Document No.
CERN Div./Group or Supplier/Contractor Document No.
BE/CO/HT
EDMS Document No.

CERN BE DEPARTMENT
CH-1211 Geneva 23
Switzerland

Date: 2009-06-26

Technical Note

WORLDFIP INSOURCING PROJECT NanoFIP WP3

NanoFIP FUNCTIONAL SPECIFICATION

Abstract

The NanoFIP is an FPGA component implementing the WorldFIP protocol that can be used in field devices able to communicate at the three standard speeds: 31.25 kbit/s, 1 Mbit/s and 2.5 Mbit/s. It is designed to be radiation tolerant by using different single event upset mitigation techniques. To improve the radiation tolerance, the NanoFIP implements a minimal subset of the WorldFIP services that it provides in a stand-alone mode without the need for any microprocessor. WorldFIP variables are transferred either directly on 16-bit I/O ports or via a memory included in the NanoFIP component.

Author:	***Checked by:***	
E. van der Bij	**R. Brun** **F. Formenti** **Q. King** **J. Palluel** **G. Peñacoba** **J. Serrano** **B. Todd**	

History of Changes

Rev. No.	***Date***	***Pages***	***Description of Changes***
1.0	09-06-2009	All	First draft
1.1	22-06-2009	All	Title: functional specification. Name changed to NanoFIP. Added definitions of PDU and MPS. Variable E0xy needs station address as data for selective resets. Comments about presence variable contents. VARx_RDY/VARx_ACC signals introduced. U_ACER status added. NOSTAT signal introduced. Received PDU type and length of variable stored in memory. Modified text for external logic for reset by long message. Added VAR1B0 reset assist output. Multiplexed identification inputs.
1.2	26-06-2009	All	8-bit Wishbone bus. Significance/refreshment style implementation. No VAR2B0. Reset variable adressed by broadcast. Examples of consumed and produced data. RSTON not activated on RSTIN.

Table of Contents

<Page Intentionally Left Blank>

1 INTRODUCTION

The NanoFIP is an FPGA component implementing the WorldFIP protocol that can be used in field devices able to communicate at the three standard speeds. The NanoFIP, that is developed as part of the WorldFIP insourcing project [1], is designed to be radiation tolerant by using different single event upset mitigation techniques such as triple module redundancy.

The device is used in conjunction with a FIELDRIVE driver chip and FIELDTR insulating transformer, both available from the company ALSTOM [2]. For implementing a robust reset external logic is needed.

To improve the radiation tolerance, the NanoFIP implements a minimal subset of the WorldFIP services:

- No need for a processor to set up data transfers.
- Consumption of one addressed variable.
- Consumption of one broadcast variable.
- Production of one addressed variable.
- Consumed variable size between 1 and 124 bytes.
- Produced variable size between 2 and 124 bytes settable by hardwired pins.
- Fixed variable addresses.
- WorldFIP speed settable to 31.25 kbit/s, 1 Mbit/s or 2.5 Mbit/s.
- Frame delimiters and Frame Checksum (FCS) using the WorldFIP/IEC standard.
- Produced variable uses PDU_TYPE 40h (compact value Protocol Data Unit).
- NanoFIP status (optional) and MPS (Manufacturing Periodical/Aperiodical Services) status sent together with the produced variable.
- Consumed variables are only updated when a correct checksum is received.
- Consumed freshness and significance data available in memory but not treated.
- Produced freshness and significance show data update status.
- Simple interface to variable data
    - Variable data transfer over an integrated memory accessible with an 8-bit WISHBONE System-on-Chip interconnection.
    - Possibility of stand-alone mode with 16 input and 16 output lines without the need to transfer data to or from memory.
    - Separate data valid outputs for each variable (consumed and produced).
- Several reset possibilities
    - External reset pin.
    - Addressed reset by broadcast consumed variable validated by station address as data.
    - Reset output available to external logic.

2 FUNCTIONALITY

2.1 VARIABLE STRUCTURE AND ADDRESSING

The station address xyh is set by the input pins SUBS[7:0].

After power-on the NanoFIP works as a station handling the following variables addressed by:

- ID_DAT = 14xyh.
  For the presence variable with a content as described in Section 3.2.
- ID_DAT = 10xyh.
  For the identification variable with a content as described in Section 3.3.
- ID_DAT = 05xyh.
  The station consumes a variable of any length up to 124 bytes followed by the 1-byte MPS status.
- ID_DAT = 04..h.
  The station consumes a variable of any length up to 124 bytes followed by the 1-byte MPS status. As the station address is not checked, this variable can be used for the consumption of variables that need to be broadcast to all stations on the segment.
- ID_DAT = 06xyh.
  The station produces a variable with a user-settable length content (2, 8, 16, 32, 64 or 124 bytes), an optional 1-byte NanoFIP status and a 1-byte MPS status.
- ID_DAT = E0..h
  The station consumes this 2-byte variable. As the station address is not checked, this variable can be broadcast to all stations on the segment. If the first byte contains the station address it will reset the NanoFIP and the FIELDRIVE component. If the second byte contains the station address it will assert the reset output pin RSTON. If none of the bytes contains the station address the variable is ignored.

2.2 FUNCTIONALITY OF VARIABLES

The produced and consumed variables are handled as follows:

- The consumed variables are only updated (assertion of VARx_RDY) when the Frame Checksum (FCS) is correct.
- The PDU_TYPE and length fields of consumed variables are stored in the communication memory before the actual data. The PDU_TYPE and length are not checked on reception.
- The refreshment and significance status of consumed variables is available in the communication memory as the last byte after the data. This byte is the MPS (Manufacturing Periodical/Aperiodical Services) status.
- The produced variable contains optionally (configurable by the pin NOSTAT) as the one but last byte the NanoFIP status. As last byte it contains the MPS status with the refreshment and the significance status as described in section 3.1.
- The produced data variable uses PDU_TYPE 40h (compact value Protocol Data Unit).

2.3 FRAME DELIMITERS AND FRAME CHECKSUM

The NanoFIP uses the frame delimiters and Frame Checksum (FCS) of the WorldFIP/IEC standard. The FIP/NFC coding is not supported.

2.4 RESPONSE AND SILENCE TIME

The WorldFIP response and silence times are fixed, but dependent on the chosen bit rate as is shown in Table 1. The Silence time defines the maximum waiting time for an RP_DAT after an ID_DAT has been received. The values are taken from the fastest settings from [3] and are also recommended values from [4] (Appendix A and Table 3.1).

Table 1: **Response time**

Speed	Response time	Silence time
31.25 kbit/s	640 µs	4160 µs
1 Mbit/s	10 µs	150 µs
2.5 Mbit/s	16 µs	100 µs

3 VARIABLE HANDLING: DATA

3.1 MPS STATUS DEFINITION

When transmitting a Manufacturing Periodical/Aperiodical Services (MPS) variable the status byte (last transmitted) is formatted as shown in Table 2:

Table 2: **MPS Status**

MPS Status bit	Name	Contents
0 (last transmitted)	refreshment	1/0
1		0
2	significance	1/0
3		0
4-7		0000

In stand-alone mode (SLONE=Vcc) the MPS Status byte will have the refreshment and significance bits always set to 1. In memory mode (SLONE=Gnd) the refreshment and significance bits will be set to 1 if the user has updated the produced variable (i.e. any address of the produced variable has been written to) since the last transmission of the variable (assertion of VAR3_RDY); otherwise the refreshment and significance bits will be set to 0. This emulates a refreshment time equal to the WorldFIP cycle time.

3.2 PRESENCE VARIABLE (14xyh)

After power-up, the NanoFIP is able to produce the SMMPS presence variable.

The frame contents dedicated to the NanoFIP embedded presence variable (14xyh identification number) are shown in Table 3. No MPS status is associated with this variable [4].

Table 3: **Presence variable fields**

Byte	Field name	Contents
1	PDU type	50h
2	Length	05h
3	3	80h
4	4	03h
5	LG_VAR_I	00h
6	Reserved	F0h
7	BA status | priority	00h

Implementation detail: for compatibility the NanoFIP sends the same presence variable as the MICROFIP. According to [4] bytes 3 and 4 (without a name or definition for these) contain the shown fixed values, byte 5 should in fact contain the length of the Identification variable, byte 6 is reserved and should contain 00h while byte 7 contains the BA status and priority (00h: BA not supported, BA priority le+).

3.3 IDENTIFICATION VARIABLE (10xyh)

After power-up, the NanoFIP is able to produce the SMMPS identification variable. The contents of the Constructor and Model fields are partially configurable by connecting C_ID[3:0] and M_ID[3:0] to Ground, Vcc, S_ID[1] or S_ID[0] (see section 7.1).

The frame contents dedicated to the NanoFIP identification variable (10xyh identification number) is shown in Table 4. No MPS status is associated with this variable [4].

Table 4: **Identification variable fields**

Byte	Field name	Contents
1	PDU type	52h
2	Length	08h
3	WorldFIP profile	01h
4	Class	00h
5	Constructor (first byte)	00h
6	Constructor (second byte)	Const[7:0]h
7	Model (first byte)	Model[7:0]h
8	Model (second byte)	00h
9	Version	00h
10	User	00h

3.4 CONSUMED VARIABLE VAR1 (05xyh)

Variables of any size up to 124 bytes can be received with the variable address 05xyh. The Frame Checksum (FCS) is verified and should be valid in order to declare the variable valid. If the received variable is not valid, VAR1_RDY will not be asserted and the communication memory (see section 5.1) may contain invalid data.

The PDU_TYPE and length fields of consumed variables are available in the communication memory as the first bytes before the actual data (see Table 5). The length field is not compared to the actual received length. The MPS status byte (see section 3.1) received from the network is also stored in the communication memory, directly after the data. This MPS status byte, containing the refreshment and significance bits, is not interpreted by the NanoFIP.

Table 5: **Consumed variable in communication memory with example**

Memory offset	Contents
0	PDU type
1	Length (n-1)
2	Data byte 0
3	Data byte 1
…	
n-1	Data byte n-2
n	MPS Status

Memory offset	Contents
0	40h
1	09h
2	01h
3	02h
…	…
9	08h
10	0Ah

3.5 CONSUMED BROADCAST VARIABLE VAR2 (04..h)

Any variable with the address 04..h (i.e. independent of the station address) is received in the same way as a normal consumed variable. This broadcast variable uses the signal VAR2_RDY and has its own area in the communication memory (see section 5.1).

3.6 PRODUCED VARIABLE VAR3 (06xyh)

The size of the produced variable is configured by the external pins P3_LGTH[2:0] (see pin description, section 7.1).

Table 6 shows how the data is sent from the communication memory to the WorldFIP bus. If the pin NOSTAT is grounded, the one but last byte sent contains the status of the NanoFIP interface as described in Table 8. If the pin NOSTAT is connected to Vcc, this status byte is not sent as is shown in Table 7. As the last byte the MPS status (see section 3.1) is automatically added at the end of the variable and sent to the network.

Table 6: **Produced variable as sent to WorldFIP (NOSTAT=Gnd) with example**

Memory offset	Contents
not from memory	PDU type
not from memory	Length (n-1)
2	Data byte 0
3	Data byte 1
…	
n-2	Data byte n-4
n-1	NanoFIP status
n	MPS Status

Memory offset	Contents
not from memory	40h
not from memory	0Ah
2	01h
3	02h
…	…
9	08h
10	00h
11	0Ah

Table 7: **Produced variable as sent to WorldFIP (NOSTAT=Vcc) with example**

Memory offset	Contents
not from memory	PDU type
not from memory	Length (n-1)
2	Data byte 0
3	Data byte 1
…	
n-1	Data byte n-2
n	MPS Status

Memory offset	Contents
not from memory	40h
not from memory	09h
2	01h
3	02h
…	…
9	08h
10	0Ah

Table 8: **NanoFIP Status**

NanoFIP Status bit	Name	Contents
0	r1	reserved, ignore on read
1	r2	reserved, ignore on read
2	u_cacer	User consumed variable access error
3	u_pacer	User produced variable access error
4	r_bner	Received bit number error
5	r_fcser	Received FCS error
6	t_txer	Transmit error (FIELDRIVE)
7	t_wder	Watchdog error (FIELDRIVE)

Bits 0 and 1 Reserved bits, ignore the value.

Bit 2=1 The user logic accessed a consumed variable (VAR1_ACC or VAR2_ACC high) when this variable was not ready (VAR1_RDY respectively VAR2_RDY was low). This may mean that the consumed variable was modified while it was being updated from the WorldFIP bus.

Bit 3=1 The user logic accessed the produced variable (VAR3_ACC high) when this variable was not ready (VAR3_RDY low). This may mean that the produced variable was modified by the user while it was being sent to the WorldFIP bus.

Bit 4=1 A variable arrived for this station with a number of bits not corresponding to a multiple of eight.

Bit 5=1 A variable arrived for this station and the Frame Checksum was wrong.

Bit 6=1 The FIELDRIVE signalled a transmission error.

Bit 7=1 The FIELDRIVE signalled a watchdog timer problem.

Bits 2 to 5 contain the status since the previously produced variable was sent to the WorldFIP bus. They are reset after the transmission of the produced variable.

Bits 6 and 7 are directly coming from the FIELDRIVE signals and are only reset when the NanoFIP is reset.

4 VARIABLE HANDLING: RESET

4.1 CONSUMED VARIABLE (E0..h)

The station consumes the 2-byte variable and MPS status (3 bytes total) with the address E0..h (i..e. independent of the station address). Whenever it receives this variable it will verify the data. If the first data byte contains the station address it will perform a reset of the NanoFIP and the FIELDRIVE. If the second data byte contains the station address it will assert the reset output pin RSTON. If none of the two data bytes contains the station address the variable is ignored.

Note that if the NanoFIP or FIELDRIVE chip has serious problems (e.g. a state-machine blocked, analog receiver not working), it will not be able to correctly receive this variable. For critical applications it is therefore suggested to implement external logic to reset the NanoFIP.

4.2 EXTERNAL RESET LOGIC

External logic may implement a very robust reset mechanism based on the timing of the Carrier Detect signal from the FIELDRIVE. For example, if the external logic detects bus activity with a data transfer of messages with a length of over 200 bytes it could activate the reset input of the NanoFIP.

Other user logic may implement a reset based on the data contents or on detecting no activity on the VARx_RDY signals.

5 MEMORY MAP

5.1 MEMORY MAP

When the pin SLONE is grounded, the exchange of the data of the three variables goes over three memory blocks organised as shown in Table 9. Access is done via an 8-bit wide Wishbone interface [5].

Table 9: **NanoFIP memory map**

Address ADR_I[9:0]	Contents	Size	Access
00_0000_0000b	PDU type Variable 1	1 byte	Read only
00_0000_0001b	Length Variable 1	1 byte	Read only
00_0000_0010b | **00**_1111_1110b	**Variable 1 05xyh (Consumed)**	max. 124 bytes followed by MPS status	Read Only
00_1111_1111b	reserved	1 byte	Do not access
01_0000_0000b	PDU type Variable 2	1 byte	Read only
01_0000_0001b	Length Variable 2	1 byte	Read only
01_0000_0010b | **01**_1111_1110b	**Variable 2 04..h (Consumed, broadcast)**	max. 124 bytes followed by MPS status	Read Only
01_1111_1111b	reserved	1 byte	Do not access
10_1000_0000b	reserved	1 byte	Do not access
10_1000_0001b	reserved	1 byte	Do not access
10_0000_0010b | **10**_0111_1100b	**Variable 3 06xyh (Produced)**	max. 124 bytes	Write/Read
10_0111_1110b	reserved	1 byte	Do not access
10_0111_1111b	reserved	1 byte	Do not access
11_1000_0000b | **11**_1111_1111b	reserved	128 bytes	Do not access

6 USER INTERFACE

6.1 STARTUP

After power-on the NanoFIP automatically starts up using the settings as defined by the input pins RATE[1:0], SUBS[7:0], M_ID[3:0], C_ID[3:0] and P3_LGTH[2:0].

The NanoFIP will directly be available to respond to the different WorldFIP requests such as Presence and Identification and it will consume and produce the variables.

6.2 STAND-ALONE MODE

When the pin SLONE is connected to Vcc, the NanoFIP will operate in stand-alone mode. In this mode the NanoFIP will consume a variable containing 2 bytes of data and a 1-byte MPS status (3 bytes total) and will produce 4-byte variables (2 bytes of data + NanoFIP status + MPS status). If the signal NOSTAT is connected to Vcc, the NanoFIP status will not be sent (i.e. it will produce only 2 bytes of data + MPS status). In stand-alone mode the NanoFIP will not be able to receive the broadcast variable.

6.2.1 CONSUMED VARIABLE

In stand-alone mode the consumed variable (05xyh) will be available on the pins DAT_O[15:0] and the reception of new variable data will be signalled by the VAR1_RDY signal. The assertion of VAR1_RDY signals that a valid update (with a correct FCS) has been received for this variable. When VAR1_RDY is asserted, DAT_O[15:0] will be stable and can safely be read. When an update of the variable is about to arrive over WorldFIP (end of ID_DAT), VAR1_RDY will be deasserted and the output may change. It is the user’s responsibility to copy the data if needed before (e.g. on the rising edge of VAR1_RDY).

6.2.2 PRODUCED VARIABLE

In stand-alone mode the produced variable (06xyh) should be presented at the pins DAT_I[15:0]. When VAR3_RDY is asserted, the user may change the input, and the NanoFIP will sample the data on the first clock cycle after it has deasserted VAR3_RDY. It is the user’s responsibility to make sure the data is stable at the moment of sampling.

6.3 MEMORY MODE

When the pin SLONE is grounded, the NanoFIP will operate in the memory mode. In this mode the NanoFIP will have its full functionality and variables of any size up to 124 bytes can be consumed and those set by the P3_LGTH[2:0] pins can be produced.

6.3.1 CONSUMED VARIABLES

Whenever a consumed variable is received the VAR1_RDY (05xyh) or VAR2_RDY (04..h) signal will be asserted (see Figure 1). During the time the VAR1_RDY or VAR2_RDY is asserted, the data in the memory will be stable and the user may safely read the contents of the memory. The data will be stable in the memory until a new consumed variable is received from the WorldFIP bus, which will be signalled by the deassertion of VAR1_RDY or VAR2_RDY. It is the user’s responsibility to copy the data if needed before, e.g. by guaranteeing to read the variable within the variable’s WorldFIP scanning period after assertion of VAR1_RDY or VAR2_RDY.

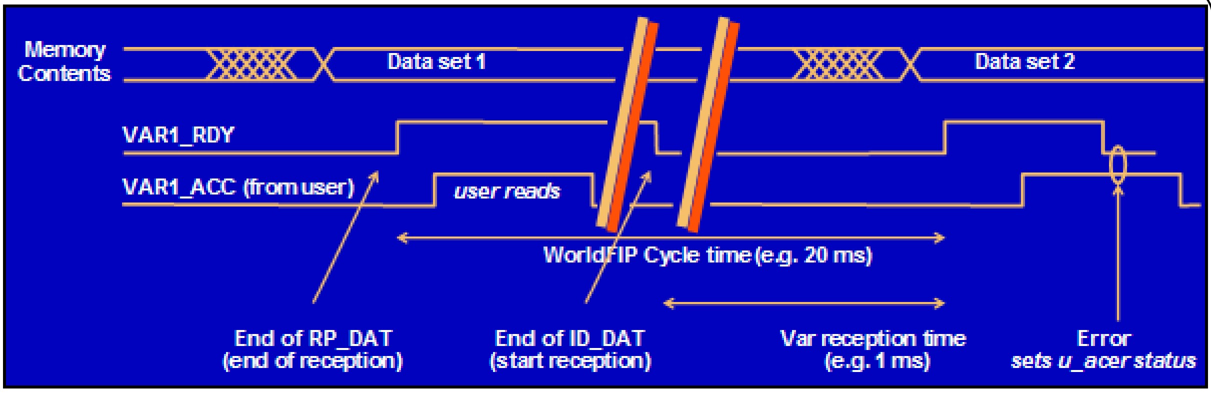

Figure 1: **Consumed variable timing**

6.3.2 PRODUCED VARIABLE

When VAR3_RDY is asserted, the user may write new variable data into the memory (see Figure 2). When the variable is requested and about to be sent to the WorldFIP bus (end of the next ID_DAT) the NanoFIP will deassert VAR3_RDY. It is the user's responsibility to guarantee that the data in the memory is stable during the time VAR3_RDY is deasserted, e.g. by guaranteeing to write the variable within the variable's WorldFIP scanning period after assertion of VAR3_RDY.

Implementation detail: after a reset, the NanoFIP will respond with the non-initialised contents of the memory with the MPS status byte set to 0. The MPS status byte will also be 0 if none of the produced variable addresses has been accessed since the last assertion of VAR3_RDY (see section 3.1).

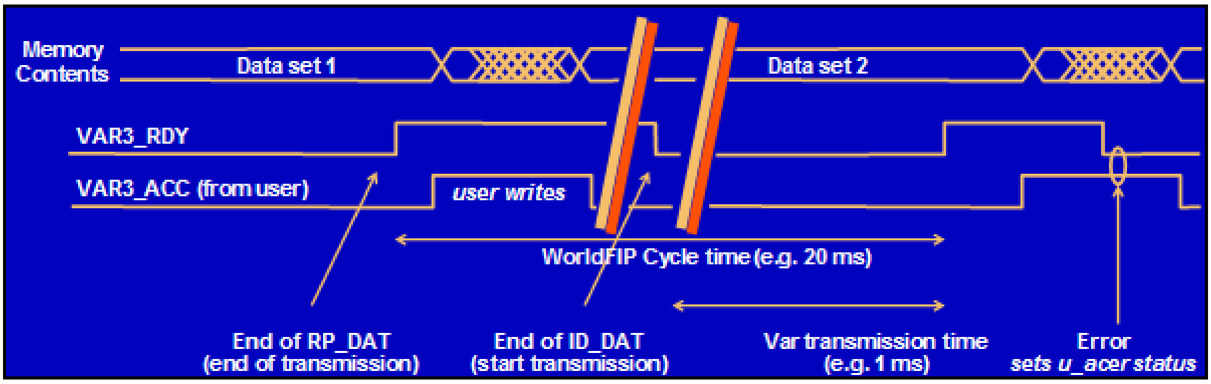

Figure 2: **Produced variable timing**

6.4 USER ACCESS VERIFICATION

The user may signal to the NanoFIP that it is accessing a variable by asserting the signal VAR1_ACC, VAR2_ACC or VAR3_ACC. When the VARx_ACC signal is asserted the NanoFIP will verify that the corresponding VARx_RDY signal is asserted (showing that the variable safely can be accessed). If the NanoFIP detects that the variable was not ready to be accessed (i.e. a consumed variable was being received from WorldFIP or the produced variable was being sent to WorldFIP), the NanoFIP status bit u_cacer or u_pacer respectively will be set (see Table 8), showing that certain data may be corrupted (Figure 1, Figure 2). When this happens the NanoFIP will continue to produce and consume variables. For applications not requiring this functionality the VARx_ACC signals may be grounded.

7 INTERFACE SIGNAL DEFINITION

7.1 PIN ASSIGNMENT

Nr.	Pin	Type	Description	MicroFIP equivalent
			POWER SUPPLY	
	P3V3	Supply	3.3 Volt power supply	
	P1V5	Supply	1.5 Volt power supply	
	Gnd	Supply	Ground	
			WORLDFIP SETTINGS	
	RATE[1:0]	Input	Bit rate 00: 31.25 kbit/s 01: 1 Mbit/s 10: 2.5 Mbit/s 11: reserved, do not use	A[1:0]
	SUBS[7:0]	Input	Subscriber number coding. Station address.	
	S_ID[1:0]	Output	Identification selection (see M_ID, C_ID)	
	M_ID[3:0]	Input	Identification variable settings (see section 3.3). Connect the ID inputs either to Gnd, Vcc, S_ID[0] or S_ID[1] to obtain different values for the Model data (i=0,1,2,3). M_ID[i] connected to: Gnd S_ID0 SID1 Vcc Model [2*i] 0 1 0 1 Model [2*i+1] 0 0 1 1	
	C_ID[3:0]	Input	Identification variable settings (see section 3.3). Connect the ID inputs either to Gnd, Vcc, S_ID[0] or S_ID[1] to obtain different values for the Constructor data (i=0,1,2,3). C_ID[i] connected to: Gnd S_ID0 SID1 Vcc Constructor [2*i] 0 1 0 1 Constructor [2*i+1] 0 0 1 1	
	P3_LGTH[2:0]	Input	Produced variable data length 000: 2 Bytes 001: 8 Bytes 010: 16 Bytes 011: 32 Bytes 100: 64 Bytes 101: 124 Bytes 110: reserved, do not use 111: reserved, do not use Actual size: +1 NanoFIP Status byte +1 MPS Status byte (last transmitted)	
			FIELDRIVE	
	FD_RSTN	Output	Initialisation control, active low	
	FD_WDGN	Input	Watchdog on transmitter	WTC1X

Nr.	Pin	Type	Description	MicroFIP equivalent
	FD_TXER	Input	Transmitter error	TER1X
	FD_TXENA	Output	Transmitter enable	TXE
	FD_TXCK	Output	Line driver half bit clock	TXCK
	FX_TXD	Output	Transmitter data	TXD
	FX_RXA	Input	Reception activity detection	RXA1X
	FX_RXD	Input	Receiver data	
			USER INTERFACE, General signals	
	CLK	Input	40 MHz clock	CLK
	SLONE	Input	Stand-alone mode	SLONE
	NOSTAT	Input	If connected to Vcc, disables sending of NanoFIP status together with the produced data.	
	RSTIN	Input	Initialisation control, active low	
	RSTON	Output	Reset output, active low. Active when the reset variable is received and the second byte contains the station address.	
			USER INTERFACE, non WISHBONE	
	VAR1_RDY	Output	Signals new data is received and can safely be read (Consumed variable 05xyh). In stand-alone mode one may sample the data on the first clock edge VAR1_RDY is high.	
	VAR1_ACC	Input	Signals that the user logic is accessing variable 1. Only used to generate a status that verifies that VAR1_RDY was high when accessing. May be grounded.	
	VAR2_RDY	Output	Signals new data is received and can safely be read (Consumed broadcast variable 04xyh).	
	VAR2_ACC	Input	Signals that the user logic is accessing variable 2. Only used to generate a status that verifies that VAR2_RDY was high when accessing. May be grounded.	
	VAR3_RDY	Output	Signals that the variable can safely be written (Produced variable 06xyh). In stand-alone mode, data is sampled on the first clock after VAR_RDY is deasserted.	
	VAR3_ACC	Input	Signals that the user logic is accessing variable 3. Only used to generate a status that verifies that VAR3_RDY was high when accessing. May be grounded.	
			USER INTERFACE, WISHBONE SLAVE	
	CLK_I	Input	Wishbone clock. May be independent of CLK.	
	DAT_I(15:0)	Input	Data in	
	DAT_O(15:0)	Output	Data out	
	ADR_I(9:0)	Input	Address	

Nr.	Pin	Type	Description	MicroFIP equivalent
	RST_I	Input	Wishbone reset. Does not reset other internal logic.	
	STB_I	Input	Strobe	
	ACK_O	Output	Acknowledge	
	WE_I	Input	Write enable	

7.2 WISHBONE DATASHEET

Description	Specification
General description	3 times 256 x 8-bit memory
Supported cycles	SLAVE READ/WRITE
Data port, size Data port, granularity Data port, maximum operand size Data transfer ordering Data transfer sequencing	8-bit 8-bit 8-bit Big endian and/or little endian Undefined
Clock frequency constraints	40 MHz max
Supported signal list and cross reference to equivalent WISHBONE signals	Signal name — WISHBONE Equiv. ACK_O — ACK_O ADR_I(9:0) — ADR_I() CLK_I — CLK_I DAT_I(15:0) — DAT_I() DAT_O(15:0) — DAT_O() STB_I — STB_I WE_I — WE_I
Special requirements	none

8 REFERENCES

[1] CernFIP pages. http://www.ohwr.org/twiki/bin/view/OHR/CernFIP/CernFIP

[2] ALSTOM FIELDRIVE and FIELDTR documentation. http://www.fipware.fr/

[3] Adaptation des Temps de Retournement et Silence des Agents (MICROFIP) sur ceux du MANAGER (FULLFIP), CERN AB-Note-2005-023(CO), R. Brun, 2005. http://cdsweb.cern.ch/record/834924

[4] MICROFIP HANDLER Software Release 1 User Reference Manual, ALS 50202 f-en, Alstom, 2007. http://www.fipware.fr/pdfs/50202_nt.pdf

[5] Specification for the: WISHBONE System-on-Chip (SoC) Interconnection Architecture for Portable IP Cores, Revision: B.3, Released: September 7, 2002. http://www.opencores.org/downloads/wbspec_b3.pdf