TRƯỜNG ĐAI HOC CÔNG NGHÊ THÔNG TIN KHOA KỸ THUẬT MÁY TÍNH

Mã đề số: 10101011

ĐỀ THI CUỐI HK 1 (2020-2021), ĐƠT 2 Môn thi: Tổ chức và Kiến trúc Máy tính 2. Thời gian thi: 80 phút

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

Chữ ký của	Cán bộ coi thi	
<u>STT</u> 	Họ và tên: MSSV: Phòng thi:	ĐIEM Bằng số: Bằng chữ:
BÅNG TRÅ	LÒI TRẮC NGHIỆM (SV ghi đáp án đúng vào bảng s	rau)

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	Câu 24						

Phần 1: Tự Luận (4 điểm)

Câu 1: Chuyển đoạn mã C sau dang mã hợp ngữ MIPS, giả sử rằng i, j được lưu trong thanh ghi \$s2, \$s3. Base address của A, B lưu trong thanh ghi S6, S7. (1 điểm)

$$B[4] = A \left[2i + j \right]$$

Câu 2: Cho độ trễ của các khối trong datapath như sau (1.5 điểm):

I-Mem	Add	Mux	ALU	Regs (Read)	Regs (write)	D-Mem	Sign-extend	Shift-left-2
400ps	100ps	30ps	120ps	200ps	0 ps	350ps	20ps	0ps

- a) Chu kỳ xung clock là bao nhiêu nếu datapath chỉ hỗ trợ các lệnh thuộc nhóm logic và số học (như add, and, ...)? (0.5 điểm)
- b) Chu kỳ xung clock là bao nhiêu nếu datapath chỉ hỗ trợ lệnh lw? (0.5 điểm)
- c) Chu kỳ xung clock là bao nhiêu nếu datapath hỗ trợ các lệnh: add, beq, lw, sw? (0.5 điểm)

Câu 3 Các câu bên dưới sử dụng dữ liệu ở bảng sau, giả sử các bộ xử lý có cùng một kiến trúc tập lệnh. (1.5 điểm)

Processor Rate	Clock	Số lệnh	Thời gian thực thi
P1	2 GHz	20.10^9	7s
P2	1.5 GHz	30.109	10s

Đề 10101011 - Trang 1/4

 a) Tìm IPC (số lệnh được thực hiện trong một chu kì – instruction per cycle) cho mỗi bộ xử lý. b) Tìm tần số xung clock mới cho P2 để P2 có thể giảm thời gian thực thi bằng P1. c) Tìm số lượng lệnh cho P2 mà giảm thời gian thực thi của nó tới bằng của P3.

 90.10^9

9s

3 GHz

P3

Phần 2: Trắc Nghiệm (6 Điểm)

Câu 1	Cá mấy loại		núa tân lânh MIDS?			
A. 2	Co may loại	toán hạng trong kiến t B. 3	C. 4	D. 5		
Câu 2	Trong lênh	2.	trường nào trong định dạng			
A. rs	Trong içini	B. rt	C. rd	D. imm		
Câu 3	Chon nhát l	piểu đúng về ý nghĩa củ		D. mmi		
			niện bước truy xuất đọc bộ nhó	y và lưu vào thanh ghi \$t3		
			niện bước truy xuất ghi bộ nhớ			
		trong lệnh này là 13	nọn cuốc truy xuất giá bộ mio	eo giù trị là thàm gin ứt		
		ni lệnh, ALU thực hiện pl	hép tính \$t1 + 4			
Câu 4		piểu sai về tập thanh gh	•			
		thi được đánh theo byte				
		ôn có giá trị không đổi				
	_	ợc sử dụng để trả về giá	trị hàm con			
		trị của các thanh ghi S				
Câu 5	Lệnh "addi	\$t0, \$s2, -8" có mã máy	là bao nhiệu?			
A. 0x26		B. 0x2248FFF8	C. 0x2684FFF8	D. 0x2248F1F8		
Câu 6	Mã máy "0x	xad50fff9" là của lệnh h	iợp ngữ nào sau đây?			
A. sw \$	t0, -7(\$t2)	B. sw \$s0, 7(\$t2)	C. lw \$s0, -7(\$t2)	D. sw \$16, -7(\$10)		
Câu 7	Lệnh nào sa	u đây không phải là lệr	nh điều khiển			
A. ja	•	B. bne	C. j	D. beq		
Câu 8	Khai báo " E	Bien1 .word 32" trong ch	ương trình hợp ngữ, Bien1 đượ	re cấp phát vùng nhớ là bao		
nhiêu'						
A. 32 bi	it	B. 32 word	C. 4 word	D. 32 byte		
Câu 9			huỗi ký tự "welcomeToUIT"			
A. 3		B. 12	C. 4	D. Cả ba đều sai		
Câu 10	Giai đoạn đ	oc opcode để xác định l	kiểu lệnh thuộc công đoạn nà	o trong chu kỳ thực thi		
lệnh I	MIPS?					
A. ALU	J	B. Memory Access	C. Instruction Decode	D. Result write		
Câu 11	Mạch nào ti	rong các mạch sau tron	g datapath là mạch tổ hợp?			
A. I-MI			B. D-MEM			
C. Regi	ster		D. ALU			
Câu 12	Địa chỉ than	nh ghi PC sẽ tăng bao n	hiêu sau mỗi lần đọc lệnh ?	_		
A. 1			B. 2			
C. 4			D. Không thay đổi			
Câu 13	Cho đoạn cl	hương trình sau :				
1xx \$xx1 0(\$	(20)		A. 4 & 2	1		
lw \$v1, 0(\$,		A. 4 & 2			
addi \$v0, \$			B. 2 & 4			
sw \$v1, 0(\$,		C. 2 & 2			
addi \$a0, \$a0, 1 Hỏi bộ nhớ lệnh và bộ nhớ dữ liệu được truy						
		nno au neu auợc truy	D. 4 & 4			
cập mấy lầ		· \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \		1 0		
Câu 14		iang nao thuọc datapati	h KHÔNG tham gia vào lệnh	I IW ?		
A. I-Me	em			B. Register		
C. Add		-16 LL: (1 1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.	D. ALU	: 640 120		
Câu 15			ệnh "sll \$t0, \$s0, 2" thì giá tr			
A. 0x04	ŀ	B. 0x58	C. 0x32	D. 0x64		

Câu 16 Cho bảng sau:

Địa chỉ	Giá trị
0x10010014	0x00000064
0x10010018	0x00000068
0x1001001c	0x0000001c

Cho \$s3 = 0x10010000, sau khi thực hiện lệnh lw \$t3,12(\$s3) giá trị \$t3 là?

Câu 17 Trong datapath của kiến trúc MIPS loại lệnh R-Type không sử dụng phần nào? A. Bộ ALU B. Tập thanh ghi C. Bộ ALU control D. Bộ nhớ dữ Câu 18 Phát biểu nào không phải là chức năng của trình biên dịch	liệu					
Câu 18 Phát biểu nào không phải là chức năng của trình biên dịch	liệu					
	01 0 1					
A. Chuyển đổi từ mã C sang mã hợp ngữ						
B. Chuyển đổi từ Java sang mã hợp ngữ						
C. Chuyển đổi ngôn ngữ thông dịch sang ngôn ngữ hợp ngữ						
D. Chuyển đổi ngôn ngữ biên dịch sang ngôn ngữ hợp ngữ	_					
Câu 19 Khi thực hiên thao tác "push" dữ liệu vào stack, giá trị của thanh ghi \$sp thay đổi như thế nào?						
A. Tăng lên 1 B. Giảm 1 C. Tăng lên 4 D. Giảm 4						
Câu 20 Khi thực hiên thao tác "pop" dữ liệu vào stack, giá trị địa chỉ thanh ghi \$sp thay đổi như	thế					
nào?						
A. Tăng lên 1 B. Không thay đổi C. Tăng lên 4 D. Giảm 4						
Câu 21 Khi thực hiện lệnh nào, giá trị tín hiệu MemtoReg là tùy định?						
A. lw B. sw C. add D. addi						
Câu 22 Cạnh xuống của xung clock được hiểu là?						
A. Tại thời điểm giá trị xung clock bằng 0						
B. Tại thời điểm giá trị xung clock bằng 1						
C. Tại thời điểm giá trị xung clock thay đổi từ 0 lên 1						
D. Tại thời điểm giá trị xung clock thay đổi từ 1 xuống 0						

Câu 23 Bảng dưới đây cho thấy 2 cách hiện thực trên 2 máy tính M1 và M2 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	CPI cho M2	Tỷ lệ lệnh
A	1	2	40%
В	3	2	30%
С	4	2	30%

Hiệu suất của máy 2 so với máy 1 như thế nào?

Câu 24 Một thuật toán tìm kiếm giá trị trong mãng được hiện thực bởi hai kỹ sư lập trình. Kỹ sư 1 và 2 hiện thực thuật toán tương ứng cho mỗi chương trình là 100 lệnh và 120 lệnh. Để so sánh dưới đây cho thấy 2 cách hiện thực trên 1 máy tính M có một tập lệnh bao gồm 3 lớp lệnh (instruction class) A, B và C. Số lượng lệnh được thực thi trong một đoạn chương trình X và Y tương ứng cho mỗi kỹ sư được thể hiện ở cột cuối cùng.

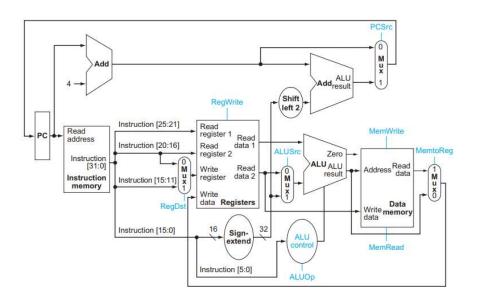
Lớp	CPI cho M	Tỷ lệ lệnh X	Tỷ lệ lệnh Y
A	1	30%	50%

В	3	40%	40%
C	4	30%	10%

Thông số CPI trung bình của đoạn chương trình Y trên máy tính M?

A. 2.1	B. 2.3	C. 2.5	D. 2.7

------ Hết ------



Duyệt đề của Khoa/Bộ Môn

(ký và ghi rõ họ tên)

Giảng viên ra đề

(ký và ghi rõ họ tên)

Trịnh Lê Huy

Trương Văn Cương

MIDC			_ 0	4	7							(FMT
MIPS	Ref	er	ence Data	6		N: 43.45	MNEMO		FOR- MAT		OPERAT	TON		/ FUN (He:
				-	OPCODE		n FP True					BranchAddr	(4)	11/8
CORE INSTRUCTION		OR-			/ FUNCT		n FP False					+BranchAddi		11/8
NAME, MNEMO	transfer to the	MAT			(Hex)	Divide		div	R	Lo=R[rs]/R[rt]; Hi=R			0//-
Add	add		R[rd] = R[rs] + R[rt]	(1)	0 / 20 _{bex}	Divide U		divu		Lo=R[rs]/R[[rs]%R[rt]	(6)	0//-
Add Immediate	addi	I	R[rt] = R[rs] + SignExtImm	(1,2)		FP Add S FP Add	ingle	add.s		F[fd]=F[fs]		6.1 266 (11)		11/10/
		1	R[rt] = R[rs] + SignExtImm	(2)		Double		add.d	FR	(Flia),Flia+		(fs],F[fs+1]) - [ft],F[ft+1])		11/11/
Add Unsigned	addu		R[rd] = R[rs] + R[rt]	(4)	0/21 _{hex}		are Single	c.x.s*	FR	FPcond = (F				11/10
					0 / 24 _{hex}	FP Comp			FR	FPcond = ([F[fs],F[fs	+1]) op		11/11/
And	and		R[rd] = R[rs] & R[rt]	190		Double		- CANGE			F[ft],F[ft	i+I]])?1:0		
And Immediate	andi.	1	R[rt] = R[rs] & ZeroExtImm	(3)	chex	FP Divide	Seq. It. (or Le) (P 15	==, <, or <=) F[fd] = F[fs]	(y is 32,	3c, or 3e)		11/10/
Branch On Equal	beq	1	if(R[rs]==R[rt]) PC=PC+4+BranchAddr	(4)	4 _{hex}	FP Divide Double		div.d			$1]) = \{F F F F F F F F F F $	[fs],F[fs+1]) [ft],F[ft+1])	1	11/11/
Branch On Not Equal	Ibne	1	if(R[rs]!=R[rt]) PC=PC+4+BranchAddr	(4)	5 _{hex}	FP Multip				F[fd] = F[fs]	* F[ft]			11/10/
Jump	5	J	PC=JumpAddr	(5)	2 _{hex}	FP Multip Double	oly	mul.d	FR	[F[1d],F[1d+		[fs],F[fs+1]) ' '[ft],F[ft+1])		11/11/
Jump And Link	jal	J	R[31]=PC+8;PC=JumpAddr	(5)			ct Sinele	sub.s	FR	F[fd]=F[fs]		fells (u. 13)		11/10/
Jump Register	1E		PC=R[rs]		0 / 08 _{bex}	FP Subtra		sub.d				[fs],F[fs+1]}		11/11/
	100		R[rt]={24'b0,M[R[rs]			Double					{F	[n],F[n+1]		
Load Byte Unsigned	1bu	1	+SignExtImm](7:0))	(2)	24 _{hex}	Load FP	Single	lwc1	1	F[rt]=M[R[r			0.00	31//-
Load Halfword	Lhu	T	R[rt]={16'b0,M[R[rs]		25 _{hex}	Load FP		idel	1	F[rt]=M[R[r			(2)	35//-
Unsigned		-50	+SignExtImm](15:0))	(2)		Double Move Fro	om Hi	m£hi	R	R[rd] = Hi	eftal+218	nExtImm+4]		0/-/-
Load Linked	11	1	R[rt] = M[R[rs]+SignExtImm]	(2,7)	30 _{hex}	Move Fro		mflo	R	R[rd] = Lo				0/-/-
Load Upper Imm.	lui	1	$R[rt] = \{imm, 16'b0\}$		f _{hex}	Move Fro	om Contro	mfc0	R	R[rd] = CR[10 /0/
Load Word	1w	I	R[rt] = M[R[rs]+SignExtImm]	(2)	- September -	Multiply		mult	R	$\{Hi,Lo\} = R$	[rs] * R[rt]		0//-
Nor	nor	R	$R[rd] = \sim (R[rs] \mid R[rt])$		0 / 27 _{bex}		Unsigned			$\{Hi,Lo\} = R$			(6)	0//-
Or	05	R	R[rd] = R[rs] R[rt]		0 / 25 _{bex}	Shift Rig Store FP		seci.	R 1	R[rd] = R[rt] M[R[rs]+Sig			(2)	39//-
Or Immediate	ori	ī	R[rt] = R[rs] ZeroExtlmm	(3)		Store FP	onigie			M[R[rs]+Sig			(2)	
Set Less Than	alt	R	R[rd] = (R[rs] < R[rt]) ? 1 : 0	100	0/2a _{bes}	Double		sdcl	I			n+4] = $F[rt+1]$		3d//
Set Less Than Imm.	siti	I	R[rt] = (R[rs] < SignExtImm)?	: 0.(2)		EL OATIS	IG-POINT	TINGT	auc	TION FORM	ATS			
Set Less Than Imm.			R[rt] = (R[rs] < SignExtImm)	1134		FR		-	imt	ft	fs	fd		func
	sltiu	1	?1:0	(2,6)			opcode 31	26 25	3	21 20 1	6 15	11 10	6.5	Tunc
Unsigned	situ	R	$R[rd] = (R[rs] \le R[rt]) ? 1 : 0$	(6)	0 / 2b _{bex}	FI	opcode		int	ft		immed	iate	
Set Less Than Unsig.		R	$R[rd] = R[rt] \ll shamt$		0 / 00 _{bex}		31	26 25		21 20 1	6 15			
Set Less Than Unsig. Shift Left Logical	s11		DI-11 - DI-11 was always		0 / 02 _{hex}	PSEUDO	DINSTRU		SET			2222		
Set Less Than Unsig.		R	R[rd] = R[rt] >>> shamt								IC	OPERA		bel
Set Less Than Unsig. Shift Left Logical	sll	R I	$\begin{aligned} M[R[rs]+SignExtImm](7:0) = \\ R[rt](7:0) \end{aligned}$	(2)	28 _{hex}		NA ch Less Tl	nan		MNEMON	if(R[r	s] <r[n]) :<="" pc="" td=""><td></td><td></td></r[n])>		
Set Less Than Unsig. Shift Left Logical Shift Right Logical	sll srl		$\begin{split} M[R[rs]+SignExtImm](7:0) &= \\ R[rt](7:0) \\ M[R[rs]+SignExtImm] &= R[rt]; \end{split}$	2775	19.	Bran	NA ch Less Tl ch Greater	nan Than		blt bgt	if(R[r if(R[r	s]>R[rt]) PC		
Set Less Than Unsig. Shift Left Logical Shift Right Logical Store Byte	sll srl sb	1	$\begin{split} &M[R[rs]+SignExtImm](7:0) = \\ &R[rt](7:0) \\ &M[R[rs]+SignExtImm] = R[rt]; \\ &R[rt] = (atomic) ? 1:0 \end{split}$	(2)	19.	Bran Bran	NA ch Less Tl	nan Than nan or E	qual	blt bgt ble	if(R[r if(R[r if(R[r		= L	abel
Set Less Than Unsig. Shift Left Logical Shift Right Logical Store Byte	sll srl sb	1	M[R[rs]+SignExtImm](7;0) = R[rt](7;0) M[R[rs]+SignExtImm] = R[rt]; R[rt] = (atomic) ? 1 : 0 M[R[rs]+SignExtImm](15;0) =	(2,7)	38 _{bes}	Bran Bran Bran	NA ch Less Tl ch Greater ch Less Tl	nan Than nan or E Than o	qual	blt bgt ble	if(R[r if(R[r if(R[r if(R[r R[rd]	s]>R[n]) PC s]<=R[n]) PC s]>=R[n]) PC = immediate	= L	abel
Set Less Than Unsig. Shift Left Logical Shift Right Logical Store Byte Store Conditional Store Halfword	sll srl sb sc sh	1 1	$\begin{split} &M[R[rs]+SignExtImm](7:0) = \\ &R[rt](7:0) \\ &M[R[rs]+SignExtImm] = R[rt]; \\ &R[rt] = (atomic) ? 1:0 \\ &M[R[rs]+SignExtImm](15:0) = \\ &R[rt](15:0) \end{split}$	(2,7)	38 _{hex}	Bran Bran Bran	NA ch Less TI ch Greater ch Less TI ch Greater I Immediat	nan Than nan or E Than o	qual	blt bgt ble al bge	if(R[r if(R[r if(R[r if(R[r R[rd]	s]>R[n]) PC : s]<=R[n]) PC s]>=R[n]) PC	= L	abel
Set Less Than Unsig. Shift Left Logical Shift Right Logical Store Byte Store Conditional Store Halfword Store Word	sll srl sb sc sh	1 1 1	$\begin{split} M[R[rs]+SignExtImm](7:0) &= \\ R[rt](7:0) \\ M[R[rs]+SignExtImm] &= R[rt]; \\ R[rt] &= (atomic) ? 1:0 \\ M[R[rs]+SignExtImm](15:0) &= \\ R[rt](15:0) \\ M[R[rs]+SignExtImm] &= R[rt] \end{split}$	(2,7)	38 _{hex} 29 _{hex} 2b _{hex}	Bran Bran Bran Load Mov	NA ch Less Tl ch Greater ch Less Tl ch Greater I Immediat e	nan Than nan or E Than o	qual r Equ	blt bgt ble al bge	if(R[r if(R[r if(R[r if(R[r R[rd] R[rd]	s]>R[n]) PC s]<=R[n]) PC s]>=R[n]) PC = immediate = R[rs]	C = L	abel abel
Set Less Than Unsig. Shift Left Logical Shift Right Logical Store Byte Store Conditional Store Halfword Store Word Subtract	sll srl sb sc sh sw	I I I R	$\begin{split} &M[R[rs]+SignExtImm](7:0) = \\ &R[rt](7:0) \\ &M[R[rs]+SignExtImm] = R[rt]; \\ &R[rt] = (atomic) ? 1:0 \\ &M[R[rs]+SignExtImm](15:0) = \\ &R[rt](15:0) \\ &M[R[rs]+SignExtImm] = R[rt] \\ &R[rd] = R[rs] - R[rt] \end{split}$	(2,7)	38 _{hex} 29 _{hex} 2b _{hex} 0 / 22 _{hex}	Bran Bran Bran Load Mov REGIST	NA ch Less TI ch Greater ch Less TI ch Greater I Immediat e	Than Than or E Than or te E, NUM	qual r Equ	blt bgt ble al bge li move	if(R[r if(R[r if(R[r if(R[r R[rd] R[rd]	s]>R[n]) PC s]<=R[n]) PC s]>=R[n]) PC = immediate = R[rs] ENTION PRESERV	Z=L Z=L	abel abel
Set Less Than Unsig. Shift Left Logical Shift Right Logical Store Byte Store Conditional Store Halfword Store Word	sll srl ab sc sh sw sub	I I I R R	$\begin{split} M[R[rs]+SignExtImm](7:0) &= \\ R[rt](7:0) \\ M[R[rs]+SignExtImm] &= R[rt], \\ R[rt] &= (atomic) ? 1:0 \\ M[R[rs]+SignExtImm](15:0) &= \\ R[rt](15:0) \\ M[R[rs]+SignExtImm] &= R[rt] \\ R[rd] &= R[rs] - R[rt] \\ R[rd] &= R[rs] - R[rt] \\ R[rd] &= R[rs] - R[rt] \\ \end{split}$	(2,7)	38 _{hex} 29 _{hex} 2b _{hex}	Bran Bran Bran Load Mov REGIST	NA ch Less TI ch Greater ch Less TI ch Greater I Immediat e ER NAMI	Than Than or E Than or E Than or E NUM UMBER	qual r Equ BER	blt bgt ble al bge li move t, USE, CAL	if(R[r if(R[r if(R[r if(R[r R[rd] R[rd]	s]>R[rt]) PC s]<=R[rt]) PC s]>=R[rt]) PC = immediate = R[rs] ENTION PRESERV	C=L C=L VED CAL	abel abel ACRO
Set Less Than Unsig. Shift Left Logical Shift Right Logical Store Byte Store Conditional Store Halfword Store Word Subtract	sll srl sb sc sh sw sub sub	I I I R R	$\begin{split} M[R[rs]+SignExtImm](7:0) &= &R[rt](7:0) \\ M[R[rs]+SignExtImm] &= R[rt]; \\ R[rt] &= (atomic)? 1:0 \\ M[R[rs]+SignExtImm](15:0) &= \\ R[rt](15:0) \\ M[R[rs]+SignExtImm] &= R[rt] \\ R[rd] &= R[rs] - R[rt] \\ R[rd] &= R[rs] - R[rt] \\ se overflow exception \\ \end{split}$	(2,7) (2) (2) (1)	38 _{hex} 29 _{hex} 2b _{bex} 0/22 _{hex} 0/23 _{hex}	Bran Bran Bran Load Mov REGIST	NA ch Less The ch Greater ch Less The ch Greater I Immediate ER NAMI AME Ni zero	Than Than on E Than or E Than or E NUM UMBER	qual r Equ BER	blt bgt ble al bge li move t, USE, CAL USE	if(R[r if(R[r if(R[r if(R[r R[rd] R[rd] L CONV	s]>R[rt]) PC s]<=R[rt]) PC s]>=R[rt]) PC = immediate = R[rs] ENTION PRESERV	C=L C=L VED CAL N.A.	abel abel ACRO L?
Set Less Than Unsig. Shift Left Logical Shift Right Logical Store Byte Store Conditional Store Halfword Store Word Subtract	sll srl sb sc sh sw sub subu (1) May (2) Sign	I I I R R cau	$\begin{split} M[R[rs]+SignExtImm](7:0) &= \\ R[rt](7:0) \\ M[R[rs]+SignExtImm] &= R[rt], \\ R[rt] &= (atomic) ? 1:0 \\ M[R[rs]+SignExtImm](15:0) &= \\ R[rt](15:0) \\ M[R[rs]+SignExtImm] &= R[rt] \\ R[rd] &= R[rs] - R[rt] \\ R[rd] &= R[rs] - R[rt] \\ R[rd] &= R[rs] - R[rt] \\ \end{split}$	(2,7) (2) (2) (1)	38 _{hex} 29 _{hex} 2b _{bex} 0/22 _{hex} 0/23 _{hex}	Bran Bran Bran Load Mov REGIST	NA ch Less TI ch Greater ch Less TI ch Greater I Immediat e ER NAMI	Than or E Than or E Than or e E, NUM UMBER	qual r Equ BER t The	blt bgt ble al bge 11 move a, USE, CAL USE c Constant Va	if(R[r if(R[r if(R[r if(R[rd] R[rd] R[rd] L CONV	S >R[n] PC S >=R[n] PC S >=R[n] PC = immediate = R[rs] ENTION PRESERV A (VED CAL N.A.	abel abel ACRO L?
Set Less Than Unsig. Shift Left Logical Shift Right Logical Store Byte Store Conditional Store Halfword Store Word Subtract	sll srl sb sc sh sw sub sub (1) May (2) Sig (3) Zen (4) Bra	I I I R R cau hExt	M[R[rs]+SignExtImm](7:0) = R[rt](7:0) M[R[rs]+SignExtImm] = R[rt]; R[rt] = (atomic) ? 1 : 0 M[R[rs]+SignExtImm](15:0) = R[rt](15:0) M[R[rs]+SignExtImm] = R[rt] R[rd] = R[rs] - R[rt] R[rd] = R[rs] - R[rt] se overflow exception Imm = {16{immediate[15]}, imm mm = {16{immediate[15]}, imm ddr = {14{immediate[15]}, imm	(2,7) (2) (2) (1) nediate	38 _{hex} 29 _{hex} 2b _{hex} 0 / 22 _{hex} 0 / 23 _{hex}	Bran Bran Bran Load Mov REGIST N	NA ch Less TI ch Greater ch Less TI ch Greater I Immediat e ER NAMI AME NI izero Sat 0-Sv1	Than or E.	qual r Equ BER The Ass Val	blt ble ble al bge li move l, USE, CAL USE c Constant Va sembler Temp ues for Funct	if(R[r if(R[r if(R[r if(R[r R[rd] R[rd] L CONV	s >R[n]) PC s =R[n]) PC s =R[n]) PC = immediate = R[rs] ENTION PRESERV A 0	Z = L Z = L Z = L Z = L Z = L Z = L Z = L	abel abel ACRO L?
Set Less Than Unsig. Shift Left Logical Shift Right Logical Store Byte Store Conditional Store Halfword Store Word Subtract	sll srl sb sc sh sub subu (1) May (2) Sig (3) Zen (4) Bra (5) Jun (6) Opo	I I I R R R y cau hExt	M[R[rs]+SignExtImm](7:0) = R[rt](7:0) M[R[rs]+SignExtImm] = R[rt]; R[rt] = (atomic) ? 1 : 0 M[R[rs]+SignExtImm](15:0) = R[rt](15:0) M[R[rs]+SignExtImm] = R[rt] R[rd] = R[rs] - R[rt] Minm = { 16 [immediate[15]], imm mm = { 16 [immediate[15]], im	(2,7) (2) (2) (1) nediate ediate, b0)	38 _{bex} 29 _{hex} 2b _{hex} 0 / 22 _{bex} 0 / 23 _{bex} 1 , 2'b0)	Bran Bran Bran Load Mov REGIST N Sv	NA ch Less Ti ch Greater ch Less Ti ch Greater I Immediat e ER NAMI AME Ni zero Sat 0-Sv1	Than or E. NUM UMBER 0 1 2-3 4-7	qual r Equ BER The Ass Val and Arg	blt bgt ble al bge 11 move 1, USE, CALI USE c Constant Va sembler Temp uses for Funct l Expression I guments	if(R[r if(R[r if(R[r if(R[r R[rd] R[rd] L CONV	s >R[n]) PC s =R[n]) PC s =R[n]) PC = immediate = R[rs] ENTION PRESERV A 0	VED CAL No No	abel abel ACRO L?
Set Less Than Unsig. Shift Left Logical Shift Right Logical Store Byte Store Conditional Store Halfword Store Word Subtract	sll srl sb sc sh sub subu (1) May (2) Sig (3) Zen (4) Bra (5) Jun (6) Opo	I I I R R R y cau hExt	M[R[rs]+SignExtImm](7:0) = R[rt](7:0) M[R[rs]+SignExtImm] = R[rt]; R[rt] = (atomic)? 1:0 M[R[rs]+SignExtImm] = R[rt]; R[rd] = R[rs] - R[rt]; R[rd	(2,7) (2) (2) (1) nediate ediate, b0)	38 _{bex} 29 _{hex} 2b _{hex} 0 / 22 _{bex} 0 / 23 _{bex} 1 , 2'b0)	Bran Bran Load Mov REGIST N Sv Sa	NA ch Less Ti ch Greater ch Less Ti ch Greater it Immediat e ER NAMI AME Ni czero Sat 0-Sv1 0-Sa3 0-St7	Than on E. Than one E, NUM UMBER 0 1 2-3 4-7 8-15	qual r Equi	blt ble al bge li move l, USE, CAL USE c Constant Va sembler Temp ues for Funct l Expression I guments mporaries	if(R[r if(R[r if(R[r if(R[r R[rd] R[rd] L CONV	s >R[n]) PC s =R[n]) PC s =R[n]) PC = immediate = R[rs] ENTION PRESERV A 0	VED CAL NA No No	abel abel ACRO L?
Set Less Than Unsig. Shift Left Logical Shift Right Logical Store Byte Store Conditional Store Halfword Store Word Subtract	sll srl sb sc sh sub subu (1) May (2) Sig (3) Zen (4) Bra (5) Jup (6) Opp (7) Ato	I I I R R R R soExt	$\begin{split} M[R[rs]+SignExtImm](7:0) &= \\ R[rt](7:0) \\ M[R[rs]+SignExtImm] &= R[rt]; \\ R[rt] &= (atomic) ? 1:0 \\ M[R[rs]+SignExtImm](15:0) &= \\ R[rt](15:0) \\ M[R[rs]+SignExtImm] &= R[rt] \\ R[rd] &= R[rs] - R[rt] \\ R[rd] &= R[rt] - R[rt] \\ R[rt] &= R[rt] - R[rt] - R[rt] - R[rt] \\ R[rt] &= R[rt] - R[rt] - R[rt] - R[rt] - R[rt] - R[rt] \\ R[rt] &= R[rt] - R[r$	(2,7) (2) (2) (1) nediate ediate, b0)	38 _{bex} 29 _{hex} 2b _{hex} 0 / 22 _{bex} 0 / 23 _{bex} 1 , 2'b0)	Bran Bran Bran Load Mov REGIST N Sv Sa Ss Ss	NA Ch Less Ti Ch Greater Ch Less Ti Ch Greater Immediat e ER NAMI AME Ni Zero Sat 0-Sv1 0-Sa3 0-St7 0-Ss7	Than one E, NUM UMBER 0 1 2-3 4-7 8-15	qual r Equ BER The Ass Val- and Arg Ten Sav	blt bgt ble al bge li move i, USE, CAL USE c Constant Va sembler Temp uses for Funct Expression I guments mporaries red Temporaries	if(R[r if(R[r if(R[r if(R[r R[rd] R[rd] L CONV	s >R[n]) PC s =R[n]) PC s =R[n]) PC = immediate = R[rs] ENTION PRESERV A 0	Z=L Z=L Z=L Z=L Z=L Z=L Z=L Z=L Z=L Z=L	abel abel ACRO L?
Set Less Than Unsig. Shift Left Logical Shift Right Logical Store Byte Store Conditional Store Halfword Store Word Subtract Subtract Unsigned	sll srl sb sc sh sub subu (1) May (2) Sig (3) Zen (4) Bra (5) Jup (6) Opp (7) Ato	I I I R R R September 1 I R R R R R R R R R R R R R R R R R R	$\begin{split} M[R[rs]+SignExtImm](7:0) &= \\ R[rt](7:0) \\ M[R[rs]+SignExtImm] &= R[rt]; \\ R[rt] &= (atomic) ? 1:0 \\ M[R[rs]+SignExtImm](15:0) &= \\ R[rt](15:0) \\ M[R[rs]+SignExtImm] &= R[rt] \\ R[rd] &= R[rs] - R[rt] \\ R[rd] &= R[rt] - R[rt] \\ R[rt] &= R[rt] - R[rt] - R[rt] - R[rt] \\ R[rt] &= R[rt] - R[rt] - R[rt] - R[rt] - R[rt] - R[rt] \\ R[rt] &= R[rt] - R[r$	(2,7) (2) (2) (1) nediate sediate, b0 } /s. 2's sic, 0 if	38 _{bex} 29 _{hex} 2b _{hex} 0 / 22 _{bex} 0 / 23 _{bex} 1 , 2'b0)	Bran Bran Bran Load Mov REGIST N Sv Sa Si Si Si Si Si	NA ch Less Ti ch Greater ch Less Ti ch Greater I Immediate e ER NAMI Ni zero Sat 0-Sv1 0-Ss3 0-St7 0-Ss7 8-St9	Than or E Than or E Than or E Than or E E, NUM UMBER 0 1 2-3 4-7 8-15 16-23 24-25	qual r Equ BER The Ass Val- and Arg Ten Sav	blt bgt ble al bge 11 move 1, USE, CAL USE Constant Va sembler Tempor uses for Funct I Expression I guments mporaries ved Temporari mporaries	if(R[r if(R[r if(R[r if(R[r R[rd] R[rd] R[rd] L CONV	s >R[n]) PC s =R[n]) PC s =R[n]) PC = immediate = R[rs] ENTION PRESERV A 0	VED CAL N.A. No No No Yes No	abel abel ACRO L?
Set Less Than Unsig. Shift Left Logical Shift Right Logical Store Byte Store Conditional Store Halfword Store Word Subtract Subtract Unsigned BASIC INSTRUCT R opcode	sll srl sb sc sh sw sub sub (1) Ma (2) Sig (3) Zer (4) Bra (5) Jun (6) Opc (7) Ato	I I I I R R R R cau nExt oExt nch/	M[R[rs]+SignExtImm](7:0) = R[rt](7:0) M[R[rs]+SignExtImm] = R[rt]; R[rt] = (atomic) ? 1 : 0 M[R[rs]+SignExtImm] = R[rt] R[rt] = R[rs] - R[rt] R[rd] = R[rs]	(2,7) (2) (2) (1) nediate sediate, b0 } (s. 2's sic, 0 if	38 _{bex} 29 _{bex} 2b _{bex} 0 / 22 _{bex} 0 / 23 _{bex} 1 2'b0) comp.)	Bran Bran Bran Bran Load Mov REGIST N S S S S S S S S S S S S S S S S S S	NA ch Less TI ch Greater ch Less TI ch Greater l Immediate e EER NAMI AME NI czero Sat 0-Sv1 0-Ss3 0-St7 0-Ss7 8-St9 0-Sk1	Than or E Than or E Than or E Than or E E, NUM UMBER 0 1 2-3 4-7 8-15 16-23 24-25 26-27	The Ass Value and Arg Ten Res	blt bgt ble al bge 11 move I, USE, CALI USE Constant Va sembler Temp ues for Funct I Expression I guments inporaries red Temporari inporaries served for OS	if(R[r if(R[r if(R[r if(R[r R[rd] R[rd] R[rd] L CONV	s >R[n]) PC s =R[n]) PC s =R[n]) PC = immediate = R[rs] ENTION PRESERV A 0	CAL NO NO NO NO NO NO	abel abel ACRO L?
Set Less Than Unsig. Shift Left Logical Shift Right Logical Store Byte Store Conditional Store Halfword Store Word Subtract Subtract Unsigned BASIC INSTRUCT R opcode	sh sc sh sub	I I I R R R y cau pAd rranc mic RM i 2	M[R[rs]+SignExtImm](7:0) = R[rt](7:0) M[R[rs]+SignExtImm] = R[rt]; R[rt] = (atomic) ? 1:0 M[R[rs]+SignExtImm](15:0) = R[rt](15:0) M[R[rs]+SignExtImm] = R[rt] R[rd] = R[rs] - R[rt] R[rd] = R[rs] - R[rt] se overflow exception Imm = {16{immediate[15]}, imm mm = {16{immediate[15]}, imm dr = {14{immediate[15]}, imm dr = {14{immediate[15]}, imm dr = {14{immediate[15]}, imm dr = {17{immediate[15]}, imm dr = {17{imme	(2,7) (2) (2) (1) nediate sediate, b0) /s. 2's nic, 0 if	38 _{hex} 29 _{hex} 2b _{hex} 0 / 22 _{hex} 0 / 23 _{hex} 1 2'b0) comp.) not atomic	Bran Bran Bran Bran Bran Bran Bran Bran	NA ch Less TI ch Greater ch Less TI ch Greater I Immediat e ER NAMI AME NI zero Sat 0-Sv1 0-Sa3 0-St7 0-Ss7 8-Ss9 0-Sk1 Sgp	Than or E Than or E Than or E Than or E E, NUM UMBER 0 1 2-3 4-7 8-15 16-23 24-25	The Ass Value and Arg Ten Res Gle	blt bgt ble al bge 11 move 1, USE, CAL USE Constant Va sembler Tempor uses for Funct I Expression I guments mporaries ved Temporari mporaries	if(R[r if(R[r if(R[r if(R[r R[rd] R[rd] R[rd] L CONV	s >R[n]) PC s =R[n]) PC s =R[n]) PC = immediate = R[rs] ENTION PRESERV A 0	VED CAL N.A. No No No Yes No	abel abel ACRO L?
Set Less Than Unsig. Shift Left Logical Shift Right Logical Store Byte Store Conditional Store Halfword Store Word Subtract Subtract Unsigned BASIC INSTRUCT R opcode 31 I opcode	stll srll stll srll stll srll srll stll srll sr	I I I R R R y cau pAd crand mic PRM/ i 2	M[R[rs]+SignExtImm](7:0) = R[rt](7:0) M[R[rs]+SignExtImm] = R[rt]; R[rt] = (atomic) ? 1 : 0 M[R[rs]+SignExtImm](15:0) = R[rt](15:0) M[R[rs]+SignExtImm] = R[rt] R[rd] = R[rs] - R[rt] R[rd] = R[rs] - R[rt] se overflow exception Imm = { 16 {immediate[15]}, imm Imm = { 16 {immediate[15]}, imm dr = { PC+4[31:28], address, 2" is considered unsigned numbers (vicest&set pair, R[rt] = 1 if pair atom ATS rt rd sham 1.20 16.15 11.10	(2,7) (2) (2) (1) nediate sediate, b0) /s. 2's nic, 0 if	38 _{hex} 29 _{hex} 2b _{hex} 0 / 22 _{hex} 0 / 23 _{hex} 1 2'b0) comp.) not atomic	Bran Bran Bran Load Mov REGIST N S S S S S S S S S S S S S S S S S S	NA ch Less TI ch Greater ch Less TI ch Greater l Immediate e EER NAMI AME NI czero Sat 0-Sv1 0-Ss3 0-St7 0-Ss7 8-St9 0-Sk1	Than or E Than or E Than or E Than or E Than or E E, NUM UMBER 0 1 2-3 4-7 8-15 16-23 24-25 26-27	qual r Equi	blt bgt ble al bge li move I, USE, CAL: USE c Constant Va sembler Temp uses for Funct Expression I guments inporaries wed Temporari inporaries served for OS obal Pointer	if(R[r if(R[r if(R[r if(R[r R[rd] R[rd] R[rd] L CONV	s >R[n]) PC s =R[n]) PC s =R[n]) PC = immediate = R[rs] ENTION PRESERV A 0	CAL NO NO NO NO NO Yes NO Yes	ACRO:

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

1. Bảng chuẩn đầu ra môn học

CĐRMH	Mô tả CĐRMH (mục tiêu cụ thể)	Mức độ giảng dạy
G1.1 (2.1)	Trình bày được các kiến thức cơ bản về kiến trúc máy tính và lập trình hợp ngữ.	I, T
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.	Т

2. Bảng câu hỏi và chuẩn đầu ra tương ứng đề thi cuối học kỳ I năm học 2020 – 2021

Câu Hỏi	Chuẩn Đầu
	ra
Phần 1: Tự Luận	
Câu 1	G1.1
Câu 2	G1.1, G1.2
Câu 3	G1.1
Phần 2: Trắc nghiệm	
Câu 1, 2, 3, 4, 5, 6, 7, 8, 9, 13, 15, 16, 17, 18, 19, 23, 24	G1.1
Câu 10, 11, 12, 14, 20, 21, 22	G1.2