



MÜHENDİSLİK VE MİMARLIK FAKÜLTESİ

SINAV KAĞIDI

2019/2020 EĞİTİM – ÖĞRETİM YILI/2020/2021 ACADEMIC YEAR
BAHAR DÖNEMİ/SEMESTER

Öğrencinin/Student's

Adı Soyadı/Name, Surname :

Numarası/Number :

Bölüm-Program /Department-Programme :

İmzası/Signature :

Kullanılan Kağıt Sayısı/Number of Papers Used:

Toplam Not – Paraf/Total Credit - Initials

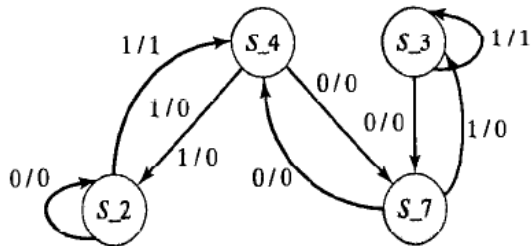
DERS Course	Adı/Name : Bilgisayar Mimarisi	SINAV Exam	Tarih/Date :								
	Kodu/Code : BLM202		Süresi/Duration : 120 Dk /Min								
	Sorumlusu/Lecturer : Dr. Öğr. Üyesi. Vecdi Emre Levent		Türü/Type : Vize / Midterm								
Soru Numarası/Numbers of the Questions	1	2	3	4	5	6	7	8	9	10	
Alınan Not/Scored Points											
SINAV KURALLARI/Exam Rules : 1. İstedığınız sorudan başlayabilirsiniz. 2. İlk 30 dakika soru sormak yasaktır.											

*Her sorunun puan değeri rakamsal olarak yanına belirtilmelidir./The points for each question must be stated next to the question.

SORULAR/Questions

Soru 1 (15 Puan):

Aşağıda verilen Mealy durum makinasının Verilog dilinde gerçekleyiniz.



Soru 2 (35 Puan):

RISC-V işlemci komut setinde (Instruction Set) aritmetik, bellek ve kontrol komutları bulunmaktadır. Bu komutlar aşağıda verilen tabloda özetlenmiştir. Komutların bit gösterimleri de ikinci tabloda verilmektedir.

Tip	Komut	Kod[3]	Kod[7]	İşlem
Opcode: 0x33 Register-Register (R-Type)	add rd, rs1, rs2	0x0	0x00	$R[rd] = R[rs1] + R[rs2]$
	mul rd, rs1, rs2	0x0	0x01	$R[rd] = (R[rs1] * R[rs2])[31:0]$
	sub rd, rs1, rs2	0x0	0x20	$R[rd] = R[rs1] - R[rs2]$
	sll rd, rs1, rs2	0x1	0x00	$R[rd] = R[rs1] \ll R[rs2]$
	mulh rd, rs1, rs2	0x1	0x01	$R[rd] = (R[rs1] * R[rs2])[63:32]$
	slt rd, rs1, rs2	0x2	0x00	$R[rd] = (R[rs1] < R[rs2]) ? 1 : 0$ (signed)
	xor rd, rs1, rs2	0x4	0x00	$R[rd] = R[rs1] \wedge R[rs2]$
	div rd, rs1, rs2	0x4	0x01	$R[rd] = R[rs1] / R[rs2]$
	srl rd, rs1, rs2	0x5	0x00	$R[rd] = R[rs1] \gg R[rs2]$
	or rd, rs1, rs2	0x6	0x00	$R[rd] = R[rs1] R[rs2]$
	rem rd, rs1, rs2	0x6	0x01	$R[rd] = R[rs1] \% R[rs2]$
	and rd, rs1, rs2	0x7	0x00	$R[rd] = R[rs1] \& R[rs2]$
Opcode: 0x03 Immediate (Değer) (I-Type)	lb rd, offset(rs1)	0x0		$R[rd] = \text{Mem}(R[rs1] + \text{offset})$ (byte) (SignedExtend)
	lh rd, offset(rs1)	0x1		$R[rd] = \text{Mem}(R[rs1] + \text{offset})$ (half) (SignedExtend)
	lw rd, offset(rs1)	0x2		$R[rd] = \text{Mem}(R[rs1] + \text{offset})$ (word)
Opcode: 0x13 Immediate (Değer) (I-Type)	addi rd, rs1, value	0x0	0x00	$R[rd] = R[rs1] + \text{value}$
	slli rd, rs1, value	0x1	0x00	$R[rd] = R[rs1] \ll \text{value}$
	slti rd, rs1, value	0x2	0x00	$R[rd] = (R[rs1] < \text{value}) ? 1 : 0$ (signed)
	xori rd, rs1, value	0x4	0x00	$R[rd] = R[rs1] \wedge \text{value}$
	srlr rd, rs1, value	0x5	0x00	$R[rd] = R[rs1] \gg \text{value}$
	ori rd, rs1, value	0x6	0x00	$R[rd] = R[rs1] \text{value}$
	andi rd, rs1, value	0x7	0x00	$R[rd] = R[rs1] \& \text{value}$
	Opcode: 0x23 Store (Kayıt) (S-Type)	sw rs2, offset(rs1)	0x2	
swge rs2, 0(rs1), offset		0x7		if($R[rs2] \geq \text{offset}$ (signed)) $\text{Mem}(R[rs1] + \text{offset}) = R[rs2]$
Opcode: 0x63 Branch (Dallanma) (B-Type)	beq rs1, rs2, offset	0x0		if($R[rs1] == R[rs2]$) $PC = PC + \text{offset}$
	blt rs1, rs2, offset	0x4		if($R[rs1] < R[rs2]$) (signed) $PC = PC + \text{offset}$
	bltu rs1, rs2, offset	0x6		if($R[rs1] < R[rs2]$) (unsigned) $PC = PC + \text{offset}$
Opcode: 0x37 Upper Immediate (U-Type)	lui rd, offset			$R[rd] = \text{offset}$
Opcode: 0x67 Upper Immediate(Atlama) (UJ-Type)	jal rd, value			$R[rd] = PC + 4$ $PC = PC + \text{value}$
Opcode: 0x6F Immediate(Atlama) (IJ-Type)	jalr rd, rs, value			$R[rd] = PC + 4$ $PC = R[rs] + \text{value}$

32-bit RISC-V Instruction Formats

Instruction Formats	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Register/register	funct7						rs2					rs1					funct3			rd					opcode							
Immediate	imm[11:0]											rs1					funct3			rd					opcode							
Upper Immediate	imm[31:12]															rd					opcode											
Store	imm[11:5]						rs2					rs1					funct3			imm[4:0]					opcode							
Branch	[12]	imm[10:5]						rs2					rs1					funct3			imm[4:1]				[11]	opcode						
Jump	[20]	imm[10:1]									[11]	imm[19:12]									rd					opcode						

- opcode (7 bit): partially specifies which of the 6 types of instruction formats
- funct7 + funct3 (10 bit): combined with opcode, these two fields describe what operation to perform
- rs1 (5 bit): specifies register containing first operand
- rs2 (5 bit): specifies second register operand
- rd (5 bit): Destination register specifies register which will receive result of computation

Saklayıcı 20: 10
Saklayıcı 21: 11
Program counter: 20

başlangıç değeri için



- a) beq 20, 21, 100 operasyonu sonucunda program counter değeri ne olur?
- b) bltu 20, 21, 100 operasyonu sonucunda program counter değeri ne olur?
- c) bltu 20, 21, 100 operasyonunun 32 bitlik gösterimini binary olarak veriniz.

Soru 3 (15 Puan):

Bir işlemcinin CPI değeri 1.5, 200 MHz ile çalışmaktadır. İkinci işlemci ise CPI değeri 1.55 ve 190 MHz de çalışmaktadır. Hangi işlemci saniyede daha fazla işlem yapabilmektedir?

Soru 4 (35 Puan):

Bir programda kullanılan komutların oranı ve kaç cycle sürdüğü bilgisi aşağıda verilmektedir.

- %25 dallanma (6 Cycles / Instruction)
- %25 aritmetik (2 Cycles / Instruction)
- %20 belleğe yükleme (5 Cycles / Instruction)
- %30 kontrol (3 Cycles / Instruction)

Buna göre bu işlemci;

Tek cycle'da işlemleri yaptığı modda 100 MHz'de
Çok cycle'da (pipelined) modda ise 150 MHz'de çalışabiliyor.

Her iki durum içinde saniyede kaç milyon işlem yapabildiğini (MIPS) hesaplayınız.