

# 数字系统设计

郑海永

上课时间地点：周 2/9 10 节/3308  
电子信息科学与技术 2010 级和 2011 级

中国海洋大学 电子工程系

2013 年 4 月



# VHDL

- ① 硬件描述语言
- ② VHDL 基础
- ③ VHDL 实例
- ④ VHDL 入门
- ⑤ VHDL 基本语句
- ⑥ VHDL 深入

# 目录

## 1 硬件描述语言

- 发展
- 语言
- 方案
- 学习

## 2 演示

- 演示

## 3 实验课

- 开发板

# 内容提要

## 1 硬件描述语言

- 发展
- 语言
- 方案
- 学习

## 2 演示

- 演示

## 3 实验课

- 开发板

# 内容提要

## 1 硬件描述语言

- 发展
- 语言
- 方案
- 学习

## 2 演示

- 演示

## 3 实验课

- 开发板

# 概述

**硬件描述语言 HDL** 是一种用形式化方法描述数字电路和系统的语言。

- ① 数字电路系统的设计可以从上层到下层（从抽象到具体）逐层描述自己的设计思想，
- ② 用一系列分层次的模块来表示极其复杂的数字系统；
- ③ 然后利用电子设计自动化（EDA）工具，逐层进行仿真验证，
- ④ 再把其中需要变为实际电路的模块组合，经过自动综合工具转换到门级电路网表；
- ⑤ 再用专用集成电路 ASIC 或现场可编程门阵列 FPGA 自动布局布线工具，把网表转换为要实现的具体电路布线结构。

美国硅谷约有 90% 以上的 ASIC 和 FPGA 采用**硬件描述语言**设计。

# 历史

## 20 世纪 80 年代

- 已出现上百种硬件描述语言，对设计自动化曾起到了极大的促进和推动作用。
- 但是，这些语言一般各自面向特定的设计领域和层次，而且众多的语言使用户无所适从。
- 因此，急需一种面向设计的多领域、多层次并得到普遍认同的标准硬件描述语言。

## 20 世纪 80 年代后期

- VHDL 和 Verilog HDL 语言适应了这种趋势的要求，先后成为 IEEE 标准。
- Superlog、SystemC、Cynlib C++、C Level、ABEL HDL、AHDL 等。

# 历史

## 20 世纪 80 年代

- 已出现上百种硬件描述语言，对设计自动化曾起到了极大的促进和推动作用。
- 但是，这些语言一般各自面向特定的设计领域和层次，而且众多的语言使用户无所适从。
- 因此，急需一种面向设计的多领域、多层次并得到普遍认同的标准硬件描述语言。

## 20 世纪 80 年代后期

- VHDL 和 Verilog HDL 语言适应了这种趋势的要求，先后成为 IEEE 标准。
- Superlog、SystemC、Cynlib C++、C Level、ABEL HDL、AHDL 等。



# 历史

## 20 世纪 80 年代

- 已出现上百种硬件描述语言，对设计自动化曾起到了极大的促进和推动作用。
- 但是，这些语言一般各自面向特定的设计领域和层次，而且众多的语言使用户无所适从。
- 因此，急需一种面向设计的多领域、多层次并得到普遍认同的标准硬件描述语言。

## 20 世纪 80 年代后期

- VHDL 和 Verilog HDL 语言适应了这种趋势的要求，先后成为 IEEE 标准。
- Superlog、SystemC、Cynlib C++、C Level、ABEL HDL、AHDL 等。

# 历史

## 20 世纪 80 年代

- 已出现上百种硬件描述语言，对设计自动化曾起到了极大的促进和推动作用。
- 但是，这些语言一般各自面向特定的设计领域和层次，而且众多的语言使用户无所适从。
- 因此，急需一种面向设计的多领域、多层次并得到普遍认同的标准硬件描述语言。

## 20 世纪 80 年代后期

- VHDL 和 Verilog HDL 语言适应了这种趋势的要求，先后成为 IEEE 标准。
- Superlog、SystemC、Cynlib C++、C Level、ABEL HDL、AHDL 等。

# 历史

## 20 世纪 80 年代

- 已出现上百种硬件描述语言，对设计自动化曾起到了极大的促进和推动作用。
- 但是，这些语言一般各自面向特定的设计领域和层次，而且众多的语言使用户无所适从。
- 因此，急需一种面向设计的多领域、多层次并得到普遍认同的标准硬件描述语言。

## 20 世纪 80 年代后期

- VHDL 和 Verilog HDL 语言适应了这种趋势的要求，先后成为 IEEE 标准。
- Superlog、SystemC、Cynlib C++、C Level、ABEL HDL、AHDL 等。

# 未来

## 发展趋势

- 设计再利用。
- 设计验证（包括硬件和软件）。
- 互连问题将决定对时间、电源及噪声要求。
- 系统级芯片设计要求。

## EDA 发展

- ① 互用性标准。
- ② 扩展其高级库格式标准，使其包含物理领域的信息，以便解决互连问题。
- ③ 制定新的系统级设计语言标准。

# 未来

## 发展趋势

- 设计再利用。
- 设计验证（包括硬件和软件）。
- 互连问题将决定对时间、电源及噪声要求。
- 系统级芯片设计要求。

## EDA 发展

- ① 互用性标准。
- ② 扩展其高级库格式标准，使其包含物理领域的信息，以便解决互连问题。
- ③ 制定新的系统级设计语言标准。

# 中国

- ① 力争实现我国芯片设计自动化，使硬件设计的底层单元库可以自主研发。
- ② 选择合适的硬件描述语言及 EDA 工具并进行推广研究。
- ③ 深入 HDL 语言的综合和仿真等模型的研究，努力建立自主知识产权的 EDA 公司。
- ④ 积极加入 EDA 的标准化工作。
- ⑤ .....

# 内容提要

## 1 硬件描述语言

- 发展
- 语言
- 方案
- 学习

## 2 演示

- 演示

## 3 实验课

- 开发板

# ① VHDL

## VHDL

### 超高速集成电路硬件描述语言

#### Very-High-Speed Integrated Circuit Hardware Description Language

1980 美国军事工业需要描述电子系统的方法，美国国防部开始进行 VHDL 的开发。

1987 由 IEEE 将 VHDL 制定为标准 IEEE 1076-1987。

1993 第二个版本 VHDL-93 增加了一些新的命令和属性。

2008 IEEE 1076-2008 版本。

- 其综合库至今也没有标准化，不具有晶体管开关级的描述能力和模拟设计的描述能力。
- 对于特大型的系統级抽象描述方面的数字电路设计，VHDL 是较为合适的。



# ① VHDL

## VHDL

### 超高速集成电路硬件描述语言

#### Very-High-Speed Integrated Circuit Hardware Description Language

**1980** 美国军事工业需要描述电子系统的方法，美国国防部开始进行VHDL的开发。

**1987** 由IEEE将VHDL制定为标准IEEE 1076-1987。

**1993** 第二个版本VHDL-93增加了一些新的命令和属性。

**2008** IEEE 1076-2008版本。

- 其综合库至今也没有标准化，不具有晶体管开关级的描述能力和模拟设计的描述能力。
- 对于特大型的系統级抽象描述方面的数字电路设计，VHDL是较为合适的。

# ① VHDL

## VHDL

### 超高速集成电路硬件描述语言

Very-High-Speed Integrated Circuit Hardware Description Language

**1980** 美国军事工业需要描述电子系统的方法，美国国防部开始进行 VHDL 的开发。

**1987** 由 IEEE 将 VHDL 制定为标准 IEEE 1076-1987。

**1993** 第二个版本 VHDL-93 增加了一些新的命令和属性。

**2008** IEEE 1076-2008 版本。

- 其综合库至今也没有标准化，不具有晶体管开关级的描述能力和模拟设计的描述能力。
- 对于特大型的系統级抽象描述方面的数字电路设计，VHDL 是较为合适的。

# ① VHDL

## VHDL

### 超高速集成电路硬件描述语言

Very-High-Speed Integrated Circuit Hardware Description Language

**1980** 美国军事工业需要描述电子系统的方法，美国国防部开始进行VHDL的开发。

**1987** 由IEEE将VHDL制定为标准IEEE 1076-1987。

**1993** 第二个版本VHDL-93增加了一些新的命令和属性。

**2008 IEEE 1076-2008 版本。**

- 其综合库至今也没有标准化，不具有晶体管开关级的描述能力和模拟设计的描述能力。
- 对于特大型的系统级抽象描述方面的数字电路设计，VHDL是较为合适的。

# ① VHDL

## VHDL

### 超高速集成电路硬件描述语言

#### Very-High-Speed Integrated Circuit Hardware Description Language

**1980** 美国军事工业需要描述电子系统的方法，美国国防部开始进行 VHDL 的开发。

**1987** 由 IEEE 将 VHDL 制定为标准 IEEE 1076-1987。

**1993** 第二个版本 VHDL-93 增加了一些新的命令和属性。

**2008** IEEE 1076-2008 版本。

- 其综合库至今也没有标准化，不具有晶体管开关级的描述能力和模拟设计的描述能力。
- 对于特大型的系統级抽象描述方面的数字电路设计，VHDL 是较为合适的。

# ① VHDL

## VHDL

### 超高速集成电路硬件描述语言

#### Very-High-Speed Integrated Circuit Hardware Description Language

**1980** 美国军事工业需要描述电子系统的方法，美国国防部开始进行 VHDL 的开发。

**1987** 由 IEEE 将 VHDL 制定为标准 IEEE 1076-1987。

**1993** 第二个版本 VHDL-93 增加了一些新的命令和属性。

**2008** IEEE 1076-2008 版本。

- 其综合库至今也没有标准化，不具有晶体管开关级的描述能力和模拟设计的描述能力。
- 对于特大型的系統级抽象描述方面的数字电路设计，VHDL 是较为合适的。

# VHDL 特点

- ① 与其他的硬件描述语言相比，VHDL 具有更强的行为描述能力，从而决定了它成为系统设计领域最佳的硬件描述语言。  
强大的行为描述能力是避开具体的器件结构，从逻辑行为上描述和设计大规模电子系统的重要保证。
- ② VHDL 丰富的仿真语句和库函数，使得在任何大系统的设计早期就能查验设计系统的功能可行性，随时可对设计进行仿真模拟。
- ③ VHDL 语句的行为描述能力和程序结构决定了它具有支持大规模设计的分解和已有设计的再利用功能。
- ④ 对于用 VHDL 完成一个确定的设计，可以利用 EDA 工具进行逻辑综合和优化，并自动的把 VHDL 描述设计转变成门级网表。
- ⑤ VHDL 对设计的描述具有相对独立性，设计者可以不懂硬件的结构，也不必管理最终设计实现的目标器件是什么，而进行独立的设计。

# VHDL 特点

- ① 与其他的硬件描述语言相比，VHDL 具有更强的行为描述能力，从而决定了它成为系统设计领域最佳的硬件描述语言。
- ② VHDL 丰富的仿真语句和库函数，使得在任何大系统的设计早期就能查验设计系统的功能可行性，随时可对设计进行仿真模拟。
- ③ VHDL 语句的行为描述能力和程序结构决定了它具有支持大规模设计的分解和已有设计的再利用功能。
- ④ 对于用 VHDL 完成一个确定的设计，可以利用 EDA 工具进行逻辑综合和优化，并自动的把 VHDL 描述设计转变成门级网表。
- ⑤ VHDL 对设计的描述具有相对独立性，设计者可以不懂硬件的结构，也不必管理最终设计实现的目标器件是什么，而进行独立的设计。

# VHDL 特点

- ① 与其他的硬件描述语言相比，VHDL 具有更强的行为描述能力，从而决定了它成为系统设计领域最佳的硬件描述语言。
- ② VHDL 丰富的仿真语句和库函数，使得在任何大系统的设计早期就能查验设计系统的功能可行性，随时可对设计进行仿真模拟。
- ③ VHDL 语句的行为描述能力和程序结构决定了它具有支持大规模设计的分解和已有设计的再利用功能。
- ④ 对于用 VHDL 完成一个确定的设计，可以利用 EDA 工具进行逻辑综合和优化，并自动的把 VHDL 描述设计转变成门级网表。
- ⑤ VHDL 对设计的描述具有相对独立性，设计者可以不懂硬件的结构，也不必管理最终设计实现的目标器件是什么，而进行独立的设计。



# VHDL 特点

- ① 与其他的硬件描述语言相比，VHDL 具有更强的行为描述能力，从而决定了它成为系统设计领域最佳的硬件描述语言。
- ② VHDL 丰富的仿真语句和库函数，使得在任何大系统的设计早期就能查验设计系统的功能可行性，随时可对设计进行仿真模拟。
- ③ VHDL 语句的行为描述能力和程序结构决定了它具有支持大规模设计的分解和已有设计的再利用功能。
- ④ 对于用 VHDL 完成一个确定的设计，可以利用 EDA 工具进行逻辑综合和优化，并自动的把 VHDL 描述设计转变成门级网表。
- ⑤ VHDL 对设计的描述具有相对独立性，设计者可以不懂硬件的结构，也不必管理最终设计实现的目标器件是什么，而进行独立的设计。

# VHDL 特点

- ① 与其他的硬件描述语言相比，VHDL 具有更强的行为描述能力，从而决定了它成为系统设计领域最佳的硬件描述语言。
- ② VHDL 丰富的仿真语句和库函数，使得在任何大系统的设计早期就能查验设计系统的功能可行性，随时可对设计进行仿真模拟。
- ③ VHDL 语句的行为描述能力和程序结构决定了它具有支持大规模设计的分解和已有设计的再利用功能。
- ④ 对于用 VHDL 完成一个确定的设计，可以利用 EDA 工具进行逻辑综合和优化，并自动的把 VHDL 描述设计转变成门级网表。
- ⑤ VHDL 对设计的描述具有相对独立性，设计者可以不懂硬件的结构，也不必管理最终设计实现的目标器件是什么，而进行独立的设计。

## ② Verilog HDL

### Verilog HDL

1983 由 GDA (GateWay Design Automation) 公司 Phil Moorby 首创。

1984-1985 Phil Moorby 设计出了第一个名为 Verilog-XL 的仿真器。

1986 Phil Moorby 提出了用于快速门级仿真的 XL 算法。

1989 Cadence 公司收购 GDA 公司, Verilog HDL 语言成为 Cadence 公司私有财产。

1990 Cadence 公司公开 Verilog HDL 语言成立 OVI (Open Verilog International) 组织促进 Verilog HDL 语言的发展。

1995 IEEE 标准: Verilog HDL 1364-1995。

2001 发布 Verilog HDL 1364-2001 标准, 加入 Verilog HDL-A 标准使 Verilog 具有模拟设计描述的能力。

2005 Verilog 2005(IEEE Standard 1364-2005)

2009 SystemVerilog+Verilog=SystemVerilog 2009(IEEE Standard 1800-2009)

## ② Verilog HDL

### Verilog HDL

1983 由 GDA (GateWay Design Automation) 公司 Phil Moorby 首创。

1984-1985 Phil Moorby 设计出了第一个名为 Verilog-XL 的仿真器。

1986 Phil Moorby 提出了用于快速门级仿真的 XL 算法。

1989 Cadence 公司收购 GDA 公司, Verilog HDL 语言成为 Cadence 公司私有财产。

1990 Cadence 公司公开 Verilog HDL 语言成立 OVI (Open Verilog International) 组织促进 Verilog HDL 语言的发展。

1995 IEEE 标准: Verilog HDL 1364-1995。

2001 发布 Verilog HDL 1364-2001 标准, 加入 Verilog HDL-A 标准使 Verilog 具有模拟设计描述的能力。

2005 Verilog 2005(IEEE Standard 1364-2005)

2009 SystemVerilog+Verilog=SystemVerilog 2009(IEEE Standard 1800-2009)

## ② Verilog HDL

### Verilog HDL

1983 由 GDA (GateWay Design Automation) 公司 Phil Moorby 首创。

1984-1985 Phil Moorby 设计出了第一个名为 Verilog-XL 的仿真器。

1986 Phil Moorby 提出了用于快速门级仿真的 XL 算法。

1989 Cadence 公司收购 GDA 公司, Verilog HDL 语言成为 Cadence 公司私有财产。

1990 Cadence 公司公开 Verilog HDL 语言成立 OVI (Open Verilog International) 组织促进 Verilog HDL 语言的发展。

1995 IEEE 标准: Verilog HDL 1364-1995。

2001 发布 Verilog HDL 1364-2001 标准, 加入 Verilog HDL-A 标准使 Verilog 具有模拟设计描述的能力。

2005 Verilog 2005(IEEE Standard 1364-2005)

2009 SystemVerilog+Verilog=SystemVerilog 2009(IEEE Standard 1800-2009)

## ② Verilog HDL

### Verilog HDL

1983 由 GDA (GateWay Design Automation) 公司 Phil Moorby 首创。

1984-1985 Phil Moorby 设计出了第一个名为 Verilog-XL 的仿真器。

1986 Phil Moorby 提出了用于快速门级仿真的 XL 算法。

1989 Cadence 公司收购 GDA 公司, Verilog HDL 语言成为 Cadence 公司私有财产。

1990 Cadence 公司公开 Verilog HDL 语言成立 OVI (Open Verilog International) 组织促进 Verilog HDL 语言的发展。

1995 IEEE 标准: Verilog HDL 1364-1995。

2001 发布 Verilog HDL 1364-2001 标准, 加入 Verilog HDL-A 标准使 Verilog 具有模拟设计描述的能力。

2005 Verilog 2005(IEEE Standard 1364-2005)

2009 SystemVerilog+Verilog=SystemVerilog 2009(IEEE Standard 1800-2009)

## ② Verilog HDL

### Verilog HDL

1983 由 GDA (GateWay Design Automation) 公司 Phil Moorby 首创。

1984-1985 Phil Moorby 设计出了第一个名为 Verilog-XL 的仿真器。

1986 Phil Moorby 提出了用于快速门级仿真的 XL 算法。

1989 Cadence 公司收购 GDA 公司, Verilog HDL 语言成为 Cadence 公司私有财产。

1990 Cadence 公司公开 Verilog HDL 语言成立 OVI (Open Verilog International) 组织促进 Verilog HDL 语言的发展。

1995 IEEE 标准: Verilog HDL 1364-1995。

2001 发布 Verilog HDL 1364-2001 标准, 加入 Verilog HDL-A 标准使 Verilog 具有模拟设计描述的能力。

2005 Verilog 2005 (IEEE Standard 1364-2005)

2009 SystemVerilog + Verilog = SystemVerilog 2009 (IEEE Standard 1800-2009)

## ② Verilog HDL

### Verilog HDL

1983 由 GDA (GateWay Design Automation) 公司 Phil Moorby 首创。

1984-1985 Phil Moorby 设计出了第一个名为 Verilog-XL 的仿真器。

1986 Phil Moorby 提出了用于快速门级仿真的 XL 算法。

1989 Cadence 公司收购 GDA 公司, Verilog HDL 语言成为 Cadence 公司私有财产。

1990 Cadence 公司公开 Verilog HDL 语言成立 OVI (Open Verilog International) 组织促进 Verilog HDL 语言的发展。

1995 IEEE 标准: Verilog HDL 1364-1995。

2001 发布 Verilog HDL 1364-2001 标准, 加入 Verilog HDL-A 标准使 Verilog 具有模拟设计描述的能力。

2005 Verilog 2005(IEEE Standard 1364-2005)

2009 SystemVerilog+Verilog=SystemVerilog 2009(IEEE Standard 1800-2009)



## ② Verilog HDL

### Verilog HDL

1983 由 GDA (GateWay Design Automation) 公司 Phil Moorby 首创。

1984-1985 Phil Moorby 设计出了第一个名为 Verilog-XL 的仿真器。

1986 Phil Moorby 提出了用于快速门级仿真的 XL 算法。

1989 Cadence 公司收购 GDA 公司, Verilog HDL 语言成为 Cadence 公司私有财产。

1990 Cadence 公司公开 Verilog HDL 语言成立 OVI (Open Verilog International) 组织促进 Verilog HDL 语言的发展。

1995 IEEE 标准: Verilog HDL 1364-1995。

2001 发布 Verilog HDL 1364-2001 标准, 加入 Verilog HDL-A 标准使 Verilog 具有模拟设计描述的能力。

2005 Verilog 2005 (IEEE Standard 1364-2005)

2009 SystemVerilog + Verilog = SystemVerilog 2009 (IEEE Standard 1800-2009)

## ② Verilog HDL

### Verilog HDL

1983 由 GDA (GateWay Design Automation) 公司 Phil Moorby 首创。

1984-1985 Phil Moorby 设计出了第一个名为 Verilog-XL 的仿真器。

1986 Phil Moorby 提出了用于快速门级仿真的 XL 算法。

1989 Cadence 公司收购 GDA 公司, Verilog HDL 语言成为 Cadence 公司私有财产。

1990 Cadence 公司公开 Verilog HDL 语言成立 OVI (Open Verilog International) 组织促进 Verilog HDL 语言的发展。

1995 IEEE 标准: Verilog HDL 1364-1995。

2001 发布 Verilog HDL 1364-2001 标准, 加入 Verilog HDL-A 标准使 Verilog 具有模拟设计描述的能力。

2005 Verilog 2005(IEEE Standard 1364-2005)

2009 SystemVerilog+Verilog=SystemVerilog 2009(IEEE Standard 1800-2009)

## ② Verilog HDL

### Verilog HDL

1983 由 GDA (GateWay Design Automation) 公司 Phil Moorby 首创。

1984-1985 Phil Moorby 设计出了第一个名为 Verilog-XL 的仿真器。

1986 Phil Moorby 提出了用于快速门级仿真的 XL 算法。

1989 Cadence 公司收购 GDA 公司, Verilog HDL 语言成为 Cadence 公司私有财产。

1990 Cadence 公司公开 Verilog HDL 语言成立 OVI (Open Verilog International) 组织促进 Verilog HDL 语言的发展。

1995 IEEE 标准: Verilog HDL 1364-1995。

2001 发布 Verilog HDL 1364-2001 标准, 加入 Verilog HDL-A 标准使 Verilog 具有模拟设计描述的能力。

2005 Verilog 2005(IEEE Standard 1364-2005)

2009 SystemVerilog+Verilog=SystemVerilog 2009(IEEE Standard 1800-2009)

## ③ Superlog

### Superlog

Verilog C/C++/Java

**Verilog Phil Moorby 和 Peter Flake 等硬件描述语言专家在 Co-Design Automation 的 EDA 公司合作对 Verilog 进行扩展研究。**

1999 Co-Design 公司发布 Superlog™ 系统设计语言和两个开发工具：SystemSIM™ 和 SystemEXT™，分别用于系统开发和高级验证。

2001 Co-Design 公司向电子产业标准化组织 Accellera 发布 Superlog 扩展综合子集 ESS，在 Verilog 语言的 RTL 级综合子集的基础上提供更多级别的硬件综合抽象级。

## ③ Superlog

### Superlog

Verilog C/C++/Java

**Verilog** Phil Moorby 和 Peter Flake 等硬件描述语言专家在 Co-Design Automation 的 EDA 公司合作对 Verilog 进行扩展研究。

**1999** Co-Design 公司发布 Superlog™ 系统设计语言和两个开发工具：SystemSIM™ 和 SystemEXT™，分别用于系统开发和高级验证。

**2001** Co-Design 公司向电子产业标准化组织 Accellera 发布 Superlog 扩展综合子集 ESS，在 Verilog 语言的 RTL 级综合子集的基础上提供更多级别的硬件综合抽象级。

## ③ Superlog

### Superlog

Verilog C/C++/Java

- Verilog** Phil Moorby 和 Peter Flake 等硬件描述语言专家在 Co-Design Automation 的 EDA 公司合作对 Verilog 进行扩展研究。
- 1999** Co-Design 公司发布 Superlog™ 系统设计语言和两个开发工具：SystemSIM™ 和 SystemEXT™，分别用于系统开发和高级验证。
- 2001** Co-Design 公司向电子产业标准化组织 Accellera 发布 Superlog 扩展综合子集 ESS，在 Verilog 语言的 RTL 级综合子集的基础上提供更多级别的硬件综合抽象级。

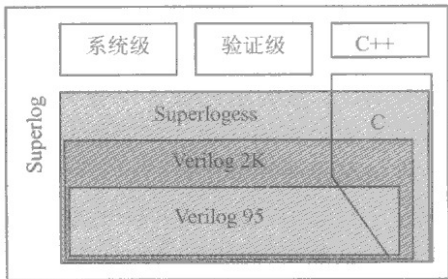


图2 Superlog体系结构

- ① Superlog 是 Verilog HDL 的超集，支持最新的 Verilog 2K 硬件模型。
- ② Superlog 提供 C 语言的结构、类型、指针，同时具有 C++ 的面向对象特性。
- ③ Superlog 扩展综合子集 ESS 提供一种新的硬件描述的综合抽象级。
- ④ 强大的验证功能，自动测试基准，如随机数产生、功能覆盖等。

## ④ SystemC

**SystemC**

SoC

SoC Synopsys 公司和 CoWare 公司合作开发 SystemC。

1999 40 多家世界著名 EDA 公司、IP 公司、半导体公司和嵌入式软件公司成立“开放 SystemC 联盟”(Open SystemC Initiative, OSCI)。

2001 著名 EDA 公司 Cadence 加入 SystemC 联盟。

Version 0.9(1999)⇒1.0⇒1.1⇒2.0(2001)⇒OSCI TLM-2.x  
IEEE Std. 1666-2005





## ④ SystemC

**SystemC**

SoC

SoC Synopsys 公司和 CoWare 公司合作开发 SystemC。

1999 40 多家世界著名 EDA 公司、IP 公司、半导体公司和嵌入式软件公司成立“开放 SystemC 联盟”(Open SystemC Initiative, OSCI)。

2001 著名 EDA 公司 Cadence 加入 SystemC 联盟。

Version 0.9(1999)⇒1.0⇒1.1⇒2.0(2001)⇒OSCI TLM-2.x  
IEEE Std. 1666-2005



## ④ SystemC

### SystemC

SoC

**SoC** Synopsys 公司和 CoWare 公司合作开发 SystemC。

**1999** 40 多家世界著名 EDA 公司、IP 公司、半导体公司和嵌入式软件公司成立“开放 SystemC 联盟”(Open SystemC Initiative, OSCI)。

**2001** 著名 EDA 公司 Cadence 加入 SystemC 联盟。

**Version** 0.9(1999)⇒1.0⇒1.1⇒2.0(2001)⇒OSCI TLM-2.x  
IEEE Std. 1666-2005



## ④ SystemC

### SystemC

SoC

**SoC** Synopsys 公司和 CoWare 公司合作开发 SystemC。

**1999** 40 多家世界著名 EDA 公司、IP 公司、半导体公司和嵌入式软件公司成立“开放 SystemC 联盟”(Open SystemC Initiative, OSCI)。

**2001** 著名 EDA 公司 Cadence 加入 SystemC 联盟。

**Version** 0.9(1999)⇒1.0⇒1.1⇒2.0(2001)⇒OSCI TLM-2.x  
IEEE Std. 1666-2005



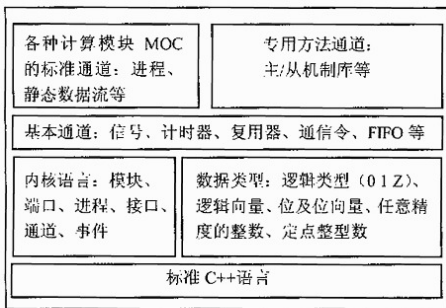


图1 SystemC体系结构

- 所有的 SystemC 都是基于 C++。
- SystemC 内核提供一个用于系统体系结构、并行、通信和同步时钟描述的模块。
- 完全支持内核描绘以外的数据类型、用户定义数据类型。
- 如果需要较低层的内容不依赖上层就可以直接使用。

## ⑤ ABEL-HDL

- 早期的硬件描述语言，从早期可编程逻辑器件的设计中发展而来。
- 在可编程逻辑器件设计中可方便准确的描述所设计电路的逻辑功能。
- 支持逻辑电路的多种表达形式：逻辑方程、真值表和状态图。
- 但 ABEL-HDL 只会在较小的范围内继续存在。

# 内容提要

## 1 硬件描述语言

- 发展
- 语言
- 方案
- 学习

## 2 演示

- 演示

## 3 实验课

- 开发板

# 硬件抽象级模型

系统级

算法级 用语言提供的高级结构实现算法运行的模型。

RTL级 描述数据在寄存器之间流动和如何处理、控制这些数据流动的模型。

门级 描述逻辑门以及逻辑门之间的连接模型。

开关级 描述器件中三极管和存储节点以及它们之间连接的模型。

验证级

综合抽象级

# 系统芯片设计方案

- 系统级采用 VHDL，软件级采用 C 语言，实现级采用 Verilog。
- VHDL 与 Verilog 互操作性已经逐步标准化，而软件越来越成为 SoC 设计的关键。
- 风险小，集成难度大，与原有方法完全兼容，有现成的开发工具。
- 但工具集成由开发者自行负责完成。
  
- 系统级及软件级采用 Superlog，硬件级和实现级均采用 Verilog HDL 描述。
- 与原有的硬件设计兼容，只要重新采购两个 Superlog 开发工具 SystemSIMTM 和 SystemEXTM。
- 风险较小，易于集成，与原硬件设计兼容性好，有集成开发环境。
  
- 系统级和软件级采用 SystemC，硬件级采用 SystemC 与常规的 Verilog HDL 互相转换。
- 与原来的软件编译环境完全兼容，只需一组描述类库和一个包含仿真核的库，就可以在通常的 ANSIC++ 编译器环境下开发；但硬件描述与原有方法完全不兼容。
- 风险较大，与原有软件开发兼容性好，硬件开发有风险。



# 系统芯片设计方案

- 系统级采用 VHDL，软件级采用 C 语言，实现级采用 Verilog。
- VHDL 与 Verilog 互操作性已经逐步标准化，而软件越来越成为 SoC 设计的关键。
- 风险小，集成难度大，与原有方法完全兼容，有现成的开发工具。
- 但工具集成由开发者自行负责完成。
  
- 系统级及软件级采用 Superlog，硬件级和实现级均采用 Verilog HDL 描述。
- 与原有的硬件设计兼容，只要重新采购两个 Superlog 开发工具 SystemSIMTM 和 SystemEXTM。
- 风险较小，易于集成，与原硬件设计兼容性好，有集成开发环境。
  
- 系统级和软件级采用 SystemC，硬件级采用 SystemC 与常规的 Verilog HDL 互相转换。
- 与原来的软件编译环境完全兼容，只需一组描述类库和一个包含仿真核的库，就可以在通常的 ANSIC++ 编译器环境下开发；但硬件描述与原有方法完全不兼容。
- 风险较大，与原有软件开发兼容性好，硬件开发有风险。

# 系统芯片设计方案

- 系统级采用 VHDL，软件级采用 C 语言，实现级采用 Verilog。
- VHDL 与 Verilog 互操作性已经逐步标准化，而软件越来越成为 SoC 设计的关键。
- 风险小，集成难度大，与原有方法完全兼容，有现成的开发工具。
- 但工具集成由开发者自行负责完成。
  
- 系统级及软件级采用 Superlog，硬件级和实现级均采用 Verilog HDL 描述。
- 与原有的硬件设计兼容，只要重新采购两个 Superlog 开发工具 SystemSIMTM 和 SystemEXTM。
- 风险较小，易于集成，与原硬件设计兼容性好，有集成开发环境。
  
- 系统级和软件级采用 SystemC，硬件级采用 SystemC 与常规的 Verilog HDL 互相转换。
- 与原来的软件编译环境完全兼容，只需一组描述类库和一个包含仿真核的库，就可以在通常的 ANSIC++ 编译器环境下开发；但硬件描述与原有方法完全不兼容。
- 风险较大，与原有软件开发兼容性好，硬件开发有风险。

# 内容提要

## 1 硬件描述语言

- 发展
- 语言
- 方案
- 学习

## 2 演示

- 演示

## 3 实验课

- 开发板

# 如何学习 HDL？

## 了解 HDL 的可综合性问题

- HDL 两种用途：系统仿真和硬件实现。
- 系统仿真几乎可以使用所有的语法和编程方法。
- 硬件实现必须保证程序“可综合”（程序的功能可以用硬件电路实现）。
- 所有的 HDL 描述都可以用于仿真，但不是所有的 HDL 描述都能用硬件实现。

## 用硬件电路设计思想来编写 HDL

- 充分理解 HDL 语句和硬件电路的关系。
- 编写 HDL 就是在描述一个电路。
- 多实践，多思考，多总结。

## 语法掌握贵在精，不在多

- 30% 的基本 HDL 语句就可以完成 95% 以上的电路设计。

# 如何学习 HDL？

## 了解 HDL 的可综合性问题

- HDL 两种用途：系统仿真和硬件实现。
- 系统仿真几乎可以使用所有的语法和编程方法。
- 硬件实现必须保证程序“可综合”（程序的功能可以用硬件电路实现）。
- 所有的 HDL 描述都可以用于仿真，但不是所有的 HDL 描述都能用硬件实现。

## 用硬件电路设计思想来编写 HDL

- 充分理解 HDL 语句和硬件电路的关系。
- 编写 HDL 就是在描述一个电路。
- 多实践，多思考，多总结。

## 语法掌握贵在精，不在多

- 30% 的基本 HDL 语句就可以完成 95% 以上的电路设计。

# 如何学习 HDL ?

## 了解 HDL 的可综合性问题

- HDL 两种用途：系统仿真和硬件实现。
- 系统仿真几乎可以使用所有的语法和编程方法。
- 硬件实现必须保证程序“可综合”（程序的功能可以用硬件电路实现）。
- 所有的 HDL 描述都可以用于仿真，但不是所有的 HDL 描述都能用硬件实现。

## 用硬件电路设计思想来编写 HDL

- 充分理解 HDL 语句和硬件电路的关系。
- 编写 HDL 就是在描述一个电路。
- 多实践，多思考，多总结。

## 语法掌握贵在精，不在多

- 30% 的基本 HDL 语句就可以完成 95% 以上的电路设计。

## HDL 与原理图输入法的关系

- HDL 和传统的原理图输入法  $\Leftrightarrow$  高级语言和汇编语言。
- HDL 可移植性好，使用方便，但效率不如原理图。
- 原理图输入的可控性好，效率高，比较直观，但设计烦琐，移植性差。
- 真正的 PLD/FPGA 设计中建议采用原理图和 HDL 结合的方法设计。
- 在最短的时间内，用自己最熟悉的工具设计出高效、稳定、符合设计要求的电路才是最终目的。

# HDL 开发流程

**文本编辑** VHDL 文件  $\Rightarrow$ .vhd；Verilog HDL 文件  $\Rightarrow$ .v。

**功能仿真** 将文件调入 HDL 仿真软件进行功能仿真，检查逻辑功能是否正确（前仿真）。

**逻辑综合** 将源文件调入逻辑综合软件进行综合生成 EDA 工业标准文件 (.edf)，即把语言综合成最简的布尔表达式和信号的连接关系。

**布局布线** 将.edf 文件调入 PLD 厂家提供的软件中进行布线，即把设计好的逻辑安放到 PLD/FPGA 内。

**时序仿真** 需要利用在布局布线中获得的精确参数，用仿真软件验证电路的时序。（后仿真）

**编程下载** 确认仿真无误后，将文件下载到芯片中。



# HDL 开发流程

**文本编辑** VHDL 文件  $\Rightarrow$ .vhd；Verilog HDL 文件  $\Rightarrow$ .v。

**功能仿真** 将文件调入 HDL 仿真软件进行功能仿真，检查逻辑功能是否正确（前仿真）。

**逻辑综合** 将源文件调入逻辑综合软件进行综合生成 EDA 工业标准文件 (.edf)，即把语言综合成最简的布尔表达式和信号的连接关系。

**布局布线** 将.edf 文件调入 PLD 厂家提供的软件中进行布线，即把设计好的逻辑安放到 PLD/FPGA 内。

**时序仿真** 需要利用在布局布线中获得的精确参数，用仿真软件验证电路的时序。（后仿真）

**编程下载** 确认仿真无误后，将文件下载到芯片中。

# HDL 开发流程

**文本编辑** VHDL 文件  $\Rightarrow$ .vhd；Verilog HDL 文件  $\Rightarrow$ .v。

**功能仿真** 将文件调入 HDL 仿真软件进行功能仿真，检查逻辑功能是否正确（前仿真）。

**逻辑综合** 将源文件调入逻辑综合软件进行综合生成 EDA 工业标准文件 (.edf)，即把语言综合成最简的布尔表达式和信号的连接关系。

**布局布线** 将.edf 文件调入 PLD 厂家提供的软件中进行布线，即把设计好的逻辑安放到 PLD/FPGA 内。

**时序仿真** 需要利用在布局布线中获得的精确参数，用仿真软件验证电路的时序。（后仿真）

**编程下载** 确认仿真无误后，将文件下载到芯片中。

# HDL 开发流程

**文本编辑** VHDL 文件  $\Rightarrow$ .vhd；Verilog HDL 文件  $\Rightarrow$ .v。

**功能仿真** 将文件调入 HDL 仿真软件进行功能仿真，检查逻辑功能是否正确（前仿真）。

**逻辑综合** 将源文件调入逻辑综合软件进行综合生成 EDA 工业标准文件 (.edf)，即把语言综合成最简的布尔表达式和信号的连接关系。

**布局布线** 将.edf 文件调入 PLD 厂家提供的软件中进行布线，即把设计好的逻辑安放到 PLD/FPGA 内。

**时序仿真** 需要利用在布局布线中获得的精确参数，用仿真软件验证电路的时序。（后仿真）

**编程下载** 确认仿真无误后，将文件下载到芯片中。

# HDL 开发流程

**文本编辑** VHDL 文件  $\Rightarrow$ .vhd；Verilog HDL 文件  $\Rightarrow$ .v。

**功能仿真** 将文件调入 HDL 仿真软件进行功能仿真，检查逻辑功能是否正确（前仿真）。

**逻辑综合** 将源文件调入逻辑综合软件进行综合生成 EDA 工业标准文件 (.edf)，即把语言综合成最简的布尔表达式和信号的连接关系。

**布局布线** 将.edf 文件调入 PLD 厂家提供的软件中进行布线，即把设计好的逻辑安放到 PLD/FPGA 内。

**时序仿真** 需要利用在布局布线中获得的精确参数，用仿真软件验证电路的时序。（后仿真）

**编程下载** 确认仿真无误后，将文件下载到芯片中。

# HDL 开发流程

**文本编辑** VHDL 文件  $\Rightarrow$ .vhd；Verilog HDL 文件  $\Rightarrow$ .v。

**功能仿真** 将文件调入 HDL 仿真软件进行功能仿真，检查逻辑功能是否正确（前仿真）。

**逻辑综合** 将源文件调入逻辑综合软件进行综合生成 EDA 工业标准文件（.edf），即把语言综合成最简的布尔表达式和信号的连接关系。

**布局布线** 将.edf 文件调入 PLD 厂家提供的软件中进行布线，即把设计好的逻辑安放到 PLD/FPGA 内。

**时序仿真** 需要利用在布局布线中获得的精确参数，用仿真软件验证电路的时序。（后仿真）

**编程下载** 确认仿真无误后，将文件下载到芯片中。

# 内容提要

## 1 硬件描述语言

- 发展
- 语言
- 方案
- 学习

## 2 演示

- 演示

## 3 实验课

- 开发板

# 内容提要

## 1 硬件描述语言

- 发展
- 语言
- 方案
- 学习

## 2 演示

- 演示

## 3 实验课

- 开发板

- 1 Quartus 软件用途，基本使用流程及参数设置。
- 2 FPGA 控制 ILX526A 线阵 CCD。
- 3 编译仿真演示。

侯桐桐，周秀华。



# 设计流程

- ① 创建项目
- ② 设计输入
- ③ 器件设置
- ④ 编译
- ⑤ 仿真
- ⑥ 引脚锁定
- ⑦ 下载编程

# 内容提要

## 1 硬件描述语言

- 发展
- 语言
- 方案
- 学习

## 2 演示

- 演示

## 3 实验课

- 开发板

# 内容提要

## 1 硬件描述语言

- 发展
- 语言
- 方案
- 学习

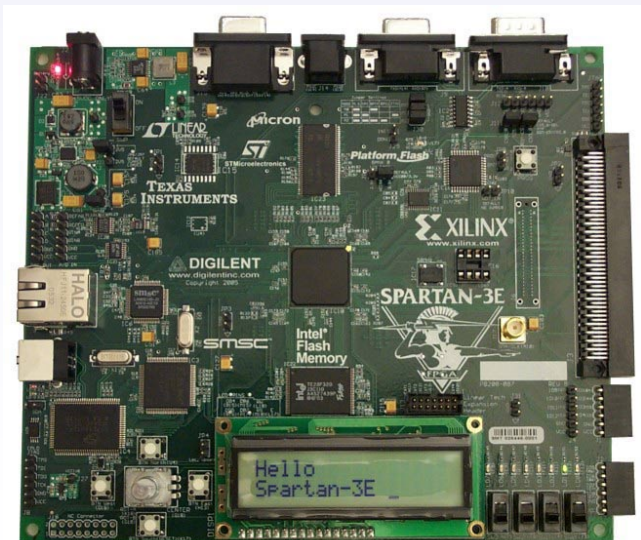
## 2 演示

- 演示

## 3 实验课

- 开发板

# Xilinx Spartan-3E Starter Kit Board



# Xilinx Spartan-3E Starter Kit Board

## What's Included

- Development board
- Universal power supply 100-240V, 50/60 Hz
- ISE® WebPACK™ software, ISE Foundation™ software evaluation, and the Embedded Development Kit (EDK)
- Handbook: Introduction to Programmable Logic Design Quick Start
- Starter Kit resource CD
- USB cable

# Xilinx Spartan-3E Starter Kit Board

## Key Features

### Xilinx Devices

- Spartan-3E FPGA (XC3S500E-4FG320C)
- CoolRunner™-II CPLD (XC2C64A-5VQ44C)
- Platform Flash (XCF04S-VO20C)

### Clocks

- 50 MHz crystal clock oscillator

### Memory

- 128 Mbit Parallel Flash
- 16 Mbit SPI Flash
- 64 MByte DDR SDRAM

### Display

- 16 character - 2 Line LCD

# Xilinx Spartan-3E Starter Kit Board

## Key Features

### Connectors and Interfaces

- Ethernet 10/100 Phy
- JTAG USB download
- Two 9-pin RS-232 serial port
- PS/2- style mouse/keyboard port, rotary encoder with push button
- Four slide switches
- Eight individual LED outputs
- Four momentary-contact push buttons
- 100-Pin expansion connection ports
- Three 6-pin expansion connectors

# Xilinx Spartan-3E Starter Kit Board

## Targeted Applications

- Markets: Consumer, Telecom/Datacom, Servers, Storage
- Applications: General Prototyping



# 资源

- Xilinx Forum : <http://forums.xilinx.com/>
- 赛灵思中文社区论坛
- OpenHW 社区 : <http://www.openhw.org/>
- EETOP 赛灵思社区 : <http://xilinx.eetop.cn/>
- 创新网赛灵思社区 : <http://xilinx.eetrend.com>
- 西电 Xilinx 俱乐部
- 北邮 Xilinx 俱乐部
- .....

# Spartan 3E 能做什么？

- MCU 8051
- TCI/IP 通信
- PacMan 吃豆子游戏
- .....