

摘要

信号发生器是数字设备运行工作中必不可少的一部分，没有良好的信号源，最终就会导致系统不能够正常工作，更不必谈什么实现其它功能了。

本次论文主要研究了基于 FPGA 的函数信号发生器的设计思路与软硬件实现。首先介绍了本次设计任务的总体设计方案，以及该方案中涉及的知识点，所使用的软件及硬件基本知识。在此基础上进行了硬件电路的设计，主要采用 DDS（直接数字频率合成）方案，采用了 Altera 公司的低成本 cyclone II 系列 FPGA 的 EP2C5QC8 作为核心芯片，构建了外围的 0832DA 转换电路，以及 1MHZ 低通滤波电路。再次介绍系统软件的设计过程，给出了 FPGA 自底向上的设计思路，以及各个底层模块的设计原理与思路分析，最后介绍了相关软件的应用知识。最后一段介绍了论文的相关结论，进行仿真调试的过程。实现了设计任务的频率从 100HZ 到 1MHZ 可调，幅度从 0.1V 到 5V 可调的功能。系统的设计方案和设计过程具有参考和学习价值。

关键词：信号发生器 FPGA Modelsim Verilog 语言

Abstract

The signal generator is an essential part of the work of digital equipment operation, without a good source, and ultimately will cause the system to work properly, not to talk about the achievement of other features. This thesis studied the software and hardware design of the FPGA-based Signal Generator. First introduced the overall design of the design task, and knowledge involved in the program, use basic software and hardware knowledge. On this basis, the hardware circuit design, using DDS (Direct Digital Frequency Synthesizer) program, using Altera's low-cost cyclone II FPGA series EP2C5QC8 core chip to build a peripheral 0832DA conversion circuit, and 1MHZ low pass filter circuit. Re-introduce the system software design process, given FPGA design ideas from the bottom up, as well as the design principles and ideas of the bottom module, and finally introduced a software application knowledge. The last paragraph of the conclusions of the paper, simulation debugging process. Design task frequency is adjustable from 100HZ to 1MHZ function of the adjustable range from 0.1V to 5V. System design and design process has a reference and learning value.

Keywords: Signal generator, FPGA, Modlesim, Verilog HDL

目录

摘要	I
ABSTRACT	II
目录	III
前言	1
1 概述	3
1.1 FPGA 简介	3
1.2 MODELSIM 简介	8
1.3 DDS 基本原理介绍	9
2 设计方案	12
2.1 总体设计思路	12
2.2 方案论证	13
2.2.1 方案一	13
2.2.2 方案二	13
2.2.3 方案三	14
2.3 方案确定	14
3 硬件电路设计	16
3.1 硬件设计注意事项	16
3.2 DA 电路	16
3.3 滤波电路	18
3.4 硬件电路实现	20
4 软件设计	21
4.1 波形产生模块	21
4.1.1 正弦波	21
4.1.2 矩形波	23
4.1.3 三角波	24
4.2 频率控制模块	24
4.3 相位累加模块	25
4.3 选择波形模块	26
4.4 幅度控制模块	27

5 调 试.....	32
5.1 设计及仿真调试使用设备	32
5.2 调试方法.....	32
5.2.1 硬件调试.....	32
5.2.2 软件调试.....	32
5.2.3 综合调试.....	33
5.3 调试结果.....	33
5.3.1 软件仿真结果及分析.....	33
5.3.2 综合调试结果.....	38
总结.....	39
致谢辞.....	40
参考文献.....	41
附件 1 ROM 生成元程序.....	42
附件 2 40 位流水线加法器程序.....	44

前言

随着我国的经济日益增长，社会对电子产品的需求量也就越来越大，目前我国的电子产品市场正在迅速的壮大，市场前景广阔。FPGA(Field Programmable Gate Array,现场可编程门阵列)在现代数字电路设计中发挥着越来越重要的作用。FPGA/CPLD(Complex Programmable Logic Device)所具有的静态可重复编程和动态在系统重构的特性，使得硬件的功能可以像软件一样通过编程来修改，这样就极大地提高了电子系统设计的灵活性和通用性，缩短了产品的上市时间并降低可电子系统的开发成本，且可以毫不夸张地讲，FPGA/CPLD 能完成任何数字器件的功能，从简单的 74 电路到高性能的 CPU。它的影响毫不亚于 20 世纪 70 年代单片机的发明和使用。

现在随着电子技术的发展，产品的技术含量越来越高，使得芯片的复杂程度越来越高，人们对数万门乃至数百万门设计的需求也越来越多，特别是专用集成电路（ASIC）设计技术的日趋进步和完善，推动了数字系统设计的迅速发展。仅靠原理图输入方式已不能满足要求，采用硬件描述语言 VHDL 的设计方式应运而生，解决了传统用电路原理图设计大系统工程时的诸多不便，成为电子电路设计人员的最得力助手。设计工作从行为、功能级开始，并向着设计的高层次发展。这样就出现了第三代 EDA 系统，其特点是高层次设计的自动化。

第三代 EDA 系统中除了引入硬件描述语言，还引入了行为综合工具和逻辑综合工具，采用较高的抽象层次进行设计，并按层次式方法进行设计，可大大提高处理复杂设计的能力，缩短设计周期，综合优化工具的采用使芯片的品质如面积、速度和功耗等获得了优化，因而第三代 EDA 系统迅速得到了推广应用。

目前，最通用的硬件描述语言有 VHDL 和 VerilogHDL 两种，现在大多设计者都使用 93 年版标准的 VHDL，并且通过了 IEEE 认定，成为世界范围内通用的数字系统设计标准。VerilogHDL 是一种新兴的程序设计语言，使用 VerilogHDL 进行设计其性能总是比常规使用 CPU 或者

MCU 的程序设计语言在性能上要高好几个数量级。这就是说，在传统上使用软件语言的地方，VerilogHDL 语言作为一种新的实现方式会应用得越来越广泛。本课题设计是采用美国 Altera 公司的 cyclone II 器件，使用的是 Altera 公司的 EDA 软件平台 quartus II 可编程逻辑器件开发软件。基于 EDA 工具的 FPGA/CPLD 的开发流程 CPLD/FPGA 器件的设计一般可分为设计输入、设计实现和编程三个设计步骤：

1.设计输入方式主要由文本输入和图形输入两种，可根据需要选择，也可混合输入。EDA 工具会自动检查语法；

2.设计实现阶段 EDA 工具对设计文件进行编译，进行逻辑综合、优化，并针对器件进行映射、布局、布线，产生相应的适配文件；

3.编程阶段 EDA 软件将适配文件配置到相应的 CPLD / FPGA 器件中，使其能够实现预期的功能。

信号发生器是数字设备运行工作中必不可少的一部分，没有良好的信号源，最终就会导致系统不能够正常工作，更不必谈什么实现其它功能了。不论是处于开发还是故障检修阶段，输出标准且性能优秀的信号发生器总是能够带来工作效率的大幅提升，使新产品有一个标准的信号源、损坏的系统得到正确校验，不会被一些故障所蒙蔽。在传统的信号发生器中，大都使用分立元件，而且体积庞大携带不便，且大部分只能输出一种脉冲信号波形。在设计领域，不管采用什么技术生产，生产的产品用在哪里，其产品设计的宗旨都是离不开以下几点：实用性高、成本低、可升级、功能完善可扩展等！使用专用的数字电路设计的信号发生器，设备成本高、使用复杂。基于以上考虑，在中小型数字电路的设计和测试中，迫切需要设计一种小型易用成本低廉的信号发生器。此课题的设计以小型经济，集成度高，性能稳定，使用方便为指导，在功能上力求完善实用，同时兼顾商业价值与应用价值的体现。

1 概述

1.1 FPGA 简介

FPGA (Field-Programmable Gate Array), 即现场可编程门阵列, 它是在 PAL、GAL、CPLD 等可编程器件的基础上进一步发展的产物。它是作为专用集成电路 (ASIC) 领域中的一种半定制电路而出现的, 既解决了定制电路的不足, 又克服了原有可编程器件门电路数有限的缺点。

(1) 背景

目前以硬件描述语言 (Verilog 或 VHDL) 所完成的电路设计, 可以经过简单的综合与布局, 快速的烧录至 FPGA 上进行测试, 是现代 IC 设计验证的技术主流。这些可编辑元件可以被用来实现一些基本的逻辑门电路 (比如 AND、OR、XOR、NOT) 或者更复杂一些的组合功能比如解码器或数学方程式。在大多数的 FPGA 里面, 这些可编辑的元件里也包含记忆元件例如触发器 (Flip-flop) 或者其他更加完整的记忆块。

系统设计师可以根据需要通过可编辑的连接把 FPGA 内部的逻辑块连接起来, 就好像一个电路试验板被放在了一个芯片里。一个出厂后的成品 FPGA 的逻辑块和连接可以按照设计者而改变, 所以 FPGA 可以完成所需要的逻辑功能。

FPGA 一般来说比 ASIC (专用集成芯片) 的速度要慢, 无法完成复杂的设计, 而且消耗更多的电能。但是他们也有很多的优点比如可以快速成品, 可以被修改来改正程序中的错误和更便宜的造价。厂商也可能会提供便宜的但是编辑能力差的 FPGA。因为这些芯片有比较差的可编辑能力, 所以这些设计的开发是在普通的 FPGA 上完成的, 然后将设计转移到一个类似于 ASIC 的芯片上。另外一种方法是用 CPLD (复杂可编程逻辑器件)。

(2).CPLD 与 FPGA 的关系

早在 1980 年代中期, FPGA 已经在 PLD 设备中扎根。CPLD 和 FPGA 包括了一些相对大数量的可编辑逻辑单元。CPLD 逻辑门的密度在几千

到几万个逻辑单元之间，而 FPGA 通常是在几万到几百万。

CPLD 和 FPGA 的主要区别是他们的系统结构。CPLD 是一个有点限制性的结构。这个结构由一个或者多个可编辑的结果之和的逻辑组列和一些相对少量的锁定的寄存器。这样的结果是缺乏编辑灵活性，但是却有可以预计的延迟时间和逻辑单元对连接单元高比率的优点。而 FPGA 却是有很多的连接单元，这样虽然让它可以更加灵活的编辑，但是结构却复杂的多。

CPLD 和 FPGA 另外一个区别是大多数的 FPGA 含有高层次的内置模块（比如加法器和乘法器）和内置的记忆体。因此一个有关的重要区别是很多新的 FPGA 支持完全的或者部分的系统内重新配置。允许他们的设计随着系统升级或者动态重新配置而改变。一些 FPGA 可以让设备的一部分重新编辑而其他部分继续正常运行。

(3).FPGA 工作原理

FPGA 采用了逻辑单元阵列 LCA (Logic Cell Array) 这样一个概念，内部包括可配置逻辑模块 CLB (Configurable Logic Block)、输出输入模块 IOB (Input Output Block) 和内部连线 (Interconnect) 三个部分。现场可编程门阵列 (FPGA) 是可编程器件。与传统逻辑电路和门阵列 (如 PAL, GAL 及 CPLD 器件) 相比，FPGA 具有不同的结构，FPGA 利用小型查找表 (16×1RAM) 来实现组合逻辑，每个查找表连接到一个 D 触发器的输入端，触发器再来驱动其他逻辑电路或驱动 I/O，由此构成了既可实现组合逻辑功能又可实现时序逻辑功能的基本逻辑单元模块，这些模块间利用金属连线互相连接或连接到 I/O 模块。FPGA 的逻辑是通过向内部静态存储单元加载编程数据来实现的，存储在存储器单元中的值决定了逻辑单元的逻辑功能以及各模块之间或模块与 I/O 间的联接方式，并最终决定了 FPGA 所能实现的功能，FPGA 允许无限次的编程。

目前主流的 FPGA 仍是基于查找表技术的，已经远远超出了先前版本的基本性能，并且整合了常用功能（如 RAM、时钟管理和 DSP）的硬核 (ASIC 型) 模块。如图 1-1 所示（注：图 1-1 只是一个示意图，实际上每一个系列的 FPGA 都有其相应的内部结构），FPGA 芯片主要由 7 部分组成，分别为：可编程输入输出单元、基本可编程逻辑单元、完整的时钟管理、嵌入式 RAM、丰富的布线资源、内嵌的底层功能单元和内嵌专用硬件模块。

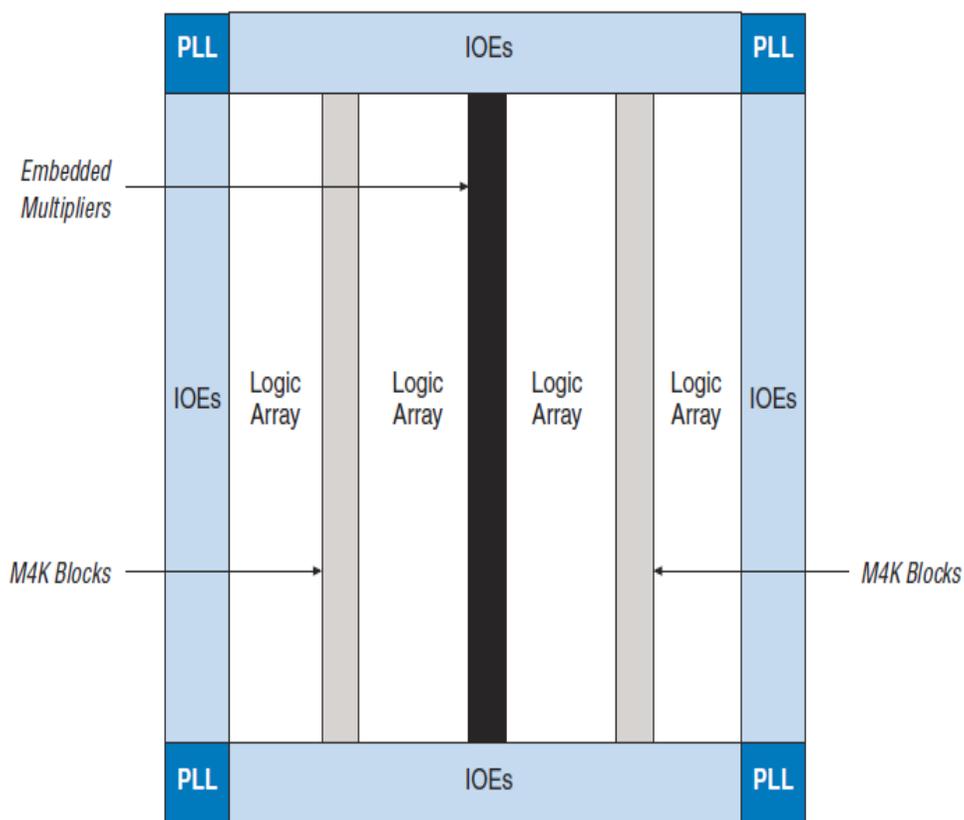


图 1.1 FPGA 芯片的内部结构

(4).FPGA 芯片的内部结构

1) 可编程输入输出单元 (IOB)

可编程输入/输出单元简称 I/O 单元，是芯片与外界电路的接口部分，完成不同电气特性下对输入/输出信号的驱动与匹配要求，其示意结构如图 1-2 所示。FPGA 内的 I/O 按组分类，每组都能够独立地支持不同的 I/O 标准。通过软件的灵活配置，可适配不同的电气标准与 I/O 物理特性，可以调整驱动电流的大小，可以改变上、下拉电阻。目前，I/O 口的频率也越来越高，一些高端的 FPGA 通过 DDR 寄存器技术可以支持高达 2Gbps 的数据速率。

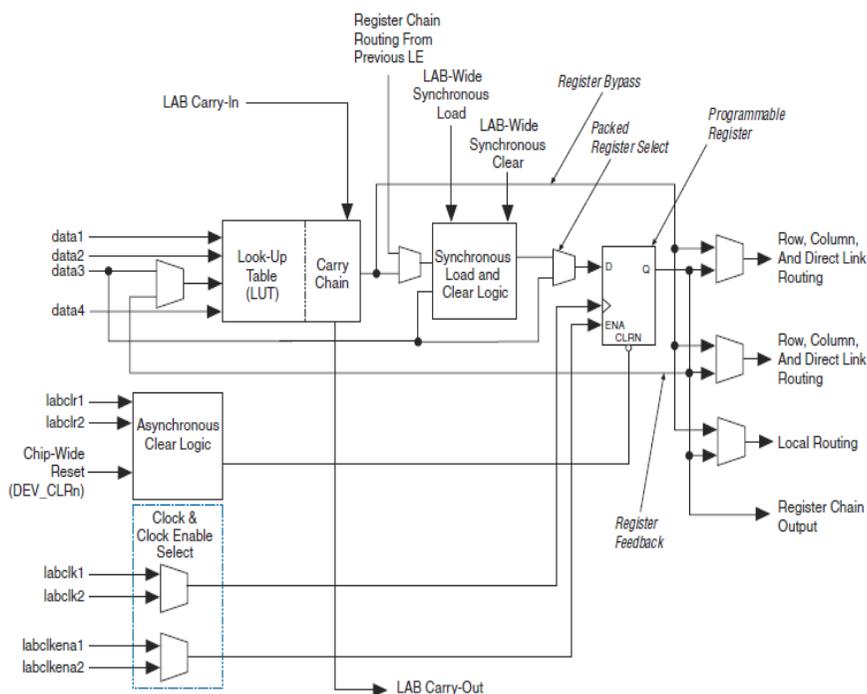


图 1.2 典型的 IOB 内部结构示意图

外部输入信号可以通过 IOB 模块的存储单元输入到 FPGA 的内部，也可以直接输入 FPGA 内部。当外部输入信号经过 IOB 模块的存储单元输入到 FPGA 内部时，其保持时间（Hold Time）的要求可以降低，通常默认为 0。

为了便于管理和适应多种电器标准，FPGA 的 IOB 被划分为若干组（bank），每个 bank 的接口标准由其接口电压 VCCO 决定，一个 bank 只能有一种 VCCO，但不同 bank 的 VCCO 可以不同。只有相同电气标准的端口才能连接在一起，VCCO 电压相同是接口标准的基本条件。

2) 可配置逻辑块 (CLB)

CLB 是 FPGA 内的基本逻辑单元。CLB 的实际数量和特性会依器件的不同而不同，但是每个 CLB 都包含一个可配置开关矩阵，此矩阵由 4 或 6 个输入、一些选型电路（多路复用器等）和触发器组成。开关矩阵是高度灵活的，可以对其进行配置以便处理组合逻辑、移位寄存器或 RAM。在 Xilinx 公司的 FPGA 器件中，CLB 由多个（一般为 4 个或 2 个）相同的 Slice 和附加逻辑构成，如图 1-3 所示。每个 CLB 模块

不仅可以用于实现组合逻辑、时序逻辑，还可以配置为分布式 RAM 和分布式 ROM。

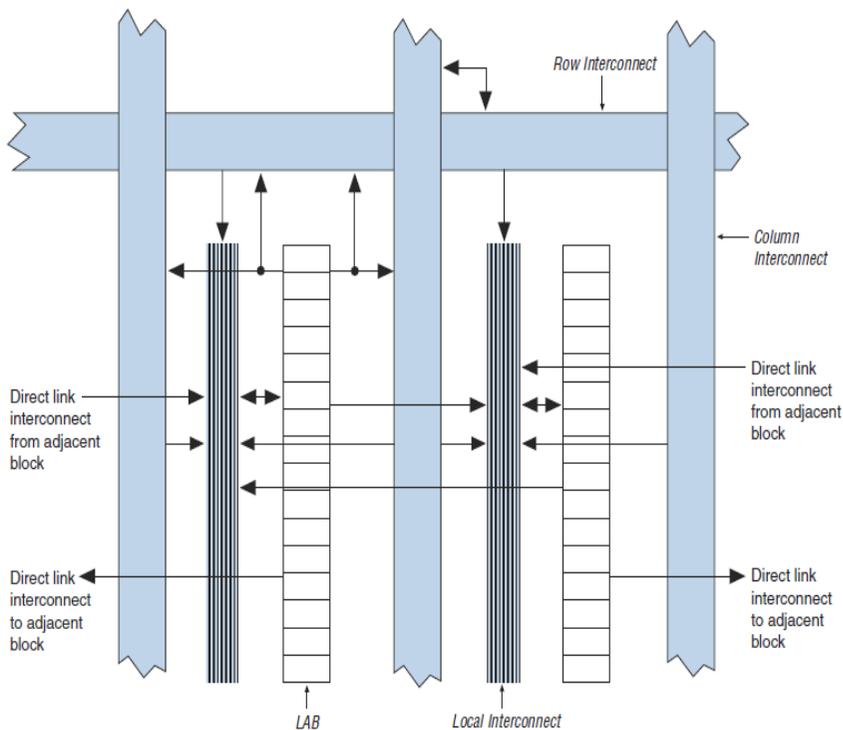


图 1.3 典型的 CLB 结构示意图

(5). 基本特点

·采用 FPGA 设计 ASIC 电路(专用集成电路),用户不需要投片生产,就能得到合用的芯片。

·FPGA 可做其它全定制或半定制 ASIC 电路的中试样片。

·FPGA 内部有丰富的触发器和 I/O 引脚。

·FPGA 是 ASIC 电路中设计周期最短、开发费用最低、风险最小的器件之一。

·FPGA 采用高速 CMOS 工艺,功耗低,可以与 CMOS、TTL 电平兼容。可以说, FPGA 芯片是小批量系统提高系统集成度、可靠性的最佳选择之一。

FPGA 是由存放在片内 RAM 中的程序来设置其工作状态的,因此,

工作时需要对片内的 RAM 进行编程。用户可以根据不同的配置模式，采用不同的编程方式。加电时，FPGA 芯片将 EPROM 中数据读入片内编程 RAM 中，配置完成后，FPGA 进入工作状态。掉电后，FPGA 恢复成白片，内部逻辑关系消失，因此，FPGA 能够反复使用。FPGA 的编程无须专用的 FPGA 编程器，只须用通用的 EPROM、PROM 编程器即可。当需要修改 FPGA 功能时，只需换一片 EPROM 即可。这样，同一片 FPGA，不同的编程数据，可以产生不同的电路功能。因此，FPGA 的使用非常灵活。

1.2 modelsim 简介

Mentor 公司的 ModelSim 是业界最优秀的 HDL 语言仿真软件，它能提供友好的仿真环境，是业界唯一的单内核支持 VHDL 和 Verilog 混合仿真的仿真器。它采用直接优化的编译技术、Tcl/Tk 技术、和单一内核仿真技术，编译仿真速度快，编译的代码与平台无关，便于保护 IP 核，个性化的图形界面和用户接口，为用户加快调错提供强有力的手段，是 FPGA/ASIC 设计的首选仿真软件。

(1)主要特点：

- RTL 和门级优化，本地编译结构，编译仿真速度快，跨平台跨版本仿真；
- 单内核 VHDL 和 Verilog 混合仿真；
- 源代码模版和助手，项目管理；
- 集成了性能分析、波形比较、代码覆盖、数据流 ChaseX、Signal Spy、虚拟对象 Virtual Object、Memory 窗口、Assertion 窗口、源码窗口显示信号值、信号条件断点等众多调试功能；
- C 和 Tcl/Tk 接口，C 调试；
- 对 SystemC 的直接支持，和 HDL 任意混合；
- 支持 SystemVerilog 的设计功能；
- 对系统级描述语言的最全面支持，SystemVerilog，SystemC，PSL；
- ASIC Sign off。

ModelSim 分几种不同的版本：SE、PE、LE 和 OEM，其中 SE 是最高级的版本，而集成在 Actel、Atmel、Altera、Xilinx 以及 Lattice 等 FPGA 厂商设计工具中的均是其 OEM 版本。SE 版和 OEM 版在功能和

性能方面有较大差别，比如对于大家都关心的仿真速度问题，以 Xilinx 公司提供的 OEM 版本 ModelSim XE 为例，对于代码少于 40000 行的设计，ModelSim SE 比 ModelSim XE 要快 10 倍；对于代码超过 40000 行的设计，ModelSim SE 要比 ModelSim XE 快近 40 倍。ModelSim SE 支持 PC、UNIX 和 LINUX 混合平台；提供全面完善以及高性能的验证功能；全面支持业界广泛的标准；Mentor Graphics 公司提供业界最好的技术支持与服务。

1.3 DDS 基本原理介绍

DDS 是一种全数字的频率合成方法，其基本结构主要由相位累加器、波形 ROM、D/A 转换器和低通滤波器四个部分构成，相位累加器，波形 ROM，DA 转换器，低通滤波器。具体结构参见上文图 2。

相位累加器由一个 N 位的加法器和一个 N 位的寄存器构成，通过把上一个时钟的累加结果反馈回加法器的输入端而实现累加功能。从而使输出结果每一个时钟周期递增 K。这里 N 为相位累加器的字长，K 称为频率控制字。

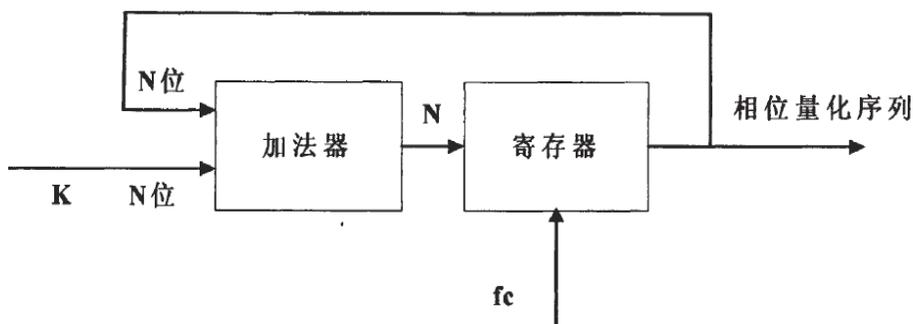


图 1.4 相位累加器示意图

其中相位累加器字长为 N，DDS 控制时钟频率为 f ，频率控制字为 K。DDS 直接从“相位”的概念出发进行频率合成。相位累加器由加法器与累加寄存器级联构成。每来一个时钟脉冲 f ，加法器将频率控制字 K 与累加寄存器输出的累加相位数据相加，把相加后的结果送至累加寄存器的数据输入端。累加寄存器将加法器在上一个时钟脉冲作用后所产生的新相位数据反馈到加法器的输入端，以使加法器在下一个时钟脉冲的作用下继续与频率控制字相加。这样，相位累加器在每一个时钟脉冲

输入时，把频率控制字累加一次，相位累加器输出的数据就是合成信号的相位，相位累加器的溢出频率就是 DDS 输出的信号频率。

DDS 的核心就是相位累加器，利用它来产生信号递增的相位信息，整个 DDS 系统在统一的参考时钟下工作，每个时钟周期相位累加器作加法运算一次。加法运算的步进越大，相应合成的相位值变化越快，输出信号的频率也就越高。对于幅值归一化的正弦波信号的瞬时幅值完全由瞬时相位来决定，因为 $w = d\Phi t /$ ，所以相位变化越快，信号的频率越高。ROM 表完成将累加器相位信息转换为幅值信息的功能。再由 D / A 完成数字抽样信号到连续时域信号的转换，D / A 输出的台阶信号再经低通滤波器平滑可以得到精确的连续正弦信号波形。

相位累加器利用 Nbit 二进制加法器的模溢出特性来模拟理想正弦波的 2π 相位周期。相位累加器输出和 ROM 输出可分别理解为理想正弦波相位信号和时域波形的时钟抽样。

用相位累加器输出的数据作为波形存储器的相位取样地址，这样就可以把存储在波形存储器内的波形抽样值(二进制编码)经查找表查出，完成相位到幅值转换。波形存储器的输出送到 D / A 转换器，D / A 转换器将数字量形式的波形幅值转换成所要求合成频率的模拟量形式信号。低通滤波器用于滤除不需要的取样分量，以便输出频谱纯净的正弦波信号。假设，相位累加器字长为 N，DDS 控制时钟频率为 f_c ，时钟周期为 $T_c = 1/f_c$ ，频率控制字为 K。系统工作时，累加器的单个时钟周期的增量值为 $\Delta\varphi = K * 2\pi / 2^n$ ，相应角频率为

$\omega = \Delta\varphi / \Delta t = \Delta\omega / T_c = 2\pi * K * 2^n * f_c$ ，所以 DDS 的输出频率为

$f_{dds} = \omega / 2\pi = K * f_c / 2^n$ ，DDS 输出频率步进间隔为 $\Delta f_{dds} = f_c / 2^n$ 。

因 DDS 输出信号是对正弦波的抽样合成的，所以应满足 Niquist 定理要求，即 $f_0 D_s \leq f_d / 2$ ，也就是要求 $K < 2^{n-1}$ ，根据频谱性能要求，一般取 $f_{dds} \leq 0.4 f_c$ 。当 DDS 相位累加器采用 32 位字长，时钟频率为 30MHz 时，它的输出频率间隔可达 $\Delta f_{dds} = f_c / 2^n = 30 * 10^6 / 2^{32} \approx 10 \text{mHz}$ 。可见，DDS 基于累加器相位控制方式给它带来微步进的优势。DDS 频率合成器具有以下优点：

(1)频率分辨率高,输出频点多,可达 $2N$ 个频点(假设 DDS 相位累加器的字长是 N);(2)频率切换速度快,可达 μs 量级;(3)频率切换时相位连续;(4)可以输出宽带正交信号;(5)输出相位噪声低,对参考频率源的相位噪声有改善作用;(6)可以产生任意波形;(7)全数字化实现,便于集成,体积小,重量轻。

但 DDS 也有比较明显的缺点:

(1)输出信号的杂散比较大;(2)输出信号的带宽受到限制。DDS 输出杂散比较大,这是由于信号合成过程中的相位截断误差、D/A 转换器的截断误差和 D/A 转换器的非线性造成的。当然随着技术的发展,这些问题正在逐步得到解决。如通过增长波形 ROM 的长度也减小相位截断误差;通过增加波形 ROM 的字长和 D/A 转换器的精度以减小 D/A 量化误差等。在比较新的 DDS 芯片中普遍都采用了 12bit 的 D/A 转换器。当然一味靠增加波形 ROM 的深度和字长的方法来减小杂散对性能的提高总是有限的。已有研究在对 DDS 输出的频谱做了大量的分析后,总结出了误差的领域分布规律建立了误差模型,在分析 DDS 频谱特性的基础上又提出了一些降低杂散功率的方法;可以通过采样的方法降低带内误差功率,可以通过随机抖动法提高无杂散动态范围,在 D/A 转换器的低位上加扰打破 DDS 输出的周期性,从而把周期性的杂散分量打散使之均匀化。

2 设计方案

2.1 总体设计思路

(1)设计步骤

此设计将按模块式实现，据任务书要求，设计总共分四大步份完成：

- 1.产生波形（三种波形：方波、三角波和矩形波）信号；
- 2.波形选择；
- 3.频率控制；
- 4.幅度控制。

2 设计思想

利用 Verilog HDL 编程，依据基本数字电路模块原理进行整合。系统各部分所需工作时钟信号由输入系统时钟信号经分频得到，系统时钟输入端应满足输入脉冲信号的要求。具备幅度和频率可调功能，幅度通过两个按键可以增减调节，频率控制模块则是一个简易的计数器，控制步径为 100HZ 的可调频率，达到设计课题所要求的输出波形频率可调及幅度可调功能。幅度可调功能由于比较简单，可以在 FPGA 外部利用硬件电路实现。总体设计框图如下图 1 所示：

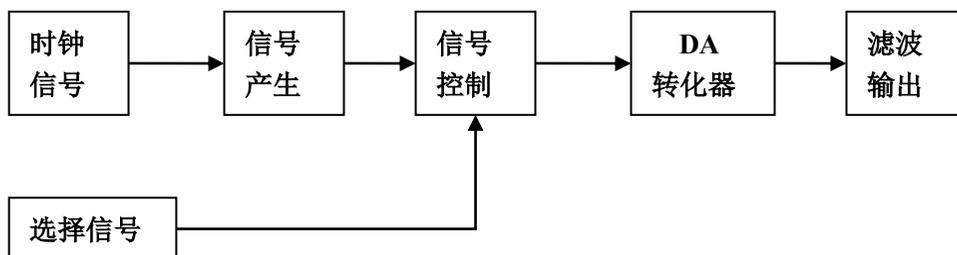


图 2.1 系统总体框图

2.2 方案论证

2.2.1 方案一

采用 DDS（直接数字频率合成器）来设计，设计总体框图如图 2 所示。在设计界里众所周知，DDS 器件采用高速数字电路和高速 D/A 转换技术，具有频率转换时间短、频率分辨率高、频率稳定度高、输出信号频率和相位可快速程控切换等优点，所以，我们可以利用 DDS 具有很好的相位控制和幅度控制功能，另外其数据采样功能也是极具精确和完善的，它可以产生较为精确的任何有规则波形信号，可以实现对信号进行全数字式调制。用 FPGA 和 DDS 实现信号调制，既克服了传统的方法实现带来的缺点，若采用它来编程设计，必定会事半功倍，且使设计趋于理想状态。但鉴于 DDS 的占用 ROM 空间较大，我们设计时就必须考虑到所用期间的 ROM 空间是否够用，结合我选用的 Cyclone II 系列的 EP2C5Q208C8N 器件所提供的 26 个 M4KRAM 存储模块，可以高达 1.1Mbits 的存储单元。应该可以满足本次设计的需要。

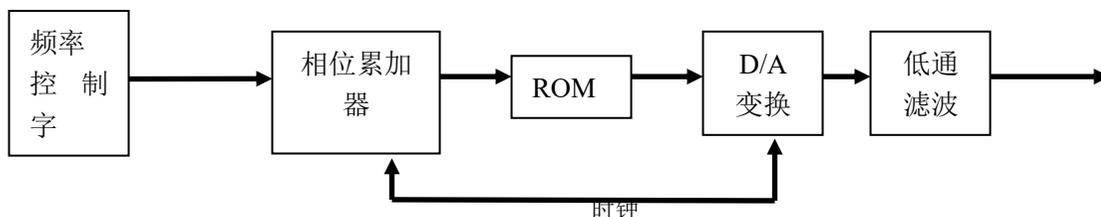


图 2.2 DDS 与 FPGA 总体设计图

2.2.2 方案二

采用振荡器频率合成方案。具体方案如下：首先通过频率合成技术产生所需要频率的方波，通过积分电路就可以得到同频率的三角波，再经过滤波器就可以得到正弦波。其优点是工作频率可望做得很高，也可以达到很高的频率分辨率；缺点是使用的滤波器要求通带可变，实现很难，高低频率比不可能做得很高。

2.2.3 方案三

采用 VHDL 语言来编程,然后下载文件到 FPGA 来实现。VHDL 语言是电子设计领域的主流硬件描述语言,具有很强的电路描述和建模能力,能从多个层次对数字系统进行建模和描述,从而大大降低了硬件设计任务,提高了设计效率和可靠性,要比模拟电路快得多。该方案是利用 FPGA 具有的静态可重复编程和动态在系统重构的特性,使得硬件的功能可以像软件一样通过编程来修改,极大地提高了电子系统设计的灵活性和通用性,而且大大缩短了系统的开发周期。

2.3 方案确定

由上述三个方案对比,采用第一种方案:

使用直接数字信号合成技术(DDS),将三种波形的数据存储在 FPGA 配置的 ROM 中,通过 FPGA 软件扫描方式将波形数据读出传输给 DAC0832 产生波形输出。这种方法在软、硬件电路设计上都简单,且与我们的设计思路紧密结合。

幅度控制部分在设计中目前有两种方法:其一是通过软件将存储的波形数据进行统一比例的缩放在输出给 DAC0832,这样就可以实现幅度的调节。第二种方法是通过外部电位器调节 DAC0832 参考电压的值来调节输出幅度。第一中方法调节精确但是除法器肯定会占用大量的 FPGA 内部资源,造成不必要的开销。而使用电位器调节 DAC0832 的参考电压的方法简单,但是调节精度很难达到 0.1V 的步进值。鉴于现在还出于理论分析时期不易决定何种方法更好,此点暂时待定等试验阶段在做确认。

波形组合如果采用分开式模块实现,也必将导致占用大量的资源,而且模块设计复杂度提高,只要采用重复调用一个模块的设计方法,既可以降低资源的占用率,也使得设计更加灵活且有针对性。此信号发生器的特点及功能集成度高,因采取整体模块式设计,在此也考虑到实际应用中,万一 FPGA 的逻辑门数量不够,特准备了一套备用方案。

备用方案:将波形数据存放在外部 SDRAM 中,SDRAM 的存储容量大,且可重复使用,可以很好的解决内存不够使用的情况。

综合以上涉及方面的分析，因此本次课题采用此方案进行设计。

3 硬件电路设计

3.1 硬件设计注意事项

此次设计利用所购买的 FPGA 核心开发板，并且通过面包板搭建 DA 电路与滤波电路，实现系统功能，能在良好的环境中正常工作。但是实际制作硬件电路时，就应考虑到干扰所带来的负面影响。为了避免干扰信号影响系统正常工作，未分配功能的 FPGA 引脚必须接地，在 FPGA 器件的电源端必须并连一 0.01uF 的退耦电容，在所有的输入引脚上串联 100 欧姆左右电阻减弱干扰信号影响，如有必要还应在输入端设置上拉或下拉电阻。

3.2 DA 电路

本设计采用 DAC0832 作为为输出 DA 电路，并采用直通方式(控制总线片选信号 CS，读写使能信号 WR,传输控制信号 XFER，输入所存使能信号 ILE 直接接地)，以得到最大的转换速度，电路示意图如下：

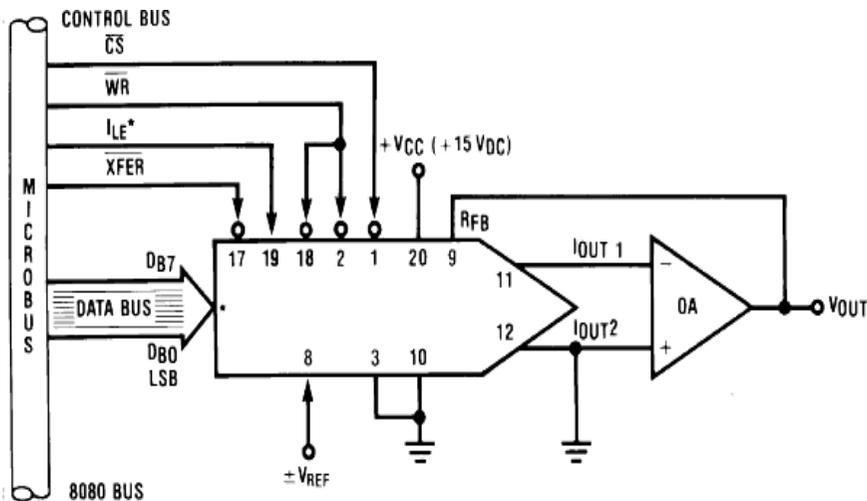


图 3.1DAC0832 电路连接图

DAC0832 简介:

DAC0832 是 8 分辨率的 D/A 转换集成芯片。与微处理器完全兼容。这个 DA 芯片以其价格低廉、接口简单、转换控制容易等优点,在单片机应用系统中得到广泛的应用。D/A 转换器由 8 位输入锁存器、8 位 DAC 寄存器、8 位 D/A 转换电路及转换控制电路构成。

主要特点:

- 分辨率为 8 位;
- 电流稳定时间 1 μ s;
- 可单缓冲、双缓冲或直接数字输入;
- 只需在满量程下调整其线性度;
- 单一电源供电 (+5V~+15V);
- 低功耗, 20mW。

DAC0832 结构:

D0~D7: 8 位数据输入线, TTL 电平, 有效时间应大于 90ns(否则锁存器的数据会出错);

ILE: 数据锁存允许控制信号输入线, 高电平有效;

CS: 片选信号输入线 (选通数据锁存器), 低电平有效;

WR1: 数据锁存器写选通输入线, 负脉冲 (脉宽应大于 500ns) 有效。由 ILE、CS、WR1 的逻辑组合产生 LE1, 当 LE1 为高电平时, 数据锁存器状态随输入数据线变换, LE1 的负跳变时将输入数据锁存;

XFER: 数据传输控制信号输入线, 低电平有效, 负脉冲 (脉宽应大于 500ns) 有效;

WR2: DAC 寄存器选通输入线, 负脉冲 (脉宽应大于 500ns) 有效。由 WR2、XFER 的逻辑组合产生 LE2, 当 LE2 为高电平时, DAC 寄存器的输出随寄存器的输入而变化, LE2 的负跳变时将数据锁存器的内容打入 DAC 寄存器并开始 D/A 转换。

IOUT1: 电流输出端 1, 其值随 DAC 寄存器的内容线性变化;

IOUT2: 电流输出端 2, 其值与 IOUT1 值之和为一常数;

Rfb: 反馈信号输入线, 改变 Rfb 端外接电阻值可调整转换满量程精度;

Vcc: 电源输入端, Vcc 的范围为+5V~+15V;

VREF: 基准电压输入线, VREF 的范围为-10V~+10V;

AGND: 模拟信号地

DGND: 数字信号地

3.3 滤波电路

DA 输出的滤波器电路可选用有源滤波器与无源滤波器。下面即那个对两种滤波器简单介绍一下：

1. 有源滤波器：有源电力滤波器（APF）是一种用于动态抑制谐波、补偿无功的新型电力电子装置，它能够对大小和频率都变化的谐波以及变化的无功进行补偿，之所以称为有源，顾名思义该装置需要提供电源（用以补偿主电路的谐波），其应用可克服 LC 滤波器等传统的谐波抑制和无功补偿方法的缺点（传统的只能固定补偿），实现了动态跟踪补偿，而且可以既补谐波又补无功；三相电路瞬时无功功率理论是 APF 发展的主要基础理论；APF 有并联型和串联型两种，前者用的多；并联有源滤波器主要是治理电流谐波，串联有源滤波器主要是治理电压谐波等引起的问题。有源滤波器同无源滤波器比较，治理效果好，主要可以同时滤除多次及高次谐波，不会引起谐振，但是价位相对高！

2. 无源滤波器：无源滤波器，又称 LC 滤波器，是利用电感、电容和电阻的组合设计构成的滤波电路，可滤除某一次或多次谐波，最普通易于采用的无源滤波器结构是将电感与电容串联，可对主要次谐波（3、5、7）构成低阻抗旁路；单调谐滤波器、双调谐滤波器、高通滤波器都属于无源滤波器。

因为本设计中要求幅度 5V 内可调节，但是无缘滤波器会造成信号的衰减，因此本设计中采用有源滤波器。

设计 D/A 输出后，通过滤波电路、输出缓冲电路，使信号平滑且具有负载能力。输出频率小于 1MHZ，为保证 1MHZ 频带内输出幅度平坦，又要尽可能抑制谐波和低频噪声,选用宽带运放 OPA227，用 TI 提供的滤波器设计软件 FilterPro 设计出截止频率为 1MHZ,为了保证通带的平坦度与阻带的截止特性,这里选用 4 阶巴特沃斯低通滤波器。

滤波器原理如下图：

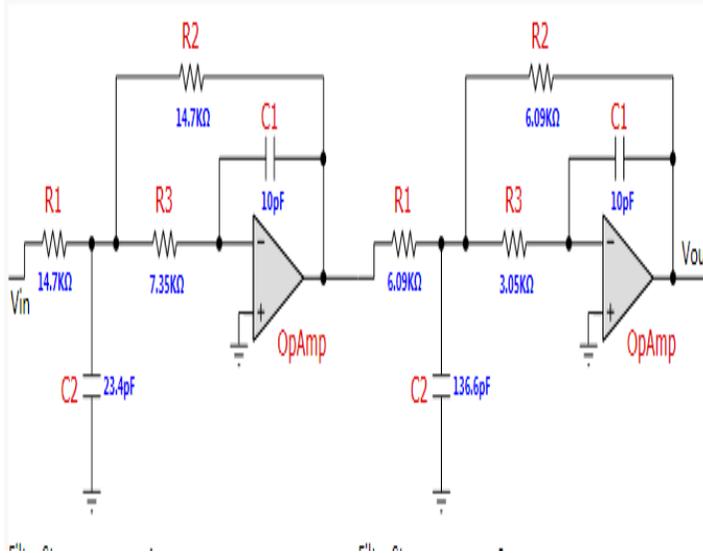


图 3.2 滤波器电路图

OPA27 是一种宽带低噪声高精度运放，带宽可达到 8MHZ,是完全可以满足本次设计的要求的。

滤波器频谱特性如下图：

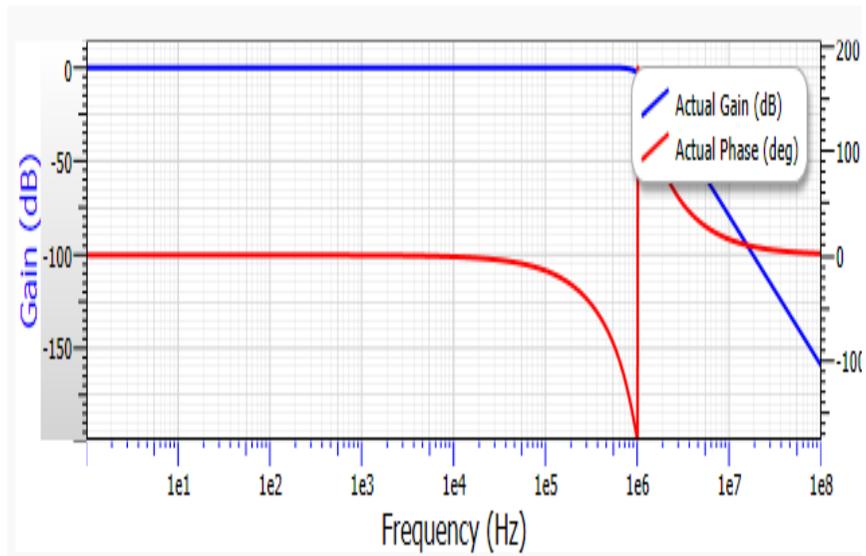


图 3.3 滤波器幅频特性

3.4 硬件电路实现

本设计使用 FPGA 最小系统板与外接面包板焊接的 DA 电路与滤波电路组成。

FPGA 最小系统板原理图如下：

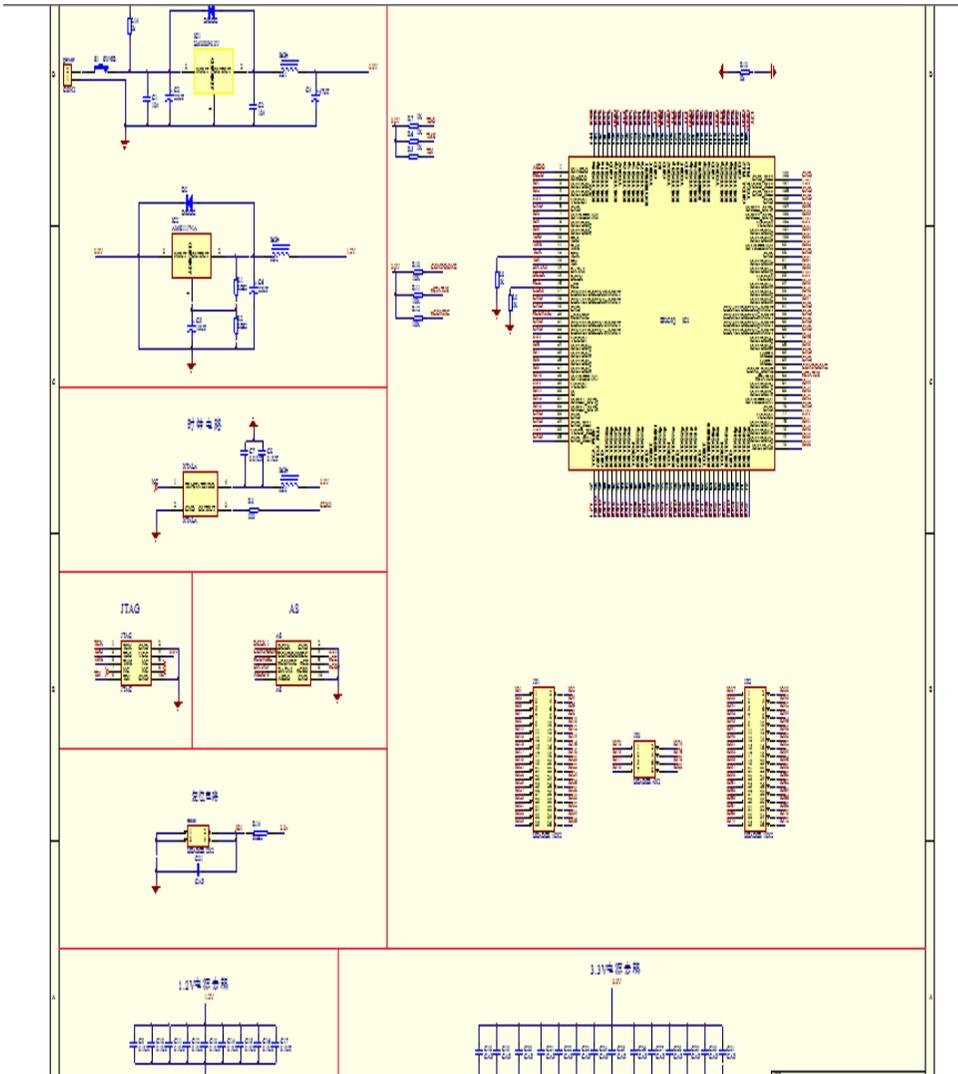


图 3.4 FPGA 最小系统版

4 软件设计

4.1 波形产生模块

本设计用 verilog 语言根据 niqustc 采样定理，对波形 ROM 进行扫描，分别产生正弦波、三角波和矩形波。以下介绍各种常用信号的原理。

4.1.1 正弦波

1.设计思想

正弦波发生分为两个步骤，即正弦波幅值采样存储和正弦波波形的还原输出。幅值采样是将一个周期正弦波进行 256 等分，如图 3 所示，将 256 个采样点进行量化处理，量化值 $y=127.5+127.5*\sin(n*\pi/128)$ ，将 256 点量化值存入存储器。正弦波形的产生是通过循环反复将存储器中的 256 点采样值通过 DAC0832 进行还原输出，得到幅值正比于 256 点采样值的正弦波。

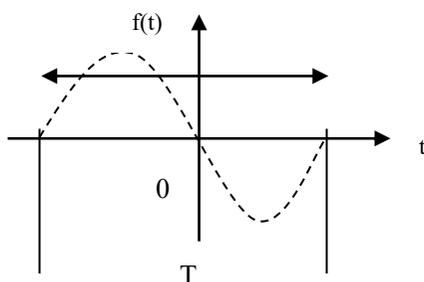


图 4.1 正弦波采样

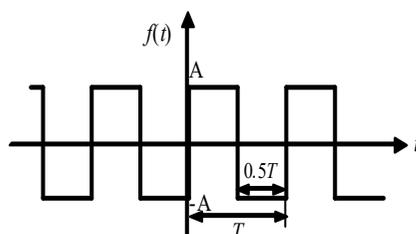


图 4.2 方波采样图

2 正弦波数据获取

1. 通过 MATLAB 获取正弦信号数据

MATLAB 程序如下:

```
x=0:255
```

```
y=127.5+127.5*sin(x*pi/128)
```

$y = \text{round}(y)$

$\text{plot}(x, y)$

波形数据如下:

Quartus II - C:/Users/tommy/Desktop/毕业设计/quartus/Graduate_design - G

Addr	+0	+1	+2	+3	+4	+5	+6	+7
0	128	131	134	137	140	143	146	149
8	152	155	158	162	165	167	170	173
16	176	179	182	185	188	190	193	196
24	198	201	203	206	208	211	213	125
32	218	220	222	224	226	228	230	232
40	234	235	237	238	240	241	243	244
48	0245	246	248	249	250	250	251	252
56	253	253	254	254	254	255	255	255
64	255	255	255	255	254	254	254	253
72	253	252	251	250	250	249	248	246
80	245	244	243	241	240	238	237	235
88	234	232	230	228	226	224	222	220
96	218	215	213	211	208	206	203	201
104	198	196	193	190	188	185	182	179
112	176	173	170	167	165	162	158	155
120	0152	149	146	143	140	137	134	131
128	128	124	121	118	115	112	109	106
136	104	100	97	93	90	88	85	82
144	79	76	73	70	67	65	62	59
152	57	54	52	49	47	44	42	40
160	37	35	33	31	29	27	25	23
168	21	20	18	15	14	12	11	11
176	10	9	67	6	5	5	4	3
184	2	1	1	1	1	0	0	0
192	0	0	0	0	1	1	1	2
200	2	3	4	5	5	6	7	9
208	10	11	12	14	15	17	18	20
216	21	23	25	27	29	31	33	35
224	37	40	42	44	47	49	52	54
232	57	59	62	65	67	70	73	76
240	79	82	85	88	90	93	97	100
248	103	106	109	112	115	118	121	124

图 4.3 正弦波 ROM 表

2. 定制 ROM 如下所示:

将以上数据保存到 Quartus 的存储器出事实文件当中, 就可用 Quartus 的宏模块定制向导 MegaWizard Pule_IN 通过简单的几步操作

定制一个 ROM 来保存正弦波数据。
MegaWizard Plug-In Manager 定制界面如下：

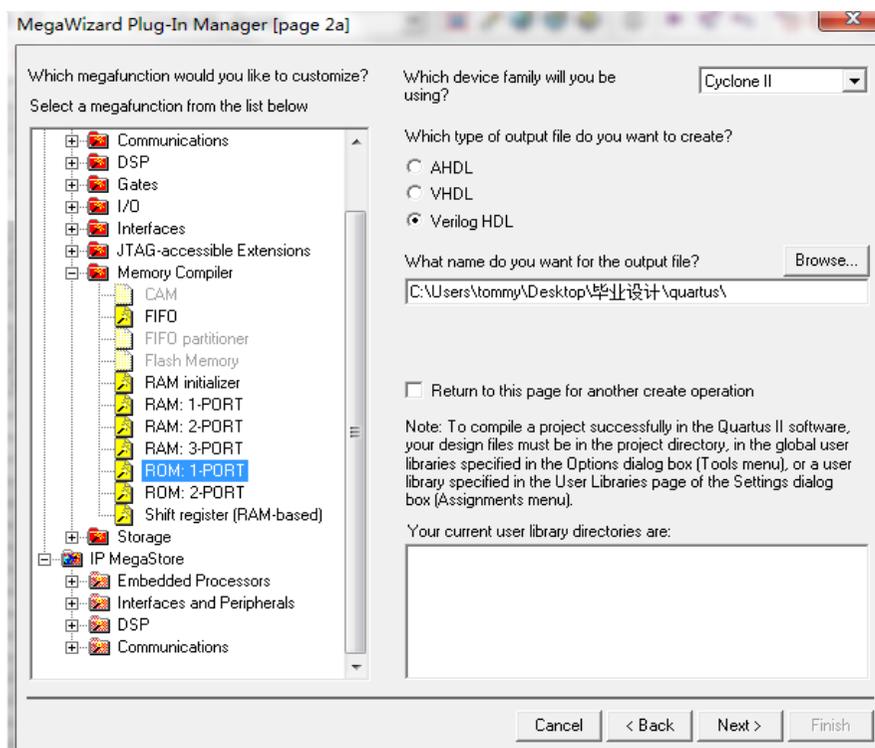


图 4.4 定制 ROM 示意图

ROM 生成元程序见附件 1。

4.1.2 矩形波

1 设计思路

矩形波的实现较之正弦波发生简单，由于矩形波是两个电平值间的交替变换，因此波形采样值的预存只要有二个不同的数值就行了，为了使矩形波发生的频率控制与正弦波一致，我们采用与正弦波相同的原理，采用 256 个采样值扫描输出来实现，循环反复将存储器中的 256 点采样值通过 DAC0832 进行还原输出，得到幅值正比于点采样值的矩形波。采样图如图 4 所示。

其波形数据产生与正弦波一致，这里就不在赘述了。

4.1.3 三角波

1 设计思路

由于三角波是线性的，比较简单就可以产生，如果最低电压 DA 参考量为 0，最高电压参考量是 255，我们同样设置 256 个点采样深度(128 点为上升阶段另外 128 点为下降阶段)，那么根据它的公式 $255/128 \approx 2$ ，每个点的电压参考量只要依次加 2 就可以得到，一个波形所以采用简单的加减算法就可实现，如图 5 所示三角波的采样图：

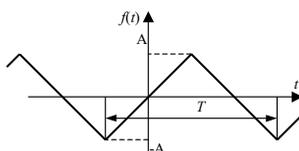


图 12-19 三角波

图 4.5 三角波采样图

4.2 频率控制模块

本课题要求频率控制是在 100HZ-1MHZ，步进为 100HZ。在本题设计中只需借助 FPGA 便可完成。根据上面对 DDS 理论的分析者里采用 25 为的相位累加器，并且本次设计的系统版时钟为 50MHZ，则有分析可知频率分辨率位 $\Delta f_{dds} = 50MHZ / 2^{25} \approx 1.4901HZ$ ，本次要求频率为 100HZ 步进可调，分析后知道 $1.4901 * 67 = 99.837HZ \approx 100HZ$ ，只要对频率控制字 $K=67$ 送入相位累加器进行累加或累减就可以实现以 100HZ 为步进的频率调节。本模块只要再设计两个按键来判断是对累加器加还是对累加器减就可以实现频率控制，并且 $K=67$ 也是满足 $K \leq 2^{n-1}$ 采样要求。

频率控制模块 verilog 程序如下：

```
module Fword(clk,key1,key2,Fword);
input key1,key2;
input clk;
output [24:0] Fword;
```

```

reg [24:0] Fword;

always@(posedge clk)
begin
if(!key1)
Fword=Fword+25'd67;
else if(!key2)
Fword=Fword-25'd67;
End
endmodule
    
```

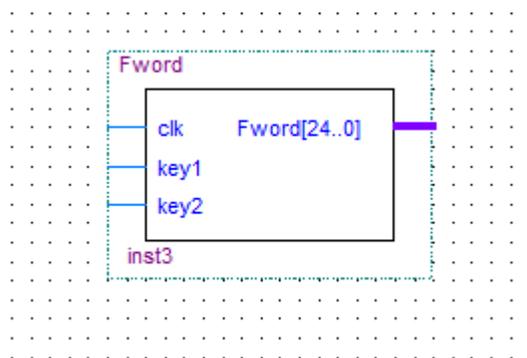


图 4.6 频率控制模块电路符号

4.3 相位累加模块

相位累加模块有一个累加器与地址产生器组成，相位累加器将输入的频率控制字累加并有地址产生器将累加结果的搞 8 位提取当做波形 ROM 的地址查询相应的数据。

在时序电路中为了提高加法器的速度。流水线结构是一种常用的设计方法。累加器采用流水线结构来实现，简单而言，就是把一个位数很长的加法，拆成 N 个位数较短的加法，在 N 个时钟周期内做完，然后输出结果，N 就是流水线的级数。采用流水线结构以后，由于加法器的字长变短了，对于 FPGA 来讲，加法器字长变短，对于提高工作频率是十分有帮助的。当然，流水线结构的使用，并不能无限制地提高电路的工作速度，这是因为，流水线结构是一种用电路规模换取工作速度的设计方法，提高工作速度的代价是电路设计的复杂化。流水线结构累加器要比普通的累加器结构复杂得多，由于累加不在一个时钟周期内完成，内部需要大量的寄存器保存中间变量。随着流水级数的提高，电路复杂程度将大大增加，当电路的复杂程度达到一定量级的时候，流水线所带来的性能改进，和电路本身由于结构复杂所带来的性能下降相抵消的时候，流水线结构就不再具有提高电路工作频率的作用了。对于不同的器件米说，采用多少级流水对性能的提升比较大要通过仿真试验才能得到一个比较肯定的值。

下面是基于流水线技术的加法器与寄存器结合在一起的相位累加

器设计。40 位相位累加器的四级流水线设计，加法器采用 5 级锁存，4 级加法，最前的一级实现 10 位数的相加，后面 3 级加法器实现 10 位数与一个进位的相加，整个加法器的速度由 10 位加法器决定，Verilog HDL 源程序见附件 2。

相位累加模块电路符号如下：

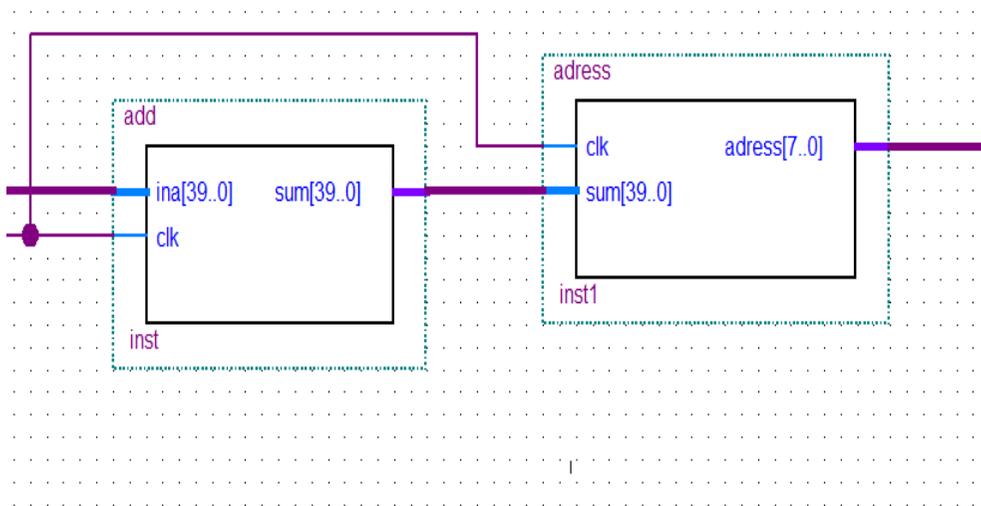


图 4.7 相位累加器电路符号

4.3 选择波形模块

波形选择模块使用一个 4 通道的数据选择器来对正弦波，方波，三角波的选择。

通过两个外部拨码开关来对三种波形选择。

当拨码开关为 00 时，输出正弦波。

当拨码开关为 01 时，输出正方波。

当拨码开关为 10 时，输出三角弦波。

波形选择模块程序如下：

```
module mux4t1(clk,sel,data1,data2,data3,outdata);
input clk;
input [1:0]sel;
input [7:0] data1,data2,data3;
output [7:0] outdata;
```

```

reg [7:0] outdata;
always@(posedge clk)
begin
case (sel)
2'b00 : outdata=data1;
2'b01 : outdata=data2;
2'b10 : outdata=data3;
default :outdata=8'd0;
endcase
end
endmodule

```

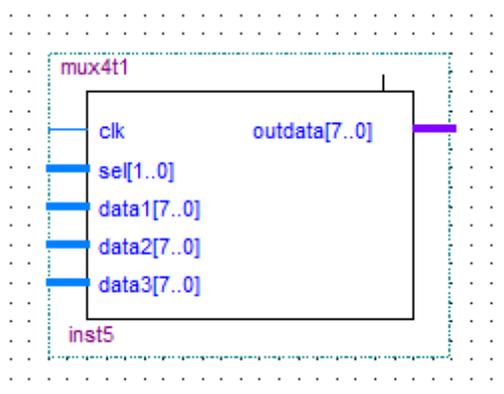


图 4.8 波形选择模块电路符号

4.4 幅度控制模块

应为本设计采用全数字 DDS 方案，所以幅度控制也采用数字处理方式。由于本次设计才用的 DAC0832 是一个 8 位的数模转换器，且参考电压定为 5 V，则其电压分辨率位 $\Delta v=5V/255=19.6\text{mv}$ ，可见 $19.6*5=98\text{mv}\approx 0.1V$ ，要实现对电压的 0.1v 可调，只要对波形数据 ROM 的数据进行对以 5 为步进的缩放即可。例如要对 5V 电压下调 0.1V，只要对 5V 的 ROM 数据 255 减 5 即可，即 $(255-5)*19.6=4.9V$ 。因此本设计还是通过两个按键来判断是对电压上调还是下调。

幅度调节模块程序：

```

module Amcontrol(clk,key1,key2,Romdata,Amplitude);
input clk;
input key1,key2;
input [7:0]Romdata;
output [7:0] Amplitude;
reg [7:0] Romdata_temp1;
reg [7:0] X1;

```

```

reg [7:0] X2;
always@(posedge clk)
begin
if (Romdata>=255)
X1=8'd0;
else if(!key1)
X1=X1+8'd1;
end
always@(posedge clk)
begin
if (Romdata<=0)
X2=8'd0;
else if(!key2)
X2=X2+8'd1;
end
always@(*)
begin
if(!key1||!key2)
Romdata_temp1=Romdata+8'd5*(X1-X2);
end
assign Amplitude=Romdata_temp1;
endmodule

```

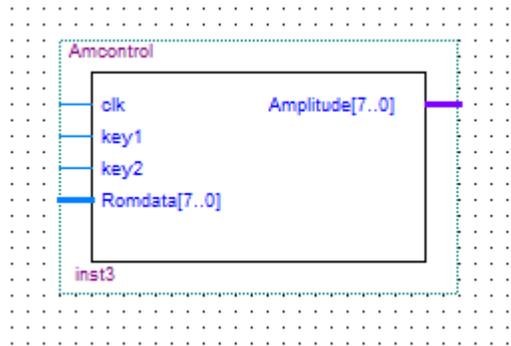


图 4.9 幅度控制模块电路符号

频率控制部分程序流程图如下：

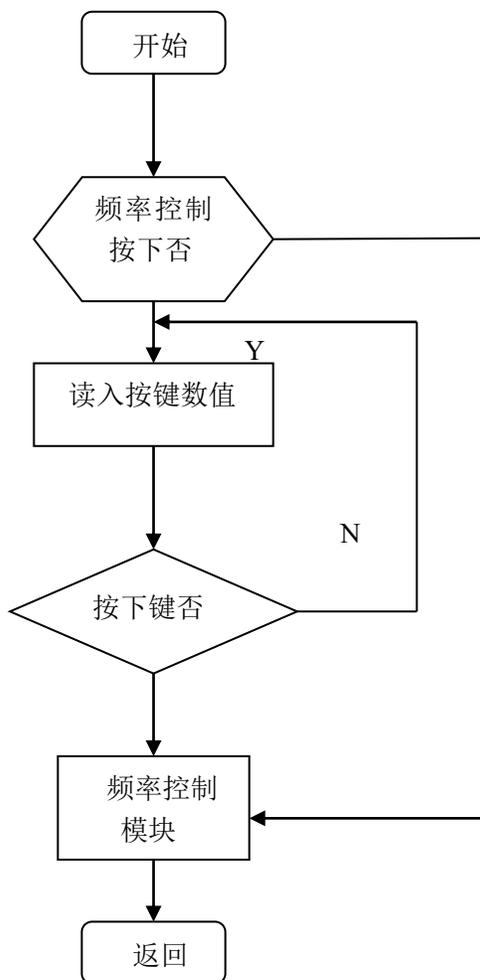


图 4.10 频率控制模块流程图

根据上文中提到的 DDS 的原理可以知道，要控制系统输出频率的大小只需要相应的改变频率控制字的大小便可以实现。频率控制模块通过两个按键实现对对频率控制字的调节。当检测到按键按下时，对频率控制字进行固定值得加减，从而调节 DDS 输出的频率大小的固定调节。具体过程如图 4.10 所示。

波形选择部分流程图如下：

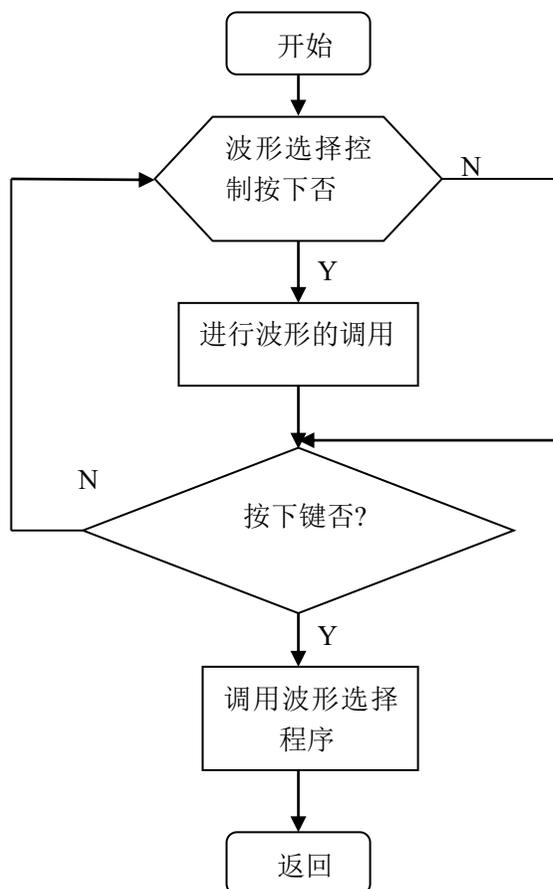


图 4.11 波形选择模块流程图

根据上文中提到的幅度控制原理可以知道，要控制系统输出幅度的大小只需要相应的改变从波形 ROM 中读出的幅度参考值的大小，对于 8 为的 DAC0832 来说每次按键时对输出的数值进行以 5 为步进的缩放便可实现以 0.1V 为步进的调节，幅度控制模块通过两个按键实现对对幅度的调节。当检测到按键按下时，对频率控制字进行固定值得加减，从而调节 DDS 输出的幅度参考值并送入 DAC 进行转化。具体过程如图 4.11 所示。

软件设计总成：

此次设计采用自低向上设计，通过先完成频率控制模块，累加器模块，波形数据 ROM，数据选择器，幅度控制器无个部分，并且生成相应的电路符号，原理图输入法实现个部分的连接。

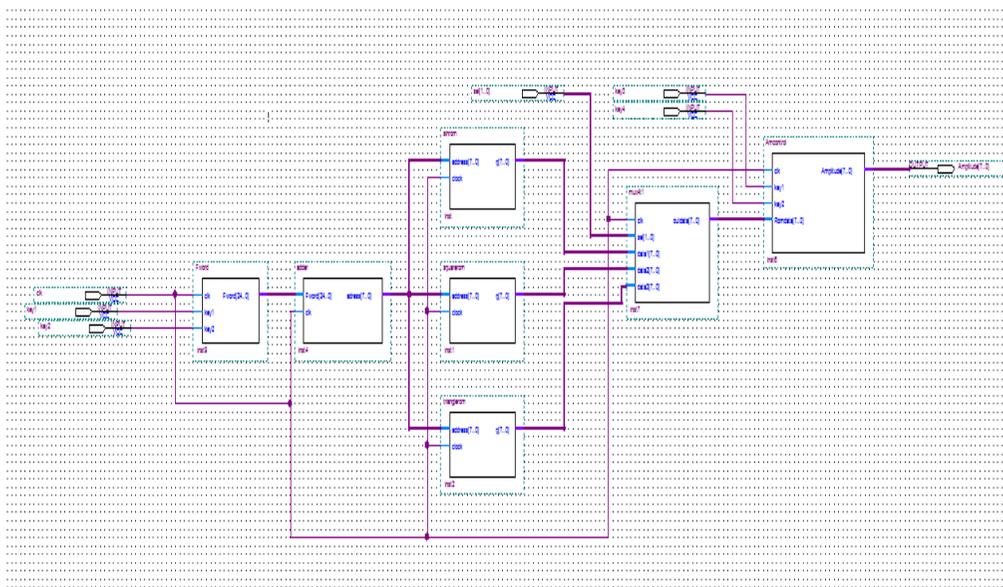


图 4.12 顶层连接图

FPGA 设计资源报告：

Flow Summary	
Flow Status	Successful - Wed May 02 15:38:50 2012
Quartus II Version	9.0 Build 132_02/25/2009 SJ Full Version
Revision Name	Graduate_design
Top-level Entity Name	Graduate_design
Family	Cyclone II
Device	EP2C5Q208C8
Timing Models	Final
Met timing requirements	Yes
Total logic elements	78 / 4,608 (2 %)
Total combinational functions	78 / 4,608 (2 %)
Dedicated logic registers	86 / 4,608 (1 %)
Total registers	86
Total pins	15 / 142 (11 %)
Total virtual pins	0
Total memory bits	6,144 / 119,808 (5 %)
Embedded Multiplier 9-bit elements	0 / 26 (0 %)
Total PLLs	0 / 2 (0 %)

图 4.13 FPGA 资源消耗图

5 调 试

5.1 设计及仿真调试使用设备

示波器:Hitachi V-1060, 万用表, FPGA 开发板, PC 机各一台。

5.2 调试方法

根据方案设计的要求, 调试过程共分三大部分: 硬件调试、软件调试和综合调试。电路用 FPGA/CPLD 实验箱按模块调试, 各模块逐个调试通过后再进行综合调试。

5.2.1 硬件调试

(1)在做控制电路的调试时, 分析输入输出, 可以发现时序与仿真结果是否有出入, 便于找出硬件电路中的故障。再调试 D/A 转换是否正常。

(2) 为提高电路抗干扰性能, 我们采取了一些抗干扰措施。如接线线尽量短, 减少交叉, 每个芯片的电源与地之间都接有去扰电容, 数字地与模拟地分开。实践证明, 这些措施对消除某些引脚上的“毛刺”及高频噪声起到了很好的效果。

(3) 运算放大器的选择 由于输出频率达到上千赫兹, 因此对放大器的带宽有一定要求。所以, 在调试滤波电路和缓冲输出电路时, 都选择了高速宽带运放。

5.2.2 软件调试

本系统的软件功能强大, 运用 VHDL 语言来编写, 先在 MAX+PLUS II 对所编的模块一一进行仿真, 排除了语法的错误编写和设计逻辑思维的错误, 当仿真完确认程序没问题时, 再直接下载到 FPGA 芯片, 用 FPGA 实验箱进行调试。采取的就是自底向上的调试方法, 即先单独调试好每一个模块, 然后再连接成一个完整的系统再调试。

5.2.3 综合调试

FPGA 系统的软硬件设计是紧密相连的，将软件编译综合并且通过下载对 FPGA 进行适配，运行中的 FPGA 是由其内部的硬件结构控制读出数据，从而产生波形。因此，如果在软硬件都基本调通的情况下，系统的软硬件综合调试难度不是很大。

5.3 调试结果

5.3.1 软件仿真结果及分析

1. 相位累加模块仿真

(1) 仿真波形图如图 11 所示：

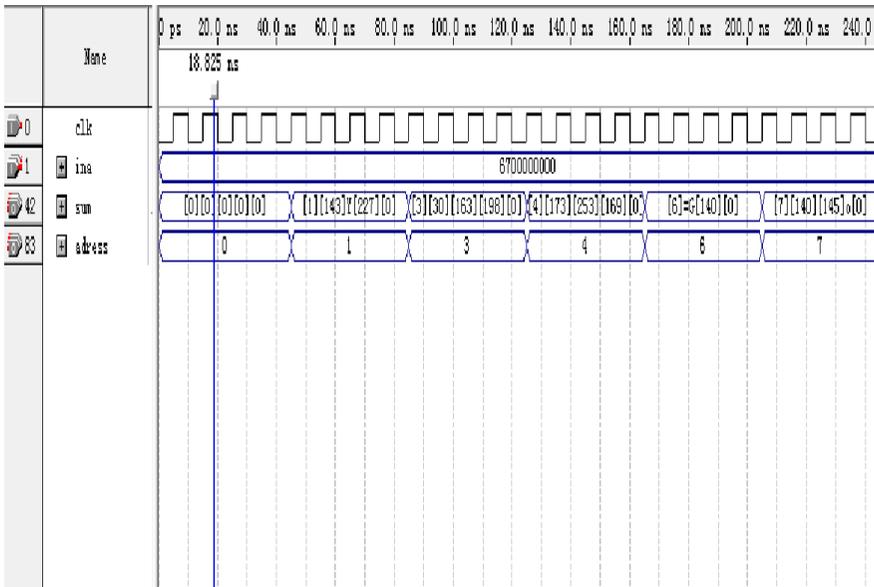


图 5.1 相位累加模块仿真图

CLK：时钟输入；

Ina：频率控制字输入；

Sum：频率控制字累加结果；

(2) 仿真结果分析

如图所示相位累加模块对输入的频率控制字累加并寄存，调节频率控制字便可调节输出频率的大小，实现频率的调节。

2.幅度控制模块仿真

(1) 仿真波形图如图 22 所示:

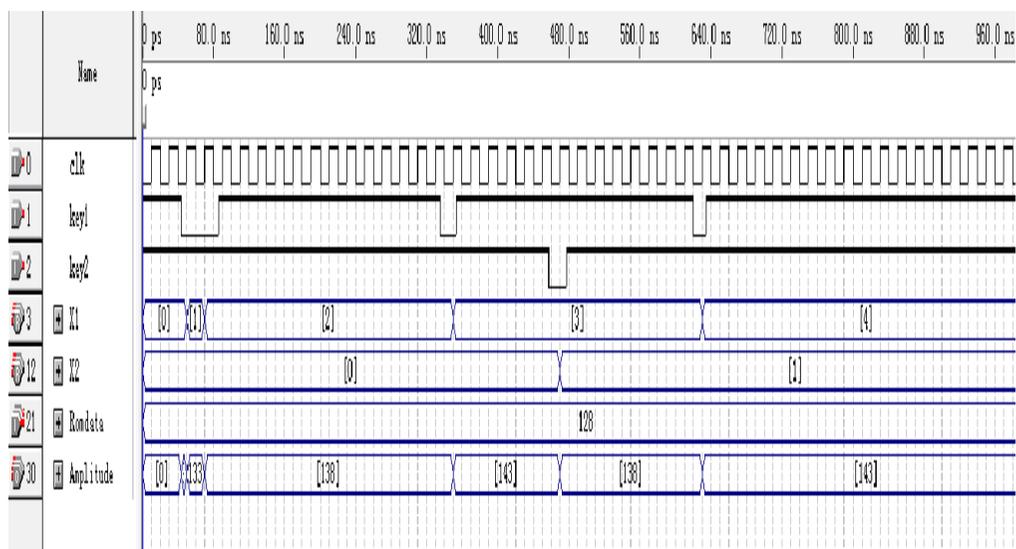


图 5.2 幅度控制模块仿真图

CLK: 时钟输入;

Key1: 幅度增加;

Key2: 幅度减少;

Romdata: 波形 ROM 输出幅值;

Amplitude: 输出波形幅值;

(2) :仿真结果分析:

如图 12 所示当设定波形 Rom 输出为 128 时, 当按下 key1 (幅度增加调节按键, 按下一次增加 0.1V, 对应幅值参数增加 5) 一次后波形幅度增加 5, 变为 133, 再按下一次变为 138, 按下第三次后变为 143, 此时按下 key2 (幅度减少调节按键, 按下一次减少 0.1V, 对应幅值参数减少 5) 后幅值减少 5 变为 138, 此后再按一次 key1 幅值增加 5, 变为 143。以上仿真说明幅度调节模块正确, 可以使用。

3.波形选择模块

(1) 仿真波形图如图 13 所示:

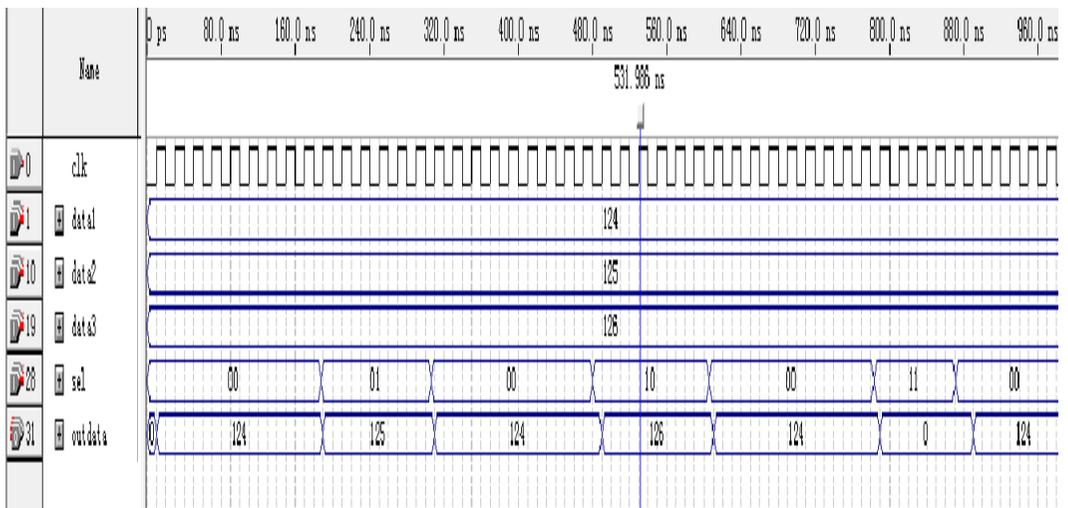


图 5.3 波形选择仿真图

CLK: 时钟输入;

Data1: 数据输入 1;

Data2: 数据输入 2;

Data3: 数据输入 3;

Sel: 选择信号输入;

Outdata: 输出信号;

(2) 仿真结果分析:

波形选择模块共有三个波形输入口, 分别对应正弦波, 方波, 三角波, 可知只要两位二进制数据便可满足选择三种波形的要求, 通过上图可见当选择信号分别为 00,01,10,是可对应选择 data1, data2, data3, 输出。从仿真可知, 波形选择模块可以正确选择对应波形。

4: 正弦波产生模块:

(1): 仿真波形图;

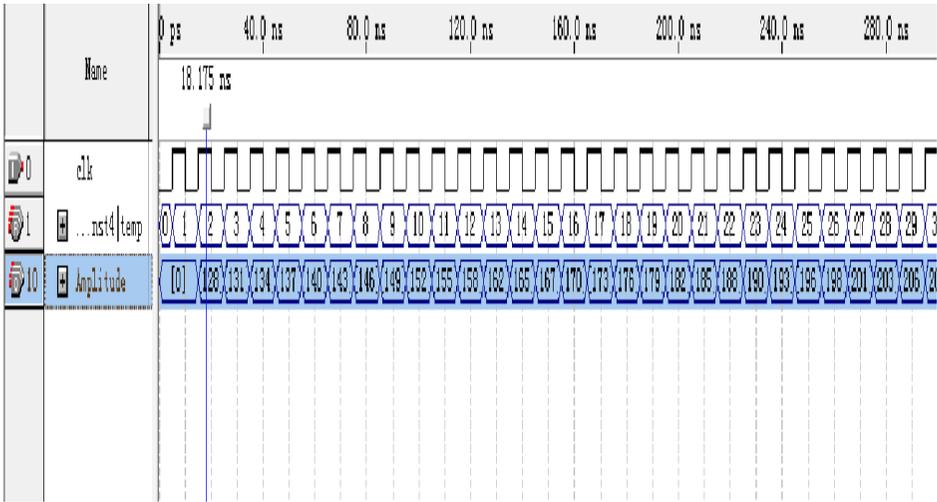


图 5.4 正弦波产生仿真图

Clk:时钟输入;

Temp: 地址寄存器;

Amplitude: 波形幅度输出;

(2) 仿真结果分析:

从仿真图课看出正弦波产生模块可以根据输入的地址，一次从正弦波 ROM 当中读出相应的幅度值，满足 DDS 原理的要求。

5: 三角波产生模块:

(1): 仿真波形图;

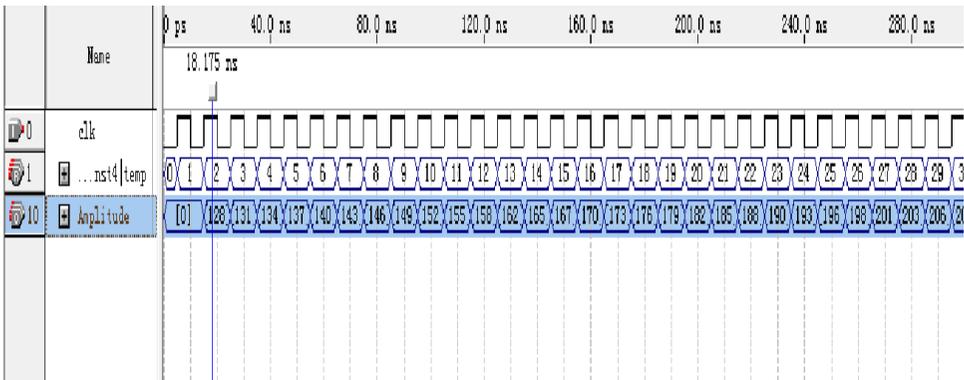


图 5.5 三角波产生仿真图

Clk:时钟输入;

Temp: 地址寄存器;

Amplitude: 波形幅度输出;

(2) 仿真结果分析:

从仿真图课看出正弦波产生模块可以根据输入的地址，一次从三角波 ROM 当中读出相应的幅度值，幅值均匀递增后在递减，满足三角波的幅度要求。

5: 方波波产生模块:

(1): 仿真波形图;

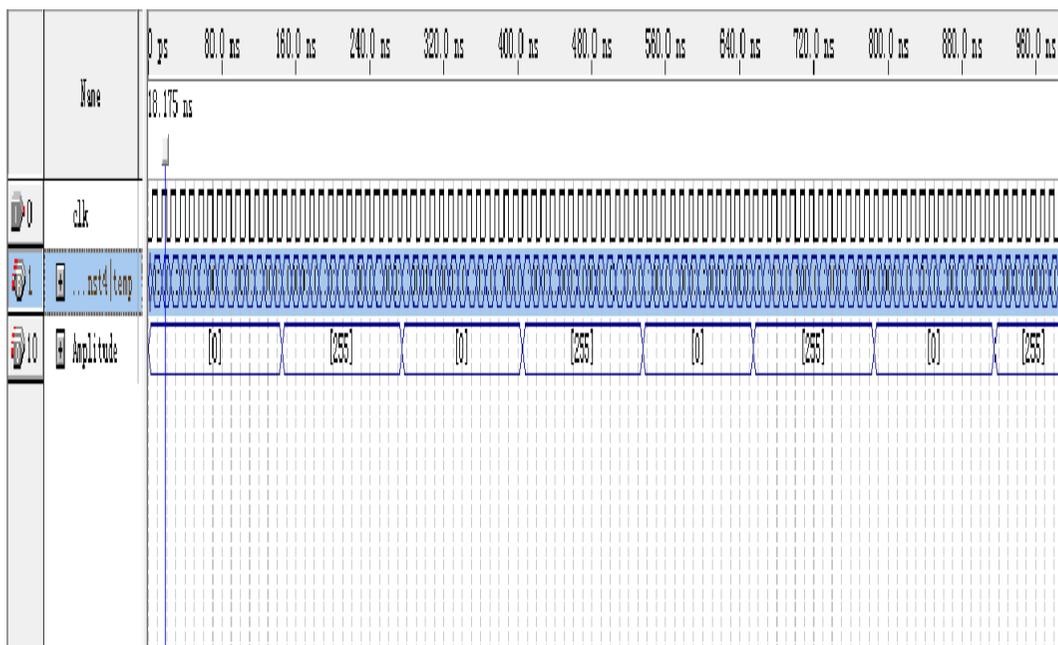


图 5.6 方波波产生仿真图

Clk:时钟输入;

Temp: 地址寄存器;

Amplitude: 波形幅度输出;

(2) 仿真结果分析:

从仿真图课看出方波产生模块可以根据输入的地址，一次从三角波 ROM 当中读出相应的幅度值，间隔输出 0 (对应电压 0V) 与 255 (对应电压 5V)。

5.3.2 综合调试结果

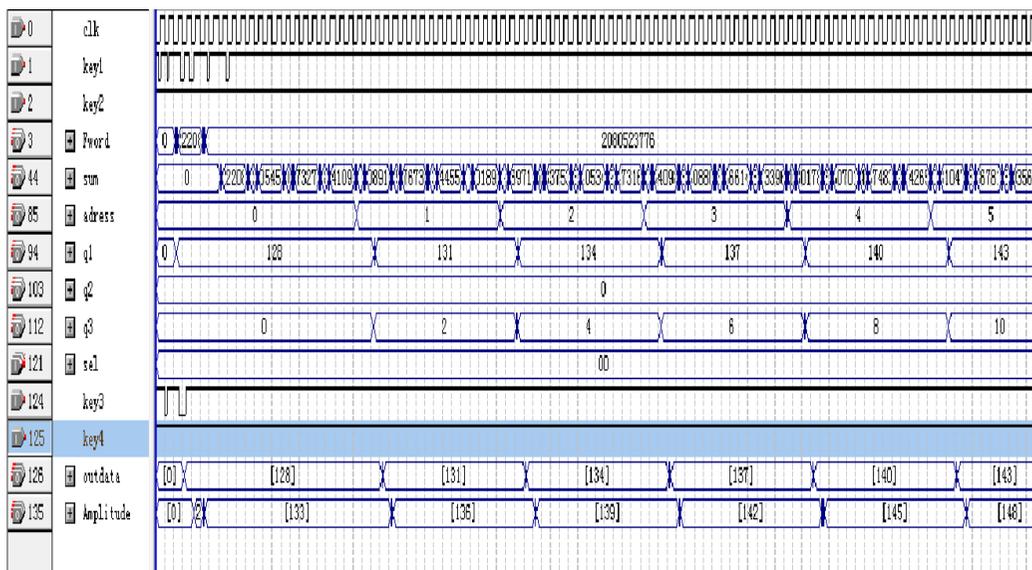


图 5.7 系统综合仿真图

Clk:时钟输入;

Key1,key2: 频率控制字控制按键;

Fword: 频率控制字输出;

Address: 波形地址;

sel: 波形选择控制信号;

key3, key4: 幅度控制按键;

Amplitude: 波形幅度输出;

(2) 仿真结果分析:

从仿真图可以看出系统检测到频率控制字按键按下后,会将预设的控制值输入到相位累加器,并由相位累加器对频率控制字进行累加,并截取波形 Rom 的地址,送到幅度控制模块进行缩放后输出。如图可见当 sel 为 00 时选择正弦波的波形输出,并且 key3 检测到按下一次,就将波形输出的值 outdata 加 5 之后再由 Amplitude 输出。由时序仿真图 5.7 可一验证本次设计方案正确,功能基本能够实现。

总结

通过此次设计，让我深深的感觉到自己所学知识真是非常的浅薄。面对电子技术日新月异的发展，利用 EDA 手段进行设计已成为不可阻挡的趋势。相对于传统至底向上的设计方式，自上而下的设计具有其显著的优越性。利用 EDA 设计软件辅助设计，方便快捷，减少了错误率的产生，缩短了产品的设计及上市周期，既减轻了设计工作量又满足了商业利益的需求。

该系统以 Cyclon 器件为核心部件，可利用软件编程实现了对 D/A 转换信号的处理。努力做到了线路简单、高性价比的特点，充分利用了软件编程，弥补了硬件元器件的不足。

在设计过程当中，遇到了软件操作不熟练，程序编写不规范等诸多问题，通过对问题的总结分析得出，应用软件的主要功能必须熟练操作，才能提高工作效率，需要规范操作的地方必须严格按照使用说明操作，避免由于软件使用不当造成的错误产生。程序的编写格式必须规范，模块、端口以及信号变量的命名应当反映实际意义，缩进格式工整明了，方便阅读理解，这样有利于程序的编写，有利于分析调试，也有利于程序的重复使用。

此次课题的设计已告一段落，在这次毕业设计过程中需要用一些不曾学过的东西时，就要去有针对性地查找资料，然后加以吸收利用，以提高自己的应用能力，而且还能增长自己见识，补充最新的专业知识，学会了一些编程方面的常用算法。作为一名电子专业的毕业生，我将会继续在新技术的道路上不断钻研、开拓进取。相信通过此次设计的锻炼，我对专业知识和技能的掌握将更加牢靠，在今后的工作和学习中，必将使我受益匪浅，取得应有的成绩。

致谢辞

在这里感谢指导老师石坚给我耐心的指导，在这次设计期间，碰到许多专业方面的难题，石老师都一一帮我解答，特别在教学繁忙的情况下，还为我们提供了许多宝贵的资料和意见，并帮我们作出了详细的分析，使我们更加顺利地完成此次毕业设计。通过这一个月多的时间，使我学到许多知识，明白了许多以前上课时无法理解的知识，还积累了一些较简单的问题解决方案。与此同时也感谢同组做课题设计的同学，在设计过程中，有许多东西我不懂，他们都耐心地给我讲解，给予我技术支援，帮助我解决了不少难题。另外由于该方面的设计水平有限，所以存在着许多缺陷和失误，恳请老师批评指正。

参考文献

- [1] 潘松, 黄继业.EDA 技术实用教程 (第二版), 北京科学出版社, 2005.2
- [2] 亿特科技, CPLD/FPGA 应用系统设计与产品开发, 人民邮电出版社, 2005.7
- [3] 李辉, PLD 与数字系统设计, 西安电子科技大学出版社, 2005.5
- [4] 王志鹏, 付丽琴, 可编程逻辑器件开发技术 MAX+PLUS II, 北京国防工业出版社, 2005.3
- [5] 王道先, VHDL 电路设计技术, 北京国防工业出版社, 2004.1
- [6] 赵不贿, 在系统可编程器件与开发技术, 机械工业出版社, 2001.6
- [7] 曾繁泰, EDA 工程的理论与实践, 电子工业出版社, 2004.5
- [8] 尹佳喜, 尹仕, 基于 CPLD 的三相多波形函数发生器设计
<http://www.pld.com.cn/application/a146.htm>, 2005-9-20。
- [9] 高淼, 基于 FPGA 的自治型 SPWM 波形发生器的设计
<http://www.gecmag.com/bdt03/10bdt9.asp>, 2005-10-17。
- [10] 李刚强, FPGA 设计中关键问题的研究
<http://www.edacn.net/viewarticle.php?id=27146>, 2005-10-15。

附件 1 ROM 生成元程序

ROM 生成元程序:

```
module sinrom (address, clock,q);
    input  [7:0] address;
    input   clock;
    output [7:0] q;
    wire [7:0] sub_wire0;
    wire [7:0] q = sub_wire0[7:0];
    altsyncramaltsyncram_component (
        .clock0 (clock),
        .address_a (address),
        .q_a (sub_wire0),
        .aclr0 (1'b0),
        .aclr1 (1'b0),
        .address_b (1'b1),
        .addressstall_a (1'b0),
        .addressstall_b (1'b0),
        .byteena_a (1'b1),
        .byteena_b (1'b1),
        .clock1 (1'b1),
        .clocken0 (1'b1),
        .clocken1 (1'b1),
        .clocken2 (1'b1),
        .clocken3 (1'b1),
        .data_a ({8{1'b1}}),
        .data_b (1'b1),
        .eccstatus (),
```

```
.q_b (),
.rden_a (1'b1),
.rden_b (1'b1),
.wren_a (1'b0),
.wren_b (1'b0));

defparam
altsyncram_component.clock_enable_input_a = "BYPASS",
altsyncram_component.clock_enable_output_a = "BYPASS",
altsyncram_component.init_file = "sinrom.mif",
altsyncram_component.intended_device_family = "Cyclone II",
altsyncram_component.lpm_hint =
"ENABLE_RUNTIME_MOD=NO",
altsyncram_component.lpm_type = "altsyncram",
altsyncram_component.numwords_a = 256,
altsyncram_component.operation_mode = "ROM",
altsyncram_component.outdata_aclr_a = "NONE",
altsyncram_component.outdata_reg_a = "CLOCK0",
altsyncram_component.widthad_a = 8,
altsyncram_component.width_a = 8,
altsyncram_component.width_byteena_a = 1;

endmodule
```

附件 2 40 位流水线加法器程序

```
module adder(sum,ina,clk);
output [39:0]sum;
input[39:0] ina;
input clk;
reg[39:0] tempa,inb,sum;
reg[40:0] tempc;
reg firstco,secondco,thirdco;
reg[9:0] firsts,thirda,thirdb;
reg[29:0] firsta,firstb, thirds;
reg[19:0] seconds, seconda,secondb;
always@(posedge clk)
begin
tempa=ina;
end
always@(posedge clk)
begin
{firstco,firsts}=tempa[9:0]+inb[9:0];
firsta=tempa[39:10];firstb=inb[39:10];
end
always@(posedge clk)
begin
{secondco,seconds}={firsta[9:0]+firstb[9:0]+firstco,firsts};
seconda=firsta[29:10];secondb=firstb[29:10];
end
always@(posedge clk)
```

```
begin
{thirdco,thirds}={seconda[9:0]+secondb[9:0]+secondco,seconds}
thirda=seconda[19:10];thirdb=secondb[19:10];
end
always@(posedge clk)
begin
tempc={thirda[9:0]+thirdb[9:0]+thirdco,thirds};
sum=tempc[39:0]; inb=tempc[39:0];
end
endmodule
```