFC_ PAUSE	FEDN	HBC	GTS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Address	IPSBAR + 0x10C4															

FEC TCR Bit Definitions

Bits	Name	Description
31–5	—	Reserved, should be cleared.
4	RFC_PAUSE	Receive frame control pause. This read-only status bit will be asserted when a full duplex flow control pause frame has been received and the transmitter is paused for the duration defined in this pause frame. This bit will automatically clear when the pause duration is complete.
3	TFC_PAUSE	Transmit frame control pause. Transmits a PAUSE frame when asserted. When this bit is set, the MAC will stop transmission of data frames after the current transmission is complete. At this time, the GRA interrupt in the EIR register will be asserted. With transmission of data frames stopped, the MAC will transmit a MAC Control PAUSE frame. Next, the MAC will clear the TFC_PAUSE bit and resume transmitting data frames. Note that if the transmitter is paused due to user assertion of GTS or reception of a PAUSE frame, the MAC may still transmit a MAC Control PAUSE frame.
2	FDEN	Full duplex enable. If set, frames are transmitted independent of carrier sense and collision inputs. This bit should only be modified when ETHER_EN is deasserted.
1	HBC	Heartbeat control. If set, the heartbeat check is performed following end of transmission and the HB bit in the status register will be set if the collision input does not assert within the heartbeat window. This bit should only be modified when ETHER_EN is deasserted.
0	GTS	Graceful transmit stop. When this bit is set, the MAC will stop transmission after any frame that is currently being transmitted is complete and the GRA interrupt in the EIR register will be asserted. If frame transmission is not currently underway, the GRA interrupt will be asserted immediately. Once transmission has completed, a "restart" can be accomplished by clearing the GTS bit. The next frame in the transmit FIFO will then be transmitted. If an early collision occurs during transmission when GTS = 1, transmission will stop after the collision. The frame will be transmitted again once GTS is cleared. Note that there may be old frames in the transmit FIFO that will be transmitted when GTS is reasserted. To avoid this deassert ECR[ETHER_EN] following the GRA interrupt.

FEC Receive Control Register (RCR)

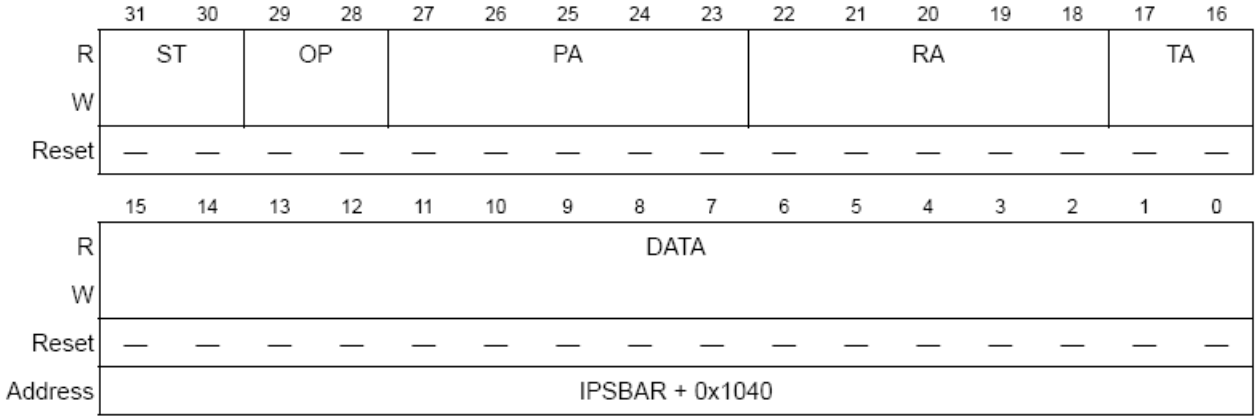
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0	0	0	0	MAX_FL										
W																
Reset	0	0	0	0	0	1	0	1	1	1	1	0	1	1	1	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	0	0	FCE	BC_ REJ	PROM	MII_ MODE	DRT	LOOP
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
Address	IPSBAR + 0x1084															

FEC RCR Bit Definitions

Bits	Name	Description
31–27	—	Reserved, should be cleared.
26–16	MAX_FL	Maximum frame length. Resets to decimal 1518. Length is measured starting at DA and includes the CRC at the end of the frame. Transmit frames longer than MAX_FL will cause the BABT interrupt to occur. Receive Frames longer than MAX_FL will cause the BABR interrupt to occur and will set the LG bit in the end of frame receive buffer descriptor. The recommended default value to be programmed by the user is 1518 or 1522 (if VLAN Tags are supported).
15–6	—	Reserved, should be cleared.
5	FCE	Flow control enable. If asserted, the receiver will detect PAUSE frames. Upon PAUSE frame detection, the transmitter will stop transmitting data frames for a given duration.
4	BC_REJ	Broadcast frame reject. If asserted, frames with DA (destination address) = FF_FF_FF_FF_FF_FF will be rejected unless the PROM bit is set. If both BC_REJ and PROM = 1, then frames with broadcast DA will be accepted and the M (MISS) bit will be set in the receive buffer descriptor.
3	PROM	Promiscuous mode. All frames are accepted regardless of address matching.
2	MII_MODE	Media independent interface mode. Selects external interface mode. Setting this bit to one selects MII mode, setting this bit equal to zero selects 7-wire mode (used only for serial 10 Mbps). This bit controls the interface mode for both transmit and receive blocks.
1	DRT	Disable receive on transmit. 0 Receive path operates independently of transmit (use for full duplex or to monitor transmit activity in half duplex mode). 1 Disable reception of frames while transmitting (normally used for half duplex mode).
0	LOOP	Internal loopback. If set, transmitted frames are looped back internal to the device and the transmit output signals are not asserted. The system clock is substituted for the ETXCLK when LOOP is asserted. DRT must be set to zero when asserting LOOP.

MII Management Frame Register (MMFR)



Bit	Name	Description
31–30	ST	Start of frame delimiter. These bits must be programmed to 01 for a valid MII management frame.
29–28	OP	Operation code. This field must be programmed to 10 (read) or 01 (write) to generate a valid MII management frame. A value of 11 will produce “read” frame operation while a value of 00 will produce “write” frame operation, but these frames will not be MII compliant.
27–23	PA	PHY address. This field specifies one of up to 32 attached PHY devices.
22–18	RA	Register address. This field specifies one of up to 32 registers within the specified PHY device.
17–16	TA	Turn around. This field must be programmed to 10 to generate a valid MII management frame.
15–0	DATA	Management frame data. This is the field for data to be written to or read from the PHY register.

Interrupts Produced by the FEC

- The FEC hardware can be configured to produce hardware interrupts when certain conditions become true.
- Thirteen different possible interrupting conditions have been defined.
- When a condition becomes true, the corresponding bit in the ***Ethernet Interrupt Event Register*** (EIR) is latched to 1 at the next rising clock edge.
- The single hardware interrupt signal from the FEC will be asserted if both a bit in the EIR and the corresponding bit in the ***Ethernet Interrupt Mask Register*** (EIMR) are 1.
- A bit in the EIR is cleared upon hardware reset, and by being written by CPU software to a 1. *Note*: Writing an EIR bit to 0 has no effect.

FEC Ethernet Interrupt Event Register (EIR)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	HB ERR	BABR	BABT	GRA	TXF	TXB	RXF	RXB	MII	EB ERR	LC	RL	UN	0	0	0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Address	IPSBAR + 0x1004															

FEC Ethernet Interrupt Mask Register (EIMR)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	HB ERR	BABR	BABT	GRA	TXF	TXB	RXF	RXB	MII	EB ERR	LC	RL	UN	0	0	0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Address	IPSBAR + 0x1008															

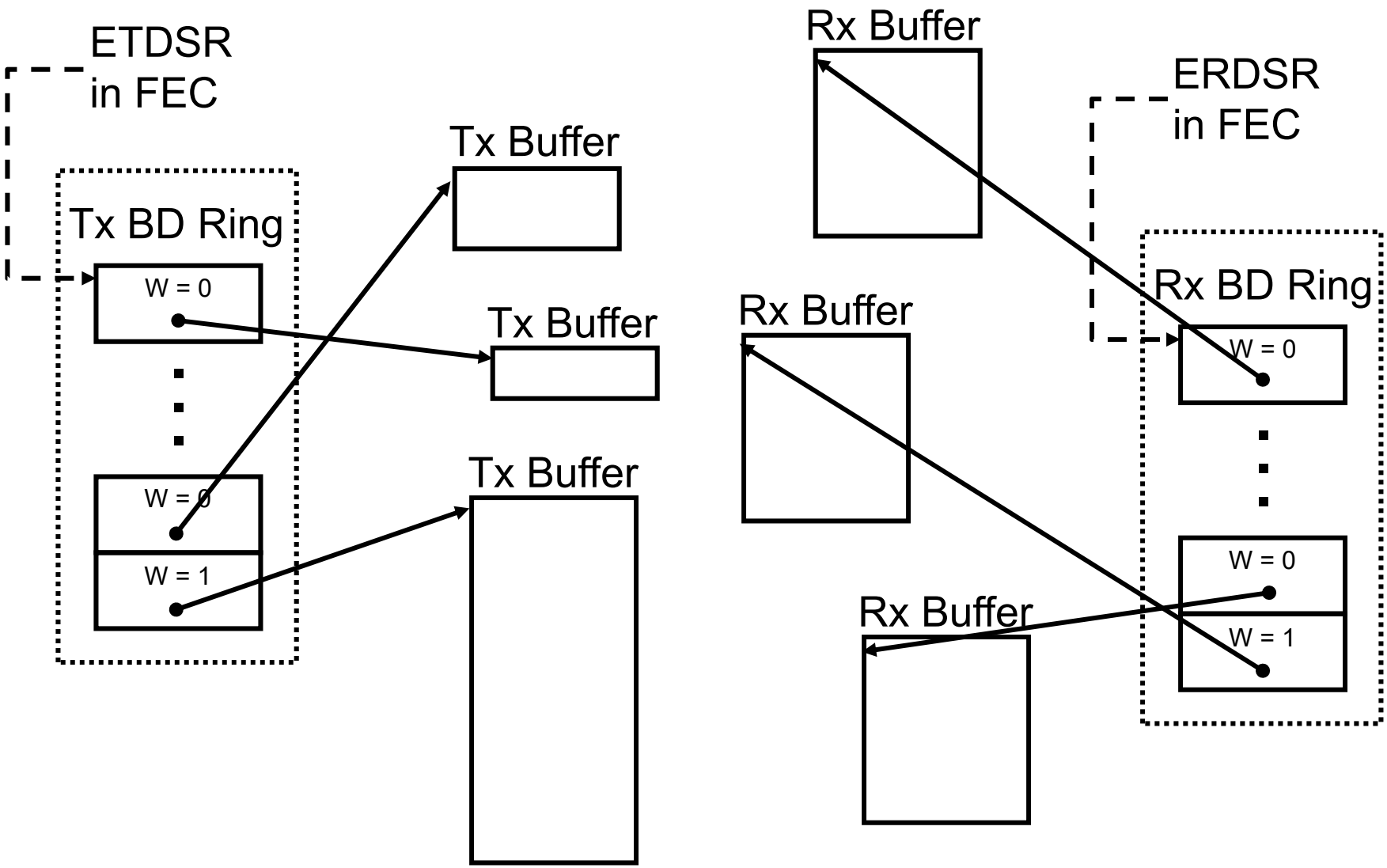
Frequently Used Bits in the EIR and EIMR

Bit No.	Name	Description
27	TXF	***Transmit Frame Interrupt***: The last Tx buffer in the Tx frame has been transmitted.
26	TXB	***Transmit Buffer Interrupt***: A Tx Buffer (not the last in the frame) has been transmitted.
25	RXF	***Receive Frame Interrupt***: The last Rx buffer for the Rx frame has been received.
24	RXB	***Receive Buffer Interrupt***: An Rx buffer (not the last) has been received.
23	MII	***MII Interrupt***: MII data transfer completed.
19	UN	***Transmit FIFO Underrun***: The Tx FIFO emptied out before the frame was fully transmitted.

FEC Transmit and Receive Buffer Overview

- Data is processed by the application(s) in blocks called ***payloads***.
- In addition to the application data, address information must be associated with the payload (e.g., TCP, IP and Ethernet ***headers*** and an Ethernet Cyclic Redundancy Check (CRC) ***trailer***).
- The entire data sequence (headers + payload) is called a ***frame***.
- The frame is typically partitioned and stored in multiple data structures called ***buffers***. Partitioning into buffers makes it easier for different software layers to access relevant data in the frame.
- The locations of the buffers in memory are controlled using data structures called ***buffer descriptors*** (BDs).
- Each buffer descriptor contains a ***pointer*** to one buffer.
- The transmit and receive directions have ***independent buffer systems***, with separate pools of buffer descriptors and buffers.

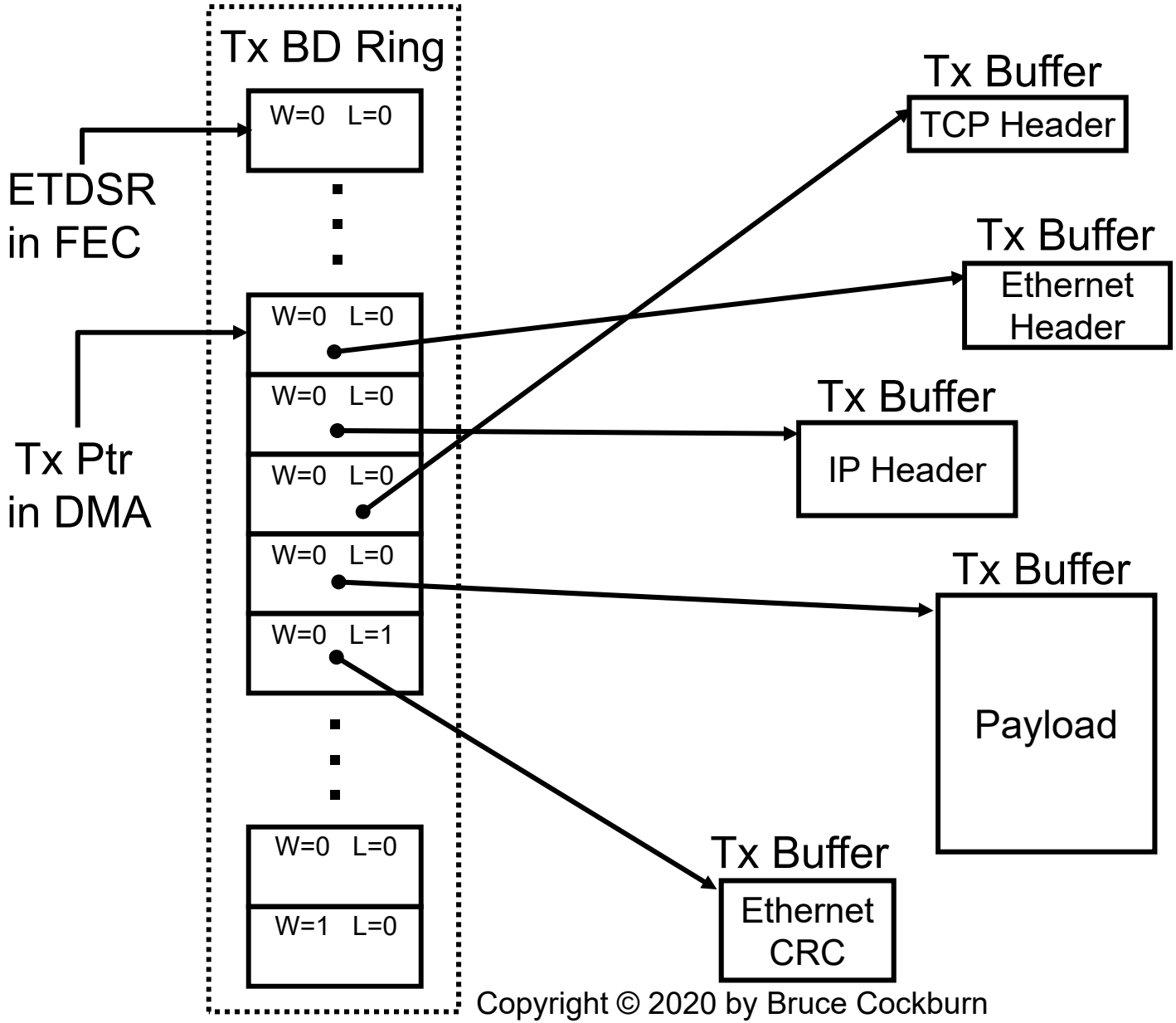
Data Buffers in External Memory



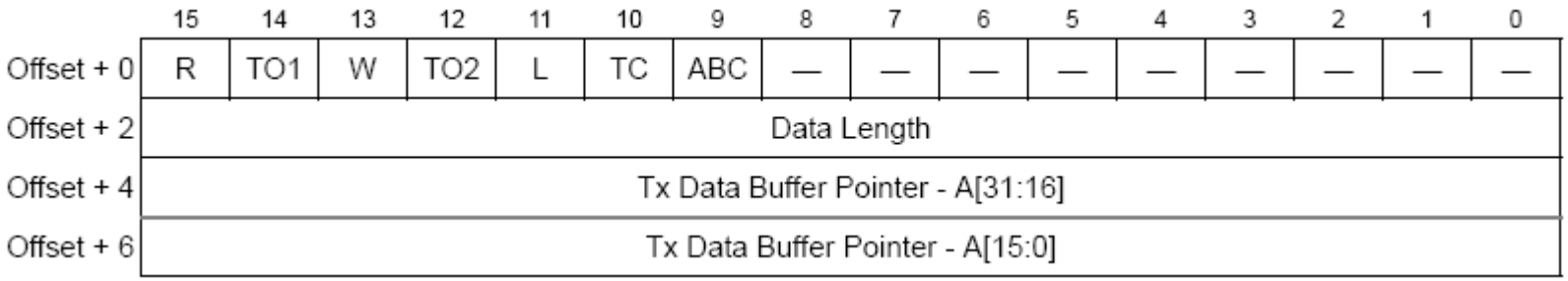
Transmit and Receive Buffer Overview (cont'd)

- The Tx and Rx buffer descriptors are 8-byte data structures that are packed into two arrays called the ***Tx*** and ***Rx BD Rings***.
- The Tx and Rx BDs have multiple fields (described later).
- The W field in the last BD in a ring is set to 1; otherwise, it is 0.
- The Tx buffers can have variable length, as specified in the data length field in the corresponding Tx BDs.
- The Rx buffers have the same length, which is specified in the ***Ethernet Maximum Receiver Buffer Size Register*** (EMRBR). This length is negotiated in the TCP layer during connection set-up.
- The Tx and Rx buffers are automatically concatenated together to form a frame. The field L = 1 in the BD corresponding to the last buffer in a frame; the field L = 0 in the BDs for all other buffers in the frame.
- A buffer management system keeps track of the pool of buffers and recycles used empty buffers into a list of idle buffers.

Example of a Tx Frame



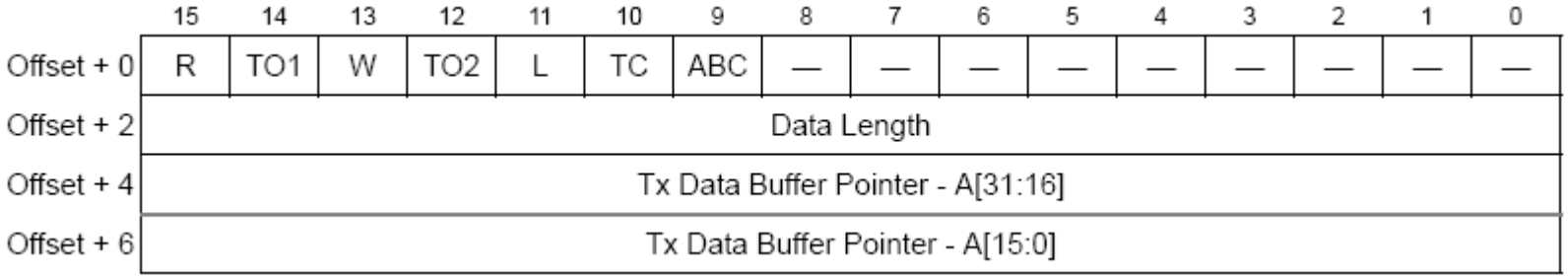
Transmit Buffer Descriptor (TxBD) 1



	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Offset + 0	R	TO1	W	TO2	L	TC	ABC	—	—	—	—	—	—	—	—	—
Offset + 2	Data Length															
Offset + 4	Tx Data Buffer Pointer - A[31:16]															
Offset + 6	Tx Data Buffer Pointer - A[15:0]															

Word	Bits	Field Name	Description
Offset + 0	15	R	Ready. Written by the FEC and the user. 0 The data buffer associated with this BD is not ready for transmission. The user is free to manipulate this BD or its associated data buffer. The FEC clears this bit after the buffer has been transmitted or after an error condition is encountered. 1 The data buffer, which has been prepared for transmission by the user, has not been transmitted or is currently being transmitted. No fields of this BD may be written by the user once this bit is set.
Offset + 0	14	TO1	Transmit software ownership. This field is reserved for software use. This read/write bit will not be modified by hardware, nor will its value affect hardware.
Offset + 0	13	W	Wrap. Written by user. 0 The next buffer descriptor is found in the consecutive location 1 The next buffer descriptor is found at the location defined in ETDSR.

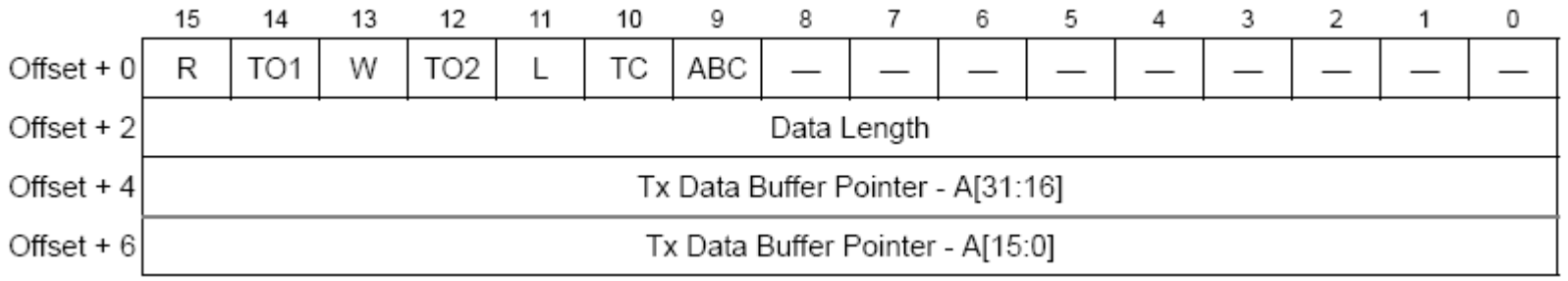
Transmit Buffer Descriptor (TxBD) 2



	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Offset + 0	R	TO1	W	TO2	L	TC	ABC	—	—	—	—	—	—	—	—	—
Offset + 2	Data Length															
Offset + 4	Tx Data Buffer Pointer - A[31:16]															
Offset + 6	Tx Data Buffer Pointer - A[15:0]															

Word	Bits	Field Name	Description
Offset + 0	11	L	Last in frame. Written by user. 0 The buffer is not the last in the transmit frame. 1 The buffer is the last in the transmit frame.
Offset + 0	10	TC	Tx CRC. Written by user (only valid if L = 1). 0 End transmission immediately after the last data byte. 1 Transmit the CRC sequence after the last data byte.
Offset + 0	9	ABC	Append bad CRC. Written by user (only valid if L = 1). 0 No effect 1 Transmit the CRC sequence inverted after the last data byte (regardless of TC value).
Offset + 0	8–0	—	Reserved.

Transmit Buffer Descriptor (TxBD) 3



	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Offset + 0	R	TO1	W	TO2	L	TC	ABC	—	—	—	—	—	—	—	—	—
Offset + 2	Data Length															
Offset + 4	Tx Data Buffer Pointer - A[31:16]															
Offset + 6	Tx Data Buffer Pointer - A[15:0]															

Word	Bits	Field Name	Description
Offset + 2	15–0	Data Length	Data Length, written by user. Data length is the number of octets the FEC should transmit from this BD's data buffer. It is never modified by the FEC. Bits [15:5] are used by the DMA engine, bits[4:0] are ignored.
Offset + 4	15–0	A[31:16]	Tx data buffer pointer, bits [31:16][1]
Offset + 6	15–0	A[15:0]	Tx data buffer pointer, bits [15:0].

Software Processing Steps for Tx Buffers

1. Software writes application payload into a first Tx buffer.
2. Software writes TCP header into a second Tx buffer.
3. Software writes IP header into a third Tx buffer.
4. Software writes Ethernet header into a fourth Tx buffer.
   (*Note*: Forming this header is actually a MAC function).
5. If the software is going to supply the Ethernet CRC, then the TC bit in all four Tx buffer descriptors must be cleared to 0. If the FEC is going to compute the Ethernet CRC, then the last buffer descriptor must have TC =1, with TC = 0 in all other buffer descriptors.
   (*Note*: Computing the Ethernet CRC is actually a MAC function.)
6. The R bits must be set to 1. The R bit in the first buffer descriptor must be set to 1 last to avoid premature transmission.
7. The FEC hardware is informed by writing (any value) to the ***Transmit Descriptor Active Register*** (TDAR).

FEC Hardware Processing Steps for Tx Buffers

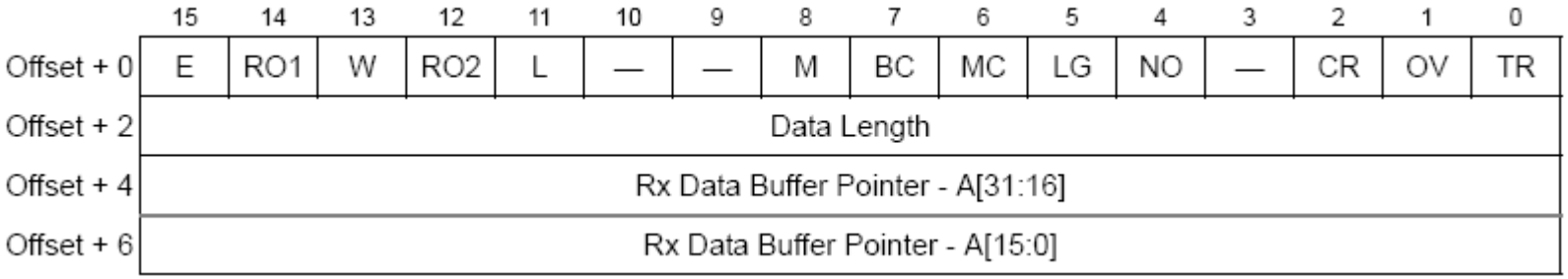
1. The FEC detects that the TDAR has been written.
2. The DMA Controller in the FEC reads the next Tx buffer descriptor in the Tx buffer descriptor ring.
3. If the R bit is 1, then the bytes are transferred from the Tx buffer into the FEC transmitter hardware by DMA.
4. After all bytes have been transferred out of a buffer, the R bit in the corresponding Tx buffer descriptor is cleared to 0. This releases the buffer to be recycled.
5. The next Tx buffer descriptor is inspected. If R = 1, then go back to step 3. If R = 0, stop further Tx buffer descriptor (and hence also buffer) processing.
6. Wait for the TDAR register to be written next.

FEC Receive Buffer Size Register (EMRBR)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W																
Reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	R_BUF_SIZE							0	0	0	0
W																
Reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Address	IPSBAR + 0x1188															

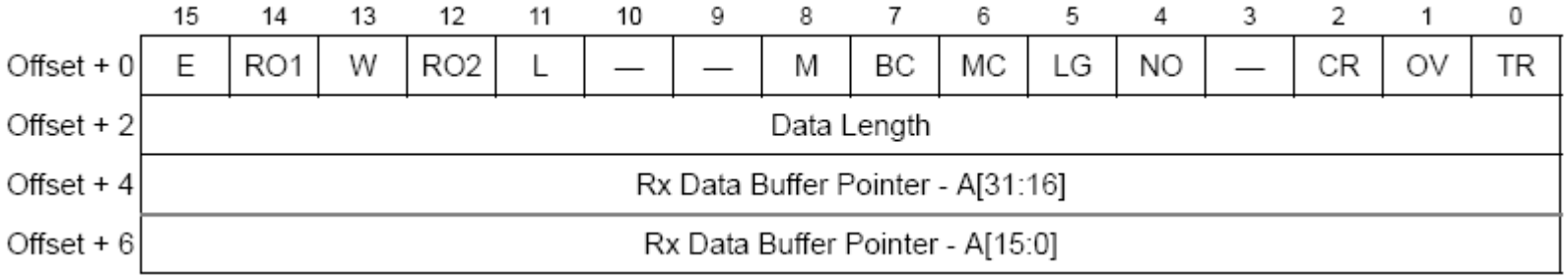
Receive Buffer Descriptor (RxBD) 1



	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Offset + 0	E	RO1	W	RO2	L	—	—	M	BC	MC	LG	NO	—	CR	OV	TR
Offset + 2	Data Length															
Offset + 4	Rx Data Buffer Pointer - A[31:16]															
Offset + 6	Rx Data Buffer Pointer - A[15:0]															

Word	Bits	Field Name	Description
Offset + 0	15	E	Empty. Written by the FEC (=0) and user (=1). 0 The data buffer associated with this BD has been filled with received data, or data reception has been aborted due to an error condition. The status and length fields have been updated as required. 1 The data buffer associated with this BD is empty, or reception is currently in progress.
Offset + 0	14	RO1	Receive software ownership. This field is reserved for use by software. This read/write bit will not be modified by hardware, nor will its value affect hardware.
Offset + 0	13	W	Wrap. Written by user. 0 The next buffer descriptor is found in the consecutive location 1 The next buffer descriptor is found at the location defined in ERDSR.

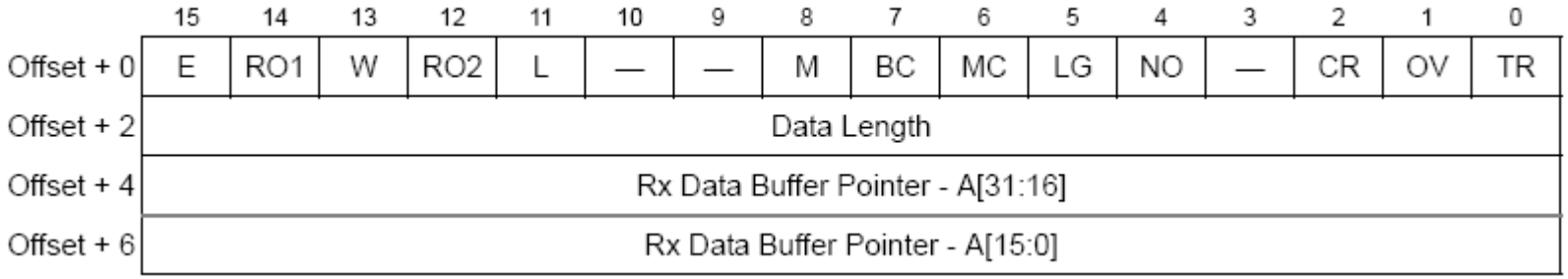
Receive Buffer Descriptor (RxBD) 2



	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Offset + 0	E	RO1	W	RO2	L	—	—	M	BC	MC	LG	NO	—	CR	OV	TR
Offset + 2	Data Length															
Offset + 4	Rx Data Buffer Pointer - A[31:16]															
Offset + 6	Rx Data Buffer Pointer - A[15:0]															

Word	Bits	Field Name	Description
Offset + 0	12	RO2	Receive software ownership. This field is reserved for use by software. This read/write bit will not be modified by hardware, nor will its value affect hardware.
Offset + 0	11	L	Last in frame. Written by the FEC. 0 The buffer is not the last in a frame. 1 The buffer is the last in a frame.
Offset + 0	10–9	—	Reserved.

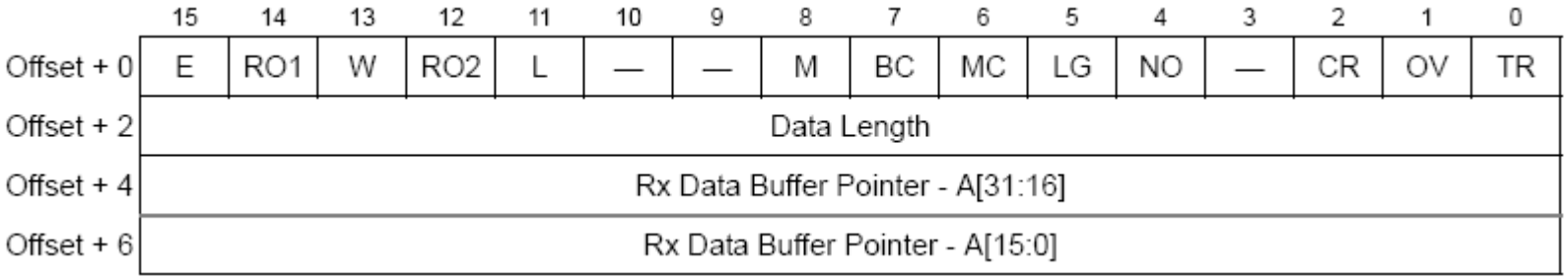
Receive Buffer Descriptor (RxBD) 3



	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Offset + 0	E	RO1	W	RO2	L	—	—	M	BC	MC	LG	NO	—	CR	OV	TR
Offset + 2	Data Length															
Offset + 4	Rx Data Buffer Pointer - A[31:16]															
Offset + 6	Rx Data Buffer Pointer - A[15:0]															

Word	Bits	Field Name	Description
Offset + 0	8	M	Miss. Written by the FEC. This bit is set by the FEC for frames that were accepted in promiscuous mode, but were flagged as a "miss" by the internal address recognition. Thus, while in promiscuous mode, the user can use the M-bit to quickly determine whether the frame was destined to this station. This bit is valid only if the L-bit is set and the PROM bit is set. 0 The frame was received because of an address recognition hit. 1 The frame was received because of promiscuous mode.
Offset + 0	7	BC	Will be set if the DA is broadcast (FF-FF-FF-FF-FF-FF).
Offset + 0	6	MC	Will be set if the DA is multicast and not BC.
Offset + 0	5	LG	Rx frame length violation. Written by the FEC. A frame length greater than RCR[MAX_FL] was recognized. This bit is valid only if the L-bit is set. The receive data is not altered in any way unless the length exceeds 2032 bytes.

Receive Buffer Descriptor (RxBD) 4



	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Offset + 0	E	RO1	W	RO2	L	—	—	M	BC	MC	LG	NO	—	CR	OV	TR
Offset + 2	Data Length															
Offset + 4	Rx Data Buffer Pointer - A[31:16]															
Offset + 6	Rx Data Buffer Pointer - A[15:0]															

Word	Bits	Field Name	Description
Offset + 0	4	NO	Receive non-octet aligned frame. Written by the FEC. A frame that contained a number of bits not divisible by 8 was received, and the CRC check that occurred at the preceding byte boundary generated an error. This bit is valid only if the L-bit is set. If this bit is set the CR bit will not be set.
Offset + 0	3	—	Reserved.
Offset + 0	2	CR	Receive CRC error. Written by the FEC. This frame contains a CRC error and is an integral number of octets in length. This bit is valid only if the L-bit is set.
Offset + 0	1	OV	Overrun. Written by the FEC. A receive FIFO overrun occurred during frame reception. If this bit is set, the other status bits, M, LG, NO, CR, and CL lose their normal meaning and will be zero. This bit is valid only if the L-bit is set.
Offset + 0	0	TR	Will be set if the receive frame is truncated (frame length > 2032 bytes). If the TR bit is set the frame should be discarded and the other error bits should be ignored as they may be incorrect.

Receive Buffer Descriptor (RxBD) 5

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Offset + 0	E	RO1	W	RO2	L	—	—	M	BC	MC	LG	NO	—	CR	OV	TR
Offset + 2	Data Length															
Offset + 4	Rx Data Buffer Pointer - A[31:16]															
Offset + 6	Rx Data Buffer Pointer - A[15:0]															

Word	Bits	Field Name	Description
Offset + 2	15–0	Data Length	Data length. Written by the FEC. Data length is the number of octets written by the FEC into this BD's data buffer if L = 0 (the value will be equal to EMRBR), or the length of the frame including CRC if L = 1. It is written by the FEC once as the BD is closed.
0ffset + 4	15–0	A[31:16]	RX data buffer pointer, bits [31:16][1]
Offset + 6	15–0	A[15:0]	RX data buffer pointer, bits [15:0]

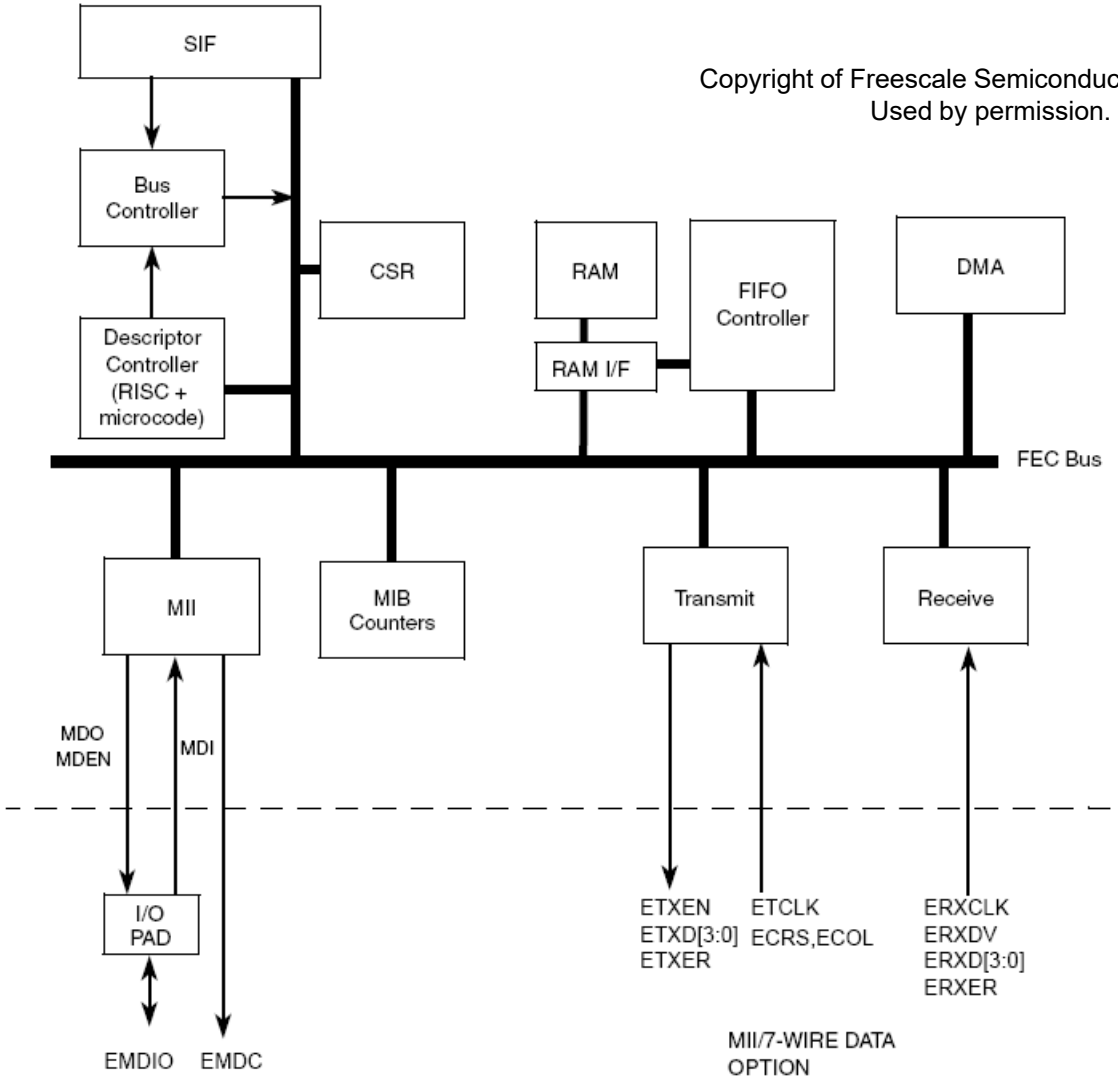
Software Processing Steps for Rx Buffers

1. Software initializes a sufficient number of Rx buffers to hold the frame that is expected to arrive from the PHY. The buffers must all have the length that was written earlier into the EMRBR.
2. The empty bit E must be set to 1 in all of the corresponding Rx buffer descriptors.
3. The FEC hardware is alerted by writing (any value) to the ***Receive Descriptor Active Register*** (RDAR).
4. The software is free to go on to other work; meanwhile, the FEC hardware will take care of transferring arriving data into the buffers.
5. The ***Receive Frame Interrupt*** (RXF) bit in the ***Ethernet Interrupt Event Register*** (EIR) is set to 1 by the FEC hardware as soon as the last buffer in a frame has been received.
6. The FEC interrupt service routine transfers the Rx buffers to Rx software. The Rx buffers are recycled later. The Rx EIMR mask bits must be cleared, or the Rx EIR status bits must be written to 1.

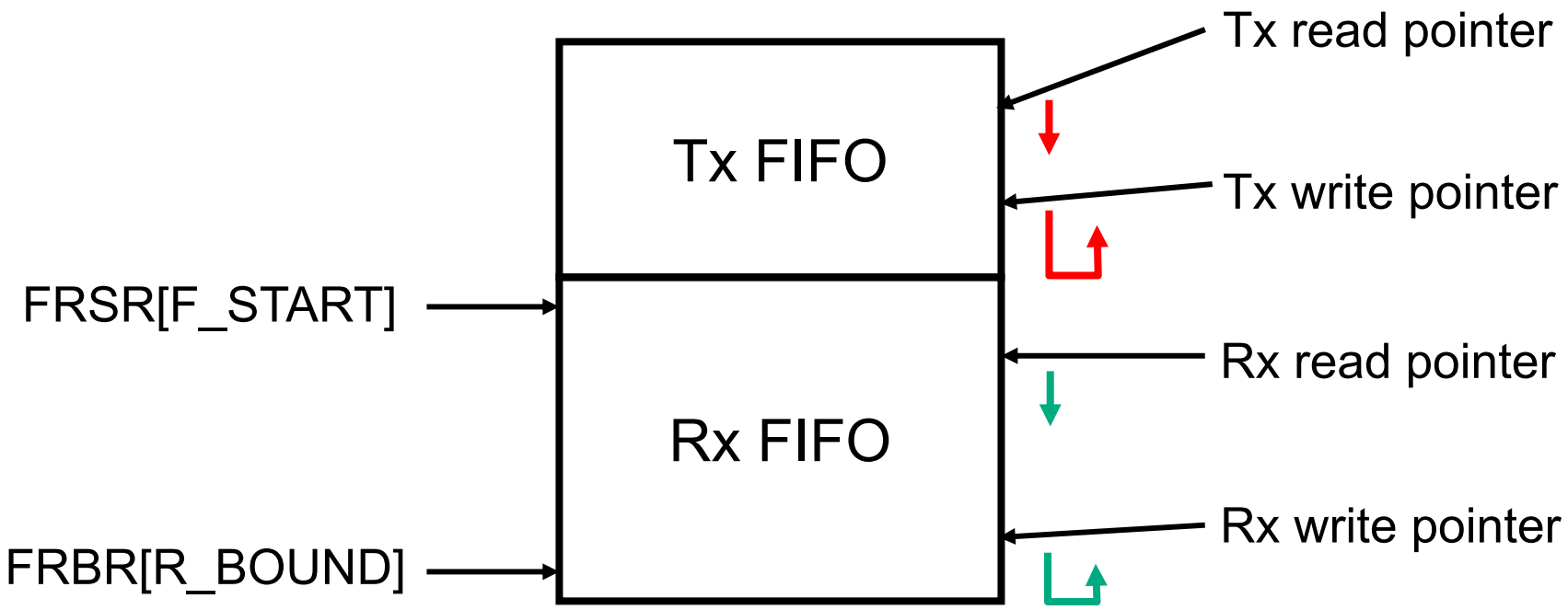
FEC Hardware Processing Steps for Rx Buffers

1. Wait until the FEC RDAR is written by CPU software.
2. The DMA Controller in the FEC reads the next Rx buffer descriptor in the Rx buffer descriptor ring.
3. If the E bit is 1, then arriving bytes are transferred by DMA from the FEC receiver hardware into the Rx buffer. If E = 0 in two read attempts, then stop Rx buffer processing and go back to step 1.
4. After an Rx buffer has been filled up (or the last data byte in the frame has been received), the E bit in the corresponding Rx buffer descriptor is cleared to 0. If the last byte in the frame was received, then go to step 6.
5. Set the ***Receive Buffer Interrupt*** (RBF) bit and clear the ***Receive Frame Interrupt*** (RXF) bit in the ***Ethernet Interrupt Event Register*** (EIR). Go back to step 2.
6. Write the total frame length in the length field of the last Rx buffer.
7. Clear the RBF bit and set the RXF bit in the EIR. Go back to step 1.

Fast Ethernet Controller (FEC) Architecture

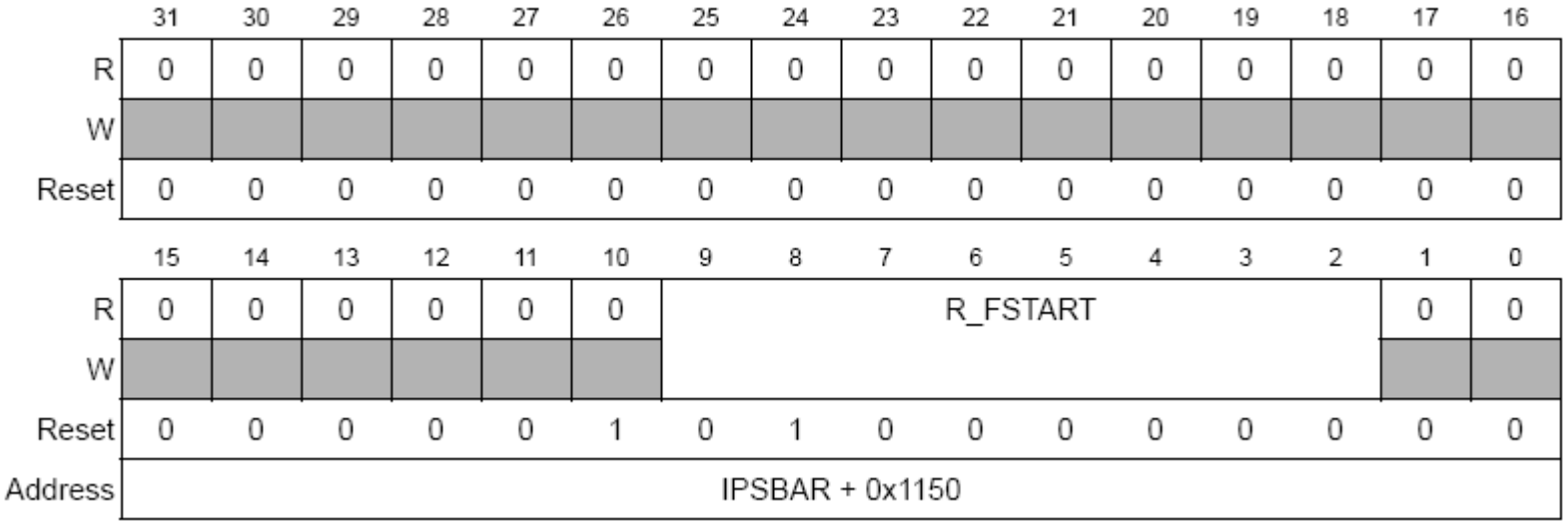


First-In First-Out Buffer



Note: The read and write pointers are incremented (with wrap-around) automatically by the FEC ***Direct Memory Access*** (DMA) controller.

FIFO Receive Start Register (FRSR)

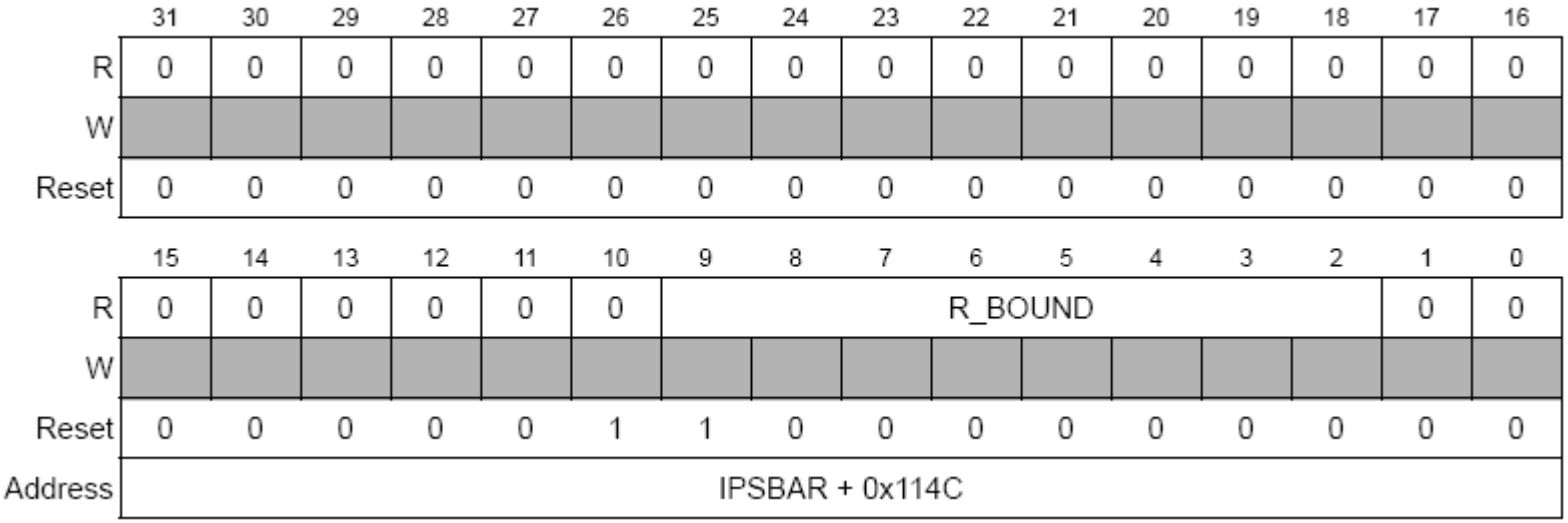


	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	R_FSTART								0	0
W																
Reset	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0
Address	IPSBAR + 0x1150															

Bits	Name	Descriptions
31–10	—	Reserved, read as 0 (except bit 10, which is read as 1).
9–2	R_FSTART	Address of first receive FIFO location. Acts as delimiter between receive and transmit FIFOs.
1–0	—	Reserved, read as 0.

FIFO Receive Bound Register (FRBR)

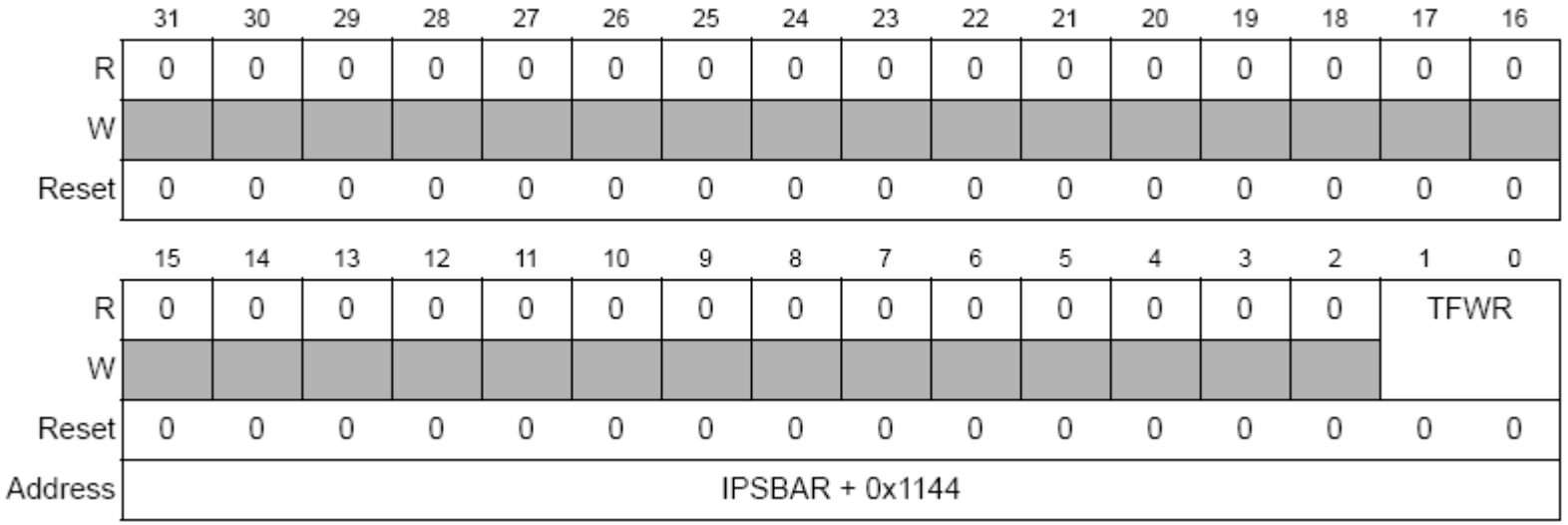


	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	R_BOUND								0	0
W																
Reset	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0
Address	IPSBAR + 0x114C															

Bits	Name	Descriptions
31–10	—	Reserved, read as 0 (except bit 10, which is read as 1).
9–2	R_BOUND	Read-only. Highest valid FIFO RAM address.
1–0	—	Reserved, should be cleared.

Transmit FIFO Watermark Register (TFWR)



	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	TFWR	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Address	IPSBAR + 0x1144															

Bits	Name	Descriptions
31–2	—	Reserved, should be cleared.
1–0	TFWR	Number of bytes written to transmit FIFO before transmission of a frame begins 0x 64 bytes written 10 128 bytes written 11 192 bytes written

FEC MAC Initialization (before ECR[ETHER_EN])

Description
Initialize EIMR
Clear EIR (write 0xFFFF_FFFF)
TFWR (optional)
IALR / IAUR
GAUR / GALR
PALR / PAUR (only needed for full duplex flow control)
OPD (only needed for full duplex flow control)
RCR
TCR
MSCR (optional)
Clear MIB_RAM (locations IPSBAR + 0x1200-0x12FC)

FEC DMA Initialization (before ECR[ETHER_EN])

Description
Initialize FRSR (optional)
Initialize EMRBR
Initialize ERDSR
Initialize ETDSR
Initialize (Empty) Transmit Descriptor ring
Initialize (Empty) Receive Descriptor ring

FEC Initialization (after ECR[ETHER_EN])

Description
Initialize BackOff Random Number Seed
Activate Receiver
Activate Transmitter
Clear Transmit FIFO
Clear Receive FIFO
Initialize Transmit Ring Pointer
Initialize Receive Ring Pointer
Initialize FIFO Count Registers