

Yıldız Teknik Üniversitesi, Bilgisayar Mühendisliği Bölümü, 2019-2020 Öğretim Yılı Bahar yy., BLM2612 Bilgisayar Donanımı Gr. 2, Yıliçi Ödevi

Ödev İlan Zamanı: 25/04/2020 saat 10:00

Son Teslim Zamanı: 27/04/2020 saat 08:59

Toplam Soru Sayısı: 4

<u>Teslim Şekli:</u> Ödeve ilişkin cevaplarınızı, A4 kağıda el yazısı ile yazıp okunur şekilde taradıktan (tarayıcı veya telefon ile) sonra, **öğrenci numaranız olarak isimlendirdiğiniz, tek bir PDF dosyasını** <u>ytudonanim@gmail.com</u> adresine **20192_Bilgisayar_Donanimi_Odev** konu başlığı ile mail atınız.

<u>Ödevin Puanlanması:</u> Ödevden aldığınız puan yıliçi puanınız olarak değerlendirilecektir. Yıliçi puanının yılsonu puanına katkısı %30 olacaktır. Son teslim zamanından sonra gönderilen cevaplar değerlendirilmeyecektir. Cevaplarınız bireysel çalışmanız olmalıdır. Kopya cevaplar 0 puan olarak değerlendirilecektir.

Soru 1)

8 bitlik A ve B girişlerine karşılık 8 bitlik F çıkışlarını aşağıdaki fonksiyon tablosuna göre üreten ALU birimini aşağıda verilen kısıtlar çerçevesinde tasarlayınız.

Not 1: S_3 , S_2 , S_1 , S_0 ALU fonksiyon seçim uçlarıdır

Not 2: ALU: aritmetik lojik birim, AU: aritmetik birim, LU: lojik birim

ALU birimini, AU ve LU şeklinde 2 alt birim olarak tasarlayınız.

AU birimini 8 bitlik tam toplayıcı kullanarak tasarlayınız.

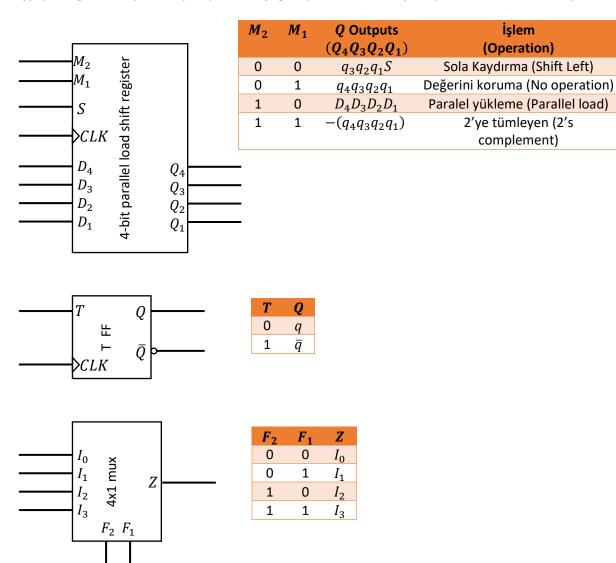
AU birimindeki tam toplayıcının C_{in} girişine, S_0 ALU fonksiyon seçim ucu doğrudan bağlı olarak verilmektedir, tasarımınızı buna göre yapınız.

AU biriminde A giriş lojiği ile B giriş lojiği için ve LU birimi tasarımında gerekli büyüklükte MUX kullanınız.

S ₃	S ₂	S ₁	S ₀	F	Açıklama
0	0	0	0	A+B	Addition
0	0	0	1	Α	A transfer
0	0	1	0	В	B transfer
0	0	1	1	A+1	Increment A
0	1	0	0	A-1	Decrement A
0	1	0	1	B+1	IncrementB
0	1	1	0	B-1	Decrement B
0	1	1	1	A-B	Subtraction
1	0	0	0	A AND B	Bitwise and
1	0	0	1	A OR B	Bitwise or
1	0	1	0	Ā	Complement A
1	0	1	1	A XOR B	Bitwise xor

Soru 2)

Blok yapısı ve fonksiyon tablosu aşağıda verilmiş olan 4 bitlik paralel yüklemeli ötelemeli yazmacın içyapısını gerekli sayıda T flip floplar, 4x1 çoğullayıcılar ve basit lojik kapılar kullanarak tasarlayınız.



Soru 3)

Verilen A ve B binary sayıları için **CMP** ve **TEST** komutlarının yürütülmesiyle oluşan sonucu ve bayrak durumlarını (**0 veya 1**) yazarak, verilen koşullu dallanma komutlarının hangi bayrak koşuluna baktığını ve komutun ürettiği sonucu **TRUE** veya **FALSE** olarak yazınız.

A = (111	1 1111)2		B = (1000 0000) ₂					
CMP	РΑ, В		İşlem Sonucu:					
Z	V		N		С			
Koşullu Dallanma Kon	nutu	Bayrak Koşulu			Komut Sonucu (TRUE/FALSE)			
JN								
JV								
JC								
JZ								
JA								
JAE								
JBE								
JL								
JGE								
JLE								

A = (1111 0000) ₂				B = (0000 1111) ₂				
TEST A, B				İşlem Sonucu:				
Z		V		N		С		

Soru 4)

Komut tablosu, datapath yapısı, komut çözümleme devresi ile komut formatı verilen tek-çevrim hardwired CPU için kontrol kelimesi alanlarının görevlerini/açıklamalarını yazarak aşağıda verilen komutlara karşılık komut çözümleme devresinin üreteceği binary değerleri tabloya yazınız.

Not: Verilen komutlar açısından "don't care" olan alan bitlerini, komut çözümleme devresinin ürettiği değere bakmaksızın X ile işaretleyiniz

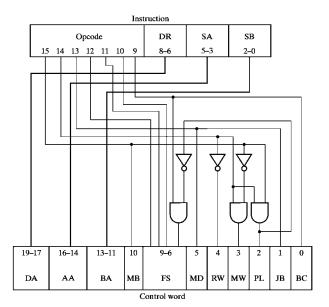
Kontrol Kelimesi Alanları	Görevi/Açıklaması
DA	
AA	
ВА	
МВ	
FS	
MD	
RW	
MW	
PL	
JB	
ВС	

Komut	DA	AA	BA	МВ	FS	MD	RW	MW	PL	JB	ВС
ADD R0, R7, R6											
DEC R1, R0											
NOT R3, R3											
SHR R4, R2											
LD RO, R7											
XOR R4, R3, R5											
ADI R3, R2, 3											
BRN R2, 10											
JMP RO											

Instruction Specifications for the Simple Computer

Instruction	Opcode	Mne- monic	Format	Description	Status Bits
Move A	0000000	MOVA	RD, RA	$R[DR] \leftarrow R[SA]^*$	N, Z
Increment	0000001	INC	RD, RA	$R[DR] \leftarrow R[SA] + 1*$	N, Z
Add	0000010	ADD	RD, RA, RB	$R[DR] \leftarrow R[SA] + R[SB]^*$	N, Z
Subtract	0000101	SUB	RD, RA, RB	$R[DR] \leftarrow R[SA] - R[SB]*$	N, Z
Decrement	0000110	DEC	RD, RA	$R[DR] \leftarrow R[SA] - 1*$	N, Z
AND	0001000	AND	RD, RA, RB	$R[DR] \leftarrow R[SA] \wedge R[SB]^*$	N, Z
OR	0001001	OR	RD, RA, RB	$R[DR] \leftarrow R[SA] \vee R[SB]^*$	N, Z
Exclusive OR	0001010	XOR	RD, RA, RB	$R[DR] \leftarrow R[SA] \oplus R[SB]^*$	N, Z
NOT	0001011	NOT	RD, RA	$R[DR] \leftarrow \overline{R[SA]}^*$	N, Z
Move B	0001100	MOVB	RD, RB	$R[DR] \leftarrow R[SB]^*$	
Shift Right	0001101	SHR	RD, RB	$R[DR] \leftarrow sr R[SB]^*$	
Shift Left	0001110	SHL	RD, RB	$R[DR] \leftarrow sl R[SB]*$	
Load Immediate	1001100	LDI	RD, OP	$R[DR] \leftarrow zf OP^*$	
Add Immediate	1000010	ADI	RD, RA, OP	$R[DR] \leftarrow R[SA] + zf OP^*$	N, Z
Load	0010000	LD	RD, RA	$R[DR] \leftarrow M[SA]^*$	
Store	0100000	ST	RA, RB	$M[SA] \leftarrow R[SB]^*$	
Branch on Zero	1100000	BRZ	RA,AD	if $(R[SA] = 0) PC \leftarrow PC + se AD$ if $(R[SA] \neq 0) PC \leftarrow PC + 1$, N, Z
Branch on Negative	1100001	BRN	RA, AD	if $(R[SA] < 0) PC \leftarrow PC + se AD$ if $(R[SA] \ge 0) PC \leftarrow PC + 1$, N, Z
Jump	1110000	JMP	RA	$PC \leftarrow R[SA]^*$	

^{*} For all of these instructions, PC ← PC + 1 is also executed to prepare for the next cycle.



(c) Jump and branch

