

ARTISAN®
TECHNOLOGY GROUP

Your **definitive** source for quality pre-owned equipment.

Artisan Technology Group
(217) 352-9330 | sales@artisantg.com | artisantg.com

Full-service, independent repair center
with experienced engineers and technicians on staff.

We buy your excess, underutilized, and idle equipment
along with credit for buybacks and trade-ins.

Custom engineering
so your equipment works exactly as you specify.

- Critical and expedited services
- In stock / Ready-to-ship
- Leasing / Rentals / Demos
- ITAR-certified secure asset solutions

Expert team | Trust guarantee | 100% satisfaction

All trademarks, brand names, and brands appearing herein are the property of their respective owners.

Visit our website - *Click HERE*

FASTCOM®

FSCC

AND

SUPERFSCC

FAMILY OF SYNCHRONOUS ADAPTERS







COMMTECH, INC.
9011 E 37th St N
Wichita KS USA
67226-2006

COMMTECH

9011 E. 37TH STREET N
WICHITA, KANSAS 67226-2006
(316) 636-1131
FAX (316) 636-1163
http://www.fastcomproducts.com/

COPYRIGHT © 2020

FASTCOM and the “Alpha Lemur” are Registered Trademarks of Commtech, Inc.

REVISION NOTES

REVISION	PAGE NUMBER	CHANGES MADE
1.0		Document Created
1.1	8, 13	Added: Fastcom: SuperFSCC-PCI-104 Fastcom: GSuperFSCC-PCI-104 Fastcom: SuperFSCC-PCI-104-LVDS Fastcom: GSuperFSCC-PCI-104-LVDS
1.2	77	Added DSYNC option to the CCR1 register
1.3	9	Updated Declaration of Conformity
1.4	9	Updated Declaration of Conformity
1.5		Updated various links
1.6	23	Updated testing procedures.
1.7	27	Added a document about termination by Texas Instruments

TABLE OF CONTENTS

FSCC/*SUPER*FSCC PRODUCT FAMILY....8

EUROPEAN UNION DECLARATION OF CONFORMITY....9

1 FASTCOM: FSCC/*SUPERFSCC*....10

1.1 FEATURES 10
1.2 BOARD DRAWINGS 11
1.2.1 2-Port Universal PCI Bus 11
1.2.2 2-Port Compact PCI Bus 12
1.2.3 2-Port PC/104+ and PCI-104 Bus 13
1.2.4 4-Port Universal PCI Bus 14
1.2.5 4-Port PCI Express x1 15
1.3 PACKING LIST 16
1.4 PINOUTS 16
1.4.1 DB62 Female (board edge) – 2-Port Universal PCI & Compact PCI Bus 16
1.4.2 CHAMP-50 (board edge) – 4-Port Universal PCI & PCI Express x1 Bus 17
1.4.4 DB25 Male (cable) – All 2-Port & 4-Port Cards 18
1.4.5 2/25 I/O Header (board edge) – 2-Port PC/104+ and PCI-104 Bus 21
1.4.6 DB62 Loop back connector 23
1.4.7 DB25 Loop back connector 23
1.5 INSTALLATION 24
1.5.1 Hardware Installation 24
1.5.2 Software Installation (Windows) 24
1.5.3 Testing the Installation: 25
1.6 RS422 / RS485 26
1.7 TERMINATION RESISTANCE 26
1.8 RS-485 MODE 28
1.9 SUPERFSCC DIFFERENCE FROM STANDARD FSCC 28

2 FASTCOM F-CORE 29

2.1 INTRODUCTION 29
2.2 HARDWARE 29
2.3 SERIAL INTERFACE 29
2.4 DATA PROTOCOLS 31

3 DATA FLOW 32

3.1 RECEIVE 32
3.1.1 Receive Interrupts 32
3.2 TRANSMIT 33
3.2.1 Transmit Modes 33
3.2.2 Transmit Interrupts 34

4 GLOBAL FEATURES DETAILS.....35

4.1 CLOCKING SYSTEM 35
4.1.1 Onboard Clock Generators..... 35
4.1.2 Clock Modes..... 36
4.1.3 Baud Rates 38
4.1.4 Clock Recovery (DPLL)..... 38
4.2 DATA ENCODING 42
4.2.1 NRZ and NRZI Encoding 42
4.2.2 FM0 and FM1 Encoding..... 42
4.2.3 Manchester Encoding 43
4.2.4 Differential Manchester Encoding..... 43
4.3 RECEIVE STATUS WORD 44
4.4 RECEIVE LENGTH CHECK..... 44
4.5 FRAME SYNC SIGNALS..... 44
4.6 MODEM CONTROL SIGNALS..... 46
4.6.1 Request to send (RTS) 46
4.6.2 Clear to send (CTS) 46
4.6.3 DTR..... 47
4.6.4 DSR, CD, RI..... 47
4.7 PREAMBLES AND POSTAMBLES..... 47
4.8 ORDER OF BIT TRANSMISSION 47
4.9 INTERFRAME TIME FILL 47
4.10 CRC FRAME CHECKING SEQUENCES 48
4.10.1 8-bit frame checking sequence (CRC-8)..... 48
4.10.2 16-Bit frame checking sequence 49
4.10.3 32-bit frame checking sequence (CRC-32)..... 49
4.11 DATA AND CLOCK INVERSION 50
4.12 HARDWARE TIMER 50
4.13 TRANSMIT ON TIMER 50
4.14 ZERO BIT INSERTION..... 51
4.15 ONE BIT INSERTION 51

5 DETAILED PROTOCOL DESCRIPTION.....52

5.1 HDLC/SDLC 52
5.1.1 General Frame Format..... 52
5.1.2 Elements of the Frame 52
5.1.3 Data Reception..... 53
5.1.4 Data Transmission..... 54
5.1.5 Shared Flag Mode..... 54
5.1.6 Invalid Frame..... 54
5.1.7 Aborted Frame..... 55
5.2 CHARACTER ORIENTED SYNCHRONOUS (X-SYNC) MODE 56
5.2.1 General Frame Format..... 56
5.2.2 Elements of the Frame 56
5.2.3 Data Reception..... 57
5.2.4 Data Transmission..... 57

5.2.5 Shared Flag Mode *58*
5.3 TRANSPARENT MODE 59
5.3.1 Data Reception *59*
5.3.2 Data Transmission *59*
5.4 TRUE-ASYNCHRONOUS MODE 60
5.4.1 Features *60*

6 REGISTER DESCRIPTION 61

6.1 BASE ADDRESS REGISTERS 61
6.2 F-CORE REGISTERS 61
6.2.1 FIFO locations (0x00/0x80) *62*
6.2.2 Byte Count FIFO locations (0x04/0x84) *62*
6.2.3 FIFO Trigger Levels (0x08/0x88) *62*
6.2.4 FIFO Byte Count (0x0C/0x8C) *63*
6.2.5 FIFO Frame Count (0x10/0x90) *64*
6.2.6 CMDR – Command Register (0x14/0x94) *65*
6.2.7 STAR – Status Register (0x18/0x98) *67*
6.2.8 CCR0 – Channel Configuration Register 0 (0x1C/0x9C) *71*
6.2.9 CCR1 – Channel Configuration Register 1 (0x20/0xA0) *75*
6.2.10 CCR2 – Channel Configuration Register 2 (0x24/0xA4) *79*
6.2.11 BGR – Baud-Rate Generator Register (0x28/0xA8) *81*
6.2.12 SSR – Synchronization Sequence Register (0x2C/0xAC) *82*
6.2.13 SMR – Synchronization Mask Register (0x30/0xB0) *83*
6.2.14 TSR – Termination Sequence Register (0x34/0xB4) *84*
6.2.15 TMR – Termination Mask Register (0x38/0xB8) *85*
6.2.16 RAR – Receive Address Register (0x3C/0xBC) *86*
6.2.17 RAMR – Receive Address Mask Register (0x40/0xC0) *87*
6.2.18 PPR – Preamble & Postamble Register (0x44/0xC4) *88*
6.2.19 TCR – Timer Control Register (0x48/0xC8) *89*
6.2.20 VSTR – Version Status Register (0x4C/0xCC) *90*
6.2.21 ISR – Interrupt Status Register (0x50/0xD0) *91*
6.2.22 IMR – Interrupt Mask Register (0x54/0xD4) *94*
6.2.23 DPLLR – DPLL Reset control register (0x58/0xD8) *95*
6.3 16C950 BASED REGISTERS 96
6.4 BOARD CONTROL 101
6.4.1 FCR – Feature Control Register (0x00) *102*
6.4.2 DMACCR - DMA Command / Control Register (0x04/0x08) *104*
6.4.3 DMA Receive Descriptor Base (0x0c/0x14) *106*
6.4.4 DMA Transmit Descriptor Base (0x10/0x18) *106*
6.4.5 DMA Current Receive Descriptor Base (0x20/0x28) *106*
6.4.6 DMA Current Transmit Descriptor Base (0x24/0x2c) *106*
6.4.7 DSTAR – DMA Status Register (0x30) *107*
6.5 DMA DESCRIPTOR LAYOUT 108
6.5.1 Descriptor Control Register (0x00) *109*
6.5.2 DataAddress (0x04) *110*
6.5.3 DataCount (0x08) *110*

6.5.4 NextDescriptor (0x0c)..... 110

7 TECHNICAL SUPPORT.....111

8 FASTCOM LIMITED LIFETIME WARRANTY.....112

8.1 WARRANTY EXCLUSIONS 112
8.2 NON-WARRANTY REPAIRS 112
8.3 LIMITATION OF LIABILITY..... 112

FSCC/*Super*FSCC Product Family

FSCC/SuperFSCC Product	Number of Ports	RoHS	CE	CE Compliant (Except for RoHS)
PCI Express Bus				
Fastcom: FSCC/4-PCIe	4			√
Fastcom: GFSCC/4-PCIe	4	√	√	
Fastcom: SuperFSCC/4-PCIe	4			√
Fastcom: GSuperFSCC/4-PCIe	4	√	√	
Fastcom: SuperFSCC/4-PCIe-LVDS	4			
Fastcom: GSuperFSCC/4-PCIe-LVDS	4	√		
Universal PCI Bus				
Fastcom: FSCC	2			√
Fastcom: GFSCC	2	√	√	
Fastcom: SuperFSCC	2			√
Fastcom: GSuperFSCC	2	√	√	
Fastcom: SuperFSCC-LVDS	2			
Fastcom: GSuperFSCC-LVDS	2	√		
Fastcom: FSCC/4	4			√
Fastcom: GFSCC/4	4	√	√	
Fastcom: SuperFSCC/4	4			√
Fastcom: GSuperFSCC/4	4	√	√	
Fastcom: SuperFSCC/4-LVDS	4			
Fastcom: GSuperFSCC/4-LVDS	4	√		
Compact PCI Bus				
Fastcom: FSCC-cPCI	2			
Fastcom: GFSCC-cPCI	2	√		
Fastcom: SuperFSCC-cPCI	2			
Fastcom: GSuperFSCC-cPCI	2	√		
Fastcom: SuperFSCC-cPCI-LVDS	2			
Fastcom: GSuperFSCC-cPCI-LVDS	2	√		
PCI-104 Bus				
Fastcom: SuperFSCC-PCI-104	2			
Fastcom: GSuperFSCC-PCI-104	2	√		
Fastcom: SuperFSCC-PCI-104-LVDS	2			
Fastcom: GSuperFSCC-PCI-104-LVDS	2	√		
PC/104+ Bus				
Fastcom: SuperFSCC-104	2			
Fastcom: GSuperFSCC-104	2	√		
Fastcom: SuperFSCC-104-LVDS	2			
Fastcom: GSuperFSCC-104-LVDS	2	√		

COMMTECH™
COMMUNICATION • TECHNOLOGIES

EUROPEAN UNION DECLARATION OF CONFORMITY

Information Technology Equipment

The Company COMMTECH, INC. declares under its own and full responsibility that the product

" Fastcom: *FSCC/SuperFSCC* - Revision 3.4 "

on which is attached this Certificate is compliant to the EU Directive 2014/30/EU.

[ ] The product identified above complies with the requirements of the above EU Directive by meeting the following standards:

Emissions Test

The product family was tested for compliance to Directive 2014/30/EU of the European Parliament relating to electromagnetic compatibility using the following harmonized standards:

EN 55022:2010/A1:2011, Class A
EN 55032:2012/AC:2013, Class A
US CFR, FCC Part 15.109
Industry Canada ICES-003. Issue 6

Radiated Emissions	Passed
Conducted Emissions	Passed

Immunity Test

The product family was tested for compliance to EN 61326-1:2013 and EN 55024:2010.

Electrostatic Discharge	IEC 61000-4-2:2008; 4 kV/8 kV Contact/Air	Passed
Radiated Immunity	IEC 61000-4-3:2006/A1:2007/A2:2010; 2sec/80% AM, 1kHz/10 V/m	Passed
EFT – AC	IEC 61000-4-4:2004/A1:2010; 2 kV	Passed
EFT – Signal Cables	IEC 61000-4-4:2004/A1:2010; 1 kV	Passed
Surge	IEC 61000-4-5:2005; 1 kV L-L /2 kV L-PE	Passed
Conducted Immunity – AC	IEC 61000-4-6:2006; 3 V_{rms}	Passed
Conducted Immunity – Signal Cables	IEC 61000-4-6:2006; 3 V_{rms}	Passed
Voltage Interrupt	IEC 61000-4-11:2004; 250 Cycle/100%	Passed
Voltage Dip	IEC 61000-4-11:2004; 0.5, 1, 10, 25 Cycle/100, 100, 40, 70%	Passed

The technical documentation required to demonstrate that this product meets the requirements of the EU Directive has been compiled by the signatory below and is available for inspection by the relevant enforcement authorities.

In WICHITA, KS on March 13th, 2018

COMMTECH™
COMMUNICATION • TECHNOLOGIES
9011 E. 37th Street North
Wichita, KS 67226-2006
(316) 636-1131
Fax (316) 636-1163

Mr. Glen R. Alvis, Test Engineer

CE

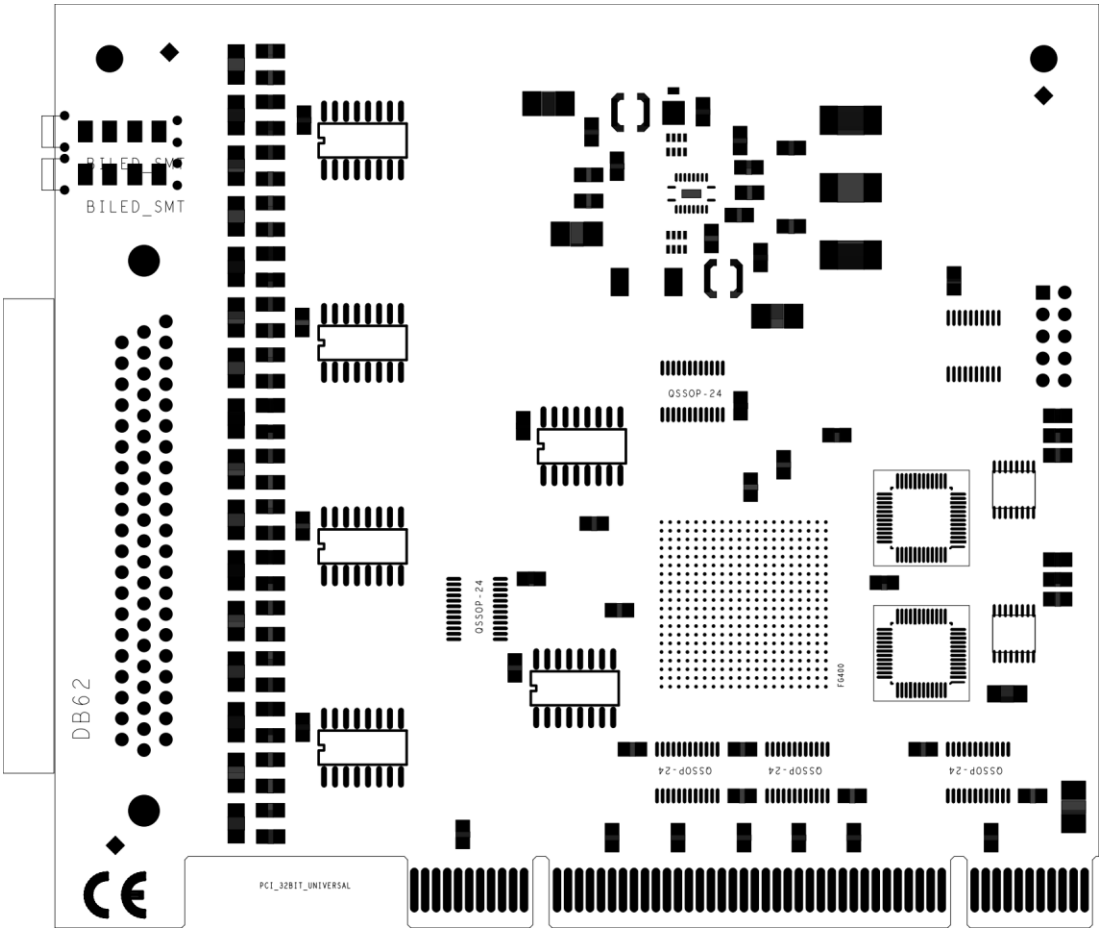
1 Fastcom: FSCC/*SuperFSCC*

1.1 Features

- High speed synchronous / asynchronous, full duplex, RS-422 / RS-485 ports
- Multiple bus interfaces
    - Universal PCI (PCI version 2.3) adapter option that operates in both 5V and 3.3V PCI slots. This means the card can work in the high speed PCI-X slots as well as standard PCI slots.
    - PCI Express x1 (PCI Express Base Specification, Revision 2.0).
    - Universal PC/104+ and PCI-104 (PCI version 2.2) adapter will operate in both 5V and 3.3V PC/104+ and PCI-104 stacks.
    - Universal cPCI (PICMG 2.0 Rev 2.1) adapter operates in both 5V and 3.3V PCI slots.
- Each port has an independent clock generator allowing each to operate at completely different baud rates.
- Each port can be operated in True-Asynchronous mode utilizing the 16C950 based UARTS. When in True-Asynchronous mode, the ports can be operated just like a standard serial (COM) port. See the True-Asynchronous section for more details.
- Automatic RS-485 handling (e.g. tri-state the driver when idle)
- Receive echo cancel. Disables the line receiver during transmits to prevent receiving the data that is transmitted in 2-wire 485 modes.
- Transmit and Receive status LEDs for each port.
- On board RS-485 termination network for each receive signal pair
    - 100 Ω termination
    - 1 kΩ biasing resistors (pull-up and pull-down)
- High speed data rates
    - SuperFSCC family (Has **D**irect **M**emory **A**ccess capability)
        1. 50 MHz (internal clock)
        2. 30 MHz (external clock)
        3. 12.5 MHz (DPLL)
    - FSCC family (Does not have DMA capability)
        1. 20 MHz (internal clock)
        2. 20 MHz (internal clock)
        3. 12.5 MHz (DPLL)

1.2 Board Drawings

1.2.1 2-Port Universal PCI Bus



1.2.1.1 Board variations

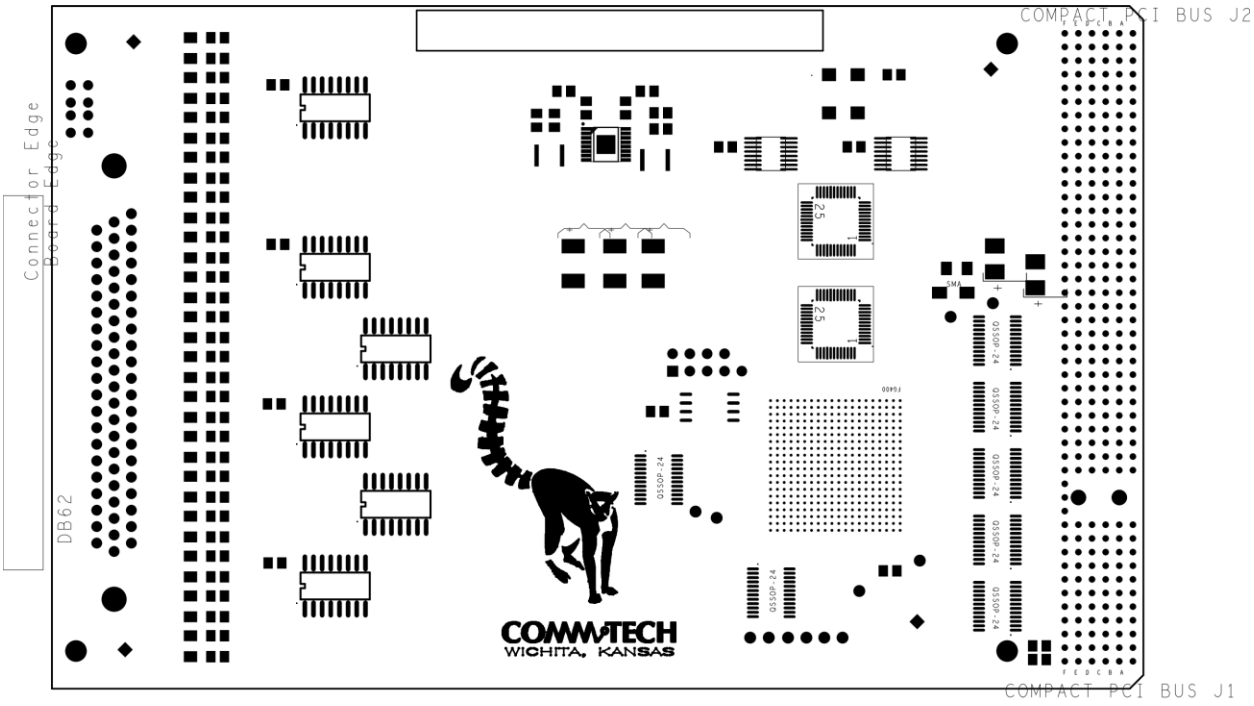
Non-RoHS

- Fastcom: FSCC
- Fastcom: SuperFSCC
- Fastcom: SuperFSCC-LVDS

RoHS

- Fastcom: GFSCC
- Fastcom: GSuperFSCC
- Fastcom: GSuperFSCC-LVDS

1.2.2 2-Port Compact PCI Bus



1.2.2.1 Board Variations

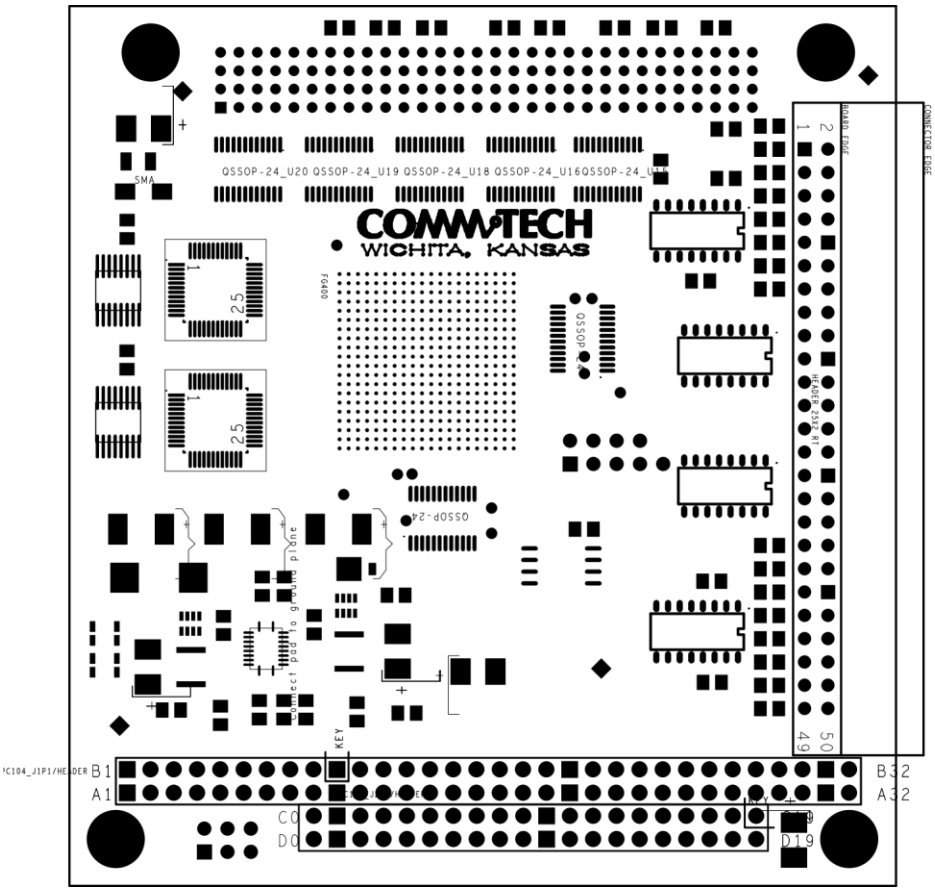
Non-RoHS

- Fastcom: FSCC-cPCI
- Fastcom: SuperFSCC-cPCI
- Fastcom: SuperFSCC-cPCI-LVDS

RoHS

- Fastcom: GFSCC-cPCI
- Fastcom: GSuperFSCC-cPCI
- Fastcom: GSuperFSCC-cPCI-LVDS

1.2.3 2-Port PC/104+ and PCI-104 Bus



1.2.3.1 Board Variations

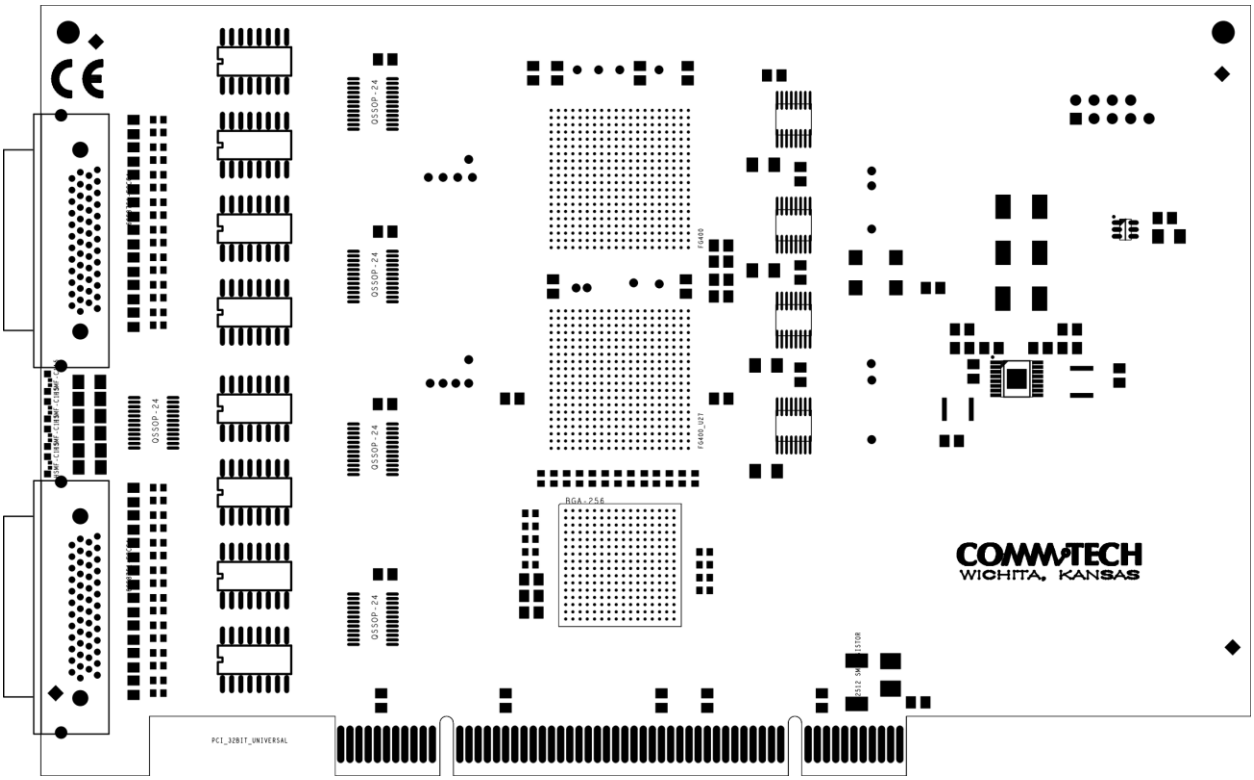
Non-RoHS

- Fastcom: SuperFSCC-104 (Extended Temp)
- Fastcom: SuperFSCC-104-LVDS (Extended Temp)
- Fastcom: SuperFSCC-PCI-104 (Extended Temp)
- Fastcom: SuperFSCC-PCI-104-LVDS (Extended Temp)

RoHS

- Fastcom: GSuperFSCC-104 (Extended Temp)
- Fastcom: GSuperFSCC-104-LVDS (Extended Temp)
- Fastcom: GSuperFSCC-PCI-104 (Extended Temp)
- Fastcom: GSuperFSCC-PCI-104-LVDS (Extended Temp)

1.2.4 4-Port Universal PCI Bus



1.2.4.1 Board Variations

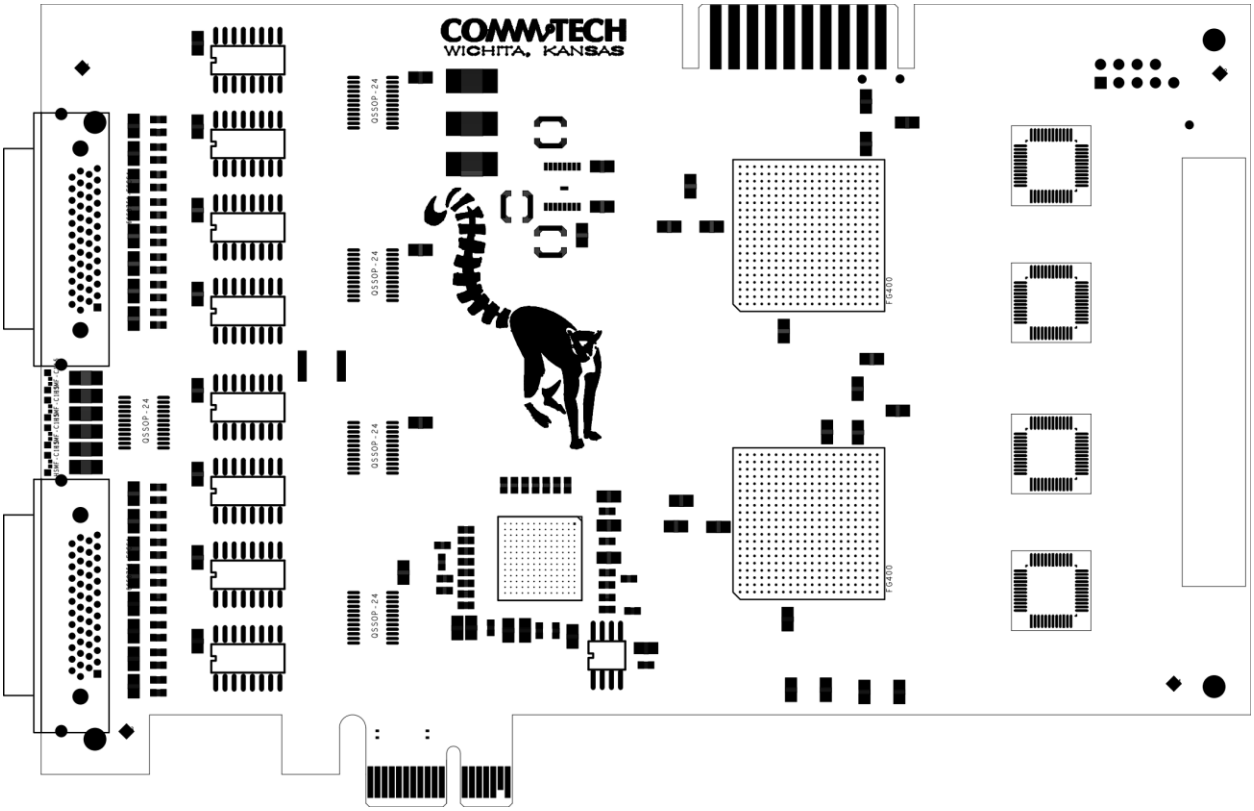
Non-RoHS

- Fastcom: FSCC/4
- Fastcom: SuperFSCC/4
- Fastcom: SuperFSCC/4-LVDS

RoHS

- Fastcom: GFSCC/4
- Fastcom: GSuperFSCC/4
- Fastcom: GSuperFSCC/4-LVDS

1.2.5 4-Port PCI Express x1



1.2.5.1 Board Variations

Non-RoHS

- Fastcom: FSCC/4-PCIe
- Fastcom: SuperFSCC/4-PCIe
- Fastcom: SuperFSCC/4-PCIe-LVDS

RoHS

- Fastcom: GFSCC/4-PCIe
- Fastcom: GSuperFSCC/4-PCIe
- Fastcom: GSuperFSCC/4-PCIe-LVDS

1.3 Packing List

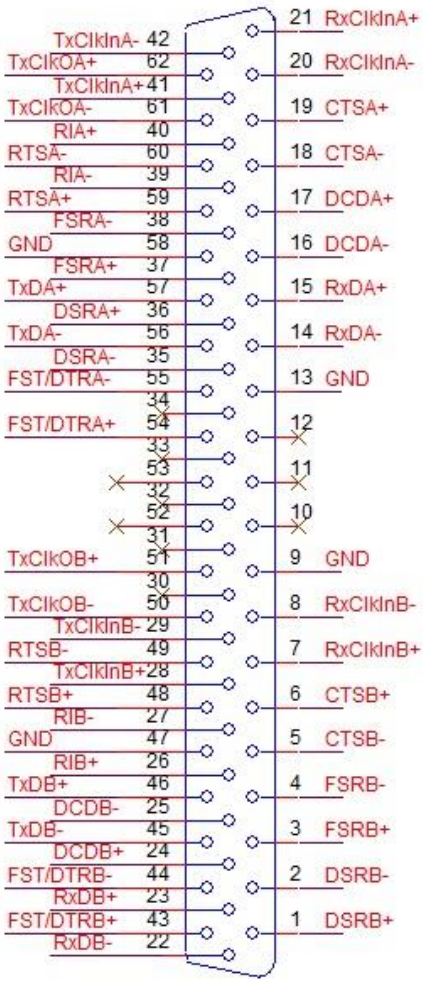
- Fastcom: FSCC/SuperFSCC card
- FSCC/SuperFSCC cable assembly
- Loop back plug[1]
- Fastcom CD

If an omission has been made, please contact technical support for a replacement.

1.4 Pinouts

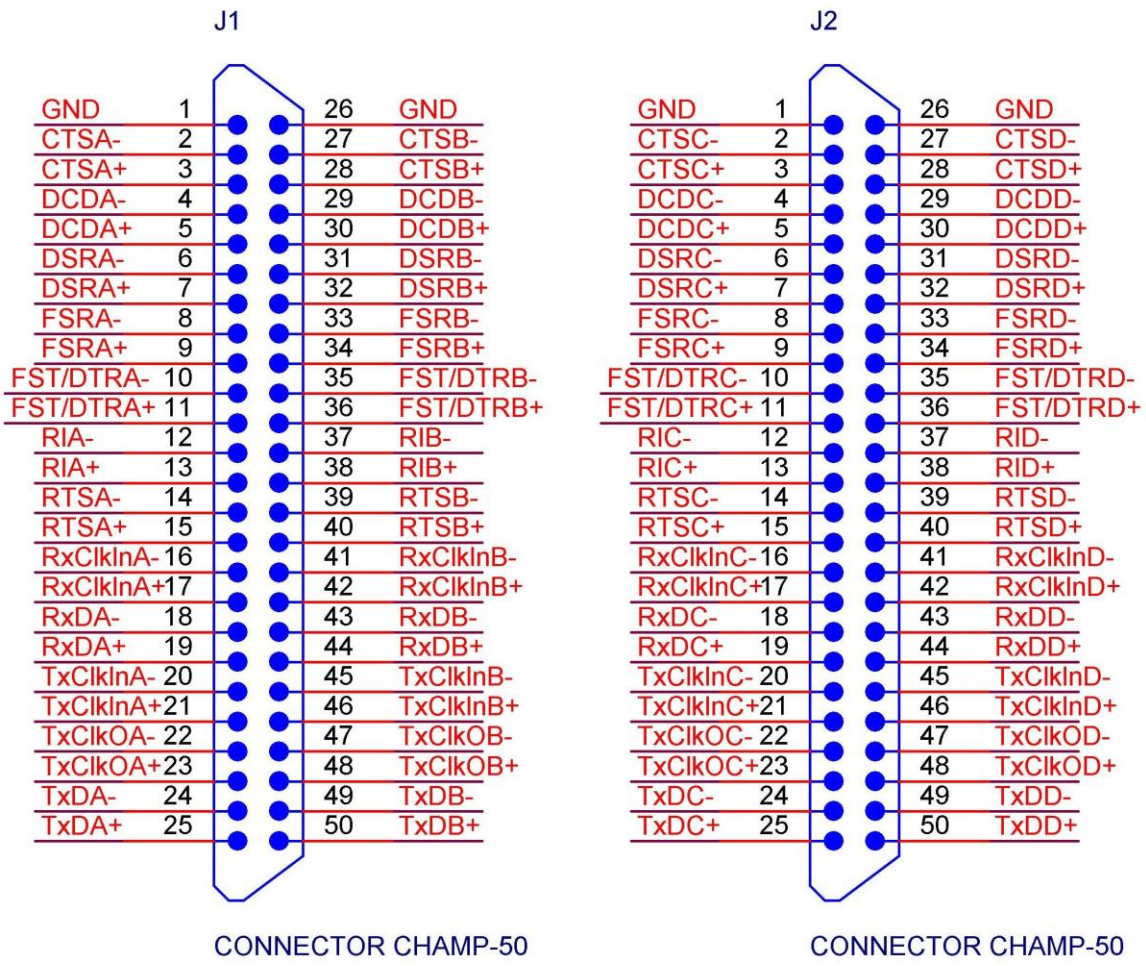
We provide access to each channel of the FASTCOM: *FSCC/SuperFSCC* through a shielded connector and an adapter cable (supplied with the board). The adapter cable is manufactured with high quality twisted pairs for improved signal performance and noise immunity.

1.4.1 DB62 Female (board edge) – 2-Port Universal PCI & Compact PCI Bus

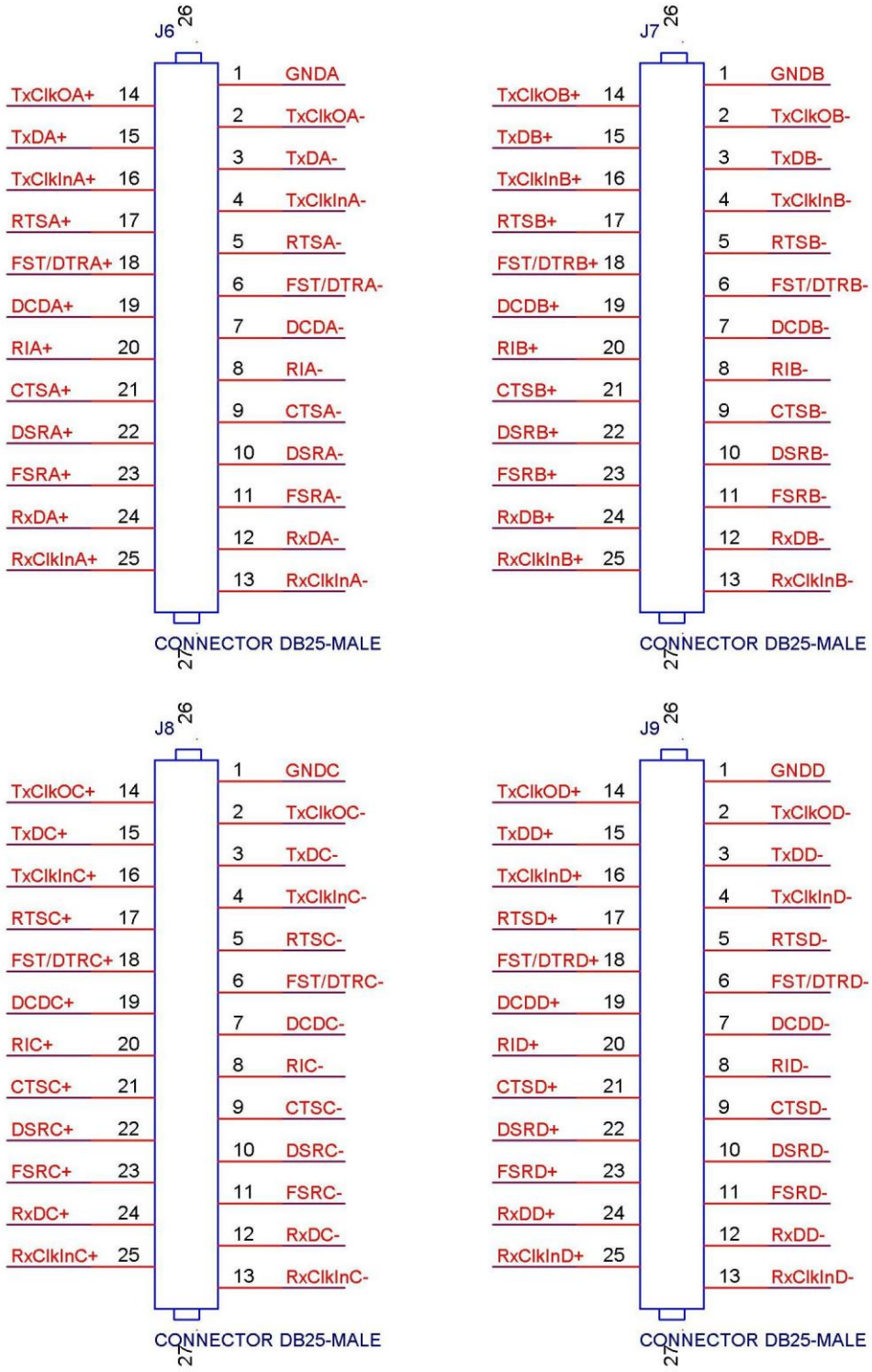


[1] A loop back plug is included with the PC/104+ boards only if the optional cable is ordered

1.4.2 CHAMP-50 (board edge) – 4-Port Universal PCI & PCI Express x1 Bus



1.4.4 DB25 Male (cable) – All 2-Port & 4-Port Cards



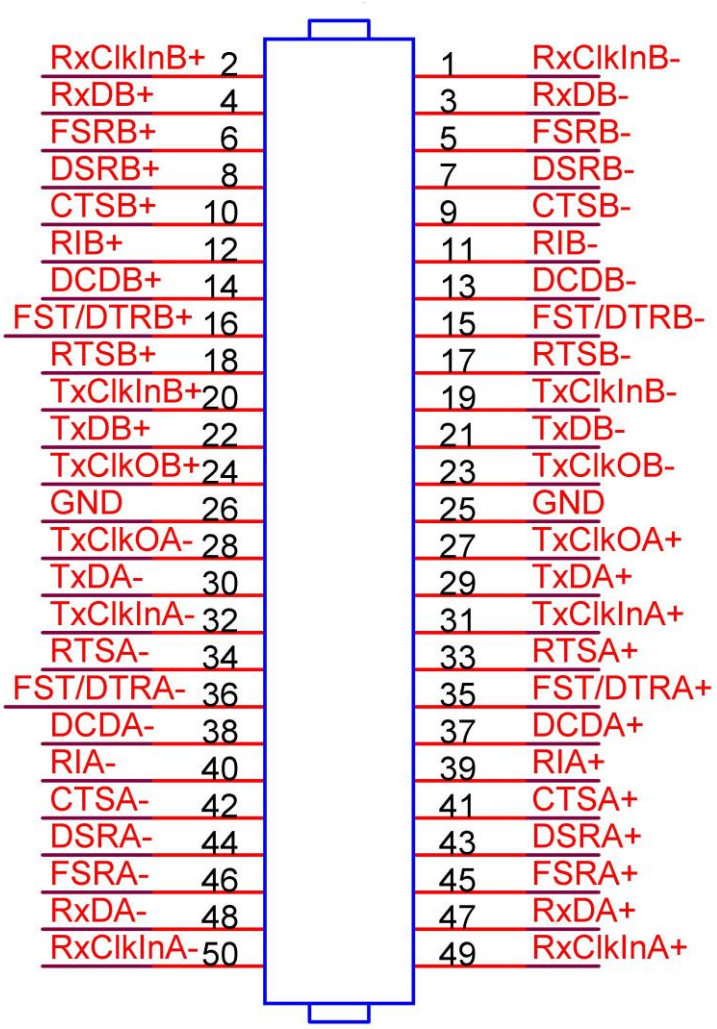
1.4.4.1 DB62 to DB25 Cable Cross-reference – 2-Port Universal PCI & Compact PCI Bus

SIGNAL(A)	DB62M	DB25M	Description	Direction	SIGNAL(B)	DB62M
GND	13,58	1	Ground		GND	9,47
TxClkOA-	61	2	Transmit Clock Out -	Output	TxClkOB-	50
TxDA-	56	3	Transmit Data -	Output	TxDB-	45
TxClkInA-	42	4	Transmit Click In -	Input	TxClkInB-	29
RTSA-	60	5	Request To Send -	Output	RTSB-	49
FST/DTRA-	55	6	Frame Sync Transmit -	Output	FST/DTRB-	44
DCDA-	16	7	Carrier Detect -	Input	DCDB-	25
RIA-	39	8	Ring Indicator -	Input	RIB-	27
CTSA-	18	9	Clear To Send -	Input	CTSB-	5
DSRA-	35	10	Data Set Ready -	Input	DSRB-	2
FSRA-	38	11	Frame Sync Receive -	Input	FSRB-	4
RxDA-	14	12	Receive Data -	Input	RxDB-	22
RxClkInA-	20	13	Receive Clock In -	Input	RxClkInB-	8
TxClkOA+	62	14	Transmit Clock Out +	Output	TxClkOB+	51
TXDA+	57	15	Transmit Data +	Output	TXDB+	46
TxClkInA+	41	16	Transmit Clock In +	Input	TxClkInB+	28
RTSA+	59	17	Request To Send +	Output	RTSB+	48
FST/DTRA+	54	18	Frame Sync Transmit +	Output	FST/DTRB+	43
DCDA+	17	19	Carrier Detect +	Input	DCDB+	24
RIA+	40	20	Ring Indicator +	Input	RIB+	26
CTSA+	19	21	Clear To Send +	Input	CTSB+	6
DSRA+	36	22	Data Set Ready +	Input	DSRB+	1
FSRA+	37	23	Frame Sync Receive +	Input	FSRB+	3
RxDA+	15	24	Receive Data +	Input	RxDB+	23
RxClkInA+	21	25	Receive Clock In +	Input	RxClkInB+	7

1.4.4.2 CHAMP-50 to DB25 Cable Cross-reference – 4-Port Universal PCI & PCI Express x1 Bus

SIGNAL(A)	CHAMP50	DB25	Description	Direction	SIGNAL(B)	CHAMP50
GND	1	1	Ground		GND	26
TxClkOA-	22	2	Transmit Clock Out -	Output	TxClkOB-	47
TxDA-	24	3	Transmit Data -	Output	TxDB-	49
TxClkInA-	20	4	Transmit Click In -	Input	TxClkInB-	45
RTSA-	14	5	Request To Send -	Output	RTSB-	39
FST/DTRA-	10	6	Frame Sync Transmit -	Output	FST/DTRB-	35
DCDA-	4	7	Carrier Detect -	Input	DCDB-	29
RIA-	12	8	Ring Indicator -	Input	RIB-	37
CTSA-	2	9	Clear To Send -	Input	CTSB-	27
DSRA-	6	10	Data Set Ready -	Input	DSRB-	31
FSRA-	8	11	Frame Sync Receive -	Input	FSRB-	33
RxDA-	18	12	Receive Data -	Input	RxDB-	43
RxClkInA-	16	13	Receive Clock In -	Input	RxClkInB-	41
TxClkOA+	23	14	Transmit Clock Out +	Output	TxClkOB+	48
TxDA+	25	15	Transmit Data +	Output	TxDB+	50
TxClkInA+	21	16	Transmit Clock In +	Input	TxClkInB+	46
RTSA+	15	17	Request To Send +	Output	RTSB+	40
FST/DTRA+	11	18	Frame Sync Transmit +	Output	FST/DTRB+	36
DCDA+	5	19	Carrier Detect +	Input	DCDB+	30
RIA+	13	20	Ring Indicator +	Input	RIB+	38
CTSA+	3	21	Clear To Send +	Input	CTSB+	28
DSRA+	7	22	Data Set Ready +	Input	DSRB+	32
FSRA+	9	23	Frame Sync Receive +	Input	FSRB+	34
RxDA+	19	24	Receive Data +	Input	RxDB+	44
RxClkInA+	17	25	Receive Clock In +	Input	RxClkInB+	42

1.4.5 2/25 I/O Header (board edge) – 2-Port PC/104+ and PCI-104 Bus



1.4.5.1 2x25 to DB25 Cable Cross-reference – 2-Port PC/104+ and PCI-104 Bus

SIGNAL(A)	2x25	DB25M	Description	Direction	SIGNAL(B)	2x25
GND	25	1	Ground		GND	26
TxClkOA-	28	2	Transmit Clock Out -	Output	TxClkOB-	23
TxDA-	30	3	Transmit Data -	Output	TxDB-	21
TxClkInA-	32	4	Transmit Click In -	Input	TxClkInB-	19
RTSA-	34	5	Request To Send -	Output	RTSB-	17
FST/DTRA-	36	6	Frame Sync Transmit -	Output	FST/DTRB-	15
DCDA-	38	7	Carrier Detect -	Input	DCDB-	13
RIA-	40	8	Ring Indicator -	Input	RIB-	11
CTSA-	42	9	Clear To Send -	Input	CTSB-	9
DSRA-	44	10	Data Set Ready -	Input	DSRB-	7
FSRA-	46	11	Frame Sync Receive -	Input	FSRB-	5
RxDA-	48	12	Receive Data -	Input	RxDB-	3
RxClkInA-	50	13	Receive Clock In -	Input	RxClkInB-	1
TxClkOA+	27	14	Transmit Clock Out +	Output	TxClkOB+	24
TXDA+	29	15	Transmit Data +	Output	TXDB+	22
TxClkInA+	31	16	Transmit Clock In +	Input	TxClkInB+	20
RTSA+	33	17	Request To Send +	Output	RTSB+	18
FST/DTRA+	35	18	Frame Sync Transmit +	Output	FST/DTRB+	16
DCDA+	37	19	Carrier Detect +	Input	DCDB+	14
RIA+	39	20	Ring Indicator +	Input	RIB+	12
CTSA+	41	21	Clear To Send +	Input	CTSB+	10
DSRA+	43	22	Data Set Ready +	Input	DSRB+	8
FSRA+	45	23	Frame Sync Receive +	Input	FSRB+	6
RxDA+	47	24	Receive Data +	Input	RxDB+	4
RxClkInA+	49	25	Receive Clock In +	Input	RxClkInB+	2

1.4.5.2 2x25 Loop back connector (not included)

SIGNAL	DB62M	SIGNAL	DB62M
TxClkOA+	62	RxClkIA+	21
TxClkOA-	61	RxClkIA-	20
TxDA+	57	RxDA+	15
TxDA-	56	RxDA-	14
RTSA+	59	CTSA+	19
RTSA-	60	CTSA-	18
FSTA+	54	FSRA+	37
FSTA-	55	FSRA-	38
TxClkOB+	51	RxClkIB+	7
TxClkOB-	50	RxClkIB-	8
TxDB+	46	RxDB+	23
TxDB-	45	RxDB-	22
RTSB+	48	CTSB+	6
RTSB-	49	CTSB-	5
FSTB+	43	FSRB+	3
FSTB-	44	FSRB-	4

1.4.6 DB62 Loop back connector

SIGNAL	DB62M	SIGNAL	DB62M
TxClkOA+	62	RxClkIA+	21
TxClkOA-	61	RxClkIA-	20
TxDA+	57	RxDA+	15
TxDA-	56	RxDA-	14
RTSA+	59	CTSA+	19
RTSA-	60	CTSA-	18
FSTA+	54	FSRA+	37
FSTA-	55	FSRA-	38
TxClkOB+	51	RxClkIB+	7
TxClkOB-	50	RxClkIB-	8
TxDB+	46	RxDB+	23
TxDB-	45	RxDB-	22
RTSB+	48	CTSB+	6
RTSB-	49	CTSB-	5
FSTB+	43	FSRB+	3
FSTB-	44	FSRB-	4

1.4.7 DB25 Loop back connector

SIGNAL	DB25M	SIGNAL	DB25M
TxClkO+	14	RxClkI+	25
TxClkO-	2	RxClkI-	13
TxD+	15	RxD+	24
TxD-	3	RxD-	12
RTS+	17	CTS+	21
RTS-	5	CTS-	9
FST+	18	FSR+	23
FST-	6	FSR-	11

1.5 Installation

1.5.1 Hardware Installation

Important: Static electricity can harm system boards. Perform service at an ESD workstation and follow proper ESD procedure to reduce the risk of damage to components. Commtech, Inc. strongly encourages you to follow proper ESD procedure, which can include wrist straps and smocks, when handling Fastcom: *FSCC/SuperFSCC* boards.

1. Turn off PC power and disconnect the power cord.
2. Remove the PC case cover (if applicable).
3. Unpack the Fastcom: *FSCC/SuperFSCC*. Keep the box and static bag for warranty repair returns.
4. Select an open slot in your PC.
5. After removing the blank bracket from your PC, install the Fastcom: *FSCC/SuperFSCC* in the PC by pressing it firmly into the slot. Install the bracket screw to hold it firmly in place.
6. Re-install the cover on your PC (if applicable).

1.5.2 Software Installation (Windows)

Once your card is installed in the computer, you can power on the computer and begin installing the drivers for the card.

1. Windows will detect a new device and start the found new hardware wizard.
2. If it asks to connect to the internet to search for the software, tell it no.
3. The next screen will ask you to automatically search for a driver. Do NOT do this. Select the option that lets you specify a location to search for the driver.
4. Next it will ask you where to search for the drivers. Put a checkbox next to the option that lets you specify a location and clear the rest of the boxes.
5. Click the Browse button and browse to where you have the FSCC software. If you are using the Fastcom CD, that will be the X:\fastcom_cd\Drivers\FSCC Drivers\windows\(version), where X is your CD driver and (version) is your version of Windows. If you downloaded the zip file from our website, you will need to browse to the path where you extracted the zip file. Again select the folder that corresponds to your version of Windows. Click OK and then click Next.
6. The next screen may ask you which board type you have and will also show you the path to the driver that it is about to use. Make sure that path is what you think it should be. Select your board and click Next.
7. The wizard will finish and should report that the installation was successful.
8. The driver for the board itself has now been installed. You will repeat this process multiple times – once for each of the async ports (COM) and once for each of the sync (FSCC) ports. The steps will be almost exactly the same as listed above.

1.5.3 Testing the Installation:

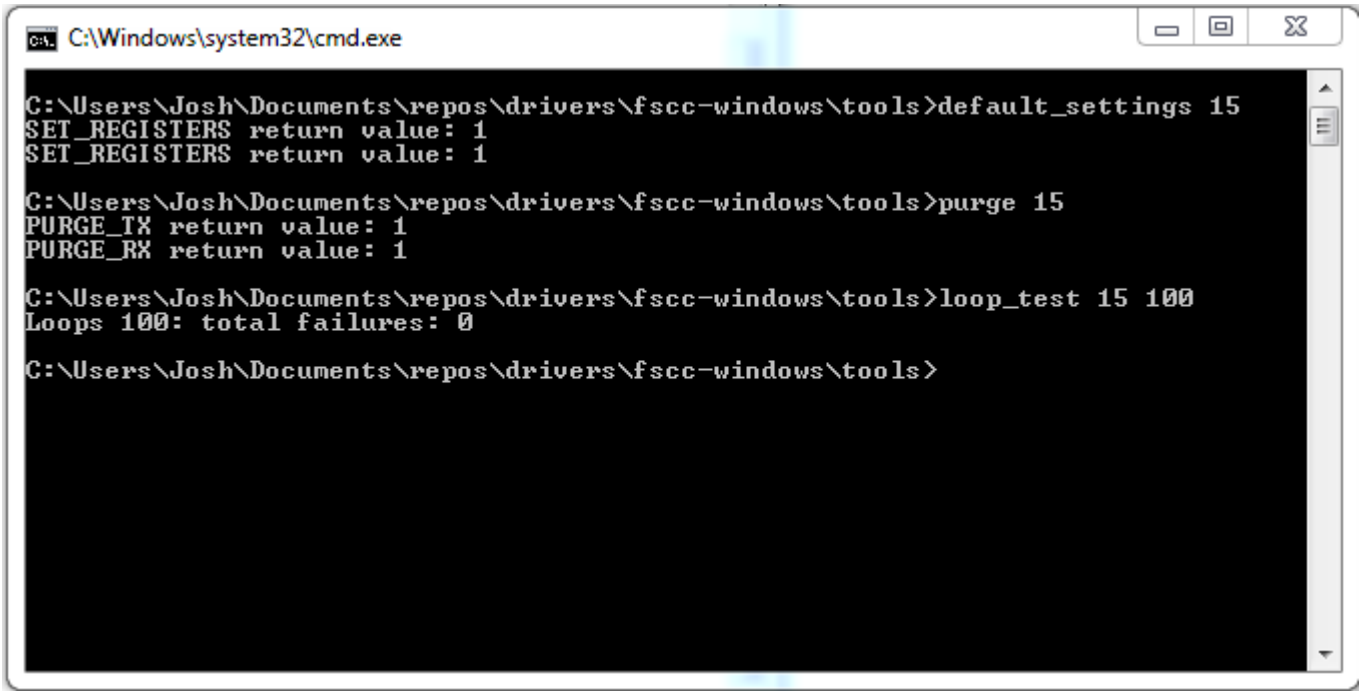
To fully test the installation of your Fastcom: *FSCC/SuperFSCC*, you will need a "loop back plug" which we have included in the box. This loop back plug can be used to test any FSCC/*SuperFSCC*.

Please note, if you purchase a PC/104+ or PCI-104 board, the loop back is included only if you purchased the optional cable. If you did not purchase the optional cable, please reference the table in section 1.4.5.2 for the loop back pin out.

These instructions assume that you have already installed the card and have followed the installation instructions. The Device Manager should show the boards/ports that are installed.

1. Attach supplied loop back plug to the rear of the *FSCC/SuperFSCC* card, or attach the cable assembly to the card and then the loop back to the end of the cable assembly.
2. Compile the programs (default_settings.c, purge.c, loop_test.c) in the \tools subdirectory of the Windows folder.
3. Open a command prompt and change to the \tools subdirectory.
4. Run default_settings, purge, and loop_test with the port as the parameter and the number of loops for loop_test:

The output will look something like this, my port is 15:



1.6 RS422 / RS485

Most engineers have worked with RS-232 devices at least once in their career. If you have never worked with RS-422 or RS-485 devices, you will be pleased to know that working with the FASTCOM: *FSCC/SuperFSCC* is not much different from working with a standard RS-232 device.

The RS-422 standard was developed to correct some of the deficiencies of RS-232. In commercial and industrial applications, RS-232 has some significant problems. First, the cable length between RS-232 devices must be short (usually less than 50 feet at 9600 baud). Second, many RS-232 errors are the result of cables picking up normal industrial electrical noises such as fluorescent lights, motors, transformers, and other EMF sources. Third, RS-232 data rates are functionally limited to 19.2K Baud. On the other hand, the newer RS-422 standard makes cable lengths up to 5000 feet possible and is highly immune to most industrial noises. Data rates are also improved -- the FASTCOM: *FSCC/SuperFSCC* features data rates up to 50 Mbit/sec. These improvements were made possible by differentially driving and receiving the data as opposed to the single ended method employed by the RS-232 standard. With the RS-422 standard, the transmit signal (TX in RS-232) is a differential signal consisting of TX+ and TX-; the receive signal (RX in RS-232) consists of RX+ and RX-.

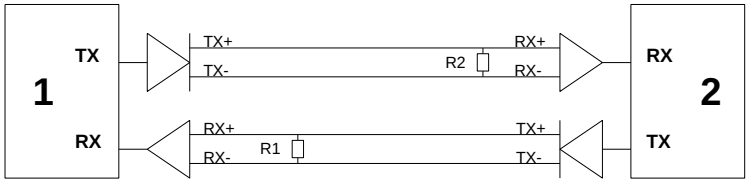
Another draw back of RS-232 is that more than two devices cannot share a single cable. This is also true of RS-422, and that's why the RS-485 standard was developed. RS-485 offers all of the benefits of RS-422 and also allows multiple units (up to 32) to share the same “twisted pair” of wires (see diagram on next page). RS-485 is often referred to as a "multi-drop" or "two-wire, half duplex" network. In order for an RS-485 system to work, only one driver (transmitter) can occupy the network at a time. This means that each station on the network must control the enabling/disabling of their drivers in order to avoid network conflicts. If two drivers engage the network at the same time, data from both will be corrupted. In RS-485 mode, the receivers are always enabled.

1.7 Termination Resistance

In both the RS-422 and the RS-485 mode, the receiver end of the cable between two stations must be terminated with a resistor equal to the characteristic impedance of the wire. This is to prevent signal reflections in the wire and to improve noise rejection. However, **you do not need to add a terminator resistor to your cables when you use the Fastcom: *FSCC/SuperFSCC*. The termination resistance is built in.** We have installed a terminator resistor for each RS422 receiver pair.

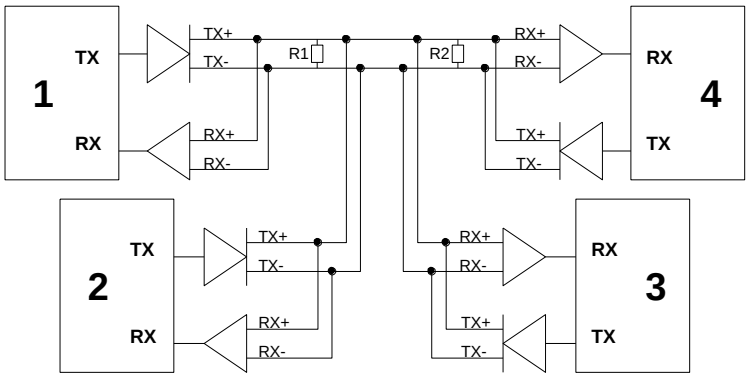
If you are using the Fastcom: *FSCC/SuperFSCC* in a multi-drop network, the termination resistor should be removed from all units except the first and last (see the RS-485 illustration below). Call for technical support if you need to modify the resistor. You may also order the Fastcom: *FSCC/SuperFSCC* without the termination resistor installed (it is easier to add the resistor than to remove it). Observe the resistors in the following drawings and remember that they are built into the Fastcom: *FSCC/SuperFSCC* (shown below):

Typical RS-422 Installation



R1 & R2 - Line Termination (100 ohms)

Typical RS-485 Installation



R1 & R2 - Line Termination (100 ohms)

In addition our termination network also contains a 1kΩ pullup resistor on the + signal and a 1kΩ pull down resistor on the – signal. This ensures that a floating RS485 signal will never bounce and create spurious data on the receive pins.

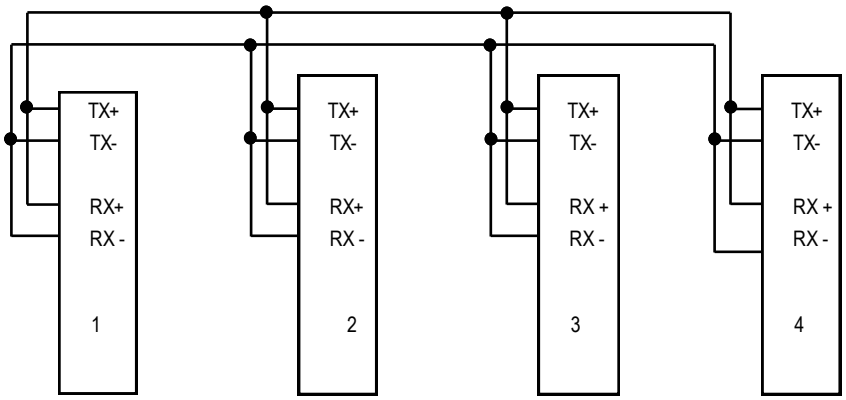
You can read more about termination here:

AN-903 A Comparison of Differential Termination Techniques by Texas Instruments

1.8 RS-485 Mode

RS-485 is often referred to as a multi-drop or two-wire, half duplex network because the drivers (transmitters) and receivers share the same two lines. In fact, up to 32 stations can share the same twisted pair. In order for an RS-485 system to work, only one driver (transmitter) can occupy the network at a time. This means that each station on the network must control the enabling/disabling of its drivers in order to avoid network conflicts. If two drivers engage the network at the same time, data from both will be corrupted. In RS-485 mode, the receivers are always enabled.

The following cable illustration shows four RS-485 Devices sharing the same twisted pair:



Note: The termination resistors from Station #2 and Station #3 have been removed.

Not all RS-422 devices feature RS-485 compatibility; only RS-485 devices can be connected to the RS-485 network.

Note that when in the RS-485 mode, you will need to externally connect TX+ to RX+ and TX- to RX-.

1.9 SuperFSCC difference from standard FSCC

The *SuperFSCC* card features bus-mastering DMA. This DMA greatly decreases the amount of interrupts that must be serviced by the host CPU, thus freeing it up for other activities. By utilizing this DMA, the *Super* is able to stream gapless, full duplex data at the full 50 Mbits/sec[1]. Where the standard FSCC card is only able to perform the same task at around 20 Mbits/sec. The standard card is capable of operating at rates up to 50 Mbit/sec with data bursts. It cannot sustain higher rates without generating gaps in the data (e.g. idle between frames).

[1] This is possible as long as the operating parameters are within reason. For instance, you cannot expect to perform continuous, full duplex operation at 50Mbps with 2 byte frames. This will require reasonable sized frames with a minimum size of around 1024 bytes. If you have any question about whether the card can meet your needs, please contact Technical Support.

2 Fastcom F-Core

2.1 Introduction

- Faced with the inevitability of obsolescence, Commtech decided to put an end to the lifespan problems that plague most computing customers. We designed a serial communications controller with our customers' needs in mind, and built a card around it.
- Wholly designed and owned by Commtech, Inc, this FPGA based SCC has most of the features that you are used to seeing in a quality Fastcom product. It also includes a few new features that come directly from customer requests.

2.2 Hardware

- Because the F-Core is an FPGA based design, it is easily customizable by Commtech's engineers to meet the needs of our demanding customers.
- If the existing FPGA technology is ever discontinued or obsoleted by the chip manufacturer, the design can simply be re-targeted to the next generation of FPGA chip with no impact on compatibility.

2.3 Serial Interface

- Independent full duplex serial channels
    - Data clocking mechanism independently selectable per channel per direction
        - Externally generated Rx and Tx clocks
        - Internally generated Rx and Tx clocks
        - On chip DPLL clock recovery
        - Independent baud rate generators
- FIFOs
    - RxFIFO = 8KB per port (non-uart modes)
    - TxFIFO = 4KB per port (non-uart modes)
    - Programmable interrupt trigger levels (watermark)
- Data protocols
    - HDLC/SDLC
    - Arbitrary maskable sync sequence of up to 4 bytes in length (X-Sync)
    - Transparent data without character framing
- Data encoding schemes
    - NRZ
    - NRZI
    - FM0/FM1

    - Manchester
    - Differential Manchester
- Receive Status Word appended to each complete frame
- Programmable maximum receive frame length checking (RLC = 64 Kbyte)
- Selectable frame sync signals
    - Pulse width of one clock at first bit of data
    - Pulse width of one clock at last bit of data
    - Pulse width of the entire frame
    - Optionally insert a variable number of clock cycles between data and FSS
    - Selectable polarity
- Optional data flow control using modem control lines (RTS, CTS)
    - Option to disable flow control and use signals as general purpose IO
    - Selectable polarity
- Programmable 8-bit preamble and postamble with selectable repetition rate (1 to 255 repetitions)
- Data can be oriented as selectable MSB first or LSB first
- Interframe time fill: idle as repetitive 1s, flags, or sync sequences
- CRC Support
    - Automatic handling in the transmit/receive direction
    - CRC-CCITT (also known as CRC16-ITU) (HDLC)
    - CRC32 (HDLC)
    - CRC16
    - CRC8 (HDLC)
    - Transparent CRC option
    - Reset as 0s / 1s
- Continuous transmission of 1 to 4096 bytes
- Selectable data rates
    - Synchronous (internally clocked) data rates up to 50 MHz (20 MHz FSCC)
    - Synchronous (externally clocked) data rates up to 30 MHz (20 MHz FSCC)
    - Asynchronous (DPLL clock recovery) data rates up to 12.5 Mbit/sec
    - These maximum data rates are estimates. They are greatly affected by a number of different factors. See the Baud Rates section for more details.
- Data and clock inversion

- Interruptible hardware timer
- Transmit a single frame on the expiration of the hardware timer

2.4 Data Protocols

- HDLC/SDLC
  - Automatic flag detection and transmission
  - General frame format: 0x7e,info,CRC,CRC,[CRC,CRC,]0x7e
  - Frame format with maskable address enabled: 0x7e,address,[address,]info,CRC,CRC,[CRC,CRC,]0x7e
  - Zero insertion/deletion
  - One insertion/deletion
  - Shared flag mode: opening flag and closing flag of back-to-back frames can use the same flag.
  - Error detection (abort, overrun, underrun, CRC error, too long/short frame)
  - Can use CRC8, CRC-CCITT, or CRC32
- Arbitrary Sync Sequence (X-Sync)
  - Any begin/end frame sync sequence of up to 4 bytes
  - Frame format: SYN1,[SYN2,SYN3,SYN4,]info,[CRC,CRC, CRC,CRC,][TCR1,TCR2,TCR3,TCR4]
  - Maskable: can select all or parts of the sync sequence to be a sync match
  - Termination sequence detection up to 4 bytes (maskable)
  - Shared flag mode: opening flag and closing flag of back-to-back frames can use the same flag.
  - Error detection
  - Can use any of the available CRC methods
- Transparent Mode
  - Fully bit transparent (no framing or bit manipulation)
  - Can synchronize using selectable frame sync signals

3 Data Flow

3.1 Receive

- In the receive direction, data is clocked into the F-Core's 8 Kbyte RxFIFO (per channel).
- The RxFIFO is capable of storing up to 512 complete frames of data. The queued frames can be any size as long as they do not exceed the RxFIFO size. For example, you could have 512 - 14 byte frames; or you could have 8 - 1022 byte frames; or 16 - 510 byte frames without overflowing the RxFIFO[1].

3.1.1 Receive Interrupts[2]

3.1.1.1 Receive Frame Start (RFS)

RFS is generated whenever an opening flag, sync sequence or frame sync signal is detected.

3.1.1.2 Receive FIFO Trigger (RFT)

RFT is generated when the number of bytes in the RxFIFO rises above the programmed RxFIFOTrigger level (e.g. high-water mark).

3.1.1.3 Receive Frame End (RFE)

RFE is generated whenever the RxStatus word for that frame is pushed into the FIFO. When this occurs, the frame is available to be read from the FIFO.

3.1.1.4 Receive Frame Overflow (RFO)

RFO is generated whenever the maximum number (512) of RxByteCount entries is exceeded.

3.1.1.5 Receive Data Overflow (RDO)

RDO is generated whenever data is received that exceeds the depth (8 Kbyte) of the RxFIFO.

3.1.1.6 Receive Frame Lost (RFL)

RFL is generated whenever the receiver detects an additional frame while an RDO interrupt is already waiting to be serviced. The RFL bit in the RxStatus word is set for the next message that is pushed into the RxFIFO. The additional frame is lost.

[1] Each frame will have an RxStatus word appended to it in the RxFIFO; the RxStatus word occupies two bytes of FIFO space. Thus each frame size is actually 2 bytes larger than the actual frame.
[2] All bits in the interrupt indication register will reset when the register is read.

3.2 Transmit

- In the transmit direction, data is pushed into the F-Core's 4 Kbyte TxFIFO (per channel).
- The TxFIFO is capable of storing up to 256 complete frames of data. The queued frames can be any size as long as they do not exceed the TxFIFO size. For example, you could have 256 - 16 byte frames; or you could have 4 - 1024 byte frames; or 16 - 256 byte frames.

3.2.1 Transmit Modes

3.2.1.1 Transmit Repeat (XREP)

- When the XREP command is issued, the F-Core transmits TxByteCount bytes in the TxFIFO repetitively (treating the repeated block as a single frame) until the transmit reset (XRES) command or the SXREP command is issued.

3.2.1.2 Transmit Frame (XF)

- The XF command should be considered normal operation of the transmitter.
- When the transmitter is idle and a frame is ready to be sent, the number of bytes in the frame is entered into TxByteCount and the data (up to 4096 or TxFIFOTrigger bytes at a time) is pushed into the TxFIFO. When the XF command is issued, the transmitter becomes active and the number of bytes in TxByteCount is pushed out of the TxFIFO into the transmitter.
- The XF command is issued with each queued frame, but is only acted upon by the controller if the transmitter is in the idle state.

3.2.1.3 Transmit on Timer (TXT)

The F-Core has the ability to transmit one queued frame on the expiration of the hardware timer (see 4.12) repetitively. When the timer expires (TIN interrupt), the transmitter sends one complete frame from the TxFIFO. When the frame is completely sent, it returns to an idle state until the next timer expiration. The user is able to send a single frame repetitively using the XREP command. Also, the user is able to queue frames up to the TxFIFO depth or a maximum of 256 frames, using the normal XF command. Data is only transmitted on timer expiration boundaries. If the user queues a frame, it is not transmitted until the next timer expiration. If the TxFIFO is empty when the timer expires then no action is taken.

3.2.1.4 Transmit on External Signal (TEXT)

The F-Core has the ability to transmit one queued frame on the transition of an external signal, repetitively. When the transition is detected, the transmitter sends one complete frame from the TxFIFO. When the frame is completely sent, it returns to an idle state until the next transition is detected. The user is able to send a single frame repetitively using the XREP command. Also, the user is able to queue frames up to the TxFIFO depth or a maximum of 256 frames, using the normal XF command. Data is only transmitted on a detected transition on the external signal. If the user queues a frame, it is not transmitted until the next transition. If the TxFIFO is empty when the transition occurs then no action is taken.

3.2.2 Transmit Interrupts

3.2.2.1 Transmit FIFO Trigger[3] (TFT)

The TFT interrupt is used to indicate that the TxFIFO is ready to accept more data. It is generated when:

- The number of bytes in the TxFIFO falls below the programmed TxFIFOTrigger level (low-water mark)
- When moving from idle to active, if the BytesInTxFIFO count is less than the TxFIFOTrigger level anytime the XF command is issued

3.2.2.2 All Sent (ALLS)

The ALLS interrupt is generated any time the transmitter moves from an active to an idle state. This is an indication that it has finished transmitting data.

3.2.2.3 Transmit Data Underrun (TDU)

The TDU interrupt is generated whenever transmitter moves from an active to an idle state, and the number of bytes pushed into the TxFIFO has not yet reached the current TxByteCount. This is an indication that the software is not moving data fast enough.

[3] There will be some level of hysteresis (TBD) in the Tx trigger level to provide time for the bus interface to fill the FIFO without retriggering the interrupt due to outgoing bytes.

4 Global Features Details

4.1 Clocking System

There are several different ways to clock data into and out of the F-Core:

- You can use externally sourced clocks in both transmit and receive directions
- The transmit clock can be derived from our onboard clock source
- The receive clock can be recovered from the received data using DPLL
- The receive clock can be used as the transmit clock.

In the transmit direction, the data is considered valid on the rising edge of the effective transmit clock. The receive data is considered valid on the falling edge of the effective receive clock. As an option, we are able to flip the clock edges that correspond to valid data.

The effective transmit clock is always outputted on the TxClkOut pin, whether it is derived on board clock generator, from the RxClk, or from a signal on the TxClkIn pins.

If you select an external clock mode, please make sure the clock is present before trying to operate the ports. If a clock is not present, and an operation is performed on the port that causes something in the Command Register to execute, it may cause the F-Core to appear to lock up and cause an error. This is because the commands execute using the transmit or receive clock and with no clock the command cannot complete.

4.1.1 Onboard Clock Generators

Each port has a fully independent, software settable clock generator that feeds the baud rate generator of the FCore. Each clock generator can be set to nearly any frequency between 20kHz and 200MHz. This clock is fed into the internal baud rate generator of the FCore where it is divided down to get the desired baud rate. In the following section, any clock mode that runs off of the internal BGR is utilizing this onboard clock generator.

The drivers for each OS contain a function or program that accepts the desired frequency and a parts per million (ppm) variable as inputs. Sometimes the clock generator cannot be set to exactly the desired frequency. If this is the case, you can utilize the ppm variable to allow for some amount of deviation in the calculation and the algorithm uses the frequency that is closest to the desired input but within the specified ppm.

4.1.2 Clock Modes

Table 1: Clock Modes of the F-Core

Clock Mode	Receive Source			Transmit Source		
	External (RxClk)	Internal BGR	DPLL	External (TxClkIn)	Internal BGR	Effective RxClk
0	•			•		
1	•				•	
2	•					•
3			•	•		
4			•		• [1]	
5			•			•
6		•		•		
7		•			•	

Note 1- In DPLL Mode TxClkOut = BGR / 16

Note 2: Clock source to BGR is always the onboard clock input to the FPGA

Note 3: Clock source to DPLL is always the BGR

4.1.2.1 Clock Mode 0

Separate, externally generated (off board) receive and transmit clocks are supplied to the F-Core via their respective pins (RxClk and TxClkIn).

4.1.2.2 Clock Mode 1

Receive clock is supplied directly from the RxClk pins. Transmit clock is derived from the internal BGR fed by the onboard clock generator.

4.1.2.3 Clock Mode 2

Both receive and transmit clocks are externally sourced from the RxClk pin.

4.1.2.4 Clock Mode 3

Receive clock comes from the DPLL clock recovery. The BGR is fed by the onboard clock generator 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. Transmit clock comes directly from the TxClkIn pin.

4.1.2.5 Clock Mode 4

Receive clock comes from the DPLL clock recovery. The BGR is fed by the onboard clock generator 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. Transmit clock is derived from the internal BGR / 16.

4.1.2.6 Clock Mode 5

Receive clock comes from the DPLL clock recovery. The BGR is fed by the onboard clock generator and it delivers a reference clock at a frequency equal to 16 times the nominal bit rate for the DPLL which in turn generates the receive clock. The generated receive clock is used as the transmit clock as well.

4.1.2.7 Clock Mode 6

Receive clock is derived from the internal BGR fed by the onboard clock generator. Transmit clock is supplied directly from the TxClkIn pins.

4.1.2.8 Clock Mode 7

Both the receive and transmit clocks are derived from the internal BGR fed by the onboard clock generator.
Note: this mode should only be used with the receiver disabled.

4.1.3 Baud Rates

Our testing indicates that you will be able to maintain continuous (gapless), 50 MHz baud rates full duplex (that is simultaneous transmit and receive) on each channel using the SuperFSCC (20 MHz with standard FSCC).

The length and quality of the cables connecting to the other device are contributing factors to maximum baud rates. Longer cable lengths and lower quality cable will decrease the achievable data rates.

Improper line termination will attenuate the signal and can also decrease the maximum data rates.

The exact speed that you can achieve will also depend on your system's ability to move the data fast enough. This means the size of your data frames - it is easier for the system to write fewer larger frames than it is to write many smaller frames.

4.1.4 Clock Recovery (DPLL)

The F-Core 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 a separate serial clock line. For this purpose, the DPLL is supplied with a 'reference clock' from the BGR, which is 16 times the expected data clock rate (clock modes 3, 4 and 5). The transmit clock may be obtained externally (clock mode 3), by dividing the output of the BGR by a constant factor of 16 (clock mode 4) or also directly from the DPLL (clock mode 5).

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 method for clock recovery depends on the selected data encoding.

The following functions are implemented:

4.1.4.1 Interference Rejection and Spike Filtering

Two or more edges in the same directional data stream within a time period of 16 reference clocks are considered to be interference and consequently no additional clock adjustment is performed.

4.1.4.2 Phase Adjustment (PA)

Referring to Figure 1, Figure 2, and Figure 3, in the case where an edge appears in the data stream within the PA fields of the time windows, the phase is adjusted by 1/16 of the data.

4.1.4.3 Phase Shift (PS) (NRZ & NRZI only)

Referring to Figure 1 in the case where an edge appears in the data stream within the PS field of the time window, a second sampling of the bit is forced and the phase is shifted by 180 degrees.

Note: Edges in all other parts of the time windows are 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 F-Core offers the option to disable the Phase Shift function for NRZ and NRZI encodings. In this case, the PA fields are extended as shown in Figure 2.

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.

Figure 1, Figure 2 and Figure 3 explain the DPLL algorithms used for the different data encodings.

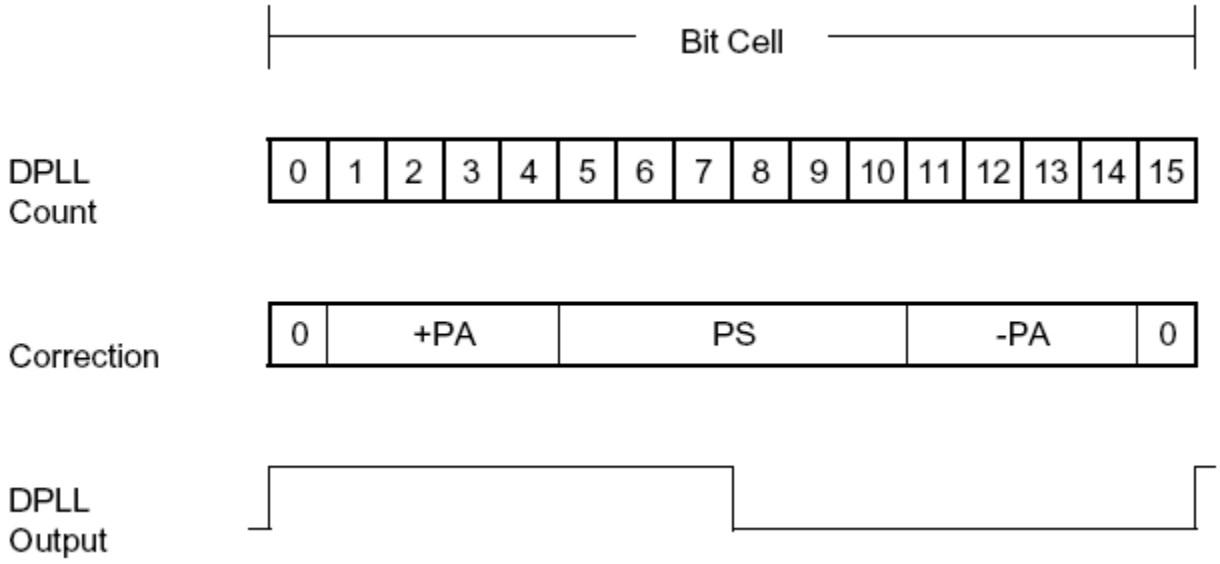


Figure 1 - DPLL Algorithm for NRZ and NRZI Coding with Phase Shift Enabled

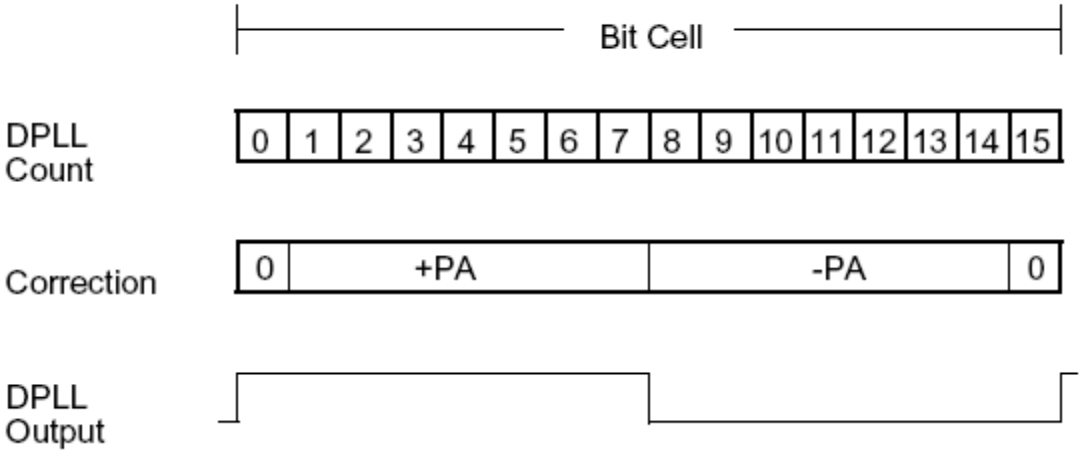


Figure 2 - DPLL Algorithm for NRZ and NRZI Coding with Phase Shift Disabled

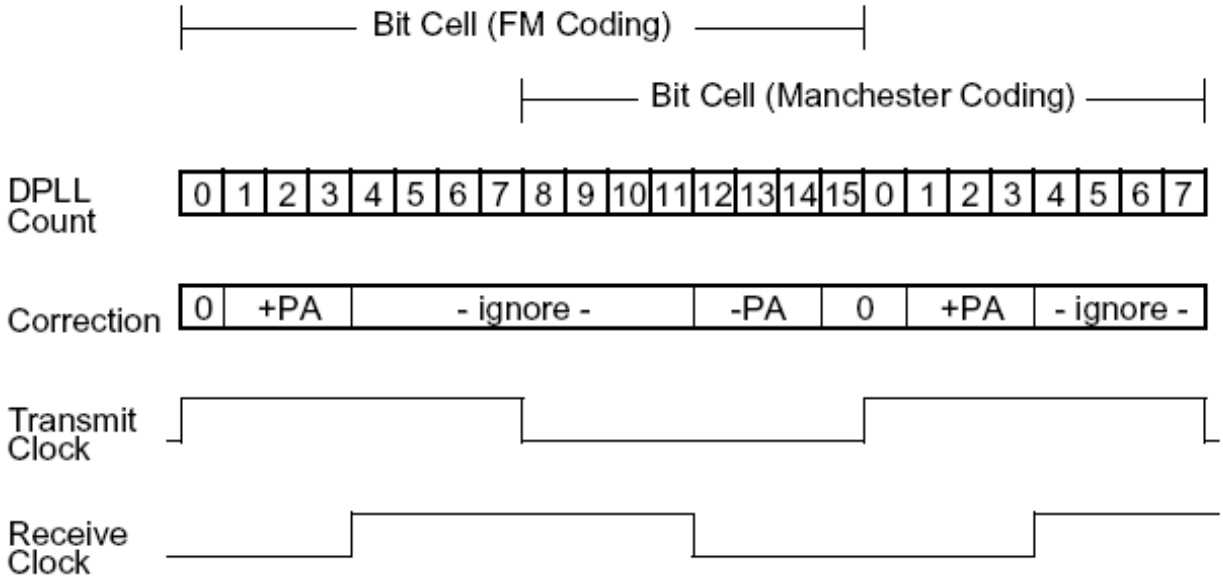


Figure 3 - DPLL Algorithm for FM0, FM1 and Manchester Coding

4.1.4.4 DPLL Status Flag

To supervise correct function, an RxStatus flag in the status register (6.2.7) reflects the DPLL's synchronous/asynchronous state. When synchronized, if the DPLL detects an edge in the data that is more than 1/16 of the data clock away from center, it enters an asynchronous state, sets the DPLLA status flag, and attempts to resynchronize. If the loss of synchronization occurs while receiving a frame, that frame is aborted, marked complete, and reception restarts on the next begin frame condition with a synchronized DPLL.

Note that it is possible for the DPLLA bit/status to not be telling the whole truth. With all revisions of firmware the DPLLA status is only updated when an edge occurs and the clock phase is out of the window (as described in the manual). As a result, a long sequence of no edges does not give a DPLLA indication until the next edge occurs.

4.1.4.5 DPLL Reset

The DPLL is forced into a reset mode if more than 32 (default setting) receive clocks elapse without an edge in the data. Practically this allows for proper operation and will lock on to gapped frames with no edges between them. The implementation inserts an extra bit at the beginning of any given frame sequence (either an extra zero or one depending on the relative phase of the free running internal receive clock and the first edge in the data) that follows one of these no edge periods. This should not present any problems for HDLC modes, or any mode that uses a synchronization sequence (X-Sync, or FSR gated receive) and will only cause difficulty in Transparent mode when no beginning sync is used. It is not recommended to receive in transparent mode, using the DPLL, with long sequences of 1s or 0s with NRZ line encoding.

The number of clock edges before the DPLL is considered out of sync is now configurable in a register (0×58) the DPLL RESET register. It may have a value from 1 to 511 and is the number of clocks without an edge that the DPLL considers OK, more clocks than this and it will snap to the next edge (with the possibility of the extra bit being clocked in as described above). The number set is in 8 clock increments, so a value of 1 is 8 clocks, 2 16 clocks, etc. The default value is 4, which yields 32 clocks as described above.

4.2 Data Encoding

The F-Core 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)
- Differential Manchester

4.2.1 NRZ and NRZI Encoding

- **NRZ:** The signal level corresponds to the value of the data bit. By setting the data inversion bit, the SCC may invert the transmission and reception of data.
- **NRZI:** A logical '0' is indicated by a transition and a logical '1' is indicated by no transition at the beginning of the bit cell.

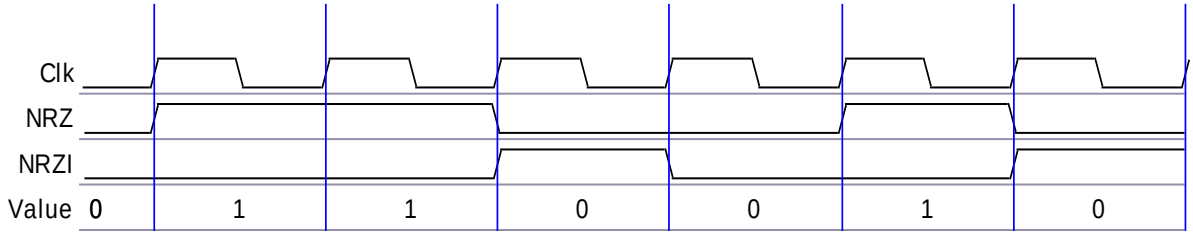


Figure 4 - NRZ and NRZI encoding

4.2.2 FM0 and FM1 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, whereas 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°.

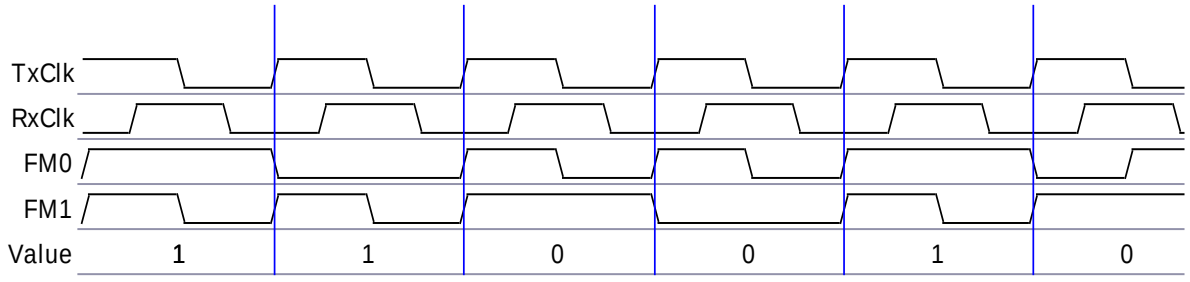


Figure 5 - FM0 and FM1 encoding

4.2.3 Manchester 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.
- The normal mode of operation would be considered the G.E. Thomas version of Manchester. You can effectively switch to the IEEE 802.3 version of Manchester by setting bit CCR1:TDP = 1.

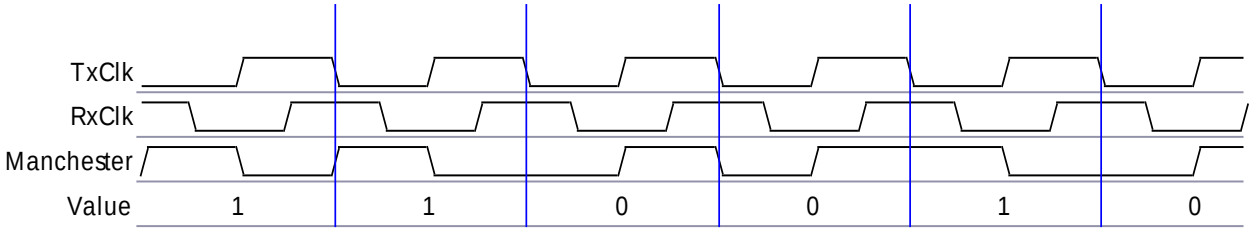


Figure 6 - Manchester encoding

4.2.4 Differential Manchester Encoding[1]

Also known as CDP, Conditional DePhase encoding is a method of encoding data in which data and clock signals are combined to form a single self-synchronizing data stream. It is a differential encoding, using the presence or absence of transitions to indicate logical value. This gives it several advantages over vanilla Manchester encoding:

- Detecting transitions is often less error-prone than comparing against a threshold in a noisy environment.
- Because only the presence of a transition is important, polarity is not. Differential schemes work exactly the same if the signal is inverted (wires swapped). (Other line codes with this property include NRZI, bipolar encoding, and MLT-3 encoding).

A '1' bit is indicated by making the first half of the signal equal to the last half of the previous bit's signal i.e. no transition at the start of the bit-time. A '0' bit is indicated by making the first half of the signal opposite to the last half of the previous bit's signal i.e. a zero bit is indicated by a transition at the beginning of the bit-time. In the middle of the bit-time there is always a transition, whether from high to low, or low to high. A reversed scheme is possible, and no advantage is given by using either scheme.

Differential Manchester is specified in the IEEE 802.5 standard for token ring LANs, and is used for many other applications, including magnetic and optical storage.

Note: In differential Manchester encoding, if a "1" is represented by one transition, then a "0" is represented by two transitions and vice versa.

[1] Customer request. From Wikipedia: http://en.wikipedia.org/wiki/Differential_Manchester_encoding

4.3 Receive Status Word

The contents of the receive status word consists of the lower 16-bits of the status register (6.2.7). At the detection of a frame end condition, the current state of the status register is pushed onto the RxFIFO as the last word of each stored frame and appears in your data with the lower order byte first. In this way, the RxStatus word describes the state of the F-Core at the time the frame ended.

Note: this causes your expected frame size to increase by two bytes.

4.4 Receive Length Check

When enabled, this programmable 16-bit value is compared to the total number of bytes received for the current frame. The RLC terminates an incoming frame when "bytes received" is greater than the programmed RLC count (CCR2:RLC). "Bytes received" increments for each byte that comes into the receiver. This includes any sync, address, data, crc or term bytes that may be enabled. It counts everything.

When the receive length condition is reached a message end condition is generated (including the receive status word) , the bytes are transferred to the RxFIFO and the RLEX bit in the status register (6.2.7) is set. The received data is transferred to the host where it shall be dealt with appropriately.

This function has a quirk that must be explained so that you can get exactly the numbers that you need for your data. If you leave CRC checking enabled, the firmware considers the last byte(s) received to be the CRC value even if it is only data. As such if you do not set the bit DRCRC (which turns off CRC checking in the receiver) you will lose the last byte(s) of your data because it thought they were CRC. In order to accurately pass all of your data over to the receive FIFO you need to set bit CCR1:DRCRC = 1.

Example: Receive the data portion of a frame that has 2 sync bytes and 1028 bytes of data, no CRC and no termination:

Set RLC = 1029, SYNC2F=0, DRCRC=1
Read size = 1030 = (1028 byte frame + 2 RxStatus bytes)

Note: Maximum RLC = 65535 bytes

4.5 Frame Sync Signals

In some cases a given system may require the use of frame synchronization signals (FSR and FST) to enable reception of data. These optional signals facilitate communication with such a system. The F-Core automatically generates these frame sync signals in relation to its transmit data. It also automatically utilizes these signals to synchronize with receive data. There are three operating modes of the frame sync signals.

- Mode 1: The sync signal completely frames the data
    - In the transmit direction, the Frame Sync Transmit (FST) signal will be active sometime before the clock edge (less than one clock period) that corresponds with the first data bit. FST will go inactive sometime after the clock edge (less than one clock period) corresponding to the last data bit.

- In the receive direction, the receiver must begin clocking in data on the first clock edge with an active Frame Sync Receive (FSR) signal and proceed with the frame start sequence. The receiver will stop clocking in data on the first clock edge with an inactive frame sync signal and mark the received data block as a complete frame with the end sequence.
- In this mode, the FSR signal will only be used to synchronize data when in Transparent Mode.

- Mode 2: The sync signal is found at the beginning of data
  - In the transmit direction, the FST signal will go active sometime before the clock edge (less than one clock period) that corresponds with the first data bit. FST will then go inactive sometime (less than one clock period) after that same clock edge.
  - In the receive direction, the receiver must begin clocking in data on the first clock edge with an active FSR signal and proceed with the frame start sequence. The receiver will continue clocking in data until a frame end condition is reached. The receiver will ignore the active FSR signal if its duration spans more than one clock edge; i.e. the FSR signal only enables the receiver when the receiver is in an idle state.
  - In this mode, the FSR signal will only be used to synchronize data when in Transparent Mode. This signal will both begin the next receive data block and end the previous data block.
- Mode 3: The sync signal is found at the end of data
  - In the transmit direction, the FST signal will go active sometime before the clock edge (less than one clock period) that corresponds with the last data bit. FST will then go inactive sometime (less than one clock period) after that same clock edge.
  - In the receive direction, the receiver must stop clocking in data on the first clock edge with an inactive FSR signal and proceed with the frame end sequence. The receiver will ignore the inactive FSR signal if its duration spans more than one clock edge; i.e. the FSR signal only disables the receiver when the receiver is in an active state.
  - In this mode, the FSR signal could be used to terminate a receive data block when in Transparent Mode as well as X-Sync mode. This signal will both terminate the current receive data block and begin the next data block when using Transparent Mode.
- Mode 4: Both Mode2 and Mode 3 at the same time
- Mode 5: Frame sync Mode 5 is similar to mode 1, except there are independent controls for both the leading and trailing edge of the FST signal. When mode 5 is enabled CCR2[7:4] control the leading edge of the FST signal and CCR2[11:8] control the trailing edge. CCR2[11:8] will have no effect in any other Frame Sync Mode.

The user can optionally choose to insert a number of clock cycles between data and FSS. The 4-bit value, CCR2:FSTO & CCR2:FSRO, specifies the number of additional transmit or receive

clocks cycles to shift the frame sync signals with a range of –8 clocks to +7 clocks. The diagram shown below is with FSTO set to the default of 0 inserted clocks.

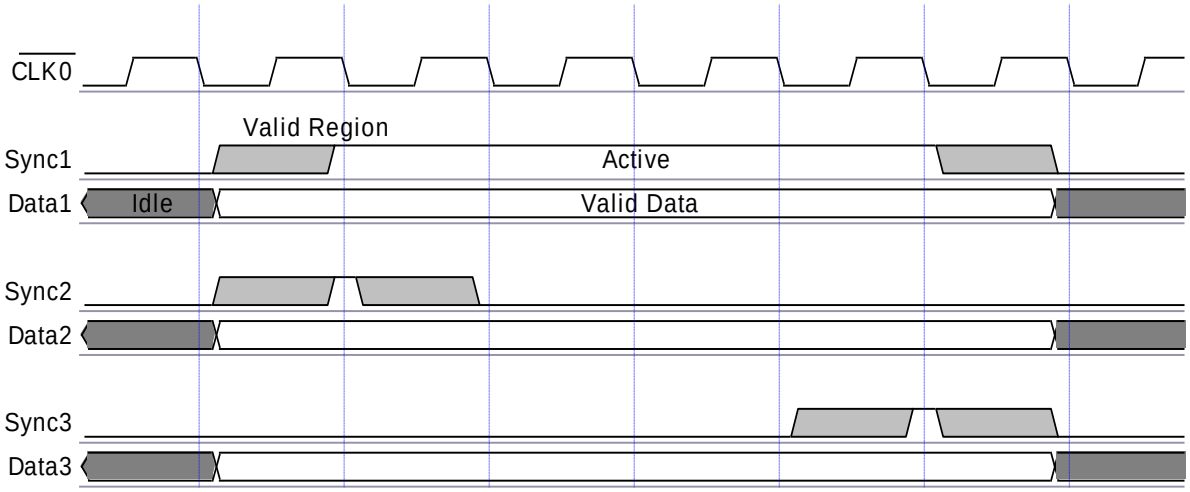


Figure 7 - Frame Sync Modes

4.6 Modem Control Signals

The F-Core provides two pins (RTS and CTS) per serial channel to facilitate transmission control. There are also several modem control signals from the True-Async mode for each channel. When not using the True-Async mode, these pins are free to be used as general purpose I/O's.

- The input pins each create a maskable interrupt upon state change
- The current state of the input pins is accessible via the Status Register
- The current state of the input pins is inserted into the RxStatus word for each frame

4.6.1 Request to send (RTS)

- In automatic mode, the RTS pin goes active as soon as data is ready to be transmitted. This implies that the RTS signal will go active at least 1 clock prior to data being transmitted, and goes inactive 1 clock after data transmission is complete. If data is ready to be transmitted, but has not actually been transmitted yet (i.e. because CTS is inactive), RTS must remain active to facilitate RTS-CTS handshaking.
- Selectable polarity. User can select whether the signal is active low or active high (default: active low).
- Can be used as a general purpose output pin

4.6.2 Clear to send (CTS)

- The transmitter is paused if the CTS input signal is inactive and enabled if active.
- The current byte is completely sent even if CTS becomes inactive during transmission.
    - In HDLC mode, the current frame is aborted with the HDLC abort sequence (seven contiguous '1' bits). The transmitter returns to the idle state.

- In X-Sync mode, the current frame will be closed using the termination sequence or sync signal (if in use). The transmitter returns to the idle state.
- A CTS Abort (CTSA) interrupt will be generated after CTS deassertion causes an abort sequence to be transmitted. If CTS deassertion does not occur during transmission of a frame, this interrupt will not be generated.

- Selectable polarity. User can select whether the signal is active low or active high (default: active low).
- Can be used as a general-purpose input pin

4.6.3 DTR

This shall be a general-purpose output pin. This signal will be present from our True-Async module, so we will connect it to the SCC as well.

4.6.4 DSR, CD, RI

These signals shall be general-purpose input pins. These signals will be present from our True-Async module, so we will connect them to the SCC as well.

4.7 Preambles and Postambles

There shall be an optional 8-bit preamble and an optional 8-bit postamble. When enabled, each shall have a selectable repetition rate of between 1 and 255 repetitions. These can be used to aid in the synchronization of a clock recovery circuit or in the enabling of an RS485 driver or receiver.

4.8 Order of Bit Transmission

Usually, data bytes within element fields shall be transmitted least significant bit first (i.e. the first bit of the sequence number that is transmitted shall have the weight 2^0). However, as an option, the order of transmission can be reversed so that bytes shall be transmitted most significant bit first (i.e. the first bit of the sequence number that is transmitted shall have the weight 2^7).

If CRC transmission is enabled, the CRC bytes shall always be transmitted to the line commencing with the coefficient of the highest term (i.e. MSB first), regardless of the order of bit transmission.

4.9 Interframe Time Fill

Inter-frame time fill shall be accomplished by repetitively transmitting contiguous flags, sync sequences or eight contiguous "1" bits. Selection of the inter-frame time fill method depends on systems requirements.

Note: The receiver must be able to ignore back-to-back SYNC until there is SYNC that is followed by non-SYNC data. At that instance, the receiver will no longer look for SYNC characters and will clock in raw data until a stop condition is reached. If you have SYNC2F set, the hardware consider the last SYNC sequence before the non-SYNC sequence to be the "real" SYNC and just put the SYNC sequence into the FIFO once. All the rest of them would be idles and are not valid data. If SYNC2F is not set, the hardware strips only the "real" SYNC and will

no longer be hunting for SYNC characters. If one should happen to appear in the data, nothing will happen (unless it also matches the TERM character)

4.10 CRC Frame Checking Sequences

The CRC frame checking sequence is calculated for the entire length of the frame, excluding the opening and/or closing sequences, the CRC, and any bits inserted for transparency. The calculation shall begin immediately after the opening (SYNC) sequence and continue over the designated portion of the entire frame and end at the closing (TERM) sequence. If the CRC calculation/comparison for a given frame does not match, the CRC bit in the STAR register will be set.

4.10.1 8-bit frame checking sequence (CRC-8)

Per the HDLC specification, the 8-bit CRC shall be the ones complement of the sum (modulo 2) of

a) the remainder of

$$x^k (x^7 + x^6 + x^5 + x^4 + x^3 + x^2 + x + 1)$$

divided (modulo 2) by the generator polynomial

$$x^8 + x^2 + x + 1,$$

where k is the number of bits in the frame existing between, but not including, the final bit of the opening flag and the first bit of the CRC, excluding start and stop elements (start/stop transmission), and bits (synchronous transmission) and octets (start/stop transmission) inserted for transparency, and

b) The remainder of the division (modulo 2) by the generator polynomial

$$x^8 + x^2 + x + 1$$

of the product of x^8 by the content of the frame existing between, but not including, the final bit of the opening flag and the first bit of the CRC, excluding start and stop elements (start/stop transmission), and bits (synchronous transmission) and octets (start/stop transmission) inserted for transparency.

As a typical implementation, at the transmitter, the initial content of the register of the device computing the remainder of the division is preset to all ones and is then modified by division by the generator polynomial (as described above) of the address, control and information fields; the ones complement of the resulting remainder is transmitted as the 8-bit CRC.

At the receiver, the initial content of the register of the device computing the remainder is preset to all ones. The final remainder after multiplication by x^8 and then division (modulo 2) by the generator polynomial

$$x^8 + x^2 + x + 1$$

of the serial incoming protected bits and the CRC, will be "1111 0011" (x^7 through x^0, respectively) in the absence of transmission errors.

4.10.2 16-Bit frame checking sequence

4.10.2.1 CRC-CCITT

Per the HDLC specification, the 16-bit CRC shall be the ones complement of the sum (modulo 2) of

a) The remainder of

$$x^k (x^{15} + x^{14} + x^{13} + x^{12} + x^{11} + x^{10} + x^9 + x^8 + x^7 + x^6 + x^5 + x^4 + x^3 + x^2 + x + 1)$$

divided (modulo 2) by the generator polynomial

$$x^{16} + x^{12} + x^5 + 1,$$

where *k* is the number of bits being protected by the CRC and

b) The remainder of the division (modulo 2) by the generator polynomial

$$x^{16} + x^{12} + x^5 + 1,$$

of the product of x^{16} by the content of the *k* bits being protected.

As a typical implementation, at the transmitter, the initial content of the register of the device computing the remainder of the division is preset to all ones and is then modified by division by the generator polynomial (as described above) of the address, control and any remaining bits of the designated *k* bits being protected; the ones complement of the resulting remainder is transmitted as the 16-bit CRC.

At the receiver, the initial content of the register of the device computing the remainder is preset to all ones. The final remainder after multiplication by x_{16} and then division (modulo 2) by the generator polynomial

$$x^{16} + x^{12} + x^5 + 1,$$

of the serial incoming protected bits and the CRC will be 0001 1101 0000 1111 (x^{15} through x^0, respectively) in the absence of transmission errors.

4.10.2.2 CRC-16

CRC-16 is not specified in the HDLC document but is commonly used as a frame checking sequence for non-HDLC devices, therefore it will be available to users for non-HDLC modes. Commonly referred to as IBM's CRC-16, this algorithm uses the following polynomial:

$$x^{16} + x^{15} + x^2 + 1$$

4.10.3 32-bit frame checking sequence (CRC-32)

Per the HDLC specification, the 32-bit CRC shall be the ones complement of the sum (modulo 2) of

a) the remainder of

$$x^k (x^{31} + x^{30} + x^{29} + x^{28} + x^{27} + x^{26} + x^{25} + x^{24} + x^{23} + x^{22} + x^{21} + x^{20} + x^{19} + x^{18} + x^{17} + x^{16} + x^{15} + x^{14} + x^{13} + x^{12} + x^{11} + x^{10} + x^9 + x^8 + x^7 + x^6 + x^5 + x^4 + x^3 + x^2 + x + 1)$$

divided (modulo 2) by the generator polynomial

$$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1,$$

where k is the number of bits being protected by the CRC and

b) The remainder of the division (modulo 2) by the generator polynomial

$$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$$

of the product of x^{32} by the content of the k bits being protected.

As a typical implementation, at the transmitter, the initial content of the register of the device computing the remainder of the division is preset to all ones and is then modified by division by the generator polynomial (as described above) of the address, control and any remaining bits of the designated k bits being protected; the ones complement of the resulting remainder is transmitted as the 32-bit CRC.

At the receiver, the initial content of the register of the device computing the remainder is preset to all ones. The final remainder after multiplication by x^{32} and then division (modulo 2) by the generator polynomial

$$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1,$$

of the serial incoming protected bits and the CRC will be 1100 0111 0000 0100 1101 1101 0111 1011 (x^{31} through x^0, respectively) in the absence of transmission errors.

4.11 Data and Clock Inversion

- With this option selected, data will be received/transmitted inverted on a per bit basis. This includes all data fields, flags and idle sequences. This inversion should take place as the data stream enters or leaves the F-Core. See CCR1:RDP and CCR1:TDP
- With this option selected, receive and transmit clocks will be inverted before entering or leaving the F-Core, effectively changing the part from a falling edge device to a rising edge device when enabled. See CCR1: TCOP, CCR1:TCIP and CCR1:RCP

4.12 Hardware Timer

The hardware timer is a configurable countdown timer. The timer can be clocked by any one of three sources: effective TxClk, effective RxClk, or PCI clock. It can optionally count down once and then stop or reset and count down continuously. The Timer register is 32-bits: three of them are used as control signals leaving a 29-bit count down value. When the timer expires it triggers a TIN interrupt.

4.13 Transmit on Timer

The FSCC has the ability to transmit one queued frame on the expiration of the hardware timer repetitively. When the timer expires (TIN interrupt), the transmitter will send one complete frame from the TxFIFO. When the frame is completely sent, it will return to an idle state until the next timer expiration. The user will be able to send a single frame repetitively using the XREP command. Also, the user will be able to queue frames up to the TxFIFO depth or a maximum of 256 frames, using the normal XF command. Data is only transmitted on timer expiration boundaries. If the user queues a frame, it is not transmitted until the next timer expiration. If the TxFIFO is empty when the timer expires then no action is taken.

4.14 Zero Bit Insertion

The transmitter shall examine the frame content between the two sync sequences including the address and CRC fields, when present, and shall insert a "0" bit after all sequences of 5 contiguous "1" bits (including the last 5 bits of the CRC) to ensure that a flag sequence is not simulated. The receiver shall examine the frame content and shall discard any "0" bit which directly follows 5 contiguous "1" bits. This is the standard HDLC bit stuffing mechanism

4.15 One Bit Insertion

Similar to the zero bit insertion (bit stuffing) mechanism, as defined by the HDLC protocol, the F-Core offers a completely new feature of inserting/deleting a ’one’ after seven consecutive ‘zeros’ into the transmit/receive data stream. This method is useful if clock recovery is performed by DPLL.

If using NRZ data encoding 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.

By enabling the one bit insertion feature 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/PEB 20534.

5 Detailed Protocol Description

5.1 HDLC/SDLC

In HDLC, all transmissions are in frames. The frame format structure does not include bits inserted for transparency.

5.1.1 General Frame Format

Each frame consists of the following fields (left to right transmission sequence, hexadecimal values):

Flag 7E	Addr-H 1 byte	Addr-L 1 byte	Info n bytes	Frame Checking Sequence			Flag 7E
				CRC8	CRC-CCITT	CRC32	
				1 byte	2 bytes	4 bytes	

Where:

`Flag` = Flag sequence
`Addr-H` = Station address field high byte
`Addr-L` = Station address field low byte
`Info` = Info field
`CRC8` = 8-bit frame checking sequence byte
`CRC-CCITT` = 16-bit frame checking sequence word
`CRC32` = 32-bit frame checking sequence dword

5.1.2 Elements of the Frame

5.1.2.1 Flag Sequence

All frames start and end with the flag sequence (0x7E). All data stations that are attached to the data link shall continuously hunt for this sequence. Thus, the flag is used for frame synchronization. A single flag may be used as both the closing flag for one frame and the opening flag for the next frame if the next frame **immediately** follows the previous frame without any idle line time (i.e., back-to-back).

5.1.2.2 Address Fields

The address field or fields immediately follow the opening flag and identify the station for which the command is intended. When more than one address field is used, they shall be present in the frame in a consecutive manner. If enabled, the receiver will scan for the appropriate 8 or 16-bit maskable address. If the address of the incoming frame does not match the address of the receiver, then the frame is rejected and the hunt for flag sequence is re-initiated. 8-bit addresses will use the Addr-H byte and the Addr-L byte will be ignored.

The Address High/Low bytes can be masked on a per bit basis by setting the corresponding bits in the mask register. This allows extended broadcast address recognition. Masked bit positions always match in comparison of the received frame address with the respective address fields. This feature is applicable to all HDLC protocol modes with address recognition. It is disabled if all bits of mask register are set to ‘zero’.

As an option the 8/16 bit address field of received frames can be pushed to the receive data buffer (first one/two bytes of the frame).

5.1.2.2.1 All-Station Address

The address field bit pattern 11111111 is defined as the all-station address. The all-station address may be used for all-station polling. When there is more than one receiving data station for which a command with the all-station is intended, any responses from these data stations shall not interfere with one another. If the 16 bit address scheme is in use, we will be comparing the first octet and ignoring the second octet.

5.1.2.2.2 No-Station Address

The bit pattern 00000000 in the first octet of the extended or non-extended address field is defined as the no-station address. The no-station address shall never be assigned to a data station. The no-station address may be used for testing when it is intended that no data station react or respond to a frame containing the no-station address. If the 16 bit address scheme is in use, we will be comparing the first octet and ignoring the second octet.

5.1.2.2.3 Group Address

The group address register is a second 16-bit maskable address comparison register that can be used to define custom group addresses. For example, in certain usages of HDLC, the address 0xFC or 0xFE can be used to address all stations in a group. This register is compared in addition to the primary address register. If the address comparison matches either of the registers, the frame will be received.

5.1.2.3 Info Field

Info field may be any sequence of 8-bit bytes of up to a maximum of MAXFRAMESIZE bytes.

Note: The data between the start and end flags must be byte aligned (8-bits long). For example, 5 bit data bytes are not allowed.

Note: MAXFRAMESIZE is 65535-2 if RLC is enabled, or 2^32 (effectively infinite) if RLC is disabled. Any sync, term, CRC or address bytes that are to be transferred to the FIFO will also have to be accounted for.

5.1.2.4 CRC Fields

Three frame checking sequences are specified; an 8-bit frame checking sequence (CRC8), a 16-bit frame checking sequence (CRC-CCITT), and a 32-bit frame checking sequence (CRC32). The 16-bit frame checking sequence is normally used. The 8-bit frame checking sequence is for use by prior agreement in those cases where short frames are used such that the protection afforded is adequate and/or the overhead of a longer frame checking sequence is of concern. The 32-bit frame checking sequence is for use by prior agreement in those cases that need a higher degree of protection than can be provided by the 16-bit frame checking sequence.

5.1.3 Data Reception

In HDLC mode, data will begin to be moved into the RxFIFO immediately following detection of the opening flag (0x7E). Optionally, if Address checking is enabled, the first byte or bytes of data will be checked against the selected station address. If the station address matches,

reception of data is continued. If the station address does not match, the received data is discarded and the receiver is returned to the non-synchronized, IDLE state.

Once synchronization is achieved, data will continue to be clocked in until one of the defined stop conditions is detected:

- The closing flag (0x7E) is detected
- The Receiver Reset command is issued
- The programmed Receive Length Check (RLC) has been reached
- An abort sequence is detected (seven contiguous '1' bits)

After any of the above stop conditions have been met:

- The receiver is to return to the non-synchronized (IDLE) state in which it is hunting for the opening flag.
- Reception of data is internally disabled until synchronization is regained.
- The current receive data block is to be marked as closed.
    - If CRC is enabled, the calculated CRC will be checked against appropriate CRC byte(s)
    - The RxStatus word will be appended to the data

5.1.4 Data Transmission

In HDLC mode, when data moved out of the TxFIFO and into the transmitter, the frame is assembled before being clocked out. The opening flag is prepended to the outbound data. The selected CRC algorithm will be run on the data as it moves through the transmitter. The final calculated CRC bytes are inserted into the appropriate field. Likewise, on its way through the transmitter, the data is parsed for places that need zero bit insertion (including the CRC bytes). Upon completion the closing flag is added and the data is clocked out.

- Optionally the CTS signal can be used to control data transmission.
- After completing data transmission, the selected Interframe time fill is automatically sent.

5.1.5 Shared Flag Mode

A single flag may be used as both the closing flag for one frame and the opening flag for the next frame if the next frame immediately follows the previous frame without any idle line time (i.e., back-to-back). The shared flag will act as a terminating sequence on the current frame and the opening sequence on the subsequent frame.

5.1.6 Invalid Frame

An invalid frame is defined as one that is not properly bounded by two flags or one that is too short (that is, shorter than 16 bits between flags when using the 8-bit CRC, shorter than 24 bits between flags when using the 16-bit CRC, and shorter than 40 bits between flags when using the 32-bit CRC). The hardware will mark the invalid frame as closed and will reset the VFR bit in the RxStatus word. The invalid frame will then be passed on to the RxFIFO and will be handled by software.

5.1.7 Aborted Frame

During reception of a frame, if there is ever detected seven or more contiguous ‘1’ bits, this shall be considered an abort command. If an abort command is detected, the hardware will set the ABF bit in the status register (6.2.7). The aborted frame will then be passed on to the RxFIFO and will be handled by software.

5.2 Character Oriented Synchronous (X-Sync) Mode

Character oriented protocols achieve synchronization between transmitting and receiving stations by means of special SYNC sequences.

5.2.1 General Frame Format

Each frame consists of the following fields (left to right transmission sequence, hexadecimal values):

X-Sync Sequence				Info n bytes	Frame Checking Sequence			Terminating Sequence			
SYNC1	SYNC2	SYNC3	SYNC4		CRC8	CRC-CCITT or CRC16	CRC32	TERM1	TERM2	TERM3	TERM4
1 byte	1 byte	1 byte	1 byte		1 byte	2 bytes	4 bytes	1 byte	1 byte	1 byte	1 byte

Where:

SYNC1	= First sync byte
SYNC2	= Optional second sync byte
SYNC3	= Optional third sync byte
SYNC4	= Optional fourth sync byte
Info	= N-byte data field
CRC8	= 8-bit frame checking sequence byte
CRC-CCITT	= 16-bit frame checking sequence word
CRC16	= 16-bit frame checking sequence word
CRC32	= 32-bit frame checking sequence dword
TERM1	= Optional first termination byte
TERM2	= Optional second termination byte
TERM3	= Optional third termination byte
TERM4	= Optional fourth termination byte

5.2.2 Elements of the Frame

5.2.2.1 X-Sync Sequence

The SYNC sequence can be one, two, three or four bytes long. Each bit of the SYNC sequence is maskable, meaning it can match anything, allowing you to receive frames with differing SYNC sequences.

5.2.2.2 Info Field

Info field may be any sequence of bits of up to MAXFRAMESIZE bytes.

Note: The data between the SYNC and TERM flags must be byte aligned (8-bits long). For example, 5 bit data bytes are not allowed.

Note: MAXFRAMESIZE is 65535-2 if RLC is enabled, or 2^32 (effectively infinite) if RLC is disabled. Any sync, term, CRC or address bytes that are to be transferred to the FIFO will also have to be accounted for.

5.2.2.3 CRC Fields

Four frame checking sequences are specified: an 8-bit frame checking sequence (CRC8); two 16-bit frame checking sequences (CRC-CCITT and CRC-16); and a 32-bit frame checking sequence (CRC32). The CRC-16 frame checking sequence is normally used. The 8-bit frame checking sequence is for use by prior agreement in those cases where short frames are used such that the protection afforded is adequate and/or the overhead of a longer frame checking sequence is of concern. The 32-bit frame checking sequence is for use by prior agreement in those cases that need a higher degree of protection than can be provided by the 16-bit frame checking sequence.

5.2.2.4 Termination Sequence

The programmed termination sequence can also be selected as one, two, three, four bytes or entirely disabled. Each bit of the TERM sequence is maskable, meaning it can match anything, allowing it to match multiple sequences.

5.2.3 Data Reception

In X-Sync mode, data will begin to be moved into the RxFIFO immediately following detection of the entire SYNC sequence (with the option of pushing the SYNC sequence itself into the RxFIFO).

Once synchronization is achieved, data will continue to be clocked in until one of the defined stop conditions is detected:

- The programmed termination sequence is detected
- The Receiver Reset command is issued
- There is a pulse on the Frame Sync Receive (FSR) signal (Mode 3)
- The programmed Receive Length Check (RLC) has been reached
- The HUNT command is issued

After any of the above stop conditions have been met:

- The receiver is to return to the non-synchronized, IDLE state in which it is hunting for the SYNC sequence.
- Reception of data is internally disabled until synchronization is regained.
- The current receive data block is to be marked as closed.
    - If CRC is enabled, the appropriate CRC byte(s) will be checked
    - The RxStatus word will be appended to the data

5.2.4 Data Transmission

- User data transmitted in X-Sync mode will be sent out following the programmed SYNC sequence.
- If enabled, the internally calculated CRC value is appended to the message frame.[1]
- Optionally the CTS signal can be used to control data transmission.

[1] Internally generated SYN characters are always excluded from CRC calculations

- If TERM sequence is enabled, the transmitter can automatically append the chosen TERM sequence. Optionally the TERM sequence can be left off.
- After completing data transmission, the selected Interframe time fill is automatically sent.

5.2.5 Shared Flag Mode

In X-Sync, this option allows the terminating sequence of the current frame to be used as the sync sequence of the next frame as long as that next frame immediately follows the current frame without any idle line time (i.e., back-to-back). The shared flag will act as the closing sequence for the current frame and the opening sequence for the subsequent frame.

5.3 Transparent Mode

In transparent mode, fully transparent data transmission and reception without any framing is performed (i.e. no flag or sync detection). This allows user-defined protocols to be used.

5.3.1 Data Reception

In the base transparent configuration, the receiver will be constantly enabled and will push everything that is on the line into the RxFIFO. There will be no frame delimiters in the data to mark the end of a frame. RxByteCount will count all bytes that enter into the RxFIFO and if its value reaches its maximum, it will simply reset and continue counting.

There are two ways to trigger a non-error end of frame condition. That would be with the Receive Length Check or the Frame Sync Receive signal.

Note: A frame end condition will still occur at the specified error conditions.

If enabled, the Receive Length Check can be used to signal an end of frame condition. The RxStatus word will be appended on to the received data.

If frame synchronization is enabled using the Frame Sync Receive (FSR) signal, reception of data will be enabled and/or disabled at the appropriate clock edges. In this case, the RxByteCount will equal the number of bytes received while data reception was enabled and the RxStatus word will be appended to the end of the received data.

5.3.2 Data Transmission

The transmitter will shift the data in the TxFIFO directly to the output. The transmitted data will be completely unmodified. When the transmitter is in an idle state, it will output either a continuous stream of 1s or a repeating byte as defined by the lower 8 bits of the SSR. If you want to output the SSR byte, you must set CCR0:ITF=1.

If frame synchronization is enabled, the generated Frame Sync Transmit (FST) signal will reflect the first and last bit of the transmitted data.

5.4 True-Asynchronous Mode

Each port can be operated in True-Asynchronous mode utilizing the on board 16C950 based UARTs. When the driver is switched to True-Asynchronous mode, the ports can be operated just like a standard serial (COM or tty) port. This will disable the synchronous FSCC interface and transfer board control over to the given COM or tty port. See FCR to enable this mode.

5.4.1 Features

- Baud rates up to 15 Mbps in normal mode
- Single full-duplex asynchronous channel
- 128-byte deep transmitter / receiver FIFO
- Fully software compatible with industry standard 16C550 type UARTs
- Flexible clock prescaler from 1 to 31.875
- 9-bit data framing as well as 5,6,7 and 8
- Detection of bad data in the receiver FIFO
- Transmitter and receiver can be disabled
- Programmable special character detection
- Arbitrary trigger levels for receiver and transmitter FIFO interrupts and automatic in-band and out-of-band flow control
- Automatic handling of RS485 (i.e. the line driver tri-states when idle)
- Receive echo cancel. Disables the line receiver during transmits to prevent receiving the data that is transmitted in 2-wire 485 modes.
- Variable frame length (number of bytes between start and stop bits)

6 Register Description

All of our current 4-port cards are two 2-port cards with a bridge chip. The following register descriptions are for a 2-port card. Ports 3 and 4 of a 4-port card will be the same.

6.1 Base Address Registers

Table 2 – Base Address Registers descriptions

Name	Address	Size	Description	Page
BAR0	0x000 – 0x054	256 Bytes	F-Core Port A	61
	0x080 – 0x0D4		F-Core Port B	61
BAR1	0x000 – 0x007	128 Bytes	16C950 Port A	96
	0x008 – 0x00F		16C950 Port B	96
BAR2	0x000 – 0x07F	128 Bytes	Board Control + DMA	101

6.2 F-Core Registers

Table 3 – F-Core Register Descriptions

Address		Access Type	Register		Page
Port A	Port B		Name	Description	
0x00	0x80	W	TxFIFO	Transmit FIFO	62
		R	RxFIFO	Receive FIFO	62
0x04	0x84	W	TxByteCount	Transmit Byte Count FIFO	62
		R	RxByteCount	Receive Byte Count FIFO	62
0x08	0x88	R/W	FIFO Trigger	Transmit and Receive FIFO Trigger Level	62
0x0C	0x8C	R	FIFO Byte Count	Number of bytes in the FIFOs	63
0x10	0x90	R	FIFO Frame Count	Number of queued frames in the FIFOs	64
0x14	0x94	W	CMDR	Command Register	65
0x18	0x98	R	STAR	Status Register	67
0x1C	0x9C	R/W	CCR0	Channel Configuration Register 0	71
0x20	0xA0	R/W	CCR1	Channel Configuration Register 1	75
0x24	0xA4	R/W	CCR2	Channel Configuration Register 2	79
0x28	0xA8	R/W	BGR	Baud-rate Generator Register	81
0x2C	0xAC	R/W	SSR	Synchronization Sequence Register	82
0x30	0xB0	R/W	SMR	Synchronization Mask Register	83
0x34	0xB4	R/W	TSR	Termination Sequence Register	84
0x38	0xB8	R/W	TMR	Termination Mask Register	85
0x3c	0xBC	R/W	RAR	Receive Address Register	86
0x40	0xC0	R/W	RAMR	Receive Address Mask Register	87
0x44	0xC4	R/W	PPR	Preamble & Postamble Register	88
0x48	0xC8	R/W	TCR	Timer Control Register	89
0x4c	0xCC	R	VSTR	Version Status Register	90
0x50	0xD0	R	ISR	Interrupt Status Register	91
0x54	0xD4	R/W	IMR	Interrupt Mask Register	94
0x58	0xD8	R/W	DPLLR	DPLL Reset control Register	95

6.2.1 FIFO locations (0x00/0x80)

Writing data to the TxFIFO is done with 32-bit dword writes to this address space.

Reading data from the RxFIFO is done with 32-bit dword reads from this address space.

6.2.2 Byte Count FIFO locations (0x04/0x84)

The TxByteCount FIFO is the total number of data bytes in the corresponding frame in the TxFIFO. This count will be written here after the data has been written into the TxFIFO.

The RxByteCount FIFO is the total number of data bytes in the corresponding frame plus the RxStatus word plus any other bytes that have been pushed to the FIFO. This number is the total number of bytes actually written to the RxFIFO. This count will be written after the frame has been completed.

6.2.3 FIFO Trigger Levels (0x08/0x88)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	TXTRG11	TXTRG10	TXTRG9	TXTRG8	TXTRG7	TXTRG6	TXTRG5	TXTRG4	TXTRG3	TXTRG2	TXTRG1	TXTRG0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	RXTRG12	RXTRG11	RXTRG10	RXTRG9	RXTRG8	RXTRG7	RXTRG6	RXTRG5	RXTRG4	RXTRG3	RXTRG2	RXTRG1	RXTRG0

TXTRG[11:0] – Transmit FIFO Trigger Level

A 12-bit value written to this location sets the TxFIFO trigger level from 0x004 (four) to 0xFFC (4092) bytes . The TxFIFO trigger level generates a TFT interrupt whenever the data level in the transmit FIFO falls below this preset trigger level.

RXTRG[12:0] – Receive FIFO Trigger Level

A 13-bit value written to this location sets the RxFIFO trigger level from 0x04 (four) to 0x1FFC (8188) bytes. The RxFIFO trigger level generates an interrupt whenever the receive FIFO rises above this preset trigger level.

Note: In terms of efficiency, it is best to set the FIFO levels to DWORD aligned values (i.e. divisible by 4) because all bus transfers happen as DWORDs, so setting a non-DWORD value does not gain anything.

6.2.4 FIFO Byte Count (0x0C/0x8C)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	TXCNT12	TXCNT11	TXCNT10	TXCNT9	TXCNT8	TXCNT7	TXCNT6	TXCNT5	TXCNT4	TXCNT3	TXCNT2	TXCNT1	TXCNT0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	RXCNT13	RXCNT12	RXCNT11	RXCNT10	RXCNT9	RXCNT8	RXCNT7	RXCNT6	RXCNT5	RXCNT4	RXCNT3	RXCNT2	RXCNT1	RXCNT0

TXCNT[12:0] – Transmit FIFO Level Counter

When read, this 13-bit value will return the total number of bytes currently in the transmit FIFO from 0x00 (zero) to 0x1000 (4096) bytes.

RXCNT[13:0] – Receive FIFO Level Counter

When read, this 14-bit value will return the total number of bytes currently in the receive FIFO from 0x00 (zero) to 0x2000 (8192) bytes.

6.2.5 FIFO Frame Count (0x10/0x90)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	TFCNT8	TFCNT7	TFCNT6	TFCNT5	TFCNT4	TFCNT3	TFCNT2	TFCNT1	TFCNT0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	RFCNT9	RFCNT8	RFCNT7	RFCNT6	RFCNT5	RFCNT4	RFCNT3	RFCNT2	RFCNT1	RFCNT0

TFCNT[8:0] – Transmit FIFO Frame Count

When read, this 9-bit value returns the number of complete frames that are queued in the TxFIFO from 0x00 (zero) to 0x100 (256). This is essentially the number of entries in the TxByteCount FIFO.

RFCNT[9:0] – Receive FIFO Frame Count

When read, this 10-bit value returns the number of complete frames that are queued in the RxFIFO from 0x00 (zero) to 0x200 (512). This is essentially the number of entries in the RxByteCount FIFO.

6.2.6 CMDR – Command Register (0x14/0x94)

				Transmit Commands								Receive Commands			
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	TXEXT	TXT	TRES	SXREP	XREP	XF	0	0	0	0	0	DRC	RRES	HUNT

												General Commands			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	STIMR	TIMR

Note 1 - All bits of the command register are self-clearing. All command bits are executed at detection of a logic level '1'.

TIMR – Start Timer Command

This command starts the hardware timer. Whenever the timer expires, a TIMR interrupt is generated.

STIMR – Stop Timer Command

This command stops the hardware timer. When executed, the timer will immediately terminate operation and will reset back to its initial state.

HUNT – Enter HUNT State Command

This command forces the receiver to enter its HUNT state immediately. Thus synchronization is released and the receiver begins searching for new SYNC sequences.

RRES – Receiver Reset Command

The receive FIFO is cleared and the receiver is reset to its initial state. The receive FIFO accepts new receive data immediately after receiver reset has completed.

DRC – Detect Receive Clock

This command starts the receive clock detect circuit. It will trigger an interrupt when it detects three clock edges. Issuing the command again will reset the clock count.

XF – Transmit Frame

After having written data to the TxFIFO, this command initiates the transmission of the frame. The appropriate opening sequence is automatically added to the data by the F-Core. See the Transmit Data Flow section 3.2 for more details.

XREP – Transmit Repeatedly

Operates the same as XF command, except that this command will transmit the given data repeatedly until stopped. See the Transmit Data Flow section 3.2 for more details.

SXREP – Stop Transmitting Repeatedly

This command signals the transmitter to stop sending the repeated frame at the end of the current frame.

TRES – Transmitter Reset

When executed, the controller will immediately stop transmitting the current frame and switch to the IDLE sequence. The transmit FIFO is cleared of any data and the transmitter is reset to its initial state. The transmit FIFO accepts new transmit data immediately after transmitter reset has completed.

TXT – Transmit on Timer

Use in conjunction with the XF or XREP commands to transmit a single, complete frame on the expiration of the timer. For example, if you wanted to transmit a single frame repeatedly on the expiration of the timer, you would send both the TXT and the XREP command at the same time.

See section 3.2.1.3 for more details on transmit on timer.

TXEXT – Transmit on External Signal

Use in conjunction with the XF command to transmit a single, complete frame on the transition of an external signal (CTS, DSR, RI, CD) as chosen via CCR0:EXTSEL. Similar in operation to the TXT command, this uses a transition on an external signal to trigger a single frame send.

See section 3.2.1.4 for more details on transmit on external signal.

6.2.7 STAR – Status Register (0x18/0x98)

												Controller Status			
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	TDO	0	TXEA	TXTA	XREPA	DPLLA	CE	TBC	TFE

Receive Status Word															
FIFO Status				Frame Status								Input Pin Status			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	RDO	RFL	VFR	RLEX	CRC	ABF	0	0	0	0	RI	CD	DSR	CTS

Note 1: Bit 24 will be reset upon a read of the status register. Bits 13 through 8 will be reset upon the pushing of the RxStatus word into the RxFIFO. Bits 21 through 16 will not reset and will always correspond to the current state of those conditions. The last four bits will not reset and will always correspond to the present state of the input pins.

Note 2: The Receive Status Word will show up as the last two bytes of your data. It is read least significant byte first. So the last two bytes of your data will read 0x0802 for a CRC error and an active RI signal

CTS – Clear To Send

Reflects the current state of the CTS input pin. Does not reset.

DSR – Data Set Ready

Reflects the current state of the DSR input pin. Does not reset.

CD – Carrier Detect

Reflects the current state of the CD input pin. Does not reset.

RI – Ring Indicator

Reflects the current state of the RI input pin. Does not reset.

ABF – Abort Frame

Indicates that an abort sequence was detected during reception of the current frame. Resets on RxStatus word push to RxFIFO.

- 0 – Normal
- 1 – ABF occurred

CRC – Cyclic Redundancy Check

Reflects the state of the current message's CRC comparison/check. Resets on RxStatus word push to RxFIFO.

- 0 – The check passed

- 1 – The check failed

RLEX – Receive Length Exceeded

When using the Receive Length Check feature, this bit will indicate that the frame has exceeded the specified size and was aborted. Resets on RxStatus word push to RxFIFO.

- 0 – Receive Length Check has not been exceeded or is not in use
- 1 – Receive Length has been exceeded

Note: if you are in HDLC mode and you trigger an RLEX then it is likely that something went wrong with this frame and it should be considered invalid.

VFR – Valid Frame

Determine whether a valid frame has been received. Resets on RxStatus word push to RxFIFO.

- 0 – Valid frame
- 1 – Invalid frame

An invalid frame is:

- A frame which is not an integer number of 8 bits (n × 8 bits) in length (e.g. 25 bits)
- A frame will contain a minimum of 1 byte. If Address checking is enabled, one or two bytes (for 8 or 16-bit addresses respectively) will be added to the minimum frame size. If CRC checking is enabled, one, two or three bytes are to be added to the minimum frame size (for 8, 16 and 32-bit CRCs respectively). For example, an 8-bit address (1 byte) and a 16-bit CRC (2 bytes) added to the minimum 1 data byte will result in a minimum frame size of 4 bytes.
- Additional criteria for marking a frame invalid may be defined for individual operating modes.

Note: Invalid frames are marked as ended and passed on to the RxFIFO

RFL – Received Frame Lost

Indicates that, previous to this frame, there had been an un-serviced RDO and more data came in. This is indicating that there was at least one complete frame lost. Resets on RxStatus word push to RxFIFO.

- 0 – Normal
- 1 – RFL occurred

RDO – Receive Data Lost

Indicates that the data in this message caused the RxFIFO to overflow its 8K boundary. This is also an indication that the current message is corrupt. Resets on RxStatus word push to RxFIFO.

- 0 – Normal

- 1 – RDO occurred

TFE – Transmit FIFO Empty

This bit will be set when the transmit FIFO becomes completely empty, allowing the ISR to complete one large data transfer of up to the TxFIFO size (4096 bytes). Does not reset.

- 0 – the TxFIFO is not empty
- 1 – the TxFIFO is empty

TBC – Transmit Byte Count Full

When the TxByteCount FIFO becomes full (after the 256th frame has been queued), this bit will be set to tell the ISR not to attempt to push any more frames. Does not reset.

- 0 – The TxByteCount FIFO is not full
- 1 – The TxByteCount FIFO is full

CE – Command Executing

This bit will be active during the execution cycle of a given command. Does not reset.

- 0 – No command is currently being executed (can send a new command)
- 1 – A previously executed command is executing (no new commands can be sent)

DPLLA – DPLL Asynchronous

This bit is only valid if the receive clock is recovered by the DPLL. It is set when the DPLL has lost synchronization. In this case reception is disabled (receive abort condition) until synchronization has been regained. In addition transmission is interrupted in clock mode 5. Interruption of transmission is performed the same way as on deactivation of the CTS signal. Does not reset.

This bit will be reset at the beginning of each received frame.

- 0 – Normal operation, DPLL synchronized or not in use
- 1 – DPLL not synchronized (asynchronous)

XREPA – XREP Command Active

Indicates whether the XREP command is currently active. Does not reset

- 0 – XREP command not active
- 1 – XREP command currently active

TXTA – Transmit by Timer Active

Indicates whether the transmit by timer command (TXT) is active. Does not reset

- 0 – TXT command not active
- 1 – TXT command currently active

TXEA –Transmit by External Signal Active

- 0 – TXEXT command not active

- 1 – TXEXT command active

TDO – Transmit Data Overflow

This flag is set when data is written to the TxFIFO that would push the TxFIFO beyond the 4K boundary (shouldn't happen, but you never can tell). Resets on read of Status register.

- 0 – Normal operation
- 1 – Overflow condition

CCR0 – Channel Configuration Register 0 (0x1C/0x9C)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	EXTS1	EXTS0	0	0	RECD	ADM1	ADM0	OBT	CRC1	CRC0	VIS	NTB2	NTB1	NTB0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NSB2	NSB1	NSB0	ITF	SFLAG	FSC2	FSC1	FSC0	LE2	LE1	LE0	CM2	CM1	CM0	MODE1	MODE0

MODE[1:0] – Mode select

- MODE[1:0] = 00 Default. Enable HDLC operating mode

  HDLC operating mode will set some configuration information that is mandated by the HDLC specification and prevent those configuration bits from being altered:

  - # of Sync Bytes = 1 CCR0:NSB[3:0] = 001
  - # of Term Bytes = 1 CCR0:NTB[3:0] = 001
  - Sync = 0x7E SSR:SYNC[31:0] = 0x0000007E
  - Term = 0x7E TSR:TERM[31:0] = 0x0000007E
  - 0 bit insertion – On CCR1:ZINS = 1
  - Transmitter appends TERM sequence CCR1:DTERM=0

  Note these two settings are what is generally considered to be used with HDLC, however they can be changed to suit your specific application.

  - CRC = CCITT CCR0:CRC[1:0] = 01
  - CRC Reset = 0xFFFF CCR1:CRCR = 0
- MODE[1:0] = 01 Enable X-Sync operating mode
- MODE[1:0] = 10 Enable Transparent operating mode

CM[2:0] – Clock Mode Select

This bit field selects one of the eight clock modes. For a detailed description of the clock modes, refer to section 4.1.

- CM = 000 clock mode 0
- CM = 001 clock mode 1
- CM = 010 clock mode 2

- CM = 011 clock mode 3
- CM = 100 clock mode 4
- CM = 101 clock mode 5
- CM = 110 clock mode 6
- CM = 111 clock mode 7

LE[2:0] – Line Encoding

This bit field selects the line encoding of the serial port. For a detailed description of the encoding options, refer to section 4.2.

- LE = 000 NRZ data encoding
- LE = 001 NRZI data encoding
- LE = 010 FM0 data encoding
- LE = 011 FM1 data encoding
- LE = 100 Manchester data encoding
- LE = 101 Differential Manchester data encoding
- LE = 110 reserved
- LE = 111 reserved

FSC[2:0] – Frame Sync Control

This bit field selects the behavior of the Frame Sync signals. For a detailed description of the Frame Sync signals, refer to section 4.5.

- FSC = 000 No Frame Sync
- FSC = 001 Mode 1
- FSC = 010 Mode 2
- FSC = 011 Mode 3
- FSC = 100 Both Mode 2 and Mode 3 at the same time.
- FSC = 101 Mode 5
- FSC = 110 reserved
- FSC = 111 reserved

SFLAG – Shared Flag Mode

This bit enables shared flag transmission and reception in HDLC and X-Sync protocol modes. If more than one transmit frame start is stored in the transmit FIFO, the closing flag of the preceding frame becomes the opening flag of the next frame.

- SFLAG = 0 Default. Do not share flags
- SFLAG = 1 Enable shared flags

Note: Enabling shared flags will override and disable preambles and postambles.

ITF – Inter-frame Time Fill

This bit selects the idle state of the transmit data pin.

- ITF = 0 Continuous logical '1' is sent during idle periods
- ITF = 1 Continuous SYNC sequences are sent during idle periods

NSB[2:0] – Number of Sync Bytes

This bit field selects the number of sync bytes to use when synchronizing to data.

- NSB = 000 No synchronization bytes used (e.g. Transparent Mode)
- NSB = 001 One byte. Use byte SYNC1 to synchronize (forced when CCR0:HDLC=1).
- NSB = 010 Two bytes. Use bytes SYNC[2:1] to synchronize
- NSB = 011 Three bytes. Use bytes SYNC[3:1] to synchronize
- NSB = 100 Four bytes. Use bytes SYNC[4:1] to synchronize

NTB[2:0] – Number of Termination Bytes

This bit field selects the number of terminating bytes to use when detecting frame end.

- NTB = 000 No termination bytes used (e.g. Transparent Mode)
- NTB = 001 One byte. Use byte TERM1 to terminate (forced when CCR0:HDLC=1).
- NTB = 010 Two bytes. Use bytes TERM[2:1] to terminate.
- NTB = 011 Three bytes. Use bytes TERM[3:1] to terminate.
- NTB = 100 Four bytes. Use bytes TERM[4:1] to terminate.

VIS – Masked Interrupts Visible

- VIS = 0 Default. Masked interrupt status bits are not visible on interrupt status register (ISR) read accesses.
- VIS = 1 Masked interrupt status bits are visible and automatically cleared on interrupt status register (ISR) read accesses. These interrupts will not generate an interrupt to the interrupt controller, they will only be visible in ISR.

CRC[1:0] – CRC Frame Check Mode

This bit field selects the algorithm used in calculating CRC.

- CRC = 00 Use CRC-8 algorithm.
- CRC = 01 Use CRC-CCITT algorithm (forced when CCR0:HDLC=1).
- CRC = 10 Use CRC-16 algorithm.
- CRC = 11 Use CRC-32 algorithm.

OBT – Order of Bit Transmission

Specifies whether data is considered least significant bit (LSB) first or most significant bit (MSB) first. See section 4.8 for more details.

- OBT = 0 LSB first
- OBT = 1 MSB first

ADM[1:0] – Address Mode (HDLC only)

This bit field selects the number of bytes used in receive address comparison.

- ADM = 00 No address checking.
- ADM = 01 1 byte address checking.
- ADM = 10 2 byte address checking.
- ADM = 11 Reserved

RECD – Receiver Disable

When set, this bit disables the receiver. This can be useful, for example, when in Transparent mode and transmitting only. You would not want to receive an endless stream of idles which do nothing but generate unnecessary interrupts.

- RECD = 0 Receiver enabled.
- RECD = 1 Receiver disabled.

EXTS[1:0] – External Signal Select

These bits select the external input source pin for the TXEXT command.

- EXTS = 00 CTS selected
- EXTS = 01 DSR selected
- EXTS = 10 RI selected
- EXTS = 11 CD selected

6.2.8 CCR1 – Channel Configuration Register 1 (0x20/0xA0)

Input / Output Pin Polarities

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	RDP	TDP	RCP	TCIP	TCOP	FSTP	FSRP	DSRP	DTRP	CTSP	RTSP	CDP	RIP

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DSYNC	DTERM	DTCRC	DRCRC	CRCR	CRC2F	ADD2F	TERM2F	SYNC2F	DPS	OINS	ZINS	CTSC	RTSC	DTR	RTS

RTS – Request To Send Pin State

When the RTS pin's control is set to manual (CCR1:RTSC=1), this bit sets the state of the RTS output pin. If CCR1:RTSC=0 this bit is ignored.

- RTS = 1 Active
- RTS = 0 Inactive

DTR – Data Terminal Ready Pin State

This bit sets the state of the DTR output pin.

- DTR = 1 Active
- DTR = 0 Inactive

RTSC – RTS Control

Select between manual and automatic mode.

- RTSC = 0 Automatic mode; the RTS pin is used to facilitate RTS-CTS handshaking. See section 4.6.1.
- RTSC = 1 Manual mode; the RTS pin is a general purpose output controlled with the CCR1:RTS bit.

CTSC – CTS Control

Select between manual and automatic modes

- CTSC = 0 Automatic mode; the CTS signal is used to activate the transmitter to facilitate RTS-CTS handshaking. See section 4.6.2.
- CTSC = 1 Manual mode; the CTS pin is a general-purpose input whose status can be found in the STAR.

ZINS – Zero Insertion Control

Enable HDLC specified zero bit insertion. See section 4.14.

- ZINS = 0 Zero insertion disabled.
- ZINS = 1 Zero insertion enabled (forced when CCR0:HDLC=1).

OINS – One Insertion Control

Enable special one bit insertion. See section 4.15.

- OINS = 0 One insertion disabled.
- OINS = 1 One insertion enabled.

DPS – DPLL Phase Shift Disable

- DPS = 0 Normal DPLL operation
- DPS = 1 The phase shift function of the DPLL is disabled. The windows for phase adjustment are extended.

SYNC2F – Transfer SYNC character(s) to RxFIFO (X-Sync mode only)

- SYCN2F = 0 Default operation; Synchronization characters are stripped from the data stream before transferring to the RxFIFO.
- SYNC2F = 1 Synchronization characters are not stripped and are forwarded to the RxFIFO as part of the valid data stream.

TERM2F – Transfer TERM character(s) to RxFIFO (X-Sync mode only)

- TERM2F = 0 Default operation; Termination characters are stripped from the data stream before transferring to the RxFIFO.
- TERM2F = 1 Termination characters are not stripped and are forwarded to the RxFIFO as part of the valid data stream.

  Note: This bit will be ignored if shared flags are enabled (CCR0:SFLAG) and the termination characters will not be forwarded to the RxFIFO.

ADD2F – Address byte(s) to RxFIFO (HDLC mode only)

- ADD2F = 0 Default operation; Receive addresses are stripped from the data stream before transferring to the RxFIFO.
- ADD2F = 1 Receive addresses are not stripped and are forwarded to the RxFIFO as part of the valid data stream.

CRC2F – CRC bytes to RxFIFO

- CRC2F = 0 Default operation; CRC bytes are stripped from the data stream before transferring to the RxFIFO.
- CRC2F = 1 CRC bytes are not stripped and are forwarded to the RxFIFO as part of the valid data stream.

CRCR – CRC reset level

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

- CRCR = 0 Default operation; Initialize to ones. Valid for most HDLC/SDLC operations.
- CRCR = 1 Initialize to zeros.

DRCRC – Disable receive CRC checking

- DRCRC = 0 Default operation; the receiver expects an 8, 16 or 32 bit CRC within a frame. Frames shorter than expected are marked invalid (see STAR:VFR).
- DRCRC = 1 The receiver does not expect any CRC within a frame. The criteria for valid frame (STAR:VFR) indication is updated accordingly. Bit CRC2F is ignored.

DTCRC – Disable transmit CRC insertion

- DTCRC = 0 Default. The CRC is to be calculated and appended to the data stream as per the CRC configuration bits.
- DTCRC = 1 A calculated CRC is not appended onto the supplied data stream. It will be assumed that the last byte or bytes of user-supplied data contains valid CRC data. Bits CRCR and CRC[1:0] are ignored.

DTERM – Disable appending TERM to transmit data

- DTERM = 0 Default. The transmitter will automatically append the selected TERM sequence to the supplied data stream immediately following the CRC (if enabled) or the data (if no CRC).
- DTERM = 1 The transmitter will NOT append the selected TERM sequence to the supplied data stream.

DSYNC – Disable appending SYNC to transmit data *(added in firmware version 22)*

- DSYNC = 0 Default. The transmitter will automatically prepend the selected SYNC sequence to the supplied data stream.
- DSYNC = 1 The transmitter will NOT prepend the selected SYNC sequence to the supplied data stream.

RIP – RI pin polarity

- RIP = 0 Default. RI pin is active low.
- RIP = 1 RI pin is active high.

CDP – CD pin polarity

- CDP = 0 Default. CD pin is active high.
- CDP = 1 CD pin is active low.

RTSP – RTS pin polarity

- RTSP = 0 Default. RTS pin is active low.
- RTSP = 1 RTS pin is active high.

CTSP – CTS pin polarity

- CTSP = 0 Default. CTS pin is active low.

- CTSP = 1 CTS pin is active high.

DTRP – DTR pin polarity

- DTRP = 0 Default. DTR pin is active low.
- DTRP = 1 DTR pin is active high.

DSRP – DSR pin polarity

- DSRP = 0 Default. DSR pin is active low.
- DSRP = 1 DSR pin is active high.

FSRP – Frame Sync Receive pin polarity

- FSRP = 0 Default. FSR pin is active high.
- FSRP = 1 FSR pin is active low.

FSTP – Frame Sync Transmit pin polarity

- FSTP = 0 Default. FST pin is active high.
- FSTP = 1 FST pin is active low.

TCOP – Transmit Clock Output pin polarity

- TCOP = 0 Default. TxClkOut pin is active high.
- TCOP = 1 TxClkOut pin is active low.

TCIP – Transmit Clock Input pin polarity

- TCIP = 0 Default. TxClkIn pin is active high.
- TCIP = 1 TxClkIn pin is active low.

RCP – RxClk pin polarity

- RCP = 0 Default. RxClk pin is active high.
- RCP = 1 RxClk pin is active low.

TDP – Transmit Data pin polarity

- TDP = 0 Default. TxD pin is active high.
- TDP = 1 TxD pin is active low.

RDP – Receive Data pin polarity

- RDP = 0 Default. RxD pin is active high.
- RDP = 1 RxD pin is active low.

6.2.9 CCR2 – Channel Configuration Register 2 (0x24/0xA4)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RLC15	RLC14	RLC13	RLC12	RLC11	RLC10	RLC9	RLC8	RLC7	RLC6	RLC5	RLC4	RLC3	RLC12	RLC1	RLC0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	FSTOT3	FSTOT2	FSTOT1	FSTOT0	FST0L3	FST0L2	FST0L1	FSTL0	FSRO3	FSRO2	FSRO1	FSRO0

FSRO[3:0] – Frame Sync Receive Offset

This value is a two's-complement number (range: -8 to +7) that defines the number of clocks by which the FSR signal PRECEDES the start/end of data. A value of 0 would result in the signals aligning as shown in Figure 7 - Frame Sync Modes. For a value of +1 (4'b0001), the frame sync signal would be one clock earlier on the diagram. For a value of -1 (4'b1111), frame sync would be one clock later on the diagram.

FSRO[3:0] = 0000 Default. No offset.

Num clocks before 1st data edge		**Num clocks after 1st data edge**	
FSRO[3:0] = 0001	1 clock	FSRO[3:0] = 1111	1 clock
FSRO[3:0] = 0010	2 clocks	FSRO[3:0] = 1110	2 clocks
FSRO[3:0] = 0011	3 clocks	FSRO[3:0] = 1101	3 clocks
FSRO[3:0] = 0100	4 clocks	FSRO[3:0] = 1100	4 clocks
FSRO[3:0] = 0101	5 clocks	FSRO[3:0] = 1011	5 clocks
FSRSO[3:0] = 0110	6 clocks	FSRO[3:0] = 1010	6 clocks
FSRO[3:0] = 0111	7 clocks	FSRO[3:0] = 1001	7 clocks
		FSRO[3:0] = 1000	8 clocks

FSTOL[3:0] – Frame Sync Transmit Offset (Leading edge in Mode 5)

This value is a two's-complement number (range: -8 to +7) that defines the number of clocks by which the FST signal is offset from the leading/trailing edge of data. A value of 0 would result in the signals aligning as shown in Figure 7 - Frame Sync Modes. For a value of +1 (4'b0001), the frame sync signal would be one clock earlier on the diagram. For a value of -1 (4'b1111), frame sync would be one clock later on the diagram.

Note: In Frame Sync Mode 5 this value adjusts only the leading edge of the FST signal.

FSTO[3:0] = 0000 Default. No offset.

Num clocks before 1st data edge		**Num clocks after 1st data edge**	
FSTOL[3:0] = 0001	1 clock	FSTOL[3:0] = 1111	1 clock
FSTOL[3:0] = 0010	2 clocks	FSTOL[3:0] = 1110	2 clocks
FSTOL[3:0] = 0011	3 clocks	FSTOL[3:0] = 1101	3 clocks
FSTOL[3:0] = 0100	4 clocks	FSTOL[3:0] = 1100	4 clocks
FSTOL[3:0] = 0101	5 clocks	FSTOL[3:0] = 1011	5 clocks
FSTOL[3:0] = 0110	6 clocks	FSTOL[3:0] = 1010	6 clocks
FSTOL[3:0] = 0111	7 clocks	FSTOL[3:0] = 1001	7 clocks
		FSTOL[3:0] = 1000	8 clocks

FSTOT[3:0] – Frame Sync Transmit Offset (Trailing edge)

This value is a two's-complement number (range: -8 to +7) that defines the number of clocks by which the FST signal is offset from the trailing edge of the last bit of data.

Note: This value is only applicable if operating in Frame Sync Mode 5.

FSTO[3:0] = 0000 Default. No offset.

Num clocks before last data		**Num clocks after last data**	
FSTOT[3:0] = 0001	1 clock	FSTOT[3:0] = 1111	1 clock
FSTOT[3:0] = 0010	2 clocks	FSTOT[3:0] = 1110	2 clocks
FSTOT[3:0] = 0011	3 clocks	FSTOT[3:0] = 1101	3 clocks
FSTOT[3:0] = 0100	4 clocks	FSTOT[3:0] = 1100	4 clocks
FSTOT[3:0] = 0101	5 clocks	FSTOT[3:0] = 1011	5 clocks
FSTOT[3:0] = 0110	6 clocks	FSTOT[3:0] = 1010	6 clocks
FSTOT[3:0] = 0111	7 clocks	FSTOT[3:0] = 1001	7 clocks
		FSTOT[3:0] = 1000	8 clocks

RLC[15:0] – Receive Length Check

This is a way to limit the number of received bytes for a single frame, making sure we have a way to terminate runaway frames. If the number of bytes actually received exceeds RLC-2 (for the RxStatus word) then the frame will be terminated with a message end condition. In other words, the size of the received data will be RLC + 3. RLC will count all of the incoming data including any extra data that is stripped away (like SYNC). See section 4.4.

- RLC[15:0] = 0 Default. Receive length check disabled.
- RLC[15:0] = 1 – 65535 Receive length check enabled.

6.2.10 BGR – Baud-Rate Generator Register (0x28/0xA8)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BGR31	BGR30	BGR29	BGR28	BGR27	BGR26	BGR25	BGR24	BGR23	BGR22	BGR21	BGR20	BGR19	BGR18	BGR17	BGR16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BGR15	BGR14	BGR13	BGR12	BGR11	BGR10	BGR9	BGR8	BGR7	BGR6	BGR5	BGR4	BGR3	BGR2	BGR1	BGR0

This register determines the division factor of the internal baud rate generator. The baud rate generator input clock frequency will be divided by the value of this register +1.

- BGR[31:0] = 0 Default. Divide by 1 (i.e. use the clock directly)
- BGR[31:0] = 1 – 4,294,967,295

Baud rate = clock frequency / (BGR + 1)

6.2.11 SSR – Synchronization Sequence Register (0x2C/0xAC)

SYNC4								SYNC3							
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SYNC31	SYNC30	SYNC29	SYNC28	SYNC27	SYNC26	SYNC25	SYNC24	SYNC23	SYNC22	SYNC21	SYNC20	SYNC19	SYNC18	SYNC17	SYNC16

SYNC2								SYNC1							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SYNC15	SYNC14	SYNC13	SYNC12	SYNC11	SYNC10	SYNC9	SYNC8	SYNC7	SYNC6	SYNC5	SYNC4	SYNC3	SYNC2	SYNC1	SYNC0

This register holds the characters used for achieving synchronization. The bit field CCR0:NSB determines which sync bytes will be used for synchronization.

If HDLC mode is selected, SYNC1 field will be automatically preset to 0x7E and all the rest of the bytes will be ignored.

6.2.12 SMR – Synchronization Mask Register (0x30/0xB0)

SM4								SM3							
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SM31	SM30	SM29	SM28	SM27	SM26	SM25	SM24	SM23	SM22	SM21	SM20	SM19	SM18	SM17	SM16

SM2								SM1							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SM15	SM14	SM13	SM12	SM11	SM10	SM9	SM8	SM7	SM6	SM5	SM4	SM3	SM2	SM1	SM0

- This register holds the values used to mask the synchronization characters. Setting one of these bits to '1' masks the corresponding bit in the SSR register. A masked bit position always matches when comparing the received data to the corresponding SSR bit.SM[31:0] = 0 Default. The bit position is NOT MASKED. This bit position in the SSR must match the incoming data to synchronize to a frame.
- SM[31:0] = 1 The bit position IS MASKED. This bit position in the SSR need not match the incoming data to synchronize to a frame.

6.2.13 TSR – Termination Sequence Register (0x34/0xB4)

TERM4								TERM3							
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TERM31	TERM30	TERM29	TERM28	TERM27	TERM26	TERM25	TERM24	TERM23	TERM22	TERM21	TERM20	TERM19	TERM18	TERM17	TERM16

TERM2								TERM1							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TERM15	TERM14	TERM13	TERM12	TERM11	TERM10	TERM9	TERM8	TERM7	TERM6	TERM5	TERM4	TERM3	TERM2	TERM1	TERM0

This register holds the characters used for terminating a frame. The bit field CCR0:NTB determines which sync bytes will be used for termination.

If HDLC mode is selected, TERM1 field will be automatically preset to 0x7E and all the rest of the bytes will be ignored.

6.2.14 TMR – Termination Mask Register (0x38/0xB8)

TM4								TM3							
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TM31	TM30	TM29	TM28	TM27	TM26	TM25	TM24	TM23	TM22	TM21	TM20	TM19	TM18	TM17	TM16

TM2								TM1							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TM15	TM14	TM13	TM12	TM11	TM10	TM9	TM8	TM7	TM6	TM5	TM4	TM3	TM2	TM1	TM0

This register holds the values used to mask the termination characters. Setting one of these bits to '1' masks the corresponding bit in the TSR register. A masked bit position always matches when comparing the received data to the corresponding TSR bit.

- TM[31:0] = 0 Default. The bit position is NOT MASKED. This bit position in the TSR must match the incoming data to terminate a frame.
- TM[31:0] = 1 The bit position IS MASKED. This bit position in the TSR need not match the incoming data to terminate a frame

6.2.15 RAR – Receive Address Register (0x3C/0xBC)

Station Address

Station Addr-H								Station Addr-L							
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SAH7	SAH6	SAH5	SAH4	SAH3	SAH2	SAH1	SAH0	SAL7	SAL6	SAL5	SAL4	SAL3	SAL2	SAL1	SAL0

Group Address

Group Addr-H								Group Addr-L							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GAH7	GAH6	GAH5	GAH4	GAH3	GAH2	GAH1	GAH0	GAL7	GAL6	GAL5	GAL4	GAL3	GAL2	GAL1	GAL0

In operation modes that provide address recognition, the high/low byte of the received address is compared with the programmable values in register RAR. This address can be masked on a per bit basis by setting the corresponding bits in the RAMR to allow extended broadcast address recognition.

Station Addrress

The address field or fields immediately follow the opening flag and identify the station for which the command is intended. If the address of the incoming frame does not match the address of the receiver, then the frame is rejected and the hunt for flag sequence is re-initiated.

When 8-bit addressing is enabled Station Addr-L will be used for the comparison and Station Addr-H will be ignored. When using 16-bit addressing both the Addr-H and the Addr-L will be used in the comparison.

Group Address

The group address register is a second 16-bit maskable address comparison register that can be used to define custom group addresses. This register is compared in addition to the primary address register. If the address comparison matches either of the registers, the frame will be received.

When 8-bit addressing is enabled Group Addr-L will be used for the comparison and Group Addr-H will be ignored. When using 16-bit addressing both the Addr-H and the Addr-L will be used in the comparison.

6.2.16 RAMR – Receive Address Mask Register (0x40/0xC0)

Station Address Mask

Station Addr-H Mask								Station Addr-L Mask							
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SAHM7	SAHM6	SAHM5	SAHM4	SAHM3	SAHM2	SAHM1	SAHM0	SALM7	SALM6	SALM5	SALM4	SALM3	SALM2	SALM1	SALM0

Group Address Mask

Group Addr-H Mask								Group Addr-L Mask							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GAHM7	GAHM6	GAHM5	GAHM4	GAHM3	GAHM2	GAHM1	GAHM0	GALM7	GALM6	GALM5	GALM4	GALM3	GALM2	GALM1	GALM0

This register holds the values used to mask the address bytes. Setting one of these bits to '1' masks the corresponding bit in the RAR register. A masked bit position always matches when comparing the received address to the corresponding RAR bit.

- Bit = 0 Default. The bit position is NOT MASKED. This bit position in the RAR must match the incoming address to receive a frame.
- Bit = 1 The bit position IS MASKED. This bit position in the RAR need not match the incoming address to receive a frame

6.2.17 PPR – Preamble & Postamble Register (0x44/0xC4)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NPRE7	NPRE6	NPRE5	NPRE4	NPRE3	NPRE2	NPRE1	NPRE0	PRE7	PRE6	PRE5	PRE4	PRE3	PRE2	PRE1	PRE0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NPOST7	NPOST6	NPOST5	NPOST4	NPOST3	NPOST2	NPOST1	NPOST0	POST7	POST6	POST5	POST4	POST3	POST2	POST1	POST0

NPRE[7:0] – Number of Preambles

- NPRE[7:0] = 0 Default. Preambles disabled
- NPRE[7:0] = 1 – 255 Preambles enabled. Transmit this number of preambles.

PRE[7:0] - The bit pattern that is sent out during preamble transmission.

- PRE[7:0] = 0x00 – 0xFF

NPOST[7:0] – Number of Postables

- NPOST[7:0] = 0 Default. Postables disabled
- NPOST[7:0] = 1 – 255 Postambles enabled. Transmit this number of postambles.

POST[7:0] - The bit pattern that is sent out during postamble transmission.

- POST[7:0] = 0x00 – 0xFF

Note: Zero and one bit insertion is disabled during postamble transmission.

6.2.18 TCR – Timer Control Register (0x48/0xC8)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TCNT28	TCNT27	TCNT26	TCNT25	TCNT24	TCNT23	TCNT22	TCNT21	TCNT20	TCNT19	TCNT18	TCNT17	TCNT16	TCNT15	TCNT14	TCNT13

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TCNT12	TCNT11	TCNT10	TCNT9	TCNT8	TCNT7	TCNT6	TCNT5	TCNT4	TCNT3	TCNT2	TCNT1	TCNT0	TTRIG	TSRC1	TSRC0

TSRC[1:0] – Timer Clock Source Select

- TSRC[1:0] = 00 Default. OSC input (the clock that drives BGR).
- TSRC[1:0] = 01 Effective transmit clock.
- TSRC[1:0] = 10 Effective receive clock.
- TSRC[1:0] = 11 PCI bus clock (external bus CLK input).

TTRIG – Timer Trigger Select

- TTRIG = 0 Default. Timer recycles and triggers an interrupt each time it expires down to zero.
- TTRIG = 1 Timer will count down to zero and trigger an interrupt one time.

TCNT[28:0] – Timer Expiration Count

This bit field determines the timer expiration count. The timer period (t) can be determined using this equation:

$$t = (TCNT + 1) * CP$$

(CP is the clock period depending on TSRC)

6.2.19 VSTR – Version Status Register (0x4C/0xCC)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PDEV15	PDEV14	PDEV13	PDEV12	PDEV11	PDEV10	PDEV9	PDEV8	PDEV7	PDEV6	PDEV5	PDEV4	PDEV3	PDEV2	PDEV1	PDEV0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PREV7	PREV6	PREV5	PREV4	PREV3	PREV2	PREV1	PREV0	FREV7	FREV6	FREV5	FREV4	FREV3	FREV2	FREV1	FREV0

FREV[7:0] – F-Core Revision number

This holds the revision number of the firmware loaded in the F-Core chip.

PREV[7:0] – PCI Revision ID

This holds the PCI Revision ID of this card.

PDEV[15:0] – PCI Device ID

This holds the PCI Device ID of this card.

Note: PCI Vendor ID = 0x18F7

6.2.20 ISR – Interrupt Status Register (0x50/0xD0)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	CTSA	CDC	DSRC	CTSS	0	0	0	0	0	TDU	ALLS	TFT

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DT_STOP	DR_STOP	DT_FE	DR_FE	DT_HI	DR_HI	0	TIN	0	RCD	RFL	RDO	RFO	RFE	RFT	RFS

RFS – Receive Frame Start Interrupt

This bit is set to a '1' if the receiver detects the beginning of a valid frame. A valid frame is detected either when a valid address field is recognized (in modes with address recognition) or when a start flag is recognized (in modes with no address recognition) or an pulse is detected on FSR (in modes that use frame sync signals).

RFT – Receive FIFO Trigger Interrupt

This bit is set to a '1' when the number of bytes in the RxFIFO exceeds the programmed RxFIFOTrigger level.

RFE – Receive Frame End Interrupt

This bit is set to a '1' whenever the RxByteCount for the current frame is written, or possibly when the RxStatus word is appended to the data. Both of these conditions would be acceptable as a trigger for RFE.

RFO – Receive Frame Overflow Interrupt

This bit is set to a '1' whenever the maximum number (512) of RxByteCount entries is exceeded.

RDO – Receive Data Overflow Interrupt

This bit is set to a '1' whenever data is received that will cause the depth (8K) of the RxFIFO to be exceeded.

RFL – Receive Frame Lost Interrupt

This bit is set to a '1' whenever the receiver detects an additional frame while an RDO interrupt is already waiting to be serviced. The RFL bit in the RxStatus word will be set for the next message that is pushed into the RxFIFO.

RCD – Receive Clock Detected Interrupt

This bit is set to a ‘1’ after the DRC command has been issued and three clocks have been detected on the receive clock pin.

TIN – Timer Expiration Interrupt

This bit is set to a ‘1’ each time the hardware timer expires.

DT_STOP – DMA Transmitter Full Stop indication

The transmit DMA engine has reached a full stop condition (processed a descriptor with the stop bit set) and no further DMA will occur in the transmit direction until a GO_T command is issued.

DR_STOP – DMA Receiver Full Stop indication

The receive DMA engine has reached a full stop condition (processed a descriptor with the stop bit set) and no further DMA will occur in the receive direction until a GO_R command is issued.

DT_FE – DMA Transmit Frame End indication

The DMA controller has processed a descriptor with the FE bit set.

DR_FE – DMA Receive Frame End indication

The DMA controller has processed a descriptor with the FE bit set, a complete (HDLC) frame is now stored in memory.

DT_HI – DMA Transmit Host Interrupt indication

The DMA controller has processed a transmit descriptor with the HI bit set.

DR_HI – DMA Receive Host Interrupt indication

The DMA controller has processed a receive descriptor with the HI bit set.

TFT – Transmit FIFO Trigger Interrupt

The TFT interrupt is used to indicate that the TxFIFO is ready to accept more data. This bit is set to a ‘1’ when either the number of bytes in the TxFIFO falls below the programmed TxFIFOTrigger level, or when moving from idle to active, if the BytesInTxFIFO count is less than the TxFIFOTrigger level anytime the XF command is issued (i.e. if you write less than the trigger size).

ALLS – All Sent Interrupt

This bit is set to a ‘1’ any time the transmitter moves from an active to an idle state.

TDU – Transmit Data Underrun Interrupt

This bit is set to a ‘1’ whenever transmitter moves from an active to an idle state, and the number of bytes pushed into the TxFIFO has not yet reached the current TxByteCount. This is to be an indication that the software is not moving data fast enough.

CTSS – CTS State Change Interrupt

This bit is set to a ‘1’ any time a change of state is detected on the CTS input pin.

DSRC – DSR Change Interrupt

This bit is set to a ‘1’ any time a change of state is detected on the DSR input pin.

CDC – CD Change Interrupt

This bit is set to a ‘1’ any time a change of state is detected on the CD input pin.

CTSA – CTS Abort Interrupt

This bit is set to a ‘1’ any time a state change on CTS triggers an abort sequence. This will only be valid when using CTS as a flow control mechanism.

6.2.21 IMR – Interrupt Mask Register (0x54/0xD4)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	CTSA	CDC	DSRC	CTSS	0	0	0	0	0	TDU	ALLS	TFT

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DT_STOP	DR_STOP	DT_FE	DR_FE	DT_HI	DR_HI	0	TIN	0	0	RFL	RDO	RFO	RFE	RFT	RFS

Each SCC interrupt event can generate an interrupt event in register ISR as well as an interrupt signal indication to the INT pin. Each bit position of register IMR is a mask for the corresponding interrupt event in the interrupt status register ISR. Masked interrupts neither generate an interrupt event nor an interrupt indication via the INT pin.

- Bit = 1 Default. The corresponding interrupt event IS MASKED and will neither generate an interrupt event nor an interrupt indication via the INT pin.
- Bit = 0 The corresponding interrupt event is NOT MASKED and will generate an interrupt event as well as an interrupt indication via the INT pin.

6.2.22 DPLLR – DPLL Reset control register (0x58/0xD8)

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	DPLLR8	DPLLR7	DPLLR6	DPLLR5	DPLLR4	DPLLR3	DPLLR2	DPLLR1	DPLLR0

The number of clock edges before the DPLL is considered out of sync (4.1.4.5). It may have a value from 1 to 511 and represents the number of clocks cycles without an edge that the DPLL will consider OK, more clocks than this and it will snap to the next edge. The number set is in 8 clock increments, so a value of 1 is 8 clocks, 2 16 clocks, etc.

DPLLR[8:0] – Timer Clock Source Select

- DPLLR[8:0] = 4 Default. 32 clock cycles

6.3 16C950 Based Registers

The following pages are extracted from the Oxford Semiconductor data sheet for the OX16C950. For more details or to view the complete document, you may download it from this link.

6 REGISTER DESCRIPTION TABLES

The three address lines select the various registers in the UART. Since there are more than 8 registers, selection of the registers is also dependent on the state of the Line Control Register 'LCR' and Additional Control Register 'ACR':

1. LCR[7]=1 enables the divider latch registers DLL and DLM.
2. LCR specifies the data format used for both transmitter and receiver. Writing 0xBF (an unused format) to LCR enables access to the 650 compatible register set. Writing this value will set LCR[7] but leaves LCR[6:0] unchanged. Therefore, the data format of the transmitter and receiver data is not affected. Write the desired LCR value to exit from this selection.
3. ACR[7]=1 enables access to the 950 specific registers.
4. ACR[6]=1 enables access to the Indexed Control Register set (ICR) registers as described on page 18.

Register Name	Address	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
THR [1]	000	W	Data to be transmitted							
RHR [1]	000	R	Data received							
IER [1,2] 650/950 Mode	001	R/W	CTS interrupt mask	RTS interrupt mask	Special Char. Detect	Sleep mode	Modem interrupt mask	Rx Stat interrupt mask	THRE interrupt mask	RxRDY interrupt mask
550/750 Mode			Unused		Alternate sleep mode					
FCR [3] 650 mode	010	W	RHR Trigger Level		THR Trigger Level		DMA Mode / Tx Trigger Enable	Flush THR	Flush RHR	Enable FIFO
750 mode			RHR Trigger Level		FIFO Size	Unused				
950 mode			Unused							
ISR [3]	010	R	FIFOs enabled		Interrupt priority (Enhanced mode)		Interrupt priority (All modes)			Interrupt pending
LCR [4]	011	R/W	Divisor latch access	Tx break	Force parity	Odd / even parity	Parity enable	Number of stop bits	Data length	
MCR [3,4] 550/750 Mode	100	R/W	Unused		CTS & RTS Flow Control	Internal Loop Back Enable	OUT2 (Int En)	OUT1	RTS	DTR
650/950 Mode			Baud prescale	IrDA mode	XON-Any					
LSR [3,5] Normal	101	R	Data Error	Tx Empty	THR Empty	Rx Break	Framing Error	Parity Error	Overrun Error	RxRDY
9-bit data mode								9th Rx data bit		
MSR [3]	110	R	DCD	RI	DSR	CTS	Delta DCD	Trailing RI edge	Delta DSR	Delta CTS
SPR [3] Normal	111	R/W	Temporary data storage register and Indexed control register offset value bits							
9-bit data mode			Unused							9th Tx data bit
Additional Standard Registers – These registers require divisor latch access bit (LCR[7]) to be set to 1.										
DLL	000	R/W	Divisor latch bits [7:0] (Least significant byte)							
DLM	001	R/W	Divisor latch bits [15:8] (Most significant byte)							

Table 4: Standard 550 Compatible Registers

Register Name	Address	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
To access these registers LCR must be set to 0xBF										
EFR	010	R/W	CTS flow control	RTS Flow control	Special char detect	Enhanced mode	In-band flow control mode			
XON1	100	R/W	XON Character 1							
9-bit mode			Special character 1							
XON2	101	R/W	XON Character 2							
9-bit mode			Special Character 2							
XOFF1	110	R/W	XOFF Character 1							
9-bit mode			Special character 3							
XOFF2	111	R/W	XOFF Character 2							
9-bit mode			Special character 4							

Table 5: 650 Compatible Registers

Register Name	Address	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ASR [1,6,7]	001	R/W [7]	Tx Idle	FIFO size	FIFO-SEL	Special Char Detect	DTR	RTS	Remote Tx Disabled	Tx Disabled
RFL [6]	011	R	Number of characters in the receiver FIFO							
TFL [3,8]	100	R	Number of characters in the transmitter FIFO							
ICR [3,8,9]	101	R/W	Data read/written depends on the value written to the SPR prior to the access of this register (see Table 7)							

Table 6: 950 Specific Registers

Register access notes:

Note 1: Requires LCR[7] = 0
Note 2: Requires ACR[7] = 0
Note 3: Requires that last value written to LCR was not 0xBF
Note 4: To read this register ACR[7] must be = 0
Note 5: To read this register ACR[6] must be = 0
Note 6: Requires ACR[7] = 1
Note 7: Only bits 0 and 1 of this register can be written
Note 8: To read this register ACR[6] must be = 1
Note 9: This register acts as a window through which to read and write registers in the Indexed Control Register set

Register Name	SPR Offset [10]	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Indexed Control Register Set										
ACR	0x00	R/W	Addit-ional Status Enable	ICR Read Enable	950 Trigger Level Enable	DTR definition and control		Auto DSR Flow Control Enable	Tx Disable	Rx Disable
CPR	0x01	R/W	5 Bit "integer" part of clock prescaler					3 Bit "fractional" part of clock prescaler		
TCR	0x02	R/W	Unused				4 Bit N-times clock selection bits [3:0]			
CKS	0x03	R/W	Tx 1x Mode	Tx CLK Select	BDOUT on DTR	DTR 1x Tx CLK	Rx 1x Mode	Disable BDOUT	Receiver Clock Sel[1:0]	
TTL	0x04	R/W	Unused	Transmitter Interrupt Trigger Level (0-127)						
RTL	0x05	R/W	Unused	Receiver Interrupt Trigger Level (1-127)						
FCL	0x06	R/W	Unused	Automatic Flow Control Lower Trigger Level (0-127)						
FCH	0x07	R/W	Unused	Automatic Flow Control Higher Trigger level (1-127)						
ID1	0x08	R	Hardwired ID byte 1 (0x16)							
ID2	0x09	R	Hardwired ID byte 1 (0xC9)							
ID3	0x0A	R	Hardwired ID byte 1 (0x50)							
REV	0x0B	R	Hardwired revision byte (0x03)							
CSR	0x0C	W	Writing 0x00 to this register will reset the UART (Except the CKS and CKA registers)							
NMR	0x0D	R/W	Unused		9th Bit SChar 4	9th Bit Schar 3	9th Bit SChar 2	9th Bit SChar 1	9th-bit Int. En.	9 Bit Enable
MDM	0x0E	R/W	Unused				Δ DCD Wakeup disable	Trailing RI edge disable	Δ DSR Wakeup disable	Δ CTS Wakeup disable
RFC	0X0F	R	FCR[7]	FCR[6]	FCR[5]	FCR[4]	FCR[3]	FCR[2]	FCR[1]	FCR[0]
GDS	0X10	R	Unused							Good Data Status
DMS	0x11	R/W	Force TxRdy inactive	Force RxRdy inactive	Unused				TxRdy status (R)	RxRdy status (R)
PIDX	0x12	R	Hardwired Port Index (0x00)							
CKA	0x13	R/W	Unused			Output sys-clk on txrdy	Use CLKSEL pin for sys-clk	Invert DTR signal	Invert internal tx clock	Invert internal rx clock

Table 7: Indexed Control Register Set

Note 10: The SPR offset column indicates the value that must be written into SPR prior to reading / writing any of the indexed control registers via ICR. Offset values not listed in the table are reserved for future use and must not be used.

To read or write to any of the Indexed Control Registers use the following procedure.

Writing to ICR registers:
Ensure that the last value written to LCR was not 0xBF (reserved for 650 compatible register access value).
Write the desired offset to SPR (address 111_2).

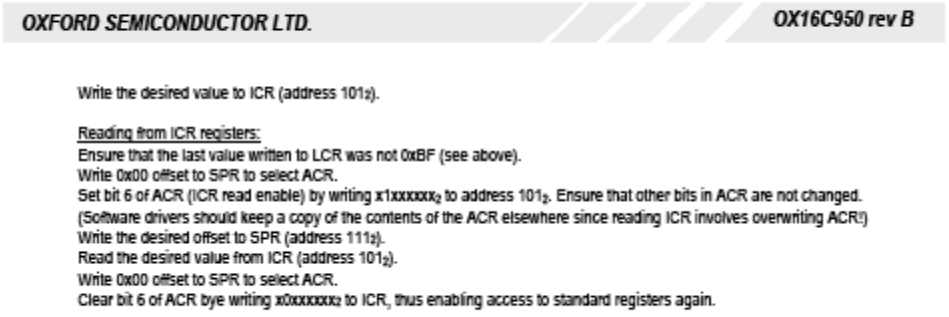


Figure 8 - UART register description

6.4 Board Control

Table 4 Board Control / DMA Registers

Address		Access Type	Register		Page
Port A	**Port B**		**Name**	**Description**	
0x00	0x00	R/W	FCR	Feature Control Register	102
0x04	0x08	R/W	DMACCR	DMA Command / Control Register	104
0x0C	0x14	R/W	DRDB	DMA Receive Descriptor Base	106
0x10	0x18	R/W	DTDB	DMA Transmit Descriptor Base	106
0x20	0x28	R	DCRDB	DMA Current Receive Descriptor Base	106
0x24	0x2C	R	DCTDB	DMA Current Transmit Descriptor Base	106
0x30	0x30	R/W	DSTAR	DMA Status Register	107

6.4.1 FCR – Feature Control Register (0x00)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	UART_B	UART_A	FSTDTRB	TD485_B	TC485_B	RECHOB	FSTDTRA	TD485_A	TC485_A	RECHOA

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	STRB_B	0	CLK_B	DTA_B	0	0	0	0	STRB_A	0	CLK_A	DTA_A

DTA_A –ICS307-03 data signal for Port A (clock generator programming signal).

CLK_A –ICS307-03 clock signal for Port A (clock generator programming signal).

STRB_A –ICS307-03 strobe signal for Port A (clock generator programming signal).

DTA_B –ICS307-03 data signal for Port B (clock generator programming signal).

CLK_B –ICS307-03 clock signal for Port B (clock generator programming signal).

STRB_B – ICS307-03 strobe signal for Port B (clock generator programming signal).

RECHOA – Receive Echo Cancel control for port A.

- RECHOA = 0 Default. Receiver is always activated.
- RECHOA = 1 Disable the receiver during transmits

TC485_A – Transmit Clock Output RS-485 enable/disable for port A.

- TC485_A = 0 Default. TxClkOut is RS-422 (transmitter always active)
- TC485_A = 1 TxClkOut is RS-485 (transmitter tri-state when idle)

TD485_A – Transmit Data RS-485 enable/disable for port A

- TD485_A = 0 Default. TxD is RS-422 (transmitter always active)
- TD485_A = 1 TxD is RS-485 (transmitter tri-state when idle)

FSTDTRA – FST/DTR pin selection for port A

- FSTDTRA = 0 FST/DTR pin is FST
- FSTDTRA = 1 FST/DTR pin is DTR

RECHOB – Receive Echo Cancel control for port B.

- RECHOB = 0 Default. Receiver is always activated.

- RECHOB = 1 Disable the receiver during transmits

TC485_B – Transmit Clock Output RS-485 enable/disable for port B.

- TC485_B = 0 Default. TxClkOut is RS-422 (transmitter always active)
- TC485_B = 1 TxClkOut is RS-485 (transmitter tri-state when idle)

TD485_B – Transmit Data RS-485 enable/disable for port B

- TD485_B = 0 Default. TxD is RS-422 (transmitter always active)
- TD485_B = 1 TxD is RS-485 (transmitter tri-state when idle)

UARTA – ASYNC driver control for port A

- UARTA = 0 Default. Enables F-Core connection (SYNC mode)
- UARTA = 1 Enables 16950 UART connection to physical 422/485 drivers/receivers (True-Async mode)

FSTDTRB – FST/DTR pin selection for port B

- FSTDTRB = 0 FST/DTR pin is FST
- FSTDTRB = 1 FST/DTR pin is DTR

UARTB – ASYNC driver control for port B

- UARTB = 0 Default. Enables F-Core connection (SYNC mode)
- UARTB = 1 Enables 16950 UART connection to physical 422/485 drivers/receivers (True-Async mode)

6.4.2 DMACCR - DMA Command / Control Register (0x04/0x08)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	XREP_E	TXEXT_E	TXT_E	0	0	0	0	0	0	0	0	0	0	0	M_RST

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	STOP_T	STOP_R	0	0	RST_T	RST_R	0	0	GO_T	GO_R

This register is only available on SuperFSCC products.

For all of these commands a ‘1’ will execute the command, and a ‘0’ will do nothing. These commands are self-clearing. (i.e. a ‘1’ written to any command will read back as a ‘0’ after it has executed).

GO_R – Start Receive DMA

Initiate processing of receive DMA, reads in base receive descriptor and processes received data.

GO_T – Start Transmit DMA

Initiate processing of transmit DMA, reads in base transmit descriptor and processes transmit data.

RST_R – Reset Receive DMA

Flushes receive DMA FIFOs and returns receive DMA section to known state. Should be used with a RRES command to the FSCC to flush the serial controller FIFO’s, and both should be done with the receiver disabled (CCR0:RECD=’1’).

RST_T – Reset Transmit DMA

Flushes transmit DMA FIFOs and returns transmit DMA section to known state. Should be used with a TRES command to the FSCC to flush the serial controller FIFOs, and both should be done after a ALLS interrupt indication (or TDU). Issuing either reset before the ALLS/TDU could result in a partial/lost transmitted frame(s).

STOP_R – Stop Receive DMA

User forced stop of receive DMA section, will generate DR_STOP interrupt indication and end receive DMA operation.

STOP_T – Stop Transmit DMA

User forced stop of transmit DMA section, will generate DT_STOP interrupt indication and end transmit DMA operation.

M_RST – Master DMA reset.

Use on detection of Master Dead status. This bit only exists for port A

TXT_E – Transmit by Timer Enable

- TXT_E = 0 Normal transmit mode (Frame start command = XF)
- TXT_E = 1 Transmit by Timer mode (Frame start command = XF|TXT)

TXEXT_E – Transmit by External Signal Enable

- TXEXT_E = 0 Normal transmit mode (Frame start command = XF)
- TXEXT_E = 1 Transmit by External mode (Frame start command = XF|TEXT)

XREP_E – Transmit Repeat Enable

- XREP_E = 0 Normal transmit mode (Frame start command = XF)
- XREP_E = 1 Transmit Repeat mode (Frame start command = XF|XREP)

Note: you must manually issue the stop XREP command

6.4.3 DMA Receive Descriptor Base (0x0c/0x14)

32bit physical (PCI bus relative) memory location of the first receive descriptor.

6.4.4 DMA Transmit Descriptor Base (0x10/0x18)

32bit physical (PCI bus relative) memory location of the first transmit descriptor.

6.4.5 DMA Current Receive Descriptor Base (0x20/0x28)

32bit physical (PCI bus relative) memory location of the current receive descriptor.

6.4.6 DMA Current Transmit Descriptor Base (0x24/0x2c)

32bit physical (PCI bus relative) memory location of the current transmit descriptor.

6.4.7 DSTAR – DMA Status Register (0x30)

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	MASTER DEAD	TSTOP_B	RSTOP_B	TSTOP_A	RSTOP_A

This register is only available on SuperFSCC products.

RSTOP_A – Port A Receive DMA activity

- RSTOP_A = 0 The receive DMA section is running/processing descriptors.
- RSTOP_A = 1 The receive DMA section is stopped.

TSTOP_A – Port B Transmit DMA activity

- TSTOP_A = 0 The transmit DMA section is running/processing descriptors.
- TSTOP_A = 1 The transmit DMA section is stopped.

RSTOP_B – Port B Receive DMA activity

- RSTOP_B = 0 The receive DMA section is running/processing descriptors.
- RSTOP_B = 1 The receive DMA section is stopped.

TSTOP_B – Port B Transmit DMA activity

- TSTOP_B = 0 The transmit DMA section is running/processing descriptors.
- TSTOP_B = 1 The transmit DMA section is stopped.

MASTER DEAD – The Master DMA engine is in a full stopped dead state and is no longer transferring data across the PCI bus (something very bad has occurred on the PCI bus). The Master must be reset via the M_RST bit in the control register, but it is unknown at this time if the cause of the trouble is resolved/resolvable by this reset.

6.5 DMA Descriptor Layout

The layout of a DMA descriptor (receive or transmit) at a given address in memory is as follows:

Offset	Name	Description	Page
0x00	Control	Descriptor Control Register	109
0x04	DataAddress	Memory address of data	110
0x08	DataCount	Number of valid bytes at Data Address	110
0x0c	NextDescriptor	Next descriptor in chain	110

6.5.1 Descriptor Control Register (0x00)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FE	C/STOP	HI	CNT28	CNT27	CNT26	CNT25	CNT24	CNT23	CNT22	CNT21	CNT20	CNT19	CNT18	CNT17	CNT16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT15	CNT14	CNT13	CNT12	CNT11	CNT10	CNT9	CNT8	CNT7	CNT6	CNT5	CNT4	CNT3	CNT2	CNT1	CNT0

FE – Frame End

- Transmit: when this bit is set CNT indicates the size of the complete frame to transmit. When spanning descriptors for a single frame, the first descriptor in the frame should have FE/CNT set.
- Receive: when this bit is set CNT indicates the size of the completed receive frame. If this bit is not set but the C/STOP bit is, then the descriptor is complete with DataCount bytes. The last descriptor of a multi-descriptor frame will have FE set with the CNT value for the entire frame, the number of bytes in the last descriptor can be calculated by subtracting the number previously received bytes from CNT.

C/STOP – Complete / STOP

Complete

When DMA finishes processing of this descriptor, the DMA controller will set this bit.

STOP

- Transmit: This bit is set in the Last descriptor in a chain to stop DMA processing
- Receive: If this bit is set the Receive DMA processing will be stopped.

HI – Host interrupt

When this bit is set the *Super*F-Core will generate an interrupt when this descriptor is processed (DR_HI / DT_HI).

CNT[28:0] – Frame Count

Number of bytes to transmit, or number of bytes received in a completed frame.

6.5.2 DataAddress (0x04)

PCI bus relative physical address of the memory buffer where data is to be transferred from/to. Must be DWORD aligned.

6.5.3 DataCount (0x08)

This is the number of bytes that are valid at DataAddress.

If spanning descriptors (one frame) only the last descriptor in the frame can have a non-DWORD size. In the receive direction all descriptors should be a multiple of 4.

6.5.4 NextDescriptor (0x0c)

The PCI bus relative physical address of the next descriptor in the descriptor chain. When the DMA controller has finished processing this descriptor it will load and process the next descriptor from this address.

7 TECHNICAL SUPPORT

Commtech provides extensive technical support and application suggestions. Most of the problems that occur with the FASTCOM: *FSCC/SuperFSCC* can be corrected by double-checking your cables and your program settings. We recommend that you use the loop back plug that is included, with that plug, you can quickly isolate the problem to the board, cables, or software.

If you still have unresolved questions, use the following procedure to get technical support:

1. Call our Technical Support Staff at (316) 636-1131. They are on duty from 9:00 AM to 5:00 PM Central Time.
2. Ask for technical support for the FASTCOM: *FSCC/SuperFSCC*. Be ready to describe the problem, your computer system, your application, and your software.
3. If necessary, our staff will give you an RMA number (Return Material Authorization). Use this number on the mailing label and in all references to your board. Put the board back in its static bag and in its box. Ship the board back to us as directed.
4. If you prefer, you may email a description of the problem to us at: techsupport@fastcomproducts.com.
   You may also use our tech support form at: http://www.fastcomproducts.com/TechSupport.html.

8 FASTCOM LIMITED LIFETIME WARRANTY

Commtech's entire FASTCOM product line is covered by a limited lifetime warranty against defects in workmanship. This warranty is available only to the original purchaser and only covers defects in our workmanship. Any FASTCOM board that is returned to Commtech will, at the option of Commtech, be repaired or replaced at no charge -- except for circumstances excluded by this warranty.

A Return Materials Authorization Number (RMA#) must be obtained from Commtech before a return will be accepted. Please contact us via telephone or email to obtain an RMA #.

You are responsible for shipping charges when you return a FASTCOM board to Commtech. We will pay the shipping charges to send the board back to you, if a defect in workmanship is found. However, if no defect in workmanship is found, or the FASTCOM is not found to be defective, or the any of the following warranty exclusions occur, you will be responsible for shipping charges both ways.

8.1 Warranty Exclusions

This warranty does not cover problems or damage resulting from, but not limited to the following:

1. Any modification, misuse, abuse, disassembly, misapplication, or unauthorized repair by anyone other than Commtech.
2. Any improper operation, including any use not in accordance with any verbal product instructions or documentation.
3. Connection to an improper voltage supply or ESD damage.
4. Any other cause not related to workmanship.

8.2 Non-Warranty Repairs

We can provide a quote for non-warranty repairs upon request.

8.3 Limitation of Liability

Commtech shall not be liable for any special, incidental, indirect, or consequential damages whatsoever, including but not limited to loss of profits, revenue, or data (whether direct or indirect), or commercial loss for breach of any express or implied warranty on your product even if Commtech has been advised previously of the possibility of such damages. Commtech does not warrant that its products will work in every system or every system configuration. We do not warrant that our products will be suitable for your application. If you are dissatisfied with our product, contact customer service to arrange for a return of our product and refund of your money. Commtech's liability, in any case, is limited to the original product purchase price and is available to the original customer only.

Artisan Technology Group is an independent supplier of quality pre-owned equipment

Gold-standard solutions

Extend the life of your critical industrial, commercial, and military systems with our superior service and support.

We buy equipment

Planning to upgrade your current equipment? Have surplus equipment taking up shelf space? We'll give it a new home.

Learn more!

Visit us at **artisantg.com** for more info on price quotes, drivers, technical specifications, manuals, and documentation.

Artisan Scientific Corporation dba Artisan Technology Group is not an affiliate, representative, or authorized distributor for any manufacturer listed herein.

We're here to make your life easier. How can we help you today?

(217) 352-9330 | sales@artisantg.com | artisantg.com

ARTISAN® TECHNOLOGY GROUP