



Fenerbahçe Üniversitesi

BLM 101 – Bilgisayar Mühendisliğine Giriş

LAB 6: Von Neumann Mimarisi Fonksiyonlar

Önemli Not: Aşamaları tamamladıkça, dersin hocası veya asistanı yanınıza çağırarak, tamamladığınız aşamayı gösterdikten sonra diğer aşamaya geçiniz.

LAB Hakkında:

Von Neumann modeli ile geliştirilmiş bir işlemci üzerinde fonksiyon mekanizmalarının yazılım tasarımları yapılacaktır.

Von Neumann Simulatörü adresi: <http://levent.tc/araclar/vonneumann/>

Komut seti (ISA – Instruction Set Architecture) aşağıdaki tabloda verilmektedir.

| Komut Adı | Görevi | Örnek Kullanım |
|-----------|----------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------|
| LOD ADDR | Yükleme (Load), Bellekteki verilen adresin içerisinde değeri alıp, ACC saklayıcısına yerleştirir. $ACC = *(ADDR)$ | LOD X LOD Y |
| STO ADDR | Kaydetme (Store), ACC'nin içerisindeki değeri alıp, bellekte verilen adrese yazar. $*(ADDR) = ACC$ | STO X STO Y |
| ADD ADDR | Bellekteki verilen adresteki değeri alır, ACC ile toplayıp, ACC'nin üzerine yazar. $ACC = ACC + *(ADDR)$ | ADD X ADD Y |
| SUB ADDR | Bellekteki verilen adresteki değeri alır, ACC ile çıkartıp, ACC'nin üzerine yazar. $ACC = ACC - *(ADDR)$ | SUB X SUB Y |
| MUL ADDR | Bellekteki verilen adresteki değeri alır, ACC ile çarpıp, ACC'nin üzerine yazar. $ACC = ACC * *(ADDR)$ | MUL X MUL Y |
| DIV ADDR | Bellekteki verilen adresteki değeri alır, ACC ile bölüp, ACC'nin üzerine yazar. $ACC = ACC / *(ADDR)$ | DIV X DIV Y |
| JMZ SAYI | ACC'in değeri 0 ise, verilen sayı değerini PC'e atar, değilse işlem yapmaz. | JMZ 5 // Yani ACC == 0 ise, program 5 nolu satıra atlar |
| JMP SAYI | PC = Sayı olur. | JMP 5 // Program doğrudan 5. satırdan devam eder |

| | | |
|-----|--------------------------------------|-----|
| NOP | No Operation, hiçbir işlem yapılmaz. | NOP |
| HLT | Uygulama durur | HLT |

LAB'ın aşamaları ve puanlar:

Algoritmanızı kurarken, bu işlemcinin çalıştırabileceği komut setine göre kurmalısınız. Komutları düşünerek, algoritmanızı nasıl ifade edebileceğinizi bulmalısınız.

1- Sayının 5 fazlasını bulan fonksiyon (0 Puan):

X değişkenine 1 sayı yazınız. Sayının 5 fazlasını bulan bir fonksiyon yazınız. Fonksiyon 40. adresten başlayacak. Fonksiyon argüman olarak X'i almaktadır. Fonksiyon işini tamamladıktan sonra, geriye değeri Y. değişkenine yazıp, çağrıldığı yere dönmektedir. Fonksiyonu çağırın kod parçacığında Y. değişkenine yazılmış olan değeri Z değişkenine kopyalanıp, program sonlandırılmalıdır.

2- 2 sayının farkını bulan fonksiyon (50 Puan):

X ve Y değişkenlerine iki sayı yazınız. Bu sayılar karşılaştırma yapılacak 1. ve 2. sayıyı ifade etmektedir. Çıkartma işlemi yapan fonksiyonu yazınız. Bu fonksiyon 50. adresten başlamalıdır. Ana kodda iken 50 adrese atlayıp fonksiyon çalışıp tekrar çağrıldığı yere dönmelidir. Fonksiyona verilecek argümanlar X. ve Y. girişlerinden alınmalıdır. Fonksiyon 50. adreste çalıştığında, X ve Y deki değerlerinin sayıların farkını alıp Z. değişkenine yazıp, program çağrıldığı adrese geri atlamalıdır. Program çağrıldığı yere dönünce Z. değişkenindeki değeri alıp, W değişkenine yazıp, uygulamayı sonlandırmalıdır.

3- Faktöriyel Fonksiyonu (50 Puan):

X değişkenine bir sayı yazınız.

Yazılacak faktöriyel fonksiyonunun argüman girişi X değişkenidir. Fonksiyon ise 60. Adresten itibaren başlamaktadır.

Fonksiyon, Y değişkeninden geri değer döndürmelidir. Fonksiyon çalıştıktan sonra çağrıldığı yere geri dönüp, sonucun geri döndürüldüğü Y değişkeninden, veriyi kopyalayıp Z adrese yazmalıdır. Sonrasında uygulama kendini durdurmalıdır.