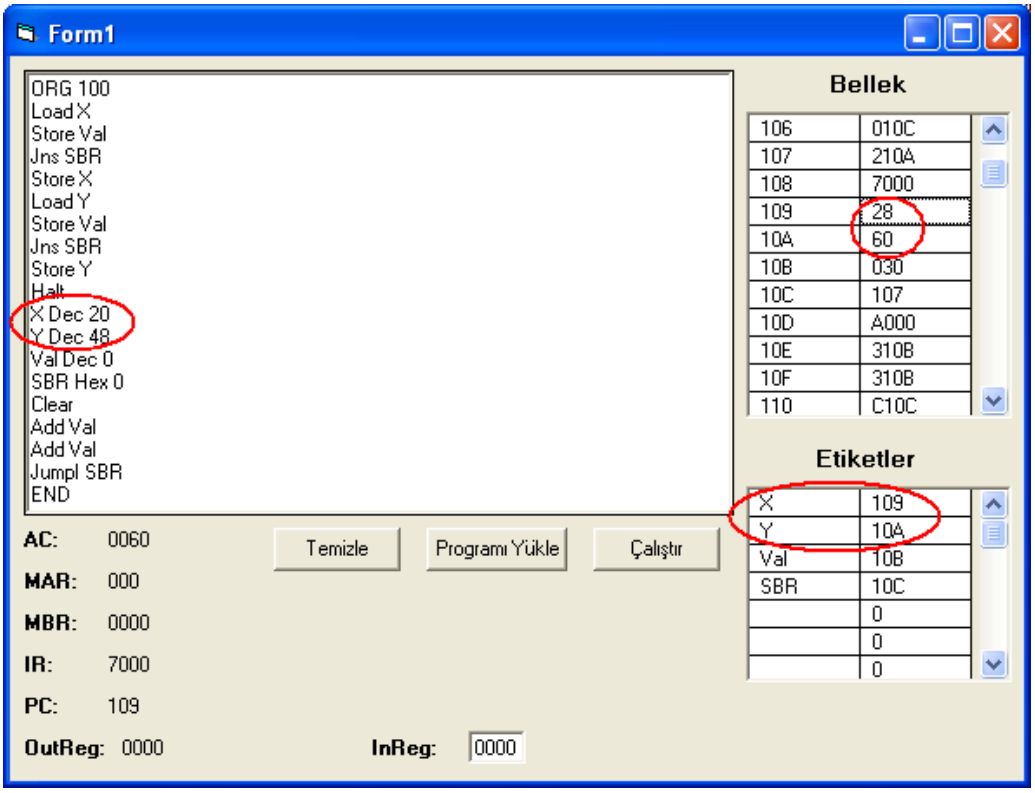
29 Nisan 2014

BİLGİSAYAR ORGANİZASYONU VE TASARIMI DERSİ PROJE ÖDEVİ

Dersin projesi kapsamında, ekte detaylı yapısı ve komut seti verilen MARIE bilgisayarı için yazılan programları çalıştıran bir simülatör geliştirmeniz istenmektedir. Simülatörün geliştirilmesinde kullanılacak programlama dili açısından herhangi bir kısıtlama yoktur. İstediğiniz dili kullanabilirsiniz.

Aşağıdaki resimde bu görev için hazırlanmış bir simülatör ve örnek çıktısı görülmektedir. Örnekte X ve Y etiketli bellek gözlerindeki sayıları kendileri ile toplayıp yeniden üzerlerine yazan bir MARIE programı görülmektedir. Assembly programı ekrana yazıldıktan sonra öncelikle "Programı Yükle" butonu vasıtasıyla hexadecimal karşılığına çevrilmeli; etiketlerin, komutların ve değişkenlerin adresleri hesaplanarak bellek üzerindeki yerleşimi ve değeri gösterilmelidir. Sonrasında "Çalıştır" butonu vasıtasıyla bellekteki hexadecimal formattaki kod çalıştırılmalı ve işlem bittikten sonraki yeni bellek yapısı ve saklayıcı içerikleri ekranda gösterilmelidir.



Hazırlayacağınız program GUI ya da konsol tabanlı olabilir. Ancak her iki şekilde de yazılan assembly programı icra edilmeden önceki ve edildikten sonraki bellek içeriği ve saklayıcı değerleri raporlanabilmelidir. Bu nedenle konsol uygulaması geliştireceklerin sonuçları ayrı bir dosyaya yazdırmaları tavsiye edilir.

Dikkat edilmesi gereken noktalar:

1- Bellekteki tüm sayılar hexadecimal formattadır. Aritmetik işlemler sırasında isterseniz kolaylık sağlaması açısından bu değerleri onluk tabana çevirip kullanabilirsiniz. Ancak sonucu bellek veya saklayıcılara yine hexadecimal formatta yazmanız gerekmektedir.

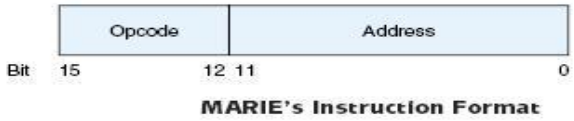
2- Simülatörünüz ORG, END, DEC, HEX gibi pseudo komutları tanımak zorundadır. Kullanıcı, yazdığı programın bellekte hangi konumdan itibaren saklanacağını ORG ile seçebilmelidir.

3- Simülatörünüz ile bir MARIE kodunu işlerken iki geçiş yapmanız gerekmektedir. İlk geçişte tanımlanan etiketleri ve her satıra karşılık gelen fiziksel adresi hesaplamanız gerekmektedir (ORG komutuna göre). İlk geçişte bulduğunuz etiketlerin değerlerini ayrı bir tablo ya da dosyada raporlamanız gerekmektedir. İkinci geçişte önceden hesaplamış olduğunuz etiket adreslerine göre hexadecimal komutları oluşturmanız gerekmektedir.

4- Saklayıcıların içerikleri raporlanırken bit sayılarına uygun notasyonda gösterilmelerine dikkat edilmelidir. AC, 16 bit ise mutlaka 4 haneli bir hexadecimal sayı ile ifade edilmelidir. Daha az hane ile gösterilemez.

MARIE İÇYAPISI

Komut formatı:



MARIE's Instruction Format

Komut icrası basamakları (sırasıyla Fetch, Decode ve Get Operand):

FETCH:	MAR ← PC	**DECODE:**	MAR ← IR[11-0]
	IR ← M[MAR]		(Decode IR[15-12)]
	PC ← PC +1	**GET OPERAND:**	MBR ← M[MAR]

MARIE Komut Seti

Opcode	Instruction	RTN
0000	JnS *X*	MBR ← PC MAR ← *X* M[MAR] ← MBR MBR ← *X* AC ← 1 AC ← AC + MBR PC ← AC
0001	Load *X*	MAR ← *X* MBR ← M[MAR], AC ← MBR
0010	Store *X*	MAR ← *X*, MBR ← AC M[MAR] ← MBR
0011	Add *X*	MAR ← *X* MBR ← M[MAR] AC ← AC + MBR
0100	Subt *X*	MAR ← *X* MBR ← M[MAR] AC ← AC - MBR
0101	Input	AC ← InREG
0110	Output	OutREG ← AC
0111	Halt	

1000	Skipcond	If IR[11–10] = 00 then
		  If AC < 0 then PC ← PC + 1
		Else If IR[11–10] = 01 then
		  If AC = 0 then PC ← PC + 1
		Else If IR[11–10] = 10 then
		  If AC > 0 then PC ← PC + 1

1001	Jump *X*	PC ← IR[11–0]
1010	Clear	AC ← 0
1011	AddI *X*	MAR ← X MBR ← M[MAR] MAR ← MBR MBR ← M[MAR] AC ← AC + MBR
1100	JumpI *X*	MAR ← *X* MBR ← M[MAR] PC ← MBR

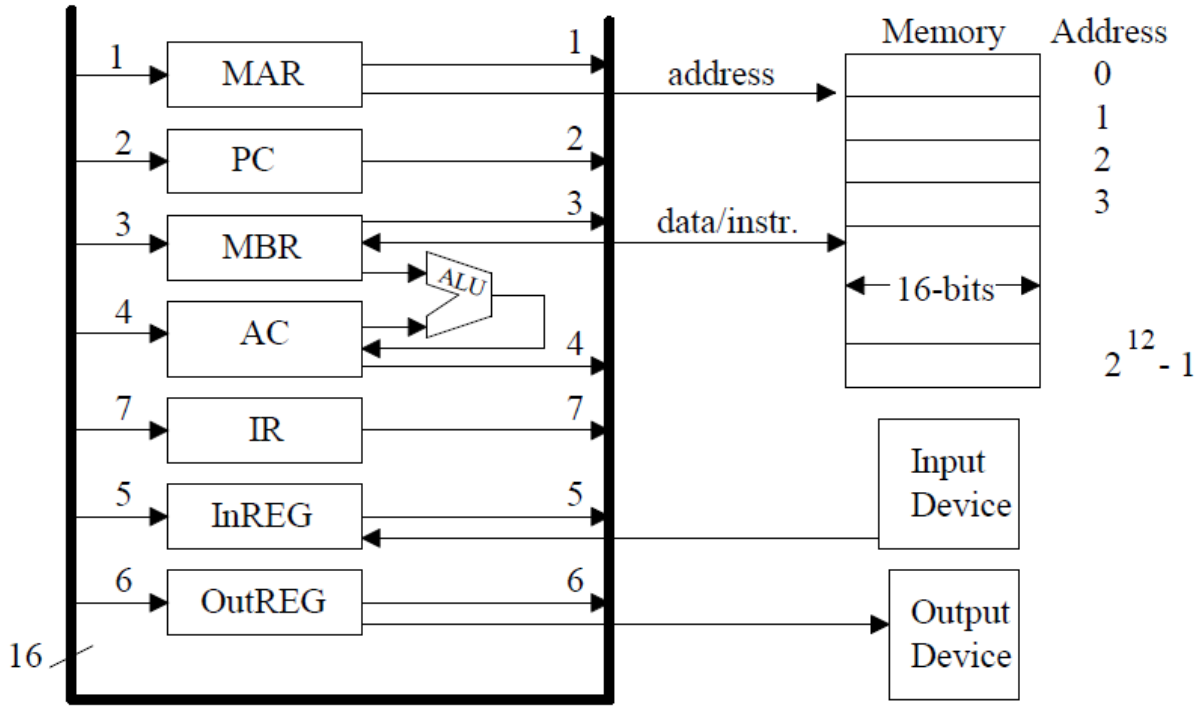
Skipcond X : X'in durumuna göre sıradaki komutu atla

X = $(000)_{16}$ iken eğer AC < 0 ise sıradaki komutu atla $(b_{11}\ b_{10}) = (00)_2$

X = $(400)_{16}$ iken eğer AC = 0 ise sıradaki komutu atla $(b_{11}\ b_{10}) = (01)_2$

X = $(800)_{16}$ iken eğer AC > 0 ise sıradaki komutu atla $(b_{11}\ b_{10}) = (10)_2$

MARIE VERİ YOLU



ÖRNEK MARIE PROGRAMI

Aşağıdaki program SONUC = X + Y – Z işlemini gerçekleştirmektedir.

Adres	**Etiket**	**Assembly İfadesi**	**Makine Dili**
0		LOAD X	1006_{16}
1		ADD Y	3007_{16}
2		SUBT Z	4008_{16}
3		STORE SONUC	2009_{16}
4		OUTPUT	6000_{16}
5		HALT	7000_{16}
6	X,	DEC 10	$000A_{16}$
7	Y,	DEC 20	0014_{16}
8	Z,	DEC 5	0005_{16}
9	SONUC,	DEC 0	0000_{16}

Simülatörünüzün test aşamasında üç farklı MARIE programı hazırlayıp yazdığınız program ile çalıştırmanız beklenmektedir. Bu programlar şu şekilde olmalıdır:

i) Bellekteki $(210)_{16}$ - $(22F)_{16}$ alanları arasındaki sayıların her birini 8 ile çarpıp üzerine yazdıran MARIE programı

ii) Fibonacci serisinin ilk 50 elemanını $(4FF)_{16}$ adresinden başlayan bellek gözlerine kaydeden MARIE programı

iii) Bellekteki $(350)_{16}$ - $(36F)_{16}$ alanları arasındaki sayıların kaçının pozitif, kaçının negatif ve kaçının sıfır olduğunu bularak bu bilgileri istediğiniz ardışık 3 bellek gözüne yazan MARIE programı

i. ve iii. programda kullanacağınız sayıları negatif ve pozitif sayılar kümesinden rasgele üretebilirsiniz.

Uyulması gereken kurallar ve hatırlatmalar:

- Proje en fazla 4'er kişilik gruplar halinde hazırlanabilir.
- Projelerinizi bir .rar sıkıştırılmış dosyası (dosya içeriğinde kaynak kod + rapor olmalı) şeklinde ödev sistemine yüklemeniz gerekmektedir. Rapor için yazılı çıktı istenmemektedir.
- Söz konusu simülatörün benzeri internette bulunmaktadır. Bunun çalışmasını inceleyebilirsiniz ancak kesinlikle kaynak kodundan kopya alamazsınız.
- Örgün ve ikinci öğretim öğrencilerinden oluşan bir proje grubu olamaz. Her bölüm öğrencisi sadece kendi bölümünden öğrencilerle proje grubu oluşturabilir.
- Ders etkinliklerinin ağırlık dağılımı; Vize: %25, Proje: %25, Final: %50 şeklinde düzenlenmiştir. Vize sonuçlarınız ve kopya ödev durumunda proje grubunun/gruplarının tüm üyelerinin 0 ile cezalandırılacağı düşünüldüğünde kopya ödev teslim etmektense eksik ama tamamıyla özgün bir projeyi teslim etmenin daha iyi olacağı hatırlatılır.
- Ödev sistemine proje grubundan sadece bir öğrencinin proje yüklemesi yeterlidir (Raporunuza proje grubundaki tüm öğrencilerin isim ve numaralarını yazdığınız sürece sorun yoktur). Çoklu yükleme yapmayın.
- **Son teslim tarihi: 16 Mayıs 2014 Cuma**