

基于Quartus II的计算机组成与 体系结构综合实验教程

杨 军 主编



科学出版社

教育部国家级特色专业(计算机科学与技术)资助

基于 Quartus II 的计算机组成与 体系结构综合实验教程

杨 军 主编

科 学 出 版 社

北 京

内 容 简 介

本书是针对计算机科学与技术专业的计算机组成与体系结构综合课程实验设计而编写的教材。全书共分 5 章, 先分别介绍计算机的组成、层次结构, Quartus II 8.0 软件的使用方法, 计算机基本器件的基本原理和 FPGA 实现方法; 然后, 通过简单加减法运算计算机实例来说明计算机各个关键部件的相互关系和工作方式; 最后, 循序渐进地讲解 14 个实例系统的设计。

本书突出包括指令系统与 CPU 设计等在内的计算机核心部件的设计方法, 通过器件的设计最终完成整个计算机系统的实现与扩展。书中采用 Quartus II 实例设计方法并提供相关代码, 方便学习与实践。

本书可作为普通高等院校计算机科学与技术、信息安全、电子信息工程、通信工程、自动化等专业的本科生教材, 也可供从事计算机体系结构研发的科研人员参考。

图书在版编目(CIP)数据

基于 Quartus II 的计算机组成与体系结构综合实验教程/杨军主编. —北京: 科学出版社, 2011.1
ISBN 978-7-03-029923-9

I. ①基… II. ①杨… III. ①可编程序逻辑器件—系统设计—高等学校—教材 IV. ①TP332.1

中国版本图书馆 CIP 数据核字(2011)第 002521 号

责任编辑: 张 濮 卜 新 / 责任校对: 李 影

责任印制: 张克忠 / 封面设计: 陈 敬

科学出版社出版

北京东黄城根北街 16 号

邮政编码: 100717

<http://www.sciencep.com>

北京市安泰印刷厂印刷

科学出版社发行 各地新华书店经销

*

2011 年 1 月第 一 版 开本: B5(720×1000)

2011 年 1 月第一次印刷 印张: 13 1/4

印数: 1—3 500 字数: 265 000

定价: 28.00 元

(如有印装质量问题, 我社负责调换)

《基于 Quartus II 的计算机组成 与体系结构综合实验教程》

编 委 会

主 编 杨 军

副 主 编 蒋慕蓉 孔 兵 武 浩

参编人员 唐佐侠 董 寅 丁 俊 郭义雄

李 娜 郭跃东 宋克俭 皇甫正茂

前 言

计算机组成与体系结构课程是计算机专业很重要的一门专业基础课，其工程性、技术性和实践性都很强。为使理论教学和实践教学紧密结合，培养学生的动手能力和解决工程问题的能力，我们编写了《基于 Quartus II 的计算机组成与体系结构综合实验教程》一书。通过本书的学习和实践，学生能够对计算机组成和系统结构的基本原理有清晰的概念和认识，对新型计算机的体系结构有比较深入的认识和理解。

近些年来，由于 EDA 技术和计算机硬件设计仿真技术的发展，计算机组成与体系结构实验课程有了新的方法。本书以可编程器件+EDA 软件+硬件描述语言的现代数字系统设计方法为出发点。学生在掌握 VHDL 或 Verilog HDL 后，进一步学习本书介绍的最新 Altera 公司系列设计软件 Quartus II 和 TD-CMA 综合实验平台，并以此来学习、研究计算机组成与系统结构的原理、方法，这对他们今后的设计工作有很大的帮助。

作者结合近几年的实践教学经验，针对学生面临的实际问题，参考大量设计书籍和技术文献，组织编写本书。在此，向这些资料的作者表示衷心的感谢。本书的实验内容充分吸纳、借鉴西安唐都科教仪器开发有限责任公司和 Altera 公司工程师的经验和资料，尤其感谢西安唐都科教仪器开发有限责任公司技术人员，他们在实例设计中给予了大量的技术支持，提高了本书的水平和实用价值。

作为本书副主编，蒋慕蓉、孔兵、武浩老师在本书的框架设计、内容安排、材料组织等方面给予了极大帮助和支持，提出了许多宝贵的意见。本书参编人员唐佐侠、董寅、丁俊、郭义雄、李娜、郭跃东、宋克俭、皇甫正茂在资料的收集、整理，源代码的设计、分析、仿真，硬件平台的验证等技术支持，以及书稿的录入、排版、绘图等方面做了大量的工作。在此，一并表示衷心的感谢！

基于 Quartus II 的计算机组成与体系结构综合实验课程涉及知识范围广，本书只是为初学者提供一些帮助和指导。本书配有实验项目工程文件和程序源代码，读者稍加修改便可应用于自己的工作或者完成自己的课题。读者可联系 gk@mail.sciencep.com 索取相关电子资源。

由于作者水平有限，加之计算机技术飞速发展，新的理念和技术层出不穷，问题很多，本书难免有不足之处，恳请广大读者批评、指正。

编 者

2010 年 10 月

目 录

前言

第 1 章 绪论	1
1.1 计算机系统的组成	1
1.1.1 计算机硬件的组成	1
1.1.2 计算机软件的组成	2
1.2 计算机系统的层次结构	4
1.2.1 计算机系统的多级层次结构	4
1.2.2 软件与硬件的逻辑等价性	5
1.3 计算机的工作过程	6
第 2 章 Quartus II 8.0 基本使用方法	8
2.1 概述	8
2.2 Quartus II 8.0 设计流程	8
2.3 Quartus II 设计方法	13
2.4 例解 Quartus II 8.0 设计	14
第 3 章 常用基本器件设计	25
3.1 寄存器设计	25
3.1.1 寄存器原理图设计	25
3.1.2 寄存器工作原理	26
3.1.3 寄存器程序描述	26
3.1.4 寄存器仿真	27
3.2 移位运算器设计	28
3.2.1 移位运算器原理图设计	28
3.2.2 移位运算器程序描述	28
3.2.3 移位运算器仿真	30
3.3 加减法运算器设计	30
3.3.1 加减法单元原理图设计	31
3.3.2 加减单元编码	31
3.3.3 多位加减法单元连接	32

3.3.4	加减法运算器原理图设计	33
3.3.5	加减法运算器程序描述	34
3.3.6	加减法运算器仿真	38
3.4	乘法运算器设计	38
3.4.1	乘法阵列原理图设计	39
3.4.2	乘法阵列编码	40
3.4.3	有符号数乘法运算器	42
3.5	同步计数器设计	45
3.5.1	设备同步工作	45
3.5.2	程序计数器	45
3.5.3	通用计数器	48
3.6	节拍器设计	51
3.6.1	节拍器电路设计	51
3.6.2	节拍器程序描述	51
3.6.3	节拍器工作原理	54
3.7	译码器设计	55
3.7.1	译码器电路设计	55
3.7.2	译码器程序描述	56
3.7.3	选择与通断控制电路	57
3.8	标志线设计	61
3.8.1	累加器标志线设计	61
3.8.2	数据监测标志设计	62
3.9	存储器设计	62
3.9.1	地址译码器设计	62
3.9.2	存储单元设计	64
3.9.3	256 存储单元存储器设计	66
第 4 章	计算机设计实例	68
4.1	简单计算机实验	68
4.1.1	设计 8 位累加器 A	68
4.1.2	设计 8 位数据寄存器 B	68
4.1.3	设计运算器 ALU	69
4.1.4	设计 8 位输出寄存器 O	70
4.1.5	设计 4 位地址寄存器 MAR	70
4.1.6	设计 EROM	70

4.1.7	设计 8 位指令寄存器 IR	71
4.1.8	设计 4 位程序计数器	72
4.1.9	设计控制器 CONT	73
4.1.10	设计加减运算计算机整机结构	76
4.1.11	功能仿真	77
4.1.12	实验小结	79
4.2	计算机整体设计	79
4.2.1	计算机组成结构	79
4.2.2	计算机功能设计目标	81
4.2.3	确定指令系统	81
4.3	总线结构设计	83
4.3.1	连接存储器和运算器	83
4.3.2	累加器、计数器连入总线	84
4.3.3	操作数寄存器、数据寄存器、输出寄存器连入总线	85
4.4	指令系统设计	86
4.4.1	指令全程分析	86
4.4.2	计算机指令全程表	87
4.5	控制器设计	92
4.5.1	控制器的基本功能和结构	92
4.5.2	时序控制信号	93
4.5.3	微程序控制器	95
4.5.4	硬联控制器	98
4.6	输入接口设备设计	99
4.6.1	缓冲区接口电路	99
4.6.2	操作系统的设计	101
4.7	计算机总体设计	102
4.7.1	顶层结构	102
4.7.2	输入程序数据控制	103
4.8	程序运行测试	104
4.8.1	顶层文件与连接驱动程序	104
4.8.2	检验程序执行	106
第 5 章	实训项目	107
5.1	运算器	107
5.1.1	基本运算器实验	107

5.1.2 多通路运算器与寄存器堆设计实验	110
5.1.3 阵列乘法器设计实验	115
5.2 存储系统	117
5.2.1 FIFO 先进先出存储器实验	117
5.2.2 Cache 控制器设计实验	119
5.3 控制器	124
5.3.1 时序发生器设计实验	124
5.3.2 微程序控制器实验	126
5.4 系统总线与总线接口	134
5.4.1 系统总线和具有基本输入、输出功能的总线接口实验	134
5.4.2 具有 DMA 控制功能的总线接口实验	139
5.5 指令系统	142
5.5.1 计算机系统的指令系统	142
5.5.2 基于 CISC 技术的模型计算机设计实验	144
5.5.3 基于 RISC 技术的模型计算机设计实验	151
5.6 时间并行性为特征的计算机系统	157
5.6.1 基于重叠技术的模型计算机设计实验	157
5.6.2 具有三级流水的模型机设计实验	167
5.7 指令并行性为特征的计算机系统	175
参考文献	184
附录 实验硬件平台及软件使用说明	185
附录 A TD-CMA 系统硬件环境	185
A1 系统硬件布局图	185
A2 系统实验单元电路	186
附录 B 软件使用说明	192
B1 TD-CMA 软件界面窗口介绍	192
B2 菜单功能介绍	194
附录 C 实用芯片介绍	198

第 1 章 绪 论

任何一台计算机都是由密切相关的硬件和软件组成的。硬件是计算机的物质基础，没有硬件计算机将不复存在；软件是发挥计算机功能，使得计算机能正常工作的程序，没有软件计算机无法投入使用。计算机的硬件和软件之间的关系很像电影放映机和电影胶片之间的关系。

1.1 计算机系统的组成

1.1.1 计算机硬件的组成

计算机硬件由运算器、控制器、存储器、输入设备和输出设备等五大部分组成，如图1-1所示。

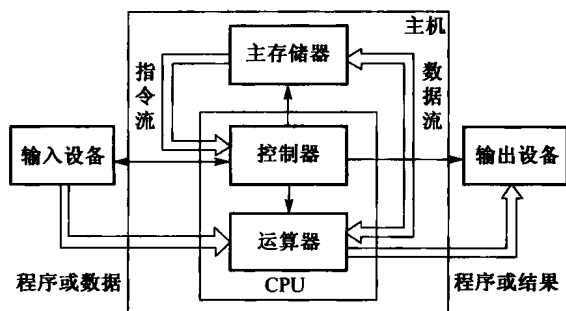


图 1-1 计算机硬件的组成

1. 运算器

运算器是用来完成各种运算的部件。通常将运算分成算术运算和逻辑运算两大类：算术运算是指需要考虑进位的加、减、乘、除运算，逻辑运算是指位对位的运算。运算器中除了包含一个能完成算术和逻辑运算功能的算术逻辑运算单元 (ALU) 之外，还要有可存放参加运算的操作数和运算结果的寄存器，以及在它们之间传送数据的通路。

2. 控制器

控制器是全机的指挥中心，由它向机内的各个部件发出各种命令，使全机构

成一个有机的整体。通常将运算器和控制器合起来称为“中央处理器”，简称 CPU (Central Processing Unit)。

3. 主存储器

主存储器是用来存放各类信息的部件，所有能在机内运行的程序和所需要的数据都要存放在存储器中，通常被称为主存储器或内存。一般来说，主存储器的容量越大，计算机的功能越强。通常将 CPU 和主存储器合起来称为计算机的“主机”。

4. 输入设备

输入设备是用来完成输入功能的部件。所有需要输入到机内的程序或数据，都是经过输入设备输入的。最常用的输入设备是键盘和鼠标。

5. 输出设备

输出设备是用来完成输出功能的部件。所有需要从机内输出的运算结果或在机内运行的程序、数据均可通过输出设备输出。最常用的输出设备是显示器和打印机。

由于这种硬件结构是由美籍数学家冯·诺依曼等提出来的，因此后人将其称为“冯·诺依曼结构”。这种结构的计算机具有如下的主要特点：

(1) “存储程序控制”是其主要特点。要求计算机完成的功能，需要事先编制成相应的程序，并将其输入到存储器中，于是计算机就在程序的控制下，有条不紊地工作。

(2) 程序和数据都是用二进制来表示的，机内进行的是二进制数的运算和存储。

(3) 程序由指令序列构成，任何一条指令都包含操作码和地址码两部分，操作码用来表明本指令的功能，地址码用来指定参加运算的操作数和运算结果存放的地址。

(4) 计算机以 CPU 为中心，输入的信息在 CPU 控制下才能写入主存储器，输出的信息也只有在 CPU 控制下才能从主存储器中取出送至相应的输出设备。

1.1.2 计算机软件的组成

软件是计算机系统的灵魂，可以这样认为，没有安装任何软件的“裸机”无法投入使用，没有安装足够的软件，计算机的功能将不能很好地发挥，计算机的应用范围也将受到很大的影响。

计算机软件可分成系统软件和应用软件两大类，所有软件都是使用某种语言

编写的、可完成各种功能的程序。由系统程序员编写的程序被称为“系统程序”；由应用程序员编写的程序被称为“应用程序”。

计算机语言有机器语言、汇编语言和高级语言之分。机器语言是计算机硬件可直接识别的最低级的语言，它要求程序员不仅要了解机器的硬件结构，而且要掌握各种机器指令的二进制编码格式，这是很难做到的。因此，利用机器语言编写程序是非常困难的。汇编语言是只能由汇编程序识别的语言，用汇编语言编写的源程序必须由汇编程序将其翻译成机器语言程序才能被机器执行，汇编语言与机器的硬件结构有一定的关系，能较好地发挥机器硬件的功能，因此要求程序员对机器硬件有一定的了解，在系统结构相同的序列机内部可实现汇编语言级的软件兼容。高级语言是完全脱离机器硬件结构，根据应用领域的不同要求设计出的通用的程序设计语言，品种繁多，目前世界上使用的高级语言有上百种，大体上可分成两大类。其一是会话型的高级语言，它必须由解释程序将其翻译成机器语言程序才能被执行，而且是一边解释一边执行，即每解释一条语句就执行一条语句，以达到人机会话的要求。例如，BASIC 语言就属于会话型的高级语言。其二是编译型的高级语言，它必须由各自的编译程序将其翻译成机器语言程序，而且必须是整个程序编译完毕后才能被机器执行。例如，FORTRAN、COBOL、PASCAL 和 C 语言等均属于编译型高级语言。由于高级语言与硬件结构无关，所以它具有很好的通用性和可移植性。任何一种高级语言可在各种不同类型的机器上运行，这显然是高级语言最大的优点和可取之处。换句话说，完全不了解计算机硬件的人，同样可使用高级语言来编写可执行的源程序。

系统软件通常包括用来管理机器的操作系统，各种高级语言的编译或解释程序，汇编程序，数据库管理程序，还有系统调试程序、故障诊断程序和错误检测程序等，它们是由系统程序员编写的，一般不允许修改，可固定存放在机内的只读存储器中，或者存放在外部存储器中，需要时调入机器的内存存储器中去执行。

操作系统(Operating System, OS)用来管理和控制计算机系统的软、硬件资源，提高这些资源的利用率，并为用户使用计算机提供方便，属于核心的系统程序。一个操作系统的研制，常常需要上百个“人·年”甚至更长。目前比较流行的操作系统有 MS-DOS、UNIX、Windows、Linux 等。汇编、编译或解释程序属于语言处理程序，是各种语言的源程序能在内存中正常运行的关键软件。

数据库管理程序是管理数据库的工具，计算机在工作过程中常常需要处理大量的数据信息，它们可集中存放在各种类型的数据库中。为建立、使用和维护数据库，各种类型的数据库管理程序应运而生。从早期的数据库管理系统 DBASE 到目前流行的面向对象的数据库管理系统 ORACLE、SYBASE 等，它们都能在操作系统的管理下，实现对数据库的管理和控制功能。

除此之外，系统调试程序、故障诊断程序和错误检测程序也是系统软件中不

可缺少的部分。应用软件是由应用程序员根据自己的要求使用特定语言编写的程序,用以实现计算机所要完成的各种功能。如上所述,计算机软件的组成可用图1-2来简明描述。

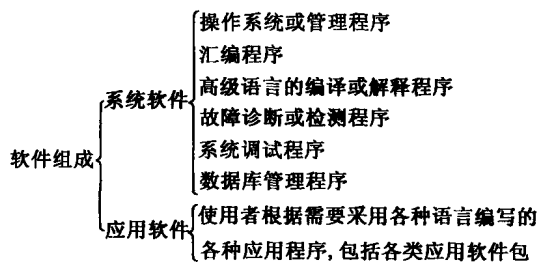


图 1-2 计算机软件的组成

1.2 计算机系统的层次结构

1.2.1 计算机系统的多级层次结构

计算机系统以硬件为基础,通过配置软件扩充功能,形成一个相当复杂的有机组合系统。它通常由 6 个或 6 个以上不同的级组成,每一级都能进行程序设计,如图1-3所示。

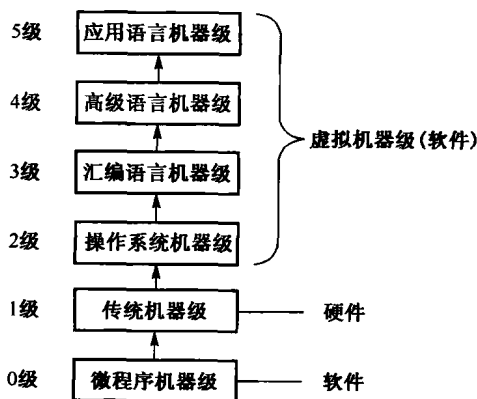


图 1-3 计算机系统的层次结构

微程序机器级,它不是任何一个计算机系统都必须存在的。在采用微程序控制方式的计算机中,它用微程序来解释机器指令,每一条机器指令都对应一个微程序。于是,任何一条机器指令的执行过程就成为一个微程序的执行过程。具体地说,它是事先编制好解释各条机器指令的微程序,并将其写入由只读存储器构

成的控制存储器中。因此一旦设计完毕,微程序是固定不变的,区别于“硬件”(hardware)和“软件”(software),可称为固件(firmware)。采用微程序控制方式,使得计算机的控制部件更规整,更便于设计和修改,可有效地缩短计算机的设计周期,受到计算机设计者们的青睐。

计算机最终所能执行的只能是二进制机器指令。传统机器级是一个硬件级,从这一级可以看到一台实际的机器。它也是计算机软、硬件的分界面。

操作系统机器级由管理机器各种资源的系统软件构成,在较小的计算机系统,例如,一些单机版计算机中,它比较简单,通常被称为“管理程序”或“监视程序”。随着计算机软、硬件技术的飞速发展,操作系统的功能越来越强,成为系统软件的核心。早期的操作系统用汇编语言编写,设计工作量很大,现代许多机器的操作系统用高级语言编写,不仅提高了设计效率,缩短了设计周期,更加增强了操作系统的功能。传统机器只有在操作系统管理下才能真正投入运行。随着操作系统的不断完善,对计算机的操作越来越方便,给操作者提供了更加友好的界面。

在操作系统机器级的上层是汇编语言机器级。汇编语言是和机器语言最接近的,它是用规定的一些助记符来编程的,助记符和指令系统一一对应。计算机只能识别机器语言,所以运行时需要把汇编语言翻译成机器码程序,再在计算机上执行。汇编语言便于记忆,比直接用机器语言编程方便得多。

高级语言机器级是运行比汇编语言更高级、接近于人类自然语言的语言所编写程序的机器级。高级语言程序员在这一级上不需要了解计算机的硬件、编译及操作系统等内容。

应用语言机器级是运行具体应用软件的机器级,应用程序员不必详细了解计算机的硬件即可方便地使用机器。

多数机器是将程序设计语言编写的程序直接翻译成机器语言在硬机器级上运行,有的是先翻译成层次较低一些的中间语言,然后再转换成机器语言在硬机器级上运行。

1.2.2 软件与硬件的逻辑等价性

随着大规模集成电路技术的发展和软件硬化的趋势,要明确划分硬件与软件已经显得比较困难了。有许多功能既可以直接由硬件来实现,也可以在硬件的支持下由软件来实现,对用户来说它们在功能上是等效的。随着 FPGA / CPLD 等技术的飞速发展,设计硬件完全可以像设计软件一样方便,可以对所设计的硬件功能随时加以修改,具有在线测试、仿真等功能,这使软、硬件之间的区别更加模糊了。例如,在本书后面的章节中要设计的一个乘法器实验,既可以在由基本单元提供的加法器和移位器功能部件的支持下,编写微程序,进行多次移位相加来实现乘法,也可以由 CPLD 直接设计一个具有乘法功能的运算部件。

一般来说,功能采用硬件实现的,其速度会比较快一些,功能的变更周期也可能会长一些,但是需要增加硬件成本;采用软件实现的,虽然其速度会慢一些,但不需要增加硬件成本,功能也比较容易改变。总之,能否选择恰当的软、硬件功能分配,取决于所选定的设计目标、系统性能、价格、速度、可靠性、存储容量、变更周期等因素,以及当时的硬件技术水平等。

1.3 计算机的工作过程

简单地说,计算机的工作过程就是执行程序的过程。程序由一系列机器指令构成,它是事先由程序员根据需要编制的。将编制好的程序经过输入设备顺序存放到主存储器中,并将其程序存放的首地址告诉控制器。于是控制器将从首地址开始从主存储器中取出一条指令,执行这条指令,再取出下条指令,执行下条指令。如此周而复始地工作一直到程序执行完毕,计算机便完成了该程序要求它完成的全部功能。图1-4所示为一个简单程序的执行过程。

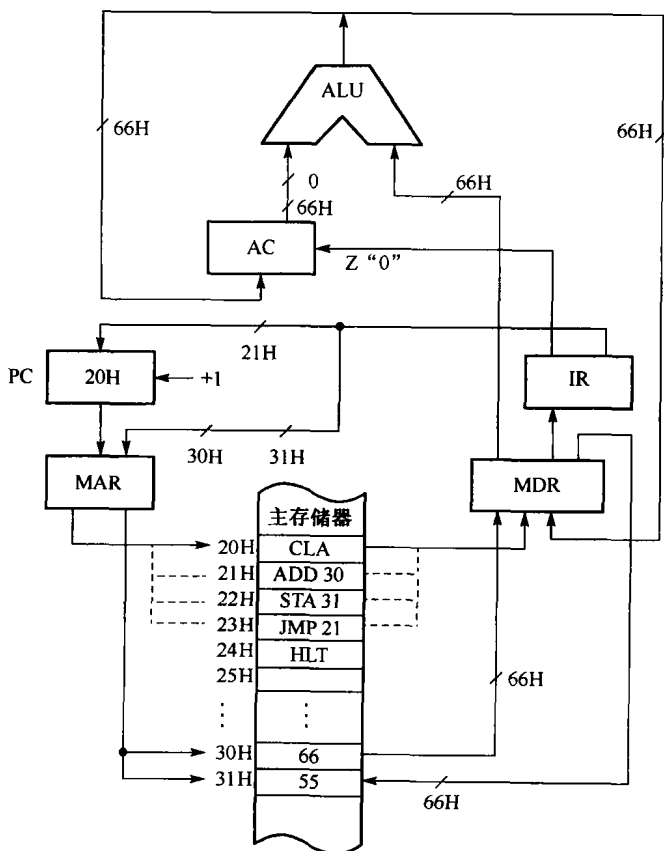


图 1-4 指令的执行过程

从图 1-4 中可以看出, 5 条机器指令用助记符表示, 已经存放在主存储器中从 20H 地址开始的一片存储区中, 本程序需要操作的原始数据是“01100110”(66H), 它已存放在主存储器的 30H 地址中。程序的首地址 20H 已装入程序计数器 PC 中, 于是控制器根据 PC 的内容从主存储器的 20H 地址中取出第一条机器指令 CLA, 这条机器指令的功能是清除累加器 AC 中的内容, 一旦这条机器指令从 20H 地址中取出, PC 中的内容已被修改为下一条机器指令的地址(21H), 这时, 执行 CLA 指令的结果将累加寄存器(AC)的内容清“0”; 接着执行的第二条指令是加法指令(ADD), 执行这条指令的结果是将累加寄存器(AC)的内容(“0”)与 30H 地址中的操作数(66H)相加后将其结果保留在 AC 中; 继续执行的第三条指令是存数指令(与 STA), 这条指令的功能是要将累加寄存器中的数据(66H)写入到主存储器的 31H 地址中; 继续执行的第四条指令是无条件转移指令(JMP), 其转移目标地址是 21H, 所以, 执行 JMP 指令的结果是将转移目标地址(21H)置入程序计数器 PC 中, 于是继续执行的下条命令不是暂停指令(HLT), 而是 21H 地址中的 ADD 指令。从而可以看出, 一般情况下, 在程序执行时, 只有遇到转移类指令, 才能改变程序的执行顺序。不难看出, 这是一个死循环程序, 任何时候不可能去执行 HLT 指令。

第2章 Quartus II 8.0 基本使用方法

2.1 概 述

Quartus II 8.0 设计软件提供完整的多平台设计环境，能够直接满足特定设计需要，为可编程芯片系统(SOPC)提供全面的设计环境。Quartus II 8.0 软件含有 FPGA 和 CPLD 设计所有阶段的解决方案，图 2-1 列出了 Quartus II 8.0 的设计流程。

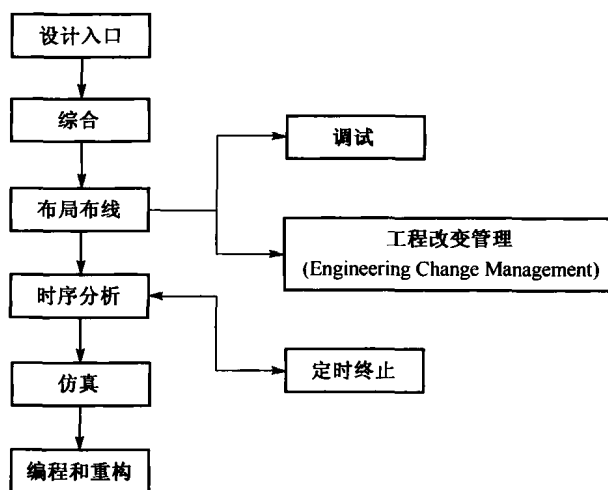


图 2-1 Quartus II 8.0 的设计流程

此外，Quartus II 软件为设计流程的每个阶段提供 Quartus II 图形用户界面、EDA 工具界面和命令行界面。可以在整个流程中只使用这些界面中的一个，也可以在设计流程的不同阶段使用不同界面。本章将对整个设计流程进行介绍，使用户对 Quartus II 的使用方法有一定的了解。

2.2 Quartus II 8.0 设计流程

可以使用 Quartus II 软件完成设计流程的所有阶段，它是一个全面易用的独立解决方案。图2-2所示为 Quartus II 图形用户界面所提供的功能。