SIEMENS

ICs for Communications

Enhanced Serial Communication Controller with 8 Channels
ESCC8
SAB 82538
SAF 82538

Version 2.2

User's Manual 03.95

Edition 03.95

This edition was realized using the software system FrameMaker.

**Published by Siemens AG,
Bereich Halbleiter, Marketing-
Kommunikation, Balanstraße 73,
81541 München**

Attention please!

As far as patents or other rights of third parties are concerned, liability is only assumed for components, not for applications, processes and circuits implemented within components or assemblies.

The information describes the type of component and shall not be considered as assured characteristics.

Terms of delivery and rights to change design reserved.

For questions on technology, delivery and prices please contact the Semiconductor Group Offices in Germany or the Siemens Companies and Representatives worldwide (see address list).

Due to technical requirements components may contain dangerous substances. For information on the types in question please contact your nearest Siemens Office, Semiconductor Group.

Siemens AG is an approved CECC manufacturer.

Packing

Please use the recycling operators known to you. We can also help you – get in touch with your nearest sales office. By agreement we will take packing material back, if it is sorted. You must bear the costs of transport.

For packing material that is returned to us unsorted or which we are not obliged to accept, we shall have to invoice you for any costs incurred.

Components used in life-support devices or systems must be expressly authorized for such purpose!

Critical components[1] of the Semiconductor Group of Siemens AG, may only be used in life-support devices or systems[2] with the express written approval of the Semiconductor Group of Siemens AG.

1 A critical component is a component used in a life-support device or system whose failure can reasonably be expected to cause the failure of that life-support device or system, or to affect its safety or effectiveness of that device or system.

2 Life support devices or systems are intended (a) to be implanted in the human body, or (b) to support and/or maintain and sustain human life. If they fail, it is reasonable to assume that the health of the user may be endangered.

SAB 82538 **SAF 82538** **Revision History:**		**Current Version: 03.95**
Previous Version:		User's Manual 01.94
Page (in Version 01.94)	Page (in new version	Subjects (changes since last revision)
11	11	V_{DD} and V_{SS} pin configuration
13	13	$\overline{RD}/\overline{DS}$ Description
14	14	$\overline{RES}$ pin number
19	19	$\overline{DACKx}$ Description
22	22	V_{DD} and V_{SS} numbers
54	54	Transparent Mode 1, Address Recognition
58	58	Transmitter Interrupts, Figure 21
61	61	Transmitter Interrupts, Figures 23/24
65	65	Transmitter Interrupts, Figures 28/29
86	86	Note in description of Clock Mode 5
86	86	CD-signal, Figure 35
110	111	Interrupt Driven Reception Sequence, Figure 47
113	114	CCR4 Register Addresses
126	127	RHCR Register Description
127	129	XBC Register Description
145	147/148	XMR Interrupt Description
157	159	RFRD Command Description
164	166	RBC Register Description
178	181	TCD Interrupt Description
190	194	RFRD Command Description
198	201	RBC Register Description
211	214	TCD Interrupt Description
228	231	Timings $t_{p\ (DRT)}$, $t_{SU\ (IE)}$
236	239	Timing $t_{SU\ (IE)}$
238	241	Timing $t_{p\ (PV\text{-}INT)}$
241	244	Number of Timing $t_{C\ (XC)}$
–	–	Minor Misprints

Data Classification

Maximum Ratings

Maximum ratings are absolute ratings; exceeding only one of these values may cause irreversible damage to the integrated circuit.

Characteristics

The listed characteristics are ensured over the operating range of the integrated circuit. Typical characteristics specify mean values expected over the production spread. If not otherwise specified, typical characteristics apply at T_A = 25 °C and the given supply voltage.

Operating Range

In the operating range the functions given in the circuit description are fulfilled.

For detailed technical information about **"Processing Guidelines"** and **"Quality Assurance"** for ICs, see our **"Product Overview"**.

Table of Contents **Page**

Table of Contents (cont'd) **Page**

Table of Contents (cont'd) **Page**

Introduction

The Enhanced Serial Communication Controller ESCC8 (SAB 82538) is a data communication device with eight serial channels. It has been designed to implement high-speed communication links and to reduce hardware and software overhead needed for serial synchronous/asynchronous communications.

Each channel contains an independent clock generator, DPLL, encoder/decoder and a programmable protocol part. Data communication with asynchronous, synchronous character oriented, and HDLC based protocols with extended support of X.25 “0”, the ISDN “0”, and SDLC protocols is implemented. Like the dual channel ESCC2 (SAB 82532) the ESCC8 is capable of handling a large set of layer-2 protocol functions independently of the host processor.

The version 82538H-10 of the Enhanced Serial Communication Controller (ESCC8) opens a wide area for applications which use time division multiplex methods (e.g. time-slot oriented PCM systems, systems designed for packet switching, ISDN applications) by its programmable telecom-specific features. Specifically in one of its timing modes (clock mode 5), which is applicable to all serial modes (HDLC/SDLC, ASYNC, BISYNC), the ESCC8 can transmit or receive data packets in one of up to 64 time-slots of programmable width.

The device is controlled via a parallel 16-bit wide interface which is directly compatible with the most popular 8/16 bit microprocessors (Siemens/Intel or Motorola type). The internal FIFOs (64 bytes per direction and channel) with additional DMA capability provide a powerful interface to the higher layers implemented in a microcontroller. For interrupt controlled systems, the ESCC8 supports daisy chaining and interrupt vector generation.

The ESCC8 is fabricated using Siemens advanced CMOS technology and is available in a P-MQFP-160 package.

Applications

- Universal, multiprotocol communication boards
- Asynchronous and synchronous terminal cluster controllers
- LAN gateways and bridges
- Multiplexers, cross-connect points, DMI boards
- Time slotted packet networks
- Packet switches, packet assemblers/disassemblers

SIEMENS

Enhanced Serial Communication Controller (ESCC8)

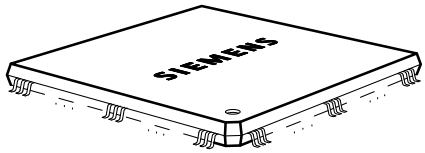
SAB 82538
SAF 82538

Preliminary Data

CMOS IC

1 General Features

Serial Interface



P-MQFP-160

- Eight independent full duplex serial channels
  - On chip clock generation or external clock source
  - On chip DPLL for clock recovery of each channel
  - Eight independent baud rate generators
  - Independent time-slot assignment for each channel with programmable time-slot length (1-256 bits)
- Async., sync. character oriented (MONOSYNC / BISYNC) or HDLC/SDLC modes (including SDLC LOOP)
- Transparent receive/transmit of data bytes without framing
- NRZ, NRZI, FM and Manchester encoding
- Modem control lines ($\overline{\text{RTS}}$, $\overline{\text{CTS}}$, CD)
- CRC support:
  - HDLC/SDLC: CRC-CCITT or CRC-32 (automatic handling for transmit/receive direction)
  - BISYNC: CRC-16 or CRC-CCITT (support for transmit direction)
- Support of bus configuration by collision detection and resolution

Type	Ordering Code	Package	Max. Data Rate clocked		Time-Slot Mode
			ext.	int. (DPLL)	
SAB 82538 H	Q67100-H6440	P-MQFP-160	2 Mbit/s	2 Mbit/s	no
SAB 82538 H-10	Q67100-H6441	P-MQFP-160	10 Mbit/s	2 Mbit/s	yes
SAF 82538 H-10	Q67100-H6442	P-MQFP-160	10 Mbit/s	2 Mbit/s	yes

- Statistical multiplexing
- Continuous transmission of 1 to 32 bytes possible
- Programmable Preamble (8 bit) with selectable repetition rate (HDLC/SDLC and BISYNC)
- Data rate up to 10 Mbit/s
- Master clock mode with data rate up to 4 Mbit/s

Protocol Support (HDLC / SDLC)

- Various types of protocol support depending on operating mode
  - Auto mode (automatic handling of S and I frames)
  - Non-auto mode
  - Transparent mode
- Handling of bit oriented functions
- Support of LAPB / LAPD / SDLC / HDLC protocol in auto mode (I- and S-frame handling)
- Modulo 8 or modulo 128 operation
- Programmable time-out and retry conditions
- Programmable maximum packet size checking

MP Interface and Ports

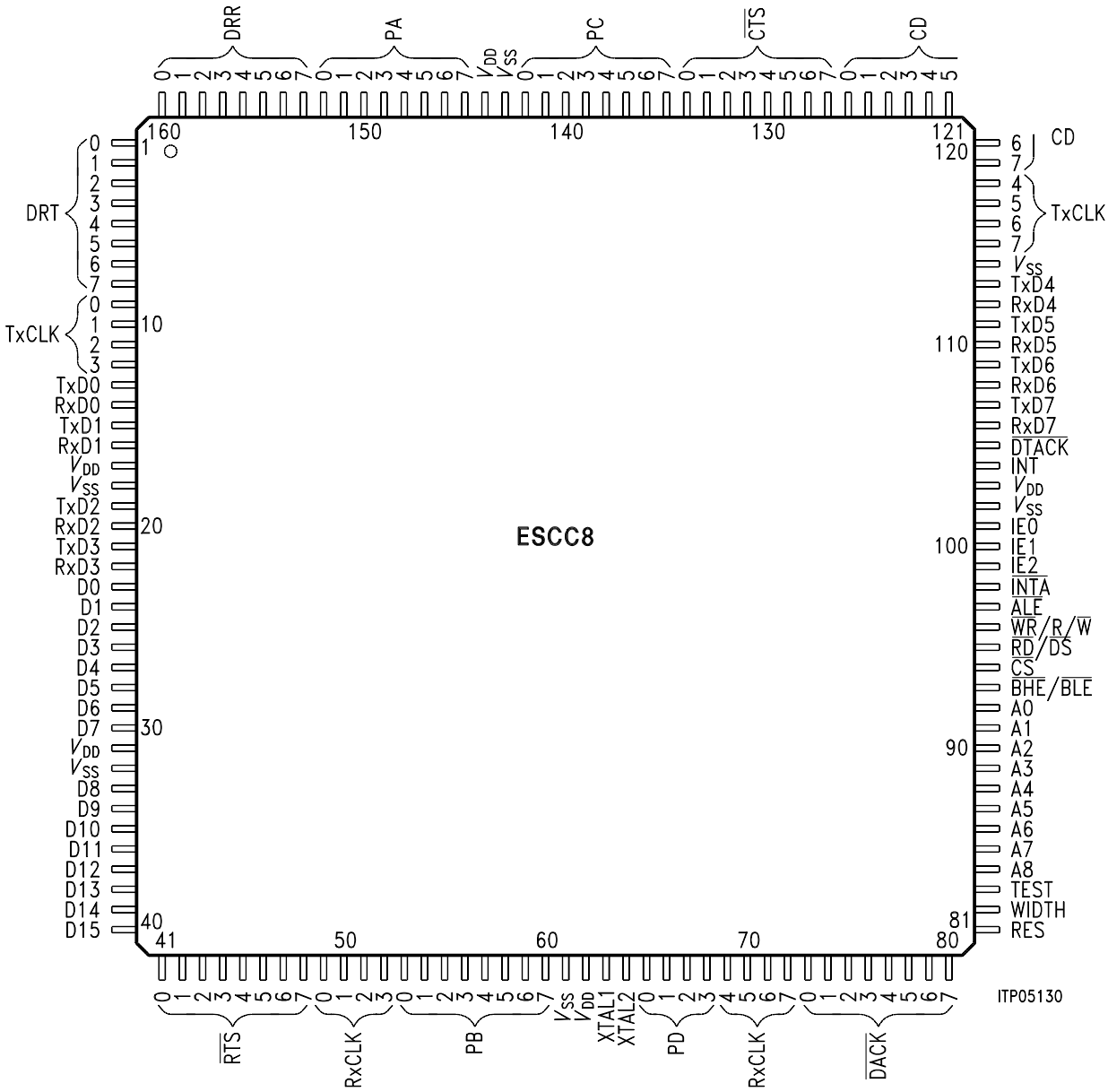
- 64 byte FIFOs per channel and direction (byte or word access)
- 8/16 bit microprocessor bus interface (Intel or Motorola type)
- All registers directly accessible (byte and word access)
- Efficient transfer of data blocks from/to system memory via DMA or interrupt request
- Support of Daisy Chaining and Slave Operation with Interrupt Vector generation
- 28-bit programmable universal I/Os

General

- Advanced CMOS technology
- Low power consumption: active 200 mW at 2 MHz/standby 20 mW (typical values)
- P-MQFP-160 Package

Pin Configuration of ESCC8
(top view)

P-MQFP-160



1.1 Pin Definitions and Function

Pin No.	Symbol	Input (I) Output (O)	Function
92 … 84	A0 … A8	I	**Address Bus** These inputs interface with nine bits of the system's address bus to select one of the internal registers for read or write.
23 … 30, 33 … 40	D0 … D15	I/O	**Data Bus** Bi-directional three-state data lines which interface with the system's data bus. Their configuration is controlled by the level of pin WIDTH: – 8-bit mode (WIDTH = 0): D0 … D7 are active. D8 … D15 are in high impedance and have to be connected to V_{DD} or V_{SS}. – 16-bit mode (WIDTH = 1): D8…D15 are active. In case of byte transfers, the active half of the bus is determined by A0 and $\overline{BHE}$/$\overline{BLE}$ and the selected bus interface mode (via ALE). The unused half is in high impedance. For detailed information, **refer to chapter 2.2.1**.
97	ALE	I	**Address Latch Enable** The level at this pin defines the bus interface mode: **Fixed to "0":** Demultiplexed Siemens/Intel bus interface **Fixed to "1":** Demultiplexed Motorola bus interface **Switching:** Multiplexed Siemens/Intel bus interface The address information provided on lines A0 … A8 is internally latched with the falling edge of ALE. This function allows the ESCC8 to be directly connected to a multiplexed address/data bus. In this case, pins A0 … A8 must be externally connected to the Data Bus pins.

Note: All unused input pins have to be connected to a defined level

Pin Definitions and Function (cont'd)

Pin No.	Symbol	Input (I) Output (O)	Function
95	$\overline{RD}/\overline{DS}$	I	**Read Enable** (Siemens/Intel bus mode) This signal indicates a read operation. When the ESCC8 is selected via $\overline{CS}$ the $\overline{RD}$ signal enables the bus drivers to output data from an internal register addressed via A0 … A8 on to Data Bus. For more information about control/status register and FIFO access in the different bus interface modes **refer to chapter 2**. If DMA transfer is selected via $\overline{DACKx}$, the $\overline{RD}$ signal enables the bus drivers to put data from the corresponding Receive FIFO on the Data Bus. Inputs A1 … A8 are ignored. A0 and $\overline{BHE}/\overline{BLE}$ are used to select byte or word access. **Data Strobe** (Motorola bus mode) This pin serves as input to control read/write operations.
96	$\overline{WR}/R/\overline{W}$	I	**Write Enable** (Siemens/Intel bus mode) This signal indicates a write operation. When $\overline{CS}$ is active the ESCC8 loads an internal register with data provided via the Data Bus. For more information about control/status register and FIFO access in the different bus interface modes **refer to chapter 2**. If DMA transfer is selected via $\overline{DACKx}$ the $\overline{WR}$ signal enables latching data from the Data Bus on the top of the corresponding Transmit FIFO. Inputs A0 … A8 are ignored. **Read/Write Enable** (Motorola bus mode) This signal distinguishes between read and write operation.
94	$\overline{CS}$	I	**Chip Select** A low signal selects the ESCC8 for read/write operations. $\overline{CS}$ has no function in interrupt acknowledge or DMA cycles.

Pin Definitions and Function (cont'd)

Pin No.	Symbol	Input (I) Output (O)	Function
81	RES	I	**Reset** A high signal on this pin forces the ESCC8 into reset state. During Reset the ESCC8 is in power up mode, after Reset in power down mode. Re-activation of each channel is done via bit CCR0.PU (**refer to chapter 3.2**). During Reset – all uni-directional output stages are in high-impedance state, – all bi-directional output stages (data bus) are in high-impedance state if signals $\overline{RD}$ and $\overline{INTA}$ are "high", – "output" XTAL2 is in high-impedance if input XTAL1 is "high" (the internal oscillator is disabled during reset)
93	$\overline{BHE}/\overline{BLE}$	I	**Bus High Enable** (Siemens/Intel bus mode) If 16-bit bus interface mode is enabled, this signal indicates a data transfer on the upper byte of the data bus (D8 … D15). In 8-bit bus interface mode this signal has no function and should be tied to V_{DD}. **Refer to chapter 2.2.1** for detailed information. **Bus Low Enable** (Motorola bus mode) If 16-bit bus interface mode is enabled, this signal indicates a data transfer on the lower byte of the data bus (D0 … D7). In 8-bit bus interface mode this signal has no function and should be tied to V_{DD}. **Refer to chapter 2.2.1** for detailed information.
82	WIDTH	I	**Width Of Bus Interface** (Bus Interface Mode) A low signal on this input selects the 8-bit bus interface mode. A high signal on this input selects the 16-bit bus interface mode. In this case word transfer to/from the internal registers is enabled. Byte transfers are implemented by using A0 and $\overline{BHE}/\overline{BLE}$.

Pin Definitions and Function (cont'd)

Pin No.	Symbol	Input (I) Output (O)	Function
105	$\overline{\text{DTACK}}$	oD	**Data Transfer Acknowledge** During a bus cycle (read/write, asynchronous bus), this signal indicates that ESCC8 is ready for data transfer. The signal remains active until the data strobe ($\overline{\text{DS}}$, $\overline{\text{RD}}$ or $\overline{\text{WR}}$) and/or the Chip Select signal ($\overline{\text{CS}}$) or the Interrupt Acknowledge ($\overline{\text{INTA}}$) go inactive. An external resistor has to be tied to V_{DD} if this function is used.
104	INT	O/oD	**Interrupt Request** INT serves as general interrupt request which may include all serial mode specific interrupt sources and the requests of the four universal ports if programmed. These interrupt sources can be masked via registers IMR0/1 (for each channel) and PIMA,B,C,D (universal ports). Interrupt status is reported via registers GIS (Global Interrupt Status), ISR0/1 (for each channel) and PISA,B,C,D (universal ports). Output characteristics (push-pull active low/high, open drain) are determined by programming the IPC register. In Daisy Chain cascading mode INT signal generation is only enabled if the Interrupt Enable input IE1 is active "high". INT is reset if –interrupts are disabled in Daisy Chain cascading mode (pin IE1 = low), –no further interrupt is pending, i.e. all interrupt status bits are reset.

Pin Definitions and Function (cont'd)

Pin No.	Symbol	Input (I) Output (O)	Function
98	$\overline{\text{INTA}}$	I	**Interrupt Acknowledge** If the interrupt is acknowledged via pin $\overline{\text{INTA}}$, an interrupt vector is output on D0…D7. All interrupt sources are organized in groups with fixed priority. The priority of the channels within a group is fixed or adjusted dynamically (rotating priority scheme, version 2 upward) (**refer to chapter 2**). The generated interrupt vector refers to the interrupt group and the requesting channel with currently highest priority (although more than one interrupt source/group may be active). Reaction on $\overline{\text{INTA}}$ signal depends on the bus interface mode and the cascading mode in conjunction with the Interrupt Enable pins IE0-2 (ref. to IPC register): **Motorola bus mode:** INT is reset with the rising edge of the following valid $\overline{\text{INTA}}$ cycle if no further interrupt is pending. The interrupt vector is output with signal $\overline{\text{DS}}$. **Siemens/Intel bus mode:** INT is reset with the rising edge of the **second** valid $\overline{\text{INTA}}$ cycle (2-cycle '86 mode) if no further interrupt is pending. **Slave mode:** Interrupt acknowledge is accepted if an interrupt signal has been generated and the slave address provided via IE0-2 corresponds to the programmed value (IPC register). **Daisy Chaining mode:** Interrupt acknowledge is accepted if an interrupt signal has been generated and Interrupt Enable input IE1 is active during the following $\overline{\text{INTA}}$ cycle. **Note:** Pins $\overline{\text{CS}}$, $\overline{\text{DACKx}}$ have to be inactive during an $\overline{\text{INTA}}$ cycle. If pin $\overline{\text{INTA}}$ is not used, it has to be tied to V_{DD}.

Pin Definitions and Function (cont'd)

Pin No.	Symbol	Input (I) Output (O)	Function
101 100 99	IE0 IE1 IE2	I/O I I	**Interrupt Enable 0, 1, 2** The function depends on the selected cascading mode: **Slave mode:** IE0-2 are inputs. Interrupt acknowledge is accepted if an interrupt signal has been generated and the slave address provided via IE0, IE1, IE2 corresponds to the programmed value (IPC register). If not used, IE0-2 should be tied to GND and the slave address should be set to "0" (e.g. single device application). **Daisy Chaining mode:** IE0 is output, IE1 is input. IE2 is unused and has to be fixed to "0" or "1". Normally, IE1 is connected to the IE0 pin of devices with higher priority. If not used, IE1 has to be fixed to "1". If IE1 is reset ("0") – the IE0 output is reset immediately, – an active INT signal will be prohibited or aborted. – INT is hold inactive unconditionally as long as IE1 is "0". As long as $\overline{\text{INTA}}$ input is inactive, IE1 = "1" enables INT signal generation. If INT goes active, pin IE0 immediately is set to "0". Interrupt acknowledge is accepted if the Interrupt Enable input IE1 is active during the following $\overline{\text{INTA}}$ cycle. During this cycle, and additionally till the end of the second $\overline{\text{INTA}}$ cycle in Siemens/Intel bus mode, triggering of INT signal generation is prohibited, i.e. no interrupt will be generated while (another) device is under service. This is valid even for devices with higher priority. Pin IE0 returns to active state (logical "1") when INT is deactivated and IE1 input is high.

Pin Definitions and Function (cont'd)

Pin No.	Symbol	Input (I) Output (O)	Function
1 2 3 4 5 6 7 8	DRT0 DRT1 DRT2 DRT3 DRT4 DRT5 DRT6 DRT7	O	**DMA Request Transmitter** (Channel 0 ... 7) The transmitter on a serial channel requests a DMA transfer by activating the corresponding DRT line. The request remains active as long as the corresponding Transmit FIFO requires data transfers. The amount of data bytes to be transferred from the system memory to the ESCC8 serial channel (= byte count) must be written first to the XBCH, XBCL registers. Always blocks of data (n x 32 bytes + REST, n = 0, 1,…) are transferred till the Byte Count is reached. DRTn is deactivated with the beginning of the last write cycle.
160 159 158 157 156 155 154 153	DRR0 DRR1 DRR2 DRR3 DRR4 DRR5 DRR6 DRR7	O	**DMA Request Receiver** (Channel 0 … 7) The receiver on a serial channel requests a DMA transfer by activating the corresponding DRT line. The request remains active as long as the corresponding Receive FIFO requires data transfers, thus always blocks of data are transferred. DRRn is deactivated immediately following the falling edge of the last read cycle.

Pin Definitions and Function (cont'd)

Pin No.	Symbol	Input (I) Output (O)	Function
73 74 75 76 77 78 79 80	$\overline{\text{DACK0}}$ $\overline{\text{DACK1}}$ $\overline{\text{DACK2}}$ $\overline{\text{DACK3}}$ $\overline{\text{DACK4}}$ $\overline{\text{DACK5}}$ $\overline{\text{DACK6}}$ $\overline{\text{DACK7}}$	I	**DMA Acknowledge** (Channel 0 … 7) A low signal on these pins informs the ESCC8 that the requested DMA cycle controlled via DRT or DRR of the corresponding channel is in progress, i.e. the DMA controller has achieved bus mastership from the CPU and will start data transfer cycles (either write or read). In conjunction with a read or write operation these inputs serve as Access Enable (similar to $\overline{\text{CS}}$) to the respective FIFOs. If $\overline{\text{DACK}}$ is active, the input to pins A1…A8 is ignored and the FIFOs are implicitly selected. A0 and $\overline{\text{BHE}}$/$\overline{\text{BLE}}$ are used to select byte or word access. If not used, these pins must be connected to V_{DD}.
14 16 20 22 112 110 108 106	RXD0 RXD1 RXD2 RXD3 RXD4 RXD5 RXD6 RXD7	I (O/oD)	**Receive Data** (Channel 0 … 7) Serial data is received on these pins. May be switched to T×D function via bit CCR2.SOC1.
49 50 51 52 69 70 71 72	RXCLK0 RXCLK1 RXCLK2 RXCLK3 RXCLK4 RXCLK5 RXCLK6 RXCLK7	I	**Receive Clock** (Channel 0 … 7) The function of these pins depends on the selected clock mode. In each channel, R×CLKn may supply either – the receive clock (clock mode 0), or – the receive and transmit clock (clock mode 1, 5), or – the clock input for the baud rate generator (clock mode 2, 3).

Pin Definitions and Function (cont'd)

Pin No.	Symbol	Input (I) Output (O)	Function
41 42 43 44 45 46 47 48	$\overline{\text{RTS0}}$ $\overline{\text{RTS1}}$ $\overline{\text{RTS2}}$ $\overline{\text{RTS3}}$ $\overline{\text{RTS4}}$ $\overline{\text{RTS5}}$ $\overline{\text{RTS6}}$ $\overline{\text{RTS7}}$	O	**Request to Send** (Channel 0 … 7) When the $\overline{\text{RTS}}$ bit in the MODE register is set, the $\overline{\text{RTS}}$ signal goes low. When the $\overline{\text{RTS}}$ bit is reset, the signal goes high if the transmitter has finished and there is no further request for a transmission. In bus configuration, $\overline{\text{RTS}}$ can be programmed via CCR2 to: – go low during the actual transmission of a frame shifted by one clock period, excluding collision bits. – go low during reception of a data frame. – stay always high ($\overline{\text{RTS}}$ disabled).
134 133 132 131 130 129 128 127	$\overline{\text{CTS0}}$/CxD0 $\overline{\text{CTS1}}$/CxD1 $\overline{\text{CTS2}}$/CxD2 $\overline{\text{CTS3}}$/CxD3 $\overline{\text{CTS4}}$/CxD4 $\overline{\text{CTS5}}$/CxD5 $\overline{\text{CTS6}}$/CxD6 $\overline{\text{CTS7}}$/CxD7	I	**Clear to Send** (Channel 0 … 7) A low on the $\overline{\text{CTSn}}$ input enables the respective transmitter. Additionally, an interrupt may be issued if a state transition occurs at the $\overline{\text{CTSn}}$ pin (programmable feature). If no "Clear To Send" function is required, the $\overline{\text{CTSn}}$ inputs can be directly connected to GND. **Collision Data** (Channel 0 … 7) In a bus configuration, the external serial bus must be connected to the corresponding C×D pin for collision detection.
126 125 124 123 122 121 120 119	CD0 CD1 CD2 CD3 CD4 CD5 CD6 CD7	I	**Carrier Detect** (Channel 0 … 7) The function of this pin depends on the selected clock mode. It can supply: – either a modem control or a general purpose input (clock modes 0,2,3,4,6,7). If auto-start is programmed, it functions as a receiver enable signal. – or a receive strobe signal (clockmode 1). – or a frame synchronization signal in time-slot oriented operation mode (clock mode 5). Additionally, an interrupt may be issued if a state transition occurs at the CDn pin (programmable feature).

Pin Definitions and Function (cont'd)

Pin No.	Symbol	Input (I) Output (O)	Function
13 15 19 21 113 111 109 107	TXD0 TXD1 TXD2 TXD3 TXD4 TXD5 TXD6 TXD7	O/oD	**Transmit Data** (Channel 0 … 7) Transmit data is shifted out via these pins. They can be programmed to be either a push-pull or open drain output to support bus configurations. **Note:** Pin TxD is "or" ed with pin $\overline{\text{RTS}}$ if NRZI encoding and IDLE as Interframe Time Fill are selected and bit MODE.RTS is reset. May be switched to RxD function via bit CCR2.SOC1.
9 10 11 12 118 117 116 115	TXCLK0 TXCLK1 TXCLK2 TXCLK3 TXCLK4 TXCLK5 TXCLK6 TXCLK7	I/O	**Transmit Clock** (Channel 0 … 7) The function of this pin depends on the selected clock mode and the value of the SSEL bit (CCR2 register). For detailed information about the clock modes **refer to chapter 2**. If programmed as an input, this pin supplies either – the transmit clock for the channel (clock mode 0, 2, 6; SSEL bit in CCR2 is reset), or – a transmit strobe signal for the channel (clock mode 1). If programmed as an output (bit CCR2.TOE is set), this pin supplies either – the transmit clock for the channel which is generated ● either from the baud rate generator (clock mode 2, 3, 6, 7; SSEL bit in CCR2 is set), ● or from the DPLL circuit (clock mode 3, 7; SSEL bit in CCR2 is reset) ● or from the crystal oscillator (clock mode 4), ● or an active-low tri-state control signal marking the programmed transmit time-slot (clock mode 5) if bit CCR2.TOE is set.

Pin Definitions and Function (cont'd)

Pin No.	Symbol	Input (I) Output (O)	Function
63 64	XTAL1 XTAL2	I (O)	**Crystal Connection** If the internal oscillator is used for clock generation the external crystal has to be connected to these pins. Moreover, XTAL1 may be used as common clock input for all channels provided by an external clock generator. All versions: common use for both channels in clock modes 4,6,7. Version 2 upward: additionally used in clock mode 0b and for master clock applications.
152 → 145 53 → 60 142 → 135 65 → 68	PA0 … 7 PB0 … 7 PC0 … 7 PD0 … 3	I/O	**Parallel Port (Port A,B,C,D)** Four general purpose bi-directional parallel ports (port A,B,C: 8 bit; port D: 4 bit). Every pin is individually programmable to operate as an output or an input (Port Configuration Register PCRA,B,C,D). – If defined as output, the state of the pin is directly controlled via the microprocessor interface(PortValueRegister PVRA,B,C,D) – If defined as input, its state can be read via PVRA,B,C,D. All changes may be indicated via an interrupt status (Port Interrupt Mask register PIMA,B,C,D, Port Interrupt Status register PISA,B,C,D, interrupt is output on pin INT).
18, 32, 61, 102, 114, 143	V_{SS}	I	**Ground** (0 V) For correct operation, **all six** pins have to be connected to ground.
17, 31, 62, 103, 144	V_{DD}	I	**Positive Power Supply** (5 V) For correct operation, **all five** pins have to be connected to positive power supply.
83	TEST	I	**Test Input** This pin always has to be connected to V_{SS}. (Test input for the manufacturer)

1.2 Logic Symbol

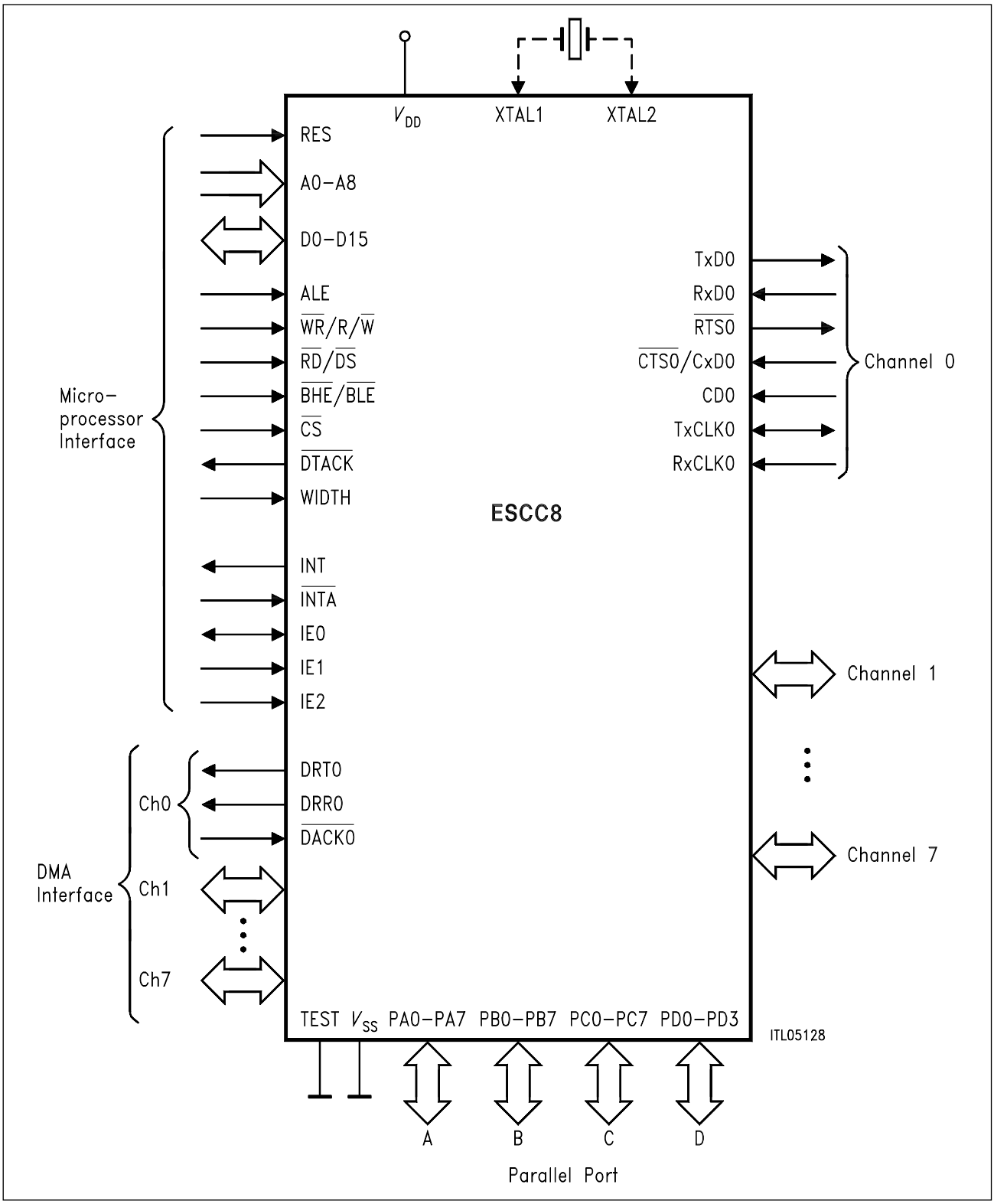


Figure 1
ESCC8 Logic Symbol

1.3 Functional Block Diagram

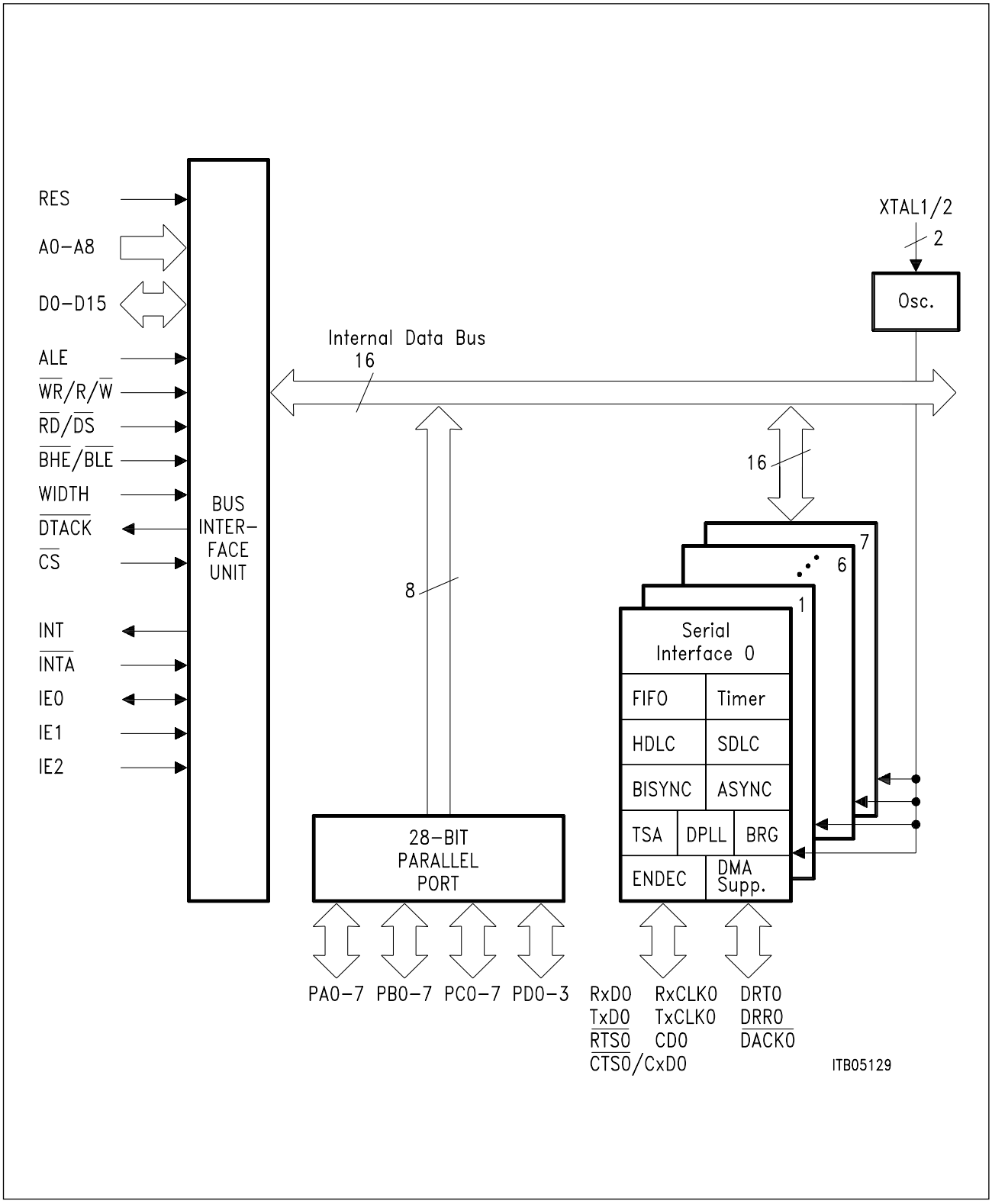


Figure 2
Functional Block Diagram SAB 82538

The ESCC8 (SAB 82538) comprises eight completely independent full-duplex serial interfaces which support HDLC/SDLC, BISYNC and ASYNC protocols. Layer-1 functions are performed by means of internal oscillator, Baud Rate Generator (BRG), Digital Phase Locked Loop (DPLL), and Time-Slot Assignment circuits (TSA, only available for version SAB 82538H-10). Encoding / decoding of serial data can be done by using NRZ, NRZI, FM0, FM1, and Manchester encoding schemes.

An 28-bit universal port is provided which can be used for additional modem control lines or for general I/O purposes.

Associated with each serial channel is a set of independent command and status registers and 64-byte deep FIFOs for transmit and receive direction. Access is done via the flexible 8/16-bit microprocessor interface. DMA capability has been added to the ESCC8 by means of a 16-channel DMA interface with one DMA request line for each transmitter and receiver of both channels. The interrupt structure of ESCC8 supports interrupt driven systems using interrupt polling, daisy chaining or interrupt vector control.

1.4 System Integration

1.4.1 General Aspects

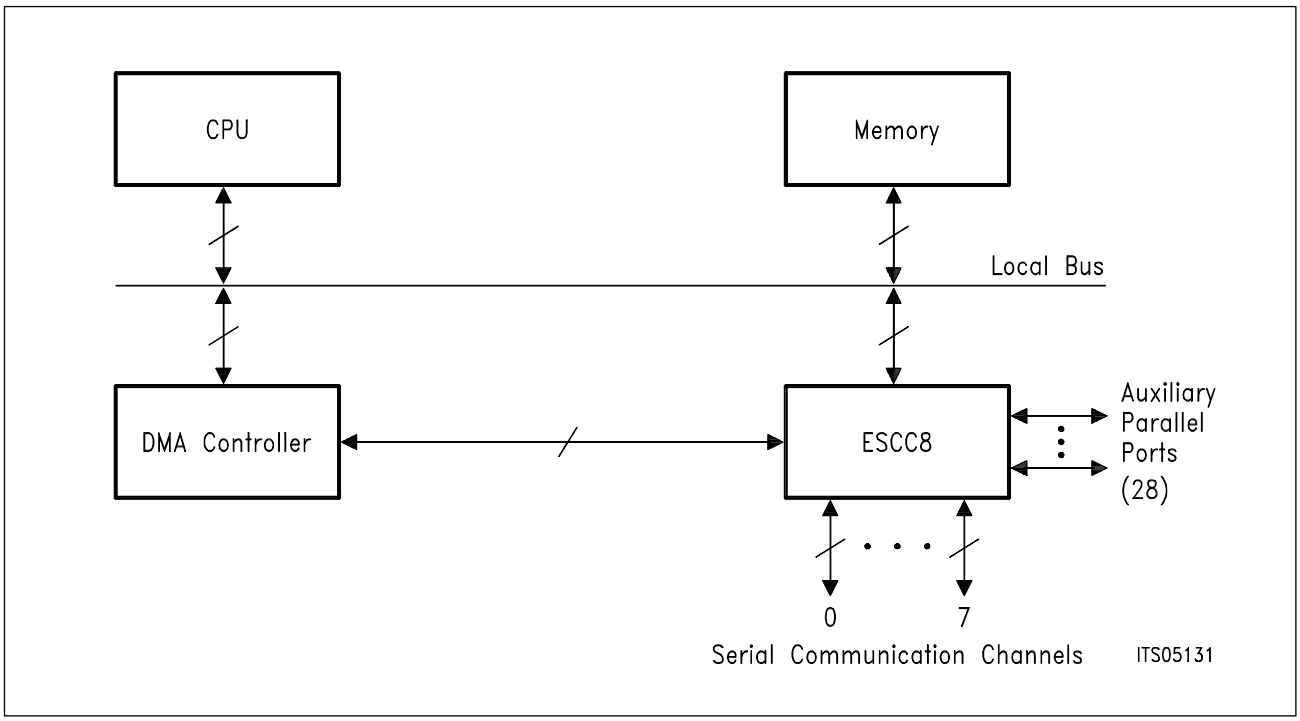


Figure 3
General System Integration of ESCC8

Figure 3 gives a general overview of system integration of ESCC8.

The ESCC8's bus interface consists of an 8/16-bit bidirectional Data bus (D0-D15), nine Address Line inputs (A0-A8), three control inputs ($\overline{RD}$ / $\overline{DS}$, $\overline{WR}$ / R/$\overline{W}$, $\overline{CS}$), five signals for interrupt support (INT, $\overline{INTA}$, IE0-2) and a 16-channel DMA interface. Mode input pins (strapping options) allow the bus interface to be configured for 8/16-bit bus width and for either Siemens/Intel or Motorola environment.

Generally, there are two types of transfers occurring via the system bus:

– Command/Status transfers, which are always controlled by the CPU. The CPU sets the operation mode (Initialization), controls function sequences and gets status information by writing or reading the ESCC8's registers (via $\overline{CS}$, $\overline{WR}$ or $\overline{RD}$, and register address via A0-A8, $\overline{BHE}$).
– Data Transfers, which are effectively performed by DMA without CPU interaction using the ESCC8's DMA interface (DMA Mode). Optionally, interrupt controlled data transfer can be done by the CPU (Interrupt Mode).

1.4.2 Environment

1.4.2.1 ESCC8 with SAB 80188 Microprocessor

A system with minimized additional hardware expense can be build up with a SAB 80188 microprocessor as shown in **figure 4**.

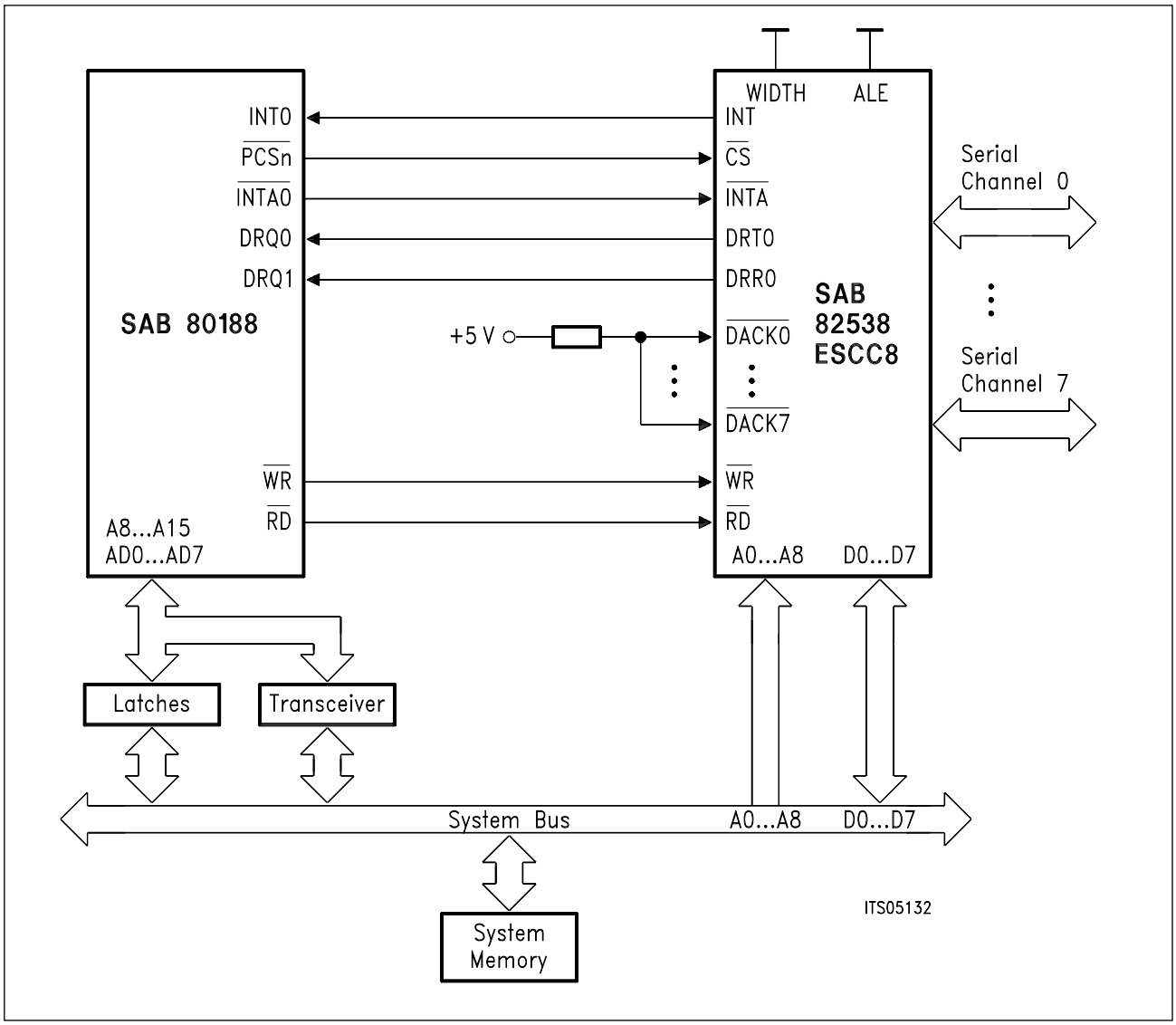


Figure 4
ESCC8 with SAB 80188 CPU

The ESCC8 is connected to the demultiplexed system bus. Data transfer for one serial channel can be done by the 2-channel on-chip DMA controller of the 80188, the other channels are serviced by interrupt. Since the 80188 does not provide DMA Acknowledge outputs, data transfer from/to ESCC8 is controlled via $\overline{CS}$, $\overline{RD}$ or $\overline{WR}$ Address information (A1 … A8) and the $\overline{DACK0\text{-}7}$ inputs are not used. A0 and $\overline{BHE}$/$\overline{BLE}$ are used to select byte or word access.

This solution supports applications with a high speed data rate in one serial channel with minimum hardware expense making use of the on-chip peripheral functions of the 80188 (chip select logic, interrupt controller, DMA controller).

1.4.2.2 ESCC8 with 80386

In high-performance 32-bit systems based on an 80386 microprocessor a separate control logic (e.g. sequencer PALs) is normally provided to generate all necessary control signals for interfacing to I/O devices. Address and data lines are buffered via latches or transceivers. An interface to ESCC8 is for this case sketched in **figure 5**.

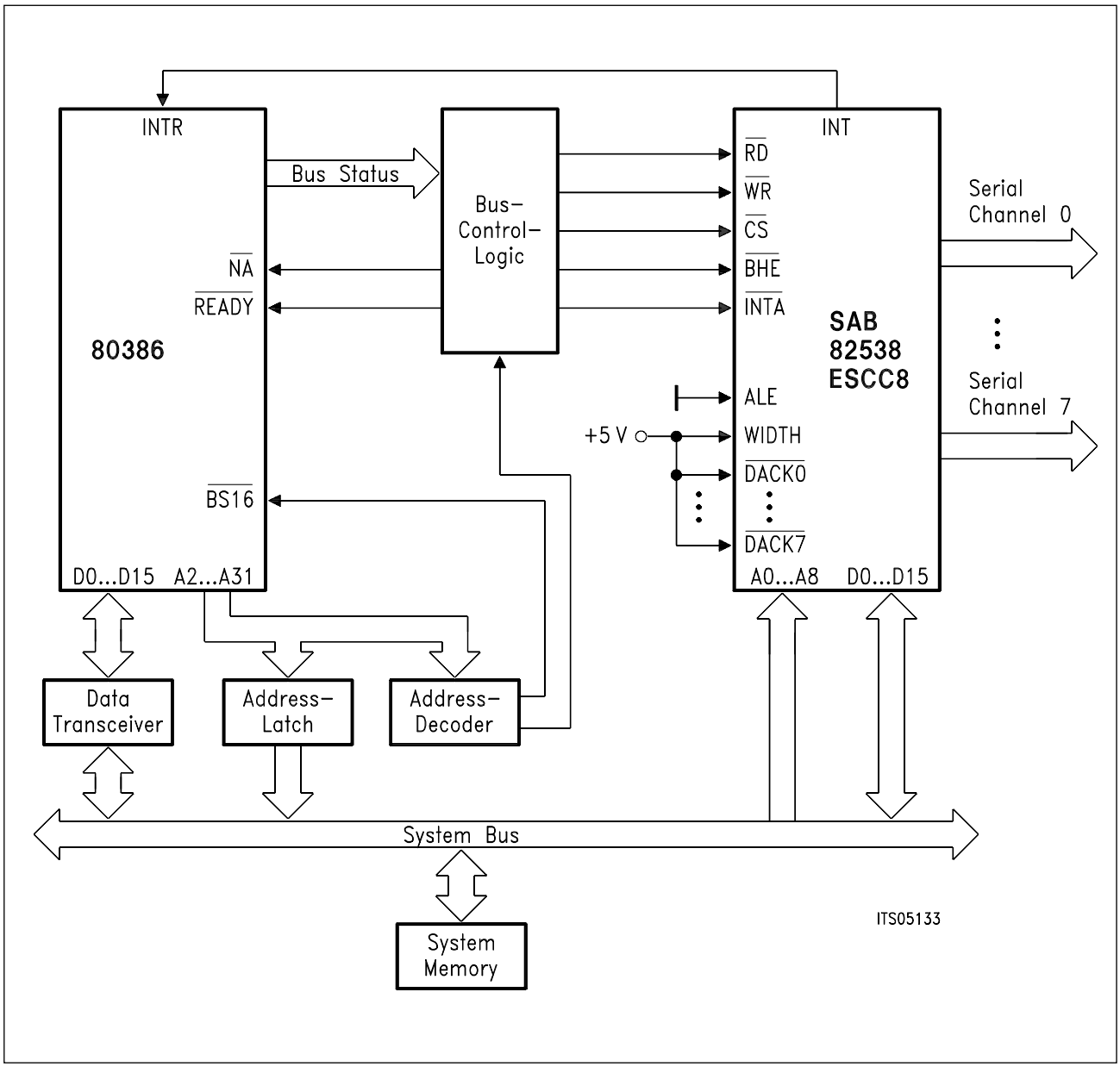


Figure 5
ESCC8 with 80386 µP

1.4.2.3 ESCC8 with MC 68020, 68030

Figure 6 gives an example of interfacing the ESCC8 to a 32-bit Motorola microprocessor. Some glue logic is necessary. The signal BUS LOW ENABLE (BLE) has to be decoded out of transfer size information (SIZ0,1) and A0. The ESCC8 interface logic has to respond as a 16-bit peripheral (DSACK1,0 = 01_H) during register access and interrupt acknowledge cycles.

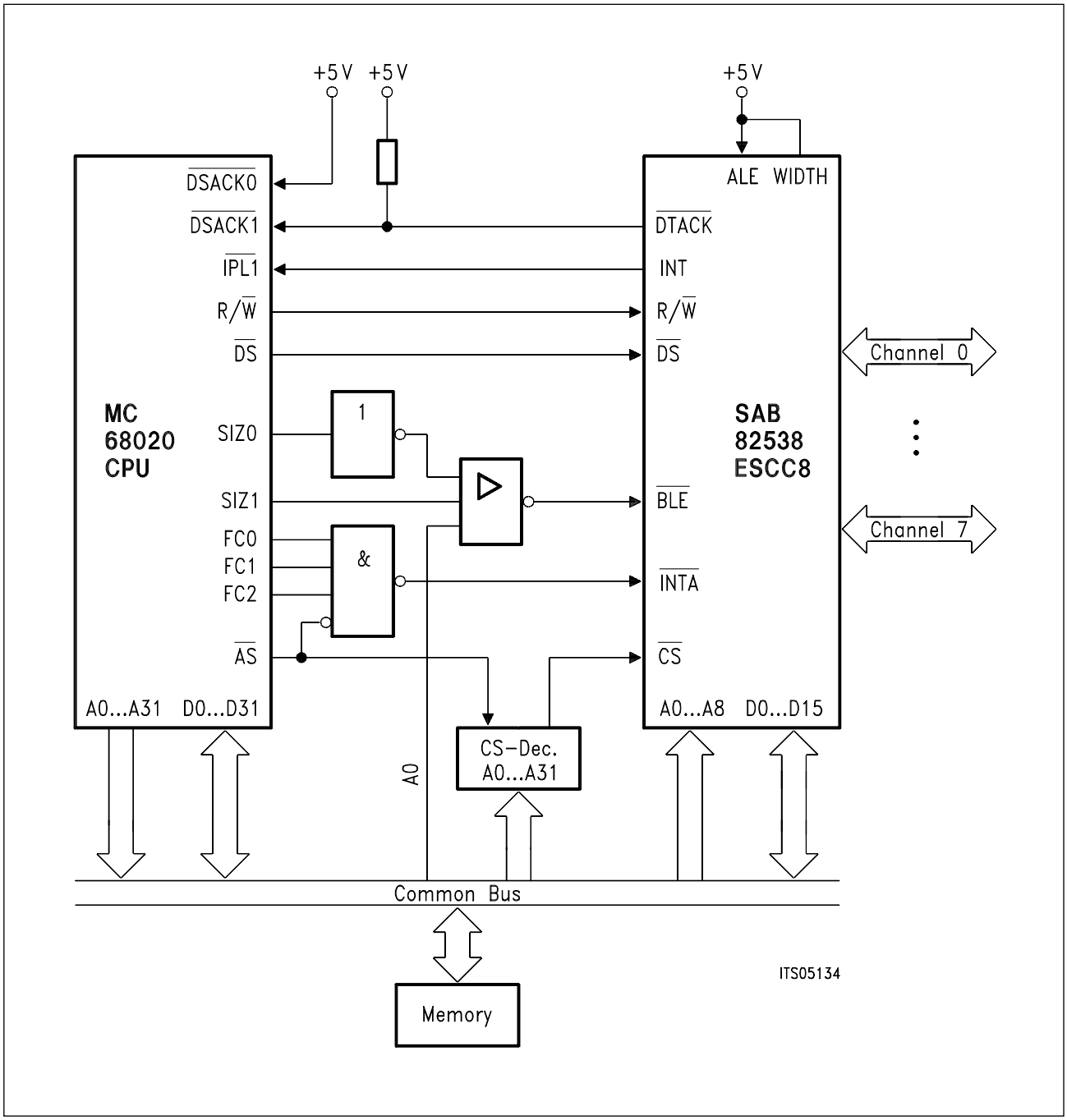


Figure 6
ESCC8 with 68020 µP

1.4.2.4 Interrupt Cascading

The ESCC8 supports two cascading schemes which can be selected by programming the IPC register:

Slave Mode

Interrupt outputs of several devices (slaves) are connected to a priority resolving unit (e.g. interrupt controller). The slave which is selected for the interrupt service routine is addressed via special address lines during the interrupt acknowledge cycle. For this application the ESCC8 offers three Interrupt Enable inputs (IE0, IE1,IE2) and a programmable 3-bit slave ID.

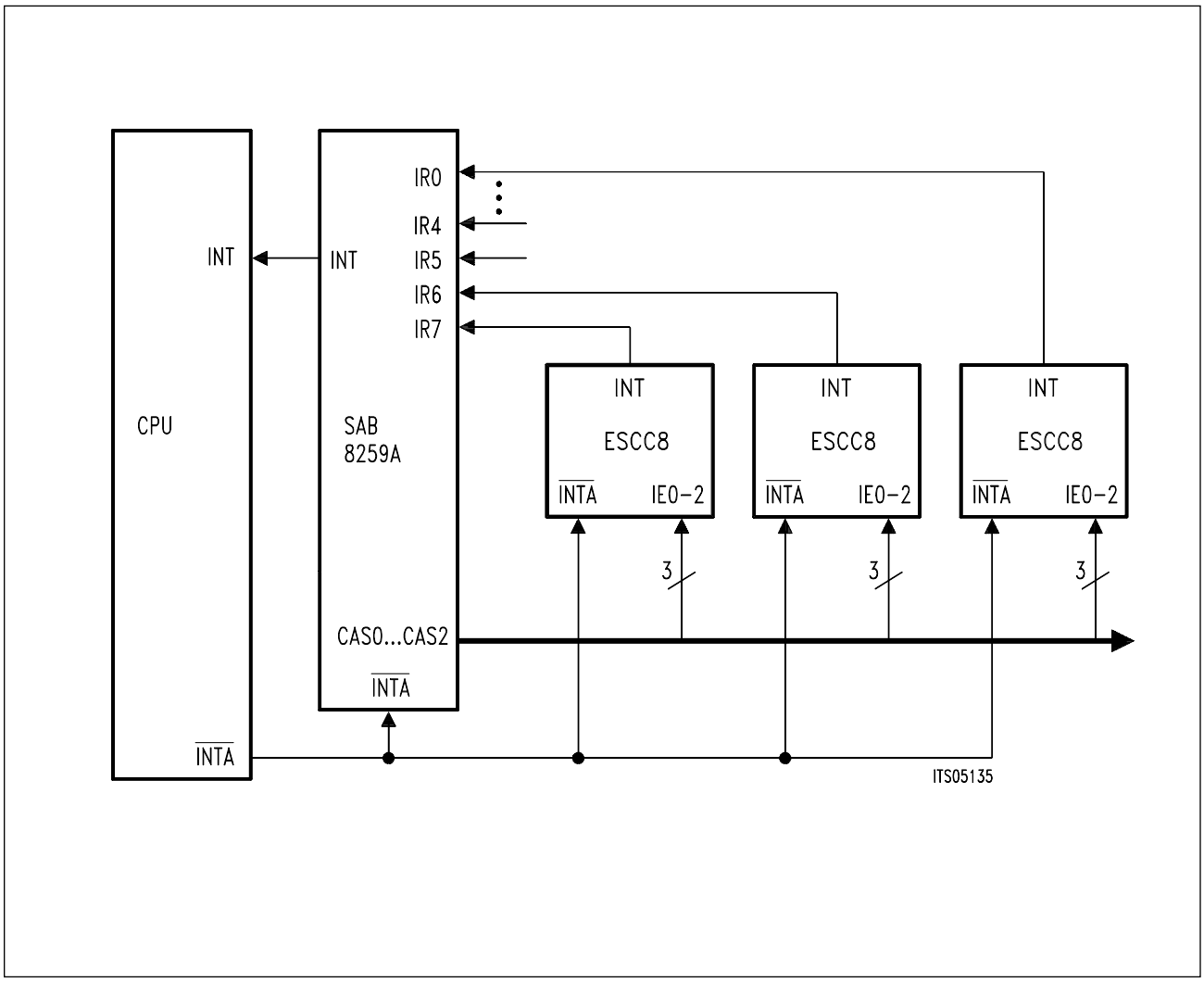


Figure 7
Interrupt Cascading (Slave Mode) in Intel Bus Mode

For Intel type microprocessor systems the 2-cycle interrupt acknowledge scheme is supported (’86 mode).

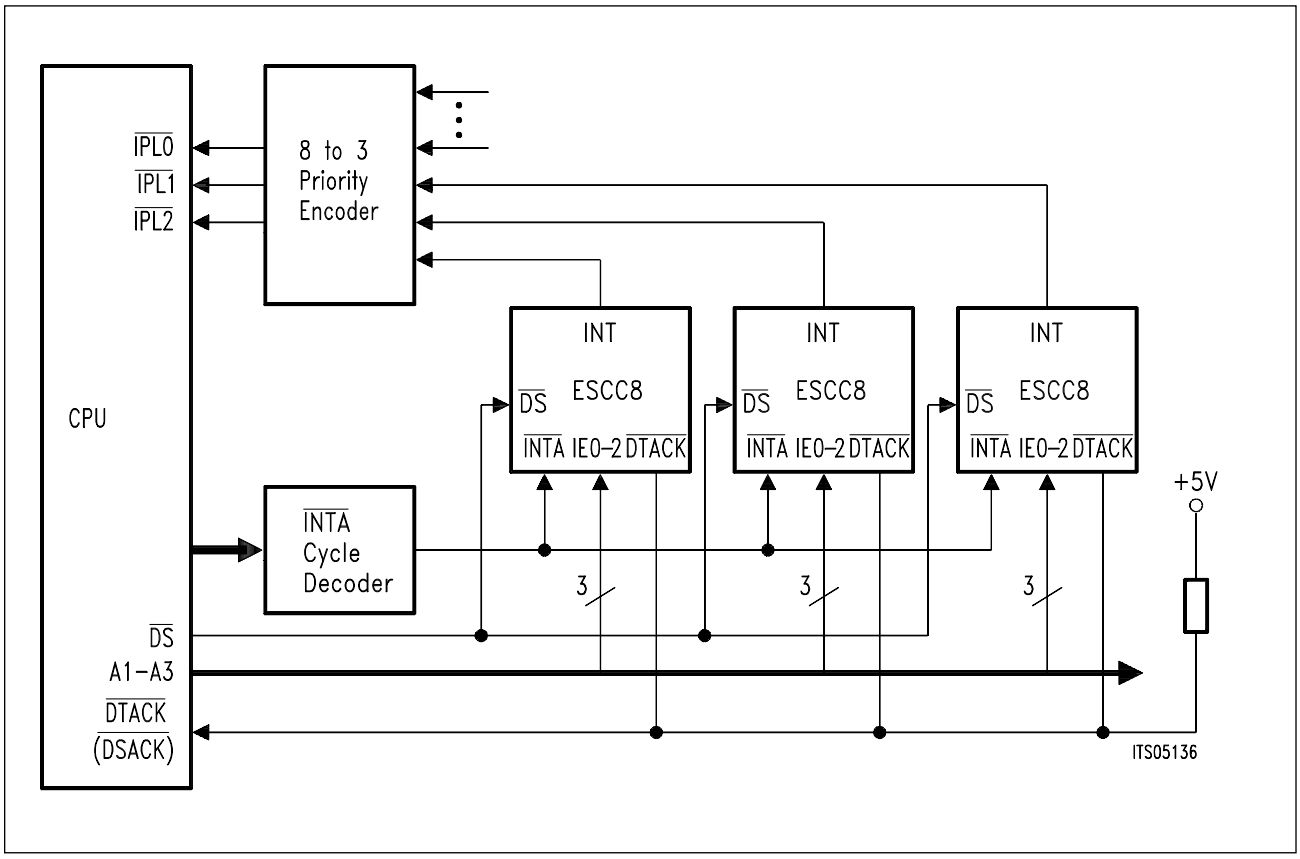


Figure 8
Interrupt Cascading (Slave Mode) in Motorola Bus Mode

Daisy Chaining

If selected via IPC register the Interrupt Enable pins IE0, IE1 are used for building a Daisy Chain by connecting the Interrupt Enable Output (IE0) of the higher priority device to the Interrupt Enable Input (IE1) of the lower priority device. The highest priority device has IE1 pulled high (**refer to figure 9 and 10**). IE2 is unused and has to be fixed to “0” or “1”.

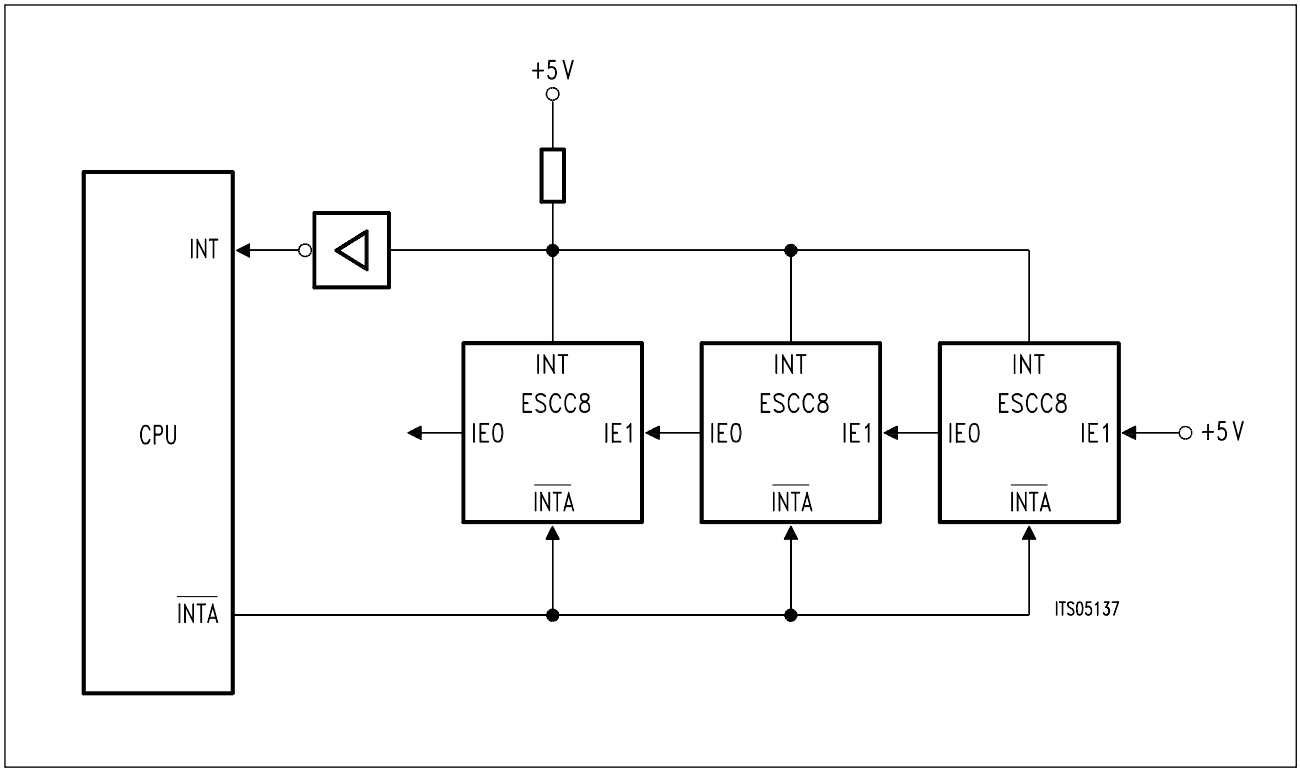


Figure 9
Interrupt Cascading (Daisy Chaining) in Intel Bus Mode

For Intel type microprocessor systems the 2-cycle interrupt acknowledge scheme is supported (’86 mode). Maximum available settling time for the chain: from the beginning of the first $\overline{\text{INTA}}$ cycle to the beginning of the second.

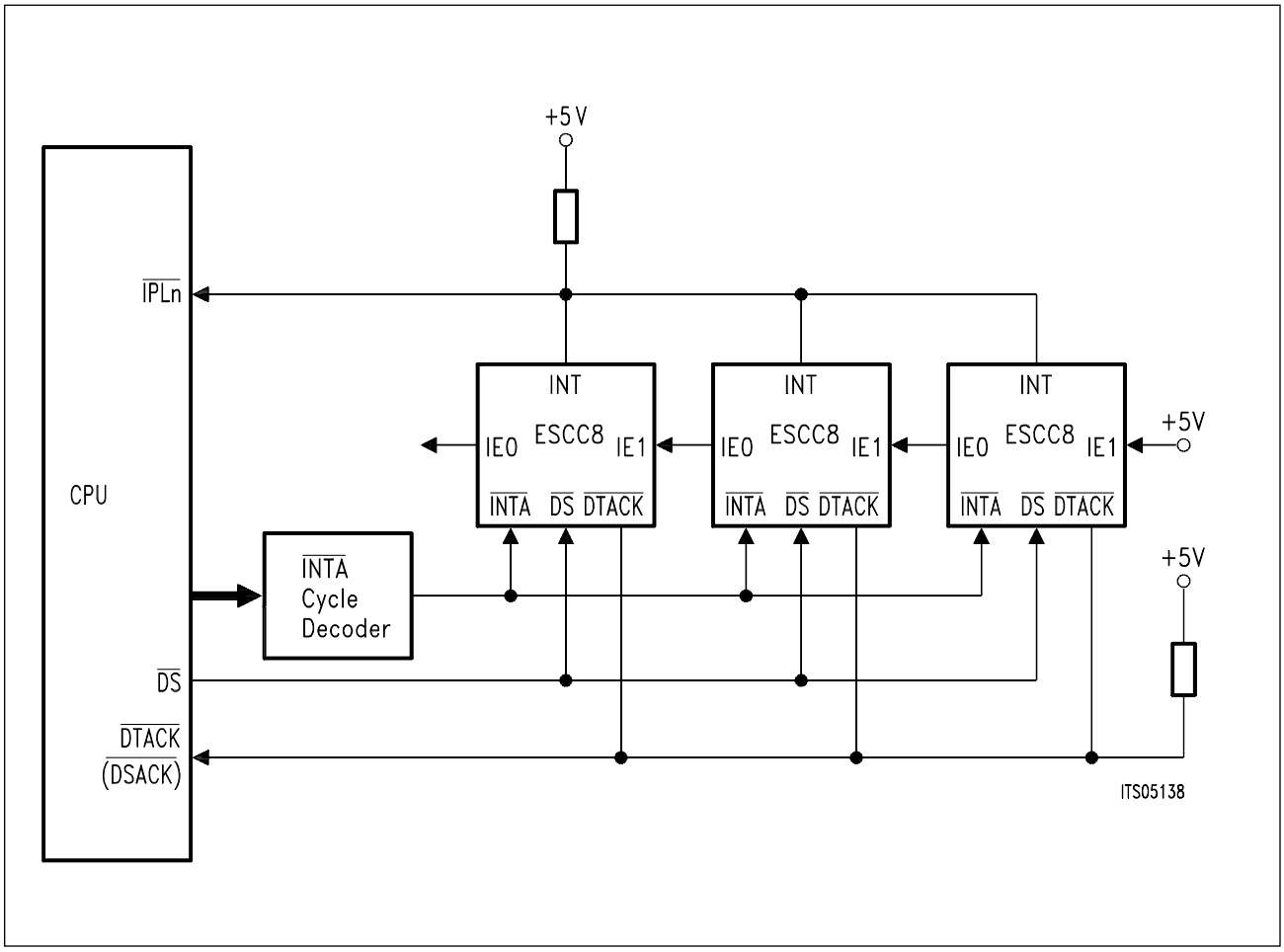


Figure 10
Interrupt Cascading (Daisy Chaining) in Motorola Bus Mode

For Motorola type microprocessor systems the maximum available settling time for the chain is much shorter: from the beginning of the $\overline{\text{INTA}}$ cycle to the falling edge of signal $\overline{\text{DS}}$.

2 Functional Description

2.1 General

The ESCC8 distinguishes itself from other communication controllers by its advanced characteristics. The most important are:

- Eight independent serial channels.
- Support of HDLC, SDLC, BISYNC/MONOSYNC and Asynchronous protocols.
- Support of layer-2 functions (HDLC mode).
  In addition to those bit-oriented functions commonly supported by HDLC controllers, such as bit stuffing, CRC check, flag and address recognition, the ESCC8 provides a high degree of procedural support.
  In a special operating mode (auto-mode), the ESCC8 processes the information transfer and the procedure handshaking (I- and S-frames of HDLC protocol) autonomously. The only restriction is that the window size (= number of outstanding unacknowledged frames) is limited to 1, which is sufficient for many applications. The communication procedures are mainly processed between the communication controllers and not between the attached processors. Thus the dynamic load on the CPU and the software expense is greatly reduced.
  The CPU is informed about the status of the procedure and has mainly to manage the receive and transmit data. In order to maintain cost effectiveness and flexibility, the handling of unnumbered (U) frames, and special functions such as error recovery in case of protocol errors, are not implemented in hardware and must be done by the user's software.
- Extended support of different link configurations.
  Besides the point-to-point configurations, the ESCC8 allows the implementation of point-to-multipoint or multi-master configurations without additional hardware or software expense.
  In point-to-multipoint configurations, the ESCC8 can be used as a master or as a slave station. Even when working as slave station, the ESCC8 can initiate the transmission of data at any time. An internal function block provides means of idle and collision detection and collision resolution, which are necessary if several stations start transmitting simultaneously. Thus, a multi-master configuration is also possible.
- Telecom specific features.
  In a special operating mode, the ESCC8 can transmit or receive data packets in one of up to 64 time-slots of programmable width (clock mode 5). Furthermore, the ESCC8 can transmit or receive variable data portions within a defined window of one or more clock cycles in conjunction with an external strobe signal (clock mode 1). These features make the ESCC8 suitable for applications using time division multiplex methods, such as time-slot oriented PCM systems or systems designed for packet switching.

- FIFO buffers for efficient transfer of data packets.
  A further speciality of ESCC8 are the 64-byte deep FIFO buffers used for the temporary storage of data packets transferred between the serial communications interface and the parallel system bus. Because of the overlapping input/output operation (dual-port behaviour), the maximum message length is not limited by the size of the buffer. The dynamic load of the CPU is drastically reduced by transferring the data packets block by block via Direct Memory Access supported by the ESCC8. The CPU only has to initiate the data transmission by the ESCC8 and determine the status in case of completed reception, but is not involved in data transfers.
- The 16-bit wide microprocessor interface enables high data throughput and offers a high flexibility for connection to both 8/16-bit Siemens/Intel and Motorola type microprocessor systems. Moreover, interrupt driven systems are supported by vectored interrupts and interrupt cascading capabilities.
- In addition to standard modem control lines associated with each of the serial channels, 28 universal ports are provided for applications related to or independent of the serial channels.

Link Configurations

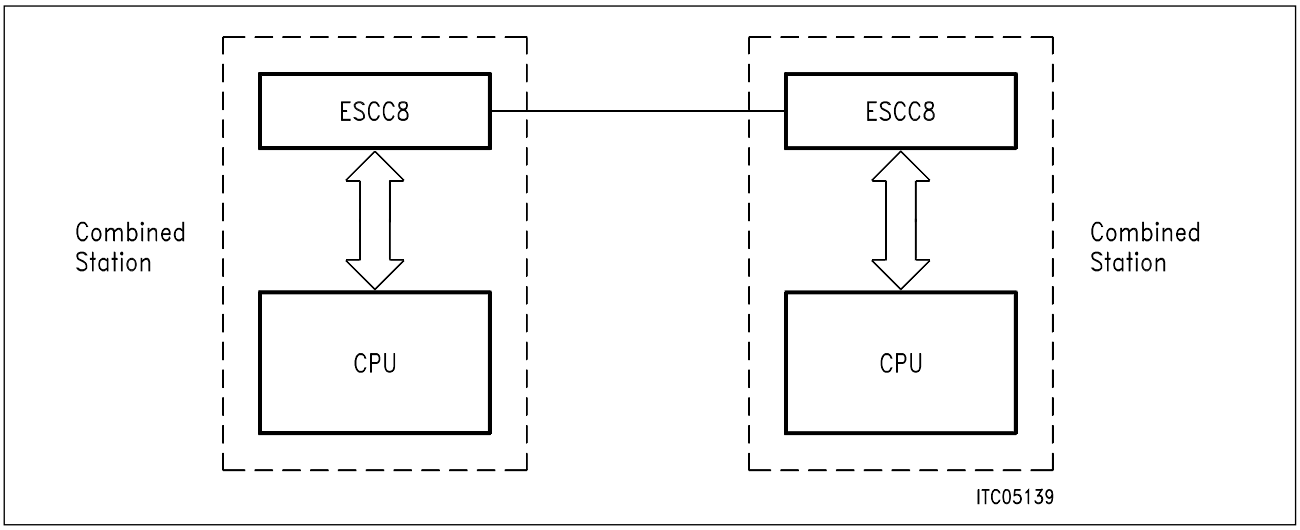


Figure 11
Point-to-Point Configuration

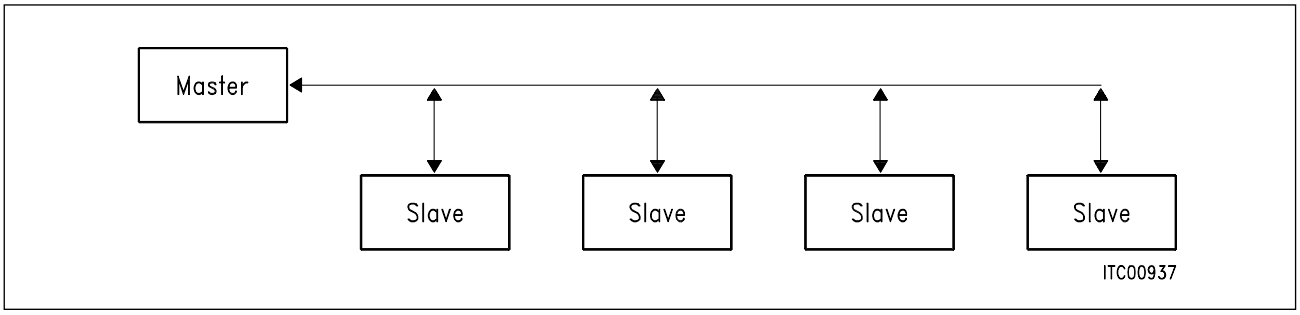


Figure 12
Point-to-Multipoint Configuration

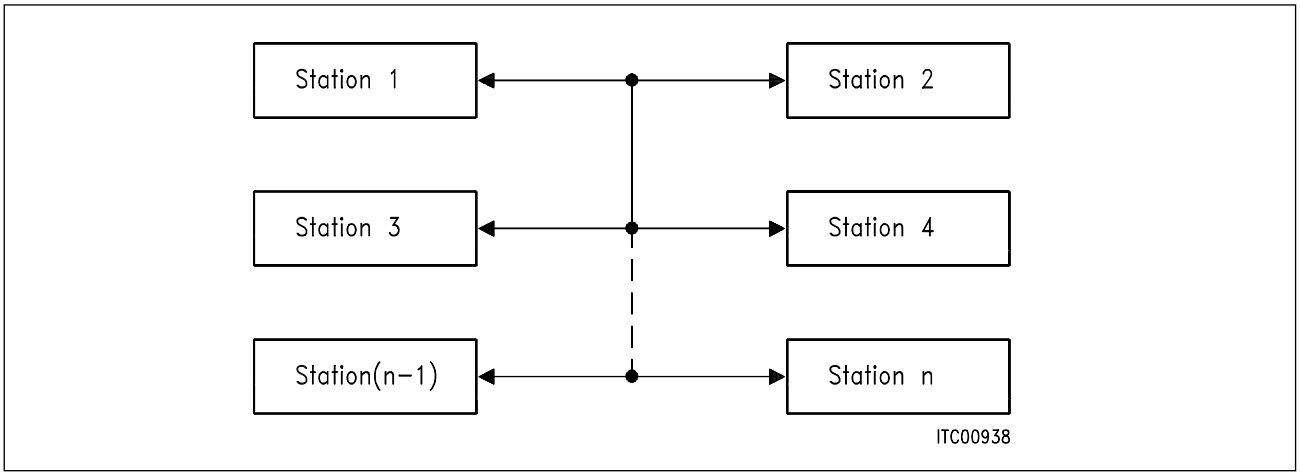


Figure 13
Multimaster Configuration

2.2 Microprocessor Interface

2.2.1 Register Set

The communication between the CPU and the ESCC8 is done via a set of directly accessible registers. The interface may be configured as Siemens/Intel or Motorola type with a selectable data bus width of 8 or 16 bits.

The CPU transfers data to/from the ESCC8 (via 64 byte deep FIFOs per direction and channel), sets the operating modes, controls function sequences, and gets status information by writing or reading control/status registers. All accesses can be done as byte or word accesses if enabled. If 16-bit bus width is selected, access to lower/upper part of the data bus is determined by address line A0 and signal $\overline{BHE}$/$\overline{BLE}$ as shown in **table 1 and 2**.

Mixed Byte/Word Access to the FIFOs

Reading from or writing to the internal FIFOs (RFIFO and XFIFO of each channel) can be done using a 8-bit (byte) or 16-bit (word) access depending on the selected bus interface mode. In version 1 of ESCC8, byte access in the case of 16-bit bus interface mode is allowed if not mixed with word accesses when reading from or writing to the same pool.

In version 2.x and upwards randomly mixed byte/word access to the FIFOs is allowed without any restrictions.

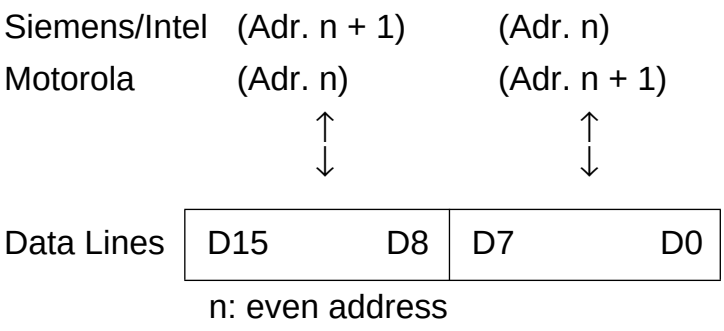
Table 1
Data Bus Access (16-Bit Intel Mode)

$\overline{BHE}$	A0	Register Access	ESCC8 Data Pins Used
0	0	FIFO word access Register word access (even addresses)	D0 – D15
0	1	Register byte access (odd addresses)	D8 – D15
1	0	Register byte access (even addresses)	D0 – D7
1	1	No transfer performed	None

Table 2
Data Bus Access (16-Bit Motorola Mode)

$\overline{\text{BLE}}$	A0	Register Access	ESCC8 Data Pins Used
0	0	FIFO word access Register word access (even addresses)	D0 – D15
0	1	Register byte access (odd addresses)	D0 – D7
1	0	Register byte access (even addresses)	D8 – D15
1	1	No transfer performed	None

The assignment of registers with even/odd addresses to the data lines in case of 16-bit register access depends on the selected microprocessor interface mode:



Complete information concerning register functions is provided in **chapter 4** – Detailed Register Description. The most important functions programmable via these registers are:

- Setting of serial, operating and clocking modes
- Layer-2 functions
- Data transfer modes (Interrupt, DMA)
- Bus mode
- DPLL mode
- Baud rate generator
- Test loop.

Each of the eight serial channels of ESCC8 is controlled via an identical, but totally independent register set (Channel 0…7). Functions which are common to or independent from all eight channels, e.g. interrupt information or universal port programming, are accessible via all register sets, which simplifies software development.

2.2.2 Data Transfer Modes

Data transfer between the system memory and the ESCC8 for all eight channels is controlled by either interrupts (Interrupt Mode), or independently from CPU, using the ESCC8's 16-channel DMA interface (DMA Mode).

After RESET, the ESCC8 operates in Interrupt Mode, where data transfer must be done by the CPU. The user selects the DMA Mode by setting the DMA bit in the XBCH register. All eight channels can be independently operated in either Interrupt or DMA Mode.

2.2.3 Interrupt Interface

Special events in the ESCC8 are indicated by means of a single interrupt output with programmable characteristics (open drain, push-pull; IPC register), which requests the CPU to read status information from the ESCC8, or, if Interrupt Mode is selected, to transfer data from/to ESCC8.

Since only one INT request output is provided, the cause of an interrupt must be determined by the CPU

- By evaluating the interrupt vector which is generated by ESCC8 during an interrupt acknowledge cycle (**Note**: For version 2 upward the format of the interrupt vector is changed to separate parallel port interrupts from channel assigned interrupts), and/or
- By reading the ESCC8's interrupt status registers (GIS, ISR0, ISR1, PIS).

The structure of the interrupt status registers is shown in **figure 14**.

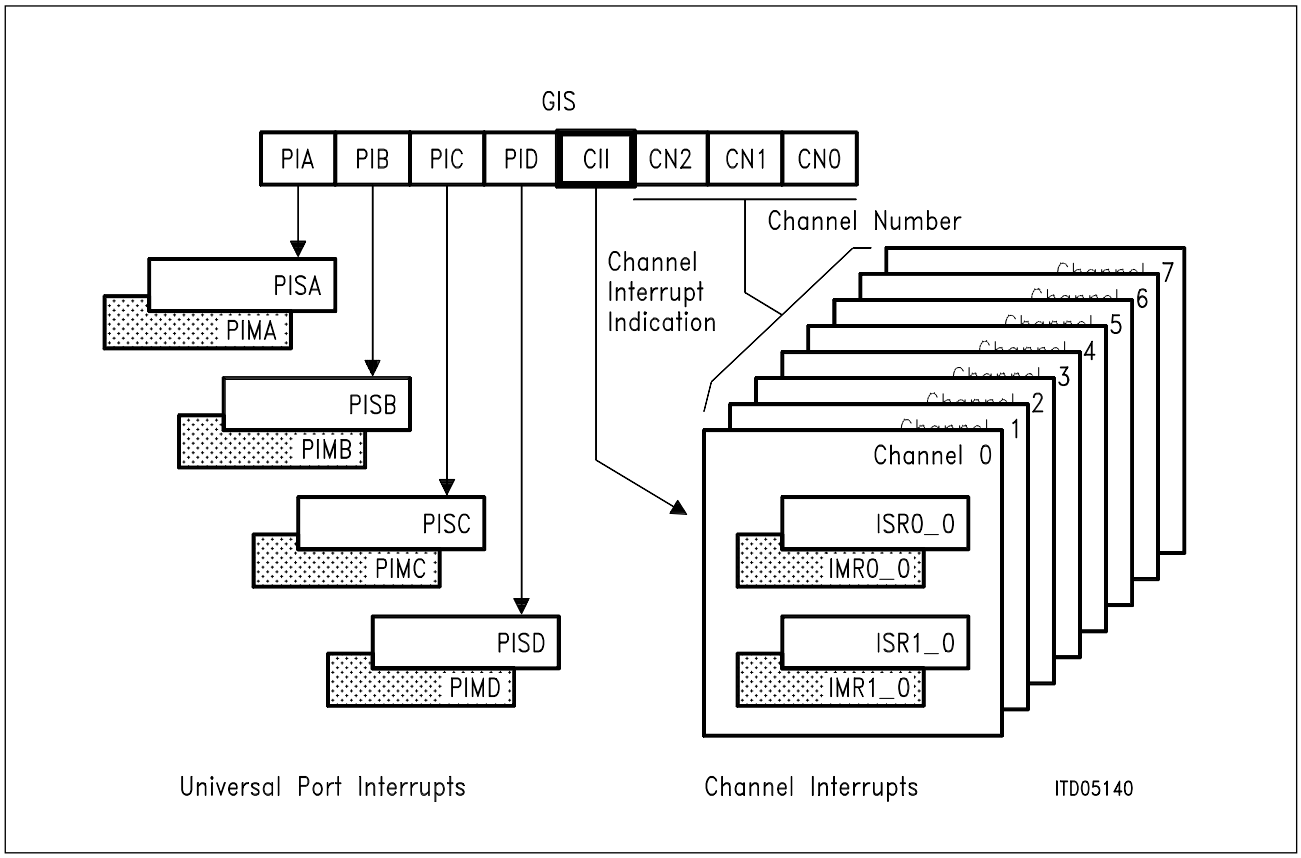


Figure 14
ESCC8 Interrupt Status Registers

Each interrupt indication of registers ISR0, ISR1 and PIS can be selectively masked by setting the corresponding bit in the corresponding mask registers IMR0, IMR1 and PIM. Use of these registers depends on the selected serial mode. GIS, the non-maskable Global Interrupt Status Register serves as pointer to pending channel related interrupts and universal port interrupts.

2.2.3.1 Priority Structure

The ESCC8 has a two level priority structure with different classifications (**refer to figure 15**):

First level: Type classification

All types of interrupt sources are divided into four groups with fixed priority levels.

Group 0 (highest priority): includes the Receive Pool Full interrupts of all channels.

…

Group 3 (lowest priority): refers to all other interrupt sources of all channels except those of group 0...2. Examples: Timer interrupt, Transmit FIFO Overflow interrupt,…

Second level: Channel classification

The second level considers the current priority of all channels. Especially for version 2 upward, selection is performed between

– Fixed priority level assignment (version 1: channel 0 has highest and channel 7 lowest priority; version 2 upward: channel with highest priority is selectable)
– Rotating priority level assignment, **all** channels (version 2 upward)
– Rotating priority level assignment, **7** channels with the selectable 8th channel fixed to highest priority (version 2 upward)

The priority level for each interrupt source results from the interrupt group (first level) and the current priority of the channel (second level). As mentioned above, in ESCC8 version 1 the priorities of the eight serial channel interrupts are fixed, with channel 0 having always the highest, channel 7 the lowest priority.

For ESCC8 Version 2 upward the priority levels are fixed or adjusted after an interrupt has been serviced, namely, the priorities are rotated cyclically so that the channel last serviced is assigned the lowest priority of all. **Port interrupts** are not affected by the priority rotation, and are **always** assigned **lowest priority.** The interrupt priority rotation mode is selectable via two control bits IVA.ROT and IPC.ROTM.

When an interrupt has been generated updating of all interrupt priorities takes place after the generated interrupt has been acknowledged, i.e.

– For interrupts that are unambiguously determined by the contents of the interrupt vector, after the end of a complete INTAQ cycle (1 or 2 pulse, whichever applies). Such interrupts are: Receive Data Interrupts RPF, RME/TCD and Transmit Data Interrupt XPR for all channels, or,
– Whenever the interrupt status which caused the currently active interrupt to be generated (i.e. one with currently the highest priority among all unmasked pending interrupts) is cleared by reading an interrupt status register ISR0_0...7, ISR1_0...7 or PISA...D. Reading other interrupt status registers may clear other pending interrupts (if any).

The following interrupt priority modes apply only to the channel identification (second level).

Interrupt priority mode 1: Fixed priority

After Reset the ESCC8 operates with fixed priority, i.e. the relative order of the priority levels assigned to the channels is fixed and there is no change after an interrupt has been serviced.

Apart from the change in the format of the interrupt vector (version 2 upward: channel 7 interrupts are separated from the parallel ports interrupts) version 2 is compatible with

version 1 if the optional features of version 2 are not enabled: channel 0 has highest and channel 7 lowest (channel) priority.

Note: Parallel ports have **always** lowest priority.

Version 2 upward provides dynamic adjustment of channel priorities by programming the “highest priority” channel. This is done via the IVA register. Although this register is unique, it is accessible via all eight channel assigned addresses. Selection of the “highest priority” channel is simply done with every write access to the IVA register in conjunction with the channel assigned IVA register address:

IVA Register Address:	Highest Priority Channel
38_H	0
78_H	1
$B8_H$	2
$F8_H$	3
138_H	4
178_H	5
$1B8_H$	6
$1F8_H$	7

The priority level becomes valid with the end of the write access to the IVA register (rising edge of $\overline{WR}$ or $\overline{DS}$, whichever applies) and remains stable until a new write access to this register occurs.

Note:

– The sequence of the channels remains unchanged. Only the pointer to the channel with highest priority is influenced. Therefore, the priority levels of all other channels follow automatically.

– Parallel ports have **always** lowest priority.

If the state after Reset shall be unchanged but bits of the IVA register have to be set, the programming has to be done via IVA register address “38H” (channel 0).

Example:

Initially after Reset, the order of the channels with descending priority from left to right is as follows:

0 1 2 3 4 5 6 7 pp (pp = parallel port interrupt)

Supposed the IVA register is programmed via the address 138_H (channel 4) the channel priorities will be reordered as follows:

4 5 6 7 0 1 2 3 pp

Interrupt priority mode 2: Rotating priority of 8 channels

With IVA.ROT = 1 and IPC.ROTM = 0 the interrupt priority rotation mode is selected.

After an interrupt has been serviced the priorities of all eight channels are rotated cyclically so that the channel last serviced is assigned the lowest priority of all. The ESCC8 will adjust the priorities according to the following scheme.

Example:

Suppose the order of the channels is as follows with descending priority from left to right:

0 1 2 3 4 5 6 7 pp (pp = parallel port interrupt)

Suppose channel 4 requires interrupt service and no other channel / interrupt group with higher priority is or becomes active, so that channel 4 has the currently highest priority of all channels at the time when the interrupt vector is output. After the interrupt in question has been acknowledged, an automatic reordering of the channels (and of pending interrupts, if any) takes place so that channel 4 is given the lowest channel priority. The relative ordering of the channels remains the same:

5 6 7 0 1 2 3 4 pp

This interrupt priority rotation guarantees fair treatment of all the channels. A reordering of the interrupts takes also place when any other channel interrupt is acknowledged by reading a corresponding interrupt status register, so that the corresponding channel is put behind the others.

Note: Parallel ports have **always** lowest priority.

Interrupt priority mode 3: Rotating priority of 7 channels

With IVA.ROT = 1 and IPC.ROTM = 1 the priority adjustment is performed only on 7 channels while one channel is fixed to the highest priority level. As described in “Interrupt Priority Mode 1” for fixed priority scheme, selection of the “highest priority” channel is simply done with every write access to the IVA register in conjunction with the channel assigned IVA register address:

IVA Register Address:	Highest Priority Channel
38_H	0
78_H	1
$B8_H$	2
$F8_H$	3
138_H	4
178_H	5
$1B8_H$	6
$1F8_H$	7

If the highest priority channel generates an interrupt and gets serviced by reading the interrupt vector or the interrupt status register of that channel a reordering of the other channels will not take place. The dynamic adjustment of the channel priorities does not affect the interrupt group of an interrupt even if it is the highest priority channel.

Example:

Let the channel priorities be labeled as follows with channel “2” as fixed highest priority channel:

2 5 6 7 0 1 3 4 pp

in descending order. Supposing the channel labeled “7” generates an interrupt which is serviced because no unmasked higher priority is pending, the channels will be reordered as follows:

2 0 1 3 4 5 6 7 pp

Even if the same channel generates another interrupt, it will not be serviced before at least one other channel (if any) requesting service by that time. If the channel labeled “2” now generates an interrupt which gets serviced the order is to be the same as above:

2 0 1 3 4 5 6 7 pp

Note: Parallel ports have **always** lowest priority.

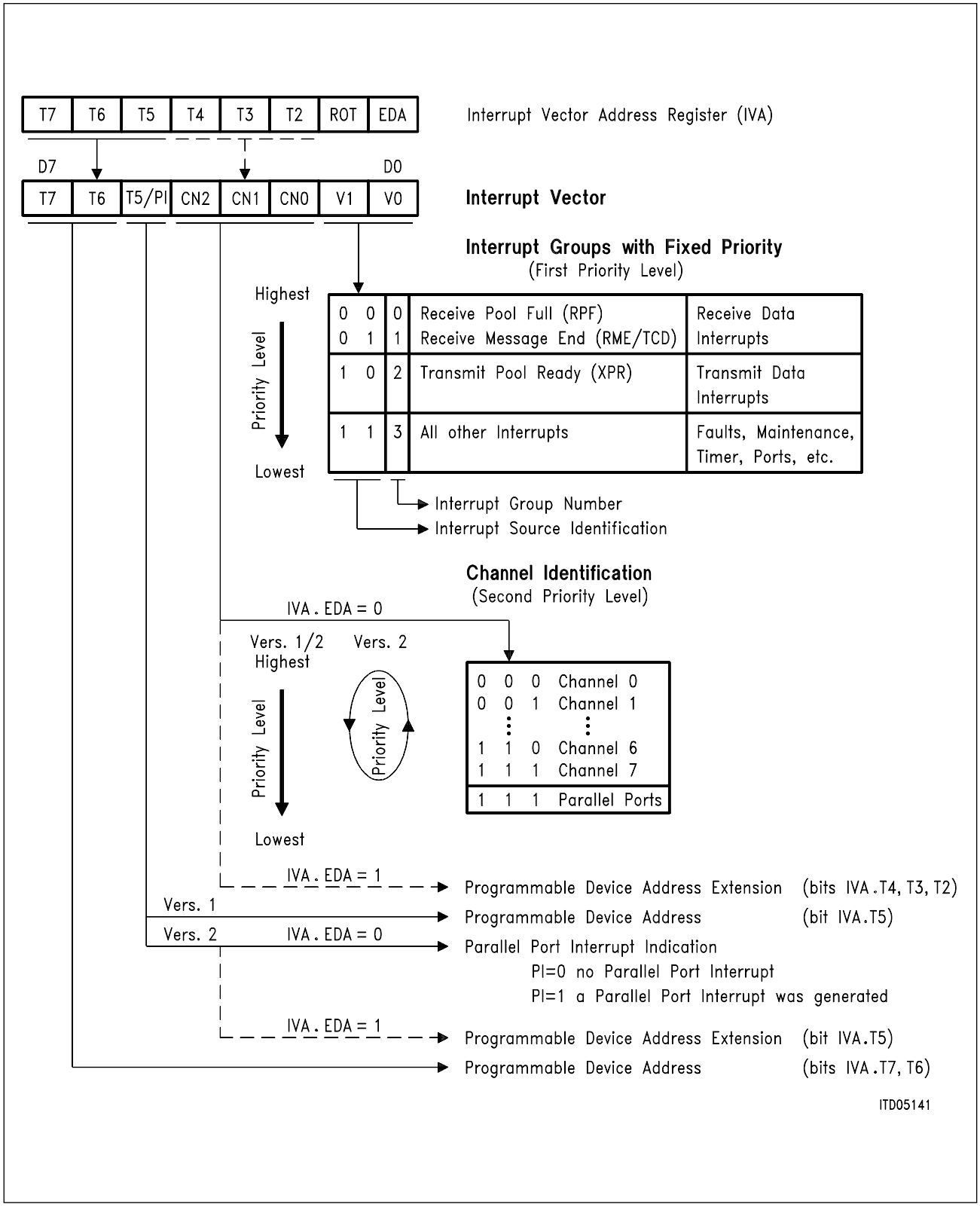


Figure 15
Structure of Interrupt Vector

Note: For IVA.EDA = 1 the interrupt vector format for version 2 is identical to version 1.

2.2.3.2 Interrupt Polling

After ESCC8 has requested an interrupt by activating its INT pin, the CPU must first read the Global Interrupt Status register GIS to identify parallel port and/or channel related interrupt indications:

- Channel related interrupts are indicated via bit GIS.CII (Channel Interrupt Indication) and the number of the requesting channel (GIS.CN2..0). After reading the assigned interrupt status registers ISR0_x and ISR1_x, the pointer in register GIS is cleared or updated if another channel requires interrupt service.
- The 28-bit universal port is divided into four groups (port A,B,C: 8 lines each, port D: 4 lines) which all have the same lowest interrupt priority. Pending interrupts are pointed out directly via GIS.PIA..D. Reading the assigned interrupt status registers (PISA..D) will reset the corresponding indication in GIS.

If **all** pending interrupts are acknowledged (GIS is reset), pin INT goes inactive.

2.2.3.3 Vectored Interrupt Structure

After ESCC8 has requested an interrupt by activating its INT pin, the system (CPU or peripherals) starts the interrupt acknowledge cycle by activating the $\overline{\text{INTA}}$ signal. If the Intel bus interface mode is selected, the two-pulse'86 mode is supported. In Motorola interface mode single pulse acknowledgement is implemented.

Interrupt acknowledge operation is determined by the selected interrupt cascading mode (IPC register) in conjunction with the Interrupt Enable Signals IE0, IE1 and IE2:

- **Slave Mode**
  The address of the slave under service has to be provided via inputs IE0, IE1 and IE2 during the valid $\overline{\text{INTA}}$ cycle. Interrupt acknowledge is accepted if this address corresponds to the programmed value (IPC register).

  If the ESCC8 is used in single device applications (no other device is present for interrupt cascading), IE0..2 have to be fixed to a defined level corresponding to the internally programmed address.
- **Daisy Chaining Mode**
  IE0 as Interrupt Enable Output and IE1 as Interrupt Enable Input are used to build a Daisy Chain (**refer to chapter 1.4**). Input IE2 is not used and has to be tied to V_{SS}. Interrupt acknowledge is accepted if IE1 is active during the valid $\overline{\text{INTA}}$ cycle. Output IE0 follows the IE1 input. Additionally, IE0 is reset when INT goes active.

  Activation of pin INT is prohibited
  – during $\overline{\text{INTA}}$ cycles
  – between the first and the second $\overline{\text{INTA}}$ cycle if Siemens/Intel mode is selected.

If interrupt acknowledge is accepted in one of the above modes, the ESCC8 generates an interrupt vector which is output on D0-D7 of the data bus independent of the selected bus interface mode (**refer to figure 15**).

Implementation of the interrupt service routines should consider the two selectable interrupt vector modes (bit IVA.EDA):

- **Interrupt vector mode 1** (EDA = 0)
  Interrupt vector includes: device address, version 2: parallel port indication, channel identification, interrupt group
  (+): fastest interrupt source identification (especially for interrupt groups 0..2).
  (–): interrupt vector table needs 32 (version 2: 64) entry points, placement of this entry field only in steps of 128 bytes (version 1: 3-bit device address; version 2: 2-bit device address), fastest service requires implementation of up to 32 (64) interrupt service routines.

  In case more than one source is active, the generated vector refers to the interrupt group with highest priority, and within a group to the requesting channel with highest priority. Although universal port interrupts and their indications via register GIS are independent from channel assigned interrupts, a vector with group 3, channel 7 indication may additionally refer to pending universal port interrupts. In version 2 upward the “PI”-bit of the interrupt vector indicates a pending parallel port interrupt.

  Interrupt groups 0 to 2 are assigned to definite single interrupt indications per channel. These are urgent receive and transmit interrupts which need to be serviced quickly. Due to this, no read access to interrupt status registers is necessary: the corresponding interrupt indication is reset after the $\overline{\text{INTA}}$ cycle has been finished.

  Interrupt group 3 combines all other interrupt sources. Thus, the interrupt status registers ISR0_x and ISR1_x which correspond to the requesting channel have to be examined. Version 1: if channel 7 is indicated, the global status register GIS has to be evaluated for pending channel and/or universal port interrupts. Version 2 upward: the “PI”-bit indicates a pending parallel port interrupt separately from channel 7 interrupts.

  The INT signal is reset when all interrupt indications are cleared (acknowledged). See also exceptions in Daisy Chaining mode.

- **Interrupt vector mode 2** (EDA = 1)
  Interrupt vector includes: extended device address, interrupt group
  (+): interrupt vector table needs only 4 entry points, placement of this entry field in steps of 16 bytes (6-bit device address), only 4 different interrupt service routines necessary.
  (–): context switching for each channel necessary (via register GIS).

  In case more than one source is active, the generated vector refers to the interrupt group with highest priority. For identification of the requesting channel, register GIS has to be read. Bits CN2..CN0 (channel number) have to be used for context switching, i.e. for computing the pointer to channel assigned data structures.
  Note: Universal port interrupts indications in register GIS are independent of channel assigned interrupts. Thus, one of indications PIA..D may be set although the generated interrupt vector refers to a group with priority and/or the channel number is less than 7.

Subsequent actions depend on the indicated interrupt group (similar to vector mode 1):
If one of interrupt groups 0 to 2 is indicated, no reading of channel assigned interrupt status registers is necessary; the corresponding interrupt indication is reset after the $\overline{\text{INTA}}$ cycle has been finished.

If interrupt group 3 is indicated, the interrupt status registers ISR0_x and ISR1_x which correspond to the requesting channel have to be examined. Version 1: in case channel 7 is indicated the global status register GIS has to be evaluated for pending channel and/or universal port interrupts. Version 2 upward: the “PI”-bit indicates a pending parallel port interrupt separately from channel 7 interrupts.

The INT signal is reset when all interrupt indications are cleared (acknowledged). See also exceptions in Daisy Chaining mode.

Note: Contents of Global Interrupt Status register GIS are frozen after every interrupt acknowledge cycle. Updating starts
- after the first read access to GIS after the interrupt vector has been output,
- after every read access to anyone of the channel assigned interrupt status registers,
- during every $\overline{\text{INTA}}$ cycle.

Updating of channel assigned interrupt status registers ISR0_x and ISR1_x is only prohibited during read access. Thus, status information may include indications of higher priority even in case of group 3 interrupts (e.g. a timer interrupt TIN channel 5 triggers an interrupt vector generation with group 3, channel 5 indication; before the service routine is able to read ISR0_5 and ISR1_5, an RPF condition occurs for channel 5; the current status information now includes TIN and RPF indication). This is implemented to avoid wasting time with servicing of low level requests while an urgent request of that channel is pending. This must be taken into consideration when designing the interrupt service routine for group 3 interrupts.

Masked Interrupts Visible in Status Registers (Version 2 Upward)

The interrupt vector contains only one interrupt at a time: the interrupt displayed in this vector results from a priority resolution among all **unmasked** active interrupt statuses. The Global Interrupt Status register (GIS) points to interrupt status registers with active interrupt indications. Register GIS should be evaluated if a pure interrupt polling scheme is used.

In version 1 of ESCC8 only unmasked interrupt statuses may:

- generate an interrupt at pin INT,
- generate an interrupt vector,
- be visible in GIS, and
- be visible in the interrupt status registers ISR0_0..7, ISR1_0..7 and PISA..D.

Masked interrupt statuses are only stored internally and they become visible when the mask is withdrawn.

In version 2 upward, an additional mode can be selected via bit IPC.VIS.

In this mode, masked interrupt status bits still neither generate an interrupt at pin INT nor generate an interrupt vector nor are visible in GIS, **but are displayed in the respective interrupt status register(s) ISR0_0..7, ISR1_0..7 and PISA..D.**

This mode is useful when some interrupt status bits are to generate an interrupt vector and other status bits are to be polled in the individual interrupt status registers.

Notes:

- In the visible mode, all active interrupt status bits, whether the corresponding actual interrupt is masked or not, are reset when the interrupt status register is read. Thus, when polling of some interrupt status bits is desired, care must be taken that unmasked interrupts are not lost in the process.
- All unmasked interrupt statuses are treated as before.
- Please note that whenever polling is used, all interrupt status registers concerned have to be polled individually (no “hierarchical” polling possible), since GIS only contains information on actually generated - i.e. unmasked-interrupts.

2.2.4 DMA Interface

The ESCC8 comprises a 16-channel DMA interface for fast and efficient data transfers. For all serial channels, a separate DMA Request output for transmit (DRT) and receive direction (DRR) as well as a DMA Acknowledgement ($\overline{\text{DACK}}$) input is provided.

The ESCC8 activates the DMA Request line as long as data transfers are needed from/ to the specific FIFO (level triggered demand transfer mode of DMA controller).

It is the responsibility of the DMA controller to perform the correct amount of bus cycles. Either read cycles will be performed if the DMA transfer has been requested from the receiver, or write cycles if DMA has been requested from the transmitter. If the DMA controller provides a DMA acknowledge signal (input to the ESCC8's $\overline{\text{DACK}}$ pin), each bus cycle implicitly selects the top of the specific FIFO and neither address (via A1-A8) nor chip select need to be supplied (I/O to Memory transfers). If no $\overline{\text{DACK}}$ signal is supplied, normal read/write operations (with addresses) must be performed (Memory to Memory transfers). The ESCC8 deactivates the DMA Request line immediately after the last read/write cycle of the data transfer has started.

As a very useful feature for single cycle DMA transfers, optional inversion of the functions of read/write control lines is implemented. If programmed via register CCR2

– $\overline{\text{RD}}$ and $\overline{\text{WR}}$ are exchanged in Intel bus interface mode,

– R/$\overline{\text{W}}$ is inverted in Motorola bus interface mode

while $\overline{\text{DACK}}$ is active. This allows easy connection to DMA controllers without dedicated I/O control lines as shown in **figure 16**.

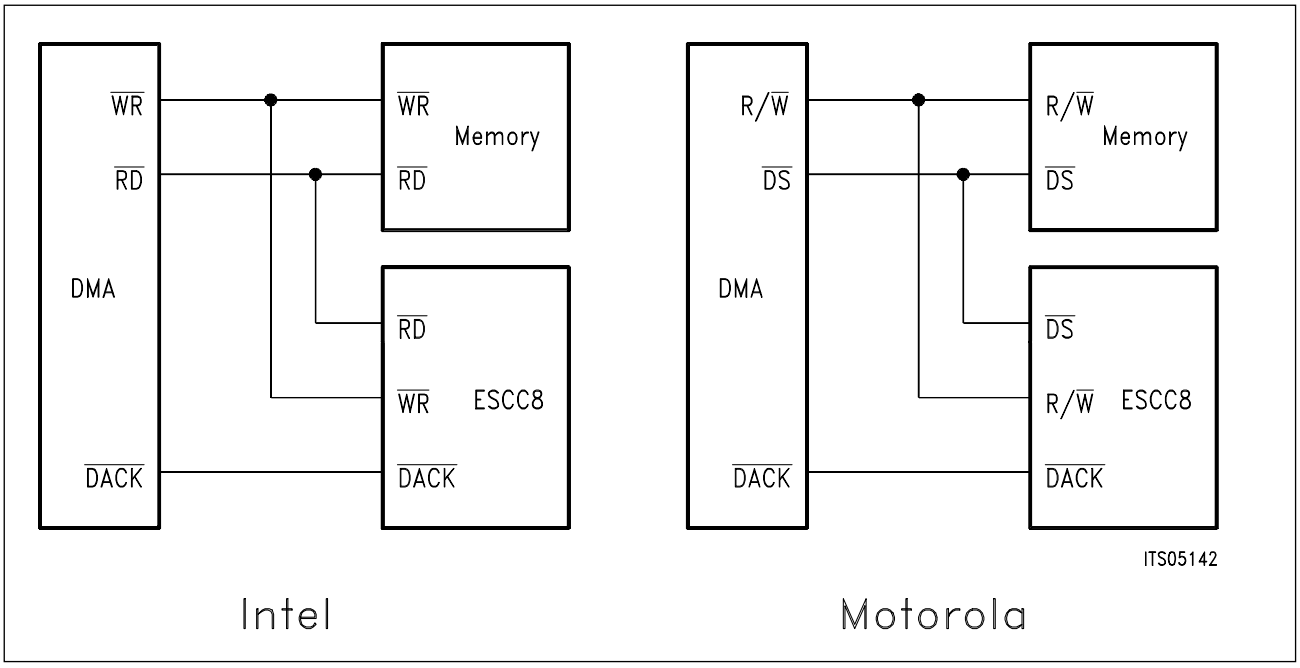


Figure 16
DMA Interfacing by Using Invert Mode

2.2.5 FIFO Structure

In all transmit and receive direction 64-byte deep FIFOs are provided for the intermediate storage of data between the serial interface and the CPU interface. The FIFOs are divided into two halves of 32-bytes. Only one half is accessible to the CPU or DMA controller at any time.

Organization of the FIFOs and access to their contents depends on the selected serial mode. For detailed information, refer to description of RFIFO and XFIFO in **chapter 4.1, chapter 4.2 and chapter 4.3**. In case 16-bit data bus width is selected by fixing pin WIDTH to logical “1” word access to the FIFOs is enabled. Data output to bus lines D0-D15 as a function of the selected interface mode is shown in **figure 17 and 18.** Of course, byte access is also allowed.

The effective length of the accessible part of RFIFO can be changed from 32 bytes (RESET value) down to 1 (ASYNC and BISYNC mode) or 2 (HDLC mode) bytes.

In version 1, only threshold 32 is available in HDLC mode.

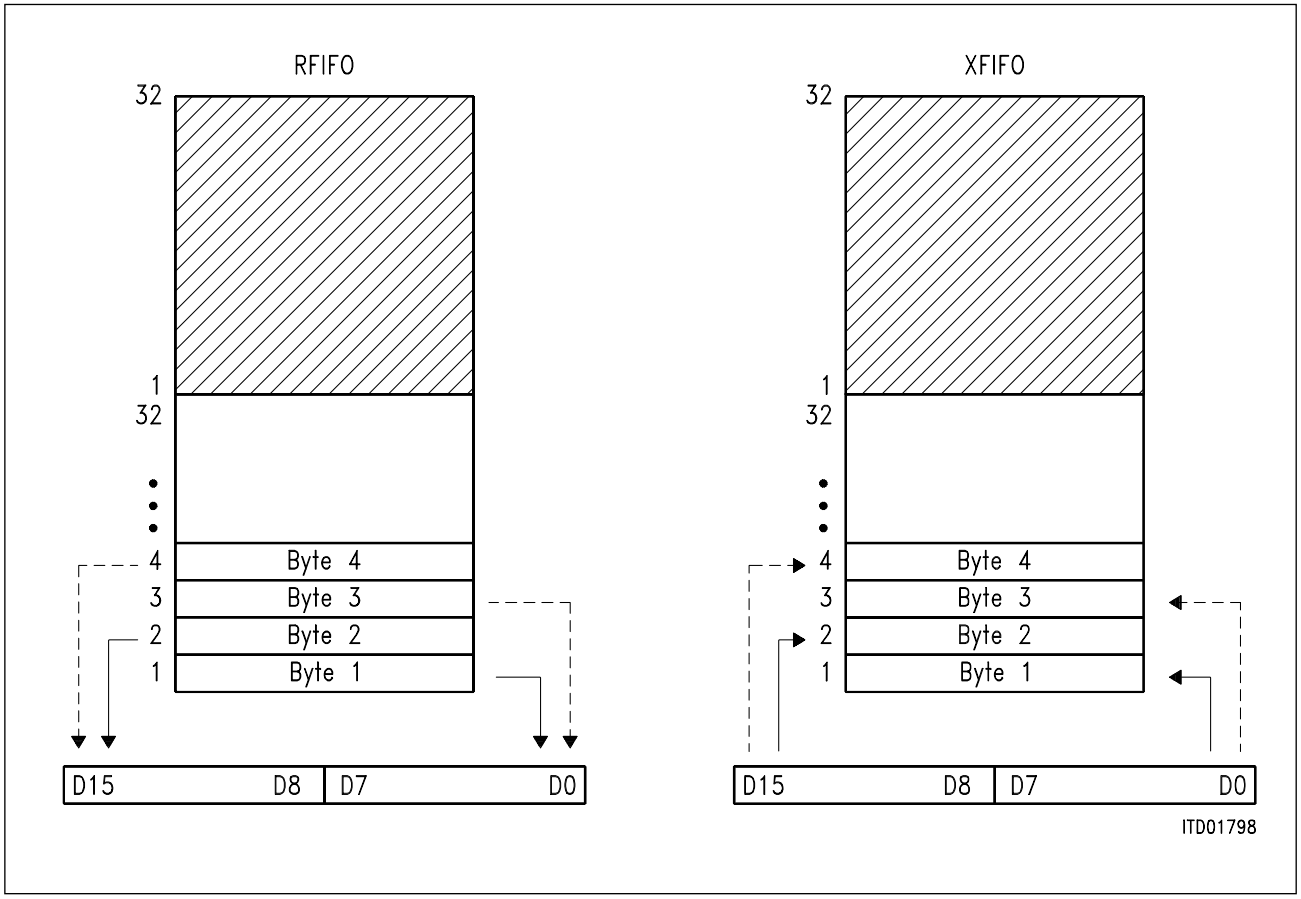


Figure 17
FIFO Word Access (Intel Mode)

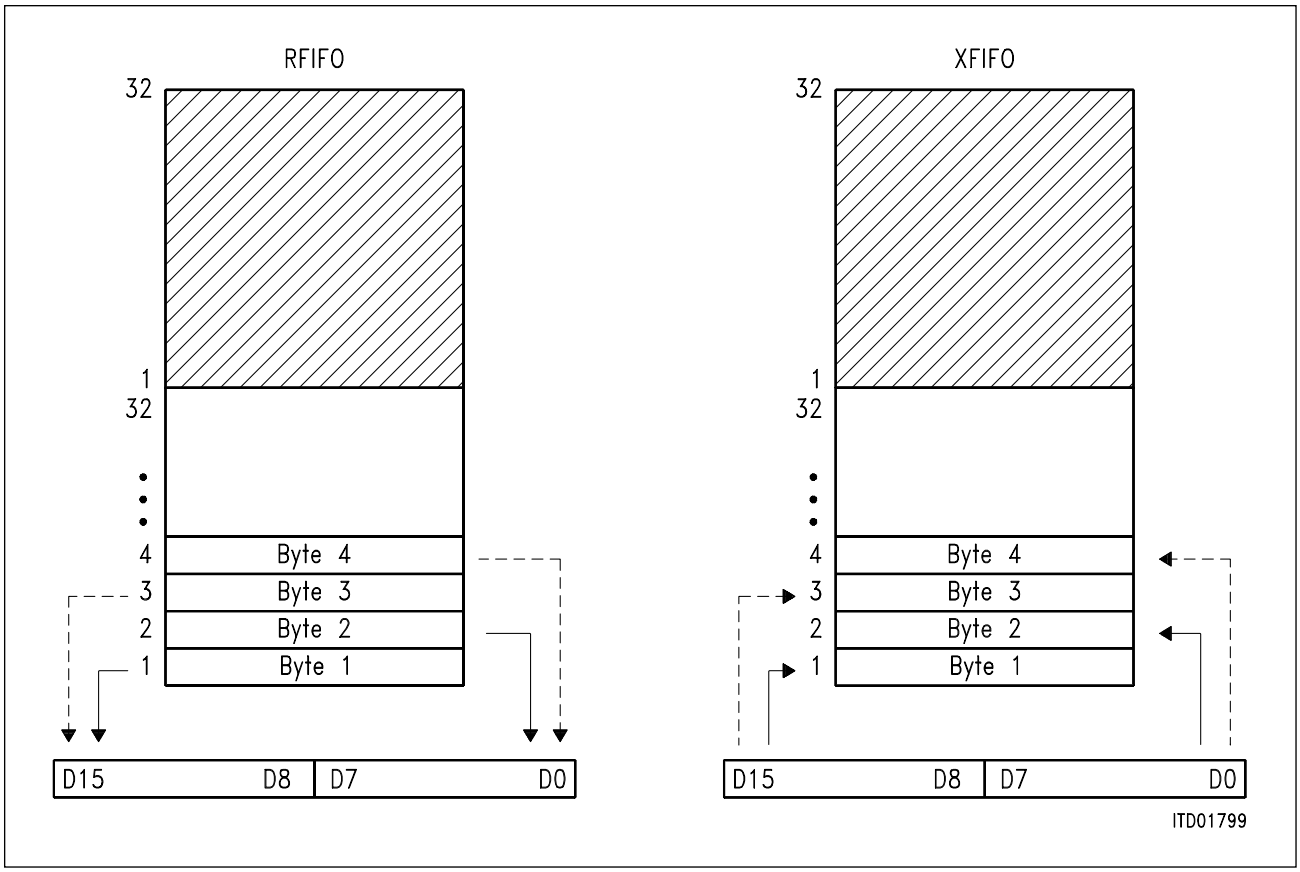


Figure 18
FIFO Word Access (Motorola Mode)

2.3 HDLC/SDLC Serial Mode

2.3.1 Operating Modes

The HDLC controller of each channel can be programmed to operate in various modes, which are different in the treatment of the HDLC frame in receive direction. Thus, the receive data flow and the address recognition features can be performed in a very flexible way, to satisfy almost any practical requirements.

There are 6 different operating modes which can be set via the MODE register.

Auto-Mode (MODE: MDS1, MDS0 = 00)

Characteristics: Window size 1, random message length, address recognition.

The ESCC8 processes autonomously all numbered frames (S-, I-frames) of an HDLC protocol. The HDLC control field, data in the I-field of the frames and an additional status byte are temporarily stored in the RFIFO. The HDLC control field as well as additional information can also be read from special registers (RHCR, RSTA).

Depending on the selected address mode, the ESCC8 can perform a 2-byte or 1-byte address recognition. If a 2-byte address field is selected, the high address byte is compared with the fixed value FEH or FCH (group address) as well as with two individually programmable values in RAH1 and RAH2 registers. According to the ISDN LAPD protocol, bit 1 of the high byte address will be interpreted as COMMAND/ RESPONSE bit (C/R), dependent on the setting of the CRI bit in RAH1, and will be excluded from the address comparison.

Similarly, two comparison values can be programmed in special registers (RAL1, RAL2) for the low address byte. A valid address will be recognized in case the high and low byte of the address field correspond to one of the compare values. Thus, the ESCC8 can be called (addressed) with 6 different address combinations, however, only the logical connection identified through the address combination RAH1, RAL1 will be processed in the auto-mode, all others in the non auto-mode. HDLC frames with address fields that do not match any of the address combinations, are ignored by the ESCC8.

In the case of a 1-byte address, RAL1 and RAL2 will be used as comparison registers. According to the X.25 LAPB protocol, the value in RAL1 will be interpreted as COMMAND and the value in RAL2 as RESPONSE.

In version 2 and upwards the address bytes can be masked to allow selective broadcast frame recognition. For further information see **chapter 2.3.4.10**.

Non-Auto-Mode (MODE: MDS1, MDS0 = 01)

Characteristics: address recognition, arbitrary window size.

All frames with valid addresses (address recognition identical to auto-mode) are forwarded directly via the RFIFO to the system memory.

The HDLC control field, data in the I-field and an additional status byte are temporarily stored in the RFIFO. The HDLC control field and additional information can also be read from special registers (RHCR, RSTA).

In non-auto-mode, all frames with a valid address are treated similarly.

In version 2 upward the address bytes can be masked to allow selective broadcast frame recognition. For further information see **chapter 2.3.4.10**.

Transparent Mode 1 (MODE: MDS1, MDS0, ADM = 101)

Characteristics: address recognition high byte

Only the high byte of a 2-byte address field will be compared.The address byte is compared with the fixed value FEH or FCH (group address) as well as with two individually programmable values in RAH1 and RAH2 registers. The whole frame excluding the first address byte will be stored in RFIFO. RAL1 contains the second and RHCR the third byte following the opening flag.

In version 2 and upwards the address bytes can be masked to allow selective broadcast frame recognition. For further information see **chapter 2.3.4.10**.

Transparent Mode 0 (MODE: MDS1, MDS0, ADM = 100)

Characteristics: no address recognition

No address recognition is performed and each frame will be stored in the RFIFO. RAL1 contains the first and RHCR the second byte following the opening flag.

Extended Transparent Modes 0, 1 (MODE: MDS1, MDS0 = 11)

Characteristics: fully transparent

In extended transparent modes, fully transparent data transmission/reception without HDLC framing is performed, i.e. without FLAG generation/recognition, CRC generation/ check, or bit-stuffing. This allows user specific protocol variations.

Data transmission is always performed out of the XFIFO. In extended transparent mode 0 (ADM = 0), data reception is done via the RAL1 register, which always contains the current data byte assembled at the R×D pin. In extended transparent mode 1 (ADM = 1), the receive data are additionally shifted into the RFIFO.

Receive Data Flow (Summary)

The following figure gives an overview of the management of the received HDLC frames in the different operating modes.

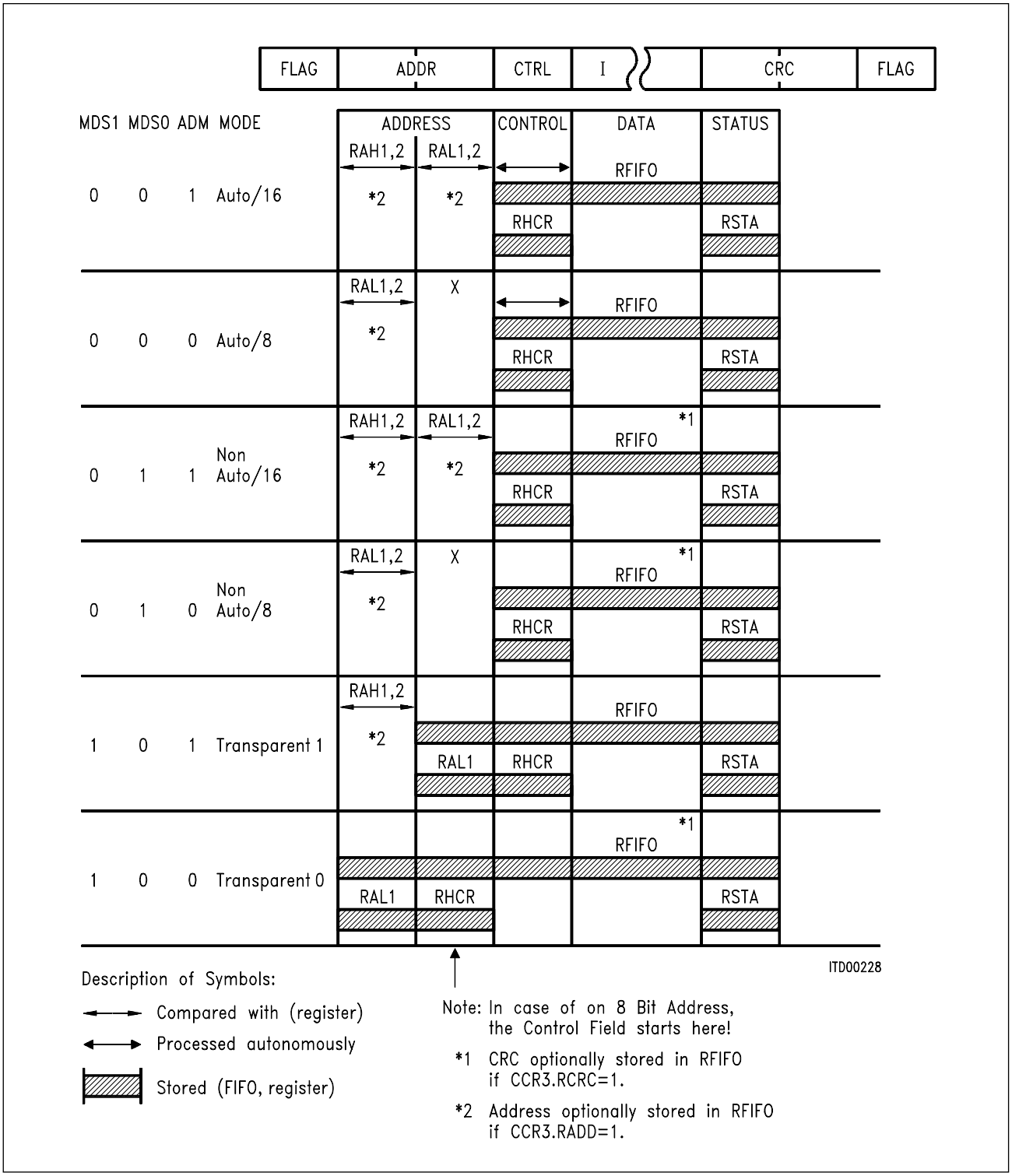


Figure 19
Receive Data Flow of ESCC8

Transmit Data Flow

Two different types of frames can be transmitted:

– frames and

– transparent frames

as shown below.

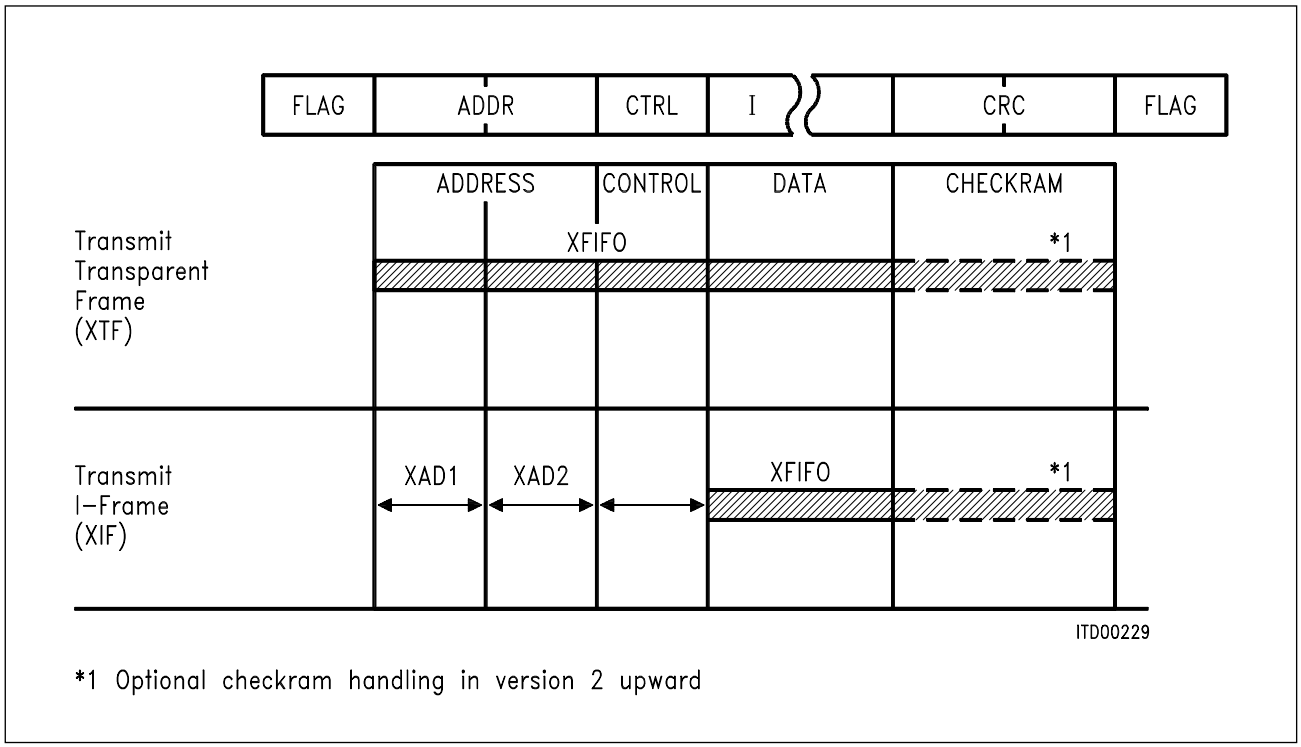


Figure 20
Transmit Data Flow of ESCC8

For I-frames (command XIF via CMDR register), the address and control fields are generated autonomously by the ESCC8 and the data in the XFIFO is entered into the information field of the frame. This is possible only if the ESCC8 is operated in the automode.

For transparent frames (command XTF via CMDR register), the address and the control fields have to be entered in the XFIFO as well. This is possible in all operating modes and used also in auto-mode for sending U-frames.

Version 2 upward:
If CCR3.XCRC is set, the CRC checksum will not be generated internally. The checksum has to be provided via the transmit FIFO (XFIFO) as the last two or four bytes. The transmitted frame will be closed automatically only with a (closing) flag.

Note: The ESCC8 does not check whether the length of the frame, i.e. the number of bytes to be transmitted makes sense or not.

2.3.2 Procedural Support (Layer-2 Functions)

When operating in the auto mode, the ESCC8 offers a high degree of protocol support. In addition to address recognition, the ESCC8 autonomously processes all (numbered) S- and I-frames (prerequisite window size 1) with either normal or extended control field format (modulo 8 or modulo 128 sequence numbers - selectable via RAH2 register).

The following functions will be performed:

- Updating of transmit and receive counter
- Evaluation of transmit and receive counter
- Processing of S commands
- Flow control with RR/RNR
- Generation of responses
- Recognition of protocol errors
- Transmission of S commands, if acknowledgement is not received
- Continuous status query of remote station after RNR has been received
- Programmable timer/repeater functions.

In addition, all unnumbered frames are forwarded directly to the processor. The logical link can be initialized by software at any time (Reset HDLC Receiver, RHR-command). Additional logical connections can be operated in parallel by software.

2.3.2.1 Full-Duplex LAPB/LAPD Operation

Initially (i.e. after RESET), the LAP controllers of the eight serial channels are configured to function as a combined (primary/secondary) station, where they autonomously perform a subset of the balanced X.25 LAPB/ISDN LAPD protocol.

Reception of Frames

The logical processing of received S-frames is performed by the ESCC8 without interrupting the CPU. The CPU is merely informed by interrupt of status changes in the remote station (receive ready/not receive ready) and protocol errors (unacceptable N(R), or S-frame with I field).

I-frames are also processed autonomously and checked for protocol errors. The I-frame will not be accepted in the case of sequence errors (no interrupt is forwarded to the CPU), but is immediately confirmed by an S response. If the CPU sets the ESCC8 into a “receive not ready” status, an I-frame will not be accepted (no interrupt) and an RNR response is transmitted. U frames are always stored in the RFIFO and forwarded directly to the CPU. The logical sequence and the reception of a frame in auto mode is illustrated in **figure 21.**

Note: The state variables N(S), N(R) are evaluated within the window size 1, i.e. the ESCC8 checks only the least significant bit of the receive and transmit counter regardless of the selected modulo count.

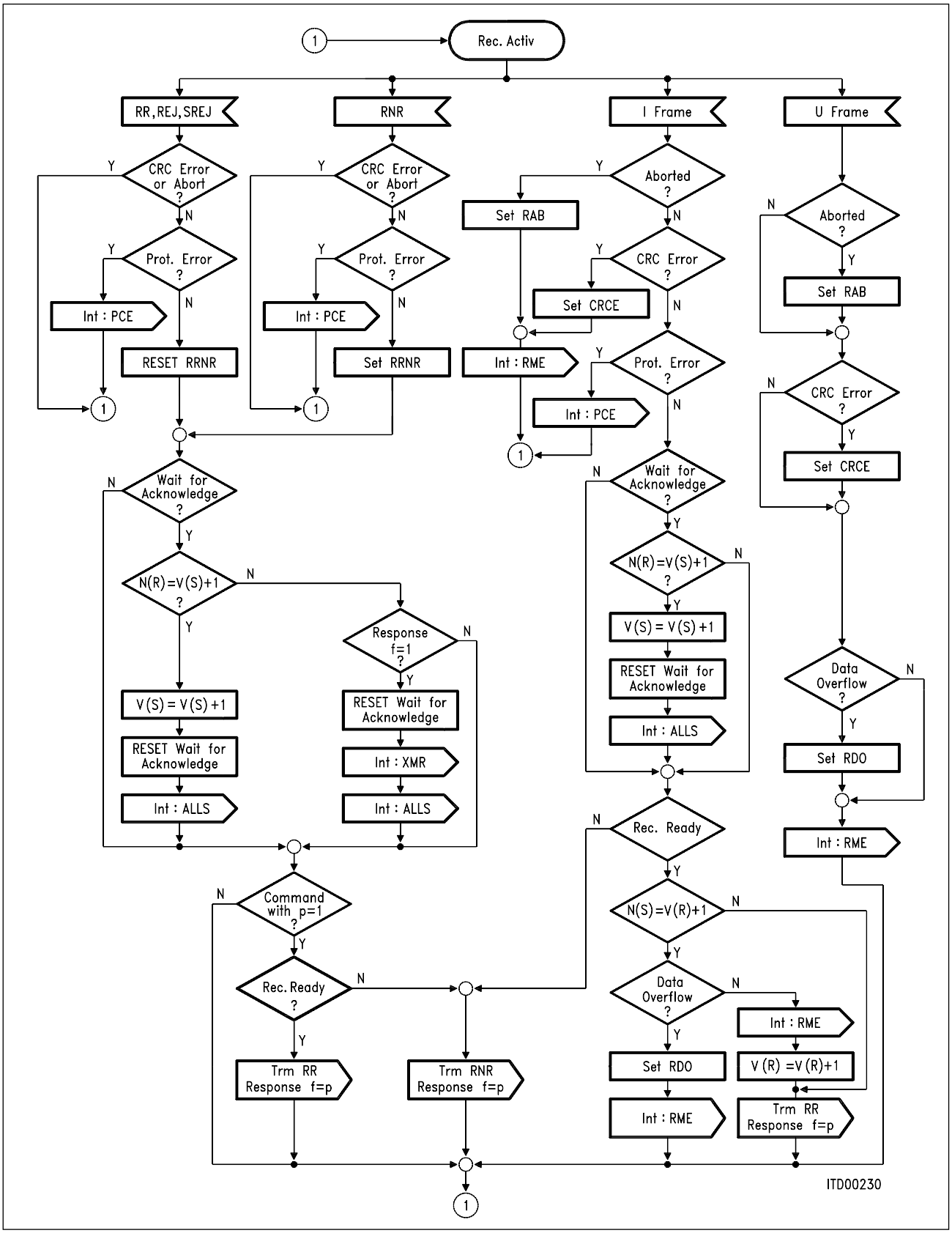


Figure 21
Processing of Received Frames in Auto Mode

Transmission of Frames

The ESCC8 autonomously transmits S commands and S responses in the auto mode. Either transparent or I-frames can be transmitted by the user. The software timer has to be operated in the internal timer mode to transmit I-frames. After the frame has been transmitted, the timer is self-started, the XFIFO is inhibited, and the ESCC8 waits for the arrival of a positive acknowledgement. This acknowledgement can be provided by means of an S- or I-frame.

If no positive acknowledgment is received during time $t1$, the ESCC8 transmits an S command (p = 1), which must be answered by an S response (f = 1). If the S response is not received, the process is performed n1 times (in HDLC known as N2, refer to register TIMR).

Upon the arrival of an acknowledgement or after the completion of this poll procedure the XFIFO is enabled and an interrupt is generated. Interrupts may be triggered by the following:

- Message has been positively acknowledged (ALLS interrupt)
- Message must be repeated (XMR interrupt)
- Response has not been received (TIN interrupt)

Additionally, XPR interrupts are generated which indicate that new data can be written to the XFIFO. Using XPR enables high data rates, e.g. in conjunction with back-to-back frames or shared flags.

In auto-mode, however, only when the ALLS interrupt has been issued may data of a new frame be written to the XFIFO!

Upon arrival of an RNR frame, the software timer is started and the status of the remote station is polled periodically after expiration of $t1$, until the status “receive ready” has been detected. The user is informed via the appropriate interrupt. If no response is received after n1 times, a TIN interrupt, and $t1$ clock periods thereafter an ALLS interrupt is generated and the process is terminated.

Note: The internal timer mode should only be used in the auto mode.

Transparent frames can be transmitted in all operating modes. After the transmission of a transparent frame the XFIFO is immediately released, which is confirmed by interrupt (XPR). In this case, time monitoring can be performed with the timer in the external timer mode.

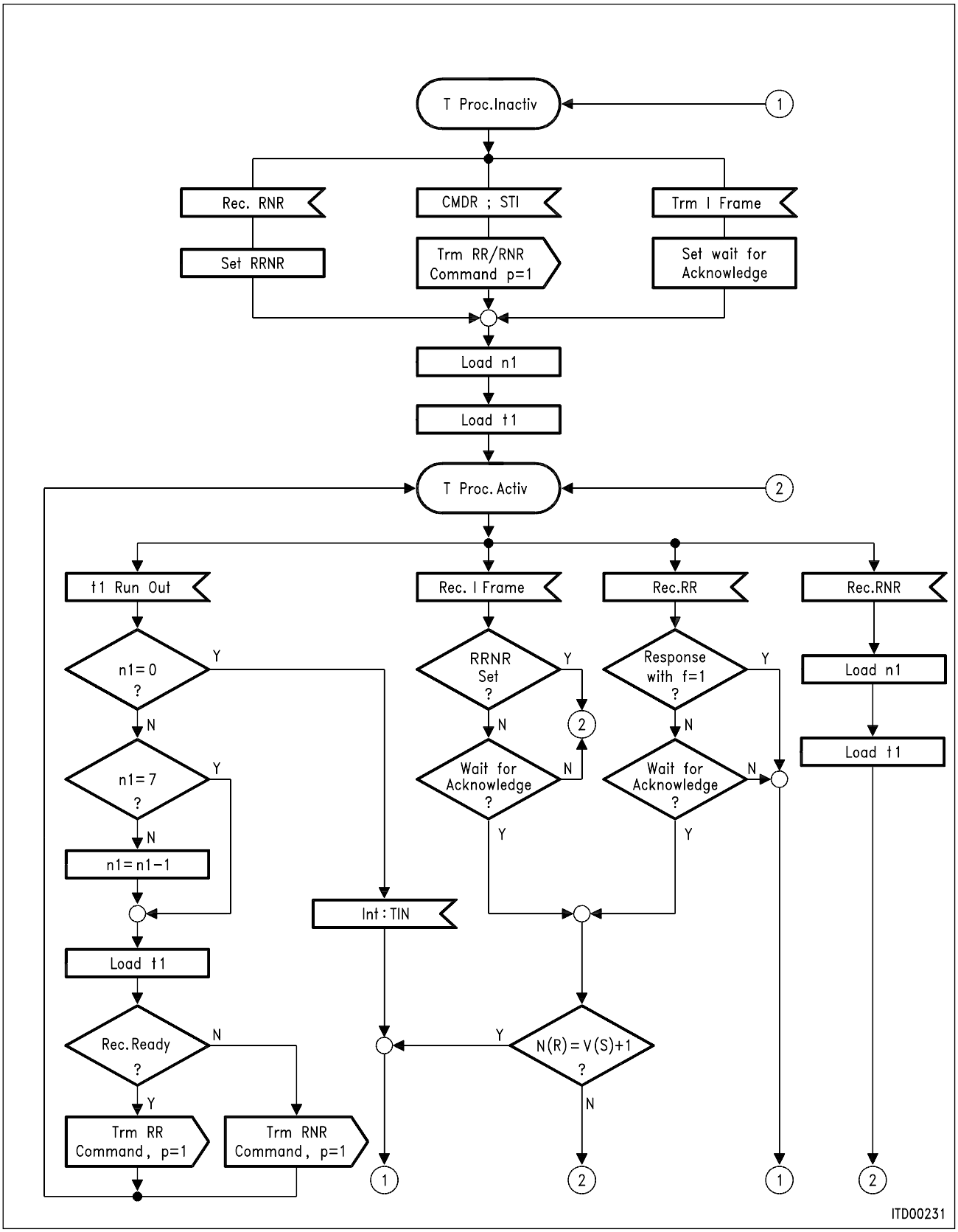


Figure 22
Timer Procedure / Poll Cycle

Examples

The interaction between ESCC8 and the CPU during transmission and reception of I-frames is illustrated in **figure 23**, the flow control with RR/RNR during reception of I-frames in **figure 24**, and during transmission of I-frames in **figure 25**. Both the sequence of the poll cycle and protocol errors are shown in **figure 26**.

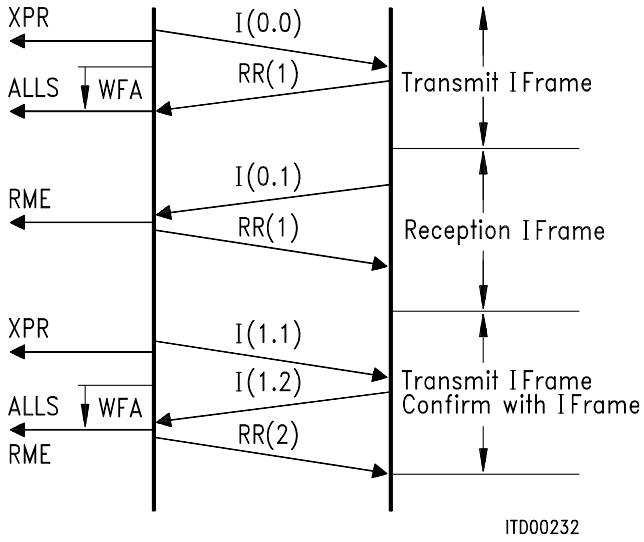


Figure 23
Transmission/Reception I-Frames

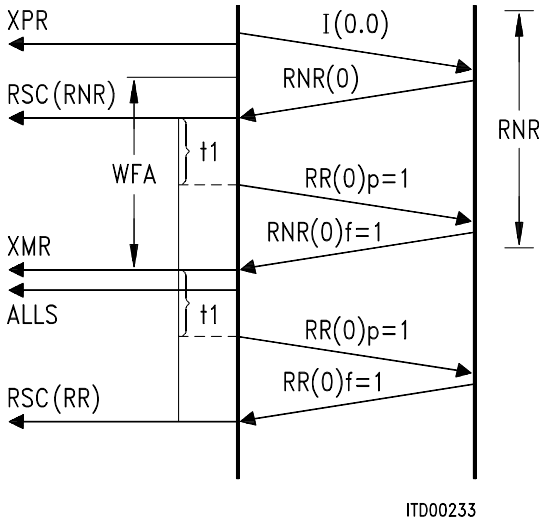


Figure 24
Flow Control/Transmission

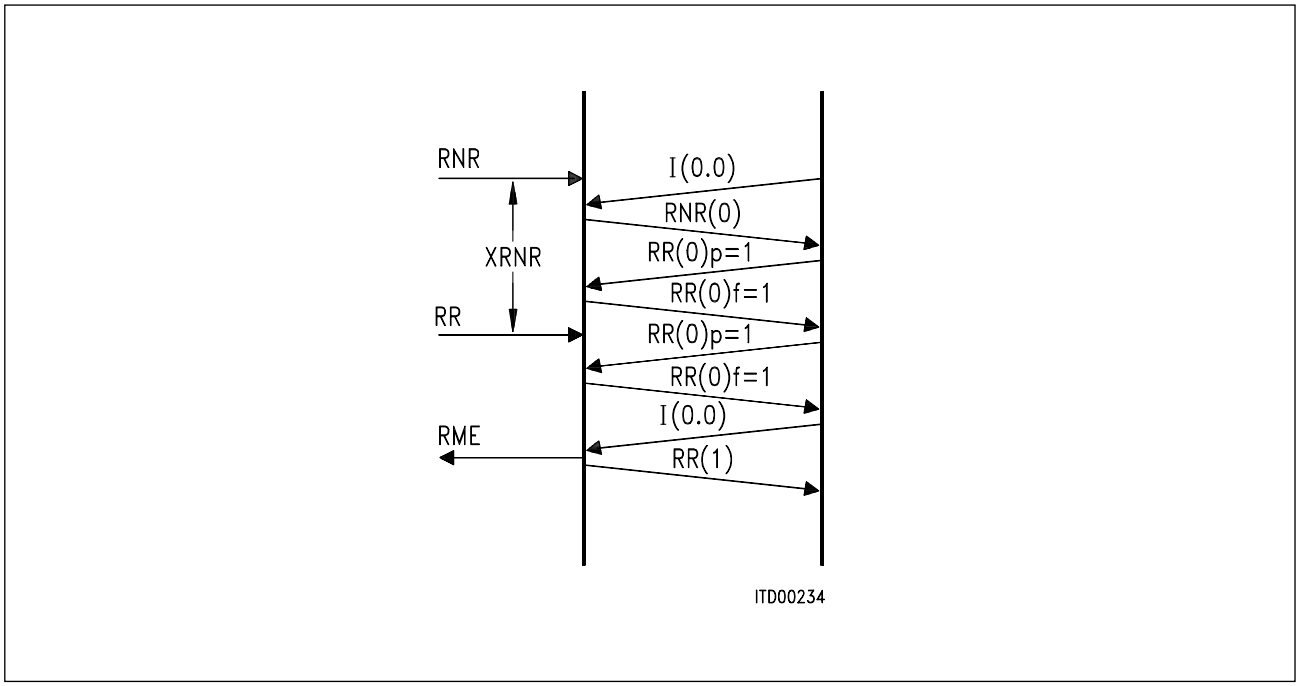


Figure 25
Flow Control/Reception

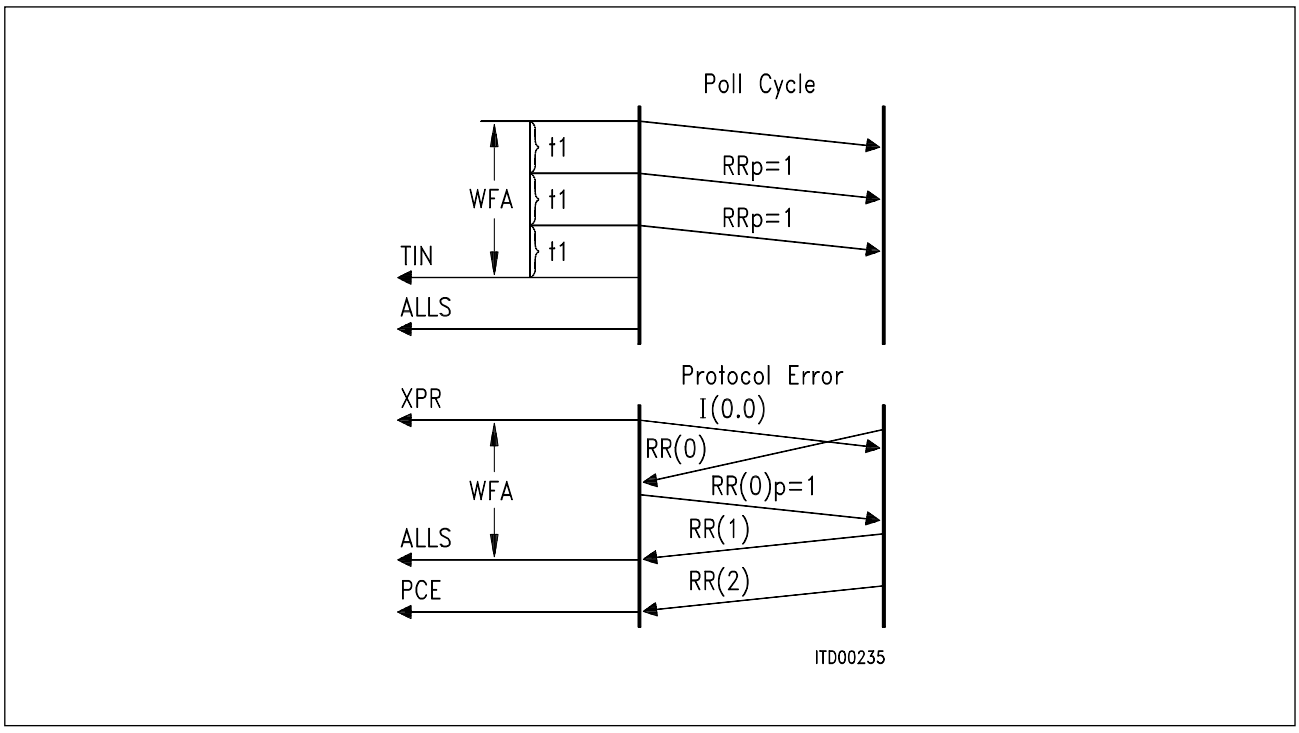


Figure 26
S Commands/Protocol Error

2.3.2.2 Half-Duplex SDLC-NRM Operation

The LAP controllers of the eight serial channels can be configured to function in a half-duplex Normal Response Mode (NRM), where they operate as a slave (secondary) station, by setting the NRM bit in the XBCH register of the corresponding channel.

In contrast to the full-duplex LAPB/LAPD operation, where the combined (primary + secondary) station transmits both commands and responses and may transmit data at any time, the NRM mode allows only responses to be transmitted **and** the secondary station may transmit only when instructed to do so by the master (primary) station. The ESCC8 gets the permission to transmit from the primary station via an S-, or I-frame with the poll bit (p) **set.**

The NRM mode can be profitably used in a point-to-multipoint configuration with a fixed master-slave relationship, which guarantees the absence of collisions on the common transmit line. It is the responsibility of the master station to poll the slaves periodically and to handle error situations.

Prerequisite for NRM operation is:

- Auto mode with 8-bit address field selected
  MODE: MDS1, MDS0, ADM = 000
- External timer mode
  MODE: TMD = 0
- Same transmit and receive addresses, since only responses can be transmitted, i.e.
  XAD1 = XAD2 = RAL1 = RAL2 (address of secondary)

Note: The broadcast address may be programmed in RAL2 if broadcasting is required. In this case RAL1 and RAL2 are not equal.
The primary station has to operate in transparent SDLC mode.

Reception of Frames

The reception of frames functions similarly to the LAPB/LAPD operation (**see 2.3.2.1**).

Transmission of Frames

The ESCC8 does **not** transmit S-, or I-frames if not instructed to do so by the primary station via an S-, or I-frame with the poll bit set.

The ESCC8 can be prepared to send an I-frame by the CPU by issuing an XIF command (via CMDR) at any time. The transmission of the frame, however, will not be initiated by the ESCC8 until reception of either an

- RR, or
- I-frame

with a poll bit set (p = 1).

After the frame has been transmitted (with the final bit set), the XFIFO is inhibited and the ESCC8 waits for the arrival of a positive acknowledgement.

Since the on-chip timer of the ESCC8 must be operated in the external mode (a secondary may not poll the primary for acknowledgements), timer supervision must be done by the primary station.

Upon the arrival of an acknowledgement the XFIFO is enabled and an interrupt is forwarded to the CPU, either the

– message has been positively acknowledged (ALLS interrupt), or the

– message must be repeated (XMR interrupt).

Additionally, the timer can be used **under CPU control** to provide timer recovery of the secondary if no acknowledgements are received at all.

Note: The transmission of transparent frames is possible only if the permission to send is given by an S-frame (p = 1) or I-frame.

Examples

A few examples of ESCC8 / CPU interaction in the case of NRM mode are shown in **figure 27** to **30**.

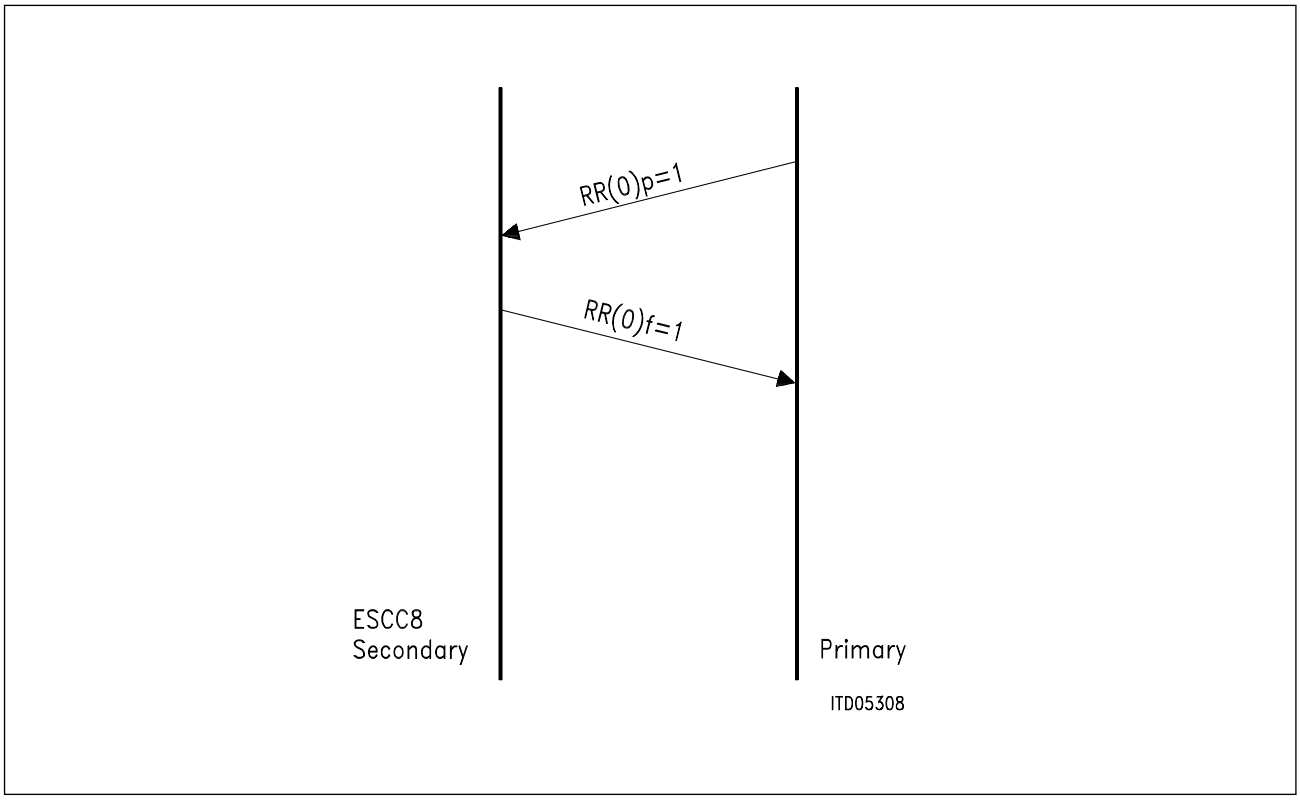


Figure 27
No Data to Send

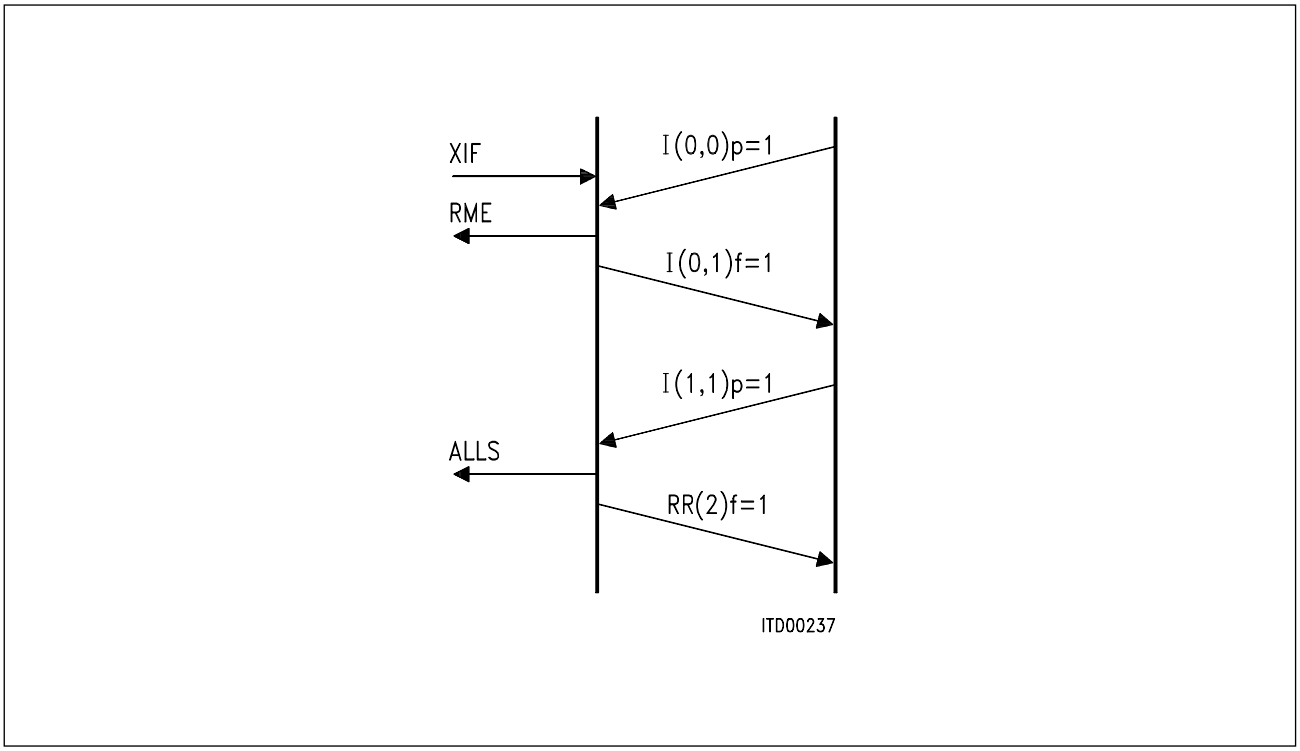


Figure 28
Data Reception/Transmission

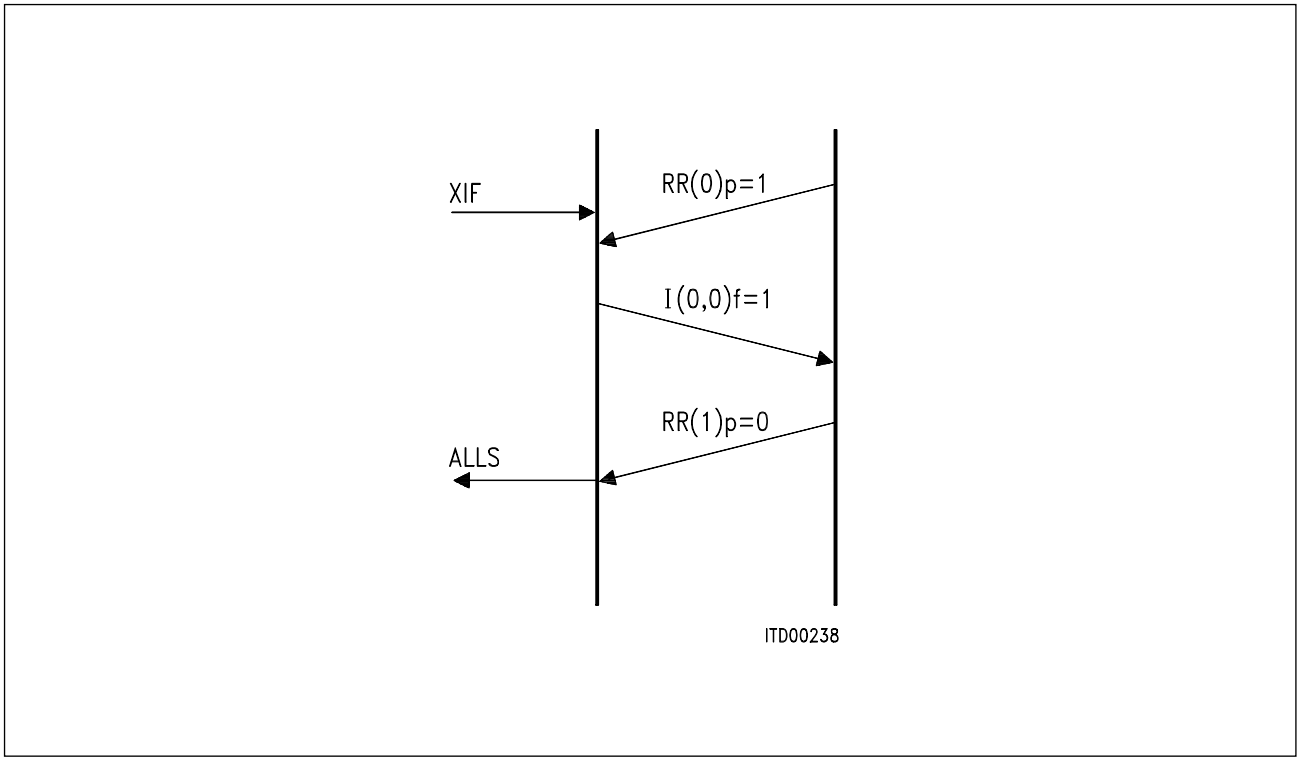


Figure 29
Data Transmission (no Error)

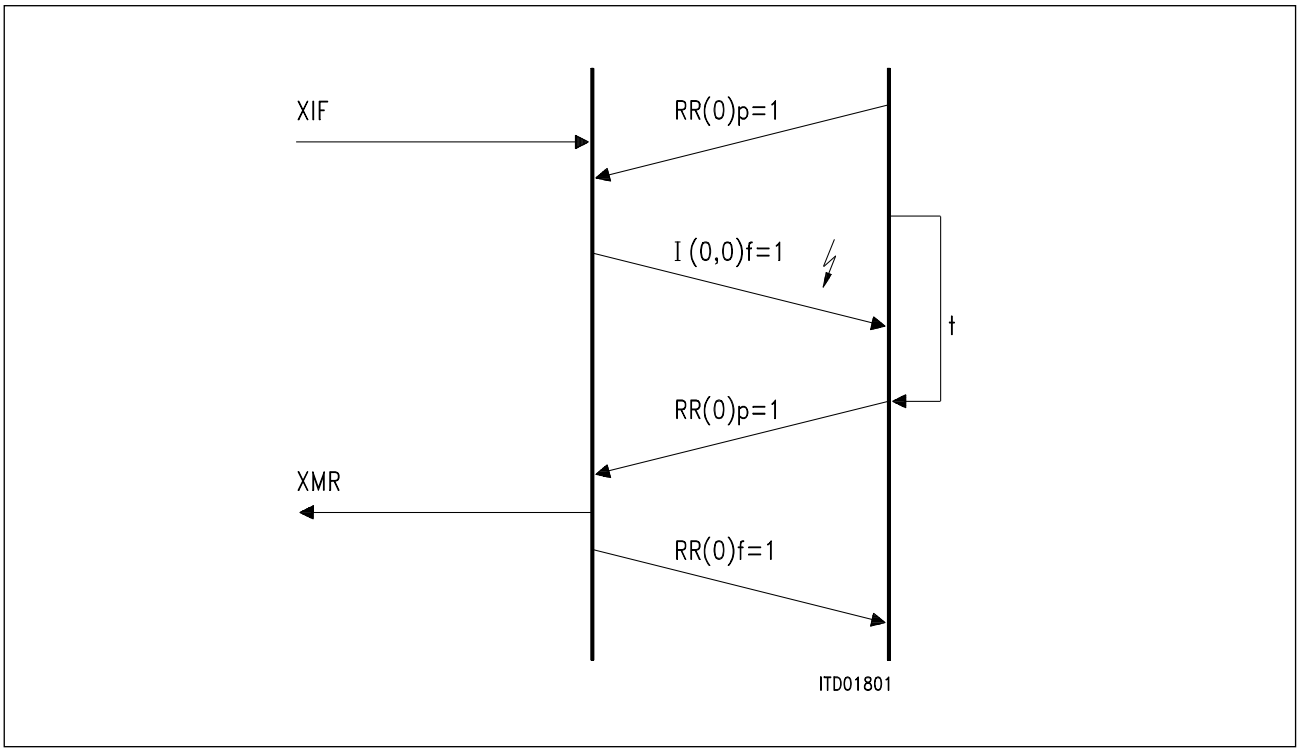


Figure 30
Data Transmission (Error)

2.3.2.3 Error Handling

Depending on the error type, erroneous frames are handled according **table 3**.

Table 3
Error Handling

Frame Type	Error Type	Generated Response	Generated Interrupt	Record Status
I	CRC error	–	RME	CRC error
	Aborted	–	RME	Abort
	Unexpected N(S)	S-frame	–	–
	Unexpected N(R)	–	PCE	–
S	CRC error	–	–	–
	Aborted	–	–	–
	Unexpected N(S)	–	PCE	–
	With I-field	–	PCE	–

Note: The station variables (V(S), V(R)) are not changed.

2.3.3 SDLC Loop

As a special variant of IBM's SDLC protocol the SDLC Loop is used to connect several Secondary (= slave) Stations to one Primary (= master) Station. Different from standard HDLC, a reserved bit sequence is defined as "End of Poll" sequence (EOP = one "0" bit, followed by at least 7 "1" bits). Note that in standard HDLC this sequence is defined as Abort Sequence, therefore with SDLC Loop frame abortion is not available.

The ESCC8 facilitates entering and leaving the loop. In contrast to the protocol support described above, autonomous processing of S- and I-frames is not implemented by the circuit but is left to software. Prerequisite for correct operation is

- SDLC Loop mode enabled (register CCR0)
- Normal Response Mode selected (XBCH:NRM = 1)
- Non-auto-mode or transparent mode with 8-bit address field selected
- External timer mode
- NRZ or NRZI data encoding enabled (register CCR0); no bus configuration
- R×CLK = T×CLK
- Interframe Timefill = Flags

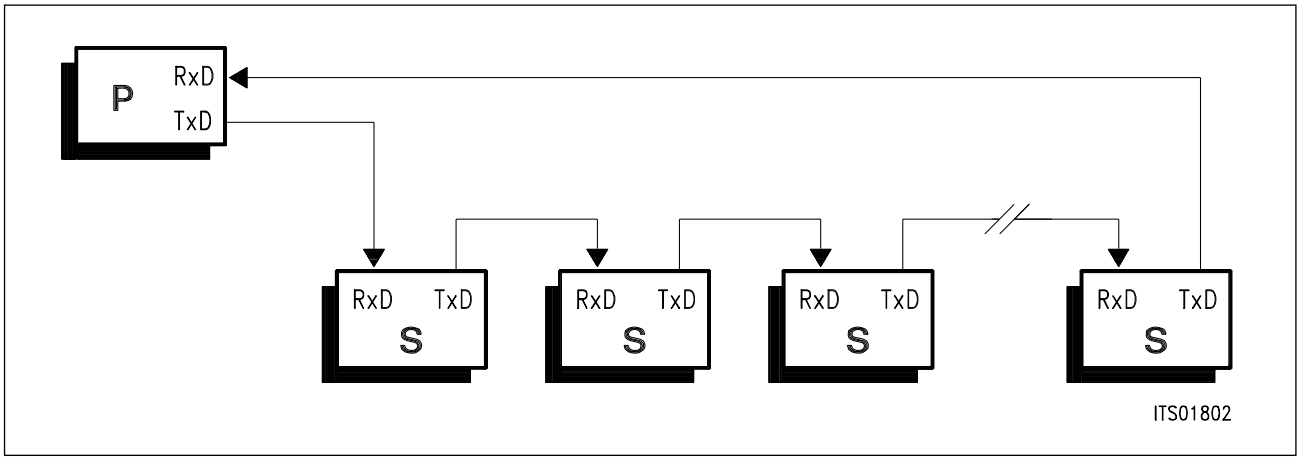


Figure 31
SDLC Loop

The loop is formed by connecting T×D output of one station to the R×D input of the next one (**refer to figure 31**). This configuration is physically a loop, but logically a point-to-multipoint configuration.

In every Secondary Station data flow from R×D to T×D is handled depending on Secondary's current state as follows:

- Initially, RxD and TxD are connected together with gate delay (**OFF Loop** state). Data sent out from the Primary is passed on by every Secondary to the next one. Thus, data is transparent to all Secondaries.

- After reception of an EOP sequence a Secondary can go to the **ON Loop** state. As opposed to the Off Loop State, all data is forwarded to the next station with one bit delay.
- If a Secondary is requested (polled) by the Primary to transmit data or responses, it has to wait for reception of a further EOP sequence. By flipping the seventh “1” of the EOP sequence to “0” it generates a flag sequence and consequently all following Secondary Stations are inhibited from sending. Simultaneously, RxD is disconnected from TxD and transmission of a frame (or several frames) may start (**Active ON Loop** state). After terminating transmission the station reconnects RxD to TxD. Thus, an EOP sequence is formed and another station may start data transmission.

Processing the EOP sequences is handled automatically by the ESCC8: commands (GLP, GALP in register CCR1) and state indications (interrupts EOP, OLP, AOLP in register ISR1) are provided to control and monitor the state of the ESCC8 as Secondary Station.

Figure 32 shows the state diagram for the Secondary. Note that in order to be able to hold “Active On Loop” state “flags” has to be selected as interframe time fill, as opposed to “idle”.

Note: The Primary Station has to operate in standard SDLC mode.

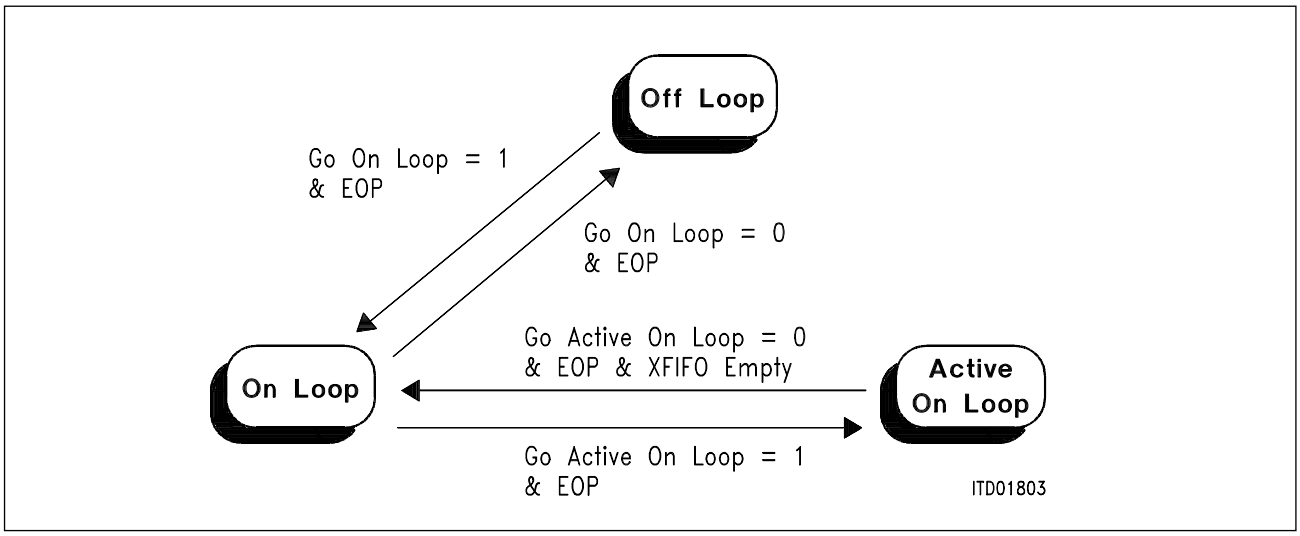


Figure 32
State Diagram of SDLC Loop/Secondary

Reception of Frames

SDLC Loop as special variant of the SDLC protocol works in half-duplex normal response mode, that means that data transmission and data reception at the same time is not permitted. Normally, data reception is only possible in the On Loop state.

The ESCC8, however, allows data reception in every state. Activation/deactivation of the receiver is effected by the user by programming the RAC bit in register MODE.

Transmission of Frames

Sending frames is only possible in the Active On Loop state. Here, transmission can start with the XTF command. If necessary, Flags as Interframe Timefill are inserted before the current frame begins (the modified EOP and the first Flag may share a “0”). After finishing frame transmission, Flags as Interframe Timefill are again sent until the “Go Active On Loop” command (GALP) is reset. By returning to On Loop state an EOP sequence is formed, the transmitter is disabled and R×D is connected to T×D again with one bit delay.

Note: XTF or XIF may be issued before the Active On Loop state is reached. In this case, transmission starts immediately after entering the Active On Loop state. The opening Flag of the first frame is sent out immediately following after the modified EOP sequence (both may share a “0”).

2.3.4 Special Functions

2.3.4.1 Shared Flags

The closing Flag of a previously transmitted frame simultaneously becomes the opening Flag of the following frame if there is one to be transmitted. The Shared Flag feature is enabled by setting bit SFLG in control register CCR1.

2.3.4.2 Preamble Transmission

If enabled via register CCR3, a programmable 8-bit pattern (register PRE) is transmitted with a selectable number of repetitions after Interframe Timefill transmission is stopped and a new frame is ready to be sent out.

Note: Zero Bit Insertion is disabled during preamble transmission. To guarantee correct function the programmed preamble value should be different from Receive Address Byte values defined for any of the connected stations.

2.3.4.3 CRC-32

In HDLC/SDLC mode, error protection is done by CRC generation and checking.

In standard applications, CRC-CCITT algorithm is used. The Frame Check Sequence at the end of each frame consists of two bytes of CRC checksum.

If required, the CRC-CCITT algorithm can be replaced by the CRC-32 algorithm, enabled via register CCR2. In this case the Frame Check Sequence consists of four bytes.

2.3.4.4 Extended Transparent Transmission and Reception

When programmed in the extended transparent mode via the MODE register (MDS1, MDS0 = 11), each channel of the ESCC8 performs fully transparent data transmission and reception without HDLC framing, i.e. without

- FLAG insertion and deletion
- CRC generation and checking
- Bit-stuffing.

In order to enable fully transparent data transfer, RAC bit in MODE has to be reset and FF_H has to be written to XAD1, XAD2 and RAH2.

Data transmission is always performed out of XFIFO by directly shifting the contents of XFIFO via the serial transmit data pin (T×D). Transmission is initiated by setting CMDR:XTF (08_H); end of transmission is indicated by ISR1:EXE (10_H).

In receive direction, the character last assembled via receive data line (R×D) is available in RAL1 register. Additionally, in extended transparent mode 1 (MODE: MDS1, MDS0, ADM = 111), received data is shifted into RFIFO.

This feature can be profitably used e.g. for:

- User specific protocol variations
- Line state monitoring, or
- Test purposes, in particular for monitoring or intentionally generating HDLC protocol rule violations (e.g. wrong CRC)

Character or octet boundary synchronization can be achieved by using clock mode 1 with an external receive strobe input to pin CD.

2.3.4.5 Cyclic Transmission (Fully Transparent)

If the extended transparent mode is selected, the ESCC8 supports the continuous transmission of the contents of the transmit FIFO.

After having written 1 to 32 bytes to XFIFO, the command

XREP.XTF.XME

via the CMDR register (bit 7…0 = “00101010” = $2A_H$) forces the ESCC8 to repeatedly transmit the data stored in XFIFO via T×D pin.

The cyclic transmission continues until a reset command (CMDR: XRES) is issued, after which continuous “1”-s are transmitted.

Note: In DMA-mode the command XREP and XTF has to be written to CMDR.

2.3.4.6 Continuous Transmission (DMA Mode only)

If data transfer from system memory to the ESCC8 is done by DMA (DMA bit in XBCH set), the number of bytes to be transmitted is usually defined via the Transmit Byte Count registers (XBCH, XBCL: bits XBC11…XBC0).

Setting the “Transmit Continuously” (XC) bit in XBCH, however, the byte count value is ignored and the DMA interface of ESCC8 will continuously request for transmit data any time 32 new bytes can be entered in XFIFO.

This feature can be used e.g. to transmit frames of length higher than the byte count specified by XBCH, XBCL (frames with more than 4096 bytes).

Note: If the XC bit is reset during continuous transmission, the transmit byte count becomes valid again, and the ESCC8 will request the amount of DMA transfers programmed via XBC11..XBC0. Otherwise, the continuous transmission and the generation of DMA requests is stopped when a data underrun condition occurs in XFIFO. Instead of CRC, continuous “1”-s (IDLE) are transmitted thereafter.

2.3.4.7 Receive Length Check Feature

The ESCC8 offers the possibility to supervise the maximum length of received frames and to terminate data reception in case this length is exceeded.

This feature is controlled via the special Receive Length Check Register (RLCR).

The function is enabled by setting the RC (Receive Check) bit in RLCR and programming the maximum frame length via bits RL6…RL0. The maximum receive length can be determined as a multiple of 32-byte blocks as follows:

$$MAX.LENGTH = (RL + 1) \times 32$$

where RL is the value written to RL6…RL0.

All frames exceeding this length are treated as if they had been aborted by the remote station, i.e. the CPU is informed via an

- RME interrupt, and the
- RAB bit in RSTA register is set.

To distinguish this from the case where an abort sequence is indeed received (sent by the remote station), the receive byte count registers RBCH, RBCL will contain a value exceeding the maximum receive length (via RL6...RL0) by one or two bytes.

2.3.4.8 One Bit Insertion

Similar to the zero bit insertion (bit-stuffing) mechanism, as defined by the HDLC protocol, the ESCC8 offers a completely new feature of inserting/deleting a one after seven consecutive zeros in the transmit/receive data stream, if the serial channel is operating in a bus configuration. This method is useful if clock recovery is to be performed by DPLL.

Since only NRZ data encoding is supported in a bus configuration, there are possibly long sequences without edges in the receive data stream in case of successive “0”-s received, and the DPLL may lose synchronization.

Using the one bit insertion feature by setting the OIN bit in the CCR1 register, however, it is guaranteed that at least after

– 5 consecutive “1”-s a “0” will appear (bit-stuffing), and after

– 7 consecutive “0”-s a “1” will appear (one insertion)

and thus a correct function of the DPLL is ensured.

Note: As with the bit-stuffing, the “one insertion” is fully transparent to the user, but it is not in accordance with the HDLC protocol, i.e. it can only be applied in proprietary systems using circuits that also implement this function, such as the SAB 82532.

2.3.4.9 CRC ON/OFF Feature (version 2 upward)

As an option in non-auto mode or transparent mode 0, the internal handling of received and transmitted CRC checksum can be influenced via control bits CCR3.RCRC and CCR3.XCRC.

Receive Direction

The received CRC checksum is always assumed to be in the 2 (CRC-CCITT) or 4 (CRC-32) last bytes of a frame, immediately preceding a closing flag. In the version 1 of ESCC8 a check is performed on the CRC but the received CRC bytes are not transferred to the RFIFO. In version 2 upwards, if CCR3.RCRC is set, the received CRC checksum will be written to RFIFO where it precedes the frame status byte (contents of register RSTA). The received CRC checksum is additionally checked for correctness. If non-auto mode is selected, the limits for “Valid Frame” check are modified (refer to description of bit RSTA.VFR).

Transmit Direction

If CCR3.XCRC is set, the CRC checksum is not generated internally. The checksum has to be provided via the transmit FIFO (XFIFO) as the last two or four bytes. The transmitted frame will only be closed automatically with a (closing) flag.

Note: The ESCC8 does not check whether the length of the frame, i.e. the number of bytes to be transmitted makes sense or not.

2.3.4.10 Receive Address Handling (version 2 upward)

Mask for Address Detection

The Receive Address Low/High Byte (RAL1/RAH1) can be masked by setting the corresponding bits in the mask registers (AML/AMH) to allow extended broadcast address recognition. This feature is applicable to all operating modes with address recognition (auto mode, non-auto mode and transparent mode 1). It is disabled if all bits of registers AML and AMH are set to zero (RESET value). The function of RAL2/RAH2 and detection of the fixed group address FEH or FCH if applicable to the selected operating mode remain unchanged.

Note: As a very useful option, the detected receive address can be pushed to RFIFO (CCR3.RADD).

Receive Address Pushed to RFIFO

As an option in the auto mode, non-auto mode and transparent mode 1, the address field of received frames can be pushed to RFIFO (first one/two bytes of the frame). This function is especially useful in conjunction with the extended broadcast address recognition. It is enabled by setting control bit CCR3.RADD.

Note: In this case the ratio of receive frequency (fr) to transmit frequency (fx) and to master clock frequency (fm) must fulfill:

fr/fx < 1.5 (normal operation),
fr/fm < 1.5 (master clock operation).

2.4 Asynchronous Serial Mode

2.4.1 Character Frame

Character framing is achieved by special Start and Stop bits. Each data character is preceded by one Start bit and terminated by one or two Stop bits. The character length is selectable from 5 up to 8 bits. Optionally, a parity bit can be added which complements the number of ones to an even or odd quantity (even/odd parity). The parity bit can also be programmed to have a fixed value (Mark or Space). **Figure 33** shows the asynchronous character format.

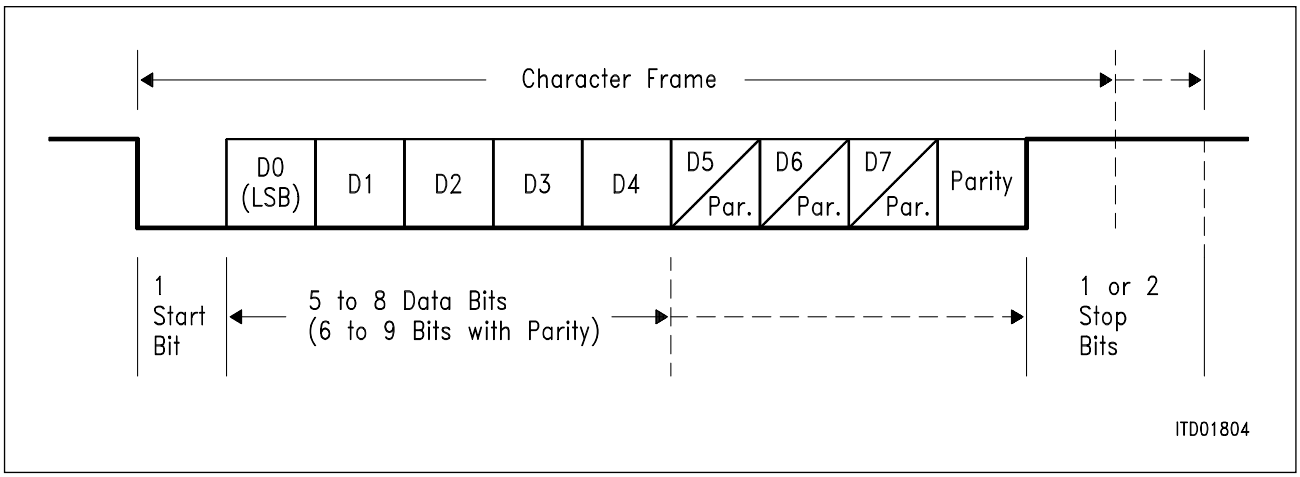


Figure 33
Asynchronous Character Frame

2.4.2 Data Reception

2.4.2.1 Operating Modes

The ESCC8 offers the flexibility to combine clock modes, data encoding and data sampling in many different ways. However, only definite combinations make sense and are recommended for correct operation:

Asynchronous Mode

Prerequisites:

- Bit clock rate 16 selected (CCR1.BCR = 1)
- Clock mode 0, 1, 3b, 4, or 7b selected
- NRZ data encoding

The receiver which operates with a clock rate equal to 16 times the nominal data bit rate, synchronizes itself to each character by detecting and verifying the Start Bit. Since character length, parity and Stop Bit length is known, the ensuing valid bits are sampled.

Oversampling (3 samples) around the nominal bit center in conjunction with majority decision is provided for every received bit (including Start Bit).

The synchronization lasts for one character, the next incoming character causes a new synchronization to be performed. As a result, the demand for high clock accuracy is reduced. Two communication stations using the asynchronous procedure are clocked independently, their clocks need not be in phase or locked to exactly the same frequency but, in fact, may differ from one another within a certain range.

Isochronous Mode

Prerequisites:

- Bit clock rate 1 selected (CCR1.BCR = 0)
- Clock mode 2, 3a, 6, or 7a (DPLL mode) has to be used in conjunction with FM0, FM1 or Manchester encoding.

The isochronous mode uses the asynchronous character format. However, each data bit is only sampled once (no oversampling).

In clock modes 0 and 1, the input clock has to be externally phase locked to the data stream. This mode allows much higher transfer rates. Clock modes 3b, 4 and 7b are not recommended due to difficulties with bit synchronization when using the internal baud rate generator.

In clock modes 2, 3a, 6, and 7a, clock recovery is provided by the internal DPLL. Correct synchronization of the DPLL is achieved if there are enough edges within the data stream, which is generally ensured only if Bi-Phase encoding (FM0, FM1 or Manchester) is used.

2.4.2.2 Storage of Data

If the receiver is enabled, received data is stored in RFIFO (the LSB is received first). Moreover, the CD input may be used to control data reception. Character length, the number of Stop Bits and the optional parity bit are checked. Storage of parity bits can be disabled. Errors are indicated via interrupts. Additionally, the character error status (framing and parity) can optionally be stored in the RFIFO (**refer to chapter 4.2.2**).

Filling of the accessible part of RFIFO is controlled by

- A programmable threshold level
- Detection of the programmable Termination Character (optional).

Additionally, the Time-Out condition as optional status information indicates that a certain time (refer to register ISR0) has elapsed since the reception of the last character.

2.4.3 Data Transmission

The selection of asynchronous or isochronous operation has no further influence on the transmitter. The bit clock rate is solely a dividing factor for the selected clock source.

Transmission of the contents of XFIFO starts after the XF command is issued (the LSB is sent out first). Further data is requested by interrupt (XPR) or DMA. The character frame for each character, consisting of Start Bit, the character itself with defined character length, optionally generated parity bit and Stop Bit(s) is assembled.

After finishing transmission (indicated by the “All Sent” interrupt), IDLE (logical “1”) is transmitted on T×D.

Additionally, the $\overline{\text{CTS}}$ signal may be used to control data transmission.

2.4.4 Special Features

2.4.4.1 Break Detection/Generation

Break generation: On issuing the XBRK command (register DAFO), the T×D pin is immediately forced to physical “0” level with the first following clock edge, and released with the first clock edge after this command has been reset.

Break detection: The ESCC8 recognizes the Break condition upon receiving consecutive (physical) “0”s for the defined character length, the optional parity and the selected number of Stop Bits (“zero” character and framing error). The “zero” character is not pushed to RFIFO. If enabled, the BRK interrupt is generated.

The Break condition will be present until a “1” is received which is indicated by the Break Terminated interrupt (BRKT).

2.4.4.2 Flow Control by XON/XOFF (version 2 upward)

Programmable XON and XOFF

Two eight-bit control registers (XON, XOFF) contain the programmable values for XON and XOFF characters. The number of significant bits in a register is determined by the programmed character length (right justified).

Two programmable eight-bit registers MXN and MXF serve as mask registers for the characters in XON and XOFF, respectively:

A “1” in a mask register has the effect that no comparison is performed between the corresponding bits in the received characters (“don’t cares”) and the XON and the XOFF register. At RESET, the mask registers are zeroed, i.e. all bit positions are compared.

A received character is considered to be recognized as a valid XON or XOFF character

– if it is correctly framed (correct length),
– if its bits match the ones in the XON or XOFF registers over the programmed character length,
– if it has correct parity (if applicable).

Received XON and XOFF characters are always stored in the receive FIFO, as any other characters.

In-Band Flow Control of Transmitted Characters

Recognition of an XON or an XOFF character causes always a corresponding maskable interrupt status to be generated (ISR1.XON / IMR1.XON; ISR1.XOFF / IMR1.XOFF).

Further action depends on the setting of a control bit MODE.FLON (Flow Control On):

0 No further action is automatically taken by the ESCC8.

1 The reception of an XOFF character automatically turns off the transmitter after the currently transmitted character (if any) has been completely shifted out (XOFF state). The reception of an XON character automatically makes the transmitter resume transmitting (XON state).

After hardware RESET, bit MODE.FLON is at “0”.

When bit MODE.FLON is made to go from “0” to “1”, the transmitter is first in the “XON state”, until an XOFF character is received.
When bit MODE.FLON is made to go from “1” to “0”, the transmitter always goes in the “XON state”, and transmission is only controlled by the user and by the $\overline{\text{CTS}}$ input.

The in-band flow control of the transmitter via received XON and XOFF characters can be combined with control via $\overline{\text{CTS}}$ pin, i.e. the effect of the $\overline{\text{CTS}}$ pin is independent of whether in-band control is used or not. The transmitter is enabled only if $\overline{\text{CTS}}$ is low and XON state has been reached.

Transmitter Status Bit

The status bit “Flow Control Status” (STAR.FCS) indicates the current state of the transmitter, as follows:

0 if the transmitter is in XON state

1 if the transmitter is in XOFF state

Note: The transmitter cannot be turned off by software without disrupting data possibly remaining in the XFIFO.

Flow Control for Received Data

After writing a character value to register TIC (Transmit Immediate Character) its contents are inserted in the outgoing character stream

- Immediately upon writing this register by the microprocessor if the transmitter is in IDLE state. If no further characters (XFIFO contents) are to be transmitted, i.e. the transmitter returns to IDLE state after transmission of TIC, an ALLS (All Sent) interrupt will be generated.
- After the end of a character currently being transmitted if the transmitter is not in IDLE state. This does not affect the contents of the XFIFO. Transmission of characters from XFIFO is resumed after the contents of register TIC are shifted out.

Transmission via this register is possible even when the transmitter is in XOFF state (however, $\overline{\text{CTS}}$ must be “low”).

The TIC register is an eight-bit register. The number of significant bits is determined by the programmed character length (right justified). Parity value (if programmed) and selected number of stop bits are automatically appended, similar to the characters written in the XFIFO. The usage of TIC is independent of flow control, i.e. is not affected by bit MODE.FLON.

To control access to register TIC, an additional status bit STAR.TEC (TIC Executing) is implemented which signals that transmission command of currently programmed TIC is accepted but not completely executed. Further access to register TIC is only allowed if bit STAR.TEC is “0”.

2.4.4.3 Continuous Transmission (DMA Mode only)

If data transfer from system memory to the ESCC8 is done by DMA (DMA bit in XBCH set), the number of characters to be transmitted is usually defined via the Transmit Byte Count registers (XBCH, XBCL: bits XBC11…XBC0).

However, if the “Transmit Continuously” (XC) bit in XBCH is set, the byte count value is ignored and the DMA interface of ESCC8 will continuously request for transmit data any time 32 new characters can be stored in XFIFO.

Note: If the XC bit is reset during continuous transmission, the transmit byte count becomes valid again, and the ESCC8 will request the amount of DMA transfers programmed via XBC11…XBC0. Otherwise, the continuous transmission is stopped when a data underrun condition occurs in XFIFO, i.e. the DMA controller does not transfer further data to ESCC8. In this case continuous “1”-s (IDLE) are transmitted.

2.5 Character Oriented Serial Mode (MONOSYNC/BISYNC)

2.5.1 Data Frame

Character oriented protocols achieve synchronization between transmitting and receiving station by means of special SYN characters. Two examples are the MONOSYNC and IBM's BISYNC procedures. BISYNC has two starting SYN characters while MONOSYNC uses only one SYN. **Figure 34** gives an example of the message format.

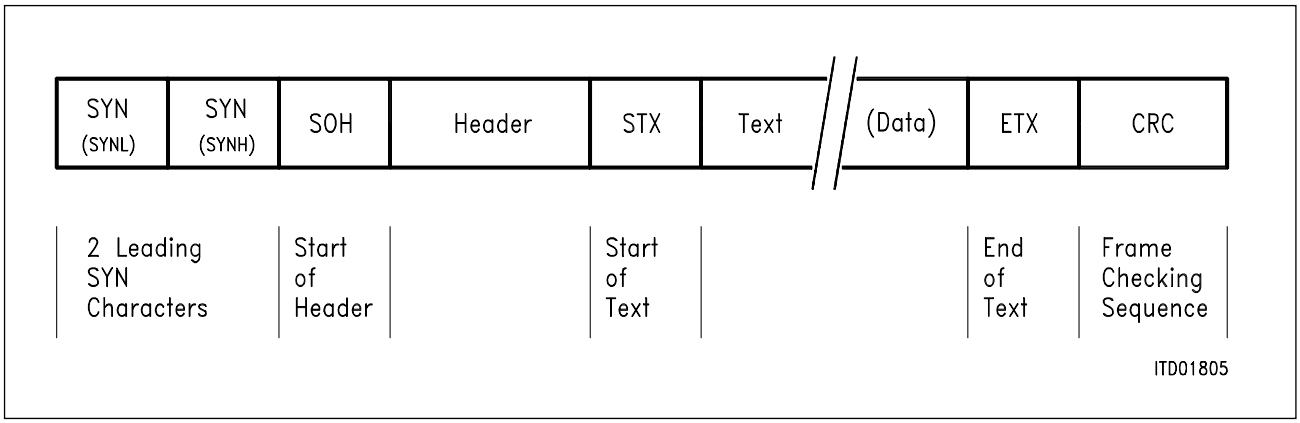


Figure 34
BISYNC Message Format

The SYN character, its length, the length of data characters and additional parity are programmable:

- SYN with 6 or 8 bit length (MONOSYNC), programmable via register SYNL.
- 2 SYN with 6 or 8 bit length each (BISYNC), programmable via registers SYNL and SYNH.
- Data character length may vary from 5 to 8 bits.
- Parity information (even/odd parity, Mark, Space) may be appended to the character.

2.5.2 Data Reception

The receiver is generally activated by setting the RAC bit in the MODE register. Additionally, the CD signal may be used to control data reception. After issuing the HUNT command, the receiver monitors the incoming data stream for the presence of specified SYN character(s). However, data reception is still disabled. If synchronization is gained by detecting the SYN character(s), SCD interrupt is generated and **all** data is pushed to RFIFO, i.e. control sequences, data characters and optional CRC frame checking sequence (the LSB is received first). In normal operation, SYN characters are excluded from storage to RFIFO. SYN character length can be specified independently of the selected data character length. If required, the character parity bit and/or parity status is FIFOed together with each data byte.

As an option, the loading of SYN characters in RFIFO may be enabled by setting the SLOAD bit in register RFC. Note that in this case SYN characters are treated as data. Consequently, for correct operation it must be guaranteed that SYN character length equals the character length + optional parity bit. This is the user's responsibility.

Filling of the accessible part of RFIFO is controlled by a programmable threshold level. RFIFO read is requested by interrupt (RPF) or DMA.

Reception is stopped if
1. the receiver is deactivated by resetting the RAC bit, or
2. the CD signal goes inactive (if Carrier Detect Auto Start is enabled), or
3. the HUNT command is issued again, or
4. the Receiver Reset command (RRES) is issued, or
5. a programmed Termination Character has been found (optional).

On actions 1. and 2., reception remains disabled until the receiver is activated again. After this is done, and generally in cases 3. and 4., the receiver returns to the (non-synchronized) Hunt state. In case 5. a HUNT command has to be issued. Reception of data is internally disabled until synchronization is regained.

Note: Further checking of frame length, extraction of text or data information and verifying the Frame Checking Sequence (e.g. CRC) has to be done by the microprocessor.

2.5.3 Data Transmission

Transmission of data written to XFIFO is initiated after the Transmit Frame command (XF) is issued (the LSB is sent out first). Additionally, the $\overline{\text{CTS}}$ signal may be used to control data transmission. Further data is requested by interrupt (XPR) or DMA. The message frame is assembled by appending all data characters to the specified SYN character(s) until Transmit Message End is detected (XME command in interrupt mode, or, in DMA mode, when the number of characters specified in XBCH, XBCL have been transferred). Internally generated parity information may be added to each character (SYN, CRC and Preamble characters are excluded).

If enabled via CRC Append bit (CAPP), the internally calculated CRC checksum (16 bit) is added to the message frame. Selection between CRC-16 and CRC-CCITT algorithms is provided. For all characters which have to be included into CRC calculation, the CON flag has to be set to “1”. This flag which controls the CRC generator is FIFOed together with each character. There is no need to modify CON for every character loaded if continuous characters are to be either included into or excluded from CRC calculation.

Note that

– internally generated SYN characters are always excluded from CRC calculation,

– CRC checksum (2 bytes) is sent without parity.

The internal CRC generator is automatically initialized before transmission of a new frame starts. The initialization value is selectable.

After finishing data transmission, Interframe Timefill (SYN characters or IDLE) is automatically sent.

2.5.4 Special Functions

2.5.4.1 Preamble Transmission

If enabled via register CCR3, a programmable 8-bit pattern (register PRE) is transmitted with a selectable number of repetitions after Interframe Timefill transmission is stopped and a new frame is ready to be sent out.

Note: If the preamble pattern equals the SYN pattern, reception is triggered by the preamble.

2.5.4.2 Continuous Transmission (DMA Mode only)

If data transfer from system memory to the ESCC8 is done by DMA (DMA bit in XBCH set), the number of characters to be transmitted is usually defined via the Transmit Byte Count registers (XBCH, XBCL: bits XBC11…XBC0).

Setting the “Transmit Continuously” (XC) bit in XBCH, however, the byte count value is ignored and the DMA interface of ESCC8 will continuously request for transmit data any time 32 new characters can be entered in XFIFO.

This feature can be used to transmit frames of length higher than the byte count specified by XBCH, XBCL (frames with more than 4095 bytes).

Note: If the XC bit is reset during continuous transmission, the transmit byte count becomes valid again, and the ESCC8 will request the amount of DMA transfers programmed via XBC11…XBC0. Otherwise, the continuous transmission and the generation of DMA input requests is stopped when a data underrun condition occurs in XFIFO. Instead of CRC, continuous “1”-s (IDLE) are transmitted thereafter.

2.5.4.3 CRC Parity Inhibit

If the internal CRC generator is not used for calculation of Frame Check Sequence, an externally calculated checksum (16 bits) can be appended to the message frame without internally generated parity information, although parity is enabled for data characters.

Prerequisites are:

- CRC generator disabled (CAPP = 0),
- CON = 0 for all data characters which are to be included into parity generation (normal operation),
- CON = 1 for both bytes defining the CRC checksum,
- Message End indication has to be issued after the checksum is written to XFIFO.

The programmed character length has no influence on this function.

2.6 Serial Interface (Layer-1 functions)

The eight serial interfaces of the ESCC8 provide eight fully independent communication channels, supporting layer-1 functions to a high degree by various means of clock generation and clock recovery.

Note: Since the eight serial channels are completely independent, the functions described in this document apply to all eight channels. For simplification purposes the indices 0 to 7 will usually be omitted from the signal names, and are implied.

2.6.1 Clock Modes

The ESCC8 includes an internal Oscillator (OSC) as well as independent Baud Rate Generator (BRG) and Digital Phase Locked Loop (DPLL) circuitry for each serial channel.

The transmit and receive clock can be generated either

– externally, and supplied via the R×CLK and/or T×CLK pins, or

– internally, by means of the

- OSC and/or BRG, and
- DPLL, recovering the receive (+ optionally transmit) clock from the received data stream.

There are a total of 8 different clocking modes programmable via the CCR1 register, providing a wide variety of clock generation and clock pin functions, as shown in **table 4.**

Table 4
Overview of Clock Modes

Clock			
Type	**Source**	**Generation**	**Mode**
Receive Clock	RxCLK Pins	Externally	0, 1, 5
	DPLL	Internally	2, 3a, 6, 7a
	OSC		4
	BRG		3b, 7b
Transmit Clock	TxCLK Pins	Externally	0a, 2a, 6a
	RxCLK Pins		1, 5
	DPLL	Internally	3a, 7a
	BRG ./. 16		2b, 6b
	OSC		4
	BRG		0b, 3b, 7b

The transmit clock pins (TxCLK) may also output clock signals in certain clock modes if enabled via CCR2.TOE.

The clocking source for the DPLL's is always the internal BRG; the scaling factor (divider) of the BRG can be programmed through CCR2 and BGR registers between 1, 2, 4, 6…2048.

The ESCC8's system clock is always derived from the transmit clock or from the master clock (if master clock mode is enabled).

Master Clock Capabilities

A new clock source can be defined as master clock to allow full functionality of the microprocessor interface (access to all status and control registers and FIFOs, DMA and interrupt support) independent from the receive and transmit clocks. This new function (enabled via bit CCR0.MCE) is useful for modem applications where continuous generation of the receive and especially of the transmit clock cannot be guaranteed. The master clock has to be supplied via pin XTAL1 (or a crystal connected to XTAL1-2). Depending on the version, the maximum clock rate is 10 MHz (SAB 82538H-10) or 2 MHz (SAB 82538H).

Notes:

- The master clock is applicable to all clock modes except clock mode 5. For details refer to **table 5**.
- If bus configuration is selected in HDLC/SDLC mode (CCR0.SC2 .. 0), the One-Insertion (CCR1.OIN) cannot be used in conjunction with the Master Clock feature.
- In SDLC Loop mode the Master clock option is not available.
- The conditions for the ratio between transmit clock, master clock and receive clock frequencies must be fulfilled to guarantee correct function (**refer to the notes of table 5**).
- The internal timers run with the master clock.
- The serial interface (transmitter and receiver) are not sequenced by the master clock however the FIFOs, DMA-UNIT and TIMER are.

Clock Mode 0 (External Clocks)

Separate, externally generated receive and transmit clocks are supplied to the ESCC8 via their respective pins. The transmit clock can be directly supplied by pin TxCLK (mode 0a) or generated by the internal baud rate generator from the clock supplied by pin XTAL1 (mode 0b). In the latter case, the transmit clock can be output via pin TxCLK.

Clock Mode 1 (Re./Trm. Strobes)

Externally generated, but identical receive and transmit clocks are supplied via RxCLK. In addition, a receive strobe can be connected via CD and a transmit strobe via TxCLK. These strobe signals work on a per bit basis. The operating mode can be applied in time division multiplex applications or for adjusting disparate transmit and receive data rates.

Note: In Extended Transparent Mode (HDLC/SDLC), the above mentioned strobe signals provide byte synchronization (byte alignment).

Clock Mode 2 (Rec. Clock from DPLL)

The BRG is driven by an external clock (R×CLK) and it delivers a reference clock of a frequency equal to 16 times the nominal bit rate for the DPLL which in turn generates the receive clock. Depending on the programming of the CCR2 register (bit SSEL), the transmit clock will be either an external clock signal (T×CLK) or the clock delivered by the BRG divided by 16. In the latter case, the transmit clock can be output via T×CLK.

Clock Mode 3 (Rec. and Trm. Clock from DPLL)

The BRG is fed with an externally generated clock via R×CLK. Depending on the value of bit CCR2.SSEL the BRG supplies either the reference clock of frequency equal to 16 times the nominal bit rate for the DPLL, which generates **both** the receive and transmit clock, or, the receive and transmit clock directly. This clock can be output via T×CLK.

Clock Mode 4 (OSC-Direct)

The receive and transmit clocks are **directly** supplied by the OSC. In addition, this clock can be output via T×CLK.

Clock Mode 5 (Time-Slots)

This operation mode has been designed for application in time-slot oriented PCM systems.

Note: Clock mode 5 is only specified for version SAB 82538H-10, but not for version SAB 82538H.
For correct operation only NRZ coding should be used

The receive and transmit clocks are common and must be supplied externally via RxCLK pin. The ESCC8 receives and transmits only during certain time-slots

– of programmable width (1…256 bit, via RCCR and XCCR registers), and
of programmable location with respect to a frame synchronization signal (via CD pin).

One of up to 64 time-slots can be programmed independently for receive and transmit direction via TSAR and TSAX registers, and an additional clock shift of 0…7 bits via TSAR, TSAX and CCR2 register. Together with bits XCS0 and RCS0 (LSB of clock shift), located in the CCR2 register, there are 9 bits to determine the location of a time-slot.

Depending to the value programmed via those bits, the receive/transmit window (time-slot) starts with a delay of 1 (minimum delay) up to 512 clock periods following the frame synchronization signal and is active for the number of clock periods programmed via RCCR, XCCR (number of bits to be received/transmitted within a time-slot) as shown in **figure 35.**

If bit CCR2.TOE is set, the transmit time-slot is indicated by a control signal via TxCLK, which is set to “low” during the transmit window.

Note: In HDLC/SDLC Extended Transparent modes above windows provide character synchronization (byte aligned). In extended transparent mode the width of the time-slots has to be nx8 bit. In all other modes they can be used to define windows down to a minimum length of one bit.

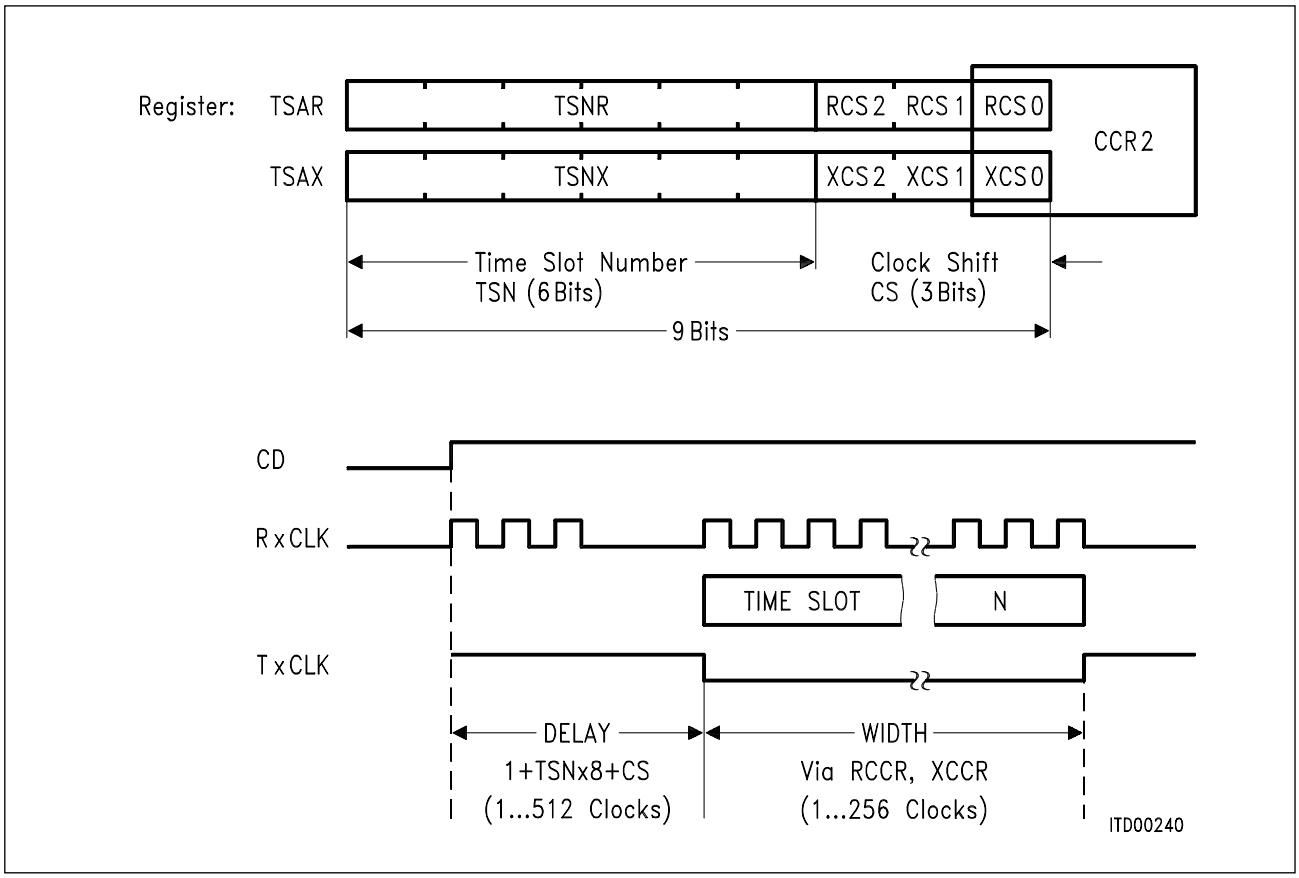


Figure 35
Location of Time-Slots

Clock Mode 6 (OSC - Rec. Clock from DPLL)

This clock mode is identical to clock mode 2 except that the clock for the BRG is delivered by the OSC and must not be supplied externally.

Clock Mode 7 (OSC - Rec. and Trm. Clock from DPLL)

Similar to clock mode 3, but BRG clock is provided by OSC.

Summary

The features of the different clock modes are summarized in **table 5**.

Table 5
Clock Modes of ESCC8

Channel Configur.		Clock Sources					Control Sources				
Clock Mode CCR1 CM2.CM1.CM0	CCR2, SSEL	Master Clock CCR0.MCE=1	BRG	DPLL	REC	TRM	CD	R-Strobe	X-Strobe	Frame-Sync	Output via TxCLK (if CCR2.TOE=1)
0a	0	OSC	–	–	RxCLK	TxCLK	CD	–	–	–	–
0b	1	OSC	OSC	–	RxCLK	BRG	CD	–	–	–	BRG
1	X	OSC	–	–	RxCLK	RxCLK	–	CD	TxCLK	–	–
2a	0	OSC	RxCLK	BRG	DPLL	TxCLK	CD	–	–	–	–
2b	1	OSC	RxCLK	BRG	DPLL	BRG/16	CD	–	–	–	BRG/16
3a	0	OSC	RxCLK	BRG	DPLL	DPLL	CD	–	–	–	DPLL
3b	1	OSC	RxCLK	–	BRG	BRG	CD	–	–	–	BRG
4	X	OSC	–	–	OSC	OSC	CD	–	–	–	OSC
5	X	–	–	–	RxCLK	RxCLK	–	(TSAR)	(TSAX)	CD	TS-Control
6a	0	OSC	OSC	BRG	DPLL	TxCLK	CD	–	–	–	–
6b	1	OSC	OSC	BRG	DPLL	BRG/16	CD	–	–	–	BRG/16
7a	0	OSC	OSC	BRG	DPLL	DPLL	CD	–	–	–	DPLL
7b	1	OSC	OSC	–	BRG	BRG	CD	–	–	–	BRG

Notes:

- If ASYNC Mode is programmed, the baud rate depends on the Bit Clock Rate (1 or 16) selected by bit CCR1.BCR:

Clock Mode	Receive	Transmit
0a	RxCLK/BCR	TxCLK
0b	RxCLK/BCR	BRG
1	RxCLK/BCR	RxCLK/BCR
3b, 7b	BRG/BCR	BRG/BCR
4	OSC/BCR	OSC/BCR

  When BCR is set to “16”, oversampling (3 samples) in conjunction with majority decision is performed. BCR has no effect when using clock mode 2, 3a, 6, or 7a.

- Restrictions for frequency ratios between receive frequency (fr), transmit frequency (fx) and master clock frequency (fm):

  Normal mode; clock mode 0, 2a, and 6a: fr/fx < 3 (*)

  Master clock mode: fm/fx ≥ 2.5 for clocks (fm and fx) with about 50 % (± 5 %) duty cycle (*);

  fr/fm < 3 (**)

  (*) for unsymmetrical clocks higher ratios have to be provided, for example:

fm (high time)	fm (low time)	fx (high time)	fx (low time)	ratio
50 %	50 %	70 %	30 %	> 4
50 %	50 %	75 %	25 %	> 5

  (**) reduced to 1.5 if receive address is pushed to RFIFO in HDLC/SDLC mode.

  There are no restrictions on the relative phases of the clocks. The conditions are valid independent of strobe signals or time-slot widths: i.e. in normal mode clock mode 1 always fulfils the condition, irrespective of how receive and transmit data are strobed. Thus, by using strobes the above condition may always be fulfilled irrespective of the net data rates.

- If one of the clock modes 0b, 4, 6 or 7 or the master clock is selected the internal oscillator (OSC) is enabled which allows connection of an external crystal to pins XTAL1-XTAL2. The output signal of the OSC can be used for one serial channel, or for all serial channels (independent baud rate generators and DPLLs). Moreover, XTAL1 alone can be used as input for an externally generated clock.

2.6.2 Clock Recovery (DPLL)

The ESCC8 offers the advantage of recovering the received clock from the received data by means of internal DPLL circuitry, thus eliminating the need to transfer additional clock information via the serial link. For this purpose, the DPLL is supplied with a “reference clock” from the BRG which is 16 times the nominal data clock rate (clock mode 2, 3a, 6, 7a). The transmit clock may be obtained by dividing the output of the BRG by a constant factor of 16 (clock mode 2b, 6b; bit SSEL in CCR2 set) or also directly from the DPLL (clock mode 3a, 7a).

The main task of the DPLL is to derive a receive clock and to adjust its phase to the incoming data stream in order to enable optimal bit sampling.

The mechanism for clock recovery depends on the selected data encoding (**refer to chapter 2.6.4**).

The following functions have been implemented to facilitate a fast and reliable synchronization:

– Interference Rejection and Spike Filtering

In the case where two or more edges appear in the data stream within a time period of 16 reference clocks, these are considered as interference and consequently no additional clock adjustment is performed.

– Phase Adjustment

In the case where an edge appears in the data stream within the PA fields of the time window, the phase will be adjusted by 1/16 of the data clock.

– Phase Shift (NRZ, NRZI only)

In the case where an edge appears in the data stream within the PS fields of the time window, a second sampling of the bit is forced and the phase is shifted by 180 degrees.

– Edges in all other parts of the time window will be ignored.

This operation facilitates a **fast** and reliable synchronization for most common applications. Above all, it implies a very fast synchronization because of the phase shift feature: one edge on the received data stream is enough for the DPLL to synchronize, thereby eliminating the need for synchronization patterns sometimes called preambles. However, in case of **extremely** high jitter of the incoming data stream the reliability of the clock recovery cannot be guaranteed.

The version 2 of ESCC8 offers the option to disable the Phase Shift function for NRZ and NRZI encodings by setting bit CCR3.PSD. In this case, the PA fields are extended as shown in **figure** .

Now, the DPLL is more insensitive to high jitter amplitudes but needs more time to reach the optimal sampling position. To ensure correct data sampling preambles should precede the data information.

Figures 36, and 38 explain the DPLL algorithms used for the different data encodings.

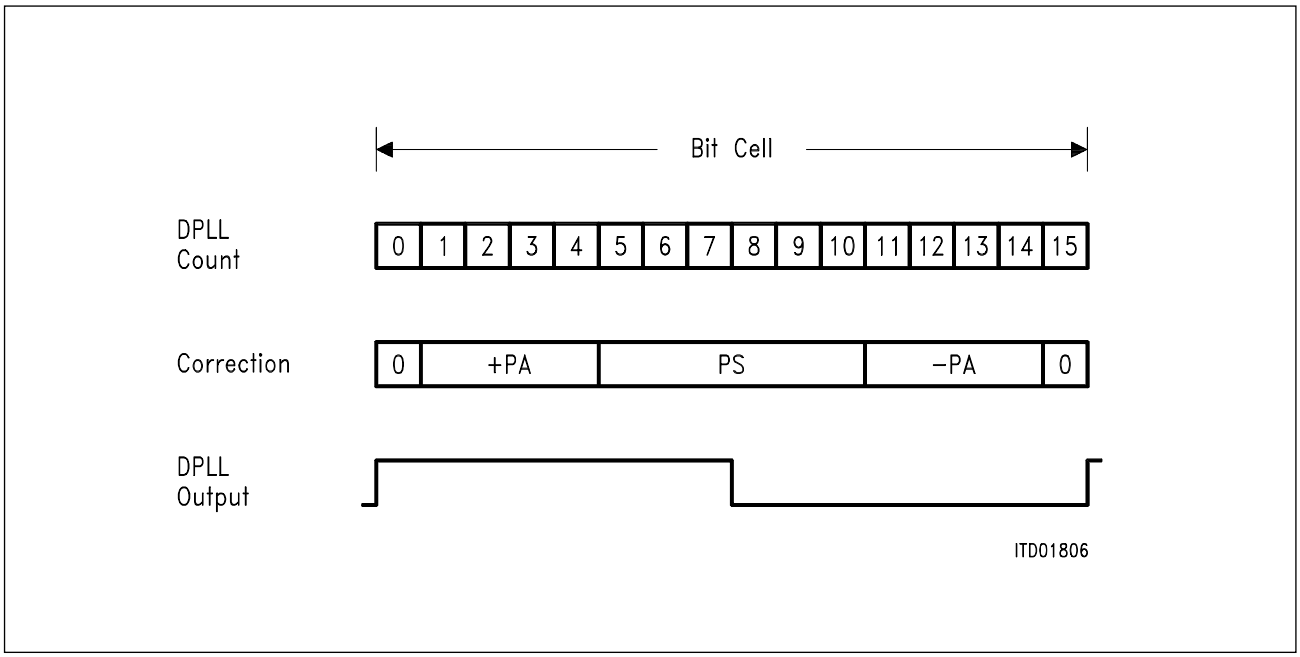


Figure 36
DPLL Algorithm for NRZ and NRZI Coding with Phase Shift Enabled (CCR3.PSD = 0)

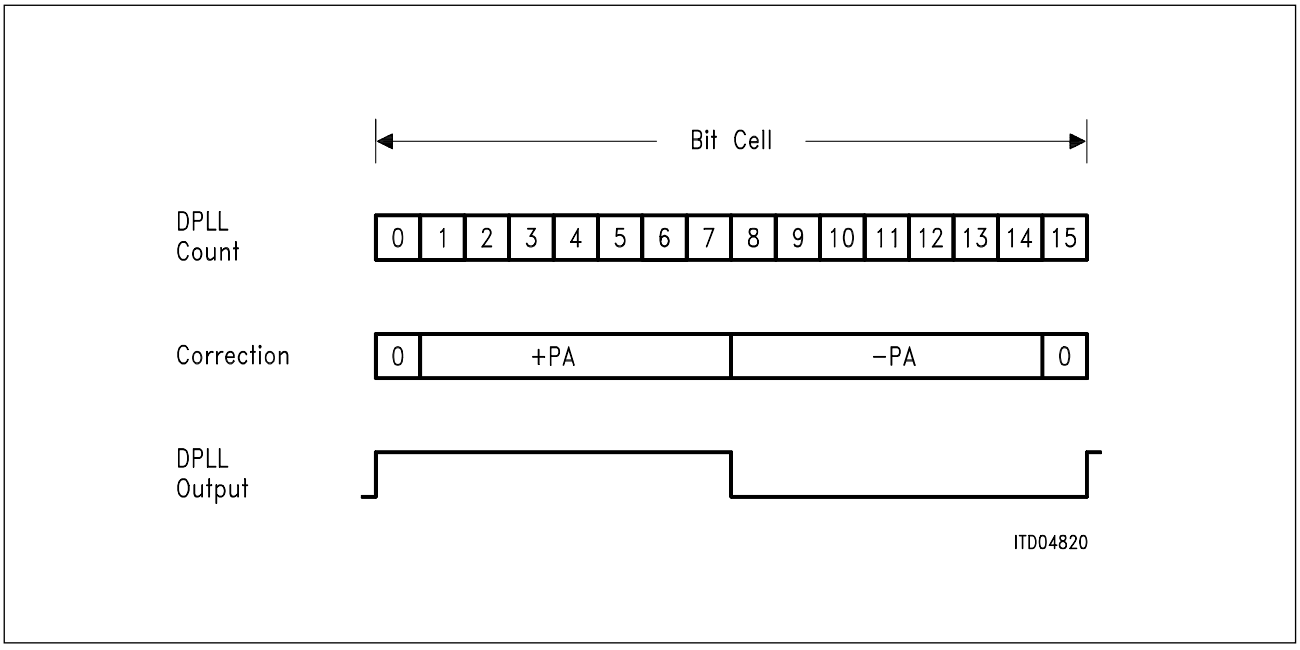


Figure 37
DPLL Algorithm for NRZ and NRZI Encoding with Phase Shift Disabled (CCR3.PSD = 1)

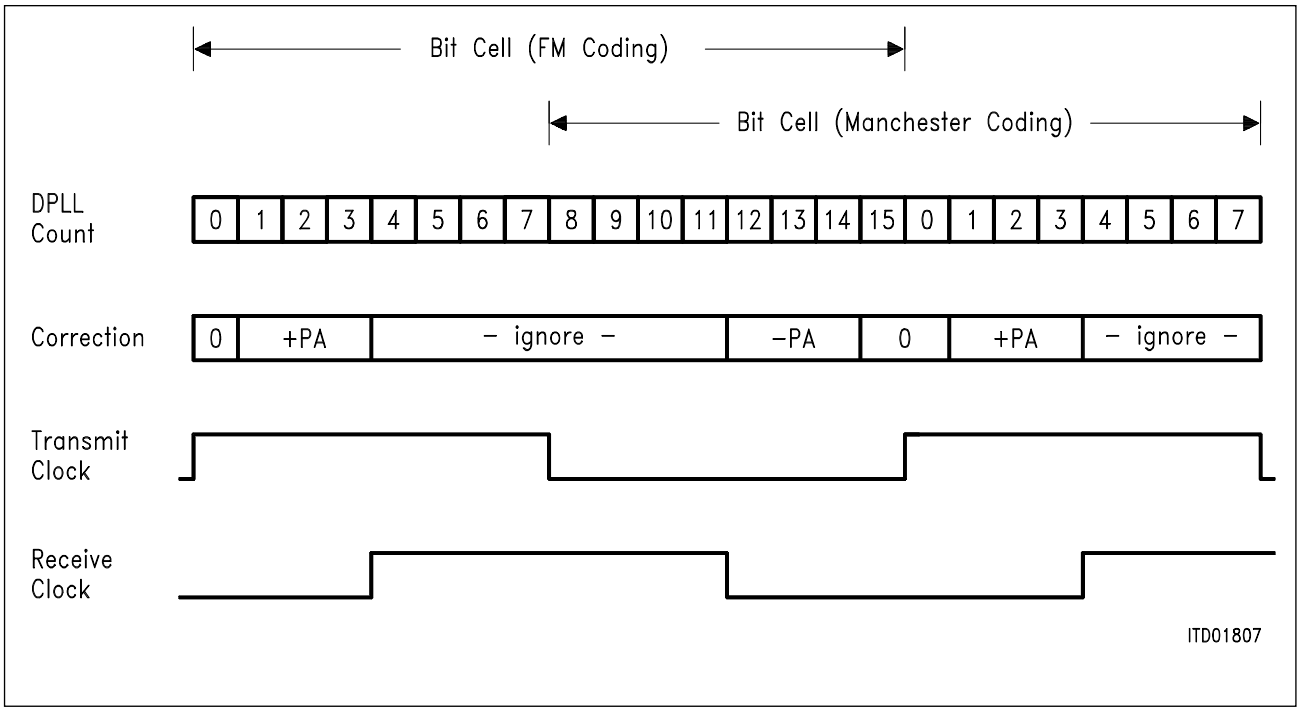


Figure 38
DPLL Algorithm for FM0, FM1 and Manchester Coding

To supervise correct function when using bi-phase encoding, a status flag and a maskable interrupt inform about synchronous/asynchronous state of the DPLL.

2.6.3 Bus Configuration

Beside the point-to-point configuration, the ESCC8 effectively supports point-to-multipoint (pt-mpt or bus) configurations by means of internal idle and collision detection/ collision resolution methods.

In a pt-mpt configuration, comprising a central station (master) and several peripheral stations (slaves), or in a multimaster configuration (**see figure 13**), data transmission can be initiated by each station over a common transmit line (bus). In case more than one station attempt to transmit data simultaneously (collision), the bus has to be assigned to one station.

– In HDLC/SDLC mode, a collision-resolution procedure is implemented by the ESCC8. Bus assignment is based on a priority mechanism with rotating priorities. This allows each station a bus access within a predetermined maximum time delay (deterministic CSMA/CD), no matter how many transmitters are connected to the serial bus.

– In BISYNC mode, the collision-resolution is implemented by the microprocessor.

– In ASYNC mode, a bus configuration is not recommended.

Prerequisites for bus operation are:

- NRZ encoding
- OR'ing of data from every transmitter on the bus (this can be realized as a wired-or, using the TxD open drain capability)
- Feedback of bus information (CxD input).

The bus configuration is selected via the CCR0 register.

Note: Central clock supply for each station is not necessary if both the receive and transmit clock is recovered by the DPLL (clock modes 3a, 7a). This minimizes the phase shift between the individual transmit clocks.

The bus mode can be operated independently of the clock mode, e.g. also during clock mode 1 (receive and transmit strobe).

2.6.3.1 Bus Access Procedure

The idle state of the bus is identified by eight or more consecutive 1's. When a device starts transmission of a frame, the bus is recognized to be busy by the other devices at the moment the first zero is transmitted (e.g. first zero of the opening flag in HDLC mode).

After the frame has been transmitted, the bus becomes available again (idle).

Note: If the bus is occupied by other transmitters and/or there is no transmit request in the ESCC8, logical 1 will be continuously transmitted on T×D.

2.6.3.2 Collisions

During the transmission, the data transmitted on T×D is compared with the data on C×D. In case of a mismatch (1 sent and 0 detected, or vice versa) data transmission is immediately aborted, and idle (logical 1) is transmitted.

HDLC/SDLC: Transmission will be initiated again by the ESCC8 as soon as possible if the first part of the frame is still present in the XFIFO. If not, an XMR interrupt is generated.
Since a zero (“low”) on the bus prevails over a 1 (high impedance) if a wired-or connection is implemented, and since the address fields of the HDLC frames sent by different stations normally differ from one another, the fact that a collision has occurred will be detected prior to or at the latest within the address field. The frame of the transmitter with the highest temporary priority (determined by the address field) is not affected and is transmitted successfully. All other stations cease transmission immediately and return to bus monitoring state.

BISYNC: Transmitter and XFIFO are reset and pin T×D goes to “1”. The XMR interrupt is provided which requests the microprocessor to repeat the whole message or block of characters.

ASYNC: Bus configuration not recommended.

Note: If a wired OR connection has been realized by an external pull-up resistor without decoupling, the data output (T×D) can be used as an open drain output and connected directly to the C×D input.
For correct identification as to which frame is aborted and thus has to be repeated after an XMR interrupt has occurred, the contents of XFIFO have to be unique, i.e. XFIFO should not contain data of more than one frame as it could happen when servicing is done after an XPR interrupt. For this purpose the All Sent interrupt (ISR1.ALLS) instead of XPR has to be used to trigger the loading of data (for the next frame) into XFIFO.

2.6.3.3 Priority (HDLC/SDLC Mode Only)

To ensure that all competing stations are given a fair access to the transmission medium, once a station has successfully completed the transmission of a frame, it is given a lower level of priority. This priority mechanism is based on the requirement that a station may attempt transmitting only when a determined number of consecutive 1's are detected on the bus.

Normally, a transmission can start when eight consecutive 1's on the bus are detected (through pin C×D). When an HDLC frame has been successfully transmitted, the internal priority class is decreased. Thus, in order for the same station to be able to transmit another frame, ten consecutive 1's on the bus must be detected. This guarantees that the transmission requests of other stations are satisfied before a same station is allowed a second bus access. When ten consecutive 1's have been detected, transmission is allowed again and the priority class is increased (to "eight 1's").

Inside a priority class, the order of transmission (individual priority) is based on the HDLC address, as explained in the preceding paragraph. Thus, when a collision occurs, it is always the station transmitting the only zero (i.e. all other stations transmit a one) in a bit position of the address field that wins, all other stations cease transmission immediately.

2.6.3.4 Timing Modes

If a bus configuration has been selected, the ESCC8 provides two timing modes, differing in the time interval between sending data and evaluation of the transmitted data for collision detection.

- Timing mode 1 (CCR0: SC1, SC0 = 01)
  Data is output with the rising edge of the transmit clock via the TxD pin, and evaluated 1/2 at the CxD pin clock period later with the falling clock edge.
- Timing mode 2 (CCR0: SC1, SC0 = 11)
  Data is output with the falling clock edge and evaluated with the next falling clock edge. Thus one complete clock period is available between the instant when data is output and collision detection.

2.6.3.5 Functions of $\overline{\text{RTS}}$ Output

In clock modes 0, 1 and 4, the $\overline{\text{RTS}}$ output can be programmed via CCR2 (SOC bits) to be active when data (frame or character) is being transmitted. This signal is delayed by one clock period with respect to the data output T×D, and marks all data bits that could be transmitted without collision. In this way a configuration may be implemented in which the bus access is resolved on a local basis (collision bus) and where the data are sent one clock period later on a separate transmission line.

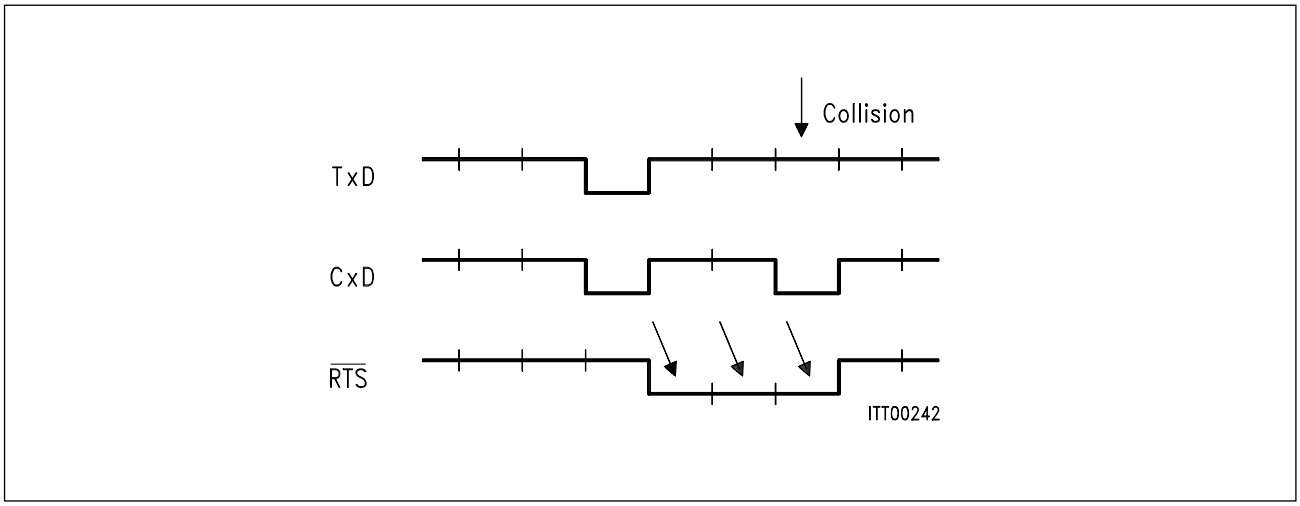


Figure 39
Request-to-Send in Bus Operation

Note: For details on the functions of the RTS pin **refer to chapter 2.6.5**.

2.6.4 Data Encoding

The ESCC8 supports the following coding schemes for serial data:

– Non-Return-To-Zero (NRZ)

– Non-Return-To-Zero-Inverted (NRZI)

– FM0 (also known as Bi-Phase Space)

– FM1 (also known as Bi-Phase Mark)

– Manchester (also known as Bi-Phase)

NRZ: The signal level corresponds to the value of the data bit. By programming bit DIV (CCR2 register) the ESCC8 may made to transmit and receive data inverted.

NRZI: A logical “0” is indicated by a transition and a logical “1” by no transition at the beginning of the bit cell.

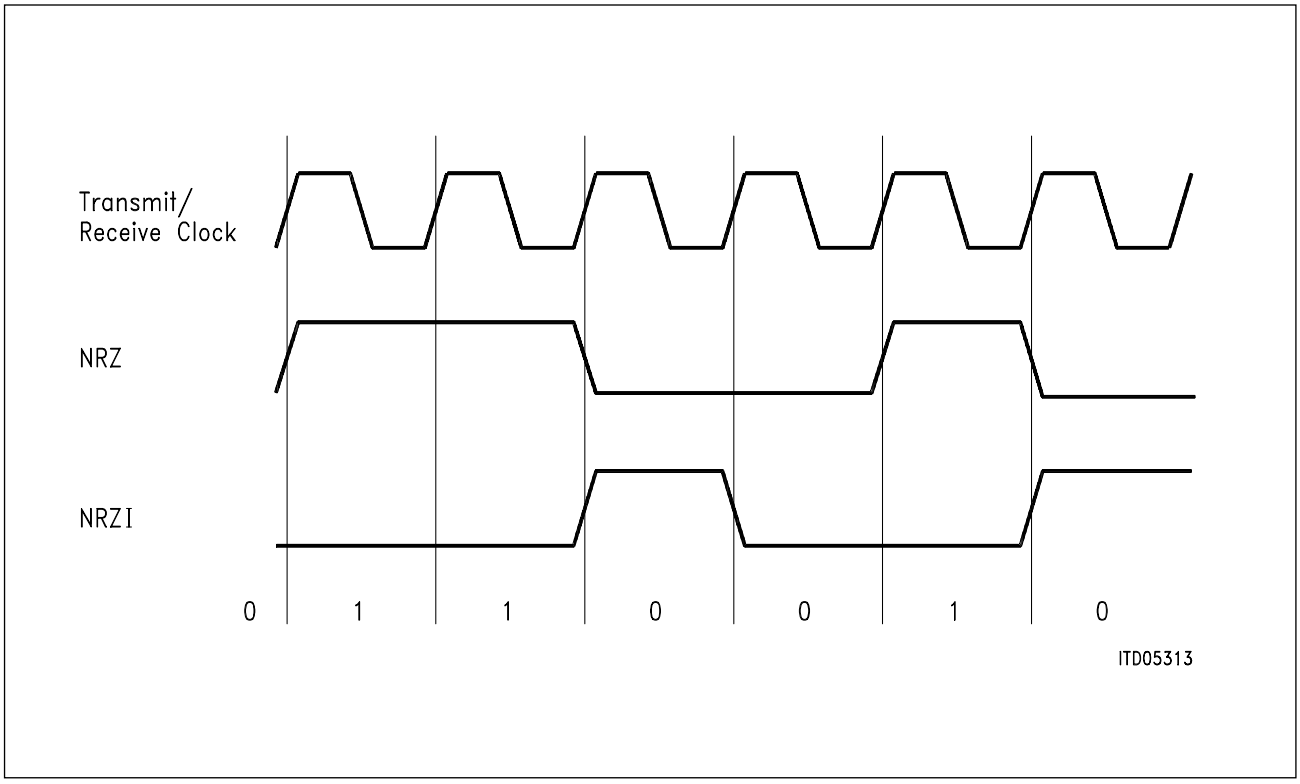


Figure 40
NRZ and NRZI Data Encoding

FM0: An edge occurs at the beginning of every bit cell. A logical “0” has an additional edge in the center of the bit cell, a logical “1” has none. The transmit clock precedes the receive clock by 90°.

FM1: An edge occurs at the beginning of every bit cell. A logical “1” has an additional edge in the center of the bit cell, a logical “0” has none. The transmit clock precedes the receive clock by 90°.

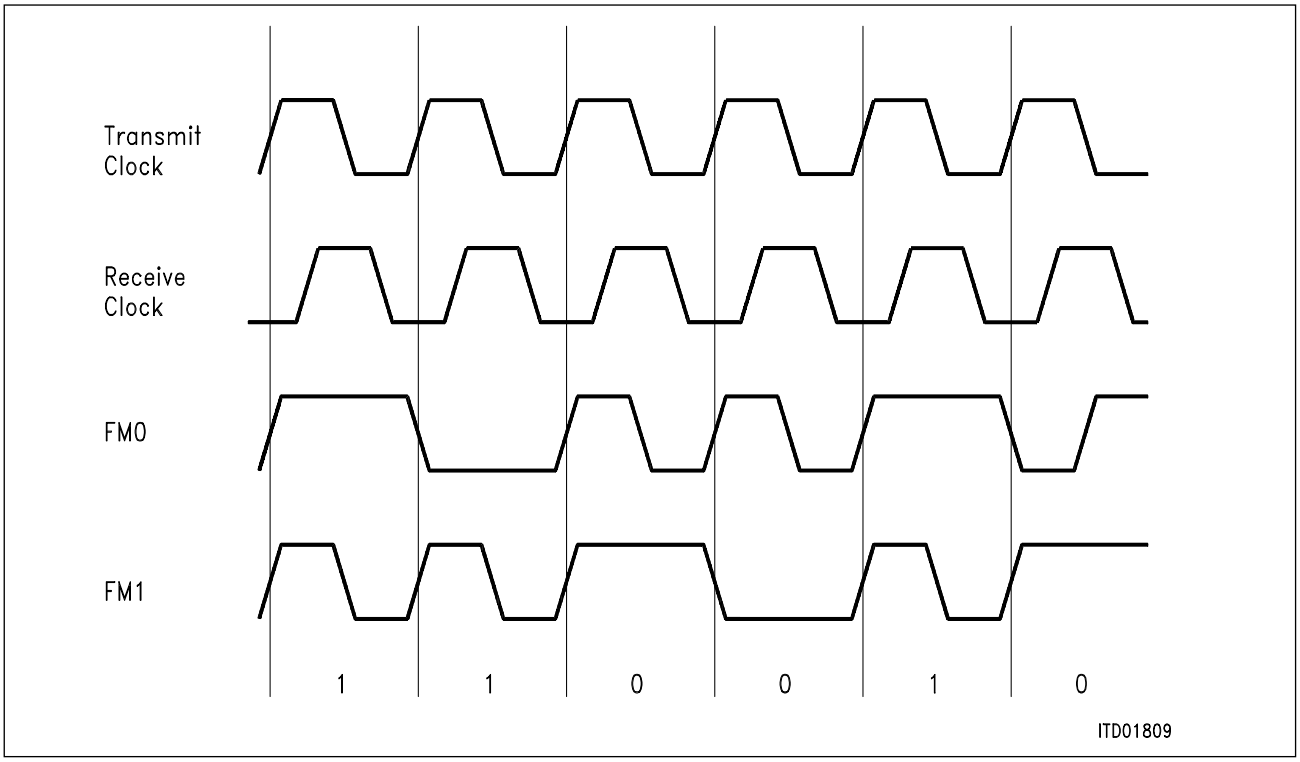


Figure 41
FM0 and FM1 Data Encoding

Manchester: In the first half of the bit cell the physical signal level corresponds to the logical value of the data bit. At the center of the bit cell this level is inverted. The transmit clock precedes the receive clock by 90°. The bit cell is shifted by 180° in comparison with FM coding.

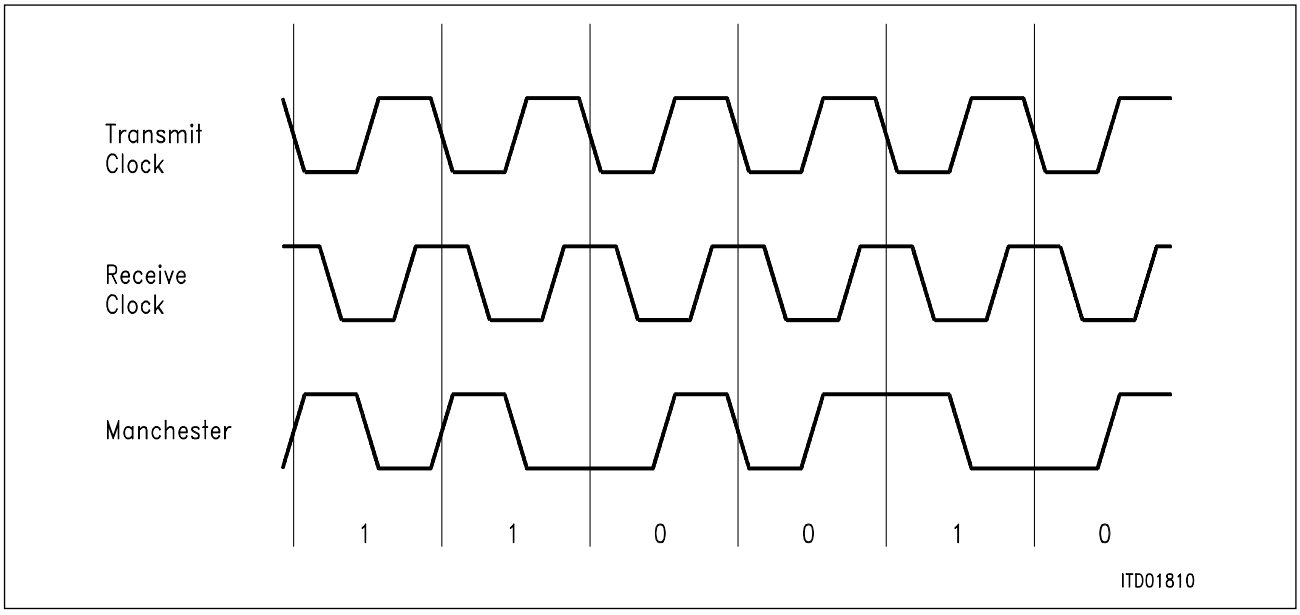


Figure 42
Manchester Data Encoding

2.6.5 Modem Control Functions ($\overline{\text{RTS}}$/$\overline{\text{CTS}}$, CD)

2.6.5.1 $\overline{\text{RTS}}$/$\overline{\text{CTS}}$ Handshaking

The ESCC8 provides two pins ($\overline{\text{RTS}}$, $\overline{\text{CTS}}$) per serial channel supporting the standard RTS-modem handshaking procedure for transmission control.

A transmit request will be indicated by outputting logical “0” on the request-to-send output ($\overline{\text{RTS}}$). It is also possible to control the $\overline{\text{RTS}}$ output by software. After having received the permission to transmit ($\overline{\text{CTS}}$) the ESCC8 starts data transmission.

HDLC/SDLC and **BISYNC:** In the case where permission to transmit is withdrawn in the course of transmission, the frame is aborted and IDLE is sent. After transmission is enabled again by re-activation of $\overline{\text{CTS}}$, and if the beginning of the frame is still available in the ESCC8, the frame will be re-transmitted (self-recovery). However, if the permission to transmit is withdrawn after the data in the first XFIFO pool has been completely transmitted and the pool is released, the transmitter and the XFIFO are reset, the $\overline{\text{RTS}}$ output is deactivated and an interrupt (XMR) is generated.

Note: For correct identification as to which frame is aborted and thus has to be repeated after an XMR interrupt has occurred, the contents of XFIFO have to be unique, i.e. XFIFO should not contain data of more than one frame, which could happen if transmission of a new frame is started by loading new data in XFIFO and issuing a transmit command upon reception of XPR interrupt. For this purpose the All Sent interrupt (ISR1. ALLS) instead of XPR has to be used to trigger the loading of data (for the next frame) into XFIFO.

ASYNC: In the case where permission to transmit is withdrawn, transmission of the current character is completed. After that, IDLE is sent. After transmission is enabled again by re-activation of CTS, the next available character is sent out.

Note: In the case where permission to transmit is not required, the CTS input can be connected directly to V_{SS}.

Additionally, any transition on the $\overline{\text{CTS}}$ input pin will generate an interrupt indicated via the ISR1 register, if this function is enabled by setting the CSC bit in the IMR1 register.

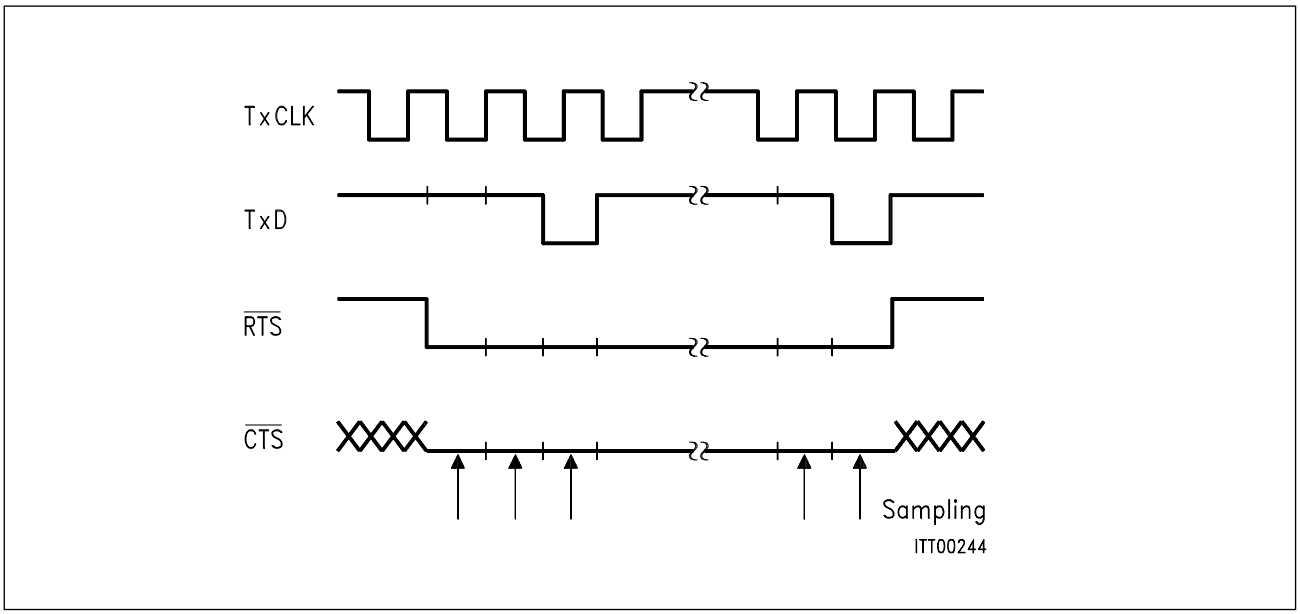


Figure 43
RTS – CTS Handshaking

Beyond this standard RTS function, signifying a transmission request of a frame (Request To Send), the RTS output may be programmed for a special function via SOC1, SOC0 bits in the CCR2 register, provided the serial channel is operating in a bus configuration in clock mode 0 or 1.

- If SOC1, SOC0 bits are set to “11”, the RTS output is active (= low) during the reception of a frame.
- If SOC1, SOC0 bits are set to “10”, the RTS output function is disabled and the RTS pin remains always high.

2.6.5.2 Carrier Detect (CD) Receiver Control

Similar to the RTS/CTS control for the transmitter, the ESCC8 supports the carrier detect modem control function for the serial receivers, if the Carrier Detect Auto Start (CAS) function is programmed by setting the CAS bit in the XBCH register. This function is always available in clock modes 0, 2, 3, 4, 6, 7 via the CD pin. In clock mode 1 the CD function is not supported. **See table 5** for an overview.

If the CAS function is selected, the receiver is enabled and data reception is started when the CD input is detected to be high. If CD input is set to “low”, reception of the current character (byte) is still completed.

2.6.6 Test Mode

To provide for fast and efficient testing, the ESCC8 can be operated in a test mode by setting the TLP bit in the MODE register.

The on-chip serial input and output (T×D-R×D) are connected, generating a local loopback.

As a result, the user can perform a self-test of the ESCC8.

2.7 Universal Port

Four general purpose bi-directional parallel ports are provided on pins PA0-7, PB0-7, PC0-7 and PD0-3. Every pin is separately programmable via the Port Configuration Register PCR to operate as an output or an input.

If defined as output, the state of the pin is directly controlled via Port Value Register PVR. A read-back is also provided.

If defined as input, the state of the pin is monitored. The value is readable via PVR. All changes may be (if desired) indicated via interrupt. Assigned registers: Port Interrupt Status register (PIS) and Port Interrupt Mask register (PIM).

3 Operational Description

3.1 Reset

The ESCC8 is forced into the reset state if the RES pin is set “high” for at least 5 microseconds. During RESET, the ESCC8 is temporarily in the power-up mode, and a subset of the registers is initialized with defined values.

During hardware reset

– all uni-directional output stages are in high-impedance state.

– all bi-directional output stages (data bus) are in high-impedance state if signals $\overline{RD}$ and $\overline{INTA}$ are “high”,

– “output” XTAL2 is high-impedance if input XTAL1 is “high” (the internal oscillator is disabled during reset).

After RESET, the ESCC8 is in power-down mode, and the following registers contain defined values:

Register	Reset Value	Meaning
CCR0	00_H	– Power down mode – HDLC/SDLC mode – NRZ coding
CCR1	00_H	– No Shared Flags – No SDLC Loop function – T×D pins are open drain outputs – pt – pt with IDLE as Interframe Time Fill – Clock mode 0
CCR2	00_H	– $\overline{RTS}$ pin standard function – READ/WRITE Exchange disabled – CRC-32 disabled – No data inversion
CCR3	00_H	– No Preambles – CRC reset level is “FFFF”$_H$– No ADDRESS to RFIFO – No CRC-Bytes to RFIFO – Transmit CRC OFF

Register	Reset Value	Meaning
MODE	00_H	– Auto/mode with 1 byte address field – External timer mode, timer resolution: k = 32768 – Receiver active – $\overline{\text{RTS}}$ output controlled by ESCC8 – No test loop
IMR0 IMR1 PIM	FF_H FF_H FF_H	– All interrupts masked
IPC	00_H	– Interrupt pin $\overline{\text{INT}}$ is an open drain output – Slave Cascading mode is enabled – Slave address is set to 00_H
PCR	FF_H	– All pins of the Universal Port are inputs
IVA	00_H	– Interrupt vector address is set to 00_H
PRE	00_H	– Preamble value is set to 00_H
XBCH	00_H	– Interrupt controlled data transfer (DMA disabled) – Full/duplex LAPB/LAPD operation of LAP controller – Carrier detect auto start of receiver disabled
STAR	48_H	– XFIFO write enabled – Receive line inactive – No commands executing
AML/MXN AMH/MXF	00_H 00_H	– Address mask disabled
TSAX TSAR	00_H	– Time-slot number: 00_H – Clock shift (together with CCR2 = 00_H): 00_H
XCCR RCCR	00_H	– 1-bit time-slot

3.2 Initialization

After Reset the CPU has to write a minimum set of registers and an optional set dependent on the required features and operating modes.

First, the serial mode, the configuration of the serial port and the clock mode have to be defined via the CCR0 and CCR1 registers. The clock mode must be set before power-up (CCR1). The CPU may switch the ESCC8 between power-up and power-down mode. This has no influence upon the contents of the registers, i.e. the internal state remains stored. In power-down mode however, all internal clocks and the oscillator circuitry are disabled, no interrupts are forwarded to the CPU (interrupts of universal port excluded). This state can be used as a standby mode, when the ESCC8 is temporarily not used, thus substantially reducing power consumption.

The ESCC8 should usually be initialized in Power-Down mode.

The need for programming further registers depends on the selected features (serial mode, clock mode specific features, operating mode, address mode, user demands). **Table 6** gives an overview about initialization of the control registers.

Table 6
Initialization of ESCC8

Item	Registers	Comment
Clock Mode Clock mode specific features	CCR0, CCR1 BGR, CCR2 TSAR, TSAX XCCR, RCCR	for Master clock mode for clock modes 2, 3, 4, 6, 7 for clock mode 5
Serial Mode	CCR0	
Serial Port Configuration	CCR0 CCR1 CCR2	encoding output driver select data inversion, RxD $\leftrightarrow$ T$\times$D

Table 6
Initialization of ESCC8 (cont'd)

Item	Registers	Comment
Serial Mode Specific Features		
HDLC/SDLC	MODE, TIMR	**refer**
	XAD1, XAD2	**to**
	RAH1, RAH2	**table 7**
	RAL1, RAL2	
	XBCH	NRM mode
	CCR1	shared flags, ITF / OIN
	CCR2	CRC32
	CCR3	CRC reset level, preamble
		CRC/ADDRESS-Bytes to RFIFO
		RFIFO Threshold
	CCR4	preamble
	PRE	receive length check
ASYNC	RLCR	bit clock rate
	CCR1	data format
	DAFO	RFIFO configuration
	RFC	termination character
	TCR	XON character
		XOFF character
BISYNC	MODE	BI-/MONO-Sync, SLEN
	SYNL, SYNH	SYN character
	DAFO	data format
	RFC	RFIFO configuration
	CCR3	CRC, preamble
	PRE	preamble
	TCR	termination character
User Demands		
Modem control lines	MODE, CCR2	$\overline{\text{RTS}}$ pins
	XBCH	CD pins
Parallel Port	PCR	port configuration
Interrupt features	IPC	port configuration, slave addr.
		cascading mode
	IVA	interrupt vector address
	IMR0, IMR1	interrupt masks
	PIM	
DMA features	XBCH	DMA
	CCR2	read/write exchange
Timer (external mode)	MODE, TIMR	

Table 7
HDLC Specific Register Setup

Operating Mode \ Address Mode	2 Byte Address Field (MODE.ADM = 1)	1 Byte Address Field (MODE.ADM = 0)
Auto	TIMR XAD1 XAD2	
	RAH2 RAH2 RAL1 RAL2 AML AMH	RAH1 set to 00_H RAL1 RAL2 AML
Non Auto	RAH2 RAH2 RAL1 RAL2 AML AMH	RAH1 set to 00_H RAL1 RAL2 AML
Transparent	RAH1 RAH2 AMH	– –

3.3 Operational Phase

After having performed the initialization, the CPU switches each individual channel of the ESCC8 into operational phase by setting the PU bit in the CCR0 register.

Initially, the CPU should bring the transmitter and receiver into a defined state by issuing a Transmitter Reset command (CMDR.XRES) and a Receiver Reset command (CMDR.RHR in HDLC/SDLC mode, CMDR.RRES in ASYNC and BISYNC mode). If data reception should be performed, the receiver must be activated by setting the bit MODE.RAC.

If no “Clear To Send” function is provided via a modem, the $\overline{\text{CTS}}$ pin(s) of the ESCC8 must be connected directly to ground in order to enable data transmission.

Now the ESCC8 is ready to transmit and receive data. Control of data transfer is mainly done by commands from CPU to ESCC8 via the CMDR register, and by interrupt indications from ESCC8 to CPU. Additional status information, which does not trigger an interrupt, is available in the STAR register.

3.3.1 Data Transmission

3.3.1.1 Interrupt Mode

In transmit direction 2 $\times$ 32 byte FIFO buffers (transmit pools) are provided for each channel. After checking the XFIFO status by polling the Transmit FIFO Write Enable bit (XFW in STAR register) or after a Transmit Pool Ready (XPR) interrupt, up to 32 bytes may be entered by the CPU into the XFIFO.

HDLC/SDLC: The transmission of a frame can be started by issuing a XTF or XIF command via the CMDR register. If enabled, a specified number of preambles (refer to registers CCR3 and PRE) are sent out optionally before transmission of the current frame starts. If the transmit command does not include an end of message indication (CMDR: XME), the ESCC8 will repeatedly request for the next data block by means of a XPR interrupt as soon as no more than 32 bytes are stored in the XFIFO, i.e. a 32-byte pool is accessible to the CPU.

This process will be repeated until the CPU indicates the end of message per XME command, after which frame transmission is finished correctly by appending the CRC and closing flag sequence. Consecutive frames may share a flag (enabled via CCR1.SFLG) or may be transmitted as back-to-back frames, if service of XFIFO is quick enough.

In case no more data is available in the XFIFO prior to the arrival of XME, the transmission of the frame is terminated with an abort sequence and the CPU is notified per interrupt (ISR1.XDU). The frame may also be aborted per software (CMDR: XRES). The data transmission sequence, from the CPU’s point of view, is outlined in **figure 44**.

ASYNC: The transmission of character(s) can be started by issuing a XF command via the CMDR register. The ESCC8 will repeatedly request for the next data block by means of a XPR interrupt as soon as no more than 32 bytes are stored in the XFIFO, i.e. a 32-byte pool is accessible to the CPU. Transmission may be aborted per software (CMDR.XRES).

BISYNC: The transmission of a block can be started by issuing a XF command via the CMDR register. Further handling of data transmission with respect to preamble transmission and command XME is similar to HDLC/SDLC mode. After XME command has been issued, the block is finished by appending the internally generated CRC if enabled (refer to description of register CCR3).

In case no more data is available in the XFIFO prior to the arrival of XME, the transmission of the block is terminated with IDLE and the CPU is notified per interrupt (ISR1.XDU). The block may also be aborted per software (CMDR.XRES). The data transmission flow, from the CPU's point of view, is outlined in **figure 44.**

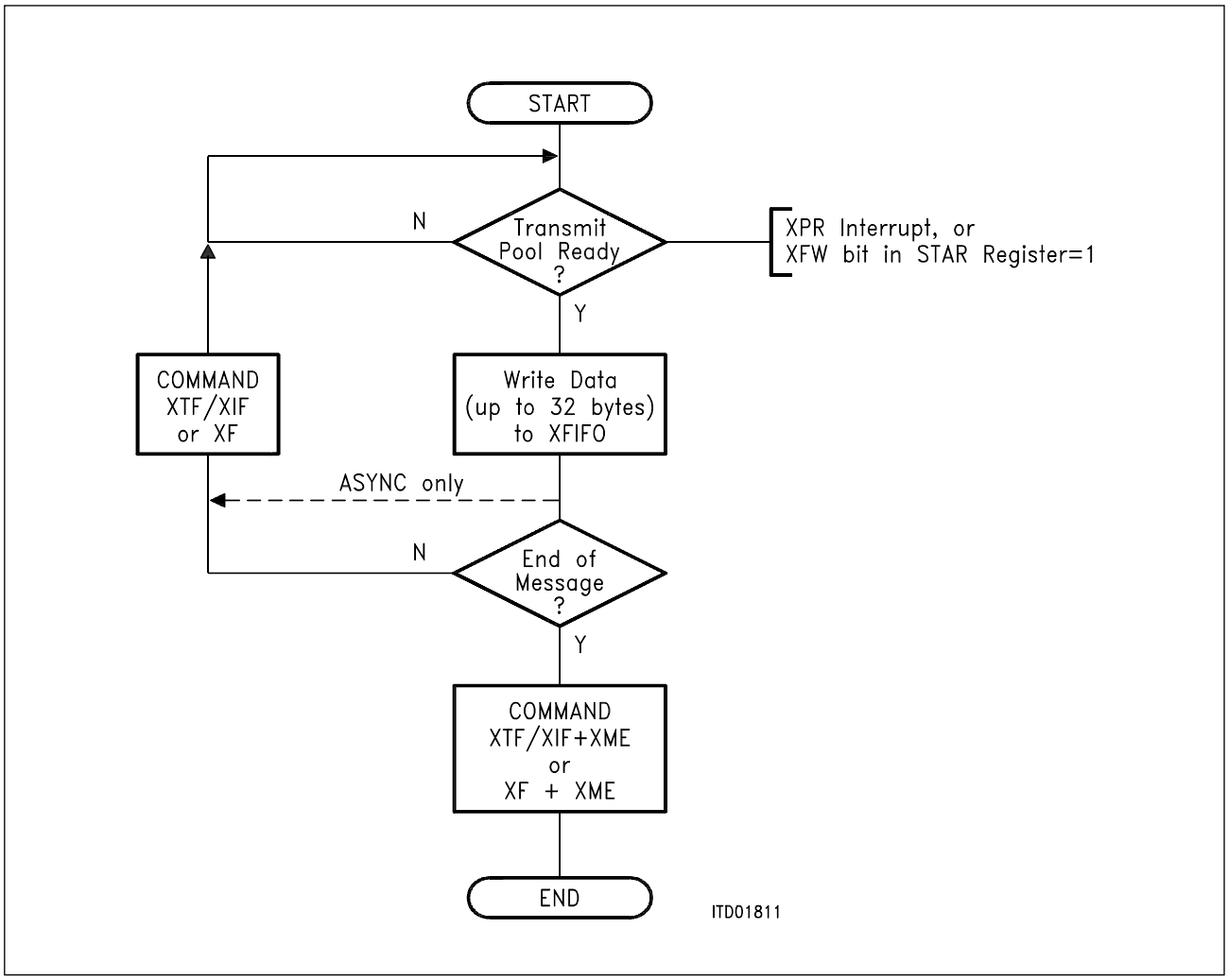


Figure 44
Interrupt Driven Data Transmission (Flow Diagram)

The activities at both serial and CPU interface during frame transmission (supposed frame length = 70 bytes) are shown in **figure 45**.

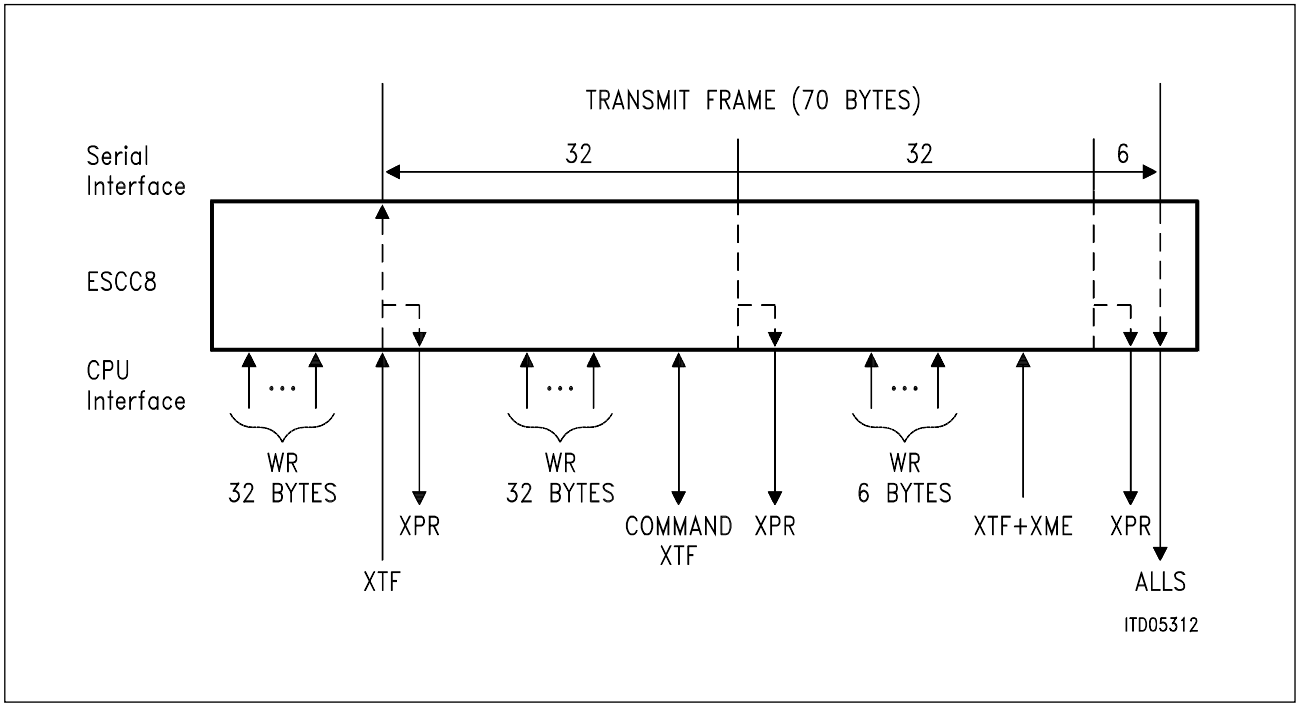


Figure 45
Interrupt Driven Transmission Sequence Example (HDLC)

3.3.1.2 DMA Mode

Prior to data transmission, the length of the next frame (or the next block of characters) to be transmitted must be programmed via the Transmit Byte Count Registers (XBCH, XBCL). The resulting byte count equals the programmed value plus one byte, i.e. since 12 bits are provided via XBCH, XBCL (XBC11…XBC0) a frame length of 1 up to 4096 bytes (4 Kbytes) can be selected.

After this, data transmission can be initiated by command (XTF or XIF in HDLC/SDLC mode, XF in ASYNC and BISYNC mode). The ESCC8 will then autonomously request the correct amount of write cycles by activating the DRT line for as long as necessary, taking into account the selected data bus width (i.e. byte or word accesses). For a frame length of L = (n $\times$ 32 + remainder) bytes (n = 0, 1,…,128), block data transfers of 32 bytes/16 words or remainder ($\div$2) bytes (words) are requested whenever a 32-byte FIFO half (transmit pool) is empty and accessible to the DMA controller.

The following figure gives an example of a DMA driven transmission sequence with a supposed frame length of 70 bytes, i.e. programmed transmit byte count (XCNT) equal to 69 bytes.

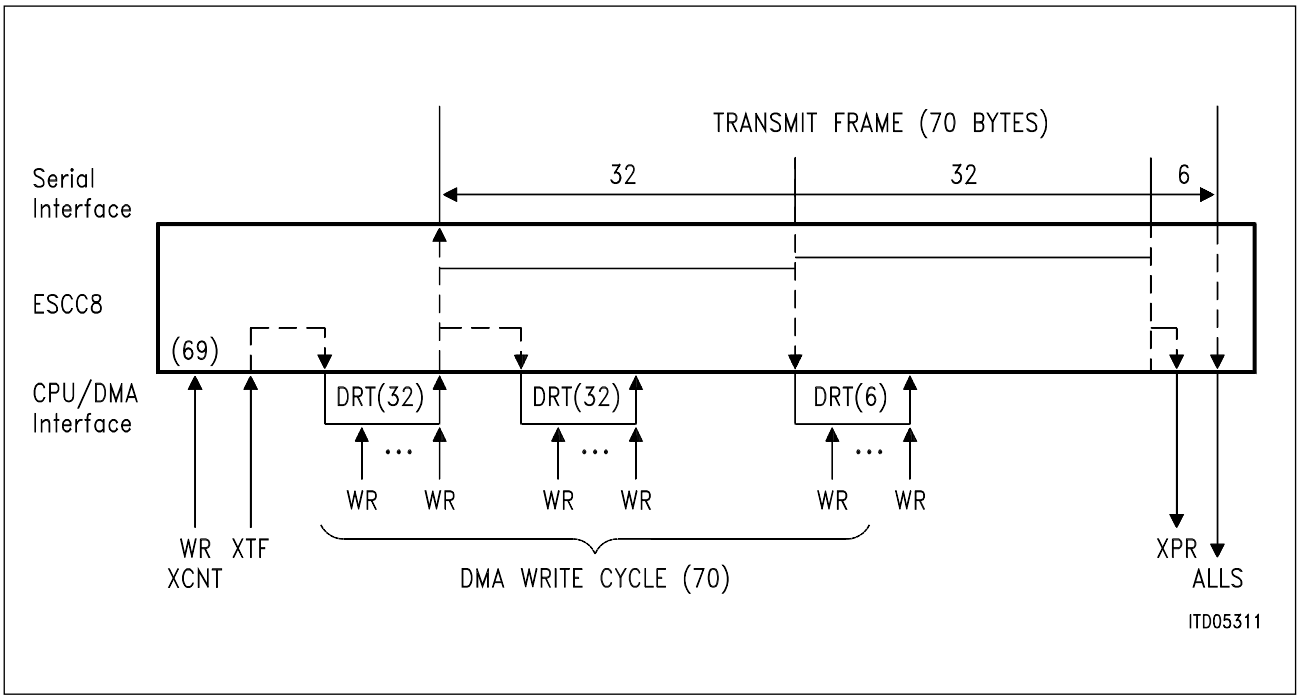


Figure 46
DMA Driven Transmission Sequence Example (HDLC)

3.3.2 Data Reception

3.3.2.1 Interrupt Mode

Also 2 × 32 byte FIFO buffers (receive pools) are provided for each channel in receive direction.

There are different interrupt indications concerned with the reception of data:

HDLC/SDLC

- RPF (Receive Pool Full) interrupt, indicating that a 32-byte block of data can be read from RFIFO and the received message is not yet complete.
- RME (Receive Message End) interrupt, indicating that the reception of one message is completed, i.e. either

  * one message with less than 32 bytes, or the
  * last part of a message with more than 32 bytes

  is stored in the RFIFO.

In addition to the message end (RME) interrupt the following information about the received frame is stored by the ESCC8 in special registers and/or RFIFO:

Table 8
Status Information after RME Interrupt

Length of message (bytes)	⇒ RBCH, RBCL	register
Address combination and/or	⇒ RSTA	RFIFO: last byte
Address field	⇒ RAL1	RFIFO
Control field	⇒ RHCR	RFIFO
Type of frame (COMMAND / RESPONSE)	⇒ RSTA	RFIFO: last byte
CRC result (good / bad)	⇒ RSTA	RFIFO: last byte
Valid frame (yes / no)	⇒ RSTA	RFIFO: last byte
ABORT sequence recognized (yes / no)	⇒ RSTA	RFIFO: last byte
Data overflow	⇒ RSTA	RFIFO: last byte

ASYNC, BISYNC

- RPF (Receive Pool Full) interrupt, indicating that a specified number of bytes (refer to register RFC) can be read from RFIFO.
- TCD (Termination Character Detected) interrupt, indicating that reception has been terminated by reception of a specified character (refer to register TCR and bit RFC.TCDE).

Additionally, the CPU can have access to contents of RFIFO without having received an interrupt (and thereby causing TCD to occur) by issuing the RFIFO Read command (CMDR.RFRD).

In addition to every received character the assigned status information Parity bit (0/1), Parity Error (yes/no), Framing Error (yes/no, ASYNC only!) is optionally stored in RFIFO.

In addition to the end conditions (TCD interrupt or after RFRD command) the length of the last received data block is stored in register RBCL.

Note: **For all serial modes!** After the received data has been read from the RFIFO, this must be explicitly acknowledged by the CPU issuing a RMC (Receive Message Complete) command. The CPU has to handle the RPF interrupt before additional 32 bytes are received via the serial interface which would cause a “Receive Data Overflow” condition.

The following figure gives an example of an interrupt controlled reception sequence, assuming that a “long” frame (66 bytes) followed by two short frames (6 bytes each) is received.

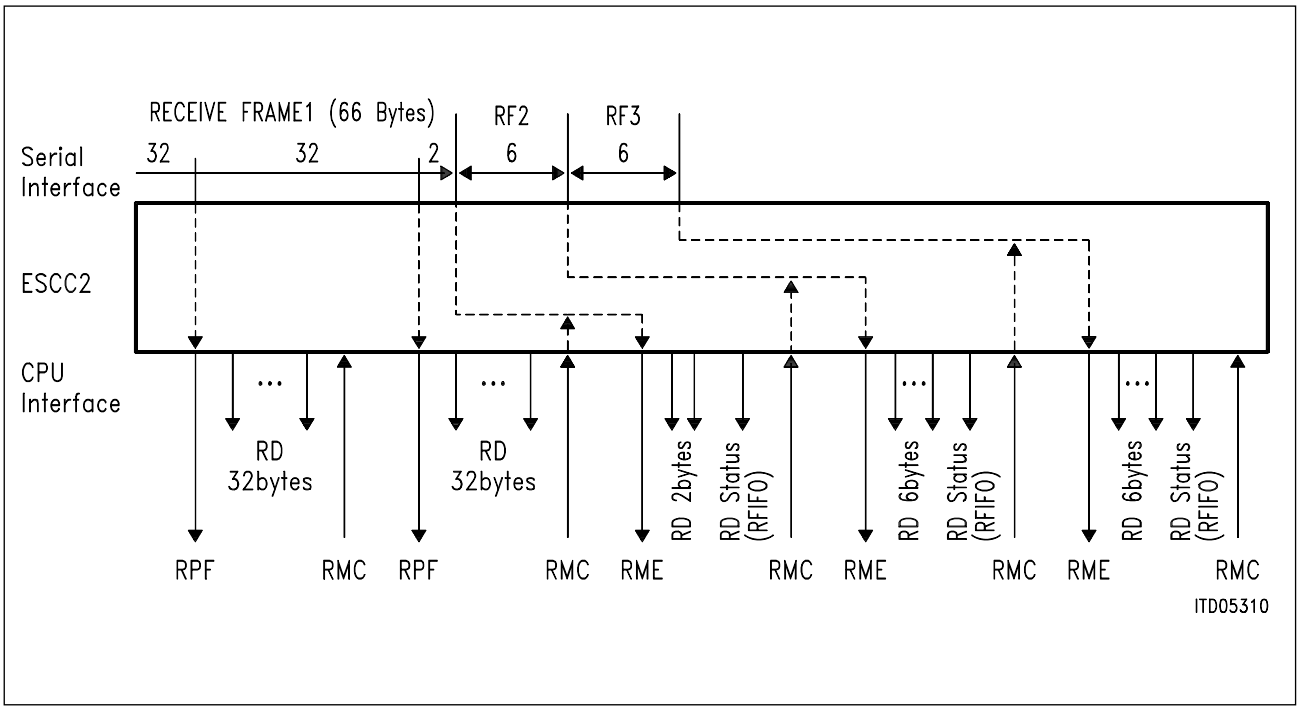


Figure 47
Interrupt Driven Reception Sequence Example (HDLC)

3.3.2.2 DMA Mode

If the RFIFO contains 32 bytes, the ESCC8 autonomously requests a block data transfer by DMA by activating the DRR line for as long as the start of the 32nd (byte access) or 16th (word access) read cycle. This forces the DMA controller to continuously perform bus cycles till 32 bytes are transferred from the ESCC8 to the system memory. If the RFIFO contains less than 32 bytes, the ESCC8 requests the correct amount of transfer cycles depending on the contents of the RFIFO and taking into account the selected bus width.

Note: All available status information for each frame/data block after the end conditions (RME or TCD) and for each character is the same as described above.

After the DMA controller has been set up for the reception of the next frame, the CPU must issue a RMC command to acknowledge the completion of received data processing. The ESCC8 will not initiate further DMA cycles by activating the DRR line prior to the reception of RMC.

In HDLC/SDLC mode the RECEIVE STATUS REGISTER is automatically read from the RFIFO with the last DMA-READ cycle of the received frame.

The status information after a RME interrupt is the same as in the interrupt driven mode.

The following figure gives an example of a DMA controlled reception sequence, supposing that a “long” frame (66 bytes) followed by two short frames (6 bytes each) is received.

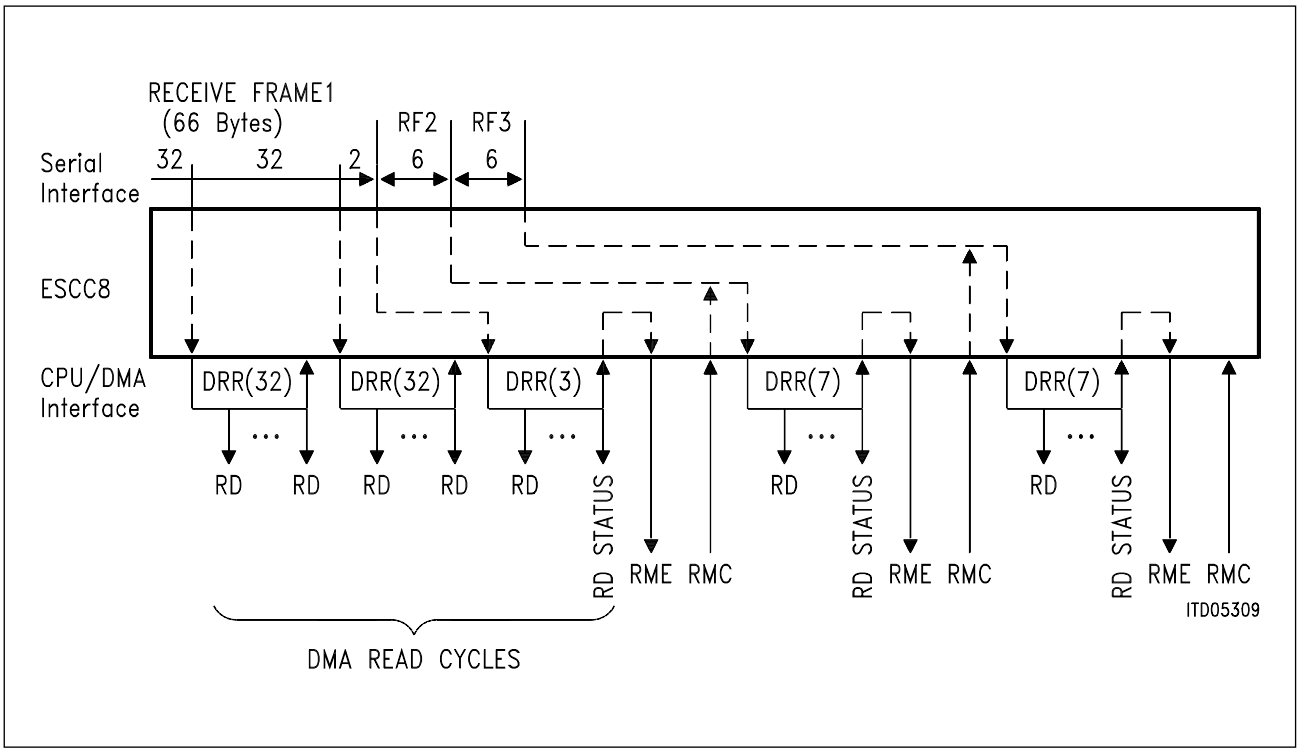


Figure 48
DMA Driven Reception Sequence Example (HDLC)

4 Detailed Register Description

In the register description the register addresses are specified by an “offset” relative to the “base addresses”, which are 000, 040, 080, 0C0, 100, 140, 180, 1C0 for the eight channels, respectively.

4.1 Status/Control Registers in HDLC Mode

4.1.1 Register Addresses

Table 9
Register Addresses in HDLC Mode

Address (A8 … A0)								Register	
Channel									
0	1	2	3	4	5	6	7	Read	Write
000 … 01F	040 … 05F	080 … 9F	0C0 … 0DF	100 … 11F	140 … 15F	180 … 19F	1C0 … 1DF	RFIFO	XFIFO
020	060	0A0	0E0	120	160	1A0	1E0	STAR	CMDR
021	061	0A1	0E1	121	161	1A1	1E1	RSTA	PRE
022	062	0A2	0E2	122	162	1A2	1E2	MODE	
023	063	0A3	0E3	123	163	1A3	1E3	TIMR	
024	064	0A4	0E4	124	164	1A4	1E4	XAD1	
025	065	0A5	0E5	125	165	1A5	1E5	XAD2	
026	066	0A6	0E6	126	166	1A6	1E6	———	RAH1
027	067	0A7	0E7	127	167	1A7	1E7	———	RAH2
028	068	0A8	0E8	128	168	1A8	1E8	RAL1	
029	069	0A9	0E9	129	169	1A9	1E9	RHCR	RAL2
02A	06A	0AA	0EA	12A	16A	1AA	1EA	RBCL	XBCL
02B	06B	0AB	0EB	12B	16B	1AB	1EB	RBCH	XBCH
02C	06C	0AC	0EC	12C	16C	1AC	1EC	CCR0	
02D	06D	0AD	0ED	12D	16D	1AD	1ED	CCR1	
02E	06E	0AE	0EE	12E	16E	1AE	1EE	CCR2	
02F	06F	0AF	0EF	12F	16F	1AF	1EF	CCR3	
030	070	0B0	0F0	130	170	1B0	1F0	———	TSAX
031	071	0B1	0F1	131	171	1B1	1F1	———	TSAR
032	072	0B2	0F2	132	172	1B2	1F2	———	XCCR
033	073	0B3	0F3	133	173	1B3	1F3	———	RCCR
034	074	0B4	0F4	134	174	1B4	1F4	VSTR	BGR
035	075	0B5	0F5	135	175	1B5	1F5	———	RLCR

Table 9
Register Addresses in HDLC Mode (cont'd)

Address (A8… A0)								Register	
Channel									
0	1	2	3	4	5	6	7	Read	Write
036	076	0B6	0F6	136	176	1B6	1F6	———	AML
037	077	0B7	0F7	137	177	1B7	1F7	———	AMH
038, 078, 0B8, 0F8, 138, 178, 1B8, 1F8								GIS *)	IVA *)
039, 079, 0B9, 0F9, 139, 179, 1B9, 1F9								IPC *)	
03A	07A	0BA	0FA	13A	17A	1BA	1FA	ISR0	IMR0
03B	07B	0BB	0FB	13B	17B	1BB	1FB	ISR1	IMR1
03C, 07C		0BC, 0FC		13C, 17C		1BC, 1FC		PVRA…D	
03D, 07D		0BD, 0FD		13D, 17D		1BD, 1FD		PISA..D	PIMA..D
03E, 07E		0BE, 0FE		13E, 17E		1BE, 1FE		PCRA…D	
03F	07F	0BF	0FF	13F	17F	1BF	1FF	CCR4	

*) All channel assigned addresses enable access to the same register(s)

Note: Read access to unused register addresses: value should be ignored,
Write access to unused register addresses: should be avoided, or set to '00'hex.

4.1.2 Register Definitions

Receive FIFO (READ) RFIFO (offset: 00…1F)

Reading data from the RFIFO can be done in 8-bit (byte) or 16-bit (word) access depending on the selected bus interface mode. The LSB is received first from the serial interface.

In Versions 2 and upwards, the size of the accessible part of RFIFO is determined by programming the bits CCR4.RFT 1 … 0 (RFIFO threshold level). It can be reduced from 32 bytes (RESET value) down to 2 bytes (four values: 32, 16, 4, 2 bytes).

- Interrupt Controlled Data Transfer (Interrupt Mode)
  Selected if DMA bit in XBCH is reset.

  Up to 32 bytes/16 words of received data can be read from the RFIFO following an RPF
  or an RME interrupt.
  RPF Interrupt: A fixed number of bytes/words to be read (version 1:32 bytes; version 2 upward: 32,16,4,2 bytes). The message is not yet complete.
  RME Interrupt: The message is completely received. The Number of valid **bytes** is determined by reading the RBCL, RBCH registers.

  RFIFO is released by issuing the "Receive Message Complete" command (RMC).

- DMA Controlled Data Transfer (DMA Mode)
  Selected if DMA bit in XBCH is set.

  If the RFIFO is filled up to its threshold level, the ESCC8 autonomously requests a block data transfer by DMA by activating the DRRn line until all read cycles are performed (the DRRn line remains active up to the beginning of the last read cycle). This forces the DMA controller to continuously perform bus cycles till all bytes/words are transferred from the ESCC8 to the system memory (level triggered transfer mode of DMA controller).

  If the RFIFO contains less bytes/words than defined via threshold level (one short frame or the last part of a long frame) the ESCC8 requests a block data transfer of size equal to the amount of data to be transferred.

Additionally, an RME interrupt is generated after the last byte has been transferred.

Further receiver DMA requests are blocked until an RMC command is issued in response to RME.

The valid byte count of the whole frame can be determined by reading the RBCH, RBCL registers following the RME interrupt.

Note: Addresses within the address space of the FIFO point all to the current data word/ byte, i.e. the current data byte can be accessed with any address within the 32-byte range.

Transmit FIFO (WRITE) XFIFO (offset: 00… 1F)

Writing data to the XFIFO can be done in 8-bit (byte) or 16-bit (word) access depending on the selected bus interface mode. The LSB is transmitted first.

- Interrupt Mode
  Selected if DMA bit in XBCH is set to zero.

  Up to 32 bytes/16 words of transmit data can be written to the XFIFO following an XPR (or ALLS) interrupt.

- DMA Mode
  Selected if DMA bit in XBCH is set to one.
  Prior to any data transfer, the actual **byte** count of the frame to be transmitted must be written to the XBCH, XBCL registers by the user.

  If data transfer is then initiated via the CMDR register (command XTF or XIF), the ESCC8 autonomously requests the correct amount of block data transfers (n*BW + Remainder; BW = 32 or 16; n = 0, 1,…).

Note: Addresses within the address space of the FIFO's all point to the current data word/byte, i.e. the current data byte can be accessed with any address within the 32-byte range.

Status Register (READ)

Value after RESET: 48_H

	7							0	
STAR	XDOV	XFW	XRNR	RRNR	RLI	CEC	CTS	WFA	(offset: 20)

XDOV… Transmit Data Overflow

More than 32 bytes have been written to the XFIFO.
This bit is reset by:
– a transmitter reset command XRES
– or when all bytes in the accessible half of the XFIFO have been moved into the inaccessible half.

XFW… Transmit FIFO Write Enable

Data can be written to the XFIFO.

XRNR… Transmit RNR (significant in auto-mode only!)

Indicates the status of the ESCC8.
0… receiver ready
1… receiver not ready

RRNR… Received RNR (significant in auto-mode only!)

Indicates the status of the remote station.
0… receiver ready
1… receiver not ready

RLI… Receive Line Inactive

Neither FLAGs as Interframe Time Fill nor frames are received via the receive line.

Note: Significant only in point-to-point configurations.

CEC… Command Executing

0…no command is currently being executed, the CMDR register can be written to.
1… a command (written previously to CMDR) is currently being executed, no further command can be temporarily written in CMDR register.

Note: CEC will be active at most 2.5 transmit clock (or master clock) periods. If the ESCC8 is in power down mode CEC will stay active.

CTS… Clear To Send State

This bit indicates the state of the $\overline{CTS}$ pin.
0… $\overline{CTS}$ is inactive (high)
1… $\overline{CTS}$ is active (low)

WFA… Wait For Acknowledgment (significant in auto-mode only).

Indicates the “Wait for I frame Acknowledgment” status of ESCC8.

Command Register (WRITE)

Value after RESET: 00_H

	7							0	
CMDR	RMC	RHR	RNR/ XREP	STI	XTF	XIF	XME	XRES	(offset: 20)

Note: The maximum time between writing to the CMDR register and the execution of the command is 2.5 clock cycles. Therefore, if the CPU operates with a very high clock rate in comparison with the ESCC8's clock, it is recommended that the CEC bit of the STAR register be checked before writing to the CMDR register to avoid any loss of commands.

RMC… Receive Message Complete

Confirmation from CPU to ESCC8 that the current frame or data block has been fetched following an RPF or RME interrupt, thus the occupied space in the RFIFO can be released.

Note: In DMA Mode, this command has to be issued after an RME interrupt, to enable the generation of further receiver DMA requests.

RHR… Reset HDLC Receiver

All data in the RFIFO and the HDLC receiver is deleted. In auto-mode, additionally the transmit and receive sequence number counters are reset.

RNR/XREP… Receiver Not Ready / Transmission Repeat

The function of this command depends on the selected operation mode (MDS1, MDS0, ADM bit in MODE):

- auto mode: RNR

Determines the status of the ESCC8 receiver, i.e. whether a received frame is acknowledged via an RR or RNR supervisory frame in auto-mode.
0… Receiver Ready (RR)
1… Receiver Not Ready (RNR)

- extended transparent mode 0, 1: XREP

If XREP is set to one together with XTF and XME (write 2AH to CMDR), the ESCC8 repeatedly transmits the contents of the XFIFO (1… 32 bytes) without HDLC framing fully transparently, i.e. without FLAG, CRC or Bit Stuffing.

The cyclic transmission is stopped with an XRES command.

STI… Start Timer

The internal timer is started.

Note: The timer is stopped by rewriting the TIMR register after start.

XTF… Transmit Transparent Frame

- Interrupt Mode

After having written up to 32 bytes to the XFIFO, this command initiates the transmission of a transparent frame. An opening flag sequence is automatically added to the data by the ESCC8.

- DMA Mode

After having written the length of the frame to be transmitted to the XBCH, XBCL registers, this command initiates the data transfer from system memory to ESCC8 by DMA. Serial data transmission starts as soon as 32 bytes are stored in the XFIFO or the Transmit Byte Counter value is reached.

XIF… Transmit I-Frame (used in auto-mode only!)

Initiates the transmission of an I-frame in auto-mode. Additionally to the opening flag sequence, the address and control field of the frame is automatically added by ESCC8.

XME… Transmit Message End (used in interrupt mode only!)

Indicates that the data block written last to the transmit FIFO completes the current frame. The ESCC8 can terminate the transmission operation properly by appending the CRC and the closing flag sequence to the data.

In DMA Mode, the end of the frame is determined by the Transmit Byte Count in XBCH, XBCL, thus, XME is not used in this case.

XRES… Transmitter Reset

XFIFO is cleared of any data and an abort sequence (seven 1's) followed by interframe time fill is transmitted. In response to XRES an XPR interrupt is generated.
This command can be used by the CPU to abort a frame currently in transmission.

Preamble Register (WRITE)

Value after RESET: 00_H

	7							0	
PRE	PR7							PR0	(offset: 21)

This register defines the pattern which is sent out during preamble transmission (refer to register CCR3).

Note: It should be taken into consideration that Zero Bit Insertion is disabled during preamble transmission.

Receive Status Register (READ)

	7							0	
RSTA	VFR	RDO	CRC	RAB	HA1	HA0	C/R	LA	(offset: 21)

Note: RSTA relates to the last received HDLC frame; it is copied into RFIFO when end-of-frame is recognized (last byte of each stored frame).

VFR… Valid Frame

Determines whether a valid frame has been received.
1… valid
0… invalid
An invalid frame is either
– a frame which is not an integer number of 8 bits (n × 8 bits) in length (e.g. 25 bits), or
– a frame which is too short taking into account the operation mode selected via MODE (MDS1, MDS0, ADM) and the selected CRC algorithm (CCR2.C32) and the selection of receive CRC ON/OFF (CCR3.RCRC) as follows:
- Auto-/Non-Auto-Mode (16 bit Address), RCRC = 0 : 4 bytes (CRC-CCITT) or 6 (CRC-32)
- Auto-/Non-Auto-Mode (16 bit Address), RCRC = 1 : 3-4 bytes (CRC-CCITT) or 3-6 (CRC-32)
- Auto-/Non-Auto-Mode (8 bit Address), RCRC = 0 : 3 bytes (CRC-CCITT) or 5 (CRC-32)
- Auto-/Non-Auto-Mode (8 bit Address), RCRC = 1 : 2-3 bytes (CRC-CCITT) or 2-5 (CRC-32)
- Transparent Mode 1: 3 bytes (CRC-CCITT) or 5 (CRC-32)
- Transparent Mode 0: 2 bytes (CRC-CCITT) or 4 (CRC-32)

Note: Shorter frames are not reported.

RDO… Receive Data Overflow

A data overflow has occurred during reception of the frame.

Additionally, an interrupt can be generated (refer to ISR1.RDO/ IMR1.RDO).

CRC… CRC Compare/Check

0… CRC check failed; received frame contains errors.
1… CRC check o.k.; received frame is error-free.

RAB… Receive Message Aborted

The received frame was aborted from the transmitting station. According to the HDLC protocol, this frame must be discarded by the receiver station.

HA1, HA0… High Byte Address Compare

Significant only if 2-byte address mode has been selected.

In operating modes which provide high byte address recognition, the ESCC8 compares the high byte of a 2-byte address with the contents of two individually programmable registers (RAH1, RAH2) and the fixed values FE_H and FC_H (broadcast address).

Dependent on the result of this comparison, the following bit combinations are possible:

10… RAH1 has been recognized
00… RAH2 has been recognized
01… broadcast address has been recognized

Note: If RAH1, RAH2 contain identical values, a match is indicated by '10'.

C/R… Command/Response

Significant only if 2-byte address mode has been selected.

Value of the C/R bit (bit in high address byte) in the received frame. The interpretation depends on the setting of the CRI bit in the RAH1 register. Refer also to the description of RAH1 register.

LA… Low Byte Address Compare

Not significant in transparent and extended transparent operating modes. The low byte address of a 2-byte address field, or the single address byte of a 1-byte address field is compared with two registers. (RAL1, RAL2).

0… RAL2 has been recognized
1… RAL1 has been recognized

According to the X.25 LAPB protocol, RAL1 is interpreted as the address of a COMMAND frame and RAL2 is interpreted as the address of a RESPONSE frame.

Mode Register (READ/WRITE)

Value after RESET: 00_H

	7							0	
MODE	MDS1	MDS0	ADM	TMD	RAC	RTS	TRS	TLP	(offset: 22)

MDS1… MDS0… Mode Select

The operating mode of the HDLC controller is selected.

00… auto-mode
01… non auto-mode
10… transparent mode
11…extended transparent mode

ADM… Address Mode

The meaning of this bit varies depending on the selected operating mode:

- auto-mode, non auto-mode
  Defines the length of the HDLC address field.
  0… 8-bit address field
  1… 16-bit address field

In transparent modes, this bit differentiates between two sub-modes:

- transparent mode
  0… transparent mode 0; no address recognition.
  1… transparent mode 1; high byte address recognition.
- extended transparent mode; without HDLC framing.
  0… extended transparent mode 0
  1… extended transparent mode 1

Note: In extended transparent modes, the RAC bit must be reset to enable fully transparent reception.

TMD… **Timer Mode**

Determines the operating mode of the timer.
0… external mode
The timer is controlled by the CPU and can be started at any time by setting the STI bit in CMDR.
1… internal mode
The timer is used internally by the ESCC8 for time-out and retry conditions in auto-mode (refer to the description of the TIMR register).

RAC… **Receiver Active**

Switches the receiver to operational or inoperational state.
0… receiver inactive
1… receiver active
In extended transparent modes this bit must be reset to enable fully transparent reception.

RTS… **Request To Send**

Defines the state and control of $\overline{\text{RTS}}$ pin.
0… The $\overline{\text{RTS}}$ pin is controlled by the ESCC8 autonomously.
$\overline{\text{RTS}}$ is activated when a frame transmission starts and deactivated when transmission is completed.
1… The $\overline{\text{RTS}}$ pin is controlled by the CPU.
If this bit is set, the $\overline{\text{RTS}}$ pin is activated immediately and remains active till this bit is reset.

TRS… **Timer Resolution**

Selects the resolution of the internal timer (factor k, see description of TIMR register):
0… k = 32 768
1… k = 512

TLP… **Test Loop**

Input and output of the HDLC channel are internally connected.
(e.g. transmitter channel 0 - receiver channel 0)

Timer Register (READ/WRITE)

	7		0	
TIMR	CNT	VALUE		(offset: 23)

VALUE… **(5 bits) Sets the time period t_1 as follows:**

$t_1 = k \times (\text{VALUE} + 1) \times \text{TCP}$

where

– k is the timer resolution factor which is either 32 768 or 512 clock cycles dependent on the programming of TRS bit in MODE.

– TCP is the clock period of transmit data.

CNT… **(3 bits) Interpreted differently depending on the selected timer mode (bit TMD in MODE).**

• Internal timer mode (MODE.TMD = 1)

– Retry Counter (in HDLC known as N2)

CNT indicates the number of S-commands (max. 6) which are transmitted autonomously by the ESCC8 after expiration of time period t_1, in case an I-frame is not acknowledged by the opposite station.

If CNT is set to 7, the number of S-commands is unlimited.

• External timer mode (MODE.TMD = 0)

CNT plus VALUE determine the time period t_2 after which a timer interrupt will be generated. The time period t_2 is

$t_2 = 32 \times k \times \text{CNT} \times \text{TCP} + t_1$.

If CNT is set to 7, a timer interrupt is periodically generated after the expiration of t_1.

Transmit Address Byte 1 (READ/WRITE)

	7			0	
2-byte address					
XAD1	XAD1 (high byte)		0	(0)	(offset: 24)
1-byte address					
XAD1	XAD1 (COMMAND)				

XAD1 (and XAD2) can be programmed with one individual address byte which is appended automatically to the frame by ESCC8 in auto-mode. The function depends on the selected address mode (bit ADM in MODE).

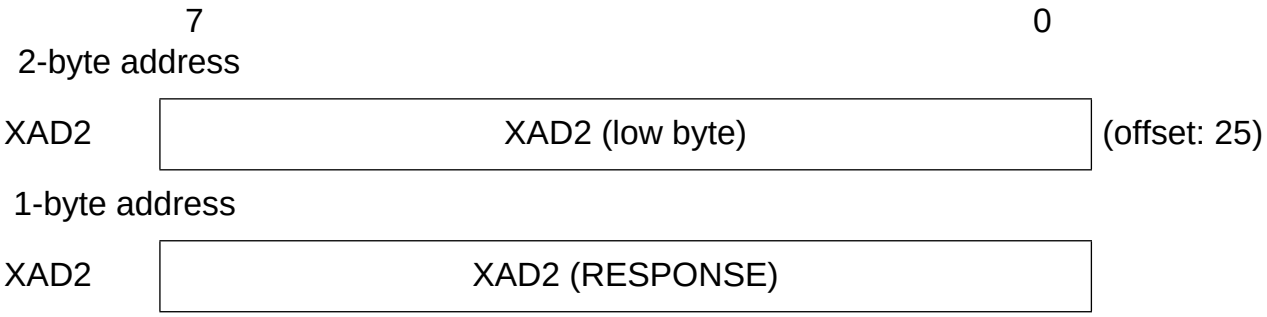
– 2-byte address field (MODE.ADM = 1)
  XAD1 constitutes the high byte of the 2-byte address field. Bit 1 must be set to 0. According to the ISDN LAPD protocol, bit 1 is interpreted as the C/R (COMMAND/ RESPONSE) bit. This bit is manipulated automatically by the ESCC8 according to the setting of the CRI bit in RAH1:

	Bit 1	**(C/R)**
Commands transmit	1	0
Response transmit	0	1
	CRI = 1	CRI = 0

  (In ISDN LAP-D, the high address byte is known as SAPI).
  In accordance with the HDLC protocol, bit 0 should be set to 0, to indicate that the address field contains (at least) one more byte.

– 1-byte address field (MODE.ADM = 0)
  According to the X.25 LAPB protocol, XAD1 is the address of a COMMAND frame.

Transmit Address Byte 2 (READ/WRITE)



Second individually programmable address byte.

– 2-byte address (MODE.ADM = 1)
  XAD2 constitutes the low byte of the 2 byte address field
  (In ISDN LAP-D, the low address byte is known as TEI).

– 1-byte address (MODE.ADM = 0)
  According to the X.25 LAPB protocol, XAD2 is the address of a RESPONSE frame.

Note: XAD1, XAD2 registers are used only if the ESCC8 is operated in auto-mode.

Receive Address Byte High Register 1 (WRITE)

	7			0	
RAH1	RAH1		CRI	0	(offset: 26)

In operating modes that provide high byte address recognition, the high byte of the received address is compared with the individually programmable values in RAH1 and RAH2.

In versions 2 and upwards, this register can be masked by setting the corresponding bits in the mask register AMH to allow extended broadcast address recognition. This feature is applicable to all operating modes with address recognition.

RAH1… **Value of the first individual high address byte**

CRI… **Command/Response Interpretation**

The setting of the CRI bit affects the meaning of the C/R bit in RSTA as follows:

C/R Meaning	C/R	Value
Commands received	0	1
Responses received	1	0
	CRI = 1	CRI = 0

Important Note: If 1-byte address field is selected in auto-mode, RAH1 must be set to 00_H.

Receive Address Byte High Register 2 (WRITE)

	7			0	
RAH2	RAH2		MCS	0	(offset: 27)

RAH2… **Value of second individual high address byte.**

MCS… **Modulo Count Select (valid in auto-mode only!)**

The MCS bit determines the HDLC control field format.
0… basic operation, one-byte control field (modulo 8)
1… extended operation, two-byte control field (modulo 128)

Note: When modulo 128 is selected, in auto mode the “RHCR'” register contains compressed information of the extended control field (see RHCR register description). RAH1, RAH2 registers are used in auto- and non-auto operating modes when a 2-byte address field has been selected (MODE.ADM = 1) and in transparent mode 0.

Receive Address Byte Low Register 1 (WRITE or READ)

	7		0	
RAL1	RAL1			(offset: 28)

The general function (WRITE or READ) and the meaning or contents of this register depend on the selected operating mode:

- **Auto-/Non-Auto-Mode (16-bit Address) - WRITE Access only:**
  (Read access not specified)

  RAL1 can be programmed with the value of the first individual low address byte.
- **Auto-/Non-Auto-Mode (8-bit Address) - WRITE Access only:**
  (Read access not specified)

  According to X.25 LAPB protocol, the address in RAL1 is considered as the address of a COMMAND frame.
- **Transparent Mode 1 (High Byte Address Recognition) - READ Access only:**
  (Write access has no influence)

  RAL1 contains the byte following the high byte of the address in the receive frame (i.e. the second byte after the opening flag).
- **Transparent Mode 0 (No Address Recognition) - READ Access only:**
  (Write access has no influence)

  RAL1 contains the first byte after the opening flag (first byte of received frame).

- **Extended Transparent Modes 0, 1 - READ Access only:**
  (Write access has no influence)

  RAL1 contains the current data byte assembled from the R × D pin, the HDLC receiver is by-passed (fully transparent reception without HDLC framing).

In versions 2 upward, this register can be masked by setting the corresponding bits in the mask register AML to allow extended broadcast address recognition. This feature is applicable to all operating modes with address recognition.

Receive HDLC Control Register (READ)

Value of the HDLC control field of the last received frame.

Note: RHCR is copied into RFIFO for every frame.

	Contents of RHCR	
Mode	**Modulo 8 (MCS = 0)**	**Modulo 128 (MCS = 1)**
Auto mode, 1-byte address (U-frames) (Note 1)	Control field	Control field in (Note 2)
Auto mode, 2-byte address (U-frames) (Note 1)	Control field	Control field in (Note 2)
Auto mode, 1-byte address (I-frames) (Note 1)	Control field	Control field in compressed form (Note 3)
Auto mode, 2-byte address (I-frames) (Note 1)	Control field	Control field in compressed form (Note 3)
Non-auto mode, 1-byte address	2nd byte after flag	
Non-auto mode, 2-byte address	3rd byte after flag	
Transparent mode 1	3rd byte after flag	
Transparent mode 2	2nd byte after flag	

Note 1: S-frames are handled automatically and are not transferred to the microprocessor.

Note 2: For U-frames (bit 0 of RHCR = 1) the control field is as in the modulo 8 case.

Note 3: For I-frames (bit 0 of RHCR = 0) the compressed control field has the same format as in the modulo 8 case, but only the three LBS's of the receive and transmit counters are visible:

bit	7	6	5	4	3	2	1	0
	N(R)			P	N(S)			0

Receive Address Byte Low Register 2 (WRITE)

	7							0	
RAL2	RAL2								(offset: 29)

Value of the second individually programmable low address byte. If a one byte address field is selected, RAL2 is considered as the address of a RESPONSE frame according to X.25 LAPB protocol.

Receive Byte Count Low (READ)

	7							0	
RBCL	RBC7							RBC0	(offset: 2A)

Together with RBCH (bits RBC11 - RBC8), indicates the length of a received frame (1…4096 bytes). Bits RBC4-0 indicate the number of valid bytes currently in RFIFO. These registers must be read by the CPU following a RME interrupt.

Transmit Byte Count Low (WRITE)

	7						0	
XBCL	XBC7						XBC0	(offset: 2A)

Together with XBCH (bits XBC11…XBC8) this register is used in DMA Mode only, to program the length (1…4096 bytes) of the next frame to be transmitted. In terms of the value xbc, programmed in XBC11…XBC0 (xbc = 0…4095), the length of the block in number of bytes is:

length = xbc + 1.

This allows the ESCC8 to request the correct amount of DMA cycles after an XTF or XIF command in CMDR.

Received Byte Count High (READ)

Value after RESET: 000_{xxxxx}

	7						0	
RBCH	DMA	NRM	CAS	OV	RBC11		RBC8	(offset: 2B)
	see XBCH							

DMA, NRM, CAS…
These bits represent the read-back value programmed in XBCH

OV… Counter Overflow

More than 4095 bytes received.

RBC11 – RBC8… Receive Byte Count (most significant bits)

Together with RBCL (bits RBC7… RBC0) indicate the length of the received frame.

Transmit Byte Count High (WRITE)

Value after RESET: 000_{xxxxx}

	7					0	
XBCH	DMA	NRM	CAS	XC	XBC11	XBC8	(offset: 2B)

DMA… DMA Mode

Selects the data transfer mode of ESCC8 to/from System Memory.

0… Interrupt controlled data transfer (Interrupt Mode).
1… DMA controlled data transfer (DMA Mode).

NRM… Normal Response Mode

Valid in auto-mode only.

Determines the function of the LAP Controller:
0… full-duplex LAPB/LAPD operation
1… half-duplex NRM operation

CAS… Carrier Detect Auto Start

When set, a high on the CD pin enables the corresponding receiver and data reception is started. When not set, if not in Clock Mode 1 or 5, the CD pin can be used as a general input.

XC… Transmit Continuously

Only valid if DMA Mode is selected.

If the XC bit is set, the ESCC8 continuously requests for transmit data ignoring the transmit byte count programmed via XBCH, XBCL.

XBC11 – XBC8… Transmit Byte Count (most significant bits)

Valid only if DMA Mode is selected.

Together with XBCL (bits XBC7… XBC0), determine the length of the frame to be transmitted.

Channel Configuration Register 0 (READ/WRITE)

Value after RESET: 00_H

	7							0	
CCR0	PU	MCE	0	SC2	SC1	SC0	SM1	SM0	(offset: 2C)

Note: Unused bits have to be set to logical “0”.

PU… Switches between power up and power down mode

0… power down (standby)
1… power up (active)

MCE… Master Clock Enable

If this bit is set to “1”, the clock provided via pin XTAL1 works as master clock to allow full functionality of the microprocessor interface (access to all status and control registers and FIFOs, DMA and interrupt support) independent of the receive and the transmit clocks. The internal oscillator in conjunction with a crystal on XTAL1-2 can be used, too. The master clock option is not applicable in clock mode 5 or in SDLC Loop mode. **Refer to table 5** for more details.

Note: The internal timers run with the master clock.

SC2 – SC0… Serial Port Configuration

000… NRZ data encoding
001… bus configuration, timing mode 1
010… NRZI data encoding
011… bus configuration, timing mode 2
100… FM0 data encoding
101… FM1 data encoding
110… MANCHESTER data encoding
111… (not used)

Note: If bus configuration is selected, only NRZ coding is supported.

SM1 – SM0… Serial Mode

00… HDLC/SDLC mode
01… SDLC Loop mode
10… BISYNC mode
11… ASYNC mode

Channel Configuration Register 1 (READ/WRITE)

Value after RESET: 00_H

	7						0	
CCR1	SFLG	GALP	GLP	ODS	ITF/ OIN	CM2	CM0	(offset: 2D)

SFLG… Enable Shared Flags

If this bit is set, the closing FLAG of a preceding frame simultaneously becomes the opening FLAG of the following frame.

GALP… Go Active On Loop

Only used if SDLC Loop is enabled.

This bit enables transmission on an SDLC Loop.

1… After detection of the next EOP sequence, the ESCC8 goes to the Sending On Loop state by changing the seventh 1-bit of the EOP sequence into a 0, thus creating a Start Flag, and by disconnecting the T × D pin from the R × D pin. The ESCC8 is now active on loop and can transmit frames as soon as data is available in the XFIFO. The time between frames is always filled by sending continuous Flags (independent from the value of bit CCR1.ITF), thus occupying the loop.

0… The ESCC8 leaves the Sending On Loop state when the XFIFO is empty by retransmitting data received on R × D to T × D (with one bit delay) after the closing flag has been transmitted (thus creating an EOP sequence).

GLP… Go On Loop

Only used if SDLC Loop is enabled.

This command controls entering and leaving the SDLC Loop.

1… The ESCC8 enters the On Loop state after detection of the next EOP sequence by adding a 1-bit delay between receive and transmit path. The On Loop state is prerequisite for sending frames on loop.

0… The ESCC8 leaves the On Loop state by suppressing the 1-bit delay after detection of the next EOP sequence.

ODS… Output Driver Select

Defines the function of the transmit data pin (T × D)

0… T × D pin is an open drain output.
1… T × D pin is a push-pull output.

Note: This feature is also valid for pin R × D if it is switched to T × D function via bit CCR2.SOC1.

ITF/OIN… Interframe Time Fill / One Insertion

The function of this bit depends on the selected Serial Port Configuration (bit SC1):

- Point-to-point configurations: ITF
  Determines the idle (= no data to send) state of the transmit data pin T × D
  0… Continuous logical “1” is output
  1… Continuous FLAG sequences are output (“01111110” bit patterns)
- Bus configurations: OIN
  When this bit is set, a “ONE” insertion (deletion) mechanism is activated: a “1” is inserted after seven consecutive “0”s in the transmit data stream and a “1” is deleted after seven consecutive “0” in the receive data stream. Similar to the HDLC bit-stuffing mechanism (inserting a “0” after five consecutive “1”s), this enables clock information to be recovered from the receive data stream by means of a DPLL even in the case of NRZ encoding, because a transition at bit cell boundary occurs at least every 7 bits. The “One Insertion” cannot be used in conjunction with the master clock option.

Note: In bus configurations, the ITF is implicitly set to 0, i.e. continuous “1”s are transmitted, and data encoding is NRZ.

CM2 – CMO… Clock Mode

Selects one of 8 different clock modes:

000 clock mode 0
• •
• •
• •
111 clock mode 7

Channel Configuration Register 2 (READ/WRITE)

Value after RESET: 00_H
The meaning of the individual bits in CCR2 depends on the clock mode selected via CCR1 as follows:

CCR2

	7							0	
clock mode 0a, 1	SOC1	SOC0	0	SSEL	0	RWX	C32	DIV	(offset: 2E)
clock mode 0b,2,3,6,7	BR9	BR8	BDF	SSEL	TOE	RWX	C32	DIV	
clock mode 4	SOC1	SOC0	0	0	TOE	RWX	C32	DIV	
clock mode 5	SOC1	SOC0	XCS0	RCS0	TOE	RWX	C32	DIV	

Note: Unused bits have to be set to logical “0”.

SOC1, SOC0… Special Output Control

In a bus configuration (selected via CCR0), defines the function of pin $\overline{\text{RTS}}$ as follows:
0X… $\overline{\text{RTS}}$ output is activated during transmission of a frame.
10… $\overline{\text{RTS}}$ output is always high ($\overline{\text{RTS}}$ disabled).
11… $\overline{\text{RTS}}$ indicates the reception of a data frame (active low).
In a point-to-point configuration (selected via CCR0) the T × D and R × D pins may be flipped
0X… data is transmitted on T × D, received on R × D (normal case).
1X… data is transmitted on R × D, received on T × D.

BR9, BR8… Baud Rate, Bit 9-8

High order bits, see description of BGR register.

XCS0, RCS0… Transmit/Receive Clock Shift, Bit 0

Together with bits XCS2, XCS1 (RCS2, RCS1) in TSAX (TSAR) the clock shift relative to the frame synchronization signal of the Transmit (Receive) time-slot can be adjusted. A clock shift of 0… 7 bits is programmable.

BDF… Baud Rate Division Factor

0… The division factor of the baud rate generator is set to 1 (constant).
1… The division factor is determined by BR9 - BR0 bits in CCR2 and BRG registers.

SSEL… **Clock Source Select**

Selects the clock source in clock modes 0, 2, 3, 6 and 7 (**refer to table 5**).

TOE… **TxCLK Output Enable**

0… T × CLK pin is input.
1… T × CLK pin is switched to output function if applicable to the selected clock mode (**refer to table 5**).

RWX… **Read/Write Exchange**

Valid only in DMA mode. If this bit is set, the
– $\overline{RD}$ and $\overline{WR}$ pins are internally exchanged (Siemens/Intel bus interface)
– R/$\overline{W}$ pin is inverted in polarity (Motorola bus interface)
while any $\overline{DACK}$ input is active. This feature allows a simple interfacing to the DMA controller.

Note: The RWX bit of all eight channels is “or”ed.

C32… **Enable CRC-32**

0… CRC-CCITT is selected.
1… CRC-32 is selected.

DIV… **Data Inversion**

Only valid if NRZ data encoding is selected. Data is transmitted and received inverted.

Channel Configuration Register 3 (READ/WRITE)

Value after RESET: 00_H

	7							0	
CCR3	PRE1	PRE0	EPT	RADD	CRL	RCRC	XCRC	PSD	(offset: 2F)

PRE1, PRE0… Number of Preamble Repetition

If Preamble transmission is initiated, the Preamble defined via register PRE is transmitted
00… 1 times
01… 2 times
10… 4 times
11… 8 times.

EPT… **Enable Preamble Transmission**

This bit enables transmission of a preamble. The preamble is started after Interframe Timefill transmission has been stopped and a new frame is to be transmitted. The preamble consists of an 8-bit pattern repeated a number of times. The pattern is defined via register PRE, the number of repetitions is selected by bits PRE0 and PRE1.

Note: The “Shared Flag” feature is not influenced by preamble transmission. Zero Bit Insertion is disabled during preamble transmission.

RADD… **Receive Address pushed to RFIFO**

If this bit is set to “1”, the received HDLC address information (1 or 2 bytes, depending on the address mode selected via MODE.ADM) is pushed to RFIFO. This function is applicable in auto mode, non-auto mode and transparent mode 1.

CRL… **CRC Reset Level**

This bit defines the initialization for the internal receive and transmit CRC generators:

0… Initialized to (FFFF)FFFF$_H$. Default value for most HDLC/SDLC applications.

1… Initialized to (0000)0000$_H$.

RCRC… **Receive CRC ON/OFF**

Only applicable in non-auto mode and transparent mode 0.

If this bit is set to “1”, the received CRC checksum will be written to RFIFO (CRC-CCITT: 2 bytes; CRC-32: 4 bytes). The checksum, consisting of the 2 (or 4) last bytes in the received frame, is followed in the RFIFO by the status information byte (contents of register RSTA). The received CRC checksum will additionally be checked for correctness. If non-auto mode is selected, the limits for “Valid Frame” check are modified (**refer to RSTA.VFR**).

XCRC… **Transmit CRC ON/OFF**

If this bit is set to “1”, the CRC checksum will not be generated internally. It has to be written as the last two or four bytes in the transmit FIFO (XFIFO). The transmitted frame will be closed automatically with a closing flag.

Note: The ESCC8 does not check whether the length of the frame, i.e. the number of bytes to be transmitted makes sense or not.

PSD… **DPLL Phase Shift Disable**

Only applicable in the case of NRZ and NRZI encoding.

If this bit is set to “1”, the Phase Shift function of the DPLL is disabled. In this case the windows for Phase Adjustment are extended.

Time-Slot Assignment Register Transmit (WRITE)

This register is only used in clock mode 5!

Value after RESET: 00_H

	7			0	
TSAX	TSNX		XCS2	XCS1	(offset: 30)

TSNX… **Time-Slot Number Transmit**

Selects one of up 64 possible timeslots (00_H–$3F_H$) in which data is transmitted. The number of bits per timeslot can be programmed via XCCR.

XCS2, XCS1… Transmit Clock Shift, Bit 2-1

Together with bit XCS0 in CCR2, transmit clock shift can be adjusted.

Time-Slot Assignment Register Receive (WRITE)

This register is only used in clock mode 5!

Value after RESET: 00_H

	7			0	
TSAR	TSNR		RCS2	RCS1	(offset: 31)

TSNR… **Time-Slot Number Receive**

Defines one of up to 64 possible time-slots (00_H-$3F_H$) in which data is received. The number of bits per time-slot can be programmed via RCCR.

RCS2, RCS1… Receive Clock Shift, Bit 2-1

Together with bit RCS0 in CCR2, the receive clock shift can be adjusted.

Transmit Channel Capacity Register (WRITE)

This register is only used in clock mode 5!

Value after RESET: 00_H

	7						0	
XCCR	XBC7						XBC0	(offset: 32)

XBC7 – XBC0… Transmit Bit Number Count, Bit 7-0

Defines the number of bits to be transmitted within a time-slot:
Number of bits = XBC + 1 (1 … 256 bits/time-slot).

Receive Channel Capacity Register (WRITE)

This register is only used in clock mode 5!

Value after RESET: 00_H

	7						0	
RCCR	RBC7						RBC0	(offset: 33)

RBC7 – RBC0… Receive Bit Count, Bit 7-0

Defines the number of bits to be received within a time-slot:
Number of bits = RBC + 1 (1 … 256 bits/time-slot).

Version Status Register (READ)

	7						0	
VSTR	CD	DPLA	0	0	VN3		VN0	(offset: 34)

CD… Carrier Detect

This bit reflects the state of the CD pin.
1… CD active
0… CD inactive

DPLA… DPLL Asynchronous

This bit is only valid when the receive clock is supplied by the DPLL and FM0, FM1 or Manchester data encoding is selected.

It is set when the DPLL has lost synchronization. Reception is disabled (receiver aborted) until synchronization has been regained. Additionally, transmission is interrupted, too, if the transmit clock is derived from the DPLL (same effect as the deactivation of pin $\overline{\text{CTS}}$).

VN3 – VN0… Version Number of Chip

0… Version 1
1… Version 2

Baud Rate Generator Register (WRITE)

	7							0	
BGR	BR7							BR0	(offset: 34)

BR7 – BR0… Baud Rate, Bit 7-0

Together with bits BR9, BR8 of CCR2, determines the division factor of the baud rate generator.

In terms of the value N programmed in BR9 - BR0 (n = 0… 1023), the division factor k is:

$k = (n + 1) \times 2$

Receive Length Check Register (WRITE)

	7							0	
RLCR	RC	RL6						RL0	(offset: 35)

RC… Receive Check (on/off)

0… Receive Length Check feature disabled
1… Receive Length Check feature enabled

RL6 – RL0… Receive Length

The maximum receive length after which data reception is suspended can be programmed here. The receive length is (RL + 1) $\times$ 32 bytes, where RL is the value programmed via RL6-0.

A frame exceeding this length is treated as if it was aborted by the opposite station (RME Interrupt, RAB bit set).

In this case, the Receive Byte Count (RBCH, RBCL) is greater than the programmed Receive Length.

Address Mask Low (WRITE)

(Version 2 upwards)

Value after RESET: 00_H

	7						0	
AML	AML7						AML0	(offset: 36)

The Receive Address Low Byte (RAL1) can be masked by setting corresponding bits in this mask register to allow extended broadcast address recognition. This feature is applicable in all operating modes with address recognition. The function is disabled if all bits of this register are set to zero (RESET value).

Address Mask High (WRITE)

(Version 2 upwards)

Value after RESET: 00_H

	7						0	
AMH	AMH7						AMH0	(offset: 37)

The function is similar to AML but with respect to register RAH1.

Global Interrupt Status Register (READ)

Value after RESET: 00_H

	7							0	
GIS	PIA	PIB	PIC	PID	CII	CN2	CN1	CN0	(038/078/0B8/0F8) (138/178/1B8/1F8)

This status register points to pending

– channel assigned interrupts (ISR0_x, ISR1_x)

– universal port interrupts (PISA..D).

GIS is accessible via eight channel addresses (038_H to $1F8_H$).

PIA, PID… Port Interrupt Indication

These status bits point to pending interrupts in corresponding Port Interrupt Status registers PISA…PISD. They may be set independently from channel assigned interrupts.

CII… Channel Interrupt Indication

Set if at least one interrupt source of any channel is active.

CN2 – CN0… Channel Number (0..7)

If at least one interrupt source is active (bit CII is set), these bits point to the channel with currently highest source priority. **Refer to chapter 2.2.3** for detailed description of the priority structure.

Contents of register GIS are frozen after every input acknowledge cycle.

– after the first read access to GIS after the interrupt vector has been output,

– after every read access to anyone of the channel assigned interrupt status registers,

– during every $\overline{\text{INTA}}$ cycle.

Interrupt Vector Address (WRITE)

Value after RESET: 00_H

	7							0	
IVA	T7	T6	T5	T4	T3	T2	ROT	EDA	(038/078/0B8/0F8) (138/178/1B8/1F8)

Note: Unused bits have to be set to logical “0”.

IVA is accessible via eight channel addresses (38_H to $1F8_H$).

Version 2 upward provides dynamic adjustment of channel priorities by programming the “highest priority channel”. Selection of the “highest priority channel” is done with every **write access** to IVA in conjunction with the channel assigned IVA register address:

IVA Register Address:	Highest Priority Channel
38_H	0
78_H	1
$B8_H$	2
$F8_H$	3
138_H	4
178H	5
1B8H	6
1F8H	7

The priority level becomes valid with the end of the write access to the IVA register (rising edge of $\overline{\text{WR}}$ or $\overline{\text{DS}}$, whichever applies) and remains stable until a new write access to this register occurs.

T7 – T6… **Device Address**

These bits define the value of bits 6 and 7 of the interrupt vector which is sent out on the data bus (D0… D7) during the interrupt acknowledge cycle.

T5… **Device Address**

Version 1: Device Address
This bit defines the value of bit 5 of the interrupt vector which is sent out on the data bus (D0… D7) during the interrupt acknowledge cycle.

Version 2: Device Address Extension
In Interrupt vector mode 2 (bit EDA set) this bit defines the value of bit 5 of the interrupt vector which is sent out on the data bus (D0… D7) during the interrupt acknowledge cycle.

T4 – T2… **Device Address Extension**

In Interrupt vector mode 2 (bit EDA set) these bits define the value of bits 2 to 4 of the interrupt vector which is sent out on the data bus (D0… D7) during the interrupt acknowledge cycle.

ROT… **Rotating Interrupt Priority (version 2 upward)**

Version 1:
This bit is unused and has to be set to logical “0”.

Version 2:

0 … Fixed Interrupt Priority
The relative order of the interrupt priority level assigned to the channels is fixed (**refer to chapter 2.3.1**).

1 … Rotating Interrupt Priority
The interrupt priority level will be adjusted after an interrupt has been serviced. Together with bit IPC.ROTM the interrupt priority mode is selected.
IPC.ROTM = 0: The priority level of all 8 serial channels are adjusted.
IPC.ROTM = 1: The priority level of only 7 channels are adjusted while one channel is fixed.

EDA… **Extended Device Address**

If set, bits 2 to 5 (version 1: bits 2 to 4) of the generated interrupt vector contain the Device Address Extension T2..T5 (version 1: T2..T4) instead of the channel number. For detailed information **refer to chapter 2.2.3**.

Interrupt Port Configuration (READ/WRITE)

Value after RESET: 00_H

	7							0	
IPC	VIS	ROTM	SLA2	SLA1	SLA0	CASM	IC1	IC0	(039/079/0B9/0F9) (139/179/1B9/1F9)

Note: Unused bits have to be set to logical “0”.

IPC is accessible via eight channel addresses (039_H to $1F9_H$).

VIS… Masked Interrupts Visible (version 2 upward)

0… Masked interrupt status bits are not visible.

1… Masked interrupt status bits are visible.

ROTM… Rotating Interrupt Priority Mode (version 2 upward)

Together with bit IVA.ROT the interrupt priority mode is selected.

0… With IVA.ROT = 1 the priorities of all 8 serial channels are rotated cyclically after an interrupt has been serviced. The channel last serviced is assigned the lowest priority of all (**refer to chapter 2.2.3.1.**).

1… With IVA.ROT = 0 the priority adjustment is performed only on 7 channels while one channel is fixed to highest priority level (**refer to chapter 2.2.3.1**).

SLA2 – SLA0… Slave Address

Only used in Slave Cascading Mode (refer to CASM).

CASM… Cascading Mode

0… Slave Cascading Mode
Pins IE0, IE1 and IE2 are used as inputs. Interrupt acknowledge is accepted if an interrupt signal has been generated and the values on pins IE0, IE1 and IE2 correspond to the programmed values in SLA0, SLA1 and SLA2 (slave address).

1… Daisy Chaining Mode
Pin IE0 as Interrupt Enable Output and pin IE1 as Interrupt Enable Input are used for building a Daisy Chain. Pin IE2 is not used. Interrupt acknowledge is accepted if an interrupt signal has been generated and Interrupt Enable Input IE1 is active “high” during a subsequent $\overline{\text{INTA}}$ cycle(s). If pin INT goes active, Interrupt Enable Output IE0 is immediately set “low”.

IC1, IC0… Interrupt Port Configuration

These bits define the function of the interrupt output stage (pin INT):

IOC1	IOC0	Function
X	0	Open drain output
0	1	Push/pull output, active low
1	1	Push/pull output, active high

Interrupt Status Register 0 (READ)

Value after RESET: 00_H

	7							0	
ISR0	RME	RFS	RSC	PCE	PLLA	CDSC	RFO	RPF	(offset: 3A)

All bits are reset when ISR0 is read. Additionally, RME and RPF are reset when the corresponding interrupt vector is output.

Note: If bit IPC.VIS is set to “1”, interrupt statuses in ISR0 may be flagged although they are masked via register IMR0. However, these masked interrupt statuses neither generate an interrupt vector or a signal on INT, nor are visible in register GIS.

RME… Receive Message End

One complete message of length less than 32 bytes, or the last part of a frame at least 32 bytes long is stored in the receive FIFO, including the status byte.

The complete message length can be determined reading the RBCH, RBCL registers, the number of bytes currently stored in RFIFO is given by RBC4–0. Additional information is available in the RSTA register.

RFS… Receive Frame Start

This is an early receiver interrupt activated after the start of a valid frame has been detected, i.e. after an address match (in operation modes providing address recognition), or after the opening flag (transparent mode 0) is detected, delayed by two bytes. After an RFS interrupt, the contents of

- RHCR
- RAL1
- RSTA - bits 3-0

are valid and can be read by the CPU.

RSC… **Receive Status Change (significant in auto-mode only)**

A status change (receiver ready/receiver not ready) of the remote station has been detected by receiving a RR/RNR supervisory frame. The actual status can be read from the STAR register (RRNR bit).

PCE… **Protocol Error (significant in auto-mode only)**

The ESCC8 has detected a protocol error, i.e. it has received
– an S- or I-frame with incorrect N (R)
– an S-frame containing an I-field.

PLLA… **DPLL Asynchronous**

This bit is only valid when the receive clock is supplied by the DPLL and FM0, FM1 or Manchester data encoding is selected.

It is set when the DPLL has lost synchronization. Reception is disabled (receiver aborted) until synchronization has been regained. Additionally, transmission is also interrupted if the transmit clock is derived from the DPLL.

CDSC… **Carrier Detect Status Change**

Indicates that a state transition has occurred on CD. The actual state can be read from the VSTR register.

RFO… **Receive Frame Overflow**

At least one complete frame was lost because no storage space was available in the RFIFO. This interrupt can be used for statistical purposes and indicates that the CPU does not respond quickly enough to an RPF or RME interrupt.

RPF… **Receive Pool Full**

32 bytes of a frame have arrived in the receive FIFO. The frame is not yet completely received.

Note: This interrupt is only generated in Interrupt Mode.

Interrupt Status Register 1 (READ)

	7							0	
ISR1	EOP	OLP/ RDO	AOLP/ ALLS	XDU/ EXE	TIN	CSC	XMR	XPR	(offset: 3B)

All bits are reset when ISR1 is read. Additionally, XPR is reset when the corresponding interrupt vector is output.

Note: If bit IPC.VIS is set to “1”, interrupt statuses in ISR1 may be flagged although they are masked via register IMR1. However, these masked interrupt statuses neither generate an interrupt vector or a signal on INT, nor are visible in register GIS.

EOP... End of Poll Sequence Detected

Only valid if SDLC Loop mode is selected.

It is set if an EOP sequence has been received.

OLP/RDO... On Loop

Only valid if SDLC Loop mode is selected.
It is set in response to a Go On Loop command, but not before an EOP sequence has been received. It is also set when returning from the Active On Loop state. All incoming bits on R × D are reflected onto T × D with one bit delay.

Receive Data Overflow
Not applicable in SDLC Loop mode
This interrupt status is an early warning that data has been lost. It is classified as group 7 or group 8 interrupt. Even when this interrupt status is generated, the frame continues to be received when space in the RFIFO is available again.

Note: Whereas the bit RSTA.RDO in the frame status byte indicates whether an overflow occurred when receiving the frame currently accessed in the RFIFO, the ISR1.RDO interrupt status is generated as soon as an overflow occurs and does not necessarily pertain to the frame currently accessed by the processor or the DMA controller.

AOLP/ALLS... Active On Loop

Only valid if SDLC Loop mode is selected.
It is set in response to a Go Active On Loop command, but not before an EOP sequence has been received. T × D is disconnected from R × D and transmission of Flags or data is started.

All Sent
Only valid if SDLC loop mode is not selected.

This bit is set
– if the last bit of the current frame is completely sent out on T × D and XFIFO is empty (non-auto mode, transparent modes).
– if an I-Frame is completely sent out on T × D and a positive acknowledgment has been received (auto mode).
– In auto-mode, if an I-frame has been sent and a timer interrupt (TIN) is generated because the internal timer expires before an acknowledgment is received: in this case ALLS is generated one clock period after (TIN).

XDU/EXE... Transmit Data Underrun/Extended Transmission End

Transmitted frame was terminated with an abort sequence because no data was available for transmission in XFIFO and no XME was issued (interrupt mode) or DMA request was not satisfied in time (DMA mode).

Note: Transmitter and XFIFO are reset and deactivated if this condition occurs. They are re-activated not before this interrupt status register has been read. Thus, XDU should not be masked via register IMR1.

In extended transparent mode, this bit indicates the transmission-end condition (EXE).

TIN... Timer Interrupt

The internal timer and repeat counter has expired (see also description of TIMR register).

CSC... Clear To Send Status Change

Indicates that a state transition has occurred on $\overline{\text{CTS}}$. The actual state can be read from STAR register (CTS bit).

XMR... Transmit Message Repeat

The transmission of the last frame has to be repeated because
– the ESCC8 has received a negative acknowledgment to an I-frame in auto-mode, or
– a collision has occurred after at least one FIFO block of data has been completely transmitted, and thus an automatic re-transmission cannot be attempted, or
– $\overline{\text{CTS}}$ (transmission enable) has been withdrawn after at least one FIFO block of data has been transmitted and the frame has not been completed.

Note: For easier recovery in the case of a collision, XFIFO should not contain data of more than one frame.
The use of ALLS interrupt is therefore recommended.

In case an XMR interrupt has occured, an ALLS interrupt is generated one clock period later automatically.

XPR... Transmit Pool Ready

A data block of up to 32 bytes can be written to the transmit FIFO. XPR enables the fastest access to XFIFO. It has to be used for transmission of long frames, back-to-back frames or frames with shared flags. However, starting transmission of a **new** frame should be initiated after ALLS interrupt instead of XPR
– in auto mode
– in bus configurations
– if contents of XFIFO have to be unique, e.g. for automatic repetition of the last frame in case of bus collisions or $\overline{\text{CTS}}$ control
(see also XMR interrupt).

Note: It is not possible to send transparent, or I-frames when a XMR or XDU interrupt remains unacknowledged.

Interrupt Mask Register 0, 1 (WRITE)

Value after RESET: FF_H, FF_H

	7							0	
IMR0	RME	RFS	RSC	PCE	PLLA	CDSC	RFO	RPF	(offset: 3A)
IMR1	EOP	OLP/ RDO	AOLP/ ALLS	XDU/ EXE	TIN	CSC	XMR	XPR	(offset: 3B)

Each interrupt source can generate an interrupt signal at port INT (characteristics of the output stage are defined via register IPC). A “1” in a bit position of IMR0 or IMR1 sets the mask active for the interrupt status in ISR0 or ISR1. Masked interrupt statuses neither generate an interrupt vector or a signal on INT, nor are they visible in register GIS. Moreover, they will

– not be displayed in the Interrupt Status Register if bit IPC.VIS is set to “0”

– be displayed in the Interrupt Status Register if bit IPC.VIS is set to “1”.

Note: After RESET, all interrupts are **dis**abled.

Port Value Register Port A...D (READ/WRITE)

	7					0	
PVRA	PVR7					PVR0	(03C/07C)
PVRB	PVR7					PVR0	(0BC/0FC)
PVRC	PVR7					PVR0	(13C/17C)
PVRD	0	0	0	0	PVR3	PVR0	(1BC/1FC)

Note: Unused bits have to be set to logical “0”.

Each PVR register is accessible via two channel addresses.

Each of the above bits is assigned to the corresponding Universal Port pin with the same number (e.g. PVRA.0 to port pin PA0).

Read Access

PVR shows the value of all pins (input and output). Input values can be separated via software by “AND”-ing PCR and PVR.

Write Access

PVR accepts values for all output pins (defined via PCR). Values written to input pin locations are ignored.

Port Interrupt Status Register Port A...D (READ)

	7					0	
PISA	PIS7					PIS0	(03D/07D)
PISB	PIS7					PIS0	(0BD/0FD)
PISC	PIS7					PIS0	(13D/17D)
PISD	0	0	0	0	PIS3	PIS0	(1BD/1FD)

Each PIS register is accessible via two channel addresses.

Each of the above bits is assigned to the corresponding Universal Port pin with the same number (e.g. PISA.0 to pin PA0). Bit PISn is set and an interrupt is generated on INT if

– the corresponding Universal Port pin Pn is defined as input via register PCR and
– the interrupt source is enabled by resetting the corresponding interrupt mask PIMn in register PIM and
– a state transition has occurred on pin Pn. For definite detection of a real state transition, pulse width should not be shorter than 20 ns.

Note: Bits PISn are reset when register PIS is read. Masked interrupts are not normally indicated when PIS is read. Instead, they remain internally stored and pending. A pending interrupt is generated when the corresponding mask bit is reset to zero. However, if bit IPC.VIS is set to “1”, interrupt statuses in PIS may be flagged although they are masked via register PIM. These masked interrupt statuses neither generate an interrupt vector or a signal on INT, nor are visible in register GIS.
If more than one consecutive state transitions occur on the same pin before the PIS register is read, only one interrupt request will be generated.

Port Interrupt Mask Register Port A...D (WRITE)

Value after RESET: FF_H

	7					0	
PIMA	PIM7					PIM0	(03D/07D)
PIMB	PIM7					PIM0	(0BD/0FD)
PIMC	PIM7					PIM0	(13D/17D)
PIMD	0	0	0	0	PIM3	PIM0	(1BD/1FD)

Note: Unused bits have to be set to logical “0”.

Each PIM register is accessible via two channel addresses.

Each of the above bits is assigned to the corresponding Universal Port pin and to the bits of register PIS with the same number (e.g. PIMA0 to pin PA0).

0… Interrupt source is enabled.

1… Interrupt source is disabled.

A “1” in a bit position of PIM sets the mask active for the interrupt status in PIS. Masked interrupt statuses neither generate an interrupt vector or a signal on INT, nor are they visible in register GIS.

Moreover, they will

– not be displayed in the Interrupt Status Register if bit IPC.VIS is set to “0”

– be displayed in the Interrupt Status Register if bit IPC.VIS is set to “1”.

Refer to description of register PIS.

Note: After RESET, all interrupt sources are **dis**abled.

Port Configuration Register Port A...D (READ/WRITE)

Value after RESET: FF_H

	7					0	
PCRA	PCR7					PCR)	(03E/07E)
PCRB	PCR7					PCR0	(0BE/0FE)
PCRC	PCR7					PCR0	(13E/17E)
PCRD	0	0	0	0	PCR3	PCR0	(1BE/1FE)

Note: Unused bits have to be set to logical “0”.

Each PCR register is accessible via two channel addresses.

Each of the above bits is assigned to the corresponding Universal Port pin with the same number (e.g. PCRA.0 to pin PA0). If bit PCRn (n = 0...7) is set to

0… pin Pn is defined as output.

1… pin Pn is defined as input.

Note: After RESET, all pins of the Universal Port are defined as **in**puts.

Channel Configuration Register 4 (READ/WRITE) (Version 2 upwards)

Value after RESET: 00_H

	7							0	
CCR4	0	0	0	0	0	0	RFT1	RFT0	(offset: 3F)

Note: Unused bits have to be set to logical “0”.

RFT1, RFT0 … RFIFO Threshold Level

The size of the accessible part of RFIFO can be determined by programming these bits. The number of valid bytes after an RPF interrupt is given in the following table:

RFT1	RFT0	Size of Accessible Part of RFIFO
0	0	32 bytes (RESET value)
0	1	16 bytes
1	0	4 bytes
1	1	2 bytes

The value of RFT 1,0 can be changed dynamically

– If reception is not running (recommended: receiver is disabled by setting MODE.RAC to “0”), or

– after RME interrupt has been generated, but before the command CMDR.RMC is issued (DMA controlled data transfer), or

– after the current data block has been read, but before the command CMDR.RMC is issued (interrupt controlled data transfer). **See Note**.

Note: It is seen that changing the value of RFT1,0 is possible even **during** the reception of one frame. The total length of the received frame can be always read directly in RBCL, RBCH after an RPF interrupt, except when the threshold is **increased** during reception of that frame. The real length can then be inferred by noting which bit positions in RBCL are reset by an RMC command (**see table below**):

RFT1	RFT0	Bit Positions in RBCL Reset by a CMDR.RMC Command
0	0	RBC4 …. 0
0	1	RBC3 … 0
1	0	RBC1,0
1	1	RBC0

4.2 Status/Control Registers in ASYNC Mode

4.2.1 Register Addresses

Table 10
Register Addresses in ASYNC Mode

Address (A8 … A0)								Register	
Channel									
0	1	2	3	4	5	6	7	Read	Write
000 … 01F	040 … 05F	080 … 09F	0C0 … 0DF	100 … 11F	140 … 15F	180 … 19F	1C0 … 1DF	RFIFO	XFIFO
020	060	0A0	0E0	120	160	1A0	1E0	STAR	CMDR
021	061	0A1	0E1	121	161	1A1	1E1	———	———
022	062	0A2	0E2	122	162	1A2	1E2	MODE	
023	063	0A3	0E3	123	163	1A3	1E3	TIMR	
024	064	0A4	0E4	124	164	1A4	1E4	XON	
025	065	0A5	0E5	125	165	1A5	1E5	XOFF	
026	066	0A6	0E6	126	166	1A6	1E6	TCR	
027	067	0A7	0E7	127	167	1A7	1E7	DAFO	
028	068	0A8	0E8	128	168	1A8	1E8	RFC	
029	069	0A9	0E9	129	169	1A9	1E9	———	———
02A	06A	0AA	0EA	12A	16A	1AA	1EA	RBCL	XBCL
02B	06B	0AB	0EB	12B	16B	1AB	1EB	RBCH	XBCH
02C	06C	0AC	0EC	12C	16C	1AC	1EC	CCR0	
02D	06D	0AD	0ED	12D	16D	1AD	1ED	CCR1	
02E	06E	0AE	0EE	12E	16E	1AE	1EE	CCR2	
02F	06F	0AF	0EF	12F	16F	1AF	1EF	CCR3	
030	070	0B0	0F0	130	170	1B0	1F0	———	TSAX
031	071	0B1	0F1	131	171	1B1	1F1	———	TSAR
032	072	0B2	0F2	132	172	1B2	1F2	———	XCCR
033	073	0B3	0F3	133	173	1B3	1F3	———	RCCR
034	074	0B4	0F4	134	174	1B4	1F4	VSTR	BGR
035	075	0B5	0F5	135	175	1B5	1F5	———	TIC
036	076	0B6	0F6	136	176	1B6	1F6	———	MXN
037	077	0B7	0F7	137	177	1B7	1F7	———	MXF
038, 078, 0B8, 0F8, 138, 178, 1B8, 1F8								GIS *)	IVA *)
039, 079, 0B9, 0F9, 139, 179, 1B9, 1F9								IPC *)	
03A	07A	0BA	0FA	13A	17A	1BA	1FA	ISR0	IMR0
03B	07B	0BB	0FB	13B	17B	1BB	1FB	ISR1	IMR1
03C, 07C		0BC, 0FC		13C, 17C		1BC, 1FC		PVRA…D	
03D, 07D		0BD, 0FD		13D, 17D		1BD, 1FD		PISA…D	PIMA…D
03E, 07E		0BE, 0FE		13E, 17E		1BE, 1FE		PCRA…D	
03F, 07F		0BF, 0FF		13F, 17F		1BF, 1FF		———	

*) All channel assigned addresses enable access to the same register(s)

Note: Read access to unused register addresses: value should be ignored,
Write access to unused register addresses: should be avoided, or set to “00”$_H$.

4.2.2 Register Definitions

Receive FIFO (READ) RFIFO (offset: 00…1F)

Received data stored in RFIFO (LSB is received first) can be organized in one of two selectable ways (**refer to figure 49**):

– pure data up to a character length of 8 bits (incl. optional parity)

– additionally, one status byte per character with information about parity (if enabled), parity error and framing error.

Reading data from RFIFO can be done in 8-bit (byte) or 16-bit (word) access depending on the selected bus interface mode. The LSB is received first from the serial interface.

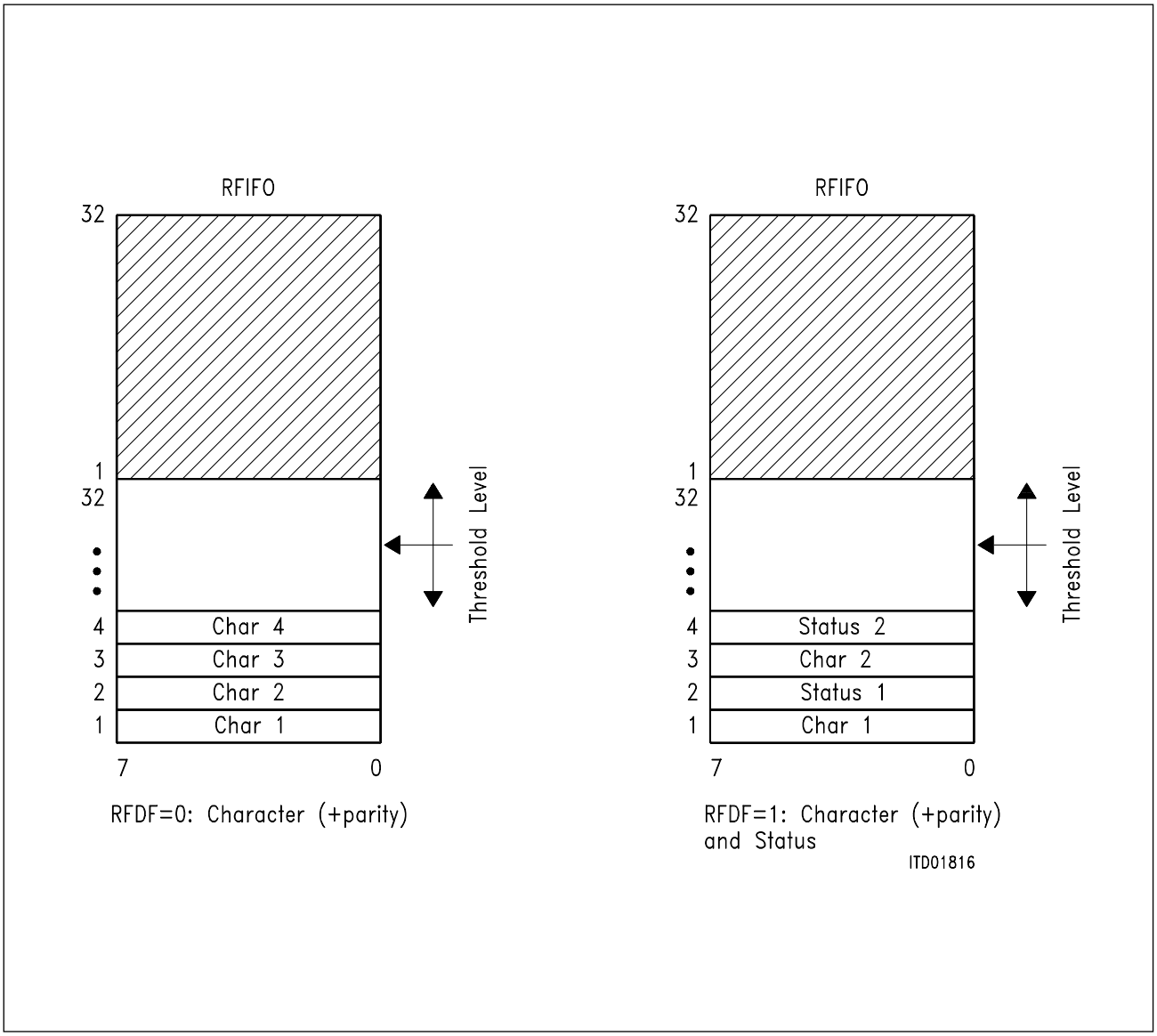


Figure 49
Organization of RFIFO

Interrupt Controlled Data Transfer (Interrupt Mode)

Selected if DMA bit in XBCH is reset.

Up to 32 bytes/16 words of received data can be read from the RFIFO following a RPF or a TCD interrupt depending on the selected RFIFO mode (refer to RFC register):
RPF interrupt: A fixed number of bytes/words (programmed threshold level RFTH0, 1) has to be read by the CPU.

TCD interrupt: Termination character detected. The received data stream is monitored for “termination character” (programmable via register TCR). The number of valid **bytes** in RFIFO is determined by reading the RBCL register.

If necessary, the CPU can access the RFIFO by issuing RFIFO Read command (CMDR.RFRD) before threshold level or the termination condition is reached. The number of valid **bytes** is determined by reading the RBCL register. Additional information: STAR.RFNE: RFIFO Not Empty.

DMA Controlled Data Transfer (DMA Mode)

Selected if DMA bit in XBCH is set.
If the RFIFO contains the number of bytes/words defined via the threshold level, the ESCC8 autonomously requests a DMA block data transfer by DMA by activating the DRRn line until the last valid data is read (the DDRn line remains active up to the beginning of the last read cycle).

This forces the DMA controller to continuously perform bus cycles till all data is transferred from the ESCC8 to the system memory (level triggered transfer mode of DMA controller). If the end condition (TCD) is reached, the same procedure as above is performed. DRRn is activated until the termination character is transferred. A TCD interrupt is issued after the last data has been transferred. Generation of further DMA requests is blocked until TCD interrupt has been acknowledged by issuing an RMC command. The valid **byte** count of the last block can be determined by reading the RBCL register following the TCD interrupt.

Note: Addresses within the 32-byte address space of the FIFO’s point all to the same byte/word, i.e. current data can be accessed with any address within the valid scope.

Transmit FIFO (WRITE) XFIFO (offset: 00…1F)

Writing data to XFIFO can be in 8-bit (byte) or 16-bit (word) access depending on the selected bus interface mode. The LSB is transmitted first.

Interrupt Mode

Selected if DMA bit in XBCH is reset.
Up to 32 bytes/16 words of transmit data can be written to the XFIFO following an XPR interrupt.

DMA Mode

Selected if DMA bit in XBCH is set.
Prior to any data transfer, the actual **byte** count to be transmitted must be written to the XBCH, XBCL registers by the user. Correct transmission of data in the case of word access and of an odd number of bytes specified in XBCH, XBCL is guaranteed.

If data transfer is then initiated via the CMDR register (command XF), the ESCC8 autonomously requests the correct amount of block data transfers (n×BW + REST; BW = 32, 16; n = 0, 1,…).

Note: Addresses within the 32-byte address space of the FIFO all point to the same byte/word, i.e. current data can be accessed with any address within the valid range.

Status Register (READ)

	7							0	
STAR	XDOV	XFW	RFNE	FCS	TEC	CEC	CTS	0	(offset: 20)

XDOV… Transmit Data Overflow

More than 32 bytes have been written to the XFIFO.

This bit is reset by:

– a transmitter reset command XRES
– or when all bytes in the accessible half of the XFIFO have been moved in the inacessible half.

XFW… Transmit FIFO Write Enable

Data can be written to the XFIFO.

RFNE… RFIFO Not Empty

This bit is set if the accessible part of RFIFO holds at least one valid byte.

FCS… Flow Control Status

If in-band flow control is enabled via bit MODE.FLON, this status bit indicates the current state of the transmitter:

0… The transmitter is in XON state, i.e. transmission is enabled or running.

1… The transmitter is in XOFF state, i.e. transmission is stopped and disabled until an XON character is detected by the receiver.

TEC… **TIC Executing**

This status bit indicates that transmission instruction of currently programmed TIC (Transmit Immediate Character) is accepted but not completely executed. Further access to register TIC is only allowed after STAR.TEC has been reset by the ESCC8.

Note: Status flag TEC is set immediately with the write access to register TIC. It remains active until the transmitter of ESCC8 is able to start transmission of currently programmed TIC. Best case: TEC remains set for at most 2.5 clock periods (transmit clock or master clock, depending of the selected mode) if transmission of the programmed TIC character can be started immediately.
The function of register TIC and status flag TEC is independent of whether flow control is enabled or not.

CEC… **Command Executing**

0… no command is currently being executed, the CMDR register can be written to.

1… a command (written previously to CMDR) is currently being executed, no further command can be temporarily written in CMDR register.

Note: CEC will be active at most 2.5 transmit clock periods. If the ESCC8 is in power down mode CEC will stay active.

CTS… **Clear To Send State**

This bit indicates the state of the CTS pin.

0… $\overline{\text{CTS}}$ is inactive (high)

1… $\overline{\text{CTS}}$ is active (low)

Command Register (WRITE)

Value after RESET: 00_H

	7							0	
CMDR	RMC	RRES	RFRD	STI	XF	0	0	XRES	(offset: 20)

Note: Unused bits have to be set to logical “0”.
The maximum time between writing to the CMDR register and the execution of the command is 2.5 clock cycles. Therefore, if the CPU operates with a very high clock rate in comparison with the ESCC8’s clock, it is recommended that the CEC bit of the STAR register be checked before writing to the CMDR register to avoid any loss of commands.

RMC… Receive Message Complete

Confirmation from CPU to ESCC8 that the current data block has been fetched following a RPF or TCD interrupt or following a user initiated read access in conjunction with the RFIFO Read command RFRD; the occupied space in the RFIFO can be released.

Note: In DMA Mode, this command has to be issued after a TCD interrupt in order to enable the generation of further receiver DMA requests.

RRES… Receiver Reset

All data in RFIFO and ASYNC receiver is deleted.

RFRD… Receive FIFO Read Enable

The CPU can have access to RFIFO by issuing the RFRD command before threshold level or the end condition (TCD) are fulfilled. After issuing the RFRD command the CPU has to wait for TCD interrupt, before reading RBC and RFIFO. The number of valid bytes is determined by reading the RBCL register.

STI… Start Timer

The internal timer is started.

Note: The timer is stopped by rewriting the TIMR register after start.

XF… Transmit Frame

- Interrupt Mode
  After having written up to 32 bytes/16 words to the XFIFO, this command initiates the transmission of data.
- DMA Mode
  After having written the amount of data to be transmitted to the XBCH, XBCL registers, this command initiates the data transfer from system

memory to ESCC8 by DMA. Serial data transmission starts as soon as 32 bytes/16 words are stored in the XFIFO or the Transmit Byte Counter value is reached.

XRES… Transmitter Reset

XFIFO is cleared of any data and IDLE (logical “1s”) is transmitted. This command can be used by the CPU to abort current data transmission. In response to XRES an XPR interrupt is generated.

Mode Register (READ/WRITE)

Value after RESET: 00_H

	7							0	
MODE	0	0	0	FLON	RAC	RTS	TRS	TLP	(offset: 22)

Note: Unused bits have to be set to logical “0”.

FLON… Flow Control ON

The in-band flow control is activated via this bit:

0… No further action is automatically taken by the ESCC8. However, recognition of an XON or an XOFF character (defined via registers XON and XOFF) causes always a corresponding maskable interrupt status to be generated (**refer to register ISR1**).

1… The reception of an XOFF character (defined via register XOFF) automatically turns off the transmitter after the currently transmitted character (if any) has been completely shifted out (XOFF state). The reception of an XON character (defined via register XON) automatically makes the transmitter resume transmitting (XON state).

RAC… Receiver Active

Switches the receiver to operational or inoperational state.

0… receiver inactive

1… receiver active

RTS… **Request To Send**

Defines the state and control of $\overline{\text{RTS}}$ pin.

0… The $\overline{\text{RTS}}$ pin is controlled by the ESCC8 autonomously.
$\overline{\text{RTS}}$ is activated when data transmission starts and deactivated when transmission is completed.

1… The $\overline{\text{RTS}}$ pin is controlled by the CPU.
If this bit is set, the $\overline{\text{RTS}}$ pin is activated immediately and remains active till this bit is reset.

TRS… **Timer Resolution**

Selects the resolution of the internal timer (factor k, see description of TIMR register):
0…k = 32 768
1…k = 512

TLP… **Test Loop**

Input and output of the ASYNC channels are internally connected.
(e.g. transmitter channel 0 - receiver channel 0)

Timer Register (READ/WRITE)

	7		0	
TIMR	CNT	VALUE		(offset: 23)

VALUE… (5 bits) sets the time period t1 as follows:

$t_1 = k \times (\text{VALUE} + 1) \times \text{TCP}$

where

– k is the timer resolution factor which is either 32 768 or 512 clock cycles dependent on the programming of TRS bit in MODE.

– TCP is the clock period of transmit data.

CNT… (3 bits)

CNT plus VALUE determine the time period t_2 after which a timer interrupt will be generated. The time period t_2 is

$t_2 = 32 \times k \times \text{CNT} \times \text{TCP} + t_1$.

If CNT is set to 7, a timer interrupt is periodically generated after the expiration of t_1.

XON Character (READ/WRITE)

Value after RESET: 00_H

	7		0	
XON	XON7		XON0	(offset: 24)

This register is used to specify the XON character. It can be used in conjunction with the interrupt status ISR1.XON for automatic in-band flow control (if MODE.FLON = “0”). The number of significant bits is determined by the programmed character length (right justified).

A received character is considered to be recognized as a valid XON character

– if it is correctly framed (correct length),

– if its bits match the (unmasked) ones in the XON register over the programmed character length,

– if it has correct parity (if applicable).

Received XON characters are always stored in the receive FIFO, similar to other characters.

XOFF Character (READ/WRITE)

Value after RESET: 00_H

	7		0	
XOFF	XOF7		XOF0	(offset: 25)

This register is used to specify the XOFF character. It can be used in conjunction with the interrupt status ISR1.XOFF for automatic in-band flow control (if MODE.FLON = “1”), or as a special character compare register for other purposes (if MODE.FLON = “0”). The number of significant bits is determined by the programmed character length (right justified).

A received character is considered to be recognized as a valid XOFF character

– if it is correctly framed (correct length),

– if its bits match the (unmasked) ones in the XOFF register over the programmed character length,

– if it has correct parity (if applicable).

Received XOFF characters are always stored in the receive FIFO, similar to other characters.

Termination Character Register (READ/WRITE)

Value after RESET: 00_H

	7						0	
TCR	TCR7						TCR0	(offset: 26)

TCR7–TCR0… Termination Character

If enabled via register RFC the received data stream is monitored for the occurrence of a programmed “termination character”. When such a character is found, an interrupt is issued if enabled via mask register IMR0. The number of valid bytes in the RFIFO up to and including the termination character is determined by reading the RBCL register.

Note: If selected character length is less than eight bits, leading (unused) bits of TCR have to be set to “0”.

Data Format (READ/WRITE)

Value after RESET: 00_H

	7						0	
DAFO	0	XBRK	STOP	PAR1 PAR0		PARE	CHL1 CHL0	(offset: 27)

Note: Unused bits have to be set to logical “0”.

XBRK… Transmit Break

0… Normal operation for data transmission.

1… This command forces the T×D pin to go low, regardless of any data being transmitted at this time. This command is executed immediately (with the next rising edge of Transmit Clock) and the transmitter is disabled. The current character is lost. However, the contents of XFIFO are still available and are sent out as soon as this bit is reset. To avoid this the Transmit Reset command XRES should be issued. If XBRK is still set when XRES is issued the Break signal on T×D stays active.

STOP… Stop Bit

This bit defines the number of Stop bits generated by the transmitter:

0…1 Stop bit.

1…2 Stop bits.

PAR1, PAR0… Parity Format

If parity check/generation is enabled by setting PARE, these bits define the parity type:

00… SPACE (“0”)
01… odd parity
10… even parity
11… MARK (“1”)

The received parity bit is stored in RFIFO

– as leading bit immediately preceding the character if character length is 5 to 7 bits and RFC.DPS is set to 0, and as LSB of the status byte pertaining to the character if the corresponding RFIFO data format is enabled.

– as LSB of the status byte pertaining to the character if character length is 8 bits and the corresponding RFIFO data format is enabled.

Parity error is indicated in the MSB of the status byte pertaining to the character, if enabled. Additionally, a parity error interrupt can be generated.

PARE… Parity Enable

0… parity check/generation disabled

1… parity check/generation enabled

CHL1–CHL0… Character Length

These bits define the length of received and transmitted characters, excluding optional parity:

00… 8 bit
01… 7 bit
10… 6 bit
11… 5 bit

RFIFO Control Register (READ/WRITE)

Value after RESET: 00_H

	7						0	
RFC	0	DPS	0	RFDF	RFTH1	RFTH0	0	TCDE	(offset: 28)

Note: Unused bits have to be set to logical “0”.

DPS… Disable Parity Storage

Only valid if parity check/generation is enabled via DAFO.PARE and character length is less than 8 bits.

0… The parity bit is stored
1… The parity bit is **not** stored
in the data byte of RFIFO.

Note: The parity bit is always stored in the status byte.

RFDF… RFIFO Data Format

0… only data bytes (character plus optional parity up to 8 bit) are stored.

1… additionally to every data byte, an attached status byte is stored.

RFDF = 0	**RFDF = 1**
– character 5 – 8 bit or – character 5 – 7(8)* bit + parity * : parity bit is lost	– character 5 – 8 bit + status or – character 5 – 7(8)* bit + parity + status * : parity bit is in status byte

	7	4	0
Data Byte		(P)	Char

	7	4	0
Data Byte		(P)	Char

	7	6	0
Status Byte	PE	FE	P

FE : framing error PE : parity error P : parity bit (P): can be disabled via bit DPS

RFTH1, RFTH0… RFIFO Threshold Level

These bits define the level up to which RFIFO is filled with valid data:

RFTH1, 0	Threshold level (bytes)	
	RFDF = 0	RFDF = 1
00	1 (1d)	**2** (**1d** + 1s)
01	4 (4d)	4 (2d + 2s)
10	16 (16d)	16 (8d + 8s)
11	32 (32d)	32 (16d + 16s)
	d: data byte	s: status byte

If the threshold level is reached, the RPF interrupt is generated if enabled. After RPF is generated, the contents of RFIFO (RFTH bytes) can be read by the CPU. To indicate that this RFIFO pool can be released, an RMC command has to be issued.

TCDE… Termination Character Detection Enable

When this bit is set, the received data stream is monitored for “termination character” (TCR register). When such a character occurs, the TCD interrupt is generated if enabled via mask register IMR0. The number of **bytes** to be read from RFIFO is determined by the value of RBCL.

Receive Byte Count Low (READ)

	7							0	
RBCL	RBC7							RBC0	(offset: 2A)

Indicates the number of valid bytes available in the accessible part of the RFIFO. This register must be read by the CPU following a TCD interrupt. In case of a TCD interrupt the number of valid bytes in the accessible part of the RFIFO can be evaluated by “AND”-ing the contents of RBCL with: threshold level (bytes) – 1.

Threshold Level	Mask
4	03_H
16	$0F_H$
32	$1F_H$

RBC is reset with RMC after preceeding TCD interrupt.

In case of RPF interrupt RBC is incremented by “threshold level (bytes)”.

Transmit Byte Count Low (WRITE)

	7						0	
XBCL	XBC7						XBC0	(offset: 2A)

Together with XBCH (bits XBC11…XBC8) this register is used in DMA Mode only, to program the length (1…4096 bytes) of the next data block to be transmitted.

In terms of the value xbc, programmed in XBC11…XBC0 (xbc = 0…4095), the length of the block in number of bytes is: length = xbc + 1.

This allows the ESCC8 to request the correct amount of DMA cycles after an XF command in CMDR.

Received Byte Count High (READ)

Value after RESET: 000xxxxx

	7						0	
RBCH	DMA	0	CAS	0	RBC11		RBC8	(offset: 2B)
	see XBCH							

DMA, CAS… These bits represent the read-back value programmed in XBCH

RBC11– RBC8… Receive Byte Count (most significant bits)

No function in ASYNC mode.

Transmit Byte Count High (WRITE)

Value after RESET: 000xxxxx

	7						0	
XBCH	DMA	0	CAS	XC	XBC11		XBC8	(offset: 2B)

Note: Unused bits have to be set to logical “0”.

DMA… DMA Mode

Selects the data transfer mode of ESCC8 to/from System Memory.

0… Interrupt controlled data transfer (Interrupt Mode).

1… DMA controlled data transfer (DMA Mode).

CAS… Carrier Detect Auto Start

When set, a high on the CD pin enables the corresponding receiver and data reception is started. When not set, if not in Clock Mode 1 or 5, the CD pin can be used as a general input.

XC… Transmit Continuously

Only valid if DMA Mode is selected.

If the XC bit is set, the ESCC8 continuously requests for transmit data ignoring the transmit byte count programmed via XBCH, XBCL.

XBC11– XBC8… Transmit Byte Count (most significant bits)

Valid only if DMA Mode is selected.

Together with XBCL (bits XBC7…XBC0), determine the number of characters to be transmitted.

Channel Configuration Register 0 (READ/WRITE)

Value after RESET: 00_H

	7							0	
CCR0	PU	MCE	0	SC2	SC1	SC0	SM1	SM0	(offset: 2C)

Note: Unused bits have to be set to logical “0”.

PU… Switches between power up and power down mode

0… power down (standby)

1… power up (active)

MCE… Master Clock Enable

If this bit is set to “1”, the clock provided via pin XTAL1 works as master clock to allow full functionality of the microprocessor interface (access to all status and control registers, DMA and interrupt support) independent of the receive and the transmit clocks. The internal oscillator in conjunction with a crystal on XTAL1-2 can be used, too. The master clock option is not applicable in clock mode 5. **Refer to table 5** for more details.

Note: The internal timers run with the master clock.

SC2– SC0… Serial Port Configuration

000… NRZ data encoding
001… (not recommended)
010… NRZI data encoding
011… (not recommended)
100… FM0 data encoding
101… FM1 data encoding
110… MANCHESTER data encoding
111… (not used)

SM1– SM0… Serial Mode

00… HDLC/SDLC mode
01… SDLC Loop mode
10… BISYNC mode
11… ASYNC mode

Channel Configuration Register 1 (READ/WRITE)

Value after RESET: 00_H

	7						0	
CCR1	0	0	0	ODS	BCR	CM2	CM0	(offset: 2D)

Note: Unused bits have to be set to logical “0”.

ODS… Output Driver Select

Defines the function of the transmit data pins (T×DA, T×DB)

0… T×D pin is an open drain output.
1… T×D pin is a push-pull output.

BCR… Bit Clock Rate

This bit is only valid in clock modes **not** using the DPLL (0, 1, 3b, 4, 7b).

0… selects isochronous operation with a bit clock rate = 1. Data is sampled once.

1… selects standard asynchronous operation with a bit clock rate = 16. Data is sampled 3 times around the nominal bit center. The effective bit value is determined by majority decision. For correct operation, **NRZ** data encoding has to be selected.

CM2– CM0… Clock Mode

Selects one of 8 different clock modes:

000 clock mode 0
• •
• •
• •
111 clock mode 7

Note: Clock mode 5 is only specified for version SAB 82538H-10, not for SAB 82538H.

Channel Configuration Register 2 (READ/WRITE)

Value after RESET: 00_H

The meaning of the individual bits in CCR2 depends on the clock mode selected via CCR1 as follows:

7 0

CCR2 clock mode 0a, 1

SOC1	SOC0	0	0	0	RWX	0	DIV	(offset: 2E)

clock mode 0b, 2, 3, 6, 7

BR9	BR8	BDF	SSEL	TOE	RWX	0	DIV

clock mode 4

SOC1	SOC0	0	0	TOE	RWX	0	DIV

clock mode 5

SOC1	SOC0	XCS0	RCS0	TOE	RWX	0	DIV

Note: Unused bits have to be set to logical “0”.

SOC1, SOC0… Special Output Control

In a bus configuration (selected via CCR0) defines the function of pin $\overline{RTS}$ as follows:

0X… $\overline{RTS}$ output is activated during transmission of characters.
10… $\overline{RTS}$ output is always high ($\overline{RTS}$ disabled).
11… $\overline{RTS}$ indicates the reception of a data frame (active low).

In a point-to-point configuration (selected via CCR0) the T×D and R×D pins may be flipped

0X… data is transmitted on T×D, received on R×D (normal case).
1X… data is transmitted on R×D, received on T×D.

BR9, BR8… Baud Rate, Bit 9-8

High order bits, see description of BGR register.

XCS0, RCS0... Transmit/Receive Clock Shift, Bit 0

Together with XCS2, XCS1 (RCS2, RCS1) in TSAX (TSAR), determines the clock shift relative to the frame synchronization signal of the Transmit (Receive) time-slot.
A clock shift of 0 ... 7 bits is programmable (clock mode 5 only).

BDF... **Baud Rate Division Factor**

0…The division factor of the baud rate generator is set to 1 (constant).
1…The division factor is determined by BR9 - BR0 bits in CCR2 and BRG registers.

SSEL... **Clock Source Select**

Selects the clock source in clock modes 0, 2, 3, 6 and 7 (**refer to table 5**).

TOE... **T×CLK Output Enable**

0… T×CLK pin is input

1… T×CLK pin is switched to output function if applicable to the selected clock mode (**refer to table 5**).

RWX... **Read/Write Exchange**

Valid only in DMA mode. If this bit is set, the
– $\overline{RD}$ and $\overline{WR}$ pins are internally exchanged (Siemens/Intel bus interface)
– R/$\overline{W}$ pin is inverted in polarity (Motorola bus interface)
while any $\overline{DACK}$ input is active. This useful feature allows a simple interfacing to the DMA controller.

Note: The RWX bit of all eight channels is “or”ed.

DIV... **Data Inversion**

Only valid if NRZ data encoding is selected. Data is transmitted and received inverted.

Channel Configuration Register 3 (READ/WRITE)

(Version 2 upwards)

Value after RESET: 00_H

	7							0	
CCR3	0	0	0	0	0	0	0	PSD	(offset: 2F)

Note: Unused bits have to be set to logical “0”.

PSD… **DPLL Phase Shift Disable**

Only applicable in the case of NRZ and NRZI encoding.

If this bit is set to “1”, the Phase Shift function of the DPLL is disabled. In this case the windows for Phase Adjustment are extended.

Time-Slot Assignment Register Transmit (WRITE)

This register is only used in clock mode 5!

Value after RESET: 00_H

	7			0	
TSAX	TSNX		XCS2	XCS1	(offset: 30)

TSNX… Time-slot Number Transmit

Selects one of up 64 possible time-slots (00_H-$3F_H$) in which data is transmitted. The number of bits per time-slot can be programmed via XCCR.

XCS2… XCS1 … Transmit Clock Shift, Bit 2-1

Together with bit XCS0 in CCR2, transmit clock shift can be adjusted.

Time-Slot Assignment Register Receive (WRITE)

This register is only used in clock mode 5!
Value after RESET: 00_H

	7			0	
TSAR	TSNR		RCS2	RCS1	(offset: 31)

TSNR… Time-slot Number Receive

Defines one of up to 64 possible time-slots (00_H– $3F_H$) in which data is received. The number of bits per time-slot can be programmed via RCCR.

RCS2– RCS1… Receive Clock Shift, Bit 2-1

Together with bit RCS0 in CCR2, the receive clock shift can be adjusted.

Transmit Channel Capacity Register (WRITE)

This register is only used in clock mode 5!
Value after RESET: 00_H

	7		0	
XCCR	XBC7		XBC0	(offset: 32)

XBC7– XBC0… Transmit Bit Number Count, Bit 7-0

Defines the number of bits to be transmitted within a time-slot:
Number of bits = XBC + 1 (1 … 256 bits/time-slot).

Receive Channel Capacity Register (WRITE)

This register is only used in clock mode 5!
Value after RESET: 00_H

	7						0	
RCCR	RBC7						RBC0	(offset: 33)

RBC7– RBC0… Receive Bit Count, Bit 7-0

Defines the number of bits to be received within a time-slot:
Number of bits = RBC + 1 (1…256 bits/time-slot).

Version Status Register (READ)

	7					0	
VSTR	CD	DPLA	0	0	VN3	VN0	(offset: 34)

CD… Carrier Detect

This bit reflects the state of the CD pin.

1… CD active
0… CD inactive

DPLA… DPLL Asynchronous

This bit is only valid when the receive clock is supplied by the DPLL and FM0, FM1 or Manchester data encoding is selected.

It is set when the DPLL has lost synchronization. Reception is disabled (IDLE is inserted) until synchronization has been regained. Additionally, transmission is interrupted, too, if the transmit clock is derived from the DPLL (same effect as the deactivation of pin $\overline{\text{CTS}}$).

VN3– VN0… Version Number of Chip

0… Version 1
1… Version 2

Baud Rate Generator Register (WRITE)

	7		0	
BGR	BR7		BR0	(offset: 34)

BR7– BR0… Baud Rate, bits 7-0

Together with bits BR9, BR8 of CCR2, determines the division factor of the baud rate generator.

In terms of the value N programmed in BR9 - BR0 (N = 0…1023), the division factor k is:

$k = (N + 1) \times 2$

Transmit Immediate Character (WRITE)

(Version 2 upwards)

	7		0	
TIC	TIC7		TIC0	(offset: 35)

When a character is written into this register its contents are inserted in the outgoing character stream

– immediately upon writing this register by the microprocessor if the transmitter is in IDLE state. If no further characters (XFIFO contents) are to be transmitted, i.e. the transmitter returns to IDLE state after transmission of TIC, an ALLS (All Sent) interrupt will be generated.

– after the end of a character currently being transmitted. This does not affect the contents of the XFIFO. Transmission of characters from XFIFO is resumed after the contents of register TIC are shifted out.

Transmission via this register is possible even when the transmitter is in XOFF state (however, $\overline{\text{CTS}}$ must be “low”).

The TIC register is an eight-bit register. The number of significant bits is determined by the programmed character length (right justified). Parity value (if programmed) and selected number of stop bits are automatically appended, similar to the characters written in the XFIFO. The usage of TIC is independent of the flow control, i.e. is not affected by bit MODE.FLON.

To control access to register TIC, an additional status bit STAR.TEC (TIC Executing) is implemented which indicates that transmission instruction of currently programmed TIC is accepted but not completely executed. Further access to register TIC is only allowed if bit STAR.TEC is reset by the ESCC8.

Mask XON Character (WRITE)

(Version 2 upwards)

Value after RESET: 00_H

	7		0	
MXN	MXN7		MXN0	(offset: 36)

This register is used to masked single bit positions of the XON character. Refer to the description of the XON register. The number of significant bits is determined by the programmed character length (right justified).

A “1” in the mask register has the effect that no comparison is performed between the corresponding bits in the received characters (“don’t cares”) and the XON register. At RESET, the mask register is zeroed, i.e. all bit positions are compared.

Mask XOFF Character (WRITE)

(Version 2 upwards)

Value after RESET: 00_H

	7		0	
MXF	MXF7		MXF0	(offset: 37)

This register is used to mask single bit positions of the XOFF character. Refer to the description of the XOFF register. The number of significant bits is determined by the programmed character length (right justified).

A “1” in the mask register has the effect that no comparison is performed between the corresponding bits in the received characters (“don’t cares”) and the XOFF register. At RESET, the mask register is zeroed. i.e. all bit positions are compared.

Global Interrupt Status Register (READ)

Value after RESET: 00_H

	7							0	
GIS	PIA	PIB	PIC	PID	CII	CN2	CN1	CN0	(038/078/0B8/0F8) (138/178/1B8/1F8)

This status register points to pending

– channel assigned interrupts (ISR0_x, ISR1_x)

– universal port interrupts (PISA…D).

GIS is accessible via eight channel addresses (038_H to $1F8_H$).

PIA– PID… Port Interrupt Indication

These status bits point to pending interrupts in corresponding Port Interrupt Status registers PISA…PISD. They may be set independently from channel assigned interrupts.

CII… Channel Interrupt Indication

Set if at least one interrupt source of any channel is active.

CN2– CN0… Channel Number (0…7)

If at least one interrupt source is active (bit CII is set), these bits point to the channel with currently highest source priority. **Refer to chapter 2.2.3** for detailed description of the priority structure.

Contents of register GIS are frozen after every input acknowledge cycle.

– after the first read access to GIS after the interrupt vector has been output,

– after every read access to anyone of the channel assigned interrupt status registers,

– during every $\overline{\text{INTA}}$ cycle.

Interrupt Vector Address (WRITE)

Value after RESET: 00_H

	7							0	
IVA	T7	T6	T5	T4	T3	T2	ROT	EDA	(038/078/0B8/0F8) (138/178/1B8/1F8)

Note: Unused bits have to be set to logical “0”.

IVA is accessible via eight channel addresses (38_H to $1F8_H$).

Version 2 upward provides dynamic adjustment of channel priorities by programming the “highest priority channel”. Selection of the “highest priority channel” is done with every **write access** to IVA in conjunction with the channel assigned IVA register address:

IVA Register Address:	Highest Priority Channel
38_H	0
78_H	1
$B8_H$	2
$F8_H$	3
138_H	4
178_H	5
$1B8_H$	6
$1F8_H$	7

The priority level becomes valid with the end of the write access to the IVA register (rising edge of $\overline{WR}$ or $\overline{DS}$, whichever applies) and remains stable until a new write access to this register occurs.

T7– T6… Device Address

These bits define the value of bits 6 and 7 of the interrupt vector which is sent out on the data bus (D0…D7) during the interrupt acknowledge cycle.

T5… Device Address

Version 1: Device Address

This bit defines the value of bit 5 of the interrupt vector which is sent out on the data bus (D0…D7) during the interrupt acknowledge cycle.

Version 2: Device Address Extension

In Interrupt vector mode 2 (bit EDA set) this bit defines the value of bit 5 of the interrupt vector which is sent out on the data bus (D0…D7) during the interrupt acknowledge cycle.

T4– T2… Device Address Extension

In Interrupt vector mode 2 (bit EDA set) these bits define the value of bits 2 to 4 of the interrupt vector which is sent out on the data bus (D0…D7) during the interrupt acknowledge cycle.

ROT… Rotating Interrupt Priority (version 2 upward)

Version 1:

This bit is unused and has to be set to logical “0”.

Version 2:

0 … Fixed Interrupt Priority
The relative order of the interrupt priority level assigned to the channels is fixed (**refer to chapter 2.3.1**).

1 … Rotating Interrupt Priority
The interrupt priority level will be adjusted after an interrupt has been serviced. Together with bit IPC.ROTM the interrupt priority mode is selected.

IPC.ROTM = 0: The priority level of all 8 serial channels are adjusted.

IPC.ROTM = 1: The priority level of only 7 channels are adjusted while one channel is fixed.

EDA… Extended Device Address

If set, bits 2 to 5 (version 1: bits 2 to 4) of the generated interrupt vector contain the Device Address Extension T2…T5 (version 1: T2…T4) instead of the channel number. For detailed information **refer to chapter 2.2.3**.

Interrupt Port Configuration (READ/WRITE)

Value after RESET: 00_H

	7							0	
IPC	VIS	ROTM	SLA2	SLA1	SLA0	CASM	IC1	IC0	(039/079/0B9/0F9) (139/179/1B9/1F9)

Note: Unused bits have to be set to logical “0”.

IPC is accessible via eight channel addresses (039_H to $1F9_H$).

VIS… Masked Interrupts Visible (version 2 upward)

0… Masked interrupt status bits are not visible.
1… Masked interrupt status bits are visible.

ROTM… Rotating Interrupt Priority Mode (version 2 upward)

Together with bit IVA.ROT the interrupt priority mode is selected.

0 … With IVA.ROT = 1 the priorities of all 8 serial channels are rotated cyclically after an interrupt has been serviced. The channel last serviced is assigned the lowest priority of all (**refer to chapter 2.2.3.1**).

1 … With IVA.ROT = 0 the priority adjustment is performed only on 7 channels while one channel is fixed to highest priority level (**refer to chapter 2.2.3.1**).

SLA2… SLA0… Slave Address

Only used in Slave Cascading Mode (refer to CASM).

CASM… Cascading Mode

0… Slave Cascading Mode

Pins IE0, IE1 and IE2 are used as inputs. Interrupt acknowledge is accepted if an interrupt signal has been generated and the values on pins IE0, IE1and IE2 correspond to the programmed values in SLA0, SLA1 and SLA2 (slave address).

1…Daisy Chaining Mode

Pin IE0 as Interrupt Enable Output and pin IE1 as Interrupt Enable Input are used for building a Daisy Chain. Pin IE2 is not used. Interrupt acknowledge is accepted if an interrupt signal has been generated and Interrupt Enable Input IE1 is active during a subsequent INTA cycle(s). If pin INT goes active, Interrupt Enable Output IE0 is immediately set to “low”.

IC1– IC0… Interrupt Port Configuration

These bits define the function of the interrupt output stage (pin INT):

IOC1	IOC0	Function
X	0	Open drain output
0	1	Push/pull output, active low
1	1	Push/pull output, active high

Interrupt Status Register 0 (READ)

Value after RESET: 00_H

	7							0	
ISR0	TCD	TIME	PERR	FERR	PLLA	CDSC	RFO	RPF	(offset: 3A)

All bits are reset when ISR0 is read. Additionally, TCD and RPF are reset when the corresponding interrupt vector is output.

Note: If bit IPC.VIS is set to “1”, interrupt statuses in ISR0 may be flagged although they are masked via register IMR0. However, these masked interrupt statuses neither generate an interrupt vector or a signal on INT, nor are visible in register GIS.

TCD… Termination Character Detected

The termination character (TCR) has been received or the execution of the RFRD command issued before has been completed. A data block is now available in the RFIFO. The actual block length can be determined by reading register RBCL first.

TIME… Time OUT

The time-out limit has been exceeded.

If the respective mask bit is reset (i.e. TIME interrupt is enabled), the received data stream is monitored for exceeding the fixed time limit after the last character has been received (time limit = 4 $\times$ CFL; character frame length CFL includes start bit, character length, parity bit and stop bits).

PERR… Parity Error

Only valid if parity check/generation is enabled.

If set, a character with parity error has been received. If enabled via RFDF, parity error information is stored in RFIFO in the status byte pertaining to that character.

FERR… Framing Error

This bit indicates that a character has been received with a framing error, i.e. the receiver has detected a “0” in a stop bit position. If enabled via RFDF, this information is stored in RFIFO in the status byte pertaining to that character.

PLLA… **DPLL Asynchronous**

This bit is only valid when the receive clock is supplied by the DPLL and FM0, FM1 or Manchester data encoding is selected.

It is set when the DPLL has lost synchronization. Reception is disabled (IDLE is inserted) until synchronization has been regained. Additionally, transmission is also interrupted if the transmit clock is derived from the DPLL.

CDSC… **Carrier Detect Status Change**

Indicates that a state transition has occurred on CD. The actual state of CD can be read from the VSTR register.

RFO… **Receive FIFO Overflow**

This interrupt is generated if RFIFO is full and a further character is received. This interrupt can be used for statistical purposes and indicates that the CPU does not respond quickly enough to an RPF or TCD interrupt.

RPF… **Receive Pool Full**

This bit is set if RFIFO is filled with data (character and optional status information) up to the programmed threshold level.

Note: This interrupt is only generated in Interrupt Mode.

Interrupt Status Register 1 (READ)

Value after RESET: 00_H

	7							0	
ISR1	BRK	BRKT	ALLS	XOFF	TIN	CSC	XON	XPR	(offset: 3B)

All bits are reset when ISR1 is read. Additionally, XPR is reset when the corresponding interrupt vector is output.

Note: If bit IPC.VIS is set “1”, interrupt statuses in ISR1 may be flagged although they are masked via register IMR1. However, these masked interrupt statuses neither generate an interrupt vector or a signal on INT, nor are visible in register GIS.

BRK… Break

This bit is set when a Break signal - static low level for a time equal to (character length + parity + stop bit(s)) – is detected on R×D.

BRKT… Break Terminated

This bit is set when a Break signal on R×D is terminated.

ALLS… All Sent

This bit is set when the XFIFO is empty and the last character is completely sent out on T×D.

XOFF… XOFF Character Detected

This interrupt status indicates that the currently received character matches the value specified via register XOFF. The function is independent of the programming of bit MODE.FLON.

TIN… Timer Interrupt

The internal timer has expired (see also description of TIMR register).

CSC… Clear To Send Status Change

Indicates that a state transition has occurred on $\overline{CTS}$. The actual state of $\overline{CTS}$ can be read from STAR register (CTS bit).

XON… XON Character Detected

This interrupt status indicates that the currently received character matches the value specified via register XON. The function is independent of the programming of bit MODE.FLON.

XPR… Transmit Pool Ready

A data block of up to 32 bytes can be written to XFIFO.

Interrupt Mask Register 0, 1 (WRITE)

Value after RESET: FF_H, FF_H

	7							0	
IMR0	TCD	TIME	PERR	FERR	PLLA	CDSC	RFO	RPF	(offset: 3A)
IMR1	BRK	BRKT	ALLS	XOFF	TIN	CSC	XON	XPR	(offset: 3B)

Each interrupt source can generate an interrupt signal at port INT (characteristics of the output stage are defined via register IPC). A “1” in a bit position of IMR0 or IMR1 sets the mask active for the interrupt status in ISR0 or ISR1. Masked interrupt statuses neither generate an interrupt vector or a signal on INT, nor are they visible in register GIS. Moreover, they will

– not be displayed in the Interrupt Status Register if bit IPC.VIS is set to “0”

– be displayed in the Interrupt Status Register if bit IPC.VIS is set to “1”.

Note: After RESET, all interrupts are **dis**abled.

Port Value Register Port A...D (READ/WRITE)

	7					0	
PVRA	PVR7					PVR0	(03C/07C)
PVRB	PVR7					PVR0	(0BC/0FC)
PVRC	PVR7					PVR0	(13C/17C)
PVRD	0	0	0	0	PVR3	PVR0	(1BC/1FC)

Note: Unused bits have to be set to logical “0”.

Each PVR register is accessible via two channel addresses.

Each of the above bits is assigned to the corresponding Universal Port pin with the same number.(e.g. PVRA.0 to port pin PA0)

Read Access

PVR shows the value of all pins (input and output). Input values can be separated via software by “AND”-ing PCR and PVR.

Write Access

PVR accepts values for all output pins (defined via PCR). Values written to input pin locations are ignored.

Port Interrupt Status Register Port A...D (READ)

	7						0	
PISA	PIS7						PIS0	(03D/07D)
PISB	PIS7						PIS0	(0BD/0FD)
PISC	PVR7						PIS0	(13D/17D)
PISD	0	0	0	0	PIS3		PIS0	(1BD/1FD)

Each PIS register is accessible via two channel addresses.

Each of the above bits is assigned to the corresponding Universal Port pin with the same number (e.g. PISA.0 to pin PA0). Bit PISn is set and an interrupt is generated on INT if

– the corresponding Universal Port pin Pn is defined as input via register PCR and
– the interrupt source is enabled by resetting the corresponding interrupt mask PIMn in register PIM and
– state transition has occurred on pin Pn. For definite detection of a real state transition, pulse width should not be shorter than 20 ns.

Note: Bits PISn are reset when register PIS is read. Masked interrupts are not normally indicated when PIS is read. Instead, they remain internally stored and pending. A pending interrupt is generated when the corresponding mask bit is reset to zero. However, if bit IPC.VIS is set to “1”, interrupt statuses in PIS may be flagged although they are masked via register PIM. These masked interrupt statuses neither generate an interrupt vector or a signal on INT, nor are visible in register GIS. If more than one consecutive state transitions occurs on the same pin before the PIS register is read, only one interrupt request will be generated.

Port Interrupt Mask Register Port A...D (WRITE)

Value after RESET: FF_H

	7						0	
PIMA	PIM7						PIM0	(03D/07D)
PIMB	PIM7						PIM0	(0BD/0FD)
PIMC	PIM7						PIM0	(13D/17D)
PIMD	0	0	0	0	PIM3		PIM0	(1BD/1FD)

Note: Unused bits have to be set to logical “0”.

Each PIM register is accessible via two channel addresses.

Each of the above bits is assigned to the corresponding Universal Port pin and to the bits of register PIS with the same number (e.g. PIMA.0 to pin PA0).
0…Interrupt source is enabled.
1…Interrupt source is disabled.
A “1” in a bit position of PIM sets the mask active for the interrupt status in PIS. Masked interrupt statuses neither generate an interrupt vector or a signal on INT, nor are they visible in register GIS.
Moreover, they will

– not be displayed in the Interrupt Status Register if bit IPC.VIS is set to “0”

– be displayed in the Interrupt Status Register if bit IPC.VIS is set to “1”.

Refer to description of register PIS.

Note: After RESET, all interrupt sources are **dis**abled.

Port Configuration Register Port A...D (READ/WRITE)

Value after RESET: FF_H

	7					0	
PCRA	PCR7					PCR0	(03E/07E)
PCRB	PCR7					PCR0	(0BE/0FE)
PCRC	PCR7					PCR0	(13E/17E)
PCRD	0	0	0	0	PCR3	PCR0	(1BE/1FE)

Note: Unused bits have to be set to logical “0”.

Each PCR register is accessible via two channel addresses.

Each of the above bits is assigned to the Universal Port pin (P0…P7) with the same number(e.g. PCRA.0 to pin PA0). If bit PCRn (*n* = 0…7) is set to
0…pin Pn is defined as output.
1…pin Pn is defined as input.

Note: After RESET, all pins of the Universal Port are defined as **in**puts.

4.3 Status/Control Registers in BISYNC Mode

4.3.1 Register Addresses

Table 11
Register Addresses in BISYNC Mode

Address (A8 … A0)								Register	
Channel									
0	1	2	3	4	5	6	7	Read	Write
000 … 01F	040 … 05F	080 … 09F	0C0 … 0DF	100 … 11F	140 … 15F	180 … 19F	1C0 … 1DF	RFIFO	XFIFO
020	060	0A0	0E0	120	160	1A0	1E0	STAR	CMDR
021	061	0A1	0E1	121	161	1A1	1E1	———	PRE
022	062	0A2	0E2	122	162	1A2	1E2	MODE	
023	063	0A3	0E3	123	163	1A3	1E3	TIMR	
024	064	0A4	0E4	124	164	1A4	1E4	SYNL	
025	065	0A5	0E5	125	165	1A5	1E5	SYNH	
026	066	0A6	0E6	126	166	1A6	1E6	TCR	
027	067	0A7	0E7	127	167	1A7	1E7	DAFO	
028	068	0A8	0E8	128	168	1A8	1E8	RFC	
029	069	0A9	0E9	129	169	1A9	1E9	———	———
02A	06A	0AA	0EA	12A	16A	1AA	1EA	RBCL	XBCL
02B	06B	0AB	0EB	12B	16B	1AB	1EB	RBCH	XBCH
02C	06C	0AC	0EC	12C	16C	1AC	1EC	CCR0	
02D	06D	0AD	0ED	12D	16D	1AD	1ED	CCR1	
02E	06E	0AE	0EE	12E	16E	1AE	1EE	CCR2	
02F	06F	0AF	0EF	12F	16F	1AF	1EF	CCR3	
030	070	0B0	0F0	130	170	1B0	1F0	———	TSAX
031	071	0B1	0F1	131	171	1B1	1F1	———	TSAR
032	072	0B2	0F2	132	172	1B2	1F2	———	XCCR
033	073	0B3	0F3	133	173	1B3	1F3	———	RCCR
034	074	0B4	0F4	134	174	1B4	1F4	VSTR	BGR
035	075	0B5	0F5	135	175	1B5	1F5	———	———
036	076	0B6	0F6	136	176	1B6	1F6	———	———
037	077	0B7	0F7	137	177	1B7	1F7	———	———
038, 078, 0B8, 0F8, 138, 178, 1B8, 1F8								GIS *)	IVA *)
039, 079, 0B9, 0F9, 139, 179, 1B9, 1F9								IPC *)	
03A	07A	0BA	0FA	13A	17A	1BA	1FA	ISR0	IMR0
03B	07B	0BB	0FB	13B	17B	1BB	1FB	ISR1	IMR1
03C, 07C		0BC, 0FC		13C, 17C		1BC, 1FC		PVRA…D	
03D, 07D		0BD, 0FD		13D, 17D		1BD, 1FD		PISA…D	PIMA…D
03E, 07E		0BE, 0FE		13E, 17E		1BE, 1FE		PCRA…D	
03F, 07F		0BF, 0FF		13F, 17F		1BF, 1FF		———	

*) All channel assigned addresses enable access to the same register(s)

Note: Read access to unused register addresses: value should be ignored,
Write access to unused register addresses: should be avoided, or set to “00”$_H$.

4.3.2 Register Definitions

Receive FIFO (READ) RFIFO (offset: 00…1F)

Received data stored in RFIFO (LSB is received first) can be organized in one of two selectable ways (**refer to figure 50**):

– pure data up to a character length of 8 bits (incl. optional parity)

– additionally, one status byte per character with information about parity (if enabled) and parity error.

Reading data from RFIFO can be done in 8-bit (byte) or 16-bit (word) access depending on the selected bus interface mode. The LSB is received first from the serial interface.

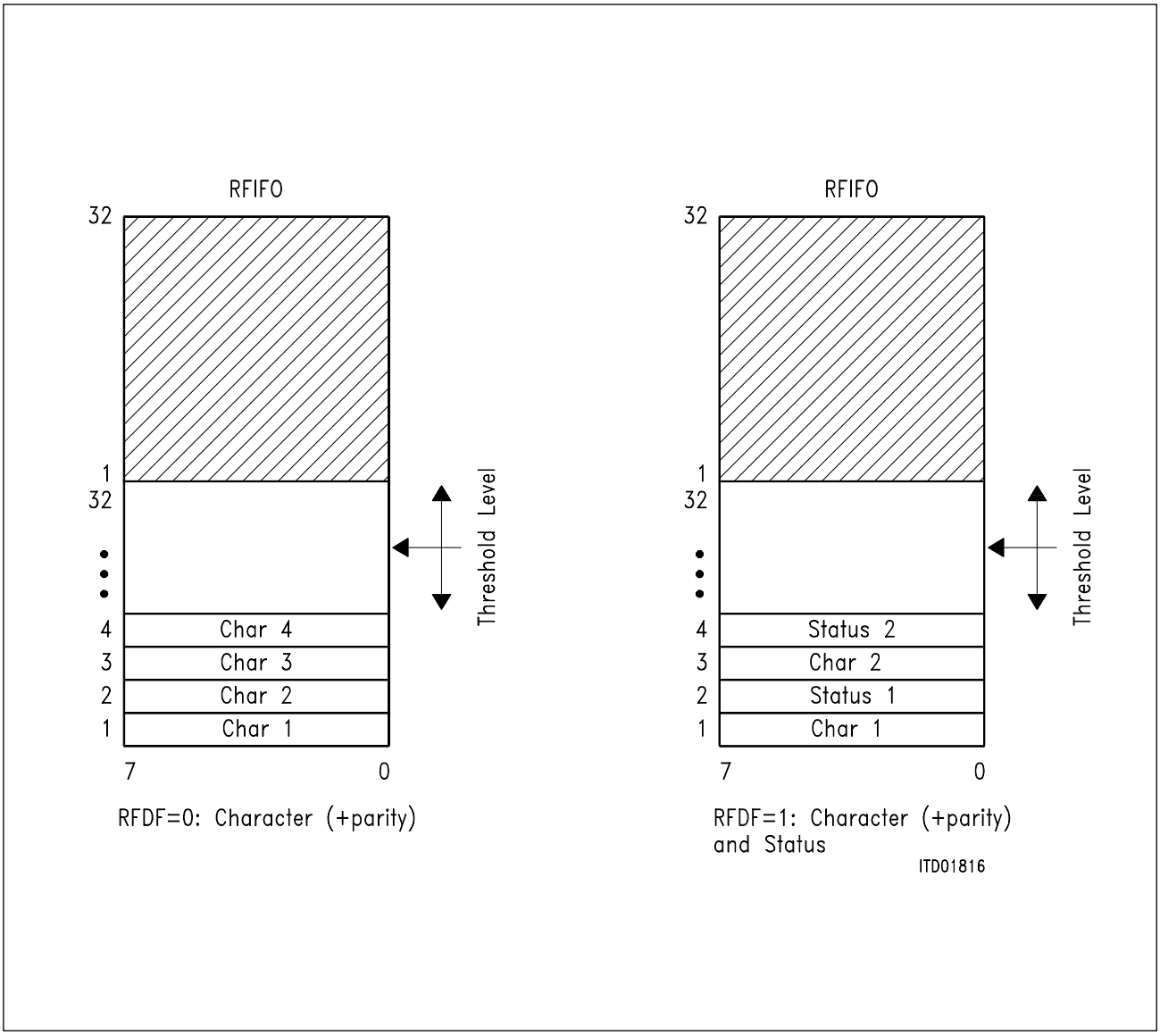


Figure 50
Organization of RFIFO

Interrupt Controlled Data Transfer (Interrupt Mode)

Selected if DMA bit in XBCH is set to “0”

Up to 32 bytes/16 words of received data can be read from the RFIFO following a RPF or a TCD interrupt depending on the selected RFIFO mode (refer to RFC register):

RPF interrupt: A fixed number of bytes/words (programmed threshold level RFTH0, 1) has to be read by the CPU.

TCD interrupt: Termination character detected. The received data stream is monitored for “termination character” (programmable via register TCR). The number of valid **bytes** in RFIFO is determined by reading the RBCL register.

If necessary, the CPU can access the RFIFO by issuing RFIFO Read command (CMDR.RFRD) before threshold level or the termination condition is reached. The number of valid **bytes** is determined by reading the RBCL register. Additional information: STAR.RFNE: RFIFO Not Empty.

DMA Controlled Data Transfer (DMA Mode)

Selected if DMA bit in XBCH is set.
If the RFIFO contains the number of bytes/words defined via the threshold level, the ESCC8 autonomously requests a DMA block data transfer by DMA by activating the DRRn line until the last valid data is read (the DDRn line remains active up to the beginning of the last read cycle).

This forces the DMA controller to continuously perform bus cycles till all data is transferred from the ESCC8 to the system memory (level triggered transfer mode of DMA controller). If the end condition (TCD) is reached, the same procedure as above is performed. DRRn is activated until the termination character is transferred. A TCD interrupt is issued after the last data has been transferred. Generation of further DMA requests is blocked until TCD interrupts has been acknowledged by issuing an RMC command. The valid **byte** count of the last block can be determined by reading the RBCL register following the TCD interrupt.

Note: Addresses within the 32-byte address space of the FIFO all point to the same byte/word, i.e. current data can be accessed with any address within the valid scope.

Transmit FIFO (WRITE) XFIFO (offset: 00…1F)

Writing data to XFIFO can be in 8-bit (byte) or 16-bit (word) access depending on the selected bus interface mode. The LSB is transmitted first.

Interrupt Mode

Selected if DMA bit in XBCH is reset.
Up to 32 bytes/16 words of transmit data can be written to the XFIFO following an XPR interrupt.

DMA Mode

Selected if DMA bit in XBCH is set.
Prior to any data transfer, the actual **byte** count to be transmitted must be written to the XBCH, XBCL registers by the user. Correct transmission of data in the case of word access and of an odd number of bytes specified in XBCH, XBCL is guaranteed.

If data transfer is then initiated via the CMDR register (command XF), the ESCC8 autonomously requests the correct amount of block data transfers (n×BW + REST; BW = 32, 16; n = 0, 1,…).

Note: Addresses within the 32-byte address space of the FIFO all point to the same byte/word, i.e. current data can be accessed with any address within the valid range.

Status Register (READ)

	7							0	
STAR	XDOV	XFW	RFNE	SYNC	0	CEC	CTS	0	(offset: 20)

XDOV… Transmit Data Overflow

More than 32 bytes have been written to the XFIFO.

This bit is reset by:

– a transmitter reset command XRES
– or when all bytes in the accessible half of the XFIFO have been moved into the inacessible half.

XFW… Transmit FIFO Write Enable

Data can be written to the XFIFO.

RFNE… RFIFO Not Empty

This bit is set if the accessible part of RFIFO holds at least one valid byte.

SYNC... Synchronization Status

The bit is reset after the HUNT command has been issued. It indicates that the receiver has lost synchronization and is searching for the presence of a SYN character. If found, SYNC will be immediately set, the SCD interrupt is generated (if enabled), and filling the RFIFO with received data is started.

CEC… **Command Executing**

0… no command is currently executed, the CMDR register can be written to.

1… a command (written previously to CMDR) is currently executed, no further command can be temporarily written in CMDR register.

Note: CEC will be active at most 2.5 transmit clock periods. If the ESCC8 is in power down mode CEC will stay active.

CTS… **Clear To Send State**

This bit indicates the state of the $\overline{\text{CTS}}$ pin.

0… $\overline{\text{CTS}}$ is inactive (high)

1… $\overline{\text{CTS}}$ is active (low)

Command Register (WRITE)

Value after RESET: 00_H

	7							0	
CMDR	RMC	RRES	RFRD	STI	XF	HUNT	XME	XRES	(offset: 20)

Note: The maximum time between writing to the CMDR register and the execution of the command is 2.5 clock cycles. Therefore, if the CPU operates with a very high clock rate in comparison with the ESCC8's clock, it is recommended that the CEC bit of the STAR register be checked before writing to the CMDR register to avoid any loss of commands.

RMC… **Receive Message Complete**

Confirmation from CPU to ESCC8 that the current data block has been fetched following a RPF or TCD interrupt or following a user initiated read access in conjunction with the RFIFO Read command RFRD; the occupied space in the RFIFO can be released.

Note: In DMA Mode, this command has to be issued after a TCD interrupt in order to enable the generation of further receiver DMA requests.

RRES… **Receiver Reset**

All data in RFIFO and BISYNC receiver is deleted.

RFRD… Receive FIFO Read Enable

The CPU can have access to RFIFO by issuing the RFRD command before threshold level or the end condition (TCD) are fulfilled. After issuing the RFRD command, the CPU has to wait for TCD interrupt, before reading RBC and RFIFO. The number of valid bytes is determined by reading the RBCL register.

STI… Start Timer

The internal timer is started.

Note: The timer is stopped by rewriting the TIMR register after start.

XF… Transmit Frame

- Interrupt Mode
  After having written up to 32 bytes/16 words to the XFIFO, this command initiates the transmission of data.
- DMA Mode
  After having written the amount of data to be transmitted to the XBCH, XBCL registers, this command initiates the data transfer from system memory to ESCC8 by DMA. Serial data transmission starts as soon as 32 bytes/16 words are stored in the XFIFO or the Transmit Byte Counter value is reached.

HUNT... Enter Hunt Phase

This command forces the receiver to immediately go into the Hunt state. Synchronization is lost and the receiver starts searching for SYN characters.

XME... Transmit Message End (used in interrupt mode only!)

Indicates that the data block written last to the transmit FIFO completes the current frame. The ESCC8 can terminate the transmission operation properly by appending the CRC sequence to the data. After that, IDLE is transmitted.
In DMA Mode, the end of the frame is determined by the Transmit Byte Count in XBCH, XBCL, thus, XME is not used in this case.

XRES… Transmitter Reset

XFIFO is cleared of any data and IDLE (logical “1s”) is transmitted. This command can be used by the CPU to abort current data transmission. In response to XRES an XPR interrupt is generated.

Preamble Register (WRITE)

Value after RESET: 00_H

	7							0	
PRE	PR7							PR0	(offset: 21)

This register defines the 8-bit pattern which is sent out during preamble transmission (refer to register CCR3).

Mode Register (READ/WRITE)

Value after RESET: 00_H

	7							0	
MODE	0	0	SLEN	BISNC	RAC	RTS	TRS	TLP	(offset: 22)

Note: Unused bits have to be set to logical “0”.

SLEN... SYN Character Length

This bit selects the length of the SYN character:
0...6 bit (MONOSYNC) / 12 bit (BISYNC)
1...8 bit (MONOSYNC) / 16 bit (BISYNC)

BISNC... Enable Bisync Mode

0...MONOSYNC mode is enabled (6/8 bit SYN character defined via register SYNL).
1...BISYNC mode is enabled (12/16 bit SYN character defined via registers SYNL and SYNH). SYNL is received/transmitted first.

RAC… Receiver Active

Switches the receiver to operational or inoperational state.

0… receiver inactive

1… receiver active

RTS… Request To Send

Defines the state and control of $\overline{\text{RTS}}$ pin.

0… The $\overline{\text{RTS}}$ pin is controlled by the ESCC8 autonomously.
$\overline{\text{RTS}}$ is activated when data transmission starts and deactivated when transmission is completed.

1… The $\overline{\text{RTS}}$ pin is controlled by the CPU.
If this bit is set, the $\overline{\text{RTS}}$ pin is activated immediately and remains active till this bit is reset.

TRS… **Timer Resolution**

Selects the resolution of the internal timer (factor k, see description of TIMR register):
0…k = 32 768
1…k = 512

TLP… **Test Loop**

Input and output of the BISYNC channels are internally connected.
(e.g. transmitter channel 0 - receiver channel 0)

Timer Register (READ/WRITE)

	7		0	
TIMR	CNT	VALUE		(offset: 23)

VALUE… (5 bits) sets the time period t1 as follows:

$t_1 = k \times (\text{VALUE} + 1) \times \text{TCP}$

where

– k is the timer resolution factor which is either 32 768 or 512 clock cycles dependent on the programming of TRS bit in MODE.

– TCP is the clock period of transmit data.

CNT… (3 bits)

CNT plus VALUE determine the time period t_2 after which a timer interrupt will be generated. The time period t_2 is

$t_2 = 32 \times k \times \text{CNT} \times \text{TCP} + t_1$.

If CNT is set to 7, a timer interrupt is periodically generated after the expiration of t_1.

SYN Character Register Low, High (READ/WRITE)

Value after RESET: 00, 00_H

	7	0	
SYNL	SYNL		(offset: 24)
SYNH	SYNH		(offset: 25)

In conjunction with bit BISNC and bit SLEN the SYN character can be specified:

– **MONOSYNC** mode (BISNC = 0)

The SYN character is defined by SYNL.
SLEN = 0: the SYN character is specified by bits 0-5
SLEN = 1: the SYN character is specified by bits 0-7

– **BISYNC** mode (BISNC = 1)

The SYN character is defined by SYNL (low byte) and SYNH (high byte).
SLEN = 0: the 12-bit SYN character is specified by bits 0-5 of both SYNL and SYNH
SLEN = 1: the 16-bit SYN character is specified by bits 0-7 of both SYNL and SYNH
SYNL is received/transmitted first

In transmit direction, the SYN character thus specified is sent continuously when no data are to be transmitted and ITF (Interframe Time Fill) control bit is set to “1”.
In receive direction, the receiver searches for the specified SYN character in the receive data stream, when in hunt mode.

Termination Character Register (READ/WRITE)

Value after RESET: 00_H

	7	0	
TCR	TCR7	TCR0	(offset: 26)

TCR7–TCR0… Termination Character

If enabled via register RFC the received data stream is monitored for the occurrence of a programmed “termination character”. When such a character is found, an interrupt is issued if enabled via mask register IMR0. The number of valid **bytes** in the RFIFO up to and including the termination character is determined by reading the RBCL register.

Note: If selected character length is less than eight bits, leading (unused) bits of TCR have to be set to “0”.

Data Format (READ/WRITE)

Value after RESET: 00_H

	7						0	
DAFO	0	0	0	PAR1 PAR0	PARE	CHL1 CHL0		(offset: 27)

Note: Unused bits have to be set to logical “0”.

PAR1, PAR0… Parity Format

If parity check/generation is enabled by setting PARE, these bits define the parity type:

00… SPACE (“0”)
01… odd parity
10… even parity
11… MARK (“1”)

The received parity bit is stored in RFIFO

– as leading bit immediately preceding the character if character length is 5 to 7 bits and RFC.DPS is set to 0, and as LSB of the status byte pertaining to the character if the corresponding RFIFO data format is enabled.

– as LSB of the status byte pertaining to the character if character length is 8 bits and the corresponding RFIFO data format is enabled.

Parity error is indicated in the MSB of the status byte pertaining to the character, if enabled. Additionally, a parity error interrupt can be generated.

PARE… Parity Enable

0… parity check/generation disabled

1… parity check/generation enabled

CHL1–CHL0… Character Length

These bits define the length of received and transmitted characters, excluding optional parity:

00… 8 bit
01… 7 bit
10… 6 bit
11… 5 bit

RFIFO Control Register (READ/WRITE)

Value after RESET: 00H

	7						0	
RFC	0	DPS	SLOAD	RFDF	RFTH1 RFTH0	0	TCDE	(offset: 28)

Note: Unused bits have to be set to logical “0”.

DPS… Disable Parity Storage

Only valid if parity check/generation is enabled via DAFO.PARE and character length is less than 8 bits.

0… The parity bit is stored
1… The parity bit is **not** stored
in the data byte of RFIFO.

Note: The parity bit is always stored in the status byte.

SLOAD... Enable SYN Character Load

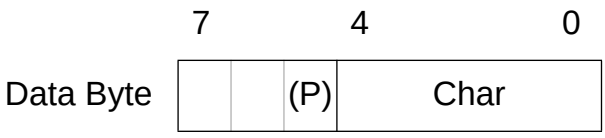
0...All data **except** SYN characters are stored in RFIFO.
1...Storage of all received SYN characters to RFIFO is enabled.

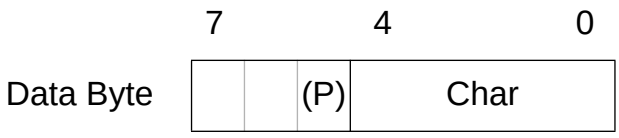
RFDF… **RFIFO Data Format**

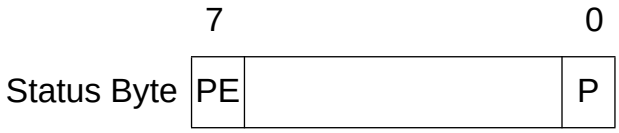
0… only data bytes (character plus optional parity up to 8 bit) are stored.

1… additionally to every data byte, an attached status byte is stored.

RFDF = 0	**RFDF = 1**
– character 5 – 8 bit or – character 5 – 7(8)* bit + parity * : parity bit is lost	– character 5 – 8 bit + status or – character 5 – 7(8)* bit + parity + status * : parity bit is in status byte







PE : parity error P : parity bit (P): can be disabled via bit DPS

RFTH1, RFTH0… RFIFO Threshold Level

These bits define the level up to which RFIFO is filled with valid data:

RFTH1, 0	**Threshold Level (bytes)**	
	RFDF = 0	**RFDF = 1**
00	1 (1d)	**2** (**1d** + 1s)
01	4 (4d)	4 (2d + 2s)
10	16 (16d)	16 (8d + 8s)
11	32 (32d)	32 (16d + 16s)

d: data byte s: status byte

If the threshold level is reached, the RPF interrupt is generated if enabled. After RPF is generated, the contents of RFIFO (RFTH bytes) can be read

by the CPU. To indicate that this RFIFO pool can be released, an RMC command has to be issued.

TCDE… Termination Character Detection Enable

When this bit is set, the received data stream is monitored for “termination character” (TCR register). When such a character occurs, the TCD interrupt is generated if enabled via mask register IMR0. The number of **bytes** to be read from RFIFO is determined by the value of RBCL.

Receive Byte Count Low (READ)

	7		0	
RBCL	RBC7		RBC0	(offset: 2A)

Indicates the number of valid bytes available in the accessible part of the RFIFO. This register must be read by the CPU following a TCD interrupt. In case of a TCD interrupt the number of valid bytes in the accessible part of the RFIFO can be evaluated by “AND”-ing the contents of RBCL with: threshold level (bytes) – 1.

Threshold Level	Mask
4	03_H
16	$0F_H$
32	$1F_H$

RBC is reset with RMC after preceeding TCD interrupt.

In case of RPF interrupt RBC is incremented by “threshold level (bytes)”.

Transmit Byte Count Low (WRITE)

	7		0	
XBCL	XBC7		XBC0	(offset: 2A)

Together with XBCH (bits XBC11…XBC8) this register is used in DMA Mode only, to program the length (1…4096 bytes) of the next data block to be transmitted.

In terms of the value xbc, programmed in XBC11…XBC0 (xbc = 0…4095), the length of the block in number of bytes is:

length = xbc + 1.

This allows the ESCC8 to request the correct amount of DMA cycles after an XF command in CMDR.

Received Byte Count High (READ)

Value after RESET: 000xxxxx

	7					0	
RBCH	DMA	0	CAS	0	RBC11	RBC8	(offset: 2B)
	see XBCH						

DMA, CAS… These bits represent the read-back value programmed in XBCH

RBC11– RBC8… Receive Byte Count (most significant bits)

No function.

Transmit Byte Count High (WRITE)

Value after RESET: 000xxxxx

	7					0	
XBCH	DMA	0	CAS	XC	XBC11	XBC8	(offset: 2B)

Note: Unused bits have to be set to logical “0”.

DMA… DMA Mode

Selects the data transfer mode of ESCC8 to/from System Memory.

0… Interrupt controlled data transfer (Interrupt Mode).

1… DMA controlled data transfer (DMA Mode).

CAS… Carrier Detect Auto Start

When set, a high on the CD pin enables the corresponding receiver and data reception is started. When not set, if not in Clock Mode 1 or 5, the CD pin can be used as a general input.

XC… Transmit Continuously

Only valid if DMA Mode is selected.

If the XC bit is set, the ESCC8 continuously requests for transmit data ignoring the transmit byte count programmed via XBCH, XBCL.

XBC11– XBC8… Transmit Byte Count (most significant bits)

Valid only if DMA Mode is selected.

Together with XBCL (bits XBC7…XBC0), determine the number of characters to be transmitted.

Channel Configuration Register 0 (READ/WRITE)

Value after RESET: 00_H

	7							0	
CCR0	PU	MCE	0	SC2	SC1	SC0	SM1	SM0	(offset: 2C)

Note: Unused bits have to be set to logical “0”.

PU… Switches between power up and power down mode

0… power down (standby)

1… power up (active)

MCE… Master Clock Enable

If this bit is set to “1”, the clock provided via pin XTAL1 works as master clock to allow full functionality of the microprocessor interface (access to all status and control registers, DMA and interrupt support) independent of the receive and the transmit clocks. The internal oscillator in conjunction with a crystal on XTAL1-2 can be used, too. The master clock option is not applicable in clock mode 5. **Refer to table 5** for more details.

Note: The internal timers run with the master clock.

SC2– SC0… Serial Port Configuration

000… NRZ data encoding
001… (not recommended)
010… NRZI data encoding
011… (not recommended)
100… FM0 data encoding
101… FM1 data encoding
110… MANCHESTER data encoding
111… (not used)

SM1– SM0… Serial Mode

00… HDLC/SDLC mode
01… SDLC Loop mode
10… BISYNC mode
11… ASYNC mode

Channel Configuration Register 1 (READ/WRITE)

Value after RESET: 00_H

	7						0	
CCR1	0	0	0	ODS	ITF	CM2	CM0	(offset: 2D)

Note: Unused bits have to be set to logical “0”.

ODS… Output Driver Select

Defines the function of the transmit data pins (T×DA, T×DB)

0… T×D pin is an open drain output.
1… T×D pin is a push-pull output.

ITF... Interframe Time Fill Format

Determines the idle (= no data to send) state of the transmit data pin (T×D)
0...Continuous logical “1” is output.
1...Continuous SYN characters are output.

CM2– CM0… Clock Mode

Selects one of 8 different clock modes:

000 clock mode 0
• •
• •
• •
111 clock mode 7

Note: Clock mode 5 is only specified for version SAB 82538H-10, not for SAB 82538H.

Channel Configuration Register 2 (READ/WRITE)

Value after RESET: 00_H

The meaning of the individual bits in CCR2 depends on the clock mode selected via CCR1 as follows:

CCR2 clock mode 0a, 1

7							0	
SOC1	SOC0	0	0	0	RWX	0	DIV	(offset: 2E)

clock mode 0b, 2, 3, 6, 7

BR9	BR8	BDF	SSEL	TOE	RWX	0	DIV

clock mode 4

SOC1	SOC0	0	0	TOE	RWX	0	DIV

clock mode 5

SOC1	SOC0	XCS0	RCS0	TOE	RWX	0	DIV

Note: Unused bits have to be set to logical “0”.

SOC1, SOC0… Special Output Control

In a bus configuration (selected via CCR0) defines the function of pin $\overline{RTS}$ as follows:

0X… $\overline{RTS}$ output is activated during transmission of characters.
10… $\overline{RTS}$ output is always high ($\overline{RTS}$ disabled).
11… $\overline{RTS}$ indicates the reception of a data frame (active low).

In a point-to-point configuration (selected via CCR0) the T×D and R×D pins may be flipped

0X… data is transmitted on T×D, received on R×D (normal case).
1X… data is transmitted on R×D, received on T×D.

BR9, BR8… Baud Rate, Bit 9-8

High order bits, see description of BGR register.

XCS0, RCS0... Transmit/Receive Clock Shift, Bit 0

Together with XCS2, XCS1 (RCS2, RCS1) in TSAX (TSAR), determines the clock shift relative to the frame synchronization signal of the Transmit (Receive) time-slot.
A clock shift of 0 ... 7 bits is programmable (clock mode 5 only).

BDF... Baud Rate Division Factor

0…The division factor of the baud rate generator is set to 1 (constant).
1…The division factor is determined by BR9 - BR0 bits in CCR2 and BRG registers.

SSEL... Clock Source Select

Selects the clock source in clock modes 0, 2, 3, 6 and 7 (**refer to table 5**).

TOE... T×CLK Output Enable

0… T×CLK pin is input

1… T×CLK pin is switched to output function if applicable to the selected clock mode (**refer to table 5**).

RWX... Read/Write Exchange

Valid only in DMA mode. If this bit is set, the
– $\overline{RD}$ and $\overline{WR}$ pins are internally exchanged (Siemens/Intel bus interface)
– R/$\overline{W}$ pin is inverted in polarity (Motorola bus interface)
while any $\overline{DACK}$ input is active. This useful feature allows a simple interfacing to the DMA controller.

Note: The RWX bit of all eight channels is “or”ed.

DIV... Data Inversion

Only valid if NRZ data encoding is selected. Data is transmitted and received inverted.

Channel Configuration Register 3 (READ/WRITE)

(Version 2 upwards)

Value after RESET: 00_H

	7							0	
CCR3	PRE1	PRE0	EPT	CON	CRL	CAPP	CRCM	PSD	(offset: 2F)

PRE1...PRE0... Number of Preamble Repetition

If Preamble transmission is enabled, the Preamble defined via register PRE is transmitted
00...1 times
01...2 times
10...4 times
11...8 times.

EPT... **Enable Preamble Transmission**

This bit enables transmission of a preamble. The preamble is started after Interframe Time Fill transmission has been stopped and a new block of data is about to be transmitted. The preamble consists of an 8-bit pattern defined via register PRE which is repeated a number of times selected by bits PRE0 and PRE1.

CON... **CRC ON**

This bit determines whether the current data written to XFIFO has to be included into CRC calculation or not. It has to be programmed **before** the assigned byte/word is written to XFIFO. In the case of word access, **both** characters are included. Since this control bit is copied in the XFIFO every time a character is written, it is not necessary to reprogram it for each character when consecutive characters are to be either all included into or all excluded from CRC calculation.
0...data not included
1...data included.

CRL... **CRC Reset Level**

This bit defines the initialization for internal transmit CRC generator.
0...Initialized to "$FFFF_H$".
1...Initialized to "0000_H".

Note: The internal transmit CRC generator is automatically initialized before transmission of a new frame starts.

CAPP... **CRC Append**

If this bit is set, the internal transmit CRC generator is activated:
1. The CRC generator is initialized every time the transmission of a new frame starts. Initialization value is defined via bit CRL.
2. During transmission all data with the CON bit set to "1" are included into CRC checksum calculation.
3. The checksum is automatically appended to the last transmitted data of the frame if a Transmit Message End command (XME) has been issued.

CRCM... **Select CRC Algorithm**

Selects the CRC algorithm for the internal transmit CRC generator:
0...CRC-16 $(X^{16} + X^{15} + X^2 + 1)$
1...CRC-CCITT $(X^{16} + X^{12} + X^5 + 1)$

PSD... **DPLL Phase Shift Disable**

Only applicable in the case of NRZ and NRZI encoding.
If this bit is set to "1", the Phase Shift function of the DPLL is disabled. In this case the windows for Phase Adjustment are extended.

Time-Slot Assignment Register Transmit (WRITE)

This register is only used in clock mode 5!

Value after RESET: 00_H

	7			0	
TSAX	TSNX		XCS2	XCS1	(offset: 30)

TSNX… Time-slot Number Transmit

Selects one of up 64 possible time-slots (00_H-$3F_H$) in which data is transmitted. The number of bits per time-slot can be programmed via XCCR.

XCS2… XCS1 … Transmit Clock Shift, Bit 2-1

Together with bit XCS0 in CCR2, transmit clock shift can be adjusted.

Time-Slot Assignment Register Receive (WRITE)

This register is only used in clock mode 5!

Value after RESET: 00_H

	7			0	
TSAR	TSNR		RCS2	RCS1	(offset: 31)

TSNR… Time-slot Number Receive

Defines one of up to 64 possible time-slots (00_H-$3F_H$) in which data is received. The number of bits per time-slot can be programmed via RCCR.

RCS2– RCS1… Receive Clock Shift, Bit 2-1

Together with bit RCS0 in CCR2, the receive clock shift can be adjusted.

Transmit Channel Capacity Register (WRITE)

This register is only used in clock mode 5!
Value after RESET: 00_H

	7						0	
XCCR	XBC7						XBC0	(offset: 32)

XBC7– XBC0… Transmit Bit Number Count, Bit 7-0

Defines the number of bits to be transmitted within a time-slot:
Number of bits = XBC + 1 (1 … 256 bits/time-slot).

Receive Channel Capacity Register (WRITE)

This register is only used in clock mode 5!
Value after RESET: 00_H

	7						0	
RCCR	RBC7						RBC0	(offset: 33)

RBC7– RBC0… Receive Bit Count, Bit 7-0

Defines the number of bits to be received within a time-slot:
Number of bits = RBC + 1 (1…256 bits/time-slot).

Version Status Register (READ)

	7						0	
VSTR	CD	DPLA	0	0	VN3		VN0	(offset: 34)

CD… Carrier Detect

This bit reflects the state of the CD pin.

1… CD active
0… CD inactive

DPLA… DPLL Asynchronous

This bit is only valid when the receive clock is supplied by the DPLL and FM0, FM1 or Manchester data encoding is selected.

It is set when the DPLL has lost synchronization. Reception is disabled (IDLE is inserted) until synchronization has been regained. Additionally, transmission is interrupted, too, if the transmit clock is derived from the DPLL (same effect as the deactivation of pin $\overline{\text{CTS}}$).

VN3– VN0… Version Number of Chip

0…Version 1
1… Version 2

Baud Rate Generator Register (WRITE)

	7							0	
BGR	BR7							BR0	(offset: 34)

BR7– BR0… Baud Rate, bits 7-0

Together with bits BR9, BR8 of CCR2, it determines the division factor of the baud rate generator.

In terms of the value N programmed in BR9 - BR0 (N = 0…1023), the division factor k is:

$k = (N + 1) \times 2$

Global Interrupt Status Register (READ)

Value after RESET: 00_H

	7							0	
GIS	PIA	PIB	PIC	PID	CII	CN2	CN1	CN0	(038/078/0B8/0F8) (138/178/1B8/1F8)

This status register points to pending

– channel assigned interrupts (ISR0_x, ISR1_x)

– universal port interrupts (PISA…D).

GIS is accessible via eight channel addresses (038_H to $1F8_H$).

PIA– PID… Port Interrupt Indication

These status bits point to pending interrupts in corresponding Port Interrupt Status registers PISA…PISD. They may be set independently from channel assigned interrupts.

CII… Channel Interrupt Indication

Set if at least one interrupt source of any channel is active.

CN2– CN0… Channel Number (0…7)

If at least one interrupt source is active (bit CII is set), these bits point to the channel with currently highest source priority. **Refer to chapter 2.2.3** for detailed description of the priority structure.

Contents of register GIS are frozen after every input acknowledge cycle.

– after the first read access to GIS after the interrupt vector has been output,

– after every read access to anyone of the channel assigned interrupt status registers,

– during every $\overline{\text{INTA}}$ cycle.

Interrupt Vector Address (WRITE)

Value after RESET: 00_H

	7							0	
IVA	T7	T6	T5	T4	T3	T2	ROT	EDA	(038/078/0B8/0F8) (138/178/1B8/1F8)

Note: Unused bits have to be set to logical “0”.

IVA is accessible via eight channel addresses (38_H to $1F8_H$).

Version 2 upward provides dynamic adjustment of channel priorities by programming the “highest priority channel”. Selection of the “highest priority channel” is done with every **write access** to IVA in conjunction with the channel assigned IVA register address:

IVA Register Address:	Highest Priority Channel
38_H	0
78_H	1
$B8_H$	2
$F8_H$	3
138_H	4
178_H	5
$1B8_H$	6
$1F8_H$	7

The priority level becomes valid with the end of the write access to the IVA register (rising edge of $\overline{\text{WR}}$ or $\overline{\text{DS}}$, whichever applies) and remains stable until a new write access to this register occurs.

T7– T6… Device Address

These bits define the value of bits 6 and 7 of the interrupt vector which is sent out on the data bus (D0…D7) during the interrupt acknowledge cycle.

T5… Device Address

Version 1: Device Address

This bit defines the value of bit 5 of the interrupt vector which is sent out on the data bus (D0…D7) during the interrupt acknowledge cycle.

Version 2: Device Address Extension

In Interrupt vector mode 2 (bit EDA set) this bit defines the value of bit 5 of the interrupt vector which is sent out on the data bus (D0…D7) during the interrupt acknowledge cycle.

T4– T2… Device Address Extension

In Interrupt vector mode 2 (bit EDA set) these bits define the value of bits 2 to 4 of the interrupt vector which is sent out on the data bus (D0…D7) during the interrupt acknowledge cycle.

ROT… Rotating Interrupt Priority (version 2 upward)

Version 1:

This bit is unused and has to be set to logical “0”.

Version 2:

0 … Fixed Interrupt Priority
The relative order of the interrupt priority level assigned to the channels is fixed (**refer to chapter 2.3.1**).

1 … Rotating Interrupt Priority
The interrupt priority level will be adjusted after an interrupt has been serviced. Together with bit IPC.ROTM the interrupt priority mode is selected.

IPC.ROTM = 0: The priority level of all 8 serial channels are adjusted.

IPC.ROTM = 1: The priority level of only 7 channels are adjusted while one channel is fixed.

EDA… Extended Device Address

If set, bits 2 to 5 (version 1: bits 2 to 4) of the generated interrupt vector contain the Device Address Extension T2…T5 (version 1: T2…T4) instead of the channel number. For detailed information **refer to chapter 2.2.3**.

Interrupt Port Configuration (READ/WRITE)

Value after RESET: 00_H

	7							0	
IPC	VIS	ROTM	SLA2	SLA1	SLA0	CASM	IC1	IC0	(039/079/0B9/0F9) (139/179/1B9/1F9)

Note: Unused bits have to be set to logical “0”.

IPC is accessible via eight channel addresses (039_H to $1F9_H$).

VIS… Masked Interrupts Visible (version 2 upward)

0… Masked interrupt status bits are not visible.
1… Masked interrupt status bits are visible.

ROTM… Rotating Interrupt Priority Mode (version 2 upward)

Together with bit IVA.ROT the interrupt priority mode is selected.

0 … With IVA.ROT = 1 the priorities of all 8 serial channels are rotated cyclically after an interrupt has been serviced. The channel last serviced is assigned the lowest priority of all (**refer to chapter 2.2.3.1**).

1 … With IVA.ROT = 0 the priority adjustment is performed only on 7 channels while one channel is fixed to highest priority level (**refer to chapter 2.2.3.1**).

SLA2… SLA0… Slave Address

Only used in Slave Cascading Mode (refer to CASM).

CASM… Cascading Mode

0… Slave Cascading Mode
Pins IE0, IE1 and IE2 are used as inputs. Interrupt acknowledge is accepted if an interrupt signal has been generated and the values on pins IE0, IE1and IE2 correspond to the programmed values in SLA0, SLA1 and SLA2 (slave address).

1…Daisy Chaining Mode
Pin IE0 as Interrupt Enable Output and pin IE1 as Interrupt Enable Input are used for building a Daisy Chain. Pin IE2 is not used. Interrupt acknowledge is accepted if an interrupt signal has been generated and Interrupt Enable Input IE1 is active during a subsequent INTA cycle(s). If pin INT goes active, Interrupt Enable Output IE0 is immediately set to “low”.

IC1– IC0… Interrupt Port Configuration

These bits define the function of the interrupt output stage (pin INT):

IOC1	IOC0	Function
X	0	Open drain output
0	1	Push/pull output, active low
1	1	Push/pull output, active high

Interrupt Status Register 0 (READ)

Value after RESET: 00_H

	7							0	
ISR0	TCD	0	PERR	SCD	PLLA	CDSC	RFO	RPF	(offset: 3A)

All bits are reset when ISR0 is read. Additionally, TCD and RPF are reset when the corresponding interrupt vector is output.

Note: If bit IPC.VIS is set to “1”, interrupt statuses in ISR0 may be flagged although they are masked via register IMR0. However, these masked interrupt statuses neither generate an interrupt vector or a signal on INT, nor are visible in register GIS.

TCD… Termination Character Detected

The termination character (TCR) has been received or the execution of the RFRD command issued before has been completed. A data block is now available in the RFIFO. The actual block length can be determined by reading register RBCL first.

PERR… Parity Error

Only valid if parity check/generation is enabled.

If set, a character with parity error has been received. If enabled via RFDF, parity error information is stored in RFIFO in the status byte pertaining to that character.

SCD... SYN Character Detected

Only valid in Hunt Mode.
This bit is set if a SYN character is found in the received data stream after the HUNT command has been issued. The receiver now is in the synchronous state.

PLLA… **DPLL Asynchronous**

This bit is only valid when the receive clock is supplied by the DPLL and FM0, FM1 or Manchester data encoding is selected.

It is set when the DPLL has lost synchronization. Reception is disabled (IDLE is inserted) until synchronization has been regained. Additionally, transmission is also interrupted if the transmit clock is derived from the DPLL.

CDSC… **Carrier Detect Status Change**

Indicates that a state transition has occurred on CD. The actual state of CD can be read from the VSTR register.

RFO… **Receive FIFO Overflow**

This interrupt is generated if RFIFO is full and a further character is received. This interrupt can be used for statistical purposes and indicates that the CPU does not respond quickly enough to an RPF or TCD interrupt.

RPF… **Receive Pool Full**

This bit is set if RFIFO is filled with data (character and optional status information) up to the programmed threshold level.

Note: This interrupt is only generated in Interrupt Mode.

Interrupt Status Register 1 (READ)

Value after RESET: 00_H

	7							0	
ISR1	0	0	ALLS	XDU	TIN	CSC	XMR	XPR	(offset: 3B)

All bits are reset when ISR1 is read. Additionally, XPR is reset when the corresponding interrupt vector is output.

Note: If bit IPC.VIS is set to “1”, interrupt statuses in ISR1 may be flagged although they are masked via register IMR1. However, these masked interrupt statuses neither generate an interrupt vector or a signal on INT, nor are visible in register GIS.

ALLS... **All Sent**

This bit is set when the XFIFO is empty and the last character is completely sent out on T×D.

XDU... Transmit Data Underrun

A block of data in transmission has been terminated with IDLE, because the XFIFO contains no further data.
Note: Transmitter and XFIFO are reset and deactivated if this condition occurs. They are re-activated not before this interrupt status register has been read. Thus, XDU should not be masked via register IMR1.

TIN... Timer Interrupt

The internal timer has expired (see also description of TIMR register).

CSC... Clear To Send Status Change

Indicates that a state transition has occurred on $\overline{CTS}$. The actual state of $\overline{CTS}$ can be read from STAR register (CTS bit).

XMR... Transmit Message Repeat

The transmission of the last block of characters has to be repeated because
- a collision has occurred when transmitting a character in a bus configuration, or
- $\overline{CTS}$ (transmission enable) has been withdrawn during transmission of a character in point-to-point configuration.

XPR... Transmit Pool Ready

A data block of up to 32 bytes can be written to XFIFO.

Interrupt Mask Register 0, 1 (WRITE)

Value after RESET: FF_H, FF_H

	7							0	
IMR0	TCD	1	PERR	SCD	PLLA	CDSC	RFO	RPF	(offset: 3A)
IMR1	1	1	ALLS	XDU	TIN	CSC	XMR	XPR	(offset: 3B)

Note: Unused bits have to be set to logical “1”.

Each interrupt source can generate an interrupt signal at port INT (function of the output stage is defined via register IPC). A “1” in a bit position of IMR0 or IMR1 sets the mask active for the interrupt status in ISR0 or ISR1. Masked interrupt statuses neither generate an interrupt vector or a signal on INT, nor are they visible in register GIS. Moreover, they will
– not be displayed in the Interrupt Status Register if bit IPC.VIS is set to “0”
– be displayed in the Interrupt Status Register if bit IPC.VIS is set to “1”

Note: After RESET, all interrupts are **dis**abled.

Port Value Register Port A...D (READ/WRITE)

	7					0	
PVRA	PVR7					PVR0	(03C/07C)
PVRB	PVR7					PVR0	(0BC/0FC)
PVRC	PVR7					PVR0	(13C/17C)
PVRD	0	0	0	0	PVR3	PVR0	(1BC/1FC)

Note: Unused bits have to be set to logical “0”.

Each PVR register is accessible via two channel addresses.

Each of the above bits is assigned to the corresponding Universal Port pin with the same number (e.g. PVRA.0 to pin PA0).

Read Access

PVR shows the value of all pins (input and output). Input values can be separated via software by “AND”-ing PCR and PVR.

Write Access

PVR accepts values for all output pins (defined via PCR). Values written to input pin locations are ignored.

Port Interrupt Status Register Port A...D (READ)

	7						0	
PISA	PIS7						PIS0	(03D/07D)
PISB	PIS7						PIS0	(0BD/0FD)
PISC	PIS7						PIS0	(13D/17D)
PISD	0	0	0	0	PIS3		PIS0	(1BD/1FD)

Each PIS register is accessible via two channel addresses.

Each of the above bits is assigned to the corresponding Universal Port pin with the same number (e.g. PISA.0 to pin PA0). Bit PISn is set and an interrupt is generated on INT if

– the corresponding Universal Port pin Pn is defined as input via register PCR and

– the interrupt source is enabled by resetting the corresponding interrupt mask PIMn in register PIM and

– a state transition has occurred on pin Pn. For definite detection of a real state transition, pulse width should not be shorter than 20 ns.

Note: Bits PISn are reset when register PIS is read. Masked interrupts are not normally indicated when PIS is read. Instead, they remain internally stored and pending. A pending interrupt is generated when the corresponding mask bit is reset to zero. However, if bit IPC.VIS is set to “1”, interrupt statuses in PIS may be flagged although they are masked via register PIM. These masked interrupt statuses neither generate an interrupt vector or a signal on INT, nor are visible in register GIS.

If more than one consecutive state transitions occur on the same pin before the PIS register is read, only one interrupt request will be generated.

Port Interrupt Mask Register Port A...D (WRITE)

Value after RESET: FF_H

	7					0	
PIMA	PIM7					PIM0	(03D/07D)
PIMB	PIM7					PIM0	(0BD/0FD)
PIMC	PIM7					PIM0	(13D/17D)
PIMD	0	0	0	0	PIM3	PIM0	(1BD/1FD)

Note: Unused bits have to be set to logical “0”.

Each PIM register is accessible via two channel addresses.

Each of the above bits is assigned to the corresponding Universal Port pin and to the bits of register PIS with the same number (e.g. PIMA.0 to pin PA0).
0...Interrupt source is enabled.
1...Interrupt source is disabled.
A “1” in a bit position of PIM sets the mask active for the interrupt status in PIS. Masked interrupt statuses neither generate an interrupt vector or a signal on INT, nor are they visible in register GIS.

Moreover, they will

– not be displayed in the Interrupt Status Register if bit IPC.VIS is set to “0”

– be displayed in the Interrupt Status Register if bit IPC.VIS is set to “1”.

Refer to description of register PIS.

Note: After RESET, all interrupt sources are disabled.

Port Configuration Register Port A...D (READ/WRITE)

Value after RESET: FF_H

	7					0	
PCRA	PCR7					PCR0	(03E/07E)
PCRB	PCR7					PCR0	(0BE/0FE)
PCRC	PCR7					PCR0	(13E/17E)
PCRD	0	0	0	0	PCR3	PCR0	(1BE/1FE)

Note: Unused bits have to be set to logical “0”.

Each PCR register is accessible via two channel addresses.

Each of the above bits is assigned to the corresponding Universal Port pin with the same number (e.g. PCRA.0 to pin PA0). If bit PCRn (n = 0...7) is set to
0...pin Pn is defined as output.
1...pin Pn is defined as input.

Note: After RESET, all pins of the Universal Port are defined as inputs.

5 Electrical Specification

5.1 Absolute Maximum Ratings

Supply voltage V_{DD} = – 0.3 to + 7.0 V
Input voltage V_I = – 0.3 to V_{DD} + 0.3 V (max. 7 V)
Output voltage V_0 = – 0.3 to V_{DD} + 0.3 V (max. 7 V)
Storage temperature T_{STG} = – 65 to + 150 ˚C

Note: Stresses above those listed under “Absolute Maximum Ratings” may cause permanent damage to the device.
Exposure to conditions beyond those indicated in the recommended operational conditions of this specification may affect device reliability.
This is a stress rating only and functional operation of the device under these conditions or at any other condition beyond those indicated in the operational conditions of this specification is not implied.

5.2 DC Characteristics

SAB T_A = 0 to 70 ˚C; V_{DD} = 5 V ± 5 %, V_{SS} = 0 V
SAF T_A = – 40 to 85 ˚C; V_{DD} = 5 V ± 5 %, V_{SS} = 0 V

Parameter	Symbol	Limit Values		Unit	Test Condition
		min.	max.		
Input low voltage (not XTAL1, WIDTH)	V_{IL}	– 0.4	0.8	V	
Input high voltage (not XTAL1, WIDTH)	V_{IH}	2.0	V_{DD} + 0.4	V	
Input low voltage (WIDTH)	V_{WIL}	– 0.4	1.0	V	
Input high voltage (WIDTH)	V_{WIH}	3.5	V_{DD} + 0.4	V	
Input low voltage (XTAL1)	V_{XIL}	– 0.4	0.5	V	
Input high voltage (XTAL1)	V_{XIH}	3.5	V_{DD} + 0.4	V	

DC Characteristics (cont'd)

Parameter		Symbol	Limit Values		Unit	Test Condition
			min.	max.		
Output low voltage		V_{OL}		0.45	V	I_{OL} = 7 mA (pins TxD, RxD) I_{OL} = 2 mA (all others except XTAL2)
Output high voltage Output high voltage		V_{OH} V_{OH}	2.4 V_{DD} – 0.5		V V	I_{OH} = – 400 µA I_{OH} = – 100 µA
Power supply current	operational	I_{CC}		40	mA	V_{DD} = 5 V C_P = 2 MHz Inputs at 0 V / V_{DD} no output loads
	power down			6	mA	
Input leakage current Output leakage curr.		I_{LI} I_{LO}		10	µA	0 V < V_{IN} < V_{DD} to 0 V 0 V < V_{OUT} < V_{DD} to 0 V

5.3 Capacitances

T_A = 25 ˚C; V_{DD} = 5 V ± 5 %, V_{SS} = 0 V

Parameter	Symbol	Limit Values		Unit
		typ.	max.	
Input capacitanceNote 1	C_{IN}	5	10	pF
Output capacitanceNote 1	C_{OUT}	8	15	pF
I/O capacitanceNote 1	C_{IO}	10	20	pF

Note 1: Not tested in production.

5.4 AC Characteristics

SAB T_A = 0 to 70 °C; V_{DD} = 5 V ± 5 %, V_{SS} = 0 V
SAF T_A = – 40 to 85 °C; V_{DD} = 5 V ± 5 %, V_{SS} = 0 V

All inputs except XTAL1 and WIDTH are driven to V_{IH} = 2.4 V for a logical “1”
and to V_{IL} = 0.4 V for a logical “0”

XTAL1 and WIDTH (CMOS inputs) are driven to V_{IH} = 4.0 V for a logical “1”
and to V_{IL} = 0.4 V for a logical “0”

Timing measurements (except for XTAL2) are made at V_H = 2.0 V for a logical “1”
and at V_L = 0.8 V for a logical “0”

Timing measurements for XTAL2 are made at V_H = 3.5 V for a logical “1”
and at V_L = 1.0 V for a logical “0”

The AC testing input/output waveforms are shown below.

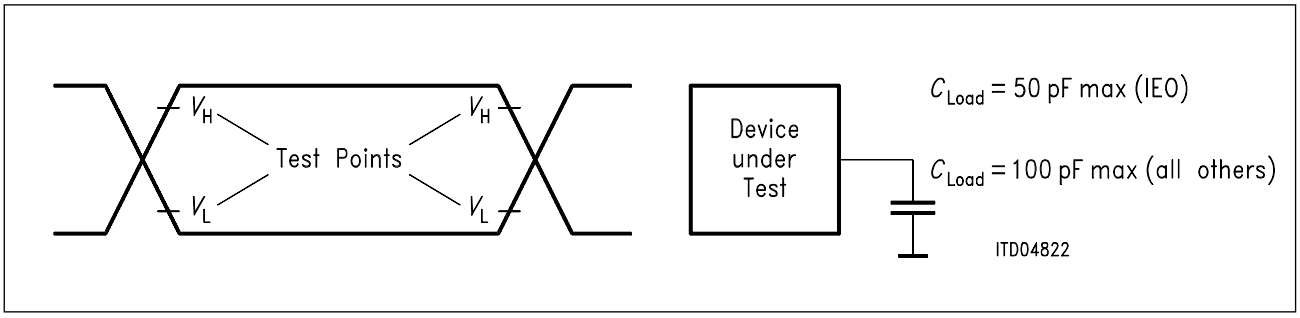


Figure 51
Input/Output Waveform for AC Tests

5.4.1 Microprocessor Interface

5.4.1.1 Siemens/Intel Bus Interface Mode

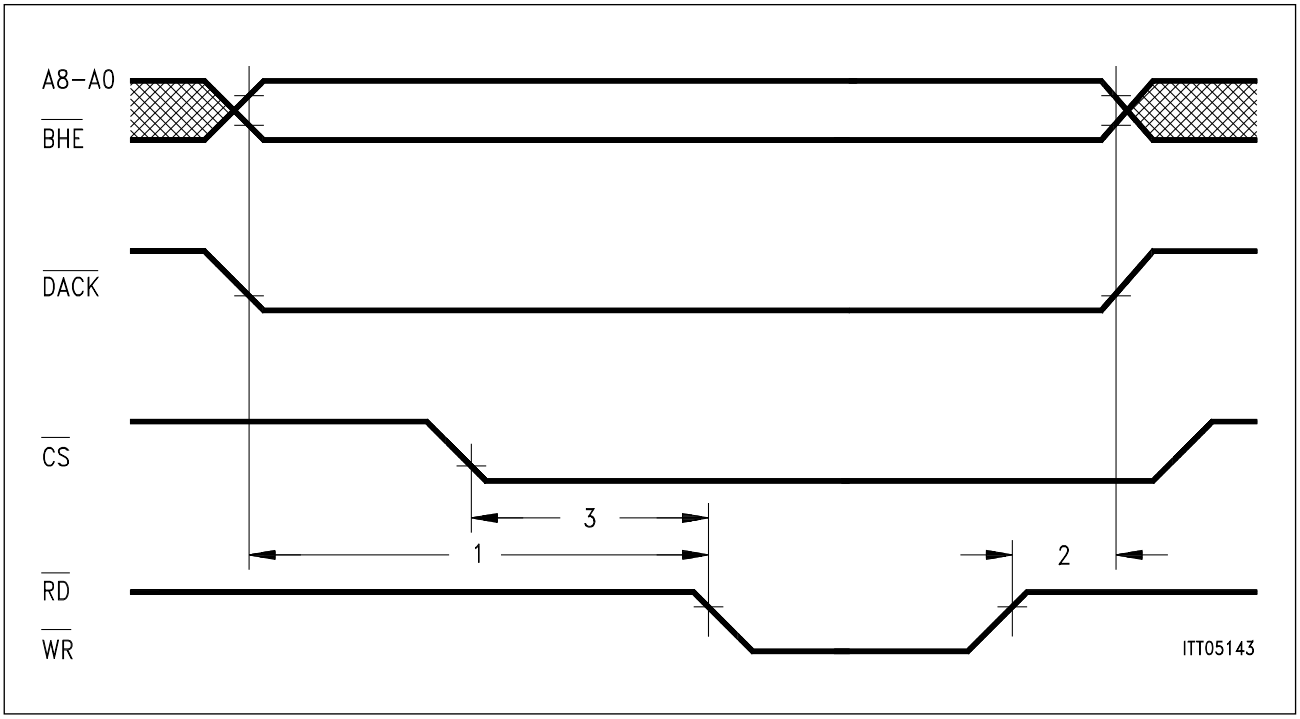


Figure 52
Siemens/Intel Non-Multiplexed Address Timing

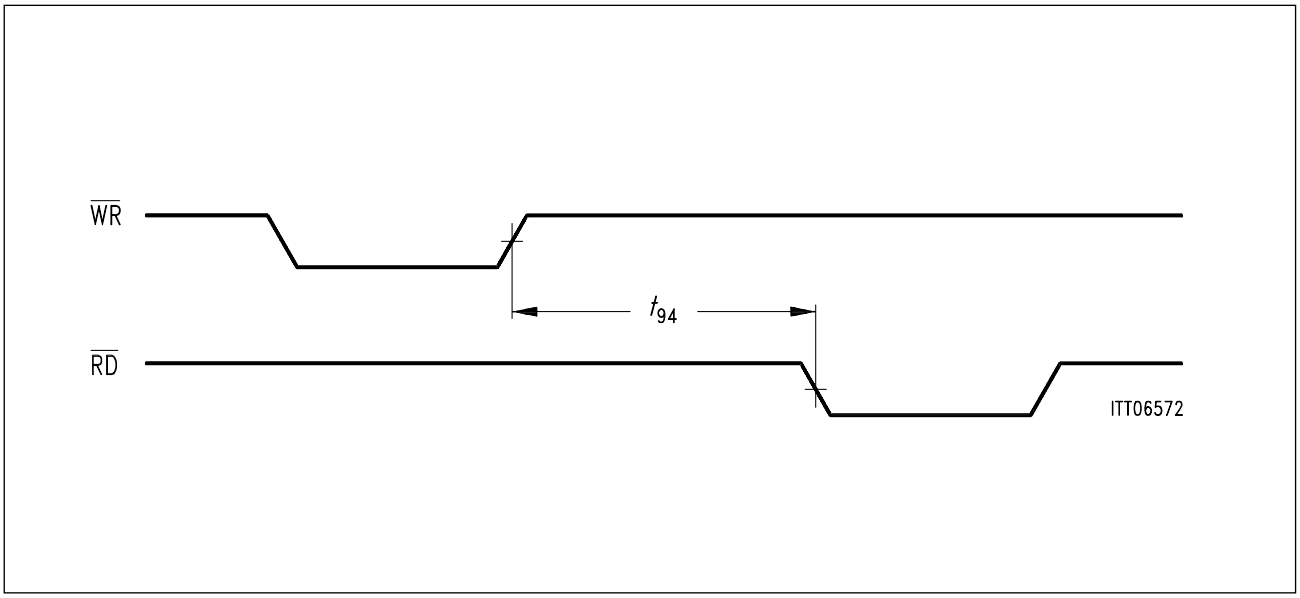


Figure 53
Siemens/Intel $\overline{WR}$ to $\overline{RD}$ Control Interval

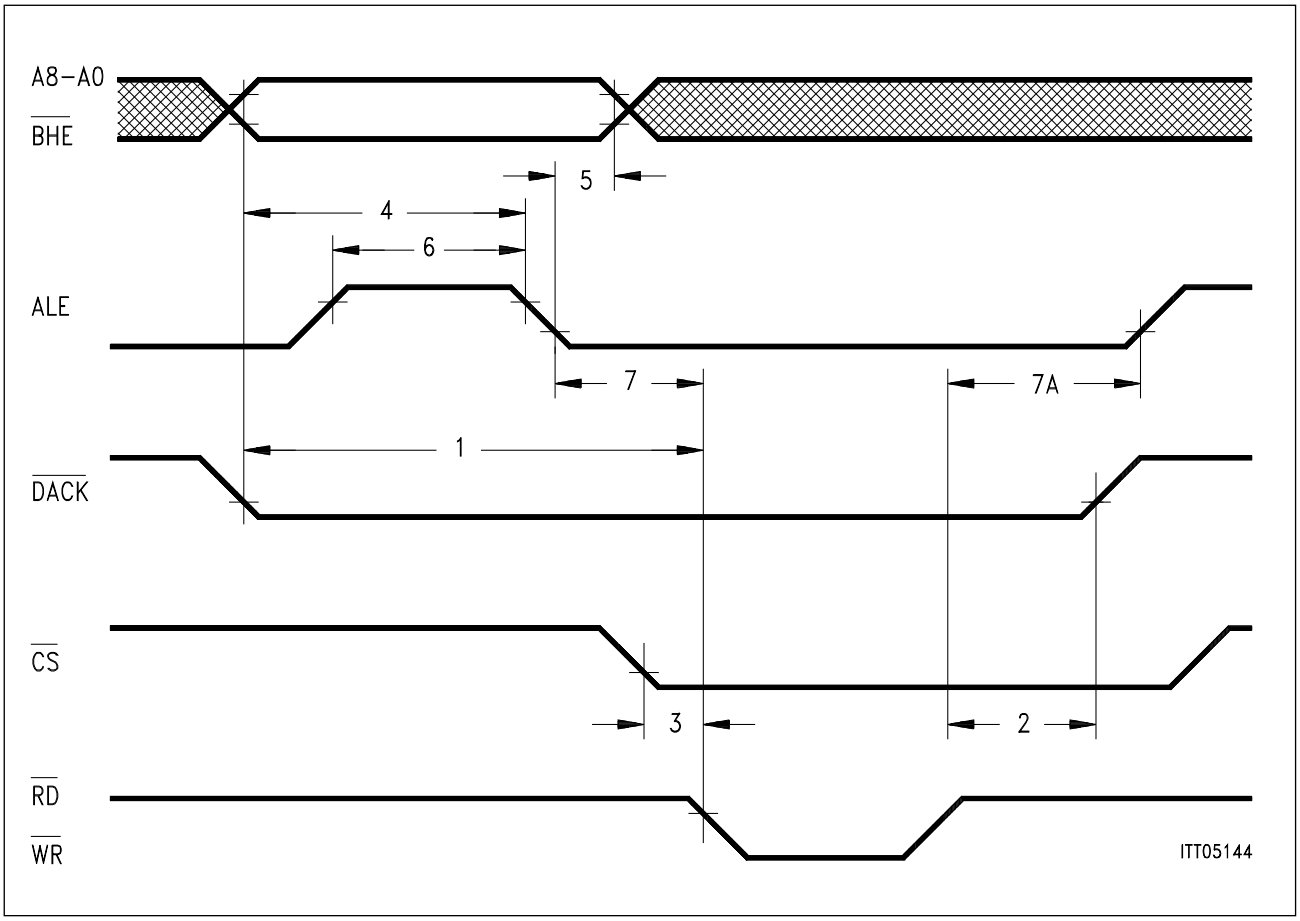


Figure 54
Siemens/Intel Multiplexed Address Timing

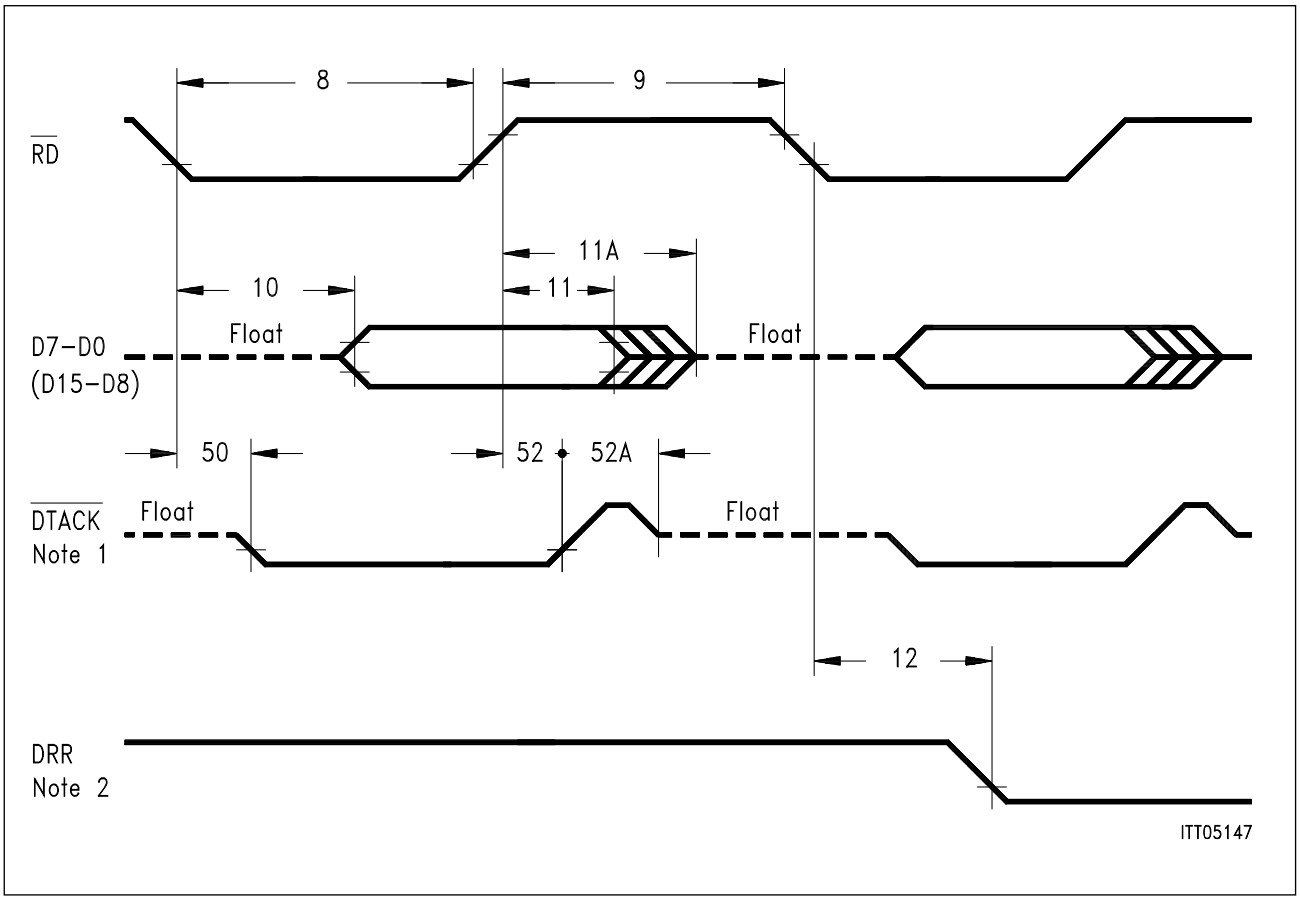


Figure 55
Siemens/Intel Read Cycle Timing

Note 1: Function of $\overline{\text{DTACK}}$ is described logically as:
$\overline{\text{DTACK}} = (\overline{\text{CS}} \times \overline{\text{DACK}} + \overline{\text{RD}} \times \overline{\text{WR}}) \times \overline{\text{INTAi}}$
$\overline{\text{INTAi}}$ is an internally generated signal.

Note 2: DRR is reset with the falling edge of $\overline{\text{RD}}$ during the last read access to RFIFO.

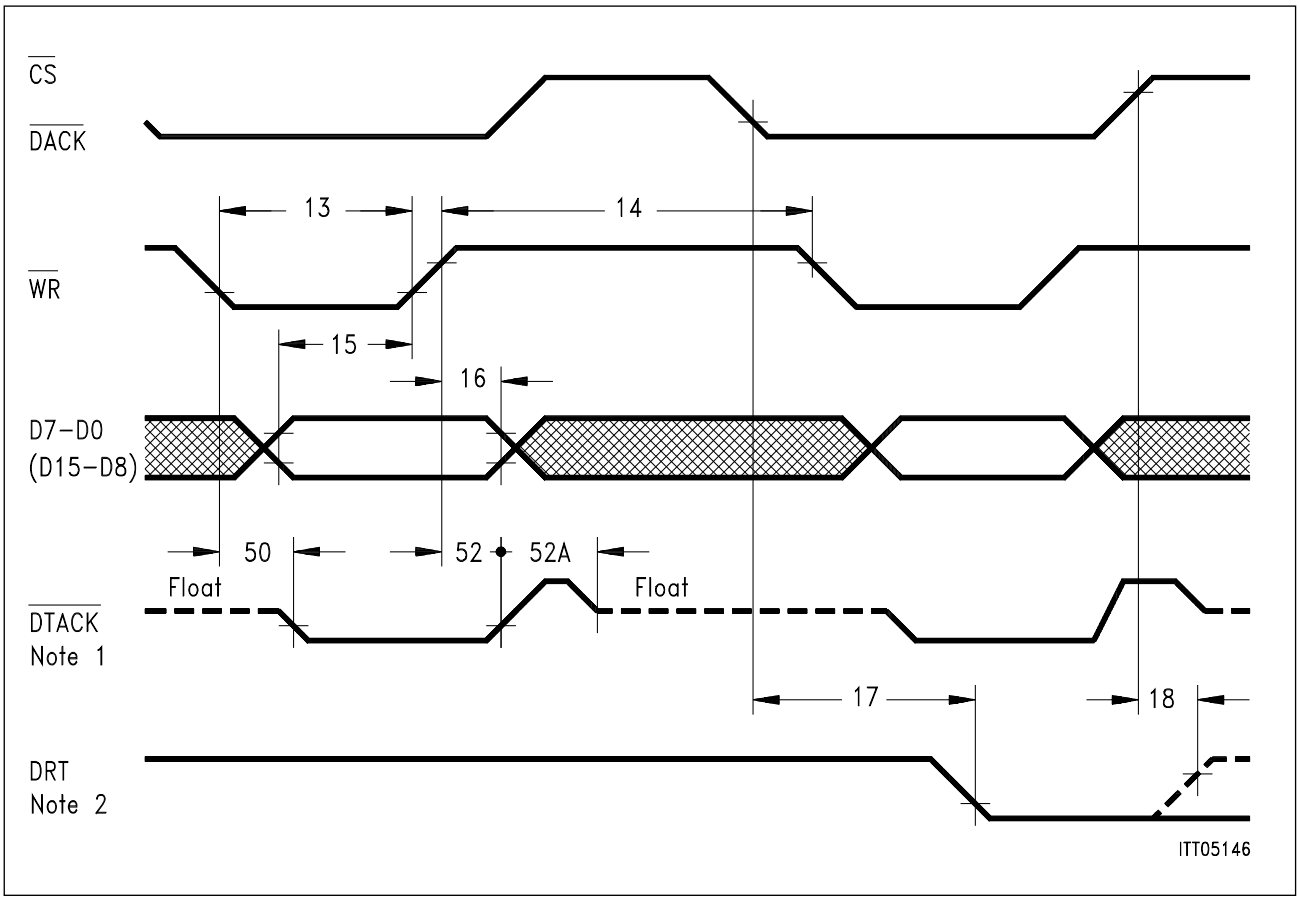


Figure 56
Siemens/Intel Write Cycle Timing

Note 1: Function of $\overline{\text{DTACK}}$ is described logically as:
$\overline{\text{DTACK}} = (\overline{\text{CS}} \times \overline{\text{DACK}} + \overline{\text{RD}} \times \overline{\text{WR}}) \times \overline{\text{INTAi}}$
$\overline{\text{INTAi}}$ is an internally generated signal.

Note 2: DRT is reset with the falling edge of $\overline{\text{CS}}$ or $\overline{\text{DACK}}$ if the last write access to XFIFO is expected. However, DRT will be activated again in the case of an access to any other register or FIFO.

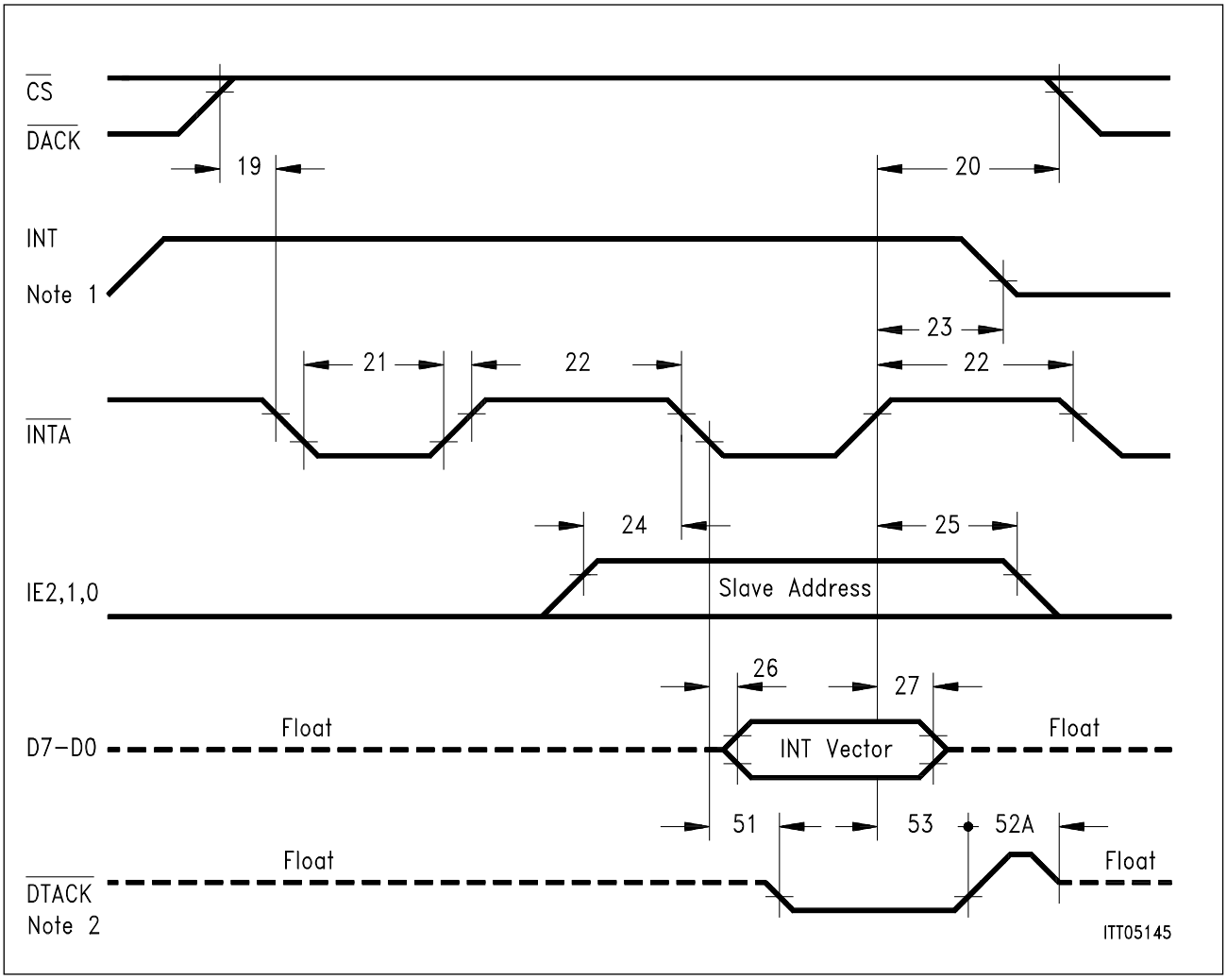


Figure 57
Siemens/Intel Interrupt Timing (slave mode)

Note 1: Timing valid for active-high push-pull signal. Timing for active-low push-pull signal is the same.
In case of an open drain output, reset time (T23) depends on external devices.

Note 2: Function of $\overline{\text{DTACK}}$ is described logically as:
$\overline{\text{DTACK}} = (\overline{\text{CS}} \times \overline{\text{DACK}} + \overline{\text{RD}} \times \overline{\text{WR}}) \times \overline{\text{INTAi}}$
$\overline{\text{INTAi}}$ is an internally generated signal. It is generated if the interrupt acknowledge cycle is considered valid.

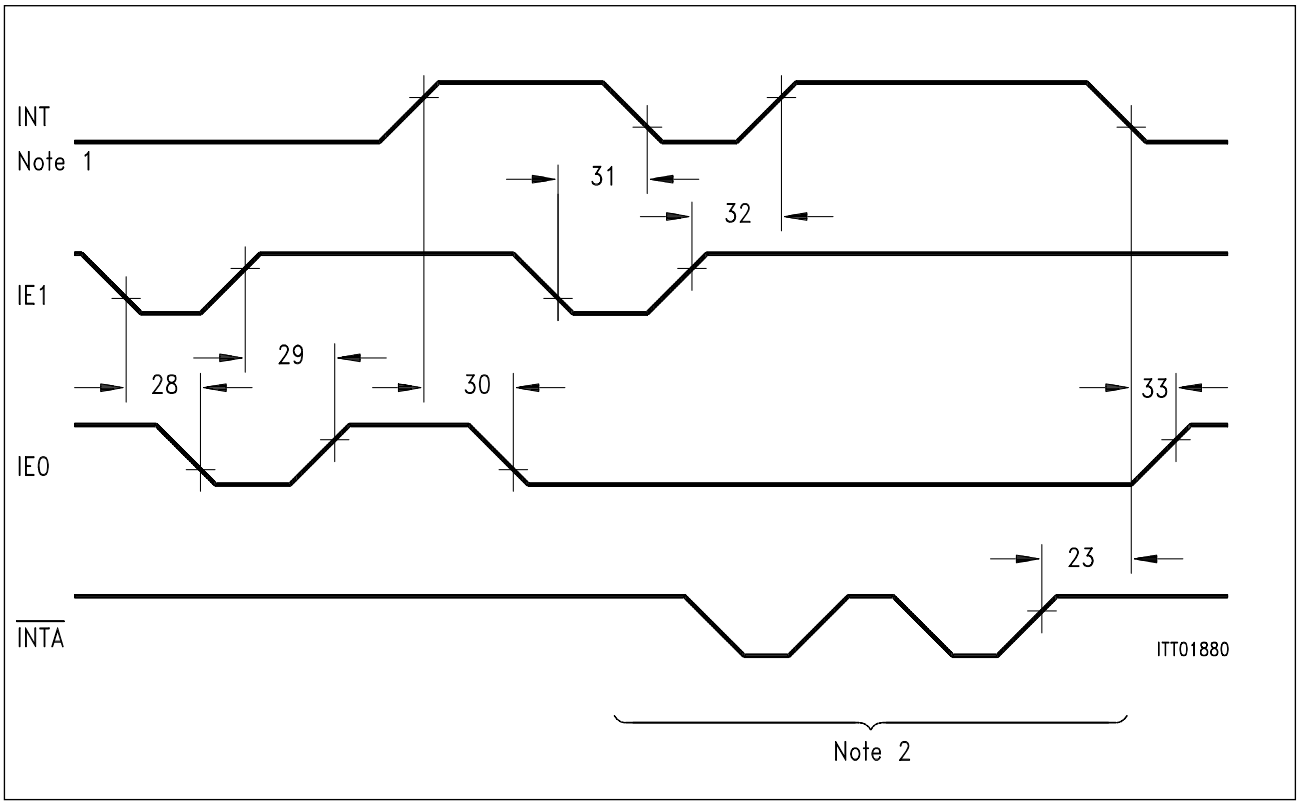


Figure 58
Siemens/Intel Interrupt Timing (Daisy chaining)

Note 1: Timing valid for active-high push-pull signal. Timing for active-low push-pull signal is the same.
In case of an open-drain output, reset times (T23, T31) depend on external devices.

Note 2: Timing for $\overline{CS}$, $\overline{DACK}$, INT, $\overline{INTA}$ and D7-D0 is similar to slave mode.

Siemens/Intel Bus Interface and Interrupt Timing

Parameter	No.	Symbol	Limit Values		Unit
			min.	max.	
Address, $\overline{BHE}$, $\overline{DACK}$ setup time	1	$t_{su(A)}$	15		ns
Address, $\overline{BHE}$, $\overline{DACK}$ hold time	2	$t_{h(A)}$	0		ns
$\overline{CS}$ setup time	3	$t_{su(S)}$	0		ns
$\overline{CS}$ hold time	3A	$t_{h(S)}$	0		ns
Address, $\overline{BHE}$ stable before ALE inactive	4	$t_{su(A-ALE)}$	20		ns
Address, $\overline{BHE}$ hold after ALE inactive	5	$t_{h(ALE-A)}$	10		ns
ALE pulse width	6	$t_{w(ALE)}$	30		ns
Address latch setup time before cmd active	7	$t_{su(ALE)}$	0		ns
ALE to command inactive delay	7A	$t_{rec(ALE)}$	20		ns
$\overline{RD}$ pulse width	8	$t_{w(R)}$	90		ns
$\overline{RD}$ control interval	9	$t_{rec(R)}$	50		ns
Data valid after $\overline{RD}$ active	10	$t_{a(R)}$		80	ns
Data hold after $\overline{RD}$ inactive	11	$t_{v(R)}$	10		ns
$\overline{RD}$ inactive to data bus tristate **Note 1**	11A	$t_{dis(R)}$		40	ns
DRR low after $\overline{RD}$ active	12	$t_{p(DDR)}$		65	ns
$\overline{WR}$ pulse width	13	$t_{w(W)}$	50		ns
$\overline{WR}$ control interval	14	$t_{rec(W)}$	35		ns
Data stable before $\overline{WR}$ inactive	15	$t_{su(D)}$	30		ns

Note: Not tested in production.

Siemens/Intel Bus Interface and Interrupt Timing (cont'd)

Parameter	No.	Symbol	Limit Values		Unit
			min.	max.	
Data hold after $\overline{WR}$ inactive	16	$t_{h(D)}$	10		ns
DRT low after $\overline{CS}$, $\overline{DACK}$ active	17	$t_{dis(DRT)}$		50	ns
DRT return to one after $\overline{CS}$, $\overline{DACK}$ inactive	18	$t_{p(DRT)}$		60	ns
$\overline{CS}$, $\overline{DACK}$ inactive setup ($\overline{INTA}$ cycle)	19	$t_{dis(S-INT)}$	0		ns
$\overline{CS}$, $\overline{DACK}$ inactive hold ($\overline{INTA}$ cycle)	20	t_{INTA-S}	0		ns
$\overline{INTA}$ pulse width	21	$t_{w(INTA)}$	75		ns
$\overline{INTA}$ control interval	22	$t_{rec(INTA)}$	30		ns
INT reset after last $\overline{INTA}$ inactive	23	$t_{INTA-INT}$		60	ns
Slave address (IE0, IE1, IE2) setup time	24	$t_{su(IE)}$	10		ns
Slave address (IE0, IE1, IE2) hold time	25	$t_{h(IE)}$	30		ns
Interrupt vector (D7-D0) valid after INTA active	26	$t_{a(VEC)}$		50	ns
Interrupt vector (D7-D0) hold after INTA inactive	27	$t_{v(VEC)}$	10		ns
Interrupt vector (D7-D0) hold after INTA inactive	27A	$t_{h(VEC)}$		40	ns
IE0 low after IE1 low	28	$t_{IE1L-IE0L}$		20	ns
IE0 high after IE1 high	29	$t_{IE1H-IE0H}$		20	ns
IE0 low after INT active	30	$t_{INTV-IE0L}$		10	ns
INT inactive after IE1 low	31	$t_{dis(INT)}$		25	ns

Note: 27A and 52A are not tested in production

Siemens/Intel Bus Interface and Interrupt Timing (cont'd)

Parameter	No.	Symbol	Limit Values		Unit
			min.	max.	
INT reactivated after IE1high	32	$t_{IE1H-INTV}$		25	ns
IE0 high after INT reset	33	$t_{INT-IE0H}$		30	ns
$\overline{DTACK}$ active after command active	50	$t_{p(DTK)}$		60	ns
$\overline{DTACK}$ active after $\overline{INTA}$ active	51	$t_{p(INT-DTK)}$		35	ns
$\overline{DTACK}$ hold after command inactive	52	$t_{v(DTK)}$	10		ns
$\overline{DTACK}$ high to $\overline{DTACK}$ high impedance	52A	$t_{h(DTK)}$		40	ns
$\overline{DTACK}$ hold after $\overline{INTA}$ inactive	53	$t_{v(INT-DTK)}$	10		ns
$\overline{WR}$ to $\overline{RD}$ control interval	94	t_{94}	100		ns

Note: 27A and 52A are not tested in production
94 tbd in 5.95

5.4.1.2 Motorola Bus Interface Mode

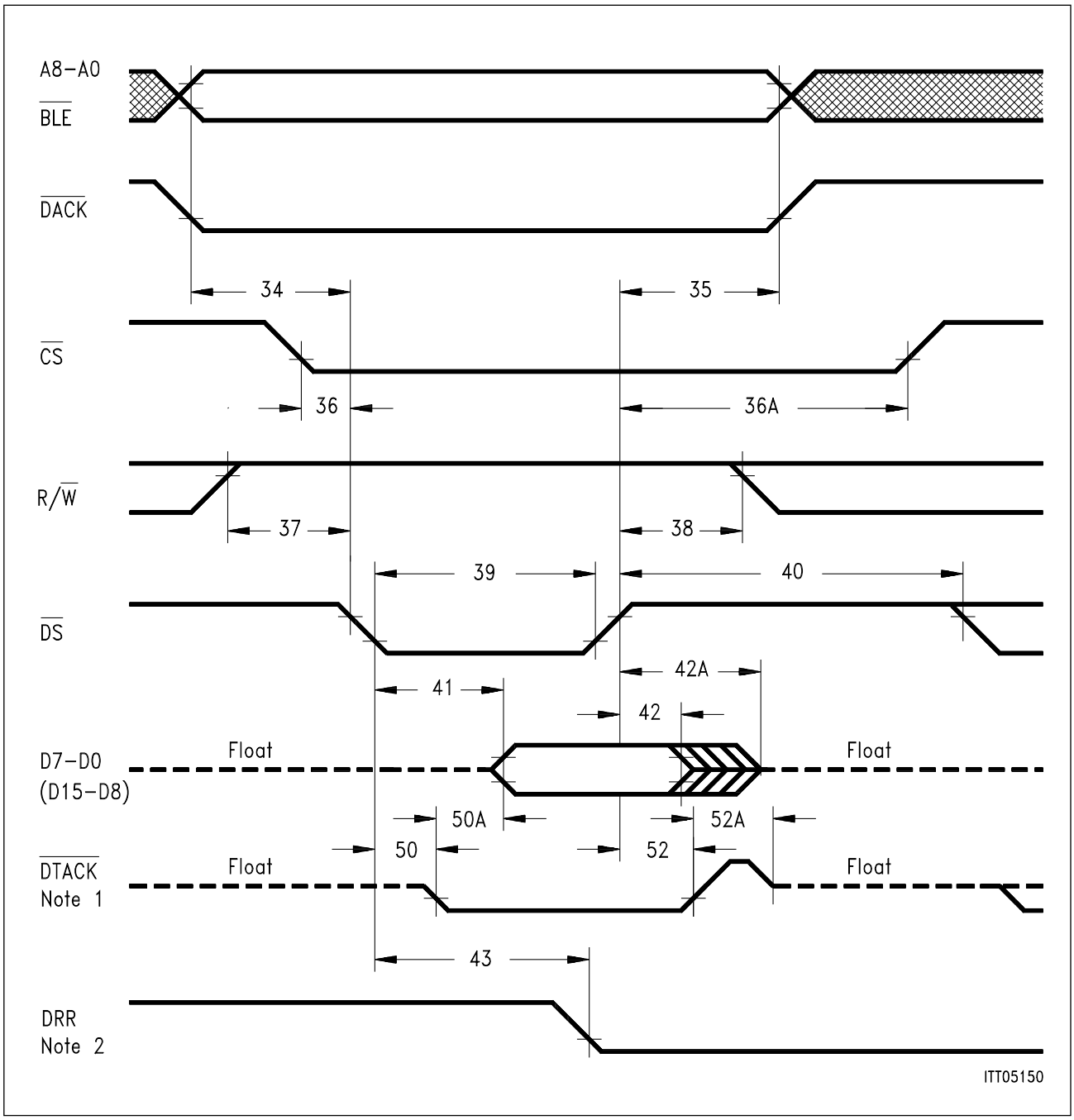


Figure 59
Motorola Read Cycle Timing

Note 1: Function of $\overline{\text{DTACK}}$ is described logically as:
$\overline{\text{DTACK}} = \overline{\text{CS}} \times \overline{\text{DACK}} \times \overline{\text{INTAi}} + \overline{\text{DS}} \times \text{R}/\overline{\text{W}}$)
i.e. in accordance with common specifications of Motorola read accesses the timing of $\overline{\text{DTACK}}$ is normally determined by $\overline{\text{DS}}$.

Note 2: DRR is reset with the falling edge of $\overline{\text{DS}}$ during the last read access to RFIFO.

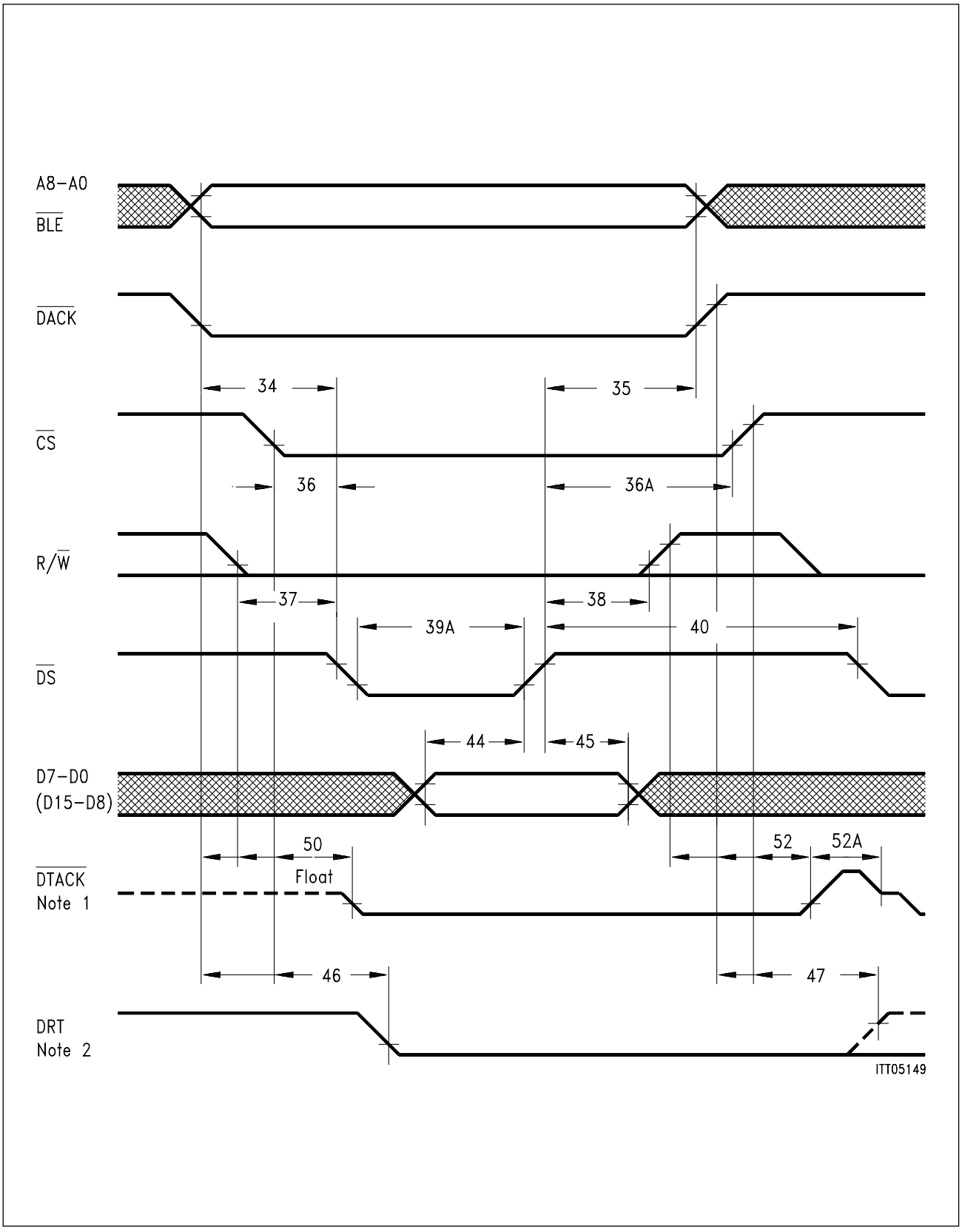


Figure 60
Motorola Write Cycle Timing

Note 1: Function of $\overline{\text{DTACK}}$ is described logically as:
$\overline{\text{DTACK}} = \overline{\text{CS}} \times \overline{\text{DACK}} \times \overline{\text{INTAi}} + \overline{\text{DS}} \times \overline{\text{R/W}}$
i.e. in accordance with common specifications of Motorola accesses
$\overline{\text{DTACK}}$ goes active if either $\overline{\text{CS}}$ or $\overline{\text{DACK}}$x is active and $\overline{\text{R/W}}$ goes low
$\overline{\text{DTACK}}$ goes inactive if $\overline{\text{CS}}$ and $\overline{\text{DACK}}$x are inactive or write $\overline{\text{R/W}}$ goes high.
To guarantee correct function in the case of write bursts signals $\overline{\text{CS}}$ and $\overline{\text{DACK}}$x have to be inactive after each write access (e.g. by deriving them from the Address Strobe $\overline{\text{AS}}$).

Note 2: DRT is reset with the falling edge of $\overline{\text{CS}}$ or $\overline{\text{DACK}}$ if the last write access to XFIFO is expected. However, DRT will be activated again in the case of an access to any other register or FIFO.

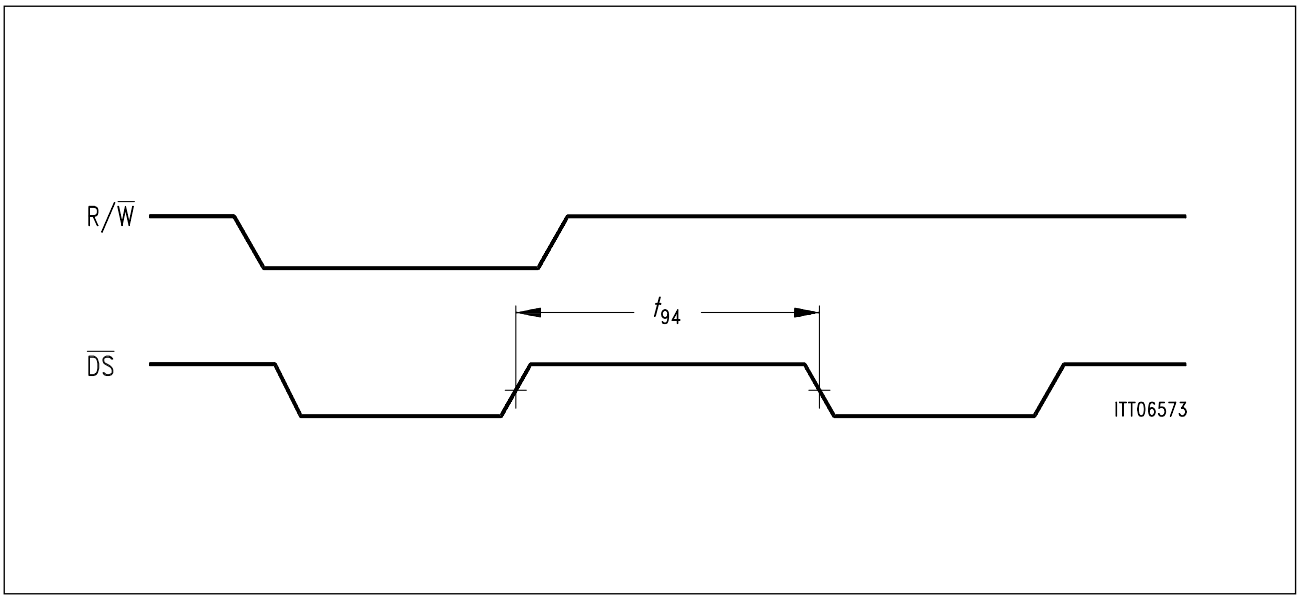


Figure 61
Motorola $\overline{\text{W}}$ to R Control Interval

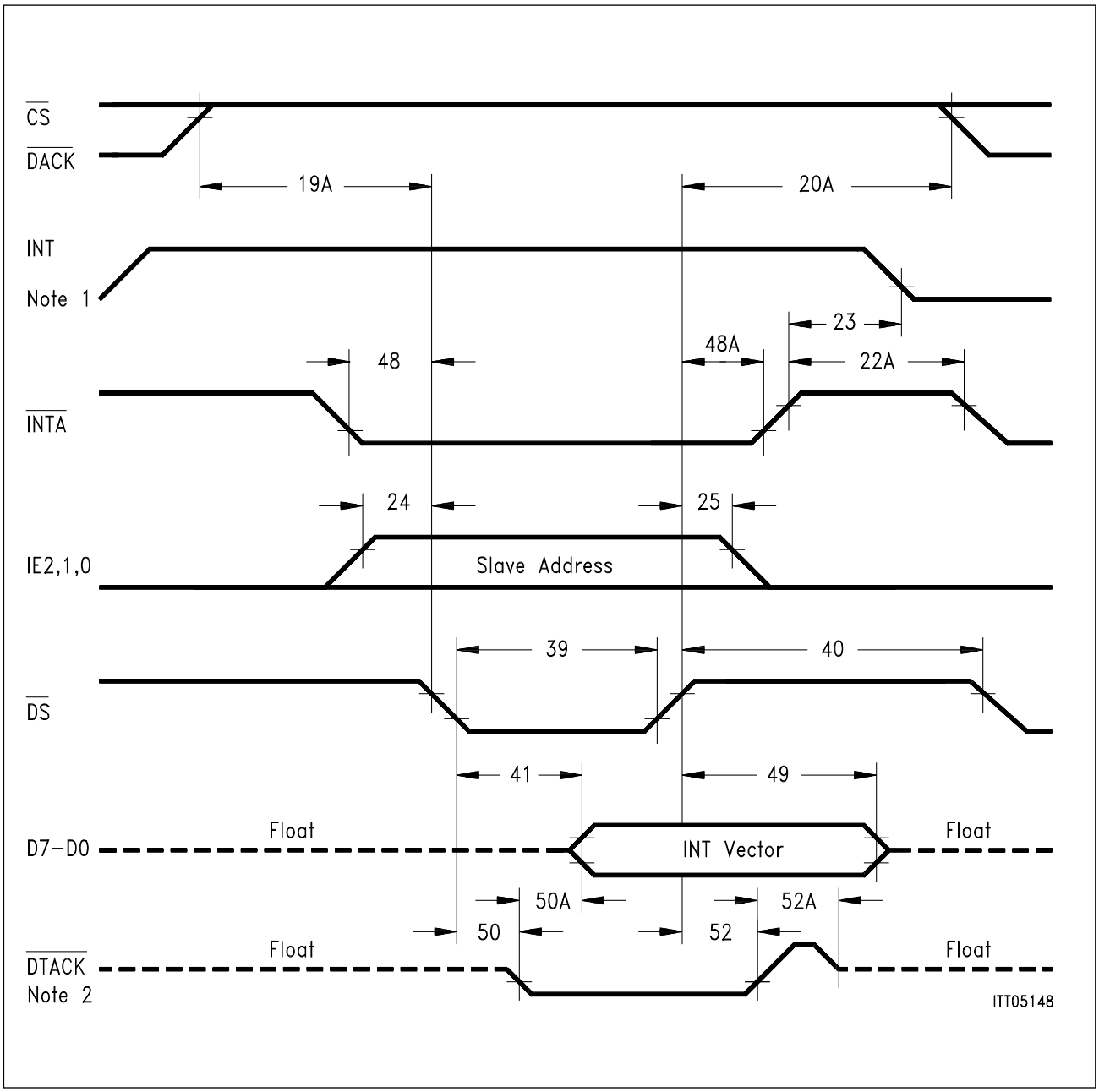


Figure 62
Motorola Interrupt Timing (slave mode)

Note 1: Timing valid for active-high push-pull signal. Timing for active-low push-pull signal is the same.
In the case of an open-drain output, reset times (T23, T31) depend on external devices.

Note 2: Function of $\overline{\text{DTACK}}$ is described logically as:
$\overline{\text{DTACK}} = \overline{\text{CS}} \times \overline{\text{DACK}} \times \overline{\text{INTAi}} + \overline{\text{DS}} \times \text{R/W}$
$\overline{\text{INTAi}}$ is an internal signal. It is generated if the interrupt acknowledge cycle is considered valid.

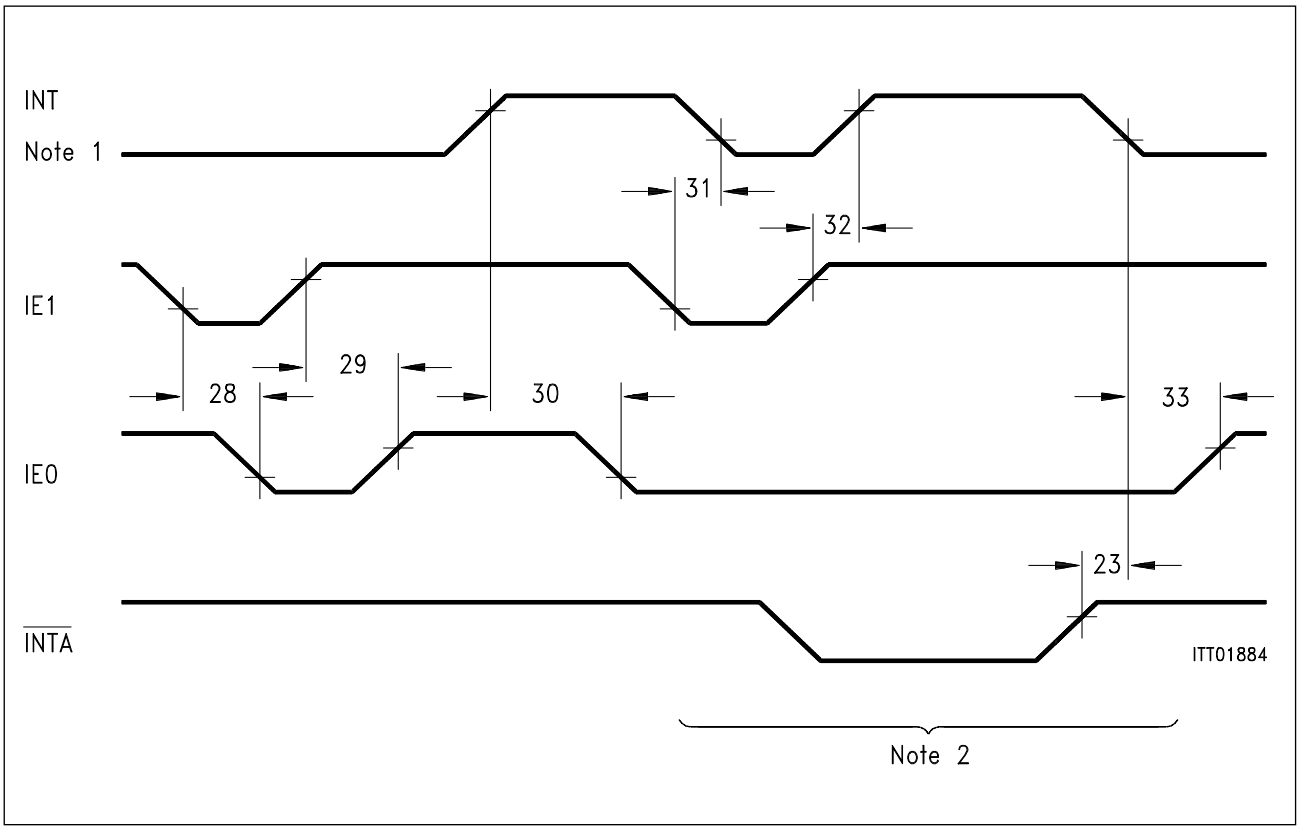


Figure 63
Motorola Interrupt Timing (Daisy chaining)

Note 1: Timing valid for active-high push-pull signal. Timing for active-low push-pull signal is the same.
In the case of an open-drain output, reset times (T23, T31) depend on external devices.

Note 2: Timing for $\overline{CS}$, $\overline{DACK}$, INT, $\overline{INTA}$, $\overline{DS}$ and D7-D0 is similar to slave mode.

Motorola Bus Interface Timing and Interrupt Timing

Parameter	No.	Symbol	Limit Values		Unit
			min.	max.	
Address, $\overline{BLE}$, $\overline{DACK}$ setup time before $\overline{DS}$ active	34	$t_{su(A)}$	15		ns
Address, $\overline{BLE}$, $\overline{DACK}$ hold after $\overline{DS}$ inactive	35	$t_{h(A)}$	0		ns
$\overline{CS}$ active before $\overline{DS}$ active	36	$t_{su(S)}$	0		ns
$\overline{CS}$ hold after $\overline{DS}$ inactive	36A	$t_{h(S)}$	0		ns
R/$\overline{W}$ stable before $\overline{DS}$ active	37	$t_{su(RW)}$	5		ns
R/$\overline{W}$ hold after $\overline{DS}$ inactive	38	$t_{h(RW)}$	0		ns
$\overline{DS}$ pulse width (read access) (write access)	39 39A	$t_{w(DS)R}$ $t_{w(DS)W}$	90 60		ns ns
$\overline{DS}$ control interval	40	$t_{rec(DS)}$	70		ns
Data valid after $\overline{DS}$ active(read access)	41	$t_{a(DS)}$		80	ns
Data hold after $\overline{DS}$ inactive (read access)	42	$t_{v(DS)}$	10		ns
$\overline{DS}$ inactive to databus tristate (read access) Note 1	42A	$t_{dis(DS)}$		40	ns
DRR low after $\overline{DS}$ active	43	$t_{p(DRR)}$		65	ns
Data stable before $\overline{DS}$ active (write access)	44	$t_{su(D)}$	30		ns
Data hold after $\overline{DS}$ inactive (write access)	45	$t_{h(D)}$	10		ns
DRT low after $\overline{DS}$ or $\overline{DACK}$ active	46	$t_{dis(DRT)}$		50	ns
DRT return to one after $\overline{CS}$ or $\overline{DACK}$ inactive	47	$t_{p(DRT)}$		50	ns
$\overline{CS}$, $\overline{DACK}$ inactive setup before $\overline{DS}$ ($\overline{INTA}$ cycle)	19A	$t_{dis(S-INTA)}$	20		ns
$\overline{CS}$, $\overline{DACK}$ inactive hold after $\overline{DS}$ ($\overline{INTA}$ cycle)	20A	$t_{h(INTA-S)}$	20		ns

Motorola Bus Interface Timing and Interrupt Timing (cont'd)

Parameter	No.	Symbol	Limit Values		Unit
			min.	max.	
$\overline{\text{INTA}}$ control interval	22A	$t_{rec(INTA)}$	30		ns
INT reset after last $\overline{\text{INTA}}$ inactive	23	$t_{INTA-INT}$		60	ns
Slave address (IE0, IE1, IE2) setup time	24	$t_{su(IE)}$	10		ns
Slave address (IE0, IE1, IE2) hold time	25	$t_{h(IE)}$	30		ns
IE0 low after IE1 low	28	$t_{IE1L-IE0L}$		20	ns
IE0 high after IE1high	29	$t_{IE1H-IE0H}$		20	ns
IE0 low after INT active	30	$t_{INTV-IE0L}$		10	ns
INT inactive after IE1 low	31	$t_{dis(INT)}$		25	ns
INT reactivated after IE1 high	32	$t_{IE1H-INTV}$		25	ns
IE0 high after INT reset	33	$t_{INT-IE0H}$		30	ns
$\overline{\text{INTA}}$ setup time	48	$t_{su(INTA)}$	0		ns
$\overline{\text{INTA}}$ hold time	48A	$t_{h(INTA)}$	0		ns
Interrupt vector hold after $\overline{\text{DS}}$ or $\overline{\text{INTA}}$ inactive	49	$t_{v(VEC)}$	10	40	ns
$\overline{\text{DTACK}}$ active delay	50	$t_{p(DTK)}$		60	ns
$\overline{\text{DTACK}}$ active to data valid (read cycle)	50A	t_{DTK-D}		45	ns
$\overline{\text{DTACK}}$ hold after command inactive	52	$t_{v(DTK)}$	10		ns
$\overline{\text{DTACK}}$ high to $\overline{\text{DTACK}}$ high impedance	52A	$t_{h(DTK)}$		40	ns
$\overline{\text{W}}$ to R control interval	95	t_{95}	100		ns

Note: 49max, 50A and 52A are not tested in production.
95 tbd in 5.95

5.4.2 Parallel Port Timing

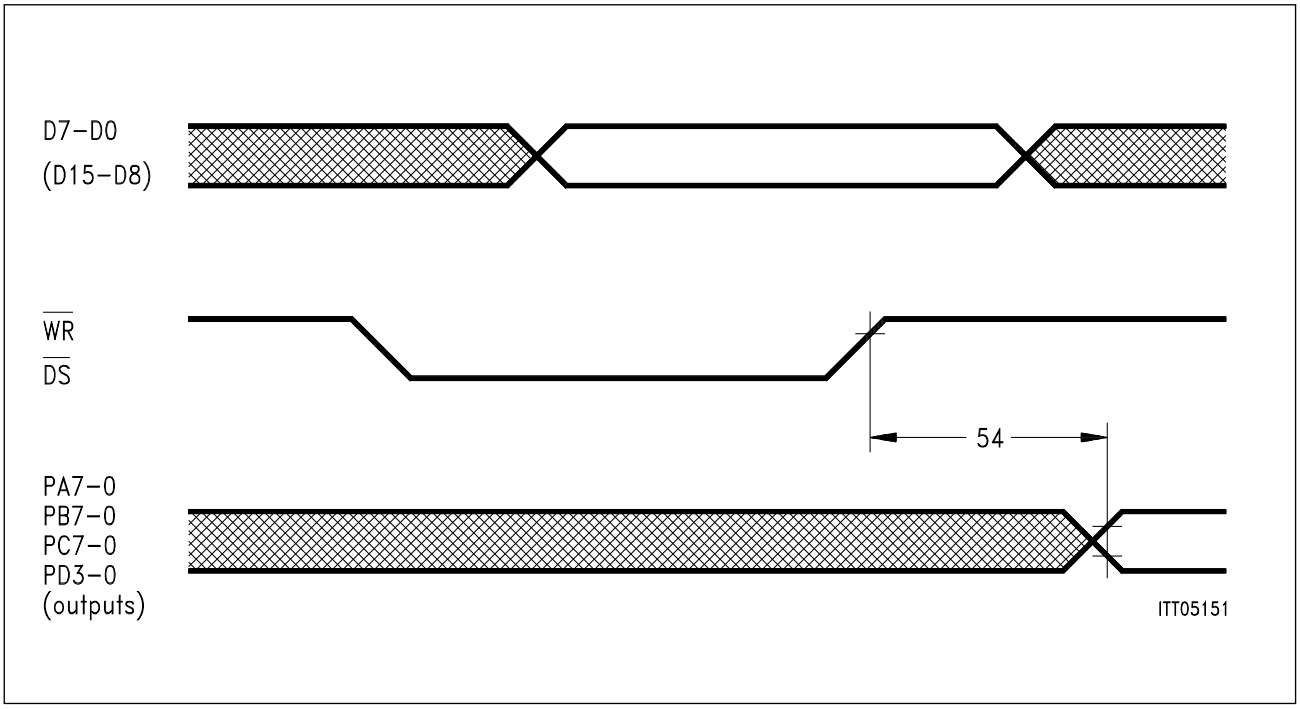


Figure 64
Parallel Port Write Access

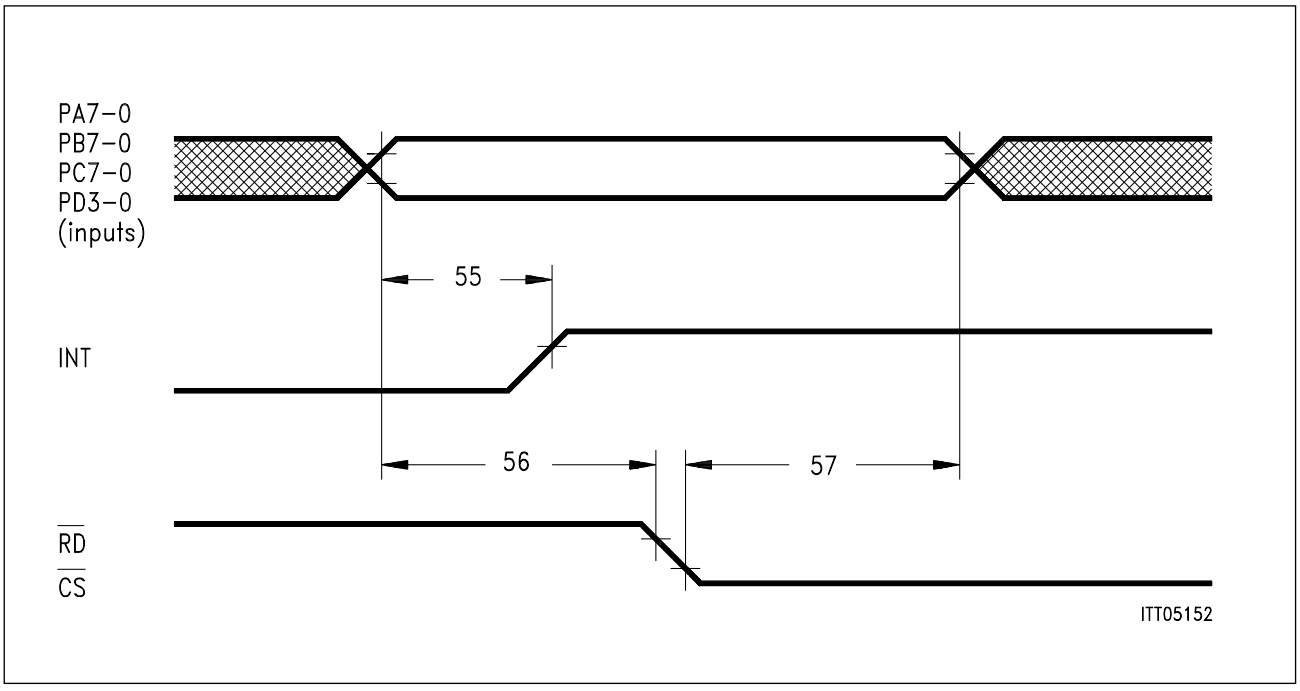


Figure 65
Parallel Port Read Access

Parallel Port Timing

Parameter	No.	Symbol	Limit Values		Unit
			min.	max.	
Port output data valid after $\overline{WR}$, $\overline{DS}$ inactive	54	t_{QV}		80	ns
Port input data change to INT active delay	55	$t_{p(PV-INT)}$		60	ns
Port input data stable before $\overline{RD}$, $\overline{DS}$ active	56	$t_{su(P)}$	20		ns
Port input data hold after $\overline{RD}$, $\overline{DS}$ active	57	$t_{h(P)}$	30		ns

5.4.3 Serial Interface

5.4.3.1 Clock Input Timing

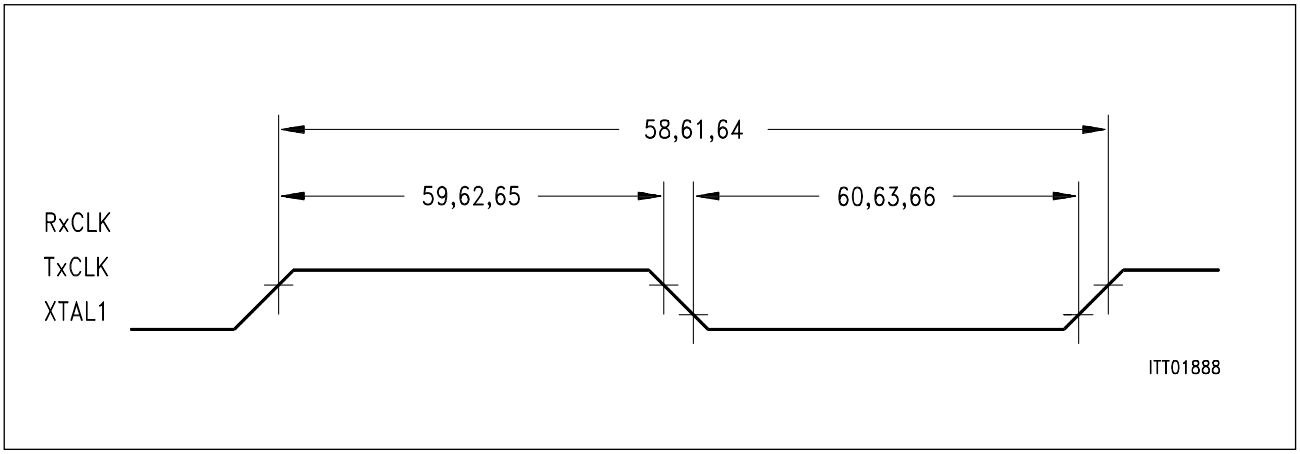


Figure 66
Clock Timing

Clock Timing

Parameter	No.	Symbol	Limit Values				Unit
			H		H-10		
			min.	max.	min.	max.	
RxCLK clock period (Note 1) (Note 3)	58	$t_{c(RxC)}$	480 50		100 50		ns ns
RxCLK high time (Note 1) (Note 3)	59	$t_{w(RxCH)}$	150 22		45 22		ns ns
RxCLK low time (Note 1) (Note 3)	60	$t_{w(RxCL)}$	150 22		45 22		ns ns
TxCLK clock period	61	$t_{c(TxC)}$	480		100		ns
TxCLK high time	62	$t_{w(TxCH)}$	150		45		ns
TxCLK low time	63	$t_{w(TxCL)}$	150		45		ns
XTAL1 clock period (Note 2) (Note 3)	64	$t_{c(XTAL1)}$	480 75		100 75		ns ns
XTAL1 high time (Note 2) (Note 3)	65	$t_{w(XTAL1H)}$	150 35		45 35		ns ns
XTAL1 low time (Note 2) (Note 3)	66	$t_{w(XTAL1L)}$	150 35		45 35		ns ns

Note 1: Externally clocked: clock mode 0, 1 except ASYNC, BCR = 16.

Note 2: Externally clocked: clock mode 4 except ASYNC, BCR = 16;
Master clock mode generally.

Note 3: Internally clocked: HDLC, BISYNC: DPLL + baud rate generator used.
ASYNC all other clocking modes.

5.4.3.2 Receive Cycle Timing

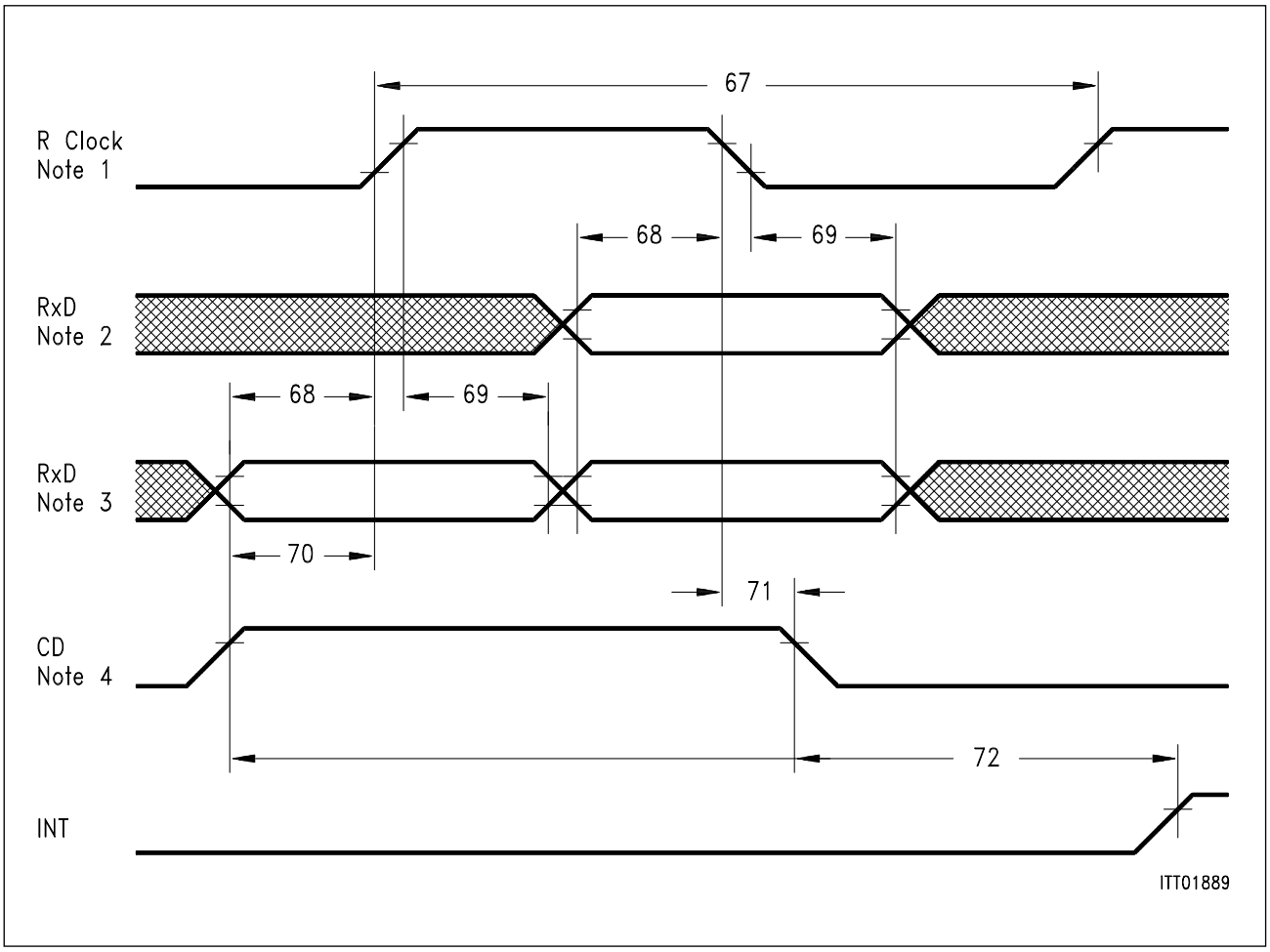


Figure 67
Receive Cycle Timing

Note 1: Whichever supplies the clock: externally clocked by R×CLK or XTAL1, or, internally derived from DPLL, BRG or BCR divider (**refer to table 5**)

Note 2: NRZ, NRZI and Manchester coding

Note 3: FM0 and FM1 coding

Note 4: Carrier detect auto start enabled (not for clock modes 1, 5)

Receive Cycle Timing

Parameter	No.	Symbol	Limit Values				Unit
			H		H-10		
			min.	max.	min.	max.	
Receive data rate							
ext. clocked (except ASYNC, BCR = 16)				2		10	Mbit/s
int. clocked (HDLC, BISYNC: only DPLL)				2		2	Mbit/s
int. clocked (all other internal modes)				2		2	Mbit/s
Clock period							
ext. clocked (except ASYNC, BCR = 16)	67	$t_{c(XC)}$	480		100		ns
int. clocked (HDLC, BISYNC: only DPLL)			480		480		ns
int. clocked (all other internal modes)			480		480		ns
Receive data setup	68	$t_{su(RxD)}$	10		10		ns
Receive data hold	69	$t_{h(RxD)}$	30		30		ns
Carrier detect setup	70	$t_{su(CD)}$	50		50		ns
Carrier detect hold	71	$t_{h(CD)}$	30		30		ns
CD status change to INT delay	72	$t_{CD\text{-}INT}$		T73 + 60		T73 + 60	ns

5.4.3.3 Transmit Cycle Timing

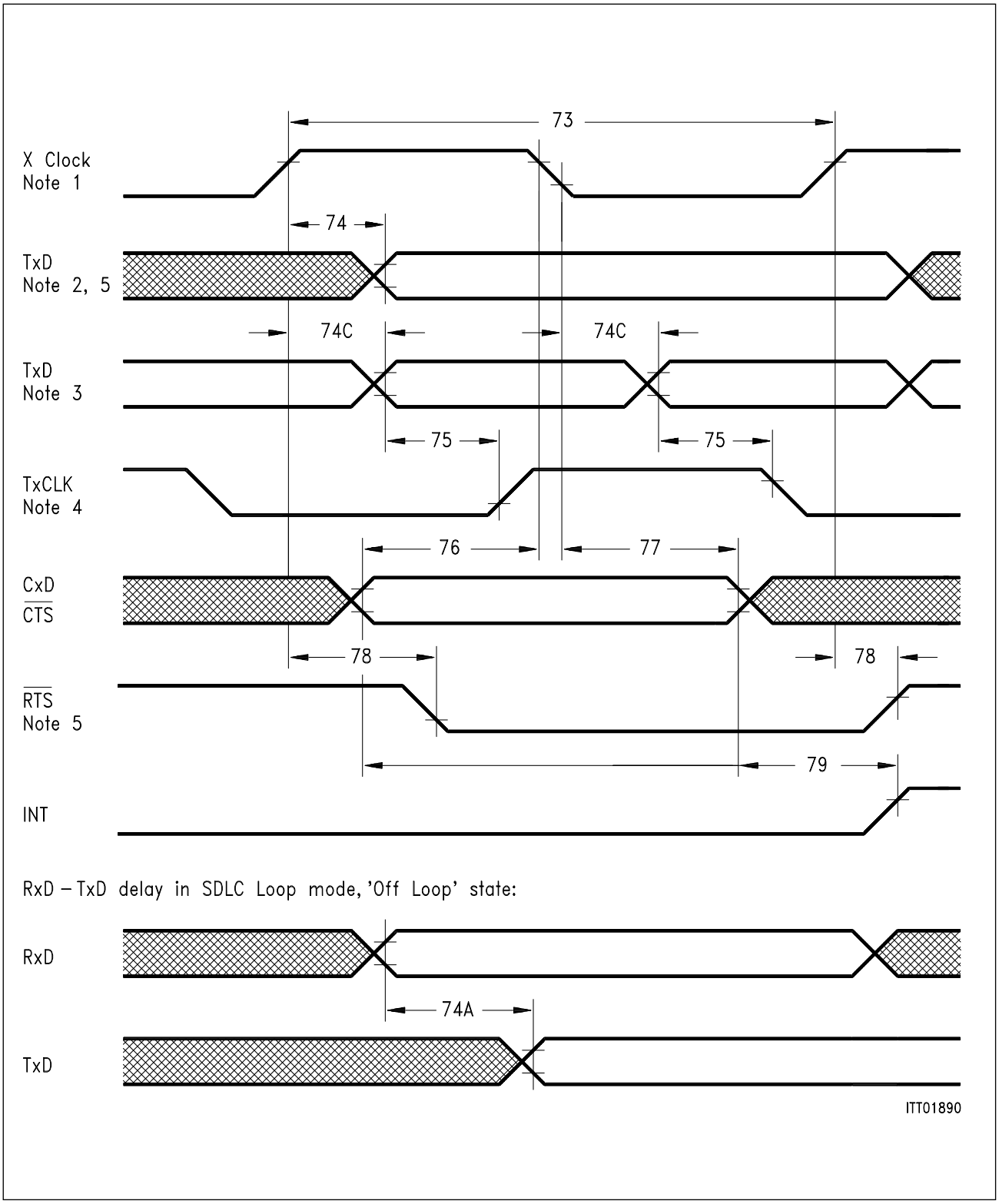


Figure 68
Transmit Cycle Timing

Note 1: Whichever supplies the clock: externally clocked by TxCLK, XTAL1 or RxCLK or, internally derived from DPLL, BRG or BCR divider (**refer to table 5**).

Note 2: NRZ and NRZI coding.

Note 3: FM0, FM1 and Manchester coding.

Note 4: If output function is enabled (**refer to table 5**).

Note 5: The timing shown is valid for normal operation and bus configuration mode 1. In bus configuration mode 2, $\overline{\text{RTS}}$ and TxD are shifted for 1/2 Xclock period.

Transmit Cycle Timing

Parameter	No.	Symbol	Limit Values				Unit
			H		H-10		
			min.	max.	min.	max.	
Transmit data rate							
ext. clocked (except ASYNC, BCR = 16)				2		10	Mbit/s
int. clocked (HDLC, BISYNC: only DPLL)				2		2	Mbit/s
int. clocked (all other internal modes)				2		2	Mbit/s
Clock period							
ext. clocked (except ASYNC, BCR = 16)	73	$t_{c(XC)}$	480		100		ns
int. clocked (HDLC, BISYNC: only DPLL)			480		480		ns
int. clocked (all other internal modes)			480		480		ns
Transmit data delay	74	$t_{p(TxD)}$		70		70	ns
Transmit data delay	74C			75		75	ns
RxD to TxD delay (SDLC loop, “Off Loop” state)	74A	$t_{p(RxD-TxD)}$		50		50	ns
Clock output to transmit data delay	75	$t_{p(XC-TxD)}$	– 30	20	– 30	20	ns
Collision data and $\overline{CTS}$ setup time	76	$t_{su(CxD)}$	10		10		ns
Collision data and $\overline{CTS}$ hold time	77	$t_{h(CxD)}$	30		30		ns
Request send delay normal operation bus configuration	78	$t_{p(RTS)}$		 65 50		 65 50	 ns ns
$\overline{CTS}$ status change to INT delay		$t_{CTS-INT}$		T73 + 60		T73 + 60	ns

5.4.3.4 Strobe Timing (clock mode 1)

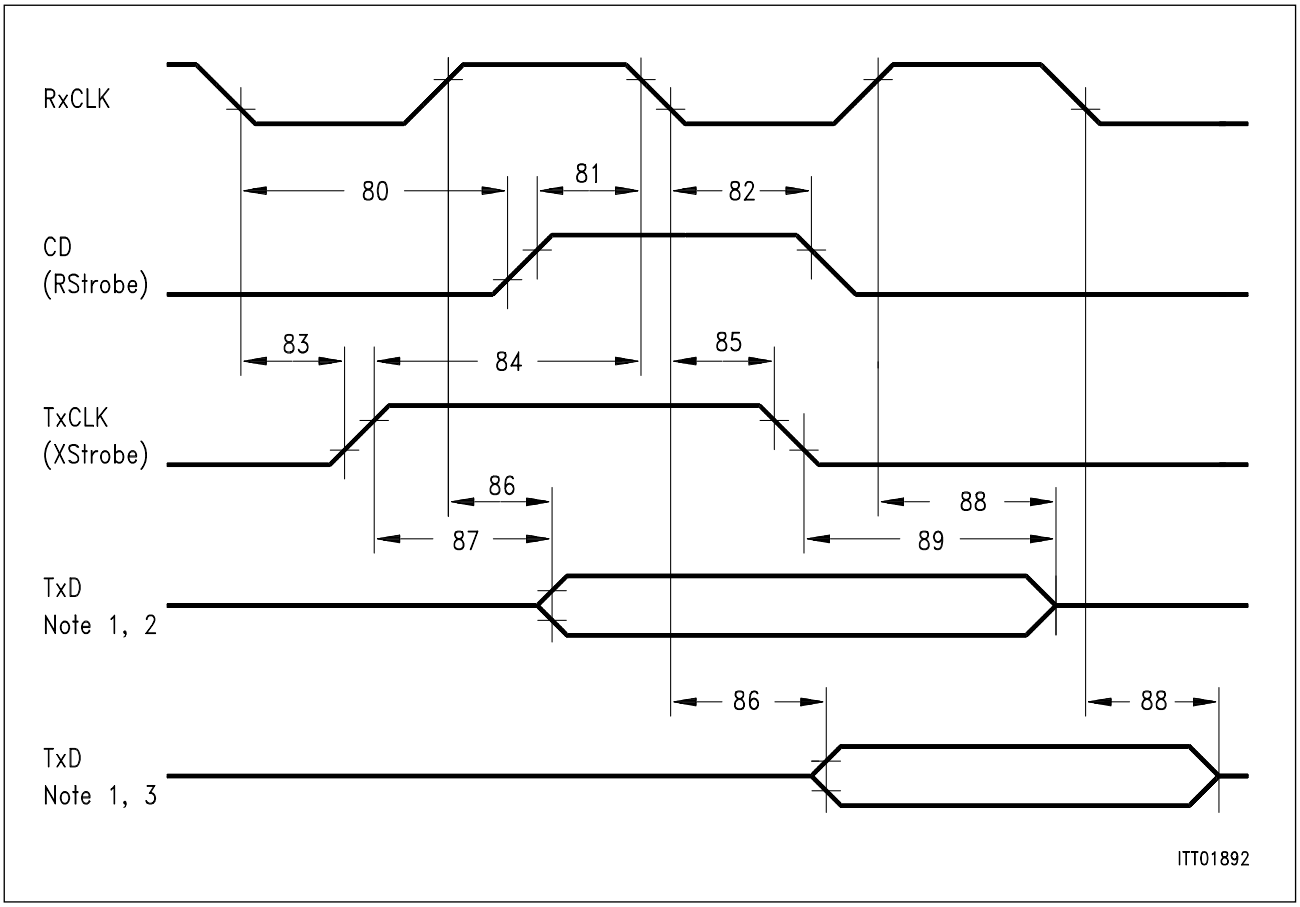


Figure 69
Strobe Timing

Note 1: High impedance if T×D is set to “open drain” function. Otherwise, active “high”.

Note 2: Normal operation and bus configuration mode 1.

Note 3: Bus configuration mode 2.

Strobe Timing

Parameter	No.	Symbol	Limit Values				Unit
			H		H-10		
			min.	max.	min.	max.	
Receive strobe delay	80	$t_{RxCL-RS}$	30		30		ns
Receive strobe setup	81	$t_{su(RS)}$	30		30		ns
Receive strobe hold	82	$t_{h(RS)}$	30		30		ns
Transmit strobe delay	83	$t_{RxCL-XS}$	30		30		ns
Transmit strobe setup	84	$t_{su(XS)}$	30		30		ns
Transmit strobe hold	85	$t_{h(XS)}$	30		30		ns
Transmit data delay from clock	86	$t_{p(RxC-TxD)}$		65		65	ns
Transmit data delay from strobe	87	$t_{p(XS-TxD)}$		50		50	ns
High impedance from clock	88	$t_{dis(RxC)}$		70		70	ns
High impedance from strobe	89	$t_{dis(XS)}$		70		70	ns

Note: 88 and 89 are not tested in production.

5.4.3.5 Synchronization Timing (clock mode 5)

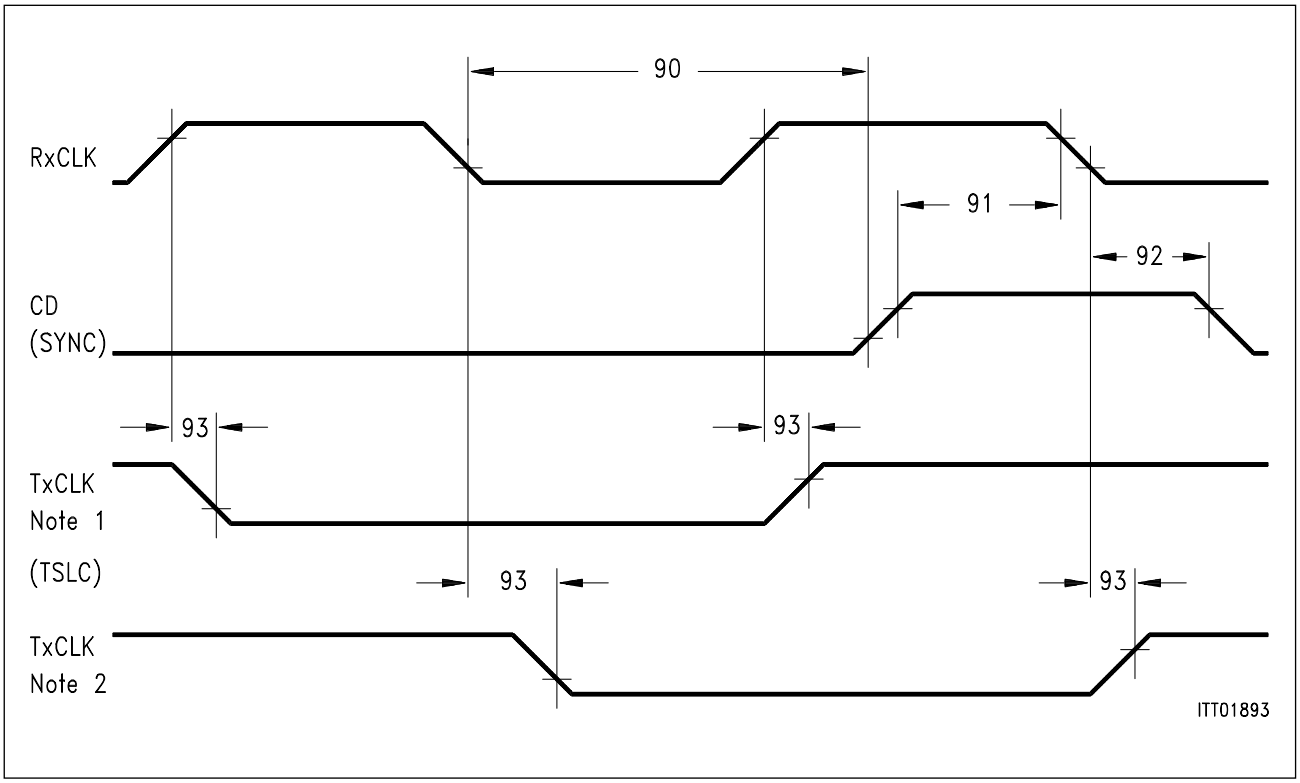


Figure 70
Synchronization Timing

Note 1: Normal operation and bus configuration mode 1.

Note 2: Bus configuration mode 2.

Synchronization Timing

Parameter	No.	Symbol	Limit Values				Unit
			H		H-10		
			min.	max.	min.	max.	
Sync pulse delay	90	$t_{RxC\text{-}SYNC}$			30		ns
Sync pulse setup	91	$t_{su(SYNC)}$			30		ns
Sync pulse hold	92	$t_{h(SYNC)}$			25		ns
Time-slot control delay	93	$t_{p(TSLC)}$			20	75	ns
XTAL1 low time (Note 2) (Note 3)	66	$t_{w(XTAL1L)}$	150		45		ns ns

Note: Clock mode 5 only specified for versions SAB 82538H-10.

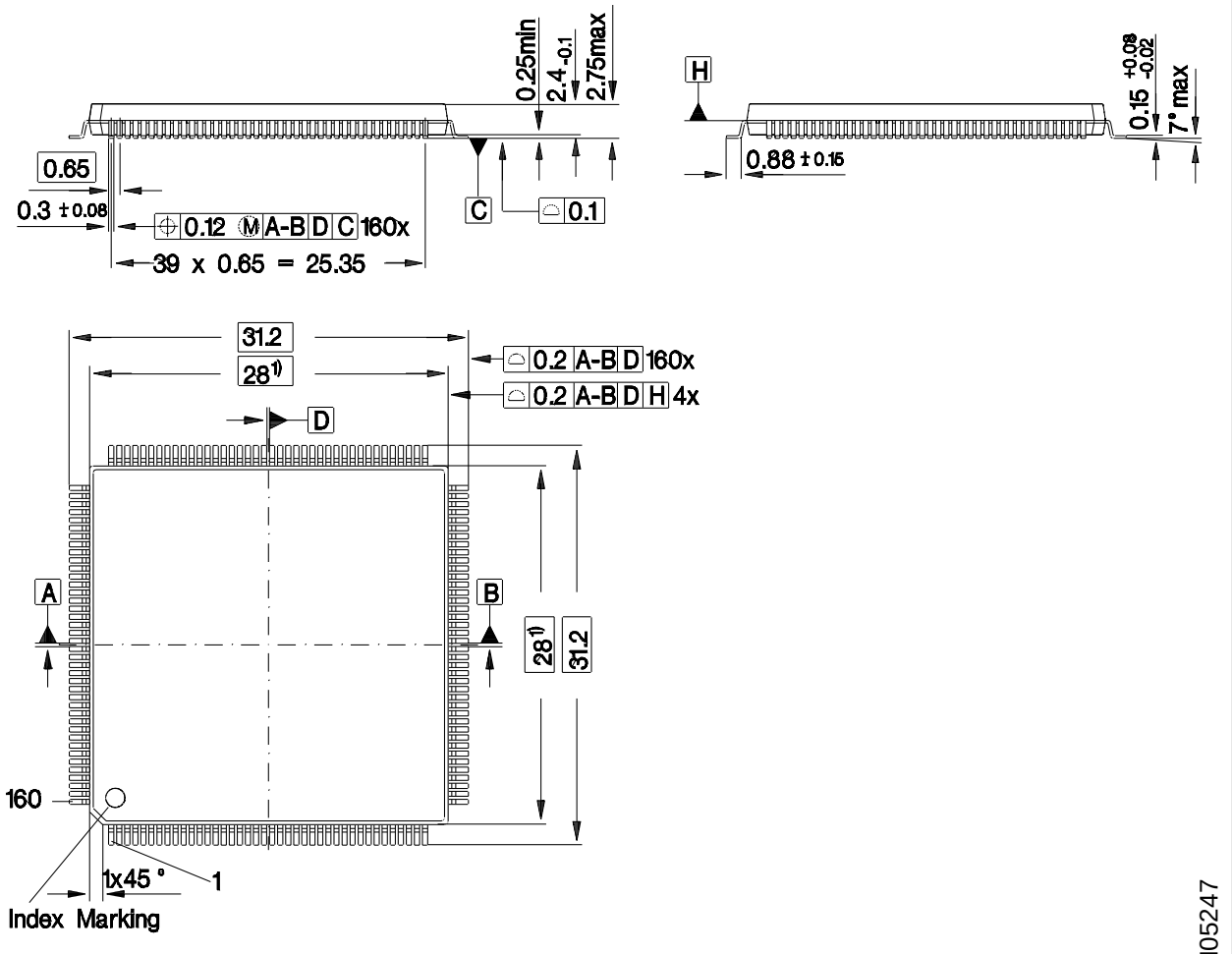
5.4.4 Reset Timing

Reset Timing

Parameter	No.	Symbol	Limit Values				Unit
			H		H-10		
			min.	max.	min.	max.	
Res pulse width		$t_{w(RES)}$	5000		5000		ns

6 Package Outlines

Plastic Package, P-MQFP-160 (SMD)
(Plastic Metric Quad Flat Package)



1) Does not include plastic or metal protrusion of 0.25 max. per side

Sorts of Packing

Package outlines for tubes, trays etc. are contained in our Data Book “Package Information”

SMD = Surface Mounted Device

Dimensions in mm

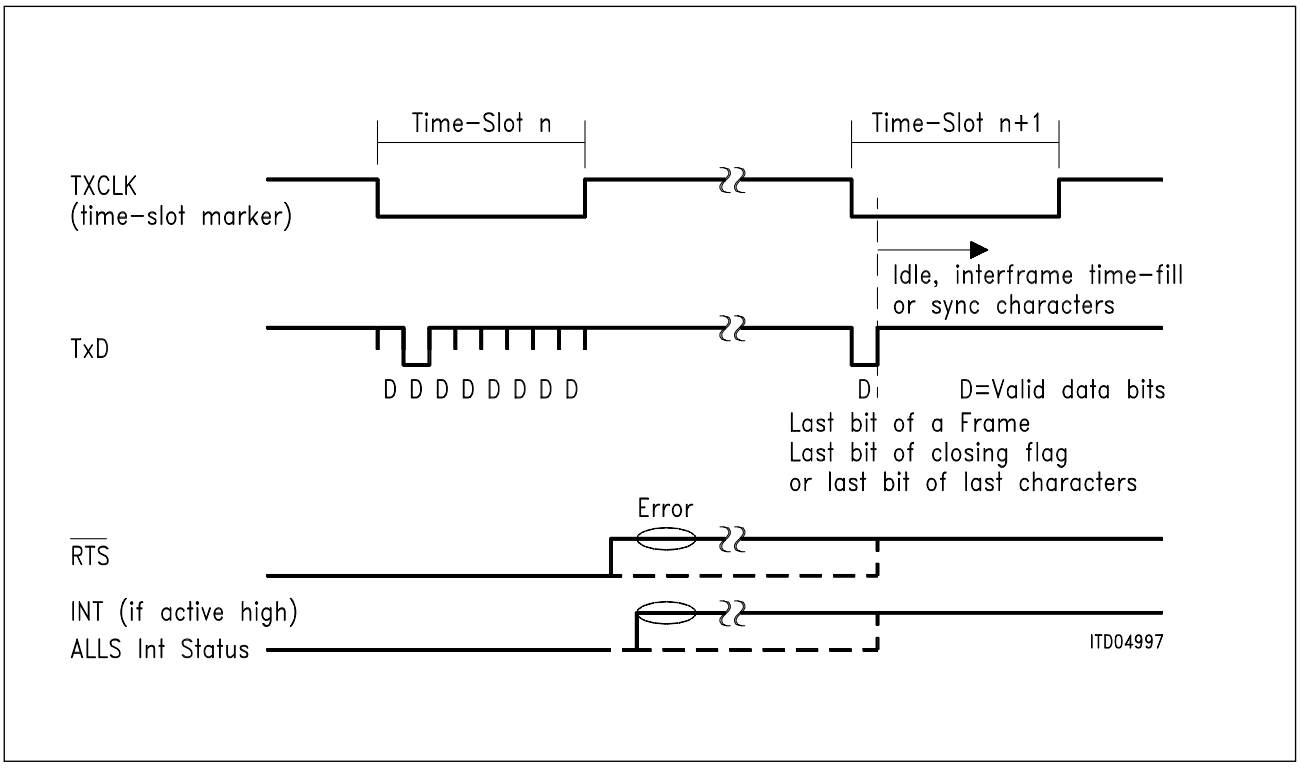
7 Appendix

Errata Sheet SAB 82538H, Version V2.2

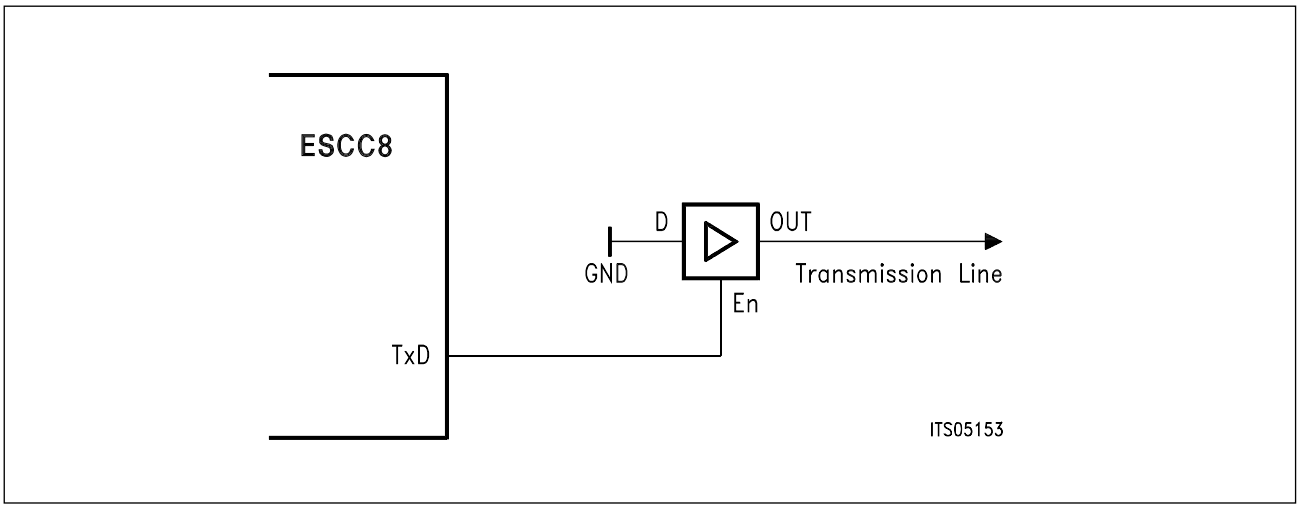
1. General

The following errata should be noted when version 2.2 of the ESCC8 (SAB 82538) is used.

a. In clock mode the $\overline{\text{RTS}}$ signal is deactivated after the transmission of the **second** last bit (instead of the last) of a closing flag (or the last character of a block of characters), if that second last bit is the last bit of the a time-slot “window”. In other words, $\overline{\text{RTS}}$ is inactive during the transmission of the last bit, transmitted in the next time-slot window. See figure below.
Furthermore, the ALLS (All Sent) interrupt status is generated after the transmission of the second last bit, one clock period after the deactivation of $\overline{\text{RTS}}$.



Recommendation: Do not use $\overline{\text{RTS}}$ in clock mode 5 e.g. to enable drivers for TxD in a bus configuration. (For example, use an arrangement of the type shown in the figure below instead.)



b. The maximum achievable bit rates in **internal timing modes** (where bit timing is extracted from the data by the ESCC8 by means of the internal DPLL) are:
1.2 Mbit/s when RxCLK is used as clock source for the baud rate generator (clock modes 2, 3).
0.8 Mbit/s when XTAL1 (1, 2) is used to supply the clock source for the baud rate generator (clock modes 6, 7).