Dr. Bob Davidov

Интеграция VHDL (язык описания аппаратных средств) и Simulink моделей

Цель работы: освоение правил моделирования систем с объектами разного типа.

Задача работы: построение динамической модели включающей VHDL (Verilog) и Simulink объекты.

Приборы и принадлежности: Персональный компьютер с компилятором Verilog и Интегрированная среда МатЛАБ с Simulink.

введение

Для описания и построения электронных устройств используется множество языков, среди них и язык VHDL. МатЛАБ позволяет объединить "инородные" VHDL модели электронных устройств и собственные средства моделирования Simulink, что позволяет решать задачи моделирования и синтеза интегрированных систем, например, для целей управления.

ОБЩИЕ СВЕДЕНИЯ

Степень интеграции современных СБИС – «Систем на кристалле» (СнК) достигает нескольких десятков миллионов вентилей на кристалле. СБИС содержат программируемые процессорные ядра, специализированные логические блоки, модули памяти, интерфейсные и периферийные устройства, аналоговые и аналого-цифровые схемы.

Для реализации полного цикла проектирования СнК используется определенный набор программных продуктов и языков программирования способных описывать параллельно протекающие во времени процессы, поддерживать множество стилей описания аппаратуры и инженерных приложений. На **Рис. 1** приведены основные этапы проектирования СнК и различные языки и средства разработки, используемые для выполнения этих этапов.



Рис. 1. Этапы проектирования «Систем на кристалле» и области применения современных средств и языков программирования

Языки описания аппаратуры (HDL-языки) имеют две основные разновидности – языки низкого уровня (аналоги языков программирования типа ассемблера) и высокого уровня.

Языки низкого уровня ближе к аппаратным средствам, вследствие чего представляют для компиляторов потенциальные возможности создания проектов с более выигрышными параметрами. Платой за это является обычно жесткая ориентация на определенную аппаратуру и производящую ее фирму. Примерами таких языков могут служить языки PLDASM (фирмa Intel), AHDL (Фирма Altera) и ABEL (Фирма Zilinx).

Языки высокого уровня менее связаны с аппаратными платформами и поэтому более универсальны. Среди них наиболее распространены языки VHDL и Verilog.

МатЛАБ обеспечивает соединение Simulink и Verilog модулей через TCP/IP на одном или удаленных компьютерах.

ПОРЯДОК СОЗДАНИЯ МОДЕЛИ SIMULINK - VERILOG

- 1. Запустите MatLAB
- 2. Настройте МатЛАБ на рабочую папку.
- 3. В рабочем каталоге откройте (или создайте) модель с ModelSim блоком, например, valve_solenoid_control.mdl

4. Установите ModelSim командой setupmodelsim ('PropertyName', 'PropertyValue'...), например,

>> setupmodelsim

Identify the ModelSim installation to be configured for MATLAB and Simulink Do you want setupmodelsim to locate installed ModelSim executables [y]/n? y Select a ModelSim installation:

[1] C:\Modeltech_6.0c\win32 ModelSim SE 6.0c [0] None

>>[1]

Selected Modelsim installation: 1 Previous MATLAB startup file found in this installation of ModelSim: C:\Modeltech_6.0c\win32\..\tcl\ModelSimTclFunctionsForMATLAB.tcl Do you want to replace this file [y]/n? y Modelsim successfully configured for MATLAB and Simulink

>>[y]

5. Для запуска ModelSim наберите следующую команду к окне команд MatLAB:

>> vsim('tclstart', 'do plot.do','socketsimulink', 4449)

6. Запустите модель Simulink на выполнение

Примечание: Для последующих запусков четвертый пункт (setupmodelsim) выполнять не надо.

РЕДАКТИРОВАНИЯ VERILOG ФАЙЛА:

- 1. Запустите ModelSim
- 2. Настройте текущую папку
- 3. Отредактируйте *.v файл, например, VLV_SC.v
- 4. Создайте VHDL файл командой ModelSim: wrapverilog VLV_SC

УСТАНОВКА ВРЕМЕННЫХ ПАРАМЕТРОВ

- 1. Величина шага моделирования (Clock) в ModelSim задается параметром "resolution" в файле modelsim.ini
- 2. Чтобы установить период моделирования Simulink равный периоду ModelSim генератора (SYS_CLK) щелкните по блоку ModelSim в Simulink модели и выберите "Clocks" > "Period"
- 3. Ввод параметров моделирования Simulink осуществляется в окне, показанном на следующем рисунке.

Simulation time	
Start time: 🛄	Stop time: 4380
Solver options	
Type: Fixed-step	▼ Solver: ode3 (Bogacki-Shanpine) ▼
Periodic sample time constraint:	Unconstrained
Fixed-step size (fundamental sample time):	0.01
Tasking mode for periodic sample times:	SingleTasking 🗾
🔲 Higher priority value indicates higher task	k priority
Automatically handle data transfers betw	een tasks

Рис. 2. Параметры моделирования Simulink.

ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

Задание 1. Совместная работа ModelSIM и Simulink модулей

```
1. В среде ModelSim создайте файл, например, CNS1.v
  //-----
  // Design unit : PRED_02_VLV_CTRL
  // Filename: CNS.v
  \parallel
  // Description:
  //
  //
         Limitations:
  \parallel
  // Author: Bob D
  //
         Ac Ltd.
  \parallel
  // Simulator: ModelSim 6.0c
  // Version 1.0 by 13th April 2006, original release
  //-----
  // CNS = 100/2^((1600 - PPO2_SNS)/100)
  module CNS( S_CLK, PPO2_SNS, CNS_OUT );
  input S_CLK;
  input [15:0] PPO2_SNS;
                          // measured PPO2 in mATA
  output CNS_OUT;
  //parameter PPO2_SNS = 940; // range is 940 - 1600
  parameter PPO2_SAT = 1600; // max PPO2 in mATA
  parameter A = 25600;
  reg [7:0] CNS_OUT = 0;
                         // range is 0 - 660
  reg [9:0] ER;
```

```
reg [2:0] ER_INT;
                          // range is 0 - 7
reg [6:0] ER_FRACT;
                            // range is 0 - 99
reg [7:0] ER_FRACT_NUM;
reg [14:0] B_SHIFT_1;
reg [13:0] B_SHIFT_2;
reg [14:0] B_SUM;
always @ (posedge S_CLK)
//initial
 begin
   if (PPO2\_SNS \ge PPO2\_SAT)
    ER = 1;
   else
    begin
      if (PPO2_SNS < 940)
        begin
         ER = 660;
        end
      else
      ER = PPO2_SAT - PPO2_SNS[13:0];
    end
// $display("ER = %d %b", ER, ER);
   ER_INT = ER/100;
// $display("ER_INT = %d %b", ER_INT, ER_INT);
   ER_FRACT = ER - 100*ER_INT;
// $display("ER FRACT = %d %b", ER FRACT, ER FRACT);
   ER_FRACT_NUM = ER_FRACT*100/39;
// $display("ER_FRACT_NUM = %d %b", ER_FRACT_NUM, ER_FRACT_NUM);
   B_SHIFT_1 = 256 \ll ER_INT;
// $display("B_SHIFT_1 = %d %b", B_SHIFT_1, B_SHIFT_1);
   B SHIFT 2 = ER FRACT NUM << ER INT;
// $display("B_SHIFT_2 = %d %b", B_SHIFT_2, B_SHIFT_2);
   B_SUM = B_SHIFT_1 + B_SHIFT_2;
// $display("B_SUM = %d %b", B_SUM, B_SUM);
   CNS_OUT = A/B_SUM;
    $display("CNS_OUT = %d", CNS_OUT);
\parallel
end
endmodule
```

```
2. Командой wrapverilog < CNS1.v > создайте оболочку файла.
```

```
-----
```

```
-- Module cns VHDL Wrapper
```

-- Generated by The MathWorks wrapverilog tcl command

```
--
```

-- Generated on: 2006-04-13 01:28:46

```
--
-----
LIBRARY IEEE;
 USE IEEE.std_logic_1164.all;
ENTITY cns_wrap IS
  generic(
    PPO2_SAT
                    : integer := 1600;
    А
              : integer := 25600
  );
  port(
    S_CLK : in std_logic;
    PPO2_SNS : in std_logic_vector(15 downto 0);
CNS_OUT : out std_logic_vector(7 downto 0)
  );
END cns_wrap;
ARCHITECTURE rtl OF cns_wrap IS
component CNS
  generic(
    PPO2_SAT
                    : integer := 1600;
    А
             : integer := 25600
  );
  port(
    S CLK
                : in std_logic;
    PPO2_SNS : in std_logic_vector(15 downto 0);
CNS_OUT : out std_logic_vector(7 downto 0)
  );
end component;
FOR ALL : cns
 USE ENTITY work.cns(ignored);
BEGIN
 u cns: cns
  PORT MAP (
   S_CLK => S_CLK,
   PPO2_SNS => PPO2_SNS ,
   CNS_OUT => CNS_OUT
  );
END rtl;
```

3. Загрузите МатЛАБ.

4. Настройте Simulink как показано ниже.

Sconfiguration I	Parameters: CNS_TST/Configuration (Active)	×
Select: 	Simulation time Start time: 0.0 Solver options Type: Fixed-step Periodic sample time constraint: Unconstrained Fixed-step size (fundamental sample time): 0.1 Tasking mode for periodic sample times: Auto Higher priority value indicates higher task priority Automatically handle data transfers between tasks	
Comments Symbols Custom Code Debug Interface		~
	OK Cancel Help	Apply

Select:	Load from workspace		^
Solver Data Import/Export Optimization	Input: [t, u] Initial state: Mnitial		
E-Diagnostics	Save to workspace		
Data Validity	Time: tout		
- Type Conversion	States: xout		
Connectivity	🗹 Output: yout		
Model Referencing	Final states: xFinal		l'
- Hardware Implementation	Signal logging: logsout		
	Inspect signal logs when	n simulation is paused/stopped	
Comments	Save options		-
Symbols Custom Code	🔽 Limit data points to last:	1000 Decimation: 1	- P
Debug Interface	Format:	Array	

Рис. 3. Параметры конфигурации Simulink. Параметры конфигурации Simulink.

Seconfiguration	Parameters: CNS_TST/Configuration (Active) 🛛 🛛 🔀
Select:	Simulation and code generation
Solver Data Import/Export Optimization	Implement logic signals as boolean data (vs. double). Implement storage reuse
⊡-Diagnostics Sample Time Data Validity	Application lifespan (days) inf
Type Conversion Connectivity Connectivity Generatibility Model Referencing Hardware Implementation Model Referencing En-Real-Time Workshop Comments Symbols	Code generation Signals Image: Signa
Custom Code Debug Interface	Integer and fixed-point Remove code from floating-point to integer conversions that wraps out-of-range values
<u>s</u>	OK Cancel Help Apply

Solver Data Import/Export Data Import/Export System target file: grt.tlc Diagnostics C Data Validity C Type Conversion Generate HTML report Connectivity Launch report automatically Build process TLC options: Hardware Implementation Makefile configuration Model Referencing Makefile configuration Makefile configuration Generate makefile Symbols Generate makefile Debug Interface Generate code only Generate code	Select:	C Target selection	^
	Solver Data Import/Export Optimization Optimization Sample Time Obta Validity Optimize Conversion Connectivity Optimize Implementation Model Referencing Hardware Implementation Model Referencing Obta Referencing Comments Symbols Obta Code Debug Interface	System target file: grt.tlc Browse Language: C Documentation Image: C Generate HTML report Image: C Launch report automatically Image: C Build process TLC options: Makefile configuration Image: C Image: C Image: C <td< td=""><td></td></td<>	

Рис. 4. Параметры конфигурации Simulink.

5. В рабочем каталоге постройте модель CNS_TST.mdl



Рис. 5. Модель CNS_TST.mdl

6. Настройте блок VHDL Cosimulation

Ports Clock	ks Timescales	Connection Tcl						
Full HDL	Name	I/O Mode	Sample Tim	e Data	Type	Fraction	Length [Auto Fill
/cns_wrap	PPO2_SNS	Input	N/A	N/A	12/00	N/A		Hato Tim
∕cns_wraµ	57CN5_001	Output	-1	Inhe	rıt	N/A	ĺ	Delete Up Down
C ull HDL	Name	I/O Mode	Sample Time		The state of the s	Fraction	∋nqth	
'cns_wrap	PPO2_SNS	Input 🗸		Inher	it v	N/A Cancel		Update
'cns_wrap Functio imulink and M osimulation of ardware signa	De Block Par odelSim Cosimulation hardware componen Is. Specify signal pat	Input	HDL Cosimu nputs from Simulink(Fical name in ModelSi	Ilation	it OK odelSim sign	N/A Cancel	Help	Update
'cns_wrap Functio imulink and M osimulation of ardware signa Ports Clock	De Block Par De Block Par odelSim Cosimulation hardware componen ls. Specify signal patt ks Timescales	Input	HDL Cosimu nputs from Simulink(Fical name in ModelSi	Ilation	It OK 	N/A Cancel	Help	Update
'ons_wrap Functio imulink and M iosimulation of ardware signa Ports Clock Full HDL	on Block Par odelSim Cosimulation hardware componen Is. Specify signal patt ks Timescales Name	Input rameters: V its with ModelSim(R). I hs by their full hierarch Connection Tcl Edge	HDL Cosimu nputs from Simulink(F ical name in ModelSi	Ilation	OK	N/A Cancel	Help	Update
'Cns_wrap Functio imulink and M iosimulation of ardware signa Ports Clock Full HDL /Cns_wraj	on Block Par odelSim Cosimulation hardware componen is. Specify signal patt ks Timescales Name p/S_CLK	Input	HDL Cosimu HDL Cosimu nputs from Simulink(F ical name in ModelSi Peri ing 2	Ilation	odelSim sign	N/A Cancel al. Outputs from the New Delete Up Down	is block are de	Update Apply srived from

Function	Block Pa	rameters	: VH	D	L		¢	С	c		9	¢	0	D	s	si	11		n		u		la	a	t	i	0	g	n																																								l	>	<	
Simulink and Model	Sim Cosimulatio	0												-												_			_														-	-	-																			-	-	-	-				-	
Cosimulation of hard hardware signals. S	lware componer pecify signal pat	its with ModelSim hs by their full hie	(R). Inp rarchic	al na	s fro nam	ror me	om je	m e ii	n S ir	s	s ı	s 1	si t	in N	mi Mo	loc	lin de	nk el	k(IS	(F Sii	R	() n.	ar	re	a	p	P	lie	ec		to) a	a M	40	00	le	Si	m	si	g	na	al.	C).	ut	pu	uts	fr	om	h t	his	; Ы	lo	c	ĸ	a	re	d	eri	riv	/e	вс	11	fre	or	'n						
Ports Clocks	Timescales	Connection	Tel																																																																					
1 second in Simulink	. corresponds to	1	Tick	~	in in	in t	i th	th	he	e	э	•	> I	F	н	DI	L	s	sii	im	nu	ul	lat	to	r																																															
																																		C			2	<u>ι</u> κ					0			5)a	he	el		כ	C			Ħ	le	۱p)					ł	Ap	p	yly				
Function	Block Pa	rameters	: VH	D	L	2	0	С	C	2	•	6	0	0	s	51	Ь	IJ	n	l	U	J	la	a	t	t	0	ŋ	n	k																																								5	Ş	1
Simulink and Mode	ISim Cosimulatio	n																																																																						
Cosimulation of hard hardware signals. S	lware componer pecify signal pat	nts with ModelSim hs by their full hie	ı(R). Inp rarchic	outs al na	s fro nam	ror me	om ie	m e i	u: ir	e n	s n	ร า	51	in N	mi Me	nuli loc	lin de	nk eľ	k((F Si	R) n.	ar	re	a	p	p	lie	e	ł	to) a	4 f	4c	50	le	Si	n	si	gi	na	al.	C).	utj	ρι	uts	fr	nc	n ti	his	ы	lo	cł	k	ar	re	d	eri	iv	/e	ed	11	frc	оп	n						
Ports Clocks	Timescales	Connection	Tcl																																																																					
🗹 the HDL simulat	or is running on	this computer																																																																						
Connection method	Socket	71						-				_		_				-				-																					_																	-		-	-	_	_	_		1	v			
Host name: pi-5289	93dd2e7e								Ī			Ī	Ī					Ī																				_						Ī	Ī	Ī													T			Ī	Ī							1		
Port number or serv	ice: 4449			_		-	-								-	-	-		-	-																												-	-	-	-	-	-				-	-	-		-			-	-	-			-	1		

<u>0</u>K

Cancel

Help

Apply

Show connection info on icon

Connection Mode

Full Simulation

🚫 No Connection

O Confirm Interface Only

🗑 Function Block Parameters: VHDL Cosimulation 🛛 🛛 🔀
Simulink and ModelSim Cosimulation
Cosimulation of hardware components with ModelSim(R). Inputs from Simulink(R) are applied to a ModelSim signal. Outputs from this block are derived from hardware signals. Specify signal paths by their full hierarchical name in ModelSim.
Ports Clocks Timescales Connection Tcl
Pre-simulation commands:
echo "Running Simulink Cosimulation block."
Post-simulation commands:
echo "Done"
<u>OK</u> <u>Cancel</u> <u>H</u> elp <u>Apply</u>

Рис.6. Параметры функционального блока.

7. Инсталлируя ModelSim в Командном окне МатЛАБ для его совместной работы с Simulink выполните следующий ввод.

```
>> setupmodelsim
...
Do you want setupmodelsim to locate installed ModelSim executables [y]/n? y
Bыберите "y"
Select a ModelSim installation:
[1] C:\Modeltech_6.0c\win32 ModelSim SE 6.0c
[0] None
Bыберите "1"
Previous MATLAB startup file found in this installation of ModelSim:
C:\Modeltech_6.0c\win32\..\tcl\ModelSimTclFunctionsForMATLAB.tcl
Do you want to replace this file [y]/n? y
Bыберите "y"
```

- 8. Запустите ModelSim командой МатЛАБ:
 >> vsim('tclstart', 'do plot.do','socketsimulink', 4449)
- 9. Запустите mdl модель Simulink.
- 10. Рассмотрите специфику взаимодействия модели Simulink и Verilog модуля.
- 11. Добавьте канал передачи данных в общую модель

- 12. Постройте осциллограмму передачи данных
- Примечание: После редактирования verilog файла, откомпилируйте его и затем выполните команду: wrapverilog <имя отредактированного файла>

Задание 2. Построение Simulnik модели управления клапаном с использованием модуля описания аппаратуры

Постройте модель управления клапаном в среде Simulink как показано на рисунке Рис.
 7.



Рис. 7. Пример модели verilog – simulink

2. Откомпилируйте следующий Verilog файл.

Verilog HDL script

//-// Design unit : VALVE CONTROL // Filename: VLV_SC.v \parallel Description: valve solenoid control // // // Limitations: \parallel // Author: BD \parallel Simulator: ModelSim 6.0c // // Version 1.1 by 3 October 2005, original release //----- // timescale 100ms / 10ms

module VLV_SC(SYS_CLK, DEPTH, NRESET, PPO2_SENS, SYNC_CLK, O2_FLOW);

input SYS_CLK, NRESET; input [15:0] DEPTH; // 16-bit DAC. Depth resolution = 1 m input [7:0] PPO2_SENS; // 8-bit ADC of PPO2 sensor

output SYNC_CLK, O2_FLOW;

reg O2_FLOW; // solenoid valve injector control reg [7:0] PPO2_GOAL; // 8-bit PPO2 set point (code 255 means PPO2 is 1.6) // code 112 means PPO2 is 0.7 // code 191 means PPO2 is 1.2

assign SYNC_CLK = SYS_CLK; // control of the MatLAB clock

```
// Real time clock is 1 s
always @SYS_CLK
                                    // Simulink lowest clock period is 2 s
 if (!NRESET)
   begin
        (DEPTH < 10) // Depth in m
PPO2_GOAL = 112; // 0.7 of PPO2
      if (DEPTH < 10)
      else
        PPO2 GOAL = 191; // 1.2 of PPO2
      if (PPO2_SENS < PPO2_GOAL)
        O2_FLOW = 1;
       else
        O2 FLOW = 0;
   end
 else
         O2 FLOW = 0;
                           //solenoid is off
```

endmodule

3. Создайте оболочку Verilog файла

VHDL wrap

--

The Simulink connects the verilog compiled data via the VHDL wrap.

The ModelSim command: wrapverilog verilog_file_name.v runs generation of the VHDL file which name is verilog_file_name_wrap.vhd

The listing of the used VHDL file is following.

-- Module vlv_sc VHDL Wrapper

-- Generated by The MathWorks wrapverilog tcl command

-- Generated on: 2005-10-03 09:34:16

LIBRARY IEEE; USE IEEE.std_logic_1164.all; ENTITY vlv_sc_wrap IS port(SYS CLK : in std logic; DEPTH : in std_logic_vector(15 downto 0); NRESET : in std_logic; PPO2 SENS : in std_logic_vector(7 downto 0); SYNC_CLK : out std_logic; O2_FLOW : out std_logic); END vlv_sc_wrap; ARCHITECTURE rtl OF vlv_sc_wrap IS component VLV_SC port(SYS_CLK : in std_logic; DEPTH : in std_logic_vector(15 downto 0); NRESET : in std_logic; PPO2_SENS : in std_logic_vector(7 downto 0); SYNC_CLK : out std_logic; O2_FLOW : out std_logic); end component; FOR ALL : vlv_sc USE ENTITY work.vlv_sc(ignored); BEGIN u_vlv_sc: vlv_sc PORT MAP (SYS_CLK => SYS_CLK, DEPTH => DEPTH, NRESET => NRESET , PPO2_SENS => PPO2_SENS , SYNC CLK => SYNC CLK, O2 FLOW => O2 FLOW);

END rtl;

4. Настройте параметры моделирования.

- 5. Запустите модель.
- 6. Убедитесь в работоспособности модели.

контрольные вопросы

- 1. Для чего предназначен язык описания аппаратных средств Verilog?
- 2. Для чего предназначен блок "VHDL Cosimulation" mdl модели?
- 3. Какие задачи можно решать при помощи объединенной модели Verilog Simulink?

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- 1. Обзор современных языков описания аппаратуры <u>http://systemc.dax.ru/book/1.html</u>
- 2. Введение в Verilog http://www.marsohod.org/index.php/ourblog/11/77-veriloglesson2
- 3. Краткий курс HDL. Описание языка Verilog. <u>http://www.kit-e.ru/articles/circuit/2008_5_154.php</u>
- 4. Dr. Bob Davidov. Компьютерные технологии управления в технических системах <u>http://portalnp.ru/author/bobdavidov</u>