





Computer Organization & Design 实验与课程设计

综合性课程设计

简易SOC或微控制器应用

汇编程序设计

施青松

Asso. Prof. Shi Qingsong (zjsqs@zju.edu.cn)
College of Computer Science and Technology, Zhejiang University

简单SOC/微控制器应用汇编程序设计



⊙ 基本作业A

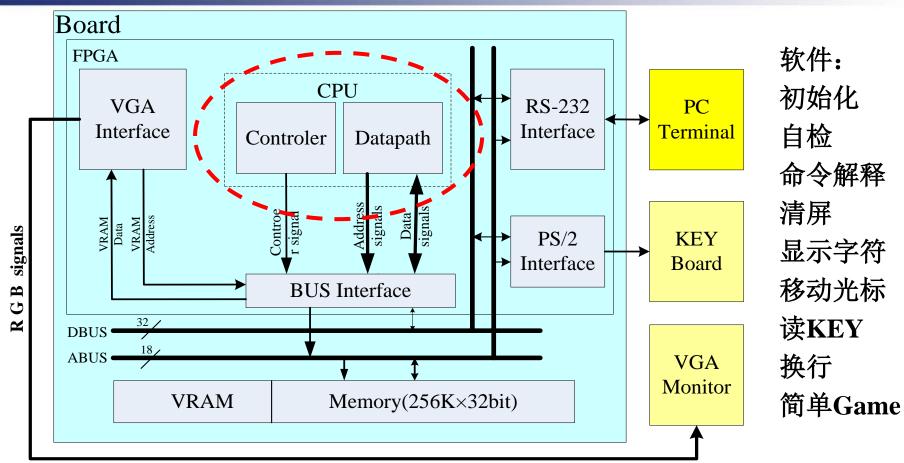
用子程序调用, 充分利用各小题结果

- € 文本显示模式(40×30/80×60,缺省VRAM缓冲基地址: 000C0000)
 - 1. 清屏,置光标00007900=00000000,设计软光标显示
 - 2. 光标移动:上、下、左、右
 - 3. 在当前光标位置显示字符ASCII码,光标右移。字符参数: \$a0=ASCII
 - 4. 读取PS2键盘扫描码,存入扫描码缓冲区,在当前光标位置直接显示
 - 5. 从缓冲区读取扫描码转换为ASCCII码,存入字符缓冲区,并在当前光标处显示
 - 6. 在当前光标位置显示一个字符,光标右移一列,判断边界
 - 7. 换行,光标置下一行首
 - 8. 在屏幕指定位置显示当前寄存器值
 - 9. 在屏幕指定位置,用16进制显示指定内存单元数据。地址参数: \$a0
 - 10. 二进转换为十进制并在屏幕指定位置显示
 - 11. 16进制数转换成ASCII码并在屏幕指定位置显示
 - 12. 屏幕上滚或下滚一行
 - 13. 读取阵列键盘16进制数并在光标处显示



硬件构架: 渐进的计算机硬件系统 充分自由想象和创新的DIY空间





简单且完整的计算机硬件系统:从10条指令开始扩展;从简单SoC扩展到采用VGA显示和P/S2键盘作为人机交互,1MB片外存储器,2K片内ROM,采用集成显卡模式的计算机硬件系统;再扩展到简易计算机系统。横跨组成、体系、短学期、SoC接口及OS等课程。



硬件和仿真平台内存分配参考(根据实际空间调整)



指令: (通用寄存器: 32×32bit)

R-Type: add addu sub subu and or xor nor slt sltu sll srl sra sllv srlv srav jr jalr

J-Type: J \ Jal

I-Type: addi addiu andi ori xori lui lw sw beq bne slti sltiu

内存分配:

ROM区域: FFFF0000 To FFFFCFFF (实际8KB/2KW: FFFF1FFF)

RAM区域(1M):

00000~077FF 系统程序(30K)

07800~08BFF 系统数据(S, 5K)S: Static

08C00~0DBFF 用户程序(20K)

0DC00~0EFFF 用户数据(S,5K)

0F000~BDFFF 用户数据(D,700K)D: Dynamic

BE000~BFFFF 堆栈(8K)

C0000~C12BF 文本charRam (1200字)

C12C0~CFFFF 字库

D0000~F57FF 图形显存(150K)

I/O交互: FFFFD000~FFFFFFF

PS/2接口: 地址分配: FFFFD000

RS-232接口: 地址分配: FFFFE000~ FFFFEFFF

BOARD_State: 地址分配: FFFFE000~ FFFFEFFF

I/O交互区

ROM区域

8K ROM

.

VRAM: 图形

D0000~F57FF

C12C0~CFFFF

C0000~C12BF

BE000~BFFFF

0F000~BDFFF

ODCO0~0EFFF

08C00~0DBFF

07800~08BFF

字库

VRAM: 文本

8K堆栈区

,700K用户数据区

5K用户数据区

20K用户代码区

5K,参数区

30K,系统程序

00000~077FF

参数区约定



```
.text 0x0000
//中断向量区
                                  //reset
           start;
                                  //00000004
  add
           $zero,$zero,$zero;
  add
           $zero,$zero,$zero;
                                  //00000008
  add
           $zero.$zero.$zero:
                                  //000000C
  add
           $zero,$zero,$zero;
                                  //00000010
  add
           $zero,$zero,$zero;
                                  //0000014
           $zero,$zero;
                                  //0000018
  add
  add
           $zero,$zero;
                                  //000001C
//参数区
                                  //00000020
                                              文本光标: 0000XXYY
  add
           $zero,$zero;
  add
           $zero,$zero;
                                  //00000024
                                              图形光标: 00XXXYYY
  add
           $zero,$zero,$zero;
                                  //00000028
                                              键盘缓冲区头指针
                                              键盘缓冲区尾指针
  add
           $zero,$zero,$zero;
                                  //0000002C
                                              键盘缓冲区低字:
                                                             最近4个ASCII码
  add
           $zero,$zero,$zero;
                                  //00000030
           $zero,$zero;
                                  //0000034
                                              键盘缓冲区第2字:次近4个ASCII码
  add
                                              键盘缓冲区第3字:次高4个ASCII码
           $zero,$zero;
                                  //0000038
  add
                                              键盘缓冲区高字: 最高4个ASCII码
  add
           $zero,$zero,$zero;
                                  //000003C
  add
           $zero,$zero,$zero;
                                  //00000040
                                             System Status Word:shif=D31,press_hold=d30,
                                  //00000044
                                              键盘扫描码缓冲区低: 去掉F0
  add
            $zero,$zero,$zero;
                                              键盘扫描码缓冲区高: 去掉F0
  add
           $zero,$zero;
                                  //0000048
  add $zero,$zero,$zero;
                                  //0000004C
                                  //....
           $zero,$zero;
                                  //000007C
  add
//系统程序:
          00000080
```



start:

计算机学院

addi \$sp,\$zero,4000;

系统结构与系统软件实验室

//堆栈初始化, SP=4000



● 基本作业B(*题只要选做一题)

- 图形显示模式(640×480, 缺省VRAM_Graph缓冲基地址: 000D0000)
 - 在屏幕指定位置显示一个点。点参数: \$a0=0000XXYY
 - 2. 在屏幕指定位置显示直线。端点参数: \$a0=XXYYXXYY
 - 在屏幕指定位置显示四边形。端点参数: \$a0,\$a1=XXYYXXYY
 - 在屏幕指定位置画圆(圆心: \$a0=0000XXYY, 半径\$a1)*
 - 5. 对指定内存区域作MIPS的反汇编*
 - 6. 在指定内存区域开始输入MIPS指令并汇编后存入内存*

● 微处理器应用课程设计:汇编部分(期末结合硬件提交)

- 设计有意义的应用程序同步Project硬件设计
 - 基于简单交互(四个简单外设)支持的小游戏或应用
 - 或简单交互(如阵列键盘、VGA、PS2等,参见硬件要求)
 - 参考:逻辑手指跳舞毯示例或作业视频



硬件要求:基本要求



■ 设计简单总线接口

- ■设计基本GPIO总线接口
 - □支持低速设备
- ■接口功能支持
 - □主存储器
 - □输出: 七段显示器、GPIO/LED
 - □输入: GPIO/SW、独立按键
 - □简单计数器

□ 任选一款实验设计的CPU

- □构建SOC应用系统
 - 其余模块可以用核或自行设计
- 设计有意义的应用程序
 - 四个简单外设支持的小游戏
 - □ 参考: 手指跳舞毯



硬件要求: 中级要求



■ 基于GPIO的Arduino接口

- 1. 扩展基于GPIO模式的接口
 - □ 支持四组8位标准GPIO
 - Bank0=DIO[7:0]; Bank1[5:0]=DIO[13:8]
 - Bank2[5:0]=AIO[5:0]; Bank2[7]=ADC_SCL
 - □ Bank2[6]=ADC_SDA; Bank3=扩展

2. 接口标准

- □ Addr: 8个地址,每组2个,单地址位控操作,偶地址字节操作
- □ Din: CPU输出数据; Dout: CPU读入数据
- □ adcSCL: ADC时钟; adcSDA: ADC数据。均双向接口信号
- □ ArduinoDIO, ArduinoAIO: Arduino接口信号,均双向信号

3.控制字模式

31-24位: 清除控制位。写"1"对应GPIO[i]=0,写"0"保持不变

23-16位: 置位控制位。写"1"对应GPIO[i]=0,写"0"保持不变

15-8位: 翻转控制位。写"1"对应GPIO[i]取反,写"0"保持不变

7-0位:数据传输位。对应实时读/写对应GPIO信号(对应偶地址)



> / 学 系统结构与系统软件实验室

硬件要求: 高级要求



■ 任选或组合均可以

- 要在自己应用中体现: 根据应用要求决定
- 1. 扩展阵列键盘输入总线接口
 - □增加阵列键盘输入支持
 - □阵列式键盘预处理模块可以用核或自行设计
- 2. 扩展VGA文本总线接口
 - □ 支持VGA 640*480,标准模式
 - □设计VGA显示接口驱动
- 3.扩展PS2键盘总线接口
 - □ 支持PS2键盘
 - □设计PS2接口驱动

■ 设计较高级的应用程序

- VGA-PS2交互应用
 - □ 参考: 吃豆子、俄罗斯方块





END