MICROFIP
User Reference Manual

ALS 50280 f–en

First issue: 11–1996
This edition: 09–2003

Meaning of terms that may be used in this document / Notice to readers

WARNING

Warning notices are used to emphasize that hazardous voltages, currents, temperatures, or other conditions that could cause personal injury exist or may be associated with use of a particular equipment.

In situations where inattention could cause either personal injury or damage to equipment, a Warning notice is used.

Caution

Caution notices are used where there is a risk of damage to equipment for example.

Note

Notes merely call attention to information that is especially significant to understanding and operating the equipment.

Index letter	Date	Nature of revision
b	**10–1999**	Use of VY27257 MICROFIP chip. Update of information related to: . microcontrolled operating mode, . control and status registers, . hardware configuration options, . user microcontroller interface timings. Deletion of information related to Safeit bank register.
c	**02–2000**	ALSTOMISATION branding of manual
d	**07–2001**	New: Figure 8.1 Redundancy Management . Modifications in the figures in Chapter 8 . Minor modifications throughout the document
e	**04–2003**	Modification of ALSTOM Technology into ALSTOM Transport
f	**09–2003**	. Modifications in the technical characteristics in Chapter 7 . Modification of the pin–out figure in Chapter 1

Revisions

1. PURPOSE OF MANUAL AND DOCUMENTED VERSION

This document describes the functionality of the MICROFIP component.

2. CONTENT OF THIS MANUAL

Chapter 1. Overview: gives the main features of the MICROFIP chip, its functional description and defines interface signals.

Chapter 2. WorldFIP communication resources: describes MICROFIP's two operating modes.

Chapter 3. Programming interface model in microcontrolled mode: gives the elements and the procedure that must be followed when programming in microcontrolled mode.

Chapter 4. Hardware configuration options: describes for each operating mode and, in general, the hardware configuration options.

Chapter 5. User Microcontroller interface: describes how to use the User Microcontroller Interface.

Chapter 6. Medium attachment units interface: describes how to use the Medium Attachment Units.

Chapter 7. Characterization: gives the environment, operating and consumption characteristics and describes basic clock generation.

Chapter 8. Application diagrams.

3. RELATED PUBLICATIONS

[1] MICROFIP HANDLER User Reference Manual ALS 50202.

4. WE WELCOME YOUR COMMENTS AND SUGGESTIONS

ALSTOM strives to produce quality technical documentation. Please take the time to fill in and return the "Reader's Comments" page if you have any remarks or suggestions

Preface

Reader's comments

ALS 50280 f–en MICROFIP User Reference Manual

Your main job is:

- ☐ System designer
- ☐ Distributor
- ☐ System integrator
- ☐ Installer
- ☐ Programmer
- ☐ Maintenance
- ☐ Operator
- ☐ Other (specify below)

If you would like a personal reply, please fill in your name and address below:

COMPANY: . . . . . NAME: . . . . . .

ADDRESS: . . . . . .

. . . . . COUNTRY: . . . . . .

Send this form directly to your ALSTOM sales representative or to this address:

ALSTOM Transport
Information Solutions
Customer Information Publishing (CIP)
23–25 avenue Morane Saulnier
92364 Meudon la Forêt Cedex
France
Fax: +33 (0)1 46 29 10 21

All comments will be considered by qualified personnel.

REMARKS

Continue on back if necessary.

Reader's comments

Contents

CHAPTER 1 – OVERVIEW

1. FEATURES 1–1
   1.1. Standalone 1–1
   1.2. Microcontrolled 1–1

2. FUNCTIONAL DESCRIPTION 1–3
   2.1. Block Partition 1–3
      2.1.1. Control 1–3
      2.1.2. Dual Port Memory 1–3
      2.1.3. User Interface 1–3
      2.1.4. PIA and PIB Input/Output Ports and Subscriber Number Input Port 1–4
      2.1.5. Protocol Machine 1–4
      2.1.6. UART 1–4
      2.1.7. Medium Redundancy Management 1–5
   2.2. Block Diagram 1–5

3. INTERFACE SIGNAL DEFINITION 1–6
   3.1. Pin–out 1–6
   3.2. Signal Description 1–7

CHAPTER 2 – WORLDFIP COMMUNICATION RESOURCES

1. STANDALONE OPERATING MODE 2–1
   1.1. Variable Structure and Addressing 2–1
   1.2. Mapping of Parallel Ports in Variables 2–2
   1.3. Start–up Sequence Definition 2–4
   1.4. Configuration Variable Contents (03xy variable) 2–5
   1.5. Parallel Port Configuration 2–6
   1.6. Demultiplexed Address Bus 2–6

2. MICROCONTROLLED OPERATING MODE 2–7
   2.1. Variable Structure and Addressing 2–7
   2.2. Refreshment and Promptness 2–8
   2.3. Variable Configuration (COBMPS) 2–9
   2.4. Variable Read and Write User Access Handling 2–10
   2.5. Messaging Exchanges 2–10
   2.6. Message Contents 2–11
   2.7. Message Services 2–11
   2.8. Variable and Message Control – Interrupt Management 2–12
   2.9. Port Configuration 2–12

3. MPS STATUS DEFINITION 2–13

4. PRESENCE VARIABLE 2–13

5. IDENTIFICATION VARIABLE 2–14

Contents

CHAPTER 3 – PROGRAMMING INTERFACE MODEL MICROCONTROLLED MODE

1. OVERALL INFORMATION 3–1

2. MEMORY MAP 3–2

3. CONTROL AND STATUS REGISTERS 3–3
   - 3.1. Register mapping 3–3
     - 3.1.1. Regis Bank 3–3
     - 3.1.2. Regi2 Bank 3–4
     - 3.1.3. Port Bank 3–4
   - 3.2. Register Contents 3–5
     - 3.2.1. Regis Bank 3–5
     - 3.2.2. Regi2 Bank 3–10
     - 3.2.3. Ports Bank 3–11

4. FLOW CHARTS 3–12
   - 4.1. Initialization Sequence 3–12
   - 4.2. Variable Write Sequence 3–14
   - 4.3. Variable Read Sequence 3–15
   - 4.4. Message Write Sequence 3–16
   - 4.5. Message Read Sequence 3–17
   - 4.6. Read Interrupt Vector Sequence 3–17
   - 4.7. Read Event Status Sequence 3–18

CHAPTER 4 – HARDWARE CONFIGURATION OPTIONS

1. STANDALONE OPERATING MODE OPTIONS 4–1
   - 1.1. Standalone Mode Selection 4–1
   - 1.2. Bit Rate Selection 4–1
   - 1.3. Frame Delimiters and FCS Calculation 4–1
   - 1.4. Response Time Selection 4–2
   - 1.5. Silence Time Selection 4–2
   - 1.6. MAU Type Selection 4–3
   - 1.7. Starting Mode Selection 4–3
   - 1.8. DEFSUP Input Selection 4–3

2. MICROCONTROLLED OPERATING MODE OPTIONS 4–4
   - 2.1. Microcontrolled Mode Selection 4–4
   - 2.2. Microcontroller Interface Selection 4–4
   - 2.3. Cascaded Mode Selection 4–4
   - 2.4. DEFSUP Input Selection 4–4

3. GLOBAL OPTIONS 4–5
3.1. Monomedium Operation Selection 4–5
3.2. Parallel input/output Ports Control 4–5
3.3. Basic Clock Selection 4–5

4. CASCADE MODE 4–6

CHAPTER 5 – USER MICROCONTROLLER INTERFACE

1. MICROCONTROLLER SELECTION 5–1

2. FUNCTIONALITY 5–1

3. TIMING 5–2
3.1. Initialisation 5–2
3.2. Intel multiplexed bus – write cycle 5–2
3.3. Intel demultiplexed bus – write cycle 5–3
3.4. Intel multiplexed bus – read cycle 5–4
3.5. Intel demultiplexed bus – read cycle 5–5
3.6. Motorola write and read accesses – multiplexed buses 5–6
3.7. Motorola write and read accesses – demultiplexed buses 5–7
3.8. Parameter Definition 5–8

CHAPTER 6 – MEDIUM ATTACHMENT UNIT INTERFACE

1. INTERFACE MODE SELECTION (FIELDRIVE OR CREOL) 6–1

2. INTERFACE SIGNAL DEFINITION 6–1

3. INTERFACE SIGNAL TIMINGS 6–3
3.1. Start of Frame – Waveforms 6–3
3.1.1. FIP/NFC physical layer configuration 6–3
3.1.2. WorldFIP physical layer configuration 6–4
3.2. End of Frame – Waveforms 6–4
3.2.1. FIP/NFC physical layer configuration 6–4
3.2.2. WorldFIP/IEC physical layer configuration 6–5
3.3. Start of Frame – Timing diagram 6–6
3.4. End of Frame – Timing diagram 6–6

4. MICROFIP RECEIVER 6–8

Contents

CHAPTER 7 – CHARACTERISTICS

1. ABSOLUTE MAXIMUM RATINGS 7–1

2. RECOMMENDED OPERATING CONDITIONS 7–1

3. CONSUMPTION 7–2

4. BASIC CLOCK GENERATION 7–3

CHAPTER 8 – APPLICATION DIAGRAMS

Figures

Figure 1.1 – Block Diagram 1–5
Figure 1.2 – Pin–out 1–6

Figure 2.1 – Standalone mode MPS variables 2–2
Figure 2.2 – Microcontrolled mode MPS variables 2–8
Figure 2.3 – Three variables of dual port communication memory 2–10
Figure 2.4 – Identification variable format 2–15

Figure 4.1 – Transceiver interface 4–6

Figure 5.1 – Intel multiplexed bus – write cycle 5–2
Figure 5.2 – Intel demultiplexed bus – write cycle 5–3
Figure 5.3 – Intel multiplexed bus – read cycle 5–4
Figure 5.4 – Intel demultiplexed bus – read cycle 5–5
Figure 5.5 – Motorola write and read accesses – multiplexed buses 5–6
Figure 5.6 – Motorola write and read accesses – demultiplexed buses 5–7

Figure 6.1 – Received frame composition 6–8

Figure 7.1 – Basic 20–MHz clock 7–3

Figure 8.1 – Redundancy management with mf_redundancy compiling option selected 8–2
Figure 8.2 – Intel 8051 with external memory 8–3
Figure 8.3 – MICROFIP used with Intel 8051 (mf_redundancy compiling option selected) 8–4
Figure 8.4 – MICROFIP used with Intel 8051
(without mf_redundancy compiling option selected) 8–5
Figure 8.5 – MICROFIP used in standalone mode 8–6
Figure 8.6 – FIELDRIVE line transceiver
(Three speeds and redundancy options) 8–7

Tables

Table 2.1 – Configuration variable contents 2–5
Table 2.2 – Standalone mode parallel port configuration 2–6
Table 2.3 – Point–to–point and broadcast messaging address format 2–11
Table 2.4 – Status byte format 2–13
Table 2.5 – Presence variable frame content 2–13
Table 2.6 –Identification variable frame content 2–14
Table 2.7 – Constructor and Model fields associated with the corresponding input pins 2–15

Table 3.1 – Memory structure 3–2
Table 3.2 – Regis bank 3–3
Table 3.3 – Regi2 bank 3–4
Table 3.4 – Port bank 3–4
Table 3.5 – Initialization Sequence Flowchart 3–12
Table 3.6 – Variable Write Sequence Flowchart 3–14
Table 3.7 – Variable Read Sequence Flowchart 3–15
Table 3.8 – Message Write Sequence Flowchart 3–16
Table 3.9 – Message Read Sequence Flowchart 3–17
Table 3.10 – Read Interrupt Vector Sequence Flowchart 3–17
Table 3.11 – Read Event Status Sequence Flowchart 3–18

Table 4.1 – Bit rate selection 4–1
Table 4.2 – Frame delimiters and FCS calculation 4–1
Table 4.3 – Response time selection 4–2
Table 4.4 – Silence time selection 4–2
Table 4.5 – MAU type selection 4–3
Table 4.6 – Starting mode selection 4–3
Table 4.7 – DEFSUP input selection 4–3
Table 4.8 – Microcontroller interface selection 4–4
Table 4.9 – Cascaded mode selection 4–4
Table 4.10 – DEFSUP input selection 4–4
Table 4.11 – Parallel input/output port control 4–5
Table 4.12 – Basic clock selection 4–5

Table 5.1 – Microcontroller Parameter Definition 5–8

Table 7.1 – Technical characteristics 7–1
Table 7.2 – Recommended operating conditons 7–1
Table 7.3 – Characteristics with a classical or low supply voltage 7–2
Table 7.4 – Maximum consumption 7–2

Chapter 1 Overview

1. FEATURES

MICROFIP AMIS 15016–528 is an ASIC solution implementing the WorldFIP protocol, to be used in field devices able to communicate at the three standard speeds: 31.25 kbits/s, 1 Mbit/s, 2.5 Mbits/s. In addition its static implementation guarantees low power consumption.

MICROFIP is a low–cost and easy–to–use solution which supplies a subset of the WorldFIP services, in particular it can be configured as a station, without being able to run the bus arbitrator. The frame delimiters and control sequence can be selected in compliance with the NFC46–604 FIP original standard or IEC11586–2. Network management facilities are directly embedded in the chip, such as the presence and identification variables. The identification contains a 32–bit tag.

The MICROFIP chip is packaged in the MQFP100 standard solution, and the 3.3–volt and 5–volt nominal power supply can be selected.

MICROFIP is able to operate with or without associated microcontroller, using various MAUs such as FIELDRIVE, CREOL for copper twisted pairs and others kind of supports (optical fiber, etc.). The two ways to run the MICROFIP, with or without microcontroller, are respectively identified **Standalone** and **Microcontrolled**, and offer benefits from dedicated features.

1.1. Standalone

MICROFIP operating in Standalone mode allows direct interfacing with the process inputs and outputs through two 8–bit bi–directional ports. The port states are directly mapped on the network using one produced variable (identifier: 06xy) for inputs, and one consumed variable (identifier: 05xy) for outputs.

MICROFIP operating in Standalone mode can be automatically started after power up initialization or after having received a start command from the configuration variable (identifier: 03xy). The configuration variable which is a consumed variable contains, in addition, bits dedicated to I/O port configuration.

MICROFIP operating in Standalone mode takes the xy subscriber number from a dedicated port, and the promptness and refreshment statuses are managed on the MPS variables and tuned by dedicated pins.

1.2. Microcontrolled

MICROFIP operating with a microcontroller doesn’t require any external logic glue for many of them (8051, 68HC11, etc.). The use of microprocessors (80186, 68000, etc.) is also simplified.

MICROFIP communication resources are accessed by the microcontroller through an embedded dual port memory which is shared with the protocol machine.

MICROFIP can be configured with 0 to 4 produced MPS variables, and 0 to 4 consumed MPS variables, using a global resource capability of 15 blocks of 8 bytes. The variable identifiers are allocated in the physical space (00xy to 07xy, with xy as subscriber number), with the possibility for one of the consumed variables to modify the identifier number, and to assign a number from the global addressing space (16 encoded bits), in a user interface register.

A MICROFIP memory of 120 bytes can be extended using another MICROFIP chip installed in “cascade” with the first one. The promptness and refreshment statuses are managed for all the MPS variables and the period can be selected from several possible values.

MICROFIP takes its subscriber number directly from the dedicated pins during chip set–up, and can be modified later by the microcontroller.

MICROFIP is fitted with a full duplex (transmit and receive) messaging channel allowing message exchanges with maximum size 128 bytes. The message exchanges may or may not be acknowledged according to user needs.

MICROFIP provides an interrupt mechanism attached to one produced variable (06xy identifier), to the four consumed variables, and to the messaging transmit and receive channels. Seven different factors may be at the origin of an interrupt.

2. FUNCTIONAL DESCRIPTION

2.1. Block Partition

MICROFIP is composed of the following block units:

- Control,
- Dual Port Memory,
- User Interface,
- Two parallel Input/Output Ports PIA and PIB, and an input Port for Subscriber Number,
- Protocol Machine,
- UART,
- Medium Redundancy Management.

2.1.1. Control

This block manages the initialization and clock signals used inside the chip.

It also handles the operating mode selections:

- operations with or without microcontroller,
- selection of the microcontroller used, if any,
- direction for the two ports PIA and PIB.

2.1.2. Dual Port Memory

This block contains the communication data base (variables and message contents) which is shared between the MICROFIP protocol machine and between the microcontroller and the I/O peripheral.

2.1.3. User Interface

This block allows a microcontroller to configure the MICROFIP component with a set of dedicated control and status registers.

It also handles the exchange between the microcontroller and the internal dual port memory which contains the communication data base (messages and variable contents).

Lastly this block handles the interrupt mechanism towards the microcontroller. The interrupts, when configured, are initiated upon network event recognition.

2.1.4. PIA and PIB Input/Output Ports and Subscriber Number Input Port

The PIA and PIB 8–bit ports can be configured by dedicated external input pins of the circuit in write or read direction mode.

When MICROFIP is operating without a microcontroller, these ports are linked to the corresponding network variables (produced variable for the port configured in input, consumed variable for the port configured in output). (Refer to Subsection 2.1.2.).

Note

In this mode, when the signal indicating a fault of the power supply is activated (DEFSUP), the refreshment status of all the variables is FALSE.

When MICROFIP is controlled by an external microcontroller, it can access the input/output ports through the microcontroller interface.

Lastly a dedicated SUBS 8–bit input port is used to assign an xy subscriber number. This subscriber number can be read and overwritten through the microcontroller interface.

2.1.5. Protocol Machine

This block manages network communication element exchanges, and is able to handle the following sequence:

- ID_DAT + RP_DAT for up to 8 MPS variables,
- ID_DAT + RP_DAT for the presence and identification variables,
- ID_DAT + RP_DAT_MSG for one MPS variable (with identifier 06xy and when handling the associated aperiodic messaging channel),
- ID_MSG + RP_MSG_ NOACK + RP_FIN for a broadcasting message,
- ID_MSG + RP_MSG_NOACK + RP_FIN for a point–to–point unacknowledged message,
- ID_MSG + RP_MSG_ACK + RP_ACK+/– + RP_FIN for an acknowledged message.

The protocol machine can be considered as a crossroad between the following elements:

- the UART input and output parallel ports,
- the PIA and PIB ports,
- the dual port memory which contains the data base.

2.1.6. UART

This block is derived from the FULLFIP2 chip. It allows management framing and FCS calculation with the possibility to be compliant with NFC or IEC standardization.

2.1.7. Medium Redundancy Management

This block is derived from the FIELDUAL chip. It allows MICROFIP to operate either a single or dual medium device.

In the latter case, the medium redundancy management is made either by the component (Standalone mode, see Chapter 8, Figure 8.6) or by host processor control (Microcontrolled mode). The use of the `mf_redundancy` MICROFIP HANDLER primitive requires the hardware implementation described in Chapter 8, Figure 8.3. For details about MICROFIP HANDLER see *ALS 50202 MICROFIP HANDLER User Reference Manua*l.

Figure 8.5 describes the hardware implementation to be used without using the `mf_redundancy` primitive.

To use the redundancy medium see Chapter 8, Figure 8.1.

2.2. Block Diagram

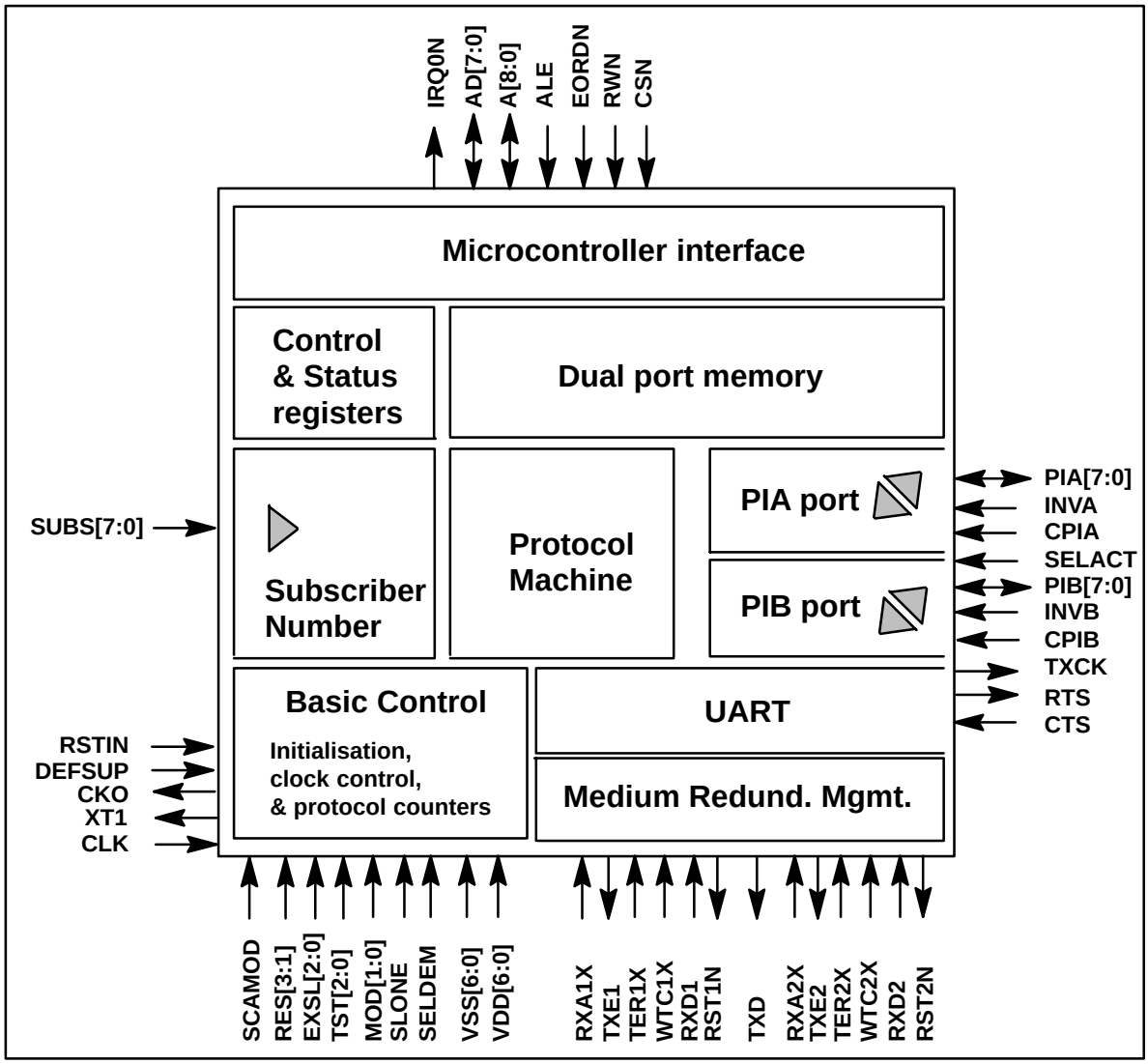


Figure 1.1 – Block Diagram

3. INTERFACE SIGNAL DEFINITION

3.1. Pin–out

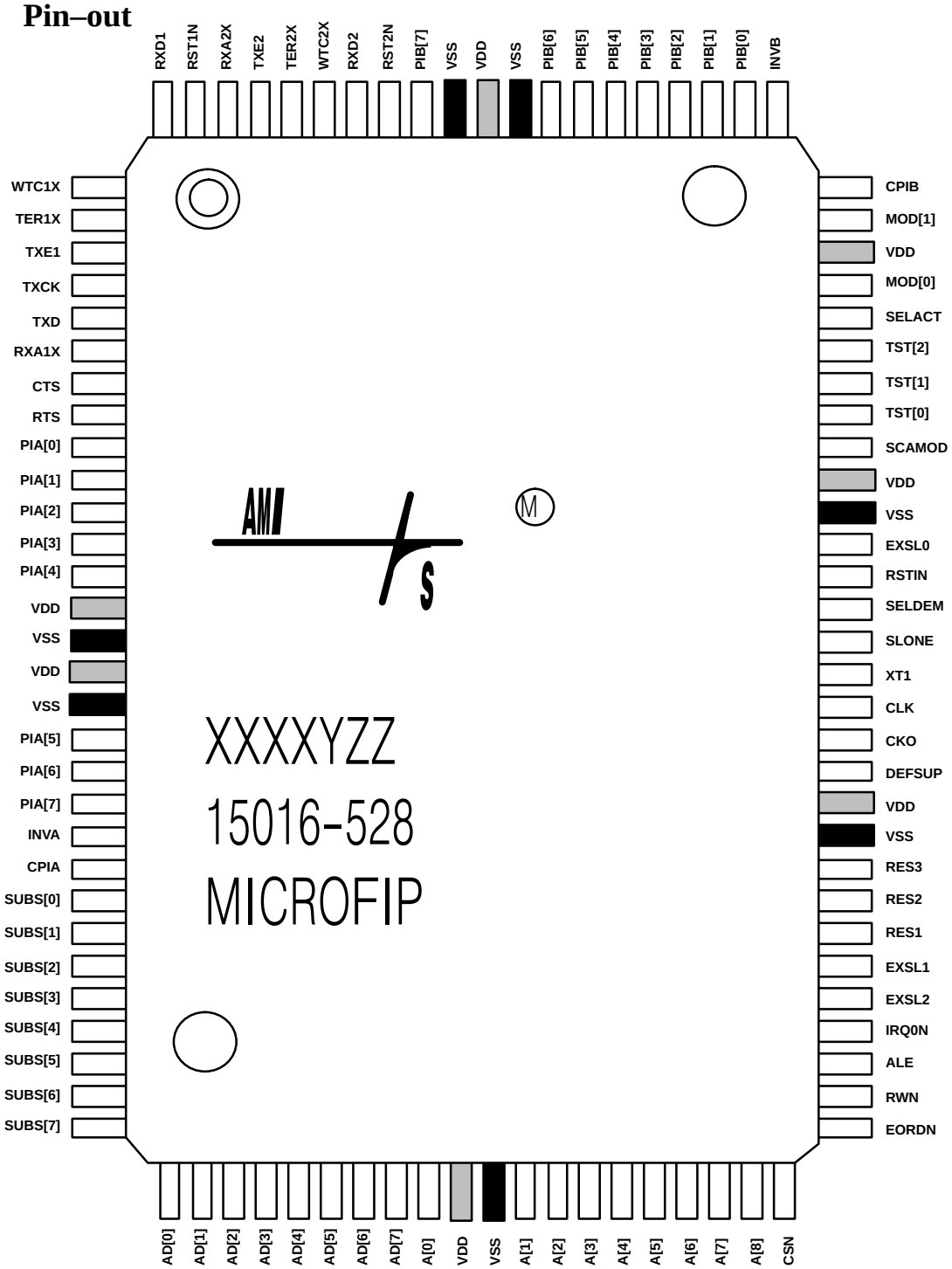


Figure 1.2 – Pin–out

3.2. Signal Description

Nr.	Pin	Type	Description
1	WTC1X	Input	Watch on transmitter, line driver 1
2	TER1X	Input	Transmitter error, line driver 1
3	TXE1	2 mA Output	Transmitter enable, line driver 1
4	TXCK	2 mA Output	Line driver half bit clock
5	TXD	2 mA Output	Transmitter data
6	RXA1X	Input	Reception activity detection, line driver 1
7	CTS	Input	Clear To Send
8	RTS	2 mA Output	Request to Send
9	PIA[0]	4 mA I/O	Parallel port A
10	PIA[1]	4 mA I/O	Parallel port A
11	PIA[2]	4 mA I/O	Parallel port A
12	PIA[3]	4 mA I/O	Parallel port A
13	PIA[4]	4 mA I/O	Parallel port A
14	VDD	Supply	Positive supply voltage
15	VSS	Supply	Negative supply voltage
16	VDD	Supply	Positive supply voltage
17	VSS	Supply	Negative supply voltage
18	PIA[5]	4 mA I/O	Parallel port A
19	PIA[6]	4 mA I/O	Parallel port A
20	PIA[7]	4 mA I/O	Parallel port A
21	INVA	Input	Parallel port A signals inversion control
22	CPIA	Input	Parallel port A signals direction control
23	SUBS[0]	Input	Parallel input port, subscriber number coding
24	SUBS[1]	Input	Parallel input port, subscriber number coding
25	SUBS[2]	Input	Parallel input port, subscriber number coding
26	SUBS[3]	Input	Parallel input port, subscriber number coding
27	SUBS[4]	Input	Parallel input port, subscriber number coding
28	SUBS[5]	Input	Parallel input port, subscriber number coding
29	SUBS[6]	Input	Parallel input port, subscriber number coding
30	SUBS[7]	Input	Parallel input port, subscriber number coding
31	AD[0]	4 mA I/O	Multiplexed address–data bus
32	AD[1]	4 mA I/O	Multiplexed address–data bus
33	AD[2]	4 mA I/O	Multiplexed address–data bus
34	AD[3]	4 mA I/O	Multiplexed address–data bus
35	AD[4]	4 mA I/O	Multiplexed address–data bus
36	AD[5]	4 mA I/O	Multiplexed address–data bus
37	AD[6]	4 mA I/O	Multiplexed address–data bus
38	AD[7]	4 mA I/O	Multiplexed address–data bus
39	A[0]	4 mA I/O	Demultiplexed address bus
40	VDD	Supply	Positive supply voltage

Nr.	Pin	Type	Description
41	VSS	Supply	Negative supply voltage
42	A[1]	4 mA I/O	Demultiplexed address bus
43	A[2]	4 mA I/O	Demultiplexed address bus
44	A[3]	4 mA I/O	Demultiplexed address bus
45	A[4]	4 mA I/O	Demultiplexed address bus
46	A[5]	4 mA I/O	Demultiplexed address bus
47	A[6]	4 mA I/O	Demultiplexed address bus
48	A[7]	4 mA I/O	Demultiplexed address bus
49	A[8]	Input	Address bit
50	CSN	Input	MICROFIP Chip select, active LOW
51	EORDN	Input	E or RDN control signal, active LOW
52	RWN	Input	RWN or WRN control signal, active LOW
53	ALE	Input	Address Latch Enable
54	IRQ0N	2 mA Output	Interrupt request, active LOW
55	EXSL2	Input	Silence timer selector in Standalone Mode. To be grounded otherwise
56	EXSL1	Input	Silence timer selector in Standalone Mode. To be grounded otherwise
57	RES1	Input	Identification configuration
58	RES2	Input	Identification configuration
59	RES3	Input	Identification configuration
60	VSS	Supply	Negative supply voltage
61	VDD	Supply	Positive supply voltage
62	DEFSUP	Input	Supply default indication. Active HIGH
63	CKO	2 mA Output	High speed output clock
64	CLK	Input	Quartz or oscillator input clock
65	XT1	Output	Quartz output
66	SLONE	Input	Standalone operating mode selection
67	SELDEM	Input	Starting mode option / cascading option
68	RSTIN	Input	Initialization control, active LOW
69	EXSL0	Input	Silence timer selector in Standalone Mode. To be grounded otherwise
70	VSS	Supply	Negative supply voltage
71	VDD	Supply	Positive supply voltage
72	SCAMOD	Input	Test mode
73	TST[0]	Input	Test & clocks control
74	TST[1]	Input	Test & clocks control
75	TST[2]	Input	Test & clocks control
76	SELACT	Input	Output ports driving mode: push–pull / open collector
77	MOD[0]	Input	Microcontroller type selection
78	VDD	Supply	Positive supply voltage
79	MOD[1]	Input	Microcontroller type selection
80	CPIB	Input	Parallel port B signals direction control

Nr.	Pin	Type	Description
81	INVB	Input	Parallel port B signals inversion control
82	PIB[0]	4 mA I/O	Parallel port B
83	PIB[1]	4 mA I/O	Parallel port B
84	PIB[2]	4 mA I/O	Parallel port B
85	PIB[3]	4 mA I/O	Parallel port B
86	PIB[4]	4 mA I/O	Parallel port B
87	PIB[5]	4 mA I/O	Parallel port B
88	PIB[6]	4 mA I/O	Parallel port B
89	VSS	Supply	Negative supply voltage
90	VDD	Supply	Positive supply voltage
91	VSS	Supply	Negative supply voltage
92	PIB[7]	4 mA I/O	Parallel port B
93	RST2N	2 mA Output	Initialization control, line driver 1
94	RXD2	Input	Received data, line driver 2
95	WTC2X	Input	Watch on transmitter, line driver 2
96	TER2X	Input	Transmitter error, line driver 2
97	TXE2	2 mA Output	Transmitter enable, line driver 2
98	RXA2X	Input	Reception activity detection, line driver 2
99	RST1N	2 mA Output	Initialization control, line driver 1
100	RXD1	Input	Received data, line driver 1

Note

Signal names ending with ”N” correspond to the active LOW signals.
Signal names ending with ”X” correspond to the active LOW or HIGH signals depending on the operation mode selected.

Chapter 2 WorldFIP Communication Resources

1. STANDALONE OPERATING MODE

1.1. Variable Structure and Addressing

When MICROFIP operates in Standalone mode, since its initialization it works as a station handling the following variables addressed by:

- ID_DAT = 14xy.

  For the presence variable with a content described in Section 4.

- ID_DAT = 10xy.

  For the identification variable with a content described in Section 5.

- ID_DAT = 03xy.

  If SELDEM input pin is active high, this variable is consumed and contains a start/reset order on 1 byte, 3 bytes dedicated to the station configuration, and the MPS status. If SELDEM is inactive low, this variable is not handled by the station.

- ID_DAT = 06xy.

  The station produces a 2–byte data and 1–byte MPS status variable. The 2–byte data contains the parallel ports PIA and PIB, whatever the port direction selected.

- ID_DAT = 05xy.

  The station consumes a 2–byte and 1–byte MPS status variable. The 2–byte data can drive the parallel ports PIA and PIB for those configured as outputs. For ports configured as inputs, the corresponding byte of the consumed variable is not used.

	Identifier (hex)	Type	Length (number of bytes)
Start/Reset & Configuration	03xy	C	5 (including the MPS status byte)
Output ports	05xy	C	3 (including the MPS status byte)
Input ports	06xy	P	3 (including the MPS status byte)

MPS variables used (Standalone mode)

xy is the station number, coded hexa.

P/C: Produced / Consumed

Figure 2.1 – Standalone mode MPS variables

1.2. Mapping of Parallel Ports in Variables

The PIA and PIB ports can be configured independently as inputs or outputs.

Independently of the ports configuration, MICROFIP collects their 16 bits to update the 06xy produced variable.

When a port is configured in outputs, its values are updated through the 05xy consumed variable, and the feedback will be inside the 06xy produced variable.

The produced variable contains the MPS status byte which contains the asynchronous refreshment status, and the meaning status, described in Section 3. In this mode, the refreshment status is true when the meaning status is true.

The consumed variable which contains the logic values for the bits of the output port is handled according to the asynchronous promptness status.

The following lines summarize the mechanisms which are attached to the input–output parallel ports and to the corresponding variables:

- Logic inversion capability between the port values and the variable values.
- Selection of the input/output port direction.
- Selection of the output driving mode (push–pull or open–drain).
- Selection of the fallback strategy on the output port (this action is performed when the corresponding promptness or refreshment variable is false, or when the significance bit is false): either the ports are forced in their non–active state, or they keep the Fall_Back_Value.
- Selection of the Fall_Back_Values on the output ports: one Fall_Back_Value for each bit or the Fall_Back_Value will be the image of the logic inversion selection.
- Selection of the input filtering period for input ports.
- Selection of the promptness value associated with output port validation.

When one of the following conditions is fulfilled:

- the promptness value is false, or
- the refreshment status of the consumed variable is false, or
- the station significance bit is false after reset, or
- a supply default is detected on the DEFSUP input pin,

then the corresponding bits of ports PIA and PIB (when configured as outputs) are:

- placed in their inactive state, or
- kept in the previous state, depending on the fallback strategy selected.

When the following conditions are fulfilled:

- the promptness and refreshment statuses of the consumed variable become valid, and
- the significance bit of the station is HIGH, and
- no fault indication is given on the DEFSUP input pin,

then the output values are again driven by the corresponding bits inside the consumed variable.

1.3. Start–up Sequence Definition

The starting phase depends on the SELDEM input pin value:

- if SELDEM = 0 the station activity is started just after station power–up. In this mode the 03xy configuration variable is not handled by the station. The meaning bit and the refreshment bit inside the MPS status of the produced variable 06xy are active HIGH just after station power–up, if the DEFSUP supply fault indication input pin is not active (when DEFSUP indicates a fault the meaning bit and the refreshment bit are false) (see Chapter 4),
- if SELDEM = 1 the station activity is controlled through the 03xy consumed variable, which manages the start and reset orders, and also the configuration bits for the parallel input/output ports PIA and PIB.

In the last case, and just after power up, the station activity is not started:

- the 06xy variable is produced, but its significance bit and the refreshment bit inside the MPS status are false,
- the output port bits remain in their inactive default state.

When the 03xy variable is consumed and contains a valid start order, and when no fault is indicated on the DEFSUP input pin, the associated configuration bits are evaluated, and the station is activated (the significance bit and the refreshment bit of the 06xy variable MPS status byte are valid and the 05xy consumed variable contents can drive the output ports).

When the station is active, the 03xy consumed variable contents are only checked to detect a potential reset order. The configuration information in this state is no longer taken into account.

A reset order inside the 03xy variable puts the station in inactive state to produced the same behavior as during the initial power–up reset.

1.4. Configuration Variable Contents (03xy variable)

The 03xy variable is composed of:

- 4–byte data,
- one MPS status byte.

Field name	Byte number	Coding
Start/Reset	First byte	96h: Start. 99h: Reset.
DEFSTRA Fallback Strategy Port A	Second byte, bit 0 *	DEFSTRA = 0: outputs forced to non active state DEFSTRA = 1: outputs forced to configured Fall_Back_Values
DEFSTRB Fallback Strategy Port B	Second byte, bit 1	DEFSTRA = 0: outputs forced to non active state DEFSTRA = 1: outputs forced to configured Fall_Back_Values
INFILA[1:0] Input filtering period Port A	Second byte, bits 3:2	INFILA[1:0] = 00: none 01: 1 ms 10: 4 ms 11: 10 ms
INFILB[1:0] Input filtering period Port B	Second byte, bits 5:4	INFILB[1:0] = 00: none 01: 1 ms 10: 4 ms 11: 10 ms
OUPROM[1:0] Output port Promptness Ports A & B	Second byte, bits 7:6	OUPROM[1:0] = 00: 50 ms 01: 250 ms 10: 1 s 11: 5 s
DEFVA[7:0] Fall_Back_Values On outputs Port A	Third byte	1 bit value for each port signal
DEFVB[7:0] Fall_Back_Values On outputs Port B	Fourth byte	1 bit value for each port signal

* bit 0 in a byte is the first transmitted

Table 2.1 – Configuration variable contents

1.5. Parallel Port Configuration

The following configuration options correspond to the PIA and PIB parallel ports.

	Standalone basic immediate start–up	**Standalone controlled start–up**
Starting mode selection	Input pin SELDEM = 0	Input pin SELDEM = 1
Logic inversion ports / variables	Input pins INVA, INVB	Input pins INVA, INVB
Port direction	Input pins CPIA, CPIB	Input pins CPIA, CPIB
Output port driving mode	Input pin SELACT	Input pin SELACT
Fallback Strategy	Fixed: Using inactive default value	Selectable: 03xy variable 1 bit by port
Configured Fall_Back_Values	Equals inactive default value: (INVA, INVB)	Tunable: 03xy variable for each pin
Input filtering period	Fixed: 10 ms	Tunable: 03xy variable by port: none, 1 ms, 4 ms, 10 ms
Output port promptness value	Fixed: 1 sec	Tunable: 03xy variable by port 50 ms, 250 ms, 1 s, 5 s.
Station Start / Station Reset	Automatic start after power up	Start/ Reset selection with 03y variable: 1 byte

Table 2.2 – Standalone mode parallel port configuration

1.6. Demultiplexed Address Bus

In Standalone mode, the demultiplexed address bus is used to define the Model of the device (see Table 2.7).

2. MICROCONTROLLED OPERATING MODE

In this mode the variables are configured using the MICROFIP control registers through the user microcontroller interface.

2.1. Variable Structure and Addressing

After initialization, the components handle the following variables addressed by:

- ID_DAT = 14xy.

  For the presence variable with a content described in Section 4.

- ID_DAT = 10xy.

  For the identification variable with a content described in Section 5.

In addition, the following user variables can be configured:

- 0 to 4 produced variables:

  00xy, 02xy, 04xy, 06xy.

- 0 to 4 consumed variables:

  01xy, 03xy, 05xy, 07xy or a global identifier.

The fourth consumed variable, Var7, can be configured choosing an identifier number inside the global addressing range. In this case the identifier is written by the microcontroller into a dedicated register.

The bytes corresponding to the PDU type=40h and PDU_Length (respectively first and second byte of a frame) of the variables are interpreted by MICROFIP itself.

The variable structure must comply with some requirements, described in Subsection 2.3.

Note

The total memory capacity available for variables is 15 blocks of 8 bytes. These blocks can be allocated to the 8 variables as described in Subsection 2.3.

	Identifier	Type	Number of blocks*
Var 0	00xy	P	0 to 15
Var 1	01xy	C	0 to 15
Var 2	02xy	P	0 to 15
Var 3	03xy	C	0 to 15
Var 4	04xy	P	0 to 15
Var 5	05xy	C	0 to 15
Var 6	06xy	P	0 to 15
Var 7	07xy or xxxx	C	0 to 15

MPS variables used (Microcontrolled mode)

xy is the station number, coded hexa.

xxxx from the global addressing space.

P/C: Produced/Consumed

*one block = 8 bytes

Figure 2.2 – Microcontrolled mode MPS variables

2.2. Refreshment and Promptness

The variables are organized in two banks:

Bank	Produced variables	Consumed variables
A	Var 0, Var 2 and Var 4	Var 1, Var 3 and Var 7
B	Var 6	Var 5

The period variables always contain the asynchronous refreshment status, which is configured with 250 ms or infinite (always true).

The consumed variables are handled with the asynchronous promptness status. The promptness period value can be either 50 ms, 250 ms, 1 s or 5 s.

The consumed variables are updated in the dual port memory, only if the refreshment and significance status bits are set to true. A global status of variable validity is provided for each variable through the user interface.

Incoming variables which contain an incorrect refreshed status do not update the communication data base.

2.3. Variable Configuration (COBMPS)

The variable definition must comply with the following requirements:

- a variable can be configured with an integer number of blocks in the range [1–15]; the total number of blocks is limited as defined in the following clause,
- the number of data blocks available in one MICROFIP is 15 (120 bytes maximum),
- when additional MICROFIP components are used in ”cascade”, each additional MICROFIP provides 15 additional data blocks (120 additional bytes) for allocation. The total number of MICROFIP components used cannot exceed 4.

  In “cascade” mode:

  - the total number of variables (8) does not change, only the size does. The messaging size characteristics are not changed,
  - the size of one variable cannot exceed 120 bytes,
  - only one identification and one presence variable are issued,
- the consumed variable length is checked and must be equal to the configured length in order to declare the variable valid and to update the communication data base,
- the MPS status byte is automatically added to the number of data blocks declared for the produced variable,
- the MPS status byte received from the network is automatically removed from the consumed variable and processed by the MICROFIP chip. The result of this processing is given in Chapter 3, Subsection 4.3. (REGIS BANK –> MPSSA).

The configuration of the variables is performed in the dedicated COBMPS table, which can be accessed within the microcontroller addressing space dedicated to MICROFIP.

As an example, the topology of the dual port communication memory includes three variables:

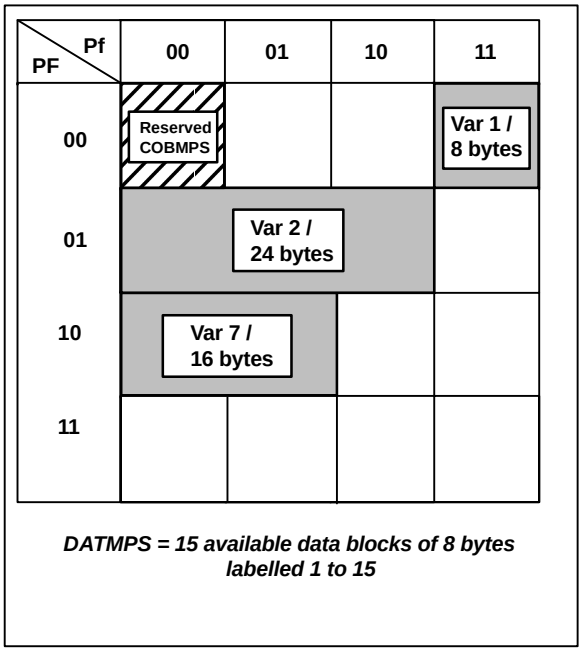


Figure 2.3 – Three variables of dual port communication memory

2.4. Variable Read and Write User Access Handling

When the user writes a produced variable value, the operation involves a local buffer to avoid having time constraints in relation to the network.

When the user reads a consumed variable value, the user gets the last valid buffer received from the network to avoid having time constraints in relation to the network.

2.5. Messaging Exchanges

Messages can be exchanged as follows:

- transmission can be triggered in the periodic or aperiodic way. In the aperiodic way, when the transmission queue is not empty, a request is attached to the Var6 variable (for example: transmission with request ID_DAT (06xy)/RP–DAT–MSG),
- reception is configured on the following LSAP [00xy to 0Fxy] for the segment number 0 or any segment number configured through the user interface.

The messages can be exchanged in non acknowledged and acknowledged mode. A broadcasting message is always non acknowledged.

Using the acknowledged mode for exchanges, the number of retries can be set to 0 or 1, through the user interface.

2.6. Message Contents

MICROFIP handles a full duplex transmit/receive channel with a storing capability of one message at a given time for each of the exchange directions.

A message frame with a maximum length of 128 bytes is composed of the following fields:

- a destination address of field 3 bytes,
- a source address field of 3 bytes,
- a user data field of 122 bytes maximum.

In transmission, MICROFIP handles the destination and source address fields and the user data field as a global information block. This message block is written by the user microcontroller as a whole for message transmission.

In reception, MICROFIP keeps a message based on its destination address, and it stores the message inside the internal dual port memory, including the source and destination address fields, and the user data field.

MICROFIP allows management of point–to–point and broadcasting communications based on the following addressing scheme:

	Messaging address format		
	LSAP number		**Segment number**
Point–to–point	00–0F	XY*	00 (and) 00–7F
Broadcasting	FF	FF	

* XY: 00 to FF subscriber number

Table 2.3 – Point–to–point and broadcast messaging address format

When receiving longer messages, an error status will be associated with the message reception indication.

2.7. Message Services

The message read and write operations are managed using control and status registers accessed through the user interface.

An indication of message reception or transmission can be given to the user, and managed either by polling or in interrupt mode.

The interrupt output pin IRQ0N is activated upon transmission or reception of a message and it can be masked. The interrupt cause can be analyzed by reading a dedicated interrupt status register.

2.8. Variable and Message Control – Interrupt Management

Each variable and each message channel is fitted with an interrupt enable/disable control bit (all bits are disabled after component initialization).

The interrupt output pin IRQ0N is activated if enabled upon:

- reception of the Var1, Var3, Var5 and Var7 consumed variables (the identifier of Var7 can be selected in the physical or global addressing range),
- transmission of Var6 produced variable,
- transmission or reception of a message,
- recognition of an IDDAT frame configured as the synchronization identifier.

Each interrupt cause can be individually masked. The interrupt cause can be analyzed by reading the dedicated interrupt status register.

All active bits of the interrupt status register read are cleared just after the read access.

2.9. Port Configuration

When MICROFIP is used in microcontrolled mode, the PIA and PIB ports are fully independent from the network activities.

Depending on the use of the `mf_redundancy` MICROFIP HANDLER primitive the PIA port is dedicated to the medium redundancy management or not. When the `mf_redundancy` primitive is not used, the PIA and PIB ports can be used by the external microcontroller, which can drive them as inputs or as outputs through the MICROFIP user interface block.

Port direction and inversion are controlled by the CPIA/B and INVA/B input pins.

The filtering option can be used when the microcontroller reads the input values of the ports.

Other options attached to port control in Standalone mode (Fall_back_values) are not relevant in this mode (see Subsection 1.5.).

3. MPS STATUS DEFINITION

When transmitting an MPS variable, the status byte (last transmitted) is formatted as follows:

MPS status bits	Contents
0 (last transmitted bit)	refreshment bit
1	0
2	meaning bit
3	0
4	0
5	0
6	0
7	0

Table 2.4 – Status byte format

When receiving a variable status byte, MICROFIP checks bits 0 and 2, respectively dedicated to the refreshment and significance information, and doesn't check for other bits.

4. PRESENCE VARIABLE

After power–up independent of the operating mode (standalone, or microcontrolled), MICROFIP is able to produce the SMMPS presence variable.

The frame contents dedicated to the MICROFIP embedded presence variable (14xy identifier number) is as follows:

Field name	Coding
1st (PDU type)	50h
2nd (Length)	05h
3rd	80h
4th	03h
5th	00h
6th	F0h
7th	00h

Table 2.5 – Presence variable frame content

5. IDENTIFICATION VARIABLE

After power–up initialization, and when MICROFIP is operating in **standalone mode**, it automatically senses the external pins which compose the `constructor` and `model` fields of the identification variable. This information is packaged into an 8–byte identification variable using identifier number 10xy.

When MICROFIP is operating in **microcontrolled mode**, the identification variable is filled after power–up or after a software reset by the user microcontroller, by writing the information contents located in a specific area of the MICROFIP internal RAM.

When the identification variable (identifier number 10xy) contents are filled, the user microcontroller enables identification variable production and the user can no longer modify the identification variable contents.

Byte	Field name	Coding Standalone mode	Coding Microcontrolled mode
1	PDU type	52h	52h
2	Length	08h	08h
3	WorldFIP profile	01	to be written by user
4	Class	SELDEM state in bit 0 (last transmitted) (other bits to zero)	to be written by user
5	Constructor (first byte)	Chip input pins	to be written by user
6	Constructor (second byte)	Chip input pins	to be written by user
7	Model (first byte)	Chip input pins	to be written by user
8	Model (second byte)	Chip input pins	to be written by user
9	Version	01	to be written by user
10	User	CPIA state on bit 0 CPIB state on bit 1 INVA state on bit 2 INVB state on bit 3 SELACT state on bit 4 (bits 5, 6, 7 = 0)	to be written by user

Table 2.6 –Identification variable frame content

In Standalone operating mode, the 32 bits to be updated in the identification variable are assigned by 16 dedicated MICROFIP input pins.

Each input pin allows 2 bits to be coded as follows:

Code	Input pin
00	grounded
01	connected to the RST1N output
10	connected to the RST2N output
11	connected to the positive supply

The `Constructor` and `Model` fields are associated with the corresponding input pins in accordance with the following table:

Field name	Coding input pin
Constructor[8]&[0]	CSN
Constructor[9]&[1]	EORDN
Constructor[10]&[2]	RWN
Constructor[11]&[3]	ALE
Constructor[12]&[4]	RES1
Constructor[13]&[5]	RES2
Constructor[14]&[6]	RES3
Constructor[15]&[7]	A[8]
Model[15:8]&[7:0]	AD[7:0]

Table 2.7 – `Constructor` and `Model` fields associated with the corresponding input pins

i.e.: with ESN = EORDN = RWN = ALE = RES1 = RES2 = A [8:0] = RST1N
the identification variable contents will be:

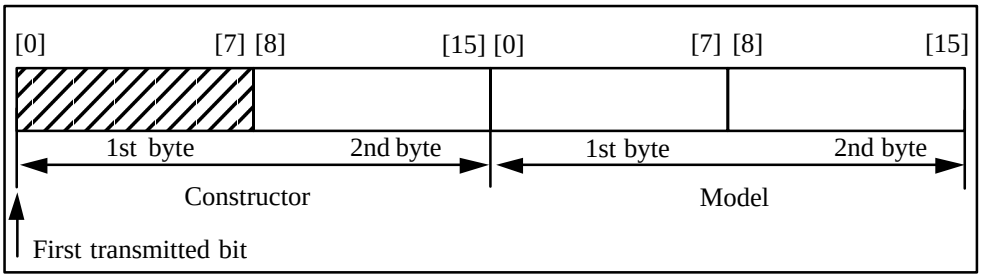


Figure 2.4 – Identification variable format

Chapter 3 Programming Interface Model Microcontrolled Mode

1. OVERALL INFORMATION

The microcontroller manages MICROFIP activities through a 512–byte data area, which is implemented as a dual port memory block between the user environment and the real–time network activities handled by MICROFIP.

Variables and message contents are stored inside this internal dual port memory area, and they are accessible through a set of control and status registers.

Any access is characterized by a fixed duration, which is independent of the network activities. This eliminates any need for an asynchronous access validation signal (DTACKN when using MOTOROLA processors).

To guarantee the external behaviour of the MICROFIP chip, it is strongly recommended to use the MICROFIP HANDLER software which implements the handling sequences to access the MICROFIP chip.

2. MEMORY MAP

The internal MICROFIP dual port memory which can be accessed by the external microcontroller is structured as follows:

MICROFIP overall mapping information		Address A [8:0]
MPS variable configuration area MPS variable contents	COBMPS DATMPS	000j to 007h 008h to 07Fh
Identification variable Configuration register bank REGIS Configuration register bank REGI2 Parallel port control PORTS	DATIDT REGIS REGI2 PORTS	080h to 087h 090h to 09Fh 0A0h to 0AFh 0B0h to 0BFh
Transmit messages	TXDMSG	100 to 17Fh
Receive messages	RXDMSG	180h to 1FFh

Table 3.1 – Memory structure

- **COBMPS and DATMPS: 8 bytes + 15 blocks of 8 bytes.**

  The COBMPS area contains the control information used to configure the variables and the DATMPS area is used to store MPS variable values.

  COBMPS accesses are also relevant for buffer validation when writing an MPS variable.

- **DATIDT: 8 bytes.**

  This area is used to store the 8 bytes which compose the identification variable. These bytes can be updated during the MICROFIP initialization sequence, and before the identification variable validation, which is performed through a writing operation in the REGIS area.

- **REGIS and REGI2 areas: 16 + 7 bytes.**

  These areas are composed of the control and status registers which are used to configure chip activity.

  The main control activities are the following:

  - initialization operations after power–up or on a specific software reset action,
  - variable exchange real–time handling (variable write and read),
  - message exchange real–time handling (message write and read),
  - event control, either in polling mode or by using an event–driven interrupt mechanism.

  The contents of these registers are described in detail in Subsection 3.2.

- **PORTS: 2 bytes.**

  The 2 bytes which compose the PORTS area are respectively attached to the PORTA and PORTB parallel ports.

  When a port is configured in output mode, the microcontroller controls its contents through write actions in the PORTS area, and can read back the written values by the corresponding read access.

  When a port is configured in input mode, the microcontroller gets its contents through the corresponding read action from the PORTS area.

- **TXDMSG and RXDMSG: 128 + 128 bytes.**

  The TXDMSG and RXDMSG areas respectively contain the written and received message contents.

  Each message size is limited to 128 bytes, including the 6 messaging address bytes.

3. CONTROL AND STATUS REGISTERS

3.1. Register mapping

3.1.1. Regis Bank

REGIS area		A[3:0]		BIT[7]	BIT[6]	BIT[5]	BIT[4]	BIT[3]	BIT[2]	BIT[1]	BIT[0]
Configuration register, A	W/R	0000	CONFA	SEL_ACK	NUM_ACK	STU_CHN [1]	STU_CHN [0]	EMST * *ENA_MST*	DMST *	EMSR * *ENA_MSR*	DMSR *
Configuration register, B	W/R	0001	CONFB	ENA_VAR [3]	ENA_VAR [2]	ENA_VAR [1]	ENA_VAR [0]	ENA_VAP [3]	ENA_VAP [2]	ENA_VAP [1]	ENA_VAP [0]
Configuration register, C	W/R	0010	CONFC	SEL_TRP [2]	SEL_TRP [1]	SEL_TRP [0]	SEL_COL	SEL_CEI		SEL_FRQ [1]	SEL_FRQ [0]
Configuration register, D	W/R	0011	CONFD	ENA_IDT	VEC_NUV [2]	VEC_NUV [1]	VEC_NUV [0]	SEL_TOT [2]	SEL_TOT [1]	SEL_TOT [0]	SEL_IDG
Configuration register, E	W/R	0100	CONFE	ENI_VAR [3]	ENI_VAR [2]	ENI_VAR [1]	ENI_VAR [0]	ENI_MST	ENI_MSR	ENI_SYN	ENI_VAP
Subscriber address register	W/R	0101	STADR	VEC_SAD [7]	VEC_SAD [6]	VEC_SAD [5]	VEC_SAD [4]	VEC_SAD [3]	VEC_SAD [2]	VEC_SAD [1]	VEC_SAD [0]
MPS status configuration register	W/R	0110	MPSPR	SEL_BPR [1]	SEL_BPR [0]	SEL_APR [1]	SEL_APR [0]	SEL_BRA	SEL_ARA	SEL_IFI [1]	SEL_IFI [0]
Messaging status register	R	0111	MSGSA	WDG_CH2	WDG_CH1	ERR_ART	OVF_MSR	ERR_MST	ERA_MST	EMP_MST	RDY_MSR
Interrupts status register	R	1000	IRQSA	STI_VAR [3]	STI_VAR [2]	STI_VAR [1]	STI_VAR [0]	STI_MSR	STI_MST	STI_SYN	STI_VAP
Transmitter messaging byte count reg.	R	1001	MSCOT	VEC_CPT [7]	VEC_CPT [6]	VEC_CPT [5]	VEC_CPT [4]	VEC_CPT [3]	VEC_CPT [2]	VEC_CPT [1]	VEC_CPT [0]
Receiver messaging byte count reg.	R	1010	MSCOR	VEC_RLG [7]	VEC_RLG [6]	VEC_RLG [5]	VEC_RLG [4]	VEC_RLG [3]	VEC_RLG [2]	VEC_RLG [1]	VEC_RLG [0]
MPS status register	R	1011	MPSSA	VEC_SVR [3]	VEC_SVR [2]	VEC_SVR [1]	VEC_SVR [0]	VEC_LVP [3]	VEC_LVP [2]	VEC_LVP [1]	VEC_LVP [0]
MPS global identifier Pf	W/R	1100	VIDGL	VEC_IGL [7]	VEC_IGL [6]	VEC_IGL [5]	VEC_IGL [4]	VEC_IGL [3]	VEC_IGL [2]	VEC_IGL [1]	VEC_IGL [0]
MPS global identifier PF	W/R	1101	VIDGH	VEC_IGH [7]	VEC_IGH [6]	VEC_IGH [5]	VEC_IGH [4]	VEC_IGH [3]	VEC_IGH [2]	VEC_IGH [1]	VEC_IGH [0]
MPS synchro identifier Pf	W/R	1110	SIDGL	VEC_ISL [7]	VEC_ISL [6]	VEC_ISL [5]	VEC_ISL [4]	VEC_ISL [3]	VEC_ISL [2]	VEC_ISL [1]	VEC_ISL [0]
MPS synchro identifier PF	W/R	1111	SIDGH	VEC_ISH [7]	VEC_ISH [6]	VEC_ISH [5]	VEC_ISH [4]	VEC_ISH [3]	VEC_ISH [2]	VEC_ISH [1]	VEC_ISH [0]

***Note: EMST, DMST, EMSR and DMSR from CONFA cannot be read: When reading CONFA, ENA_MST & ENA_MSR are read as bits 3 and 1.**

Table 3.2 – Regis bank

3.1.2. Regi2 Bank

REGI2 area		A[3:0]		BIT[7]	BIT[6]	BIT[5]	BIT[4]	BIT[3]	BIT[2]	BIT[1]	BIT[0]
Messaging segment register	W/R	0000	MSSEG	VEC_MYS [7]	VEC_MYS [6]	VEC_MYS [5]	VEC_MYS [4]	VEC_MYS [3]	VEC_MYS [2]	VEC_MYS [1]	VEC_MYS [0]
UART configuration register Pf	W/R	0001	ARCNL	VEC_ARL [7]	VEC_ARL [6]	VEC_ARL [5]	VEC_ARL [4]	VEC_ARL [3]	VEC_ARL [2]	VEC_ARL [1]	VEC_ARL [0]
UART configuration register PF	W/R	0010	ARCNH	VEC_ARH [7]	VEC_ARH [6]	VEC_ARH [5]	VEC_ARH [4]	VEC_ARH [3]	VEC_ARH [2]	VEC_ARH [1]	VEC_ARH [0]
Reset medium 1	W	1100	RESM1								
Reset medium 2	W	1101	RESM2								
Reset medium 1 & 2	W	1110	RES12								
Reset station	W	1111	RESET								

40 MHz CLK: 31.25 kbits vec_arl = 40h vec_arh= 04h/1 Mbit vec_arf = 02h vec_arh = 04h/2.5 Mbits vec_arf = 01h vec_arh = 00h /Reserved: vec_arf = 01h vec_arh = 10 h
20 MHz CLK: 31.25 kbits vec_arl = 20h vec_arh = 04h/1 Mbit vec_arf = 01h vec_arh = 04h/2.5 Mbits vec_arf = 01h vec_arh = 10 h

Table 3.3 – Regi2 bank

3.1.3. Port Bank

PORTS area		A[3:0]		BIT[7]	BIT[6]	BIT[5]	BIT[4]	BIT[3]	BIT[2]	BIT[1]	BIT[0]
Ports A	W/R	0XXX	PIA	PIA [7]	PIA [6]	PIA [5]	PIA [4]	PIA [3]	PIA [2]	PIA [1]	PIA [0]
Ports B	W/R	1XXX	PIB	PIB [7]	PIB [6]	PIB [5]	PIB [4]	PIB [3]	PIB [2]	PIB [1]	PIB [0]

Table 3.4 – Port bank

3.2. Register Contents

3.2.1. Regis Bank

CONFA (R/W) Configuration register, A: messaging control.

sel_ack	bit7 = 1	Messages are sent in acknowledged mode.
	bit7 = 0	Messages are sent in non–acknowledged mode.
num_ack	bit6 = 1	In acknowledged mode, messages are sent with one retry
	bit6 = 0	In acknowledged mode, messages are sent with no retry
ena_mst	bit3, bit2 = 00	No change
	bit3, bit2 = 01	Message transmission is disabled
	bit3, bit2 = 10	Message transmission is enabled
	bit3, bit2 = 11	Reserved
ena_msr	bit1, bit0 = 00	No change
	bit1, bit0 = 01	Message reception is disabled
	bit1, bit0 = 10	Message reception is enabled without broadcasting
	bit1, bit0 = 11	Message reception is enabled with broadcasting
bit5, bit4	bit5, bit4 = 00	No effect.
	bit5, bit4 = 01	After fault, medium is automatically restarted by the Presence variable
	bit5, bit4 = 10	After fault, medium is automatically restarted by the Presence variable or internal one–second timer
	bit5, bit4 = 11	After fault, medium is not restarted

CONFB (R/W) Configuration register, B: variables control.

ena_var[3:0]	bit7, bit6, bit5, bit4 = 1	Var7, Var5, Var3, Var1 enabled (consumed)
	bit7, bit6, bit5, bit4 = 0	Var7, Var5, Var3, Var1 disabled
ena_vap[3:0]	bit3, bit2, bit1, bit0 = 1	Var6, Var4, Var2, Var0 enabled (produced)
	bit3, bit2, bit1, bit0 = 0	Var6, Var4, Var2, Var0 disabled

CONFC (R/W) Configuration register, C: initialization control.

		RPTime	31.25 kbits/s	1 Mbit/s	2.5 Mbits/s
sel_trp[2:0]	bit7, bit6, bit5 =	000	320 µs	10 µs	4 µs
		001	640 µs	20 µs	8 µs
		010	960 µs	30 µs	12 µs
		011	1280 µs	40 µs	16 µs
		100	1600 µs	50 µs	20 µs
		101	1920 µs	60 µs	24 µs
		110	2240 µs	70 µs	28 µs
		111	64 µs	80 µs	32 µs

sel_col	bit4 = 1	CREOL Medium Attachment Unit is selected
	bit4 = 0	FIELDRIVE Medium Attachment Unit is selected.
sel_cei	bit3 = 1	WorldFIP/IEC 61158 compliant frame delimiters and FCS are selected
	bit3 = 0	FIP/NFC 46603 compliant frame delimiters and FCS are selected
sel_frq[1:0]	bit1, bit0 = 00	31.25 kbits/s operations are selected
	bit1, bit0 = 01	1 Mbit/s operations are selected
	bit1, bit0 = 10	2.5 Mbits/s operations are selected
	bit1, bit0 = 11	Reserved
bit2		Reserved use.

CONFD (R/W) Configuration register, D: initialization and variables control.

ena_idt	bit7 = 1	Identification variable is enabled. Station is started.
	bit7 = 0	Identification variable is disabled. Station is stopped.

vec_nuv[2:0]	bit6, bit5, bit4 =	000	variable Var0 is selected
		001	variable Var1 is selected
		010	variable Var2 is selected
		011	variable Var3 is selected
		100	variable Var4 is selected
		101	variable Var5 is selected
		110	variable Var6 is selected
		111	variable Var7 is selected

		Silence Time	31.25 kbits/s	1 Mbit/s	2.5 Mbits/s
sel_tot[2:0]	bit3, bit2, bit1 =	000	2880 µs	90 µs	36 µs
		001	3520 µs	110 µs	44 µs
		010	4160 µs	130 µs	52 µs
		011	4800 µs	150 µs	60 µs
		100	5440 us	170 µs	68 µs
		101	6080 µs	190 µs	76 µs
		110	6720 µs	210 µs	84 µs
		111	480 µs	250 µs	100 µs

sel_idg	bit0 = 1	Var7 is global and programmed on VIDGH/L registers
	bit0 = 0	Var7 identifier is 07xy (xy is the station address number)

CONFE (R/W) Configuration register, E: interrupt control.

eni_var[3:0]	bit7, bit6, bit5, bit4 = 1	Interrupts on Var7, Var5, Var3, Var1 enabled
	bit7, bit6, bit5, bit4 = 0	Interrupts on Var7, Var5, Var3, Var1 disabled
eni_mst	bit3 = 1	Interrupt on transmitter messaging enabled
	bit3 = 0	Interrupt on transmitter messaging disabled
eni_msr	bit2 = 1	Interrupt on receiver messaging enabled
	bit2 = 0	Interrupt on receiver messaging disabled
eni_syn	bit1 = 1	Interrupt on synchro identifier enabled
	bit1 = 0	Interrupt on synchro identifier disabled
eni_vap	bit0 = 1	Interrupt on Var6 enabled
	bit0 = 0	Interrupt on Var6 disabled

STADR (R/W) Subscriber address register: initialization control.

vec_sad[7:0]	bit7–>bit0	After reset the subscriber address is detected on SUBS[7:0]
	bit7–>bit0	Write into STADR changes to the subscriber address.

MPSPR (R/W) MPS status configuration register: initialization control.

sel_bpr[1:0]	bit7, bit6 = 00	Bank B promptness is 50 ms (Var5)
	bit7, bit6 = 01	Bank B promptness is 250 ms (Var5)
	bit7, bit6 = 10	Bank B promptness is 1 s (Var5)
	bit7, bit6 = 11	Bank B promptness is 5 s (Var5).
sel_apr[1:0]	bit5, bit4 = 00	Bank A promptness is 50 ms (Var1, Var3, Var7)
	bit5, bit4 = 01	Bank A promptness is 250 ms (Var1, Var3, Var7)
	bit5, bit4 = 10	Bank A promptness is 1 s (Var1, Var3, Var7)
	bit5, bit4 = 11	Bank A promptness is 5 s (Var1, Var3, Var7).
sel_bra	bit3 = 1	Bank B refreshment is infinite (Var6)
	bit3 = 0	Bank B refreshment is 250 ms (Var6)
sel_ara	bit2 = 1	Bank A refreshment is infinite (Var0, Var2, Var4)
	bit2 = 0	Bank A refreshment is 250 ms (Var0, Var2, Var4)

Note

With a refreshment period **sel_bra**, **sel_ara** selected at infinite, the refreshment status of the produced variable always will be TRUE.

sel_ifi[1:0]	bit1, bit0 = 00	Parallel input ports are not filtered
	bit1, bit0 = 01	Parallel input ports are filtered at 1 ms
	bit1, bit0 = 10	Parallel input ports are filtered at 4 ms
	bit1, bit0 = 11	Parallel input ports are filtered at 10 ms

MSGSA (R) **Messaging status register: global & messaging control.**

wdg_ch2	bit7 = 1	Watchdog on channel 2 is wrong (FIELDRIVE option only)
	bit7 = 0	Watchdog on channel 2 is idle (FIELDRIVE option only)
wdg_ch1	bit6 = 1	Watchdog on channel 1 is wrong (FIELDRIVE option only)
	bit6 = 0	Watchdog on channel 1 is idle (FIELDRIVE option only)
err_art	bit5 = 1	Error was traced on receiver activities (transmitted frame, Manchester error, FCS error ; bit number incorrect)
	bit5 = 0	No error was traced on receiver activities
	This bit is reset idle LOW when the MSGSA register is read.	
ovf_msr	bit4 = 1	Overflow error traced on messaging receiver activities
	bit4 = 0	No overflow error
	This bit is reset LOW when messaging receiver is disabled.	
	The error cause can be:	
		1/ A new message came and the buffer was not free (In this case the first message remains correct). RDY_MSR = 1 and OVF_MSR = 1 in this case. 2/ A message sized with more than 128 bytes has been received. (In this case the message stored must be ignored). RDY_MSR = 0 and OVF_MSR = 1 in this case.
err_mst	bit3 = 1	Error: lack of acknowledge; ack. messaging transmitter
	bit3 = 0	Acknowledge was received (ACK+ or ACK–) or messaging activity was performed non acknowledged.
	This bit is reset LOW when messaging transmitter is disabled.	
era_mst	bit2 = 1	Error: negative acknowledge; ack. messaging transmitter
	bit2 = 0	Positive Acknowledge was received (ACK+) or messaging activity was performed non acknowledged.
	This bit is reset LOW when messaging transmitter is disabled.	
emp_mst	bit1 = 1	Messaging transmitter buffer is empty
	bit1 = 0	Messaging transmitter buffer is occupied
	This bit is reset LOW when messaging transmitter activities are disabled.	
rdy_msr	bit0 = 1	Messaging receiver buffer contains a valid message
	bit0 = 0	Messaging receiver buffer contains no valid message
	This bit is reset LOW when messaging receiver is disabled.	

IRQSA (R) **Interrupt status register: interrupt control.**

sti_var[3:0]	bit7, bit6, bit5, bit4 = 1	Interrupt traced on Var7, Var5, Var3, Var1
	bit7, bit6, bit5, bit4 = 0	Interrupt idle on Var7, Var5, Var3, Var1
sti_msr	bit3 = 1	Interrupt traced on receiver messaging
	bit3 = 0	Interrupt idle on receiver messaging
sti_mst	bit2 = 1	Interrupt traced on transmitter messaging
	bit2 = 0	Interrupt idle on transmitter messaging
sti_syn	bit1 = 1	Interrupt traced on synchro identifier
	bit1 = 0	Interrupt idle on synchro identifier
sti_vap	bit0 = 1	Interrupt traced on Var6
	bit0 = 0	Interrupt idle on Var6

All bits are reset LOW when IRQSA register is read.

MSCOT (W) **Transmitter messaging bytes counter: messaging control.**

vec_cpt[7:0]	bit7–>bit0	Message byte number (valid range: 6 to 128)

MSCOR (R) **Receiver messaging bytes counter: messaging control.**

vec_rlg[7:0]	bit7–>bit0	Message byte number (valid range: 6 to 128)

MPSSA (R) **Variables status register: variable control.**

vec_svr[3:0] bit7, bit6, bit5, bit4 = 1 Var7, Var5, Var3, Var1 contents are valid
bit7, bit6, bit5, bit4 = 0 Var7, Var5, Var3, Var1 contents are not valid
A consumed variable is valid if the following conditions are satisfied:
1/ the station is started (ena_idt is ACTIVE)
2/ the variable is configured (COBMPS) and validated.
3/ the variable was received with:
a: a correct status from the UART
b: a PDU_TYPE = 40h
c: a number of bytes which fits the configuration
d: a valid MPS status (significance and refreshment)
4/ a valid promptness status

vec_svp[3:0] bit3, bit2, bit1, bit0 = 1 Var6, Var4, Var2, Var0 are temporarily locked
bit3, bit2, bit1, bit0 = 0 Var6, Var4, Var2, Var0 can be accessed
A produced variable may be locked if the last WRITE_LOC action has not yet been performed. If two consecutive WRITE_LOC are separated by more than one MPS transaction time (IDDAT + RPDATxxx) the variables can always been accessed.

VIDGL (W/R) Global variable identifier, Pf: variable control.

vec_igl[7:0]	bit7–>bit0	Global identifier, lower byte This register is significant only if sel_idg is ACTIVE.

VIDGH (W/R) Global variable identifier, PF: variable control.

vec_igh[7:0]	bit7–>bit0	Global identifier, higher byte This register is significant only if sel_idg is ACTIVE.

SIDGL (W/R) Synchronization identifier, Pf: variable & global control.

vec_isl[7:0]	bit7–>bit0	Synchronization identifier, lower byte This register is significant only if eni_syn is ACTIVE.

SIDGH (W/R) Synchronization identifier, PF: variable & global control.

vec_ish[7:0]	bit7–>bit0	Synchronization identifier, higher byte This register is significant only if eni_syn is ACTIVE.

3.2.2. Regi2 Bank

MSSEG (W/R) Messaging segment register: messaging control.

vec_mys[7:0]	bit7–>bit0	Messaging segment register. This register is relevant for receiver messaging activities only.

ARCNL (W/R) UART configuration register, Pf: initialization control

vec_arl[7:0]	bit7–>bit0	40 MHz clock, 31.25 kbits/s:	vec_arl = 40h
		20 MHz clock, 31.25 kbits/s:	vec_arl = 20h
		40 MHz clock, 1 Mbit/s:	vec_arl = 02h
		20 MHz clock, 1 Mbit/s:	vec_arl = 01h
		40 MHz clock, 2.5 Mbits/s:	vec_arl = 01h
		20 MHz clock, 2.5 Mbits/s:	vec_arl = 01h

ARCNH (W/R) UART configuration register, PF: initialization control

vec_arh[7:0]	bit7–>bit0	40 MHz clock, 31.25 kbits/s:	vec_arh = 04h
		20 MHz clock, 31.25 kbits/s:	vec_arh = 04h
		40 MHz clock, 1 Mbit/s:	vec_arh = 04h
		20 MHz clock, 1 Mbit/s:	vec_arh = 04h
		40 MHz clock, 2.5 Mbits/s:	vec_arh = 00h
		20 MHz clock, 2.5 Mbits/s:	vec_arh = 01h

RES M1 Reset medium 1
RES M2 Reset medium 2
RES M12 Reset media 1 and 2

RESET (W) **Reset station: initialization control**

Each write into this register reinitialises the station activities whatever the written contents are. Safe operations are ensured if at least 40 basic clock cycles are inserted between the write access into the RESET register and the next access to MICROFIP.

3.2.3. Ports Bank

PORTA (W/R) **PortA register: global control.**

pia[7:0] bit7–>bit0 When PORTA is an input port, a write access is not significant.
When PORTA is an output port, written values are output on PIA.

Read values are detected on PIA and filtered with period selected on sel_ifi.

Note

Port direction configuration: refer to Chapter 3, Subsection 3.2.

PORTB (W/R) **PortB register: global control.**

pib[7:0] bit7–>bit0 When PORTB is an input port, a write access is not significant.
When PORTB is an output port, written values are output on PIB.

Read values are detected on PIB and filtered with period selected on sel_ifi.

Note

Port direction configuration: refer to Chapter 3, Subsection 3.2.

4. FLOW CHARTS

The following handling sequence presentation is given as an illustration of what is implemented in the MICROFIP HANDLER software to be used to access the MICROFIP chip.

4.1. Initialization Sequence

In.	Operation	Data area	r/w	Data bus	Contents
1	software initialization	REGI2–> RESET	w	xxxxxxxx	not significant
2	CONFA initialization	REGIS–> CONFA	w	xc000110	sel_ack not significant num_ack user selection disable transmit messaging enable receive messaging
3	CONFB initialization	REGIS–> CONFB	w	11110000	consumed variables enabled produced variables disabled
4	CONFC initialization	REGIS–> CONFC	w	ccccc0cc	sel_trp user selection sel_col user selection sel_cei user selection sel_frq user selection
5	CONFD initialization	REGIS–> CONFD	w	0xxxcccc	ena_idt must be kept 0 vec_nuv not significant sel_tot user selection sel_idg user selection
6	CONFE initialization	REGIS–> CONFE	w	cccccccc	eni_var user selection eni_mst user selection eni_msr user selection eni_syn user selection eni_vap user selection
7	STADR initialization If subscriber unused number is detected on SUBS input lines. In this case, this subscriber number detected on SUBS input lines can be read from STADR register.	REGIS–> STADR	w	cccccccc	subscriber number
8	MPSP R initialization	REGIS–> MPSPR	w	cccccccc	sel_bpr user selection sel_apr user selection sel_bra user selection sel_ara user selection sel_ifi user selection
9	VIDGL & VIDGH initialization	REGIS–> VIDGL/H	w	cccccccc cccccccc	global identifier (significant if sel_idg = 1)
10	SIDGL & SIDGH initialization	REGIS–> SIDGL/H	w	cccccccc cccccccc	synchronization identifier (significant if eni_syn = 1)
11	MSSEG initialization	REGI2–> MSSEG	w	cccccccc	segment number
12	ARCNL/H initialization	REGI2–> ARCNL/H	w	cccccccc cccccccc	UART initialization (refer to Subsection.3.2.2.)

Table 3.5 – Initialization Sequence Flowchart

In.	Operation	Data area	r/w	Data bus	Contents
13	COBMPS initialization	COBMPS	8xw	8x cccccccc	MPS variable configuration table. COBMPS[i] <> Var[i] COBMPS[i], bits7:4: number of blocks in Var[i]: which must be from 1 to 8 (1 block = 8 bytes) COBMPS[i], bits3:0: First block for Var[i]: which must be from 1 to 15 If variable Var[i] is not handled, COBMPS[i] must be written with all bits at 0. (refer to Subsection 2.3. in Chapter 2)
14	DATIDT initialization	DATIDT	8x w	8x cccccccc	Identification variable contents (8 bytes) (refer to Section 5. in Chapter 2)
15	Station start	REGIS–> CONFD	w	1xxxcccc	sel_idt must be activated vec_nuv not significant sel_tot user selection sel_idg user selection

Table 3.5 – Initialization Sequence Flowchart (Cont'd)

4.2. Variable Write Sequence

The Variable accessed can be Var0, Var2, Var4 or Var6 = Var[i] with i = 0, 2, 4, 6.

In.	Operation	Data area	r/w	Data bus	Contents
1	Check for buffer availability	REGIS–> MPSSA	r	xxxxcccc	Access is allowed if: bit3 is 0 for Var6 access bit2 is 0 for Var4 access bit1 is 0 for Var2 access bit0 is 0 for Var0 access If access is not allowed: either return to step 1 (wait for buffer available) or exit the function.
2	Identify accessed variable	REGIS–> CONFD REGIS–> CONFD	r w	cxxxcccc cccccccc	Stores ena_idt, sel_tot, sel_idg. vec_nuv = i (variable number) and rewrite other bits.
3	Write bytes into DATMPS	DATMPS	nx w	cccccccc	Write the variable contents from the first byte of the first block (COBMPS config.)
4	Rewrite COBMPS in order to validate the written contents.	COBMPS	w	cccccccc	MPS variable configuration table. COBMPS[i] <> Var[i] COBMPS[i], bits7:4: number of blocks in Var[i]: which must be from 1 to 8 (1 block = 8 bytes) COBMPS[i], bits3:0: First block for Var[i]: which must be from 1 to 15
5	Enable accessed variable	REGIS–> CONFB REGIS–> CONFB	r w	1111cccc 1111cccc	Stores contents of other variables bit3 is 1 for Var6 enable bit2 is 1 for Var4 enable bit1 is 1 for Var2 enable bit0 is 1 for Var0 enable

Table 3.6 – Variable Write Sequence Flowchart

4.3. Variable Read Sequence

The Variable accessed can be Var1, Var3, Var5 or Var7 = Var[i] with i = 1, 3, 5, 7.

In.	Operation	Data area	r/w	Data bus	Contents
1	Check for variable status validity	REGIS–> MPSSA	r	xxxxxxxx	Variable is valid if: bit7 is 1 for Var7 access bit6 is 1 for Var5 access bit5 is 1 for Var3 access bit4 is 1 for Var1 access If variable is not valid: exit the function with error code
2	Disable accessed variable	REGIS–> CONFB REGIS–> CONFB	r w	cccccccc cccccccc	Stores contents of other var. bit7 is 0 for Var7 disable bit6 is 0 for Var5 disable bit5 is 0 for Var3 disable bit4 is 0 for Var1disable
3	Identify accessed variable	REGIS–> CONFD	r	cxxxcccc	Stores ena_idt, sel_tot, sel_idg. vec_nuv = i (variable number) and rewrite other bits.
4	Send bytes from DATMPS	DATMPS	nx r	cccccccc	Read the variable contents from the first byte of the first block (COBMPS config.)
5	Enable accessed variable	REGIS–> CONFB REGIS–> CONFB	r w	1111cccc cccccccc	Stores contents of other variables bit7 is 1 for Var7 enable bit6 is 1 for Var5 enable bit5 is 1 for Var3 enable bit4 is 1 for Var1 enable

Table 3.7 – Variable Read Sequence Flowchart

4.4. Message Write Sequence

In.	Operation	Data area	r/w	Data bus	Contents
1	Check for empty message queue	REGIS–> MSGSA	r	xxxxxxcx	Message queue is empty if: bit2 is 1 If message queue is not empty: exit the function with error code
2	Disable transmit messaging channel	REGIS–> CONFA REGIS–> CONFA	r w	xcxxxxxx cc000110	Stores contents of num_ack sel_ack user option
3	Write byte into TXDMSG	TXDMSG	nx w	cccccccc	Write the message contents from the first location of TXDMSG
4	Write message byte count	REGIS–> MSCOT	w	cccccccc	Byte count is 6 to 128.
5	Enable transmit messaging channel	REGIS–> CONFA REGIS–> CONFA	r w	xcxxxxxx cc000110	Stores contents of num_ack sel_ack user option

Table 3.8 – Message Write Sequence Flowchart

Completion of message transmission can be performed either by:

- polling the MSGSA status register,
- waiting for a HIGH level on the EMP_MST bit, or
- in interrupt mode, by an IRQ0N interruption associated with the STI_MST bit inside the IRQSA status register.

In any case, the transmission can be considered valid if the ERA_MST and ERR_MST inside the MSGSA status register are checked idle LOW.

4.5. Message Read Sequence

The message read sequence is entered when a valid message received indication was given, either by:

- polling on the MSGSA status register,
- waiting for a HIGH level on the RDY_MSR bit, or
- in interrupt mode, by an IRQ0N interruption associated with the STI_MSR bit inside the IRQSA status register.

In.	Operation	Data area	r/w	Data bus	Contents
1	Check for message ready	REGI2–> MSGSA	r	xxxxxxxx	If message not ready (bit0 = 0) memorize error code and go to step
2	Check for message size	REGI2–> MSCOR	r	cccccccc	Message size must be 6 to 128
3	Read bytes from RXDMSG	RXDMSG	nx r	cccccccc	Read the message contents from the first location of RXDMSG
4	Check for message overflow	REGI2–> MSGSA	r	xxxcxxxx	If overflow (bit4 = 1) memorize error code
5	Disable then enable receive messaging channel	REGIS–> CONFA REGIS–> CONFA REGIS–> CONFA REGIS–> CONFA	r w r w	ccxxxxxx cc000001 ccxxxxxx cc000010	Stores contents of sel_ack and num_ack Stores contents of sel_ack and num_ack Return, eventually with an error code trace on steps 1 and 4

Table 3.9 – Message Read Sequence Flowchart

4.6. Read Interrupt Vector Sequence

In.	Operation	Data area	r/w	Data bus	Contents
1	Provides the contents of IRQSA register	IRQSA–> IRQSA IRQSA–> IRQSA	w r	xxxxxxxx cccccccc	Not significant Contents of IRQSA (refer to Subsection 3.2.1.) All active bits are cleared after the read sequence

Table 3.10 – Read Interrupt Vector Sequence Flowchart

4.7. Read Event Status Sequence

In.	Operation	Data area	r/w	Data bus	Contents
1	Provides the contents of MSGSA register	MSGSA–> MSGSA	r	cccccccc	Contents of MSGSA (refer to Subsection 3.2.1.) Read doesn't affect the register contents

Table 3.11 – Read Event Status Sequence Flowchart

Chapter 4 Hardware Configuration Options

1. STANDALONE OPERATING MODE OPTIONS

1.1. Standalone Mode Selection

Standalone operating mode is selected when SLONE input pin is pulled HIGH.

In this mode the A[7:0] and AD[7:0] buses are configured as inputs, the AD[7:0] is used to configure the standalone mode, and the AD[7:0] is used to choose the identification variable contents.

1.2. Bit Rate Selection

This option is selected through the A[1:0] input pins.

A[1:0] pin	00	01	10	11
Bit rate	31.25 kbits/s	1 Mbit/s	2.5 Mbits/s	Reserved

Table 4.1 – Bit rate selection

Note

In microcontrolled operating mode this selection is performed through the user interface control registers.

1.3. Frame Delimiters and FCS Calculation

This option is selected through the A[2] input pin.

A[2] pin	0	1
Del and FCS	FIP/NFC	WorldFIP/IEC

Table 4.2 – Frame delimiters and FCS calculation

Note

In microcontrolled operating mode this selection is performed through the user interface control registers.

1.4. Response Time Selection

This option is selected through the A[6:4] input pins.

A[6:4] pins	31.25 kbits/s	1 Mbit/s	2.5 Mbits/s
000	320 µs	10 µs	4 µs
001	640 µs	20 µs	8 µs
010	960 µs	30 µs	12 µs
011	1280 µs	40 µs	16 µs
100	1600 µs	50 µs	20 µs
101	1920 µs	60 µs	24 µs
110	2240 µs	70 µs	28 µs
111	64 µs	80 µs	32 µs

Table 4.3 – Response time selection

Note

In microcontrolled operating mode this selection is performed through the user interface control registers.

1.5. Silence Time Selection

This option is selected through the EXSL[2:0] input pins.

EXSL[2:0] pins	31.25 kbits/s	1 Mbit/s	2.5 Mbits/s
000	280 µs	90 µs	36 µs
001	3520 µs	110 µs	44 µs
010	4160 µs	130 µs	52 µs
011	4800 µs	150 µs	60 µs
100	5440 µs	170 µs	68 µs
101	6080 µs	190 µs	76 µs
110	6720 µs	210 µs	84 µs
111	480 µs	250 µs	100 µs

Table 4.4 – Silence time selection

Note

In microcontrolled operating mode this selection is performed through the user interface control registers. We recommend grounding the EXSL[2:0] pins.

1.6. MAU Type Selection

This option is selected through the A[7] input pin.

A[7] pin	0	1
MAU type	FIELDRIVE	CREOL

Table 4.5 – MAU type selection

Note

In microcontrolled operating mode this selection is performed through the user interface control registers.

1.7. Starting Mode Selection

This option is selected through the SELDEM input pin.

	0	1
SELDEM	automatic start after power–up	remote controlled start from configuration variable

Table 4.6 – Starting mode selection

1.8. DEFSUP Input Selection

The refreshment status of produced variables is managed as follows depending on the DEFSUP input.

	0	1
DEFSUP	Refreshment status depends on write update cycle	Resfreshment status always FALSE

Table 4.7 – DEFSUP input selection

2. MICROCONTROLLED OPERATING MODE OPTIONS

2.1. Microcontrolled Mode Selection

The microcontrolled operating mode is selected when SLONE input pin is pulled LOW.

2.2. Microcontroller Interface Selection

This mode is selected through the MOD[1:0] input pins.

MOD[1:0]	**00**	**01**	**10**	**11**
microcontroller interface	INTEL multiplexed	INTEL demultiplexed	MOTOROLA multiplexed	MOTOROLA demultiplexed

Table 4.8 – Microcontroller interface selection

2.3. Cascaded Mode Selection

This option is selected through the SELDEM input pin.

	0	**1**
SELDEM	normal mode (first MICROFIP)	cascaded mode (complementary MICROFIPs)

Table 4.9 – Cascaded mode selection

2.4. DEFSUP Input Selection

The refreshment status of produced variables is managed as follows depending on the DEFSUP input.

	0	**1**
DEFSUP	Refreshment status depends on write update cycle	Resfreshment status always FALSE

Table 4.10 – DEFSUP input selection

3. GLOBAL OPTIONS

3.1. Monomedium Operation Selection

When MICROFIP has to be used in monomedium, RXA2X, TER2X and WTC2X must be driven as follows:

	Level
RXA2X	inactive level
TER2X	active level
WTC2X	inactive level

which means:

- LOW in CREOL coupling mode,
- HIGH in FIELDRIVE coupling mode.

3.2. Parallel input/output Ports Control

	0	**1**
CPIA/CPIB	Parallel port A/B in output mode	Parallel port A/B in input mode
INVA/INVB	No logic inversion port A/B / variable	logic inversion port A/B / variable
SELACT	Output port signals are driven open_drain_N	Output port signals are driven push–pull
	This concerns the following output and bi–directionnal signals:	
	PIA (7:0) (pull–up required) PIB (7:0) (pull–up required) TXD (pull–down 1 kΩ) TXE1 (pull–down 1 kΩ) TXE2 (pull–down 1 kΩ)	PIA (7:0) k PIB (7:0) TXD TXE1 TXE2

Table 4.11 – Parallel input/output port control

3.3. Basic Clock Selection

This option is selected through the TST[2:0] input pins.

	000	**101**
TST[2:0]	40 MHz basic clock	20 MHz basic clock

Table 4.12 – Basic clock selection

Other configurations on TST[2:0] input pins are reserved for testing.

4. CASCADE MODE

Several MICROFIP chips could be used in parallel to extend the number of memory blocks available to be allocated on variables.

Main principles:

- Only one of the MICROFIPs will be unchanged:
  - to produce the presence and identification variable, and
  - to manage the messaging. It will set the SELDEM input drive to LOW,
- up to three other MICROFIPs could be installed in parallel with the SELDEM input drive to HIGH,
- all the MICROFIP chips in parallel have the same subscriber number,
- the total number of variables remains eight (four produced, four consumed) and each one is managed by one and only one of the MICROFIP chips.

To interface the transceiver, the following rules must be followed:

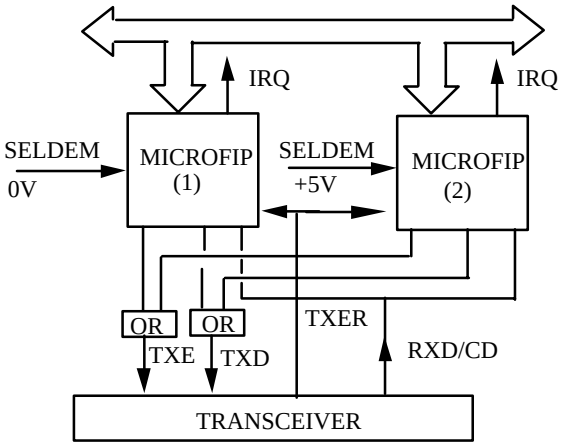


Figure 4.1 – Transceiver interface

Chapter 5 User Microcontroller Interface

In standalone operating mode this chapter is not used.

1. MICROCONTROLLER SELECTION

The microcontroller type selection is performed during the initialization phase (RSTIN active LOW) detecting the values on MOD[1:0] pins:

- MOD[1:0] = 00 80C51 compatible interface.
- MOD[1:0] = 01 80C196 compatible interface.
- MOD[1:0] = 10 68HC11 compatible interface, multiplexed bus mode.
- MOD[1:0] = 11 68HC11 compatible interface, demultiplexed bus mode.

2. FUNCTIONALITY

The user microcontroller can access the following resources in MICROFIP:

- a write and read register bank is used for the MICROFIP chip to control tasks.
- a dual port memory contains the communication data base. The variables are located inside the DATMPS area and the message contents are located inside the DATMSG area.

The COBMPS area of the dual port memory is used for the MPS variable configuration task.

The user access cycles are handled in fixed duration mode. This eliminates any need for data acknowledge signal management (no DTACKN signal).

When using microcontrollers with a multiplexed bus interface, address and data information are exchanged on the same AD[7:0] bus lines.

For external peripherals, MICROFIP provides the demultiplexed and latched address lines for every processor access (not only MICROFIP ones) on the A[7:0], configured as outputs in this mode.

When using processors with a demultiplexed bus interface, the A[7:0] lines support address information and are configured as inputs. In this case the AD[7:0] lines only support data.

3. TIMING

3.1. Initialisation

After reset (RSTIN –> HIGH), 40 CLK cycles must be elapsed before internal logic becomes operational.

3.2. Intel multiplexed bus – write cycle

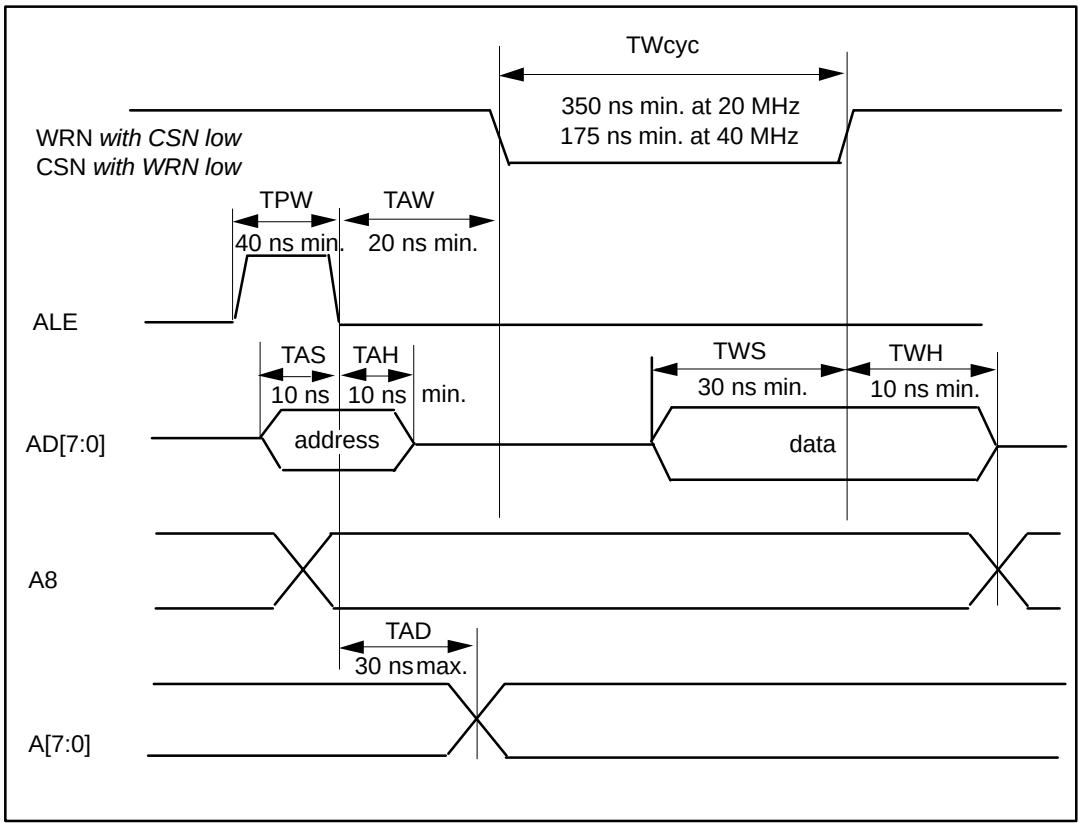


Figure 5.1 – Intel multiplexed bus – write cycle

3.3. Intel demultiplexed bus – write cycle

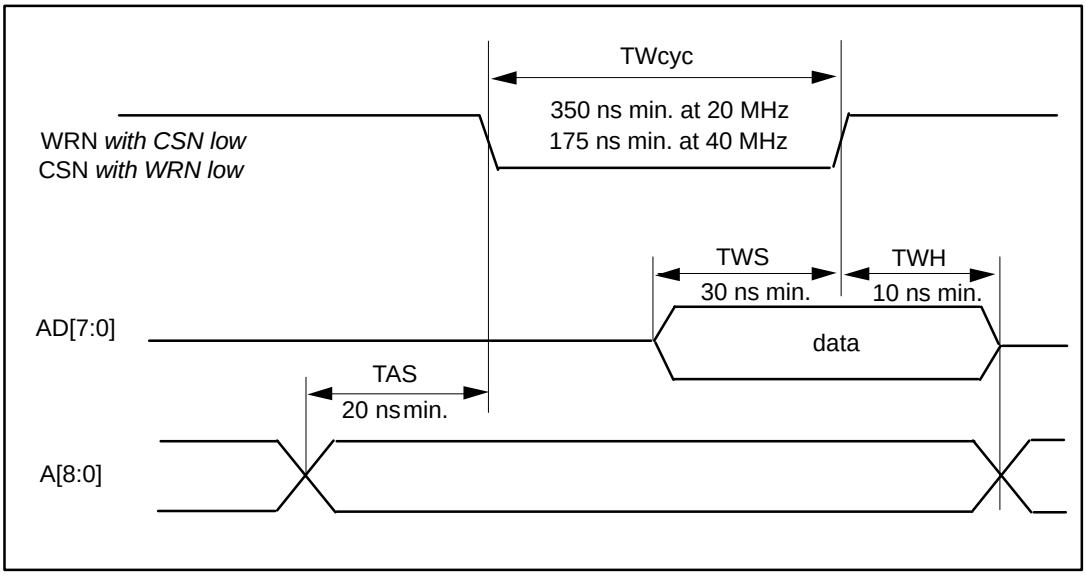


Figure 5.2 – Intel demultiplexed bus – write cycle

3.4. Intel multiplexed bus – read cycle

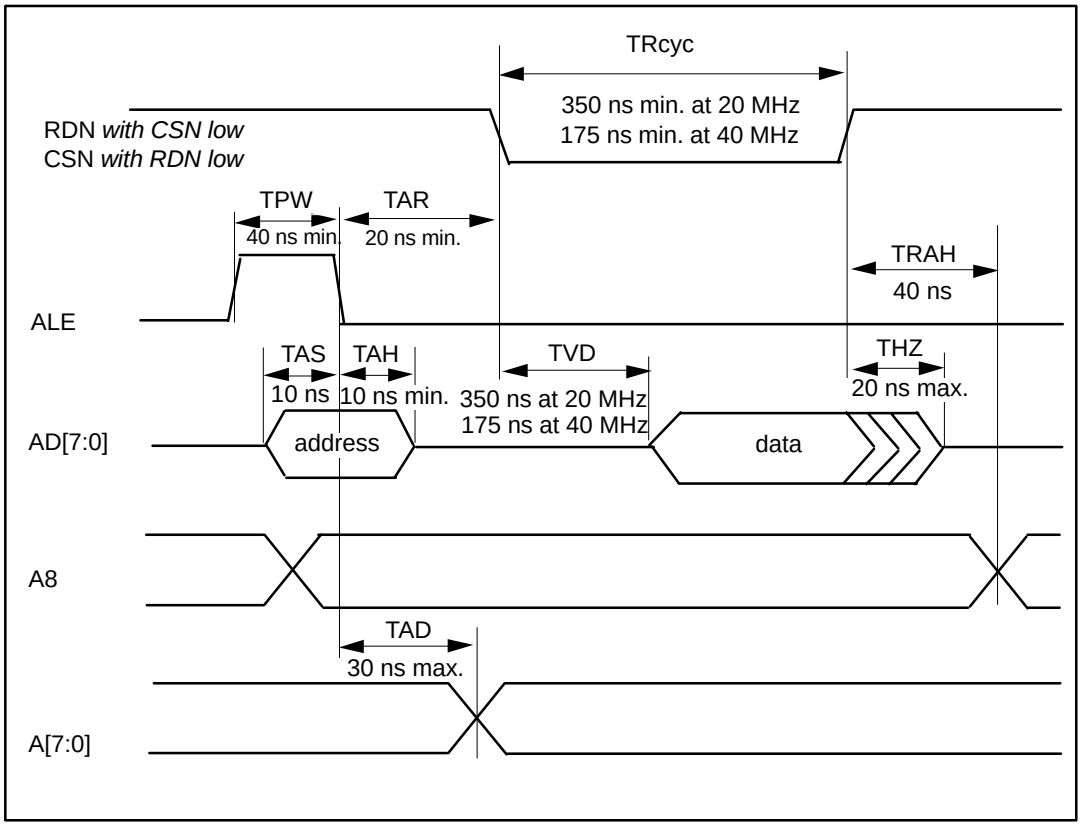


Figure 5.3 – Intel multiplexed bus – read cycle

3.5. Intel demultiplexed bus – read cycle

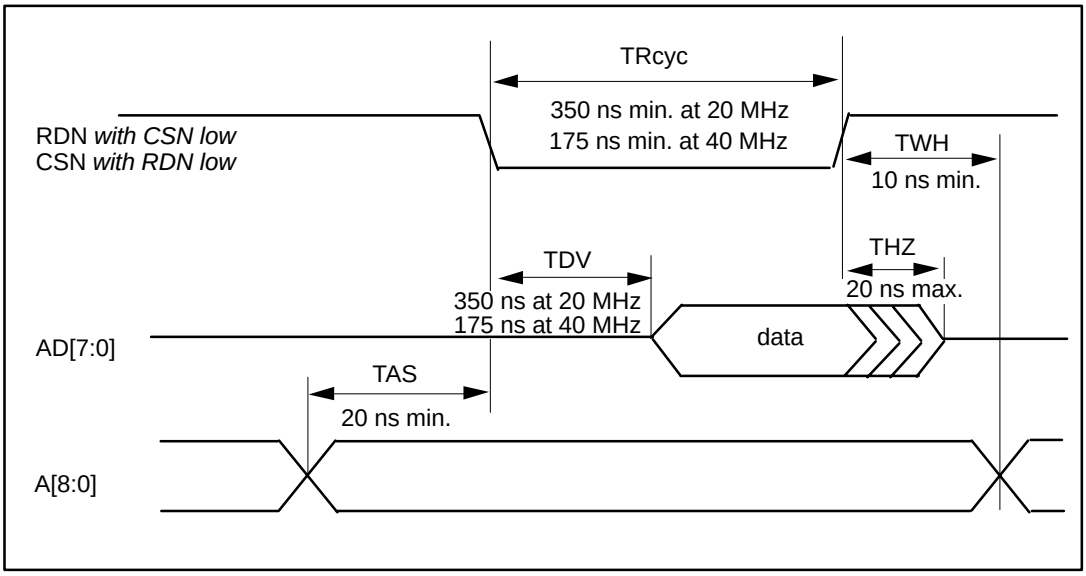


Figure 5.4 – Intel demultiplexed bus – read cycle

3.6. Motorola write and read accesses – multiplexed buses

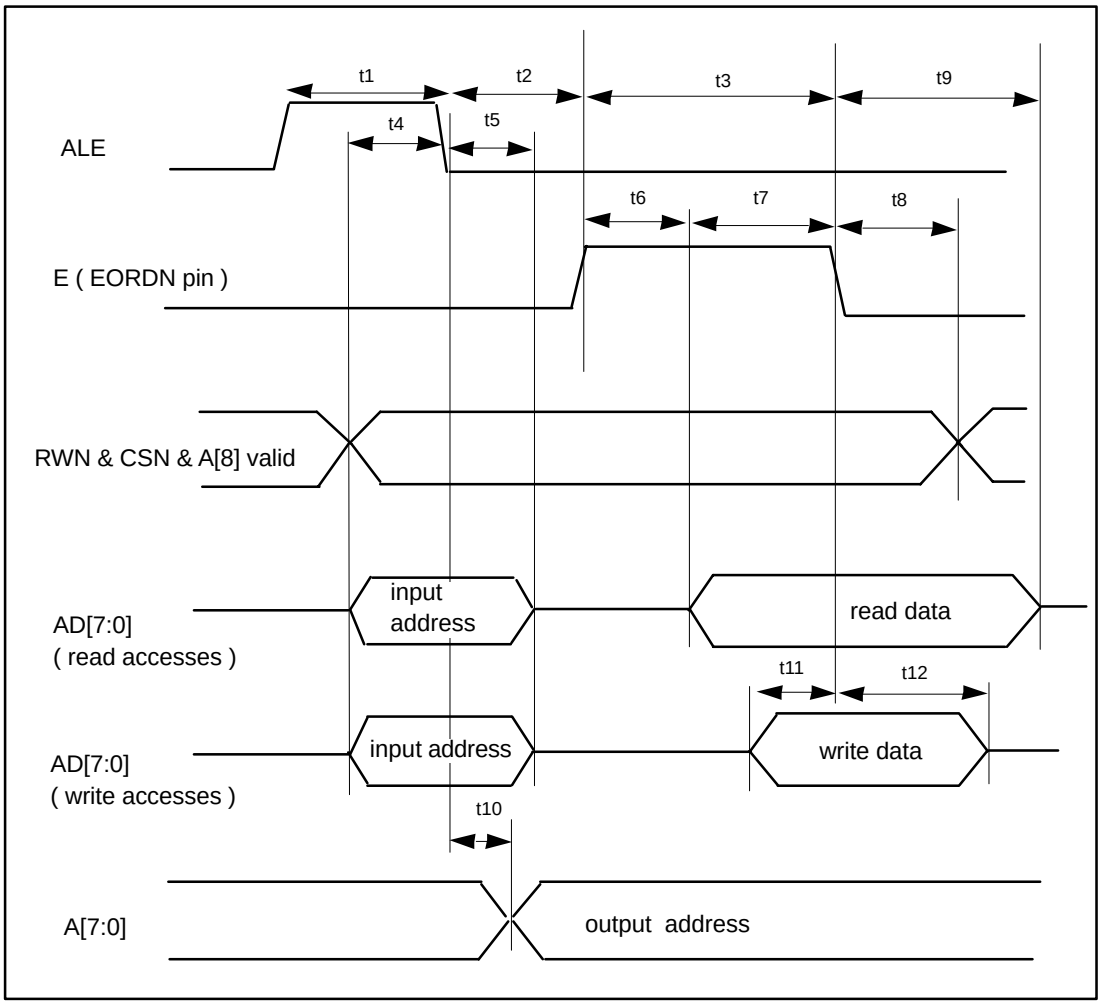


Figure 5.5 – Motorola write and read accesses – multiplexed buses

3.7. Motorola write and read accesses – demultiplexed buses

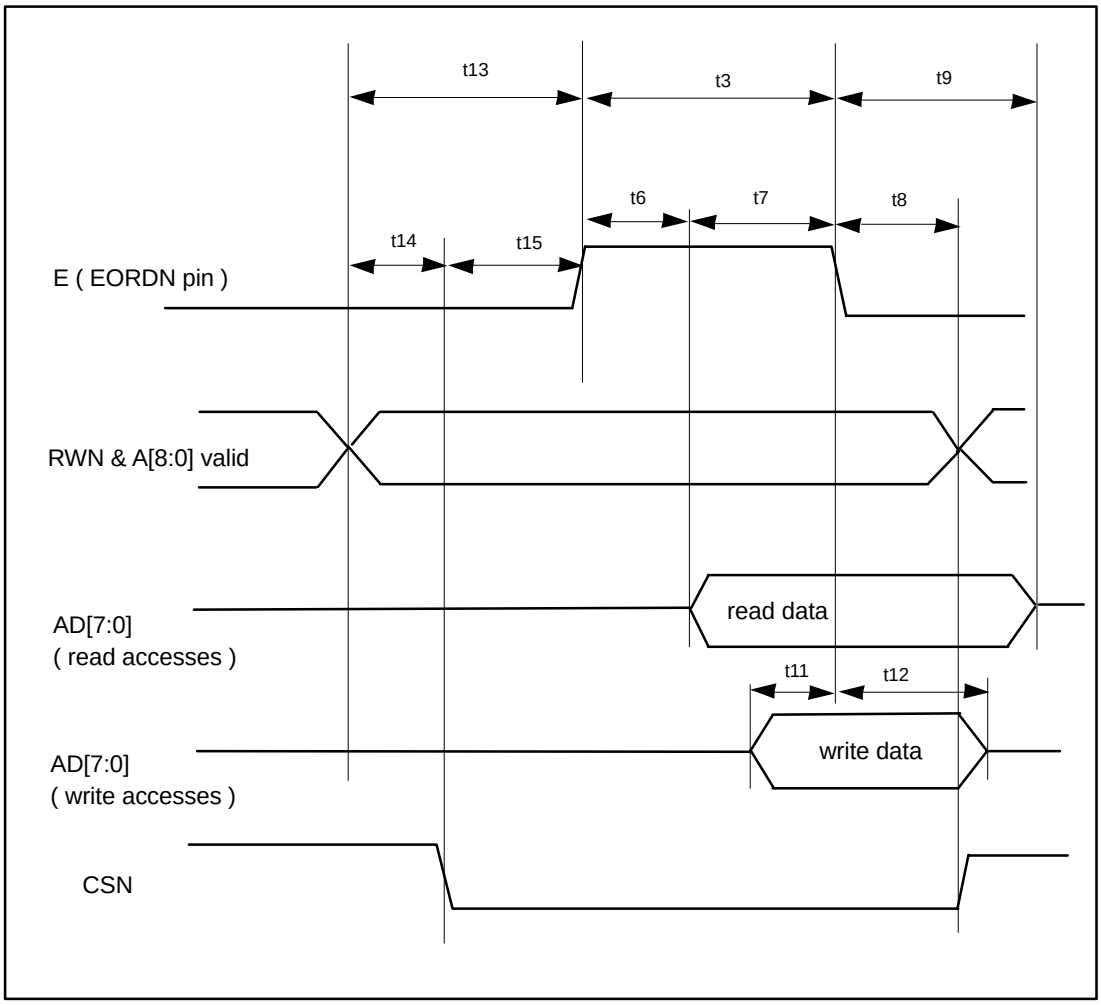


Figure 5.6 – Motorola write and read accesses – demultiplexed buses

3.8. Parameter Definition

Symbol	Description	Minimum	Maximum
t1	ALE address latch enable signal width	40 ns	
t2	Address latch to data cycle set–up time	20 ns	
t3	Data cycle selection width: 20 MHz clock 40 MHz clock	 350 ns 175 ns	
t4	Control & address information set–up time	10 ns	
t5	Address information hold time	10 ns	
t6	Read information access time: 20 MHz clock 40 MHz clock		 350 ns 175 ns
t7	AD[7:0] End of selection to high impedance state		20 ns
t8	Control information hold time from end of selection	10 ns	
t9	Read information hold time from end of selection	0 ns	
t10	Address output delay time from address latch		30 ns
t11	Write data set up time to end of selection	30 ns	
t12	Write data hold time from end of selection	10 ns	
t13	Control information set–up time to data cycle	30 ns	
t14	Control information set–up time to chip selection	0 ns	
t15	Chip selection set–up time to data cycle	0 ns	

Table 5.1 – Microcontroller Parameter Definition

Notes

1. Output load on AD[7:0], A[7:0]: 50 pF.
2. Unless otherwise specified, all timings are compatible for 3.3 V and 5 V operations, and for 20 MHz and 40 MHz basic operating frequencies.
3. When the 3.3 V nominal supply voltage is used, the 20 MHz basic clock option must be selected.

Chapter 6 Medium Attachment Unit Interface

1. INTERFACE MODE SELECTION (FIELDRIVE OR CREOL)

MAU coupling interface mode selection is performed through the user interface control registers when MICROFIP is run in microcontrolled mode, and through a dedicated input pin when MICROFIP is run in standalone mode.

2. INTERFACE SIGNAL DEFINITION

The line transceiver interface is composed as follows:

- Six signals are duplicated for each medium:

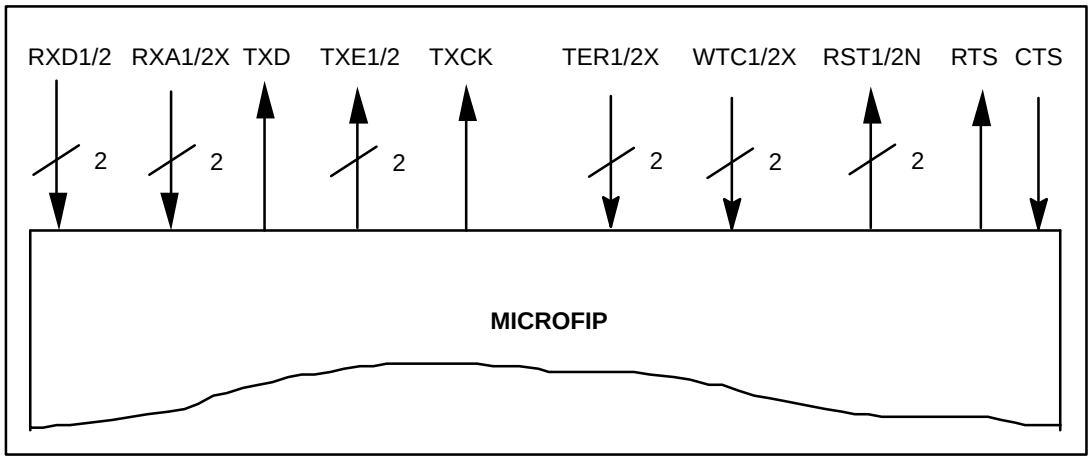
Signal	Function	Note
RXD1/2	Received data. Serial input from the line driver.	When idle (outside frames) this input can either be HIGH or LOW.
RXA1/2X	Receiver activity detection.	This input from the line driver is active: - LOW in FIELDRIVE mode, and - HIGH in CREOL mode. The RXA1/2X is used in MICROFIP: - to validate RXD1/2 input, - to detect no echo when transmitting. At end of frame, the RXA1/2X signal may still remain active for 4.5 bit periods.
TXE1/2	Transmitter enable. Output to the line driver, active HIGH.	
TER1/2X	Transmitter error. This input takes into account during frame transmissions the status errors from the line drivers.	This error input from the line drivers is active: - HIGH in FIELDRIVE mode, - LOW in CREOL mode.
WTC1/2X	Watching of transmission consumption on CREOL, transmission watchdog on FILEDRIVE.	This input is active: - LOW in FIELDRIVE, - HIGH in CREOL.
RST1/2N	Line driver initialization control. Output to the line driver, active LOW.	When MICROFIP is used in FIELDRIVE mode, the RST1/2N allows FIELDRIVE to be reinitialized after a watchdog error is indicated.

- two signals are common:

Signal	Function	Note
TXCK	Clock output at twice the transmission frequency (for example: 2 MHz at 1 Mbit/s).	
TXD	Transmitted data. Serial output to the line driver.	

- two signals, which are of optional use, allow the frame transmission starting sequences to be run when handshaking is used:

Signal	Function	Note
RTS	Request to Send.	This output, active HIGH, is activated in order to request a frame transmission.
CTS	Clear to Send.	This input, active HIGH, acknowledges the request for frame transmission. When the handshaking control option is not implemented, CTS can be kept permanently active HIGH.



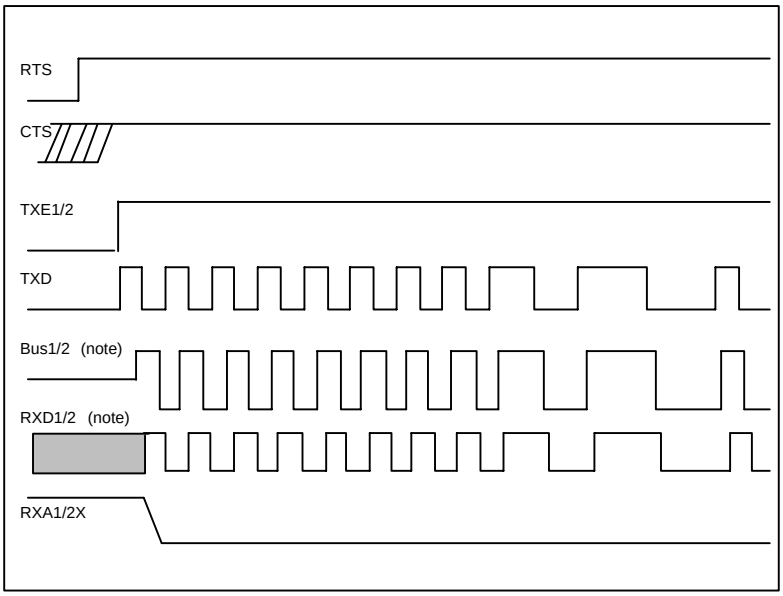
Note

Output load on RTS, TXD, TXE1/2, TXCK: 50 pF.

3. INTERFACE SIGNAL TIMINGS

3.1. Start of Frame – Waveforms

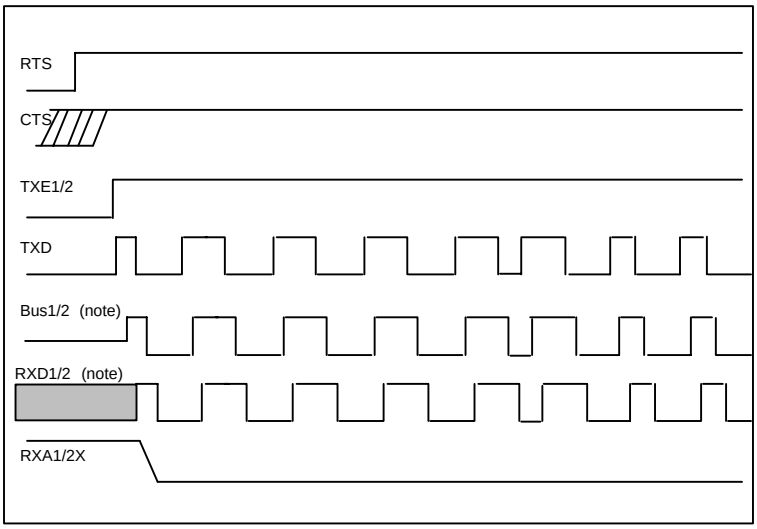
3.1.1. FIP/NFC physical layer configuration



Notes

1. The behaviour of the bus signal and the RXD signal depends on the dynamic response of the line transceiver used.
2. The waveform drawn corresponds to the FIELDRIVE coupling mode. In CREOL coupling mode, RXA1/2X polarity is inverted (active HIGH).

3.1.2. WorldFIP physical layer configuration

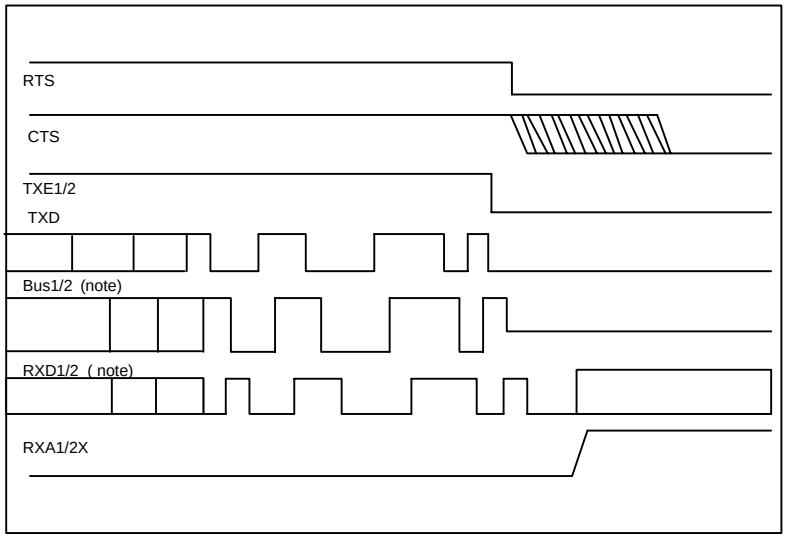


Notes

1. The behaviour of the bus signal and the RXD signal depends on the dynamic response of the line transceiver used.
2. The waveform drawn corresponds to the FIELDRIVE coupling mode. In CREOL coupling mode, RXA1/2X polarity is inverted (active HIGH).

3.2. End of Frame – Waveforms

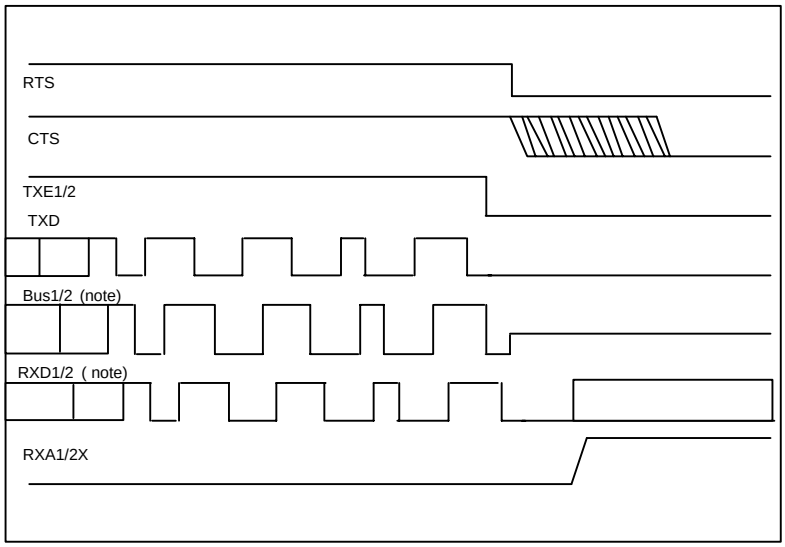
3.2.1. FIP/NFC physical layer configuration



Notes

1. The behaviour of the bus signal and the RXD signal depends on the dynamic response of the line transceiver used.
2. The waveform drawn corresponds to the FIELDRIVE coupling mode. In CREOL coupling mode, RXA1/2X polarity is inverted (signal active HIGH).

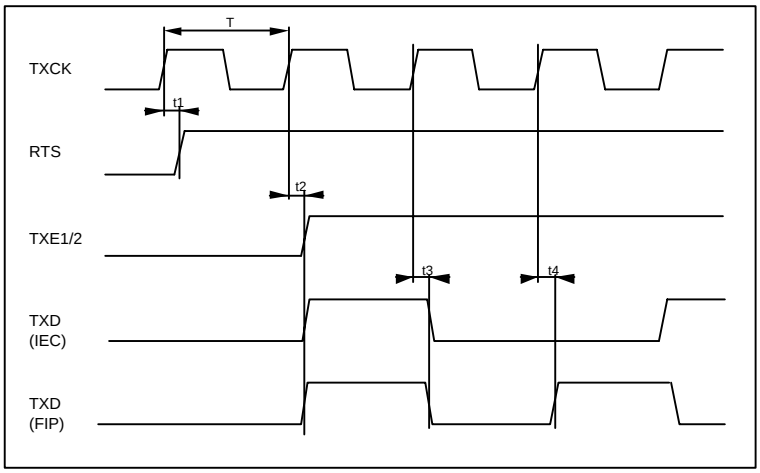
3.2.2. WorldFIP/IEC physical layer configuration



Notes

1. The behaviour of the bus signal and the RXD signal depends on the dynamic response of the line transceiver used.
2. The waveform drawn corresponds to the FIELDRIVE coupling mode. In CREOL coupling mode, RXA1/2X polarity is inverted (signal active HIGH).

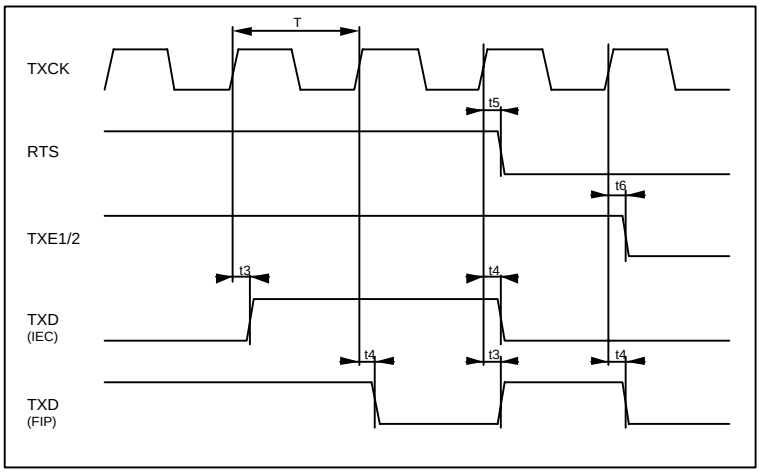
3.3. Start of Frame – Timing diagram



Note

T = 16 µs @ Network bit rate = 31.25 kbits/s
500 ns @ Network bit rate = 1 Mbit/s
200 ns @ Network bit rate = 2.5 Mbits/s

3.4. End of Frame – Timing diagram



Note

T = 16 µs @ Network bit rate = 31.25 kbits/s
500 ns @ Network bit rate = 1 Mbit/s
200 ns @ Network bit rate = 2.5 Mbits/s

Symbol	Description	Min	Max
t1	TXCK high to RTS high	– 20 ns	60 ns
t2	TXCK high to TXE1/2 high	– 20 ns	60 ns
t3	TXCK high to TXD high	– 20 ns	60 ns
t4	TXCK high to TXD low	– 20 ns	60 ns
t5	TXCK high to RTS low	– 20 ns	60 ns
t6	TXCK high to TXE1/2 low	– 20 ns	60 ns

Notes

1. Output local on TRS, TXD, TXE1/2, TXCK: 50 pF.
2. TXCK polarity is drawn according to the CREOL coupling option and is inverted when the FIELDRIVE option is selected.

4. MICROFIP RECEIVER

In the receiver section the input RXA1/2X identifies the frame envelope. When RXA1/2X is inactive the serial manchester input RXD is ignored. Moreover this carrier detection signal is used for the sequence checking of the frames in order to determine the bus activity.

A received frame is composed of consecutive symbols:

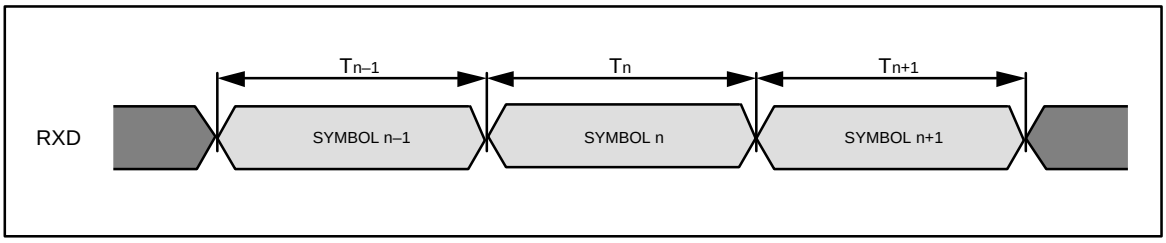


Figure 6.1 – Received frame composition

Typical duration and shape of a received symbol shall be in accordance with the IEC and FIP communication protocol standards.

Temporal distortion of the 'n' symbol received at the RXD input is calculated as follows:

$$\textit{Distortion (\%)} = 100 * | T_n - T_{ideal} | / T_{ideal}$$

where T_n : duration of the 'n' symbol received

T_{ideal} : duration of the ideal symbol.

For the MICROFIP component:

- accepted temporal distortion shall not be greater than 15%, and
- accepted jitter on 48 consecutive received bits shall not be greater than 1% whatever the network bit rate.

Chapter 7 Characteristics

1. ABSOLUTE MAXIMUM RATINGS

Table 7.1 gives MICROFIP technical characteristics.

Characteristic	Value
Storage temperature range	–40 to +125°C
Absolute supply voltage range	–0.3 V to 6.0 V
Absolute voltage range on input pins	–0.3 V to VDD + 0.3 V
Operating frequency	20 MHz or 40 MHz 100 ppm max.

Table 7.1 – Technical characteristics

2. RECOMMENDED OPERATING CONDITIONS

Table 7.2 gives MICROFIP recommended operating conditions.

Description	Minimum	Nominal	Maximum
Operating temperature range	– 40°C		+ 85°C
Supply voltage, classical mode	4.5 V	5 V	5.5 V
Supply voltage, low voltage mode	3 V	3.3 V	3.6 V

Table 7.2 – Recommended operating conditons

Table 7.3 gives MICROFIP characteristics with a 5 V classical or a 3.3 V low supply voltage.

Characteristics		**Classical supply voltage (5 volts nominal)**		**Low supply voltage (3.3 volts nominal)**	
Symbol	**Description**	**Minimum**	**Maximum**	**Minimum**	**Maximum**
VIH	Input HIGH voltage	0.7 x VDD	VDD + 0.3 V	0.7 x VDD	VDD + 0.3 V
VIL	Input LOW voltage	– 0.3 V	0.3 X VDD	– 0.3 V	0.2 X VDD
VOH	Output high voltage: IOH = 100 µA IOH = 2 mA	2.4 V	VDD – 0.3 V	2.4 V	VDD – 0.3 V
VOL	Output low voltage		0.4 V		0.4 V
In, Out, IO capacitance	Pad capacitance excluding package		4 pF		4 pF
In and IOz leakage current		– 10 µA	+ 10 µA	– 10 µA	+ 10 µA

Table 7.3 – Characteristics with a classical or low supply voltage

3. CONSUMPTION

Table 7.4 gives MICROFIP maximum consumption.

ICC supply current	**Maximum consumption**
3.3 V and 20 MHz basic clock	10 mA
5 V and 20 MHz basic clock	15 mA
5 V and 40 MHz basic clock	30 mA

Table 7.4 – Maximum consumption

4. BASIC CLOCK GENERATION

When MICROFIP is operating at 20–MHz clock frequency, the clock can be generated by using either:

- an external oscillator with a CMOS compatible output, or
- an external quartz.

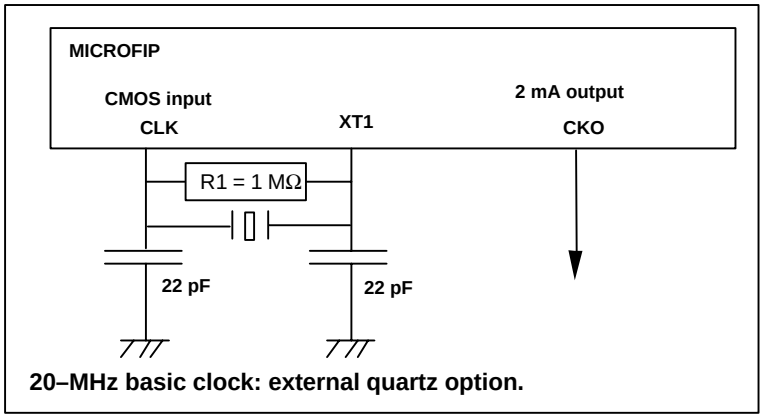


Figure 7.1 – Basic 20–MHz clock

When MICROFIP is operating with a 40–MHz clock frequency, the clock must be provided by using an external oscillator.

Chapter 8 Application Diagrams

Application diagram	See ...
Redundancy Management with `mf_redundancy` compiling option selected	Figure 8.1
Intel 8051 with external memory	Figure 8.2
MICROFIP used with Intel 8051 (`mf_redundancy` compiling option selected)	Figure 8.3
MICROFIP used with Intel 8051 (without `mf_redundancy` compiling option selected)	Figure 8.4
MICROFIP used in standalone	Figure 8.5
FIELDRIVE line transceiver (three speeds and redundancy options)	Figure 8.6

For details on redundancy management, see *ALS 50202 MICROFIP HANDLER User Reference Manual*.

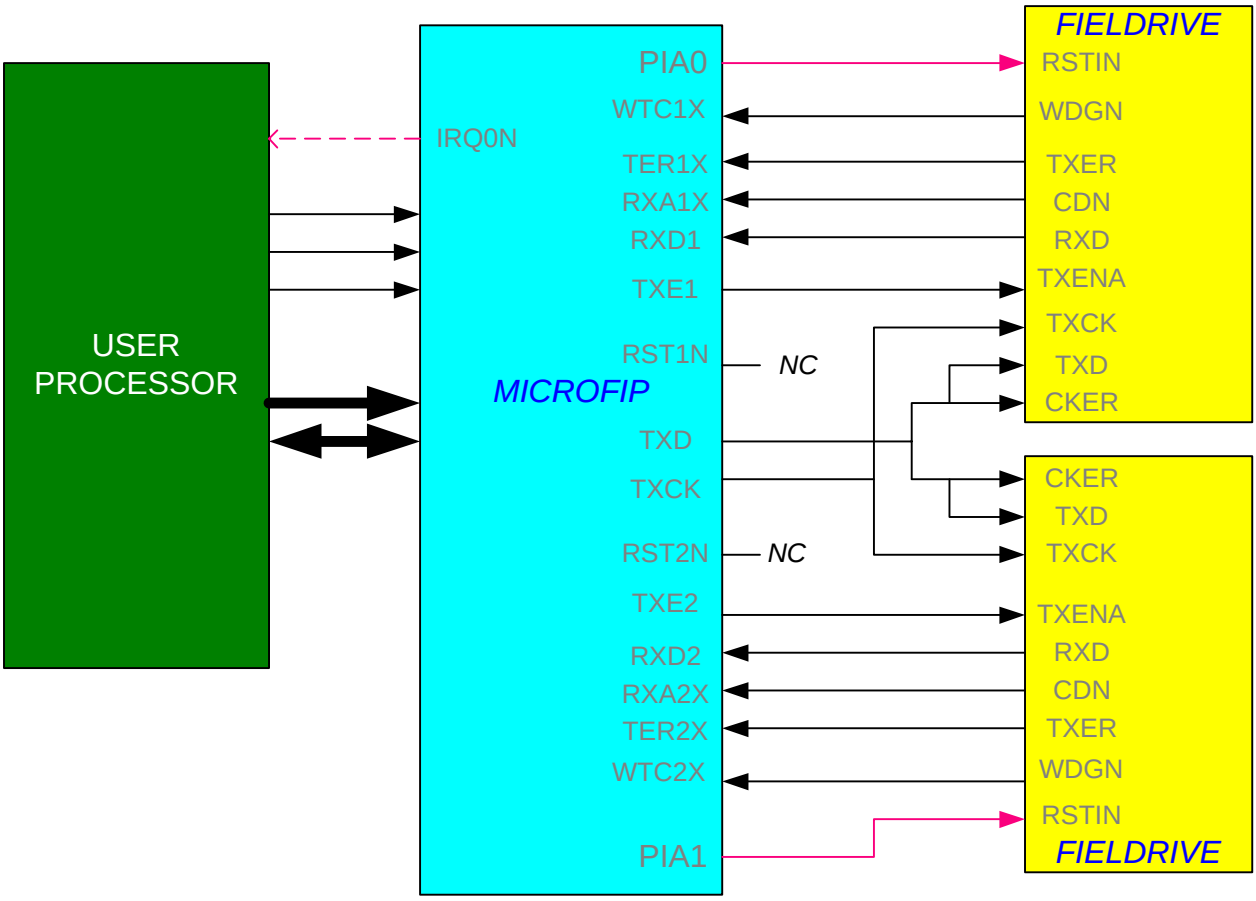


Figure 8.1 – Redundancy management with `mf_redundancy` compiling option selected

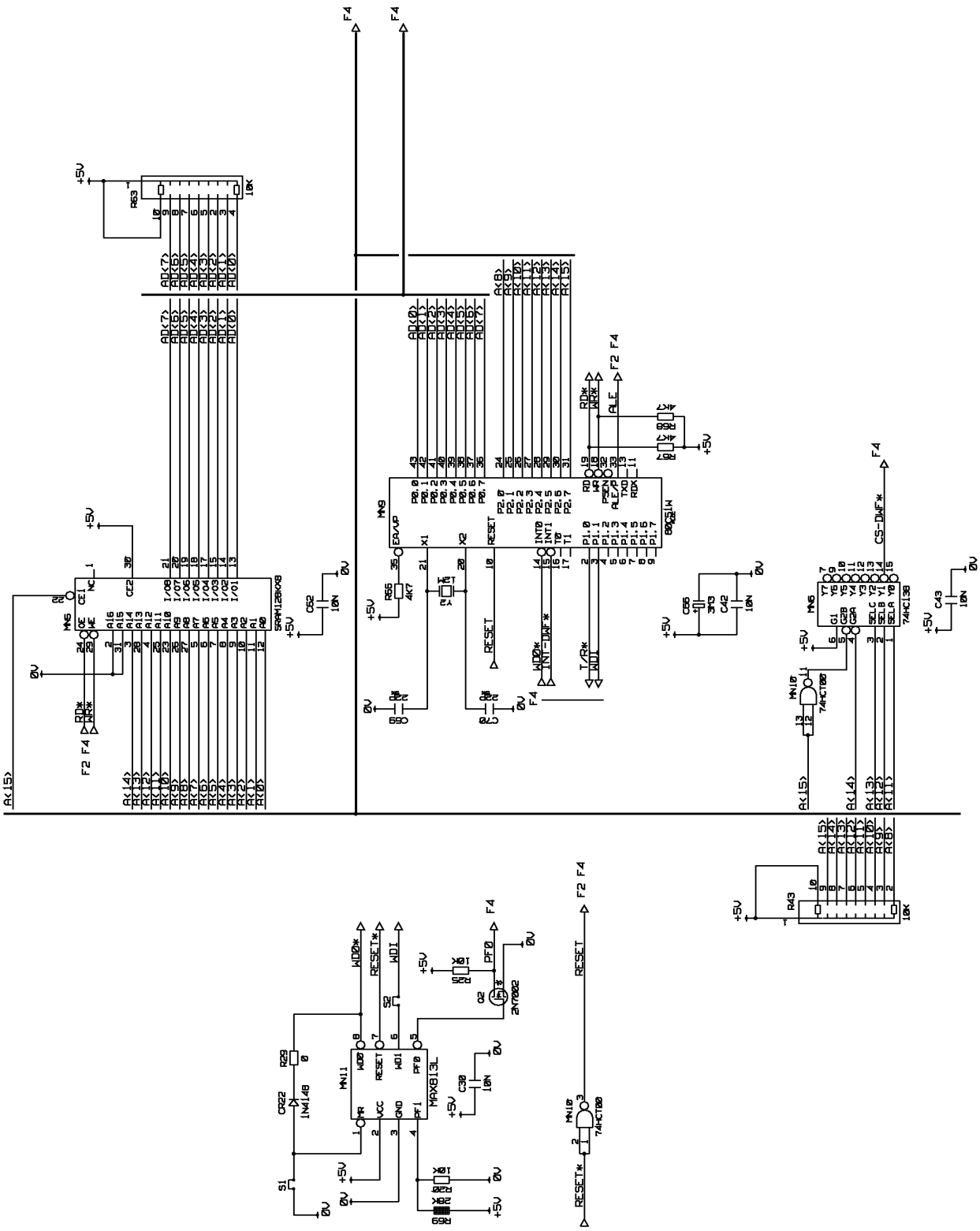


Figure 8.2 – Intel 8051 with external memory

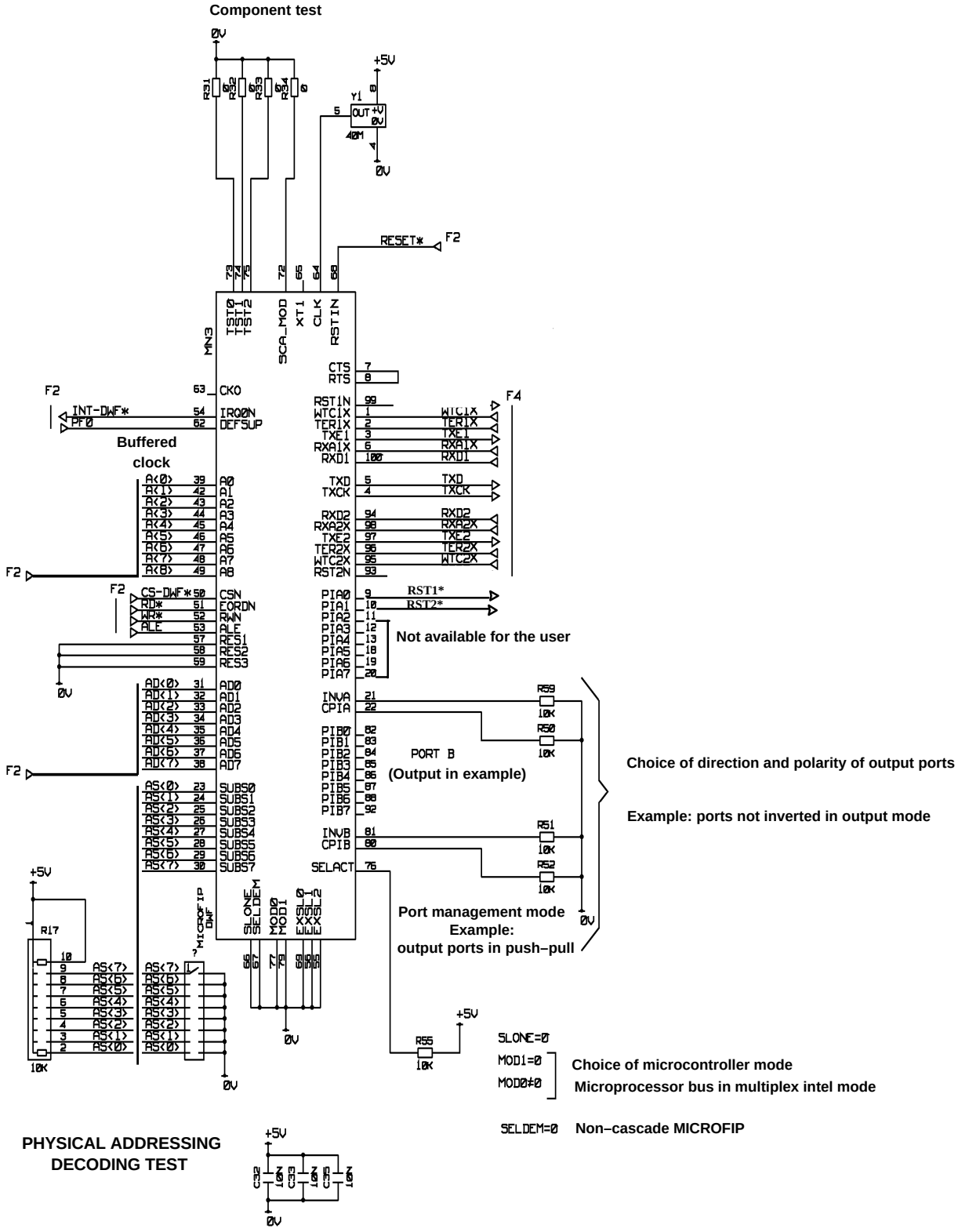


Figure 8.3 – MICROFIP used with Intel 8051 (`mf_redundancy` compiling option selected)

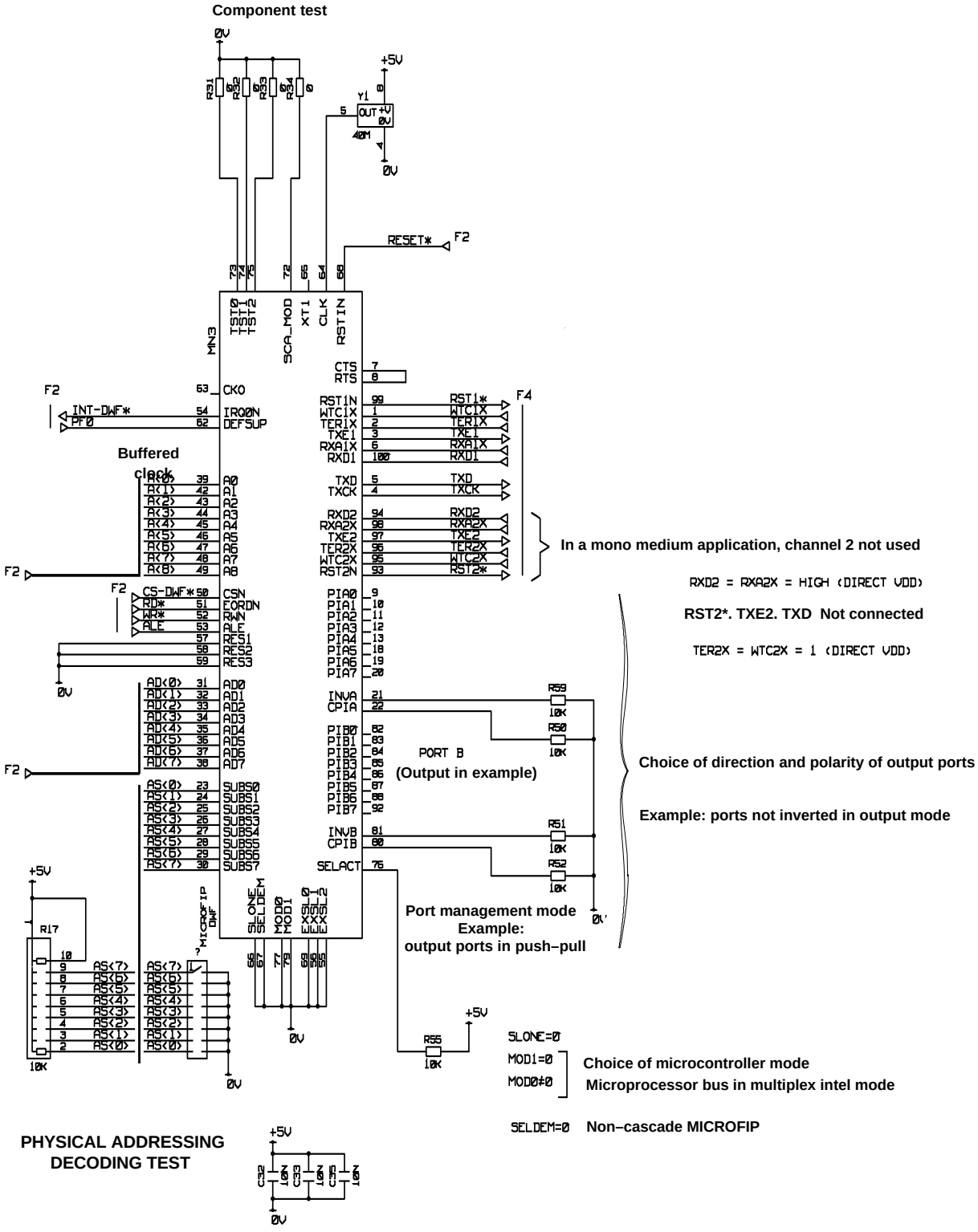


Figure 8.4 – MICROFIP used with Intel 8051
(without `mf_redundancy` compiling option selected)

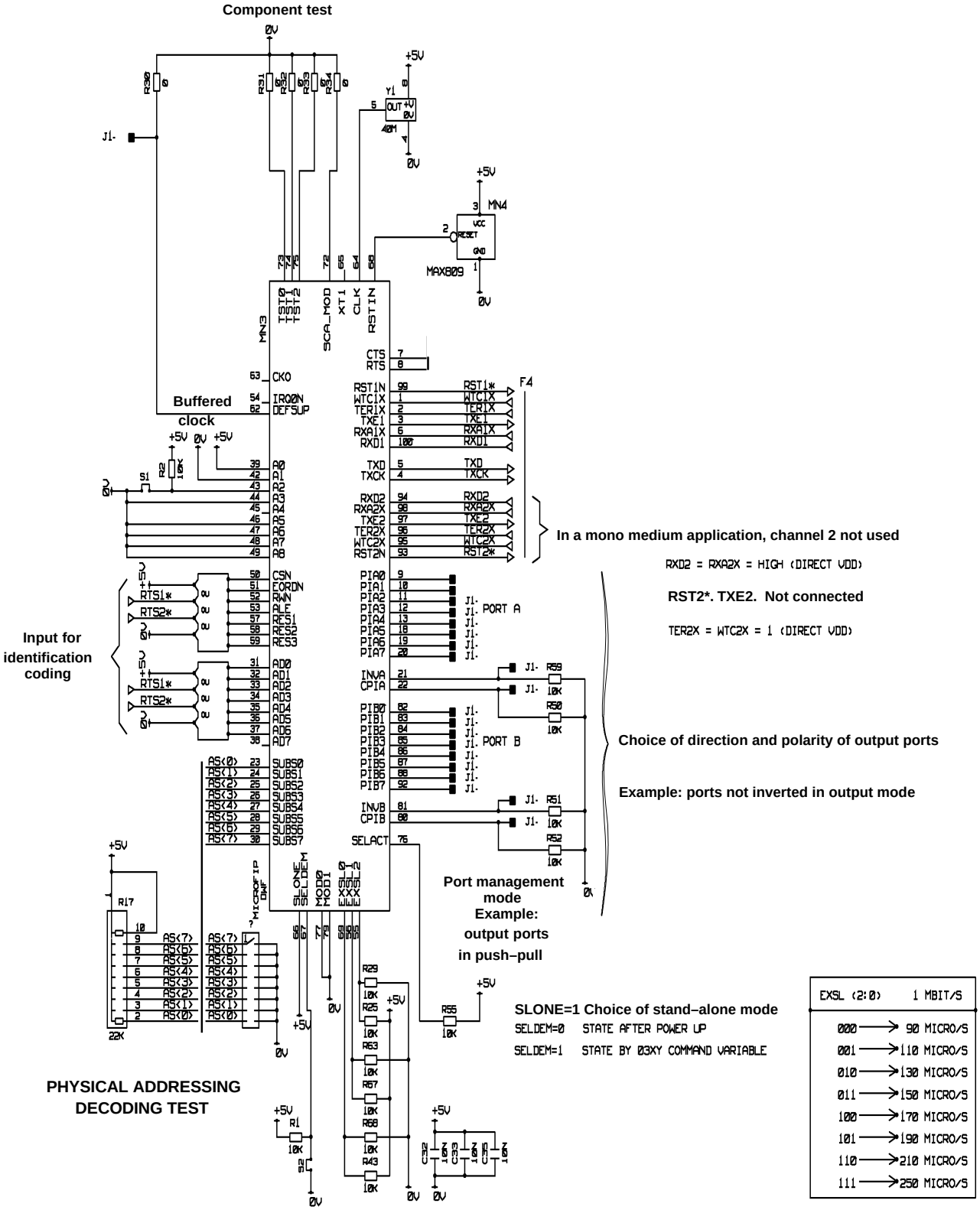
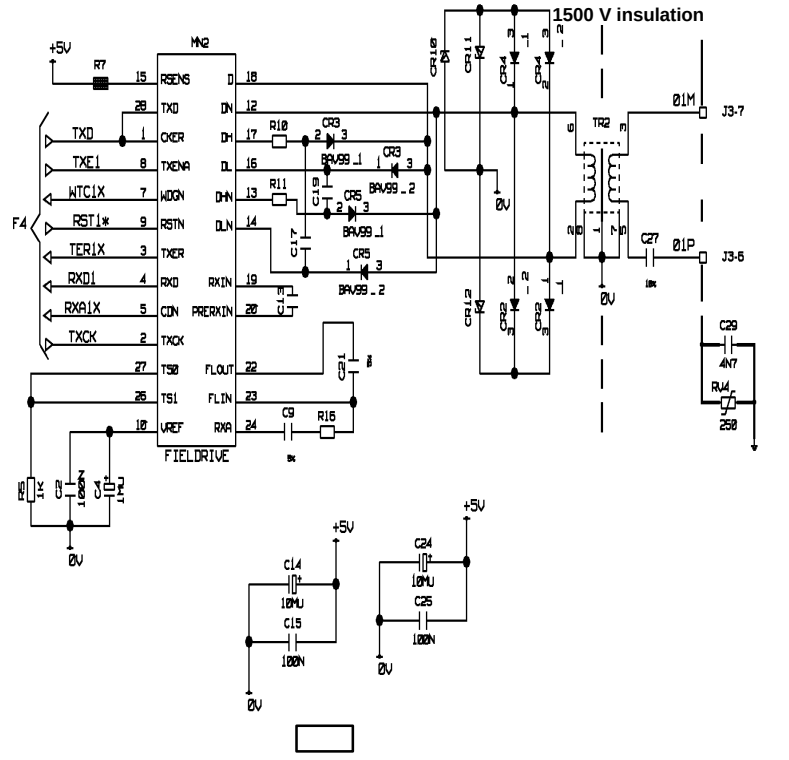


Figure 8.5 – MICROFIP used in standalone mode



The passive elements to be chosen depending on the speed are indicated in the FIELDRIVE data sheet.

Figure 8.6 – FIELDRIVE line transceiver
(Three speeds and redundancy options)