

(Sinh viên không được sử dụng tài liệu, máy tính bỏ túi. Làm bài trực tiếp trên đề)

HỌ VÀ TÊN SV:	<u>ĐIỂM</u>	<u>CÁN BỘ COI THI</u>
MSSV:		
STT:		
PHÒNG THI:		

BẢNG TRẢ LỜI TRẮC NGHIỆM (8 điểm, 0.4 điểm/câu)

Câu 1	Câu 2	Câu 3	Câu 4	Câu 5	Câu 6	Câu 7	Câu 8	Câu 9	Câu 10
Câu 11	Câu 12	Câu 13	Câu 14	Câu 15	Câu 16	Câu 17	Câu 18	Câu 19	Câu 20

I. Phần I: TỰ LUẬN (2 điểm, Sinh viên làm trong khuôn khổ để trống bên dưới)

Câu 1. (0.5đ) (G1.1, G1.2)

- o Lệnh là gì? Mô tả chu trình thực thi một lệnh (0.25đ)
 - o Trình bày một số phương pháp nâng cao hiệu suất của máy tính (0.25đ)
-
.....
.....
.....
.....
.....
.....

Câu 2. (0.5đ) Cho 3 bộ xử lý P1, P2 và P3: cùng chạy một tập lệnh với các tần số xung clock và CPI được cho như bảng bên dưới. Nếu các bộ xử lý chạy 1 chương trình nào đó hết 10 giây, tìm tổng số chu kì và tổng số lượng lệnh tương ứng của mỗi bộ xử lý. (G1.1, G1.2)

Bộ xử lý	Clock Rate	CPI
P3	2.5 Ghz	1.5
P2	2 Ghz	1.0
P1	4 Ghz	2.5

.....
.....
.....
.....
.....
.....

Câu 3. (1đ) Cho mảng A có 10 phần tử, địa chỉ nền (base address) của mảng A được lưu trong thanh ghi \$t0. (G1.1)

- Viết chương trình MIPS tìm giá trị nhỏ nhất của mảng A, và lưu kết quả vào thanh ghi \$s0 (0.5đ)
- Viết chương trình MIPS sắp xếp lại mảng A theo thứ tự tăng dần. (0.5đ)

--	--

II. PHẦN TRẮC NGHIỆM (8 điểm, 1 câu 0.4 điểm, SV chọn 1 đáp án đúng và điền vào BẢNG TRẢ LỜI TRẮC NGHIỆM)

Câu 1 Với datapath đã học, khi thực hiện lệnh “lw” thì ALU thực hiện chức năng gì? (G1.2)

A. Sub	B. Add	C. Compare	D. AND	E. Đáp án khác
--------	--------	------------	--------	----------------

Câu 2 Thanh phần nào sau đây là mạch tò hợp? (G1.2)

A. ALU	B. D-MEM	C. I-MEM	D. Registers	E. Tất cả các thành phần
--------	----------	----------	--------------	--------------------------

Câu 3 Khi thực hiện lệnh add \$t9, \$t1, \$t2, trường rs trong phần định dạng lệnh bằng bao nhiêu ở giá trị thập phân? (G1.1)

A. 8	B. 9	C. 24	D. 23	E. Đáp án khác
------	------	-------	-------	----------------

Câu 4 Cho giá trị thanh ghi \$s3=0x0000000C, hỏi sau khi thực hiện câu lệnh "sll \$s3, \$s3, 3" thì giá trị thanh ghi \$s3 bằng bao nhiêu? (G1.1)

A. 0x0000000C	B. 0x00000060	C. 0x0000002C	D. 0x00000024	E. Đáp án khác
---------------	---------------	---------------	---------------	----------------

Câu 5 Máy tính A hoạt động ở tần số 4Ghz, máy tính B hoạt động ở tần số 5Ghz. hỏi máy tính nào có hiệu suất cao hơn, biết cả hai máy đều thực thi một chương trình có số lượng lệnh giống nhau? (G1.2)

A. Máy A	B. Máy B	C. Thiếu thông tin	D. Hai máy bằng nhau	E. Đáp án khác
----------	----------	--------------------	----------------------	----------------

Câu 6 Sau khi thực hiện lệnh slti \$t0, \$zero, -5 giá trị thập phân thanh ghi \$t0 bằng bao nhiêu? (G1.1)

A. 5	B. -5	C. 0	D. Không xác định được	E. Đáp án khác
------	-------	------	------------------------	----------------

Câu 7 Khi bộ xử lý thực hiện lệnh “sw \$t2, 4(\$t1)”, giá trị các tín hiệu điều khiển RegDst, RegWrite, MemWrite lần lượt là: (G1.1)

A. 1,0,1	B. 1,0,0	C. 0,0,1	D. 1,0,0	E. Đáp án khác
----------	----------	----------	----------	----------------

Câu 8 Bộ nhớ thanh ghi trong MIPS được đánh địa chỉ theo? (G1.1)

A. Bit	B. Byte	C. Word	D. Halfword
--------	---------	---------	-------------

Câu 9 Tập lệnh MIPS có mấy loại toán hạng? (G1.1)

A. 2	B. 3	C. 4	D. 5
------	------	------	------

Câu 10 Thanh ghi \$t5 có địa chỉ dạng nhị phân là bao nhiêu trong tập thanh ghi của MIPS? (G1.1)

A. 10111	B. 10110	C. 10101	D. Đáp án khác
----------	----------	----------	----------------

Câu 11 Mã máy của lệnh add \$s1, \$t5, \$t1 là gì? (G1.1)

A. 0x023D8210	B. 0x01A98820
C. 0x012D8024	D. 0x012D8820

Câu 12 Cho câu lệnh addi \$s3, \$s5, 15. Biểu diễn số nhị phân của trường shamt (G1.1)

A. 10011	B. 10101
C. 01111	D. 00000

Câu 13 Câu lệnh slti \$s3, \$s5, 0 thuộc định dạng nào trong lệnh hợp ngữ MIPS (G1.1)

A. Định dạng lệnh R	B. Định dạng lệnh I
C. Định dạng lệnh J	D. Cả ba đều sai

Câu 14 Kiến trúc tập lệnh MIPS được thiết kế theo loại nào sau đây: (G1.2)

A. Ngăn xếp (stack)	B. Bộ tích lũy (accumulator)
C. Thanh ghi – Bộ nhớ (register-memory)	D. Register-Register/load-store)

Câu 15 Cho hai bộ vi xử lý X và Y có tần số xung clock lần lượt là 800 MHz và 1000 MHz. Giả sử X thực thi một lệnh trung bình mất 5 chu kỳ, Y thực thi một lệnh trung bình mất 3 chu kỳ. Vậy để thực thi cùng một chương trình, bộ vi xử lý nào thực thi nhanh hơn? (G1.2)

A. Y	B. X	C. X bằng Y	D. Thiếu thông tin
------	------	-------------	--------------------

Câu 16 Cho máy tính X có CPI = 5. Máy tính X thực thi một chương trình có 1 triệu lệnh mất 5 ns. Hỏi tần số hoạt động của máy tính X là bao nhiêu? (G1.2)

A. 1 Mhz	B. 1Ghz	C. 2 Mhz	D. 2 Ghz	E. Đáp án khác
----------	---------	----------	----------	----------------

Câu 17 Giá trị thanh ghi PC sẽ tăng bao nhiêu sau mỗi lần đọc lệnh? (G1.1)

A. Tùy thuộc vào từng lệnh	B. Thiếu thông tin, chưa xác định được
C. 4	D. Không thay đổi

Câu 18 Số thập lục phân 0x014B4822 là mã lệnh của lệnh nào sau đây?

A. add \$9, \$10, \$11	B. sub \$9, \$10, \$10
C. addi \$t1, \$t2, \$t2	D. addu \$t1, \$t2, \$t2
E. Đáp án khác	

Câu 19 Cho một bộ xử lý MIPS 32 bits (có datapath và control như hình đã học).

Biết PC = 0x00400000; \$t1 = 0x10010030; \$t3 = 0x00000015; Word nhớ tại địa chỉ 0x10010030 có nội dung/giá trị bằng 0x00000015

Nếu đoạn chương trình sau được thực thi:

```

addi $s0, $t1, 8
lw $t2, 8($s0)
lw $t2, 8($s0)
beq $t3, $t2, ABC
add $t2, $t3, $t4
ABC: sub $t3, $t4, $t5

```

Giá trị thanh ghi PC bằng bao nhiêu khi bộ xử lý bắt đầu nạp lệnh (beq \$t3, \$t2, ABC)? (G1.1)

A. 0x00400008	B. 0x00400004	C. 0x00400018	D. 0x0040000C	E. Đáp án khác
---------------	---------------	---------------	---------------	----------------

Câu 20 "PSEUDO INSTRUCTION" là gì? (G1.1)

A. Lệnh giả	B. Nhóm lệnh số học
C. Nhóm lệnh bộ nhớ	D. Nhóm lệnh nhảy
E. Đáp án khác	

HẾT

CĐRMH	Mô tả CĐRMH (mục tiêu cụ thể)
G1.1 (2.1)	Trình bày được các kiến thức cơ bản về kiến trúc máy tính và lập trình hợp ngữ.
G1.2 (2.1)	Trình bày, phân tích được các thành phần và nguyên lý hoạt động bên trong một máy tính, cơ chế thực thi lệnh của máy tính.



1. Pull along perforation to separate card 2. Fold bottom side (columns 3 and 4) together

MIPS Reference Data

CORE INSTRUCTION SET

NAME, MNEMONIC	MAT	FOR- OPERATION (in Verilog)	/ FUNCT (Hex)	OPCODE
Add	add	R R[rd] = R[rs] + R[rt]	(1) 0 / 20 _{hex}	
Add Immediate	addi	I R[rt] = R[rs] + SignExtImm	(1,2) 8 _{hex}	
Add Imm. Unsigned	addiu	I R[rt] = R[rs] + 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] & ZeroExtImm	(3) c _{hex}	
Branch On Equal	beq	I if(R[rs]==R[rt]) PC=PC+4+BranchAddr	(4) 4 _{hex}	
Branch On Not Equal	bne	I if(R[rs]!=R[rt]) PC=PC+4+BranchAddr	(4) 5 _{hex}	
Jump	j	J PC=JumpAddr	(5) 2 _{hex}	
Jump And Link	jal	J R[31]=PC+8;PC=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]] +SignExtImm(7:0))	(2) 24 _{hex}	
Load Halfword Unsigned	lhu	I R[rt]=(16'b0,M[R[rs]] +SignExtImm(15:0))	(2) 25 _{hex}	
Load Linked	ll	I R[rt] = M[R[rs]+SignExtImm]	(2,7) 30 _{hex}	
Load Upper Imm.	lui	I R[rt] = {imm, 16'b0}	f _{hex}	
Load Word	lw	I R[rt] = M[R[rs]+SignExtImm]	(2) 23 _{hex}	
Nor	nor	R R[rd] = ~ (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] 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] < SignExtImm) ? 1 : 0 (2)	a _{hex}	
Set Less Than Imm. Unsigned	sltiu	I R[rt] = (R[rs] < SignExtImm) ? 1 : 0 (2,6)	b _{hex}	
Set Less Than Unsigned	sltu	R R[rd] = (R[rs] < R[rt]) ? 1 : 0 (6)	0 / 2b _{hex}	
Shift Left Logical	sll	R R[rd] = R[rt] << sham	0 / 00 _{hex}	
Shift Right Logical	srl	R R[rd] = R[rt] >> sham	0 / 02 _{hex}	
Store Byte	sb	I M[R[rs]+SignExtImm(7:0)] = R[rt](7:0) (2)	28 _{hex}	
Store Conditional	sc	I M[R[rs]+SignExtImm] = R[rt]; R[rt] = (atomic) ? 1 : 0 (2,7)	38 _{hex}	
Store Halfword	sh	I M[R[rs]+SignExtImm(15:0)] = R[rt](15:0) (2)	29 _{hex}	
Store Word	sw	I M[R[rs]+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] (1)	0 / 23 _{hex}	

- (1) May cause overflow exception
- (2) SignExtImm = { 16 (immediate[15]), immediate }
- (3) ZeroExtImm = { 16 (1b'0), immediate }
- (4) BranchAddr = { 14(immediate[15]), immediate, 2'b0 }
- (5) JumpAddr = { PC+4[31:28], 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	0			
I	opcode	rs	rt	immediate						
	31	26 25	21 20	16 15	0					
J	opcode	address								
	31	26 25	0							

Copyright 2009 by Elsevier, Inc., All rights reserved. From Patterson and Hennessy, Computer Organization and Design, 4th ed.



①

ARITHMETIC CORE INSTRUCTION SET

NAME, MNEMONIC	MAT	FOR- OPERATION	OPERATION (Hex)	OPCODE / FMT / FT / FUNCT
Branch On FP True	bclt	FI if(FPcond)PC=PC+4+BranchAddr	(4) 11/8/1--	
Branch On FP False	bclf	FI if(!FPcond)PC=PC+4+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	add.d	FR {F[fd],F[fd+1]} = {F[fs],F[fs+1]} + {F[ft],F[ft+1]}	11/11/-/0	
Double	div.s	FR F[fd] = F[fs]/F[ft]	11/10/-/3	
FP Compare Single	c.x.s*	FR FPcond = (F[fs] op F[ft]) ? 1 : 0	11/10/-/y	
FP Compare	c.x.d*	FR FPcond = ({F[fs],F[fs+1]} op {F[ft],F[ft+1]}) ? 1 : 0	11/11/-/y	
Double	*	(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	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	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	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]+SignExtImm]	(2) 31/-/-/	
Load FP	ldc1	I F[rt]=M[R[rs]+SignExtImm]; F[r+1]=M[R[rs]+SignExtImm+4]	(2) 35/-/-/	
Move From HI	mfc1	R R[rd] = Hi	0/-/-/10	
Move From LO	mfl1	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] >> sham	0/-/-/3	
Store FP Single	swc1	I M[R[rs]+SignExtImm] = F[rt]	(2) 39/-/-/	
Store FP	sdcl	I M[R[rs]+SignExtImm] = F[rt]; M[R[rs]+SignExtImm+4] = F[r+1]	(2) 3d/-/-/	

FLOATING-POINT INSTRUCTION FORMATS

FR	opcode	fmnt	ft	fs	fd	funct	
	31	26 25	21 20	16 15	11 10	6 5	0
FI	opcode	fmnt	ft	immediate			
	31	26 25	21 20	16 15	0		

PSEUDOINSTRUCTION SET

NAME	MNEMONIC	OPERATION
Branch Less Than	blt	if(R[rs]<R[rt]) PC = Label
Branch Greater Than	bgt	if(R[rs]>R[rt]) PC = Label
Branch Less Than or Equal	ble	if(R[rs]<=R[rt]) PC = Label
Branch Greater Than or Equal	bge	if(R[rs]>=R[rt]) PC = Label
Load Immediate	li	R[rd] = 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	No

