Scicos-HDL Tutorial 0.4

SCICOS-HDL

Scicos Hardware Description Language Tools

Scicos-HDL is a tool to design digital circuit system; it integrates the hardware circuit, algorithm and Scilab/Scicos environment as a plat for digital circuit design, simulation and Hardware Description Language generation. We develop EMB for helping the hardware engineers and other digital system engineers to design digital circuit faster and with lower cost. At present, Scicos-HDL has sequential logic library, combinational logic library and IPcore blocks, you can design interface circuit of digital system, micro-digital system. We have been doing our best to improve Scicos-HDL; we expect that Scicos-HDL can be as a Digital signal processing (DSP) system design tool in the future. Supports VHDL&Verilog Language.

1 Scicos-HDL

Scicos-HDL integrates the high-level algorithm development, simulation and INRIA Scilab and Scicos environment with VHDL & Verilog and design flow. It can help you create hardware representation of digital circuit system in an algorithm-friendly development environment. You can combine existing Scilab functions and Scicos blocks with Scicos-HDL blocks to link system-level design and implementation with system algorithm development. In this way, Scicos-HDL allows system, algorithm, and hardware designers to share a common development platform. Links INRIA Scilab and Scicos software with EDA.

Rules for this tutorial

The following rules are used：

>> This sign will guide you to get into the subdirectories and select the final operation. For example:

Edit>>palette>>HDL_Sequential_Lib, it means that please select the **Edit** menu, then select the **palette**, and at last execute **HDL_Sequential_Lib**.

This sign is a prompt, means there is significant information for you.

Bold-face It means the name of menu, the option of dialog box and so on, which you can click or select.

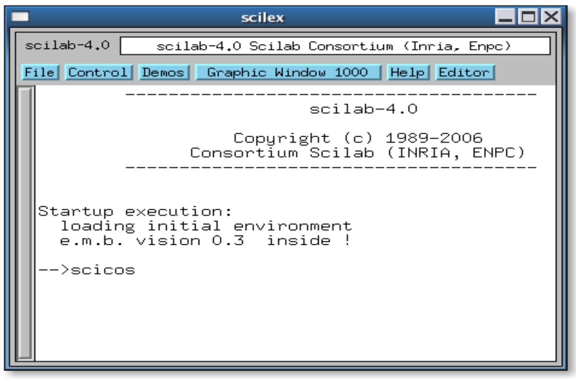
2 Create an example by Scicos-HDL

We will use an example a counter of digital circuit design, **counter4.cos**, to demonstrate the Scicos-HDL design flow.

The example has a counter block, a clock block, a display block, a binary code- decimal code converter, in put ports, out put ports and Scicos-HDL Compiler which are in the Scicos-HDL libraries.

After you have finished the design and simulation, then run **Scicos-HDL Compilers** to generate **VHDL / Verilog code** of your design. By this example you will know the whole flow of using Scicos-HDL to design circuit and generate **VHDL / Verilog** code.

Start the **Scilab**, first sight in the main command window is: **Scicos-HDL vision 0.3 inside!** It means that Scicos-HDL has been installed in **Scilab.**



```
scilab-4.0 Scilab Consortium (Inria, Enpc)

           ---------------------------------------
                         scilab-4.0

                   Copyright (c) 1989-2006
              Consortium Scilab (INRIA, ENPC)
           ---------------------------------------


Startup execution:
  loading initial environment
  e.m.b. vision 0.3  inside !

-->scicos
```

Figure 2.1 successful installation

Open **Scicos,** and then perform the following steps:

1、 Open **Edit>>palette>>HDL_Sequential_Lib** and **Edit>>palette>>HDL_cmd**
In **HDL_Sequential_Lib** library, select counter block, the interface of the block is: **counter4auto.**
In **HDL_cmd** library, select the blocks which the interfaces of the blocks are: **red clock, IN, OUT, 1, 00000.00, Scicos-HDLClock, A/D4** and **Compiler.**

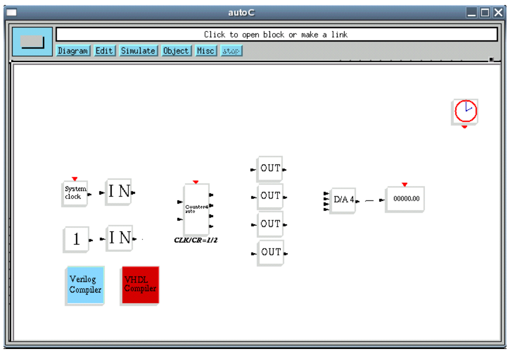
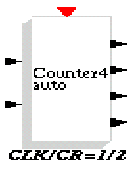


Figure 2.2 the blocks of this example

Counter4auto is a 4-bit binary counter with self-set up, the I/O ports:



In port 1: clock input;
In port 2: CR , ’ 1’ is effective;
Out ports 1—4: From high bits to low, binary bit
Red input port: Used in the simulation

2、 Click **Scicos >>Edit>>link** to link every blocks

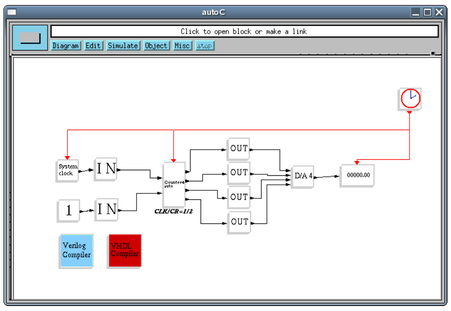
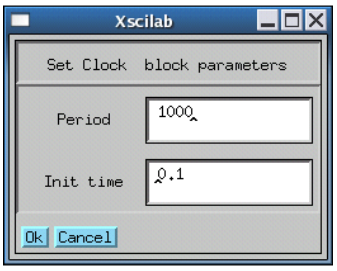


Figure 2.3 the finished design

3、 Save the model file

do not use Chinese name, here save as: **counter4.cos**

4、 Simulation parameter setting



Xscilab

Set Clock block parameters

Period	1000
Init time	0.1

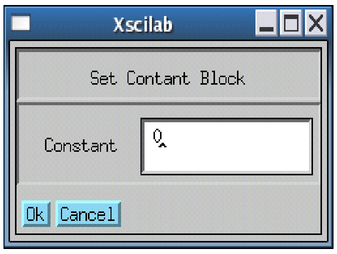
Ok Cancel

Period: 1000

NOTE: The CR input port is effective

Figure 2.4 clock setting

Run simulation, the display show **“0000”**, then change the CR port of the counter to ‘0’, simulate again. ’ ,simulate again.



Set Contant Block

Constant 0

Change CR input port

NOTE: change like this **0→1**。

Figure 2.5 CR port setting

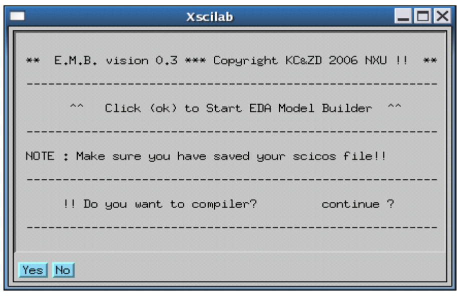
Simulate again, the display show the numbers.

Till now, you can run the **Scicos-HDL Compilers** to generate **VHDL / Verilog** codes

5、 Run **Scicos-HDL Compiler**

Scicos-HDL Compiler includes two compilers: VHDL compiler and Verilog compiler. And the operations for both compilers are same.

Click the **block** in your model file, there will be a **Scicos-HDL Compiler** dialog box, click "YES", next doing, "NO" return.

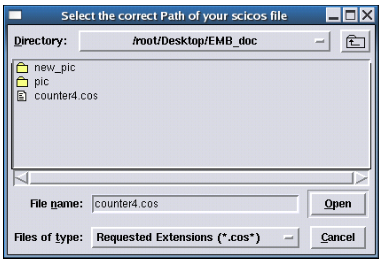


```
** E.M.B. vision 0.3 *** Copyright KC&ZD 2006 NXU !! **
-------------------------------------------------------
   ^^  Click (ok) to Start EDA Model Builder  ^^
-------------------------------------------------------
NOTE : Make sure you have saved your scicos file!!
-------------------------------------------------------
   !! Do you want to compiler?        continue ?
-------------------------------------------------------
```

Figure 2.6 Compiler dialog box

6、 Path selection

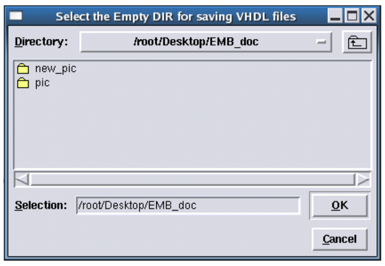
- NOTE: Make sure you have saved you model file without a Chinese name.
- Make sure the directory you have select for saving the VHDL or Verilog code is empty or at lest without the same name as your model file.
- Make sure to do these correctly


Select the correct Path of your scicos file

Directory: /root/Desktop/EMB_doc

new_pic
pic
counter4.cos

File name: counter4.cos
Files of type: Requested Extensions (*.cos*)


Select the Empty DIR for saving VHDL files

Directory: /root/Desktop/EMB_doc

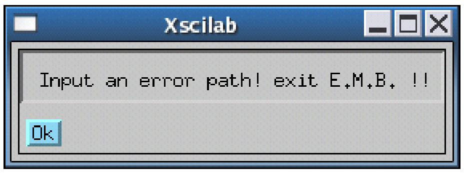
new_pic
pic

Selection: /root/Desktop/EMB_doc

Figure 2.7 .cos file path　　　　Figure 2.8 saving VHDL

7、 Dialog boxes

1. Error dialog box


Xscilab

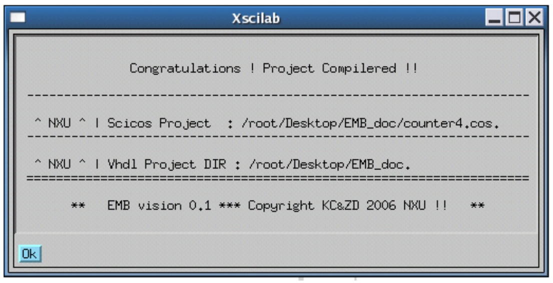
Input an error path! exit E.M.B. !!

Figure 2.9 Error dialog box

Error info: Error path, exit Scicos-HDL compiler.

2. Successfully compiled dialog box

info: Show the path of the model file and **VHDL / Verilog** code files directory and successfully compiled information.


```
Xscilab

        Congratulations ! Project Compilered !!
--------------------------------------------------------------
 ^ NXU ^ | Scicos Project  : /root/Desktop/EMB_doc/counter4.cos.
--------------------------------------------------------------
 ^ NXU ^ | Vhdl Project DIR : /root/Desktop/EMB_doc.
==============================================================
    **   EMB vision 0.1 *** Copyright KC&ZD 2006 NXU !!   **
```

Figure 2.10 successfully compiled

You can use other EDA tools to synthesize the **VHDL / Verilog** codes compiled by Scicos-HDL

3 EM.B. Libraries

We have developed 4 libraries for Scicos-HDL to design sequential logic circuit and combinational logic circuit, what innovative thought in Scicos-HDL is that we combine Scicos, Scicos-HDL with IPcore , this make Scicos and IPcore of EDA work together.

◆ Combinational logic library **HDL_Combinational_Lib**

This library has the basic components of digital system design, including Gate circuit, Multiplexer, Encoder, Decoder and BUS related components.

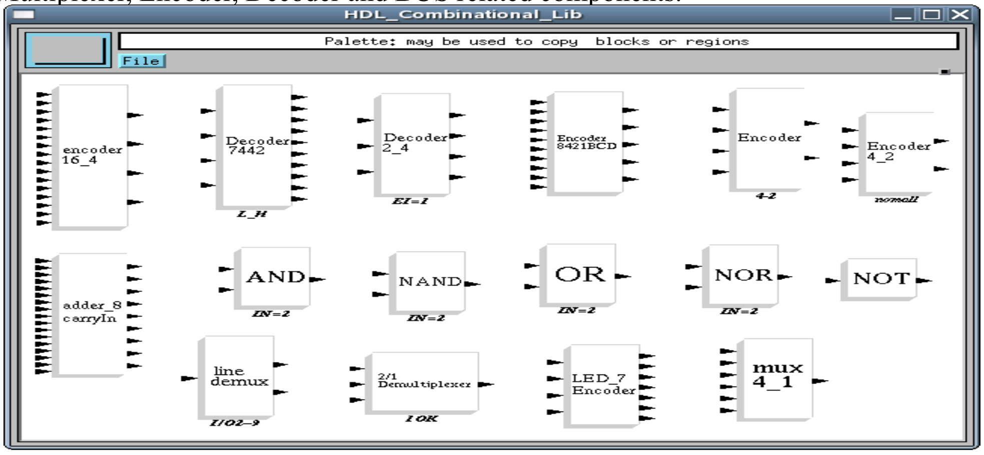


Figure3.1 Combinational logic library

◆ Sequential logic library **HDL_Sequential_Lib library**

This library includes D trigger,JK trigger,RS trigger without CR port, and D trigger,JK trigger,RS trigger with CR port, 4-bit counter with self-set up.

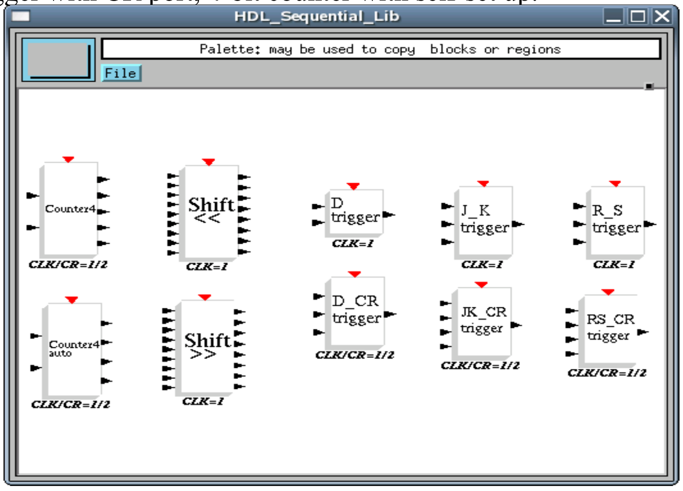


Figure 3.2 Sequential logic library

- Ipcore library **HDL_IPcore library**
  At present, we have Multiplexer, and decoder74138 and BUS related components.

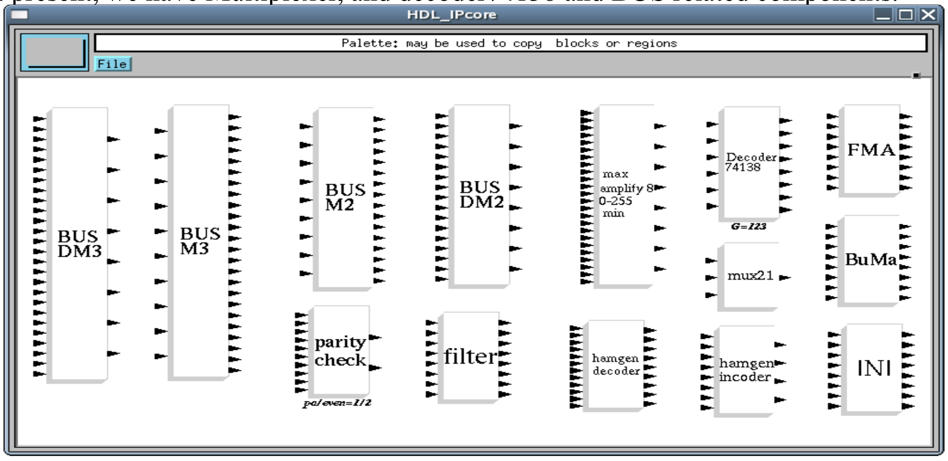


Figure 3.3 Ipcore library

- Digital system design basic library **HDL_cmd**

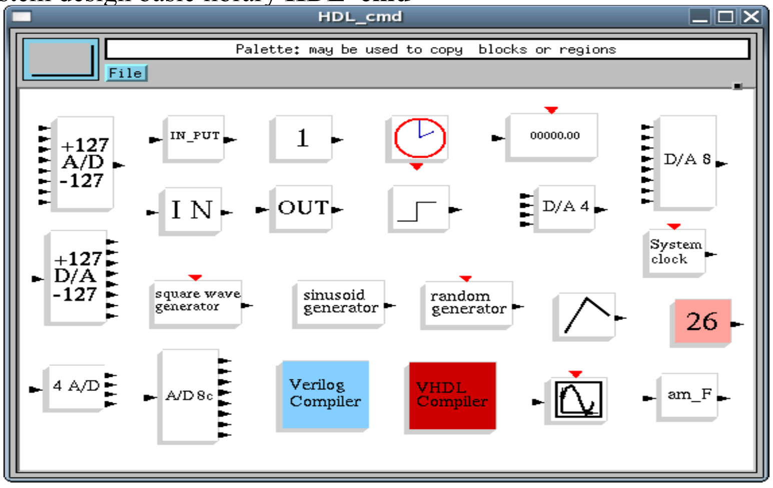
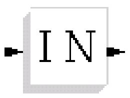


Figure 3.4 Digital system design basic library

4 Rules

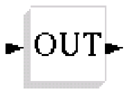
4.1 Design NOTE：

1) SCICOS-HDLIN



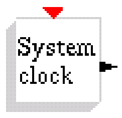
This is the input port of Scicos-HDL, the number of the block is equal to the one of the whole system input ports, each in-signal must through this block, it must be in every model file.

2) SCICOS-HDLOUT



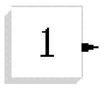
This is the output port of Scicos-HDL, the number of the block is equal to the one of the whole system output ports , each out-signal must through this block, it must be in every model file.

3) Scicos-HDL Clock



E.M.B. clock, used in sequential logic circuit

4) input



Scicos-HDL clock, used in sequential logic circuit

4.2 SCICOS-HDL Compiler

The compiler of Scicos-HDL--SCICOS-HDL Compiler, is the heart of Scicos-HDL, its main function is to convert Scilab/Scicos model file to standard **VHDL / Verilog** RTL code. And you can load these **VHDL / Verilog** code to FPGA.

1. How to operate the compiler

EMB
Compiler

It must be at every model file, when the whole system is finished, click this block to start compiler, then follow the dialog box

2. Scicos-HDL VHDL compiler

Verilog
Compiler

Scicos-HDL Verilog HDL compiler

Scicos-HDL numerical value

● NOTE

- Make sure you have put the block **Scicos-HDL IN** as **the input ports**, signals are transferred in through this block;
- Make sure you have put the block **Scicos-HDL OUT** as **the output ports**, signals are transferred out through this block;
- Self-connected is not allowed in every block;
- If one output port is needed to be connected with many other blocks ,use the block **LineDemux of HDL_Combinational_Lib** library;
- Make sure the path and name of model file are correct as the rule in Scicos;
- Make sure the directory of saving **VHDL / Verilog** code file is correct, empty or at least without the same name to the model file you will compile;
- Under the Scilab's license;
- Under the syntax of VHDL/Verilog;
- Make sure no super block in the model file.

The **VHDL / Verilog** codes generated by compilers can be used in following EDA tools.
Synplify Pro 7.6, Quartus® II,Mux+plus II ,ISE, Modelsim, etc.

5 The successful examples in Scicos-HDL

❖ 4-bit shift register

Register is a logic component as a memorizer in computer and other digital system, the main element of it is trigger, every trigger can store one binary bit, so N bits binary code need the same number of triggers.

This is a 4 bits shift register, it is composed of 4 triggers, and we use the fall edge of the clock, figure 4.1

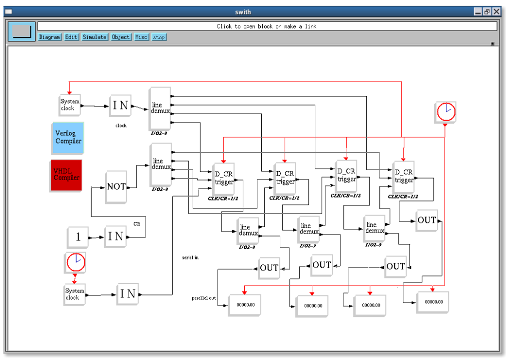


Figure 4.1 4 bits shift register

Synthesized the VHDL / Verilog codes in Synplify Pro 7.6

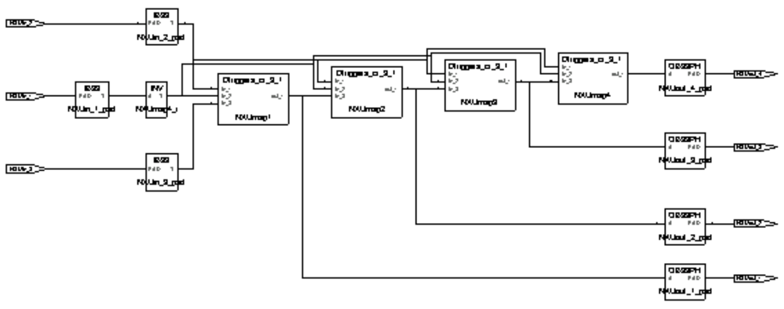


Figure 4.2 Synthesized in Synplify Pro 7.6

❖ Peripheral circuit of single chip

Scicos-HDL can help hardware engineer, especially the single chip engineer design the peripheral circuit of single chip or other digital circuit. When some digital systems are high speed system and need the high interface circuit, then FPGA is first choice, so what we do just use the blocks of Scicos-HDL to finish the interface circuit design, Scicos-HDL compiler will convert the hardware design to standard **VHDL / Verilog**, then load them to FPGA. It can shorten the design cycle for a project; what' s more better, it can help some engineers although they are not familiar with **VHDL / Verilog** use FPGA.

This is a peripheral circuit for a **single Chip**; it has an **A/D** interface, **LED** interface, **key board** interface, **D/A** interface, with a **flip-latch** integrated, **decoders** and **encoders**, designed by Scicos-HDL

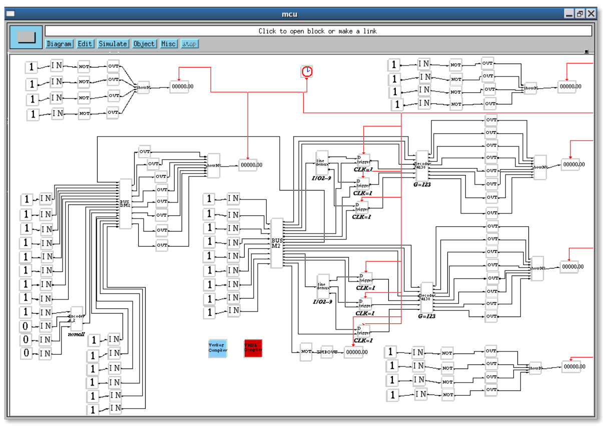
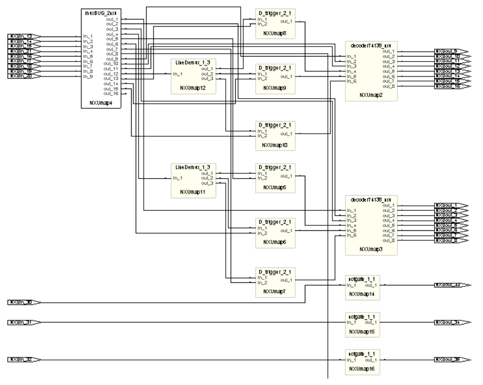


Figure 4.3 peripheral circuits for a single chip



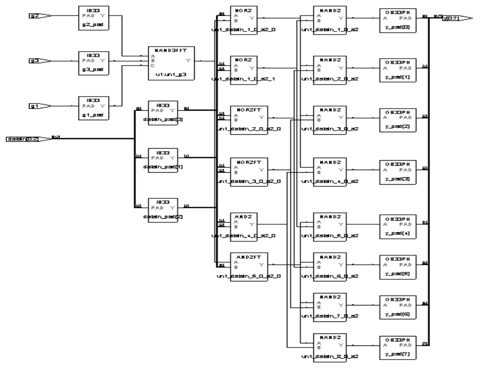


Figure 4.4 Synthesized in Synplify Pro 7.6 Figure 4.5 Synthesized in Synplify Pro 7.6

Figure 4.4 is the **VHDL / Verilog** code of the interface circuit;

Figure 4.5 is the **VHDL / Verilog** code of one block named decoder74138 of the interface circuit.

❖ Decoder / encoder circuit

This is Decoder / encoder circuit; it has an **A/D** interface, **LED** interface, **key board** interface, **D/A** interface, with a **flip-latch** integrated, **decoders** and **encoders**, designed by Scicos-HDL

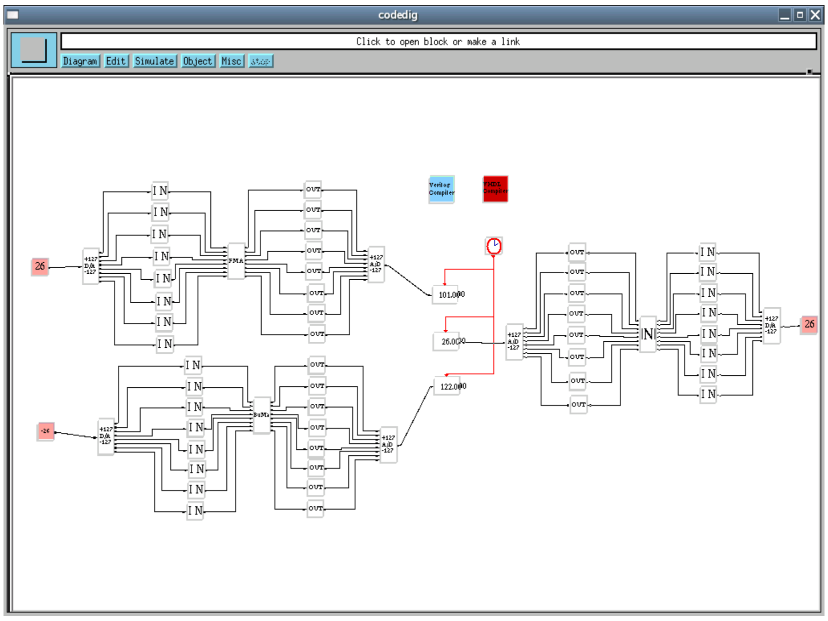
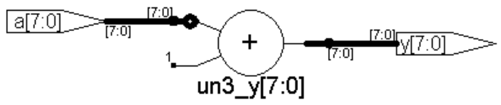


Figure 4.6 Decoder / encoder circuit



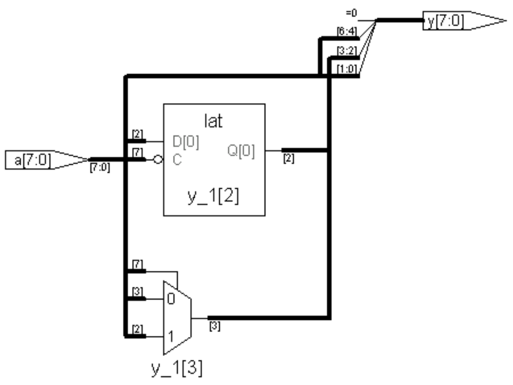


Figure 4.7 Synthesized in Synplify Pro 7.6 Figure 4.8 Synthesized in Synplify Pro 7.6

Figure 4.7 is the **VHDL / Verilog** code of the `Complement number`;
Figure 4.8 is the **VHDL / Verilog** code of one block named `Absolute value` of circuit.

6 Summarize

You can use the blocks in **Scicos-HDL** to create a hardware implementation of a system modeled in Scicos in sampled time. The **Scicos-HDL Compiler** block reads Scicos Model Files (**.cos**) that are built using **Scicos-HDL** and generates **VHDL / Verilog** code, hardware implementation, and simulation. **Scicos-HDL** makes Scicos have hardware design and simulation function. It builds a bridge between Scilab and EDA.