

基于 FPGA 的多功能波形发生器 设计课程设计

武汉理工大学

毕业设计

题目 多功能波形发生器
的设计

学院 信息工程学院

专业

班级

姓名

指导教师

年 月 日

<u>摘要</u>	4
<u>Abstract</u>	6
<u>多功能波形发生器的设计</u>	6
<u>1. 课题简介</u>	8
<u>1.1 课设目的</u>	8
<u>1.2 课设要求</u>	8
<u>2 设计方案</u>	10
<u>2.1 方案选择</u>	10
<u>2.2 设计原理</u>	10
<u>2.3 设计流程</u>	11
<u>3 仿真结果</u>	17
<u>3.1 编译警告</u>	17
<u>3.2 编译结果</u>	20
<u>3.3 建立仿真文件</u>	20
<u>3.4 仿真结果</u>	22
<u>3.5 RTL 视图</u>	23
<u>4 程序分析</u>	25
<u>4.1 VHDL 语言分析</u>	25
<u>4.2 主要函数语句分析</u>	26

5 小结27

6 参考文献. 28.

7 附录源程序代码. 30.

摘要

多功能信号发生器已成为现代测试领域应用最为广泛的通用仪器之一，代表了信号源的发展方向。直接数字频率合成(DDS)是二十世纪七十年代初提出的一种全数字的频率合成技术，其查表合成波形的方法可以满足产生任意波形的要求。由于现场可编程门阵列(FPGA)具有高集成度、高速度、可实现大容量存储器功能的特性，能有效地实现 DDS 技术，极大的提高函数发生器的性能，降低生产成本。

本文首先介绍了函数信号发生器的研究背景和 DDS 的理论。然后详尽地叙述了利用 VHDL 语言描述 DDS 模块的设计过程,以及设计过程中应注意的问题。文中详细地介绍了多种信号的发生理论、实现方法、实现过程、部分 VHDL 代码以及利用 Quartus 仿真的结果。

文中还介绍了 Altera公司的 DE2 多媒体开发平台的部分功能及使用，并最终利用 DE2 平台完成了多功能信号发生器的大部分功能。包括由 LCD 显示和按键输入构成的人机界面和多种信号的发生。数字模拟转换器是 BURR-BROWN 公司生产的 DAC902 。

该信号发生器能输出 8 种不同的信号，并且能对输出信号的频率、相位以及调制信号的频率进行修改设定。

关键词：VHDL D/A 接口



Abstract

Multi function signal generator has become the most widely used in modern testing field of general instrument, and has represented one of the development direction of the source. Direct digital frequency synthesis (DDS) is a totally digital frequency synthesis technology, which been put forward in the early 1970s. Using a look-up table method to synthetic waveform, it can satisfy any requirement of waveform produce. Due to the field programmable gates array (FPGA) with high integrity, high speed, and large storage properties, it can realize the DDS technology effectively, increase signal generator's performance and reduce production costs.

Firstly, this article introduced the function signal generator of the research background and DDS theory. Then, it described how to design a DDS module by VHDL, and introduced various signal occurs theory, method and the implementation process, VHDL code and simulation results.

This paper also introduce the function of DE2 multimedia development platform, and completed most of the functions of multi-function signal generator on DE2 platform finally. Including the occurrence of multiple signal and the man-machine interface which composed by LCD display and key input. Digital-to-analog converters is DAC902, which produced by company BURR-BROWN.

This signal generator can output eight different kinds of signals, and the frequency of the output signal, phase and modulation frequency signal also can be modified. Keywords: VHDL D/A Interface

多功能波形发生器的设计

信号发生器又称信号源或振荡器，在生产实践和科技领域中有着广泛的应

用。它能够产生多种波形，如三角波、锯齿波、矩形波（含方波）、正弦波等，在电路实验和设备检测中具有十分广泛的用途。例如在通信、广播、电视系统中，都需要射频（高频）发射，这里的射频波就是载波，把音频（低频）、视频信号或脉冲信号运载出去，就需要能够产生高频的振荡器。在工业、农业、生物医学等领域内，如高频感应加热、熔炼、淬火、超声诊断、核磁共振成像等，都需要功率或大或小、频率或高或低的信号发生器。

传统的信号发生器主要有两类：正弦波和脉冲波信号发生器，而函数发生器介于两类之间。它能够提供正弦波、锯齿波、方波、三角波等几种常用标准波形，产生其它波形时，需要采用较复杂的电路和机电结合的方法。这个时期的波形发生器多采用模拟电子技术，但是模拟器件构成的电路存在着尺寸大、价格贵、功耗大等缺点。而且要产生的信号波形越复杂，则电路结构也会越复杂。同时还有两个突出问题，一是通过电位器的调节来实现输出频率的调节，因此很难将频率调到某一固定值；二是脉冲的占空比不可调节。

现代科学技术的飞速发展对信号源提出了越来越高的要求。这些要求主要表现在高分辨率、高输出频率、任意波形等方面。此时传统的信号发生器已经无法满足要求。

直接数字频率合成（Direct Digital Synthesis，简称 DDS）技术是一种新的全数字的频率合成原理，它从相位的角度出发直接合成所需波形。这种技术由美国学者 J. Tiercy, M. Rader 和 B. Gold 于 1971 年首次提出，但限于当时的技术和工艺水平，DDS 技术仅仅在理论上进行了一些探讨，而没有应用到实际中去。但是随着电子技术的飞速发展，微处理器性能大幅提高，高速的 D/A 以及随机存储器大量涌现，DDS 技术已经变得很容易实现。它已广泛应用于通讯、雷达、遥控测试、电子对抗以及现代化的仪器仪表工业等许多领域。将其与简单电路相结合就可以精确模拟仿真各种信号。

不论是在生产、实验还是在科研与教学上，多功能信号发生器都是用于仿真实验的最佳工具。随着我国经济和科技的发展，对相应的测试仪器和测试手段也提出了更高的要求，多功能信号生器已成为测试仪器中至关重要的一类，因此开发多功能信号发生器具有重大意义。

1. 课题简介

1.1 课设目的

- (1) 懂得多功能波形发生器的结构组成
- (2) 懂得利用 FPGA 芯片实现多种波形的产生方法
- (3) 懂得一种复杂 FPGA 电路的设计

1.2 课设要求

设计一个多功能波形发生器。该波形发生器能产生正弦波、方波、三角波和由用户编辑的特定形状波形。具体要求如下：

- (1) 具有产生正弦波、方波、三角波、锯齿波 4 种周期性波形的功能。
- (2) 用键盘输入编辑生成上述 4 种波形（同周期）的线性组合波形。
- (3) 具有波形存储功能。
- (4) 输出波形的频率范围为 100Hz ~ 200kHz；重复频率可调，频率步进间隔 $\leq 100\text{Hz}$ 。
- (5) 输出波形幅度范围 0~5V（峰-峰值），可按步进 0.1V（峰-峰值）调整。
- (6) 具有显示输出波形的类型、重复频率（周期）和幅度的功能。
- (7) 用键盘或其他输入装置产生任意波形。多功能波形发生器系统由以下四部分组成：输入部分、FPGA 部分、DAC、显示部分组成。

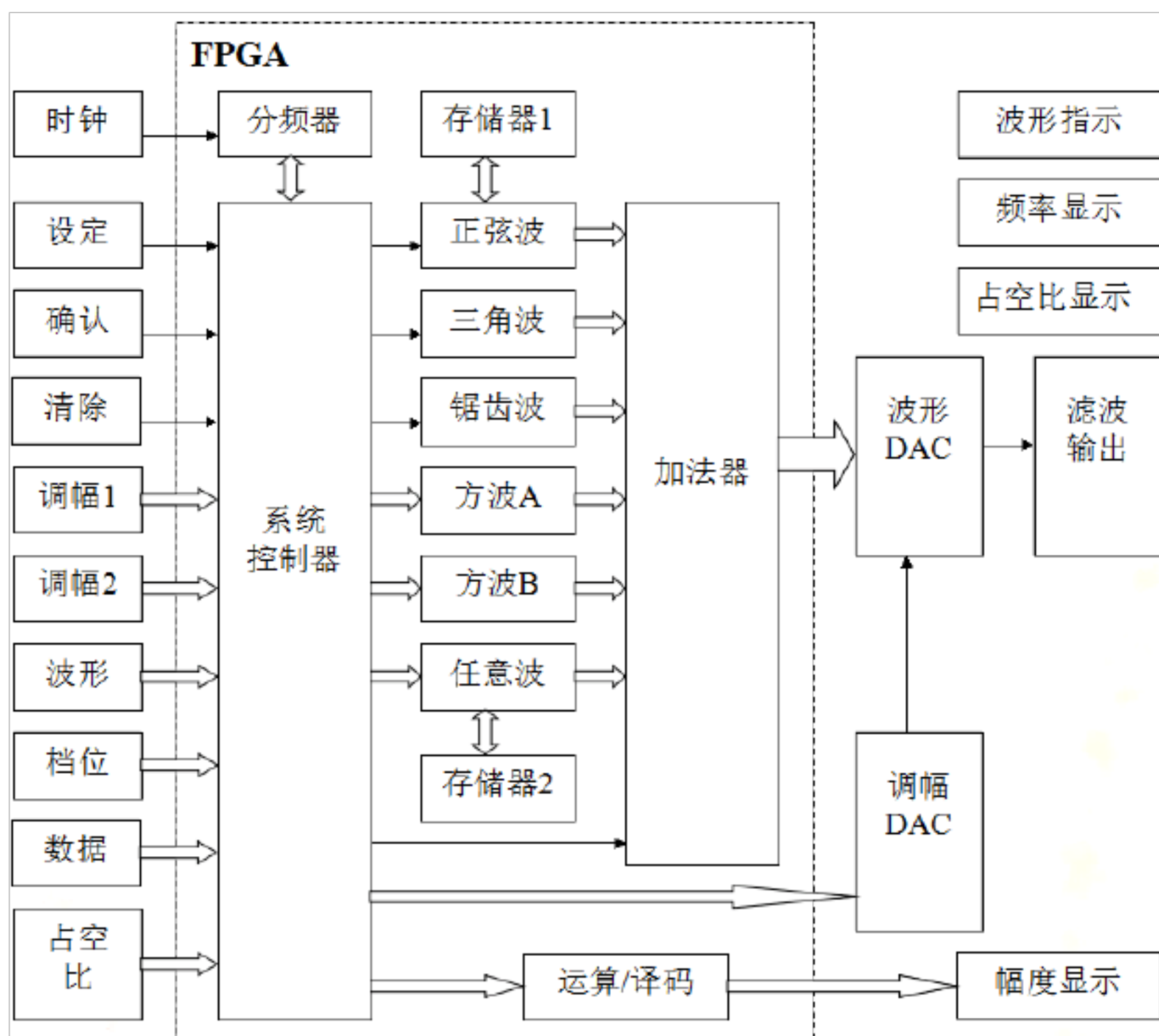


图 1 多功能波形发生器

系统框图

1.3课设工具

本次设计是基于 Altera公司的 QuartusII 软件。

Quartus II是 Altera公司的综合性 PLD 开发软件，支持原理图、VHDL 、VerilogHDL 以及 AHDL （Altera Hardware Description Language等多种设计输入形式，内嵌自有的综合器以及仿真器，可以完成从设计输入到硬件配置的完整 PLD 设计流程。

QuartusII支持 Altera的 IP 核, 包含了 LPM/MegaFuction 宏功能模块库, 使用它, 可以简化设计的复杂性, 加快设计速度。QuartusII平台支持一个工作组环境下的设计要求, 其中包括支持基于 Internet的协作设计。此外, QuartusII 通过和 DSP Builder工具与 Matlab/SIMULINK 相结合, 可以方便地实现各种 DSP 应用系统; 支持 Altera的片上可编程系统开发, 集系统级设计、嵌入式软件设计、可编程逻辑设计于一体, 是一种综性的开发平台。

2 设计方案

2.1 方案选择

1、模拟锁相环实现

模拟锁相环技术是一项比较成熟的技术。应用模拟锁相环, 可将基准频率倍频, 或分频得到所需的频率, 且调节精度可以做到相当高、稳定性也比较好。但模拟锁相环模拟电路复杂, 不易调节, 成本较高, 且由于受模拟器件的影响, 波形变换调节时间较长, 输出波形的毛刺较多, 因此模拟锁相环实现在低频 ($0\sim 500\text{KHz}$) 信号发生系统中不是很好的方案。

2、直接数字频率合成实现

直接数字频率合成 (DDFS) 技术是经典的数字频率合成技术。由于数字量的可操作性远远高于模拟量, 采用 DDFS 的优点在于频率精度高、波形调节方便、且输出波形毛刺少等。基于单片机的数字波形发生系统受单片机指令频率的限制, 输出波形频率较低, 而基于 FPGA 的波形发生系统就不存在这样的问题, 其输出频带较单片机实现有很大的展宽。本系统设计选定以 FPGA 作为系统控制核心的直接数字频率合成实现方案。

2.2 设计原理

采用 DDS 技术可以很方便地产生各种高质量的波形。DDS 技术是从相位概念出发之结合成所需要波形的一种频率合成技术。以正弦波为例, 首先要按照一定的采样点数将正弦波形一个周期的数据信息存于 ROM 表中, 表中包含着一个周期正弦波的数字幅度信息, 每个地址对应正弦波中 0 到 360 度范围内的一个相位点的幅度值, 查找表时即是把输入的地址相位信息映射成正弦波幅度的数字量信号, 通过设置的输出端口输出。在实物设计中, 可以使用 D/A 接口来实现波形信号的输出。

为简化设计过程，本设计并未采用 DDS 技术，而是采用描点输出的方式，实现波形发生器的设计。程序中设置一个波形的起始点，经过比较、计算得出波形的其他数值，将这些点依次连续输出，从而实现波形的仿真。以递增锯齿波为例，首先定义初始点为 tmp=“00000000”；在时钟上升沿到来时，执行 tmp<=tmp+1；语句，同时将 tmp 输出，当 tmp=“11111111”；时，将 tmp 值清零，执行下一个循环。

在本设计中，采用 QuartusII 软件仿真，所以可以通过波形文件直观的反映出输出的数字量的变化情况，以达到波形输出的仿真。

正弦波：通过循环不断地从 RAM 中依次读取正弦波一个周期在时域上 64 个采样点的波形数据送入波形 DAC，从而产生正弦波。正弦波的频率取决于读取数据的速度。

任意波：首先通过键盘把任意波形数据存储在存储器中。然后循环不断地从存储器 2 中依次读取任意波一个周期在时域上 64 个采样点的波形数据送入波形 DAC，从而产生任意波。任意波的频率取决于读取数据的速度。

三角波：三角波波形是对称的，每边呈线形变化，所以可以根据地址数据做简单运算，就可以得到三角波

锯齿波：产生单调性锯齿波，因此把地址数据进行左移 2 位，结果送波形 DAC 就可。

方波 A：方波 A 产生也是由 64 个采样点组成，64 个采样点的数据只有“低电平”和“高电平”2 种状态。更改“低电平”和“高电平”出现的比例，可以达到调节占空比的目的。

波形 DAC：根据输入的波形数据（即 FPGA 输出的数据），产生相应的模拟波形的输出。调幅 DAC：根据输入的幅度调节数据（即 FPGA 输出的数据），用来调节波形 DAC 的基准电压，到达输出波形幅度调节的目的。

波形指示：每种波形具有一个选择开关，而每个选择开关与 FPGA 接口的一端都并有一个发光二极管，发光二极管起指示作用。

频率显示：频率可以直接用 4 位 BCD 拨盘开关上的数字和档位开关位置一起来表示频率。

幅度显示：用 4 个数码管来显示输出波形的幅度。

2.3 设计流程

首先启动 Quartus II 软件如图 2 所示：

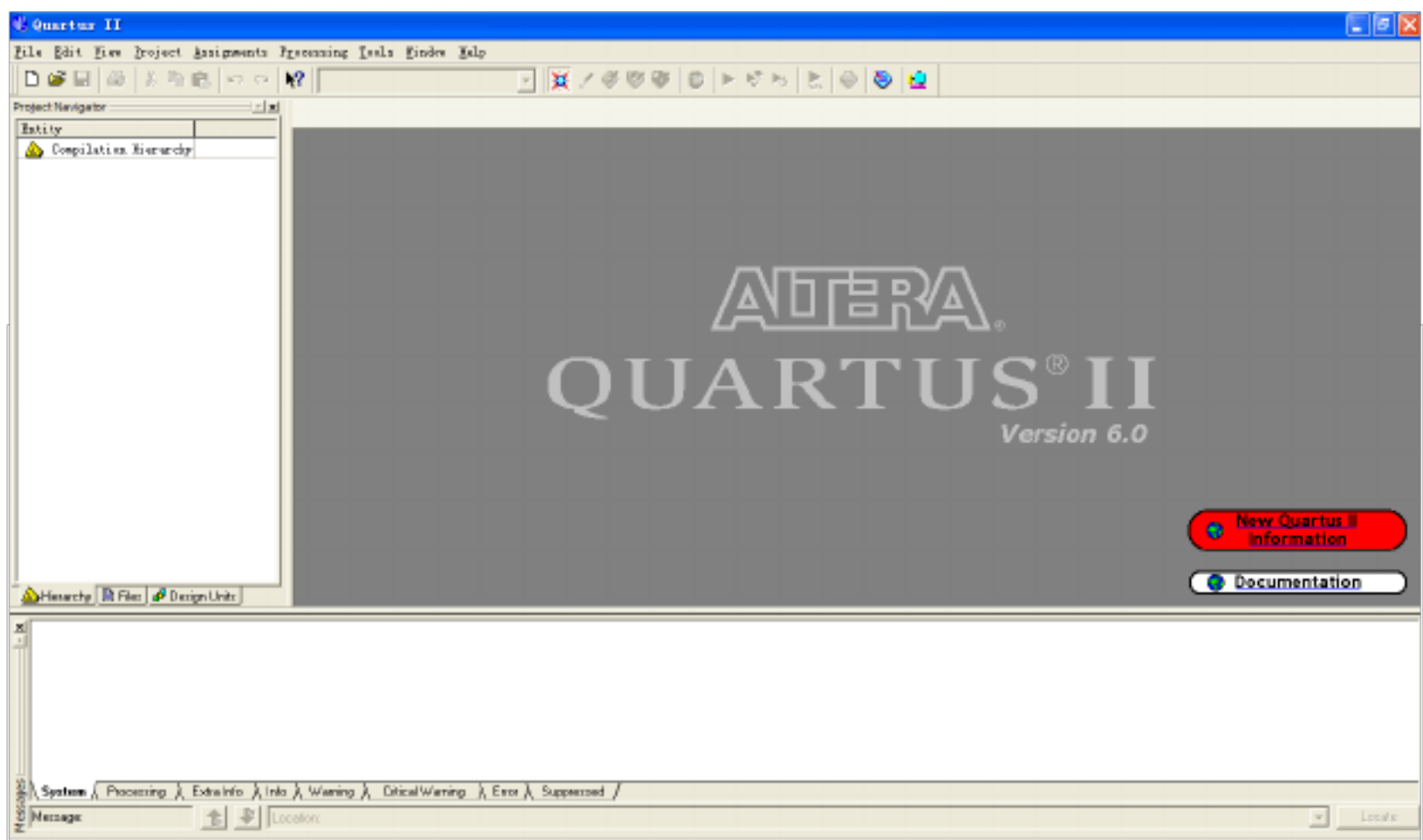


图 2 Quartus II 启动界面

接着利用向导，建立一个新的工程。在 File 菜单中选择 New Project Wizard 选项启动项目向导。如图 2 所示，分别指定创建工程的路径，工程名和顶层文件名。工程名和顶层文件可以一致也可以不同。一个工程中可以有多个文件，但只能有一个顶层文件。这里我将工程名取为：keshe。如图 3 所示。

New Project Wizard: Directory, Name, Top-Level Entity [page 1 of 5]

What is the working directory for this project?

c:\altera\90\quartus ...

What is the name of this project?

keshe ...

What is the name of the top-level design entity for this project? This name is case sensitive and must exactly match the entity name in the design file.

keshe ...

Use Existing Project Settings ...

< Back Next > Finish 取消

图 3 创建工程界面

在图 2 所示界面点击 NEXT 按钮出现对话框如图 3 所示直接点 next 按钮然后在出现的界面中选择芯片出现如图 4 所示界面。点击 File-new 按键, 选择 VHDL 语言, 如图 5 所示。即进入 VHDL 语言编辑区如图 7 所示。

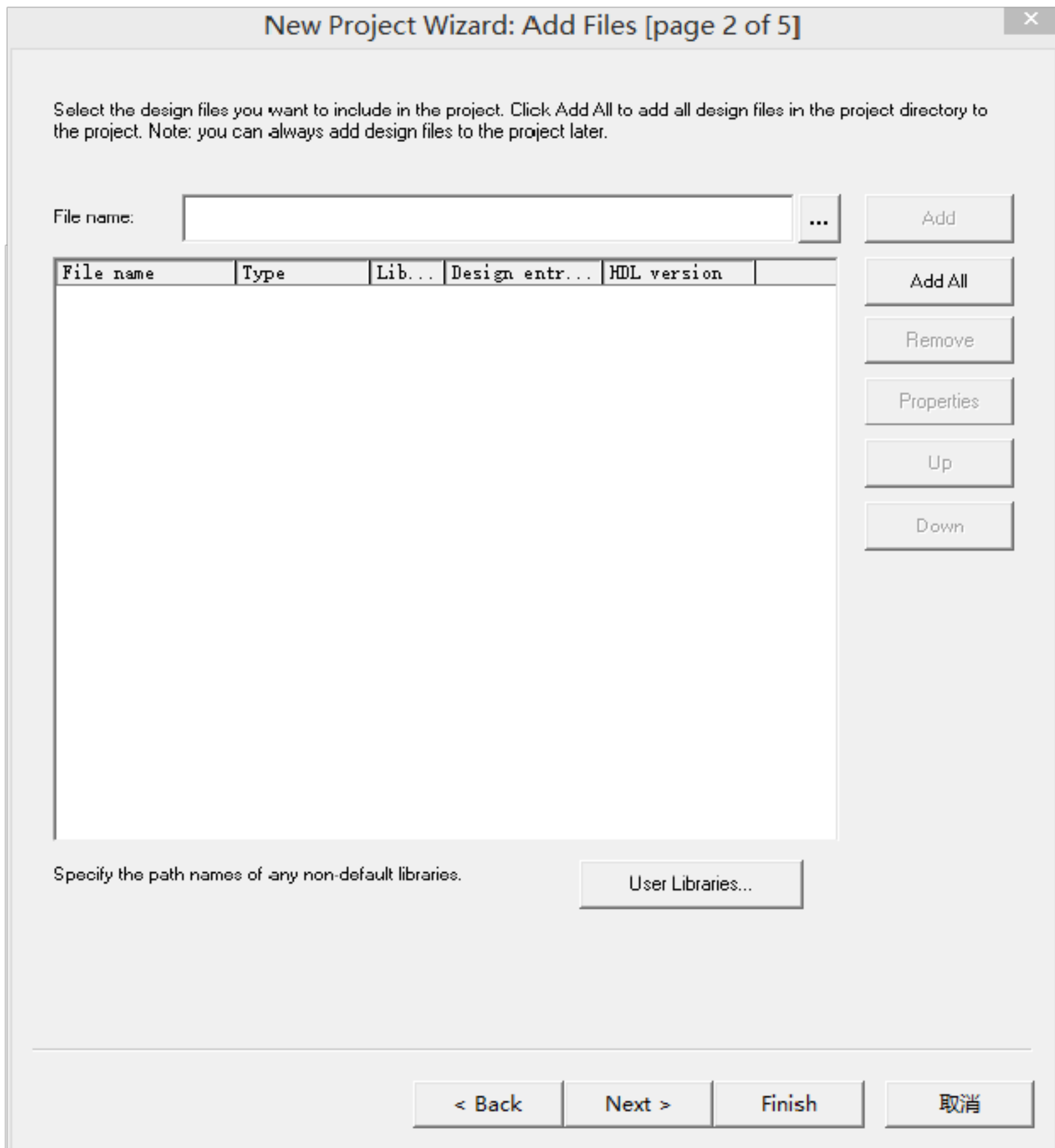


图 4 创建工程的结束界面

New Project Wizard: Family & Device Settings [page 3 of 5]

Select the family and device you want to target for compilation.

Device family

Family: Cyclone II

Devices: All

Show in 'Available device' list

Package: Any

Pin count: 208

Speed grade: Any

☒ Show advanced devices

☐ HardCopy compatible only

Target device

☐ Auto device selected by the Filter

☒ Specific device selected in 'Available devices' list

Available devices:

Name	Core v...	LEs	User I/O...	Memor...	Embed...	PLL	Global ...
EP2C5Q208C7	1.2V	4608	142	119808	26	2	8
EP2C5Q208C8	1.2V	4608	142	119808	26	2	8
EP2C5Q208I8	1.2V	4608	142	119808	26	2	8
EP2C8Q208C7	1.2V	8256	138	165888	36	2	8
EP2C8Q208C8	1.2V	8256	138	165888	36	2	8
EP2C8Q208I8	1.2V	8256	138	165888	36	2	8

Companion device

HardCopy:

☒ Limit DSP & RAM to HardCopy device resources

< Back Next > Finish 取消

图 5 芯片选择

New

SDPC Builder System

- Design Files
 - AHDL File
 - Block Diagram/Schematic File
 - EDIF File
 - State Machine File
 - SystemVerilog HDL File
 - Tcl Script File
 - Verilog HDL File
 - VHDL File
- Memory Files
 - Hexadecimal (Intel-Format) File
 - Memory Initialization File
- Verification/Debugging Files
 - In-System Sources and Probes File
 - Logic Analyzer Interface File
 - SignalTap II Logic Analyzer File
 - Vector Waveform File
- Other Files
 - AHDL Include File
 - Block Symbol File
 - Chain Description File
 - Synopsys Design Constraints File
 - Text File

OK Cancel

图 6 创建 VHDL File 对话框

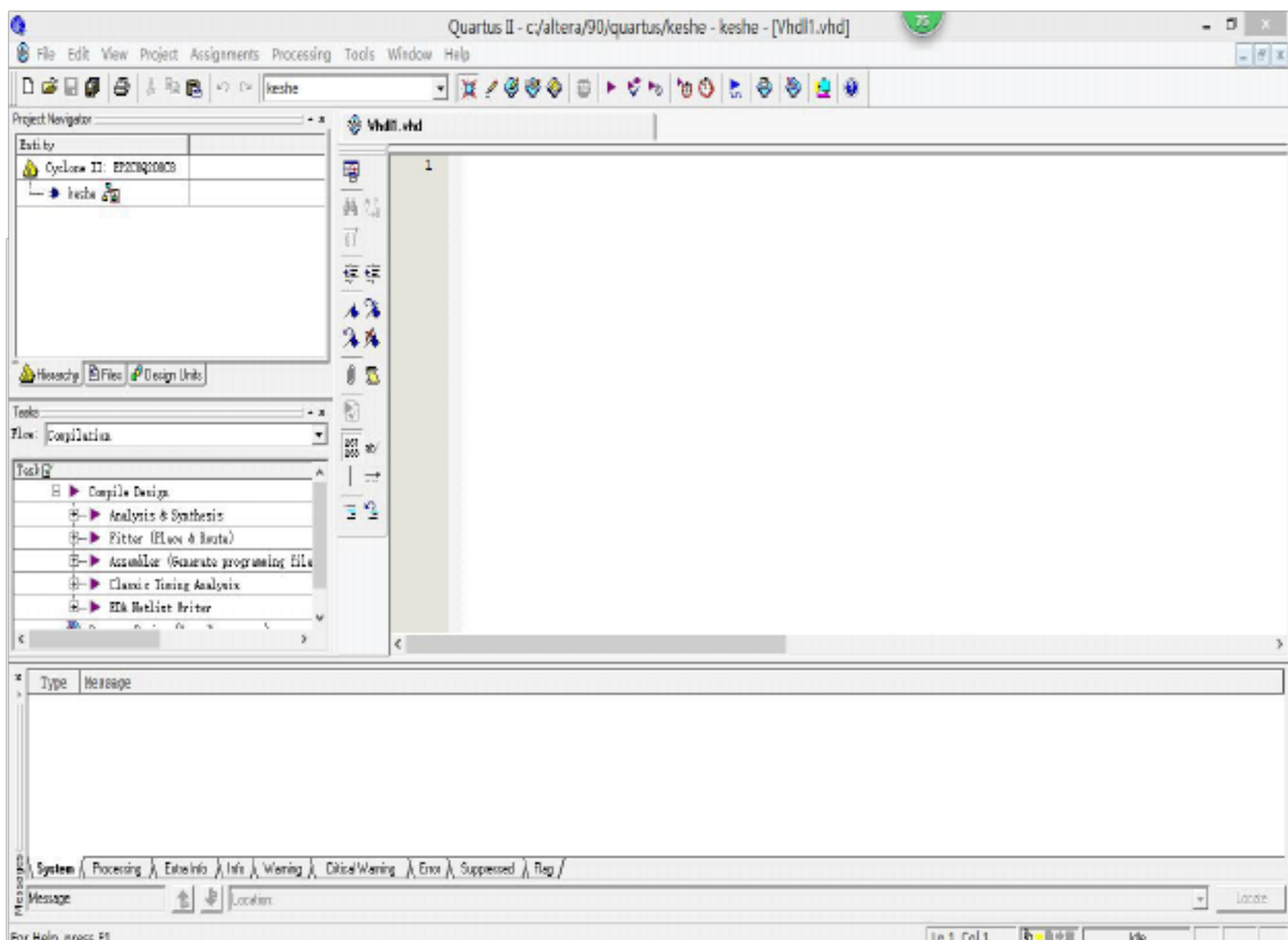


图 7 VHDL 语言编辑框

在图 7 所示的 VHDL 语言编辑框中依次输入分频器、四选一选择器、循环加法计数器等 4 钟计数器、七段译码器等功能模块的 VHDL 语言源程序。输入完成之后单击保存图标并输入相应的文件名。保存之后即可对源程序进行编译。如果编译成功则源程序完全正确，否则应该返回到出错处改正错误直至编译成功为止。编译成功之后再进行波形仿真，从中得出相应数据及现象。

3 仿真结果

3.1 编译警告

输入相应源代码，点击，得出结果如图 8

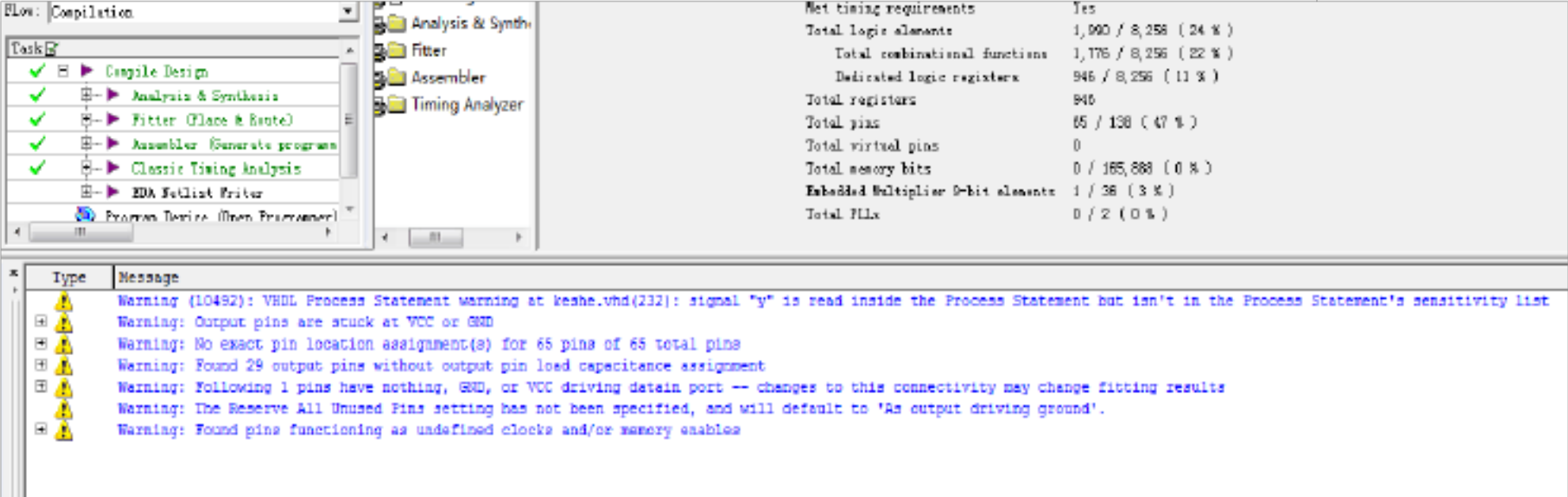


图 8 编译结果

程序在编译阶段出现了八个警告，以下面两个为例：

1. Warning (10492):VHDL Process Statement warning at mine4.vhd(232):signal“y” is read inside the Process Statement but isn’t in the Process Statement’s sensitivity list

2. Warning: Output pins are stuck at VCC or GND

Warning (13410):Pin “p180” is stuck at VCC

第一个警告，在 process里作为被判断信号(i或者 case后面的)或者赋值语句右端信号通常应该写在 process的敏感信号表里。有些 EDA 工具不检查这个，可能会导致仿真结果与综合出来的电路不一致。实际上，综合工具在综合的时候会自动把这类信号添加到敏感信号表里，但仿真工具不会，而是完全按照代码

体现的语意来仿真。

--7

case y is

段码译码

```
when 0 => lcd(7 downto 1) <= "0000001";
when 1 => lcd(7 downto 1) <= "1001111";
when 2 => lcd(7 downto 1) <= "0010010";
when 3 => lcd(7 downto 1) <= "0000110";
when 4 => lcd(7 downto 1) <= "1001100";
when 5 => lcd(7 downto 1) <= "0100100";
when 6 => lcd(7 downto 1) <= "0100000";
when 7 => lcd(7 downto 1) <= "0001111";
when 8 => lcd(7 downto 1) <= "0000000";
when 9 => lcd(7 downto 1) <= "0000100";
when others => lcd(7 downto
1) <= "0000001";
end case;
end process;
```

综合时，信号 y 被添加到敏感信号表中。

第二个警告，认为引脚 p180 直接到正电源了。

```
port (clk          :          in          std_logic;
```

—时钟信号输入

```
set,clr,up, down, zu, zd : in std_logic;
```

—各个波形特征的调节触发信号

```
posting          :          in          std_logic;
```

-任意波键盘置入信号

u0, d0, sw : in std_logic; --方波

A、B 的切换 sw，和方波 B 的幅度调节按钮

```
ss : in std_logic_vector(3 downto 0);
```

-档位选择信号

```
sss : in std_logic_vector(4 downto 0);
```

-波形选择信号

```
Data3, Data2, Data1, Data0 : in  
std_logic_vector(3 downto 0); --四位输入
```

```
p180 : out std_logic;
```

-预留接口

```
lcd : out std_logic_vector(7 downto 0);
```

-显示输出

```
shift: out std_logic_vector(3 downto 0);
```

-位码输出

```
dd, a : out std_logic_vector( 7 downto 0));
```

-波形、幅度数据输出

但实际上引脚 p180 是预留接口未用。

3.2 编译结果

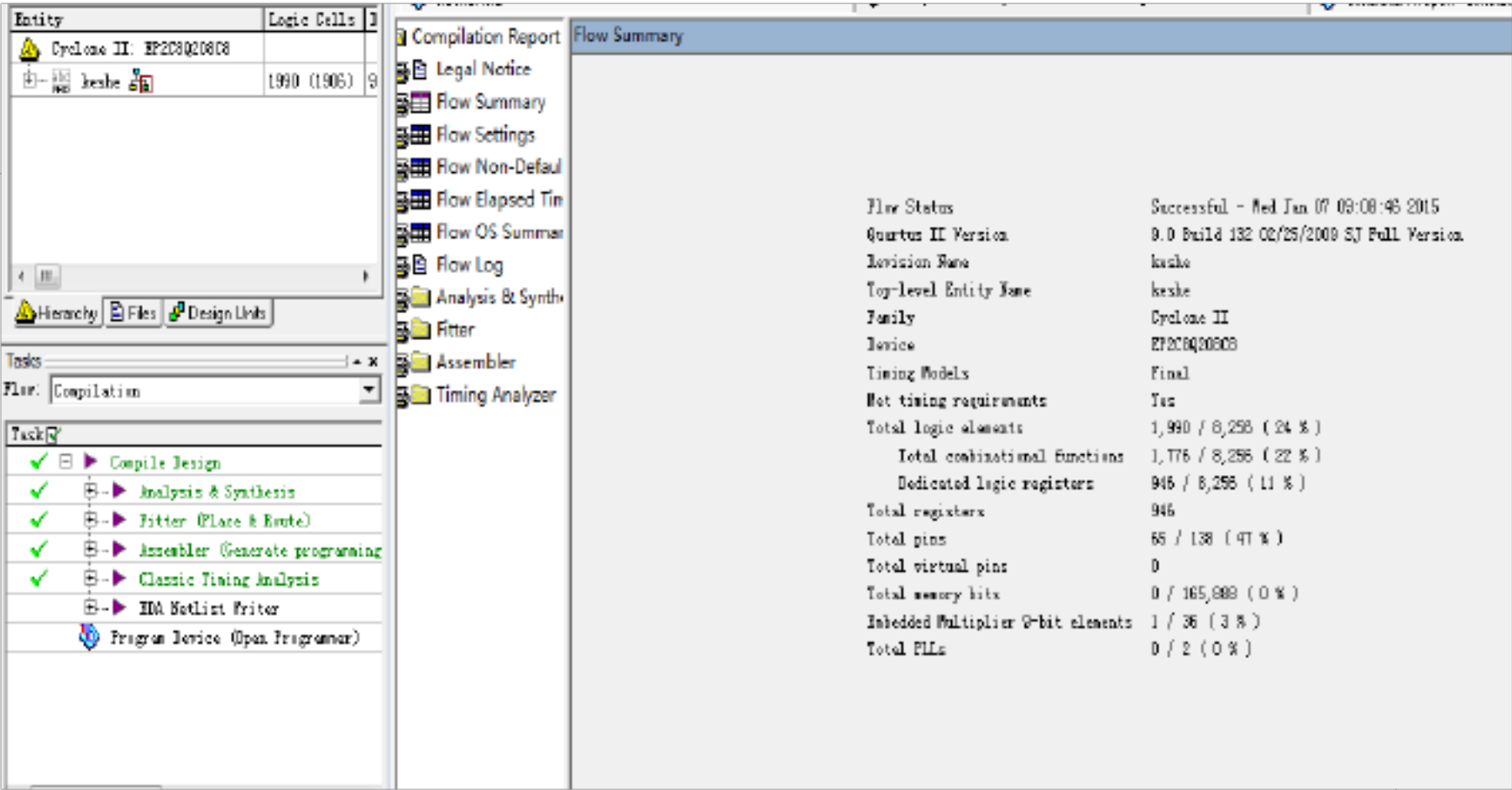


图 9 编译结果

3.3 建立仿真文件

在编译通过后，要建立后缀为 vwf 的仿真波形文件。执行菜单命令，选择 new，再选择需要的 Vector Waveform File 单击 ok。在波形文件编辑方式下，右键选择 insert 添加信号节点，设置相应的参数。如下图所示。

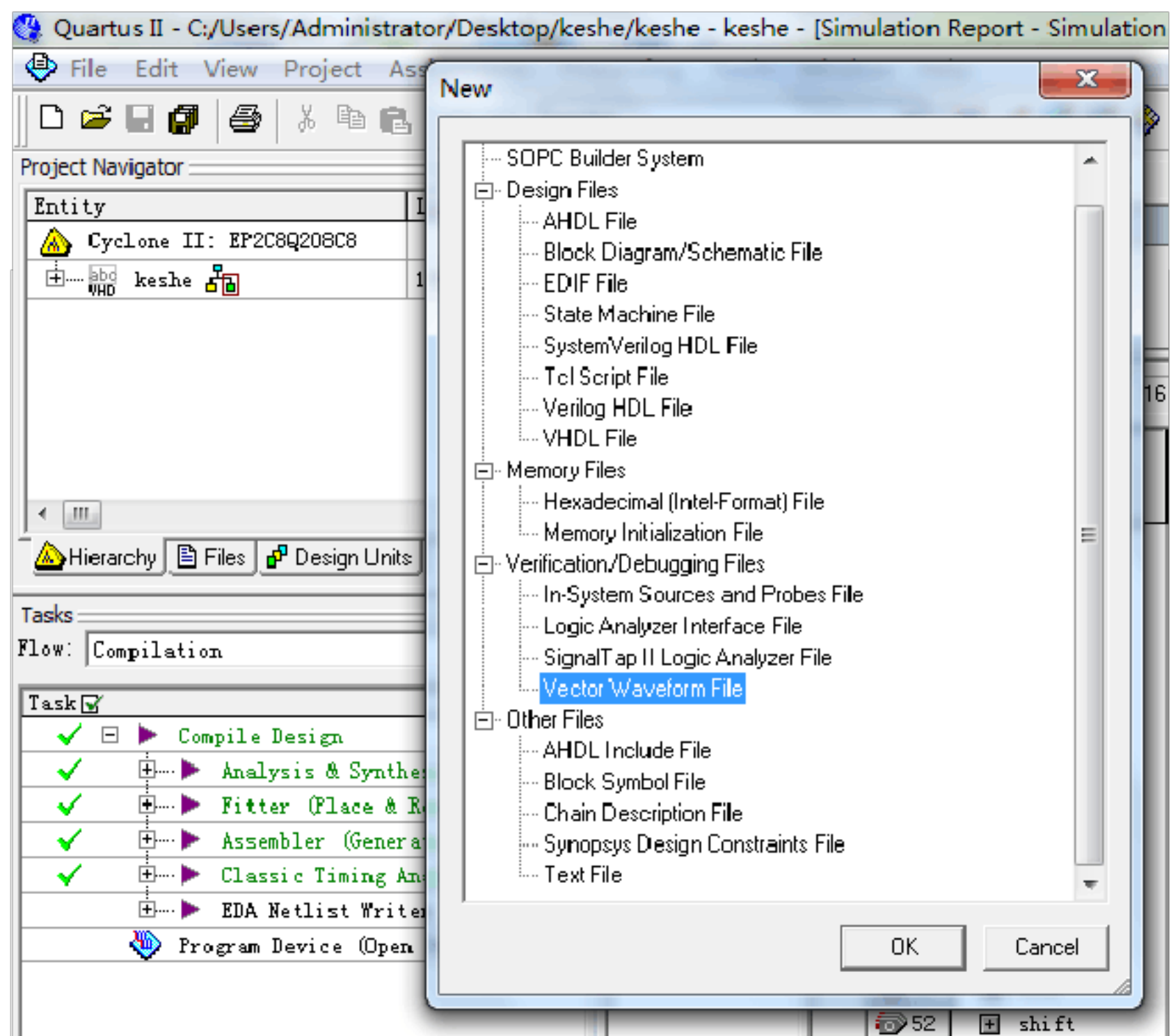


图 10 仿真波形文件建立

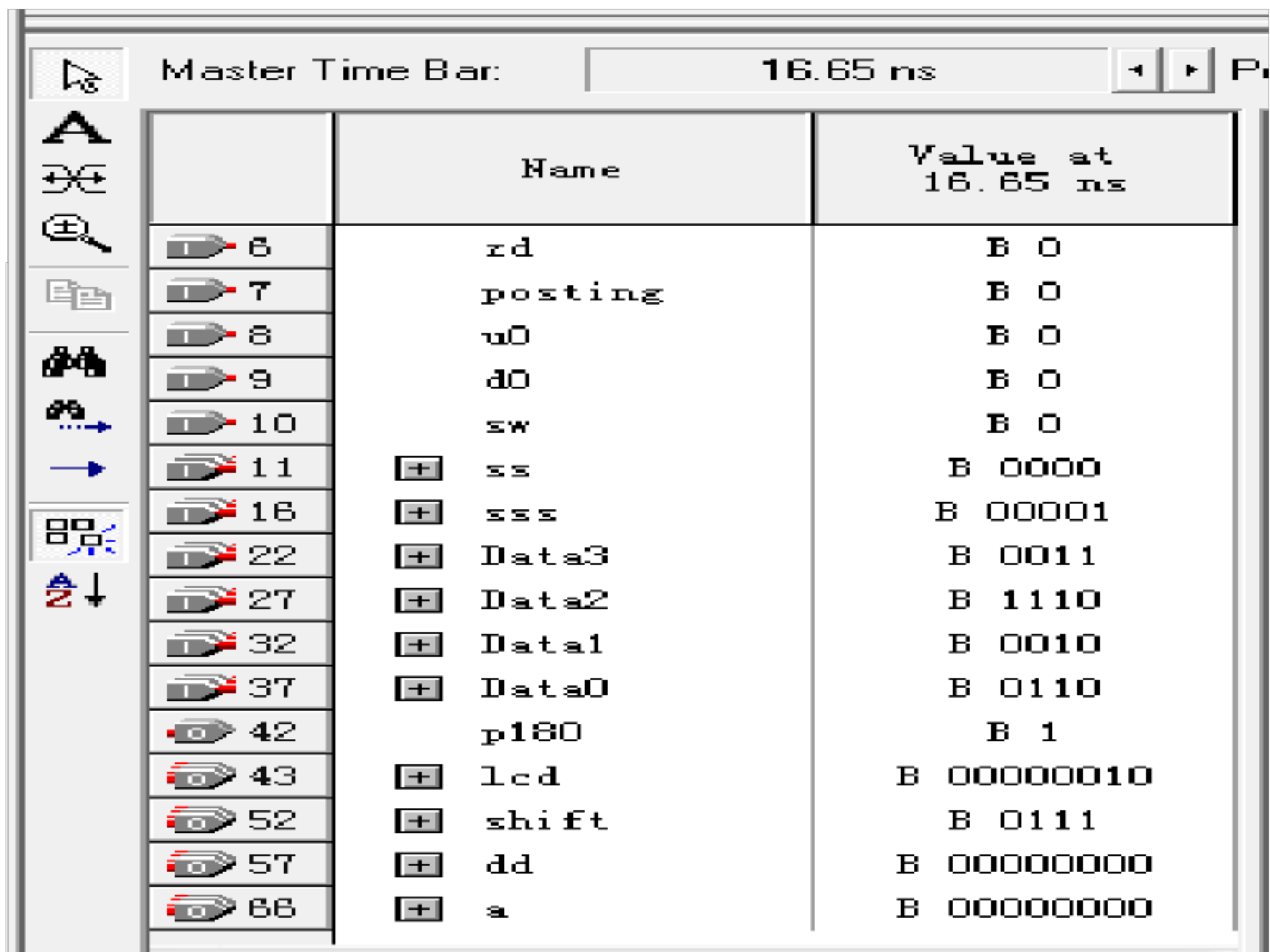
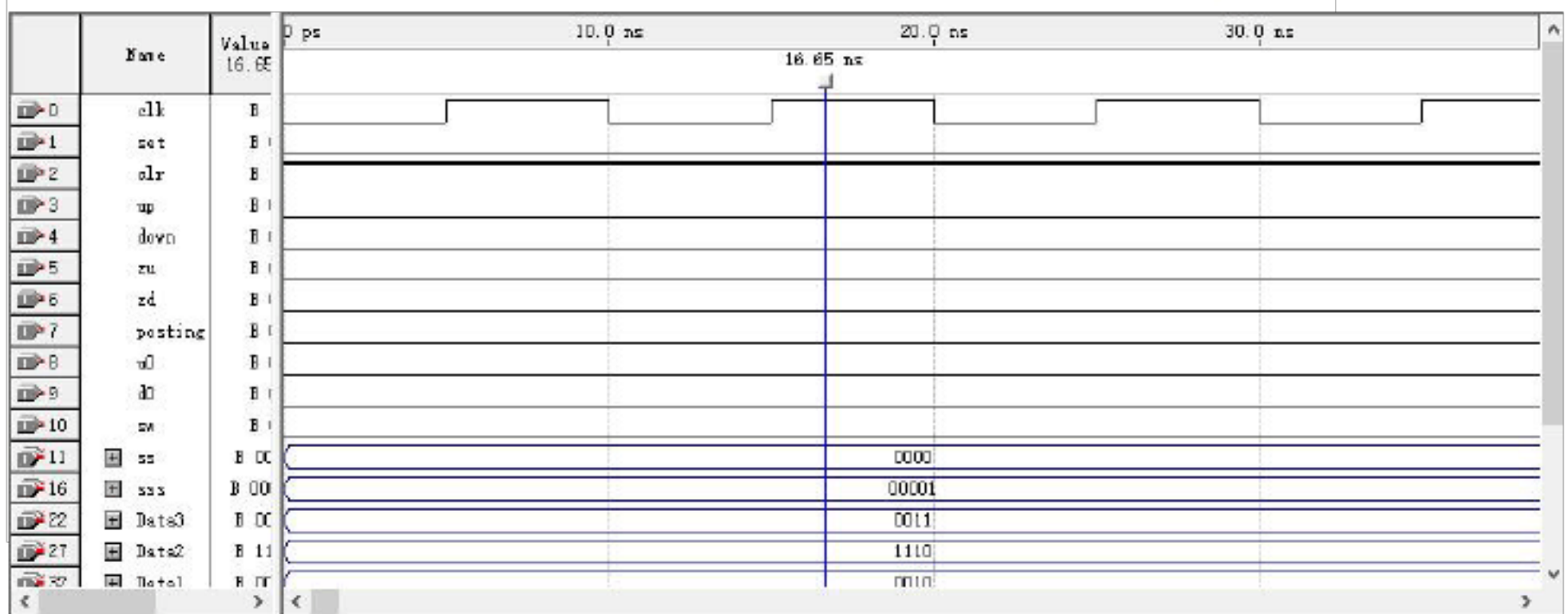


图 11 仿真波形参数设置

3.4 仿真结果

仿真结果如下图所示



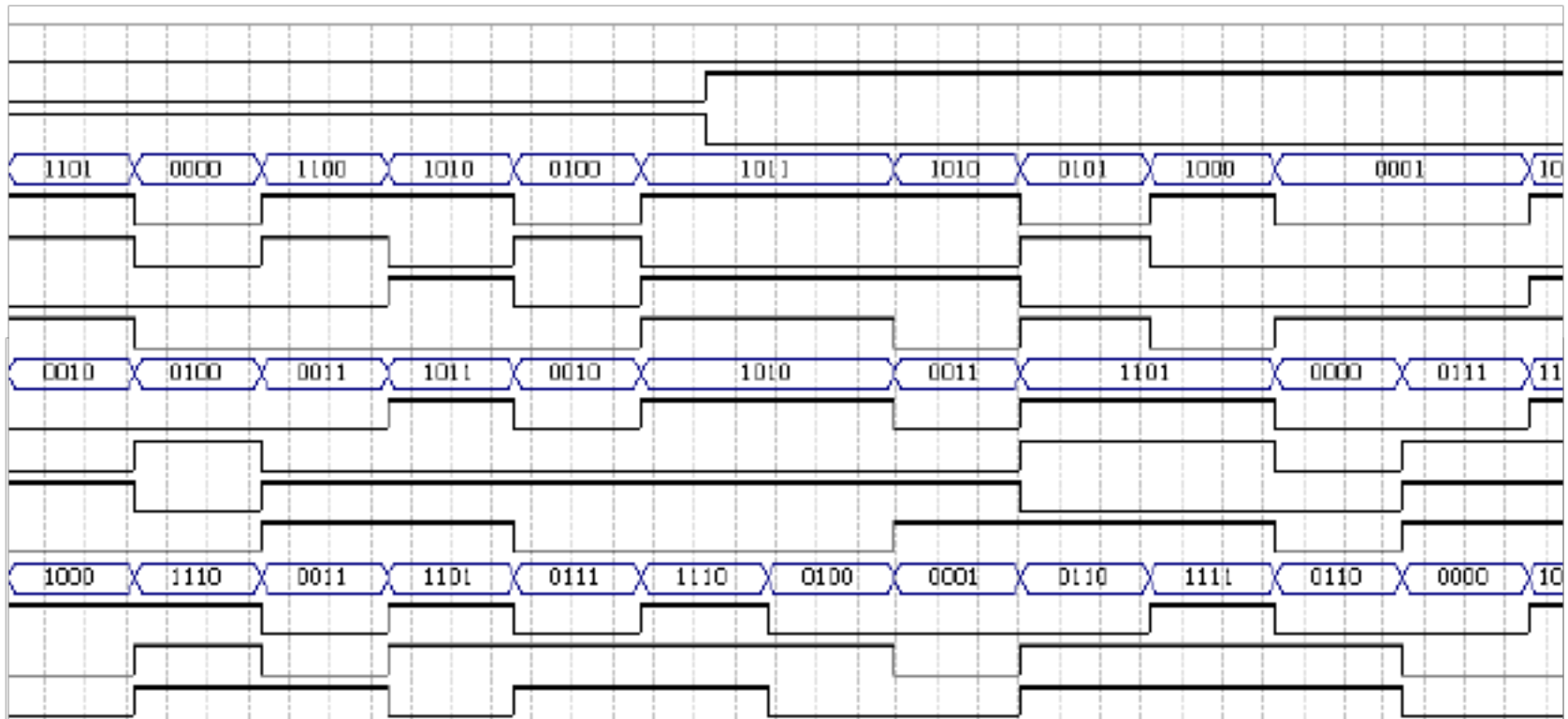


图 12 仿真波形

3.5 RTL 视图

生成的 RTL 视图如下

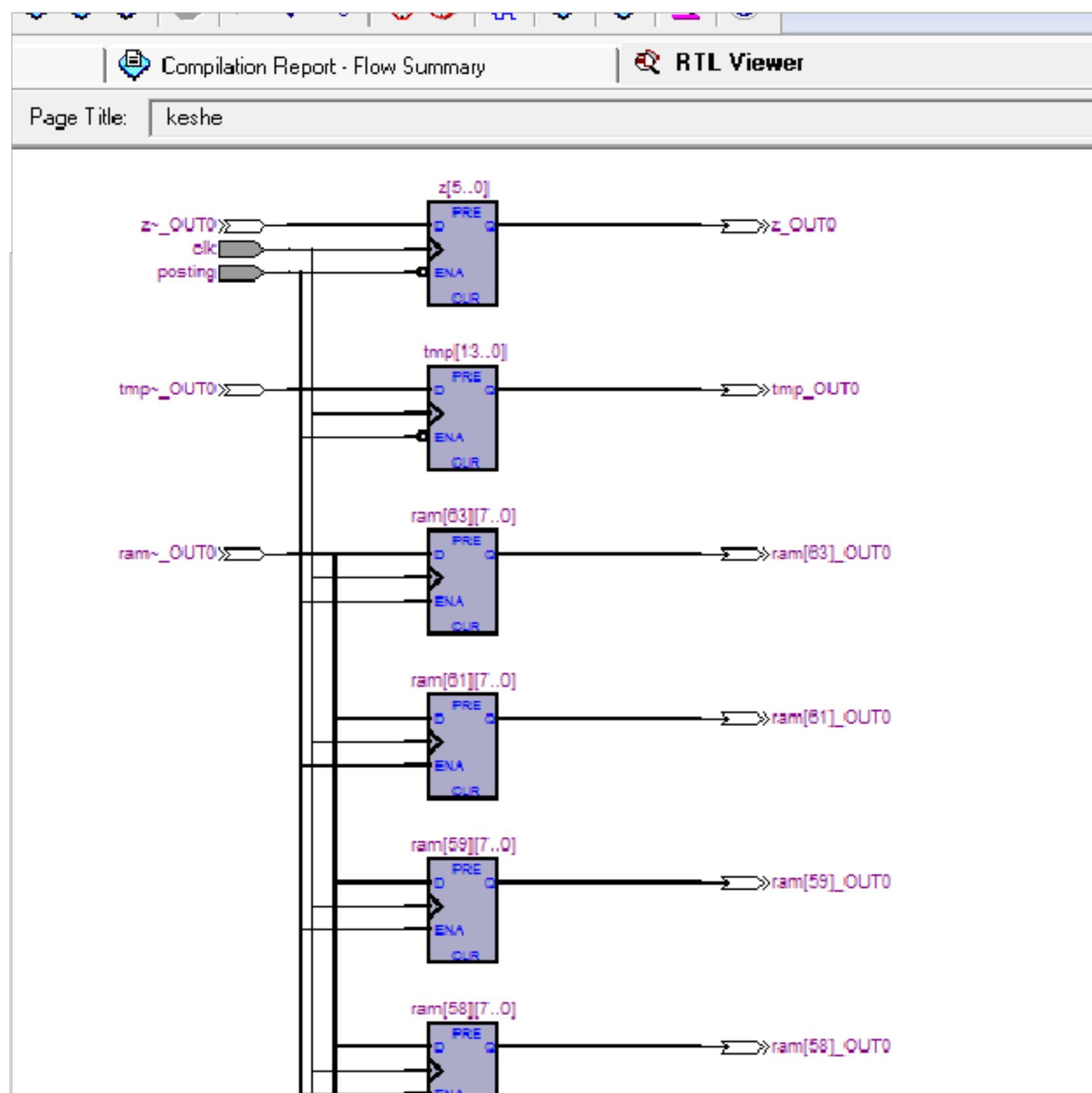


图 13 RTL 视图

4 程序分析

4.1 VHDL 语言分析

一个 VHDL 语言的设计程序描述的是一个电路单元，这个电路单元可以是一个门电路，或者是一个计数器，也可以是一个 CPU，一般情况下，一个完整的 VHDL 语言程序至少包括实体、结构体和程序包三个部分。实体给出电路单元的外部输入输出接口信号和引用信息，结构体给出了电路单元的内部结构和信号的行为特点，程序包定义在设计结构体和实体中将用到的常数、数据类型、子程序和设计好的电路单元等。

第一部分是程序包，程序包是用 VHDL 语言编写的共享文件，定义在设计结构体和实体中将要用到的常数、数据类型、子程序和设计好的电路单元等，放在文件目录名称为 IEEE 的程序包库中。

```
library ieee;  
use ieee.std_logic_1164.all;  
use ieee.std_logic_arith.all;  
use ieee.std_logic_unsigned.all;
```

第二部分是程序的实体，定义电路单元的输入/输出引脚名称。程序的实体名称可以任意取，但必须与 VHDL 程序的文件名称相同。实体的标示符是 ENTITY，实体以 ENTITY 开头，以 END 结束。

```
ENTITY    fulladder IS  
    PORT(a,b,Ci:in std_logic;  
          Co,s: out std_logic_vector(7 downto 0));  
END    fulladder;
```

其中，定义了 a, b, Ci 为输入信号引脚，定义 Co, s 为输出信号引脚。

第三部分是程序的结构体，具体描述电路的内部结构和逻辑功能。结构体有三种描述方式，分别是行为（BEHAVIOR）描述方式、数据流（DATAFLOW）描述方式和结构描述方式。其中数据流描述方式又被称为寄存器（RTL）描述方式。结构体以表示 ARCHITECTURE 开头，以 END 结尾。结构体的名称可以任取。

```
architecture behav of fulladder is  
BEGIN  
    s<=a xor b xor Ci;  
    Co<=(a and b)or(a and Ci)or(b and Ci);  
END fulladder
```

上面程序段中结构体的描述方式属于程序流描述方式。

4.2 主要函数语句分析

在程序设计中，主要使用的函数语句有两种：If-else语句和 case-when语句。这两种语句也是 VHDL 程序设计中常用的语句。二者都属于流程控制语句。流程控制语句通过条件控制开关决定是否执行一条或几条语句或重复执行一条或几条语句或跳过一条或几条语句。

IF 语句是一种条件语句，它根据语句中所设置的一种或多种条件，有选择地执行指定的顺序语句。IF 语句的语句结构有以下三种：

```
IF 条件句 Then      -- 第一种 IF 语句结构
    顺序语句
END IF
```

```
IF 条件句 Then      -- 第二种 IF 语句结构
    顺序语句
ELSE
    顺序语句
END IF
```

```
IF 条件句 Then      -- 第三种 IF 语句结构
    顺序语句
    ELSIF 条件句 Then
        顺序语句
    ...
ELSE
    顺序语句
END IF
```

CASE 语句根据满足的条件直接选择多项顺序语句中的一项执行，CASE 语句的结构如下：

```
CASE 表达式 IS
When 选择值 => 顺序语句
When 选择值 => 顺序语句
...
END CASE
```

当执行到 CASE 语句时，首先计算表达式的值，然后根据条件句中与之相同的选择值。执行对应的顺序语句，最后结束 CASE 语句。表达式可以是一个整数类型或枚举类型的值，也可以是由这些数据类型的值构成的数组。

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/588040142033007004>