

北京大学信息科学技术学院考试试卷

考试科目： 计算机组成 姓名： _____ 学号： _____

考试时间： 2017 年 6 月 19 日 任课教师： 陆俊林

题号	一	二	三	四	五	六	七	八	总分
分数									
阅卷人									

北京大学考场纪律

1、考生进入考场后，按照监考老师安排隔位就座，将学生证放在桌面上。无学生证者不能参加考试；迟到超过 15 分钟不得入场。在考试开始 30 分钟后方可交卷出场。

2、除必要的文具和主考教师允许的工具书、参考书、计算器以外，其它所有物品（包括空白纸张、手机、或有存储、编程、查询功能的电子用品等）不得带入座位，已经带入考场的必须放在监考人员指定的位置。

3、考试使用的试题、答卷、草稿纸由监考人员统一发放，考试结束时收回，一律不准带出考场。若有试题印制问题请向监考教师提出，不得向其他考生询问。提前答完试卷，应举手示意请监考人员收卷后方可离开；交卷后不得在考场内逗留或在附近高声交谈。未交卷擅自离开考场，不得重新进入考场答卷。考试结束时间到，考生立即停止答卷，在座位上等待监考人员收卷清点后，方可离场。

4、考生要严格遵守考场规则，在规定时间内独立完成答卷。不准交头接耳，不准偷看、夹带、抄袭或者有意让他人抄袭答题内容，不准接传答案或者试卷等。凡有违纪作弊者，一经发现，当场取消其考试资格，并根据《北京大学本科考试工作与学术规范条例》及相关规定严肃处理。

5、考生须确认自己填写的个人信息真实、准确，并承担信息填写错误带来的一切责任与后果。

学校倡议所有考生以北京大学学生的荣誉与诚信答卷，共同维护北京大学的学术声誉。

以下为试题和答题纸，共 14 页。

得分

一、填空题（每空 1 分，共 15 分）

1. 冯·诺依曼计算机结构五个基本组成部分：_____、_____、_____、输入设备和输出设备。
2. 强调高速输入输出、海量数据存储空间和并行事务处理等方面特性，以面向大容量数据的事务处理为主、兼顾科学计算的计算机应属于_____。
3. 为了从存储器中取出指令，控制器通常把 PC 的内容加上一个数值，形成下一条指令的地址，但在遇到转移指令时，控制器则把_____送入 PC。
4. 常见的 I/O 端口编址方式包括：x86 体系结构使用的_____方式和 ARM 体系结构使用的_____方式。
5. 8086 系统的中断向量表中，若从 0000H:003CH 单元开始由低地址到高地址依次存放 30H、40H、00H 和 B0H 四个字节，则相应的中断类型码和中断服务程序的入口地址分别为_____和_____。
6. 转移预测主要包括对_____的预测和对_____的预测。
7. 返回地址栈 RAS 中保存了_____指令的地址。
8. IBM PC XT/AT 中，8253 通道 1 工作在方式 2 专用于 DRAM 的刷新。如果 8253 的输入时钟为 1.19318MHz，DRAM 需要每 2ms 刷新 128 次，则计数初值应设为_____。
9. 64-bit 数据宽度的 DDR3-1600 SDRAM 的峰值带宽为 12.8GB/s，接口时钟频率为 800MHz，芯片内部采用了 8 位数据预取技术，其核心频率为_____MHz。
10. 设异步传输时的比特率为 2400bps，若每个字符对应一位起始位、七位有效数据位、一位校验位和一位停止位，则每秒钟传输的字符数最多为_____个。

得分

第二题（10 分）

汇编语言编程

用 MIPS 基本指令能够容易地判断“小于”(<), 那如何实现>、≥和≤的判断? 请分别写出对应的 MIPS 汇编语言代码, 并加上注释。每种情况最多用两条指令完成, 可用指令参见下页附加材料。

(1) 实现>的判断

C 语言代码:

```
if (g > h)    goto Greater;    # g:$s0, h:$s1
```

MIPS 汇编代码:

(2) 实现≥的判断

C 语言代码:

```
if (g >= h)   goto GreaterEqual;    # g:$s0, h:$s1
```

MIPS 汇编代码:

(3) 实现≤的判断

C 语言代码:

```
if (g <= h)   goto LessEqual;    # g:$s0, h:$s1
```

MIPS 汇编代码:

MIPS Reference Data

①



CORE INSTRUCTION SET				OPCODE / FUNCT (Hex)
NAME, MNEMONIC	FOR- MAT	OPERATION (in Verilog)		
Add	add	R	$R[rd] = R[rs] + R[rt]$	(1) 0 / 20 _{hex}
Add Immediate	addi	I	$R[rt] = R[rs] + \text{SignExtImm}$	(1,2) 8 _{hex}
Add Imm. Unsigned	addiu	I	$R[rt] = R[rs] + \text{SignExtImm}$	(2) 9 _{hex}
Add Unsigned	addu	R	$R[rd] = R[rs] + R[rt]$	0 / 21 _{hex}
And	and	R	$R[rd] = R[rs] \& R[rt]$	0 / 24 _{hex}
And Immediate	andi	I	$R[rt] = R[rs] \& \text{ZeroExtImm}$	(3) c _{hex}
Branch On Equal	beq	I	if($R[rs] == R[rt]$) $PC = PC + 4 + \text{BranchAddr}$	(4) 4 _{hex}
Branch On Not Equal	bne	I	if($R[rs] != R[rt]$) $PC = PC + 4 + \text{BranchAddr}$	(4) 5 _{hex}
Jump	j	J	$PC = \text{JumpAddr}$	(5) 2 _{hex}
Jump And Link	jal	J	$R[31] = PC + 8; PC = \text{JumpAddr}$	(5) 3 _{hex}
Jump Register	jr	R	$PC = R[rs]$	0 / 08 _{hex}
Load Byte Unsigned	lbu	I	$R[rt] = \{24'b0, M[R[rs] + \text{SignExtImm}](7:0)\}$	(2) 24 _{hex}
Load Halfword Unsigned	lhu	I	$R[rt] = \{16'b0, M[R[rs] + \text{SignExtImm}](15:0)\}$	(2) 25 _{hex}
Load Linked	ll	I	$R[rt] = M[R[rs] + \text{SignExtImm}]$	(2,7) 30 _{hex}
Load Upper Imm.	lui	I	$R[rt] = \{\text{imm}, 16'b0\}$	f _{hex}
Load Word	lw	I	$R[rt] = M[R[rs] + \text{SignExtImm}]$	(2) 23 _{hex}
Nor	nor	R	$R[rd] = \sim (R[rs] R[rt])$	0 / 27 _{hex}
Or	or	R	$R[rd] = R[rs] R[rt]$	0 / 25 _{hex}
Or Immediate	ori	I	$R[rt] = R[rs] \text{ZeroExtImm}$	(3) d _{hex}
Set Less Than	slt	R	$R[rd] = (R[rs] < R[rt]) ? 1 : 0$	0 / 2a _{hex}
Set Less Than Imm.	slti	I	$R[rt] = (R[rs] < \text{SignExtImm}) ? 1 : 0$	(2) a _{hex}
Set Less Than Imm. Unsigned	sltiu	I	$R[rt] = (R[rs] < \text{SignExtImm}) ? 1 : 0$	(2,6) b _{hex}
Set Less Than Unsig.	sltu	R	$R[rd] = (R[rs] < R[rt]) ? 1 : 0$	(6) 0 / 2b _{hex}
Shift Left Logical	sll	R	$R[rd] = R[rt] \ll \text{shamt}$	0 / 00 _{hex}

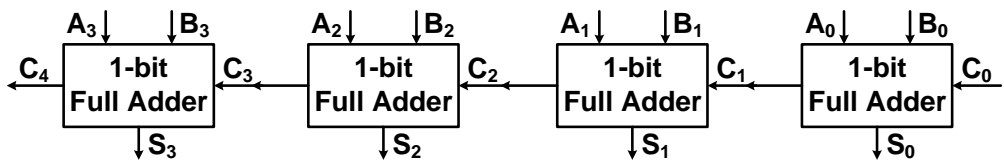
得分

第三题（15 分）

加法器

1. 请画出 1 位全加器的门电路实现结构。

2. 下图是 4 个全加器组成的 4 位行波进位加法器，请结合第 1 小题，给出其关键路径的门延迟数，并说明计算方法，然后推导出 n 位行波进位加法器的门延迟通用表达式。



答：

3. 根据第 2 小题图，给出第 i 个全加器的进位输出 C_{i+1} 的产生逻辑表达式。

答：

4. 如果采用“超前进位”的方式，则请写出 C_4 的产生逻辑表达式。需从 A_i 和 B_i 开始，写明推导的过程。

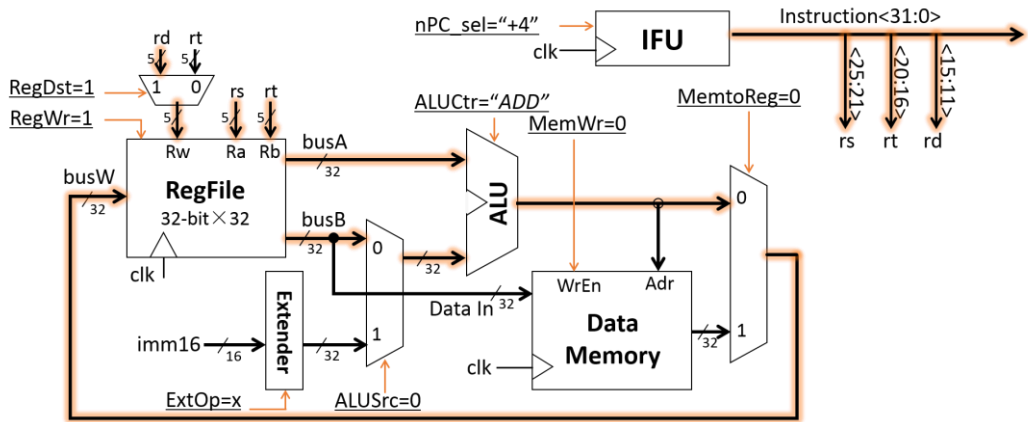
答：

得分

第四题（15 分）

控制器

下图为单周期 MIPS 处理器的控制信号和数据通路图，并以 add 指令为例标出了控制信号的值，请据此回答如下问题。



1. 请分析不同指令所需的控制信号的值，并将下表填写完整

func	100000	100010	/			
opcode (op)	000000	000000	001101	100011	101011	000100
	add	sub	ori	lw	sw	beq
RegDst	1			0		x
ALUSrc	0	0		1		
MemtoReg	0	0			x	x
RegWr	1		1		0	
MemWr	0		0		1	0
nPC_sel	0			0		
ExtOp	x			1		
ALUctr<1:0>	00 (ADD)	01 (SUB)	10 (OR)			

2. 请写出 ALUSrc、RegWr 信号的逻辑表达式，输入信号为指令编码，对应上表内容分别记为：op5-op0、func5-func0。

3. 请画出 ExtOp 信号的门级电路图，可以使用与、或、非门，门的输入信号数量不限。电路图应尽量直观的体现该信号的产生逻辑，不必进行优化。

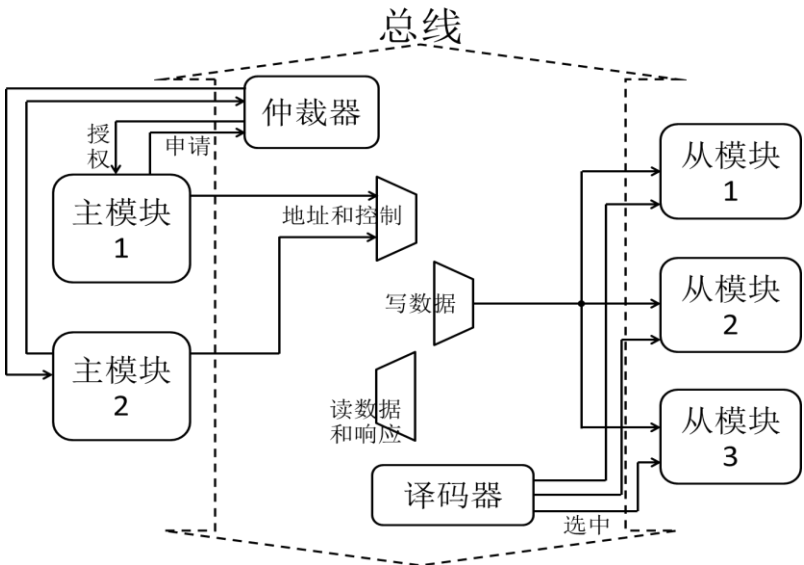
得分

第五题（25 分）

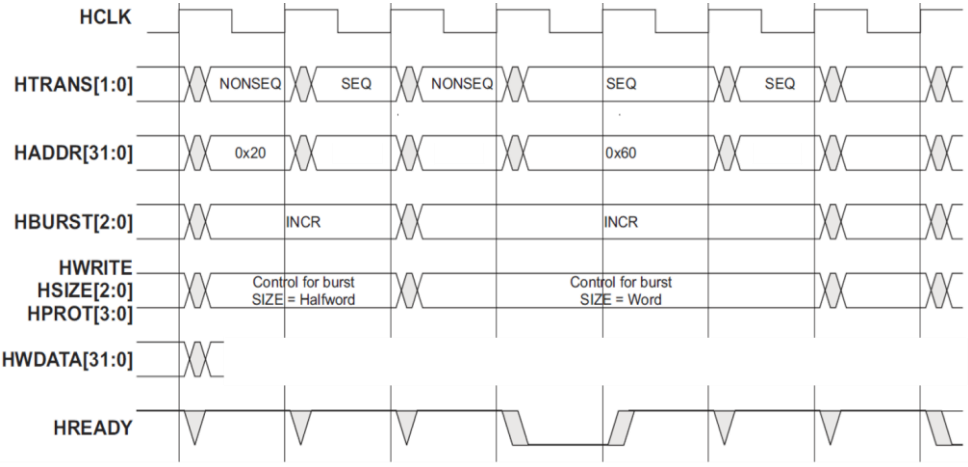
总线

根据 AMBA2.0 AHB 总线规范，分析并回答如下问题。

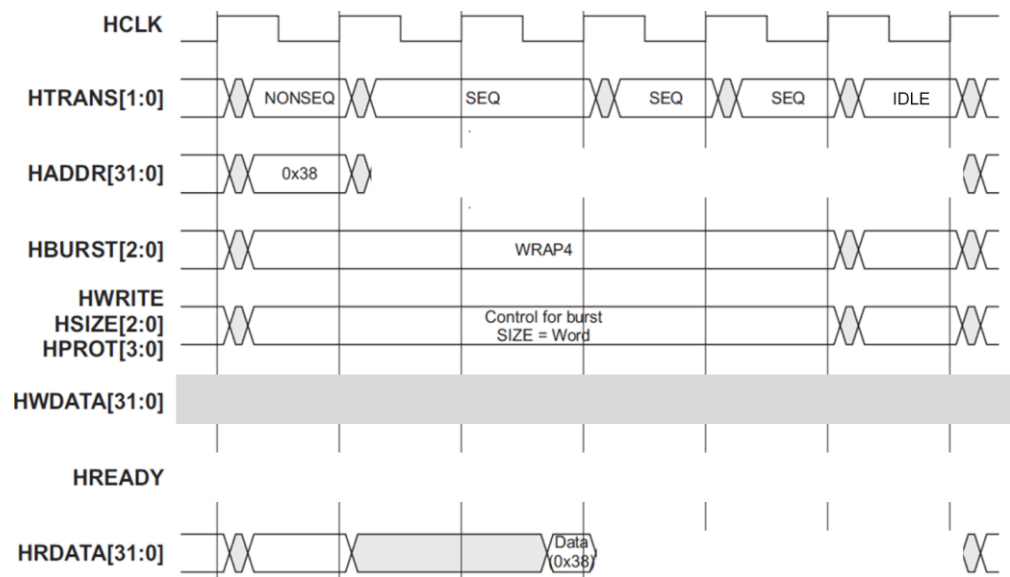
1. 下图是一个 AHB 总线系统的结构图，包含 2 个主模块和 3 个从模块，请补全该结构图中的连线。



2. 下图是 AHB 总线上的连续两次“不定长的写传输”，请据此补全 HADDR[31:0] 信号的波形。



3. 下图是 AHB 总线上的一次“突发读传输”，地址从 0x38 开始，数据记为 Data(地址)（例如地址 0x38 对应的数据记为 Data(0x38)）。请据此补全 HADDR[31:0]信号、HREADY 信号和 HRDATA[31:0]信号。

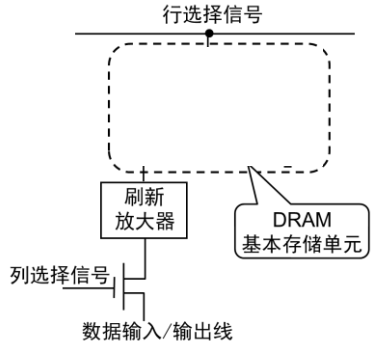


得分

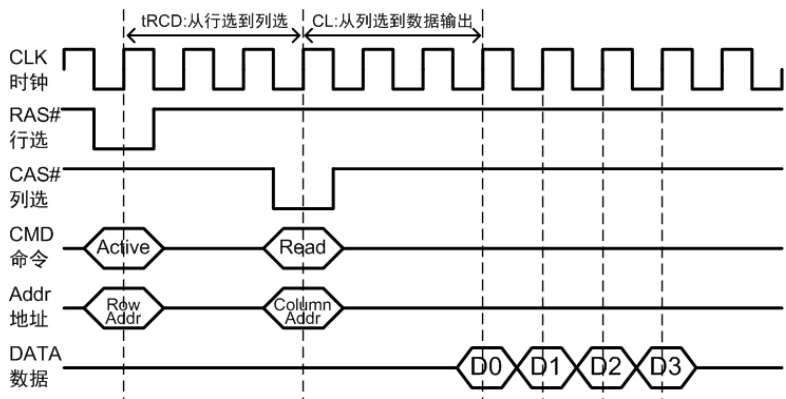
第六题（8分）

存储器

1、下图为 DRAM 基本单元的结构，请补全其中缺失的部分。



2、如果用 SDRAM 作为计算机系统的内存，则一次典型的读取内存的数据传输过程如下图所示。



(1) 请从 SDRAM 芯片内部结构分析，为什么一次内存访问需要先后发出两次地址（Row Addr 和 Column Addr）？

答：

(2) 内存访问什么情况下不需要发出两次地址？读取数据的耗时是多少？

答：

得分

第八题（12 分）

I/O 接口

在 IBM PC 系统中，有一段 x86 汇编语言程序。请在每条 OUT 指令后加注释，详细说明其作用。

```

OUT  0DH,  AL  ;
MOV  AL,  08H  ;
OUT  83H,  AL
MOV  AL,  00H  ;
OUT  02H,  AL
MOV  AL,  04H  ;
OUT  02H,  AL
MOV  AL,  00H  ;
OUT  03H,  AL
MOV  AL,  30H  ;
OUT  03H,  AL
MOV  AL,  89H  ;
OUT  0BH,  AL
MOV  AL,  80H  ;
OUT  08H,  AL
MOV  AL,  01H  ;
OUT  0AH,  AL

```

最后，请分析说明这段程序的作用。

答：

附加材料 1：IBM PC 系统地址空间分配

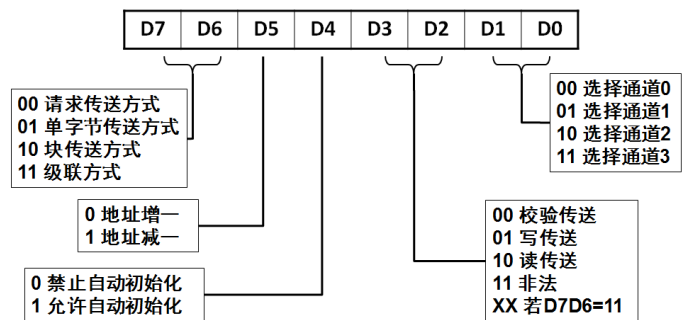
地址空间	器件/接口适配器	实际使用端口地址
0000~001FH	DMA 控制器 8237	00~0FH
0020~003FH	中断控制器 8259A	20~21H
0040~005FH	计数器/定时器 8253	40~43H
0060~007FH	并行接口片 8255A	60~63H
0080~009FH	DMA 页面寄存器（74LS670）	80~83H
00A0~00BFH	NMI 寄存器	A0H

附加材料 2：DMA 控制器 8237

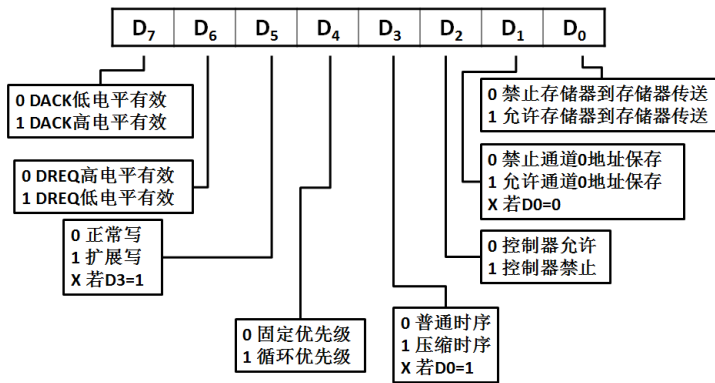
内部端口

A ₃	A ₂	A ₁	A ₀	通道号	读操作(IOR)	写操作 (IOW)
0	0	0	0	0	读当前地址寄存器	写基和当前地址寄存器
0	0	0	1		读当前字节计数寄存器	写基和当前字节计数寄存器
0	0	1	0	1	读当前地址寄存器	写基和当前地址寄存器
0	0	1	1		读当前字节计数寄存器	写基和当前字节计数寄存器
0	1	0	0	2	读当前地址寄存器	写基和当前地址寄存器
0	1	0	1		读当前字节计数寄存器	写基和当前字节计数寄存器
0	1	1	0	3	读当前地址寄存器	写基和当前地址寄存器
0	1	1	1		读当前字节计数寄存器	写基和当前字节计数寄存器
1	0	0	0	四个通道公用	读状态寄存器	写命令寄存器
1	0	0	1		— —	写请求寄存器
1	0	1	0		— —	写屏蔽寄存器某一位
1	0	1	1		— —	写模式寄存器
1	1	0	0		— —	清除高低位触发器命令
1	1	0	1		读暂存寄存器	主清除命令
1	1	1	0		— —	— —
1	1	1	1		— —	写屏蔽寄存器所有位

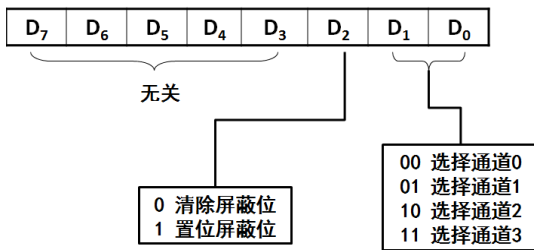
模式字



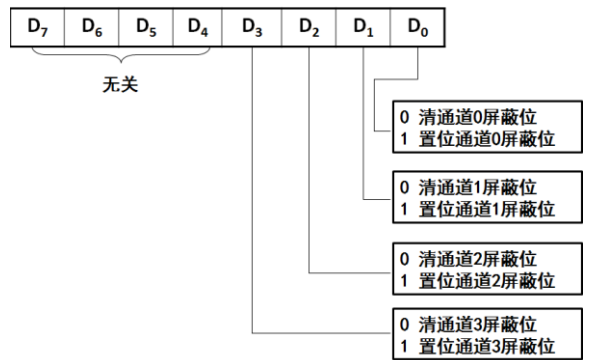
命令字



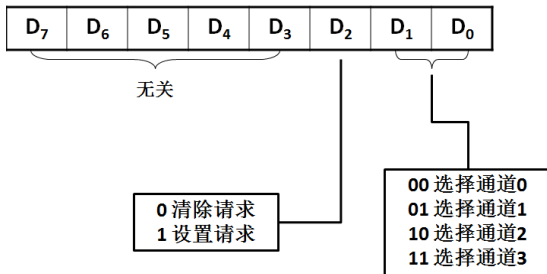
屏蔽字 格式 1



屏蔽字 格式 2



请求字



状态字

