Micro-Research Finland Oy
Välitalontie 83 C, FI-00660 Helsinki, Finland

8-Way CompactPCI Fan-Out Concentrator

cPCI-FCT-8, Firmware 30000001

Technical Reference

Contents

Introduction

The main function of the fan-out modules is to distribute the event stream generated by an event generator to a network of event receivers in star configuration.

Main features include:

- Small form factor pluggable (SFP) transceivers with LC connectors, by default short wavelength (850 nm) transceivers for multimode fibre are provided.
- 6U 4 HP module which only takes power from the host bus

Fan-Out

The 8-way fan-out receives the optical event signal through a fibre connected to the uplink RX port. In the SFP transceiver this signal is converted to a differential electrical signal which gets cleaned by a CDR (clock and data recovery circuit). This signal is fanned out to all eight downstream SFP ports 1-8. Here the differential electrical signal is converted to an optical signal and sent out through a fibre.

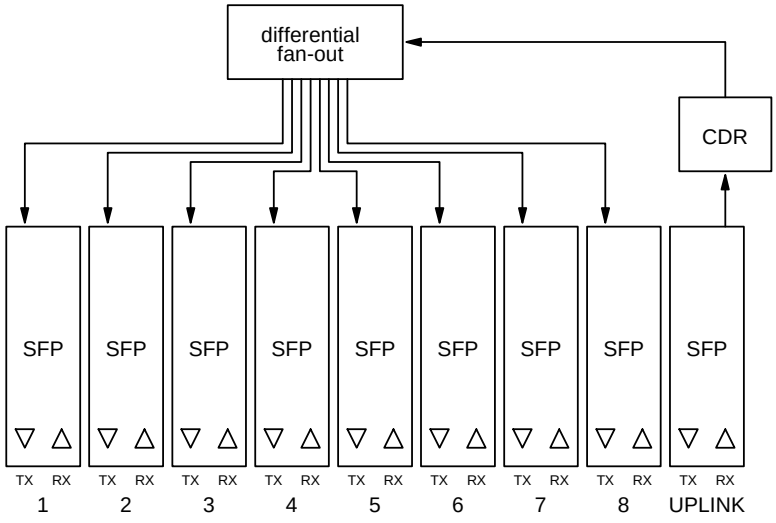

Figure 1: Fan-out block diagram

Concentrator

The concentrator receives signals from up to eight EVRs/downstream concentrators and forwards the signals upstream. The link channel is capable of carrying different type of information: events, distributed bus bit and data buffers to up to 2 kbytes.

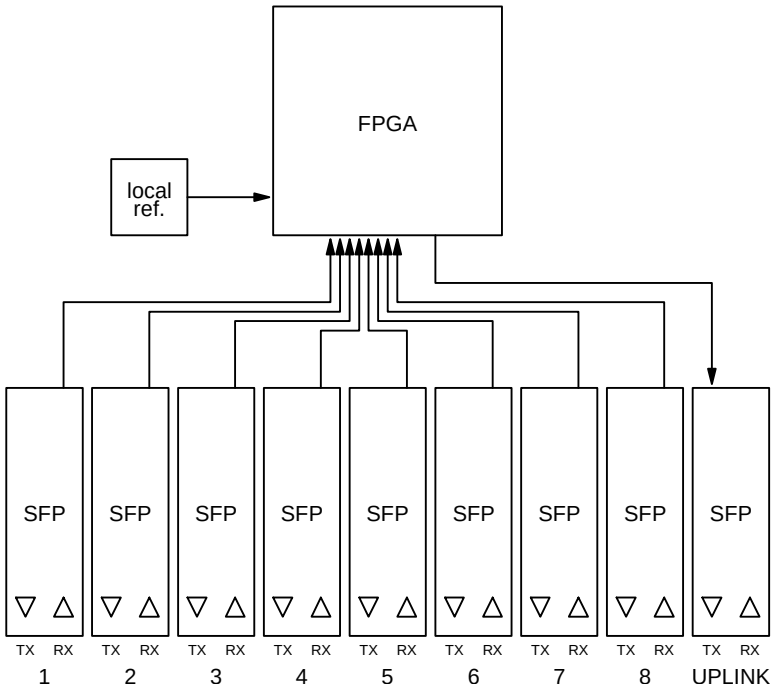

Figure 2: Concentrator block diagram

Event Forwarding

Events are forwarded on first come first served principle. In case there are lots of simultaneous events a round-robin priority scheme takes care that one port does not block the upstream TX port. Each receiving port has a FIFO memory that can hold up to 2 k events so that not events are lost in case of a sudden burst of events. Status bits indicate overflow conditions of the FIFOs.

Distributed Bus Forwarding

The distributed bus consists of eight bits that are updated simultaneously. Each forwarded bit is formed by combining the corresponding bits from all RX ports.

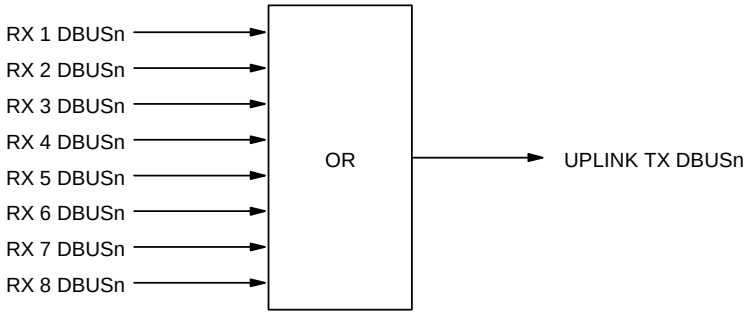

Figure 3: Distributed Bus Bit Forwarding for each bit n=0…7

Data Buffer Forwarding

Data Buffers are data packets that can carry a payload of 4 bytes to 2048 bytes in four byte increments. Each receiving channels has a 2 kbyte FIFO to buffer incoming data. The priority encoder is similar to the event encoder: if there are simultaneous transfers from several sources a

round-robin arbitration scheme is used. Latency is minimized in forwarding the packets immediately – the concentrator does not wait for the end of packet before it starts forwarding the packet.

Front Panel

The front panel of the board is shown in Figure 4. The front panel shows nine SFP transceivers, one for upstream and eight for downstream connections. Each transceiver has two LEDs on top of each transceivers RX and TX port. The uplink port is on the right hand side.

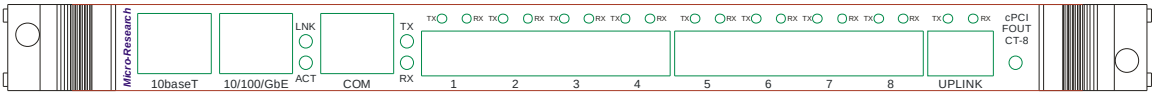

Figure 4: FOUT-7 Front Panel

The connections and LED on the front panel have following functionality:

Port/Led	Description
10baseT	Configuration Ethernet interface
10/100/GbE	10/100/GbE Port (reserved)
LNK	10/100/GbE Link led
ACT	10/100/GbE Active led
COM	Serial Port (reserved)
TX	Serial Port TX led
RX	Serial Port RX led
SFP Port 1-8	Downstream Connection 1-8 (to EVR/downstream fan-out concentrator)
SFP Port 1-8 TX led	Red – no signal on UPLINK RX port Green – signal detected on UPLINK RX port
SFP Port 1-8 RX led	Static red – no RX signal Static green – RX link OK, no errors Green/Red transitioning – RX link OK, RX violation not cleared Flashing blue – RX event
UPLINK SFP	Upstream Connection (to EVG/upstream fan-out concentrator)
SFP Uplink TX led	Red – onboard reference oscillator not locked Green – reference oscillator locked Flashing – TX event
SFP Uplink RX led	Static red – no input signal Static green – RX link OK, no errors Green/Red transitioning – RX link OK, RX violation not cleared

Power Sequencing/Power Consumption

The fan-out concentrator boards have been designed hot-swappable with crates supporting hot-swapping. Power sequencing is controlled as specified by the CompactPCI hot-plug specification.

Supply Voltage	Power Consumption (typical)
+3.3 VDC	13.3 W
+5 VDC	4.5 W

Configuration Network Interface

A 10baseT network interface is provided to upgrade the FPGA firmware and set up boot options. It is also possible to control the module over the network interface.

Assigning an IP Address to the Module

By default the modules uses DHCP (dynamic host configuration protocol) to acquire an IP address. In case a lease cannot be acquired the IP address set randomly in the 169.254.x.x subnet. The board can be programmed to use a static address instead if DHCP is not available.

The module can be located looking at the lease log of the DHCP server or using a Windows tool called Locator.exe.

Using Telnet to Configure Module

To connect to the configuration utility of the module issue the following command:

```
telnet 192.168.1.32 23
```

The latter parameter is the telnet port number and is required in Linux to prevent negotiation of telnet parameters which the telnet server of the module is not capable of.

The telnet server responds to the following commands:

Command	Description
B	Show/change boot parameters, IP address etc.
H / ?	Show Help
M <address> [<data>]	Read/Write FPGA PLB attached devices
R	Reset Board
S	Save boot configuration & dynamic configuration values into non-volatile memory
U	Update IP2022 software
Q	Quit Telnet

Boot Configuration (command b)

Command b displays the current boot configuration parameters of the module. The parameter may be changed by giving a new parameter value. The following parameters are displayed:

Parameter	Description
Use DHCP	0 = use static IP address, 1 = use DHCP to acquire address, net mask etc.
IP address	IP address of module
Subnet mask	Subnet mask of module
Default GW	Default gateway
FPGA mode	FPGA configuration mode 0 – FPGA is not configured after power up 1 – FPGA configured from internal Flash memory 2 – FPGA is configured from FTP server
FTP server	FTP server IP address where configuration bit file resides
Username	FTP server username

Password	FTP server password
FTP Filename	FTP server configuration file name
Flash Filename	Configuration file name on internal flash
µs divider	Integer divider to get from event clock to 1MHz, e.g. 125 for 124.9135 MHz
Fractional divider configuration word	Micrel SY87739UMI fractional divider configuration word to set refenrence for event clock

Note that after changing parameters the parameters have to be saved to internal flash by issuing the Save boot configuration (s) command. The changes are applied only after resetting the module using the reset command or hardware reset/power sequencing.

Upgrading IP2022 Microprocessor Software (command u)

To upgrade the Ubicom IP2022 microprocessor software download the upgrade image containing the upgrade to the module using TFTP:

Linux

In Linux use tftp:

```
$ tftp 192.168.1.32 –m binary –c put upgrade.bin /fw
```

Windows

In Windows command prompt issue the following command:

```
C:\> tftp –i 192.168.1.32 PUT upgrade.bin /fw
```

When the upgrade image has been downloaded and verified, enter at the telnet prompt following:

```
EVR-200 -> u ↵
Really update firmware (yes/no) ? yes ↵
Self programming triggered.
```

The Event Receiver starts programming the new software and restarts.

Upgrading FPGA Configuration File

When the FPGA configuration file resides in internal flash memory a new file system image has to be downloaded to the module. This is done using TFTP protocol:

Linux

In Linux use e.g. interactive tftp:

```
$ tftp 192.168.1.32 –m binary –c put filesystem.bin /
```

Windows

In Windows command prompt issue the following command:

```
C:\> tftp –i 192.168.1.32 PUT filesystem.bin /
```

Now the FPGA configuration file has been upgraded and the new configuration is loaded after next reset/power sequencing.

Note! Due to the UDP protocol it is recommended to verify (read back and compare) the filesystem image before restarting the module. This is done following:

Linux

In Linux use e.g. interactive tftp:

```
$ tftp 192.168.1.32 –m binary –c get / verify.bin
$ diff filesystem.bin verify.bin
$
```

If files differ you should get following message:

```
Binary files filesystem.bin and verify.bin differ
```

Windows

In Windows command prompt issue the following command:

```
C:\> tftp –i 192.168.1.32 GET / verify.bin
C:\> fc /b filesystem.bin verify.bin
Comparing files filesystem.bin and verify.bin
FC: no differences encountered
```

Remote Programming Protocol

The board is remotely programmable from the 10baseT Ethernet interface using a protocol over UDP (User Datagram Protocol) which runs on top of IP (Internet Protocol). The default port for the remote programming protocol is UDP port 2000. The UDP commands are built upon the following structure:

Access_type (1 byte)	status (1 byte)	data (2 bytes)
address (4 bytes)		
ref (4 bytes)		

The first cell defines the access type:

access_type	Description
0x01	Read Register from FPGA PLB
0x02	Write and Read back Register from FPGA PLB

The second cell tells the status of the access:

status	Description
0	Command OK
-1	Bus ERROR (Invalid read/write address)
-2	Timeout (FPGA did not respond)
-3	Invalid command

Read Access (Type 0x01)

The host sends a UDP packet to port 2000 with the following contents:

access_type (1 byte) 0x01	status (1 byte) 0x00	data (2 bytes) 0x0000
address (4 bytes) 0x80000000 (PLB read address)		
ref (4 bytes) 0x00000000		

If the read access is successful the fan-out concentrator replies to the same host and port the message came from with the following packet:

access_type (1 byte) 0x01	status (1 byte) 0x00	data (2 bytes) data at PLB address
address (4 bytes) 0x80000000 (PLB address)		
ref (4 bytes) 0x00000000		

Write Access (Type 0x02)

The host sends a UDP packet to port 2000 with the following contents:

access_type (1 byte) 0x02	status (1 byte) 0x00	data (2 bytes) 0x0005
address (4 bytes) 0x80000000 (PLB write address)		
ref (4 bytes) 0x00000000		

If the write access is successful the ASP-EVG replies to the same host and port the message came from with the following packet:

access_type (1 byte) 0x02	status (1 byte) 0x00	data (2 bytes) (readback data at address)
address (4 bytes) 0x80000000 (PLB write address)		
ref (4 bytes) 0x00000000		

Notice that in the reply message the data returned really is the data read from the address specified in the address field so one can verify that the data really was written ok.

Register Map

Address	Register	Type	Description
0x10000000	Status	UINT32	Link Status Register
0x10000004	Control	UINT32	Link Control Register

0x10000008	Enable	UINT32	Link Enable Register
0x1000000C	Queue Status	UINT32	Event Queue Status and Control Register
0x1000002C	FWVersion	UINT32	Firmware Version Register
0x10000080	FracDiv	UINT32	Fractional Synthesizer Control Register

Status Register

Address	bit 31	bit 30	bit 29	bit 28	bit 27	bit 26	bit 25	bit 24
0x10000000	RXUP8	RXUP7	RXUP6	RXUP5	RXUP4	RXUP3	RXUP2	RXUP1

address	bit 23	bit 22	bit 21	bit 20	bit 19	bit 18	bit 17	bit 16
0x10000001								RXUPUL

address	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
0x10000002	RXVIO 8	RXVIO 7	RXVIO 6	RXVIO 5	RXVIO 4	RXVIO 3	RXVIO 2	RXVIO 1

Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0x10000003								RXVIOUL

Bit	Function
RXUP8	0 – RX link OK on port 8 down
	1 – RX link OK on port 8 up
RXUP7	0 – RX link OK on port 7 down
	1 – RX link OK on port 7 up
RXUP6	0 – RX link OK on port 6 down
	1 – RX link OK on port 6 up
RXUP5	0 – RX link OK on port 5 down
	1 – RX link OK on port 5 up
RXUP4	0 – RX link OK on port 4 down
	1 – RX link OK on port 4 up
RXUP3	0 – RX link OK on port 3 down
	1 – RX link OK on port 3 up
RXUP2	0 – RX link OK on port 2 down
	1 – RX link OK on port 2 up
RXUP1	0 – RX link OK on port 1 down
	1 – RX link OK on port 1 up
RXUPUL	(reserved)
	0 – UPLINK RX link OK on port 1 down
	1 – UPLINK RX link OK on port 1 up
RXVIO8	0 – RX violation flag clear, no violation detected
	1 – RX violation flag set, violation detected
RXVIO7	0 – RX violation flag clear, no violation detected
	1 – RX violation flag set, violation detected
RXVIO6	0 – RX violation flag clear, no violation detected
	1 – RX violation flag set, violation detected
RXVIO5	0 – RX violation flag clear, no violation detected
	1 – RX violation flag set, violation detected
RXVIO4	0 – RX violation flag clear, no violation detected

	1 – RX violation flag set, violation detected
RXVIO3	0 – RX violation flag clear, no violation detected
	1 – RX violation flag set, violation detected
RXVIO2	0 – RX violation flag clear, no violation detected
	1 – RX violation flag set, violation detected
RXVIO1	0 – RX violation flag clear, no violation detected
	1 – RX violation flag set, violation detected
RXVIOUL	(reserved)
	0 – RX violation flag clear, no violation detected
	1 – RX violation flag set, violation detected

Control Register

address	bit 31	bit 30	bit 29	bit 28	bit 27	bit 26	bit 25	bit 24
0x10000004								

Address	bit 23	bit 22	bit 21	bit 20	bit 19	bit 18	bit 17	bit 16
0x10000005								DBUF

Address	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
0x10000006	CVIO8	CVIO7	CVIO6	CVIO5	CVIO4	CVIO3	CVIO2	CVIO1

Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0x10000007								CVIOUL

Bit	Function
DBUF	Data Buffer Mode
	0 – data transfers inhibited (distributed bus running at full speed)
	1 – data transfers allowed (distributed bus running at half rate)
CVIO8	0 – No change
	1 – Clear port 8 RX violation flag
CVIO7	0 – No change
	1 – Clear port 8 RX violation flag
CVIO6	0 – No change
	1 – Clear port 8 RX violation flag
CVIO5	0 – No change
	1 – Clear port 8 RX violation flag
CVIO4	0 – No change
	1 – Clear port 8 RX violation flag
CVIO3	0 – No change
	1 – Clear port 8 RX violation flag
CVIO2	0 – No change
	1 – Clear port 8 RX violation flag
CVIO1	0 – No change
	1 – Clear port 8 RX violation flag
CVIOUL	0 – No change
	1 – Clear UPLINK RX violation flag

Enable Register

address	bit 31	bit 30	bit 29	bit 28	bit 27	bit 26	bit 25	bit 24
0x10000008	RXEN8	RXEN7	RXEN6	RXEN5	RXEN4	RXEN3	RXEN2	RXEN1

address	bit 23	bit 22	bit 21	bit 20	bit 19	bit 18	bit 17	bit 16
0x10000009								

Address	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
0x1000000A	RXDB8	RXDB7	RXDB6	RXDB5	RXDB4	RXDB3	RXDB2	RXDB1

Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0x1000000B								

Bit	Function
RXEN8	0 – Port 8 RX disabled
	1 – Port 8 RX enabled
RXEN7	0 – Port 7 RX disabled
	1 – Port 7 RX enabled
RXEN6	0 – Port 6 RX disabled
	1 – Port 6 RX enabled
RXEN5	0 – Port 5 RX disabled
	1 – Port 5 RX enabled
RXEN4	0 – Port 4 RX disabled
	1 – Port 4 RX enabled
RXEN3	0 – Port 3 RX disabled
	1 – Port 3 RX enabled
RXEN2	0 – Port 2 RX disabled
	1 – Port 2 RX enabled
RXEN1	0 – Port 1 RX disabled
	1 – Port 1 RX enabled
RXDB8-1	0 – Port 1 RX databuf reception disabled
	1 – Port 1 RX databuf reception enabled

Event Queue Status Register

address	bit 31	bit 30	bit 29	bit 28	bit 27	bit 26	bit 25	bit 24
0x1000000C	RXQF8	RXQF7	RXQF6	RXQF5	RXQF4	RXQF3	RXQF2	RXQF1

Address	bit 23	bit 22	bit 21	bit 20	bit 19	bit 18	bit 17	bit 16
0x1000000D								

Bit	Function
RXQF8-1	RX Port 8 Event Queue full flag:
	When 1: Queue has been full and events could be lost
	Write 1 to clear event queue and flag

FPGA Firmware Version Register

address	bit 31	bit 27	bit 26	bit 24
0x02C	Fan-out concentrator = 0x3		Form Factor	

address	bit 23	bit 8
0x02D	Reserved	

address	bit 7	bit 0
0x02F	Version ID	

Bits	Function
Form Factor	0 – CompactPCI 1 – PMC 2 – VME64x

SY87739L Fractional Divider Configuration Word

address	bit 31	bit 0
0x080	SY87739L Fractional Divider Configuration Word	

Configuration Word	Frequency with 24 MHz reference oscillator
0x0C928166	124.907 MHz
0x0C9282A6	62.454 MHz
0x009743AD	50 MHz
0xC25B43AD	49.978 MHz
0x0176C36D	49.965 MHz