

VAGEN: verilogA generator written in python

Generated by Doxygen 1.9.1

1 VAGEN - VerilogA generator	1
1.1 Description	1
1.2 License	1
2 Namespace Index	3
2.1 Namespace List	3
3 Hierarchical Index	5
3.1 Class Hierarchy	5
4 Class Index	7
4.1 Class List	7
5 File Index	9
5.1 File List	9
6 Namespace Documentation	11
6.1 vagen Namespace Reference	11
6.1.1 Detailed Description	11
7 Class Documentation	13
7.1 vagen.veriloga.Above Class Reference	13
7.1.1 Detailed Description	14
7.1.2 Constructor & Destructor Documentation	14
7.1.2.1 __init__()	14
7.2 vagen.veriloga.Block Class Reference	14
7.2.1 Detailed Description	15
7.2.2 Constructor & Destructor Documentation	15
7.2.2.1 __init__()	15
7.2.3 Member Function Documentation	16
7.2.3.1 getHeader()	16
7.2.3.2 getVA()	16
7.3 vagen.veriloga.Bool Class Reference	16
7.3.1 Detailed Description	17
7.3.2 Constructor & Destructor Documentation	17
7.3.2.1 __init__()	17
7.3.3 Member Function Documentation	18
7.3.3.1 __and__()	18
7.3.3.2 __eq__()	18
7.3.3.3 __invert__()	18
7.3.3.4 __ne__()	18
7.3.3.5 __or__()	20
7.3.3.6 __rand__()	20
7.3.3.7 __ror__()	20

7.3.3.8 __rxor__()	21
7.3.3.9 __str__()	21
7.3.3.10 __xor__()	21
7.3.3.11 getValue()	21
7.4 vagen.veriloga.BoolVar Class Reference	22
7.4.1 Detailed Description	23
7.4.2 Constructor & Destructor Documentation	23
7.4.2.1 __init__()	23
7.4.3 Member Function Documentation	23
7.4.3.1 eq()	23
7.4.3.2 toggle()	23
7.5 vagen.veriloga.Branch Class Reference	24
7.5.1 Detailed Description	24
7.5.2 Constructor & Destructor Documentation	24
7.5.2.1 __init__()	24
7.6 vagen.hilevelmod.Bus Class Reference	25
7.6.1 Detailed Description	26
7.6.2 Constructor & Destructor Documentation	26
7.6.2.1 __init__()	26
7.6.3 Member Function Documentation	26
7.6.3.1 __getitem__()	26
7.6.3.2 append()	26
7.7 vagen.veriloga.CaseClass Class Reference	27
7.7.1 Detailed Description	28
7.7.2 Constructor & Destructor Documentation	28
7.7.2.1 __init__()	28
7.7.3 Member Function Documentation	28
7.7.3.1 append()	28
7.7.3.2 getBlockList()	28
7.7.3.3 getVA()	29
7.8 vagen.hilevelmod.Clock Class Reference	29
7.8.1 Detailed Description	29
7.8.2 Constructor & Destructor Documentation	29
7.8.2.1 __init__()	30
7.8.3 Member Function Documentation	30
7.8.3.1 off()	30
7.8.3.2 on()	30
7.9 vagen.veriloga.Cmd Class Reference	30
7.9.1 Detailed Description	31
7.9.2 Constructor & Destructor Documentation	31
7.9.2.1 __init__()	31
7.9.3 Member Function Documentation	31

7.9.3.1 __str__()	32
7.9.3.2 getVA()	32
7.10 vagen.veriloga.CmdList Class Reference	32
7.10.1 Detailed Description	33
7.10.2 Constructor & Destructor Documentation	33
7.10.2.1 __init__()	33
7.10.3 Member Function Documentation	33
7.10.3.1 __str__()	34
7.10.3.2 append()	34
7.10.3.3 flat()	34
7.10.3.4 getVA()	34
7.11 vagen.veriloga.Cond Class Reference	35
7.11.1 Detailed Description	36
7.11.2 Constructor & Destructor Documentation	36
7.11.2.1 __init__()	36
7.11.3 Member Function Documentation	36
7.11.3.1 append()	36
7.11.3.2 Else()	36
7.11.3.3 getBlock()	37
7.11.3.4 getCond()	37
7.11.3.5 getVA()	37
7.12 vagen.veriloga.Cross Class Reference	38
7.12.1 Detailed Description	38
7.12.2 Constructor & Destructor Documentation	38
7.12.2.1 __init__()	39
7.13 vagen.hilevelmod.DigBusIn Class Reference	39
7.13.1 Detailed Description	40
7.13.2 Constructor & Destructor Documentation	40
7.13.2.1 __init__()	40
7.13.3 Member Function Documentation	40
7.13.3.1 read()	41
7.14 vagen.hilevelmod.DigBusInOut Class Reference	41
7.14.1 Detailed Description	42
7.14.2 Constructor & Destructor Documentation	42
7.14.2.1 __init__()	42
7.14.3 Member Function Documentation	43
7.14.3.1 hiZ()	43
7.14.3.2 lowZ()	43
7.15 vagen.hilevelmod.DigBusOut Class Reference	43
7.15.1 Detailed Description	45
7.15.2 Constructor & Destructor Documentation	45
7.15.2.1 __init__()	45

7.15.3 Member Function Documentation	45
7.15.3.1 setDelay()	45
7.15.3.2 setRiseFall()	45
7.15.3.3 write()	46
7.16 vagen.hilevelmod.DigIn Class Reference	46
7.16.1 Detailed Description	47
7.16.2 Constructor & Destructor Documentation	47
7.16.2.1 __init__()	47
7.16.3 Member Function Documentation	48
7.16.3.1 read()	48
7.17 vagen.hilevelmod.DigInOut Class Reference	48
7.17.1 Detailed Description	50
7.17.2 Constructor & Destructor Documentation	50
7.17.2.1 __init__()	50
7.17.3 Member Function Documentation	50
7.17.3.1 hiZ()	50
7.17.3.2 lowZ()	51
7.18 vagen.hilevelmod.DigOut Class Reference	51
7.18.1 Detailed Description	52
7.18.2 Constructor & Destructor Documentation	52
7.18.2.1 __init__()	52
7.18.3 Member Function Documentation	53
7.18.3.1 setDelay()	53
7.18.3.2 setRiseFall()	53
7.18.3.3 write()	54
7.19 vagen.veriloga.Electrical Class Reference	54
7.19.1 Detailed Description	55
7.19.2 Constructor & Destructor Documentation	55
7.19.2.1 __init__()	55
7.19.3 Member Function Documentation	55
7.19.3.1 getName()	55
7.19.3.2 iAttr()	55
7.19.3.3 iCont()	56
7.19.3.4 iInd()	56
7.19.3.5 vAttr()	56
7.19.3.6 vCont()	57
7.19.3.7 vInd()	57
7.20 vagen.veriloga.Event Class Reference	57
7.20.1 Detailed Description	58
7.20.2 Constructor & Destructor Documentation	58
7.20.2.1 __init__()	58
7.20.3 Member Function Documentation	59

7.20.3.1 <code>__or__()</code>	59
7.20.3.2 <code>__str__()</code>	59
7.21 <code>vagen.veriloga.FinalStep</code> Class Reference	59
7.21.1 Detailed Description	60
7.21.2 Constructor & Destructor Documentation	60
7.21.2.1 <code>__init__()</code>	60
7.22 <code>vagen.veriloga.ForLoop</code> Class Reference	60
7.22.1 Detailed Description	62
7.22.2 Constructor & Destructor Documentation	62
7.22.2.1 <code>__init__()</code>	62
7.22.3 Member Function Documentation	62
7.22.3.1 <code>getCond()</code>	62
7.22.3.2 <code>getInc()</code>	63
7.22.3.3 <code>getStart()</code>	63
7.23 <code>vagen.hilevelmod.HiLevelMod</code> Class Reference	63
7.23.1 Detailed Description	65
7.23.2 Constructor & Destructor Documentation	65
7.23.2.1 <code>__init__()</code>	65
7.23.3 Member Function Documentation	65
7.23.3.1 <code>clock()</code>	65
7.23.3.2 <code>dig()</code>	66
7.23.3.3 <code>getEqs()</code>	66
7.23.3.4 <code>getOcn()</code>	67
7.23.3.5 <code>idc()</code>	67
7.23.3.6 <code>marker()</code>	67
7.23.3.7 <code>seq()</code>	68
7.23.3.8 <code>seqNested()</code>	68
7.23.3.9 <code>smu()</code>	68
7.23.3.10 <code>sw()</code>	69
7.23.3.11 <code>var()</code>	69
7.23.3.12 <code>vdc()</code>	70
7.24 <code>vagen.hilevelmod.Idc</code> Class Reference	70
7.24.1 Detailed Description	71
7.24.2 Constructor & Destructor Documentation	71
7.24.2.1 <code>__init__()</code>	71
7.24.3 Member Function Documentation	72
7.24.3.1 <code>applyl()</code>	72
7.24.3.2 <code>setRiseFall()</code>	72
7.25 <code>vagen.hilevelmod.IdcBus</code> Class Reference	72
7.25.1 Detailed Description	73
7.25.2 Constructor & Destructor Documentation	74
7.25.2.1 <code>__init__()</code>	74

7.25.3 Member Function Documentation	74
7.25.3.1 applyI()	74
7.25.3.2 setRiseFall()	74
7.26 vagen.veriloga.InitialStep Class Reference	75
7.26.1 Detailed Description	75
7.26.2 Constructor & Destructor Documentation	75
7.26.2.1 __init__()	75
7.27 vagen.veriloga.Integer Class Reference	76
7.27.1 Detailed Description	78
7.27.2 Constructor & Destructor Documentation	78
7.27.2.1 __init__()	78
7.27.3 Member Function Documentation	78
7.27.3.1 __abs__()	78
7.27.3.2 __add__()	78
7.27.3.3 __and__()	79
7.27.3.4 __eq__()	79
7.27.3.5 __ge__()	79
7.27.3.6 __gt__()	79
7.27.3.7 __invert__()	80
7.27.3.8 __le__()	80
7.27.3.9 __lshift__()	80
7.27.3.10 __lt__()	81
7.27.3.11 __mod__()	81
7.27.3.12 __mul__()	81
7.27.3.13 __ne__()	81
7.27.3.14 __neg__()	82
7.27.3.15 __or__()	82
7.27.3.16 __pos__()	82
7.27.3.17 __pow__()	83
7.27.3.18 __radd__()	83
7.27.3.19 __rand__()	83
7.27.3.20 __rlshift__()	83
7.27.3.21 __rmod__()	84
7.27.3.22 __rmul__()	84
7.27.3.23 __ror__()	84
7.27.3.24 __rpow__()	85
7.27.3.25 __rrshift__()	85
7.27.3.26 __rshift__()	85
7.27.3.27 __rsub__()	85
7.27.3.28 __rtruediv__()	86
7.27.3.29 __rxor__()	86
7.27.3.30 __str__()	86

7.27.3.31 <code>__sub__()</code>	87
7.27.3.32 <code>__truediv__()</code>	87
7.27.3.33 <code>__xor__()</code>	87
7.27.3.34 <code>getValue()</code>	87
7.28 <code>vagen.veriloga.IntegerVar</code> Class Reference	88
7.28.1 Detailed Description	89
7.28.2 Constructor & Destructor Documentation	89
7.28.2.1 <code>__init__()</code>	89
7.28.3 Member Function Documentation	89
7.28.3.1 <code>dec()</code>	89
7.28.3.2 <code>eq()</code>	89
7.28.3.3 <code>inc()</code>	90
7.29 <code>vagen.hilevelmod.Mark</code> Class Reference	90
7.29.1 Detailed Description	91
7.29.2 Constructor & Destructor Documentation	91
7.29.2.1 <code>__init__()</code>	91
7.29.3 Member Function Documentation	91
7.29.3.1 <code>__str__()</code>	91
7.29.3.2 <code>getCmd()</code>	91
7.29.3.3 <code>getVA()</code>	92
7.30 <code>vagen.hilevelmod.Marker</code> Class Reference	92
7.30.1 Detailed Description	93
7.30.2 Constructor & Destructor Documentation	93
7.30.2.1 <code>__init__()</code>	93
7.30.3 Member Function Documentation	93
7.30.3.1 <code>getEqs()</code>	93
7.30.3.2 <code>getName()</code>	93
7.30.3.3 <code>high()</code>	94
7.30.3.4 <code>low()</code>	94
7.30.3.5 <code>mark()</code>	94
7.31 <code>vagen.veriloga.Module</code> Class Reference	95
7.31.1 Detailed Description	96
7.31.2 Constructor & Destructor Documentation	96
7.31.2.1 <code>__init__()</code>	96
7.31.3 Member Function Documentation	96
7.31.3.1 <code>addNode()</code>	96
7.31.3.2 <code>analog()</code>	97
7.31.3.3 <code>beginningAnalog()</code>	97
7.31.3.4 <code>electrical()</code>	97
7.31.3.5 <code>endAnalog()</code>	97
7.31.3.6 <code>fixName()</code>	98
7.31.3.7 <code>getModuleName()</code>	98

7.31.3.8 getVA()	98
7.31.3.9 par()	98
7.31.3.10 var()	99
7.32 vagen.veriloga.Real Class Reference	99
7.32.1 Detailed Description	101
7.32.2 Constructor & Destructor Documentation	101
7.32.2.1 __init__()	101
7.32.3 Member Function Documentation	101
7.32.3.1 __abs__()	101
7.32.3.2 __add__()	101
7.32.3.3 __eq__()	102
7.32.3.4 __ge__()	102
7.32.3.5 __gt__()	102
7.32.3.6 __le__()	102
7.32.3.7 __lt__()	103
7.32.3.8 __mul__()	103
7.32.3.9 __ne__()	103
7.32.3.10 __neg__()	104
7.32.3.11 __pos__()	104
7.32.3.12 __pow__()	104
7.32.3.13 __radd__()	104
7.32.3.14 __rmul__()	105
7.32.3.15 __rpow__()	105
7.32.3.16 __rsub__()	105
7.32.3.17 __rtruediv__()	106
7.32.3.18 __str__()	106
7.32.3.19 __sub__()	106
7.32.3.20 __truediv__()	106
7.32.3.21 getValue()	108
7.33 vagen.veriloga.RealVar Class Reference	108
7.33.1 Detailed Description	109
7.33.2 Constructor & Destructor Documentation	109
7.33.2.1 __init__()	109
7.33.3 Member Function Documentation	109
7.33.3.1 eq()	109
7.34 vagen.veriloga.RepeatLoop Class Reference	110
7.34.1 Detailed Description	111
7.34.2 Constructor & Destructor Documentation	111
7.34.2.1 __init__()	111
7.34.3 Member Function Documentation	112
7.34.3.1 getN()	112
7.35 vagen.hilevelmod.Smu Class Reference	112

7.35.1 Detailed Description	113
7.35.2 Constructor & Destructor Documentation	113
7.35.2.1 <code>__init__()</code>	113
7.35.3 Member Function Documentation	114
7.35.3.1 <code>applyI()</code>	114
7.35.3.2 <code>applyR()</code>	114
7.35.3.3 <code>applyV()</code>	115
7.36 <code>vagen.hilevelmod.SmuBus</code> Class Reference	115
7.36.1 Detailed Description	116
7.36.2 Constructor & Destructor Documentation	116
7.36.2.1 <code>__init__()</code>	116
7.36.3 Member Function Documentation	117
7.36.3.1 <code>applyI()</code>	117
7.36.3.2 <code>applyR()</code>	117
7.36.3.3 <code>applyV()</code>	117
7.37 <code>vagen.hilevelmod.Sw</code> Class Reference	118
7.37.1 Detailed Description	118
7.37.2 Constructor & Destructor Documentation	118
7.37.2.1 <code>__init__()</code>	118
7.37.3 Member Function Documentation	119
7.37.3.1 <code>setCond()</code>	119
7.37.3.2 <code>setRiseFall()</code>	119
7.38 <code>vagen.veriloga.Timer</code> Class Reference	119
7.38.1 Detailed Description	120
7.38.2 Constructor & Destructor Documentation	120
7.38.2.1 <code>__init__()</code>	120
7.39 <code>vagen.hilevelmod.Vdc</code> Class Reference	121
7.39.1 Detailed Description	122
7.39.2 Constructor & Destructor Documentation	122
7.39.2.1 <code>__init__()</code>	122
7.39.3 Member Function Documentation	122
7.39.3.1 <code>applyV()</code>	122
7.39.3.2 <code>setRiseFall()</code>	123
7.40 <code>vagen.hilevelmod.VdcBus</code> Class Reference	123
7.40.1 Detailed Description	124
7.40.2 Constructor & Destructor Documentation	124
7.40.2.1 <code>__init__()</code>	124
7.40.3 Member Function Documentation	124
7.40.3.1 <code>applyV()</code>	125
7.40.3.2 <code>setRiseFall()</code>	125
7.41 <code>vagen.veriloga.WaitAnalogEvent</code> Class Reference	125
7.41.1 Detailed Description	127

7.41.2 Constructor & Destructor Documentation	127
7.41.2.1 <code>__init__()</code>	127
7.42 <code>vagen.hilevelmod.WaitSignal</code> Class Reference	127
7.42.1 Detailed Description	128
7.42.2 Constructor & Destructor Documentation	128
7.42.2.1 <code>__init__()</code>	128
7.42.3 Member Function Documentation	128
7.42.3.1 <code>__str__()</code>	129
7.42.3.2 <code>getEvtnt()</code>	129
7.42.3.3 <code>getVA()</code>	129
7.43 <code>vagen.hilevelmod.WaitUs</code> Class Reference	129
7.43.1 Detailed Description	130
7.43.2 Constructor & Destructor Documentation	130
7.43.2.1 <code>__init__()</code>	131
7.43.3 Member Function Documentation	131
7.43.3.1 <code>__str__()</code>	131
7.43.3.2 <code>getDelay()</code>	131
7.43.3.3 <code>getVA()</code>	131
7.44 <code>vagen.veriloga.WhileLoop</code> Class Reference	132
7.44.1 Detailed Description	133
7.44.2 Constructor & Destructor Documentation	133
7.44.2.1 <code>__init__()</code>	133
7.44.3 Member Function Documentation	134
7.44.3.1 <code>getCond()</code>	134
8 File Documentation	135
8.1 <code>hilevelmod.py</code> File Reference	135
8.1.1 Detailed Description	136
8.1.2 License	136
8.2 <code>veriloga.py</code> File Reference	136
8.2.1 Detailed Description	140
8.2.2 License	140
8.2.3 Function Documentation	141
8.2.3.1 <code>absDelay()</code>	141
8.2.3.2 <code>acos()</code>	141
8.2.3.3 <code>acosh()</code>	141
8.2.3.4 <code>acStim()</code>	142
8.2.3.5 <code>analysis()</code>	142
8.2.3.6 <code>asin()</code>	142
8.2.3.7 <code>asinh()</code>	143
8.2.3.8 <code>At()</code>	143
8.2.3.9 <code>atan()</code>	143

8.2.3.10 atan2()	143
8.2.3.11 atanh()	144
8.2.3.12 block()	144
8.2.3.13 blockComment()	144
8.2.3.14 BoundStep()	145
8.2.3.15 Case()	145
8.2.3.16 ceil()	145
8.2.3.17 checkBool()	145
8.2.3.18 checkInstance()	146
8.2.3.19 checkInteger()	146
8.2.3.20 checkNotInstance()	146
8.2.3.21 checkNumber()	147
8.2.3.22 checkReal()	147
8.2.3.23 checkType()	147
8.2.3.24 cos()	148
8.2.3.25 cosh()	148
8.2.3.26 ddt()	148
8.2.3.27 Discontinuity()	148
8.2.3.28 exp()	149
8.2.3.29 expDistInt()	149
8.2.3.30 expDistReal()	149
8.2.3.31 Fclose()	149
8.2.3.32 Finish()	150
8.2.3.33 floor()	150
8.2.3.34 Fopen()	150
8.2.3.35 For()	150
8.2.3.36 Fstroke()	151
8.2.3.37 Fwrite()	151
8.2.3.38 gaussDistInt()	151
8.2.3.39 gaussDistReal()	152
8.2.3.40 hypot()	152
8.2.3.41 idt()	152
8.2.3.42 If()	153
8.2.3.43 lastCrossing()	153
8.2.3.44 limexp()	153
8.2.3.45 ln()	154
8.2.3.46 log()	154
8.2.3.47 parseBool()	154
8.2.3.48 parseInteger()	154
8.2.3.49 parseNumber()	155
8.2.3.50 parseReal()	155
8.2.3.51 poissonDistInt()	155

8.2.3.52 poissonDistReal()	155
8.2.3.53 random()	156
8.2.3.54 Repeat()	156
8.2.3.55 sin()	156
8.2.3.56 sinh()	157
8.2.3.57 slew()	157
8.2.3.58 sqrt()	157
8.2.3.59 Strobe()	157
8.2.3.60 tan()	158
8.2.3.61 tanh()	158
8.2.3.62 ternary()	158
8.2.3.63 transition()	159
8.2.3.64 uDistInt()	159
8.2.3.65 uDistReal()	159
8.2.3.66 unfoldParams()	160
8.2.3.67 unfoldSimTypes()	160
8.2.3.68 While()	160
8.2.3.69 Write()	160
8.2.4 Variable Documentation	161
8.2.4.1 anaTypes	161
8.2.4.2 temp	161
Index	163

Chapter 1

VAGEN - VerilogA generator

1.1 Description

vagen is a verilogA generator that can be used for transient verification and modeling of complex analog IPs. It provides a wide range of basic models with voltage sources, current sources, digital interfaces, clocks, switches, and source measure units that you can use to build your model upon.

1.2 License

Author

Rodrigo Pedroso Mendes

Version

V1.0

Date

14/02/23 13:37:31

Copyright (c) 2023 Rodrigo Pedroso Mendes

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Chapter 2

Namespace Index

2.1 Namespace List

Here is a list of all documented namespaces with brief descriptions:

vagen	VerilogA generator	11
-----------------------	------------------------------	----

Chapter 3

Hierarchical Index

3.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

vagen.veriloga.Bool	16
vagen.veriloga.BoolVar	22
vagen.hilevelmod.Clock	29
vagen.veriloga.Cmd	30
vagen.hilevelmod.Mark	90
vagen.hilevelmod.WaitSignal	127
vagen.hilevelmod.WaitUs	129
vagen.veriloga.CaseClass	27
vagen.veriloga.CmdList	32
vagen.veriloga.Block	14
vagen.veriloga.ForLoop	60
vagen.veriloga.RepeatLoop	110
vagen.veriloga.WaitAnalogEvent	125
vagen.veriloga.WhileLoop	132
vagen.veriloga.Cond	35
vagen.veriloga.Electrical	54
vagen.hilevelmod.DigIn	46
vagen.hilevelmod.DigInOut	48
vagen.hilevelmod.DigOut	51
vagen.hilevelmod.DigInOut	48
vagen.hilevelmod.Idc	70
vagen.hilevelmod.Smu	112
vagen.hilevelmod.Vdc	121
vagen.veriloga.Branch	24
vagen.veriloga.Event	57
vagen.veriloga.Above	13
vagen.veriloga.Cross	38
vagen.veriloga.FinalStep	59
vagen.veriloga.InitialStep	75
vagen.veriloga.Timer	119
vagen.veriloga.Integer	76
vagen.veriloga.IntegerVar	88
list	

vagen.hilevelmod.Bus	25
vagen.hilevelmod.DigBusIn	39
vagen.hilevelmod.DigBusInOut	41
vagen.hilevelmod.DigBusOut	43
vagen.hilevelmod.DigBusInOut	41
vagen.hilevelmod.IdcBus	72
vagen.hilevelmod.SmuBus	115
vagen.hilevelmod.VdcBus	123
vagen.veriloga.CmdList	32
vagen.hilevelmod.Marker	92
vagen.veriloga.Module	95
vagen.hilevelmod.HiLevelMod	63
vagen.veriloga.Real	99
vagen.veriloga.RealVar	108
vagen.hilevelmod.Sw	118

Chapter 4

Class Index

4.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

vagen.veriloga.Above	
Above Class	13
vagen.veriloga.Block	
Command Block Class	14
vagen.veriloga.Bool	
Class of Bool operators	16
vagen.veriloga.BoolVar	
Boolean variable class	22
vagen.veriloga.Branch	
Branch class	24
vagen.hilevelmod.Bus	
Bus class	25
vagen.veriloga.CaseClass	
Condition Class	27
vagen.hilevelmod.Clock	
Clock class	29
vagen.veriloga.Cmd	
Command class	30
vagen.veriloga.CmdList	
Command List class	32
vagen.veriloga.Cond	
Condition Class	35
vagen.veriloga.Cross	
Cross Class	38
vagen.hilevelmod.DigBusIn	
DigBusIn class	39
vagen.hilevelmod.DigBusInOut	
DigBusInOut class	41
vagen.hilevelmod.DigBusOut	
DigBusOut class	43
vagen.hilevelmod.DigIn	
DigIn class	46
vagen.hilevelmod.DigInOut	
DigInOut class	48
vagen.hilevelmod.DigOut	
DigOut class	51

vagen.veriloga.Electrical	
Class of electrical signals	54
vagen.veriloga.Event	
Class of events	57
vagen.veriloga.FinalStep	
FinalStep class	59
vagen.veriloga.ForLoop	
ForLoop class	60
vagen.hilevelmod.HiLevelMod	
HiLevelMod class	63
vagen.hilevelmod.Idc	
Idc class	70
vagen.hilevelmod.IdcBus	
IdcBus class	72
vagen.veriloga.InitialStep	
InitialStep class	75
vagen.veriloga.Integer	
Class of Integer operators	76
vagen.veriloga.IntegerVar	
Integer variable class	88
vagen.hilevelmod.Mark	
Mark command class	90
vagen.hilevelmod.Marker	
Marker class	92
vagen.veriloga.Module	
VerilogA class	95
vagen.veriloga.Real	
Class of Real operators	99
vagen.veriloga.RealVar	
Real variable class	108
vagen.veriloga.RepeatLoop	
RepeatLoop class	
110	
vagen.hilevelmod.Smu	
Smu class	112
vagen.hilevelmod.SmuBus	
SmuBus class	115
vagen.hilevelmod.Sw	
Sw class	118
vagen.veriloga.Timer	
Timer Class	119
vagen.hilevelmod.Vdc	
Vdc class	121
vagen.hilevelmod.VdcBus	
VdcBus class	123
vagen.veriloga.WaitAnalogEvent	
Wait analog event class	125
vagen.hilevelmod.WaitSignal	
WaitSignal command class	127
vagen.hilevelmod.WaitUs	
WaitUs command class	129
vagen.veriloga.WhileLoop	
WhileLoop class	132

Chapter 5

File Index

5.1 File List

Here is a list of all documented files with brief descriptions:

hilevelmod.py	
Hi level modeling	135
veriloga.py	
VerilogA modeling	136

Chapter 6

Namespace Documentation

6.1 vagen Namespace Reference

VerilogA generator.

6.1.1 Detailed Description

VerilogA generator.

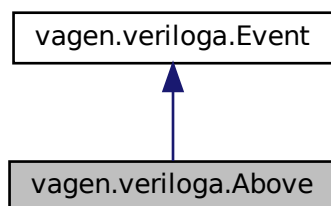
Chapter 7

Class Documentation

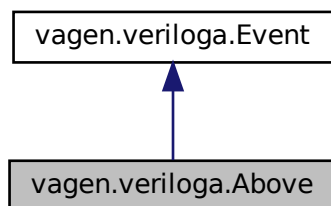
7.1 vagen.veriloga.Above Class Reference

[Above](#) Class.

Inheritance diagram for vagen.veriloga.Above:



Collaboration diagram for vagen.veriloga.Above:



Public Member Functions

- `def __init__(self, signal, threshold, *pars)`
Constructor.

Additional Inherited Members

7.1.1 Detailed Description

[Above](#) Class.

7.1.2 Constructor & Destructor Documentation

7.1.2.1 __init__()

```
def vagen.veriloga.Above.__init__ (
    self,
    signal,
    threshold,
    * pars )
```

Constructor.

Parameters

<i>self</i>	object pointer
<i>signal</i>	Real class or build-in real representing the signal
<i>threshold</i>	Real class or build-in real representing the threshold that must be crossed
<i>*pars</i>	optional Real or build-in real parameters timeTol and expTol in this order

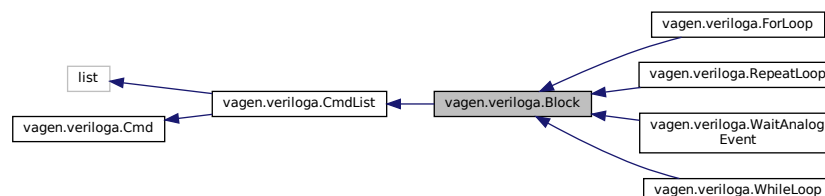
The documentation for this class was generated from the following file:

- [veriloga.py](#)

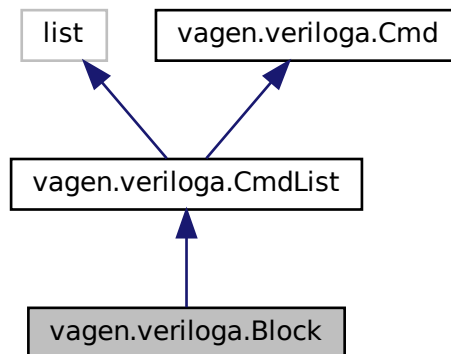
7.2 vagen.veriloga.Block Class Reference

Command [Block](#) Class.

Inheritance diagram for vagen.veriloga.Block:



Collaboration diagram for vagen.veriloga.Block:



Public Member Functions

- def `__init__` (self, header, *cmds)
Constructor.
- def `getHeader` (self)
Return the header of a block command.
- def `getVA` (self, padding)
Return the VA verilog command.

Public Attributes

- `header`

7.2.1 Detailed Description

Command [Block](#) Class.

7.2.2 Constructor & Destructor Documentation

7.2.2.1 `__init__()`

```
def vagen.veriloga.Block.__init__ (
    self,
    header,
    * cmds )
```

Constructor.

Parameters

<i>self</i>	object pointer
<i>header</i>	header of the command block
<i>*cmds</i>	variable number of Cmd or CmdList to be added

Reimplemented in [vagen.veriloga.RepeatLoop](#), [vagen.veriloga.WaitAnalogEvent](#), and [vagen.veriloga.WhileLoop](#).

7.2.3 Member Function Documentation

7.2.3.1 getHeader()

```
def vagen.veriloga.Block.getHeader (
    self )
```

Return the header of a block command.

Parameters

<i>self</i>	object pointer
-------------	----------------

Returns

header o the block

7.2.3.2 getVA()

```
def vagen.veriloga.Block.getVA (
    self,
    padding )
```

Return the VA verilog command.

Parameters

<i>self</i>	object pointer
<i>padding</i>	number of tabs by which the text will be right shifted

Returns

verilog command

Reimplemented from [vagen.veriloga.CmdList](#).

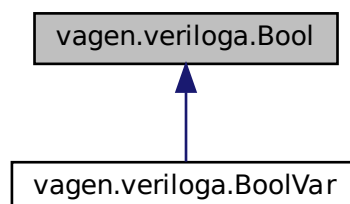
The documentation for this class was generated from the following file:

- [veriloga.py](#)

7.3 vagen.veriloga.Bool Class Reference

Class of [Bool](#) operators.

Inheritance diagram for vagen.veriloga.Bool:



Public Member Functions

- def `__init__` (self, value)
Constructor.
- def `getValue` (self)
Return the operator value.
- def `__and__` (self, other)
And logic override.
- def `__rand__` (self, other)
Reverse and logic override.
- def `__or__` (self, other)
Or logic override.
- def `__ror__` (self, other)
Reverse or logic override.
- def `__xor__` (self, other)
Xor logic override.
- def `__rxor__` (self, other)
Reverse xor logic override.
- def `__invert__` (self)
Inversion override.
- def `__str__` (self)
str override
- def `__eq__` (self, other)
Equal override.
- def `__ne__` (self, other)
Not equal override.

Public Attributes

- `value`

7.3.1 Detailed Description

Class of [Bool](#) operators.

7.3.2 Constructor & Destructor Documentation

7.3.2.1 `__init__()`

```
def vagen.veriloga.Bool.__init__ (
    self,
    value )
```

Constructor.

Parameters

<i>Self</i>	The object pointer.
<i>Value</i>	String representing a Real expression, an Integer , a Bool , or a value that can be converted to Bool .

Reimplemented in [vagen.veriloga.BoolVar](#).

7.3.3 Member Function Documentation

7.3.3.1 `__and__()`

```
def vagen.veriloga.Bool.__and__ (
    self,
    other )
```

And logic override.

Parameters

<i>Self</i>	First operand object pointer.
<i>Other</i>	Second operand.

Returns

Result of the and operation.

7.3.3.2 `__eq__()`

```
def vagen.veriloga.Bool.__eq__ (
    self,
    other )
```

Equal override.

Parameters

<i>self</i>	Left operand object pointer.
<i>other</i>	Right operand.

Returns

expression representing the comparison.

7.3.3.3 `__invert__()`

```
def vagen.veriloga.Bool.__invert__ (
    self )
```

Inversion override.

Parameters

<i>Self</i>	Object pointer.
-------------	-----------------

Returns

Expression representing inversion.

7.3.3.4 `__ne__()`

```
def vagen.veriloga.Bool.__ne__ (
    self,
    other )
```


Not equal override.

Parameters

<i>self</i>	Left operand object pointer.
<i>other</i>	Right operand.

Returns

expression representing the comparison.

7.3.3.5 __or__()

```
def vagen.veriloga.Bool.__or__ (
    self,
    other )
```

Or logic override.

Parameters

<i>Self</i>	First operand object pointer.
<i>Other</i>	Second operand.

Returns

Result of the or operation.

7.3.3.6 __rand__()

```
def vagen.veriloga.Bool.__rand__ (
    self,
    other )
```

Reverse and logic override.

Parameters

<i>Self</i>	First operand object pointer.
<i>Other</i>	Second operand.

Returns

Result of the and operation.

7.3.3.7 __ror__()

```
def vagen.veriloga.Bool.__ror__ (
    self,
    other )
```

Reverse or logic override.

Parameters

<i>Self</i>	First operand object pointer.
<i>Other</i>	Second operand.

Returns

Result of the or operation.

7.3.3.8 __rxor__()

```
def vagen.veriloga.Bool.__rxor__ (
    self,
    other )
```

Reverse xor logic override.

Parameters

<i>Self</i>	First operand object pointer.
<i>Other</i>	Second operand.

Returns

Result of the exclusive or operation.

7.3.3.9 __str__()

```
def vagen.veriloga.Bool.__str__ (
    self )
```

str override

Parameters

<i>Self</i>	Object pointer.
-------------	-----------------

Returns

String representing the expression

7.3.3.10 __xor__()

```
def vagen.veriloga.Bool.__xor__ (
    self,
    other )
```

Xor logic override.

Parameters

<i>Self</i>	First operand object pointer.
<i>Other</i>	Second operand.

Returns

Result of the exclusive or operation.

7.3.3.11 getValue()

```
def vagen.veriloga.Bool.getValue (
    self )
```

Return the operator value.

Parameters

<i>Self</i>	The object pointer.
-------------	---------------------

Returns

String representing the [Bool](#) expression.

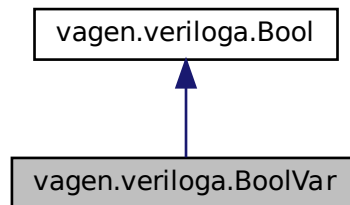
The documentation for this class was generated from the following file:

- [veriloga.py](#)

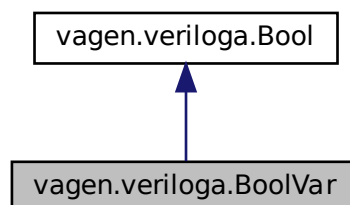
7.4 vagen.veriloga.BoolVar Class Reference

Boolean variable class.

Inheritance diagram for vagen.veriloga.BoolVar:



Collaboration diagram for vagen.veriloga.BoolVar:



Public Member Functions

- `def __init__(self, value)`
Constructor.
- `def toggle(self)`
Toogle.
- `def eq(self, value)`
Atribution.

Additional Inherited Members

7.4.1 Detailed Description

Boolean variable class.

7.4.2 Constructor & Destructor Documentation

7.4.2.1 `__init__()`

```
def vagen.veriloga.BoolVar.__init__ (
    self,
    value )
```

Constructor.

Parameters

<i>self</i>	object pointer
<i>value</i>	string representing the value

Reimplemented from [vagen.veriloga.Bool](#).

7.4.3 Member Function Documentation

7.4.3.1 `eq()`

```
def vagen.veriloga.BoolVar.eq (
    self,
    value )
```

Attribution.

Parameters

<i>self</i>	object pointer
<i>value</i>	A number representing the value

Returns

Return a command representing the attribution to a variable

7.4.3.2 `toggle()`

```
def vagen.veriloga.BoolVar.toggle (
    self )
```

Toogle.

Parameters

<i>self</i>	object pointer
-------------	----------------

Returns

Return a command representing the state toggle

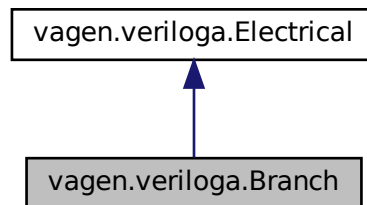
The documentation for this class was generated from the following file:

- [veriloga.py](#)

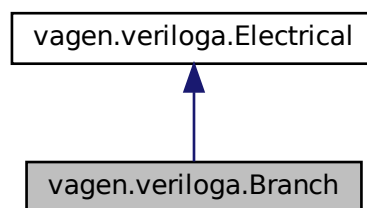
7.5 vagen.veriloga.Branch Class Reference

[Branch](#) class.

Inheritance diagram for vagen.veriloga.Branch:



Collaboration diagram for vagen.veriloga.Branch:



Public Member Functions

- `def __init__(self, node1, node2)`
constructor

Additional Inherited Members

7.5.1 Detailed Description

[Branch](#) class.

7.5.2 Constructor & Destructor Documentation

7.5.2.1 __init__()

```
def vagen.veriloga.Branch.__init__ (
    self,
```

```

    node1,
    node2 )

```

constructor

Parameters

<i>self</i>	The object pointer.
<i>node1</i>	Electrical signal representing the first node
<i>node2</i>	Electrical signal representing the second node

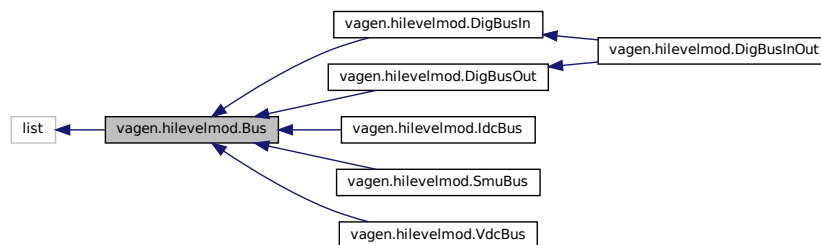
The documentation for this class was generated from the following file:

- [veriloga.py](#)

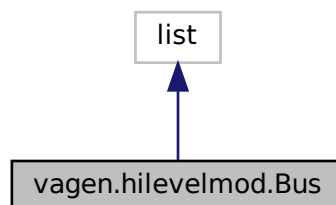
7.6 vagen.hilevelmod.Bus Class Reference

[Bus](#) class.

Inheritance diagram for vagen.hilevelmod.Bus:



Collaboration diagram for vagen.hilevelmod.Bus:



Public Member Functions

- def [__init__](#) (self, Type, busType)
Constructor.
- def [__getitem__](#) (self, key)
Slice override.
- def [append](#) (self, item)
Append override.

Public Attributes

- **Type**
- **busType**

7.6.1 Detailed Description

Bus class.

Child of a list. It implements additional methods to deal with read and write operations to a bus. It also overrides the slice method, so it works similar to a slice of a bus in verilog.

7.6.2 Constructor & Destructor Documentation

7.6.2.1 `__init__()`

```
def vagen.hilevelmod.Bus.__init__ (
    self,
    Type,
    busType )
```

Constructor.

Parameters

<i>self</i>	The object pointer.
<i>Type</i>	Type of the bus elements
<i>busType</i>	Type of the bus

7.6.3 Member Function Documentation

7.6.3.1 `__getitem__()`

```
def vagen.hilevelmod.Bus.__getitem__ (
    self,
    key )
```

Slice override.

Override the slice operator, so it will be in the format [msb:lsb:step]

Parameters

<i>self</i>	The object pointer.
<i>key</i>	Key can be an slice or index

Returns

Another bus or an element.

7.6.3.2 `append()`

```
def vagen.hilevelmod.Bus.append (
    self,
    item )
```

Append override.

Parameters

<i>self</i>	The object pointer.
<i>item</i>	Item to be appended to the bus

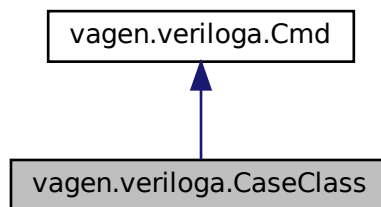
The documentation for this class was generated from the following file:

- [hilevelmod.py](#)

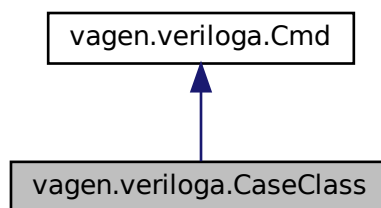
7.7 vagen.veriloga.CaseClass Class Reference

Condition Class.

Inheritance diagram for vagen.veriloga.CaseClass:



Collaboration diagram for vagen.veriloga.CaseClass:



Public Member Functions

- `def __init__ (self, test, *cmds)`
Constructor.
- `def getBlockList (self)`
Return the list of block of commands.
- `def append (self, *cmds)`
Add command.
- `def getVA (self, padding)`
Return the VA verilog command.

Public Attributes

- `test`
- `cmds`

7.7.1 Detailed Description

Condition Class.

It is used by the function `Case` in order to provide the case structure

7.7.2 Constructor & Destructor Documentation

7.7.2.1 `__init__()`

```
def vagen.veriloga.CaseClass.__init__ (
    self,
    test,
    * cmds )
```

Constructor.

Parameters

<i>self</i>	object pointer
<i>test</i>	Must be Integer , Bool , or Real
<i>*cmds</i>	variable number of tuples containing a condition and a command

7.7.3 Member Function Documentation

7.7.3.1 `append()`

```
def vagen.veriloga.CaseClass.append (
    self,
    * cmds )
```

Add command.

Parameters

<i>self</i>	object pointer
<i>*cmds</i>	variable number of tuples containing a condition and a command

7.7.3.2 `getBlockList()`

```
def vagen.veriloga.CaseClass.getBlockList (
    self )
```

Return the list of block of commands.

Parameters

<i>self</i>	object pointer
-------------	----------------

Returns

a list of block of commands

7.7.3.3 getVA()

```
def vagen.veriloga.CaseClass.getVA (
    self,
    padding )
```

Return the VA verilog command.

Parameters

<i>self</i>	object pointer
<i>padding</i>	number of tabs by which the text will be right shifted

Returns

verilog command

Reimplemented from [vagen.veriloga.Cmd](#).

The documentation for this class was generated from the following file:

- [veriloga.py](#)

7.8 vagen.hilevelmod.Clock Class Reference

[Clock](#) class.

Public Member Functions

- def [__init__](#) (self, hiLevelMod, pin)
Constructor.
- def [on](#) (self, frequency)
Turn the clock generator on.
- def [off](#) (self)
Turn the clock generator off.

Public Attributes

- **clockCount**
- **isOn**
- **halfPeriod**
- **time**
- **at**

Static Public Attributes

- int **clockCount** = 1

7.8.1 Detailed Description

[Clock](#) class.

7.8.2 Constructor & Destructor Documentation

7.8.2.1 `__init__()`

```
def vagen.hilevelmod.Clock.__init__ (
    self,
    hiLevelMod,
    pin )
```

Constructor.

Parameters

<i>self</i>	The object pointer.
<i>hiLevelMod</i>	Hi level model in which the analog command will be added.
<i>pin</i>	DigIn or DigInOut

7.8.3 Member Function Documentation

7.8.3.1 `off()`

```
def vagen.hilevelmod.Clock.off (
    self )
```

Turn the clock generator off.

Parameters

<i>self</i>	The object pointer.
-------------	---------------------

7.8.3.2 `on()`

```
def vagen.hilevelmod.Clock.on (
    self,
    frequency )
```

Turn the clock generator on.

Parameters

<i>self</i>	The object pointer.
<i>frequency</i>	frequency of the clock generator.

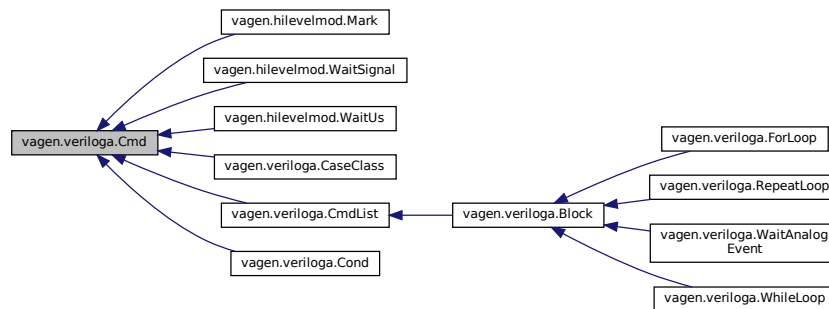
The documentation for this class was generated from the following file:

- [hilevelmod.py](#)

7.9 `vagen.veriloga.Cmd` Class Reference

Command class.

Inheritance diagram for vagen.veriloga.Cmd:



Public Member Functions

- `def __init__(self, cmd)`
Constructor.
- `def __str__(self)`
Return string representation.
- `def getVA(self, padding)`
Return the VA verilog command.

Public Attributes

- `cmd`

7.9.1 Detailed Description

Command class.

7.9.2 Constructor & Destructor Documentation

7.9.2.1 __init__()

```
def vagen.veriloga.Cmd.__init__(
    self,
    cmd )
```

Constructor.

Parameters

<i>self</i>	object pointer
<i>cmd</i>	command to be added to the va

Reimplemented in [vagen.hilevelmod.WaitSignal](#), [vagen.hilevelmod.WaitUs](#), and [vagen.hilevelmod.Mark](#).

7.9.3 Member Function Documentation

7.9.3.1 `__str__()`

```
def vagen.veriloga.Cmd.__str__ (
    self )
```

Return string representation.

Parameters

<i>self</i>	object pointer
-------------	----------------

Returns

string representation

Reimplemented in [vagen.veriloga.CmdList](#), [vagen.hilevelmod.WaitUs](#), [vagen.hilevelmod.WaitSignal](#), and [vagen.hilevelmod.Mark](#).

7.9.3.2 `getVA()`

```
def vagen.veriloga.Cmd.getVA (
    self,
    padding )
```

Return the VA verilog command.

Parameters

<i>self</i>	object pointer
<i>padding</i>	padding number of tabs by which the text will be right shifted

Returns

verilog command

Reimplemented in [vagen.veriloga.CaseClass](#), [vagen.veriloga.Cond](#), [vagen.veriloga.Block](#), [vagen.veriloga.CmdList](#), [vagen.hilevelmod.WaitUs](#), [vagen.hilevelmod.WaitSignal](#), and [vagen.hilevelmod.Mark](#).

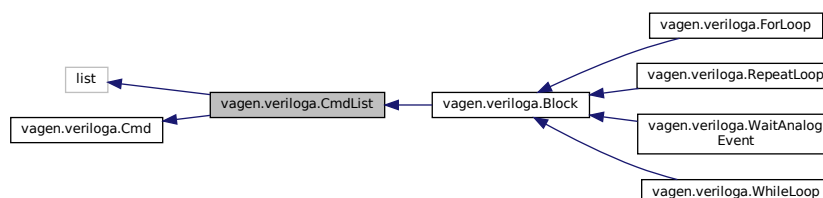
The documentation for this class was generated from the following file:

- [veriloga.py](#)

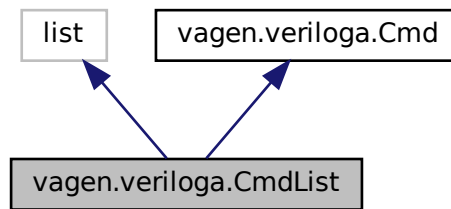
7.10 `vagen.veriloga.CmdList` Class Reference

Command List class.

Inheritance diagram for `vagen.veriloga.CmdList`:



Collaboration diagram for vagen.veriloga.CmdList:



Public Member Functions

- def `__init__` (self, *cmds)
Constructor.
- def `__str__` (self)
Return string representation.
- def `flat` (self)
Return a flat command list Fatten.
- def `append` (self, *cmds)
append override
- def `getVA` (self, padding)
Return the VA verilog command.

Additional Inherited Members

7.10.1 Detailed Description

Command List class.

7.10.2 Constructor & Destructor Documentation

7.10.2.1 `__init__`()

```
def vagen.veriloga.CmdList.__init__ (
    self,
    * cmds )
```

Constructor.

Parameters

<i>self</i>	object pointer
<i>cmds</i>	commands to be added to the va

7.10.3 Member Function Documentation

7.10.3.1 __str__()

```
def vagen.veriloga.CmdList.__str__ (
    self )
```

Return string representation.

Parameters

<i>self</i>	object pointer
-------------	----------------

Returns

string representation

Reimplemented from [vagen.veriloga.Cmd](#).

7.10.3.2 append()

```
def vagen.veriloga.CmdList.append (
    self,
    * cmds )
```

append override

Parameters

<i>self</i>	object pointer
-------------	----------------

7.10.3.3 flat()

```
def vagen.veriloga.CmdList.flat (
    self )
```

Return a flat command list Fatten.

Parameters

<i>self</i>	object pointer
-------------	----------------

Returns

flat command list. Only imediate CmdLists will be open.

7.10.3.4 getVA()

```
def vagen.veriloga.CmdList.getVA (
    self,
    padding )
```

Return the VA verilog command.

Parameters

<i>self</i>	object pointer
<i>padding</i>	number of tabs by which the text will be right shifted

Returns

verilog command

Reimplemented from [vagen.veriloga.Cmd](#).

Reimplemented in [vagen.veriloga.Block](#).

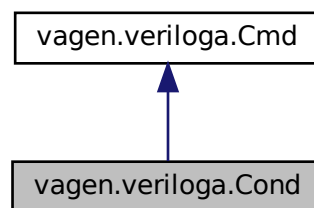
The documentation for this class was generated from the following file:

- [veriloga.py](#)

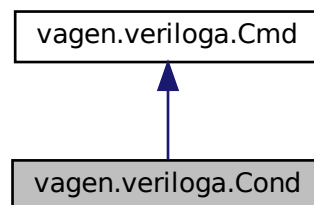
7.11 vagen.veriloga.Cond Class Reference

Condition Class.

Inheritance diagram for vagen.veriloga.Cond:



Collaboration diagram for vagen.veriloga.Cond:



Public Member Functions

- `def __init__(self, cond, *cmds)`
Constructor.
- `def getCond(self)`
Return the [Cond](#) condition.
- `def getBlock(self, state=True)`
Return the block of commands for a given state.
- `def append(self, state, *cmds)`
Add command.
- `def Else(self, *cmds)`

List of commands to be run when condition is false.

- def `getVA` (self, padding)

Return the VA verilog command.

Public Attributes

- `cond`
- `cmdDict`

7.11.1 Detailed Description

Condition Class.

It is used inside the function `If` in order to provide an `If` and `else` structure

7.11.2 Constructor & Destructor Documentation

7.11.2.1 `__init__()`

```
def vagen.veriloga.Cond.__init__ (
    self,
    cond,
    * cmds )
```

Constructor.

Parameters

<i>self</i>	object pointer
<i>cond</i>	condition that must be satisfied in order to run the sequence of commands in the block
<i>*cmds</i>	variable number of Cmd or CmdList to be added

7.11.3 Member Function Documentation

7.11.3.1 `append()`

```
def vagen.veriloga.Cond.append (
    self,
    state,
    * cmds )
```

Add command.

Parameters

<i>self</i>	object pointer
<i>state</i>	true or false
<i>*cmds</i>	variable number of Cmd or CmdList to be added

7.11.3.2 `Else()`

```
def vagen.veriloga.Cond.Else (
    self,
    * cmds )
```

List of commands to be run when condition is false.

Parameters

<i>self</i>	object pointer
<i>*cmds</i>	variable number of Cmd or CmdList to be added

Returns

pointer to self

7.11.3.3 **getBlock()**

```
def vagen.veriloga.Cond.getBlock (
    self,
    state = True )
```

Return the block of commands for a given state.

Parameters

<i>self</i>	object pointer
<i>state</i>	true or false

Returns

block of commands for True and False conditions

7.11.3.4 **getCond()**

```
def vagen.veriloga.Cond.getCond (
    self )
```

Return the [Cond](#) condition.

Parameters

<i>self</i>	object pointer
-------------	----------------

Returns

[Bool](#) class representing the condition that must be satisfied in order run the sequence of commands in the block

7.11.3.5 **getVA()**

```
def vagen.veriloga.Cond.getVA (
    self,
    padding )
```

Return the VA verilog command.

Parameters

<i>self</i>	object pointer
<i>padding</i>	number of tabs by which the text will be right shifted

Returns

verilog command

Reimplemented from [vagen.veriloga.Cmd](#).

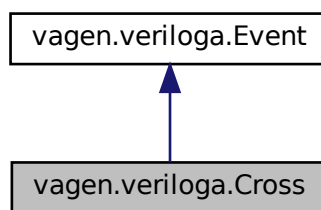
The documentation for this class was generated from the following file:

- [veriloga.py](#)

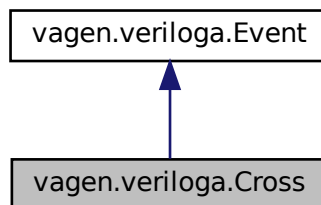
7.12 vagen.veriloga.Cross Class Reference

[Cross](#) Class.

Inheritance diagram for vagen.veriloga.Cross:



Collaboration diagram for vagen.veriloga.Cross:



Public Member Functions

- `def __init__(self, signal, threshold, edge, *pars)`
Constructor.

Additional Inherited Members

7.12.1 Detailed Description

[Cross](#) Class.

7.12.2 Constructor & Destructor Documentation

7.12.2.1 `__init__()`

```
def vagen.veriloga.Cross.__init__ (
    self,
    signal,
    threshold,
    edge,
    * pars )
```

Constructor.

Parameters

<i>self</i>	object pointer
<i>signal</i>	Real class or build-in real representing the signal
<i>threshold</i>	Real class or build-in real representing the threshold that must be crossed
<i>edge</i>	It can be rising, falling or both
<i>*pars</i>	optional Real or build-in real parameters timeTol and expTol in this order

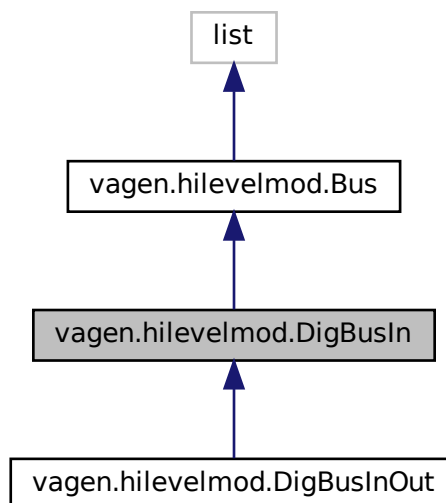
The documentation for this class was generated from the following file:

- [veriloga.py](#)

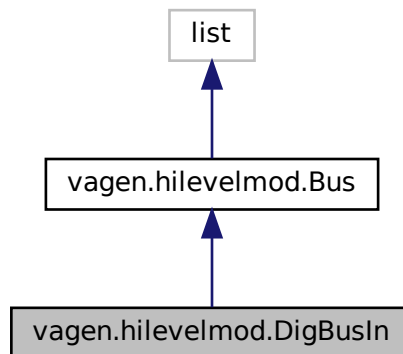
7.13 vagen.hilevelmod.DigBusIn Class Reference

[DigBusIn](#) class.

Inheritance diagram for vagen.hilevelmod.DigBusIn:



Collaboration diagram for `vagen.hilevelmod.DigBusIn`:



Public Member Functions

- `def __init__ (self)`
Constructor.
- `def read (self, signed=False)`
Read a binary from the digital input bus.

Additional Inherited Members

7.13.1 Detailed Description

[DigBusIn](#) class.

Child of a list. It implements additional methods to deal with read and write operations to a bus. It also overrides the `slice` method, so it works similar to a slice of a bus in verilog

7.13.2 Constructor & Destructor Documentation

7.13.2.1 `__init__()`

```
def vagen.hilevelmod.DigBusIn.__init__ (
    self )
```

Constructor.

Parameters

<code>self</code>	The object pointer.
-------------------	---------------------

Reimplemented in [vagen.hilevelmod.DigBusInOut](#).

7.13.3 Member Function Documentation

7.13.3.1 read()

```
def vagen.hilevelmod.DigBusIn.read (
    self,
    signed = False )
```

Read a binary from the digital input bus.

Parameters

<i>self</i>	The object pointer.
<i>signed</i>	Read as signed if True and unsigned otherwise.

Returns

The commands to read a digital bus as binary.

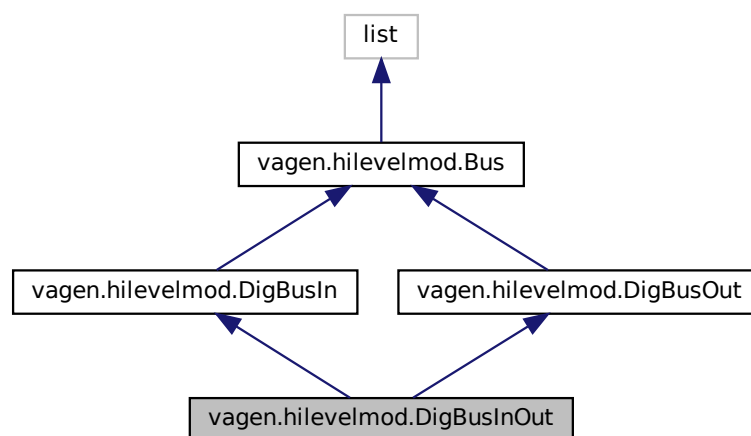
The documentation for this class was generated from the following file:

- [hilevelmod.py](#)

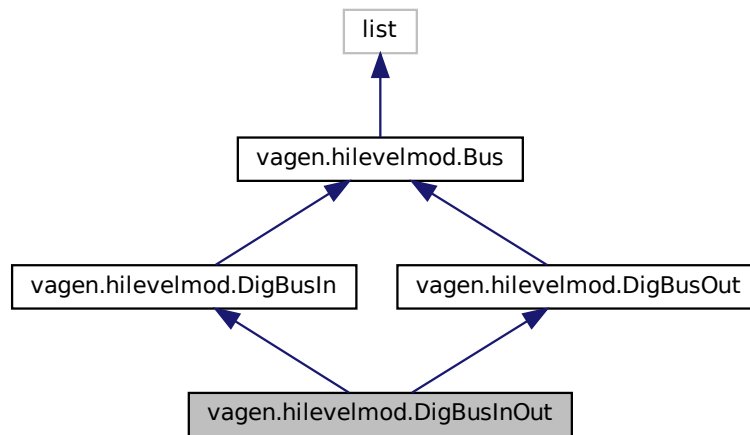
7.14 vagen.hilevelmod.DigBusInOut Class Reference

[DigBusInOut](#) class.

Inheritance diagram for vagen.hilevelmod.DigBusInOut:



Collaboration diagram for `vagen.hilevelmod.DigBusInOut`:



Public Member Functions

- `def __init__ (self)`
Constructor.
- `def hiZ (self)`
Set the pins at hiz in order to use the read function.
- `def lowZ (self)`
Set the pins to low impedance in order to use the write function.

Additional Inherited Members

7.14.1 Detailed Description

[DigBusInOut](#) class.

Child of a list. It implements additional methods to deal with read and write operations to a bus. It also overrides the slice method, so it works similar to a slice of a bus in verilog

7.14.2 Constructor & Destructor Documentation

7.14.2.1 `__init__()`

```
def vagen.hilevelmod.DigBusInOut.__init__ (
    self )
```

Constructor.

Parameters

<code>self</code>	The object pointer.
-------------------	---------------------

Reimplemented from [vagen.hilevelmod.DigBusOut](#).

7.14.3 Member Function Documentation

7.14.3.1 hiZ()

```
def vagen.hilevelmod.DigBusInOut.hiZ (  
    self )
```

Set the pins at hiz in order to use the read function.

Parameters

<i>self</i>	The object pointer.
-------------	---------------------

Returns

The commands to change the inOut pin to hiZ (input).

7.14.3.2 lowZ()

```
def vagen.hilevelmod.DigBusInOut.lowZ (  
    self )
```

Set the pins to low impedance in order to use the write function.

Parameters

<i>self</i>	The object pointer.
-------------	---------------------

Returns

The commands to change the inOut pin to lowZ (output).

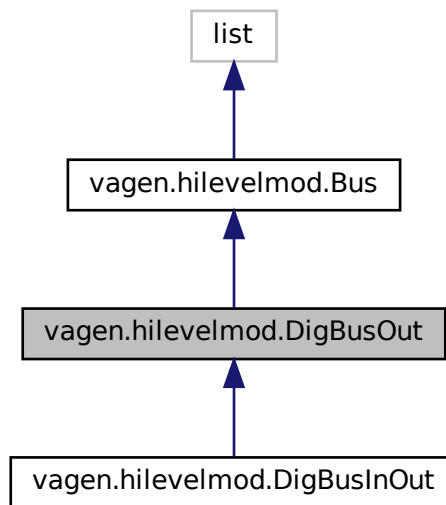
The documentation for this class was generated from the following file:

- [hilevelmod.py](#)

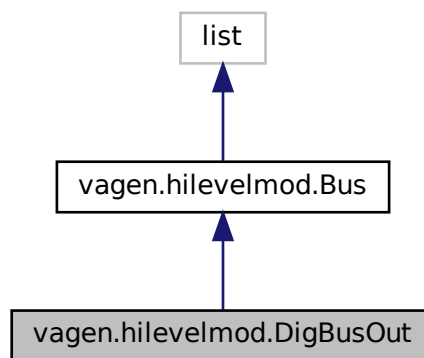
7.15 vagen.hilevelmod.DigBusOut Class Reference

[DigBusOut](#) class.

Inheritance diagram for vagen.hilevelmod.DigBusOut:



Collaboration diagram for vagen.hilevelmod.DigBusOut:



Public Member Functions

- def `__init__` (self)
Constructor.
- def `setDelay` (self, delay)
Set the delay times for all digital output pin.
- def `setRiseFall` (self, rise, fall)
Set the rise and the fall times of all digital output pin.
- def `write` (self, value)
Write a binary to the digital output bus.

Additional Inherited Members

7.15.1 Detailed Description

[DigBusOut](#) class.

Child of a list. It implements additional methods to deal with read and write operations to a bus. It also overrides the slice method, so it works similar to a slice of a bus in verilog

7.15.2 Constructor & Destructor Documentation

7.15.2.1 __init__()

```
def vagen.hilevelmod.DigBusOut.__init__ (
    self )
```

Constructor.

Parameters

<i>self</i>	The object pointer.
-------------	---------------------

Reimplemented in [vagen.hilevelmod.DigBusInOut](#).

7.15.3 Member Function Documentation

7.15.3.1 setDelay()

```
def vagen.hilevelmod.DigBusOut.setDelay (
    self,
    delay )
```

Set the delay times for all digital output pin.

Parameters

<i>self</i>	The object pointer.
<i>delay</i>	Real expression holding the delay time.

Returns

The commands to change the delay times.

7.15.3.2 setRiseFall()

```
def vagen.hilevelmod.DigBusOut.setRiseFall (
    self,
    rise,
    fall )
```

Set the rise and the fall times of all digital output pin.

Parameters

<i>self</i>	The object pointer.
<i>Rise</i>	Real expression holding the rise time.
<i>Fall</i>	Real expression holding the fall time.

Returns

The commands to change the rise and fall times.

7.15.3.3 write()

```
def vagen.hilevelmod.DigBusOut.write (
    self,
    value )
```

Write a binary to the digital output bus.

Parameters

<i>self</i>	The object pointer.
<i>value</i>	Integer expression representing the value to be written.

Returns

The commands to write to a digital bus.

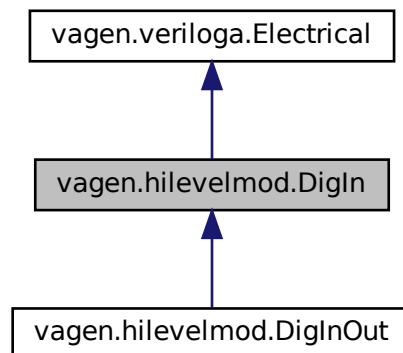
The documentation for this class was generated from the following file:

- [hilevelmod.py](#)

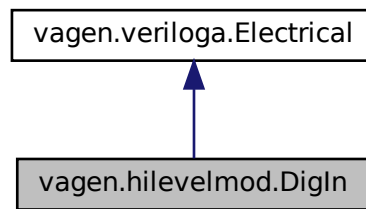
7.16 vagen.hilevelmod.DigIn Class Reference

[DigIn](#) class.

Inheritance diagram for vagen.hilevelmod.DigIn:



Collaboration diagram for vagen.hilevelmod.DigIn:



Public Member Functions

- def `__init__` (self, hiLevelMod, name, state, domain, inCap, serRes, gnd, delay, rise, fall)
Constructor.
- def `read` (self)
Read a state from the digital input.

Public Attributes

- `domain`
- `inCap`

7.16.1 Detailed Description

`DigIn` class.

Child of `Electrical` implementing additional features in order to work as a digital input pin

7.16.2 Constructor & Destructor Documentation

7.16.2.1 `__init__()`

```

def vagen.hilevelmod.DigIn.__init__ (
    self,
    hiLevelMod,
    name,
    state,
    domain,
    inCap,
    serRes,
    gnd,
    delay,
    rise,
    fall )
  
```

Constructor.

Parameters

<code>self</code>	The object pointer.
<code>hiLeveMod</code>	Hi level model in which the analog command will be added.

Parameters

<i>name</i>	Name of the electrical pin.
<i>state</i>	Dummy parameter for consistency.
<i>domain</i>	Electrical pin. The voltage across the domain will be equal the voltage in the digital pins when the logical state is 1.
<i>inCap</i>	Real expression holding the value of the input capacitance. This value will be set at the beginning of the simulation and can't be changed afterwards.
<i>serRes</i>	Dummy parameter for consistency.
<i>gnd</i>	Electrical representing the ground reference.
<i>delay</i>	Dummy parameter for consistency.
<i>rise</i>	Dummy parameter for consistency.
<i>fall</i>	Dummy parameter for consistency.

Reimplemented in [vagen.hilevelmod.DigInOut](#).

7.16.3 Member Function Documentation

7.16.3.1 read()

```
def vagen.hilevelmod.DigIn.read (
    self )
```

Read a state from the digital input.

Parameters

<i>self</i>	The object pointer.
-------------	---------------------

Returns

The commands to read the state of a digital pin.

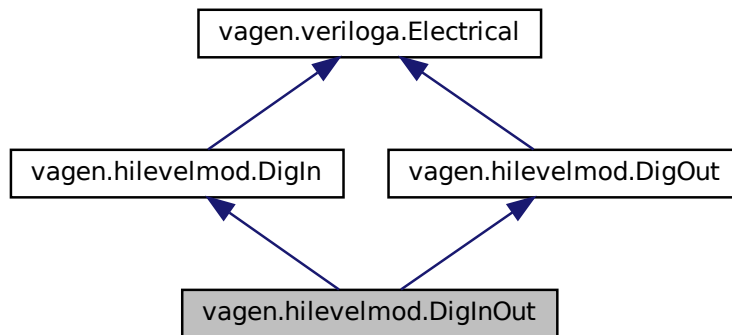
The documentation for this class was generated from the following file:

- [hilevelmod.py](#)

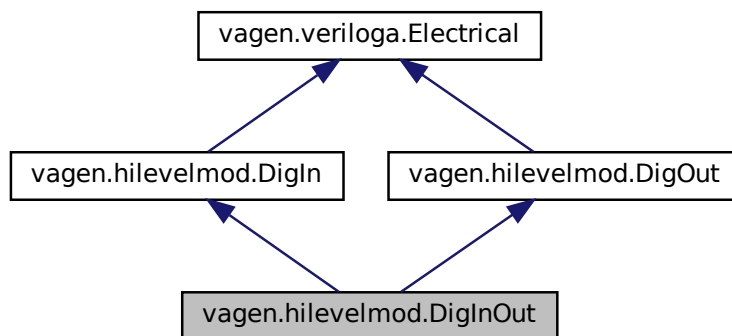
7.17 vagen.hilevelmod.DigInOut Class Reference

[DigInOut](#) class.

Inheritance diagram for vagen.hilevelmod.DigInOut:



Collaboration diagram for vagen.hilevelmod.DigInOut:



Public Member Functions

- `def __init__ (self, hiLevelMod, name, state, domain, inCap, serRes, gnd, delay, rise, fall)`
Construtor.
- `def hiZ (self)`
Set the pin at hiz in order to use the read function.
- `def lowZ (self)`
Set the pin to low impedance in order to use the write function.

Public Attributes

- `st`
- `serRes`
- `inCap`
- `res`
- `delay`

- **rise**
- **fall**
- **domain**

7.17.1 Detailed Description

[DigInOut](#) class.

Child of Electrical implementing additional features in order to work as a digital input/output pin

7.17.2 Constructor & Destructor Documentation

7.17.2.1 `__init__()`

```
def vagen.hilevelmod.DigInOut.__init__ (
    self,
    hiLevelMod,
    name,
    state,
    domain,
    inCap,
    serRes,
    gnd,
    delay,
    rise,
    fall )
```

Constructor.

Parameters

<i>self</i>	The object pointer.
<i>hiLeveMod</i>	Hi level model in which the analog command will be added.
<i>name</i>	Name of the electrical pin.
<i>state</i>	Boolean expression holding the intial state of the digital pin.
<i>domain</i>	electrical pin. The voltage across the digial pins will be equal to the domain when the logical state is 1.
<i>inCap</i>	Real expression holding the value of the input capacitance. This value will be set at the beggining of the simulation and can't be changed afterwards.
<i>serRes</i>	Real expression holding the value of the series resistance. This value will be set at the beggining of the simulation and can't be changed afterwards.
<i>gnd</i>	Electrical representing the ground reference.
<i>delay</i>	Real expression holding the initial delay time.
<i>rise</i>	Real expression holding the initial rise time.
<i>fall</i>	Real expression holding the initial fall time.

Reimplemented from [vagen.hilevelmod.DigOut](#).

7.17.3 Member Function Documentation

7.17.3.1 `hiZ()`

```
def vagen.hilevelmod.DigInOut.hiZ (
    self )
```

Set the pin at hiz in order to use the read function.

Parameters

<i>self</i>	The object pointer.
-------------	---------------------

Returns

The commands to change the inOut pin to hiZ (input).

7.17.3.2 lowZ()

```
def vagen.hilevelmod.DigInOut.lowZ (
    self )
```

Set the pin to low impedance in order to use the write function.

Parameters

<i>self</i>	The object pointer.
-------------	---------------------

Returns

The commands to change the inOut pin to lowZ (output).

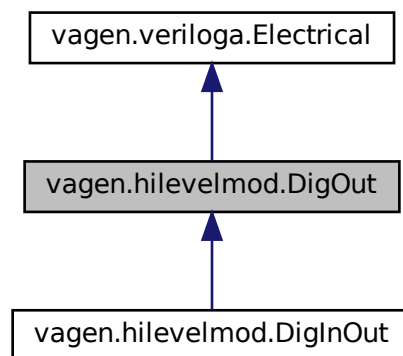
The documentation for this class was generated from the following file:

- [hilevelmod.py](#)

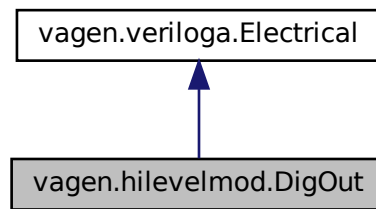
7.18 vagen.hilevelmod.DigOut Class Reference

[DigOut](#) class.

Inheritance diagram for vagen.hilevelmod.DigOut:



Collaboration diagram for vagen.hilevelmod.DigOut:



Public Member Functions

- `def __init__ (self, hiLevelMod, name, state, domain, inCap, serRes, gnd, delay, rise, fall)`
Construtor.
- `def setDelay (self, delay)`
Set the delay times of the digital output pin.
- `def setRiseFall (self, rise, fall)`
Set the rise and the fall times of the digital output pin.
- `def write (self, value)`
Write a state to the digital output.

Public Attributes

- `st`
- `serRes`
- `delay`
- `rise`
- `fall`

7.18.1 Detailed Description

`DigOut` class.

Child of `Electrical` implementing additional features in order to work as a digital output pin.

7.18.2 Constructor & Destructor Documentation

7.18.2.1 __init__()

```

def vagen.hilevelmod.DigOut.__init__ (
    self,
    hiLevelMod,
    name,
    state,
    domain,
    inCap,
    serRes,
    gnd,
    delay,

```

```

        rise,
        fall )

```

Construtor.

Parameters

<i>self</i>	The object pointer
<i>hiLeveMod</i>	Hi level model in which the analog command will be added.
<i>name</i>	Name of the electrical pin.
<i>state</i>	Boolean expression holding the intial state of the digital pin.
<i>domain</i>	electrical pin. The voltage across the digial pins will be equal to the domain when the logical state is 1.
<i>inCap</i>	Dummy parameter for consistency.
<i>serRes</i>	Real expression holding the value of the series resistance. This value will be set at the beggining of the simulation and can't be changed afterwards.
<i>gnd</i>	Electrical representing the ground reference.
<i>delay</i>	Real expression holding the initial delay time.
<i>rise</i>	Real expression holding the initial rise time.
<i>fall</i>	Real expression holding the initial fall time.

Reimplemented in [vagen.hilevelmod.DigInOut](#).

7.18.3 Member Function Documentation

7.18.3.1 setDelay()

```

def vagen.hilevelmod.DigOut.setDelay (
    self,
    delay )

```

Set the delay times of the digital output pin.

Parameters

<i>self</i>	The object pointer.
<i>delay</i>	Real expression holding the delay time.

Returns

The commands to change the delay.

7.18.3.2 setRiseFall()

```

def vagen.hilevelmod.DigOut.setRiseFall (
    self,
    rise,
    fall )

```

Set the rise and the fall times of the digital output pin.

Parameters

<i>self</i>	The object pointer.
<i>Rise</i>	Real expression holding the rise time.
<i>Fall</i>	Real expression holding the fall time.

Returns

The commands to change the rise and fall times.

7.18.3.3 write()

```
def vagen.hilevelmod.DigOut.write (
    self,
    value )
```

Write a state to the digital output.

Parameters

<i>self</i>	The object pointer.
<i>value</i>	Boolean expression representing the state to be written.

Returns

The commands to change the stare of a digital pin.

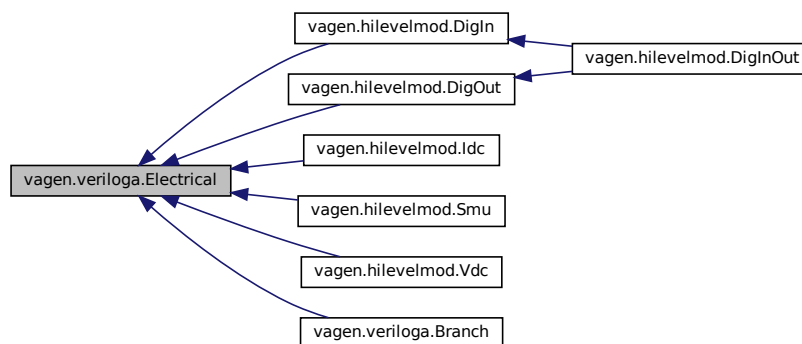
The documentation for this class was generated from the following file:

- [hilevelmod.py](#)

7.19 vagen.veriloga.Electrical Class Reference

Class of electrical signals.

Inheritance diagram for vagen.veriloga.Electrical:

**Public Member Functions**

- def `__init__` (self, name)
constructor
- def `getName` (self)
Return electrical name.
- def `vCont` (self, value)
Return a command representing voltage contribution.
- def `iCont` (self, value)
Return a command representing current contribution.
- def `vAttr` (self, value)

- Return a command representing voltage attribution.*
- def `iAttr` (self, value)
Return a command representing current attribution.
- def `vInd` (self, value)
Return a command representing voltage indirect assignment (Voltage that makes value true)
- def `iInd` (self, value)
Return a command representing current indirect assignment (Current that makes value true)

Public Attributes

- `name`
- `v`
- `i`

7.19.1 Detailed Description

Class of electrical signals.

7.19.2 Constructor & Destructor Documentation

7.19.2.1 `__init__()`

```
def vagen.veriloga.Electrical.__init__ (
    self,
    name )
```

constructor

Parameters

<i>self</i>	The object pointer.
<i>name</i>	string representing the name of the electrical signal

7.19.3 Member Function Documentation

7.19.3.1 `getName()`

```
def vagen.veriloga.Electrical.getName (
    self )
```

Return electrical name.

Parameters

<i>self</i>	The object pointer.
-------------	---------------------

Returns

string representing the name of the signal

7.19.3.2 `iAttr()`

```
def vagen.veriloga.Electrical.iAttr (
    self,
    value )
```

Return a command representing current attribution.

Parameters

<i>self</i>	The object pointer.
<i>value</i>	Real , float or int representig the value of the attribution

Returns

a [Cmd](#) representing the current attribution

7.19.3.3 iCont()

```
def vagen.veriloga.Electrical.iCont (
    self,
    value )
```

Return a command representing current contribution.

Parameters

<i>self</i>	The object pointer.
<i>value</i>	Real , float or int representig the value of the contribution

Returns

a [Cmd](#) representing the current contribution

7.19.3.4 iInd()

```
def vagen.veriloga.Electrical.iInd (
    self,
    value )
```

Return a command representing current indirect assignment (Current that makes value true)

Parameters

<i>self</i>	The object pointer.
<i>value</i>	Bool or bool condition

Returns

a [Cmd](#) representing the current indirect assigment

7.19.3.5 vAttr()

```
def vagen.veriloga.Electrical.vAttr (
    self,
    value )
```

Return a command representing voltage attribution.

Parameters

<i>self</i>	The object pointer.
<i>value</i>	Real , float or int representig the value of the attribution

Returns

a [Cmd](#) representing the voltage attribution

7.19.3.6 vCont()

```
def vagen.veriloga.Electrical.vCont (
    self,
    value )
```

Return a command representing voltage contribution.

Parameters

<i>self</i>	The object pointer.
<i>value</i>	Real , float or int representig the value of the contribution

Returns

a [Cmd](#) representing the voltage contribution

7.19.3.7 vInd()

```
def vagen.veriloga.Electrical.vInd (
    self,
    value )
```

Return a command representing voltage indirect assigment (Voltage that makes value true)

Parameters

<i>self</i>	The object pointer.
<i>value</i>	Bool or bool condition

Returns

a [Cmd](#) representing the voltage indirect assigment

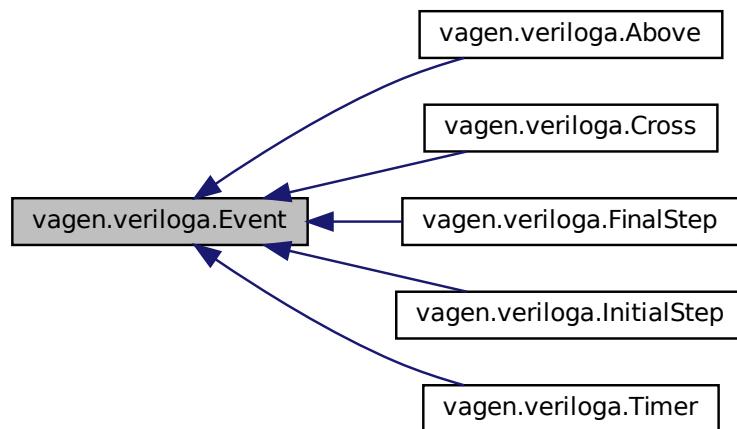
The documentation for this class was generated from the following file:

- [veriloga.py](#)

7.20 vagen.veriloga.Event Class Reference

Class of events.

Inheritance diagram for vagen.veriloga.Event:



Public Member Functions

- `def __init__(self, value)`
Constructor.
- `def __or__(self, other)`
or logic override
- `def __str__(self)`
string representation

Public Attributes

- `value`

7.20.1 Detailed Description

Class of events.

7.20.2 Constructor & Destructor Documentation

7.20.2.1 __init__()

```
def vagen.veriloga.Event.__init__(
    self,
    value )
```

Constructor.

Parameters

<i>self</i>	object pointer
<i>value</i>	string representing the event

7.20.3 Member Function Documentation

7.20.3.1 `__or__()`

```
def vagen.veriloga.Event.__or__ (
    self,
    other )
```

or logic override

Parameters

<i>self</i>	object pointer
<i>other</i>	pointer to another Event object

Returns

Return an [Event](#) representing the or logic between the two

7.20.3.2 `__str__()`

```
def vagen.veriloga.Event.__str__ (
    self )
```

string representation

Parameters

<i>self</i>	object pointer
<i>other</i>	pointer to another Event object

Returns

The string representation of the [Event](#)

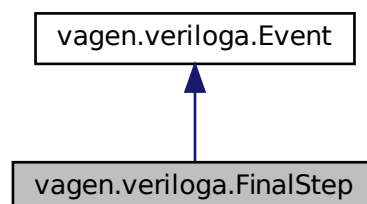
The documentation for this class was generated from the following file:

- [veriloga.py](#)

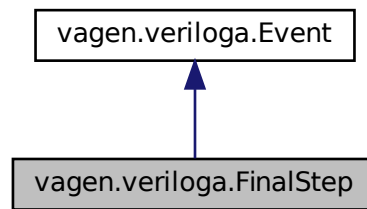
7.21 vagen.veriloga.FinalStep Class Reference

[FinalStep](#) class.

Inheritance diagram for vagen.veriloga.FinalStep:



Collaboration diagram for `vagen.veriloga.FinalStep`:



Public Member Functions

- `def __init__(self, *simTypes)`
Constructor.

Additional Inherited Members

7.21.1 Detailed Description

[FinalStep](#) class.

7.21.2 Constructor & Destructor Documentation

7.21.2.1 __init__()

```
def vagen.veriloga.FinalStep.__init__ (
    self,
    * simTypes )
```

Constructor.

Parameters

<i>self</i>	object pointer
<i>*simTypes</i>	optional parameters representing the simulation type

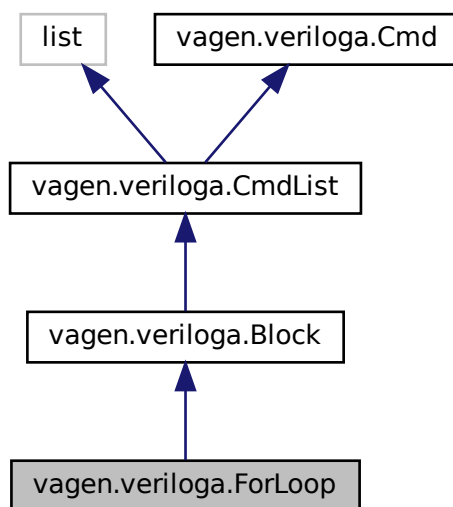
The documentation for this class was generated from the following file:

- [veriloga.py](#)

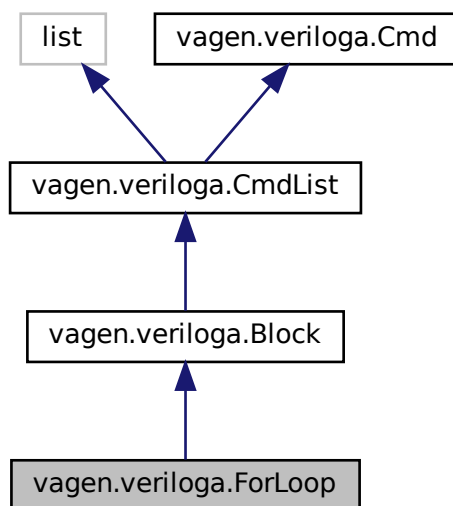
7.22 vagen.veriloga.ForLoop Class Reference

[ForLoop](#) class.

Inheritance diagram for vagen.veriloga.ForLoop:



Collaboration diagram for vagen.veriloga.ForLoop:



Public Member Functions

- def `__init__` (self, start, cond, inc, *cmds)
Constructor.
- def `getCond` (self)

- *Return the Forloop condition.*
- `def getStart (self)`
Return the Forloop start.
- `def getInc (self)`
Return the Forloop increment.

Public Attributes

- `cond`
- `start`
- `inc`

7.22.1 Detailed Description

[ForLoop](#) class.

7.22.2 Constructor & Destructor Documentation

7.22.2.1 `__init__()`

```
def vagen.veriloga.ForLoop.__init__ (
    self,
    start,
    cond,
    inc,
    * cmds )
```

Constructor.

Parameters

<i>self</i>	object pointer
<i>start</i>	command executed at the beggining
<i>cond</i>	condition that must be satisfied in order repeat the sequence of commands in the block
<i>inc</i>	command executed at the end of each step
<i>*cmds</i>	variable number of Cmd or CmdList to be added

7.22.3 Member Function Documentation

7.22.3.1 `getCond()`

```
def vagen.veriloga.ForLoop.getCond (
    self )
```

Return the Forloop condition.

Parameters

<i>self</i>	object pointer
-------------	----------------

Returns

[Bool](#) class representing the condition that must be satisfied in order repeat the sequence of commands in the block

7.22.3.2 `getInc()`

```
def vagen.veriloga.ForLoop.getInc (
    self )
```

Return the Forloop increment.

Parameters

<i>self</i>	object pointer
-------------	----------------

Returns

[Cmd](#) class representing the increment command run at each iteration

7.22.3.3 `getStart()`

```
def vagen.veriloga.ForLoop.getStart (
    self )
```

Return the Forloop start.

Parameters

<i>self</i>	object pointer
-------------	----------------

Returns

[Cmd](#) class representing the initial command run by the loop

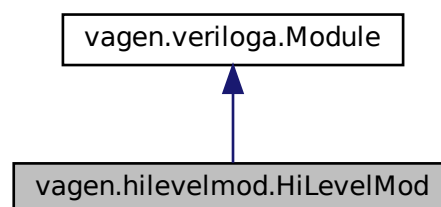
The documentation for this class was generated from the following file:

- [veriloga.py](#)

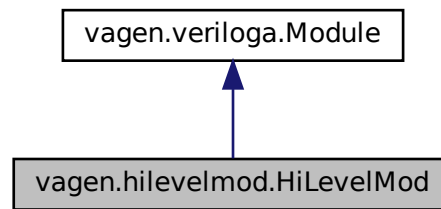
7.23 vagen.hilevelmod.HiLevelMod Class Reference

[HiLevelMod](#) class.

Inheritance diagram for vagen.hilevelmod.HiLevelMod:



Collaboration diagram for vagen.hilevelmod.HiLevelMod:



Public Member Functions

- def `__init__` (self, tbName, timeTol=None)
Constructor.
- def `var` (self, value=0, name="")
Add variable to the module.
- def `marker` (self, name, riseFall=100e-12)
Return a marker object.
- def `dig` (self, domain, name="", width=1, direction="internal", value=0, inCap=1e-14, serRes=100.0, gnd=None, delay=0, rise=1e-12, fall=1e-12)
Return a `DigIn`, `DigOut`, or `DigInOut` object.
- def `sw` (self, pin1, pin2, cond=0.0, rise=1e-6, fall=1e-6)
switch
- def `clock` (self, pin)
Build a clock model using a digital pin.
- def `smu` (self, name="", width=1, direction="internal", volt=0, minCur=0, maxCur=0, res=1e12, gnd=None)
Return a `Smu` object or a `SmuBus` object if width > 1.
- def `vdc` (self, name="", width=1, direction="internal", value=0, gnd=None, rise=0, fall=0)
Return a `Vdc` object or a `VdcBus` object if width > 1.
- def `ldc` (self, name="", width=1, direction="internal", value=0, gnd=None, rise=0, fall=0)
Return a `Ldc` object or a `LdcBus` object if width > 1.
- def `seqNested` (self, cmdsIn)
Sequence.
- def `seq` (self, cond)
Sequence.
- def `getEqs` (self)
Return the equations in a format that can be imported by the maestro view.
- def `getOcn` (self)
Return a ocean script that add equations to the opened session of adexl.

Public Attributes

- `dcCmdList`
- `time`
- `state`
- `runSt`
- `eventId`

- `evntList`
- `pEventList`
- `evntListG`
- `markers`
- `nSeq`
- `testSeqs`
- `timeArgs`
- `nState`
- `pCase`
- `cond`

7.23.1 Detailed Description

[HiLevelMod](#) class.

Child of the module class in the veriloA module. It provides additional methods for dealing with digital bus, current sources, voltage sources, clocks and switches

7.23.2 Constructor & Destructor Documentation

7.23.2.1 `__init__()`

```
def vagen.hilevelmod.HiLevelMod.__init__ (
    self,
    tbName,
    timeTol = None )
```

Constructor.

Parameters

<i>self</i>	The object pointer.
<i>tbName</i>	Name of the test bench.
<i>timeTol</i>	Time tolerances for the timer.

7.23.3 Member Function Documentation

7.23.3.1 `clock()`

```
def vagen.hilevelmod.HiLevelMod.clock (
    self,
    pin )
```

Build a clock model using a digital pin.

Parameters

<i>self</i>	The object pointer.
<i>pin</i>	DigIn or DigInOut .

Returns

a [Clock](#) class.

7.23.3.2 dig()

```
def vagen.hilevelmod.HiLevelMod.dig (
    self,
    domain,
    name = "",
    width = 1,
    direction = "internal",
    value = 0,
    inCap = 1e-14,
    serRes = 100.0,
    gnd = None,
    delay = 0,
    rise = 1e-12,
    fall = 1e-12 )
```

Return a [DigIn](#), [DigOut](#), or [DigInOut](#) object.

A [DigBusIn](#), [DigBusOut](#) or [DigBusInOut](#) will be returned if width > 0.

Parameters

<i>self</i>	The object pointer.
<i>domain</i>	electrical pin. The voltage across the digial pins will be equal to the domain when the logical state is 1.
<i>name</i>	Name of the electrical pin.
<i>value</i>	Integer expression holding the intial value of the digital pin.
<i>width</i>	If width is greather than 1, It returns a bus.
<i>direction</i>	It can be internal, input, output, or inout.
<i>inCap</i>	Real expression holding the value of the input capacitance. This value will be set at the beggining of the simulation and can't be changed afterwards.
<i>serRes</i>	Real expression holding the value of the series resistance. This value will be set at the beggining of the simulation and can't be changed afterwards.
<i>gnd</i>	Electrical representing the ground reference.
<i>delay</i>	Real expression holding the delay.
<i>rise</i>	Real expression holding the initial rise time.
<i>fall</i>	Real expression holding the initial fall time.

Returns

[DigIn](#), [DigOut](#), or [DigInOut](#) object. A [DigBusIn](#), [DigBusOut](#) or [DigBusInOut](#) will be returned if width > 0.

7.23.3.3 getEqs()

```
def vagen.hilevelmod.HiLevelMod.getEqs (
    self )
```

Return the equations in a format that can be imported by the maestro view.

Parameters

<i>self</i>	The object pointer.
-------------	---------------------

7.23.3.4 getOcn()

```
def vagen.hilevelmod.HiLevelMod.getOcn (
    self )
```

Return a ocean script that add equations to the opened session of adexl.

Parameters

<i>self</i>	The object pointer.
-------------	---------------------

7.23.3.5 ldc()

```
def vagen.hilevelmod.HiLevelMod ldc (
    self,
    name = "",
    width = 1,
    direction = "internal",
    value = 0,
    gnd = None,
    rise = 0,
    fall = 0 )
```

Return a [ldc](#) object or a [ldcBus](#) object if width > 1.

Parameters

<i>self</i>	The object pointer.
<i>name</i>	Name of the voltage source.
<i>width</i>	If width is greather than 1, It returns a list.
<i>direction</i>	It can be internal, input, output, or inout.
<i>value</i>	Real expression holding the inital value.
<i>gnd</i>	Electrical representing the ground reference.
<i>rise</i>	Real expression holding the initial rise time.
<i>fall</i>	Real expression holding the initial fall time.

Returns

[ldc](#) or [ldcBus](#) depending on the width.

7.23.3.6 marker()

```
def vagen.hilevelmod.HiLevelMod.marker (
    self,
    name,
    riseFall = 100e-12 )
```

Return a marker object.

Parameters

<i>self</i>	The object pointer.
<i>name</i>	Name of the marker.
<i>riseFall</i>	Rise and fall times of the marker pin. Default is 100ps.

Returns

[Marker](#) class.

7.23.3.7 seq()

```
def vagen.hilevelmod.HiLevelMod.seq (
    self,
    cond )
```

Sequence.

Parameters

<i>cond</i>	condition to run the sequence.
-------------	--------------------------------

Returns

function that accepts variable number of commands to be added to the sequence.

7.23.3.8 seqNested()

```
def vagen.hilevelmod.HiLevelMod.seqNested (
    self,
    cmdsIn )
```

Sequence.

Do not use it! Use Seq instead.

Parameters

<i>cmdsIn</i>	list of commands to be processed.
---------------	-----------------------------------

Returns

The list of remaining commands to be processed.

7.23.3.9 smu()

```
def vagen.hilevelmod.HiLevelMod.smu (
    self,
    name = "",
    width = 1,
    direction = "internal",
    volt = 0,
    minCur = 0,
    maxCur = 0,
    res = 1e12,
    gnd = None )
```

Return a [Smu](#) object or a [SmuBus](#) object if width > 1.

Parameters

<i>self</i>	The object pointer.
<i>name</i>	Name of the smu electrical pin.
<i>width</i>	If width is greather than 1, It returns a SmuBus .
<i>direction</i>	It can be internal, input, output, or inout.

Parameters

<i>volt</i>	Real expression holding the inital voltage.
<i>minCur</i>	Real expression holding the inital minimum current.
<i>maxCur</i>	Real expression holding the inital maximum current.
<i>res</i>	Real expression holding the resitance.

Returns

[Smu](#) or [SmuBus](#) depending on the width.

Parameters

<i>gnd</i>	Electrical representing the ground reference.
------------	-----------------------------------------------

7.23.3.10 `sw()`

```
def vagen.hilevelmod.HiLevelMod.sw (
    self,
    pin1,
    pin2,
    cond = 0.0,
    rise = 1e-6,
    fall = 1e-6 )
```

switch

Parameters

<i>self</i>	The object pointer.
<i>pin1</i>	First node (Electrical)
<i>pin2</i>	Second node (Electrical)
<i>cond</i>	Initial switch conductance. Default is 0S.
<i>rise</i>	Rise time for changes in the conductance. Default is 1us.
<i>fall</i>	Fall time for changes in the conductance. Default is 1us.

Returns

a [Sw](#) class.

7.23.3.11 `var()`

```
def vagen.hilevelmod.HiLevelMod.var (
    self,
    value = 0,
    name = "" )
```

Add variable to the module.

Also, the inital value of the variable will be set during the static analysis and the initial step of transient. The type of the variable will be compatible with the type of the initial value.

Parameters

<i>self</i>	The object pointer.
<i>name</i>	Name of the variable.
<i>value</i>	Initial value. Default is 0.

Returns

a variable class.

Reimplemented from [vagen.veriloga.Module](#).

7.23.3.12 vdc()

```
def vagen.hilevelmod.HiLevelMod.vdc (
    self,
    name = "",
    width = 1,
    direction = "internal",
    value = 0,
    gnd = None,
    rise = 0,
    fall = 0 )
```

Return a [Vdc](#) object or a [VdcBus](#) object if width > 1.

Parameters

<i>self</i>	The object pointer.
<i>name</i>	Name of the voltage source.
<i>width</i>	If width is greather than 1, It returns a list.
<i>direction</i>	It can be internal, input, output, or inout.
<i>value</i>	Real expression holding the inital value.
<i>gnd</i>	Electrical representing the ground reference.
<i>rise</i>	Real expression holding the initial rise time.
<i>fall</i>	Real expression holding the initial fall time.

Returns

[Vdc](#) or [VdcBus](#) depending on the width.

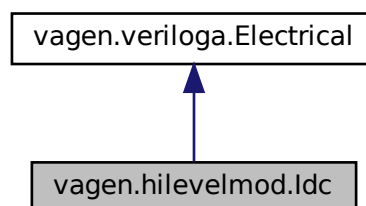
The documentation for this class was generated from the following file:

- [hilevelmod.py](#)

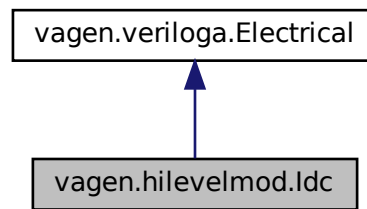
7.24 vagen.hilevelmod.Idc Class Reference

[Idc](#) class.

Inheritance diagram for vagen.hilevelmod.Idc:



Collaboration diagram for vagen.hilevelmod.Idc:



Public Member Functions

- def `__init__` (self, hiLevelMod, name, value, gnd, rise, fall)
Construtor.
- def `setRiseFall` (self, rise, fall)
Set the rise and the fall times for changes in the voltage.
- def `applyI` (self, value)
Change the value of the current source.

Public Attributes

- `cur`
- `rise`
- `fall`

7.24.1 Detailed Description

`Idc` class.

Child of `Electrical` implementing additional features in order to work as a current source.

7.24.2 Constructor & Destructor Documentation

7.24.2.1 `__init__()`

```
def vagen.hilevelmod.Idc.__init__ (
    self,
    hiLevelMod,
    name,
    value,
    gnd,
    rise,
    fall )
```

Construtor.

Parameters

<code>self</code>	The object pointer.
<code>hiLeveMod</code>	Hi level model in which the analog command will be added.
<code>name</code>	Name of the current source electrical pin.

Parameters

<i>value</i>	Real expression holding the initial voltage.
<i>gnd</i>	Electrical representing the ground reference.
<i>rise</i>	Real expression holding the initial rise time.
<i>fall</i>	Real expression holding the initial fall time.

7.24.3 Member Function Documentation

7.24.3.1 applyI()

```
def vagen.hilevelmod.Idc.applyI (
    self,
    value )
```

Change the value of the current source.

Parameters

<i>self</i>	The object pointer.
<i>value</i>	Teal expression holding the current.

Returns

The commands to change the current.

7.24.3.2 setRiseFall()

```
def vagen.hilevelmod.Idc.setRiseFall (
    self,
    rise,
    fall )
```

Set the rise and the fall times for changes in the voltage.

Parameters

<i>self</i>	The object pointer.
<i>rise</i>	Real expression holding the rise time for changes in the current.
<i>fall</i>	Real expression holding the fall time for changes in the current.

Returns

The commands to change the rise and fall times.

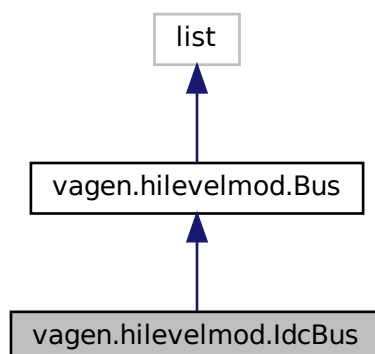
The documentation for this class was generated from the following file:

- [hilevelmod.py](#)

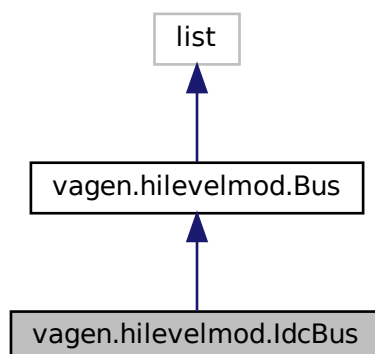
7.25 vagen.hilevelmod.IdcBus Class Reference

[IdcBus](#) class.

Inheritance diagram for vagen.hilevelmod.IdcBus:



Collaboration diagram for vagen.hilevelmod.IdcBus:



Public Member Functions

- def `__init__` (self)
Constructor.
- def `setRiseFall` (self, rise, fall)
Set the rise and the fall times for changes in the voltage.
- def `applyI` (self, value)
Change the value of the current source.

Additional Inherited Members

7.25.1 Detailed Description

`IdcBus` class.

Child of a list. It implements additional methods to deal with read and write operations to a bus. It also overrides the slice method, so it works similar to a slice of a bus in verilog.

7.25.2 Constructor & Destructor Documentation

7.25.2.1 `__init__()`

```
def vagen.hilevelmod.IdcBus.__init__ (
    self )
```

Constructor.

Parameters

<i>self</i>	The object pointer.
-------------	---------------------

7.25.3 Member Function Documentation

7.25.3.1 `applyI()`

```
def vagen.hilevelmod.IdcBus.applyI (
    self,
    value )
```

Change the value of the current source.

Parameters

<i>self</i>	The object pointer.
<i>value</i>	Real expression holding the current.

Returns

The commands to change the current.

7.25.3.2 `setRiseFall()`

```
def vagen.hilevelmod.IdcBus.setRiseFall (
    self,
    rise,
    fall )
```

Set the rise and the fall times for changes in the voltage.

Parameters

<i>self</i>	The object pointer.
<i>rise</i>	Real expression holding the rise time for changes in the current.
<i>fall</i>	Real expression holding the fall time for changes in the current.

Returns

The commands to change the rise and fall times.

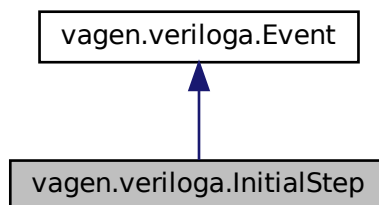
The documentation for this class was generated from the following file:

- [hilevelmod.py](#)

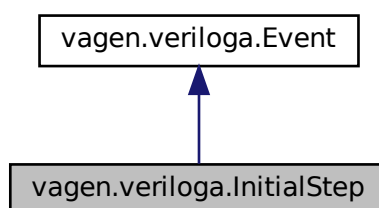
7.26 vagen.veriloga.InitialStep Class Reference

[InitialStep](#) class.

Inheritance diagram for vagen.veriloga.InitialStep:



Collaboration diagram for vagen.veriloga.InitialStep:



Public Member Functions

- `def __init__(self, *simTypes)`
Constructor.

Additional Inherited Members

7.26.1 Detailed Description

[InitialStep](#) class.

7.26.2 Constructor & Destructor Documentation

7.26.2.1 __init__()

```
def vagen.veriloga.InitialStep.__init__(
    self,
    * simTypes )
```

Constructor.

Parameters

<i>self</i>	object pointer
<i>*simTypes</i>	optional parameters representing the simulation type

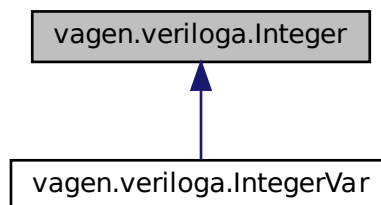
The documentation for this class was generated from the following file:

- [veriloga.py](#)

7.27 vagen.veriloga.Integer Class Reference

Class of [Integer](#) operators.

Inheritance diagram for vagen.veriloga.Integer:



Public Member Functions

- def [__init__](#) (self, value)
Constructor.
- def [getValue](#) (self)
Return the operator value.
- def [__add__](#) (self, other)
Addition override.
- def [__radd__](#) (self, other)
Reverse addition override.
- def [__sub__](#) (self, other)
Subtraction override.
- def [__rsub__](#) (self, other)
Reverse subtraction override.
- def [__mul__](#) (self, other)
Multiplication override.
- def [__rmul__](#) (self, other)
Reverse multiplication override.
- def [__truediv__](#) (self, other)
Division override.
- def [__rtruediv__](#) (self, other)
Reverse division override.
- def [__mod__](#) (self, other)
module override
- def [__rmod__](#) (self, other)

- reverse module override*
- def `__pow__` (self, other)
Pow override.
- def `__rpow__` (self, other)
Reverse pow override.
- def `__rshift__` (self, other)
right shift override.
- def `__rrshift__` (self, other)
Reverse right shift override.
- def `__lshift__` (self, other)
left shift override.
- def `__rlshift__` (self, other)
Reverse left shift override.
- def `__and__` (self, other)
Bitwise and logic.
- def `__rand__` (self, other)
Reverse bitwise and logic.
- def `__or__` (self, other)
Bitwise or logic.
- def `__ror__` (self, other)
Reverse bitwise or logic.
- def `__xor__` (self, other)
Bitwise xor logic.
- def `__rxor__` (self, other)
Reverse bitwise xor logic.
- def `__lt__` (self, other)
Less than override.
- def `__gt__` (self, other)
Greater than override.
- def `__le__` (self, other)
Less than equal override.
- def `__ge__` (self, other)
Greater than equal override.
- def `__eq__` (self, other)
Equal override.
- def `__ne__` (self, other)
Not equal override.
- def `__neg__` (self)
negation override
- def `__abs__` (self)
abs override
- def `__pos__` (self)
pos override
- def `__invert__` (self)
invert override
- def `__str__` (self)
str override

Public Attributes

- `value`

7.27.1 Detailed Description

Class of [Integer](#) operators.

7.27.2 Constructor & Destructor Documentation

7.27.2.1 `__init__()`

```
def vagen.veriloga.Integer.__init__ (
    self,
    value )
```

Constructor.

Parameters

<i>Self</i>	The object pointer.
<i>Value</i>	String representing a Real expression, an Integer , a Bool , or a value that can be converted to Integer .

Reimplemented in [vagen.veriloga.IntegerVar](#).

7.27.3 Member Function Documentation

7.27.3.1 `__abs__()`

```
def vagen.veriloga.Integer.__abs__ (
    self )
```

abs override

Parameters

<i>self</i>	Object pointer.
-------------	-----------------

Returns

expression representing absolute value.

7.27.3.2 `__add__()`

```
def vagen.veriloga.Integer.__add__ (
    self,
    other )
```

Addition override.

Parameters

<i>self</i>	The object pointer.
<i>other</i>	expression to be added.

Returns

expression representing the addition.

7.27.3.3 __and__()

```
def vagen.veriloga.Integer.__and__ (
    self,
    other )
```

Bitwise and logic.

Parameters

<i>self</i>	first operator.
<i>other</i>	second operator.

Returns

expression representing the bitwise and.

7.27.3.4 __eq__()

```
def vagen.veriloga.Integer.__eq__ (
    self,
    other )
```

Equal override.

Parameters

<i>self</i>	Left operand object pointer.
<i>other</i>	Right operand.

Returns

expression representing the comparison.

7.27.3.5 __ge__()

```
def vagen.veriloga.Integer.__ge__ (
    self,
    other )
```

Greater than equal override.

Parameters

<i>self</i>	Left operand object pointer.
<i>other</i>	Right operand.

Returns

expression representing the comparison.

7.27.3.6 __gt__()

```
def vagen.veriloga.Integer.__gt__ (
```

```

        self,
        other )

```

Greater than override.

Parameters

<i>self</i>	Left operand object pointer.
<i>other</i>	Right operand.

Returns

expression representing the comparison.

7.27.3.7 `__invert__()`

```

def vagen.veriloga.Integer.__invert__ (
    self )

```

invert override

Parameters

<i>self</i>	Object pointer.
-------------	-----------------

Returns

expression representing bitwise not in all bits

7.27.3.8 `__le__()`

```

def vagen.veriloga.Integer.__le__ (
    self,
    other )

```

Less than equal override.

Parameters

<i>self</i>	Left operand object pointer.
<i>other</i>	Right operand.

Returns

expression representing the comparison.

7.27.3.9 `__lshift__()`

```

def vagen.veriloga.Integer.__lshift__ (
    self,
    other )

```

left shift override.

Parameters

<i>self</i>	Integer to be shifted.
<i>other</i>	number of times the number will be shifted.

Returns

expression representing the shift.

7.27.3.10 __lt__()

```
def vagen.veriloga.Integer.__lt__ (
    self,
    other )
```

Less than override.

Parameters

<i>self</i>	Left operand object pointer.
<i>other</i>	Right operand.

Returns

expression representing the comparison.

7.27.3.11 __mod__()

```
def vagen.veriloga.Integer.__mod__ (
    self,
    other )
```

module override

Parameters

<i>self</i>	Dividend.
<i>other</i>	Quotient.

Returns

expression representing the module.

7.27.3.12 __mul__()

```
def vagen.veriloga.Integer.__mul__ (
    self,
    other )
```

Multiplication override.

Parameters

<i>self</i>	Multiplicand object pointer.
<i>other</i>	Multiplier.

Returns

expression representing the multiplication.

7.27.3.13 __ne__()

```
def vagen.veriloga.Integer.__ne__ (
```

```

        self,
        other )

```

Not equal override.

Parameters

<i>self</i>	Left operand object pointer.
<i>other</i>	Right operand.

Returns

expression representing the comparison.

7.27.3.14 `__neg__()`

```

def vagen.veriloga.Integer.__neg__ (
    self )

```

negation override

Parameters

<i>self</i>	Object pointer.
-------------	-----------------

Returns

expression representing negation.

7.27.3.15 `__or__()`

```

def vagen.veriloga.Integer.__or__ (
    self,
    other )

```

Bitwise or logic.

Parameters

<i>self</i>	first operator.
<i>other</i>	second operator.

Returns

expression representing the bitwise and.

7.27.3.16 `__pos__()`

```

def vagen.veriloga.Integer.__pos__ (
    self )

```

pos override

Parameters

<i>self</i>	Object pointer.
-------------	-----------------

Returns

copy of the same object.

7.27.3.17 __pow__()

```
def vagen.veriloga.Integer.__pow__ (
    self,
    other )
```

Pow override.

Parameters

<i>self</i>	Base object pointer.
<i>other</i>	Exponent.

Returns

expression representing the power.

7.27.3.18 __radd__()

```
def vagen.veriloga.Integer.__radd__ (
    self,
    other )
```

Reverse addition override.

Parameters

<i>self</i>	The object pointer.
<i>other</i>	expression to be added.

Returns

expression representing the addition.

7.27.3.19 __rand__()

```
def vagen.veriloga.Integer.__rand__ (
    self,
    other )
```

Reverse bitwise and logic.

Parameters

<i>self</i>	first operator.
<i>other</i>	second operator.

Returns

expression representing the bitwise and.

7.27.3.20 __rlshift__()

```
def vagen.veriloga.Integer.__rlshift__ (
```

```

        self,
        other )

```

Reverse left shift override.

Parameters

<i>self</i>	number of times the number will be shifted.
<i>other</i>	Integer to be shifted.

Returns

expression representing the shift.

7.27.3.21 __rmod__()

```

def vagen.veriloga.Integer.__rmod__ (
    self,
    other )

```

reverse module override

Parameters

<i>self</i>	Quotient.
<i>other</i>	Dividend.

Returns

expression representing the mdule.

7.27.3.22 __rmul__()

```

def vagen.veriloga.Integer.__rmul__ (
    self,
    other )

```

Reverse multiplication override.

Parameters

<i>self</i>	Multiplier object pointer.
<i>other</i>	Multiplicand.

Returns

expression representing the multiplication.

7.27.3.23 __ror__()

```

def vagen.veriloga.Integer.__ror__ (
    self,
    other )

```

Reverse bitwise or logic.

Parameters

<i>self</i>	first operator.
<i>other</i>	second operator.

Returns

expression representing the bitwise and.

7.27.3.24 __rpow__()

```
def vagen.veriloga.Integer.__rpow__ (
    self,
    other )
```

Reverse pow override.

Parameters

<i>self</i>	Exponent object pointer.
<i>other</i>	Base.

Returns

expression representing the power.

7.27.3.25 __rrshift__()

```
def vagen.veriloga.Integer.__rrshift__ (
    self,
    other )
```

Reverse right shift override.

Parameters

<i>self</i>	number of times the number will be shifted.
<i>other</i>	Integer to be shifted.

Returns

expression representing the shift.

7.27.3.26 __rshift__()

```
def vagen.veriloga.Integer.__rshift__ (
    self,
    other )
```

right shift override.

Parameters

<i>self</i>	Integer to be shifted.
<i>other</i>	number of times the number will be shifted.

Returns

expression representing the shift.

7.27.3.27 __rsub__()

```
def vagen.veriloga.Integer.__rsub__ (
```

```

        self,
        other )

```

Reverse subtraction override.

Parameters

<i>self</i>	Subtrahend object pointer.
<i>other</i>	Minuend.

Returns

expression representing the subtraction.

7.27.3.28 __rtruediv__()

```

def vagen.veriloga.Integer.__rtruediv__ (
    self,
    other )

```

Reverse division override.

Parameters

<i>self</i>	Quotient object pointer.
<i>other</i>	Dividend.

Returns

expression representing the division.

7.27.3.29 __rxor__()

```

def vagen.veriloga.Integer.__rxor__ (
    self,
    other )

```

Reverse bitwise xor logic.

Parameters

<i>self</i>	first operator.
<i>other</i>	second operator.

Returns

expression representing the bitwise and.

7.27.3.30 __str__()

```

def vagen.veriloga.Integer.__str__ (
    self )

```

str override

Parameters

<i>self</i>	Object pointer.
-------------	-----------------

Returns

string representing the expression

7.27.3.31 __sub__()

```
def vagen.veriloga.Integer.__sub__ (
    self,
    other )
```

Subtraction override.

Parameters

<i>self</i>	Minuend object pointer.
<i>other</i>	Subtrahend.

Returns

expression representing the subtraction.

7.27.3.32 __truediv__()

```
def vagen.veriloga.Integer.__truediv__ (
    self,
    other )
```

Division override.

Parameters

<i>self</i>	Dividend object pointer.
<i>other</i>	Quotient.

Returns

expression representing the division.

7.27.3.33 __xor__()

```
def vagen.veriloga.Integer.__xor__ (
    self,
    other )
```

Bitwise xor logic.

Parameters

<i>self</i>	first operator.
<i>other</i>	second operator.

Returns

expression representing the bitwise and.

7.27.3.34 getValue()

```
def vagen.veriloga.Integer.getValue (
    self )
```

Return the operator value.

Parameters

<i>Self</i>	The object pointer.
-------------	---------------------

Returns

String representing the [Bool](#) expression.

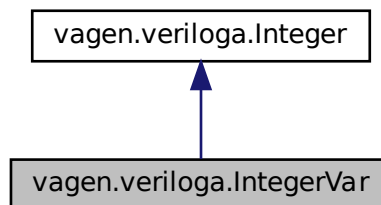
The documentation for this class was generated from the following file:

- [veriloga.py](#)

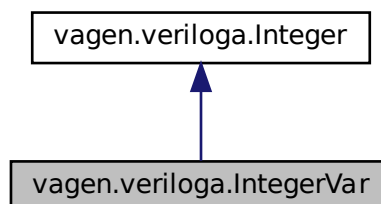
7.28 vagen.veriloga.IntegerVar Class Reference

[Integer](#) variable class.

Inheritance diagram for vagen.veriloga.IntegerVar:



Collaboration diagram for vagen.veriloga.IntegerVar:



Public Member Functions

- def `__init__` (self, value)
Constructor.
- def `inc` (self)
Increment.
- def `dec` (self)

- Decrement.*
- def `eq` (self, value)
- Atribution.*

Additional Inherited Members

7.28.1 Detailed Description

[Integer](#) variable class.

7.28.2 Constructor & Destructor Documentation

7.28.2.1 `__init__()`

```
def vagen.veriloga.IntegerVar.__init__ (
    self,
    value )
```

Constructor.

Parameters

<i>self</i>	object pointer
<i>value</i>	string representing the value

Reimplemented from [vagen.veriloga.Integer](#).

7.28.3 Member Function Documentation

7.28.3.1 `dec()`

```
def vagen.veriloga.IntegerVar.dec (
    self )
```

Decrement.

Parameters

<i>self</i>	object pointer
-------------	----------------

Returns

command representing the decrement

7.28.3.2 `eq()`

```
def vagen.veriloga.IntegerVar.eq (
    self,
    value )
```

Atribution.

Parameters

<i>self</i>	object pointer
<i>value</i>	A number representing the value

Returns

Return a command representing the attribution to a variable

7.28.3.3 inc()

```
def vagen.veriloga.IntegerVar.inc (
    self )
```

Increment.

Parameters

<i>self</i>	object pointer
-------------	----------------

Returns

command representing the increment

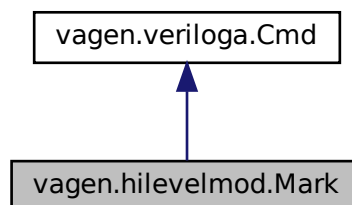
The documentation for this class was generated from the following file:

- [veriloga.py](#)

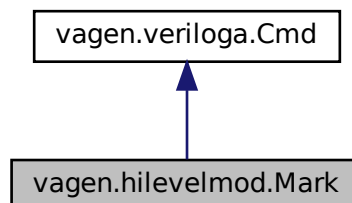
7.29 vagen.hilevelmod.Mark Class Reference

[Mark](#) command class.

Inheritance diagram for vagen.hilevelmod.Mark:



Collaboration diagram for vagen.hilevelmod.Mark:



Public Member Functions

- def `__init__` (self, cmd)
Construtor.
- def `getCmd` (self)
Return the command.
- def `__str__` (self)
Dummy method.
- def `getVA` (self, padding)
Dummy method.

Public Attributes

- `cmd`

7.29.1 Detailed Description

[Mark](#) command class.

This class of commands are responsible for storing the command thar marks an specific event

7.29.2 Constructor & Destructor Documentation

7.29.2.1 `__init__()`

```
def vagen.hilevelmod.Mark.__init__ (
    self,
    cmd )
```

Construtor.

Parameters

<i>self</i>	The object pointer.
<i>cmd</i>	Command to be added to the marker.

Reimplemented from [vagen.veriloga.Cmd](#).

7.29.3 Member Function Documentation

7.29.3.1 `__str__()`

```
def vagen.hilevelmod.Mark.__str__ (
    self )
```

Dummy method.

Raise exception when runned.

Parameters

<i>self</i>	The object pointer.
-------------	---------------------

Reimplemented from [vagen.veriloga.Cmd](#).

7.29.3.2 `getCmd()`

```
def vagen.hilevelmod.Mark.getCmd (
```

self)

Return the command.

Parameters

<i>self</i>	The object pointer.
-------------	---------------------

Returns

Command passed to the constructor.

7.29.3.3 getVA()

```
def vagen.hilevelmod.Mark.getVA (
    self,
    padding )
```

Dummy method.

Raise exception when runned.

Parameters

<i>self</i>	The object pointer.
-------------	---------------------

Reimplemented from [vagen.veriloga.Cmd](#).

The documentation for this class was generated from the following file:

- [hilevelmod.py](#)

7.30 vagen.hilevelmod.Marker Class Reference

[Marker](#) class.

Public Member Functions

- def [__init__](#) (self, hiLevelMod, name, riseFall)
Construtor.
- def [getName](#) (self)
Return the name of the [Marker](#).
- def [mark](#) (self, name)
[Mark](#) a particular event by flipping the internal variable.
- def [low](#) (self)
Force the internal variable low.
- def [high](#) (self)
Force the internal variable high.
- def [getEqs](#) (self)
Return a dictionary with the cadence equations for the marker

Public Attributes

- **name**
- **markList**
- **markerPin**
- **markSt**

7.30.1 Detailed Description

[Marker](#) class.

Responsible for flipping the state of one variable to mark events and generates cadence equations that calculate the time of the events

7.30.2 Constructor & Destructor Documentation

7.30.2.1 __init__()

```
def vagen.hilevelmod.Marker.__init__ (
    self,
    hiLevelMod,
    name,
    riseFall )
```

Constructor.

Parameters

<i>self</i>	The object pointer.
<i>hiLevelMod</i>	Hi level model in which the analog command will be added
<i>name</i>	Name of the marker.
<i>riseFall</i>	Rise and fall times of the marker pin.

7.30.3 Member Function Documentation

7.30.3.1 getEqs()

```
def vagen.hilevelmod.Marker.getEqs (
    self )
```

Return a dictionary with the cadence equations for the marker

Parameters

<i>self</i>	The object pointer.
-------------	---------------------

Returns

The dictionary with the cadence equations.

7.30.3.2 getName()

```
def vagen.hilevelmod.Marker.getName (
    self )
```

Return the name of the [Marker](#).

Parameters

<i>self</i>	The object pointer.
-------------	---------------------

Returns

Name of the [Marker](#).

7.30.3.3 high()

```
def vagen.hilevelmod.Marker.high (
    self )
```

Force the internal variable high.

You shouldn't use because it will break the synchronism between the cadence equations and the events. It was implemented for usage in specific power down conditions only.

Parameters

<i>self</i>	The object pointer.
-------------	---------------------

Returns

The [Mark](#) command.

7.30.3.4 low()

```
def vagen.hilevelmod.Marker.low (
    self )
```

Force the internal variable low.

You shouldn't use because it will break the synchronism between the cadence equations and the events. It was implemented for usage in specific power down conditions only.

Parameters

<i>self</i>	The object pointer.
-------------	---------------------

Returns

The [Mark](#) command.

7.30.3.5 mark()

```
def vagen.hilevelmod.Marker.mark (
    self,
    name )
```

[Mark](#) a particular event by flipping the internal variable.

Parameters

<i>self</i>	The object pointer.
<i>name</i>	Name of the event.

Returns

The [Mark](#) command.

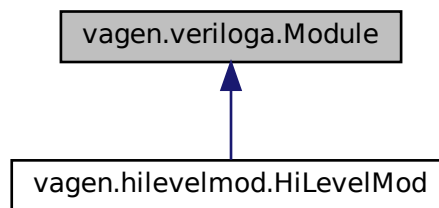
The documentation for this class was generated from the following file:

- [hilevelmod.py](#)

7.31 vagen.veriloga.Module Class Reference

verilogA class

Inheritance diagram for vagen.veriloga.Module:



Public Member Functions

- def [__init__](#) (self, moduleName)
constructor
- def [getModuleName](#) (self)
return module name
- def [fixName](#) (self, name)
If name is an empty string, get the next name available in the namespace.
- def [var](#) (self, vType=[Integer](#), name="")
Add variable to the module.
- def [par](#) (self, value, name)
Add parameter to the module.
- def [analog](#) (self, *args)
Add commands to the analog block.
- def [beginningAnalog](#) (self, *args)
Add commands to beginning of the analog block.
- def [endAnalog](#) (self, *args)
Add commands to the end of the analog block.
- def [addNode](#) (self, name, width, direction)
Add node.
- def [electrical](#) (self, name="", width=1, direction="internal")
Return electrical class.
- def [getVA](#) (self)
Return the VA verilog code.

Public Attributes

- `moduleName`
- `nameCount`
- `nameSpace`
- `nodes`
- `ports`
- `parameters`
- `variables`
- `cmds`
- `endCmds`
- `beginningCmds`

7.31.1 Detailed Description

verilogA class

7.31.2 Constructor & Destructor Documentation

7.31.2.1 `__init__()`

```
def vagen.veriloga.Module.__init__ (
    self,
    moduleName )
```

constructor

Parameters

<i>self</i>	The object pointer.
<i>node1</i>	Electrical signal representing the first node
<i>moduleName</i>	name of the module (the first word after module in the va)

7.31.3 Member Function Documentation

7.31.3.1 `addNode()`

```
def vagen.veriloga.Module.addNode (
    self,
    name,
    width,
    direction )
```

Add node.

Parameters

<i>self</i>	The object pointer.
<i>name</i>	string representing the name of the electrical signal
<i>width</i>	int representing the width of the electrical signal
<i>direction</i>	direction of the signal. It can be on the strings "internal", "input", "output", or "inout"

Returns

string with the name of the node

7.31.3.2 analog()

```
def vagen.veriloga.Module.analog (
    self,
    * args )
```

Add commands to the analog block.

Parameters

<i>self</i>	The object pointer.
<i>*args</i>	variable number of Cmd or CmdList to be added

7.31.3.3 beginningAnalog()

```
def vagen.veriloga.Module.beginningAnalog (
    self,
    * args )
```

Add commands to beginning of the analog block.

Parameters

<i>self</i>	The object pointer.
<i>*args</i>	variable number of Cmd or CmdList to be added

7.31.3.4 electrical()

```
def vagen.veriloga.Module.electrical (
    self,
    name = "",
    width = 1,
    direction = "internal" )
```

Return electrical class.

Parameters

<i>self</i>	The object pointer.
<i>name</i>	string representing the name of the electrical signal
<i>width</i>	int representing the width of the electrical signal
<i>direction</i>	direction of the signal. It can be on the strings "internal", "input", "output", or "inout"

Returns

list of electrical classes or an electrical class depending on the width

7.31.3.5 endAnalog()

```
def vagen.veriloga.Module.endAnalog (
    self,
    * args )
```

Add commands to the end of the analog block.

Parameters

<i>self</i>	The object pointer.
<i>*args</i>	variable number of Cmd or CmdList to be added

7.31.3.6 fixName()

```
def vagen.veriloga.Module.fixName (
    self,
    name )
```

If name is an empty string, get the next name available in the namespace.

If name isn't empty, check if the name is available in the veriloga namespace and raise an exception if it doesn't

Parameters

<i>self</i>	The object pointer.
<i>name</i>	string to be checked

Returns

string representing a valid name in the veriloga namespace

7.31.3.7 getModuleName()

```
def vagen.veriloga.Module.getModuleName (
    self )
return module name
```

Parameters

<i>self</i>	The object pointer
-------------	--------------------

Returns

string representing the name of the module

7.31.3.8 getVA()

```
def vagen.veriloga.Module.getVA (
    self )
```

Return the VA verilog code.

Parameters

<i>self</i>	The object pointer.
-------------	---------------------

Returns

string with the veriloga code

7.31.3.9 par()

```
def vagen.veriloga.Module.par (
    self,
```



```

        value,
        name )

```

Add parameter to the module.

Parameters

<i>self</i>	The object pointer.
<i>value</i>	Initial value. It can be Real , Integer , int or float.
<i>name</i>	string representing the name of the parameter in the veriloga

Returns

[RealVar](#) or [RealVar](#) depending on the initial value

7.31.3.10 var()

```

def vagen.veriloga.Module.var (
    self,
    vType = Integer,
    name = "" )

```

Add variable to the module.

Parameters

<i>self</i>	The object pointer.
<i>vType</i>	it can be Integer Bool or Real
<i>name</i>	string representing the name of the variable in the veriloga

Returns

[RealVar](#), [IntegerVar](#) or [BoolVar](#) depending on the vType

Reimplemented in [vagen.hilevelmod.HiLevelMod](#).

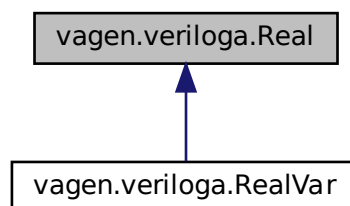
The documentation for this class was generated from the following file:

- [veriloga.py](#)

7.32 vagen.veriloga.Real Class Reference

Class of [Real](#) operators.

Inheritance diagram for vagen.veriloga.Real:



Public Member Functions

- def `__init__` (self, value)
Constructor.
- def `getValue` (self)
Return the operator value.
- def `__add__` (self, other)
Addition override.
- def `__sub__` (self, other)
Subtraction override.
- def `__mul__` (self, other)
Multiplication override.
- def `__truediv__` (self, other)
Division override.
- def `__pow__` (self, other)
Pow override.
- def `__gt__` (self, other)
Greater than override.
- def `__lt__` (self, other)
Less than override.
- def `__le__` (self, other)
Less than equal override.
- def `__ge__` (self, other)
Greater than equal override.
- def `__eq__` (self, other)
Equal override.
- def `__ne__` (self, other)
Not equal override.
- def `__radd__` (self, other)
Reverse addition override.
- def `__rsub__` (self, other)
Reverse subtraction override.
- def `__rmul__` (self, other)
Reverse multiplication override.
- def `__rtruediv__` (self, other)
Reverse division override.
- def `__rpow__` (self, other)
Pow override.
- def `__neg__` (self)
negation override
- def `__pos__` (self)
pos override
- def `__abs__` (self)
abs override
- def `__str__` (self)
str override

Public Attributes

- `value`

7.32.1 Detailed Description

Class of [Real](#) operators.

7.32.2 Constructor & Destructor Documentation

7.32.2.1 `__init__()`

```
def vagen.veriloga.Real.__init__ (
    self,
    value )
```

Constructor.

Parameters

<i>self</i>	The object pointer.
<i>value</i>	String representing a Real expression, an Integer , a Bool , or a value that can be converted to Real .

Reimplemented in [vagen.veriloga.RealVar](#).

7.32.3 Member Function Documentation

7.32.3.1 `__abs__()`

```
def vagen.veriloga.Real.__abs__ (
    self )
```

abs override

Parameters

<i>self</i>	Object pointer.
-------------	-----------------

Returns

expression representing absolute value.

7.32.3.2 `__add__()`

```
def vagen.veriloga.Real.__add__ (
    self,
    other )
```

Addition override.

Parameters

<i>self</i>	The object pointer.
<i>other</i>	expression to be added.

Returns

expression representing the addition.

7.32.3.3 `__eq__()`

```
def vagen.veriloga.Real.__eq__ (
    self,
    other )
```

Equal override.

Parameters

<i>self</i>	Left operand object pointer.
<i>other</i>	Right operand.

Returns

expression representing the comparison.

7.32.3.4 `__ge__()`

```
def vagen.veriloga.Real.__ge__ (
    self,
    other )
```

Greater than equal override.

Parameters

<i>self</i>	Left operand object pointer.
<i>other</i>	Right operand.

Returns

expression representing the comparison.

7.32.3.5 `__gt__()`

```
def vagen.veriloga.Real.__gt__ (
    self,
    other )
```

Greater than override.

Parameters

<i>self</i>	Left operand object pointer.
<i>other</i>	Right operand.

Returns

expression representing the comparison.

7.32.3.6 `__le__()`

```
def vagen.veriloga.Real.__le__ (
```

```
self,  
other )
```

Less than equal override.

Parameters

<i>self</i>	Left operand object pointer.
<i>other</i>	Right operand.

Returns

expression representing the comparison.

7.32.3.7 `__lt__()`

```
def vagen.veriloga.Real.__lt__ (  
    self,  
    other )
```

Less than override.

Parameters

<i>self</i>	Left operand object pointer.
<i>other</i>	Right operand.

Returns

expression representing the comparison.

7.32.3.8 `__mul__()`

```
def vagen.veriloga.Real.__mul__ (  
    self,  
    other )
```

Multiplication override.

Parameters

<i>self</i>	Multiplicand object pointer.
<i>other</i>	Multiplier.

Returns

expression representing the multiplication.

7.32.3.9 `__ne__()`

```
def vagen.veriloga.Real.__ne__ (  
    self,  
    other )
```

Not equal override.

Parameters

<i>self</i>	Left operand object pointer.
<i>other</i>	Right operand.

Returns

expression representing the comparison.

7.32.3.10 __neg__()

```
def vagen.veriloga.Real.__neg__ (
    self )
```

negation override

Parameters

<i>self</i>	Object pointer.
-------------	-----------------

Returns

expression representing negation.

7.32.3.11 __pos__()

```
def vagen.veriloga.Real.__pos__ (
    self )
```

pos override

Parameters

<i>self</i>	Object pointer.
-------------	-----------------

Returns

copy of the same object.

7.32.3.12 __pow__()

```
def vagen.veriloga.Real.__pow__ (
    self,
    other )
```

Pow override.

Parameters

<i>self</i>	Base object pointer.
<i>other</i>	Exponent.

Returns

expression representing the power.

7.32.3.13 __radd__()

```
def vagen.veriloga.Real.__radd__ (
    self,
    other )
```

Reverse addition override.

Parameters

<i>self</i>	The object pointer.
<i>other</i>	expression to be added.

Returns

expression representing the addition.

7.32.3.14 __rmul__()

```
def vagen.veriloga.Real.__rmul__ (
    self,
    other )
```

Reverse multiplication override.

Parameters

<i>self</i>	Multiplier object pointer.
<i>other</i>	Multiplicand.

Returns

expression representing the multiplication.

7.32.3.15 __rpow__()

```
def vagen.veriloga.Real.__rpow__ (
    self,
    other )
```

Pow override.

Parameters

<i>self</i>	Exponent object pointer.
<i>other</i>	Base.

Returns

expression representing the power.

7.32.3.16 __rsub__()

```
def vagen.veriloga.Real.__rsub__ (
    self,
    other )
```

Reverse subtraction override.

Parameters

<i>self</i>	Subtrahend object pointer.
<i>other</i>	Minuend.

Returns

expression representing the subtraction.

7.32.3.17 __rtruediv__()

```
def vagen.veriloga.Real.__rtruediv__ (
    self,
    other )
```

Reverse division override.

Parameters

<i>self</i>	Quotient object pointer.
<i>other</i>	Dividend.

Returns

expression representing the division.

7.32.3.18 __str__()

```
def vagen.veriloga.Real.__str__ (
    self )
```

str override

Parameters

<i>self</i>	Object pointer.
-------------	-----------------

Returns

string representing the expression

7.32.3.19 __sub__()

```
def vagen.veriloga.Real.__sub__ (
    self,
    other )
```

Subtraction override.

Parameters

<i>self</i>	Minuend object pointer.
<i>other</i>	Subtrahend.

Returns

expression representing the subtraction.

7.32.3.20 __truediv__()

```
def vagen.veriloga.Real.__truediv__ (
    self,
    other )
```


Division override.

Parameters

<i>self</i>	Dividend object pointer.
<i>other</i>	Quotient.

Returns

expression representing the division.

7.32.3.21 getValue()

```
def vagen.veriloga.Real.getValue (
    self )
```

Return the operator value.

Parameters

<i>self</i>	The object pointer.
-------------	---------------------

Returns

String representing the [Real](#) expression.

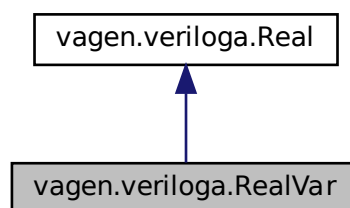
The documentation for this class was generated from the following file:

- [veriloga.py](#)

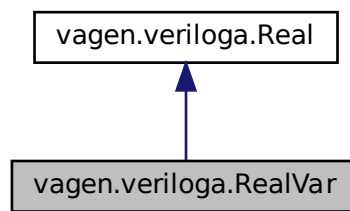
7.33 vagen.veriloga.RealVar Class Reference

[Real](#) variable class.

Inheritance diagram for vagen.veriloga.RealVar:



Collaboration diagram for vagen.veriloga.RealVar:



Public Member Functions

- def `__init__` (self, value)
Constructor.
- def `eq` (self, value)
Attribution.

Additional Inherited Members

7.33.1 Detailed Description

[Real](#) variable class.

7.33.2 Constructor & Destructor Documentation

7.33.2.1 `__init__()`

```
def vagen.veriloga.RealVar.__init__ (
    self,
    value )
```

Constructor.

Parameters

<i>self</i>	object pointer
<i>value</i>	string representing the value

Reimplemented from [vagen.veriloga.Real](#).

7.33.3 Member Function Documentation

7.33.3.1 `eq()`

```
def vagen.veriloga.RealVar.eq (
    self,
    value )
```

Attribution.

Parameters

<i>self</i>	object pointer
<i>value</i>	A number representing the value

Returns

Return a command representing the attribution to a variable

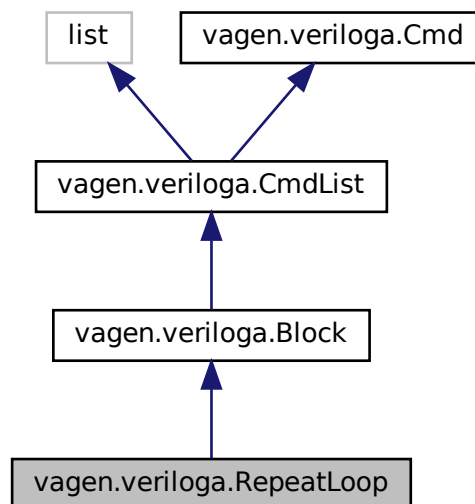
The documentation for this class was generated from the following file:

- [veriloga.py](#)

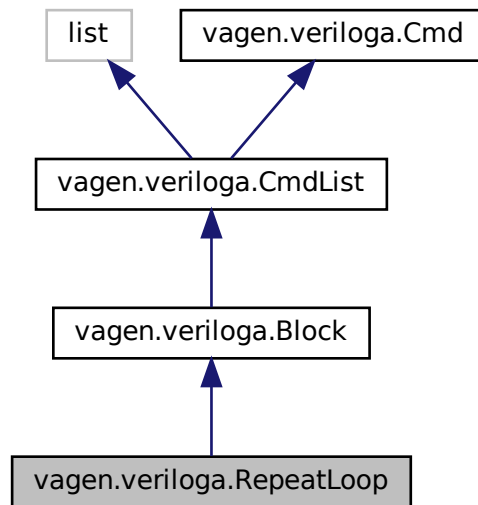
7.34 vagen.veriloga.RepeatLoop Class Reference

[RepeatLoop](#) class

Inheritance diagram for vagen.veriloga.RepeatLoop:



Collaboration diagram for vagen.veriloga.RepeatLoop:



Public Member Functions

- `def __init__ (self, n, *cmds)`
Constructor.
- `def getN (self)`
Return the repeat count.

Public Attributes

- `n`

7.34.1 Detailed Description

[RepeatLoop](#) class

7.34.2 Constructor & Destructor Documentation

7.34.2.1 __init__()

```
def vagen.veriloga.RepeatLoop.__init__ (
    self,
    n,
    * cmds )
```

Constructor.

Parameters

<i>self</i>	object pointer
<i>n</i>	Integer class or int representing the number of times the block of commands must be repeated
<i>*cmds</i>	variable number of Cmd or CmdList to be added

Reimplemented from [vagen.veriloga.Block](#).

7.34.3 Member Function Documentation

7.34.3.1 getN()

```
def vagen.veriloga.RepeatLoop.getN (
    self )
```

Return the repeat count.

Parameters

<i>self</i>	object pointer
-------------	----------------

Returns

[Integer](#) class representing the number of times the block of commands will be repeated

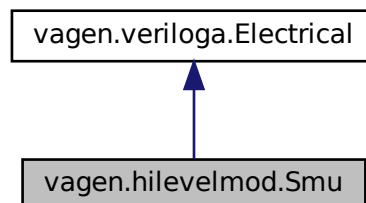
The documentation for this class was generated from the following file:

- [veriloga.py](#)

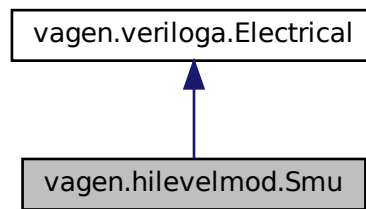
7.35 vagen.hilevelmod.Smu Class Reference

[Smu](#) class.

Inheritance diagram for vagen.hilevelmod.Smu:



Collaboration diagram for vagen.hilevelmod.Smu:



Public Member Functions

- `def __init__ (self, hiLevelMod, name, volt, minCur, maxCur, res, gnd)`
Construtor.
- `def applyV (self, value, limit)`
Configure the smu as current limited voltage source and apply the desired voltage.
- `def applyI (self, value, limit)`
Configure the smu as voltage limited current source and apply the desired current.
- `def applyR (self, value)`
Configure the resistive load.

Public Attributes

- `volt`
- `maxCur`
- `minCur`
- `res`
- `vDelay`
- `iDelay`
- `rDelay`
- `riseFall`

7.35.1 Detailed Description

`Smu` class.

Child of `Electrical` implementing additional features in order to work as a Source Measure Unit

7.35.2 Constructor & Destructor Documentation

7.35.2.1 __init__()

```

def vagen.hilevelmod.Smu.__init__ (
    self,
    hiLevelMod,
    name,
    volt,
    minCur,
    maxCur,

```

```

        res,
        gnd )

```

Construtor.

Parameters

<i>self</i>	The object pointer.
<i>hiLeveMod</i>	Hi level model in which the analog command will be added.
<i>name</i>	Name of the smu electrical pin.
<i>volt</i>	Real expression holding the inital voltage.
<i>minCur</i>	Real expression holding the inital minimum current.
<i>maxCur</i>	Real expression holding the inital maximum current.
<i>res</i>	Real expression holding the resitance.
<i>gnd</i>	Electrical representing the ground reference.

7.35.3 Member Function Documentation

7.35.3.1 applyI()

```

def vagen.hilevelmod.Smu.applyI (
    self,
    value,
    limit )

```

Configure the smu as voltage limited current source and apply the desired current.

Positive currents are sink current sources. The limit corresponds to the uppper voltage when value < 0 and to the lower voltage when value > 0.

Parameters

<i>self</i>	The object pointer.
<i>value</i>	Real expression holding the current to be applied.
<i>limit</i>	Real expression holding the voltage limit.

Returns

The commands to configure the [Smu](#) in current mode.

7.35.3.2 applyR()

```

def vagen.hilevelmod.Smu.applyR (
    self,
    value )

```

Configure the resistive load.

Parameters

<i>self</i>	The object pointer.
<i>value</i>	Real expression holding the value of the resistor.

Returns

The commands to configure the [Smu](#) in resistance mode.

7.35.3.3 applyV()

```
def vagen.hilevelmod.Smu.applyV (
    self,
    value,
    limit )
```

Configure the smu as current limited voltage source and apply the desired voltage.

Parameters

<i>self</i>	The object pointer.
<i>value</i>	Real expression holding the voltage to be applied.
<i>limit</i>	Real expression holding the current limit.

Returns

The commands to configure the [Smu](#) in voltage mode.

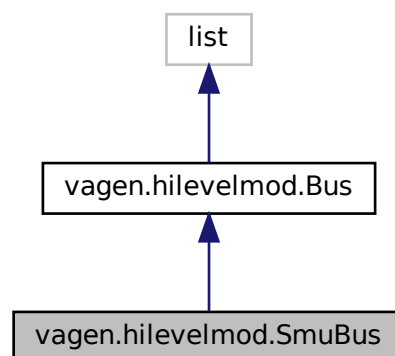
The documentation for this class was generated from the following file:

- [hilevelmod.py](#)

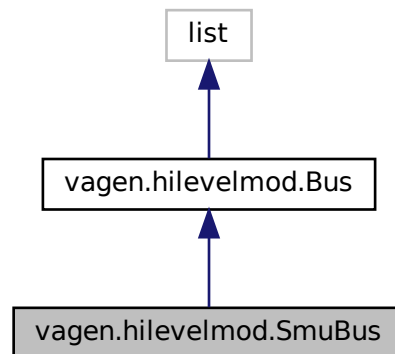
7.36 vagen.hilevelmod.SmuBus Class Reference

[SmuBus](#) class.

Inheritance diagram for vagen.hilevelmod.SmuBus:



Collaboration diagram for vagen.hilevelmod.SmuBus:



Public Member Functions

- def `__init__` (self)
Constructor.
- def `applyI` (self, value, limit)
Configure the smu as voltage limited current source and apply the desired current.
- def `applyV` (self, value, limit)
Configure the smu as current limited voltage source and apply the desired voltage.
- def `applyR` (self, value)
Configure the resistive load.

Additional Inherited Members

7.36.1 Detailed Description

`SmuBus` class.

Child of a list. It implements additional methods to deal with read and write operations to a bus. It also overrides the `slice` method, so it works similar to a slice of a bus in verilog.

7.36.2 Constructor & Destructor Documentation

7.36.2.1 `__init__()`

```
def vagen.hilevelmod.SmuBus.__init__ (
    self )
```

Constructor.

Parameters

<code>self</code>	The object pointer.
-------------------	---------------------

7.36.3 Member Function Documentation

7.36.3.1 applyI()

```
def vagen.hilevelmod.SmuBus.applyI (
    self,
    value,
    limit )
```

Configure the smu as voltage limited current source and apply the desired current.

Positive currents are sink current sources. The limit corresponds to the upper voltage when value < 0 and to the lower voltage when value > 0.

Parameters

<i>self</i>	The object pointer.
<i>value</i>	Real expression holding the current to be applied.
<i>limit</i>	Real expression holding the voltage limit.

Returns

The commands to configure the [Smu](#) in current mode.

7.36.3.2 applyR()

```
def vagen.hilevelmod.SmuBus.applyR (
    self,
    value )
```

Configure the resistive load.

Parameters

<i>self</i>	The object pointer.
<i>value</i>	Real expression holding the value of the resistor.

Returns

The commands to configure the [Smu](#) in resistance mode.

7.36.3.3 applyV()

```
def vagen.hilevelmod.SmuBus.applyV (
    self,
    value,
    limit )
```

Configure the smu as current limited voltage source and apply the desired voltage.

Parameters

<i>self</i>	The object pointer.
<i>value</i>	Real expression holding the voltage to be applied.
<i>limit</i>	Real expression holding the current limit.

Returns

The commands to configure the [Smu](#) in voltage mode.

The documentation for this class was generated from the following file:

- [hilevelmod.py](#)

7.37 vagen.hilevelmod.Sw Class Reference

[Sw](#) class.

Public Member Functions

- def [__init__](#) (self, hiLevelMod, pin1, pin2, cond, rise, fall)
Construtor.
- def [setRiseFall](#) (self, rise, fall)
Set the rise and the fall times of all digital output pin.
- def [setCond](#) (self, cond)
Set the conductance.

Public Attributes

- **swCount**
- **cond**
- **rise**
- **fall**
- **branch**

Static Public Attributes

- int **swCount** = 1

7.37.1 Detailed Description

[Sw](#) class.

Switch between two nodes.

7.37.2 Constructor & Destructor Documentation

7.37.2.1 __init__()

```
def vagen.hilevelmod.Sw.__init__ (
    self,
    hiLevelMod,
    pin1,
    pin2,
    cond,
    rise,
    fall )
```

Construtor.

Parameters

<i>self</i>	The object pointer.
<i>hiLeveMod</i>	Hi level model in which the analog command will be added.

Parameters

<i>pin1</i>	First node
<i>pin2</i>	Second node
<i>cond</i>	Real expression representing the initial switch conductance
<i>rise</i>	Real expression representing the rise time for changes in the conductance
<i>fall</i>	Real expression representing the fall time for changes in the conductance

7.37.3 Member Function Documentation

7.37.3.1 setCond()

```
def vagen.hilevelmod.Sw.setCond (
    self,
    cond )
```

Set the conductance.

Parameters

<i>self</i>	The object pointer.
<i>cond</i>	Real expression holding the conductance value.

Returns

The commands to change the conductance.

7.37.3.2 setRiseFall()

```
def vagen.hilevelmod.Sw.setRiseFall (
    self,
    rise,
    fall )
```

Set the rise and the fall times of all digital output pin.

Parameters

<i>self</i>	The object pointer.
<i>Rise</i>	Real expression holding the rise time.
<i>Fall</i>	Real expression holding the fall time.

Returns

The commands to change the rise and fall times.

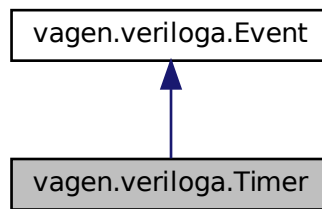
The documentation for this class was generated from the following file:

- [hilevelmod.py](#)

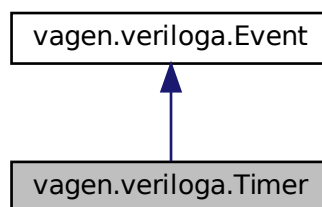
7.38 vagen.veriloga.Timer Class Reference

[Timer](#) Class.

Inheritance diagram for vagen.veriloga.Timer:



Collaboration diagram for vagen.veriloga.Timer:



Public Member Functions

- `def __init__ (self, startTime, *pars)`
Constructor.

Additional Inherited Members

7.38.1 Detailed Description

[Timer](#) Class.

7.38.2 Constructor & Destructor Documentation

7.38.2.1 `__init__()`

```
def vagen.veriloga.Timer.__init__ (
    self,
    startTime,
    * pars )
```

Constructor.

Parameters

<i>self</i>	object pointer
-------------	----------------

Parameters

<i>startTime</i>	Real or build-in real representing the time tolerance
<i>*pars</i>	optional Real or build-in real parameters timeTol and expTol in this order

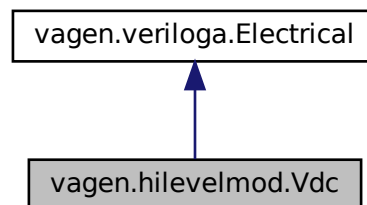
The documentation for this class was generated from the following file:

- [veriloga.py](#)

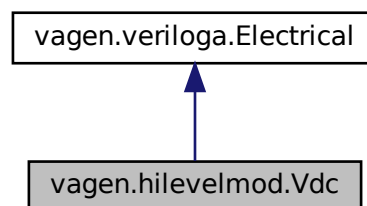
7.39 vagen.hilevelmod.Vdc Class Reference

[Vdc](#) class.

Inheritance diagram for vagen.hilevelmod.Vdc:



Collaboration diagram for vagen.hilevelmod.Vdc:



Public Member Functions

- def `__init__` (self, hiLevelMod, name, value, gnd, rise, fall)
Construtor.
- def `setRiseFall` (self, rise, fall)
Set the rise and the fall times for changes in the voltage.
- def `applyV` (self, value)
Change the value of the voltage source.

Public Attributes

- **volt**
- **rise**
- **fall**

7.39.1 Detailed Description

[Vdc](#) class.

Child of Electrical implementing additional features in order to work as a voltage source.

7.39.2 Constructor & Destructor Documentation

7.39.2.1 __init__()

```
def vagen.hilevelmod.Vdc.__init__ (
    self,
    hiLevelMod,
    name,
    value,
    gnd,
    rise,
    fall )
```

Constructor.

Parameters

<i>self</i>	The object pointer.
<i>hiLeveMod</i>	Hi level model in which the analog command will be added.
<i>name</i>	Name of the voltage source electrical pin.
<i>value</i>	Real expression holding the initial voltage.
<i>gnd</i>	Electrical representing the ground reference.
<i>rise</i>	Real expression holding the initial rise time.
<i>fall</i>	Real expression holding the initial fall time.

7.39.3 Member Function Documentation

7.39.3.1 applyV()

```
def vagen.hilevelmod.Vdc.applyV (
    self,
    value )
```

Change the value of the voltage source.

Parameters

<i>self</i>	The object pointer.
<i>value</i>	Real expression holding the voltage.

Returns

The commands to change the voltage.

7.39.3.2 setRiseFall()

```
def vagen.hilevelmod.Vdc.setRiseFall (
    self,
    rise,
    fall )
```

Set the rise and the fall times for changes in the voltage.

Parameters

<i>self</i>	The object pointer.
<i>rise</i>	Real expression holding the rise time for changes in the voltage.
<i>fall</i>	Real expression holding the fall time for changes in the voltage.

Returns

The commands to change the rise and fall times.

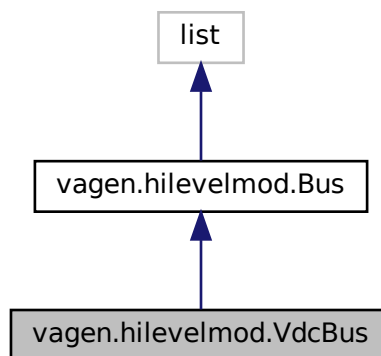
The documentation for this class was generated from the following file:

- [hilevelmod.py](#)

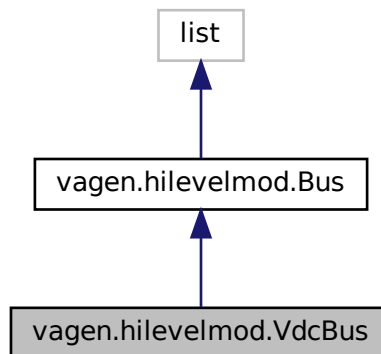
7.40 vagen.hilevelmod.VdcBus Class Reference

[VdcBus](#) class.

Inheritance diagram for vagen.hilevelmod.VdcBus:



Collaboration diagram for vagen.hilevelmod.VdcBus:



Public Member Functions

- def `__init__` (self)
Constructor.
- def `setRiseFall` (self, rise, fall)
Set the rise and the fall times for changes in the voltage.
- def `applyV` (self, value)
Change the value of the voltage source.

Additional Inherited Members

7.40.1 Detailed Description

`VdcBus` class.

Child of a list. It implements additional methods to deal with read and write operations to a bus. It also overrides the slice method, so it works similar to a slice of a bus in verilog.

7.40.2 Constructor & Destructor Documentation

7.40.2.1 `__init__`()

```
def vagen.hilevelmod.VdcBus.__init__ (
    self )
```

Constructor.

Parameters

<code>self</code>	The object pointer.
-------------------	---------------------

7.40.3 Member Function Documentation

7.40.3.1 applyV()

```
def vagen.hilevelmod.VdcBus.applyV (
    self,
    value )
```

Change the value of the voltage source.

Parameters

<i>self</i>	The object pointer.
<i>value</i>	Real expression holding the voltage.

Returns

The commands to change the voltage.

7.40.3.2 setRiseFall()

```
def vagen.hilevelmod.VdcBus.setRiseFall (
    self,
    rise,
    fall )
```

Set the rise and the fall times for changes in the voltage.

Parameters

<i>self</i>	The object pointer.
<i>rise</i>	Real expression holding the rise time for changes in the voltage.
<i>fall</i>	Real expression holding the fall time for changes in the voltage.

Returns

The commands to change the rise and fall times.

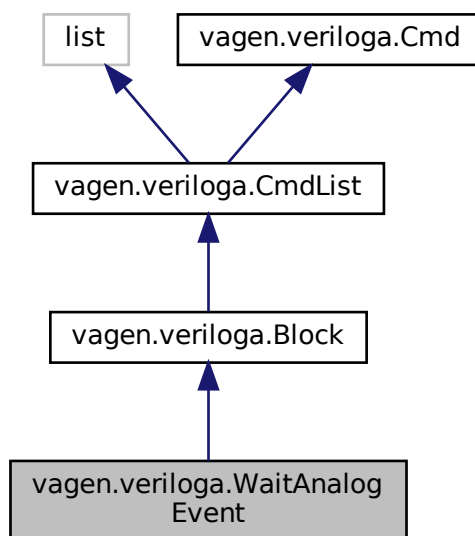
The documentation for this class was generated from the following file:

- [hilevelmod.py](#)

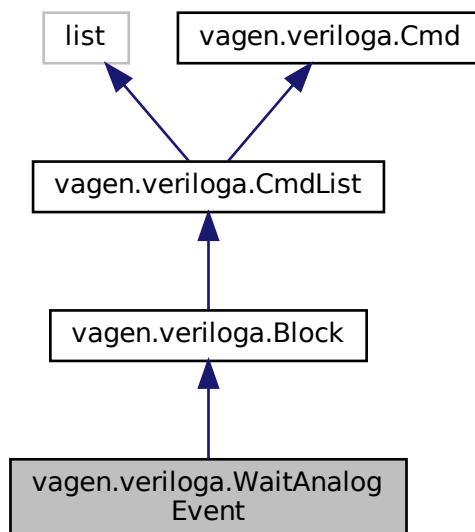
7.41 vagen.veriloga.WaitAnalogEvent Class Reference

Wait analog event class.

Inheritance diagram for vagen.veriloga.WaitAnalogEvent:



Collaboration diagram for vagen.veriloga.WaitAnalogEvent:



Public Member Functions

- `def __init__(self, event, *cmds)`
Constructor.

Additional Inherited Members

7.41.1 Detailed Description

Wait analog event class.

7.41.2 Constructor & Destructor Documentation

7.41.2.1 `__init__()`

```
def vagen.veriloga.WaitAnalogEvent.__init__ (
    self,
    event,
    * cmds )
```

Constructor.

Parameters

<i>self</i>	object pointer
<i>event</i>	Event to be waited for
<i>*cmds</i>	variable number of Cmd or CmdList to be added

Reimplemented from [vagen.veriloga.Block](#).

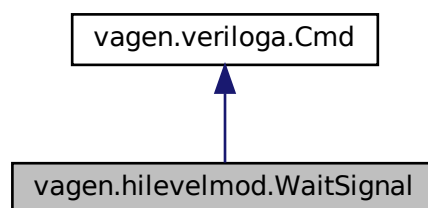
The documentation for this class was generated from the following file:

- [veriloga.py](#)

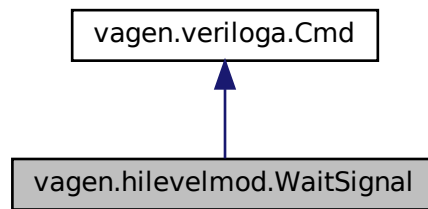
7.42 vagen.hilevelmod.WaitSignal Class Reference

[WaitSignal](#) command class.

Inheritance diagram for vagen.hilevelmod.WaitSignal:



Collaboration diagram for `vagen.hilevelmod.WaitSignal`:



Public Member Functions

- `def __init__ (self, evnt)`
Construtor.
- `def getEvt (self)`
Return the event that triggers the next state.
- `def __str__ (self)`
Dummy method.
- `def getVA (self, padding)`
Dummy method.

Public Attributes

- `evnt`

7.42.1 Detailed Description

[WaitSignal](#) command class.

This class of commands are responsible for wating a specific Event before allowing a test sequence to continue

7.42.2 Constructor & Destructor Documentation

7.42.2.1 __init__()

```
def vagen.hilevelmod.WaitSignal.__init__ (
    self,
    evnt )
```

Construtor.

Parameters

<i>self</i>	The object pointer.
<i>evnt</i>	Event to be waited for.

Reimplemented from [vagen.veriloga.Cmd](#).

7.42.3 Member Function Documentation

7.42.3.1 __str__()

```
def vagen.hilevelmod.WaitSignal.__str__ (
    self )
```

Dummy method.

Raise exception when runned.

Parameters

<i>self</i>	The object pointer.
-------------	---------------------

Reimplemented from [vagen.veriloga.Cmd](#).

7.42.3.2 getEvt()

```
def vagen.hilevelmod.WaitSignal.getEvt (
    self )
```

Return the event that triggers the next state.

Parameters

<i>self</i>	The object pointer.
-------------	---------------------

Returns

Event passed to the constructor.

7.42.3.3 getVA()

```
def vagen.hilevelmod.WaitSignal.getVA (
    self,
    padding )
```

Dummy method.

Raise exception when runned.

Parameters

<i>self</i>	The object pointer.
-------------	---------------------

Reimplemented from [vagen.veriloga.Cmd](#).

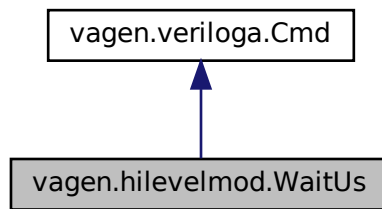
The documentation for this class was generated from the following file:

- [hilevelmod.py](#)

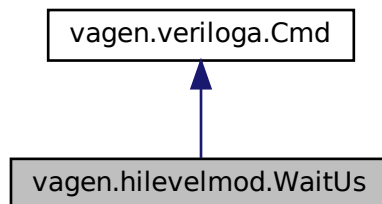
7.43 vagen.hilevelmod.WaitUs Class Reference

[WaitUs](#) command class.

Inheritance diagram for vagen.hilevelmod.WaitUs:



Collaboration diagram for vagen.hilevelmod.WaitUs:



Public Member Functions

- def `__init__` (self, delay)
Construtor.
- def `getDelay` (self)
Return the delay that triggers the next state.
- def `__str__` (self)
Dummy method.
- def `getVA` (self, padding)
Dummy method.

Public Attributes

- `delay`

7.43.1 Detailed Description

`WaitUs` command class.

This class of commands are responsible for wating a specific delay before allowing a test sequence to continue.

7.43.2 Constructor & Destructor Documentation

7.43.2.1 __init__()

```
def vagen.hilevelmod.WaitUs.__init__ (
    self,
    delay )
```

Construtor.

Parameters

<i>self</i>	The object pointer.
<i>delay</i>	Delay to be waited for.

Reimplemented from [vagen.veriloga.Cmd](#).

7.43.3 Member Function Documentation**7.43.3.1 __str__()**

```
def vagen.hilevelmod.WaitUs.__str__ (
    self )
```

Dummy method.

Raise exception when runned.

Parameters

<i>self</i>	The object pointer.
-------------	---------------------

Reimplemented from [vagen.veriloga.Cmd](#).

7.43.3.2 getDelay()

```
def vagen.hilevelmod.WaitUs.getDelay (
    self )
```

Return the delay that triggers the next state.

Parameters

<i>self</i>	The object pointer.
-------------	---------------------

Returns

Delay passed to the constructor.

7.43.3.3 getVA()

```
def vagen.hilevelmod.WaitUs.getVA (
    self,
    padding )
```

Dummy method.

Raise exception when runned.

Parameters

<i>self</i>	The object pointer.
-------------	---------------------

Reimplemented from [vagen.veriloga.Cmd](#).

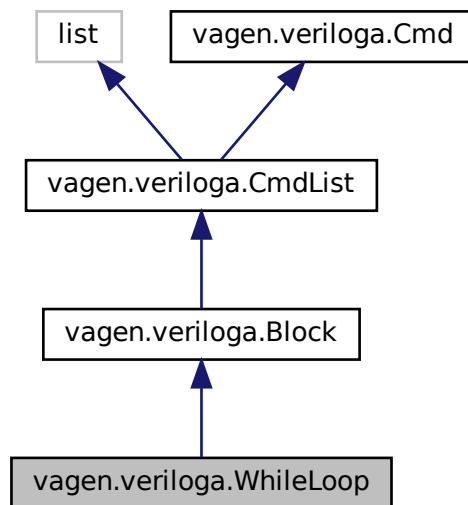
The documentation for this class was generated from the following file:

- [hilevelmod.py](#)

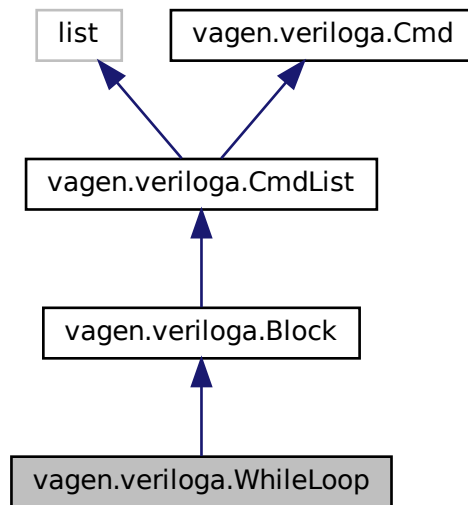
7.44 vagen.veriloga.WhileLoop Class Reference

[WhileLoop](#) class.

Inheritance diagram for vagen.veriloga.WhileLoop:



Collaboration diagram for vagen.veriloga.WhileLoop:



Public Member Functions

- def `__init__` (self, cond, *cmds)
Constructor.
- def `getCond` (self)
Return the while loop condition.

Public Attributes

- `cond`

7.44.1 Detailed Description

[WhileLoop](#) class.

7.44.2 Constructor & Destructor Documentation

7.44.2.1 `__init__()`

```
def vagen.veriloga.WhileLoop.__init__ (
    self,
    cond,
    * cmds )
```

Constructor.

Parameters

<i>self</i>	object pointer
<i>cond</i>	Bool class or build-in bool representing the condition that must be satisfied in order repeat the sequence of commands in the block

Parameters

<i>*cmds</i>	variable number of Cmd or CmdList to be added
--------------	-------------------------------------------------------------------------------

Reimplemented from [vagen.veriloga.Block](#).

7.44.3 Member Function Documentation

7.44.3.1 `getCond()`

```
def vagen.veriloga.WhileLoop.getCond (
    self )
```

Return the while loop condition.

Parameters

<i>self</i>	object pointer
-------------	----------------

Returns

[Bool](#) class representing the condition that must be satisfied in order repeat the sequence of commands in the block

The documentation for this class was generated from the following file:

- [veriloga.py](#)

Chapter 8

File Documentation

8.1 hilevelmod.py File Reference

Hi level modeling.

Classes

- class [vagen.hilevelmod.Mark](#)
Mark command class.
- class [vagen.hilevelmod.Marker](#)
Marker class.
- class [vagen.hilevelmod.WaitSignal](#)
WaitSignal command class.
- class [vagen.hilevelmod.WaitUs](#)
WaitUs command class.
- class [vagen.hilevelmod.Bus](#)
Bus class.
- class [vagen.hilevelmod.Vdc](#)
Vdc class.
- class [vagen.hilevelmod.VdcBus](#)
VdcBus class.
- class [vagen.hilevelmod.Idc](#)
Idc class.
- class [vagen.hilevelmod.IdcBus](#)
IdcBus class.
- class [vagen.hilevelmod.Smu](#)
Smu class.
- class [vagen.hilevelmod.SmuBus](#)
SmuBus class.
- class [vagen.hilevelmod.DigOut](#)
DigOut class.
- class [vagen.hilevelmod.DigIn](#)
DigIn class.
- class [vagen.hilevelmod.DigInOut](#)
DigInOut class.
- class [vagen.hilevelmod.DigBusOut](#)
DigBusOut class.
- class [vagen.hilevelmod.DigBusIn](#)
DigBusIn class.

- class [vagen.hilevelmod.DigBusInOut](#)
DigBusInOut class.
- class [vagen.hilevelmod.Sw](#)
Sw class.
- class [vagen.hilevelmod.Clock](#)
Clock class.
- class [vagen.hilevelmod.HiLevelMod](#)
HiLevelMod class.

8.1.1 Detailed Description

Hi level modeling.

8.1.2 License

Author

Rodrigo Pedroso Mendes

Version

V1.0

Date

14/02/23 13:37:31

Copyright (c) 2023 Rodrigo Pedroso Mendes

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

8.2 veriloga.py File Reference

VerilogA modeling.

Classes

- class [vagen.veriloga.Real](#)
Class of Real operators.
- class [vagen.veriloga.Bool](#)
Class of Bool operators.
- class [vagen.veriloga.Integer](#)
Class of Integer operators.
- class [vagen.veriloga.IntegerVar](#)

- Integer variable class.*
- class `vagen.veriloga.RealVar`
 - Real variable class.*
- class `vagen.veriloga.BoolVar`
 - Boolean variable class.*
- class `vagen.veriloga.Event`
 - Class of events.*
- class `vagen.veriloga.Cmd`
 - Command class.*
- class `vagen.veriloga.CmdList`
 - Command List class.*
- class `vagen.veriloga.Block`
 - Command Block Class.*
- class `vagen.veriloga.WaitAnalogEvent`
 - Wait analog event class.*
- class `vagen.veriloga.Cross`
 - Cross Class.*
- class `vagen.veriloga.Above`
 - Above Class.*
- class `vagen.veriloga.Timer`
 - Timer Class.*
- class `vagen.veriloga.InitialStep`
 - InitialStep class.*
- class `vagen.veriloga.FinalStep`
 - FinalStep class.*
- class `vagen.veriloga.RepeatLoop`
 - RepeatLoop class*
- class `vagen.veriloga.WhileLoop`
 - WhileLoop class.*
- class `vagen.veriloga.ForLoop`
 - ForLoop class.*
- class `vagen.veriloga.Cond`
 - Condition Class.*
- class `vagen.veriloga.CaseClass`
 - Condition Class.*
- class `vagen.veriloga.Electrical`
 - Class of electrical signals.*
- class `vagen.veriloga.Branch`
 - Branch class.*
- class `vagen.veriloga.Module`
 - verilogA class*

Functions

- def `vagen.veriloga.checkType` (param, var, Type)
 - Check if the type of variable matches the Type.*
- def `vagen.veriloga.checkInstance` (param, var, Type)
 - Check if the variable is an instance of Type.*
- def `vagen.veriloga.checkNotInstance` (param, var, Type)
 - Check if the variable isn't an instance of Type.*

- def [vagen.veriloga.parseReal](#) (param, var)
Return a [Real](#) instance.
- def [vagen.veriloga.checkReal](#) (param, var)
Check if the var is [Real](#) or it can be parsed to [Real](#).
- def [vagen.veriloga.parseInteger](#) (param, var)
Return an [Integer](#) instance.
- def [vagen.veriloga.checkInteger](#) (param, var)
Check if the variable is [Integer](#) or can be parsed to [Integer](#).
- def [vagen.veriloga.parseBool](#) (param, var)
Return a [Bool](#) instance.
- def [vagen.veriloga.checkBool](#) (param, var)
Check if the variable is [Bool](#) or can be parsed to [Bool](#).
- def [vagen.veriloga.parseNumber](#) (param, var)
Return a [Real](#), [Integer](#) or Boolean instance.
- def [vagen.veriloga.checkNumber](#) (param, var)
Check if the variable is a number or can be parsed to [Bool](#).
- def [vagen.veriloga.blockComment](#) (padding, message, align="center")
Creates a comment block.
- def [vagen.veriloga.block](#) (header)
Returns the pointer to a function that add commands to an analog event.
- def [vagen.veriloga.At](#) (event)
Returns the pointer to a function that add commands to an analog event.
- def [vagen.veriloga.unfoldSimTypes](#) (*simTypes)
Unfold variable number of simulation types.
- def [vagen.veriloga.analysis](#) (*simTypes)
Type of analysis.
- def [vagen.veriloga.acStim](#) (mag, phase=0, simType="ac")
ac stimulus
- def [vagen.veriloga.Repeat](#) (n)
Returns the pointer to a function that add commands to an Repeat block.
- def [vagen.veriloga.While](#) (cond)
Returns the pointer to a function that add commands to a While loop.
- def [vagen.veriloga.For](#) (start, cond, inc)
Returns the pointer to a function that add commands to a [ForLoop](#).
- def [vagen.veriloga.If](#) (cond)
Returns the pointer to a function that add commands to a [Cond](#) Class.
- def [vagen.veriloga.Case](#) (test)
Returns the pointer to a function that add commands to a [Block](#).
- def [vagen.veriloga.unfoldParams](#) (*params)
Unfold variable number of parameters.
- def [vagen.veriloga.Strobe](#) (msg, *params)
Strobe.
- def [vagen.veriloga.Write](#) (msg, *params)
Write.
- def [vagen.veriloga.Fopen](#) (fileName)
Fopen.
- def [vagen.veriloga.Fclose](#) (desc)
Fclose.
- def [vagen.veriloga.Fstrobe](#) (desc, msg, *params)
Fstrobe.
- def [vagen.veriloga.Fwrite](#) (desc, msg, *params)

- Fwrite.*
- def `vagen.veriloga.Discontinuity` (degree=0)
discontinuity
- def `vagen.veriloga.Finish` ()
finish
- def `vagen.veriloga.BondStep` (step)
bond step
- def `vagen.veriloga.lastCrossing` (signal, threshold, edge='both')
last time a signal crossed a treshold
- def `vagen.veriloga.random` (seed)
random number generator
- def `vagen.veriloga.uDistInt` (seed, start, end)
Uniforme distribution random number generator.
- def `vagen.veriloga.uDistReal` (seed, start, end)
Uniforme distribution random number generator.
- def `vagen.veriloga.gaussDistInt` (seed, mean, std)
Gaussian distribution random number generator.
- def `vagen.veriloga.gaussDistReal` (seed, mean, std)
Gaussian distribution random number generator.
- def `vagen.veriloga.expDistInt` (seed, mean)
Exponential distribution random number generator.
- def `vagen.veriloga.expDistReal` (seed, mean)
Exponential distribution random number generator.
- def `vagen.veriloga.poissonDistInt` (seed, mean)
Poisson distribution random number generator.
- def `vagen.veriloga.poissonDistReal` (seed, mean)
Poisson distribution random number generator.
- def `vagen.veriloga.exp` (x)
Exponential function.
- def `vagen.veriloga.limexp` (x)
Limited Exponential function.
- def `vagen.veriloga.absDelay` (x, delay)
Absolute Delay.
- def `vagen.veriloga.transition` (x, delay=0, riseTime=1e-6, fallTime=1e-6)
transition filter
- def `vagen.veriloga.ternary` (test, op1, op2)
ternary function
- def `vagen.veriloga.slew` (x, riseSlope=10e-6, fallSlope=10e-6)
slew filter
- def `vagen.veriloga.ddt` (x)
Diferential function.
- def `vagen.veriloga.idt` (x, start=Real(0))
Integral function.
- def `vagen.veriloga.ceil` (x)
Ceil function.
- def `vagen.veriloga.floor` (x)
floor function
- def `vagen.veriloga.ln` (x)
natural log function
- def `vagen.veriloga.log` (x)
log function

- def [vagen.veriloga.sqrt](#) (x)
square root function
- def [vagen.veriloga.sin](#) (x)
sin function
- def [vagen.veriloga.cos](#) (x)
cos function
- def [vagen.veriloga.tan](#) (x)
tan function
- def [vagen.veriloga.asin](#) (x)
arc sin function
- def [vagen.veriloga.acos](#) (x)
arc cos function
- def [vagen.veriloga.atan](#) (x)
arc tan function
- def [vagen.veriloga.atan2](#) (x, y)
arc tanh2 function.
- def [vagen.veriloga.hypot](#) (x, y)
hypot function.
- def [vagen.veriloga.sinh](#) (x)
sinh function
- def [vagen.veriloga.cosh](#) (x)
cosh function
- def [vagen.veriloga.tanh](#) (x)
tanh function
- def [vagen.veriloga.asinh](#) (x)
arc sinh function
- def [vagen.veriloga.acosh](#) (x)
arc cosh function
- def [vagen.veriloga.atanh](#) (x)
arc tanh function

Variables

- list [vagen.veriloga.anaTypes](#)
types of analysis
- [vagen.veriloga.temp](#) = Real("\$temperature")
Constants for tasks that represents numbers.
- [vagen.veriloga.abstime](#) = Real("\$abstime")
- [vagen.veriloga.vt](#) = Real("\$vt")

8.2.1 Detailed Description

VerilogA modeling.

8.2.2 License

Author

Rodrigo Pedroso Mendes

Version

V1.0

Date

05/02/23 22:36:08

Copyright (c) 2023 Rodrigo Pedroso Mendes

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR other LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR other DEALINGS IN THE SOFTWARE.

8.2.3 Function Documentation**8.2.3.1 absDelay()**

```
def vagen.veriloga.absDelay (
    x,
    delay )
```

Absolute Delay.

Parameters

<i>x</i>	Real, float or int input
<i>delay</i>	Real, float or int delay input

Returns

Real expressing the absolute delay function

8.2.3.2 acos()

```
def vagen.veriloga.acos (
    x )
```

arc cos function

Parameters

<i>x</i>	Real, float or int angle input
----------	--------------------------------

Returns

Real expressing the arc cos function in radians

8.2.3.3 acosh()

```
def vagen.veriloga.acosh (
    x )
```

arc cosh function

Parameters

<i>x</i>	Real, float or int angle input
----------	--------------------------------

Returns

Real expressing the arc cosh function in radians

8.2.3.4 acStim()

```
def vagen.veriloga.acStim (
    mag,
    phase = 0,
    simType = "ac" )
```

ac stimulus

Parameters

<i>mag</i>	Real class or build-in real representing the magnitude
<i>phase</i>	Real class or build-in representing the phase (default it 0)
<i>simType</i>	string representing the simulation type (default is "ac")

Returns

Real expression representing the ac stimulus command

8.2.3.5 analysis()

```
def vagen.veriloga.analysis (
    * simTypes )
```

Type of analysis.

Parameters

<i>*simTypes</i>	optional parameters representing the simulation type
------------------	------------------------------------------------------

Returns

Bool expression representing the analysis type test

8.2.3.6 asin()

```
def vagen.veriloga.asin (
    x )
```

arc sin function

Parameters

<i>x</i>	Real, float or int angle input
----------	--------------------------------

Returns

Real expressing the arc sin function in radians

8.2.3.7 asinh()

```
def vagen.veriloga.asinh (
    x )
```

arc sinh function

Parameters

<i>x</i>	Real, float or int angle input
----------	--------------------------------

Returns

Real expressing the arc sinh function in radians

8.2.3.8 At()

```
def vagen.veriloga.At (
    event )
```

Returns the pointer to a function that add commands to an analog event.

Parameters

<i>event</i>	instance of the Event class representing the analog event
--------------	-----------------------------------------------------------

Returns

function pointer

8.2.3.9 atan()

```
def vagen.veriloga.atan (
    x )
```

arc tan function

Parameters

<i>x</i>	Real, float or int angle input
----------	--------------------------------

Returns

Real expressing the arc tan function in radians

8.2.3.10 atan2()

```
def vagen.veriloga.atan2 (
    x,
    y )
```

arc tanh2 function.

Equivalent to atan(x/y)

Parameters

<i>x</i>	Real, float or int angle input
<i>x</i>	Real, float or int angle input

Returns

Real expressing the arc tan function in radians

8.2.3.11 atanh()

```
def vagen.veriloga.atanh (
    x )
```

arc tanh function

Parameters

<i>x</i>	Real, float or int angle input
----------	--------------------------------

Returns

Real expressing the arc tanh function in radians

8.2.3.12 block()

```
def vagen.veriloga.block (
    header )
```

Returns the pointer to a function that add commands to an analog event.

Parameters

<i>header</i>	header of the block
---------------	---------------------

Returns

pointer to a function that creates a block object

8.2.3.13 blockComment()

```
def vagen.veriloga.blockComment (
    padding,
    message,
    align = "center" )
```

Creates a comment block.

Parameters

<i>message</i>	String representing the comment
<i>padding</i>	Number of tabs by which the text will be shifted left align center, left or right

Returns

The comment block.

8.2.3.14 BoundStep()

```
def vagen.veriloga.BoundStep (  
    step )
```

bond step

Parameters

<i>step</i>	Real, float or int representing the step
-------------	------------------------------------------

Returns

Cmd representing the BondStep

8.2.3.15 Case()

```
def vagen.veriloga.Case (  
    test )
```

Returns the pointer to a function that add commands to a Block.

Parameters

<i>variable</i>	under test of the case structure
-----------------	----------------------------------

Returns

pointer to a function that returns a CaseClass

8.2.3.16 ceil()

```
def vagen.veriloga.ceil (  
    x )
```

Ceil function.

Parameters

<i>x</i>	Real, float or int input
----------	--------------------------

Returns

Integer expressing the ceil function

8.2.3.17 checkBool()

```
def vagen.veriloga.checkBool (  
    param,  
    var )
```

Check if the variable is Bool or can be parsed to Bool.

Parameters

<i>param</i>	String representing the name of the variable.
<i>var</i>	Variable.

Returns

True if it can be parsed to Bool or False otherwise.

8.2.3.18 checkInstance()

```
def vagen.veriloga.checkInstance (
    param,
    var,
    Type )
```

Check if the variable is an instance of Type.
Raise an assertion error if it doesn't.

Parameters

<i>param</i>	String representing the name of the variable.
<i>var</i>	Variable.
<i>Type</i>	Type that the variable should match.

Returns

True if it is an instance or False otherwise.

8.2.3.19 checkInteger()

```
def vagen.veriloga.checkInteger (
    param,
    var )
```

Check if the variable is Integer or can be parsed to Integer.

Parameters

<i>param</i>	String representing the name of the variable.
<i>var</i>	Variable.

Returns

True if it can be parsed to Integer or False otherwise.

8.2.3.20 checkNotInstance()

```
def vagen.veriloga.checkNotInstance (
    param,
    var,
    Type )
```

Check if the variable isn't an instance of Type.
Raise an assertion error if it doesn't.

Parameters

<i>param</i>	String representing the name of the variable.
<i>var</i>	Variable.
<i>Type</i>	Type that the variable should match.

Returns

False if it is an instance or True otherwise.

8.2.3.21 checkNumber()

```
def vagen.veriloga.checkNumber (
    param,
    var )
```

Check if the variable is a number or can be parsed to Bool.

Parameters

<i>param</i>	String representing the name of the variable.
<i>var</i>	Variable.

Returns

True if it can be parsed to number or False otherwise.

8.2.3.22 checkReal()

```
def vagen.veriloga.checkReal (
    param,
    var )
```

Check if the var is Real or it can be parsed to Real.

Parameters

<i>param</i>	String representing the name of the variable.
<i>var</i>	Variable.

Returns

True if it can be parsed to Real or False otherwise.

8.2.3.23 checkType()

```
def vagen.veriloga.checkType (
    param,
    var,
    Type )
```

Check if the type of variable matches the Type.
Raise an assertion error if it doesn't.

Parameters

<i>param</i>	String representing the name of the variable.
<i>var</i>	Variable.
<i>Type</i>	Type that the variable should match.

Returns

True if it matches the type or False otherwise.

8.2.3.24 cos()

```
def vagen.veriloga.cos (
    x )
```

cos function

Parameters

<i>x</i>	Real, float or int angle in radians
----------	-------------------------------------

Returns

Real expressing the cos function

8.2.3.25 cosh()

```
def vagen.veriloga.cosh (
    x )
```

cosh function

Parameters

<i>x</i>	Real, float or int angle in radians
----------	-------------------------------------

Returns

Real expressing the cosh function

8.2.3.26 ddt()

```
def vagen.veriloga.ddt (
    x )
```

Diferential function.

Parameters

<i>x</i>	Real, float or int input
----------	--------------------------

Returns

Real expressing the diferential function

8.2.3.27 Discontinuity()

```
def vagen.veriloga.Discontinuity (
    degree = 0 )
```

discontinuity

Parameters

<i>degree</i>	Integer or int representing the degree of the derivative with discontinuity
---------------	-----------------------------------------------------------------------------

Returns

Cmd representing the discontinuity

8.2.3.28 exp()

```
def vagen.veriloga.exp (
    x )
```

Exponential function.

Parameters

<i>x</i>	Real, float or int input
----------	--------------------------

Returns

Real expressing the exponential function

8.2.3.29 expDistInt()

```
def vagen.veriloga.expDistInt (
    seed,
    mean )
```

Exponential distribution random number generator.

Parameters

<i>seed</i>	IntegerVar with the seed
<i>mean</i>	Integer or int representing the start of the range

Returns

random Integer

8.2.3.30 expDistReal()

```
def vagen.veriloga.expDistReal (
    seed,
    mean )
```

Exponential distribution random number generator.

Parameters

<i>seed</i>	IntegerVar with the seed
<i>mean</i>	Real, float or int representing the start of the range

Returns

random Real

8.2.3.31 Fclose()

```
def vagen.veriloga.Fclose (
    desc )
```

Fclose.

Parameters

<i>desc</i>	Integer or int representing the file descriptor
-------------	-------------------------------------------------

Returns

Cmd to close the file

8.2.3.32 Finish()

```
def vagen.veriloga.Finish ( )
finish
```

Returns

Cmd representing the finish

8.2.3.33 floor()

```
def vagen.veriloga.floor (
                        x )
```

floor function

Parameters

<i>x</i>	Real, float or int input
----------	--------------------------

Returns

Integer expressing the floor function

8.2.3.34 Fopen()

```
def vagen.veriloga.Fopen (
                        fileName )
```

Fopen.

Parameters

<i>fileName</i>	name of the file
-----------------	------------------

Returns

Integer representing the file descriptor

8.2.3.35 For()

```
def vagen.veriloga.For (
                        start,
                        cond,
                        inc )
```

Returns the pointer to a function that add commands to a ForLoop.

Parameters

<i>start</i>	command executed at the beggining
--------------	-----------------------------------

Parameters

<i>cond</i>	condition that must be satisfied in order repeat the sequence of commands in the block
<i>inc</i>	command executed at the end of each step

Returns

pointer to a function that returns a ForLoop class

8.2.3.36 Fstrobe()

```
def vagen.veriloga.Fstrobe (
    desc,
    msg,
    * params )
```

Fstrobe.

Parameters

<i>desc</i>	Integer or int representing the file descriptor
<i>msg</i>	message to be written
<i>*params</i>	variable number of parameters

Returns

Cmd representing the Fstrobe

8.2.3.37 Fwrite()

```
def vagen.veriloga.Fwrite (
    desc,
    msg,
    * params )
```

Fwrite.

Parameters

<i>desc</i>	Integer or int representing the file descriptor
<i>msg</i>	message to be written
<i>*params</i>	variable number of parameters

Returns

Cmd representing the FWrite

8.2.3.38 gaussDistInt()

```
def vagen.veriloga.gaussDistInt (
    seed,
    mean,
    std )
```

Gaussian distribution random number generator.

Parameters

<i>seed</i>	IntegerVar with the seed
<i>mean</i>	Integer or int representing the start of the range
<i>std</i>	Integer or int representing the end of the range

Returns

random Integer

8.2.3.39 gaussDistReal()

```
def vagen.veriloga.gaussDistReal (
    seed,
    mean,
    std )
```

Gaussian distribution random number generator.

Parameters

<i>seed</i>	IntegerVar with the seed
<i>mean</i>	Real, float or int representing the start of the range
<i>std</i>	Real, float or int representing the end of the range

Returns

random Real

8.2.3.40 hypot()

```
def vagen.veriloga.hypot (
    x,
    y )
```

hypot function.

Equivalent to $\sqrt{x*x + y*y}$

Parameters

<i>x</i>	Real, float or int angle input
<i>y</i>	Real, float or int angle input

Returns

Real expressing the hypot function

8.2.3.41 idt()

```
def vagen.veriloga.idt (
    x,
    start = Real(0) )
```

Integral function.

Parameters

<i>x</i>	Real, float or int input
----------	--------------------------

Parameters

<i>start</i>	Real, float or int input
--------------	--------------------------

Returns

Real expressing the integral function

8.2.3.42 If()

```
def vagen.veriloga.If (
    cond )
```

Returns the pointer to a function that add commands to a Cond Class.

Parameters

<i>cond</i>	condition that must be satisfied in order to run the sequence of commands in the block
-------------	----------------------------------------------------------------------------------------

Returns

pointer to a function that returns a Cmd class

8.2.3.43 lastCrossing()

```
def vagen.veriloga.lastCrossing (
    signal,
    threshold,
    edge = 'both' )
```

last time a signal crossed a treshold

Parameters

<i>signal</i>	Real, float or int representing the signal
<i>threshold</i>	Real, float or int representing the threshold that must be crossed
<i>edge</i>	It can be one the strings "rising", "falling" or "both"

Returns

Real class representing the last crossing.

8.2.3.44 limexp()

```
def vagen.veriloga.limexp (
    x )
```

Limited Exponential function.

Parameters

<i>x</i>	Real, float or int input
----------	--------------------------

Returns

Real expressing the limited exponential function

8.2.3.45 ln()

```
def vagen.veriloga.ln (  
    x )
```

natural log function

Parameters

<i>x</i>	Real, float or int input
----------	--------------------------

Returns

Real expressing the natural log function

8.2.3.46 log()

```
def vagen.veriloga.log (  
    x )
```

log function

Parameters

<i>x</i>	Real, float or int input
----------	--------------------------

Returns

Real expressing the log function

8.2.3.47 parseBool()

```
def vagen.veriloga.parseBool (  
    param,  
    var )
```

Return a Bool instance.

Parameters

<i>param</i>	String representing the name of the variable.
<i>var</i>	Variable.

Returns

Bool object.

8.2.3.48 parseInteger()

```
def vagen.veriloga.parseInteger (  
    param,  
    var )
```

Return an Integer instance.

Parameters

<i>param</i>	String representing the name of the variable.
<i>var</i>	Variable.

Returns

Integer object.

8.2.3.49 parseNumber()

```
def vagen.veriloga.parseNumber (
    param,
    var )
```

Return a Real, Integer or Boolean instance.

Parameters

<i>param</i>	String representing the name of the variable.
<i>var</i>	Variable.

Returns

Real, Integer or Bool instance.

8.2.3.50 parseReal()

```
def vagen.veriloga.parseReal (
    param,
    var )
```

Return a Real instance.

Parameters

<i>param</i>	String representing the name of the variable.
<i>var</i>	Variable.

Returns

Real object.

8.2.3.51 poissonDistInt()

```
def vagen.veriloga.poissonDistInt (
    seed,
    mean )
```

Poisson distribution random number generator.

Parameters

<i>seed</i>	IntegerVar with the seed
<i>mean</i>	Integer or int representing the start of the range

Returns

random Integer

8.2.3.52 poissonDistReal()

```
def vagen.veriloga.poissonDistReal (
```

```

    seed,
    mean )

```

Poisson distribution random number generator.

Parameters

<i>seed</i>	IntegerVar with the seed
<i>mean</i>	Real, float or int representing the start of the range

Returns

random Real

8.2.3.53 random()

```

def vagen.veriloga.random (
    seed )

```

random number generator

Parameters

<i>seed</i>	IntegerVar with the seed
-------------	--------------------------

Returns

random Integer

8.2.3.54 Repeat()

```

def vagen.veriloga.Repeat (
    n )

```

Returns the pointer to a function that add commands to an Repeat block.

Parameters

<i>n</i>	Integer class or int representing the number of times the sequence must be repeated
----------	-------------------------------------------------------------------------------------

Returns

pointer to a function that returns a RepeatLoop class

8.2.3.55 sin()

```

def vagen.veriloga.sin (
    x )

```

sin function

Parameters

<i>x</i>	Real, float or int angle in radians
----------	-------------------------------------

Returns

Real expressing the sin function

8.2.3.56 sinh()

```
def vagen.veriloga.sinh (
    x )
```

sinh function

Parameters

<i>x</i>	Real, float or int angle in radians
----------	-------------------------------------

Returns

Real expressing the sinh function

8.2.3.57 slew()

```
def vagen.veriloga.slew (
    x,
    riseSlope = 10e-6,
    fallSlope = 10e-6 )
```

slew filter

Parameters

<i>x</i>	Real, float or int input
<i>riseSlope</i>	Real, float or int delay input
<i>fallSlope</i>	Real, float or int delay input

Returns

Real expressing the slew filter

8.2.3.58 sqrt()

```
def vagen.veriloga.sqrt (
    x )
```

square root function

Parameters

<i>x</i>	Real, float or int input
----------	--------------------------

Returns

Real expressing the square root function

8.2.3.59 Strobe()

```
def vagen.veriloga.Strobe (
    msg,
    * params )
```

Strobe.

Parameters

<i>msg</i>	message to be printed
------------	-----------------------

Parameters

<i>*params</i>	variable number of parameters
----------------	-------------------------------

Returns

Cmd representing the strobe

8.2.3.60 tan()

```
def vagen.veriloga.tan (
    x )
```

tan function

Parameters

<i>x</i>	Real, float or int angle in radians
----------	-------------------------------------

Returns

Real expressing the tan function

8.2.3.61 tanh()

```
def vagen.veriloga.tanh (
    x )
```

tanh function

Parameters

<i>x</i>	Real, float or int angle in radians
----------	-------------------------------------

Returns

Real expressing the tanh function

8.2.3.62 ternary()

```
def vagen.veriloga.ternary (
    test,
    op1,
    op2 )
```

ternary function

Parameters

<i>test</i>	Bool or bool representing the test
<i>op1</i>	any Bool, Real, Integer, int, float or bool that represents the expression when test is true
<i>op2</i>	any Bool, Real, Integer, int, float or bool that represents the expression when test is false

Returns

Bool, Real or Integer representing the ternary operator

8.2.3.63 transition()

```
def vagen.veriloga.transition (
    x,
    delay = 0,
    riseTime = 1e-6,
    fallTime = 1e-6 )
```

transition filter

Parameters

<i>x</i>	Real, float or int input
<i>delay</i>	Real, float or int delay input
<i>riseTime</i>	delay Real, float or int delay input
<i>fallTime</i>	delay Real, float or int delay input

Returns

Real expressing the transition filter

8.2.3.64 uDistInt()

```
def vagen.veriloga.uDistInt (
    seed,
    start,
    end )
```

Uniforme distribution random number generator.

Parameters

<i>seed</i>	IntegerVar with the seed
<i>start</i>	Integer or int representing the start of the range
<i>end</i>	Integer or int representing the end of the range

Returns

random Integer

8.2.3.65 uDistReal()

```
def vagen.veriloga.uDistReal (
    seed,
    start,
    end )
```

Uniforme distribution random number generator.

Parameters

<i>seed</i>	IntegerVar with the seed
<i>start</i>	Real, float or int representing the start of the range
<i>end</i>	Real, float or int representing the end of the range

Returns

random Real

8.2.3.66 unfoldParams()

```
def vagen.veriloga.unfoldParams (
    * params )
```

Unfold variable number of parameters.

Parameters

<i>*params</i>	variable number of parameters
----------------	-------------------------------

Returns

string representing the parameters separated by comma

8.2.3.67 unfoldSimTypes()

```
def vagen.veriloga.unfoldSimTypes (
    * simTypes )
```

Unfold variable number of simulation types.

Parameters

<i>list</i>	of simulation types
-------------	---------------------

Returns

string with the unfolded simulation types

8.2.3.68 While()

```
def vagen.veriloga.While (
    cond )
```

Returns the pointer to a function that add commands to a While loop.

Parameters

<i>cond</i>	Bool class or build-in bool representing the condition that must be satisfied in order repeat the sequence of commands in the block
-------------	-------------------------------------------------------------------------------------------------------------------------------------

Returns

pointer to a function that returns a WhileLoop class

8.2.3.69 Write()

```
def vagen.veriloga.Write (
    msg,
    * params )
```

Write.

Parameters

<i>msg</i>	message to be printed
<i>*params</i>	variable number of parameters

Returns

Cmd representing the Write

8.2.4 Variable Documentation

8.2.4.1 anaTypes

```
list vagen.veriloga.anaTypes
```

Initial value:

```
1 = ["ac",
2     "dc",
3     "ic",
4     "tran",
5     "pac",
6     "pnoise",
7     "pss",
8     "pxf",
9     "sp",
10    "static",
11    "tdr",
12    "xf"]
```

types of analysis

8.2.4.2 temp

```
vagen.veriloga.temp = Real("$temperature")
```

Constants for tasks that represents numbers.

Index

- [__abs__](#)
 - [vagen.veriloga.Integer](#), [78](#)
 - [vagen.veriloga.Real](#), [101](#)
- [__add__](#)
 - [vagen.veriloga.Integer](#), [78](#)
 - [vagen.veriloga.Real](#), [101](#)
- [__and__](#)
 - [vagen.veriloga.Bool](#), [18](#)
 - [vagen.veriloga.Integer](#), [79](#)
- [__eq__](#)
 - [vagen.veriloga.Bool](#), [18](#)
 - [vagen.veriloga.Integer](#), [79](#)
 - [vagen.veriloga.Real](#), [102](#)
- [__ge__](#)
 - [vagen.veriloga.Integer](#), [79](#)
 - [vagen.veriloga.Real](#), [102](#)
- [__getitem__](#)
 - [vagen.hilevelmod.Bus](#), [26](#)
- [__gt__](#)
 - [vagen.veriloga.Integer](#), [79](#)
 - [vagen.veriloga.Real](#), [102](#)
- [__init__](#)
 - [vagen.hilevelmod.Bus](#), [26](#)
 - [vagen.hilevelmod.Clock](#), [29](#)
 - [vagen.hilevelmod.DigBusIn](#), [40](#)
 - [vagen.hilevelmod.DigBusInOut](#), [42](#)
 - [vagen.hilevelmod.DigBusOut](#), [45](#)
 - [vagen.hilevelmod.DigIn](#), [47](#)
 - [vagen.hilevelmod.DigInOut](#), [50](#)
 - [vagen.hilevelmod.DigOut](#), [52](#)
 - [vagen.hilevelmod.HiLevelMod](#), [65](#)
 - [vagen.hilevelmod.Idc](#), [71](#)
 - [vagen.hilevelmod.IdcBus](#), [74](#)
 - [vagen.hilevelmod.Mark](#), [91](#)
 - [vagen.hilevelmod.Marker](#), [93](#)
 - [vagen.hilevelmod.Smu](#), [113](#)
 - [vagen.hilevelmod.SmuBus](#), [116](#)
 - [vagen.hilevelmod.Sw](#), [118](#)
 - [vagen.hilevelmod.Vdc](#), [122](#)
 - [vagen.hilevelmod.VdcBus](#), [124](#)
 - [vagen.hilevelmod.WaitSignal](#), [128](#)
 - [vagen.hilevelmod.WaitUs](#), [130](#)
 - [vagen.veriloga.Above](#), [14](#)
 - [vagen.veriloga.Block](#), [15](#)
 - [vagen.veriloga.Bool](#), [17](#)
 - [vagen.veriloga.BoolVar](#), [23](#)
 - [vagen.veriloga.Branch](#), [24](#)
 - [vagen.veriloga.CaseClass](#), [28](#)
 - [vagen.veriloga.Cmd](#), [31](#)
 - [vagen.veriloga.CmdList](#), [33](#)
 - [vagen.veriloga.Cond](#), [36](#)
 - [vagen.veriloga.Cross](#), [38](#)
 - [vagen.veriloga.Electrical](#), [55](#)
 - [vagen.veriloga.Event](#), [58](#)
 - [vagen.veriloga.FinalStep](#), [60](#)
 - [vagen.veriloga.ForLoop](#), [62](#)
 - [vagen.veriloga.InitialStep](#), [75](#)
 - [vagen.veriloga.Integer](#), [78](#)
 - [vagen.veriloga.IntegerVar](#), [89](#)
 - [vagen.veriloga.Module](#), [96](#)
 - [vagen.veriloga.Real](#), [101](#)
 - [vagen.veriloga.RealVar](#), [109](#)
 - [vagen.veriloga.RepeatLoop](#), [111](#)
 - [vagen.veriloga.Timer](#), [120](#)
 - [vagen.veriloga.WaitAnalogEvent](#), [127](#)
 - [vagen.veriloga.WhileLoop](#), [133](#)
- [__invert__](#)
 - [vagen.veriloga.Bool](#), [18](#)
 - [vagen.veriloga.Integer](#), [80](#)
- [__le__](#)
 - [vagen.veriloga.Integer](#), [80](#)
 - [vagen.veriloga.Real](#), [102](#)
- [__lshift__](#)
 - [vagen.veriloga.Integer](#), [80](#)
- [__lt__](#)
 - [vagen.veriloga.Integer](#), [81](#)
 - [vagen.veriloga.Real](#), [103](#)
- [__mod__](#)
 - [vagen.veriloga.Integer](#), [81](#)
- [__mul__](#)
 - [vagen.veriloga.Integer](#), [81](#)
 - [vagen.veriloga.Real](#), [103](#)
- [__ne__](#)
 - [vagen.veriloga.Bool](#), [18](#)
 - [vagen.veriloga.Integer](#), [81](#)
 - [vagen.veriloga.Real](#), [103](#)
- [__neg__](#)
 - [vagen.veriloga.Integer](#), [82](#)
 - [vagen.veriloga.Real](#), [104](#)
- [__or__](#)
 - [vagen.veriloga.Bool](#), [20](#)
 - [vagen.veriloga.Event](#), [59](#)
 - [vagen.veriloga.Integer](#), [82](#)
- [__pos__](#)
 - [vagen.veriloga.Integer](#), [82](#)
 - [vagen.veriloga.Real](#), [104](#)
- [__pow__](#)
 - [vagen.veriloga.Integer](#), [83](#)

- vagen.veriloga.Real, 104
- __radd__
 - vagen.veriloga.Integer, 83
 - vagen.veriloga.Real, 104
- __rand__
 - vagen.veriloga.Bool, 20
 - vagen.veriloga.Integer, 83
- __rlshift__
 - vagen.veriloga.Integer, 83
- __rmod__
 - vagen.veriloga.Integer, 84
- __rmul__
 - vagen.veriloga.Integer, 84
 - vagen.veriloga.Real, 105
- __ror__
 - vagen.veriloga.Bool, 20
 - vagen.veriloga.Integer, 84
- __rpow__
 - vagen.veriloga.Integer, 85
 - vagen.veriloga.Real, 105
- __rrshift__
 - vagen.veriloga.Integer, 85
- __rshift__
 - vagen.veriloga.Integer, 85
- __rsub__
 - vagen.veriloga.Integer, 85
 - vagen.veriloga.Real, 105
- __rtruediv__
 - vagen.veriloga.Integer, 86
 - vagen.veriloga.Real, 106
- __rxor__
 - vagen.veriloga.Bool, 21
 - vagen.veriloga.Integer, 86
- __str__
 - vagen.hilevelmod.Mark, 91
 - vagen.hilevelmod.WaitSignal, 128
 - vagen.hilevelmod.WaitUs, 131
 - vagen.veriloga.Bool, 21
 - vagen.veriloga.Cmd, 31
 - vagen.veriloga.CmdList, 33
 - vagen.veriloga.Event, 59
 - vagen.veriloga.Integer, 86
 - vagen.veriloga.Real, 106
- __sub__
 - vagen.veriloga.Integer, 87
 - vagen.veriloga.Real, 106
- __truediv__
 - vagen.veriloga.Integer, 87
 - vagen.veriloga.Real, 106
- __xor__
 - vagen.veriloga.Bool, 21
 - vagen.veriloga.Integer, 87
- absDelay
 - veriloga.py, 141
- acos
 - veriloga.py, 141
- acosh
 - veriloga.py, 141
- acStim
 - veriloga.py, 142
- addNode
 - vagen.veriloga.Module, 96
- analog
 - vagen.veriloga.Module, 97
- analysis
 - veriloga.py, 142
- anaTypes
 - veriloga.py, 161
- append
 - vagen.hilevelmod.Bus, 26
 - vagen.veriloga.CaseClass, 28
 - vagen.veriloga.CmdList, 34
 - vagen.veriloga.Cond, 36
- applyI
 - vagen.hilevelmod.Idc, 72
 - vagen.hilevelmod.IdcBus, 74
 - vagen.hilevelmod.Smu, 114
 - vagen.hilevelmod.SmuBus, 117
- applyR
 - vagen.hilevelmod.Smu, 114
 - vagen.hilevelmod.SmuBus, 117
- applyV
 - vagen.hilevelmod.Smu, 115
 - vagen.hilevelmod.SmuBus, 117
 - vagen.hilevelmod.Vdc, 122
 - vagen.hilevelmod.VdcBus, 124
- asin
 - veriloga.py, 142
- asinh
 - veriloga.py, 143
- At
 - veriloga.py, 143
- atan
 - veriloga.py, 143
- atan2
 - veriloga.py, 143
- atanh
 - veriloga.py, 144
- beginningAnalog
 - vagen.veriloga.Module, 97
- block
 - veriloga.py, 144
- blockComment
 - veriloga.py, 144
- BoundStep
 - veriloga.py, 144
- Case
 - veriloga.py, 145
- ceil
 - veriloga.py, 145
- checkBool
 - veriloga.py, 145
- checkInstance
 - veriloga.py, 146
- checkInteger

- veriloga.py, 146
- checkNotInstance
 - veriloga.py, 146
- checkNumber
 - veriloga.py, 147
- checkReal
 - veriloga.py, 147
- checkType
 - veriloga.py, 147
- clock
 - vagen.hilevelmod.HiLevelMod, 65
- cos
 - veriloga.py, 147
- cosh
 - veriloga.py, 148
- ddt
 - veriloga.py, 148
- dec
 - vagen.veriloga.IntegerVar, 89
- dig
 - vagen.hilevelmod.HiLevelMod, 66
- Discontinuity
 - veriloga.py, 148
- electrical
 - vagen.veriloga.Module, 97
- Else
 - vagen.veriloga.Cond, 36
- endAnalog
 - vagen.veriloga.Module, 97
- eq
 - vagen.veriloga.BoolVar, 23
 - vagen.veriloga.IntegerVar, 89
 - vagen.veriloga.RealVar, 109
- exp
 - veriloga.py, 149
- expDistInt
 - veriloga.py, 149
- expDistReal
 - veriloga.py, 149
- Fclose
 - veriloga.py, 149
- Finish
 - veriloga.py, 150
- fixName
 - vagen.veriloga.Module, 98
- flat
 - vagen.veriloga.CmdList, 34
- floor
 - veriloga.py, 150
- Fopen
 - veriloga.py, 150
- For
 - veriloga.py, 150
- Fstrobe
 - veriloga.py, 151
- Fwrite
 - veriloga.py, 151
- gaussDistInt
 - veriloga.py, 151
- gaussDistReal
 - veriloga.py, 152
- getBlock
 - vagen.veriloga.Cond, 37
- getBlockList
 - vagen.veriloga.CaseClass, 28
- getCmd
 - vagen.hilevelmod.Mark, 91
- getCond
 - vagen.veriloga.Cond, 37
 - vagen.veriloga.ForLoop, 62
 - vagen.veriloga.WhileLoop, 134
- getDelay
 - vagen.hilevelmod.WaitUs, 131
- getEqs
 - vagen.hilevelmod.HiLevelMod, 66
 - vagen.hilevelmod.Marker, 93
- getEvt
 - vagen.hilevelmod.WaitSignal, 129
- getHeader
 - vagen.veriloga.Block, 16
- getInc
 - vagen.veriloga.ForLoop, 62
- getModuleName
 - vagen.veriloga.Module, 98
- getN
 - vagen.veriloga.RepeatLoop, 112
- getName
 - vagen.hilevelmod.Marker, 93
 - vagen.veriloga.Electrical, 55
- getOcn
 - vagen.hilevelmod.HiLevelMod, 66
- getStart
 - vagen.veriloga.ForLoop, 63
- getVA
 - vagen.hilevelmod.Mark, 92
 - vagen.hilevelmod.WaitSignal, 129
 - vagen.hilevelmod.WaitUs, 131
 - vagen.veriloga.Block, 16
 - vagen.veriloga.CaseClass, 29
 - vagen.veriloga.Cmd, 32
 - vagen.veriloga.CmdList, 34
 - vagen.veriloga.Cond, 37
 - vagen.veriloga.Module, 98
- getValue
 - vagen.veriloga.Bool, 21
 - vagen.veriloga.Integer, 87
 - vagen.veriloga.Real, 108
- high
 - vagen.hilevelmod.Marker, 94
- hilevelmod.py, 135
- hiZ
 - vagen.hilevelmod.DigBusInOut, 43
 - vagen.hilevelmod.DigInOut, 50

- hypot
 - veriloga.py, [152](#)
- iAttr
 - vagen.veriloga.Electrical, [55](#)
- iCont
 - vagen.veriloga.Electrical, [56](#)
- idc
 - vagen.hilevelmod.HiLevelMod, [67](#)
- idt
 - veriloga.py, [152](#)
- If
 - veriloga.py, [153](#)
- ilnd
 - vagen.veriloga.Electrical, [56](#)
- inc
 - vagen.veriloga.IntegerVar, [90](#)
- lastCrossing
 - veriloga.py, [153](#)
- limexp
 - veriloga.py, [153](#)
- In
 - veriloga.py, [153](#)
- log
 - veriloga.py, [154](#)
- low
 - vagen.hilevelmod.Marker, [94](#)
- lowZ
 - vagen.hilevelmod.DigBusInOut, [43](#)
 - vagen.hilevelmod.DigInOut, [51](#)
- mark
 - vagen.hilevelmod.Marker, [94](#)
- marker
 - vagen.hilevelmod.HiLevelMod, [67](#)
- off
 - vagen.hilevelmod.Clock, [30](#)
- on
 - vagen.hilevelmod.Clock, [30](#)
- par
 - vagen.veriloga.Module, [98](#)
- parseBool
 - veriloga.py, [154](#)
- parseInteger
 - veriloga.py, [154](#)
- parseNumber
 - veriloga.py, [155](#)
- parseReal
 - veriloga.py, [155](#)
- poissonDistInt
 - veriloga.py, [155](#)
- poissonDistReal
 - veriloga.py, [155](#)
- random
 - veriloga.py, [156](#)
- read
 - vagen.hilevelmod.DigBusIn, [40](#)
 - vagen.hilevelmod.DigIn, [48](#)
- Repeat
 - veriloga.py, [156](#)
- seq
 - vagen.hilevelmod.HiLevelMod, [68](#)
- seqNested
 - vagen.hilevelmod.HiLevelMod, [68](#)
- setCond
 - vagen.hilevelmod.Sw, [119](#)
- setDelay
 - vagen.hilevelmod.DigBusOut, [45](#)
 - vagen.hilevelmod.DigOut, [53](#)
- setRiseFall
 - vagen.hilevelmod.DigBusOut, [45](#)
 - vagen.hilevelmod.DigOut, [53](#)
 - vagen.hilevelmod.Idc, [72](#)
 - vagen.hilevelmod.IdcBus, [74](#)
 - vagen.hilevelmod.Sw, [119](#)
 - vagen.hilevelmod.Vdc, [122](#)
 - vagen.hilevelmod.VdcBus, [125](#)
- sin
 - veriloga.py, [156](#)
- sinh
 - veriloga.py, [156](#)
- slew
 - veriloga.py, [157](#)
- smu
 - vagen.hilevelmod.HiLevelMod, [68](#)
- sqrt
 - veriloga.py, [157](#)
- Strobe
 - veriloga.py, [157](#)
- sw
 - vagen.hilevelmod.HiLevelMod, [69](#)
- tan
 - veriloga.py, [158](#)
- tanh
 - veriloga.py, [158](#)
- temp
 - veriloga.py, [161](#)
- ternary
 - veriloga.py, [158](#)
- toggle
 - vagen.veriloga.BoolVar, [23](#)
- transition
 - veriloga.py, [158](#)
- uDistInt
 - veriloga.py, [159](#)
- uDistReal
 - veriloga.py, [159](#)
- unfoldParams
 - veriloga.py, [160](#)
- unfoldSimTypes
 - veriloga.py, [160](#)

- vagen, 11
- vagen.hilevelmod.Bus, 25
 - __getitem__, 26
 - __init__, 26
 - append, 26
- vagen.hilevelmod.Clock, 29
 - __init__, 29
 - off, 30
 - on, 30
- vagen.hilevelmod.DigBusIn, 39
 - __init__, 40
 - read, 40
- vagen.hilevelmod.DigBusInOut, 41
 - __init__, 42
 - hiZ, 43
 - lowZ, 43
- vagen.hilevelmod.DigBusOut, 43
 - __init__, 45
 - setDelay, 45
 - setRiseFall, 45
 - write, 46
- vagen.hilevelmod.DigIn, 46
 - __init__, 47
 - read, 48
- vagen.hilevelmod.DigInOut, 48
 - __init__, 50
 - hiZ, 50
 - lowZ, 51
- vagen.hilevelmod.DigOut, 51
 - __init__, 52
 - setDelay, 53
 - setRiseFall, 53
 - write, 54
- vagen.hilevelmod.HiLevelMod, 63
 - __init__, 65
 - clock, 65
 - dig, 66
 - getEqs, 66
 - getOcn, 66
 - idc, 67
 - marker, 67
 - seq, 68
 - seqNested, 68
 - smu, 68
 - sw, 69
 - var, 69
 - vdc, 70
- vagen.hilevelmod.Idc, 70
 - __init__, 71
 - applyI, 72
 - setRiseFall, 72
- vagen.hilevelmod.IdcBus, 72
 - __init__, 74
 - applyI, 74
 - setRiseFall, 74
- vagen.hilevelmod.Mark, 90
 - __init__, 91
 - __str__, 91
 - getCmd, 91
 - getVA, 92
- vagen.hilevelmod.Marker, 92
 - __init__, 93
 - getEqs, 93
 - getName, 93
 - high, 94
 - low, 94
 - mark, 94
- vagen.hilevelmod.Smu, 112
 - __init__, 113
 - applyI, 114
 - applyR, 114
 - applyV, 115
- vagen.hilevelmod.SmuBus, 115
 - __init__, 116
 - applyI, 117
 - applyR, 117
 - applyV, 117
- vagen.hilevelmod.Sw, 118
 - __init__, 118
 - setCond, 119
 - setRiseFall, 119
- vagen.hilevelmod.Vdc, 121
 - __init__, 122
 - applyV, 122
 - setRiseFall, 122
- vagen.hilevelmod.VdcBus, 123
 - __init__, 124
 - applyV, 124
 - setRiseFall, 125
- vagen.hilevelmod.WaitSignal, 127
 - __init__, 128
 - __str__, 128
 - getEvt, 129
 - getVA, 129
- vagen.hilevelmod.WaitUs, 129
 - __init__, 130
 - __str__, 131
 - getDelay, 131
 - getVA, 131
- vagen.veriloga.Above, 13
 - __init__, 14
- vagen.veriloga.Block, 14
 - __init__, 15
 - getHeader, 16
 - getVA, 16
- vagen.veriloga.Bool, 16
 - __and__, 18
 - __eq__, 18
 - __init__, 17
 - __invert__, 18
 - __ne__, 18
 - __or__, 20
 - __rand__, 20
 - __ror__, 20
 - __rxor__, 21
 - __str__, 21

- __xor__, 21
- getValue, 21
- vagen.veriloga.BoolVar, 22
 - __init__, 23
 - eq, 23
 - toggle, 23
- vagen.veriloga.Branch, 24
 - __init__, 24
- vagen.veriloga.CaseClass, 27
 - __init__, 28
 - append, 28
 - getBlockList, 28
 - getVA, 29
- vagen.veriloga.Cmd, 30
 - __init__, 31
 - __str__, 31
 - getVA, 32
- vagen.veriloga.CmdList, 32
 - __init__, 33
 - __str__, 33
 - append, 34
 - flat, 34
 - getVA, 34
- vagen.veriloga.Cond, 35
 - __init__, 36
 - append, 36
 - Else, 36
 - getBlock, 37
 - getCond, 37
 - getVA, 37
- vagen.veriloga.Cross, 38
 - __init__, 38
- vagen.veriloga.Electrical, 54
 - __init__, 55
 - getName, 55
 - iAttr, 55
 - iCont, 56
 - iInd, 56
 - vAttr, 56
 - vCont, 57
 - vInd, 57
- vagen.veriloga.Event, 57
 - __init__, 58
 - __or__, 59
 - __str__, 59
- vagen.veriloga.FinalStep, 59
 - __init__, 60
- vagen.veriloga.ForLoop, 60
 - __init__, 62
 - getCond, 62
 - getInc, 62
 - getStart, 63
- vagen.veriloga.InitialStep, 75
 - __init__, 75
- vagen.veriloga.Integer, 76
 - __abs__, 78
 - __add__, 78
 - __and__, 79
 - __eq__, 79
 - __ge__, 79
 - __gt__, 79
 - __init__, 78
 - __invert__, 80
 - __le__, 80
 - __lshift__, 80
 - __lt__, 81
 - __mod__, 81
 - __mul__, 81
 - __ne__, 81
 - __neg__, 82
 - __or__, 82
 - __pos__, 82
 - __pow__, 83
 - __radd__, 83
 - __rand__, 83
 - __rlshift__, 83
 - __rmod__, 84
 - __rmul__, 84
 - __ror__, 84
 - __rpow__, 85
 - __rrshift__, 85
 - __rshift__, 85
 - __rsub__, 85
 - __rtruediv__, 86
 - __rxor__, 86
 - __str__, 86
 - __sub__, 87
 - __truediv__, 87
 - __xor__, 87
 - getValue, 87
- vagen.veriloga.IntegerVar, 88
 - __init__, 89
 - dec, 89
 - eq, 89
 - inc, 90
- vagen.veriloga.Module, 95
 - __init__, 96
 - addNode, 96
 - analog, 97
 - beginningAnalog, 97
 - electrical, 97
 - endAnalog, 97
 - fixName, 98
 - getModuleName, 98
 - getVA, 98
 - par, 98
 - var, 99
- vagen.veriloga.Real, 99
 - __abs__, 101
 - __add__, 101
 - __eq__, 102
 - __ge__, 102
 - __gt__, 102
 - __init__, 101
 - __le__, 102
 - __lt__, 103

- `__mul__`, [103](#)
 - `__ne__`, [103](#)
 - `__neg__`, [104](#)
 - `__pos__`, [104](#)
 - `__pow__`, [104](#)
 - `__radd__`, [104](#)
 - `__rmul__`, [105](#)
 - `__rpow__`, [105](#)
 - `__rsub__`, [105](#)
 - `__rtruediv__`, [106](#)
 - `__str__`, [106](#)
 - `__sub__`, [106](#)
 - `__truediv__`, [106](#)
 - `getValue`, [108](#)
- `vagen.veriloga.RealVar`, [108](#)
 - `__init__`, [109](#)
 - `eq`, [109](#)
- `vagen.veriloga.RepeatLoop`, [110](#)
 - `__init__`, [111](#)
 - `getN`, [112](#)
- `vagen.veriloga.Timer`, [119](#)
 - `__init__`, [120](#)
- `vagen.veriloga.WaitAnalogEvent`, [125](#)
 - `__init__`, [127](#)
- `vagen.veriloga.WhileLoop`, [132](#)
 - `__init__`, [133](#)
 - `getCond`, [134](#)
- `var`
 - `vagen.hilevelmod.HiLevelMod`, [69](#)
 - `vagen.veriloga.Module`, [99](#)
- `vAttr`
 - `vagen.veriloga.Electrical`, [56](#)
- `vCont`
 - `vagen.veriloga.Electrical`, [57](#)
- `vdc`
 - `vagen.hilevelmod.HiLevelMod`, [70](#)
- `veriloga.py`, [136](#)
 - `absDelay`, [141](#)
 - `acos`, [141](#)
 - `acosh`, [141](#)
 - `acStim`, [142](#)
 - `analysis`, [142](#)
 - `anaTypes`, [161](#)
 - `asin`, [142](#)
 - `asinh`, [143](#)
 - `At`, [143](#)
 - `atan`, [143](#)
 - `atan2`, [143](#)
 - `atanh`, [144](#)
 - `block`, [144](#)
 - `blockComment`, [144](#)
 - `BoundStep`, [144](#)
 - `Case`, [145](#)
 - `ceil`, [145](#)
 - `checkBool`, [145](#)
 - `checkInstance`, [146](#)
 - `checkInteger`, [146](#)
 - `checkNotInstance`, [146](#)
 - `checkNumber`, [147](#)
 - `checkReal`, [147](#)
 - `checkType`, [147](#)
 - `cos`, [147](#)
 - `cosh`, [148](#)
 - `ddt`, [148](#)
 - `Discontinuity`, [148](#)
 - `exp`, [149](#)
 - `expDistInt`, [149](#)
 - `expDistReal`, [149](#)
 - `Fclose`, [149](#)
 - `Finish`, [150](#)
 - `floor`, [150](#)
 - `Fopen`, [150](#)
 - `For`, [150](#)
 - `Fstroke`, [151](#)
 - `Fwrite`, [151](#)
 - `gaussDistInt`, [151](#)
 - `gaussDistReal`, [152](#)
 - `hypot`, [152](#)
 - `idt`, [152](#)
 - `If`, [153](#)
 - `lastCrossing`, [153](#)
 - `limexp`, [153](#)
 - `In`, [153](#)
 - `log`, [154](#)
 - `parseBool`, [154](#)
 - `parseInteger`, [154](#)
 - `parseNumber`, [155](#)
 - `parseReal`, [155](#)
 - `poissonDistInt`, [155](#)
 - `poissonDistReal`, [155](#)
 - `random`, [156](#)
 - `Repeat`, [156](#)
 - `sin`, [156](#)
 - `sinh`, [156](#)
 - `slew`, [157](#)
 - `sqrt`, [157](#)
 - `Strobe`, [157](#)
 - `tan`, [158](#)
 - `tanh`, [158](#)
 - `temp`, [161](#)
 - `ternary`, [158](#)
 - `transition`, [158](#)
 - `uDistInt`, [159](#)
 - `uDistReal`, [159](#)
 - `unfoldParams`, [160](#)
 - `unfoldSimTypes`, [160](#)
 - `While`, [160](#)
 - `Write`, [160](#)
- `vInd`
 - `vagen.veriloga.Electrical`, [57](#)
- `While`
 - `veriloga.py`, [160](#)
- `Write`
 - `veriloga.py`, [160](#)
- `write`
 - `vagen.hilevelmod.DigBusOut`, [46](#)

vagen.hilevelmod.DigOut, [54](#)