YASKAWA

# VPC3+S
# User Manual

**Revision 1.09**

1 Introduction ... 10

2 Functional Description ... 11

2.1 Overview ... 11

3 Pin Description ... 13

3.1 Pinout ... 13
3.2 Pin Assignment (Overview) ... 15
3.2.1 Asynchronous Intel Mode ... 17
3.2.2 Synchronous Intel Mode ... 18
3.2.3 Asynchronous Motorola Mode ... 19
3.2.4 Synchronous Motorola Mode ... 20
3.2.5 SPI Mode ... 21
3.2.6 $I^2C$ Mode ... 21

4 Memory Organization ... 22

4.1 Overview ... 22
4.2 Control Parameters (Latches/Registers) ... 24
4.3 Organizational Parameters (RAM) ... 26

5 ASIC Interface ... 28

5.1 Mode Registers ... 28
5.1.1 Mode Register 0 ... 28
5.1.2 Mode Register 1 ... 30
5.1.3 Mode Register 2 ... 32
5.1.4 Mode Register 3 ... 33
5.2 Status Register ... 34
5.3 Interrupt Controller ... 36
5.3.1 Interrupt Request Register ... 37
5.3.2 Interrupt Acknowledge / Mask Register ... 38
5.4 Watchdog Timer ... 39
5.4.1 Automatic Baud Rate Identification ... 39
5.4.2 Baud Rate Monitoring ... 39
5.4.3 Response Time Monitoring ... 40

6 PROFIBUS DP Interface ... 41

6.1 DP Buffer Structure ... 41
6.2 Description of the DP Services ... 44
6.2.1 Set_Slave_Add (SAP 55) ... 44
6.2.2 Set _Prm (SAP 61) ... 46
6.2.3 Chk_Cfg (SAP 62) ... 49
6.2.4 Slave_Diag (SAP 60) ... 51
6.2.5 Write_Read_Data / Data_Exchange (Default_SAP) ... 53

6.2.6 Global_Control (SAP 58) ....58
6.2.7 RD_Input (SAP 56) ....59
6.2.8 RD_Output (SAP 57) ....59
6.2.9 Get_Cfg (SAP 59) ....59

7 PROFIBUS DP Extensions ....60

7.1 Set_(Ext_)Prm (SAP 53 / SAP 61) ....60
7.2 PROFIBUS DP-V1 ....62
7.2.1 Acyclic Communication Relationships ....62
7.2.2 Diagnosis Model ....64
7.3 PROFIBUS DP-V2 ....65
7.3.1 DXB (Data eXchange Broadcast) ....65
7.3.2 IsoM (Isochronous Mode) ....70
7.3.2.1 IsoM-PLL ....74
7.3.3 CS (Clock Synchronization) ....80

8 Hardware Interface ....85

8.1 Universal Processor Bus Interface ....85
8.1.1 Overview ....85
8.1.2 Parallel Interface Modes ....86
8.1.3 SPI Interface Mode ....90
8.1.4 I2C Interface Mode ....98
8.1.5 Application Examples (Principles) ....103
8.1.6 Application with 80C32 (2K Byte RAM Mode) ....105
8.1.7 Application with 80C32 (4K Byte RAM Mode) ....106
8.1.8 Application with 80C165 ....107
8.2 Dual Port RAM Controller ....107
8.3 UART ....108
8.4 ASIC Test ....108

9 PROFIBUS Interface ....109

9.1 Pin Assignment ....109
9.2 Example for the RS485 Interface ....110

10 Operational Specifications ....111

10.1 Absolute Maximum Ratings ....111
10.2 Recommended Operating Conditions ....111
10.3 General DC Characteristics ....111
10.4 Ratings for the Output Drivers ....112
10.5 DC Electrical Characteristics ....112
10.6 Timing Characteristics ....113
10.6.1 System Bus Interface ....113
10.6.2 Timing in the Synchronous Intel Mode ....114
10.6.3 Timing in the Asynchronous Intel Mode ....116

10.6.4 Timing in the Synchronous Motorola Mode ........ 118
10.6.5 Timing in the Asynchronous Motorola Mode ........ 120
10.6.6 Timing in SPI Interface Mode ........ 122
10.6.7 Timing in I2C Interface Mode ........ 123
10.7 Package Specifications ........ 124
10.7.1 LFBGA48 ........ 124
10.7.2 LQFP48 ........ 126
10.8 Processing Instructions ........ 128
10.9 Ordering Information ........ 128
10.10 Reflow Soldering Profile ........ 129

**11 Revision History ........ 130**

# List of Figures


Figure 3-1: VPC3+S LFBGA48 Pinout (TOP VIEW) ........ 13
Figure 3-2: VPC3+S LQFP48 Pinout (TOP VIEW) ........ 14
Figure 3-3: Pin Assignment ........ 16
Figure 3-4: Interface Configuration: Asynchronous Intel Mode ........ 17
Figure 3-5: Interface Configuration: Synchronous Intel Mode ........ 18
Figure 3-6: Interface Configuration: Asynchronous Motorola Mode ........ 19
Figure 3-7: Interface Configuration: Synchronous Motorola Mode ........ 20
Figure 3-8: Interface Configuration: SPI Mode ........ 21
Figure 3-9: Interface Configuration: I2C Mode ........ 21
Figure 4-1: Memory Table ........ 22
Figure 4-2: Assignment of the Internal Parameter-Latches for READ ........ 24
Figure 4-3: Assignment of the Internal Parameter-Latches for WRITE ........ 25
Figure 4-4: Assignment of the Organizational Parameters ........ 27
Figure 5-1: Coding of Mode Register 0, Low-Byte ........ 29
Figure 5-2: Coding of Mode Register 0, High-Byte ........ 30
Figure 5-3: Coding of Mode Register 1 ........ 31
Figure 5-4: Coding of Mode Register 2 ........ 32
Figure 5-5: Coding of Mode Register 3 ........ 33
Figure 5-6: Status Register, Low-Byte ........ 35
Figure 5-7: Status Register, High-Byte ........ 35
Figure 5-8: Block Diagram of Interrupt Controller ........ 36
Figure 5-9: Interrupt-Request-Register, Low-Byte ........ 37
Figure 5-10: Interrupt Request Register, High-Byte ........ 38
Figure 5-11: Interrupt Acknowledge / Mask Register ........ 38
Figure 5-12: Watchdog State Machine (WD_SM) ........ 39
Figure 6-1: DP_SAP Buffer Structure ........ 42
Figure 6-2: Aux-Buffer Management ........ 43
Figure 6-3: Coding of SSA_Buffer_Free_Command ........ 44
Figure 6-4: Structure of the Set_Slave_Add Telegram ........ 45
Figure 6-5: Format of the Set_Prm Telegram ........ 46
Figure 6-6: Spec_User_Prm_Byte / DPV1_Status_1 ........ 47
Figure 6-7: Coding of User_Prm_(Not)_Okay_Cmd ........ 48
Figure 6-8: Coding of User_Cfg_(Not)_Okay_Cmd ........ 50
Figure 6-9: Diagnosis Buffer Assignment ........ 51
Figure 6-10: Coding of New_Diag_Cmd ........ 52
Figure 6-11: Format of the Diagnosis-Buffer ........ 52
Figure 6-12: Dout-Buffer Management ........ 54
Figure 6-13: Coding of Next_Dout_Buf_Cmd ........ 55
Figure 6-14: Din-Buffer Management ........ 56
Figure 6-15: Coding of New_Din_Buf_Cmd ........ 56
Figure 6-16: Format of the Global_Control Telegram ........ 58
Figure 7-1: Set_Prm with DPV1_Status bytes ........ 60
Figure 7-2 : Format of the Structured_Prm_Data block ........ 61

Figure 7-3: SAP-List entry ........ 62
Figure 7-4: Buffer Header ........ 63
Figure 7-5: acyclic communication sequence ........ 64
Figure 7-6: FDL-Interface of VPC3+S (e.g. same Buffer for Indication and Response) ........ 64
Figure 7-7 : Overview DXB ........ 65
Figure 7-8: Format of the Structured_Prm_Data with DXB Linktable (specific link is grey scaled) ........ 66
Figure 7-9: Format of the Structured_Prm_Data with DXB Subscribertable (specific link is grey scaled) ........ 66
Figure 7-10: DXB_Link_Status_Buf (specific link is grey scaled) ........ 67
Figure 7-11: DXBout-Buffer ........ 68
Figure 7-12: DXBout-Buffer Management ........ 68
Figure 7-13: Coding of Next_DXBout_Buf_Cmd ........ 69
Figure 7-14: Link_Status handling ........ 69
Figure 7-15: Telegram sequences in IsoM with one DP-Master (Class 1) ........ 70
Figure 7-16: IsoM SYNCH telegram ........ 70
Figure 7-17: SYNC-signal and interrupts for synchronization modes ........ 71
Figure 7-18: Format of Set_Prm telegram for IsoM ........ 72
Figure 7-19: Format of Set_Prm for DP-Slave using isochronous cycles ........ 73
Figure 7-20: SYNC clock and status signals of PLL ........ 74
Figure 7-21: Inputs and outputs of the PLL ........ 75
Figure 7-22: Format of Structured_Prm_Data with IsoM Parameter ........ 76
Figure 7-23: Format of the PLL_Buffer ........ 79
Figure 7-24: configuration of $T_{PLL\_O}$ and $T_{PLL\_I}$ ........ 79
Figure 7-25: Start up of PLL (grey scaled task omitted if SYNC_Mode=0) ........ 79
Figure 7-27: Clock Synchronization Mechanism ........ 80
Figure 7-28: Format of Structured_Prm_Data with Time AR ........ 81
Figure 7-29: Format of Clock_Value ........ 81
Figure 7-30: Format of the Clock_Sync-Buffer ........ 84
Figure 7-31: communication scheme ........ 84
Figure 8-1: Configuration of the parallel Processor Interface Modes ........ 86
Figure 8-2: Microprocessor Bus Signals ........ 87
Figure 8-3: SPI Master-Slave-Transfer (Block Diagram) ........ 90
Figure 8-4: SPI Transfer Format (CPHA='0') ........ 91
Figure 8-5: SPI Transfer Format (CPHA='1') ........ 92
Figure 8-6: SPI Instruction Set ........ 93
Figure 8-7: READ BYTE Sequence ........ 94
Figure 8-8: READ ARRAY Sequence ........ 95
Figure 8-9: WRITE BYTE Sequence ........ 96
Figure 8-10: WRITE ARRAY Sequence ........ 97
Figure 8-11: Bit Transfer on the I2C bus ........ 98
Figure 8-12: START and STOP condition ........ 98
Figure 8-13: Data Transfer on the I2C Bus ........ 99
Figure 8-14: Control Byte Format ........ 99

Figure 8-15: Address Sequence Bit Assignments ....................................................100
Figure 8-16: I2C WRITE Sequence ....................................................100
Figure 8-17: I2C Current Address READ Operation ....................................................101
Figure 8-18: I2C Random READ Operation ....................................................101
Figure 8-19: I2C Sequential READ Operation ....................................................102
Figure 8-20: Low Cost System with 80C32 ....................................................103
Figure 8-21: 80C32 System with External Memory....................................................103
Figure 8-22: 80286 System (X86 Mode) ....................................................104
Figure 8-23: 80C32 Application in 2K Byte mode ....................................................105
Figure 8-24: Internal Chipselect Generation in Synchronous Intel Mode, 2K Byte RAM .....105
Figure 8-25: 80C32 Application in 4K Byte mode ....................................................106
Figure 8-26 : Internal Chipselect Generation in Synchronous Intel Mode, 4K Byte RAM ...106
Figure 8-27: 80C165 Application ....................................................107
Figure 8-28: Test Ports ....................................................108
Figure 9-1: PROFIBUS Signals ....................................................109
Figure 9-2: Example for the RS485 Interface ....................................................110
Figure 10-1: Absolute Maximum Ratings ....................................................111
Figure 10-2: Recommended Operating Conditions....................................................111
Figure 10-3: General DC Characteristics ....................................................111
Figure 10-4: Ratings for the Output Drivers....................................................112
Figure 10-5: DC Specification of I/O Drivers for 3.3 V Operation....................................................112
Figure 10-6: Clock Timing....................................................113
Figure 10-7: End-of-Interrupt Timing....................................................113
Figure 10-8: Synchronous Intel Mode, READ (XWR = 1)....................................................114
Figure 10-9: Synchronous Intel Mode, WRITE (XRD = 1)....................................................114
Figure 10-10: Timing, Synchronous Intel Mode ....................................................115
Figure 10-11: Asynchronous Intel Mode, READ (XWR = 1) ....................................................116
Figure 10-12: Asynchronous Intel Mode, WRITE (XRD = 1) ....................................................117
Figure 10-13: Timing, Asynchronous Intel Mode ....................................................117
Figure 10-14: Synchronous Motorola-Mode, READ (AS = 1) ....................................................118
Figure 10-15: Synchronous Motorola-Mode, WRITE (AS = 1)....................................................118
Figure 10-16: Timing, Synchronous Motorola Mode ....................................................119
Figure 10-17: Asynchronous Motorola Mode, READ (E_CLOCK = 0) ....................................................120
Figure 10-18: Asynchronous Motorola Mode (WRITE) ....................................................121
Figure 10-19: Timing, Asynchronous Motorola Mode ....................................................121
Figure 10-20: Timing Diagram SPI Interface Mode (CPHA='0')....................................................122
Figure 10-21: Timing Diagram SPI Interface Mode (CPHA='1')....................................................122
Figure 10-22: Timing, SPI Interface Mode ....................................................123
Figure 10-23: Timing Diagram I2C Interface Mode....................................................123
Figure 10-24: Timing, I2C Interface Mode....................................................123
Figure 10-25: LFBGA48 Package Drawing ....................................................124
Figure 10-26 : LFBGA48 Package Dimensions and Tolerances....................................................125
Figure 10-27: LQFP48 Package Drawing ....................................................126
Figure 10-28 : LQFP48 Package Dimensions and Tolerances ....................................................127

# List of Tables

Table 10-1: Ordering information........................................................................128
Table 10-2: Reflow soldering profile ....................................................................129
Table 11-1: Revision History ..............................................................................130

# 1 Introduction

Profichip's **VPC3+S** is a communication chip with 8-Bit parallel processor interface for intelligent PROFIBUS DP-Slave applications. Alternatively, an SPI or $I^2C$ 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 addition, 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 synchronized 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 3.3 V single supply voltage. All inputs are 5 V 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 input/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.

# 2 Functional Description

## 2.1 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 $I^2C$ 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 start 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 communication 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/processors. 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 $I^2C$ 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 acknowledge 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 DP-V2 protocol extension.

# 3 Pin Description

## 3.1 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 configurable clock phase and clock polarity are supported. Please see the following chapters for details.

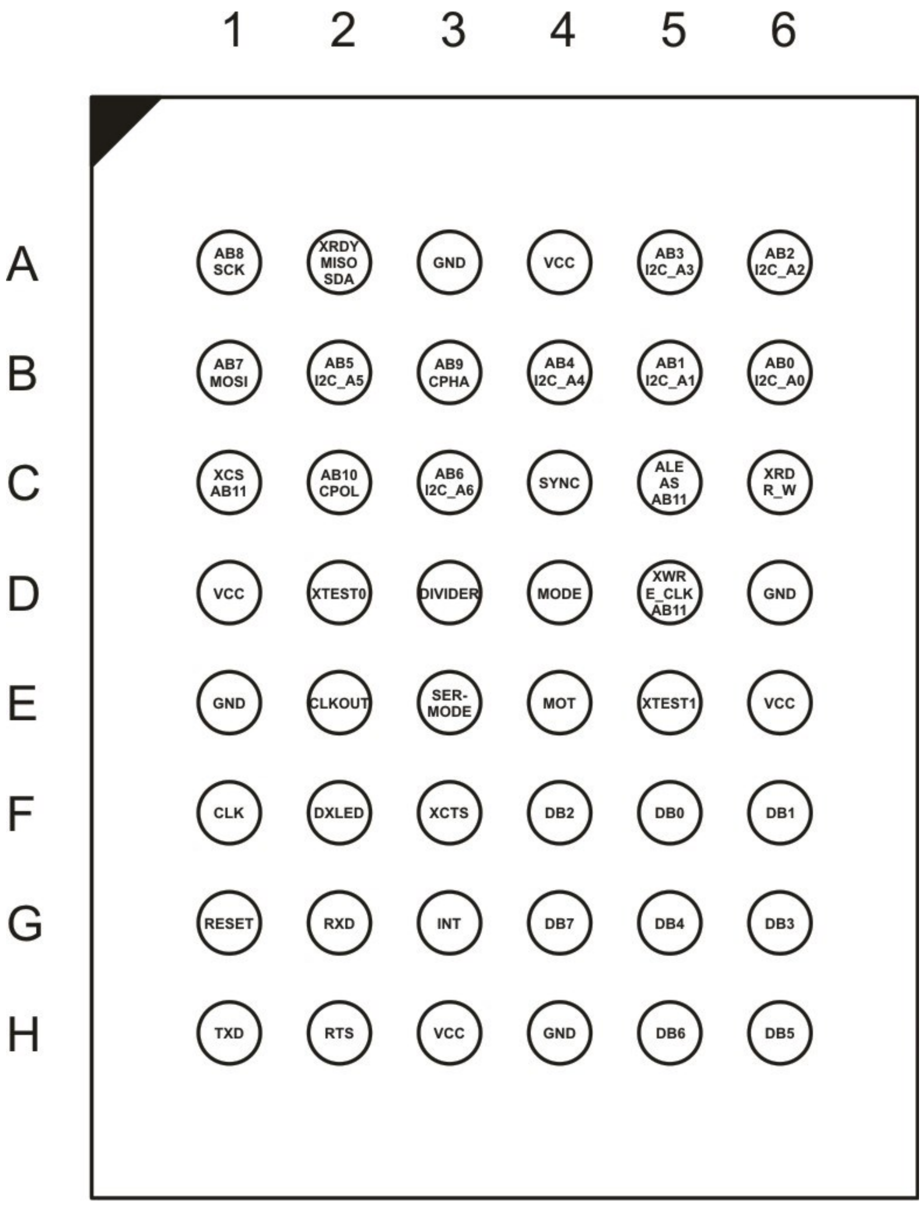



**Figure 3-1: VPC3+S LFBGA48 Pinout (TOP VIEW)**

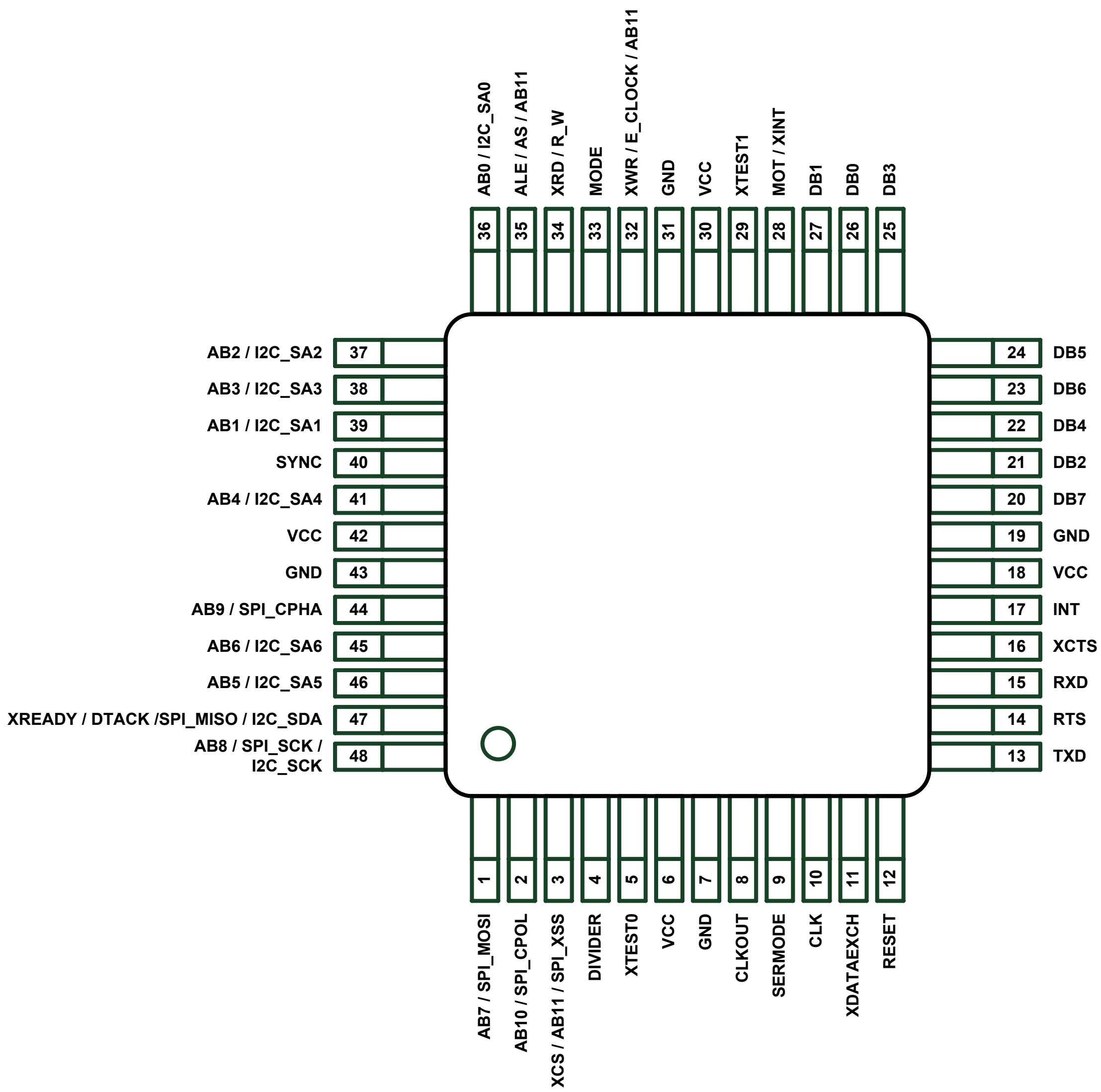



**Figure 3-2: VPC3+S LQFP48 Pinout (TOP VIEW)**

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

## 3.2 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(PD) | 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<br>'1': 24 MHz | Configuration Pin |
| D4 | 33 | MODE | I | '0': Asynchronous Mode (Parallel Interface Mode)<br>'1': Synchronous Mode (Parallel Interface Mode)<br>'0': SPI (Serial Interface Mode)<br>'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<br>'1': Serial Interface (SPI or I2C) | Configuration Pin |
| E4 | 28 | MOT/XINT | I | '0': Parallel Interface Intel Format<br>'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 Characteristics:**

I : LVTTL
(S) : Input with Schmitt-Trigger characteristic
(PD) : Internal Pull Down resistor (75 kΩ)

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.

### 3.2.1 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(PD) | 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**

### 3.2.2 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]<br>multiplexed with lower address bits [7:0]<br><br>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(PD) | In Synchronous Intel Mode these inputs are used to generate the internal Chip-Select signal.<br><br>Chip-Select is active if all inputs are '0'. | Use one (inverted) CPU Address Line for generating the VPC3+S Chip-Select signal.<br><br>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<br>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**

### 3.2.3 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(PD) | 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**

### 3.2.4 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.

<table>
<tr><th>Ball BGA</th><th>Pin QFP</th><th>Signal Name</th><th>In/Out</th><th>Description</th><th>Connect to</th></tr>
<tr><td>E3</td><td>9</td><td>SERMODE</td><td>I</td><td>‘0’: Parallel Interface</td><td>GND</td></tr>
<tr><td>E4</td><td>28</td><td>MOT/XINT</td><td>I</td><td>‘1’: Motorola Format</td><td>VCC</td></tr>
<tr><td>D4</td><td>33</td><td>MODE</td><td>I</td><td>‘1’: Synchronous Interface Mode</td><td>VCC</td></tr>
<tr><td>C5</td><td>35</td><td>AB11</td><td>I</td><td>Address Lines Bit 11</td><td>CPU Address Bus 11</td></tr>
<tr><td>C2</td><td>2</td><td>AB10</td><td>I(PD)</td><td rowspan="11">Address Lines Bits [10:0]</td><td rowspan="11">CPU<br>Address Bus [10:0]</td></tr>
<tr><td>B3</td><td>44</td><td>AB9</td><td>I</td></tr>
<tr><td>A1</td><td>48</td><td>AB8</td><td>I(S)</td></tr>
<tr><td>B1</td><td>1</td><td>AB7</td><td>I(S)</td></tr>
<tr><td>C3</td><td>45</td><td>AB6</td><td>I</td></tr>
<tr><td>B2</td><td>46</td><td>AB5</td><td>I</td></tr>
<tr><td>B4</td><td>41</td><td>AB4</td><td>I</td></tr>
<tr><td>A5</td><td>38</td><td>AB3</td><td>I</td></tr>
<tr><td>A6</td><td>37</td><td>AB2</td><td>I</td></tr>
<tr><td>B5</td><td>39</td><td>AB1</td><td>I</td></tr>
<tr><td>B6</td><td>36</td><td>AB0</td><td>I</td></tr>
<tr><td>G4</td><td>20</td><td>DB7</td><td>IO</td><td rowspan="8">Data Bus [7:0]</td><td rowspan="8">CPU Data Bus [7:0]</td></tr>
<tr><td>H5</td><td>23</td><td>DB6</td><td>IO</td></tr>
<tr><td>H6</td><td>24</td><td>DB5</td><td>IO</td></tr>
<tr><td>G5</td><td>22</td><td>DB4</td><td>IO</td></tr>
<tr><td>G6</td><td>25</td><td>DB3</td><td>IO</td></tr>
<tr><td>F4</td><td>21</td><td>DB2</td><td>IO</td></tr>
<tr><td>F6</td><td>27</td><td>DB1</td><td>IO</td></tr>
<tr><td>F5</td><td>26</td><td>DB0</td><td>IO</td></tr>
<tr><td>C1</td><td>3</td><td>XCS</td><td>I</td><td>Chip-Select Signal (active low)</td><td>CPU Chip-Select</td></tr>
<tr><td>D5</td><td>32</td><td>E_CLOCK</td><td>I</td><td>E-Clock</td><td>CPU E-Clock</td></tr>
<tr><td>C6</td><td>34</td><td>R_W</td><td>I</td><td>Read-Write Signal (‘1’ = Read)</td><td>CPU Read-Write</td></tr>
</table>

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

### 3.2.5 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(PD) | 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**

### 3.2.6 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**

# 4 Memory Organization

## 4.1 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 parameters. In addition, general parameter data (Station_Address, Ident_Number, etc.) and status information (Global_Control command, etc.) are also stored in these cells.

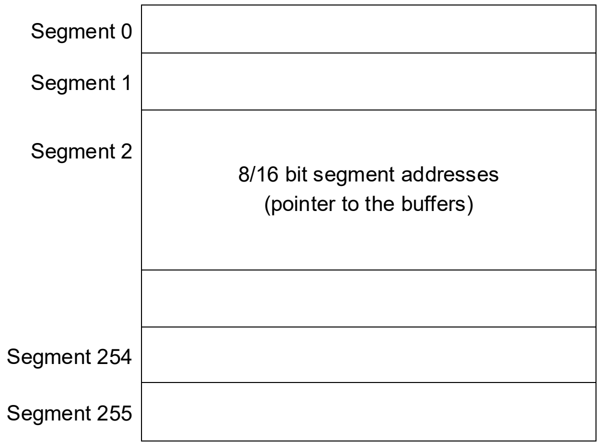
Corresponding to the parameter setting of the Organizational Parameters, the user-generated buffers are located beginning with address 40H. All buffers or lists must begin at segment addresses (8 bytes segmentation for 2K Byte mode, 16 bytes segmentation for 4K Byte mode).

| Address | Function | | |
|---|---|---|---|
| 000H<br>:<br>015H | Control Parameters<br>(latches/registers) (21 bytes) | | Internal working cells |
| 016H<br>:<br>03FH | Organizational Parameters (42 bytes) | | |
| 040H<br>:<br>:<br>7FFH (FFFH) | DP-buffers:<br><br><br><br><br><br><br>DP-V1-buffer:<br><br>DP-V2-buffer: | Data in (3) [1]<br>Data out (3) [2]<br>Diagnosis data(2)<br>Parameter data (1)<br>Configuration data (2)<br>Auxiliary buffers (2)<br>SSA-buffer (1)<br>SAP-List (1)<br>Indication / Response buffers [3]<br>DXB out (3) [4]<br>DXB-buffers (2)<br>CS-buffer (1)<br>PLL-buffer (1) | |

**Figure 4-1: Memory Table**

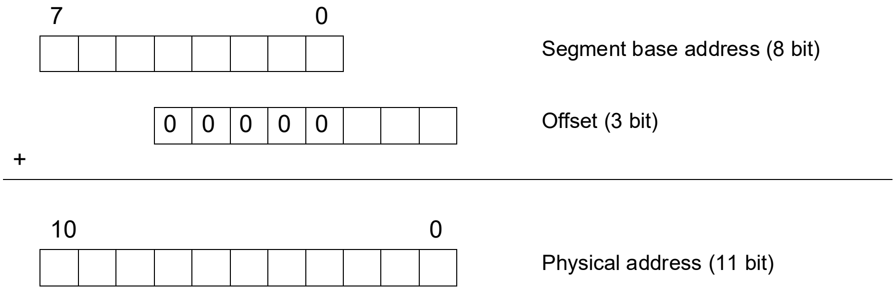
[1] Data in means input data from DP-Slave to DP-Master
[2] Data out means output data from DP-Master to DP-Slave
[3] Number of buffers depends on the entries in the SAP-List
[4] DXB out means input data from another DP-Slave (slave-to-slave communication)

**Internal VPC3+S RAM (2K/4K Byte)**

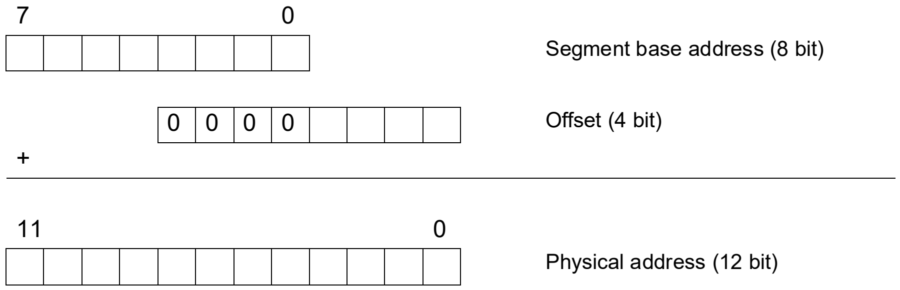



Building of the physical buffer address:

2K Byte Mode:




4K Byte Mode:



## 4.2 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 | Address 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 | Address 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 | | $minT_{SDR}$_Val | 7..0 | $minT_{SDR}$ 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**

# 4.3 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 | Address 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<br>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**

# 5 ASIC Interface

## 5.1 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).

### 5.1.1 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 0/2/3, 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<br>7 .. 0<br><br>See below for coding |

| Address | Bit Position | | | | | | | | Designation |
|---|---|---|---|---|---|---|---|---|---|
| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | |
| 07H (Intel) | Reserved | PrmCmd_ Supported | Spec_Clear_ Mode [1] | Spec_Prm_ Buf_Mode [2] | Set_Ext_Prm _Supported | User_Time_ Base | EOI_Time_ Base | DP_Mode | Mode Reg 0<br>15 .. 8<br><br>See below for coding |

[1] 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.

[2] 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<br>rw-0 | **Freeze_Supported**: Freeze_Mode support<br>0 = Freeze_Mode is not supported.<br>1 = Freeze_Mode is supported |
| bit 6<br>rw-0 | **Sync_Supported:** Sync_Mode support<br>0 = Sync_Mode is not supported.<br>1 = Sync_Mode is supported. |
| bit 5<br>rw-0 | **Early_Rdy**: Early Ready<br>0 = Normal Ready: Ready is generated when data is valid (write) or when data has been accepted (read).<br>1 = Ready is generated one clock pulse earlier |
| bit 4<br>rw-0 | **INT_Pol:** Interrupt Polarity<br>0 = The interrupt output is low-active.<br>1 = The interrupt output is high-active. |
| bit 3<br>rw-0 | **CS_Supported:** Enable Clock Synchronization<br>0 = Clock Synchronization is disabled (default)<br>1 = Clock Synchronization is enabled |
| bit 2<br>rw-0 | **WD_Base:** Watchdog Time Base<br>0 = Watchdog time base is 10 ms (default state)<br>1 = Watchdog time base is 1 ms |
| bit 1<br>rw-0 | **Dis_Stop_Control:** Disable Stopbit Control<br>0 = Stop bit monitoring is enabled.<br>1 = Stop bit monitoring is switched off<br>Set_Prm telegram overwrites this memory cell in the DP_Mode. (Refer to the user specific data.) |
| bit 0<br>rw-0 | **Dis_Start_Control:** Disable Startbit Control<br>0 = Monitoring the following start bit is enabled.<br>1 = Monitoring the following start bit is switched off<br>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<br>rw-0 | **Reserved** |
| bit 14<br>rw-0 | **PrmCmd_Supported**: PrmCmd support for redundancy<br>0 = PrmCmd is not supported.<br>1 = PrmCmd is supported |
| bit 13<br>rw-0 | **Spec_Clear_Mode:** Special Clear Mode (Fail Safe Mode)<br>0 = No special clear mode.<br>1 = Special clear mode. VPC3+S will accept data telegrams with data unit = 0 |
| bit 12<br>rw-0 | **Spec_Prm_Buf_Mode:** Special-Parameter-Buffer Mode<br>0 = No Special-Parameter-Buffer.<br>1 = Special-Parameter-Buffer mode. Parameterization data will be stored directly in the Special-Parameter-Buffer. |
| bit 11<br>rw-0 | **Set_Ext_Prm_Supported:** Set_Ext_Prm telegram support<br>0 = SAP 53 is deactivated<br>1 = SAP 53 is activated |
| bit 10<br>rw-0 | **User_Time_Base:** Timebase of the cyclical User_Time_Clock-Interrupt<br>0 = The User_Time_Clock-Interrupt occurs every 1 ms.<br>1 = The User_Time_Clock-Interrupt occurs every 10 ms. |
| bit 9<br>rw-0 | **EOI_Time_Base:** End-of-Interrupt Timebase<br>0 = The interrupt inactive time is at least 1 µs long.<br>1 = The interrupt inactive time is at least 1 ms long |
| bit 8<br>rw-0 | **DP_Mode:** DP_Mode enable<br>0 = DP_Mode is disabled.<br>1 = DP_Mode is enabled. VPC3+S sets up all DP_SAPs (default configuration!) |

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

### 5.1.2 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<br><br>See below for coding |

| | Mode Register 1, Set, Address 08H: |
|---|---|
| bit 7<br>rw-0 | **Reserved** |
| bit 6<br>rw-0 | **Reserved** |
| bit 5<br>rw-0 | **Res_User_WD:** Resetting the User_WD_Timer<br>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<br>rw-0 | **En_Change_Cfg_Buffer:** Enabling buffer exchange (Config-Buffer for Read_Config-Buffer)<br>0 = With User_Cfg_Data_Okay_Cmd, the Config-Buffer may not be exchanged for the Read_Config-Buffer.<br>1 = With User_Cfg_Data_Okay_Cmd, the Config-Buffer must be exchanged for the Read_Config-Buffer. |
| bit 3<br>rw-0 | **User_LEAVE-MASTER.** Request to the DP_SM to go to WAIT-PRM.<br>1 = The user causes the DP_SM to go to WAIT-PRM.<br>After this action, VPC3+ sets User_LEAVE-MASTER to '0' again. |
| bit 2<br>rw-0 | **Go_Offline:** Going into the Offline state<br>1 = After the current request ends, VPC3+S goes to the Offline state and sets Go_Offline to '0' again. |
| bit 1<br>rw-0 | **EOI:** End-of-Interrupt<br>1 = VPC3+S disables the interrupt output and sets EOI to '0' again. |
| bit 0<br>rw-0 | **Start_VPC3:** Exiting the Offline state<br>1 = VPC3+S exits offline and goes to Passive_Idle<br>In addition the Idle Timer and Watchdog Timer are started and 'Go_Offline = 0' is set |

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

### 5.1.3 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<br>7 .. 0 |

| | Mode Register 2, Address 0CH: |
|---|---|
| bit 7<br>w-0 | **4KB_Mode:** size of internal RAM<br>0 = 2K Byte RAM (default).<br>1 = 4K Byte RAM |
| bit 6<br>w-0 | **No_Check_Prm_Reserved:** disables checking of the reserved bits in DPV1_Status_2/3 of Set_Prm telegram<br>0 = reserved bits of a Set_Prm telegram are checked (default).<br>1 = reserved bits of a Set_Prm telegram are not checked. |
| bit 5<br>w-0 | **SYNC_Pol:** polarity of SYNC pulse (for Isochronous Mode only)<br>0 = negative polarity of SYNC pulse (default)<br>1 = positive polarity of SYNC pulse |
| bit 4<br>w-0 | **SYNC_Ena:** enables generation of SYNC pulse (for Isochronous Mode only)<br>0 = SYNC pulse generation is disabled (default)<br>1 = SYNC pulse generation is enabled |
| bit 3<br>w-0 | **DX_Int_Port:** Port mode for DX_Out interrupt (ignored if SYNC_Ena set)<br>0 = DX_Out interrupt is not assigned to port DATAEXCH (default).<br>1 = DX_Out Interrupt (synchronized to SYNCH telegram) is assigned to port DATAEXCH. |
| bit 2<br>w-0 | **DX_Int_Mode:** Mode of DX_out interrupt<br>0 = DX_Out interrupt is only generated, if Len_Dout_Buf is unequal 0 (default).<br>1 = DX_Out interrupt is generated after every Data_Exchange telegram |
| bit 1<br>w-0 | **No_Check_GC_Reserved:** Disables checking of the reserved bits in Global_Control telegram<br>0 = reserved bits of a Global_Control telegram are checked (default).<br>1 = reserved bits of a Global_Control telegram are not checked. |
| bit 0<br>w-1 | **GC_Int_Mode:** Controls generation of New_GC_Command interrupt<br>0 = New_GC_Command interrupt is only generated, if a changed Global_Control telegram is received<br>1 = New_GC_Command interrupt is generated after every Global_Control telegram (default) |

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

### 5.1.4 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<br>7 .. 0 |

| Mode Register 3, Address 12H: | |
|---|---|
| bit 7<br>w-0 | **Reserved** |
| bit 6<br>w-0 | **Reserved** |
| bit 5<br>w-0 | **Reserved** |
| bit 4<br>w-0 | **Reserved** |
| bit 3<br>w-0 | **PLL_Supported:** Enables IsoM-PLL<br>0 = PLL is disabled<br>1 = PLL is enabled; For use of PLL, SYNC_Ena must be set. |
| bit 2<br>w-0 | **En_Chk_SSAP:** Evaluation of Source Address Extension<br>0 = VPC3+ accept any value of S_SAP<br>1 = VPC3+ only process the received telegram if the S_SAP match to the default values presented by the IEC 61158 |
| bit 1<br>w-0 | **DX_Int_Mode_2:** Mode of DX_out interrupt<br>0 = DX_Out interrupt is generated after each Data_Exch telegram<br>1 = DX_Out interrupt is only generated, if received data is not equal to current data in DX_Out buffer of user |
| bit 0<br>w-0 | **GC_Int_Mode_Ext:** extend GC_Int_Mode, works only if GC_Int_Mode=0<br>0 = GC Interrupt is only generated, if changed GC telegram is received<br>1 = GC Interrupt is only generated, if GC telegram with changed Control_Command is received |

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

## 5.2 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 (Intel) | WD_State | | DP_State | | Reserved | Diag_Flag | Reserved | Offline/ Passive_Idle | Status-Reg 7..0<br>See below for coding |
| | 1 | 0 | 1 | 0 | | | | | |

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

| | **Status Register,Low-Byte, Address 04H (Intel):** |
|---|---|
| bit 7,6<br>r-00 | **WD_State 1..0:** State of the Watchdog State Machine<br>00 = BAUD_SEARCH state<br>01 = BAUD_CONTROL state<br>10 = DP_CONTROL state<br>11 = Not possible |
| bit 5,4<br>r-00 | **DP_State 1..0:** State of the DP State Machine<br>00 = WAIT-PRM state<br>01 = WAIT-CFG state<br>10 = DATA-EXCH state<br>11 = Not possible |
| bit 3<br>r-0 | **Reserved** |
| bit 2<br>r-0 | **Diag_Flag:** Status of the Diagnosis-Buffer<br>0 = The Diagnosis-Buffer had been fetched by the DP-Master.<br>1 = The Diagnosis-Buffer had not been fetched by the DP-Master yet. |
| bit 1<br>r-0 | **Reserved** |
| bit 0<br>r-0 | **Offline/Passive-Idle:** Offline-/Passive_Idle state<br>0 = VPC3+S is in Offline.<br>1 = VPC3+S is in Passive_Idle. |

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

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

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

# 5.3 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 Acknowledge Register (IAR).

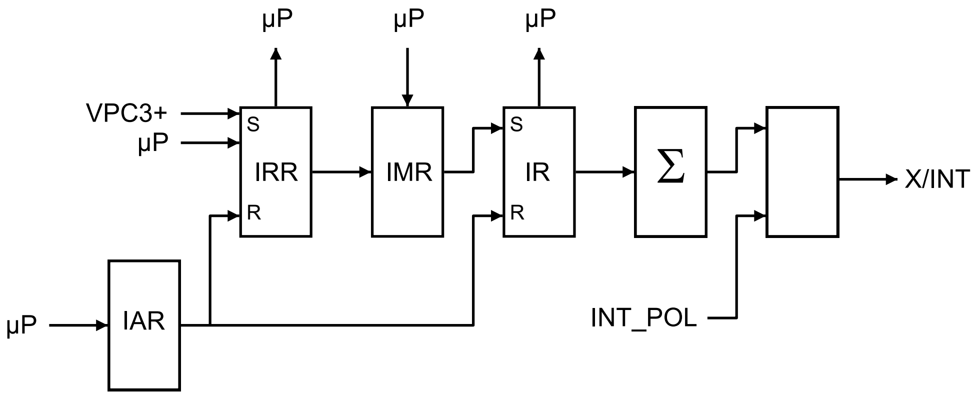



**Figure 5-8: Block Diagram of Interrupt Controller**

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.

### 5.3.1 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<br>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<br>See below for coding |

| | Interrupt-Request-Register, Low-Byte, Address 00H (Intel): |
|---|---|
| bit 7<br>rw-0 | **DXB_Out:**<br>VPC3+S has received a DXB telegram and made the new output data available in the 'N' buffer. |
| bit 6<br>rw-0 | **New_Ext_Prm_Data:**<br>The VPC3+S has received a Set_Ext_Prm telegram and made the data available in the Parameter-Buffer. |
| bit 5<br>rw-0 | **DXB_Link_Error:**<br>The Watchdog cycle is elapsed and at least one Publisher-Subscriber connection breaks down. |
| bit 4<br>rw-0 | **User_Timer_Clock:**<br>The time base for the User_Timer_Clocks is run out (1 / 10ms). |
| bit 3<br>rw-0 | **WD_DP_CONTROL_Timeout:**<br>The watchdog timer expired in the DP_CONTROL state. |
| bit 2<br>rw-0 | **Baud_Rate_Detect:**<br>The VPC3+S has left the BAUD_SEARCH state and found a baud rate. |
| bit 1<br>rw-0 | **Go/Leave_DATA-EXCH:**<br>The DP_SM has entered or exited the DATA-EXCH state. |
| bit 0<br>rw-0 | **MAC_Reset (used if CS_Supported=0)**:<br>After processing the current request, the VPC3+D has entered the Offline state (by setting the Go_Offline bit).<br>**Clock_Sync (used if CS_Supported=1)**:<br>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<br>rw-0 | **FDL_Ind:**<br>The VPC3+S has received an acyclic service request and made the data available in an Indication-Buffer. |
| bit 14<br>rw-0 | **Poll_End_Ind:**<br>The VPC3+S have send the response to an acyclic service. |
| bit 13<br>rw-0 | **DX_Out:**<br>The VPC3+S have received a Data_Exchange telegram and made the new output data available in the ‘N’ buffer. |
| bit 12<br>rw-0 | **Diag_Buffer_Changed:**<br>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<br>rw-0 | **New_Prm_Data:**<br>The VPC3+S have received a Set_Prm telegram and made the data available in the Parameter-Buffer. |
| bit 10<br>rw-0 | **New_Cfg_Data:**<br>The VPC3+S have received a Chk_Cfg telegram and made the data available in the Config-Buffer. |
| bit 9<br>rw-0 | **New_SSA_Data:**<br>The VPC3+S have received a Set_Slave_Add telegram and made the data available in the Set_Slave_Add-Buffer. |
| bit 8<br>rw-0 | **New_GC_Command:**<br>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**

### 5.3.2 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<br>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<br>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.).**

## 5.4 Watchdog Timer

The VPC3+S is able to identify the baud rate automatically. The state machine is in the BAUD_SEARCH state after each RESET and also after the Watchdog (WD) Timer has expired in the BAUD_CONTROL state.

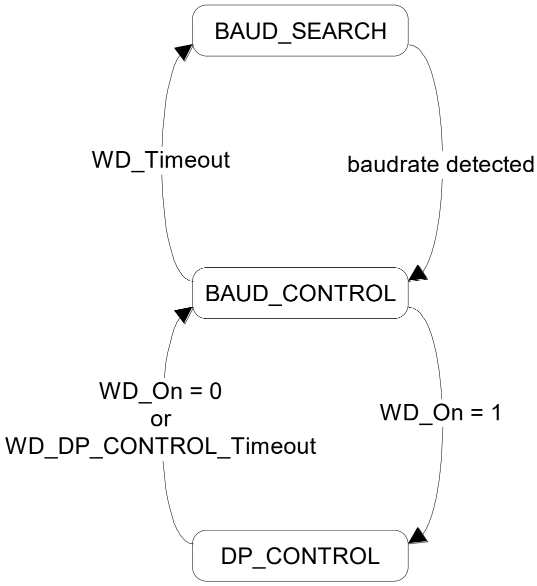



**Figure 5-12: Watchdog State Machine (WD_SM)**

### 5.4.1 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.

### 5.4.2 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.

### 5.4.3 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):

$T_{WD}$ = 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).

# 6 PROFIBUS DP Interface

## 6.1 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 configuration 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.

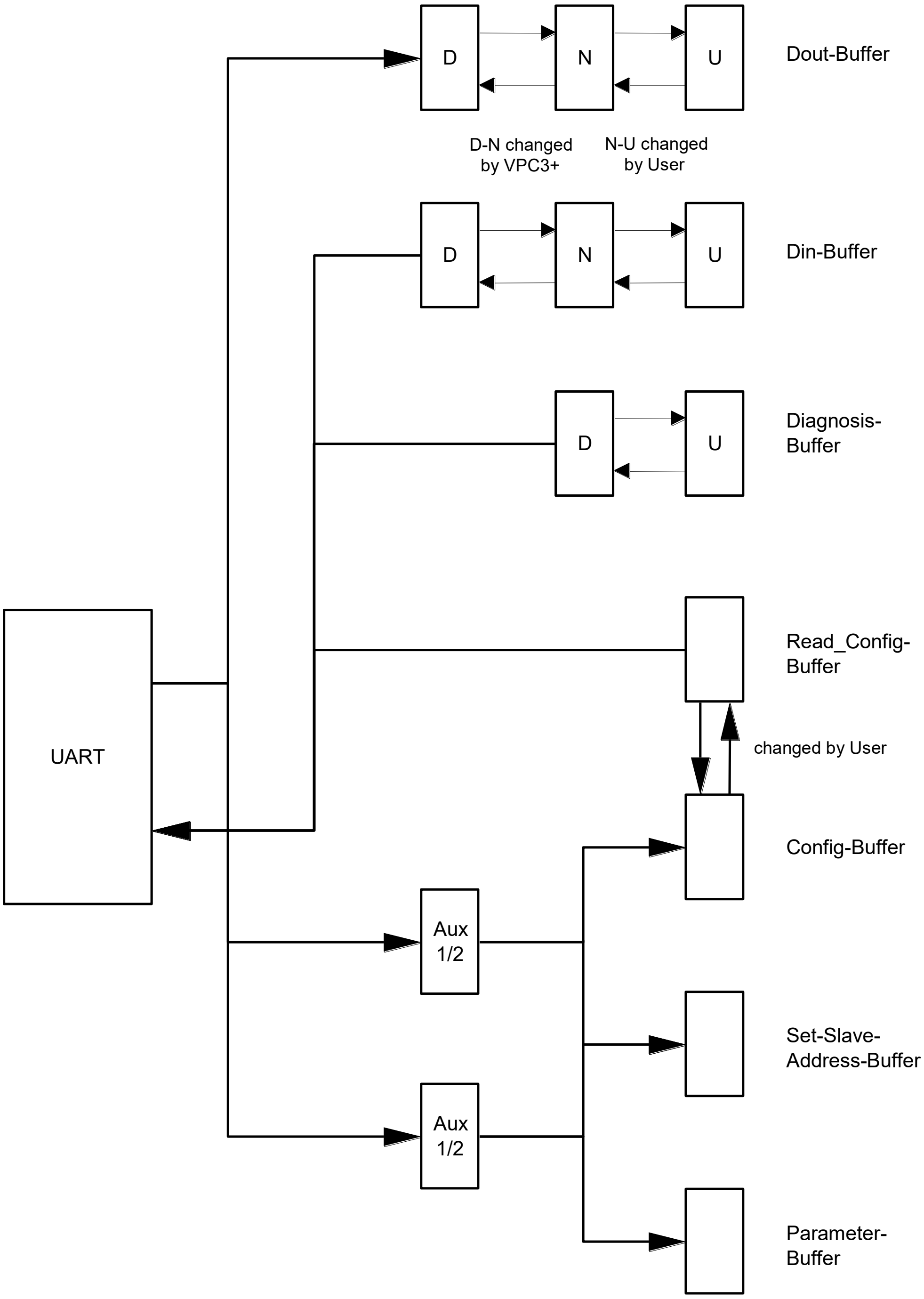



**Figure 6-1: DP_SAP Buffer Structure**

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) 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<br><br>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<br>0 = Aux-Buffer 1<br>1 = Aux-Buffer 2 |
| bit 1 | **Chk_Cfg:** Check Configuration<br>0 = Aux-Buffer 1<br>1 = Aux-Buffer 2 |
| bit 0 | **Set_Prm:** Set (Extended) Parameter<br>0 = Aux-Buffer 1<br>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.

## 6.2 Description of the DP Services

### 6.2.1 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<br>:<br>243 | | | | | | | | | Rem_Slave_Data<br>additional application specific data |

**Figure 6-4: Structure of the Set_Slave_Add Telegram**

### 6.2.2 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, "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 | | | | | | | | | $minT_{SDR}$ |
| 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:**<br>0 = DP-V1 extensions disabled (default)<br>1 = DP-V1 extensions enabled |
| bit 6 | **Fail_Safe:**<br>0 = Fail Safe mode disabled (default)<br>1 = Fail Safe mode enabled |
| bit 5 | **Publisher_Enable:**<br>0 = Publisher function disabled (default)<br>1 = Publisher function enabled |
| bit 4-3 | **Reserved:** To be parameterized with ‘0’ |
| bit 2 | **WD_Base:** Watchdog Time Base<br>0 = Watchdog time base is 10 ms (default)<br>1 = Watchdog time base is 1 ms |
| bit 1 | **Dis_Stop_Control:** Disable Stop bit Control<br>0 = Stop bit monitoring in the receiver is enabled (default)<br>1 = Stop bit monitoring in the receiver is disabled |
| bit 0 | **Dis_Start_Control:** Disable Start bit Control<br>0 = Start bit monitoring in the receiver is enabled (default)<br>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.

### 6.2.3 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_<br>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_<br>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**

### 6.2.4 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<br>00 = Nil<br>01 = User<br>10 = VPC3+<br>11 = VPC3_Send_Mode |
| bit 1-0 | **Diag_Buf1:** Assignment of Diagnosis Buffer 1<br>00 = Nil<br>01 = User<br>10 = VPC3+<br>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_<br>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_<br>Overflow | Stat_Diag | Ext_Diag | |
| 1 | | | | | | | | | |
| 2 | | | | | | | | | |
| 3 | | | | | | | | | |
| 4 | | | | | | | | | |
| 5 | | | | | | | | | |
| 6<br>:<br>n | user input | | | | | | | | Ext_Diag_Data<br>(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).

### 6.2.5 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<br>00 = Nil<br>01 = Dout_Buf_Ptr1<br>10 = Dout_Buf_Ptr2<br>11 = Dout_Buf_Ptr3 |
| bit 5-4 | **U:** Assignment of the U-Buffer<br>00 = Nil<br>01 = Dout_Buf_Ptr1<br>10 = Dout_Buf_Ptr2<br>11 = Dout_Buf_Ptr3 |
| bit 3-2 | **N:** Assignment of the N-Buffer<br>00 = Nil<br>01 = Dout_Buf_Ptr1<br>10 = Dout_Buf_Ptr2<br>11 = Dout_Buf_Ptr3 |
| bit 1-0 | **D:** Assignment of the D-Buffer<br>00 = Nil<br>01 = Dout_Buf_Ptr1<br>10 = Dout_Buf_Ptr2<br>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<br>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<br>0 = U buffer contains data<br>1 = U buffer is cleared |
| bit 2 | **State_U_Buffer:** State of the User-Buffer<br>0 = no new U buffer<br>1 = new U buffer |
| bit 1-0 | **Ind_U_Buffer:** Indicated User-Buffer<br>01 = Dout_Buf_Ptr1<br>10 = Dout_Buf_Ptr2<br>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<br>00 = Nil<br>01 = Din_Buf_Ptr1<br>10 = Din_Buf_Ptr2<br>11 = Din_Buf_Ptr3 |
| bit 5-4 | **U:** Assignment of the U-Buffer<br>00 = Nil<br>01 = Din_Buf_Ptr1<br>10 = Din_Buf_Ptr2<br>11 = Din_Buf_Ptr3 |
| bit 3-2 | **N:** Assignment of the N-Buffer<br>00 = Nil<br>01 = Din_Buf_Ptr1<br>10 = Din_Buf_Ptr2<br>11 = Din_Buf_Ptr3 |
| bit 1-0 | **D:** Assignment of the D-Buffer<br>00 = Nil<br>01 = Din_Buf_Ptr1<br>10 = Din_Buf_Ptr2<br>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.

### 6.2.6 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<br><br>See below for coding |

| | **R_GC_Command, Address 3CH:** |
|---|---|
| bit 7-6 | **Reserved** |
| bit 5 | **Sync:**<br>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:**<br>The Unsync command cancels the Sync command. |
| bit 3 | **Freeze:**<br>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:**<br>The Unfreeze command cancels the Freeze command. |
| bit 1 | **Clear_Data:**<br>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**

### 6.2.7 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.

### 6.2.8 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.

### 6.2.9 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.

# 7 PROFIBUS DP Extensions

## 7.1 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 : 243 | | | | | | | | | User_Prm_Data |

**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<br>:<br>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.**

# 7.2 PROFIBUS DP-V1

## 7.2.1 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<br>0 = no Response sent<br>1 = Response sent<br>**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.<br>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<br>USER buffer assigned to user<br>IND indication data included in buffer<br>RESP response data included in buffer<br>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)**

### 7.2.2 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.

# 7.3 PROFIBUS DP-V2

## 7.3.1 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.

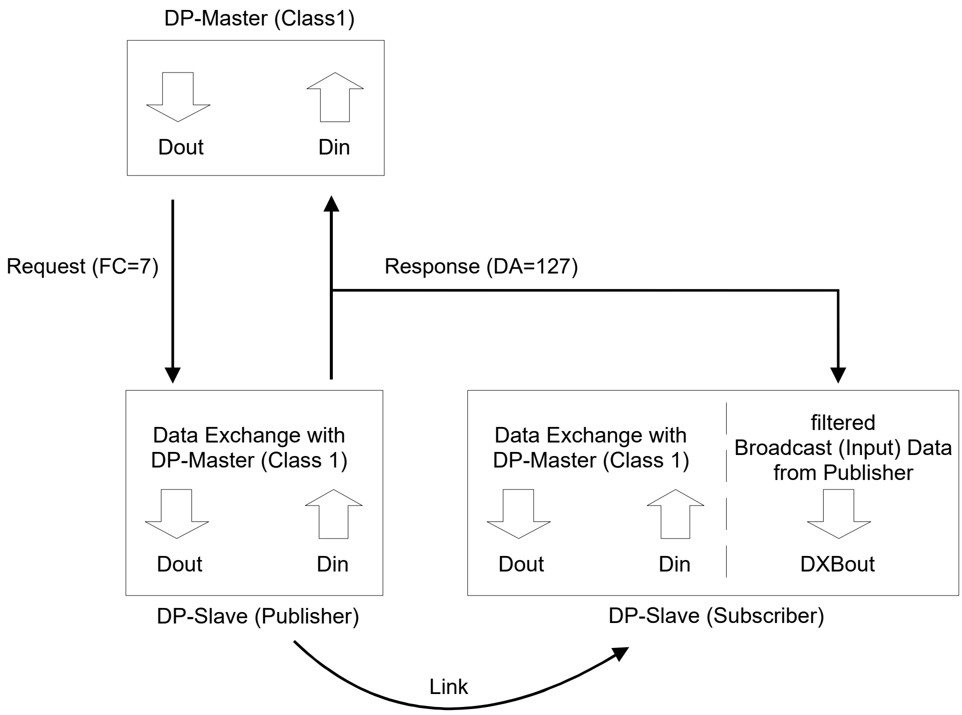



**Figure 7-7 : Overview DXB**

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 $minT_{SDR}$ must be set to '$T_{ID1} = 37\ t_{bit} + 2\ T_{SET} + T_{QUI}$'.

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 watchdog 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<br>:<br>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<br>:<br>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<br>:<br>61 | | | | | | | | | further link entries |

| | **Link_Status:** |
|---|---|
| bit 7 | **Link_Status :**<br>1 = active, valid data receipt during last monitoring period<br>0 = not active, no valid data receipt during last monitoring period (DEFAULT) |
| bit 6 | **Link_Error:**<br>0 = no faulty Broadcast data receipt (DEFAULT)<br>1 = wrong length, error occurred during reception |
| bit 0 | **Data_Exist:**<br>0 = no correct Broadcast data receipt during current monitoring period (DEFAULT)<br>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<br>:<br>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<br>00 = Nil<br>01 = DXBout_Buf_Ptr1<br>10 = DXBout_Buf_Ptr2<br>11 = DXBout_Buf_Ptr3 |
| bit 5-4 | **U:** Assignment of the U-Buffer<br>00 = Nil<br>01 = DXBout_Buf_Ptr1<br>10 = DXBout_Buf_Ptr2<br>11 = DXBout_Buf_Ptr3 |
| bit 3-2 | **N:** Assignment of the N-Buffer<br>00 = Nil<br>01 = DXBout_Buf_Ptr1<br>10 = DXBout_Buf_Ptr2<br>11 = DXBout_Buf_Ptr3 |
| bit 1-0 | **D:** Assignment of the D-Buffer<br>00 = Nil<br>01 = DXBout_Buf_Ptr1<br>10 = DXBout_Buf_Ptr2<br>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<br><br>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<br>0 = no new U buffer<br>1 = new U buffer |
| bit 1-0 | **Ind_U_Buffer:** Indicated User-Buffer<br>01 = DXBout_Buf_Ptr1<br>10 = DXBout_Buf_Ptr2<br>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.**

### 7.3.2 IsoM (Isochronous Mode)

The IsoM synchronizes DP-Master, DP-Slave and DP-Cycle. The isochronous 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 synchronization signal at Pin C4 (SYNC) is generated by each reception of a SYNCH telegram. The SYNCH telegram is a special coded Global_Control request.

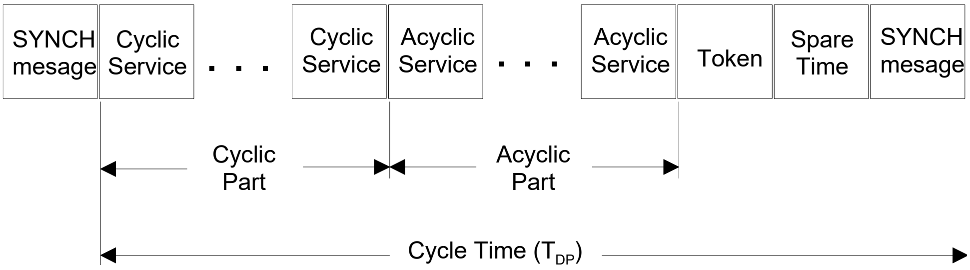



**Figure 7-15: Telegram sequences in IsoM with one DP-Master (Class 1)**

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 synchronization.

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.

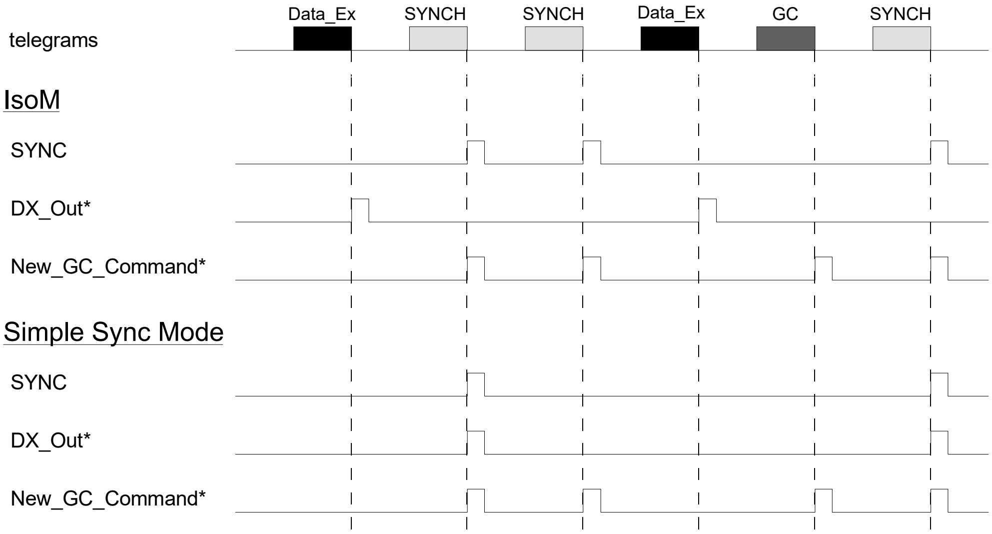



**Figure 7-17: SYNC-signal and interrupts for synchronization modes**

(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 µs 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 | | | | | | | | | $minT_{SDR}$ |
| 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<br>:<br>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 | | | | | | | | | $minT_{SDR}$ |
| 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.**

### 7.3.2.1 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.

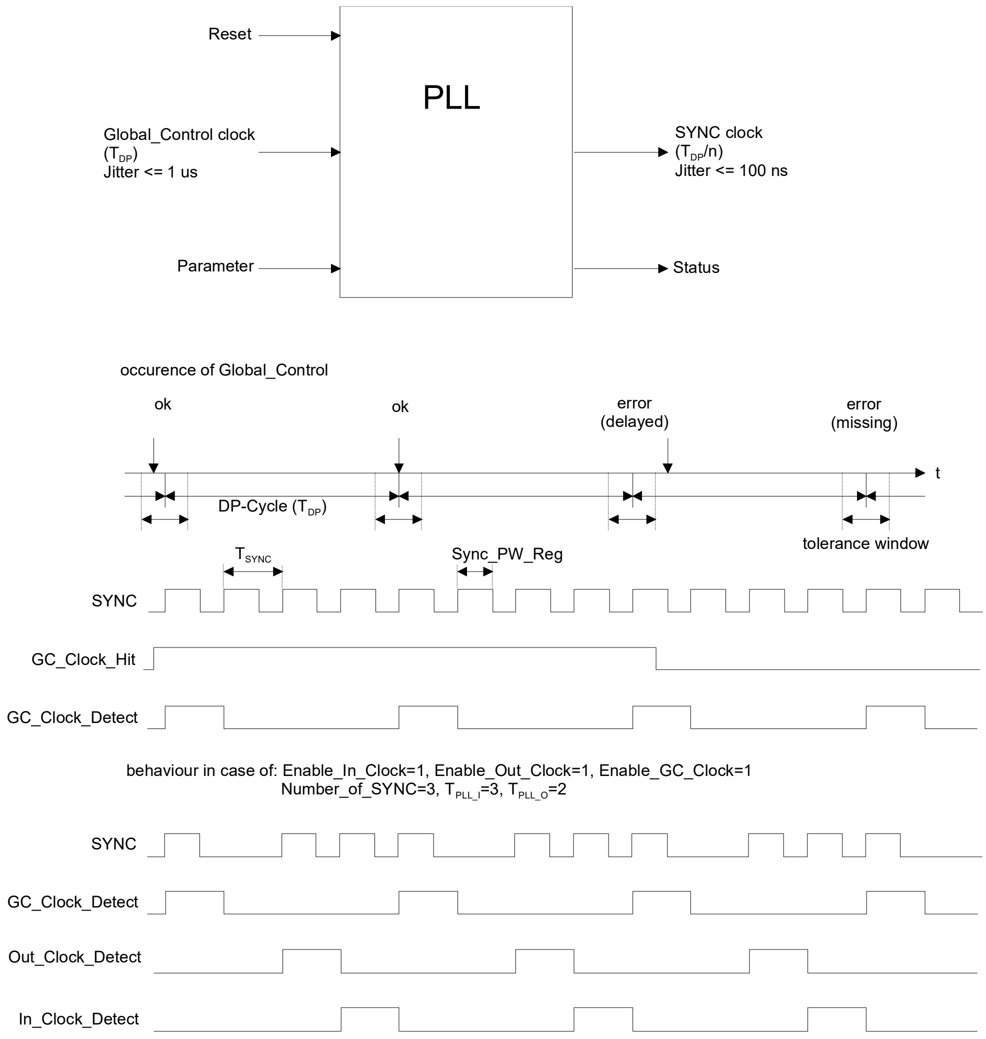



**Figure 7-20: SYNC clock and status signals of PLL**

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 ($T_{SYNC}$) | 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 $T_{DP}$ |
| | E_limit | number of acceptable synchronization errors |
| | input time ($T_{PLL\_I}$) | point in time for actual value acquisition |
| | output time ($T_{PLL\_O}$) | point in time for setpoint transfer |
| | PLL window ($T_{PLL\_W}$) | half the width of the tolerance window |
| | First_Window | start value of PLL window |
| | PLL delay time ($T_{PLL\_D}$) | 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 µs / 62,5 µs / 125 µs / 250 µs / 500 µs / 1000 µs) | | | | | | | | $T_{BASE\_DP}$: Time Base for $T_{DP}$ (Time Base 1/12 µs) |
| 9 : 10 | 1..($2^{16}$-1) Note: GSD-Spezifikation: $T_{DP\_MAX}$=32 ms | | | | | | | | $T_{DP}$: DP Cycle Time (Time Base $T_{BASE\_DP}$) |
| 11 | 1..14 | | | | | | | | $T_{MAPC}$: Master Application Cycle Time (Time Base $T_{DP}$) |
| 12 : 15 | 375 / 750 / 1500 / 3000 / 6000 / 12000 (= 31,25 µs / 62,5 µs / 125 µs / 250 µs / 500 µs / 1000 µs) | | | | | | | | $T_{BASE\_IO}$: Time Base of $T_I$ ,$T_O$ (Time Base 1/12 µs) |
| 16 : 17 | 0..($2^{16}$-1) | | | | | | | | $T_I$: Instant in Time of the Actual Value Acquistion (Time Base $T_{BASE\_IO}$) |
| 18 : 19 | 0..($2^{16}$-1) | | | | | | | | $T_O$: Instant in Time of the setpoint transfer (Time Base $T_{BASE\_IO}$) |
| 20 : 23 | 0..($2^{32}$-1) | | | | | | | | $T_{DX}$: Data_Exchange Time (Time Base 1/12 µs) |
| 24 : 25 | 1..($2^{16}$-1) (Default: 12) | | | | | | | | $T_{PLL\_W}$: PLL Window (Time Base 1/12 µs) |
| 26 : 27 | 0..($2^{16}$-1) (Default: 0) | | | | | | | | $T_{PLL\_D}$: PLL Delay Time (Time Base 1/12 µs) |

**Figure 7-22: Format of Structured_Prm_Data with IsoM Parameter**

The following input parameters have to be calculated by firmware:

- SYNC cycle time:

$$T_{SYNC} = \frac{T_{DP}}{n} = \frac{T_{DP}}{Number\_of\_SYNC + 1}$$

- start value of PLL window:

$$First\_Window \geq T_{PLL\_W} \cdot n1 + T_{DP} \cdot n2 \qquad \text{with } n1 > 1;\ n2 > 0{,}0003$$

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 : 3 | 1..($2^{16}$-1) (Default: 12) | | | | | | | | $T_{PLL\_W}$: PLL_Window (Time Base $\frac{1}{12}$ µs) |
| 4 : 5 | 0..($2^{16}$-1) (Default: 0) | | | | | | | | $T_{PLL\_D}$: PLL_Delay_Time (Time Base $\frac{1}{12}$ µs) |
| 6 : 9 | 1..($2^{32}$-1) | | | | | | | | $T_{SYNC}$: SYNC_Cycle_Time (Time Base $\frac{1}{48}$ µs) |
| 10 : 11 | reserved | | | | | | Number_of_SYNC(9:8) | | Number_of_SYNC |
| | Number_of_SYNC(7:0) | | | | | | | | |
| 12 : 15 | 1..($2^{32}$-1) | | | | | | | | First_Window (Time Base $\frac{1}{48}$ µs) |
| 16 : 17 | reserved | | | | | | $T_{PLL\_I}$ (9:8) | | $T_{PLL\_I}$: Input_Time (Time Base $T_{SYNC}$) |
| | $T_{PLL\_I}$ (7:0) | | | | | | | | |
| 18 : 19 | reserved | | | | | | $T_{PLL\_O}$ (9:8) | | $T_{PLL\_O}$: Output_Time (Time Base $T_{SYNC}$) |
| | $T_{PLL\_O}$ (7:0) | | | | | | | | |
| 20 | 0..255 | | | | | | | | E_limit |

| | **PLL Buffer** |
|---|---|
| GC_Clock_Hit<br>r-0 | **GC_clock_Hit**:<br>The VPC3+ has received a valid 'SYNCH telegram' during the tolerance window. |
| GC_Clock_ Detect<br>r-0 | **GC_Clock_Detect**:<br>Last SYNC signal coincides with the (expected) 'SYNCH telegram'. |
| GC_Clock_Errror<br>r-0 | **GC_Clock_Error**:<br>PLL detects Synchronization Errors and has to be resynchronized. |
| PLL_synched<br>r-0 | **PLL_synched**:<br>PLL is synchronized with the DP-Masters SYNCH. |
| Out_Clock_Detect<br>r-0 | **Out_Clock_Detect**:<br>Last SYNC signal coincides with the instant in time of the setpoint transfer. |
| In_Clock_Detect<br>r-0 | **In_Clock_Detect**:<br>Last SYNC signal coincides with the instant in time of the actual value acquisition. |
| PLL_Start<br>rw-0 | **PLL_Start**:<br>0 = PLL is stopped<br>1 = PLL is started |
| SYNC_Enable<br>rw-0 | **SYNC_Enable**:<br>0 = SYNC signal is not enabled<br>1 = SYNC signal is send to DATAEXCH_N |
| SYNC_Mode<br>rw-0 | **SYNC_Mode**:<br>0 = SYNC signal not synchronized to 'SYNCH telegram'<br>1 = SYNC signal synchronized to 'SYNCH telegram' |
| Enable_GC_Clock<br>rw-0 | **Enable_GC_Clock**:<br>0 = generate no SYNC signal coincides with the (expected) 'SYNCH telegram'<br>1 = generate SYNC signal coincides with the (expected) 'SYNCH telegram' |
| Enable_Out_Clock<br>rw-0 | **Enable_Out_Clock**:<br>0 = generate no SYNC signal at $T_O$<br>1 = generate SYNC signal at $T_O$ |
| Enable_In_Clock<br>rw-0 | **Enable_In_Clock**:<br>0 = generate no SYNC signal at $T_I$<br>1 = generate SYNC signal at $T_I$ |

| | **PLL Buffer** |
|---|---|
| Number_of_SYNC<br>rw-0 | **Number_of_SYNC**:<br>Number of SYNC cycles per DP cycle:<br>Number_of_SYNC + 1 |
| $T_{PLL\_I}$ rw-0 | **Input_Time**:<br>Number of SYNC cycles from start of DP cycle up to $T_I$ |
| $T_{PLL\_O}$ rw-0 | **Output_Time**:<br>Number of SYNC cycles from start of DP cycle up to $T_O$ |
| E_limit rw-0 | **E_limit**:<br>Number of acceptable synchronization errors during time interval. |

**Figure 7-23: Format of the PLL_Buffer**

**$T_I$ in the Structured_Prm_Data block is the period of time between actual value acquisition and the start of new DP cycle whereas $T_{PLL\_I}$ is the period of time from the start of DP cycle to the point of data acquisition.**

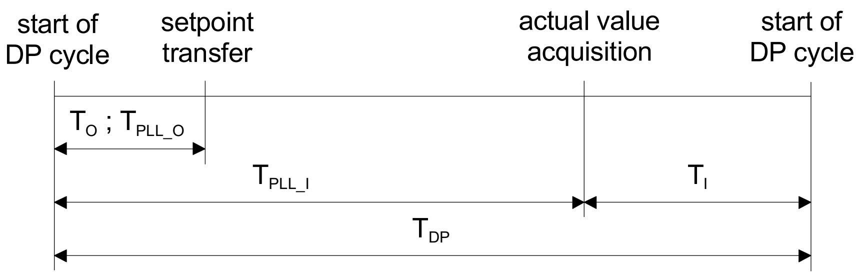



**Figure 7-24: configuration of $T_{PLL\_O}$ and $T_{PLL\_I}$**

If none of the Enable_xx_Clock bits is set the PLL generates a SYNC clock after every expiration of the slave application cycle (= $T_{SYNC}$).

| **VPC3+S** | **Firmware** |
|---|---|
| | configure DP-Slave for IsoM<br>set PLL_Support |
| receive Set_(Ext_)Prm<br>set New_(Ext_)Prm_Data interrupt | |
| | acknowledge New_(Ext_)Prm_Data interrupt<br>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)**

### 7.3.3 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.

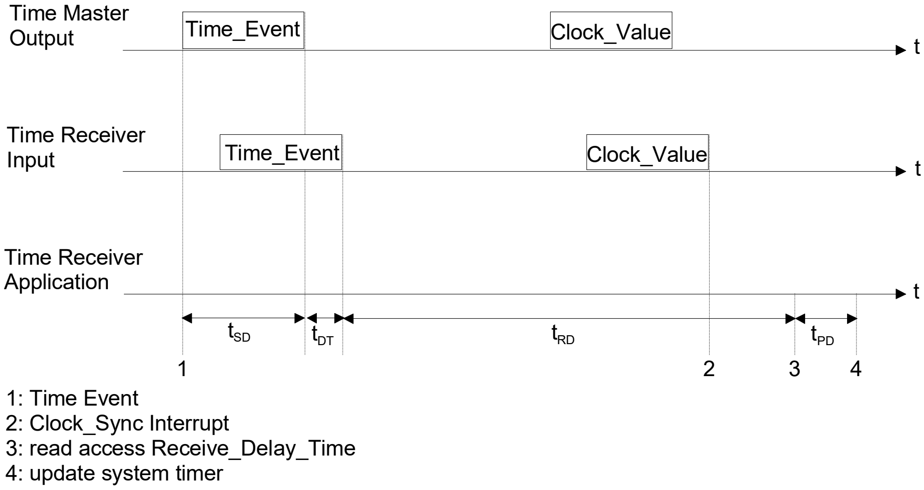



**Figure 7-26: Clock Synchronization Mechanism**

The clock synchronization sequence consists of two messages broadcasted by the time master. When the first message, called Time_Event, is received the VPC3+S starts the receive delay timer ($t_{RD}$). 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 ($t_{SD}$). 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 ($t_{PD}$: process delay time).

The time for transmission delay ($t_{DT}$: 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: $t_S$ = Clock_Value_Time_Event + $t_{DT}$ + $t_{RD}$ + $t_{PD}$

| 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 ($2^{31}$..0) | | | | | | | | CS Delay Time can be omitted |
| | Fraction Part of Seconds ($2^{31}$..0) Base is 1/($2^{32}$) Seconds | | | | | | | | |

**Figure 7-27: Format of Structured_Prm_Data with Time AR**

| Byte | Bit Position | | | | | | | | Designation |
|---|---|---|---|---|---|---|---|---|---|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
| 0 : 7 | Seconds ($2^{31}$..0) since 1.1.1900 0:00,00 or since 7.2.2036 6:28:16 if value < 0x9dff4400 | | | | | | | | Clock_Value_ Time_Event |
| | Fraction Part of Seconds ($2^{31}$..0) Base is 1/($2^{32}$) Seconds | | | | | | | | |
| 8 : 15 | Seconds ($2^{31}$..0) since 1.1.1900 0:00,00 or since 7.2.2036 6:28:16 if value < 0x9dff4400 | | | | | | | | Clock_Value_ previous_TE |
| | Fraction Part of Seconds ($2^{31}$..0) Base is 1/($2^{32}$) 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

<table>
<tr><th rowspan="2">Byte</th><th colspan="8">Bit Position</th><th rowspan="2">Designation</th></tr>
<tr><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td>0</td><td colspan="6">reserved</td><td>Clock_Sync_Violation</td><td>Set_Time</td><td>Status</td></tr>
<tr><td>1</td><td colspan="5">reserved</td><td>Clock_Value_Check_Ena</td><td>Ignore_Cyclic_State_Machine</td><td>Stop_Clock_Sync</td><td>Command</td></tr>
<tr><td>2</td><td>C</td><td colspan="5">CV</td><td colspan="2">reserved</td><td>Clock_Value_Status1</td></tr>
<tr><td>3</td><td>ANH</td><td>SWT</td><td>reserved</td><td colspan="2">CR</td><td colspan="2">reserved</td><td>SYF</td><td>Clock_Value_Status2</td></tr>
<tr><td rowspan="2">4<br>:<br>11</td><td colspan="8">Seconds ($2^{32}$-1 .. 0) since 1.1.1900 0:00,00<br>or since 7.2.2036 6:28:16 if value < 9DFF4400H</td><td rowspan="2">Clock_Value_<br>Time_Event</td></tr>
<tr><td colspan="8">Fraction Part of Seconds ($2^{32}$-1 .. 0)<br>Base is 1/($2^{32}$) Seconds</td></tr>
<tr><td>12<br>:<br>15</td><td colspan="8">($2^{32}$-1 .. 0)<br>Time Base 1 µs</td><td>Receive_Delay_Time</td></tr>
<tr><td rowspan="2">16<br>:<br>23</td><td colspan="8">Seconds ($2^{32}$-1 .. 0) since 1.1.1900 0:00,00<br>or since 7.2.2036 6:28:16 if value < 9DFF4400H</td><td rowspan="2">Clock_Value_<br>previous_TE</td></tr>
<tr><td colspan="8">Fraction Part of Seconds ($2^{32}$-1 .. 0)<br>Base is 1/($2^{32}$) Seconds</td></tr>
<tr><td>24<br>:<br>25</td><td colspan="8">($2^{16}$-1 .. 0)<br>Time Base 10 ms</td><td>Clock_Sync_Interval</td></tr>
</table>

| | Clock_Sync-Buffer |
|---|---|
| Status<br>bit 7-2<br>r-000000 | **Reserved** |
| Status<br>bit 1<br>r-0 | **Clock_Sync_Violation**:<br>Wrong telegram or Time period of 2*$T_{CSI}$ expired after reception of Time_Event. |
| Status<br>bit 0<br>r-0 | **Set_Time**:<br>The VPC3+D has received a valid 'Clock_Value telegram' and made the data available in the Clock_Sync-Buffer. |
| Command<br>bit 7-3<br>r-00000 | **Reserved** |
| Command<br>bit 2<br>rw-0 | **Clock_Value_Check_Ena**:<br>0 = don't evaluate Clock_Value_previous_TE<br>1 = check Clock_Value_previous_TE with local variable Time_Last_Rcvd |
| Command<br>bit 1<br>rw-0 | **Ignore_Cyclic_State_Machine**:<br>0 = Clock Synchronization stops after the receiption of a new Set_Prm or a LEAVE-MASTER<br>1 = Clock Synchronization continues until the user set Stop_Clock_Sync |
| Command<br>bit 0<br>w-0 | **Stop_Clock_Sync**:<br>Stop the Clock Synchronization, in order to write a new $T_{CSI}$ without a previous Set_Prm or LEAVE-MASTER. The Bit is cleared by the Time_Receiver State Machine. |
| Clock_Value_ Status1<br>bit 7<br>r-0 | **C: Sign of CV**<br>0 = add correction value to Time<br>1 = substract correction value to Time |
| Clock_Value_ Status1<br>bit 6-2<br>r-00000 | **CV: Correction Value**<br>0 = 0 min<br>1..31 = 30..930 min |
| Clock_Value_Status1<br>bit 1-0<br>r-00 | **Reserved** |
| Clock_Value_ Status2<br>bit 7<br>r-0 | **ANH: Announcement Hour**<br>0 = no change planned within the next hour<br>1 = a change of SWT will occur within the next hour |
| Clock_Value_ Status2<br>bit 6<br>r-0 | **SWT: Summertime**<br>0 = Winter Time<br>1 = Summer Time |
| Clock_Value_Status2<br>bit 5<br>r-0 | **Reserved** |

| | **Clock_Sync-Buffer** |
|---|---|
| Clock_Value_ Status2<br>bit 4-3<br>r-00 | **CR: Accuracy**<br>0 = 1 ms<br>1 = 10 ms<br>2 = 100 ms<br>3 = 1 s |
| Clock_Value_Status2<br>bit 2-1<br>r-00 | **Reserved** |
| Clock_Value_ Status2<br>bit 0<br>r-0 | **SYF: Synchronisation Active**:<br>0 = Clock_Value_Time_Event is synchronized<br>1 = Clock_Value_Time_Event is not synchronized |
| r-0 | **Clock_Value_Time_Event**:<br>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**:<br>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**:<br>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**:<br>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<br>set New_(Ext_)Prm_Data interrupt | |
| | acknowledge interrupt<br>write Clock_Sync_Interval to CS-Buffer |
| reception of Time_Event<br>start Receive_Delay_Timer<br>reception of Clock_Value<br>set Clock_Sync interrupt | |
| | read CS_Status<br>IF (Set_Time='1') THEN |
| stop Receive_Delay Timer | read CS_Buffer |
| | update system time<br>END IF<br>acknowledge interrupt |

**Figure 7-30: communication scheme**

# 8 Hardware Interface

## 8.1 Universal Processor Bus Interface

### 8.1.1 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 microcontrollers 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.

### 8.1.2 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 configuration 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.

**Note:**
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 | | AB10 has a pull down resistor. |
| MODE | I | | Configuration: syn/async interface |
| XWR/E_CLOCK<br>AB11 | I | | Intel: Write Sync. Motorola: E-Clk<br>AB11 (Asynchronous Motorola Mode) |
| XRD/R_W | I | | Intel: Read Motorola: Read/Write |
| XCS<br>AB11 | I | | Chip Select<br>AB11 (Synchronous Intel Mode) |
| ALE/AS<br>AB11 | I | | Intel/Motorola: Address Latch Enable<br>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 [1] | 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**

[1] 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 internal 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 microcontroller
- 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, priority, 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 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.
- 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.

### 8.1.3 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.

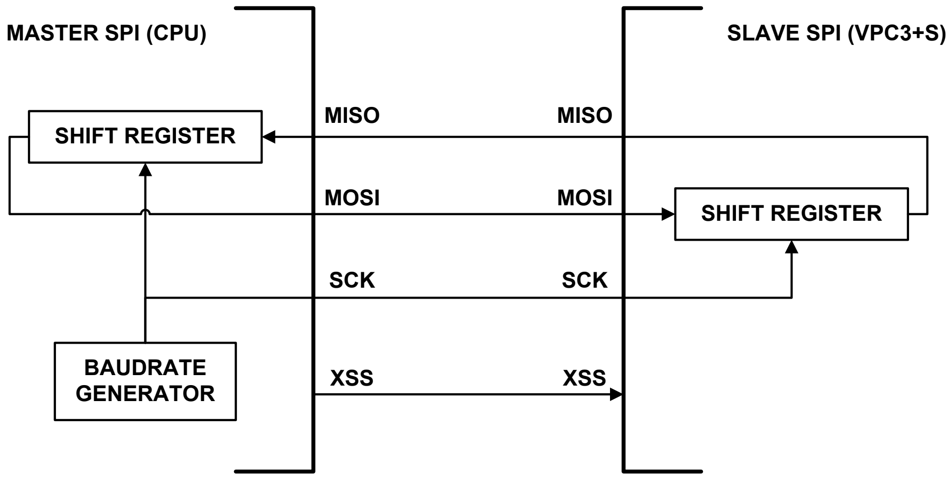



**Figure 8-3: SPI Master-Slave-Transfer (Block Diagram)**

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 received (shifted in serially) simultaneously. The serial clock (SCK) synchronizes 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.

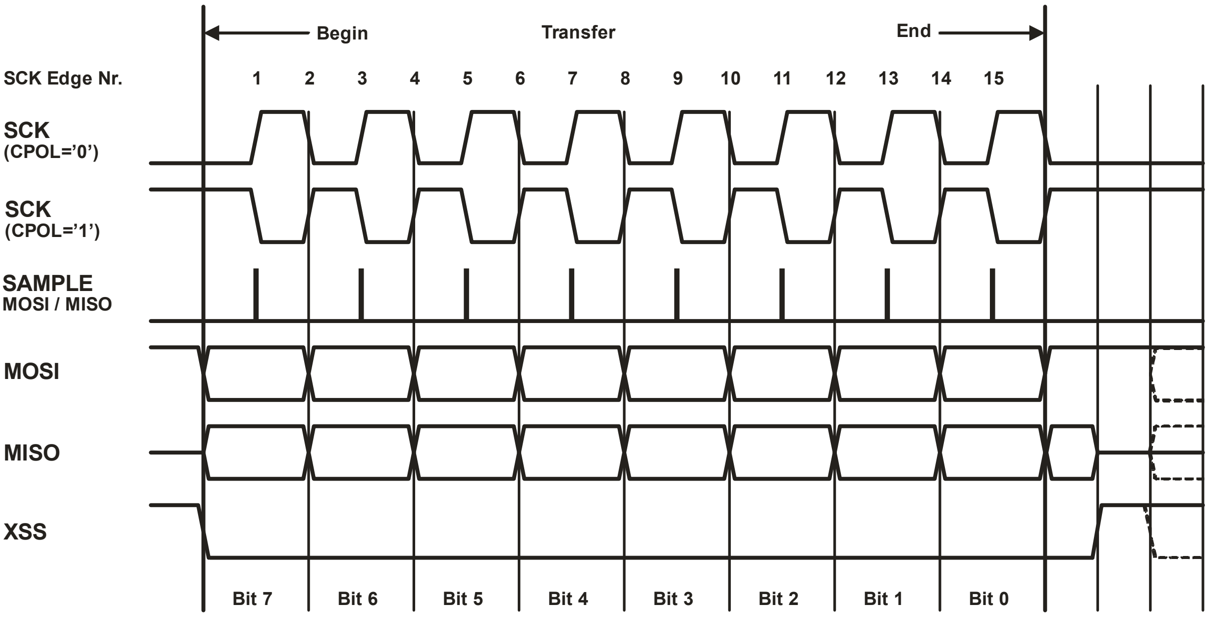



**Figure 8-4: SPI Transfer Format (CPHA='0')**

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.

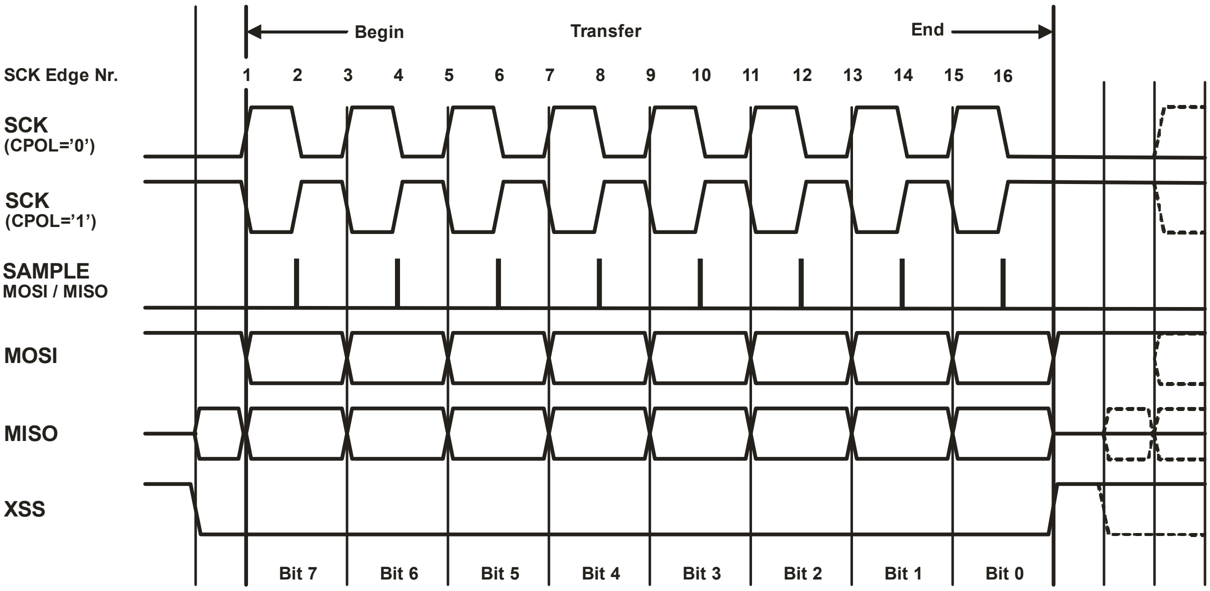



**Figure 8-5: SPI Transfer Format (CPHA='1')**

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 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).

**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.

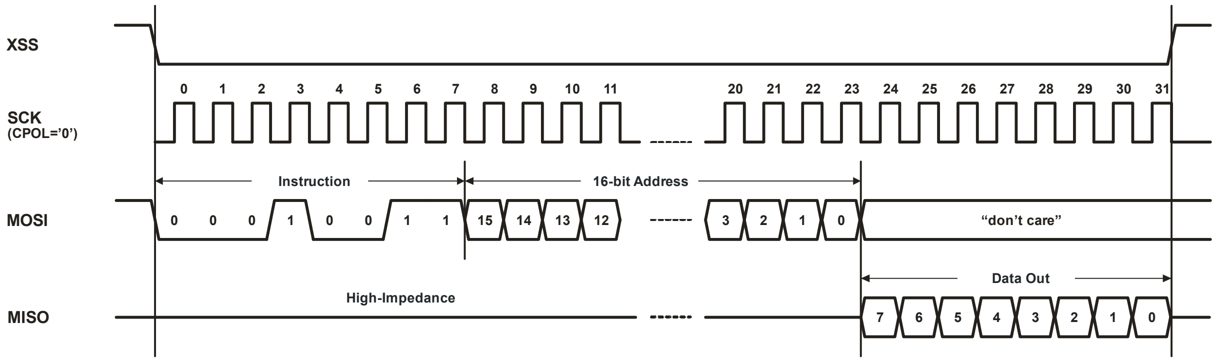



**Figure 8-7: READ BYTE Sequence**

## READ ARRAY 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 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 Address 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 8-8).

**Note:**
The SPI instruction READ ARRAY may <u>not</u> 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 location will occur leading to an unpredictable behavior of the VPC3+S.

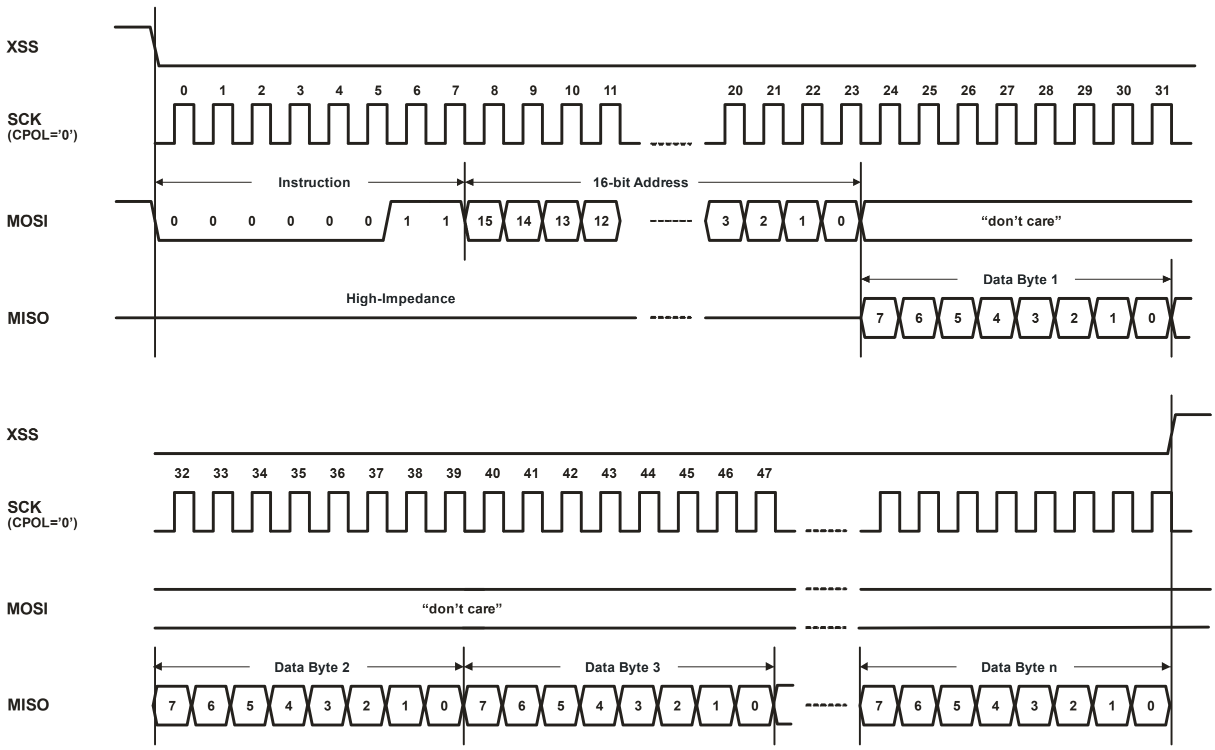



**Figure 8-8: READ ARRAY Sequence**

## 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.

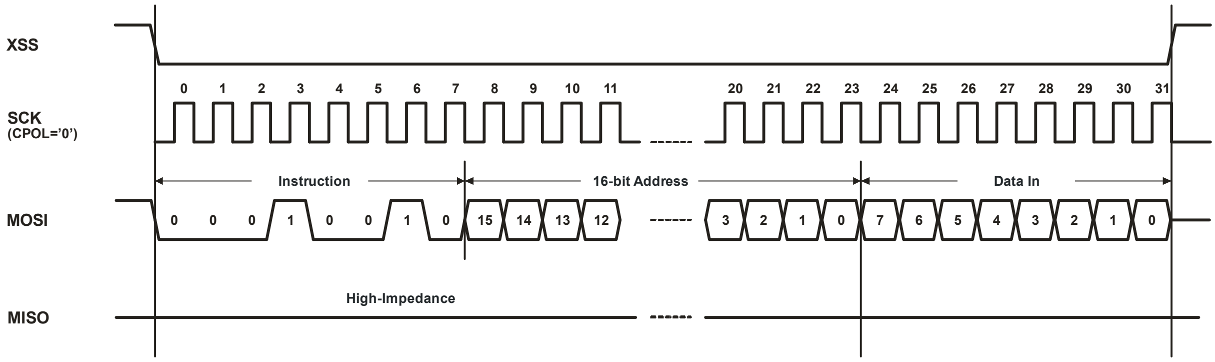



**Figure 8-9: WRITE BYTE Sequence**

## 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.

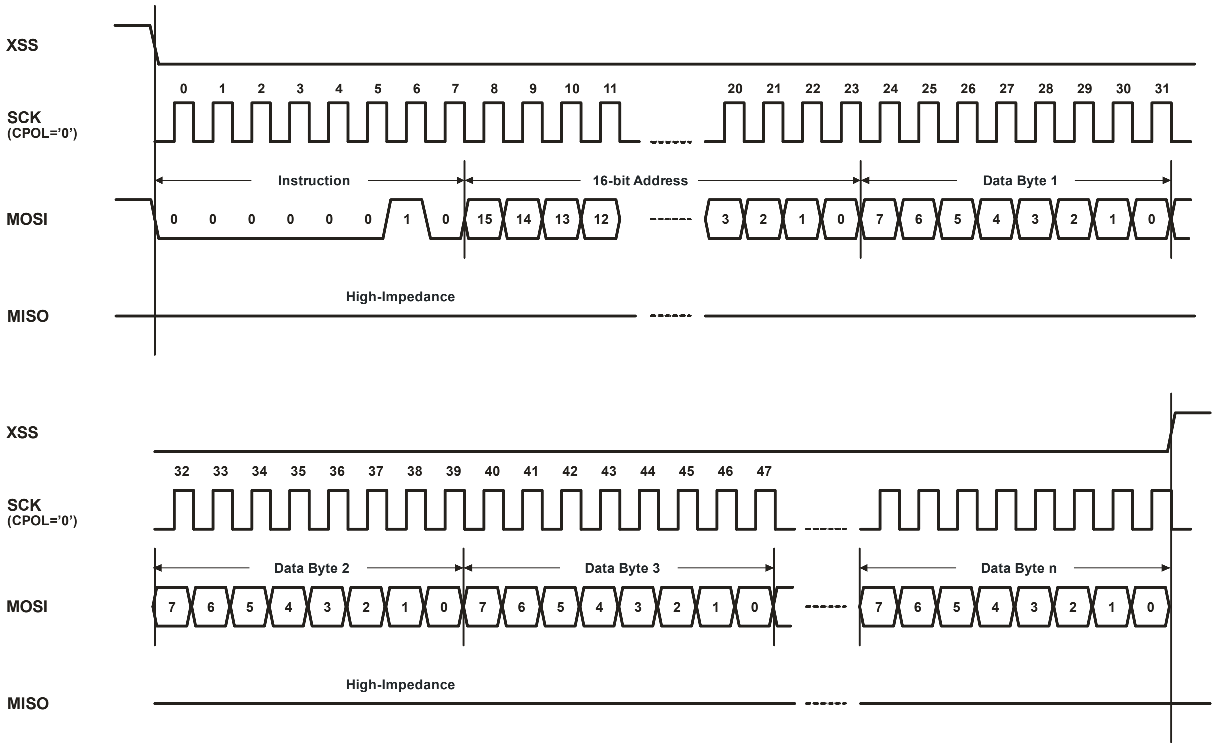



**Figure 8-10: WRITE ARRAY Sequence**

### 8.1.4 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). One clock pulse is generated for each data bit transferred.

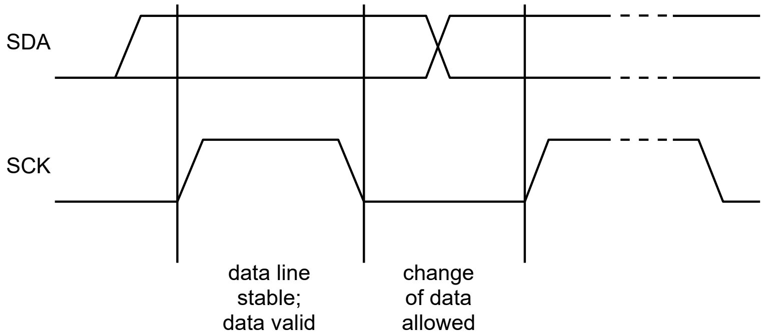



**Figure 8-11: Bit Transfer on the I2C bus**

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.

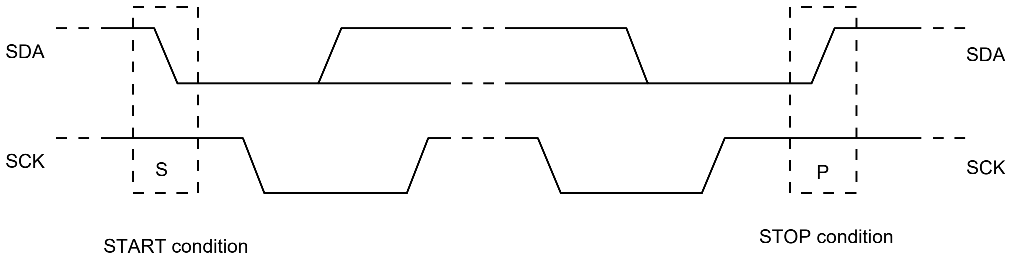



**Figure 8-12: START and STOP condition**

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.

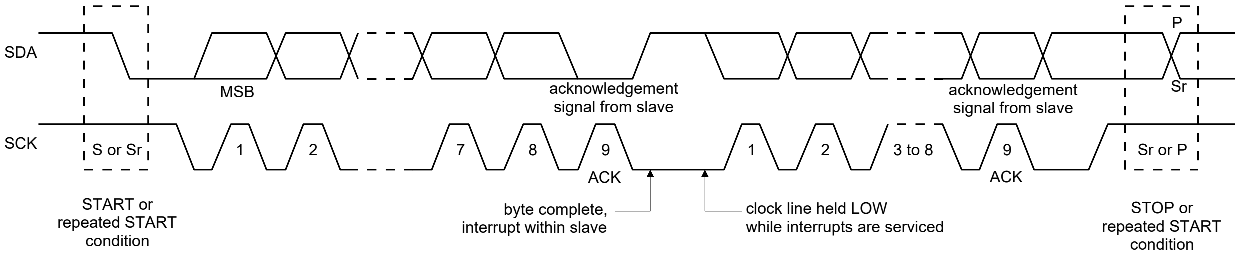



**Figure 8-13: Data Transfer on the I2C Bus**

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). 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.

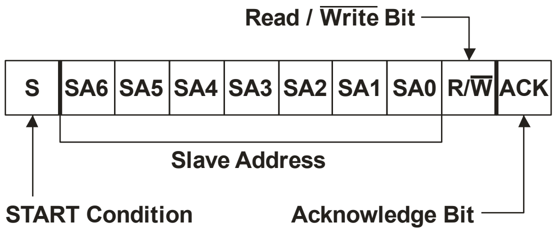



**Figure 8-14: Control Byte Format**

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 8-15). 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.

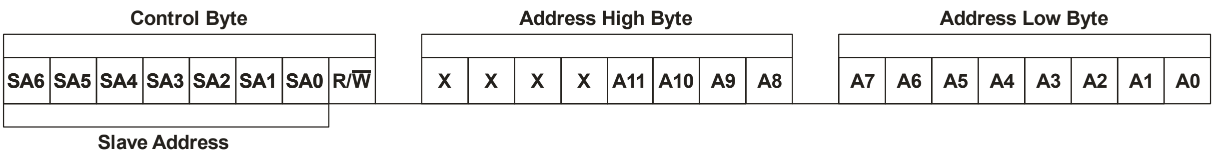



**Figure 8-15: Address Sequence Bit Assignments**

## 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.

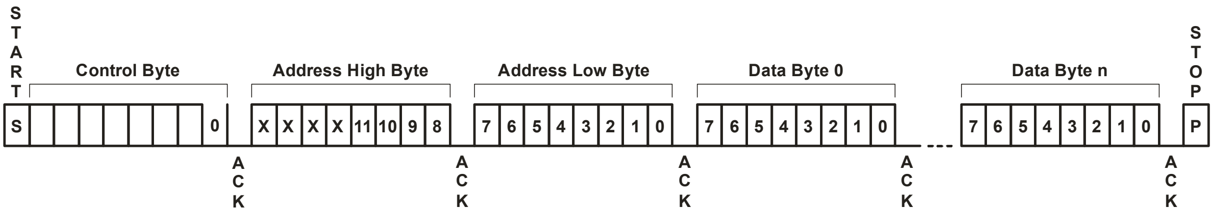



**Figure 8-16: I2C WRITE Sequence**

## 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.

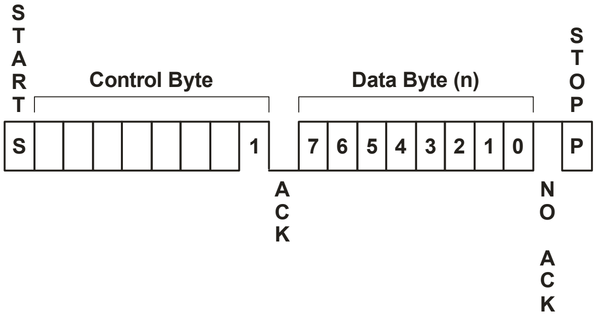



**Figure 8-17: I2C Current Address READ Operation**

## 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). After a random Read command, the internal address counter will point to the address location following the one that was just read.

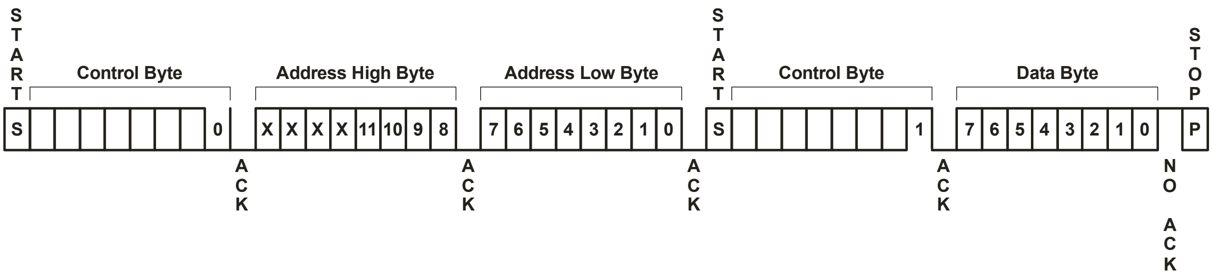



**Figure 8-18: I2C Random READ Operation**

## 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). 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).

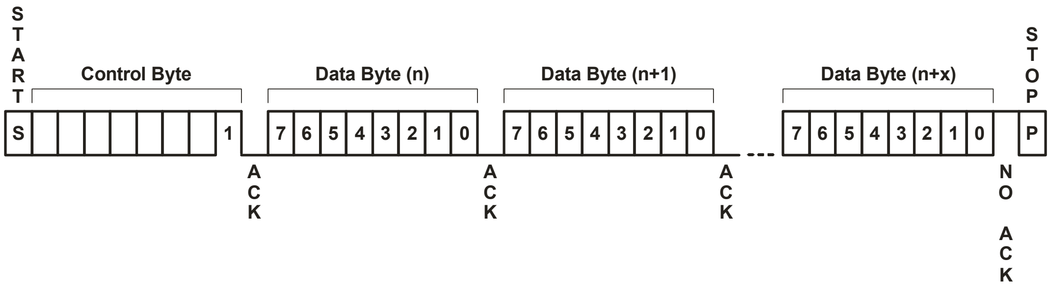



**Figure 8-19: I2C Sequential READ Operation**

### 8.1.5 Application Examples (Principles)

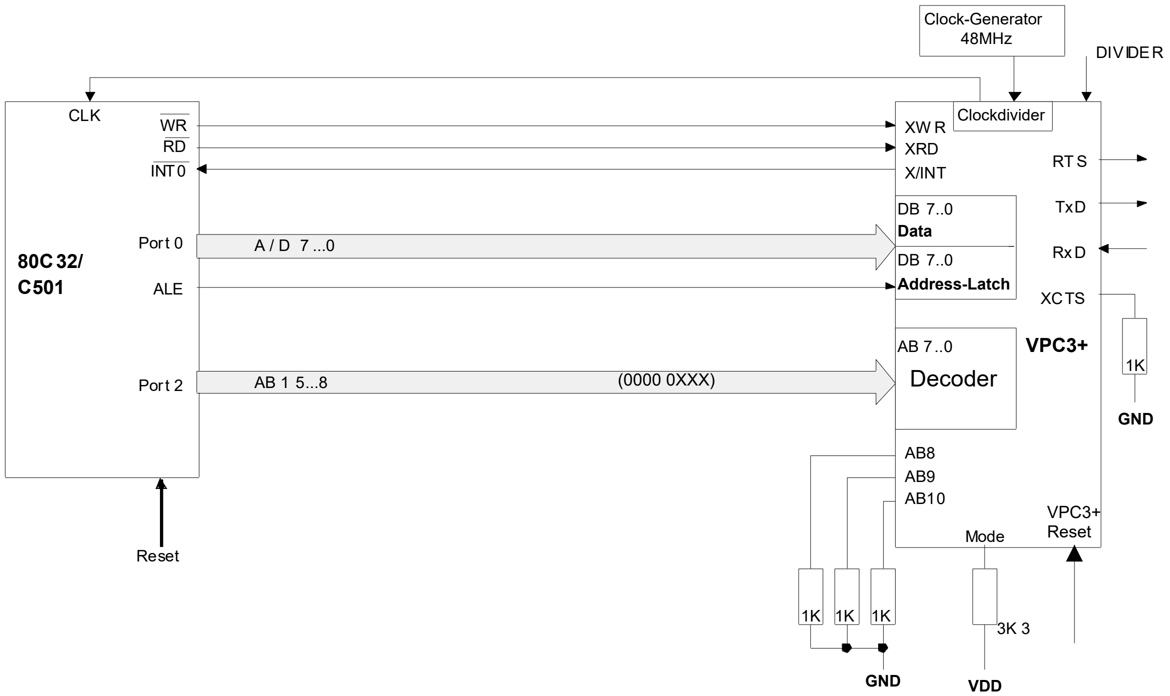



**Figure 8-20: Low Cost System with 80C32**

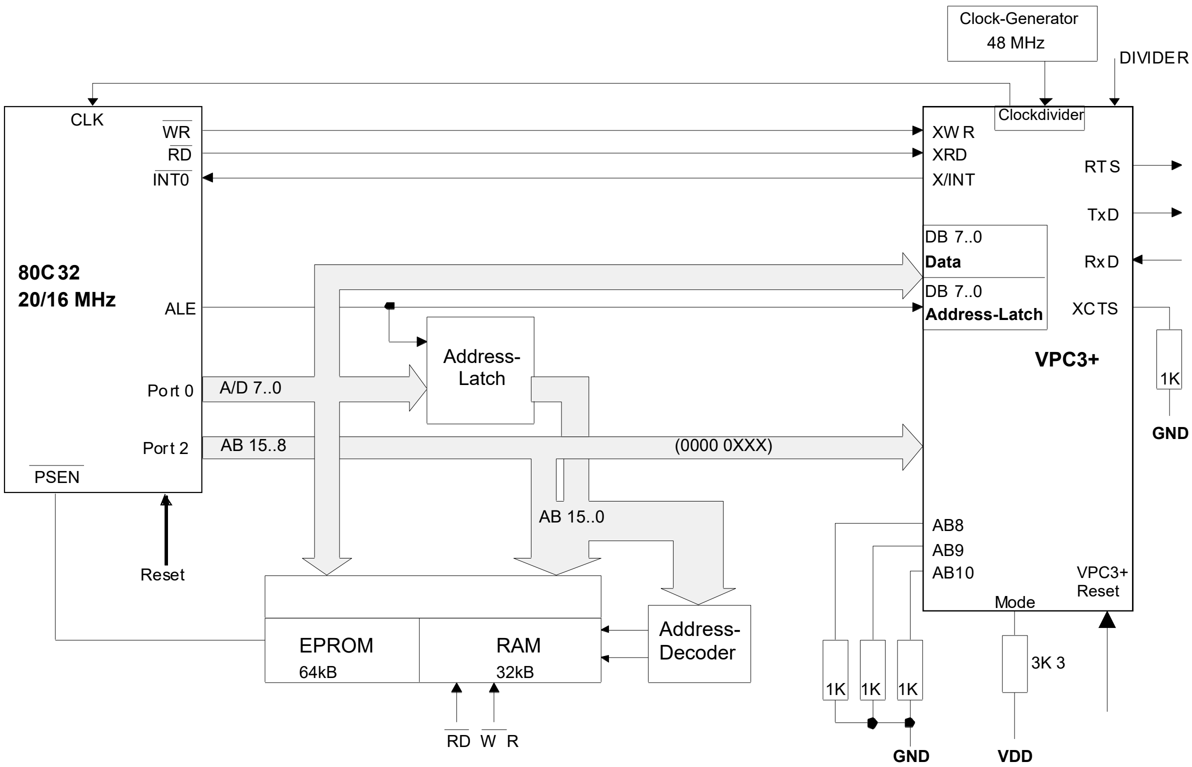



**Figure 8-21: 80C32 System with External Memory**

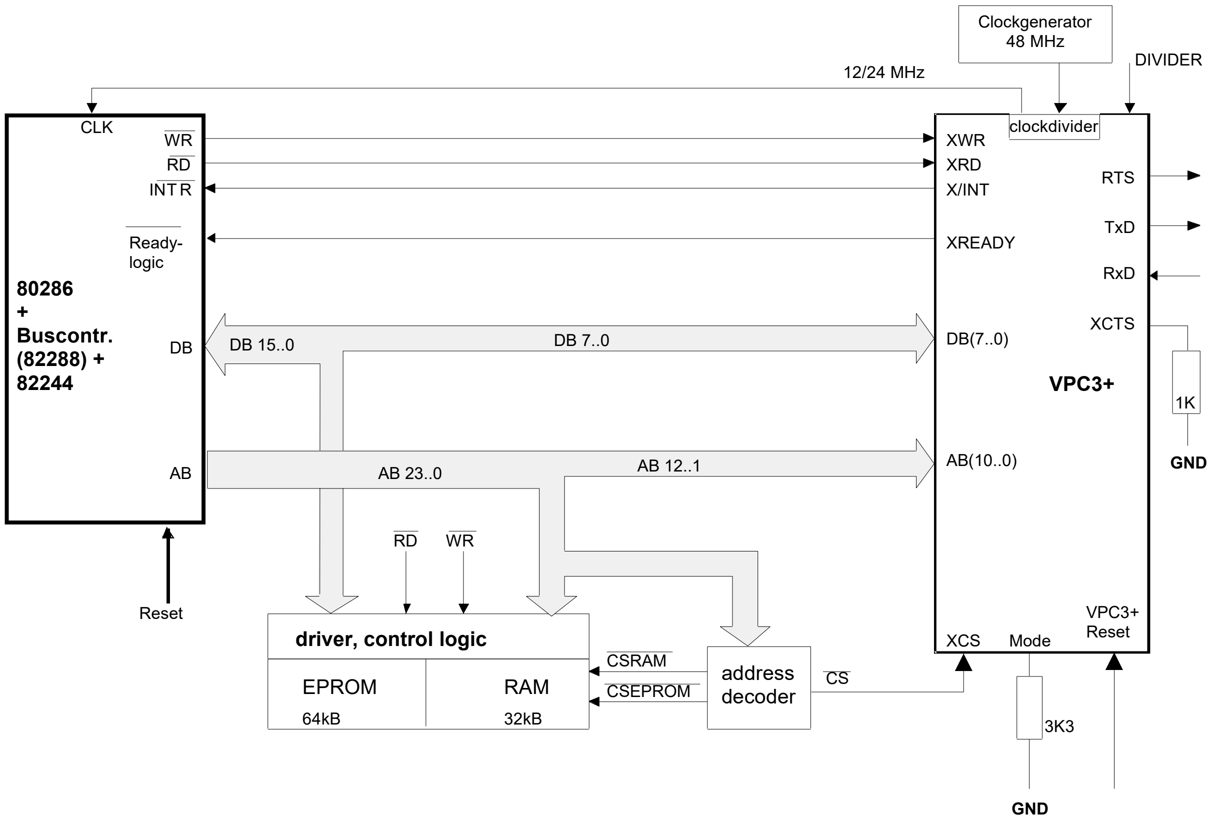



**Figure 8-22: 80286 System (X86 Mode)**

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

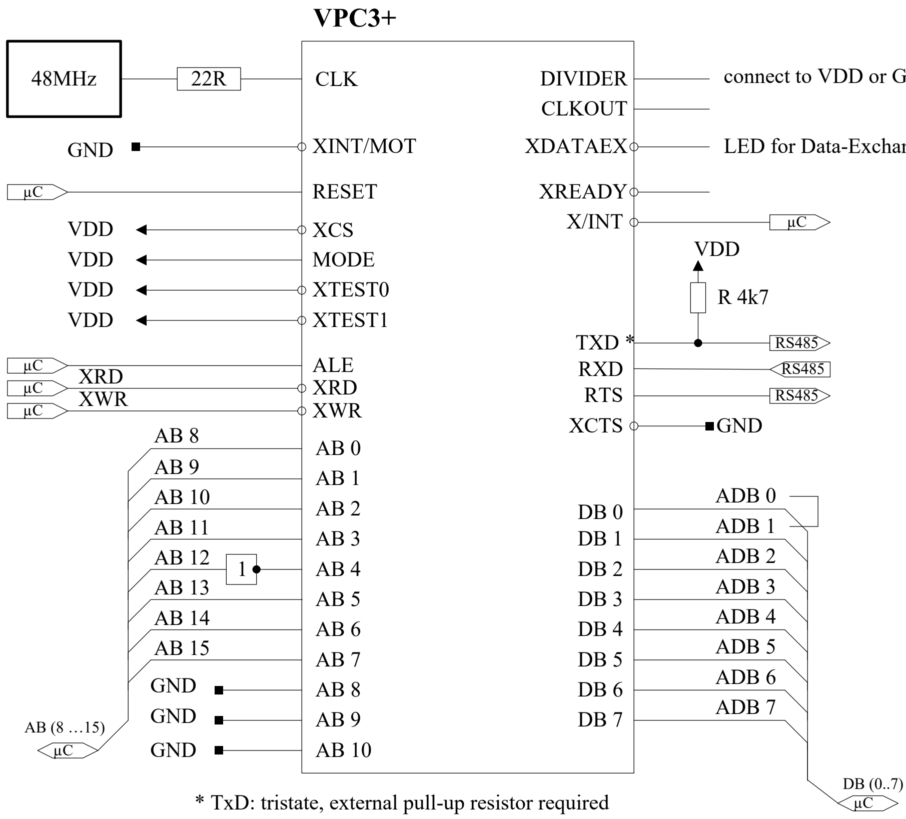



**Figure 8-23: 80C32 Application in 2K Byte mode**

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.

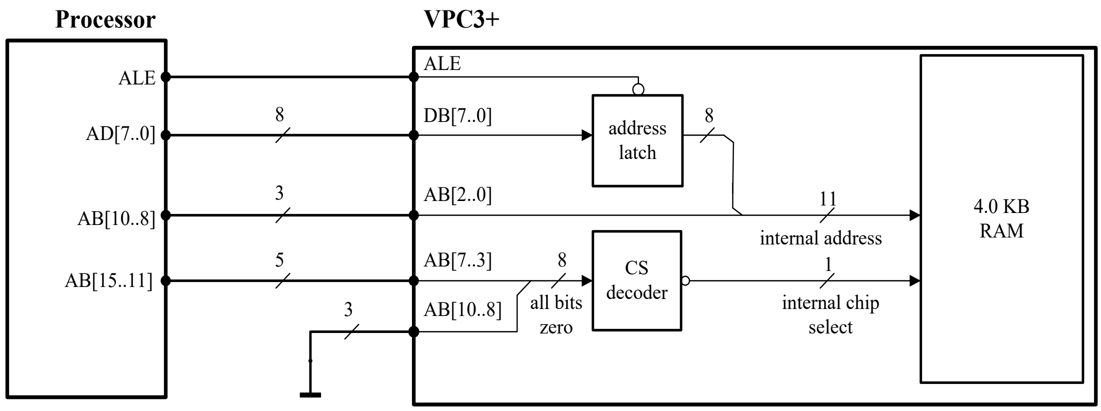



**Figure 8-24: Internal Chipselect Generation in Synchronous Intel Mode, 2K Byte RAM**

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

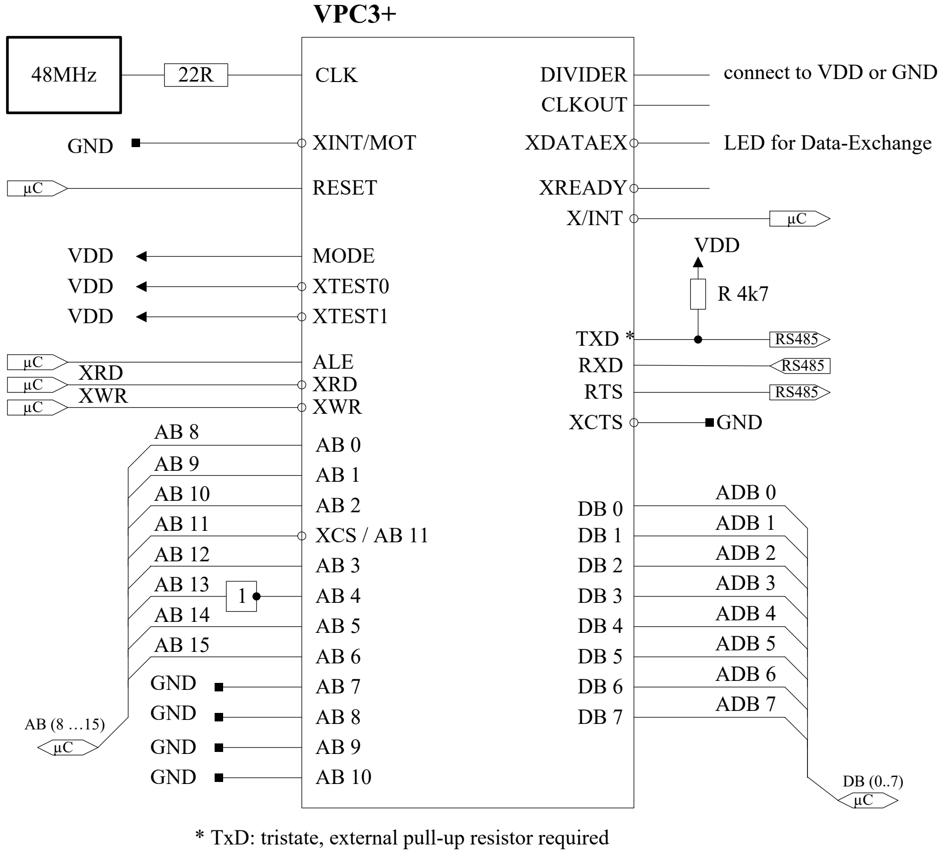



**Figure 8-25: 80C32 Application in 4K Byte mode**

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.

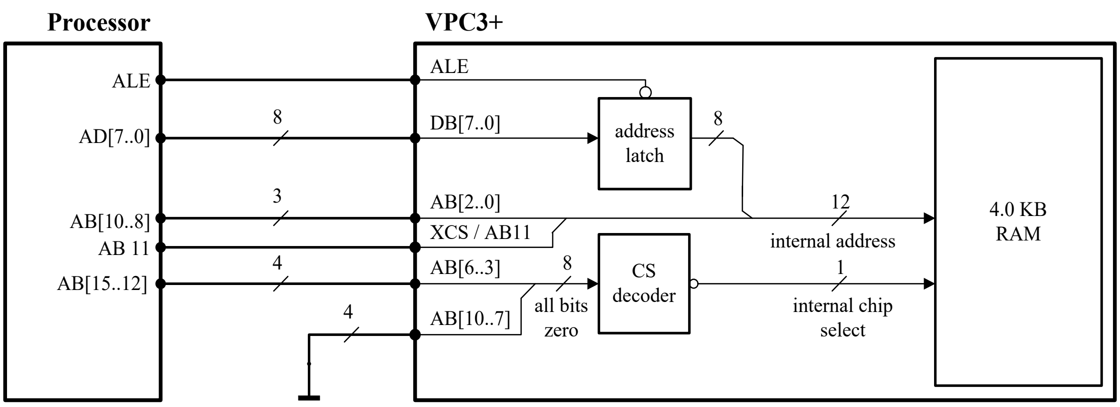



**Figure 8-26 : Internal Chipselect Generation in Synchronous Intel Mode, 4K Byte RAM**

### 8.1.8 Application with 80C165

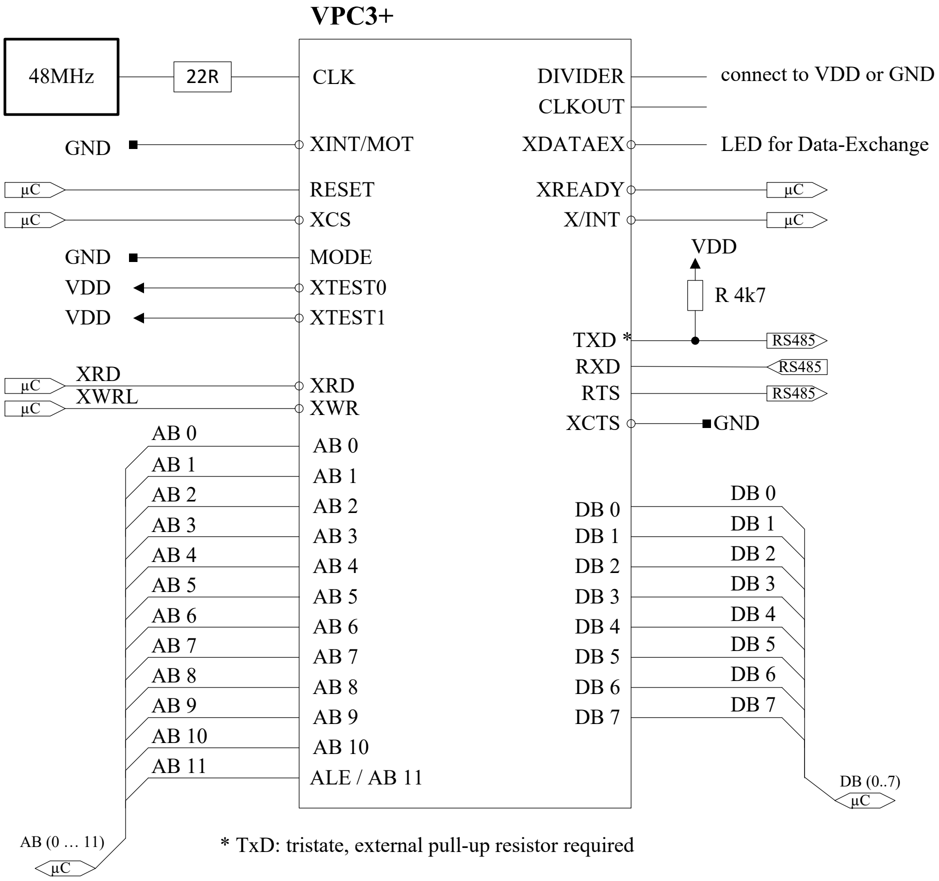



**Figure 8-27: 80C165 Application**

## 8.2 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.

## 8.3 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 activates 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 generates 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.

## 8.4 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**

# 9 PROFIBUS Interface

## 9.1 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, D-Sub, plug connector with the following pin assignment.

Pin 1 - Free
Pin 2 - Free
Pin 3 - B line (Receive data / transmission data plus)
Pin 4 - Request to send (RTS)
Pin 5 - Ground 5V (M5)
Pin 6 - Potential 5V (floating P5)
Pin 7 - Free
Pin 8 - A line (Receive data / transmission data negative)
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.

## 9.2 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.

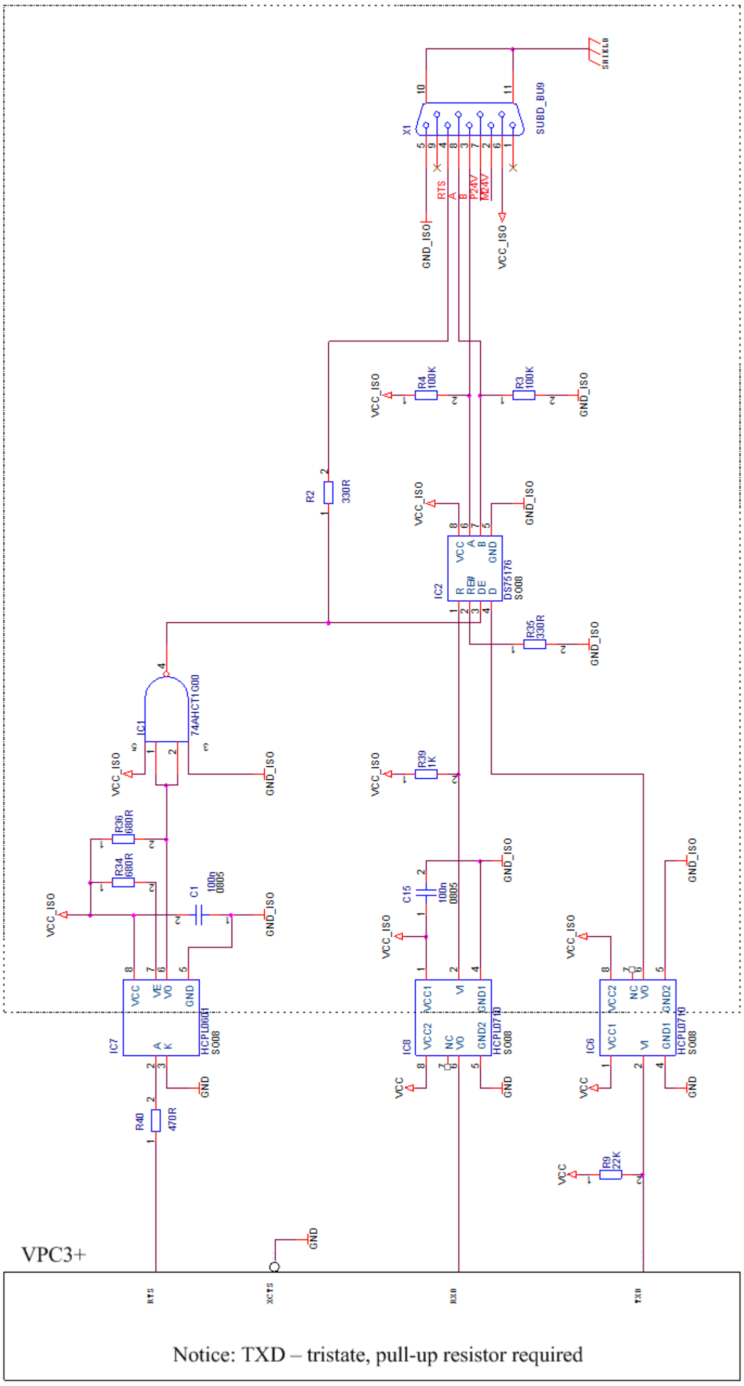



**Figure 9-2: Example for the RS485 Interface**

# 10 Operational Specifications

## 10.1 Absolute Maximum Ratings

| Parameter | Symbol | Limits | Unit |
|---|---|---|---|
| DC supply voltage | VCC | -0.3 to 3.9 | V |
| Input voltage | $V_I$ | -0.3 to 5.5 | V |
| Output voltage | $V_O$ | -0.3 to VCC +0.3 | V |
| DC output current | $I_O$ | See Figure 10-4 | mA |
| Storage temperature | $T_{store}$ | -40 to +150 | °C |

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

## 10.2 Recommended Operating Conditions

| Parameter | Symbol | MIN | MAX | Unit |
|---|---|---|---|---|
| DC supply voltage | VCC | 3.00 | 3.6 | V |
| Static supply current | $I_{cc}$ | | 100[1] | µA |
| Circuit ground | GND | 0 | 0 | V |
| Input voltage | $V_I$ | 0 | 5.50 | V |
| Input voltage (HIGH level) | $V_{IH}$ | 2.00 | 5.50 | V |
| Input voltage (LOW level) | $V_{IL}$ | 0 | 0.8 | V |
| Output voltage | $V_O$ | 0 | VCC | V |
| Ambient temperature | $T_A$ | -40 | +85 | °C |

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

[1] Static $I_{DD}$ current is exclusively of input/output drive requirements and is measured with the clock stopped and all inputs tied to VCC or GND.

## 10.3 General DC Characteristics

| Parameter | Symbol | MIN | TYP | MAX | Unit |
|---|---|---|---|---|---|
| Input LOW current | $I_{IL}$ | -1 | | +1 | µA |
| Input HIGH current | $I_{IH}$ | -1 | | +1 | µA |
| Tri-state leakage current | $I_{OZ}$ | -10 | | +10 | µA |
| Current consumption (3.3V) | $I_A$ | | 30 | | mA |
| Input capacitance | $C_{IN}$ | | 5 | | pF |
| Output capacitance | $C_{OUT}$ | | 5 | | pF |
| Bi-directional buffer capacitance | $C_{BID}$ | | 5 | | pF |
| Thermal Resist. (BGA48) | $\Theta_{JA}$ | | 43.6 | | K/W |
| Thermal Resist. (QFP48) | $\Theta_{JA}$ | | 72.2 | | K/W |

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

## 10.4 Ratings for the Output Drivers

| Signal | Direction | Driver Type | Driver Strength | Max. Cap. Load |
|---|---|---|---|---|
| DB 0-7 | I/O | Tristate | 8 mA | 50 pF |
| RTS | O | Push/Pull | 8 mA | 50 pF |
| TXD | O | Tristate | 8 mA | 50 pF |
| INT | O | Push/Pull | 8 mA | 50 pF |
| XREADY/XDTACK<br>SPI_MISO<br>I2C_SDA | O<br>O<br>I/O | Push/Pull<br>Push/Pull<br>Tristate | 8 mA | 50 pF |
| XDATAEXCH | O | Push/Pull | 8 mA | 50 pF |
| CLKOUT | O | Push/Pull | 8 mA | 50 pF |

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

## 10.5 DC Electrical Characteristics

| Parameter | Symbol | MIN | TYP | MAX | Unit |
|---|---|---|---|---|---|
| DC supply voltage | VCC | 3.00 | 3.30 | 3.60 | V |
| Input voltage LOW level | $V_{IL}$ | | | 0.8 | V |
| Input voltage HIGH level | $V_{IH}$ | 2.0 | | | V |
| Output voltage LOW level | $V_{OL}$ | | | 0.4 | V |
| Output voltage HIGH level | $V_{OH}$ | 2.4 | | | V |
| Schmitt Trigger<br>negative going threshold voltage | $V_{T-}$ | 0.9 | 1.1 | | V |
| Schmitt Trigger<br>positive going threshold voltage | $V_{T+}$ | | 1.6 | 1.9 | V |
| Input LOW current | $I_{IL}$ | -1 | | +1 | µA |
| Input HIGH current | $I_{IH}$ | -1 | | +1 | µA |
| Tri-state leakage current | $I_{OZ}$ | -10 | ±1 | +10 | µA |
| Output current LOW level, 8 mA cell | $I_{OL}$ | +8 | | | mA |
| Output current HIGH level, 8 mA cell | $I_{OH}$ | -8 | | | mA |
| Input pull-down resistance | $R_{PD}$ | 40 | 75 | 190 | kΩ |

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

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

## 10.6 Timing Characteristics

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

### 10.6.1 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 | $T_{CH}$ | 6.25 | 14.6 | ns |
| Clock low time | $T_{CL}$ | 6.25 | 14.6 | ns |
| Clock rise time | $T_{CR}$ | | 4 | ns |
| Clock fall time | $T_{CF}$ | | 4 | ns |

**Figure 10-6: Clock Timing**

**Note:**
The VPC3+S is equipped with 5 V 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.

### 10.6.2 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).

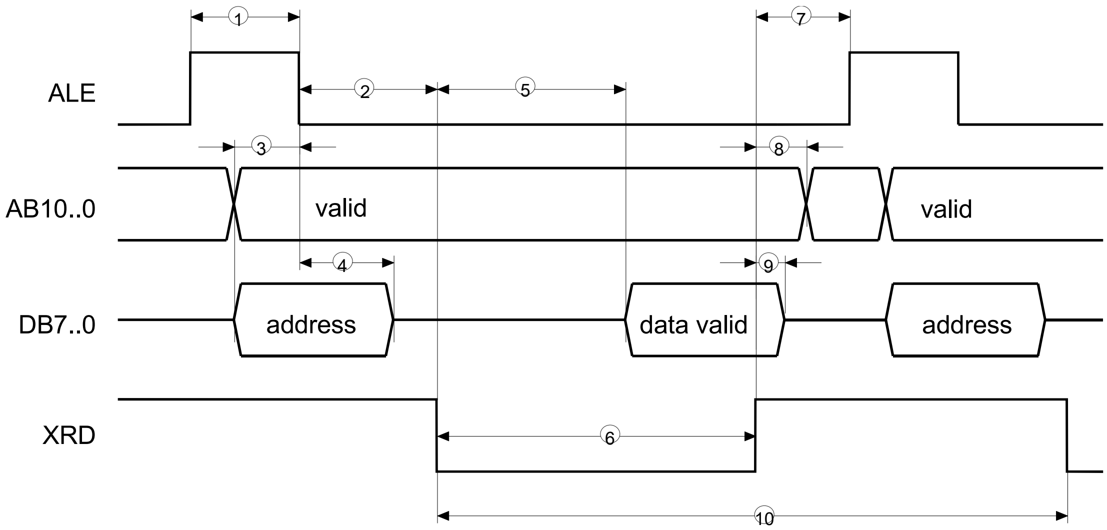



**Figure 10-8: Synchronous Intel Mode, READ (XWR = 1)**

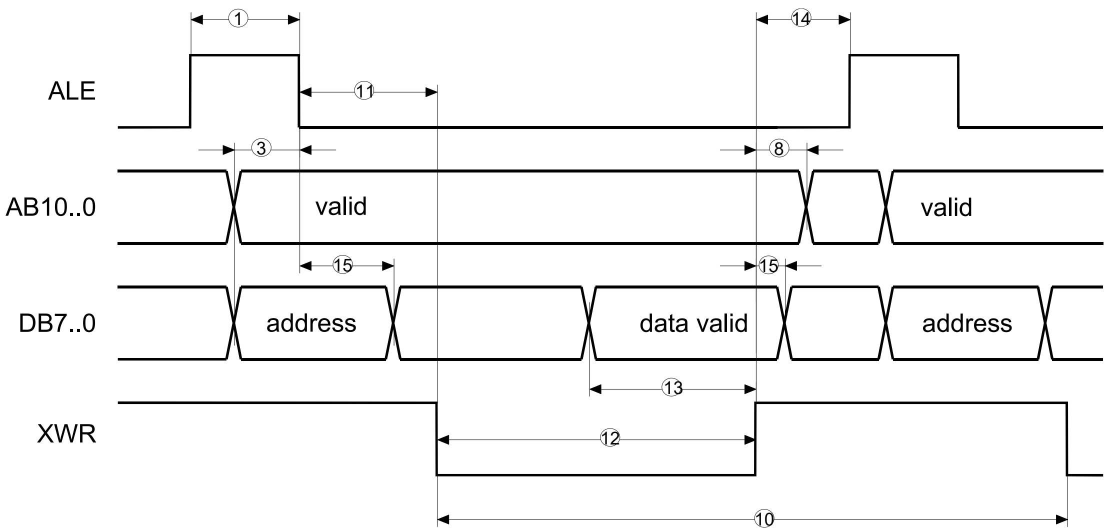



**Figure 10-9: Synchronous Intel Mode, WRITE (XRD = 1)**

| 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**

### 10.6.3 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'.

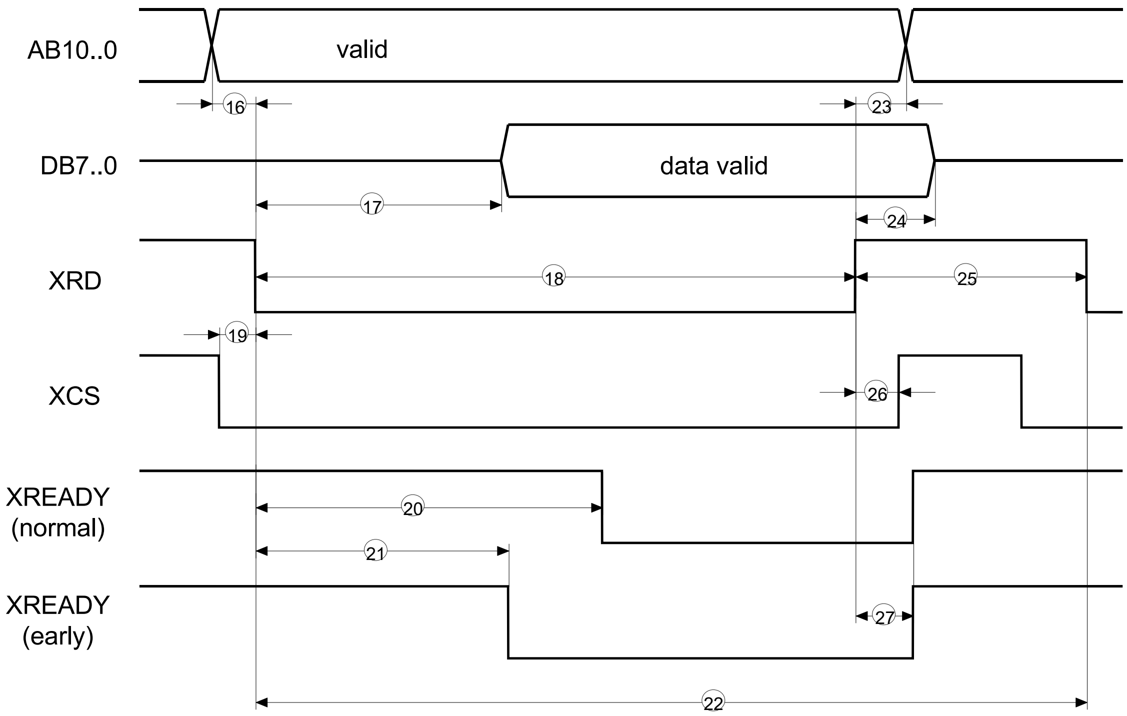



**Figure 10-11: Asynchronous Intel Mode, READ (XWR = 1)**

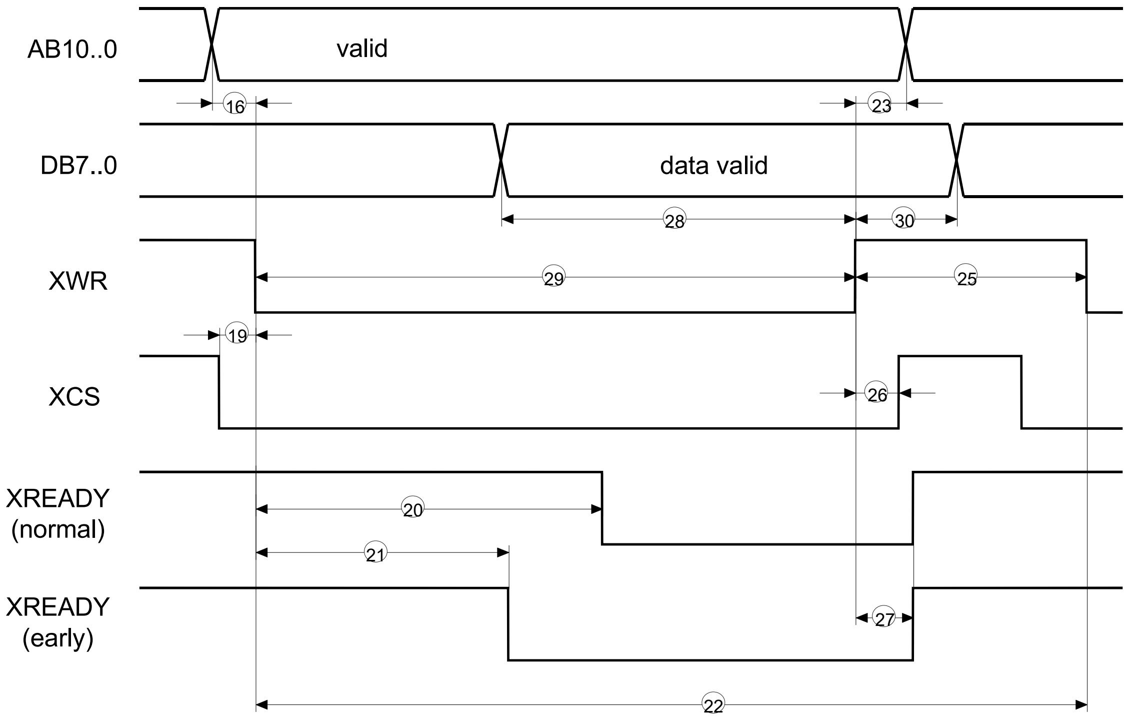



**Figure 10-12: Asynchronous Intel Mode, WRITE (XRD = 1)**

| 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**

### 10.6.4 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).

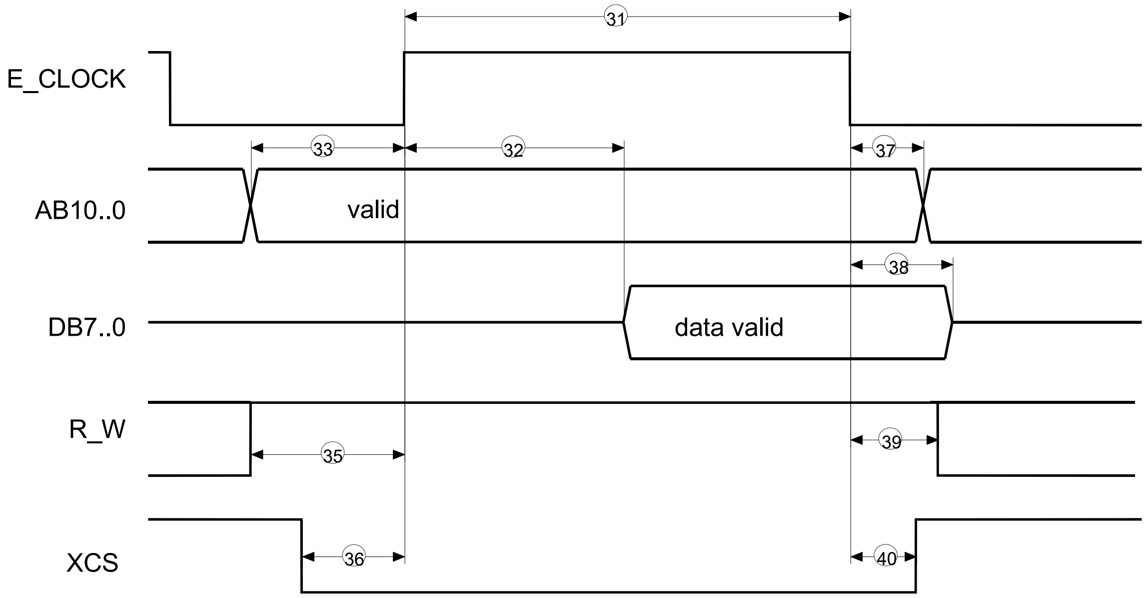



**Figure 10-14: Synchronous Motorola-Mode, READ (AS = 1)**

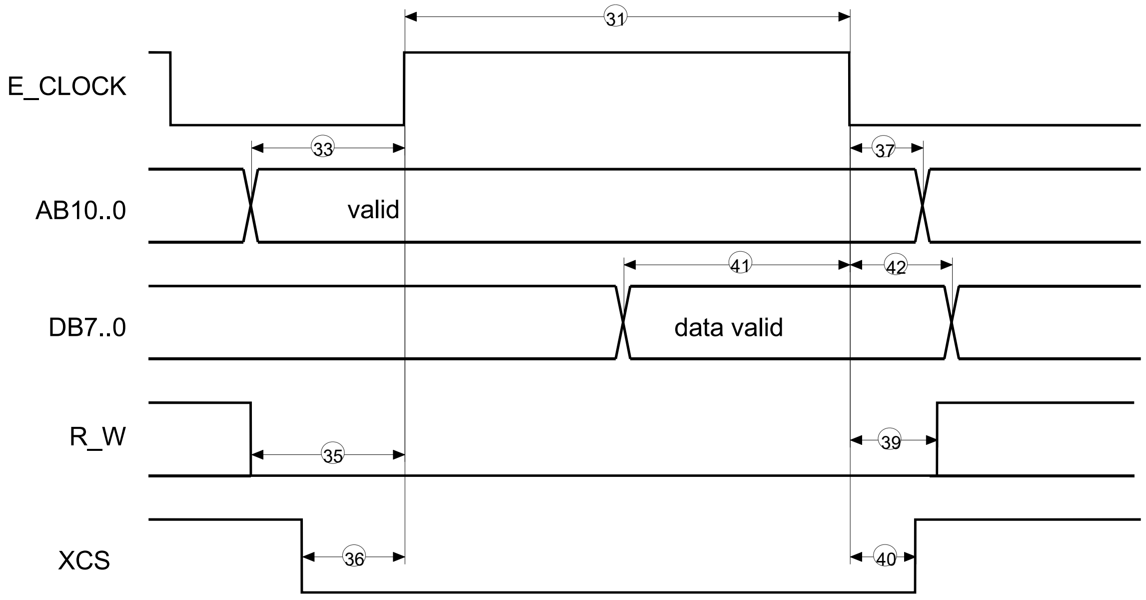



**Figure 10-15: Synchronous Motorola-Mode, WRITE (AS = 1)**

| 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**

### 10.6.5 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').

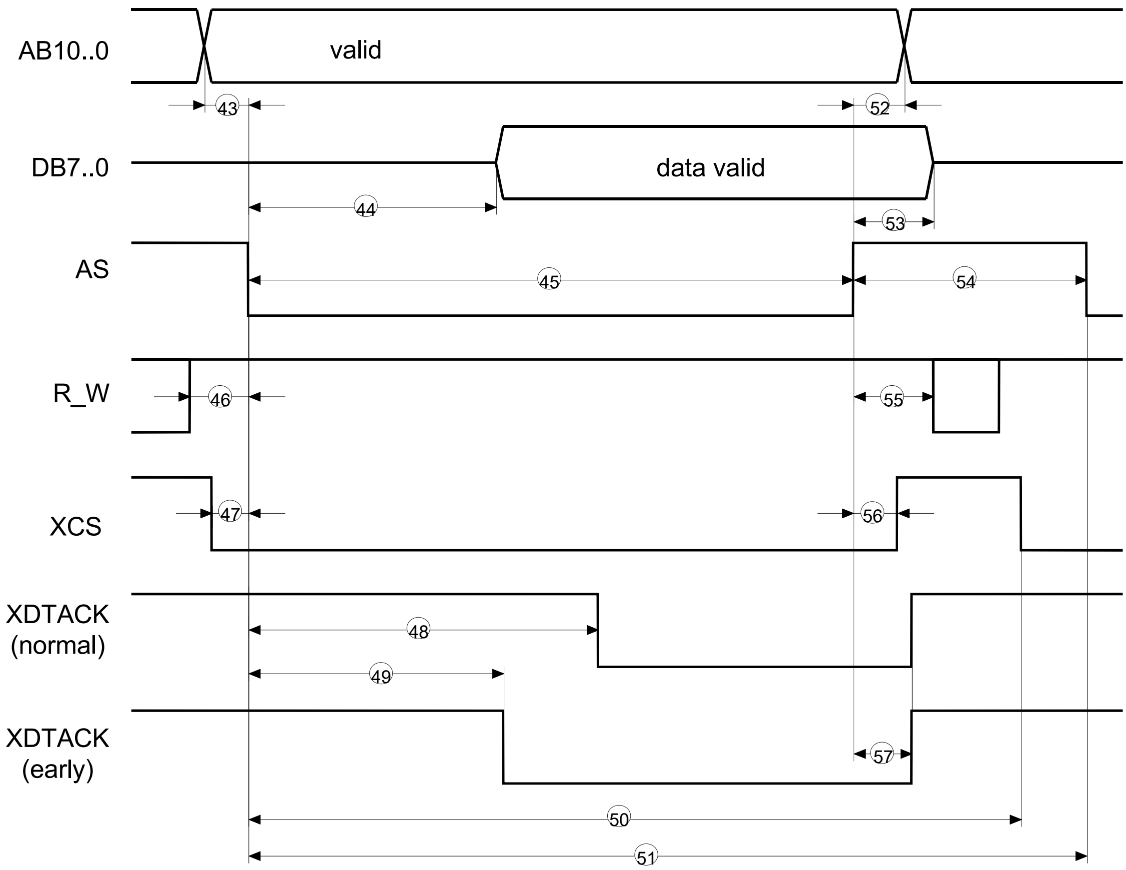



**Figure 10-17: Asynchronous Motorola Mode, READ (E_CLOCK = 0)**

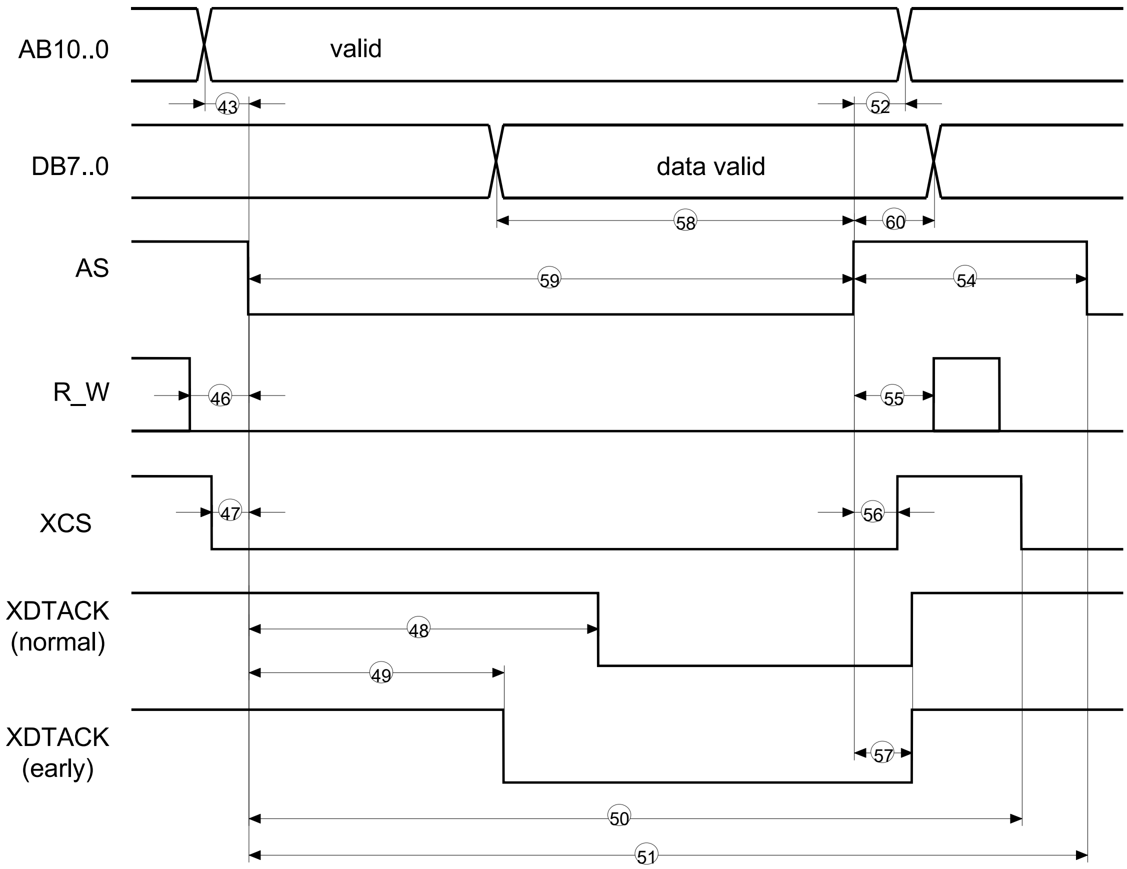



**Figure 10-18: Asynchronous Motorola Mode (WRITE)**

| 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**

### 10.6.6 Timing in SPI Interface Mode

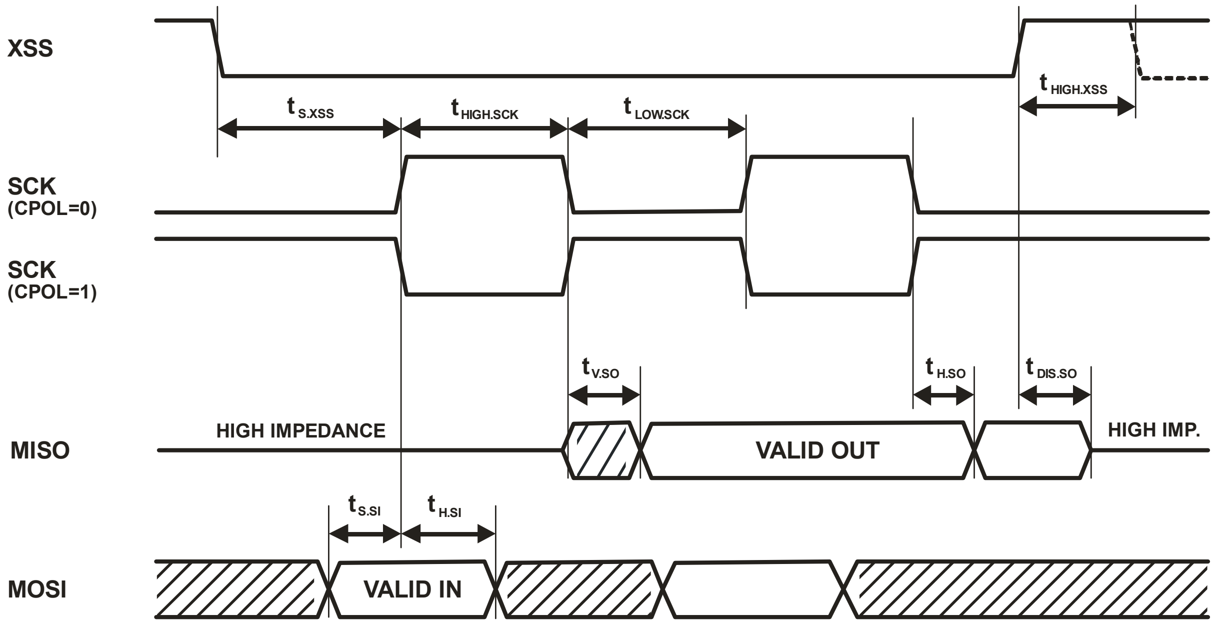



Figure 10-20: Timing Diagram SPI Interface Mode (CPHA='0')

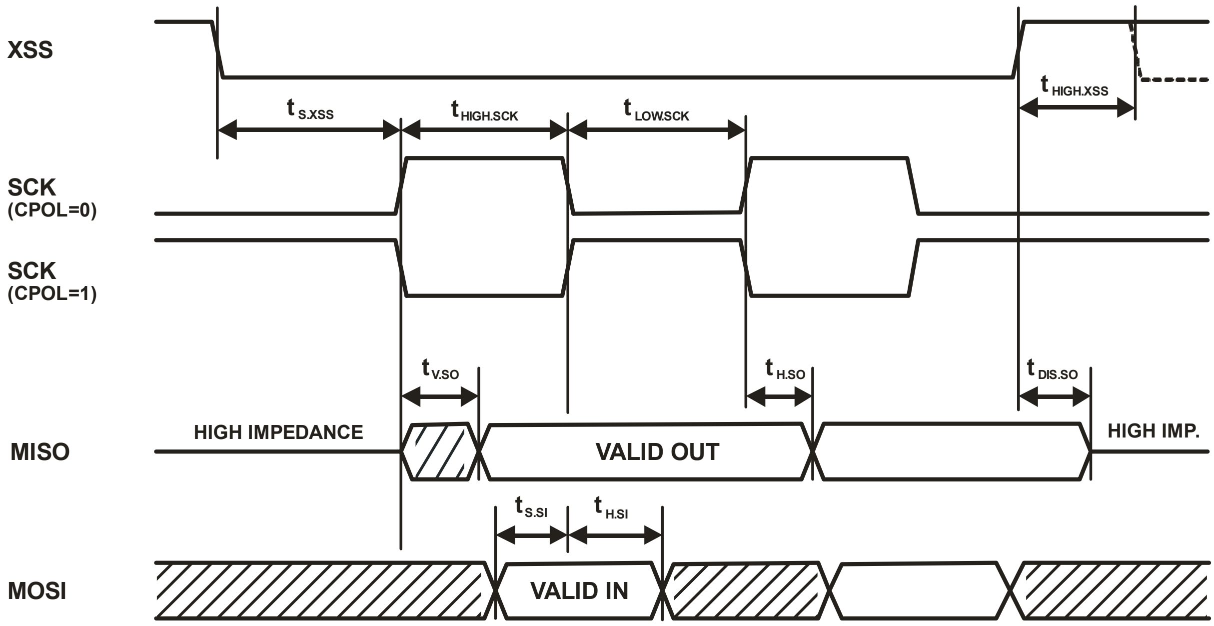



Figure 10-21: Timing Diagram SPI Interface Mode (CPHA='1')

| 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**

## 10.6.7 Timing in I2C Interface Mode

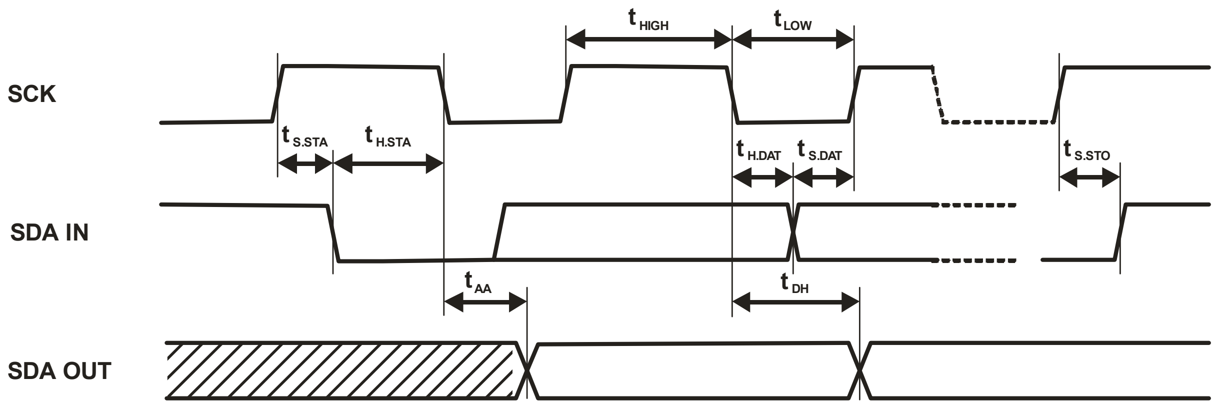



**Figure 10-23: Timing Diagram I2C Interface Mode**

| 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**

# 10.7 Package Specifications

## 10.7.1 LFBGA48

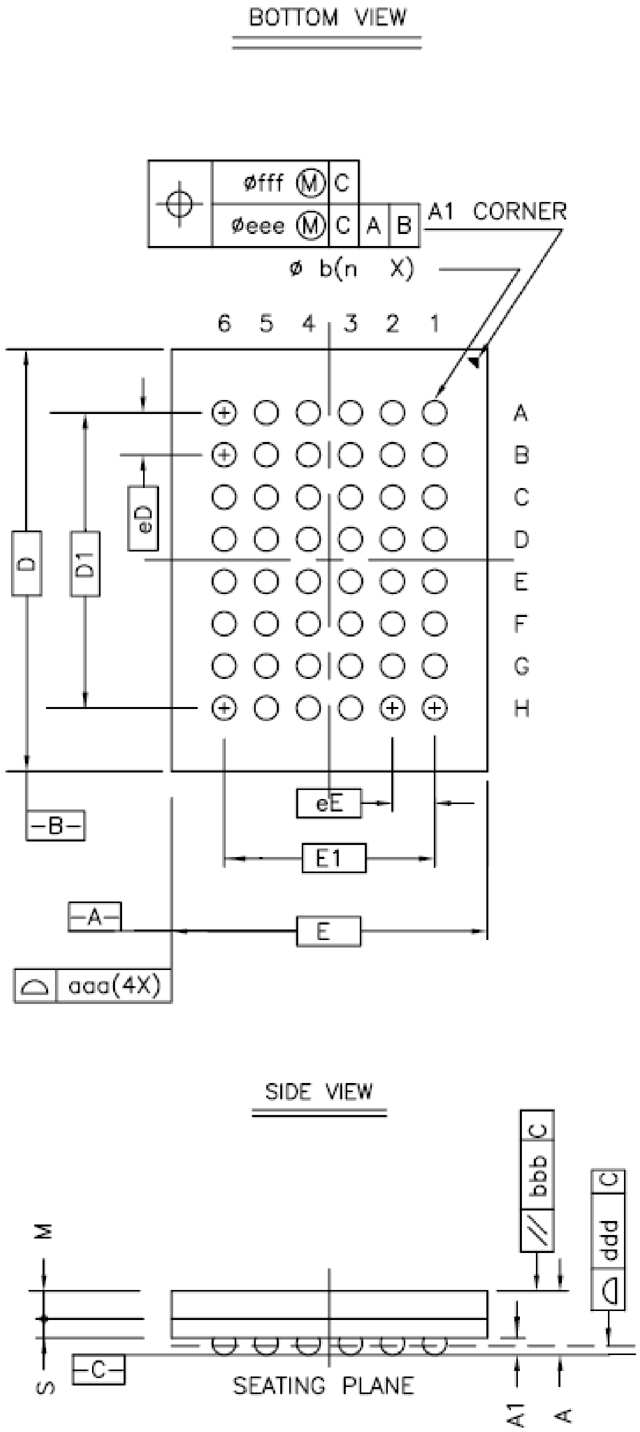
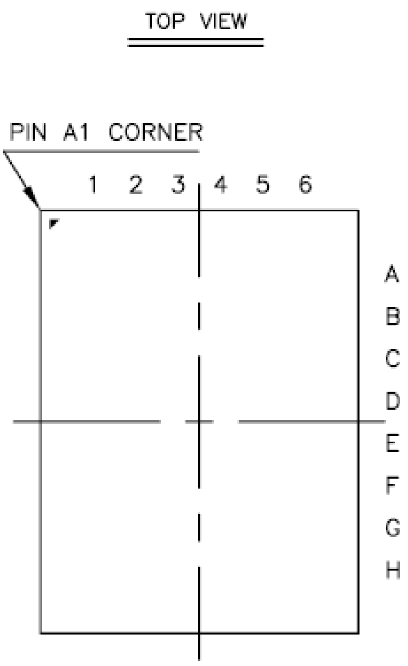



**Figure 10-25: LFBGA48 Package Drawing**




| | | Symbol | Common Dimensions |
|---|---|---|---|
| Package : | | | LF BGA |
| Body Size: | X | E | 6.000 |
| | Y | D | 8.000 |
| Ball Pitch : | X | eE | 0.800 |
| | Y | eD | 0.800 |
| Total Thickness : | | A | 1.400 MAX. |
| Mold Thickness : | | M | 0.530 Ref. |
| Substrate Thickness : | | S | 0.360 Ref. |
| Ball Diameter : | | | 0.450 |
| Stand Off : | | A1 | 0.250~0.400 |
| Width : | | b | 0.400~0.500 |
| Package Edge Tolerance : | | aaa | 0.150 |
| Mold Flatness : | | bbb | 0.200 |
| Coplanarity: | | ddd | 0.150 |
| Ball Offset (Package) : | | eee | 0.150 |
| Ball Offset (Ball) : | | fff | 0.080 |
| Ball Count : | | n | 48 |
| Edge Ball Center to Center : | X | E1 | 4.000 |
| | Y | D1 | 5.600 |

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

### 10.7.2 LQFP48

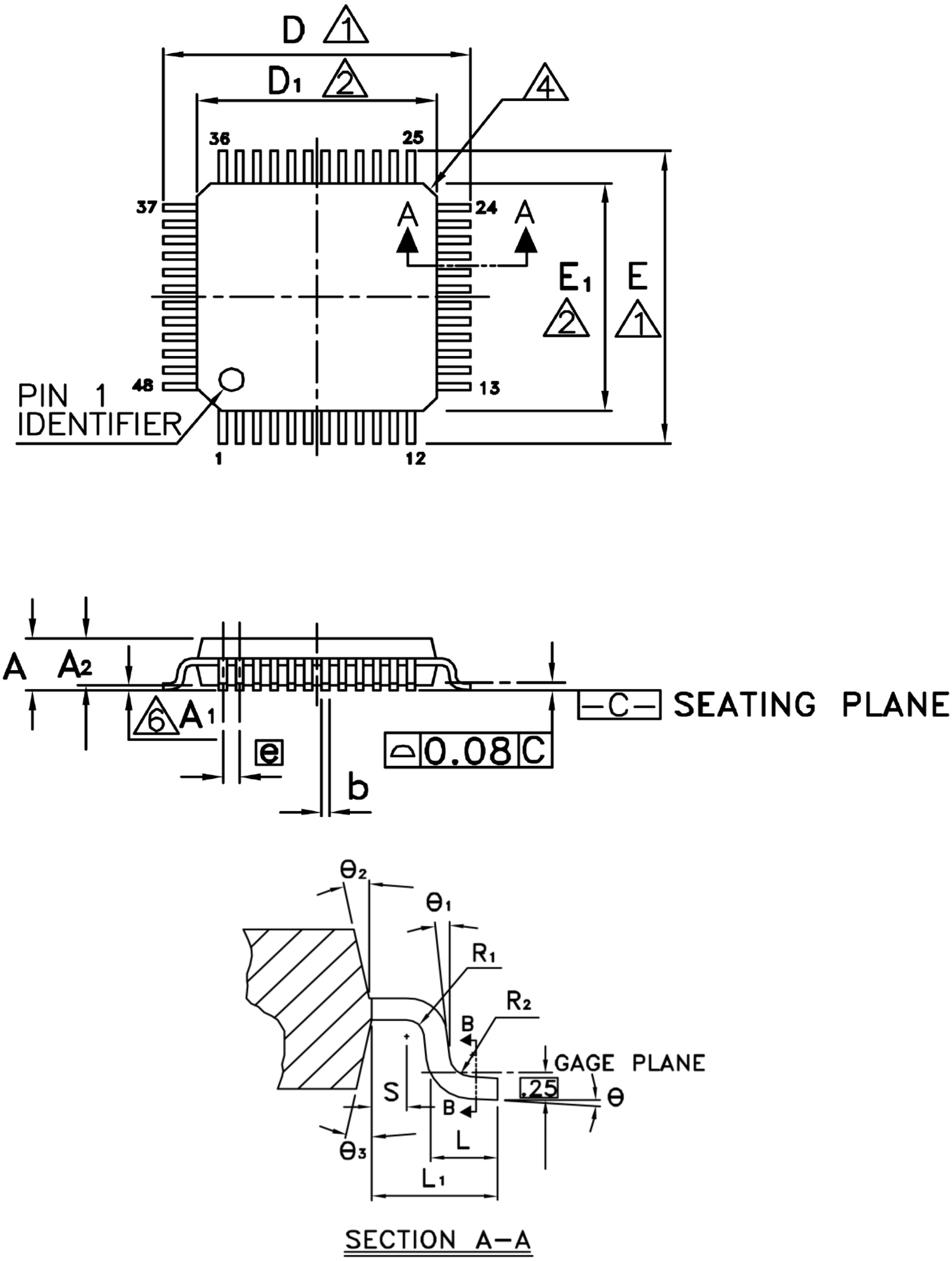



**Figure 10-27: LQFP48 Package Drawing**

| Symbol | Dimensions in mm | | |
|---|---|---|---|
| | MIN | NOM | MAX |
| A | | | 1.60 |
| $A_1$ | 0.05 | | 0.15 |
| $A_2$ | 1.35 | 1.40 | 1.45 |
| b | 0.17 | 0.22 | 0.27 |
| $b_1$ | 0.17 | 0.20 | 0.23 |
| c | 0.09 | | 0.20 |
| $c_1$ | 0.09 | | 0.16 |
| D | 9.00 BSC | | |
| $D_1$ | 7.00 BSC | | |
| E | 9.00 BSC | | |
| $E_1$ | 7.00 BSC | | |
| e | 0.50 BSC | | |
| L | 0.45 | 0.60 | 0.75 |
| $L_1$ | 1.00 REF | | |
| $R_1$ | 0.08 | | |
| $R_2$ | 0.08 | | 0.20 |
| S | 0.20 | | |
| Θ | 0° | 3.5° | 7° |
| $Θ_1$ | 0° | | |
| $Θ_2$ | 12° TYP | | |
| $Θ_3$ | 12° TYP | | |

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

## 10.8 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.**

## 10.9 Ordering Information

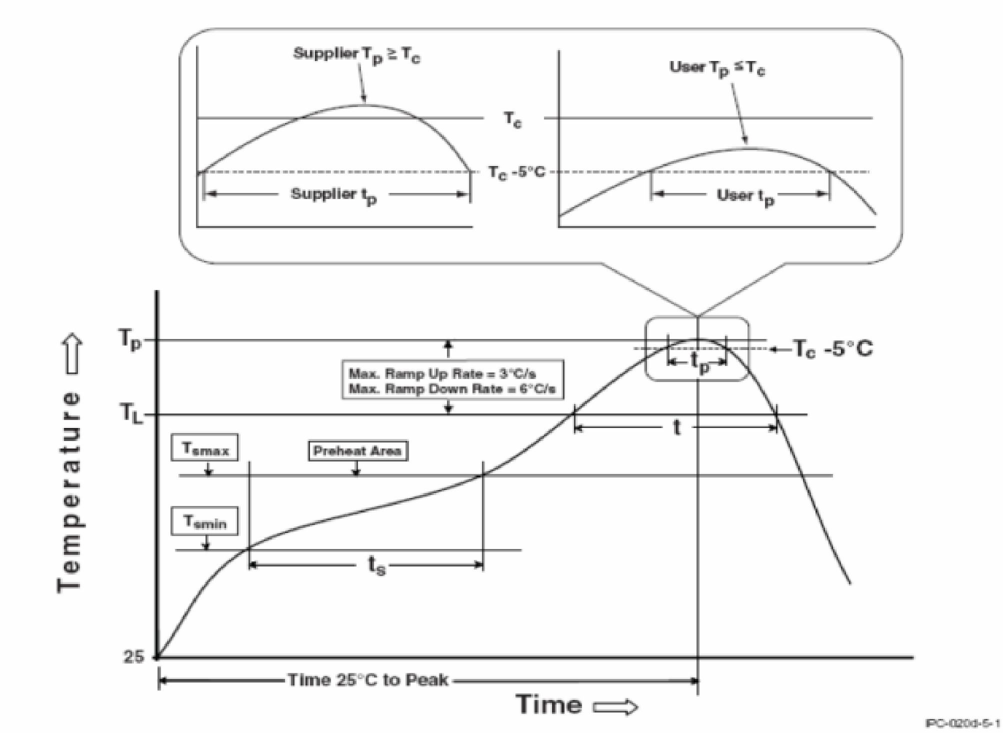
| Version / Part Number | Order Code | Package | Temperature Range | Notes |
|---|---|---|---|---|
| VPC3+S-BGA48 | PALF2009 | LFBGA48 | Industrial (-40°C to +85°C) | |
| VPC3+S-QFP48 | PALF2012 | LQFP48 | Industrial (-40°C to +85°C) | |

**Table 10-1: Ordering information**

## 10.10 Reflow Soldering Profile

Green Package Reflow Profile based on IPC/JEDEC J-STD-022D

Products: VPC3+S-BGA48 (PALF2009)
VPC3+S-QFP48 (PALF2012)




| Profile Feature | Pb-Free Assembly (260°C) |
|---|---|
| Preheat and Soak<br>Temperature min ($T_{smin}$)<br>Temperature max ($T_{smax}$)<br>Time ($T_{smin}$ to $T_{smax}$) ($t_s$) | 150 °C<br>200 °C<br>60-120 seconds |
| Average ramp-up rate<br>Time ($T_{smax}$ to $T_p$) | 3 °C/second max. |
| Liquid temperature ($T_L$)<br>Time at liquid ($t_L$) | 217 °C<br>60-150 seconds |
| Peak package body temperature ($T_p$) | 260°C |
| Time ($t_p$) ** within 5 °C of the specified classification temperature ($T_c$) | 30** seconds |
| Average ramp-down rate ($T_p$ to $T_{smax}$) | 6 °C/second max. |
| Time 25 °C to peak temperature | 8 minutes max. |
| * Tolerance for peak profile temperature ($T_p$) is defined as a supplier minimum and a user maximum.<br>** Tolerance for time at peak profile temperature ($t_p$) is defined as a supplier minimum and a user maximum. | |

**Table 10-2: Reflow soldering profile**

# 11 Revision History

| Version | Date | Remarks |
|---|---|---|
| V1.00 | 10.08.2009 | First release |
| V1.01 | 25.05.2010 | Description of GC_Int_Mode_Ext in Mode Register 3 corrected<br>Some hints for configuration of PLL added<br>Name of I2C clock changed from SCL to SCK<br>Current consumption and thermal resistance added |
| V1.02 | 28.05.2010 | SPI instruction “WRITE ARRAY” added to Figure 8-6<br>Instruction coding in Figure 8-10 “WRITE ARRAY Sequence” corrected |
| V1.03 | 10.10.2012 | Timing table for SPI interface mode corrected (MIN/MAX values swapped)<br>Timing table for I2C interface mode corrected (MIN/MAX values swapped) |
| V1.04 | 18.02.2014 | Pin assignment of LQFP48 package version added<br>Thermal resistance of LQFP48 package added<br>LQFP48 package drawing added<br>Processing instructions revised and ordering information added |
| V1.05 | 18.07.2014 | AB11 (pin 3) added to pinout figure and pin assignment table<br>Part number added to ordering information |
| V1.06 | 30.03.2015 | Notes regarding external pull-up on TXD added<br>Modification of signal names<br>Modification of figure 8-23, 8-24, 8-25, 8-26, 8-27, 9-2 |
| V1.07 | 01.04.2009 | Modification of pin description SUB-D connector |
| V1.08 | 06.12.2019 | new document format<br>Reflow soldering profile added |
| V1.09 | 20.08.2025 | Add information of I/O cell |

**Table 11-1: Revision History**

YASKAWA Europe GmbH
Philipp-Reis-Str. 6
65795 Hattersheim
Germany
Phone: +49 (0)6196 569-300
E-Mail: support.profichip@yaskawa.eu.com
http://www.profichip.com