



## BİL362L- Mikroişlemciler Laboratuvarı Lab 1

Bu laboratuvar dersinde sizden çevirici dil kullanarak aşağıda vereceğiniz görevleri gerçekleştirmenizi beklemekteyiz. Bu laboratuvar boyunca alt seviye dil olan assembly dili ile STM32F4 kartımızda bulunan ARM Cortex M4 çekirdeğini programlayacağız.

- 1) ARM buyruklarını kullanarak, 0'dan 20'ye kadar olan tek sayıların toplamını hesaplayıp bir yazmaca yazacak bir program yazmak istiyoruz.

a) Bu programı herhangi bir atlama, dallanma (branch) buyruğu kullanmadan gerçekleştiriniz.

b) Bu programı atlama, dallanma (branch) buyruğu kullanarak gerçekleştiriniz. Buradaki hedef programı tek döngü ile gerçekleştirmek olmalıdır.

2) Fibonacci Dizisi, her sayının kendisinden bir önceki sayı ile toplanması ile elde edilen sayılar serisidir. Fibonacci Dizisinde yer alan rakamların özelliği, Fibonacci Dizisinde yer alan sayıların kendilerinden bir öncekiyle oranlandığında oluşan serinin altın orana yaklaşarak ilerlemesidir. Örnek olarak 0,1,1,2,3,5,8,13... şeklinde olabilir.

Bu bölümde Fibonacci dizisini bir yazmaç (register) üzerinde görmek istiyoruz.

- a) İlk elemanı 0 ve ikinci elemanı 1 olan bir fibonacci dizisinin 14. Elemanına kadar hesaplayan ve bunu döngü kullanarak yapan bir program yazınız.

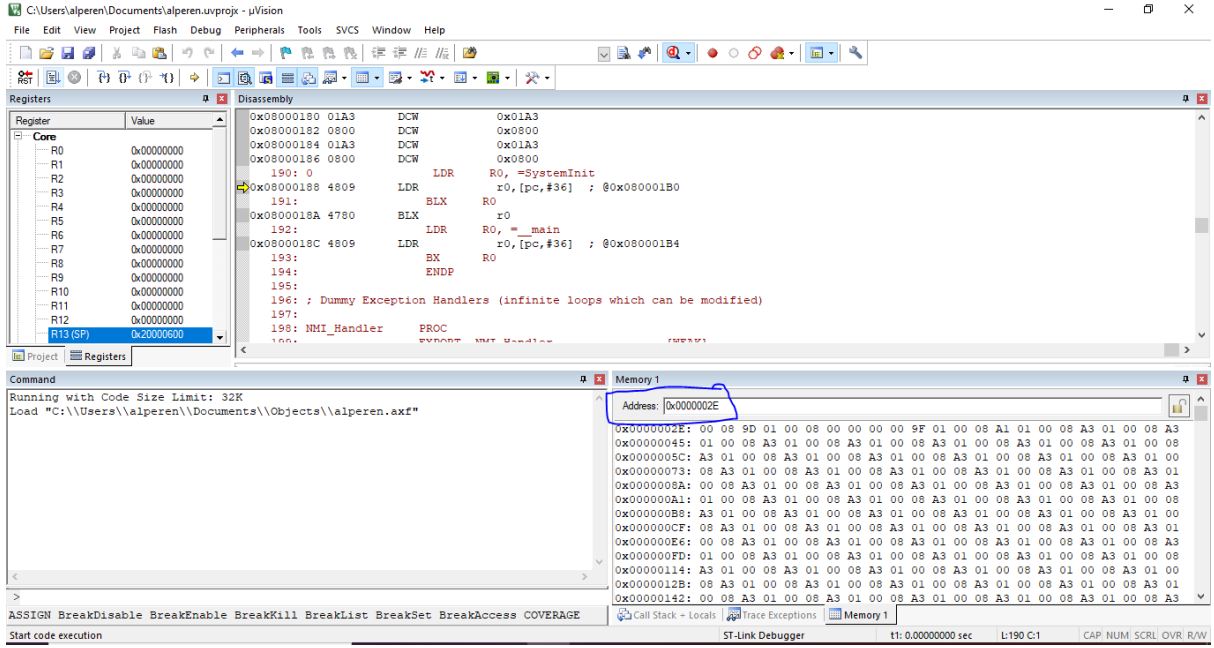
Bunun için dallanma (branch) buyrukları kullanmanız gerekecektir. Fibonacci dizisinin o anki değerini 'R0' yazmacında tutunuz. Her atlama öncesi (ya da yeni fibonacci sayısı oluşturulmadan önceki durumda) 'R0' yazmacındaki Fibonacci'nin dizisine ait elemanın, dizide kaçınıcı eleman olduğunu 'R1' yazmacında tutulmasını sağlayın. 14. Eleman R0 yazmacında tutulduktan ve 'R1' 14 değerini aldıktan sonra programınız döngüden çıkmalıdır.

NOT: 1. Ve 2. Elemanı döngüden önceki kısımda tanımlayıp döngünün ilk çalışmasında 3. Elemanın hesaplaması beklenmektedir.

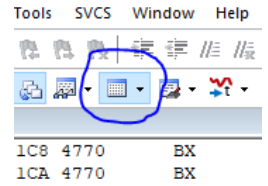
Örnek olarak döngülere göre yazmaçların o anki değerleri aşağıdaki gibi olmalıdır.

Döngü Durumu	R0	R1
1.Döngü	1	3
2.Döngü	2	4
3.Döngü	3	5
7.Döngü	21	9

Öncelikle sorunun ikinci kısmı için kısa bir bilgilendirme:



Programınızı yazdıktan sonra “Debug” kısmında kontrol ederken yanda işaretlenen kısımdaki butona tıklayarak “Memory” sekmesini oluşturabilir ve yukarıdaki şekilde belirtildiği gibi “Memory 1” sekmesinin içindeki “Address” kısmına yazacağınız parametre ile belleğin o adresindeki verileri görüntüleyebilirsiniz. Memory1 içerisinde gördüğünüz her ikili grup **1 Bayt** alan kaplamakta ve bellekte farklı bir adrese sahiptir. Örnek olarak görseldeki 0x0000002E adresinde ‘00’ değeri varken, 0x0000002F adresinde ‘08’ değeri bulunmaktadır.



Kullanacağımız STM32F4 kartında bulunan SRAM bölgesi **0x20000000** adresinden başlamaktadır. Kartta hangi adreslerin hangi görevler için ayrıldığını aşağıdaki PDF’ten ayrıntılı olarak inceleyebilirsiniz. Bu laboratuvarı gerçekleştirmek için PDF’i incelemeniz **gerekmemektedir**.

[https://www.st.com/content/ccc/resource/technical/document/reference\\_manual/3d/6d/5a/66/b4/99/40/d4/DM00031020.pdf/files/DM00031020.pdf/jcr:content/translations/en.DM00031020.pdf](https://www.st.com/content/ccc/resource/technical/document/reference_manual/3d/6d/5a/66/b4/99/40/d4/DM00031020.pdf/files/DM00031020.pdf/jcr:content/translations/en.DM00031020.pdf)

- b) a seçeneğinde yazdığınız ve her döngüde fibonnacci sayısının elemanını ve sırasını veren programı geliştirmenizi ve SRAM’ın ilk adresinden başlayarak döngünün her adımında o anki fibonnacci sayısını ve sırasını SRAM’ın ilk baytından itibaren yazmanızı istiyoruz. Her sayının bellekte 1 bayt yer tutması yeterlidir. Örnek olarak işlem bittiğinde SRAM’e ait ilk 8 Bayt şu şekilde gözükmelidir. Yazma işlemine 3. Elemandan başlamanız beklenmektedir. 0, 1, **1, 2, 3, 5, 8, 13, 21...**

Adres	Değer
0x20000000	0x01(Fibonacci sayısı)
0x20000001	0x03(Fibonacci dizisindeki sırası)
0x20000002	0x02(Fibonacci sayısı)
0x20000003	0x04(Fibonacci dizisindeki sırası)
0x20000004	0x03(Fibonacci sayısı)
0x20000005	0x05(Fibonacci dizisindeki sırası)
0x20000006	0x05(Fibonacci sayısı)
0x20000007	0x06 (Fibonacci dizisindeki sırası)

