

VPC3+S

User Manual

**Revision 1.06**



#### Liability Exclusion

We have tested the contents of this document regarding agreement with the hardware and software described. Nevertheless, there may be deviations and we do not guarantee complete agreement. The data in the document is tested periodically, however. Required corrections are included in subsequent versions. We gratefully accept suggestions for improvements.

#### Copyright

Copyright © profichip GmbH 2009-2012. All Rights Reserved.

Unless permission has been expressly granted, passing on this document or copying it, or using and sharing its content are not allowed. Offenders will be held liable. All rights reserved, in the event a patent is granted or a utility model or design is registered.

This document is subject to technical changes.

1. Introduction 5
2. Functional Description 7
   1. [Overview 7](#_TOC_250076)
3. Pin Description 9
   1. [Pinout 9](#_TOC_250075)
   2. [Pin Assignment (Overview) 11](#_TOC_250074)
      1. [Asynchronous Intel Mode 13](#_TOC_250073)
      2. [Synchronous Intel Mode 14](#_TOC_250072)
      3. [Asynchronous Motorola Mode 15](#_TOC_250071)
      4. [Synchronous Motorola Mode 16](#_TOC_250070)
      5. [SPI Mode 17](#_TOC_250069)
      6. [I2C Mode 17](#_TOC_250068)
4. Memory Organization 19
   1. [Overview 19](#_TOC_250067)
   2. [Control Parameters (Latches/Registers) 21](#_TOC_250066)
   3. [Organizational Parameters (RAM) 23](#_TOC_250065)
5. ASIC Interface 25
   1. [Mode Registers 25](#_TOC_250064)
      1. [Mode Register 0 25](#_TOC_250063)
      2. [Mode Register 1 27](#_TOC_250062)
      3. [Mode Register 2 29](#_TOC_250061)
      4. [Mode Register 3 31](#_TOC_250060)
   2. [Status Register 32](#_TOC_250059)
   3. [Interrupt Controller 34](#_TOC_250058)
      1. [Interrupt Request Register 35](#_TOC_250057)
      2. [Interrupt Acknowledge / Mask Register 38](#_TOC_250056)
   4. [Watchdog Timer 38](#_TOC_250055)
      1. [Automatic Baud Rate Identification 39](#_TOC_250054)
      2. [Baud Rate Monitoring 39](#_TOC_250053)
      3. [Response Time Monitoring 39](#_TOC_250052)
6. PROFIBUS DP Interface 41
   1. [DP Buffer Structure 41](#_TOC_250051)
   2. [Description of the DP Services 44](#_TOC_250050)
      1. [Set\_Slave\_Add (SAP 55) 44](#_TOC_250049)
      2. [Set \_Prm (SAP 61) 45](#_TOC_250048)
      3. [Chk\_Cfg (SAP 62) 49](#_TOC_250047)
      4. [Slave\_Diag (SAP 60) 50](#_TOC_250046)
      5. Write\_Read\_Data / Data\_Exchange (Default\_SAP) 52
      6. [Global\_Control (SAP 58) 56](#_TOC_250045)
      7. [RD\_Input (SAP 56) 57](#_TOC_250044)
      8. [RD\_Output (SAP 57) 57](#_TOC_250043)
      9. [Get\_Cfg (SAP 59) 58](#_TOC_250042)
7. PROFIBUS DP Extensions 59
   1. [Set\_(Ext\_)Prm (SAP 53 / SAP 61) 59](#_TOC_250041)
   2. [PROFIBUS DP-V1 60](#_TOC_250040)
      1. [Acyclic Communication Relationships 60](#_TOC_250039)
      2. [Diagnosis Model 63](#_TOC_250038)
   3. [PROFIBUS DP-V2 64](#_TOC_250037)
      1. [DXB (Data eXchange Broadcast) 64](#_TOC_250036)
      2. [IsoM (Isochronous Mode) 70](#_TOC_250035)
         1. [IsoM-PLL 74](#_TOC_250034)
      3. [CS (Clock Synchronization) 80](#_TOC_250033)
8. Hardware Interface 87
   1. [Universal Processor Bus Interface 87](#_TOC_250032)
      1. [Overview 87](#_TOC_250031)
      2. [Parallel Interface Modes 88](#_TOC_250030)
      3. [SPI Interface Mode 91](#_TOC_250029)
      4. [I2C Interface Mode 97](#_TOC_250028)
      5. [Application Examples (Principles) 103](#_TOC_250027)
      6. [Application with 80C32 (2K Byte RAM Mode) 105](#_TOC_250026)
      7. [Application with 80C32 (4K Byte RAM Mode) 106](#_TOC_250025)
      8. [Application with 80C165 107](#_TOC_250024)
   2. [Dual Port RAM Controller 107](#_TOC_250023)
   3. [UART 108](#_TOC_250022)
   4. [ASIC Test 108](#_TOC_250021)
9. PROFIBUS Interface 109
   1. [Pin Assignment 109](#_TOC_250020)
   2. [Example for the RS485 Interface 110](#_TOC_250019)
10. Operational Specifications 111
    1. [Absolute Maximum Ratings 111](#_TOC_250018)
    2. [Recommended Operating Conditions 111](#_TOC_250017)
    3. [General DC Characteristics 111](#_TOC_250016)
    4. [Ratings for the Output Drivers 112](#_TOC_250015)
    5. [DC Electrical Characteristics 112](#_TOC_250014)
    6. [Timing Characteristics 113](#_TOC_250013)
       1. [System Bus Interface 113](#_TOC_250012)
       2. [Timing in the Synchronous Intel Mode 114](#_TOC_250011)
       3. [Timing in the Asynchronous Intel Mode 116](#_TOC_250010)
       4. [Timing in the Synchronous Motorola Mode 118](#_TOC_250009)
       5. [Timing in the Asynchronous Motorola Mode 120](#_TOC_250008)
       6. [Timing in SPI Interface Mode 123](#_TOC_250007)
       7. [Timing in I2C Interface Mode 125](#_TOC_250006)
    7. [Package Specifications 126](#_TOC_250005)
       1. [LFBGA48 126](#_TOC_250004)
       2. [LQFP48 128](#_TOC_250003)
    8. [Processing Instructions 130](#_TOC_250002)
    9. [Ordering Information 130](#_TOC_250001)

[Revision History 131](#_TOC_250000)

Profichip’s **VPC3+S** is a communication chip with 8-Bit parallel processor interface for intelligent PROFIBUS DP-Slave applications. Alternatively an SPI or I2C interface can be used to communicate with the chip.

The VPC3+S handles the message and address identification, the data security sequences and the protocol processing for PROFIBUS DP. In ad- dition the acyclic communication and alarm messages, described in DP-V1 extension, are supported. Furthermore the slave-to-slave communication Data eXchange Broadcast (DXB) and the Isochronous Bus Mode (IsoM), described in DP-V2 extension, are also provided. For high-precision syn- chronized motion control applications the chip is equipped with an HW-PLL for IsoM.

Automatic recognition and support of data transmissions rates up to 12 Mbit/s, the integration of the complete PROFIBUS DP protocol, 4K Byte communication RAM and the configurable processor interface are features to create high-performance PROFIBUS DP-Slave applications. The device is to be operated with 3.3V single supply voltage. All inputs are 5V tolerant.

Profichip’s **VPC3+S** is another member of profichip’s successful VPC3+ family. It is software compatible to other VPC3+ series devices however it offers some unique features like serial processor interfaces, IsoM-PLL and a very small package.

As there are also simple devices in the automation engineering area, such as switches or thermo elements, that do not require a microcontroller for data preprocessing, profichip offers a DP-Slave ASIC with 32 direct in- put/output bits. The **VPCLS2** handles the entire data traffic independently. No additional microprocessor or firmware is necessary. The VPCLS2 is compatible to existing chips.

Further information about our products or current and future projects is available on our web page: [http://www.profichip.com](http://www.profichip.com/).

## Notes:

## Overview

The VPC3+S makes a cost optimized design of intelligent PROFIBUS DP- Slave applications possible.

Due to the very flexible processor interface the VPC3+S supports a broad range of processor types and families. Please check the corresponding chapters of this manual for details. Here are just some common examples:

Intel: 80C31, 80C51, 80X86 and their derivates Siemens: 80C166/165/167

Motorola: HC11-, HC16-, and HC916 types

ARM: all ARM derivates with parallel, SPI or I2C interface

The VPC3+S handles the physical layer 1 and the data link layer 2 of the ISO/OSI-reference-model excluding the analog RS485 drivers.

The **integrated 4K Byte Dual-Port-RAM** serves as an interface between the VPC3+S and the software/application. In case of using 2K Byte the entire memory is divided into 256 segments, with 8 bytes each. Otherwise in the 4K Byte mode the segment base addresses starts at multiple of 16. Addressing by the user is done directly; however, the internal Micro Sequencer (MS) addresses the RAM by means of the so-called base- pointer. The base-pointer can be positioned at the beginning of a segment in the memory. Therefore, all buffers must be located at the beginning of a segment.

If the VPC3+S carries out a DP communication it automatically sets up all DP-SAPs. The various telegram information is made available to the user in separate data buffers (for example, parameter and configuration data). Three buffers are provided for data communication (three for output data and three for input data). As one buffer is always available for communica- tion no resource problems can occur. For optimal diagnosis support, the VPC3+S offers two Diagnosis-Buffers. The user enters the updated diagnosis data into these buffers. One Diagnosis-Buffer is always assigned to the VPC3+S.

The **Bus Interface Unit** is a parameterizable synchronous/asynchronous 8- bit parallel interface for various Intel and Motorola microcontrollers/pro- cessors. The user can directly access the internal 2K/4K Byte RAM or the parameter latches and control registers via the 11/12-bit address bus. Alternatively serial standard protocols like SPI or I2C can be used to access the VPC3+S.

Procedure-specific parameters (Station\_Address, control bits, etc.) must be transferred to the **Parameter Registers** and to the **Mode Registers** after power-on.

The MAC status can be observed at any time in the **Status Register**.

Various events (e.g. various indications, error events, etc.) are entered in the **Interrupt Controller**. These events can be individually enabled via a mask register. Acknowledgement takes place by means of the acknowl- edge register. The VPC3+S has a common interrupt output.

The integrated **Watchdog Timer** is operated in three different states: BAUD\_SEARCH, BAUD\_CONTROL and DP\_CONTROL.

The **Micro Sequencer** (MS) controls the entire process. It contains the DP- Slave state machine (DP\_SM).

The integrated **4K Byte RAM** that operates as a Dual-Port-RAM contains procedure-specific parameters (buffer pointer, buffer lengths, Station\_Address, etc.) and the data buffers.

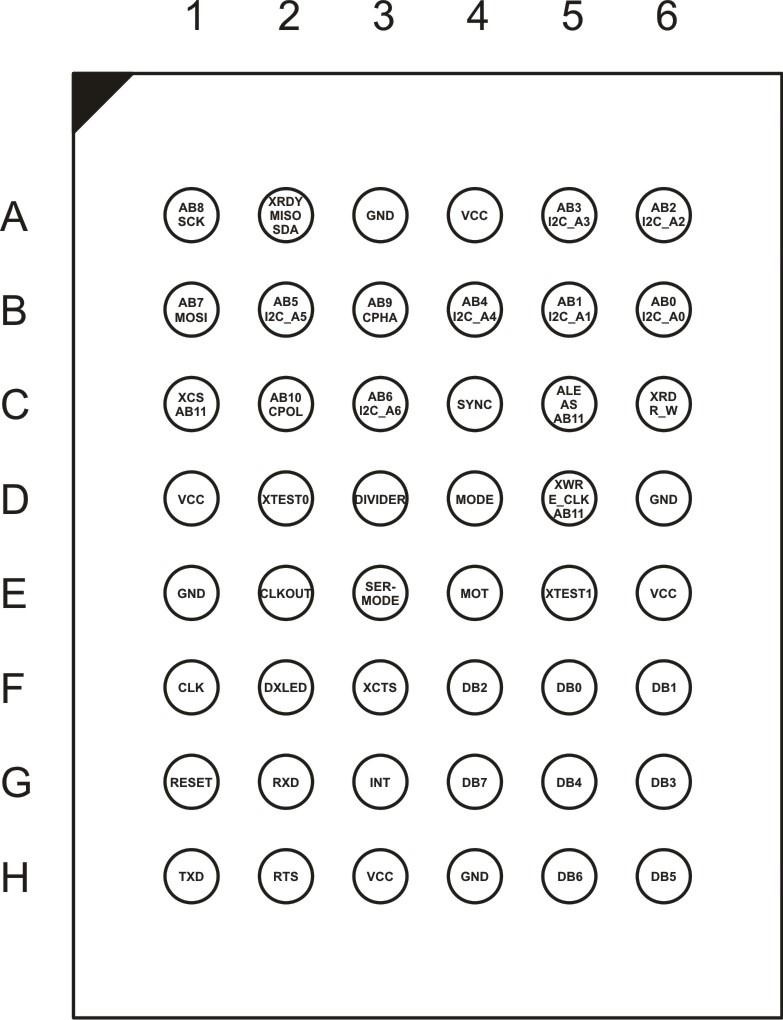
In the **UART**, the parallel data flow is converted into the serial data flow and vice-versa. The VPC3+S is capable of automatically identifying the baud rates (9.6 Kbit/s - 12 Mbit/s).

The **Idle Timer** directly controls the bus times on the serial bus line.

The **IsoM-PLL** provides high-precision synchronization mechanisms as defined in the PROFIBUS DPV2 protocol extension.

## Pinout

The VPC3+S is available in two package versions: LFBGA48 or LQFP48. Several pins are sharing different functions. Which pin function actually applies depends on the interface mode selected by the configuration pins. Four parallel interface modes as well as I2C and SPI mode with con- figurable clock phase and clock polarity are supported. Please see the following chapters for details.



**Figure 3-1: VPC3+S LFBGA48 Pinout (TOP VIEW)**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **36** | **35** | **34** | **33** | **32** | **31** | **30** | **29** | **28** | **27** | **26** | **25** |
|  |  |  |  |  |  |  |  |  |  |  |  |

**AB2 / I2C\_SA2 AB3 / I2C\_SA3 AB1 / I2C\_SA1**



**AB0 / I2C\_SA0**

**ALE / AS / AB11**

**XRD / R\_W**

**MODE**

**XWR / E\_CLOCK / AB11**

**GND**

**VCC**

**XTEST1**

**MOT / XINT**

**DB1**

**DB0**

**DB3**

|  |  |
| --- | --- |
| **37** |  |
| **38** |  |
| **39** |  |
| **40** |  |
| **41** |  |
| **42** |  |
| **43** |  |
| **44** |  |
| **45** |  |
| **46** |  |
| **47** |  |
| **48** |  |

**SYNC AB4 / I2C\_SA4**

**VCC GND**

**AB9 / SPI\_CPHA AB6 / I2C\_SA6 AB5 / I2C\_SA5**

**XREADY / DTACK /SPI\_MISO / I2C\_SDA**

**AB8 / SPI\_SCK /**

**I2C\_SCK**

**AB7 / SPI\_MOSI**

**AB10 / SPI\_CPOL**

**XCS / AB11 / SPI\_XSS**

**DIVIDER**

**XTEST0**

**VCC**

**GND**

**CLKOUT**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** |

**Figure 3-2: VPC3+S LQFP48 Pinout (TOP VIEW)**

**DB5 DB6 DB4 DB2 DB7 GND VCC INT XCTS RXD RTS TXD**

**SERMODE**

**CLK**

**RESET**

|  |  |
| --- | --- |
|  | **24** |
|  | **23** |
|  | **22** |
|  | **21** |
|  | **20** |
|  | **19** |
|  | **18** |
|  | **17** |
|  | **16** |
|  | **15** |
|  | **14** |
|  | **13** |

Details about package outlines and dimensions are listed in section 10.7.

**XDATAEXCH**

## Pin Assignment (Overview)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Ball**  **BGA** | **Pin**  **QFP** | **Signal Name** | **In/Out** | **Description** | **Source / Destination** |
| A1 | 48 | AB8 | I(S) | Address Bus 8 | CPU |
| SPI\_SCK / I2C\_SCK | SPI: Serial Clock / I2C: Serial Clock |
| A2 | 47 | XREADY / XDTACK | I(S)/O | READY / DTACK for external CPU | CPU |
| SPI\_MISO / I2C\_SDA | SPI: Master-In-Slave-Out / I2C: Serial Data |
| A3 | 7 | GND |  |  |  |
| A4 | 6 | VCC |  |  |  |
| A5 | 38 | AB3 | I | Address Bus 3 | CPU |
| I2C\_SA3 | I2C: Slave Address 3 | Configuration Pin |
| A6 | 37 | AB2 | I | Address Bus 2 | CPU |
| I2C\_SA2 | I2C: Slave Address 2 | Configuration Pin |
| B1 | 1 | AB7 | I(S) | Address Bus 7 | CPU |
| SPI\_MOSI | SPI: Master-Out-Slave-In | Configuration Pin |
| B2 | 46 | AB5 | I | Address Bus 5 | CPU |
| I2C\_SA5 | I2C: Slave Address 5 | Configuration Pin |
| B3 | 44 | AB9 | I | Address Bus 9 | CPU |
| SPI\_CPHA | SPI: Clock Phase | Configuration Pin |
| B4 | 41 | AB4 | I | Address Bus 4 | CPU |
| I2C\_SA4 | I2C: Slave Address 4 | Configuration Pin |
| B5 | 39 | AB1 | I | Address Bus 1 | CPU |
| I2C\_SA1 | I2C: Slave Address 1 | Configuration Pin |
| B6 | 36 | AB0 | I | Address Bus 0 | CPU |
| I2C\_SA0 | I2C: Slave Address 0 | Configuration Pin |
| C1 | 3 | XCS / AB11 | I | Chip-Select / Address Bus 11 | CPU |
| SPI\_XSS | SPI: Slave-Select |
| C2 | 2 | AB10 | I | Address Bus 10 | CPU |
| SPI\_CPOL | SPI: Clock Polarity | Configuration Pin |
| C3 | 45 | AB6 | I | Address Bus 6 | CPU |
| I2C\_SA6 | I2C: Slave Address 6 | Configuration Pin |
| C4 | 40 | SYNC | O | Synchronization Pulse | CPU / Motion Control |
| C5 | 35 | ALE / AS / AB11 | I | Address Latch Enable / Address Strobe / Address Bus 11 | CPU |
| C6 | 34 | XRD / R\_W | I | Read / Read-Write | CPU |
| D1 | 18 | VCC |  |  |  |
| D2 | 5 | XTEST0 | I | Test Pin 0 (to be connected to VCC) | Test Pin |
| D3 | 4 | DIVIDER | I | Divider setting for CLKOUT: ‘0’: 12 MHz  ‘1’: 24 MHz | Configuration Pin |
| D4 | 33 | MODE | I | ‘0’: Asynchronous Mode (Parallel Interface Mode) ‘1’: Synchronous Mode (Parallel Interface Mode)  ‘0’: SPI (Serial Interface Mode) ‘1’: I2C (Serial Interface Mode) | Configuration Pin |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Ball**  **BGA** | **Pin**  **QFP** | **Signal Name** | **In/Out** | **Description** | **Source / Destination** |
| D5 | 32 | XWR / E\_CLOCK / AB11 | I | Write / E-Clock (Motorola) / Address Bus 11 | CPU |
| D6 | 19 | GND |  |  |  |
| E1 | 31 | GND |  |  |  |
| E2 | 8 | CLKOUT | O | Clock Output (12 MHz or 24 MHz) | CPU / System |
| E3 | 9 | SERMODE | I | ‘0’: Parallel Interface  ‘1’: Serial Interface (SPI or I2C) | Configuration Pin |
| E4 | 28 | MOT/XINT | I | ‘0’: Parallel Interface Intel Format  ‘1’: Parallel Interface Motorola Format | Configuration Pin |
| E5 | 29 | XTEST1 | I | Test Pin 1 (to be connected to VCC) | Test Pin |
| E6 | 30 | VCC |  |  |  |
| F1 | 10 | CLK | I(S) | System Clock (48 MHz) | System |
| F2 | 11 | XDATAEXCH | O | Indicates state ‘Data-Exchange’ for PROFIBUS DP | LED |
| F3 | 16 | XCTS | I | Clear-To-Send (for FSK-Modem) | PB-Interface |
| F4 | 21 | DB2 | IO | Data Bus 2 | CPU |
| F5 | 26 | DB0 | IO | Data Bus 0 | CPU |
| F6 | 27 | DB1 | IO | Data Bus 1 | CPU |
| G1 | 12 | RESET | I(S) | Master-Reset (connect to port pin of CPU) | CPU |
| G2 | 15 | RXD | I | Receive Data | PB-Interface |
| G3 | 17 | INT | O | Interrupt | CPU / IRQ Controller |
| G4 | 20 | DB7 | IO | Data Bus 7 | CPU |
| G5 | 22 | DB4 | IO | Data Bus 4 | CPU |
| G6 | 25 | DB3 | IO | Data Bus 3 | CPU |
| H1 | 13 | TXD | O | Transmit Data (external pull-up resistor required) | PB-Interface |
| H2 | 14 | RTS | O | Request-To-Send | PB-Interface |
| H3 | 42 | VCC |  |  |  |
| H4 | 43 | GND |  |  |  |
| H5 | 23 | DB6 | IO | Data Bus 6 | CPU |
| H6 | 24 | DB5 | IO | Data Bus 7 | CPU |

**Figure 3-3: Pin Assignment**

**Notes:** All signals beginning with ‘X’ are LOW active.

VCC = +3.3 V

GND = 0 V

The assignment of AB11 depends on the parallel interface mode selected. All unused inputs must be connected to GND.

**Input Levels:**

|  |  |
| --- | --- |
| I : | LVTTL |
| I (S) : | LVTTL, Schmitt-Trigger |

The following chapters are describing the different processor interface modes supported by the VPC3+S. For every interface mode the settings of the configuration pins and the signals necessary to communicate with the microcontroller are listed. Common signals for all interface types (like clock divider, interrupt and PROFIBUS interface signals) are not explicitly listed in this overview.

### Asynchronous Intel Mode

In Asynchronous Intel Mode the data and address busses are separate (non-multiplexed). Address line 11 is to be connected to pin BGA\_C5/ QFP\_35 of the VPC3+S.

XREADY mechanism is supported.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Ball**  **BGA** | **Pin**  **QFP** | **Signal Name** | **In/Out** | **Description** | **Connect to** |
| E3 | 9 | SERMODE | I | ‘0’: Parallel Interface | **GND** |
| E4 | 28 | MOT/XINT | I | ‘0’: Intel Format | **GND** |
| D4 | 33 | MODE | I | ‘0’: Asynchronous Interface Mode | **GND** |
| C5 | 35 | AB11 | I | Address Lines Bit 11 | CPU Address Bus 11 |
| C2 | 2 | AB10 | I | Address Lines Bits [10:0] | CPU  Address Bus [10:0] |
| B3 | 44 | AB9 | I |
| A1 | 48 | AB8 | I(S) |
| B1 | 1 | AB7 | I(S) |
| C3 | 45 | AB6 | I |
| B2 | 46 | AB5 | I |
| B4 | 41 | AB4 | I |
| A5 | 38 | AB3 | I |
| A6 | 37 | AB2 | I |
| B5 | 39 | AB1 | I |
| B6 | 36 | AB0 | I |
| G4 | 20 | DB7 | IO | Data Bus [7:0] | CPU Data Bus [7:0] |
| H5 | 23 | DB6 | IO |
| H6 | 24 | DB5 | IO |
| G5 | 22 | DB4 | IO |
| G6 | 25 | DB3 | IO |
| F4 | 21 | DB2 | IO |
| F6 | 27 | DB1 | IO |
| F5 | 26 | DB0 | IO |
| C1 | 3 | XCS | I | Chip-Select Signal (active low) | CPU Chip-Select |
| D5 | 32 | XWR | I | Write Signal (active low) | CPU Write |
| C6 | 34 | XRD | I | Read Signal (active low) | CPU Read |

**Figure 3-4: Interface Configuration: Asynchronous Intel Mode**

### Synchronous Intel Mode

In Synchronous Intel Mode the lower 8 bits of the address lines are multiplexed with the 8 bit data bus DB[7:0]. The upper address lines (bits 10 to 8) need to be connected to the AB[2:0] inputs of the VPC3+S. Address line 11 is to be connected to pin BGA\_C1/QFP\_3 of the VPC3+S.

XREADY mechanism is not supported in this interface mode.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Ball**  **BGA** | **Pin**  **QFP** | **Signal Name** | **In/Out** | **Description** | **Connect to** |
| E3 | 9 | SERMODE | I | ‘0’: Parallel Interface | **GND** |
| E4 | 28 | MOT/XINT | I | ‘0’: Intel Format | **GND** |
| D4 | 33 | MODE | I | ‘1’: Synchronous Interface Mode | **VCC** |
| C1 | 3 | AB11 | I | Address Bit 11 | CPU Address Bus 11 |
| A6 | 37 | AB2 | I | Address Bit 10 | CPU Address Bus 10 |
| B5 | 39 | AB1 | I | Address Bit 9 | CPU Address Bus 9 |
| B6 | 36 | AB0 | I | Address Bit 8 | CPU Address Bus 8 |
| G4 | 20 | DB7 | IO | Data Bus [7:0]  multiplexed with lower address bits [7:0] ALE used to latch the lower address bits. | CPU Data/Address Bus [7:0] |
| H5 | 23 | DB6 | IO |
| H6 | 24 | DB5 | IO |
| G5 | 22 | DB4 | IO |
| G6 | 25 | DB3 | IO |
| F4 | 21 | DB2 | IO |
| F6 | 27 | DB1 | IO |
| F5 | 26 | DB0 | IO |
| C2 | 2 | AB10 | I | In Synchronous Intel Mode these inputs are used to generate the internal Chip-Select signal.  Chip-Select is active if all inputs are ‘0’. | Use one (inverted) CPU Address Line for generating the VPC3+S Chip-Select signal.  Connect all other inputs to GND. |
| B3 | 44 | AB9 | I |
| A1 | 48 | AB8 | I(S) |
| B1 | 1 | AB7 | I(S) |
| C3 | 45 | AB6 | I |
| B2 | 46 | AB5 | I |
| B4 | 41 | AB4 | I |
| A5 | 38 | AB3 | I |
| C5 | 35 | ALE | I | Address Latch Enable  The lower address bits [7:0] are latched with the falling edge of ALE | CPU ALE |
| D5 | 32 | XWR | I | Write Signal (active low) | CPU Write |
| C6 | 34 | XRD | I | Read Signal (active low) | CPU Read |

**Figure 3-5: Interface Configuration: Synchronous Intel Mode**

### Asynchronous Motorola Mode

In Asynchronous Motorola Mode the data and address busses are separate (non-multiplexed). When using HC11 types with a multiplexed bus the address signals AB[7:0] must be generated from the DB[7:0] signals externally. Address line 11 is to be connected to pin BGA\_D5/QFP32 of the VPC3+S.

XDTACK mechanism is supported.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Ball**  **BGA** | **Pin**  **QFP** | **Signal Name** | **In/Out** | **Description** | **Connect to** |
| E3 | 9 | SERMODE | I | ‘0’: Parallel Interface | **GND** |
| E4 | 28 | MOT/XINT | I | ‘1’: Motorola Format | **VCC** |
| D4 | 33 | MODE | I | ‘0’: Asynchronous Interface Mode | **GND** |
| D5 | 32 | AB11 | I | Address Lines Bit 11 | CPU Address Bus 11 |
| C2 | 2 | AB10 | I | Address Lines Bits [10:0] | CPU  Address Bus [10:0] |
| B3 | 44 | AB9 | I |
| A1 | 48 | AB8 | I(S) |
| B1 | 1 | AB7 | I(S) |
| C3 | 45 | AB6 | I |
| B2 | 46 | AB5 | I |
| B4 | 41 | AB4 | I |
| A5 | 38 | AB3 | I |
| A6 | 37 | AB2 | I |
| B5 | 39 | AB1 | I |
| B6 | 36 | AB0 | I |
| G4 | 20 | DB7 | IO | Data Bus [7:0] | CPU Data Bus [7:0] |
| H5 | 23 | DB6 | IO |
| H6 | 24 | DB5 | IO |
| G5 | 22 | DB4 | IO |
| G6 | 25 | DB3 | IO |
| F4 | 21 | DB2 | IO |
| F6 | 27 | DB1 | IO |
| F5 | 26 | DB0 | IO |
| C1 | 3 | XCS | I | Chip-Select Signal (active low) | CPU Chip-Select |
| C5 | 35 | AS | I | Address Strobe (active low) | CPU Address Strobe |
| C6 | 34 | R\_W | I | Read-Write Signal (‘1’ = Read) | CPU Read-Write |

**Figure 3-6: Interface Configuration: Asynchronous Motorola Mode**

### Synchronous Motorola Mode

In Synchronous Motorola Mode the data and address busses are separate (non-multiplexed). When using HC11 types with a multiplexed bus the address signals AB[7:0] must be generated from the DB[7:0] signals externally. Address line 11 is to be connected to pin BGA\_C5/QFP\_35 of the VPC3+S.

XDTACK mechanism is not supported.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Ball**  **BGA** | **Pin**  **QFP** | **Signal Name** | **In/Out** | **Description** | **Connect to** |
| E3 | 9 | SERMODE | I | ‘0’: Parallel Interface | **GND** |
| E4 | 28 | MOT/XINT | I | ‘1’: Motorola Format | **VCC** |
| D4 | 33 | MODE | I | ‘1’: Synchronous Interface Mode | **VCC** |
| C5 | 35 | AB11 | I | Address Lines Bit 11 | CPU Address Bus 11 |
| C2 | 2 | AB10 | I | Address Lines Bits [10:0] | CPU  Address Bus [10:0] |
| B3 | 44 | AB9 | I |
| A1 | 48 | AB8 | I(S) |
| B1 | 1 | AB7 | I(S) |
| C3 | 45 | AB6 | I |
| B2 | 46 | AB5 | I |
| B4 | 41 | AB4 | I |
| A5 | 38 | AB3 | I |
| A6 | 37 | AB2 | I |
| B5 | 39 | AB1 | I |
| B6 | 36 | AB0 | I |
| G4 | 20 | DB7 | IO | Data Bus [7:0] | CPU Data Bus [7:0] |
| H5 | 23 | DB6 | IO |
| H6 | 24 | DB5 | IO |
| G5 | 22 | DB4 | IO |
| G6 | 25 | DB3 | IO |
| F4 | 21 | DB2 | IO |
| F6 | 27 | DB1 | IO |
| F5 | 26 | DB0 | IO |
| C1 | 3 | XCS | I | Chip-Select Signal (active low) | CPU Chip-Select |
| D5 | 32 | E\_CLOCK | I | E-Clock | CPU E-Clock |
| C6 | 34 | R\_W | I | Read-Write Signal (‘1’ = Read) | CPU Read-Write |

**Figure 3-7: Interface Configuration: Synchronous Motorola Mode**

### SPI Mode

The VPC3+S can be interfaced like an SPI compatible memory device. Depending on the setting of CPOL and CPHA four different SPI modes can be selected. All unused inputs (including DB[7:0]) must be connected to GND.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Ball**  **BGA** | **Pin**  **QFP** | **Signal Name** | **In/Out** | **Description** | **Connect to** |
| E3 | 9 | SERMODE | I | ‘1’: Serial Interface | **VCC** |
| E4 | 28 | MOT/XINT | I | ‘0’: not used in this mode | **GND** |
| D4 | 33 | MODE | I | ‘0’: SPI Mode | **GND** |
| C2 | 2 | SPI\_CPOL | I | Clock Polarity | VCC or GND |
| B3 | 44 | SPI\_CPHA | I | Clock Phase | VCC or GND |
| C1 | 3 | SPI\_XSS | I | Slave-Select Signal (active low) | CPU Slave-Select |
| A1 | 48 | SPI\_SCK | I(S) | Serial Clock | CPU SCK |
| B1 | 1 | SPI\_MOSI | I | Master-Out-Slave-In (Serial Data Input) | CPU MOSI |
| A2 | 47 | SPI\_MISO | O | Master-In-Slave-Out (Serial Data Output) | CPU MISO |

**Figure 3-8: Interface Configuration: SPI Mode**

### I2C Mode

The VPC3+S can be interfaced like an I2C compatible memory device. The VPC3+S is always in slave mode, master mode is not supported. The slave address can be configured by using the AB[6:0] inputs. All unused inputs (including DB[7:0]) must be connected to GND.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Ball**  **BGA** | **Pin**  **QFP** | **Signal Name** | **In/Out** | **Description** | **Connect to** |
| E3 | 9 | SERMODE | I | ‘1’: Serial Interface | **VCC** |
| E4 | 28 | MOT/XINT | I | ‘0’: not used in this mode | **GND** |
| D4 | 33 | MODE | I | ‘1’: I2C Mode | **VCC** |
| C3 | 45 | I2C\_SA6 | I | I2C Slave Address | VCC or GND |
| B2 | 46 | I2C\_SA5 | I | VCC or GND |
| B4 | 41 | I2C\_SA4 | I | VCC or GND |
| A5 | 38 | I2C\_SA3 | I | VCC or GND |
| A6 | 37 | I2C\_SA2 | I | VCC or GND |
| B5 | 39 | I2C\_SA1 | I | VCC or GND |
| B6 | 36 | I2C\_SA0 | I | VCC or GND |
| A1 | 48 | I2C\_SCK | I(S) | Serial Clock | CPU SCK |
| A2 | 47 | I2C\_SDA | I(S) / O | Serial Data Line | CPU SDA |

**Figure 3-9: Interface Configuration: I2C Mode**

## Notes:

## Overview

The internal Control Parameters are located in the first 21 addresses. The latches/registers either come from the internal controller or influence the controller. Certain cells are read- or write-only. The internal working cells, which are not accessible by the user, are located in RAM at the same address locations.

The Organizational Parameters are located in RAM beginning with address 16H. The entire buffer structure (for the DP-SAPs) is based on these pa- rameters. In addition, general parameter data (Station\_Address, Ident\_Number, etc.) and status information (Global\_Control command, etc.) are also stored in these cells.

Corresponding to the parameter setting of the Organizational Parameters, the user-generated buffers are located beginning with address 40H. All buffers or lists must begin at segment addresses (8 bytes segmentation for 2K Byte mode, 16 bytes segmentation for 4K Byte mode).

|  |  |  |  |
| --- | --- | --- | --- |
| **Address** | **Function** | | |
| 000H  : 015H | Control Parameters (latches/registers) (21 bytes) | | Internal working cells |
| 016H |  | | |
| : | Organizational Parameters (42 bytes) | | |
| 03FH |  | | |
| 040H | DP-buffers:  DP-V1-buffer: DP-V2-buffer: | Data in (3)\* Data out (3)\*\*  Diagnosis data(2) Parameter data (1)  Configuration data (2)  Auxiliary buffers (2)  SSA-buffer (1)  SAP-List (1)  Indication / Response buffers \*\*\* DXB out (3)\*\*\*\*  DXB-buffers (2)  CS-buffer (1)  PLL-buffer (1) | |
| : |
| : |
| 7FFH (FFFH) |

**Figure 4-1: Memory Table**

\* Data in means input data from DP-Slave to DP-Master

\*\* Data out means output data from DP-Master to DP-Slave

\*\*\* Number of buffers depends on the entries in the SAP-List

\*\*\*\* DXB out means input data from another DP-Slave (slave-to-slave communication)

|  |  |
| --- | --- |
| **Internal VPC3+S RAM (2K/4K Byte)** | |
| Segment 0 |  |
| Segment 1 |  |
| Segment 2 | 8/16 bit segment addresses (pointer to the buffers) |
|  |  |
| Segment 254 |  |
| Segment 255 |  |

Building of the physical buffer address: 2K Byte Mode:

7 0

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |

Segment base address (8 bit)

Offset (3 bit)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 0 | 0 |  |  |  |

+

10 0

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |

Physical address (11 bit)

4K Byte Mode:

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 7 |  |  |  |  |  |  | 0 |  |  |  |  |  |
|  |  |  |  |  |  |  |  | Segment base address (8 bit) | | | | |
|  | | | | | | | | | | | | |
|  | | | | 0 | 0 | 0 | 0 |  |  |  |  | Offset (4 bit) |
| + |  |  |  |  |  |  |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |  |  |  |  | 0 |  |
|  |  |  |  |  |  |  |  |  |  |  |  | Physical address (12 bit) |

## Control Parameters (Latches/Registers)

These cells can be either read-only or write-only. In the Motorola Mode the VPC3+S carries out ‘address swapping’ for an access to the address locations 00H - 07H (word registers). That is, the VPC3+S internally generates an even address from an odd address and vice-versa.

|  |  |  |  |
| --- | --- | --- | --- |
| **Address**  **Intel Mot.** | | **Name Bit No.** | **Significance (Read Access!)** |
| 00H | 01H | Int-Req-Reg 7..0 | Interrupt Controller Register |
| 01H | 00H | Int-Req-Reg 15..8 |
| 02H | 03H | IntReg 7..0 |
| 03H | 02H | IntReg 15..8 |
| 04H | 05H | Status-Reg 7..0 | Status Register |
| 05H | 04H | Status-Reg 15..8 |
| 06H | 07H | Mode-Reg 0 7..0 | Mode Register 0 |
| 07H | 06H | Mode-Reg 0 15..8 |
| 08H | | Din\_Buffer\_SM 7..0 | Buffer assignment of the DP\_Din\_Buffer\_State\_Machine |
| 09H | | New\_Din\_Buffer\_Cmd 1..0 | The user makes a new DP Din\_Buf available in the N state. |
| 0AH | | Dout\_Buffer\_SM 7..0 | Buffer assignment of the DP\_Dout\_Buffer\_State\_Machine |
| 0BH | | Next\_Dout\_Buffer\_Cmd 3..0 | The user fetches the last DP Dout\_Buf from the N state |
| 0CH | | Diag\_Buffer\_SM 3..0 | Buffer assignment for the DP\_Diag\_Buffer\_State\_Machine |
| 0DH | | New\_Diag\_Buffer\_Cmd 1..0 | The user makes a new DP Diag\_Buf available to the VPC3+S. |
| 0EH | | User\_Prm\_Data\_Okay 1..0 | The user positively acknowledges the user parameter setting data of a Set\_(Ext\_)Prm telegram. |
| 0FH | | User\_Prm\_Data\_Not\_Okay 1..0 | The user negatively acknowledges the user parameter setting data of a Set\_(Ext\_)Prm telegram. |
| 10H | | User\_Cfg\_Data\_Okay 1..0 | The user positively acknowledges the configuration data of a Chk\_Cfg telegram. |
| 11H | | User\_Cfg\_Data\_Not\_Okay 1..0 | The user negatively acknowledges the configuration data of a Chk\_Cfg telegram. |
| 12H | | DXBout\_Buffer\_SM 7..0 | Buffer assignment of the DXBout\_Buffer\_State\_Machine |
| 13H | | Next\_DXBout\_Buffer\_Cmd 2..0 | The user fetches the last DXBout\_Buf from the N state |
| 14H | | SSA\_Buffer\_Free\_Cmd | The user has fetched the data from the SSA\_Buf and enables the buffer again. |
| 15H | | Mode-Reg 1 7..0 |  |

**Figure 4-2: Assignment of the Internal Parameter-Latches for READ**

|  |  |  |  |
| --- | --- | --- | --- |
| **Address**  **Intel Mot.** | | **Name Bit No.** | **Significance (Write Access!)** |
| 00H | 01H | Int-Req-Reg 7..0 | Interrupt-Controller-Register |
| 01H | 00H | Int-Req\_Reg 15..8 |
| 02H | 03H | Int-Ack-Reg 7..0 |
| 03H | 02H | Int-Ack-Reg 15..8 |
| 04H | 05H | IntMask-Reg 7..0 |
| 05H | 04H | IntMask-Reg 15..8 |
| 06H | 07H | Mode-Reg0 7..0 | Setting parameters for individual bits |
| 07H | 06H | Mode-Reg0 15..8 |
| 08H | | Mode-Reg1-S 7..0 |  |
| 09H | | Mode-Reg1-R 7..0 |
| 0AH | | WD\_BAUD\_CONTROL\_Val 7..0 | Square-root value for baud rate monitoring |
| 0BH | | minTSDR\_Val 7..0 | minTSDR time |
| 0CH | | Mode-Reg2 7..0 | Mode Register 2 |
| 0DH | | Sync\_PW\_Reg 7..0 | Sync Pulse Width Register |
| 0EH | | Control\_Command\_Reg 7..0 | Control\_Command value for comparison with SYNCH telegram |
| 0FH | | Group\_Select\_Reg 7..0 | Group\_Select value for comparison with SYNCH telegram |
| 10H | | Reserved |  |
| 11H | |
| 12H | | Mode-Reg3 7..0 | Mode Register 3 |
| 13H | | Reserved |  |
| 14H | |
| 15H | |

**Figure 4-3: Assignment of the Internal Parameter-Latches for WRITE**

## Organizational Parameters (RAM)

The user stores the organizational parameters in the RAM under the specified addresses. These parameters can be written and read.

|  |  |  |  |
| --- | --- | --- | --- |
| **Address**  **Intel Mot.** | | **Name Bit No.** | **Significance** |
| 16H | | R\_TS\_Adr | Setup Station\_Address of the VPC3+S |
| 17H | | SAP\_List\_Ptr | Pointer to a RAM address which is preset with FFh or to SAP-List |
| 18H | 19H | R\_User\_WD\_Value 7..0 | In DP\_Mode an internal 16-bit watchdog timer monitors the user. |
| 19H | 18H | R\_User\_WD\_Value 15..8 |
| 1AH | | R\_Len\_Dout\_Buf | Length of the 3 Dout\_Buf |
| 1BH | | R\_Dout\_Buf\_Ptr1 | Segment base address of Dout\_Buf 1 |
| 1CH | | R\_Dout\_Buf\_Ptr2 | Segment base address of Dout\_Buf 2 |
| 1DH | | R\_Dout\_Buf\_Ptr3 | Segment base address of Dout\_Buf 3 |
| 1EH | | R\_Len\_Din\_Buf | Length of the 3 Din\_Buf |
| 1FH | | R\_Din\_Buf\_Ptr1 | Segment base address of Din\_Buf 1 |
| 20H | | R\_Din\_Buf\_Ptr2 | Segment base address of Din\_Buf 2 |
| 21H | | R\_Din\_Buf\_Ptr3 | Segment base address of Din\_Buf 3 |
| 22H | | R\_Len\_DXBout\_Buf | Length of the 3 DXBout\_Buf |
| 23H | | R\_DXBout\_Buf\_Ptr1 | Segment base address of DXBout\_Buf 1 |
| 24H | | R\_Len Diag\_Buf1 | Length of Diag\_Buf 1 |
| 25H | | R\_Len Diag\_Buf2 | Length of Diag\_Buf 2 |
| 26H | | R\_Diag\_Buf\_Ptr1 | Segment base address of Diag\_Buf 1 |
| 27H | | R\_Diag\_Buf\_Ptr2 | Segment base address of Diag\_Buf 2 |
| 28H | | R\_Len\_Cntrl\_Buf1 | Length of Aux\_Buf 1 and the corresponding control buffer, for example SSA\_Buf, Prm\_Buf, Cfg\_Buf, Read\_Cfg\_Buf |
| 29H | | R\_Len\_Cntrl\_Buf2 | Length of Aux\_Buf 2 and the corresponding control buffer, for example SSA\_Buf, Prm\_Buf, Cfg\_Buf, Read\_Cfg\_Buf |
| 2AH | | R\_Aux\_Buf\_Sel | Bit array; defines the assignment of the Aux\_Buf 1 and 2 to the control buffers SSA\_Buf, Prm\_Buf, Cfg\_Buf |
| 2BH | | R\_Aux\_Buf\_Ptr1 | Segment base address of Aux\_Buf 1 |
| 2CH | | R\_Aux\_Buf\_Ptr2 | Segment base address of Aux\_Buf 2 |
| 2DH | | R\_Len\_SSA\_Data | Length of the input data in the Set\_Slave\_Address\_Buf |
| 2EH | | R\_SSA\_Buf\_Ptr | Segment base address of the Set\_Slave\_Address\_Buf |
| 2FH | | R\_Len\_Prm\_Data | Length of the input data in the Prm\_Buf |

|  |  |  |
| --- | --- | --- |
| **Address**  **Intel Mot.** | **Name Bit No.** | **Significance** |
| 30H | R\_Prm\_Buf\_Ptr | Segment base address of the Prm\_Buf |
| 31H | R\_Len\_Cfg\_Data | Length of the input data in the Cfg\_Buf |
| 32H | R\_Cfg\_Buf\_Ptr | Segment base address of the Cfg\_Buf |
| 33H | R\_Len\_Read\_Cfg\_Data | Length of the input data in the Read\_Cfg\_Buf |
| 34H | R\_Read\_Cfg\_Buf\_Ptr | Segment base address of the Read\_Cfg\_Buf |
| 35H | R\_Len\_DXB\_Link\_Buf | Length of the DXB\_Linktable |
| 36H | R\_DXB\_Link\_Buf\_Ptr | Segment base address of the DXB\_Link\_Buf |
| 37H | R\_Len\_DXB\_Status\_Buf | Length of the DXB\_Status |
| 38H | R\_DXB\_Status\_Buf\_Ptr | Segment base address of the DXB\_Status\_Buf |
| 39H | R\_Real\_No\_Add\_Change | This parameter specifies whether the Station\_Address may be changed again later. |
| 3AH | R\_Ident\_Low | The user sets the parameters for the Ident\_Number\_Low value. |
| 3BH | R\_Ident\_High | The user sets the parameters for the Ident\_Number\_High value. |
| 3CH | R\_GC\_Command | The Control\_Command of Global\_Control last received |
| 3DH | R\_Len\_Spec\_Prm\_Buf | If parameters are set for the Spec\_Prm\_Buffer\_Mode (see Mode Register 0), this cell defines the length of the Prm\_Buf. |
| 3EH | R\_DXBout\_Buf\_Ptr2 | Segment base address of DXBout\_Buf 2 |
| 3FH | R\_DXBout\_Buf\_Ptr3 | Segment base address of DXBout\_Buf 3 |

**Figure 4-4: Assignment of the Organizational Parameters**

## Mode Registers

In the VPC3+S parameter bits that access the controller directly or which the controller directly sets are combined in three Mode Registers (0, 1, 2 and 3).

### Mode Register 0

**Setting parameters for Mode Register 0 may take place in the Offline state only** (for example, after power-on). The VPC3+S may not exit the Offline state until Mode Register 0, all Control and Organizational Parameters are loaded (START\_VPC3 = 1 in Mode Register 1).

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Address** | **Bit Position** | | | | | | | | **Designation** |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| 06H  (Intel) | Freeze\_ Supported | Sync\_ Supported | Early\_Rdy | Int\_Pol | CS\_ Supported | WD\_Base | Dis\_Stop\_ Control | Dis\_Start\_ Control | Mode Reg 0  7 .. 0  See below for coding |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Address** | **Bit Position** | | | | | | | | **Designation** |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| 07H  (Intel) | Reserved | PrmCmd\_ Supported | Spec\_Clear\_ Mode \*) | Spec\_Prm\_ Buf\_Mode \*\*) | Set\_Ext\_Prm  \_Supported | User\_Time\_ Base | EOI\_Time\_ Base | DP\_Mode | Mode Reg 0  15 .. 8  See below for coding |

\*) If Spec\_Clear\_Mode = 1 (Fail Safe Mode) the VPC3+S will accept Data\_Exchange telegrams without any output data (data unit length = 0) in the state DATA-EXCH. The reaction to the outputs can be parameterized in the parameterization telegram.

\*\*) When a large number of parameters have to be transmitted from the DP-Master to the DP-Slave, the Aux-Buffer 1/2 must have the same length as the Parameter-Buffer. Sometimes this could reach the limit of the available memory in the VPC3+S. When Spec\_Prm\_Buf\_Mode = 1 the parameterization data are processed directly in this special buffer and the Aux-Buffers can be held compact.

|  |  |
| --- | --- |
| **Mode Register 0, Low-Byte, Address 06H (Intel):** | |
| bit 7 rw-0 | **Freeze\_Supported**: Freeze\_Mode support  0 = Freeze\_Mode is not supported.  1 = Freeze\_Mode is supported |
| bit 6 rw-0 | **Sync\_Supported:** Sync\_Mode support  0 = Sync\_Mode is not supported.  1 = Sync\_Mode is supported. |
| bit 5 rw-0 | **Early\_Rdy**: Early Ready  0 = Normal Ready: Ready is generated when data is valid (write) or when data has been accepted (read).  1 = Ready is generated one clock pulse earlier |
| bit 4 rw-0 | **INT\_Pol:** Interrupt Polarity  0 = The interrupt output is low-active.  1 = The interrupt output is high-active. |
| bit 3 rw-0 | **CS\_Supported:** Enable Clock Synchronization  0 = Clock Synchronization is disabled (default)  1 = Clock Synchronization is enabled |
| bit 2 rw-0 | **WD\_Base:** Watchdog Time Base  0 = Watchdog time base is 10 ms (default state)  1 = Watchdog time base is 1 ms |
| bit 1 rw-0 | **Dis\_Stop\_Control:** Disable Stopbit Control  0 = Stop bit monitoring is enabled.  1 = Stop bit monitoring is switched off  Set\_Prm telegram overwrites this memory cell in the DP\_Mode. (Refer to the user specific data.) |
| bit 0 rw-0 | **Dis\_Start\_Control:** Disable Startbit Control  0 = Monitoring the following start bit is enabled.  1 = Monitoring the following start bit is switched off  Set\_Prm telegram overwrites this memory cell in the DP\_Mode. (Refer to the user specific data.) |

**Figure 5-1: Coding of Mode Register 0, Low-Byte**

|  |  |
| --- | --- |
| **Mode Register 0, High-Byte, Address 07H (Intel):** | |
| bit 15 rw-0 | **Reserved** |
| bit 14 rw-0 | **PrmCmd\_Supported**: PrmCmd support for redundancy  0 = PrmCmd is not supported.  1 = PrmCmd is supported |
| bit 13 rw-0 | **Spec\_Clear\_Mode:** Special Clear Mode (Fail Safe Mode)  0 = No special clear mode.  1 = Special clear mode. VPC3+S will accept data telegrams with data unit = 0 |
| bit 12 rw-0 | **Spec\_Prm\_Buf\_Mode:** Special-Parameter-Buffer Mode  0 = No Special-Parameter-Buffer.  1 = Special-Parameter-Buffer mode. Parameterization data will be stored directly in the Special-Parameter-Buffer. |
| bit 11 rw-0 | **Set\_Ext\_Prm\_Supported:** Set\_Ext\_Prm telegram support  0 = SAP 53 is deactivated  1 = SAP 53 is activated |
| bit 10 rw-0 | **User\_Time\_Base:** Timebase of the cyclical User\_Time\_Clock-Interrupt  0 = The User\_Time\_Clock-Interrupt occurs every 1 ms.  1 = The User\_Time\_Clock-Interrupt occurs every 10 ms. |
| bit 9 rw-0 | **EOI\_Time\_Base:** End-of-Interrupt Timebase  0 = The interrupt inactive time is at least 1 µs long.  1 = The interrupt inactive time is at least 1 ms long |
| bit 8 rw-0 | **DP\_Mode:** DP\_Mode enable  0 = DP\_Mode is disabled.  1 = DP\_Mode is enabled. VPC3+S sets up all DP\_SAPs (default configuration!) |

**Figure 5-2: Coding of Mode Register 0, High-Byte**

### Mode Register 1

Some control bits must be changed during operation. These control bits are combined in Mode Register 1 and can be set independently of each other (Mode-Reg\_1\_S) or can be reset independently of each other (Mode- Reg\_1\_R). Separate addresses are used for setting and resetting. A logical ‘1’ must be written to the bit position to be set or reset.

For example, to set START\_VPC3 write a '1' to address 08H, in order to reset this bit, write a '1' to address 09H.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Address** | **Bit Position** | | | | | | | | **Designation** |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| 08H | Reserved | Reserved | Res\_ User\_WD | En\_Change\_ Cfg\_Buffer | User\_LEAVE- MASTER | Go\_Offline | EOI | START\_ VPC3 | Mode-Reg\_1\_S 7..0 |
| 09H | Reserved | Reserved | Res\_ User\_WD | En\_Change\_ Cfg\_Buffer | User\_LEAVE- MASTER | Go\_Offline | EOI | START\_ VPC3 | Mode-Reg\_1\_R 7..0  See below for coding |

|  |  |
| --- | --- |
| **Mode Register 1, Set, Address 08H:** | |
| bit 7 rw-0 | **Reserved** |
| bit 6 rw-0 | **Reserved** |
| bit 5 rw-0 | **Res\_User\_WD:** Resetting the User\_WD\_Timer  1 = VPC3+S sets the User\_WD\_Timer to the parameterized value User\_WD\_Value. After this action, VPC3+S sets Res\_User\_WD to ’0'. |
| bit 4 rw-0 | **En\_Change\_Cfg\_Buffer:** Enabling buffer exchange (Config-Buffer for Read\_Config-Buffer)  0 = With User\_Cfg\_Data\_Okay\_Cmd, the Config-Buffer may not be exchanged for the Read\_Config-Buffer.  1 = With User\_Cfg\_Data\_Okay\_Cmd, the Config-Buffer must be exchanged for the Read\_Config-Buffer. |
| bit 3 rw-0 | **User\_LEAVE-MASTER.** Request to the DP\_SM to go to WAIT-PRM. 1 = The user causes the DP\_SM to go to WAIT-PRM.  After this action, VPC3+ sets User\_LEAVE-MASTER to ’0’ again. |
| bit 2 rw-0 | **Go\_Offline:** Going into the Offline state  1 = After the current request ends, VPC3+S goes to the Offline state and sets Go\_Offline to ’0’ again. |
| bit 1 rw-0 | **EOI:** End-of-Interrupt  1 = VPC3+S disables the interrupt output and sets EOI to ’0‘ again. |
| bit 0 rw-0 | **Start\_VPC3:** Exiting the Offline state  1 = VPC3+S exits offline and goes to Passive\_Idle  In addition the Idle Timer and Watchdog Timer are started and ‘Go\_Offline = 0’ is set |

**Figure 5-3: Coding of Mode Register 1**

### Mode Register 2

#### Setting parameters for Mode Register 2 may take place in the Offline State only (like Mode Register 0).

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Address** | **Bit Position** | | | | | | | | **Designation** |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
|  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | Reset Value |
| 0CH | 4kB\_Mode | No\_Check\_ Prm\_Reserved | SYNC\_Pol | SYNC\_Ena | DX\_Int\_Port | DX\_Int\_Mode | No\_Check\_ GC\_Reserved | GC\_Int\_Mode | Mode Reg 2  7 .. 0 |

|  |  |
| --- | --- |
| **Mode Register 2, Address 0CH:** | |
| bit 7 w-0 | **4KB\_Mode:** size of internal RAM  0 = 2K Byte RAM (default).  1 = 4K Byte RAM |
| bit 6 w-0 | **No\_Check\_Prm\_Reserved:** disables checking of the reserved bits in DPV1\_Status\_2/3 of Set\_Prm telegram  0 = reserved bits of a Set\_Prm telegram are checked (default).  1 = reserved bits of a Set\_Prm telegram are not checked. |
| bit 5 w-0 | **SYNC\_Pol:** polarity of SYNC pulse (for Isochronous Mode only)  0 = negative polarity of SYNC pulse (default)  1 = positive polarity of SYNC pulse |
| bit 4 w-0 | **SYNC\_Ena:** enables generation of SYNC pulse (for Isochronous Mode only)  0 = SYNC pulse generation is disabled (default)  1 = SYNC pulse generation is enabled |
| bit 3 w-0 | **DX\_Int\_Port:** Port mode for DX\_Out interrupt (ignored if SYNC\_Ena set)  0 = DX\_Out interrupt is not assigned to port DATAEXCH (default).  1 = DX\_Out Interrupt (synchronized to SYNCH telegram) is assigned to port DATAEXCH. |
| bit 2 w-0 | **DX\_Int\_Mode:** Mode of DX\_out interrupt  0 = DX\_Out interrupt is only generated, if Len\_Dout\_Buf is unequal 0 (default).  1 = DX\_Out interrupt is generated after every Data\_Exchange telegram |
| bit 1 w-0 | **No\_Check\_GC\_Reserved:** Disables checking of the reserved bits in Global\_Control telegram  0 = reserved bits of a Global\_Control telegram are checked (default).  1 = reserved bits of a Global\_Control telegram are not checked. |
| bit 0 w-1 | **GC\_Int\_Mode:** Controls generation of New\_GC\_Command interrupt 0 = New\_GC\_Command interrupt is only generated, if a changed  Global\_Control telegram is received  1 = New\_GC\_Command interrupt is generated after every Global\_Control telegram (default) |

**Figure 5-4: Coding of Mode Register 2**

### Mode Register 3

#### Setting parameters for Mode Register 3 may take place in the Offline State only (like Mode Register 0).

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Address** | **Bit Position** | | | | | | | | **Designation** |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| 12H | Reserved | | | | PLL\_  Supported | En\_Chk\_SSAP | DX\_Int\_Mode \_2 | GC\_Int\_Mode \_Ext | Mode Reg 3  7 .. 0 |

|  |  |
| --- | --- |
| **Mode Register 3, Address 12H:** | |
| bit 7 w-0 | **Reserved** |
| bit 6 w-0 | **Reserved** |
| bit 5 w-0 | **Reserved** |
| bit 4 w-0 | **Reserved** |
| bit 3 w-0 | **PLL\_Supported:** Enables IsoM-PLL  0 = PLL is disabled  1 = PLL is enabled; For use of PLL, SYNC\_Ena must be set. |
| bit 2 w-0 | **En\_Chk\_SSAP:** Evaluation of Source Address Extension  0 = VPC3+ accept any value of S\_SAP  1 = VPC3+ only process the received telegram if the S\_SAP match to the default values presented by the IEC 61158 |
| bit 1 w-0 | **DX\_Int\_Mode\_2:** Mode of DX\_out interrupt  0 = DX\_Out interrupt is generated after each Data\_Exch telegram  1 = DX\_Out interrupt is only generated, if received data is not equal to current data in DX\_Out buffer of user |
| bit 0 w-0 | **GC\_Int\_Mode\_Ext:** extend GC\_Int\_Mode, works only if GC\_Int\_Mode=0  0 = GC Interrupt is only generated, if changed GC telegram is received  1 = GC Interrupt is only generated, if GC telegram with changed Control\_Command is received |

**Figure 5-5: Coding of Mode Register 3**

## Status Register

The Status Register shows the current VPC3+S status and can be read only.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Address** | **Bit Position** | | | | | | | | **Designation** |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| 04H |  | |  | | Reserved | Diag\_Flag | Reserved | Offline/ Passive\_Idle | Status-Reg |
| (Intel) | WD\_State | | DP\_State | | 7..0 |
|  |  | |  | | See below |
| 1 | 0 | 1 | 0 |
|  | for coding |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Address** | **Bit Position** | | | | | | | | **Designation** |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| 05H |  | | | |  | | | | Status-Reg |
| (Intel) | VPC3+ Release | | | | Baud Rate | | | | 15..8 |
|  |  | | | |  | | | | See below |
|  | 3 | 2 | 1 | 0 | 3 | 2 | 1 | 0 | for coding |

|  |  |
| --- | --- |
| **Status Register,Low-Byte, Address 04H (Intel):** | |
| bit 7,6 r-00 | **WD\_State 1..0:** State of the Watchdog State Machine 00 = BAUD\_SEARCH state  01 = BAUD\_CONTROL state  10 = DP\_CONTROL state  11 = Not possible |
| bit 5,4 r-00 | **DP\_State 1..0:** State of the DP State Machine 00 = WAIT-PRM state  01 = WAIT-CFG state  10 = DATA-EXCH state  11 = Not possible |
| bit 3 r-0 | **Reserved** |
| bit 2 r-0 | **Diag\_Flag:** Status of the Diagnosis-Buffer  0 = The Diagnosis-Buffer had been fetched by the DP-Master.  1 = The Diagnosis-Buffer had not been fetched by the DP-Master yet. |
| bit 1 r-0 | **Reserved** |
| bit 0 r-0 | **Offline/Passive-Idle:** Offline-/Passive\_Idle state  0 = VPC3+S is in Offline.  1 = VPC3+S is in Passive\_Idle. |

**Figure 5-6: Status Register, Low-Byte**

|  |  |
| --- | --- |
| **Status Register, High-Byte, Address 05H (Intel):** | |
| bit 15-12 r-1110 | **VPC3+-Release 3..0 :** Release number for VPC3+ 1110 |
| bit 11-8 r-1111 | **Baud Rate 3..0 :** The baud rate found by VPC3+S 0000 = 12,00 Mbit/s  0001 = 6,00 Mbit/s  0010 = 3,00 Mbit/s  0011 = 1,50 Mbit/s  0100 = 500,00 Kbit/s  0101 = 187,50 Kbit/s  0110 = 93,75 Kbit/s  0111 = 45,45 Kbit/s  1000 = 19,20 Kbit/s  1001 = 9,60 Kbit/s  1111 = after reset and during baud rate search Rest = not possible |

**Figure 5-7: Status Register, High-Byte**

## Interrupt Controller

The processor is informed about indication messages and various error events via the interrupt controller. Up to a total of 16 events are stored in the interrupt controller. The events are summed up to a common interrupt output. The controller does not have a prioritization level and does not provide an interrupt vector (not 8259A compatible!).

The controller consists of an Interrupt Request Register (IRR), an Interrupt Mask Register (IMR), an Interrupt Register (IR) and an Interrupt Acknowl- edge Register (IAR).

µP µP µP

VPC3+



S

IRR

R

IMR



S

IR

R

IAR



µP

X/INT

µP

INT\_POL

**Figure 5-8: Block Diagram of Interrupt Controller**

Each event is stored in the IRR. Individual events can be suppressed via the IMR. The input in the IRR is independent of the interrupt masks. Events that are not masked in the IMR set the corresponding IR bit and generate the X/INT interrupt via a sum network. The user can set each event in the IRR for debugging.

Each interrupt event that was processed by the microcontroller must be deleted via the IAR (except for New\_(Ext\_)Prm\_Data and New\_Cfg\_Data). A logical ‘1’ must be written on the specific bit position. If a new event and an acknowledge from the previous event are present at the IRR at the same time, the event remains stored. If the microcontroller enables a mask subsequently, it must be ensured that no prior IRR input is present. To be on the safe side, the position in the IRR must be deleted prior to the enabling of the mask.

Before leaving the interrupt routine, the microprocessor must set the ‘end of interrupt bit' (EOI = 1) in Mode Register 1. The interrupt output is switched to inactive with this edge change. If another event occurs, the interrupt output is not activated again until the interrupt inactive time of at least 1 µs or 1 ms expires. This interrupt inactive time can be set via EOI\_Time\_Base in Mode Register 0. This makes it possible to enter the interrupt routine again when an edge-triggered interrupt input is used.

The polarity of the interrupt output is parameterized via the Int\_Pol bit in Mode Register 0. After hardware reset, the output is low-active.

### Interrupt Request Register

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Address** | **Bit Position** | | | | | | | | **Designation** |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| 00H  (Intel) | DXB\_Out | New\_Ext\_ Prm\_Data | DXB\_Link\_ Error | User\_Timer\_ Clock | WD\_DP\_ CONTROL\_Timeout | Baud\_Rate\_ Detect | Go/Leave\_ DATA-EXCH | MAC\_Reset / Clock\_Sync | Int-Req-Reg 7 .. 0  See below for coding |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Address** | **Bit Position** | | | | | | | | **Designation** |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| 01H  (Intel) | FDL\_Ind | Poll\_End\_Ind | DX\_Out | Diag\_Buffer\_ Changed | New\_Prm\_ Data | New\_Cfg\_ Data | New\_SSA\_ Data | New\_GC Command | Int-Req-Reg 15 .. 8  See below for coding |

|  |  |
| --- | --- |
| **Interrupt-Request-Register, Low-Byte, Address 00H (Intel):** | |
| bit 7 rw-0 | **DXB\_Out:**  VPC3+S has received a DXB telegram and made the new output data available in the ‘N’ buffer. |
| bit 6 rw-0 | **New\_Ext\_Prm\_Data:**  The VPC3+S has received a Set\_Ext\_Prm telegram and made the data available in the Parameter-Buffer. |
| bit 5 rw-0 | **DXB\_Link\_Error:**  The Watchdog cycle is elapsed and at least one Publisher-Subscriber connection breaks down. |
| bit 4 rw-0 | **User\_Timer\_Clock:**  The time base for the User\_Timer\_Clocks is run out (1 / 10ms). |
| bit 3 rw-0 | **WD\_DP\_CONTROL\_Timeout:**  The watchdog timer expired in the DP\_CONTROL state. |
| bit 2 rw-0 | **Baud\_Rate\_Detect:**  The VPC3+S has left the BAUD\_SEARCH state and found a baud rate. |
| bit 1 rw-0 | **Go/Leave\_DATA-EXCH:**  The DP\_SM has entered or exited the DATA-EXCH state. |
| bit 0 rw-0 | **MAC\_Reset (used if CS\_Supported=0)**:  After processing the current request, the VPC3+D has entered the Offline state (by setting the Go\_Offline bit).  **Clock\_Sync (used if CS\_Supported=1)**:  The VPC3+D has received a Clock\_Value telegram or an error occurs. Further differentiation is made in the Clock\_Sync-Buffer. |

**Figure 5-9: Interrupt-Request-Register, Low-Byte**

|  |  |
| --- | --- |
| **Interrupt Request Register 0, High-Byte, Address 01H (Intel):** | |
| bit 15 rw-0 | **FDL\_Ind:**  The VPC3+S has received an acyclic service request and made the data available in an Indication-Buffer. |
| bit 14 rw-0 | **Poll\_End\_Ind:**  The VPC3+S have send the response to an acyclic service. |
| bit 13 rw-0 | **DX\_Out:**  The VPC3+S have received a Data\_Exchange telegram and made the new output data available in the ‘N’ buffer. |
| bit 12 rw-0 | **Diag\_Buffer\_Changed:**  Due to the request made by New\_Diag\_Cmd, the VPC3+S exchanged the Diagnosis-Buffers and made the old buffer available to the user again. |
| bit 11 rw-0 | **New\_Prm\_Data:**  The VPC3+S have received a Set\_Prm telegram and made the data available in the Parameter-Buffer. |
| bit 10 rw-0 | **New\_Cfg\_Data:**  The VPC3+S have received a Chk\_Cfg telegram and made the data available in the Config-Buffer. |
| bit 9 rw-0 | **New\_SSA\_Data:**  The VPC3+S have received a Set\_Slave\_Add telegram and made the data available in the Set\_Slave\_Add-Buffer. |
| bit 8 rw-0 | **New\_GC\_Command:**  The VPC3+S have received a Global\_Control telegram and stored the Control\_Command in the R\_GC\_Command RAM cell. |

**Figure 5-10: Interrupt Request Register, High-Byte**

### Interrupt Acknowledge / Mask Register

The other interrupt controller registers are assigned in the bit positions like the Interrupt Request Register.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Address** | **Register** |  | **Reset state** | **Assignment** |
| 02H / 03H | Interrupt Register (IR) | Readable only | All bits cleared |  |
| 04H / 05H | Interrupt Mask Register (IMR) | Writeable, can be changed during operation | All bits set | 1 = Mask is set and the interrupt is disabled  0 = Mask is cleared and the interrupt is enabled |
| 02H / 03H | Interrupt Acknowledge Register (IAR) | Writeable, can be changed during operation | All bits cleared | 1 = Interrupt is acknowledged and the IRR bit is cleared  0 = IRR bit remains unchanged |

**Figure 5-11: Interrupt Acknowledge / Mask Register**

The New\_(Ext\_)Prm\_Data, New\_Cfg\_Data interrupts cannot be acknowledged via the Interrupt Acknowledge Register. The relevant state machines clear these interrupts through the user acknowledgements (for example, User\_Prm\_Data\_Okay etc.).

## Watchdog Timer

The VPC3+S is able to identify the baud rate automatically. The state ma- chine is in the BAUD\_SEARCH state after each RESET and also after the Watchdog (WD) Timer has expired in the BAUD\_CONTROL state.

WD\_Timeout baudrate detected

BAUD\_SEARCH

BAUD\_CONTROL

WD\_On = 0

or WD\_DP\_CONTROL\_Timeout

WD\_On = 1

DP\_CONTROL

**Figure 5-12: Watchdog State Machine (WD\_SM)**

### Automatic Baud Rate Identification

The VPC3+S starts searching for the transmission rate using the highest baud rate. If no SD1 telegram, SD2 telegram, or SD3 telegram was received completely and without errors during the monitoring time, the search continues using the next lower baud rate.

After identifying the correct baud rate, the VPC3+S switches to the BAUD\_CONTROL state and observes the baud rate. The monitoring time can be parameterized (WD\_BAUD\_CONTROL\_Val). The watchdog uses a clock of 100 Hz (10 ms). Each telegram to its own Station\_Address received with no errors resets the Watchdog. If the timer expires, the VPC3+S switches to the BAUD\_SEARCH state again.

### Baud Rate Monitoring

The detected baud rate is permanently monitored in BAUD\_CONTROL. The Watchdog is triggered by each error-free telegram to its own Station\_Address. The monitoring time results from multiplying twice WD\_BAUD\_CONTROL\_Val (user sets this parameter) by the time base (10 ms). If the timer expires, WD\_SM again goes to BAUD\_SEARCH. If the user uses the DP protocol (DP\_Mode = 1, see Mode Register 0), the watchdog is used for the DP\_CONTROL state, after a Set\_Prm telegram was received with an enabled response time monitoring (WD\_On = 1). The watchdog timer remains in the baud rate monitoring state when the master monitoring is disabled (WD\_On = 0). The DP\_SM is not reset when the timer expires in the state BAUD\_CONTROL. That is, the DP-Slave remains in the DATA-EXCH state, for example.

### Response Time Monitoring

The DP\_CONTROL state serves as the response time monitoring of the DP-Master (Diag\_Master\_Add). The used monitoring time results from multiplying both watchdog factors and then multiplying this result with the time base (1 ms or 10 ms):

TWD = WD\_Base \* WD\_Fact\_1 \* WD\_Fact\_2 (See byte 7 of the Set\_Prm telegram.)

The user can load the two watchdog factors (WD\_Fact\_1 and WD\_Fact\_2) and the time base that represents a measurement for the monitoring time via the Set\_Prm telegram with any value between 1 and 255.

#### EXCEPTION:

**The WD\_Fact\_1 = WD\_Fact\_2 = 1 setting is not allowed. The circuit does not check this setting.**

A monitoring time between 2 ms and 650 s - independent of the baud rate - can be implemented with the allowed watchdog factors.

If the monitoring time expires, the VPC3+S goes to BAUD\_CONTROL state again and generates the WD\_DP\_CONTROL\_Timeout interrupt. In addition, the DP State Machine is reset, that is, it generates the reset states of the buffer management. This operation mode is recommended for the most applications.

If another DP-Master takes over the VPC3+S, the Watchdog State Machine either branches to BAUD\_CONTROL (WD\_On = 0) or to DP\_CONTROL (WD\_On = 1).

## DP Buffer Structure

The DP\_Mode is enabled in the VPC3+S with ‘DP\_Mode = 1’ (see Mode Register 0). In this mode, the following SAPs are permanently reserved:

Default SAP: Write and Read data (Data\_Exchange)

SAP 53: Sending extended parameter setting data (Set\_Ext\_Prm) SAP 55: Changing the Station\_Address (Set\_Slave\_Add)

SAP 56: Reading the inputs (RD\_Input) SAP 57: Reading the outputs (RD\_Output)

SAP 58: Control commands to the DP-Slave (Global\_Control) SAP 59: Reading configuration data (Get\_Cfg)

SAP 60: Reading diagnosis information (Slave\_Diag) SAP 61: Sending parameter setting data (Set\_Prm) SAP 62: Checking configuration data (Chk\_Cfg)

The DP-Slave protocol is completely integrated in the VPC3+S and is handled independently. The user must correspondingly parameterize the ASIC and process and acknowledge received messages. All SAPs are always enabled except the Default SAP, SAP 56, SAP 57 and SAP 58. The remaining SAPs are not enabled until the DP\_SM goes into the DATA- EXCH state. The user can disable SAP 55 to not permit changing the Station\_Address. The corresponding buffer pointer R\_SSA\_Buf\_Ptr must be set to ‘00H’ for this purpose.

The DP\_SAP Buffer Structure is shown in Figure 6-1. The user configures all buffers (length and buffer start) in the Offline state. During operation, the buffer configuration must not be changed, except for the length of the Dout-

/Din-Buffers.

The user may still adapt these buffers in the WAIT-CFG state after the con- figuration telegram (Chk\_Cfg). Only the same configuration may be accepted in the DATA-EXCH state.

The buffer structure is divided into the data buffers, Diagnosis-Buffers and the control buffers. Both the output data and the input data have three buffers available with the same length. These buffers are working as changing buffers. One buffer is assigned to the data transfer (D) and one buffer is assigned to the user (U). The third buffer is either in a next state

(N) or a free state (F). One of the two states is always unoccupied.

For diagnosis two Diagnosis-Buffers, that can have different lengths, are available. One Diagnosis-Buffer (D) is always assigned to the VPC3+S for sending. The other Diagnosis-Buffer (U) belongs to the user for preprocessing new diagnosis data.

Dout-Buffer



U

N



D-N changed N-U changed by VPC3+ by User

Aux 1/2

Aux 1/2

UART

D

Din-Buffer



N

U

D

Diagnosis- Buffer



D

U

Read\_Config- Buffer

changed by User

Config-Buffer

Set-Slave- Address-Buffer

Parameter- Buffer

**Figure 6-1: DP\_SAP Buffer Structure**

The VPC3+S first stores the parameter telegrams (Set\_Slave\_Add and Set\_(Ext\_)Prm) and the configuration telegram (Chk\_Cfg) in Aux-Buffer 1 or Aux-Buffer 2. If the telegrams are error-free, data is exchanged with the corresponding target buffer (Set\_Slave\_Add-Buffer, Parameter-Buffer and Config-Buffer). Each of the buffers to be exchanged must have the same length. In the R\_Aux\_Buf\_Sel parameter cell (see [Figure 6-2](#_bookmark0)) the user defines which Aux\_buffers are to be used for the telegrams mentioned

above. The Aux-Buffer 1 must always be available, Aux-Buffer 2 is optional. If the data profiles of these DP telegrams are very different (for example the length of the Set\_Prm telegram is significantly larger than the length of the other telegrams) it is suggested to make an Aux-Buffer 2 available (R\_Aux\_Buf\_Sel: Set\_Prm = 1) for this telegram. The other telegrams are then read via Aux-Buffer 1 (R\_Aux\_Buf\_Sel: Set\_Slave\_Adr = 0, Chk\_Cfg

= 0). If the buffers are too small, the VPC3+S responds with “no resources” (RR)!

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Address** | **Bit Position** | | | | | | | | **Designation** |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| 2AH | 0 | 0 | 0 | 0 | 0 | Set\_ Slave\_Add | Chk\_Cfg | Set\_Prm | R\_Aux\_Buf\_Sel  See below for coding |

|  |  |
| --- | --- |
| **R\_Aux\_Buf\_Sel, Address 2AH:** | |
| bit 7-3 | **Don’t Care:** Read as ‘0’ |
| bit 2 | **Set\_Slave\_Adr:** Set Slave Address  0 = Aux-Buffer 1  1 = Aux-Buffer 2 |
| bit 1 | **Chk\_Cfg:** Check Configuration  0 = Aux-Buffer 1  1 = Aux-Buffer 2 |
| bit 0 | **Set\_Prm:** Set (Extended) Parameter  0 = Aux-Buffer 1  1 = Aux-Buffer 2 |

**Figure 6-2: Aux-Buffer Management**

The user makes the configuration data (Get\_Cfg) available in the Read\_Config-Buffer for reading. The Read\_Config-Buffer must have the same length as the Config-Buffer.

The RD\_Input telegram is serviced from the Din-buffer in the ‘D’ state and the RD\_Output telegram is serviced from the Dout-Buffer in the ‘U’ state.

All buffer pointers are 8-bit segment addresses, because the VPC3+S have only 8-bit address registers internally. For a RAM access, VPC3+S adds an 8-bit offset address to the segment address shifted by 4 bits (result: 12-bit physical address) in case of 4K Byte RAM or shifted by 3 bits (result: 11- bit physical address) in case of 2K Byte RAM. With regard to the buffer start addresses, this specification results either in a 16-byte or in an 8-byte granularity.

## Description of the DP Services

### Set\_Slave\_Add (SAP 55)

**Sequence for the Set\_Slave\_Add service**

The user can disable this service by setting ‘R\_SSA\_Puf\_Ptr = 00H’. The Station\_Address must then be determined, for example, by reading a DIP- switch or an EEPROM and writing the address in the RAM cell R\_TS\_Adr.

There must be a non-volatile memory available (for example an external EEPROM) to support this service. It must be possible to store the Station\_Address and the Real\_No\_Add\_Change (‘True’ = FFH) parameter in this EEPROM. After each restart caused by a power failure, the user must read these values from the EEPROM again and write them to the R\_TS\_Adr und R\_Real\_No\_Add\_Change RAM registers.

If SAP55 is enabled and the Set\_Slave\_Add telegram is received correctly, the VPC3+S enters the pure data in the Aux-Buffer 1/2, exchanges the Aux-Buffer 1/2 for the Set\_Slave\_Add-Buffer, stores the entered data length in R\_Len\_SSA\_Data, generates the New\_SSA\_Data interrupt and internally stores the New\_Slave\_Add as Station\_Address and the No\_Add\_Chg as Real\_No\_Add\_Chg. The user does not need to transfer this changed parameter to the VPC3+S again. After reading the buffer, the user generates the SSA\_Buffer\_Free\_Cmd (read operation on address 14H). This makes the VPC3+S ready again to receive another Set\_Slave\_Add telegram (for example, from a different DP-Master).

The VPC3+S reacts automatically to errors.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Address** | **Bit Position** | | | | | | | | **Designation** |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| 14H | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | SSA\_Buf\_ Free\_Cmd |

|  |  |
| --- | --- |
| **SSA\_Buf\_Free\_Cmd, Address 14H:** | |
| bit 7-0 | **Don’t care:** Read as ‘0’ |

**Figure 6-3: Coding of SSA\_Buffer\_Free\_Command**

### Structure of the Set\_Slave\_Add Telegram

The net data are stored as follows in the SSA buffer:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Byte** | **Bit Position** | | | | | | | | **Designation** |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| 0 |  |  |  |  |  |  |  |  | New\_Slave\_Address |
| 1 |  |  |  |  |  |  |  |  | Ident\_Number\_High |
| 2 |  |  |  |  |  |  |  |  | Ident\_Number\_Low |
| 3 |  |  |  |  |  |  |  |  | No\_Add\_Chg |
| 4  : 243 |  |  |  |  |  |  |  |  | Rem\_Slave\_Data additional application specific data |

**Figure 6-4: Structure of the Set\_Slave\_Add Telegram**

### Set \_Prm (SAP 61)

**Parameter Data Structure**

The VPC3+S evaluates the first seven data bytes (without User\_Prm\_Data), or the first eight data bytes (with User\_Prm\_Data). The first seven bytes are specified according to the standard. The eighth byte is used for VPC3+S specific characteristics. The additional bytes are available to the application.

#### If a PROFIBUS DP extension shall be used, the bytes 7-9 are called DPV1\_Status and must be coded as described in section [7](#_bookmark1), “PROFIBUS DP Extensions”. Generally it is recommended to start the User\_Prm\_Data first with byte 10.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Byte** | **Bit Position** | | | | | | | | **Designation** |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| 0 | Lock\_ Req | Unlock\_ Req | Sync\_ Req | Freeze\_ Req | WD\_On | Reserved | Reserved | Reserved | Station Status |
| 1 |  |  |  |  |  |  |  |  | WD\_Fact\_1 |
| 2 |  |  |  |  |  |  |  |  | WD\_Fact\_2 |
| 3 |  |  |  |  |  |  |  |  | minTSDR |
| 4 |  |  |  |  |  |  |  |  | Ident\_Number\_High |
| 5 |  |  |  |  |  |  |  |  | Ident\_Number\_Low |
| 6 |  |  |  |  |  |  |  |  | Group\_Ident |
| 7 | DPV1\_  Enable | Fail\_Safe | Publisher\_ Enable | 0 | 0 | WD\_Base | Dis\_Stop\_ Control | Dis\_Start\_ Control | Spec\_User\_Prm\_Byte  /DPV1\_Status\_1 |
| 8 |  |  |  |  |  |  |  |  | DPV1\_Status\_2 |
| 9 |  |  |  |  |  |  |  |  | DPV1\_Status\_3 |
| 10  : 243 |  |  |  |  |  |  |  |  | User\_Prm\_Data |

**Figure 6-5: Format of the Set\_Prm Telegram**

|  |  |
| --- | --- |
| **Spec\_User\_Prm\_Byte / DPV1\_Status\_1:** | |
| bit 7 | **DPV1\_Enable:**  0 = DP-V1 extensions disabled (default)  1 = DP-V1 extensions enabled |
| bit 6 | **Fail\_Safe:**  0 = Fail Safe mode disabled (default)  1 = Fail Safe mode enabled |
| bit 5 | **Publisher\_Enable:**  0 = Publisher function disabled (default)  1 = Publisher function enabled |
| bit 4-3 | **Reserved:** To be parameterized with ‘0’ |
| bit 2 | **WD\_Base:** Watchdog Time Base  0 = Watchdog time base is 10 ms (default)  1 = Watchdog time base is 1 ms |
| bit 1 | **Dis\_Stop\_Control:** Disable Stop bit Control  0 = Stop bit monitoring in the receiver is enabled (default)  1 = Stop bit monitoring in the receiver is disabled |
| bit 0 | **Dis\_Start\_Control:** Disable Start bit Control  0 = Start bit monitoring in the receiver is enabled (default)  1 = Start bit monitoring in the receiver is disabled |

**Figure 6-6: Spec\_User\_Prm\_Byte / DPV1\_Status\_1**

**It is recommended not to use the DPV1\_Status bytes (bytes 7-9) for user parameter data.**

### Parameter Data Processing Sequence

In the case of a positive validation of more than seven data bytes, the VPC3+S carries out the following reaction:

The VPC3+S exchanges Aux-Buffer 1/2 (all data bytes are entered here) for the Parameter-Buffer, stores the input data length in R\_Len\_Prm\_Data and triggers the New\_Prm\_Data interrupt. The user must then check the User\_Prm\_Data and either reply with User\_Prm\_Data\_Okay\_Cmd or with User\_Prm\_Data\_Not\_Okay\_Cmd. The entire telegram is entered in this buffer. The user parameter data are stored beginning with data byte 8, or with byte 10 if DPV1\_Status bytes used.

#### The user response (User\_Prm\_Data\_Okay\_Cmd or User\_Prm\_Data\_Not\_Okay\_Cmd) clears the New\_Prm\_Data interrupt. The user cannot acknowledge the New\_Prm\_Data interrupt in the IAR register.

With the User\_Prm\_Data\_Not\_Okay\_Cmd message, relevant diagnosis bits are set and the DP\_SM branches to WAIT-PRM.

The User\_Prm\_Data\_Okay and User\_Prm\_Data\_Not\_Okay acknow- ledgements are read accesses to defined registers with the relevant signals:

|  |  |
| --- | --- |
| * User\_Prm\_Finished: | No additional parameter telegram is present. |
| * Prm\_Conflict: | An additional parameter telegram is present, processing again |
| * Not\_Allowed: | Access not permitted in the current bus state |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Address** | **Bit Position** | | | | | | | | **Designation** |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| 0EH | 0 | 0 | 0 | 0 | 0 | 0 |  |  | User\_Prm\_  Data\_Okay |
|  | | | | | | | 0 | 0 | User\_Prm\_Finished |
| 0 | 1 | Prm\_Conflict |
| 1 | 1 | Not\_Allowed |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Address** | **Bit Position** | | | | | | | | **Designation** |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| 0FH | 0 | 0 | 0 | 0 | 0 | 0 |  |  | User\_Prm\_  Data\_Not\_Okay |
|  | | | | | | | 0 | 0 | User\_Prm\_Finished |
| 0 | 1 | Prm\_Conflict |
| 1 | 1 | Not\_Allowed |

**Figure 6-7: Coding of User\_Prm\_(Not)\_Okay\_Cmd**

If another Set\_Prm telegram is supposed to be received in the meantime, the signal Prm\_Conflict is returned for the positive or negative acknowledgement of the first Set\_Prm telegram. Then the user must repeat the validation because the VPC3+S has made a new Parameter-Buffer available.

### Chk\_Cfg (SAP 62)

The user checks the correctness of the configuration data. After receiving an error-free Chk\_Cfg telegram, the VPC3+S exchanges the Aux-Buffer 1/2 (all data bytes are entered here) for the Config-Buffer, stores the input data length in R\_Len\_Cfg\_Data and generates the New\_Cfg\_Data interrupt.

Then the user has to check the User\_Config\_Data and either respond with User\_Cfg\_Data\_Okay\_Cmd or with User\_Cfg\_Data\_Not\_Okay\_Cmd. The pure data is entered in the buffer in the format of the standard.

#### The user response (User\_Cfg\_Data\_Okay\_Cmd or the User\_Cfg\_Data\_Not\_Okay\_Cmd response) clears the New\_Cfg\_Data interrupt. The user cannot acknowledge the New\_Cfg\_Data in the IAR register.

If an incorrect configuration is reported, several diagnosis bits are changed and the VPC3+S branches to state WAIT-PRM.

For a correct configuration, the transition to DATA-EXCH takes place immediately, if trigger counters for the parameter telegrams and configuration telegrams are at 0. When entering into DATA-EXCH, the VPC3+S also generates the Go/Leave\_DATA-EXCH Interrupt.

If the received configuration data from the Config-Buffer is supposed to result in a change to the Read\_Config-Buffer (contains the data for the Get\_Cfg telegram), the user have to make the new Read\_Config data available in the Read\_Config-Buffer before the User\_Cfg\_Data\_Okay\_Cmd acknowledgement, that is the user has to copy the new configuration data into the Read\_Config-Buffer.

During acknowledgement, the user receives information about whether there is a conflict or not. If another Chk\_Cfg telegram was supposed to be

received in the meantime, the user receives the Cfg\_Conflict signal during the positive or negative acknowledgement of the first Chk\_Cfg telegram. Then the user must repeat the validation, because the VPC3+S have made a new Config-Buffer available.

The User\_Cfg\_Data\_Okay\_Cmd and User\_Cfg\_Data\_Not\_Okay\_Cmd acknowledgements are read accesses to defined memory cells with the relevant Not\_Allowed, User\_Cfg\_Finished, or Cfg\_Conflict signals.

#### If the New\_Prm\_Data and New\_Cfg\_Data are supposed to be present simultaneously during start-up, the user must maintain the Set\_Prm and then the Chk\_Cfg acknowledgement sequence.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Address** | **Bit Position** | | | | | | | | **Designation** |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| 10H | 0 | 0 | 0 | 0 | 0 | 0 |  |  | User\_Cfg\_  Data\_Okay |
|  | | | | | | | 0 | 0 | User\_Cfg\_Finished |
| 0 | 1 | Cfg\_Conflict |
| 1 | 1 | Not\_Allowed |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Address** | **Bit Position** | | | | | | | | **Designation** |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| 11H | 0 | 0 | 0 | 0 | 0 | 0 |  |  | User\_Cfg\_  Data\_Not\_Okay |
|  | | | | | | | 0 | 0 | User\_Cfg\_Finished |
| 0 | 1 | Cfg\_Conflict |
| 1 | 1 | Not\_Allowed |

**Figure 6-8: Coding of User\_Cfg\_(Not)\_Okay\_Cmd**

### Slave\_Diag (SAP 60)

**Diagnosis Processing Sequence**

Two buffers are available for diagnosis. These two buffers can have different lengths. One Diagnosis-Buffer, which is sent on a diagnosis request, is always assigned to the VPC3+S. The user can pre-process new diagnosis data in the other buffer parallel. If the new diagnosis data are to be sent, the user issues the New\_Diag\_Cmd to make the request to exchange the Diagnosis-Buffers. The user receives confirmation of the buffer exchange with the Diag\_Buffer\_Changed interrupt.

When the buffers are exchanged, the internal Diag\_Flag is also set. For an activated Diag\_Flag, the VPC3+S responds during the next

Data\_Exchange with high-priority response data. That signals the DP- Master that new diagnosis data are present at the DP-Slave. The DP- Master then fetches the new diagnosis data with a Slave\_Diag telegram. Then the Diag\_Flag is cleared again. However, if the user signals ‘Diag.Stat\_Diag = 1’ (that is static diagnosis, see the structure of the Diagnosis-Buffer), the Diag\_Flag still remains activated after the relevant DP-Master has fetched the diagnosis. The user can poll the Diag\_Flag in the Status Register to find out whether the DP-Master has already fetched the diagnosis data before the old data is exchanged for the new data.

#### According to IEC 61158, Static Diagnosis should only be used during start-up.

Status coding for the diagnosis buffers is stored in the Diag\_Buffer\_SM control parameter. The user can read this cell with the possible codings for both buffers: User, VPC3+, or VPC3+\_Send\_Mode.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Address** | **Bit Position** | | | | | | | | **Designation** |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| 0CH | 0 | 0 | 0 | 0 | Diag\_Buf2 | | Diag\_Buf1 | | Diag\_Buffer\_SM |

|  |  |
| --- | --- |
| **Diag\_Buffer\_SM, Address 0CH:** | |
| bit 7-4 | **Don’t care:** Read as ‘0’ |
| bit 3-2 | **Diag\_Buf2:** Assignment of Diagnosis Buffer 2 00 = Nil  01 = User  10 = VPC3+  11 = VPC3\_Send\_Mode |
| bit 1-0 | **Diag\_Buf1:** Assignment of Diagnosis Buffer 1 00 = Nil  01 = User  10 = VPC3+  11 = VPC3\_Send\_Mode |

**Figure 6-9: Diagnosis Buffer Assignment**

The New\_Diag\_Cmd is also a read access to a defined control parameter indicating which Diagnosis-Buffer belongs to the user after the exchange or whether both buffers are currently assigned to the VPC3+S (No\_Buffer, Diag\_Buf1, Diag\_Buf2).

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Address** | **Bit Position** | | | | | | | | **Designation** |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| 0DH | 0 | 0 | 0 | 0 | 0 | 0 |  |  | New\_Diag\_  Buffer\_Cmd |
|  | | | | | | | 0 | 0 | No\_Buffer |
| 0 | 1 | Diag\_Buf1 |
| 1 | 0 | Diag\_Buf2 |

**Figure 6-10: Coding of New\_Diag\_Cmd**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Byte** | **Bit Position** | | | | | | | | **Designation** |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| 0 |  |  |  |  |  | Ext\_Diag\_ Overflow | Stat\_Diag | Ext\_Diag |  |
| 1 |  |  |  |  |  |  |  |  |  |
| 2 |  |  |  |  |  |  |  |  |  |
| 3 |  |  |  |  |  |  |  |  |  |
| 4 |  |  |  |  |  |  |  |  |  |
| 5 |  |  |  |  |  |  |  |  |  |
| 6  :  n | user input | | | | | | | | Ext\_Diag\_Data (n = max. 243) |

**Figure 6-11: Format of the Diagnosis-Buffer**

The Ext\_Diag\_Data must be entered into the buffers after the VPC3+S internal diagnosis data. Three different formats are possible here: device- related, ID-related and port-related. If PROFIBUS DP extensions shall be used, the device-related diagnosis is substituted by alarm and status messages. In addition to the Ext\_Diag\_Data, the buffer length also includes the VPC3+S diagnosis bytes (R\_Len\_Diag\_Buf 1, R\_Len\_Diag\_Buf 2).

### Write\_Read\_Data / Data\_Exchange (Default\_SAP) Writing Outputs

The VPC3+S writes the received output data in the 'D' buffer. After an error- free receipt, the VPC3+S shifts the newly filled buffer from ‘D’ to ‘N'. In addition, the DX\_Out interrupt is generated. The user now fetches the current output data from ‘N’. The buffer changes from ‘N’ to ‘U’ with the Next\_Dout\_Buffer\_Cmd, so that the current data can be transmitted to the application by a RD\_Output request from a DP-Master.

If the user’s evaluation cycle time is shorter than the bus cycle time, the user does not find any new buffers with the next Next\_Dout\_Buffer\_Cmd in ‘N'. Therefore, the buffer exchange is omitted. At a 12 Mbit/s baud rate, it is more likely, however, that the user’s evaluation cycle time is larger than the bus cycle time. This makes new output data available in ‘N’ several times before the user fetches the next buffer. It is guaranteed, however, that the user receives the data last received.

For power-on, LEAVE-MASTER and the Global\_Control telegram with ‘Clear\_Data = 1’, the VPC3+S deletes the ‘D’ buffer and then shifts it to ‘N'. This also takes place during power-up (entering the WAIT-PRM state). If the user fetches this buffer, he receives U\_Buffer\_Cleared during the Next\_Dout\_Buffer\_Cmd. If the user is supposed to enlarge the output data buffer after the Chk\_Cfg telegram, the user must delete this deviation in the 'N' buffer himself (possible only during the start-up phase in the WAIT-CFG state).

If ‘Diag.Sync\_Mode = 1’, the ‘D’ buffer is filled but not exchanged with the Data\_Exchange telegram. It is exchanged at the next Sync or Unsync command sent by Global\_Control telegram.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Address** | **Bit Position** | | | | | | | | **Designation** |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| 0AH | F | | U | | N | | D | | Dout\_Buffer\_SM |

|  |  |
| --- | --- |
| **Dout\_Buffer\_SM, Address 0AH:** | |
| bit 7-6 | **F:** Assignment of the F-Buffer 00 = Nil  01 = Dout\_Buf\_Ptr1  10 = Dout\_Buf\_Ptr2  11 = Dout\_Buf\_Ptr3 |
| bit 5-4 | **U:** Assignment of the U-Buffer 00 = Nil  01 = Dout\_Buf\_Ptr1  10 = Dout\_Buf\_Ptr2  11 = Dout\_Buf\_Ptr3 |
| bit 3-2 | **N:** Assignment of the N-Buffer 00 = Nil  01 = Dout\_Buf\_Ptr1  10 = Dout\_Buf\_Ptr2  11 = Dout\_Buf\_Ptr3 |
| bit 1-0 | **D:** Assignment of the D-Buffer 00 = Nil  01 = Dout\_Buf\_Ptr1  10 = Dout\_Buf\_Ptr2  11 = Dout\_Buf\_Ptr3 |

**Figure 6-12: Dout-Buffer Management**

When reading the Next\_Dout\_Buffer\_Cmd the user gets the information which buffer (‘U’ buffer) belongs to the user after the change, or whether a change has taken place at all.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Address** | **Bit Position** | | | | | | | | **Designation** |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| 0BH | 0 | 0 | 0 | 0 | U\_Buffer\_ Cleared | State\_U\_ Buffer | Ind\_U\_ Buffer | | Next\_Dout\_ Buf\_Cmd  See coding below |

|  |  |
| --- | --- |
| **Next\_Dout\_Buf\_Cmd, Address 0BH:** | |
| bit 7-4 | **Don’t care:** Read as ‘0’ |
| bit 3 | **U\_Buffer\_Cleared:** User-Buffer-Cleared Flag  0 = U buffer contains data  1 = U buffer is cleared |
| bit 2 | **State\_U\_Buffer:** State of the User-Buffer  0 = no new U buffer  1 = new U buffer |
| bit 1-0 | **Ind\_U\_Buffer:** Indicated User-Buffer 01 = Dout\_Buf\_Ptr1  10 = Dout\_Buf\_Ptr2  11 = Dout\_Buf\_Ptr3 |

**Figure 6-13: Coding of Next\_Dout\_Buf\_Cmd**

The user must clear the ‘U’ buffer during initialization so that defined (cleared) data can be sent for a RD\_Output telegram before the first data cycle.

### Reading Inputs

The VPC3+S sends the input data from the ‘D’ buffer. Prior to sending, the VPC3+S fetches the Din-Buffer from ‘N’ to ‘D'. If no new buffer is present in ‘N', there is no change.

The user makes the new data available in ‘U’. With the New\_Din\_Buffer\_Cmd, the buffer changes from ‘U’ to ‘N’. If the user’s preparation cycle time is shorter than the bus cycle time, not all new input data are sent, but just the most current. At a 12 Mbit/s baud rate, it is more likely, however, that the user’s preparation cycle time is larger than the bus cycle time. Then the VPC3+S sends the same data several times in succession.

During start-up, the VPC3+S does not go to DATA-EXCH before all parameter telegrams and configuration telegrams have been acknowledged.

If ‘Diag.Freeze\_Mode = 1’, there is no buffer change prior to sending.

The user can read the status of the state machine cell with the following codings for the four states: Nil, Dout\_Buf\_Ptr1, Dout\_Buf\_Ptr2 and Dout\_Buf\_Ptr3. The pointer for the current data is in the 'N' state.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Address** | **Bit Position** | | | | | | | | **Designation** |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| 08H | F | | U | | N | | D | | Din\_Buffer\_SM |

|  |  |
| --- | --- |
| **Din\_Buffer\_SM, Address 08H:** | |
| bit 7-6 | **F:** Assignment of the F-Buffer 00 = Nil  01 = Din\_Buf\_Ptr1  10 = Din\_Buf\_Ptr2  11 = Din\_Buf\_Ptr3 |
| bit 5-4 | **U:** Assignment of the U-Buffer 00 = Nil  01 = Din\_Buf\_Ptr1  10 = Din\_Buf\_Ptr2  11 = Din\_Buf\_Ptr3 |
| bit 3-2 | **N:** Assignment of the N-Buffer 00 = Nil  01 = Din\_Buf\_Ptr1  10 = Din\_Buf\_Ptr2  11 = Din\_Buf\_Ptr3 |
| bit 1-0 | **D:** Assignment of the D-Buffer 00 = Nil  01 = Din\_Buf\_Ptr1  10 = Din\_Buf\_Ptr2  11 = Din\_Buf\_Ptr3 |

**Figure 6-14: Din-Buffer Management**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Address** | **Bit Position** | | | | | | | | **Designation** |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| 09H | 0 | 0 | 0 | 0 | 0 | 0 |  |  | New\_Din\_Buf\_Cmd |
|  | | | | | | | 0 | 1 | Din\_Buf\_Ptr1 |
| 1 | 0 | Din\_Buf\_Ptr2 |
| 1 | 1 | Din\_Buf\_Ptr3 |

**Figure 6-15: Coding of New\_Din\_Buf\_Cmd**

### User\_Watchdog\_Timer

After start-up (DATA-EXCH state), it is possible that the VPC3+S continually answers Data\_Exchange telegrams without the user fetching the received Dout-Buffers or making new Din-Buffers available. If the user processor ‘hangs up' the DP-Master would not receive this information. Therefore, a User\_Watchdog\_Timer is implemented in the VPC3+S.

This User\_WD\_Timer is an internal 16-bit RAM cell that is started from a user parameterized value R\_User\_WD\_Value and is decremented by the VPC3+S with each received Data\_Exchange telegram. If the timer reaches the value 0000H, the VPC3+S goes to the WAIT-PRM state and the DP\_SM carries out a LEAVE-MASTER. The user must cyclically set this timer to its start value. Therefore, ‘Res\_User\_WD = 1’ must be set in Mode Register 1. Upon receipt of the next Data\_Exchange telegram, the VPC3+S again loads the User\_WD\_Timer to the parameterized value R\_User\_WD\_Value and sets ‘Res\_User\_WD = 0’ (Mode Register 1). During power-up, the user must also set ‘Res\_User\_WD = 1’, so that the User\_WD\_Timer is set to its parameterized value.

### Global\_Control (SAP 58)

The VPC3+S processes the Global\_Control telegrams like already described.

The first byte of a valid Global\_Control is stored in the R\_GC\_Command RAM cell. The second telegram byte (Group\_Select) is processed internally.

The interrupt behavior regarding to the reception of a Global\_Control telegram can be configured via bit 8 of Mode Register 2. The VPC3+S either generates the New\_GC\_Control interrupt after each receipt of a Global\_Control telegram (default) or just in case if the Global\_Control differs from the previous one.

The R\_GC\_Command RAM cell is not initialized by the VPC3+S. Therefore the cell has to be preset with 00H during power-up. The user can read and evaluate this cell.

In order to use Sync and Freeze, these functions must be enabled in the Mode Register 0.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Address** | **Bit Position** | | | | | | | | **Designation** |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| 3CH | Reserved | Reserved | Sync | Unsync | Freeze | Unfreeze | Clear\_Data | Reserved | R\_GC\_ Command  See below for coding |

|  |  |
| --- | --- |
| **R\_GC\_Command, Address 3CH:** | |
| bit 7-6 | **Reserved** |
| bit 5 | **Sync:**  The output data transferred with a Data\_Exchange telegram is changed from ‘D’ to ‘N’. The following transferred output data is kept in ‘D’ until the next Sync command is given. |
| bit 4 | **Unsync:**  The Unsync command cancels the Sync command. |
| bit 3 | **Freeze:**  The input data is fetched from ‘N’ to ‘D’ and „frozen“. New input data is not fetched again until the DP-Master sends the next Freeze command. |
| bit 2 | **Unfreeze:**  The Unfreeze command cancels the Freeze command. |
| bit 1 | **Clear\_Data:**  With this command, the output data is deleted in ‘D’ and is changed to ‘N’. |
| bit 0 | **Reserved** |

**Figure 6-16: Format of the Global\_Control Telegram**

### RD\_Input (SAP 56)

The VPC3+S fetches the input data like it does for the Data\_Exchange telegram. Prior to sending, ‘N’ is shifted to ‘D', if new input data are available in ‘N'. For ‘Diag.Freeze\_Mode = 1', there is no buffer change.

### RD\_Output (SAP 57)

The VPC3+S fetches the output data from the Dout\_Buffer in ‘U’. The user must preset the output data with ‘0’ during start-up so that no invalid data can be sent here. If there is a buffer change from ‘N’ to ‘U’ (through the

Next\_Dout\_Buffer\_Cmd) between the first call-up and the repetition, the new output data is sent during the repetition.

### Get\_Cfg (SAP 59)

The user makes the configuration data available in the Read\_Config-Buffer. For a change in the configuration after the Chk\_Cfg telegram, the user writes the changed data in the Config-Buffer, sets ‘En\_Change\_Cfg\_buffer = 1’ (see Mode Register 1) and the VPC3+S then exchanges the Config-Buffer for the Read\_Config-Buffer. If there is a change in the configuration data during operation (for example, for a modular DP systems), the user must return with Go\_Offline command (see Mode Register 1) to WAIT-PRM.

## Set\_(Ext\_)Prm (SAP 53 / SAP 61)

The PROFIBUS DP extensions require three bytes to implement the new parameterization function. The bits of the Spec\_User\_Prm\_Byte are included.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Byte** | **Bit Position** | | | | | | | | **Designation** |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| 0 |  |  |  |  |  |  |  |  |  |
| : |
| 6 |
| 7 | DPV1\_  Enable | Fail\_Safe | Publisher\_ Enable | Reserved | Reserved | WD\_Base | Dis\_Stop\_ Control | Dis\_Start\_ Control | DPV1\_Status\_1 |
| 8 | Enable\_ Pull\_Plug\_Alarm | Enable\_ Process\_Alarm | Enable\_ Diagnostic\_Alarm | Enable\_  Manufacturer\_ Specific\_Alarm | Enable\_ Status\_Alarm | Enable\_ Update\_Alarm | 0 | Chk\_Cfg\_Mode | DPV1\_Status\_2 |
| 9 | PrmCmd | 0 | 0 | IsoM\_Req | Prm\_ Structure | Alarm\_Mode | | | DPV1\_Status\_3 |
| 10 |  |  |  |  |  |  |  |  |  |
| : | User\_Prm\_Data |
| 243 |  |

**Figure 7-1: Set\_Prm with DPV1\_Status bytes**

#### If the extensions are used, the bit Spec\_Clear\_Mode in Mode Register 0 serves as Fail\_Safe\_required. Therefore it is used for a comparison with the bit Fail\_Safe in parameter telegram. Whether the DP-Master supports the Fail\_Safe mode or not is indicated by the telegram bit. If the DP-Slave requires Fail\_Safe but the DP-Master doesn’t the Prm\_Fault bit is set.

If the VPC3+S should be used for DXB, IsoM or redundancy mode, the parameterization data must be packed in a Structured\_Prm\_Data block to distinguish between the User\_Prm\_Data. The bit Prm\_Structure indicates this.

If redundancy should be supported, the PrmCmd\_Supported bit in Mode Register 0 must be set.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Byte** | **Bit Position** | | | | | | | | **Designation** |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| 0 |  |  |  |  |  |  |  |  | Structured\_Length |
| 1 |  |  |  |  |  |  |  |  | Structure\_Typ |
| 2 |  |  |  |  |  |  |  |  | Slot\_Number |
| 3 |  |  |  |  |  |  |  |  | Reserved |
| 4  : 243 |  |  |  |  |  |  |  |  | User\_Prm\_Data |

**Figure 7-2 : Format of the Structured\_Prm\_Data block**

Additional to the Set\_Prm telegram (SAP 61) a Set\_Ext\_Prm (SAP 53) telegram can be used for parameterization. This service is only available in state WAIT-CFG after the reception of a Set\_Prm telegram and before the reception of a Chk\_Cfg telegram. The new Set\_Ext\_Prm telegram simply consists of Structured\_Prm\_Data blocks.

The new service uses the same buffer handling as described by Set\_Prm. By means of the New\_Ext\_Prm\_Data interrupt the user can recognize which kind of telegram is entered in the Parameter-Buffer. Additional the SAP 53 must be activated by Set\_Ext\_Prm\_Supported bit in Mode Register 0.

**The Aux-Buffer for the Set\_Ext\_Prm is the same as the one for Set\_Prm and has to be different from the Chk\_Cfg Aux-Buffer. Furthermore the Spec\_Prm\_Buf\_Mode in Mode Register 0 must not be used together with SAP 53.**

## PROFIBUS DP-V1

### Acyclic Communication Relationships

The VPC3+S supports acyclic communication as described in the DP-V1 specification. Therefore a memory area is required which contains all SAPs needed for the communication. The user must do the initialization of this area (SAP-List) in Offline state. Each entry in the SAP-List consists of 7 bytes. The pointer at address 17H contains the segment base address of the first element of the SAP-List. The last element in the list is always indicated with FFH. If the SAP-List shall not be used, the first entry must be FFH, so the pointer at address 17H must point to a segment base address location that contains FFH.

The new communication features are enabled with DPV1\_Enable in the Set\_Prm telegram.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Byte** | **Bit Position** | | | | | | | | **Designation** |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| 0 | Response  \_Sent | SAP\_Number | | | | | | | SAP\_Number |
| 1 |  |  |  |  |  |  |  |  | Request\_SA |
| 2 |  |  |  |  |  |  |  |  | Request\_SSAP |
| 3 |  |  |  |  |  |  |  |  | Service\_Supported |
| 4 |  |  |  |  |  |  |  |  | Ind\_Buf\_Ptr[0] |
| 5 |  |  |  |  |  |  |  |  | Ind\_Buf\_Ptr[1] |
| 6 |  |  |  |  |  |  |  |  | Resp\_Buf\_Ptr |

|  |  |
| --- | --- |
| **SAP-List entry:** | |
| Byte 0 | **Response\_Sent:** Response-Buffer sent  0 = no Response sent  1 = Response sent  **SAP\_Number:** 0 – 51 |
| Byte 1 | **Request\_SA:** The source address of a request is compared with this value. At differences, the VPC3+S response with “no service activated” (RS). The default value for this entry is 7FH. |
| Byte 2 | **Request\_SSAP:** The source SAP of a request is compared with this value. At differences, the VPC3+S response with “no service activated” (RS). The default value for this entry is 7FH. |
| Byte 3 | **Service\_Supported:** Indicates the permitted FDL service. 00 = all FDL services allowed |
| Byte 4 | **Ind\_Buf\_Ptr[0]:** pointer to Indication-Buffer 0 |
| Byte 5 | **Ind\_Buf\_Ptr[1]:** pointer to Indication-Buffer 1 |
| Byte 6 | **Resp\_Buf\_Ptr:** pointer to Response-Buffer |

**Figure 7-3: SAP-List entry**

In addition an Indication- and Response-Buffer are needed. Each buffer consists of a 4-byte header for the buffer management and a data block of configurable length.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Byte** | **Bit Position** | | | | | | | | **Designation** |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| 0 | USER | IND | RESP | INUSE |  |  |  |  | Control |
| 1 |  |  |  |  |  |  |  |  | Max\_Length |
| 2 |  |  |  |  |  |  |  |  | Length |
| 3 |  |  |  |  |  |  |  |  | Function Code |

|  |  |
| --- | --- |
| **SAP-List entry:** | |
| Byte 0 | **Control:** bits for buffer management USER buffer assigned to user  IND indication data included in buffer  RESP response data included in buffer INUSE buffer assigned to VPC3+S |
| Byte 1 | **Max\_Length:** length of buffer |
| Byte 2 | **Length:** length of data included in buffer |
| Byte 3 | **Function Code:** function code of the telegram |

**Figure 7-4: Buffer Header**

### Processing Sequence

A received telegram is compared with the values in the SAP-List. If this check is positive, the telegram is stored in an Indication-Buffer with the INUSE bit set. In case of any deviations the VPC3+S responses with “no service activated” (RS) or if no free buffer is available with “no resource” (RR). After finishing the processing of the incoming telegram, the INUSE bit is reset and the bits USER and IND are set by VPC3+S. Now the FDL\_Ind interrupt is generated. Polling telegrams do not produce interrupts. The RESP bit indicates response data, provided by the user in the Response- Buffer. The Poll\_End\_Ind interrupt is set after the Response-Buffer is sent. Also bits RESP and USER are cleared.

|  |  |  |
| --- | --- | --- |
| **DP-Master** | **PROFIBUS** | **DP-Slave** |
|  | Request to acyclic SAP -> | fill Indication-Buffer |
| <- short acknowledgement (SC) |  |
| Polling telegram to acyclic SAP -> | process data |
| <- short acknowledgement (SC) |
| : |  |
| : |  |
| : | update Response- Buffer |
| Polling telegram to acyclic SAP -> |
| <- Response from acyclic |  |

**Figure 7-5: acyclic communication sequence**

|  |  |
| --- | --- |
| **VPC3+S** | **Firmware** |
| set Request\_SA / Request\_SSA set INUSE in Control of Ind\_Buf write data in Ind\_Buf  clear INUSE and set USER and IND in Control of Ind\_Buf  set FDL\_Ind interrupt | |
|  | clear FDL\_Ind interrupt search for Ind\_Buf with IND = 1  read Ind\_Buf clear IND in Control of Ind\_Buf write Response in Resp\_Buf  set RESP in Control of Resp\_Buf |
| check on RESP = 1 read Resp\_Buf  clear RESP and USER in Control of Resp\_Buf set Response\_Sent  set Poll\_End\_Ind interrupt | |
|  | clear Poll\_End\_Ind interrupt search for SAP with Response\_Sent = 1  clear Response\_Sent |

**Figure 7-6: FDL-Interface of VPC3+S (e.g. same Buffer for Indication and Response)**

### Diagnosis Model

The format of the device related diagnosis data depends on the GSD keyword DPV1\_Slave in the GSD. If 'DPV1\_Slave = 1', alarm and status messages are used in diagnosis telegrams. Status messages are required by the Data eXchange Broadcast service, for example. Alarm\_Ack is used as the other acyclic services.

## PROFIBUS DP-V2

### DXB (Data eXchange Broadcast)

The DXB mechanism enables a fast slave-to-slave communication. A DP- Slave that holds input data significant for other DP-Slaves, works as a Publisher. The Publisher can handle a special kind of Data\_Exchange request from the DP-Master and sends its answer as a broadcast telegram. Other DP-Slaves which are parameterized as Subscribers can monitor this telegram. A link is opened to the Publisher if the address of the Publisher is registered in the linktable of the Subscriber. If the link has been established correctly, the Subscriber can receive the input data from the Publisher.

DP-Master (Class1)

Dout Din

Request (FC=7) Response (DA=127)



Data Exchange with DP-Master (Class 1)

Dout

Din

Data Exchange with filtered

DP-Master (Class 1) Broadcast (Input) Data

from Publisher

Dout

Din

DXBout

DP-Slave (Publisher) DP-Slave (Subscriber)



Link

**Figure 7-7 : Overview DXB**

The VPC3+S can handle a maximum of 29 links simultaneously.

### Publisher

A Publisher is activated with 'Publisher\_Enable = 1' in DPV1\_Status\_1. The time minTSDR must be set to 'TID1 = 37 tbit + 2 TSET + TQUI'.

All Data\_Exchange telegrams containing the function code 7 (Send and Request Data Multicast) are responded with destination address 127. If Publisher mode is not enabled, these requests are ignored.

### Subscriber

A Subscriber requires information about the links to its Publishers. These settings are contained in a DXB Linktable or DXB Subscribertable and transferred via the Structured\_Prm\_Data in a Set\_Prm or Set\_Ext\_Prm telegram. Each Structured\_Prm\_Data is treated like the User\_Prm\_Data and therefore to be evaluated by the user. From the received data the user has to generate DXB\_Link\_Buf and DXB\_Status\_Buf entries. The watch- dog must be enabled to make use of the monitoring mechanism. The user must check this.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Byte** | **Bit Position** | | | | | | | | **Designation** |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| 0 |  |  |  |  |  |  |  |  | Structured\_Length |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | Structure\_Type |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | Slot\_Number |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | Reserved |
| 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | Version |
| 5 |  |  |  |  |  |  |  |  | Publisher\_Addr |
| 6 |  |  |  |  |  |  |  |  | Publisher\_Length |
| 7 |  |  |  |  |  |  |  |  | Sample\_Offset |
| 8 |  |  |  |  |  |  |  |  | Sample\_Length |
| 9  : 120 |  |  |  |  |  |  |  |  | further link entries |

**Figure 7-8: Format of the Structured\_Prm\_Data with DXB Linktable (specific link is grey scaled)**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Byte** | **Bit Position** | | | | | | | | **Designation** |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| 0 |  |  |  |  |  |  |  |  | Structured\_Length |
| 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | Structure\_Type |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | Slot\_Number |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | Reserved |
| 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | Version |
| 5 |  |  |  |  |  |  |  |  | Publisher\_Addr |
| 6 |  |  |  |  |  |  |  |  | Publisher\_Length |
| 7 |  |  |  |  |  |  |  |  | Sample\_Offset |
| 8 |  |  |  |  |  |  |  |  | Dest\_Slot\_Number |
| 9 |  |  |  |  |  |  |  |  | Offset\_Data\_Area |
| 10 |  |  |  |  |  |  |  |  | Sample\_Length |
| 11  : 120 |  |  |  |  |  |  |  |  | further link entries |

**Figure 7-9: Format of the Structured\_Prm\_Data with DXB Subscribertable (specific link is grey scaled)**

The user must copy the link entries of DXB Linktable or DXB Subscribertable, without Dest\_Slot\_Number and Offset\_Data\_Area, into the DXB\_Link\_Buf and set R\_Len\_DXB\_Link\_Buf. Also the user must enter the default status message in DXB\_Status\_Buf with the received links and write the appropriate values to R\_Len\_DXB\_Status\_Buf. After that, the parameterization interrupt can be acknowledged.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Byte** | **Bit Position** | | | | | | | | **Designation** |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| 0 | 0 | 0 | Block\_Length | | | | | | Header\_Byte |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | Status\_Type |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | Slot\_Number |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | Status\_Specifier |
| 4 |  |  |  |  |  |  |  |  | Publisher\_Addr |
| 5 | Link\_ Status | Link\_ Error | 0 | 0 | 0 | 0 | 0 | Data\_ Exist | Link\_Status |
| 6  : 61 |  |  |  |  |  |  |  |  | further link entries |

|  |  |
| --- | --- |
| **Link\_Status:** | |
| bit 7 | **Link\_Status :**  1 = active, valid data receipt during last monitoring period  0 = not active, no valid data receipt during last monitoring period (DEFAULT) |
| bit 6 | **Link\_Error:**  0 = no faulty Broadcast data receipt (DEFAULT)  1 = wrong length, error occurred during reception |
| bit 0 | **Data\_Exist:**  0 = no correct Broadcast data receipt during current monitoring period (DEFAULT)  1 = error free reception of Broadcast data during current monitoring period |

**Figure 7-10: DXB\_Link\_Status\_Buf (specific link is grey scaled)**

### Processing Sequence

The VPC3+S processes DXBout-Buffers like the Dout-Buffers. The only difference is that the DXBout-Buffers are not cleared by the VPC3+S.

The VPC3+S writes the received and filtered broadcast data in the 'D' buffer. The buffer contains also the Publisher\_Address and the Sample\_Length. After error-free receipt, the VPC3+S shifts the newly filled buffer from 'D' to 'N'. In addition, the DXBout interrupt is generated. The user now fetches the current output data from 'N'. The buffer changes from 'N' to 'U' with the Next\_DXBout\_Buffer\_Cmd.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Byte** | **Bit Position** | | | | | | | | **Designation** |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| 0 |  |  |  |  |  |  |  |  | Publisher\_Addr |
| 1 |  |  |  |  |  |  |  |  | Sample\_Length |
| 2  : 246 |  |  |  |  |  |  |  |  | Sample\_Data |

**Figure 7-11: DXBout-Buffer**

When reading the Next\_DXBout\_buffer\_Cmd the user gets the information which buffer ('U' buffer) is assigned to the user after the change, or whether a change has taken place at all.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Address** | **Bit Position** | | | | | | | | **Designation** |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| 12H | F | | U | | N | | D | | DXBout\_Buffer\_SM |

|  |  |
| --- | --- |
| **DXBout\_Buffer\_SM, Address 0AH:** | |
| bit 7-6 | **F:** Assignment of the F-Buffer 00 = Nil  01 = DXBout\_Buf\_Ptr1  10 = DXBout\_Buf\_Ptr2  11 = DXBout\_Buf\_Ptr3 |
| bit 5-4 | **U:** Assignment of the U-Buffer 00 = Nil  01 = DXBout\_Buf\_Ptr1  10 = DXBout\_Buf\_Ptr2  11 = DXBout\_Buf\_Ptr3 |
| bit 3-2 | **N:** Assignment of the N-Buffer 00 = Nil  01 = DXBout\_Buf\_Ptr1  10 = DXBout\_Buf\_Ptr2  11 = DXBout\_Buf\_Ptr3 |
| bit 1-0 | **D:** Assignment of the D-Buffer 00 = Nil  01 = DXBout\_Buf\_Ptr1  10 = DXBout\_Buf\_Ptr2  11 = DXBout\_Buf\_Ptr3 |

**Figure 7-12: DXBout-Buffer Management**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Address** | **Bit Position** | | | | | | | | **Designation** |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| 13H | 0 | 0 | 0 | 0 | 0 | State\_U\_ Buffer | Ind\_U\_ Buffer | | Next\_DXBout\_ Buf\_Cmd  See coding below |

|  |  |
| --- | --- |
| **Next\_DXBout\_Buf\_Cmd, Address 0BH:** | |
| bit 7-3 | **Don’t care:** Read as ‘0’ |
| bit 2 | **State\_U\_Buffer:** State of the User-Buffer  0 = no new U buffer  1 = new U buffer |
| bit 1-0 | **Ind\_U\_Buffer:** Indicated User-Buffer 01 = DXBout\_Buf\_Ptr1  10 = DXBout\_Buf\_Ptr2  11 = DXBout\_Buf\_Ptr3 |

**Figure 7-13: Coding of Next\_DXBout\_Buf\_Cmd**

### Monitoring

After receiving the DXB data the Link\_Status in DXB\_Status\_Buf of the corresponding Publisher is updated. In case of an error the bit Link\_Error is set. If the processing is finished without errors, the bit Data\_Exist is set.

In state DATA-EXCH the links are monitored in intervals defined by the parameterized watchdog time. After the monitoring time runs out, the VPC3+S evaluates the Link\_Status of each Publisher and updates the bit Link\_Status. The timer restarts again automatically.

|  |  |  |  |
| --- | --- | --- | --- |
| **Event** | **Link\_**  **Status** | **Link\_**  **Error** | **Data\_**  **Exist** |
| valid DXB data receipt |  | 0 | 1 |
| faulty DXB data receipt | 0 | 1 | 0 |
| WD\_Time elapsed AND Data\_Exist = 1 | 1 | 0 | 0 |
| WD\_Time elapsed AND Link\_Error = 1 | 0 | 0 | 0 |

**Figure 7-14: Link\_Status handling**

**To enable the monitoring of Publisher-Subscriber links the watchdog timer must be enabled in the Set\_Prm telegram. The user must check this.**

### IsoM (Isochronous Mode)

The IsoM synchronizes DP-Master, DP-Slave and DP-Cycle. The isochro- nous cycle time starts with the transmission of the SYNCH telegram by the IsoM master. If the IsoM support of the VPC3+S is enabled, a synchroniza- tion signal at Pin C4 (SYNC) is generated by each reception of a SYNCH telegram. The SYNCH telegram is a special coded Global\_Control request.

. . .

|  |  |  |
| --- | --- | --- |
| SYNCH  mesage | Cyclic Service | . . . Cyclic Service  Cyclic Part |
|  |  |
|  |

|  |  |
| --- | --- |
|  | Acyclic Service |
|  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Acyclic Service | Token | Spare Time | SYNCH  mesage |
|  |  | | |

Acyclic Part

Cycle Time (TDP)

**Figure 7-15: Telegram sequences in IsoM with one DP-Master (Class 1)**

Two operation modes for cyclic synchronization are available in the VPC3+S:

1. Isochronous Mode: Each SYNCH telegram causes an impulse on the SYNC output and a New\_GC\_Command interrupt. In this mode the IsoM-PLL can be used for compensation of jitter and loss of synchroni- zation.
2. Simple Sync Mode: A Data\_Exchange telegram no longer causes a DX\_Out interrupt immediately, rather the event is stored in a flag. By a following SYNCH message reception, the DX\_Out interrupt and a synchronization signal are generated at the same time. Additionally a New\_GC\_Command interrupt is produced, as the SYNCH telegram behaves like a regular Global\_Control telegram to the DP state machine. If no Data\_Exchange telegram precedes the SYNCH telegram, only the New\_GC\_Command interrupt is generated.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Byte** | **Bit Position** | | | | | | | | **Designation** |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| 0 | 0 | 0 |  |  |  |  |  | 0 | Control\_Command |
| 1 | Group\_8  = 1 |  |  |  |  |  |  |  | Group\_Select |

**Figure 7-16: IsoM SYNCH telegram**

Each Global\_Control is compared with the values that can be adjusted in Control\_Command\_Reg (0Eh) and Group\_Select\_Reg (0Fh). If the values are equal a SYNCH telegram will be detected.

Data\_Ex SYNCH SYNCH Data\_Ex GC SYNCH

telegrams 



IsoM

SYNC

DX\_Out\*  New\_GC\_Command\* 

Simple Sync Mode

SYNC 

DX\_Out\*  New\_GC\_Command\*

**Figure 7-17: SYNC-signal and interrupts for synchronization modes (picture only shows the effects by reception of telegrams; time between telegrams is not equal)**

### Isochronous Mode

To enable the Isochronous Mode in the VPC3+S, bit SYNC\_Ena in Mode Register 2 must be set. Additionally the Spec\_Clear\_Mode in Mode Register 0 must be set. The polarity of the SYNC signal can be adjusted with the SYNC\_Pol bit. The register Sync\_PW contains a multiplicator with the base of 1/12 s to adjust the SYNC pulse width. Settings in the Set\_Prm telegram are shown below.

#### The Structured\_Prm\_Data block IsoM (Structure\_Type = 4) is also required for the application. If it is sent by Set\_Prm telegram the bit Prm\_Structure must be set.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Byte** | **Bit Position** | | | | | | | | **Designation** |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| 0 |  |  | Sync\_Req = 0 | Freeze\_Req = 0 |  |  |  |  | Station\_Status |
| 1 |  |  |  |  |  |  |  |  | WD\_Fact\_1 |
| 2 |  |  |  |  |  |  |  |  | WD\_Fact\_2 |
| 3 |  |  |  |  |  |  |  |  | minTSDR |
| 4 |  |  |  |  |  |  |  |  | Ident\_Number\_High |
| 5 |  |  |  |  |  |  |  |  | Ident\_Number\_Low |
| 6 | Group\_8 = 0 |  |  |  |  |  |  |  | Group\_Ident |
| 7 |  | Fail\_Safe = 1 |  |  |  |  |  |  | DPV1\_Status\_1 |
| 8 |  |  |  |  |  |  |  |  | DPV1\_Status\_2 |
| 9 |  |  |  | IsoM\_Req = 1 |  |  |  |  | DPV1\_Status\_3 |
| 10  : 246 |  |  |  |  |  |  |  |  | User\_Prm\_Data |

**Figure 7-18: Format of Set\_Prm telegram for IsoM**

### DP-Slave in an IsoM network

To enable cyclic synchronization via the ‘Simple Sync Mode', the bit DX\_Int\_Port in Mode Register 2 has to be set. Bit SYNC\_Ena must not be set. The settings of the pulse polarity are adjusted like described in the IsoM section.

For the parameterization telegram the DP format is used. Though the DPV1\_Status bytes 1-3 could be used as User\_Prm\_Data, it is generally recommended starting the User\_Prm\_Data at byte 10.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Byte** | **Bit Position** | | | | | | | | **Designation** |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| 0 |  |  | Sync\_Req = depends on SYNCH-format | Freeze\_Req =  depends on SYNCH-format |  |  |  |  | Station\_Status |
| 1 |  |  |  |  |  |  |  |  | WD\_Fact\_1 |
| 2 |  |  |  |  |  |  |  |  | WD\_Fact\_2 |
| 3 |  |  |  |  |  |  |  |  | minTSDR |
| 4 |  |  |  |  |  |  |  |  | Ident\_Number\_High |
| 5 |  |  |  |  |  |  |  |  | Ident\_Number\_Low |
| 6 | Group\_8 = 1 |  |  |  |  |  |  |  | Group\_Ident |
| 7 |  |  |  |  |  |  |  |  | DPV1\_Status\_1 |
| 8 |  |  |  |  |  |  |  |  | DPV1\_Status\_2 |
| 9 |  |  |  |  |  |  |  |  | DPV1\_Status\_3 |
| 10  : 246 |  |  |  |  |  |  |  |  | User\_Prm\_Data |

**Figure 7-19: Format of Set\_Prm for DP-Slave using isochronous cycles**

In opposite to IsoM the first DX\_Out interrupt is generated after the receipt of a SYNCH telegram. If no Data\_Exchange telegram had been received before a SYNCH occurred, no synchronization signal is generated.

**For this mechanism the interrupt controller is used. Hence no signal will be generated, if the mask for DX\_Out in the IMR is set. Since the synchronization signal is now the DX\_Out interrupt, it remains active until the interrupt is acknowledged.**

### IsoM-PLL

The PLL shall handle following issues:

* + - * + The jitter of the SYNCH telegrams has to be smoothed by the PLL. If the jitter exceeds a certain limit, the PLL will recognize a loss of the synchronization.
        + SYNCH telegrams lost due to bus disturbances have to be compensated.
        + Phase shifts due to line delay between the different DP-slaves may be compensated.
        + Generation of a SYNC clock in every slave cycle. The slave application cycle time must be an integer part of DP cycle time.

Reset

PLL

Global\_Control clock (TDP)

Jitter <= 1 us

SYNC clock (TDP/n)

Jitter <= 100 ns

Parameter Status

occurence of Global\_Control

ok ok

error (delayed)

error (missing)

t

DP-Cycle (TDP)

TSYNC Sync\_PW\_Reg



tolerance window

SYNC

GC\_Clock\_Hit

GC\_Clock\_Detect

behaviour in case of: Enable\_In\_Clock=1, Enable\_Out\_Clock=1, Enable\_GC\_Clock=1 Number\_of\_SYNC=3, TPLL\_I=3, TPLL\_O=2

SYNC

GC\_Clock\_Detect

Out\_Clock\_Detect

In\_Clock\_Detect

**Figure 7-20: SYNC clock and status signals of PLL**

To enable the IsoM-PLL in the VPC3+S, bit PLL\_Supported in Mode Register 3 must be set and the IsoM must be parameterized. A Structured\_Prm\_Data block for IsoM in the parameter telegram contains the configuration values for the PLL.

#### The PLL can be used in Isochronous Mode only (not in Simple Sync Mode). The user has to take care that the value of SYNC\_PW\_Reg matches the SYNC cycle time, which could be smaller than the DP cycle time now.

**If E\_limit is reached, a SYNC clock is generated, too**.

|  |  |  |
| --- | --- | --- |
| **Direction** | **Parameter** | **Description** |
| IN | Global\_Control clock | indicates arriving SYNCH telegram |
| PLL start | start and stop of PLL |
| SYNC mode | SYNC clock synchronized to Global\_Control clock |
| SYNC enable | enable SYNC clock after successful synchronization |
| specific clock enable | enable only clock0, input or output clock |
| SYNC cycle time (TSYNC) | period of SYNC clock cycle; shall be an integer part of DP cycle time |
| ratio of DP cycle to SYNC cycle (n) | number of SYNC clock cycles per TDP |
| E\_limit | number of acceptable synchronization errors |
| input time (TPLL\_I) | point in time for actual value acquisition |
| output time (TPLL\_O) | point in time for setpoint transfer |
| PLL window (TPLL\_W) | half the width of the tolerance window |
| First\_Window | start value of PLL window |
| PLL delay time (TPLL\_D) | delay of the generated SYNC clock, to compensate phase shifts between slaves due to the runtimes of SYNCH telegram |
| OUT | SYNC clock | output clock of the PLL |
| SYNCH error | synchronization errors detected, resynchronization necessary |
| PLL synchronized | PLL is synchronized with the DP-Masters SYNCH |
| hit display | SYNCH telegram arrived within tolerance window |
| clock0 display | SYNC clock coincides with the (expected) Global\_Control clock |
| input clock display | SYNC clock designated for actual value acquisition |
| output clock display | SYNC clock designated for setpoint transfer |

**Figure 7-21: Inputs and outputs of the PLL**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Byte** | **Bit Position** | | | | | | | | **Designation** |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | Structured\_Length |
| 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | Structure\_Type |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | Slot\_Number |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | Reserved |
| 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | Version |
| 5  : 8 | 375 / 750 / 1500 / 3000 / 6000 / 12000  (= 31,25 s / 62,5 s / 125 s / 250 s / 500 s / 1000 s) | | | | | | | | TBASE\_DP:  Time Base for TDP (Time Base 1/12 s) |
| 9  : 10 | 1..(216-1)  Note: GSD-Spezifikation: TDP\_MAX=32 ms | | | | | | | | TDP:  DP Cycle Time (Time Base TBASE\_DP) |
| 11 | 1..14 | | | | | | | | TMAPC:  Master Application Cycle Time  (Time Base TDP) |
| 12  : 15 | 375 / 750 / 1500 / 3000 / 6000 / 12000  (= 31,25 s / 62,5 s / 125 s / 250 s / 500 s / 1000 s) | | | | | | | | TBASE\_IO:  Time Base of TI ,TO (Time Base 1/12 s) |
| 16  : 17 | 0..(216-1) | | | | | | | | TI:  Instant in Time of the Actual Value Acquistion (Time Base TBASE\_IO) |
| 18  : 19 | 0..(216-1) | | | | | | | | TO:  Instant in Time of the setpoint transfer (Time Base TBASE\_IO) |
| 20  : 23 | 0..(232-1) | | | | | | | | TDX:  Data\_Exchange Time (Time Base 1/12 s) |
| 24  : 25 | 1..(216-1) (Default: 12) | | | | | | | | TPLL\_W:  PLL Window  (Time Base 1/12 s) |
| 26  : 27 | 0..(216-1) (Default: 0) | | | | | | | | TPLL\_D:  PLL Delay Time (Time Base 1/12 s) |

**Figure 7-22: Format of Structured\_Prm\_Data with IsoM Parameter**

The following input parameters have to be calculated by firmware:

* + - * + SYNC cycle time:

*TSYNC*

 *TDP*

*n*

 *TDP*

*Number* \_ *of* \_ *SYNC*  1

* + - * + start value of PLL window:

*First* \_*Window*  *TPLL* \_*W*  *n*1 *TDP*  *n*2

with *n*1  1; *n*2  0,0003

The base address of the PLL-Buffer depends on the memory mode: 2K Byte mode: 7C0H

4K Byte mode: FC0H

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Byte** | **Bit Position** | | | | | | | | **Designation** |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| 0 | reserved | | In\_Clock\_ Detect | Out\_Clock\_ Detect | PLL\_Synched | GC\_Clock  \_Error | GC\_Clock\_ Detect | GC\_Clock\_ Hit | Status |
| 1 | reserved | | Enable\_ In\_Clock | Enable\_ Out\_Clock | Enable\_ GC\_Clock | SYNC\_Mode | SYNC\_Enable | PLL\_Start | Command |
| 2 |  | | | | | | | | TPLL\_W:  PLL\_Window  1  (Time Base s)  12 |
| : | 1..(216-1) (Default: 12) | | | | | | | |
| 3 |  | | | | | | | |
| 4 |  | | | | | | | | TPLL\_D:  PLL\_Delay\_Time  1  (Time Base s)  12 |
| : | 0..(216-1) (Default: 0) | | | | | | | |
| 5 |  | | | | | | | |
| 6 |  | | | | | | | | TSYNC:  SYNC\_Cycle\_Time  1  (Time Base s)  48 |
| : | 1..(232-1) | | | | | | | |
| 9 |  | | | | | | | |
| 10  : | reserved | | | | | | Number\_of\_ SYNC(9:8) | | Number\_of\_SYNC |
| Number\_of\_SYNC(7:0) | | | | | | | |
| 11 |  |
| 12 |  | | | | | | | | First\_Window  1  (Time Base s)  48 |
| : | 1..(232-1) | | | | | | | |
| 15 |  | | | | | | | |
| 16 | reserved | | | | | | TPLL\_I (9:8) | | TPLL\_I: |
| : 17 | Input\_Time  (Time Base TSYNC) |
| TPLL\_I (7:0) | | | | | | | |
| 18 | reserved | | | | | | TPLL\_O (9:8) | | TPLL\_O: |
| : 19 | Output\_Time (Time Base TSYNC) |
| TPLL\_O (7:0) | | | | | | | |
| 20 | 0..255 | | | | | | | | E\_limit |

|  |  |
| --- | --- |
| **PLL Buffer** | |
| GC\_Clock\_Hit r-0 | **GC\_clock\_Hit**:  The VPC3+ has received a valid ‘SYNCH telegram’ during the tolerance window. |
| GC\_Clock\_ Detect r-0 | **GC\_Clock\_Detect**:  Last SYNC signal coincides with the (expected) ‘SYNCH telegram’. |
| GC\_Clock\_Errror r-0 | **GC\_Clock\_Error**:  PLL detects Synchronization Errors and has to be resynchronized. |
| PLL\_synched r-0 | **PLL\_synched**:  PLL is synchronized with the DP-Masters SYNCH. |
| Out\_Clock\_Detect r-0 | **Out\_Clock\_Detect**:  Last SYNC signal coincides with the instant in time of the setpoint transfer. |
| In\_Clock\_Detect r-0 | **In\_Clock\_Detect**:  Last SYNC signal coincides with the instant in time of the actual value acquisition. |
| PLL\_Start rw-0 | **PLL\_Start**:  0 = PLL is stopped  1 = PLL is started |
| SYNC\_Enable rw-0 | **SYNC\_Enable**:  0 = SYNC signal is not enabled  1 = SYNC signal is send to DATAEXCH\_N |
| SYNC\_Mode rw-0 | **SYNC\_Mode**:  0 = SYNC signal not synchronized to ‘SYNCH telegram’  1 = SYNC signal synchronized to ‘SYNCH telegram’ |
| Enable\_GC\_Clock rw-0 | **Enable\_GC\_Clock**:  0 = generate no SYNC signal coincides with the (expected) ‘SYNCH telegram’  1 = generate SYNC signal coincides with the (expected) ‘SYNCH telegram’ |
| Enable\_Out\_Clock rw-0 | **Enable\_Out\_Clock**:  0 = generate no SYNC signal at TO  1 = generate SYNC signal at TO |
| Enable\_In\_Clock rw-0 | **Enable\_In\_Clock**:  0 = generate no SYNC signal at TI  1 = generate SYNC signal at TI |

|  |  |
| --- | --- |
| Number\_of\_SYNC rw-0 | **Number\_of\_SYNC**:  Number of SYNC cycles per DP cycle: Number\_of\_SYNC + 1 |
| TPLL\_I  rw-0 | **Input\_Time**:  Number of SYNC cycles from start of DP cycle up to TI |
| TPLL\_O  rw-0 | **Output\_Time**:  Number of SYNC cycles from start of DP cycle up to TO |
| E\_limit rw-0 | **E\_limit**:  Number of acceptable synchronization errors during time interval. |

**Figure 7-23: Format of the PLL\_Buffer**

#### TI in the Structured\_Prm\_Data block is the period of time between actual value acquisition and the start of new DP cycle whereas TPLL\_I is the period of time from the start of DP cycle to the point of data acquisition.

start of DP cycle

setpoint transfer

actual value acquisition

start of DP cycle

TO ; TPLL\_O

TPLL\_I

TDP

TI

**Figure 7-24: configuration of TPLL\_O and TPLL\_I**

If none of the Enable\_xx\_Clock bits is set the PLL generates a SYNC clock after every expiration of the slave application cycle (= TSYNC).

|  |  |
| --- | --- |
| **VPC3+S** | **Firmware** |
|  | configure DP-Slave for IsoM  set PLL\_Support |
|  |  |
| receive Set\_(Ext\_)Prm  set New\_(Ext\_)Prm\_Data interrupt |  |
|  | acknowledge New\_(Ext\_)Prm\_Data interrupt  configure PLL |
|  |  |
| receive SYNCH telegrams |  |
|  | set PLL\_Start |
| synchronization of PLL to GC clock → set hit display |  |
|  | set Sync\_Enable |
| release clock on SYNC pin |  |

**Figure 7-25: Start up of PLL (grey scaled task omitted if SYNC\_Mode=0)**

### CS (Clock Synchronization)

The Clock Synchronization mechanism synchronizes the time between devices on a PROFIBUS segment. A time master is a DP-Master. The scheme used is a “backwards time based correction”. The knowledge of when a special timer event message was broadcasted is subsequently used to calculate appropriate clock adjustments.

The synchronized time can be used for time stamp mechanism.

Time Master

Time\_Event

Clock\_Value

Output t

Time Receiver

Clock\_Value

Time\_Event

Input t

Time Receiver Application

1

1: Time Event

tSD

tDT

tRD

t

tPD

2 3 4

2: Clock\_Sync Interrupt

3: read access Receive\_Delay\_Time 4: update system timer

**Figure 7-26: clock synchronization mechanism**

The clock synchronization sequence consists of two messages broad- casted by the time master. When the first message, called Time\_Event, is received the VPC3+S starts the receive delay timer (tRD). The time master then sends a second message, called Clock\_Value, which contains the actual time when the Time\_Event was sent plus the send delay time (tSD). By receiption of the second message the Clock\_Sync interrupt will be generated. To achieve the most accuracy the receive delay timer is running until the user reads the Clock\_Sync-Buffer.

The VPC3+S only synchronizes the received telegrams, the system time management is done by the user. The user has also to account for the time after the receive delay timer has been read till the update of the system time (tPD: process delay time).

The time for transmission delay (tDT: CS\_Delay\_Time) and the Clock\_Sync\_Interval are communicated to the VPC3+S by a Structured\_Prm\_Data block. The CS\_Delay\_Time is used by the user to calculate the system time: tS = Clock\_Value\_Time\_Event + tDT + tRD + tPD

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Byte** | **Bit Position** | | | | | | | | **Designation** |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| 0 |  |  |  |  |  |  |  |  | Structured\_Length |
| 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | Structure\_Type |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | Slot\_Number |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | Reserved |
| 4  : 5 |  |  |  |  |  |  |  |  | Clock\_Sync\_Interval Time Base 10 ms |
| 6  : 13 | Seconds (231..0) | | | | | | | | CS Delay Time can be omitted |
| Fraction Part of Seconds (231..0)  Base is 1/(232) Seconds | | | | | | | |

**Figure 7-27: Format of Structured\_Prm\_Data with Time AR**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Byte** | **Bit Position** | | | | | | | | **Designation** |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
|  | Seconds (231..0) since 1.1.1900 0:00,00 | | | | | | | |  |
| 0  : 7 | or since 7.2.2036 6:28:16 if value < 0x9dff4400 | | | | | | | | Clock\_Value\_  Time\_Event |
| Fraction Part of Seconds (231..0) | | | | | | | |
|  | Base is 1/(232) Seconds | | | | | | | |  |
|  | Seconds (231..0) since 1.1.1900 0:00,00 | | | | | | | |  |
| 8  : 15 | or since 7.2.2036 6:28:16 if value < 0x9dff4400 | | | | | | | | Clock\_Value\_  previous\_TE |
| Fraction Part of Seconds (231..0) | | | | | | | |
|  | Base is 1/(232) Seconds | | | | | | | |  |
| 16 | C | CV | | | | | reserviert | | Clock\_Value\_Status1 |
| 17 | ANH | SWT | reserviert | CR | | reserviert | | SYF | Clock\_Value\_Status2 |

**Figure 7-28: Format of Clock\_Value**

### Processing Sequence

The Clock\_Sync\_Interval is a time for monitoring and has to be written into the Clock\_Sync-Buffer by the user. The Time Receiver state machine in the VPC3+S is started after this write access. The value for Clock\_Sync\_Interval is locked until the next LEAVE-MASTER or a new parameterization occurs. In addition it can be unlocked if the user set the Stop\_Clock\_Sync in Command byte.

Following to a clock synchronization sequence the Clock\_Sync interrupt will be asserted. Further information is contained in the Status byte. If an overflow of the Receive\_Delay\_Timer occurs the Status byte will be cleared. The VPC3+S cannot write new data to the Clock\_Sync-Buffer until the user has acknowledged the Clock\_Sync interrupt. Hence to ensure no new data overwrites the buffer, the user should read out the buffer before acknowledging the interrupt.

The base address of the Clock\_Sync-Buffer depends on the memory mode: 2K Byte mode: 7E0H

4K Byte mode: FE0H

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Byte** | **Bit Position** | | | | | | | | **Designation** |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| 0 | reserved | | | | | | Clock\_Sync\_  Violation | Set\_Time | Status |
| 1 | reserved | | | | | Clock\_Value\_ Check\_Ena | Ignore\_Cyclic\_ State\_Machine | Stop\_  Clock\_Sync | Command |
| 2 | C | CV | | | | | reserved | | Clock\_Value\_Status1 |
| 3 | ANH | SWT | reserved | CR | | reserved | | SYF | Clock\_Value\_Status2 |
| 4  : 11 | Seconds (232-1 .. 0) since 1.1.1900 0:00,00  or since 7.2.2036 6:28:16 if value < 9DFF4400H | | | | | | | | Clock\_Value\_ Time\_Event |
| Fraction Part of Seconds (232-1 .. 0)  Base is 1/(232) Seconds | | | | | | | |
| 12  : 15 | (232-1 .. 0)  Time Base 1 μs | | | | | | | | Receive\_Delay\_Time |
| 16  : 23 | Seconds (232-1 .. 0) since 1.1.1900 0:00,00  or since 7.2.2036 6:28:16 if value < 9DFF4400H | | | | | | | | Clock\_Value\_ previous\_TE |
| Fraction Part of Seconds (232-1 .. 0)  Base is 1/(232) Seconds | | | | | | | |
| 24  : 25 | (216-1 .. 0)  Time Base 10 ms | | | | | | | | Clock\_Sync\_Interval |

|  |  |
| --- | --- |
| **Clock\_Sync-Buffer** | |
| Status bit 7-2  r-000000 | **Reserved** |
| Status bit 1  r-0 | **Clock\_Sync\_Violation**:  Wrong telegram or Time period of 2\*TCSI expired after reception of Time\_Event. |
| Status bit 0  r-0 | **Set\_Time**:  The VPC3+D has received a valid ‘Clock\_Value telegram’ and made the data available in the Clock\_Sync-Buffer. |
| Command bit 7-3  r-00000 | **Reserved** |
| Command bit 2  rw-0 | **Clock\_Value\_Check\_Ena**:  0 = don’t evaluate Clock\_Value\_previous\_TE  1 = check Clock\_Value\_previous\_TE with local variable Time\_Last\_Rcvd |
| Command bit 1  rw-0 | **Ignore\_Cyclic\_State\_Machine**:  0 = Clock Synchronization stops after the receiption of a new Set\_Prm or a LEAVE-MASTER  1 = Clock Synchronization continues until the user set Stop\_Clock\_Sync |
| Command bit 0  w-0 | **Stop\_Clock\_Sync**:  Stop the Clock Synchronization, in order to write a new TCSI without a previous Set\_Prm or LEAVE-MASTER. The Bit is cleared by the Time\_Receiver State Machine. |
| Clock\_Value\_ Status1  bit 7 r-0 | **C: Sign of CV**  0 = add correction value to Time  1 = substract correction value to Time |
| Clock\_Value\_ Status1  bit 6-2  r-00000 | **CV: Correction Value**  0 = 0 min  1..31 = 30..930 min |
| Clock\_Value\_ Status1  bit 1-0 r-00 | **Reserved** |

|  |  |
| --- | --- |
| **Clock\_Sync-Buffer** | |
| Clock\_Value\_ Status2  bit 7 r-0 | **ANH: Announcment Hour**  0 = no change planned within the next hour  1 = a change of SWT will occur within the next hour |
| Clock\_Value\_ Status2  bit 6 r-0 | **SWT: Summertime**  0 = Winter Time  1 = Summer Time |
| Clock\_Value\_ Status2  bit 5 r-0 | **Reserved** |
| Clock\_Value\_ Status2  bit 4-3 r-00 | **CR: Accuracy**  0 = 1 ms  1 = 10 ms  2 = 100 ms  3 = 1 s |
| Clock\_Value\_ Status2  bit 2-1 r-00 | **Reserved** |
| Clock\_Value\_ Status2  bit 0 r-0 | **SYF: Synchronisation Active**:  0 = Clock\_Value\_Time\_Event is synchronized  1 = Clock\_Value\_Time\_Event is not synchronized |
| r-0 | **Clock\_Value\_Time\_Event**:  Same format as defined in IEC 61158-6 is used. Value is stored with the most significant byte at the lowest address. No address swapping is done for Intel format. |
| r-0 | **Receive\_Delay\_Time**:  Value is stored with the most significant byte in address 12. No address swapping is done for Intel format. |
| r-0 | **Clock\_Value\_previous\_TE**:  Same format as defined in IEC 61158-6 is used. Value is stored with the most significant byte at the lowest address. No address swapping is done for Intel format. |
| rw-0 | **Clock\_Sync\_Interval**:  Value is stored with the most significant byte in address 24. No address swapping is done for Intel format. |

**Figure 7-29: Format of the Clock\_Sync-Buffer**

|  |  |
| --- | --- |
| **VPC3+S** | **Firmware** |
|  | set CS\_Supported |
|  |  |
| reception of Set\_(Ext\_)Prm  set New\_(Ext\_)Prm\_Data interrupt |  |
|  | acknowledge interrupt  write Clock\_Sync\_Interval to CS-Buffer |
|  |  |
| reception of Time\_Event start Receive\_Delay\_Timer reception of Clock\_Value  set Clock\_Sync interrupt |  |
| stop Receive\_Delay Timer | read CS\_Status IF (Set\_Time=’1’) THEN  read CS\_Buffer update system time  END IF  acknowledge interrupt |

**Figure 7-30: communication scheme**

## Notes:

## Universal Processor Bus Interface

### Overview

The VPC3+S can be interfaced by using either a parallel 8-bit data interface or an SPI or I2C interface.

In parallel mode the VPC3+S provides an 8-bit data interface with an 11-bit address bus. The VPC3+S supports all 8-bit processors and micro- controllers based on the 80C51/52 (80C32) from Intel, the Motorola HC11 family, as well as 8- /16-bit processors or microcontrollers from the Siemens 80C166 family, X86 from Intel and the HC16 and HC916 family from Motorola. Because the data formats from Intel and Motorola are different, VPC3+S automatically carries out ‘byte swapping’ for accesses to the following 16-bit registers (Interrupt Register, Status Register and Mode Register 0) and the 16-bit RAM cell (R\_User\_WD\_Value). This makes it possible for a Motorola processor to read the 16-bit value correctly. Reading or writing takes place, as usual, through two accesses (8-bit data bus).

Four SPI modes are supported which differ in clock polarity and clock phase. In these interface modes the VPC3+S acts like a memory device with serial (SPI) interface connected to the CPU. The chip needs to be selected by pulling the Slave-Select pin (SPI\_XSS) low before receiving clock pulses via SPI\_SCK pin from the CPU. Depending on the OP-code received the VPC3+S carries out a read or write operation starting at the specified address inside the internal memory. Serial data is shifted in via SPI\_MOSI pin and shifted out via SPI\_MISO pin.

In I2C mode the VPC3+S can be connected to an I2C network by using the pins I2C\_SCK and I2C\_SDA. In this mode the VPC3+S acts like a memory device with serial (I2C) interface connected to the CPU. The chip supports slave mode only and the desired slave address can be selected by using the pins I2C\_A[6:0]. Upon reception of the correct slave address and depending on the status of the R/W bit the VPC3+S carries out a read or write operation starting at the specified address inside the internal memory.

The Bus Interface Unit (BIU) and the Dual Port RAM Controller (DPC) that controls accesses to the internal RAM belong to the processor interface of the VPC3+S.

The VPC3+S is supplied with a clock pulse rate of 48MHz. In addition, a clock divider is integrated. The clock pulse is divided by 2 (Pin: DIVIDER = '1') or 4 (Pin: DIVIDER = '0') and applied to the pin CLKOUT. This allows the connection of a slower controller without additional expenditures in a low-cost application.

### Parallel Interface Modes

The Bus Interface Unit (BIU) is the interface to the connected processor/microcontroller. This is a synchronous or asynchronous 8-bit interface with an 11-bit (12-bit in 4K Byte mode) address bus. The interface is configurable via 2 pins (XINT/MOT, MODE). The connected processor family (bus control signals such as XWR, XRD, or R\_W and the data format) is specified with the XINT/MOT pin. Synchronous or asynchronous bus timing is specified with the MODE pin.

|  |  |  |  |
| --- | --- | --- | --- |
| **SERMODE** | **XINT/MOT** | **MODE** | **Processor Interface Mode** |
| 0 | 0 | 1 | Synchronous Intel mode |
| 0 | 0 | 0 | Asynchronous Intel mode |
| 0 | 1 | 0 | Asynchronous Motorola mode |
| 0 | 1 | 1 | Synchronous Motorola mode |

**Figure 8-1: Configuration of the parallel Processor Interface Modes**

Examples of various Intel system configurations are given in subsequent sections. The internal address latch and the integrated decoder must be used in the synchronous Intel mode. One figure shows the minimum con- figuration of a system with the VPC3+S, where the chip is connected to an EPROM version of the controller. Only a clock generator is necessary as an additional device in this configuration. If a controller is to be used without an integrated program memory, the addresses must be latched for the external memory.

#### Notes:

If the VPC3+S is connected to an 80286 or similar processor, it must be taken into consideration that the processor carries out word accesses. That is, either a ‘swapper’ is necessary that switches the characters out of the VPC3+S at the correct byte position of the 16-bit data bus during reading or the least significant address bit is not connected and the 80286 must read word accesses and evaluate only the lower byte.

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Input/ Output** | **Type** | **Comments** |
| DB(7..0) | I/O | Tristate | High-resistance during RESET |
| AB(10..0) | I |  | AB(10) has a pull down resistor. |
| MODE | I |  | Configuration: syn/async interface |
| XWR/E\_CLOCK AB11 | I |  | Intel: Write Sync. Motorola: E-Clk AB11 (Asynchronous Motorola Mode) |
| XRD/R\_W | I |  | Intel: Read Motorola: Read/Write |
| XCS AB11 | I |  | Chip Select  AB11 (Synchronous Intel Mode) |
| ALE/AS  AB11 | I |  | Intel/Motorola: Address Latch Enable  AB11 (Async. Intel / Sync. Motorola Mode) |
| DIVIDER | I |  | Scaling factor 2/4 for CLKOUT 2/4 |
| X/INT | O | Push/Pull | Polarity programmable |
| XRDY/XDTACK | O | Push/Pull \* | Intel/Motorola: Ready-Signal |
| CLK | I |  | 48 MHz |
| XINT/MOT | I |  | Setting: Intel/Motorola |
| CLKOUT2/4 | O | Push/Pull | 24/12 MHz |
| RESET | I | Schmitt-Trigger | Minimum of 4 clock cycles |

**Figure 8-2: Microprocessor Bus Signals**

\* Due to compatibility reasons to existing competitive chips the XRDY/XDTACK output of the VPC3+S has push/pull characteristic (no tristate!).

### Synchronous Intel Mode

In this mode Intel CPUs like 80C51/52/32 and compatible processor series from several manufacturers can be used.

* Synchronous bus timing without evaluation of the XREADY signal
* 8-bit multiplexed bus: ADB7..0
* The lower address bits AB7..0 are stored with the ALE signal in an in- ternal address latch.
* The internal CS decoder is activated. VPC3+S generates its own CS signal from the address lines AB10..3. The VPC3+S selects the relevant address window from the AB2..0 signals.
* A11 from the microcontroller must be connected to XCS (pin BGA\_C1/QFP\_3) in 4K Byte mode as this is the additional address bus signal in this mode. In 2K Byte mode this pin is not used and should be pulled to VDD.

### Asynchronous Intel Mode

In this mode various 16-/8-bit microcontroller series like Intel’s x86, Siemens 80C16x or compatible series from other manufacturers can be used.

* Asynchronous bus timing with evaluation of the XREADY signal
* 8-bit non-multiplexed bus: DB7..0, AB10..0 (AB11..0 in 4K Byte mode)
* The internal VPC3+S address decoder is disabled, the XCS input is used instead.
* External address decoding is always necessary.
* External chip select logic is necessary if not present in the microcon- troller
* A11 from the microcontroller must be connected to ALE/AS (pin BGA\_C5/QFP\_35) in 4K Byte mode as this is the additional address bus signal in this mode. In 2K Byte mode this pin is not used and should be pulled to GND.

### Asynchronous Motorola Mode

Motorola microcontrollers like the HC16 and HC916 can be used in this mode. When using HC11 types with a multiplexed bus the address signals AB7..0 must be generated from the DB7..0 signals externally.

* Asynchronous bus timing with evaluation of the XREADY signal
* 8-bit non-multiplexed bus: DB7..0, (AB11..0 in 4K Byte mode)
* The internal VPC3+S address decoder is disabled, the XCS input is used instead.
* Chip select logic is available and programmable in all microcontrollers mentioned above.
* AB11 must be connected to XWR/E\_CLOCK (pin BGA\_D5/QFP\_32) in 4K Byte mode as this is the additional address bus signal in this mode. In 2K Byte mode this pin is not used and should be pulled to GND.

### Synchronous Motorola Mode

Motorola microcontrollers like the HC11 types K, N, M, F1 or the HC16- and HC916 types with programmable E\_Clock timing can be used in this mode. When using HC11 types with a multiplexed bus the address signals AB7..0 must be generated from the DB7..0 signals externally.

* Synchronous bus timing without evaluation of the XREADY signal
* 8-bit non-multiplexed bus: DB7..0, AB10..0 (AB11..0 in 4K Byte mode)
* The internal VPC3+S address decoder is disabled, the XCS input is used instead.
* For microcontrollers with chip select logic (K, F1, HC16 and HC916), the chip select signals are programmable regarding address range, pri- ority, polarity and window width in the write cycle or read cycle.
* For microcontrollers without chip select logic (N and M) and others, an external chip select logic is required. This means additional hardware and a fixed assignment.
* If the CPU is clocked by the VPC3+S, the output clock pulse (CLKOUT 2/4) must be 4 times larger than the E\_Clock. That is, a clock pulse sig- nal must be present at the CLK input that is at least 10 times larger than the desired system clock pulse (E\_Clock). The Divider-Pin must be connected to ‘0’ (divider 4). This results in an E\_Clock of 3 MHz.
* AB11 must be connected to ALE/AS (pin BGA\_C5/QFP\_35) in 4K Byte mode as this is the additional address bus signal in this mode. In 2K Byte mode this pin is not used and should be pulled to GND.

### SPI Interface Mode

The VPC3+S is designed to interface directly with the Serial Peripheral Interface (SPI) port of many of today’s popular microcontroller families. It may also interface with microcontrollers that do not have a built-in SPI port by using discrete I/O lines programmed to match the SPI protocol.

The SPI mode allows a duplex, synchronous, serial communication between the CPU and peripheral devices. The CPU is always master while the VPC3+S is always slave in this configuration.

Four associated SPI port pins are dedicated to the SPI function as:

* + - * Slave-Select (SPI\_XSS)
      * Serial Clock (SPI\_SCK)
      * Master-Out-Slave-In (SPI\_MOSI)
      * Master-In-Slave-Out (SPI\_MISO)

The clock phase control bit (SPI\_CPHA) and the clock polarity control bit (SPI\_CPOL) select one of four possible clock formats to be used by the SPI system. The CPOL bit simply selects a non-inverted or inverted clock. The CPHA bit is used to accommodate two fundamentally different protocols by sampling data on odd numbered SCK edges (SPI\_CPHA=’0’) or on even numbered SCK edges (SPI\_CPHA=’1’).

The main element of the SPI system is the SPI Data Register. The 8-bit data register in the master and the 8-bit data register in the slave are linked by the MOSI and MISO pins to form a distributed 16-bit register. When a data transfer operation is performed, this 16-bit register is serially shifted eight bit positions by the SCK clock from the master, so data is exchanged between the master and the slave.

**MASTER SPI (CPU)**

**SLAVE SPI (VPC3+S)**

**MISO**

**MISO**

**MOSI**

**MOSI**

**SHIFT REGISTER**

**SCK**

**SCK**

**XSS**

**XSS**

**BAUDRATE GENERATOR**

**SHIFT REGISTER**

**Figure 8-3: SPI Master-Slave-Transfer (Block Diagram)**

Data written to the master SPI Data Register becomes the output data for the slave, and data read from the master SPI Data Register after a transfer operation is the input data from the slave.

### Transmission Formats

During an SPI transmission, data is transmitted (shifted out serially) and re- ceived (shifted in serially) simultaneously. The serial clock (SCK) synchro- nizes shifting and sampling of the information on the two serial data lines. The slave select line allows selection of an individual slave SPI device, slave devices that are not selected do not interfere with SPI bus activities.

The **CPOL** clock polarity control bit specifies an active high or low clock and has no significant effect on the transmission format. The **CPHA** clock phase control bit selects one of two fundamentally different transmission formats. Clock phase and polarity should be identical for the master SPI device and the communicating slave device.

### CPHA = 0 Transfer Format

The first edge on the SCK line is used to clock the first data bit of the slave into the master and the first data bit of the master into the slave. In some peripherals, the first bit of the slave’s data is available at the slave’s data out pin as soon as the slave is selected. In this format, the first SCK edge is issued a half cycle after SS has become low.

A half SCK cycle later, the second edge appears on the SCK line. When this second edge occurs, the value previously latched from the serial data input pin is shifted into the shift register.

**SCK Edge Nr.**



**Begin**

**Transfer**

**End**

**1 2 3 4 5 6 7 8 9 10 11 12 13 14 15**

**Bit 7**

**Bit 6**

**Bit 5**

**Bit 4**

**Bit 3**

**Bit 2**

**Bit 1**

**Bit 0**

**SCK**

**(CPOL=’0’)**

**SCK**

**(CPOL=’1’)**

**SAMPLE**

**MOSI / MISO**

**MOSI**

**MISO**

**XSS**

**Figure 8-4: SPI Transfer Format (CPHA='0')**

After this second edge, the next bit of the SPI master data is transmitted out of the serial data output pin of the master to the serial input pin on the slave. This process continues for a total of 16 edges on the SCK line, with data being latched on odd numbered edges and shifted on even numbered edges.

Data reception is double buffered. Data is shifted serially into the SPI shift register during the transfer and is transferred to the parallel SPI Data Register after the last bit is shifted in.

### CPHA = 1 Transfer Format

Some peripherals require the first SCK edge before the first data bit becomes available at the data out pin, the second edge clocks data into the system. In this format, the first SCK edge is issued by setting the CPHA bit at the beginning of the 8-cycle transfer operation.

The first edge of SCK occurs immediately after the half SCK clock cycle synchronization delay. This first edge commands the slave to transfer its first data bit to the serial data input pin of the master.

A half SCK cycle later, the second edge appears on the SCK pin. This is the latching edge for both the master and slave.

When the third edge occurs, the value previously latched from the serial data input pin is shifted into the SPI shift register. After this edge, the next bit of the master data is coupled out of the serial data output pin of the master to the serial input pin on the slave.

**SCK Edge Nr.**



**Begin**

**Transfer**

**End**

**1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16**

**Bit 7**

**Bit 6**

**Bit 5**

**Bit 4**

**Bit 3**

**Bit 2**

**Bit 1**

**Bit 0**

**SCK**

**(CPOL=’0’)**

**SCK**

**(CPOL=’1’)**

**SAMPLE**

**MOSI / MISO**

**MOSI**

**MISO**

**XSS**

**Figure 8-5: SPI Transfer Format (CPHA='1')**

This process continues for a total of 16 edges on the SCK line with data being latched on even numbered edges and shifting taking place on odd numbered edges.

Data reception is double buffered, data is serially shifted into the SPI shift register during the transfer and is transferred to the parallel SPI Data Register after the last bit is shifted in.

### Principles of Operation

The VPC3+S contains an 8-bit instruction register and a 16-bit address register. The device is accessed via the MOSI pin, with data being clocked in on the configured edge of SCK. The XSS pin must be held low for the entire operation.

The first byte received during a valid SPI transfer is interpreted as SPI instruction. [Figure 8-6](#_bookmark2) lists the supported instruction bytes and formats for the device operation. All instructions, addresses, and data are transferred MSB first, LSB last.

|  |  |  |
| --- | --- | --- |
| **Instruction Name** | **Instruction Format** | **Description** |
| READ BYTE | 0001 0011 | Read a single data byte from selected address |
| READ ARRAY | 0000 0011 | Read several data bytes beginning at selected address (with auto-increment) |
| WRITE BYTE | 0001 0010 | Write a single data byte to selected address |
| WRITE ARRAY | 0000 0010 | Write several data bytes beginning at selected address (with auto-increment) |

**Figure 8-6: SPI Instruction Set**

#### Note:

In SPI interface mode all internal addresses are interpreted in Intel format. Motorola format (byte swapping for certain addresses) is not supported in SPI mode.

### READ BYTE Sequence

The device is selected by pulling XSS low. The 8-bit READ BYTE instruction is transmitted to the VPC3+S followed by the 16-bit address, with the four MSBs of the address being “don’t care” bits (in case of 2 kB RAM mode the five MSBs of the address are “don’t care”).

After the correct READ BYTE instruction and address are sent, the data byte stored in the memory at the selected address is shifted out on the MISO pin. After additional 8 SCK pulses the complete data byte has sent and no more valid data bits are shifted out on the MISO pin. There is no auto-increment mechanism for this instruction. The read operation is terminated by raising the XSS pin ([Figure 8-7](#_bookmark3)).

#### Note:

When reading from the Control Parameter memory (address 0x000 to address 0x015) only the READ BYTE instruction may be used.

Otherwise an unintended read operation to the subsequent memory location will occur leading to an unpredictable behavior of the VPC3+S.

**XSS**



**0 1 2 3 4 5 6 7 8 9 10 11**

**20 21 22 23 24 25 26 27 28 29 30 31**

**Instruction**

**16-bit Address**

**0 0 0 1 0 0 1 1 15 14 13 12**

**3 2 1 0**

**“don’t care”**

**Data Out**

**High-Impedance**

**7 6 5 4 3 2 1 0**

**SCK**

**(CPOL=’0’)**

**MOSI**

**MISO**

**Figure 8-7: READ BYTE Sequence**

### READ ARRAY Sequence

The device is selected by pulling XSS low. The 8-bit READ BYTE instruc- tion is transmitted to the VPC3+S followed by the 16-bit address, with the four MSBs of the address being “don’t care” bits (in case of 2 kB RAM mode the five MSBs of the address are “don’t care”).

After the correct READ ARRAY instruction and address are sent, the data byte stored in the memory at the selected address is shifted out on the MISO pin. After additional 8 SCK pulses the complete first data byte has been sent. The data byte stored in the memory at the next address can be read sequentially by continuing to provide clock pulses. The internal Ad- dress Pointer is automatically incremented to the next higher address after each byte of data is shifted out. When the highest address is reached (0x7FF in case of 2 kB RAM mode or 0xFFF in 4 kB mode), the address counter rolls over to address 0x000 allowing the read cycle to be continued indefinitely. The read operation is terminated by raising the XSS pin ([Figure](#_bookmark4) [8-8](#_bookmark4)).

#### Note:

The SPI instruction READ ARRAY may not be used when reading from the Control Parameter memory (address 0x000 to address 0x015).

Otherwise (due to the auto-increment mechanism of the READ ARRAY instruction) an unintended read operation to the subsequent memory loca- tion will occur leading to an unpredictable behavior of the VPC3+S.

**XSS**



**0 1 2 3 4 5 6 7 8 9 10 11**

**20 21 22 23 24 25 26 27 28 29 30 31**

**Instruction**

**16-bit Address**

**0 0 0 0 0 0 1 1 15 14 13 12**

**3 2 1 0**

**“don’t care”**

**Data Byte 1**

**High-Impedance**

**7 6 5 4 3 2 1 0**

**SCK**

**(CPOL=’0’)**

**MOSI**

**MISO**

**XSS**

**32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47**

**“don’t care”**

**Data Byte 2**

**Data Byte 3**

**Data Byte n**

**7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 0**

**7 6 5 4 3 2 1 0**

**SCK**

**(CPOL=’0’)**

**MOSI**

**MISO**

**Figure 8-8: READ ARRAY Sequence**

### WRITE BYTE Sequence

The VPC3+S is selected by pulling XSS low. The 8-bit WRITE BYTE instruction is transmitted to the device followed by the 16-bit address, with the four MSBs of the address being “don’t care” bits (in case of 2 kB RAM mode the five MSBs of the address are “don’t care”).

After the correct WRITE BYTE instruction and address are sent, the data byte is shifted in on the MOSI pin. Once 8 SCK clock pulses are received the sampled data byte is written to the selected address. Providing more SCK clock pulses does not affect the VPC3+S. The write operation is terminated by raising the XSS pin.

**XSS**



**0 1 2 3 4 5 6 7 8 9 10 11**

**20 21 22 23 24 25 26 27 28 29 30 31**

**Instruction**

**16-bit Address**

**Data In**

**0 0 0 1 0 0 1 0 15 14 13 12**

**3 2 1 0 7 6 5 4 3 2 1 0**

**High-Impedance**

**SCK**

**(CPOL=’0’)**

**MOSI**

**MISO**

**Figure 8-9: WRITE BYTE Sequence**

### WRITE ARRAY Sequence

The WRITE ARRAY sequence is similar to the WRITE BYTE sequence unless more than one data byte is transferred. After the reception of every data byte the internal destination address is auto-incremented by ‘1’. When the highest address is reached (0x7FF in case of 2 kB RAM mode or 0xFFF in 4 kB mode), the address counter rolls over to address 0x000 allowing the write cycle to be continued indefinitely. The write operation is terminated by raising the XSS pin.

**XSS**



**0 1 2 3 4 5 6 7 8 9 10 11**

**20 21 22 23 24 25 26 27 28 29 30 31**

**Instruction**

**16-bit Address**

**Data Byte 1**

**0 0 0 0 0 0 1 0 15 14 13 12**

**3 2 1 0 7 6 5 4 3 2 1 0**

**High-Impedance**

**SCK**

**(CPOL=’0’)**

**MOSI**

**MISO**

**XSS**

**32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47**

**Data Byte 2**

**Data Byte 3**

**Data Byte n**

**7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 0**

**7 6 5 4 3 2 1 0**

**High-Impedance**

**SCK**

**(CPOL=’0’)**

**MOSI**

**MISO**

**Figure 8-10: WRITE ARRAY Sequence**

### I2C Interface Mode

The VPC3+S supports a bidirectional, 2-wire bus and data transmission protocol. A device that sends data onto the bus is defined as transmitter, while a device receiving data is defined as a receiver. The bus has to be controlled by a master device which generates the Serial Clock (SCK), controls the bus access and generates the Start and Stop conditions, while the VPC3+S works as slave. Both master and slave can operate as transmitter or receiver, but the master device determines which mode is activated.

The data on the SDA line must be stable during the HIGH period of the clock. The HIGH or LOW state of the data line can only change when the clock signal on the SCK line is LOW ([Figure 8-11](#_bookmark5)). One clock pulse is generated for each data bit transferred.

SDA

SCK

data line stable; data valid

change of data allowed

**Figure 8-11: Bit Transfer on the I2C bus**

All transactions begin with a START (S) and can be terminated by a STOP

(P) condition. A HIGH to LOW transition on the SDA line while SCK is HIGH defines a START condition. A LOW to HIGH transition on the SDA line while SCK is HIGH defines a STOP condition.

SDA

SCK

SDA

SCK

P

START condition STOP condition

S

**Figure 8-12: START and STOP condition**

START and STOP conditions are always generated by the master. The bus is considered to be busy after the START condition. The bus is considered to be free again a certain time after the STOP condition.

Every byte sent on the SDA line must be 8 bits long. The number of bytes that can be transmitted per transfer is unrestricted. Each byte has to be followed by an Acknowledge bit. Data is transferred with the Most Significant Bit (MSB) first.

SDA

MSB

acknowledgement signal from slave

P

acknowledgement Sr signal from slave



SCK

S or Sr

1 2 7 8 9 1 2 3 to 8 9

ACK ACK

Sr or P

START or

repeated START condition

byte complete, interrupt within slave

clock line held LOW

while interrupts are serviced

STOP or repeated START condition

**Figure 8-13: Data Transfer on the I2C Bus**

Each receiving device, when addressed, is obliged to generate an Acknowledge after the reception of each byte. The master device must generate an extra clock pulse which is associated with this Acknowledge bit. The device that acknowledges, has to pull down the SDA line during the acknowledge clock pulse in such a way that the SDA line is stable low during the high period of the acknowledge related clock pulse. Of course, setup and hold times must be taken into account. During reads, a master must signal an end of data to the slave by not generating an Acknowledge bit on the last byte that has been clocked out of the slave. In this case, the slave (VPC3+S) will leave the data line high to enable the master to generate the Stop condition.

A control byte is the first byte received following the Start condition from the master device ([Figure 8-14](#_bookmark6)). The control byte consists of a seven-bit Slave Address SA[6:0] to select which device is accessed. The Slave Address bits in the control byte must correspond to the logic levels on the I2C\_SA[6:0] pins for the VPC3+S to respond.

**Read / Write Bit**

**S SA6 SA5 SA4 SA3 SA2 SA1 SA0 R/W ACK**

**Slave Address**

**START Condition**

**Acknowledge Bit**

**Figure 8-14: Control Byte Format**

The last bit of the control byte defines the operation to be performed. When set to a ‘1’, a read operation is selected. When set to a ‘0’, a write operation is selected.

The next two bytes received define the address of the first data byte ([Figure](#_bookmark7) [8-15](#_bookmark7)). In case of the 4 kB RAM mode is selected only A11 to A0 are used, the upper four address bits are “don’t care” bits (in case of 2 kB RAM mode the upper five address bits are “don’t care”).

The upper address bits (MSB) are transferred first, followed by the Less Significant bits (LSB). Following the Start condition, the VPC3+S monitors the SDA line checking the control byte transmitted and, upon receiving appropriate Slave Address bits, the device outputs an Acknowledge signal on the SDA line. Depending on the state of the R/W bit, the VPC3+S will select a read or write operation.

**Control Byte Address High Byte Address Low Byte**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | |  |  | | | | | | | |  |  | | | | | | | |
| **SA6** | **SA5** | **SA4** | **SA3** | **SA2** | **SA1** | **SA0** |  |  | **X** | **X** | **X** | **X** | **A11** | **A10** | **A9** | **A8** | **A7** | **A6** | **A5** | **A4** | **A3** | **A2** | **A1** | **A0** |
| **R/W** | |
|  | | | | | | |  | | | | | | | | | | | | | | | | | | | |

**Slave Address**

**Figure 8-15: Address Sequence Bit Assignments**

### WRITE Sequence

Following the START condition from the master, Slave Address (6 bits) and the R/W bit (which is a logic low) are clocked onto the bus by the master transmitter. This indicates to the addressed slave receiver that the address high byte will follow once it has generated an Acknowledge bit during the ninth clock cycle. Therefore, the next byte transmitted by the master is the high-order byte of the address and will be written into the Address Pointer of the VPC3+S. The next byte is the Least Significant Address Byte. After receiving another Acknowledge signal from the VPC3+S, the master device will transmit the data byte to be written into the addressed memory location. The VPC3+S acknowledges again and the master either generates a STOP condition or transfers more data bytes to the VPC3+S. Upon receipt of each data byte, the VPC3+S generates an Acknowledge signal and the internal Address Pointer is incremented by ‘1’. When the highest address is reached (0x7FF in case of 2 kB RAM mode or 0xFFF in 4 kB mode), the address counter rolls over to address 0x000 allowing the write sequence to be continued indefinitely. The write operation is terminated by receiving a STOP condition from the master.

**S T A**

**R Control Byte Address High Byte T**

**P**

**0**

**1**

**2**

**3**

**4**

**5**

**6**

**7**

**0**

**1**

**2**

**3**

**4**

**5**

**6**

**7**

**0**

**1**

**2**

**3**

**4**

**5**

**6**

**7**

**8**

**9**

**10**

**11**

**X**

**X**

**X**

**X**

**S**

**0**

**Address Low Byte**

**Data Byte 0**

**S T**

**Data Byte n O P**

**A A A A A**

**C C C C C**

**K K K K K**

**Figure 8-16: I2C WRITE Sequence**

### READ Operations

Read operations are initiated in the same way as write operations, with the exception that the R/W bit of the control byte is set to ‘1’. There are three basic types of read operations: current address read, random read and sequential read.

### Current Address READ Operation

The VPC3+S contains an address counter that maintains the address of the last byte accessed, internally incremented by ‘1’. Therefore, if the previous read access was to address ‘n’ (n is any legal address), the next current address read operation would access data from address n + 1.

Upon receipt of the control byte with R/W bit set to ‘1’, the VPC3+S issues an acknowledge and transmits the 8-bit data byte. The master will not acknowledge the transfer, but does generate a STOP condition and the VPC3+S discontinues transmission.

**S T A**

**R Control Byte T**

**S T**

**Data Byte (n) O P**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **S** |  |  |  |  |  |  |  | **1** |  | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |  | **P** |

**A N**

**C O**

**K**

**A C K**

**Figure 8-17: I2C Current Address READ Operation**

### Random READ Operation

Random read operations allow the master to access any memory location in a random manner. To perform this type of read operation, the byte address must first be set. This is accomplished by sending the byte address to the VPC3+S as part of a write operation (R/W bit set to ‘0’). Once the byte address is sent, the master generates a START condition following the acknowledge. This terminates the write operation, but not before the internal Address Pointer is set. The master issues the control byte again, but with the R/W bit set to a ‘1’. The VPC3+S will then issue an acknowledge and transmit the 8-bit data byte. The master will not acknowledge the transfer, but does generate a Stop condition which causes the VPC3+S to discontinue transmission ([Figure 8-17](#_bookmark8)). After a random Read command, the internal address counter will point to the address location following the one that was just read.

**S T A**

**R Control Byte Address High Byte T**

**0**

**1**

**2**

**3**

**4**

**5**

**6**

**7**

**8**

**9**

**10**

**11**

**X**

**X**

**X**

**X**

**S**

**0**

**1**

**S**

**P**

**0**

**1**

**2**

**3**

**4**

**5**

**6**

**7**

**S T A**

**Address Low Byte R T**

**Control Byte**

**S T**

**Data Byte O P**

**A A A A N**

**C C C C O**

**K K K K**

**A C K**

**Figure 8-18: I2C Random READ Operation**

### Sequential READ Operation

Sequential reads are initiated in the same way as a random read, except that once the VPC3+S transmits the first data byte, the master issues an acknowledge as opposed to the Stop condition used in a random read. This acknowledge directs the VPC3+S to transmit the next sequentially addressed data byte ([Figure 8-19](#_bookmark9)). Following the final byte transmitted to the master, the master will NOT generate an acknowledge, but will generate a STOP condition. To provide sequential reads, the VPC3+S contains an internal Address Pointer which is incremented by ‘1’ upon completion of each operation. This Address Pointer allows the entire memory contents to be serially read during one operation. The internal

Address Pointer will automatically roll over from address 0xFFF (0x7FF in 2 kB mode) to address 0x000 if the master acknowledges the byte received from address 0xFFF (0x7FF).

**S T A**

**R Control Byte T**

**1**

**S**

**7**

**0**

**1**

**2**

**3**

**4**

**5**

**6**

**7**

**Data Byte (n)**

**Data Byte (n+1)**

**S T**

**Data Byte (n+x) O P**

**P**

**0**

**1**

**2**

**3**

**4**

**5**

**6**

**7**

**0**

**1**

**2**

**3**

**4**

**5**

**6**

**A A A N**

**C C C O**

**K K K**

**A C** **K**

**Figure 8-19: I2C Sequential READ Operation**

### Application Examples (Principles)

DIVIDER



CLK

WR RD

INT 0

Clockdivider

XW R

XRD

X/INT RT S

TxD

Por t 0 A / D 7 ...0

**80C 32/**

**C501**

Rx D

ALE

XC TS

**VPC3+**

Por t 2

AB 1 5...8

(0000 0XXX)

**GND**

AB8

AB9 AB10

Mode

VPC3+

Reset

Reset

3K 3

1K

1K

1K

1K

AB 7 ..0

Decoder

DB 7..0

**Address-Latch**

DB 7..0

**Data**

Clock-Generator 48MHz

**GND**

**VDD**

**Figure 8-20: Low Cost System with 80C32**

**GND VDD**



DIVIDER

CLK

Clockdivider

WR RD

INT0

XW R XRD

X/INT

RT S

TxD

**80C 32**

**20/16 MHz**

ALE

DB 7..0

**Data**

DB 7..0

**Address-Latch**

Rx D

XCTS

Po rt 0

Address- Latch

**VPC3+**

A/D 7..0

Port 2 AB 15..8

(0000 0XXX)

**GND**

PSEN

AB 15..0

Reset

AB8

AB9 AB10

Mode

VPC3+

Reset

EPROM

64kB

RAM

32kB

Address- Decoder

3K 3

RD W R

1K

1K

Clock-Generator

48 MHz

1K

1K

**Figure 8-21: 80C32 System with External Memory**



12/24 MHz

DIVIDER

XWR XRD

X/INT

RTS

TxD

XREADY

RxD

XCTS

DB 15..0

DB 7..0

DB(7..0)

**VPC3+**

AB 23..0

AB 12..1

AB(10..0)

**GND**

RD WR

Reset

**driver, control logic**

XCS Mode

VPC3+

Reset

EPROM

RAM

CSRAM

CSEPROM

64kB 32kB

address CS

decoder

3K3

1K

clockdivider

AB

DB

**80286**

**+**

**Buscontr. (82288) +**

**82244**

WR RD INT R

Ready- logic

CLK

Clockgenerator 48 MHz

**GND**

**Figure 8-22: 80286 System (X86 Mode)**

### Application with 80C32 (2K Byte RAM Mode)

**VPC3+**

48MHz

22R

CLK

connect to VDD or GND

GND

XINT/MOT

DIVIDER CLKOUT

XDATAEX

LED for Data-Exchange

µC

VDD VDD VDD

VDD

RESET

XCS MODE XTEST0

XTEST1

XREADY

X/INT

µC

VDD

R 4k7

µC

µC µC

XRD XWR

ALE XRD XWR

TXD \* RXD RTS

XCTS

RS485

RS485 RS485

AB 8 AB 0

AB 9 AB 1

AB 10 AB 2

AB 11 AB 3

GND

AB 12 1

AB 4

AB 13 AB 5

AB 14 AB 6

AB 15

GND

AB (8 …15)

µC

GND

GND

AB 7

AB 8

AB 9

AB 10

DB 0

DB 1

DB 2

DB 3

DB 4

DB 5

DB 6

DB 7

ADB 0

ADB 1

ADB 2

ADB 3

ADB 4

ADB 5

ADB 6

ADB 7

DB (0..7)

\* TxD: tristate, external pull-up resistor required µC

**Figure 8-23: 80C32 Application in 2K Byte mode**

The internal chipselect is activated when the address inputs AB[10..3] of the VPC3+S are set to '0'.

In the example above the start address of the VPC3+S is set to 1000H.

**Processor VPC3+**



ALE

DB[7..0]

address latch

8

AB[2..0]

AB[7..3]

8

AB[10..8]

all bits zero

CS

decoder

11

internal address 1

internal chip select

8

3

5

3

4.0 KB RAM

ALE AD[7..0]

AB[10..8]

AB[15..11]

**Figure 8-24: Internal Chipselect Generation in Synchronous Intel Mode, 2K Byte RAM**

### Application with 80C32 (4K Byte RAM Mode)

**VPC3+**

connect to VDD or GND

GND

LED for Data-Exchange

µC

µC

VDD VDD

VDD

VDD

R 4k7

µC

µC µC

XRD XWR

AB 8

AB 9

AB 10 AB 11

AB 12

AB 13 1

AB 14

AB 15

GND

RS485 RS485

RS485

GND

ADB 0

ADB 1

ADB 2

ADB 3

ADB 4

ADB 5

ADB 6

ADB 7

AB (8 …15)

µC

GND GND

GND

DB (0..7)

µC

DB 0

DB 1

DB 2

DB 3

DB 4

DB 5

DB 6

DB 7

TXD \* RXD RTS

XCTS

MODE XTEST0 XTEST1

ALE XRD XWR

AB 0

AB 1

AB 2

XCS / AB 11 AB 3

AB 4

AB 5

AB 6

AB 7

AB 8

AB 9

AB 10

XREADY

X/INT

RESET

DIVIDER CLKOUT

XDATAEX

CLK

XINT/MOT

22R

48MHz

\* TxD: tristate, external pull-up resistor required

**Figure 8-25: 80C32 Application in 4K Byte mode**

The internal chipselect is activated when the address inputs AB[10..3] of the VPC3+S are set to '0'.

In the example above the start address of the VPC3+S is set to 2000H.

**Processor VPC3+**



ALE

DB[7..0]

address latch

8

AB[2..0] XCS / AB11

AB[6..3]

8

all bits zero

CS

decoder

AB[10..7]

12

internal address 1

internal chip select

8

3

4

4

4.0 KB RAM

ALE AD[7..0]

AB[10..8]

AB 11 AB[15..12]

**Figure 8-26 : Internal Chipselect Generation in Synchronous Intel Mode, 4K Byte RAM**

### Application with 80C165

**VPC3+**

connect to VDD or GND

GND

LED for Data-Exchange

µC µC

µC µC

GND VDD

VDD

VDD

R 4k7

µC

µC

XRD XWRL

AB 0

AB 1

AB 2

AB 3

AB 4

AB 5

AB 6 AB 7

AB 8

AB 9

AB 10

AB 11

RS485 RS485

RS485

GND

DB 0

DB 1

DB 2

DB 3

DB 4

DB 5

DB 6

DB 7

DB (0..7)

µC

AB (0 … 11)

µC

\* TxD: tristate, external pull-up resistor required

DB 0

DB 1

DB 2

DB 3

DB 4

DB 5

DB 6

DB 7

TXD \* RXD RTS

XCTS

XRD XWR

AB 0

AB 1

AB 2

AB 3

AB 4

AB 5

AB 6

AB 7

AB 8

AB 9

AB 10

ALE / AB 11

XREADY

X/INT

RESET XCS

MODE XTEST0 XTEST1

DIVIDER CLKOUT

XDATAEX

CLK

XINT/MOT

22R

48MHz

**Figure 8-27: 80C165 Application**

## Dual Port RAM Controller

The internal 4K Byte RAM of the VPC3+S is a single-port RAM. An integrated Dual-Port RAM controller, however, permits an almost simultaneous access of both ports (bus interface and microsequencer interface). When there is a simultaneous access of both ports, the bus interface has priority. This guarantees the shortest possible access time. If the VPC3+S is connected to a microcontroller with an asynchronous interface, the controller can evaluate the Ready signal.

## UART

The transmitter converts the parallel data structure into a serial data flow. Signal Request-to-Send (RTS) is generated before the first character. The XCTS input is available for connecting a modem. After RTS active, the transmitter must hold back the first telegram character until the modem acti- vates XCTS. XCTS is checked again after each character.

The receiver converts the serial data flow into the parallel data structure and scans the serial data flow with the four-fold transmission speed. Stop bit testing can be switched off for test purposes ('Dis\_Stop\_Control = 1' in Mode Register 0 or Set\_Prm telegram for DP). One requirement of the PROFIBUS protocol is that no rest states are permitted between the telegram characters. The VPC3+S transmitter ensures that this specification is maintained.

The synchronization of the receiver starts with the falling edge of the start bit. The start bit is checked again in the middle of the bit-time for low level. The data bits, the parity and the stop bit are also scanned in the middle of the bit-time. To compensate for the synchronization error, a repeater gen- erates a 25% distortion of the stop bit at a four-fold scan rate. In this case the VPC3+ should be parameterized with 'Dis\_Start\_Control = 1' (in Mode Register 0 or Set\_Prm telegram for DP) in order to increase the permissible distortion of the stop bit.

## ASIC Test

All output pins and I/O pins can be switched to the high-resistance state via the XTEST0 test pin. An additional XTEST1 input is provided to test the chip on automatic test devices (not in the target hardware environment!).

|  |  |  |  |
| --- | --- | --- | --- |
| **Pin** | **Name** | **Value** | **Function** |
| D2 | XTEST0 | GND | All outputs high-resistance |
| VCC | Normal VPC3+ function |
| E5 | XTEST1 | GND | Various test modes |
| VCC | Normal VPC3+ function |

**Figure 8-28: Test Ports**

## Pin Assignment



The data transmission is performed in RS485 operating mode (i.e., physical RS485). The VPC3+S is connected via the following signals to the galvanically isolated interface drivers.

|  |  |  |
| --- | --- | --- |
| **Signal Name** | **Input/Output** | **Function** |
| RTS | Output | Request to send |
| TXD | Output | Sending data, tristate output, pull-up resistor required |
| RXD | Input | Receiving data |

**Figure 9-1: PROFIBUS Signals**

The PROFIBUS interface is a 9-way, sub D, plug connector with the following pin assignment.

Pin 1 - Free Pin 2 - Free Pin 3 - B line

Pin 4 - Request to send (RTS) Pin 5 - Ground 5V (M 5 )

Pin 6 - Potential 5V (floating P5 ) Pin 7 - Free

Pin 8 - A line Pin 9 - Free

The cable shield must be connected to the plug connector housing. The free pins are described as optional in IEC 61158-2.

#### CAUTION:

The pin names A and B on the plug connector refer to the signal names in the RS485 standard and not the pin names of driver ICs.

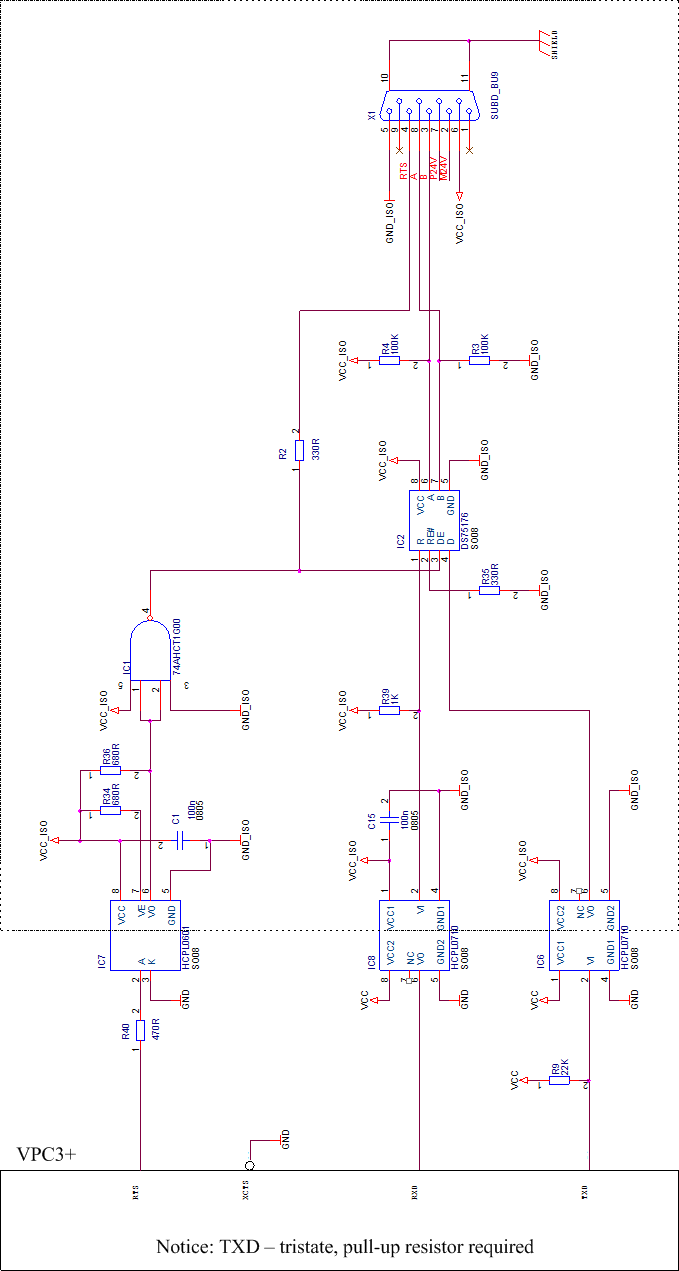
Keep the wires from driver to connector as short as possible.

#### Note:

TXD is tristate output and requires external pull-up resistor for correct operation with common line drivers.

## Example for the RS485 Interface

To minimize the capacity of the bus lines the user should avoid additional capacities. The typical capacity of a bus station should be 15...25 pF.



**Figure 9-2: Example for the RS485 Interface**

## Absolute Maximum Ratings

|  |  |  |  |
| --- | --- | --- | --- |
| **Parameter** | **Symbol** | **Limits** | **Unit** |
| DC supply voltage | VCC | -0.3 to 3.9 | V |
| Input voltage | VI | -0.3 to 5.5 | V |
| Output voltage | VO | -0.3 to VCC +0.3 | V |
| DC output current | IO | See [Figure 10-4](#_bookmark10) | mA |
| Storage temperature | Tstore | -40 to +150 | °C |

**Figure 10-1: Absolute Maximum Ratings**

## Recommended Operating Conditions

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Symbol** | **MIN** | **MAX** | **Unit** |
| DC supply voltage | VCC | 3.00 | 3.6 | V |
| Static supply current | Icc |  | 1001) | µA |
| Circuit ground | GND | 0 | 0 | V |
| Input voltage | VI | 0 | 5.50 | V |
| Input voltage (HIGH level) | VIH | 2.00 | 5.50 | V |
| Input voltage (LOW level) | VIL | 0 | 0.8 | V |
| Output voltage | VO | 0 | VCC | V |
| Ambient temperature | TA | -40 | +85 | °C |

1) : Static IDD current is exclusively of input/output drive requirements and is measured with the clock stopped and all inputs tied to VCC or GND.

**Figure 10-2: Recommended Operating Conditions**

## General DC Characteristics

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Parameter** | **Symbol** | **MIN** | **TYP** | **MAX** | **Unit** |
| Input LOW current | IIL | -1 |  | +1 | µA |
| Input HIGH current | IIH | -1 |  | +1 | µA |
| Tri-state leakage current | IOZ | -10 |  | +10 | µA |
| Current consumption (3.3V) | IA |  | 30 |  | mA |
| Input capacitance | CIN |  | 5 |  | pF |
| Output capacitance | COUT |  | 5 |  | pF |
| Bi-directional buffer capacitance | CBID |  | 5 |  | pF |
| Thermal Resist. (BGA48) | ΘJA |  | 43.6 |  | K/W |
| Thermal Resist. (QFP48) | ΘJA |  | 72.2 |  | K/W |

**Figure 10-3: General DC Characteristics**

## Ratings for the Output Drivers

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Signal** | **Direction** | **Driver**  **Type** | **Driver Strength** | **Max. Cap.**  **Load** |
| DB 0-7 | I/O | Tristate | 8mA | 50pF |
| RTS | O | Push/Pull | 8mA | 50pF |
| TXD | O | Tristate | 8mA | 50pF |
| INT | O | Push/Pull | 8mA | 50pF |
| XREADY/XDTACK SPI\_MISO I2C\_SDA | O O  I/O | Push/Pull Push/Pull Tristate | 8mA | 50pF |
| XDATAEXCH | O | Push/Pull | 8mA | 50pF |
| CLKOUT | O | Push/Pull | 8mA | 50pF |

**Figure 10-4: Ratings for the Output Drivers**

## DC Electrical Characteristics

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Parameter** | **Symbol** | **MIN** | **TYP** | **MAX** | **Unit** |
| DC supply voltage | VCC | 3.00 | 3.30 | 3.60 | V |
| Input voltage LOW level | VIL |  |  | 0.8 | V |
| Input voltage HIGH level | VIH | 2.0 |  |  | V |
| Output voltage LOW level | VOL |  |  | 0.4 | V |
| Output voltage HIGH level | VOH | 2.4 |  |  | V |
| Schmitt Trigger  negative going threshold voltage | VT- | 0.9 | 1.1 |  | V |
| Schmitt Trigger  positive going threshold voltage | VT+ |  | 1.6 | 1.9 | V |
| Input LOW current | IIL | -1 |  | +1 | µA |
| Input HIGH current | IIH | -1 |  | +1 | µA |
| Tri-state leakage current | IOZ | -10 | ±1 | +10 | µA |
| Output current LOW level, 8mA cell | IOL | +8 |  |  | mA |
| Output current HIGH level, 8mA cell | IOH | -8 |  |  | mA |

**Figure 10-5: DC Specification of I/O Drivers for 3.3V Operation**

#### Notes:

The VPC3+S is equipped with 5V tolerant inputs.

## Timing Characteristics

All signals beginning with ‘X’ are ‘low active’. All timing values are based on the capacitive loads specified in the table above.

### System Bus Interface

**Clock**

Clock frequency is 48 MHz. Distortion of the clock signal is permissible up to a ratio of 30:70 at the threshold levels 0.9 V and 2.1 V.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Symbol** | **MIN** | **MAX** | **Unit** |
| Clock period | T | 20.83 | 20.83 |  |
| Clock high time | TCH | 6.25 | 14.6 | ns |
| Clock low time | TCL | 6.25 | 14.6 | ns |
| Clock rise time | TCR |  | 4 | ns |
| Clock fall time | TCF |  | 4 | ns |

**Figure 10-6: Clock Timing**

#### Note:

The VPC3+S is equipped with 5V tolerant inputs.

### Interrupt:

After acknowledging an interrupt with EOI, the interrupt output of the VPC3+S is deactivated for at least 1 us or 1 ms depending on the bit EOI\_Time\_Base in Mode Register 0.

|  |  |  |  |
| --- | --- | --- | --- |
| **Parameter** | **MIN** | **MAX** | **Unit** |
| Interrupt inactive time EOI\_Timebase = ‘0’ | 1 | 1 | µs |
| Interrupt inactive time EOI\_Timebase = ‘1’ | 1 | 1 | ms |

**Figure 10-7: End-of-Interrupt Timing**

### Reset:

VPC3+S requires a minimum reset phase of 100 ns at power-on.

### Timing in the Synchronous Intel Mode

In the synchronous Intel mode, the VPC3+S latches the least significant addresses with the falling edge of ALE. At the same time, the VPC3+S expects the most significant address bits on the address bus. An internal chipselect signal is generated from the most significant address bits. The request for an access to the VPC3+S is generated from the falling edge of the read signal (XRD) and from the rising edge of the write signal (XWR).

ALE



1

7

2

5

3

8

valid

9

address

address

6

10

data valid

4

valid

AB10..0

DB7..0

XRD

**Figure 10-8: Synchronous Intel Mode, READ (XWR = 1)**

ALE



1

14

11

3

8

valid

15

address

address

13

12

10

data valid

15

valid

AB10..0

DB7..0

XWR

**Figure 10-9: Synchronous Intel Mode, WRITE (XRD = 1)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Parameter** | **MIN** | **MAX** | **Unit** |
| 1 | ALE pulsewidth | 10 |  | ns |
| 2 | ALE  to XRD  | 20 |  | ns |
| 3 | Address to ALE  setuptime | 10 |  | ns |
| 4 | Address holdtime after ALE  | 0 |  | ns |
| 5 | XRD  to data valid |  | 83 | ns |
| 6 | XRD pulsewidth | 105 |  | ns |
| 7 | XRD  to ALE  | 10 |  | ns |
| 8 | address (AB7..0) holdtime after XRD/XWR  | 0 |  | ns |
| 9 | data holdtime after XRD  | 3 | 12 | ns |
| 10 | XRD / XWR cycletime | 155 |  | ns |
| 11 | ALE  to XWR  | 20 |  | ns |
| 12 | XWR pulsewidth | 83 |  | ns |
| 13 | data setuptime to XWR  | 10 |  | ns |
| 14 | XWR  to ALE  | 10 |  | ns |
| 15 | data holdtime after XWR  | 0 |  | ns |

**Figure 10-10: Timing, Synchronous Intel Mode**

### Timing in the Asynchronous Intel Mode

In the asynchronous Intel mode, the VPC3+S acts like a memory with ready logic. The access time depends on the type of access. The request for an access to the VPC3+S is generated from the falling edge of the read signal (XRD) or the rising edge of the write signal (XWR).

The VPC3+S generates the Ready signal synchronously to the system clock. The Ready signal gets inactive when the read or the write signal is deactivated. The data bus is switched to Tristate with XRD = '1'.

AB10..0



valid

16

23

17

24

18

25

19

26

20

21

27

22

data valid

DB7..0

XRD

XCS

XREADY

(normal)

XREADY

(early)

**Figure 10-11: Asynchronous Intel Mode, READ (XWR = 1)**

AB10..0



valid

16

23

28

30

29

25

19

26

20

21

27

22

data valid

DB7..0

XWR

XCS

XREADY

(normal)

XREADY

(early)

**Figure 10-12: Asynchronous Intel Mode, WRITE (XRD = 1)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Parameter** | **MIN** | **MAX** | **Unit** |
| 16 | address-setuptime to XRD / XWR  | 0 |  | ns |
| 17 | XRD  to data valid |  | 83 | ns |
| 18 | XRD pulsewidth | 105 |  | ns |
| 19 | XCS  setuptime to XRD / XWR  | 0 |  | ns |
| 20 | XRD  to XREADY  (Normal-Ready) |  | 125 | ns |
| 21 | XRD  to XREADY  (Early-Ready) |  | 104 | ns |
| 22 | XRD / XWR cycletime | 125 |  | ns |
| 23 | address holdtime after XRD / XWR  | 0 |  | ns |
| 24 | data holdtime after XRD  | 3 | 12 | ns |
| 25 | read/write inactive time | 10 |  | ns |
| 26 | XCS holdtime after XRD / XWR  | 0 |  | ns |
| 27 | XREADY holdtime after XRD / XWR | 3 | 15 | ns |
| 28 | data setuptime to XWR  | 10 |  | ns |
| 29 | XWR pulsewidth | 83 |  | ns |
| 30 | data holdtime after XWR  | 0 |  | ns |

**Figure 10-13: Timing, Asynchronous Intel Mode**

### Timing in the Synchronous Motorola Mode

If the CPU is clocked by the VPC3+S, the output clock pulse (CLKOUT 2/4) must be 4 times larger than the E\_Clock. That is, a clock pulse signal must be present at the CLK input that is at least 10 times larger than the desired system clock pulse (E\_Clock). The Divider-Pin must be connected to ‘0’ (divider 4). This results in an E\_Clock of 3 MHz.

The request for a read access to the VPC3+S is derived from the rising edge of the E\_Clock (in addition: XCS = 0, R\_W = 1). The request for a write access is derived from the falling edge of the E\_Clock (in addition: XCS = 0, R\_W = 0).

E\_CLOCK



31

33

32

37

valid

38

data valid

35

39

36

40

AB10..0

DB7..0

R\_W

XCS

**Figure 10-14: Synchronous Motorola-Mode, READ (AS = 1)**

E\_CLOCK



31

33

37

valid

42

data valid

35

39

36

40

41

AB10..0

DB7..0

R\_W

XCS

**Figure 10-15: Synchronous Motorola-Mode, WRITE (AS = 1)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Parameter** | **MIN** | **MAX** | **Unit** |
| 31 | E\_Clock pulse width | 136.7 |  | ns |
| 33 | Address setuptime (A10..0) to E\_Clock  | 10 |  | ns |
| 37 | Address holdtime after E\_Clock  | 0 |  | ns |
| 32 | E\_Clock  to Data valid |  | 83 | ns |
| 38 | Data holdtime after E\_Clock  | 3 | 12 | ns |
| 35 | R\_W setuptime to E\_Clock  | 10 |  | ns |
| 39 | R\_W holdtime after E\_Clock  | 5 |  | ns |
| 36 | XCS setuptime to E\_Clock  | 0 |  | ns |
| 40 | XCS holdtime after E\_Clock  | 0 |  | ns |
| 41 | Data setuptime to E\_Clock  | 10 |  | ns |
| 42 | Data holdtime after E\_Clock  | 0 |  | ns |

**Figure 10-16: Timing, Synchronous Motorola Mode**

### Timing in the Asynchronous Motorola Mode

In the asynchronous Motorola mode, the VPC3+S acts like a memory with Ready logic, whereby the access times depend on the type of access.

The request for an access of the VPC3+S is generated from the falling edge of the AS signal (in addition: XCS = '0', R\_W = '1'). The request for a write access is generated from the rising edge of the AS signal (in addition: XCS = '0', R\_W = '0').

AB10..0



valid

43

52

44

53

45

54

46

55

47

56

48

49

57

50

51

data valid

DB7..0

AS

R\_W

XCS

XDTACK

(normal)

XDTACK

(early)

**Figure 10-17: Asynchronous Motorola Mode, READ (E\_CLOCK = 0)**

AB10..0



valid

43

52

58

60

59

54

46

55

47

56

48

49

57

50

51

data valid

DB7..0

AS

R\_W

XCS

XDTACK

(normal)

XDTACK

(early)

**Figure 10-18: Asynchronous Motorola Mode (WRITE)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Parameter** | **MIN** | **MAX** | **Unit** |
| 43 | address setuptime to AS  | 0 |  | ns |
| 44 | AS  to data valid |  | 83 | ns |
| 45 | AS pulsewidth (read access) | 115 |  | ns |
| 46 | R\_W  setuptime to AS  | 10 |  | ns |
| 47 | XCS  setuptime to AS  | 5 |  | ns |
| 48 | AS  to XDTACK  (Normal-Ready) |  | 125 | ns |
| 49 | AS  to XDTACK  (Early-Ready) |  | 104 | ns |
| 50 | last AS  to XCS  | 93 |  | ns |
| 51 | AS cycletime | 125 |  | ns |
| 52 | address holdtime after AS  | 0 |  | ns |
| 53 | Data holdtime after AS  | 3 | 12 | ns |
| 54 | AS inactive time | 10 |  | ns |
| 55 | R\_W holdtime after AS  | 10 |  | ns |
| 56 | XCS holdtime after AS  | 0 |  | ns |
| 57 | XDTACK holdtime after AS  | 3 | 15 | ns |
| 58 | Data setuptime to AS  | 10 |  | ns |
| 59 | AS pulsewidth (write access) | 83 |  | ns |
| 60 | Data holdtime after AS  | 0 |  | ns |

**Figure 10-19: Timing, Asynchronous Motorola Mode**

### Timing in SPI Interface Mode

**XSS**



**t HIGH.XSS**

**t S.XSS**

**t HIGH.SCK**

**t LOW.SCK**

**t V.SO**

**t H.SO**

**t DIS.SO**

**HIGH IMPEDANCE**

**HIGH IMP.**

**VALID OUT**

**t S.SI t H.SI**

**VALID IN**

**SCK**

**(CPOL=0)**

**SCK**

**(CPOL=1)**

**MISO**

**MOSI**

**Figure 10-20: Timing Diagram SPI Interface Mode (CPHA='0')**

**XSS**



**t HIGH.XSS**

**t S.XSS**

**t HIGH.SCK**

**t LOW.SCK**

**t V.SO**

**t H.SO**

**t DIS.SO**

**HIGH IMPEDANCE**

**HIGH IMP.**

**VALID OUT**

**t S.SI t H.SI**

**VALID IN**

**SCK**

**(CPOL=0)**

**SCK**

**(CPOL=1)**

**MISO**

**MOSI**

**Figure 10-21: Timing Diagram SPI Interface Mode (CPHA='1')**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Symbol** | **Parameter** | **MIN** | **MAX** | **Unit** |
| f SCK | Clock Frequency, SCK |  | 6 | MHz |
| t LOW.SCK | Clock Pulse Width Low | 83 |  | ns |
| t HIGH.SCK | Clock Pulse Width High | 83 |  | ns |
| t S.XSS | XSS Setup Time | 83 |  | ns |
| t V.SO | Clock to Data Out Valid |  | 76 | ns |
| t H.SO | Data Out Hold Time | 21 |  | ns |
| t S.SI | Data In Set-up Time | 10 |  | ns |
| t H.SI | Data In Hold Time | 10 |  | ns |
| t DIS.SO | Output Disable Time |  | 83 | ns |
| t HIGH.XSS | XSS Inactive (High) Time | 83 |  | ns |

**Figure 10-22: Timing, SPI Interface Mode**

### Timing in I2C Interface Mode

**t HIGH t LOW**



**t S.STA t H.STA**

**t H.DAT t S.DAT**

**t**

**S.STO**

**t AA**

**t DH**

**SCK**

**SDA IN**

**SDA OUT**

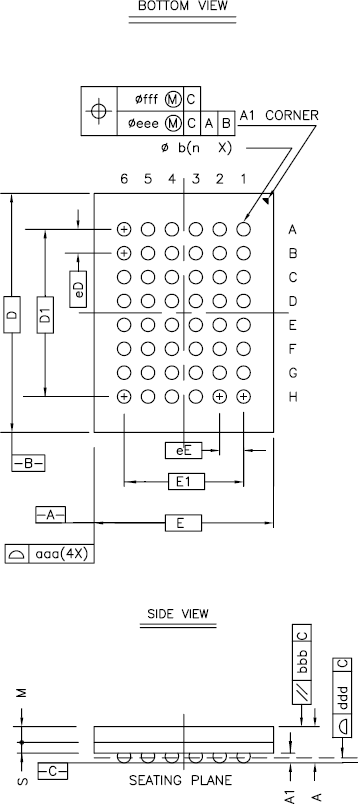
**Figure 10-23: Timing Diagram I2C Interface Mode**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Symbol** | **Parameter** | **MIN** | **MAX** | **Unit** |
| f SCK | Clock Frequency, SCK |  | 6 | MHz |
| t LOW | Clock Pulse Width Low | 83 |  | ns |
| t HIGH | Clock Pulse Width High | 83 |  | ns |
| t AA | Clock Low to Data Out Valid |  | 76 | ns |
| t H.STA | Start Condition Hold Time | 21 |  | ns |
| t S.STA | Start Condition Set-up Time | 21 |  | ns |
| t H.DAT | Data In Hold Time | 10 |  | ns |
| t S.DAT | Data In Set-up Time | 10 |  | ns |
| t S.STO | Stop Condition Set-up Time | 21 |  | ns |
| t DH | Data Out Hold Time | 21 |  | ns |

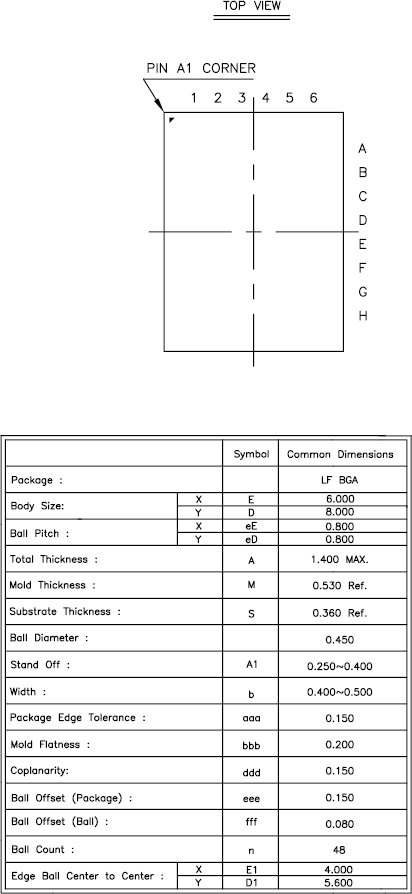
**Figure 10-24: Timing, I2C Interface Mode**

## Package Specifications

### LFBGA48

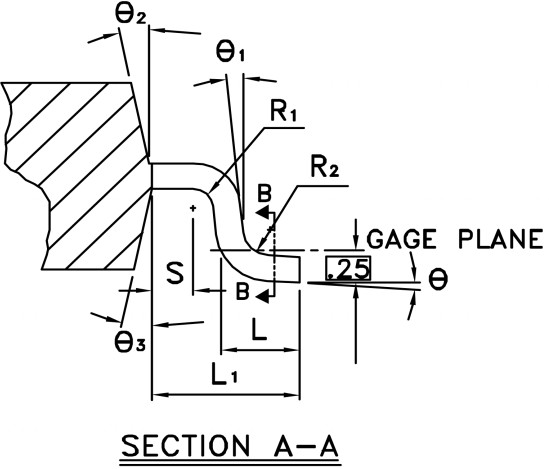
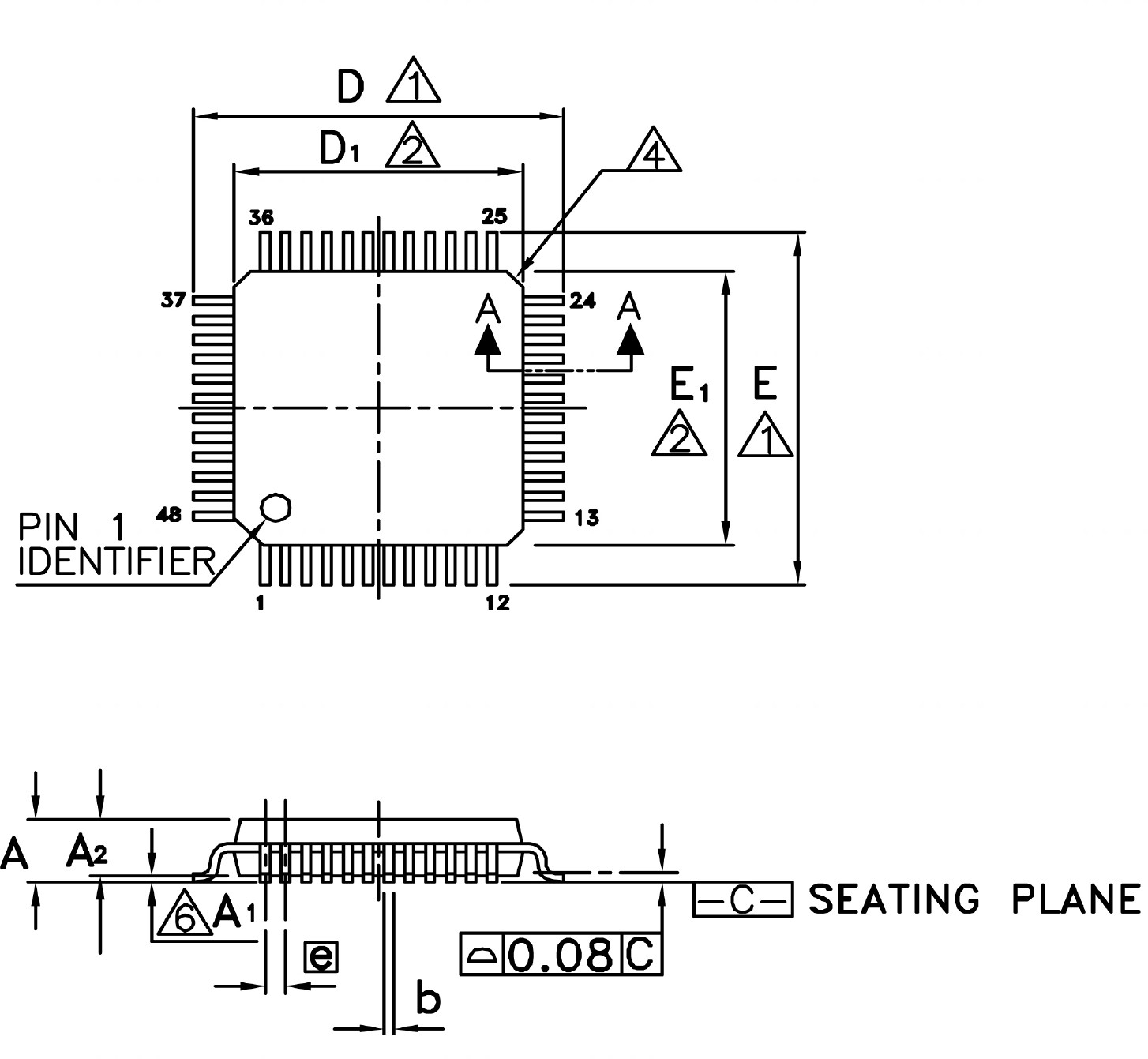


**Figure 10-25: LFBGA48 Package Drawing**



**Figure 10-26 : LFBGA48 Package Dimensions and Tolerances**

### LQFP48



**Figure 10-27: LQFP48 Package Drawing**

|  |  |  |  |
| --- | --- | --- | --- |
| **Symbol** | **Dimensions in mm** | | |
| **MIN** | **NOM** | **MAX** |
| A |  |  | 1.60 |
| A1 | 0.05 |  | 0.15 |
| A2 | 1.35 | 1.40 | 1.45 |
| b | 0.17 | 0.22 | 0.27 |
| b1 | 0.17 | 0.20 | 0.23 |
| c | 0.09 |  | 0.20 |
| c1 | 0.09 |  | 0.16 |
| D | 9.00 BSC | | |
| D1 | 7.00 BSC | | |
| E | 9.00 BSC | | |
| E1 | 7.00 BSC | | |
| e | 0.50 BSC | | |
| L | 0.45 | 0.60 | 0.75 |
| L1 | 1.00 REF | | |
| R1 | 0.08 |  |  |
| R2 | 0.08 |  | 0.20 |
| S | 0.20 |  |  |
| Θ | 0° | 3.5° | 7° |
| Θ1 | 0° |  |  |
| Θ2 | 12° TYP | | |
| Θ3 | 12° TYP | | |

**Figure 10-28 : LQFP48 Package Dimensions and Tolerances**

## Processing Instructions

Generally, ESD protective measures must be maintained for all electronic components. The VPC3+S is a cracking-endangered component that must be handled properly.

Profichip products are tested and classified for moisture sensitivity according to the procedures outlined by JEDEC. The VPC3+S is classified as moisture sensitivity level (MSL) 3.

In order to minimize any potential risk caused by moisture trapped inside non-hermetic packages it is a general recommendation to perform a drying process before soldering.

## Ordering Information

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version /**  **Part Number** | **Order Code** | **Package** | **Temperature Range** | **Notes** |
| VPC3+S-BGA48 | PALF2009 | LFBGA48 | Industrial  (-40°C to +85°C) |  |
| VPC3+S-QFP48 | PALF2012 | LQFP48 | Industrial  (-40°C to +85°C) |  |

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Page** | **Remarks** |
| V1.00 | 01.04.2009 |  | First release |
| V1.01 | 10.08.2009 | 29 | Description of GC\_Int\_Mode\_Ext in Mode Register 3 corrected |
|  |  | 72-77 | Some hints for configuration of PLL added |
|  |  | 96 | Name of I2C clock changed from SCL to SCK |
|  |  | 111 | Current consumption and thermal resistance added |
| V1.02 | 25.05.2010 | 92 | SPI instruction “WRITE ARRAY” added to Figure 8-6 |
|  |  | 95 | Instruction coding in Figure 8-10 “WRITE ARRAY Sequence” corrected |
| V1.03 | 28.05.2010 | 122 | Timing table for SPI interface mode corrected (MIN/MAX values swapped) |
|  |  | 123 | Timing table for I2C interface mode corrected (MIN/MAX values swapped) |
| V1.04 | 10.10.2012 | 10-17 | Pin assignment of LQFP48 package version added |
|  |  | 111 | Thermal resistance of LQFP48 package added |
|  |  | 128-129 | LQFP48 package drawing added |
|  |  | 130 | Processing instructions revised and ordering information added |
| V1.05 | 18.02.2014 | 10-11 | AB11 (pin 3) added to pinout figure and pin assignment table |
|  |  | 130 | Part number added to ordering information |
| V1.06 | 18.07.2014 | 13-16 | Notes regarding external pull-up on TXD added |
|  |  | 89, 90 | Modification of signal names |
|  |  | 105-107, | Modification of figure 8-23, 8-24, 8-25, 8-26, 8-27, 9-2 |
|  |  | 109,110 |  |

profichip GmbH Einsteinstrasse 6

91074 Herzogenaurach Germany

Phone : +49.9132.744-200

Fax: +49.9132.744-2164

[www.profichip.com](http://www.profichip.com/)