VHDL to SystemVerilog Translation

Overview

SyoSil offers VHDL to SystemVerilog translation services, using our own in-house developed translation tool, capable of converting VHDL 1076-87/93 RTL into SystemVerilog 3.1a RTL. The tool supports translation of the RTL synthesis subsets of both languages, as de-facto standardized by major EDA synthesis tools.

EDA SystemVerilog Vendors

SystemVerilog is a relatively newborn language, and many EDA vendors currently develop tools to support SystemVerilog. Validation of front-ends and tool engines is made difficult by the fact that the amount of real SystemVerilog RTL designs is still minimal.

By translating existing VHDL designs from regression suites into SystemVerilog, SyoSil is able to create comprehensive validation suites in a very short time. Compared to most available SystemVerilog EDA test suites, which are comprised of very small single test cases, the translation process offered by SyoSil produces SystemVerilog designs of any complexity and size. This enables the EDA vendor to stress tool front-ends and engines to their maximum limit.

Translation of VHDL RTL requires intensive use of both simple and advanced RTL constructs in SystemVerilog – for instance most data types, interfaces and system function calls such as $left and $length. This ensures large test coverage of the various SystemVerilog language constructs.

Design IP Vendors

Existing commercial VHDL-to-Verilog 1364-95 RTL translators only support mapping of a limited set of VHDL constructs due to the simplicity of Verilog’95. The SyoSil translator is able to map all VHDL RTL constructs due to the rich feature set of SystemVerilog.

Design IP today only existing as VHDL can easily be converted into SystemVerilog, allowing vendors to sell their IP to Verilog based customers as well. This is made possible as most EDA vendors currently upgrade Verilog tools upwards to support SystemVerilog at no additional cost for the EDA tool customer.

Migration Path for System Vendors

Companies which traditionally have been using VHDL, can initiate an easy migration towards doing SystemVerilog design by letting SyoSil translate their VHDL code. This is doable both for existing complete designs, but also for designs currently being developed. Once translated into SystemVerilog, the development effort can continue in that language, as the SyoSil translator preserves structure and comments from the VHDL code in the generated SystemVerilog model.
**Validation of Translated RTL**

The SyoSil in-house translation flow includes trial synthesis runs of both the input VHDL and output SystemVerilog RTL models to prove the compatibility with a major EDA synthesis tool. Furthermore, a formal netlist-netlist compare proves that the synthesis functionality of the VHDL RTL model is preserved in the generated SystemVerilog RTL model.

The translation flow also checks that the simulation functionality of the VHDL model is preserved in the generated SystemVerilog model. A VHDL test bench is used to create random stimuli which are injected into the VHDL and SystemVerilog models simultaneously. A checker then verifies that these two models are equivalent by simulation. This methodology also checks that unknown values (X’es) are propagated equivalently inside the two designs.

The complete translation flow certifies that the SystemVerilog RTL delivered by SyoSil is 100% functionally equivalent to the original VHDL RTL.

**Supported VHDL RTL**

The SyoSil translator supports the full RTL subset of VHDL, as de facto defined by leading synthesis tools. This includes, but is not limited to, the following important VHDL RTL constructs:

- Both IEEE VHDL 1076-87 and 1076-1993
- Primary units: Entities, configurations, package headers
- Secondary units: Architectures, package bodies
- IEEE / standardized packages:
  - std_logic_1164
  - std_logic_arith
  - numeric_std
  - std_logic_unsigned
  - std_logic_signed
- Libraries and use constructs
- Subprograms & functions
- Signals, constants, variables
- Types, subtypes: Enumerated types, bit, bit_vector, integers, std_(u)logic(_vector), signed, unsigned, arrays, records
- Operators: Arithmetic (signed/unsigned), logic, shift.
- Processes and concurrent statements
- Sequential statements: if, case, loop, next, exit, return
- Generics and generate statements
- Assertions
- Comments in VHDL code

The translator tool furthermore recognizes common VHDL templates, and translates these into the appropriate SystemVerilog template. For instance, a VHDL process inferring registers is translated into a SystemVerilog always_ff process, with the appropriate sensitivity list.

All VHDL comments, their formatting and position in the RTL code are preserved. This allows for ongoing VHDL development work to continue seamlessly with the produced SystemVerilog code.
Translation Example

The example below shows how VHDL RTL is translated into SystemVerilog RTL, and how readability is preserved to allow further development of the SystemVerilog RTL.

VHDL

```vhdl
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

entity localvars is port {
  d : in  unsigned(3 downto 0);
  q : out unsigned(3 downto 0);
  clk : in std_ulogic;
  rst : in std_ulogic);
end localvars;

architecture rtl of localvars is begin
  process
  variable count : unsigned(3 downto 0);
  begin
    wait until (clk'event and clk = '1');
    if (rst = '1') then
      count := d;
    else
      count := count + 1;
    end if;
    q <= count;
  end process;
end rtl;
```

SystemVerilog

```systemverilog
module localvars(
  input logic unsigned [3:0] d,
  output logic unsigned [3:0] q,
  input logic clk,
  input logic rst
);

always_ff @(posedge clk) begin
  logic unsigned [3:0] count;
  if (rst == 1'b1) count = d;
  else count = count + 1;
  q <= count;
end
endmodule
```

VHDL Test Benches & Behavioral Models

With our expert level knowledge of both VHDL and SystemVerilog, we also offer to convert VHDL test benches and verification environments into SystemVerilog based environments, utilizing state-of-the-art object oriented test bench methodologies and assertions.

SyoSil also possesses the required skills to address translation of PSL assertions and 3rd party Hardware Verification Languages (HVL's) into SystemVerilog.

For more information related to the VHDL translation services, please contact

E-mail : sales@syosil.com

For more information on SyoSil, please visit

Web : www.syosil.com

SyoSil is a consulting company holding broad expertise within the field of System-on-Chip and ASIC solutions, including specification, methodologies, design and verification. We are specialized on verification strategies, advanced EDA verification tools including formal methods (property checking) and upcoming EDA tool languages such as SystemVerilog.

Combined with our knowledge of state-of-the-art EDA tools from major vendors, we are capable of materializing the benefits of SystemVerilog within your organization, leading to shorter design times and improved verification quality.