

Ödevi Sisteme Yükleme için Son Tarih ve Saat = 11/12/2024 Çarşamba 23:00

Geç gönderimler ve ilgili kurallara uymayan gönderimler KESİNLİKLE kabul edilmeyecektir! Tüm kuralları dikkatli bir şekilde okuyunuz.

Diğer Kurallar:

- Her öğrenci ödevi kendisi yapmalıdır. Birlikte yapılan ödevler veya internetten alınan hazır kodlar (dosya okuma) **kopya** olarak değerlendirilecektir. Kontrol sonucu **kopya** tespit edilen ödevlere ödev notu olarak 0 verilecektir. Dersin öğretim elemanları kopya ödev gönderen öğrenciler hakkında **disiplin sürecini** başlatma hakkını saklı tutar.
- Dersin öğretim elemanları ödevi gönderen öğrencileri çağırıp (veya zoom üzerinden) çözümleri hakkında soru sorma ve anlattırma hakkını saklı tutar.
- Ödevler, UBYS (**Eders Değil!**) BSM301 Bilgisayar Mimarisi ve Organizasyonu sayfasındaki Ödevler sekmesinden sisteme yüklenmelidir. Mail yolu ile gönderilen ödevler kabul edilmeyecektir ve değerlendirmeye alınmaz.
- Uygulamanızın çalışıp çalışmadığı ve uygulamanızdaki eksik kısımlar vb. hakkında bilgi veren 1-2 satırlık kısa bir açıklama içeren .txt uzantılı metin dosyası (**açıklama.txt**) oluşturup, bunu proje klasörü ile birlikte sisteme yüklemeniz gerekmektedir.
- Proje klasörünü (açıklama içeren “açıklama.txt” dosyasıyla birlikte) sıkıştırıp (*.zip) uzantılı tek bir dosya sisteme yükleyiniz. ((*.rar) uzantılı dosyaların sisteme yüklenmesinde sıkıntı yaşanmaktadır ve ödeviniz değerlendirmeye alınmaz!)
- Proje klasörü yerine, sisteme tek bir class dosyası, tek bir java dosyası vb. farklı şekillerde yükleme yapanların ödevleri değerlendirilmeyecek ve 0 ile notlandırılacaktır.
- Projenizi “_ÖğrenciNo_Ad_Soyad” (örn. _2013510001_Ali_Bilir) şeklinde isimlendiriniz. Birden fazla ismi olan öğrenciler sadece ilk ismini yazmalıdır. Ayrıca, projede tanımladığınız her bir sınıfın ismi de öğrenci numaranız ile başlamalıdır (örn. _2013510001_RAM gibi).
- Kodunuza KESİNLİKLE **yorum satırları eklemeyiniz**.
- Son teslim tarihi ve saatine kadar ödevini sisteme yüklemeyenlerin ödevleri değerlendirmeye alınmayacaktır ve 0 ile notlandırılacaktır.

Başarılar.

Ödev Açıklaması:

Bu ödevde, Eclipse (Netbeans DEĞİL!) ve Java dilini kullanarak, Temel Bilgisayar’ın Kontrol (Denetim) Ünitesinin kontrol mantık kapıları ve bazı bileşenler için bir simülatör tasarlamanız beklenmektedir. Simülatör **dinamik** bir **konsol uygulaması** şeklinde olacaktır ve aşağıdaki özelliklere sahip olmalıdır:

- Sistemdeki tüm bileşenler (AR, PC, DR ve IR Register’ları, S ve IEN bayrakları, kontrol mantık kapıları, kontrol sinyalleri) nesneye yönelik programlama tekniğine uygun şekilde tasarlanmalıdır. S bayrağının D tipi IEN bayrağının ise JK tipi flip-flop olduğunu varsayınız. Kontrol mantık kapılarını oluştururken VE, VEYA, DEĞİL kapılarını kullanınız. Sınıflarınızı tasarlarken register’ların ve bayrakların iç yapılarını (lojik kapılar, flip-floplar vb.) dikkate almayınız. (Projenizde belirli sayıda ve uygun şekilde sınıflar ve metotlar tanımlanmalıdır.)
- Uygulama çalıştığında ilk olarak denetim_fonksiyonlari.txt isimli dosyayı okumalı ve dosyadaki her bir denetim fonksiyonuna karşılık gelen mikro işlemin veya mikro işlemlerin çalışması için hangi bileşenlerin hangi kontrol girişlerinin (Load, Clear vb.) aktif edildiğini ekrana yazmalıdır. denetim_fonksiyonlari.txt dosyasının her bir satırında farklı bir denetim fonksiyonu yer almaktadır. Örnek bir denetim_fonksiyonlari.txt dosyasının ekran görüntüsü aşağıda verilmiştir. Ödevle birlikte örnek bir dosya sisteme yüklenecektir. Kendinizin de formata uygun farklı dosyalar oluşturup uygulamanızı test etmenizde yarar vardır. Ayrıca, denetim_fonksiyonlari.txt dosyasındaki denetim fonksiyonlarının sayısı değişebilir.
- denetim_fonksiyonlari.txt dosyasının her bir satırında birbirinden boşluk karakteriyle ayrılmış ve denetim fonksiyonunu oluşturan sinyaller (R, T0-T15, D0-D7, I ve B0-B11) bulunmaktadır. Bu sinyallerden bazıları normal

formda veya tümleyen formda bulunabilir. (Temel bilgisayarın denetim fonksiyonu ve mikro işlemler tablosunda bulunan if'li kısımlar ve bunlara bağlı olan mikro işlemler göz ardı edilecektir. Örneğin, "D7 I' T3 B1" denetim fonksiyonunun karşısında if'li ifade olduğu için o satır komple göz ardı edilecektir ve denetim_fonksiyonlari.txt dosyasında "D7 I' T3 B1" denetim fonksiyonu yer almaz.)

- Örnek bir ekran görüntüsü aşağıda verilmiştir. Sonuçlar, örnek ekran görüntüsündeki formata uygun şekilde ekrana yazılmalıdır.

Örnek bir ekran görüntüsü aşağıda verilmiştir:

denetim_fonksiyonlari.txt dosyası okundu.

Denetim fonksiyonu: D0 T4

DR register'ının Load girişi aktif edilir.

Denetim fonksiyonu: R' T1

IR register'ının Load girişi aktif edilir. PC register'ının Increment girişi aktif edilir.

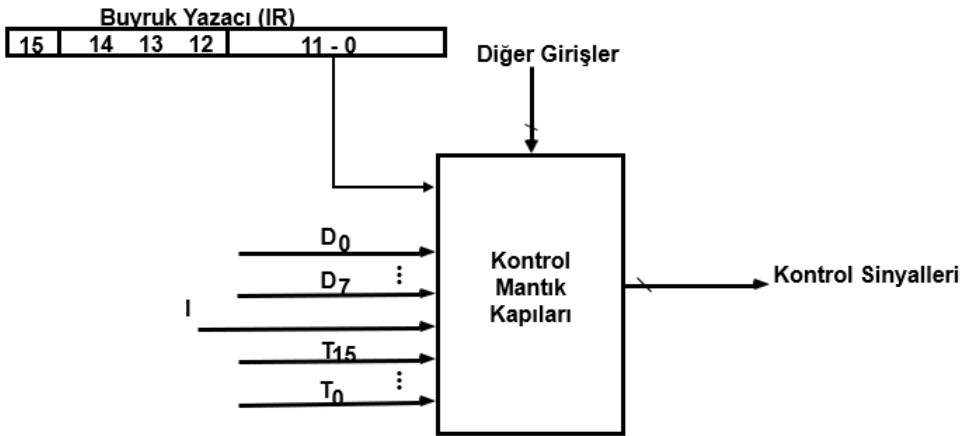
Denetim fonksiyonu: D5 T4

AR register'ının Increment girişi aktif edilir.

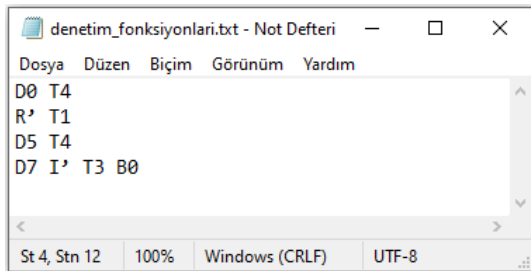
Denetim fonksiyonu: D7 I' T3 B0

S bayrağının D girişine 0 uygulanır.

Temel Bilgisayar kontrol (denetim) ünitesinin kontrol mantık kapılarına ilişkin görsel aşağıda verilmiştir:



Örnek bir denetim_fonksiyonlari.txt dosyasının ekran görüntüsü aşağıda verilmiştir:



Temel Bilgisayar için Denetim Fonksiyonu ve Mikro işlemler Tablosu aşağıda verilmiştir:

Fetch	R'T ₀ :	AR ← PC	Register-Reference	D ₇ 'T ₃ = r	(Common to all register-reference instr) (i = 0,1,2, ..., 11)
	R'T ₁ :	IR ← M[AR], PC ← PC + 1		IR(i) = B _i	
	R'T ₂ :	D ₀ , ..., D ₇ ← Decode IR(12 ~ 14), AR ← IR(0 ~ 11), I ← IR(15)		r:	
Decode			CLA	rB ₁₁ :	AC ← 0
			CLE	rB ₁₀ :	E ← 0
Indirect Interrupt	D ₇ 'T ₃ :	AR ← M[AR]	CMA	rB ₉ :	AC ← AC'
			CME	rB ₈ :	E ← E'
	T ₀ 'T ₁ 'T ₂ '(IEN)(FGI + FGO):	R ← 1	CIR	rB ₇ :	AC ← shr AC, AC(15) ← E, E ← AC(0)
Memory-Reference	RT ₀ :	AR ← 0, TR ← PC	CIL	rB ₆ :	AC ← shl AC, AC(0) ← E, E ← AC(15)
	RT ₁ :	M[AR] ← TR, PC ← 0	INC	rB ₅ :	AC ← AC + 1
	RT ₂ :	PC ← PC + 1, IEN ← 0, R ← 0, SC ← 0	SPA	rB ₄ :	If(AC(15)=0) then (PC ← PC + 1)
	D ₀ T ₄ :	DR ← M[AR]	SNA	rB ₃ :	If(AC(15)=1) then (PC ← PC + 1)
	D ₀ T ₅ :	AC ← AC ∧ DR, SC ← 0	SZA	rB ₂ :	If(AC = 0) then (PC ← PC + 1)
	D ₁ T ₄ :	DR ← M[AR]	SZE	rB ₁ :	If(E=0) then (PC ← PC + 1)
	D ₁ T ₅ :	AC ← AC + DR, E ← C _{out} , SC ← 0	HLT	rB ₀ :	S ← 0
	D ₂ T ₄ :	DR ← M[AR]	Input-Output	D ₇ 'T ₃ = p	(Common to all input-output instructions) (i = 6,7,8,9,10,11)
	D ₂ T ₅ :	AC ← DR, SC ← 0		IR(i) = B _i	
	D ₃ T ₄ :	M[AR] ← AC, SC ← 0		p:	
	D ₃ T ₅ :	PC ← AR, SC ← 0		pB ₁₁ :	
	D ₄ T ₄ :	PC ← AR, SC ← 0		pB ₁₀ :	
	D ₄ T ₅ :	M[AR] ← PC, AR ← AR + 1		pB ₉ :	
	D ₅ T ₄ :	PC ← AR, SC ← 0		pB ₈ :	
	D ₅ T ₅ :	DR ← M[AR]		pB ₇ :	
	D ₆ T ₄ :	DR ← DR + 1		pB ₆ :	
	D ₆ T ₅ :	M[AR] ← DR, if(DR=0) then (PC ← PC + 1), SC ← 0			
AND	D ₀ T ₄ :	DR ← M[AR]	INP		
ADD	D ₀ T ₅ :	AC ← AC ∧ DR, SC ← 0	OUT		
LDA	D ₁ T ₄ :	DR ← M[AR]	SKI		
STA	D ₁ T ₅ :	AC ← AC + DR, E ← C _{out} , SC ← 0	SKO		
BUN	D ₂ T ₄ :	DR ← M[AR]	ION		
BSA	D ₂ T ₅ :	AC ← DR, SC ← 0	IOF		
ISZ	D ₃ T ₄ :	M[AR] ← AC, SC ← 0			
	D ₃ T ₅ :	PC ← AR, SC ← 0			
	D ₄ T ₄ :	PC ← AR, SC ← 0			
	D ₄ T ₅ :	M[AR] ← PC, AR ← AR + 1			
	D ₅ T ₄ :	PC ← AR, SC ← 0			
	D ₅ T ₅ :	DR ← M[AR]			
	D ₆ T ₄ :	DR ← DR + 1			
	D ₆ T ₅ :	M[AR] ← DR, if(DR=0) then (PC ← PC + 1), SC ← 0			