

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN	ĐỀ THI GK1 (2019-2020)
KHOA KỸ THUẬT MÁY TÍNH	KIẾN TRÚC MÁY TÍNH
Đề 10	Thời gian: 70 phút

(Sinh viên không được sử dụng tài liệu. Làm bài trực tiếp trên đề)

<u>STT</u> .....	Họ và tên: ..... MSSV: ..... Phòng thi: .....	<u>ĐIỂM</u> .....
---------------------	---	----------------------

### TRẮC NGHIỆM (9.2 điểm, 0.4 điểm/câu), SV chọn 1 đáp án đúng

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	Câu 21	Câu 22	Câu 23	

1. Một bức ảnh có độ phân giải 2K có kích thước 2560x1440 pixels. Mỗi pixel chứa thông tin 3 màu cơ bản đỏ, xanh lá cây, xanh lam. Mỗi màu cơ bản được thể hiện bởi 8 bits. Để lưu trữ bức ảnh đó trên bộ nhớ thì dung lượng tối thiểu của bộ nhớ là bao nhiêu Mbytes? (G3)

- |       |       |       |       |
|-------|-------|-------|-------|
| A. 10 | B. 11 | C. 12 | D. 20 |
|-------|-------|-------|-------|

2. Đoạn lệnh sau tương ứng với đoạn lệnh C nào, với i và k tương ứng với thanh ghi s3 và s5, địa chỉ nền của mảng save là thanh ghi s6? (G1)

Loop:  $sll \$t1,\$s3,2$

$add \$t1,\$t1,\$s6$

$lw \$t0,0(\$t1)$

$bne \$t0,\$s5, Exit$

$addi \$s3,\$s3,1$

$j Loop$

Exit:

A.	while (save[i*4] == k) i += 1;
B.	while (save[i] == k) i = i + 1;
C.	while (save[i] # k) i += 1;
D.	while (save[i]*4 == k) i += 1;

3. Một bạn sinh viên sử dụng gói cước super 50 (50Mbps) của nhà mạng FPT và bạn sinh viên đó đang cần download 1 tập phim “Tiếng sét trong mưa tập 33.mp4” có kích thước là 600MB để cho mẹ bạn xem. Hỏi bạn sinh viên đó cần ít nhất bao nhiêu thời gian (s) để download xong bộ phim, giả sử rằng toàn bộ băng thông mạng chỉ sử dụng cho việc download phim đó.? (G3)

A. 12s	B. 96s	C. 83s	D. 15s
--------	--------	--------	--------

4. Một Terabyte bằng bao nhiêu byte? (G1)

A. $2^{10}$ byte	B. $2^{20}$ byte	C. $2^{30}$ byte	D. $2^{40}$ byte
------------------	------------------	------------------	------------------

5. Trong các loại bộ nhớ sau, bộ nhớ nào có tốc độ nhanh nhất (G1)

A. RAM	B. Cache	C. Đĩa Quang	D. Bộ nhớ Flash
--------	----------	--------------	-----------------

6. Máy tính Acer E1 có tần số xung clock là 2.0 GHz. Để thực thi một chương trình gồm 2019 lệnh thì máy tính thực hiện trong bao lâu? Biết trung bình mỗi lệnh kéo dài 5 chu kỳ. (G2)

A. 2024s	B. 2014ns	C. 5047.5s	D. 5047.5ns
----------	-----------	------------	-------------

7. Điện thoại thông minh (smart phone) thuộc nhóm máy tính nào? (G1)

A.	Máy tính cá nhân
B.	Máy tính nhúng
C.	Máy tính chủ
D.	Siêu máy tính

8. Trong các câu lệnh assembly MIPS bên dưới. Câu lệnh nào chuyển đúng cho câu lệnh cấp cao  $a = b - 5$ , biết biến a, b lưu trữ trong thanh ghi \$s3, \$s4 (G1)

A.	addi \$s3, \$s4, -5
B.	add \$s3, \$s4, -5
C.	sub \$s3, \$s4, 5
D.	subi \$s3, \$s4, 5

9. Bảng dưới đây mô tả số lệnh và thời gian thực thi tương ứng trên máy tính Acer E1 khi thực hiện một chương trình, trong đó tập lệnh này gồm 4 lớp lệnh (instruction class) A, B, C và D.

Lớp	CPI cho Acer E1	Số lệnh
A	1	650
B	5	120
C	5	500
D	2	50

Tính thời gian thực thi của chương trình biết máy tính có tần số 2Ghz? (G2)

A. 2340ns	B. 660ns	C. 1925ns	D. 7700ns
-----------	----------	-----------	-----------

10. Trong các câu lệnh assembly MIPS bên dưới. Câu lệnh nào nhảy đến nhãn KTMT, biết thanh ghi \$s1 = \$s0 (G1)

A.	slt \$s1, \$s0, KTMT
B.	beq \$s1, \$s0, KTMT
C.	bne \$s1, \$s0, KTMT
D.	blt \$s1, \$s0, KTMT

11. Trong các câu lệnh nhị phân biểu diễn dưới dạng thập lục phân bên dưới. Câu lệnh nào dùng để biểu diễn lệnh addi \$t3, \$t5, -146 (G1)

A.	0x21ABFF6E
B.	0x31ABFFD2
C.	0x35ABFF6E
D.	0x29ABFFD2

12. Trong các câu lệnh assembly MIPS bên dưới. Câu lệnh nào dùng để biểu diễn lệnh 0x29ABFF79 (G1)

A.	slti \$t3, \$t5, -135
B.	addi \$t3, \$t5, -135
C.	slti \$t3, \$t5, 135
D.	addi \$t3, \$t5, 135

13. Trong theo slide về kiến trúc tập lệnh đã học kiến trúc MIPS có bao nhiêu loại toán hạng? (G1)

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

14. Cho đoạn mã chương trình assembly như bên dưới:

```

slti $t0, $s1, 5
beq $t0, $zero, ELSE
sll $t1, $s1, 2
add $s2, $s2, $t1
j End

```

ELSE: add \$s2, \$s1, \$zero

End

Biết thanh ghi \$s1 = 1, thanh ghi \$s2 = 0. Cho biết thanh ghi \$s2 bằng bao nhiêu sau khi thực hiện đoạn lệnh chương trình trên (G1)

A. 3	B. 4	C. 5	D. 6
------	------	------	------

15. Chức năng của thanh ghi \$ra (G1)

A.	Thanh ghi lưu tham số truyền tham số cho hàm/thủ tục
B.	Thanh ghi dùng để lưu giá trị trả về của hàm
C.	Thanh ghi chứa địa chỉ của lệnh ngay sau lệnh gọi thủ tục
D.	Thanh ghi dùng để lưu địa chỉ của stack

16. Thanh ghi nào sau đây mà giá trị của nó không thể thay đổi (G1)

A.	Stack Pointer
B.	Zero
C.	Frame Pointer
D.	Return Address

17. Cho biết giá trị của thanh ghi \$t3 sau khi thực hiện lệnh sau

```

add $t2, $t1, $t0
addi $t3, $t2, 80000

```

Giả sử giá trị ban đầu chứa trong thanh ghi \$t0 = 0, \$t1 = 1, \$t2 = 2 (G1)

A.	\$t3 = 80000
B.	\$t3 = 80001

C.	$\$t3 = 80003$
D.	Cả 3 đáp án trên đều sai

18. Cho  $\$t1 = 0xffffffff$

Giá trị của thanh ghi  $\$t2$  và  $\$t3$  là bao nhiêu sau khi thực thi lệnh sau: (G1)

*sltiu \$t2, \$t1, 0x73*

*slti \$t3, \$t1, 0x73*

A.	$\$t2 = 1; \$t3 = 1$
B.	$\$t2 = 0; \$t3 = 1$
C.	$\$t2 = 1; \$t3 = 0$
D.	$\$t2 = 0; \$t3 = 0$

19. Trong kiến trúc máy tính MIPS, khi máy tính thực thi lệnh “jal 400” thì (G1)

A.	$\$ra = PC + 4$ và $PC = 400$
B.	$\$ra = PC$ và $PC = PC + 400$
C.	$\$ra = PC + 4$ và $PC = 1600$
D.	$\$ra = PC$ và $PC = PC + 1600$

20. Cần ít nhất bao nhiêu lệnh hợp ngữ để biểu diễn câu lệnh trong C sau : (G1)  
 $f = g - A[B[4]]$

A. 3	B. 4	C. 5	D. 6
------	------	------	------

21. Lệnh MIPS nào tương đương với mã lệnh C sau đây: (G1)

*if(\$s2 < \$s3)*

*\$s1 = 1;*

*else*

*\$s1 = 0*

A.	<i>beq \$s1,\$s2,\$s3</i>
B.	<i>slt \$s1,\$s2,\$s3</i>
C.	<i>sltu \$s1,\$s2,\$s3</i>
D.	<i>sltu \$s2,\$s1,\$s3</i>

22. Thực hiện phép nhân cho 2 số 4 bit sau  $0010_2 \times 0011_2$  trên phần cứng 3 thanh ghi. Cho biết giá trị của thanh ghi tích bằng bao nhiêu sau bước lặp số 2. (G3)

A. 0000 0110	B. 0000 0011	C. 0000 0001	D. 0000 0010
--------------	--------------	--------------	--------------

23. Kết quả thực hiện phép tính  $01000110_2 + 01011100_2$  dưới dạng thập phân là: (G3)

A. 154	B. 168	C. 162	D. 160
--------	--------	--------	--------

### Tự Luận (0.8 điểm) (G1, G4)

Chuyển đoạn lệnh C sau sang assembly của MIPS (tối đa 10 câu lệnh).

Biết i và j tương ứng với các thanh ghi  $\$s0$  và  $\$s1$ . Mảng A là mảng mà các phần tử là số nguyên, mỗi phần tử chiếm 1 từ nhớ (4 bytes) và địa chỉ nền của mảng A lưu trong thanh ghi  $\$s3$

```
for(i = 1; i < j; i++)  

    A[4] = 5 + A[i*16];  

    j = 6;
```

## **Duyệt đề Khoa/Bộ Môn**

Giáo viên ra đề

# MIPS Reference Data


**CORE INSTRUCTION SET**

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

- (1) May cause overflow exception
- (2) SignExtImm = { 16{immediate[15]}, immediate }
- (3) ZeroExtImm = { 16{1'b0}, 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.

**ARITHMETIC CORE INSTRUCTION SET**

NAME, MNEMONIC	MAT	FOR-OPERATION	OPCODE / FMT / FUNCT (Hex)		
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]	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		FP Compare Single c.e.s*	FPcond = (F[fs] op F[ft]) ? 1 : 0	11/10/-/ly	
		FP Compare c.e.d*	FPcond = ({F[fs],F[fs+1]} op {F[ft],F[ft+1]}) ? 1 : 0	11/11/-/ly	
		*	(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		
Double		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	
		Double	FP Subtract Single sub.s	F[fd]=F[fs] - F[ft]	11/10/-/1
			FP Subtract sub.d	[F[fd],F[fd+1]] = {F[fs],F[fs+1]} - {F[ft],F[ft+1]}	11/11/-/1
Load FP Single	lwcl	I F[rt]=M[R[rs]+SignExtImm]	31/-/-/-		
Load FP	ldcl	I F[rt]=M[R[rs]+SignExtImm]; F[rt+1]=M[R[rs]+SignExtImm+4]	35/-/-/-		
Double		Move From Hi mfh1	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] >> shampt	0/-/-/3	
		Store FP Single swcl	I M[R[rs]+SignExtImm] = F[rt]	(2) 39/-/-/-	
		Store FP adcl	M[R[rs]+SignExtImm] = F[rt]; M[R[rs]+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	0
FI	opcode	fmt	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	Yes