## Report on Identified Refactoring Opportunities for (SC/ST) Refactoring

The tool identifies the Control-Fields and conditional constructs (Switch/If) that use these Control-Fields to simulate the (SC / ST) refactoring. It prioritizes the Control-Fields (refactoring opportunities) based on the following criteria:

- 1. Number of conditional constructs that switch on Control-Field Group i represents i different conditional constructs where the Control-Field is used
- 2. Average size of the conditional body
- 3. Number of control values
  - 2-3 control values
  - 3-6 control values
  - 6-n control values
- 4. Presence of conditional constructs with respect to the class of declaration (COD) of Control-Field

In COD (A)

Outside (B)

Mixed (C)

- 5. Qualified for SC or ST
- 6. Static Field
- 7. Have subclasses already

## **Input Benchmark Statistics:**

No. Of Classes: 1731

No. Of Primitive Felds: 1888

No. Of Control-Fields for Subclass Pattern: 20 No. Of Control-Fields for State pattern: 92

Total No. Of Control-Fields: 112

Uses	Replace Type Code with Subclass (SC)		Uses	Replace Type Code with State (ST)	
24	<cck.util.option.bool, value=""></cck.util.option.bool,>	(B)	66	<pre><jintgen.isdl.parser.isdlparser,jj_ntk></jintgen.isdl.parser.isdlparser,jj_ntk></pre>	(A)
8	<avrora.sim.output.eventgen, enabled=""></avrora.sim.output.eventgen,>	(A)	40	<avrora.syntax.atmel.atmelparser,jj_nt< td=""><td>(A)</td></avrora.syntax.atmel.atmelparser,jj_nt<>	(A)
5	<avrora.monitors.interruptmonitor.mon< td=""><td>(A)</td><td></td><td>k&gt;</td><td></td></avrora.monitors.interruptmonitor.mon<>	(A)		k>	
4	, show> <avrora.monitors.interruptmonitor.mon< td=""><td>(A)</td><td>35</td><td><pre><jintgen.isdl.parser.isdlparsertokenm< td=""><td>(A)</td></jintgen.isdl.parser.isdlparsertokenm<></pre></td></avrora.monitors.interruptmonitor.mon<>	(A)	35	<pre><jintgen.isdl.parser.isdlparsertokenm< td=""><td>(A)</td></jintgen.isdl.parser.isdlparsertokenm<></pre>	(A)
	, invokeOnly>		30	<cck.text.verbose.printer, enabled=""></cck.text.verbose.printer,>	(C)
3	<avrora.sim.mcu.defaultmcu.pin, outputDir&gt;</avrora.sim.mcu.defaultmcu.pin, 	(A)	23	<avrora.syntax.objdump.objdumpparse r,jj_ntk=""></avrora.syntax.objdump.objdumpparse>	(A)
	<avrora.monitors.packetmonitor.mon, showPackets&gt;</avrora.monitors.packetmonitor.mon, 	(A)	19	<avrora.syntax.objdump.objdumpparse rTokenManager,curChar&gt;</avrora.syntax.objdump.objdumpparse 	(A)
	<avrora.sim.radio.cc1000radio.serial ConfigurationInterface, writeCommand&gt;</avrora.sim.radio.cc1000radio.serial 	(A)		<avrora.syntax.atmel.atmelparsertoke nManager,curChar&gt;</avrora.syntax.atmel.atmelparsertoke 	(A)
			17	<avrora.syntax.atmel.atmelparser,< td=""><td>(A)</td></avrora.syntax.atmel.atmelparser,<>	(A)
	<avrora.monitors.sniffermonitor.mon, showTransmitted&gt;</avrora.monitors.sniffermonitor.mon, 	(A)		jj_la>	
	<avrora.monitors.sniffermonitor.mon,< td=""><td>(A)</td><td>12</td><td><avrora.test.probes.probeparsertokenm anager,curChar&gt;</avrora.test.probes.probeparsertokenm </td><td>(A)</td></avrora.monitors.sniffermonitor.mon,<>	(A)	12	<avrora.test.probes.probeparsertokenm anager,curChar&gt;</avrora.test.probes.probeparsertokenm 	(A)
	showReceived> <avrora.monitors.packetmonitor.mon,< td=""><td>(A)</td><td>11</td><td><avrora.stack.statecache.set, delegating&gt;</avrora.stack.statecache.set, </td><td>(A)</td></avrora.monitors.packetmonitor.mon,<>	(A)	11	<avrora.stack.statecache.set, delegating&gt;</avrora.stack.statecache.set, 	(A)
2	cc2420radio> <avrora.monitors.sniffermonitor.mon,< td=""><td>(A)</td><td>9</td><td><avrora.syntax.objdump.objdumpparse rTokenManager,jjmatchedPos&gt;</avrora.syntax.objdump.objdumpparse </td><td>(A)</td></avrora.monitors.sniffermonitor.mon,<>	(A)	9	<avrora.syntax.objdump.objdumpparse rTokenManager,jjmatchedPos&gt;</avrora.syntax.objdump.objdumpparse 	(A)
	Print>		8	<avrora.sim.simulation, running=""></avrora.sim.simulation,>	(C)
	<avrora.syntax.module.seg, acceptsData&gt;</avrora.syntax.module.seg, 	(A)	7	<cck.text.status, enabled=""></cck.text.status,>	(A)
	<pre><avrora.sim.clock.barriersynchronizer. removed="" synchevent,=""></avrora.sim.clock.barriersynchronizer.></pre>	(A)	6	<avrora.sim.radio.cc1000radio.mainr corepd="" egister,=""></avrora.sim.radio.cc1000radio.mainr>	(C)
	<pre><cck.elf.elfdatainputstream,< pre=""></cck.elf.elfdatainputstream,<></pre>	(A)		<avrora.sim.atmelinterpreter, c=""></avrora.sim.atmelinterpreter,>	(B)
	bigEndian>	A		<avrora.sim.util.mem16, state=""></avrora.sim.util.mem16,>	
	V	(C)		<avrora.test.probes.probeparser,jj_ntk></avrora.test.probes.probeparser,jj_ntk>	(A)
	or, on> <avrora.syntax.syntacticoperand.expr,< td=""><td>(A)</td><td>5</td><td><avrora.sim.radio.cc1000radio.mainr biaspd="" egister,=""></avrora.sim.radio.cc1000radio.mainr></td><td>(C)</td></avrora.syntax.syntacticoperand.expr,<>	(A)	5	<avrora.sim.radio.cc1000radio.mainr biaspd="" egister,=""></avrora.sim.radio.cc1000radio.mainr>	(C)
	simplified>	(11)		<avrora.arch.avr.avrstate, c=""></avrora.arch.avr.avrstate,>	(B)
	<avrora.monitors.calltimemonitor.call< td=""><td>(A)</td><td></td><td><avrora.sim.atmelinterpreter, nextpc=""></avrora.sim.atmelinterpreter,></td><td>(B)</td></avrora.monitors.calltimemonitor.call<>	(A)		<avrora.sim.atmelinterpreter, nextpc=""></avrora.sim.atmelinterpreter,>	(B)
	TimeMon, ignore_interrupts> <avrora.sim.util.memprint, log=""></avrora.sim.util.memprint,>	(A)		<avrora.sim.radio.medium.txrx, activated=""></avrora.sim.radio.medium.txrx,>	(B)
	<avrora.stack.statetransitiongraph.edg e, type&gt;</avrora.stack.statetransitiongraph.edg 	(B)		<avrora.sim.mcu.registerset.field,valu e=""></avrora.sim.mcu.registerset.field,valu>	(B)
	<jintgen.isdl.instrdecl, pseudo=""></jintgen.isdl.instrdecl,>	(B)		<avrora.syntax.atmel.atmelparsertoke nManager,jjmatchedPos&gt;</avrora.syntax.atmel.atmelparsertoke 	(A)

Uses	Replace Type Code with State (ST)	
4	<avrora.sim.radio.cc1000radio.mainr egister,="" rxtx=""></avrora.sim.radio.cc1000radio.mainr>	(C)
	<avrora.sim.radio.cc1000radio.mainr egister, fsPd&gt;</avrora.sim.radio.cc1000radio.mainr 	(C)
	<avrora.arch.msp430.msp430state, z=""></avrora.arch.msp430.msp430state,>	(B)
	<pre><jintgen.isdl.parser.isdlparsertokenm< td=""><td>(A)</td></jintgen.isdl.parser.isdlparsertokenm<></pre>	(A)
	<avrora.sim.atmelinterpreter, sleeping=""></avrora.sim.atmelinterpreter,>	
	<avrora.arch.msp430.msp430state, c=""></avrora.arch.msp430.msp430state,>	
	<avrora.sim.atmelinterpreter, n=""></avrora.sim.atmelinterpreter,>	
	<avrora.sim.atmelinterpreter, s=""></avrora.sim.atmelinterpreter,>	
	<avrora.sim.atmelinterpreter, h=""></avrora.sim.atmelinterpreter,>	
	<avrora.sim.atmelinterpreter, i=""></avrora.sim.atmelinterpreter,>	
	<avrora.sim.atmelinterpreter, t=""></avrora.sim.atmelinterpreter,>	
	<avrora.sim.atmelinterpreter, v=""></avrora.sim.atmelinterpreter,>	
	<avrora.sim.atmelinterpreter, z=""></avrora.sim.atmelinterpreter,>	
	<avrora.sim.radio.medium.receiver, locked=""></avrora.sim.radio.medium.receiver,>	
	<avrora.arch.msp430.msp430operand, op_type&gt;</avrora.arch.msp430.msp430operand, 	(B)
3	<avrora.sim.radio.cc1000radio.mainr egister,="" txpd=""></avrora.sim.radio.cc1000radio.mainr>	(C)
	<avrora.stack.statecache.state, isexplored=""></avrora.stack.statecache.state,>	(B)
	<avrora.arch.avr.avrstate, v=""></avrora.arch.avr.avrstate,>	
	<avrora.arch.avr.avrstate, s=""></avrora.arch.avr.avrstate,>	
	<avrora.arch.avr.avrstate, t=""></avrora.arch.avr.avrstate,>	
	<avrora.arch.avr.avrstate, z=""></avrora.arch.avr.avrstate,>	(B)
	<avrora.stack.statecache.set, empty=""></avrora.stack.statecache.set,>	(A)
	<avrora.arch.avr.avrstate, h=""></avrora.arch.avr.avrstate,>	
	<avrora.arch.avr.avrstate, n=""></avrora.arch.avr.avrstate,>	
	<avrora.arch.avr.avrstate, i=""></avrora.arch.avr.avrstate,>	(B)
	<avrora.sim.util.mem16, count=""></avrora.sim.util.mem16,>	(A)
	<avrora.sim.util.mem8, count=""></avrora.sim.util.mem8,>	(A)
	<avrora.sim.mcu.eeprom, writeEnable&gt;</avrora.sim.mcu.eeprom, 	(B)
	<avrora.sim.finitestatemachine, curState&gt;</avrora.sim.finitestatemachine, 	(C)
	<avrora.stack.analyzer, trace=""></avrora.stack.analyzer,>	(A)
	<pre><jintgen.isdl.parser.isdlparsertokenm< td=""><td>(A)</td></jintgen.isdl.parser.isdlparsertokenm<></pre>	(A)
	<avrora.sim.platform.externalflash,dfo pcode=""></avrora.sim.platform.externalflash,dfo>	(B)
	<avrora.arch.avr.avroperand,op_type< td=""><td>(B)</td></avrora.arch.avr.avroperand,op_type<>	(B)
	<avrora.sim.radio.cc2420radio.receiv er,state=""></avrora.sim.radio.cc2420radio.receiv>	(A)

Cavrora.sim.mcu.ATMegaTimer, countUp>   Cavrora.syntax.Module, caseSensitivity>   Cavrora.monitors.PacketMonitor.Mon, bufferPos>   Cavrora.sim.radio.CC1000Radio.MainR egister, rxPd>   Cavrora.sim.mcu.Timer8Bit, period>   Cavrora.sim.mcu.Timer8Bit, period>   Cavrora.sim.mcu.Timer8Bit, period>   Cavrora.sim.radio.CC2420Radio, configRAMBank>   Cavrora.sim.radio.CC2420Radio, configRAMBank>   Cavrora.stack.isea.ISEAbstractState.Ele ment, read>   Cavrora.stack.isea.ISEAbstractState.Ele ment, read>   Cavrora.sim.util.MemTimer, timer state>   Cavrora.sim.util.MemTimer, timer state>   Cavrora.sim.state.StateCache.State, onFrontier>   Cack.parser.AbstractParseException, specialConstructor>   Cavrora.sim.mcu.SPI.SPCRReg, prev_spie>   Cack.text.Terminal, htmlColors>   Cavrora.sim.mcu.SPI.SPCRReg, prev_spie>   Cack.util.Util.Error, STACKTRACES>   Cavrora.sim.energy.EnergyControl, active>   Cavrora.sim.radio.CC1000Radio.SPITicker, activated>   Cavrora.sim.mcu.ATMegaTimer, timerEnabled>   Cavrora.sim.mcu.ADC.ControlRegister, converting>   Cavrora.sim.mcu.ADC.ControlRegister, converting>   Cavrora.sim.mcu.ADC.ControlRegister, converting>   Cavrora.sim.mcu.SPI.TransferEvent, transmitting>   Cavrora.gui.GraphEvents.MyVector, current>   Cavrora.sim.platform.ExternalFlash, isReading>   Cavrora.sim.platform.ExternalFlash, isReading>   Cavrora.sin.atal.Atmcl.A	Uses	Replace Type Code with State (ST)	
cavrora.syntax.Module, caseSensitivity>   cavrora.monitors.PacketMonitor.Mon, bufferPos>   cavrora.sim.radio.CC1000Radio.MainR egister, rxPd>   cavrora.sim.mcu.Timer8Bit, period>   cock.text.Printer, first>   (A)     cavrora.sim.radio.CC2420Radio, configRAMBank>   cavrora.sim.seal.SEAbstractState.Ele ment, read>   cavrora.siad.OperandTypeDecl.Access or, polymorphic>   cavrora.sim.util.MemTimer, timer_state>   cavrora.sim.util.MemTimer, timer_state>   cavrora.stack.StateCache.State, onFrontier>   ccck.parser.AbstractParseException, specialConstructor>   cavrora.m.mcu.SPLSPCRReg, prev_spie>   ccck.text.Terminal, htmlColors>   cavrora.monitors.TraceMonitor.Mon, nesting>   ccck.util.Util.Error, STACKTRACES>   (C)   cavrora.sim.radio.CC1000Radio.SPITic ker, activated>   cavrora.sim.mcu.APLspiner, timerEnabled>   cavrora.sim.mcu.SPI.TransferEvent, transmitting>   cavrora.sim.mcu.SPI.TransferEvent, transmitting>   cavrora.sim.mcu.SPI.TransferEvent, transmitting>   cavrora.sim.mcu.SPI.TransferEvent, transmitting>   cavrora.sim.mcu.SPI.TransferEven	_	<avrora.sim.mcu.atmegatimer,< td=""><td></td></avrora.sim.mcu.atmegatimer,<>	
<pre><avrora.monitors.packetmonitor.mon, bufferpos=""> <avrora.sim.radio.cc1000radio.mainr egister,="" rxpd=""> <avrora.sim.mcu.timer8bit, period=""></avrora.sim.mcu.timer8bit,></avrora.sim.radio.cc1000radio.mainr></avrora.monitors.packetmonitor.mon,></pre>		<avrora.syntax.module,< td=""><td>(A)</td></avrora.syntax.module,<>	(A)
<avrora.sim.radio.cc1000radio.mainr egister,="" rxpd=""> <avrora.sim.mcu.timer8bit, period=""> <avrora.sim.mcu.timer8bit, period=""> <avrora.sim.radio.cc2420radio, configrambank=""> <jintgen.gen.disassembler.decoder, chained=""> <avrora.stack.isea.iseabstractstate.ele ment,="" read=""> <jintgen.isdl.operandtypedecl.access or,="" polymorphic=""> <avrora.sim.util.memtimer, timer_state=""> <avrora.stack.statecache.state, onfrontier=""> <cck.parser.abstractparseexception, specialconstructor=""> <avrora.sim.mcu.spi.spcrreg, prev_spie=""> <cck.text.terminal, htmlcolors=""> <avrora.sim.energy.energycontrol, active=""> <jintgen.gen.disassembler.decoder, multiple=""> <avrora.sim.radio.cc1000radio.spitic activated="" ker,=""> <avrora.sim.radio.cc1000radio.spitic activated="" ker,=""> <avrora.sim.mcu.atmegatimer, timerenabled=""> <avrora.sim.mcu.adc.controlregister, converting=""> <avrora.sim.mcu.adc.controlregister, converting=""> <avrora.sim.mcu.spi.transferevent, transmitting=""> <avrora.sim.mcu.spi.transferevent, transmitting=""> <avrora.gui.graphevents.myvector, current=""> <avrora.gui.graphevents.myvector, current=""> <avrora.sim.platform.externalflash, isreading=""> <avrora.syntax.atmel.atmelparsertoke (a)="" <avrora.syntax.atmel.atme<="" <avrora.syntax.atmel.atmelparsertoke="" td=""><td></td><td><avrora.monitors.packetmonitor.mon,< td=""><td>(A)</td></avrora.monitors.packetmonitor.mon,<></td></avrora.syntax.atmel.atmelparsertoke></avrora.sim.platform.externalflash,></avrora.gui.graphevents.myvector,></avrora.gui.graphevents.myvector,></avrora.sim.mcu.spi.transferevent,></avrora.sim.mcu.spi.transferevent,></avrora.sim.mcu.adc.controlregister,></avrora.sim.mcu.adc.controlregister,></avrora.sim.mcu.atmegatimer,></avrora.sim.radio.cc1000radio.spitic></avrora.sim.radio.cc1000radio.spitic></jintgen.gen.disassembler.decoder,></avrora.sim.energy.energycontrol,></cck.text.terminal,></avrora.sim.mcu.spi.spcrreg,></cck.parser.abstractparseexception,></avrora.stack.statecache.state,></avrora.sim.util.memtimer,></jintgen.isdl.operandtypedecl.access></avrora.stack.isea.iseabstractstate.ele></jintgen.gen.disassembler.decoder,></avrora.sim.radio.cc2420radio,></avrora.sim.mcu.timer8bit,></avrora.sim.mcu.timer8bit,></avrora.sim.radio.cc1000radio.mainr>		<avrora.monitors.packetmonitor.mon,< td=""><td>(A)</td></avrora.monitors.packetmonitor.mon,<>	(A)
<pre><avrora.sim.mcu.timer8bit, period=""></avrora.sim.mcu.timer8bit,></pre>		<avrora.sim.radio.cc1000radio.mainr< td=""><td>(A)</td></avrora.sim.radio.cc1000radio.mainr<>	(A)
<pre><avrora.sim.radio.cc2420radio,< td=""><td></td><td>-</td><td>(B)</td></avrora.sim.radio.cc2420radio,<></pre>		-	(B)
configRAMBank> <jintgen.gen.disassembler.decoder, chained=""> <avrora.stack.isea.iseabstractstate.ele ment,="" read=""> <jintgen.isdl.operandtypedecl.access or,="" polymorphic=""> <avrora.sim.util.memtimer, timer_state=""> <avrora.stack.statecache.state, onfrontier=""> <cck.parser.abstractparseexception, specialconstructor=""> <avrora.sim.mcu.spi.spcrreg, prev_spie=""> <cck.text.terminal, htmlcolors=""> <avrora.monitors.tracemonitor.mon, nesting=""> <cck.util.util.error, stacktraces=""> <avrora.sim.energy.energycontrol, active=""> <jintgen.gen.disassembler.decoder, multiple=""> <avrora.sim.radio.cc1000radio.spitic activated="" ker,=""> <cck.text.printer, begline=""> <avrora.sim.mcu.atmegatimer, timerenabled=""> <avrora.sim.mcu.adc.controlregister, converting=""> <avrora.sim.mcu.adc.controlregister, converting=""> <avrora.sim.mcu.spi.transferevent, transmitting=""> <cck.text.status, timing=""> <avrora.gui.graphevents.myvector, current=""> <cck.text.testengine, verbose=""> <avrora.sim.platform.externalflash, isreading=""> <avrora.syntax.atmel.atmelparsertoke (a)="" (a)<="" <avrora.syntax.atmel.atmelparsertoke="" td=""><td></td><td></td><td>(A)</td></avrora.syntax.atmel.atmelparsertoke></avrora.sim.platform.externalflash,></cck.text.testengine,></avrora.gui.graphevents.myvector,></cck.text.status,></avrora.sim.mcu.spi.transferevent,></avrora.sim.mcu.adc.controlregister,></avrora.sim.mcu.adc.controlregister,></avrora.sim.mcu.atmegatimer,></cck.text.printer,></avrora.sim.radio.cc1000radio.spitic></jintgen.gen.disassembler.decoder,></avrora.sim.energy.energycontrol,></cck.util.util.error,></avrora.monitors.tracemonitor.mon,></cck.text.terminal,></avrora.sim.mcu.spi.spcrreg,></cck.parser.abstractparseexception,></avrora.stack.statecache.state,></avrora.sim.util.memtimer,></jintgen.isdl.operandtypedecl.access></avrora.stack.isea.iseabstractstate.ele></jintgen.gen.disassembler.decoder,>			(A)
chained> <avrora.stack.isea.iseabstractstate.ele ment,="" read=""> <jiintgen.isdl.operandtypedecl.access or,="" polymorphic=""> <avrora.sim.util.memtimer, timer_state=""> <avrora.stack.statecache.state, onfrontier=""> <cck.parser.abstractparseexception, specialconstructor=""> <avrora.sim.mcu.spi.spcrreg, prev_spie=""> <cck.text.terminal, htmlcolors=""> <avrora.monitors.tracemonitor.mon, nesting=""> <cck.util.util.error, stacktraces=""> <avrora.sim.energy.energycontrol, active=""> <jiintgen.gen.disassembler.decoder, multiple=""> <avrora.sim.radio.cc1000radio.spitic activated="" ker,=""> <cck.text.printer, begline=""> <avrora.sim.mcu.atmegatimer, timerenabled=""> <avrora.sim.mcu.atmegatimer, timerenabled=""> <avrora.sim.mcu.adc.controlregister, converting=""> <avrora.sim.mcu.spi.transferevent, transmitting=""> <cck.text.status, timing=""> <avrora.gui.graphevents.myvector, current=""> <cck.test.testengine, verbose=""> <avrora.sim.platform.externalflash, isreading=""> <avrora.syntax.atmel.atmelparsertoke (a)="" (a)<="" <avrora.syntax.atmel.atmelparsertoke="" td=""><td></td><td></td><td>(A)</td></avrora.syntax.atmel.atmelparsertoke></avrora.sim.platform.externalflash,></cck.test.testengine,></avrora.gui.graphevents.myvector,></cck.text.status,></avrora.sim.mcu.spi.transferevent,></avrora.sim.mcu.adc.controlregister,></avrora.sim.mcu.atmegatimer,></avrora.sim.mcu.atmegatimer,></cck.text.printer,></avrora.sim.radio.cc1000radio.spitic></jiintgen.gen.disassembler.decoder,></avrora.sim.energy.energycontrol,></cck.util.util.error,></avrora.monitors.tracemonitor.mon,></cck.text.terminal,></avrora.sim.mcu.spi.spcrreg,></cck.parser.abstractparseexception,></avrora.stack.statecache.state,></avrora.sim.util.memtimer,></jiintgen.isdl.operandtypedecl.access></avrora.stack.isea.iseabstractstate.ele>			(A)
ment, read>   cjintgen.isdl.OperandTypeDecl.Access or, polymorphic>   cavrora.sim.util.MemTimer, timer_state>   cavrora.stack.StateCache.State, onFrontier>   cck.parser.AbstractParseException, specialConstructor>   cavrora.sim.mcu.SPI.SPCRReg, prev_spie>   cck.text.Terminal, htmlColors>   cavrora.monitors.TraceMonitor.Mon, nesting>   cck.util.Util.Error, STACKTRACES> (C)   cavrora.sim.energy.EnergyControl, active>   cjintgen.gen.disassembler.Decoder, multiple>   cavrora.sim.radio.CC1000Radio.SPITic ker, activated>   cck.text.Printer, begLine> (A)   cavrora.sim.mcu.ATMegaTimer, timerEnabled>   cavrora.sim.mcu.ADC.ControlRegister, converting> (A)   cavrora.sim.mcu.SPI.TransferEvent, transmitting> (A)   cavrora.gui.GraphEvents.MyVector, current> (C)   cck.text.Terminal, useColors> (A)   cck.text.Terminal, useColors> (A)   cavrora.sim.platform.ExternalFlash, isReading> (A)   cavrora.syntax.atmel.AtmelParserToke (A)   cavrora.syntax.atmel.AtmelParserToke			(B)
or, polymorphic> <avrora.sim.util.memtimer, timer_state=""> <avrora.stack.statecache.state, onfrontier=""> <cck.parser.abstractparseexception, specialconstructor=""> <avrora.sim.mcu.spi.spcrreg, prev_spie=""> <cck.text.terminal, htmlcolors=""> (B)  <avrora.monitors.tracemonitor.mon, nesting=""> <cck.util.util.error, stacktraces=""> (C)  <avrora.sim.energy.energycontrol, active=""> <jiintgen.gen.disassembler.decoder, multiple=""> <avrora.sim.radio.cc1000radio.spitic activated="" ker,=""> <cck.text.printer, begline=""> <avrora.sim.mcu.atmegatimer, timerenabled=""> <avrora.sim.mcu.adc.controlregister, converting=""> <avrora.sim.mcu.api.transferevent, transmitting=""> <avrora.sim.mcu.spi.transferevent, converting=""> <avrora.gui.graphevents.myvector, current=""> <cck.text.testengine, verbose=""> <avrora.sim.platform.externalflash, isreading=""> <avrora.syntax.atmel.atmelparsertoke (a)="" <avrora.syntax.atmel.atmelparsertoke="" <avrora.syntax.atmel.atmelparsertoke<="" a=""></avrora.syntax.atmel.atmelparsertoke></avrora.sim.platform.externalflash,></cck.text.testengine,></avrora.gui.graphevents.myvector,></avrora.sim.mcu.spi.transferevent,></avrora.sim.mcu.api.transferevent,></avrora.sim.mcu.adc.controlregister,></avrora.sim.mcu.atmegatimer,></cck.text.printer,></avrora.sim.radio.cc1000radio.spitic></jiintgen.gen.disassembler.decoder,></avrora.sim.energy.energycontrol,></cck.util.util.error,></avrora.monitors.tracemonitor.mon,></cck.text.terminal,></avrora.sim.mcu.spi.spcrreg,></cck.parser.abstractparseexception,></avrora.stack.statecache.state,></avrora.sim.util.memtimer,>			(B)
timer_state> <avrora.stack.statecache.state, onfrontier=""> <cck.parser.abstractparseexception, specialconstructor=""> <avrora.sim.mcu.spi.spcrreg, prev_spie=""> <cck.text.terminal, htmlcolors=""> (B)  <avrora.monitors.tracemonitor.mon, nesting=""> <cck.util.util.error, stacktraces=""> (C)  <avrora.sim.energy.energycontrol, active=""> <jiintgen.gen.disassembler.decoder, multiple=""> <avrora.sim.radio.cc1000radio.spitic activated="" ker,=""> <cck.text.printer, begline=""> <avrora.sim.mcu.atmegatimer, timerenabled=""> <avrora.sim.mcu.adc.controlregister, (a)="" converting=""> <avrora.sim.mcu.spi.transferevent, transmitting=""> <cck.text.status, timing=""> <avrora.gui.graphevents.myvector, current=""> <cck.test.testengine, verbose=""> <avrora.sim.platform.externalflash, isreading=""> <avrora.syntax.atmel.atmelparsertoke (a)="" <avrora.syntax.atmel.atmelparsertoke="" <avrora.syntax.atmel.atmelparsertoke<="" a=""></avrora.syntax.atmel.atmelparsertoke></avrora.sim.platform.externalflash,></cck.test.testengine,></avrora.gui.graphevents.myvector,></cck.text.status,></avrora.sim.mcu.spi.transferevent,></avrora.sim.mcu.adc.controlregister,></avrora.sim.mcu.atmegatimer,></cck.text.printer,></avrora.sim.radio.cc1000radio.spitic></jiintgen.gen.disassembler.decoder,></avrora.sim.energy.energycontrol,></cck.util.util.error,></avrora.monitors.tracemonitor.mon,></cck.text.terminal,></avrora.sim.mcu.spi.spcrreg,></cck.parser.abstractparseexception,></avrora.stack.statecache.state,>			(B)
onFrontier> <pre> <pre></pre></pre>			(A)
specialConstructor> <avrora.sim.mcu.spi.spcrreg, prev_spie=""> <cck.text.terminal, htmlcolors=""> (B) <avrora.monitors.tracemonitor.mon, nesting=""> <cck.util.util.error, stacktraces=""> (C) <avrora.sim.energy.energycontrol, active=""> <jintgen.gen.disassembler.decoder, multiple=""> <avrora.sim.radio.cc1000radio.spitic activated="" ker,=""> <cck.text.printer, begline=""> <avrora.sim.mcu.atmegatimer, timerenabled=""> <avrora.sim.mcu.adc.controlregister, converting=""> <avrora.sim.mcu.spi.transferevent, transmitting=""> <avrora.sim.mcu.spi.transferevent, converting=""> <avrora.gui.graphevents.myvector, current=""> <avrora.gui.graphevents.myvector, current=""> <acck.text.testengine, verbose=""> (A) <acck.text.terminal, usecolors=""> (A) <avrora.sim.platform.externalflash, isreading=""> <avrora.syntax.atmel.atmelparsertoke (a)<="" td=""><td></td><td></td><td>(B)</td></avrora.syntax.atmel.atmelparsertoke></avrora.sim.platform.externalflash,></acck.text.terminal,></acck.text.testengine,></avrora.gui.graphevents.myvector,></avrora.gui.graphevents.myvector,></avrora.sim.mcu.spi.transferevent,></avrora.sim.mcu.spi.transferevent,></avrora.sim.mcu.adc.controlregister,></avrora.sim.mcu.atmegatimer,></cck.text.printer,></avrora.sim.radio.cc1000radio.spitic></jintgen.gen.disassembler.decoder,></avrora.sim.energy.energycontrol,></cck.util.util.error,></avrora.monitors.tracemonitor.mon,></cck.text.terminal,></avrora.sim.mcu.spi.spcrreg,>			(B)
prev_spie> <pre></pre>			(B)
<avrora.monitors.tracemonitor.mon, nesting=""> <cck.util.util.error, stacktraces=""> (C) <avrora.sim.energy.energycontrol, active=""> <jintgen.gen.disassembler.decoder, multiple=""> <avrora.sim.radio.cc1000radio.spitic activated="" ker,=""> <avrora.sim.mcu.atmegatimer, timerenabled=""> <avrora.sim.mcu.adc.controlregister, converting=""> <avrora.sim.mcu.spi.transferevent, transmitting=""> <avrora.sim.mcu.spi.transferevent, converting=""> <avrora.gui.graphevents.myvector, current=""> <avrora.gui.graphevents.myvector, current=""> <avrora.sim.mcu.spi.transferevent, converting=""> <avrora.gui.graphevents.myvector, current=""> <a rowspan="2"> <avrora.gui.graphevents.myvector, current=""> <a rowspan="2"> <a rowspan="&lt;/td"><td></td><td></td><td>(A)</td></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></a></avrora.gui.graphevents.myvector,></a></avrora.gui.graphevents.myvector,></avrora.gui.graphevents.myvector,></avrora.gui.graphevents.myvector,></avrora.gui.graphevents.myvector,></avrora.gui.graphevents.myvector,></avrora.gui.graphevents.myvector,></avrora.gui.graphevents.myvector,></avrora.gui.graphevents.myvector,></avrora.sim.mcu.spi.transferevent,></avrora.gui.graphevents.myvector,></avrora.gui.graphevents.myvector,></avrora.sim.mcu.spi.transferevent,></avrora.sim.mcu.spi.transferevent,></avrora.sim.mcu.adc.controlregister,></avrora.sim.mcu.atmegatimer,></avrora.sim.radio.cc1000radio.spitic></jintgen.gen.disassembler.decoder,></avrora.sim.energy.energycontrol,></cck.util.util.error,></avrora.monitors.tracemonitor.mon,>			(A)
nesting> <cck.util.util.error, stacktraces=""> (C)  <avrora.sim.energy.energycontrol, active=""> <jintgen.gen.disassembler.decoder, multiple=""> <avrora.sim.radio.cc1000radio.spitic activated="" ker,=""> <cck.text.printer, begline=""> <avrora.sim.mcu.atmegatimer, timerenabled=""> <avrora.sim.mcu.adc.controlregister, converting=""> <avrora.sim.mcu.spi.transferevent, transmitting=""> <avrora.sim.mcu.spi.transferevent, converting=""> <avrora.gui.graphevents.myvector, current=""> <cck.text.testengine, verbose=""> <ack.text.testengine, verbose=""> <ack.text.terminal, usecolors=""> <avrora.sim.platform.externalflash, isreading=""> <avrora.syntax.atmel.atmelparsertoke (a)<="" td=""><td></td><td><cck.text.terminal, htmlcolors=""></cck.text.terminal,></td><td>(B)</td></avrora.syntax.atmel.atmelparsertoke></avrora.sim.platform.externalflash,></ack.text.terminal,></ack.text.testengine,></cck.text.testengine,></avrora.gui.graphevents.myvector,></avrora.sim.mcu.spi.transferevent,></avrora.sim.mcu.spi.transferevent,></avrora.sim.mcu.adc.controlregister,></avrora.sim.mcu.atmegatimer,></cck.text.printer,></avrora.sim.radio.cc1000radio.spitic></jintgen.gen.disassembler.decoder,></avrora.sim.energy.energycontrol,></cck.util.util.error,>		<cck.text.terminal, htmlcolors=""></cck.text.terminal,>	(B)
<pre><avrora.sim.energy.energycontrol, active="">  <jintgen.gen.disassembler.decoder, multiple="">  <avrora.sim.radio.cc1000radio.spitic activated="" ker,="">      <cck.text.printer, begline="">     <avrora.sim.mcu.atmegatimer, timerenabled="">  <avrora.sim.mcu.adc.controlregister, converting="">  <avrora.sim.mcu.spi.transferevent, transmitting="">          <cck.text.status, timing="">         <avrora.gui.graphevents.myvector,< td=""><td></td><td>· · · · · · · · · · · · · · · · · · ·</td><td>(B)</td></avrora.gui.graphevents.myvector,<></cck.text.status,></avrora.sim.mcu.spi.transferevent,></avrora.sim.mcu.adc.controlregister,></avrora.sim.mcu.atmegatimer,></cck.text.printer,></avrora.sim.radio.cc1000radio.spitic></jintgen.gen.disassembler.decoder,></avrora.sim.energy.energycontrol,></pre>		· · · · · · · · · · · · · · · · · · ·	(B)
active> <ip><ip><ip><ip><ip><ip><ip><ip><ip><ip></ip></ip></ip></ip></ip></ip></ip></ip></ip></ip>		<cck.util.util.error, stacktraces=""></cck.util.util.error,>	(C)
multiple> <avrora.sim.radio.cc1000radio.spitic activated="" ker,=""> <avrora.sim.radio.cc1000radio.spitic activated="" ker,=""> <avrora.sim.mcu.atmegatimer, timerenabled=""> <avrora.sim.mcu.adc.controlregister, converting=""> <avrora.sim.mcu.spi.transferevent, transmitting=""> <avrora.sim.mcu.spi.transferevent, converting=""> <avrora.gui.graphevents.myvector, current=""> <avrora.gui.graphevents.myvector, current=""> <avrora.gui.graphevents.myvector, current=""> <avrora.sim.maxmean, somedata=""> <avrora.sim.maxmean, somedata=""> <avrora.sim.platform.externalflash, isreading=""> <avrora.syntax.atmel.atmelparsertoke <avrora.syntax.atmel.atmelparsertoke="" <avrora.syntax.atmel.atmelparsertoke<="" a=""></avrora.syntax.atmel.atmelparsertoke></avrora.sim.platform.externalflash,></avrora.sim.maxmean,></avrora.sim.maxmean,></avrora.gui.graphevents.myvector,></avrora.gui.graphevents.myvector,></avrora.gui.graphevents.myvector,></avrora.sim.mcu.spi.transferevent,></avrora.sim.mcu.spi.transferevent,></avrora.sim.mcu.adc.controlregister,></avrora.sim.mcu.atmegatimer,></avrora.sim.radio.cc1000radio.spitic></avrora.sim.radio.cc1000radio.spitic>			
ker, activated> <pre></pre>		multiple>	
<avrora.sim.mcu.atmegatimer, timerenabled=""> <avrora.sim.mcu.adc.controlregister, converting=""> <avrora.sim.mcu.spi.transferevent, transmitting=""> <cck.text.status, timing=""> <avrora.gui.graphevents.myvector, current=""> <cck.test.testengine, verbose=""> <cck.stat.minmaxmean, somedata=""> <cck.text.terminal, usecolors=""> <avrora.sim.platform.externalflash, isreading=""> <avrora.syntax.atmel.atmelparsertoke (a)="" <="" arrora.syntax.atmel.atmelparsertoke<="" td=""><td></td><td></td><td></td></avrora.syntax.atmel.atmelparsertoke></avrora.sim.platform.externalflash,></cck.text.terminal,></cck.stat.minmaxmean,></cck.test.testengine,></avrora.gui.graphevents.myvector,></cck.text.status,></avrora.sim.mcu.spi.transferevent,></avrora.sim.mcu.adc.controlregister,></avrora.sim.mcu.atmegatimer,>			
timerEnabled> <avrora.sim.mcu.adc.controlregister, converting=""> <avrora.sim.mcu.spi.transferevent, transmitting=""> <cck.text.status, timing=""> <avrora.gui.graphevents.myvector, current=""> <cck.test.testengine, verbose=""> <cck.stat.minmaxmean, somedata=""> <cck.text.terminal, usecolors=""> <avrora.sim.platform.externalflash, isreading=""> <avrora.syntax.atmel.atmelparsertoke (a)<="" td=""><td></td><td></td><td></td></avrora.syntax.atmel.atmelparsertoke></avrora.sim.platform.externalflash,></cck.text.terminal,></cck.stat.minmaxmean,></cck.test.testengine,></avrora.gui.graphevents.myvector,></cck.text.status,></avrora.sim.mcu.spi.transferevent,></avrora.sim.mcu.adc.controlregister,>			
converting> <avrora.sim.mcu.spi.transferevent, transmitting=""> <cck.text.status, timing=""> <avrora.gui.graphevents.myvector, current=""> <cck.test.testengine, verbose=""> <cck.test.minmaxmean, somedata=""> <cck.text.terminal, usecolors=""> <avrora.sim.platform.externalflash, isreading=""> <avrora.syntax.atmel.atmelparsertoke (a)<="" (c)="" <avrora.syntax.atmel.atmelparsertoke="" td=""><td></td><td>timerEnabled&gt;</td><td></td></avrora.syntax.atmel.atmelparsertoke></avrora.sim.platform.externalflash,></cck.text.terminal,></cck.test.minmaxmean,></cck.test.testengine,></avrora.gui.graphevents.myvector,></cck.text.status,></avrora.sim.mcu.spi.transferevent,>		timerEnabled>	
transmitting> <cck.text.status, timing=""> <avrora.gui.graphevents.myvector, current=""> <cck.test.testengine, verbose=""> <cck.stat.minmaxmean, somedata=""> <cck.text.terminal, usecolors=""> <avrora.sim.platform.externalflash, isreading=""> <avrora.syntax.atmel.atmelparsertoke (a)<="" td=""><td></td><td>converting&gt;</td><td></td></avrora.syntax.atmel.atmelparsertoke></avrora.sim.platform.externalflash,></cck.text.terminal,></cck.stat.minmaxmean,></cck.test.testengine,></avrora.gui.graphevents.myvector,></cck.text.status,>		converting>	
<avrora.gui.graphevents.myvector, current=""> <cck.test.testengine, verbose=""> (A)  <cck.stat.minmaxmean, somedata=""> (C)  <cck.text.terminal, usecolors=""> (A)  <avrora.sim.platform.externalflash, isreading=""> <avrora.syntax.atmel.atmelparsertoke (a)<="" td=""><td></td><td>t and the second se</td><td>(C)</td></avrora.syntax.atmel.atmelparsertoke></avrora.sim.platform.externalflash,></cck.text.terminal,></cck.stat.minmaxmean,></cck.test.testengine,></avrora.gui.graphevents.myvector,>		t and the second se	(C)
current> <cck.test.testengine, verbose=""> (A)  <cck.stat.minmaxmean, somedata=""> (C)  <cck.text.terminal, usecolors=""> (A)  <avrora.sim.platform.externalflash, isreading=""> <avrora.syntax.atmel.atmelparsertoke (a)<="" td=""><td></td><td></td><td></td></avrora.syntax.atmel.atmelparsertoke></avrora.sim.platform.externalflash,></cck.text.terminal,></cck.stat.minmaxmean,></cck.test.testengine,>			
<pre><cck.stat.minmaxmean, somedata=""> (C)   <cck.text.terminal, usecolors=""> (A)   <avrora.sim.platform.externalflash,< td=""><td></td><td></td><td>(B)</td></avrora.sim.platform.externalflash,<></cck.text.terminal,></cck.stat.minmaxmean,></pre>			(B)
<pre><cck.text.terminal, usecolors=""> (A) <avrora.sim.platform.externalflash,< td=""><td></td><td></td><td></td></avrora.sim.platform.externalflash,<></cck.text.terminal,></pre>			
<avrora.sim.platform.externalflash, isreading=""> <avrora.syntax.atmel.atmelparsertoke (a)<="" td=""><td></td><td></td><td></td></avrora.syntax.atmel.atmelparsertoke></avrora.sim.platform.externalflash,>			
isReading> <avrora.syntax.atmel.atmelparsertoke (a)<="" td=""><td></td><td></td><td></td></avrora.syntax.atmel.atmelparsertoke>			
		isReading>	
			(A)

Uses	Replace Type Code with State (ST)		
	<avrora.syntax.objdump.objdumpparse jjmatchedkind="" rtokenmanager,=""></avrora.syntax.objdump.objdumpparse>	(A)	
	<avrora.test.probes.probeparsertokenm anager,="" jjmatchedkind=""></avrora.test.probes.probeparsertokenm>	(A)	
	<avrora.sim.platform.externalflash, isselected=""></avrora.sim.platform.externalflash,>	(B)	
	<avrora.sim.radio.cc2420radio,config Command&gt;</avrora.sim.radio.cc2420radio,config 	(A)	

Back To Top