

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 4 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 5 页。

得分

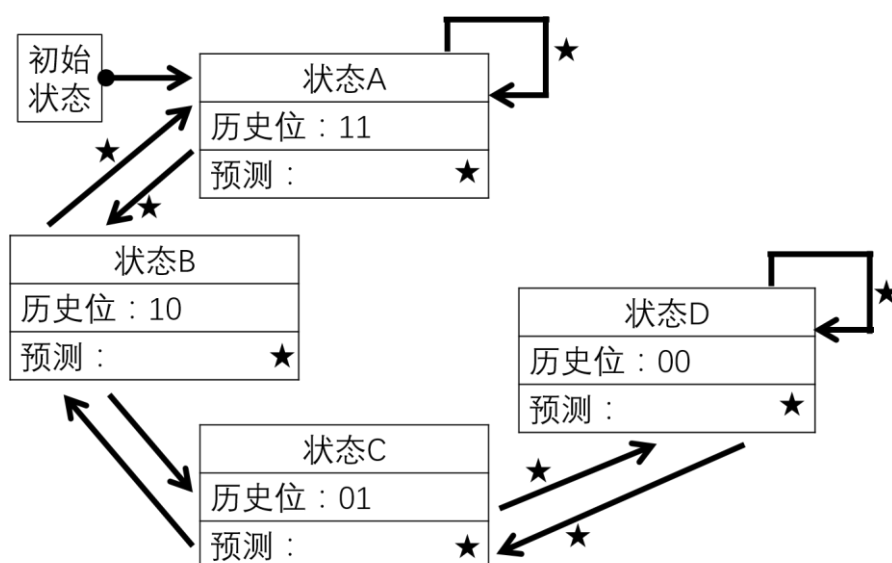
一、填空题（每空 2 分，共 20 分）

1. 指令执行的四个基本步骤为：取指、_____、执行、_____。
2. 通常认为第一台存储程序式的电子计算机是_____。
3. 冯·诺依曼计算机结构五个基本组成部分的是_____、控制器、_____、_____和_____。
4. 冯·诺依曼计算机结构的控制器有两种常见类型，即_____和_____，其中后者的指令执行速度快，但是控制逻辑的电路复杂、设计和验证的难度大。
5. 在 x86 实模式下，假设 CS=3000H，DS=2000H，则指令“MOV AX, [3000H]”执行时访问的内存地址是_____。

得分

第二题（共 10 分）

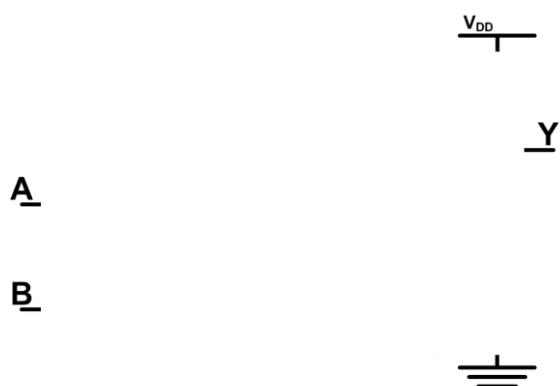
下图是 Intel Pentium 的两位历史信息转移预测器的状态转换图，请在图中所有标五角星处填上相关信息：在表格中填写预测是否发生转移，在带箭头的线上填写状态转换条件。



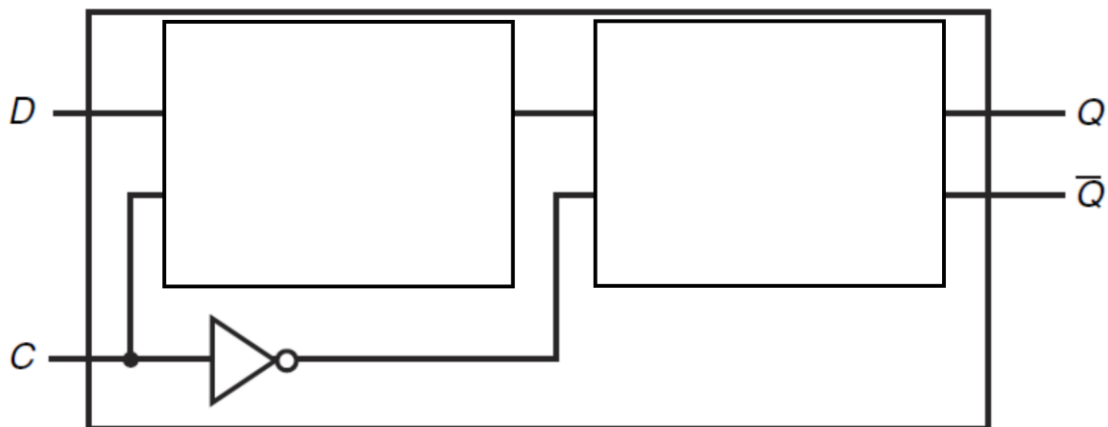
得分

第三题 (20 分)

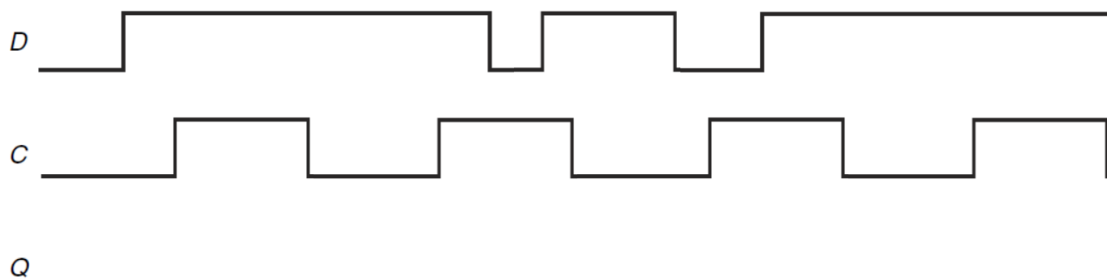
- 1、请以 PMOS 和 NMOS 为基本单元，在下面的框架基础上，画出或非门的结构图。



- 2、请以逻辑门为基本单元，在下面的框架基础上，画出下降沿触发的 D 触发器的实现结构。



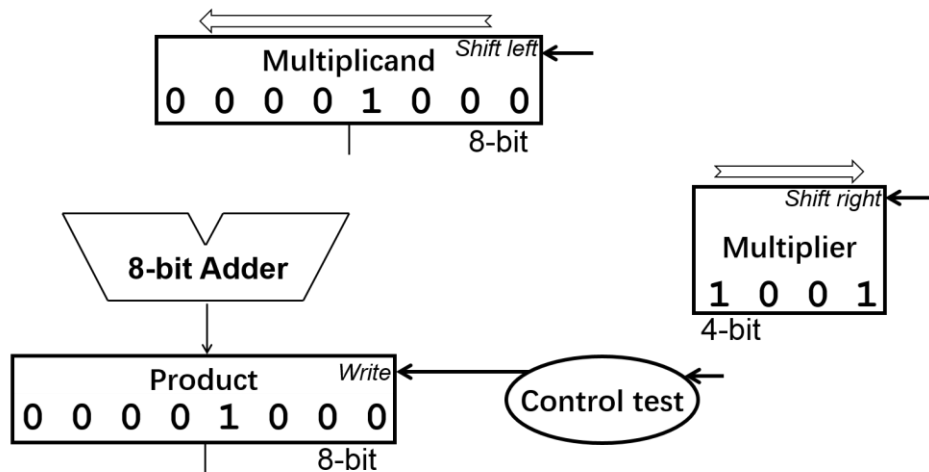
- 3、请根据上升沿触发的 D 触发器的特性，在下面时序图的基础上，画出 Q 信号的波形。



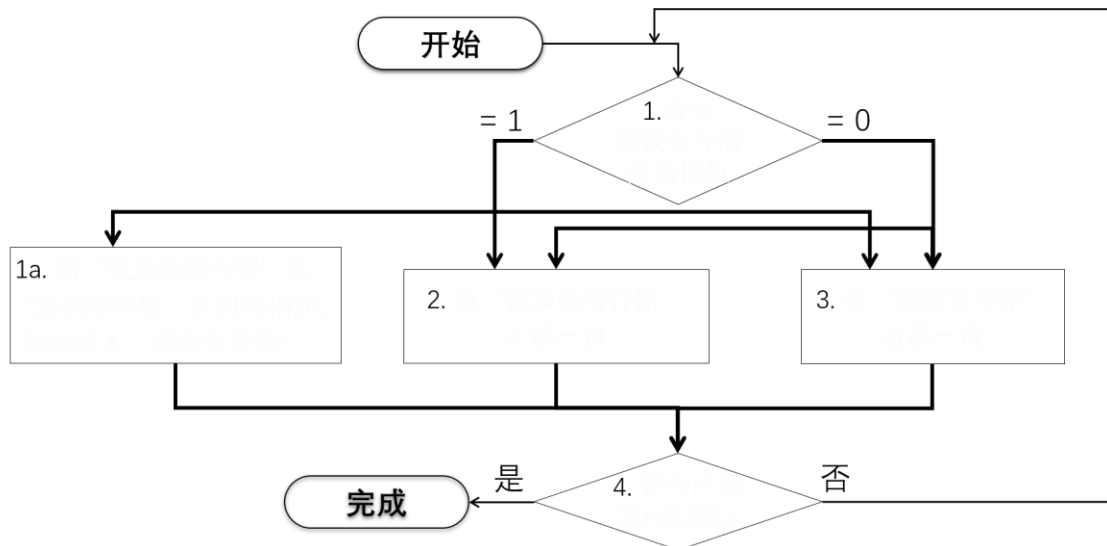
得分

第四题（20 分）

1、下图是一种乘法器的实现结构，请补全其中的连线。



2、对应上图的乘法器实现结构，请在下图中补全乘法器的工作流程。



得分

第五题 (30 分)

1、现有一段 MIPS 机器代码，见下表 B 列；其汇编语言源代码已残缺不全，见下表 A 列。请以 A 列为参考，手工反汇编 B 列，填写在 C 列。C 列中标星号 (*) 的空格必须填写。其它空格不计分，可不填，但对回答本大题中后续小题有作用。

	汇编语言代码 (A 列)	机器语言代码(B 列)	反汇编代码 (C 列)
1.	l [redacted] 0x1 [redacted]	3c011001	
2.	[redacted]	342e0040	
3.	[redacted] \$t7, 1	adc00000	*
4.	sw \$t [redacted]	240f0001	*
5.	[redacted] \$t3, 0	adcf0004	
6.	[redacted]	240b0000	
7.	[redacted] \$t5, 8	240c0004	
8.	[redacted]	240d0008	
9.	[redacted]	2418002c	
10.	LOOP:	24190014	*
11.	[redacted], \$t3	01cb7820	*
12.	[redacted]	8de80000	*
13.	add \$t7, \$t6, \$t4	01cc7820	
14.	lw [redacted]	8de90000	
15.	[redacted] \$t6, \$t5	01cd7820	
16.	[redacted]	01095020	*
17.	[redacted]	adea0000	*
18.	[redacted]	000c5821	*
19.	move [redacted]	000d6021	
20.	[redacted]	21ad0004	
21.	[redacted] EXIT	01b8082a	*
22.	b [redacted]	10200002	*
23.	EXIT:	0159082a	*
24.		1420fff2	*

2. 请问 A 列第 1 行的代码应该是什么？

答：

3、这段代码最有可能的功能是什么？

答：

MIPS Reference Data

①



CORE INSTRUCTION SET

NAME, MNEMONIC	FOR-MAT	OPERATION (in Verilog)	OPCODE / FUNCT (Hex)
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	$R[rd] = \sim (R[rs] \mid R[rt])$	0 / 27 _{hex}
Or	or R	$R[rd] = R[rs] \mid R[rt]$	0 / 25 _{hex}
Or Immediate	ori I	$R[rt] = R[rs] \mid \text{ZeroExtImm}$	(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	$R[rt] = (R[rs] < \text{SignExtImm}) ? 1 : 0$	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}
Shift Right Logical	srl R	$R[rd] = R[rt] \gg \text{shamt}$	0 / 02 _{hex}
Store Byte	sb I	$M[R[rs] + \text{SignExtImm}(7:0)] = R[rt](7:0)$	(2) 28 _{hex}
Store Conditional	sc I	$M[R[rs] + \text{SignExtImm}] = R[rt];$ $R[rt] = (\text{atomic}) ? 1 : 0$	(2,7) 38 _{hex}
Store Halfword	sh I	$M[R[rs] + \text{SignExtImm}(15:0)] = R[rt](15:0)$	(2) 29 _{hex}
Store Word	sw I	$M[R[rs] + \text{SignExtImm}] = R[rt]$	(2) 2b _{hex}
Subtract	sub R	$R[rd] = R[rs] - R[rt]$	(1) 0 / 22 _{hex}
Subtract Unsigned	subu R	$R[rd] = R[rs] - R[rt]$	0 / 23 _{hex}

- (1) May cause overflow exception
- (2) $\text{SignExtImm} = \{16\{\text{immediate}[15]\}, \text{immediate}\}$
- (3) $\text{ZeroExtImm} = \{16\{1b'0\}, \text{immediate}\}$
- (4) $\text{BranchAddr} = \{14\{\text{immediate}[15]\}, \text{immediate}, 2'b0\}$
- (5) $\text{JumpAddr} = \{PC + 4[31:28], \text{address}, 2'b0\}$
- (6) Operands considered unsigned numbers (vs. 2's comp.)
- (7) Atomic test&set pair; $R[rt] = 1$ if pair atomic, 0 if not atomic

BASIC INSTRUCTION FORMATS

R	opcode	rs	rt	rd	shamt	funct
	31	26 25	21 20	16 15	11 10	6 5
	31	26 25	21 20	16 15		
I	opcode	rs	rt	immediate		
	31	26 25	21 20	16 15		
J	opcode	address				
	31	26 25				

ARITHMETIC CORE INSTRUCTION SET

NAME, MNEMONIC	FOR-MAT	OPERATION	OPCODE / FUNCT (Hex)
Branch On FP True	bc1t FI	if($FPcond$) $PC = PC + 4 + \text{BranchAddr}$	(4) 11/8/1/--
Branch On FP False	bc1f FI	if(! $FPcond$) $PC = PC + 4 + \text{BranchAddr}$	(4) 11/8/0/--
Divide	div R	$Lo = R[rs]/R[rt]; Hi = R[rs]\%R[rt]$	0/--/1a
Divide Unsigned	divu R	$Lo = R[rs]/R[rt]; Hi = R[rs]\%R[rt]$	(6) 0/--/1b
FP Add Single	add.s FR	$F[fd] = F[fs] + F[ft]$	11/10/--/0
FP Add Double	add.d FR	$\{F[fd], F[fd+1]\} = \{F[fs], F[fs+1]\} + \{F[ft], F[ft+1]\}$	11/11/--/0
FP Compare Single	c.x.s* FR	$FPcond = (F[fs] \text{ op } F[ft]) ? 1 : 0$	11/10/--/y
FP Compare Double	c.x.d* FR	$FPcond = (\{F[fs], F[fs+1]\} \text{ op } \{F[ft], F[ft+1]\}) ? 1 : 0$	11/11/--/y
* (x is eq, lt, or le) (op is ==, <, or <=) (y is 32, 3c, or 3e)			
FP Divide Single	div.s FR	$F[fd] = F[fs] / F[ft]$	11/10/--/3
FP Divide Double	div.d FR	$\{F[fd], F[fd+1]\} = \{F[fs], F[fs+1]\} / \{F[ft], F[ft+1]\}$	11/11/--/3
FP Multiply Single	mul.s FR	$F[fd] = F[fs] * F[ft]$	11/10/--/2
FP Multiply Double	mul.d FR	$\{F[fd], F[fd+1]\} = \{F[fs], F[fs+1]\} * \{F[ft], F[ft+1]\}$	11/11/--/2
FP Subtract Single	sub.s FR	$F[fd] = F[fs] - F[ft]$	11/10/--/1
FP Subtract Double	sub.d FR	$\{F[fd], F[fd+1]\} = \{F[fs], F[fs+1]\} - \{F[ft], F[ft+1]\}$	11/11/--/1
Load FP Single	lwc1 I	$F[rt] = M[R[rs] + \text{SignExtImm}]$	(2) 31/--/--
Load FP Double	ldc1 I	$F[rt] = M[R[rs] + \text{SignExtImm}];$ $F[rt+1] = M[R[rs] + \text{SignExtImm} + 4]$	(2) 35/--/--
Move From Hi	mfhi R	$R[rd] = Hi$	0/--/--/10
Move From Lo	mflo R	$R[rd] = Lo$	0/--/--/12
Move From Control	mfc0 R	$R[rd] = CR[rs]$	10/0/--/0
Multiply	mult R	$\{Hi, Lo\} = R[rs] * R[rt]$	0/--/--/18
Multiply Unsigned	multu R	$\{Hi, Lo\} = R[rs] * R[rt]$	(6) 0/--/--/19
Shift Right Arith.	sra R	$R[rd] = R[rt] \gg \text{shamt}$	0/--/--/3
Store FP Single	swc1 I	$M[R[rs] + \text{SignExtImm}] = F[rt]$	(2) 39/--/--
Store FP Double	sdc1 I	$M[R[rs] + \text{SignExtImm}] = F[rt];$ $M[R[rs] + \text{SignExtImm} + 4] = F[rt+1]$	(2) 3d/--/--

FLOATING-POINT INSTRUCTION FORMATS

FR	opcode	fmt	ft	fs	fd	funct
	31	26 25	21 20	16 15	11 10	6 5
FI	opcode	fmt	ft	immediate		
	31	26 25	21 20	16 15		

PSEUDOINSTRUCTION SET

NAME	MNEMONIC	OPERATION
Branch Less Than	blt	if($R[rs] < R[rt]$) $PC = \text{Label}$
Branch Greater Than	bgt	if($R[rs] > R[rt]$) $PC = \text{Label}$
Branch Less Than or Equal	ble	if($R[rs] \leq R[rt]$) $PC = \text{Label}$
Branch Greater Than or Equal	bge	if($R[rs] \geq R[rt]$) $PC = \text{Label}$
Load Immediate	li	$R[rd] = \text{immediate}$
Move	move	$R[rd] = R[rs]$

REGISTER NAME, NUMBER, USE, CALL CONVENTION

NAME	NUMBER	USE	PRESERVED ACROSS A CALL?
\$zero	0	The Constant Value 0	N.A.
\$at	1	Assembler Temporary	No
\$v0-\$v1	2-3	Values for Function Results and Expression Evaluation	No
\$a0-\$a3	4-7	Arguments	No
\$t0-\$t7	8-15	Temporaries	No
\$s0-\$s7	16-23	Saved Temporaries	Yes
\$t8-\$t9	24-25	Temporaries	No
\$k0-\$k1	26-27	Reserved for OS Kernel	No
\$gp	28	Global Pointer	Yes
\$sp	29	Stack Pointer	Yes
\$fp	30	Frame Pointer	Yes
\$ra	31	Return Address	Yes