User's Manual
VBT-325
Including XMEM325-PB

Rev. 1.5B - Valid for Firmware Version 5.0X, 5.1X

VBT-325B  VMEbus Analyzer
VBT-325C  VMEbus & VSB/SCSI/P2 Analyzer
XMEM325-PB Extended Trace Memory for the VBT-325

Related products described in separate documents:
VBAT-PB  VMEbus Anomaly Trigger Piggyback module
VBAT64-PB  VME/VME64 Anomaly Trigger Piggyback module
TIMBAT-PB  200 MHz Timing Analyzer & Anomaly Trigger for VMEbus Piggyback module
S/TIM200-PB  200MHz Timing Analyzer and Bus Master / Pattern Generator Piggyback module


No part of this document can be copied or reproduced, in any form, on any media, without written permission from VMETRO Inc. (Houston, TX, USA) or VMETRO A/S (Oslo, Norway).
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.
1180 Diary Ashford, #535.
Houston, TX 77077, USA
Tel.: (713) 584-0728
Fax: (713) 584-9034

Europe, Asia: VMETRO A/S
Nedre Rommen 5E
N-0988 OSLO, Norway
Tel.: +47 2210 6090
Fax: +47 2210 6202
Preface

This Manual

This manual covers the VBT-325 and the XMEM325-PB piggyback module, and corresponds to firmware versions 5.0X as well as 5.1X.

The XMEM325-PB is described separately only under the “XMEM325-PB Piggyback Module”, “Product overview” and “Installation”. All other instructions governing the operation of the piggyback module are identical to those for the VBT-325, and is found in the general text.
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 logical, 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.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>TABLE OF CONTENTS</td>
<td>v</td>
</tr>
<tr>
<td>CONVENTIONS</td>
<td>1</td>
</tr>
<tr>
<td>INTRODUCTION</td>
<td>2</td>
</tr>
<tr>
<td><strong>1 VBT-325 PRODUCT OVERVIEW</strong></td>
<td>5</td>
</tr>
<tr>
<td>Main blocks</td>
<td>6</td>
</tr>
<tr>
<td>Model B and C</td>
<td>7</td>
</tr>
<tr>
<td>Applications</td>
<td>8</td>
</tr>
<tr>
<td>Specification Highlights</td>
<td>8</td>
</tr>
<tr>
<td>Piggyback modules</td>
<td>9</td>
</tr>
<tr>
<td>Piggyback Carrier (VPC-MkII)</td>
<td>10</td>
</tr>
<tr>
<td>Piggyback User's Manuals</td>
<td>11</td>
</tr>
<tr>
<td>VBAT-PB / VBAT64-PB / TIMBAT-PB</td>
<td>11</td>
</tr>
<tr>
<td>S/TIM200-PB</td>
<td>11</td>
</tr>
<tr>
<td>VDRIVE-PB</td>
<td>11</td>
</tr>
<tr>
<td>XMEM325-PB</td>
<td>11</td>
</tr>
<tr>
<td>Accessories</td>
<td>12</td>
</tr>
<tr>
<td><strong>2 INSTALLATION</strong></td>
<td>13</td>
</tr>
<tr>
<td>Static electricity - Precautions</td>
<td>13</td>
</tr>
<tr>
<td>Preparations</td>
<td>14</td>
</tr>
<tr>
<td>Inspection</td>
<td>14</td>
</tr>
<tr>
<td>Jumper settings</td>
<td>14</td>
</tr>
<tr>
<td>P2 rows A/C connections</td>
<td>15</td>
</tr>
<tr>
<td>TTL/CMOS Input only (0-5V)</td>
<td>15</td>
</tr>
<tr>
<td>Isolation of P2 rows a/c</td>
<td>15</td>
</tr>
<tr>
<td>Slot Selection</td>
<td>16</td>
</tr>
<tr>
<td>VMEbus</td>
<td>16</td>
</tr>
<tr>
<td>VSB</td>
<td>16</td>
</tr>
<tr>
<td>Daisy-Chains</td>
<td>16</td>
</tr>
<tr>
<td>Slot 1</td>
<td>17</td>
</tr>
<tr>
<td>User-defined P2</td>
<td>17</td>
</tr>
<tr>
<td>Power supply</td>
<td>18</td>
</tr>
<tr>
<td>Air Cooling</td>
<td>18</td>
</tr>
<tr>
<td>5V readout</td>
<td>18</td>
</tr>
<tr>
<td>+/-12V supplies</td>
<td>19</td>
</tr>
<tr>
<td>Flash 12V Jumper</td>
<td>19</td>
</tr>
</tbody>
</table>
External Power........................................................................................................19

Terminal Connection ..........................................................................................21
  Transparent mode.............................................................................................23
  Terminal-Host RS232 handshake..................................................................23

Power-On .............................................................................................................24
  Automatic Baud Rate detection ..................................................................24

Start-up menu ..................................................................................................25
  T: Select New Terminal type .......................................................................25
  PC as terminal .........................................................................................26
  CR: Continue .........................................................................................26
  C : Clear Non-volatile memory ...............................................................26

3 FUNCTIONAL DESCRIPTION .......................................................................27

Main blocks .......................................................................................................27
  Sampling stage ............................................................................................28
    State (Synchronous) sampling ...............................................................28
    Timing (Asynchronous) sampling ...........................................................29

Word Recognition / Triggering stage ...............................................................30
  Sequencer ...................................................................................................31

Sample Storage stage .....................................................................................31
  Trace Buffer .............................................................................................31
  Trigger position .........................................................................................31

Statistics Counters ..........................................................................................32

Sampling methods ..........................................................................................33
  VMEbus state sampling ..........................................................................33
    Bus Request latching ...........................................................................33
    Bus Grant latching - BgL .................................................................33
    Data cycle sampling .............................................................................34
    IRQ -> IACK .....................................................................................34
    RMW, Block .......................................................................................34
    VME64 ..................................................................................................34
    SSBLT ..................................................................................................35

VSB state sampling ........................................................................................35

SCSI state sampling .........................................................................................36
  SCSI Pinout ...............................................................................................36
  SCSI-2 FAST ............................................................................................36

VXI sampling ....................................................................................................37

User-defined P2 state sampling ........................................................................37

External inputs ................................................................................................37
  In3:0 in VME part ....................................................................................37
  In3:0 to P2 part using cross trigger .........................................................37
  Shared In3:0 and Time Tag bits ..............................................................38
  Shared In1 and Temperature Probe .........................................................38
4 XMEM325-PB PIGGYBACK MODULE ................................................................. 39

Product Overview ................................................................................................. 39
Main Blocks ........................................................................................................... 40

Installation............................................................................................................. 41
Static Electricity - Precautions ............................................................................ 41
Mounting................................................................................................................ 41
Removal.................................................................................................................. 42
Power Supply ........................................................................................................ 43

5 OPERATION ....................................................................................................... 44

Screen elements and Commands ........................................................................ 45
Pull down menus .................................................................................................... 45
Accelerator keys .................................................................................................... 46
Dialog Boxes .......................................................................................................... 47
Function keys ......................................................................................................... 47
Numeric keypad .................................................................................................... 48
Keyboard template ................................................................................................. 49
Multiple windows ................................................................................................ 49
Status line ............................................................................................................. 49
Refresh Screen ...................................................................................................... 49

User-interface structure ....................................................................................... 50
Targets .................................................................................................................... 50
Screen categories ................................................................................................ 50
Setup screen .......................................................................................................... 51
Trace Display screen ............................................................................................ 52
Statistics screen ..................................................................................................... 53

Event Patterns ...................................................................................................... 54
Control signals .................................................................................................... 55
Signal groups with predefined values ............................................................... 55
Signal polarity ....................................................................................................... 55
NOT operator ....................................................................................................... 55
Editing event patterns .......................................................................................... 56
Field mnemonics and other options ................................................................. 57
Clearing contents of fields .................................................................................. 59
Hiding field columns ........................................................................................... 59
Adding field columns .......................................................................................... 60
Renaming, clearing, deleting and copying entire events .................................... 61
Adding events ....................................................................................................... 61
Individual signals ................................................................................................. 62
Different Signal templates in State and Timing mode ......................................... 63
BR* and IRQ* format ............................................................................................ 64
BgL editing ............................................................................................................. 64
Address/Data options .......................................................................................... 65
Edit Range ............................................................................................................. 65
Edit range without using the Field Options dialog ............................................ 65
Binary details ...................................................................................................... 66
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cross-triggering</td>
<td>88</td>
</tr>
<tr>
<td>Cross-trigger VSB from VME</td>
<td>89</td>
</tr>
<tr>
<td>Trace Display</td>
<td>90</td>
</tr>
<tr>
<td>Alphanumeric trace list</td>
<td>90</td>
</tr>
<tr>
<td>Navigation and Signal selection</td>
<td>91</td>
</tr>
<tr>
<td>Formatting options</td>
<td>91</td>
</tr>
<tr>
<td>Changing alphanumeric formatting template</td>
<td>92</td>
</tr>
<tr>
<td>Change format for individual columns</td>
<td>93</td>
</tr>
<tr>
<td>Navigating the Trace buffer</td>
<td>95</td>
</tr>
<tr>
<td>Setting bookmarks</td>
<td>95</td>
</tr>
<tr>
<td>The Jump commands</td>
<td>96</td>
</tr>
<tr>
<td>Opening additional windows</td>
<td>97</td>
</tr>
<tr>
<td>Searching for and filtering trace data</td>
<td>99</td>
</tr>
<tr>
<td>Searching</td>
<td>101</td>
</tr>
<tr>
<td>Extracting trace data</td>
<td>102</td>
</tr>
<tr>
<td>Waveforms</td>
<td>104</td>
</tr>
<tr>
<td>Hiding and Inserting signals</td>
<td>107</td>
</tr>
<tr>
<td>Edge Jumping and Timing Markers</td>
<td>108</td>
</tr>
<tr>
<td>Time per division</td>
<td>110</td>
</tr>
<tr>
<td>Trace Dump to PC/Host</td>
<td>110</td>
</tr>
<tr>
<td>Simulator used for trace review</td>
<td>110</td>
</tr>
<tr>
<td>Statistics</td>
<td>111</td>
</tr>
<tr>
<td>Statistics screen</td>
<td>112</td>
</tr>
<tr>
<td>Display features</td>
<td>112</td>
</tr>
<tr>
<td>Time History Curves</td>
<td>113</td>
</tr>
<tr>
<td>Counters Operation</td>
<td>114</td>
</tr>
<tr>
<td>Update rate</td>
<td>115</td>
</tr>
<tr>
<td>Accumulate/Reset mode</td>
<td>115</td>
</tr>
<tr>
<td>Idle interval</td>
<td>116</td>
</tr>
<tr>
<td>Event Counting / Bus Level Histogram</td>
<td>117</td>
</tr>
<tr>
<td>Bus Utilization Histogram</td>
<td>118</td>
</tr>
<tr>
<td>VMEbus Utilization measurements</td>
<td>118</td>
</tr>
<tr>
<td>VSB Utilization measurements</td>
<td>119</td>
</tr>
<tr>
<td>Bus Transfer Rate Histogram</td>
<td>120</td>
</tr>
<tr>
<td>6 COMMANDS REFERENCE</td>
<td>121</td>
</tr>
<tr>
<td>Main menu - Setup screen</td>
<td>121</td>
</tr>
<tr>
<td>Trace</td>
<td>122</td>
</tr>
<tr>
<td>- Run VME</td>
<td>122</td>
</tr>
<tr>
<td>- Run Multiple</td>
<td>122</td>
</tr>
<tr>
<td>- Sampling Status</td>
<td>123</td>
</tr>
<tr>
<td>- Halt</td>
<td>124</td>
</tr>
<tr>
<td>- Halt All</td>
<td>124</td>
</tr>
<tr>
<td>- Show</td>
<td>124</td>
</tr>
</tbody>
</table>
Edit ................................................................. 124
  - Event Patterns .................................................. 124
  - Sequencer ...................................................... 125
  - Trigger position ............................................. 125
  - Sampling mode ............................................... 125
  - Options / State Sampling options ....................... 125

Target ............................................................... 126
  - VME ............................................................. 126
  - P2 bus .......................................................... 126
  - XMEM VME .................................................... 127
  - XMEM P2 ....................................................... 127
  - TIM200 bus .................................................... 127
  - Reconfigure .................................................. 127

Statistics .......................................................... 127

Setups ............................................................... 128
  - Initialize ..................................................... 128
  - Store .......................................................... 128
  - Delete ......................................................... 128
  - 1 Default .................................................... 128
  - Load from PC/Host ........................................... 129
  - Dump to PC/Host ............................................ 130

Utilities ............................................................ 132
  - Transparent Mode .......................................... 132
  - Transparent Mode Options .............................. 132
  - Trigger Output Options .................................. 132
  - LED Display ................................................ 133
  - Serial Ports ................................................ 134
  - Assert VMEbus SYSRESET* ............................ 134
  - Selftest ..................................................... 134
  - Reset Analyzer .......................................... 134
  - Specials ..................................................... 135

Help ................................................................. 135

Trace Display menu ............................................ 136

Trace ............................................................... 136
  - Dump to PC .................................................. 137
  - Load from PC .............................................. 138
  - Print .......................................................... 138

Search ............................................................. 139

Jump ............................................................... 139
  - First Line .................................................... 139
  - Last Line .................................................... 139
  - Trigger Line ............................................... 140
  - Marker Y ..................................................... 140
  - Marker Z ..................................................... 140
  - Line number ............................................... 140
  - Edge Options .............................................. 140
Count ............................................................................................................. 140
Format .............................................................................................................. 141
   - Time/Div .................................................................................................. 141
   - Absolute/Relative Time tags .................................................................. 141
   - Decoding and Formatting ...................................................................... 141
   - Formatting Template ............................................................................ 142
Markers ........................................................................................................... 142
Window ............................................................................................................ 142
   - New ........................................................................................................ 142
   - Close ....................................................................................................... 142
Quit .................................................................................................................. 142
Help ................................................................................................................. 142
Statistics menu ............................................................................................... 143
Session .............................................................................................................. 143
   - Run ........................................................................................................ 143
   - Continue ............................................................................................... 143
   - Halt ....................................................................................................... 144
   - Immediate Start .................................................................................. 144
   - Start On Trigger .................................................................................. 144
Function ........................................................................................................... 144
   - Event Counting .................................................................................... 144
   - Bus Utilization (VME and VSB) ............................................................ 144
   - Bus Transfer Rate (VMEbus only) ......................................................... 145
Options ............................................................................................................ 145
   - Standard Histograms vs. Time History Curves ..................................... 145
   - Bar Markers ........................................................................................ 145
   - History Curve Options ...................................................................... 146
   - Maximum Scale ................................................................................. 147
   - Count Options ................................................................................... 147
   - Select Events ..................................................................................... 148
   - Sampling Mode ................................................................................. 149
Target ............................................................................................................... 149
   - VME ................................................................................................. 149
   - P2 bus ............................................................................................... 149
Utilities .......................................................................................................... 149
Quit ................................................................................................................. 149
Help ................................................................................................................. 149

7 SIGNAL REFERENCE ................................................................................. 150

   VMEbus Signal Fields ............................................................................. 150
   Default Signal Fields in State Mode ...................................................... 150
       Cross-trigger signals .......................................................................... 155
   Default Signal Fields in Timing Mode .................................................... 156
   Signal fields not included by default ...................................................... 157
       Signal fields available in Events and Trace ........................................ 157
       Signals available in Trace only .......................................................... 158
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>VSB Signal Groups</td>
<td>159</td>
</tr>
<tr>
<td>Default Signal Fields in State Mode</td>
<td>159</td>
</tr>
<tr>
<td>Default Signal Fields in Timing Mode</td>
<td>161</td>
</tr>
<tr>
<td>Signal fields not included by default</td>
<td>162</td>
</tr>
<tr>
<td>Signal fields available in Events and Trace</td>
<td>162</td>
</tr>
<tr>
<td>Signals available in Trace only</td>
<td>162</td>
</tr>
<tr>
<td>Default Signal Fields in State Mode</td>
<td>163</td>
</tr>
<tr>
<td>Default Signal Fields in Timing Mode</td>
<td>164</td>
</tr>
<tr>
<td>Signal fields not included by default</td>
<td>164</td>
</tr>
<tr>
<td>Default Signal Fields in State and Timing Modes</td>
<td>165</td>
</tr>
<tr>
<td>SCSI Signal Groups</td>
<td>163</td>
</tr>
<tr>
<td>Default Signal Fields in State Mode</td>
<td>163</td>
</tr>
<tr>
<td>Default Signal Fields in Timing Mode</td>
<td>164</td>
</tr>
<tr>
<td>Signal fields not included by default</td>
<td>164</td>
</tr>
<tr>
<td>VXI Signal Groups</td>
<td>165</td>
</tr>
<tr>
<td>Default Signal Fields in State and Timing Modes</td>
<td>165</td>
</tr>
<tr>
<td>8 TRACE EXAMPLES</td>
<td>166</td>
</tr>
<tr>
<td>VMEbus</td>
<td>166</td>
</tr>
<tr>
<td>VSB</td>
<td>167</td>
</tr>
<tr>
<td>9 VBT-325 FIRMWARE UPGRADE PROCEDURE</td>
<td>168</td>
</tr>
<tr>
<td>Diskette for PCs</td>
<td>168</td>
</tr>
<tr>
<td>Boot PROM</td>
<td>168</td>
</tr>
<tr>
<td>Firmware version 5.10</td>
<td>168</td>
</tr>
<tr>
<td>Firmware upgrade preparations</td>
<td>169</td>
</tr>
<tr>
<td>RS232 connection</td>
<td>169</td>
</tr>
<tr>
<td>+12V required</td>
<td>169</td>
</tr>
<tr>
<td>Load firmware from PC</td>
<td>169</td>
</tr>
<tr>
<td>If things goes wrong</td>
<td>173</td>
</tr>
<tr>
<td>Communication errors</td>
<td>173</td>
</tr>
<tr>
<td>Flash memory errors</td>
<td>173</td>
</tr>
<tr>
<td>Tuning Parameters Lost</td>
<td>174</td>
</tr>
<tr>
<td>Missing PCB and ECO level</td>
<td>175</td>
</tr>
<tr>
<td>Procedure if Boot PROM is missing or wrong version</td>
<td>175</td>
</tr>
<tr>
<td>Run from Boot PROM</td>
<td>177</td>
</tr>
<tr>
<td>10 JUMPER SETTINGS</td>
<td>178</td>
</tr>
<tr>
<td>VME &amp; VSB (Twin mode)</td>
<td>180</td>
</tr>
<tr>
<td>VSB Bus Grant Daisy chain jumper</td>
<td>180</td>
</tr>
<tr>
<td>VME &amp; SCSI (Twin mode)</td>
<td>182</td>
</tr>
<tr>
<td>VME &amp; VXI (Twin mode)</td>
<td>184</td>
</tr>
<tr>
<td>VME &amp; P2ac User defined (Twin mode)</td>
<td>186</td>
</tr>
<tr>
<td>XMEM325-VME, VSB &amp; VXI</td>
<td>188</td>
</tr>
<tr>
<td>XMEM325-VME &amp; SCSI</td>
<td>189</td>
</tr>
</tbody>
</table>
11 PINOUT P1/P2
VME & VSB Connection on P1 & P2 .............................................................. 192
SCSI Connection on P2 ................................................................................. 192
VBT-325 SCSI P2 adapter ............................................................................. 195
SCSI-cable type VSC-8 .............................................................................. 195
SCSI-cable type VSC-3 .............................................................................. 196
Customized isolation of P2a/c .................................................................. 196

12 P2AC USER DEFINED TRACER .................................................................. 198
Slot Selection ............................................................................................... 198
Jumper Settings ............................................................................................. 199
  Sampling clock select .............................................................................. 199
  P2 Cross Trigger Jumper ....................................................................... 199
Command Reference ..................................................................................... 200
  Selecting Target ...................................................................................... 200
  VME ........................................................................................................... 200
  P2 bus ........................................................................................................ 200
  XMEN_VME bus ........................................................................................ 200
  XMEN_P2 bus ........................................................................................... 201
  TIM200 bus .............................................................................................. 201
  Reconfigure ............................................................................................... 201
Defining Signal Fields .................................................................................. 202
  Edit an existing field definition ................................................................. 204
  How to delete a signal field definition ......................................................... 205
Using time tags ............................................................................................. 206
Dump/Load Setup to/from PC or Host ........................................................... 207

13 VMETRO VT100 TERMINAL EMULATOR .................................................. 208
How to use the VT100 ................................................................................... 208
  The VT100 environment variable ............................................................... 210
  Terminal types to use on the tracer ............................................................. 211
  Built-in XMODEM CRC protocol ............................................................... 211
Built-in script language .................................................................................. 211
  Script control commands ........................................................................... 211
  Function keys in script files ...................................................................... 213
  Script example #1 ...................................................................................... 214
  Script example #2 ...................................................................................... 215
Trace File Format .......................................................................................... 215
  Trace Data Line format, VBT-325 VME ..................................................... 219
  Details of the time tag bits ........................................................................ 219
  How to convert the time tag to a time value ............................................ 220
Details of the Am, Irq, Str, Bg, St2 and St3 bytes.................................224
Details on internally generated bits.................................................224
Differences for the XMEM-PB VME..................................................225
Trace Data Line format, VBT-325 and XMEM-PB VSB targets..............225
Details of St4 and St5 bytes.........................................................226
Trace Data Line format, VBT-325 and XMEM-PB SCSI targets..............226

14 SIMULATOR FOR PC.........................................................................227
Installation......................................................................................227
Self-running Demo...........................................................................228
Simulate bus activity........................................................................228
Trace Files........................................................................................229
Included Trace example files.............................................................230

INDEXES.............................................................................................232
Command Index.................................................................................232
Setup Screen.....................................................................................232
Trace Screen......................................................................................233
Statistics Screen...............................................................................233
Index..................................................................................................234
CONVENTIONS

The VBT-325 User’s Guide includes several usability aids and conventions to help you find and identify the information you need. The following section describes these usability aids and conventions.

Conventions

What it means and what it is used for:

• (Bullet) Used for step-by-step instructions and itemized lists.

^P

Press the control-key (look for CTRL on your keyboard) together with the indicated key.

Screen text

Text displayed on the terminal screen is set in Courier font.

\[<\text{Cr}>\]

The carriage return key. Also denoted <cr> in the running text.

<Key>

<Key> denotes a generic function key or character.

command ⊲

User input is also set in Courier font. Underlined courier text indicates literal user input, typically commands and responses to prompts that should be typed exactly as printed in the manual. The ⊲-sign at the end of the line marks that the input should be terminated by the carriage return key.

<italic>

Italic courier font indicates variables or parameters that you replace with an appropriate word or string.

{one | two}

Curly braces “{..}” means select one of the listed items.

Trace/Run

A menu command is written like this. The underlined characters indicates shortcuts. The command Trace/Run, can thus be executed by either (1) highlighting the Trace command, type <cr>, then highlighting the Run command and type <cr>, or (2) by typing a <T> and then a <R>.

The STOP symbol indicates a section of critical importance. Overlooking this information may cause damage to the VBT and/or other equipment. Consequential damages related to the use of the VBT is not covered by the warranty.

Indicates important, but not crucial information. Still, you should take notice if you want to use all capabilities built into your VBT-325.
INTRODUCTION

Quick-start

The following sections should assist you in the basic operations of the VBT-325. It will guide you through the basic State and Timing analysis capabilities of the product. Refer to the indicated sections in this User's Manual.

Intro

To keep this Introduction compact, and as informative as possible, references are made to other parts of the manual. If needed, go to the referred sections, read the detailed information up to the point where a symbol [as shown in the left margin] is encountered.

☐ Install

To install the product, go to page 13 for detailed instructions regarding ESD and voltage precautions. Make sure that you do not have abnormal voltages in the system, i.e. ±15V at the 12V pins, or any other abnormal voltages, as this may physically damage the VBT-325.

☐ Start-up

Power up your VMEbus system. Make sure that everything works as it did before the tracer was installed. The front panel of the VBT-325 should now alternatively display “Type CR” and “19k2 81n”

☐ Connect terminal

Connect the RS-232 port of a terminal [or a PC] to the port marked Terminal on the VBT-325. Refer to page 21 for how to make a suitable cable. Set up the terminal [or the Terminal emulator on the PC], to have the following serial parameters:

19200 baud, 8 bits word length, 1 stop bit, no parity.

Type <Cr> once or twice, until the following text is displayed on the terminal:

TERMINAL PORT : 19K2 81N
HOST PORT : 9600 81N
TERMINAL TYPE : DEC VT-100/VT-102
PIGGYBACK CONNECTED: (NONE)
START-UP OPTIONS:
1: SELECT NEW TERMINAL TYPE.
C: CLEAR NON-VOLATILE MEMORY.
SELECT AN OPTION OR TYPE <CR> TO CONTINUE:

☐ Set terminal type

Type <Cr> to continue if you are using an VT100, VT102, VT220, VT3xx or VT4xx or any other terminal [or terminal emulator software] that works as an VT-100 terminal. To select another terminal type, follow the instructions on page 25.
Type <cr> once. The following text is displayed on the terminal:

<table>
<thead>
<tr>
<th>VMETRO VBT-325C</th>
<th>VME Bus Analyzer</th>
<th>Setup: Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trace</td>
<td>Edit Target</td>
<td>Statistics</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Setups</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Utilities Help</td>
</tr>
<tr>
<td>Event</td>
<td>Patterns</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Any**thing x xx xxxxxxx x xxx x xx x xxx x xx
- **VME0** x xx xxxxxxx x xxx x xx x xxx
- **VME1** x xx xxxxxxx x xxx x xx x xxx
- **VME2** x xx xxxxxxx x xxx x xx x xxx
- **VME3** x xx xxxxxxx x xxx x xx x xxx

**VMETRO VBT-325C** | **VME Trace** | **Sampling: STATE at Start**
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Trace</td>
<td>Search</td>
<td>Jump</td>
</tr>
<tr>
<td></td>
<td>Jump</td>
<td>Count</td>
</tr>
<tr>
<td></td>
<td>Format</td>
<td>Markers</td>
</tr>
<tr>
<td></td>
<td>Window Quit</td>
<td>Help</td>
</tr>
</tbody>
</table>

- **Sampling Modes**
  - The VBT-325 is capable of sampling in **State mode**, and in an up to 50Mhz **Timing mode**. See page 28 for more information on sampling methods.
  - **State sampling** is the default sampling mode. To get a snap-shot of the current VMEbus activity, execute the command `Trace/Run`:
  - **Trace/Run** Type `<T>` and `<R>` to execute the command `Trace/Run`. After a few moments, a similar screen should appear:
Introduction

The contents of the Trace screen [as shown in the figure at the previous page] is, of course, dependent on the current bus traffic.

Quit

Type <Q> to quit the Trace screen and go back to the Setup screen.

Timing mode

Change the sampling mode to Timing by executing the command

Edit/Sampling Mode/50Mhz Timing

Trace/Run

Execute the command Trace/Run once more. After a few moments, a screen, similar to this, will appear:

![Trace Screen Screenshot]

The values of the signal in the trace window displayed on your terminal will differ from the above figure.

Specifications

Continue reading on the next page for VBT-325 product specifications.

Usage

Continue reading in chapter 4, at page 44, for more details on how to use the product.
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 a 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 512 Kbytes [or 1 Mbytes] of Flash EPROM, and 512 Kbytes static RAM, controls the hardware and runs the user-interface, which is operated from an ASCII terminal, PC or workstation through a RS-232 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

As can be seen from the block diagram above, 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 is field-upgradable to a VBT-325C by means of a firmware/PLD upgrade.
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 or 128K 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 Timing Analyzer and Bus Master / 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 cycles.

**XMEM325-PB**

See Chapter 4, starting at page 39.

**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 contains rule-based parallel trigger elements that 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.

**VBAT64-PB**

An enhanced VBAT-PB. Includes rules to detect violations of the VME64 specification.
**TIMBAT-PB**  The TIMBAT-PB combines the VBAT64-PB and the TIM200-PB VMEbus functionality into one piggyback module.

**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 connectors. A trigger in/output connector is also present, allowing cross-triggering to/from the VBT-325 or another instrument like an oscilloscope, logic analyzer 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, as well as P2 timing analysis (VSB, SCSI or UserP2).

![Diagram](image)

**Note**  For applications that do not need 200 MHz analysis of P2- buses like VSB, SCSI, or User Defined P2, the TIMBAT-PB can be used to achieve *State-Timing-Anomaly* for VMEbus in only one slot.
Piggyback User's Manuals

**VBAT-PB / VBAT64-PB / TIMBAT-PB**

The VBAT, VBAT64 and TIMBAT piggyback modules are described in a separate User's Manual volume.

**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) are described in a separate User's Manual volume.

**VDRIVE-PB**

The VDRIVE-PB is described in a separate User's Manual volume.

**XMEM325-PB**

The XMEM325-PB Expansion Trace Memory piggyback module offers an additional tracer with the same basic capabilities as the VBT-325 [model B or C]. The few differences that exist regarding the use and capabilities of the VBT-325 and XMEM325 are explained in the chapter XMEM325-PB, and in the appropriate sections throughout 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).

<table>
<thead>
<tr>
<th>Part Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>401-TER-232</td>
<td>Terminal Cable (RS-232 DB9M-DB25F), 3m/9ft</td>
</tr>
<tr>
<td>401-TM-232</td>
<td>Transparent-Mode Cable (RS-232 DB9M-DB9M), 0.5m/1.5ft</td>
</tr>
<tr>
<td>401-PC-232</td>
<td>PC Cable (RS-232 DB9M-DB9F X), 3m/9ft</td>
</tr>
<tr>
<td>401-325-EPC</td>
<td>External Power Cable, 1m/3ft</td>
</tr>
<tr>
<td>401-325-IOBN5</td>
<td>5-way Trigger In/Out BNC Coax Cable Assembly, 1m/3ft</td>
</tr>
<tr>
<td>401-325-ETS-1</td>
<td>Temperature Sensor, 1m/3ft</td>
</tr>
<tr>
<td>401-325-ALL</td>
<td>Cable Package, includes all of the above.</td>
</tr>
<tr>
<td>401-325-IOBN1</td>
<td>Single Ext. Input/Trigger output BNC Coax Cable. 1</td>
</tr>
<tr>
<td>401-325-STA</td>
<td>VBT-325 to VPC-MkII Trigger Cable. 2</td>
</tr>
<tr>
<td>401-SCSI-VSC8</td>
<td>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. See page 195 for an illustration.</td>
</tr>
<tr>
<td>401-SCSI-VCS3</td>
<td>SCSI P2 Cable with two 50-pin connectors (male and female), and one P2 64-pin DIN female in between. Use to extend existing disk cable. See page 196 for an illustration.</td>
</tr>
</tbody>
</table>

1 One included with each VBT-325.
2 One Included with each 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

Introduction 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 the enclosed conductive wrist-strap, or similar, Static electricity: 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.)
2 Installation

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).
- A small plastic bag containing an anti-ESD wrist wrap.
- One BNC-to-MiniCoax cable 3ft/1m (for Ext. Input/Trigger Output).
- Diskette with VBT-325 Simulator (for trace file review etc, refer to the chapter SIMULATOR at 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).

<table>
<thead>
<tr>
<th>Model</th>
<th>Target bus</th>
<th>Mode (Wide/Twin)</th>
<th>Factory setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>VBT-325B</td>
<td>VME</td>
<td>-</td>
<td>X (Only option)</td>
</tr>
<tr>
<td>VBT-325C</td>
<td>VME &amp; VSB</td>
<td>Twin</td>
<td>X</td>
</tr>
<tr>
<td>VBT-325C</td>
<td>VME &amp; SCSI</td>
<td>Twin</td>
<td></td>
</tr>
<tr>
<td>VBT-325C</td>
<td>VME &amp; P2</td>
<td>Twin</td>
<td></td>
</tr>
<tr>
<td>VBT-325C</td>
<td>Futurebus+</td>
<td>Wide</td>
<td></td>
</tr>
</tbody>
</table>

1 From PCB revision “C”, the VBT-325 has a small DIL PLD device located near to the RS-232 connector of the Terminal port. Look for the label VBT-325B or VBT-325C.
VBT-325C  The factory setting for VBT-325C is twin mode, VMEbus & VSB (except VSB BGIN/OUT* daisy-chain bypass, ref. JUMPER SETTINGS.)

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.

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. Refer to a separate data sheet for detailed information.)

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 to 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. Refer to page 196 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. Refer to VDRIVE-PB User's Manual.

*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 neighbor 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.)
2 Installation

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 EPROM only) |

VPC

The VPC-MkII consumes about 1.1A @ 5.0V. The consumption at 12V is as for the VBT-325.

Note

Make sure that the board is always powered both from P1 and P2 connectors.

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 app. 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 supplies

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.

If the voltage on the +/− 12V pins in the backplane is higher than 12V, severe damage may occur!

Flash 12V Jumper

The +12V supply from the backplane is normally connected to the Flash EPROM devices via jumper J67. This jumper is, by default, installed as shown in the illustration to the left. The +12V supply is disconnected from the Flash EPROMs by moving jumper J67 to the opposite position.

Move jumper J67 to the upper position if the stability of the +12V supply is in doubt.

Moving J67 will not prevent damage to the voltage/temperature monitoring circuitry and decoupling capacitors if the voltage exceeds +15V.

Intro

If you are reading this section as referred to by the “Introduction” on page 2, return to page 2 now.

External Power

By repositioning two power jumpers (see next page) 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.

---

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

**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).

*Never move just one jumper! Both jumpers must always be placed in either Z1 & Z2 or in Z3 & Z4!*

---

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

Introduction The VBT-325 can be operated from an VT100, VT102, VT220, VT320 or VT420 compatible ASCII terminal, an IBM-compatible PC or a workstation. If run from a PC or workstation, a terminal emulator must be used. VMETRO supplies a VT100 terminal emulator for PCs that are tailored to the user-interface of the VBT-325. This terminal emulator, vt100.exe, is found on the firmware distribution diskette.

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 above. 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.

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).
2 Installation

**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. Note that the signals on pin 2 and 3 are crossed.

---

**Intro** If you are reading this section as referred to by the “Introduction” on page 2, return to page 2 now.
Baud rates etc.  The VBT-325 is able to run with baud rates from 600 to 115k\(^1\) 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.

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 DSR/DTR hardware handshake, 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.)

---

\(^1\) Firmware version dependent. Version 5.01 supports only up to 19K2 baud.
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 or Workstation running a terminal emulator)

Power can now be applied, and after power-up, the front panel display on the VBT-325 will alternate between these two texts:

19k2 81n Type CR

Automatic Baud Rate detection

When the LED display is flashing, the VBT-325 is in the auto baud rate mode, waiting for CR (i.e., \(\downarrow\)) from the terminal to determine the actual baud rate. Any baud rate between 19k2 and 1200 baud will be detected. When the baud rate is determined, the start-up menu will be written to the terminal, see the next page.

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

19k2 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, 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.

Display OK, but no text appears

If the start-up menu does not appear on the screen, check that the cables are connected as described in the chapter "Terminal connection". Also check that the terminal / PC / WS is set to 8 bits per character, 1 stop bit and no parity, and the baud rate is between 19k2 and 1200 baud.

1 The firmware 5.0X for BusView does not have automatic baud rate detection, but defaults to 19K2 baud.
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, the terminal type (default or previously selected type), and the type of installed piggyback module, if any.

VMETRO VBT-325 VME+ ANALYZER  FIRMWARE VERSION: 5.81

TERMINAL PORT: 19K2 8IN
HOST PORT: 9600 8IN
TERMINAL TYPE: DEC VT-100/VT-102
PIGGYBACK CONNECTED: (NONE)

START-UP OPTIONS:

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

SELECT AN OPTION OR TYPE <CR> TO CONTINUE:

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

T: Select New Terminal type

Introduction The user-interface of the VBT-325 is fully screen-oriented, taking advantage of the graphical properties of VT100 compatible terminals. This requires that the user specifies which terminal or terminal emulator he is using. By typing a 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 VGA SCREEN
3. VMETRO VT-100 EMULATOR / ANSI.SYS ON COLOR SCREEN
4. VMETRO VT-100 / ANSI.SYS VGA COLOR 50 LINES
5. TANDBERG TN 1280, 2200, 2280/9, 2200S
6. DEC VT-220/320/420 w/ANSI KEYBOARD, 7-BIT MODE, 25 LINES
7. DEC VT-420 w/ANSI KEYBOARD, 7-BIT MODE, 48 LINES

**Terminal Type:**

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.
2 Installation

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. See page 208 for more information.

CR: Continue

Typing CR (i.e. \n) 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.)

Intro If you are reading this section as referred to by “Quick Introduction” on page 2, return to page 2 now.

Setup screen Continue reading on page 44.

C : Clear Non-volatile memory

Type C to clear all contents of the Non-volatile RAM memory on the board. This command will cause all user setups to be lost. Use this command if a fatal software crash has occurred, e.g. if the operation of the user-interface does not behave correctly etc., or whenever you have mounted another piggyback.

The non-volatile memory can also be cleared by removing jumper J68. Do as follows: Remove the VBT-325 from the rack. Locate the backup-battery jumper, J68, to the left of the blue battery. Move the jumper to the other, left, position, and let it remain there for a few seconds. Then, move the jumper back into the original, right, position.

When the power is re-applied, the tracer firmware should start as normal, for then to display the message “Non volatile memory lost” on the status line. More information about jumpers is found at page 178.
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. 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

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.
3 Functional description

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).

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

Introduction

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. (Refer to section "JUMPER SETTINGS").

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.)

---

**Intro**

If you are reading this section as referred to by the “Introduction” on page 3, return to page 3 now.
3 Functional description

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 \( \text{VME64} \) 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.

Negation

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

Outside range can also be obtained, by using the not operator on an address or data range.

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.

**15 states**

Up to 15 states can be programmed in the Sequencer, and each state allows a level of "If..Elsif..Else" statements. 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%.
3 Functional description

<table>
<thead>
<tr>
<th>Start</th>
<th>Middle</th>
<th>End</th>
</tr>
</thead>
<tbody>
<tr>
<td>0%</td>
<td>25%</td>
<td>50%</td>
</tr>
<tr>
<td></td>
<td>75%</td>
<td>100%</td>
</tr>
</tbody>
</table>

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 straightforward 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. This is 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. 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. See the timing diagram on the next page. 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 the daisy-chain considerations in the Installation chapter. Information about the actual bus level is presented in the trace as the symbols "−−0" to "3---", or as "-----" 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.
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**

Sampling:BBIf a Read-Modify-Write cycle or Block cycle is transferred, this is signaled by keeping AS* asserted low between the 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 $\text{DTACK}^*$, while data will be sampled on both falling and rising edge of $\text{DS0}^*$ during write cycles, and at both falling and rising edge of $\text{DTACK}^*$ during read cycles, see figure below.

![SSBLT WRITE and SSBLT READ diagram]

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 ($\text{DS}^*$ is 'ADDR' in trace)
- Data phase ($\text{DS}^*$ is 'DTA' in trace)

The parallel arbitration phase is clocked on the rising edge of the $\text{AC}$ signal when an internally generated signal called $\text{ARB}^*$ is low (refer to figure on the next page). 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 $\text{ASACK1}^*$ or $\text{ASACK0}^*$, while the data phases are sampled at falling edge of $\text{ACK}^*$ (or $\text{ERR}^*$). Refer to the figure below.

![VSB state sampling diagram]
3 Functional description

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. 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*.

See the figure below:

![Waveform Diagram](image)

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 Connection on P2".)

SCSI-2 FAST

Both SCSI-1 and SCSI-2 with 8 and 16-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. TTLLTRG7:0* 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 must be connected to a pin socket on the board. See page 199 for details. The P2 pins will then be sampled at each rising edge, [optional falling edge] of this signal, as indicated here:

```
UserClk
P2a/c
```

See page 198 for detailed information on the User Defined P2 tracer.

External inputs

In3:0 in VME part

The VBT-325 has provision for four external signal inputsExternal on its front panel, through mini-coax connectors labeled 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.)
3  Functional description

Shared In3:0 and Time Tag bits

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. See page 133 for details. Refer to page 133 for how to display voltages and temperatures on the LED display.
4 XMEM325-PB PIGGYBACK MODULE

Product Overview

The XMEM325-PB is a piggyback module with 128K, 256K, 512K, or 1M samples extended trace memory with the same basic sampling capabilities as the VBT-325 model “B” or “C”. It duplicates all the sampling circuitry from the VBT-325, allowing it to work as an independent tracer, with full cross-trigger capabilities between the tracers. The module does also offer 8 trigger events on VME [except for the 128K model]. The XMEM325 supports the same bus architectures as the VBT-325, both in state- and timing modes. The unique depth of the trace memory makes it suitable for applications such as:

- Track events long before system crash
- Post-mortem analysis
- Data logging and documentation
- Software verification tasks

Models

The XMEM325 is available in two basic models, XMEM325-PB/B and XMEM325-PB/C. Model “C” is the full-featured version that supports both VMEbus and VSB/SCSI/P2, with upgradeability to other busses like VXI and Futurebus+. Model “B” is a lower-cost version, where only the VMEbus is supported, but otherwise with the same performance as the model “C”.

A model “B” cannot be upgraded to a model “C” due to HW limitations.

Each model is delivered with 128K samples, 256K, 512K or 1 Msamples trace memory. The trace buffers are separate for the VMEbus and the P2-bus.
Main Blocks

The XMEM325-PB1 contains all necessary HW to function as one VMEbus analyzer, model “B”, or as two independent analyzers, model “C”. Each analyzer part consists of three main stages, through which samples pass during the acquisition process:

- Sampling stage.
- Word recognition / Triggering stage.
- Sample storage / Statistics counting stage.

Four analyzers1 Together with the VMEbus and P2 bus analyzers found on the VBT-325C, the XMEM325-PB/C provides simultaneously state and timing analysis on the VMEbus and a user selectable P2-bus. Full cross-trigger capabilities between the four analyzers exist.

---

1 The XMEM325-PB 128K low cost model supports extended VMEbus trace only. Support for simultaneous state and timing analysis, and 8 events do not exist in this model.
Installation

This chapter describes how to install the XMEM325-PB on a VBT-325 or an VPC-MkII.

Static Electricity - Precautions

Before unpacking the XMEM325-PB from its shipping container, make sure that this takes place in an environment with controlled static electricity. Refer to page 13 for recommended procedures.

Mounting

Place the VBT-325 or VPC-MkII on a surface with controlled static environment. Place the VBT/VPC such that the P1/P2 connectors are facing towards you. Then align the XMEM325-PB, without applying any pressure, on the piggyback connectors on the VBT/VPC such that the BTCs [large, square devices] are closest to the P1 and P2 VMEbus connectors. The correct orientation of the XMEM325 piggyback is shown below:

Make sure that the table surface under the VBT/VPC is smooth, and while the boards are lying flat on the table, apply pressure with your hands gently around all the piggyback connectors to achieve proper seating of the module. Make sure that all the connectors are firmly seated.
Removal

A XMEM325-PB can be removed from the VBT-325 or VPC-MkII by means of a specially designed tool. The same tool can also be used to remove other piggybacks. To remove a XMEM325, always use the thick end of the tool as indicated in figure to the right.

To remove a XMEM325 with this tool, place the *thick* end under the edge of the piggyback module, and press carefully, a little bit at a time at different locations around the board, while the tool rests on either the front panel or the P1/P2 connectors, see the figure below:

Put a ruler etc. under the P1/P2 connectors to prevent bending of the VBT-325 board, as shown in the figure below. Never try to remove the XMEM325 or any other piggyback without the special removal tool!
Power Supply

The following table gives the power consumption of the XMEM325-PB in various configurations and situations:

<table>
<thead>
<tr>
<th>Mode</th>
<th>Current consumption @ 5.00V XMEM325-PB/C</th>
<th>Current consumption @ 5.00V XMEM325-PB/B</th>
<th>Active Target</th>
</tr>
</thead>
<tbody>
<tr>
<td>Idle (i.e not sampling)</td>
<td>3.15A</td>
<td>2.15A</td>
<td>-</td>
</tr>
<tr>
<td>ASYNC Sampling @ 6.125MHz</td>
<td>4.20A</td>
<td>3.20A</td>
<td>VME only</td>
</tr>
<tr>
<td>ASYNC Sampling @ 25MHz</td>
<td>4.60A</td>
<td>3.60A</td>
<td>VME only</td>
</tr>
<tr>
<td>ASYNC Sampling @ 6.125MHz</td>
<td>4.60A</td>
<td>-</td>
<td>VME and P2</td>
</tr>
<tr>
<td>ASYNC Sampling @ 25MHz</td>
<td>5.50A</td>
<td>-</td>
<td>VME and P2</td>
</tr>
</tbody>
</table>

For total current consumption, add the consumption of the carrier board. See page 18.
5 OPERATION

The VBT-325 is equipped with Flash EPROM devices which hold all the firmware necessary to operate the product. 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 or other common terminal emulators as Windows Terminal.

The user-interface of the VBT-325 employs most of the same principles as applied to the most modern computer products like Microsoft Windows, with menu bars, pull down menus, dialog boxes and multiple windows.

Menu bar
All main commands are shown in a menu bar on the top of the screen.

Pull-downs
Most menu bar commands have pull down menus attached, containing 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.

Dialog box

1 As an optional feature, VMETRO offers BusView for Windows which allows operation of the VBT-325 from a true Windows application. BusView for Windows is documented in a separate User's Manual.
Screen elements and Commands

Pull down menus

Use the left and right arrows [i.e. keys marked ← and →] to move the cursor to the desired command. Type <Cr> [i.e. ↵] to open the pull-down menu.

Type ↓ to select  Place the cursor on the wanted command and type <Cr> to select.

<Esc> aborts  Type <Esc> or a dot, ".", to close an opened pull-down menu or abort a selected command.

Grey text  Commands that cannot be executed in the current context are shown in low intensity. Trying to use one of these, a message like

"Msg: Cannot execute this command now".

will be printed on the status line.

For example, Trace/Halt cannot be executed before Trace/Run. If tried, the message "Err: Not running." will be displayed.
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, as in the example to the left, where "Relative Time Tags" are currently selected. To switch to the other type, place the cursor on "Absolute Time Tags" and type <Cr>.

### Accelerator keys

- **Trace**
  
  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).

- **Edit**
  
  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.

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

- **Statistics**
  
  Accelerator keys are also used for the individual selections in the pull down menu. This does not only allow 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. Refer to the section "VMETRO VT100 EMULATOR - VT100.EXE" for more information about making scripts.
Dialog Boxes

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 top half of the dialog box to the left, you can select one or more of the alternatives in each group.

If parenthesis are used, also called "radio buttons", you can only select one of the alternatives in each group.

↑,↓,←,→, Tab Use the arrow keys, or <Tab>, for navigation between the fields in the dialog box. The <Down> key will work fine unless you are positioned on a field with pull down options:

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.

Or, type <Tab> to select an entry and move to the next field in the dialog box. <Esc> will close the pull down menu without making a selection.

[x] When the cursor is placed at the desired option, type <Space> to toggle between selected [x] and un-selected [ ].

(x) Type <Space> to select the desired option. To select another option within this group, move the cursor to the other option and type <Space>.

<Cr> When the choices are made, a dialog box command is executed by typing <Cr>. Alternatively, type <Space> with the cursor positioned at the "<Ok >" button.

<Esc> Type <Esc> or "." [a dot] to close the dialog box command. Alternatively, type <Space> with the cursor positioned at the "<Cancel>" button. Closing the box undo any changes you may have done.

Function keys

PF1..PF4 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": ^_).
5 Operation

Note

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

<?>

A question mark brings up the Help screens. These can also be activated by the Help command in the main command bar.

<PF1>

Enter *Transparent Mode*.

<PF2> or ^E

Brings cursor to the menu, or when in the menu, back to the last edited window. <Esc> [or a dot "."] 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. Or, finds the previous edge when positioned in a waveform window.

<PF4>

Finds the next edge when positioned in a waveform window.

<F5> or ^R

Same as 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.

<F7> or ^V

Available for User Defined Targets only. Enter the signal field editor.

<Del> or ^Bs

<Ctrl>+<BackSpace> 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 (i.e. terminal type #1) the numeric keypad can be used just as on PC keyboards for cursor movements, and <Del> and <Ins>, see figure at right.

NB: Remember to turn off "NUM LOCK".
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.

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.

Status 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.

Cursor type

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, it may be obscured in places like tick-boxes and in single-character fields in the event patterns, unless it is made blinking.

Refresh Screen

Type \(\leftarrow\) twice to refresh the screen. This is useful if characters are lost, when you change terminal, etc.
User-interface structure

Targets

With the various piggyback options available to the VBT-325, the product may consist of up to four analyzers in its maximum configuration:

<table>
<thead>
<tr>
<th></th>
<th>VBT-325B</th>
<th>VBT-325C</th>
<th>VBT-325 with TIM200-PB or TIMBAT-PB</th>
<th>VBT-325 with XMEM325-PB</th>
</tr>
</thead>
<tbody>
<tr>
<td>VMEbus State/Timing Analyzer</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>P2 State/Timing Analyzer</td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>200MHz Timing Analyzer</td>
<td>√</td>
<td></td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Extended trace VMEbus Analyzer</td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Extended trace P2 Analyzer</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

1 Model “C”

Targets

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 subcommands under the Target command in the main command bar [see page 126 for details]

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

Note! The \textit{Target} command is not present on the VBT-325B unless one of the following piggyback modules are installed; S/TIM200-PB, TIMBAT-PB or XMEM325-PB.

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 \textit{Event Patterns} window and the \textit{Sequencer} window, used to define triggers, store qualifiers etc. These two windows are described in detail later in this section.
5 Operation

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.

Trace full during sampling

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").

Quit

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 on the previous page.
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.
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 units for the VME part and for the VSB/SCSI/P2 part), but the user may define a larger number of predefined patterns that can easily be taken into use.

By default, four user-editable patterns are provided. They are labeled \texttt{VME0..VME3}, when VMEbus is selected, \texttt{VSB0..VSB3} when VSB is selected and \texttt{SCSI0..SCSI3} when SCSI is the selected target bus.

\texttt{AnyThing} In addition, there is an fixed, i.e. not editable, event named \texttt{AnyThing}. This pattern will always be empty, i.e. contain an *all don't care* pattern, which makes it suitable to use as unconditional trigger.

In the figure above, one user-defined event pattern "\texttt{MyTrigger}" is included.

The default Event pattern window contains the most important signals and signal groups for the current sampling mode.

*Totally user configurable* The user may insert additional signals or signal groups, as well as additional patterns with user-defined labels. 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.

*Max 4 events* No more than *four* of the defined patterns may be used simultaneously in the Sequencer program. The XMEM325 allows up to 8 events to be used when mounted on an VBT-325 and sampling in \texttt{State} mode.
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. 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 predefined values

Combining control signals into groups allow symbolic values to be used for cycle type, transfer size, status etc., eliminating the need for the user to remember the various bit combinations of control signals. See page 58.

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' 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.

NOT operator

Grouping of control signals allow, with some restrictions, using a NOT operator on certain signal fields in the event pattern.
5 Operation

**Editing 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 event patterns 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 field columns, the sequence of the signal field columns may be altered.

**Edit/Event Patterns**

Execute the command *Edit/Event Patterns* [shortcut: <i>, then <e>] to move the cursor into the Event Patterns window.

**PF2 or F2**

will move the cursor *between* the menu bar and the *last edited window*. Initially, the "last edited" window is the Event Patterns window, so typing <F2> is an alternative way of moving the cursor to this window. Another <F2> will bring the cursor back to the menu bar.

↓

Type <Down> to move the cursor to the event VME0.

→ →

Type <Right> twice to move the cursor to the Address field. The cursor should now be placed as in the figure below:

```
  +----------------+----------------+----------------+----------------+----------------+----------------+----------------+----------------+----------------+----------------+
  |                |                |                |                |                |                |                |                |                |
  +----------------+----------------+----------------+----------------+----------------+----------------+----------------+----------------+----------------+
  | BgL-AM-Address | Date            | Size           | Cycle          | Stat-IRQ7:1+   | Iack-Fail-In3:0|               |                |                |
  +----------------+----------------+----------------+----------------+----------------+----------------+----------------+----------------+----------------+
  | Anything       | x xx xx xxxxxxx | xxxxxxx x x x x | x xx x xxx x x x | x xx x xxx x x x | x xx x xxx x x x | x xx x xxx x x x | x xx x xxx x x x | x xx x xxx x x x |
  +----------------+----------------+----------------+----------------+----------------+----------------+----------------+----------------+----------------+
  | VME0           | x xx xx xxxxxxx | xxxxxxx x x x x | x xx x xxx x x x | x xx x xxx x x x | x xx x xxx x x x | x xx x xxx x x x | x xx x xxx x x x | x xx x xxx x x x |
  +----------------+----------------+----------------+----------------+----------------+----------------+----------------+----------------+----------------+
  | VME1           | x xx xx xxxxxxx | xxxxxxx x x x x | x xx x xxx x x x | x xx x xxx x x x | x xx x xxx x x x | x xx x xxx x x x | x xx x xxx x x x | x xx x xxx x x x |
  +----------------+----------------+----------------+----------------+----------------+----------------+----------------+----------------+----------------+
```

00400000

Type 00400000 to change the Address field to 0x00400000 [400000 Hex]. Typing errors can be corrected by moving the cursor with the <Left> and <Right> arrow keys, and then type over the erroneous digit(s).

↓

Type <Cr> to finish editing of the field:

```
  +----------------+----------------+----------------+----------------+----------------+----------------+----------------+----------------+----------------+----------------+
  |                |                |                |                |                |                |                |                |                |
  +----------------+----------------+----------------+----------------+----------------+----------------+----------------+----------------+----------------+----------------+
  | BgL-AM-Address | Date            | Size           | Cycle          | Stat-IRQ7:1+   | Iack-Fail-In3:0|               |                |                |                |
  +----------------+----------------+----------------+----------------+----------------+----------------+----------------+----------------+----------------+----------------+
  | Anything       | x xx xx xxxxxxx | xxxxxxx x x x x | x xx x xxx x x x | x xx x xxx x x x | x xx x xxx x x x | x xx x xxx x x x | x xx x xxx x x x | x xx x xxx x x x |
  +----------------+----------------+----------------+----------------+----------------+----------------+----------------+----------------+----------------+----------------+
  | VME0           | x xx xx xxxxxxx | xxxxxxx x x x x | x xx x xxx x x x | x xx x xxx x x x | x xx x xxx x x x | x xx x xxx x x x | x xx x xxx x x x | x xx x xxx x x x |
  +----------------+----------------+----------------+----------------+----------------+----------------+----------------+----------------+----------------+----------------+
  | VME1           | x xx xx xxxxxxx | xxxxxxx x x x x | x xx x xxx x x x | x xx x xxx x x x | x xx x xxx x x x | x xx x xxx x x x | x xx x xxx x x x | x xx x xxx x x x |
  +----------------+----------------+----------------+----------------+----------------+----------------+----------------+----------------+----------------+----------------+
```

*Notice the ♦*

The diamond indicates that the event pattern contains a pattern different from all don’t care. Used as a trigger condition, this event, VME0, will now cause the VBT-325 to trigger for bus cycles where the address is equal to 0x00400000. Used as a storage qualifier, the VBT-325 will store all cycles where the address is 0x00400000, and skip all others.
→ → Type <Right> twice. The cursor should now be in the Size field as in the figure below:

![Event Patterns Table]

A scrollable and a fixed part

Notice the divider, " ", between the Data and the Size fields. This divider marks the border between the fixed area, to the left, and the scrollable area of the Events window.

→ .. Type <Right> seven times, or until the window contents start to scroll sideways. The cursor should now be placed as in the figure below:

![Event Patterns Table]

The More.. mark indicates if, and where, invisible field columns are hidden.

Field mnemonics and other options

Field options

Most fields have a dialog box for selection of predefined values, assert negation and other field options.

← .. Type <Left> until the cursor positioned on the Cycle field.

↓ Type <Cr> to open the dialog box for the Cycle field:

![Cycle Dialog]

Bit list

The cycle field consist of the AM1, Vme64*, RmwBlk and WR*.

WR*

The WR* bit, corresponding to the VMEbus WR* signal, is the only bit that can be edited directly by typing an 0, 1 or X into the field. The contents of the remaining bits, AM1, Vme64* and RmwBlk are set by selecting one of the predefined values. See the next page.
Type <Down> to list predefined values:

<table>
<thead>
<tr>
<th>Mnemonic</th>
<th>AM1</th>
<th>Vme64*</th>
<th>RmwBlk</th>
<th>WR*</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Numeric</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0,1, X</td>
<td>WR* will be set to the value typed directly into the field.</td>
</tr>
<tr>
<td>RD</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>1</td>
<td>WR* = 1, giving any read cycle.</td>
</tr>
<tr>
<td>WRI</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>0</td>
<td>WR* = 0, giving any write cycle.</td>
</tr>
<tr>
<td>RBLK</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>Read block</td>
</tr>
<tr>
<td>WBLK</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>Write block</td>
</tr>
<tr>
<td>RMBL</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>Read VME64 block</td>
</tr>
<tr>
<td>WMBL</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>Write VME64 block</td>
</tr>
<tr>
<td>RMW</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>A read-modify-write cycle</td>
</tr>
</tbody>
</table>

RD or WRI

These two values equals writing 1 or 0 directly into the field. The other symbolic values contain additional values for AM1, i.e. the VMEbus signal AM1, and the internally generated signals Vme64* and RmwBlk.

↓ ↑ ↓ <Esc>

Use <Up> and <Down> to move the cursor to the desired value, then type <Cr> to select. Type <Esc> to close the drop-down box without making any selection. In both cases, the focus is returned to the Field Options dialog.

WRI

For the sake of this example, select the symbol WRI and type <Cr>: 

<table>
<thead>
<tr>
<th>Cycle</th>
<th>Bits : (AM1) (Vme64*) (RmwBlk) WR*</th>
</tr>
</thead>
<tbody>
<tr>
<td>Predefined values :</td>
<td>WRI</td>
</tr>
<tr>
<td>&lt; Ok &gt;</td>
<td>&lt;Cancel&gt;</td>
</tr>
</tbody>
</table>
Type <Cr> to close the field options dialog for the Cycle field:

<table>
<thead>
<tr>
<th>VME Event Patterns</th>
</tr>
</thead>
<tbody>
<tr>
<td>Byt-Am-Address-Data-&lt;Cycle-Stat-IRQ7:1+-Lack-Fail-In3:9-P2trg+</td>
</tr>
<tr>
<td>AnyThing x xx xxxxxxx x xxxxxxx x x x xxx x</td>
</tr>
<tr>
<td>VME0  x xx 00000000 xxxxxxx xxx x x x xxx x</td>
</tr>
<tr>
<td>VME1  x xx xxxxxxx xxxxxxx x x x xxx x</td>
</tr>
</tbody>
</table>

VME0

The event VME0 will now form a trigger [or storage qualifier] for Write cycles to the address 0x00400000.

### Clearing contents of fields

Typing X'es into a field will set the corresponding bit(s) to don’t care. An x means that this bit [signal on the bus] will be ignored when the tracer is looking for a trigger, or when using this event as storage qualifier.

By typing <Del>, all bits in a field will be set to x.

### Hiding field columns

If positioned on an empty [all X] field, you will be asked to hide the field column.

Type <Right> twice, or until the cursor is on the IRQ7:1 field:

<table>
<thead>
<tr>
<th>VME Event Patterns</th>
</tr>
</thead>
<tbody>
<tr>
<td>Byt-Am-Address-Data-&lt;Cycle-Stat-IRQ7:1+-Lack-Fail-In3:9-P2trg+</td>
</tr>
<tr>
<td>AnyThing x xx xxxxxxx x xxxxxxx x x x xxx x</td>
</tr>
<tr>
<td>VME0  x xx 00000000 xxxxxxx xxx x x x xxx x</td>
</tr>
<tr>
<td>VME1  x xx xxxxxxx xxxxxxx x x x xxx x</td>
</tr>
</tbody>
</table>

Type <Del> to hide the IRQ7:1 field:

```
<Hide Signal Field >

<Cancel>
```

Type <Space> or <Cr> to hide the field IRQ7:1:

<table>
<thead>
<tr>
<th>VME Event Patterns</th>
</tr>
</thead>
<tbody>
<tr>
<td>Byt-Am-Address-Data-&lt;Cycle-Stat-IRQ7:1+-Lack-Fail-In3:9-P2trg+</td>
</tr>
<tr>
<td>AnyThing x xx xxxxxxx x xxxxxxx x x x xxx x</td>
</tr>
<tr>
<td>VME0  x xx 00000000 xxxxxxx xxx x x x xxx x</td>
</tr>
<tr>
<td>VME1  x xx xxxxxxx xxxxxxx x x x xxx x</td>
</tr>
</tbody>
</table>

The field column IRQ7:1 is now hidden. It can be re-inserted later, if needed, at any place in the event window.
5 Operation

Adding field columns

Type `<Ins>` to insert a field column to the left of the cursor:

<table>
<thead>
<tr>
<th>VMETRO VBT-325C</th>
<th>VME Bus Analyzer</th>
<th>Select Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trace</td>
<td>Edit</td>
<td>Target</td>
</tr>
<tr>
<td>VME Event Patterns</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Anything</td>
<td>x</td>
<td>xx</td>
</tr>
<tr>
<td>VMEO</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>VMEL</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>VMER</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>VMES</td>
<td>x</td>
<td>x</td>
</tr>
</tbody>
</table>

Use `<Up>` and `<Down>` to move the cursor to the desired signal [field column] name. In some situations, the length of the list may exceed the available screen space. In this case, type `<Down>` repeatedly to move to the bottom of the list box. Then, use `<Down>` to scroll the box until the wanted signal shows up. The list of fields are alphabetized to ease location of signals.

Type `<Esc>` to select a signal and close the list box. Typing `<Esc>` closes the list without making any selection.
Renaming, clearing, deleting and copying entire events

← .. Type <Left> until the whole event is highlighted:

<table>
<thead>
<tr>
<th></th>
<th>BgI-AM-Address—Data—Size-Cycle-Stat-IRQ7:1—Lack-Fail-In3:0—</th>
</tr>
</thead>
<tbody>
<tr>
<td>Anything</td>
<td>x xx xxxxxxxxx xxxxxxxxx xx x xx xxxxxxxxx x xx xxx</td>
</tr>
<tr>
<td>VME0</td>
<td>x xxx 08:4000000 xxxxxxxxx xxx WRI xx xxxxxxxxx x xx xxxxx</td>
</tr>
<tr>
<td>VME1</td>
<td>x xx xxxxxxxxx xxxxxxxxx xxx x xx xxxxxxxxx x xx xxx</td>
</tr>
</tbody>
</table>

Rename event

To rename the event, do as follows:

↓ Type <cr> to open editing.

MyEvent ↓ Type the new name, MyEvent, and then <cr> to close:

<table>
<thead>
<tr>
<th></th>
<th>BgI-AM-Address—Data—Size-Cycle-Stat-IRQ7:1—Lack-Fail-In3:0—</th>
</tr>
</thead>
<tbody>
<tr>
<td>Anything</td>
<td>x xx xxxxxxxxx xxxxxxxxx xx x xx xxxxxxxxx x xx xxx</td>
</tr>
<tr>
<td>MyEvent</td>
<td>x xxx 08:4000000 xxxxxxxxx xxx WRI xx xxxxxxxxx x xx xxxxx</td>
</tr>
<tr>
<td>VME1</td>
<td>x xx xxxxxxxxx xxxxxxxxx xxx x xx xxxxxxxxx x xx xxx</td>
</tr>
</tbody>
</table>

VME0 changed to MyEvent.

<Del> Type <Del> to clear or delete the event:

Select <Clear> to set all fields in the event to don't care. Select <Remove> to remove the event pattern entirely. Type <Esc> to close the box without doing anything.

Adding events

<Ins> Type <Ins> to insert an event above the current event. The new event will be a copy of the current one:

<table>
<thead>
<tr>
<th></th>
<th>BgI-AM-Address—Data—Size-Cycle-Stat-IRQ7:1—Lack-Fail-In3:0—</th>
</tr>
</thead>
<tbody>
<tr>
<td>Anything</td>
<td>x xx xxxxxxxxx xxxxxxxxx xx x xx xxxxxxxxx x xx xxx</td>
</tr>
<tr>
<td>MyEvent</td>
<td>x xxx 08:4000000 xxxxxxxxx xxx WRI xx xxxxxxxxx x xx xxxxx</td>
</tr>
<tr>
<td>VME1</td>
<td>x xx xxxxxxxxx xxxxxxxxx xxx x xx xxxxxxxxx x xx xxx</td>
</tr>
</tbody>
</table>

↓ Type <cr> to open editing.
5 Operation

Change the name New to a name of your choice, and type <Cr>:

```
MyNewOne x xx 00000000 xxxxxxxxx xxx x xx x xxxxx x xx xxx
```

**Note!** If you try to keep the name New, the following error message will be printed:

```
Err: Fill in other name or type ESC to abort.
```

**Add event at end** Events can also be added at the end of the list. Type <Down> until the cursor is one line beyond the last event [only possible when the whole event is highlighted], then type <Ins>:

```
NewName x xx 00000000 xxxxxxxxx xxx x xx x xxxxx x xx xxx
```

Change the name New to a name of your choice and type <Cr>.

**Individual signals**

Most signals in signal groups, as Size, Cycle, Stat etc., are also available as individual signals.

**See Chapter 6** All signal fields are described in detail in Chapter 6, starting at page 150.
Different Signal templates in State and Timing mode

Changing the sampling mode from State to Timing (see page 33) also changes how the event patterns are displayed. All control signal will now be displayed individually rather than grouped. Changing from Timing to State, reverse this process. Note that only the display of the event patterns are changed, not the logical contents of the events:

Timing mode vs. State mode

The screen above shows the same events as used in the examples on the preceding pages, but now displayed in Timing mode. Notice that the symbol WRI that was displayed in the Cycle field in State mode, is presented as WR*==0. If WR* is changed to 1, and then the sampling mode back to State, Cycle will be presented as RD.

Field columns [or signals] can be added and deleted in any mode, as previously explained. A field column, inserted and edited in either mode, will also show up in the other mode, unless the contained bits can be displayed correctly by some other field column(s) in that mode.
5 Operation

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):

<table>
<thead>
<tr>
<th>BgL AM Address Data</th>
<th>Size Cycle Stat IRQ7:1 Latch BR3:0 Fail</th>
</tr>
</thead>
<tbody>
<tr>
<td>Anything</td>
<td>x xx xxxxxxx xxxxxxxx xx x xx xxxxxxx x xxx xx</td>
</tr>
<tr>
<td>VMEO</td>
<td>+ xx 80 400000 xxxxxxxx xxx WRI xx ....Z x 32xx xx</td>
</tr>
<tr>
<td>VMFI</td>
<td>x xx xxxxxxx xxxxxxxx xx x xx xxxxxxx x xxx xx</td>
</tr>
</tbody>
</table>

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.

BgL editing

The BgL field [= Bus grant Latched] represents the BG3:0* signals latched during arbitration of the VMEbus cycle. See also page 33.

<table>
<thead>
<tr>
<th>BgL AM Address Data</th>
<th>Size Cycle Stat IRQ7:1 Latch BR3:0 Fail</th>
</tr>
</thead>
<tbody>
<tr>
<td>Anything</td>
<td>x xx xxxxxxx xxxxxxxx xx x xx xxxxxxx x xxx xx</td>
</tr>
<tr>
<td>VMEO</td>
<td>+ 1 xx 80 400000 xxxxxxxx xxx WRI xx ....Z x 32xx xx</td>
</tr>
<tr>
<td>VMFI</td>
<td>x xx xxxxxxx xxxxxxxx xx x xx xxxxxxx x xxx xx</td>
</tr>
</tbody>
</table>

The BgL field is normally edited by selecting a predefined symbol, but some shortcuts exist:

<table>
<thead>
<tr>
<th>Shortcut</th>
<th>Symbol</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;0&gt;</td>
<td>---0</td>
<td>BG0* asserted</td>
</tr>
<tr>
<td>&lt;1&gt;</td>
<td>--1-</td>
<td>BG1* asserted</td>
</tr>
<tr>
<td>&lt;2&gt;</td>
<td>-2--</td>
<td>BG2* asserted</td>
</tr>
<tr>
<td>&lt;3&gt;</td>
<td>3---</td>
<td>BG3* asserted</td>
</tr>
<tr>
<td>&lt;4&gt;</td>
<td>----</td>
<td>No BG* asserted</td>
</tr>
</tbody>
</table>
Address/Data options

Field Options  Typing <Cr> in the Address field brings up a dialog box that will give the options for the Address field. A similar dialog box will also be displayed for the Data field. All examples given for the Address, except for some A64/D64 details, are also valid for the Data field:

```
Address   Data   | Size   | Cycle | Stat | IRQ7:1* |
-----------------|--------|-------|------|---------|
xxxxxxxxxxxx  xxx  x  xxx  xxxxxxxx |
00100000  xxxxxxxx  xxx  WRI  xxxxxxxx |
xxxxxxxxxxxx  xxx  x  xxx  xxxxxxxx  
```

Address

Bits: Address bits A(31:1) DS1* as A00

<Edit range...>  [ ] Not
<Binary details...>  [ ] 64 bit

< Ok >  <Cancel>

Edit Range

00100000  Fill in wanted low range, type <Tab> to move to high range,
002FFFFF  fill in high range, then type <Cr> to close the box.

00100000-002FFFFF

Edit range without using the Field Options dialog

Move the cursor to the Address [or Data] field. Type the range directly as shown below:

00100000-002FFFFF ↓

I.e., type the low range, then a hyphen, "-", then the upper range. Use the arrow keys to move the cursor within the field for correcting typing errors. The example pattern will give match for any address between 0x00100000 and 0x002FFFFF, the boundary values included.
Binary details

Binary details makes it possible to specify don't care bits down the bit level:

<table>
<thead>
<tr>
<th>Address: Binary details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expanded hex : 00-400000</td>
</tr>
</tbody>
</table>

Fill in HEX digits or binary nibbles in parenthesis as shown here: (1XX0). The example gives 1 in the two high order bits and don't care in the two low order bits.

Individual bits

within a 4-bit-group, or nibble, can be specified binary like this:

(1XX0)

The "(" marks the start of the nibble, the "0", "1" and "X" are bit values, and the ")" marks the end of the nibble. More than one nibble in the Address, or any other hex field, can be specified like this. The example will give match when bit 3 is one and the least significant bit is zero.

0040000$

Nibbles that contains single don't care bits will be displayed as an "$" in the normal event display. Use the <Binary details> to display and/or change these individual X'es.

The Not operator

[X] Not

Negation, i.e. the Not operator, can be specified for Address and Data values [and some other fields] by enabling the "[X] Not" option.

!00400000

This results in a "!" being displayed before the value/mnemonic. It is also possible to type <!> directly in front of the value/mnemonic. The example pattern will give match for all values, except for 0x400000.

Outside range

<!>

Specifying "!" on a range, programs the VBT-325 to give match for all values outside the range. The following pattern, !00100000-002FFFFC, gives match for all values from 0x0 up to, but not including, 0x100000 and for values from, but not including 0x2FFFFC through 0xFFFFFFE.

Restrictions

There are some restrictions when you use NOT together with an address/data that includes "don't care" bits.

For legal combinations, see the on-line Help for Negate.
A64/D64

[X] 64 bit  For use in VME64 applications, the VBT-325 allows 64-bit addresses or data to be specified in the event patterns. Enabling for the 64 bit option on the Address field makes the following change to the event:

<table>
<thead>
<tr>
<th>VME Event Patterns</th>
</tr>
</thead>
<tbody>
<tr>
<td>BgL-AM-Address-Data</td>
</tr>
<tr>
<td>Anything</td>
</tr>
<tr>
<td>VME0</td>
</tr>
<tr>
<td>VME1</td>
</tr>
</tbody>
</table>

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.

A64  

uses A(31:1) as the lower address bits, then D(31:0) as the upper address bits, A(63:32).

D64  

Uses D(31:0) as the lower 32 data bits [as ordinary 32 bit cycles], LWORD* as D32, then A(31:1) as D(63:33):

<table>
<thead>
<tr>
<th>VME Event Patterns</th>
</tr>
</thead>
<tbody>
<tr>
<td>BgL-AM-Address-Data</td>
</tr>
<tr>
<td>Anything</td>
</tr>
<tr>
<td>A64</td>
</tr>
<tr>
<td>D64</td>
</tr>
<tr>
<td>VME2</td>
</tr>
</tbody>
</table>

A64 bit range  

For the Address field, it is also possible to set ranges on A64. In this case, only the lower portion of the 64-bit boundaries are shown, with some dots ... indicating this. To see the entire value, do <Edit range>.

64-bit range is not allowed on the Data field.
Sequencer

The Sequencer defines how the output of the event comparators 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 and TIMING. See page 33 for technical details. 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.

Single event and Sequencer modes

Single Event mode

The Sequencer program is initially set to Single Event mode, where the trigger condition is a single event pattern, the current event pointed at in the Event Patterns window:

```
<table>
<thead>
<tr>
<th>VME Bus Analyzer</th>
<th>Setup: Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trace</td>
<td>Edit</td>
</tr>
</tbody>
</table>
```

The current event is used as trigger condition in Single Event mode.

Changes

The Trigger event, the sampling mode and speed if timing, and the trigger position is changed by commands in the Edit menu. The Sequencer window will be updated to show the selected parameters.
Change of Trigger condition

To change the trigger condition to another event name, simply enter the Event Patterns window, and move the cursor to the wanted trigger event:

Change of sampling mode

The default sampling mode is State, as the "*" indicates. To change sampling mode [in Single Event mode] to Timing, execute the sub-command 50 MHz Timing. To change back, execute the sub-command State. The sampling mode options are covered in detail later. See also page 33 for technical details.

Change of Trigger position

The default Trigger position is Start of Trace. This means that when the trace is displayed, no samples are shown before the trigger. The other options are as indicated. Refer to page 31 for technical details.
5 Operation

Sequencer mode

Edit/Sequencer

If a more complex trigger, store qualifier or count/delay is needed, you may edit the Sequencer program explicitly by the `Edit/Sequencer` command:

Type `<cr>` to enter the Sequencer window. As soon as the Sequencer window is entered, the tracer is forced into Sequencer Mode. In this mode, the lock to the current event in the Event Patterns window is broken. Everything is fully controlled within the Sequencer program window:

---

Return to Single Event mode

In order to return to Single Event mode, type `<del>` and select the option `<Single Event Mode>`.

Editing the Sequencer program

When you enter the Sequencer, the cursor will be placed at the first editable line.

Opens editing of the current Sequencer statement:
Type <Down> to set the cursor on 50 MHz TIMING. Then type <Cr> to select. At this point, typing <Esc> will close the editing without doing any changes.

```
<table>
<thead>
<tr>
<th>VME Sequencer</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt; 1.a: Sampling in 50 MHz TIMING mode</td>
</tr>
<tr>
<td>1.b: Store (ALL)</td>
</tr>
<tr>
<td>&gt; 1.c: If (ANYTHING) then</td>
</tr>
<tr>
<td>1.d: Trigger at START of trace</td>
</tr>
</tbody>
</table>
```

**Change event**

↓↓ Type <Down> twice, or until the cursor is at the If statement:

```
<table>
<thead>
<tr>
<th>VME Sequencer</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt; 1.a: Sampling in 50 MHz TIMING mode</td>
</tr>
<tr>
<td>1.b: Store (ALL)</td>
</tr>
<tr>
<td>&gt; 1.c: If (ANYTHING) then</td>
</tr>
<tr>
<td>1.d: Trigger at START of trace</td>
</tr>
</tbody>
</table>
```

↓ Then, type <Cr> to edit:

```
<table>
<thead>
<tr>
<th>VME Sequencer</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt; 1.a: Sampling in 50 MHz TIMING mode</td>
</tr>
<tr>
<td>1.b: Store (ALL)</td>
</tr>
<tr>
<td>&gt; 1.c: If (ANYTHING) then</td>
</tr>
<tr>
<td>1.d: Trigger at START of trace</td>
</tr>
</tbody>
</table>
```

↓ Type <Down> to open the list of selectable events:

```
<table>
<thead>
<tr>
<th>VME Sequencer</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt; 1.a: Sampling in 50 MHz TIMING mode</td>
</tr>
<tr>
<td>1.b: Store</td>
</tr>
<tr>
<td>&gt; 1.c: If (ANYTHING) then</td>
</tr>
<tr>
<td>1.d: Trigger at START of trace</td>
</tr>
</tbody>
</table>
```

VME0 Type <Down> then <Cr> to select the event VME0:
Notice the cursor position. If all you wanted was to change the event from \textit{ANYTHING} to \textit{VME0}, type \texttt{<Cr>} to confirm. If you want to follow the example, read on.

\textbf{Edit event expressions}

An Event expression is one or more event names as defined in the Event Pattern Window in a combinatorial expression. A simple example is:

\[ A+B*C \]

The logical AND operator, \texttt{"*"}, have precedence over the OR operator, \texttt{"+"}. Brackets can be used to changed the order of evaluation:

\[ A+(B*C) \quad \text{is equivalent to} \quad A+B*C \]

due to the order of evaluation.

\[ (A+B)*C \quad \text{is not equivalent to} \quad A+B*C \]

The parenthesis around the OR expression forces the OR to be evaluated before the AND.

\[ !(A+B)*C \quad \text{is not equivalent to} \quad !A+B*C \]

Negation, i.e. the logical NOT operator, can be used on single event names, or on sub-expressions within brackets. The logical NOT operator, \texttt{"!"} is always evaluated first.

\textbf{Editing keys}

The status line hints about some of the available editing keys:

\texttt{Ok, <INS> <DEL> <!-Not> <+:-Or> <+:-And> <ESC;/Cancel> <CR/-OK>}

\texttt{<Ins>} \quad \text{Inserts a event name into the event expression at the cursor position.}

\texttt{<Del>} \quad \text{Deletes the current token, i.e. event name, operator, or bracket.}

\texttt{^O} \quad \text{Type Ctrl+O to undo your last editing.}

\texttt{<Esc>} \quad \text{Type <Esc> to cancel all changes made to the event expression.}

\texttt{<End>} \quad \text{Type <Cr> to confirm and close editing of the event expression.}

\texttt{<Home>} \quad \text{Moves the cursor to the leftmost column of the event expression.}

\texttt{<End>} \quad \text{Moves the cursor to the right of the rightmost column of the event expression.}
Operators

* + !

Symbols for the logical operators OR, AND and NOT.

( )

Parenthesis can be used to change the order of how the expression is evaluated. See the previous page.

Error message

Illegal expressions, as unbalanced parenthesis, etc. will cause the following dialog box to be displayed:

```
<table>
<thead>
<tr>
<th>Syntax Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>o '}' can not follow 'VMEB'.</td>
</tr>
<tr>
<td>You must either correct the event expression or cancel the editing.</td>
</tr>
<tr>
<td>Do you want to edit or cancel?</td>
</tr>
</tbody>
</table>
```

Example

In the following example, the current If statement will be changed to form a trigger on VME0 or VME1 and !VME2:


+ Notice the cursor position. Type an <+> to insert an OR operator to the right of the event name VME0:

```
1.a: Sampling in 50 MHz TIMING node
1.b: Store [ALL]
> 1.c: If (VMEB) then
1.d: Trigger at START of trace
```

<Ins> Type <Ins> to insert an event name at the cursor position:

```
<table>
<thead>
<tr>
<th>VME Sequencer</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.a: Sampling in 50 MHz TIMING node</td>
</tr>
<tr>
<td>1.b: Store [ALL]</td>
</tr>
<tr>
<td>&gt; 1.c: If (VMEB+VME0) then</td>
</tr>
<tr>
<td>1.d: Trigger at START of trace</td>
</tr>
</tbody>
</table>
```

↓ ↓ ↓ ↓ The Tracer suggest VME0. Type <Down> to open list of events, and select the event VME1. Type <Cr> once to close the list of events:

```
<table>
<thead>
<tr>
<th>VME Sequencer</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.a: Sampling in 50 MHz TIMING node</td>
</tr>
<tr>
<td>1.b: Store [ALL]</td>
</tr>
<tr>
<td>&gt; 1.c: If (VMEB+VME1) then</td>
</tr>
<tr>
<td>1.d: Trigger at START of trace</td>
</tr>
</tbody>
</table>
```
Type <*> to insert an AND operator just after the name VME1:

1.a: Sampling in 50 MHz TIMING mode
1.b: Store (ALL)
1.c: If \((\text{VME0}+\text{VME1})\) then
1.d: Trigger at START of trace

Type <Ins> and insert the event VME2 as explained earlier:

1.a: Sampling in 50 MHz TIMING mode
1.b: Store (ALL)
1.c: If \((\text{VME0}+\text{VME1}+\text{VME2})\) then
1.d: Trigger at START of trace

Type <Left> until the cursor is highlighting the name VME2. Type <!> to negate VME2:

1.a: Sampling in 50 MHz TIMING mode
1.b: Store (ALL)
1.c: If \((\text{VME0}+\text{VME1}+\text{VME2})\) then
1.d: Trigger at START of trace

The event expression is now as wanted. Type <Cr> to finish the editing.

**Adding states to the Sequencer program**

### State Machine

The Sequencer is a state machine which can be in one of 16 possible states, of which 15 can be programmed by the user. Certain rules control the transition from one state to another:

1. When started, the tracer samples in STATE mode, i.e. one sample per VMEbus cycle. Line 1.b orders the tracer to store all samples in the trace memory. Each sample is compared to the patterns in the current event comparator, VME0. The Sequencer program will stay in state #1, storing all cycles until a cycle matching VME0 occurs. If equal to VME0, the Sequencer will change to state #2.

2. The tracer will compare all incoming bus cycles with the pattern VME1. Only samples that are equal to VME1 will be stored. The Sequencer will stay in state #2 until a sample matching VME2 occurs.
2 ⇒ 3 When VME2 occurs, the tracer triggers during the transition from state #2 to state #3. Then, the tracer starts to fill the rest of the trace buffer. Note that the trigger sample will also be stored.

**State #3** Only samples matching VME3 will be stored. Before the sampling stops, a given number of samples [matching VME3] are stored. The Sequencer program in the example results in 25% of the stored samples stored after the trigger.

When the trace becomes full, the tracer will stop and display the captured trace.

**Example**

Typing <Del> opens a dialog box with the following options:

```
Delete

<Restore default program>
<Set Single Event mode>
<CANCEL>
```

Type <Cr> to restore the default Sequencer program. The <Del> dialog box is contexts sensitive and will also provide options for deleting entire If .. Elsif .. Else trees and/or the current line when possible:

When the default program is restored, type <Down> until the cursor is positioned on the If (Anything) statement:

VME1 Change ANYTHING to VME1, then type <Down> until the cursor is positioned on the Trigger statement:

```
VME Sequencer

1.a: Sampling in STATE mode
1.b: Store (ALL)
1.c: If (VME1) then
> 1.d: Trigger at START of trace
```

When the default program is restored, type <Down> until the cursor is positioned on the Trigger statement:

```
VME Sequencer

1.a: Sampling in STATE mode
1.b: Store (ALL)
1.c: If (VME1) then
> 1.d: Trigger at MIDDLE of trace
```

Change the Trigger position to MIDDLE of trace:
5 Operation

↓ Type <Down> to move the cursor to the empty line below the Trigger statement:

```
1.a: Sampling in STATE node
1.b: Store (ALL)
1.c: If (VME1) then
  1.d: Trigger at MIDDLE of trace
> -
```

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

<Ins> Type <Ins> to insert a new Sequencer statement:

```
1.a: Sampling in STATE node
1.b: Store (ALL)
1.c: If (VME1) then
  1.d: Trigger at MIDDLE of trace
> -
```

Ok. <CR-Enter> <ESC/.Close> <Del>

↓↓ Type <Down> to move the cursor to the Store operator, then <Cr> to select Store:

```
1.a: Sampling in STATE node
1.b: Store (ALL)
1.c: If (VME1) then
  1.d: Trigger at MIDDLE of trace
> 2.a: Store (ALL)
```

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

VME0 Change ALL to VME0:

```
1.a: Sampling in STATE node
1.b: Store (VME0)
1.c: If (VME1) then
  1.d: Trigger at MIDDLE of trace
> 2.a: Store (VME0)
```

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

Done! The new Sequencer program will provide:

Store all samples until a sample matching VME1 occurs, then trigger and store only samples matching VME0 until the trace buffer is full and the sampling stops.
Sequencer Reference

General structure of one state

Possible actions

In each state, a number of actions can be defined to take place, like:

- Sampling in STATE or TIMING mode,
- Store only samples matching 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 captured trace.

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

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

```plaintext
Sampling in {STATE|TIMING} mode
Store (<Event Expression>)
If (<Event expression>) then
  Trigger | Goto <State> | Halt
:
```

or

```plaintext
Sampling in {STATE|TIMING} mode
Store (<Event Expression>)
Count <Numeric expression> of (<Event expression>) then
  Trigger | Goto <State> | Halt
:
```

or

```plaintext
Sampling in {STATE|TIMING} mode
Store (<Event Expression>)
Delay <Time expression> then If (<Event expression>) then
  Trigger | Goto <State> | Halt
:
```

Each If, Count or Delay may also have an Elsif .. Else branch as shown below. More than one Elsif can be used.

\[
\text{Elsif } (<\text{Event Expression}> ) \text{ then} \\
\text{ Trigger } | \text{ Goto } <\text{State}> | \text{ Halt} \\
\text{ Else} \\
\text{ Trigger } | \text{ Goto } <\text{State}> | \text{ Halt} \\
\]

- The colon “:” indicates that other states of the same construction can be entered at 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.

**Sequencer Notation**

**UPPER/lower case**

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

\[\text{VME0} \quad \text{both predefined, as VME0, and user-defined events are shown as in Event Patterns window.}\]

\[\text{Trigger} \quad \text{Operators are shown with initial Caps, otherwise lower case.}\]

\[\text{then} \quad \text{Filler words, like in, of etc. are shown in lower case.}\]

**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 ...

Current state indicator

> 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.

The maximum number of usable events

Max four events Note that it is not possible to use more than four different event names in the
Sequencer program at the same time. A warning will be printed when the fifth
event is taken into use. There is no limit on how many times one event name can
be used in event expressions in the same program. The same event may serve
both as both as Trigger condition(s), Store and Count conditions.

XMEM325-PB The XMEM325-PB allows up 8 events to be used in the “VME” [not
“XMEM_VME”] Sequencer program in State mode\(^1\). This is done by sharing
resources with the VBT-325 itself. For this reason, the Dual analyzer capability
of the XMEM325 cannot be used at the same time as more than 4 events are
used.

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 \(<\text{Mode}>\) mode

Parameter

The \(<\text{Mode}>\) parameter can be selected here as one of:

\[
\text{STATE} \\
\text{<Speed> TIMING}
\]

The timing sampling speed, \(<\text{Speed}>\), is selected by executing
Edit/Sampling Mode, and then selecting one of the indicated speeds.

---

\(^1\) The XMEM325-PB 128K low cost model does not support 8 events.
Operation

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. 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 in a special way in the Sequencer hardware, so that it does not consume a separate event (out of the four usable).

Syntax

\[
\text{Store \{\langle Event expression\rangle | \text{ALL} | \text{NOTHING}\}}
\]

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 ordinary 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

\[
\text{If \langle Event expression\rangle | \text{ANYTHING} \rangle \text{ then :}}
\]

Elsif \langle Event expression\rangle | \text{ANYTHING} \rangle \text{ then :}

Else

Indents

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. In such cases, the used will be asked to confirm the state number.
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 StateNumber

StateNumber = 2, means 2.a.

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 {{<EvExpr>|ANYTHING}} then

Where N is a number 2..1048576. If works as “Count 1”.

Up to 4 Count statements can be used in the sequencer program.

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 {ns|us|ms} then if {{<EvExpr>|ANYTHING}} then

Where N is a number 1..999 of the given delay unit, ns, μs or ms,

The delay time can be minimum 40ns, maximum 335ms. The value is automatically truncated to the selected clock period, which can be 40, 80, 160 or 320ns. Up to 3 Delay statements can be used in the sequencer program.

Note #1

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 #2

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.
5 Operation

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 {START|25%|MIDDLE|75%|END} of trace

As the parameter for Trigger must be the same throughout the Sequencer program, Halt can be used to replace “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 the tracer. 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-Elseif sequence if no Else is specified, so that the If-test will always be repeated for the next sample 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 sequences

Loose 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

\[ A \Rightarrow B \Rightarrow X \Rightarrow C \Rightarrow Y \Rightarrow D \]

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 ..

Tight sequence  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

\[ A \Rightarrow B \Rightarrow C \Rightarrow D \]

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 below:

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 an Else.
5 Operation

Sequencer programming 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 what happened in this period after the trigger cycle. 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

VSB Sequencer examples

**Multiplexed bus** The VSBbus is a multiplexed bus, meaning that both the address and the data is transferred on the same physical lines. The VSBbus cycle is divided into two phases, the address phase where the signal DS*=1, and the data phase, where DS*=0. The multiplexing makes triggering on a specific address and data pattern a bit more complex than for the VMEbus, since one event must be used to specify the address, and another event must be used to specify the data. The following examples show how to solve some of the VSBbus difficulties.

Trigger on Address range and Data
Store data only when address within given range (no Trigger)

Firmware 5.01

Does not allow negate, "!", on the VSB Address and Data fields. As a workaround, insert and use the field AD(31:0) instead of Address. Firmware 5.02 and newer does not have this restriction.
Store and trigger on data when address within given range

Above is the event patterns for store and trigger 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

**Firmware 5.01** Does not allow negate, “!” on the VSB Address and Data fields. As a workaround, insert and use the field AD(31:0) instead of Address. Firmware 5.02 and newer does not have this restriction.
Store data when address within given range, trigger on other data

Above is the event patterns for 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

Firmware 5.01 Does not allow negate, “!”, on the VSB Address and Data fields. As a workaround, insert and use the field AD(31:0) instead of Address. Firmware 5.02 and newer does not have this restriction.
Cross-triggering is possible between the VME and P2 analyzers on the VBT-325, as well as to/from the XMEM325-PB/C piggyback module. Cross trigger signals do also exist between the VBT-325 and other piggyback modules.

**Signal names**

See page 155 for cross trigger signal names for the different targets.
Cross-trigger VSB from VME

**Target/VSb**

To specify that the P2 analyzer (e.g. VSB) should trigger when the VME analyzer triggers, go to 'Target/VSb':

**Edit/Event Patterns**

→ ..

Type <Right> until the cursor is on the VMEtrg* signal:

```
<table>
<thead>
<tr>
<th>VME Event Patterns</th>
<th>USB Event Patterns</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Anything</td>
<td>x</td>
</tr>
<tr>
<td>USB0</td>
<td>x</td>
</tr>
<tr>
<td>USB1</td>
<td>x</td>
</tr>
<tr>
<td>USB2</td>
<td>x</td>
</tr>
<tr>
<td>USB3</td>
<td>x</td>
</tr>
</tbody>
</table>
```

**TRIG**

Type <CR> to open the Signal options dialog, change the contents of the field to the predefined symbol TRIG:

```
<table>
<thead>
<tr>
<th>VME Event Patterns</th>
<th>USB Event Patterns</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Anything</td>
<td>x</td>
</tr>
<tr>
<td>USB0</td>
<td>x</td>
</tr>
<tr>
<td>USB1</td>
<td>x</td>
</tr>
<tr>
<td>USB2</td>
<td>x</td>
</tr>
<tr>
<td>USB3</td>
<td>x</td>
</tr>
</tbody>
</table>
```

**Trace/Run Multiple**

This command starts both the VME and the VSB tracers simultaneously. Programmed as above, the VSB tracer will wait until the VME tracer triggers. Combined with other event and Sequencer programming, it is possible to capture concurrent data from both busses in one operation.
Trace Display

The data in the trace display is automatically displayed when the current target has filled out its trace buffer.

**Alphanumeric** display format is by default used for presenting STATE, and mixed STATE-TIMING sampled trace data.

**Waveform** display format is by default used for TIMING sampling.

Additional windows, of both formats, can be opened by the user to form more than one view of the same trace data.

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:

![Trace Display Example](image)

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 alphanumeric trace list form, which is also the default form.
Navigation and 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.

The user can also insert or delete signals/groups, as well as reorganize the signals/groups by using the <Ins> and <Del> keys, as explained for the Event Patterns window. See page 59. Details specific to the Trace windows will be explained below.

Formatting options

Grouped or Single

There are two different ways of presenting the control signals in the trace list, either grouped like Size, Cycle, Status, or as single (individual) signals like DS0*, DTACK* etc.

Grouped format

This is the default alphanumeric format. The grouped presentation form is best suited for state (synchronous) sampling, with the values shown as mnemonics, just as in the Event Patterns window.
Changing alphanumeric formatting template

F_rmat/F

Execute the command F_rmat/F:

<table>
<thead>
<tr>
<th>Formatting Template</th>
</tr>
</thead>
</table>
| Template name: Grouped |< Ok > <Cancel>

Single
Select Single and type <Cr>:

Single signal presentation is recommended, and used by default, for traces captured in TIMING mode.
Go back to the Grouped Data column:

The asterisk, '*', in front of the D command indicates that decoding and formatting is turned on. This is default setting for the display template.
Turn formatting/decoding:

Type <N> to execute the command:

The topmost option, <X>, rules for all field columns, the lower for the current field, <Y>. When turned off, i.e. <N> selected, the Data of the Size field is now displayed as 32-bit hexadecimal fields without any formatting at all.

Most grouped field columns, as Size Cycle etc., will, when possible, display the same values as used in the Event Patterns Window. Use the Format/Decoding command to turn this feature on.

Address and the field is now displayed as 32-bit hexadecimal fields without any formatting at all.

Notice that both formatting templates, and Grouped are configurable.

1 See page  for details.
Navigating the Trace buffer

Several commands are available to make it easy to move around in the trace buffer. These commands falls into 4 categories:

1. Move the cursor to the next or previous line, or the next or previous page.
2. Marking of the current trace line. Two different bookmarks are available.
3. Move the cursor to a marked line, the first or last or the trigger line.
4. Search [forwards] for a given pattern.

Category 1..3 will be covered here. See page 99 for details on searching.

The <Left> and <Right> keys move the cursor sideways. Use these keys to scroll the scrollable area, i.e. the area to the right of the “” marker.

The <Up> and <Down> keys moves the cursor one line up and down, respectively.

The <PgUp> or ^U scrolls the window one page upwards.

The <PgDn> or ^D scrolls the window one page downwards.

Setting bookmarks

Bookmarks are convenient for marking places of interest in the trace buffer. Two marks can also be used to limit statistics functions to a given area, or be used to measure the time between two places.

Markers

Type an <M> to open the Markers menu:

Markers

Set Marker Y
Set Marker Z

Type <Cr> to place the Y marker at the current line:

Bookmark Y is now placed at the TRIG line.

1 These keys are marked <Prev Scrn> and <Next Scrn> on VT220, VT320 and VT420 keyboards. The keys are missing on VT100 keyboards. Use <Ctrl>+<U> and <Ctrl>+<D> instead.
Marketers

\[ \text{Marketers} \]

\[ <M> \text{ to open the Markers menu once more:} \]

\[ \text{Markers} \]

\[ \text{Remove Marker Y} \]

\[ \text{Set Marker Z} \]

As a reminder on that the marker Y exists, the command has been changed to Remove Marker Y. Type <Esc> to close the Markers menu.

The \underline{Jump} commands

\underline{Jump}

Type \[ <J> \] to open the \underline{Jump} menu:

\underline{Jump}

\[ \text{First Line} \]

\[ \text{Last Line} \]

\[ \text{Trigger Line} \]

\[ \text{Marker Y} \]

\[ \text{Marker Z} \]

\[ \text{Line Number} \]

\[ \text{Edge Options} \]

First line  

Jump to the first line in the trace buffer. In this case, with the trigger at \textit{Start of Trace}, the first line is the same as the Trigger line.

Last line  

Jump to the last line of trace. In the example case, this will be line 37676.

Trigger line  

Jump to the Trigger line, i.e. line 0.

Marker Y  

Jump to the line where the bookmark Y is placed. This command is disabled when no marker Y is present.

Marker Z  

Jump to the line where the bookmark Z is placed. This command is disabled when no marker Z is present.

Line Number  

Jump to a given line number.

Edge Options  

See the section on Waveform windows, page 104.
Opening additional windows

Additional windows, or views of the current trace buffer, may be opened and closed when needed.

Type `<w> <o>`, to execute the `window/ 0` command:

```
New window type

(%) Alphanumeric
( _) Waveform

<ok> <cancel>  
```

Type `<cr>`

The new window is labeled VME #2
VME #3 and so on.
Jump/ ast  Execute _ump/_L , and type <Right>

The windows are totally independent views of the same trace memory and can thus be scrolled individually.

or ^W  <F6> will move the cursor between the two trace windows.

or ^E  <F2> will move the cursor between the last edited trace window and the
Searching for and filtering trace data

The **Search** command offers powerful search and extract functions. **Search** locates 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 pattern are displayed.

**F6**

Type `<F6>` [or `^w`] to move the cursor to the window VME #1:

---

**Search** `<S>` to open the search menu:

**Edit Pattern**  
**Next Match** F3  
**Extract**

---

**Edit Pattern** is the only command available at this point. The other commands, **Next Match** and **Extract** can only be executed when search and extract patterns are defined.
5 Operation

<i> it </i> to open the pattern editor:

Notice the window labels: “VME #1” and “Search/Extract in VME#1”. This means that the search and extract will take place in the window where the Search/Extract Patterns was executed.

The Search/Extract edit window supports a subset of the functionality in the Event editor with these exceptions: The names of the Search and Extract events cannot be changed. Neither of these events can be deleted. No new events can be inserted. Corresponding trace window will scroll too. Columns inserted or deleted will also be inserted or deleted in the corresponding trace window.

Change the Cycle Search event to

---

1 56 for details on Event editing.
After editing the Search pattern, type F3 to start searching from the current line trace line:

The first occurrence of a RMW cycle is found in trace line #3. Type <F3> again to search for the next occurrence of the pattern. If found, the cursor will be

Type <Y> to continue searching from the beginning of the trace buffer. Type <N> to abort the searching now.
To be able to extract trace data, or, in other words, display only the trace data

If not already there, place the cursor in the window. Then, execute the pattern to UBYTE:

Type $S$ to open the Search menu:

Type $E$ to start the Extract operation:

The extract process will take some time. Type $\langle Esc\rangle$ to abort if necessary. After a while, the extract itself is finished, but in order to display the time tags correctly the absolute time tags must be calculated:
Extracting, the trace screen will look something like this:

<table>
<thead>
<tr>
<th>Line</th>
<th>Time</th>
<th>Dgl.</th>
<th>AM Address</th>
<th>Data</th>
<th>Size</th>
<th>Cycle</th>
<th>Stat</th>
<th>IRQ7:1&lt;&lt;Tack</th>
<th>Fail</th>
<th>In3:0</th>
<th>P2trg</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.88us</td>
<td>80</td>
<td>00100082</td>
<td>......</td>
<td>11</td>
<td>UBYTE</td>
<td>RD</td>
<td>OK</td>
<td>......</td>
<td>......</td>
<td>......</td>
</tr>
<tr>
<td>3</td>
<td>171.76us</td>
<td>80</td>
<td>00100082</td>
<td>......</td>
<td>22</td>
<td>UBYTE</td>
<td>RD</td>
<td>OK</td>
<td>......</td>
<td>......</td>
<td>......</td>
</tr>
<tr>
<td>4</td>
<td>171.28us</td>
<td>80</td>
<td>00100082</td>
<td>......</td>
<td>11</td>
<td>UBYTE</td>
<td>RD</td>
<td>OK</td>
<td>......</td>
<td>......</td>
<td>......</td>
</tr>
<tr>
<td>6</td>
<td>310ns</td>
<td>80</td>
<td>00100082</td>
<td>......</td>
<td>22</td>
<td>UBYTE</td>
<td>RD</td>
<td>OK</td>
<td>......</td>
<td>......</td>
<td>......</td>
</tr>
<tr>
<td>9</td>
<td>171.34us</td>
<td>80</td>
<td>00100082</td>
<td>......</td>
<td>11</td>
<td>UBYTE</td>
<td>RD</td>
<td>OK</td>
<td>......</td>
<td>......</td>
<td>......</td>
</tr>
<tr>
<td>9</td>
<td>330ns</td>
<td>80</td>
<td>00100082</td>
<td>......</td>
<td>22</td>
<td>UBYTE</td>
<td>RD</td>
<td>OK</td>
<td>......</td>
<td>......</td>
<td>......</td>
</tr>
</tbody>
</table>

Msg: 2114 The status line will show the number of matching trace lines.
5 Operation

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 are presented as ladders, 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.

Waveforms are used as the default presentation form when TIMING (asynchronous) sampling is used.

This example

To be able to follow this example, capture some general bus traffic with the following settings of the Tracer:

![Image of Tracer setup]

Change the Event VME0 to suit the traffic found in the system. Then execute the command _Trace/Run_ to capture some traffic.
5 Operation

Top line

The top line [in the Waveform window] contains, as fixed fields, the contents of the Address, the Data and the AM code of the sample [i.e. trace line] under the cursor. Above the fixed fields, at the window border, the current time per division is displayed.

Signal list

The signal list shows the value of each signal under the cursor in addition the signal names.

← →

The <Left> and <Right> arrow keys move the timing cursor sideways to the previous / next step. One step, or division, is by default one sample, but can be changed by the means of the Format/Time Per Division command.

<PgUp> <PgDn>

Move the timing cursor one page left or right.

↑ ↓

The <Up> and <Down> arrow keys move the signal cursor to the previous or next signal name.
Type <Right> a few times to move the timing cursor to the right:

X-T: 100ns The X-T field will always display the distance in time between the timing cursor and the Trigger line [TRIG].

AS* The current value, i.e. the value of the signal under the cursor is always displayed beneath the signal name.
Hiding and Inserting signals

↓.. Type <Down> a few times, or until then cursor is on the AM signal:

![Diagram showing signal hiding and inserting process]

<Del> Type <Del> to hide the AM signal. Repeat <Del> until the AM, the Address and the Data signal are hidden.

<Ins> Use <Ins> to insert signals above the current signal.
5 Operation

Edge Jumping and Timing Markers

↓..

Scroll downwards until DTACK* is highlighted:

F3

Type <F3> once to move the cursor to the previous edge, i.e. where the current signal changes polarity:

Markers/Set Marker Y

Execute this command to set a marker at the cursor position.
Move the cursor to the signal AS*. Then type <F3> to move to the cursor to the previous edge:

\[ X-Y: -140\text{ns} \]

The time from the falling AS*, i.e. the master assert the address, to the DTACK* goes low, i.e. when the slave has responded with data, is 140 ns.

F4 Moves the cursor to the next edge. By default, any edge will be the target for the <F3> and <F4> keys. The command Jump/Edge Options allows for other settings:

**Jump**

- First Line
- Last Line
- Trigger Line
- Marker Y
- Marker Z
- Line Number ...

**Edge Options**

- * Any Edge
- Falling Edge
- Rising Edge
5 Operation

Time per division
VME #1 -- 20ns/div

The default time per division is 20ns. On the VBT-325, this corresponds to one sample, or trace line, per division when sampling at 50MHz.

Format/Time Div

Allows for changing the time per division:

<table>
<thead>
<tr>
<th>Time/Div</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time per division: 20ns</td>
</tr>
<tr>
<td>&lt; Ok &gt; &lt; Cancel &gt;</td>
</tr>
</tbody>
</table>

Type <Down> to open a list of selectable settings. On the VBT-325, the selectable time per division goes from 20ns to 2.56\(\mu\)s. High settings will slow down scrolling and other window refresh due to the large amount of trace lines read when the window is refreshed.

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\(^1\). You can use the PC Simulator to view the trace, or to make a printout to a file or a printer. The trace file can also be loaded back into the VBT-325. Partial traces can also be dumped. 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. It 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.

STIM200-PB

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

---

\(^1\) See page 215 for the Trace file format.
Statistics

The VBT-325 provides powerful statistical measurements of target bus activity:

- 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 method. It 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 bus 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 smaller part of all bus activity is covered, so in order to give a true picture of the behavior 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.
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 below.

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, 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 5% as the max. reading (by the commands 'Options/Maximum scale').
Normally, the histogram is shown as a horizontal bar (line) where the end point represents the last value read from the statistics counters. A statistics session indication of the lowest and highest values recorded, and the average of all the counts. The command _options/B _Show of minimum, maximum and average markers on the histogram bars. The current (last read) value is indicated as a bullet. Some terminals displays this as a solid

<-

- a Average marker

> Maximum marker

•– Current value marker

*Note:* to current value.

It is possible to reset the marker positions by the commands _Options/_ar Markers/_eset_, so that only subsequent counter readouts affect the marker

**Time History Curves**

The Time History Diagram will show the variations of the bus signals with 'Options/Time History Curves', and set the 'Options/Graph Display Options'.

<table>
<thead>
<tr>
<th>VMETRO VBT-325C</th>
<th>VMETR Bus Statistics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Session</td>
<td>Function Options Target Utilities Quit Help</td>
</tr>
<tr>
<td>80%</td>
<td>80%</td>
</tr>
<tr>
<td>40%</td>
<td>40%</td>
</tr>
<tr>
<td>VME8</td>
<td>VME1</td>
</tr>
<tr>
<td>80%</td>
<td>80%</td>
</tr>
<tr>
<td>40%</td>
<td>40%</td>
</tr>
<tr>
<td>VME3</td>
<td>Level 0</td>
</tr>
<tr>
<td>80%</td>
<td>80%</td>
</tr>
<tr>
<td>40%</td>
<td>40%</td>
</tr>
<tr>
<td>Level 2</td>
<td>Level 3</td>
</tr>
</tbody>
</table>

Count All Samples in State Update Every 1 Seconds Reset Mode

Ok. <F2-Window> <F6-Nxt umd> | Run |
As for the signals and their explanation, see the two 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.
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.:

\[
\text{Displayed Value} = \left( \frac{\text{Counter reading}}{\text{Update rate}} \right) \times 100\%
\]

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

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.:

\[
\text{Displayed Value} = \left(\sum \text{Counter readings} / \text{Update rate} \times N\right) \times 100\%
\]

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 app. 225 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.

<table>
<thead>
<tr>
<th>VMETRO VBT-325C</th>
<th>VME Bus Statistics</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Session Function Options Target Utilities Quit Help</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Event No.</th>
<th>Count</th>
<th>0%</th>
<th>10%</th>
<th>20%</th>
<th>30%</th>
<th>40%</th>
<th>50%</th>
<th>60%</th>
<th>70%</th>
<th>80%</th>
<th>90%</th>
<th>100%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Words</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Longs</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Read</td>
<td>18278</td>
<td>56%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Write</td>
<td>14490</td>
<td>44%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bus Level</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>Unknown</td>
<td>32768</td>
</tr>
</tbody>
</table>

Total Number Of Samples: 32768

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

Bus Utilization
'Functions/Bus Utilization', performs its function using determined signal state parameters. Also, a fixed timing (asynchronous)

provide any signal state specifications prior to initiating a measurement, meaning that configuration and manipulation of this utility is accomplished exclusively

<table>
<thead>
<tr>
<th>Function</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>BBSY+</td>
<td>49722824</td>
</tr>
<tr>
<td>AS*</td>
<td>3248873</td>
</tr>
<tr>
<td>DATA</td>
<td>22362420</td>
</tr>
<tr>
<td>Total DTB</td>
<td>3596622</td>
</tr>
</tbody>
</table>

VMEbus Utilization measurements

Four histograms are provided in the VMEbus Bus Utilization function, each on the bus:

BBSY* signal is active,

AS* indicates percentage

includes total time spent on block (BLT/MBLT) transfers and RMW cycles, since AS* is active throughout these cycle types.

The logical or DS1*, DTACK*, BERR*, indicating percentage of

Total DTB The logical of AS* DS1*, DTACK*, BERR, indicating percentage
VSB Utilization measurements

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

<table>
<thead>
<tr>
<th>Session</th>
<th>Function</th>
<th>Options</th>
<th>Target</th>
<th>Utilities</th>
<th>Quit</th>
<th>Help</th>
</tr>
</thead>
</table>

**Total Number Of Samples**: 49776175

<table>
<thead>
<tr>
<th>Bus Util</th>
<th>Count</th>
<th>0%</th>
<th>10%</th>
<th>20%</th>
<th>30%</th>
<th>40%</th>
<th>50%</th>
<th>60%</th>
<th>70%</th>
<th>80%</th>
<th>90%</th>
<th>100%</th>
</tr>
</thead>
<tbody>
<tr>
<td>BUSY*</td>
<td>49776175</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ADDRESS</td>
<td>4265680</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DATA</td>
<td>6402978</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total DTB</td>
<td>13349283</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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
5 Operation

Bus Transfer Rate Histogram

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

when the collected data is being processed. You must select whether the counters should accumulate the numbers, or reset the numbers for each trace.

Statistics from the main window, and Function/Bus Transfer Rate.

<table>
<thead>
<tr>
<th>Bus Level Mtransfer/sec</th>
<th>0.0</th>
<th>0.2</th>
<th>0.4</th>
<th>0.6</th>
<th>0.8</th>
<th>1.0</th>
<th>1.2</th>
<th>1.4</th>
<th>1.6</th>
<th>1.8</th>
<th>2.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unknown</td>
<td>1.19</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>1.19</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bus Level Mbyte/sec</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unknown</td>
<td>3.99</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>3.99</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Immediate Start
Msg: Filtering trace buffer. 96% finished.

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 2 MXFers/s, and for bytes per second to 8 Mbytes/s:
6 COMMANDS REFERENCE

In this chapter you will find a complete reference of all the menu commands for the VBT-325 and the XMEM325-PB.

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. The figure below shows the entire command tree. The number after each entry refer to the page where the command is described.

<table>
<thead>
<tr>
<th>Trace</th>
<th>Edit</th>
<th>Target</th>
<th>Statistics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Run VME 122</td>
<td>Event Patterns 124</td>
<td>(See page 126)</td>
<td>(See page 143)</td>
</tr>
<tr>
<td>Run Multiple 122</td>
<td>Sequence 125</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Halt VME 124</td>
<td>Trigger Position</td>
<td>*Start of Trace 125</td>
<td></td>
</tr>
<tr>
<td>Halt All 124</td>
<td></td>
<td>25% of Trace</td>
<td></td>
</tr>
<tr>
<td>Show 124</td>
<td></td>
<td>Middle of Trace</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>75% of Trace</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>End of Trace</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Sampling Mode</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>*State 125</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>50 MHz Timing</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Options</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>State Options 125</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>50 Mhz Timing</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>25 MHz Timing</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>6.25 MHz Timing</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.56 MHz Timing</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>781.3 KHz Timing</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>390.6 KHz Timing</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>195.3 KHz Timing</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>97.7 KHz Timing</td>
<td></td>
</tr>
</tbody>
</table>

Setups | Utilities | Help 135 |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Initialize 128</td>
<td>Transparent Mode 132</td>
<td>FW Version 135</td>
</tr>
<tr>
<td>Store 128</td>
<td>Transparent Mode Options 132</td>
<td>ECO Level</td>
</tr>
<tr>
<td>Delete 128</td>
<td>Trigger Output Options 132</td>
<td>Set Time And Date</td>
</tr>
<tr>
<td></td>
<td>LED Display 133</td>
<td></td>
</tr>
<tr>
<td></td>
<td>*1 Default</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Serial Ports</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Terminal Port 134</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Host Port</td>
<td></td>
</tr>
<tr>
<td>Assert VMEbus SYSRES* 134</td>
<td>Selftest 134</td>
<td></td>
</tr>
<tr>
<td>Reset Analyzer 134</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Specials</td>
<td>FW Version 135</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ECO Level</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Set Time And Date</td>
<td></td>
</tr>
</tbody>
</table>
Trace

The Trace menu controls starting, stopping, and displaying the trace. Since the VBT-325 is a multiple analyzer product (except the VBT-325B without no piggyback analyzer), 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 menu: Trace/Run <Current target>, and Trace/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 on page 126.

- Run Multiple

This command exist on the VBT-325C, or on the VBT-325B when a piggyback tracer, such as the S/TIM200-PB, the TIMBAT-PB, or the XMEM325-PB is installed.

Execute this command to start simultaneous operation of all the analyzers present in the actual hardware configuration:

```
Trace/Run Multiple
```
- Sampling Status

A 'Sampling Status' box appears on the screen when any 'Trace/Run..' commands are 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.

### Annunciators

Note also that the same information is present in the annunciators in the lower right corner on the screen. The annunciators show the status of up to four analyzers, in the following sequence:

| VME | P2 | TIM200 or XMEM_VME | XMEM_P2 |

Their status may be one of the following:

<table>
<thead>
<tr>
<th>Annunciator text</th>
<th>Sampling status</th>
</tr>
</thead>
<tbody>
<tr>
<td>WfT</td>
<td>Waiting for Trigger</td>
</tr>
<tr>
<td>Trg</td>
<td>Triggered</td>
</tr>
<tr>
<td>Tco</td>
<td>Trace completed</td>
</tr>
</tbody>
</table>
- Halt

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

\texttt{Trace/Halt}

- Halt All

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

\texttt{<Halt All>}

button in the Sampling Status box, or by the means of the command:

\texttt{Trace/Halt All}

- Show

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

\texttt{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.

\textbf{Show trace from another target}

In order to see the trace from another target bus, one must first use the \texttt{Target} command to go to the desired target, and then use \texttt{Trace/Show}.

\textbf{Edit}

The \texttt{Edit} menu 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.

*Single event* The command is valid in Single Event mode only (see page 68). In Sequencer mode, the Trigger position is selected directly in the sequencer program.

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.

*Single event* The command is valid in Single Event mode only (see page 68). In Sequencer mode, the Sampling mode position is selected directly in the sequencer program.

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 four shared signals in the trace:

<table>
<thead>
<tr>
<th>State Sampling Options</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="State Sampling Options" /></td>
</tr>
</tbody>
</table>

- Time tag options

- Limited time tag range (50ns - 6 min 8 sec) (In3:0 available in Trace)
- Extended time tag range (50ns - 1 hr 38 min) (In3:0 NOT available in Trace)

< Ok > < Cancel >
**Shared bits**

The **time-tag counter** can operate in either 16-bit or 12-bit modes. 16-bit mode gives the best resolution and the largest measurable time interval. Four bits in the time tag counter on the VME part of the VBT-325 are shared with the external inputs, In0...In3. Similarly, VBTvtrg* and ExtIn0, are shared with the time-tag bits on the VME part of the XMEM325-PB.

[X] Limited time tag range  
(... available in trace)

Set 12-bit tag. Limits the maximum range of the time tag to 6 min 8 sec.

[X] Extended time tag range  
(... NOT available in trace)

Set 16-bit tag. The tag counter operates at its highest resolution, and with its largest measurable time interval. Time intervals up to 1 hour and 38 minutes can be measured. The indicated signal names are not sampled, and thus not presented in the Trace screen. The signals are still available in the Event patterns. See also page 38.

**Target**

The VBT-325C allows simultaneous analysis of both the VMEbus and a P2 bus like VSB, SCSI or VXI. Use the **Target** command to switch control of the user interface between the different analyzers present in the VBT-325 and the installed piggyback. After execution, the **current target** will have its Setup screen, containing the Event patterns and the Sequencer program, displayed.

Execute the command **Trace/Show** to display the Trace screen of the current target. During and after sampling, the sampling status of the different targets are shown directly in the **Target** menu. The current target is shown in low intensity, since it is already selected.

- **VBT-325B**

VBT-325B supports VMEbus only. The **Target** command is not present unless a analyzer piggyback, such as the S/TIM200-PB, the TIMBAT-PB, or the XMEM325-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. User defined P2 is described in a separate section, starting at page 198.
- **XMEM VME**

  This entry is available when the XMEM325-PB is installed. The primary target bus for the XMEM325-PB is VMEbus.

- **XMEM P2**

  This entry is available when the XMEM325-PB/C is installed. The P2 bus on XMEM325-PB/C can be selected to be VSB, SCSI, and VXI_P2ac. Only the target currently selected is shown in the Target menu, and by default, this is VSB.

- **TIM200 bus**

  When the TIMBAT-PB or S/TIM200-PB is installed, it can be selected in the Target command. If the P2 option is purchased, it can be enabled by the Reconfigure command as described below.

- **Reconfigure**

  The default setting on the P2 bus of the VBT-325C is VSB. 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 TIM200-PB and the XMEM325-PB.

  Reconfiguring the P2 and XMEM_P2 busses normally requires some jumpers to be moved, so a warning about this is given when closing this command if changes have been made.

  The selection of P2 and XMEM_P2 must match. Thus, if SCSI is selected, XMEM_P2 must be selected as XMEM_SCSI.

- **Statistics**

  The Statistics command activates the Statistics screen, containing the statistics histograms and a special menu bar. Refer to page 143.
6 Commands reference

**Setups**

The *Setups* command is used to initialize, store, delete or retrieve user-defined setups of event patterns, the Sequencer program, as well as the trigger position and sampling mode, and all Statistics options.

The number of storable setups vary, but 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. The event patterns, Sequencer, trigger position, sampling mode, and the Statistics options are returned to the default conditions, as 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 setup cannot be deleted.

- **Delete**

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

- **1 Default**

  The first 9 stored setups are represented as the short-cuts <1>..<9>. Type the corresponding digit to retrieve the wanted setup.

  If more than 9 setups are defined, an *All setups...* will appear at the end of the list. Executing this command will open a list with all defined setups.
Load from PC/Host...

Note  The following section apply to firmware version 5.10 or newer only. See page 168 for HW requirements.

A setup can be loaded into the tracer again at any time. Remember if the user defined signal fields have been edited since the setup was saved, all other setups in the tracer will be deleted. To load a setup, select the menu command Setup/Load From PC/Host.... The dialog box in the figure below appears on the screen.

![Load Setup from PC/Host dialog box]

When `<OK>` is selected the following message box appears on the screen:

![Start XMODEM Transmit on PC now]

The PC/Host should be setup to send the data using the X-modem protocol.

For example for the Windows Terminal program, select Transfer/Send Binary File. The Settings/Binary Transfers must first be set to `Xmodem/CRC`. When the dump is finished the screen is redrawn with the loaded setup active.

VT100.EXE  Refer to page 208 for how to use the VMETRO Terminal Emulator for receiving and sending files.

Possible warnings  Depending on the state of the tracer, different warning dialog boxes must be answered. If the active setup has been changed the following dialog box will appear.

![The setup "Test" has been changed.
Keep the changes?]

- `<OK>`  
- `<Cancel>`
If the current signal field definitions have been changed, the dialog box below appears. Notice that all setups in the tracer including the signal field definitions will be lost. To save everything the Setup/Dump To PC/Host... command described in the previous section should be used.

If a setup with the same name as the one loaded exists, the dialog below appears on the screen. Answer <No> and use the Setup/Store command to store the old setup with a new name. Remember the setup you want to store must first be made active. Alternatively the old setup can be dumped to a file with the Setup/Dump To PC/Host... command as described in the previous section. If you do not want to save the old setup, just answer <Yes>.

- Dump to PC/Host...

**Note**  
The following section apply to firmware version 5.10 or newer only. See page 168 for HW requirements.

The Dump Setup to PC/Host command stores the same way as the Setup/Store... command. Instead of storing it in non-volatile memory, it uses the XMODEM protocol to dump it to a file on the PC/host. The dump setup to PC/host is a feature that is available for all user defined P2 targets and all other targets except the STIM.

The following parameters belong to a setup:

- Signal field definitions. These must be the same for all setups loaded into the tracer simultaneously. They are only stored for user defined P2, since they are fixed for other targets. For the STIM, only the signal field definitions are stored. This means the Setup command is not available for other STIM targets than user defined P2.

- Events with the inserted list of signal fields and their values (one for state and one for timing analysis).
• The Sequencer program.
• The selected sampling frequency.
• Trigger input options for the TIMBAT
• The statistics option parameters, one for each statistics function.

To dump the setup, select the menu command
'Setup/Dump To PC/Host...' (For STIM user P2 only the two last menu items are available). The dialog box in figure below appears on the screen.

The 'Setup name' string is the name of the setup, which will be used when it is loaded again later. The 'Comment' is a string used in the pull down menu for a better description of the setup. These parameters are exactly the same as for 'Setup/Store...'. When you select '<OK>' the following message box appears on the screen.

The PC/Host should be setup to receive the data using the XMODEM protocol. For example from the Windows Terminal program, select Transfer/Receive Binary File. The Settings/Binary Transfers... must first be set to Xmodem/CRC. When the dump is finished the screen is redrawn.

VT100.EXE Refer to page 208 for how to use the VMETRO Terminal Emulator for receiving and sending files.
Utilities

Under the 'Utilities' menu, 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

<table>
<thead>
<tr>
<th>Trigger Output Options</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Mode</strong></td>
</tr>
<tr>
<td>[ ] Level on Trigger</td>
</tr>
<tr>
<td>[ ] Follow Trigger</td>
</tr>
<tr>
<td>[ ] Follow Store</td>
</tr>
<tr>
<td><strong>Polarity</strong></td>
</tr>
<tr>
<td>[ ] Active high</td>
</tr>
<tr>
<td>[ ] Active low</td>
</tr>
</tbody>
</table>

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

**Follow Store** 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 **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'.

Trigger output: 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.
- LED Display

Specifies the default text on the front panel display:

<table>
<thead>
<tr>
<th>LED Display options</th>
</tr>
</thead>
<tbody>
<tr>
<td>() String :</td>
</tr>
<tr>
<td>() Time of day</td>
</tr>
<tr>
<td>(x) +5V readout</td>
</tr>
<tr>
<td>() +12V readout</td>
</tr>
<tr>
<td>() -12V readout</td>
</tr>
<tr>
<td>() Temperature readout</td>
</tr>
</tbody>
</table>

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] +5V  

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.

[X] Temp  

Select Temperature, then install the optional temperature probe¹ the In1/T input on the front panel to display the temperature on the front panel LED display.

Temp range  

The valid range of the temperature reading is 0-80°C / 32-176°F.

V+T/V jumper  

The jumper with the yellow label "V+T and V" must be in the V+T position to get correct temperature measurements.

Boards with ECO level lower than B8 does not have the yellow "V+T and V" labeled jumper. On boards with ECO level lower than B8, the temperature probe must not be present when doing voltage measurements.

Boards with ECO level C0 and higher does not have the V+T and V jumper.

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).

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

The baud rate etc. of the two serial ports can be defined independently of each other by the 'Serial Ports' command. The default settings are as shown to the left. If one of the ports is set to 38K4, the other must either be set to the same, or to 9600 baud.

<table>
<thead>
<tr>
<th>Terminal port</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baud rate : Auto</td>
</tr>
<tr>
<td>Bit per char : 8</td>
</tr>
<tr>
<td>Parity : None</td>
</tr>
<tr>
<td>Stop bits : 1</td>
</tr>
</tbody>
</table>

- **Assert VMEbus SYSRESET**

This command will assert, i.e. drive the VMEbus SYSRES* signal low for 250 mS.

- **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:**
The Selftest command starts an extensive test of most hardware modules on the selected analyzer(s). Running the selftest for the VBT-325C will take 1 minute and 20 seconds.

**Example**
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 command 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 lost due to back-up battery failure.

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

The main menu in the Trace screen contains a command bar with command
groups for displaying the current trace buffer. The figure below shows the trace
screen command tree. The number after each entry refer to the page where the
command is described:

<table>
<thead>
<tr>
<th>Trace</th>
<th>Search</th>
<th>Jump</th>
<th>Count 140</th>
<th>Window</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dump to PC/Host 137</td>
<td>Edit Pattern 100</td>
<td>First Line 139</td>
<td></td>
<td>New 142</td>
</tr>
<tr>
<td>Load From PC/Host 138</td>
<td>Next Match F3 101</td>
<td>Last Line 139</td>
<td></td>
<td>Close 142</td>
</tr>
<tr>
<td>Print 138</td>
<td>Extract 102</td>
<td>Trigger Line 140</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Print To File 229</td>
<td>Close pattern Window</td>
<td>Marker Y 140</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Marker Z 140</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Line Number 140</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Edge Options</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>*Any Edge 140</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Falling Edge</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Rising Edge</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Trace

The 'Trace' command in the Trace Display screen differs from the one in the
Setup screen:

```
Trace   Search   Jump
Dump To PC/Host ... Edit Pattern 100
Load From PC/Host ... Next Match F3 101
Print ... Extract 102
```

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:

![Screenshot of Dump to PC/Host dialog box]

**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.

**To line:** The trace buffer dump will end after this line is dumped. Both numbers are decimal trace line numbers.

**Packing** When dumping a Timing-mode trace buffer, two extra options are presented:

- [ ] Use no packing
- [X] Use run length packing

Using run length packing with trace buffer lines as basis, will in most cases reduce the amount of transferred data considerably. State traces are always dumped with no packing because all trace lines then are different. The default value for timing traces is to use run length packing, but if very few lines are equal, or the SYSCLK signal for VME traces must be saved, this feature should be turned off.

When you select < Ok > the following message box below will be displayed:

![Screenshot of Start XMODEM Receive on PC now button]

**XMODEM** Refer to the documentation of your terminal emulator software for how to set it up to receive [download] a file utilizing the XMODEM CRC protocol.

**VT100.EXE** The VMETRO VT100 Terminal Emulator is described on page 208.
6 Commands reference

- Load from PC

The trace files may be loaded back into the VBT-325 with the 'Load from PC' sub-command:

```
Load from PC/Host

Load the trace buffer from a trace file on a host.

- To proceed you must be using a terminal emulator on a PC or
  or any other host supporting the XMODEM file transfer protocol.
  Example: PC with VMETRO's VT100.EXE, WINDOWS Terminal etc.

Before starting, make sure the current 'target' corresponds to the
  target of the trace file. If not, cancel now, change target and
  reactivate this command.

< Ok >  < Cancel >
```

No options

There are no options to this command. When "< Ok >" is selected, the
following message will be displayed on the screen:

```
Start XMODEM Transmit on PC now
```

XMODEM

Refer to the documentation of your terminal emulator software for how to set it
up to receive [download] a file utilizing the XMODEM CRC protocol.

VT100.EXE

The VMETRO VT100 Terminal Emulator is described on page 208.

- 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:

```
Print Trace

First Line : 0
Last Line : 61
Number of Lines : 62

- Print options

Send output to : (X) Host port
               (_1 Term port -> Screen
               (_) Term port -> Screen + Printer

Waveform chars : (X) Standard ASCII
                (_) IBM extended ASCII

Lines/page : 62  [] Manual feed

<Start print>  < Cancel >
```

Lines to print

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.
(X) Host Sends the output to the host port.

(X) Term Sends the output to the terminal port.

(x) Term port -> Screen + Printer

Sends the output to the printer which is connected to your terminal. The control character sequences for relay printing [as defined for your terminal type] will be used. If the Windows TERMINAL.EXE is used, the printout will be sent to the default Windows printer.

Waveform chars The following two options can be specified when printing waveforms:

(x) Standard ASCII

The default is to print the waveforms using standard ASCII characters, as hyphen ‘-’, underscore ‘_’, and vertical bar ‘|’.

( ) IBM Extended ASCII

Select this option if the connected printer supports IBM Extended ASCII, IBM PC code page 437. If selected, semigraphic characters will be used for the printout. Most PC printers supports this option if properly set up.

Note: In the event of lost characters and/or garbled printout, try an lower baud rate. Also, make sure that the serial printer supports the XON/XOFF protocol.

Search

Search for a given pattern, or display only trace lines that match a given pattern. See page 99.

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).

- **Marker Y**
  Jumps to the line where the marker Y is placed. This command is disabled when no marker Y exist.

- **Marker Z**
  Jumps to the line where the marker Z is placed. This command is disabled when no marker Z exist.

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

- **Edge Options**
  Sets the behavior of the <F4> and <F3> keys in Waveform windows.
  The following options exist:
  
  *Any edge*  
  The default selection. Look for both falling and rising edges.
  
  *Falling edge*  
  Look for the previous/next falling edge.
  
  *Rising Edge*  
  Look for the previous/next rising edge.

**Count**

Counts Mbytes per sec. 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):

![Count Data Bytes](image)

Change the line numbers and select "< Ok >" to start counting. After a few seconds, the following data will be displayed:
Select <Continue> to go back for specifying a new area to be counted, or type <Esc> twice to go back to the menu.

Format

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

<table>
<thead>
<tr>
<th>Format</th>
<th>Markers</th>
<th>Window</th>
<th>Quit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time/Div</td>
<td>...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Absolute Time Tags</td>
<td>Relative Time Tags</td>
<td></td>
<td></td>
</tr>
<tr>
<td>* Decoding And Formatting</td>
<td>...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Formatting Template</td>
<td>...</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Time/Div

In Waveform mode, the 'Time/Div' command can be used to "zoom" in or out in the waveform display. This command will not be available in State mode.

- 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. A “how-to-use” description is found on page 93.
- Formatting Template

The 'Trace signals' sub-command selects between 'Grouped' or 'Single' i.e. individual control signals. (VMEbus only). A “ho-to-use” description is found on page 92.

ASCII display

It is possible to display VMEbus data as ASCII text in by inserting the column DataASCII. See page 158.

Markers

Markers or bookmarks are used as target for Jump’ing, or for measuring time in Waveform windows. See page 108 for details.

Window

The Window command allows another window to be added to the screen, either as a trace list or waveform. A “how-to-use” description is found on page 97.

- New

Create a new window:

```
New window type

(8) Alphanumeric
(7) Waveform

<Ok> <Cancel>
```

Select wanted type and type <Cr>. 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

Closes the current window. 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.

F6 or ^W

Move the cursor to the appropriate window before closing. The active window will have its frame highlighted.

Quit

'Quit' brings back the Setup screen for the selected target.

Help

Help brings up the general Help utility, describing general user-interface mechanisms etc.
Statistics menu

The menu bar in the Statistics screen contains commands for controlling all the Statistics options and capabilities. The figure below shows the entire Statistics command tree. The number after each entry refer to the page where the command is described:

<table>
<thead>
<tr>
<th>Session</th>
<th>Function</th>
<th>Options</th>
<th>Target</th>
<th>Utilities</th>
<th>Quit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Run 143</td>
<td>*Event Counting 144</td>
<td>Standard Histograms 145</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Continue 143</td>
<td>Bus Utilization 144</td>
<td>Time History Curves</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Halt 144</td>
<td>Bus Transfer rate 145</td>
<td>Count Options 147</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>*Immediate Start 144</td>
<td></td>
<td>Bar Markers</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Start on Trigger 144</td>
<td></td>
<td>Show 145</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>History Curve Options 146</td>
<td></td>
<td>Reset</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Select Events 148</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sampling Mode</td>
<td>*State 149</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>50 MHz Timing Options</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>50 Mhz Timing</td>
<td>50 Mhz Timing</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>25 MHz Timing</td>
<td>25 MHz Timing</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6.25 MHz Timing</td>
<td>6.25 MHz Timing</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.56 MHz Timing</td>
<td>1.56 MHz Timing</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>781.3 KHz Timing</td>
<td>781.3 KHz Timing</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>390.6 KHz Timing</td>
<td>390.6 KHz Timing</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>195.3 KHz Timing</td>
<td>195.3 KHz Timing</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>97.7 KHz Timing</td>
<td>97.7 KHz Timing</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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 commands for operating the statistics:

<table>
<thead>
<tr>
<th>Session</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Run</td>
<td></td>
</tr>
<tr>
<td>Continue</td>
<td></td>
</tr>
<tr>
<td>Halt</td>
<td></td>
</tr>
<tr>
<td>* Immediate Start</td>
<td></td>
</tr>
<tr>
<td>Start On Trigger</td>
<td></td>
</tr>
</tbody>
</table>

- Run

_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 options set by the command _Options/Count Options._

- Continue

_Continue_ resumes counting from the values reached the last time the statistics was stopped with _Halt._
6 Commands reference

- Halt

Halt stops the statistics session and freezes the histogram screen.

- Immediate Start

This command is inactive for all functions except for the, trace based, Bus Transfer Rate. The default selection, Immediate Start, causes the statistics to start counting immediately.

- Start On Trigger

This command is inactive for all functions except for the, trace based, Bus Transfer Rate. Selecting “Start On Trigger” causes the Bus Transfer Rate statistics to wait for the trigger defined in the user Sequencer program before it starts counting. The user Sequencer program must provide Trigger at Start of Trace, and Sampling Mode STATE. The Sequencer program should not contain any Halt statement.

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:

<table>
<thead>
<tr>
<th>Function</th>
<th>Options: I</th>
</tr>
</thead>
<tbody>
<tr>
<td>Event Counting</td>
<td></td>
</tr>
<tr>
<td>Bus Utilization</td>
<td></td>
</tr>
<tr>
<td>Bus Transfer Rate</td>
<td></td>
</tr>
</tbody>
</table>

- 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*)

Bus Transfer rate

The `Bus Transfer Rate` sub-command invokes the preconfigured statistics function that provides trace-based histograms of the VMEbus transfer rate in *bytes* and *cycles* per second.

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 the selection of `Standard Histograms` or `Time History Curves`:

```
Options: Utilities Quit Help
```

- **Standard Histograms**

  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.

  To make a bar marker visible, perform the sub-commands `Bar Markers/Show`:

```
Show Markers

[ ] Maximum -->
[ ] Average ---
[ ] Minimum <--
< Ok > <Cancel>
```
Select the desired marker(s) in the dialog box by using the <Space> -bar or by typing <M> for Maximum, <A> for Average and/or <I> for Minimum.

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

**Reset**

Execute Bar Markers/Reset to reset the recorded values for the selected marker(s), so that only subsequent count values will be taken into account when displaying new bar markers:

<table>
<thead>
<tr>
<th>Reset Marker Readings</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Maximum</td>
</tr>
<tr>
<td>2 Average</td>
</tr>
<tr>
<td>3 Minimum</td>
</tr>
<tr>
<td>(X) Reset Now</td>
</tr>
<tr>
<td>( ) Autoreset Every _ Min.</td>
</tr>
<tr>
<td>&lt; Ok &gt; &lt; Cancel &gt;</td>
</tr>
</tbody>
</table>

(X) Reset Now  Cause a reset of the markers now.

( ) Autoreset  Enable for automatic reset every specified period. The default period is one minute. This can be adjusted within the range 1..60 minutes.

- **History Curve Options**

The History Curve 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:

<table>
<thead>
<tr>
<th>History Curve Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>Screen Split</td>
</tr>
<tr>
<td>2 Parts Horizontally</td>
</tr>
<tr>
<td>3 Parts Vertically</td>
</tr>
<tr>
<td>Graph Drawing</td>
</tr>
<tr>
<td>(Y) Retrace Graph</td>
</tr>
<tr>
<td>( ) Scroll Graph</td>
</tr>
<tr>
<td>&lt; Ok &gt; &lt; Cancel &gt;</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Histogram Scale</th>
<th>Histogram Scale</th>
</tr>
</thead>
<tbody>
<tr>
<td>(X) 100 % ( ) 40 %</td>
<td>( ) 20 MXfer/s ( ) 160 Mbytes/s</td>
</tr>
<tr>
<td>( ) 80 % ( ) 30 %</td>
<td>( ) 10 MXfer/s ( ) 80 Mbytes/s</td>
</tr>
<tr>
<td>( ) 70 % ( ) 20 %</td>
<td>( ) 5 MXfer/s ( ) 40 Mbytes/s</td>
</tr>
<tr>
<td>( ) 60 % ( ) 10 %</td>
<td>( ) 2 MXfer/s ( ) 16 Mbytes/s</td>
</tr>
<tr>
<td>( ) 50 % ( ) 5 %</td>
<td>( ) 1 MXfer/s ( ) 8 Mbytes/s</td>
</tr>
</tbody>
</table>

< Ok > <Cancel>

- Count Options

To optimize a statistics session to the actual system behavior, there are several Count Options that can be selected:

<table>
<thead>
<tr>
<th>Count Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>Qualifier</td>
</tr>
<tr>
<td>(X) Count All Samples</td>
</tr>
<tr>
<td>( ) Count Valid Samples Only</td>
</tr>
<tr>
<td>Update Every</td>
</tr>
<tr>
<td>(X) Time Interval Of 2 Seconds</td>
</tr>
<tr>
<td>( ) Sample Count Of 64K Samples</td>
</tr>
<tr>
<td>Mode</td>
</tr>
<tr>
<td>( ) Accumulate</td>
</tr>
<tr>
<td>(X) Reset</td>
</tr>
</tbody>
</table>

< Ok > <Cancel>

The selected Count Options are always shown at the status line below the histograms in the Statistics screen.

*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
If the screen update parameter is set too short, it will be increased automatically to a rate where the screen will be refreshed completely between each update. When the sampling stops, the user will be asked if the adjusted update rate should be kept for subsequent sessions.

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 start 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. 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:

```
Event Selection
Event 0: Event 1: Event 2: Event 3:
WHE0 WHE1 WHE2 WHE3

< Ok > <Cancel>
```

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.
- Sampling Mode

Use this command to change the sampling mode for the Event Counting statistics:

![Sampling Mode screenshot]

See page 69 for about how to use this command.

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. Refer to the section Jumper Settings at page 178.

Utilities

The Utilities menu in the Statistics screen duplicates the capabilities found under Utilities in the Setup screen. Refer to page 132.

Quit

Quit brings back the Setup screen for the selected target

Help

Help brings up the general Help utility, describing general user-interface mechanisms etc.
This chapter gives you a complete reference to all the signals used by the VBT-325 and the XMEM325-PB. Some of the fields contain bits, which values only can be set by selecting predefined symbols. This category of bits are listed within square brackets, as in the example below.

Size [RmwBlk, Vme64*, A01] DS1*, DS0*, LWORD*.

### VMEbus Signal Fields

**VBT-325 and XMEM325**

This chapter gives you a complete reference to all the signals used by the VME targets on VBT-325 and the XMEM325-PB Piggyback Module.

#### Default Signal Fields in State Mode

**BgL**

BG(3:0)*, internally latched.

This field contains bus grant information latched on the falling edge of BBSY*.

<table>
<thead>
<tr>
<th>Predefined symbol</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>---0</td>
<td>BG0* active.</td>
</tr>
<tr>
<td>--1-</td>
<td>BG1* active.</td>
</tr>
<tr>
<td>-2--</td>
<td>BG2* active.</td>
</tr>
<tr>
<td>3---</td>
<td>BG3* active.</td>
</tr>
<tr>
<td>----</td>
<td>No BG* active as seen from the tracer.</td>
</tr>
</tbody>
</table>
AM

AM (5:0)

The Address Modifier bits:

<table>
<thead>
<tr>
<th>Predefined symbol</th>
<th>AM (5:0)</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>SBLTE</td>
<td>07</td>
<td>Long supervisory block transfer.</td>
</tr>
<tr>
<td>SBLTL</td>
<td>06</td>
<td>Long supervisory program access.</td>
</tr>
<tr>
<td>SU64B</td>
<td>38</td>
<td>Standard non-privileged 64-bit block transfer.</td>
</tr>
<tr>
<td>SS64B</td>
<td>3C</td>
<td>Standard supervisory 64-bit block transfer.</td>
</tr>
<tr>
<td>LU64B</td>
<td>00</td>
<td>Long non-privileged 64-bit block transfer.</td>
</tr>
<tr>
<td>LS64B</td>
<td>04</td>
<td>Long supervisory 64-bit block transfer.</td>
</tr>
<tr>
<td>EU64B</td>
<td>08</td>
<td>Extended non-privileged 64-bit block transfer.</td>
</tr>
<tr>
<td>ES64B</td>
<td>0C</td>
<td>Extended supervisory 64-bit block transfer.</td>
</tr>
<tr>
<td>SUA</td>
<td>29</td>
<td>Short non-privileged access.</td>
</tr>
<tr>
<td>SSA</td>
<td>2D</td>
<td>Short supervisory access.</td>
</tr>
<tr>
<td>SUD</td>
<td>39</td>
<td>Standard non-privileged data access.</td>
</tr>
<tr>
<td>SUP</td>
<td>3A</td>
<td>Standard non-privileged program access.</td>
</tr>
<tr>
<td>SUB</td>
<td>3B</td>
<td>Standard non-privileged block transfer.</td>
</tr>
<tr>
<td>SSD</td>
<td>3D</td>
<td>Standard supervisory data access.</td>
</tr>
<tr>
<td>SSP</td>
<td>3E</td>
<td>Standard supervisory program access.</td>
</tr>
<tr>
<td>SSB</td>
<td>3F</td>
<td>Standard supervisory block transfer.</td>
</tr>
<tr>
<td>EUD</td>
<td>09</td>
<td>Extended non-privileged data access.</td>
</tr>
<tr>
<td>EUP</td>
<td>0A</td>
<td>Extended non-privileged program access.</td>
</tr>
<tr>
<td>EUB</td>
<td>0B</td>
<td>Extended non-privileged block transfer.</td>
</tr>
<tr>
<td>ESD</td>
<td>0D</td>
<td>Extended supervisory data access.</td>
</tr>
<tr>
<td>ESP</td>
<td>0E</td>
<td>Extended supervisory program access.</td>
</tr>
<tr>
<td>ESB</td>
<td>0F</td>
<td>Extended supervisory block transfer.</td>
</tr>
</tbody>
</table>

Address

A (31:1), plus DS1* acting as virtual A00. Address ranges and field negation is supported. A64 cycles, including A64 range is also supported.

Data

D (31:0). Data range and negation is supported. D64 cycles may be specified, but no D64 range.
Size

[RmwBlk, Vme64*, A01] DS1*, DS0*, LWORD*.

These control signals are combined into the field 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. The internally generated bit RmwBlk is also included in this group, set during Read-Modify-Write and Block cycles (in second cycle).

Vme64*

This, internally generated, signal is used to identify cycles with VME64 AM-codes, i.e. \( \text{AM} = 00, 01, 03, 06, 07, 08, 0C, 38, \text{or } 3C. \)

<table>
<thead>
<tr>
<th>Predefined symbol</th>
<th>RmwBlk</th>
<th>Vme64*</th>
<th>A01</th>
<th>DS1*</th>
<th>DS0*</th>
<th>LWORD*</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNAL3L</td>
<td>X</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>UNAL3H</td>
<td>X</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>UNAL2</td>
<td>X</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>UBYTE</td>
<td>X</td>
<td>1</td>
<td>X</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>LBYTE</td>
<td>X</td>
<td>1</td>
<td>X</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>WORD</td>
<td>X</td>
<td>1</td>
<td>X</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>LONG</td>
<td>X</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>A-MBLT</td>
<td>0</td>
<td>0</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>VME64</td>
</tr>
<tr>
<td>D64</td>
<td>1</td>
<td>0</td>
<td>X</td>
<td>0</td>
<td>0</td>
<td>X</td>
<td>VME64</td>
</tr>
</tbody>
</table>
The Cycle field contains the bus signals that define the current cycle type. The internally generated bits RmwBlk and Vme64* are also included in this group.

RmwBlk

Is asserted during Read-Modify-Write and Block cycles (in second cycle). Together with address modifier bit AM1, this field can be used directly to set mnemonics for Read, Write, RMW and Block cycles.

<table>
<thead>
<tr>
<th>Predefined symbol</th>
<th>AM1</th>
<th>Vme64*</th>
<th>RmwBlk</th>
<th>WR*</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>RD</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>1</td>
<td>Generic Read cycle.</td>
</tr>
<tr>
<td>WRI</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>0</td>
<td>Generic Write cycle.</td>
</tr>
<tr>
<td>RBLK</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>Block Read cycle.</td>
</tr>
<tr>
<td>WBLK</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>Block Write cycle.</td>
</tr>
<tr>
<td>RMBL</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>VME64 Block Read.</td>
</tr>
<tr>
<td>WMBL</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>VME64 Block Write.</td>
</tr>
<tr>
<td>RMW</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>Read-Modify-Write cycle.</td>
</tr>
</tbody>
</table>

The bus cycle status field:

<table>
<thead>
<tr>
<th>Predefined symbol</th>
<th>DTACK*</th>
<th>BERR*</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>OK</td>
<td>0</td>
<td>1</td>
<td>DTACK* asserted. Cycle ended Ok.</td>
</tr>
<tr>
<td>BERR</td>
<td>X</td>
<td>0</td>
<td>BERR* asserted by master.</td>
</tr>
<tr>
<td>Err!</td>
<td>0</td>
<td>0</td>
<td>Illegal combination of signals</td>
</tr>
<tr>
<td>Off</td>
<td>1</td>
<td>1</td>
<td>Neither DTACK* or BERR* asserted.</td>
</tr>
</tbody>
</table>

The IRQ field. Can be negated. No predefined symbols, but special formatting applies. All asserted, i.e. low IRQ* signals, are displayed with their IRQ number, all non-asserted, i.e. high values are displayed as “.”. For example, IRQ5* asserted, and IRQ1* as don’t care, and the rest non-asserted, i.e. high, will be displayed “...5...X”.

IRQ7:1*

IRQ(7:1)*
Interrupt acknowledge is defined as a numeric code on the signals A(3:1) and driving IACK* low.

<table>
<thead>
<tr>
<th>Predefined symbol</th>
<th>A03</th>
<th>A02</th>
<th>A01</th>
<th>IACK*</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>----</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>1</td>
<td>No Interrupt Acknowledge cycle.</td>
</tr>
<tr>
<td>IACKx</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>0</td>
<td>Any Interrupt Acknowledge cycle.</td>
</tr>
<tr>
<td>IACK7</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>IACK cycle on level 7.</td>
</tr>
<tr>
<td>IACK6</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>IACK cycle on level 6.</td>
</tr>
<tr>
<td>IACK5</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>IACK cycle on level 5.</td>
</tr>
<tr>
<td>IACK4</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>IACK cycle on level 4.</td>
</tr>
<tr>
<td>IACK3</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>IACK cycle on level 3.</td>
</tr>
<tr>
<td>IACK2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>IACK cycle on level 2.</td>
</tr>
<tr>
<td>IACK1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>IACK cycle on level 1.</td>
</tr>
<tr>
<td>UnDef</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Spurious, or undefined interrupt.</td>
</tr>
</tbody>
</table>

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*.

<table>
<thead>
<tr>
<th>Predefined symbol</th>
<th>ACF*</th>
<th>SYSF*</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>AcSy</td>
<td>0</td>
<td>0</td>
<td>Both SYSFAIL* and ACFAIL* asserted.</td>
</tr>
<tr>
<td>---Sy</td>
<td>1</td>
<td>0</td>
<td>SYSFAIL* asserted.</td>
</tr>
<tr>
<td>Ac--</td>
<td>0</td>
<td>1</td>
<td>ACFAIL* asserted.</td>
</tr>
<tr>
<td>----</td>
<td>1</td>
<td>1</td>
<td>None asserted.</td>
</tr>
</tbody>
</table>
In3:0

In3, In2, In1, In0

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, for the VBT-325 PCB revision “B”, the "V+T, V" jumper, see page 133. Also note that In3:0 is not available in the trace if Extended Time Tag is selected. See page 125.

P2trg*

P2Trg*

The cross trigger signal from the VBT-325C P2 part. This signal is not available on the VBT-325B.

<table>
<thead>
<tr>
<th>Predefined symbol</th>
<th>P2trg*</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>1</td>
<td>No trigger from the P2 part.</td>
</tr>
<tr>
<td>TRIG</td>
<td>0</td>
<td>The P2 tracer has triggered.</td>
</tr>
</tbody>
</table>

Cross-trigger signals

Cross trigger signals are asserted, i.e. set as TRIG, when the associated tracer triggers. The following table gives an overview of all possible cross trigger signals between the VBT-325 and the associated piggybacks:

<table>
<thead>
<tr>
<th>Signal name</th>
<th>Produced by</th>
<th>Appears on targets</th>
</tr>
</thead>
<tbody>
<tr>
<td>VMEtrg*</td>
<td>VBT-325 VME</td>
<td>VBT-325 P2</td>
</tr>
<tr>
<td>P2trg*</td>
<td>VBT-325 P2</td>
<td>VBT-325 VME</td>
</tr>
<tr>
<td>XMVtrg*</td>
<td>XMEM325 VME</td>
<td>VBT-325 VME</td>
</tr>
<tr>
<td>XMP2trg*</td>
<td>XMEM325 P2</td>
<td>VBT-325 VME</td>
</tr>
<tr>
<td>XVtrg*</td>
<td>XMEM325 VME</td>
<td>XMEM325 P2</td>
</tr>
<tr>
<td>XP2trg*</td>
<td>XMEM325 P2</td>
<td>XMEM325 VME</td>
</tr>
<tr>
<td>VBTVtrg*</td>
<td>VBT-325 VME</td>
<td>XMEM325 VME</td>
</tr>
<tr>
<td>TIMtrg*</td>
<td>TIM-200</td>
<td>VBT-325 VME</td>
</tr>
</tbody>
</table>

Vbat

Trigger signal from the VBAT piggyback. This signal is asserted whenever the VBAT detects a violation to the VMEbus protocol specification. The following predefined symbols applies:

<table>
<thead>
<tr>
<th>Predefined symbol</th>
<th>Vbat</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>OK</td>
<td>1</td>
<td>No VMEbus protocol violation detected by the VBAT.</td>
</tr>
<tr>
<td>VIOL</td>
<td>0</td>
<td>Some VMEbus protocol violation detected by the VBAT.</td>
</tr>
</tbody>
</table>
**Default Signal Fields in Timing Mode**

**BR3:0***  
BR3*, BR2*, BR1*, BR0*  
The Bus Request signals.

**BG3:0***  
BR3*, BR2*, BR1*, BR0*  
The Bus Grant signals.  
Latched BG information is also available as BgL. See page 150.

**Other signals**  
The others default signals in Timing mode use the standard VMEbus signals names. Signals ending in an asterix, “*”, are active low.

**BBSY***  
BBSY*  

**AS***  
AS*  

**WR***  
WRITE*  

**AM**  
See page 151.

**Address**  
See page 151.

**Data**  
See page 151.

**LWORD***  
LWORD*  

**DS1***  
DS1*  

**DS0***  
DS0*  

**DTACK***  
DTACK*  

**BERR***  
BERR*  

**IRQ7:1***  
See page 153.

**P2trg***  
See page 155.
Signal fields not included by default

Signal fields available in Events and Trace

The following signals are not, by default, included in the State or Timing screen templates. Use <Ins> to open a list of signals when in the Events, Trace or Search/Extract window:

- ACF*
- ACFAIL*
- BBSY*
- BUSBUSY*
- BCLR*
- BCLR*
- SCLK
- SYSCLK
- SYSR*
- SYSRESET*
- SYSF*
- SYSFAIL*
- IACK*
- IACK*
- IACKIO*
- IACKIO*
- RETRY*
- RETRY*
- Vme64*
- VME64*
- RmwBlk
- RmwBlk

This, internally generated, signal is used to identify cycles with VME64 AM-codes, i.e. AM = 00, 01, 03, 06, 07, 08, 0C, 38, or 3C. VME64* is a component of the Size and the Cycle fields.

RmwBlk

Internally generated. Asserted when AS* is not taken high between cycles. This means that the current cycle is the second (or subsequent) cycle in a VMEbus Read-Modify-Write cycle sequence, or a block cycle. RmwBlk is a component of the Size and the Cycle fields.
Signals available in Trace only

**DataASCII**  
D\text{\(31:0\)}, decoded as ASCII.  
Byte values outside the printable range, 0x20 - 0x7E are displayed as ".".

**AM5..AM0**  
The AM5, AM4, AM3, AM2, AM1 and AM0, normally grouped as AM, are also available as individual signals.  
Other individual signals are:

**BG3*..BG0**  
BG3*, BG2*, BG1* and BG0*.

**BR3*..BR0**  
BR3*, BR2*, BR1* and BR0*.

**In3..In0**  
The external inputs, In3..In0.

**IRQ7*..IRQ1**  
IRQ7*, .. IRQ1*.

**StateMode**  
Indicates the sampling mode of the current sample; 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.)
VSB Signal Groups

This chapter gives you a complete reference to all the signals used by the VSB targets on VBT-325C and the XMEM325 piggyback.

Default Signal Fields in State Mode

**Space**

Space [WR*], SPACE1, SPACE0

This group select between the different address spaces: System, I/O and Alternative. In addition, the WR* signal is included in this field to be able to select Interrupt Acknowledge cycles.

<table>
<thead>
<tr>
<th>Predefined symbol</th>
<th>WR*</th>
<th>SPC1</th>
<th>SPC0</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYS</td>
<td>X</td>
<td>1</td>
<td>1</td>
<td>System address space.</td>
</tr>
<tr>
<td>I/O</td>
<td>X</td>
<td>1</td>
<td>0</td>
<td>I/O address space.</td>
</tr>
<tr>
<td>ALT</td>
<td>X</td>
<td>0</td>
<td>1</td>
<td>Alternative address space.</td>
</tr>
<tr>
<td>IACK</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>Interrupt Acknowledge cycle.</td>
</tr>
</tbody>
</table>

**DS***

The data strobe, DS*, determines the phase of the VSBbus cycle. A cycle consist of one Address phase [ADDR], where the address information is driven by the master on the AD(31:0) lines. The Data [DATA] phase follows directly after where the master or the slave drives data on the same AD(31:0) lines. Block transfers consist of additional Data phases.

<table>
<thead>
<tr>
<th>Predefined symbol</th>
<th>DS*</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADDR</td>
<td>1</td>
<td>Address phase.</td>
</tr>
<tr>
<td>DATA</td>
<td>0</td>
<td>Data phase.</td>
</tr>
</tbody>
</table>

Editing this field, will change the contents of the Address and Data fields. If something is filled into the Address field, and DS* is changed to 1 or DATA, the contents of the Address field will be moved into the Data field - and vice versa.

**Address**

AD(31:0), Address phase.

When editing this field, the Data field will automatically be cleared and the DS* set to ADDR.
7 Signal reference

Data

AD(31:0), Data phase.

When editing this field, the Address field will automatically be cleared and the DS* set to DATA.

Size

SIZE1, SIZE0

The master asserts these signals to indicated wanted data transfer size.

<table>
<thead>
<tr>
<th>Predefined symbol</th>
<th>SIZ1</th>
<th>SIZ0</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>LONG</td>
<td>0</td>
<td>0</td>
<td>32 bit data transfer.</td>
</tr>
<tr>
<td>3BYT</td>
<td>1</td>
<td>1</td>
<td>24 bit data transfer.</td>
</tr>
<tr>
<td>WORD</td>
<td>1</td>
<td>0</td>
<td>16 bit data transfer.</td>
</tr>
<tr>
<td>BYTE</td>
<td>0</td>
<td>1</td>
<td>8 bit transfer.</td>
</tr>
</tbody>
</table>

ASACK*

ASACK1*, ASACK0*

Slave data transfer size acknowledge. The slave asserts these signals to answer on the master’s Size demand.

<table>
<thead>
<tr>
<th>Predefined symbol</th>
<th>ASACK1*</th>
<th>ASACK0*</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>NONE</td>
<td>1</td>
<td>1</td>
<td>Slave doesn't respond</td>
</tr>
<tr>
<td>D08</td>
<td>1</td>
<td>0</td>
<td>Response from 8-bit slave.</td>
</tr>
<tr>
<td>D16</td>
<td>0</td>
<td>1</td>
<td>Response from 16-bit slave.</td>
</tr>
<tr>
<td>D32</td>
<td>0</td>
<td>0</td>
<td>Response from 32-bit slave.</td>
</tr>
</tbody>
</table>

Cycle

WR*, PARB*, BLK

VSBus cycle type. The BLK signal is internally generated and asserted in the data phase(s) of the cycle when a block cycle is detected.

<table>
<thead>
<tr>
<th>Predefined symbol</th>
<th>WR*</th>
<th>PARB*</th>
<th>BLK</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read</td>
<td>1</td>
<td>X</td>
<td>X</td>
<td>Read cycle.</td>
</tr>
<tr>
<td>Write</td>
<td>0</td>
<td>X</td>
<td>X</td>
<td>Write cycle.</td>
</tr>
<tr>
<td>WBLK</td>
<td>0</td>
<td>X</td>
<td>1</td>
<td>Write block.</td>
</tr>
<tr>
<td>RBLK</td>
<td>1</td>
<td>X</td>
<td>1</td>
<td>Read block.</td>
</tr>
<tr>
<td>ARB</td>
<td>X</td>
<td>0</td>
<td>X</td>
<td>Arbitration cycle.</td>
</tr>
</tbody>
</table>
**Status**

ERR*, ACK*

VSbus cycle completion status:

<table>
<thead>
<tr>
<th>Predefined symbol</th>
<th>ERR*</th>
<th>ACK*</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERR</td>
<td>0</td>
<td>X</td>
<td>Error status.</td>
</tr>
<tr>
<td>-</td>
<td>1</td>
<td>1</td>
<td>Neither signal asserted.</td>
</tr>
<tr>
<td>OK</td>
<td>1</td>
<td>0</td>
<td>Ok status.</td>
</tr>
</tbody>
</table>

**LOCK***

LOCK*

VSbus LOCK* signal.

**IRQ***

IRQ*

VSbus IRQ* signal.

**VMEtrg***

VMEtrg*

The cross trigger signal from the VBT-325C VME part.

<table>
<thead>
<tr>
<th>Predefined symbol</th>
<th>VMEtrg*</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>1</td>
<td>No trigger from the VME part.</td>
</tr>
<tr>
<td>TRIG</td>
<td>0</td>
<td>The VME tracer has triggered.</td>
</tr>
</tbody>
</table>

**Default Signal Fields in Timing Mode**

**Space, DS***

As in State mode. See page 159.

**AD(31:0)**

AD(31:0)

VSbus Address and Data signals, any phase. Specify phase in the DS* signal.

**Other signals**

The others default signals in Timing mode use the standard VSbus signals names with some exceptions. Signals ending in an asterix, “*”, are active low.

BREQ*

BREQ*

BGIO*

BGIO*

BUSY*

BUSY*

PAS*

PAS*

ASACK*

See page 160.

ACK*

ACK*

ERR*

ERR*

WR*

WR*
### 7 Signal reference

<table>
<thead>
<tr>
<th>Signal</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size</td>
<td>See page 160.</td>
</tr>
<tr>
<td>LOCK*</td>
<td>LOCK*</td>
</tr>
<tr>
<td>WAIT*</td>
<td>WAIT*</td>
</tr>
<tr>
<td>ERR*</td>
<td>ERR*</td>
</tr>
<tr>
<td>IRQ*</td>
<td>IRQ*</td>
</tr>
<tr>
<td>VMEtrg*</td>
<td>Cross trigger from VME part.</td>
</tr>
</tbody>
</table>

#### Signal fields not included by default

**Signal fields available in Events and Trace**

The following signals are not, by default, included in the State or Timing screen templates. Use `<Ins>` to open a list of signals when in the Events, Trace or Search/Extract window:

- **AC**
  - VSBbus AC signal.

- **BLK**
  - Internally generated. Asserted when a block cycle is detected in the data phase(s) of the VSBbus cycle.

**Signals available in Trace only**

- **StateMode**
  - Indicates the sampling mode of the current sample; 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.)
SCSI Signal Groups

VBT-325C and XMEM325 This chapter gives you a complete reference to all the signals used by the SCSI targets on VBT-325C and the XMEM325 piggyback.

Default Signal Fields in State Mode

**DBP***

Data bus parity. Presented as-is, see below.

**DB7:0***

DB(7:0) *, Inverted.

The transferred data. Presented in inverted form.

**Phase***

MSG*, CD*, IO*

This group select between the different SCSI phases:

<table>
<thead>
<tr>
<th>Predefined symbol</th>
<th>MSG*</th>
<th>CD*</th>
<th>IO*</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSG_IN</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Message In phase</td>
</tr>
<tr>
<td>MSG_OUT</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>Message Out phase</td>
</tr>
<tr>
<td>STATUS</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>Status phase</td>
</tr>
<tr>
<td>COMMAND</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>Command phase</td>
</tr>
<tr>
<td>DATA_IN</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>Data In phase (i.e. Read data from target)</td>
</tr>
<tr>
<td>DATA_OUT</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>Data Out phase (i.e. Write data to target)</td>
</tr>
</tbody>
</table>

Other SCSI single bit signals:

**ATN***

ATN*

**RST***

RST*

**REQ***

REQ*

**SEL***

SEL*

**ACK***

ACK*

**BSY***

BSY*

**VMEtrg***

VMETrg*

The cross trigger signal from the VBT-325C VME part.

<table>
<thead>
<tr>
<th>Predefined symbol</th>
<th>VMEtrg*</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRIG</td>
<td>0</td>
<td>The VME tracer has triggered.</td>
</tr>
</tbody>
</table>
Default Signal Fields in Timing Mode

MSG* CD* IO*  The signal field layout in Timing mode is as in State mode, with the exception of the Phase field is missing. The single bits MSG*, CD* and IO* are presented in its place.

Signal fields not included by default

The following signals are not, by default, included in the State or Timing screen templates. Use <Ins> to open a list of signals when in the Events, Trace or Search/Extract window:

DBP1*  
Parity for the upper 8 data bits for 16-bit Wide SCSI-II. Presented as-is, see below.

DB15:8  
The upper 8 data bits for 16-bit Wide SCSI-II. Presented in inverted form.
VXI Signal Groups

**VBT-325C and XMEM325** This chapter gives you a complete reference to all the signals used by the VXI targets on VBT-325C and the XMEM325 piggyback using the VXE-35C VXI adapter. Only TTL level signals are routed via the adapter to the VBT-325. The VMEbus portion of the VXIbus is sampled as a normal VMEbus. ECL and ANALOG level signals are isolated from the VBT-325 by the VXI adapter.

Default Signal Fields in State and Timing Modes

**LBUS11:0** 
**LBUS (11:0)**

A 12-bit local, daisy-chained, bus. The VXE-35C connects the Local bus input pins directly to the corresponding output pins, allowing the VBT-325C / VXE-35C to be installed between two VXI hosts without breaking the LBUS connection.

**TTRGx**

TTRG7* .. TTRG0*

Eight, TTL level, open collector, trigger lines.

**VMEtrg**

VMETrg*

A cross trigger signal from the VBT-325 VME part.

<table>
<thead>
<tr>
<th>Predefined symbol</th>
<th>VMEtrg*</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>1</td>
<td>No trigger from the VME part.</td>
</tr>
<tr>
<td>TRIG</td>
<td>0</td>
<td>The VME tracer has triggered.</td>
</tr>
</tbody>
</table>
8 TRACE EXAMPLES

VMEbus

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.
8 Trace Examples

VSB

VMETRO VBT-325C

VSB Trace

Sampling: STATE at 25%

Trace Search Jump Format Markers Window Quit Help

Trc

Time | Space | Address | Data | SIZE-ASACK+Cycle-Status-LOCK+-->
--- | --- | --- | --- | ---
-4 0ns | SYS | 00800000 | ........ | LONG D32 | WRITE - 1
-3 310ns | SYS | ........ | 12345678 | LONG D32 | WRITE OK 1
-2 930ns | SYS | 00800000 | ........ | WORD D32 | WRITE - 1
-1 310ns | SYS | 00800000 | ........ | WORD D32 | WRITE OK 1
TRIG 1.17us | SYS | 00800000 | ........ | BYTE D32 | READ - 0
1 310ns | SYS | ........ | FF | BYTE D32 | READ OK 0
2 310ns | SYS | 00800000 | ........ | BYTE D32 | WRITE - 0
3 310ns | SYS | ........ | FF | BYTE D32 | WRITE OK 0
4 1.71us | SYS | 00800000 | ........ | LONG D32 | WRITE - 1
5 310ns | SYS | 00800000 | ........ | LONG D32 | WRITE OK 1
6 930ns | SYS | 00800000 | ........ | LONG D32 | WRITE OK 1
7 310ns | SYS | 00800000 | ........ | WORD D32 | WRITE OK 1
8 1.17us | SYS | 00800000 | ........ | BYTE D32 | READ - 0
9 310ns | SYS | ........ | FF | BYTE D32 | READ OK 0
10 310ns | SYS | 00800000 | ........ | BYTE D32 | WRITE - 0
11 310ns | SYS | ........ | FF | BYTE D32 | WRITE OK 0
12 1.71us | SYS | 00800000 | ........ | LONG D32 | WRITE - 1
13 310ns | SYS | 12345678 | LONG D32 | WRITE OK 1

Ok  <F2=Menu>  <F6=Nxt und>  <F6>  TGo

VMETRO VBT-325C

VSB Trace

Sampling: TIMING at 25%

Trace Search Jump Format Markers Window Quit Help

USB #1 - 80ns/div

X-T: 0ns | AD(31:0):00800000

TRIG SPACE

1

DS#

1

AD(31:0)

00800000

<

USB #2

Time | Space-DS+ | AD(31:0):BREQ+BGID+BUSY+PAS+ASACK+ACK+-->
--- | --- | --- | --- | ---
-4 29ns | 11 | 1 | 00040010 | 1 | 0 | 1 | 1 | 1
-3 29ns | 11 | 1 | 00040010 | 1 | 0 | 1 | 1 | 1
-2 29ns | 11 | 1 | 00040010 | 1 | 0 | 1 | 1 | 1
-1 29ns | 11 | 1 | 00040010 | 1 | 0 | 1 | 1 | 1
TRIG 29ns | 11 | 1 | 00040010 | 1 | 0 | 1 | 1 | 1
1 29ns | 11 | 1 | 00040010 | 1 | 0 | 1 | 1 | 1
2 29ns | 11 | 1 | 00040010 | 1 | 0 | 1 | 1 | 1

Ok  <F2=Menu>  <F6=Nxt und>  <F3=Prev edge>  <F4=Nxt>  TGo

VMETRO VBT-325 User’s Manual 167
9 VBT-325 FIRMWARE UPGRADE PROCEDURE

Upgrade

This chapter describes how to upgrade the firmware of your VBT-325.

Diskette for PCs

The firmware in VBT-325 is normally executed out of the onboard Flash Memory. Firmware upgrades are distributed as a diskette for IBM-compatible PCs, to be loaded via the serial port of the PC. When an upgrade is done, the new firmware is copied from the distribution diskette into the Flash memory by the means of code resident in a Boot PROM on the board.

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 memory, 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. Make sure you are using the correct Boot PROM version.

Version needed

For FW version 5.00 and higher, the Boot PROM must be of version 2.00 or higher. For VBT-325 with 2 Mbit Flash devices,1 Boot PROM version 2.14 or higher must be used.

No BOOT prom?
Boards with FW version lower than 4.99.08 do not have a Boot PROM. See page 175 for instructions about how to install a Boot PROM.

Firmware version 5.10

HW needed
Firmware Version 5.10 needs 2 Mbit Flash devices and BOOT Prom 2.14 or newer to load.

Extra features
included are loadable/storable setups and User Defined P2 tracer. See page 198 for more information.

1 A ECO label [board underside] “–2M” identifies 2 Mbit Flash devices.
Firmware upgrade preparations

RS232 connection

COM1 or COM2  Before starting the upgrade procedure, connect a RS232 cable from the COM1 or COM2 port on an IBM compatible PC to the Terminal port of the VBT-325. The recommended cable is shown in the figure on page 22.

+12V required

Before starting the upgrade procedure, make sure that +12V is supplied to the VBT-325 (at least 120mA).
Check the jumper J67. It should be installed as indicated in the figure to the left. See also page 178.
If there is any doubt whether +12V is present, execute the command Utilities/LED display to use the built-in voltage monitor.

Load firmware from PC

Insert the Firmware diskette into the floppy drive.

A: ↵
Type A: and <Cr> to set the floppy drive as the current drive. Replace the A with B if the diskette was inserted in drive B:

upload ↵
Type upload and <Cr> to start the Firmware upload procedure:

<table>
<thead>
<tr>
<th>VBT-325 Firmware</th>
<th>Version 5.02</th>
<th>03. Jan 1995</th>
</tr>
</thead>
<tbody>
<tr>
<td>Welcome to the VBT-325 Firmware Upload Procedure.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PLEASE READ ALL INSTRUCTIONS CAREFULLY BEFORE PROCEEDING AT ANY STEP!</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Select a number from the menu below that match the COM port which is connected to the TERMINAL port of the Tracer:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1. COM1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2. COM2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0. Abort installation.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>COM3 and COM4 are not supported.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Select port  Your Choice? _
9 VBT-325 firmware upgrade procedure

COM1 or COM2  Type <1> to select COM1, or <2> to select COM2, dependent of which port that is connected to the Terminal port of the VBT-325.

Reset the VBT-325. (If the VBT-325 is just powered up, this is not necessary.)

```
vt100: -p 9600 -P COM2 -x on -H off -D on -c on -t on -v C080 -h 25
```

<table>
<thead>
<tr>
<th>VBT-325 Firmware</th>
<th>Version 5.02</th>
<th>03. Jan 1995</th>
</tr>
</thead>
<tbody>
<tr>
<td>* Reset the VBT-325 (or the VPC-MkII)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>* Select a number from the menu below that match the text (i.e baud rate) displayed on the front panel display:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1. Blinking '19k2 Bin' .. 'Type CR'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2. Displaying '115k Bin'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3. .. '38k4 Bin'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4. .. '19k2 Bin'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5. .. '9600 Bin' or 'BootProm'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6. .. '4800 Bin'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7. .. '2400 Bin'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8. .. '1200 Bin'</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

8. Abort installation.

If none of the above match the text on the front panel display, select '0' to abort the installation.

**Select baud-rate**

Your Choice (1,2,3,4,5,6 or 0 to Abort)?  

1..6  Study the front panel display. Type the number of the menu item that match the text on the front panel display. The next step performed by the upload procedure is to wipe the current firmware from the Flash memory:

**WARNING**

The Flash EPROMS will now be erased. While erasing, the text XERASE is displayed in the front panel display.

I DO NOT TURN OFF THE POWER OR RESET THE TRACER AS LONG AS 'XERASE' IS DISPLAYED!

Erasing may take up to 30 seconds to complete.

Do not attempt to reset the tracer or turn off the power as long as this text is displayed. If you do, tuning parameters, vital to the operation of the tracer, will be lost. If the tuning parameters are lost, call VMETRO Support to get new values. See also the section *If things goes wrong* on page 173.
9 VBT-325 firmware upgrade procedure

Load baud rate

1, 2 or 3 Type <1>, <2> or <3> to select the baud rate to be used during the actual firmware file upload. On most PCs, 38k4 can be selected.

vt100: Sending file "flash.bin" using XMODEM CRC.
vt100: Sending block 4089 of 4088 Ok.

Loading.. Wait a while when the firmware is uploaded into the tracer. After a few minutes, the following screen will be displayed:

The firmware has been sucessfully uploaded. Please reset the Tracer.

Type RETURN when you have reset the Tracer. _

Reset the Tracer and type <Cr>.

If upload stops If the upload stops, start the upload procedure from the beginning, and try a slower baud-rate in the Load baud rate menu. If you are running the upload procedure in a Windows DOS box, and run into problems, try to exit Windows before making another attempt. Also, if you have special TSR programs bound to the used COM port, or network drivers, try a clean boot\(^1\) of the PC before making another attempt. See also the section If things goes wrong on page 173.

---

\(^1\) With MS-DOS 6.00 or later, this can be accomplished with holding both SHIFT-keys while the texts “Starting MS-DOS...” is displayed when the PC is booted.
Upload finished  When typing the last <Ct>, the following information is displayed:

```
VMETRO Terminal Emulator, VT100.EXE can be used with the
following parameters to communicate with the tracer:

vt100 -P COM2 -p 19k2

Or: Include the following statement in AUTOEXEC.BAT:

SET VT100=-P COM2 -p 19k2

This sets the default parameters used by VT100.EXE when the
PC is restarted. Starting VT100 without parameters starts
the VMETRO Terminal Emulator using the default parameters.
Parameters given on the command line will override the
default values in the VT100 environment variable.

Hints on using the VT100.EXE:

* Starting the VT100 with the single parameter “?” lists all
  accepted parameters.
```

The proposed settings for the VT100 emulator reflects the choices you have
made during the firmware upload. See page 208+ for more information on the
VMETRO VT100.

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.
If things goes wrong

Communication errors

The following message indicates communication problems:

```
ERROR

Cannot establish contact with the tracer. Please check your serial ports and cables before proceeding.

Refer to the documentation of the Tracer and configure the TERM port of the tracer to AUTO. Then, reset the Tracer and restart the installation.
```

☑ Check the cable. It should be in accordance to the drawing on page 22.
☑ Also check if the cable is connected to same COM-port as selected in the menu #1 on page 170.
☑ Check if the baud rate selection in the Select baud rate menu (page 170) matches the texts displayed on the front panel display of the VBT-325.

Flash memory errors

The following message indicates that 12V is missing or some other problem with the Flash memory:

```
ERROR

Cannot erase the Flash memory.

Check out whether the VME backplane supply 12V, and jumper J67 (12V jumper) is configured properly. Then, try the upload once more. If it still does not work, the Tracer may need service.
```

☑ Check if the system provides 12V.
☑ Check if jumper J67 is installed properly. See page 178.

If both these items checks out Ok, the Flash memory may be damaged. Please call VMETRO Support for further instructions.
Tuning Parameters Lost

If the following dialog box is displayed when the tracer is restarted, the board tuning parameters are lost:

```
In order to ensure reliable operation of your board, the tuning parameters must be re-entered.

Please identify the serial number, PCB revision, and ECO level on the underside of the board, and contact VMETRO to get the correct tuning parameters.
```

Type <Cr> to see the following dialog box:

```
Board Dependent Tuning Parameters

PCB rev : A (char)
ECO level: 99 (dec)

-BCS PW DelOn DelOff
  0:08 09 08 (hex)
  1:08 09 08
  2:08 09 08
  3:08 09 08
  4:08 09 08
  5:08 09 08

-AD converter tuning
  +5V slope: 47 offset: 447 (dec)
  +12V : 106 : 1033
  -12V : 99 : -1331
  Temp : 106 : -23

< Update HU > < Update Flash > < Cancel >
```

Firmware versions 4.99.05 or lower did not store tuning parameters. In this case, you will need to call VMETRO Support for the correct values for your tracer.

Call VMETRO Call VMETRO Support to get the correct tuning parameters for the Tracer. Without it, the tracer still works, but may show inaccurate results. Also, the firmware is dependent of a correct PCB revision and ECO level to fully utilize the hardware configuration of the tracer.
Missing PCB and ECO level

Some older firmware [version 4.99.10 and below] did not store the PCB revision and the ECO level together with the tuning parameters. The current firmware is dependent of a correct PCB revision and ECO level to fully utilize the hardware configuration of the tracer:

<table>
<thead>
<tr>
<th>Serial No</th>
<th>PCB ECO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Serial number, the PCB Revision, and the ECO level are found on a label on the underside of the board, and should be in accordance with the values below.</td>
<td></td>
</tr>
<tr>
<td>Serial Number: 00255378 (dec)</td>
<td></td>
</tr>
<tr>
<td>PCB rev: 1 (char)</td>
<td></td>
</tr>
<tr>
<td>ECO level: 21 (dec)</td>
<td></td>
</tr>
</tbody>
</table>

If this dialog box is displayed when the tracer is restarted, fill in the correct PCB revision and the ECO level. Then select <Update Flash> to store these parameters.

The command Utilities/Specials/ECO Level will allow you to enter or verify this at a later time if needed.

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, it must be replaced.

To install the Boot PROM and to complete the firmware upgrade procedure, 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. Refer to "Piggyback removal instructions" in the Users Manual for the piggyback modules.
- 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 J21, J22 and J23 are installed as shown in figure 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.
Run from Boot PROM

Now, when the board is powered up, it should come up with one of the texts

\[
\begin{align*}
&19k2 \quad 81n \\
&9600 \quad 81n
\end{align*}
\]

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. Follow the instructions starting on page 169.

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.
10 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.

Miscellaneous jumper settings VBT-325 PCB rev B.
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.

Jumper settings for VME and VSB, VBT-325 PCB revision “B”.
Jumper settings for VME and VSB, VBT-325 PCB revision “C”.
Jumper settings for VME and SCSI, VBT-325 PCB revision “B”.
Jumper settings for VME and SCSI, VBT-325 PCB revision “C”.
VMEbus and VXI (Twin mode)

VMEbus and VXI

Jumper Configuration
(Arrows indicate differences from VSB)

Temp/Ext1 (Shown in Ext1 pos.)

PROM size
Up: 28-pins
Down: 32-pins

Boot from PROM/FLASH
(PROM when missing)

Remove

VXI Jumper configuration for VXI_P2ac State analysis on VBT-325 PCB revision “B”. VXI_P2ac Timing analysis can use VSB jumper settings. Connect sampling clock to pin socket J43.
VXI Jumper configuration for VXI_P2ac *State* analysis on VBT-325C PCB revision “C”. VXI_P2ac *Timing* analysis can use VSB jumper settings. Connect sampling clock to pin socket J43.
Jumper configuration for User P2ac on VBT-325C PCB revision “B”. Connect P2AC sampling clock to pin socket J43.
Jumper configuration for User P2ac on VBT-325C PCB revision “C”. Connect P2AC sampling clock to pin socket J43.
Jumper configuration for VME, VSB and VXI (default) on XMEM325-PB PCB revision. “A”. Connect VXI sampling clock to pin-socket J43 [indicated as circle in the lower right corner].
Jumper configuration for VME and SCSI on XMEM325-PB print revision “A”.
Jumper configuration for VME and P2ac Udef on XMEM325-PB PCB revision “A”. Connect P2AC sampling clock to pin socket J43.
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.
VME & VSB Connection on P1 & P2

Names in *italics* denotes VSB signals.

<table>
<thead>
<tr>
<th>Pin #</th>
<th>P1 A</th>
<th>P1 B</th>
<th>P1 B</th>
<th>P2 A</th>
<th>P2 B</th>
<th>P2 B</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>D00</td>
<td>BBSY*</td>
<td>D08</td>
<td>AD00</td>
<td>+5V</td>
<td>AD01</td>
</tr>
<tr>
<td>2</td>
<td>D01</td>
<td>BCLR*</td>
<td>D09</td>
<td>AD02</td>
<td>GND</td>
<td>AD03</td>
</tr>
<tr>
<td>3</td>
<td>D02</td>
<td>ACFAIL*</td>
<td>D10</td>
<td>AD04</td>
<td>RETRY*</td>
<td>AD05</td>
</tr>
<tr>
<td>4</td>
<td>D03</td>
<td>BG0IN*</td>
<td>D11</td>
<td>AD06</td>
<td>A24</td>
<td>AD07</td>
</tr>
<tr>
<td>5</td>
<td>D04</td>
<td>BG0OUT*</td>
<td>D12</td>
<td>AD08</td>
<td>A25</td>
<td>AD09</td>
</tr>
<tr>
<td>6</td>
<td>D05</td>
<td>BG1IN*</td>
<td>D13</td>
<td>AD10</td>
<td>A26</td>
<td>AD11</td>
</tr>
<tr>
<td>7</td>
<td>D06</td>
<td>BG1OUT*</td>
<td>D14</td>
<td>AD12</td>
<td>A27</td>
<td>AD13</td>
</tr>
<tr>
<td>8</td>
<td>D07</td>
<td>BG2IN*</td>
<td>D15</td>
<td>AD14</td>
<td>A28</td>
<td>AD15</td>
</tr>
<tr>
<td>9</td>
<td>GND</td>
<td>BG2OUT*</td>
<td>GND</td>
<td>AD16</td>
<td>A29</td>
<td>AD17</td>
</tr>
<tr>
<td>10</td>
<td>SYCLK</td>
<td>BG3IN*</td>
<td>SYSFAIL*</td>
<td>AD18</td>
<td>A30</td>
<td>AD19</td>
</tr>
<tr>
<td>11</td>
<td>GND</td>
<td>BG3OUT*</td>
<td>BERR*</td>
<td>AD20</td>
<td>A31</td>
<td>AD21</td>
</tr>
<tr>
<td>12</td>
<td>DS1*</td>
<td>BR0*</td>
<td>SYSRES*</td>
<td>AD22</td>
<td>GND</td>
<td>AD23</td>
</tr>
<tr>
<td>13</td>
<td>DSO*</td>
<td>BR1*</td>
<td>LWORD*</td>
<td>AD24</td>
<td>+5V</td>
<td>AD25</td>
</tr>
<tr>
<td>14</td>
<td>WRITE*</td>
<td>BR2*</td>
<td>AM5</td>
<td>AD26</td>
<td>D16</td>
<td>AD27</td>
</tr>
<tr>
<td>15</td>
<td>GND</td>
<td>BR3*</td>
<td>A23</td>
<td>AD28</td>
<td>D17</td>
<td>AD29</td>
</tr>
<tr>
<td>16</td>
<td>DTACK*</td>
<td>AM0</td>
<td>A22</td>
<td>AD30</td>
<td>D18</td>
<td>AD31</td>
</tr>
<tr>
<td>17</td>
<td>GND</td>
<td>AM1</td>
<td>A21</td>
<td>GND</td>
<td>D19</td>
<td>GND</td>
</tr>
<tr>
<td>18</td>
<td>AS*</td>
<td>AM2</td>
<td>A20</td>
<td>IRQ*</td>
<td>D20</td>
<td>GND</td>
</tr>
<tr>
<td>19</td>
<td>GND</td>
<td>AM3</td>
<td>A19</td>
<td>DS*</td>
<td>D21</td>
<td>GND</td>
</tr>
<tr>
<td>20</td>
<td>IACK*</td>
<td>GND</td>
<td>A18</td>
<td>WR*</td>
<td>D22</td>
<td>GND</td>
</tr>
<tr>
<td>21</td>
<td>IACKIN*</td>
<td>SERA</td>
<td>A17</td>
<td>SPACE0</td>
<td>D23</td>
<td>SIZE0</td>
</tr>
<tr>
<td>22</td>
<td>IACKO*</td>
<td>SERB</td>
<td>A16</td>
<td>SPACE1</td>
<td>GND</td>
<td>PAS*</td>
</tr>
<tr>
<td>23</td>
<td>AM4</td>
<td>GND</td>
<td>A15</td>
<td>LOCK*</td>
<td>D24</td>
<td>SIZE1</td>
</tr>
<tr>
<td>24</td>
<td>A07</td>
<td>IRQ7*</td>
<td>A14</td>
<td>ERR*</td>
<td>D25</td>
<td>GND</td>
</tr>
<tr>
<td>25</td>
<td>A06</td>
<td>IRQ6*</td>
<td>A13</td>
<td>GND</td>
<td>D26</td>
<td>ACK</td>
</tr>
<tr>
<td>26</td>
<td>A05</td>
<td>IRQ5*</td>
<td>A12</td>
<td>GND</td>
<td>D27</td>
<td>AC</td>
</tr>
<tr>
<td>27</td>
<td>A04</td>
<td>IRQ4*</td>
<td>A11</td>
<td>GND</td>
<td>D28</td>
<td>ASACK1*</td>
</tr>
<tr>
<td>28</td>
<td>A03</td>
<td>IRQ3*</td>
<td>A10</td>
<td>GA0</td>
<td>D29</td>
<td>ASACK0*</td>
</tr>
<tr>
<td>29</td>
<td>A02</td>
<td>IRQ2*</td>
<td>A09</td>
<td>GA1</td>
<td>D30</td>
<td>CACHE*</td>
</tr>
<tr>
<td>30</td>
<td>A01</td>
<td>IRQ1*</td>
<td>A08</td>
<td>GA2</td>
<td>D31</td>
<td>WAIT*</td>
</tr>
<tr>
<td>31</td>
<td>-12V</td>
<td>+5V</td>
<td>STBY</td>
<td>+12V</td>
<td>BGIN*</td>
<td>GND</td>
</tr>
<tr>
<td>32</td>
<td>+5V</td>
<td>+5V</td>
<td>+5V</td>
<td>BREQ*</td>
<td>+5V</td>
<td>BGOUT*</td>
</tr>
</tbody>
</table>

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) *busied* 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.

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 flat cable, 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 flat cable, or that VMETRO’s "VSC-3" cable is used to extend an existing SCSI cable to allow it to pass the slot where the VBT-325C is located.
The pins shown in **bold** text are connected to the VBT-325 when the board is configured for SCSI.

<table>
<thead>
<tr>
<th>Pin #</th>
<th>P2 row A</th>
<th>P2 row B</th>
<th>P2 row C</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td><strong>DB0</strong>*</td>
<td></td>
<td><strong>Gnd</strong></td>
</tr>
<tr>
<td>9</td>
<td><strong>DB1</strong>*</td>
<td></td>
<td><strong>Gnd</strong></td>
</tr>
<tr>
<td>10</td>
<td><strong>DB2</strong>*</td>
<td></td>
<td><strong>Gnd</strong></td>
</tr>
<tr>
<td>11</td>
<td><strong>DB3</strong>*</td>
<td></td>
<td><strong>Gnd</strong></td>
</tr>
<tr>
<td>12</td>
<td><strong>DB4</strong>*</td>
<td></td>
<td><strong>Gnd</strong></td>
</tr>
<tr>
<td>13</td>
<td><strong>DB5</strong>*</td>
<td></td>
<td><strong>Gnd</strong></td>
</tr>
<tr>
<td>14</td>
<td><strong>DB6</strong>*</td>
<td></td>
<td><strong>Gnd</strong></td>
</tr>
<tr>
<td>15</td>
<td><strong>DB7</strong>*</td>
<td></td>
<td><strong>Gnd</strong></td>
</tr>
<tr>
<td>16</td>
<td><strong>DPB</strong>*</td>
<td></td>
<td><strong>Gnd</strong></td>
</tr>
<tr>
<td>17</td>
<td><strong>Gnd</strong></td>
<td></td>
<td><strong>Gnd</strong></td>
</tr>
<tr>
<td>18</td>
<td><strong>Gnd</strong></td>
<td></td>
<td><strong>Gnd</strong></td>
</tr>
<tr>
<td>19</td>
<td><strong>Gnd</strong></td>
<td></td>
<td><strong>Gnd</strong></td>
</tr>
<tr>
<td>20</td>
<td><strong>TermPWR</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td><strong>Gnd</strong></td>
<td></td>
<td><strong>Gnd</strong></td>
</tr>
<tr>
<td>22</td>
<td><strong>Gnd</strong></td>
<td></td>
<td><strong>Gnd</strong></td>
</tr>
<tr>
<td>23</td>
<td><strong>ATN</strong>*</td>
<td></td>
<td><strong>DB8</strong>*</td>
</tr>
<tr>
<td>24</td>
<td><strong>Gnd</strong></td>
<td></td>
<td><strong>DB9</strong>*</td>
</tr>
<tr>
<td>25</td>
<td><strong>BSY</strong>*</td>
<td></td>
<td><strong>DB10</strong>*</td>
</tr>
<tr>
<td>26</td>
<td><strong>ACK</strong>*</td>
<td></td>
<td><strong>DB11</strong>*</td>
</tr>
<tr>
<td>27</td>
<td><strong>RST</strong>*</td>
<td></td>
<td><strong>DB12</strong>*</td>
</tr>
<tr>
<td>28</td>
<td><strong>MSG</strong>*</td>
<td></td>
<td><strong>DB13</strong>*</td>
</tr>
<tr>
<td>29</td>
<td><strong>SEL</strong>*</td>
<td></td>
<td><strong>DB14</strong>*</td>
</tr>
<tr>
<td>30</td>
<td><strong>C/D</strong>*</td>
<td></td>
<td><strong>DB15</strong>*</td>
</tr>
<tr>
<td>31</td>
<td><strong>REQ</strong>*</td>
<td></td>
<td><strong>DBP1</strong>*</td>
</tr>
<tr>
<td>32</td>
<td><strong>I/O</strong>*</td>
<td></td>
<td><strong>Gnd</strong></td>
</tr>
</tbody>
</table>
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-cable type VSC-8

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.
SCSI-cable type VSC-3

To SCSI port on CPU with "paddle board"

To P2 of VBT-325C analyzer

To SCSI devices via existing SCSI cable

50-lead Flat cable

45cm/1.5ft

B

C

A Mounted on other side of cable!

5cm/2"

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.

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**.

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. Refer to the figure 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.

STOP

Before performing any cuts, please contact VMETRO Support for approval regarding warranty.
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. See the previous page for instructions.
12 P2AC USER DEFINED TRACER

This section describes the User Defined P2A/C software, available in firmware version 5.10 and newer. The User Defined P2A/C software supports:

- Definition of tracers for proprietary busses on the P2 A/C connector for the VBT-325C, the XMEM325-PB/C and S/TIM200-PB/C piggybacks.
- Dump and load setups, signal field definitions, and traces, to and from a PC or other hosts supporting the XMODEM-CRC protocol.

Slot Selection

In order to perform bus analysis of a user-defined P2 bus, one must make sure that the slot where the Tracer 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 Tracer in a neighboring slot to the board that has the P2 bus, and provide an "L-shaped" extension of the flat cable to the slot of the Tracer (see figure below.)
Jumper Settings

**VBT-325C**  By default, the VBT-325C is shipped with the P2 tracer configured for VSB. This is indicated by a label, P2=VSB, on the P2 connector itself. In order to use the P2 part of the VBT-325C for proprietary busses, the jumper configuration must be changed. Refer to page 186 for jumper settings for the VBT-325C, and page 190 for jumper settings for the XMEM325-PB/C. Check on the bottom of the PCB for the print version. The B print is labeled PCB-B, and the C print is labeled PCB-C.

If the P2-bus consists of TTL level [0-5V] signals only, this is all that needs to be done regarding configuring the HW. However, if the bus includes ECL-level, or other non-TTL level signals, these pins must be isolated from the VBT-325. Ignoring to do so may cause major HW damage to the VBT-325, and possible, to other boards connected to the proprietary P2-bus. See page 196 for instructions.

**Sampling clock select**

The sampling clock needed for State sampling is normally connected to the jumper J43. See page 186 for how to locate this jumper on the VBT-325C. See page 190 for how to locate this jumper on the XMEM325.

**VBT-325C only** Connect a wire from the wanted P2ac signal to the jumper J43. The P2ac User defined tracer will sample the state of the P2ac signals at every rising edge of the J43 clock signal.

If sampling on a falling edge is wanted, remove the jumper J42 and insert the wire of the clock signal into the middle position of the jumper J42 field as indicated.

**XMEM325-PB/C** Future firmware revisions will provide for selection between falling- and rising edge on the sampling clock.

**P2 Cross Trigger Jumper**

The cross trigger is controlled with a jumper on both the VBT-325C and the XMEM325-PB/C. The S/TIM200-PB has only one channel used for the cross trigger, so jumper settings are not necessary to use the cross trigger.

**Note** The figures below show the jumper position to connect a cross trigger to the analyzer. Since the analyzer only has 64 inputs [sampling channels], the input P2-c32 may be alternatively used as sampling input, or as a cross-trigger input.

**VBT-325C** The P2 cross trigger on the VBT-325C is controlled by the jumper J61 as shown below. Refer to the figure on page 186 to locate this jumper:

- P2-c32
- VBT-VME Cross trigger
- Piggyback Cross trigger
XMEM325  

The P2 cross trigger on the XMEM325-PB/C is controlled by the jumper J61 as shown below. See the figure on page 190 to locate this jumper:

P2-c32  XMEM325-VME Cross trigger  VBT-VME Cross trigger

Command Reference

In this chapter a complete reference of all additional menu commands, necessary to control the user defined P2 software, can be found.

Selecting Target

The VBT-325C and the XMEM325-PB/C allow simultaneous analysis of both the VMEbus and a P2ac user defined bus (or other P2 busses like VSB, SCSI or VXI). If the S/TIM200-PB is present, 200MHz analysis can be performed on VME or a P2 user defined bus (or other P2 busses like VSB, SCSI or VXI). The 'Target' command is used to switch between the different analyzers present in the tracer hardware configuration.

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

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

VME

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

P2 bus

The P2 bus on the VBT-325C can be selected from User P2, VSB, SCSI, and VXI_P2ac. Only the target currently selected is shown in the Target pull down menu, and by default, this is VSB. Other P2 busses can be selected by using the 'Reconfigure' command.

XMEM_VME bus

The primary target bus for the XMEM325-PB/C is XMEM_VME. This can be selected simultaneously with the VBT-325 VMEbus [except for the 128K model].
**XMEM_P2 bus**

The XMEM_P2 bus can be selected from User P2, VSB, SCSI, and VXI_P2ac. If the XMEM_P2 option is purchased, it can be enabled by the 'Reconfigure' option as described below. This can be selected simultaneously with the VBT-325 P2 bus [except for the 128K model].

**TIM200 bus**

When the S/TIM200-PB 200MHz Timing Analyzer is installed, it can be selected under the 'Target' command. If the P2 option 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 User P2 (SCSI or VXI_P2ac), the Reconfigure command must be used. (This command is also used to switch the bus for the XMEM325-PB/C and the S/TIM200-PB.) Reconfiguring the P2 bus on the VBT-325C normally requires moving some jumpers, therefore a warning is given when closing this command if changes have been made. See the installation chapter for a more detailed description.

**Note**  
All P2 targets must be configured to the same bus in order to exit.

The figure below shows the reconfigure dialog box for the VBT325C, left, and VBT325-C with XMEM325-PB/C, right:
Defining Signal Fields

The signal fields can be defined or edited from the event editor or a trace window [alphanumeric or waveform]. To be able to do this, select the target User Defined P2.

<Ins> or ^N

Type <Ins> or ^N to insert a new signal:

```
Insert
<Signal Field...>
<New Signal Field...>
(Cancel)
```

No signals?

The dialog box above will be displayed when there are signal fields to inserted in the current window. Select <Signal field...> to display the list of insertable signal. Select <New Signal Field...> to enter the signal edit dialog:

```
Field name : Dummy
Info Text  : Dummy Field : P2aM1
Display format : (x) Binary (_) Hex [ ] Inverted
Default presentation: (x) Visible, fixed ( ) Visible, scrollable ( ) Insertable [ ] Show value in waveform header
Pin List
(bits - 1)
(digits - 1)
[ ] Include pins used in other fields

Mnemonics
(no of - 0)

Select Name Value
< OK > < Cancel >
```

Field name

This item contains the name of the signal field that will be used in the event editor or trace window. The maximum length of the name is 9 characters. To make the event editor and trace window readable this field must be filled in.

Info text

The information text is the text displayed in the detailed editing dialog box. This is shown when pressing <Enter> on a signal in the event editor. If no information text is given, then a list of names of the selected pins is provided. (Note that the detail editing dialog box is shown only if the field is a hex field, or if mnemonics are defined.)

Display format

A signal field can be displayed either as a binary or hex number. By default low levels on the pins are displayed as 0, and high levels as 1. This can be inverted by selecting the check box ‘Inverted’.
Default
Presentation

These items control how the signal fields are presented after ‘Setup/Initialize’ or clearing the non-volatile memory. The STIM does not have the ‘Setup/Initialize’ command, but each time the tracer is reset or powered up, the signal fields are reset as specified by the default presentation parameters.

Fields can be displayed in the fixed area on the left side of the screen, in the scrollable area to the right side of the screen, or be in the insert list (which means it is not visible). The ‘Show value in waveform header’ controls if the signal field should be present in the header line of the trace waveform window. The value under the cursor will be displayed. This is normally used for signal fields like address and data.

[ ] Include pins used in other fields

This check box controls the selection list for the ‘Pin list’. When it is not set (default), only those pins not used by other signals and signal fields are shown in the selection list. When it is set all pins except the pins used by the signal itself are shown in the selection list.

Pin List

This is a list of all the pins assigned to the signal field definition. The number of bits is a count of the bits in the list. The number of digits is the width of the field on the screen.1

↓

To add a new pin place the cursor on the last pin which contains “?” and type <Down>:

↑ ↓ then ↓

Use the arrow keys to select the pin and press <Cr> to select. To change an already defined pin, move the cursor to that pin and follow the steps described above.

<Ins>

A new pin can be added between the pin currently pointed to by the cursor and the previous pin by typing <Ins>. The first pin in the selection list is then inserted. To edit this, use the <Down> - arrow as described above.

<Del>

Type <Del> to remove the pin pointed to by the cursor. All pins to the right of the one deleted are shifted one position to the left.

---

1 The default field, Dummy, which uses the A01 pin, may be deleted when the first “real” field is defined.
Mnemonics

Mnemonics, or predefined values with a specific bit pattern, can be entered for each field. Example: A signal named Write can be defined to have the mnemonics WR with value “0” and RD with value “1”. When editing a signal in the event editor these values can be selected. The signal will also be displayed as WR or RD in an alphanumeric trace window when signal decoding is on.

An accelerator key (the character with the underscore in the example above, and in the dialog box below) can be defined for each value by typing “&” in front of the letter to be used.

Select mnemonic

Type <Down> on this item to see a list of the defined mnemonics:

![Select Object](image)

Select the “?” mark to add a new one in the bottom, or select one of those already defined, i.e. WR or RD, to change the name or the value.

<Ins>

A new mnemonic can be inserted by typing <Ins> in the select item (The selection list must not be open). The mnemonic receives the name (NEW) and the value is set to “X”, i.e. all bits don’t care.

<Del>

Type <Del> to delete the currently selected mnemonic. The next in the list or “?” will become the current.

Note

The first value defined should always be the most stringent one because this is the one the analyzer compares the sampled data to first.

Name

The name should be defined when the wanted mnemonic is selected (eventually ‘?’ for a new one). Use “&” in front of a letter to define it as the accelerator key.

Value

Fill in the defined value (pattern) the mnemonic in this item. Digits and ‘X’-s (do not care) can be used. Go back to the select area and select ‘?’ to define the next mnemonic.

Edit an existing field definition

<F7> or ^V

Type <F7> [or ^V] to edit the current signal field. Follow the instructions starting on page 202.
How to delete a signal field definition

<Del> Move the cursor to a signal field and type <Del> to deleted it. If the field contains any values these will be cleared the first time <DEL> is pressed. The second time it is pressed or if no values are defined, the dialog box below appears on the screen.

![Delete dialog box]

<Delete...> Select <Delete Signal Field> to delete the signal field from the list of signal fields. This will delete the signal field from all windows it is used in and redraw the windows. The signal field can not be inserted again later, since the definition of it is removed from memory!

<Hide...> Select <Hide Signal Field> to remove the signal field from the current window. The signal field will not be hidden in other trace windows. It can later be inserted again by pressing the <Ins> key and select <Signal Field...>. 
Using time tags

The time tag is supported for user defined P2 on the VBT-325C and the XMEM325-PB/C (same as for other targets). Storing the time tag requires 12 or 16 bits of the trace memory, which must be taken from the P2 pins. The pins that will be used is controlled by the dialog box shown in figure below.

Execute the command Edit/Sampling Mode/Options/State Options... to open the dialog box:

```
User P2 State Sampling Options

(X) No Time Tag

Extended time tag range (50ns - 1 hr 38 min)
P2a80:01 and P2c80:01 not available in trace

Limited time tag range (50ns - 6 min 8 sec)
P2a80:03 and P2c80:03 not available in trace

[J59 in P2a30 position]

< Ok > < Cancel >
```

**No time tag**  When *No Time Tag* is selected, all pins on the P2 connector can be used. The time tag field will be displayed as “...” indicating that no time tag is available.

**Extended**  The next two selections give the largest range of the time tags (50ns -1hr 38 min). These require 16 bits to store are therefore 16 pins are not available in the trace memory. If they still are used, they will be displayed as ‘.’ (dots) or ‘$’ (hex digits where not all bits are used by the time tag) in the trace windows.

**Limited**  The last two selections provide a much smaller range (50ns-6min 8sec). They require only 12 bits to be stored, therefore the number of pins not available in the trace buffer is also reduced to 12.

**[x] J59 in..**  The jumper J59 (see page 186) is by default installed in the lower position to allow sampling of the P2a30 signal. To be able to use time tags in *mixed sampling mode*, the J59 should be installed in the upper position. In this case, the P2a30 signal is not sampled, but replaced by the internally generated signal StateMode. See page 162.

**Note**  Signal fields with pins also used by the time tags, can still be used in an event pattern as part of the trigger- or store-condition.
Dump/Load Setup to/from PC or Host

The Dump Setup to PC/Host command stores the same way as the ‘Setup/Store...’ command. Instead of storing it in non-volatile memory, it uses the X-modem protocol to dump it to a file on the PC/host. The dump setup to PC/host is a feature that is available for all user defined P2 targets and all other targets except the STIM. The following parameters belong to a setup:

- Signal field definitions must be the same for all setups loaded into the tracer simultaneously. They are only stored for user defined P2, since they are fixed for other targets. For the STIM, only the signal field definitions are stored. This means the Setup command is not available for other STIM targets than user defined P2.

- Events with the inserted list of signal fields and their values (one for state and one for timing analysis).

- The Sequencer program.

- The selected sampling frequency.

- Trigger input options for the TIMBAT

- The statistics option parameters, one for each statistics function.

The Load from PC/Host works in the opposite direction. For more details, see page 129 and page 130.
Features

The program `VT100.EXE` on the Firmware Distribution Diskette is a VT100 terminal emulator program for IBM-compatible PCs. It offers a number of valuable features that help you take full advantage of the VBT-325 product:

- A VT100 emulator program tailored to the VBT-325.
- Facilitates firmware upgrade through the serial port.
- Allows trace dump/load to/from file on a PC.
- A powerful script language.

In addition to emulating a standard 25 lines x 80 character Vt100 screen, the `vt100` also allows you to take advantage of a 50 lines x 80 character display on VGA and VGA compatible display adapters.

Serial port

The `vt100` will by default use the COM1 port. Using the COM2 port is controlled by the `-P` option. COM3 and COM4 is not supported.

ANSI emulation

`vt100` is dependent of the driver `ANSI.SYS`. Make sure that your `config.sys` file includes the following statement:

```
device=c:\dos\ansi.sys
```

If you do not find this or a similar statement, locate the directory where `ansi.sys` resides on your PC, normally `c:\` or `c:\dos`, and add the above statement (with correct path) to `config.sys`.

The PC needs to be restarted to reflect changes in `config.sys`.

How to use the VT100

Usage

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, for example:

```
c> vt100 -p 19k2
```

Starts the emulator with a baud rate of 19200. Notice the space between the option `-p` and the baudrate, `19K2`. 
### Options

To display all options, start vt100 as indicated below:

```
c> vt100 ?  ↓
```

### Notation

An option consist of a hyphen, the option character, a space and a modifier. The option characters are case significant.

<table>
<thead>
<tr>
<th>Command line option</th>
<th>Default</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>-?</td>
<td>-</td>
<td>Display all options.</td>
</tr>
<tr>
<td>-p ?</td>
<td>-</td>
<td>Display all baud rates.</td>
</tr>
<tr>
<td>-P {1</td>
<td>COM1</td>
<td>2</td>
</tr>
<tr>
<td>-b {300</td>
<td>1200</td>
<td>2400</td>
</tr>
<tr>
<td>-x {on</td>
<td>off}</td>
<td>on</td>
</tr>
<tr>
<td>-H {on</td>
<td>off}</td>
<td>off</td>
</tr>
<tr>
<td>-c {on</td>
<td>off}</td>
<td>on</td>
</tr>
<tr>
<td>-v {CO80</td>
<td>MONO}</td>
<td>*)</td>
</tr>
<tr>
<td>-h {25</td>
<td>43</td>
<td>50}</td>
</tr>
<tr>
<td>-D {on</td>
<td>off}</td>
<td>on</td>
</tr>
<tr>
<td>-t {on</td>
<td>off}</td>
<td>on</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td>-s file</td>
<td>Send file from PC to tracer using XMODEM CRC protocol. The tracer must have been set in receive mode before this command can be issued.</td>
<td></td>
</tr>
<tr>
<td>-r file</td>
<td>Receive a file from the tracer using XMODEM CRC protocol. The tracer must have been set up for transmit before this command can be issued.</td>
<td></td>
</tr>
<tr>
<td>-i file</td>
<td>Read input from file instead of the keyboard.</td>
<td></td>
</tr>
<tr>
<td>-o file</td>
<td>Direct output to file instead of the screen.</td>
<td></td>
</tr>
</tbody>
</table>

*) Depends on the video adapter. Defaults to MONO on MDA adapters, and CO80 on CGA, EGA, VGA, SVGA and others.

**The VT100 environment variable**

Vt100 looks for the environment variable VT100 (or vt100) when started. The environment variable before the command line, allowing command line options to override options set in the environment variable.

Typing the command

```
c> set VT100=-p 38k4 -H on ↓
```

once, and then

```
c> vt100 ↓
```

is equivalent to

```
c> vt100 -p 38k4 -H on ↓
```

but save you from tedious writing every time vt100 is started.

No space should be present between the word VT100 and the equal sign, ",=". Including the command `set VT100 in autoexec.bat`, sets the vt100 defaults every time the PC is started.
Terminal types to use on the tracer

If your PC has a color or monochrome CGA, EGA or VGA screen, you should select terminal type # 3. Select terminal type #2 if you have a monochrome screen (MDA display adapter).

Built-in XMODEM CRC protocol

When started, vt100 displays the following lines:

Options: Type Alt-R to receive a file, type Alt-S to send a file. Type Alt-X or Ctrl-Z to end communication.

Alt+S Type Alt+S to send a file to the host (i.e. the tracer). The host must have been set in receive mode before the command is issued.

Alt+R Type Alt+R to receive a file from the host (i.e. the tracer). The host must have been set in transmit mode before the command is issued.

Alt+X Exit vt100.

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.

```
c> vt100 -i script.inp
```

Will start the vt100 emulator, taking input from the file script.inp instead of the keyboard.

Script control commands

The script file should be standard ASCII text. All the keywords and options are case significant.

"italics" Replace with the actual file name, baud-rate or string.

*italics* Replace with actual numeric argument.

{on|off} Use one of the listed modifiers.

[options] Parameters between square brackets, [ ] are optional.

; Semicolon separates multiple statements in one line.

stmt A statement, or statement separated by semicolon.
<table>
<thead>
<tr>
<th>% Comment</th>
<th>Comment.</th>
</tr>
</thead>
<tbody>
<tr>
<td>:Label</td>
<td>Label. Target for the GOTO statement. 8 significant characters.</td>
</tr>
<tr>
<td>AWAIT &quot;string&quot;</td>
<td>Wait for string. Use TIMEOUT to set AWAIT timeout.</td>
</tr>
<tr>
<td>BAUD baudrate</td>
<td>Set new baudrate. Should be followed by a PAUSE statement settle the I/O before continuing.</td>
</tr>
<tr>
<td>DOWNLOAD &quot;file&quot;</td>
<td>Start XMODEM download (receive) of the given file.</td>
</tr>
<tr>
<td>ECHO (ON</td>
<td>OFF)</td>
</tr>
<tr>
<td>EXIT {exitcode}</td>
<td>Exit the program. The numeric exit code may be tested by the controlling environment.</td>
</tr>
<tr>
<td>GOTO label</td>
<td>Go to named label. The &quot;:&quot; in the label statement should not be used here.</td>
</tr>
<tr>
<td>IF AWAIT; stmt</td>
<td>If the last AWAIT statement found a match, execute the statement [or statements] that follows.</td>
</tr>
<tr>
<td>IF TIMEOUT; stmt</td>
<td>If the last AWAIT statement timed out, execute the statement [or statements] that follows.</td>
</tr>
<tr>
<td>IF &quot;string&quot;; stmt</td>
<td>Compare last user input (READ or READCH) with string. If match execute the statement [or statements] that follows.</td>
</tr>
<tr>
<td>PAUSE time</td>
<td>Wait time * 10ms before continuing. Output from the host will be ignored in the meantime.</td>
</tr>
<tr>
<td>READ</td>
<td>Read line from the keyboard (or stdin, if redirected).</td>
</tr>
<tr>
<td>READCH</td>
<td>Read one character from the keyboard (or stdin, if redirected). Useful for &quot;(Y/N)&quot; type of questions.</td>
</tr>
<tr>
<td>SEND &quot;string&quot;</td>
<td>Send string to host (i.e. the tracer). Ampersands, &quot;;&quot; in the string are translated to CR. See also table below for how to specify function- and navigation-keys.</td>
</tr>
<tr>
<td>TIMEOUT seconds</td>
<td>Set the time-out period for AWAIT, i.e. how long the AWAIT should wait for a given string in seconds.</td>
</tr>
</tbody>
</table>
UPLOAD "file"  Start XMODEM upload (send) of the given file.

USER  Enter user mode. In user mode, keyboard input are transmitted to the host and host output directed to the screen.

The script continues when the user types ^C or Alt+X.

WRITE "text"  Print text on the screen "\n" in the string is treated as newline.

**Host output** is read completely [and echoed to the screen if ECHO ON] between the execution of each script line, so

SEND "string"; AWAIT "response"

should be written as one line to avoid missing the expected response.

**Function keys in script files**

The table below explains how to specify function keys in scripts. This way of specifying function keys is not specific to the VT100.EXE. It is built into the tracer firmware, not the VT100.EXE itself..

<table>
<thead>
<tr>
<th>Name in script file</th>
<th>Keystroke</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>\U</td>
<td>Cursor up, ↑</td>
<td></td>
</tr>
<tr>
<td>\D</td>
<td>Cursor down, ↓</td>
<td></td>
</tr>
<tr>
<td>\R</td>
<td>Cursor right, →</td>
<td></td>
</tr>
<tr>
<td>\L</td>
<td>Cursor left, ←</td>
<td></td>
</tr>
<tr>
<td>\H</td>
<td>HOME</td>
<td></td>
</tr>
<tr>
<td>\1</td>
<td>Function key F1</td>
<td>Help</td>
</tr>
<tr>
<td>\2</td>
<td>Function key F2</td>
<td>Edit last window/ go to menu</td>
</tr>
<tr>
<td>\3</td>
<td>Function key F3</td>
<td>(Trace Display menu only) Find next search pattern</td>
</tr>
<tr>
<td>\4</td>
<td>Function key F4</td>
<td></td>
</tr>
<tr>
<td>\</td>
<td>\ \</td>
<td>Refresh screen</td>
</tr>
<tr>
<td>\F1</td>
<td>Function key F1</td>
<td></td>
</tr>
<tr>
<td>\F2</td>
<td>Function key F2</td>
<td></td>
</tr>
</tbody>
</table>
Script example #1

The following example shows most of the features of the script language:

```
TIMEOUT 5
ECHO OFF
:Start
  WRITE "\n\nReset the VBT-325."
  WRITE "\nDoes the display blink '19k2' and 'Type CR' (Y/N)? "
  READCH
  IF "Y"; GOTO Auto
  IF "N"; GOTO Cont
  GOTO Start

:Auto
  SEND ";"; PAUSE 10; SEND ";"; AWAIT "TO CONTINUE: "
  IF AWAIT; GOTO Cont
  WRITE "\nCannot establish contact with tracer.."
  GOTO Start

:Cont
  SEND ";"; AWAIT "Version 2.00"
  IF AWAIT; GOTO Cont2
  WRITE "\nCannot establish contact with tracer.."
  GOTO Start

:Cont2
  PAUSE 50
  SEND "debug&"; AWAIT "XMON> "
  WRITE "\nEntering interactive mode.."
  ECHO ON
  SEND "&"

:UserMode
  USER
  WRITE "Finished? (Y/N) "
  READ
  IF "N"; GOTO UserMode
  EXIT 0
```
Script example #2

The following example shows how to change the baud rate:

```%``
% Select baud rate:
%```
```
:BaudRate
WRITE "\n Please select a baud-rate: \n"
WRITE "\n 1. 38k4               
WRITE "\n 2. 19k2               
WRITE "\n 3. 9600               
WRITE "\n 0. Abort installation."
WRITE "\n Your Choice (1,2,3 or 0 to Abort)? "
READCH
IF "0"; GOTO Exit
IF "1"; SEND "speed 38K4&e"; PAUSE 5; BAUD 38k4; GOTO Upload
IF "2"; SEND "speed 19K2&e"; PAUSE 5; BAUD 19k2; GOTO Upload
IF "3"; SEND "speed 9600&e"; PAUSE 5; BAUD 9600; GOTO Upload
GOTO BaudRate
```

:Upload

upload.inp  Look at the two files upload.bat and upload.inp on the Firmware Distribution Diskette for a more elaborate example.

---

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:

<table>
<thead>
<tr>
<th>VMETRO_TRACE</th>
<th>MODEL Tag</th>
<th>Comments</th>
<th>^Z</th>
</tr>
</thead>
</table>

The "VMETRO TRACE" identifies the file type. The Tag 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:

<table>
<thead>
<tr>
<th>ID</th>
<th>W</th>
<th>Data with length W bytes</th>
</tr>
</thead>
</table>

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:

<table>
<thead>
<tr>
<th>ID number</th>
<th>Width</th>
<th>Description of the data field</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>62</td>
<td>Main Header</td>
</tr>
<tr>
<td>200</td>
<td>1</td>
<td>Unpacked Trace buffer data</td>
</tr>
<tr>
<td>201</td>
<td>2</td>
<td>Run length packed Trace buffer data</td>
</tr>
</tbody>
</table>

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):

<table>
<thead>
<tr>
<th>Runs</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-65536</td>
<td>Trace line data to be repeated &quot;Runs&quot; times.</td>
</tr>
</tbody>
</table>
### Main Header

The Main Header has the following data fields:

<table>
<thead>
<tr>
<th>Name of field</th>
<th>Size in bytes</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>char nlTag[10]</code></td>
<td>10</td>
<td>Target ID</td>
</tr>
<tr>
<td><code>FLAGS nlLastRunFlags</code></td>
<td>4</td>
<td>Trace control flags:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- NF_TRCASYNC 0x00008000  Timing trace</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- NF_TIMETAG 0x00010000  Time tag used</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- NF_TAG16 0x00020000  16-bit time tag</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- NF_TRCMIXED 0x00800000 Mixed trace</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- NF_XMEMADJ 0x01000000 XMEM tag</td>
</tr>
<tr>
<td><code>BYTE nhLastRunTimingIdx</code></td>
<td>1</td>
<td>Speed used for last run</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0=50 MHz, 1=25 MHz, 2=6.25 MHz, 3=1.46 MHz, 4=781.3 KHz, 5=390.6KHz, 6=195.3 KHz, 7=97.7KHz</td>
</tr>
<tr>
<td><code>BYTE nhLastRunTrigPos</code></td>
<td>1</td>
<td>Trigger position used for run</td>
</tr>
<tr>
<td><code>BYTE nhTrcWidth</code></td>
<td>1</td>
<td>Width of sample in bytes</td>
</tr>
<tr>
<td><code>INT32 nhDelay</code></td>
<td>4</td>
<td>Trig delay (given by trig position)</td>
</tr>
<tr>
<td><code>INT32 nhFirstTrig</code></td>
<td>4</td>
<td>Trig address in trace memory (abs)</td>
</tr>
<tr>
<td><code>INT32 nhFirstValTrcLine</code></td>
<td>4</td>
<td>First valid line (log) in trace buffer</td>
</tr>
<tr>
<td><code>INT32 nhLastValTrcLine</code></td>
<td>4</td>
<td>Last valid line (log) in trace buffer</td>
</tr>
<tr>
<td><code>BOOLEAN nhTrgFound</code></td>
<td>2</td>
<td>Indicates trigger found</td>
</tr>
<tr>
<td><code>BOOLEAN nhTrcCompleted</code></td>
<td>2</td>
<td>Indicates trace completed</td>
</tr>
<tr>
<td><code>char nhTrigLineTxt[10]</code></td>
<td>10</td>
<td>Trigger line text</td>
</tr>
<tr>
<td><code>char nhTime[8]</code></td>
<td>8</td>
<td>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]</td>
</tr>
<tr>
<td><code>INT16 nhCalcADCVal[4]</code></td>
<td>8</td>
<td>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</td>
</tr>
</tbody>
</table>
**Main header**

As a “C” struct definition, the MAINHEADER looks like:

```c
typedef short INT16;        /* ................ Basic types */
typedef unsigned short UINT16;
typedef long INT32;
typedef unsigned long UINT32;
typedef unsigned long FLAGS;
typedef unsigned short BOOLEAN;
typedef unsigned char BYTE;

typedef packed struct _MAINHEADER {
    char nlTag [NL_TAGSIZE];/* .......... Tag, or name string */
    FLAGS nhLastRunFlags;    /* .. .nlFlags copied at last RUN */
    BYTE nhLastRunTimingIdx;/* ...... Speed used for last run */
    BYTE nhLastRunTrigPos;  /* ... Trig-pos used for last run */
    BYTE nhTrcWidth;        /* ..... Width of sample in bytes */
    INT32 nhDelay;           /* ................... Trig delay */
    INT32 nhFirstTrig;       /* ............ Trigger addr (abs) */
    INT32 nhFirstValTrcLine; /* ........ First valid line (log) */
    INT32 nhLastValTrcLine; /* ........ Last valid line (log) */
    BOOLEAN nhTrgFound;      /* ........... Trigger found flag */
    BOOLEAN nhTrcCompleted;  /* .............. Trace full flag */
    char nhTrigLineTxt[10];  /* ..........  "TRIG", "HALT" etc */
    char nhTime[TRCTIMESIZE];/* ...Time when trigged or halted */
    INT16 nhCalcADCVal[ADC_SIZE]; /* ADC values when trig/halt */
} MAINHEADER;
```

`packed` indicates that the struct is packed on one-byte border, i.e. contains no filler bytes. Some compilers do not recognize the keyword `packed`. Microsoft compilers uses `#pragma pack( 1 )` to specify packing on byte borders. If in doubt, confer your compiler manuals.
Trace Data Line format, VBT-325 VME

Each trace data struct consist of several Header->nhTrcWidth wide trace lines. For the VBT-325 VME target, the structure of each trace data line is as follows.

typedef unsigned char BYTE;
typedef unsigned long UINT32;

typedef packed struct _VMETRACE {
    union {
        BYTE cAddr[4]; /* LSB in cAddr[0], MSB in cAddr[3] */
             UINT32 Addr;
    };
    union {
        BYTE cData[4]; /* LSB in cAddr[0], MSB in cAddr[3] */
             UINT32 Data;
    };
    BYTE TagL, TagU; /* Time-tag. See below */
    BYTE Am, Irq, Str, Bg;
    BYTE St2, St3;
} VMETRACE;

The trace address and data is stored as Intel format long integers, i.e. the least significant byte is byte cAddr[0] or cData[0], the most significant byte is byte cAddr[3] or cData[3].

Details of the time tag bits

The format of the time tag is the same for all targets. Check the Header->n1LastRunFlags.bit NF_TIMETAG to see whether the time tag is in use or not. If this bit is 1, check the bit NF_TAG16. If this bit is 1, the 16-bit time tag is used. For traces sampled in timing mode, the time tag is not in use. Check the Header->n1LastRunFlags.bit NF_TRCASYN.

A one in this bit indicates timing mode.

The default 12-bit tag maps the four external inputs into the four lower bits of the TagL byte:

<table>
<thead>
<tr>
<th>Bit#</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>TagL</td>
<td></td>
<td>Tag[3:0]</td>
<td>EXT3</td>
<td>EXT2</td>
<td>EXT1</td>
<td>EXT0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

When the 16-bit tag is used,

<table>
<thead>
<tr>
<th>Bit#</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Bit#</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
</table>
How to convert the time tag to a time value

The counter starts out with a 20 ns resolution, but when the time gets larger it changes automatically the frequency. The four most significant bits, called the tag prescale value, tell us at what frequency the counter last used, and thereby the counter resolution.

Here are two examples of how the tag value can be calculated:

0xABCD (16 bit) : 125.74720ms + (0xBCD * 81.92us) = 373.22752ms
0x123 (12 bit) : 5.12us + (0x23 * 40ns) = 6.52us

<table>
<thead>
<tr>
<th>Prescale</th>
<th>16 bit</th>
<th>12 bit</th>
<th>Resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x0</td>
<td>0</td>
<td>0</td>
<td>20 ns</td>
</tr>
<tr>
<td>0x1</td>
<td>81.92 us</td>
<td>5.12 us</td>
<td>40 ns</td>
</tr>
<tr>
<td>0x2</td>
<td>245.76 us</td>
<td>15.36 us</td>
<td>80 ns</td>
</tr>
<tr>
<td>0x3</td>
<td>573.44 us</td>
<td>35.84 us</td>
<td>160 ns</td>
</tr>
<tr>
<td>0x4</td>
<td>1.22880 ms</td>
<td>76.80 us</td>
<td>320 ns</td>
</tr>
<tr>
<td>0x5</td>
<td>2.53952 ms</td>
<td>158.72 us</td>
<td>640 ns</td>
</tr>
<tr>
<td>0x6</td>
<td>5.16096 ms</td>
<td>322.56 us</td>
<td>1.28 ms</td>
</tr>
<tr>
<td>0x7</td>
<td>10.40384 ms</td>
<td>650.24 us</td>
<td>2.56 ms</td>
</tr>
<tr>
<td>0x8</td>
<td>20.88960 ms</td>
<td>1.30560 ms</td>
<td>5.12 us</td>
</tr>
<tr>
<td>0x9</td>
<td>41.86112 ms</td>
<td>2.61632 ms</td>
<td>20.48 us</td>
</tr>
<tr>
<td>0xA</td>
<td>125.74720 ms</td>
<td>7.85920 ms</td>
<td>81.92 us</td>
</tr>
<tr>
<td>0xB</td>
<td>461.29152 ms</td>
<td>28.83072 ms</td>
<td>327.68 us</td>
</tr>
<tr>
<td>0xC</td>
<td>1.8034688 s</td>
<td>112.71680 ms</td>
<td>1.31072 ms</td>
</tr>
<tr>
<td>0xD</td>
<td>7.17217792 s</td>
<td>448.26112 ms</td>
<td>10.48576 ms</td>
</tr>
<tr>
<td>0xE</td>
<td>50.12185088 s</td>
<td>3.13261568 s</td>
<td>83.88608 ms</td>
</tr>
<tr>
<td>0xF</td>
<td>6m33.71923456s</td>
<td>24.60745216 s</td>
<td>1.34217728 s</td>
</tr>
<tr>
<td>Max val</td>
<td>1h38m11.27737356 s</td>
<td>6m8.20483584 s</td>
<td></td>
</tr>
</tbody>
</table>

The “C” code on the next pages contains GetTimeTag(), a function to convert a binary time tag into TIME, i.e. milliseconds and nanoseconds, and TimeToStr(), that formats TIME into a character string.
typedef packed struct {
  UINT32 ms;    /* ..................... Millisecond part */
  UINT32 ns;    /* ...................... Nanosecond part */
  short neg;    /* .......................... Negate flag */
} TIME;

#define X_US  12  /* ....... Where * 1000 is used to get nS */
#define X_10US 15  /* ...... Where * 10000 is used to get nS */
#define MIN_ADJ 50  /* ............ 30 nS. Added to all tags. */
#define XMEM_ADJ 20 /* Another 20 nS. added to all XMEM tags. */
#define NT_NEG    0x1
#define NT_MAXTAG 0x2

static const UINT32 ScaleArr[16] = {
  0x00000014L, /* [ 0]    0 ms     20 ns */
  0x000000028L, /* [ 1]    0 ms     40 ns */
  0x000000050L, /* [ 2]    0 ms     80 ns */
  0x0000000A0L, /* [ 3]    0 ms    160 ns */
  0x000000140L, /* [ 4]    0 ms    320 ns */
  0x000000280L, /* [ 5]    0 ms    640 ns */
  0x000000500L, /* [ 6]    0 ms   1280 ns */
  0x000000A00L, /* [ 7]    0 ms   2560 ns */
  0x000001400L, /* [ 8]    0 ms   5120 ns */
  0x000005000L, /* [ 9]    0 ms  20480 ns */
  0x000014000L, /*[10]    0 ms  81920 ns */
  0x000050000L, /*[11]    0 ms 327680 ns */
  0x00000051EL, /*[12]    1 ms 310  us (Should be 310.72 us) */
  0x0000028F6L, /*[13]   10 ms 486  us (Should be 485.76 us) */
  0x0000147AEel, /*[14]   83 ms 886  us (should be 886.08 us) */
  0x00020C49L  /*[15] 1342 ms 170  us (should be 177.28 us) */
};

static const UINT32 ScaleOffset[16] = {
  0x00000000L, /* [ 0] */
  0x000000014L, /* [ 1] */
  0x00000003CL, /* [ 2] */
  0x00000008CL, /* [ 3] */
  0x000000012CL, /* [ 4] */
  0x000000026CL, /* [ 5] */
  0x00000004ECL, /* [ 6] */
  0x00000009ECL, /* [ 7] */
  0x00000003ECL, /* [ 8] */
  0x000000027ECL, /* [ 9] */
  0x00000077ECL, /*[10] */
  0x00000017ECL, /*[11] */
  0x0000001B8L, /*[12] (Actual value: 0x0006B7ECL) */
  0x000000067DL, /*[13] (Actual value: 0x001AB7ECL) */
  0x000002FCCel, /*[14] (Actual value: 0x00BAB7ECL) */
  0x00000258CL /* [15] (Actual value: 0x05BAB7ECL) */
};
/*== Get Time Tag ========================================
Function : Convert a time tag to TIME format.
Parameters: See below.
Returns : Nothing.
*/

void GetTimeTag (TIME *Time, /* ........ Returned time tag */
                UINT16 BinTag, /* ...... 16 [or 12] bit tag */
                BOOLEAN HiRes) /* ..... True for 16 bit tag */
{

    UINT32 TagnS, Offs;
    UINT16 Tag, Scale;

    Scale = BinTag >> 12;    /* .............. Get scale factor */
    Tag = BinTag & 0x0FFF;  /* ................. Get tag value */
    if (!HiRes) Tag >>= 4;  /* Adjust tag value in 12-bit mode */

    if (Scale == 0 && Tag == 0) { /* ............... Null tag? */
        Time->ms = 0;   /* ..... Yes. Return minimum value */
        Time->ns = MIN_ADJ;
        Time->neg = Null;
        return;
    }

    TagnS = ScaleArr[Scale] * Tag;
    Offs = ScaleOffset[Scale] <<= (HiRes? 12: 8);
    TagnS += Offs;

    if (Scale < X_US) { /* In scale area where no res. lost? */
        TagnS += MIN_ADJ;

        /* .. If the trace file originates from an XMEM targets,
        the flag NF_XMEMADJ must be checked to ensure
correct calculations: */

        if (Header->nhLastRunFlags & NF_XMEMADJ) TagnS+= XMEM_ADJ;
        Time->ms = TagnS / 1000000L;
        Time->ns = TagnS % 1000000L;
    }
    else if (Scale < X_10US) { /* ...... 1 uS resolution area? */
        Time->ms = TagnS / 1000L;
        Time->ns = TagnS % 1000L * 1000L;
    }
    else { /* ......................... 10 uS resolution area? */
        Time->ms = TagnS / 100L;
        Time->ns = TagnS % 100L * 10000L;
    }

    Time->neg = Null;
    if (Scale == 0xF &&
        ((Tag == 0x00FF && !HiRes) || (Tag == 0x0FFF && HiRes))) {  
        Time->neg |= NT_MAXTAG;
    }
}
13 VMETRO VT100 Terminal Emulator
/*====== T i m e T o S t r =====================================
Function

: Make a string from a given TIME struct. Format the
time with 5 significant decimals.
Resolution goes down to 0.1 us.

Parameters: None
Returns
: Nothing.
*/
void TimeToStr (char *Buffer, TIME Time)
{
UINT32 hrs, mins, secs, msecs, usecs, nsecs;
char *p;
if (Time.neg & NT_ASYNCQ) return;
msecs = Time.ms % 1000; /*...... Calculate milli-seconds
secs =
Time.ms / 1000; /* ........... Calculate seconds
mins = secs / 60;
/*............ Calculate minutes
hrs = mins / 60;
if (hrs) {
mins %= 60;
}
else if (mins) {
secs %= 60;
/* ................. And seconds
mins %= 60;
}
else {
usecs = Time.ns / 1000; /* .. Calculate micro seconds
nsecs = Time.ns % 1000; /* ... Calculate nano seconds
}

*/
*/
*/

*/

*/
*/

strcpy (Buffer, (Time.neg & NT_MAXTAG) ? ">" : " ");
Buffer++;
if (hrs)
/* ............... [
xxhxx] */
sprintf (Buffer, "
%2luh%02lu", hrs, mins);
else if (mins)
/* ............... [ xxmxx.x] */
sprintf (Buffer," %2lum%02lu.%01lu",mins,secs, msecs/100);
else if (secs >= 10)
/* ............... [ xx.xxs] */
sprintf (Buffer, " %2lu.%02lus", secs, msecs/10);
else if (Time.ms >= 1000)
/* ............. 1 - 9
sec */
sprintf(Buffer, " %1lu.%03lus",secs,msecs);
else if (Time.ms >= 100)
/* ............. 100- 999 ms */
sprintf(Buffer, "%3lu.%02lums",msecs,usecs/10);
else if (Time.ms >= 10)
/* ............. 10 - 99 ms */
sprintf(Buffer, "%2lu.%03lums",msecs,usecs);
else if (Time.ms >= 1)
/* ............. 1 - 9
ms */
sprintf(Buffer, "%1lu.%03lu%1lums",msecs,usecs,nsecs/100);
else if (Time.ns >= 100000) /* ............. 100- 999 us */
sprintf(Buffer, "%3lu.%02luus",usecs,nsecs/10);
else if (Time.ns > 999)
/* ............. 10 - 99 us */
sprintf(Buffer, " %2lu.%02luus",usecs,nsecs/10);
else
sprintf(Buffer, "
%3uns",nsecs);
if (Time.neg & NT_NEG) {
for (p = Buffer; *p && *p == SPACE; *p++);
p--; *p = '-';
}
}

VMETRO

VBT-325 User’s Manual

223


Details of the Am, Irq, Str, Bg, St2 and St3 bytes

The signal names conforms mostly to VMEbus signal names. Signal names ending in an asterix '*', are active low. Signal names having an underscore as the first letter are internally generated signals. An explanation of these will be found at the end of this section. Bit #0 is the least significant bit, bit #7 is the most significant bit.

<table>
<thead>
<tr>
<th>Bit#</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Am</td>
<td>IAIO*</td>
<td>AM5</td>
<td>AM4</td>
<td>AM3</td>
<td>SYSRES*</td>
<td>AM2</td>
<td>AM1</td>
<td>AM0</td>
</tr>
<tr>
<td>Bit#</td>
<td>7</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Irq</td>
<td>IACK*</td>
<td>IRQ7*</td>
<td>IRQ6*</td>
<td>IRQ5*</td>
<td>IRQ4*</td>
<td>IRQ3*</td>
<td>IRQ2*</td>
<td>IRQ1*</td>
</tr>
<tr>
<td>Bit#</td>
<td>7</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Str</td>
<td>_SAMP</td>
<td>WR*</td>
<td>_RMW</td>
<td>RETRY*</td>
<td>_BGVAL*</td>
<td>_MLEV0</td>
<td>_MLEV1</td>
<td></td>
</tr>
<tr>
<td>Bit#</td>
<td>7</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Bg</td>
<td>BR3*</td>
<td>BR2*</td>
<td>BR1*</td>
<td>BR0*</td>
<td>BG3*</td>
<td>BG2*</td>
<td>BG1*</td>
<td>BG0*</td>
</tr>
<tr>
<td>Bit#</td>
<td>7</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>St2</td>
<td>SYSF*</td>
<td>ACF*</td>
<td>BCLR*</td>
<td>BBSY*</td>
<td>_A01</td>
<td>LWORD*</td>
<td>DS1*</td>
<td>DS0*</td>
</tr>
<tr>
<td>Bit#</td>
<td>7</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>St3</td>
<td>SYSCLK</td>
<td>_XTRG1*</td>
<td>_XTRG2*</td>
<td>_XTRG3*</td>
<td>AS*</td>
<td>DTACK*</td>
<td>BERR*</td>
<td></td>
</tr>
</tbody>
</table>

Details on internally generated bits

_SAMP Indicates the sampling mode for the current trace line. The bit is set to 1 for state mode, and to 0 for timing mode. Useful for traces sampled in mixed state and timing mode. Check the Header->nlLastRunFlags bit NF_TRCMIXED. A one in this bit indicates mixed state and timing mode.

_RMW Indicates that the current sample contains a read-modify-write cycle.

_BGVAL* If 0, the contents in the _MLEV1 and _MLEV0 bits are valid. These two bits give granted the bus level of the sample, ranging from 0..3 If _BGVAL* is 1, the tracer could not see the bus grants when the trace was sampled.

_A01 A copy of the A01 bit, also found in the cAddr[0] byte.

_XTRG1* Cross trigger from S/TIM200-PB or TIMBAT-PB.

_XTRG2* Cross trigger from VBAT-PB or VBAT part of if TIMBAT-PB, or XMEM.

_XTRG3* Cross trigger from VBT-325C P2 tracer, i.e VSB, SCSI etc.
Differences for the XMEM-PB VME

Some signals are different for the XMEM-PB VME target. The differences are indicated below:

In 12-bit tag mode, the cross trigger from the VBT-325 VME part, \texttt{XVBTV*}, replaces \texttt{EXT1}:

<table>
<thead>
<tr>
<th>Bit#</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>TagL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>XVBTV*</td>
<td></td>
</tr>
</tbody>
</table>

When the time tag is unused, as in timing mode, the SERCLK and the SERDAT signals are sampled as shown here:

<table>
<thead>
<tr>
<th>Bit#</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>TagU</td>
<td>SERCLK</td>
<td>SERDAT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bit#</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>St3</td>
<td>SYSCLK</td>
<td>XP2*</td>
<td>AS*</td>
<td>DTACK*</td>
<td>BERR*</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\texttt{XP2*} is the cross trigger from the XMEM-P2 part. Blank spaces are unused bits, which value should be ignored.

Trace Data Line format, VBT-325 and XMEM-PB VSB targets

For the VSB targets, the structure of each trace data line is as follows.

```c
typedef unsigned char BYTE;
typedef unsigned long UINT32;
typedef packed struct _VSBTRACE {
  union {
    BYTE cAD [4]; /* ... LSB in cAD[0], MSB in cAD[3] */
    UINT32 AD;    /* ..... Multiplexed Address & Data */
  }
  BYTE TagL, TagU; /* ............ Time-tag. See below */
  BYTE St4, St5;   /* ............. Strobes. See below */
} VSBTRACE;
```

\textit{Intel format}  
As for the VME target, the trace address/data is stored as an \textit{Intel format} long integer. The main difference is that the VBS bus is multiplexed, i.e the \texttt{AD(31:0)} holds data when the \texttt{DS*=0} and an address when \texttt{DS*=1}.

\textit{Time tag}  
The time tag bytes does also follow the same format with the following exceptions: In 12-bit tag mode, or when not used at all [as in Timing mode], the following VSBbus signals are mapped into the TagL and TagU bytes:

<table>
<thead>
<tr>
<th>Bit#</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>TagL</td>
<td>BGIO*</td>
<td>BUSY*</td>
<td>BREQ*</td>
<td>WAIT*</td>
<td>ERR*</td>
<td>PAS*</td>
<td>ACK*</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bit#</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>TagU</td>
<td>AC*</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Details of St4 and St5 bytes

<table>
<thead>
<tr>
<th>Bit#</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>St4</td>
<td>_BLK</td>
<td>ARB*</td>
<td>DS*</td>
<td>WR*</td>
<td>SIZE1*</td>
<td>SIZE0*</td>
<td>SPACE1*</td>
<td>SPACE0*</td>
</tr>
<tr>
<td>St5</td>
<td>7</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>ExtInp</td>
<td>_SAM</td>
<td>CACHE*</td>
<td>IRQ*</td>
<td>_TRG</td>
<td>LOCK*</td>
<td>ASACK1*</td>
<td>ASACK0*</td>
</tr>
</tbody>
</table>

_BLK
Set to 1 when a the trace line contains data belonging to a VSB block cycle.

_SAMP
As for the VBT-325 VME part.

---

Trace Data Line format, VBT-325 and XMEMPB SCSI targets

For the SCSI targets, the structure of each trace data line is as follows.

typedef unsigned char BYTE;
typedef unsigned long UINT32;

typedef packed struct _SCSITRACE {
    BYTE TagL, TagU;          /* ..... Time-tag. See below */
    BYTE Da0, Da1;
    BYTE St6, St7, St8, St9;
} SCSITRACE;

16-bit tag
The SCSI target will always use the 16-bit time tag format:

<table>
<thead>
<tr>
<th>Bit#</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>TagL</td>
<td>7</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Tag[7:0]</td>
<td>Tag[3:0]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TagU</td>
<td>7</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Details, Da0 and Da1 bytes:

<table>
<thead>
<tr>
<th>Bit#</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Da0</td>
<td>7</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>DB4*</td>
<td>DB3*</td>
<td>DB2*</td>
<td>DB0*</td>
<td>DB1*</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Da1</td>
<td>7</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>DBP*</td>
<td>DB7*</td>
<td>DB6*</td>
<td>DB5*</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Details, Da6, St7, St8 and St9 bytes:

<table>
<thead>
<tr>
<th>Bit#</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>St6</td>
<td>7</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>REQ*</td>
<td>DBP1*</td>
<td>I/O*</td>
<td>DB15*</td>
<td>DB10*</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>St7</td>
<td>7</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>DB11*</td>
<td>BUSY*</td>
<td>ACK*</td>
<td>RST*</td>
<td>DB9*</td>
<td>DB8*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>St8</td>
<td>7</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>SEL*</td>
<td>MSG*</td>
<td>TERMPWR</td>
<td>C/D*</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>St9</td>
<td>7</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>_SAM</td>
<td>DB14*</td>
<td>ATN*</td>
<td>DP12*</td>
<td>DB13*</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Blank fields indicates unassigned bits. Italic fields contains wide-SCSI2 bits.
_SAMP works as for the VBT-325 VME target.
The Simulator diskette contains a simulator for DOS which truly represents the user-interface of the VBT-325 product, including the piggyback options and the Futurebus+ option. Demonstration trace files captured in real bus systems are also included, as well as a self-running demo.

The Simulator serves three purposes:

- **Demonstration/Training** of product capabilities without having the actual product at hand.
- **Reviewing 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>`.)
- **Printing trace files** loaded into the simulator to file in ASCII format with the command: `Trace/Print To File`
- **Creating 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.

**Installation**

To use the Simulator you must install it to a hard disk. Put the diskette in drive A: and type:

```
install
```

You will then be asked to enter the path to where to install the program. You may also give the path directly at the command line, e.g. if you type

```
install c:
```

the directory `C:\VMETRO` will be created and the program will be put there.

Note that the Simulator requires about **2.5 Mbytes** of hard disk space.

Start the simulator by changing the current directory to the simulator directory and type:

```
vmetro
```

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.
Self-running Demo

After starting the program, Type <Cr> twice to enter the main Setup screen. Execute the following command to start a self-running demo:

Macro/Play

The demo may be stopped at any time by typing ^C, i.e. Ctrl and C.

Simulate bus activity

When operating the simulator, one may emulate actual analyzing activity to get a trigger, to see a trace display etc. Three commands are used to simulate such activity:

Trace/Simulate Trigger (or ^T)

Simulate TRIGGER FOUND. Reads the demonstration trace files described below if they exist or creates random trace data.

Vbat/Simulate Violation (or ^V)

Simulate VIOLATIONS FOUND on the bus when the VBAT is active.

Session/Simulate Count (or ^A)

Simulate STATISTICS UPDATE when the bus statistics is running.
Trace Files

Trace files captured in real bus environments by the VBT-325 are included to show examples of the analyzing capabilities. When the user simulates a trigger with <Ctrl-T> or executes "Simulate Trace Full" from the Trace menu, the corresponding file from the table above will be loaded. The simulator will generate a random trace if the files for the selected target does not exist.

View Trace Files The simulator is a powerful tool to use on PC for viewing/post-processing captured trace-files. When a trace is captured on a VMETRO analyzer, a terminal emulator can be used (like VMETRO’s vt100 emulator, ‘Terminal’ in Windows etc.) to dump the trace, or part of it, to a PC. The file contains all data needed to redisplay the trace as first captured. The file can be reloaded to the analyzer, or loaded into the simulator with the command Trace/Load From PC/Host located in the trace display menu.

ASCII trace files When a trace is loaded into the simulator, there is yet another feature that will ease post processing of traces. The command Trace/Print To File, also located in the trace display menu, will print the trace in ASCII characters to a file. The new file can easily be loaded into a text editor or a spread-sheet program, such as Microsoft Excel®, or Lotus 123®.
Included Trace example files

<table>
<thead>
<tr>
<th>Product</th>
<th>Target</th>
<th>Sampling Mode : Timing</th>
<th>Sampling Mode : State</th>
</tr>
</thead>
<tbody>
<tr>
<td>VBT-325B/C</td>
<td>VME</td>
<td>VMET.TRC</td>
<td>VMES.TRC</td>
</tr>
<tr>
<td>VBT-325C</td>
<td>VSB</td>
<td>VSBT.TRC</td>
<td>VSBS.TRC</td>
</tr>
<tr>
<td>VBT-325C</td>
<td>SCSI</td>
<td>SCSIT.TRC</td>
<td>SCSIS.TRC</td>
</tr>
<tr>
<td>VBT-325C</td>
<td>VXI_P2ac</td>
<td>VXI_P2AT.TRC</td>
<td>VXI_P2AS.TRC</td>
</tr>
<tr>
<td>VBT-325C</td>
<td>P2ac</td>
<td>P2ACT.TRC</td>
<td>P2ACS.TRC</td>
</tr>
<tr>
<td>TIMBAT-PB</td>
<td>TBAT_VME</td>
<td>TBAT_VME.TR C</td>
<td>N/A</td>
</tr>
<tr>
<td>TIM200-PB</td>
<td>TIM_VME</td>
<td>TIM_VME.TR C</td>
<td>N/A</td>
</tr>
<tr>
<td>TIM200-PB</td>
<td>TIM_VSB</td>
<td>TIM_VSB.TR C</td>
<td>N/A</td>
</tr>
<tr>
<td>TIM200-PB</td>
<td>TIM_SCSI</td>
<td>TIM_SCSI.TR C</td>
<td>N/A</td>
</tr>
<tr>
<td>FBT-625</td>
<td>FB+Phase</td>
<td>FBPHASSET.TR C</td>
<td>FBPHASES.TR C</td>
</tr>
<tr>
<td>FBT-625</td>
<td>FB+Demux</td>
<td>N/A</td>
<td>FBDEMUXS.TR C</td>
</tr>
<tr>
<td>FTIM200-PB</td>
<td>TIM_FB+</td>
<td>TIM_FB.TR C</td>
<td>N/A</td>
</tr>
</tbody>
</table>

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.

**RAM\(i\)j.SIM** Setup files. One of these files exist for each combination of a main product and a piggyback option. The \(i\) is the number of the Main product, and the \(j\) is the number of the piggyback option.

**^T** Files used when the user simulates trace buffer full. [type ^T or execute Trace/Simulate Trigger] are shown in the table above. The simulator will generate a random trace if the files for the selected target does not exist.
## Command Index

### Setup Screen

<table>
<thead>
<tr>
<th>Trace</th>
<th>Edit</th>
<th>Target</th>
<th>Statistics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Run VME</td>
<td>Event Patterns</td>
<td>(See page 126)</td>
<td>(See page 233)</td>
</tr>
<tr>
<td>Run Multiple</td>
<td>Sequencer</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Halt VME</td>
<td>Trigger Position</td>
<td>*Start of Trace</td>
<td></td>
</tr>
<tr>
<td>Halt All</td>
<td></td>
<td>25% of Trace</td>
<td></td>
</tr>
<tr>
<td>Show</td>
<td></td>
<td>Middle of Trace</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>75% of Trace</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>End of Trace</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sampling Mode</td>
<td>*State 125</td>
<td></td>
</tr>
<tr>
<td></td>
<td>50 MHz Timing Options</td>
<td>State Options</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>50 MHz Timing</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>25 MHz Timing</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>6.25 MHz Timing</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.56 MHz Timing</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>781.3 KHz Timing</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>390.6 KHz Timing</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>195.3 KHz Timing</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>97.7 KHz Timing</td>
<td></td>
</tr>
</tbody>
</table>

### Setups

<table>
<thead>
<tr>
<th>Initialize</th>
<th>Transparent Mode 132</th>
</tr>
</thead>
<tbody>
<tr>
<td>Store</td>
<td>Transparent Mode Options 132</td>
</tr>
<tr>
<td>Delete</td>
<td>Trigger Output Options 132</td>
</tr>
<tr>
<td></td>
<td>LED Display 133</td>
</tr>
<tr>
<td></td>
<td>Serial Ports</td>
</tr>
<tr>
<td></td>
<td>Terminal Port 134</td>
</tr>
<tr>
<td></td>
<td>Host Port</td>
</tr>
<tr>
<td></td>
<td>Assert VMEbus SYSRES* 134</td>
</tr>
<tr>
<td></td>
<td>Selftest 134</td>
</tr>
<tr>
<td></td>
<td>Reset Analyzer 134</td>
</tr>
</tbody>
</table>

### Utilities

<table>
<thead>
<tr>
<th>Specials</th>
<th>FW Version 135</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ECO Level</td>
</tr>
<tr>
<td></td>
<td>Set Time And Date</td>
</tr>
</tbody>
</table>

---

*INDEXES*
Indexes

Trace Screen

<table>
<thead>
<tr>
<th>Trace</th>
<th>Search</th>
<th>Jump</th>
<th>Count 140</th>
<th>Window</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dump to PC/Host 137</td>
<td>Edit Pattern 100</td>
<td>First Line 139</td>
<td>New 142</td>
<td></td>
</tr>
<tr>
<td>Load From PC/Host 138</td>
<td>Next Match F3 101</td>
<td>Last Line 139</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Print 138</td>
<td>Extract 102</td>
<td>Trigger Line 140</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Print To File 229</td>
<td>Close pattern Window</td>
<td>Marker Y 140</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Statistics Screen

<table>
<thead>
<tr>
<th>Session</th>
<th>Function</th>
<th>Options</th>
<th>Target</th>
<th>Utilities</th>
<th>Quit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Run 143</td>
<td>*Event Counting 144</td>
<td>Standard Histograms 145</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Continue 143</td>
<td>Bus Utilization 144</td>
<td>Time History Curves</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Halt 144</td>
<td>Bus Transfer rate 145</td>
<td>Count Options 147</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>*Immediate Start 144</td>
<td>Show 145</td>
<td>Bar Markers</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Start on Trigger 144</td>
<td>Reset</td>
<td>History Curve Options 146</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Maximum Scale 147</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Select Events 148</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Sampling Mode *State 149</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>50 MHz Timing Options</td>
<td>50 Mhz Timing</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>25 MHz Timing</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>6.25 MHz Timing</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.56 MHz Timing</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>781.3 KHz Timing</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>390.6 KHz Timing</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>195.3 KHz Timing</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>97.7 KHz Timing</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Index

1
12V, 19
  Jumper J67, 19

5
5V, 18

A
Accelerator keys, 46
Accumulate mode, 116
Actions, Sequencer, 77
Adding Event patterns, 61
Adding field columns, 60
Annunciators, 123
ANSI.SYS, 208
Applications, 8
ASCII terminal, 5
Asynchronous sampling, 29
Attributes
  Address, 65
  Data, 65

B
Backplane
  VMEbus, 16
  VSB, 16
Bar markers, 113
Binary details, 66
Block cycle, 34
Bookmarks, 95
BR* and IRQ* formatting, 64
Brackets, use of, 78
Bus Grant
  Statistics, 117
  VMEbus, 16
Bus Tracer Chip, 6
Bus Transfer rate, 111; 120
Bus Utilization, 111; 118

C
Cable
  Coax, 12
  External power, 12
  PC, 12
  PC as terminal, 22
  RS-232, 12
  Temperature sensor, 12
  Terminal, 12; 22
  Transparent mode, 12
Cables, 12
  SCSI VSC-3, 196
  SCSI VSC-8, 195
Carrier
  Piggyback, 10
Change Scale, 120
Clearing field contents, 59
COM1 port, 208
Command
  Absolute/relative time tags, 141
  Assert SYSRES*, 134
  Bus Transfer rate, 145
  Bus Utilization, 144
  Count, 140
  Decoding and Formatting, 141
  Dump trace to PC, 137
  Edge Options, 140
  Edit Event Patterns, 124
  Edit Sampling mode, 125
  Edit Sequencer, 125
  Edit Trigger position, 125
  Event Counting, 144
  First Line, 139
  Formatting Template, 142
  Halt, 124
  Halt All, 124
  Help, 135; 142
  Jump Line Number, 140
  Jump Marker Y, 140
  Jump Marker Z, 140
  Last Line, 140
  LED-display, 133
  Load trace from PC/Host, 138
Quit (from Trace screen), 142
Reset Analyzer, 134
Run, 122
Run Multiple, 122
Sampling Status, 123
Selftest, 134
Serial Ports, 134
Session Continue, 143
Session Halt, 144
Session Run, 143
Setups Delete, 128
Setups Dump, 130
Setups Initialize, 128
Setups Load, 129
Setups Store, 128
Show, 124
Specials, 135
State Sampling Options, 125
Statistics, 127
Target P2-bus, 126
Target Reconfigure, 127
Target TIM200-bus, 127
Target VME, 126
Target XMEM P2-bus, 127
Target XMEM VME, 127
Time/Div, 141
Trace print, 138
Transparent mode, 132
Transparent mode options, 132
Trigger Line, 140
Trigger output options, 132
Window, 142
Consumption
Power, 18
Control signals, 55
Copper tracks, cutting, 16
Count command, 140
Count operator, 81
Counter driven Statistics, 111
Counters
Statistics, 32
VMEbus, 32
Counting
Bus Transfer rate, 145
Bus utilization, 144
Events, 144
Cross trigger, 37; 199

Cross-triggering, 88

D
Daisy-Chains
VMEbus, 16
Debugging, 8
Delay operator, 81
Delete setups, 128
Dialog boxes, 47
Display the trace buffer, 124
DTACK, 34
dump setups to PC/Host, 130
dump Trace, 110
dump trace to PC, 137

E
Edge jumping, 109
Edit
Event Patterns, 124
Sampling mode, 125
Search pattern, 99
Sequencer, 125
State Sampling Options, 125
Trigger position, 125
Edit event patterns, 56
Edit menu, 124
Else operator, 80
Elsif operator, 80
Event Counting, 111; 117
Event patterns, 54
Adding, 61
Edit, 56
Rename, 61
Events
Counting (statistics), 144
Number of usable, 79
XMEM325-PB, 79
Extender board, 18
External inputs, 37
External Power supply, 19
External signal, 37
Extract trace data, 102
Indexes

F
Field options, 57
File format, 215
Firmware
  Upgrade, 168
Flash Memory, 168
Format
  Trace file, 215
Format menu, 141
Front-panel display, 133
Function keys, 47
Function menu, 144
Futurebus+, 7; 39

G
Generator
  Pattern, 9
  Stimuli, 9
Goto operator, 81
Grey Text, 45

H
Halt, 124
Halt All, 124
Halt operator, 82
Handshake, 23
Help command, 142
Hiding field columns, 59

I
IACK, 34
If operator, 80
Indents, Sequencer, 79
Initialize setups, 128
Inputs
  External, 37
  IRQ, 34
Isolation of P2A/C pins, 196

J
Jump
  To first line, 139
  To last line, 140
  To line number, 140
  To marker Y, 140
  To marker Z, 140
  To trigger line, 140
Jump menu, 139
Jumper settings, 178
  User Defined P2A/C, 199
Jumpers
  Power supply, 20
  Settings, 14
  Spare, 14

K
Keyboard
  Template, 49

L
Latching
  Bus grant, 33
  Bus request, 33
LED-display, 133
Line number, Sequencer, 78
Load setups from PC/HOST, 129
Load trace from PC/Host, 138

M
Main menu, 121
Markers, 95
  Commands, 95
  Master/slave, 9
Maximum scale, 112
Measure
  Elapsed time, 28
Memory
  Non-volatile, 26
  Trace buffer, 31
Menu
  Edit, 124
  Format, 141
  Function (in Statistics screen), 144
  Jump (in Trace screen), 139
  Options (in Statistics screen), 145
Indexes

Pull down, 45  
Search, 139  
Session (in Statistics screen), 143  
Setups, 128  
Start-up, 25  
Statistics, 143  
Target, 126  
Target (in Statistics screen), 149  
Target (User Defined P2-tracer), 200  
Trace, 122  
Trace (in Trace screen), 136  
Trace display, 136  
Utilities, 132; 149  
menu selection, 46  
Message line, 49  
Mode  
  sequencer, 70  
  single event, 68  
  Transparent, 23  
  Twin, 27  
  Wide, 27  
Modules  
  Piggyback, 9  
Mounting XMEM325-PB, 41  

N  
Non-volatile memory, 26  
NOT operator, 55; 66  
Notation  
  Sequencer, 78  
Numeric keypad, 48  

O  
Online help, 135  
Operator  
  Count, 81  
  Delay, 81  
  Else, 80  
  Elsif, 80  
  Goto, 81  
  Halt, 82  
  If, 80  
  NOT, 55  
  Sampling, 79  
  Store, 80  
  Trigger, 82  
Operators, Sequencer, 79  
Options  
  Edge, 140  
  Statistics, 145  
  Transparent mode, 132  
  Trigger output, 132  
Outside range, 66  

P  
P2 bus  
  User defined, 17  
Pattern generator, 9  
Performance analysis, 8  
Piggyback, 9  
  Carrier, 10  
  Mounting XMEM325-PB, 41  
  Removal, 42  
  STIM200-PB, 9  
  TIM200-PB, 9  
  TIMBAT-PB, 10  
  VBAT64-PB, 10  
  VBAT-PB, 9  
  VDRIVE-PB, 9  
  XMEM325-PB, 9  
Piggyback modules, 9  
Pinout P1/P2, 192  
Ports  
  Baud rate, 24  
  Hardware handshake, 23  
  Host/Printer, 21  
  Serial, 21  
  Terminal, 21  
  Terminal type, 25  
Position, Trigger, 31  
Power supply  
  12V, 19  
  5V, 18  
  Consumption, 18  
  External, 19  
  Jumpers, 20  
  Power supply, 18  
Print, 21; 138  
  Trace, 21  
  Trace to File, 227
Indexes

Print trace, 136
Product overview, 5
Program
  Sequencer, 77
Pull down menu, 45

Q
Quit command, 142

R
Range, 30
  Edit, 65
    Outside, 30; 66
  VME64, 31
Reconfigure target, 127
Refresh screen, 49
Removal, 42
Removal tool, 42
Reset Mode, 115
Resetting the VBT-325, 134
RMW cycle, 34
RS-232
  Cable, 12
    Hardware handshake, 23
  Port, 5
Run, 122
Run multiple, 122
Running selftests, 134

S
Sampling
  Asynchronous, 28
  Block cycle, 34
  Data cycle, 34
  Methods, 33
  Registers, 28
  RMW cycle, 34
  SCSI state, 36
  State, 28
  Synchronous, 28
  Timing, 28
  User-defined P2, 37
  VSB state, 35
  VXI, 37
Sampling Clock
    Select for User Defined P2 A/C, 199
Sampling operator, 79
Sampling rate, 29
Sampling Status, 123
Screen
  Refresh, 49
  Setup, 51
    Statistics, 53; 112; 143
    Trace display, 52; 136
Screen update rate, 115
Script files, 46
Script language, 211
SCSI, 36
    Cable VSC-3, 196
    Cable VSC-8, 195
  Connection on P2, 193
  Differential, 195
  Pinout, 36
Search
  Edit Pattern, 99
  Extract, 102
    Find Next [F3], 101
Search menu, 139
Select events, 112
Select signals, 91
Select slot, 16
Selftests, 134
Sequencer, 31; 68
  Actions, 77
  Brackets, 78
  Count, 81
  Delay, 81
  Else, 80
  Elsif, 80
  Goto, 81
  Halt, 82
  If, 80
  Implicit actions, 82
  Indents, 79
  Line number, 78
  Loose sequence, 83
  Notation, 78
  Operators, 79
  Program, 77
  Programming examples, 84
<table>
<thead>
<tr>
<th>Sampling, 79</th>
<th>Polarity, 55</th>
</tr>
</thead>
<tbody>
<tr>
<td>State number, 78</td>
<td>Templates, 63</td>
</tr>
<tr>
<td>Store, 80</td>
<td>Simulator, 14; 227</td>
</tr>
<tr>
<td>Tight sequence, 83</td>
<td>Trace review, 110</td>
</tr>
<tr>
<td>Transitions, 77</td>
<td>Single event mode, 68</td>
</tr>
<tr>
<td>Trigger, 82</td>
<td>Slot selection</td>
</tr>
<tr>
<td>Sequencer mode, 70</td>
<td>P2 bus, 198</td>
</tr>
<tr>
<td>Serial ports, 21</td>
<td>VMEbus, 16</td>
</tr>
<tr>
<td>Session</td>
<td>Spare jumpers, 14</td>
</tr>
<tr>
<td>Continue, 143</td>
<td>SSBLT cycle, 35</td>
</tr>
<tr>
<td>Halt, 144</td>
<td>Start-up menu, 25</td>
</tr>
<tr>
<td>Immediate Start, 144</td>
<td>State analysis, 28</td>
</tr>
<tr>
<td>Run, 143</td>
<td>State number, Sequencer, 78</td>
</tr>
<tr>
<td>Start on Trigger, 144</td>
<td>State Sampling Options, 125</td>
</tr>
<tr>
<td>Session menu, 143</td>
<td>Static electricity</td>
</tr>
<tr>
<td>Settings</td>
<td>Conductive wrist-chain, 13</td>
</tr>
<tr>
<td>Baud rate, 134</td>
<td>Precautions, 13; 41</td>
</tr>
<tr>
<td>Jumper, 14</td>
<td>Statistics</td>
</tr>
<tr>
<td>Setup screen, 51; 121</td>
<td>Accumulate Mode, 116</td>
</tr>
<tr>
<td>Setups</td>
<td>Bar markers, 113</td>
</tr>
<tr>
<td>Delete, 128</td>
<td>Bus Level, 117</td>
</tr>
<tr>
<td>Dump to PC/Host, 130</td>
<td>Bus Transfer rate, 111; 120</td>
</tr>
<tr>
<td>Initialize, 128</td>
<td>Bus Utilization, 111; 118</td>
</tr>
<tr>
<td>Load from PC/Host, 129</td>
<td>Change Scale, 120</td>
</tr>
<tr>
<td>Store, 128</td>
<td>Counter driven, 111</td>
</tr>
<tr>
<td>Setups menu, 128</td>
<td>Counters, 114</td>
</tr>
<tr>
<td>Show</td>
<td>Event Counting, 111; 117</td>
</tr>
<tr>
<td>Trace, 124</td>
<td>Maximum scale, 112</td>
</tr>
<tr>
<td>Trace for another target, 124</td>
<td>Reset Mode, 115</td>
</tr>
<tr>
<td>Signal inputs, 37</td>
<td>Screen update rate, 115</td>
</tr>
<tr>
<td>Signal polarity, 55</td>
<td>Select events, 112</td>
</tr>
<tr>
<td>Signal selection, 91</td>
<td>Time History Curves, 113</td>
</tr>
<tr>
<td>Signals</td>
<td>Trace driven, 111</td>
</tr>
<tr>
<td>Adding, 60</td>
<td>Statistics command, 127</td>
</tr>
<tr>
<td>Address options, 65</td>
<td>Statistics counters, 32</td>
</tr>
<tr>
<td>Binary details, 66</td>
<td>Statistics menu, 143</td>
</tr>
<tr>
<td>Busses, 55</td>
<td>Statistics screen, 53; 112</td>
</tr>
<tr>
<td>Clearing contents of, 59</td>
<td>Statistics.screen, 143</td>
</tr>
<tr>
<td>Control, 55</td>
<td>Stimuli generator, 9</td>
</tr>
<tr>
<td>Data options, 65</td>
<td>Store operator, 80</td>
</tr>
<tr>
<td>Define (User Defined P2-tracer), 202</td>
<td>Store setups, 128</td>
</tr>
<tr>
<td>Delete (User Defined P2-tracer), 205</td>
<td>Synchronous sampling, 28</td>
</tr>
<tr>
<td>Edit BgL, 64</td>
<td>SYSRES*, 134</td>
</tr>
<tr>
<td>Grouped, 62</td>
<td>System controller, 9</td>
</tr>
<tr>
<td>Hiding, 59</td>
<td></td>
</tr>
<tr>
<td>Individual, 62</td>
<td></td>
</tr>
<tr>
<td>Options, 57</td>
<td></td>
</tr>
</tbody>
</table>
| T | Tuning, 8  
|   | Twin mode, 27  

| U | Upgrade firmware, 168  
|   | User defined  
|   | Cross trigger, 199  
|   | Define Signal fields, 202  
|   | Delete signals, 205  
|   | Display format, 202  
|   | Jumper settings, 199  
|   | Mnemonics, 204  
|   | P2 bus, 17  
|   | P2A/C Slot selection, 198  
|   | P2A/C Tracer, 198  
|   | Sampling, 37  
|   | Sampling Clock select, 199  
|   | Using time-tags, 206  
|   | User Defined Tracer, 198  
|   | Utilities, 149  
|   | Utilities menu, 132  

| V | VIC068 chip, 9  
|   | VME64, 34; 67; 152; 157  
|   | Range, 31  
|   | Signal, 30  
|   | VMEbus, 7; 39  
|   | Backplane, 16  
|   | Bus grant, 16  
|   | Daisy-Chains, 16  
|   | Slot selection, 16  
|   | Traffic, 9  
|   | Voltage drop, 18  
|   | VSB  
|   | Backplane, 16  
|   | VSB/ SCSI/P2, 7; 39  
|   | VT100.EXE, 21; 208  
|   | Script files, 46  
|   | VXIbus, 7; 39  

| W | Waveforms, 104  

| 240 | VBT-325 User's Manual | VMETRO |
Indexes

Wide mode, 28
Window
  Open, 97
Window command, 97; 142
Word recognition, 30
workstation, 5

X

XMEM325-PB, 11; 39
  8 Trigger events, 79
  Product overview, 39
Indexes