



**TÜBİTAK-2209-A ÜNİVERSİTE ÖĞRENCİLERİ ARAŞTIRMA  
PROJELERİ DESTEĞİ PROGRAMI**

**ARAŞTIRMA ÖNERİSİ FORMU**

2022

2. Dönem Başvurusu

**2209/A ÜNİVERSİTE ÖĞRENCİLERİ ARAŞTIRMA PROJELERİ DESTEĞİ PROGRAMI**  
**ARAŞTIRMA ÖNERİSİ FORMU**

**A. GENEL BİLGİLER**

<b>Başvuru Sahibinin Adı Soyadı:</b> Musa Akyüz
<b>Araştırma Önerisinin Başlığı:</b> Basit Elektrik-Elektronik Elemanları Kullanarak 8 Bitlik Bilgisayar Mimarisi ve Komut Seti Tasarımı
<b>Danışmanın Adı Soyadı:</b> Dr. Öğr. Üyesi Ümit ŞENTÜRK
<b>Araştırmanın Yürütüleceği Kurum/Kuruluş:</b> Bolu Abant İzzet Baysal Üniversitesi Mühendislik Fakültesi

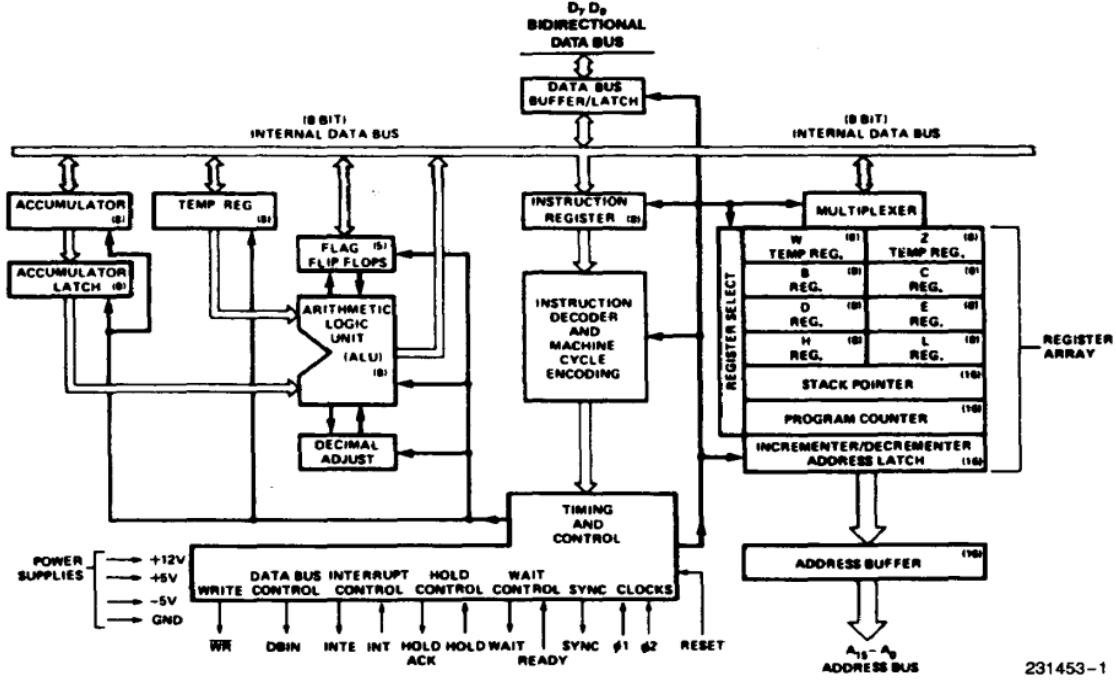
**ÖZET**

Ve, veya, değil, veya değil gibi mantık kapıları kullanarak basite indirgenmiş şekilde, toplama, çıkarma, büyüktür, küçüktür, eşittir gibi temel işlemleri yapan elektronik devrelerin tasarımını yapmayı ve gerçekleştirme döngüsünü içerir. Bu elektronik devreleri kaydediciler ve bellek yapıları ile destekleyerek arka arkaya işlemler yapabilen hale getirilir. Kodlama sistemi geliştirilerek, bu sistemin derleyicisini, tasarlanacak bilgisayarın mimarisine uygun halde yazmak araştırma önerisinin diğer konusudur. Tasarlanacak sistemi genişletilebilir ve ucu açık halde bırakmak ve tüm alt parçacıkları paralel giriş-çıkışa izin veren entegre devrelerden seçmek ve paralelliği korumak benimsenen yaklaşımlardandır.

**Anahtar Kelimeler:** 8 bitlik bilgisayar mimarisi, komut seti tasarımı, derleyici tasarımı, aritmetik mantık birimi, kontrol ünitesi tasarımı

**1. ÖZGÜN DEĞER**

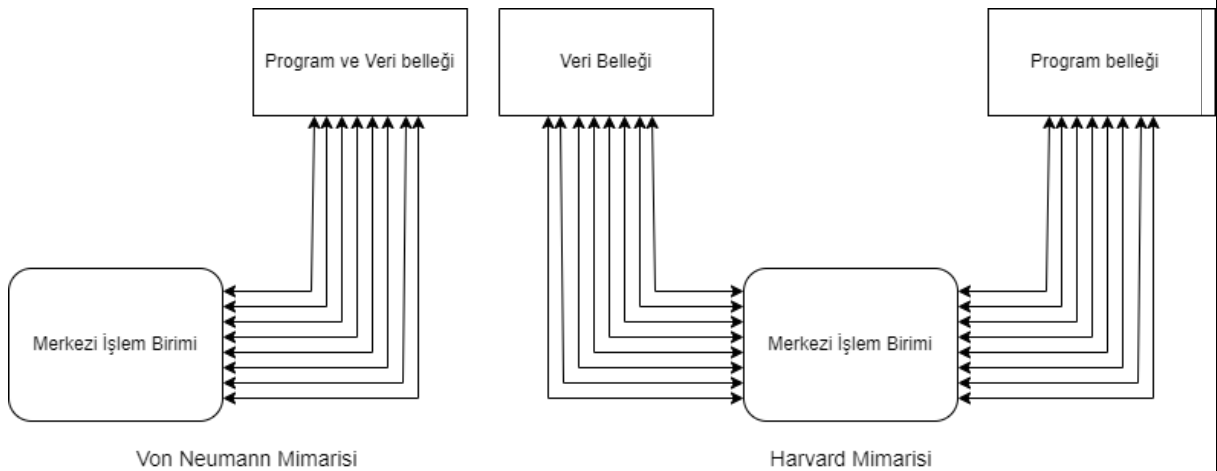
Hesap yapan makinelerin tasarlanmasının başından bu yana çok şey çok hızlı şekilde gelişmiştir ve gelişmeye devam etmiştir. İşlemciler ve mikroişlemciler her geçen gün hızlanmıştır. Hızlanmanın yanı sıra teknoloji de küçülmüştür. Transistör boyutları nanometreler seviyelerine gelmiştir. Gordon Moore bu hızlanma ve küçülmenin oranını 1965 yılında öngörmüştür [1]. Gordon Moore'un hesaplamalarına göre belirtilen küçülmenin her 3 yılda bir 0.7 kat oranında olacağı belirlenmiştir [2].



INTEL firması 1971-1975 yılları arasında ilk geniş amaçlı işlemcisi 8080'i piyasaya sürmüştür (Şekil 1). İlk işlemci diyebileceğimiz 8 bit veri yolu olan bu işlemci, günümüz işlemcileri ve işlem kapasitelerinin yanında çok yavaş kalmaktadır. Veri yolları katlanarak artmıştır. Veri yollarının genişlemesi, daha çok işlem yapan sistemin bir arada bulunması gibi ihtiyaçlarla birlikte Von Neuman ve Harvard modelleri ile tasarlanılan sistemler yapılmıştır.

Günümüz bilgisayarlarında benimsenmiş olan John Von Neumann tarafından geliştirilmiş Von Neumann mimarisi hem komutları hem de verileri aynı bellek üzerinde bulundurur [4]. Veri yolu üzerinden aynı anda iki farklı bilgi geçemediği için Merkezi İşlem Birimi (MİB) önce komut verisini alır sonra verinin gelmesini bekler. Hafıza biriminin MİB'den konum olarak uzakta olması ve sürekli olarak veri gelgiti olması bu mimarinin bir dezavantajı olarak görülebilir. Günümüz teknolojilerinde bu sorunun önüne geçmek için MİB ile hafıza biriminin arasına ön bellekler yerleştirilir.

Harvard mimarisi ise Von Neumann'ın yaklaşımından farklı olarak program belleği ile veri belleğini farklı veri yolları üzerinden Merkezi İşlem Birimine (MİB) bağlamıştır (Şekil 2).



Şekil 2: Von Neumann ve Harvard mimarilerinin kıyaslanması

## 2209/A ÜNİVERSİTE ÖĞRENCİLERİ ARAŞTIRMA PROJELERİ DESTEĞİ PROGRAMI ARAŞTIRMA ÖNERİSİ FORMU

Harvard mimarisinde belleklerin ayrı veri yolları ile birbirine bağlı olması sayesinde Von Neumann mimarisinde iki çevrimde yapılan bir işlemi tek çevrimde yapılabilir hale getirmiştir. Bu zamandan ve hızdan kazanma durumu kontrol ünitesi tasarımını karmaşıktırmıştır ve maliyeti arttırmıştır. Bu nedenle günümüz bilgisayarlarında, daha çok kişisel bilgisayarlarda Von Neumann mimarisi kullanılır. Sinyal işleme gibi işlemler yapılırken veya mikrokontrolcülerde harvard mimarisi tercih edilir [5].

Kontrol ünitesi ardışık olarak gelen komutlara göre hangi veri yollarından veriler geçeceğini, hangi seçim uçlarının (multiplexer veya MUX) seçileceğini, hangi hesaplamaların ve kontrollerin yapılacağını belirleyen devredir [6]. Komutların genelde başında veya sonunda bulunan ve o anda hangi operasyonun yapılacağını belirten "opcode" kısmı bulunur. Buradaki değer veya veri doğrudan kontrol ünitesine gider. Kontrol ünitesinin içerisinde VE ve VEYA mantık kapıları sayesinde gerçekleştirilecek seçimler, açılacak veya kapanacak veri yolları, belleğe yazma veya okuma işlemleri belirlenmiş olur.

Bilgisayarlara istediğimiz işlemleri yaptırmak için birden çok ardışık kod içeren programlar yazarız. Bu sırada bildiğimiz üst seviye programlama dilleri kullanılır. Derleyiciler bu programı veya kodu alt seviye programlama dili olan makine diline çevirir. Makine dili bilgisayarın içindeki donanımın nasıl tasarlandığına, hangi mimariyi kullandığına, hangi komut seti ile çalıştığına göre değişir. Komut setinde, en temel ve en alt seviyede komutlar bulunur. Bunlar doğrudan donanıma ne yapması gerektiğini anlatan komutlardır. Genelde veri aktarımı, aritmetik işlemler, karşılaştırma işlemleri, bit işlemleri, giriş çıkış işlemleri yapan komutlardır [7]. Örneğin Arduino mikrokontrolcüsünün kalbi olan Atmega328 mikroişlemcisi AVR komut setini kullanır.

Bu kapsamda incelenen mimari modeller, komut setleri ve mimari tasarımlardan ilham alınarak temel işlemleri hesaplayabilen, giriş-çıkış ünitesi, kontrol ünitesi, hafıza birimi, kaydedici bloğu, veri yolları, program belleği ve değiştirilebilir frekansta çalışan CLOCK(saat) sinyali bulunan bir bilgisayar tasarımı yapılacaktır. Tasarlanacak bilgisayar Harvard modelini benimseyecektir. Yöntem kısmında da anlatılacağı üzere kendine has komut seti tasarımı vardır. Tüm komutlar aynı uzunlukta ve 3 bayt (byte) uzunluğundadır. 24 bit paralel çıkış veren bir program belleği kullanmak için 3 adet 8 bit çıkış veren hafıza birimleri kullanılmıştır. Tek bir komut ile hem kontrol ünitesine, hem kaydedici adreslerine, hem de anlık değer verisine erişilebilir. Tüm bu aşamaların bitişinde çalışan bir sistemi kolay programlanabilir hale getirmek için derleyici tasarımı yapılacaktır. Mimari tasarımı, komut seti tasarımı ve derleyici tasarımı bu araştırma önerisinin özgün olan aşamalarıdır.

Tüm bunların sonucunda bu araştırma ve gerçekleştirme ile son zamanlarda yaşanan dünya geneli çip sorununa, yerli mikroişlemci, komut seti, mimari ve derleyici tasarımına, öğrenerek katkıda bulunulmaya çalışılacaktır. Bununla birlikte önce Bolu Abant İzzet Baysal öğrencilerine, sonra tüm Türkiye'deki araştırmacılara en baştan bir bilgisayar tasarımının, üretiminin ve çalıştırılmasının nasıl bir yol ve yöntem ile yapıldığı aktarılmaya çalışılacaktır. Bu araştırma aynı zamanda simülasyon ortamında test edilen ve çalışan bir sistemin, laboratuvar ortamında uygulamaya dökerek test edilmesini, çalıştırılmasını içerir.

### 1.2. Amaç ve Hedefler

Araştırmanın amacı 8 bitlik programlanabilir mikroişlemci tasarlamak ve uygulama ortamında test edip prototipini üretmektir.

Hedefler ;

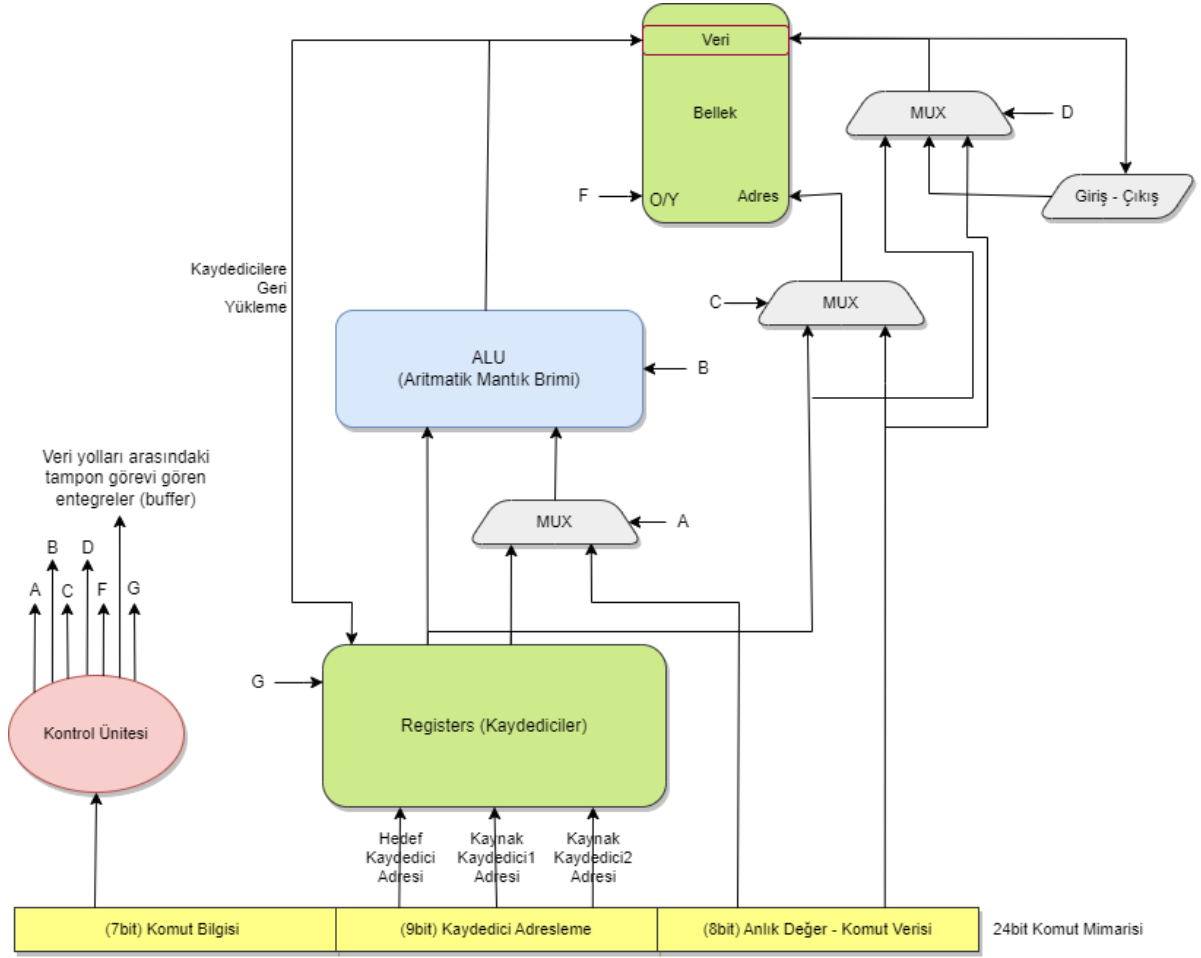
- Mimari tasarımı yapmak
- Kontrol ünitesi tasarlamak
- Hafıza birimi tasarlamak
- Komut Seti tasarlamak
- Derleyici tasarlamak
- Devreyi uygulamalı bir şekilde breadboard üzerinde kurularak test etmek
- Baskı devreye verilip üretime geçirmek

### 2. YÖNTEM

Elektronik bileşenler (ve, veya, değil, veya değil mantık kapıları / toplama, çıkarma, büyüktür, küçüktür, eşittir hesaplama entegre devreleri / paralel veri yolları / kaydedici entegre devreleri / hafıza birimleri) kullanılarak 8 bit veri yollu bilgisayar mimarisi tasarlanmıştır. Yukarıda sayılan entegre devreler tamamen paralel giriş ve paralel çıkış pinleri bulunduran entegreler kullanılarak yapılmıştır. Bu özelliği paralel sistem yaklaşımını benimsiyor ve üretimi sırasında hata tespitini hızlandırmaya yarıyor. Tüm sistem önce delikli deneysel elektrik tahtası (breadboard) üzerinde kurulacaktır. Daha sonra baskı devreye verilebilir şekilde hazırlanacaktır.

Literatür taramasına istinaden yapılacak olan araştırmada tasarlanacak olan bilgisayarın mimari modeli olarak

harvard mimarisi seçilmiştir. Veri belleği ile komutların tutulacağı program belleği birbirinden ayrı veri yolları ile aritmetik mantık birimine, kaydedicilere ve giriş-çıkış birimine bağlıdır.



**Şekil 3: Tasarlanacak mimari genel görünüm**

Tasarımda program belleği ve üzerinde işlem yaptığımız verileri saklayan veri belleği, birbirinden ayrı şekilde fakat eş zamanlı olarak çalışıyor. Komut belleği tasarımı 3 adet bellek kullanıldı. Buradaki amaç hem sistemin kontrolünün kolay olması ve hata tespitini hızlandırmak hem de paralel veri yolu sistem yaklaşımını bozmamak ve aynı zamanda komut seti tasarımını kolaylaştırmaktır.

Komut seti tasarlanırken tüm komutların tek bir saat darbesi ile tamamlanabilir olmasına dikkat edilmiştir. Her bir komut 3 bayt yani 24 bit uzunluğundadır.

- İlk 7 bit ile (OPCODE) kontrol ünitesi kontrol edilmiştir (Şekil 3 Komut Bilgisi). 7 bit ile  $2^7 = 128$  farklı komut kullanılabilir fakat şimdilik 28 farklı temel komuttan oluşturulmuştur. Daha sonradan yeni komutlar eklenebilir, genişletilebilir şekilde bırakılmıştır. Komutların arasında 4 aritmetik işlem komutu, 4 mantıksal işlem komutu, 2 bit kaydırma komutu, veri kaydetme ve işleme komutları, dallanma komutları, giriş ve çıkış kontrolü komutları bulunur (Tablo 1).
- Sonraki 9 bitin her 3 biti ile sırasıyla hedef kaydedici (HY), kaynak kaydedici 1 (KY1) ve kaynak kaydedici 2 (KY2) adreslemesi yapılmaktadır. Bu bitler sayesinde o an çalışan komuta göre hangi kaydediciye verinin yazılacağı veya hangi kaydediciden verinin okunacağı kontrol edilmiş olur.
- Son 8 bit ise gelen komuta göre anlık bir değer, adres versisi veya kayıt versisi olabilir. Tablo 1 açıklamalar kısmında her bir komut detaylı olarak açıklanmıştır.

**Tablo 1: Komut Seti**

			7 BİT	3BİT	3 BİT	3 BİT	8 BİT	
--	--	--	-------	------	-------	-------	-------	--

**2209/A ÜNİVERSİTE ÖĞRENCİLERİ ARAŞTIRMA PROJELERİ DESTEĞİ PROGRAMI**  
**ARAŞTIRMA ÖNERİSİ FORMU**

	Komut	Açıklama	Kontrol Ünitesi Kodu	HY	KY1			KY2	Anlık Değer	Açıklama
1	ADD	İki kaydedicideki veriyi toplar	0000001	✓	✓			✓	×	HY = KY1 + KY2
2	ADDIMM	Bir kaydedicideki veri ile anlık değeri toplar	0000010	✓	✓			×	✓	HY = KY1 + ANLIK
3	SUB	İki kaydedicideki veriyi çıkarır	0000011	✓	✓			✓	×	HY = KY1 – KY2
4	SUBIMM	Bir kaydedicideki veriden anlık değeri çıkarır	0000100	✓	✓			×	✓	HY = KY1 – ANLIK
5	AND	İki kaydedicideki verileri VE işlemine sokar	0000101	✓	✓			✓	×	HY = KY1 && KY2
6	ANDIMM	Bir kaydedicideki veri ile anlık değeri VE işlemine sokar	0000110	✓	✓			×	✓	HY = KY1 && ANLIK
7	OR	İki kaydedicideki verileri VEYA işlemine sokar	0000111	✓	✓			✓	×	HY = KY1    KY2
8	ORIMM	Bir kaydedicideki veri ile anlık değeri VEYA işlemine sokar	0001000	✓	✓			×	✓	HY = KY1    ANLIK
9	SHIFTLEFT	Bir kaydedicideki veriyi bir BİT sola kaydırır	0001001	✓	KAYDEDİCİ <sub>3</sub>			×	×	HY = KY1[0:6]
10	SHIFTRIGHT	Bir kaydedicideki veriyi bir BİT sağa kaydırır	0001010	✓	KAYDEDİCİ <sub>3</sub>			×	×	HY = KY1[1:7]
11	LBFROMIMM	Anlık değerden aldığı veriyi hedef kaydediciye yükler	0001011	✓		×		×	✓	HY = ANLIK
12	LBFROMMEM	Hafızadan aldığı veriyi hedef kaydediciye yükler, Hafıza adreslerken anlık değeri kullanır	0001100	✓		×		×	✓	HY = MEM(ANLIK)
13	REGTOREG	Bir kaydedicideki veriyi başka bir kaydediciye aktarır, yükler	0001101	✓	✓	×	×	✓	×	HY = KY
14	SBFROMIMM	Bir kaydedicideki veriyi adres olarak kullanır ve hafıza birimine anlık değeri yazar	0001110	×	✓	×	×	✓	✓	MEM(KY) = ANLIK
15	SBFROMREG	Anlık değerdeki veriyi adres olarak kullanır ve hafıza birimine kaydedicideki veriyi yazar	0001111	×	✓	×	×	✓	✓	MEM(ANLIK) = KY

**2209/A ÜNİVERSİTE ÖĞRENCİLERİ ARAŞTIRMA PROJELERİ DESTEĞİ PROGRAMI**  
**ARAŞTIRMA ÖNERİSİ FORMU**

16	BEQ	İki kaydedicideki verinin eşit olma durumunu karşılaştırır. Eğer eşitse anlık değer kadar komut atlar	0010000	X	✓	✓	✓	IF KY1 == KY2 JUMP ANLIK		
17	BNE	İki kaydedicideki verinin eşit olmama durumunu karşılaştırır. Eşit değilse anlık değer kadar komut atlar	0010001	X	✓	✓	✓	IF KY1 != KY2 JUMP ANLIK		
18	BLT	Birinci kaydedicideki verinin ikinci kaydedicideki veriden küçük olma durumunu karşılaştırır. Eğer büyükse, anlık değer kadar komut atlar	0010010	X	✓	✓	✓	IF KY1 < KY2 JUMP ANLIK		
19	BGT	Birinci kaydedicideki verinin ikinci kaydedicideki veriden büyük olma durumunu karşılaştırır. Eğer büyükse, anlık değer kadar komut atlar	0010011	X	✓	✓	✓	IF KY1 > KY2 JUMP ANLIK		
20	BGE	Birinci kaydedicideki verinin ikinci kaydedicideki veriden büyük ve eşit olma durumunu karşılaştırır. Eğer büyük ve eşitse, anlık değer kadar komut atlar	0010100	X	✓	✓	✓	IF KY1 >= KY2 JUMP ANLIK		
21	BLE	Birinci kaydedicideki verinin ikinci kaydedicideki veriden küçük ve eşit olma durumunu karşılaştırır. Eğer küçük ve eşit ise, anlık değer kadar komut atlar	0010101	X	✓	✓	✓	IF KY1 <= KY2 JUMP ANLIK		
22	PRINTREG	Bir kaydedicideki veriyi çıkış birimine gönderir	0010110	X	✓	X	X	✓	X	OUTPUT(KY)

**2209/A ÜNİVERSİTE ÖĞRENCİLERİ ARAŞTIRMA PROJELERİ DESTEĞİ PROGRAMI**  
**ARAŞTIRMA ÖNERİSİ FORMU**

23	PRINTREGAND LOAD	Bir kaydedicideki veriyi çıkış birimine gönderir ve başka bir kaydediciye yükler	0010111	✓	✓	X	X	✓	X	HY = OUTPUT(KY)
24	PRINTMEMFR OMREG	Bir kaydedicideki veriyi adres olarak kullanır ve hafıza birimine gönderir. O adresteki veriyi çıkış birimine gönderir	0011000	X	✓	X	X	✓	X	OUTPUT(MEM(KY ))
25	PRINTMEMFR OMIMM	Anlık değeri adres olarak kullanır ve hafıza birimine gönderir. O adresteki veriyi çıkış birimine gönderir	0011001	X	X	X	X	✓		OUTPUT(MEM(AN LIK))
26	WRITETOREG	Giriş biriminden gelen veriyi bir kaydediciye yükler	0011010	✓	X	X	X	X		HY = INPUT
27	WRITETOMEM FROMREG	Bir kaydedicideki veriyi adres olarak kullanır ve giriş biriminden gelen veriyi hafıza birimine yükler	0011011	X	✓	X	X	✓	X	MEM(KY) = INPUT
28	WRITETOMEM FROMIMM	Anlık değerden aldığı veriyi adres olarak kullanır ve giriş biriminden aldığı veriyi hafıza birimine yükler	0011100	X	X	X	X	✓		MEM(ANLIK) = INPUT

✓ işareti komuta göre kullanılacak alanları temsil eder

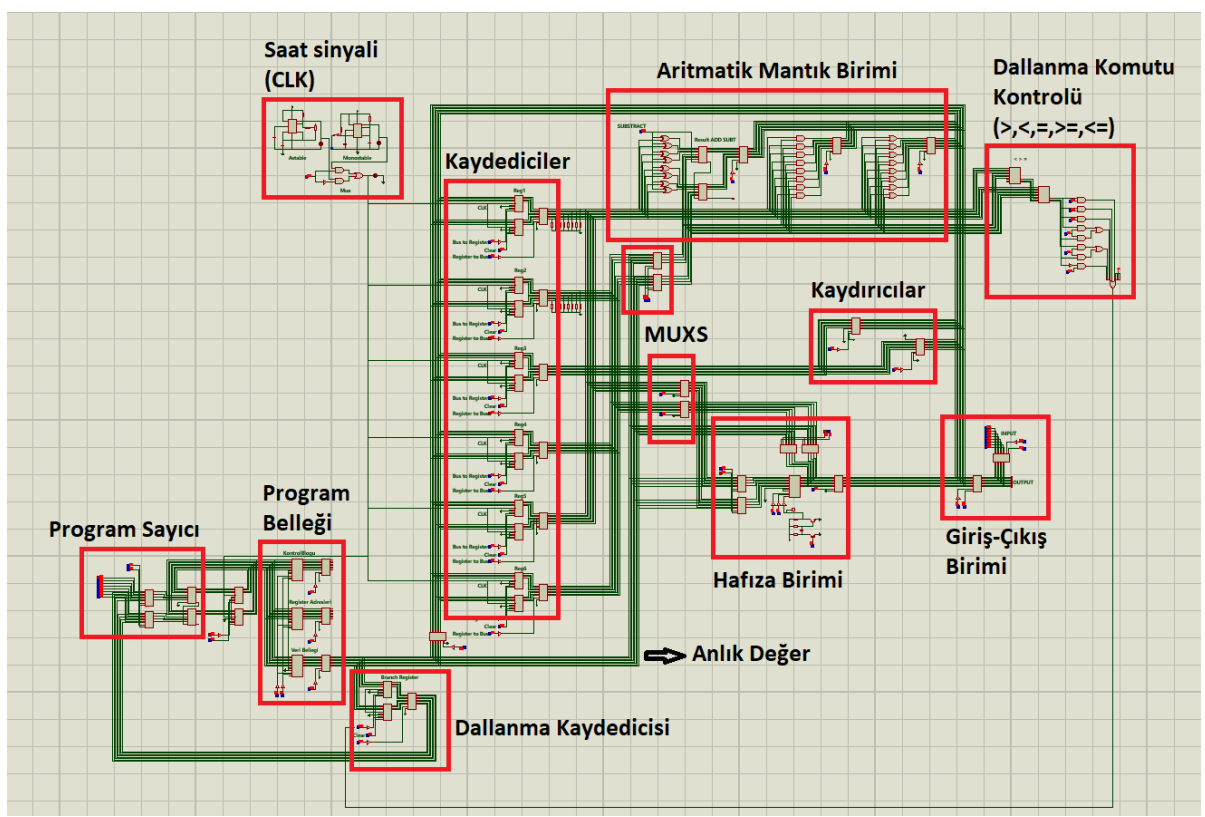
X işareti komuta göre kullanılmayan alanları temsil eder

6 adet genel amaçlı kaydedici (register) bulunur.  $2^3 = 8$  olduğundan 6 adet kaydediciyi adreslemek için 3 bit gerekmektedir. Bu nedenle komut seti tasarımında HY, KY1 ve KY2 için toplam 9 bit adresleme alanı ayrılmıştır.

- 1, 3, 5 numaralı kaydediciler doğrudan Aritmetik Mantık Birimini ilk girişine bağlanmıştır.
- 2, 4, 6 numaralı kaydediciler aradaki bir seçici (multiplexer) aracılığıyla Aritmetik Mantık Biriminin ikinci girişine bağlanmıştır. Seçicinin (multiplexer) diğer seçim ucu ise anlık değerden gelen veriye bağlanmıştır. Bu seçici, gelen komuta göre seçim yapar ve Aritmetik Mantık Birimine veriyi iletir.
- 3 numaralı kaydedici özel olarak kaydırma işlemi yapmak için kullanılmaktadır. Diğer kaydedicilerdeki verileri kaydırma işlemi yapabilmek için önce 3 numaralı kaydediciye yüklenmesi gerekmektedir.

1 adet özel amaçlı kaydedici bulunur. Dallanma komutları için atlayacağı değeri tutan bir kaydedici kullanılmıştır.





Şekil 4: Proteus Programında Tasarımın gösterimi

Kontrol ünitesine gelecek 7 bitin sadece 5 biti ile bir decoder (kod çözücü) kullanılarak veri yollarının aktifliği, yazma-okuma işlemlerinin durumları düzenlenecektir. Buradan anlayacağınız üzere tek çevrimde bütün komutları çalıştırabilmek için kontrol ünitesinin karmaşıklığı kabul edilmiştir. Şekil 3 de görüldüğünün aksine az sayıda kontrol değil, çok sayıda buffer (tampon) kontrolü vardır. Bağlantı karmaşıklığını azaltmak için tüm verilerin 8 bitlik ortak veri yolunda dolaşması sağlanmıştır.

Şekil 4'de de görüleceği üzere üretilecek bilgisayar test edilebilir şekilde elektronik simülasyon programı Proteus'da tasarlandı. (EK – 2). Son olarak kontrol ünitesinin ve kaydedici adresleme sisteminin eklenmesi ve derleyici tasarımı kalmıştır.

## 3 PROJE YÖNETİMİ

## 3.1 İş- Zaman Çizelgesi

## İŞ-ZAMAN ÇİZELGESİ (\*)

İP No	İş Paketlerinin Adı ve Hedefleri	Kim(ler) Tarafından Gerçekleştirileceği	Zaman Aralığı (... Ay)	Başarı Ölçütü ve Projenin Başarısına Katkısı
1	Kontrol ünitesinin tasarımı	Musa Akyüz	0-2 Ay	%20 -- Devrenin kendi kontrollerini anlık olarak yapabilmesi için şarttır. Saat sinyalinin düşük ve yüksek frekanslarında aynı doğrulukta çalışan bir kontrol sistemi başarının ölçütüdür. Sadece belli bir saat sinyali ile çalışan kontrol ünitesi ise minimum başarı kriteridir.
2	Kaydedici adresleme sistemi tasarımı	Musa Akyüz	2-4 Ay	%10 – Herhangi bir kaydedici adresleme yapmadan sadece iki adet kaydedici kullanılarak da aynı sistem tasarlanabilir. Fakat fazla kaydedici olması komut yazma aşamasında kolaylık sağlayacağı için adresleme sistemi yapılmalıdır. Komut ile gelen bitlerden adresi ayırt ederek kaydedicileri doğru seçen sistem başarının ölçütüdür.
3	Programlayıcı tasarımı – Program belleğine yeni bir program yazılırken kullanılan sistemdir.	Musa Akyüz	4-6 Ay	%10 – Kullanıcıların sisteme kendi programını yükleyebilmesi için şarttır. Bir aksesuar olarak görülebilir. Sistemin çalışırılığına etkisi yoktur. Sadece program yükleme işlemlerini hızlandıracaktır.
4	Derleyici tasarımı	Musa Akyüz	6-8 Ay	%30 – Programlanabilir bir sistem olması için derleyiciye ihtiyaç yoktur fakat kullanıcıların komut verirken rahat anlamaları ve yazmaları önemlidir. Bu nedenle bir çevirici şarttır. Yazılan komutları istenilen şekilde ikilik sisteme çeviren bir derleyici başarının ölçütüdür.
5	Derleyici gerçekleştirmesi (implementation)	Musa Akyüz	8-10 Ay	%20 – Derleyici tasarımı (4) iş paketinin başarı ölçütü ile aynı
6	Baskı Devre için sistemin hazırlanması	Musa Akyüz	10-12 Ay	%10 – Daha küçük, elle tutulur, taşınabilir, programlanabilir olmasını sağlar. Projenin çalışırılığına etkisi yoktur ama efektifliği artırır.

(\*) Çizelgedeki satırlar ve sütunlar gerektiği kadar genişletilebilir ve çoğaltılabilir.

**2209/A ÜNİVERSİTE ÖĞRENCİLERİ ARAŞTIRMA PROJELERİ DESTEĞİ PROGRAMI**  
**ARAŞTIRMA ÖNERİSİ FORMU**

**3.2 Risk Yönetimi**

**RİSK YÖNETİMİ TABLOSU\***

İP No	En Önemli Riskler	Risk Yönetimi (B Planı)
1	Mimari tasarımındaki verilen bir karardan dolayı kontrol ünitesinin tasarımında aksaklık yaşanması	Mimari tasarımında ufak değişikliklere gidilir. Kontrol ünitesi tasarımı tekrar yapılır.
2	Kaydedici sayısının yetersiz kalması veya aşırı olması	Mimari tasarımda ufak değişikliklere gidilir. Duruma göre özel işlevleri olan kaydediciler eklenir.
3	Programlayıcıyı gerçekleştirememek	Tüm komutlar manuel olarak denenir. Yeni program belleği tasarımı oluşturulur. RAM modülleri yerine ROM gibi tek program yüklenebilen veya EEPROM gibi tekrar tekrar yüklenebilen bellekler tercih edilir.

**3.3. Araştırma Olanakları**

**ARAŞTIRMA OLANAKLARI TABLOSU (\*)**

Kuruluşta Bulunan Altyapı/Ekipman Türü, Modeli (Laboratuvar, Araç, Makine-Teçhizat, vb.)	Projede Kullanım Amacı
Donanım Laboratuvarı	Projenin gerçekleştirmesi sırasında çalışma mekanı olarak kullanılacaktır.
Bilgisayar ve Yazılım Laboratuvarı	Simülasyon ortamında projenin test edilmesi, çalıştırılması ve derleyici gerçekleştirmesi sırasında çalışma mekanı olarak kullanılacaktır.
Multimetre	Hata tespiti ve kablo kalitesi, direnç, volt ve amper ölçümleri sırasında kullanılacaktır.
Osiloskop	Hata tespiti ve frekans ölçümü sırasında kullanılacaktır.
Güç kaynağı	Projenin gerçekleştirilmesi sırasında ve tasarımın çalıştırılması ve test edilmesinde kullanılacaktır.
Lehimleme Ekipmanları	Projenin gerçekleştirilmesi sırasında kullanılacaktır.
Kablolama ekipmanları, pense, yan keski, kablo soyucu gibi araçlar	Projenin gerçekleştirilmesi sırasında kullanılacaktır.
Güvenlik Ekipmanları	Güvenli çalışma ortamını sağlamak amacıyla kullanılacaktır.

**4. YAYGIN ETKİ**

**ARAŞTIRMA ÖNERİSİNDEN BEKLENEN YAYGIN ETKİ TABLOSU**

Yaygın Etki Türleri	Önerilen Araştırmadan Beklenen Çıktı, Sonuç ve Etkiler
<b>Bilimsel/Akademik</b> (Makale, Bildiri, Kitap Bölümü, Kitap)	İzlenen yolu, adımları, yenilikleri açıklayan bir bildiri veya makale yazılması.

**2209/A ÜNİVERSİTE ÖĞRENCİLERİ ARAŞTIRMA PROJELERİ DESTEĞİ PROGRAMI**  
**ARAŞTIRMA ÖNERİSİ FORMU**

<b>Ekonomik/Ticari/Sosyal</b> (Ürün, Prototip, Patent, Faydalı Model, Üretim İzni, Çeşit Tescilli, Spin-off/Start- up Şirket, Görsel/İşitsel Arşiv, Envanter/Veri Tabanı/Belgeleme Üretimi, Telif Konu Olan Eser, Medyada Yer Alma, Fuar, Proje Pazarı, Çalıştay, Eğitim vb. Bilimsel Etkinlik, Proje Sonuçlarını Kullanacak Kurum/Kuruluş, vb. diğer yaygın etkiler)	Tasarımı yapılan ürünün prototipini üretilmesi. Patent haklarının araştırılması ile ülkemize katkısının incelenmesi. Konuya ilgisi olan araştırmacılara izlenecek bir yol haritası oluşturulması.
<b>Araştırmacı Yetiştirilmesi ve Yeni Proje(ler) Oluşturma</b> (Yüksek Lisans/Doktora Tezi, Ulusal/Uluslararası Yeni Proje)	

**5. BÜTÇE TALEP ÇİZELGESİ**

Bütçe Türü	Talep Edilen Bütçe Miktarı (TL)	Talep Gerekçesi
<b>Sarf Malzeme</b>	6000	Tasarlanan devre üzerindeki tüm entegre devrelerin (zamanlayıcı devre, dirençler, kapasitörler, ledler, kablolar ve kablolama ekipmanları, anahtar veya switchler, mantık kapıları, toplama devreleri, kaydediciler, flipflop, multiplexerler, bufferlar, ram, rom veya eeprom modülleri, transistörler ve diğer entegrelerin) satın alınması
<b>Makina/Teçhizat (Demirbaş)</b>		
<b>Hizmet Alımı</b>		
<b>Ulaşım</b>		
<b>TOPLAM</b>	6000	

**NOT:** Bütçe talebiniz olması halinde hem bu tablonun hem de TÜBİTAK Yönetim Bilgi Sistemi (TYBS) başvuru ekranında karşınıza gelecek olan bütçe alanlarının doldurulması gerekmektedir. Yukardaki tabloda girilen bütçe kalemlerindeki rakamlar ile, TYBS başvuru ekranındaki rakamlar arasında farklılık olması halinde TYBS ekranındaki veriler dikkate alınır ve başvuru sonrasında değiştirilemez.

**6. BELİRTMEK İSTEDİĞİNİZ DİĞER KONULAR**

Sadece araştırma önerisinin değerlendirilmesine katkı sağlayabilecek bilgi/veri (grafik, tablo, vb.) eklenebilir.

--

**7. EKLER**

**EK-1: KAYNAKLAR**

- [1] Moore, G. E. (1965). Cramming more components onto integrated circuits.
- [2] Kish, L. B. (2002). End of Moore's law: thermal (noise) death of integration in micro and nano electronics. *Physics Letters A*, 305(3-4), 144-149.
- [3] Datasheet 8080A/8080A-1/8080A-2 8-BIT N-CHANNEL MICROPROCESSOR
- [4] Von Neumann, J. (1993). First Draft of a Report on the EDVAC. *IEEE Annals of the History of Computing*, 15(4), 27-75.
- [5] [Difference Between Von Neumann and Harvard Architecture \(byjus.com\)](http://byjus.com/difference-between-von-neumann-and-harvard-architecture/)
- [6] Patterson, D. A., & Hennessy, J. L. Computer Organization and Design RISC-V Edition: The Hardware Software Interface (The Morgan Kaufmann).
- [7] [Komut kümesi - Vikipedi \(wikipedia.org\)](https://tr.wikipedia.org/wiki/Komut_kumesi)