

Keysight M8195A Arbitrary Waveform Generator Revision 2

User's Guide



Notices

Copyright Notice

© Keysight Technologies 2016

No part of this manual may be reproduced in any form or by any means (including electronic storage and retrieval or translation into a foreign language) without prior agreement and written consent from Keysight Technologies, Inc. as governed by United States and international copyright laws.

Manual Part Number

M8195-91040

Edition

Edition 4.0, April 2016

Published by:

Keysight Technologies
Deutschland GmbH,
Herrenberger Str. 130,
71034 Böblingen, Germany

Technology Licenses

The hardware and/or software described in this document are furnished under a license and may be used or copied only in accordance with the terms of such license.

ESD Sensitive Device

All front-panel connectors of the M8195A are sensitive to Electrostatic discharge (ESD). We recommend to operate the instrument in an electrostatic safe environment.

There is a risk of instrument malfunction when touching a connector.

Please follow this instruction:

Before touching the front-panel connectors, discharge yourself by touching the properly grounded mainframe.

U.S. Government Rights

The Software is “commercial computer software,” as defined by Federal Acquisition Regulation (“FAR”) 2.101. Pursuant to FAR 12.212 and 27.405-3 and Department of Defense FAR Supplement (“DFARS”) 227.7202, the U.S. government acquires commercial computer software under the same terms by which the software is customarily provided to the public. Accordingly, Keysight provides the Software to U.S. government customers under its standard commercial license, which is embodied in its End User License Agreement (EULA), a copy of which can be found at <http://www.keysight.com/find/sweula>. The license set forth in the EULA represents the exclusive authority by which the U.S. government may use, modify, distribute, or disclose the Software. The EULA and the license set forth therein, does not require or permit, among other things, that Keysight: (1) Furnish technical information related to commercial computer software or commercial computer software documentation that is not customarily provided to the public; or (2) Relinquish to, or otherwise provide, the government rights in excess of these rights customarily provided to the public to use, modify, reproduce, release, perform, display, or disclose commercial computer software or commercial computer software documentation. No additional government requirements beyond those set forth in the EULA shall apply, except to the extent that those terms, rights, or licenses are explicitly required from all providers of commercial computer software pursuant to the FAR and the DFARS and are set forth specifically in writing elsewhere in the EULA. Keysight shall be under no obligation to update, revise or otherwise modify the Software. With respect to any technical data as defined by FAR 2.101, pursuant to FAR 12.211 and 27.404.2 and DFARS 227.7102, the U.S. government acquires no greater than Limited Rights as defined in FAR 27.401 or DFAR 227.7103-5 (c), as applicable in any technical data.

Warranty

THE MATERIAL CONTAINED IN THIS DOCUMENT IS PROVIDED “AS IS,” AND IS SUBJECT TO BEING CHANGED, WITHOUT NOTICE, IN FUTURE EDITIONS. FURTHER, TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, KEYSIGHT DISCLAIMS ALL WARRANTIES, EITHER EXPRESS OR IMPLIED, WITH REGARD TO THIS MANUAL AND ANY INFORMATION CONTAINED HEREIN, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. KEYSIGHT SHALL NOT BE LIABLE FOR ERRORS OR FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES IN CONNECTION WITH THE FURNISHING, USE, OR PERFORMANCE OF THIS DOCUMENT OR OF ANY INFORMATION CONTAINED HEREIN. SHOULD KEYSIGHT AND THE USER HAVE A SEPARATE WRITTEN AGREEMENT WITH WARRANTY TERMS COVERING THE MATERIAL IN THIS DOCUMENT THAT CONFLICT WITH THESE TERMS, THE WARRANTY TERMS IN THE SEPARATE AGREEMENT SHALL CONTROL.

Safety Information

CAUTION

A CAUTION notice denotes a hazard. It calls attention to an operating procedure, practice, or the like that, if not correctly performed or adhered to, could result in damage to the product or loss of important data. Do not proceed beyond a CAUTION notice until the indicated conditions are fully understood and met.

WARNING

A WARNING notice denotes a hazard. It calls attention to an operating procedure, practice, or the like that, if not correctly performed or adhered to, could result in personal injury or death. Do not proceed beyond a WARNING notice until the indicated conditions are fully understood and met.

Safety Summary

| | |
|---|---|
| General Safety Precautions | The following general safety precautions must be observed during all phases of operation of this instrument. Failure to comply with these precautions or with specific warnings elsewhere in this manual violates safety standards of design, manufacture, and intended use of the instrument. For safe operation the general safety precautions for the M9502A and M9505A AXIe chassis, must be followed. See: http://www.keysight.com/find/M9505A Keysight Technologies Inc. assumes no liability for the customer's failure to comply with these requirements. Before operation, review the instrument and manual for safety markings and instructions. You must follow these to ensure safe operation and to maintain the instrument in safe condition. |
| Initial Inspection | Inspect the shipping container for damage. If there is damage to the container or cushioning, keep them until you have checked the contents of the shipment for completeness and verified the instrument both mechanically and electrically. The Performance Tests give procedures for checking the operation of the instrument. If the contents are incomplete, mechanical damage or defect is apparent, or if an instrument does not pass the operator's checks, notify the nearest Keysight Technologies Sales/Service Office. WARNING To avoid hazardous electrical shock, do not perform electrical tests when there are signs of shipping damage to any portion of the outer enclosure (covers, panels, etc.). |
| General | This product is a Safety Class 3 instrument. The protective features of this product may be impaired if it is used in a manner not specified in the operation instructions. |
| Environment Conditions | This instrument is intended for indoor use in an installation category II, pollution degree 2 environment. It is designed to operate within a temperature range of 0 °C – 40 °C (32 °F – 105 °F) at a maximum relative humidity of 80% and at altitudes of up to 2000 meters. This module can be stored or shipped at temperatures between -40 °C and +70 °C. Protect the module from temperature extremes that may cause condensation within it. |
| Before Applying Power | Verify that all safety precautions are taken including those defined for the mainframe. |
| Line Power Requirements | The Keysight M8195A operates when installed in an Keysight AXIe mainframe. |
| Do Not Operate in an Explosive Atmosphere | Do not operate the instrument in the presence of flammable gases or fumes. |
| Do Not Remove the Instrument Cover | Operating personnel must not remove instrument covers. Component replacement and internal adjustments must be made only by qualified personnel. Instruments that appear damaged or defective should be made inoperative and secured against unintended operation until they can be repaired by qualified service personnel. |

Safety Symbols

Table 1: Safety symbol

| Symbol | Description |
|--------|---|
| | Indicates warning or caution. If you see this symbol on a product, you must refer to the manuals for specific Warning or Caution information to avoid personal injury or damage to the product. |
| | C-Tick Conformity Mark of the Australian ACA for EMC compliance. |
| | CE Marking to state compliance within the European Community: This product is in conformity with the relevant European Directives. |
| | General Recycling Mark |

Table 2: Compliance and environmental information

| Symbol | Description |
|--------|--|
| | This product complies with the WEEE Directive (2002/96/EC) marketing requirements. The affixed label indicates that you must not discard this electrical/electronic product in domestic household waste. Product category: With reference to the equipment types in the WEEE Directive Annexure I, this product is classed as a "Monitoring and Control instrumentation" product. |
| | Do not dispose in domestic household waste. |
| | To return unwanted products, contact your local Keysight office, or see http://about.keysight.com/en/companyinfo/environment/takeback.shtml for more information. |

Contents

Contents

1 Introduction

| | | |
|-------|-------------------------------------|----|
| 1.1 | Document History | 17 |
| 1.2 | Options | 18 |
| 1.3 | Installing Licenses | 19 |
| 1.4 | The Front Panel of the M8195A Rev 2 | 21 |
| 1.4.1 | Status LED | 22 |
| 1.4.2 | DATA Out LED | 22 |
| 1.4.3 | Trigger IN and Event In LED | 23 |
| 1.4.4 | Ref CLK IN LED | 24 |
| 1.5 | Theory of Operation | 25 |
| 1.5.1 | M8195A Block Diagram | 25 |
| 1.5.2 | Timing Block Diagram | 28 |
| 1.5.3 | Delay Adjust | 30 |
| 1.5.4 | Extended Memory Configuration | 31 |
| 1.5.5 | Instrument Modes | 33 |

2 M8195A User Interface

| | | |
|--------|---|----|
| 2.1 | Introduction | 37 |
| 2.2 | Launching the M8195A Soft Front Panel | 38 |
| 2.3 | M8195A User Interface Overview | 40 |
| 2.3.1 | Title Bar | 40 |
| 2.3.2 | Menu Bar | 40 |
| 2.3.3 | Status Bar | 42 |
| 2.3.4 | Clock/Output/Trigger/FIR Filter/Standard Waveform/Multi-Tone Waveform/Complex Modulated Waveform/Serial Data Waveform/Import Waveform/Sequence/Control Tabs | 42 |
| 2.3.5 | Numeric Control Usage | 43 |
| 2.4 | Driver Call Log | 44 |
| 2.5 | Errors List Window | 45 |
| 2.6 | Clock Tab | 47 |
| 2.7 | Output Tab | 49 |
| 2.8 | Trigger Tab | 52 |
| 2.9 | FIR Filter Tab | 54 |
| 2.10 | Standard Waveform Tab | 57 |
| 2.11 | Multi-Tone Waveform Tab | 63 |
| 2.12 | Complex Modulated Waveform Tab | 69 |
| 2.13 | Serial Data Waveform Tab | 79 |
| 2.13.1 | Bitmapping for Binary Data to PAM Signals | 91 |

| | | |
|--------|--|-----|
| 2.14 | Import Waveform Tab | 93 |
| 2.15 | Sequence/Control Tab | 100 |
| 2.16 | Correction File Format | 106 |
| 3 | Sequencing | |
| 3.1 | Introduction | 109 |
| 3.1.1 | Sequencing Internal Memory | 109 |
| 3.1.2 | Option Sequencing for Extended Memory | 110 |
| 3.1.3 | Sequence Table | 110 |
| 3.1.4 | Sequencer Granularity | 111 |
| 3.2 | Sequencing Hierarchy | 112 |
| 3.2.1 | Segment | 112 |
| 3.2.2 | Sequence | 112 |
| 3.2.3 | Scenario | 113 |
| 3.3 | Trigger Modes | 113 |
| 3.3.1 | Continuous | 113 |
| 3.3.2 | Triggered | 113 |
| 3.3.3 | Gated | 114 |
| 3.4 | Arm Mode | 114 |
| 3.4.1 | Self Armed | 114 |
| 3.4.2 | Armed | 114 |
| 3.5 | Advancement Modes | 114 |
| 3.5.1 | Auto | 115 |
| 3.5.2 | Conditional | 115 |
| 3.5.3 | Repeated | 115 |
| 3.5.4 | Single | 115 |
| 3.6 | Sequencer Controls | 115 |
| 3.6.1 | External Inputs | 116 |
| 3.6.2 | Logical Functions | 118 |
| 3.6.3 | Internal Trigger Generator | 119 |
| 3.6.4 | Mapping External Inputs to Logical Functions | 119 |
| 3.7 | Sequencer Execution Flow | 121 |
| 3.8 | Sequencer Modes | 122 |
| 3.8.1 | Arbitrary Mode | 122 |
| 3.8.2 | Sequence Mode | 127 |
| 3.8.3 | Scenario Mode | 134 |
| 3.9 | Dynamic Sequencing | 138 |
| 3.9.1 | Dynamic Continuous | 139 |
| 3.9.2 | Dynamic Triggered | 140 |
| 3.10 | Idle Command Segments | 141 |
| 3.11 | Limitations | 142 |
| 3.11.1 | Segment Length and Linear Playtime | 142 |

| | | |
|----------|--|-----|
| 4 | Streaming | |
| 4.1 | Introduction | 145 |
| 4.2 | Streaming Implementation Using Dynamic Modes | 145 |
| 5 | Markers | |
| 5.1 | Introduction | 147 |
| 5.2 | Dealing with Markers | 147 |
| 5.2.1 | Limitations | 148 |
| 5.2.2 | Sample Marker in Segments which are Addressed Offset Based | |
| | | 150 |
| 6 | General Programming | |
| 6.1 | Introduction | 152 |
| 6.2 | IVI-COM Programming | 154 |
| 6.3 | SCPI Programming | 154 |
| 6.3.1 | AgM8195SFP.exe | 155 |
| 6.4 | Programming Recommendations | 157 |
| 6.5 | System Related Commands (SYSTem Subsystem) | 158 |
| 6.5.1 | :SYSTem:EIN:MODE[?] EIN TOUT | 158 |
| 6.5.2 | :SYSTem:ERRor[:NEXT]? | 158 |
| 6.5.3 | :SYSTem:HELP:HEADers? | 159 |
| 6.5.4 | :SYSTem:LICense:EXTended:LIST? | 160 |
| 6.5.5 | :SYSTem:SET[?] | 160 |
| 6.5.6 | :SYSTem:VERSion? | 161 |
| 6.5.7 | :SYSTem:COMMunicate:*? | 161 |
| 6.6 | Common Command List | 165 |
| 6.6.1 | *IDN? | 165 |
| 6.6.2 | *CLS | 165 |
| 6.6.3 | *ESE | 165 |
| 6.6.4 | ESR? | 165 |
| 6.6.5 | *OPC | 165 |
| 6.6.6 | *OPC? | 166 |
| 6.6.7 | *OPT? | 166 |
| 6.6.8 | *RST | 166 |
| 6.6.9 | *SRE[?] | 166 |
| 6.6.10 | *STB? | 166 |
| 6.6.11 | *TST? | 167 |
| 6.6.12 | *LRN? | 167 |
| 6.6.13 | *WAI? | 167 |
| 6.7 | Status Model | 168 |
| 6.7.1 | :STATus:PRESet | 170 |
| 6.7.2 | Status Byte Register | 170 |
| 6.7.3 | Questionable Data Register Command Subsystem | 171 |
| 6.7.4 | Operation Status Subsystem | 173 |

| | | |
|--------|---|-----|
| 6.7.5 | Voltage Status Subsystem | 175 |
| 6.7.6 | Frequency Status Subsystem | 176 |
| 6.7.7 | Sequence Status Subsystem | 176 |
| 6.7.8 | Connection Status Subsystem | 177 |
| 6.7.9 | Run Status Subsystem | 178 |
| 6.8 | :ARM/TRIGger Subsystem | 179 |
| 6.8.1 | :ABORT[1 2 3 4] | 179 |
| 6.8.2 | :ARM[:SEQUence][:STARt][:LAYer]:MDELay[?] <module_delay> MINimum MAXimum | 179 |
| 6.8.3 | ARM[:SEQUence][:STARt][:LAYer]:SDELay[1 2 3 4][?] <delay> MINimum MAXimum | 180 |
| 6.8.4 | :INITiate:CONTinuous:ENABLE[?] SELF ARMed | 180 |
| 6.8.5 | :INITiate:CONTinuous[:STATe][?] OFF ON 0 1 | 181 |
| 6.8.6 | :INITiate:GATE[:STATe][?] OFF ON 0 1 | 182 |
| 6.8.7 | :INITiate:IMMEDIATE[1 2 3 4] | 183 |
| 6.8.8 | :ARM[:SEQUence][:STARt][:LAYer]:TRIGger:LEVel[?] <level> MINimum MAXimum | 183 |
| 6.8.9 | :ARM[:SEQUence][:STARt][:LAYer]:TRIGger:SLOPe[?] POSitive NEGative EITHER | 184 |
| 6.8.10 | :ARM[:SEQUence][:STARt][:LAYer]:TRIGger:SOURce[?] TRIGger EVENT INTERNAL | 185 |
| 6.8.11 | :ARM[:SEQUence][:STARt][:LAYer]:TRIGger:FREQuency[?] <frequency> MINimum MAXimum | 185 |
| 6.8.12 | :ARM[:SEQUence][:STARt][:LAYer]:TRIGger:OPERation[?] ASYNchronous SYNchronous | 186 |
| 6.8.13 | :ARM[:SEQUence][:STARt][:LAYer]:EVENT:LEVel[?] <level> MINimum MAXimum | 186 |
| 6.8.14 | :ARM[:SEQUence][:STARt][:LAYer]:EVENT:SLOPe[?] POSitive NEGative EITHER | 187 |
| 6.8.15 | :TRIGger[:SEQUence][:STARt]:SOURce:ENABLE[?] TRIGger EVENT 187 | |
| 6.8.16 | :TRIGger[:SEQUence][:STARt]:ENABLE:HWDISable[:STATe][?] 0 1 OFF ON | 188 |
| 6.8.17 | :TRIGger[:SEQUence][:STARt]:BEGin:HWDISable[:STATe][?] 0 1 OFF ON | 188 |
| 6.8.18 | :TRIGger[:SEQUence][:STARt]:ADVance:HWDISable[:STATe][?] 0 1 OFF ON | 189 |
| 6.9 | TRIGger - Trigger Input | 190 |
| 6.9.1 | :TRIGger[:SEQUence][:STARt]:SOURce:ADVance[?] TRIGger EVENT INTERNAL | 190 |
| 6.9.2 | :TRIGger[:SEQUence][:STARt]:ENABLE[:IMMEDIATE] | 190 |
| 6.9.3 | :TRIGger[:SEQUence][:STARt]:BEGin[:IMMEDIATE] | 191 |
| 6.9.4 | :TRIGger[:SEQUence][:STARt]:BEGin:GATE[:STATe][?] OFF ON 0 1 | 191 |
| 6.9.5 | :TRIGger[:SEQUence][:STARt]:ADVance[:IMMEDIATE] | 192 |
| 6.10 | :FORMAT Subsystem | 193 |
| 6.10.1 | :FORMAT:BORDER NORMAL SWAPPED | 193 |
| 6.11 | :INSTRument Subsystem | 194 |
| 6.11.1 | :INSTRument: SLOT[:NUMBER]? 194 | |

| | | |
|---------|--|-----|
| 6.11.2 | :INSTrument:IDENTify [<seconds>] | 194 |
| 6.11.3 | :INSTrument:IDENTify:STOP | 195 |
| 6.11.4 | :INSTrument: HWRevision? | 195 |
| 6.11.5 | :INSTrument:DACMode[?] SINGLe DUAL FOUR MARKeR DCDuPliCate DCMaRKer | 196 |
| 6.11.6 | :INSTrument:MEMory:EXTended:RDIVider [?] DIV1 DIV2 DIV4 197 | |
| 6.11.7 | :INSTrument:MMODule:CONFig? | 197 |
| 6.11.8 | :INSTrument:MMODule:MODE? | 198 |
| 6.12 | :MMEMemory Subsystem | 199 |
| 6.12.1 | :MMEMemory:CATalog? [<directory_name>] | 199 |
| 6.12.2 | :MMEMemory:CDIRectory [<directory_name>] | 200 |
| 6.12.3 | :MMEMemory:COOPY <string>,<string>[,<string>,<string>] | 201 |
| 6.12.4 | :MMEMemory:DElete <file_name>[,<directory_name>] | 202 |
| 6.12.5 | :MMEMemory:DATA <file_name>,<data> | 202 |
| 6.12.6 | :MMEMemory:DATA? <file_name> | 203 |
| 6.12.7 | :MMEMemory:MDIRectory <directory_name> | 203 |
| 6.12.8 | :MMEMemory:MOVE <string>,<string>[,<string>,<string>] | 204 |
| 6.12.9 | :MMEMemory:RDIRectory <directory_name> | 204 |
| 6.12.10 | :MMEMemory:LOAD:CSTate <file_name> | 205 |
| 6.12.11 | :MMEMemory:STORE:CSTate <file_name> | 205 |
| 6.13 | :OUTPut Subsystem | 206 |
| 6.13.1 | :OUTPut[1 2 3 4]:STATe[?] OFF ON 0 1 | 206 |
| 6.13.2 | :OUTPut: ROSeLLator:SOURce[?] INternal EXternal SCLK1 SCLK2 | 206 |
| 6.13.3 | :OUTPut: ROSeLLator:SCD[?] <sample_clock_divider> MINimum MAXimum | 207 |
| 6.13.4 | :OUTPut: ROSeLLator:RCD1[?] < reference_clock_divider1> MINimum MAXimum | 207 |
| 6.13.5 | :OUTPut: ROSeLLator:RCD2[?] <reference_clock_divider2> MINimum MAXimum | 208 |
| 6.13.6 | :OUTPut[1 2 3 4]:DIOFFset[?] <value> MINimum MAXimum | 208 |
| 6.13.7 | :OUTPut[1 2 3 4]:FILTter:FRATe[:VALue][?] | 209 |
| 6.13.8 | :OUTPut[1 2 3 4]:FILTter:FRATe:TYPE[?] LOWPass ZOH USER 209 | |
| 6.13.9 | :OUTPut[1 2 3 4]:FILTter:FRATe:SCALE[?] <scale> MINimum MAXimum | 210 |
| 6.13.10 | :OUTPut[1 2 3 4]:FILTter:FRATe:DELay[?] <delay> MINimum MAXimum | 210 |
| 6.13.11 | :OUTPut[1 2 3 4]:FILTter:HRATe[:VALue] [?] | 211 |
| 6.13.12 | :OUTPut[1 2 3 4]:FILTter:HRATe:TYPE[?] NYQuist LINEar ZOH USER | 211 |
| 6.13.13 | :OUTPut[1 2 3 4]:FILTter:HRATe:SCALE[?] <scale> MINimum MAXimum | 212 |
| 6.13.14 | :OUTPut[1 2 3 4]:FILTter:HRATe:DELay[?] <delay> MINimum MAXimum | 212 |
| 6.13.15 | :OUTPut[1 2 3 4]:FILTter:QRATe[:VALue] [?] | 213 |
| 6.13.16 | :OUTPut[1 2 3 4]:FILTter:QRATe:TYPE[?] NYQuist LINEar ZOH USER | 213 |

| | | |
|---------|--|-----|
| 6.13.17 | :OUTPut[1 2 3 4]:FILTter:QRATE:SCALE[?] <scale> MINimum MAXimum | 214 |
| 6.13.18 | :OUTPut[1 2 3 4]:FILTter:QRATE:DELay[?] <delay> MINimum MAXimum | 214 |
| 6.14 | Sampling Frequency Commands | 215 |
| 6.14.1 | [:SOURce]:FREQuency:RASTer[?] <frequency> MINimum MAXimum | 215 |
| 6.15 | Reference Oscillator Commands | 216 |
| 6.15.1 | [:SOURce]:ROSCillator:SOURce[?] EXTERNAL AXI INTernal | 216 |
| 6.15.2 | [:SOURce]:ROSCillator:SOURce:CHECK? EXTERNAL AXI INTernal | 217 |
| 6.15.3 | [:SOURce]:ROSCillator:FREQuency[?] <frequency> MINimum MAXimum | 217 |
| 6.15.4 | [:SOURce]:ROSCillator:RANGe[?] RANG1 RANG2 | 218 |
| 6.15.5 | [:SOURce]:ROSCillator:RNG1 RNG2:FREQuency[?] <frequency> MINimum MAXimum | 219 |
| 6.16 | :VOLTage Subsystem | 220 |
| 6.16.1 | [:SOURce]:VOLTage[1 2 3 4][:LEVel][[:IMMEDIATE][?]] <level> MINimum MAXimum | 220 |
| 6.16.2 | [:SOURce]:VOLTage[1 2 3 4][:LEVel][[:IMMEDIATE]]:OFFSet[?] <level> MINimum MAXimum | 221 |
| 6.16.3 | [:SOURce]:VOLTage[1 2 3 4][:LEVel][[:IMMEDIATE]]:HIGH[?] <level> MINimum MAXimum | 221 |
| 6.16.4 | [:SOURce]:VOLTage[1 2 3 4][:LEVel][[:IMMEDIATE]]:LOW[?] <level> MINimum MAXimum | 222 |
| 6.16.5 | [:SOURce]:VOLTage[1 2 3 4][:LEVel][[:IMMEDIATE]]:TERMination[?] <level> MINimum MAXimum | 222 |
| 6.17 | [:SOURce]:FUNCtion:MODE ARBITRARY STSequence STSCenario | 223 |
| 6.18 | :STABle Subsystem | 224 |
| 6.18.1 | [:SOURce]:STABle:RESET | 224 |
| 6.18.2 | [:SOURce]:STABle:DATA[?] <sequence_table_index>,<length> <block> <value>,<value>...) | 224 |
| 6.18.3 | [:SOURce]:STABle:DATA:BLOCK? <sequence_table_index>,<length> | 230 |
| 6.18.4 | [:SOURce]:STABle:SEQuence:SELect[?] <sequence_table_index> MINimum MAXimum | 230 |
| 6.18.5 | [:SOURce]:STABle:SEQuence:STATE? | 231 |
| 6.18.6 | [:SOURce]:STABle:DYNAMIC:[STATE][?] | 232 |
| 6.18.7 | [:SOURce]:STABle:DYNAMIC:SELect <sequence_table_index> | 232 |
| 6.18.8 | [:SOURce]:STABle:SCENario:SELect[?] <sequence_table_index> MINimum MAXimum | 233 |
| 6.18.9 | [:SOURce]:STABle:SCENario:ADVance[?] AUTO CONDITIONAL REPEAT SINGLE | 233 |
| 6.18.10 | [:SOURce]:STABle:SCENario:COUNT[?] <count> MINimum MAXimum | 234 |
| 6.19 | Frequency and Phase Response Data Access | 235 |
| 6.19.1 | [:SOURce]:CHARacteris[1 2 3 4][:VALUE]? [<amplitude>,<sample_frequency>]] | 235 |

| | | |
|---------|---|-----|
| 6.20 | :TRACe Subsystem | 236 |
| 6.20.1 | Waveform Data Format | 236 |
| 6.20.2 | Arbitrary Waveform Generation | 237 |
| 6.20.3 | TRACe[1 2 3 4]:MMODE[?] | 237 |
| 6.20.4 | :TRAC[1 2 3 4]:DEF | 238 |
| 6.20.5 | :TRAC[1 2 3 4]:DEF:NEW? | 238 |
| 6.20.6 | :TRAC[1 2 3 4]:DEF:WONL | 239 |
| 6.20.7 | :TRAC[1 2 3 4]:DEF:WONL:NEW? | 239 |
| 6.20.8 | :TRAC[1 2 3 4]:DATA[?] | 240 |
| 6.20.9 | :TRAC[1 2 3 4]:IMP | 242 |
| 6.20.10 | :TRAC[1 2 3 4]:DEL | 249 |
| 6.20.11 | :TRAC[1 2 3 4]:DEL:ALL | 249 |
| 6.20.12 | :TRAC[1 2 3 4]:CAT? | 250 |
| 6.20.13 | :TRAC[1 2 3 4]:FREE? | 250 |
| 6.20.14 | :TRAC[1 2 3 4]:NAME[?] | 251 |
| 6.20.15 | :TRAC[1 2 3 4]:COMM[?] | 251 |
| 6.20.16 | :TRAC[1 2 3 4]:SEL[?]<segment_id> MINimum MAXimum | 252 |
| 6.20.17 | :TRAC[1 2 3 4]:ADV[?] | 252 |
| 6.20.18 | :TRAC[1 2 3 4]:COUN[?]<count> MINimum MAXimum | 253 |
| 6.20.19 | :TRAC[1 2 3 4]:MARK[?] | 253 |
| 6.21 | :TEST Subsystem | 254 |
| 6.21.1 | :TEST:PON? | 254 |
| 6.21.2 | :TEST:TST? | 254 |

7 Examples

| | | |
|-----|---------------------------------|-----|
| 7.1 | Introduction | 255 |
| 7.2 | Remote Programming Examples | 255 |
| 7.3 | Example Files for Import | 255 |
| 7.4 | Example Correction Files | 256 |
| 7.5 | Example Custom Modulation Files | 256 |

8 Appendix

| | | |
|-------|---|-----|
| 8.1 | Resampling Algorithms for Waveform Import | 257 |
| 8.1.1 | Resampling Requirements | 257 |
| 8.1.2 | Resampling Methodology | 258 |

1 Introduction

| | | | |
|-----|-------------------------------------|---|----|
| 1.1 | Document History | / | 17 |
| 1.2 | Options | / | 18 |
| 1.3 | Installing Licenses | / | 19 |
| 1.4 | The Front Panel of the M8195A Rev 2 | / | 21 |
| 1.5 | Theory of Operation | / | 25 |

Introduction

The Keysight M8195A is a 65 GSa/s Arbitrary Waveform Generator with highest bandwidth and channel density. It offers up to 16 GSa waveform memory. The M8195A is ideally suited to address following key applications:

- Coherent optical – a single M8195A module can generate 2 independent I/Q baseband signals (dual polarization = 4 channels) at up to 32 Gbaud and beyond.
- Multi-level / Multi-channel digital signals – generate NRZ, PAM4, PAM8, DMT, etc. signals at up to 32 Gbaud. Embed/De-embed channels, add Jitter, ISI, noise and other distortions.
- Physics, chemistry, and electronics research – generate any mathematically defined arbitrary waveforms, ultra-short yet precise pulses and extremely wideband chirps.
- Wideband RF/µW – generate extremely wideband RF signals with an instantaneous bandwidth of DC to 20 GHz for aerospace/defense and communication applications.

| | |
|--|---|
| Features and Benefits | <p>The M8195A is an arbitrary waveform generator with highest sample rate, bandwidth, and channel density:</p> <p>Sample rate up to 65 GSa/s (on each channel)</p> <p>Analog bandwidth: 25 GHz</p> <p>Vertical resolution: 8 bits</p> <p>1, 2, or 4 differential channels per 1-slot high AXIe module (number of channels is software upgradable)</p> <p>Built-in frequency and phase response calibration</p> <p>Amplitude up to 1 Vpp (single ended); 2 Vpp (differential)</p> <p>Transition Times: $t_{Rise,20\%...80\%}$; $t_{Fall,20\%...80\%}$: 18 ps (typ)</p> <p>Ultra low intrinsic Random Jitter: $RJ_{rms} < 200$ fs (typ)</p> <p>Form factor: 1-slot AXIe module controlled via external PC or embedded AXIe system controller M9536A</p> |
| Supporting Operating System | <p>The Keysight M8195A supports the following operating systems:</p> <p>Windows 10 (32 bit or 64 bit)</p> <p>Windows 8.1 (32 bit or 64 bit)</p> <p>Windows 8 (32 bit or 64 bit)</p> <p>Windows 7 (32 bit or 64 bit)</p> |
| Control M8195A from M8070A System Software for BER Test Solutions | <p>For digital applications that require multi-level signaling like PAM-4, the M8195A arbitrary waveform generator can be integrated with the M8070A System Software for M8000 Series of BER Test Solutions.</p> <p>Once integrated, M8195A will be visible in the module view of M8070A GUI, just like any other M8000 Series module.</p> <p>For M8195A integration with M8070A, following must be installed:</p> <ul style="list-style-type: none">• Keysight IO Libraries Suite 16.3 or higher• M8195A software version 1.3 or later• M8070A software version 3.0 or later <p>Once the M8195A module is mounted on to the chassis and relevant connections are made, it will be visible in the module view of the M8070A GUI.</p> <p>Please see the <i>M8070A User Guide</i> and <i>Online Help</i> for further information. Click the following link for latest version of the mentioned documents: http://www.keysight.com/find/M8020A</p> <p>Please note that Keysight M8070A can only be installed on 64 bit operating system for Windows 7, 8, and 8.1.</p> |

NOTE

M8195A Soft Front Panel must not be launched while using the instrument with M8070A.

NOTE

The M8070A is a licensed software, and thus requires a license to communicate with the M8020A/M8030A hardware. You can either purchase an M8070A license to install on a dedicated host computer (M8070A-OTP) or one that can be installed on a network server that will be used as a license server for operating over a company network (M8070A-ONP, floating/networked).

Please see the *M8020A/M8030A Getting Started Guide* for further information about licensing.

Click the following link for latest version of the mentioned document:

<http://www.keysight.com/find/M8020A>

<http://www.keysight.com/find/M8030A>

Additional Documents

Additional documentation can be found at:

<http://www.keysight.com/find/M9514A> for 13-slot chassis related documentation.

<http://www.keysight.com/find/M9505A> for 5-slot chassis related documentation.

<http://www.keysight.com/find/M9502A> for 2-slot chassis related documentation.

<http://www.keysight.com/find/M9048A> for PCIe desktop adapter card related documentation.

<http://www.keysight.com/find/M9536A> for embedded AXIe controller related documentation.

<http://www.keysight.com/find/M8195A> for AXIe based AWG module related documentation.

1.1 Document History

| | |
|------------------------------------|--|
| First Edition (October, 2015) | The first edition of the user's guide describes the functionality of the M8195A Revision 2 Version 2.0. In addition, it includes the description of dynamic sequencing which is not a part of the software version 2.0, but will be added with version 2.5. Changes are possible. |
| Second Edition (November, 2015) | The second edition of the user's guide describes the functionality of the M8195A Revision 2 Version 2.5. |
| Third Edition (February, 2016) | The third edition of the user's guide describes the functionality of the M8195A Revision 2 Version 2.6. |
| Fourth Edition (April, 2016) | The fourth edition of the user's guide describes the functionality of the M8195A Revision 2 Version 3.0. |

1.2 Options

For the M8195A Rev 2, following product options are available.

Table 3: Options provided by M8195A

| Product Number | Description | Available as SW upgrade? | Comment |
|----------------|--|-----------------------------|--|
| M8195A-001 | 1 channel, 65 GSa/s, 2 GSa per module | N/A (minimum configuration) | Must order either -001 or -002 or -004 |
| M8195A-002 | 2 channel, 65 GSa/s, 2 GSa per module | Yes | Must order either -001 or -002 or -004 |
| M8195A-004 | 4 channel, 65 GSa/s, 2 GSa per module | Yes | Must order either -001 or -002 or -004 |
| M8195A-U02 | Upgrade from one channel to two channels | Yes | Software upgradeable |
| M8195A-U04 | Upgrade from two channels to four channels | Yes | Software upgradeable |
| M8195A-16G | Upgrade to 16 GSa per module | Yes | Software upgradeable |
| M8195A-SEQ | Sequencer functionality | Yes | Software upgradeable |
| M8195A-FSW | Fast switching per module | Yes | Software upgradeable |
| M8195A-1A7 | ISO17025 | No | Calibration option |
| M8195A-Z54 | Z540 | No | Calibration option |

As a standard configuration, the M8195A contains 2 GSa of memory.

- | | |
|-----------------------------------|--|
| Option -001, -002, or -004 | With this option the number of channels is selected. The M8195A is available in a one channel (-001), two channel (-002) or 4 channel (-004) version. A software upgrade from one to two channels is possible by installing option U02. A software upgrade from two to four channels is possible by installing option U04. In order to upgrade from one to four channels, first option -U02 and next -U04 must be installed. |
| Option -16G | This option offers 16384 MSa (=16 GSa) waveform memory for the M8195A. Option -16G is software upgradeable. |
| Option -SEQ | This option offers extensive sequencing capabilities. For more details, refer to the chapter Sequencing. Option -SEQ is software upgradeable. |
| Option -FSW | This option enables the M8195A to externally select or step through segments or sequences faster than every 500 µs. Option -FSW is export controlled and is software upgradeable. |
| Option -1A7, -Z54 | Calibration options. |

1.3 Installing Licenses

After you purchase a license and you acquire the corresponding license file, you need to install the license on M8195A.

You can install the new license in the following ways:

1. In Keysight License Manager, click the **File** menu, and then select **Install....**
An **Install License File(s)** window appears. In this window, browse to the location where you saved the license file. Select the license file, and then click the **Open** button.
2. To manually install a license by entering the appropriate license file information, click the **Tools** menu, click **Enter License Text....** The **License Text Entry and Installation** dialog box appears.
Type in the license data exactly as you received from Keysight. Click the **Install** button to install the license.
3. On Windows-based systems, you can install the license by copying the license file into the license directory
`C:\Program Files\Keysight\licensing`.

Once the licenses are installed, you can use the Keysight License Manager to view all licenses for the local system as depicted in the following figure.

Options –U02, –U04, –SEQ, –16G, and –FSW are upgradable using the Keysight License Manager (KLM); see [Table 3](#).

Observe following steps while installing licenses:

1. Close the firmware of the M8195A
2. Install the licenses using KLM
3. Start the firmware of the M8195A. The firmware finds the new licenses in KLM and installs them in the M8195A.

In case of an upgrade from one channel (-001) to four channels (-004) following steps must be observed:

1. Close the firmware of the M8195A
2. Install license -U02 using KLM
3. Start the firmware of the M8195A. The firmware finds the new license –U02 in KLM and installs it in the M8195A.
4. Close the firmware of the M8195A
5. Install license -U04 using KLM
6. Start the firmware of the M8195A. The firmware finds the new license –U04 in KLM and installs it in the M8195A.

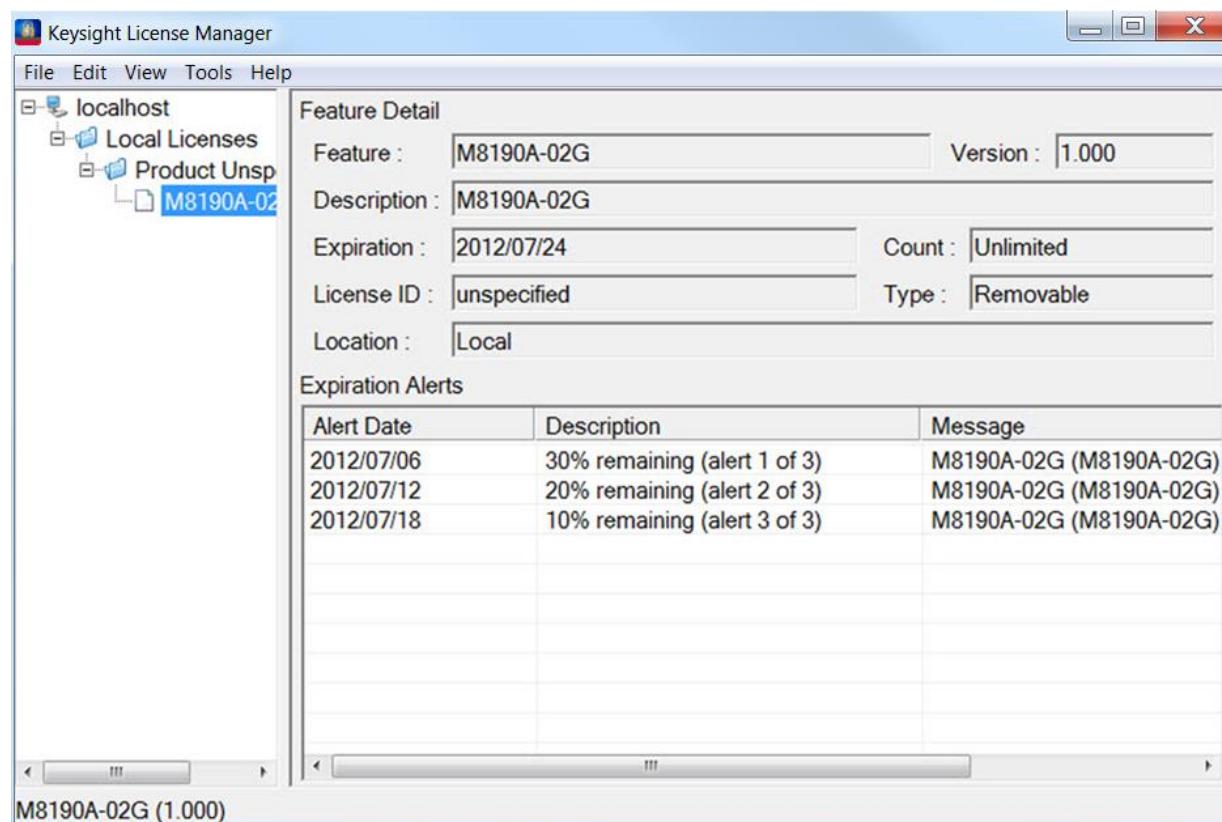


Figure 1: Using Keysight License Manager to view installed licenses

NOTE

Licenses for instrument options are transferred to the M8195A module. They are later no longer visible in the Keysight License Manager.

1.4 The Front Panel of the M8195A Rev 2

The Front Panel of the M8195A Rev 2 is shown in the figure below.



Figure 2: Front panel of M8195A

Data Outputs

The M8195A is always delivered with four physically available differential Data Outputs of the Digital to Analog Converter (DAC). The analog DAC outputs are labelled with DATA OUT CHANNEL 1, DATA OUT CHANNEL 2; DATA OUT CHANNEL 3, DATA OUT CHANNEL 4. Depending on the channel option (-001 or -002 or -004) that has been installed, the M8195A one, two, or four differential analog outputs of the Digital to Analog Converters (DAC) are enabled for data generation.

Option -001: The differential output DATA OUT CHANNEL 1 is enabled for analog data generation. Also, one or two digital markers can be generated at DATA OUT CHANNEL 3 and DATA OUT CHANNEL 4.

Option -002: The selected Instrument Mode (see section [Instrument Modes](#)) determines, which channels are enabled for analog data and marker generation.

- In 'Dual Channel' mode the differential outputs DATA OUT CHANNEL 1 and DATA CHANNEL 4 are enabled for analog data generation. DATA OUT CHANNEL 2 and DATA OUT CHANNEL 3 are disabled.
- In 'Dual Channel with Marker' mode the differential outputs DATA OUT CHANNEL 1 and DATA OUT CHANNEL 2 are enabled for analog data generation. One or two digital markers can be generated at DATA OUT CHANNEL 3 and DATA OUT CHANNEL 4.
- In 'Dual Channel Duplicate' mode the differential outputs DATA OUT CHANNEL 1, DATA OUT CHANNEL 2, DATA OUT CHANNEL 3 and DATA OUT CHANNEL 4 are enabled for analog data generation.

Option -004: The differential outputs DATA OUT CHANNEL 1, DATA OUT CHANNEL 2, DATA OUT CHANNEL 3 and DATA OUT CHANNEL 4 are enabled for analog data generation.

Note: The Data Outputs can be used differentially or single-ended. In case the output is used single-ended, the unused output must be terminated with 50 Ohm to GND to achieve optimum signal quality.

TRIG IN

The Trigger Input has a combined functionality as Trigger or Gate and is used to start the M8195A by an external signal. This input is defined in detail in the chapter Sequencing

EVENT IN

The Event Input (EVENT IN) is used to e.g. step through segments or scenarios by an external signal. This input is defined in detail in the chapter Sequencing.

REF CLK IN

The Reference Clock Input can be used to synchronize to an external clock. The input frequency can vary between 10MHz and 17 GHz.

REF CLK OUT

The Reference Clock Output can be used to synchronize a DUT to the M8195A. The adjustable output frequency covers a large frequency range.

1.4.1 Status LED

Following LEDs are available at the front panel to indicate the status of the AWG module:

The green '**Access**' LED:

It indicates that the controlling PC exchanges data with the AWG module.

The red '**Fail**' LED has following functionality:

It is 'ON' for about 30 seconds after powering the AXIe chassis.

After about 30 seconds the LED is switched 'OFF'. If an external PC is used to control the AXIe chassis, this PC can be powered after this LED has switched OFF.

During normal operation of the module this LED is 'OFF'. In case of an error condition such as e.g. a self-test error, the LED is switched 'ON'.

1.4.2 DATA Out LED

Table 4: Data out LED

| Color | Meaning | Description |
|-----------|---------------------------|---|
| Off | Output disabled | <p>Represents the state 'Disable'. Selectable from SFP or SCPI. The output amplifier is not powered</p> <p>After Power-On the LED is off.</p> <p>After successful initialization of the M8195A, the LED turns to its default state which is OFF.</p> |
| ON, green | Output enabled | <p>Represents the state 'Enable'. Selectable from SFP or SCPI</p> <ul style="list-style-type: none"> • Output amplitude is equal to the adjusted amplitude • Offset is equal to the adjusted amplitude • External Termination voltage is equal to the adjusted termination voltage |
| On, red | Protection circuit active | <p>Error condition such as</p> <ul style="list-style-type: none"> • The externally applied termination voltage significantly differs from the adjusted termination voltage • External termination resistor significantly differs from 50 Ohm <p>The protection circuit overwrites amplifier settings (amplitude, offset) such that the amplifier's output stage will not be destroyed => Amplifier is not powered</p> <p>User interaction is required to remove the externally applied error condition. After removal, the user must actively enable the output again.</p> |

NOTE

The DATA Output LED does not represent the RUN / Stop Status. Also, the Output LED does not indicate whether a valid pattern is loaded in a certain channel.

1.4.3 Trigger IN and Event IN LED

This LED indicates that an externally applied signal matches the adjusted threshold to be used as a Trigger or Event. The LED turns on for ~100 ms for each detected edge of the correct polarity. I.e. a rising edge turns the LED on for 100 ms if the polarity is adjusted to rising. If the polarity is adjusted to rising and a falling edge is externally applied, the LED remains OFF.

Notes:

- In case the edges are applied faster than every 100 ms, the LED is continuously ON.
- In trigger mode ‘Gated’, the LED is turned on for 100 ms when the gate signal becomes active. I.e. when the polarity is set to positive, the LED turns on for 100ms after the rising edge. When the polarity is set to negative, the LED turns on for 100 ms after the falling edge.
- In trigger mode ‘Gated’, the polarity cannot be set to ‘Either’

Table 5: Trigger IN and Event IN LED

| Color | Meaning | Description |
|-----------|---|--|
| Off | No external Trigger (Event) | In case the trigger source is not set to external, this LED is OFF. |
| ON, green | Valid external Trigger (Event) detected | In case the trigger mode is set to ‘asynchronous’, a Trigger (Event) is always valid. Set-up or hold time violations do not exist. Note: A ‘Force Trigger’ from the SFP or SCPI does not turn the LED ON |
| ON, red | Invalid external Trigger (Event) detected | In case the trigger mode is set to ‘synchronous’, a Trigger (Event) can be invalid because of a set-up or hold time violation. The LED turns On red in case a set-up or a hold time violation has been detected. Note A ‘Force Trigger’ from the SFP or SCPI does not turn the LED ON |

1.4.4 Ref CLK IN LED

Table 6: Ref CLK IN LED

| Color | Meaning | Description |
|-----------|-------------------------------------|--|
| Off | Applied Clock cannot be used | In case the clock reference is not set to Ref CLK IN, this LED is OFF. |
| ON, green | Valid signal at Ref CLK IN detected | <ul style="list-style-type: none">• CDR has locked on Ref CLK In and• The externally applied frequency is correct and• Ref CLK In has been selected as the clock reference |
| ON, red | No valid signal at Ref CLK IN | <ul style="list-style-type: none">• Ref CLK In has been selected as the clock reference• The externally applied clock signal is not valid. E.g. the frequency does not match the adjusted value or the amplitude is outside the specified range |

1.5 Theory of Operation

1.5.1 M8195A Block Diagram

The drawing below shows a block diagram of the instrument.

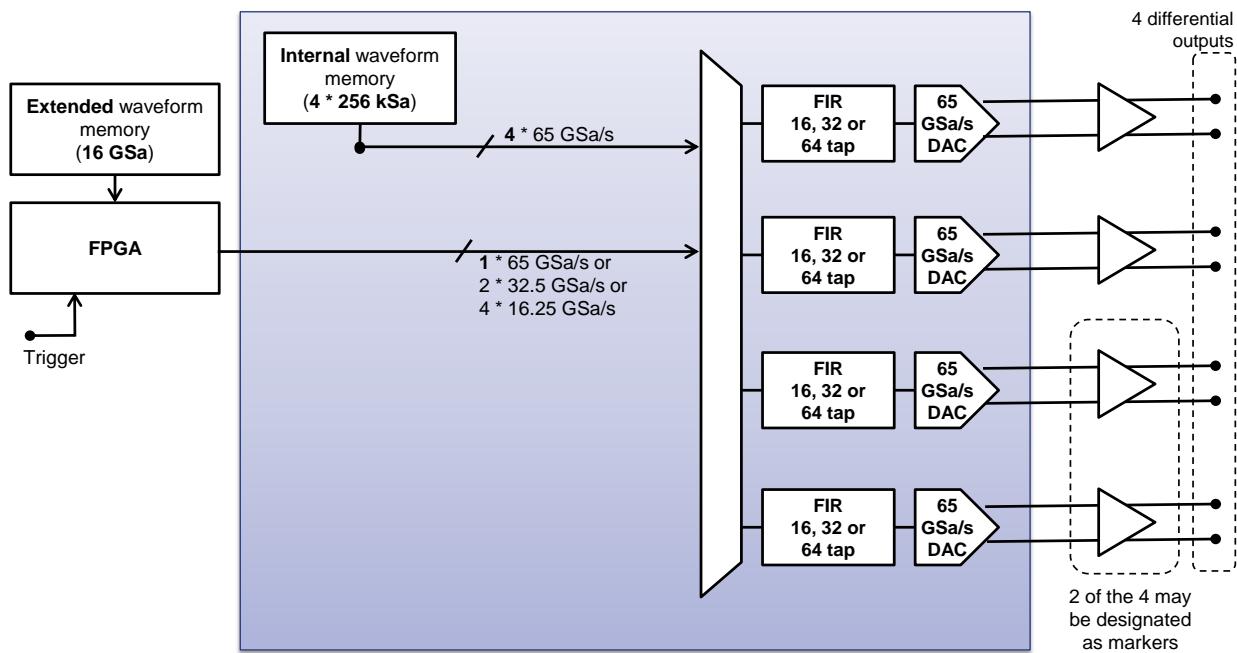


Figure 3: M8195A block diagram

The M8195A can operate in different modes: 'Single Channel', 'Single Channel with markers', 'Dual Channel', 'Dual Channel Duplicate', 'Dual Channel with markers', or 'Four Channel'

There are two different memory modes available: 'Internal' and 'Extended'. The memory mode is configurable for each channel.

The Sample Rate of all four Digital to Analog Converters (DAC) is selectable between 53.76 GSa/s ... 65 GSa/s. The internal waveform memory always operates at the sample rate. The extended waveform memory can operate at sample rate 53.76 GSa/s ... 65 GSa/s or at one half of the sample rate 26.88 GSa/s ... 32.5 GSa/s or at one fourth of the sample rate 13.44 GSa/s ... 16.25 GSa/s. The speed of operation of the extended memory is adjustable using the parameter 'Sample Rate Divider (Extended Memory)' which can be changed by the user. Possible values are 1, 2, and 4. The Sample Rate Divider is identical for all channels that are sourced from extended memory. In case the Sample Rate Divider is adjusted to two or four, the FIR filters are used as interpolation filters by factors of two or four. The interpolation is necessary as the DAC always operates in the range 53.76 GSa/s ... 65 GSa/s.

Each channel has a programmable FIR Filter.

The number of filter coefficients depends on the Sample Rate Divider; 16, 32, or 64 filter coefficients are available if the Sample Rate Divider is set to 1, 2 or, 4 respectively.

In case the Sample Rate Divider is changed, the FIR filter coefficients of each channel sourced from extended memory are loaded to operate as a by one or by two or by four interpolation filter.

[Figure 4](#) depicts how the FIR filters are used as interpolation filters. If the sample rate divider is set to two, the sample value '0' is inserted between each sample that is read from extended memory. If the sample rate divider is set to four, three consecutive times the sample value of '0' is inserted between each sample that is read from extended memory.

| Sample rate divider | Number of FIR filter coefficient | Default interpolation filter characteristic |
|---------------------|----------------------------------|--|
| 1 | 16 | No interpolation. No filter. Center tap is 1. All other coefficients are 0 (filter type zero order hold). |
| 2 | 32 | By two interpolation using a Nyquist (half-band) filter with rolloff factor 0.2. |
| 4 | 64 | By four interpolation using a Nyquist (quarter-band) filter with rolloff factor 0.2. |

There are two sets of filter coefficients for each channel. One set is currently used for data generation. The other set can be reconfigured in parallel with new coefficients. After reconfiguration, the entire reconfigured set can be used simultaneously for data generation. This allows reconfiguration during data transmission without generating distortions at the output signal. By pressing the corresponding 'Send To Instrument' button of the SFP or by sending the corresponding API command, the new set of filter coefficients is applied.

There are predefined sets of FIR filter coefficients which can be selected by the user. When selecting the 'user-defined' filter type, all FIR filter coefficients are fully controllable by the user.

There is a scaling multiplier at the output of the FIR filter which can be used to digitally scale the output signal by factor between 0.0 and 1.0.

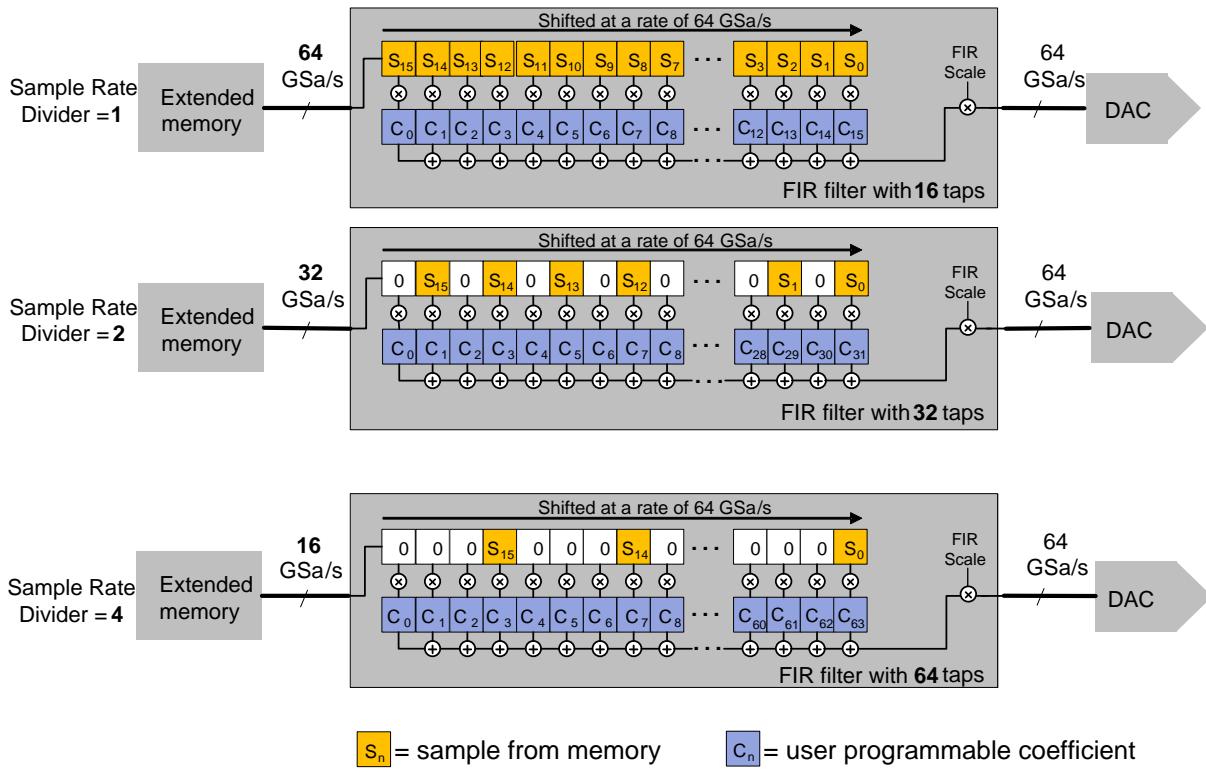


Figure 4: M8195A FIR filter operation with different dividers. Note: Sample rates are shown as 64, 32 and 16 GSa/s in the diagram.
 In fact, they are adjustable in the range: 53.76 ... 65; 26.88 ... 32.5 and 13.44 ... 16.25 GSa/s

1.5.2 Timing Block Diagram

The drawing below shows a block diagram of the instrument.

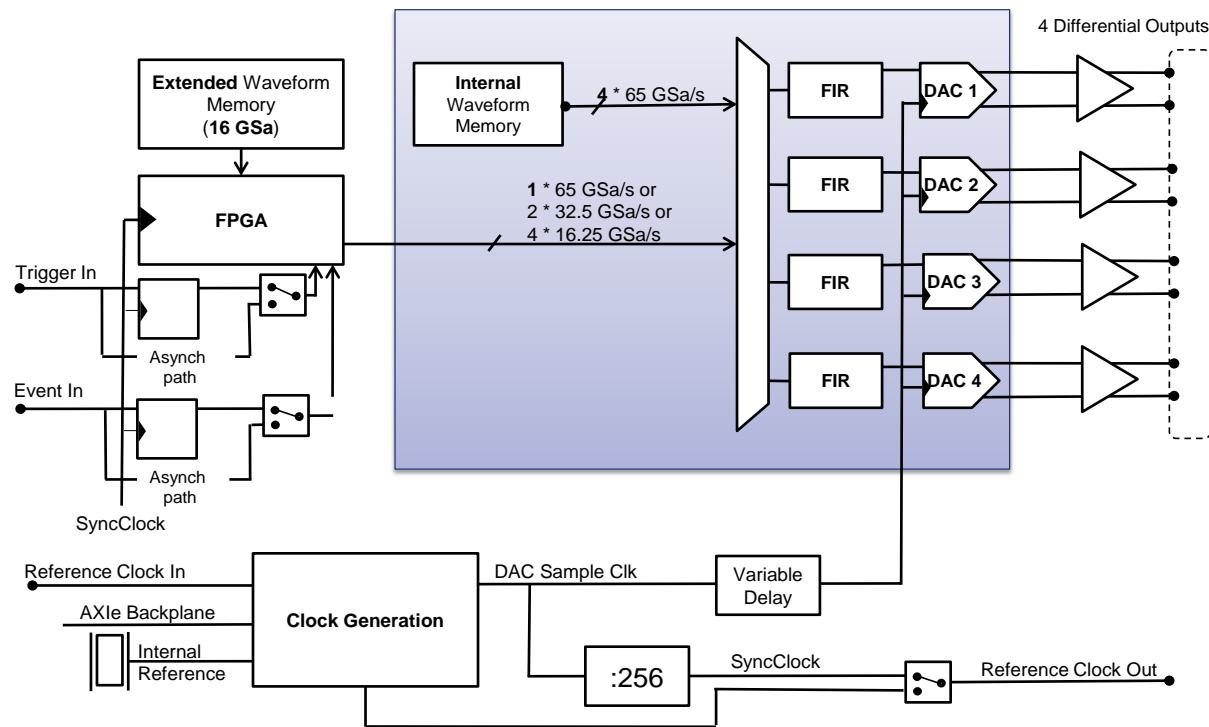


Figure 5: M8195A Timing Block Diagram

The level of detail is chosen to provide a general high level understanding of how the instrument is working. Therefore, not all of ports are shown in the above diagram.

Definitions**DAC Sample Rate:**

The DAC Sample rate is always in the range of 53.76 GSa/s ... 65 GSa/s. The DAC sample rate indicates how many samples per seconds the DAC can generate. The unit of the sample rate is Sa/s

DAC Sample Frequency:

The DAC Sample frequency is always in the range of 53.76 GHz ... 65 GHz. As the DAC sample frequency references to a clock, the unit of the sample frequency is Hz.

DAC Sample CLK:

The DAC Sample CLK is the clock signal that sources the four DAC of the M8195A. There is a variable delay element between the clock generation block and the DAC.

SyncClock:

$$\text{SyncClock} = \text{DAC Sample Rate} / 256$$

The SyncClock is the timing reference for the M8195A. Latency specifications such as the trigger to output latency are referenced to it. Also, the set-up and hold timing specification for synchronous trigger is referenced to the Sync Clock. The sequencer is also working with this clock. The Sync Clock is an internal clock signal that can be output at the Reference Clock Out in order to accurately align the timing with an external DUT or additional test equipment.

Operation**Delay Alignment:**

The Synch Clock is the internal timing reference of the M8195A. After power on and after each DAC sample rate change, the M8195A performs an internal delay alignment. This delay alignment ensures that the latency from a synchronously applied Trigger or Event signal is 157 Synch Clock cycles.

Synchronous operation:

Synchronous operation means that the M8195A is started synchronously with an externally applied trigger. Also, sequencing is controlled synchronously by externally applied Trigger or Event signals. In order to operate the M8195A synchronously, the SyncClock must be output at the Reference Clock Out, which can be done by setting internal switches accordingly. The Trigger and Event signal must meet set-up and hold timing requirements as specified in the data sheet of the M8195A. The latency (Trigger In to DATA_OUT or Event In to DATA_OUT) through the M8195A has no variation.

Asynchronous operation:

Asynchronous operation means that the M8195A is started asynchronously with an externally applied trigger. Also, sequencing is controlled asynchronously by externally applied Trigger or Event signals. For asynchronous operation, it is not required to output the SyncClock at the Reference Clock Out and consistently there are no set-up and hold timing requirements to be met. The latency (Trigger In to DATA_OUT or Event In to DATA_OUT) through the M8195A has a small uncertainty. Please refer to the data sheet for the Delay accuracy specification

1.5.3 Delay Adjust

The variable delay is used in order to compensate for e.g. external cable length differences as well as the initial skew. The variable delay has a very high timing resolution. Modifying the variable delay always affects the delay of all four Data Outputs.

Setting the variable delay to e.g. 10 ps has following effects:

Data Out 1, Data Out 2, Data Out 3, and Data Out 4 are delayed by 10 ps with respect to Trigger/Gate Input or Event Input.

Data Out 1, Data Out 2, DataOut 3, and Data Out 4 are delayed by 10 ps with respect to the internal Sync Clock. Note that the Sync Clock is the M8195A timing reference that can be output at Ref Clk out.

In case the M8195A is sourced from Ref CLk In (or the AXIe backplane), Data Out1, Data Out 2, Data Out 3, and Data Out 4 are delayed by 10 ps with respect to Ref CLk In (or the AXIe backplane).

In case the M8197A synchronization module is used to configure a synchronous system of multiple M8195A AWGs, the variable delay can be used to align the Data Out among individual M8195A AWGs.

1.5.4 Extended Memory Configuration

The drawing below provides a more detailed overview regarding the extended memory configuration.

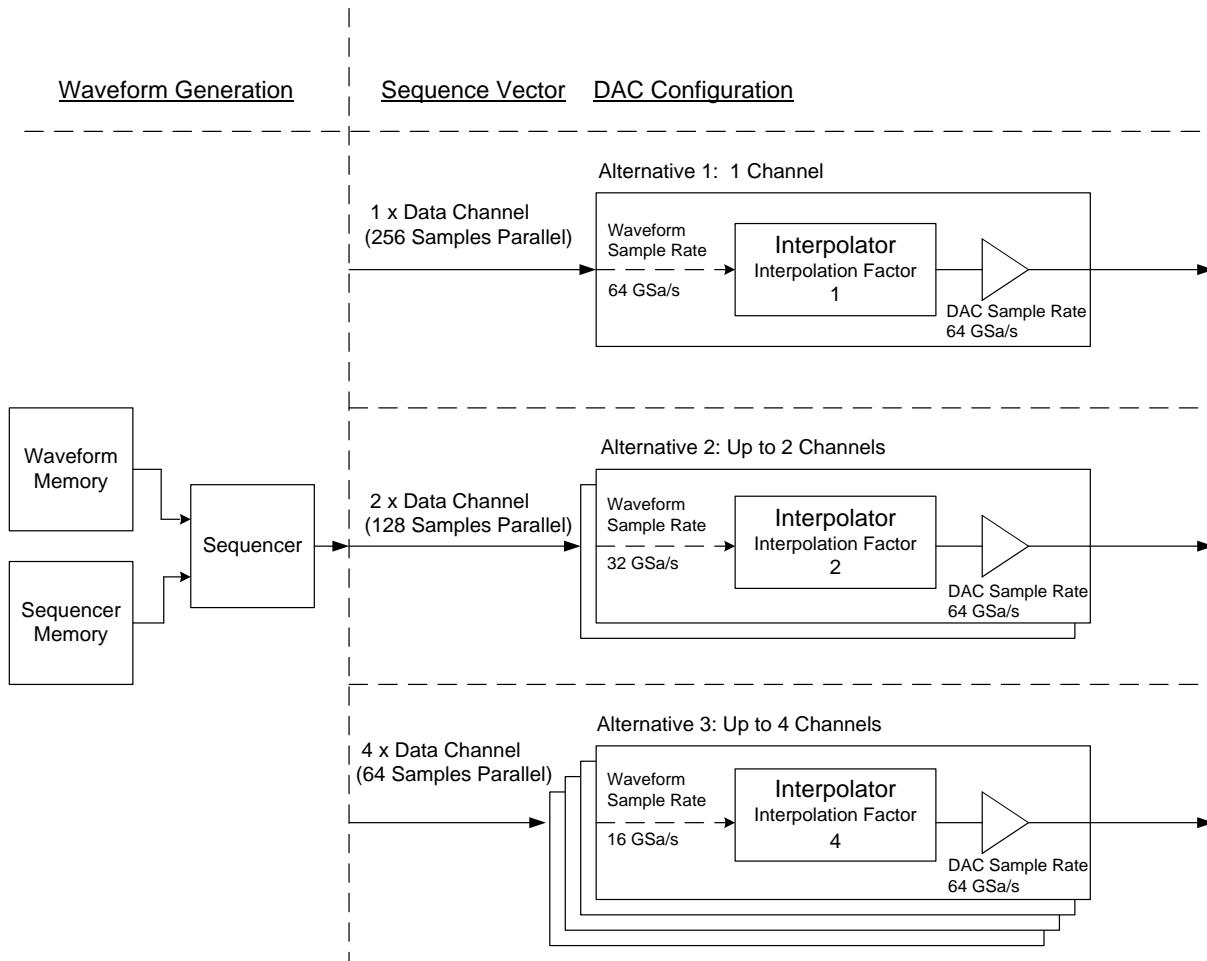


Figure 6: Extended memory configuration

The sequencer generates an ongoing stream of vectors containing 256 parallel samples of 8 bits based on the waveform data and sequencer instructions which are stored in the corresponding memories. Depending on the sample rate divider, which corresponds to the interpolation factor mentioned in the above picture, 3 alternative configurations are possible of how the vector of 256 parallel samples is used to source up to 4 channels. The selected extended memory configuration is exclusive, mixed modes are not possible.

Definitions

Waveform Sample Rate:

The Waveform Sample Rate is the sample rate before the interpolators. Depending on the Sample Rate Divider, this sample rate differs from the DAC Sample Rate which is always in the range of 53.76 GSa/s ... 65 GSa/s.

The dependency is:

$$\text{DAC Sample Rate} = \text{Waveform Sample Rate} * \text{Sample Rate Divider}$$

Waveform Granularity:

Depending on the Sample Rate Divider, the 256 sample wide output of the sequencer is divided by 1, 2 or 4. This generates output vectors with a width of 256, 128 or 64 samples. This vector size is called waveform granularity and is the number of samples per channel processed within one sync clock cycle.

1.5.5 Instrument Modes

The following chapters provide an overview of all available instrument modes and show allowed combinations for using Internal Memory, Extended Memory and Marker Channels.

1.5.5.1 Instrument Mode: Single Channel

Option –001 allows the selection of the instrument mode ‘Single Channel’ or ‘Single Channel with Marker’.

The waveform is always sent at channel 1. The digital markers are always sent at channel 3 and 4.

Table 7: Instrument mode single channel

| Memory configuration | Waveform source | Sample memory size | Waveform memory access rate | Mapped to channel |
|---|-------------------|---|---|--|
| One channel internal memory & no marker | Internal Memory 1 | Int 1: 1 MSa | Int 1: 53.76...65 GSa/s | Ch 1: Waveform 1 Ch 2,3,4: Inactive |
| One channel extended memory & no marker | Extended Memory 1 | Ext 1: No -16G: 2GSa With -16G: 16 GSa @ 53.76...65 GSa/s 8 GSa @ 26.88...32.5 GSa/s 4 GSa @ 13.44...16.25 GSa/s | Ext 1: 53.76...65 GSa/s, 26.88...32.5 GSa/s, or 13.44...16.25 GSa/s | Ch 1: Waveform 1 Ch 2,3,4: Inactive |
| One channel extended memory & with marker | Extended Memory 1 | Ext 1: No -16 G: 2 GSa With -16G: 16 GSa @ 53.76...65 GSa/s 8 GSa @ 26.88...32.5 GSa/s 4 GSa @ 13.44...16.25 GSa/s | Ext 1: 53.76...65 GSa/s, 26.88...32.5 GSa/s, or 13.44...16.25 GSa/s | Ch 1: Waveform Ch 2: Inactive Ch 3: Marker 1 Ch 4: Marker 2 |

1.5.5.2 Instrument Mode: Dual Channel

Option -002 allows the selection of the Instrument Mode ‘Single Channel’, ‘Single Channel with Marker’, ‘Dual Channel with Marker’, ‘Dual Channel’, or ‘Dual Channel Duplicate’.

In Instrument mode ‘Dual Channel’ and ‘Dual Channel Duplicate’, no digital markers are available.

Each channel can be enabled and disabled independently from other channels.

When sourcing one channel from extended memory and the other channel from internal memory, the waveform sourced from extended memory is always sent at channel 1.

Table 8: Instrument mode dual channel

| Memory configuration | Waveform source | Sample memory size | Waveform memory access rate | Mapped to channel |
|--|--|--|--|--|
| Two channels internal memory | Internal Memory 1 Internal Memory 4 | Int 1: 512 kSa Int 4: 512 kSa | Int 1: 53.76...65 GSa/s Int 4: 53.76...65 GSa/s | Ch 1: Waveform 1 Ch 2,3: Inactive Ch 4: Waveform 4 |
| One channel extended & one channel internal memory | Extended Memory 1 Internal Memory 4 | Ext 1: No -16G: 2GSa With -16G: 16 GSa @ 53.76...65 GSa/s 8 GSa @ 26.88...32.5 GSa/s 4 GSa @13.44...16.25 GSa/s Int 4: 1 MSa | Ext 1: 53.76...65 GSa/s, 26.88...32.5 GSa/s, or 13.44...16.25 GSa/s Ext 1 = Ext 4: 26.88...32.5 GSa/s or 13.44...16.25 GSa/s | Ch 1: Waveform 1 (extended, only) Ch 2,3: Inactive Ch 4: Waveform 4 (internal, only) |
| Two channels extended memory | Extended Memory 1 Extended Memory 4 | Ext 1 = Ext 4: No -16G: 1GSa per channel. With -16G: 8 GSa @26.88...32.5 GSa/s 4 GSa @13.44...16.25 GSa/s per channel | Ext 1 = Ext 4: 26.88...32.5 GSa/s or 13.44...16.25 GSa/s | Ch 1: Waveform 1 Ch 2,3: Inactive Ch 4: Waveform 4 |
| Two channels extended memory (duplicated) | Extended Memory 1 Extended Memory 2 | Ext 1 = Ext 2: No -16G: 1GSa per channel. With -16G: 8 GSa @26.88...32.5 GSa/s per channel | Ext 1 = Ext 2: 26.88...32.5 GSa/s | Ch 1: Waveform 1 Ch 2: Waveform 2 Ch 3: Waveform 1 (copy of Ch 1) Ch 4: Waveform 2 (copy of Ch 2) |
| One channel extended & one channel internal memory & with marker | Extended Memory 1 Internal Memory 2 | Ext 1: No -16G: 2GSa With -16G: 16 GSa @ 53.76...65 GSa/s 8 GSa @ 26.88...32.5 GSa/s 4 GSa @13.44...16.25 GSa/s Int 2: 1 MSa | Ext 1: 53.76...65 GSa/s, 26.88...32.5 GSa/s, or 13.44...16.25 GSa/s Int 2: 53.76...65 GSa/s | Ch 1: Waveform 1 (extended, only) Ch 4: Waveform 2 (internal, only) Ch 3: Marker 1 Ch 4: Marker 2 |
| Two channels extended memory & with marker | Extended Memory 1 Extended Memory 2 | Ext 1 = Ext 2: No -16G: 1GSa per channel. With -16G: 8 GSa @ 26.88...32.5 GSa/s 4 GSa @13.44...16.25 GSa/s per channel | Ext 1 = Ext 2: 26.88...32.5 GSa/s or 13.44...16.25 GSa/s | Ch 1: Waveform 1 Ch 4: Waveform 2 Ch 3: Marker 1 Ch 4: Marker 2 |

1.5.5.3 Instrument Mode: Four Channel

Option -004 allows the selection of the Instrument Mode ‘Single channel’, ‘Single Channel with Marker’, ‘Dual Channel with Marker’, ‘Dual Channel’, ‘Dual Channel Duplicate’, or ‘Four Channel’.

In Instrument mode ‘Four Channel’, no digital markers are available.

Each Channel can be enabled and disabled independently from other channels.

Table 9: Instrument mode four channel

| Memory configuration | Waveform source | Sample memory size | Waveform memory access rate | Mapped to channel |
|--|--|---|---|---|
| Four channels Internal Memory | Internal Memory 1 Internal Memory 2 Internal Memory 3 Internal Memory 4 | <u>Int 1:</u> 256 kSa <u>Int 2:</u> 256 kSa <u>Int 3:</u> 256 kSa <u>Int 4:</u> 256 kSa | Int 1 = Int 2 = Int 3 = Int 4: 53.76...65 GSa/s | Ch 1: Waveform 1 Ch 2: Waveform 2 Ch 3: Waveform 3 Ch 4: Waveform 4 |
| One channel extended Memory & Three channels internal memory | Extended Memory 1 Internal Memory 2, 3 or 4 | <u>Ext 1:</u> No -16G: 2 GSa With -16G: 16 GSa@53.76...65 GSa/s 8 GSa@26.88...32.5 GSa/s 4 GSa@13.44...16.25 GSa/s <u>Int 2, Int 3, Int 4:</u> 256 kSa | <u>Ext 1:</u> 53.76...65 GSa/s or 26.88...32.5 GSa/s or 13.44...16.25 GSa/s <u>Int 2 = Int 3 = Int 4:</u> 53.76...65 GSa/s | Ch 1: Waveform 1 (extended, only) Ch 2: Waveform 2 Ch 3: Waveform 3 Ch 4: Waveform 4 |
| Two channels extended Memory & Two channels internal memory | Extended Memory 1 Extended Memory 2 Internal Memory 3 Internal Memory 4 | <u>Ext 1 = Ext 2:</u> No -16G: 1 GSa per channel With -16G: 8GSa@26.88...32.5 GSa/s 4GSa@13.44...16.25 GSa/s per channel <u>Int 3 = Int 4:</u> 512 kSa | <u>Ext 1 = Ext 2:</u> 26.88...32.5 GSa/s or 13.44...16.25 GSa/s <u>Int 3 = Int 4:</u> 53.76...65 GSa/s | Ch 1, Ch 2: Waveform 1, 2 (extended, only) Ch 3, Ch 4 : Waveform 3, 4 (internal, only) |
| Three channels extended Memory & One channel internal memory | Extended Memory 1 Extended Memory 2 Extended Memory 3 Internal Memory 4 | <u>Ext 1 = Ext 2 = Ext 3:</u> No -16G: 0.5 GSa/ch With -16G: 4 GSa/ch <u>Int 4:</u> 1 MSa | <u>Ext 1 = Ext 2 = Ext3:</u> 13.44...16.25 GSa/s <u>Int 4:</u> 53.76...65 GSa/s | Ch 1, Ch 2, Ch3: Waveform 1, 2, 3 (extended, only) Ch 4 : Waveform 4 (internal, only) |
| Four channels Extended Memory | Extended Memory 1 Extended Memory 2 Extended Memory 3 Extended Memory 4 | <u>Ext 1 = Ext 2 = Ext 3 = Ext 4:</u> No -16G: 0.5GSa/ch With -16G: 4GSa/ch | <u>Ext 1 = Ext 2 = Ext 3 = Ext 4:</u> 13.44...16.25 GSa/s | Ch 1: Waveform 1 Ch 2: Waveform 2 Ch 3: Waveform 3 Ch 4: Waveform 4 |

2 M8195A User Interface

| | | | |
|------|---------------------------------------|---|-----|
| 2.1 | Introduction | / | 37 |
| 2.2 | Launching the M8195A Soft Front Panel | / | 38 |
| 2.3 | M8195A User Interface Overview | / | 40 |
| 2.4 | Driver Call Log | / | 44 |
| 2.5 | Errors List Window | / | 45 |
| 2.6 | Clock Tab | / | 47 |
| 2.7 | Output Tab | / | 49 |
| 2.8 | Trigger Tab | / | 52 |
| 2.9 | FIR Filter Tab | / | 54 |
| 2.10 | Standard Waveform Tab | / | 57 |
| 2.11 | Multi-Tone Waveform Tab | / | 63 |
| 2.12 | Complex Modulated Waveform Tab | / | 69 |
| 2.13 | Serial Data Waveform Tab | / | 79 |
| 2.14 | Import Waveform Tab | / | 93 |
| 2.15 | Sequence/Control Tab | / | 100 |
| 2.16 | Correction File Format | / | 106 |

2.1 Introduction

This chapter describes the M8195A Soft Front Panel.

2.2 Launching the M8195A Soft Front Panel

There are three ways to launch the M8195A Soft Front Panel:

- Select **Start > All Programs > Keysight M8195 > Keysight M8195 Soft Front Panel** from the **Start Menu**.
- From the Keysight Connection Expert select the discovered M8195 module, right-click to open the context menu and select “**Send Commands To This Instrument**”.
- From the Keysight Connection Expert select the discovered M8195 module, select the “**Installed Software**” tab and press the “**Start SFP**” button.

The following screen will appear:

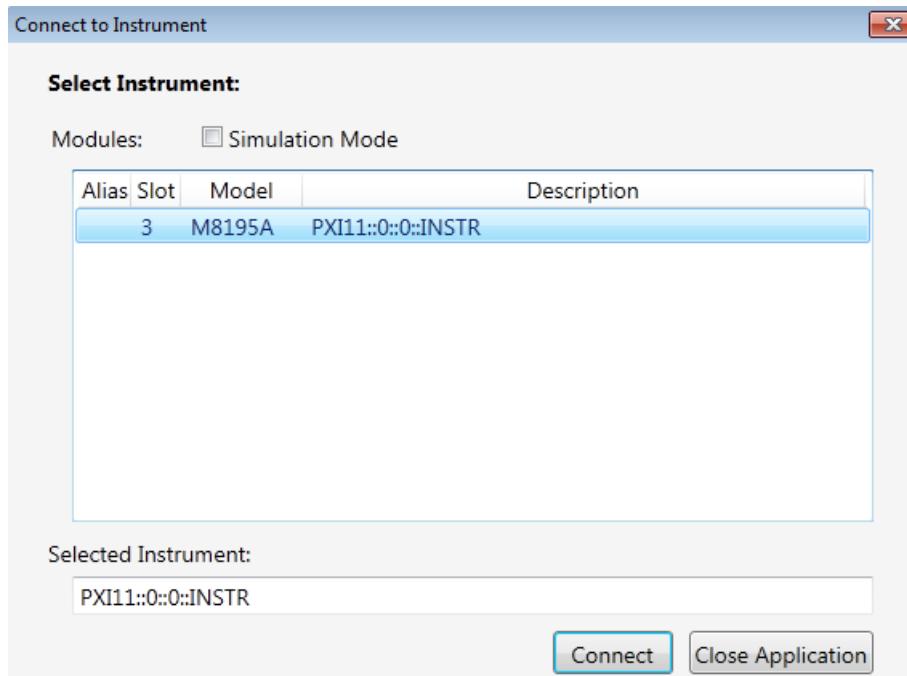


Figure 7: M8195A connected to PC

The instrument selection dialog shows the addresses of the discovered M8195A modules. Select a module from the list and press “Connect”.

If no M8195A module is connected to your PC, you can check “Simulation Mode” to simulate an M8195A module.

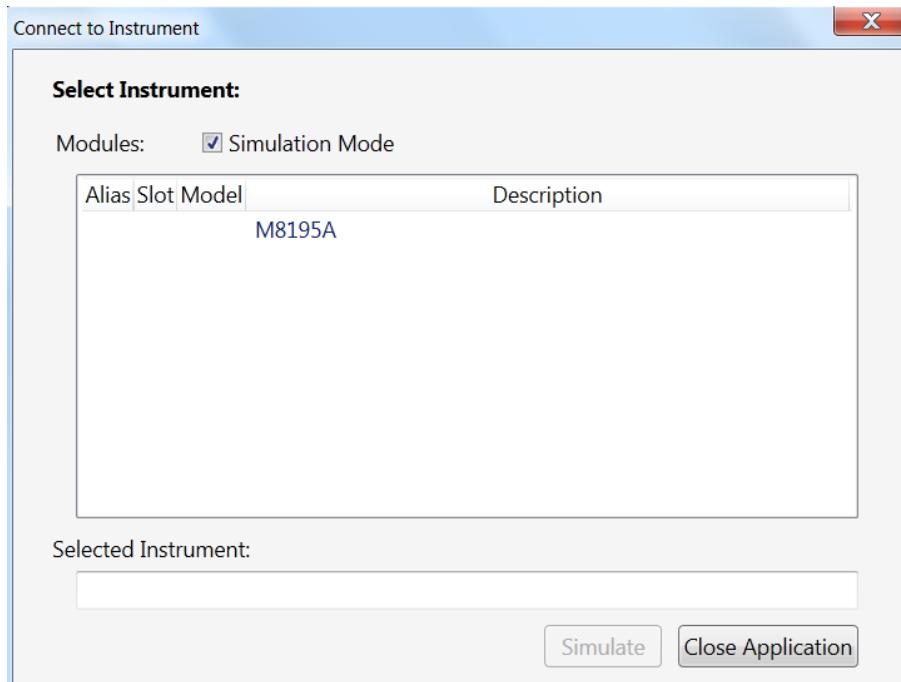


Figure 8: M8195A connected in simulation mode

2.3 M8195A User Interface Overview

The M8195A user interface includes the following GUI items:

- Title Bar
- Menu Bar
- Status Bar
- Tabs (Clock, Output, Standard Waveform, Multi-Tone Waveform, Complex Modulated Waveform, Serial Data Waveform, and Import Waveform)

The detailed information on these GUI items is described in the sections that follow.

2.3.1 Title Bar

The title bar contains the standard Microsoft Windows elements such as the window title and the icons for minimizing, maximizing, or closing the window.

2.3.2 Menu Bar

The menu bar consists of various pull down menus that provide access to the different functions and launch interactive GUI tools.

The menu bar includes the following pull down menu:

- File
- View
- Utilities
- Tools
- Help

Each menu and its options are described in the following sections.

2.3.2.1 File Menu

The File menu includes the following selections:

- File – Connect...
Opens the instrument selection dialog.
- File – Save Configuration As...
Saves configuration as a text file.
- File – Load Configuration...
Load the previously saved configuration file.
- File – Exit
Exits the user interface.

2.3.2.2 View Menu

The View menu includes the following selections:

- View – Refresh
Reads the instrument state and updates all fields.
- View – Hide
Minimizes the GUI to notify icon.

2.3.2.3 Utilities Menu

The Utility menu includes the following selections:

- Utility – Identify
Identify the instrument by flashing the green “Access” LED on the front panel for a certain time.
- Utility – Reset
Resets the instrument, reads the state and updates all fields.
- Utility – Self Test...
Opens a window to start the self-test and display the result after completion.

2.3.2.4 Tools Menu

The Tools menu includes the following selections:

- Tools – Monitor Driver Calls
Opens the [Driver Call Log](#) window.

2.3.2.5 Help Menu

The Help menu includes the following selections:

- User Guide
Opens the User Guide of the M8195A.
- Driver Help
Opens the online help of the IVI-COM and IVI-C drivers.
- Help – Online Support
Opens the instrument's product support web page.
- Help – About
Displays revision information for hardware, software and firmware. Displays the serial number of the connected module.

2.3.3 Status Bar

The Status Bar contains three fields from left to right:

- Connection state
“Not Connected” – No instrument is connected.
“Connected: <Instrument resource string>” – An instrument is connected. The resource string, for example PXI36::0::0::INSTR is displayed.
“Simulation Mode” – No real instrument is connected. The user interface is in simulation mode.
Click this field to open the Instrument Selection Dialog.
- Instrument status
Displays the instrument status, for example “Reset complete” after issuing a reset command. In case of error it displays additional error information.
- Error status
“Error” – The connected instrument reported an error.
“No Error” – No errors occurred.
Click this icon to open the Error List Window.
- Run/Stop button:
The Run/Stop button is used to switch between Run and Program mode.

2.3.4 Clock/Output/Trigger/FIR Filter/Standard Waveform/Multi-Tone Waveform/Complex Modulated Waveform/Serial Data Waveform/Import Waveform/Sequence/Control Tabs

These tabs are used to configure the most important parameters of the M8195A module. They are described in detail in the sections that follow.

2.3.5 Numeric Control Usage

The numeric control is used to adjust the value and units. Whenever you bring the mouse pointer over the numeric control, a tooltip appears which shows the possible values in that range.

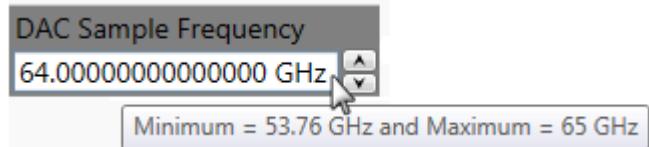


Figure 9: Tooltip showing possible values in the range

The numeric controls can be used in the following ways:

Use the up/down arrows to change the value. The control automatically stops at the maximum/minimum allowed value.

You can increase or decrease the value starting at a specific portion of the value. To do this, place the cursor to the right of the targeted digit and use the up/down arrows. This is especially useful when changing a signal characteristic that is immediately implemented, and observing the result in another instrument. For example, you can change the signal generator's frequency by increments of 10 MHz and observe the measured result in a signal analyzer:

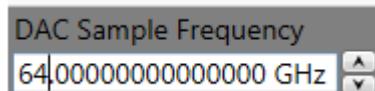


Figure 10: Typing directly into the field

Type directly into the field and press the Enter key. If you enter a value outside the allowed range, the control automatically limits the entered value to the maximum or minimum allowed value.

When you type the value, you can type the first letter of the allowed unit of measure to set the units. For example, in the Frequency control you can use "H", "K", "M", or "G" to specify hertz, kilohertz, megahertz, or gigahertz, respectively. (The control is not case sensitive.)

The controls allow scientific notation if it is appropriate to the allowed range. Type the first decimal number, enter an "E", and omit any trailing zeroes. For example, in the Frequency control you can type 2.5e+9 and press [Enter] to set the frequency to 2.5 GHz. (The plus sign is automatically inserted if it is omitted.)

2.4 Driver Call Log

Use this window to inspect the sequence of IVI driver calls and SCPI commands used to configure the M8195A module.

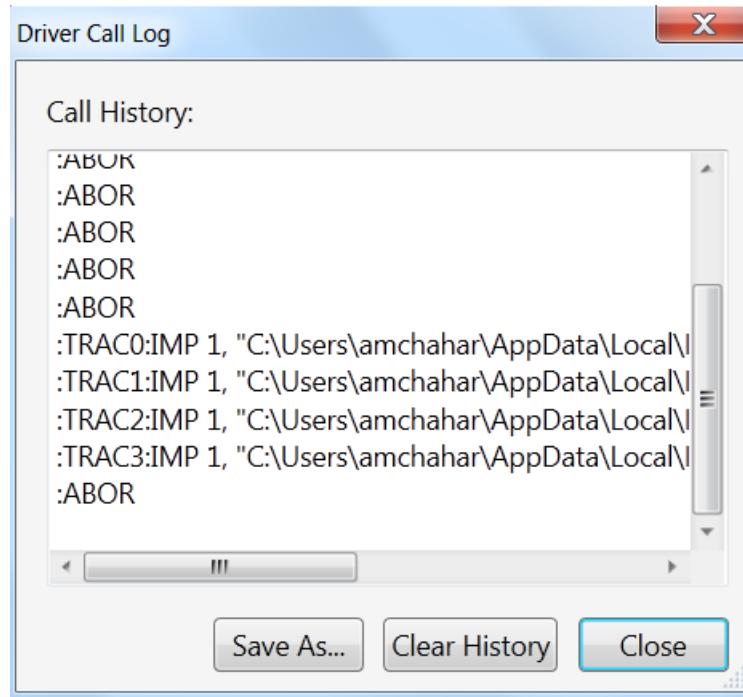


Figure 11: Driver call log window

It has the following buttons:

- Save As...
Saves the Driver Call Log as a text file.
- Clear History
Clears the Driver Call Log.
- Close
Exits the window.

2.5 Errors List Window

Use this window to view errors, warnings, and information.



Figure 12: Errors list window

It has the following controls, signs, and columns:

- Open On Error

Select this check box to automatically open the errors list window whenever an error occurs. This window will show error details i.e. time stamp and description.

-  (Clear All)

Use this option to clear all the errors from the errors list window.



Use this toggle option to respectively show or hide the errors list window. It also shows total number of errors in the list. When the window has no errors, the green tick icon will appear.

-  (Error)

This icon represents an error.

-  (Warning)

This icon represents a warning.

-  (Information)

This icon represents an information.

- Time Stamp

This column lists the time stamp of individual errors in the format DD/MM/YYYY HH:MM:SS.

- Description

This column provides the description of individual errors.



This drop down list provides window control options like:

- Float
- Dock
- Auto Hide
- Close

2.6 Clock Tab

Use this tab to configure the sample clock and the reference clock of M8195A module. The sample clock for all four Digital to Analog Converters (DAC) of the four channels is identical. It allows user to configure clock source, reference clock range and frequency, and DAC sample frequency.

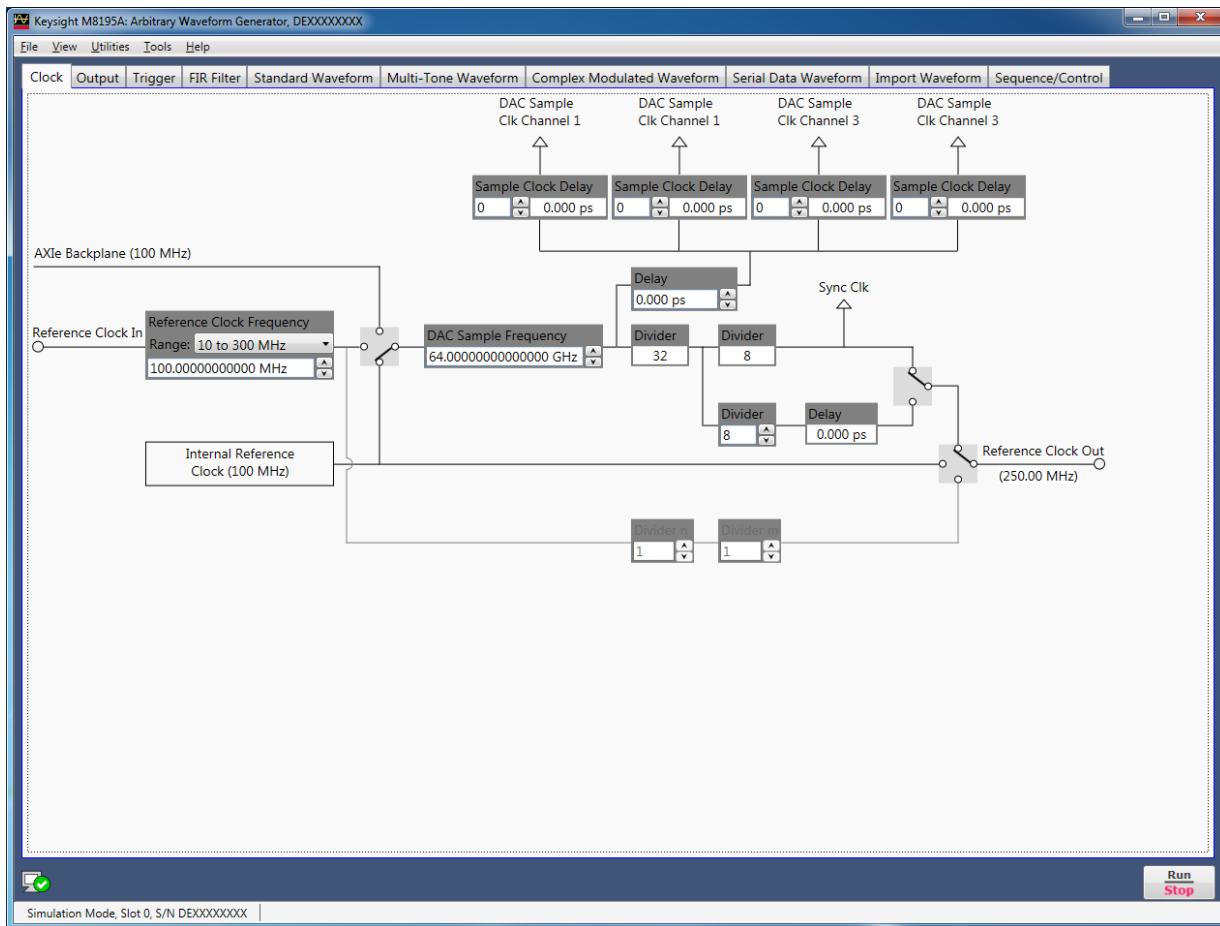


Figure 13: Clock tab

- Reference Clock Selection Switch
This switch selects between the different reference clock sources.
 - Internal 100 MHz: Reference from internal oscillator
 - Internal Backplane 100 MHz: Reference from AXIe Backplane
 - External: Reference from Ref Clock In
- DAC Sample Frequency
This field specifies the DAC sample frequency for all the channels.
The range is 53.76 to 65 GHz.
- Module Delay
This field specifies the module delay for all the channels.
The range is 0 to 10 ns.
- Sample Clock Delay
This field specifies the sample clock delay individually per channel as an integral number of DAC sample clocks.
The range is 0 to 95 DAC sample clocks.
- Reference Clock Frequency and Range
This field allows to select a reference clock frequency range among the two options 10 to 300 MHz and 210 MHz to 17 GHz. Further, it provides a field to enter the frequency value within the selected range.
- Reference Clock Out Switches
These two switches allow selecting reference clock out source depending on reference clock input source.
- Frequency Dividers
There are in total five frequency dividers in the path to the Reference Clock Out. Three of them can be changed.

2.7 Output Tab

Use this tab to configure the Data Outputs (Channel 1, Channel 2, Channel 3, and Channel 4) of the M8195A AWG module.

- The M8195A has six different modes of operation:
 - Single Channel: If this mode is selected, Channel 1 is used to generate data; Channel 2, Channel 3, and Channel 4 are disabled.
 - Single Channel with Markers: If this mode is selected, Channel 1 is used to generate data, and channel 3 and 4 are used to generate digital markers. Channel 2 is disabled. The memory mode for Channel 1 is 'Extended' and cannot be changed.
 - Dual Channel: If this mode is selected, Channel 1 and Channel 4 are used to generate data. Channel 2 and Channel 3 are disabled. This mode is selectable, if option 002 or 004 is present.
 - Dual Channel Duplicate: If this mode is selected, Channels 1, 2, 3 and 4 are used to generate a signal. Channel 3 generates the same signal as channel 1. Channel 4 generates the same signal as channel 2. The memory mode for Channels 1 and 2 is not configurable and is always 'Extended' memory. This mode is selectable, if option 002 or 004 is present.
 - Dual Channel with Markers: If this mode is selected, Channels 1 and 2 are used to generate a signal. Channel 1 has two markers output on channel 3 and 4. Channel 2 can generate a signal without markers. The memory mode for Channel 1 is not configurable and is always 'Extended' memory. This mode is selectable, if option 002 or 004 is present.
 - Four Channel: This mode is only selectable, if option 004 is installed. If this mode is selected, all four channels can be used to generate data.
- Sample Rate Divider (Extended Mem):

The speed of operation of the extended memory is adjustable using the parameter 'Sample Rate Divider (extended memory)'. Possible values are 1, 2, and 4. The sample rate divider is identical for all channels that are sourced from extended memory. In case the sample rate divider is adjusted to two or four, the FIR filters are used as interpolation filters by factors of two or four. The interpolation is necessary as the DAC always operates in the range 53.76 GSa/s ... 65 GSa/s.

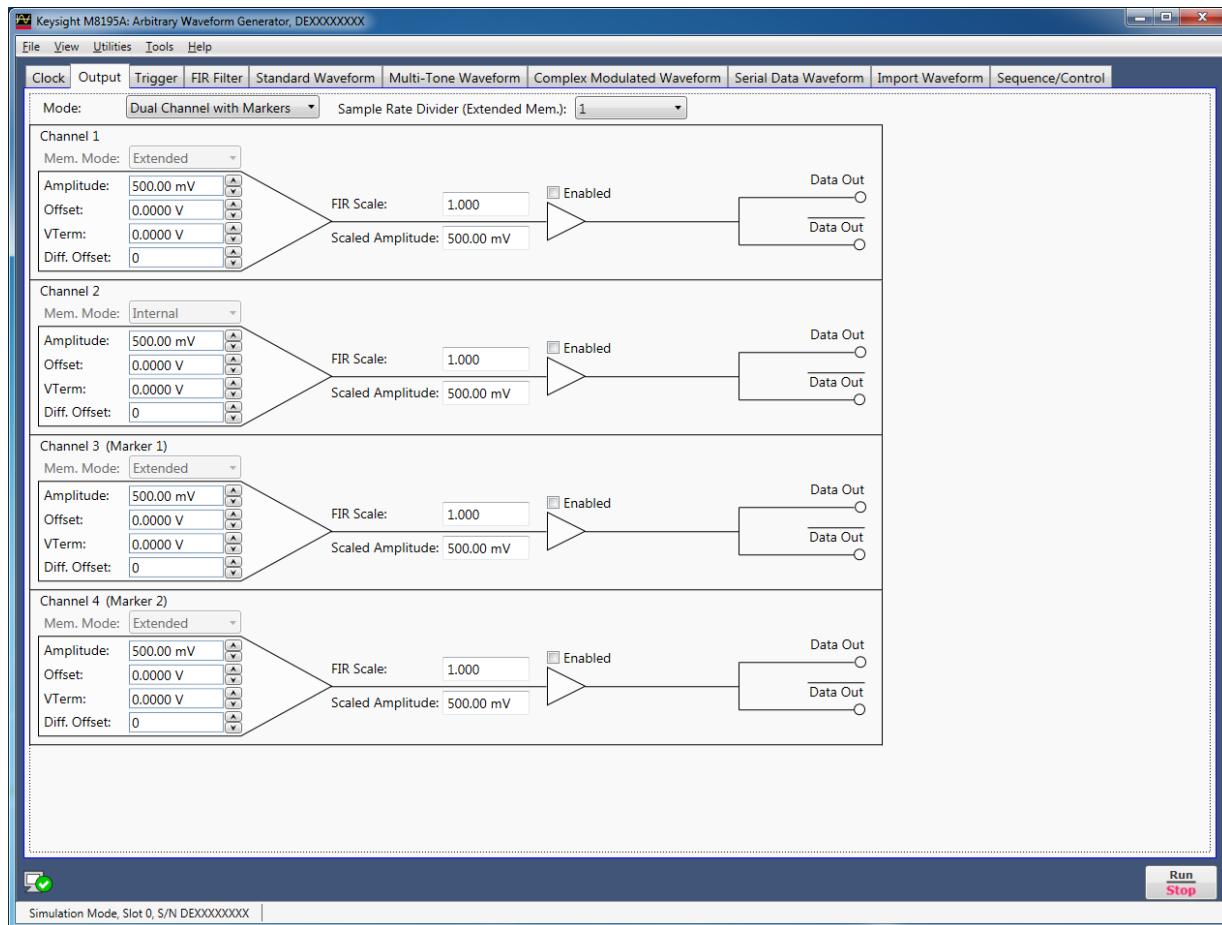


Figure 14: Output tab

Each channel has the following input fields:

- Memory mode
Specifies the memory mode of the channel. Available options are ‘Internal’ (default) and ‘Extended’.
The Sample Rate of all the four Digital to Analog Converters (DAC) is selectable between 53.76 GSa/s ... 65 GSa/s. The Internal waveform memory always operates at the sample rate. The Extended waveform memory can operate at sample rate 53.76 GSa/s ... 65 GSa/s or at one half of the sample rate 26.88 GSa/s ... 32.5 GSa/s or at one fourth of the sample rate 13.44 GSa/s ... 16.25 GSa/s. The speed of operation of the extended memory is adjustable using the parameter ‘Sample Rate Divider (extended memory)’.
- Amplitude
Specifies the amplitude of the output signal.
- Offset
Specifies the offset of the output signal.
- Diff. Offset (Differential Offset)
Specifies the differential offset of the output signal.
- V Term (Termination Voltage)
Specifies the termination voltage.
- Output status indicator. This indicator reflects the color of the ‘**Channel**’ LED on the front panel:
 - It is ‘OFF’ when the channel is disabled and no overload condition at this channel has been detected.
 - It is ‘GREEN’ if the channel is enabled and no overload condition at this channel has been detected.
 - It is ‘RED’ if the internal protection circuit of that channel has detected an overload condition. Potential overload conditions are e.g. an external short to GND or 50 Ohm termination to a wrong externally applied termination voltage VTerm. In case an overload condition is detected, remove the overload condition of the test set-up and enable the channel.
- Output enable switch
If set to enabled position, the generated signal is present at the output.
- FIR Scale
Shows the currently active scaling factor. This parameter can be adjusted in the FIR filter tab.
- Scaled Amplitude
Shows the effective output amplitude after the ‘FIR Scale’ had been applied.

2.8 Trigger Tab

Use this tab to configure the trigger and event input parameters. It allows user to send software triggers and events to the module.

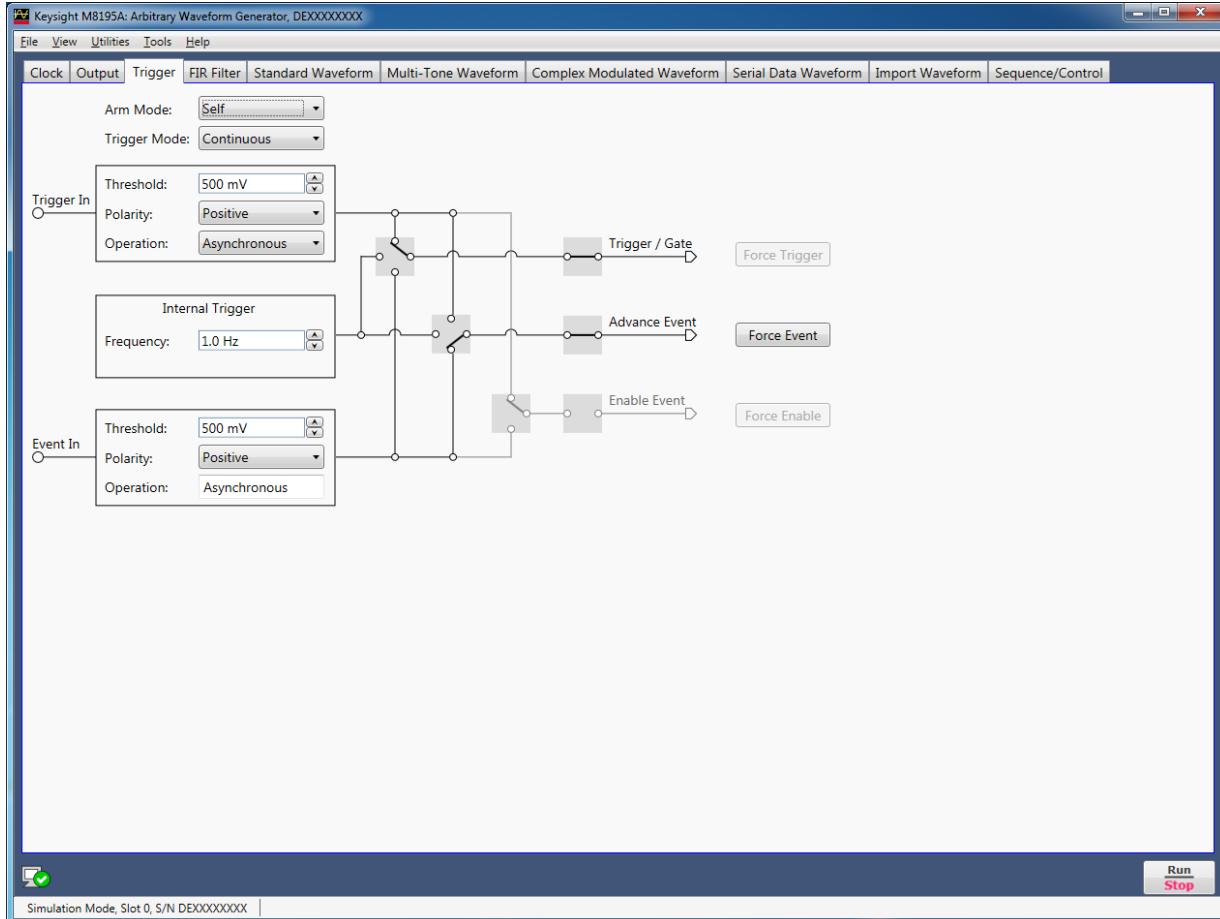


Figure 15: Trigger tab

This tab has the following configurable fields:

- Arm mode
 - Armed – Signal generation starts when an “enable” event is received as defined by the trigger mode..
 - Self – Signal generation starts as defined by the trigger mode.
- Trigger mode
 - Continuous – Signal generation starts immediately after pressing the Run button. No trigger needed.
 - Triggered – Signal generation starts after a trigger is received.
- Gated – Signal generation starts when a rising edge is received on the trigger input and pauses when a falling edge is received. Signal generation restarts after the next rising edge.
- Threshold
Specifies the threshold voltage for a software trigger or event.
- Polarity
Specifies the polarity for a software trigger or event viz. Negative, Positive, or Either.
- Operation
Specifies whether the trigger or event operation is Synchronous or Asynchronous. Operation mode is same for both trigger and event input.
- Frequency
Specifies the frequency for internal trigger.
- Force Trigger
Use this button to send a software trigger to a channel.
- Force Event
Use this button to send a software event to a channel.
- Force Enable
Use this button to send a software “enable” to a channel.

2.9 FIR Filter Tab

Use this tab to configure the FIR filter coefficient values for Channel 1, Channel 2, Channel 3, and Channel 4. The number of coefficients depends on the extended memory sample rate divider (see [Sample Rate Divider](#) ‘extended memory’ in Output tab).

For sample rate divider 1, 2, and 4 the number of coefficients are 16, 32, and 64, respectively.

For complete details, refer to the section “[Theory of Operation](#)”.

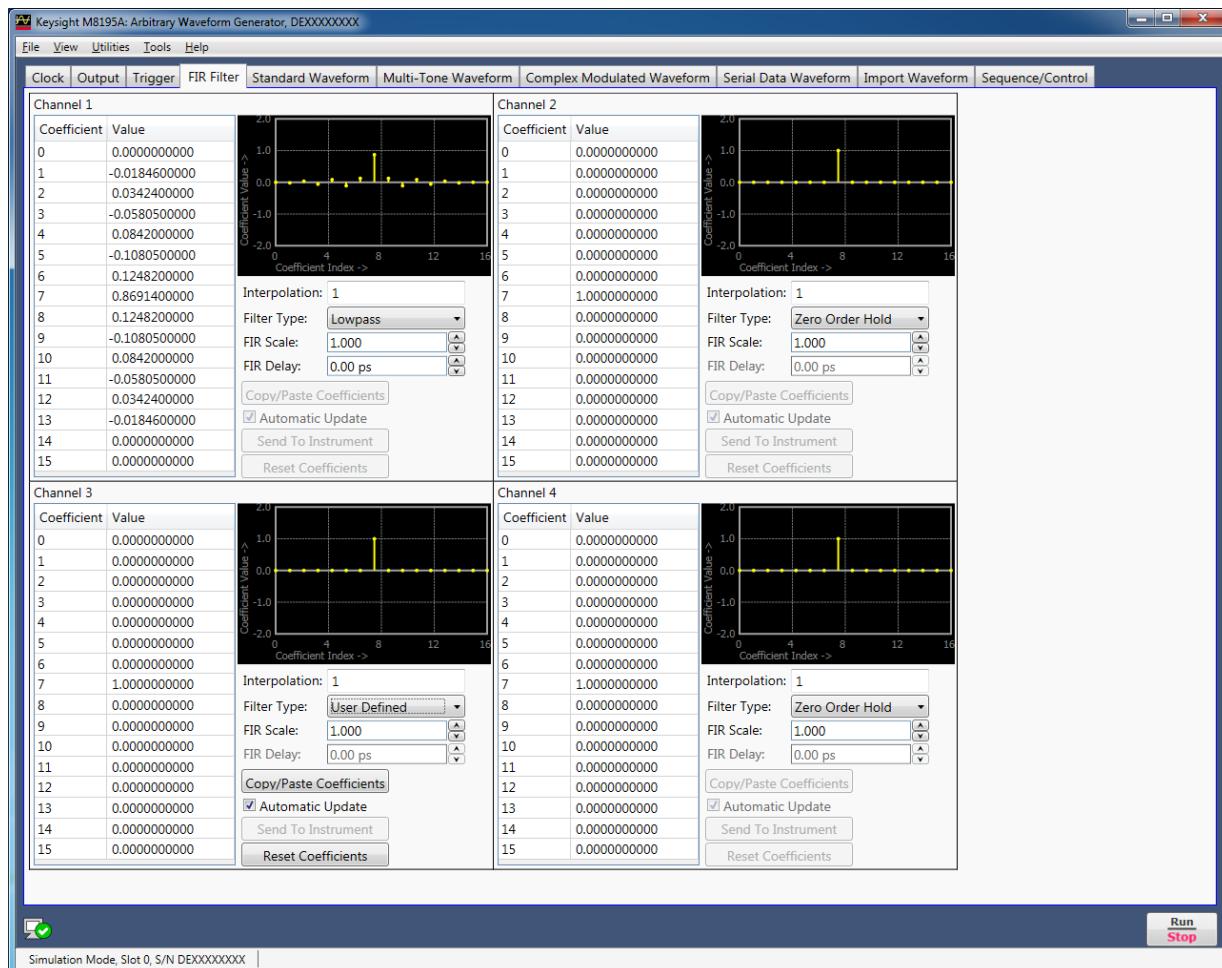


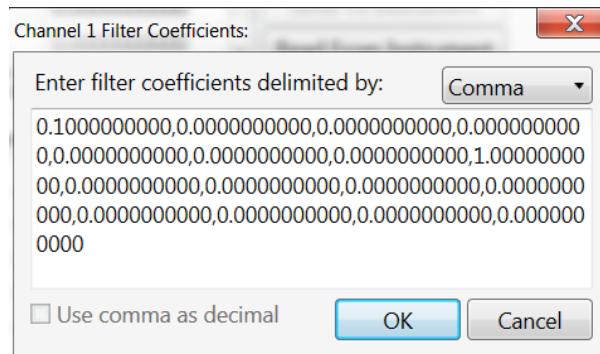
Figure 16: FIR Filter tab

This tab has the following fields and buttons:

- **Coefficient Values**
Specifies the coefficient values for channel 1, channel 2, channel 3, and channel 4. To edit a coefficient value, double-click on the value field. The range for a coefficient value is -1 to 1.

| Channel 1 | |
|-------------|--------------|
| Coefficient | Value |
| 0 | 0.0000000000 |
| 1 | 0.0000000000 |
| 2 | 0.0000000000 |

- **Copy/Paste Coefficients**
Use this button to bulk copy/paste coefficient values for a specific channel. Clicking the button opens a dialog box displaying all the coefficient values ready to be copied or replaced by other values.
The dialog box provides an option to view the values separated by Comma, Semi Colon, Space, Tab, or Enter.



- **Automatic Update**
If checked, the FIR coefficients are updated in the hardware whenever they change (change of Sample Rate Divider, Filter Type, FIR Scale, FIR Delay, or manual change of the coefficients).
If not checked, the FIR coefficients are updated in the hardware only when "Send To Instrument" is pressed.
- **Reset Coefficients**
This button resets the coefficient values for a certain channel to default.
- **Send To Instrument**
This button sends the coefficient values to the instrument.

Interpolation

- This shows the interpolation factor for a channel.

Filter Type

- The following FIR filter types for a channel can be used when the interpolation factor is 1.
 - Lowpass – equiripple lowpass filter with a passband edge at 75% of Nyquist
 - Zero-order hold filter
 - User-defined filter
- The following FIR filter types for a channel can be used when the interpolation factor is 2.
 - Nyquist filter (half-band filter) with rolloff factor 0.2
 - Linear interpolation filter
 - Zero-order hold filter
 - User-defined filter
- The following FIR filter types for a channel can be used when the interpolation factor is 4.
 - Nyquist filter (quarter-band filter) with rolloff factor 0.2
 - Linear interpolation filter
 - Zero-order hold filter
 - User-defined filter

FIR Scale

- FIR filter scaling factor for a channel.. The range is between 0 and 1.

FIR Delay

- FIR filter delay for a channel.. The delay is only adjustable for the filter types ‘Lowpass’, ‘Nyquist’, and ‘Linear Interpolation’.

2.10 Standard Waveform Tab

Use this tab to create a variety of standard waveform types. It provides the controls which allow the complete definition of signal generation parameters for the following waveform shapes:

- Sinusoidal
- Square with linear transitions
- Square with cosine-shaped transitions
- Triangle
- Sinc ($\text{Sin } x/x$)
- Bandwidth-limited Gaussian noise

The standard waveform tab allows you to generate signals for both direct and I/Q data generation modes. It also provides a graphic waveform preview functionality, which can be used to validate created signals before sending them to the instrument. The created signals can also be stored in a file for later use. The application takes care of handling the requirements and limits of the target hardware in aspects such as maximum and minimum record lengths and sampling rate and record length granularity. As a result, the signals designed in this tab will be always feasible to be generated by the instrument and free of distortions such as wrap-around or timing artifacts, even if the signal is generated in looped mode.

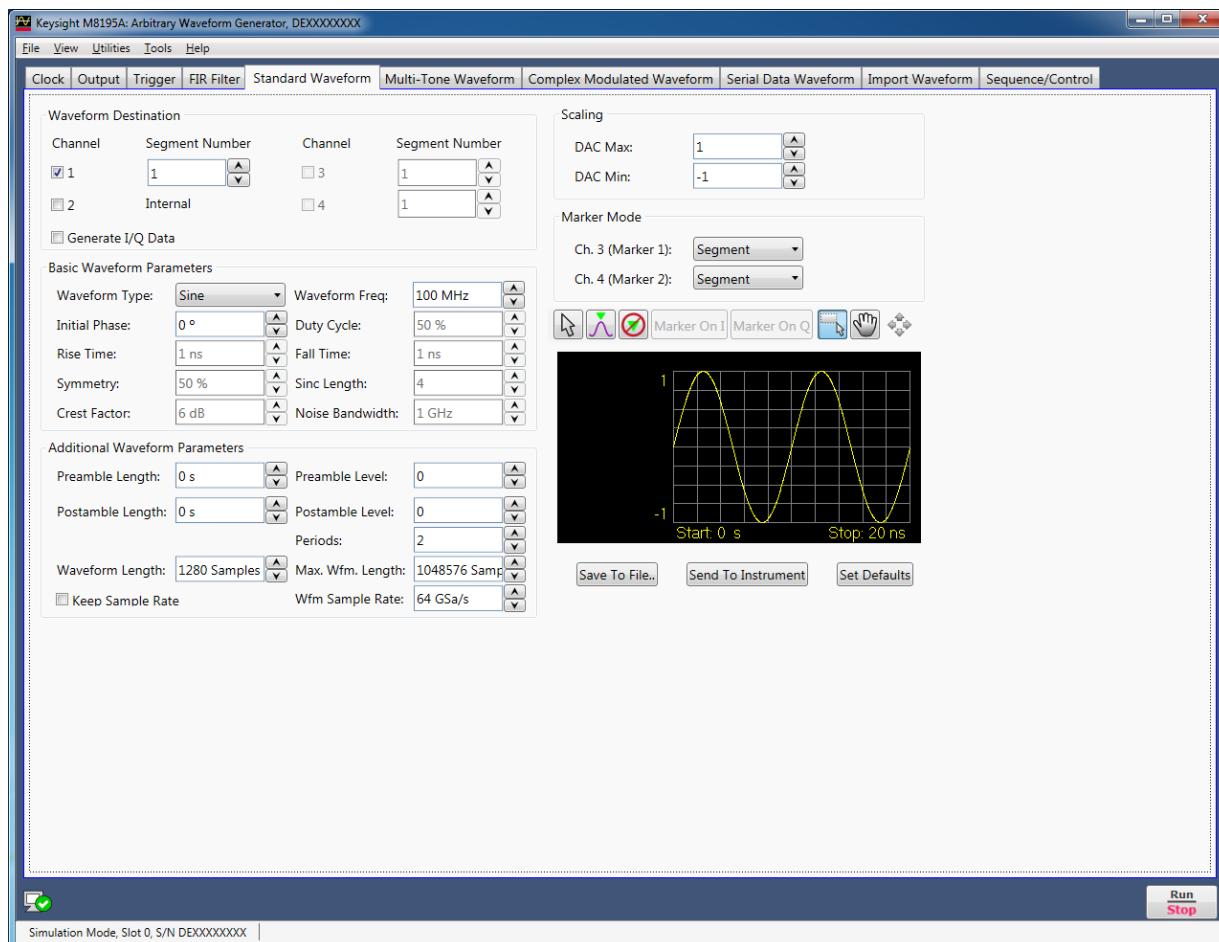


Figure 17: Standard waveform tab

This tab has the following controls:

Waveform Destination Section

- Channel
Independent checkboxes allow the definition of standard waveforms for Channel 1, Channel 2, Channel 3, or Channel 4. One of the boxes is always checked. When pressing the 'Send To Instrument' button, the waveform is sent to all channels that are checked.
- Generate I/Q Data
If checked, baseband (I/Q) signals will be generated. The effect of this control depends on the selected signal type. For Sinusoidal waves, the resulting complex signal will be a single spectral line located at positive or negative frequencies. This implies that users can type negative numbers into the "Waveform Freq." field. . For noise, the resulting complex signal will be a limited-bandwidth Gaussian noise with uncorrelated positive and negative frequency components. All the other waveform types result in the same signal being generated by both I and Q assigned channels.
I/Q selection toggle buttons for each channel will be shown when the Generate I/Q Data checkbox is checked. In-Phase (I) and Quadrature (Q) components can be independently assigned to each channel.
- Segment Number
Target segment for each channel can be defined independently. This field is configurable only for channels sourced from 'extended' memory. The segment range is 1 to 16777216. For channels sourced from 'Internal' memory, the segment is always set to 1, and it displays the text 'Internal'.

Basic Waveform Parameters Section

- Waveform Type:
The following waveform types are available:
 - Sine: Sinusoidal waveform. Frequency and Initial Phase parameters can be defined for this waveform type using the corresponding controls. If the Generate I/Q checkbox is checked, two sine waves with a 90° phase difference will be assigned to the I and Q components.
 - Square_Linear: Square signal with linear transitions. Frequency, Rise Time, Fall Time, Duty Cycle, and Initial Phase parameters can be defined for this waveform type using the corresponding controls.
 - Square_Cos: Square signal with cosine shaped transitions. Frequency, Rise Time, Fall Time, Duty Cycle, and Initial Phase parameters can be defined for this waveform type using the corresponding controls.
 - Triangle: Triangular waveform with linear transitions. Frequency, Symmetry, and Initial Phase parameters can be defined for this waveform type using the corresponding controls.
 - Sinc: Sin x/x waveform. Frequency, Symmetry, Sinc Length, and Initial Phase parameters can be defined for this waveform type using the corresponding controls.

- Noise: Gaussian noise with limited bandwidth. Frequency, Crest Factor, and Noise Bandwidth parameters can be defined for this waveform type using the corresponding controls. If the Generate I/Q checkbox is checked, two uncorrelated noise waveforms will be assigned to the I and Q components.
- Waveform Frequency

Repetition rate for one cycle of the standard waveform. It is always a positive number except when Signal Type is set to Sine and the Generate I/Q Data checkbox is checked. In this case, frequency may be negative so the resulting SSB (Single-Side Band) will be located over or below the carrier frequency.
- Initial Phase

The phase within a normalized cycle of the standard waveform for the first sample in the segment.
- Duty Cycle

The relative width as a percentage of the mark and the space sections of square waves.
- Rise Time

The transition time (10%–90%) for the rising edge in square waveforms.
- Fall Time

The transition time (10%–90%) for the falling edge in square waveforms.
- Symmetry

For both triangular and sinc waveforms, it marks the location as a percentage of the positive highest peak within a period of the basic signal.
- Sinc Length

The number of zero crossings in a single period for the sinc waveform type.
- Crest Factor

The peak-to-average power ratio in dBs for Noise samples before low-pass filtering. Actual crest factor in the final signal after filtering will be higher.
- Noise Bandwidth

Baseband noise bandwidth for Noise waveforms. For IQ modes, noise bandwidth around the carrier frequency will be twice this parameter.

Additional Waveform Parameters Section

- Preamble Length
The duration of a DC section before the defined Standard waveform starts.
- Preamble Level
The level for the DC section before the defined Standard waveform starts. Acceptable range for this parameter is -1/+1, being the full dynamic range of the instrument's DAC.
- Postamble Length
The duration of a DC section after the defined Standard waveform stops.
- Postamble Level
The level for the DC section after the defined Standard waveform stops. Acceptable range for this parameter is -1/+1, being the full dynamic range of the instrument's DAC.
- Keep Periods
This checkbox is only available when “Keep Sample Rate” is selected. When this option is selected, the waveform calculation algorithm preserves the user-defined number of periods.
- Set WL to Max
This checkbox is only available when “Keep Sample Rate” is selected. When this option is selected, the waveform calculation algorithm always takes the maximum waveform length as defined in the “Max. Wfm. Length”. As the waveform length must always be identical for all four channels, it is recommended to check the “Set WL to Max” box in case different waveforms shall be downloaded to different channels.
- Periods
The number of repetition of single periods of the standard waveform within the target segment. This parameter is set automatically when Frequency is changed and preamble and postamble lengths are set to zero in order to obtain the best timing accuracy and meet the record length granularity requirements.
- Waveform Length
The length in samples of the resulting segment. It may be set within acceptable limits and it may be calculated automatically to properly implement other signal and instrument parameters such as sampling rate.
- Max. Wfm. Length
Maximum waveform length must be used to force the resulting waveform to be shorter than or equal to a user-set limit.
- Keep Sample Rate
This check box preserves the sampling rate to a user-defined value no matter how any other signal parameters may be defined. Keeping the sampling rate to a fixed value may be necessary when multiple waveforms are created to be used in a sequence or scenario. The “Set WL to Max” check box gets activated when this check box is checked.
Set WL to Max. This check box forces the usage of the number of samples defined in the “Max. Wfm. Length” numeric entry field. Some waveform parameters may be adjusted to make sure that continuous play-back of the waveform is seamless.

Marker Mode

These controls are available when the “Single Channel with Marker” or “Dual Channel with Marker” mode is selected in the Output tab.

- Ch. 3 (Marker 1)
Marker 1 is output on Channel 3. Signaling the beginning of each segment may be activated (Segment selection) and deactivated (None selection).
- Ch. 4 (Marker 2)
Marker 2 is output on Channel 4. Signaling the beginning of each segment may be activated (Segment selection) and deactivated (None selection).

Scaling Section

- DAC Max

Standard waveforms may occupy a limited range of the DAC’s full scale. This parameter sets the maximum level. If set to a lower level than DAC Min, this will be automatically set to the same level. Acceptable range for this parameter is -1/+1, being the full dynamic range of the instrument’s DAC.

- DAC Min

Standard waveforms may occupy a limited range of the DAC’s full scale. This parameter sets the minimum level. If set to a higher level than DAC Max, this will be automatically set to the same level. Acceptable range for this parameter is -1/+1, being the full dynamic range of the instrument’s DAC.

Preview Section

- Waveform Preview Toolbar

The waveform preview toolbar includes the icons to preview the waveform. The following icons are available:

| | |
|---|---|
|  | Uses the mouse to control the marker. The respective position of marker at X and Y axis are displayed on the top of waveform. |
|  | Takes the marker to the peak position |
|  | Sets the marker on the I data part of the waveform |
|  | Sets the marker on the Q data part of the waveform |
|  | Turns off the marker |
|  | Provides zoom functionality. Use the mouse pointer to select the area on waveform that you want to zoom. Once done, you can click Auto scale icon to zoom out the waveform. |
|  | Uses the mouse pointer to move the waveform around. You can also use the pan tool when the waveform is zoomed in. |
|  | Auto scale the waveform |

- Save To File...

Signals can be stored in files in whether BIN (for non IQ modes) or IQBIN (for IQ modes) formats. These files may be reused within the Import Waveform tab.
- Send To Instrument

Signal will be transferred to the selected segments of the selected channels. The previous running status for the target instrument will be preserved but sampling rate may be modified depending on the waveform requirements.
- Set Default

All the standard waveform parameters are set automatically to their corresponding default values.

2.11 Multi-Tone Waveform Tab

Use this tab to create signals made-up of multiple tones, either equally or arbitrarily spaced. It also allows for the definition of a frequency interval without tones (or notch) for NPR (Noise Power Ratio) testing. Amplitudes and phases of the individual tones can be corrected through correction factor files defined by the user. The Multi-Tone tab allows you to generate both RF and baseband (I/Q Data) signals. It also provides a graphic waveform preview functionality, which can be used to validate the location and amplitudes of the tones in the signal before sending it to the instrument or be stored in a file for later use. The signal's crest factor or Peak-to-Average Power Ratio (PAPR) is also shown. The application handles requirements and limits of the target hardware in aspects such as maximum and minimum record lengths, sampling rate, and record length granularity. As a result, generation of signals designed in this tab will always be feasible through the instrument, and they will be free of distortions such as wrap-around or timing artifacts, even if they are generated in looped mode.

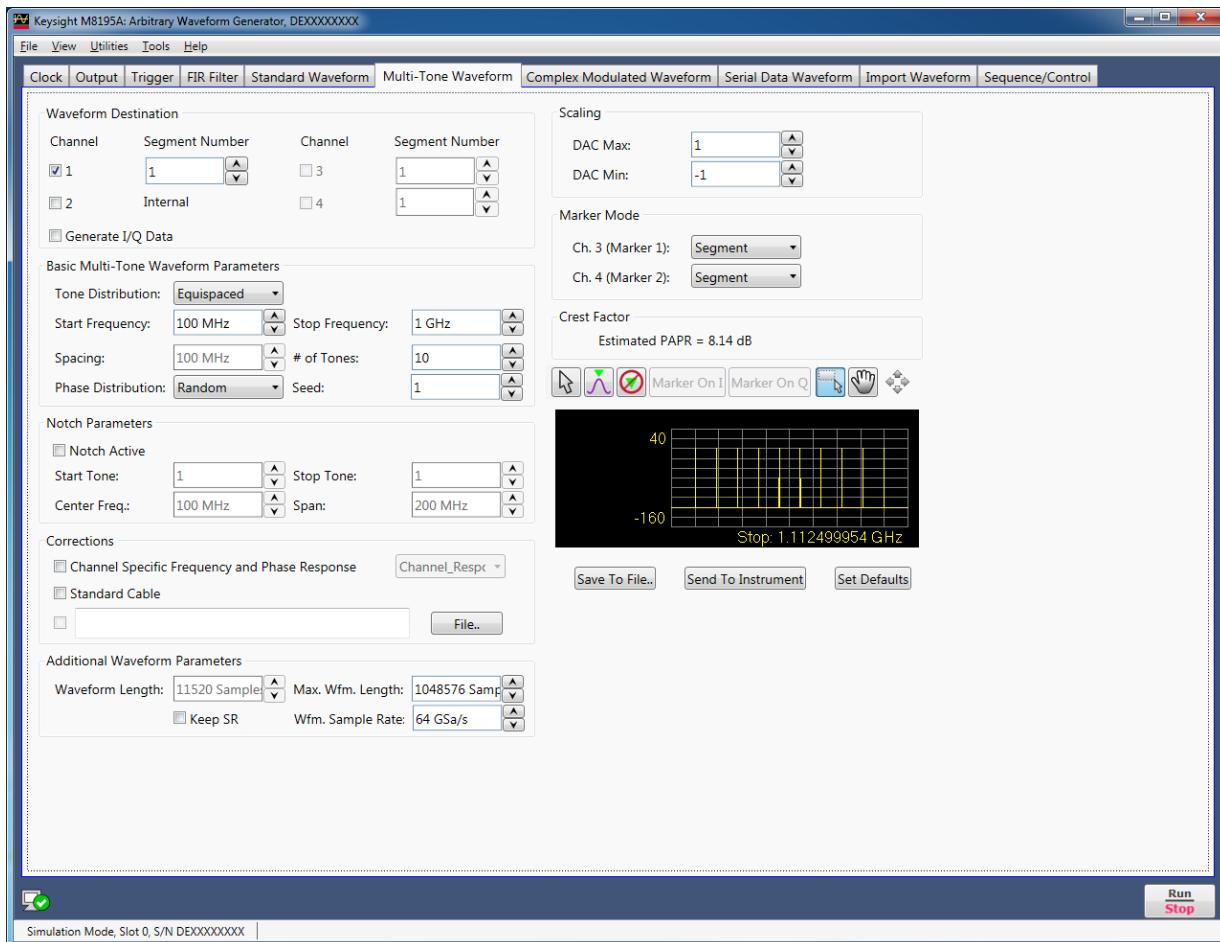


Figure 18: Multi-Tone waveform tab

There are two basic operation modes for the definition of equally spaced or arbitrarily distributed tones respectively. The selection between the two modes is made through the “Tone Distribution” drop-down list. This control affects the contents of the “Basic Multi-Tone Waveform Parameters” section of the user interface and the presence of the

"Notch Parameter" section, which only makes sense in case of equally spaced tones. However, controls in the other control groups are valid and operative for both operating modes. Equally spaced tones are defined on the basis of their common parameters such as start and stop frequencies, and tone spacing or number of tones or both. Arbitrarily distributed tones are defined through a table. In order to simplify the creation of complex scenarios, the tones defined in the equally spaced mode are loaded into the tone table every time the user switches to the arbitrary mode and the tone table is empty. In this way, any number of tones may be easily defined in the equally spaced mode, and then the resulting table may be edited for frequency, amplitude, or phase for each individual tone. Tones may also be deleted or added.

This tab has the following controls:

Waveform Destination Section

- Generate I/Q Data
If checked, baseband (I/Q) signals will be generated. The resulting complex signal will be a series of tones located at positive and/or negative frequencies. As a consequence, negative values can be typed into any waveform frequency edition field in this panel when this checkbox is checked.
- I/Q Toggle buttons
I/Q selection toggle buttons for each channel will be shown when the Generate I/Q Data checkbox is checked. In-Phase (I) and Quadrature (Q) components can be independently assigned to each channel.
- Channel Independent checkboxes allow the definition of Multi-Tone waveforms for Channel 1, Channel 2, Channel 3 or Channel 4. One of the boxes is always checked. When pressing the 'Send To Instrument' button, the waveform is sent to all channels that are checked.
- Segment Number
Target segment for each channel can be defined independently. This field is configurable only for channels sourced from 'extended' memory. The segment range is 1 to 16777216. For channels sourced from 'Internal' memory, the segment is always set to 1, and it displays the text 'Internal'.

Corrections Section

- File...
Open a correction file selection dialog box. Default file extensions match the File Format selection. The name of the successfully loaded correction factors file is shown in the field located at the left of this button. The accepted format for correction files may be found in the [Correction File Format](#) section.
- Channel Specific Frequency and Phase Response
This checkbox activates the application of corrections based on frequency-domain calibration data stored in the target instrument in an internal non-volatile memory. It improves flatness and linear phase distortion.
- Standard Cable
This checkbox activates the application of correction factors based on a typical high-quality, high-bandwidth 0.85m microwave cable (Huber+Suhner type M8041-61616).

Additional Waveform Parameters Section

- **Waveform Length**
It is indicator only. The length is in samples of the resulting segment.
- **Max. Wfm. Length**
Maximum waveform length must be used to force the resulting waveform to be shorter than or equal to the limit set by the user.
- **Keep Sample Rate**
This check box preserves the sampling rate to a user-defined value irrespective of the manner in which other signal parameters may be defined. Keeping the sampling rate to a fixed value may be necessary when multiple waveforms are created for usage in a sequence or scenario. The “Set WL to Max” checkbox shows up when this check box is checked.
- **Set WL to Max**
This checkbox is only available when “Keep Sample Rate” is selected. When this option is selected, the waveform calculation algorithm always takes the maximum waveform length as defined in the “Max. Wfm. Length”. As the waveform length must always be identical for all four channels, it is recommended to check the “Set WL to Max” box in case different waveforms shall be downloaded to different channels.
- **Sample Rate**
Final DAC conversion rate for the resulting signal. It may be set by the user or automatically calculated depending on other signal parameters.

Scaling Section

- **DAC Max**
Multi-Tone waveforms may occupy a limited range of the DAC’s full scale. This parameter sets the maximum level. If set to a lower level than DAC Min, this will be automatically set to the same level. Acceptable range for this parameter is -1/+1, being the full dynamic range of the instrument’s DAC.
- **DAC Min**
Multi-Tone waveforms may occupy a limited range of the DAC’s full scale. This parameter sets the minimum level. If set to a higher level than DAC Max, this will be automatically set to the same level. Acceptable range for this parameter is -1/+1, being the full dynamic range of the instrument’s DAC.

Marker Mode

These controls are available when the “Single Channel with Marker” or “Dual Channel with Marker” mode is selected in the Output tab.

- **Ch. 3 (Marker 1)**
Marker 1 is output on Channel 3. Signaling the beginning of each segment may be activated (Segment selection) and deactivated (None selection).
- **Ch. 4 (Marker 2)**
Marker 2 is output on Channel 4. Signaling the beginning of each segment may be activated (Segment selection) and deactivated (None selection).

Crest Factor Section

- It is an indicator only.
It shows the estimated PAPR for the current waveform in dB. Although the definition of the PAPR parameter is always the ratio between the peak and the average power for a signal, results change depending on the working mode. For the I/Q Data Generation mode, the result reflects the PAPR of the envelope of the resulting signal while for direct generation it reflects the overall signal. The difference between the former and the latter values is close to +3dBs in most cases.

Preview Section

- Multi-Tone Preview Toolbar
The waveform preview toolbar includes the icons that provide different functionality to preview the waveform. For details, see [Preview Section](#)
[Waveform Preview Toolbar](#).

Compilation and Panel Control Section

- Save To File...
Signals can be stored in files either in BIN (for non IQ modes) or IQBIN (for IQ modes) formats. These files may be reused within the Import Waveform tab.
- Send To Instrument
Signal will be transferred to the selected segments of the selected channels. The previous running status for the target instrument will be preserved but sampling rate may be modified depending on the waveform requirements.
- Set Default
All the Multi-Tone waveform parameters are set automatically to their corresponding default values. Entries in the Arbitrary Tone table are not modified by this button.

Two control sections show-up for equally spaced tone definition (“Equispaced” selected in the Tone Distribution drop-down list): “Basic Multi-Tone Waveform Parameters” and “Notch Parameters”.

Basic Multi-Tone Waveform Parameters Section

- Start Frequency
It is the frequency of the first tone. If it is set to a value higher than the one in the Stop Frequency field, this is changed back to the previous Start Frequency.
- Stop Frequency
It is the frequency of the last tone. If it is set to a value lower than the one in the Stop Frequency field, this is changed back to the previous Stop Frequency.
- Spacing
It is an indicator only.
Spacing = (Stop Frequency – Start Frequency)/(# of Tones – 1).
of Tones
It is the total number of tones in the Multi-Tone signal including the ones in the notch, if any.

- Phase Distribution
Phase for each tone can be set in the three different modes: constant, random, and parabolic. While constant phase Multi-Tone signals show a high crest factor, a random phase distribution results in a much lower value for this parameter while a parabolic distribution results in a close to optimal (or minimum) crest factor.
- Seed
This parameter is associated to the random phase distribution and allows generating the same or different random sequences for the phases of each tone. It is also useful to look for a distribution resulting in a desired crest factor value.

Notch Parameters Section

- Notch Active
This check box activates or deactivates the generation of a notch in the equally spaced Multi-Tone signal.
- Start Tone
It is the index of the first tone to be removed in a notch. Acceptable indexes start with 1.
- Stop Tone
It is the index of the last tone to be removed in a notch. Acceptable indexes start with 1.
- Center Frequency
It is an indicator only. The central frequency for the notch is computed and shown in this field.
- Span
It is an indicator only. The tone-free frequency span for the notch is computed and shown in this field.

Arbitrary Tones Section

Alternatively, an edition table shows-up for arbitrarily spaced tones definition (“Arbitrary” selected in the Tone Distribution drop-down list). When not previously edited (or empty), the table is automatically loaded with the parameters of the tones defined in the equally spaced tone section. This allows for easy edition of individual tones or the creation of multiple notches, or both. Parameters for each tone include its frequency (in Hz), its relative amplitude (in dB), and phase (in degrees). Entries in the table may be added, edited, and deleted. Entries in the table may be also sorted in ascending or descending order of any parameter by clicking in the corresponding field name.

Addition of a new entry in the table must be done by editing the empty edition field located at the bottom of the table. Deletion of any number of entries can be performed by selecting the ones to be deleted and then hitting the `` key of the keyboard. Meaningful numeric values must be typed into the edition fields. Otherwise an error condition is triggered. While a valid frequency entry must be always entered, any of the amplitude and phase edition fields may kept empty so they take the default values (0.0 dB for Amplitude and 0.0 degrees for Phase).

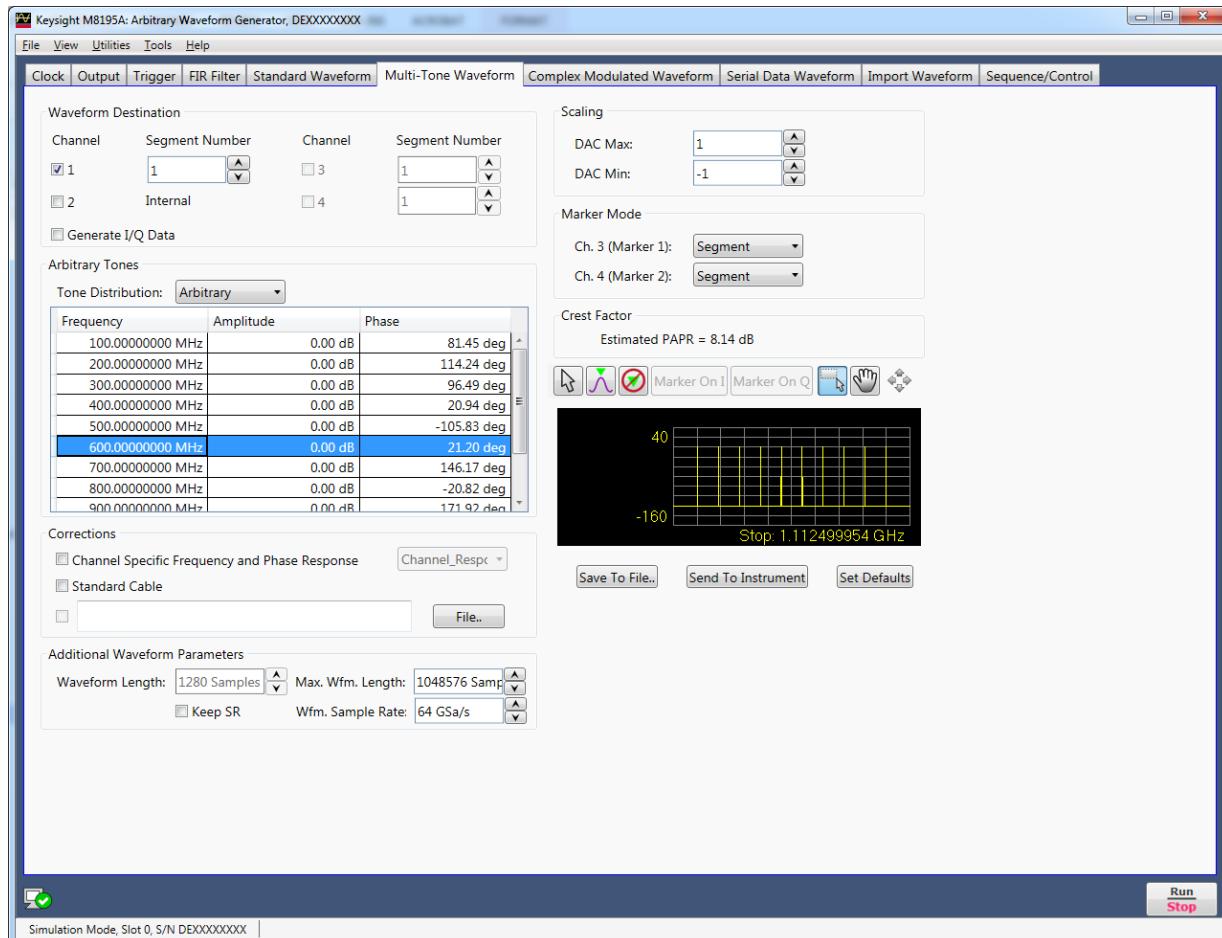


Figure 19: Multi-Tone waveform tab, arbitrary tone distribution

2.12 Complex Modulated Waveform Tab

Use this tab to create baseband and IF/RF digitally modulated signals. User-defined corrections may be applied to signals to compensate for (or emulate) instrument, interconnections and channel linear distortions. The complex modulation tab allows you to generate both RF and Baseband (I/Q) signals. It directly supports a large variety of signal-carrier modulation schemes. This is a list of the currently supported standards, modulation orders, and modulation parameters:

- ASK (Amplitude Shift Keying): Modulation Index (0%-100%).
- PSK (Phase Shift Keying): BPSK, QPSK, $\pi/4$ -DQPSK, Offset-QPSK (OQPSK), 8PSK, and $3\pi/8$ 8PSK (EDGE).
- QAM (Quadrature Amplitude Modulation): 8QAM, 16QAM, 32QAM, 64QAM, 128QAM, 256QAM, 512QAM, AND 1024QAM.
- MSK (Minimum Shift Keying)
- APSK (Amplitude-Phase Shift Keying): 16APSK and 32 APSK. R2/R1 and R3/R1 can be set by the user to any desired value.
- STAR: STAR16 and STAR32. The R2/R1 parameter may be set for the STAR16 modulation scheme.
- VSB (Vestigial Side Band): 8VSB and 16VSB.
- FSK (Frequency Shift Keying): 2FSK, 4FSK, 8FSK, and 16FSK. Peak deviation frequency may be set by the user to any desired value.
- Custom: Users may define arbitrary constellations through simple ASCII files that may be read by the SFP application. Modulations with offset (Q delayed by half a symbol time) and rotating constellations may be also defined.

Pulse Shaping type, characteristics, and different data options may be selected by the user. The panel provides a constellation preview functionality, which can be used to validate the selected modulation scheme and the corresponding modulation parameters. The application takes care of handling the requirements and limits of the target hardware with respect to maximum and minimum record lengths, sampling rate, and record length granularity. As a result, generation of the signals designed in this tab will always be feasible by the instrument and free of distortions such as wrap-around or timing artifacts at any signal domain (time, frequency, and modulation), even if the signal is generated in looped mode.

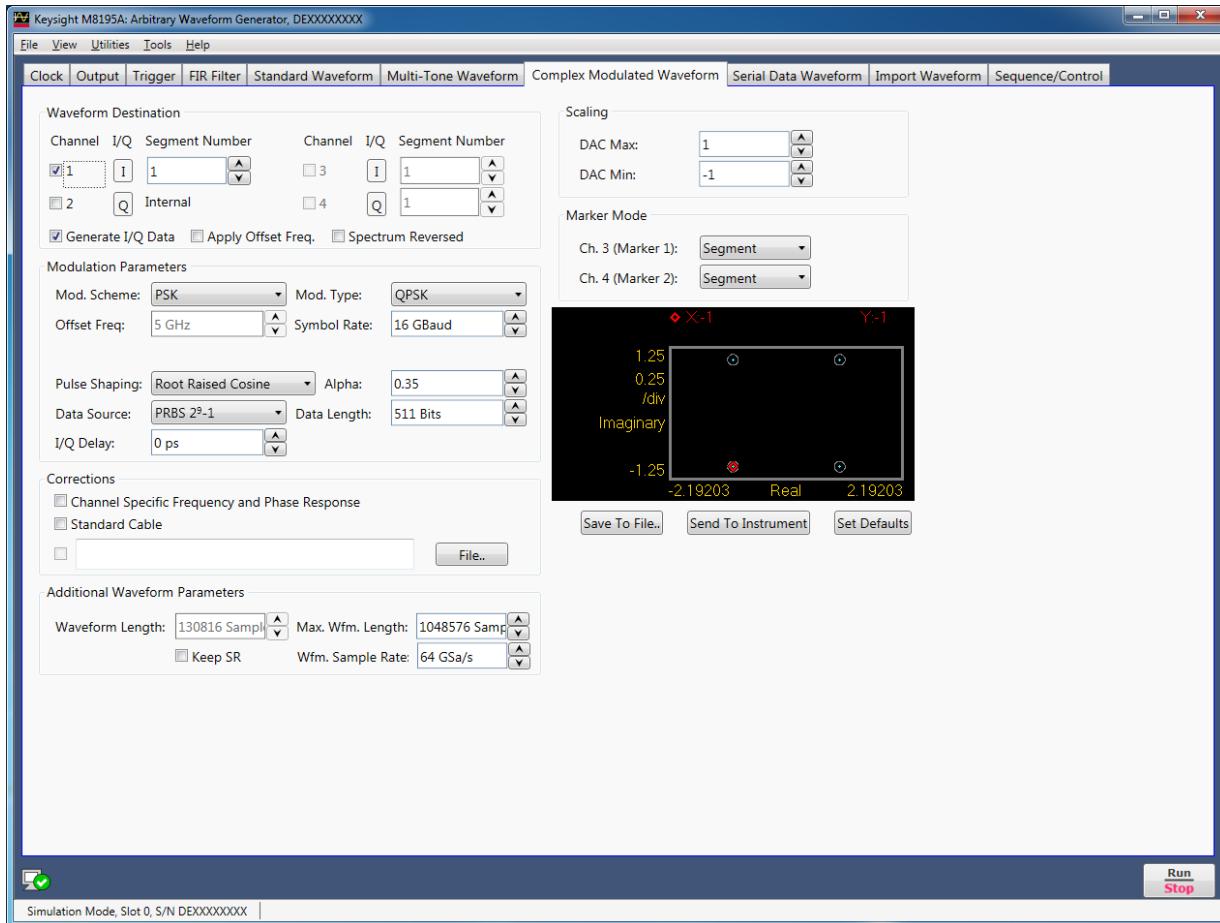


Figure 20: Complex modulated waveform tab

Only relevant parameters and edition fields are shown in the GUI at any time depending on the selected generation mode (RF or I/Q) and modulation scheme.

Waveform Destination Section

- Generate I/Q Data
If checked baseband (I/Q) signals will be generated.
- I/Q Toggle buttons
I/Q selection toggle buttons for each channel will be shown when the Generate I/Q Data checkbox is checked. In-Phase (I) and Quadrature (Q) components can be independently assigned to each channel.
- Apply Offset Freq.
This checkbox is only active for the I/Q Data Generation mode and it applies a frequency shift to the signal according to the 'Offset Freq.' edition field. Frequency shift, unlike carrier frequency, may be positive or negative.
- Spectrum Reversed
This checkbox must be selected for generation of signals in the second Nyquist band (FS/2 – FS). Its effect is the reversion of the fundamental signal (in the 1st Nyquist Band) in the frequency domain. It also reverses the effect of any correction so correction factors obtained for the second Nyquist band will be applied appropriately.

- Channel
Independent checkboxes allow the definition of waveforms for Channel 1, Channel 2, Channel 3 or Channel 4.. One of the boxes will be always checked. When pressing the 'Send To Instrument' button, the waveform is sent to all channels that are checked.
- Segment Number
Target segment for each channel can be defined independently. This field is configurable only for channels sourced from 'extended' memory. The segment range is 1 to 16777216. For channels sourced from 'Internal' memory, the segment is always set to 1, and it displays the text 'Internal'.

Modulation Parameters Section

- Mod. Scheme
This drop-down list selects the different modulation scheme categories that are supported (see list above).
- Mod. Type/Mod. Order
This drop-down list selects the different modulation orders or modulation scheme sub-types for the selected modulation scheme category.
- Carrier Freq. / Offset Freq.
The purpose and labeling of this edition field changes depending on the generation mode. For the direct RF generation mode, it handles the carrier frequency while for the I/Q Data Generation mode it deals with the offset frequency (see the [Apply Offset Freq.](#) control). Units in both cases are in Hz.
- Symbol Rate
This edition field must be used to enter the signaling speed (or baud rate) for the modulated signal expressed in Bauds (1 Baud = 1 Symbol/s).
- Mod. Index(%)
This edition field only shows up when the ASK modulation scheme is selected. It sets the modulation index as a percentage for the signal.
- R2/R1 Ratio
This edition field only shows up when the 16APSK, 32APSK, and 16STAR modulation schemes are selected. It sets the ratio between the radius of the two inner symbol rings in the constellation.
- R3/R1 Ratio
This edition field only shows up when the 32APSK modulation scheme is selected. It sets the ratio between the radius of the outer and the most internal symbol rings in the constellation.
- Freq. Dev.
This edition field only shows up when the FSK modulation schemes are selected. It sets the peak frequency deviation in Hz.
- Mod. File..
This button only shows up when 'Custom' modulation scheme is selected. It opens a file selection window where modulation definition files may be selected. If a valid file is selected, its name will show up in the text field located at the left of this button. Otherwise, a "File Loading Error" message is shown.
- Pulse Shaping
This drop-down list can select different pulse shaping to be applied to the baseband symbols; choices are 'Root Raised Cosine', 'Raised Cosine', 'Gaussian', 'Rectangular', 'None', 'EDGE', and 'Half Sine'.
Notes:
 - The default pulse shape is 'Gaussian'.

- The filter types ‘None’ and ‘Rectangular’ define the pulse shape in time domain. These filter types can only be applied for integer oversampling. Examples: Filter type ‘None’ with 4 times oversampling generates one sample with the actual value followed by 3 samples with a value of zero (Dirac-Pulse). The filter type ‘Rectangular’ with 4 times oversampling generates 4 identical sample values.
- The filter types ‘Root Raised Cosine’, ‘Raised Cosine’, ‘Gaussian’, ‘EDGE’, and ‘Half Sine’ describe the filter shape in frequency domain.
- Alpha / BT
The meaning and labeling of this edition field depends on the selected pulse shaping. For “Nyquist” filters (Raised Cosine and Square Root of Raised Cosine) it is the ‘Alpha’ parameter (or roll-off factor) of the filter. For Gaussian filters it is the BT (Bandwidth/symbol period product) parameter. Some filter types do not require an additional filter parameter.
- Data Source
This drop-down list allows the selection of different pseudo random binary sequences as data sources for modulation. Choices are PRBS7 (Polynomial x^7+x^6+1), PRBS10 (Polynomial $x^{10}+x^7+1$), PRBS11 (Polynomial $x^{11}+x^9+1$), PRBS15 (Polynomial $x^{15}+x^{14}+1$), PRBS23 (Polynomial $x^{23}+x^{18}+1$), PRBS23p (Polynomial $x^{23}+x^{21}+x^{18}+x^{15}+x^7+x^2+1$), and PRB31 (Polynomial $x^{31}+x^{28}+1$).
- Data Length
This edition field may be used to set a given data length to be implemented by the modulated signal. This field defaults to the maximum non-repeating length of the selected PRBS. It also defaults to this value if the user types ‘0’ (Zero). Otherwise, the sequence will be truncated when the number of bits set by this control is reached. If this number is longer than the PRBS maximum length, the sequence will be re-started as many times as necessary.
- I/Q Delay
This numeric edition field allows for the definition of the time skew between the I and the Q baseband components. It can be used to compensate or emulate timing misalignments caused by cabling, external modulators and other devices. This control is activated only when the Generate I/Q Data checkbox is selected. Delay is applied differentially to both components.
- Gray Coding
This checkbox enables gray coding for the applicable modulation modes.

Corrections Section

- File...
Opens a correction file selection dialog box. Default file extension is CSV (Comma-Separated Values). The name of the successfully loaded correction factors file is shown in the field located at the left of this button. The accepted format for correction files may be found in the [Correction File Format](#) section.
- Channel Specific Frequency and Phase Response
This checkbox activates the application of corrections based on frequency-domain calibration data stored in the target instrument in non-volatile memory. It improves flatness and linear phase distortions.
- Standard Cable
This checkbox activates the application of correction factors based on a typical high-quality, high-bandwidth 0.85m cable (Huber+Suhner type M8041 - 61616).

Additional Waveform Parameters Section

- **Waveform Length**
It is an indicator only. The length is in samples of the resulting segment.
- **Max. Length**
Maximum waveform length must be used to force the resulting waveform to be shorter or equal to a limit set by the user.
- **Keep Sample Rate**
This check box preserves the sampling rate to a user-defined value irrespective of any other defined signal parameter. Keeping the sampling rate to a fixed value may be necessary when multiple waveforms are created for usage in a sequence or scenario. The “Set WL to Max” check box gets activated when this check box is checked
- **Set WL to Max**
This checkbox is only available when “Keep Sample Rate” is selected. When this option is selected, the waveform calculation algorithm always takes the maximum waveform length as defined in the “Max. Wfm. Length”. As the waveform length must always be identical for all four channels, it is recommended to check the “Set WL to Max” box in case different waveforms shall be downloaded to different channels.
- **Sample Rate**
It is the final DAC conversion rate for the resulting signal. It may be set by the user or automatically calculated depending on other signal parameters.

Marker Mode

These controls are available when the “Single Channel with Marker” or “Dual Channel with Marker” mode is selected in the Output tab.

- **Ch. 3 (Marker 1)**
Marker 1 is output on Channel 3. Signaling the beginning of each segment may be activated (Segment selection) and deactivated (None selection).
- **Ch. 4 (Marker 2)**
Marker 2 is output on Channel 4. Signaling the beginning of each segment may be activated (Segment selection) and deactivated (None selection).

Scaling Section

- **DAC Max**
Standard waveforms may occupy a limited range of the DAC’s full scale. This parameter sets the maximum level. If set to a lower level than DAC Min, this will be automatically set to the same level. Acceptable range for this parameter is -1/+1, being the full dynamic range of the instrument’s DAC.
- **DAC Min**
Standard waveforms may occupy a limited range of the DAC’s full scale. This parameter sets the minimum level. If set to a higher level than DAC Max, this will be automatically set to the same level. Acceptable range for this parameter is -1/+1, being the full dynamic range of the instrument’s DAC.

Constellation Diagram Section

The constellation diagram section shows a graphic representation of the ideal constellation corresponding to the selected modulation scheme and modulation parameters. It also shows the location of symbols from valid modulation definition files for validation. The line above the constellation diagram shows the following modulation parameters:

- BPS (Bits Per Symbol)
- Per symbol rotation angle (in degrees)
- I/Q delay (in symbol times)

Compilation and Panel Control Section

- Save To File...
Signals can be stored in files in whether BIN (for non IQ modes) or IQBIN (for IQ modes) formats. These files may be reused within the Import Waveform tab.
Send To Instrument
Signal will be transferred to the selected segments of the selected channels. The previous running status for the target instrument will be preserved but sampling rate may be modified depending on the waveform requirements.
- Set Default
All the waveform parameters are set automatically to their corresponding default values.
- Abort
This button allows canceling signal calculation at any moment. It only shows up during signal compilation.

Custom Modulation File Format A custom modulation file is an ASCII delimited file including all the information required to define a single carrier modulated signal based in quadrature (IQ) modulation. The file must be composed of a header including a series of lines with identifiers and parameters, and a list of numerical correction factors. For lines including more than one item (i.e. one identifier and one parameter), those must be separated using commas. Identifiers and parameters are not case sensitive. These are the significant fields for the header:

- #N: This is a mandatory field and it must be the first in the file. The N parameter is the bits per symbol parameter. $0 < N < 11$.
- Offset: It indicates if the Q component must be delayed by half a symbol time respect to the I component. Accepted parameters are 'yes' or 'no'. This parameter is optional. It defaults to 'no' if not included in the file.
- Rotation: It sets the rotation of the constellation for each consecutive symbol in degrees. This parameter is optional. It defaults to 0.0 if not included in the file.
- RotMode: Rotation mode. Parameter may be 'cont' (continuous) or 'alt' (alternate). This parameter is optional. It defaults to 'cont' if not included in the file.
- Vsb: It indicates that vestigial side band baseband filtering must be applied. Accepted parameters are 'yes' or 'no'. This parameter is optional. It defaults to 'no' if not included in the file.

The order of the above entries is not relevant except for the '#N' field that must be placed first in the file. The symbol location section starts with a line including the 'IQ' characters (not case-sensitive). Entries in this section are made by IQ pairs separated by commas. The number of entries must be at least 2^N although additional entries will be ignored. Data to symbol mapping depends on the order of the symbols in the file so its position expressed in binary format corresponds to the binary code assigned to that symbol. Comments must start with the '//' character sequence and may use a complete line or be located at the end of any valid line (including the first line). Empty lines are also valid.

The following example illustrates a simple example of a 3 bit per symbol QAM8 modulation with a particular constellation.

```
#3 // MyModulationFile
Iq
// Inner symbols
2.0, 0.0
0.0, -2.0
-2.0, 0.0
0.0, 2.0
// Outer symbols
3.0, 3.0
-3.0, 3.0
-3.0, -3.0
3.0, -3.0 // Final symbol
```

The above file does not include any unnecessary line in the header as it defines a non-rotating, non-offset modulation so default values for these fields are used instead. The resulting constellation after loading this file is shown as following:



The following example illustrates another possible use of custom modulation to define a distorted constellation. In this particular case, a Q-QPSK modulation with a quadrature error (non-perpendicular I and Q axis) is defined:

```
#2
Offset, yes
iq
1.05, 1.05
-0.95, 0.95
-1.05, -1.05
0.95, -0.95
```

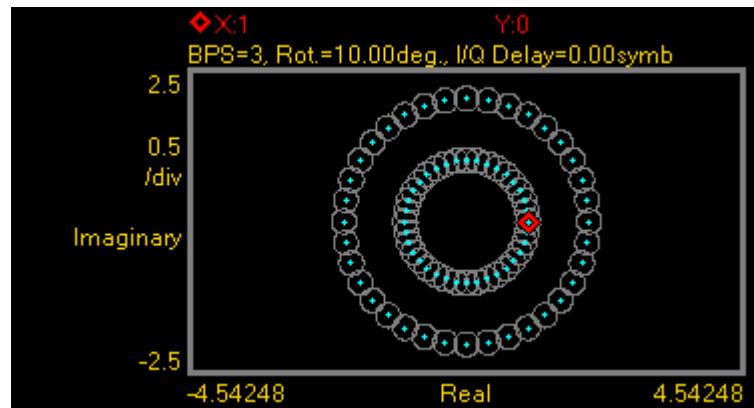
The above file includes a line to indicate that this is an offset modulation. The resulting constellation after loading this file is shown as following:



The following is a more complex example:

```
#3
Offset, no
Rotation, 10.0
RotMODE, cont
iq
1.0, 0.0
2.0, 0.0
0.0, 1.0
0.0, 2.0
-1.0, 0.0
-2.0, 0.0
0.0, -1.0
0.0, -2.0
```

The above file is composed of a header with relevant information. In this particular case, the file contains 8 (2^3) IQ pairs. The 'IQ' characters indicate the starting point for the symbol location list composed by 8 lines with I/Q pairs separated by commas. I and Q will not be delayed ('Offset, no') and constellation will rotate by 10.0 degrees ('Rotation, 10.0') in a continuous fashion ('RotMODE, cont'). In fact, the 'Offset' and 'RotMode' fields could be removed without any effect on the final signal as these fields take the default values. The resulting constellation after loading this file is shown as following:



2.13 Serial Data Waveform Tab

Use this tab to create single lane and multilane bi-level and multi-level high-speed digital serial signals and clocks. User-defined corrections may be applied to signals to compensate for (or emulate) instrument, interconnections and interconnect linear distortions. The serial data tab allows you to generate both data and clock signals. It directly supports a large variety of channel coding and modulation schemes. This is a list of the currently supported modulation and channel coding formats:

- NRZ (Not Return to Zero).
- Unipolar RZ (Return to Zero).
- Polar RZ (Return to Zero).
- PAM-4 (Pulse-Amplitude Modulation, 4 level)
- PAM-5 (Pulse-Amplitude Modulation, 5 level)
- PAM-8 (Pulse-Amplitude Modulation, 8 level)
- PAM-10 (Pulse-Amplitude Modulation, 10 level)
- PAM-12 (Pulse-Amplitude Modulation, 12 level)
- PAM-16 (Pulse-Amplitude Modulation, 16 level)

Users can set the bit/signaling rate, basic pulse shape characteristics, and transition time. Any AWG channel may be selected to generate either a serial signal or a :2 or :4 synchronous clock. A series of standard PRBS sequences with different lengths may be selected in order to produce realistic traffic and to allow bit-error rate testing with standard BER testers. Signals may be corrected for cabling and the AWG frequency response in a channel by channel basis. Additionally, external correction data may be applied to account for the distortions added by additional cabling, passive or active system blocks or test fixturing. Channel to channel skew can be also adjusted with resolutions as low as 100 fs. A variety of Jitter and SSC (Spread Spectrum Clock) profiles can be added to serial data and clock waveforms. Link characteristics can also be emulated. Gaussian noise injection, low-pass filtering and S-parameter-based embedding/de-embedding can be set-up in order to emulate or compensate interconnections and test fixturings. Finally, a 10-taps (5 pre-cursor and five post-cursor) de-emphasis filter is available. An Eye Diagram preview display is shown on the righthand side of the tab. With the help of this Eye diagram display all the physical characteristics of the output waveform and the effects of all the impairments added to it can be easily observed and interactively adjusted. The application takes care of handling the requirements and limits of the target hardware with respect to maximum and minimum record lengths, sampling rate, and record length granularity. As a result, generation of the signals designed in this tab will always be feasible by the instrument and free of distortions such as wrap-around or timing artifacts at any signal domain (time, frequency, and modulation), even if the signal is generated in looped mode.

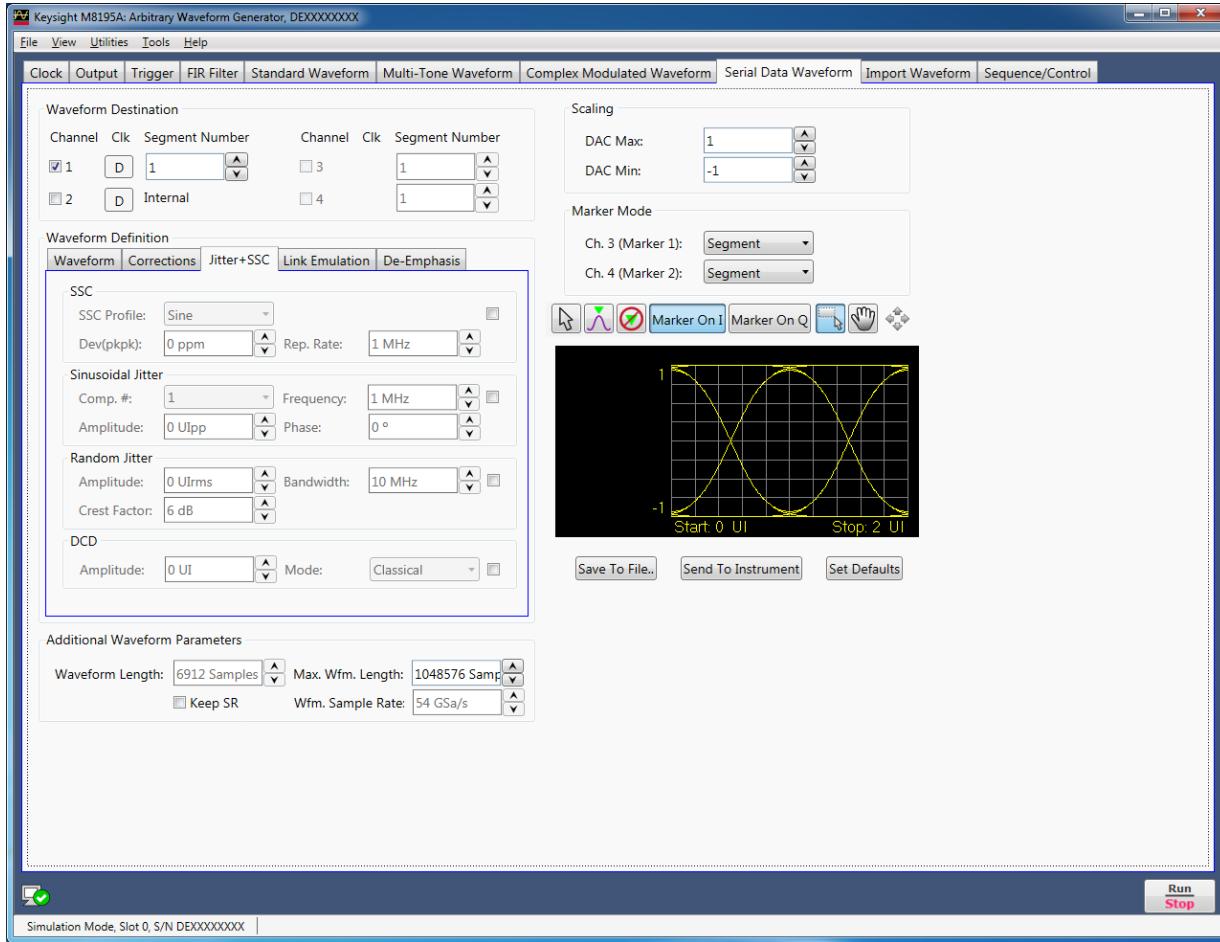


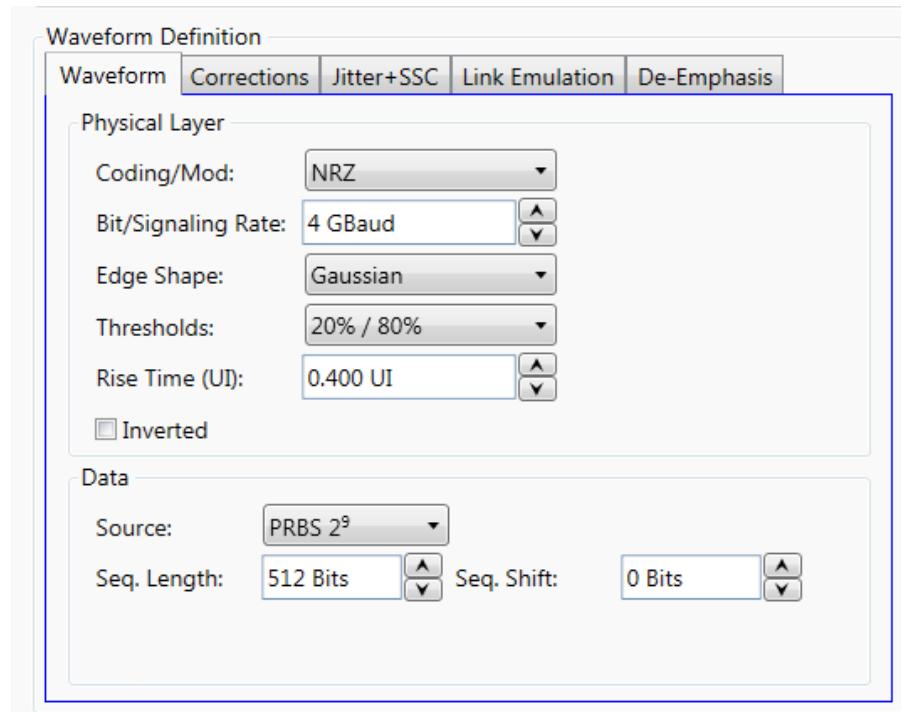
Figure 21: Serial data waveform tab

Only relevant parameters and edition fields are shown in the GUI at any time depending on the selected channel coding scheme.

- **Clock Toggle buttons**
Data/clock selection toggle buttons for each channel. The Data(D), Clock:2 (C/2), and Clock:4 (C/4) can be independently assigned to each channel. The nominal timing for the 50% level in the rising edge for the clock signals is located in the center of the eye for the current symbol.
- **Channel**
Independent checkboxes allow the definition of waveforms for Channel 1, Channel 2, Channel 3, or Channel 4. One of the boxes will be always checked. When pressing the 'Send To Instrument' button, the corresponding waveforms are sent to all channels that are checked.
- **Segment Number**
Target segment for each channel can be defined independently. This field is configurable only for channels sourced from 'extended' memory. The segment range is 1 to 16777216. For channels sourced from 'Internal' memory, the segment is always set to 1, and it displays the text 'Internal'.

Waveform Definition Section

The Waveform Definition section is organized in several tabs where controls are grouped by their functionality: Waveform, Corrections, Jitter&SSC, Link Emulation, De-Emphasis.



Waveform tab:

Physical Layer Section: Physical characteristics of the waveform can be set up in this section. These include the following controls:

- Coding/Mod.
This drop-down list selects the different channel coding and modulation schemes that are supported (see list above). NRZ is the default selection.
- Bit/Signaling Rate
This edition field must be used to enter the signaling speed (or baud rate) for the modulated signal expressed in Bauds (1 Baud = 1 Symbol/s). Baud rate is equal to the bit rate for two-level line coding schemes. 4GBaud is the default value.
- Edge Shape
This drop-down list allows the selection of shape for the transitions (edges); choices are 'Rectangular', 'Trapezoidal' (linear), First Order' (RC network), 'Gaussian', 'Bessel Thompson' (4th order Bessel-Thomson reference receiver filter), 'Raised Cosine' and 'Root Raised Cosine' (Square Root Raised Cosine).
Notes:
 - The default edge shape is 'Gaussian'.
 - For clock signals (i.e. the Clock Toggle button is set to 'C/2' or 'C/4') the edge shape is always Gaussian.

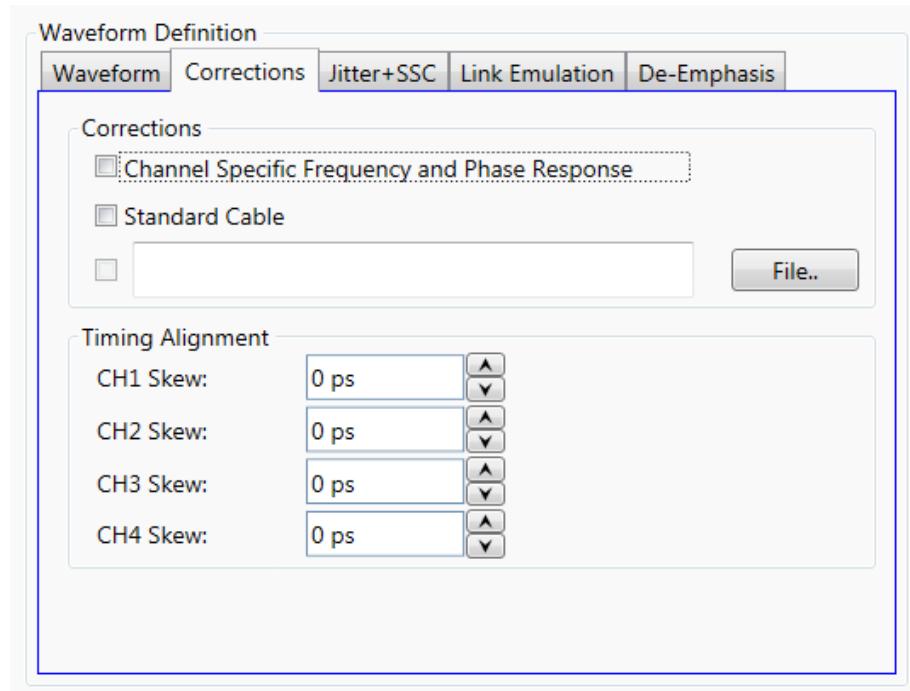
- Thresholds
This drop-down list sets the level threshold convention for the measure rise/fall time parameters. Choices are '20%/80%' and '10%/90%'. '20%/80%' is the default selection for this control.
- Rise Time (UI)
Rise/fall times can be set-up through this edition field. Time must be expressed in UIs (Unit Interval) as a fraction of the symbol duration. Rise time can be set up for all the edges shapes except for the Raised-Cosine and Square Root of Raised-Cosine shapes. Rise time is fixed for clock signals to two sample periods in order to minimize clock jitter. 400mUI (0.4 UI) is the default value for this field.
- Alpha
This edition field only shows up when the Raised Cosine and Square-root of Raised Cosine edge shapes are selected. With it, the excess bandwidth parameter (alpha) of the isolated pulses can be set up. Alpha = 1.0 is the default value.
- Inverted
This checkbox (if checked) reverses the polarity of the output waveform. Default state is unchecked.

Data Section: The sequence of data to be generated can be set up in this section. To do so, the following controls are available:

- Source
This drop-down list allows the selection of different pseudo random binary sequences as data sources for signal generation. Choices are PRBS 2^7 -1 (Polynomial x^7+x^6+1), PRBS 2^9 -1 (Polynomial x^9+x^5+1), PRBS 2^{10} -1 (Polynomial $x^{10}+x^7+1$), PRBS 2^7 (Polynomial x^7+x^6+1), PRBS 2^9 (Polynomial x^9+x^5+1), PRBS 2^{10} (Polynomial $x^{10}+x^7+1$), PRBS 2^{11} (Polynomial $x^{11}+x^9+1$), and PRBS 2^{15} (Polynomial $x^{15}+x^{14}+1$). The sequences are identified by its non-repeating length. The 2^x sequences add an extra '0' to the longest sequence of consecutive '0' in the corresponding 2^{x-1} sequence.
- Seq. Length
This edition field may be used to set a given data length to be implemented by the modulated signal. This field defaults to the maximum non-repeating length of the selected PRBS. It also defaults to this value if the user types '0' (Zero). Otherwise, the sequence will be truncated when the number of bits set by this control is reached. If this number is longer than the PRBS maximum length, the sequence will be re-started as many times as necessary. The actual number of symbols (and record length) in the waveform memory will depend on the line coding/modulation and record length granularity requirements. The simultaneous generation of a clock signal can also influence the actual sequence length as an integer number of clock cycles must be accommodated to keep its integrity (i.e. ISI distortion free characteristics).
- Seq. Shift
This numeric edition field adds a shift to the PRBS sequence being generated by each channel. In this way, uncorrelated data streams may be generated to simulate multi-lane links (i.e. to test the effects of crosstalk) or to emulate IQ baseband channels to feed electrical or optical coherent quadrature modulators. The shift added to each channel may be calculated (in bits) for each channel using the expression Shift = (Channel Number -1) * (Seq. Shift). Unshifted PRBS sequences always start with the longest run of '1' for that particular sequence.

Corrections Tab:

The purpose of these controls is the correction (de-embedding) of different linear distortions and differential delays added by cabling and fixturing, PCB interconnections, etc.



The following controls are included:

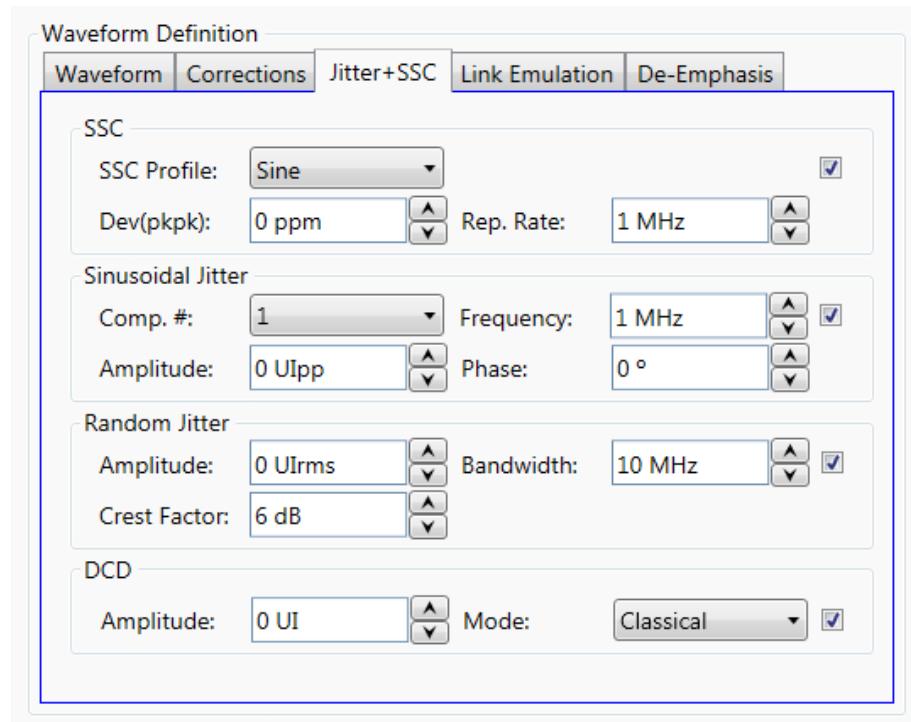
- Channel Specific Frequency and Phase Response
This checkbox activates the application of corrections based on frequency-domain calibration data stored in the target instrument in non-volatile memory. It improves flatness and linear phase distortions.
- Standard Cable
This checkbox activates the application of correction factors based on a typical high-quality, high-bandwidth 0.85m cable (Huber+Suhner type M8041-61616).
- File...
Opens a correction file selection dialog box. Default file extension is CSV (Comma-Separated Values). The name of the successfully loaded correction factors file is shown in the field located at the left of this button. The accepted format for correction files may be found in the [Correction File Format](#) section. In particular, adaptive equalizer models obtained through the Keysight 89600 VSA software can be imported through this procedure to compensate for linear distortions added by any intermediate component, PCB trace, or cable. To obtain this model, apply a NRZ signal with sufficient bandwidth to an 89600 equipped oscilloscope and export the resulting equalizer model. Isolated pulse characteristics of the waveform must be known by the 89600 software so it is advisable to calibrate the SUT (System Under Test) using a Raised-Cosine signal with alpha = 1 to maximize the nominal bandwidth for a given bitrate. The 89600 software must be set up to analyze a BPSK signal with the same baud rate and baseband filter characteristics.

- CH1 Skew / CH2 Skew / CH3 Skew / CH4 Skew

These numeric fields can be used to set-up the absolute delay for each channel in seconds. The valid range for them is -100ps ... +100ps. This feature may be used to control the skew of data and clock signals.

Jitter+SSC Tab:

This tab includes different sections to control a variety of the signal timing characteristics, the injection of several jitter profiles and timing impairments. Each timing control section can be enabled by checking the checkbox located at the top right corner of each section. In order to edit the parameters in each section and to activate their effects in the waveform being generated, the corresponding checkbox must be checked.



SSC Section: Spread-Spectrum Clock characteristics are defined in this section. The following controls are available:

- **SSC Profile:** Sine, Square, Triangular, and Sawtooth profiles are available. SSC profiles are always symmetrical with respect to the nominal signaling rate.
- **Dev(pkpk):** The peak-to-peak symbol clock deviation is expressed in ppm (parts per million) of the nominal (average) signaling rate .
- **Rep. Rate:** This is the repetition rate for the active SSC profile and is expressed in Hz.

Sinusoidal Jitter Section: Up to ten components of sinusoidal jitter can be independently set-up in this section. As amplitude, frequency and phase can be individually defined for each component, more complex periodic patterns can be also defined through their Fourier series development. The following controls are available:

- **Comp #:** This combobox allows the selection of any of the ten components (1-10) for edition. The rest of the controls in this section will be referred, then, to the component # visible in this control. For example; for component #1 you

have certain values for Amplitude, Frequency and Phase. For component #2 you have other values for Amplitude, Frequency and Phase. Each component has its individual values for Amplitude, Frequency and Phase.

- Frequency: The sinusoidal component's frequency is in Hertz (Hz).
- Amplitude: The amplitude of the current component is in Unit Intervals (UI) peak-to-peak.
- Phase: The initial phase is in sexagesimal degrees for the current component.

Random Jitter Section: Gaussian random jitter is defined in this section. Although the PDF (probability distribution function) of the jitter profile follows accurately the Gaussian distribution, the corresponding profile is implemented by embedding the timing deviations in the synthesized waveform, so the same jitter profile will be repeated if the waveform is generated continuously by looping the same segment. As a result, the statistical quality of the jitter distribution will improve with longer waveform lengths.

The following controls are available:

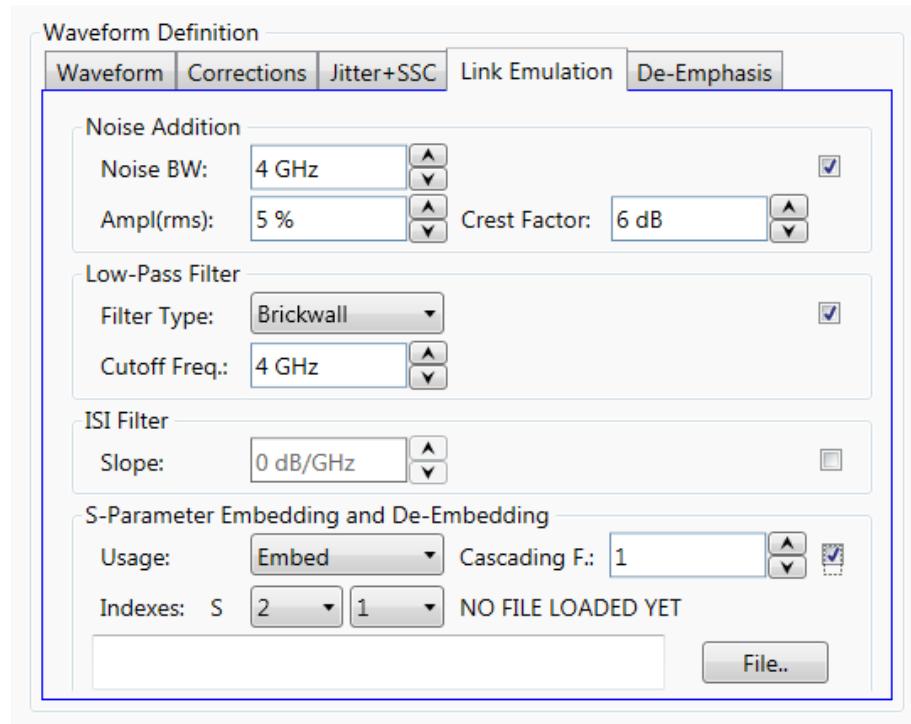
- Bandwidth: The bandwidth of the random jitter profile can be controlled thorough this control. It is expressed in Hertz (Hz).
- Amplitude: This is the rms (root-mean square) amplitude of the random jitter (1 sigma) and is expressed in Unit Intervals (UI).
- Crest Factor: This control allows for the random jitter profile clipping before low-pass filtering. It is expressed in dB as a ratio between the maximum peak and the rms value of the random jitter profile. For example: A crest factor of 0 dB corresponds to a peak-to-peak jitter of 2 times the rms value. A crest factor of 6 dB corresponds to a peak-to-peak jitter of 4 times the rms value. The actual crest factor will be higher than this setting as bandwidth limiting is applied to the random jitter profile after clipping to make sure the desired bandwidth is preserved.

DCD Section: Duty Cycle Distortion (DCD) jitter can be set in this section. There are two types of impairments supported: “Classical” DCD, where there is a timing difference between “marks” and “spaces” in the symbol sequence, and F/2 jitter, where the duration of a symbol flips between two values from one symbol to the next. The following controls are available:

- Amplitude: This is the peak-to-peak amplitude of the DCD jitter expressed in Unit Intervals (UI).
- Mode: This combo box allows for the selection of the “Classical” DCD or F/2 jitter mode.

Link Emulation Tab:

In this tab, linear distortions and noise can be added to the waveforms to emulate the physical effects of interconnections and crosstalk. In addition to standard low-pass filters whose parameters can be set by the user, it is possible to import Touchstone files to embed or de-embed S-parameter responses obtained through frequency-domain or time-domain analysis instruments or directly synthesized by simulation tools. Each section in the tab can be independently enabled by checking the checkbox located at the top right corner of each section. Every corresponding checkbox must be checked to edit the parameters in each section and to activate their effects in the waveform being generated.



Noise Addition Section: Bandwidth-limited Additive White Gaussian Noise (AWGN) can be added to the waveform to emulate a plurality of interfering noise sources. Noise is always added to the waveform before applying any linear distortion found in this tab (so it will behave as a “near-end” noise). Although the PDF (probability distribution function) of the AWGN noise follows accurately the Gaussian distribution, the corresponding profile is implemented by adding the noise to the synthesized waveform, so the same noise will be repeated if the waveform is generated continuously by looping the same segment. As a result, the statistical quality of the jitter distribution will improve with longer waveform lengths. The following controls are available:

- Noise BW: Bandwidth for the Gaussian noise can be set in Hertz (Hz).
- Ampl(rms): Root-mean-square amplitude for noise is set as a percentage of the reference “low” to “high” excursion. Reference “low” and “high” amplitudes correspond to the final, steady level after a long run of consecutive “low” or “high” states. These levels are not influenced by any low-pass or de-emphasis filters applied to the waveform (see [Figure 22](#)), so it is used to establish an absolute reference for relative amplitudes.
- Crest Factor: This parameter is expressed in dB relative to the rms amplitude of the noise. If the combined waveform (signal + noise) goes beyond the valid lower and upper limits, samples are clipped to the corresponding limit. The upper limit can be calculated by adding the Nominal “high” value and an additional headroom resulting from the Ampl(rms) parameter corrected by the Crest Factor parameter. The lower limit can be symmetrically calculated in a similar way from the Nominal “low” level (see [Figure 22](#)).

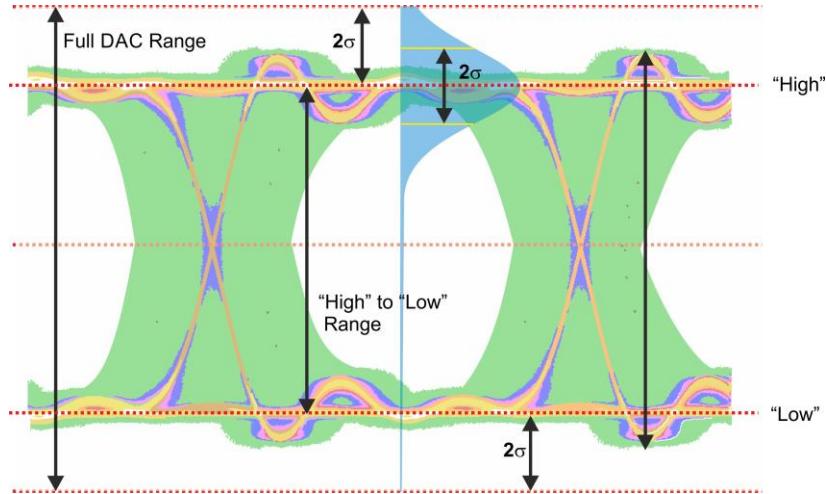


Figure 22: Clipping and noise amplitude

Low-Pass Filter Section: Brickwall, and first and second order low-pass filters can be applied to waveforms, so the effects of discrete components or distributed resistance, capacitance, and inductance in interconnections can be emulated. DC Gain for all filters is always 0dB so the amplitude of the reference “high” and “low” levels remain unchanged (see the Ampl(rms) control description in the Noise Addition Section). The activation of any Low-Pass Filter and the ISI Filter are mutually exclusive. The following controls are available:

- **Filter Type:** This combo box allows for the selection of brickwall, first order and second order low-pass filters.
- **Cutoff Freq.:** This is a context sensitive control and it is available only for Brickwall and First-Order filters (where it refers to the 3dB-attenuation frequency).
- **Ress. Freq.:** This is a context sensitive control and it is available only for Second-Order filters. The resonance frequency of the filter can be set through this control.
- **Q:** This is a context sensitive control and it is available only for Second-Order filters. The Quality Factor (or Q Factor) can be set through this control.

ISI Filter Section: The purpose of this filter is the emulation of the effects of some hardware filters used in combination of traditional pattern generators to cause some controlled, traceable level of ISI (Inter Symbolic Interference). Gain at DC for all filters is always 0dB so the amplitude of the reference “high” and “low” levels remain unchanged (see the Ampl(rms) control description in the Noise Addition Section). The activation of any Low-Pass Filter and the ISI Filter are mutually exclusive. The following control is available:

- **Slope:** ISI filter is modeled as a linear attenuation slope, expressed in dB, so it can be fully defined by setting up the filter’s slope and is expressed in dB/GHz units.

S-Parameter Embedding and De-Embedding Section: Embedding (emulating) or de-embedding (compensating) the response of actual components or interconnections can be accomplished by importing S-parameter files in the Touchstone ® v1.1 and v2.0 formats. Files containing information for up to ten ports are supported.

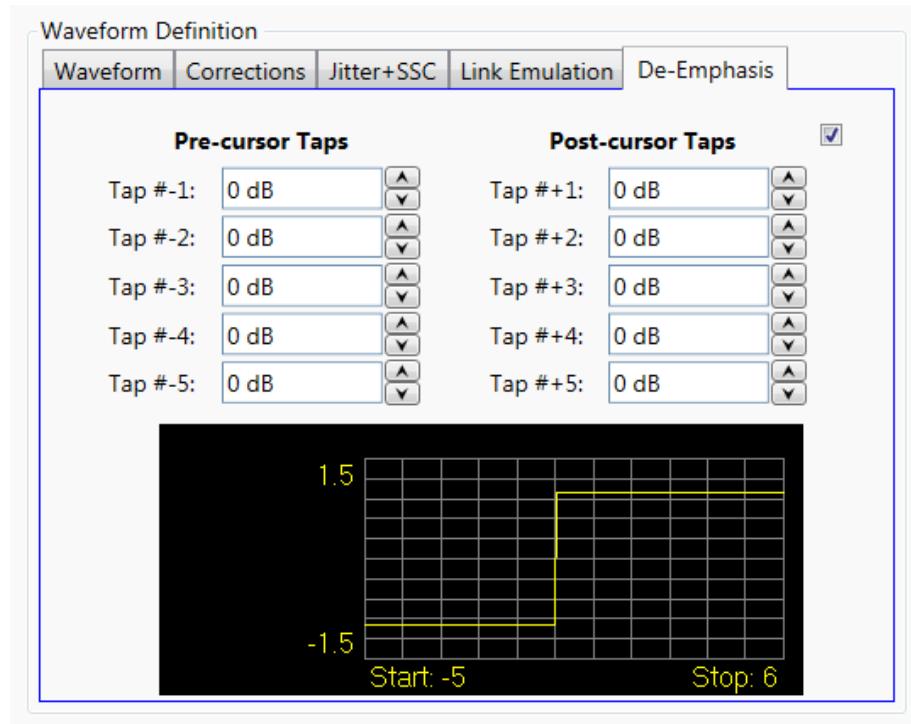
The following controls are available:

- Usage: This combo box can be used to choose between embedding (embed) or de-embedding (deembed) the frequency response data from the imported S-parameter file.
- Cascading F: The cascading factor allows for the emulation of multiple cascaded identical blocks from the description of a single block. For PCB traces or cables, this control can be used to simulate the effects of sections of a different length to the one characterized in the imported file.
- Indexes: These two combo boxes allow for the selection of the right parameter within the S-parameter matrix. It is not possible to select a parameter relative to a single port so the contents of the two combo boxes cannot be identical. If available in the imported file, physical information about the selected ports is shown. For hybrid S-Parameter files, the type of parameter being defined (S for Single-Ended, C for Common Mode, and D for Differential Mode) and the associated physical ports is also listed.
- File: This button opens a file selection dialog box. Default extensions for files is “*.s?p” so most standard v1.1 and v2.0 Touchstone ® files will be automatically shown. If importing the file is successful, the name of the file will be shown in the text field next to this control and some basic information about the file is shown in the line over it. This information includes the number of factors (frequency entries) in the file and the identification information for the physical ports related to the the selected S-parameter.

De-Emphasis Tab:

The De-Emphasis tab can be used to generate complex emphasis filters. It allows you to define up to five post-cursor and 5 pre-cursor taps. An interactive graph located at the bottom of the tab shows the step response corresponding to the defined filter. A checkbox located at the upper right corner of the tap enables/disables the application of the de-emphasis filter to the symbol sequence in the waveform and the edition of the associated controls.

Definitions for tap levels are derived from the N4916B De-emphasis Signal Converter data sheet (see <http://literature.cdn.keysight.com/litweb/pdf/5990-4630EN.pdf>).



The following controls are available:

- Pre-cursor Taps #1/#-5: Precursor taps can be edited from tap #-1 (the closest to the transition) up to tap #5.
- Post-cursor Taps #1/#5: Postcursor taps can be edited from tap #1 (the closest to the transition) up to tap #5.

Additional Waveform Parameters Section

- Waveform Length
It is an indicator only. The length is in samples of the resulting segment.
- Max. Length
Maximum waveform length must be used to force the resulting waveform to be shorter or equal to a limit set by the user.
- Keep Sample Rate
This check box preserves the sampling rate to a user-defined value irrespective of any other defined signal parameter. Keeping the sampling rate to a fixed value may be necessary when multiple waveforms are created for usage in a sequence or scenario. The "Set WL to Max" check box gets activated when this check box is checked
- Set WL to Max
This checkbox is only available when "Keep Sample Rate" is selected. When this option is selected, the waveform calculation algorithm always takes the maximum waveform length as defined in the "Max. Wfm. Length" field. As the waveform length must always be identical for all four channels, it is recommended to check the "Set WL to Max" box in case different waveforms from different SGFP tabs shall be downloaded to different channels. Record length are calculated to contain an integer number of complete PRBS sequences except when the "Set WL to Max" is checked. In this case the number of symbols in the resulting waveform will be the closest integer for the

signaling rate set by the user. As a result, signaling rate will be adjusted, if necessary, so it is consistent with the resulting time window (Time Window = Record Length * Sampling Rate).

- Sample Rate

Indicator only. It is the final DAC conversion rate for the resulting signal. It is automatically calculated depending on other signal parameters if the “Keep Sample Rate” checkbox is not checked.

Marker Mode

These controls are available when the “Single Channel with Marker” or “Dual Channel with Marker” mode is selected in the Output tab.

- Ch. 3 (Marker 1)

Marker 1 is output on Channel 3. Signaling the beginning of each segment may be activated (Segment selection) and deactivated (None selection).

- Ch. 4 (Marker 2)

Marker 2 is output on Channel 4. Signaling the beginning of each segment may be activated (Segment selection) and deactivated (None selection).

Scaling Section

- DAC Max

Standard waveforms may occupy a limited range of the DAC’s full scale. This parameter sets the maximum level. If set to a lower level than DAC Min, this will be automatically set to the same level. Acceptable range for this parameter is -1/+1, being the full dynamic range of the instrument’s DAC.

- DAC Min

Standard waveforms may occupy a limited range of the DAC’s full scale. This parameter sets the minimum level. If set to a higher level than DAC Max, this will be automatically set to the same level. Acceptable range for this parameter is -1/+1, being the full dynamic range of the instrument’s DAC.

Compilation and Panel Control Section

- Save To File

Signals can be stored in files in BIN format. These files may be reused within the Import Waveform tab.

NOTE

The waveform is always saved without applying corrections. Also, the waveform of the data signal (Clock Toggle button is set to ‘D’) and not the clock signal (Clock Toggle button is set to ‘C/2’ or ‘C/4’) is saved.

- Send To Instrument

Signal will be transferred to the selected segments of the selected channels. The previous running status for the target instrument will be preserved but sampling rate may be modified depending on the waveform requirements.

- Set Default

All the waveform parameters are set automatically to their corresponding default values.

- Abort

This button allows canceling signal calculation at any moment. It only shows up during signal compilation.

2.13.1 Bitmapping for Binary Data to PAM Signals

This section describes how the binary data of the data source (e.g. a PRBS) is mapped to the different levels of a PAM-4, PAM-5, PAM-8, PAM-10, PAM-12 or PAM-16 signal.

Definition:

- A PAM-n signal has n levels.
- The level number 1 is associated with the low level.
- The level number n is associated with the high level.

Table 10: PAM4

| Level number | Binary data ('Inverted' not checked) | Binary data ('Inverted' checked) |
|--------------|--------------------------------------|----------------------------------|
| 4 | 11 | 00 |
| 3 | 10 | 01 |
| 2 | 01 | 10 |
| 1 | 00 | 11 |

Table 11: PAM8

| Level number | Binary data ('Inverted' not checked) | Binary data ('Inverted' checked) |
|--------------|--------------------------------------|----------------------------------|
| 8 | 111 | 000 |
| 7 | 110 | 001 |
| 6 | 101 | 010 |
| 5 | 100 | 011 |
| 4 | 011 | 100 |
| 3 | 010 | 101 |
| 2 | 001 | 110 |
| 1 | 000 | 111 |

Table 12: PAM16

| Level number | Binary data ('Inverted' not checked) | Binary data ('Inverted' checked) |
|--------------|--------------------------------------|----------------------------------|
| 16 | 1111 | 0000 |
| 15 | 1110 | 0001 |
| 14 | 1101 | 0010 |
| 13 | 1100 | 0011 |
| 12 | 1011 | 0100 |
| 11 | 1010 | 0101 |
| 10 | 1001 | 0110 |
| 9 | 1000 | 0111 |
| 8 | 0111 | 1000 |
| 7 | 0110 | 1001 |
| 6 | 0101 | 1010 |
| 5 | 0100 | 1011 |
| 4 | 0011 | 1100 |
| 3 | 0010 | 1101 |
| 2 | 0001 | 1110 |
| 1 | 0000 | 1111 |

- **PAM-5:** Two bits of the binary data are used. The same mapping as for the PAM-4 modulation is applied to get the 4 outer levels. The level in the middle is generated randomly with 1/5th probability.
- **PAM-10 (or PAM-12):** 4 bits of the binary data are used. This gives 16 possible levels. However, only 10 (or 12) values are needed. If the value is lower than 10 (or 12), direct mapping is applied. If the value is equal to or greater than 10 (or 12), random mapping is applied to any of the valid 10 (or 12) levels.

2.14 Import Waveform Tab

Use this tab to perform the functions such as importing, scaling, and resampling waveform files in a variety of formats for their generation by the M8195A arbitrary waveform generator. It provides the controls which allow the complete definition of signal processing parameters for the waveform file format (see [File Format](#)).

Depending on the file format and contents, information regarding the original sampling rate of the input waveforms can be extracted and re-used within the import tool.

Resampling is performed so no images or aliases show up in the resampled waveform.

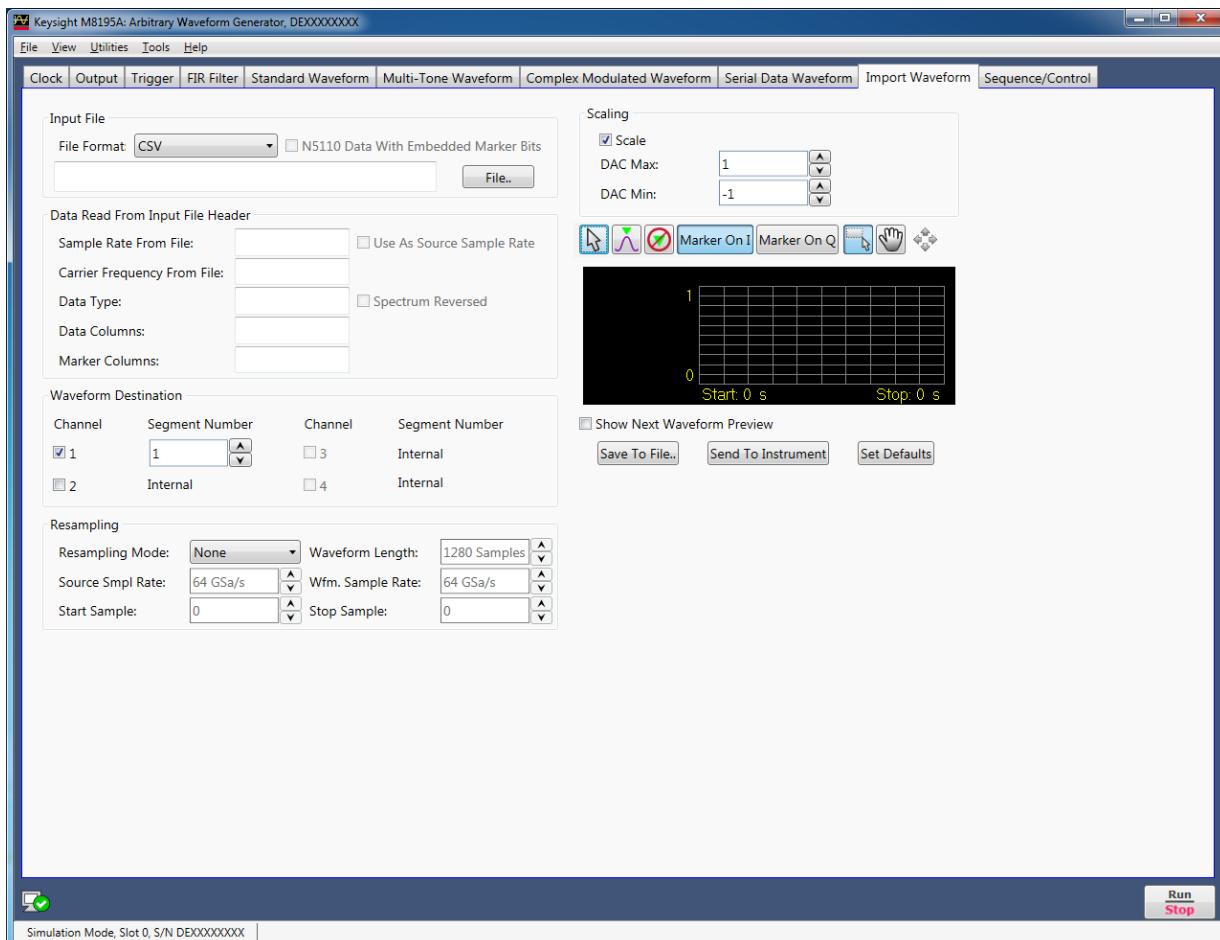


Figure 23: Import waveform tab

This tab has the following controls and indicators:

Input File Section

- File Format

For details on the available file format, see [File Format](#).

Sample waveform data files are available in different formats as listed in the [Table 13](#). The files can be simply imported using the Input File section and can be sent to the instrument to view the waveform preview. The sample waveform data can be found at the location: **Start > All Programs > Keysight M8195 > Keysight M8195 Examples**

Steps to view the sample data file waveform preview:

- Select the Show Next Waveform Preview check box.
- Select the required **File Format** from the drop-down list.
- Click **File...**
- In the **Open** dialog box, select the sample waveform file (as per selected file format)
- Click **Open**.
- Click Send to Instrument.

Table 13: Sample waveform data files

| File format | Waveform data file |
|-------------|--------------------------|
| TXT | Sin10MHzAt64GHz.txt |
| BIN | Sin10MHzAt64GHz.bin |
| BIN8 | Sin10MHzAt64GHz.bin8 |
| BIN6030 | Sin10MHzAt64GHz.bin6030 |
| BIN5110 | SinDelta10MHzIQ.bin5110 |
| IQBIN | SinDelta10MHzIQ.iqbin |
| MAT89600 | Sin10MHzAt64GHz.mat89600 |
| CSV | Sin10MHzAt64GHz.csv |
| DSA90000 | Sin10MHzAt64GHz.dsa90000 |

- N5110 Data With Embedded Marker Bits

This checkbox is only enabled, if the File Format is BIN5110. If checked, the BIN5110 format with 14-bit data for I and Q and embedded marker bits is used. If unchecked, the BIN5110 format with 16-bit data for I and Q and no marker bits is used.

- File...

Open a file selection dialog. Default file extensions match the File Format selection. Successful loading of a waveform updates multiple information fields through the panel reflecting the waveform settings and a graph of the waveform is shown in the preview display.

Data Read From Input File Header Section

- Sample Rate From File
Indicator only. It shows the input waveform sample rate, if any, contained in the loaded file. If no sample rate is specified “n.a.” (not available) is shown.
- Use As Source Sample Rate
This checkbox assigns the sample rate specified in the file as the Source Sample Rate used for resampling.
- Carrier Frequency From File
Indicator only. It shows the input waveform carrier frequency, if any, contained in the loaded file. If no carrier frequency is specified “n.a.” (not available) is shown.
- Data Type
This is the organization of samples within the file. It may be Single (real-only waveform) or IQ (complex waveforms).
- Spectrum Reversed
This checkbox is only active for complex (IQ) waveforms. It results in an imported signal which is the complex conjugate of the input signal, thus its spectrum will be reversed.
- Data Columns
It shows the internal organization of the file regarding waveforms. It can show from one column (Y1) up to 4 (Y1, Y2, Y3, Y4).
- Marker Columns
It shows the internal organization of the file regarding markers. It can show from one column (M1) up to 4 (M1, M2, M3, M4).

Waveform Destination Section

- Channel
Independent checkboxes allow to import waveforms for Channel 1, Channel 2, Channel 3 or Channel 4. One of the boxes is always checked. If the file contains only one waveform, when pressing the ‘Send To Instrument’ button, the waveform is sent to all channels that are checked.
If the file contains multiple waveforms (file types MAT89600 and CSV), they can be sent to multiple channels in one operation.

The following two tables show the standard column-to-channel mapping for the case of no additional data header in the CSV file or no reordering of the column names in the MAT89600 file.

Table 14: Standard column-to-channel mapping in four-channel mode

| Number of columns in file for real values | Import and download to M8195A, when corresponding channel box is checked |
|---|---|
| 1. | Column 1 to Ch 1 and Ch 2 and Ch 3 and Ch4 |
| 2. | Column 1 to Ch 1 and Column 2 to Ch 2 |
| 3. | Column 1 to Ch 1 and Column 2 to Ch 2 and Column 3 to Ch 3 |
| 4. | Column 1 to Ch 1 and Column 2 to Ch 2 and Column 3 to Ch 3 and Column 4 to Ch 4 |

Table 15: Standard Column to channel mapping in two-channel mode

| Number of columns in file for real values | Import and download to M8195A, when corresponding channel box is checked |
|---|--|
| 1. | Column 1 to Ch 1 and Ch4 |
| 2. | Column 1 to Ch 1 and Column 2 to Ch 4 |
| 3. | Column 1 to Ch 1 and Column 2 to Ch 4, Column 3 is ignored |
| 4. | Column 1 to Ch 1 and Column 2 to Ch 4, Column 3 and 4 are ignored |

For MAT89600 file and CSV file with data header, the mapping shown below applies:

Table 16: Modified column-to-channel mapping in four-channel mode

| Name of column | Import and download to M8195A, when corresponding channel box is checked |
|----------------|--|
| Y1 | Ch 1 |
| Y2 | Ch 2 |
| Y3 | Ch 3 |
| Y4 | Ch 4 |

Table 17: Modified column-to-channel mapping in two-channel mode

| Name of column | Import and download to M8195A, when corresponding channel box is checked |
|----------------|--|
| Y1 | Ch 1 |
| Y2 | Ch 4 |
| Y3 | ignored |
| Y4 | Ch 4, if Y2 is not present; ignored, if Y2 is present |

- I/Q Toggle buttons
I/Q selection toggle buttons for each channel will be shown when the file containing an I/Q waveform is selected for import. In-Phase (I) and Quadrature (Q) components can be independently assigned to each channel.
- Segment Number
Target segment for each channel can be defined independently. This field is configurable only for channels sourced from 'extended' memory. The segment range is 1 to 16777216. For channels sourced from 'Internal' memory, the segment is always set to 1, and it displays the text 'Internal'.

Resampling Section

- Resampling Mode
It controls the way waveforms are imported and resampled. Please refer to the description of the [Resampling Methodology](#) in the Appendix chapter. The following modes are available:
 - None: Baseband Sample Rate will be the same as the Source Sampling Rate. The output waveform will use the same number of samples as the selected portion of the input waveform. Granularity requirements will be met by repeating the basic waveform the minimum number of times so the combined length is a multiple of the granularity for the current DAC mode.
 - Timing: The time window of the input signal (Waveform Length / Sample Rate) will be used to calculate the best value for the output record length being a multiple of the granularity for the current DAC mode according to the output sampling rate defined by the user. Final output sampling rate will be slightly adjusted to accurately keep the timing of the original signal.
 - Output_SR: The user-defined output sampling rate will be used to calculate the best value for the output record length being a multiple of the granularity for the current DAC mode according to the time window of the input signal. Final time window will be slightly adjusted to keep the selected output sampling rate. This change is reflected in the Source Sampling Rate numeric entry field value.
 - Output_RL: The user-defined output Waveform Length will be used to calculate the best value for the output Sample Rate according to the time window of the input signal. Waveform Length will be adjusted to the nearest multiple of the granularity for the current DAC mode according to the time window of the input signal.
 - Zero_Padding: Output Waveform Length is calculated based on the input waveform time window and the user-defined output sampling rate. The resulting waveform length will not be, in general, a multiple of the granularity. To meet the granularity conditions, a number of zero samples are added until the combined number of samples is a multiple of the granularity. Output Sample Rate will be slightly adjusted to keep the input waveform time window.
 - Truncate: Output Waveform Length is calculated based on the input waveform time window and the user-defined output sampling rate. The resulting waveform length will not be, in general, a multiple of the granularity. To meet the granularity conditions, a number of samples is removed until the resulting number of samples is a multiple of the

- granularity. Output Sample Rate will be slightly adjusted to keep the input waveform time window.
- Repeat: Output Waveform Length is calculated based on the input waveform time window and the user-defined output sampling rate. The resulting waveform length will not be, in general, a multiple of the granularity. To meet the granularity conditions, the base waveform is repeated the minimum number of times so the overall number of samples is a multiple of the granularity. Output Sample Rate will be slightly adjusted to keep the input waveform time window. The Waveform Length field will show the length of the combined waveform.
 - **Waveform Length**
It shows the number of samples of the resampled output waveform. It can be set when Resampling Mode is Output_RL. Otherwise, this field is an indicator.
 - **Source Sample Rate**
The speed at which samples in the input waveform are sampled. It can be set by typing a valid value unless the "Use As Source Sample Rate" checkbox is checked. In this particular case, the sampling rate information contained in the input waveform file will be always used.
 - **Baseband Sample Rate**
The speed at which samples in the output waveform will be converted. It can be set in all Resampling Modes except for the Output_RL mode.
 - **Start Sample**
This field can be used to select the starting sample of the section of the input waveform to be imported. It cannot be set to a value higher than the Stop Sample.
 - **Stop Sample**
This field can be used to select the final sample of the section of the input waveform to be imported. It cannot be set to a value lower than the Start Sample.
 - **Scale**
This checkbox controls the way the input output waveform will be scaled after resampling. If unchecked, the output waveform samples will not be re-scaled. Sample levels over +1.0 or under -1.0 will be clipped.

Scaling Section

- **DAC Max**
Imported waveforms may occupy a limited range of the DAC's full scale. This parameter sets the maximum level. If set to a lower level than DAC Min, this will be automatically set to the same level. Acceptable range for this parameter is -1/+1, being the full dynamic range of the instrument's DAC.
- **DAC Min**
DAC Max: Imported waveforms may occupy a limited range of the DAC's full scale. This parameter sets the minimum level. If set to a higher level than DAC Max, this will be automatically set to the same level. Acceptable range for this parameter is -1/+1, being the full dynamic range of the instrument's DAC.

Preview Section

- Waveform Preview Toolbar
 - The waveform preview toolbar includes the icons which provide different functionality to preview the waveform. For details, see [Preview Section](#)
- Waveform Preview Toolbar.
- Show Next Waveform Preview
This checkbox affects the behavior of the preview for the next waveform. If selected, a preview of the imported waveform is displayed. Leave this checkbox unselected to speed up the import of large waveforms.
 - Save To File...
Signals can be stored in files in whether BIN (for non IQ modes) or IQBIN (for IQ modes) formats. These files may be reused within the Import Waveform tab.
 - Send To Instrument
Signal will be transferred to the selected segments of the selected channels. The previous running status for the target instrument will be preserved but sampling rate may be modified depending on the waveform requirements.
 - Set Default
All the imported waveform parameters are set automatically to their corresponding default values.

2.15 Sequence/Control Tab

Use this tab to create a sequence with one or more (upto 16M) sequence entries. The characteristics of a sequence depend on the parameters' values of the constituent entries. This tab allows to create, configure, and send new sequence configuration to the instrument, and also to extract the existing one. The sequencing functionality is only available for channels with 'extended' memory, and all the channels share the same sequence information (i.e. the sequence created using this tab will be same for all the channels sourced from 'extended' memory). The option 'SEQ' must be installed for sequencing to work.

You can also configure various sequence/control parameters using this tab.

The following figure shows Sequence/Control tab (STScenario mode):

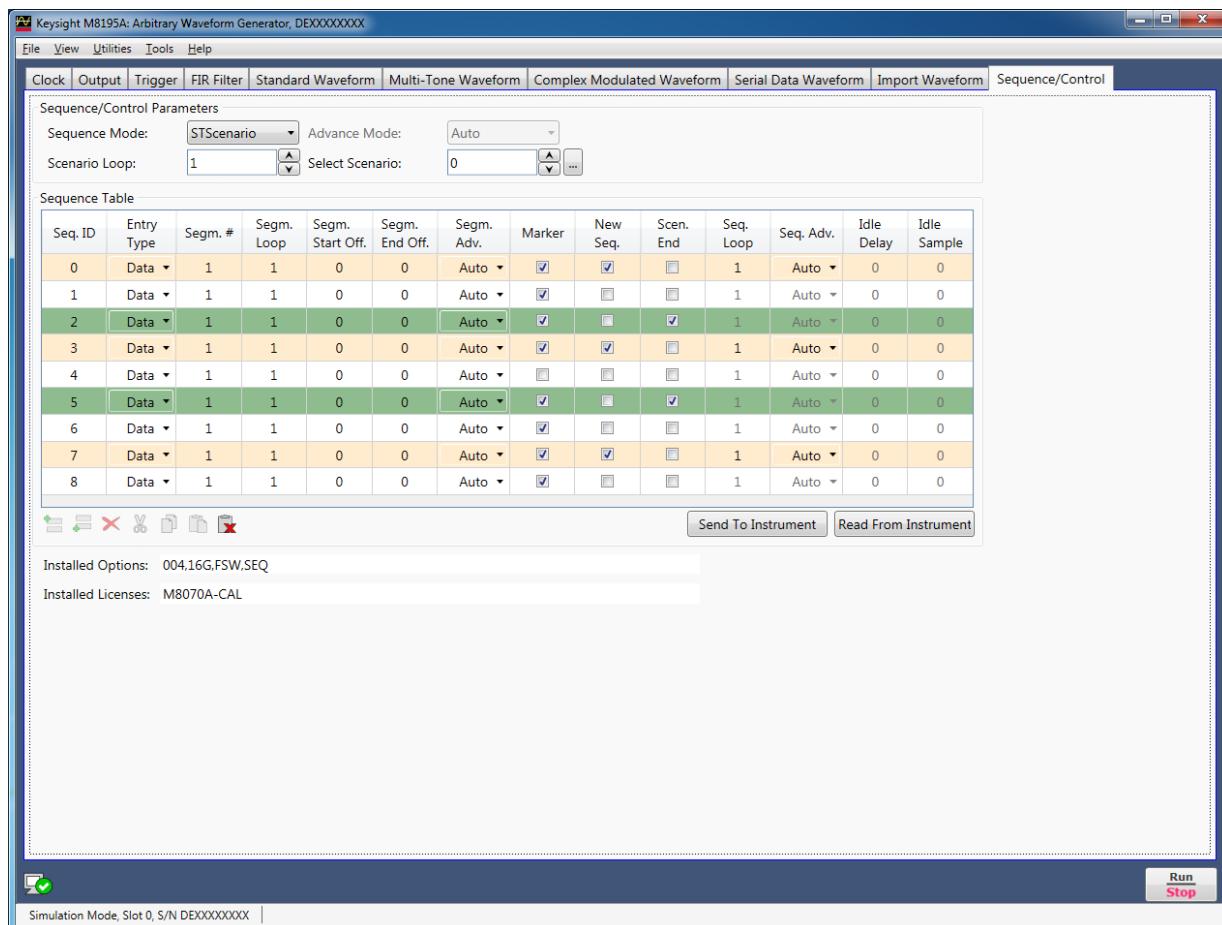


Figure 24: Sequence/Control tab (STScenario mode)

The following figure shows Sequence/Control tab (STSequence Mode with Dynamic Control on and signal generation stopped):

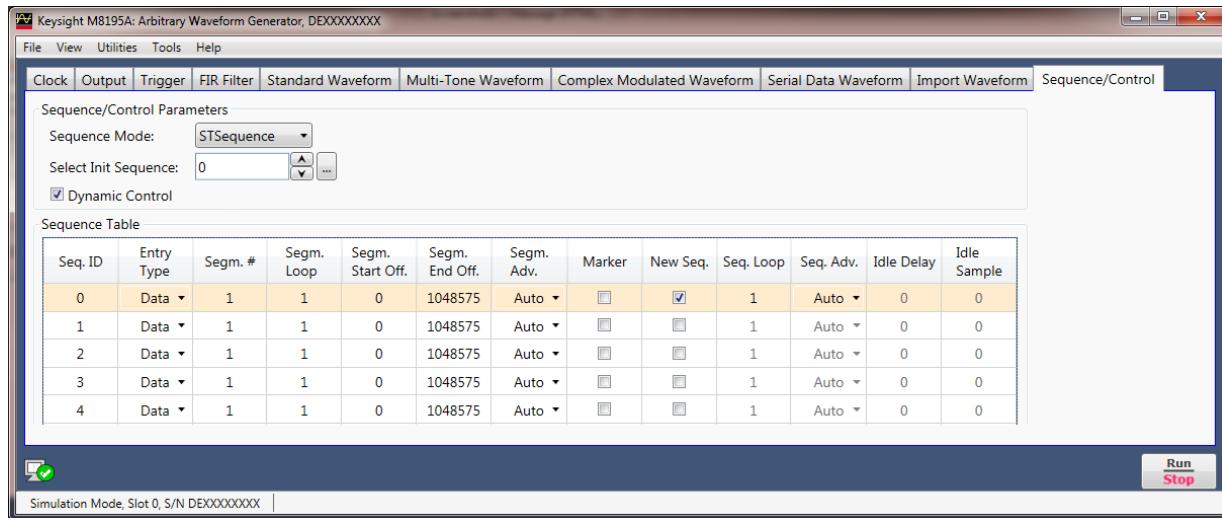


Figure 25: Sequence/Control tab (STSequence mode with Dynamic Control on and Signal Generation stopped)

The following figure shows Sequence/Control tab (STSequence Mode with Dynamic Control on and signal generation started):

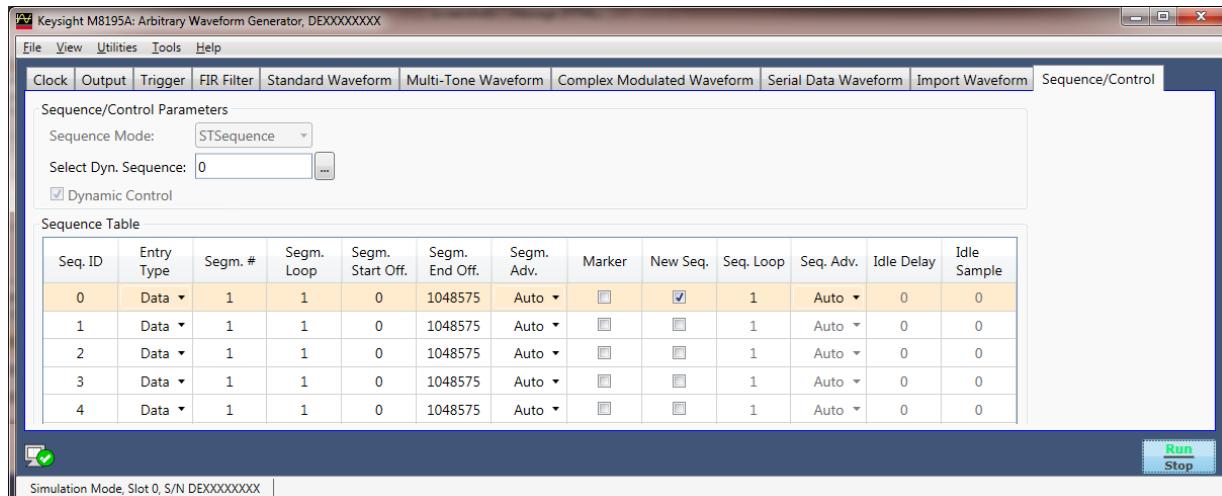


Figure 26: Sequence/Control tab (STSequence mode with Dynamic Control on and Signal Generation started)

This tab has the following controls:

Sequence/Control Parameters

- Sequence Mode
 - Allows to select a sequence mode (ARBitrary, STSequence, or STSCenario):
 - ARBitrary – Generate arbitrary waveform segments
 - STSequence – Generate sequences of segments
 - STSCenario – Generate scenarios (sequences of sequences)
- Advance Mode
 - Specifies the advance mode for waveform segment or sequence depending on selected sequence mode. This option is available only if Arbitrary or STScenario mode is selected.
- Segment Loop
 - Specifies the number of times a segment will be executed. This option is available only if Arbitrary mode is selected.
- Select Segment
 - Allows to select the segment that has to be executed. This option is available only if Arbitrary mode is selected.
- Scenario Loop
 - Specifies the number of times a scenario will be executed. This option is available only if STScenario mode is selected.
- Select Scenario
 - Allows to select the scenario that has to be executed. This option is available only if STScenario mode is selected.
- Select Sequence
 - Allows to select the sequence that has to be executed. This option is available only if STSequence mode is selected.
- Dynamic Control
 - Enable or disable dynamic sequence control. If dynamic control is switched on, segments or sequences can be switched dynamically when signal generation is active. This option is available only if Arbitrary or STSequence mode is selected.
- Select Init Segment
 - Select the initial segment to be played when dynamic control for segments is enabled . This option is available only if Arbitrary mode is selected.
- Select Init Sequence
 - Select the initial sequence to be played when dynamic control for sequences is enabled. This option is available only if STSequence mode is selected.
- Select Dyn. Sequence
 - Allows to select the next sequence to be played when dynamic control for segments or sequences is enabled. This option is available only if Arbitrary or STSequence mode is selected.

Sequence Table

- Seq.ID (Sequence ID)
When a new sequence entry is created, it is automatically allocated a numeric ID termed as Seq.ID. First entry has Seq.ID Q, second 1, and so on.
- Entry Type
 - Data: Each data entry has a waveform segment associated with it which is played during sequence execution. It is also possible to specify the number of iterations for the segment. Amplitude and frequency table is available for standard data entry in interpolated mode.
 - Idle: Idle entry allows setting a pause between segments in a granularity that is smaller than the sync clock granularity. You can specify the sample to be played during the pause. A minimum length of this pause is required. The idle command segment is treated as a segment within sequences or scenarios. There is no segment loop count but a sequence loop counter value is required for cases where the idle command segment is the first segment of a sequence.
 - Empty: Select this option to create an empty segment entry. An entry after the empty segment is automatically marked as a new sequence.
- Segm.# (Segment Number)
Allows to enter the segment number.
- Segm. Loop (Segment Loop)
Specifies the segment loop count (number of times the selected sequence entry is repeated).
- Segm. Start Off. (Segment Start Offset)
Allows specifying a segment start address in samples, if only part of a segment loaded into waveform data memory is to be used. The value must obey the granularity of the selected waveform output mode. .
- Segm. End Off. (Segment End Offset)
Allows specifying a segment end address in samples if only part of a segment loaded into waveform data memory is to be used. The value must obey the granularity of the selected waveform output mode.
- Segm.Adv. (Segment Advance)
Allows the user to set the segment advancement mode.
Any of the following segment advancement modes can be selected:
 - Auto (Automatic): After having executed all loops, the sequencer advances to the next element automatically. No external interaction is required for advancement.
 - Cond (Conditional): The sequencer repeats the current element until it receives the correct advancement event. After having received the advancement event, the current element is played to the end before switching to the next one.
 - Repeat: After having executed all loops the sequencer stops and plays the last value of the current element. After having received the advancement event, the sequencer starts playing the next element. When receiving the advancement event before having played all repetitions, all repetitions will be played before moving to the next element.
 - Single: After having executed an element once, the sequencer stops and plays the last value of the element. After having received the next advancement event, the process is repeated until having executed all loops of the current element. Then the execution advances to the next element.

- Marker
This option allows to enable or disable the marker.
- New Seq. (New Sequence)
Select the check box to start a new sequence.
- Scen. End
Select the check box to mark end of the scenario.
- Seq. Loop (Sequence Loop)
Specifies the sequence loop count (number of times the selected sequence is to be repeated).
- Seq.Adv. (Sequence Advance)
Allows the user to set the sequence advancement mode.
Any of the following sequence advancement modes can be selected:
 - Auto (Automatic): After having executed all loops, the sequencer advances to the next sequence automatically. No external interaction is required for advancement.
 - Cond (Conditional): The sequencer repeats the current sequence until it receives the correct advancement event. After having received the advancement event, the current sequence is played to the end before switching to the next one.
 - Repeat: After having executed all loops the sequencer stops and plays the last value of the current sequence. After having received the advancement event, the sequencer starts playing the next sequence. When receiving the advancement event before having played all repetitions, all repetitions will be played before moving to the next sequence.
 - Single: Once a sequence is executed, the sequencer stops and plays the last value of the sequence. After having received the next advancement event, the process is repeated until having executed all loops of the current sequence. Then the execution advances to the next sequence.
- Idle Delay
The field is enabled only when the Entry Type is chosen as “Idle”. It is used to insert a numeric idle delay value into the sequence.
- Idle Sample
Idle Sample is the sample played during the pause time. The field is enabled only when the Entry Type is chosen as “Idle”. It is used to insert a numeric idle sample value into the sequence. In case of interpolated mode, there are two idle sample values corresponding to I and Q data, respectively. So, for interpolated mode there will be two columns for idle samples i.e. Idle Samp. I and Idle Samp. Q.
-  (Insert Above)
Insert a new sequence entry row above the selected entry.
-  (Insert Below)
Insert a new sequence entry row below the selected entry.
-  (Delete)
Delete the selected sequence entries.

-  (Cut)
Cut the selected sequence entries for pasting to another position in the present or a new sequence. “Paste” option will be enabled.
-  (Copy)
Copy the selected sequence entries for pasting to another position in the present or a new sequence. “Paste” option will be enabled.
-  (Paste)
Paste the copied or cut sequence entries to the target sequence entry.
-  (Clear)
Use this option to undo the cut or copy action. Once the option is clicked, data on the clipboard will be erased, and the “Paste” option will be automatically disabled.
- Send To Instrument
Send sequence configuration to the instrument.
- Read From Instrument
Extract existing sequence configuration from the instrument.

Licenses and Options Section

- Installed Options
This field displays the installed options for the M8195A module.
- Installed Licenses
This field displays the installed licenses for the M8195A module.

2.16 Correction File Format

A correction file is an ASCII delimited file carrying all the information required to compensate or embed a given frequency response in the multi-tone, complex modulation and serial data signals. The file must be composed of a header including a series of lines with identifiers and parameters, and a list of numerical correction factors. In lines including more than one item (i.e., one identifier and one parameter), the items must be separated using commas. Identifiers and parameters are not case sensitive.

These are the significant fields for the header:

- InputBlockSize: It states the number of valid correction factors in the file. It is a mandatory field.
- XStart: It is frequency in Hz corresponding to the first entry in the correction factor section of the file. It is a mandatory field for serial data and multi-tone generation in direct mode and optional for multitone in upconverter mode and complex modulation.
- XDelta: It is frequency distance in Hz between consecutive entries in the correction factor section of the file. It is a mandatory field.
- YUnit: Units for the amplitude values in the correction factor section of the file. Parameter associated to it may be 'dB' (for logarithmic relative amplitudes) or 'lin' (for dimensionless linear relative amplitude). This parameter is optional and its default value is 'lin'. Phase unit must be always stated in radians.

The order of the above entries is not relevant. The correction factor section starts with a line including a single 'Y' or 'y' character. Entries in this section are made by Amp1(Fi), Phase1(Fi) pairs. In particular, this format is compatible with adaptive equalizer files exported in comma-separated value (CSV) format from the Keysight 89600 VSA software package. These files reflect the channel response corrected by the equalizer so they should be applied through the selection of the 'Channel_Response' option in the corresponding 'CorrectionMode' drop-down list in the 'Corrections' section of the 'Multi-Tone' panel. 'Complex Modulation' and 'Serial Data' panels always expect "frequency response" data so correction will be obtained by inverting the supplied data. Comments must start with the '//' character sequence and may use a complete line or be located at the end of any valid line. Empty lines are also valid.

For signal created from the 'Serial Data Panel', when correction data is obtained through an oscilloscope and the Keysight 89600 VSA software, adaptive equalizer analysis should be applied to a real-only baseband signal (so only one oscilloscope channel is involved). A direct, bandwidth-limited, straight-forward frequency response information can be obtained by generating a NRZ signal with sufficient bandwidth and a convenient edge shape filter that can be handled by the analysis software (i.e. raised cosine) and setting up the analysis according to the signal characteristics(symbol rate, filter parameter). NRZ signals can be analyzed through the 89600 VSA software by selecting "BPSK" as the modulation scheme.

This is an example correction file:

```
// MyCorrectionFile
InputBlockSize, 1024
XStart, 1.0E+09 // 1.0GHz
XDelta, 1.0E+06
YUnit, lin
Y
0.987, -0.2343
0.995, 0.5674
...
...
1.269, -0.765
```

The above files are composed of a header with relevant information. In these particular cases, the files contain 1024 linear correction factors spaced by 1 MHz and starting at 1GHz. The 'Y' character indicates the starting point for the correction factor list composed of 1024 lines with amplitude/phase pairs separated by commas. For one-channel files there is a amplitude/phase pair per line while for two channel files there are two pairs (Amp1, Phase1, Amp2, Phase2).

The way this information is applied by the Soft Front Panel software depends on the signal generation mode and the signal category. For direct conversion multi-tone RF generation modes ('Generate IQ Data' unchecked), corrections are applied directly to the tones based on their absolute frequency. For up-converted multi-tone baseband generation(I/Q) modes ('Generate IQ Data' checked), corrections are applied to the complex baseband signals. So, the internal or external carrier frequency is represented by the central entry in the list (i.e., entry #512 in the 1024 entries example shown above) regardless of the 'XStart' parameter. For Complex Modulated waveforms, corrections are always applied to the complex baseband signals regardless of the 'Generate IQ Data' checkbox setting so, as it happens with the correction of multi-tone baseband signals, the internal or external carrier frequency is represented by the central entry in the list.

3 Sequencing

| | | | |
|------|--------------------------|---|-----|
| 3.1 | Introduction | / | 109 |
| 3.2 | Sequencing Hierarchy | / | 112 |
| 3.3 | Trigger Modes | / | 113 |
| 3.4 | Arm Mode | / | 114 |
| 3.5 | Advancement Modes | / | 114 |
| 3.6 | Sequencer Controls | / | 115 |
| 3.7 | Sequencer Execution Flow | / | 121 |
| 3.8 | Sequencer Modes | / | 122 |
| 3.9 | Dynamic Sequencing | / | 138 |
| 3.10 | Idle Command Segments | / | 141 |
| 3.11 | Limitations | / | 142 |

3.1 Introduction

This chapter describes the sequencing capabilities of the instrument.

3.1.1 Sequencing Internal Memory

Channels sourced from internal memory have the following functionality:

- Waveform generation from internal memory and waveform generation from extended memory always starts synchronously on all four channels
- The start of waveform generation can be initiated from the SFP, a software or hardware trigger-
- Number of segments from: 1
- Minimum segment length: 128 samples
- Waveform Granularity: 128 samples
- Maximum segment length: See section [1.5.4](#).

- Loop count: Always infinite. i.e. when the waveform generation has started, waveforms are being generated until the instrument is stopped.
- The segment lengths of channel 1, channel 2, channel 3 and channel 4 may be different
- Sequences and scenarios are not available from internal memory

The option sequencing (Option -SEQ) does not affect the capabilities of the internal memory.

3.1.2 Option Sequencing for Extended Memory

The M8195A offers sequencing functionality for channels sourced from extended memory. Option sequencing (Option -SEQ) enables an extended set of sequencing functionality.

With option -SEQ following sequencing functionality is available:

- Up to ~16 Mio unique segments can be defined where each segment length may be different
- Sequencing hierarchy: Segment, sequence, scenario
- Trigger modes: Continuous, triggered, gated
- Arm Mode: Self armed and armed
- Advancement modes: Auto, conditional,repeat, single
- Sequencer modes: Arbitrary, sequence scenario
- Dynamic sequencing

Without option -SEQ the sequencing capabilities of the instrument are:

- One segment. Loop counter for this segment
- Trigger modes: Continuous, triggered, gated
- Arm Mode: Self armed and armed
- Advancement modes: Auto, conditional,repeat, single
- Sequencer Mode: Arbitrary

For operation in instrument mode 'Dual Channel' or instrument mode 'Four Channel', all channels sourced from extended memory of M8195A behave identical with respect to sequencing. i.e. there is one sequence table available for the M8195A. Certainly the waveforms of the channels can be different for any segment number.

3.1.3 Sequence Table

The sequencer is implemented in a table. Each table entry consists of a sequence vector, which contains all the necessary information that is required to play one single waveform segment like loop counter values, advancement parameters and references to the sample memory. Multiple adjacent sequence vectors can also be played together within one run. The first sequence table entry is marked with a start pointer. After having finished one segment, the next table entry of the list is selected. If an actually executed segment is the last segment of a loop a jump to the starting point of the loop might be initiated depending on the loop count.

The following drawing shows an example:

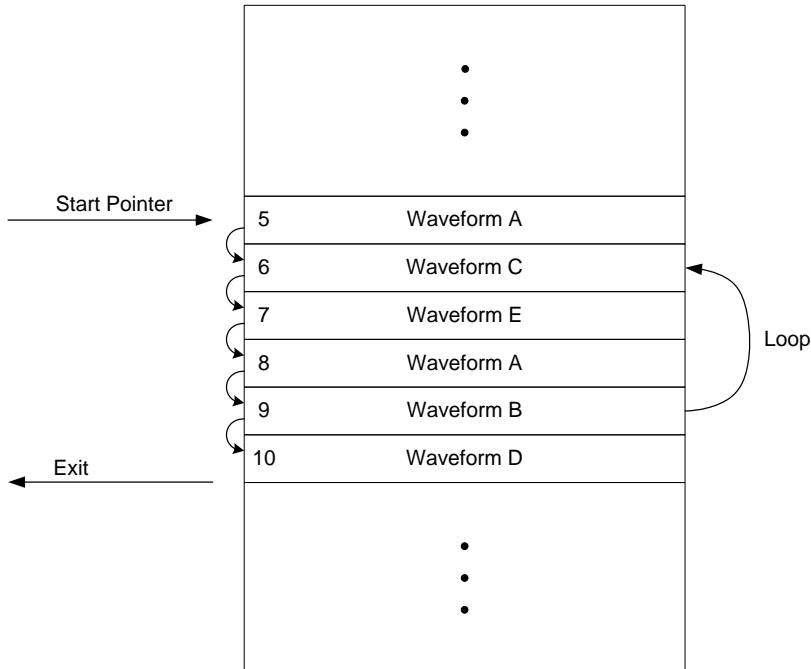


Figure 27: Sequence table

The execution flow is started at address 5 and the waveform of every table entry is played. Within the sequence, a loop from address 9 to 6 is executed for a number of times, specified by loop counter values. It is possible to access the same sample data from different sequence vectors. In this example, waveform A is accessed from sequence vector 5 and 8.

3.1.4 Sequencer Granularity

The sequencer is running at a lower clock speed than the sample rate of the instrument. Therefore, the sequencer has to play multiple samples within one sync clock cycle.

The number of samples played within one sync clock cycle is called waveform granularity or segment granularity. For details, refer to the block diagrams in section 1.5.4.

3.2 Sequencing Hierarchy

3.2.1 Segment

A waveform segment consists of a defined number of samples, which are played, in a consecutive order. It is treated as a unit and can be repeated a specified number of times or can run continuously. The sample count of segments must be in multiples of the segment granularity. A minimum length is also required (see datasheet of the instrument).

A segment can be played standalone (see [Arbitrary Mode](#)) or can be part of a sequence.

3.2.2 Sequence

Multiple segments can be combined to a sequence. A sequence can be executed continuously or for a specified number of times.

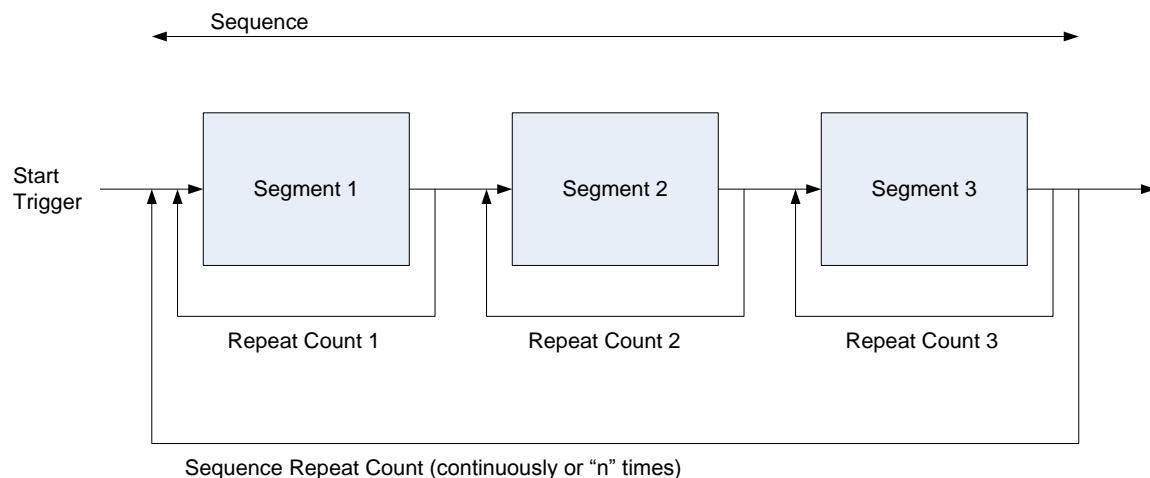


Figure 28: Sequence

A sequence can be played standalone (see [Sequence Mode](#)) or can be part of a scenario.

3.2.3 Scenario

Multiple sequences can be combined into a scenario (see [Scenario Mode](#)). A scenario can be executed continuously or a specified number of times.

3.3 Trigger Modes

The trigger mode defines the way, how segments, sequences and scenarios begin with playing waveform data. After having setup the instrument, it is started. Then the start of segments, sequences and scenarios depends on the different trigger modes.

3.3.1 Continuous

In the trigger mode **Continuous**, the sequencer is started immediately after the instrument. In this mode, the waveform execution is infinite.

3.3.2 Triggered

In the trigger mode **Triggered**, the sequencer needs a trigger to start. After having received the trigger, the waveform is played a defined number of times, and then the sequencer is stopped again and is prepared to accept the next trigger. Every trigger that occurs before the currently running segment/sequence/scenario has completed, is ignored. Alternatively, after having received a trigger, a waveform can also be played infinitely. See [Sequencer Modes](#) for more details.

3.3.3 Gated

In the trigger mode **Gated**, both edges of the gate signal are used to start and stop the execution of the sequencer. After being stopped, the sequencer is prepared to accept a new rising edge of the gate and can be restarted again.

In Gated Mode, the advancement mode of the top level (e.g. sequence advancement mode for sequences) must be set to Continuous.

3.4 Arm Mode

Sometimes it is desired to play an idle waveform instead of a static idle value before having started to play the real waveform. With the arm mode it is possible to select the output signal of the instrument before having started the sequencer.

3.4.1 Self Armed

Whenever the arm mode is set to **Self Armed**, the instrument starts as defined by the selected trigger mode.

3.4.2 Armed

For all cases where the trigger mode is set to **Continuous** and the arm mode to **Armed**, the first segment/sequence is played infinitely after start. After having received a rising edge of Enable, the sequence/scenario advances to the next segment/sequence and continues to execute as described in trigger mode **Continuous**. This mode doesn't make any sense for the execution of standalone segments or for the trigger modes **Triggered** and **Gated**. Therefore in these cases the described behavior is not available and **Armed** is treated like **Self Armed** with an additional enable flag as start condition.

3.5 Advancement Modes

The advancement mode specifies the way of how one element like a segment, sequence or scenario advances to the next element or how it is repeated.

The advancement mode can be individually specified for each single element. The exact behavior depends on the sequencing, arm and trigger mode.

There could be different advancement modes on different hierarchy levels. Some of these modes require an advancement event to proceed. In cases where the advancement event has to be evaluated simultaneously in multiple hierarchy levels, the output behavior could be unexpected, especially when conditional advancement modes are used. For more details, refer to the examples given in the section [Sequencer Modes](#).

3.5.1 Auto

After having executed all loops, the sequencer advances to the next element automatically. No external interaction is required for advancement.

3.5.2 Conditional

The sequencer repeats the current element until it receives the correct advancement event. After having received the advancement event, the current element is played to the end before switching to the next one.

3.5.3 Repeated

After having executed all loops the sequencer stops and plays the last value of the current element. This last value can be specified in the corresponding sequence vector (default value is the offset voltage). After having received the advancement event, the sequencer starts playing the next element. When receiving the advancement event before having played all repetitions, all repetitions will be played before moving to the next element.

3.5.4 Single

After having executed an element once, the sequencer stops and plays the last value of the element. This last value can be specified in the corresponding sequence vector (default value is the offset voltage). After having received the next advancement event the process is repeated until having executed all loops of the current element. Then the execution advances to the next element.

3.6 Sequencer Controls

Sequencer Controls are used to influence the sequencer. So they can control the waveform generation.

3.6.1 External Inputs

3.6.1.1 TRIGGER/EVENT

The M8195A accepts a wide range of external trigger signal levels to easily adapt to a measurement setup. The input threshold is user configurable along with the polarity or whether rising, falling or both edges are to be taken into account. Two modes of operation are available: Asynchronous and Synchronous triggering.

3.6.1.1.1 Synchronous Triggering

The TRIGGER and EVENT input signals are clocked internally with the SYNC clock. [SYNC clock = Sample clock divided by 256]. To reduce the TRIGGER to DATA out uncertainty the signal applied to the external input connector needs to meet a setup and hold window. The timing is specified with respect to the SYNC Clk Out port. See the data sheet for further details.

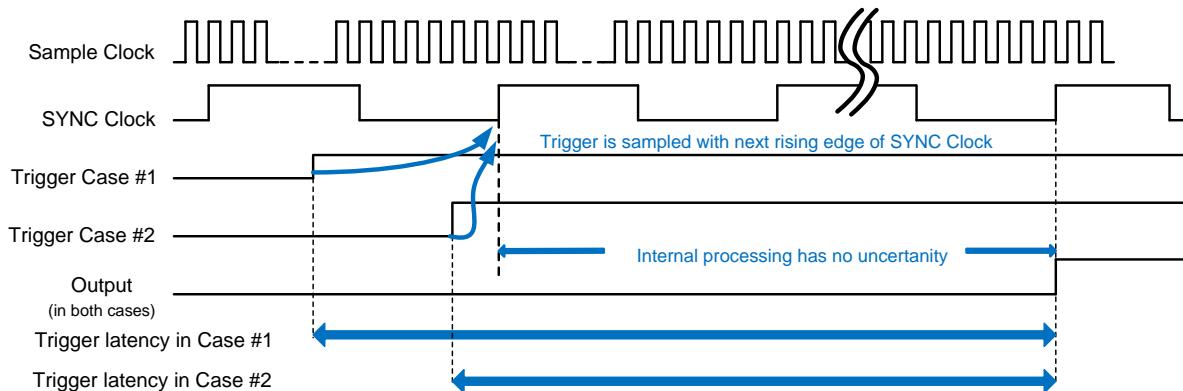


Figure 29: TRIGGER/EVENT synchronous to the sync clock (synchronous tmode)

3.6.1.1.2 Asynchronous Triggering

In synchronous trigger mode the incoming trigger and event signals are sampled with the SYNC clock which is the DAC sample rate divided by 256 and the input signals need to be provided synchronous to the SYNC clock to get a precise output signal.

When using the asynchronous mode, the trigger and event input signals are sampled with a clock that is the DAC sampling rate divided by 8. This provides a more precise trigger/event to ouput latency without the need of providing the inputs synchronous to any reference.

3.6.2 Logical Functions

3.6.2.1 Trigger/Gate/Enable

The **trigger**, **gate** and **enable** signals are used to control the start behavior of the sequencer, depending on the selected mode. The **trigger** starts the sequencer in trigger mode **triggered**; the **gate** has the corresponding functionality (start and stop) in trigger mode **gated**. The **enable** is needed in the **armed/continuous** mode. In this mode the first element is hold in the conditional advancement mode until **enable** becomes active. During further loops of the sequence or scenario, the **enable** is ignored and the element is executed with the advancement mode specified in the sequence table. So the **enable** allows providing not only an initial offset value, before the real start of the sequencer, but also an initial segment or sequence.

3.6.2.2 Advancement Event

The advancement event is used to advance within a scenario or sequence. Responsible for the type of advancement is the selected advancement mode of the element. The advancement event is stored internally until the sequencer uses it.

Example:

When receiving an advancement event while executing a conditional segment, the advancement event is stored until reaching the end of the segment where the advancement is used. Then the stored advancement event is cleared and the instrument is able to receive the next one.

3.6.2.3 Dynamic Select

The instrument provides a dynamic sequencing mode, which allows changing the actually running segment or sequence without stopping and reprogramming the instrument. The selected sequencer index is modified either by the external DYNAMIC CONTROL input of the M8197A in multi module configuration or via remote programming. Up to 16M sequencer table indices can be addressed.

3.6.2.4 Run

The run input is a software button or command, which switches the instrument from programming mode to run mode.

3.6.3 Internal Trigger Generator

The M8195A provides a configurable internal trigger generator that allows for generation of a periodic trigger signal that is frequency locked to the clock of the sequencer engine. In Gated mode, the internal trigger generator provides a gate with a width of 50% of the trigger generator period.

3.6.4 Mapping External Inputs to Logical Functions

The logical functions controlling the sequencer can be connected to multiple sources. The following table shows all possible mappings.

Table 18: Mapping external inputs to logical functions

| Functions | Inputs | | | |
|--------------------------------|---------------------------------------|----------------------|--------------------------------|----------|
| | Trigger/Gate Input (SMA Connector) | Trigger Generator | Event Input (SMA Connector) | Software |
| Trigger/Gate | Default | | | |
| Enable (= Start in armed mode) | Default (Armed) | | | |
| Advancement Event | | | Default | |
| Dynamic Select | | | | Default |
| RUN | | | | |

NOTE

The software controls are logically ored with the external input.

Dynamic control inputs are only available, when using the M8197A. Then in case of the dynamic control, the software controls have precedence unless the hardware inputs are explicitly disabled using the commands.

See also chapter [Trigger Tab](#).

3.7 Sequencer Execution Flow

The given drawing shows an overview of the different trigger modes and the interaction with some of the conditional inputs.

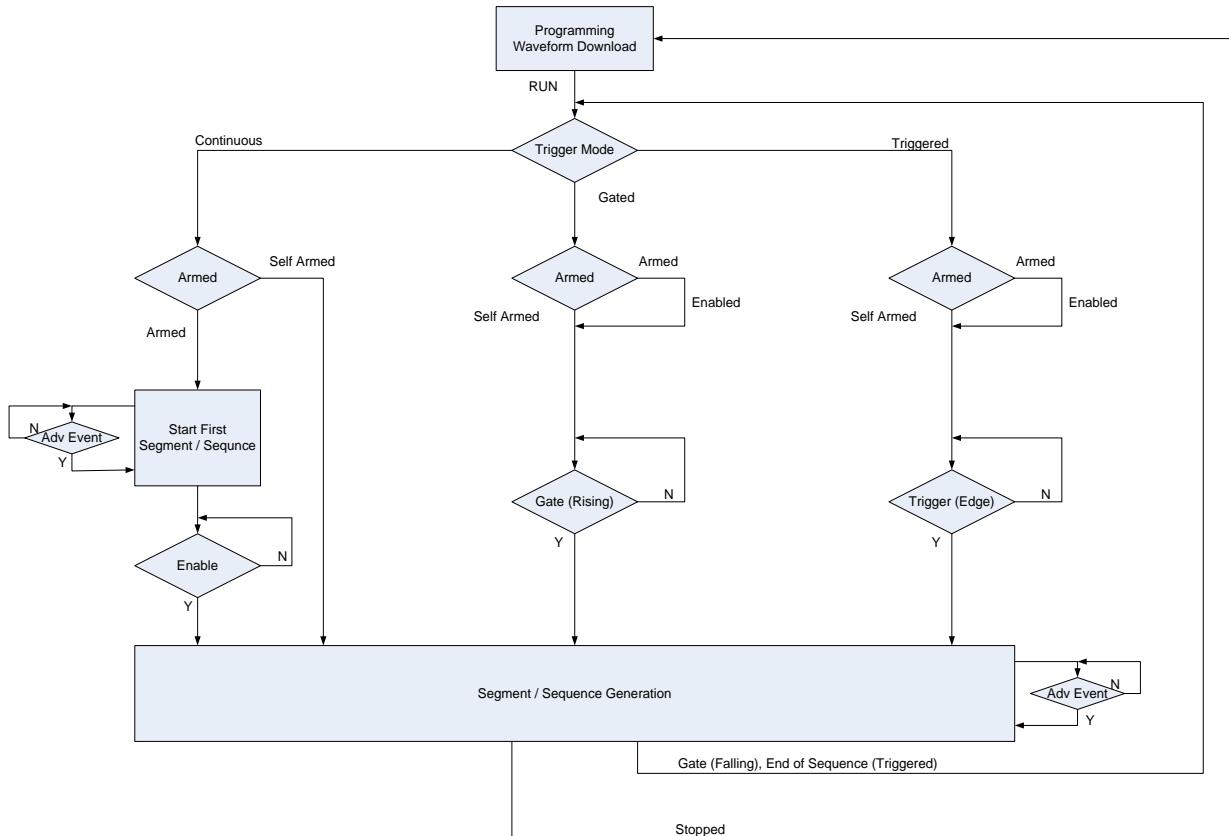


Figure 30: Sequencer execution flow

RUN is moving the instrument from the programming mode to execution mode. Dependent on the selected trigger mode, the behavior in **Armed** mode is different. In trigger mode **Continuous**, the enable signal is used to control the execution of the first segment or sequence. In trigger mode **Gated** or **Triggered**, the **enable** is used as an additional start input.

3.8 Sequencer Modes

This section describes the various sequence modes and their behavior depending on trigger mode and arm mode. Some of them are illustrated with examples. Every run of the sequencer starts with a static offset value, which represents the DAC value zero in the signed interpretation.

So this value is:

$$\text{Offset} = \frac{\text{Max.Dac} - \text{Min.Dac}}{2}$$

A stop (See SCPI command :ABORT[1|2|3|4]) of the instrument is an abort initiated by software which is unrelated to the currently running sequencer. So the currently running segment/sequence or scenario is not completed before stopping.

3.8.1 Arbitrary Mode

In Arbitrary Mode, a single segment is played.

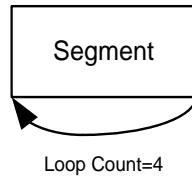


Figure 31: Segment

3.8.1.1 Self Armed

Trigger Mode Continuous After programming, the segment is started automatically and is repeated infinitely.

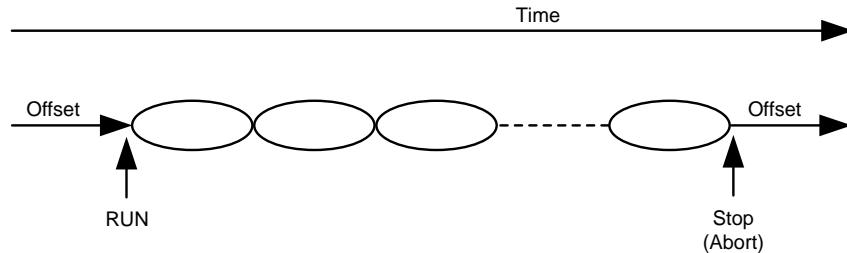


Figure 32: Trigger mode continuous

- Trigger Mode Triggered An Offset value is provided after programming. A trigger starts the segment.
 The following segment advancement modes are available:
 Auto: The segment is executed the number of times specified by its loop count. Then the last sample is played at the end.
 Repeat: This advancement mode is quite the same like "Auto" with the difference that an advancement event is required at the end.
 Single: An advancement event is required for each segment repetition.
 Conditional: The segment is played infinitely after receiving a trigger. After being stopped (See SCPI command :ABORT[1|2|3|4]) the offset value is played.

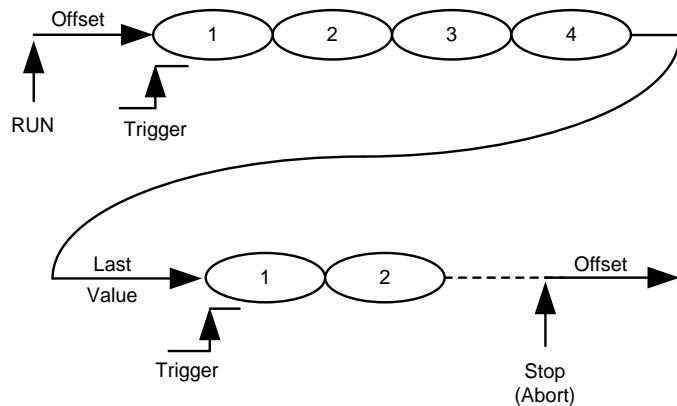


Figure 33: Segment advance = auto

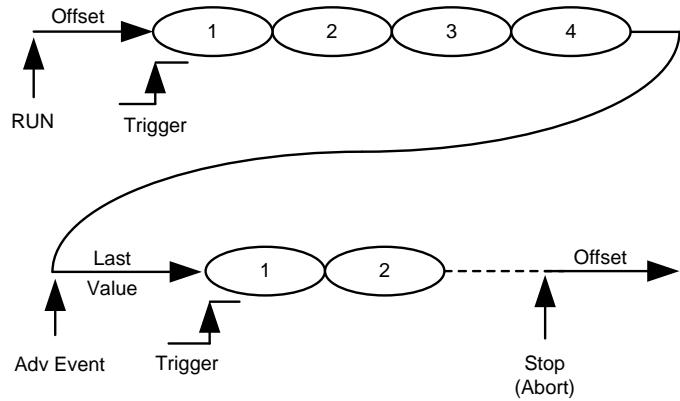


Figure 34: Segment advance = repeat

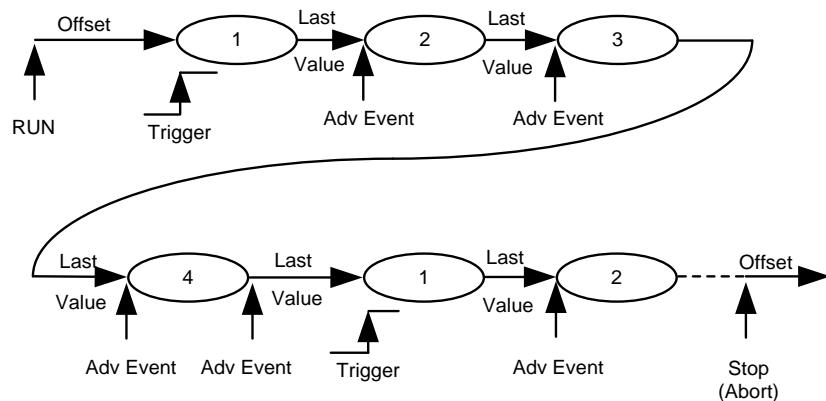


Figure 35: Segment advance = single

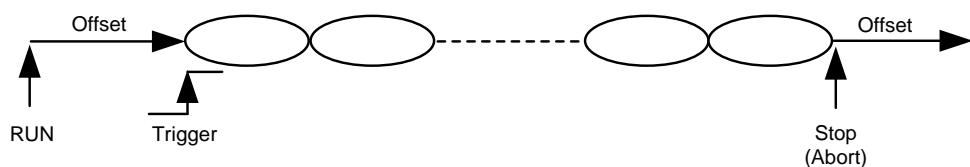


Figure 36: Segment advance = conditional

Trigger Mode Gated

An Offset value is provided after programming. The rising edge of the gate starts the sequence and plays the segment infinitely until receiving the falling edge of the gate. After having received the falling edge of the gate, the segment is played for a number of times specified by the segment loop count. Then the segment is stopped at its end. Then the last sample value is provided.

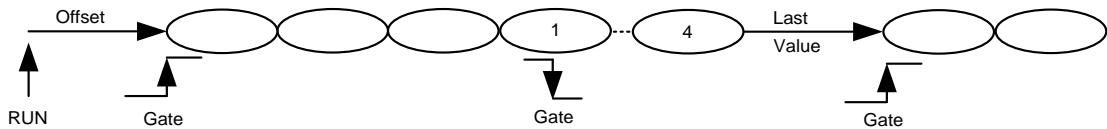


Figure 37: Trigger mode gated

3.8.1.2 Armed

Behavior is like self armed with an additional ENABLE. The enable is evaluated only once at the beginning. Later, changes of this signal are ignored.

Trigger Mode Continuous

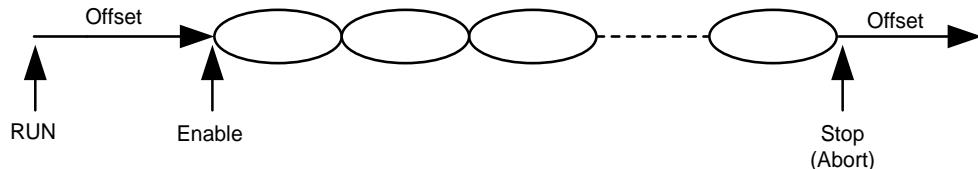


Figure 38: Trigger mode continuous

Trigger Mode Triggered

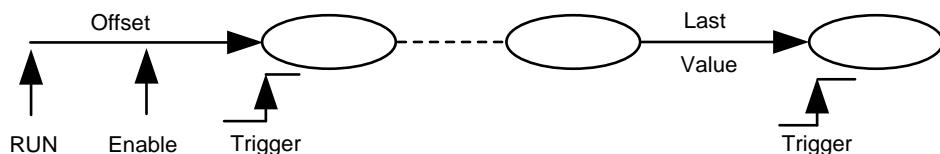


Figure 39: Trigger mode triggered

Trigger Mode Gated

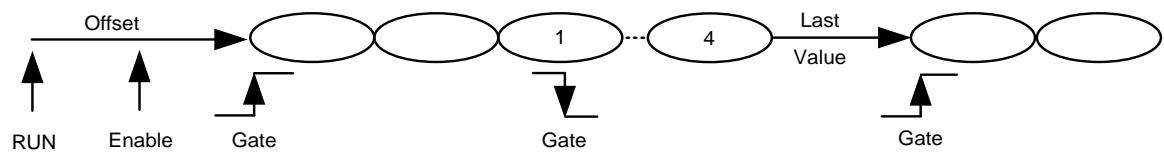


Figure 40: Trigger Mode Gated

3.8.2 Sequence Mode

In Sequence Mode, one or multiple segments are played.

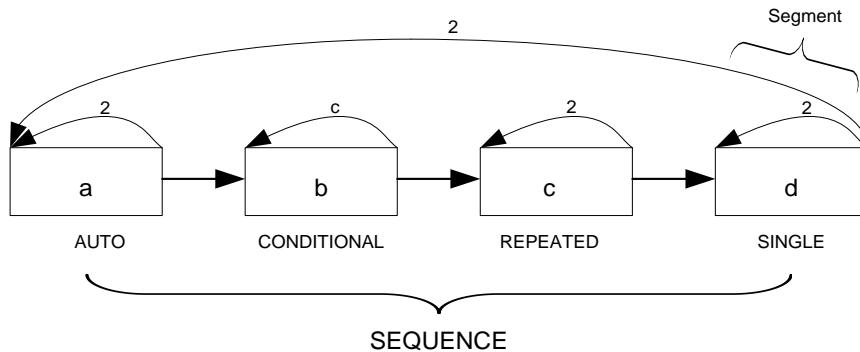


Figure 41: Sequence Mode

3.8.2.1 Self Armed

Trigger Mode
Continuous

After programming, the sequence is started automatically and played infinitely.
The following segment advancement modes are available:

- Auto
- Conditional (Advancement Event)
- Repeated (Advancement Event)
- Single (Advancement Event)

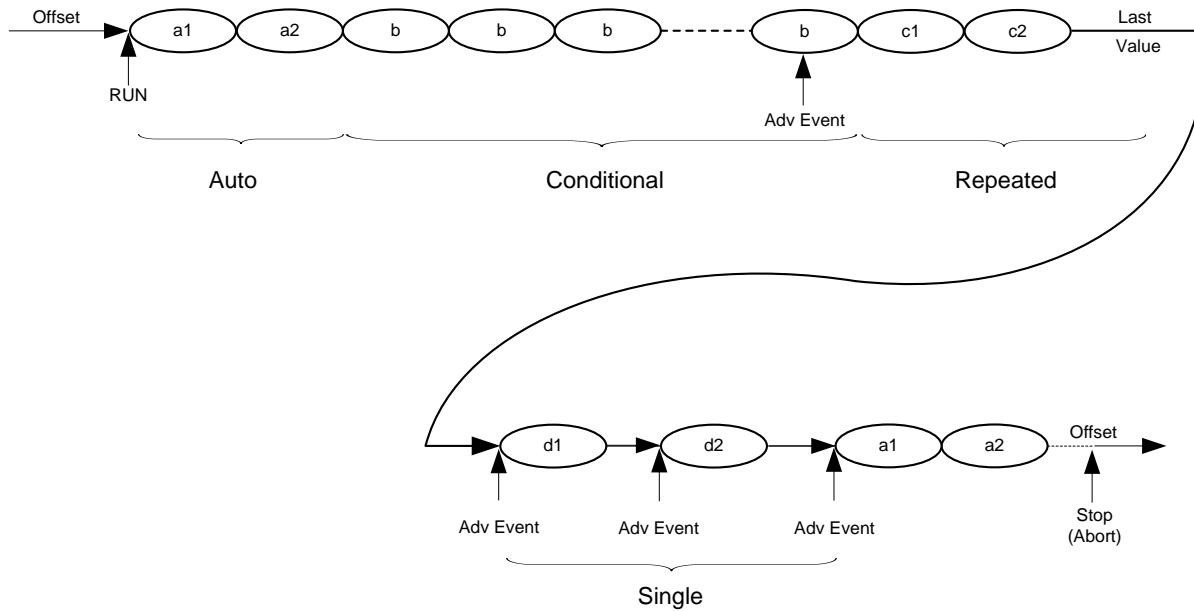


Figure 42: Trigger mode continuous

Trigger Mode Triggered An Offset value is provided after programming. A trigger starts the sequence. The following sequence advancement modes are available:

- Auto: The sequence is executed the number of times specified by its loop count. Then the last sample is played at the end.
- Repeat: This advancement mode is quite the same like “Auto” with the difference that an advancement event is required at the end.
- Single: An advancement event is required for each sequence repetition.
- Conditional: The sequence is played infinitely after receiving a trigger. After being stopped (See SCPI command :ABORT[1|2|3|4]) the offset value is played.

The following segment advancement modes are available:

- Auto
- Conditional (Advancement Event)
- Repeat (Advancement Event)
- Single (Advancement Event)

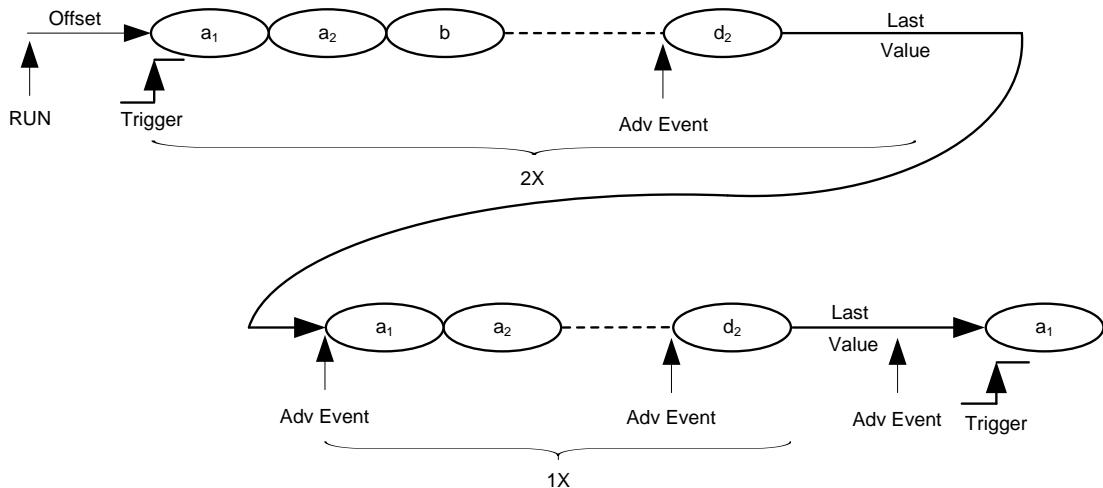


Figure 43: Sequence advance = auto

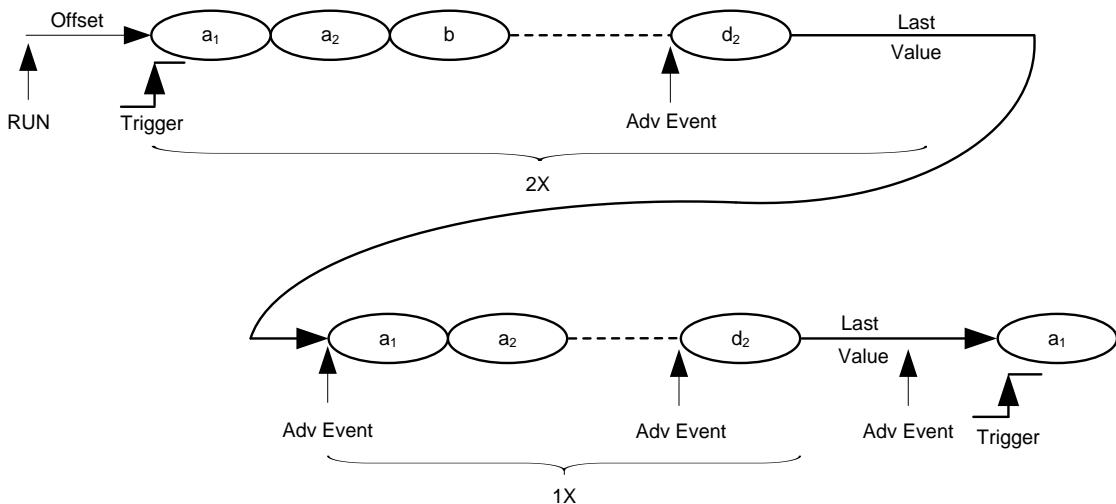


Figure 44: Sequence advance = repeated

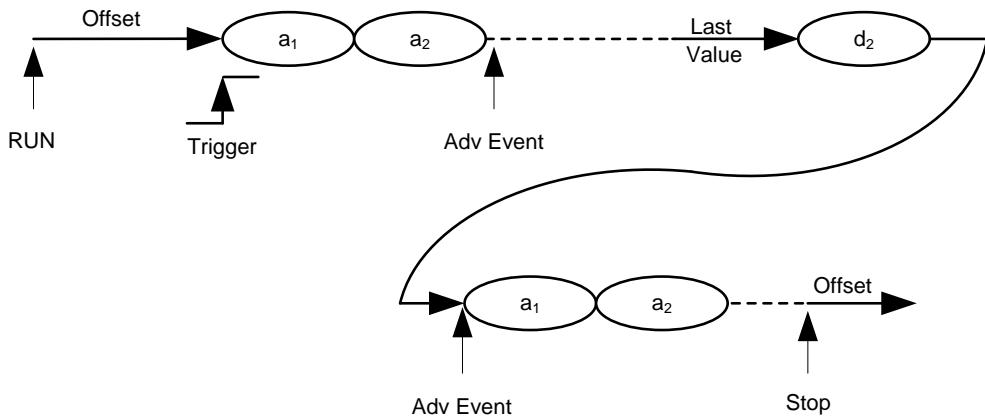


Figure 45: Sequence advance = conditional

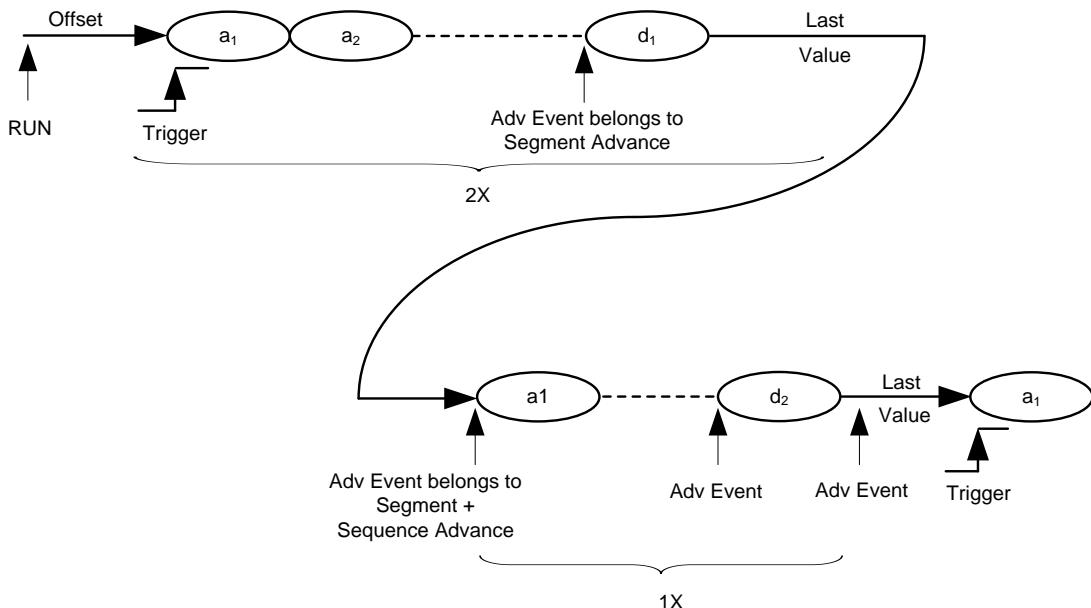


Figure 46: Sequence advancement = single

Trigger Mode Gated

An Offset value is provided after programming. The rising edge of the gate starts the sequence and plays the sequence infinitely until receiving the falling edge of the gate. After having received the falling edge of the gate, the sequence is played for a number of times specified by the sequence loop count. Then the sequence is stopped at its end. Then the last sample value is provided.

The following segment advancement modes are available:

- Auto
- Conditional (Advancement Event)
- Repeat (Advancement Event)
- Single (Advancement Event)

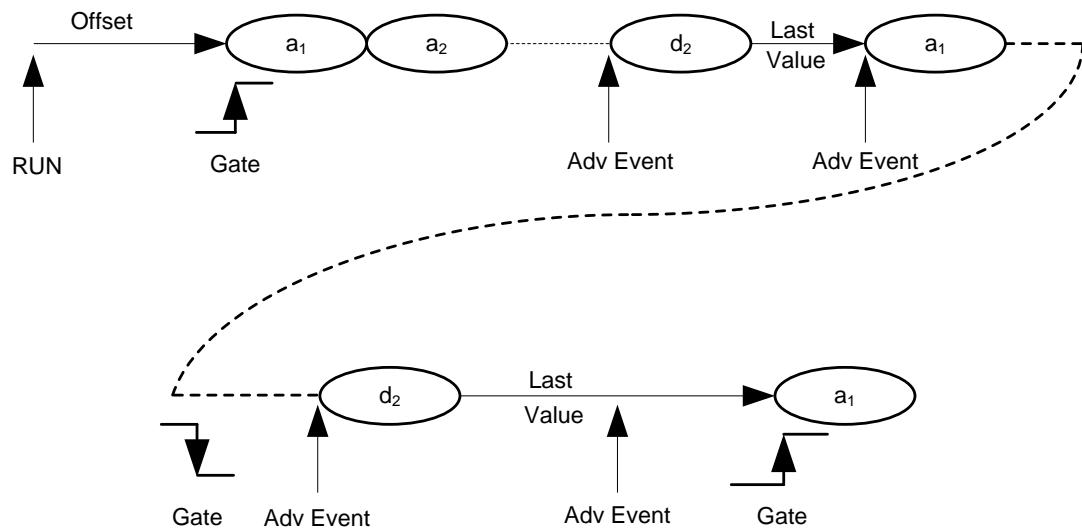


Figure 47: Trigger mode gated

3.8.2.2 Armed

Trigger Mode Continuous After programming, the sequence is started automatically and the first segment is played repetitively until receiving an Enable. Then the first segment is played until the end and the sequence is continued.

The following segment advancement modes are available:

- Auto
- Conditional (Advancement Event)
- Repeat (Advancement Event)
- Single (Advancement Event)

The following sequence advancement mode is available:

The sequence is played infinitely until being stopped. After being restarted, the first segment is played until it receives an enable.

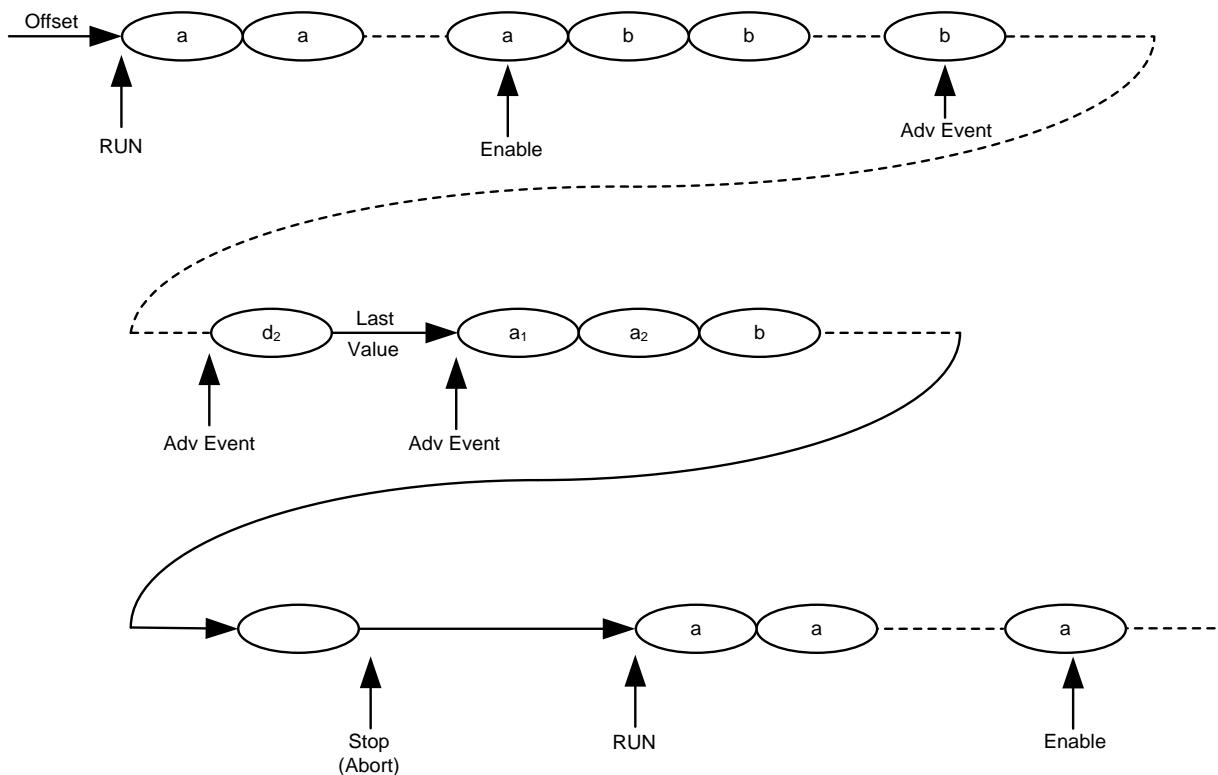


Figure 48: Trigger mode continuous

Trigger Mode Triggered Behavior is like self armed with an additional ENABLE. The enable is evaluated only once at the beginning. Later changes of this signal are ignored.

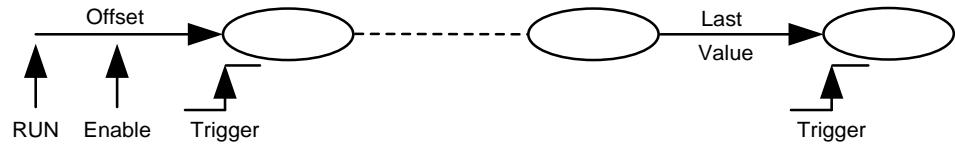


Figure 49: Trigger mode triggered

Trigger Mode Gated Behavior is like self armed with an additional ENABLE. The enable is evaluated only once at the beginning. Later changes of this signal are ignored.

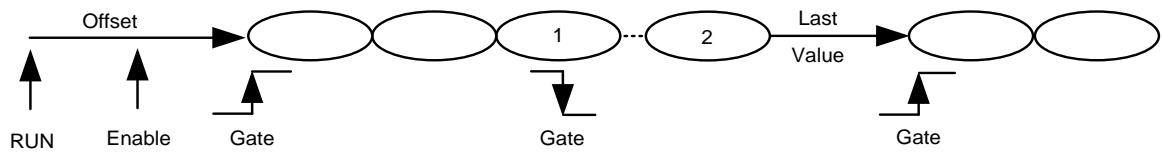


Figure 50: Trigger mode gated

3.8.3 Scenario Mode

In Scenario Mode, one or multiple sequences are played.

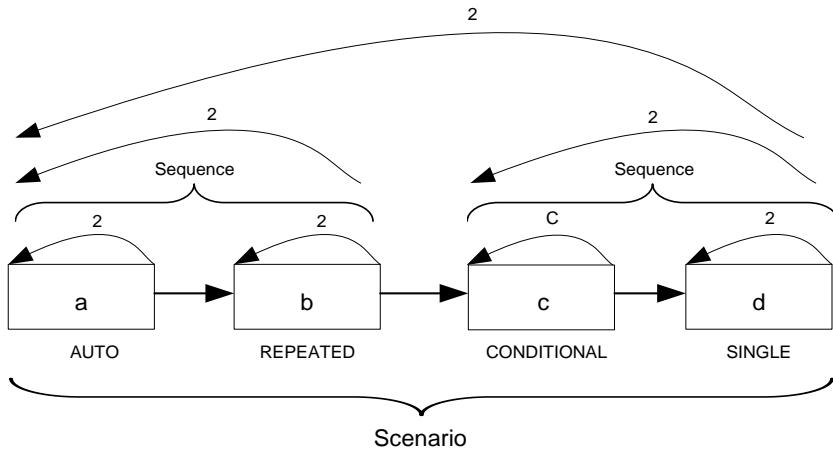


Figure 51: Scenario mode

3.8.3.1 Self Armed

Trigger Mode ContinuousAfter programming, the scenario is started automatically and played infinitely.

The following segment/sequence advancement modes are available:

- Auto
- Conditional (Advancement Event)
- Repeat (Advancement Event)
- Single (Advancement Event)

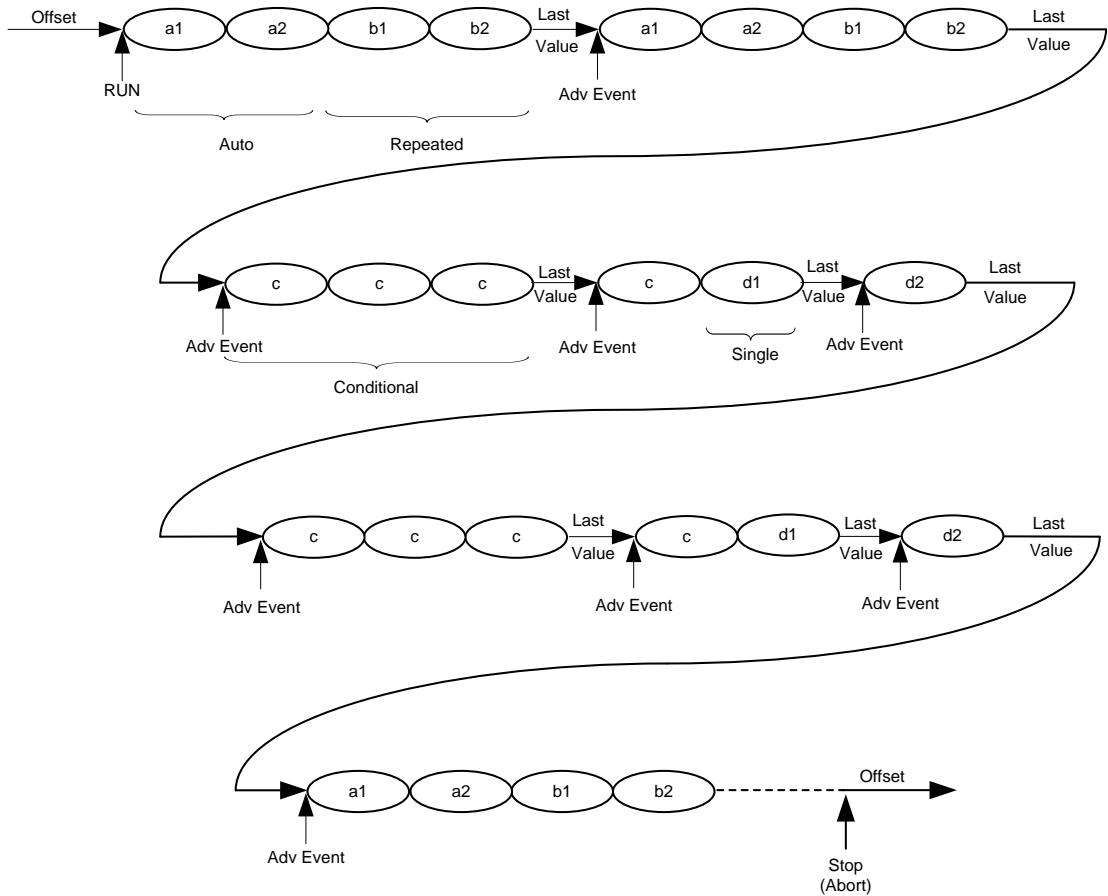


Figure 52: Trigger mode continuous

Trigger Mode Triggered An Offset value is provided after programming. A trigger starts the scenario. The following scenario advancement modes are available:

- Auto: The scenario is executed the number of times specified by its loop count. Then the last sample is played at the end.
- Repeat: This advancement mode is quite the same like "Auto" with the difference that an advancement event is required at the end.
- Single: An advancement event is required for each scenario repetition.
- Conditional: The scenario is played infinitely after receiving a trigger. After being stopped (See SCPI command `:ABORT[1|2|3|4]`) the offset value is played.

The following segment/sequence advancement modes are available:

- Auto
- Conditional (Advancement Event)
- Repeat (Advancement Event)
- Single (Advancement Event)

| | |
|--------------------|--|
| Trigger Mode Gated | An Offset value is provided after programming. The rising edge of the gate starts the scenario and plays the scenario infinitely until receiving the falling edge of the gate. After having received the falling edge of the gate, the scenario is played for a number of times specified by the scenario loop count. Then the scenario is stopped at its end. Then the last sample value is provided. The following segment/sequence advancement modes are available: <ul style="list-style-type: none">• Auto• Conditional (Advancement Event)• Repeat (Advancement Event)• Single (Advancement Event) |
|--------------------|--|

3.8.3.2 Armed

Trigger Mode ContinuousAfter programming, the scenario is started automatically and the first sequence is played repetitively until receiving an Enable. Then the first sequence is played until the end and the scenario is continued

The following segment/sequence advancement modes are available:

- Auto
- Conditional (Advancement Event)
- Repeat (Advancement Event)
- Single (Advancement Event)

The following scenario advancement mode is available:

The scenario is played infinitely until being stopped. After being restarted, the first sequence is played until it receives an enable.

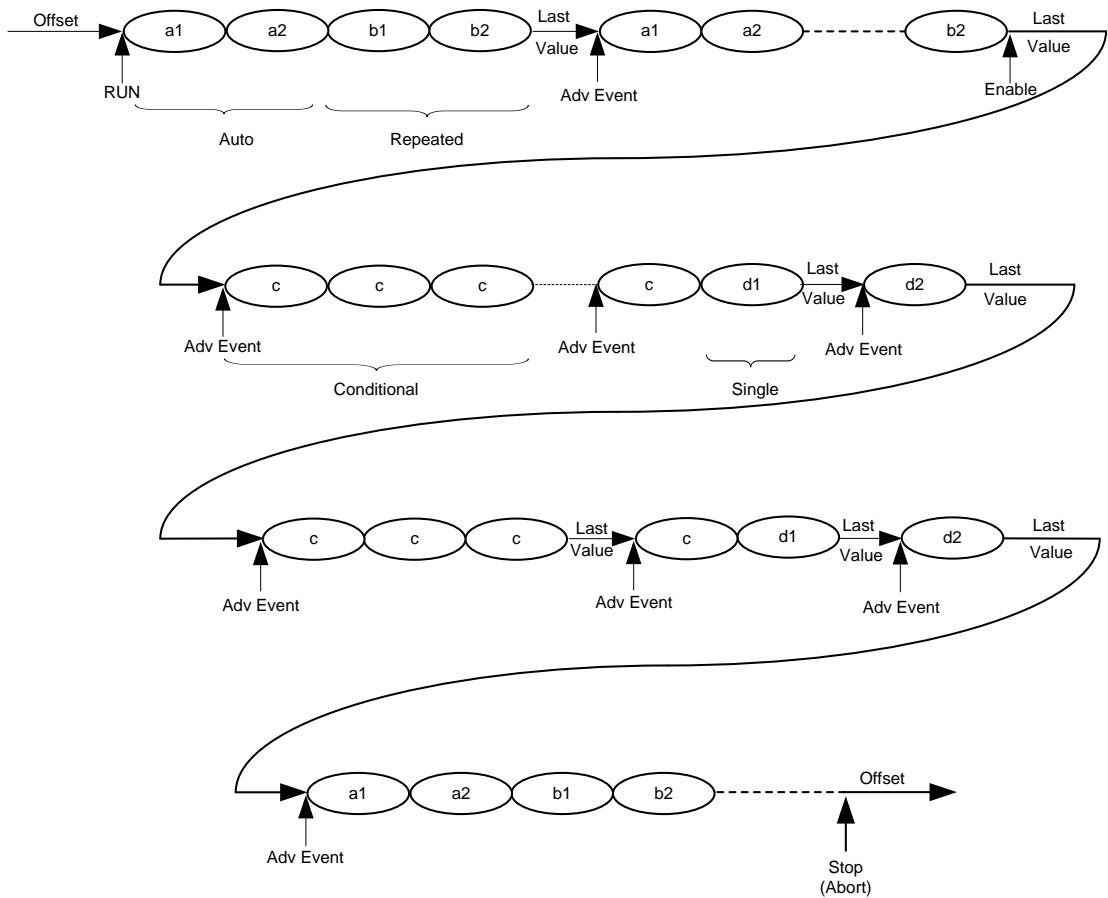


Figure 53: Trigger mode continuous

Trigger Mode Triggered Behavior is like self armed with an additional ENABLE. The enable is evaluated only once at the beginning. Later changes of this signal are ignored.

Trigger Mode Gated Behavior is like self armed with an additional ENABLE. The enable is evaluated only once at the beginning. Later changes of this signal are ignored.

3.9 Dynamic Sequencing

Dynamic Sequencing is a way to dynamically select segments/sequences to be played. The selection can be done by software or by the external dynamic input port (Hardware driven dynamic changes via the dynamic input port are only possible in systems containing the M8197A). The time from selecting a new segment/sequence to the time the change is visible at the output is not specified and is dependent on the actually played segment's/sequence's end relative to arrival of the change event.

When using dynamic sequencing, the arm mode must be set to **self-armed** and all advancement modes must be set to **Auto**. Additionally, the trigger mode **Gated** is not allowed.

3.9.1 Dynamic Continuous

The selected segment or sequence is infinitely played until a new segment/sequence is selected which then is played instead. After a change request the actually selected segment/sequence is played until the end (including loop counts). Then the change towards the new segment or sequence is performed without any gap.

Limitations:

The time between two change requests of waveforms must be bigger than the waveform length of the biggest waveform including the loop counts.

The change delay from applying changes at the dynamic port to seeing them at the output is the trigger to output delay (see datasheet) plus 256 sync clock cycles minimum. Due to instrument internal functionality, this delay cannot be specified exactly and it is always possible that one more segment/sequence A is played before switching to B.

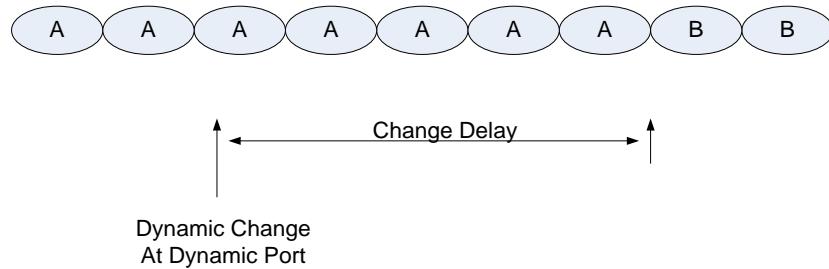


Figure 54: Dynamic continuous

3.9.2 Dynamic Triggered

After having received a trigger, the selected segment or sequence is played (including loop counts). After having selected a new sequence/segment, this sequence/segment is played instead. Based on the timing relationship of the change request and the next trigger, it is possible that the actually selected (old) waveform is played one more time, before switching to the new one.

Limitations:

The trigger period must be bigger than the waveform length of the biggest waveform including the loop counts.

The change delay from applying changes at the dynamic port to seeing them at the output is the trigger to output delay (see datasheet) plus 256 sync clock cycles minimum. Due to instrument internal functionality, this delay cannot be specified exactly and it is always possible that one more segment/sequence A is played before switching to B.

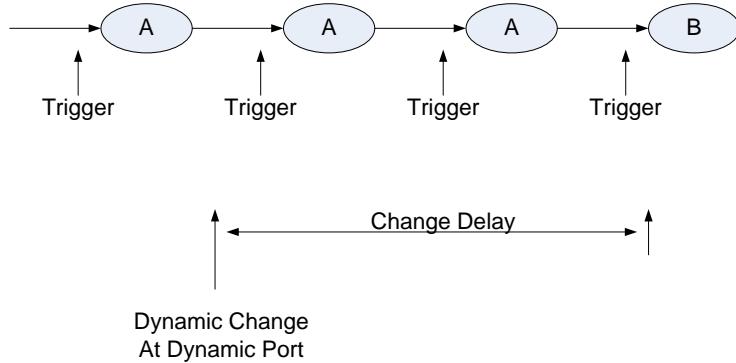


Figure 55: Dynamic triggered

3.10 Idle Command Segments

For some waveform types, like e.g. radar pulses, huge pause segments with a static output are required between the real waveform segments. The gap between the real segments should be adjustable in a fine granularity.

The idle command segment allows setting a pause between segments in a granularity that is smaller than the sync clock granularity. A minimum length of this pause is required (see section [6.18.2](#)). The idle command segment is treated as a segment within sequences or scenarios. There is no segment loop count but a sequence loop counter value is required for cases where the idle command segment is the first segment of a sequence.

The granularity of the idle delay is equal to the waveform sample rate. The following table shows the granularity of the idle delay in DAC samples:

Table 19: Idle delay granularity

| Mode | Idle Delay Granularity |
|---------------------------|------------------------|
| Sample Clock Divider = 1 | 1 DAC Output Sample |
| Sample Clock Divider = 2 | 2 DAC Output Samples |
| Sample Clock Dividier = 4 | 4 DAC Output Samples |

Limitations:

The logic that executes idle command segments uses some elements, which are not in sync clock granularity. To guarantee the trigger to sample output delay or the advancement event to sample output delay, these elements need to be reset before accepting new trigger or advancement events. This requires the waveform generation to be stopped for at least 3 sync clock cycles before being restarted by a trigger or an advancement event. A violation of this requirement leads to an unexpected output behavior for some sync clock cycles.

Multiple adjacent idle command segments are not allowed. If the playtime of one idle command segment is not sufficient, the overall required idle length can be separated into multiple idle command segments where a normal data segment providing the static idle value is put in between. Even this wouldn't be really necessary. One idle command segment (delay of up to 2^{24} sync clock cycles) and one additional small segment (e.g. length: $10 * \text{segment vectors}$, loop count: up to 2^{32}) would provide an idle delay of more than 165 seconds in high speed mode at 64 GSa/s and should be sufficient for most applications.

3.11 Limitations

3.11.1 Segment Length and Linear Playtime

Due to the type of memory technology and the implementation of the memory interface, every physical address jump within the sample memory will reduce the bandwidth at the memory interface. The drawing below shows such an address jump.

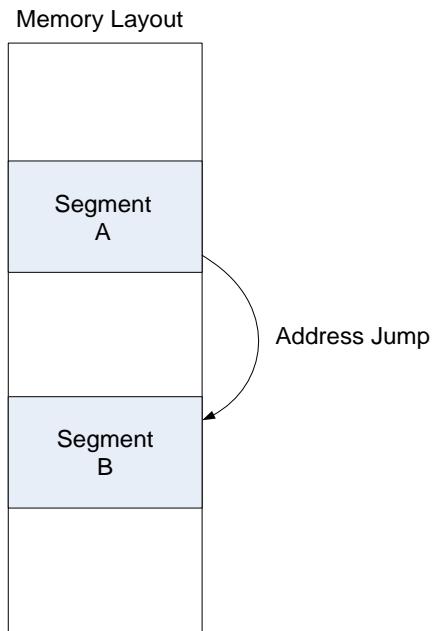


Figure 56: Address jump within segments

To put the density of address jumps below a limit, a minimum segment length of 257 sample vectors (big segment) is required. Small segments (256 vectors down to 5 vectors) are also possible but then the Linear Playtime Requirement must be met.

Linear Playtime Requirement:

The playtime of at least 257 sample vectors (257 sync clock cycles) must be placed in the sample memory in an ascending address order. When writing samples to a totally cleared memory, the order of segments is the order of how these segments are written to the memory.

Idle delay segments are also considered in computing the playtime. The corresponding playtime in sample vectors is computed from the idle delay value. When the data segments before and after the idle delay segment are adjacent in memory the playtime is computed as the sum of all three segments.

The last adjacent segments in a sequence in sequence mode or the last adjacent segments in a scenario in scenario mode can be shorter than 257 sample vectors in total.

Examples:

One segment with 257 or more sample vectors (big segment)

One segment with 129 vectors and a loop count of 2 (Loop count multiplies the segment length)

Two segments with 126 and 131 vectors. (Multiple small segments are combined to meet the requirement)

One segment of 5 vectors and one big segment. (One or multiple small segments which don't meet the linear playtime requirement by themselves, must be located in the memory in front of the next big segment)

Any small segment with a conditional advancement causes the linear playtime requirement to be met automatically. The advancement event to exit the segment is delayed internally until the linear playtime condition is met. A status register signals any linear playtime violation.

Any small segment with an advancement mode set to repeated or single causes the linear playtime requirement to be met automatically. The advancement event is delayed internally until the linear playtime condition is met. A status register signals any linear playtime violation.

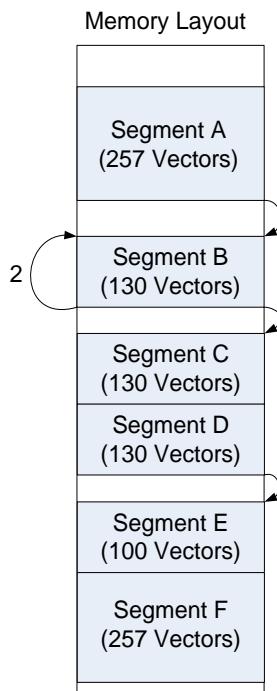


Figure 57: Linear playtime requirement

For the given example sequence the linear play time requirement is met. Segment A is a segment with a play time that is bigger than 256 vectors. Due to its loop count, segment B is also bigger than 256 vectors. Segment C and D are placed next to each other and the resulting length is 260 vectors. The small segment E is placed in front of a big segment.

4 Streaming

4.1 Introduction / 145

4.2 Streaming Implementation Using Dynamic Modes / 145

This chapter describes the streaming capabilities of the M8195A.

4.1 Introduction

The streaming feature of the M8195A allows re-loading the sample memory while being in the run mode. This capability provides a method to generate waveforms with an infinite playtime. Streaming is supported by the Dynamic Mode.

4.2 Streaming Implementation Using Dynamic Modes

The dynamic modes (refer to the section 3.9) allow switching between segments (Arbitrary Mode) or sequences (Sequence Mode) using the external dynamic input port (M8197A Module required) or by the software. A continuous or triggered execution is possible.

It is possible to modify the content of the sample memory when having selected one of the dynamic modes. Therefore, all segments or sequences that are currently not in use can be changed in run mode. Dynamic modifications of sequence table entries is also possible. This type of streaming implementation requires Dynamic Sequencing.

The following rules apply for implementing streaming using dynamic modes:

The sample data and sequence vector data can be changed in run mode.

Changing the content of segments or sequences, which are currently executed or which are already selected by the dynamic port or by software to be executed next, is not allowed.

NOTE

The hardware or software is not able to check this limitation. Obeying this rule is the responsibility of the user. In order to meet this rule, the user can query the segment number that is currently played by the M8195A.

The dynamic modes have some limitations. The main problem for streaming applications is the fact that a pre-defined timing relationship is not always guaranteed when switching from one sequence to another sequence. Therefore, especially in the continuous modes, it might happen that the current sequence is played one or more times before switching to the next sequence. This means the exact number of repetitions of a certain sequence cannot be determined. I.e. streaming implementation using dynamic modes is not entirely deterministic.

5 Markers

5.1 Introduction / 147

5.2 Dealing with Markers / 147

5.1 Introduction

The instrument provides output signals with a defined timing relationship to the output sample stream. These signals are called markers.

There are up to 2 marker channels available:

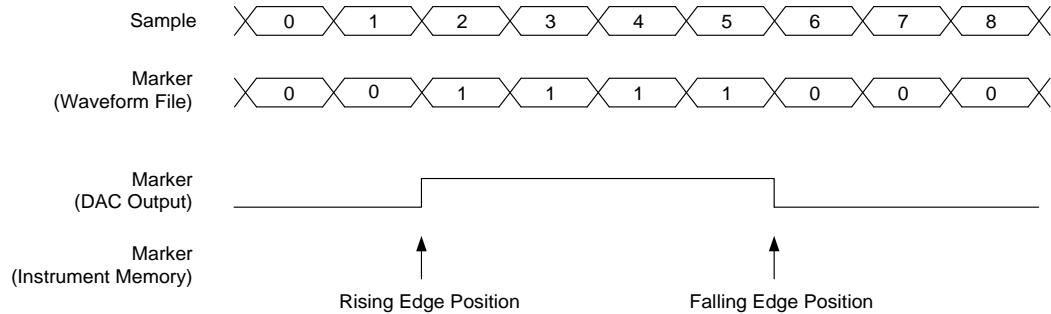
The details of these markers are explained in the sections that follow.

5.2 Dealing with Markers

Depending on the data format, the input files for waveforms provide marker information directly related to the samples. This means that each sample has its own bit signaling whether this sample is marked or not.

The M8195A marker logic uses an edge based concept and therefore the sample based marker waveform is converted to an edge based marker waveform by the firmware of the instrument.

The following picture illustrates this more in detail:

**Figure 58: Marker information**

The upper 3 rows show the sample data together with the corresponding marker bits and the expected output waveform of a marker channel.

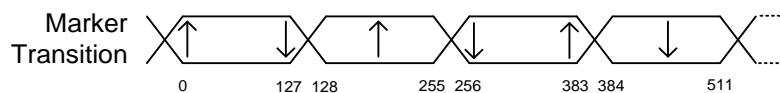
The rising and the falling edge positions represent the information written into the marker memory of the instrument.

5.2.1 Limitations

5.2.1.1 Marker Transition Density

The minimum distance between two rising or two falling edges of markers is 128 DAC samples.

The following drawing shows some examples of rising and falling marker edge positions. The distance is always more than 128 DAC samples.

**Figure 59: Marker transition**

The transition density restriction is related to 128 **DAC** samples independently from the instrument modes. According to the section [Theory of Operation](#), the waveform granularity varies with the Sample Clock Divider. Therefore the transition density restriction varies, too.

Table 20: Marker transition density

| Extended Memory Waveform Memory Access Rate | Marker Transition Density |
|---|--|
| 64 GSa/s | One rising/falling edge within 128 samples |
| 32 GSa/s | One rising/falling edge within 64 samples |
| 16 GSa/s | One rising/falling edge within 32 samples |

5.2.1.2 Markers and Sequencing

The transformation of marked samples into rising and falling edges is individually done for each data segment and the download routine that is responsible for the marker conversion is not able to detect transitions at the beginning or at the end of data segments. This would lead to an unexpected behavior at the boundary of segments in cases where e.g. one segment ends with markers set and a following segment starts with non-marked samples. Therefore, depending on the marker bit of the first sample, the software always places a falling or a rising edge at the beginning of each waveform. This needs to be taken into account by the user, because due to the limitations mentioned in the previous chapter, the corresponding edge can't be placed again within the first 128 samples of a segment. The same problem exists for (big) segments which are divided up into smaller portions that are separately downloaded. In such case each portion is treated individually and rising or falling edges are automatically inserted at the beginning with the already mentioned limitations.

5.2.2 Sample Marker in Segments which are Addressed Offset Based

The instrument provides a mode where sequence table entries address the content of segments by offset (see section [2.15](#)). Whenever a sequence table entry accesses a segment with an offset not equal to zero (not starting from the beginning of the segment), this may result in unexpected sample marker behavior because any marker edges placed at positions not covered by the address offset are ignored. This needs to be taken into account by the user, when setting up markers.

6 General Programming

| | | |
|------|---|-----|
| 6.1 | Introduction / | 152 |
| 6.2 | IVI-COM Programming / | 154 |
| 6.3 | SCPI Programming / | 154 |
| 6.4 | Programming Recommendations / | 157 |
| 6.5 | System Related Commands (SYSTem Subsystem) / | 158 |
| 6.6 | Common Command List / | 165 |
| 6.7 | Status Model / | 168 |
| 6.8 | :ARM/TRIGger Subsystem / | 179 |
| 6.9 | TRIGger - Trigger Input / | 190 |
| 6.10 | :FORMat Subsystem / | 193 |
| 6.11 | :INSTRument Subsystem / | 194 |
| 6.12 | :MMEMory Subsystem / | 199 |
| 6.13 | :OUTPut Subsystem / | 206 |
| 6.14 | Sampling Frequency Commands / | 215 |
| 6.15 | Reference Oscillator Commands / | 216 |
| 6.16 | :VOLTage Subsystem / | 220 |
| 6.17 | [:SOURce]:FUNCTION:MODE ARBITrary STSequence STSCenario / | 223 |
| 6.18 | :STABle Subsystem / | 224 |
| 6.19 | Frequency and Phase Response Data Access / | 235 |
| 6.20 | :TRACe Subsystem / | 236 |
| 6.21 | :TEST Subsystem / | 254 |

6.1 Introduction

Introduction

The M8195A can be programmed like other modular instruments using IVI-COM driver. In addition classic instrument programming using SCPI commands is supported.

The following picture gives an overview about how things work together:

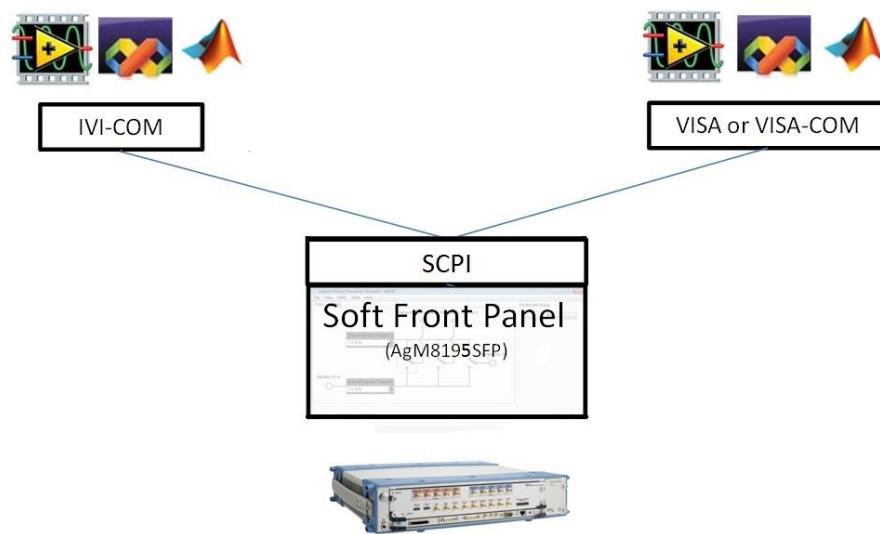


Figure 60: M8195A programming

The Soft Front Panel talks to the actual M8195A module using a PCI express or USB connection. I/O to the module is done using VISA library of Keysight I/O library. Addressing is done with PXI resource strings, e.g. “PXI36::0::0::INSTR” or USB resource strings, e.g. “USB-PXI0::5564::4819::DE00000001::INSTR”. The purpose of the Soft Front Panel is to provide a classic instrument like SCPI interface that is exposed via LAN.

IVI-COM wraps the SCPI commands into an API based programming model. To select what module is programmed, the resource string of the module is used. The IVI-driver will automatically locate an already running Soft Front Panel that is handling the module. If no such Soft Front Panel exists, it is started automatically. This way it is completely hidden that the IVI driver actually needs the Soft Front Panel for programming the M8195A module.

VISA or VISA-COM are libraries from an installed I/O library such as the Keysight I/O library to program the instrument using SCPI command strings. The Soft Front Panel must be already running to connect to it.

The Soft Front Panel is also providing the user interface. It is used for interactively changing settings. In addition, it can log what IVI or SCPI calls need to be done when changing a setting. This can be activated with *Tools → Monitor Driver calls....* In addition, you can verify changes done from a remote program.

6.2 IVI-COM Programming

The recommended way to program the M8195A module is to use the IVI drivers. See documentation of the IVI drivers how to program using IVI drivers. The connection between the IVI-COM driver and the Soft Front Panel is hidden. To address a module therefore the PXI or USB resource string of the module is used. The IVI driver will connect to an already running Soft Front Panel. If the Soft Front Panel is not running, it will automatically start it.

6.3 SCPI Programming

Introduction

In addition to IVI programming SCPI programming using a LAN connection is also supported. Three LAN protocols are supported. The correct resource strings are shown in the Soft Front Panel's About window. A context menu is provided to copy the resource strings.

VXI-11: The Visa resource string is e.g. "TCPIPO::localhost::inst0::INSTR".

HiSLIP: This protocol is recommended. It offers the functionality of VXI-11 protocol with better performance that is near socket performance. Visa resource strings look like "TCPIPO::localhost::hislip0::INSTR". To use the HiSlip protocol an I/O library such as the Keysight I/O Libraries Suite must be installed. Since the protocol is new it might not be supported by the installed I/O library. The Keysight I/O Libraries Suite 16.3 and above supports it. However, the Keysight I/O Libraries Suite might be installed as secondary I/O library. In this case, check if the primary I/O library supports HiSLIP. If it does not, the socket protocol must be used.

Socket: This protocol can be used with any I/O library or using standard operating system socket functionality connecting to port 5025. This protocol must be used if the used I/O library is not supporting HiSLIP protocol. Visa resource string looks like "TCPIPO::localhost::5025::SOCKET", the exact resource string can be seen in the Ag8195 Soft Front Panel main window.

NOTE

AgM8195SFP.exe must be started prior to sending SCPI to the instrument. (See [AgM8195SFP.exe](#))

6.3.1 AgM8195SFP.exe

Before sending SCPI commands to the instrument, the Soft Front Panel (AgM8195SFP.exe) must be started. This can be done in the Windows Start menu (**Start > All Programs > Keysight M8195 > Keysight M8195 Soft Front Panel**).

6.3.1.1 Command Line Arguments

(See [Communication](#) for details about /Socket, /Telnet, /Inst, /HiSLIP, /AutoID, /NoAutoID, /FallBack).

Table 21: Command line arguments

| Option | Description |
|------------------------|--|
| /Socket socketPort | Set the socket port at which the Soft Front Panel waits for SCPI commands |
| /Telnet telnetPort | Set the telnet port at which the Soft Front Panel waits for SCPI commands |
| /Inst instrumentNumber | Set the instrument number (instN, hislipN) at which the Soft Front Panel waits for SCPI commands on VXI-11.3 and HiSLIP connections (if not specified with /HiSLIP). |
| /HiSLIP hislipNumber | Set the instrument number for HiSLIP SCPI communication. If not specified, the same number as for VXI-11.3 is used. |
| /AutoID | Automatically select ports and numbers for the connections (default behavior). |
| /NoAutoID | Disable the default behavior; i.e. do not automatically select ports and numbers for the connections. |
| /FallBack | Try to find unused ports and number if starting a server fails. |
| /NoSplash | Don't show the splash screen. |
| /Minimized | Start with the SFP window minimized to the Windows task bar. |
| /Title "title" | Additional information shown in the SFP window title. |
| /OutputDir | Set the output directory for the log file and temporary files. |
| /r resourceName | Visa PXI resource string of the module to connect to, e.g. PXI12::0::INSTR. "auto" selects the next free instrument. |

6.3.1.2 Communication

Depending on the command line arguments /Socket, /Telnet, /Inst, /AutoID, /NoAutoID, /FallBack, the Soft Front Panel starts several servers to handle SCPI commands. (Refer to the table above.)

/Socket, /Telnet, /Inst, /HiSLIP: If -1, don't start the respective servers

Defaults:

- Socket port: 5025 (e.g. TCPIP0::localhost::5025::SOCKET)
- Telnet port: 5024
- HiSLIP: 0 (e.g. TCPIP0::localhost::hiSLIP0::INSTR)
- VXI-11.3: 0 (e.g. TCPIP0::localhost::inst0::INSTR)

/FallBack: If starting a server fails because of a conflict, try using another port or number

- **HiSLIP, VXI-11.3:** increase the index until a server can be started successfully
- **Socket, Telnet:** start with port 60000, then increase it until the servers can be started successfully. If neither socket nor telnet is disabled the Soft Front Panel tries to start the servers on two consecutive ports (socket port = telnet port + 1)

/AutoID: Automatically select ports and number for the connections, which are unique per instrument.

This is the default behavior; it is not necessary to specify this argument on the command line.

If only one AXle module is connected to this PC and it is an M8195 module, first try to use the command line arguments /Socket, /Telnet, /Inst, or their respective default values if they are not specified. If starting the servers fails, proceed with the steps below.

/Socket, /Telnet, /Inst, /HiSLIP are ignored (unless they are -1 and a server is disabled)

If the Soft Front Panel detects more than one AXle module, use a special mechanism to obtain a number for the HiSLIP and VXI-11.3 servers, which makes sure that the Soft Front Panel uses always the same VISA resource string per module

The socket and telnet port are then calculated from the HiSLIP index:

- telnet port = $60000 + 2 * \langle\text{HiSLIP index}\rangle$
- socket port = $60000 + 2 * \langle\text{HiSLIP index}\rangle + 1$

Note: Ports may already be in use by Windows or other applications, so they are not available for M8195A.

/NoAutoID: Do not automatically select ports and number for the connections, use the values specified with /Socket, /Telnet, /Inst, /HiSLIP or their respective default values instead.

If both /NoAutoID and /AutoID are specified, /AutoID overrides /NoAutoID.

NOTE

The first port not assigned by IANA is 49152 (IANA, Internet Assigned Numbers Authority, <http://www.iana.org>)

6.4 Programming Recommendations

This section lists some recommendations for programming the instrument. Start programming from the default setting. The common command for setting the default setting is:

***RST**

Use the binary data format when transferring waveform data.

The SCPI standard defines a long and a short form of the commands. For fast programming speed, it is recommended to use the short forms. The short forms of the commands are represented by upper case letters. For example the short form of the command to set 10mV offset is:

:VOLT:OFFS 0.01

To improve programming speed it is also allowed to skip optional subsystem command parts. Optional subsystem command parts are depicted in square brackets, e.g.: Set amplitude [:SOURce]:VOLTage[1|2] [:LEVel] [:IMMEDIATE] [:AMPLitude]

Sufficient to use:

:VOLT

M8195A is a 4 channel instrument. Parameters have to be specified for output 1, 2, 3, and 4. If there is no output specified the command will set the default output 1. So, for setting an offset of 10mV for output 1 and output 2 the commands are:

```
:VOLT:OFFS 0.01      # sets offset of 10mV at output 1
:VOLT1:OFFS 0.01    # sets offset of 10mV at output 1
:VOLT2:OFFS 0.01    # sets offset of 10mV at output 2
```

If it is important to know whether the last command is completed then send the common query:

***OPC?**

It is recommended to test the new setting which will be programmed on the instrument by setting it up manually. When you have found the correct setting, then use this to create the program.

In the program it is recommended to send the command for starting data generation (:INIT:IMM) as the last command. This way intermediate stop/restarts (e.g. when changing sample rate or loading a waveform) are avoided and optimum execution performance is achieved.

```
*RST          # set default settings
...           # other commands to set modes
...           # and parameters
:OUTP1 ON    # enable the output 1
:INIT:IMM    # start data generation.
```

6.5 System Related Commands (SYSTem Subsystem)

6.5.1 :SYSTem:EIN:MODE[?] EIN|TOUT

| | |
|-------------------------|--|
| Command | :SYST:EIN:MODE[?] |
| Long | :SYSTem:EIN:MODE[?] |
| Parameters | EIN TOUT |
| Parameter Suffix | None |
| Description | <p>The Event In and Trigger Out functionality use a shared connector on the front panel. This command switches between trigger output and event input functionality. When Trigger Out functionality is active, Event In functionality is disabled and vice versa.</p> <p>Note: Trigger Out is for future use. There are no plans to support Trigger Out functionality directly from M8195A firmware. Trigger Out is tentatively supported by 81195A optical modulation generator software (V2.1 or later).</p> |
| Example | <p>Command :SYST:EIN:MODE TOUT</p> |

6.5.2 :SYSTem:ERRor[:NEXT]?

| | |
|-------------------------|----------------|
| Command | :SYST:ERR? |
| Long | :SYSTem:ERRor? |
| Parameters | None |
| Parameter Suffix | None |

| | |
|--------------------|---|
| Description | <p>Read and clear one error from the instrument's error queue.</p> <p>A record of up to 30 command syntax or hardware errors can be stored in the error queue. Errors are retrieved in first-in-first-out (FIFO) order. The first error returned is the first error that was stored. Errors are cleared as you read them.</p> <p>If more than 30 errors have occurred, the last error stored in the queue (the most recent error) is replaced with "Queue overflow". No additional errors are stored until you remove errors from the queue.</p> <p>If no errors have occurred when you read the error queue, the instrument responds with 0, "No error".</p> <p>The error queue is cleared by the *CLS command, when the power is cycled, or when the Soft Front Panel is re-started.</p> <p>The error queue is not cleared by a reset (*RST) command.</p> <p>The error messages have the following format (the error string may contain up to 255 characters):</p> <p>error number,"Description", e.g. -113,"Undefined header".</p> |
| Example | <p>Query :SYST:ERR?</p> |

6.5.3 :SYSTem:HELP:HEADers?

| | |
|-------------------------|---|
| Command | :SYST:HELP:HEAD? |
| Long | :SYSTem:HELP:HEADers? |
| Parameters | None |
| Parameter Suffix | None |
| Description | <p>The HEADers? query returns all SCPI commands and queries and IEEE 488.2 common commands and common queries implemented by the instrument. The response is a <DEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA> element. The full path for every command and query is returned separated by linefeeds. The syntax of the response is defined as: The <nonzero digit> and sequence of <digit> follow the rules in IEEE 488.2, Section 8.7.9. A <SCPI header> is defined as: It contains all the nodes from the root. The <SCPI program mnemonic> contains the node in standard SCPI format. The short form uses uppercase characters while the additional characters for the long form are in lowercase characters. Default nodes are surrounded by square brackets ([]).</p> |
| Example | <p>Query :SYST:HELP:HEAD?</p> |

6.5.4 :SYSTem:LICense:EXTended:LIST?

| | |
|-------------------------|--|
| Command | :SYST:LIC:EXT:LIST? |
| Long | :SYSTem:LICense:EXTended:LIST? |
| Parameters | None |
| Parameter Suffix | None |
| Description | This query lists the licenses installed. |
| Example | Query :SYST:LIC:EXT:LIST? |

6.5.5 :SYSTem:SET[?]

| | |
|-------------------------|---|
| Command | :SYST:SET[?] |
| Long | :SYSTem:SET[?] |
| Parameters | <binary block data> |
| Parameter Suffix | None |
| Description | In query form, the command reads a block of data containing the instrument's complete set-up. The set-up information includes all parameter and mode settings, but does not include the contents of the instrument setting memories or the status group registers. The data is in a binary format, not ASCII, and cannot be edited. In set form, the block data must be a complete instrument set-up read using the query form of the command. This command has the same functionality as the *LRN command. |
| Example | Command :SYST:SET <binary block data> Query :SYST:SET? |

6.5.6 :SYSTem:VERSion?

| | |
|-------------------------|--|
| Command | :SYST:VERS? |
| Long | :SYSTem:VERSION? |
| Parameters | None |
| Parameter Suffix | None |
| Description | This query returns a formatted numeric value corresponding to the SCPI version number for which the instrument complies. |
| Example | Query :SYST:VERS? |

6.5.7 :SYSTem:COMMunicate:*

| | |
|-------------------------|---|
| Command | :SYST:COMM:*? |
| Long | :SYSTem:COMMunicate:* |
| Parameters | None |
| Parameter Suffix | None |
| Description | <p>These queries return information about the instrument Soft Front Panel's available connections. If a connection is not available, the returned value is -1.</p> <p>This is only useful if there is more than one Keysight module connected to a PC, otherwise one would normally use the default connections (HiSLIP and VXI-11 instrument number 0, socket port 5025, telnet port 5024)</p> <p>One can never be sure if a socket port is already in use, so one could e.g. specify a HiSLIP number on the command line (<code>AgM8195SFP.exe /AutoID /Inst5 /FallBack /r ...</code>) and let the Soft Front Panel find an unused socket port. Then this socket port can be queried using the HiSLIP connection.</p> |
| Example | Query :SYST:COMM:* |

6.5.7.1 :SYSTem:COMMunicate:INSTR[:NUMBER]?

| | |
|-------------------------|---|
| Command | :SYST:COMM:INST? |
| Long | :SYSTem:COMMunicate:INSTR? |
| Parameters | None |
| Parameter Suffix | None |
| Description | This query returns the VXI-11 instrument number used by the Soft Front Panel. |
| Example | Query :SYST:COMM:INST? |

6.5.7.2 :SYSTem:COMMunicate:HISLip[:NUMBER]?

| | |
|-------------------------|--|
| Command | :SYST:COMM:HISL? |
| Long | :SYSTem:COMMunicate:HISLip? |
| Parameters | None |
| Parameter Suffix | None |
| Description | This query returns the HiSLIP number used by the Soft Front Panel. |
| Example | Query :SYST:COMM:HISL? |

6.5.7.3 :SYSTem:COMMunicate:SOCKet[:PORT]?

| | |
|-------------------------|--|
| Command | :SYST:COMM:SOCK? |
| Long | :SYSTem:COMMunicate:SOCKet? |
| Parameters | None |
| Parameter Suffix | None |
| Description | This query returns the socket port used by the Soft Front Panel. |
| Example | Query :SYST:COMM:SOCK? |

6.5.7.4 :SYSTem:COMMunicate:TELNet[:PORT]?

| | |
|-------------------------|--|
| Command | :SYST:COMM:TELN? |
| Long | :SYSTem:COMMunicate:TELNet? |
| Parameters | None |
| Parameter Suffix | None |
| Description | This query returns the telnet port used by the Soft Front Panel. |
| Example | Query :SYST:COMM:TELN? |

6.5.7.5 :SYSTem:COMMUnicatE:TCPip:CONTrol?

| | |
|-------------------------|---|
| Command | :SYST:COMM:TCP:CONT? |
| Long | :SYSTem:COMMUnicatE:TCPip:CONTrol? |
| Parameters | None |
| Parameter Suffix | None |
| Description | This query returns the port number of the control connection. You can use the control port to send control commands (for example “Device Clear”) to the instrument. |
| Example | Query :SYST:COMM:TCP:CONT? |

6.6 Common Command List

6.6.1 *IDN?

Read the instrument's identification string which contains four fields separated by commas. The first field is the manufacturer's name, the second field is the model number, the third field is the serial number, and the fourth field is a revision code which contains four numbers separated dots and a fifth number separated by a dash:

Keysight Technologies, M8195A, <serial number>, x.x.x.x-h

x.x.x.x= Soft Front Panel revision number, e.g. 2.0.0.0

h= Hardware revision number

6.6.2 *CLS

Clear the event register in all register groups. This command also clears the error queue and cancels a *OPC operation. It doesn't clear the enable register.

6.6.3 *ESE

Enable bits in the Standard Event Status Register to be reported in the Status Byte. The selected bits are summarized in the "Standard Event" bit (bit 5) of the Status Byte Register. The *ESE? query returns a value which corresponds to the binary-weighted sum of all bits enabled decimal by the *ESE command. These bits are not cleared by a *CLS command. Value Range: 0–255.

6.6.4 ESR?

Query the Standard Event Status Register. Once a bit is set, it remains set until cleared by a *CLS (clear status) command or queried by this command. A query of this register returns a decimal value which corresponds to the binary-weighted sum of all bits set in the register.

6.6.5 *OPC

Set the "Operation Complete" bit (bit 0) in the Standard Event register after the previous commands have been completed.

6.6.6 *OPC?

Return “1” to the output buffer after the previous commands have been completed. Other commands cannot be executed until this command completes.

6.6.7 *OPT?

Read the installed options. The response consists of any number of fields separated by commas.

6.6.8 *RST

Reset instrument to its factory default state.

6.6.9 *SRE[?]

Enable bits in the Status Byte to generate a Service Request. To enable specific bits, you must write a decimal value which corresponds to the binary-weighted sum of the bits in the register. The selected bits are summarized in the “Master Summary” bit (bit 6) of the Status Byte Register. If any of the selected bits change from “0” to “1”, a Service Request signal is generated. The *SRE? query returns a decimal value which corresponds to the binary-weighted sum of all bits enabled by the *SRE command.

6.6.10 *STB?

Query the summary (status byte condition) register in this register group. This command is similar to a Serial Poll but it is processed like any other instrument command. This command returns the same result as a Serial Poll but the “Master Summary” bit (bit 6) is not cleared by the *STB? command.

6.6.11 *TST?

Execute Self Tests. If self-tests pass, a 0 is returned. A number larger than 0 indicates the number of failed tests.

To get actual messages, use :TEST:TST?

6.6.12 *LRN?

Query the instrument and return a binary block of data containing the current settings (learn string). You can then send the string back to the instrument to restore this state later. For proper operation, do not modify the returned string before sending it to the instrument. Use :SYST:SET to send the learn string. See :SYST:SET[?].

6.6.13 *WAI?

Prevents the instrument from executing any further commands until the current command has finished executing.

6.7 Status Model

Introduction

This section describes the structure of the SCPI status system used by the M8195A. The status system records various conditions and states of the instrument in several register groups as shown on the following pages. Each of the register groups is made up of several low level registers called Condition registers, Event registers, and Enable registers which control the action of specific bits within the register group.

These groups are explained below:

A condition register continuously monitors the state of the instrument. The bits in the condition register are updated in real time and the bits are not latched or buffered. This is a read-only register and bits are not cleared when you read the register. A query of a condition register returns a decimal value which corresponds to the binary-weighted sum of all bits set in that register.

An event register latches the various events from changes in the condition register. There is no buffering in this register; while an event bit is set, subsequent events corresponding to that bit are ignored. This is a read only register. Once a bit is set, it remains set until cleared by query command (such as STAT:QUES:EVEN?) or a *CLS (clear status) command. A query of this register returns a decimal value which corresponds to the binary-weighted sum of all bits set in that register.

An enable register defines which bits in the event register will be reported to the Status Byte register group. You can write to or read from an enable register. A *CLS (clear status) command will not clear the enable register but it does clear all bits in the event register. A STAT:PRES command clears all bits in the enable register. To enable bits in the enable register to be reported to the Status Byte register, you must write a decimal value which corresponds to the binary weighted sum of the corresponding bits.

Transition Filters are used to detect changes of the state in the condition register and set the corresponding bit in the event register. You can set transition filter bits to detect positive transitions (PTR), negative transitions (NTR) or both. Transition filters are read/write registers. They are not affected by *CLS.

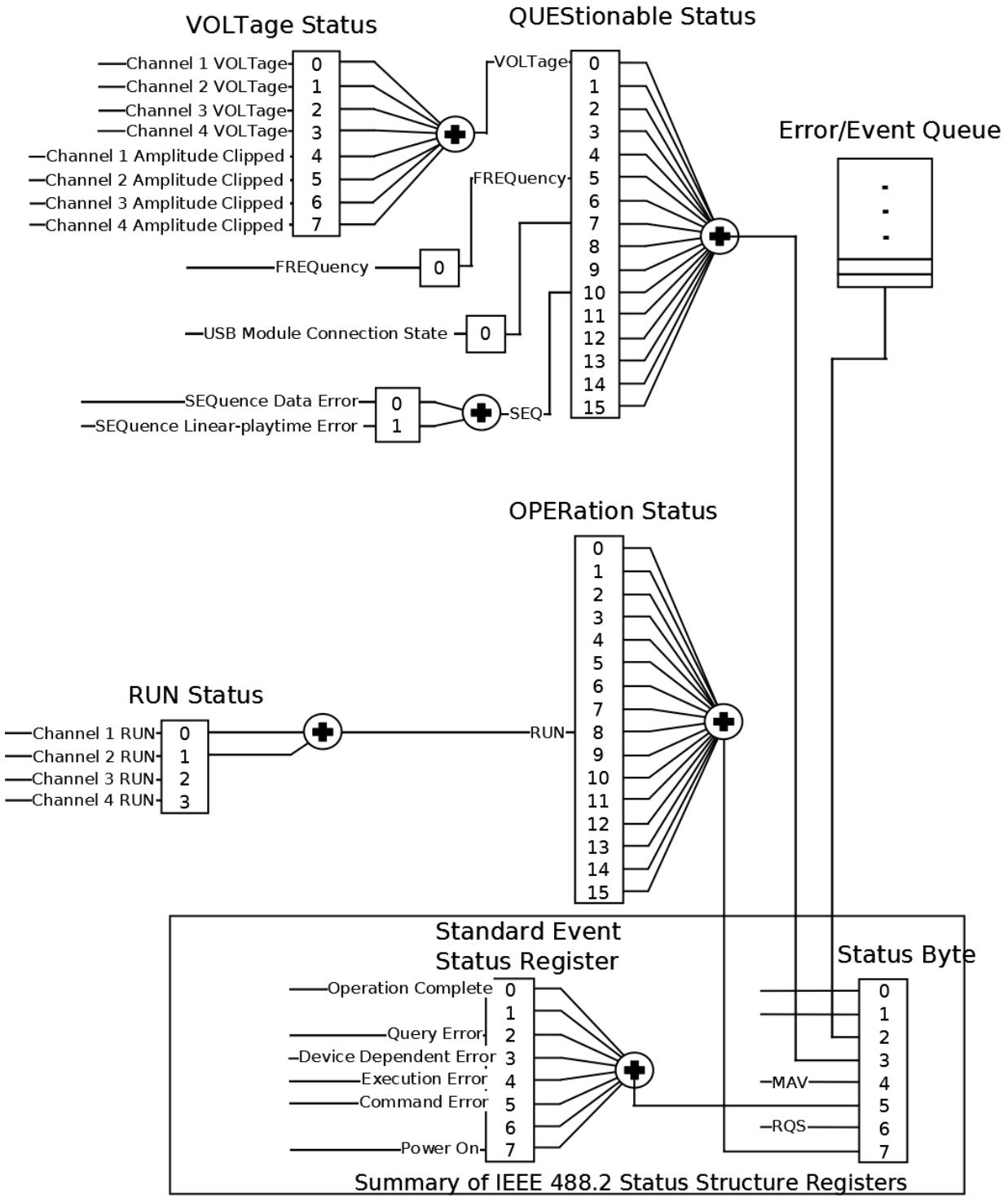


Figure 61: Status register structure

6.7.1 :STATus:PRESet

Clears all status group event registers. Presets the status group enables PTR and NTR registers as follows:

ENABLE = 0x0000, PTR = 0xffff, NTR = 0x0000

6.7.2 Status Byte Register

The Status Byte summary register reports conditions from the other status registers. Data that is waiting in the instrument's output buffer is immediately reported on the "Message Available" bit (bit 4) for example. Clearing an event register from one of the other register groups will clear the corresponding bits in the Status Byte condition register. Reading all messages from the output buffer, including any pending queries, will clear the "Message Available" bit. To set the enable register mask and generate an SRQ (service request), you must write a decimal value to the register using the *SRE command.

Table 22: Status byte register

| Bit Number | Decimal Value | Definition |
|-------------------|---------------|---|
| 0 | 1 | Not Used. Returns "0" |
| Not used | | |
| 1 | 2 | Not Used. Returns "0" |
| Not used | | |
| 2 | 4 | One or more error are stored in the Error Queue |
| Error Queue | | |
| 3 | 8 | One or more bits are set in the Questionable Data Register (bits must be enabled) |
| Questionable Data | | |
| 4 | 16 | Data is available in the instrument's output buffer |
| Message Available | | |
| 5 | 32 | One or more bits are set in the Standard Event Register |
| Standard Event | | |
| 6 | 64 | One or more bits are set in the Status Byte Register |
| Master Summary | | |
| 7 | 128 | One or more bits set in the Operation Data Register (bits must be enabled) |
| Operational Data | | |

6.7.3 Questionable Data Register Command Subsystem

The Questionable Data register group provides information about the quality or integrity of the instrument. Any or all of these conditions can be reported to the Questionable Data summary bit through the enable register.

Table 23: Questionable data register

| Bit Number | Decimal Value | Definition |
|------------|-------------------|---|
| 0 | Voltage warning | 1 Output has been switched off (to protect itself) |
| 1 | Not used | 2 Returns "0" |
| 2 | Not used | 4 Returns "0" |
| 3 | Not used | 8 Returns "0" |
| 4 | Not used | 16 Returns "0" |
| 5 | Frequency warning | 32 Output signal is invalid, because of an instable or missing external reference clock. |
| 6 | Not used | 64 Returns "0" |
| 7 | USB disconnected | 128 USB module connection state |
| 8 | Not used | 256 Returns "0" |
| 9 | Not used | 512 Returns "0" |
| 10 | Sequence Status | 1024 Sequence generation errors happened |
| 11 | Not used | 2048 Returns "0" |
| 12 | Not used | 4096 Returns "0" |
| 13 | Not used | 8192 Returns "0" |
| 14 | Not used | 16384 Returns "0" |
| 15 | Not used | 32768 Returns "0" |

The following commands access the questionable status group.

6.7.3.1 :STATus:QUEStionable[:EVENT]?

Reads the event register in the questionable status group. It's a read-only register. Once a bit is set, it remains set until cleared by this command or the *CLS command. A query of the register returns a decimal value which corresponds to the binary-weighted sum of all bits set in the register.

6.7.3.2 :STATus:QUEStionable:CONDition?

Reads the condition register in the questionable status group. It's a read-only register and bits are not cleared when you read the register. A query of the register returns a decimal value which corresponds to the binary-weighted sum of all bits set in the register.

6.7.3.3 :STATus:QUEStionable:ENABLE[?]

Sets or queries the enable register in the questionable status group. The selected bits are then reported to the Status Byte. A *CLS will not clear the enable register but it does clear all bits in the event register. To enable bits in the enable register, you must write a decimal value which corresponds to the binary-weighted sum of the bits you wish to enable in the register.

6.7.3.4 :STATus:QUEStionable:NTRansition[?]

Sets or queries the negative-transition register in the questionable status group. A negative transition filter allows an event to be reported when a condition changes from true to false. Setting both positive/negative filters true allows an event to be reported anytime the condition changes. Clearing both filters disable event reporting. The contents of transition filters are unchanged by *CLS and *RST.

6.7.3.5 :STATus:QUEStionable:PTRansition[?]

Set or queries the positive-transition register in the questionable status group. A positive transition filter allows an event to be reported when a condition changes from false to true. Setting both positive/negative filters true allows an event to be reported anytime the condition changes. Clearing both filters disable event reporting. The contents of transition filters are unchanged by *CLS and *RST.

6.7.4 Operation Status Subsystem

The Operation Status register contains conditions which are part of the instrument's normal operation.

Table 24: Operation status register

| Bit Number | | Decimal Value | Definition |
|------------|------------|---------------|--------------------------------|
| 0 | Not used | 1 | Returns "0" |
| 1 | Not used | 2 | Returns "0" |
| 2 | Not used | 4 | Returns "0" |
| 3 | Not used | 8 | Returns "0" |
| 4 | Not used | 16 | Returns "0" |
| 5 | Not used | 32 | Returns "0" |
| 6 | Not used | 64 | Returns "0" |
| 7 | Not used | 128 | Returns "0" |
| 8 | Run Status | 256 | Indicates if system is running |
| 9 | Not used | 512 | Returns "0" |
| 10 | Not used | 1024 | Returns "0" |
| 11 | Not used | 2048 | Returns "0" |
| 12 | Not used | 4096 | Returns "0" |
| 13 | Not used | 8192 | Returns "0" |
| 14 | Not used | 16384 | Returns "0" |
| 15 | Not used | 32768 | Returns "0" |

The following commands access the operation status group.

6.7.4.1 :STATus:OPERation[:EVENT]?

Reads the event register in the operation status group. It's a read-only register. Once a bit is set, it remains set until cleared by this command or *CLS command. A query of the register returns a decimal value which corresponds to the binary-weighted sum of all bits set in the register.

6.7.4.2 :STATus:OPERation:CONDition?

Reads the condition register in the operation status group. It's a read-only register and bits are not cleared when you read the register. A query of the register returns a decimal value which corresponds to the binary-weighted sum of all bits set in the register.

6.7.4.3 :STATus:OPERation:ENABLE[?]

Sets or queries the enable register in the operation status group. The selected bits are then reported to the Status Byte. A *CLS will not clear the enable register but it does clear all bits in the event register. To enable bits in the enable register, you must write a decimal value which corresponds to the binary-weighted sum of the bits you wish to enable in the register.

6.7.4.4 :STATus:OPERation:NTRansition[?]

Sets or queries the negative-transition register in the operation status group. A negative transition filter allows an event to be reported when a condition changes from true to false. Setting both positive/negative filters true allows an event to be reported anytime the condition changes. Clearing both filters disable event reporting. The contents of transition filters are unchanged by *CLS and *RST.

6.7.4.5 :STATus:OPERation:PTRansition[?]

Set or queries the positive-transition register in the operation status group. A positive transition filter allows an event to be reported when a condition changes from false to true. Setting both positive/negative filters true allows an event to be reported anytime the condition changes. Clearing both filters disable event reporting. The contents of transition filters are unchanged by *CLS and *RST.

6.7.5 Voltage Status Subsystem

The Voltage Status register contains the voltage conditions of the individual channels.

The following SCPI commands and queries are supported:

```
:STATus:QUEstionable:VOLTage[:EVENT]?
:STATus:QUEstionable:VOLTage:CONDition?
:STATus:QUEstionable:VOLTage:ENABLE[?]
:STATus:QUEstionable:VOLTage:NTRansition[?]
:STATus:QUEstionable:VOLTage:PTRansition[?]
```

Table 25: Voltage status register

| Bit Number | | Decimal Value | Definition |
|------------|-------------------|---------------|---|
| 0 | Voltage warning | 1 | Output 1 has been switched off (to protect itself) |
| 1 | Voltage warning | 2 | Output 2 has been switched off (to protect itself) |
| 2 | Voltage warning | 4 | Output 3 has been switched off (to protect itself) |
| 3 | Voltage warning | 8 | Output 4 has been switched off (to protect itself) |
| 4 | Amplitude clipped | 16 | The amplitude for output 1 has been clipped to the highest/lowest possible DAC value. |
| 5 | Amplitude clipped | 32 | The amplitude for output 2 has been clipped to the highest/lowest possible DAC value. |
| 6 | Amplitude clipped | 64 | The amplitude for output 3 has been clipped to the highest/lowest possible DAC value. |
| 7 | Amplitude clipped | 128 | The amplitude for output 4 has been clipped to the highest/lowest possible DAC value. |

6.7.6 Frequency Status Subsystem

The Frequency Status register contains the frequency conditions of the module.

The following SCPI commands and queries are supported:

```
:STATUs:QUESTIONable:FREQuency[:EVENT]?
:STATUs:QUESTIONable:FREQuency:CONDITION?
:STATUs:QUESTIONable:FREQuency:ENABLE[?]
:STATUs:QUESTIONable:FREQuency:NTRansition[?]
:STATUs:QUESTIONable:FREQuency:PTRansition[?]
```

Table 26: Frequency status register

| Bit Number | Decimal Value | Definition |
|---------------------|---------------|---|
| 0 Frequency warning | 1 | Output signal is invalid, because of an instable or missing external reference clock. |

6.7.7 Sequence Status Subsystem

The Sequence Status register is used to indicate errors in the sequence table data provided by the user.

The following SCPI commands and queries are supported:

```
:STATUs:QUESTIONable:SEQUence[:EVENT]?
:STATUs:QUESTIONable:SEQUence:CONDITION?
:STATUs:QUESTIONable:SEQUence:ENABLE[?]
:STATUs:QUESTIONable:SEQUence:NTRansition[?]
:STATUs:QUESTIONable:SEQUence:PTRansition[?]
```

Table 27: Sequence status register

| Bit Number | Decimal Value | Definition |
|----------------------------------|---------------|--------------------------------------|
| 0 Sequence data error | 1 | Sequence has errors |
| 1 Sequence linear-playtime error | 2 | Sequence has a linear-playtime error |

6.7.8 Connection Status Subsystem

The Connection Status register contains the state of the USB connection to the M8195A module.

The following SCPI commands and queries are supported:

```
:STATus:QUEstionable:CONNnection[:EVENT]?
:STATus:QUEstionable:CONNnection:CONDition?
:STATus:QUEstionable:CONNnection:ENABLE[?]
:STATus:QUEstionable:CONNnection:NTRansition[?]
:STATus:QUEstionable:CONNnection:PTRansition[?]
```

Table 28: Connection status register

| Bit Number | Decimal Value | Definition | |
|------------|------------------|------------|-----------------------------|
| 0 | USB disconnected | 1 | USB module connection state |

6.7.9 Run Status Subsystem

The Run Status register contains the run status conditions of the individual channels.

The following SCPI commands and queries are supported:

```
:STATUs:OPERation:RUN[:EVENT]?
:STATUs:OPERation:RUN:CONDition?
:STATUs:OPERation:RUN:ENABLE[?]
:STATUs:OPERation:RUN:NTRansition[?]
:STATUs:OPERation:RUN:PTRansition[?]
```

Table 29: Run status register

| Bit Number | Decimal Value | Definition | |
|------------|---------------|------------|-----------------------------------|
| 0 | Run Status | 1 | Indicates if channel 1 is running |
| 1 | Run Status | 2 | Indicates if channel 2 is running |
| 2 | Run Status | 4 | Indicates if channel 3 is running |
| 3 | Run Status | 8 | Indicates if channel 4 is running |

6.8 :ARM/TRIGger Subsystem

6.8.1 :ABORT[1|2|3|4]

| | |
|-------------------------|--|
| Command | :ABOR [1 2 3 4] |
| Long | :ABORT [1 2 3 4] |
| Parameters | None |
| Parameter Suffix | None |
| Description | Stop signal generation on all channels. The channel suffix is ignored. |
| Example | Command :ABOR1 |

6.8.2 :ARM[:SEQUence][:START][:LAYer]:MDELay[?] <module_delay>|MINimum|MAXimum

| | |
|-------------------------|--|
| Command | :ARM:MDEL [?] |
| Long | :ARM[: SEQuence] [START] [: LAYer] :MDELay[?] |
| Parameters | {<delay> MINimum MAXimum} |
| Parameter Suffix | [s ms us ns ps] |
| Description | Set or query the module delay settings (see section 1.5.3) . The unit is in seconds. |
| Example | Command :ARM:MDEL 1E-13 |
| | Query :ARM:MDEL? |

6.8.3 ARM[:SEQUence][:STARt][:LAYer]:SDELay[1|2|3|4][?] <delay>|MINimum|MAXimum

| | |
|-------------------------|--|
| Command | :ARM:SDEL[?] |
| Long | :ARM[:SEQUence] [STARt] [:LAYer]:SDELay[?] |
| Parameters | {<delay> MINimum MAXimum} |
| Parameter Suffix | None |
| Description | Set or query the channel-specific sample delay in integral DAC sample clock periods. The range is 0..95 |
| Example | Command :ARM:SDEL 10 |
| | Query :ARM:SDEL? |

6.8.4 :INITiate:CONTinuous:ENABLE[?] SELF|ARMed

| | |
|-------------------------|---------------------------------|
| Command | :INIT:CONT:ENAB[?] |
| Long | :INITiate:CONTinuous:ENABLE[?] |
| Parameters | SELF ARMed |
| Parameter Suffix | None |
| Description | Set or query the arming mode. |
| Example | Command :INIT:CONT:ENAB SELF |
| | Query :INIT:CONT:ENAB? |

6.8.5 :INITiate:CONTinuous[:STATe][?] OFF|ON|0|1

| | |
|-------------------------|---|
| Command | <code>:INIT:CONT:STAT[?]</code> |
| Long | <code>:INITiate:CONTinuous:STATe[?]</code> |
| Parameters | OFF ON 0 1 |
| Parameter Suffix | None |
| Description | <p>Set or query the continuous mode. This command must be used together with INIT:GATE to set the trigger mode.</p> <ul style="list-style-type: none"> • 0/OFF – Continuous mode is off. If gate mode is off, the trigger mode is “triggered”, else it is “gated”. • 1/ON – Continuous mode is on. Trigger mode is “automatic”. The value of gate mode is not relevant. |
| Example | <p>Command <code>:INIT:CONT:STAT ON</code></p> <p>Query <code>:INIT:CONT:STAT?</code></p> |

6.8.6 :INITiate:GATE[:STATe][?] OFF|ON|0|1

| | |
|-------------------------|--|
| Command | :INIT:GATE:STAT[?] |
| Long | :INITiate:GATE:STATe[?] |
| Parameters | OFF ON 0 1 |
| Parameter Suffix | None |
| Description | Set or query the gate mode. This command must be used together with INIT:CONT to set the trigger mode. |
| | <ul style="list-style-type: none"> • 0/OFF – Gate mode is off. • 1/ON – Gate mode is on. If continuous mode is off, the trigger mode is “gated”. |
| Example | <p>Command :INIT:GATE:STAT ON</p> <p>Query :INIT:GATE:STAT?</p> |

Table 30: Trigger mode settings

| INIT:CONT | INIT:GATE | Trigger Mode |
|-----------|-----------|--------------|
| 0 | 0 | Triggered |
| 0 | 1 | Gated |
| 1 | 0 | Continuous |
| 1 | 1 | Continuous |

6.8.7 :INITiate:IMMEDIATE[1|2|3|4]

| | |
|-------------------------|---|
| Command | <code>:INIT:IMM[1 2 3 4]</code> |
| Long | <code>:INITiate:IMMEDIATE[1 2 3 4]</code> |
| Parameters | None |
| Parameter Suffix | None |
| Description | Start signal generation on all channels. The channel suffix is ignored. |
| Example | Command <code>:INIT:IMM</code> |

6.8.8 :ARM[:SEQUence][:STARt][:LAYer]:TRIGger:LEVel[?] <level>|MINimum|MAXimum

| | |
|-------------------------|--|
| Command | <code>:ARM:TRIG:LEV[?]</code> |
| Long | <code>:ARM:TRIGGER:LEVEL[?]</code> |
| Parameters | <code><level> MINimum MAXimum</code> |
| Parameter Suffix | None |
| Description | Set or query the trigger input threshold level. <ul style="list-style-type: none"> • <code><level></code> – Threshold level voltage. |
| Example | Command <code>:ARM:TRIG:LEV 3e-9</code> Query <code>:ARM:TRIG:LEV?</code> |

6.8.9 :ARM[:SEQUence][:STARt][:LAYer]:TRIGger:SLOPe[?] POSitive|NEGative|EITHER

| | |
|-------------------------|--|
| Command | <code>:ARM:TRIG:SLOP[?]</code> |
| Long | <code>:ARM:TRIGger:SLOPe[?]</code> |
| Parameters | POSitive NEGative EITHER |
| Parameter Suffix | None |
| Description | Set or query the trigger input slope. <ul style="list-style-type: none">• POSitive – rising edge• NEGative – falling edge• EITHER – both |
| Example | Command <code>:ARM:TRIG:SLOP POS</code> Query <code>:ARM:TRIG:SLOP?</code> |

6.8.10 :ARM[:SEQuence][:STARt][:LAYer]:TRIGger:SOURce[?] TRIGger|EVENT|INTernal

| | |
|-------------------------|--|
| Command | :ARM:TRIG:SOUR[?] |
| Long | :ARM:TRIGGER:SOURce[?] |
| Parameters | TRIGger EVENT INTernal |
| Parameter Suffix | None |
| Description | <p>Set or query the source for the trigger function.</p> <ul style="list-style-type: none"> • TRIGger – trigger input • EVENT – event input • INTernal – internal trigger generator |
| Example | <p>Command :ARM:TRIG:SOUR TRIG</p> <p>Query :ARM:TRIG:SOUR?</p> |

6.8.11 :ARM[:SEQuence][:STARt][:LAYer]:TRIGger:FREQuency[?] <frequency>|MINimum|MAXimum

| | |
|-------------------------|---|
| Command | :ARM:TRIG:FREQ[?] |
| Long | :ARM:TRIGGER:FREQuency[?] |
| Parameters | <frequency> MINimum MAXimum |
| Parameter Suffix | None |
| Description | <p>Set or query the frequency of the internal trigger generator.</p> <ul style="list-style-type: none"> • <frequency> – internal trigger frequency |
| Example | <p>Command :ARM:TRIG:FREQ 1</p> <p>Query :ARM:TRIG:FREQ?</p> |

6.8.12 :ARM[:SEQUence][:STARt][:LAYer]:TRIGger:OPERation[?] ASYNchronous|SYNChronous

| | |
|-------------------------|--|
| Command | :ARM:TRIG:OPER[?] |
| Long | :ARM:TRIGger:OPERation[?] |
| Parameters | ASYNchronous SYNChronous |
| Parameter Suffix | None |
| Description | <p>Set or query the operation mode for the trigger and event input.</p> <ul style="list-style-type: none"> • ASYNchronous – asynchronous operation (see section 1.5.2) • SYNChronous – synchronous operation (see section 1.5.2) |
| Example | <p>Command :ARM:TRIG:OPER SYNC</p> <p>Query :ARM:TRIG:OPER?</p> |

6.8.13 :ARM[:SEQUence][:STARt][:LAYer]:EVENT:LEVel[?] <level>|MINimum|MAXimum

| | |
|-------------------------|---|
| Command | :ARM:EVEN:LEV[?] |
| Long | :ARM:EVENT:LEVel[?] |
| Parameters | <level> MINimum MAXimum |
| Parameter Suffix | None |
| Description | <p>Set or query the input threshold level.</p> <ul style="list-style-type: none"> • <level> – Threshold level voltage. |
| Example | <p>Command :ARM:EVEN:LEV 2e-9</p> <p>Query :ARM:EVEN:LEV?</p> |

6.8.14 :ARM[:SEQUence][:STARt][:LAYer]:EVENT:SLOPe[?] POSitive|NEGative|EITHer

| | |
|-------------------------|---|
| Command | :ARM:EVEN:SLOP[?] |
| Long | :ARM:EVENT:SLOPe[?] |
| Parameters | POSitive NEGative EITHer |
| Parameter Suffix | None |
| Description | <p>Set or query the event input slope.</p> <ul style="list-style-type: none"> • POSitive – rising edge • NEGative – falling edge • EITHer – both |
| Example | <p>Command :ARM:EVEN:SLOP POS</p> <p>Query :ARM:EVEN:SLOP?</p> |

6.8.15 :TRIGger[:SEQUence][:STARt]:SOURce:ENABLE[?] TRIGger|EVENT

| | |
|-------------------------|---|
| Command | :TRIG:SOUR:ENAB[?] |
| Long | :TRIGger:SOURce:ENABLE[?] |
| Parameters | TRIGger EVENT |
| Parameter Suffix | None |
| Description | <p>Set or query the source for the enable event.</p> <ul style="list-style-type: none"> • TRIGger – trigger input • EVENT – event input |
| Example | <p>Command :TRIG:SOUR:ENAB TRIG</p> <p>Query :TRIG:SOUR:ENAB?</p> |

6.8.16 :TRIGger[:SEQUence][:STARt]:ENABLE:HWDISable[:STATe][?] 0|1|OFF|ON

| | |
|-------------------------|---|
| Command | :TRIG:ENAB:HWD [?] |
| Long | :TRIGger:ENABLE:HWDISable[?] |
| Parameters | 0 1 OFF ON |
| Parameter Suffix | None |
| Description | Set or query the hardware input disable state for the enable function. When the hardware input is disabled, an enable event can only be generated using the :TRIGger[:SEQUence][:STARt]:ENABLE[:IMMEDIATE] command. When the hardware input is enabled, an enable event can be generated by command or by a signal present at the trigger or event input. |
| Example | <p>Command :TRIG:ENAB:HWD ON</p> <p>Query :TRIG:ENAB:HWD?</p> |

6.8.17 :TRIGger[:SEQUence][:STARt]:BEGin:HWDISable[:STATe][?] 0|1|OFF|ON

| | |
|-------------------------|--|
| Command | :TRIG:BEG:HWD [?] |
| Long | :TRIGger:BEGin:HWDISable[?] |
| Parameters | 0 1 OFF ON |
| Parameter Suffix | None |
| Description | Set or query the hardware input disable state for the trigger function. When the hardware input is disabled, a trigger can only be generated using the :TRIGger[:SEQUence][:STARt]:BEGin[:IMMEDIATE] command. When the hardware input is enabled, a trigger can be generated by command, by a signal present at the trigger input or the internal trigger generator. |

Example Command
 :**TRIG:BEG:HWD** ON

Query
 :**TRIG:BEG:HWD?**

6.8.18 :TRIGger[:SEQUence][:STARt]:ADVance:HWDisable[:STATe][?] 0|1|OFF|ON

Command :**TRIG:ADV:HWD** [?]

Long :**TRIGger:ADVance:HWDisable** [?]

Parameters 0|1|OFF|ON

Parameter Suffix None

Description Set or query the hardware input disable state for the advancement function. When the hardware input is disabled, an advancement event can only be generated using the :TRIGger[:SEQUence][:STARt]:ADVance[:IMMEDIATE] command. When the hardware input is enabled, an advancement event can be generated by command or by a signal present at the trigger or event input.

Example Command
 :**TRIG:ADV:HWD** 0

Query
 :**TRIG:ADV:HWD?**

6.9 TRIGger - Trigger Input

6.9.1 :TRIGger[:SEQUence][:STARt]:SOURce:ADVance[?] TRIGger|EVENT|INTernal

| | |
|-------------------------|--|
| Command | :TRIG:SOUR:ADV[?] |
| Long | :TRIGger:SOURce:ADVance[?] |
| Parameters | TRIGger EVENT INTernal |
| Parameter Suffix | None |
| Description | Set or query the source for the advancement event. <ul style="list-style-type: none"> • TRIGger – trigger input • EVENT – event input • INTernal – internal trigger generator |
| Example | Command :TRIG:SOUR:ADV TRIG Query :TRIG:SOUR:ADV? |

6.9.2 :TRIGger[:SEQUence][:STARt]:ENABLE[:IMMediate]

| | |
|-------------------------|-------------------------------------|
| Command | :TRIG:ENAB |
| Long | :TRIGger:ENABLE |
| Parameters | None |
| Parameter Suffix | None |
| Description | Send the enable event to a channel. |
| Example | Command :TRIG:ENAB |

6.9.3 :TRIGger[:SEQUence][:STARt]:BEGin[:IMMEDIATE]

| | |
|-------------------------|--|
| Command | :TRIG:BEG |
| Long | :TRIGger:BEGin |
| Parameters | None |
| Parameter Suffix | None |
| Description | In triggered mode send the start/begin event to a channel. |
| Example | Command :TRIG:BEG |

6.9.4 :TRIGger[:SEQUence][:STARt]:BEGin:GATE[:STATe][?] OFF|ON|0|1

| | |
|-------------------------|---|
| Command | :TRIG:BEG:GATE[?] |
| Long | :TRIGger:BEGin:GATE[?] |
| Parameters | OFF ON 0 1 |
| Parameter Suffix | None |
| Description | In gated mode send a “gate open” (ON 1) or “gate close” (OFF 0) to a channel. |
| Example | Command :TRIG:BEG:GATE ON Query :TRIG:BEG:GATE? |

6.9.5 :TRIGger[:SEQUence][:STARt]:ADVance[:IMMEDIATE]

| | |
|-------------------------|--|
| Command | :TRIG:ADV |
| Long | :TRIGger:ADVance |
| Parameters | None |
| Parameter Suffix | None |
| Description | Send the advancement event to a channel. |
| Example | Command :TRIG:ADV |

6.10 :FORMat Subsystem

6.10.1 :FORMat:BORDer NORMal|SWAPPed

| | | | | | |
|-------------------------|--|---------|-----------------|-------|-------------|
| Command | :FORM:BORD[?] | | | | |
| Long | :FORMat:BORDer[?] | | | | |
| Parameters | NORMal SWAPPed | | | | |
| Parameter Suffix | None | | | | |
| Description | Byte ORDer. Controls whether binary data is transferred in normal (“big endian”) or swapped (“little endian”) byte order. Affects [:SOURce]:STABle:DATA, OUTPut:FILTer:FRATe, OUTPut:FILTer:HRATe and OUTPut:FILTer:QRATe. | | | | |
| Example | <table><tr><td>Command</td></tr><tr><td>:FORM:BORD NORM</td></tr><tr><td>Query</td></tr><tr><td>:FORM:BORD?</td></tr></table> | Command | :FORM:BORD NORM | Query | :FORM:BORD? |
| Command | | | | | |
| :FORM:BORD NORM | | | | | |
| Query | | | | | |
| :FORM:BORD? | | | | | |

6.11 :INSTRument Subsystem

6.11.1 :INSTRument:SLOT[:NUMBER]?

| | |
|-------------------------|---|
| Command | :INST:SLOT? |
| Long | :INSTRument:SLOT? |
| Parameters | None |
| Parameter Suffix | None |
| Description | Query the instrument's slot number in its AXIe frame. |
| Example | Query :INST:SLOT? |

6.11.2 :INSTRument:IDENTify [<seconds>]

| | |
|-------------------------|--|
| Command | :INST:IDEN |
| Long | :INSTRument:IDENTify |
| Parameters | <seconds> |
| Parameter Suffix | None |
| Description | Identify the instrument by flashing the green “Access” LED on the front panel for a certain time. <ul style="list-style-type: none">• <seconds> - optional length of the flashing interval, default is 10 seconds. |
| Example | Command :INST:IDEN 5 |

6.11.3 :INSTRument:IDENTify:STOP

| | |
|-------------------------|---|
| Command | :INST:IDEN:STOP |
| Long | :INSTRument:IDENTify:STOP |
| Parameters | None |
| Parameter Suffix | None |
| Description | Stop the flashing of the green “Access” LED before the flashing interval has elapsed. |
| Example | Command :INST:IDEN:STOP |

6.11.4 :INSTRument: HWRevision?

| | |
|-------------------------|--|
| Command | :INST:HWR? |
| Long | :INSTRument:HWRevision? |
| Parameters | None |
| Parameter Suffix | None |
| Description | Returns the M8195A hardware revision number. |
| Example | Query :INST:HWR? |

6.11.5 :INSTRument:DACMode[?] SINGle|DUAL|FOUR|MARGer|DCDUPLICATE|DCMarker

| | |
|-------------------------|---|
| Command | :INST:DACM[?] |
| Long | :INSTRUMENT:DACMode[?] |
| Parameters | SINGle DUAL FOUR MARGer DCDUPLICATE DCMarker <ul style="list-style-type: none"> • SINGle – Channel 1 can generate a signal • DUAL – Channels 1 and 4 can generate a signal, channels 2 and 3 are unused • FOUR – Channels 1, 2, 3, and 4 can generate a signal • MARGer – Channel 1 with two markers output on channel 3 and 4 • DCDUPLICATE – dual channel duplicate: Channels 1, 2, 3, and 4 can generate a signal. Channel 3 generates the same signal as channel 1. Channel 4 generates the same signal as channel 2. • DCMarker – dual channel with marker: Channels 1 and 2 can generate a signal. Channel 1 has two markers output on channel 3 and 4. Channel 2 can generate signals without markers. |
| Parameter Suffix | None |
| Description | Use this command or query to set or get the operation mode of the DAC. The value of the operation mode determines, to which channels waveforms can be transferred and the format of the waveform data. In operation mode SINGle, DUAL, DCDUPLICATE, or FOUR the data consists of 1-byte waveform samples only. In operation mode MARGer or DCMarker the data loaded to channel 1 consists of interleaved 1-byte waveform and 1-byte marker samples (see section :TRACe Subsystem). In operation mode DDUPlicate waveforms can only be loaded to channels 1 and 2. |
| Example | Command :INST:DACM DUAL |

6.11.6 :INSTRUMENT:MEMORY:EXTENDED:RDIVIDER [?] DIV1|DIV2|DIV4

| | |
|-------------------------|--|
| Command | :INST:MEM:EXT:RDIV[?] |
| Long | :INSTRUMENT:MEMORY:EXTENDED:RDIVIDER[?] |
| Parameters | DIV1 DIV2 DIV4 <ul style="list-style-type: none"> • DIV1 – Memory sample rate is the DAC Sample Rate. • DIV2 – Memory sample rate is the DAC Sample Rate divided by 2. • DIV4 – Memory sample rate is the DAC Sample Rate divided by 4. |
| Parameter Suffix | None |
| Description | Use this command or query to set or get the Sample Rate Divider of the Extended Memory. This value determines also the amount of available Extended Memory for each channel (see section 1.5.5). |
| Example | <p>Command :INST:MEM:EXT:RDIV DIV4</p> <p>Query :INST:MEM:EXT:RDIV?</p> |

6.11.7 :INSTRUMENT:MMODULE:CONF?

| | |
|-------------------------|--|
| Command | :INST:MMOD:CONF? |
| Long | :INSTRUMENT:MMODULE:CONF? |
| Parameters | None |
| Parameter Suffix | None |
| Description | This query returns the state of the multi-module configuration mode (0: disabled, 1: enabled). |
| Example | <p>Query :INST:MMOD:CONF?</p> |

6.11.8 :INSTRUMENT:MMODULE:MODE?

| | |
|-------------------------|--|
| Command | :INST:MMOD:MODE? |
| Long | :INSTRUMENT:MMODULE:MODE? |
| Parameters | None |
| Parameter Suffix | None |
| Description | This query returns the multi-module mode. <ul style="list-style-type: none">• NORMAl – Module does not belong to a multi-module group.• SLAVe – Module is a slave in a multi-module group |
| Example | Query :INST:MMOD:MODE? |

6.12 :MMEMory Subsystem

NOTE

MMEM commands requiring <directory_name> assume the current directory if a relative path or no path is provided. If an absolute path is provided, then it is ignored.

6.12.1 :MMEMory:CATalog? [<directory_name>]

| | |
|-------------------------|---|
| Command | :MMEM:CAT? |
| Long | :MMEMory:CATalog? |
| Parameters | None |
| Parameter Suffix | None |
| Description | <p>Query disk usage information (drive capacity, free space available) and obtain a list of files and directories in a specified directory in the following format:</p> <p><numeric_value>,<numeric_value>,{<file_entry>}</p> <p>This command returns two numeric parameters and as many strings as there are files and directories. The first parameter indicates the total amount of storage currently used in bytes. The second parameter indicates the total amount of storage available, also in bytes. The <file_entry> is a string. Each <file_entry> indicates the name, type, and size of one file in the directory list:</p> <p><file_name>,<file_type>,<file_size></p> <p>As the Windows file system has an extension that indicates file type, <file_type> is always empty. <file_size> provides the size of the file in bytes. In case of directories, <file_entry> is surrounded by square brackets and both <file_type> and <file_size> are empty.</p> |
| Example | <p>Query</p> <p>:MMEM:CAT?</p> |

6.12.2 :MMEMory:CDIRectory [<directory_name>]

| | |
|-------------------------|--|
| Command | :MMEM:CDIR |
| Long | :MMEMory:CDIRectory |
| Parameters | None |
| Parameter Suffix | None |
| Description | Changes the default directory for a mass memory file system. The <directory_name> parameter is a string. If no parameter is specified, the directory is set to the *RST value. At *RST, this value is set to the default user data storage area, that is defined as System.Environment.SpecialFolder.Personal e.g. C:\Users\Name\Documents MMEMory:CDIRectory? – Query returns full path of the default directory. |
| Example | <p>Command :MMEM:CDIR "C:\Users\Name\Documents"</p> <p>Query :MMEM:CDIR?</p> |

6.12.3 :MMEMory:COPY <string>,<string>[,<string>,<string>]

| | |
|-------------------------|--|
| Command | :MMEM:COPY |
| Long | :MMEMory:COPY |
| Parameters | <string>,<string> |
| Parameter Suffix | None |
| Description | <p>Copies an existing file to a new file or an existing directory to a new directory. Two forms of parameters are allowed. The first form has two parameters. In this form, the first parameter specifies the source, and the second parameter specifies the destination.</p> <p>The second form has four parameters. In this form, the first and third parameters specify the file names. The second and fourth parameters specify the directories. The first pair of parameters specifies the source. The second pair specifies the destination. An error is generated if the source doesn't exist or the destination file already exists.</p> |
| Example | <p>Command :MMEM:COPY "C:\data.txt", "C:\data_new.txt"</p> |

6.12.4 :MMEMory:DELetE <file_name>[,<directory_name>]

| | |
|-------------------------|--|
| Command | :MMEM:DEL |
| Long | :MMEMory:DELetE |
| Parameters | <file_name> |
| Parameter Suffix | None |
| Description | Removes a file from the specified directory. The <file_name> parameter specifies the file to be removed. |
| Example | Command :MMEM:DEL "C:\data.txt" |

6.12.5 :MMEMory:DATA <file_name>, <data>

| | |
|-------------------------|--|
| Command | :MMEM:DATA |
| Long | :MMEMory:DATA |
| Parameters | <file_name>, <data> |
| Parameter Suffix | None |
| Description | The command form is MMEMory:DATA <file_name>,<data>. It loads <data> into the file <file_name>. <data> is in 488.2 block format. <file_name> is string data. |
| Example | Command :MMEM:DATA "C:\data.txt", #14test |

6.12.6 :MMEMory:DATA? <file_name>

| | |
|-------------------------|---|
| Command | :MMEM:DATA? |
| Long | :MMEMory:DATA? |
| Parameters | <file_name> |
| Parameter Suffix | None |
| Description | The query form is <code>MMEMory:DATA? <file_name></code> with the response being the associated <data> in block format. |
| Example | Query : <code>MMEM:DATA? "C:\data.txt"</code> |

6.12.7 :MMEMory:MDIRectory <directory_name>

| | |
|-------------------------|---|
| Command | :MMEM:MDIR |
| Long | :MMEMory:MDIRectory |
| Parameters | <directory_name> |
| Parameter Suffix | None |
| Description | Creates a new directory. The <directory_name> parameter specifies the name to be created. |
| Example | Command : <code>MMEM:MDIR "C:\data_dir"</code> |

6.12.8 :MMEMory:MOVE <string>,<string>[,<string>,<string>]

| | |
|-------------------------|---|
| Command | :MMEM:MOVE |
| Long | :MMEMory:MOVE |
| Parameters | <string>,<string> |
| Parameter Suffix | None |
| Description | <p>Moves an existing file to a new file or an existing directory to a new directory. Two forms of parameters are allowed. The first form has two parameters. In this form, the first parameter specifies the source, and the second parameter specifies the destination.</p> <p>The second form has four parameters. In this form, the first and third parameters specify the file names. The second and fourth parameters specify the directories. The first pair of parameters specifies the source. The second pair specifies the destination. An error is generated if the source doesn't exist or the destination file already exists.</p> |
| Example | <p>Command :MMEM:MOVE "C:\data_dir","C:\newdata_dir"</p> |

6.12.9 :MMEMory:RDIRectory <directory_name>

| | |
|-------------------------|---|
| Command | :MMEM:RDIR |
| Long | :MMEMory:RDIRectory |
| Parameters | <directory_name> |
| Parameter Suffix | None |
| Description | Removes a directory. The <directory_name> parameter specifies the directory name to be removed. All files and directories under the specified directory are also removed. |
| Example | <p>Command :MMEM:RDIR "C:\newdata_dir"</p> |

6.12.10 :MMEMory:LOAD:CSTate <file_name>

| | |
|-------------------------|--|
| Command | :MMEM:LOAD:CST |
| Long | :MMEMory:LOAD:CSTate |
| Parameters | <file_name> |
| Parameter Suffix | None |
| Description | Current state of instrument is loaded from a file. |
| Example | Command :MMEM:LOAD:CST "C:\data.txt" |

6.12.11 :MMEMory:STORe:CSTate <file_name>

| | |
|-------------------------|--|
| Command | :MMEM:STOR:CST |
| Long | :MMEMory:STORe:CSTate |
| Parameters | <file_name> |
| Parameter Suffix | None |
| Description | Current state of instrument is stored to a file. |
| Example | Command :MMEM:STOR:CST "C:\data.txt" |

6.13 :OUTPut Subsystem

6.13.1 :OUTPut[1|2|3|4][:STATe][?] OFF|ON|0|1

| | |
|-------------------------|--|
| Command | :OUTP [?] |
| Long | :OUTPut [?] |
| Parameters | OFF ON 0 1 |
| Parameter Suffix | None |
| Description | Switch the amplifier of the output path for a channel on or off. |
| Example | Command :OUTP ON |
| | Query :OUTP? |

6.13.2 :OUTPut: ROSCillator:SOURce[?] INTernal|EXTernal|SCLK1|SCLK2

| | |
|-------------------------|---|
| Command | :OUTP:ROSC:SOUR [?] |
| Long | :OUTPut:ROSCillator:SOURce[?] |
| Parameters | INTernal EXTernal SCLK1 SCLK2 |
| Parameter Suffix | None |
| Description | Select which signal source is routed to the reference clock output: <ul style="list-style-type: none"> • INTernal: the module internal reference oscillator • EXTernal: the external reference clock from REF CLK IN with two variable dividers • SCLK1: DAC sample clock with variable divider and variable delay • SCLK2: DAC sample clock with fixed divider |

| | |
|----------------|--------------------------------|
| Example | Command :OUTP:ROSC:SOUR INT |
| | Query :OUTP:ROSC:SOUR? |

6.13.3 :OUTPut: ROSCillator:SCD[?] <sample_clock_divider>|MINimum|MAXimum

| | |
|-------------------------|---|
| Command | :OUTP:ROSC:SCD[?] |
| Long | :OUTPut:ROSCillator:SCD[?] |
| Parameters | sample_clock_divider MINimum MAXimum |
| Parameter Suffix | None |
| Description | Set or query the divider of the DAC sample clock signal routed to the reference clock output. |
| Example | Command :OUTP:ROSC:SCD 1 |
| | Query :OUTP:ROSC:SCD? |

6.13.4 :OUTPut: ROSCillator:RCD1[?] <reference_clock_divider1>|MINimum|MAXimum

| | |
|-------------------------|--|
| Command | :OUTP:ROSC:RCD1[?] |
| Long | :OUTPut:ROSCillator:RCD1[?] |
| Parameters | reference_clock_divider1 MINimum MAXimum |
| Parameter Suffix | None |
| Description | Set or query the first divider of the reference clock signal routed to the reference clock output. |
| Example | Command :OUTP:ROSC:RCD1 2 |
| | Query :OUTP:ROSC:RCD1? |

6.13.5 :OUTPut:ROSCillator:RCD2[?] <reference_clock_divider2>|MINimum|MAXimum

| | |
|-------------------------|--|
| Command | :OUTP:ROSC:RCD2 [?] |
| Long | :OUTPut:ROSCillator:RCD2[?] |
| Parameters | reference_clock_divider2 MINimum MAXimum |
| Parameter Suffix | None |
| Description | Set or query the second divider of the external reference clock signal routed to the reference clock output. |
| Example | <p>Command :OUTP:ROSC:RCD2 1</p> <p>Query :OUTP:ROSC:RCD2?</p> |

6.13.6 :OUTPut[1|2|3|4]:DIOFfset[?] <value>|MINimum|MAXimum

| | |
|-------------------------|---|
| Command | :OUTP:DIOF[?] |
| Long | :OUTPut:DIOFFset[?] |
| Parameters | <value> MINimum MAXimum |
| Parameter Suffix | None |
| Description | Differential Offset: The hardware can compensate for little offset differences between the normal and complement output. “<value>” is the offset to the calibrated optimum DAC value, so the minimum and maximum depend on the result of the calibration. |
| Example | <p>Command :OUTP:DIOF MAX</p> <p>Query :OUTP:DIOF?</p> |

Table 31: Differential offset

| Value | Normal Output | Complement Output |
|-------|------------------|-------------------|
| < 0 | Offset decreased | Offset increased |
| 0 | No offset | |
| > 0 | Offset increased | Offset decreased |

Due to the use of DAC values, the granularity is 1.

6.13.7 :OUTPut[1|2|3|4]:FILTer:FRATe[:VALue][?]

| | |
|-------------------------|--|
| Command | :OUTP:FILT:FRAT[?] |
| Long | :OUTPut:FILTer:FRATE[?] |
| Parameters | <value0>, <value1>...<value15> <block> |
| Parameter Suffix | None |
| Description | <p>Set or get the FIR filter coefficients for a channel to be used when the Sample Rate Divider for the Extended Memory is 1. The number of coefficients is 16 and the values are doubles between -2 and 2. They can be given as a list of comma-separated values or as IEEE binary block data of doubles.</p> <p>The coefficients can only be set using this command, when the predefined FIR filter type is set to USER.</p> |
| Example | <p>Query:</p> <pre>OUTP:FILT:FRAT?</pre> |

6.13.8 :OUTPut[1|2|3|4]:FILTer:FRATE:TYPE[?] LOWPass|ZOH|USER

| | |
|-------------------------|------------------------------|
| Command | :OUTP:FILT:FRAT:TYPE[?] |
| Long | :OUTPut:FILTer:FRATE:TYPE[?] |
| Parameters | LOWPass ZOH USER |
| Parameter Suffix | None |

| | |
|--------------------|---|
| Description | Set or get the predefined FIR filter type for a channel to be used when the Sample Rate Divider for the Extended Memory is 1. |
| | <ul style="list-style-type: none"> • LOWPass – equiripple lowpass filter with a passband edge at 75% of Nyquist • ZOH – Zero-order hold filter • USER – User-defined filter <p>The command form modifies the FIR filter coefficients according to the set filter type, except for type USER.</p> |
| Example | Command: OUTP:FILT:FRAT:TYPE LOWP |

6.13.9 :OUTPut[1|2|3|4]:FILTer:FRATe:SCALe[?] <scale>|MINimum|MAXimum

| | |
|-------------------------|--|
| Command | :OUTP:FILT:FRAT:SCAL[?] |
| Long | :OUTPut:FILTer:FRATe:SCALe[?] |
| Parameters | <scale> MINimum MAXimum |
| Parameter Suffix | None |
| Description | Set or get the FIR filter scaling factor for a channel to be used when the Sample Rate Divider for the Extended Memory is 1. The range is between 0 and 1. |
| Example | Command: OUTP:FILT:FRAT:SCAL 0.9 |

6.13.10 :OUTPut[1|2|3|4]:FILTer:FRATe:DELay[?] <delay>|MINimum|MAXimum

| | |
|-------------------------|---|
| Command | :OUTP:FILT:FRAT:DEL[?] |
| Long | :OUTPut:FILTer:FRATe:DELay[?] |
| Parameters | <delay> MINimum MAXimum |
| Parameter Suffix | [s ms us ns ps] |
| Description | <p>Set or get the FIR filter delay for a channel to be used when the Sample Rate Divider for the Extended Memory is 1. The range is -50ps..+50ps. The delay value has only effect for filter type LOWPass.</p> <p>The command form modifies the FIR filter coefficients according to the set delay value.</p> |

| | |
|----------------|-------------------------------------|
| Example | Command: OUTP:FILT:FRAT:DEL 10ps |
|----------------|-------------------------------------|

6.13.11 :OUTPut[1|2|3|4]:FILTer:HRATe[:VALue] [?]

| | |
|-------------------------|---|
| Command | :OUTP:FILT:HRAT[?] |
| Long | :OUTPut:FILTer:HRATe[?] |
| Parameters | <value0>, <value1>...<value31> <block> |
| Parameter Suffix | None |
| Description | <p>Set or get the FIR filter coefficients for a channel to be used when the Sample Rate Divider for the Extended Memory is 2. The number of coefficients is 32 and the values are doubles between -2 and 2. They can be given as a list of comma-separated values or as IEEE binary block data of doubles.</p> <p>The coefficients can only be set, when the predefined FIR filter type is set to USER.</p> |
| Example | <p>Query:</p> <pre>OUTP:FILT:HRAT?</pre> |

6.13.12 :OUTPut[1|2|3|4]:FILTer:HRATe:TYPE[?] NYQuist|LINEar|ZOH|USER

| | |
|-------------------------|---|
| Command | :OUTP:FILT:HRAT:TYPE[?] |
| Long | :OUTPut:FILTer:HRATE:TYPE[?] |
| Parameters | NYQuist LINEar ZOH USER |
| Parameter Suffix | None |
| Description | <p>Set or get the predefined FIR filter type for a channel to be used when the Sample Rate Divider for the Extended Memory is 2.</p> <ul style="list-style-type: none"> • NYQuist – Nyquist filter (half-band filter) with rolloff factor 0.2 • LINEar – Linear interpolation filter • ZOH – Zero-order hold filter • USER – User-defined filter <p>The command form modifies the FIR filter coefficients according to the set filter type, except for type USER.</p> |

Example Command:
OUTP:FILT:HRAT:TYPE NYQ

6.13.13 :OUTPut[1|2|3|4]:FILTer:HRATe:SCALe[?] <scale>|MINimum|MAXimum

Command : OUTP:FILT:HRAT:SCAL[?]
Long : OUTPut:FILTer:HRATE:SCALe[?]
Parameters <scale>|MINimum|MAXimum
Parameter Suffix None
Description Set or get the FIR filter scaling factor for a channel to be used when the Sample Rate Divider for the Extended Memory is 2. The range is between 0 and 1.
Example Command:
OUTP:FILT:HRAT:SCAL 0.9

6.13.14 :OUTPut[1|2|3|4]:FILTer:HRATe:DELay[?] <delay>|MINimum|MAXimum

Command : OUTP:FILT:HRAT:DEL[?]
Long : OUTPut:FILTer:HRATE:DELy[?]
Parameters <delay>|MINimum|MAXimum
Parameter Suffix [s|ms|us|ns|ps]
Description Set or get the FIR filter delay for a channel to be used when the Sample Rate Divider for the Extended Memory is 2. The range is -100ps..+100ps. The delay value has only effect for filter types NYQuist and LINear.
The command form modifies the FIR filter coefficients according to the set delay value.
Example Command:
OUTP:FILT:HRAT:DEL 10ps

6.13.15 :OUTPut[1|2|3|4]:FILTer:QRATe[:VALue] [?]

| | |
|-------------------------|---|
| Command | :OUTP:FILT:QRAT[?] |
| Long | :OUTPut:FILTer:QRATe[?] |
| Parameters | <value0>, <value1>...<value63> <block> |
| Parameter Suffix | None |
| Description | <p>Set or get the FIR filter coefficients for a channel to be used when the Sample Rate Divider for the Extended Memory is 4. The number of coefficients is 64 and the values are doubles between -2 and 2. They can be given as a list of comma-separated values or as IEEE binary block data of doubles.</p> <p>The coefficients can only be set, when the predefined FIR filter type is set to USER.</p> |
| Example | <p>Query: OUTP:FILT:QRAT?</p> |

6.13.16 :OUTPut[1|2|3|4]:FILTer:QRATe:TYPE[?] NYQuist|LINear|ZOH|USER

| | |
|-------------------------|--|
| Command | :OUTP:FILT:QRAT:TYPE[?] |
| Long | :OUTPut:FILTer:QRATe:TYPE[?] |
| Parameters | NYQuist LINear ZOH USER |
| Parameter Suffix | None |
| Description | <p>Set or get the predefined FIR filter type for a channel to be used when the Sample Rate Divider for the Extended Memory is 4.</p> <ul style="list-style-type: none"> • NYQuist – Nyquist filter (quarter-band filter) with rolloff factor 0.2 • LINear – Linear interpolation filter • ZOH – Zero-order hold filter • USER – User-defined filter <p>The command form modifies the FIR filter coefficients according to the set filter type, except for type USER.</p> |
| Example | <p>Command: OUTP:FILT:QRAT:TYPE NYQ</p> |

6.13.17 :OUTPut[1|2|3|4]:FILT:QRATe:SCALe[?] <scale>|MINimum|MAXimum

| | |
|-------------------------|--|
| Command | :OUTP:FILT:QRAT:SCAL [?] |
| Long | :OUTPut:FILT:QRATe:SCALe [?] |
| Parameters | <scale> MINimum MAXimum |
| Parameter Suffix | None |
| Description | Set or get the FIR filter scaling factor for a channel to be used when the Sample Rate Divider for the Extended Memory is 4. The range is between 0 and 1. |
| Example | Command: OUTP:FILT:QRAT:SCAL 0.9 |

6.13.18 :OUTPut[1|2|3|4]:FILT:QRATe:DELay[?] <delay>|MINimum|MAXimum

| | |
|-------------------------|--|
| Command | :OUTP:FILT:QRAT:DEL [?] |
| Long | :OUTPut:FILT:QRATe:DELay [?] |
| Parameters | <delay> MINimum MAXimum |
| Parameter Suffix | [s ms us ns ps] |
| Description | Set or get the FIR filter delay for a channel to be used when the Sample Rate Divider for the Extended Memory is 4. The range is -200ps..+200ps. The delay value has only effect for filter types NYQuist and LINear. The command form modifies the FIR filter coefficients according to the set delay value. |
| Example | Command: OUTP:FILT:QRAT:DEL 10ps |

6.14 Sampling Frequency Commands

6.14.1 [:SOURce]:FREQuency:RASTer[?] <frequency>|MINimum|MAXimum

| | | | | | | | | | |
|-------------------------|--|---------|--|---|---------------|-------|--|---|------------|
| Command | :FREQ:RAST [?] | | | | | | | | |
| Long | :FREQuency:RASTer[?] | | | | | | | | |
| Parameters | <frequency> MINimum MAXimum | | | | | | | | |
| Parameter Suffix | None | | | | | | | | |
| Description | Set or query the sample frequency of the output DAC. | | | | | | | | |
| Example | <table><tr><td>Command</td><td></td></tr><tr><td>:</td><td>FREQ:RAST MIN</td></tr><tr><td>Query</td><td></td></tr><tr><td>:</td><td>FREQ:RAST?</td></tr></table> | Command | | : | FREQ:RAST MIN | Query | | : | FREQ:RAST? |
| Command | | | | | | | | | |
| : | FREQ:RAST MIN | | | | | | | | |
| Query | | | | | | | | | |
| : | FREQ:RAST? | | | | | | | | |

6.15 Reference Oscillator Commands

6.15.1 [:SOURce]:ROSCillator:SOURce[?] EXTernal|AXI|INTernal

| | |
|-------------------------|--|
| Command | <code>:ROSC:SOUR[?]</code> |
| Long | <code>:ROSCillator:SOURce[?]</code> |
| Parameters | EXTernal AXI INTernal |
| Parameter Suffix | None |
| Description | <p>Set or query the reference clock source.</p> <ul style="list-style-type: none">• EXTernal: reference is taken from REF CLK IN.• AXI: reference is taken from AXI backplane.• INTernal: reference is taken from module internal reference oscillator. May not be available with every hardware. <p>Command not supported with Revision 1 hardware.</p> |
| Example | <p>Command <code>:ROSC:SOUR AXI</code></p> <p>Query <code>:ROSC:SOUR?</code></p> |

6.15.2 [:SOURce]:ROSCillator:SOURce:CHECK? EXternal|AXI|INTernal

| | |
|-------------------------|--|
| Command | :ROSC:SOUR:CHEC? |
| Long | :ROSCillator:SOURce:CHECK? |
| Parameters | EXTernal AXI INTernal |
| Parameter Suffix | None |
| Description | Check if a reference clock source is available. Returns 1 if it is available and 0 if not. |
| Example | Query :ROSC:SOUR:CHEC? AXI |

6.15.3 [:SOURce]:ROSCillator:FREQuency[?] <frequency>|MINimum|MAXimum

| | |
|-------------------------|--|
| Command | :ROSC:FREQ[?] |
| Long | :ROSCillator:FREQuency[?] |
| Parameters | <frequency> MINimum MAXimum |
| Parameter Suffix | None |
| Description | Set or query the expected reference clock frequency, if the external reference clock source is selected. |
| Example | Command :ROSC:FREQ MIN Query :ROSC:FREQ? |

6.15.4 [:SOURce]:ROSCillator:RANGE[?] RANG1| RANG2

| | |
|-------------------------|---|
| Command | <code>:ROSC:RANG [?]</code> |
| Long | <code>:ROSCillator:RANGE[?]</code> |
| Parameters | RANG1 RANG2 |
| Parameter Suffix | None |
| Description | Set or query the reference clock frequency range, if the external reference clock source is selected. <ul style="list-style-type: none">• RANG1: 10...300 MHz• RANG2: 210MHz...17GHz |
| Example | Command <code>:ROSC:RANG RANG1</code> Query <code>:ROSC:RANG?</code> |

6.15.5 [:SOURce]:ROSCillator:RNG1|RNG2:FREQuency[?] <frequency>|MINimum|MAXimum

| | | | | | | | | | | | |
|-------------------------|--|----------------|--|---------------------|--|------|--|--------------|--|------------------|--|
| Command | :ROSC:RNG1 RNG2:FREQ [?] | | | | | | | | | | |
| Long | :ROSCillator:RNG1 RNG2:FREQuency[?] | | | | | | | | | | |
| Parameters | <frequency> MINimum MAXimum | | | | | | | | | | |
| Parameter Suffix | None | | | | | | | | | | |
| Description | Set or query the reference clock frequency for a specific reference clock range. Current range remains unchanged. | | | | | | | | | | |
| | <ul style="list-style-type: none">• RNG1: 10...300 MHz• RNG2: 210MHz...17GHz | | | | | | | | | | |
| Example | <table><tr><td>Command</td><td></td></tr><tr><td>:ROSC:RNG1:FREQ MIN</td><td></td></tr><tr><td> </td><td></td></tr><tr><td>Query</td><td></td></tr><tr><td>:ROSC:RNG1:FREQ?</td><td></td></tr></table> | Command | | :ROSC:RNG1:FREQ MIN | | | | Query | | :ROSC:RNG1:FREQ? | |
| Command | | | | | | | | | | | |
| :ROSC:RNG1:FREQ MIN | | | | | | | | | | | |
| | | | | | | | | | | | |
| Query | | | | | | | | | | | |
| :ROSC:RNG1:FREQ? | | | | | | | | | | | |

6.16 :VOLTage Subsystem

Set the output voltages for a channel.

6.16.1 [:SOURce]:VOLTage[1|2|3|4][:LEVel][[:IMMediate][[:AMPLitude][?]] <level>|MINimum|MAXimum

| | |
|-------------------------|---|
| Command | :VOLT [?] |
| Long | :VOLTage [?] |
| Parameters | <level> MINimum MAXimum |
| Parameter Suffix | None |
| Description | Set or query the output amplitude. |
| Example | Command :VOLT 0.685 Query :VOLT? |

6.16.2 [:SOURce]:VOLTage[1|2|3|4][:LEVel][:IMMediate]:OFFSet[?] <level>|MINimum|MAXimum

| | |
|-------------------------|---------------------------------|
| Command | :VOLT:OFFS [?] |
| Long | :VOLTage:OFFSet [?] |
| Parameters | <level> MINimum MAXimum |
| Parameter Suffix | None |
| Description | Set or query the output offset. |
| Example | Command :VOLT:OFFS 0.02 |
| | Query :VOLT:OFFS? |

6.16.3 [:SOURce]:VOLTage[1|2|3|4][:LEVel][:IMMediate]:HIGH[?] <level>|MINimum|MAXimum

| | |
|-------------------------|-------------------------------------|
| Command | :VOLT:HIGH [?] |
| Long | :VOLTage:HIGH [?] |
| Parameters | <level> MINimum MAXimum |
| Parameter Suffix | None |
| Description | Set or query the output high level. |
| Example | Command :VOLT:HIGH 3e-1 |
| | Query :VOLT:HIGH? |

**6.16.4 [:SOURce]:VOLTage[1|2|3|4][:LEVel][:IMMediate]:LOW[?]
 <level>|MINimum|MAXimum**

Command :VOLT:LOW[?]

Long :VOLTage:LOW[?]

Parameters <level>|MINimum|MAXimum

Parameter Suffix None

Description Set or query the output low level.

Example Command
:VOLT:LOW -0.3

Query
:VOLT:LOW?

**6.16.5 [:SOURce]:VOLTage[1|2|3|4][:LEVel][:IMMediate]:TERMination[?]
 <level>|MINimum|MAXimum**

Command :VOLT:TERM[?]

Long :VOLTage:TERMination[?]

Parameters <level>|MINimum|MAXimum

Parameter Suffix None

Description Set or query the termination voltage level.

Example Command
:VOLT:TERM 0.3

Query
:VOLT:TERM?

6.17 [:SOURce]:FUNCtion:MODE ARBitrary|STSequence|STSCenario

| | |
|-------------------------|---|
| Command | <code>:FUNC:MODE [?]</code> |
| Long | <code>:FUNCtion:MODE [?]</code> |
| Parameters | ARBitrary STSequence STSCenario |
| Parameter Suffix | None |
| Description | <p>Use this command to set or query the type of waveform that will be generated on the channels that use the extended memory.</p> <ul style="list-style-type: none">• ARBitrary – arbitrary waveform segment• STSequence – sequence• STSCenario – scenario <p>The channels that use internal memory are always in ARBitrary mode.</p> |
| Example | <p>Command <code>:FUNC:MODE ARB</code></p> <p>Query <code>:FUNC:MODE?</code></p> |

6.18 :STABle Subsystem

Use the Sequence Table subsystem to prepare the instrument for sequence and scenario generation. The Sequencing capabilities can only be used by the channels sourced from Extended Memory. These channels share a common Sequence Table and execute the same sequence or scenario. The channels sourced from Internal Memory play only one waveform.

Follow these steps for all function modes:

- First create waveform data segments in the module memory like described in the “Arbitrary Waveform Generation” paragraph of the “TRACe subsystem”.
- Create sequence table entries that refer to the waveform segments using the STAB:DATA command.

6.18.1 [:SOURce]:STABle:RESet

| | |
|-------------------------|---|
| Command | : STAB:RES |
| Long | : STABle:RESet |
| Parameters | None |
| Parameter Suffix | None |
| Description | Reset all sequence table entries to default values. |
| Example | Command : STAB:RES |

6.18.2 [:SOURce]:STABle:DATA[?]

<sequence_table_index>,(<length>|<block>|<value>,<value>...)

| | |
|-------------------------|--|
| Command | : STAB:DATA[?] |
| Long | : STABle:DATA[?] |
| Parameters | <sequence_table_index>,(<length> <block> <value>,<value>...) |
| Parameter Suffix | None |

| | |
|--------------------|---|
| Description | <p>The command form writes directly into the sequencer memory. The query form reads the data from the sequencer memory, if all segments are read-write. The query returns an error, if at least one write-only segment in the waveform memory exists. Reading is only possible, when the signal generation is stopped. Writing is possible, when signal generation is stopped or when signal generation is started in dynamic mode.</p> <p>The sequencer memory has 16777215 (16M – 1) entries. With this command entries can be directly manipulated using 6 32-bit words per entry. Individual entries or multiple entries at once can be manipulated. The data can be given in IEEE binary block format or in comma-separated list of 32-bit values.</p> <p><sequence_table_index> – index of the sequence table entry to be accessed <length> – number of entries to be read <block> – multiple sequence vectors, each consisting of 6 32-bit parameter values <value> – a 32-bit parameter value; the meaning depends on the type of sequence entry to be created and on the index position for an entry, see following tables.</p> |
| Example | <p>Command using comma separated parameter:</p> <pre>// Data Entry: [:SOURce]:STABle:DATA <sequence_id>,<control_parameter>,<sequence_loop><segment_loop><segment_id>,<start_address>,<end_address></pre> <p>Example:</p> <pre>// Create a data entry at index 0 of the sequence table. // Mark as start of sequence, sequence loop count = 1, segment loop count = 2, // segment id = 1, // segment start offset is 0, segment end offset is equal to the end of the segment. :STAB:DATA 0,#0x10000000,1,2,1,0, #0xFFFFFFFF</pre> <pre>// Idle entry [:SOURce]:STABle:DATA <sequence_id>,<control_parameter>,<sequence_loop><command-code><idle_sample>,<idle_delay>,<0></pre> <p>Example:</p> <pre>// Create an idle delay entry at index 0 of the sequence table. // Mark as command (control_parameter = 0x80000000) , sequence loop count = // // 1, command code = 0 , idle sample = 0, idle delay = 960 :STAB:DATA 0, #0x80000000,1,0,0,960,0</pre> <p>Using Data block:</p> <pre>[:SOURce]:STABle:DATA <sequence_id>,<data_block> :STAB:DATA 0, <data_block></pre> <p>Query</p> <pre>[:SOURce]:STABle:DATA? <sequence_id>,<length> :STAB:DATA? 0, 6</pre> |

The sequence table can contain data and idle delay entries. The following table shows, which parameters are needed to create the different entry types.

Table 32: Sequencer table entries

| Parameter Index | Data Entry | Idle Delay Entry |
|-----------------|----------------------|---------------------|
| 0 | Control | Control |
| 1 | Sequence Loop Count | Sequence Loop Count |
| 2 | Segment Loop Count | Command Code |
| 3 | Segment Id | Idle Sample |
| 4 | Segment Start Offset | Idle Delay |
| 5 | Segment End Offset | 0 |

The following tables show the meaning of the parameters and the applicability per sequence entry type. Bits marked as “Reserved” or “N/A” must be set to 0.

Table 33: Control

| Bit | Width | Meaning | Data | Idle |
|-------|-------|--|------|------|
| 31 | 1 | Data/command selection 0: Data 1: Command (type of command is selected by command code) | X | X |
| 30 | 1 | End Marker Sequence | X | X |
| 29 | 1 | End Marker Scenario | X | X |
| 28 | 1 | Init Marker Sequence | X | X |
| 27:25 | 3 | Reserved | X | X |
| 24 | 1 | Marker Enable | X | N/A |
| 23:20 | 4 | Advancement Mode Sequence 0: Auto 1: Conditional 2: Repeat 3: Single 4 – 15: Reserved | X | X |
| 19:16 | 4 | Advancement Mode Segment 0: Auto 1: Conditional 2: Repeat 3: Single 4 – 15: Reserved | X | N/A |
| 11:0 | 12 | Reserved | X | X |

Table 34: Sequence loop count

| Bit | Width | Meaning | Data | Idle |
|------|-------|---|------|------|
| 31:0 | 32 | Number of sequence iterations (1..4G-1), only applicable in the first entry of a sequence | X | X |

Table 35: Segment loop count

| Bit | Width | Meaning | Data | Idle |
|------|-------|--|------|------|
| 31:0 | 32 | Number of segment iterations (1..4G-1) | X | N/A |

Table 36: Segment Id

| Bit | Width | Meaning | Data | Idle |
|------------|--------------|-----------------------|-------------|-------------|
| 31:25 | 7 | Reserved | X | N/A |
| 24:0 | 25 | Segment id (1 .. 16M) | X | N/A |

Table 37: Segment start offset

| Bit | Width | Meaning | Data | Idle |
|------------|--------------|---|-------------|-------------|
| 31:0 | 32 | Allows specifying a segment start address in samples, if only part of a segment loaded into waveform data memory is to be used. The value must be a multiple of twice the granularity of the selected waveform output mode. | X | N/A |

Table 38: Segment end offset

| Segment End Offset | | | | |
|---------------------------|--------------|---|-------------|-------------|
| Bit | Width | Meaning | Data | Idle |
| 31:0 | 32 | Allows specifying a segment end address in samples, if only part of a segment loaded into waveform data memory is to be used. The value must obey the granularity of the selected waveform output mode. You can use the value 0xffffffff, if the segment end address equals the last sample in the segment. | X | N/A |

Table 39: Command code

| Command Code | | | | |
|---------------------|--------------|-------------------------------|-------------|-------------|
| Bit | Width | Meaning | Data | Idle |
| 31:16 | 16 | Reserved | N/A | X |
| 15:0 | 16 | Command code 0: Idle Delay | N/A | X |

Table 40: Idle sample

| Idle Sample | | | | |
|--------------------|--------------|---|-------------|-------------|
| Bit | Width | Meaning | Data | Idle |
| 31:8 | 24 | Reserved | N/A | X |
| 7:0 | 8 | Sample to be played during pause. Bits 7:0 contain the DAC value. | N/A | X |

Table 41: Idle delay

| Idle Delay | | | Data | Idle |
|------------|-------|---------------------------------------|--------|----------------------|
| Bit | Width | Meaning | | |
| 31:0 | 32 | Idle delay in Waveform Sample Clocks. | N/A | X |
| | | Sample Rate Divider | Min | Max |
| | | 1 | 10*256 | $(2^{24}-1)*256+255$ |
| | | 2 | 10*128 | $(2^{24}-1)*128+127$ |
| | | 4 | 10*64 | $(2^{24}-1)*64+63$ |

Example:

```
// Create a data entry at index 0 of the sequence table.  
// Mark as start of sequence, sequence loop count = 1, segment loop count = 2,  
// segment id = 3,  
// segment start offset is 240; segment end offset is equal to the end of the segment.  
STAB:DATA 0,#h10000000,1,2,3,240,#ffffffff  
  
// Create an idle delay entry at index 0 of the sequence table.  
// Mark as command, sequence loop count = 1, command code = 0 , idle sample = 0,  
// idle delay = 2560 Waveform Sample Clocks, last parameter word is unused.  
STAB:DATA 0,#h80000000,1,0,0,2560,0
```

6.18.3 [:SOURce]:STABle:DATA:BLOC? <sequence_table_index>,<length>

| | |
|-------------------------|---|
| Command | :STAB:DATA:BLOC? |
| Long | :STABle:DATA:BLOC? |
| Parameters | <sequence_table_index>,<length> |
| Parameter Suffix | None |
| Description | <p>This query returns the same data as the “:STAB:DATA?” query, but in IEEE binary block format.</p> <ul style="list-style-type: none"> • <sequence_table_index> – index of the sequence table entry to be accessed • <length> - number of entries to be read |
| Example | <p>Query</p> <pre>[:SOURce]:STABle:DATA:BLOC? <sequence_id>,<length> :STAB:DATA:BLOC? 0, 6</pre> |

6.18.4 [:SOURce]:STABle:SEQuence:SELect[?] <sequence_table_index>|MINimum|MAXimum

| | |
|-------------------------|---|
| Command | :STAB:SEQ:SEL[?] |
| Long | :STABle:SEQuence:SELect[?] |
| Parameters | <sequence_table_index> MINimum MAXimum |
| Parameter Suffix | None |
| Description | Select where in the sequence table the sequence starts in STSequence mode. In dynamic sequence selection mode select the sequence that is played before the first sequence is dynamically selected. |
| Example | <p>Command</p> <pre>:STAB:SEQ:SEL 0</pre> <p>Query</p> <pre>:STAB:SEQ:SEL?</pre> |

6.18.5 [:SOURce]:STABle:SEQuence:STATe?

| | |
|-------------------------|---|
| Command | :STAB:SEQ:STAT? |
| Long | :STABle:SEQuence:STATE? |
| Parameters | None |
| Parameter Suffix | None |
| Description | This query returns an integer value containing the sequence execution state and the currently executed sequence table entry |

Table 42: Returned sequence state

| Bit | Width | Meaning |
|-------|-------|---|
| 31:27 | 5 | Reserved |
| 26:25 | 2 | Sequence execution state 0: Idle 1: Waiting for Trigger 2: Running 3: Waiting for Advancement Event |
| 24:0 | 25 | Index of currently executed sequence table entry. In Idle state the value is undefined. |

| | |
|----------------|-----------------|
| Example | Query |
| | :STAB:SEQ:STAT? |

6.18.6 [:SOURce]:STABle:DYNamic:[STATe][?] OFF|ON|0|1

| | |
|-------------------------|--|
| Command | <code>:STAB:DYN[?]</code> |
| Long | <code>:STABle:DYNamic[?]</code> |
| Parameters | OFF ON 0 1 |
| Parameter Suffix | None |
| Description | <p>Use this command to enable or disable dynamic mode.</p> <p>If dynamic mode is switched off, segments or sequences can only be switched in program mode, that is signal generation must be stopped. In arbitrary mode use TRACe[1 2 3 4]:SElect to switch to a new segment. In sequence mode use [:SOURce]:STABle:SEQUence:SElect to switch to a new sequence.</p> <p>If dynamic mode is switched on, segments or sequences can be switched dynamically when signal generation is active. The next segment or sequence is either selected by the command [:SOURce]:STABle:DYNamic:SElect or by a signal fed into the dynamic port of the M8197 module. The external input values select sequence table entries with corresponding indices.</p> |
| Example | <p>Command <code>:STAB:DYN 0</code></p> <p>Query <code>:STAB:DYN?</code></p> |

6.18.7 [:SOURce]:STABle:DYNamic:SELect <sequence_table_index>

| | |
|-------------------------|---|
| Command | <code>:STAB:DYN:SEL</code> |
| Long | <code>:STABle:DYNamic:SELect</code> |
| Parameters | <sequence_table_index> |
| Parameter Suffix | None |
| Description | When the dynamic mode for segments or sequences is active, set the sequence table entry to be executed next.. |
| Example | <p>Command <code>:STAB:DYN:SEL 0</code></p> |

6.18.8 [:SOURce]:STABle:SCENario:SElect[?] <sequence_table_index>|MINimum|MAXimum

| | |
|-------------------------|--|
| Command | : STAB:SCEN:SEL [?] |
| Long | : STABle:SCENario:SElect[?] |
| Parameters | <sequence_table_index> MINimum MAXimum |
| Parameter Suffix | None |
| Description | Select where in the sequence table the scenario starts in STSCenario mode. |
| Example | Command : STAB:SCEN:SEL 0 |
| | Query : STAB:SCEN:SEL? |

6.18.9 [:SOURce]:STABle:SCENario:ADVance[?] AUTO|CONDitional|REPeat|SINGle

| | |
|-------------------------|--|
| Command | : STAB:SCEN:ADV [?] |
| Long | : STABle:SCENario:ADVance[?] |
| Parameters | AUTO COND REP SING |
| Parameter Suffix | None |
| Description | Set or query the advancement mode for scenarios. |
| Example | Command : STAB:SCEN:ADV AUTO |
| | Query : STAB:SCEN:ADV? |

6.18.10 [:SOURce]:STABle:SCENario:COUNt[?] <count>|MINimum|MAXimum

| | |
|-------------------------|--|
| Command | <code>:STAB:SCEN:COUN[?]</code> |
| Long | <code>:STABle:SCENario:COUNt[?]</code> |
| Parameters | <code><count> MINimum MAXimum</code> |
| Parameter Suffix | None |
| Description | Set or query the loop count for scenarios. <ul style="list-style-type: none">• <code><count></code> – 1..4G-1: number of times the scenario is repeated. |
| Example | Command <code>:STAB:SCEN:COUN 2</code> Query <code>:STAB:SCEN:COUN?</code> |

6.19 Frequency and Phase Response Data Access

6.19.1 [:SOURce]: CHARacteris[1|2|3|4][:VALue]? [<amplitude>,<sample_frequency>]]

| | |
|-------------------------|---|
| Command | :CHAR? |
| Long | :CHARacteris? |
| Parameters | <ul style="list-style-type: none"> • <amplitude> the output amplitude • <sample_frequency> the sample frequency |
| Parameter Suffix | None |
| Description | <p>Query the frequency and phase response data for a channel. The query returns the data for the AWG sample frequency and output amplitude passed as parameters as a string of comma-separated values. If the sample frequency or both parameters are omitted, the currently configured AWG sample frequency and output amplitude are used.</p> <p>The frequency and phase response includes the sin x/ x roll-off of the currently configured AWG sample frequency. As a result the query delivers different results when performed at e.g. 60GSa/s or 65 GSa/s.</p> <p>To achieve optimum frequency and phase compensation results, the frequency and phase response has been characterized individually per channel and for different output amplitudes. As a result, the query delivers different results when performed at e.g. 500 mV or 800 mV.</p> <p>The frequency and phase response refers to the 2.92 mm connector. In case external cables from the 2.92 mm connector to the Device Under Test (DUT) shall be mathematically compensated for as well, the corresponding S-Parameter of that cable must be taken into account separately.</p> <p>Format: The first three values are output frequency 1 in Hz, corresponding relative magnitude in linear scale, corresponding phase in radians. The next three values are output frequency 2, corresponding relative magnitude, corresponding phase, and so on.</p> |
| Example | <p>Query</p> <pre>:CHAR2? "0,1.01068,0, 1e+008,1.00135,-6.11215e-005, 2e+008,0.993992,-0.000179762, ... 3.19e+010,0.0705237,-3.82659, 3.2e+010,0.0665947,-3.85028"</pre> |

6.20 :TRACe Subsystem

Use the :TRACe subsystem to control the arbitrary waveforms and their respective parameters:

- Create waveform segments of arbitrary size with optional initialization.
- Download waveform data with or without marker data into the segments.
- Delete one or all waveform segments from the waveform memory.

6.20.1 Waveform Data Format

In the data formats shown below the fields have the following meanings:

DB7...DB0 – Sample as signed 8-bit value, valid range is -128 to +127.

M1, M2 – Marker bits for Marker 1 and 2 to be output on channel 3 and 4, respectively.

Table 43: Sample data format without markers

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-----|-----|-----|-----|-----|-----|-----|-----|
| DB7 | DB6 | DB5 | DB4 | DB3 | DB2 | DB1 | DB0 |
| DB7 | DB6 | DB5 | DB4 | DB3 | DB2 | DB1 | DB0 |
| ... | | | | | | | |
| DB7 | DB6 | DB5 | DB4 | DB3 | DB2 | DB1 | DB0 |

Table 44: Sample data format with markers

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-----|-----|-----|-----|-----|-----|-----|-----|
| DB7 | DB6 | DB5 | DB4 | DB3 | DB2 | DB1 | DB0 |
| X | X | X | X | X | X | M2 | M1 |
| DB7 | DB6 | DB5 | DB4 | DB3 | DB2 | DB1 | DB0 |
| X | X | X | X | X | X | M2 | M1 |
| ... | | | | | | | |
| DB7 | DB6 | DB5 | DB4 | DB3 | DB2 | DB1 | DB0 |
| X | X | X | X | X | X | M2 | M1 |

6.20.2 Arbitrary Waveform Generation

To prepare your module for arbitrary waveform generation follow these steps:

- Set Instrument Mode (number of channels), Memory Sample Rate Divider, and memory usage of the channels (Internal/Extended).
- Define a segment using the various forms of the
 - :TRAC[1|2|3|4]:DEF command.
- Fill the segment with sample values using
 - :TRAC[1|2|3|4]:DATA.
- Signal generation starts after calling INIT:IMM.
- Use the :TRAC[1|2|3|4]:CAT? query to read the length of a waveform loaded into the memory of a channel. Use the :TRAC[1|2|3|4]:DEL:ALL command to delete a waveform from the memory of a channel.

6.20.3 TRACe[1|2|3|4]:MMODe[?]

| | |
|-------------------------|---|
| Command | :TRAC [1 2 3 4] :MMOD [?] |
| Long | :TRACe [1 2 3 4] :MMODE [?] |
| Parameters | INTernal EXTended <ul style="list-style-type: none"> • INTernal – the channel uses Internal Memory • EXTended – the channel uses Extended Memory • NONE – the channel is not used in this configuration (query only) |
| Parameter Suffix | None |
| Description | Use this command or query to set or get the source of the waveform samples for a channel. There are dependencies between this parameter, the same parameter for other channels, the memory sample rate divider and the instrument mode (number of channels). The tables in section 1.5.5 show the available combinations. It is recommended to set these parameters in one transaction. The value of this parameter for each channel determines the target (Internal/Extended Memory) of the waveform transfer operation using the TRAC:DATA command. |
| Example | Command Set 2-channel instrument mode, memory sample rate divider = 2, channels 1 and 4 use the extended memory. Channels 2 and 3 are not used in this configuration. :INST:DACM DUAL;:INST:MEM:EXT:RDIV DIV2;:TRAC1:MMOD EXT;:TRAC4:MMOD EXT |

6.20.4 :TRAC[1|2|3|4]:DEF

| | |
|-------------------------|--|
| Command | :TRAC [1 2 3 4] :DEF |
| Long | :TRACe [1 2 3 4] :DEFine |
| Parameters | <p><segment_id>,<length>[,<init_value>]</p> <ul style="list-style-type: none"> • <segment_id> – id of the segment • <length> – length of the segment in samples, marker samples do not count • <init_value> – optional initialization DAC value |
| Parameter Suffix | None |
| Description | <p>Use this command to define the size of a waveform memory segment. If <init_value> is specified, all values in the segment are initialized. If not specified, memory is only allocated but not initialized.</p> <p>If the channel is sourced from Extended Memory, the same segment is defined on all other channels sourced from Extended Memory.</p> |
| Example | <p>Commands</p> <p>Define a segment with id 1 and length 1280 samples on channel 1. Initialize all samples to 0.</p> <pre>TRAC1:DEF 1,1280,0</pre> |

6.20.5 :TRAC[1|2|3|4]:DEF:NEW?

| | |
|-------------------------|--|
| Command | :TRAC [1 2 3 4] :DEF:NEW? |
| Long | :TRACe [1 2 3 4] :DEFine:NEW? |
| Parameters | <p><length> [,<init_value>]</p> <ul style="list-style-type: none"> • <length> – length of the segment in samples, marker samples do not count • <init_value> – optional initialization DAC value |
| Parameter Suffix | None |
| Description | <p>Use this query to define the size of a waveform memory segment. If <init_value> is specified, all values in the segment are initialized. If not specified, memory is only allocated but not initialized. If the query was successful, a new <segment_id> will be returned.</p> <p>If the channel is sourced from Extended Memory, the same segment is defined on all other channels sourced from Extended Memory.</p> |
| Example | <p>Query</p> <p>Define a segment of length 1280 samples on channel 1. Returns the segment id.</p> <pre>TRAC1:DEF:NEW? 1280</pre> |

6.20.6 :TRAC[1|2|3|4]:DEF:WONL

| | |
|-------------------------|---|
| Command | :TRAC[1 2 3 4]:DEF:WONL |
| Long | :TRACe[1 2 3 4]:DEFine:WONLy |
| Parameters | <p><segment_id>,<length>[,<init_value>]</p> <ul style="list-style-type: none"> • <segment_id> – id of the segment • <length> – length of the segment in samples, marker samples do not count • <init_value> – optional initialization DAC value. |
| Parameter Suffix | None |
| Description | <p>Use this command to define the size of a waveform memory segment. If <init_value> is specified, all values in the segment are initialized. If not specified, memory is only allocated but not initialized. The segment will be flagged write-only, so it cannot be read back or stored.</p> <p>If the channel is sourced from Extended Memory, the same segment is defined on all other channels sourced from Extended Memory.</p> |
| Example | <p>Command</p> <p>Define a write-only segment with id 1 and length 1280 samples on channel 1.</p> <pre>:TRAC1:DEF:WONL 1,1280</pre> |

6.20.7 :TRAC[1|2|3|4]:DEF:WONL:NEW?

| | |
|-------------------------|--|
| Command | :TRAC[1 2 3 4]:DEF:WONL:NEW? |
| Long | :TRACe[1 2 3 4]:DEFine:WONLy:NEW? |
| Parameters | <ul style="list-style-type: none"> • <length>[,<init_value>] • <length> – length of the segment in samples, marker samples do not count • <init_value> – optional initialization DAC value |
| Parameter Suffix | None |
| Description | <p>Use this query to define the size of a waveform memory segment. If <init_value> is specified, all sample values in the segment are initialized. If not specified, memory is only allocated but not initialized. If the query was successful, a new <segment_id> will be returned. The segment will be flagged write-only, so it cannot be read back or stored.</p> <p>If the channel is sourced from Extended Memory, the same segment is defined on all other channels sourced from Extended Memory.</p> |

| | |
|----------------|---|
| Example | Query Define a write-only segment with length 1280 samples on channel 1. Returns the segment Id. <code>:TRAC1:DEF:WONL:NEW? 1280</code> |
|----------------|---|

6.20.8 :TRAC[1|2|3|4]:DATA[?]

| | |
|-------------------------|--|
| Command | <code>:TRAC [1 2 3 4] :DATA[?]</code> |
| Long | <code>:TRACe [1 2 3 4] :DATA[?]</code> |
| Parameters | <code><segment_id>,<offset>,(<length> <block> <numeric_values>)</code> <ul style="list-style-type: none"> • <code><segment_id></code> – id of the segment • <code><offset></code> – offset from segment start in samples (marker samples do not count) to allow splitting the transfer in smaller portions • <code><length></code> – number of samples to read in the query case • <code><block></code> – waveform data samples in the data format described above in IEEE binary block format • <code><numeric_values></code> – waveform data samples in the data format described above in comma separated list format |
| Parameter Suffix | None |
| Description | <p>Use this command to load waveform data into the module memory. If <code><segment_id></code> is already filled with data, the new values overwrite the current values. If length is exceeded error -223 (too much data) is reported.</p> <p>Reading is only possible, when the signal generation is stopped. Writing is possible, when signal generation is stopped or when signal generation is started in dynamic mode.</p> <p>The target (Internal/Extended Memory) of the waveform transfer is given by the value set by the TRAC:MMOD command for the channel. The data format (waveform samples only or interleaved waveform and marker samples) is given by the DAC Mode set by the INST:DACM command.</p> <p>When transferring data to Extended Memory, the parameter <code><offset></code> must contain a value corresponding to an even number of memory vectors. The number of samples in a memory vector equals the waveform memory granularity. This limitation does not exist for transferring data to Internal Memory.</p> <p>This SCPI has the following syntax for command/query:</p> <p>Command</p> <code>:TRACe [1 2 3 4] [:DATA] <segment_id>,<offset>,(<block> <numeric_values>)</code> <p>Query</p> <code>:TRACe [1 2 3 4] [:DATA] [?] <segment_id>,<offset>,<length></code> |

Example**Command**

Load data consisting of 1280 samples (waveform data samples only) as comma-separated list into previously defined segment 1 starting at sample offset 0.

```
:TRAC1:DATA 1,0,0,1,2,...,1279
```

Load data consisting of 1280 waveform data samples and same number of marker samples (interleaved waveform data and marker samples) as comma-separated list into previously defined segment 1 starting at sample offset 0. The marker sample "3" corresponds to a high level for Marker 1 and 2 for the first sample of the waveform segment. Marker samples with value 0 correspond to a low level for Marker 1 and 2.

```
:TRAC1:DATA 1,0,0,3,1,0,2,0,...,1279,0
```

Query

```
:TRAC:DATA? 1,0,1280
```

NOTE

If the segment is split in smaller sections, the sections have to be written in order of ascending <offset> values. If modification of the segment contents is necessary, the whole segment with all sections must be rewritten.

If segments are created and deleted in arbitrary order, their position and order in memory cannot be controlled by the user, because the M8195 reuses the memory space of deleted segments for newly created segments. To fulfill the streaming and minimum linear playtime requirements the only way to control the position of the first downloaded segment and the order of the following segments is to delete all segments from memory (:TRACe[1|2|3|4]:DElete:ALL) and then creating the segments in the order in which they should be placed in memory.

6.20.9 :TRAC[1|2|3|4]:DATA:BLOC?

| | |
|-------------------------|--|
| Command | :TRAC:DATA:BLOC? |
| Long | :TRACe:DATA:BLOCK? |
| Parameters | <segment_id>,<offset>,<length> |
| Parameter Suffix | None |
| Description | <p>This query returns the same data as the “:TRAC:DATA?” query, but in IEEE binary block format.</p> <ul style="list-style-type: none"> • <segment_id> – id of the segment • <offset> – offset from segment start in samples (marker samples do not count) to allow splitting the transfer in smaller portions • <length> – number of samples to read |
| Example | :TRAC:DATA:BLOC? 1, 0, 2560 |

6.20.10 :TRAC[1|2|3|4]:IMP

| | |
|--------------------|--|
| Command | :TRAC[1 2 3 4]:IMP |
| Long | :TRACe[1 2 3 4]:IMPort |
| Parameters | <segment_id>,<file_name>,TXT BIN BIN8 IQBIN BIN6030 BIN5110 MAT89600 DSA90000 CSV,IONLY QONLY,ON OFF 1 0,[,ALENgh F ILL],[<init_value>][,<ignore_header_parameters>] |
| File Format | <p>Import segment data from a file. Different file formats are supported. An already existing segment can be filled, or a new segment can be created. This can be used to import real waveform data as well as complex I/Q data.</p> <p><segment_id> the number of the segment, into which the data will be written</p> <p><file_name> the complete path of the file.</p> <p>TXT BIN BIN8 IQBIN BIN6030 BIN5110 MAT89600 DSA90000 CSV. (See File Type)</p> <p><data_type> IONLY QONLY. This parameter is only used, if the file contains I/Q data.</p> <p><marker_flag> ON OFF 1 0 (See Marker Flag)</p> <p><padding> ALENgh FILL (See Padding)</p> <p><init_value> – optional initialization value. For non-I/Q format this is a DAC value. For I/Q file format this is the I-part or Q-part of an I/Q sample pair in binary format (int8). Defaults to 0 if not specified.</p> <p><ignore_header_parameters> ON OFF 1 0 (See Ignore Header Parameters)</p> |

File Type**TXT**

Compatibility: Keysight M8190A, Tek AWG 7000

One file contains waveform samples for one M8195A channel as normalized values (-1.0 .. +1.0) and (optionally marker values) separated by ';' or ':' or '\t'. Not given marker values are just set to 0. Space ' ' and '\t' are ignored. Line end can be '\r' or '\r\n'. The waveform samples can be imported to any of the four M8195A channels.

Example (US locale)

0,7,0,1

0,9,1

Example (German locale):

0,7;0;1

0,9;1

NOTE

In German locale it is recommended (but not required) to use ';' or '\t' as separator. But it must then be ensured that the double really has a decimal point (',') or there is some space inserted to ensure correct parsing:

0,7,0,1

0 ,0,1

BIN

Compatibility: Keysight M8190A.

One file contains waveform samples for one channel. The waveform samples can be imported to any of the four M8195A channels. Samples consist of binary int16 values (in little endian byte order).

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|------|------|------|------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|------|
| DB13 | DB12 | DB11 | DB10 | DB9 | DB8 | DB7 | DB6 | DB5 | DB4 | DB3 | DB2 | DB1 | DB0 | SYNM | SMPM |

When imported the MSBs DB13 to DB6 are used as 8-bit sample values. All other bits including marker bits SYNM and SMPM are ignored.

BIN8

BIN8 is the most memory efficient file format for the M8195A without digital markers. As a result, the fastest file download can be achieved

One file contains waveform samples for one channel. The waveform samples can be imported to any of the four M8195A channels. Samples consist of binary int8 values:

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-----|-----|-----|-----|-----|-----|-----|-----|
| DB7 | DB6 | DB5 | DB4 | DB3 | DB2 | DB1 | DB0 |

IQBIN

Compatibility: Keysight M8190A.

One file contains waveform samples for two M8190A channels plus optionally digital marker information.

Channel mapping

I is mapped to channel 1

Q mapped to channel 2

Sample Marker 1 is ignored

Sample Marker 2 is ignored

BIN6030

Compatibility: Keysight N6030

Binary int16 values (in little endian byte order). The Keysight N6030 has 15 bits and uses the most significant digits, ignoring the LSB. While importing, the 8 LSBs are used as sample values, all other bits are ignored.

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|------|------|------|------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|---|---|
| DB13 | DB12 | DB11 | DB10 | DB9 | DB8 | DB7 | DB6 | DB5 | DB4 | DB3 | DB2 | DB1 | DB0 | X | X |

BIN5110

Compatibility: Keysight 5110A

Binary int16 I/Q sample pairs (in little endian byte order). May contain full 16 bit DAC values without the marker bits or 14 bit value plus two markers.

When importing 16 bit values without markers the marker flag should be set to 'OFF' so that 8 LSB are ignored.

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-----|-----|-----|-----|----|----|----|----|----|----|----|----|----|----|---|------|
| I13 | I12 | I11 | I10 | I9 | I8 | I7 | I6 | I5 | I4 | I3 | I2 | I1 | I0 | X | SMPM |
| Q13 | Q12 | Q11 | Q10 | Q9 | Q8 | Q7 | Q6 | Q5 | Q4 | Q3 | Q2 | Q1 | Q0 | X | SYNM |

MAT89600

Compatibility: Keysight 89600 VSA

One file contains waveform samples for one or two or three or four M8195A channels. It is a 89600 VSA recording file in MATLAB binary format (.mat) containing floating point values (without markers). Only MATLAB level 4.0 and 5.0 files are supported.

MATLAB binary files with one, two, three or four columns are supported. If the MATLAB file consists of one column, the data can be imported to channel 1 or channel 2 or channel 3 or channel 4. If it consists of multiple columns, column 1 can be imported to channel 1, column 2 to channel 2, column 3 to channel 3 and column 4 to channel 4.

If it consists of multiple columns, the handling depends on the number of available channels.

Four-channel mode: Column 1 can be imported to channel 1, column 2 to channel 2, column 3 to channel 3 and column 4 to channel 4,

Two-channel mode: Column 1 can be imported to channel 1, column 2 to channel 4. If column 2 is not present, column 4 can be imported to channel 4.

Both real and complex I/Q data formats are supported. For I/Q format the values are stored as array of complex numbers with the real part corresponding to I value and the imaginary part corresponding to Q value. The header variable 'XDelta' (1/XDelta) is used to set the currently selected sample frequency.

DSA90000

Compatibility: Keysight DSA90000 Oscilloscope

One file contains waveform samples for one M8195A channel. The waveform samples can be imported to any of the four M8195A channels.

DSA90000 waveform file in binary format (.bin) containing header and floating point data (without markers). Only waveform type 'Normal' is supported. If the file contains more than one waveform only the first waveform will be imported.

CSV

Compatibility: M8190A

One file contains waveform samples for one, two, three, or four M8195A channels.

Normalized values (-1.0 .. +1.0) and markers in comma delimited format. Without header information, the columns are pre-defined in the following way:

1 column: waveform data for channel 1

2 columns: waveform data for channel 1 & 2

3 columns: waveform data for channel 1 & 2 & 3

4 columns: waveform data for channel 1 & 2 & 3 & 4

If the file consists of one column, the data can be imported to channel 1 or channel 2 or channel 3 or channel 4. If it consists of multiple columns, the handling depends on the number of available channels.

Four-channel mode: Column 1 can be imported to channel 1, column 2 to channel 2, column 3 to channel 3 and column 4 to channel 4,

Two-channel mode: Column 1 can be imported to channel 1, column 2 to channel 4. If column 2 is not present (possible when a data header is used, see below), column 4 can be imported to channel 4.

Examples:

1 channel (without markers):

0.7

0.9

...

2 channel (without markers):

0.7,0.7

0.9,1.0

...

3 channels (without markers):

0.7,0.65,0.36

0.8,0.66,0.35

0.9,0.67,0.34

...

4 channels (without markers):

0.7,0.65,0.36,-0.1

0.8,0.66,0.35;-0.2

0.9,0.67,0.34;-0.33

...

The CSV format may contain optional header information as follows:

Parameter Header

The parameter header contains optional header parameters as name and value pairs separated by '='. Each parameter should be specified in a single line. This header is optional. There are following header parameters:

SampleRate

The sample rate.

SetConfig

Flag to indicate if the header parameters need to be set. This can be set to either 'true' or 'false'. If this flag is 'false' header parameters will not be set. If this flag is omitted header parameters are set.

Data Header

The data header contains the names of the data columns separated by ','. The waveform data are specified after the data header. This header is optional. If this header is not specified the data need to be defined similar to CSV files without the header (see above). The data columns are as follows:

Y1

Waveform data for channel 1.

Y2

Waveform data for channel 2.

Y3

Waveform data for channel 3.

Y4

Waveform data for channel 4.

SampleMarker1

Sample Marker for channel 1.

SampleMarker2

Sample Marker for channel 2.

Notes: If any of the marker columns (SampleMarker1 or Sample Marker2) is present for a channel the data header must contain the waveform data column (Y1 or Y2). It is possible to have only the data columns (Y1 , Y2, Y3, Y4 or any combination) without the marker columns though.

Examples:

SampleRate = 7.2 GHz

Y1, Y2, SampleMarker1, SampleMarker2

0.7,0.7,0,0

0.9,1.0,0,1

0.3,-0.3,1,1

...

Y1, SampleMarker1

0.7,0

0.9,1

0.3,0

...

Y1, SampleMarker1, SampleMarker2

0.7,0,0

0.9,0,1

0.3,1,1

...

Y1, Y2, Y4

0.7,0,65,0.36

0.8,0,66,0.35

0.9,0,67,0.34

...

Marker Flag

This flag is used to specify if the marker data need to be downloaded or not. If this flag is 'OFF' marker data will not be downloaded. Default value is 'ON'. This flag is applicable to BIN5110 format only. If BIN5110 format consists of full 16 bit DAC values (without markers) this flag should be set to 'OFF' so that 2 LSB's are ignored.

| | |
|--------------------------------------|---|
| Padding | ALENgth: Automatic LENgth : <segment_id> may or may not exist. After execution <segment_id> has exactly the length of the pattern in file or a multiple of this length to fulfill granularity and minimum segment length requirements. This behavior is default if <padding> is omitted. FILL: <segment_id> must exist. If pattern in file is larger than the defined segment length, just ignore excessive samples. If pattern in file is smaller than defined segment length, fill remaining samples with <init_value>. <init_value> defaults to 0 if it is not specified. |
| Ignore Header Parameters Flag | This flag is used to specify if the header parameters need to be set. If this flag is 'ON' header parameters will not be set. This flag is optional and the default value is 'OFF' i.e. by default the header parameters are set. This flag is applicable to formats (CSV and MAT89600) that contain header parameters. |
| Example | Command <code>:TRAC1:IMP 1, "C:\Program Files (x86)\Keysight\M8195\Examples\WaveformDataFiles\ Sin10MHzAt64GHz.bin", BIN, IONLY, ON, ALEN</code> |

6.20.11 :TRAC[1|2|3|4]:DEL

| | |
|-------------------------|--|
| Command | :TRAC[1 2 3 4]:DEL |
| Long | :TRACe[1 2 3 4]:DELetE |
| Parameters | <segment_id> – id of the segment |
| Parameter Suffix | None |
| Description | Delete a segment. The command can only be used in program mode. If the channel is sourced from Extended Memory, the same segment is deleted on all other channels sourced from Extended Memory. |
| Example | Command :TRAC:DEL 1 |

6.20.12 :TRAC[1|2|3|4]:DEL:ALL

| | |
|-------------------------|---|
| Command | :TRAC[1 2 3 4]:DEL:ALL |
| Long | :TRACe[1 2 3 4]:DELetE:ALL |
| Parameters | None |
| Parameter Suffix | None |
| Description | Delete all segments. The command can only be used in program mode. If the channel is sourced from Extended Memory, the same segment is deleted on all other channels sourced from Extended Memory. |
| Example | Command :TRAC:DEL:ALL |

6.20.13 :TRAC[1|2|3|4]:CAT?

| | |
|-------------------------|--|
| Command | :TRAC[1 2 3 4]:CAT? |
| Long | :TRACe[1 2 3 4]:CATalog? |
| Parameters | None |
| Parameter Suffix | None |
| Description | The query returns a comma-separated list of segment-ids that are defined and the length of each segment. So first number is a segment id, next length ... If no segment is defined, "0, 0" is returned. |
| Example | Query :TRAC1:CAT? |

6.20.14 :TRAC[1|2|3|4]:FREE?

| | |
|-------------------------|---|
| Command | :TRAC[1 2 3 4]:FREE? |
| Long | :TRACe[1 2 3 4]:FREE? |
| Parameters | None |
| Parameter Suffix | None |
| Description | The query returns the amount of memory space available for waveform data in the following form: <bytes available>, <bytes in use>, < contiguous bytes available>. |
| Example | Query :TRAC:FREE? |

6.20.15 :TRAC[1|2|3|4]:NAME[?]

| | |
|-------------------------|--|
| Command | :TRAC[1 2 3 4]:NAME[?] |
| Long | :TRACe[1 2 3 4]:NAME[?] |
| Parameters | <segment_id>,<name> |
| Parameter Suffix | None |
| Description | This command associates a name to a segment. The query gets the name for a segment. <ul style="list-style-type: none"> • <segment_id> – the number of the segment • <name> – string of at most 32 characters |
| Example | <p>Command :TRAC:NAME 1,"ADY"</p> <p>Query :TRAC:NAME? 1</p> |

6.20.16 :TRAC[1|2|3|4]:COMM[?]

| | |
|-------------------------|--|
| Command | :TRAC[1 2 3 4]:COMM[?] |
| Long | :TRACe[1 2 3 4]:COMMent[?] |
| Parameters | <segment_id>,<comment> |
| Parameter Suffix | None |
| Description | This command associates a comment to a segment. The query gets the comment for a segment. <ul style="list-style-type: none"> • <segment_id> – the number of the segment • <comment> – string of at most 256 characters |
| Example | <p>Command :TRAC:COMM 1, "Comment"</p> <p>Query :TRAC:COMM? 1</p> |

6.20.17 :TRAC[1|2|3|4]:SEL[?]<segment_id>|MINimum|MAXimum

| | |
|-------------------------|---|
| Command | :TRAC[1 2 3 4]:SEL[?] |
| Long | :TRACe[1 2 3 4]:SELect[?] |
| Parameters | <segment_id> MINimum MAXimum |
| Parameter Suffix | None |
| Description | <p>Selects the segment, which is output by the instrument in arbitrary function mode.</p> <ul style="list-style-type: none"> • <segment_id> – the number of the segment <p>The command has only effect, If the channel is sourced from Extended Memory. In this case the same value is used for all other channels sourced from Extended Memory.</p> |
| Example | <p>Command :TRAC:SEL 5</p> <p>Query :TRAC:SEL?</p> |

6.20.18 :TRAC[1|2|3|4]:ADV[?]

| | |
|-------------------------|--|
| Command | :TRAC[1 2 3 4]:ADV[?] |
| Long | :TRACe[1 2 3 4]:ADVance[?] |
| Parameters | AUTO COND REP SING |
| Parameter Suffix | None |
| Description | <p>Use this command or query to set or get the advancement mode for the selected segment. The advancement mode is used, if the segment is played in arbitrary mode.</p> <p>The command has only effect, If the channel is sourced from Extended Memory. In this case the same value is used for all other channels sourced from Extended Memory.</p> |
| Example | <p>Command :TRAC:ADV AUTO</p> <p>Query :TRAC:ADV?</p> |

6.20.19 :TRAC[1|2|3|4]:COUN[?]<count>|MINimum|MAXimum

| | |
|-------------------------|--|
| Command | :TRAC[1 2 3 4]:COUN[?] |
| Long | :TRACe[1 2 3 4]:COUNT[?] |
| Parameters | <count> MINimum MAXimum |
| Parameter Suffix | None |
| Description | <p>Use this command or query to set or get the segment loop count for the selected segment. The segment loop count is used, if the segment is played in arbitrary mode.</p> <p><count> – 1..4G-1: number of times the selected segment is repeated.</p> <p>The command has only effect, If the channel is sourced from Extended Memory. In this case the same value is used for all other channels sourced from Extended Memory.</p> |
| Example | <p>Command :TRAC:COUN 1</p> <p>Query :TRAC:COUN?</p> |

6.20.20 :TRAC[1|2|3|4]:MARK[?]

| | |
|-------------------------|---|
| Command | :TRAC[1 2 3 4]:MARK[?] |
| Long | :TRACe[1 2 3 4]:MARKer[?] |
| Parameters | OFF ON 0 1 |
| Parameter Suffix | None |
| Description | <p>Use this command to enable or disable markers for the selected segment. The query form gets the current marker state.</p> <p>The command has only effect, If the channel is sourced from Extended Memory. In this case the same value is used for all other channels sourced from Extended Memory.</p> |
| Example | <p>Command :TRAC:MARK 1</p> <p>Query :TRAC:MARK?</p> |

6.21 :TEST Subsystem

6.21.1 :TEST:PON?

| | |
|-------------------------|--|
| Command | :TEST:PON? |
| Long | :TEST:PON? |
| Parameters | None |
| Parameter Suffix | None |
| Description | Return the results of the power on self-tests. |
| Example | Query :TEST:PON? |

6.21.2 :TEST:TST?

| | |
|-------------------------|--|
| Command | :TEST:TST? |
| Long | :TEST:TST? |
| Parameters | None |
| Parameter Suffix | None |
| Description | Same as *TST? but the actual test messages are returned. |
| Example | Query :TEST:TST? |

NOTE

Currently same as :TEST:PON?

7 Examples

- 7.1 Introduction / 255
- 7.2 Remote Programming Examples / 255
- 7.3 Example Files for Import / 255
- 7.4 Example Correction Files / 256
- 7.5 Example Custom Modulation Files / 256

7.1 Introduction

In a standard installation the examples can be found in the folder “C:\Program Files (x86)\Keysight\M8195\Examples”.

7.2 Remote Programming Examples

The MATLAB IQtools are described in file “README.docx” in subfolder “MATLAB\iqtools”. The C++, C# and VB programs are provided as Visual Studio 2008 solutions. However, they can be easily converted to more recent Visual Studio versions. They show how to connect to the AWG, write a sine wave into the memory and start signal generation. They use the VISA or VISA-COM libraries.

7.3 Example Files for Import

The subfolder “WaveformDataFiles” contains examples for all supported import file formats. To import them use either the SFP Import Waveform panel or the SCPI command TRAC[1|2|3|4]:IMP.

7.4 Example Correction Files

The subfolder “CorrectionFiles” contains examples to be used in the SFP Multi-Tone and Complex Modulation panels.

7.5 Example Custom Modulation Files

The subfolder “CustomModulationFiles” contains examples to be used in the SFP Complex Modulation panel.

8 Appendix

8.1 Resampling Algorithms for Waveform Import / 257

8.1 Resampling Algorithms for Waveform Import

8.1.1 Resampling Requirements

Resampling is typically associated to a series of processes applied to a waveform sampled at a given sampling frequency to generate a new waveform with a different sampling rate while preserving all the original information contained in the signal within the Nyquist bandwidth corresponding to the output sampling rate. Processes involved in resampling may vary depending on the output to input sampling rate ratio (or resampling factor) and the integer nature of the ratio itself. Resampling calculations, when applied to arbitrary waveform generation, must meet additional constraints such as available record length boundaries, record length granularity requirements, or acceptable sampling rate range.

Typically, the characteristics of the input waveform (sampling rate, record length) are externally defined (i.e. by the horizontal settings of an oscilloscope used to capture the waveform). Users may be interested in resampling the signal to adapt the input waveform to the AWG requirements or the user desires. In some cases, it may be necessary to reduce the sampling rate if it has been captured at a higher sampling rate than the one allowed by the AWG or to reduce the record length required to generate it. The opposite is also true as oversampling may help to “smooth” the signal as increasing sampling rate will shift the images created by the DAC to a higher frequency. Finally, resampling may be also necessary to adapt the record length of the input waveform to a legal record length that can be applied to a real AWG (i.e. to meet the record length granularities) without applying truncation or “zero padding” to the input waveform.

8.1.2 Resampling Methodology

Generally speaking, resampling factors do not have to be an integer or a simple fractional ratio. Because of that, traditional methods based in upsampling/filtering/decimation techniques may not be suitable given the amount of calculations resulting from the typical input waveform sizes involved. Instead of this, a more straight forward approach has been chosen. This approach is based in the following principles:

Only output samples will be calculated so there is not any up-sampling and/or down-sampling operations involved.

Filtering calculations will be kept to a minimum by using a filter with a fast enough roll-off and sufficient stop band attenuation according to the target AWG dynamic range.

Interpolation filter and anti-alias filters are exactly the same although the filter parameters will depend on the resampling parameters.

The implemented algorithm does perform filtering and interpolation simultaneously so the number of calculations is greatly reduced. Additionally, filters are implemented as look-up tables so those are calculated only once during the process.

Timing parameters are based in double precision floating-point numbers while amplitude related parameters are single precision numbers. Most calculations consist in multiplication/addition operations required by convolution processes and only involve amplitude related variables (input samples and filter coefficients). Single precision numbers will minimize calculation time while offering more than enough dynamic range.

Interpolators and anti-aliasing filters share most characteristics as they are required to be low-pass with good flatness, linear phase, fast roll-off, and high stop-band rejections ratio. Ideal interpolator filters show a “brick-wall” response. However, such filters require a very long “sinc-like” impulse response to obtain good-enough performance. Impulse response length has a direct effect on calculation times resulting of applying the filter. Roll-off characteristics are especially important when applying the filter as the anti-alias filter required for down-sampling. The filter implemented in these algorithms has been designed with the following objectives:

Pass band flatness better than 0.01 dB

Stop band attenuation better than 80 dB

F80dB/F3dB ratio better than 1.15

The final filter consists in a sinc signal with a 41 sample periods length after applying a Blackman-Harris time-domain window.

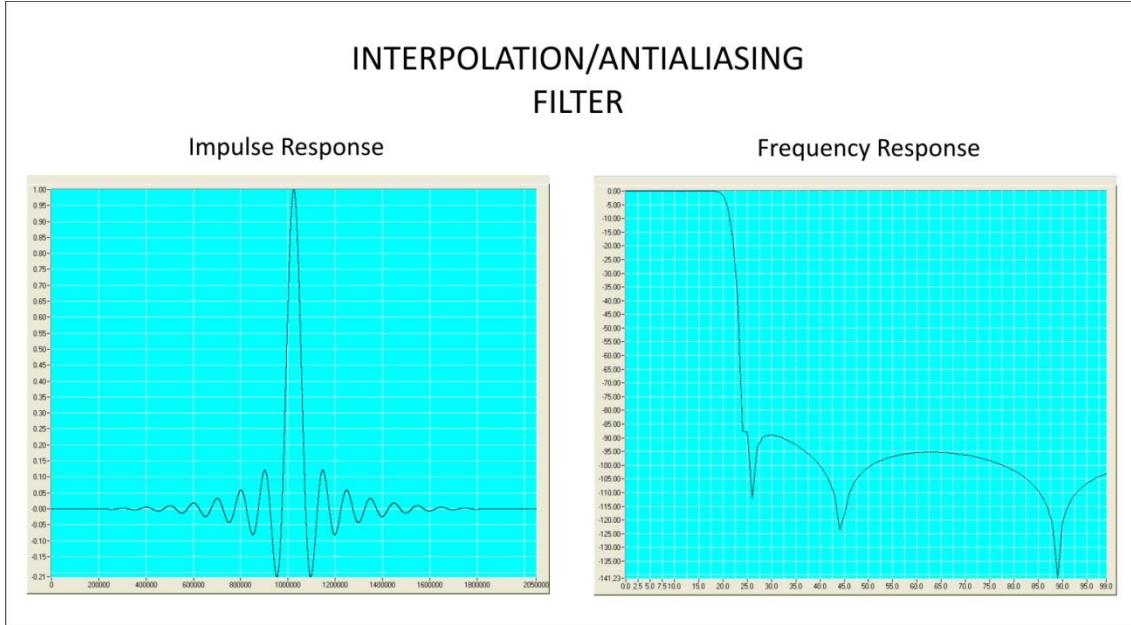


Figure 62: Interpolation/Antialiasing filter

The filter shape remains the same no matter the resampling characteristics. For resampling ratios greater than 1.0, filter will implement an interpolator so nulls in the impulse response must be located at multiples of the sampling period of the input signal. For ratios lower than 1.0 the filter will implement an antialiasing filter. In this case, distance between nulls will have to be longer than the output waveform sampling period so the filter reach the required attenuation (>80 dB) at the output signal Nyquist frequency. For the implemented filter this is accomplished by choosing 0.89 ratio between the output sampling period and the distance between consecutive nulls in the filter's impulse response.

This information is subject to change without notice.

© Keysight Technologies 2016
Edition 4.0, April 2016



M8195-91040

www.keysight.com

