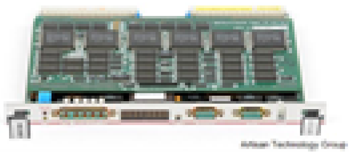
VMETRO VBT-325C

VMEbus Analyzer



User's Manual

VBT-325
Including VBAT-PB

Rev. 1.4 - Valid for Firmware Version 4.99.120

VBT-325B	VMEbus Analyzer
VBT-325C	VMEbus & VSB/SCSI/P2 Analyzer
VBAT-PB	VMEbus Anomaly Trigger Piggyback module

Related documents:

S/TIM200-PB 200MHz Timing Analyzer and Stimuli / Pattern Generator Piggyback module

Warranty

VMETRO products are warranted against defective materials and workmanship within the warranty period of 1 (one) year from date of invoice. Within the warranty period, VMETRO will, free of charge, repair or replace any defective unit covered by this warranty, shipping prepaid. A Return Authorization Code should be obtained from VMETRO prior to return of any defective product. With any returned product, a written description of the nature of malfunction should be enclosed.

This warranty assumes normal use. Products subjected to unreasonably rough handling, negligence, abnormal voltages, abrasion, unauthorized parts replacements and repairs, or theft are not covered by this warranty and will be repaired for time and material charges in effect at the time of repair.

VMETRO's warranty is limited to the repair or replacement policy described above and neither VMETRO nor its agent shall be responsible for consequential or special damages related to the use of their products.

USA:

VMETRO, Inc.
16010 Barker's Point Ln, #575.
Houston, TX 77079, U.S.A.
Tel.: (713) 584-0728
Fax: (713) 584-9034

Europe, Asia:

VMETRO A/S
Nedre Rommen 5E
N-0988 OSLO, Norway
Tel.: +47 22 10 60 90
Fax: +47 22 10 62 02

Preface

Notes - Firmware version 4.99.120

This version of the User's Manual is written for the firmware version 4.99.120. VMETRO is continuously updating the firmware for the VBT-325 and piggybacks, implementing new features and improvements.

The main features to be provided in upcoming firmware releases are:

- Timing markers in waveform diagrams.
- Ability to show trace windows from VME and/or P2 and/or TIM200 at the same time.
- Ability to dump/load "Setups" to/from files on a PC/Host.
- User-defined P2 support, allowing customized definition of signal names, allocation and groups etc. for the P2 bus.

Note: *A preliminary version of User-defined P2 is available upon request. Please contact VMETRO for more information.*

These features and more will come in subsequent firmware releases which will be sent to all VBT-325 customers un-requested and free of charge.

The Bus Analyzer concept

A Bus Analyzer is a pre-configured logic analyzer designed as a plug-in card for a specific bus, conforming to the electrical and mechanical specification of the target bus. The primary use of a Bus Analyzer is to monitor the activity on a backplane bus and provide a trace of bus cycles between modules on the bus, presenting this as alphanumeric trace lists or as waveforms on a standard ASCII terminal. This is done without the need for connecting and configuring large numbers of probes to the backplane, a time-consuming and error-prone process necessary with general-purpose logic analyzers. Statistics analysis in bus systems is also an important application of bus analyzers.

A basic idea behind bus analyzers is that the analyzer is "hard-wired" to capture the protocol of the target bus, thereby reducing the need for the user to understand all the details of the bus protocol in order to perform meaningful analysis of activity in the target system. This offers the user maximum productivity and convenience during development, debugging, testing and verification of bus based computer systems.

VMETRO is a company totally committed to building the finest Bus Analyzers, and is recognized in development labs around the world as providing superior tools for developers and manufacturers of bus based computer equipment. With the VBT-325, VMETRO offers the fourth generation, state-of-the art product based on 10 years of experience in building bus analyzers.

VMETRO

The Bus Analyzer Specialist

TABLE OF CONTENTS

TABLE OF CONTENTS v

1 VBT-325 PRODUCT OVERVIEW 1

Main blocks 1

Model B and C 2

Applications 4

Specification Highlights 4

Piggyback modules 5
Piggyback Carrier (VPC-MkII) 6
Piggyback User's Manuals 6
VBAT-PB 6
S/TIM200-PB 6
VDRIVE-PB 6

Accessories 7

2 INSTALLATION 9

Static electricity - Precautions 9

Preparations 10
Inspection 10
Jumper settings 10
P2 rows A/C connections 11
TTL/CMOS Input only (0-5V) 11
Isolation of P2 rows a/c 11

Slot Selection 12
VMEbus 12
VSB 12
Daisy-Chains 12
Slot 1 12
User-defined P2 13

Power supply 14
Air Cooling 14
5V readout 14
+/-12V 15
External Power 15

Terminal Connection 17
Transparent mode 19
Terminal-Host RS232 handshake 19

Power-On 20

Automatic Baud Rate detection....20

Start-up menu....21
Select Terminal type (T)....21
PC as terminal....22
<CR> = Continue....22
Clear Non-volatile memory....22

3 FUNCTIONAL DESCRIPTION....23

Main blocks....23
Sampling stage....24
State (Synchronous) sampling....25
Timing (Asynchronous) sampling....25
Word Recognition / Triggering stage....25
Sequencer....27
Sample Storage stage....27
Trace Buffer....27
Trigger position....27
Statistics Counters....28

Sampling methods....29
VMEbus state sampling....29
Bus Request latching....29
Bus Grant latching - BgL....29
Data cycle sampling....30
IRQ -> IACK....30
RMW, Block....30
VME64....31
SSBLT....31
VSB state sampling....31
SCSI state sampling....32
SCSI Pinout....33
SCSI-2 FAST....33
VXI sampling....33
User-defined P2 state sampling....33

External inputs....34
In3:0 in VME part....34
In3:0 to P2 part using cross trigger....34
Shared In3:0 and Time Tag bits....34
Shared In1 and Temperature Probe....34

4 OPERATION....35

User-interface structure....35
Targets....35
Screen categories....36
Setup screen....37
Trace Display screen....37
Statistics screen....37

Screen and Command elements 37
Pull down menus 37
Sub-command selection 38
Accelerator keys 38
Dialog Boxes 39
Function keys 40
Numeric keypad 41
Keyboard template 41
Multiple windows 41
Message line 42
Block Cursor recommended 42
Refresh Screen 42

Event Patterns 42
Control signals 43
Signal groups with mnemonics 43
NOT operator 43
Individual signals 44
Signal polarity 44
Edit event patterns 45
Delete fields/event patterns 45
Insert signal fields 45
Insert event patterns 46
Horizontal scrolling 46
BR* and IRQ* format 47
Address/Data attributes 47
Binary details 47
Not 47
64-bit 47
Range 48

Sequencer 48
General structure of one state 49
Single event and Sequencer modes 50
Single Event mode 50
Sequencer mode 50
Event Expression 50
Sequencer Notation 51
UPPER/lower case 51
Use of brackets 51
State and Line numbers 51
Indents 51
Input indicators 51
Change event 52
Modify the sequencer program 53
Operators 53
Sampling 53
Store 54
If/Elsif/Else 54

Goto 55
Count 55
Delay 55
Trigger 56
Halt 56
Implicit actions and transitions 56
Loose and Tight sequence 57
Sequencer examples 58
Count, Delay and Switch sampling mode 58
VSB sequencer examples 59

Cross-triggering 62

Trace Display 63
Alphanumeric trace list 63
Signal selection 64
Horizontal scrolling 64
Waveforms 64
Hex/Binary value 65
Signal selection 65
Zooming (Time/Div) 65
Add window 66
Trace Dump to PC/Host 66
Simulator used for trace review 66

Statistics 66
Statistics screen 67
Display features 68
Time History Diagram 69
Counters Operation 70
Update rate 71
Accumulate/Reset mode 71
Idle interval 72
Event Counting / Bus Level Histogram 72
Bus Utilization Histogram 73
VMEbus Utilization measurements 74
VSB Utilization measurements 75
Bus Transfer Rate Histogram 76

5 EXAMPLE OF USE 77

Setting trigger 78
Event Patterns 78
Edit a signal field 78
Single Event Mode 79
The Sequencer 79
Edit the Sequencer 79

Start Sampling 79
To take a VMEbus trace 79
To trace another bus 80

Trace Multiple busses 80

Trace Display Screen 80
Time Tags 80
Jump 81
Search 81
Add a trace window 81

Timing sampling 81

Statistics 81

Piggybacks 82

6 COMMANDS REFERENCE 83

Main menu - Setup screen 83
Trace 83
- Run VME 83
- Run Multiple 83
- Sampling Status 83
- Halt VME 84
- Show 85
Edit 85
- Event Patterns 85
- Sequencer 85
- Trigger position 86
- Sampling mode 86
- Options / State Sampling options 86
Target 87
- VME 87
- P2 bus 88
- TIM200 bus 88
- STIM200 bus 88
- Reconfigure 88
Statistics 88
Setups 88
- Initialize 89
- Store 89
- Delete 89
Utilities 89
- Transparent Mode 89
- Transparent Mode Options 89
- Trigger Output Options 90
- LED Display 90
[X] Voltage readout 90
[X] Temperature readout 91
- Serial Ports 92
- Selftest 92
- Reset Analyzer 92
- Specials 92

Help....92

Trace Display menu....93
Trace....93
- Dump to PC....93
- Load from PC....93
- Print....93
Search....94
- Find....94
- Extract Mode....95
- Close Search Window....95
Jump....95
- First Line....95
- Last Line....95
- Trigger Line....95
- Line number....95
Count....96
Format....96
- Time/Div....96
- Absolute/Relative Time tags....96
- Decoding and Formatting....96
- ASCII Decoding....96
- Trace signals....96
- Global options....97
Window....97
- New....97
- Close....97
Quit....97
Help....97

Statistics menu....98
Session....98
- Run....98
- Continue....98
- Halt....98
Function....98
- Event Counting....98
- Bus Utilization *(VME and VSB)*....99
- Bus Transfer Rate *(VMEbus only)*....99
Options....99
- Standard Histograms vs. Time History Curves....99
- Bar Markers....99
- Graph Display Options....100
- Maximum Scale....100
- Count Options....100
- Select Events....101
Target....102
- VME....102
- P2 bus....102

Utilities 102
Quit 102
Help 102

7 SIGNAL REFERENCE 103

VMEbus Signal Groups 103
VMEbus Signals 104
Other signals - Target VME 104

VSB Signal Groups 105
VSB Signals 105
Other signals - Target VSB 106

8 TRACE EXAMPLES 107

VMEbus 107

VSB 108

SCSI 109

9 VBAT-PB VME BUS ANOMALY TRIGGER 111

VBAT-PB Product Overview 111
Finds incompatibilities 111
Example of violation found 112
Mask violations 112

Complements the VBT-325 113

Related documents 113

VBAT-PB Features 114

VBAT-PB Specifications 115
Violations screened for 115
Non violations indicated 116

Trigger characteristics 116

Power requirements 117

Possibilities and limitations 117
What the VBAT can do 117
What the VBAT cannot do 117

Arbiter location 118
Onboard Arbiter 118

VBAT-PB Installation 119

VBAT-PB Operation 120
VBAT-to-VBT trigger signal 120
VBAT command screen 121
The Violations command 121
Enable 121

Disable 122
Clear 122
The Mask command 122
Select 122
Select All 122
Select None 122
The Modes command 122
Manual and Automatic clear 122
Lock on first 123
Accumulate 123
Black Cockpit 123
High intensity and Inverse video 124
Explain violation 124
Violation descriptions 125
VBAT-PB Switch settings 130
Arbiter Enable/Disable 130
Trigger Output Signals 130
Front panel Trig output 130
ERROR output on VBAT 130
Individual trigger outputs 130
10 VBT-325 FIRMWARE UPGRADE PROCEDURE 132
Diskette for PCs 132
Boot PROM 132
RS232 connection 132
Load firmware from PC 133
Specify PCB and ECO Level 133
Setups lost 134
Procedure if Boot PROM is missing or wrong version 134
Run from Boot PROM 136
11 JUMPER SETTINGS 137
VME & VSB (Twin mode) 139
VSB Bus Grant Daisy chain jumper 139
VME & SCSI (Twin mode) 141
VME & User-defined P2 (Twin mode) 143
Futurebus+ (Wide mode) 143
12 SCSI CONNECTION ON P2 145
VBT-325 SCSI P2 adapter 146
SCSI Connection on P2 147

13 CUSTOMIZED ISOLATION OF P2A/C 149

14 VMETRO VT100 EMULATOR - VT100.EXE 151

Usable "terminal types" 152

Built-in script language 152
Script control commands 153
Function keys in script files 154

15 DUMP/LOAD OF TRACE TO/FROM PC 155

Dump to PC/Host Command 155
Comments 156
From line 156
To line 156
Packing 156

Load From PC/Host Command 157
Insert from line 157
Initialize non-loaded trace lines 157

Dump/Load Trace using the WINDOWS Terminal Emulator 158

Dump Trace using VMETRO VT100 Emulator 158

Load Trace using VMETRO VT100 Emulator 158

Dump/Load with PROCOMM-PLUS 159

Trace File Format 160

16 SIMULATOR FOR PC 163

Demonstration / training 163

Reviewing trace files 163

Creating patterns 164

Simulate bus activity 164

Trace Files 164

INDEX 167

1 VBT-325 PRODUCT OVERVIEW

The VBT-325 "VME+ Analyzer" is a bus analyzer for VME and VSB, SCSI or other P2 busses, e.g. the TTL level signals on P2a/c on VXI (when used with an VXE-35C adapter). The board contains two separate independent analyzers, one 128-bit wide analyzer pre-configured for the VMEbus, and one 64-bit wide analyzer that can be configured either for VSB, SCSI or other busses on the P2 connector of VME boards. Both analyzers on the board have individual sampling logic, word recognizers, trace memories and triggering circuitry. An onboard 68EC020 microprocessor with 512KByte of Flash EPROMs controls the hardware and runs the user-interface, which is operated from an ASCII terminal, PC or workstation through a RS232 port.

Main blocks

Each analyzer part of the VBT-325 consists of three main stages, through which samples are passing during the acquisition process:

- Sampling stage
- Word Recognition / Triggering stage
- Sample Storage / Statistics Counting stage

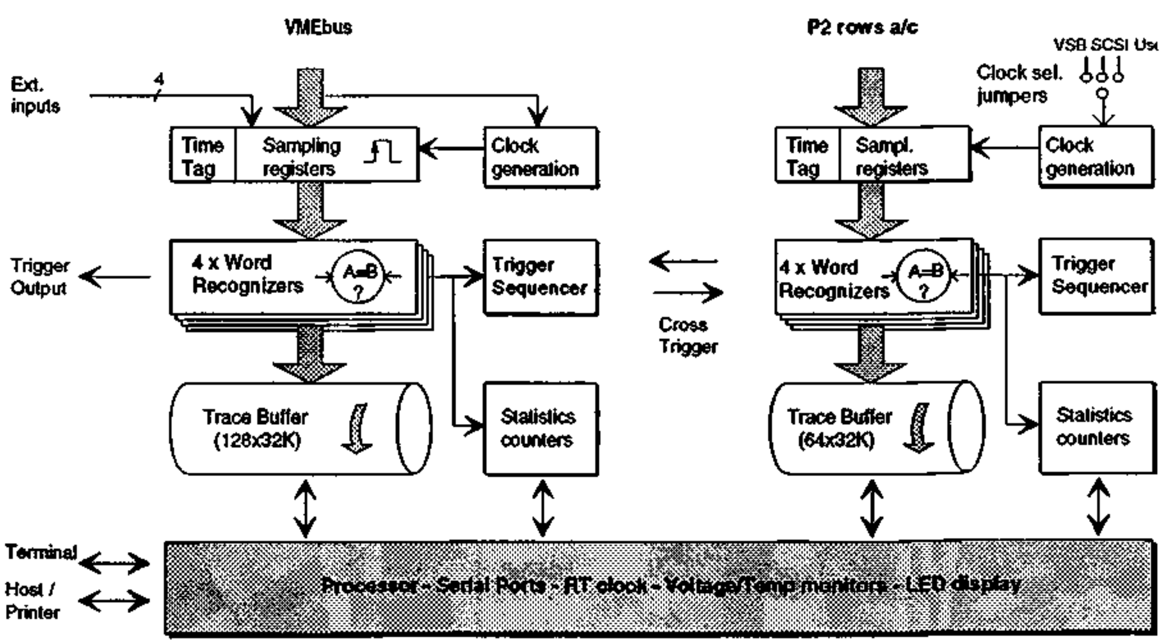


Figure 1. VBT-325 Block Diagram (shown in "Twin mode")

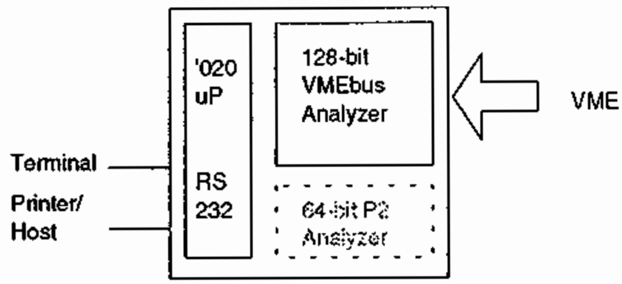
As can be seen from the block diagram in figure 1, the VBT-325 contains substantial amount of hardware functionality. This is achieved through six advanced ASICs designed and developed by VMETRO called the Bus Tracer Chip (BTC). These devices implement all the sample acquisition, recognition and storage capabilities of the board, as well as numerous counters for statistics and time measurements. This gives the VBT-325 remarkable performance and functionality, like sampling rate up to 50MHz in state or timing mode, advanced triggers with *Not* and *Range* capabilities, as well as store filters and occurrence and delay counters.

The chapter FUNCTIONAL DESCRIPTION later in this manual gives more details of the functions of the different blocks in the product.

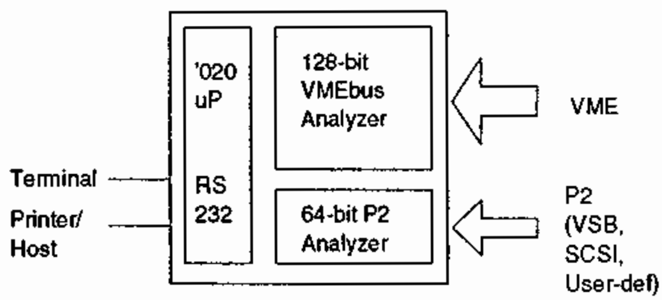
Model B and C

The VBT-325 is available in two models, VBT-325B and VBT-325C. Model VBT-325C is the full-featured version that supports both VMEbus and VSB/SCSI/P2, with upgradeability to other busses like VXI and Futurebus+. The VBT-325B is a lower-cost version, where only the VMEbus is supported, but otherwise with the same performance as the VBT-325C. The VBT-325B can easily be upgraded, in the field, to a VBT-325C by means of a firmware/PLD upgrade.

VBT-325B



VBT-325C in Twin Mode



VBT-325C in Wide Mode

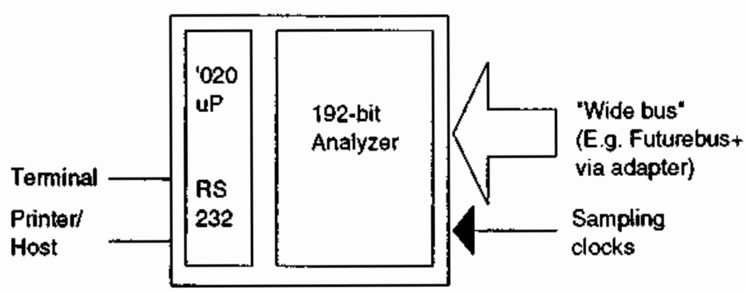


Figure 2. VBT-325 models and configurations.

Applications

Applications of the VBT-325 include hardware and software debugging and testing, system tuning, and performance analysis. Other applications are repair and field service of a number of different bus architectures, primarily VMEbus, with or without subsystem busses like VSB, SCSI, user defined P2 bus and VXI or Futurebus+ with the use of specially designed adapters.

Working with the product involves utilizing one of three basic analyzing capabilities:

- State analysis (capturing bus cycles synchronously one by one).
- Timing analysis (capturing bus cycles at a fixed sampling rate asynchronously to the bus traffic).
- Statistical analysis (providing histograms of various bus activity).

Specification Highlights

VBT-325C:

- 32K Trace memory, separate for VME and P2.
- 101 ch. VMEbus Analysis, plus 4 ext. inputs on mini-coax
- 64 ch. VSB/SCSI/P2 Analysis with separate trigger sequencer.
- Simultaneous VME and VSB/SCSI/P2 analysis with cross-triggering and integrated user-interface.
- 50MHz Timing Analysis on VME and VSB/SCSI/P2.
- State Analysis up to 25MHz on VME and VSB/SCSI/P2,

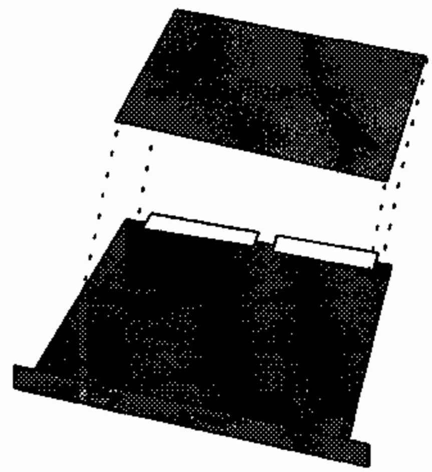
  up to 50MHz on Futurebus+ with full-speed trigger.
- VMEbus rev.D compatible, incl. VME64 and SSBLT sampling.
- Voltage, Temperature monitoring and Time-of-Day clock.

VBT-325B:

- As VBT-325C, but without P2 support. Field upgradeable to VBT-325C by firmware/PLD replacement.

Piggyback modules

The VBT-325B/C is equipped with connectors that allow it to carry piggyback modules for added functionality or performance. Below is a short presentation of the piggyback modules currently available for the VBT-325.



TIM200-PB The TIM200-PB is a 200MHz Timing Analyzer piggyback module for the VBT-325 for high-speed analysis of the VMEbus *or* P2 bus. The TIM200-PB has a 32K trace buffer and samples up to 107 signals with 5ns resolution, and offers full-speed trigger on any bit or bit combination, including cross-trigger from the VBT-325. The trigger pattern can be qualified with a "duration filter", to specify valid pattern as greater than or less than in the range 5-635ns. Signals sampled are presented as graphical waveforms with zoom, cursors and timing markers. The TIM200-PB can be upgraded to a STIM200-PB, see below.

STIM200-PB The STIM200-PB is a 200MHz stimuli/pattern generator piggyback module for the VBT-325. It is essentially a TIM200-PB with firmware that permits its trace memory to be put in reverse. The STIM200-PB can generate bus cycles on VME or VSB and has user-defined timing with 5ns edge-to-edge resolution, as well as, true bus grant and slave handshake. By means of a screen-oriented pattern editor the user may create any type of cycles and signal sequences. Cycle templates are also provided that include all VMEbus cycles, including VME64 and SSBLT 64-bit block cycles.

VDRIVE-PB The VDRIVE-PB is a piggyback module that implements a true VMEbus Master/Slave and System Controller by means of the industry standard VIC068 chip. From the same user-interface as that of the VBT-325, the user can then generate any cycle type, perform memory tests, and generate interrupts, IACKs etc. The slave memory can also be set at user-defined limits, and there is a programmable DTACK* generator that can give DTACK* at any address with a user-defined delay.

VBAT-PB The VBAT-PB is a piggyback module that automatically monitors all VMEbus traffic, screening the bus for violations of the VME specification. The board's rule-based parallel trigger elements continuously, and simultaneously, detect bus timing violations like address not stable while AS* asserted, bus granted to two masters, etc. Violations are directed to the trigger circuitry and trace memory of the VBT-325, and the rule violations are explained in plain English.

Piggyback Carrier (VPC-MkII)

In many applications it is desirable to use more than one piggyback module at the same time. For this purpose, VMETRO offers a "Piggyback Carrier", part name VPC-MkII. This is essentially a VBT-325 board stripped for all the analyzer features, containing only the processor, serial ports and the piggyback contacts. A trigger in/output connector is also present, allowing cross-triggering to/from the VBT-325 or another instrument like an oscilloscope etc.

A typical example where the VPC-MkII is used, is a 2-slot "State-Timing-Anomaly" analyzer solution, illustrated below. This includes both a TIM200-PB and a VBAT-PB, where the 200MHz Timing Analyzer of the TIM200-PB is used for examination of bus violations found by the VBAT-PB.

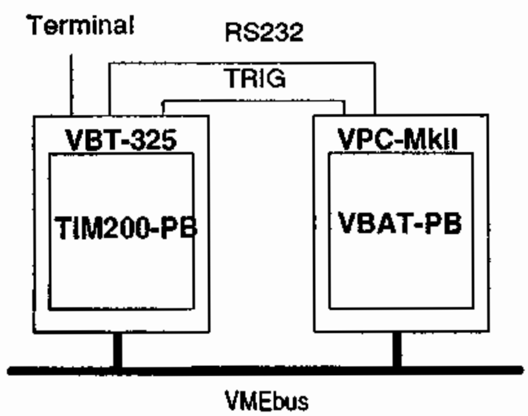


Figure 3. The VPC-MkII Piggyback Carrier allows more than one piggyback module to be used simultaneously, like in this 2-slot "State-Timing-Anomaly" analyzer solution.

Piggyback User's Manuals

VBAT-PB

The VBAT-PB is described a separate section in this manual.

S/TIM200-PB

The TIM200-PB 200MHz Timing Analyzer and STIM200-PB Stimuli / Pattern Generator (the common term S/TIM200 is used to describe both) is described in a separate User's Manual volume.

VDRIVE-PB

The VDRIVE-PB is for the moment only described in the "VDRIVE-PB User's Manual for FW 3.31" for use with the VBT-321B. However, the commands supporting the VDRIVE-PB are more or less the same as for the VBT-321B with VDRIVE-PB. For the VBT-325, all the commands are readily available as pull-down menus under 'VDRIVE' in the main command menu bar, for intuitive operation. For understanding of how the product works in general (in particular **Daisy-chain considerations**), please refer to the manual as mentioned above. (A User's Manual for the VDRIVE-PB when used on the VBT-325 will be available 2Q-94).

VDRIVE-PB Installation

The VDRIVE piggyback should be installed with the same alignment as the VBAT-PB (although, it fills the entire area), please refer to the ch. "VBAT-PB VME BUS ANOMALY TRIGGER, VBAT Installation" in the back of this manual.

Accessories

VMETRO offers a complete set of cable accessories that will help the user to take full advantage of the VBT-325. For connection to a terminal, PC or workstation, various RS232 cables are available. A special cable is designed for External Power Supply, and a temperature probe is available. A five-way BNC-to-MiniCoax transition cable assembly allows convenient connection of four external input signals and one Trigger Output. Also single BNC-to-Mini-coax cables are available, one is delivered standard with the VBT-325 (Part #401-325-IOBN1).

Part Number	Description	
401-TER-232	Terminal Cable (RS232 DB9M-DB25F), 3m/9ft	
401-TM-232	Transparent-Mode Cable (RS232 DB9M-DB9M), 0.5m/1.5ft	
401-PC-232	PC Cable (RS232 DB9M-DB9F X), 3m/9ft	
401-325-EPC	External Power Cable, 1m/3ft	
401-325-IOBN5	5-way Trigger In/Out BNC Coax Cable Assembly, 1m/3ft	
401-325-ETS-1	Temperature Sensor, 1m/3ft	
401-325-ALL	Cable Package, includes all of the above.	
401-325-IOBN1	Single Ext.Input/Trigger output BNC Coax Cable.	1)
401-325-STA	VBT-325 to VPC-MkII Trigger Cable.	2)
401-SCSI	SCSI P2 Cable w/five 50-pin connectors, incl. one D-Type Cannon, and three P2 64-pin DIN female for P2 connections in VME slots, 4m/12ft.	

1) One included with VBT-325.

2) One included with VPC-MkII.

2 INSTALLATION

This chapter describes the installation and start-up of the VBT - 325. Please read this chapter carefully before you unpack and install the analyzer.

Static electricity - Precautions

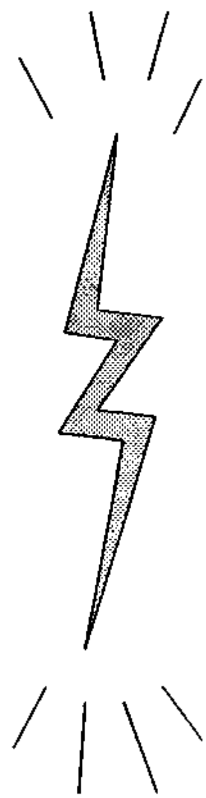
Before unpacking the VBT-325 from its shipping container, make sure that this takes place in an environment with controlled static electricity. The following recommendations should be followed:

- Make sure your body is discharged to the static voltage level on the floor, table and system chassis by wearing a conductive wrist-chain connected to a common reference point.

- If a conductive wrist-chain is not available, touch the surface where the board is to be put (like table, chassis etc.) before unpacking the board.

- Leave the board only on surfaces with controlled static characteristics, i.e. specially designed anti-static table covers.

- If handing the board over to another person, touch this persons hand, wrist etc. to discharge any static potential.



Important -> Never put the board on top of the conductive plastic bag in which the board is shipped. The external surface of this bag is highly conductive and may cause rapid static discharge causing damage. (The internal surface of the bag is isolating.)

A safe place to leave the board is on the pink coating found inside the shipping container (and of course, *inside* the plastic bag.)

Preparations

Inspection

Make sure that the VBT-325 you have received is according to your purchase order with respect to model. The VBT-325 model B and model C can be distinguished by looking at the label on the big MACH PLD device on the board. You will find either the text VBT-325B or VBT-325C on this label.

Accessories

With the VBT-325 you should find the following accessories:

- A small plastic bag with spare jumpers (for reconfiguration of the board).
- One BNC-to-MiniCoax cable 3ft/1m (for Ext. Input/Trigger Output).
- Diskette with VBT-325 Simulator (for trace review etc. Please refer to the chapter SIMULATOR in the end of this manual).

You should also inspect the board to verify that no mechanical damage appears to have occurred. Please report any discrepancies or damage to your distributor or to VMETRO immediately.

Jumper settings

There are a number of jumpers on the VBT-325 board that define target bus, operating mode (Wide/Twin) and RS232 handshake bypass (see ch. Terminal Connection).

VBT-325C

The factory setting for VBT-325C is twin mode, VMEbus & VSB (except VSB BGIN/OUT* daisy-chain bypass, ref. JUMPER SETTINGS.)

Model	**Target bus**	**Mode (Wide/Twin)**	**Factory setting**
VBT-325B	VME	-	X (Only option)
VBT-325C	VME & VSB	Twin	X
VBT-325C	VME & SCSI	Twin	
VBT-325C	VME & P2	Twin	
VBT-325C	Futurebus+	Wide	

VBT-325B For the VBT-325B, the only possible configuration is VMEbus, so there is never a need to reconfigure jumpers on this model, unless the board is upgraded to a model C or the RS232 handshake bypass between terminal/host is used.

Please refer to the section JUMPER SETTINGS for the correct jumper setting for the other possibilities listed in the table above.

P2 rows A/C connections

No pins on the P2 rows A/C are driven, connected together or to GND on the factory jumper setting. However, this may change after configuring the board for VSB and SCSI. The jumper J64 for VSB BGIN/OUT* (see section JUMPER SETTINGS) will short together P2a31-c32 when installed and may cause problems in systems using other P2 configurations. Similarly, grounding of the SCSI bus takes place with four jumpers in field J51 (see section JUMPER SETTINGS), and these must not be in place when used with other P2 busses.

TTL/CMOS Input only (0-5V)

The input channels on the VBT-325 are designed for TTL/CMOS voltage levels only (0.0-5.0V), and **damage may occur if other voltages are applied.**

max. V_{ih}: 5.25V Absolute maximum tolerated input voltage is 5.25V. If the P2 bus contains signals with voltage levels other than TTL/CMOS, (for example ECL, analogue or special power supply voltages as found in VXI systems), **these must be isolated from the VBT-325.** (For this purpose, VMETRO offers a special adapter, the VXE-35C, which extends the VBT-325C to fit in C-size VXI systems and isolates the non-TTL signals from the backplane. Please refer to separate data sheet.)

Isolation of P2 rows a/c

Isolation of P2 pins with illegal voltage levels can be done in one of the following ways:

1. Place the VBT-325 in a slot without the illegal voltages on the P2 connector.
2. Place the VBT-325 on an extender board which isolates the signals with illegal voltage. (For VXI, use VMETRO's VXE-35C).
3. Customize your VBT-325 by cutting copper tracks that are specially laid out for all P2 signals for this purpose. These can be found as a row of 32 pairs or solder pads connected with a thin copper track on each side of the

board just next by the P2 connector. The pads on the bottom side are signals from P2 row C, and the top side row A. If a connection needs to be re-established, a short piece of un-isolated wire (AWG30 or similar) can be soldered between the two pads in a pair. Please refer to section "CUSTOMIZED ISOLATION OF P2A/C" for detailed instructions.

Slot Selection

VMEbus

The VBT-325 can be installed in any slot in a VME backplane. However, it is recommended that the VBT-325 be located as far to the left (closest to slot #1) as possible. This ensures that the Bus Grants (BG3-0IN/OUT*) and IACKIN/OUT* daisy-chains pass the VBT-325, and these signals are captured by the analyzer. The bus grants are of special interest since the VBT-325 contains special circuitry to sample these signals even in synchronous (state) sampling mode (described in detail in the chapter "FUNCTIONAL DESCRIPTION, Sampling methods").

VSB

Similarly, for analysis of VSB backplanes it is recommended to place the VBT-325C as far to the left as possible on the VSB backplane to ensure that the VSB Bus Grant (BGIN/OUT*) daisy-chain pass the VBT-325C. This is relevant if asynchronous sampling (timing mode) is used to see the bus arbitration process (serial arbitration).

Daisy-Chains

The VBT-325 will bypass the VMEbus daisy-chains BG(3-0)IN/OUT* and IACKIN/OUT* directly. The VSB BGIN/OUT* daisy-chain is not installed in the factory setting. The user may want to install this before using the board in a VSB system, refer to the section "JUMPER SETTINGS, VME & VSB".

Slot 1

When the VDRIVE-PB piggyback module is installed, the VBT-325 may be placed in slot #1 if the VDRIVE-PB is configured as System Controller (Please refer to VDRIVE-PB User's Manual).

Note: *Bus grants received by any Bus Master located to the left of the VBT-325 (lower slot number) will not be visible to the analyzer, due the nature of VME/VSB daisy-chaining. Similarly, IACKIN/OUT* received by any Interrupter located to the left of the VBT-325 (lower slot number) will not be visible to the analyzer.*

User-defined P2

In order to perform bus analysis of a user-defined P2 bus, one must make sure that the slot where the VBT-325C is inserted has access to the user-defined bus on the a- and c- rows of the P2 connector. In many cases, the P2 bus is taken directly with a flat cable from the P2 connector on a VME board to an I/O device, without any form of bussing of the P2 bus along the backplane. In such cases, the user should insert the VBT-325C in a neighbour slot to the board that has the P2 bus, and provide an "L-shaped" extension of the flat cable to the slot of the VBT-325C (see figure below.)

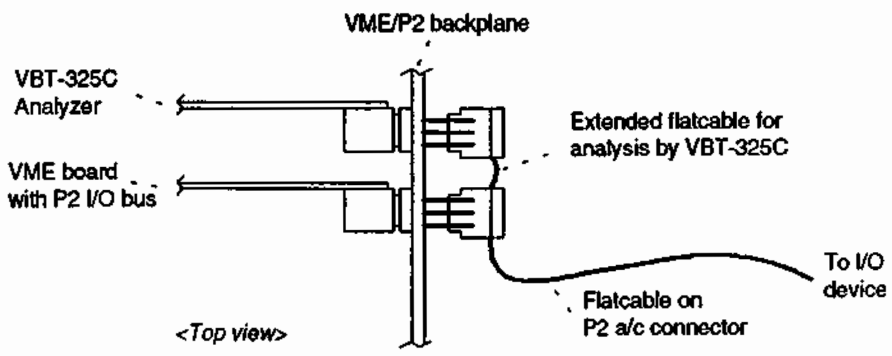


Figure 4.. Extension of flatcable for P2 I/O-bus for analysis by the VBT-325C.

Power supply

The VBT-325 is normally powered directly from the +5V and +/- 12V rails in the VME backplane, and it is important to make sure that the power supply in the card crate has sufficient capacity to supply the VBT-325. Current consumption is depending on operating mode, as is given in the following table:

Power consumption:		
+5V		
Idle	:	2,8A typ.
Sampling	:	3,7A typ. @ 6MHz, VME only
		4,8A typ. @ 50MHz, VME only
		5,4A typ. @ 50MHz, VME and P2
+/-12V	:	10mA typ.
		(+12V/120mA max., when programming
		Flash EPROMs only)

Note: *Make sure that the board is always powered both from P1 and P2.*

Air Cooling

Forced air cooling is necessary to keep the operating temperature of the VBT-325 board at reasonable levels. The power consumption varies depending on the state the board is in, from appr. 14W when idle, to almost the double when sampling all channels at max. speed (see power consumption table above.)

5V readout

The front panel LED display will show the actual 5V voltage supplied to the VBT-325 when user interface is started. A reading of **at least 4.95V** is recommended when the board is idle. This normally gives sufficient margin for the increase in power consumption when the analyzer is running.

Be aware that if the VBT-325 is placed on an **extender board**, there may be excessive **voltage drop** which may need to be compensated by increasing the system voltage slightly.

+/-12V

The VBT-325 board uses +/-12V for powering the voltage and temperature monitoring circuitry, and it uses +12V for programming the onboard Flash EPROMs. The RS232 transceivers operate independently from the +/-12V supply.

Warning: ***If the voltage on the +/- 12V pins in the backplane is higher than 12V, remove jumper J67. See figure A3, Miscellaneous jumper settings, on page 138.***

External Power

By repositioning two power jumpers (see below) the VBT-325 can also be powered from an external +5V power source through a front panel inlet (*). This is useful if there is insufficient capacity in the power supply in the card crate to supply the additional current required by the VBT-325. External power supply also allows tracing the activity in a VME system during power up sequences.

When powered from an external 5V supply, the +/-12V is still supplied from the backplane. However, this is only used for powering the voltage and temperature readout circuitry, and for programming of Flash EPROMs during firmware upgrades. Thus, it is fully possible to operate the board only from an external 5V supply, keeping in mind that the functions mentioned above will not be operational. This is a typical situation when analyzing power up sequences, where the VBT-325 is powered externally, but the VME chassis is not yet powered up.

*) Connector type: Molex, Part #5557-02R with 5556-TL terminals. VMETRO can supply cable with connector, VMETRO part # 401-325EPC.

Power jumpers

There are two "heavy-duty" jumpers (blue, with handle) that connects 5V power to the board from the VME backplane or from the external power source through the front panel inlet. The factory setting of these jumpers are in the Z1 and Z2 positions between the P1 and P2 connectors, supplying 5V from the VME backplane (see figure below). To allow for external power supply, **move both jumpers** to the Z3 and Z4 positions on each side of the power inlet connector (see figure below).

Note: *Never leave one jumper in each side or the board! Both jumpers must always be placed in either Z1 & Z2 or in Z3 & Z4!*

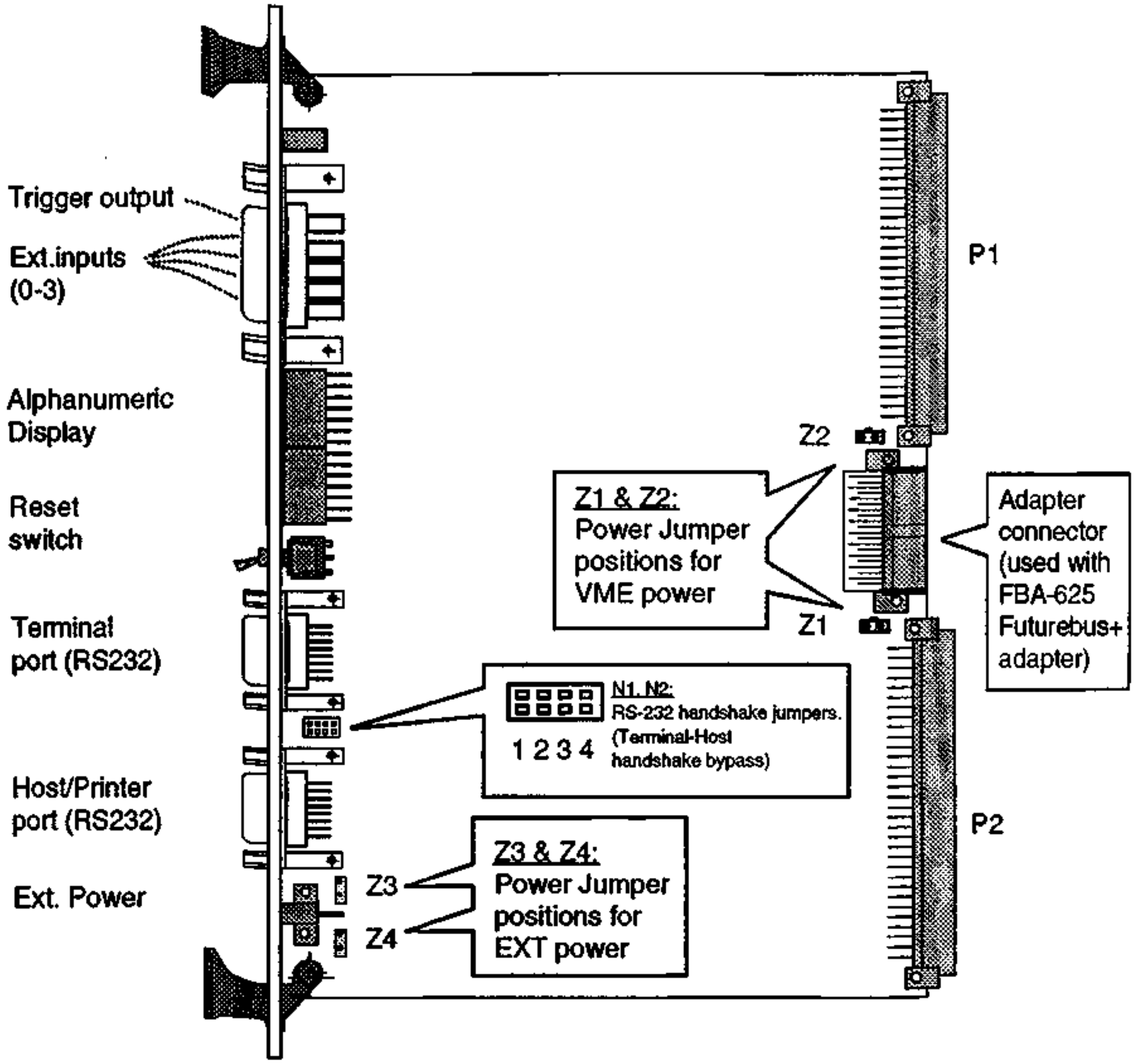


Figure 5. VBT-325 Board layout.

(Jumpers for external power supply and RS232 handshake bypass are shown. See "Terminal Connection" chapter below and the section "JUMPER SETTINGS" at the end of this manual for more details on jumper settings.)

Terminal Connection

The VBT-325 can be operated from an ASCII terminal (like VT100 etc.), an IBM-compatible PC or a workstation. If run from a PC or workstation, a terminal emulator must be used. VMETRO may supply VT100 terminal emulator for PCs that are tailored to the user-interface of the VBT-325 with respect to function keys etc. (This terminal emulator, VT100.EXE, is supplied on the firmware distribution diskette free of charge.)

The same basic screen appearance and look-and-feel of the user-interface is achieved no matter whether a terminal or PC/workstation is used. However, use of a PC with the VMETRO-supplied terminal emulator gives a choice of different screen colors for the background, menus etc.

RS232

There are two serial ports on the VBT-325. One is used for connection to a terminal, PC or workstation (see above), and the other is used for connection to a host (like a CPU board in the VME system where the VBT-325 is installed) or a printer for dumping screen images or trace data.

The RS232 ports are connected as shown in figure below. Normally, only the TxD, RxD and GND (pins 2, 3 and 5) need to be connected. As can be seen, the Host/printer port have the transmit/receive signals swapped compared to the terminal port.

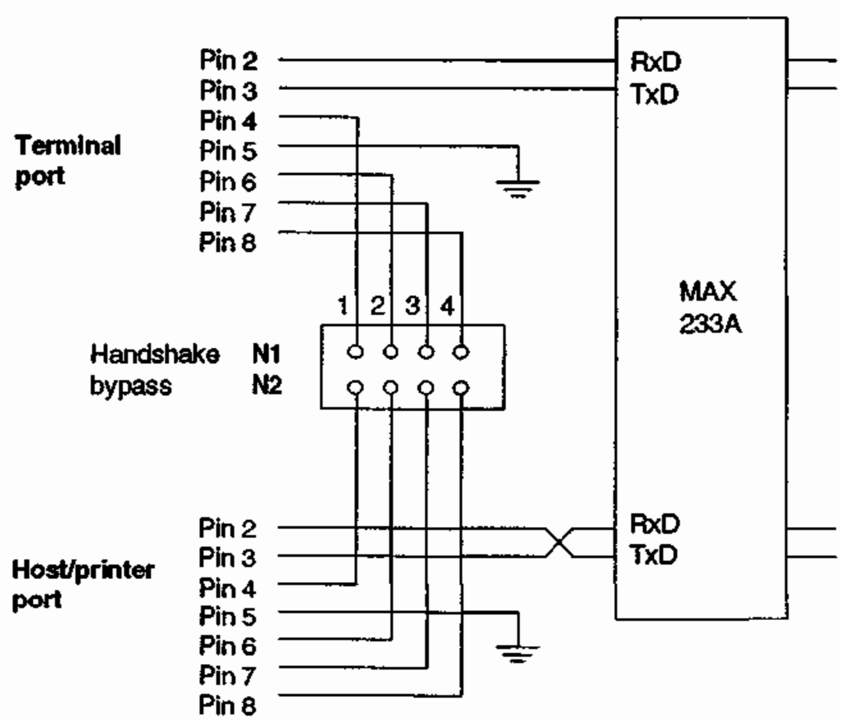


Figure 6. RS232 connections on VBT-325.

Terminal cable Most terminals use a 25-pin connector (DB25) for the RS232 connection, and a cable as shown in the figure below should be used. (This cable can be purchased from VMETRO, part # 401-TER-232).

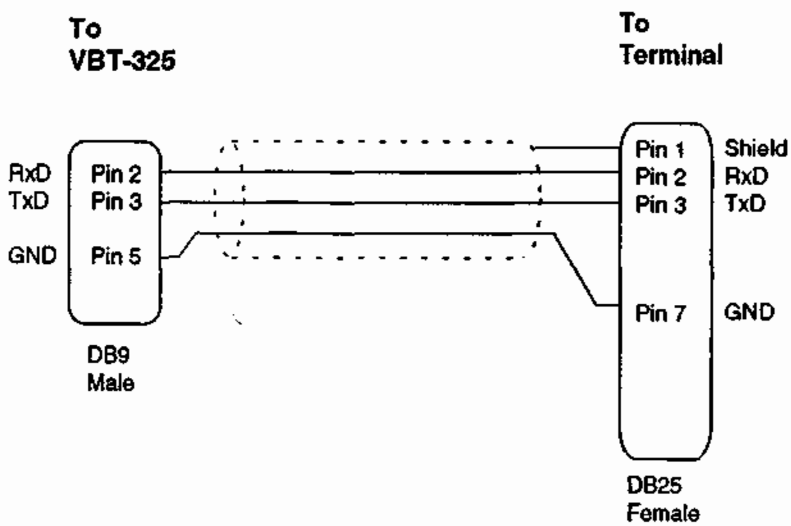


Figure 7. Recommended terminal cable connections

PC cable If a PC is used as a terminal, or when upgrading firmware through the RS232 port from a PC (refer to the section "FIRMWARE UPGRADE PROCEDURE"), a cable with crossed TxD and RxD lines must be used, see figure below. (This cable can be purchased from VMETRO, part # 401-PC-232).

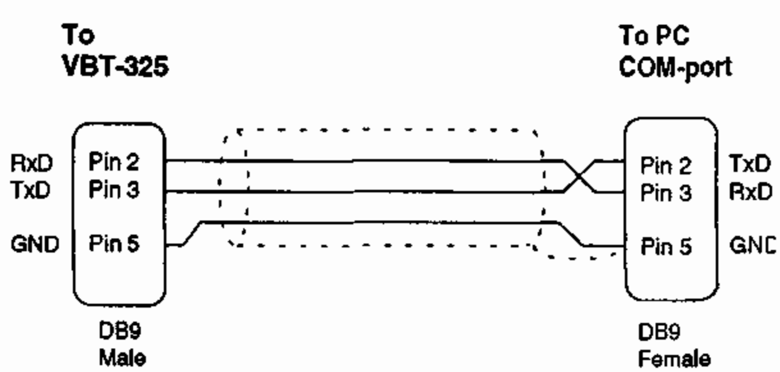


Figure 8. Recommended PC cable connections. (Note that pins 2 and 3 are crossed.)

Note: *If hardware handshake is enabled in the terminal emulator, it may be necessary to short together pin 4,6 and 8 in the plug connected to the COM-port for proper operation.*

Baud rate etc.

The VBT-325 is able to run with baud rates from 1200 to 19200 baud. By default, the board automatically selects the baud rate of the connected terminal when the user types <CR> after reset (refer to next chapter, "Power-on"). The baud rate can also be manually changed by the command `'Utilities/Serial ports'`.

Transparent mode

A convenient way of using the VBT-325 is to let the analyzer be inserted between the terminal and CPU in your VMEbus system, so that the RS232 cable normally attached to the CPU board is instead connected to the Terminal port on the VBT-325. Then, a one-to-one RS232 cable (with pins 2, 3 and 5 connected) can be connected from the Host port of the VBT-325 to the terminal port on the CPU. (This cable can be purchased from VMETRO, part # 401-TM-232). See figure below.

Note: *Do not Dump or Load Trace while in Transparent mode.*

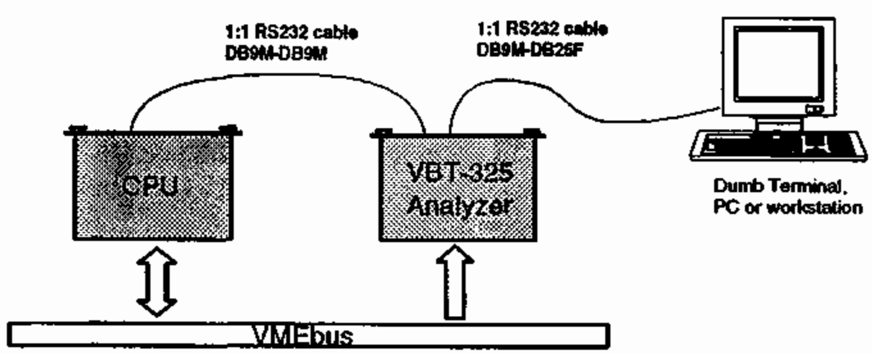


Figure 9. RS232 connections for Transparent mode

Terminal-Host RS232 handshake

The VBT-325 does not use or provide any hardware handshake signal on RS232. However, when using transparent mode between a terminal and a

CPU/host that utilizes hardware handshake (DTR, DSR, RTS, CTS), a convenient way of retaining this connection is simply to bypass the handshake signals from the terminal port to the host port through the VBT-325. This can easily be done by installing jumpers in the N1 and N2 jumper field (N1-1 to N2-1 etc.), thereby making a one-to-one bypass of the handshake signals between the terminal and the host. (Note: Since a time delay is introduced between the two serial ports, this approach is only recommended when the handshake signals are used in a static manner.)

Power-On

At this point you should have:

- Installed the VBT-325 in a proper slot
- Checked the power supply
- Connected a terminal (or PC/WS)

Power can now be applied, and after power-up, the front panel display on the VBT-325 will keep flashing with the text:

```
19k2 81n
```

Automatic Baud Rate detection

When the LED display is flashing, the VBT-325 is in the auto baud rate mode, and is expecting one or more <CR> from the terminal to determine the actual baud rate. When the baud rate is determined, the start-up menu will be written to the terminal (see next page).

By default, 8 bits per character, 1 stop bit and no parity is used. This can be changed by the '`Utilities/Serial ports`' command.

19200 baud recommended To get an optimum response time when operating the VBT-325, it is recommended to set the terminal/PC to 19200 baud.

No response? If the display does not show the baud rate after power-up, try to activate the reset switch on the front panel. If this still does not work, please refer to the section "FIRMWARE UPGRADE PROCEDURE", and check that the proper Boot EPROM is installed, that the jumper settings are correct, especially the jumpers that affects PROM vs. FLASH selection.

If the start-up menu does not appear on the screen, please check that cables are connected as described in the chapter "Terminal connection", and that your terminal / PC / WS is set to 8 bits per character, 1 stop bit and no parity.

Start-up menu

After power-on, the start-up menu is written to the terminal as shown in the figure below. This menu identifies product model, firmware version, baud rate, terminal type (default or previously selected type), and what kind of piggyback module is installed (if any).

```
V M E T R O       VBT-325C VME+ ANALYZER   FIRMWARE VERSION: 4.99.120

TERMINAL PORT       : 19K2 81N
HOST PORT           : 9600 81N
TERMINAL TYPE       : DEC VT-100/VT-102
PIGGYBACK CONNECTED: VBAT-PB

START-UP OPTIONS:

T:  SELECT NEW TERMINAL TYPE.
C:  CLEAR NON-VOLATILE MEMORY.

SELECT AN OPTION OR TYPE <CR> TO CONTINUE:
```

Figure 10. Start-up menu (as shown when VBAT-PB installed).

The start-up menu contains two menu options described below. Both are activated with a single key as indicated.

Select Terminal type (T)

The user-interface of the VBT-325 is fully screen-oriented, taking advantage of the graphical properties of terminals like VT100 etc. This requires that the user specifies which terminal or terminal emulator he is using. By typing 'T', a list of the supported terminal types is given.

Usable terminal types are:

1. DEC VT-100/VT-102
2. VMETRO VT-100 EMULATOR / ANSI.SYS ON MDA SCREEN

3. VMETRO VT-100 EMULATOR / ANSI.SYS ON COLOR SCREEN
4. VMETRO VT-100 / ANSI.SYS VGA COLOR 50 LINES
5. VMETRO VT-100 EMULATOR / ANSI.SYS BLACK BACKGROUND
6. TANDBERG TDV 1200, 2200, 2200/9, 2200S
7. DEC VT-52

The selected terminal type is stored in non-volatile memory, and unless this is cleared, it is not necessary to select the terminal type every time the board is powered up.

PC as terminal

By means of a terminal emulator, a PC can be used as a terminal for the VBT-325. For this purpose, VMETRO supplies a VT100 emulator free of charge, included on the firmware upgrade diskette. (Please contact VMETRO or its agent for a free copy if required). The user may select various options for running a PC as terminal emulator, where the colors and number of lines per screen varies. For a PC with a standard size (12-14") VGA color screen, the recommended type is #3, "ANSI.SYS on color screen". This gives 24 lines with a blue background.

50 lines/ screen

For PCs with high-quality and/or large screens, one may want to take advantage of a 50-lines display to allow more information to be shown on the screen. This is provided by selecting terminal type #4, "VGA Color 50 lines". (If VMETRO's VT100 emulator (V2.08 or higher) is used, it must be started with the ` -h 50` option. Ref. section "VMETRO VT100 EMULATOR - VT100.EXE" in the back of this manual for more details).

<CR> = Continue

Typing <CR> will bring up the Setup screen of the user-interface. At the same time, the front panel **LED display** will show the actual voltage of the 5V supplied to the board.

5V OK? A reading of at least 4.95V should be seen in the display for safe operation of the board. (Ref. power supply considerations in previous chapter.)

Clear Non-volatile memory

C = Clear This command (typing 'C') will clear all contents of the Non-volatile RAM memory on the board, and should be used only if a fatal software crash has occurred, e.g. if the operation of the user-interface does not behave correctly etc.

STOP

Warning: ***This command will cause all user setups to be lost.***

3 FUNCTIONAL DESCRIPTION

In this chapter you will find a description of the main blocks of the VBT - 325. The function of each block is described, and the signal path is shown.

Main blocks

The VBT-325 contains two separate independent analyzers, one 128-bit wide analyzer pre-configured for the VMEbus, and one 64-bit wide analyzer that can be configured either for VSB, SCSI or other busses on the P2 connector of VME boards. Both analyzers on the board have individual sampling logic, word recognizers, trace memories and triggering circuitry.

Each analyzer part of the VBT-325 consists of three main stages, through which samples are passing during the acquisition process:

- Sampling stage
- Word Recognition / Triggering stage
- Sample Storage / Statistics Counting stage

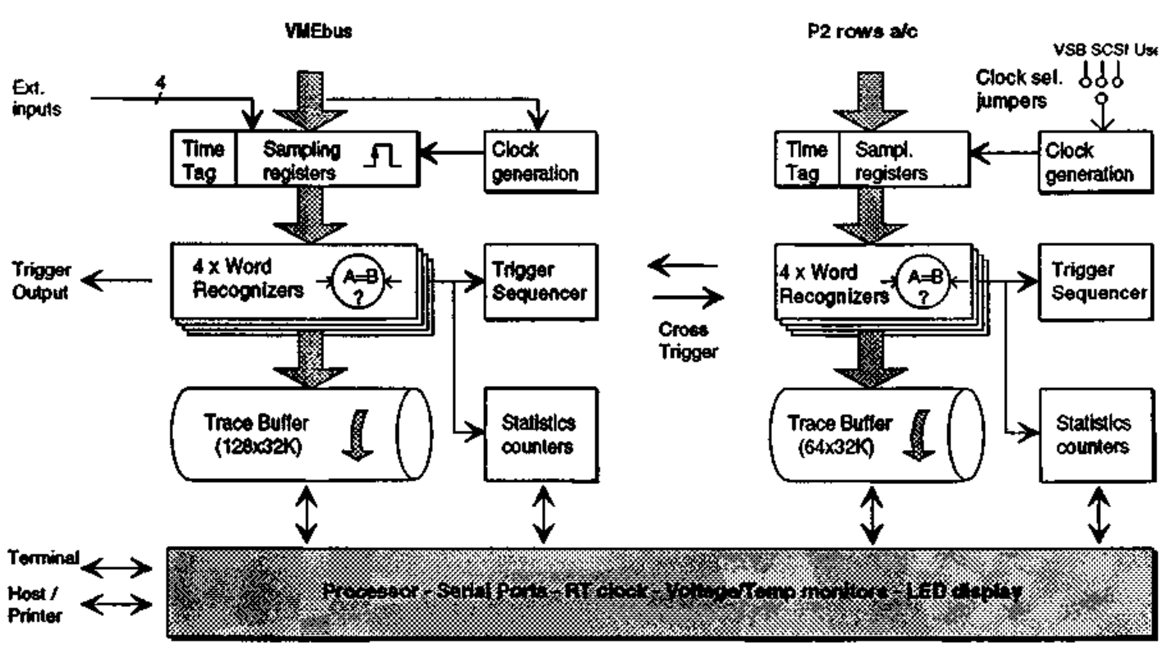


Figure 11. VBT-325 Block Diagram (shown in "Twin mode")

Twin mode The two analyzer parts of the VBT-325 can be operated in "twin mode" or "wide mode", defined by jumper settings. In twin mode, as shown in the figure above, the two analyzers operate independently, with separate acquisition paths, word recognizers, trigger sequencer, time tags, statistics counters and trace memory. This allows independent and simultaneous analysis of the VMEbus and the P2 bus. Cross-triggering between the two analyzers is possible both ways by means of a dedicated trace channel bit which is hardwired directly from the trigger output of the other analyzer part.

Wide mode For use in applications that require one wide analyzer, the two analyzer parts can be configured with jumper settings to operate as one wide analyzer with as many as 177 input channels. Together with a provision for externally generated sampling clocks (taken through a connector located between the P1 and P2 connectors), this gives flexibility to use the VBT-325 for other busses than VME/P2. As an example, the FBA-625 Futurebus+ Adapter transforms the VBT-325 into a complete, standalone Futurebus+ Bus Analyzer (the FBT-625).

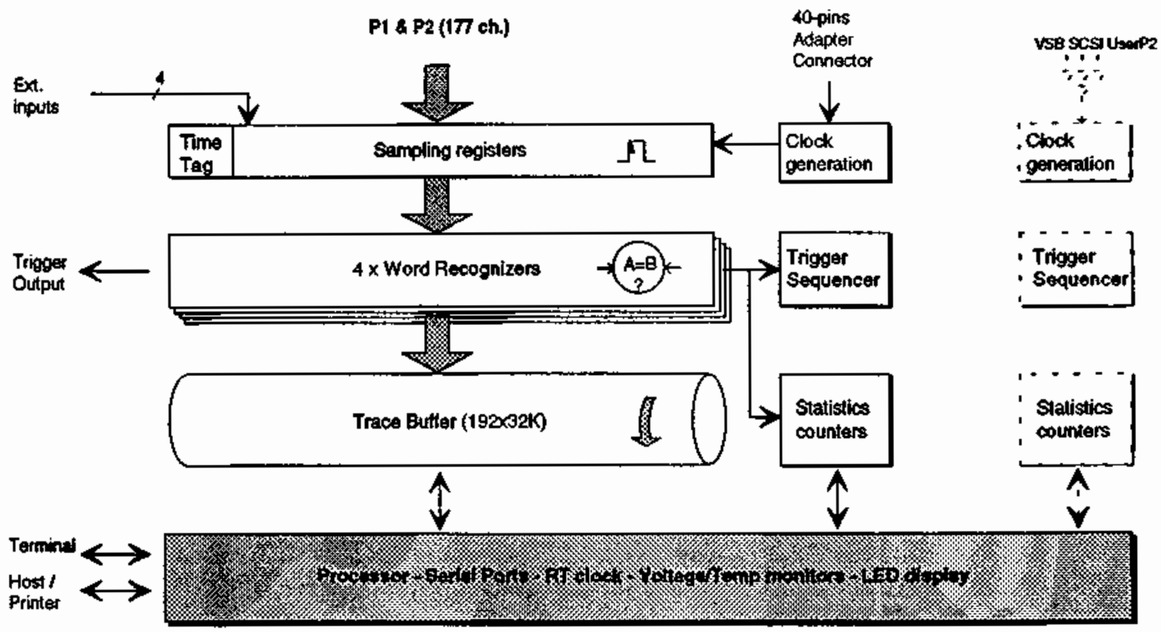


Figure 12. VBT-325 Block Diagram, shown in "Wide mode". (As used when configured as a Futurebus+ Analyzer together with the FBA-625 Futurebus+ Adapter.)

Sampling stage

The sampling stage contains sampling registers and clock generation circuitry that provides synchronous or asynchronous sampling of the target bus.

State (Synchronous) sampling

Synchronous sampling is used for *state* analysis, and captures cycles from the target system one by one, so that each collected cycle forms one line in the trace buffer. This sampling mode requires that the sampling logic extract sampling clocks from the target bus at the correct times in order to store information like address, data, transfer size and status in a compact form in the trace buffer.

Time Tag In order to measure the elapsed time between each sample stored in the trace buffer, the sampling stage includes a "Time Tag counter". The value of the Time Tag counter is stored in separate bits in the trace buffer together with each sample. This allows the time to be displayed either as relative time between samples, or as absolute time from the trigger point.

The VBT-325 is equipped with protocol-sensitive state sampling logic for VMEbus, VSB and SCSI, as described below. Jumpers are used to select VSB or SCSI sampling. For user-defined P2 busses, the user must supply a proper sampling clock through a pin socket on the board. (Please refer to section "JUMPER SETTINGS"). (For VXI P2a/c, the VSB jumper setting is used).

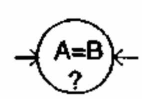
The ch. "FUNCTIONAL DESCRIPTION, Sampling methods" gives details on how the VBT-325 performs synchronous sampling of the supported target busses.

Timing (Asynchronous) sampling

Asynchronous sampling is used for *timing* analysis, i.e. the bus is sampled at a fixed rate that can be selected at eight different speeds as fast as 50MHz or as slow as 97.7KHz. This sampling mode is available for all the possible target busses supported by the VBT-325.

With a sampling rate of 50MHz the bus is sampled every 20ns, which is sufficient to show the general timing relation between signals on a bus like VME or VSB. However, for really detailed timing analysis of difficult hardware problems like glitches etc., a faster sampling rate is required, like the 200MHz sampling rate offered by the S/TIM200-PB piggyback module. This gives 5ns resolution and the ability to trigger on any signal on the bus, also specifying a particular duration of the timing pattern. (A short-form description of the piggyback modules available for the VBT-325 is given later in this chapter.)

Word Recognition / Triggering stage



A central element of any logic analyzer is the ability to recognize events, i.e. a particular signal pattern, in the target system so that the acquisition of event samples can stop at the desired moment, i.e. the process referred to as "triggering".

There are four full-width word recognizers both for the VME part and the 64-bit P2 part of the VBT-325, and these form the foundation not only for triggering, but also for store qualification (store filter) and counting purposes. Counting can be used to delay the triggering process until a particular number of bus cycles occur, and it is also used for statistical purposes. Thus, the three main purposes of the word recognizers are:

- Triggering
- Store qualification
- Occurrence Counting

Busses, groups Any signal or signal group can be included with a particular value or as "don't care" in the word recognizers. Signals from the target bus may be included in the word recognizers as a *bus* (like address and data), they may be combined into *groups*, like the SIZE group for VME, consisting of DS1*, DS0*, LWORD* and A01, or simply as individual signals.

VME64 The VBT-325 fully supports 64-bit address and data as defined in the VMEbus specifications, rev.D, which is used for the multiplexed block transfer modes MBLT and SSBLT. This means that the 32-bit data bus D(31:0) and the 31-bit address bus A(31:1), plus the LWORD* signal acting as data bit 32, can be treated as one 64-bit entity in the word recognizers.

VME64 signal A special signal called VME64 is available in the word recognizers. This signal is asserted when an AM-code for VME64 is detected, and can be used as part of an event pattern.

Not When multiple signals are combined into a bus or group, a "Not" (!) operator is available in many cases, allowing the specified value to be treated as true if the condition does not occur. This allows conditions like:

$$\text{Data} \neq 0000\ 0000$$

Range Each of the four word recognizers allows ranges to be specified on both the VMEbus address and data, as well as for the VSB multiplexed address/data bus.

This allows functions like:

$$X \leq \text{Address} \leq Y$$

"Outside" range can also be obtained, by using the "Not" operator on the address or data field.

The hardware allows four 32-bit VME address ranges, four 32-bit VME data ranges and four 32-bit VSB address or data ranges to be active simultaneously.

A64 range Alternatively, 64-bit address ranges (A64 according to VMEbus rev.D) can be specified in each of the four word recognizers for the VMEbus part. (Since a 64-bit address is using the 32 data lines, no data range can be used together with A64.)

Sequencer

The Sequencer is a triggering state machine that allows the analyzer to trigger not only on one particular event pattern or cycle, but also a sequence or combination of such. This allows the user to trigger on complex situations, for example when a particular memory cell is written to immediately after an interrupt occurred, while all other references to the same location are ignored.

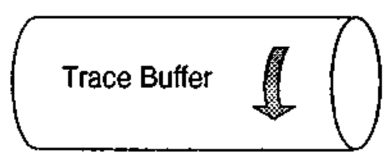
16 states Up to 16 states can be programmed in the sequencer, and each state allows a level of "If..Elsif..Else" statements. Please refer to the chapter "OPERATION" for more details.

Sample Storage stage

After the collected samples have passed the sampling stage and the word recognition/triggering stage, they will arrive either in the sample storage or statistics counting stage.

Trace Buffer

During normal trace sessions, the samples are stored in the trace buffer, a 32K deep circular memory addressed by an address counter which is incremented after each stored sample. This buffer is written to continuously until a trigger is found, overwriting previous samples when full.



Trigger position

When a trigger occurs, the process of storing further cycles depends on the selected *trigger position.* If the trigger position is set to "End of Trace" (100%), no more samples will be stored after the trigger, and the samples recorded in the trace buffer will be presented on the screen. By contrast, if the trigger position is set to "Start of Trace" (0%), the entire trace buffer will be filled with new cycles before the acquisition process stops. In between, there are possibilities to select trigger positions as 25, 50 and 75%.

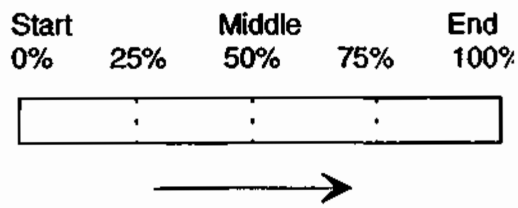


Figure 13. Five trigger positions in the trace buffer can be selected.

Note that the trigger may occur before the trace buffer has been filled completely ahead of its specified trigger position. In such cases the trace buffer will be only partly filled before the trigger. (Imagine if the second sample collected was the trigger, and the trigger position was 50%, then 16K-1 cells in the first part of the trace buffer will be empty.)

Note also that if the trace is not completely filled *after* trigger and then halted manually (possible in all cases except End of Trace), the unused post-trigger portion of the trace buffer may contain valid pre-trigger samples from the previous "round" of sampling (remember, the trace buffer is circular). If this is the case, these samples will be shown since they may contain useful information.

Statistics Counters

When the VBT-325 is used for statistical purposes rather than for storing samples in the trace buffer, the bank of counters will be incremented according to the values programmed into the word recognizers.

VMEbus There is a 20-bit counter attached to each of the word recognizers, as well as a similar counter for each of four VMEbus Bus Level detectors, for a total of eight statistics counters. In addition, there is a ninth counter used for counting the total sample count. For VMEbus statistics, these counters are used to provide histograms of four user-defined events at the same time as the distribution of VMEbus traffic among the four bus levels are shown. There are also statistics functions that use pre-defined event patterns to provide direct readout of e.g. Bus Utilization.

VSB/SCSI/P2 For the P2 part there is a 20-bit counter attached to each of the word recognizers. In addition, there is a fifth counter used for counting the total sample count. These counters are used to provide histograms of four user-defined events on VSB, SCSI or P2 bus.

Sampling methods

The principal task of the VBT-325 is sampling the bus activity. While *timing* (asynchronous) sampling is straight forward in the sense that samples are taken at fixed time intervals, *state* (synchronous) sampling depends heavily on the bus protocol.

In order to fully interpret what is seen in the trace display when state sampling is used, it may be necessary to understand how the analyzer samples the bus. Thus, a detailed explanation of the employed state sampling methods is given below.

VMEbus state sampling

To properly capture all information of the transactions on a VMEbus backplane during state analysis, the VMEbus sampling logic latches bus signals both at the completion of a bus arbitration and at each data phase. This is for normal cycles, Read-Modify-Write cycles and Block cycles, which include SSBLT cycles.

Bus Request latching

Normally, the active bus request signal (BRx*) on the VME backplane goes away immediately after the corresponding bus grant (BGx)* is taken low, too early for being sampled together with address, data etc. on the falling edge of DTACK* or BERR (described below). To ensure capture of the active bus request(s) even during state sampling, the pending BRx* may be latched internally until DTACK* goes low.

(This feature can be turned off by the command '`Edit/Sampling mode/Options/State Sampling options`'.)

Bus Grant latching - BgL

A VME system arbiter completes an arbitration by issuing one of four Bus Grants (BG3-0*), and when this is received by a bus master with pending bus request, it asserts BBSY*. This causes the Bus Grant to go away immediately, so in order to keep track on which bus level the following transactions belong to it is necessary to store the actual bus grant at falling edge of BBSY*. The VBT-325 stores information about the active bus grant as an internal two-bit value called *Bus Level,* (or "Bus Grant Level"), shown as **BgL** in the trace. In addition there is an extra bit that is set if the VBT-325 sees BBSY* being asserted without any valid Bus Grant, a situation that will occur if the VBT-325 is located to the right of the granted bus master in the card crate. (Refer to daisy-chain considerations in the Installation chapter). Information about the actual bus level is presented in the trace as a number from 0 to 3, or as a '-' signifying "No bus grant detected".

Note: *Do not activate the RESET button after VMEbus cycles has started on the backplane. This may cause BgL to be erroneously shown as BgL=0.*

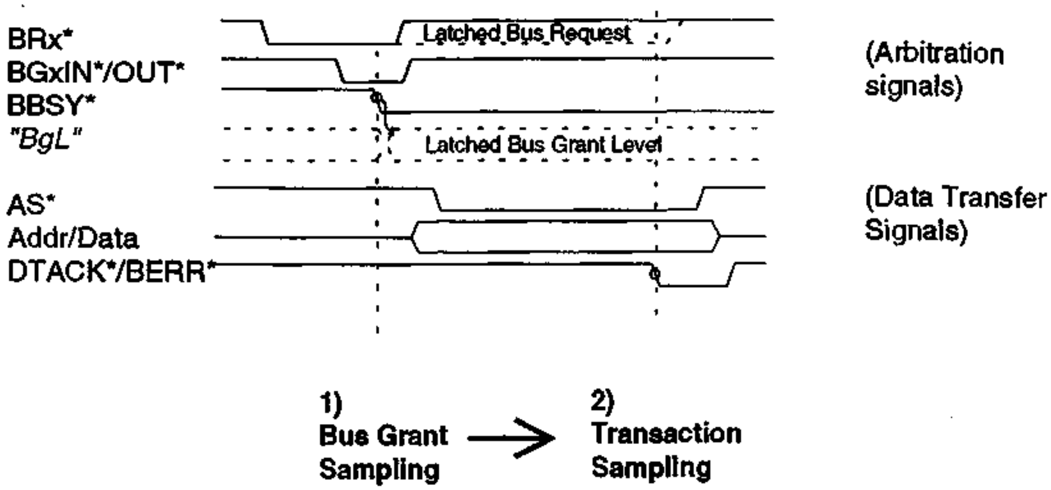


Figure 14. Synchronous (State) sampling of VMEbus, showing the internal latching of the BR and BG* signals in the analyzer (dashed lines).*

Data cycle sampling

Normally, bus transactions are sampled on the falling edge of DTACK* or BERR*, conditioned by DS1* or DS0* being low. This will capture all address, data and control signal information, as well as the stored Bus Level from the Bus Grant sampling as described above.

IRQ -> IACK

Normally, the active interrupt request signal (IRQx*) on the VME backplane goes away immediately after IACK* is taken low, too early for being sampled by the falling edge of DTACK*. To ensure capture of the active interrupt request line in interrupt acknowledge cycles ("IACK") even during state sampling, the pending IRQx* may be latched internally until DTACK* goes low (actually until IACK* goes high).

This feature can be turned off by the command `'Edit/Sampling mode/Options/State Sampling options'` (Note: HW ECO level B6 or higher is required for this feature to be available).

RMW, Block

If a Read-Modify-Write cycle or Block cycle is transferred, this is signalled on the VMEbus by not taking AS* high between cycles. This will be detected by special circuitry in the VBT-325, and a separate bit in the trace is set to indicate this. The address modifiers are then used by the trace decoding firmware to distinguish between RMW and Block cycles.

VME64

If an AM-code for VME64 is detected on the VME bus (i.e. AM = 00, 01, 03, 06, 07 ,08 ,0C, 38, 3C), a special signal called "VME64*" (active low) is generated internally in the VBT-325. This signal is also available in the word recognizers.

SSBLT

If the address modifiers signify an SSBLT cycle (Source-Synchronous Block Transfer, AM-codes 6 or 7), the VBT-325 will sample only the address at falling edge of DTACK*, while data will be sampled on both falling and rising edge of DS0* during write cycles, and at both falling and rising edge of DTACK* during read cycles, see figure below.

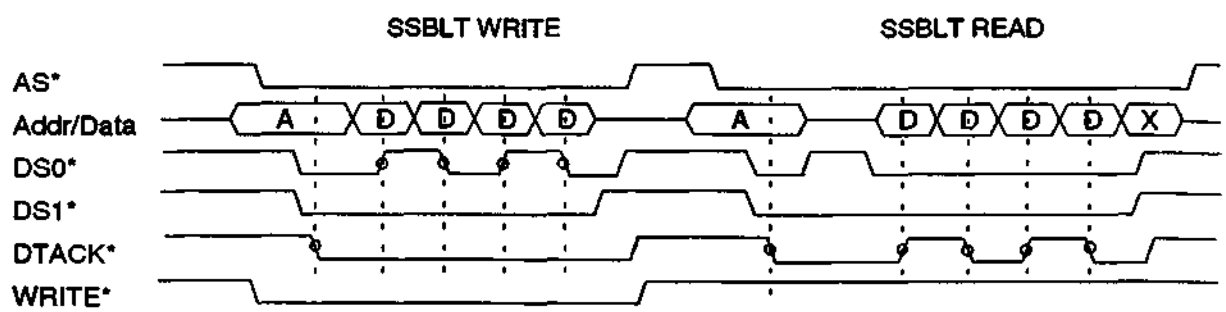


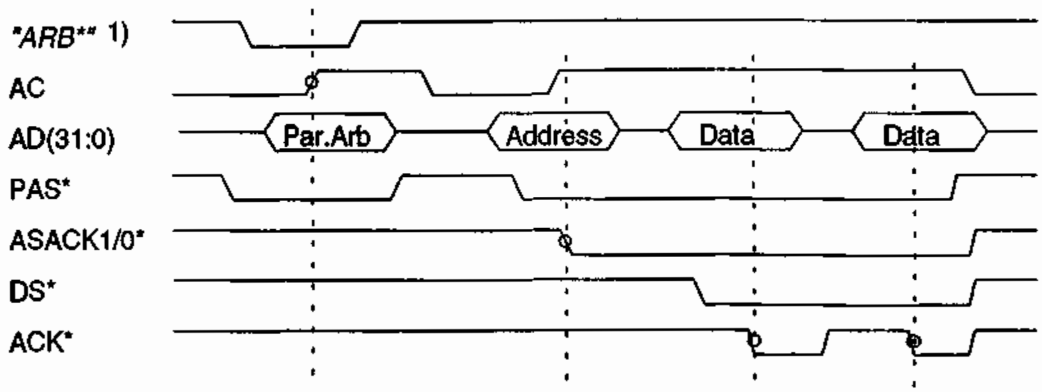
Figure 15. Synchronous (State) sampling of SSBLT cycles on the VMEbus

VSB state sampling

The VSB (VME Subsystem Bus) is a multiplexed bus defined on the P2 a/c rows on certain VME boards. The P2-part of the VBT-325 has protocol-sensitive sampling logic for VSB that will provide samples at three different phases of the bus protocol during state sampling. These are:

- Parallel arbitration
- Address phase (DS* is 'ADDR' in trace)
- Data phase (DS* is 'DTA' in trace)

The parallel arbitration phase is clocked on the rising edge of the AC signal when an internally generated signal called "ARB*" is low (refer to figure below). This will provide a sample in the trace buffer that will show the arbitration value found on the AD-bus, bits (30:24). The address phase is sampled at the falling edge of ASACK1* or ASACK0*, while the data phases are sampled at falling edge of ACK* (or ERR*). Refer to the figure below.



1) "ARB*" is a signal generated internally in the VBT-325 as follows:
ARB=!SPACE0 & !SPACE1& WR & BREQ & PAS & !BUSY

Figure 16. Synchronous (State) sampling of VSB

SCSI state sampling

A single-ended SCSI-bus may be attached directly to the P2 a/c rows 8-32 by using a standard 50-lead flat cable, please refer to the section "SCSI PINOUT ON P2" for actual pinout. In synchronous (state) sampling mode, the SCSI bus is sampled both in the arbitration cycle, to capture the actual device number (ident), and then once for each of the command, data, status and message cycles. The arbitration cycle is sampled on rising edge of the BSY* signal, while the other cycles are sampled at falling edge of ACK*, please refer to figure below.

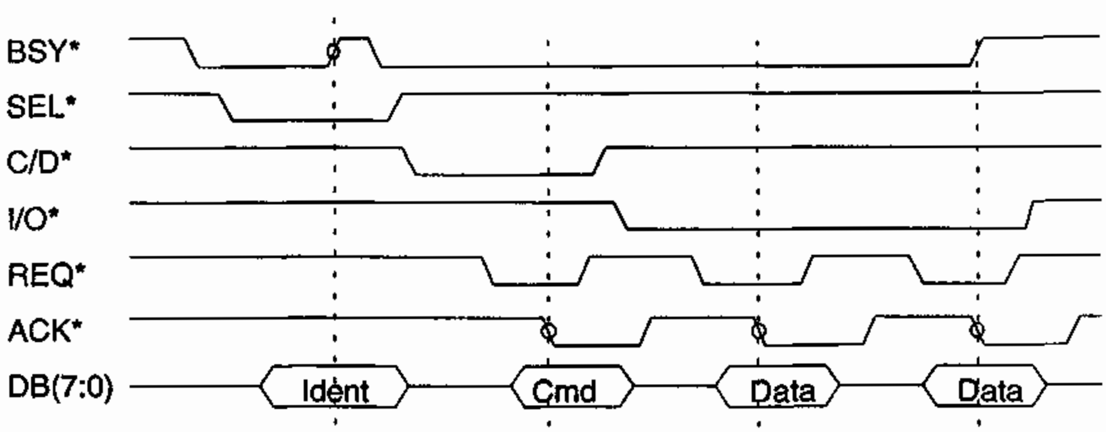


Figure 17. Synchronous (State) sampling of SCSI

Note: *The VBT-325 generates an internal signal called "ARB*" with the logic function ARB = !I/O & SEL, used to sample the arbitration phase in synchronous mode. This signal can be found in the SCSI trace as "ARB*".*

SCSI Pinout

The predefined pinout follows the Motorola standard for an 8-bit single-ended SCSI. If another pinout, or differential SCSI, is used by the application, it is possible to make a small adapter board that plugs on to the P2 connector, routing the signals properly. This could also include converters from differential to single-ended signals levels if necessary. (Note: To take advantage of the SCSI-bus specific sampling, it is essential that the signals SEL*, BSY*, I/O*, and ACK* are on the same predefined pins as defined in the section "SCSI PINOUT ON P2".)

SCSI-2 FAST

Both SCSI-1 and SCSI-2 with 8-bit data is supported, including FAST SCSI-2.

VXI sampling

When used with the VXE-35C, the VBT 325C will sample the VMEbus portion of VXI just as when it is used as a pure VMEbus analyzer. The TTL signals (i.e. TTLTRG7:0* and LBUS11:0) on P2a/c of VXI systems may be sampled using up to 50MHz timing mode, or in state mode by using the user-defined sampling clock as described below.

User-defined P2 state sampling

For synchronous (state) sampling of the 64 pins on rows a/c on the P2 connector, a user-supplied clock can be connected to a pin socket on the board (J43 located close to pin P2c1, please refer to section "JUMPER SETTINGS"). The P2 pins will then be sampled at each rising edge of this signal, see figure 6.

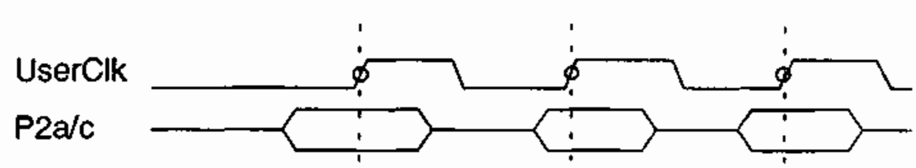


Figure 18. Synchronous (State) sampling of a user-defined P2 bus.

Note: *Future firmware will support user definable signal groups, pin allocation etc., for analysis of proprietary P2 busses.*

External inputs

In3:0 in VME part

The VBT-325 has provision for four external signal inputs on its front panel, through mini-coax connectors labelled In0-In3. These inputs are available in the VME part of the VBT-325, found under the signal group EXT. One cable is supplied with the VBT-325 that fits these mini-coax connectors, and provides transition to a standard BNC connector.

In3:0 to P2 part using cross trigger

If the user wants to trigger on any of the In3-0 signals together with the VSB/SCSI/P2 part (in twin mode), there is a cross trigger mechanism that can be used. This means that the VME part must be set to trigger on the In3-0 signals, and the P2 analyzer set to trigger on the cross trigger signal "VMEtrg". (Refer to the ch. "OPERATION, Cross triggering" for more information on cross triggering.)

Shared In3:0 and Time Tag bits

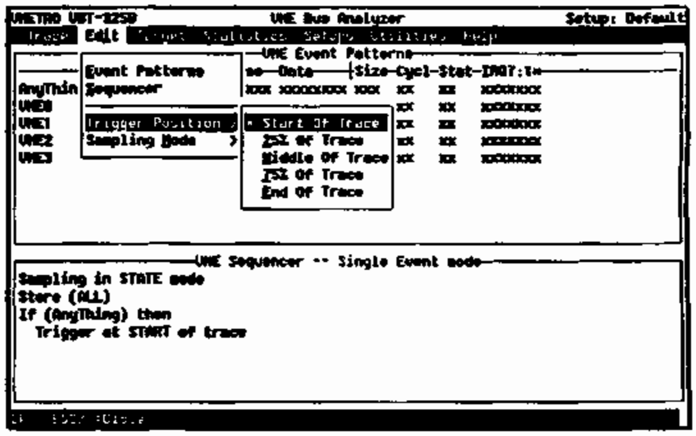
Please note that the four external input signals share four bits in the trace buffer with the "Time tag counter", and are controlled by the user (command `'Edit/Sampling mode/Options/State Sampling options'`.) If these bits are dedicated to the Time tag, i.e. "Extended time tag" is selected, then time intervals between samples up to 1 hour 38 minutes can be measured by the time tag in the trace. This comes at the cost of not being able to see the value of the external inputs in the trace buffer, but the external inputs can still take part in the trigger or store qualifier etc. When "Limited time tag" is selected, the max. time interval that can be measured is 6 minutes 8 seconds, and then the four external signals are present in the trace.

Shared In1 and Temperature Probe

One of the external signal inputs, the "In1", is also shared with the optional temperature probe, controlled by two jumpers (J66 and the "V+T/V" jumper.) Please refer to the section. "COMMANDS REFERENCE, Main menu, Utilities, LED Display, Temperature readout" for details).

4 OPERATION

The user-interface can be operated from a standard character-oriented terminal like VT100 etc., or from PC running a terminal emulator, like VMETRO's "VT100 Emulator V2.X" or other common terminal emulators. The user-interface of the VBT-325 employs the same principles as applied to the most modern computer products like Windows etc., with menu bars, pull down menus, dialog boxes and multiple windows.



All main commands are shown in a **command bar** on the top of the screen, and each command has a **pull down menu** attached to it that contains a list of **sub-commands**. Some sub-commands may present a **dialog box** for detailed specification of various parameters or choices, while others may present a secondary pull-down menu for further selections.

User-interface structure

Targets

With the various piggyback options available to the VBT-325, the product may consist of up to three analyzers and one pattern generator in its maximum configuration, see table below:

	VBT-325B	VBT-325C	VBT-325 w/TIM200-PB	VBT-325 w/STIM200-PB
VMEbus State/Timing Analyzer	√	√	√	√
P2 State/Timing Analyzer		√	√ [1]	√ [1]
200MHz Timing Analyzer			√	√
200MHz Stimuli/Pattern Generator				√

1) VBT-325C.

Throughout the user-interface, these four functional units are all referred to as *targets*, and the ones that are present in the given configuration appear as sub-commands under the `'Target'` command in the main command bar (described later under the ch. "COMMANDS REFERENCE, Main menu - Setup screen, Target").

Note that the VBAT-PB and VDRIVE-PB piggyback modules are of a different category, and thus are treated differently.

Screen categories

For the two possible *analyzer targets* that reside onboard the VBT-325 itself, the user-interface is based on three different screens:

- Setup screen
- Trace Display screen
- Statistics screen

The TIM200-PB 200MHz Timing Analyzer piggyback module is similar, but lacks the Statistics screen. Please refer to the separate User's Manual for the S/TIM200-PB Timing Analyzer and Stimuli/Pattern Generator module.

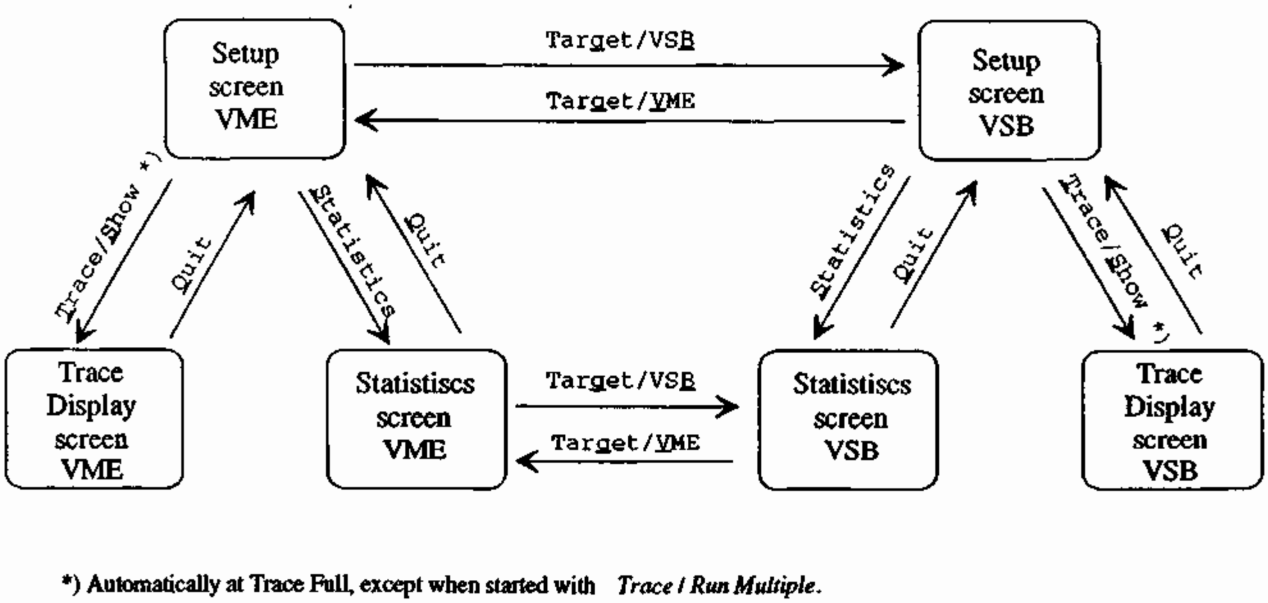


Figure 19. Screen structure for VBT-325C, shown with VSB as the selected P2 target bus. (Note that VBT-325B lacks the Target term unless the S/TIM200-PB is installed.)

Setup screen

The Setup screen is the main "control panel" of the selected analyzer (target). In addition to the "Main menu", this screen contains two major elements, the **Event Patterns** window and **Sequencer** window, used to define triggers, store qualifiers etc. These two windows are described in detail later in this section.

Trace Display screen

The Trace Display screen is where the contents of the trace buffer is displayed. The trace data may be displayed as an alphanumeric trace list or as waveforms. Multiple trace windows of either type may also be created. The command bar in the Trace Display screen is tailored to perform efficient navigation, searching and formatting of the trace data contents.

The Trace Display screen is automatically presented when the trace buffer becomes full during sampling. (Except when more than one analyzer (target) is started at the same time with the command sequence `'Run/Multiple'`. In this case, a sampling status box will be shown, see ch. "COMMANDS REFERENCE, Main menu - Setup screen, Trace, Sampling Status").

The command `'Quit'` is used to return to the Setup screen. In order to see the trace data from another target, one must revert to the Setup screen with `'Quit'`, select another target, and then execute `'Trace/Show'`, refer to the screen structure figure above.

(Future firmware versions will allow displaying trace data from another target without going via the Setup screen.)

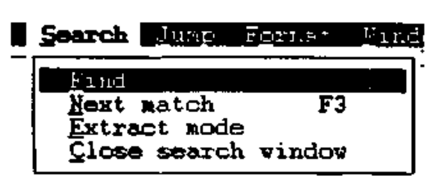
Statistics screen

The Statistics screen is used to control and see the results of a statistics session for the selected target. A special command bar menu is given, designed to provide a flexible and powerful environment for statistics measurements.

The command `'Quit'` is used to return to the Setup screen. However, it is possible to jump directly to another target from the Statistics screen in order to perform or review statistics measurements on another target without going to the Setup screen first.

Screen and Command elements

Pull down menus



← / → The pull down menus in the command bar are activated by a <CR> (↵) after the cursor is placed on the desired command using the left/right keys (← / →).

Sub-command selection

<CR> = Select When cursor is placed on a sub-command in a pull down menu, select this command by typing <CR>.

<ESC>= Abort <ESC> will abort a selection and close an open menu box. If no menu box is open, <ESC> will cause an exit to a menu level.

Dot '.' = <ESC> Since some terminal keyboards do not have an <ESC> key, the dot character '.' is used as an alias for <ESC>.

Commands that cannot be executed in the current context are shown in low intensity. Trying to use one of these will give a message like "Err: Cannot execute this command now". For example, the

```
Trace/Halt
```

command is disabled until you have executed

```
Trace/Run.
```

*** = Selection** In some pull-down menus, it is possible to select between options directly in the menu. In such cases, the current selection is shown with an asterisk '*' in front of the command in the pull-down menu.

Example: Under 'Format' in the Trace Display menu, the time tag type is shown like this:

```
  Absolute Time Tags
* Relative Time Tags
```

indicating that 'Relative..' is currently selected. To switch to the other type, simply type <CR> when cursor is placed on 'Absolute..'.

Accelerator keys

All commands and sub-commands can also be activated by a single-key command (accelerator key). The accelerator keys are high-lighted with an underscore (on terminals) or high-intensity (on PCs). By using these keys, a common sequence of commands can be quickly executed.

Example:

Type

```
's' for Search
```

or type

`'tr' for Trace/Run`

Note: *Command names starting with any of the letters A,B,C,D,E,F do not use the first letter for the accelerator key, since these letters are hexadecimal digits that may be typed directly into address and data fields etc. in the event patterns windows. In these cases, another key is used.*

Example:

`'i' for Edit`

Also, if several commands start with the same letter, a key other than the first letter is used as the accelerator key.

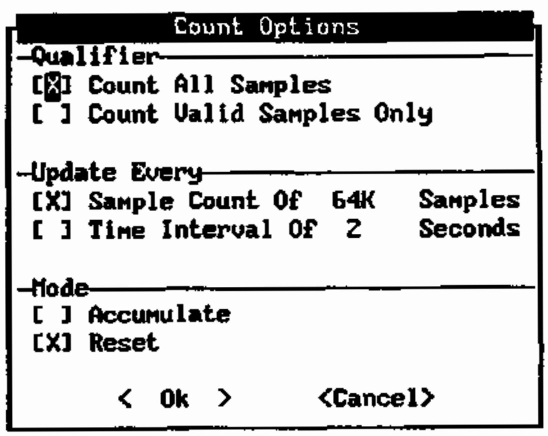
Script files Accelerator keys are also used for the individual selections in the pull down menu. This not only allows for quick operation by experienced users, but it will also simplify the process of generating self-running scripts, e.g. a text file created on a PC that executes commands to the VBT-325 to perform automated tests etc. Please refer to the section "VMETRO VT100 EMULATOR - VT100.EXE" for more information about making scripts.

Dialog Boxes

[x] (x) Dialog boxes are used when the user may choose between several alternatives by marking the desired options within the square brackets or parenthesis. If square brackets are used, as in the example to the right, you can select one or more of the alternatives in each groups. If parenthesis are used, also called "radio buttons", you can only select one of the alternatives in each group.

<SPACE BAR> When the cursor is placed at the desired option, use the <SPACE BAR> to toggle between selected [x] and unselected [ ].

As an example, under `'Options/Count options'` in the Statistics menu, a dialog box is presented so that the user may select the desired options.



↑/← / → For navigation within a dialog box, one may use the 'UP' (↑), 'RIGHT' (→) and 'LEFT' (←) keys.

'TAB' The 'TAB' and 'BACK TAB' keys can also be used to navigate in dialog boxes.

The 'DOWN' (↓) key will work fine unless you are positioned on a field with pull down options (like the 'Update every.. Sample Count ..' in the dialog box shown above). In this case, the 'DOWN' (↓) key will open a box displaying the possible options. Within pull down menus, use 'UP' and 'DOWN' to navigate and <CR> to select an entry. 'TAB' may also be used, this will select an entry and move to the next field in the dialog box. <ESC> will close the pull down menu without making a selection.

<CR>=Confirm When the choices are made, a dialog box command is executed by typing <CR>. Alternatively, the cursor may be moved explicitly to the "< OK >" position. Both <CR> and <SPACE BAR> will confirm the choices and close the dialog box.

To cancel a dialog box command, type <ESC> or '.' no matter where the cursor is, or type <CR> or the <SPACE BAR> at the "< Cancel>" position. Any selections made will then be ignored.

Function keys

To speed up the operation of the VBT-325 for experienced users, several function keys are implemented. However, certain terminal types lack some or all of the function keys, so each function key has a control character or other character as alias, as described below (control characters are indicated by a "hat character": ^_).

Note: *VT100, VT220 etc. have function keys labelled PF1-PF4, while the corresponding keys on PC keyboards normally are labelled F1-F4, with additional functions keys labelled F5-F12.*

'?' Brings up the Help screens. These can also be activated by the Help command in the main command bar.

<PF1> Enters Transparent Mode.

<PF2> or '^E' Brings cursor to the menu, or when in the menu, back to the last edited window. (Note that <ESC> or '.' will always bring the cursor to the menu from a window.)

<PF3> or '^F' Finds the next match to the search pattern when the searching in the trace buffer.

<F5> or '^R' Run - i.e. start sampling of bus cycles. Same as the command

```
Trace/Run.
```

<F6> or '^W' Moves cursor to the next [editable] window. Especially handy to switch quickly between the "Event patterns" and the "Sequencer" windows, or between opened trace windows.

<DEL> or '^BS' Deletes an object. A context sensitive dialog box will appear that explains the delete options at the actual cursor position. (Refer to dialog box editing above).

<INS> or '^N' Inserts an object. A context sensitive dialog box will appear that explains the insert options at the actual cursor position. (Refer to dialog box editing above).

Numeric keypad

When using a VT100 terminal, the numeric keypad can be used just as on PC keyboards, i.e. for cursor movements and <DEL> and <INS>, see figure at right.

NB: Remember to turn off "NUM LOCK".

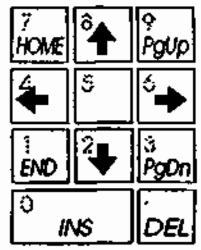


Figure 20. Numeric keypad key assignments

Keyboard template

For convenience, VMETRO has designed a keyboard template that will explain the use of the function keys, see figure below. This can be obtained from VMETRO upon request.

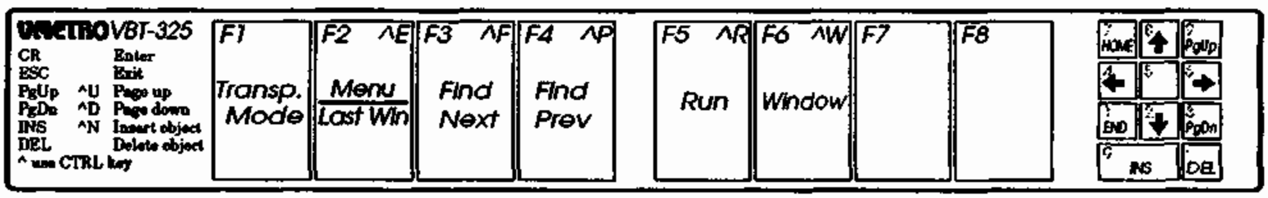


Figure 21. Keyboard template

Multiple windows

All screens presented throughout the user-interface consist of one or more windows. The window system offers a high degree of flexibility for presenting the desired information on the screen. For example, when displaying captured traces, the user may want to show several windows of trace data at the same time. One part of the trace can be shown as an alphanumeric trace list and one part shown as waveforms.

(Future firmware versions will also allow showing trace windows from VME and e.g. VSB at the same time, also a 200MHz waveform window with trace data captured by the TIM200-PB).

Message line

The bottom line of the screen (in inverse video) is used to present simple messages about the status of the analyzer and guide to the user as to which keys can be typed etc. This line will also show error messages.

Block Cursor recommended

When using terminals like the VT100 etc., it is recommended to select the cursor type **Block** for good visibility of the cursor throughout the VBT-325 user interface. If a Line cursor is used, this may be invisible in places like tick-boxes [ ] and in single-character fields in the event patterns.

Refresh Screen

The screen may be refreshed by typing \\ (backslash twice). This is useful if . characters are lost, when you change terminal, etc.

Event Patterns

The Event Patterns window defines the patterns to be loaded into the word recognizers of the VBT-325. The hardware provides four parallel word recognizers (separate for the VME part and the VSB/SCSI/P2 part), but the user may define a larger number of predefined patterns that can easily be taken into use.

Per default, a list of four user-alterable patterns is provided. They are labelled ***VME0..VME3,*** when VMEbus is selected, ***VSB0..VSB3*** when VSB is selected and ***SCSI0..SCSI3*** when SCSI is the selected target bus. In addition, there is an unalterable pattern called ***AnyThing***, which is always set to "Don't Care" (x). In the figure below, one user-defined event pattern "Mytrigger" is included.

In the default setup, the most important signals and signal groups are shown in the Event patterns window. The user may insert additional signals or signal groups, as well as additional patterns with user-defined labels (instructions are given below). Vertical or horizontal scrolling takes place when the user moves the cursor around in the window, to show the part of the patterns that is hidden outside the window borders.

Please note that no more than four of the defined patterns may be used simultaneously in the Sequencer program.

VME Event Patterns

	BgL	AS*	AM	Address	Data	Size	Cycl	Stat	IRQ7:1*
AnyThing	x	x	xx	xxxxxxxx	xxxxxxxx	xxx	xx	xx	xxxxxxx
Mytrigger	x	x	SSD	xxxxxxxx	xxxxxxxx	UBYTE	xx	xx	xx5xxxx
VME0	-2-	x	xx	xxxxxxxx	xxxxxxxx	xxx	xx	BERR	xx5xxxx
VME1	x	0	0D	FF0432xx	xxxxxxxx	xxx	xx	xx	xxxxxxx
VME2	x	x	xx	xxxxxxxx	!00001122	xxx	VBLK	xx	xxxxxxx
VME3	x	x	xx	xxxxxxxx	xxxxxxxx	xxx	xx	xx	xxxxxxx

Control signals

Control signals are treated individually or grouped. Grouped control signals are widely used in order to achieve a compact and easily readable presentation form. These are especially suited for state analysis by software engineers. Individual signals are suitable for revealing hardware errors, both using state and timing analysis.

Signal groups with mnemonics

Combining control signals into groups allow mnemonics to be used for e.g. cycle type, transfer size, status etc., eliminating the need for the user to remember the various bit combinations of control signals.

NOT operator

Grouping of control signals allow, with some restrictions, using a NOT operator on fields like size, status etc. This is selected by ticking the "[X] Not" button in the signal dialog box that appears when <CR> is typed in a field. This results in a '!' shown before the value/mnemonic. It is also possible to type '!' in front of the value/mnemonic.

Note: *It is not possible to use NOT together with a mnemonic that includes don't care bits (example UBYTE in the Size field). If you do so, you will get an error message.*

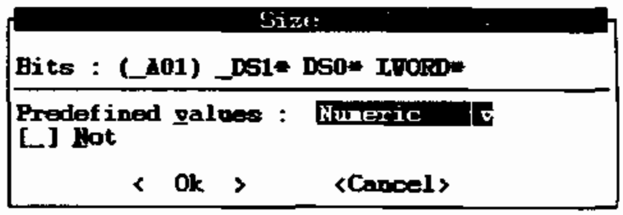
<CR>: Select mnemonic

The actual signal combinations for all the control signal groups are shown in the corresponding dialog box that appear when <CR> is typed in the actual signal field. See chapter "SIGNAL REFERENCE" for details of the defined groups.

Individual signals

If the user wants to include only one individual control signal, or a signal combination that is not listed as a mnemonic, it is always possible to type the 1/X/0 values directly into the field. To do this, place cursor on the actual signal field, then type <CR> to get the dialog box for this field in order to determine where the desired bit is placed in the field. Each signal in the group appear in the XX.X field in the sequence shown in the dialog box shown when <CR> is typed in the field.

Example: Set **DS0* = 0**, other size-bit to X (VMEbus):



Type <CR> in the Size field in the pattern line to be used, and the dialog box for this field appears. It can be seen that DS0* is the second bit in the three bit field. Then close the dialog box (with <CR> or <ESC>) and type **X0X** directly in the field.

(The address bit A01 is shown in brackets in the Size field since it a duplicate of the A01 in the address field, and is not explicitly available in the Size field, only implicitly if a mnemonic like UNALx is selected).

Note: *In the trace display, the signal groups can be "opened" into individual signals by using the command*

```
Format/Trace signals ↓ /Single
```

(The firmware 4.99 does not support individual signals for VSB. This will be provided in the future firmware versions.)

Signal polarity

Active low: * Signals which are defined as active low in the target bus are normally shown by the VBT-325 also as active low. This is then indicated by a asterisk '*' after the signal name. (Example: AS*). This means that the signal is shown as a '0' or 'low' in the trace when true.

Busses Busses that are defined as active low in the target bus (like the 8-bit data bus in SCSI, DB7-0*) are inverted and presented in the more human-readable active high form. The names of such buses are then shown by the VBT-325 without any asterisk.

Parity Parity bits that accompany such buses (like DBP* in SCSI) are then also inverted by the VBT-325 to correspond to the bus polarity. Thus, the SCSI data bus with parity is presented by the VBT-325 as DB and DBP.

Edit event patterns

The user may fill in event patterns as binary, hexadecimal or mnemonic values in the various signal fields in any of the predefined states except *AnyThing,* which is unalterable. The user may delete or insert new event patterns and signal fields, new event patterns may be given user-defined names. By inserting and/or deleting signal fields, the sequence of the signal fields may be altered.

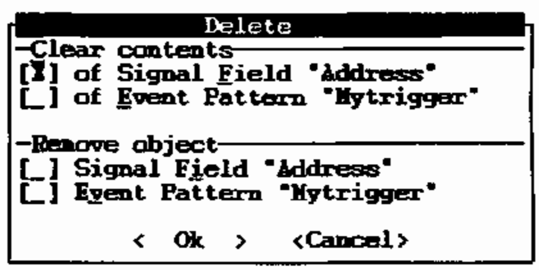
Enter value Signal group fields can be specified as a numerical value (hex or binary), and/or by a predefined mnemonic. To modify a field, type a hex or binary value directly when the cursor is placed in the actual field.

Mnemonics Signal fields can be assigned a mnemonic by typing <CR> in a field. A dialog box appears with a selection list (type '↓' to see list of mnemonics.)

Numeric If the entry called *Numeric* is chosen, the actual signal value that is represented by a selected mnemonic is shown. (E.g. in the *Size* group the mnemonic *Word* equals to the bit combination x001, that is: A01 is don't care, DS1* and DS0* is '0' and LWORD* is '1'.)

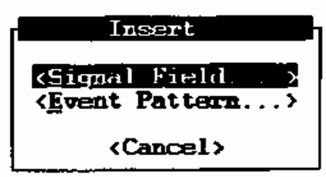
'X' - Don't Care To set a field to "Don't Care", simply type <DEL> and <CR>. (<DEL> will bring up a dialog box that has the cursor placed at "Clear contents of field". For other delete options, see *Delete fields/states* below). Another way to set "Don't Care" in a field is to type one or more 'X's directly in the field. This allows separate parts of the field to be "Don't Care".

Delete fields/event patterns



<DEL> Typing the <DEL> key (or '^BS' i.e. 'Ctrl-Backspace') when the cursor is placed in a signal group field will cause the DELETE dialog box to appear. This allows you to clear the contents (i.e. set to 'X' - "Don't Care") of the actual field or the entire event pattern (line) at the cursor position. It also allows you to remove the field column or the entire event pattern (line) itself from the event patterns window.

Insert signal fields



<INS>/'^N' Typing <INS> or '^N' when the cursor is placed in a signal group field will cause the INSERT dialog box to appear. It allows you to insert a new signal field column by means of a selection list of all available signals or groups.

Insert event patterns

<INS>/'^N' A new event pattern (line) with a user-defined name can also be inserted by typing <INS> or '^N'. It is possible to have a large number of user-defined event patterns, to make it simple to take a particular pattern into use as a trigger or qualifier.

Horizontal scrolling

To see signal fields that are not present on the screen, horizontal scrolling takes place when the cursor is "moved beyond" the last signal field on the screen. However, signal fields to the left of the divider marker 'I' in the headline will not move when the cursor is tried moved beyond the screen border. This allows certain fields to stay locked on the screen, while others can be scrolled "behind" the locked part. If some signal fields are scrolled behind the locked part, the divider marker changes from 'I' to '<'. Similarly, a '>' at the rightmost position in the headline indicates that there are more fields available "outside" the screen at right.

Lock ↔ Scroll The user may decide whether fields should be locked on the screen or take part in the scrolling. For example, to move a signal field from the moving part to the locked part, type <DEL> when the cursor is placed in the column to be moved. Then move the cursor to the where the signal field is to be inserted and type <INS>, and select the proper signal field from the selection list.

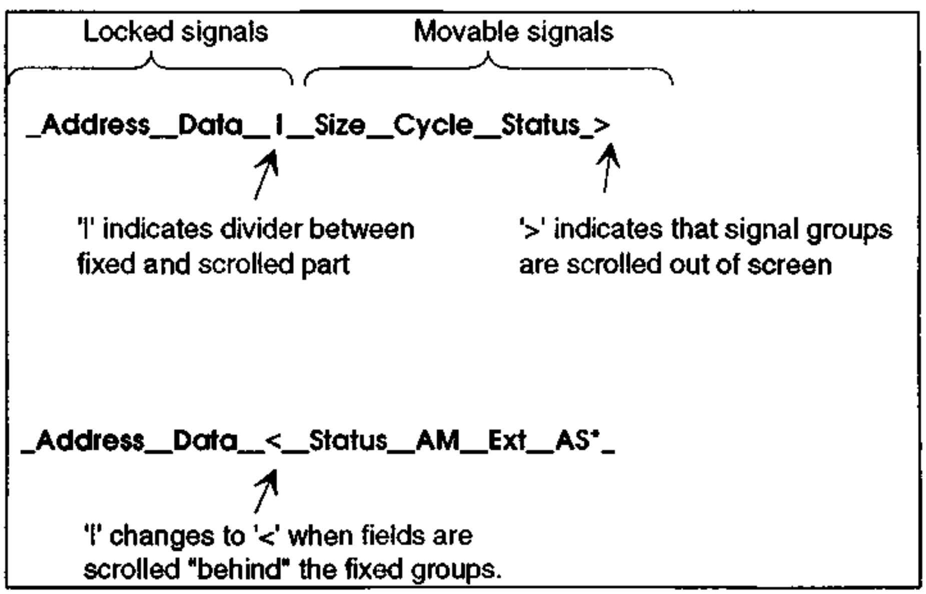
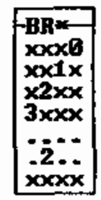


Figure 22. The signal groups can be locked or scrolled horizontally in the Event patterns window.

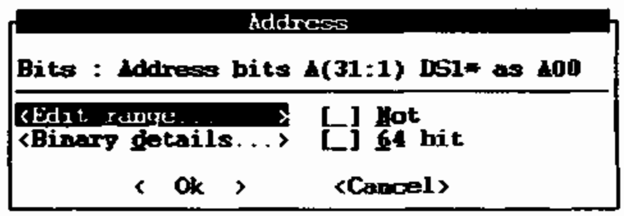
BR* and IRQ* format

As with all active low signals, a '0' is typed to indicate true and a '1' to indicate false. This is also true for the four Bus Request signals (BR3-0*) and the seven Interrupt Request signals (IRQ7-1*). However, the value shown when leaving the field is the *number* of the actual request signal if a '0' was typed (true), and a *dot* '.' if a '1' was typed (false).

The motivation for this is to get a very good readability of active request signals in the alphanumeric trace list display while keeping a consistent presentation both in the Event patterns and the Trace display.



Address/Data attributes



Typing <CR> in a address or data field in event patterns will bring up a dialog box that will give the options in the field at the cursor position.
The options may include NOT, Range, 64-bit or binary representation.

Binary details

It is possible to specify don't care bits down the bit level by using the <Binary details> option in the dialog box.

Example:

```
XXXXXXX(1XX0)
```

is true when bit 3 is one and the least significant bit is zero..

Not

NOT can be specified with address and data values by ticking the "[X] Not" button in the signal dialog box that appears when <CR> is typed in a field. This results in a '!' shown before the value/mnemonic. It is also possible to type '!' directly in front of the value/mnemonic.

Note: *There are some restrictions when you use NOT together with an address/data that includes "don't care" bits. For legal combinations see the Negate help screen. To get the help screen press H (Help). Then press N (Next) until you get the help screen 'Help - Negate'. To leave the help system press X (eXit).*

64-bit

For use in VME64 applications, the VBT-325 allows 64-bit addresses or data to be specified in the event patterns. Since VME64 is a multiplexing of the address and data lines, it is not possible to define anything in the data when a 64-bit address is specified, and vice versa.

Range

It is possible to define ranges for the address and data selecting <Edit range> in the dialog box, or simply by typing a hyphen '-' after the first (lower) value (like 12345678-9ABCDEF0).

64-bit range For the VME address, it is also possible to set ranges on 64-bit data (VME64). In this case, only the lower portion of the 64-bit boundaries are shown, with some dots ... indicating this. To see the entire value, type <CR> and do <Edit range>.

The boundary values themselves are included in the range.

Outside range using NOT ('!')

By combining a range with the NOT operator (see below), an "outside range" criteria is achieved.

Example:

```
!00100000-00200000
```

is true for all values less than 100000 and greater than 200000.

Sequencer

The Sequencer defines how the output of the event comparators shall be used to form triggers, store qualifiers, count conditions etc. The Sequencer program allows event patterns to be combined sequentially, using multiple levels of *If...Elsif...Else* statements, or combinatorially using NOT/OR/AND operators. The sequencer also allow dynamic switching of sampling mode between state or timing sampling. Although not dynamically changeable, the Trigger Position is also defined in the Sequencer program as a parameter to the Trigger statement, so that all information regarding the trace capturing and stopping is available in one place.

```
                         VME Sequencer
  1.a: Sampling in STATE mode
  1.b: Store (ALL)
  1.c: If (VME0) then
  1.d:   Trigger at 75% of trace
  2.a:   Store (VME0+VME1)
  2.b:   Delay 200 ns then if (ANYTHING) then
  3.a:     Count 4 occurences of (VME2) then
> 3.b:       Halt
Ok. <F2=Menu> <F6=Nxt wnd> <CR=Edit> <DEL=Delete> <INS=Insert>
```

Figure 23. The Sequencer program defines among others the Sampling Mode, the Trigger Sequencer, and the Trigger Position.

Actions The sequencer is a state machine which can be in one of 16 possible states. In each state, a number of **actions** can be defined to take place, like:

Sampling in STATE or TIMING mode,
Store only samples satisfying given event patterns,
Count occurrences of given event patterns,
Delay a certain time.

Transitions Actions may take place as a function of event pattern match and next state number. (However, *Sampling* and *Delay* will only be a function of current state number as a self-imposed restriction.)

In addition to actions, each state may lead to **transitions**, i.e.:

Goto another state in the sequencer,
Trigger the analyzer.
Halt the sampling and show the trace

Transitions may occur as a function of event pattern match (*If-Elsif-Else* tests), counter carry, delay carry and next state number.

General structure of one state

The general structure and capabilities of one *state* in the Sequencer program is as shown below.

(**Bold** text indicates operators, CAPs indicates parameters, bar "|" evt. in "[ ]" indicates alternatives):

```
Sampling in [STATE|TIMING] mode
Store (EVENT EXPRESSION)
[If|Count N occurrences of|Delay TIME then if] (EVENT
EXPRESSION) then
   Trigger|Goto N|Halt*
          :
Elsif (EVENT EXPRESSION) then
   Trigger|Goto N|Halt*
          :
Else
   Trigger|Goto N|Halt*
          :
```

- The colon ':' indicates that other states of the same construction can be entered in this location.
- Multiple branch conditions are possible by combining an *If*, *Count* or *Delay* statement followed by a number of *Elsif*, optionally ending with an *Else*. Note that *Count* and *Delay* only may come as an alternative to the *If*, and that it is still allowed to use *Elsif* and *Else* after *Count* and *Delay*.
- *) Halt may only be used alone.

Numbering Each state is numbered sequentially 1a, 1b, 1c etc. The next state will be 2a, 2b etc. This numbering is particularly important for the *Goto* statement. *Goto N* always means goto Na (Goto 5 means goto 5a etc). Also see the section **Goto** later in this chapter.

Single event and Sequencer modes

Single Event mode

The Sequencer program is initially set to a simple, default type program where the Sampling mode is set to *STATE* mode (synchronous sampling), the Store parameter is *ALL* samples, the Trigger position is *START* of trace, and the trigger is a single event pattern, the one pointed to with the cursor in the Event Patterns window. This is called "Single Event mode" and is in many cases sufficient for narrowing down on a specific cycle (event) on the bus.

To change the trigger to another event pattern, simply move the cursor up or down to the desired line in the Event Patterns window. (If the window is not entered already, enter the Event Patterns window by 'Edit->Event Patterns', or type <F6> or '^W' until the cursor is in this window.) The sampling mode and the trigger position can be changed from the `'Edit'` menu.

Sequencer mode

If a more complex trigger, store qualifier or count/delay is needed, you may edit the Sequencer program by the `'Edit/Sequencer'` command. Upon doing this, a dialog box appears that asks the user to confirm whether he really wants to enter the sequencer.

As soon as the sequencer window is entered, the tracer is forced into the "Sequencer Mode". In this mode, there is no "interlock" with the Event Patterns window, everything is fully controlled within the Sequencer program.

In order to return to Single Event mode, simply type <DEL> and select the "Single Event Mode" option.

Event Expression

The `EVENT EXPRESSION` is one or more `EVENT` names as defined in the Event Pattern Window in a combinatorial expression like this:

```
!A*!B*!C*!D + !A*!B*!C*!D + + .... !A*!B*!C*!D
```

This may also be written with brackets, as shown below:

```
A*(B+C)+A*B*C*(!D+!B)
```

A,B,C,D are event names, ! is an optional NOT operator, + means OR, * means AND. The term EVENT EXPRESSION is used throughout this document wherever events are used as parameters. Rules for how to resolve logical expressions with a combination of operators are given in the section "Logical operators" on page 52.

Sequencer Notation

UPPER/lower case

Parametric keywords are shown in UPPER CASE letters for better visibility, like STATE, 50MHz TIMING (except the z in MHz!) and START, MIDDLE, END.

Event names both predefined (VME0 etc.) and user-defined are shown in UPPER CASE (as in Event Patterns window).

Operators are shown with initial CAPs, otherwise lower case (e.g. *Store*, *Trigger* etc.)

Fill-in words like *in*, *of* etc. are shown in lower case.

Use of brackets

Brackets are used to indicate that fields are expandable, like event expressions where event terms can be expanded with a logical expression like '+', '*', '!'.

State and Line numbers

Each line in the sequencer has a number consisting of state number and line within state as a lower-case letter, separated by a dot '.'. (e.g. 1.a etc.). Two letters (e.g. 1.aa etc.) are used if more than 26 lines used. Line numbers are used in the sequencer except when in Single Event Mode.

Indents

Indents are used after *If, Count, Delay, Elsif and Else* statements. Example:

```
2.a:  If (VME2) then
2.b:     Trigger ...
3.a:     Sampling ...
```

See "Indent control" on page 54, and examples on page 58.

Input indicators

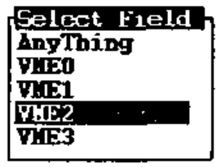
An arrow '>' at the left of the line will indicate the current active line. The cursor is shown in the leftmost column of the parameter that will be edited when <CR> is entered. During sampling the arrow '>' will indicate current active state. It will always point to the first line in each state. All line numbers in the state will also be displayed in inverse video.

Change event

To select a specific event pattern in the Sequencer program, for example in an 'If' statement, type <CR> when the cursor is at the correct line to "activate" this line.

<CR> Opens and Close event expression editing.

'↓' Then the event pattern name will be highlighted with a 'Iv' to indicate that a pull down selection list of the available "states" can be given. Then type '↓' to open the list and select a name (with <CR>) from the list. All the defined event patterns from the Event Patterns window will appear in this list.



← / → The cursor left and right key moves the cursor left and right in the event expression.

<HOME> Moves the cursor to the leftmost column of the event expression.

<END> Moves the cursor to the right of the rightmost column of the event expression.

<ESC> Cancels all changes made to the event expression after the editing was opened with <CR>.

Logical operators The sequencer allows use of the logical operators NOT, OR and AND in `EVENT EXPRESSION, i.e.` wherever an event is used.

```
Syntax:    !   means  NOT
           +   means  OR
           *   means  AND
```

Evaluation of expressions like A + B * C is done like A + (B*C).

The logical operators cannot be used together with predefined expressions like ALL, ANYTHING, NOTHING.

'(' / ')' Left and right brackets can be used in the expression like (A+B)*C.

<INS> Inserts a new event name into the expression. Change it with the cursor down arrow as described above.

<DEL> Deletes the event, the operator, or bracket in the position where the cursor is located.

Max. four different event names

Note that it is not possible to use more than four different event names (VME0, VME1 etc.) in the Sequencer program at the same time. If you try to select a fifth, all the occurrences of the event name you replace will be changed accordingly all places in the program (after a warning message).

Modify the sequencer program

Arrow keys To modify the Sequencer program, use the arrow keys to place the cursor at the desired position. Up/down moves the cursor to the previous/next statement, and left/right moves the cursor between the editable parameters of a statement.

<TAB> The tabulator moves the cursor to the next editable parameter. If no more parameters exist in the current line the cursor is moved to the next line.

Home/End The Home key moves the cursor to the first statement in the sequencer program, and End to the last statement.

<INS> / <DEL> The <INS> key will bring up a selection list of the possible choices of operators at the cursor position. The <DEL> key deletes the current statement, sub-state, reset to default program, or set Single Event Mode.

<CR> Opens and Closes the editing of the statement parameter at the cursor position.

CTRL-O With CTRL-O you can undo your last editing.

The user will never be allowed to insert any illegal statements in the Sequencer program. If he does, he will get an error message.

Operators

The following is an explanation of the Sequencer operators.

Sampling

The *Sampling* operator is used to specify sampling mode. The first line in the sequencer program must always be a *Sampling* line, so this line cannot be deleted. The sampling mode can be changed dynamically by entering a new *Sampling* statement inside the sequencer program.

```
Syntax:      Sampling in .... mode
Parameters:  STATE
             50MHz TIMING
             25MHz TIMING
```

```
6.25MHz TIMING
1.56MHz TIMING
781.3KHz TIMING
390.6KHz TIMING
195.3KHz TIMING
97.7KHz TIMING
```

Store

The *Store* operator is used to achieve "filtering" of the captured samples. A *Store* expression is implicitly valid for the rest of the sequencer program, until superseded by a new *Store* condition. The second line in the sequencer program must always be a *Store* condition, so this line cannot be deleted. The predefined expressions ALL and NOTHING are available as parameters in addition to an `EVENT EXPRESSION`.

ALL and NOTHING is programmed a special way in the sequencer hardware, so that it does not consume a separate event (out of the four usable).

```
Syntax:      Store (Parameters)

Parameters:  EVENT EXPRESSION
             ALL
             NOTHING
```

If/Elsif/Else

If / Elsif / Else statements may be used to control the branching of the sequencer program. Multiple *Elsif* is possible, limited only by the number of possible EVENT EXPRESSIONs. Both *Elsif* and *Else* is optional after an *If*.

Note: *An "If-Elsif" sequence without an "Else" will always repeat itself if none of the conditions were met, so that a statement like*

```
Else
   Goto Current state
```

can be considered as an implicit closing statement.

The predefined expression ANYTHING is available as parameter in addition to an `EVENT EXPRESSION`. ANYTHING is programmed a special way in the sequencer hardware, so that it does not consume a separate event (out of the four usable).

```
Syntax:      If (EVENT EXPRESSION) then
                Elsif (EVENT EXPRESSION) then
                Else
```

Indent control When multiple *If* states exist ahead of a point in the sequencer program where an *Elsif* or *Else* is to be inserted, there is a need to determine which *If* state the *Elsif* or *Else* shall belong to. This is achieved by a dialog box, where the user is asked to enter which state number the inserted *Elsif* or *Else* shall belong to. Refer to **Examples** later in this section.

Goto

The *Goto* statement moves the execution of the sequencer program to the beginning of another state. *Goto 1* will function as a restart of the Sequencer program.

Syntax `Goto N`

(N=2 will point to state 2, i.e. line 2.a, N=4 to line 4.a and so on.)

Warning: *"Goto" cannot be used to repeat "Count" or "Delay" statements, since the counters are not reloadable during sampling.*

Count

Count controls counters that can be used to count occurrences of specific cycles/events on the target bus. If a count statement is used, the Sequencer program will not advance until the specified number of cycles occurs that matches the event pattern attached to the 'Count' statement.

Syntax: `Count N occurrences of (EVENT EXPRESSION) then`

Parameters:
```
N = 2 through 1048576
EVENT EXPRESSION
ANYTHING
```

If you intend to use *Count 1*, you should instead use the *If* statement.

Delay

Delay controls timers that can be used to delay a certain time before the sequencer program is allowed to advance to another state.

Syntax: `Delay N <unit> then if (EVENT EXPRESSION) then`

Parameters:
```
N = 1 through 999
unit: ns/us/ms
```

The delay time can be min = 40ns, and max = 335ms. The value is automatically truncated to the selected clock period, which can be 40, 80, 160 or 320ns.

Note: *When state 1 contains a delay statement, the delay counter starts to count between 500-900 us before the sampling is started. This means that delays less than this time have no meaning in state 1. The delay counter can be synchronized by putting an "If (ANYTHING) then" before the first delay. The delay counter will then start to count when the first sample occurs on the bus after the sampling is started.*

A construction like "Delay ... Elsif" can be used to exit a delay interval on a certain condition, before the delay time expires.

Note: *A sample is required after the delay time is counted down, before the sequencer will proceed to the next state, or a trigger will occur.*

Trigger

The *'Trigger'* operator determines where in the sequencer program the trigger should be. It is possible to program a *'Trigger'* statement at different places in the Sequencer program, but only one of these will actually lead to a trigger, depending on the progress through the specified trigger sequence.

The Trigger *position* is a parameter to the trigger statement. If multiple trigger statements exist, the trigger position will be kept the same throughout the sequencer, so modifying this one place will automatically update the others accordingly.

```
Syntax:      Trigger at .... of trace

Parameters:  START
             25%
             MIDDLE
             75%
             END
```

As the parameter for *Trigger* must be the same throughout the sequencer program, *Halt* must be used instead of *Trigger End* if *Trigger* already has been used with one of the other parameters. You should, however, use *Trigger End* where possible.

Halt

The *'Halt'* operator causes the tracer to halt and display the trace.

```
Syntax:      Halt
```

Implicit actions and transitions

The sequencer is no programming language, but a compact practical way of controlling the operation of a logic analyzer. Thus, to minimize the need for user programming, there are a number of implicit actions in the sequencer that gives the user the desired results in the absence of explicit commands:

- A *Sampling* expression is implicitly valid for all subsequent states in the sequencer program, until superseded by a new *Sampling* condition.
- A *Store* expression is implicitly valid for all subsequent states in the sequencer program, until superseded by a new *Store* condition.

- The sample causing a *Trigger* is always stored.
- If no states follow an *If .. then, Trigger* in the sequencer program, like in the default program, an implicit jump to a state where the prevailing store condition is repeated takes place. This is to avoid storing both the specified store condition and the trigger condition if the trigger condition should occur again (according to the above rule saying that trigger samples are stored.)
- An implicit *Else Goto "current state"* is always present after an "If-Elsif" sequence if no "Else" is specified, so that the If-test will always repeat itself if none of the conditions were met.
- *Goto "next state"* is implicit after a *then* or after an *Else*, where *next state* is the state belonging to the next line containing an *If*.

Loose and Tight sequence

A **loose sequence** is defined as a sequence of events (bus cycles) that simply occur sequentially, without any constraints on other events appearing in between. For example, the events A,B,C and D come in a loose sequence if they occur mixed with the events X and Y like

A - B - X - C - Y - D.

The following sequencer program will trigger on a loose sequence of the events A,B,C and D:

```
1.a: If (A) then
2.a     If (B) then
3.a        If (C) then
4.a           If (D) then
4.b              Trigger at ..
```

On the other hand, a **tight sequence** is defined as a sequence of events (bus cycles) that occur **without** any other event appearing in between, strictly like

A - B - C - D .

The trigger sequencer on the VBT-325 can be programmed to trigger on tight sequences by using *'Goto 1'* and *'Goto 2'* operators as shown on the next page.

```
1.a: If (A) then
2.a:    If (B) then
3.a:       If (C) then
4.a:          If (D) then
```

```
4.b:            Trigger at ..
4.c:         Elsif (A) then
4.d:            Goto 2
4.e:         Else
4.f:            Goto 1
3.b:      Elsif (A) then
3.c:           Goto 2
3.d:      Else
3.e:         Goto 1
2.b:   Elsif (A) then
2.c:      Goto 2
2.d:   Else
2.e:      Goto 1
```

If the '*Else Goto 1*' terms were missing, the trigger would be reached even if a cycle X occurred in between the A,B,C or D cycles (a loose sequence).

The '*Goto 2*' statements are necessary to trigger also if the actual sequence is partially fulfilled, and then immediately followed by the sought sequence, like A-B-C-A-B-C-D. If the '*Goto 2*' were missing, the second A would give a '*Goto* ', starting a new search for A, but this time the A does not come again before the B-C-D.

Note that it is not necessary to include an '*Else Goto 1*' at the outer '*If*' level (bottom), because of the implicit '*Else Goto current*' in an '*If*' statement not ending with en '*Else*'.

Sequencer examples

Count, Delay and Switch sampling mode

The sequencer program below will count 10 occurrences of VME0 or VME1, then cause a trigger if VME2 is found, then switch to 50MHz timing sampling for 760ns after the trigger, for detailed review of the trigger cycle, and then revert to state sampling.

```
    1.a: Sampling in STATE mode
    1.b: Store (ALL)
    1.c: Count 10 of (VME0 + VME1) then
    2.a:    If (VME2) then
    2.b:       Trigger at START of trace
=>  3.a:       Sampling in 50MHz TIMING mode
    3.b:       Delay 760ns then if (ANYTHING) then
    4.a:          Sampling in STATE mode
```

```
VMETRO VBT-325C                      VSB Bus Analyzer                    Setup: Default
  Trace  Edit  Target  Statistics  Setups  Utilities  Help
                                 VSB Event Patterns
            SPACE DS*      AD(31:0)        SIZE ASACK* Cycle Status VMEtrg
AnyThing     xx    x          xxxxxxxx       xx   xx     xxx   xx      x
Address      xx    ADDR  00810000-00813FFF  WORD  xx     READ  xx      x
Wrongaddr    xx    ADDR :00810000-00813FFFxx      xx     xxx   xx      x
Data         xx    DTA        xxxxxxxx      WORD  xx     READ  xx      x
```

Figure 26. Above is the event setting for the Sequenser example on trigger and store on data in address range on VSB. Below is the sequencer program.

```
1.a: Sampling in STATE mode
1.b: Store (Address)
1.c: If (Address) then
2.a:    Store (Address+Data)
2.b:    If (Wrongaddr) then
2.c:       Goto 1
2.d:    Else
2.e:       Trigger at START of trace
3.a:       Store (Address+Data)
3.b:       If (Wrongaddr) then
4.a:          Store (Address)
4.b:          If (Address) then
4.c:             Goto 3
```

VSB sequencer examples

```
VMETRO VBT-325C          -         VSB Bus Analyzer                    Setup: Default
 Trace  Edit  Target  Statistics  Setups  Utilities  Help
                               VSB Event Patterns
            SPACE-DS*------AD(31:0)-----|SIZE-ASACK*-Cycle-Status-VMEtrg
AnyThing    xx     x       xxxxxxxx      xx    xx    xxx    xx     x
Address     xx     ADDR 00810000-00813FFF WORD  xx    READ   xx     x
Data        xx     DTA     xxxxxxxx      WORD  xx    READ   xx     x

                                VSB Sequencer
> 1.a: Sampling in STATE mode
  1.b: Store (ALL)
  1.c: If (Address) then
  2.a:   If (Data) then
  2.b:     Trigger at START of trace
  2.c:   Else
  2.d:     Goto 1
```

Figure 24. The example above shows trigger on data in address range on VSB.

```
VMETRO VBT-325C                    VSB Bus Analyzer                    Setup: Default
 Trace  Edit  Target  Statistics  Setups  Utilities  Help
                               VSB Event Patterns
            SPACE-DS*------AD(31:0)-----|SIZE-ASACK*-Cycle-Status-VMEtrg
AnyThing    xx     x       xxxxxxxx      xx    xx    xxx    xx     x
Address     xx     ADDR 00810000-00813FFF WORD  xx    READ   xx     x
Wrongaddr   xx     ADDR !00810000-00813FFFxx    xx    xxx    xx     x
Data        xx     DTA     xxxxxxxx      WORD  xx    READ   xx     x

                                VSB Sequencer
  1.a: Sampling in STATE mode
  1.b: Store (Address)
  1.c: If (Address) then
  2.a:   Store (Address+Data)
  2.b:   If (Wrongaddr) then
> 2.c:     Goto 1

Ok.  <F2=Menu> <F6=Next wnd> <CR=Edit> <DEL=Delete> <INS=Insert>
```

Figure 25. The example above shows store on data in address range on VSB (no trigger).

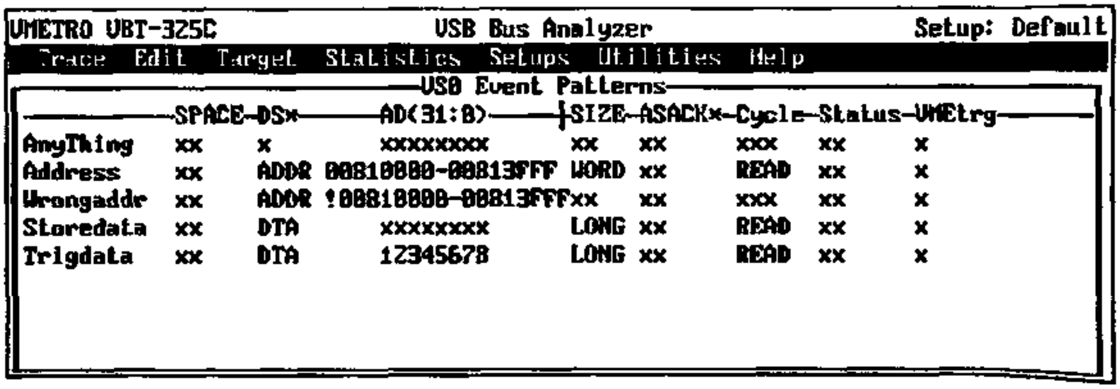


Figure 27. Above is the event setting for the example on store on data in address range on VSB, trigger on other data. Below is the sequencer program..

```
1.a: Sampling in STATE mode
1.b: Store (Address)
1.c: If (Address) then
2.a:    Store (Address+Storedata)
2.b:    If (Wrongaddr) then
2.c:       Goto 1
2.d:    Elsif (Trigdata)
2.e:       Trigger at MIDDLE of trace
3.a:       Store (Address+Storedata)
3.b:       If (Wrongaddr) then
4.a:          Store (Address)
4.b:          If (Address) then
4.c:             Goto 3
```

Cross-triggering

Cross-triggering is possible between the VME and P2 analyzers on the VBT-325, as well as to/from the TIM200-PB 200MHz Timing Analyzer piggyback module. This is accomplished using the cross-trigger signals **VMEtrg**, **P2trg** and **TIMtrg**, refer t.› .he figure below.

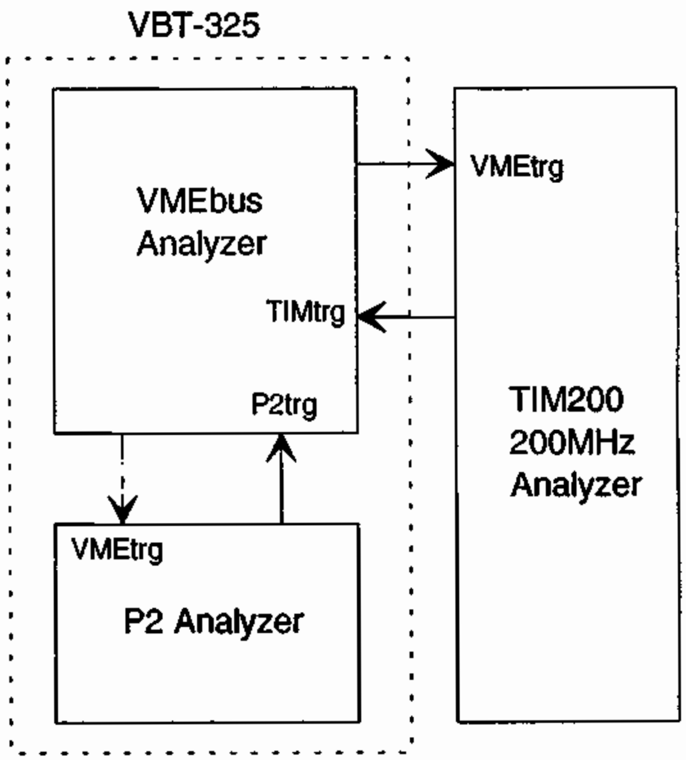


Figure 28. Cross triggering is possible by means of trigger signals between the various sub-analyzers.

Example: Cross-trigger VSB from VME

To specify that the P2 analyzer (e.g. VSB) should trigger when the VME analyzer triggers, go to 'Target/VSB', and do 'Edit/Event Patterns', insert the cross-trigger signal by typing <INS> or '^N' and select **VMEtrg** from the selection list. Then assign a value to **VMEtrg** by typing <CR> and select 'TRIG'. Start both the VME and VSB analyzer at the same time with 'Trace/Run Multiple'.

Trace Display

The data in the trace buffer can be presented either as one or more alphanumeric trace list windows, or as one or more waveform windows.

<PF2> or '^E' To enter a trace window with the cursor, type <PF2> (on VT100 etc.), <F2> (on PCs) or '^E'. Whenever the cursor is inside a trace window, the same <F2> / <PF2> / '^E' will bring cursor back to the menu.

<F6> or '^W' To move the cursor to the next [editable] window, type <F6> or '^W'. This is especially handy to switch quickly between multiple trace windows (see "Add window" below.)

'^U' = PageUp /'^D' = PageDown

To move page by page in the trace buffer, type '^U' for PageUp and '^D' for PageDown when the cursor is in the trace window.

Alphanumeric trace list

The alphanumeric trace list shows the samples collected in the trace buffer as a list of binary or hex values for each signal group.

The alphanumeric trace list presentation form can be selected independently of the selected sampling mode, although, it makes more sense to display samples collected with state (synchronous) sampling in the trace list form, which is also the default form.

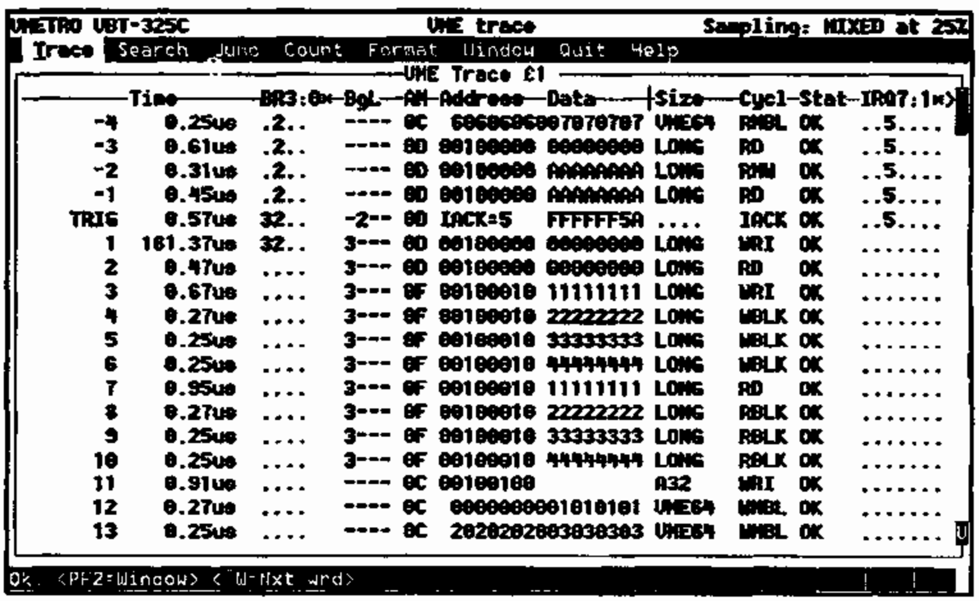


Figure 29. Example of alphanumeric VMEbus trace list.

There are two different ways of presenting the control signals in the trace list, either grouped like "Size"/"Cycle"/"Status", or as individual (single) signals like "DS0*", "DTACK*" etc. The grouped presentation form is best suited for state (synchronous) sampling, with the values shown as mnemonics, while the individual signal presentation is recommended if timing (asynchronous) sampling is used together with a trace list, with values shown as ones and zeros. (Refer to description of the command `'Format/Trace Signals'` groups in the chapter "COMMANDS REFERENCE, Trace Display menu".)

Signal selection

The default VBT-325 trace display will show a selection of signals and signal groups in the trace list that are most relevant for the selected target bus. However, since most busses will contain more signals/groups than there is space for on the screen, the user can see other signals by horizontal scrolling (just as in the Event Pattern window). Similarly, the user can also insert or delete signals/groups, as well as reorganize the signals/groups by using the <INS>/<DEL> keys.

<DEL> / '^BS' Typing <DEL> when the cursor is placed in a signal field will cause the DELETE dialog box to appear. This allows you to delete the signal field column from the trace list window. This causes the signal/field to appear in the list of unused signals presented when INSERT is performed.

<INS> / '^N' Typing <INS> when the cursor is placed in a signal field will cause the INSERT dialog box to appear. This allows you to insert a new signal group as a field column in the trace list window.

Horizontal scrolling

To see signal fields that are not present on the screen, horizontal scrolling takes place when the cursor is "moved beyond" the last signal field on the screen. However, signal fields to the left of the divider marker '|' in the headline will not move when the cursor is moved beyond the screen border. This allows certain fields to stay locked on the screen, while others can be scrolled "behind" the locked part. If some signal fields are scrolled behind the locked part, the divider marker changes from '|' to '<'. Similarly, a '>' at the rightmost position in the headline indicates that there are more fields available "outside" the screen at right. This is illustrated under Event Patterns.

Waveforms

Waveforms are provided to show the logic level of individual signals graphically as a function of time. This is particularly useful to show timing relations between different signals for hardware analysis. Busses may be presented as a "ladder", with the value of the bus at the cursor position shown in hex value within the window. The ladder will contain a "step" when the bus changes value. The individual signals in a bus can also be selected, see below.

Note that waveforms is used as the default presentation form when Timing (asynchronous) sampling is used.

Hex/Binary value

The signal value at the cursor position is always shown directly under each visible signal/field name in the left column in the waveforms display. This allows for convenient recognition and location of cycles in the trace buffer.

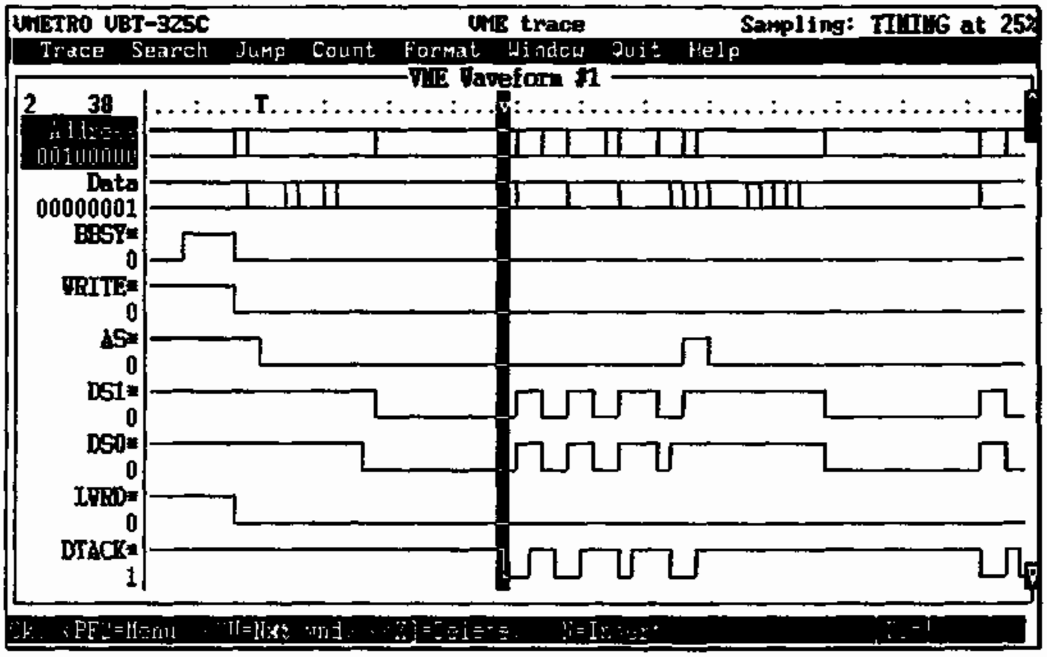


Figure 30. Example of VMEbus waveform display.

Signal selection

Any signal can be selected in the waveform display by using <DEL> and <INS> in the same manner previously described for the Trace list presentation.

Zooming (Time/Div)

The default "time per division" setting is one sample per screen column. This can be changed using the command

```
Format / Time/div
```

to allow a more or less detailed view of the signals in waveform mode.

1,2,4,8.... only

Note: *Only powers of two are legal entries (1,2,4,8...etc.) for the time per division. Other values will give an error message.*

Add window

A powerful feature of the VBT-325 firmware is the ability to add other trace windows, either as alphanumeric trace lists, or as waveform displays. The windows may be considered completely independent from each other. Different parts of a trace may be shown in each window, and scrolling in one window takes place without affecting another etc.

Refer to the Window command in the COMMANDS REFERENCE chapter later in this manual, and trace examples in the following chapter to see how different trace windows can be used.

Trace Dump to PC/Host

Trace buffer data can be dumped to a file on a PC. The file format contains a header with Target type, Sampling mode, Trigger position, Trigger line number etc, so that the file can be reviewed exactly as captured. You can use the PC Simulator (see below), or load the file back into the VBT-325. Partial traces can also be dumped, and packing can be used to reduce transfer time. A user comment can be added to the file, this can be read by using the DOS command `type <filename>`.

Simulator used for trace review

The simulator, which runs under DOS, is a true representation of the firmware of the VBT-325. This is supplied on a diskette free of charge with each shipped VBT-325 product. The simulator is extremely useful for reviewing trace files captured by the analyzer and dumped to a file on a PC. (Ref. ch. SIMULATOR at the end of this manual.)

(For users of the STIM200-PB, the simulator may be very useful for creating patterns locally on the PC, and when a good pattern file is made, it can be loaded into the STIM200-PB itself for execution. Ref. STIM200-PB User's Manual.)

Statistics

The VBT-325 provides powerful statistical measurements of target bus activity. Firmware Version 4.99.120 implements three functions, including:

- Event Counting, incl. Bus Level Histogram (counter driven)
- Bus Utilization Histogram (counter driven).
- Bus Transfer Rate Histogram (trace driven).

Counter driven

The most accurate method to collect data for the histograms is the counter driven. This method is based on hardware counters which are programmed to increment on certain VMEbus events. There are eight 20 bit hardware counters which count the occurrences of each of the events or the granted bus levels. In addition, there is a ninth 20-bit counter counting the total number of samples taken. Every time this counter reaches its maximum count, user programmable up to 1 million samples, the eight counters are disabled; their values read, and immediately re-enabled to resume counting while the histograms are computed and displayed. This method ensures that only a minimal amount of micro processor activity is missed from the measurement between each update of the histograms, giving a capture ratio of close to 100%. This mode is therefore called Real-Time Statistics.

Trace driven

The other method is based on taking a series of traces, each with 32 K samples. This gives greater flexibility for what to present, since it is all up to the software to process the collected data in the trace buffer. However, only a fraction of all bus activity is covered, so in order to give a true picture of the behaviour of the target bus, this mode should be left running for a while to collect a reasonable number of samples. The necessary time depends on the size and nature of the bus traffic to be analyzed. For small, repetitive programs it will be sufficient with only a few traces, while larger programs may require a substantial number of traces to give a correct reading. The total number of samples are always displayed on the screen during these measurements.

Note: *Programming of the current Event Patterns is used for the Event counting statistics only. All the other statistic modes overrules the current event- and sequencer programming and the trigger position, except when the "Start on Trigger" option is used for Bus Transfer Rate.*

Statistics screen

Interactive control and operation of the statistics functions is provided in a dedicated screen which may be accessed by selecting `'Statistics'` from the menu bar in the Setup screen. The user is then presented with the Statistics screen, like the one illustrated in the figure on the next page.

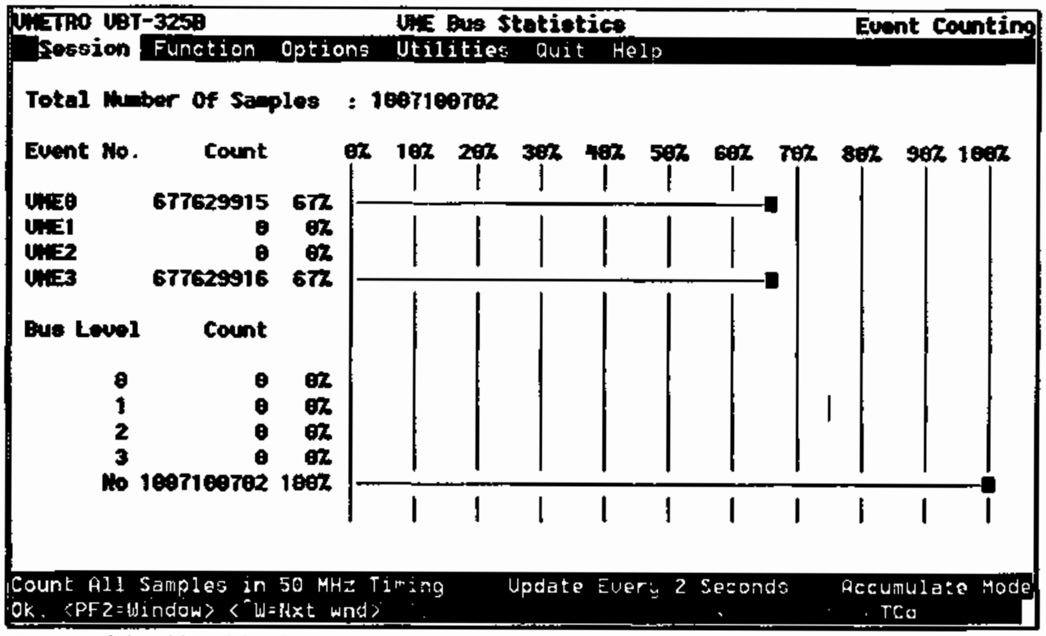


Figure 31: VBT-325 Statistics screen - Event Counting incl. Bus Level Histogram. (Note that the Target command only applies to the VBT-325C.)

The Statistics screen consists of a menu bar along the top of the screen, a window section which consumes most of the screen, and a status line along the bottom of the screen. Like the rest of the VBT-325, the interactive controls needed to operate the statistical features are located in the menu bar.

A general description of the capabilities and features of the statistical measurements is given below. For a detailed description of each of the commands in the Statistics screen, please refer to the section "COMMANDS REFERENCE" later in this manual.

Display features

Select events The four events shown in the upper part of the Statistics screen when the Event Counting function is selected are user-selectable. Any of the events as defined in the Event Patterns window in the Setup screen can be included (by the commands '`Options/Select Events`'). All other histogram elements are fixed.

Max. scale By default, the histogram diagram has 100% as the maximum horizontal scale. For better resolution of low readings, this scale can be adjusted in steps down to 30% as the max. reading (by the commands '`Options/Maximum scale`').

Bar markers Normally, the histograms are shown as a horizontal bar (line) where the end point represents the last value read from the statistics counters. A statistics session normally involves a series of counter readings, so it may be desirable to get an indication of the lowest and highest values recorded, and the average of all the counts. This is possible with the commands '`Options/Bar`

Markers/Show', which gives the user a choice of minimum, maximum and average markers on the histogram bars. The current (last read) value is indicated as a solid ball (on some terminals this is shown as a solid rectangle).

<–	Minimum marker
–a–	Average marker
–>	Maximum marker
–•–	Current value marker

Note: *Average marker is not available in "Accumulate mode" as this would be equal to current value.*

It is possible to reset the marker positions by the commands 'Options/Bar Markers/Reset', so that only subsequent counter readouts affect the marker positions.

Time History Diagram

The Time History Diagram will show the variations of the bus signals with respect to time. Select 'Options/Time History Curves', and set the display options with 'Options/Graph Display Options'.

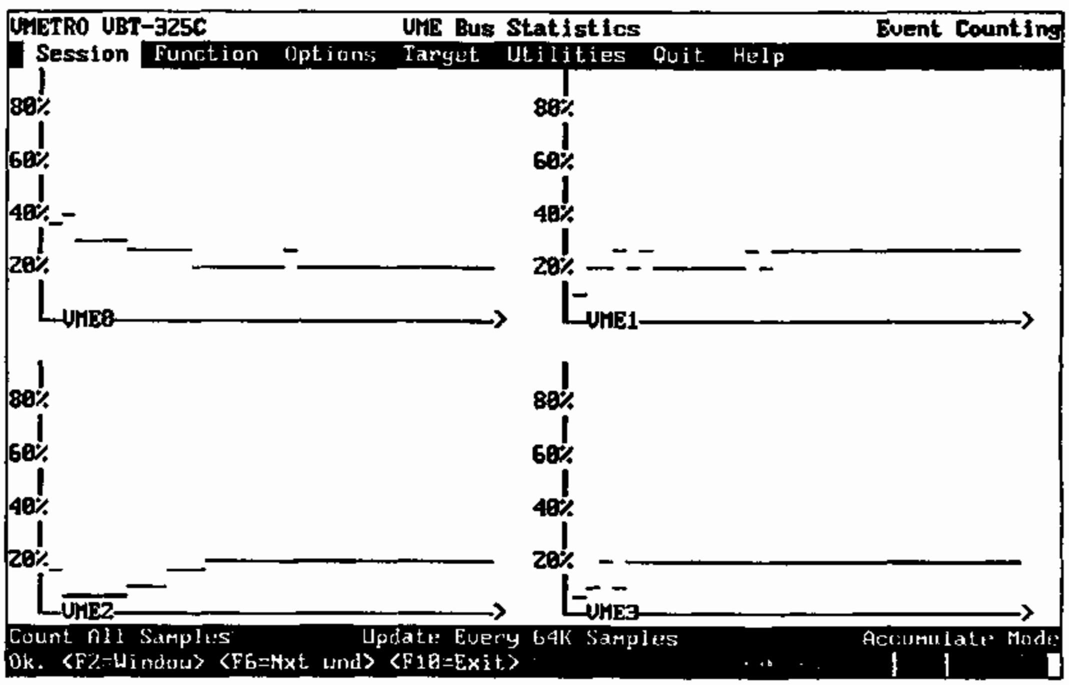


Figure 32. Time history diagram

As for the signals and their explanation, see the two previous sections **VMEbus Utilization measurement**, and **VSB Utilization measurement.**

Counters Operation

The VBT-325 implements eight 20-bit wide statistics counters for VMEbus and four 20-bit counters for the P2 bus:

- Four VMEbus Event Counters
- Four VMEbus Bus Level Counters
- Four P2 Event Counters

Each of these counters may count up to 1048576 (1M). A counter is incremented only when a bus sample matches the criteria assigned to its word recognizer. For the Event Counters, one particular event pattern is assigned to each counter, and for the Bus Level Counters, one of the four VMEbus Bus Levels is assigned to each counter. (The VMEbus Bus Levels are derived from the Bus Grant lines BG3:0*, refer to the ch. "FUNCTIONAL DESCRIPTION, Sampling methods, Bus Grant latching - BgL" for details.).

The block diagram below highlights the parts of the VBT-325 which take part in the counter-driven statistics measurements.

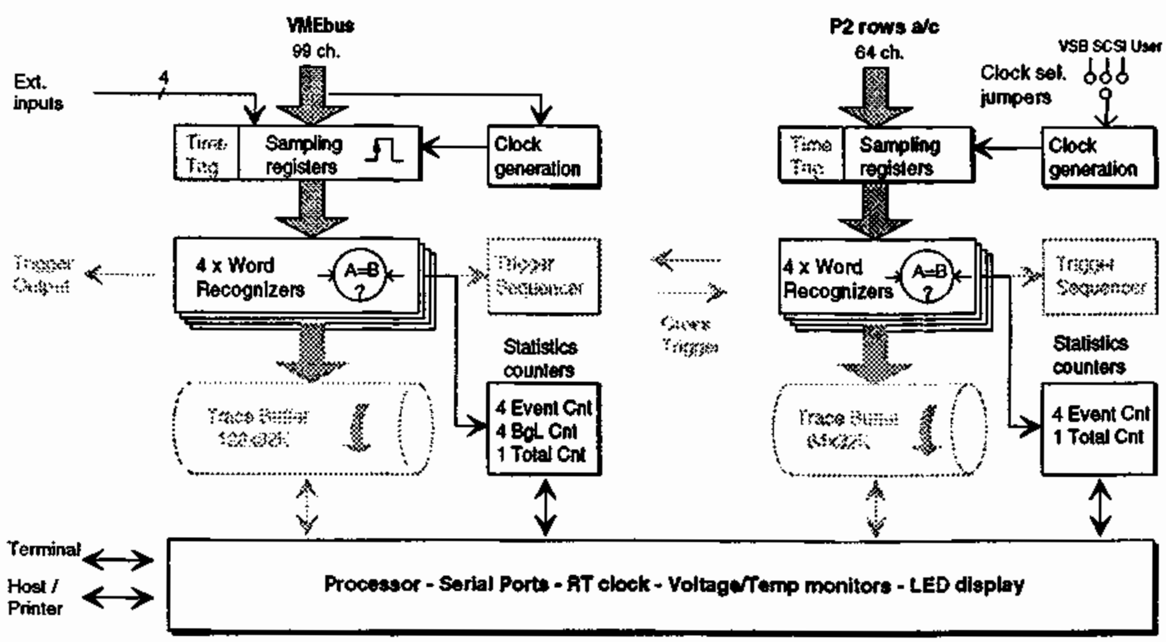


Figure 33. VBT-325 Block diagram with parts involved in counter-driven statistics highlighted.

Update rate

Sample Count

In addition to the counters mentioned above, the VMEbus and P2 parts each have a 20-bit **total sample counter** which is used to determine the histogram update rate. When this counter reaches its terminal count, the processor of the VBT-325 is interrupted and the current value of the other counters is read. The counters are then cleared and re-enabled for counting, while the processor displays the counter values in the histograms. This method ensures that the time interval during which no counting takes place is minimized (see ch. "Idle interval" below).

In order to give a suitable histogram update rate for the actual bus traffic in the system, the update rate is programmable in progressively larger intervals of samples, beginning at 1K and increasing to a max. of 1M. This is done with the command sequence `'Options/Count options/Update Every'`. Simple experimentation with this display control will assist the user to quickly determine the optimum parameter needed to acquire the maximum recording resolution for the application under test. Applications generating low bus cycle frequencies will typically require a smaller screen update parameter. As bus cycle frequency increases, the screen update parameter should be increased to prevent the utility from updating the screen unnecessarily and to reduce the effect of the "idle interval" (described below).

Time Interval

The `'Update Every Time Interval..'` feature provides control of the screen update interval by explicitly defining the time between screen updates. The time intervals starts at 1 second, and increases to a max. of 120 seconds.

Use the `'Update Every Time Interval..'` feature to get a constant refresh rate, independent of bus activity.

Accumulate/Reset mode

Reset mode

Normally, the histograms are updated with a figure which is calculated simply as a percentage of the sample count for each counter update, i.e.:

Displayed Value = (Counter reading / Update rate) * 100%

This gives a "dynamic activity indicator", showing a new "fresh" measurement at every update.

Accumulate mode

In order to provide an average value taken over a longer period of time, the range of the total samples counter may be extended in software by selecting the `'Accumulate'` mode under `'Options/Count options'`. In this mode, the displayed value is the cumulative sum of all previous counter readings shown as a percentage of the accumulated total number of samples, i.e.:

$$\text{Displayed Value} = (\Sigma \text{ Counter readings} / \text{Update rate} * N) * 100\%$$

where N is the number of updates in this session.

Which mode to choose?

Selection of the Accumulate versus Reset mode is typically driven by the total number of samples to be observed in the measurement. Measurements made with timing (asynchronous) sampling typically require the use of the Accumulate mode to yield significant results by virtue of the fact the counters reach terminal count very rapidly in response to the fixed frequency of the sampling clock. Bus cycle measurements made with the state (synchronous) sampling option may or may not require the Accumulate option to yield significant results.

Bus cycle measurements are affected by two key application specific factors: The total number of cycle operations occurring on the backplane and the frequency at which the cycles occur. The measurement of applications consisting of less than 1048576 (1M) bus cycles may be accomplished within the limits of the Reset mode of operation. This mode is often quite sufficient to support detailed characterization of new software and firmware in an isolated environment. However, characterization of applications inside fully operational system environments typically requires use of the Accumulate mode to achieve the desired measurements.

Idle interval

It is important to note that during histogram updates, there is an idle interval of appr. 224 µs when the counters are being read by the processor. During this interval, the counters are inactive, and **no bus traffic is recorded**. Normally, this idle interval is negligible, especially when high update rates and/or Reset mode is used.

Although the counters are re-enabled before any screen update takes place (which is inherently slow due to the serial line), the idle interval may influence measurements in certain applications. Especially if the application calls for accurate counting of bus cycles, one should restrict this kind of measurement to a number of cycles less than the update rate (up to 1M cycles).

Event Counting / Bus Level Histogram

The Event Counting / Bus Level Histogram, selected under `'Functions/ Event Counting'`, allows statistical measurements to be taken using both state (synchronous per bus cycle) and timing (asynchronous) sampling methods. The sampling mode is chosen under `'Options/Sampling`

Mode' in the Statistics screen. The signal state and bus cycle specifications for event based measurements may be input in the Event Pattern window in the Setup screen.

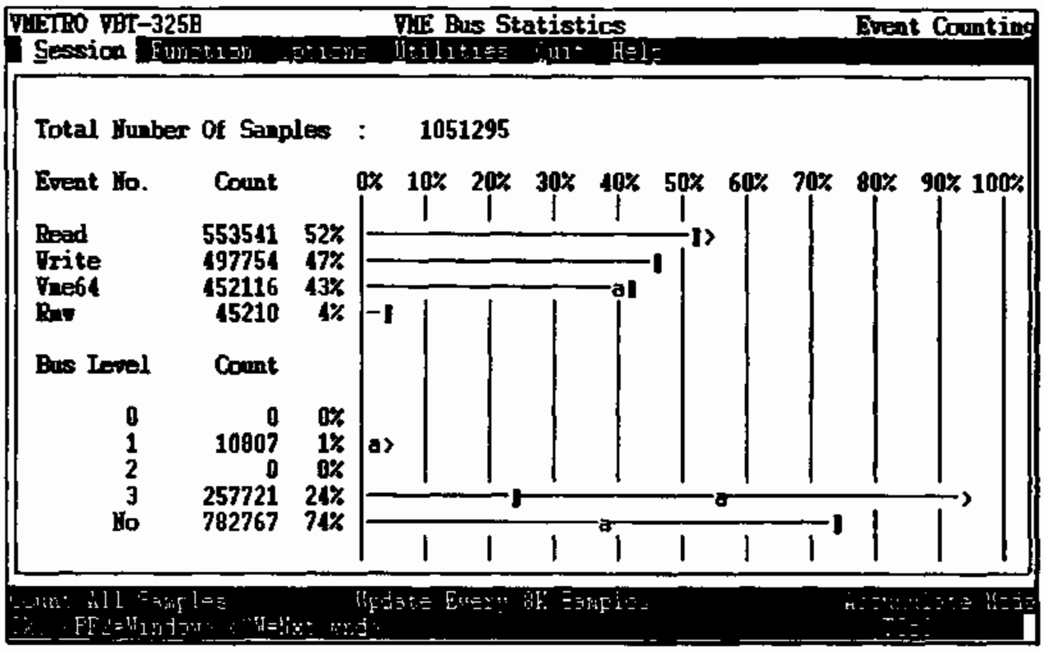


Figure 34. Histogram example when running Event Counting with max. and average bar markers enabled. Note that user-defined events are selected.

Configuration and manipulation of this function is accomplished from both the **Setup screen** of the VBT-325 and also from the **Statistics screen.** General configuration parameters, such as the sampling mode and the actual event pattern criteria which cause the statistical counters to increment in response to activity observed on the target bus, are specified through the 'Edit' command of the menu bar in the Setup screen. The basic control parameters which are used to start, stop, and restart the utility are invoked through 'Session' command of the menu bar in the Statistics screen.

Note that the "Event Counting" function is available for all supported target busses in VBT-325C. Also, any defined event can be included by using the command "Options/Select Events"

Bus Utilization Histogram

The Bus Utilization Histogram, selected under 'Functions/Bus Utilization', performs its function using **pre-determined** signal state parameters. Also, a fixed timing (asynchronous) sampling rate of 25MHz is used. As such, this mode does not require the user to provide any signal state specifications prior to initiating a measurement, meaning that configuration and manipulation of this utility is accomplished exclusively from the VBT-325 Statistics screen.

This function is available for both the VMEbus and VSB in the 4.99.120 firmware.

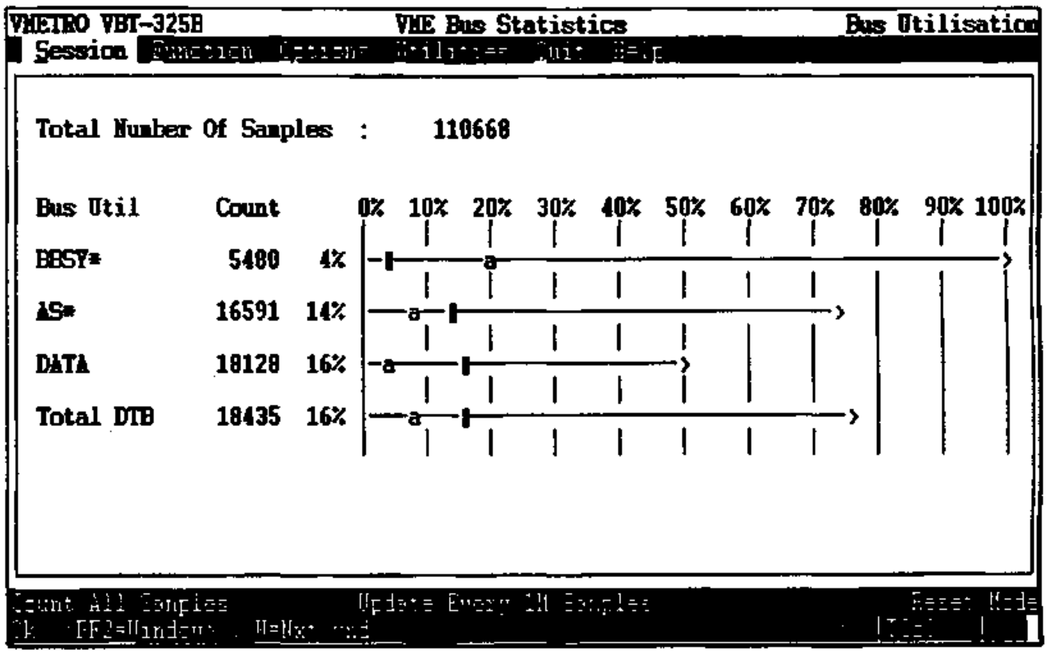


Figure 35. Histogram example when running Bus Utilization with min, max. and average bar markers enabled.

VMEbus Utilization measurements

Four histograms are provided in the VMEbus Bus Utilization function, each measuring the percentage of time when a signal or signal combination is active on the bus:

BBSY* — Indicates percentage of time when the VMEbus signal BBSY* signal is active, i.e. when there is a bus master that "owns" the bus.

AS* — Indicates percentage of time when the VMEbus address strobe, AS*, is active. This not only shows how much of the time the *address bus* is utilized, it also includes total time spent on block (BLT/MBLT) transfers and RMW cycles, since AS* is active throughout these cycle types.

DATA	The logical 'or' of DS1*, DS0*, DTACK* and BERR*, indicating percentage of time when the VMEbus *Data bus* is utilized.
Total DTB	The logical 'or' of AS*, DS1*, DS0*, DTACK* and BERR, indicating percentage of time when the *Data Transfer Bus* of the VMEbus is utilized.

VSB Utilization measurements

If you select 'Target/VSB' and then 'Function/Bus Utilization' you get the VSB Bus Statistics screen.

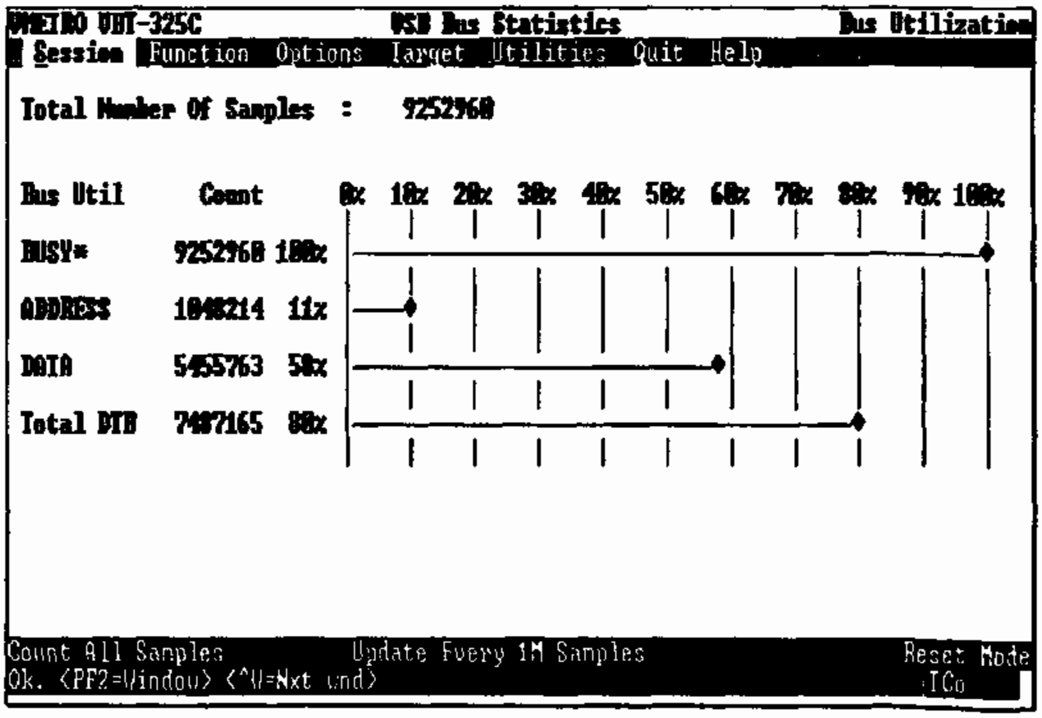


Figure 36. Histogram example when running Bus Utilization on a VSB.

Four histograms are provided, each showing the percentage of time when the following signal conditions are found:

BUSY*	BUSY*=0
ADDRESS	PAS*=0 and AC=0
DATA	DS*=0 or ACK*=0 or ERR*=0
Total DTB	PAS*=0 or DS*=0 or ACK*=0 or ERR*=0

Bus Transfer Rate Histogram

The Bus transfer rate statistics takes a series of samples with state (synchronous) sampling and calculates the transfer rate in MTransfers/Sec and Mbytes/Sec.

Note that the tracer does not collect samples in a period between each trace when the collected data is being processed. You must select whether the counters should accumulate the numbers, or reset the numbers for each trace.

You get this function by selecting `Statistics` from the main window, and then select `Function/Bus Transfer Rate`.

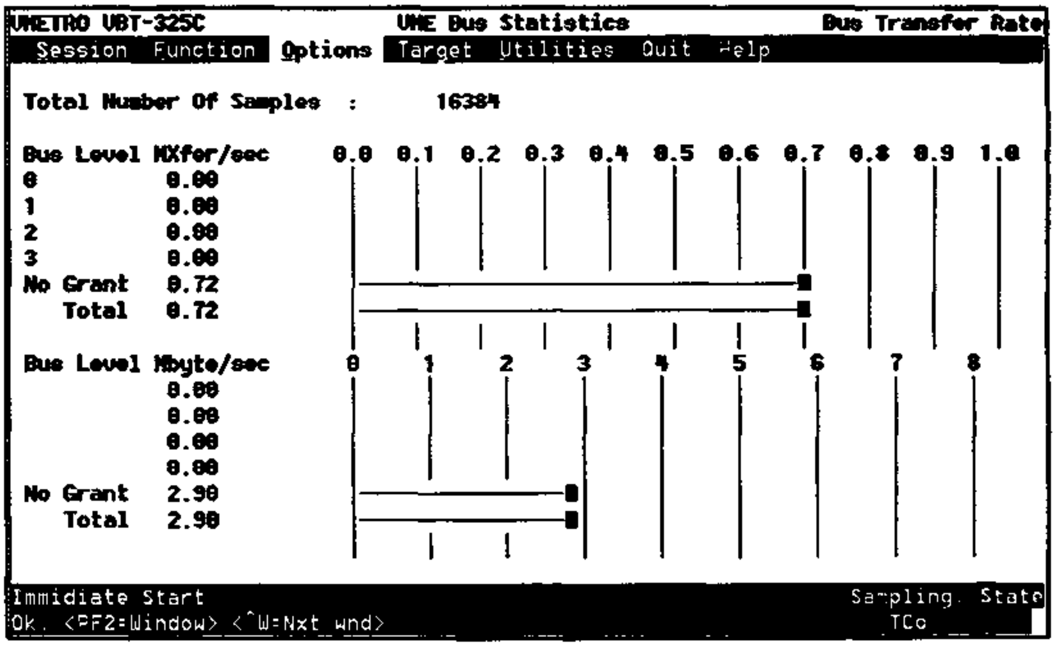
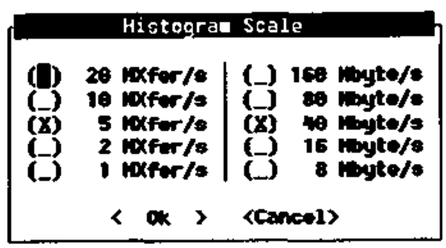


Figure 37. Bus Transfer Rate histograms. The upper shows transfers per second, and the lower bytes per second.

Change scale A very convenient function is `Options/Maximum Scale`. With this function you can change the scale for the two histograms individually. On the figure above is maximum scale for transfers per second set to 1 MXFers/s, and for bytes per second to 8 Mbytes/s.



5 EXAMPLE OF USE

The operation of the VBT-325 is controlled from the **Setup screen** as shown below. The Setup screen consists of three main elements:

- The main command bar
- The "Event Patterns" window
- The "Sequencer" window

In addition, there is a status line at the bottom of the screen that displays error messages, possible keystrokes as well as sampling status.

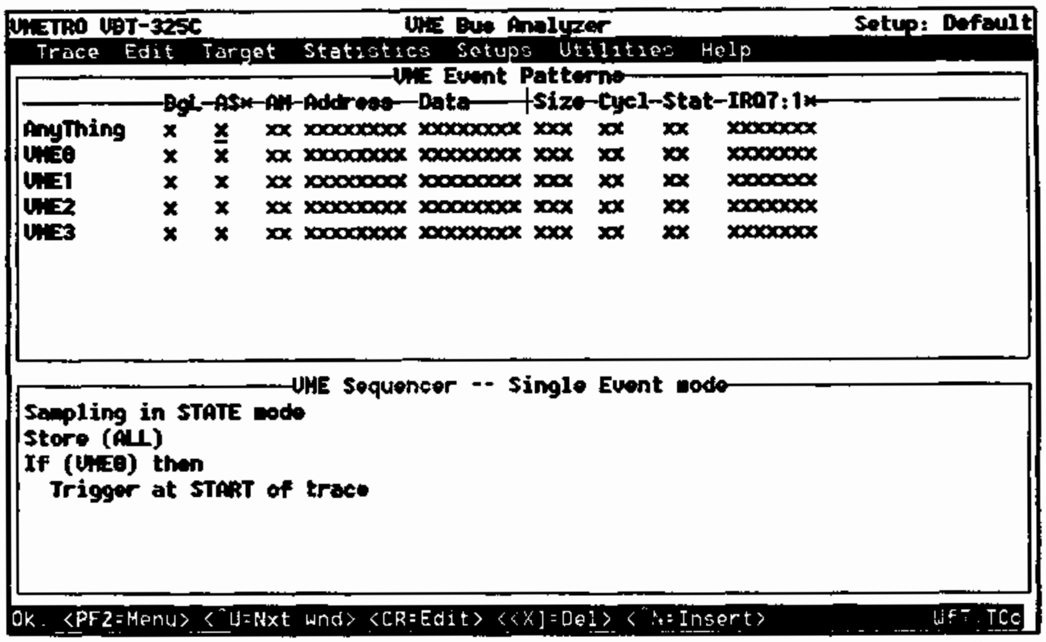


Figure 38. Setup screen, VBT-325C. (Note that for VBT-325B, the "Target" command is not present in the main menu unless the S/TIM200-PB piggyback module is installed.)

The following chapter gives a quick guide of how to use the VBT-325 to set triggers, capture data and see a trace. Throughout the examples below, the accelerator keys, as found in the command bar and pull-down menus, are used to execute the commands. However, it is always possible to activate commands by moving the cursor and selecting with a <CR>.

Setting trigger

The VBT-325 uses the **Event Patterns** and **Sequencer** as the main control elements to define the trigger and store conditions. These two functions are always present as separate windows in the Setup screen for the selected target bus.

Event Patterns

The Event Patterns define the trigger, store or count conditions. To edit the Event Patterns: type **I** for Edit and **E** for Event Patterns. Use the arrow keys to move to the different fields.

Edit a signal field

Position the cursor on a field (please note that the Event Pattern *AnyThing* cannot be edited), press <CR> and a dialog box will appear. Position the cursor on desired field, press the *<SPACE BAR>* and then <CR>. If a "v" is present next to the highlighted field, press the *down arrow* key to reveal the mnemonic options. To exit the dialog box, press <ESC> or dot ('.'). Also, you may enter binary values directly into any desired field.

To add signal fields:

Press <INS> or '^N', and a selection list will appear. Place the cursor on the desired signal field and press <CR>. To see signals that are not present on the screen, you can scroll horizontally by moving the cursor beyond the last signal field on the screen.

To delete contents of a field or a signal:

Press <DEL> or '^BS' (Ctrl-Backspace) and a dialog box will appear. Position the cursor on the desired field, press the *<SPACE BAR>*, and then press <CR>.

To add another Event Pattern:

Press <INS> or '^N', position the cursor on "Event Pattern", and press <CR>. You should now be prompted to specify a name for a new event pattern. Type in the desired name and press <CR>.

To remove an Event Pattern:

Press <DEL> or '^BS', place the cursor on "remove object Event Pattern", select by pressing the *<SPACE BAR>* and then press <CR>.

Single Event Mode

For simple triggers that only includes one single event pattern and no store qualifier, there is a default "Single Event Mode" which causes the event pointed to in the Event Patterns window to be chosen as the trigger event, thus eliminating explicit editing of the sequencer. Note that the Sequencer window is always used to display the current setting of trigger position and sampling method, and also the name of the event selected as the trigger, also in "Single Event Mode" .

The Sequencer

The Sequencer is used to define a complex trigger condition, store qualifiers etc., and is also used to display the current setting of trigger position and sampling method.

Edit the Sequencer

Type I for Edit and S for Sequencer, confirm by <CR> (if this is first time the sequencer is entered) and place the cursor at the desired position by using the up and down arrow keys. Press <INS> or '^N' to bring up a list of the possible choices of operators, place cursor on your selection and press <CR> twice. If a "v" is present next to the highlighted field, press the down arrow key to reveal the options available cursor down to the desired option and press <CR> twice. To delete a line or set in single event mode, press <DEL>, position the cursor on your selection and press <CR>.

Start Sampling

To take a VMEbus trace

Press **T R** and a Sampling Status box will appear. The Sampling Status box will indicate when a trigger has been found. Make some traffic on the VMEbus, and the sampling status box will indicate when the analyzer triggers. If there is a lot of VMEbus traffic, the trace will soon become full, and then the trace will be displayed automatically. If the trace does not become full, type **H** or <CR> in the <Halt all> button, and the **T S** to see whatever has been captured in the trace.

To trace another bus

Press **G** for Target. In this window you may select **B** for VSB bus. The screen will now appear almost as it did for VME, but the event patterns now have VSB signal names.

(VSB is default factory setting. In order to trace another P2 bus, type **R** to reconfigure to VXI, SCSI, or P2ac, place cursor next to selection, press <SPACE BAR> then <CR> again at second dialog box with warning about the need to reconfigure jumpers. If you really want to change the P2 target bus, take the board out of the crate and reconfigure the jumpers according to the section "JUMPER SETTINGS".)

After the reconfiguration, the new target bus will now appear in the Target menu. You may now change to the new target and edit event patterns or sequencer just as you did for VME.

Trace Multiple busses

If you want to trace multiple busses simultaneously, e.g. VME and VSB: press **T M** for Run Multiple and a sampling status box will appear, showing the status of all the tracers currently running. (May be up to three when the TIM200-PB 200MHz module is present). When the Sampling Status box indicates that cycles have been captured, type **H** or <CR> in the <Halt all> button, and the **T S** to see whatever has been captured in the trace.

To display VSB or other target, press **Q** to Quit and **G** for Target, select VSB or other bus and press **T S** for Trace Show.

You should now see the Trace Display Screen.

Trace Display Screen

With the cursor in the trace window, you can move across the screen by using the *right arrow* key, if you want to see more signal fields, type <INS> or '^N' and then press <CR> for the fields you want to see. The VBT-325 captures all 99 VME signals, and you may select the signals you want to view at this point. You can also delete fields by pressing <DEL> or '^BS'.

Time Tags

To change time tags from relative to absolute: type **O** for Format and **A** for Absolute Time. There will a short time delay to recalculate the time tags for the entire buffer.

Jump

To jump to another line in the trace buffer: type **J** for the Jump Menu. To show the last line, type **J L** and to see the first line, type **J F**. To see a particular line type **J** N, and that number.

Search

To search for a particular pattern: type **S** for the Search Menu and type **F** for Find. A search window will appear at the bottom of your screen with a search pattern, as well as the event patterns that were entered in the Setup screen. You can edit this search pattern by pressing <CR> at any field. To bring up the next sample with the same pattern, press **<F3>** or **<PF3>**. You can extract these samples by typing S for Search and **E** for Extract mode. Typing **S E** once more turns extract off.

Add a trace window

To add a Window: type **W** for the Windows menu and type **N** for New. Place your cursor on either Waveform or Alphanumeric, press the *<SPACE BAR>* and then press <CR>. Now you should see another window on your screen. The windows are independent *(correlating them will be an option in future firmware versions)*. To move between the windows, press **<F6> or '^W'**. To go back to the main menu: type Q for Quit.

Timing sampling

Now to change Sampling Mode from state to timing: type **I** for Edit and **M** for Sampling Mode. Use the down arrow key to position the cursor on your selection, and then press <CR>. The trigger position can also be changed here by pressing **T**. Then press **T R** and **T M** to do another Trace Run.

When the analyzer triggers, it will now display trace information as waveforms by default. It is still possible to see the trace as an alphanumeric list, do this by opening another trace window with **W N A** and confirm with **CR**. If you only want an alphanumeric list, close the waveform window by placing cursor in that window with <F6> or **^W** and do **W C** for Window Close.

Statistics

Statistics provides statistical measurements of target bus activity. From the main menu press S for Statistics. You should now see the Event Counting

Histogram. In order to see the Bus Utilization Histogram, press **F** for Function and **B** for Bus Utilization (see pgs 57-63, 83-87).

To run Statistics: Press **S** for Session , **R** for Run, and now you will see the event Histogram. There are several options that can be changed like the bar markers, number of samples counted and which events are to be counted. These options can be changed by pressing **O** for Options and choosing your selection. To run Statistics for VSB or other P2 bus, press **G** for target and select VSB or other P2 bus and the press **S R**.

Piggybacks

If you have a piggyback module connected, the following will apply:

VBAT-PB: Type **V** to enter the VBAT menu. To enable the VBAT, type **M E** and the press <CR>. To see a description of a violation, type **X** to get into the violation window, then move the cursor to the violation, and press **X** to get an explanation. To return to the main menu type **Q** to Quit.

VDRIVE-PB: Type **V** to select the VDRIVE and a window will appear with the different options for the VDRIVE, move the cursor to your selection and press <CR>. If the Master or Slave option is chosen, a window will appear. Use the <TAB> to move throughout the window and remember if a "v" is present, press the *down arrow* to see the options.

TIM200-PB: To use the TIM, press **G** for target and select TIM VME. The TIM setup screen is very similar to the VBT-325, so you can edit the same way. To edit the duration filter, press **I D** and a popup window will appear to make your selection. To cross trigger with the VBT-325, press <INS> to add a signal field, and then select the VMEtrg field. To return to VME, press **G** and select VME.

STIM200-PB: To use the STIM, press **G** for target and select STIM VME. When the main screen for the STIM appears, it is an empty buffer. To initialize the buffer, type **R I**, or to load examples, type **R A**. To edit the buffer, type **I R** and at each field, press <CR> and a dialog box will appear. To add signal fields, press <INS> and to delete signal fields, press <DEL>. There are two commands, fill and comb, available to allow manipulation of either multiple fields or lines. This can be done with a single command. The fill command overwrites the buffer with the specified pattern. Comb will only change those signals identified in the specified pattern. To use fill or comb, type **I S** and a window will appear, enter a pattern and then type **I F** to fill or **I C** to comb. To start the Stimuli, type **R S**, and to stop it, type **R T**.

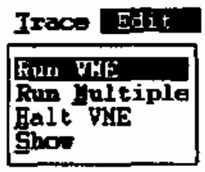
6 COMMANDS REFERENCE

In these chapter you will find a complete reference of all the menu commands in the VBT - 325.

Main menu - Setup screen

The main menu in the Setup screen contains a command bar with the major command groups for controlling all aspects of operation of the VBT-325, as described below.

Trace



The `'Trace'` command controls starting, stopping and displaying the trace. Since the VBT-325 is a multiple analyzer product (except the VBT-325B without an S/TIM200-PB), it is possible to start more than one analyzer at a time, or just start the one for which the Setup screen is currently seen. For this purpose, there are two Run command entries in the Trace command, `'Run <current target>'`, and `'Run Multiple'`.

- Run VME

To start the analyzer for the target currently selected , simply execute

`Trace/Run VME`

where VME indicates that the current target is VMEbus. (The 'Target' command is explained later in this chapter.)

- Run Multiple

To start simultaneous operation of all the analyzers present in the actual hardware configuration (up to three, as when the VBT-325C is equipped with a S/TIM200-PB), there is a variation of the Run command called

`Trace/Run Multiple`

- Sampling Status

A `'Sampling Status'` box appears on the screen when any `'Trace/Run..'` command is executed, indicating which analyzer is running, and its status, see figure below.

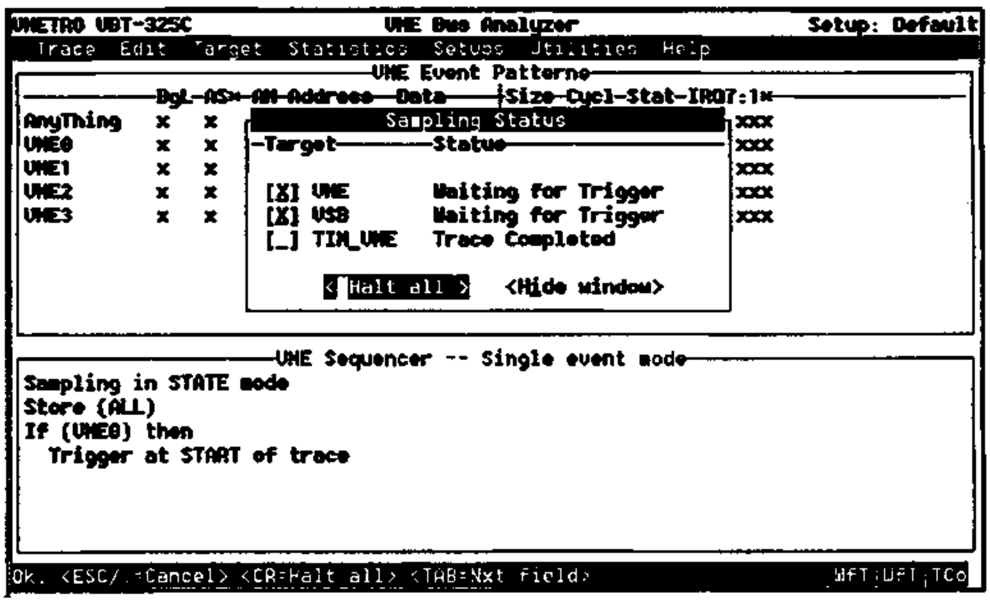
In order to access other commands during sampling (e.g. to enter Transparent Mode), the Sampling Status box may be hidden with the <Hide Window> button.

Status flags When the sampling status box is hidden, it is still possible to see the sampling status on the screen , as "flags" in the lower right corner of the screen. The flags show the status of up to three analyzers, in the following sequence:

`| VME | P2 | TIM200 |`

The status may be one of the following:

Sampling status	Abbreviation in "Flags"
Waiting for Trigger	WfT
Triggered	Trg
Trace Completed	TCo



igure 39. The Sampling Status box shows the current status of each of the sub-analyzers that are present on the board. Note also that the same information is present in the "Status flags" in the lower right corner on the screen. (The sequence of the analyzers is the same as in the status box.)

- Halt VME

Normally, the trace acquisition will stop by itself and present the Trace Display screen after the trigger is found and the trace is filled. However, if the trigger is never found, or the trace buffer does not get completely filled after the trigger, one may want to stop the trace manually. When only a single analyzer is running (i.e. the current target), this is done with the command

`Trace/Halt`

If the `'Run/Multiple'` was used to start all analyzers, it is possible to stop all of them (or the ones still running) through the

`<Halt All>`

button in the Sampling Status box.

- Show

To see the contents of the trace buffer if the trace was not completely filled with samples and automatically shown, the

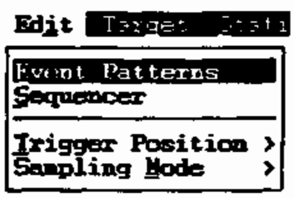
`Trace/Show`

command will bring up the Trace Display screen for the currently selected Target. This screen has its own set of commands, described in the ch. "Trace Display menu" later in this section.

Show trace from another target

In order to see the trace from another target bus, one must first use the `'Target'` command to enter the desired target, and then use `'Trace/Show'`.

Edit



The `'Edit'` command allows the user to modify all the parameters that control the process of capturing, triggering and stopping a trace. This involves the 'Event Patterns' window and the 'Sequencer' window, which are used to specify the desired trigger and store conditions, as well as the Trigger Position and the Sampling Mode.

- Event Patterns

Editing the `'Event Patterns'` window specifies the desired patterns to be used by the word recognizers during sampling. Refer to detailed description of how to edit the Event patterns in the previous chapter, "OPERATION".

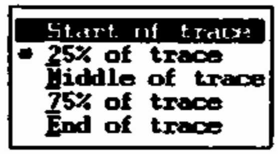
- Sequencer

Editing the `'Sequencer'` program specifies complex triggers, and store conditions. Refer to detailed description of how to edit the Sequencer in the previous chapter, "OPERATION".

Note that it is not necessary to enter the sequencer for simple triggers without any store qualifiers, since the default **"Single Event Mode"** automatically selects the event pointed to in the Event patterns window to be the trigger.

- Trigger position

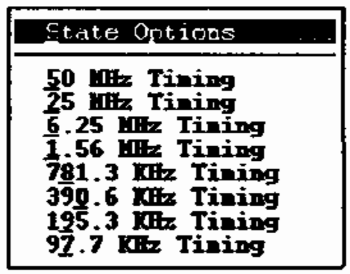
The `'Trigger Position'` is defined in a secondary pull-down menu, with selections for Start (0%), 25%, Middle (50%), 75% and End (100%) of trace.



The selected sampling mode and trigger position is reflected in the Sequencer program.

- Sampling mode

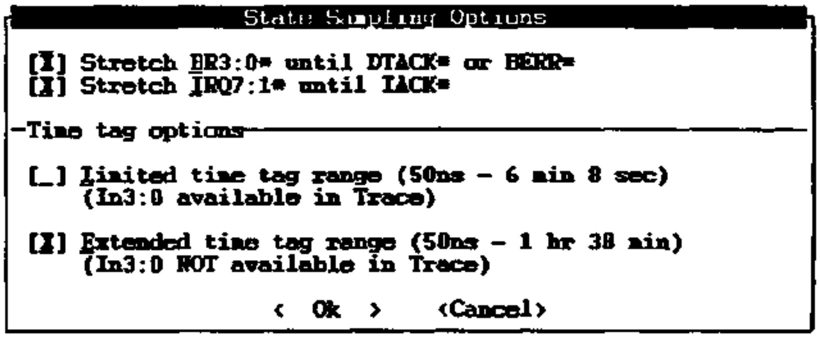
`'Sampling mode'` is selected in a secondary pull-down menu, and allows the user to select State (synchronous) or Timing (asynchronous) sampling.



There is also an `'Options'` sub-command which allows the user to select other timing sampling rates, down to 97.7KHz. The selected sampling rate is the one that appears in the secondary menu next time the Sampling mode command is entered.

- Options / State Sampling options

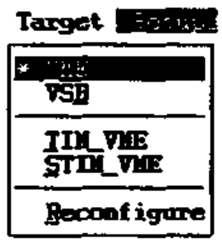
This command controls whether the BR* and IRQ* signals shall be stretched for capture on DTACK*/BERR* in VMEbus state sampling. It also controls the Limited/Extended time tag options vs. the availability of the four external signals In3:0 in the trace. Please refer to FUNCTIONAL DESCRIPTION, External Inputs, Shared In3:0 and Time Tag bits.



Extended time tag vs. External In3:0 signals

Four bits in the "Time tag counter" on the VME part of the VBT-325 are shared with the external inputs (In0-In3). The "Extended time tag" option sets the time tag counter to its full width (16 bits), capable of measuring time intervals up to 1 hour 38 min. In this mode the four External input signals taken from the four mini-coax inputs **In3-0 are not available in the trace buffer** (although they are still available in the Event patterns). When the "Extended time tag" option is off, the time tag counter is only 12 bits, and the four In0-In3 signals are available also in the trace buffer. In this mode, time intervals up to 6 min 8 sec. can be measured.

Target



The VBT-325C allows simultaneous analysis of both the VMEbus and a P2 bus like VSB, SCSI or VXI. If the S/TIM200-PB is equipped, 200MHz analysis can be performed on VME or a P2 bus. The `'Target'` command is used to switch control of the user interface between the different analyzers present in a VBT-325.

The `'Target'` command makes the selection to which bus the Setup screen (containing Event patterns and Sequencer) and Trace Display screen apply.

Note that during and after sampling, the sampling status of the different targets are shown directly in the Target pull down menu. Current target is shown in low intensity, since it is already selected.

Note to VBT-325B users: VBT-325B supports VMEbus only, and the Target command is then not present unless the S/TIM200-PB is installed.

- VME

The primary target bus for the VBT-325 is VMEbus, and this is the first entry in the Target command.

- P2 bus

The P2 bus on VBT-325C can be selected to be VSB, SCSI, and VXI_P2ac. Only the target currently selected is shown in the Target pull down menu, and by default, this is VSB.

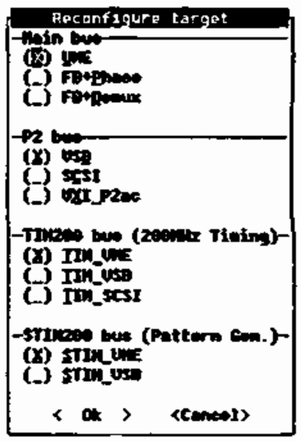
- TIM200 bus

When the TIM200-PB 200MHz Timing Analyzer is installed, it can be selected in the 'Target' command. If the P2 option is purchased, it can be enabled by the 'Reconfigure' option as described below.

- STIM200 bus

When the STIM200-PB 200MHz Stimuli / Pattern Generator and Timing Analyzer is installed, there will be two entries for this module in the 'Target' command: The *TIMxxx* selects the Timing Analyzer and the *STIMxxx* selects the Stimuli section of the STIM200-PB module. If the P2 options is purchased, it can be enabled by the 'Reconfigure' option as described below.

- Reconfigure

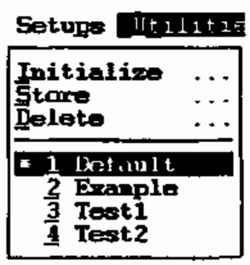


The default setting on the P2 bus of the VBT-325C is VSB. If the user wants to use another P2 bus, like SCSI or VXI_P2ac, the Reconfigure command must be used. (This command is also used to switch the bus for the S/TIM200-PB.) Reconfiguring the P2 bus on the VBT-325C normally requires that some jumpers are moved, so a warning about this is given when closing this command if changes have been made.

Statistics

The '`Statistics`' command activates the Statistics screen, containing the statistics histograms and a special menu bar. Please refer to the ch. "Statistics menu" later in this section.

Setups



The '`Setup`' command is used to initialize, store, delete or retrieve user-defined setups of event patterns and the sequencer program, as well as the trigger position and sampling mode. At least 50 setups with four events each may be stored.

- Initialize

'Initialize' will reset all values that have been entered into the Setup screen, i.e. the event patterns, sequencer, trigger position and sampling mode, to the default conditions seen after initial power up or after clearing non-volatile memory.

- Store

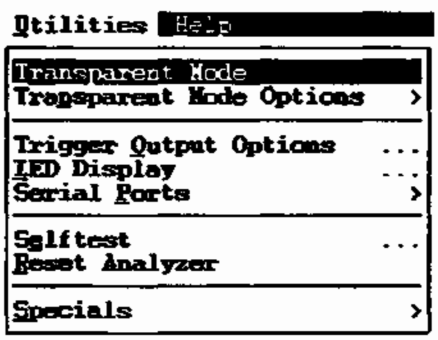
The 'Store' sub-command will ask for a name, and store the current setup or "context" as seen on the screen. The selected name will then appear in the 'Setups' pull down menu. When a stored setup is to be used again, this is done simply by selecting it in the pull down menu.

The current setup is shown in the upper right corner of the Setup screen. There is always one setup called 'Default' present, this cannot be deleted.

- Delete

A setup can be deleted from the list by the 'Delete' sub-command.

Utilities



Under the 'Utilities' command group, a number of utility functions are available.

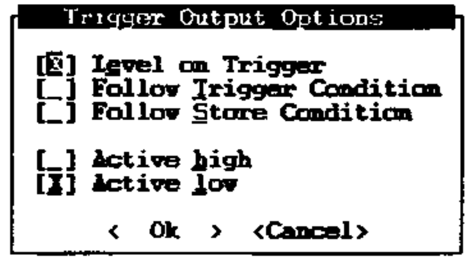
- Transparent Mode

The 'Transparent Mode' is used to communicate with another CPU etc. through the second serial port on the VBT-325 (see the chapter "INSTALLATION, Terminal connection, Transparent mode" for more information). In this mode, the processor on the VBT-325 will simply pass all characters between the terminal and the host CPU. <F1> can also be used to enter transparent mode.

- Transparent Mode Options

A special termination character is used to exit transparent mode, and by default <ESC> is used. However, as this character may cause problems in some cases, the user can define any other character as termination character by the sub-command 'Transparent Mode Options/Termination char..'.

- Trigger Output Options



The front panel trigger output may be programmed to change on trigger, or to follow the trigger or the store condition that prevails in the current state of the sequencer, and to be active high or low. This is selected by the Trigger output options dialog box.

Actions on the Trigger Output signal:

Level on Trigger:

Signal will go active when the trigger sample occurs, and will stay active until new Trace/Run is given.

Follow Trigger Condition:

Signal will go active when the trigger sample occurs, but will revert to inactive state on the next sample.

Follow Store Condition:

Signal will go active on samples satisfying the Store Condition in the current state of the sequencer. Signal reverts to inactive state on samples not satisfying the prevailing store condition.

Note: *When the* `'Trigger Output Option/Follow Trigger'` *or* `'Follow Store'` *is selected, a short pulse is generated on the TRIGGER output signal when the tracer is started with* `'Trace/Run'`.

- LED Display

The user can select different options for use of the LED display on the front panel by means of a dialog box, accessed by the command

`Utilities/LED display.`

[X] Voltage readout

The power supply voltage in the system where the VBT-325 is inserted can be displayed on the front panel LED display by selecting either 5V or +12V or -12V in the "LED display" dialog box. (The 5V reading is the default reading on the LED display when the analyzer is idle.)

V+T/V jumper The jumper with the yellow label "V+T and V" must be in the V position (factory setting) to get correct voltage measurements.

Note: *Boards with ECO level lower than B8 does not have the yellow "V+T and V" labelled jumper. On these boards, some restrictions apply to voltage measurements when the temperature probe is installed, see below.*

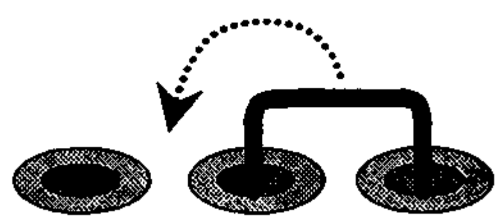
ECO < B8? *Note: On boards with ECO level lower than B8, the temperature probe **must not be present** when doing voltage measurements. If the probe is present, the voltage readings may be wrong with as much as 0.3V. (Future FW versions will check for the ECO level automatically, and disable the voltage measurement if the temperature probe is detected when the ECO level is too low. For an upgrade of your board, please contact VMETRO).*

[X] Temperature readout

When the optional temperature probe* is installed in the **In1/T** input on the front panel, the temperature can be displayed on the LED display by choosing "Temperature" in the dialog box under the "LED display" sub-command.

*) Temperature probe: VMETRO part number 401-325ETS-1.

Move jumpers! *Note: The jumper J66 (located just inside of the mini-coax connectors) must be in position closest to the front panel to allow temperature measurements (see also fig. A3 in the section JUMPER SETTINGS). In the other position, In1 is used for one of the four possible "External signal inputs" to the tracer (factory setting).*



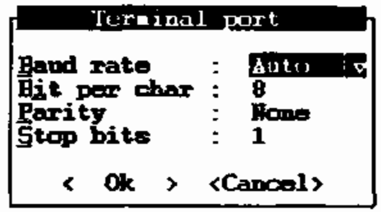
V+T/V Also, the jumper with the yellow label "**V+T** and **V**" must be set in the **V+T** position. (Boards with ECO level lower than B8 or >= C0 does not have the yellow **V+T/V** labelled jumper. See below.)

ECO < B8? *Note: On boards with ECO level lower than B8, the temperature probe **must not be present** when doing voltage measurements, refer to "[X] Voltage readout" above.*

Temp. range The valid range of the temperature reading is **0-80°C / 32-176F**. If a reading outside this range is seen, this should be ignored. *(Future FW versions will check for this automatically).*

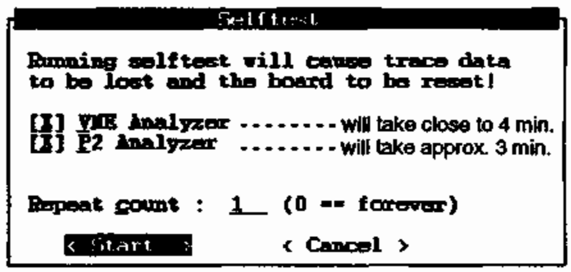
- Serial Ports

The baud rate etc. of the two serial ports can be defined independently of each other by the `'Serial Ports'` command.



- Selftest

The 'Selftest' sub-command brings up a dialog box that allows the individual hardware modules to be selected as part of the selftest.



Note: *Please note that the selftest command starts an extensive test of most hardware modules on the selected analyzer(s). It will take from 2-4 min. for each analyzer for the selftest to complete. Running the selftest for the VBT-325C will take close to 7 minutes.*

The dialog box shown above is the one for VBT-325C, without any piggyback module installed. If the actual product is a VBT-325B, the P2 part will not be present, and if a piggyback module is installed, its name will appear in the list (except for VBAT, for which there is no selftest available).

- Reset Analyzer

This commands resets the VBT-325 just as would an activation of the front panel reset. This can for example be used to get the start-up screen, to change terminal type or to clear non-volatile memory.

- Specials

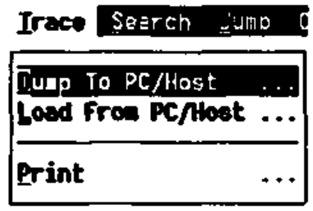
The `'Specials'` command allows the user to read and set the current ECO level and Time and Date. The ECO level is normally set during manufacturing, and if a hardware ECO upgrade has been performed. The Time and Date need to be set if the non-volatile memory has been cleared/lost.

Help

The `'Help'` command will bring up a help screen that contains a short-form tutorial of how to navigate with cursors, how to use the function keys, etc. A `<Next>, <Previous>` button can be used to access to the different pages in the Help texts.

Trace Display menu

Trace



The 'Trace' command in the Trace Display screens differs from the one in the Setup screens. It has commands for dumping the samples in the trace buffer to a PC, and loading from a file on a PC to the trace buffer. Also, a command to print the samples in the trace buffer is included.

- Dump to PC

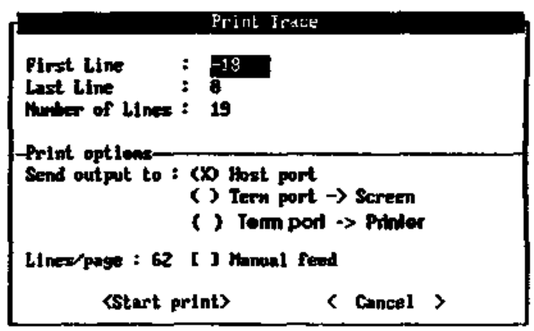
The 'Dump to PC' sub-command dumps the contents of the trace buffer to a file on a PC for archival or later comparison with subsequent trace sessions. A dialog box appears to assist the user in performing this. Please refer to the ch. "VMETRO VT100 EMULATOR - VT100.EXE, Trace Dump" for detailed instructions about this feature.

Note: *To review dumped trace files on a PC, the "Simulator" is most useful. Please refer to ch. "SIMULATOR" in the end of this manual.*

- Load from PC

The trace files may be loaded back in to the VBT-325 with the 'Load from PC' sub-command. Please refer to the section VMETRO VT100 EMULATOR - VT100.EXE" for more information about loading trace buffer files to/from a PC.

- Print



The 'Print' command sends the trace data sequentially in a formatted manner, just as shown on the screen, to a printer on the host port or to the terminal. It is possible to specify to/from line numbers or just number of lines. Further you can specify lines/page and if you want manual feed.

To terminal port and printer If you select Term port -> Printer the printout will be sent to a local printer, for example to your Windows printer if the terminal emulator is the one found under Windows.

Note: *If the terminal emulator is run with baud rate = 19200 you can loose characters on the printout. A baud rate of max 9600 is therefore recommended.*

Search

```
Search  Jump  Count  Form
┌──────────────────────────┐
│ Find                 ... │
│ Next Match          F3   │
│ Extract Mode             │
│ Close Search Window      │
└──────────────────────────┘
```

The 'Search' command offers powerful search and extract functions. *Search* allows the user to locate a particular pattern in the trace buffer, while *Extract* provides a qualified presentation of samples from the trace buffer, so that only samples matching the specified search pattern are displayed. As in the Event Patterns window, the search pattern may contain "don't care", specific values or mnemonics in the various signal fields.

- Find

SrcPatt

The 'Find' sub-command brings up the **Search window.** This window duplicates the list of event patterns defined in the Event Patterns window, with the addition of one event pattern called 'SrcPatt'. The Search window works in the same manner as the Event Patterns window with respect to editing etc. This means that it is possible to insert new event patterns directly in the Search window.

```
──────────────────────────Search ── VME Trace #1 ──────────────────────────
──────────────────────────BR3:0=─BgI─AM───Address──Data─────┼Size──Cycl─Stat────
SrcPatt                   xxxx    x   xx   xxxxxxxx xxxxxxxx xxx   xx   xx
AnyThing                  xxxx    x   xx   xxxxxxxx xxxxxxxx xxx   xx   xx
VME0                      xxxx    x   xx   xxxxxxxx xxxxxxxx xxx   xx   xx
VME1                      xxxx    x   xx   xxxxxxxx xxxxxxxx xxx   xx   xx
```

<F3> = Search The event pattern that shall be used for searching is the one where the cursor is placed (similar to the selection of trigger event in "Single Event mode"). Searching is started by typing <F3>, or activating the command sequence 'Search/Next Match'.

When a search target is found, the cursor will be placed on the found data, and the Search window will disappear. However, the search can be repeated simply by typing another <F3>. If a search operation failed to find the search target, the Search window will remain on the screen, for convenient entry of another search pattern. The Search window can be closed by the command sequence 'Search/Close Search Window', described below.

Wrap-around When the search process reaches the last sample in the trace buffer, it will wrap around and continue searching from the beginning of the trace buffer. (This feature can be turned off by 'Format/Global Options, []Wrap-around search mode').

Note that the 'SrcPatt' event (and any other event that have been inserted in the Search window) will be included in the list of events patterns when returning to the Setup screen. This allows immediate use of a search pattern as a trigger/store condition in the Sequencer program.

- Extract Mode

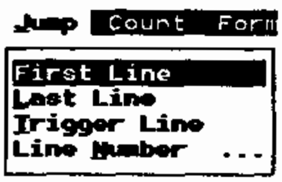
A particular pattern may be "extracted", so that only samples satisfying the search pattern are displayed on the screen. This is done by executing `'Search/Extract Mode'` when the Search window is open (after a pattern is edited by the `'Search/Find'` command. Then simply type the accelerator keys `'S'` and `'E'`).

When Extract is "on", this is indicated by an asterisk in the pull down menu. Extract can be turned off by executing the command `'Search/Extract Mode'` again (`'S'` and `'E'`).

- Close Search Window

The sub-command `'Close Search Window'` will cause the search window to disappear from the Trace Display screen. This is useful when search operations are no longer needed. When this is done, it is no longer possible to repeat the last search by <F3>.

Jump



The `'Jump'` command moves the cursor to a given line number in the trace, to the first or last line or to the trigger line. This gives quick and convenient navigation in the trace buffer.

- First Line

Jumps to the first line in the trace buffer that contains a valid sample.

- Last Line

Jumps to the last line in the trace buffer that contains a valid sample.

- Trigger Line

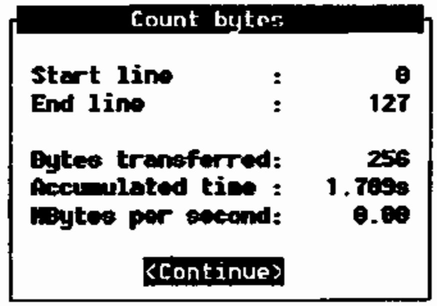
Jumps to the line in the trace buffer that contains the trigger sample (line number 0).

- Line number

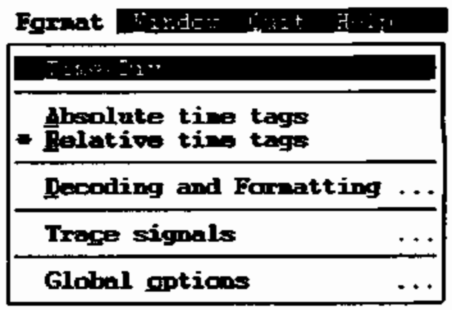
Jumps to a given line number in the trace buffer. A dialog box is given where the user enters a line number.

Count

Counts Mb/s between given lines in the trace buffer. When you select Count, you will first get a dialog box where you select start line and end line (inclusive). When you select <Ok> you will get three new lines displayed in the dialog box, showing bytes transferred, accumulated time and Mbytes per second within the selected range of lines.



Format



The 'Format' command gives the user the choice of various presentation forms of the contents of the trace buffer.

- Time/Div

In Waveform mode, the 'Time/Div' command can be used to "zoom" in or out in the waveform display.

- Absolute/Relative Time tags

When alphanumeric trace is selected, the time tags can be selected as Relative or Absolute from the trigger point with the 'Absolute/Relative time tags' command. Current setting is shown with an asterisk in the pull down menu.

- Decoding and Formatting

'Decoding and Formatting' allows the signal group at the cursor position to be presented in a decoded form (with mnemonics), or simply as binary or hex values.

- ASCII Decoding

Please note that it is possible to display "Data" as ASCII text in a VMEbus trace by inserting the column "DataASCII" in a VME trace window. See also the chapter "SIGNAL REFERENCE".

- Trace signals

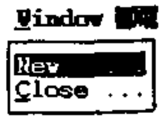
The 'Trace signals' sub-command selects between 'Grouped' or 'Single' i.e. individual control signals. (VMEbus only).

- Global options

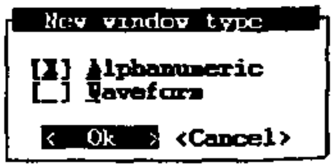
The 'Global option' sub-command contains some flags that control some "look-and-feel" elements of the user interface.

Window

The 'Window' command allows another window to be added to the screen, either as a trace list or waveform.



- New



When the command sequence

```
Window/New
```

is executed, a dialog box appears that selects between these two presentation forms for the trace data. This will cause a trace window in addition to the one already present to appear on the screen. Two windows will normally look fine on most screens, and on larger screens, three windows may be very useful.

- Close

If the user wants to see only one window of waveforms covering the full screen, this can be done by opening a waveform window as described above, and then closing the old trace window with

```
Window/Close
```

Remember to move the cursor to the appropriate window (using <F6> or '^W') before closing. The active window will have its frame highlighted.

Quit

'Quit' brings back the Setup screen for the selected target. (See screen structure figure in the ch. "OPERATION, User-interface structure").

Help

'Help' brings up the *general* Help utility, describing general user-interface mechanisms etc. *(There is no specific Help regarding the Trace Display functions in the 4.99.120 firmware version.)*

Statistics menu

Session

The '`Session`' command in the Statistics screen is the functional equivalent of the '`Trace`' command in the Setup screen, providing '`Run`', '`Continue`' and '`Halt`' controls for operating the statistics.

- Run

To start the statistics function, simply execute

`Session/Run`

All the statistics counters for the target currently selected will then start to count from zero, and the histograms will be updated according to the update rate etc. selected under '`Options/Count Options`' (see below.)

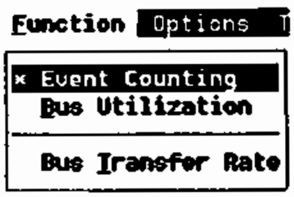
- Continue

'`Continue`' resumes counting from the values reached the last time the statistics was stopped with '`Halt`'.

- Halt

'`Halt`' stops the statistics session and freezes the histogram screen.

Function



The '`Function`' command is used to select between the statistics functions. Three functions are available, '`Event Counting`', '`Bus Utilization`', and '`Bus Transfer Rate`' the latter is only available for VMEbus.

- Event Counting

The '`Event Counting`' sub-command invokes the statistics function that provides real-time histograms of the occurrence of four user-specified events. For VMEbus, this function also includes the distribution of bus cycles among the four Bus Levels, counted among all cycles or only cycles matching the selected events (controlled by '`Options/Count Options,Qualifier`' see below).

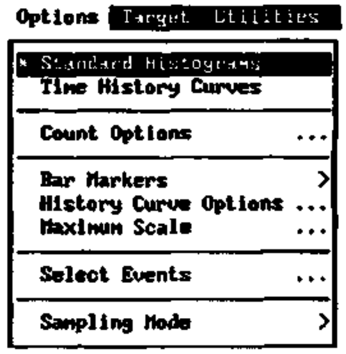
- Bus Utilization *(VME and VSB)*

The 'Bus Utilization' sub-command invokes the preconfigured statistics function that provides real-time histograms of the percentage of time when selected VME or VSB signals and signal combinations are active. This gives a picture of how much of the time the bus or parts of it is busy.

- Bus Transfer Rate *(VMEbus only)*

You get this function by selecting Statistics from the main window, and then select 'Function/Bus Transfer Rate'.

Options

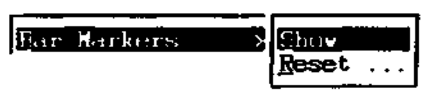


The 'Options' commands are used to configure various window control and display features. They include 'Bar Markers', 'Graph Display Options','Max. Scale','Count Options' and 'Select Events', in addition to selection of 'Standard Histograms' or 'Time History Curves'.

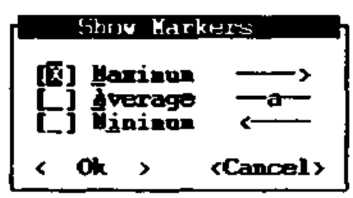
- Standard Histograms vs. Time History Curves

Two graphical display options are available. The 'Standard Histograms' option uses histogram bars, showing the current reading of the statistics counters. 'Time History Curves' shows how the values change in time, by means of a curve in an X-Y diagram, where the X-axis represents time.

- Bar Markers



The bar marker function calculates minimum, maximum and average values for the ongoing series of counter readings. The calculated values are indicated in the proper positions in the histograms with the symbols as shown in the dialog box below.



Show To make a bar marker visible, perform the sub-commands 'Bar Markers/Show' and select the desired marker(s) in the dialog box by using the <SPACE BAR>.

Note: *Average Bar Markers is not available in Accumulate mode, as this would equal to current value.*

Reset The sub-command sequence 'Bar Markers/Reset' resets the recorded values for the selected marker(s), so that only subsequent count values (after this command is executed) will be taken into account when displaying new bar markers. Select the desired marker(s) to be reset in the dialog box by using the <SPACE BAR>.

- Graph Display Options

The 'Graph Display Options...' command is used to tailor the display to an application when 'Time History Curves' are used. This command cannot be activated when the 'Standard Histograms' are used.

```
Graph Display Options
-Screen Split-------------
  2  Parts Horizontally
  2  Parts Vertically

-Graph Drawing------------
(X) Retrace Graph
( ) Scroll Graph

  <  Ok  >     <Cancel>
```

The 'Screen Split' entry is used to define how many horizontal and vertical parts the screen should be divided into, to display the X-Y diagrams (max. 4) for the time history curves. The 'Retrace Graph vs. Scroll Graph' is used to define the action to be taken when the time history curve reaches the end of the X-Y diagram. If 'Retrace Graph' is chosen, the curve wraps around, and starts over again from zero. 'Scroll Graph' causes the entire curve to scroll when reaching the end. With the latter option selected, the current value is always found in the end of the curve.

- Maximum Scale

'Maximum Scale' provides graduated horizontal scaling of the histograms, ranging from 5% to 100%. Choosing lower max. scale allows for better resolution of measurements with mostly low count values. For Bus Transfer Rate, the scale options are 1-20 MXfer/s and 8-160 Mbytes/s.

```
Histogram Scale

(X) 100 %   ( )  40 %
( )  80 %   ( )  30 %
( )  70 %   ( )  20 %
( )  60 %   ( )  10 %
( )  50 %   ( )   5 %

  <  Ok   > <Cancel>
```

- Count Options

To optimize a statistics session to the actual system behaviour, there are several 'Count Options' that can be selected. The selected Count Options are always shown at the status line below the histograms in the Statistics screen.

```
Count Options
-Qualifier-------------------------
 (X) Count All Samples
 ( ) Count Valid Samples Only

-Update Every----------------------
 (X) Time Interval Of  1     Seconds
 ( ) Sample Count Of  64K    Samples

-Mode------------------------------
 ( ) Accumulate
 (X) Reset

      <  Ok  >       <Cancel>
```

Note: *'Count Options' do not apply to the Bus Utilization function.*

Qualifier The Qualifier selects whether "all samples" or only samples satisfying one of the selected events ("valid samples") shall be counted as the basis (100%) for the percentage calculations.

Update Every Sample Count

The 'Update Every Sample Count..' feature provides control of the screen update interval, beginning at 1K and increasing to a max. of 16M

Samples. Simple experimentation with this display control will assist the user to quickly determine the optimum parameter needed to acquire the maximum recording resolution for the application under test. Applications generating low bus cycle frequencies will typically require a smaller screen update parameter.

Note that if the screen update parameter is set too short, parts of the screen may not be refreshed completely, and the tracer may look impossible to control. If this symptom is seen, hit `'^C'` and wait for a dialog box. Press the `'<Stop Activity>'` button, and the session will be halted. Choose a less frequent screen update rate, or choose the `'Update Every Time Interval..'` option, and start a new session.

Update Every Time Interval

The `'Update Every Time Interval..'` feature provides control of the screen update interval by explicitly defining the time between screen updates. The time intervals starts at 1 second, and increases to a max. of 120 seconds.

Use the `'Update Every Time Interval..'` feature to get a constant refresh rate, independent of bus activity.

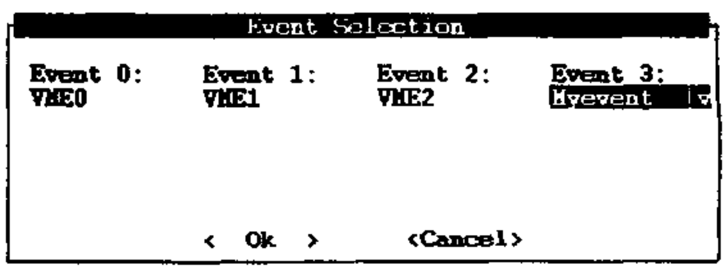
Mode

The count values to be shown as histograms may be calculated as a percentage of the total sample count in *each update*, or as a *cumulative* percentage of the total sample count in the current session. The first mode is referred to as the **Reset mode**, while the latter is referred to as **Accumulate mode**. Please refer to the ch. "OPERATION, Statistics, Counters Operation, Accumulate/Reset mode" for details of how the calculations are performed.

- Select Events

The user may select any four events from the Event Patterns window in the Setup screen to be used in the Statistics measurements by means of the `'Select Events'` sub-command. By default, the first four events in the Event Patterns window except the one named AnyThing are used.

To select another event, place the cursor on one of the four event positions in the Event Selection box and type down arrow ↓. Then select the desired event from the event selection list by <CR>.



Note: *`'Select Events'` does not apply to the Bus Utilization function.*

Please note that Bar Markers, Max. Scale, Count Options and Event selections will be set back to default values after board reset with the 4.99.120 firmware version. In future FW versions, these selections will survive board reset.

Target

The 'Target' command in the Statistics screen selects the target bus for the statistical measurement. (Note that the 'Target' command in the Statistics screen only applies to the VBT-325C.)

- VME

The primary target bus for the VBT-325 is VMEbus, and this is the first entry in the 'Target' command.

- P2 bus

The P2 bus on VBT-325C can be selected to be VSB, SCSI, and VXI_P2ac. Only the target currently selected is shown in the Target pull down menu, and by default, this is VSB.

It is possible to change to another P2 target bus by means of the 'Reconfigure' option found under the 'Target' command in the Setup screen. (Note that this requires jumper reconfiguration in most cases, ref. section "JUMPER RECONFIGURATION" in the back of this manual.)

Utilities

The 'Utilities' command in the Statistics screen duplicates the capabilities found under 'Utilities' in the Setup screen, please refer to the ch. "Main menu - Setup screen" previously in this section.

Quit

'Quit' brings back the Setup screen for the selected target. (See screen structure figure in the ch. "OPERATION, User-interface structure").

Help

'Help' brings up the *general* Help utility, describing general user-interface mechanisms etc. *(There is no specific Help regarding the Statistics functions in the 4.99.120 firmware version.)*

7 SIGNAL REFERENCE

This chapter gives you a complete reference to all the signals the VBT - 325 is working with.

VMEbus Signal Groups

Group name	Signals included in group
Address	**A01-A31,** plus **DS1*** acting as virtual A00.
Data	**D00-D31**
DataASCII	**D00-D31, decoded as ASCII.** Byte values not in the range 0x20 - 0x7E are displayed as "." (a dot).
BgL	**BG3-0* internally latched.** The "BgL" group contains latched bus grant information (latched on falling edge of BBSY*), displayed as ---0, --1-, -2--, 3--- for the four possible bus levels. BgL will show ---- if no bus grant was seen by the VBT-325 (due to position of VBT-325 in the daisy chain).
Size	**A01, DS1*, DS0*, LWORD*.** The control signals DS1*, DS0*, LWORD* are combined into the group called "Size". This group also takes into account the address bit A01, since this bit, together with the ones mentioned above determine the transfer size and the data aligning on the VMEbus.
Cycl	**IACK*, AM0, AM1, WRITE*.** The "Cycl" group contains the bus signals that define the current cycle type. The internally generated bit "_RMW" is also included in this group, set during Read-Modify-Write and Block cycles (in second cycle). Together with address modifier bits 1,0 (AM1, AM0), this field can be used directly to set mnemonics for Read, Write, RMW and Block cycles. Also, IACK* is included to select Interrupt Acknowledge cycle type.
Status	**DTACK*, BERR*.** The "Status" group consists of DTACK* and BERR*, with the mnemonics OK (equals to DTACK* = 0 and BERR* = 1), and ERR (equals to DTACK* = 1 and BERR* = 0).
Fail	**ACFAIL*, SYSFAIL*.** The "Fail" group contains the two VMEbus fail signals ACFAIL* and SYSFAIL*. These are available as a group to occupy less screen space horizontally, but the signals may also be shown individually as SYSF* and ACF*.
In3:0	These are the external inputs in the mini-coax connector on the front panel. Note that *In1* is shared with the temperature probe, so if the probe is used, *In1* is not available. (Determined by jumper J66 and "V+T, V" jumper, see under "LED Display" in the "COMMANDS REFERENCE" section). Also note that *In3:0* is not available in the trace if *Extended Time Tag* is selected with the

command 'Edit/Sampling Mode/Options/State Sampling Options' (Ref. section "COMMANDS REFERENCE"). This is also described in more detail in the ch. "FUNCTIONAL DESCRIPTION, External Inputs, Shared In3:0 and Time Tag bits".

VMEbus Signals

Symbol	VMEbus name
BBSY*	**BBSY***
BCLR*	**BCLR***
BR3:0*	**BR3:0***
BG3:0*	**BG3:0IN/OUT***
AS*	**AS***
DS1:0*	**DS1:0***
DTACK*	**DTACK***
BERR*	**BERR***
LWRD*	**LWORD***
IACK*	**IACK***
IAIO*	**IACKIN/OUT***
SYSR*	**SYSRES***
SYSF*	**SYSFAIL***
RETRY*	**RETRY***

Other signals - Target VME

Symbol	Description
In3:0	External inputs from front panel min-coax inlet.
P2trg	Cross-trigger from onboard P2 analyzer.
TIMtrg	Cross-trigger from S/TIM200-PB when this piggyback module is installed.
Vbat	Cross-trigger from VBAT-PB when this piggyback module is installed.
RMW	Read-Modify-Write. Signal generated internally in the VBT-325 when AS* is not taken high between cycles, and AM-codes are not indicating Block cycle. This means that the current cycle is the second (or subsequent) cycle in a VMEbus Read-Modify-Write cycle sequence.

StateMode	Indicates that sampling mode is 'State' when high (1). When low (0), sampling mode is 'Timing'. This bit is useful if a mix of 'State' and 'Timing' sampling is used within a trace (according to Sequencer program.)
VME64*	This signal is used to identify cycles with VME64 AM-codes, i.e. AM = 00, 01, 03, 06, 07 ,08 ,0C, 38, and 3C.

VSB Signal Groups

Group name	Signals included in group
AD(31:0)	**Multiplexed Address/Data Lines**
SPACE	**WR*, SPACE1, SPACE0.** This group select between the different address spaces: SYS, I/O, ALT. In addition, the WR* is included in this field to be able to select Interrupt Acknowledge cycle type (WR* = 1, SPACE1/0 = 00.) as a mnemonic in this field (IACK).
SIZE	**SIZ1, SIZ0.** Select transfer size as requested from master, i.e. LONG, 3BYT, WORD or BYTE.
ASACK*	**ASACK1, ASACK0.** Selects port size as responded by slave (D08, D16, D32).
Cycle	**WR*, ARB***. This group selects cycle type. The signals WR* and ARB* are grouped into the group called "Cycle", together with the internally generated bit "BLOCK", which is set when a VSB block cycle is detected (in second cycle).
Status	**ERR*, ACK***. Selects between bus error or OK cycle status.

VSB Signals

Symbol	VSB name
BREQ*	**BREQ***
BGIO*	**BGIO***
BUSY*	**BUSY***
AC	**AC**
DS*	**DS* Indicates Address/Data phase as 'ADDR'/'DTA'.**
LOCK*	**LOCK***
CACHE*	**CACHE***
PAS*	**PAS***
IRQ*	**IRQ***
WAIT*	**WAIT***
WR*	**WR***

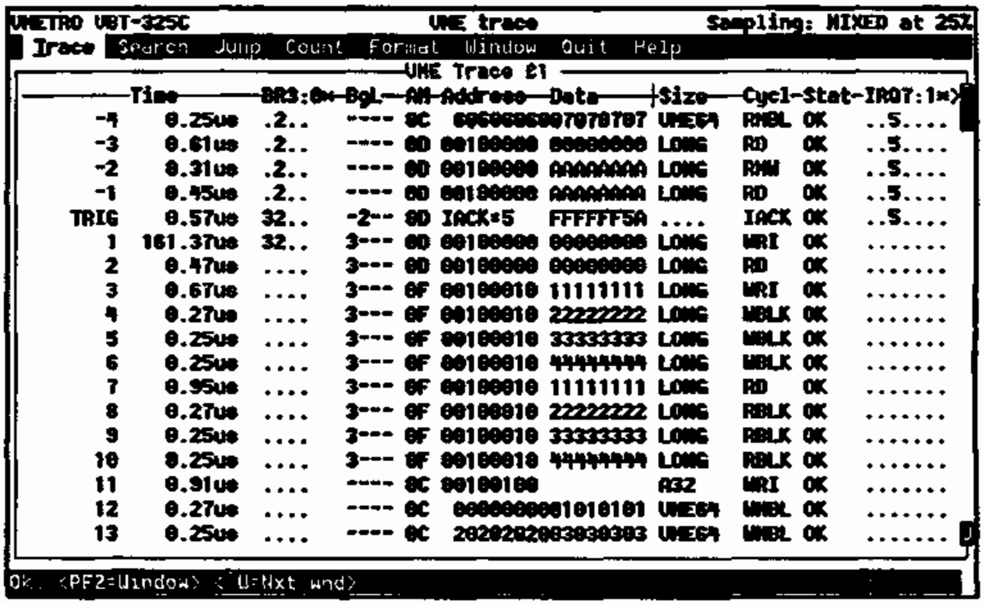
ACK*	**ACK***
ERR*	**ERR***

Other signals - Target VSB

Symbol	Description
VMEtrg	Cross-trigger from onboard VMEbus analyzer.
StateMode	Indicates that sampling mode is 'State' when high (1). When low (0), sampling mode is 'Timing'. This bit is useful if a mix of 'State' and 'Timing' sampling is used within a trace (according to Sequencer program.)

8 TRACE EXAMPLES

VMEbus



VMETRO VBT-325C — VME trace — Sampling: MIXED at 25%

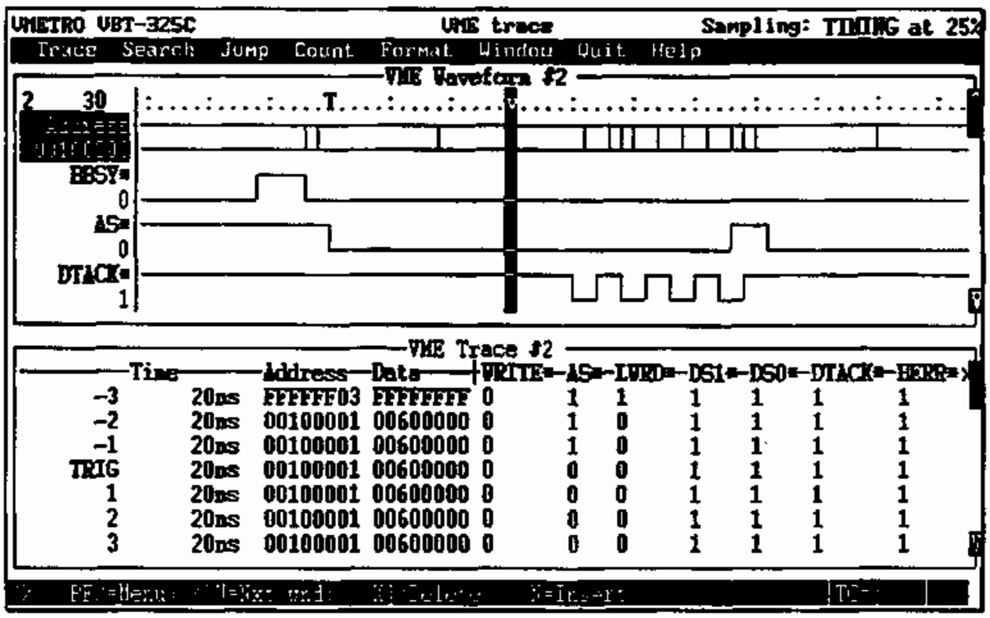
Trace Search Jump Count Format Window Quit Help

VME Trace #1

Time		BR3:0*	BgL	AM	Address	Data	Size	Cycl	Stat	IRQ7:1*
-4	0.25us	.2..	----	0C	0000000007070707		UMEG4	RMBL	OK	..5....
-3	0.61us	.2..	----	0D	00100000	00000000	LONG	RD	OK	..5....
-2	0.31us	.2..	----	0D	00100000	AAAAAAAA	LONG	RMW	OK	..5....
-1	0.45us	.2..	----	0D	00100000	AAAAAAAA	LONG	RD	OK	..5....
TRIG	0.57us	32..	-2--	0D	IACK*5	FFFFFF5A	....	IACK	OK	..5....
1	161.37us	32..	3---	0D	00100000	00000000	LONG	WRI	OK	....
2	0.47us	....	3---	0D	00100000	00000000	LONG	RD	OK	....
3	0.67us	....	3---	0F	00100010	11111111	LONG	WRI	OK	....
4	0.27us	....	3---	0F	00100010	22222222	LONG	WBLK	OK	....
5	0.25us	....	3---	0F	00100010	33333333	LONG	WBLK	OK	....
6	0.25us	....	3---	0F	00100010	44444444	LONG	WBLK	OK	....
7	0.95us	....	3---	0F	00100010	11111111	LONG	RD	OK	....
8	0.27us	....	3---	0F	00100010	22222222	LONG	RBLK	OK	....
9	0.25us	....	3---	0F	00100010	33333333	LONG	RBLK	OK	....
10	0.25us	....	3---	0F	00100010	44444444	LONG	RBLK	OK	....
11	0.91us	....	----	0C	00100100		A32	WRI	OK	....
12	0.27us	....	----	0C	0000000001010101		UMEG4	WMBL	OK	....
13	0.25us	....	----	0C	2020202003030303		UMEG4	WMBL	OK	....

Ok. <PF2=Window> <U=Nxt wnd>

Note: *The BgL (Bus Grant Level) column will show '----' when no bus grant has been detected by the VBT-325, normally the case when the analyzer is located in a slot to the right of the granted bus master. (BG daisy-chain.)*



VME Trace #2

Time		Address	Data	WRITE*	AS*	LWRD*	DS1*	DS0*	DTACK*	BERR*
-3	20ns	FFFFFF03	FFFFFFFF	0	1	1	1	1	1	1
-2	20ns	00100001	00600000	0	1	0	1	1	1	1
-1	20ns	00100001	00600000	0	1	0	1	1	1	1
TRIG	20ns	00100001	00600000	0	0	0	1	1	1	1
1	20ns	00100001	00600000	0	0	0	1	1	1	1
2	20ns	00100001	00600000	0	0	0	1	1	1	1
3	20ns	00100001	00600000	0	0	0	1	1	1	1

VSB

VMETRO VBT-325C   VSB trace   Sampling: STATE at 25%

Trace Search Jump Format Window Quit Help

VSB Trace £1

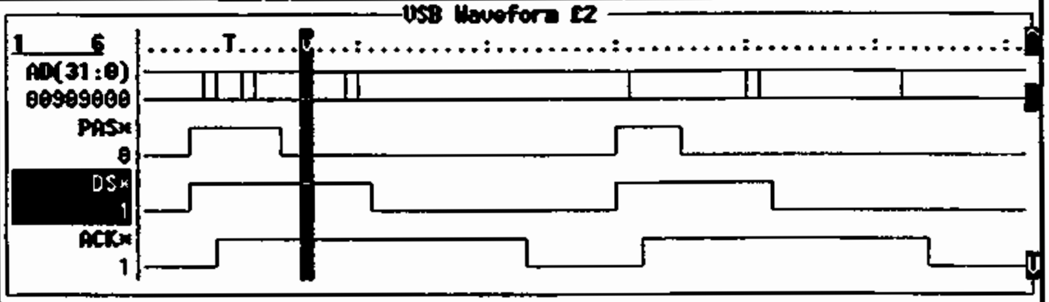
	Time	SPACE	DS*	AD(31:0)	SIZE	ASACK*	Cycle	Status	LOCK*	CACHE*
-2	12.86s	SYS	ADDR	00064000	LONG	D32	READ	-	1	0
-1	0.33us	SYS	DTA	FFFFFFCF	LONG	D32	READ	OK	1	0
TRIG	3.216s	SYS	ADDR	00064000	LONG	D32	WRITE	-	1	0
1	0.33us	SYS	DTA	11223344	LONG	D32	WRITE	OK	1	0
2	3.39us	SYS	ADDR	00064000	LONG	D32	READ	-	1	0
3	0.33us	SYS	DTA	11223344	LONG	D32	READ	OK	1	0
4	16.870ms	SYS	ADDR	00064004	LONG	D32	READ	-	1	0
5	0.33us	SYS	DTA	FFFFFFFF	LONG	D32	READ	OK	1	0
6	12.52s	SYS	ADDR	00087766	WORD	D32	READ	-	1	0
7	0.33us	SYS	DTA	0000	WORD	D32	READ	OK	1	0
8	616.03ms	SYS	ADDR	00087768	WORD	D32	READ	-	1	0
9	0.31us	SYS	DTA	0000....	WORD	D32	READ	OK	1	0
10	24.60s	SYS	ADDR	00064000	BYTE	D32	READ	-	1	0
11	0.33us	SYS	DTA	11......	BYTE	D32	READ	OK	1	0
12	448.25ms	SYS	ADDR	00064001	BYTE	D32	READ	-	1	0
13	0.33us	SYS	DTA	..22....	BYTE	D32	READ	OK	1	0
14	408.70ms	SYS	ADDR	00064002	BYTE	D32	READ	-	1	0
15	0.31us	SYS	DTA	....33..	BYTE	D32	READ	OK	1	0

Ok. <PF2=Window> <^W=Nxt wnd>

VMETRO VBT-325C   VSB trace   Sampling: TIMING at 25%

Trace Search Jump Format Window Quit Help

VSB Waveform £2



VSB Trace £3

	Time	SPACE	DS*	AD(31:0)	SIZE	ASACK*	Cycle	Status	LOCK*	CACHE*
-3	20ns	11	1	90004EF9	00	00	111	10	1	0
-2	20ns	11	1	9F9FFEF9	00	00	111	10	1	0
-1	20ns	11	1	FFFFFFFF	00	11	110	11	1	1
TRIG	20ns	11	1	FFFFFFFF	01	11	010	11	1	1
1	20ns	11	1	FF909000	01	11	010	11	1	1
2	20ns	11	1	00909000	01	11	010	11	1	1
3	20ns	11	1	00909000	01	11	010	11	1	1

Ok. <PF2=Menu> <^W=Nxt wnd> <<X]=Delete> <^N=Insert>

SCSI

```
VMETRO VBT-325C                    SCSI trace              Sampling: MIXED at Start
  Trace  Search  Jump  Format  Window  Quit  Help
                                 SCSI Trace #2
          Time         DBP Data Phase     |ATN* RST* REQ* SEL* ACK* BSY* ARB*
      0          0ns   1    00  DATA_IN   -    -    REQ  -    ACK  BSY  -
      1    318.13us    1    00  DATA_IN   -    -    REQ  -    ACK  BSY  -
      2    318.13us    1    00  DATA_IN   -    -    REQ  -    ACK  BSY  -
      3    318.13us    1    00  DATA_IN   -    -    REQ  -    ACK  BSY  -
      4    318.13us    1    00  DATA_IN   -    -    REQ  -    ACK  BSY  -
      5    318.13us    1    00  DATA_IN   -    -    REQ  -    ACK  BSY  -
      6    318.13us    1    00  DATA_IN   -    -    REQ  -    ACK  BSY  -
      7    318.13us    1    00  DATA_IN   -    -    REQ  -    ACK  BSY  -
      8    318.13us    1    00  DATA_IN   -    -    REQ  -    ACK  BSY  -
      9    318.13us    1    00  DATA_IN   -    -    REQ  -    ACK  BSY  -
     10    318.13us    1    00  DATA_IN   -    -    REQ  -    ACK  BSY  -
     11    318.13us    1    00  DATA_IN   -    -    REQ  -    ACK  BSY  -
     12    318.13us    1    00  DATA_IN   -    -    REQ  -    ACK  BSY  -
     13    318.13us    1    00  DATA_IN   -    -    REQ  -    ACK  BSY  -
     14    318.13us    1    00  DATA_IN   -    -    REQ  -    ACK  BSY  -
     15    318.13us    1    00  DATA_IN   -    -    REQ  -    ACK  BSY  -
     16    318.13us    1    00  DATA_IN   -    -    REQ  -    ACK  BSY  -
     17    318.13us    1    00  DATA_IN   -    -    REQ  -    ACK  BSY  -
Ok. <PF2=Menu> < W=Nxt wnd> <<X]=Del> < N=Insert>
```

9 VBAT-PB VME BUS ANOMALY TRIGGER

The VBAT-PB is an optional piggyback module for the VBT-325. This chapter gives you a functional and technical overwiev, and describes the installation and operation.

VBAT-PB Product Overview

The VME Bus Anomaly Trigger (VBAT/PB-325) is a massively parallel trigger board which recognizes violations of the VMEbus specification in real-time.

The VBAT-PB is a piggyback module for VMETRO's VMEbus Analyzers like the VBT-325, or the piggyback carrier VPC-MkII. (*)

The VBAT-PB automatically monitors all bus traffic and continuously and concurrently screens for a broad range of common timing violations of the VMEbus protocol. Each violation posts a message to the VBT-325 control screen and generates a fast trigger output within 20 to 80 nanoseconds, giving the engineer a direct lead to timing problems etc. on the VMEbus. This powerful broad-coverage trigger is sent to the VBT-325 and can be incorporated into its event definitions as the VBAT signal. One of the VBAT trigger outputs, or the VBT-325 Trigger Output, can then trigger supplementary test equipment such a an oscilloscope for immediate viewing of the problem.

(*) The VBAT-PB will also fit on the VBT-321, and on the piggyback carrier VPC. For use with these carrier boards, refer to the VBT-321 User's Manual.

Finds incompatibilities

The VBAT aids designers and integrators of VME boards and systems by finding serious latent non-compliance errors in new products. It also finds violations that cause common "incompatibilities" between VME boards from different vendors.

The VBAT will find design errors by watching every bus cycle during actual operation. There are 104 asynchronous rule-based trigger elements that continuously and simultaneously screen 98 VMEbus lines, to detect 28 classes of VMEbus timing violations. It detects any data, address and strobe lines that are not stable during the required intervals as well as strobes that are shorter than allowed or have improper timing relationships with other signals. Finally, by recognizing timing edges that are out-of-sequence, the VBAT finds violations of the important VMEbus arbitration, data transfer and interrupt protocols.

The VBAT can be used as a partial "non-compliance detector", as it uncovers design, manufacturing and field-failure-induced flaws in portions of the bus interface circuitry of both VME masters and slaves. It will also uncover a wide variety of hardware problems, originating in other parts of a VME board, that indirectly cause illegal bus activity. It does this by watching the boards in question during real-world use in the actual system configuration.

A VBT-325 with a VBAT-PB can for example help determine why a board from a particular manufacturer does not function properly in a system from another manufacturer.

Example of violation found

Below is an example of a "RULE 3.10 VIOLATION", i.e. the VBAT-PB recognizes a master illegally rescinding BBSY* before the bus grant goes high, often caused by ringing on BBSY* as shown.

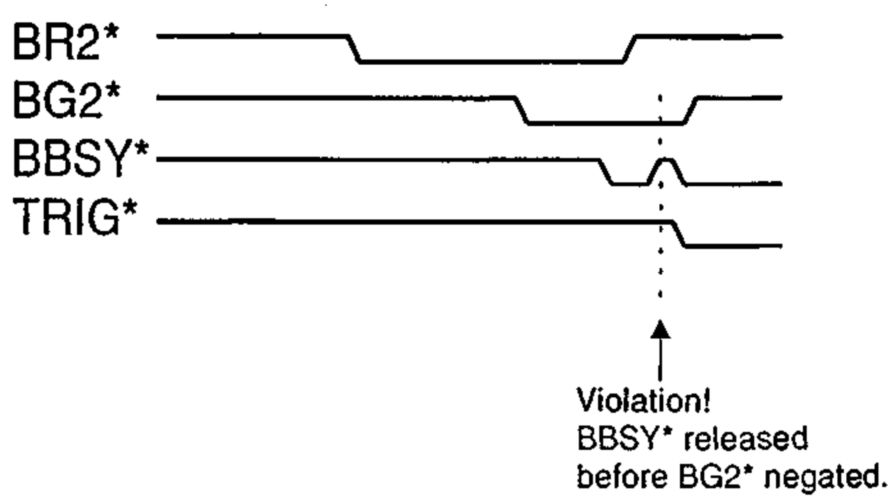


Figure 40. Rule 3.10 violation

Mask violations

The VBAT-PB may be programmed to mask out "innocent" VMEbus violations, i.e. violations that occur sporadically, and generally do not restrict the overall system performance. Using this feature to reduce "noise", it allows you to look for the VMEbus violations that really cause the problems.

Complements the VBT-325

One of the most difficult parts of debugging solely with a bus analyzer like the VBT-325 is determining on what to trigger. Often the symptom of failure gives no clue as to its cause. The 104 parallel triggers of the VBAT continuously comb the bus for illegal activity, and can often establish an immediate trigger on extremely obscure bugs. This combination of the VBAT-PB and VBT-325 is a very powerful troubleshooting aid, eliminating the trial and error approach often required to devise a suitable triggering strategy. This makes the VBAT-PB ideal for field service/customer support purposes. The VBAT-PB is even fast enough to trigger some oscilloscopes on repetitive violations.

The capabilities of the VBAT significantly enhance the utility of the VBT-325. While the VBT-325 is monitoring the bus for defined states, the 104 concurrently active trigger elements of the VBAT screen for violations of the bus specifications which may not be detected by other means. It is through such sheer "brute force" fault coverage that the automatic triggering of the VBAT becomes practical for debugging. Also, its asynchronous triggers find faults of much shorter duration than that can be detected by the triggering circuitry of the VBT-325. Finally, the edge-sequence-rule-based triggering nature of the VBAT recognizes events that cannot be directly detected using the VBT-325 alone. For example, the VBAT-PB automatically recognizes "instability", (i.e. changes on a line) on all address and data lines at once, without the need to be told the correct state of these lines beforehand. This is essential for concurrently screening for all address and data stability violations in all VMEbus cycles. This capability further enables the VBAT-PB to detect extraneous transitions on strobe lines due to metastability and bus ringing and noise.

Related documents

For more details on the VMEbus behaviour for the possible VMEbus violations that can be found by the VBAT-PB (including timing waveforms), please also refer to the following documents:

"Ultraview, VBAT User's Manual, June 1988"

"The VMEbus Specifications, IEEE STD1014-1987".

VBAT-PB Features

- The VBAT-PB is passive, and therefore does not alter, in any way, the operation of the VME bus based system. The VBAT-PB merely watches bus activity, and does not drive any lines.
- Concurrently screens 98 lines for 28 classes of violations.
- Finds serious latent non-compliance error in new VME boards.
- Ends finger-pointing by isolating the true cause(s) of "incompatibilities" between boards from different vendors.
- Triggers on many deep bugs in VME boards that are indirectly visible at the bus, *without* the need to devise a trigger strategy.
- Fully automatic. 104 preset triggers eliminate the need to specify trigger words or attach probes.
- Asynchronous edge-sequence triggers isolate events too short or complex to reasonably trigger conventional analyzers.
- Detects complex arbitration, interrupt and data transfer problems.
- Descriptive Error Messages displayed on the VBT-325 Control Screen.
- Separate trigger outputs for all violations, plus OR'ed master output.
- Detects extra transitions on strobe lines caused by metastability or bus ringing, crosstalk and ground bounce.
- Identifies memory boards (and other slaves) which drive DTACK* before providing valid data.
- Screens all types and sizes of VMEbus data transfer cycles except VME64 and SSBLT.
- Real-world use reveals that a large number of VME systems have violations, most of which are simple to rectify.

VBAT-PB Specifications

Violations screened for

1. Data lines D31-D00 changing when they should be stable. (VMEbus specification rules 2.16, 2.48, 2.54, 2.56, 4.36,4.38)
2. Address lines A31-A01 changing when they should be stable. (Rules 2.30, 2.40, 4.18)
3. Multiple bus grants (more than one at a time)
4. Bus request line(s) aborted prior to BBSY* or bus grant active.(Part of Rule 3.11, also caused by Rule 3.13 violation)
5. Illegal combination of DS0*, DS1*, A01, LWORD* (Rule 2.1)
6. WRITE* line unstable when it should be stable (Rules 2.38, 2.49, 4.23 and 4.31)
7. Address Modifiers AM5-AM0 changing when they should be stable. (Rules 2.30 and 2.40)
8. IACKIN asserted while DTACK* or BERR* is low (Rule 4.45)
9. DS0,1* asserted before DTACK* high (Rules 2.35, 4.20)
10. IACK* asserted before bus granted (Part of Rules 2.28, 4.16)
11. WRITE* asserted before bus granted (Part of Rule 2.28)
12. LWORD* asserted before bus granted (Part of Rules 2.28, 4.16)
13. DS0,1* asserted before bus granted (Part of Rules 2.28, 4.16)
14. AS* asserted before bus granted (Rules 2.20, 4.16)
15. New bus grant generated before BBSY* rescinded (Rule 3.6)
16. DS0,1* rescinded before DTACK asserted (Rules 2.17, 4.30)
17. AS* removed before last falling edge on DTACK* (Rules 2.44, 4.27)
18. Excessive skew between DSA*, DSB* assertion (Rules 2.39, 4.24)
19. BBSY* rescinded before bus grant goes high (Rule 3.10)
20. Insufficient duration of BBSY* - <90ns (Rule 3.7)
21. Insufficient duration of AS* (Rules 2.45, 4.28)
22. IACKOUT* lingers more than 30ns after AS* ends (Rules 4.41, 4.47)
23. IACKOUT* driven low less than 40ns after DSA* (Rule 4.46)
24. +5 Volt power dips below 4.85 Volts.
25. SYSRESET* high before +5V power reaches 4.85V (part of rule 5.1)

26. Interrupt request line(s) aborted prior to interrupt acknowledge cycle (Rules 4.5 and 4.6)
27. Insufficient DS* deassertion time (Rules 2.37, 4.22)
28. Insufficient AS* deassertion time (Rules 2.31 and 4.17)

Note: *All above rules are defined in detail later in this manual ("Violations descriptions"), and in the document*

"Ultraview, VBAT User's Manual, June 1988"

Non violations indicated

1. Daisy-chained bus grant(s) hidden from VBAT-PB during error - unable to fully examine bus arbitration functionality.

Trigger characteristics

1. Master active-low trigger. Separate, active-low, trigger outputs for each violation on socket on the board.

2. Trigger speed:
Asserted within 100 ns of fault

4. Trigger lockout modes:
A. If more than one fault, all faults indicated ("Accumulate" mode).
B. If more than one fault, only first fault generally indicated, if possible, later faults locked out for clearer isolation of the instigating violation. ("Lock on first" mode).

5. Trigger reset modes:
A. Manual - Trigger latched until reset via command.
B. Automatic - Trigger pulse is reset approximately 40 us after occurrence, in order to trigger an external instrument like an oscilloscope.

6. Trigger mask modes:
A. All violations screened [default].
B. Mask out violations that should not be screened.

Power requirements

+5V DC: 4.6A max, 3.0A typical

(In addition to the VBT-325 or VPC-MkII on which it is mounted.)

Possibilities and limitations

What the VBAT can do

The following are examples of problems the VBAT-PB can help uncover:

1. Improper design of a board's bus interface circuitry.
2. Meta-stable conditions in the arbitration circuitry.
3. Errors in the interrupter circuitry.
4. Bus noise, including crosstalk, ringing and ground-bounce.
5. Two masters "fighting on the bus".
6. Two slaves responding to the same address.
7. Slave boards which do not respond with stable data at the time they assert DTACK*.

What the VBAT cannot do

The VBAT-PB cannot be used as a completely "Definitive bus compliance validator", since it does not check for all possible bus specification violations. Also, while the VBAT-PB triggers on most significant violations of bus timing; a signal only out of spec by several nanoseconds may not trigger it.

The VBAT-PB cannot detect all violations reliably unless it is able to monitor all bus signals, including the daisy-chained bus grant and interrupt acknowledge signals. The VBAT-PB's ability to detect some errors is therefore depending on its physical position in the VMEbus system. It is therefore recommended that the VBAT-PB is placed as close to slot #1 as possible. The impact this has on the individual violations is described in the chapter "Violation Description".

The VBAT-PB cannot detect bugs which do not, at least indirectly, cause illegal bus timing on the signals lines tested by the VBAT. Fortunately, a wide variety of hardware design bugs in VME boards *do* affect the bus timing in such a manner as to be detectable by the VBAT.

While the VBAT, when properly used, is believed to be free of any deficiencies which could indicate false errors, an error indication by the VBAT-PB should never, by itself, be used to implicate a vendor. All errors reported by the VBAT-PB should be verified by actually viewing the bus activity causing the error by using an oscilloscope or a high speed logic analyzer (like VMETRO's TIM200-PB 200MHz Timing Analyzer piggyback). This is extremely easy to do, since the VBAT trigger will point to the precise time the error occurs. Only on the basis of such subsequent confirmation of the existence of errors should any vendor be presumed to be at fault.

Please refer to the section "Operating Instructions" for the proper methods for preventing false error indications.

Arbiter location

Backplane If the system arbiter is located on the backplane itself, the VBAT should be located in the very FIRST slot on the bus.

Slot #1 board If the system arbiter is located on the master board in the first VMEbus slot, then the VBAT should initially be located in bus slot 2. In this case certain errors may be falsely indicated by the VBAT, as discussed later.

Onboard Arbiter

The VBAT-PB has a built-in VMEbus priority arbiter (PRI, four levels). If the system has no arbiter already, the Arbiter Enable/Disable Switches should be in the Enable position.

Note: *Make sure that the Arbiter switches on the VBAT-PB are in the Disable position if another arbiter is to be used. (see Switch Settings).*

VBAT-PB Installation

1. Mount the VBAT-PB onto the piggyback connector pins of the VBT-325 or VPC-MkII according to the figure below. (If you have a VPC not of type MkII (i.e. without LED display), then you should refer to the "VBT-321 User's Manual")

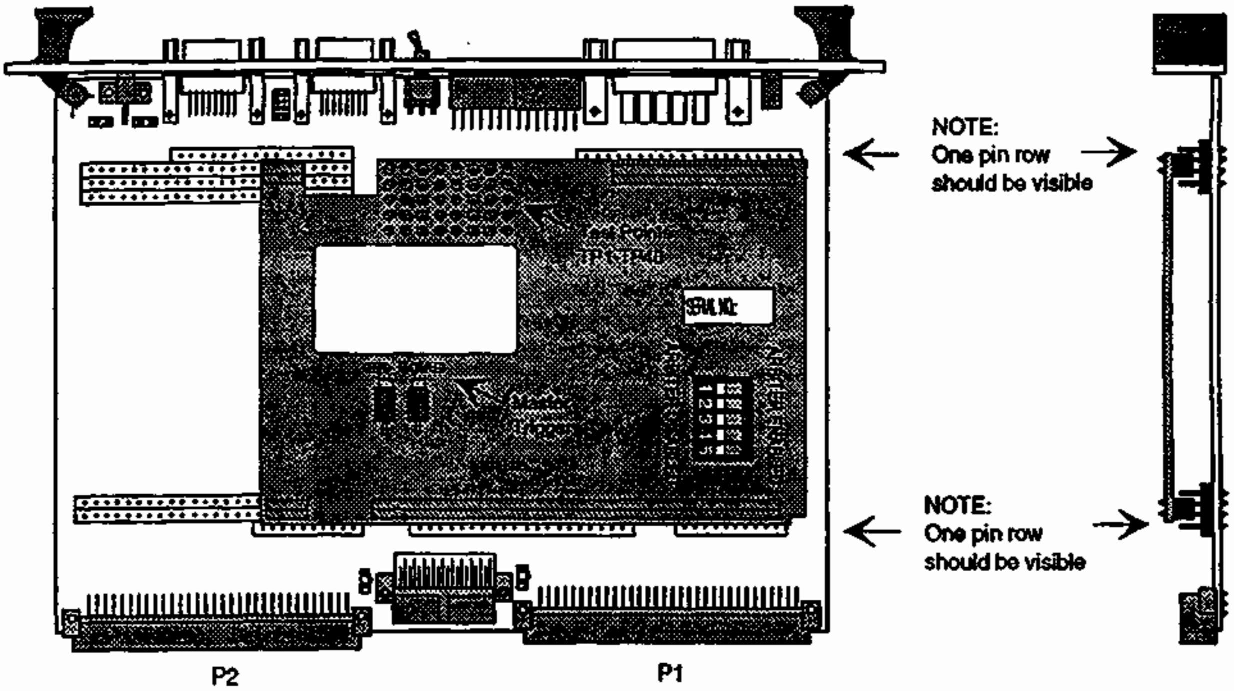


Figure 41. How to mount the VBAT-PB on the VBT-325 (or VPC-MkII)

2. Turn off the power supply in the VME system.

3. Plug the module with VBAT-PB installed into an empty slot on the bus. It is not necessary to remove or install daisy-chain jumpers for that slot - they may be left in whatever condition they were originally. Generally, the VBAT should be as close to slot #1 as possible, so that the operation of the beginning of the bus grant daisy chains will be visible to the VBT-325/VBAT, and not blocked by bus masters using the bus.

4. Connect the control terminal cable as described in the VBT-325 section of this manual.

5. Turn the power on. The reset screen should now include the VBAT-PB under "Piggyback connected".

VBAT-PB Operation

When the VBAT is properly installed and detected by the VBT-325 (or VPC), the Setup screen will include the command 'Vbat' in the main command bar, see figure below.

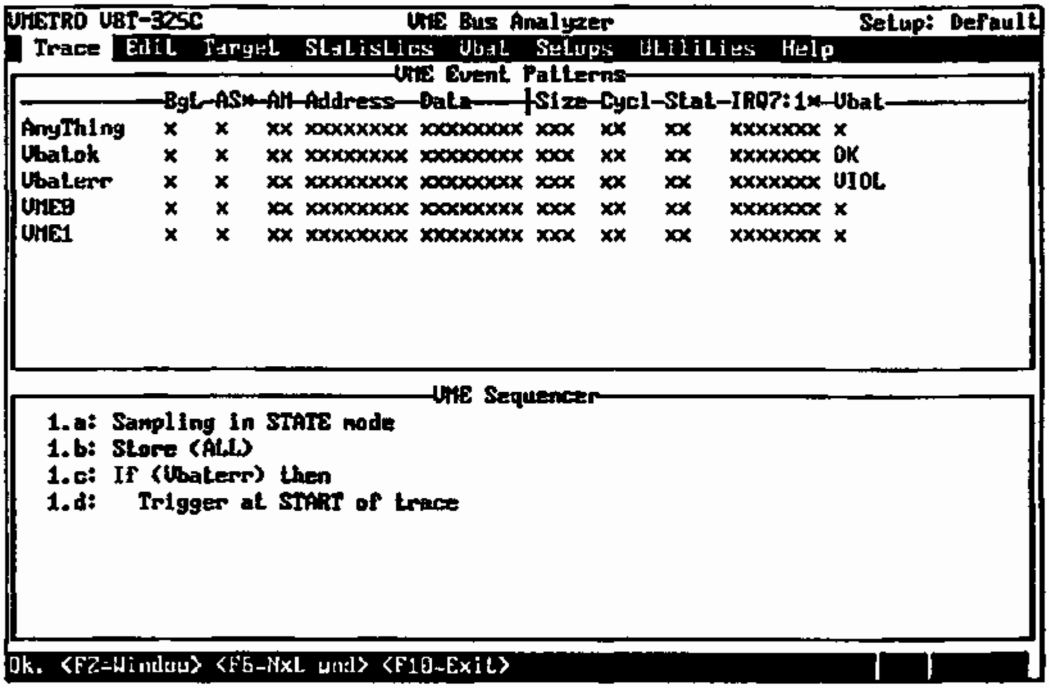


Figure 42. The Setup screen with the 'Vbat' command in the main command bar.

VBAT-to-VBT trigger signal

In addition to trigger output signals that can be used to trigger other external instruments (see later in this manual), the VBAT-PB also generates a trigger signal that is automatically fed to a dedicated input channel in the VME part of the VBT-325. As such, this signal can be part of the VBT-325 trigger, and the trigger signal is also visible in the trace.

If the VBAT is going to be used interactively together with the VBT-325 (e.g. to identify which cycles caused a VMEbus violation), it is a good idea to insert the VBAT trigger signal to the VBT-325 in the Event Patterns window at this point. This is easily done by executing the command

`'Edit/Event Patterns'`

and then typing <INS> (or '^N') to insert signals. Select 'Vbat' in the selection list and type <CR>.

The VBAT trigger signal can be selected as X (don't care), OK (no violation), or ERR (VMEbus violation found).

VBAT command screen

The operation of the VBAT-PB is controlled from a dedicated command screen. This is activated by executing the 'Vbat' command, i.e. by placing the cursor on 'Vbat' on the main command bar and typing <CR>, or simply type 'V'.

The VBAT command screen should now appear, as shown below.

```
VMETRO VBAT                    VMEbus Violation Trigger
 Violations  Mask  Modes  Explain  Quit  Help
                       VMEbus Violations

[X] D(7:0)   Unstable    [X] 2.44 AS*  off early   [X] Grant hidden (not err)
[X] D(15:8)  Unstable    [X] 2.39 DS** skew        [X] Multiple Grants
[X] D(23:16) Unstable    [X] 3.7  BBSY* on <90ns   [X] BR0* Aborted
[X] D(31:24) Unstable    [X] 2.45 AS*   on <30ns   [X] BR1* Aborted
[X] A(7:1)   Unstable    [X] 2.37 DSx*  off <30ns  [X] BR2* Aborted
[X] A(15:8)  Unstable    [X] 2.31 AS*   off <30ns  [X] BR3* Aborted
[X] A(23:16) Unstable    [X] IRQ(7:1)* Aborted     [X] 2.20 AS*    before BG
[X] A(31:24) Unstable    [X] 4.45 IACKIN* on early [X] 2.28 IACK*  before BG
[X] 2.1 Ill. sign. comb. [X] 4.41 IACKOUT* lingers [X] 2.28 WRITE* before BG
[X] WRITE*  Unstable     [X] 4.46 IACKOUT* early   [X] 2.28 LWORD* before BG
[X] AM(5:0) Unstable     [X] 5V supply <4.85V      [X] 2.28 DSx*   before BG
[X] 2.36 DSx* on early   [X] 5.1 SYSRES* off<4.85V [X] 3.10 BBSY*  off early
[X] 2.17 DSx* off early                            [X] 3.6  BG bef. BBSY* off

VBAT: Disabled.
Msg: The VBAT is disabled. Use Violations/Enable to enable it.
```

Figure 43. VBAT command screen.

The Violations command

Enable

The VBAT-PB is by default disabled when the power is turned on [or whenever the VBT-325 is reset]. It could otherwise have jammed the VBT-325 if your system suffers from repetitive VMEbus timing violations. To enable the VBAT, select the command 'Violations/Enable'

Disable

Select 'Violations/Disable' to disable the VBAT-PB if it has been enabled.

Clear

Selecting the command 'Violations/Clear' will clear all detected violations in the VBAT command screen, and the VBAT-PB will start screening for the next VMEbus violation(s). Also read about Manual and Automatic Clear later in this section.

The Mask command

With the 'Mask' command you can select the violations you will test for.

Select

If you select 'Mask/Select', you will be placed in the command window. Use the arrow keys to move around in the window, and use the spacebar to toggle on/off.

Select All

If you select 'Mask/Select All' all the violation rules will be selected.

Select None

If you select 'Mask/Select None' none of the violation rules will be selected.

The Modes command

Manual and Automatic clear

Manual With the 'Manual clear' mode selected, the trigger signal will remain active until manually cleared with the command 'Violations/Clear'. This is the recommended mode for normal troubleshooting.

```
-Clear modes-----------------
 (X) Manual clear
 ( ) Automatic clear

-Lock modes------------------
 ( ) Lock on first
 (X) Accumulate

-Violations screen attribute
 [X] Black Cockpit

 (X) High intensity
 ( ) Inverse video

      <  Ok  >    <Cancel>
```

Automatic

In the 'Automatic clear' mode, the trigger signal will automatically reset itself after approximately 40 microseconds. This option allows fast re-arming in the case that multiple occurrences of the trigger signal are to be used for triggering external equipment like an oscilloscope. Any temporary VBAT-PB violation will be visible in the trace-display for their duration.

The automatic reset mode is the recommended mode when the VBAT-PB is used together with an external instrument.

Lock on first

The 'Lock on first' mode will generally lock on the first found violation, ignoring subsequent violations. If more than one violation is found, and if the second violation occurs more than 90 nanoseconds after the first, only the violation which first occurred will usually be detected, resulting in a single Trigger signal.

Only the violation message associated with the first violation will be posted to the Error Log for display on the VBAT-PB Screen. This enables the "real initiator" of the trouble to be distinguished from subsequent events.

Note: *The following violations CANNOT be blocked, even if they occur subsequent to the first:*

BR0 ABORTED	RULE 2.35	RULE 4.41
BR1 ABORTED	RULE 2.17	RULE 4.45
BR2 ABORTED	RULE 2.44	RULE 4.46
BR3 ABORTED,	RULE 2.39	RULE 2.37
WRITE UNSTBL	RULE 2.45	RULE 2.31

Accumulate

In the 'Accumulate' mode, all violations which occur will cause the Master Trigger to toggle, and will be posted to the Error Message Log for display on the VBAT-PB Screen.

Black Cockpit

This selection is based on the philosophy from aeroplanes, where all the lights in the cockpit are off when everything is ok. Only error situations will cause light on instruments. Thus, if Black Cockpit is selected, the VBAT command screen will be blank if there is no violations. Only the violations found will be written on the screen. This is a very convenient way to get an overview of the violations at a glance.

High intensity and Inverse video

Whether you choose to display violations in High intensity or in Inverse video, depends on your screen and on your surroundings (light etc). Try and see what suit you best.

Explain violation

Whenever the VBAT-PB has detected a violation, a message will be posted in the lower left corner of the screen, and the actual violation will be highlighted. To get a detailed explanation on the reported violation, place the cursor on it, and type <CR> or 'X'.

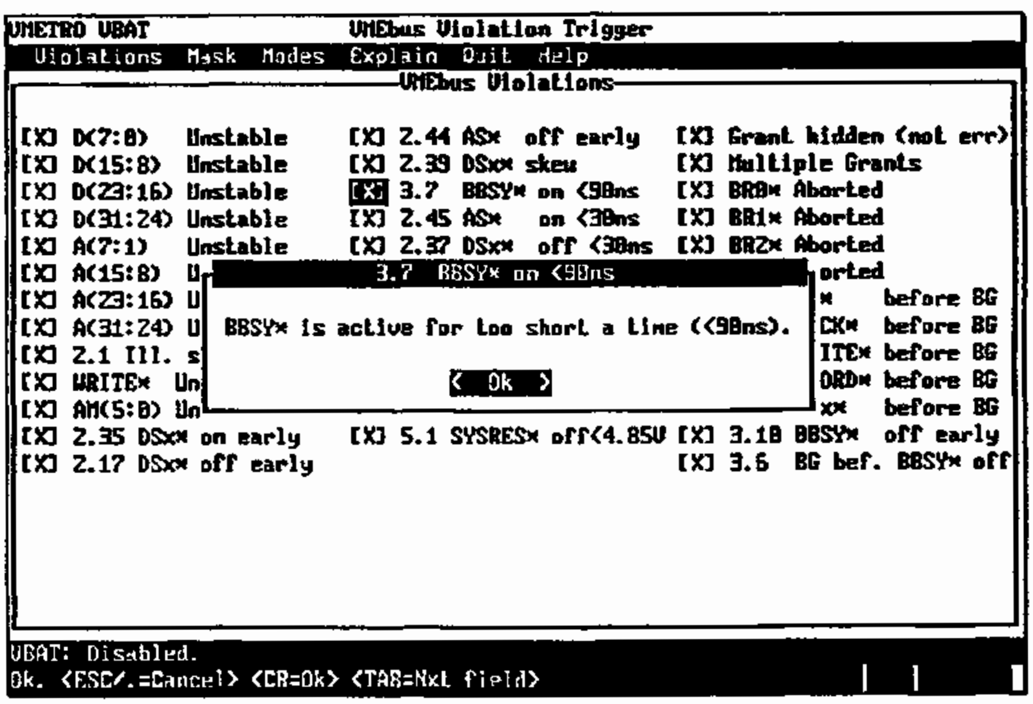


Figure 44. VBAT violations can be explained by typing <CR> or 'X' on a reported violation.

Violation descriptions

The different violations that can be detected by the VBAT-PB are described below. The location of the individual trigger output for each violation (TP1-TP40) is also given in the table. (Refer fig. 2 and silk screen on the VBAT-PB module for location of the test points).

An *H* or *F* is used some places to indicate hidden or falsely detected violations.

H Hidden violations, denotes an error which may NOT be detected if a bus grant or IACKIN/IACKOUT daisy chain, is interrupted before the appropriate BGx* or IACKOUT* signal could reach the VBAT. To remedy this, the VBAT-PB should, where possible, be located to the left of an active master which could block the daisy chain.

F Falsely detected, denotes an error which may FALSELY be detected, even though no error condition actually exists, if a bus grant daisy chain is interrupted before the appropriate BGx* signal could reach the VBAT. To prevent this [false] error indication, the VBAT-PB should, where possible, be located to the left of a master which could block the daisy chain. If not possible, and if momentary low-going trigger cannot be tolerated, it may be necessary to mask off these violations (See "Ignore violations" above.)

Violation	Test point	Description	False/ Hidden
D(7:0) UNSTB	TP1	Data lines D0-D7 change during an interval when they should be stable.	
D(15:8) UNSTB	TP2	Data lines D8-D15 change during an interval when they should be stable.	
D(23:16) UNSTB	TP3	Data lines D16-D23 change during an interval when they should be stable.	
D(31:24) UNSTB	TP4	Data lines D24-D31 change during an interval when they should be stable.	
A(7:1) UNSTB	TP5	Address lines A0-A7 change during an interval when they should be stable.	

A(15:8) UNSTB	TP6	Address lines A8-A15 change during an interval when they should be stable.	
A(23:16) UNSTB	TP7	Address lines A16-A23 change during an interval when they should be stable.	
A(31:24) UNSTB	TP8	Address lines A24-A31 change during an interval when they should be stable.	
MULTI GRANT	TP9	More than one grant (BG0* - BG3*) occurs at a time.	
BR0 ABORTED	TP10	Bus request BR0* is rescinded before BG0* occurs.	
BR1 ABORTED	TP11	Bus request BR1* is rescinded before BG1* occurs.	
BR2 ABORTED	TP12	Bus request BR2* is rescinded before BG2* occurs.	
BR3 ABORTED	TP13	Bus request BR3* is rescinded before BG3* occurs.	
RULE 2.1	TP14	Either of the following illegal combinations occur: DS1* = high, DS0* = low, A01 = high, LWORD* = low or DS1* = low, DS0* = high, A01 = high, LWORD* = low.	*H*
WRITE UNSTB	TP15	WRITE* line is not stable (i.e. changes state) during time when data strobes DS0*, DS1* are active.	*H*
AM(5:0) UNSTB	TP16	Address Modifier lines AM0-AM5, or IACK line change during an interval where they should be stable.	*H*

RULE 4.45	TP17	IACKIN* is asserted while DTACK* or BERR* is active (low).	*H*
RULE 2.35	TP18	DS0* or DS1* are driven low before DTACK* and BERR* have gone high (inactive).	*H*
RULE 2.28 IACK	TP19	IACK* is driven low before the bus has been properly granted to the master.	*F*
RULE 2.28 WRITE	TP20	WRITE* is driven low before the bus has been properly granted to the master.	*F*
RULE 2.28 LWORD	TP21	LWORD* is driven low before the bus has been properly granted to the master.	*F*
RULE 2.28 DS	TP22	DS0* or DS1* are driven low before the bus has been properly granted to the master.	*F*
RULE 2.20	TP23	AS* is driven low before the bus has been properly granted to the master.	*F*
RULE 3.6	TP24	A new bus grant is generated before BBSY* goes away.	*H*
RULE 2.17	TP25	DS0* or DS1* are rescinded before DTACK* or BERR* are received.	
RULE 2.44	TP26	AS* is rescinded before the last falling edge on DTACK* or BERR* occurs.	
RULE 2.39	TP27	Excessive skew (>20ns) between the starting edges of the first and second data strobes.	
RULE 3.10	TP28	Requester rescinds BBSY* before bus grant goes high.	*H*

RULE 3.7	TP29	BBSY* is active for too short a time (< 90ns).	
RULE 2.45	TP30	AS* is active for too short a time (less than 30ns).	
RULE 4.41	TP31	IACKOUT* line of a master is not driven high within 30ns after AS* is driven high. (Error condition is only visible for boards which are located to the left of the VBAT.)	*H*
RULE 4.46	TP32	IACKOUT* line is driven low less than 40ns after the falling edge on AS*, when the IACKIN* is low when the IACK DAISY-CHAIN DRIVER detects a falling edge on DSA. (A false positive will be indicated for this error if the VBAT-PB is in Slot 1.)	*H*
GRANT HIDDEN NOTE:	TP33	The VBAT-PB was not able to fully detect errors on the bus grant daisy chain signals BG0-BG3, because the VBAT-PB was located to the right of the active master, and the daisy chain signal was BLOCKED, NOT ITSELF AN ERROR CONDITION.	
5V TOO LOW	TP34	+5 Volt power on the bus dipped below 4.85 volts.	
RULE 5.1	TP35	During power up, the SYSRESET* signal goes away before the +5V lines go above 4.85 volts.	
IRQx ABORTED	TP36	An interrupt line IRQ1-IRQ7 is aborted before the appropriate interrupt acknowledge cycle is started.	
RULE 2.37	TP37	DS0* or DS1* are inactive for an insufficient time between successive bus cycles (<30ns).	
RULE 2.31	TP38	AS* is inactive for an insufficient time between successive bus cycles (<30ns).	
	TP39	GND	

TP40	GND

VBAT-PB Switch settings

Arbiter Enable/Disable

Switches S1-S5, mounted in the right corner of the board may be used to enable the onboard Bus Arbitration Function on the VBAT if there is no a other arbiter in the system. The normal position for these switches is in the *Disable* position.

Trigger Output Signals

Front panel Trig output

If the VBAT signal is used to trigger the VBT-325, the trigger output on the front panel (mini-coax) will go low within 100ns after a valid trigger condition (violation) has been detected. A trigger condition may consist of all the VMEbus lines, in addition to an external input and the VBAT-PB error-bit.

ERROR output on VBAT

The delay of up to 100ns introduced by the VBT-325, can in many situations be too long to trigger an external scope properly. Use the ERROR output on the piggyback itself if you are looking for short temporary VMEbus violations, or other fast transitions.

The ERROR output will go low as an immediate effect to any VMEbus violation. Note that this ERROR output is independent of the VBT-325 circuitry.

Individual trigger outputs

Individual trigger outputs for each possible VMEbus violation is available on the test-point field, TP1 through TP40. Each of these outputs will go low when the corresponding VMEbus violation occurs.

The individual trigger outputs can be used to trigger an external instrument (as a scope or a logic analyzer like VMETRO's TIM200-PB 200MHz Timing Analyzer piggyback module) on one explicit VMEbus violation.

One of the outputs may also be connected to the External Input connector of the VBT-325 if you want to trigger the VBT-325 on one distinct violation without masking off all others using the violation mask.

10 VBT-325 FIRMWARE UPGRADE PROCEDURE

This chapter describes how to upgrade the firmware of your VBT - 325.

Diskette for PCs

Firmware upgrades are distributed as a diskette for IBM-compatible PCs, to be loaded through an RS232 port on the PC. The firmware in VBT-325 is normally executed out of the onboard FLASH PROMs, and when an upgrade is done, the new firmware is copied from the distribution diskette into the FLASH PROMs by means of code resident in a Boot PROM on the board.

Note: *If the current firmware version is 4.99.05 or lower, some special HW parameters must be loaded into the board after FW upgrade. In this case, please contact VMETRO for instructions.*

Boot PROM

The Boot PROM serves two purposes: 1) Boot the board at power up and reset and transfer control to the main program which resides in FLASH PROMs, and 2) to receive new firmware through the serial port during firmware upgrades.

Normally, the Boot PROM does not need to be changed during FW upgrades. However, early in the product life time, when a major new FW release has been made, there might be a need to replace the Boot PROM. Please make sure you are using the correct Boot PROM version.

For FW version 4.99.120, the Boot PROM must be of version 2.00 or higher.

Note: *Please note that boards with FW version lower than 4.99.08 do not have a Boot PROM installed, they are normally run entirely out of FLASH PROMs alone. In these cases, the jumper J21 must be removed when a boot PROM is installed. (see figure A2).*

RS232 connection

COM1 on PC — Before starting the upgrade procedure, connect a RS232 cable from the COM1 port on an IBM compatible PC to the terminal port of the VBT-325, see recommended cable connections in the figure below. (Suitable cable can be obtained from VMETRO, part # 401-PC-232).

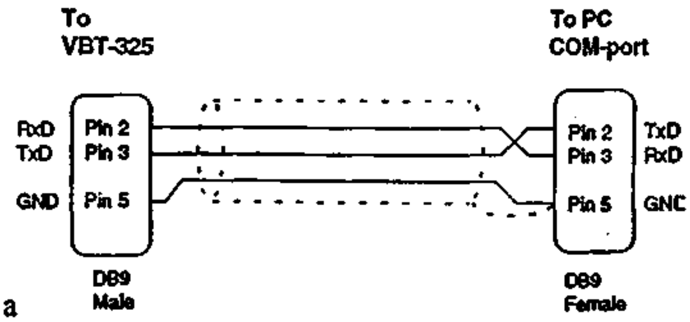


Figure A1. Recommended cable for connecting a PC to the VBT-325.

Load firmware from PC

+12V required!

Before starting the upgrade procedure, make sure that +12V is supplied to the VBT-325 (at least 120mA). (If there is any doubt whether +12V is present, use the built-in voltage monitor, giving the command 'Utilities/LED display'.)

Start the upgrade procedure by inserting the firmware distribution diskette in the floppy drive on an IBM compatible PC. (Below it is assumed that the "A:" drive is used.). Start the upgrade program on the PC by typing

```
a:↵
upload↵
```

Then follow the instructions given on the PC screen, and answer Y or N as appropriate to the questions given. After a few minutes, the new firmware is loaded.

Specify PCB and ECO Level

The firmware needs to know the PCB version and ECO-level of the VBT-325 to take full advantage of all features. If this information is not already entered into the board, you will be asked to enter the PCB and ECO level the first time after a FW upgrade. The PCB/ECO level is found on a label on the bottom side of the VBT-325, close to the serial number (e.g. B8, where B designates the PCB version and 8 is the ECO level).

ECO < B5?

Note: *The firmware 4.99.120 does not accept PCB/ECO levels lower than B5. However, if your board has PCB/ECO level B4, you may operate the board without any problems by entering B5 as the PCB/ECO level. For lower levels, please consult VMETRO for an upgrade of the board.*

The command 'Utilities/Specials/ECO Level' will allow you to enter or verify this at a later time if desired.

The VBT-325 is now ready to run, and can be operated immediately from the VT100 emulator that comes with the firmware distribution diskette, or you can move to your familiar terminal.

Setups lost

Please note that the firmware upgrade procedure will clear all contents of the non-volatile memory on the VBT-325, causing all stored setups to be lost.

This completes the firmware upgrade procedure.

* * * * *

Procedure if Boot PROM is missing or wrong version

To upgrade firmware from a PC to a VBT-325, there must be a PROM installed in the 32-pin socket on the board. If a PROM of wrong version is already installed, this must be replaced.

To install the Boot PROM and to complete the firmware upgrade procedure, please follow the instructions given below:

- Power down the system in which the VBT-325 is inserted.
- Take out the VBT-325 and leave it on a surface with controlled characteristics with respect to static electricity (ref. "Static Electricity - Precautions" in the "INSTALLATION" section.)
- If a piggyback module is installed, this must be removed before the PROM can be replaced or inserted, please refer to "Piggyback removal instructions" in the User's Manual for the piggyback modules (for the VBAT-PB, this is included in the section VBAT-PB in this manual.)
- If a PROM is already installed in the 32-pin socket, remove this carefully with a screwdriver etc. *NB: Take care not to damage the components underneath the PROM!*
- If the jumper in pos. J21, close to the big PLD device, is inserted, then remove it. (If this jumper is present, the board has been run and booted from FLASH PROMs only, without any separate Boot PROM.)
- Check that the position of the jumpers J22 and J23 are installed as shown in figure A2 below.
- Insert the new boot PROM into the 32-pin socket. If the boot PROM is a 28-pin device, leave four vacant pins closer to the pin 1 marking (see figure A2 below). The pin 1 is marked with a little "notch" in the silk

screen. The marker on the PROM should be facing "upwards", closest to the blue battery.

NB: Make sure that all pins are straight and parallel before inserting the device into the socket. It might be necessary to straighten the pins before insertion.

- Before inserting the board back into the system, make a note of the serial number and ECO level (e.g. B6 etc.) found on the labels on the underside of the board. This will be asked for after the new FW is loaded (if not already entered by VMETRO, valid for 4.99.50 and higher).
- Insert the board in the system and apply power.

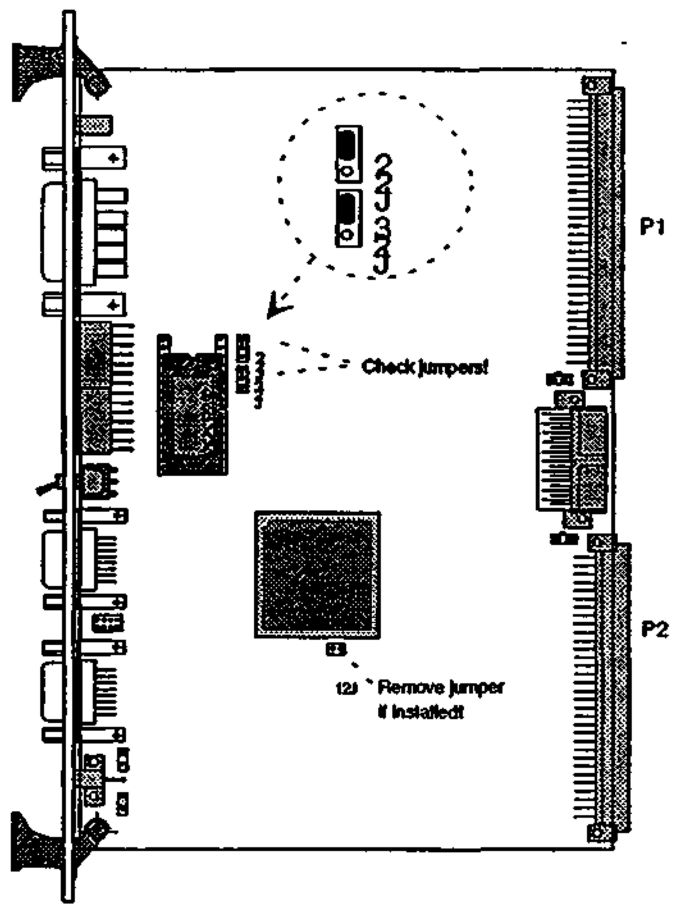


Figure A2. Boot PROM position and jumper settings.

Run from Boot PROM

Now, when the board is powered up, it should come up with one of the texts

`19k2 81n` `9600 81n`

flashing or steady in the front panel LED display, indicating that the board is running from the boot PROM and is ready to accept the new firmware, loaded through the RS232 port. Now do this according to the procedure "Load firmware from PC" given above.

No response? If the text mentioned above does not appear in the LED display, check that the Boot PROM is the right version and is inserted in the proper direction and position (see figure A2), and that no pins are bent.

11 JUMPER SETTINGS

The VBT-325 has a number of jumpers that define the operating mode (wide/twin), P2 target bus (VSB/SCSI/UserDef), and functions like Temperature probe/External input, boot PROM size etc. The jumpers have a slightly different layout for different PCB versions of the VBT-325. Use the `'Utilities/Specials/ECO level'` command to find the PCB version of your VBT-325. Refer to figure A3, A4, A5 B and C respectively for locations and setting of these jumpers.

Refer to the following chapters for jumper setting for VSB and SCSI bus.

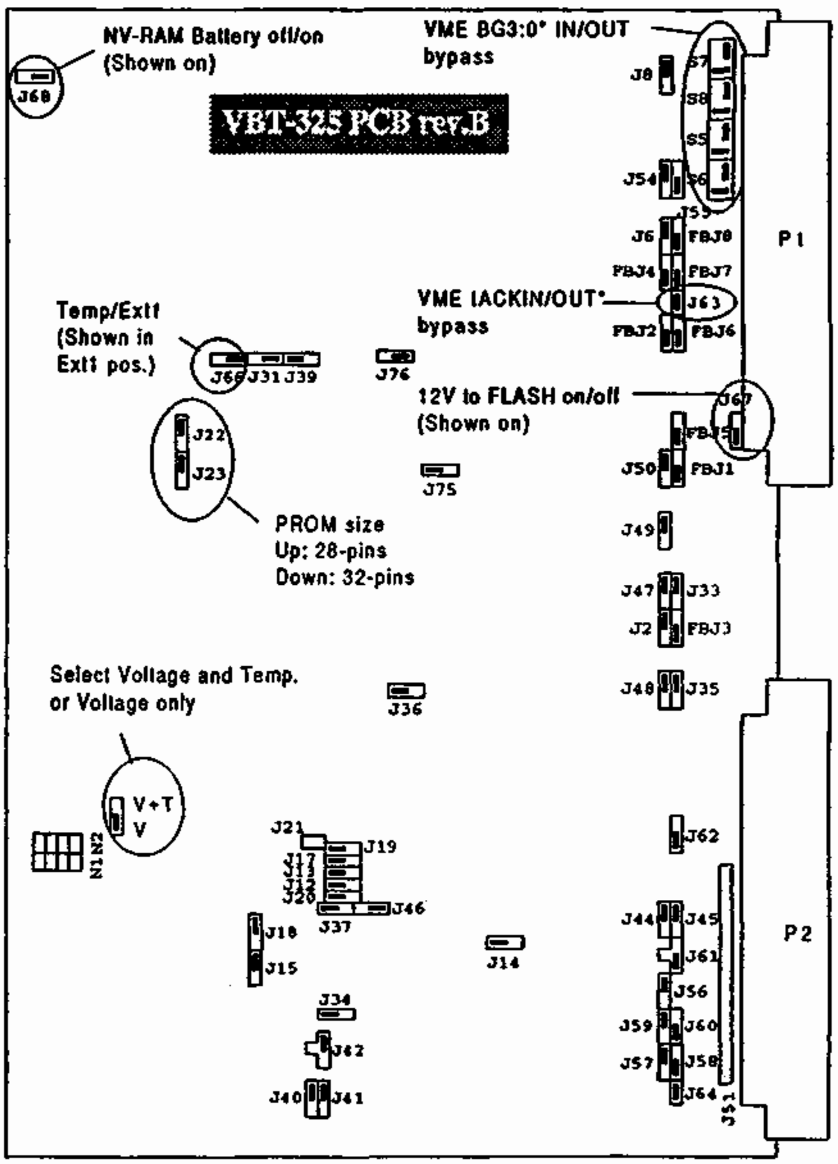


Figure A3B. Miscellaneous jumper settings VBT-325 PCB rev B.

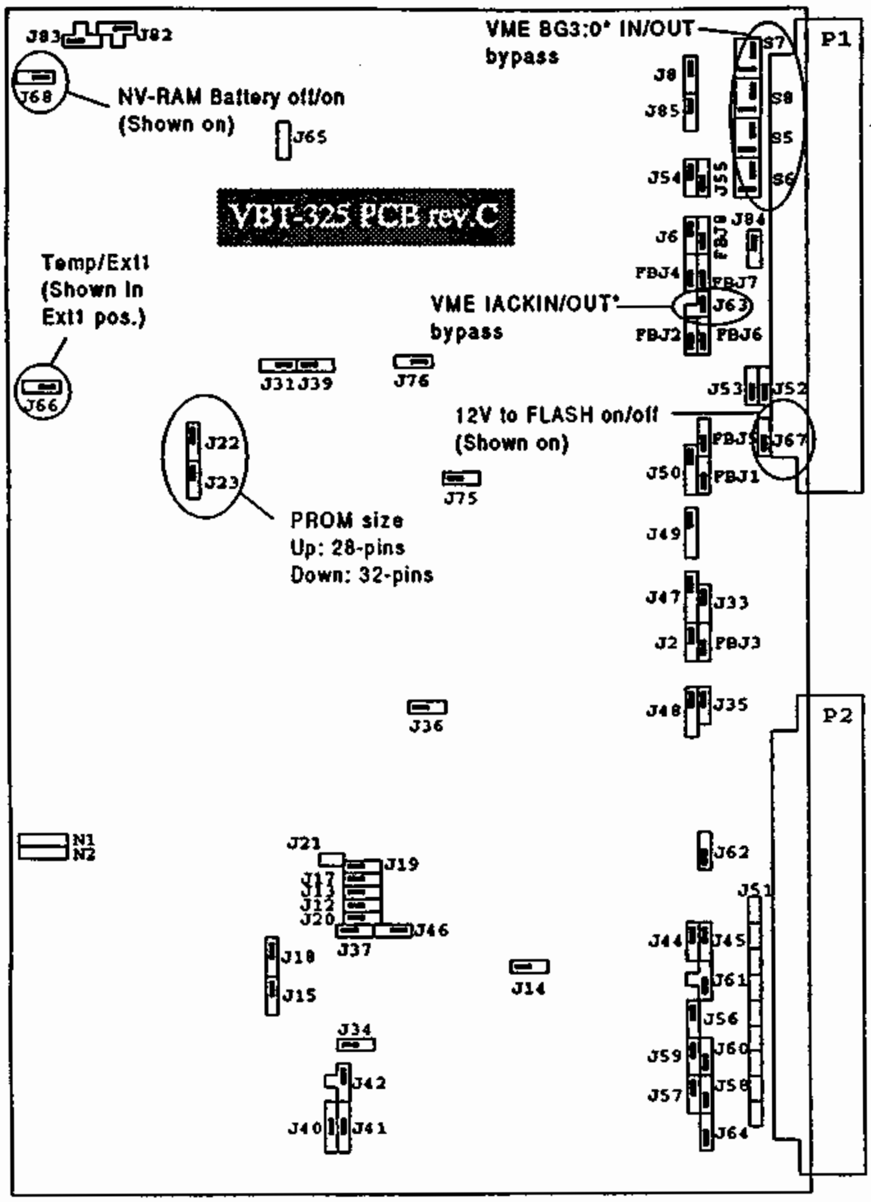


Figure A3C. Miscellaneous jumper settings VBT-325 PCB rev C.

VME & VSB (Twin mode)

VSB Bus Grant Daisy chain jumper

The VBT-325 is default configured for VSB (factory setting) except the BGIN/OUT daisy chain jumper in pos. J64 as shown. (This is to avoid damage if the board is inserted in a crate with other use of the P2 connector.)

Unless the VSB BGIN/OUT is bypassed in the backplane, a jumper should be inserted in J64 before installing the VBT-325 in a system with VSB.

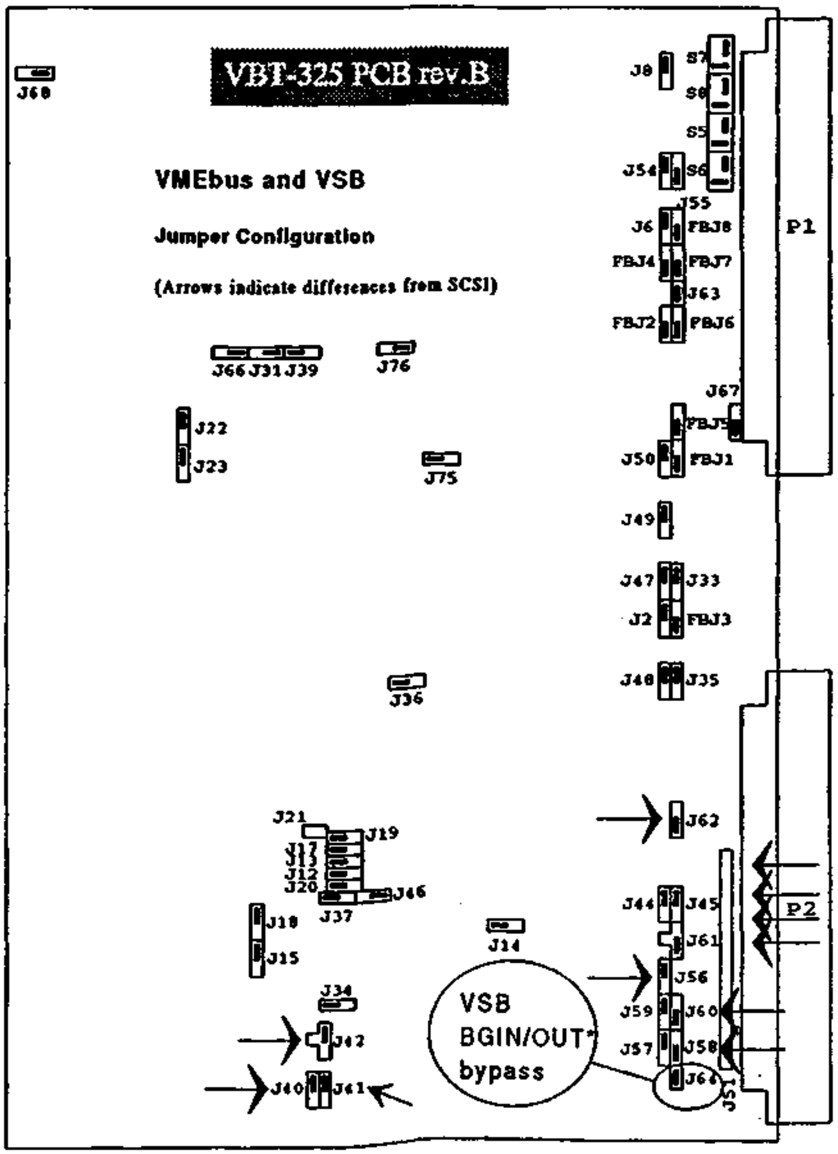


Figure A4B. Jumper settings for VME and VSB, VBT-325 PCB rev B.

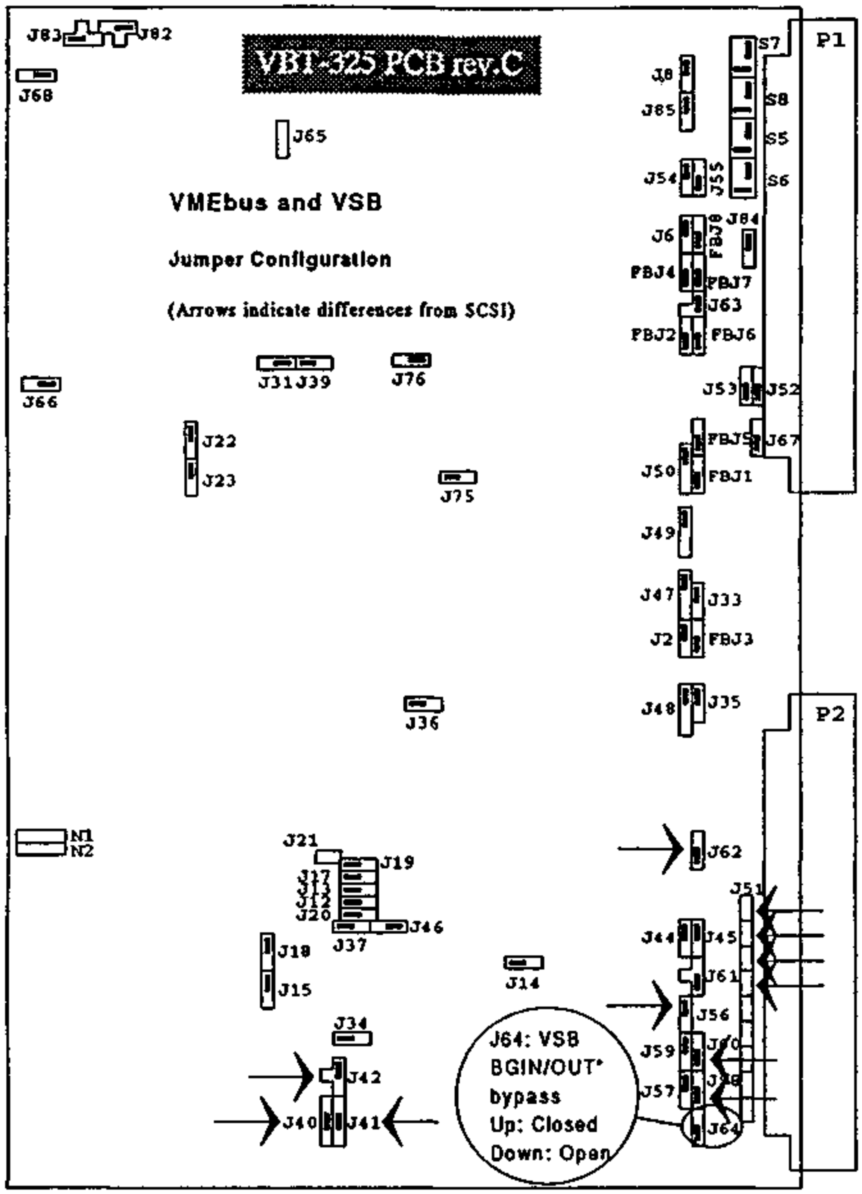


Figure A4C. Jumper settings for VME and VSB, VBT-325 PCB rev C.

VME & SCSI (Twin mode)

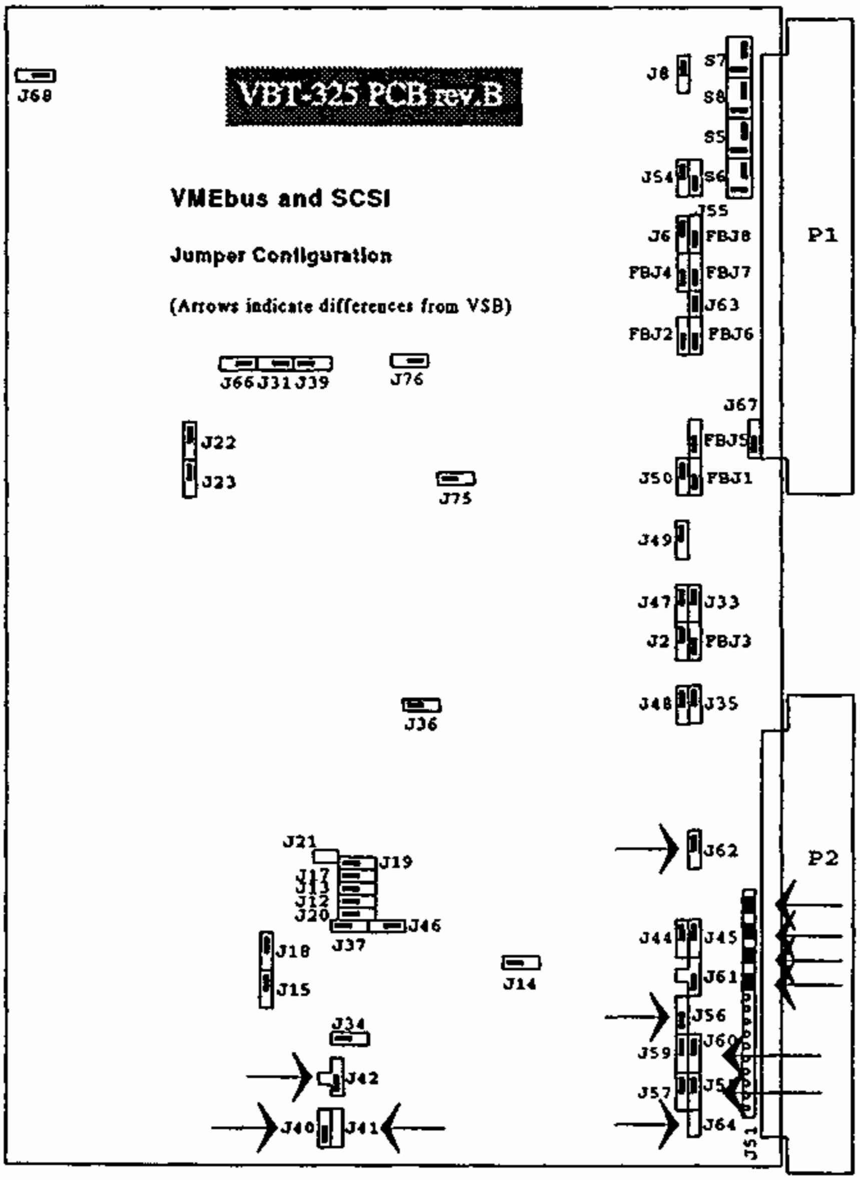


Figure A4B. Jumper settings for VME and SCSI, VBT-325 PCB rev B.

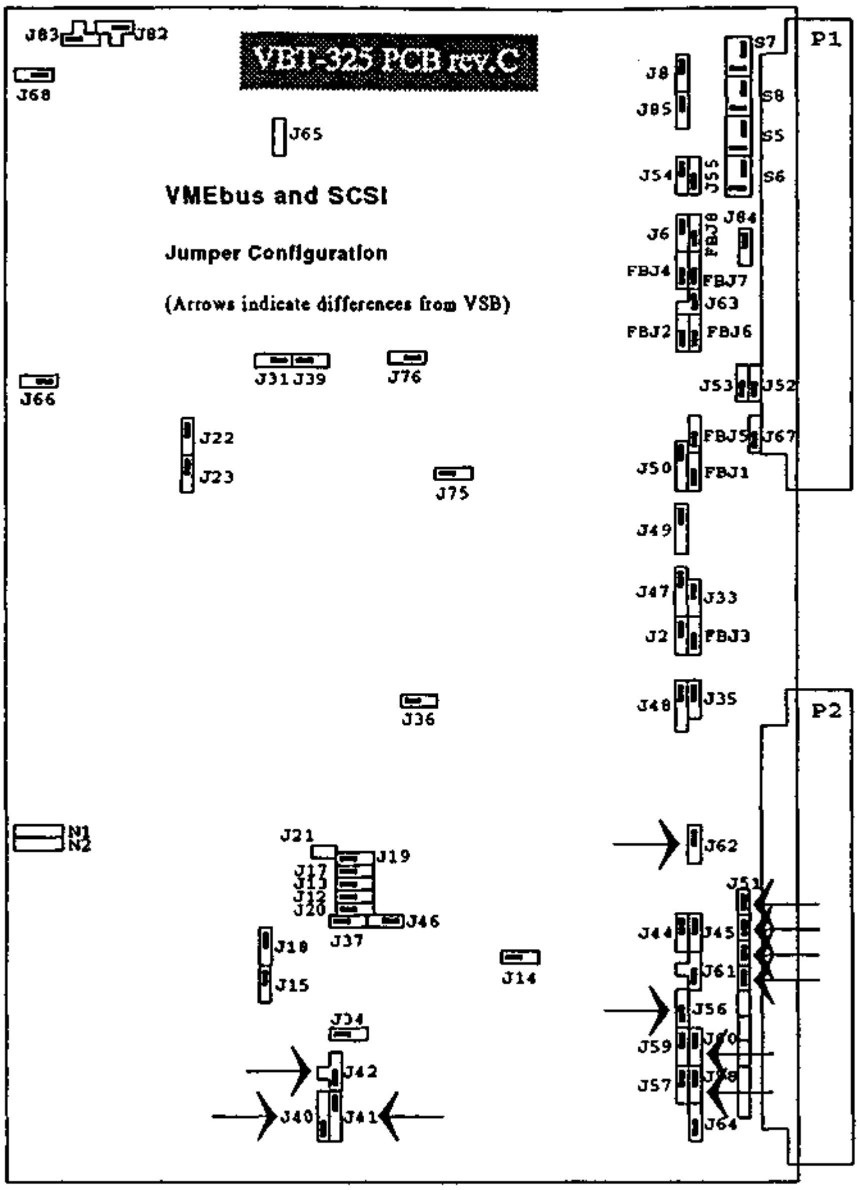


Figure A5C. Jumper settings for VME and SCSI, VBT-325 PCB rev C.

VME & User-defined P2 (Twin mode)

(The 4.99 firmware does not support user-defined P2 busses. Jumper configuration for this will be described in the 5.X User's Manual.)

Futurebus+ (Wide mode)

For Futurebus+ Analysis, the VBT-325 must be inserted into the FBA-625 adapter, and a special jumper setting must be used. Refer to separate manual for the FBT/FBA-625 product.

12 SCSI CONNECTION ON P2

There is no industry standard connection for SCSI on P2 of VME boards. This is not so strange, since the SCSI-bus is rarely (or never) *bussed* in the backplane, it goes normally only out from one board (CPU w/SCSI interface, or specific SCSI controller VME board like DVME-718). If the connection matches the VBT-325C, then the arrangement shown below can be as used for SCSI analysis.

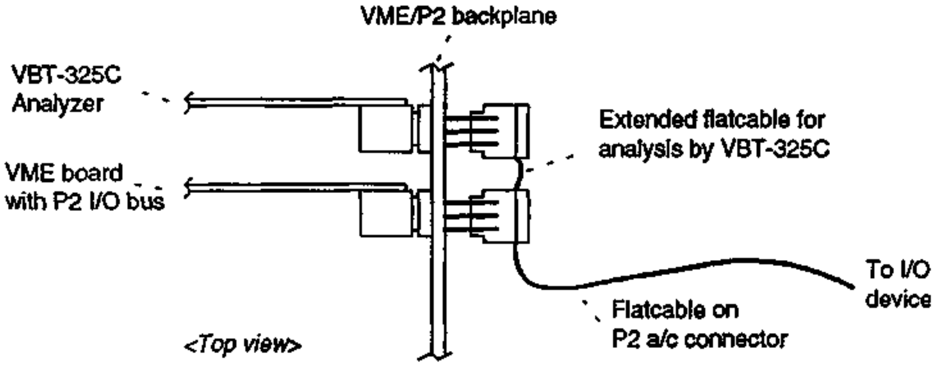


Figure A 6. Example of SCSI-connection to VBT-325 on P2 backplane.

The connection of the SCSI bus on the VBT-325C is taken from the Motorola SCSI I/F board DVME-718, and is chosen since it maps easily on to a 50-lead flatcable, as done by VMETRO's SCSI cable "VSC-8", see enclosed drawing. When this is used as shown above, it is the *cable* which creates the "bus" between the adjacent P2 connectors (on rows a,c).

For CPU boards like Motorola MVME147/167/187 etc., the SCSI connection on P2 is rather random, since it is intended to go via a "P2 paddle board", supplied by the CPU board vendor, to convert all the I/O, including Ethernet, serial etc., to industry standard connectors.

In order to connect a VBT-325C to the SCSI cable in such systems, this requires that a 64-pin DIN connector is pressed on to the existing 50 lead flatcable, or that VMETRO's "VSC-3" cable (see enclosed drawing) is used to extend an existing SCSI cable to allow it to pass the slot where the VBT-325C is located.

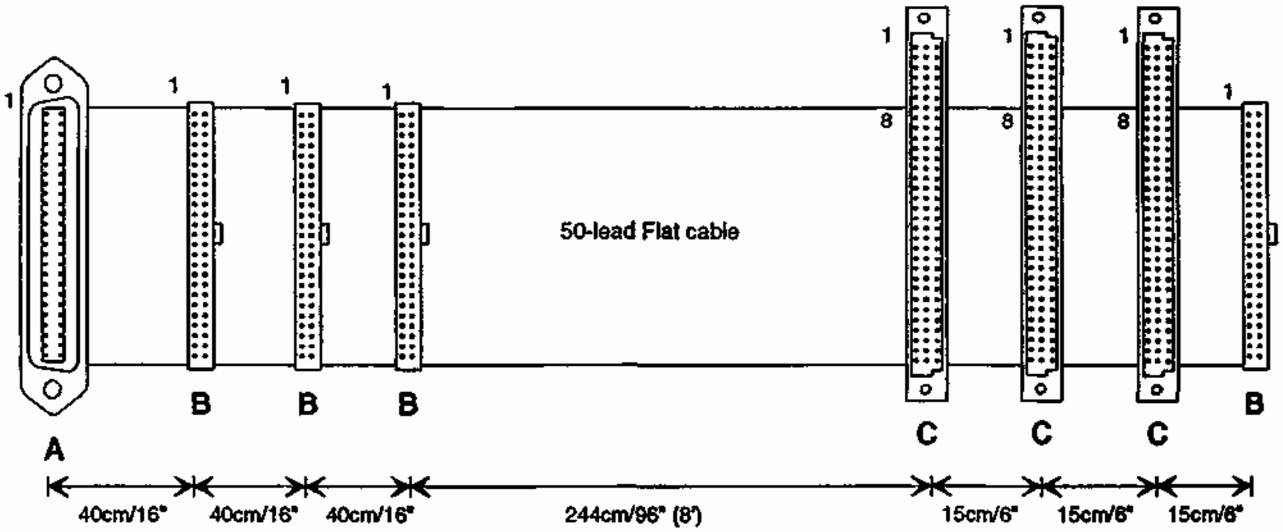
VBT-325 SCSI P2 adapter

VMETRO is developing an adapter to be plugged on the P2, where the VBT-325 connection is converted to standard SCSI and SCSI-2 plugs, including support for differential SCSI. This will provide conversion to industry standard connectors, so there will be no need to cramp a special 64-pin DIN to the SCSI cable, a standard SCSI/SCSI-2 cable which normally has a number of connectors attached to it can be used.

SCSI Connection on P2

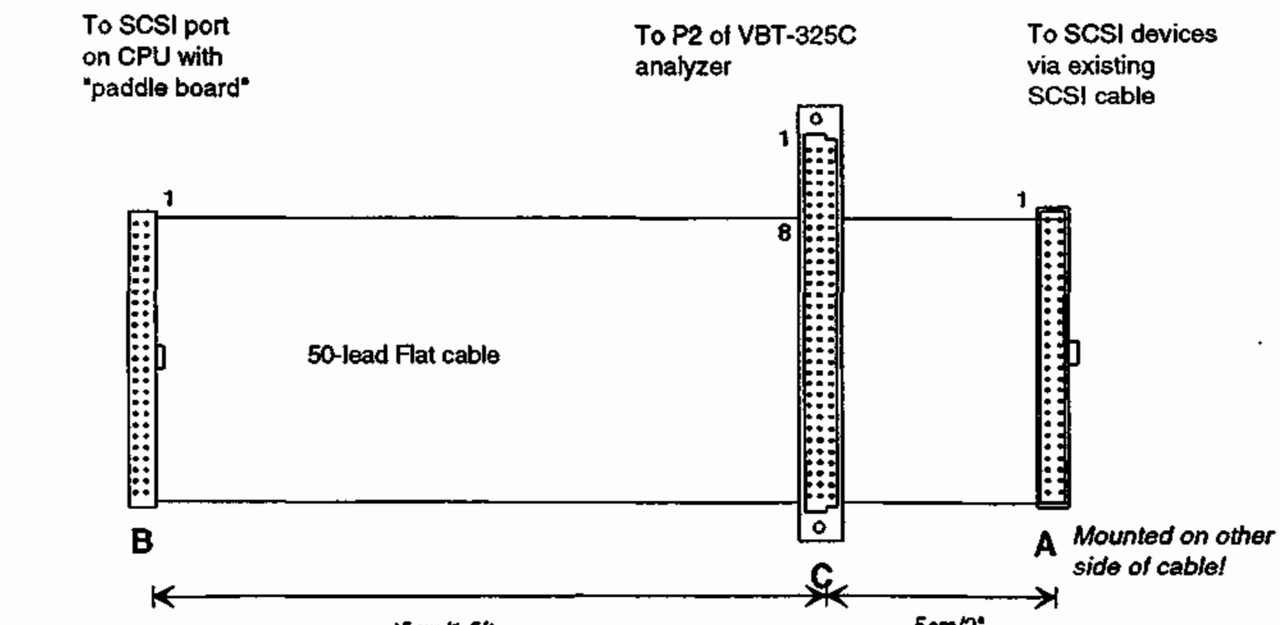
The pins shown in BOLD text are connected to the VBT-325 when the board is configured for SCSI.

Pin #	P2 row A	P2 row B	P2 row C
1			
2			
3			
4			
5			
6			
7			
8	**DB0***		Gnd
9	**DB1***		Gnd
10	**DB2***		Gnd
11	**DB3***		Gnd
12	**DB4***		Gnd
13	**DB5***		Gnd
14	**DB6***		Gnd
15	**DB7***		Gnd
16	**DPB***		Gnd
17	**Gnd**		**Gnd**
18	Gnd		**Gnd**
19	Gnd		**Gnd**
20	TermPWR		
21	Gnd		Gnd
22	Gnd		Gnd
23	**ATN***		Gnd
24	Gnd		Gnd
25	**BSY***		Gnd
26	**ACK***		Gnd
27	**RST***		Gnd
28	**MSG***		Gnd
29	**SEL***		Gnd
30	**C/D***		Gnd
31	**REQ***		Gnd
32	**I/O***		Gnd



Connectors:
Type A: 50-pin D-shape Male SCSI connector (Amphenol etc)
Type B: 50-pin Female FC-50P or similar (3M etc.) NB: With polarizer as shown!
Type C: 64-pin DIN Female with A and C row only, B row empty. Pins 1-7 unconnected.

Figure A 7.. SCSI Cable - Type VSC-8



Connectors:
Type A: 50-pin Male FC-50P or similar (3M etc.) NB: With polarizer as shown!
Type B: 50-pin Female FC-50P or similar (3M etc.) NB: With polarizer as shown!
Type C: 64-pin DIN Female with A and C row only, B row empty. Pins 1-7 unconnected.

Figure A 8:. SCSI Cable - Type VSC-3

13 CUSTOMIZED ISOLATION OF P2A/C

The input channels on the VBT-325 are designed for TTL/CMOS voltage levels only (0.0-5.0V), and **damage may occur if other voltages are applied.** If the P2 bus contains signals with voltage levels other than TTL/CMOS, (for example ECL, analogue or special power supply voltages as found in VXI systems), **these must be isolated from the VBT-325.**

(For analysis of VXI systems, VMETRO is developing an adapter that will include the necessary isolation and even ECL to TTL conversion, part # VXA-035. Available 1Q94.)

If the P2 pins cannot be isolated by means of selecting another slot or by using an isolating extender board, your VBT-325 board can be customized by cutting copper tracks that are specially laid out for all P2 signals for this purpose. These can be found as a row of 32 pairs or solder pads connected with a thin copper track on each side of the board just next to the P2 connector. The pads on the bottom side are signals from P2 row C, and the top side row A. Please refer to figure A7 on the next page for details.

NB: Cut very carefully with a thin, sharp surgeon knife, and cut the surface only to avoid damaging inner layers.

If a connection needs to be re-established, a short piece of un-isolated wire (AWG30 or similar) can be soldered between the two pads in a pair.

NB: Before performing any cuts, please contact VMETRO for approval regarding warranty.

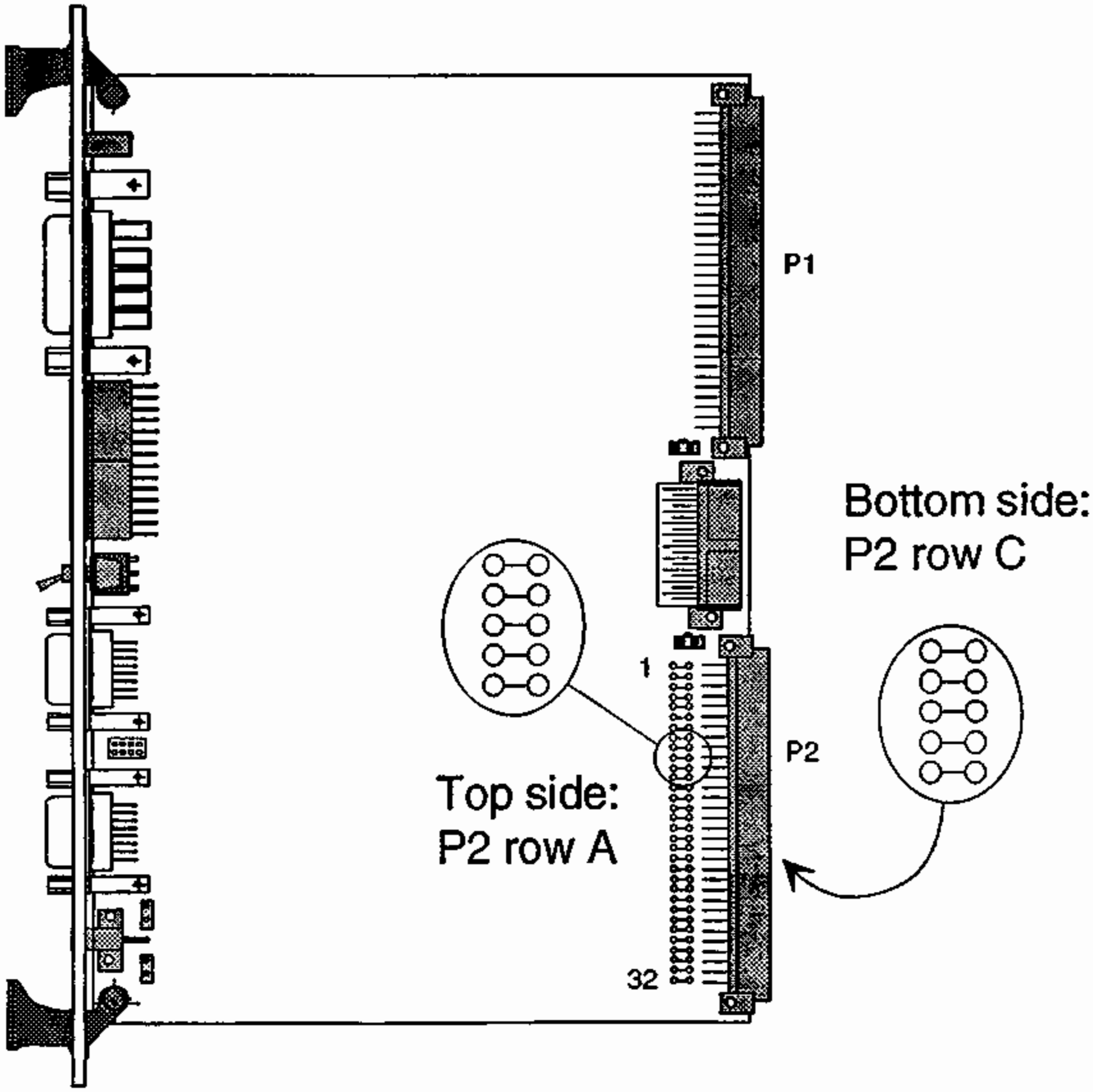


Figure A9. The VBT-325 has specially laid out tracks for P2 rows a/c that can be cut if there are pins with illegal voltage on the P2 a/c connector in the target system.

14 VMETRO VT100 EMULATOR - VT100.EXE

The program VT100.EXE on the firmware distribution diskette is a VT100 terminal emulator for IBM-compatible PCs which offers a number of valuable features that help you take full advantage of the VBT-325 product. These are:

- A VT100 emulator tailored to the VBT-325.
- Facilitates firmware upgrade through the serial port.
- Allows trace dump/load to/from file on a PC.
- Allows running command scripts to the VBT-325.

In addition to emulating a standard 24 lines x 80 character VT100 screen, the VT100.EXE also allows you to take advantage of a 50 lines x 80 character display (`' -h 50'` option).

COM1-port The VT100.EXE will default run from the COM1 port on an IBM compatible PC.

COM2-port The '-P' (capital P) option allows operation through the COM2 port, like this:

```
vt100 -P COM2 ↵
```

Make sure that your `config.sys` file includes the following statement:

ANSI.SYS

```
device=ansi.sys
```

If you do not find this or a similar statement, locate the directory where `ansi.sys` resides on your PC, and add the above statement (with correct directory path if necessary) to config.sys.

The PC needs to be rebooted to reflect changes in `config.sys`.

Start the program directly from the diskette by executing the following command:

```
a:vt100 ↵
```

or copy the `vt100.exe` file into your disk drive and start it from there.

The program will by default use the COM1 port at 9600 baud. Other baud rates can be selected by the -p option, example:

```
a: vt100 -p 19K2 ↵
```

Starts the emulator with a baud rate of 19200. To display information about all supported baud rates, do as follows:

```
a: vt100 -p ? ↵
```

Help

If an illegal option character or a question mark '?' is typed, information about all options will be displayed. E.g.:

```
a: vt100 ? ↵
```

Usable "terminal types"

If your PC has a color or monochrome VGA screen, you should select terminal type # 3. Select terminal type #2 if you have a monochrome screen (MDA display adapter).

50 lines display If you have a screen that supports 50 lines, you may select terminal type #4 and start the VT100 emulator with:

```
a: vt100 -h 50↵
```

Built-in script language

Script files are especially edited files containing VBT-325 commands and function keys which are sent to the VBT-325 exactly as they were typed on the keyboard. Special *script control commands* steer the execution of the script, making it possible to take action after interpreting how the VBT-325 response to a given command.

```
a> vt100 -i scriptf.inp ↵
```

Will start the vt100 emulator, taking input from the file `scriptf.inp` instead of the keyboard.

Script control commands

Script commands should always start the line.

^e	Turn echo off. This is the default setting.
^E	Turn echo on
^w "*string*"	The script will wait 30 seconds for the given *string*.
^u "*filename*"	Start XMODEM upload (transmit) of the given *filename*.
^d "*filename*"	Start XMODEM download (receive) of the given *filename*.
^I	Enter interactive mode. The script continues when the user type CTRL-Z.
^M "*message*"	Display *message* on the screen.
% *comment*	Treat the rest of the line as a comment. Can be placed anywhere in the text.
~	Send single CR.
string~	Send *string* followed by CR.
string&	Send *string*. Do not send CR.
$	Wait 100 ms.

script lines *not* starting with an "^", "~", or "%" will be sent to the VBT-325 *as-is*. Trailing spaces will be ignored. To end a command with a CRLF, add a tilde, "~" to the end of the string as described above.

```
% ---------------------------------------------
% upload.inp
%
^e                         % Echo off
fe~                        % Erase flash
~                          % Confirm erase
~                          % Confirm no tuning
^W "XMON >"                % Wait for command to finish
up~
^M "Uploading firmware"    % Inform user
^U "flash.bin"             % Upload file
```

Function keys in script files

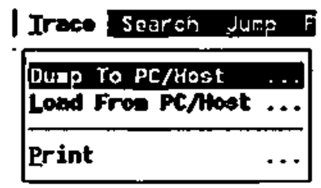
To use function keys in scripts, see the table above for which "\-command" that can be used for the different function keys. This way of specifying function keys is not specific for the `vt100.exe`, but applies also to other terminal emulators, such as PROCOMM+.

Function key in script text	Comments	Function
\U	Cursor up, ↑	
\D	Cursor down, ↓	
\R	Cursor right, →	
\L	Cursor left, ←	
\H	HOME	
\1	Function key F1	Help
\2	Function key F2	Edit last window/ go to menu
\3	Function key <F3>	(Trace Display menu only) Find next search pattern
\4	Function key F4	(Trace Display menu only) Find previous search pattern
\\		Re-display screen
\F1	Function key F1	
\F2	Function key F2	
\F3	Function key <F3>	
\F4	Function key F4	
\F5	Function key F5	Trace/Run
\F6	Function key <F6>	Edit next window
\PU	PgUp	Page up
\PD	PgDn	Page down
\I	INS	Insert object
\E	END	Goto end of line/page/trace (Not used yet)
\L	DEL	Delete object

15 DUMP/LOAD OF TRACE TO/FROM PC

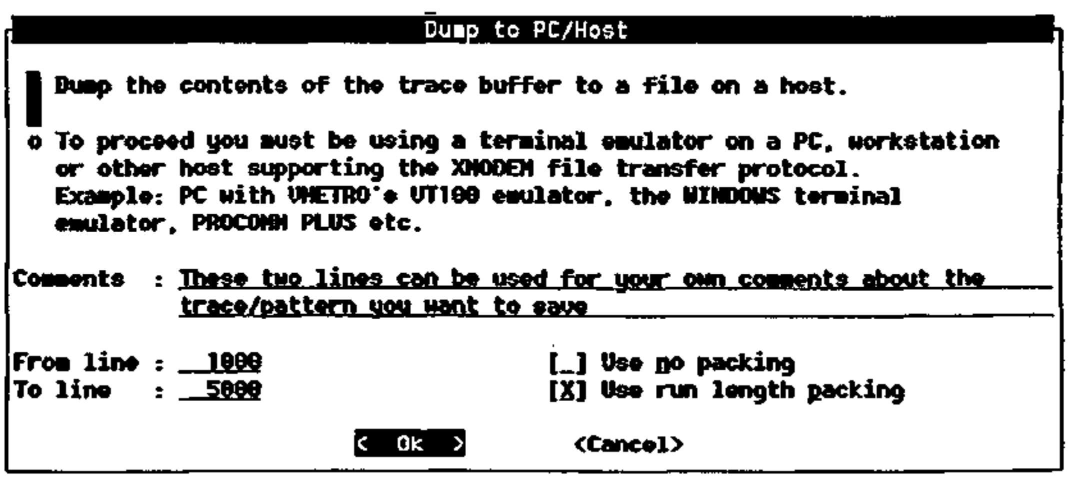
Trace data may be dumped to or loaded from files on a PC (or other kind of host) by means of the 'Trace/Dump to PC/Host...' and 'Trace/Load from PC/Host...' commands in the trace display. These commands use the XMODEM protocol to transfer trace data to a binary file on the host. To do this, the analyzer must be run from a terminal emulator on the PC (or other host).

Note: *Do not Dump or Load Trace while in Transparent mode.*



Dump to PC/Host Command

The following dialogue box will show up when the 'Dump to PC/Host...' command is selected:



Comments

The `'Comments'` are two lines where general comments about the trace can be written. This comments are displayed after the file ID line "VMETRO TRACE MODEL : <target> when you type the dumped trace file. The field will already be filled with the comments from the last loaded trace file, or with the last typed comments if no file has been loaded since the previous dump.

From line

The trace buffer dump will start with this line. If a line less than the first line in the trace buffer is specified the first trace buffer line will be used. The default value is the first trace buffer line after a trace or a `'Load From PC/Host...'`. After a dump the value used then is default. The value is a hexadecimal number for a STIM target and a decimal number for all other targets.

To line

The trace buffer dump will end after this line is dumped. If a line greater than the last line in the trace buffer is specified the first last buffer line will be used. If this line is less than the `'From Line'` the whole buffer will be dumped. The default value is the last trace buffer line after a trace or a `'Load From PC/Host...'`. After a dump the value used then is default. The value is a hexadecimal number for a STIM target and a decimal number for all other targets.

Packing

To reduce the transfer time, data compression (packing) can be selected. The buttons `'Use no packing'` and `'Use run length packing'` select between no packing or run length packing with trace buffer lines as basis. (See the description of the trace buffer file format later in this section.) The buttons will only show up when dumping a timing trace. State traces are always dumped with no packing because all trace lines then are different. The default value for timing traces is `'Use run length packing'`, but if very few lines are equal, or the SYSCLK signal for VME traces must be saved, `'Use no packing'` should be selected.

When you select <OK> the message box below will be displayed. See later in this section for how to operate the different terminal emulators.

```
Ready for transfer
(Set PC/Host to RECEIVE data)
```

Load From PC/Host Command

The following dialogue box will show up when the 'Load From PC/Host...' command is selected:

```
                         Load from PC/Host

 Load the trace buffer from a trace file on a host.

 o To proceed you must be using a terminal emulator on a PC, workstation
   or other host supporting the XMODEM file transfer protocol.
   Example: PC with VMETRO's VT100 emulator, the WINDOWS terminal
   emulator, PROCOMM PLUS etc.

   Before starting, make shure the current 'target' corresponds to the
   target of the trace file. If not, cancel now, change target and
   reactivate this command.

Insert from line : 0000 (hex)    [X] Initialize non-loaded trace lines

                        < Ok >          <Cancel>
```

The fields described below will only show up when a STIM target is selected.

Insert from line

The trace buffer load will be inserted from this line. If a line greater or equal than the last line in the trace buffer is specified, the last trace buffer line is the only line which will be loaded. The default value is 0 which is the first trace buffer line. The value is a hexadecimal number.

Initialize non-loaded trace lines

This flag controls whether the non-loaded lines should be initialized or not. Note that initialization always will be done when the trace buffer does not contain any valid data. The flag is always checked when the dialogue box is entered.

When you select <OK> the message box below will be displayed. See later in this section for how to operate the different terminal emulators.

```
    Ready for transfer
(SEND data from PC/Host)
```

Dump/Load Trace using the WINDOWS Terminal Emulator

If you run the terminal emulator found under MicroSoft Windows 3.0/3.1, use the commands '`Transfers/Send (Receive) Binary File`'.

Dump Trace using VMETRO VT100 Emulator

- Run the tracer by activating VMETRO's VT100 emulator V2.11 or newer.
- When you have a trace you want to dump to the PC, activate the '`Dump to PC/Host...`' command found under `Trace` in the menu bar. Fill in the relevant fields in the dialog box (ref. previous sections).
- When the text "Ready for transfer (Set PC/Host to RECEIVE data)" is shown, type `Alt-R` to set VMETRO's VT100 emulator to receive data.
- The VT100 emulator will then prompt for a filename at the bottom line on the screen: `Name of file to receive?`

  Fill in filename (and path), and hit <CR> to start the transfer. This may take some time to finish.
- If your screen needs to be refreshed after the process, hit \\ (two back slashes) for a screen refresh.

Load Trace using VMETRO VT100 Emulator

- Run the tracer using VMETRO's VT100 emulator V2.11 or newer.
- When the file saved with the `Dump to PC` command is to be loaded back to trace memory the same analyzer and target bus must be selected.
- Do `Trace/Show` to enter the trace display if you are not there already.

- Activate the '`Trace/Load from PC/Host...`' command found under `Trace` in the menu bar. Fill in the relevant fields in the dialog box (ref. previous sections).
- When the text "Ready for transfer (Send data from PC/Host)" is shown, type `Alt-S` to set VMETRO's VT100 emulator to send data.
- The VT100 emulator will then prompt for a filename at the bottom line on the screen: `Name of file to send?`

  Fill in filename (and path), and hit <CR> to start the transfer. This may take some time to finish.
- If your screen needs to be refreshed after the process, hit \\ (two back slashes) for a screen refresh.

Dump/Load with PROCOMM-PLUS

If you do not have the VMETRO VT100 Emulator V2.02 or newer version, you may use other commercial terminal emulators like the Procomm-Plus to move trace buffer data between a PC and the VBT-325, as explained below.

- Run the tracer from Procomm-Plus (PC+)
- Set-up PC+:

  `Alt-S`

  `PROTOCOL OPTIONS`↵

  `GENERAL OPTIONS`↵

  `C` *Abort xfer if CD lost*

  Press `<SPACE BAR>` *to toggle to NO*↵

  `ESC` *(to leave menu)*

  `ESC` *(to leave menu)*

  `ESC` *(to leave menu)*

  `Y` *MAKE CHANGES PERMANENT? (Y/N) Yes*

- When you have a trace you want to dump to PC, activate the '`Dump to PC/Host...`' command found under `Trace` in the menu bar.
- Press the `Page Down` key to start sending to PC. PC+ will prompt you for protocol. Type `X` for XMODEM protocol. PC+ will then prompt you for file name to save the trace in.

This will take some time to finish. Make a note of target bus selected.

- When the file saved with the 'Dump to PC/Host...' command is to be loaded back to trace memory the same target bus must be selected before the 'Load from PC/Host...' command is activated.
- Press the Page Up key to start receiving from PC. PC+ will prompt you for protocol. Type X for XMODEM protocol. PC+ will then prompt you for file name of file to load.

This will take some time to finish.

Trace File Format

This section describes the file format used by the Dump/Load commands. The file is built up of a set of records starting with a record ID and a record length. This makes it possible for a version of the product to read both older and newer versions of the file just by skipping the unknown records. New features will therefore be added as new records when the file format is changed.

Note: *All numbers in the file format use Motorola layout (big endian).*

File ID The file ID header contains the following text fields:

VMETRO TRACE	MODEL <nlTag>	Comments	Ctrl Z

The "VMETRO TRACE" identifies the file type. The <nlTag> is the same as the first parameter in the Main Header (see below). This copy of the string makes it easy to recognize the type of trace when typing the file. The "Comments" are private user comments that may be added when the file is created. The Ctrl Z is added at the end of the strings to make it possible to type the file and just get the header text strings displayed. (Use the DOS command `TYPE <file name>`)

File ID The File ID string is followed by records with the following layout:

ID	W	Data with length W bytes

The ID is always a byte that describes the contents of the data field. The W (Width) parameter is always four bytes (long word). It is the width (or length) of the data field in bytes. This makes it possible to skip unknown records.

Record IDs

The following ID values are defined in the current format:

ID number	Width	Description of the data field
0	62	Main Header
200	1)	Unpacked Trace buffer data
201	2)	Run length packed Trace buffer data

1) The "Unpacked Trace buffer data" record will always have the width (nhLastValTrcLine - nhFirstValTrcLine + 1) * nhTrcWidth.

2) The width of the "Run length packed Trace buffer data" will always be 0xFFFFFFFF which means the rest of the file is read as data for record 201. The record width is not calculated because the software need to read the whole trace buffer and find out how much it can be packed to calculate it.

Run length packed trace buffer data is packed on the basis of trace buffer lines as follows (the size of the "Runs" parameter is 2 bytes):

Runs	Data
1-65536	Trace line data to be repeated "Runs" times.

Main Header The Main Header has the following data fields:

Main Header		
Name of field	Size of field in bytes	Description
char nlTag[10]	10	Target ID
FLAGS nlLastRunFlags	4	Trace control flags with bits NL_HWDATA=0x00B00000 Always set NF_TIMETAG=0x00008000 Set if trace started in timing NF_TIMTAG=0x00010000 Set if time tag is in use NF_TAG16=0x00020000 Set if a 16 bits tag counter is used
BYTE nhLastRunTimingIdx	1	Speed used for last run 0=50 MHz, 1=25MHz, 2=6.25 MHz, 3=1.46 MHz, 4=781.3 KHz, 5=390.6KHz, 6=195.3 KHz, 7=97.7KHz
BYTE nhLastRunTrigPos	1	Trigger position used for run
BYTE nhTrcWidth	1	Width of sample in bytes
INT32 nhDelay	4	Trig delay (given by trig position)
INT32 nhFirstTrig	4	Trig address in trace memory (abs)
INT32 nhFirstValTrcLine	4	First valid line (log) in trace buffer
INT32 nhLastValTrcLine	4	Last valid line (log) in trace buffer
BOOLEAN nhTrgFound	2	Indicates trigger found
BOOLEAN nhTrcCompleted	2	Indicates trace completed
char nhTrigLineTxt[10]	10	Trigger line text
char nhTime[8]	8	Time when trace trigged, or was halted. The bytes are coded as follows: 0=RTC_64HZ = 64Hz counter 1=RTC_SEC = Seconds BCD coded [0..59] 2=RTC_MIN = Minute BCD coded [0..59] 3=RTC_HR = Hour BCD coded [0..23] 4=RTC_DOW = Day-of-week [0..6] == [Sunday .. Saturday] 5=RTC_DAY = Day-of-month BCD coded [1..31] 6=RTC_MNTH = Month BCD coded [1..12] 7=RTC_YEAR = Year BCD coded [0..99]
INT16 nhCalcADCVal[4]	8	Tuned ADC values when traced trigged or was halted. The 4 values are coded as follows: 0=ADC_5V = 5V value * 100 1=ADC_12V = 12V value * 100 2=ADC_N12V = -12V value * 100 3=ADC_TEMP = Temperature in degrees Celsius

16 SIMULATOR FOR PC

The diskette enclosed with the VBT-325 contains a simulator for DOS which truly represents the user-interface of the VBT-325 product with all options, including demonstration trace files for the VMETRO products. The Simulator serves three purposes:

- **Demonstration / training**
- **Reviewing trace files**
- **Creating patterns**

To use the Simulator you must install it to a hard disk. This is necessary because the program is so large that it requires use of overlays and files to support full depth of the trace buffers. Put the diskette in drive A:, create a directory on your hard disk (C:) where you want the simulator installed, change current directory to this directory and the drive to A: and type:

```
INSTALL C:<CR>
```

Note that the Simulator requires about **2.5 Mbyte** of hard disk space and about **523 Kbyte** (appr. 536000 Bytes) memory available for programs under DOS.

The installation program may start the simulator. Later you can start the simulator by changing the current directory to the simulator directory and type:

```
VMETRO<CR>
```

Demonstration / training

The simulator can be used for demonstration or training of product capabilities without having the actual product at hand. (Refer to ch. "EXAMPLE OF USE" for a Quick Guide.)

Reviewing trace files

The simulator can be used to review trace files captured by the analyzer and dumped to a file on the PC. (Like the demonstration trace files included on the diskette, loaded by: `Trace/Show`, `Trace/Load from PC <file>`. File names are listed on next page.)

Creating patterns

The simulator can be used to create patterns for the STIM200-PB pattern generator locally on a PC, and when a good pattern file is made, it can be loaded into the STIM200-PB itself for execution.

Simulate bus activity

The simulator has three keys used to simulate hardware conditions:

Ctrl-T Simulate **TRIGGER FOUND**. Reads the demonstration trace files described below if they exists or creates random trace data.

Ctrl-V Simulate **VIOLATIONS FOUND** on the bus when the VBAT-PB is active.

Ctrl-A Simulate **STATISTICS UPDATE** when the bus statistics is running.

It is not necessary to go to the directory where the simulator is located, but all files used by the simulator must be in the current active directory.

Trace Files

Trace buffer data files for simulated trace full		
Target	**Sampling Mode : Timing**	**Sampling Mode : State**
VME	VMET.TRC	VMES.TRC
VSB	VSBT.TRC	VSBS.TRC
SCSI	SCSIT.TRC	SCSIS.TRC
VXI_P2ac	VXI_P2AT.TRC	VXI_P2AS.TRC
P2ac	P2ACT.TRC	P2ACS.TRC
TIM_VME	TIM_VME.TRC	N/A
TIM_VSB	TIM_VSB.TRC	N/A
TIM_SCSI	TIM_SCSI.TRC	N/A
FB+Phase	FBPHASET.TRC	FBPHASES.TRC
FB+Demux	N/A	FBDEMUXS.TRC
TIM_FB+	TIM_FB.TRC	N/A

Other files used by the simulator are:

VMETRO.PAR Holds information about the previously selected product option and piggyback.

TRACEMEM.DAT

Trace buffer for the main product.

STIMMEM.DAT Trace buffer for the S/TIM piggybacks.

RAMij.SIM Setup files. One of this files exist for each combination of a main product and a piggyback option. The 'i' is the number of the Main product with range <0,2>, and the 'j' is the number of the piggyback option with the following ranges:

i=0, j=<0,7>

i=1, j=<0,7>

i=2, j=<0,3>

The order of the numbers are the same as the order of the products as they are shown in the simulator product selection screen.

For example, the setup file for VBT-325C with TIM200-PB/VME-P2 will be "RAM06.SIM".

Files used when the user simulates trace buffer full with CTRL-T or by selecting the "Simulate Trace Full" from the Trace menu are shown in the table above. The simulator will generate a random trace if the files for the selected target does not exist.

INDEX

A

accelerator keys, 38
accumulate, 123
accumulate mode, 71
ANSI.SYS, 151
applications, 4
ASCII terminal, 1
asynchronous sampling, 25
attributes
  address, 47
  data, 47

B

bar markers, 68
black cockpit, 123
block cycle, 30
brackets, use of, 51
bus grant, 12
Bus Tracer Chip, 2
bus transfer rate, 76

C

cable
  coax, 7
  external power, 7
  PC, 7, 18
  RS232, 7
  terminal, 7, 18
  transparent mode, 7
cables, 7
carrier, piggiback, 6
COM1 port, 151
COM2 port, 151
conductive wrist-chain, 9
control signals, 43
copper tracks, cutting, 11
count, 55, 96
counters
  statistics, 28
  VMEbus, 28
cross trigger, 34
cross-triggering, 62
cutting copper tracks, 149

D

debugging, 4
defining ranges, 48
delay, 55
dialog boxes, 39
DTACK, 30
dump tracedata, 155

E

ECO level, 133
edit command, 85
edit event patterns, 45
elapsed time, 25
event names, 53
event patterns, 42
  edit, 45, 78
  insert, 46
extender board, 14
external inputs, 34
external power, 15
external signal, 34

F

fields
  delete, 45
  insert, 45
file format, 160
format command, 96
format, trace file, 160
function command, 98
function keys, 40
futurebus+, 2

G

generator
pattern, 5
stimuli, 5
goto, 55
grant, 12

H

halt operator, 56
handshake, 19
hardware handshake, 19
help command, 97, 102
histograms
counter driven, 67
trace driven, 67
horizontal scrolling, 46

I

IACK, 30
If / Elsif / Else, 54
indents, 51
IRQ, 30

J

jump command, 95
jumper settings, 10, 137
jumpers, spare, 10

K

keyboard template, 41

L

latching
bus grant, 29
bus request, 29
line number, 51
load tracedata, 155
lock on first, 123
loose sequence, 57

M

main menu, 83
mask violations, 112
master/slave, 5
max scale, 68
memory, non-volatile, 22
menu selection, 38
menu, start-up, 21
message line, 42
mode
sequencer, 50
single event, 50
mode, transparent, 19

N

non-volatile memory, 22
NOT operator, 43
numeric keypad, 41

O

operator
NOT, 43
sampling, 53
store, 54
operators, 53
options command, 99
overview, 1

P

P2 bus, 13
pattern generator, 5
PCB level, 133
performance analysis, 4
piggyback, 5
piggyback carrier, 6
piggybacks, 82
ports
host/printer, 17
ports, serial, 17
power
external, 15
jumpers, 15

power supply, 14
print, 17, 93
pull down menus, 37

Q

quit command, 97, 102

R

ranges, defining, 48
refresh screen, 42
RMW cycle, 30
RS232 cable, 7
RS232 port, 1

S

sampling
  asynchronous, 24
  data cycle, 30
  methods, 29
  registers, 24
  SCSI state, 32
  state, 24
  synchronous, 24
  VSB state, 31
  VXI, 33
sampling operator, 53
sampling rate, 25
script commands, 153
script files, 39
script language, 152
scrolling, horizontal, 46
SCSI connection, 145
SCSI pinout, 33
search, 81
search command, 94
select event, 68
select signals, 64
select slot, 12
sequencer, 27, 48
  actions, 49
  transitions, 49
sequencer mode, 50
sequencer operators, 53
sequencer program, 49
sequencer program, modify, 53
serial ports, 17
session command, 98
setup command, 88
setup screen, 37, 83
signal inputs, 34
signal polarity, 44
signal selection, 64
simulator, 10, 66, 163
single event mode, 50, 79
slot selection, 12
spare jumpers, 10
specifications, 4
SSBLT cycle, 31
start-up menu, 21
state analysis, 25
state number, 51
static electricity, 9
statistics command, 88
statistics counters, 28
statistics screen, 37, 68
stimuli generator, 5
store operator, 54
synchronous sampling, 25
system controller, 5

T

target command, 87, 102
temperature probe, 34
temperature sensor, 7
template, keyboard, 41
terminal type, selecting, 21
tight sequence, 57
time tag, 25, 34
timing analysis, 25
timing analyzer, 5
trace buffer, 27
trace command, 83, 93
trace display screen, 37
trace dump, 66
trace format, 160
trace list, 63
trace print, 17, 93
transparent mode, 19

trigger operator, 56
trigger position, 27
tuning, 4
twin mode, 24

U

upgrade, 132
utilities command, 102

V

VBAT, 111
VBAT commands, 121
VBAT limitations, 117
VBAT mask command, 122
VBAT modes command, 122
VBAT operation, 120
VBAT specifications, 115
VBAT violations command, 121
VIC068 chip, 5
VME backplane, 12
VME64, 26, 31, 47, 48, 105, 114
VMEbus, 2
  traffic, 5
voltage
  12V, 15
  5V, 14
voltage drop, 14
VSB backplanes, 12
VSB/ SCSI/P2, 2
VT100.EXE, 17, 151
VXIbus, 2

W

waveforms, 64
wide mode, 24
window command, 97
word recognition, 25
workstation, 1
wrist-chain, conductive, 9