



基于FPGA的MIPS32流水线 处理器的设计与实现

Design and Implementation of the MIPS32
Pipeline Processor Based on FPGA



参赛队员：许 佳，薛双百，许信辉

指导老师：张泽生，袁春风，黄宜华

南京大学计算机科学与技术系



报告内容

第一部分：项目设计概述

第二部分：详细设计与实现

第三部分：开发测试与结果分析

第四部分：总结与讨论



第一部分：项目设计概述

项目主要内容

基于Altera FPGA开发平台，设计和实现一个32位MIPS流水线处理器及系统



第一部分：项目设计概述

设计指导思想

独创性

基于**Altera FPGA**开发平台，从零开始完成一个较大规模处理器的设计实现，为**Altera**实验教学开发平台开辟一种新的应用模式，促进该平台的推广应用

实践性

结合专业课程的学习培养我们的实践能力和实际的计算机系统的设计能力

挑战性

设计兼容于**MIPS32** 指令集**2.62**版、超过**80**条指令的**5**级流水线处理器，作为本科阶段的学习，本项目的设计实现具有很大的难度和挑战性



第一部分：项目设计概述

设计原则

兼容性

充分考虑与业界标准的兼容性，在指令集层面上与**MIPS32 指令集2.62**兼容，在总线构架上采用**Wishbone**总线标准

完备性

与一般教学实验所实现的数条到数十条指令的处理器不同，设计一个覆盖**MIPS32 指令集2.62**版绝大多数指令的完备的指令集体系结构

可扩展性

在处理器和系统构架设计上充分考虑今后的功能和结构的扩充



第一部分：项目设计概述

设计目标

指令集体系结构

RISC结构，兼容于**MIPS32** 指令集**2.62**，分为：

算术和逻辑运算指令

分支转移指令

存取控制指令

数据操作指令

自陷指令

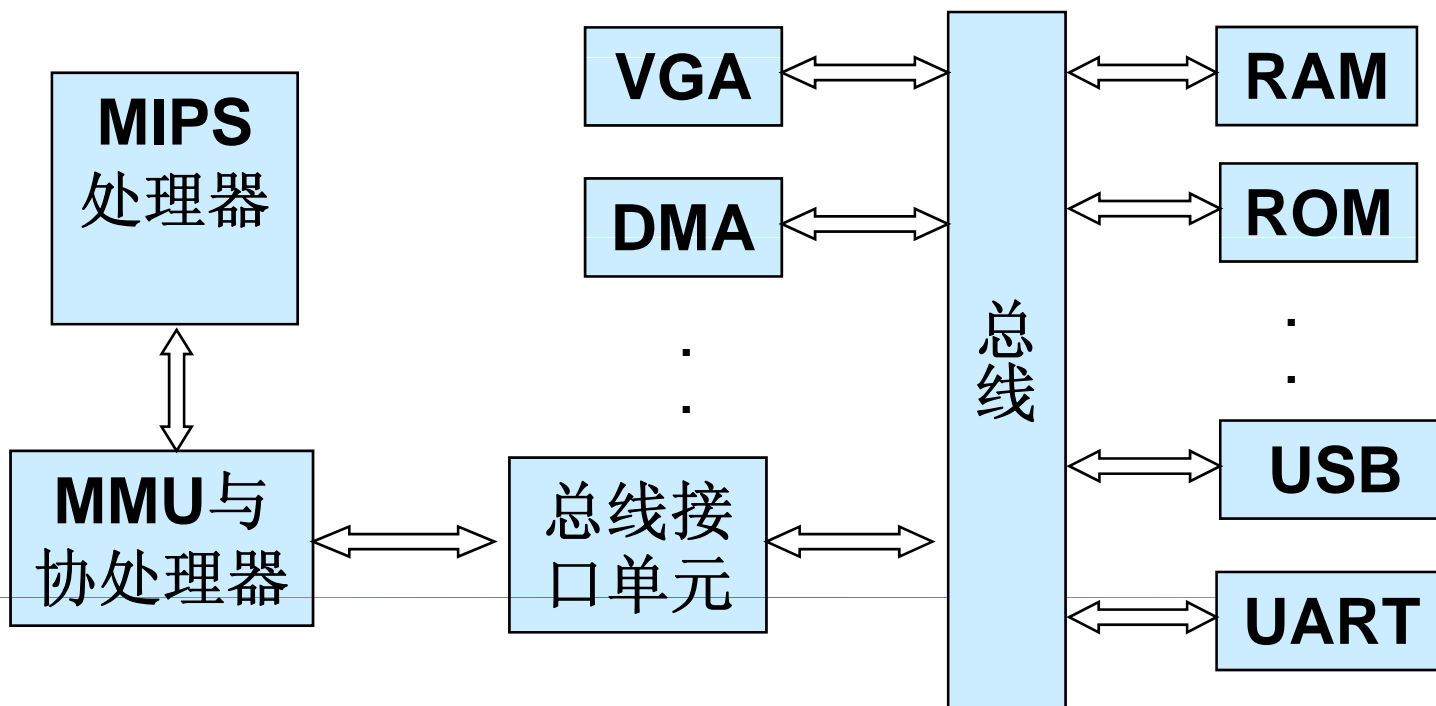


第一部分：项目设计概述

设计目标

处理器和系统总体构架

应用程序
编译/汇编程序
操作系统





第二部分：详细设计与实现

指令集设计

算术运算指令（19+4）

| | | | |
|--------------|---------------|--------------|--------------|
| add | 加法（带溢出位） | addi | 立即数加法（带溢出位） |
| addiu | 立即数加法（不带溢出位） | addu | 加法（不带溢出位） |
| clo | 计算前导一 | clz | 计算前导零 |
| madd | 乘加（有符号） | maddu | 乘加（无符号） |
| msub | 乘减（有符号） | msubu | 乘减（无符号） |
| mul | 乘法（结果写到通用寄存器） | mult | 乘法（有符号） |
| multu | 乘法（有符号） | slt | 小于置一（有符号） |
| slti | 立即数小于置一（有符号） | sltiu | 立即数小于置一（无符号） |
| sltu | 小于置一（无符号） | sub | 减法（有符号） |
| subu | 减法（无符号） | | |
| *div | 除法（有符号） | *divu | 除法（无符号） |
| *seb | 符号扩展字节 | *seh | 符号扩展半字 |



第二部分：详细设计与实现

指令集设计

逻辑运算指令（8）

| | | | |
|------------|-----------|-------------|-------|
| and | 与 | andi | 立即数与 |
| lui | 取立即数的高16位 | nor | 或非 |
| or | 或 | ori | 立即数或 |
| xor | 异或 | xori | 立即数异或 |

移位指令（6+2）

| | | | |
|--------------|------|---------------|--------|
| sll | 逻辑左移 | sllv | 逻辑左移变量 |
| sra | 算术右移 | srav | 算术右移变量 |
| srl | 逻辑右移 | srlv | 逻辑右移变量 |
| *rotr | 循环右移 | *rotrv | 循环右移变量 |



第二部分：详细设计与实现

指令集设计

转移指令（13+3）

| | | | |
|---------------|-----------|-----------------|------------------|
| bal | 转移并链接 | beq | 相等转移 |
| bgez | 大于等于零转移 | bgezal | 大于等于零转移并链接 |
| bgtz | 大于零转移 | blez | 小于等于零转移 |
| bltz | 小于零转移 | bltzal | 小于等于零转移并链接 |
| bne | 不相等转移 | j | 无条件跳转 |
| jal | 无条件跳转并链接 | jalr | 无条件跳转并链接寄存器 |
| jr | 寄存器跳转 | | |
| *B | 无条件转移 | *jalr.hb | 无条件跳转并链接到冒险阻塞寄存器 |
| *jr.hb | 冒险阻塞寄存器跳转 | | |



第二部分：详细设计与实现

指令集设计

存取指令（14+3）

| | | | |
|---------------|----------|--------------|----------|
| lb | 取字节（有符号） | lbu | 取字节（无符号） |
| lh | 取半字（有符号） | lhu | 取半字（无符号） |
| ll | 取链接字 | lw | 取字 |
| lwl | 取左半字 | lwr | 取右半字 |
| sb | 保存字节 | sc | 保存条件字 |
| sh | 保存半字 | sw | 保存字 |
| swl | 保存左半字 | swr | 保存右半字 |
| *pref | 预取 | *sync | 同步访存 |
| *synci | 同步缓存 | | |



第二部分：详细设计与实现

指令集设计

控制指令（1+3）

| | | | |
|---------------|--------|---------------|----------|
| nop | 空指令 | | |
| *ehb | 执行冒险阻塞 | *pause | 等待LL位来清除 |
| *ssnop | 超标量空指令 | | |

自陷指令（14）

| | | | |
|--------------|--------------|----------------|-----------|
| break | 跳出点 | syscall | 系统跳用 |
| teq | 相等自陷 | teqi | 立即数相等自陷 |
| tge | 大于等于自陷 | tgei | 立即数大于等于自陷 |
| tgeiu | 无符号立即数大于等于自陷 | tgeu | 无符号大于等于自陷 |
| tlr | 小于自陷 | tlri | 立即数小于自陷 |
| tlriu | 无符号立即数小于自陷 | tlru | 无符号小于自陷 |
| tne | 不等自陷 | tnei | 立即数不等自陷 |

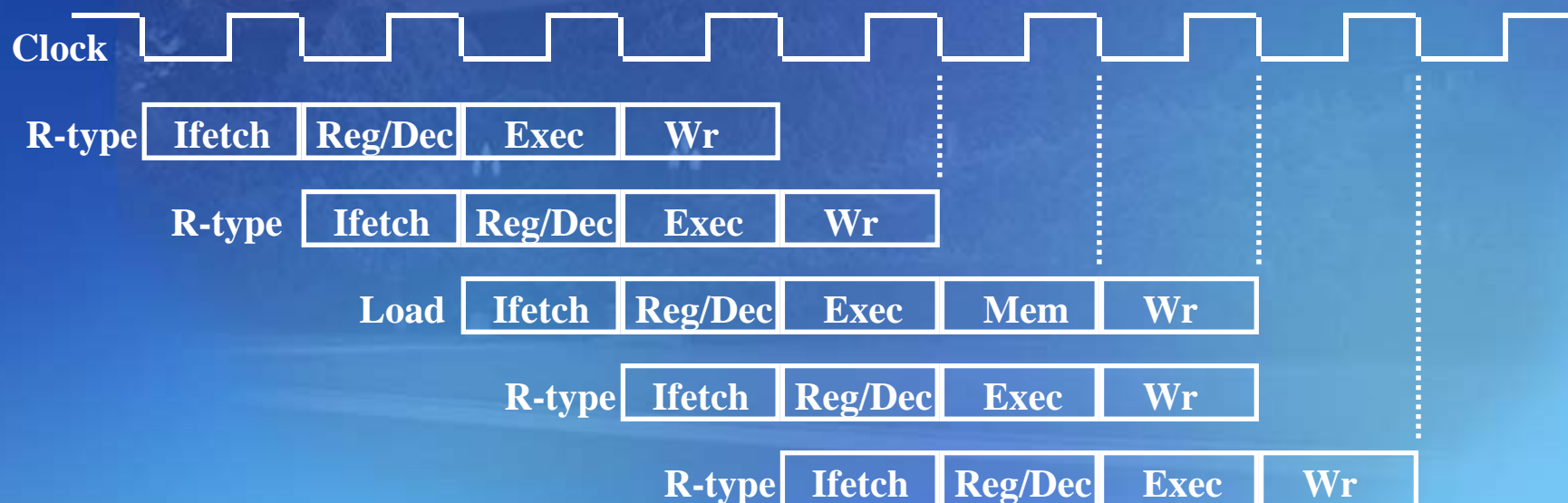


第二部分：详细设计与实现

流水线结构设计

采用5级流水线，分为：

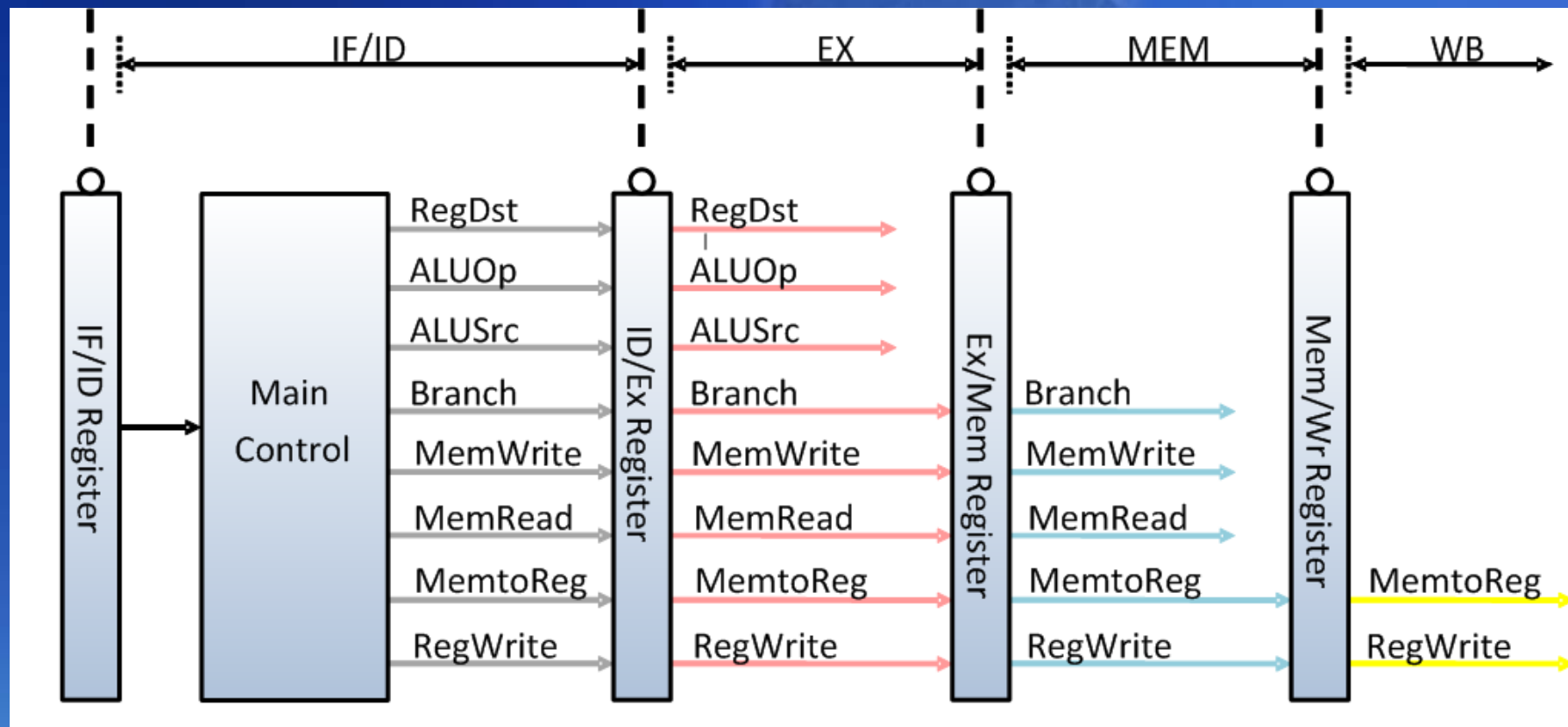
取指（IF），译码（ID），执行（EXE），
访存（MEM），写回（WB）





第二部分：详细设计与实现

流水线结构设计





第二部分：详细设计与实现

流水线冒险处理

三种冒险的处理

结构冒险，数据冒险，控制冒险

数据相关冒险的处理

采用转发技术解决指令执行不同阶段的数据相关

- 执行和访存阶段
- 执行和写回阶段
- 译码和写回阶段



第二部分：详细设计与实现

流水线冒险处理

其它冒险的检测处理

异常和中断

分支和跳转

Load-Use

乘法或除法指令

采用以下不同方法处理以保证后续指令的正确执行：

流水线停顿

指令冲刷



第二部分：详细设计与实现

异常与中断检测与处理

异常

MIPS标准定义的各阶段的异常都预留了接口，包括：

取指和访存阶段：如取指异常，Cache缺失

译码阶段：如非法指令，断点异常，系统调用

执行阶段：如溢出异常

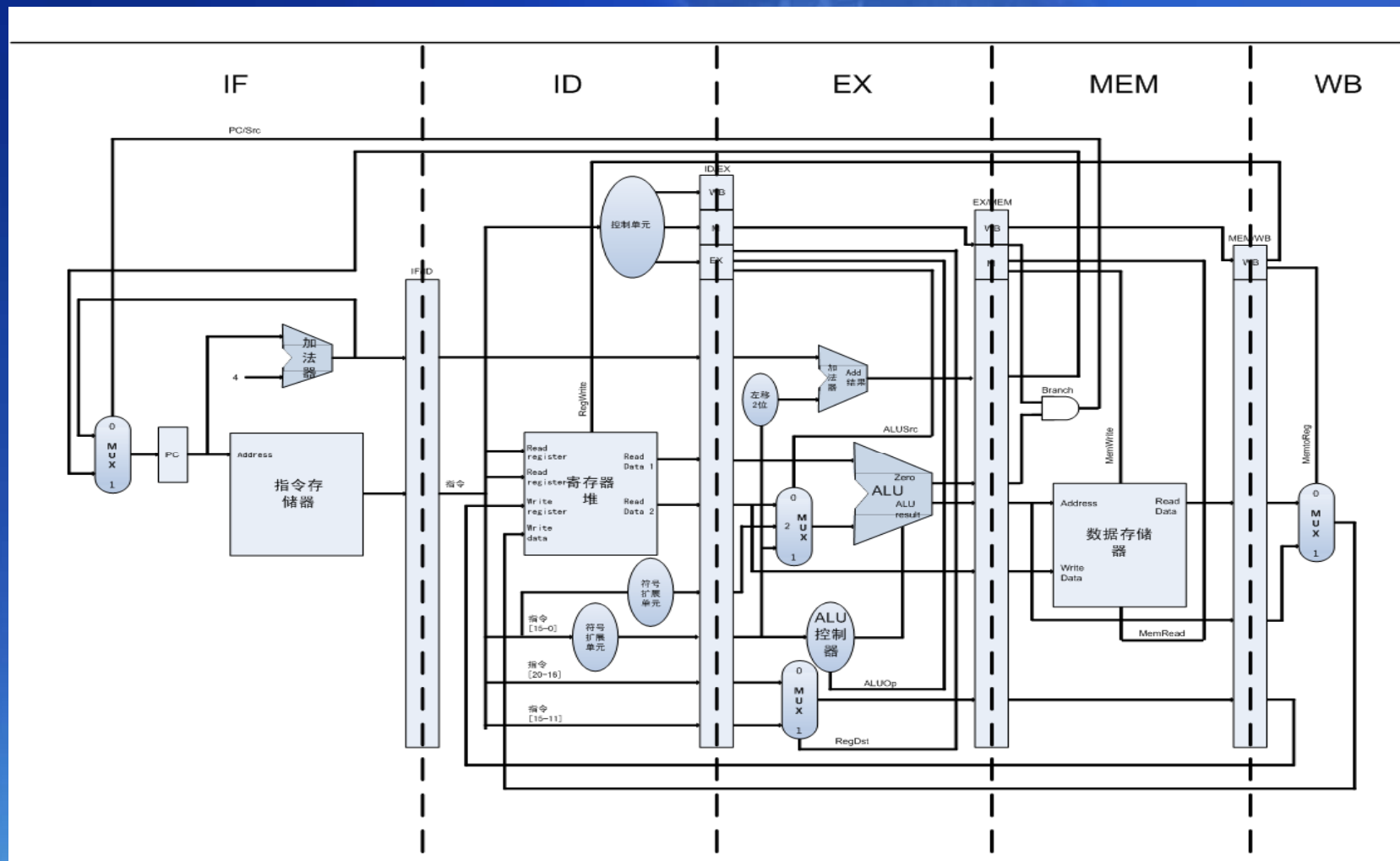
中断

外部中断请求



第二部分：详细设计与实现

处理器功能结构设计



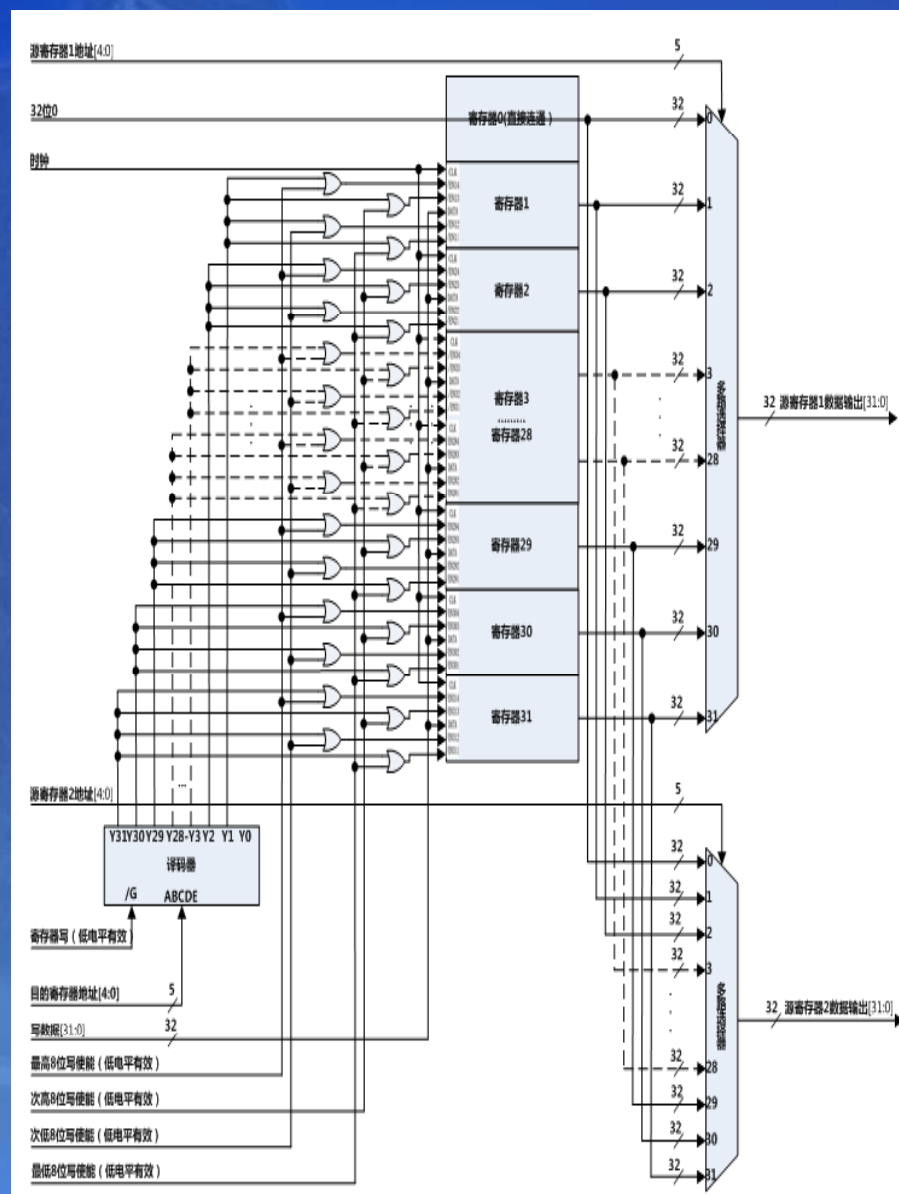


第二部分：详细设计与实现

处理器功能部件设计

寄存器组(32个x32位)

支持8/16/24/32位不同长度数据访问，适应所有指令对不同长度数据访问的处理





第二部分：详细设计与实现

处理器功能部件设计

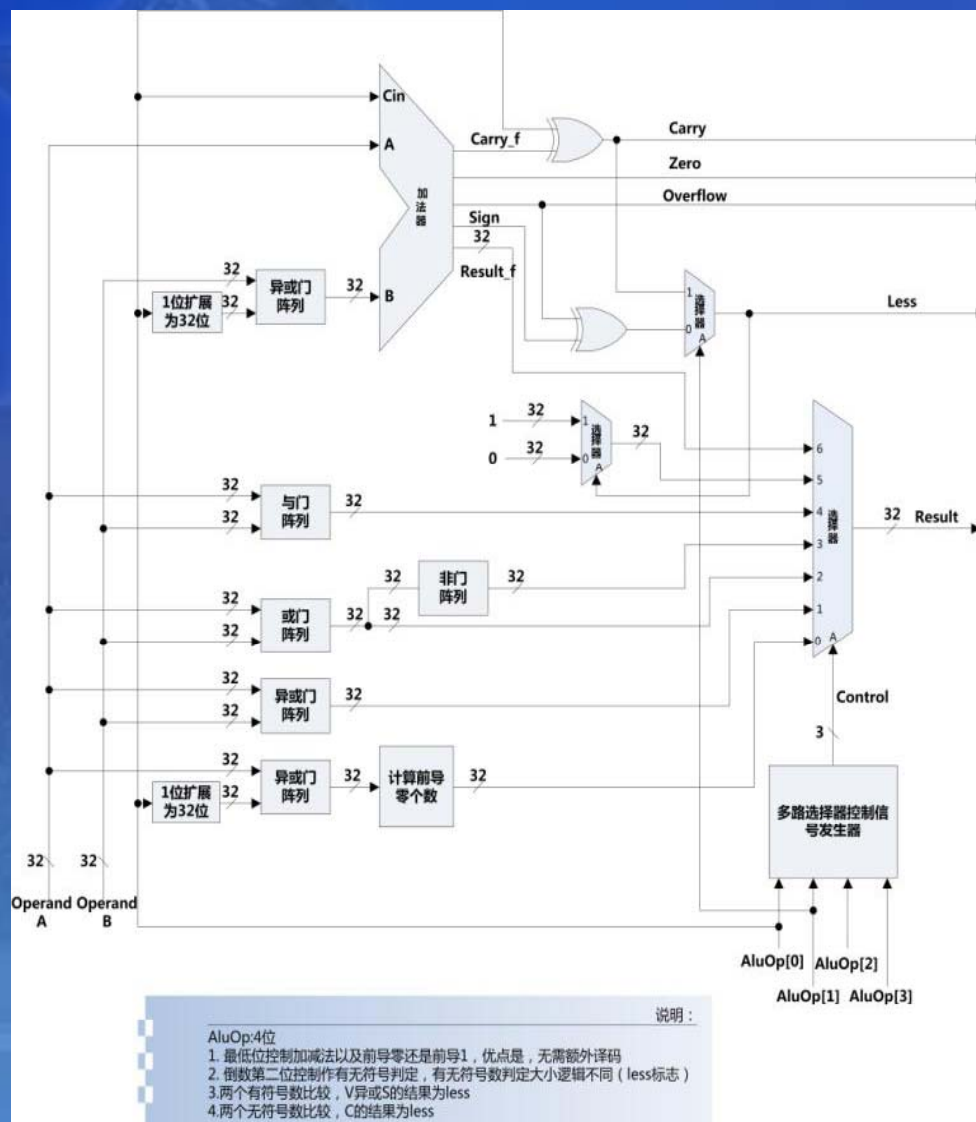
32位运算部件ALU

算术和逻辑运算

超前进位加法器

基于编码的快速

前导0/1计算





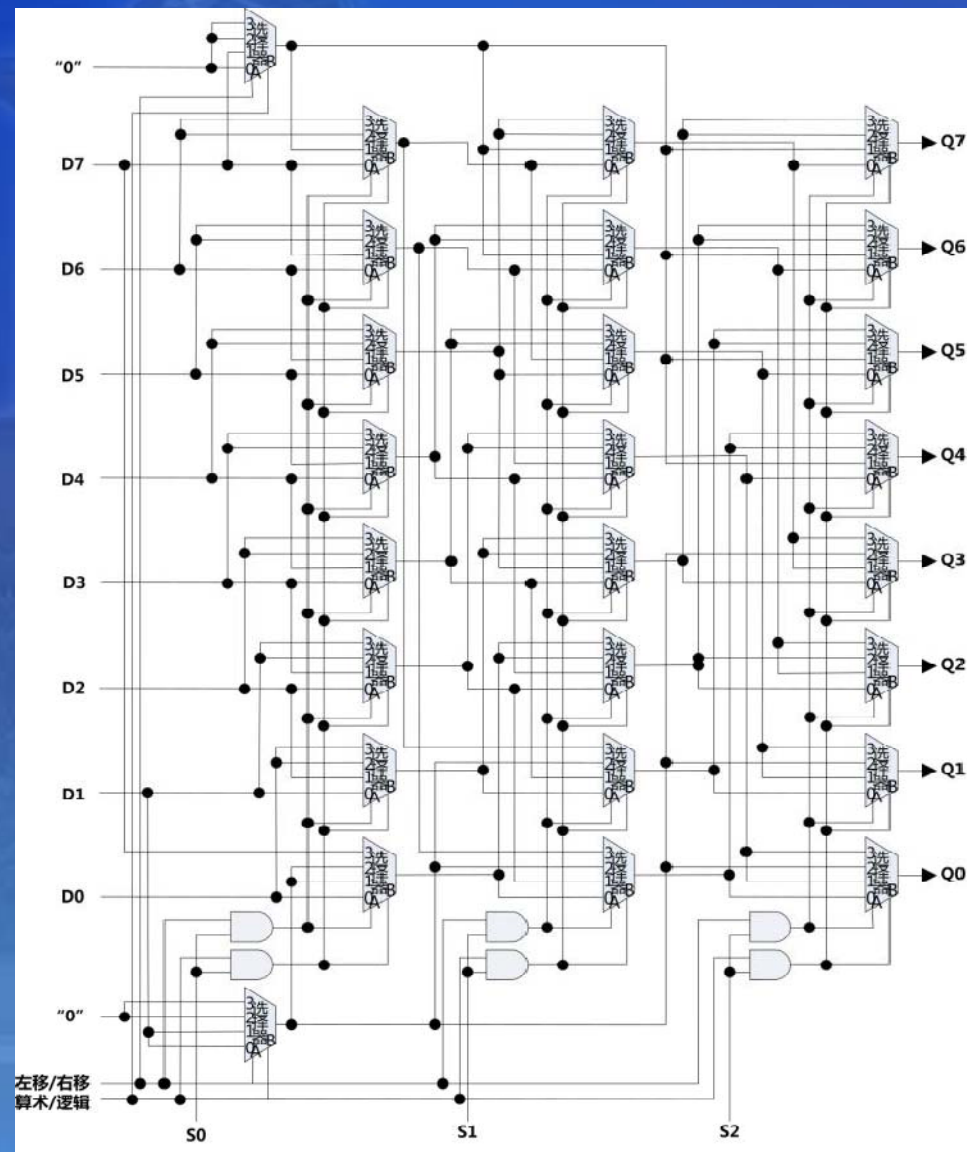
第二部分：详细设计与实现

处理器功能部件设计

32位桶型移位寄存器

运算高效

使用部件少





第二部分：详细设计与实现

处理器功能部件设计

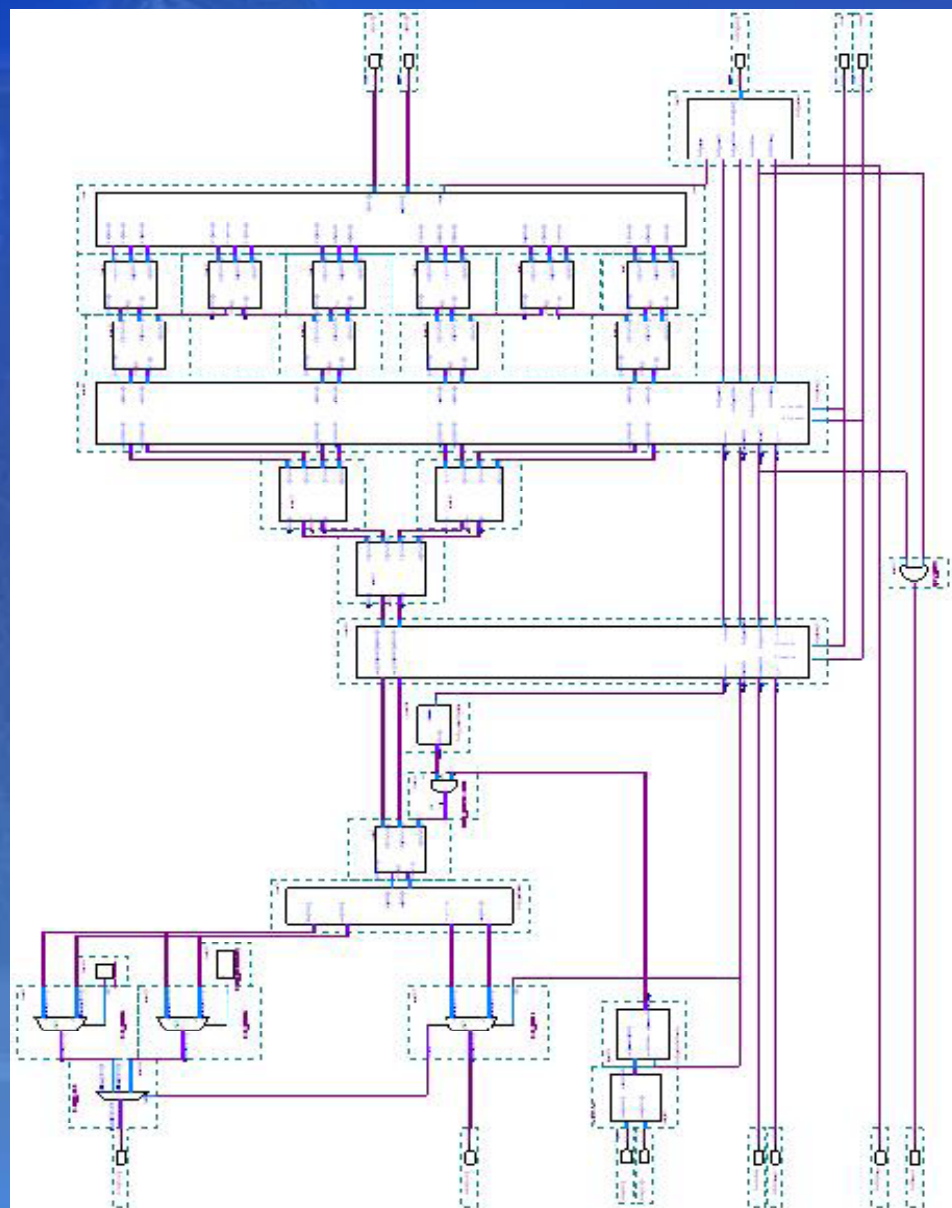
32位乘加器

基4 Booth编码

Wallace树压缩

三级流水，可与
主流水并行

支持各种乘加
/乘减操作



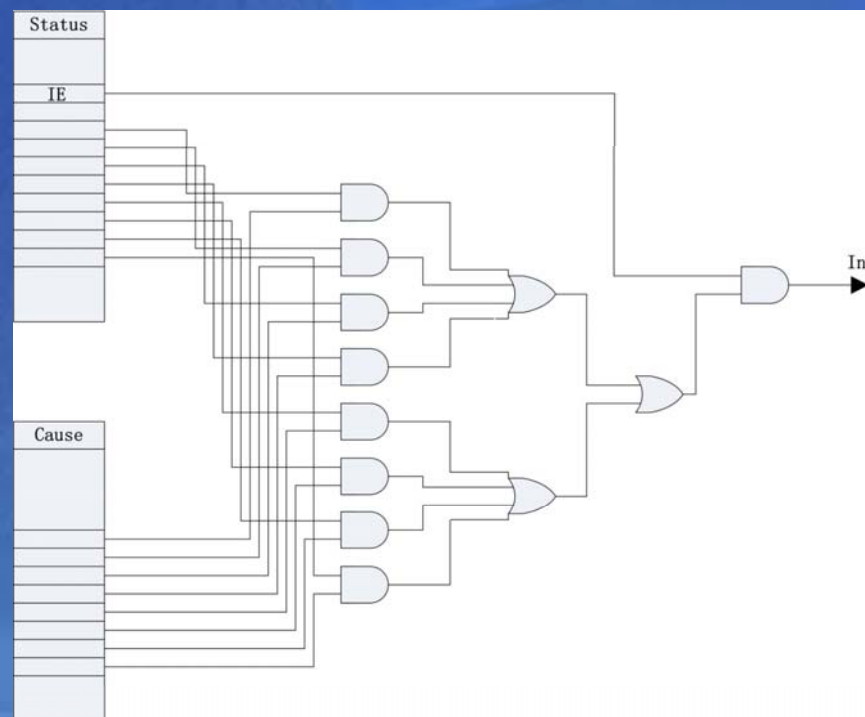
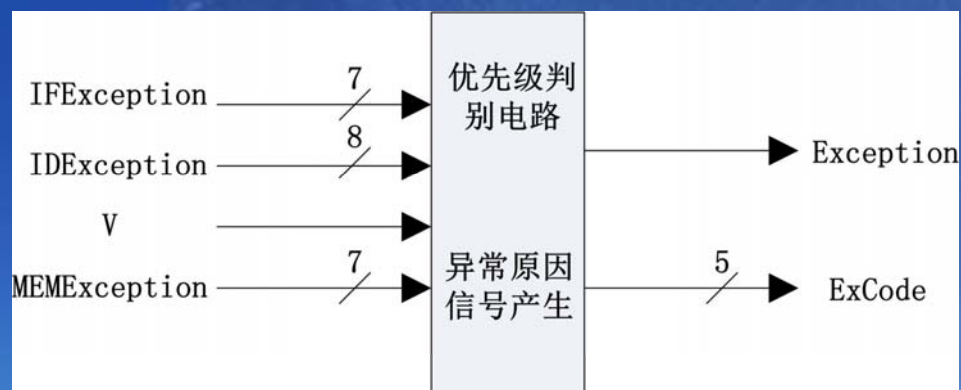


第二部分：详细设计与实现

系统功能部件设计

协处理器CP0

负责异常和
中断的处理





第二部分：详细设计与实现

系统功能部件设计

总线设计与互连

Wishbone总线标准

共享型总线

总线接口单元

总线控制器（引用IP核）





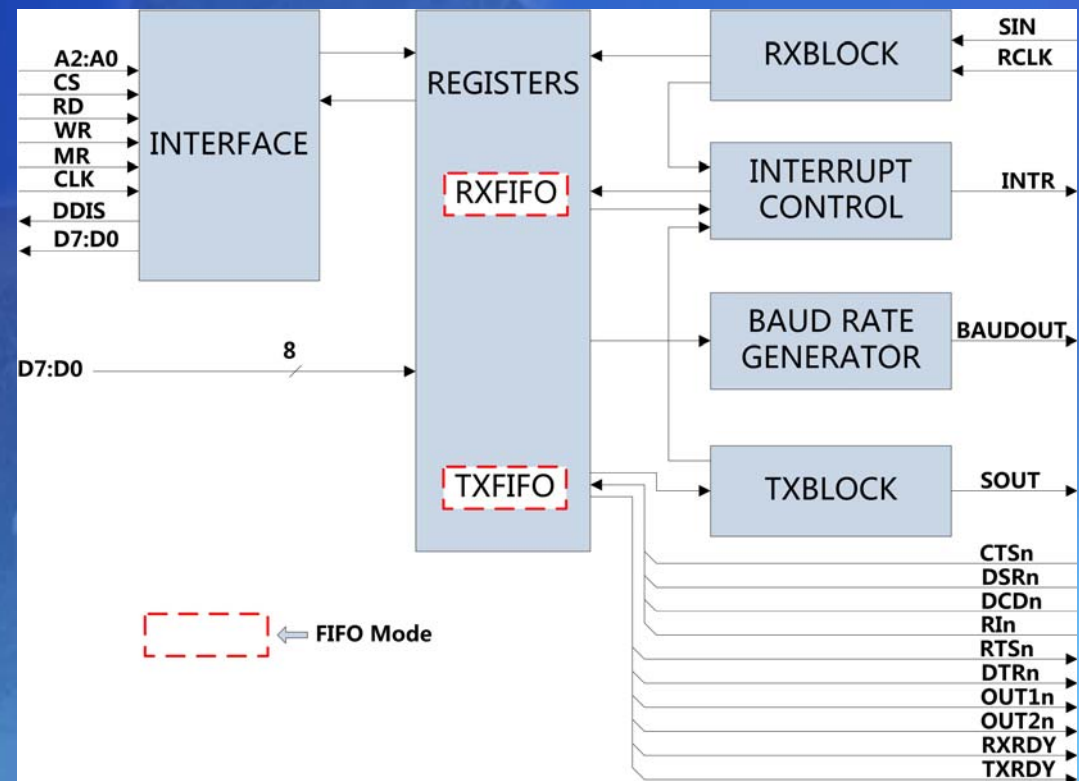
第二部分：详细设计与实现

系统功能部件设计

串行输入输出口

UART16550

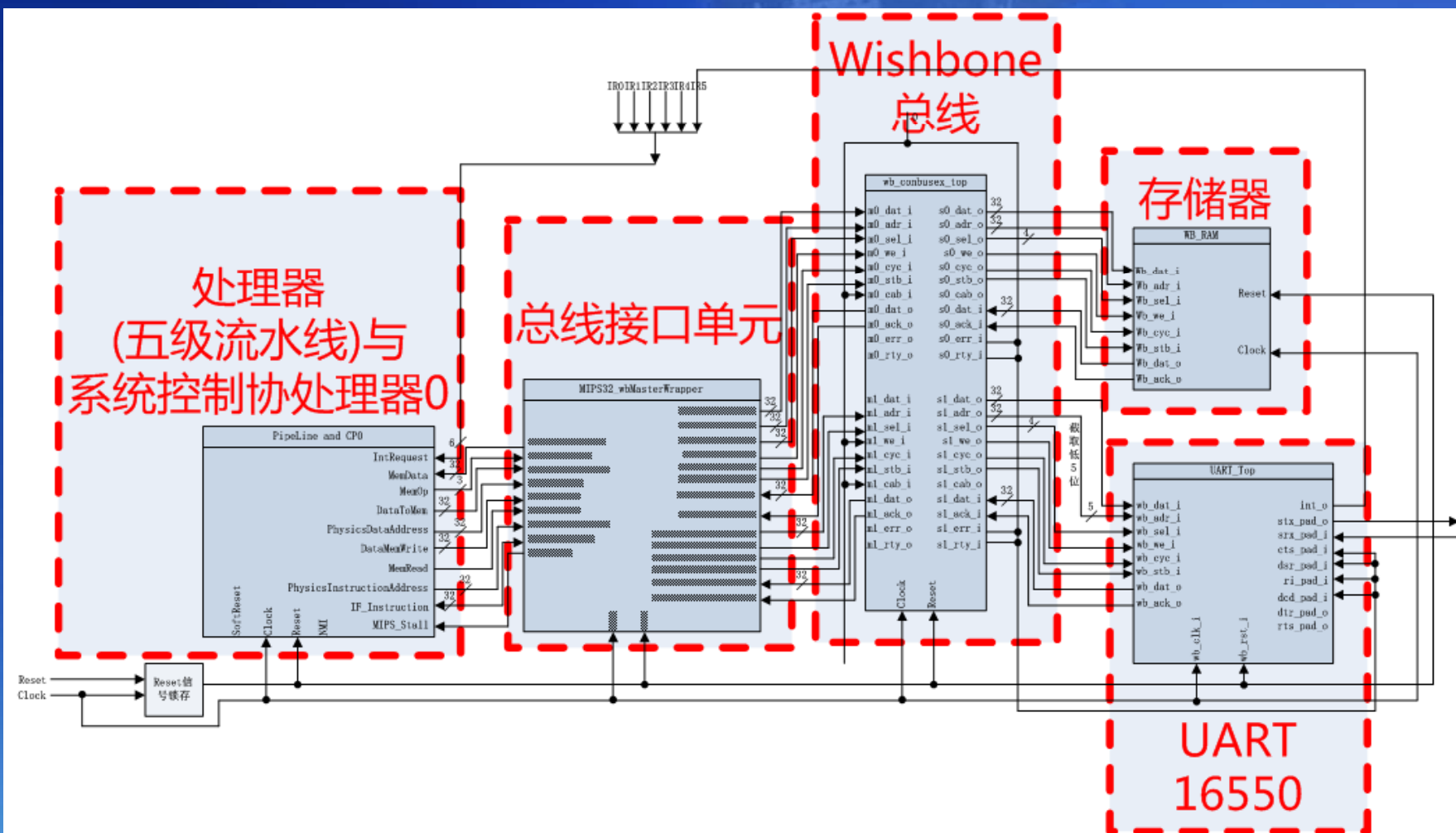
用于监控测试I/O





第二部分：详细设计与实现

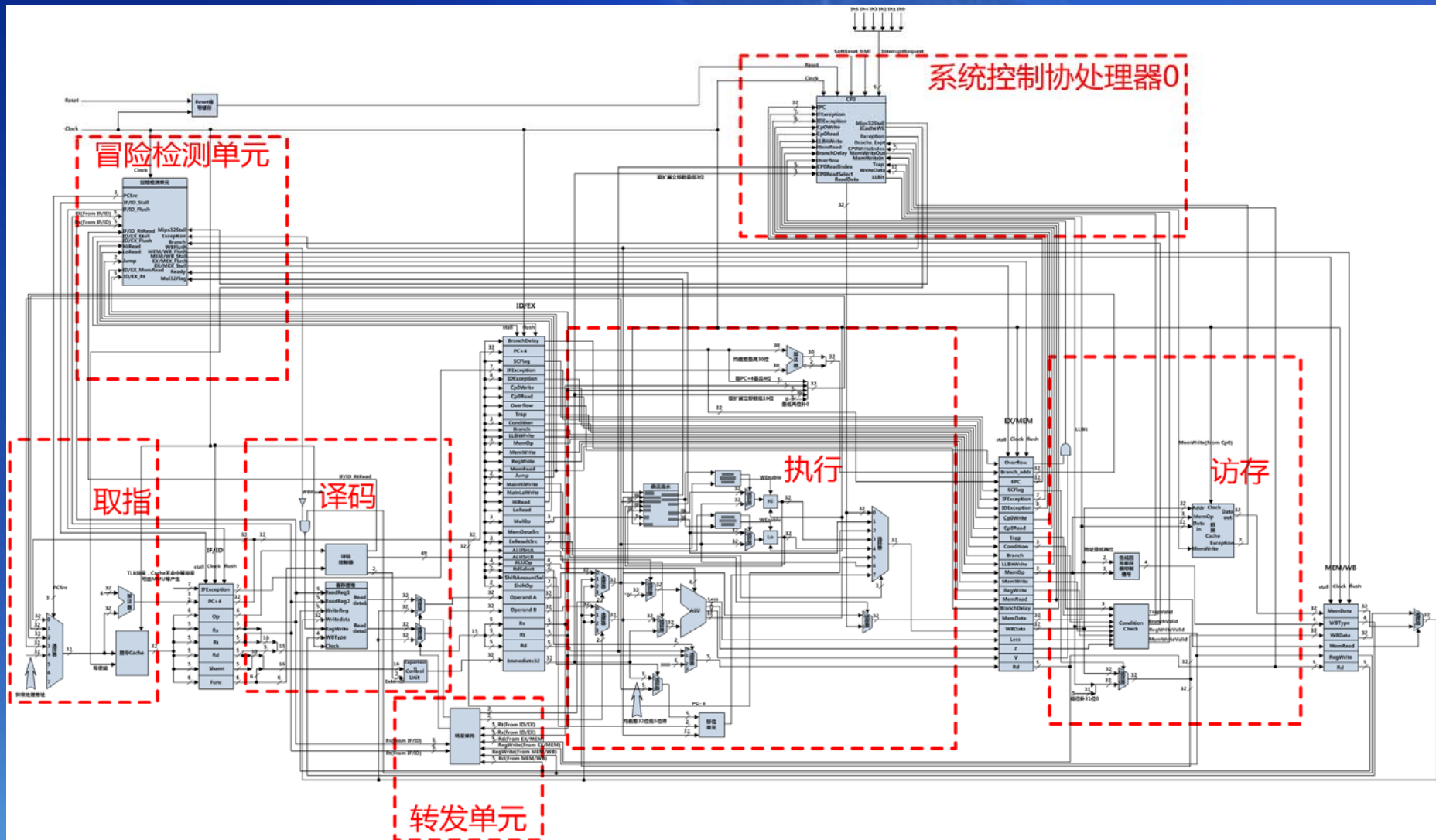
处理器与系统总体结构设计实现





第二部分：详细设计与实现

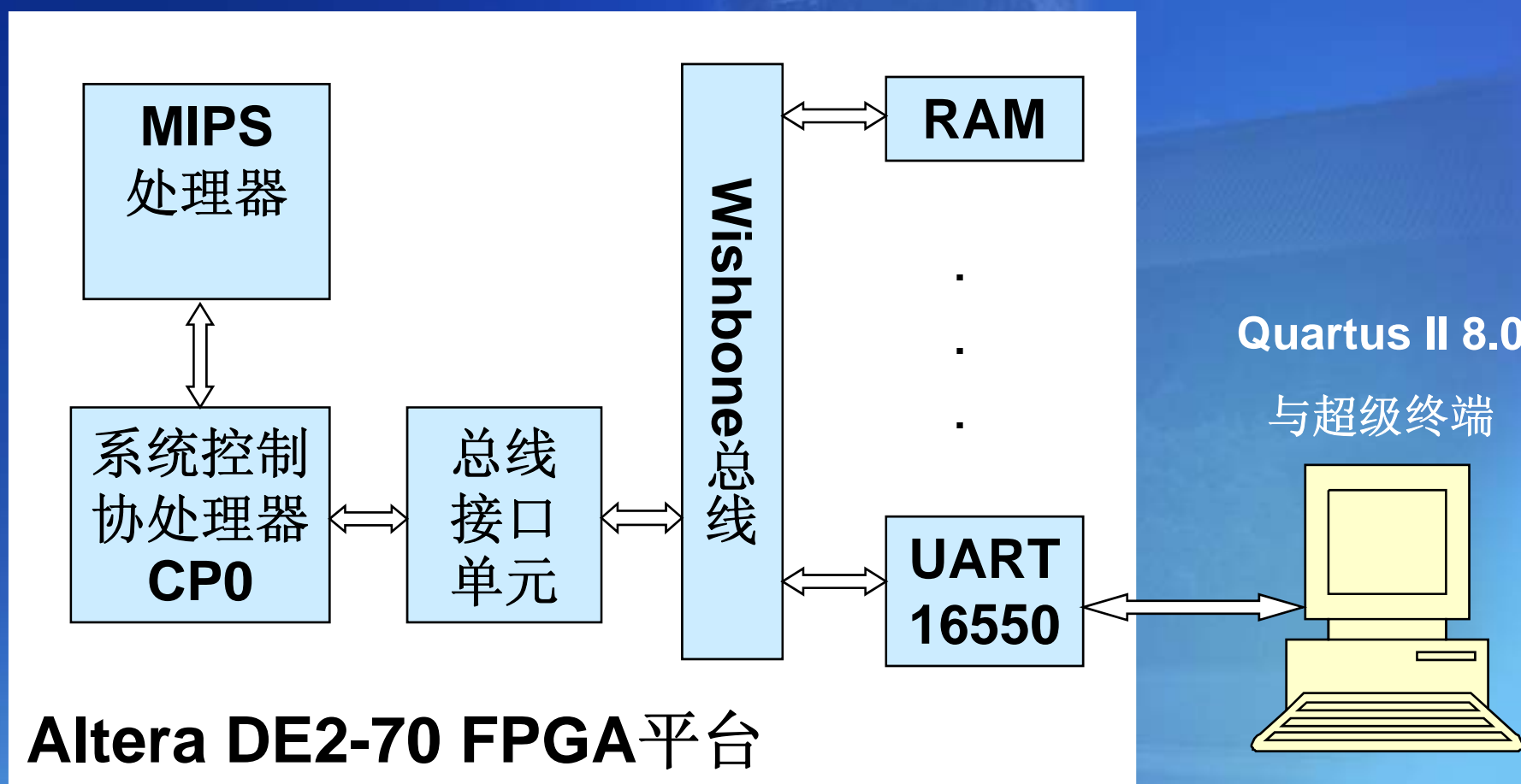
处理器与系统总体结构设计实现





第三部分：开发测试与结果分析

开发测试环境与平台



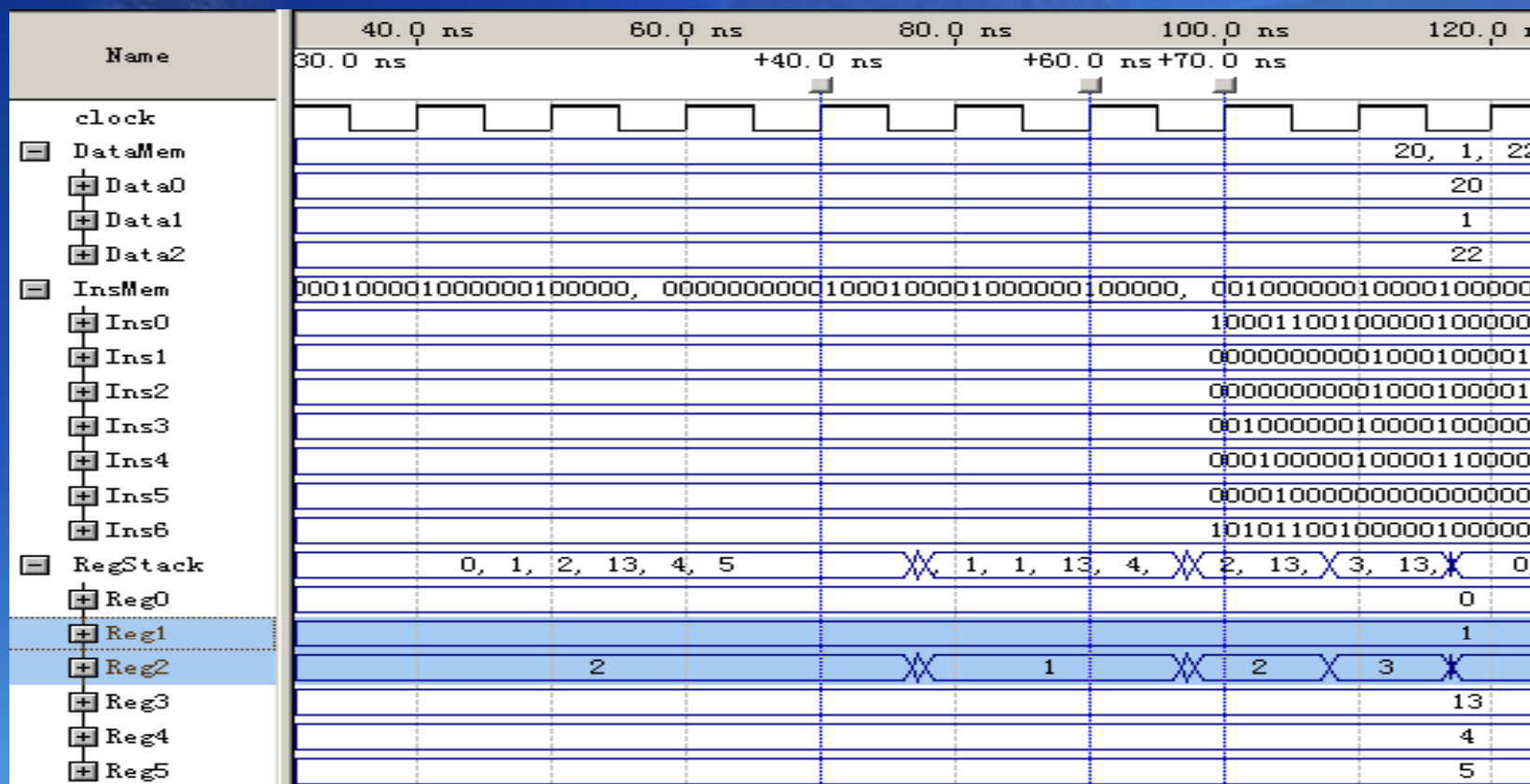


第三部分：开发测试与结果分析

测试方案与过程

仿真测试

时序分析与测试

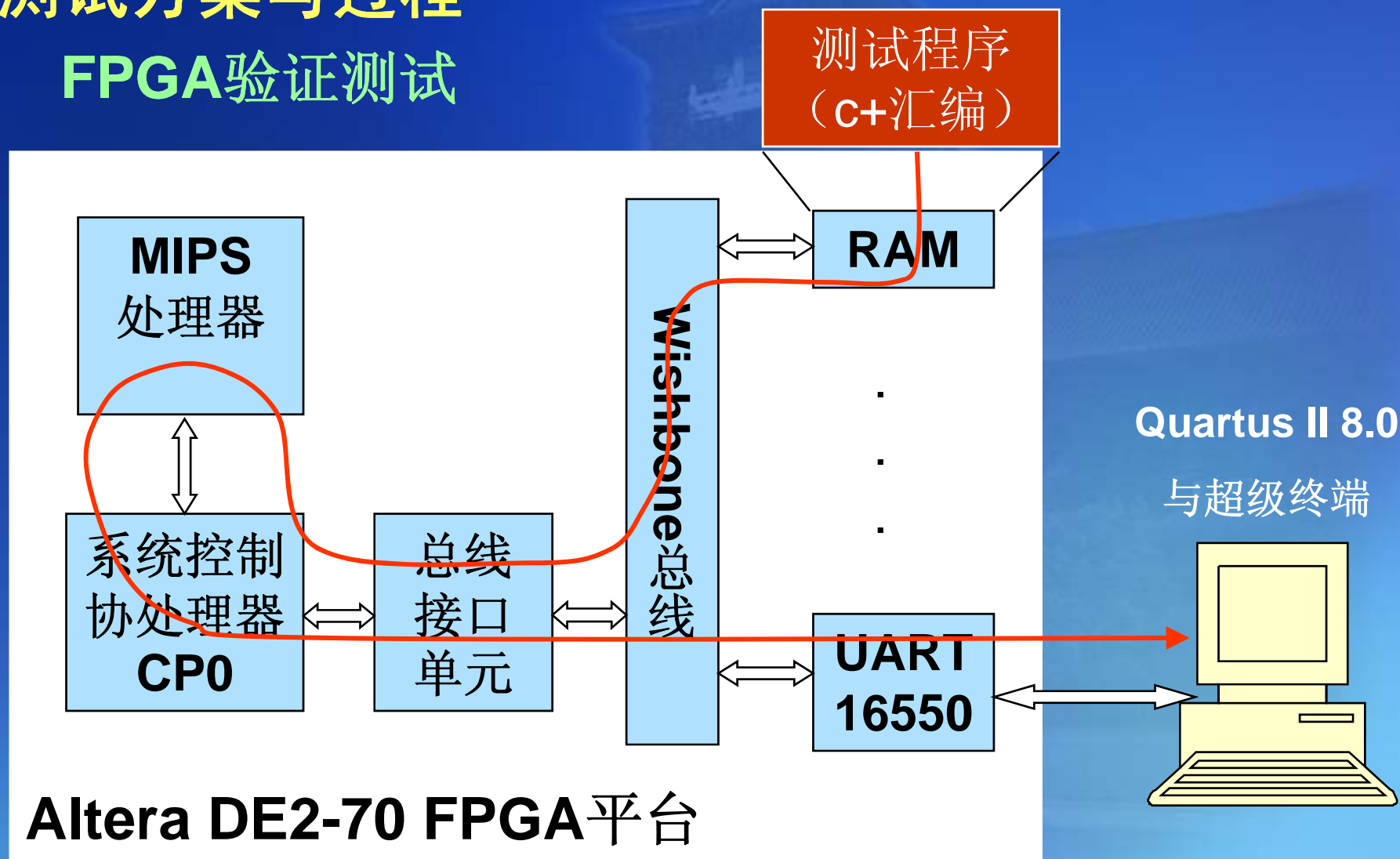




第三部分：开发测试与结果分析

测试方案与过程

FPGA验证测试





第三部分：开发测试与结果分析

演示

从超级终端上输出 “Nanjing University”

运算器测试

指令集测试

中断测试（接收键盘输入并回显到终端）

冒险测试

转发测试



第三部分：开发测试与结果分析

结果分析

硬件与性能参数

FPGA芯片：Cyclone II（EP2C70F896C6）

逻辑单元：

存储单元：

时钟频率：28MHz/50MHz



第四部分：总结与讨论

设计特点

指令的兼容性和完备性

指令集与MIPS32 指令集 2.62版兼容
完成超过80条定点指令

先进的处理器设计技术

运用转发和冒险检测处理五级流水线中结构、数据和控制冒险
在CP0中实现了异常和中断处理机制
采用Wishbone总线

高效的运算部件设计

加法运算采用超前进位加法器
移位采用桶形移位器
乘法采用基于基4Booth编码和Wallace树压缩的乘加器

FPGA验证测试

在Quartus II 8.0下采用硬件描述语言实现电路的逻辑设计，并仿真正确使用Altera DE2-70开发板进行了FPGA验证并实现了简单的应用测试



第四部分：总结与讨论

进一步的工作

浮点运算指令

除法指令

MMU

总线扩展，增加更多丰富的外设

软件（OS，编译，应用程序）



第四部分：总结与讨论

总结与体会

本次大赛参赛收获

理论和实践相结合，加深了对课本理论知识以及计算机系统的理解，培养了分析和解决实际问题的能力，以及实际的计算机系统的设计能力；

培养了团队合作精神和能力，以及交流、表达等综合素质

应当将类似的创新设计实践活动大力推广到本科相关课程的教学中

使用Altera开发平台的心得体会

Altera平台的模块式开发功能使用方便高效；

Altera DE2-70平台通用性强，接口丰富，功能强大，使用方便



致 谢

- 感谢Altera公司所提供的创新学习和展示自我的机会，以及在此过程中所给予的帮助
- 感谢南京大学计算机科学与技术系张泽生老师，在他的全程指导下完成了本项目的设计、实现和测试工作
- 感谢袁春风和黄宜华教授，在项目后期给出的宝贵意见和帮助
- 感谢南京大学计算机科学与技术系，为本参赛项目所提供的实验设备和条件方面的帮助



Q&A

The Altera logo, featuring the word "ALTERA" in a stylized, blue, outlined font.



Thank You !

答辩现场



许佳(右1)和
薛双百(右2)
两位同学正在
演示测试程序。

左边四位为
Altera公司答
辩组成员。

答辩现场



许佳(右1)和
薛双百(右2)
两位同学正在
演示测试程序。

左边四位为
Altera公司答
辩组成员。

颁奖现场



二等奖获奖
者颁奖现场

张泽生老师
(右3) 在
颁奖台上