Verilog Translator - Bug #9212

ru.ispras.verilog.parser.VerilogVisVerilog2SmvTestCase.runTest_Vlunc_vlunc: Module 'transform' cannot be found

08/11/2018 06:08 PM - Sergey Smolov

Status:	Closed	**Start date:**	08/11/2018
Priority:	High	**Due date:**	
Assignee:	Alexander Kamkin	**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:	0.1		
Detected in build:	master	**Published in build:**	
Platform:			

Description

Here is the test output:

```
Module name: lunc
Including file '/home/ssedai/projects/veritrans/src/test/verilog/verilog2smv-vis-tests/Vlunc/vlunc
.v' ...
/home/ssedai/projects/veritrans/src/test/verilog/verilog2smv-vis-tests/Vlunc/vlunc.v line 115:31 m
issing COLON at '('
/home/ssedai/projects/veritrans/src/test/verilog/verilog2smv-vis-tests/Vlunc/vlunc.v line 115:36 m
ismatched input ':' expecting SEMI
/home/ssedai/projects/veritrans/src/test/verilog/verilog2smv-vis-tests/Vlunc/vlunc.v line 116:9 mi
smatched input '?' expecting LPAREN
/home/ssedai/projects/veritrans/src/test/verilog/verilog2smv-vis-tests/Vlunc/vlunc.v line 116:23 m
ismatched input ':' expecting SEMI
/home/ssedai/projects/veritrans/src/test/verilog/verilog2smv-vis-tests/Vlunc/vlunc.v line 117:9 mi
smatched input '?' expecting LPAREN
/home/ssedai/projects/veritrans/src/test/verilog/verilog2smv-vis-tests/Vlunc/vlunc.v line 117:14 m
ismatched input ':' expecting LPAREN
/home/ssedai/projects/veritrans/src/test/verilog/verilog2smv-vis-tests/Vlunc/vlunc.v line 118:9 mi
smatched input '?' expecting LPAREN
/home/ssedai/projects/veritrans/src/test/verilog/verilog2smv-vis-tests/Vlunc/vlunc.v line 118:26 m
ismatched input ':' expecting SEMI
/home/ssedai/projects/veritrans/src/test/verilog/verilog2smv-vis-tests/Vlunc/vlunc.v line 123:12 m
issing RPAREN at '('
/home/ssedai/projects/veritrans/src/test/verilog/verilog2smv-vis-tests/Vlunc/vlunc.v line 125:1 mi
smatched input 'else' expecting KW_END
/home/ssedai/projects/veritrans/src/test/verilog/verilog2smv-vis-tests/Vlunc/vlunc.v line 126:13 m
ismatched input '<=' expecting LPAREN
ru/ispras/verilog/parser/grammar/VerilogTreeBuilder.g: node from after line 99:52 mismatched tree
node: <mismatched token: [@516,2641:2643='end',<50>,127:4], resync=moduletransform(in,Lcmd,Ucmd,Nc
md,Ccmd,out);input[7:0]in;inputLcmd;inputUcmd;inputNcmd;inputCcmd;output[7:0]out;assignout=Lcmd?to
Lower(in):Ucmd?toUpper(in):Ncmd?in:Ccmd?changeCase(in):8'hxx;function[7:0]toLower;input[7:0]in;beg
in:_toLowerif(isUpper(in))toLower<=in+8'h20;elsetoLower<=in;endendfunctionfunction[7:0]toUpper;inp
ut[7:0]in;begin:_toUpperif> expecting <UP>
Starting the backend 'static-checker'...
Declaration of 'regIn[]' has been found: DECLARATION(regIn)
Declaration of 'dataOut[]' has been found: DECLARATION(dataOut)
Instance: c
Module: control
Port connection: null
Port connection: set clock
Port connection: null
Port connection: set reset
Port connection: null
Port connection: set in
Port connection: null
Port connection: set Lcmd
Port connection: null
```

```
Port connection: set Ucmd
Port connection: null
Port connection: set Ncmd
Port connection: null
Port connection: set Ccmd
Declaration of 'clock[]' has been found: DECLARATION(clock)
Declaration of 'reset[]' has been found: DECLARATION(reset)
Declaration of 'regIn[]' has been found: DECLARATION(regIn)
Declaration of 'Lcmd[]' has been found: DECLARATION(Lcmd)
Declaration of 'Ucmd[]' has been found: DECLARATION(Ucmd)
Declaration of 'Ncmd[]' has been found: DECLARATION(Ncmd)
Declaration of 'Ccmd[]' has been found: DECLARATION(Ccmd)
Instance: t
Declaration of 'regIn[]' has been found: DECLARATION(regIn)
Declaration of 'Lcmd[]' has been found: DECLARATION(Lcmd)
Declaration of 'Ucmd[]' has been found: DECLARATION(Ucmd)
Declaration of 'Ncmd[]' has been found: DECLARATION(Ncmd)
Declaration of 'Ccmd[]' has been found: DECLARATION(Ccmd)
Declaration of 'transformed[]' has been found: DECLARATION(transformed)
Declaration of 'clock[]' has been found: DECLARATION(clock)
Declaration of 'reset[]' has been found: DECLARATION(reset)
Declaration of 'dataOut[]' has been found: DECLARATION(dataOut)
Declaration of 'regIn[]' has been found: DECLARATION(regIn)
Declaration of 'dataOut[]' has been found: DECLARATION(dataOut)
Declaration of 'transformed[]' has been found: DECLARATION(transformed)
Declaration of 'regIn[]' has been found: DECLARATION(regIn)
Declaration of 'dataIn[]' has been found: DECLARATION(dataIn)
Declaration of 'clock[]' has been found: DECLARATION(clock)
Declaration of 'reset[]' has been found: DECLARATION(reset)
Declaration of 'dataIn[]' has been found: DECLARATION(dataIn)
Declaration of 'dataOut[]' has been found: DECLARATION(dataOut)
Declaration of 'clock[]' has been found: DECLARATION(clock)
Declaration of 'reset[]' has been found: DECLARATION(reset)
Declaration of 'dataIn[]' has been found: DECLARATION(dataIn)
Declaration of 'dataOut[]' has been found: DECLARATION(dataOut)
Declaration of 'Lcmd[]' has been found: DECLARATION(Lcmd)
Declaration of 'Ucmd[]' has been found: DECLARATION(Ucmd)
Declaration of 'Ncmd[]' has been found: DECLARATION(Ncmd)
Declaration of 'Ccmd[]' has been found: DECLARATION(Ccmd)
Declaration of 'prev[]' has been found: DECLARATION(prev)
Declaration of 'clock[]' has been found: DECLARATION(clock)
Declaration of 'reset[]' has been found: DECLARATION(reset)
Declaration of 'prev[]' has been found: DECLARATION(prev)
Declaration of 'prev[]' has been found: DECLARATION(prev)
Declaration of 'in[]' has been found: DECLARATION(in)
Declaration of 'load[]' has been found: DECLARATION(load)
Declaration of 'prev[]' has been found: DECLARATION(prev)
Declaration of 'clock[]' has been found: DECLARATION(clock)
Declaration of 'reset[]' has been found: DECLARATION(reset)
Declaration of 'Ncmd[]' has been found: DECLARATION(Ncmd)
Declaration of 'Lcmd[]' has been found: DECLARATION(Lcmd)
Declaration of 'Ucmd[]' has been found: DECLARATION(Ucmd)
Declaration of 'Ccmd[]' has been found: DECLARATION(Ccmd)
Declaration of 'load[]' has been found: DECLARATION(load)
Declaration of 'in[]' has been found: DECLARATION(in)
Declaration of 'Lcmd[]' has been found: DECLARATION(Lcmd)
Declaration of 'Ucmd[]' has been found: DECLARATION(Ucmd)
Declaration of 'Ncmd[]' has been found: DECLARATION(Ncmd)
Declaration of 'Ccmd[]' has been found: DECLARATION(Ccmd)
Declaration of 'Lcmd[]' has been found: DECLARATION(Lcmd)
Declaration of 'Ucmd[]' has been found: DECLARATION(Ucmd)
Declaration of 'Ncmd[]' has been found: DECLARATION(Ncmd)
Declaration of 'Ccmd[]' has been found: DECLARATION(Ccmd)
Declaration of 'Lcmd[]' has been found: DECLARATION(Lcmd)
Declaration of 'Ucmd[]' has been found: DECLARATION(Ucmd)
Declaration of 'Ncmd[]' has been found: DECLARATION(Ncmd)
Declaration of 'Ccmd[]' has been found: DECLARATION(Ccmd)
```

```
Declaration of 'Lcmd[]' has been found: DECLARATION(Lcmd)
Declaration of 'Ucmd[]' has been found: DECLARATION(Ucmd)
Declaration of 'Ncmd[]' has been found: DECLARATION(Ncmd)
Declaration of 'Ccmd[]' has been found: DECLARATION(Ccmd)
Declaration of 'Lcmd[]' has been found: DECLARATION(Lcmd)
Declaration of 'Ucmd[]' has been found: DECLARATION(Ucmd)
Declaration of 'Ncmd[]' has been found: DECLARATION(Ncmd)
Declaration of 'Ccmd[]' has been found: DECLARATION(Ccmd)
Declaration of 'clock[]' has been found: DECLARATION(clock)
Declaration of 'reset[]' has been found: DECLARATION(reset)
Declaration of 'in[]' has been found: DECLARATION(in)
Declaration of 'Lcmd[]' has been found: DECLARATION(Lcmd)
Declaration of 'Ucmd[]' has been found: DECLARATION(Ucmd)
Declaration of 'Ncmd[]' has been found: DECLARATION(Ncmd)
Declaration of 'Ccmd[]' has been found: DECLARATION(Ccmd)
Declaration of 'clock[]' has been found: DECLARATION(clock)
Declaration of 'reset[]' has been found: DECLARATION(reset)
Declaration of 'in[]' has been found: DECLARATION(in)
Declaration of 'Lcmd[]' has been found: DECLARATION(Lcmd)
Declaration of 'Ucmd[]' has been found: DECLARATION(Ucmd)
Declaration of 'Ncmd[]' has been found: DECLARATION(Ncmd)
Declaration of 'Ccmd[]' has been found: DECLARATION(Ccmd)
Starting the backend 'printer'...
module lunc(clock /* DECL: clock */, reset /* DECL: reset */, dataIn /* DECL: dataIn */, dataOut /
* DECL: dataOut */, clock /* DECL: clock */, reset /* DECL: reset */, dataIn /* DECL: dataIn */, d
ataOut /* DECL: dataOut */);
  input clock;
  inout reset;
  input [00000000000000000000000000000111:00000000000000000000000000000000] dataIn;
  output [00000000000000000000000000000111:00000000000000000000000000000000] dataOut;
  reg [00000000000000000000000000000111:00000000000000000000000000000000] dataOut;
  reg [00000000000000000000000000000111:00000000000000000000000000000000] regIn;
  wire [00000000000000000000000000000111:00000000000000000000000000000000] transformed;
  wire Lcmd;
  wire Ucmd;
  wire Ncmd;
  wire Ccmd;
  initial
    begin
       regIn /* DECL: regIn */ = 00000000000000000000000000000000;
       dataOut /* DECL: dataOut */ = 00000000000000000000000000000000;
    end
  /* DECL: control */
  control c
  (
    .clock(clock)
    .reset(reset)
    .in(regIn)
    .Lcmd(Lcmd)
    .Ucmd(Ucmd)
    .Ncmd(Ncmd)
    .Ccmd(Ccmd)
  );
  /* DECL: null */
  transform t
  (
    regIn
    Lcmd
    Ucmd
    Ncmd
    Ccmd
    transformed
  );
  always
    @(posedge clock)
      begin
        if(reset)
```

```
          /* ASSERT: reset */
          begin
             dataOut /* DECL: dataOut */ <= 00000000000000000000000000000000;
             regIn /* DECL: regIn */ <= 00000000000000000000000000000000;
          end
        else
          /* ASSERT:  ! reset */
          begin
             dataOut /* DECL: dataOut */ <= transformed;
             regIn /* DECL: regIn */ <= dataIn;
          end
        end
      end
endmodule

module control(clock /* DECL: clock */, reset /* DECL: reset */, in /* DECL: in */, Lcmd /* DECL:
Lcmd */, Ucmd /* DECL: Ucmd */, Ncmd /* DECL: Ncmd */, Ccmd /* DECL: Ccmd */, clock /* DECL: clock
 */, reset /* DECL: reset */, in /* DECL: in */, Lcmd /* DECL: Lcmd */, Ucmd /* DECL: Ucmd */, Ncm
d /* DECL: Ncmd */, Ccmd /* DECL: Ccmd */);
  input clock;
  input reset;
  input [00000000000000000000000000000111:00000000000000000000000000000000] in;
  output reg Lcmd;
  output reg Ucmd;
  output reg Ncmd;
  output reg Ccmd;
  reg Lcmd;
  reg Ucmd;
  reg Ncmd;
  reg Ccmd;
  wire load;
  reg [00000000000000000000000000000111:00000000000000000000000000000000] prev;
  initial
    begin
       Lcmd /* DECL: Lcmd */ = 00000000000000000000000000000000;
       Ucmd /* DECL: Ucmd */ = 00000000000000000000000000000000;
       Ncmd /* DECL: Ncmd */ = 00000000000000000000000000000001;
       Ccmd /* DECL: Ccmd */ = 00000000000000000000000000000000;
       prev /* DECL: prev */ = 00000000000000000000000000000000;
    end
  always
    @(posedge clock)
      if(reset)
        /* ASSERT: reset */
         prev /* DECL: prev */ <= 00000000000000000000000000000000;
      else
        /* ASSERT:  ! reset */
         prev /* DECL: prev */ <= in;
      end
  assign load /* DECL: load */ = (prev == 00011011);
  always
    @(posedge clock)
      if(reset)
        /* ASSERT: reset */
        begin
           Ncmd /* DECL: Ncmd */ <= 00000000000000000000000000000001;
           Lcmd /* DECL: Lcmd */ <= 00000000000000000000000000000000;
           Ucmd /* DECL: Ucmd */ <= 00000000000000000000000000000000;
           Ccmd /* DECL: Ccmd */ <= 00000000000000000000000000000000;
        end
      else
        /* ASSERT:  ! reset */
        if(load)
          /* ASSERT: load */
          begin
            case(in)
              01001100:
```

```
                /* ASSERT: (in === 01001100) */
                begin
                   Lcmd /* DECL: Lcmd */ <= 00000000000000000000000000000001;
                   Ucmd /* DECL: Ucmd */ <= 00000000000000000000000000000000;
                   Ncmd /* DECL: Ncmd */ <= 00000000000000000000000000000000;
                   Ccmd /* DECL: Ccmd */ <= 00000000000000000000000000000000;
                end
              01010101:
                /* ASSERT: (in === 01010101) */
                begin
                   Lcmd /* DECL: Lcmd */ <= 00000000000000000000000000000000;
                   Ucmd /* DECL: Ucmd */ <= 00000000000000000000000000000001;
                   Ncmd /* DECL: Ncmd */ <= 00000000000000000000000000000000;
                   Ccmd /* DECL: Ccmd */ <= 00000000000000000000000000000000;
                end
              01001110:
                /* ASSERT: (in === 01001110) */
                begin
                   Lcmd /* DECL: Lcmd */ <= 00000000000000000000000000000000;
                   Ucmd /* DECL: Ucmd */ <= 00000000000000000000000000000000;
                   Ncmd /* DECL: Ncmd */ <= 00000000000000000000000000000001;
                   Ccmd /* DECL: Ccmd */ <= 00000000000000000000000000000000;
                end
              01000011:
                /* ASSERT: (in === 01000011) */
                begin
                   Lcmd /* DECL: Lcmd */ <= 00000000000000000000000000000000;
                   Ucmd /* DECL: Ucmd */ <= 00000000000000000000000000000000;
                   Ncmd /* DECL: Ncmd */ <= 00000000000000000000000000000000;
                   Ccmd /* DECL: Ccmd */ <= 00000000000000000000000000000001;
                end
              default:
                /* ASSERT:  ! ((((in === 01001100) || (in === 01010101)) || (in === 01001110)) ||
(in === 01000011)) */
                begin
                   Lcmd /* DECL: Lcmd */ <= 0;
                   Ucmd /* DECL: Ucmd */ <= 0;
                   Ncmd /* DECL: Ncmd */ <= 0;
                   Ccmd /* DECL: Ccmd */ <= 0;
                end
            end
          end
        else
          /* ASSERT:  ! load */
          begin
          end
        end
      end
endmodule

Starting the backend 'design-elaborator'...
Expanding node 'MODULE(lunc)'...
Bindings: {clock=clock, reset=reset, dataIn=dataIn, dataOut=dataOut, regIn=regIn, transformed=tran
sformed, Lcmd=Lcmd, Ucmd=Ucmd, Ncmd=Ncmd, Ccmd=Ccmd}
Variables: {clock=DECLARATION(clock), reset=DECLARATION(reset), dataIn=DECLARATION(dataIn), dataOu
t=DECLARATION(dataOut), regIn=DECLARATION(regIn), transformed=DECLARATION(transformed), Lcmd=DECLA
RATION(Lcmd), Ucmd=DECLARATION(Ucmd), Ncmd=DECLARATION(Ncmd), Ccmd=DECLARATION(Ccmd)}
Module 'transform' cannot be found
```

To reproduce the bug, uncomment the **runTest_Vlunc_vlunc** method in **ru.ispras.verilog.parser.VerilogVisVerilog2SmvTestCase**

History

#1 - 08/14/2018 04:28 PM - Alexander Kamkin

- Status changed from New to Resolved

Grammar ambiguity has been avoided.

#2 - 08/14/2018 05:06 PM - Sergey Smolov

- % Done changed from 0 to 100

- Status changed from Resolved to Verified

#3 - 08/14/2018 05:06 PM - Sergey Smolov

- Status changed from Verified to Closed