

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:			
ĐIỂM:	CHỮ 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	D. 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: $\text{Clock rate} / \text{CPI}$

C. Tổng số lượng lệnh chia cho thời gian thực thi

D. Được tính bằng công thức: $(\text{Clock rate} / \text{CPI}) * 10^6$

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:

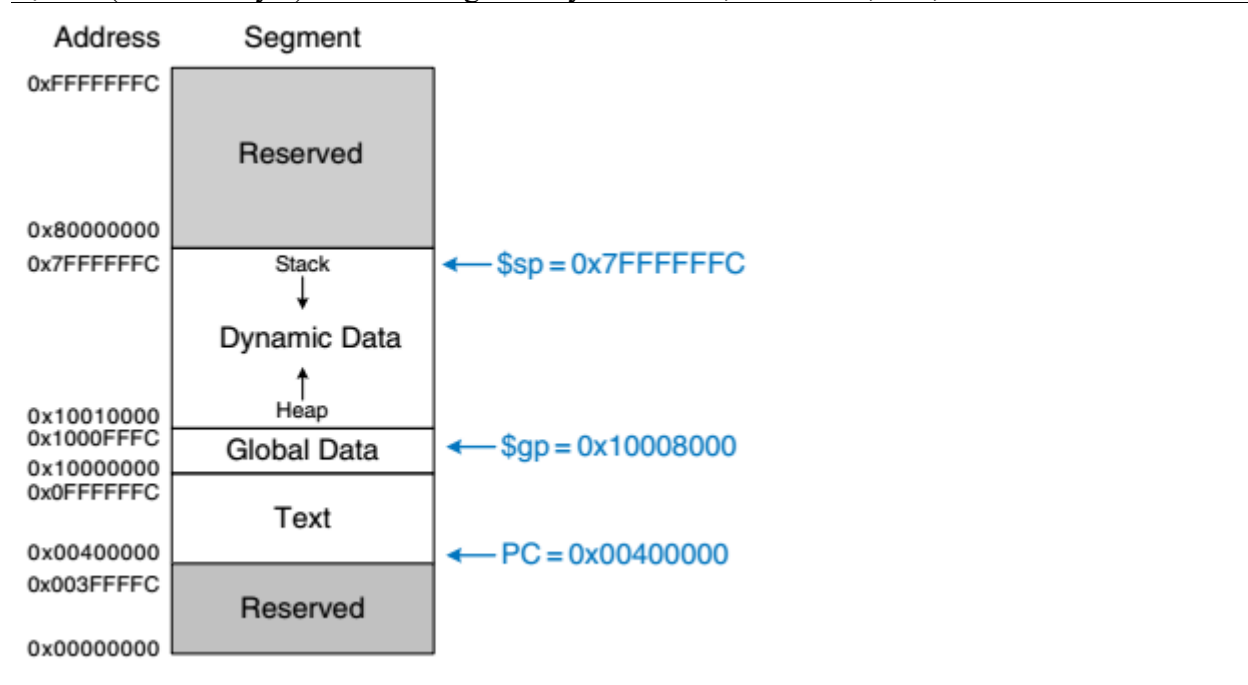
A. 0x02d7a024	B. 0x02cfa020.	C. 0x02cfa025.	D. 0x02cfa022
---------------	----------------	----------------	---------------

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
Phần 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
Phần 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

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) 0 _{hex}
Branch On Equal	beq I	$\text{if}(R[rs] == R[rt])$ $\text{PC} = \text{PC} + 4 + \text{BranchAddr}$	(4) 4 _{hex}
Branch On Not Equal	bne I	$\text{if}(R[rs] \neq R[rt])$ $\text{PC} = \text{PC} + 4 + \text{BranchAddr}$	(4) 5 _{hex}
Jump	j J	$\text{PC} = \text{JumpAddr}$	(5) 2 _{hex}
Jump And Link	jal J	$R[31] = \text{PC} + 8; \text{PC} = \text{JumpAddr}$	(5) 3 _{hex}
Jump Register	jr R	$\text{PC} = R[rs]$	0/08 _{hex}
Load Byte Unsigned	lb I	$R[rt] = \{24'b0, M[R[rs]](7:0)\}$ $+ \text{SignExtImm}(7:0)$	(2) 24 _{hex}
Load Halfword Unsigned	lhu I	$R[rt] = \{16'b0, M[R[rs]](15:0)\}$ $+ \text{SignExtImm}(15:0)$	(2) 25 _{hex}
Load Linked	ll I	$R[rt] = M[R[rs]] + \text{SignExtImm}$	(2,7) 30 _{hex}
Load Upper Imm.	lui I	$R[rt] = (\text{imm}, 16'b0)$	0/27 _{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) 0 _{hex}
Set Less Than	slt R	$R[rd] = (R[rs] < R[rt]) ? 1 : 0$	0/28 _{hex}
Set Less Than Imm.	slti I	$R[rt] = (R[rs] < \text{SignExtImm}) ? 1 : 0$	(2) 0 _{hex}
Set Less Than Imm. Unsigned	sltiu I	$R[rt] = (R[rs] < \text{SignExtImm}) ? 1 : 0$	(2,6) 0 _{hex}
Set Less Than Unsig.	sltu R	$R[rd] = (R[rs] < R[rt]) ? 1 : 0$	(6) 0/2b _{hex}
Shift Left Logical	sll R	$R[rd] = R[rs] \ll \text{shamt}$	0/00 _{hex}
Shift Right Logical	srl R	$R[rd] = R[rs] \gg \text{shamt}$	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) 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
I	opcode	rs	rt	immediate		
	31	26-25	21-20	16-15		
J	opcode	address				
	31	26-25				

ARITHMETIC CORE INSTRUCTION SET

②

NAME, MNEMONIC	FOR-MAT	OPERATION	OPCODE / FUNCT (Hex)
Branch On FP True	bclt FI	$\text{if}(\text{FPCond}) \text{PC} = \text{PC} + 4 + \text{BranchAddr}$	(4) 11/01 _{hex}
Branch On FP False	bclt FI	$\text{if}(\neg \text{FPCond}) \text{PC} = \text{PC} + 4 + \text{BranchAddr}$	(4) 11/00 _{hex}
Divide	div R	$\text{Lo} = R[rs]/R[rt]; \text{Hi} = R[rs]\%R[rt]$	0/-/-/1a
Divide Unsigned	divu R	$\text{Lo} = R[rs]/R[rt]; \text{Hi} = R[rs]\%R[rt]$	(6) 0/-/-/1b
FP Add Single	add.s FR	$F[fs] = F[fs] + F[ft]$	11/10/-/0
FP Add Double	add.d FR	$F[fs] = F[fs] + F[ft]$	11/11/-/0
FP Compare Single	c.s.s* FR	$\text{FPCond} = (F[fs] \text{ op } F[ft]) ? 1 : 0$	11/10/-/y
FP Compare Double	c.d.d* FR	$\text{FPCond} = ((F[fs], F[fs+1]) \text{ op } (F[ft], F[ft+1])) ? 1 : 0$	11/11/-/y
* (s is eq, lt, or le) (op is ==, <, or <=) (y is 32, 3c, or 3e)			
FP Divide Single	div.s FR	$F[fs] = F[fs] / F[ft]$	11/10/-/3
FP Divide Double	div.d FR	$F[fs] = F[fs] / F[ft]$	11/11/-/3
FP Multiply Single	mul.s FR	$F[fs] = F[fs] * F[ft]$	11/10/-/2
FP Multiply Double	mul.d FR	$F[fs] = F[fs] * F[ft]$	11/11/-/2
FP Subtract Single	sub.s FR	$F[fs] = F[fs] - F[ft]$	11/10/-/1
FP Subtract Double	sub.d FR	$F[fs] = F[fs] - F[ft]$	11/11/-/1
Load FP Single	lwc1 I	$F[rt] = M[R[rs] + \text{SignExtImm}]$	(2) 31/-/-/0
Load FP Double	lwc1 I	$F[rt] = M[R[rs] + \text{SignExtImm}]$	(2) 35/-/-/0
Move From Hi	mfc1 R	$R[rt] = \text{Hi}$	0/-/-/10
Move From Lo	mfc1 R	$R[rt] = \text{Lo}$	0/-/-/12
Move From Control	mfc0 R	$R[rt] = \text{CR}[rs]$	10/0/-/0
Multiply	mult R	$\{\text{Hi}, \text{Lo}\} = R[rs] * R[rt]$	0/-/-/18
Multiply Unsigned	multu R	$\{\text{Hi}, \text{Lo}\} = R[rs] * R[rt]$	(6) 0/-/-/19
Shift Right Arith.	sra R	$R[rd] = R[rs] \gg \text{shamt}$	0/-/-/23
Store FP Single	swc1 I	$M[R[rs] + \text{SignExtImm}] = F[rt]$	(2) 39/-/-/0
Store FP Double	swc1 I	$M[R[rs] + \text{SignExtImm} + 4] = F[rt+1]$	(2) 3d/-/-/0

FLOATING-POINT INSTRUCTION FORMATS

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

PSEUDOINSTRUCTION SET

NAME	MNEMONIC	OPERATION
Branch Less Than	slt	$\text{if}(R[rs] < R[rt]) \text{PC} = \text{Label}$
Branch Greater Than	sgt	$\text{if}(R[rs] > R[rt]) \text{PC} = \text{Label}$
Branch Less Than or Equal	sle	$\text{if}(R[rs] \leq R[rt]) \text{PC} = \text{Label}$
Branch Greater Than or Equal	sge	$\text{if}(R[rs] \geq R[rt]) \text{PC} = \text{Label}$
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