计算机体系结构基础

习题课（6-10章）

第06章 第1题

- **Q：在观看网络视频时，网卡将接收到的网络包放到内存并通知处理器对视频数据流进行解码并发送给显示单元。此时，处理器对该网络包的中断处理是否应分上下半部？若是，应如何切分？**
- **A：**
  - 视频数据流大小可观，且需要处理器进行解码，若都在上半部进行处理会使中断上下文占时过多，不利于任务调度。
  - 如何切分？
    - 上：将网络包拷贝到应用程序缓冲区，设置下半部任务
    - 下：应用程序得到时间片后进行解码并按序显示

第06章 第2题

- **Q： 1. 用MIPS汇编程序来举例并分析未同步的线程之间进行共享数据访问出错的情况**

  2. 用LL/SC指令改写你的程序，使它们的共享数据访问正确。

- **A:**
  - **线程a/b都要将内存X的值自增1（如线程共用的计数器）,当a/b在多处理器中同时执行时，可能会同时取回旧值加1存回**

```
li   t0,X
lw   t1,0(t0)
addi t1,t1,1
sw   t1,0(t0)
```

```
li   t0,X
1:
ll   t1,0(t0)
addi t1,t1,1
sc   t1,0(t0)
beqz t1,1b
nop
```

第06章 第3题-Q

- **Q：**
  1. **在你的机器上安装MIPS交叉编译器，通过编译-反汇编的方式提取函数调用的核心片断。**
  2. **改变编译的优化选项，记录函数调用核心片断的变化，并分析不同优化选项的效果。**

第06章 第3题-A

- **安装交叉编译器：**
  - *http://ftp.loongnix.org/toolchain/gcc/release/CROSS_COMPILE/gcc-4.4.0-pmon.tgz*
  - 下载后,在/usr/local/目录下依次创建目录:comp/mips-elf/
  - 解压 gcc 包:tar zxvf gcc-4.4.0-pmon.tgz -C /usr/local/comp/mips-elf/
  - 设置如下环境变量:
  - `export LD_LIBRARY_PATH=/usr/local/comp/mips-elf/gcc-4.4.0-pmon/lib:`
  - `export CROSS_COMPILE=mipsel-linux-`
  - `export PATH=/usr/local/comp/mips-elf/gcc-4.4.0-pmon/bin/:$PATH`
- **编译和反汇编程序：**
  - `mipsel-linux-gcc 6-3.c -g -static <-Ox>`
  - `mipsel-linux-objdump -ald a.out |vi -`

course/6-3.c

```c
#include <stdio.h>

int mad(int a, int b, int c)
{
  return a+b*c;
}

int main()
{
  int a,b,c;
  int d;

  a=1; b=2; c=3;
  d = mad(a,b,c);
  printf("d=%d\n",d);

  return 0;
}
```

第06章 第3题-A

```
004003f0 <mad>:
mad():
/home/liusu/course/6-3.c:4
  4003f0:       27bdfff8        addiu   sp,sp,-8
  4003f4:       afbe0004        sw      s8,4(sp)
  4003f8:       03a0f021        move    s8,sp
  4003fc:       afc40008        sw      a0,8(s8)
  400400:       afc5000c        sw      a1,12(s8)
  400404:       afc60010        sw      a2,16(s8)
/home/liusu/course/6-3.c:5
  400408:       8fc3000c        lw      v1,12(s8)
  40040c:       8fc20010        lw      v0,16(s8)
  400410:       00000000        nop
  400414:       00620018        mult    v1,v0
  400418:       00001812        mflo    v1
  40041c:       8fc20008        lw      v0,8(s8)
  400420:       00000000        nop
  400424:       00621021        addu    v0,v1,v0
/home/liusu/course/6-3.c:6
  400428:       03c0e821        move    sp,s8
  40042c:       8fbe0004        lw      s8,4(sp)
  400430:       27bd0008        addiu   sp,sp,8
  400434:       03e00008        jr      ra
  400438:       00000000        nop
```

```
/home/liusu/course/6-3.c:13
  400458:       24020001        li      v0,1
  40045c:       afc20024        sw      v0,36(s8)
  400460:       24020002        li      v0,2
  400464:       afc20020        sw      v0,32(s8)
  400468:       24020003        li      v0,3
  40046c:       afc2001c        sw      v0,28(s8)
/home/liusu/course/6-3.c:14
  400470:       8fc40024        lw      a0,36(s8)
  400474:       8fc50020        lw      a1,32(s8)
  400478:       8fc6001c        lw      a2,28(s8)
  40047c:       0c1000fc        jal     4003f0 <mad>
  400480:       00000000        nop
```

无优化：

存入栈，又从栈取出

第06章 第3题-A

```
004003f0 <mad>:
mad():
/home/liusu/course/6-3.c:4
  4003f0:       00c50018        mult    a2,a1
  4003f4:       00001012        mflo    v0
/home/liusu/course/6-3.c:6
  4003f8:       03e00008        jr      ra
  4003fc:       00441021        addu    v0,v0,a0


/home/liusu/course/6-3.c:14
  400414:       24040001        li      a0,1
  400418:       24050002        li      a1,2
  40041c:       0c1000fc        jal     4003f0 <mad>
  400420:       24060003        li      a2,3
```

优化-O/-O1：

只用寄存器，充分利用延迟槽

第06章 第3题-A

```
004003f0 <mad>:
mad():
/home/liusu/course/6-3.c:4
  4003f0:       00c50018        mult    a2,a1
  4003f4:       00001012        mflo    v0
/home/liusu/course/6-3.c:6
  4003f8:       03e00008        jr      ra
  4003fc:       00441021        addu    v0,v0,a0

00400400 <main>:
main():
/home/liusu/course/6-3.c:9
  400400:       3c1c004b        lui     gp,0x4b
  400404:       27bdffe0        addiu   sp,sp,-32
  400408:       279ce880        addiu   gp,gp,-6016
  40040c:       afbf001c        sw      ra,28(sp)
  400410:       afbc0010        sw      gp,16(sp)
/home/liusu/course/6-3.c:15
  400414:       8f998974        lw      t9,-30348(gp)
  400418:       3c040048        lui     a0,0x48
  40041c:       248491d4        addiu   a0,a0,-28204
  400420:       0320f809        jalr    t9
  400424:       24050007        li      a1,7
/home/liusu/course/6-3.c:18
  400428:       8fbf001c        lw      ra,28(sp)
  40042c:       00001021        move    v0,zero
  400430:       03e00008        jr      ra
  400434:       27bd0020        addiu   sp,sp,32
        ...
```

优化-O2/-O3/-Os：

此时已不再调用mad，而是直接由编译器算出结果

第06章 第4题-Q

- **Q：ABI中会包含对结构体中各元素的对齐和摆放方式的定义。**
  1. **在你的机器上用C语言编写一段包含不同类型的结构体，并获得结构体总空间占用。**
  2. **调整结构体元素顺序，推测并分析结构体对齐的方式。**

第06章 第4题-A

- 已知
  - sizeof(char)=1
  - sizeof(short)=2
  - sizeof(int)=4
- 运行结果：
  - size=8

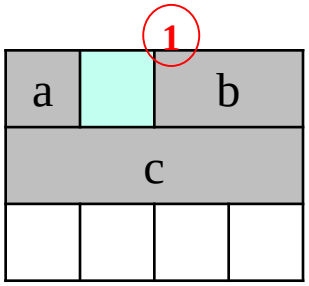
course/6-4.c

```c
#include <stdio.h>

typedef struct{
  char  a;
  short b;
  int   c;
} Tstruct;

int main()
{

printf("size=%d\n",sizeof(Tstruct));
}
```

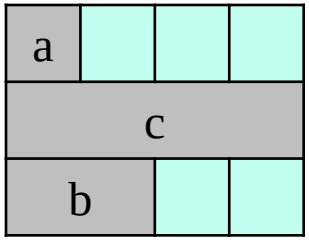
第06章 第4题-A

1. 按顺序优先放在可用的对齐的低地址
2. 以当前最大的size为基本单位，不足基本单位的被浪费
3. 引入64位类型：总size以32位对齐（32位机器上）
   总size以64位对齐（64位机器上）

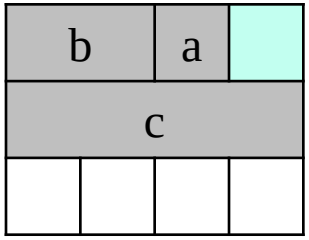
```
3 typedef struct{
4   char  a;
5   short b;
6   int   c;
7 } Tstruct;
```



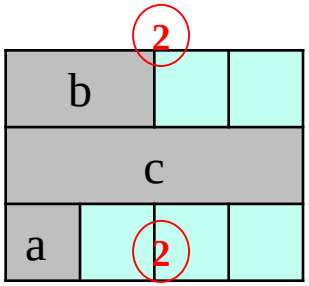

```
3 typedef struct{
4   char  a;
5   int   c;
6   short b;
7 } Tstruct;
```



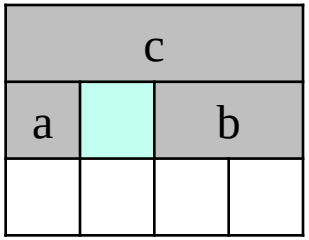

```
3 typedef struct{
4   short b;
5   char  a;
6   int   c;
7 } Tstruct;
```



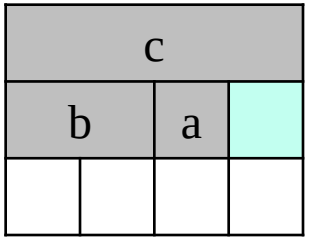

```
3 typedef struct{
4   short b;
5   int   c;
6   char  a;
7 } Tstruct;
```




```
3 typedef struct{
4   int   c;
5   char  a;
6   short b;
7 } Tstruct;
```




```
3 typedef struct{
4   int   c;
5   short b;
6   char  a;
7 } Tstruct;
```



第06章 第5题

- **Q：函数调用、系统调用和中断处理都需要上下文切换，请结合MIPS O32的ABI说明上述三种上下文切换时保留现场有什么不同（内容、位置）？**

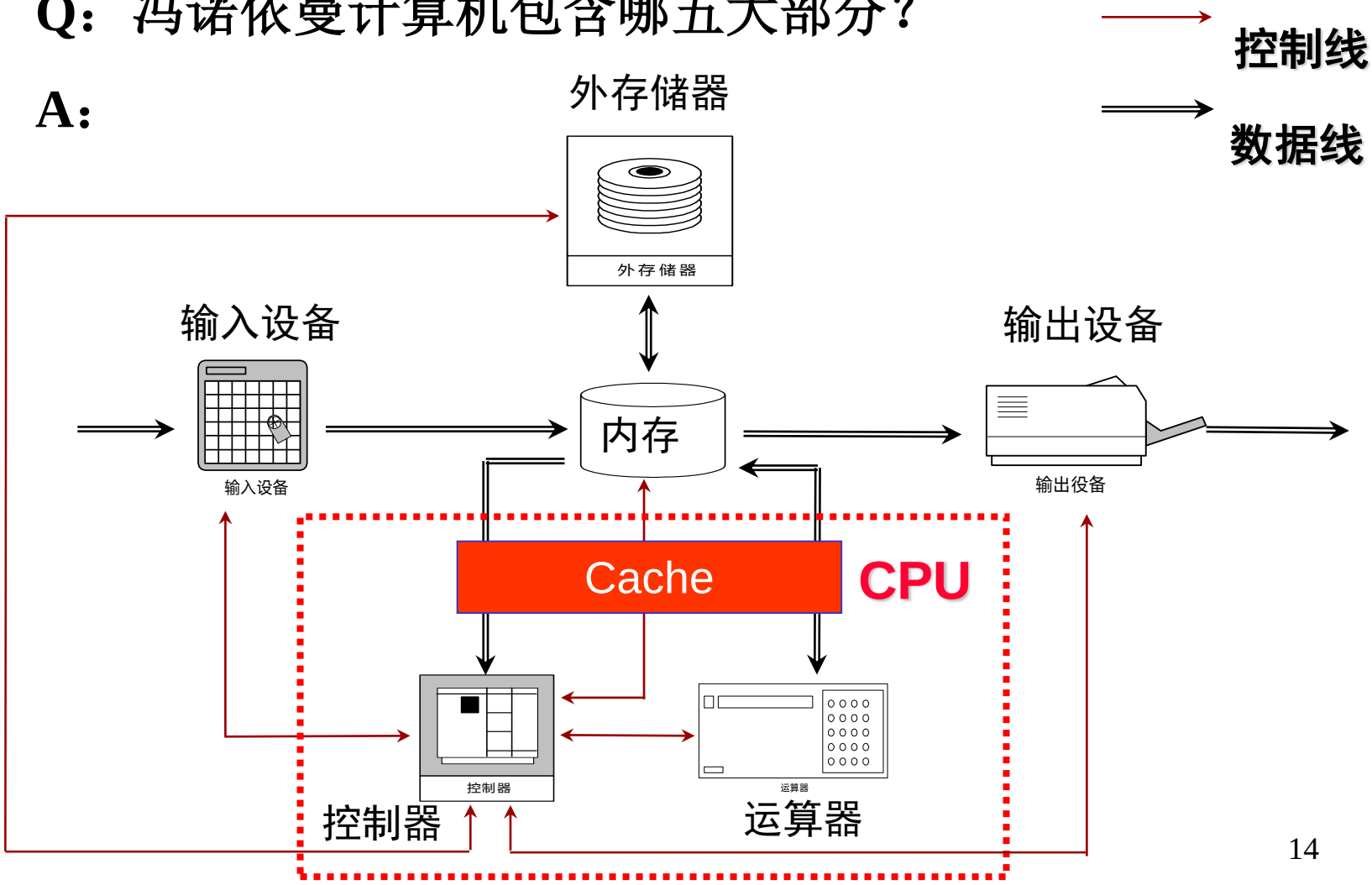
第06章 第5题-A

- **A：**

	位置	内容
函数调用	当前栈	Caller saved寄存器（$1-$15,$24-$25,$28） Callee saved寄存器（$16-$23,$29-$31)
系统调用	系统栈	SAVE_SOME,v*/a*/SR/CR
中断处理		SAVE_ALL,k*外的所有通用寄存器

第07章 第1题

- **Q：冯诺依曼计算机包含哪五大部分？**
- **A：**



第07章 第2题

- **Q：按照冯诺依曼结构，硬盘属于什么类型的部件。**
- **A：硬盘属于外存储器**
  - 从功能上看，属于冯诺依曼中的存储器
  - 从访问方式上，属于输入设备/输出设备

第07章 第3题

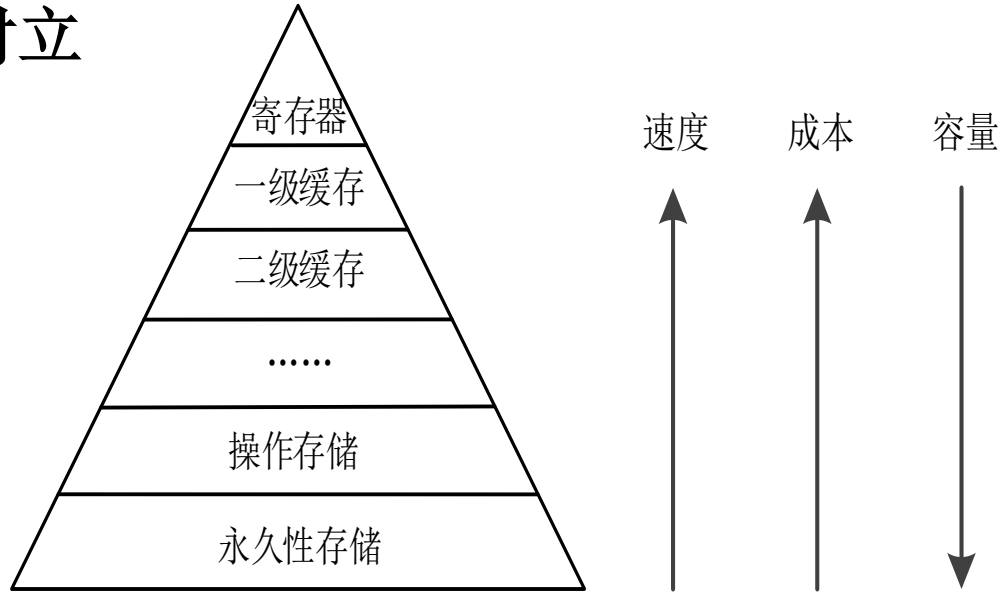
- **Q：列举几种计算机输入设备的例子。**
- **A：**
  - **纯输入设备：鼠标、键盘、触摸板、扫描仪。。**
  - **输入输出设备：网卡、U盘、光驱。。**

第07章 第4题

- **Q：为什么现代高性能计算机采用层次化的存储系统？**
- **A：**

  容量和成本/速度的对立

  局部性原理



第07章 第5题

- **Q：说明北桥、南桥芯片的主要功能**
- **A：**
  - **北桥：连接CPU、GPU、南桥、PCIE、内存等，作为数据传输的核心通道**
  - **南桥：搭载USB、网络、SATA、UART、IIC等低速设备，提供丰富的输入/输出设备支持**

第07章 第6题

- **Q：处理器和IO设备间的通信方式有哪几种？**
- **A：**
  - **事件的获取：分为轮询方式和中断方式**
  - **数据的传输：分为PIO方式和DMA方式**

第07章 第7题

- **Q：为什么要使用DMA？**
- **A：**
  - **CPU效率：**PIO方式中，CPU作为内存和IO数据传输的中转，实际有效工作量不大
  - **CPU时间：**数据量非常大时，PIO方式会占用非常可观的CPU时间
  - **传输效率：**IO设备直接访问内存，可利用burst传输等特性提高效率
  - **通信方式：**DMA方式和中断的配合

第07章 第8题

- **Q：GPU和CPU有什么不同？它们的主要功能有什么不同？**
- **A：**
  - **CPU：**控制器和运算器，负责整个计算机的控制和通用计算
  - **GPU：**负责图形图像的处理和绘制、一些特定的计算加速应用（GPGPU）

第07章 第9题

- **Q：调查当前市面上光盘、磁盘、U盘、内存条的主流价格，并计算每GB存储容量的价格。**
- **A：以JD销量第一产品为例**

	品牌	容量	价格（元）	每GB价格
光盘	ARITA	4.7G×50	46	0.196
硬盘	WD 蓝盘	1TB	329	0.329
U盘	Kingston DTSE9	32GB	79.9	2.497
内存	Kingston Fury	8GB	299	37.375

第07章 第10题

- **Q：中断控制器的作用是什么？**
- **A：**
  - 保存中断状态，特别是脉冲型中断
  - 归一化中断线类型（电平/脉冲、高触发/低触发）
  - 集线，节省处理器中断输入
  - 为处理器提供查询、清除、控制中断的接口

第08章 第1题

- **Q：找一台电脑，打开机箱说明每条连线都是什么总线。**
- **A：不设标准答案**

第08章 第2题

- **Q：说明总线包含哪些层次。**
- **A：课件Ch8.p4**

  总线规范包括以下层次：
  - 机械层：接口的外形、尺寸、信号排列、连接线的长度等等
  - 电气层：信号描述、电源电压、电平标准、信号质量等等
  - 协议层：信号时序、握手规范、命令格式、出错处理等等
  - 架构层：硬件模型、软件框架等等

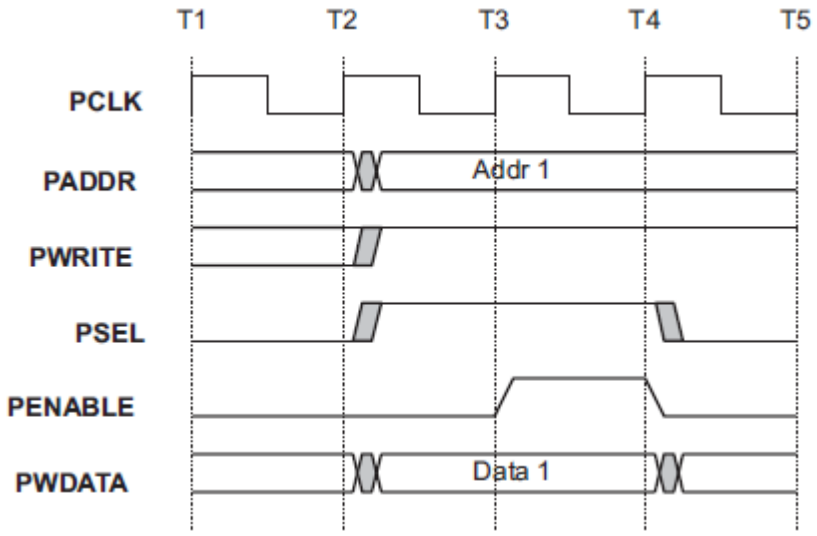
第08章 第3题

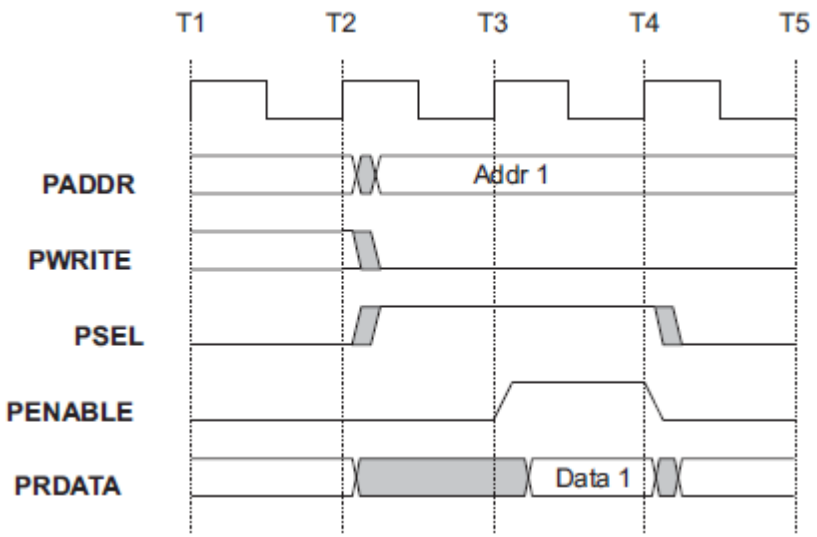
- **Q：计算一组AXI总线需要的信号线个数。**
- **A：基于AMBA AXI Protocol v1.0，总计209根，会根据ID、地址、数据的宽度变化而变**

通道	信号	总信号线个数
全局	ACLK,ARESETn	2
AW	AWID[3:0],AWADDR[31:0],AWLEN[3:0],AWSIZE[2:0],AWBURST[1:0],AWLOCK[1:0],AWCACHE[3:0],AWPROT[2:0],AWVALID,AWREADY	56
W	WID[3:0],WDATA[31:0],WSTRB[3:0],WLAST,WVALID,WREADY	43
B	BID[3:0],BRESP[1:0],BVALID,BREADY	8
AR	ARID[3:0],ARADDR[31:0],ARLEN[3:0],ARSIZE[2:0],ARBURST[1:0],ARLOCK[1:0],ARCACHE[3:0],ARPROT[2:0],ARVALID,ARREADY	56
R	RID[3:0],RDATA[31:0],RRESP[1:0],RLAST,RVALID,RREADY	41
LP	CSYSREQ,CSYSACK,CACTIVE	3

第08章 第4题

- **Q：阅读AMBA APB协议并设计一个APB接口的GPIO模块。**
- **A：两个关键点：APB接口、GPIO模块**





第08章 第4题-A

```
module GPIO(
    input pclk,
    input presetn,
    input penable,
    input psel,
    input pwrite,
    input [9:0] paddr,
    input [31:0] pwdata,
    output[31:0] prdata,

    input [31:0] gpio_i,
    output[31:0] gpio_o,
    output[31:0] gpio_oe
);
reg [31:0] reg_gpio_i,reg_gpio_o,reg_gpio_oe;
reg [31:0] reg_gpio_i_t;
always@(posedge pclk)
    if(~presetn) begin
        reg_gpio_i   <= 32'h0;
        reg_gpio_i_t <= 32'h0;
    end
    else begin
        reg_gpio_i_t <= gpio_i;
        reg_gpio_i   <= reg_gpio_i_t;
    end
```

```
always@(posedge pclk)
    if(~presetn) begin
        reg_gpio_o <= 32'h0;
        reg_gpio_oe<= 32'h0;
    end
    else if(penable & psel & pwrite)
        case(paddr)
            0: reg_gpio_o <= pwdata;
            1: reg_gpio_oe<= pwdata;
        endcase

assign prdata = paddr == 0 ? reg_gpio_o :
                paddr == 1 ? reg_gpio_oe :
                             reg_gpio_i;

assign gpio_o  = reg_gpio_o;
assign gpio_oe = reg_gpio_oe;

endmodule

//--------GPIO--------
assign GPIO00 = gpio_oe[0] ? gpio_o[0] : 1'bz;
assign gpio_i[0] = GPIO00;
```

第08章 第5题

- **Q：DRAM的寻址包含哪几部分？**
- **A：课件Ch8.p31**

  地址译码
  - **物理地址→SDRAM地址（片选、bank、行、列）**

第08章 第6题

- **Q：假设一个处理器支持两个DDR3通道，每个通道为64位宽，内存地址线为15，片选个数为4，计算该处理器实际支持的最大内存容量。**
- **A：根据DDR3协议规定，bank选择最多3位，行地址最大15位，列地址最大10位。**
- **总字节数为 2 * 8 * 4 * 2^3 * 2^15 * 2^10 = 16GB**

双通道 64位 片选 bank数 行地址数 列地址数

第08章 第7题

- **Q：查找资料了解SATA总线的发展过程。**
- **A：参考课本8.5.1小节**

  https://en.wikipedia.org/wiki/Serial_ATA#Revisions

第08章 第8题

- **Q：SATA规范是指那两部分的接口？**
- **A：**

  （wikipedia）

  Serial ATA (SATA, abbreviated from Serial AT Attachment)[2] is a computer bus interface that connects host bus adapters to mass storage devices such as hard disk drives, optical drives, and solid-state drives.

  连接主机总线控制器和大规模存储设备

第08章 第9章

- **Q：AHCI规范的作用是什么？**
- **A：**

（wikipedia）

The Advanced Host Controller Interface (AHCI) is a technical standard defined by Intel that specifies the operation of Serial ATA (SATA) host bus adapters in a non-implementation-specific manner.

The specification describes a system memory structure for computer hardware vendors to exchange data between host system memory and attached storage devices. AHCI gives software developers and hardware designers a standard method for detecting, configuring, and programming SATA/AHCI adapters.

（课本8.5.4）

为驱动程序提供一个支持异步命令-响应机制的队列接口。

第08章 第10题

- **Q：阅读Linux内核中SATA驱动程序。**
- **A：**

  给予认真的同学加分

第09章 第1题

- **Q：什么情况下需要对cache进行初始化？cache初始化过程中所使用的cache指令index store tag的作用是什么？列举一种MIPS的其他cache指令并解释。**
- **A：**
  - **1. 有cache、用cache（硬件电路初始化和软件初始化）**
  - **2. Index Store Tag是通过Index将指定路的Cache行的TAG域初始化为指定的数据（TagHi、TagLo）。**
  - **3. MIPS的cache指令有很多种，可以按照两种正交的方式分类： (a)操作哪个cache (b)什么类型的操作**
    - **其中操作什么cache分为Icache Dcache L2和L3四类**
    - **什么操作，分为index无效、hit无效、预取、读写tag等**

第09章 第1题-A

- 例如：
  - **Index Invalidate D: 通过Index将指定的路的Dcache无效**
  - **Hit Invalidate Writeback L2：根据地址，判断地址是否在L2中命中，如果命中，则将该cache行无效并写回到（L3/内存）**
  - **Fetch and lock I：根据地址，判断地址是否在Icache中，如果不在，将该地址取回并放入Icache中，并锁定**
  - **Index Load Tag L3：通过Index将指定的路的L3的tag内容读出，放到（TagHi、TagLo）中**

第09章 第2题

- **Q：cache初始化和内存初始化的目的有什么不同？系统如何识别内存的更换？**
- **A：**
  - **1. cache初始化是为了在使用cache时，不会因为cache内的随机数据导致访问内容出错，是对cache存储内容的初始化。内存初始化主要是对内存控制器访问内存的时序控制进行初始化，是对内存访问链路的初始化。**
  - **2.系统通过I2C总线对外部内存条的SPD芯片进行读取，比对其中的内容与上一次记录的内容是否相同，来识别内存的更换。**

第09章 第3题

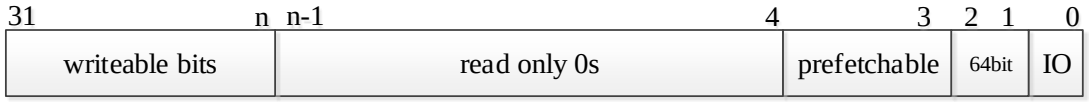
- **Q：从HyperTransport配置地址空间的划分上，计算整个系统能支持的总线数量，设备数量及功能数量。**
- **A：**

	位域	位宽	总数
Bus	23:16	8	256
Device	15:11	5	256*32
Func	10:8	3	256*32*8

第09章 第4题

- **Q：根据PCI地址空间的命中方法和BAR的配置方式，给出PCI地址空间的命中公式。**
- **A：高位31:n是软件可配的基地址，同时也规定了地址空间的范围。命中就是指访问的地址(Addr_in)落在该范围中。由此，命中的公式是**

  HIT = BAR[31:n] == Addr_in[31:n]



第09章 第5题

- **Q：多核唤醒时，如果采用核间中断方式，从核的唤醒流程是怎样的？**
- **A：当采用核间中断方式，意味着从核不再需要一直执行轮询操作。那么此时从核在等待唤醒之前只需要使能中断，然后就可以进行一些其它的任何操作，例如单纯的WAIT指令。而主核向从核发出核间中断之后，从核将首先进入中断入口，当判断为核间中断时，再跳转至唤醒处理程序开始执行即可。之后的流程与采用轮询的方式一致。**

第10章 第1题-Q

- **Q：请将下列无符号数据在不同的进制表达间进行转换。**
  - ① 二进制转换为十进制：101011_2、001101_2、01011010_2、0000111010000101_2
  - ② 十进制转换为二进制：42_{10}、79_{10}、811_{10}、374_{10}
  - ③ 十六进制转换为十进制：$8AE_{16}$、$C18D_{16}$、$B379_{16}$、100_{16}
  - ④ 十进制转换为十六进制：81783_{10}、1922_{10}、345208_{10}、5756_{10}

第10章 第1题-A

- **A：**
  - ① 43_{10}、13_{10}、90_{10}、3717_{10}
  - ② 101010_{2}、1001111_{2}、1100101011_{2}、101110110_{2}
  - ③ 2222_{10}、49549_{10}、45945_{10}、256_{10}
  - ④ $13F77_{16}$、782_{16}、54478_{16}、$167C_{16}$

第10章 第2题

- **Q：请给出32位二进制数分别视为无符号数、原码、补码时所表示的数的范围。**
- **A：**
  - 无符号数：$[0,2^{32}-1]$
  - 原码　　：$[1-2^{31},2^{31}-1]$
  - 补码　　：$[-2^{31},2^{31}-1]$

第10章 第3题

- **Q：请将下列十进制数表示为8位原码和8位补码，或者表明该数据会溢出：45_{10}、-59_{10}、-128_{10}、119_{10}、127_{10}、128_{10}、0_{10}、-1_{10}**
- **A:**

十进制数	8位原码	8位补码
45_{10}	0010 1101	0010 1101
-59_{10}	1011 1011	1100 0101
-128_{10}	溢出	1000 0000
119_{10}	0111 0111	0111 0111
127_{10}	0111 1111	0111 1111
128_{10}	溢出	溢出
0_{10}	0000 0000/1000 0000	0000 0000
-1_{10}	1000 0001	1111 1111

第10章 第4题

- **Q：请将下列数据分别视作原码和补码，从8位扩展为16位：00101100_2、11010100_2、10000001_2、00010111_2。**
- **A：**

原8位数	16位原码	16位补码
0010 1100	0000 0000 0010 1100	0000 0000 0010 1100
1101 0100	1000 0000 0101 0100	1111 1111 1101 0100
1000 0001	1000 0000 0000 0001	1111 1111 1000 0001
0001 0111	0000 0000 0001 0111	0000 0000 0001 0111

第10章 第5题-Q

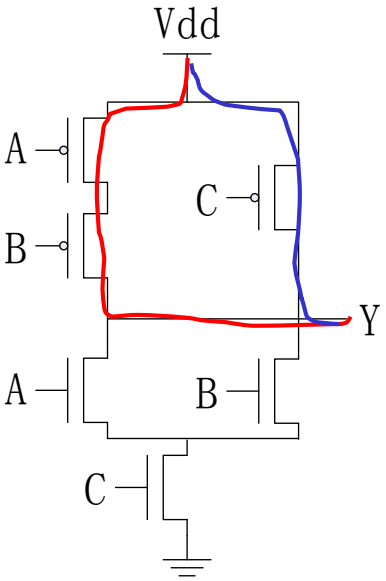
- **Q：请将下列浮点数在不同进制间进行转换。**
  - ① **十进制数转换为单精度数：0、116.25、-4.375**
  - ② **十进制数转换为双精度数：-0、116.25、-2049.5**
  - ③ **单精度数转换为十进制数：0xff800000、0x7fe00000**
  - ④ **双精度数转换为十进制数：0x8008000000000000、0x7065020000000000**

第10章 第5题-A

- **A：**
  - ① **0x0、0x42E88000、0xC08C0000**
  - ② **0x8000000000000000、0x405D100000000000、0xC0A0030000000000**
  - ③ 负无穷、非数
  - ④ **-2^{-1023}(非规格化数) 、$2^{775}+2^{773}+2^{771}+2^{764}$**

第10章 第6题

- **Q：请写出下图所示晶体管级电路图的真值表，并给出对应的逻辑表达式。**



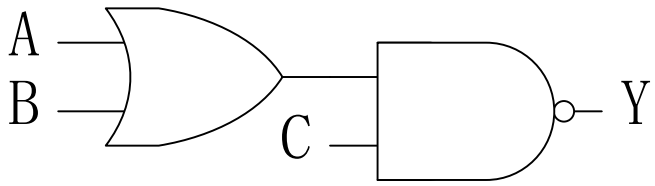
- **A：**

A	B	C	Y
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

Y = ~C | ~A & ~B

第10章 第7题

- **Q：请写出下图所示逻辑门电路的真值表。**



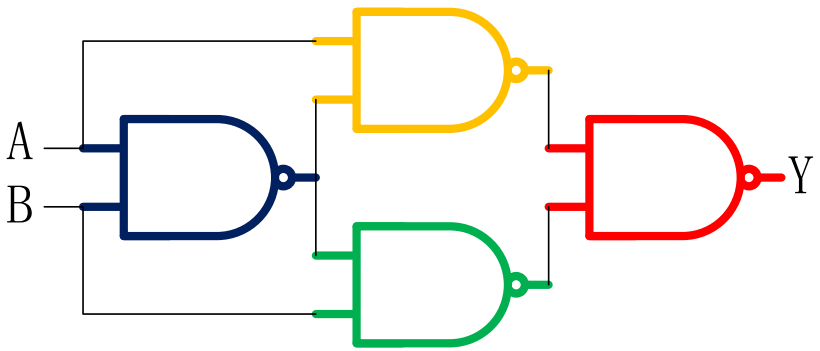
- **A:**

A	B	C	Y
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

第10章 第8题

- **Q：请用尽可能少的2输入NAND门搭建出一个具有2输入XOR功能的电路。**
- **A：**

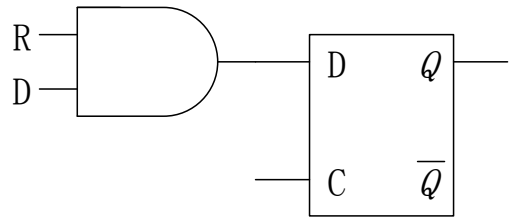
```
Y = ~A & B | A & ~B
  = ~(~(~A&B) & ~(A&~B))
  = ~(~(~(A&B)&B) & ~(A&~(A&B)))
```



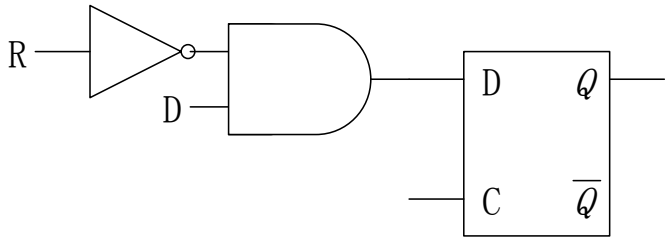
第10章 第9题

- **Q：请用D触发器和常见组合逻辑门搭建出一个具有同步复位为0功能的触发器的电路。**
- **A：**
  - **同步：有时钟才有复位效果**
  - **复位：R（0有效/1有效）**
  - **为0 ：R有效时输入为0**

R为0有效：Q <= R & D;



R为1有效：Q <= ~R & D;



Q & A