

TRƯỜNG ĐH. CÔNG NGHỆ THÔNG TIN KHOA KỸ THUẬT MÁY TÍNH	ĐỀ THI GIỮA HỌC KỲ 2 (2018-2019) MÔN: KIẾN TRÚC MÁY TÍNH Thời gian: 65 phút
HỌ VÀ TÊN SV:MSSV:STT:	
DIỂM:	CHỦ KÝ CÁN BỘ COI THI: Cán bộ coi thi 1:.....Cán bộ coi thi 2:.....

Lưu ý trắc nghiệm: - Câu hỏi có thể có nhiều đáp án đúng, hoặc SV có thể điền đáp án đối với các câu có đáp án E để trống

Bảng trả lời trắc nghiệm: chọn bỏ chọn chọn lại

Câu													
A													
Đáp án													
B													
C													
D													
E													

I. Trắc nghiệm (7 điểm)

1. Cho bảng dưới:

Processor Rate	Clock	No. Instructions	Time
P1	2 GHz	$20 \cdot 10^9$	7s

Tìm IPC (số lệnh được thực hiện trong một chu kỳ – instruction per cycle) cho bộ xử lý trên?

A. 10	B. 14.2	C. 1.42	1.0	E. ...
2.	Giả sử trong một chương trình A gồm 1000 lệnh thì có đến 200 lệnh tính toán số học. Người thiết kế giảm đi 50% số lượng chu kỳ cần thiết cho lệnh tính toán số học này. Chương trình này đã được tăng tốc như thế nào?			
A. 11.11%	B. 12.12%	C. 13.13%	D. 14.14%	
3.	Một bức ảnh không nén RGB 8bit độ phân giải 1920x1080 có kích thước là bao nhiêu Megabytes?			
A. 5.93	B. 47.46	C. 59.3	D. 4.76	
4.	Chu kỳ xung clock là 2×10^{-6} thì tần số của xung clock là:			
A. 500 Mhz	B. 500 KHz	C. 500.000 KHz	D. 0.05 Ghz	E. ...
5.	Địa chỉ của thanh ghi \$S5 có giá trị:			
A. 5	B. 21	C. 22	D. 16	E. ...
6.	Chọn đáp án đúng cho giá trị MIPS (Million instructions per second):			

A. Là đại lượng đo tốc độ thực thi của chương trình dựa trên số triệu lệnh trên giây

- B. Được tính bằng công thức: Clock rate / CPI
 C. Tổng số lượng lệnh chia cho thời gian thực thi
 D. Được tính bằng công thức: $(Clock\ rate / CPI) * 10^6$

7. Phương pháp nào sau đây có thể tăng hiệu suất của một máy tính?

A. Tăng số chu kỳ xung clock của chương trình	A. Giảm tần số xung clock CPU
C. Tăng chu kỳ xung clock CPU	C. Giảm CPI của máy tính

8. Bảng dưới đây cho thấy cách hiện thực trên máy tính M1 với cùng một tập lệnh, trong đó tập lệnh này gồm 3 lớp lệnh (instruction class) A, B và C. Số lượng lệnh được thực thi trong một đoạn chương trình X được thể hiện ở cột cuối cùng.

Lớp	CPI cho M1	Tần số của lệnh
A	1	40%
B	3	30%
C	4	30%

Thông số CPI trung bình của đoạn chương trình X trên máy tính M2?

A. 2.1	B. 2.3	C. 2.5	D. 3.0	E.....
--------	--------	--------	--------	--------

9. Trong các chương trình sau, chương trình nào thuộc về phần mềm hệ thống

A. Hệ điều hành	B. Trình Biên dịch	C. Web Browser	D. Window OS
-----------------	--------------------	----------------	--------------

10. Sắp xếp đúng theo thứ tự giảm dần về tốc độ bộ nhớ:

A. RAM > Đĩa quang > bộ nhớ Flash	B. Cache > RAM > bộ nhớ Flash
C. RAM > Register > Đĩa Quang.	D. Register > Bộ nhớ Flash > RAM

11. Cho 1 file có kích thước là 2MB, sử dụng công nghệ mạng Ethernet có tốc độ 100Mbit/s để truyền dữ liệu. Hỏi sau bao lâu thì truyền hết file này?

A. 0.16 s	B. 16 ms	C. 20 ms	D. 0.2 s	E. ...
-----------	----------	----------	----------	--------

12. Trong Vi xử lý MIPS 32, bộ nhớ Register có dung lượng bao nhiêu?

A. 32 Word	B. 32 byte	C. 128 Byte	D. 128 Kbyte
------------	------------	-------------	--------------

13. Trong các câu lệnh nhị phân biểu diễn dưới dạng thập lục phân bên dưới. Câu lệnh nào dùng để biểu diễn lệnh and \$s4, \$s6, \$s7:

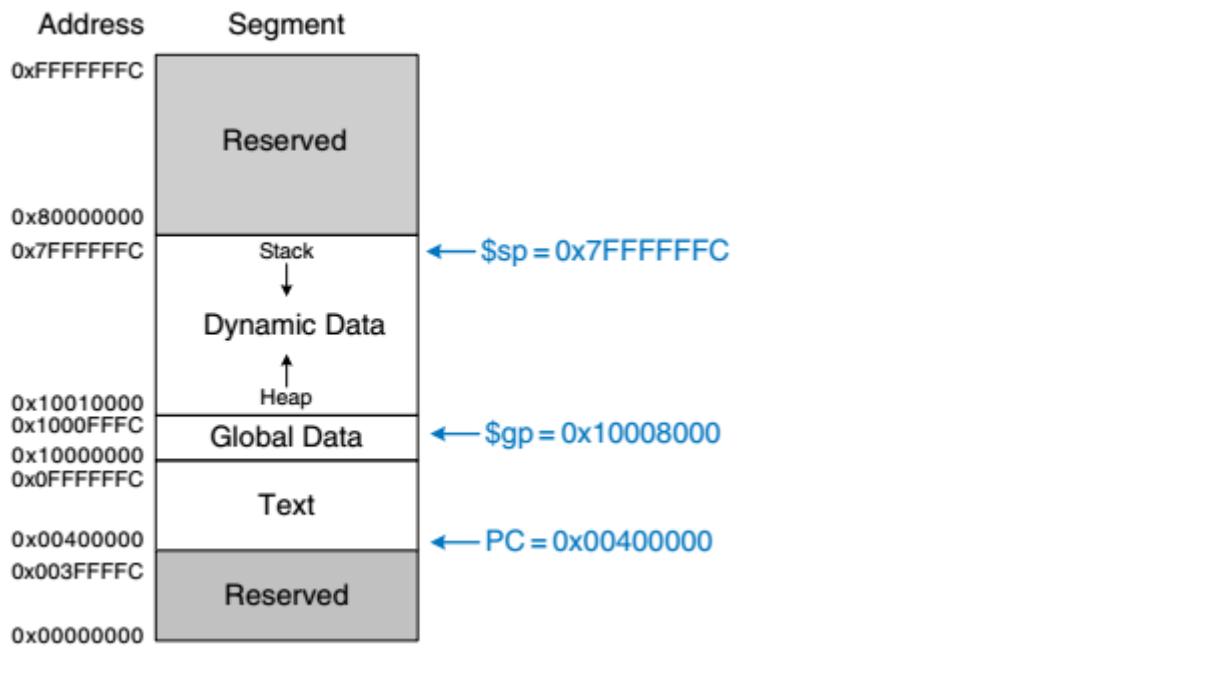
A. 0x02d7a024	B. 0x02cfa020.	C. 0x02cfa025.	D. 0x02cfa022
---------------	----------------	----------------	---------------

14. Lệnh tương ứng với mã máy 0x8e110020 là :

A. lw \$s1, 32(\$s0).	B. sw \$s1, 32(\$s0).	C. lw \$s2, 32(\$s0).	D. sw \$s2, 32(\$s0).
-----------------------	-----------------------	-----------------------	-----------------------

II. Tự luận (3 điểm)

Câu 1. (1 điểm) Cho tổ chức sơ đồ tổ chức bộ nhớ của MIPS như hình vẽ. Xác định dung lượng bộ nhớ (tính theo byte) của các vùng nhớ Dynamic data, static data, text, và reserved.



Câu 2. (1 điểm) Mảng A có số phần tử lưu trong thanh ghi \$s1, biết Base address của mảng A được lưu trong thanh ghi \$s2. Viết đoạn mã hợp ngữ tìm giá trị lớn nhất trong mảng A và lưu vào thanh ghi \$s3.

Câu 3. (1 điểm) Viết một đoạn mã hợp ngữ để hoán đổi giá trị giữa hai thanh ghi \$t0 và \$s1.

Hết-----

Đây là phần đánh giá chuẩn đầu ra của đề thi theo đề cương chi tiết môn học (CDRMH) (thí sinh không cần quan tâm mục này trong quá trình làm bài)

Câu hỏi	CDRMH	Mô tả
Phản trắc nghiệm		
Câu 1, Câu 6, Câu 7, Câu 8	G2	Đánh giá, so sánh hiệu suất giữa các máy tính
Câu 9, Câu 10, Câu 11, Câu 4,Câu 3,Câu 7	G2	Hiểu các kiến thức cơ bản về kiến trúc máy tính
Câu 2,Câu 13,Câu 14,Câu 5	G1	Hiểu, và trình bày được định dạng của lệnh, lập trình ngôn ngữ hợp ngữ
Phản tự luận		
Câu 1	G1	Hiểu được kiến trúc cơ bản của máy tính MIPS, cụ thể là tổ chức bộ nhớ trong MIPS 32
Câu 2, Câu 3	G1	Hiểu, và lập trình được trên ngôn ngữ hợp ngữ
Duyệt đề Khoa/Bộ Môn		Giáo viên ra đề

MIPS Reference Data Card ("Green Card") 1. Pull along perforation to separate card 2. Fold bottom side (columns 3 and 4) together

MIPS Reference Data



CORE INSTRUCTION SET

			FOR- NAME, MNEMONIC MAT	OPERATION (in Verilog) (Hex)	OPCODE
Add	add	R	R[n] = R[r1] + R[r2]	(1) 0/20hex	
Add Immediate	addi	I	R[r1] = R[r2] + SignExtImm	(1,2) 8hex	
Add Imm. Unsigned	addiu	I	R[r1] = R[r2] + SignExtImm	(2) 9hex	
Add Unsigned	addu	R	R[n] = R[r1] + R[r2]	0/21hex	
And	and	R	R[n] = R[r1] & R[r2]	0/24hex	
And Immediate	andi	I	R[r1] = R[r2] & ZeroExtImm	(3) 4hex	
Branch On Equal	beq	I	R[r1] == R[r2] & BranchAddr	(4) 4hex	
Branch On Not Equal	beqz	I	R[r1] != R[r2] & BranchAddr	(4) 5hex	
Branch On Equal or	beqlz	I	R[r1] == R[r2] & BranchAddr	(4) 5hex	
Jump	j	J	PC = JumpAddr	(5) 2hex	
Jump And Link	jal	J	R[31] = PC + 8, PC = JumpAddr	(5) 5hex	
Jump Register	jr	R	PC = R[r1]	0/08hex	
Load Byte Unsigned	lbu	I	R[r1] = {24'b0, M[R[r2]] + SignExtImm}(7:0)	(2) 24hex	
Load Halfword Unsigned	lh	I	R[r1] = {16'b0, M[R[r2]] + SignExtImm}(15:0)	(2) 25hex	
Load Linked	ll	I	R[r1] = M[R[r2]] + SignExtImm	(2,7) 30hex	
Load Upper Imm.	lui	I	R[r1] = {imm, 16'b0}	4hex	
Load Word	lw	I	R[r1] = M[R[r2]] + SignExtImm	(2) 23hex	
Not	nor	R	R[r1] = ~ (R[r1] R[r2])	0/27hex	
Or	or	R	R[r1] = R[r1] R[r2]	0/25hex	
Or Immediate	ori	I	R[r1] = R[r2] ZeroExtImm	(3) 4hex	
Set Less Than	slt	R	R[r1] = R[r2] < R[r3] ? 1 : 0	0/29hex	
Set Less Than Imm.	slti	I	R[r1] = R[r2] < SignExtImm? 1 : 0 (2)	2hex	
Set Less Than Imm. Unsigned	sltiu	I	R[r1] = R[r2] < SignExtImm? 1 : 0 (2,6)	2hex	
Set Less Than Unsigned	sltu	R	R[r1] = R[r2] < R[r3] ? 1 : 0 (6) 0/2bhex		
Shift Left Logical	sll	R	R[r1] = R[r2] << shamt	0/00hex	
Shift Right Logical	srl	R	R[r1] = R[r2] >> shamt	0/02hex	
Store Byte	sb	I	M[R[r1]] + SignExtImm(7:0) = R[r2]	28hex	
Store Conditional	sc	I	M[R[r1]] + SignExtImm = R[r2]; R[r1] = (atomic) ? 1 : 0 (2,7)	38hex	
Store Halfword	sh	I	M[R[r1]] + SignExtImm(15:0) = R[r2]	29hex	
Store Word	sw	I	M[R[r1]] + SignExtImm = R[r2]	28hex	
Subtract	sub	R	R[r1] = R[r2] - R[r3]	0/22hex	
Subtract Unsigned	subu	R	R[r1] = R[r2] - R[r3]	0/23hex	

BASIC INSTRUCTION FORMATS

R	opcode	rs	rt	rd	shamt	funct
I	opcode	rs	rt			immediate
J	opcode	rs	rt			address

- (1) May cause overflow exception
- (2) SignExtImm = {16{immediate[15]}, immediate}
- (3) ZeroExtImm = {16{lb'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[r1] = 1 if pair atomic, 0 if not atomic

ARITHMETIC CORE INSTRUCTION SET

① OPCODE / FMT / FT	② OPCODE / FMT / FUNCT
NAME, MNEMONIC	FOR- NAME, MNEMONIC MAT
Branch On FP True	Branch On FP True
Branch On FP False	Branch On FP False
Divide	div
Divide Unsigned	divu
FP Add Single	add.s
FP Add Double	add.d
FP Compare Single	c.s*
FP Compare Double	c.d*
FP Divide	div.s
FP Divide Double	div.d
FP Multiply Single	mul.s
FP Multiply Double	mul.d
FP Subtract Single	sub.s
FP Subtract Double	sub.d
Load FP Single	lwc1
Load FP Double	lwd1
Double	ldc1
Move From HI	mfhi
Move From Lo	mflo
Move From Control	mfc0
Multiply	mult
Multiply Unsigned	multu
Shift Right Arith.	sr
Store FP Single	swc1
Store FP Double	swd1

FLOATING-POINT INSTRUCTION FORMATS

FR	opcode	fmt	f1	f2	f3	fd	funct
FI	opcode	fmt	f1	f2	f3	fd	immediate

PSEUDOINSTRUCTION SET

NAME	MNEMONIC	OPERATION
Branch Less Than	blt	iif[R[r1]<R[r2]] PC = Label
Branch Greater Than	bgt	iif[R[r1]>R[r2]] PC = Label
Branch Less Than Or Equal	ble	iif[R[r1]<=R[r2]] PC = Label
Branch Greater Than Or Equal	bge	iif[R[r1]>=R[r2]] PC = Label
Load Immediate	li	R[r1] = immediate
Move	move	R[r1] = R[r2]

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

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