

**T.C.  
KASTAMONU ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ**

**FPGA TABANLI BİLGİSAYAR MİMARİSİ ÖĞRETİMİNDE  
MİKRO ÖĞRENME VE PROJE TABANLI ÖĞRENME  
YÖNTEMLERİNİ KULLANARAK KARMAŞIK PROJELERE  
KOMBİNE BİR YAKLAŞIM**

**Ismail Ibrahim Mohamed SAID**

<b>Danışman</b>	<b>Dr. Öğr. Üyesi Muhammet Serdar ÇAVUŞ</b>
<b>Jüri Üyesi</b>	<b>Dr. Öğr. Üyesi Faruk ERKEN</b>
<b>Jüri Üyesi</b>	<b>Dr. Öğr. Üyesi Abdulkadir KARACI</b>
<b>Jüri Üyesi</b>	<b>Doç. Dr. Hüseyin DEMİREL</b>
<b>Jüri Üyesi</b>	<b>Dr. Öğr. Üyesi Ersagun Kürşat YAYLACI</b>

**DOKTORA TEZİ  
MALZEME BİLİMİ VE MÜHENDİSLİĞİ ANA BİLİM DALI**

**KASTAMONU – 2019**

## TEZ ONAYI

**Ismail Ibrahim Mohamed SAID** tarafından hazırlanan "**FPGA Tabanlı Bilgisayar Mimarisi Öğretiminde Mikro Öğrenme ve Proje Tabanlı Öğrenme Yöntemlerini Kullanarak Karmaşık Projelere Kombine Bir Yaklaşım**" adlı tez çalışması aşağıdaki jüri üyeleri önünde savunulmuş ve **oy birliği** ile Kastamonu Üniversitesi Fen Bilimleri Enstitüsü **Malzeme Bilimi ve Mühendisliği Ana Bilim Dalı**'nda **DOKTORA TEZİ** olarak kabul edilmiştir.

Danışman	Dr. Öğr. Üyesi Muhammet Serdar ÇAVUŞ..... Kastamonu Üniversitesi	
Jüri Üyesi	Dr. Öğr. Üyesi Abdulkadir KARACI Kastamonu Üniversitesi	
Jüri Üyesi	Doç. Dr. Hüseyin DEMİREL Karabük Üniversitesi	
Jüri Üyesi	Dr. Öğr. Üyesi Faruk ERKEN Kastamonu Üniversitesi	
Jüri Üyesi	Dr. Öğr. Üyesi Ersagun Kürşat YAYLACI ... Karabük Üniversitesi	

06/09/2019

Enstitü Müdürü Doç. Dr. Nur BELKAYALI



## TAAHHÜTNAME

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, ayrıca tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildirir ve taahhüt ederim.



Ismail Ibrahim Mohamed SAID

## ÖZET

Doktora Tezi

### FPGA TABANLI BİLGİSAYAR MİMARİSİ ÖĞRETİMİNDE MİKRO ÖĞRENME VE PROJE TABANLI ÖĞRENME YÖNTEMLERİNİ KULLANARAK KARMAŞIK PROJELERE KOMBİNE BİR YAKLAŞIM

Ismail Ibrahim Mohamed SAID  
Kastamonu Üniversitesi  
Fen Bilimleri Enstitüsü  
Malzeme Bilimi ve Mühendisliği Ana Bilim Dalı

Danışman: Dr. Öğr. Üyesi Muhammet Serdar ÇAVUŞ

Bu çalışma, öğrencilerin, Alanda Programlanabilir Kapı Dizisi (FPGA) teknolojisi, Mikro Öğrenme (ML) ve Proje Tabanlı Öğrenme (PBL) yöntemlerini kullanarak karmaşık projeleri kısa sürede anlamalarına, başarılı bir şekilde gerçekleştirmelerine ve yeni projeler geliştirmelerine yardımcı olmak amacıyla yapılmıştır. Bu çalışmanın merkezi işlem birimi (CPU) tasarımı, VHSIC Donanım Tanımlama Dili (VHDL) yardımıyla, PBL ve ML öğrenme yöntemleri kullanılarak gerçekleştirilmiştir. Bu yeni yaklaşımlar ve uygulamalarla, üniversite düzeyinde, bilgisayar ve elektronik mühendisliği eğitimi alanında geleneksel eğitim/öğretimin eksiklikleri ele alınmış, bazı karmaşık araştırma projeleri üzerinde PBL ve ML yöntemleri kullanılarak bilgisayar mimarisi öğreniminin hızlandırılması amaçlanmıştır. Birleşik yöntemin, çekicilik ve geçerlik faktörleri açısından daha fazla yarar sağlağı belirlenmiştir. Bununla birlikte ev ödevleri, sınavlar ve projeler açısından birleşik yöntem diğer yöntemlere göre daha başarılı sonuçlar vermiştir.

**Anahtar Kelimeler:** CPU, FPGA, VHDL, dijital tasarım Elektronik, Mühendislik Eğitimi, Mikro Öğrenme, Proje Tabanlı Öğrenme

**2019, 150 sayfa**  
**Bilim Kodu: 91**

## ABSTRACT

Ph.D. Thesis

### AN COMBINED APPROACH TO COMPLEX PROJECTS USING MICRO LEARNING AND PROJECT BASED LEARNING METHODS IN FPGA BASED COMPUTER ARCHITECTURE

Ismail Ibrahim Mohamed SAID  
Kastamonu University  
Graduate School of Natural and Applied Sciences  
Department of Material Science and Engineering

Supervisor: Asst. Prof. Dr. Muhammet Serdar ÇAVUŞ

**Abstract:** The aim of this study is to help students understand complex projects in a short time by using Field Programmable Gate Arrays (FPGAs) technology, Micro Learning (ML) and Project Based Learning (PBL) methods and to develop new projects successfully. The central processing unit (CPU) design of this study was carried out using PBL and ML learning methods with the help of VHSIC Hardware Identification Language (VHDL). With these new approaches and applications, the deficiencies of traditional education / training in the field of computer and electronic engineering education at university level have been addressed and it is aimed to accelerate computer architecture learning by using PBL and ML methods on some complex research projects. It has been shown that the combined method has benefited more in terms of attractive and effective factors. However, in terms of homework, exams and projects, the combined method gave more successful results than other methods.

**Key Words:** CPU, FPGA, VHDL, Digital design electronics, Engineering education, Micro learning, Project based learning

**2019, 150 pages**

**Science Code: 91**

## TEŞEKKÜR

Tez çalışmam ve doktora eğitimim sırasında bilgilerinden ve tecrübelerinden ziyadesiyle faydalandığım, bana olan desteği, sabrı, motivasyonu, gayreti, ahlaki cömertliği ve kibarlığından dolayı başta danışmanım Dr. Öğr. Üyesi Muhammet Serdar ÇAVUŞ'a çok teşekkür ederim. Ayrıca, teşvikleri, içgörülü yorumları ve tezime değer katan gözlemleri için Dr. Öğr. Üyesi Abdulkadir KARACI ve Dr. Öğr. Üyesi Faruk ERKEN'e, işbirliğinden ve yardımından dolayı Dr. Öğr. Üyesi Can Doğan VURDU'ya da ayrıca teşekkürlerimi sunarım.

Her durumda desteğini yanımda hissettiğim sevgili aileme ve dostlarıma da sonsuz teşekkürlerimi ve saygılarımı sunarım. Son olarak, anneme, eşime, kardeşlerime ve çocuklarıma bana verdikleri tüm destek, sevgi ve anlayış için teşekkür etmek istiyorum.

Tez konumuzun ve içerisine eklediğimiz her türlü destekleyici bilginin, bu yolda ilerleyen arkadaşlarıma da faydalı olması temennisiyle.

Ismail Ibrahim Mohamed SAID  
Kastamonu, Eylül, 2019

## İÇİNDEKİLER

	<b>Sayfa</b>
TEZ ONAYI .....	ii
TAAHHÜTNAME.....	iii
ÖZET .....	iv
ABSTRACT .....	v
TEŞEKKÜR.....	vi
İÇİNDEKİLER .....	vii
SİMGELER VE KISALTMALAR DİZİNİ.....	xi
ŞEKİLLER DİZİNİ.....	xii
TABLOLAR DİZİNİ .....	xiv
1. GİRİŞ .....	1
1.1. Arka Plan.....	3
1.2. Geleneksel Eğitimin Sorunları .....	5
1.3. Araştırmanın Motivasyonu.....	7
1.4. Çalışmanın Amacı ve Önemi .....	7
1.5. Çalışmanın Hipotezi .....	8
1.6. Çalışma Grupları ve Yaklaşımlar .....	9
1.7. Araştırma Metodolojisi.....	10
1.8. Çalışmanın Özeti ve Önemi .....	11
1.9. Örneklem .....	11
1.10. Verilerin Toplanması ve Analizi .....	11
1.10.1. Anket .....	12
2. ÖNCEKİ ÇALIŞMALAR VE KURAMSAL ÇERÇEVE.....	15
2.1. Mikro Öğrenme (ML) .....	15
2.1.1. Faktörlerin ML ile İlişkisi.....	16
2.1.1.1. <i>Öğretim Faktörü - ML ilişkisi</i> .....	16
2.1.1.2. <i>Çekicilik Faktörü - ML ilişkisi</i> .....	16
2.1.1.3. <i>Geçerlik Faktörü - ML ilişkisi</i> .....	17
2.1.1.4. <i>Yeterlik Faktörü - ML ilişkisi</i> .....	17
2.1.1.5. <i>Karmaşıklığa İzin Verme Faktörü - ML ilişkisi</i> .....	17
2.1.2. Mikro Öğrenmenin Avantajları .....	18
2.2. Proje Tabanlı Öğrenme (PBL) .....	18
2.2.1. Faktörlerin PBL ile ilişkisi.....	18
2.2.1.1. <i>Öğretim Faktörü - PBL ilişkisi</i> .....	19
2.2.1.2. <i>Çekicilik Faktörü - PBL ilişkisi</i> .....	19
2.2.1.3. <i>Geçerlik Faktörü - PBL ilişkisi</i> .....	19
2.2.1.4. <i>Yeterlik Faktörü - PBL ilişkisi</i> .....	19
2.2.1.5. <i>Karmaşıklığa İzin Verme Faktörü - PBL ilişkisi</i> .....	20
2.2.2. Proje Tabanlı Öğrenmenin Avantajları.....	20

2.3. Geleneksel Öğrenme (TR).....	21
2.4. Alanda Programlanabilir Kapı Dizisi (FPGA) .....	21
2.4.1. FPGA'nın Yapısı .....	22
2.5. Quartus Yazılımı .....	23
2.6. CPU Mimarisi ve Organizasyonu.....	24
3. MATERYAL VE METOD .....	26
3.1. Geleneksel Öğrenme .....	27
3.2. Mikro Öğrenme (ML) .....	29
3.3. Proje Tabanlı Öğrenme (PBL) .....	34
3.4. PBL ve ML'nin Kombinasyonu .....	38
3.5. Alanda Programlanabilir Kapı Dizisi (FPGA) .....	42
3.6. Araştırmanın Amacı .....	43
3.7. Çalışmanın Kapsamı.....	45
3.7.1. Çalışma Projesi Örnekleri.....	47
3.7.1.1. Birinci Seviye: FPGA Teknolojisini ve ML Yöntemini Kullanma	48
3.7.1.1.1. VHDL ile Çoklayıcı 4-1 .....	48
3.7.1.1.2. VHDL ile 8 Bit Aritmetik Komutlar .....	49
3.7.1.1.3. VHDL ile 8 bit Mantıksal Komutlar .....	51
3.7.1.1.4. VHDL ile 8 Bitlik kaydırma ve döndürme komut tasarımı .....	52
3.7.1.1.5. VHDL ile Kayıt (Bir Bayt) .....	54
3.7.1.2. İkinci seviye: FPGA Teknolojisini ve ML Yöntemini Kullanma...	55
3.7.1.2.1. ALUSHIFT'in VHDL ile Tasarımı ve Simülasyonu.....	55
3.7.2. VHDL ile ROM .....	58
3.7.3. VHDL ile Ana Bellek (RAM).....	59
3.7.4. VHDL ile Kontrol Birimi .....	59
3.7.5. VHDL ile Program Sayacı.....	64
3.7.6. VHDL ile CPU Tasarımı .....	64
3.7.6.1. İşlemci.....	64
3.7.6.2. İşlem Birimi .....	65
3.7.6.3. Kontrol Döngüsü .....	66
3.7.6.4. Veri Akışı .....	66
3.8. Araştırma Süreci.....	67
3.9. Araştırma Stratejisi.....	69
3.10. Veri Toplam .....	69
3.11. Veri Analizi .....	69
3.12. Projede Kullanılan Yöntemler ve Testler .....	70
3.13. Çalışmada İncelenen Faktörler .....	70
3.14. Çalışmaya Katılanlar .....	71
3.15. Çalışmadaki Karşılaştırmalar .....	71
3.16. Çalışmanın Güvenilirliği .....	72
3.17. Altera ile Tasarım.....	73
3.18. Quartus II.....	73

4. BULGULAR.....	74
4.1. Test Faktörleri .....	74
4.1.1. Öğretim Faktörü .....	74
4.1.1.1. <i>Normallik Testi</i> .....	74
4.1.1.2. <i>Güvenirlilik Analizi</i> .....	75
4.1.1.3. <i>ANOVA Analizi</i> .....	77
4.1.1.4. <i>Varyans Eşitliği - Levene Testi</i> .....	78
4.1.2. Çekicilik Faktörü .....	79
4.1.2.1. <i>Normallik Testi</i> .....	79
4.1.2.2. <i>Güvenirlilik Analizi</i> .....	80
4.1.2.3. <i>ANOVA Analizi</i> .....	82
4.1.2.4. <i>Varyans Eşitliği - Levene Testi</i> .....	83
4.1.3. Geçerlik Faktörü .....	84
4.1.3.1. <i>Normallik Testi</i> .....	84
4.1.3.2. <i>Güvenirlilik Analizi</i> .....	85
4.1.3.3. <i>ANOVA Analizi</i> .....	86
4.1.3.4. <i>Varyans Eşitliği - Levene Testi</i> .....	87
4.1.4. Yeterlik Faktörü.....	88
4.1.4.1. <i>Normallik Testi</i> .....	88
4.1.4.2. <i>Güvenirlilik Analizi</i> .....	89
4.1.4.3. <i>ANOVA Analizi</i> .....	90
4.1.4.4. <i>Varyans Eşitliği - Levene Testi</i> .....	91
4.1.5. Karmaşıklığa İzin Verme Faktörü .....	92
4.1.5.1. <i>Normallik Testi</i> .....	92
4.1.5.2. <i>Güvenirlilik Analizi</i> .....	93
4.1.5.3. <i>ANOVA Analizi</i> .....	94
4.1.5.4. <i>Varyans Eşitliği - Levene Testi</i> .....	95
4.2. İkinci Bölüm: Ev Ödevlerine Dayalı Test.....	96
4.2.1. Normallik Testi.....	96
4.2.2. ANOVA Analizi .....	97
4.3. Üçüncü Bölüm: Test Bazlı Ara Sınav .....	99
4.3.1. Normallik Testi.....	99
4.3.2. ANOVA Analizi .....	100
4.3.3. Varyans Eşitliği - Levene Testi .....	100
4.4. Dördüncü Bölüm: Final Sınavına Dayalı Test .....	102
4.4.1. Normallik Testi.....	102
4.4.2. ANOVA Analizi .....	102
4.4.3. Varyans Eşitliği - Levene Testi .....	103
4.5. Bulgular ve Tartışma .....	104
4.5.1. Anket Test Faktörleri.....	105
4.5.1.1. <i>Öğretim Faktörü</i> .....	105
4.5.1.2. <i>Çekicilik Faktörü</i> .....	106
4.5.1.3. <i>Geçerlik Faktörü</i> .....	106

4.5.1.4. <i>Yeterlik Faktörü</i> .....	107
4.5.1.4. <i>Karmaşıklığa İzin Verme Faktörü</i> .....	108
4.5.2. Ev Ödevleri Testi (V1, V2, V3) .....	109
4.5.3. Ara Sınav Testi .....	110
4.5.4. Final Sınavı Testi .....	111
5. SONUÇLAR VE ÖNERİLER .....	113
KAYNAKLAR .....	117
EKLER .....	123
EK 1. Anketler .....	124
EK 2. Birinci Seviye Proje .....	126
EK 3. İkinci Seviye Proje .....	129
EK 4. CPU .....	131
ÖZGEÇMİŞ .....	134

## SİMGELER VE KISALTMALAR DİZİNİ

CPU	Merkezi işlem birimi
FPGA	Alanda Programlanabilir Kapı Dizisi
VHDL	Yüksek hızlı tümleşik devre donanım tanımlama dili
TR	Geleneksel öğrenim
ML	Mikro öğrenim
PBL	Proje temelli öğrenim
ML+PBL	Mikro ve Proje temelli öğrenimin birleşimi/kombinasyonu
AU	Aritmetik birim
LU	Mantık birimi
SRU	Kayıdırma ve döndürme birimi
ALU	Aritmetik mantık birimi
ALUSHIFT	Aritmetik mantık kaydırma ve döndürme birimi
RAM	Rasgele erişimli bellek
ROM	Sadece okunabilir bellek
CU	Kontrol birimi
PC	Program sayacı
CAD	Bilgisayar destekli tasarım
LEs	Mantık elemanları

## ŞEKİLLER DİZİNİ

	<b>Sayfa</b>
Şekil 1.1. Geleneksel eğitimde karşılaşılan sorunlar.....	6
Şekil 1.2. Çalışma grupları .....	9
Şekil 2.1. FPGA'nın temel yapısı.....	23
Şekil 2.2. Komut döngüsü diyagramı .....	24
Şekil 3.1. Öğretim yöntemlerinde veri toplama planı .....	27
Şekil 3.2. Geleneksel öğrenme yöntemi .....	28
Şekil 3.3. Geleneksel öğrenim gören birinci grup için öğretim planı .....	28
Şekil 3.4. FPGA - ML etkileşimi .....	29
Şekil 3.5. ML öğretim kademeleri.....	30
Şekil 3.6. Basitten karmaşığa alt blok basamakları.....	31
Şekil 3.7. ML kullanarak hazırlanan tasarım yolları .....	32
Şekil 3.8. İkinci grup için mikro öğrenme ders akış diyagramı .....	34
Şekil 3.9. FPGA ve PBL'nin etkileşimi (Kurs 3 için). .....	36
Şekil 3.10. Proje tabanlı öğrenmenin seviyeleri.....	36
Şekil 3.11. Küçük alt bloklara bölünmüş CPU projesi .....	37
Şekil 3.12. Derslerin, üçüncü grup için PBL ile düzenlenmesi .....	38
Şekil 3.13. FPGA ile ML-PBL etkileşimi.....	38
Şekil 3.14. Basitten karmaşığa küçük alt bloklar halinde projenin aşamaları ....	39
Şekil 3.15. CPU projesinin basitten karmaşığa küçük alt bloklara bölünmesi ...	40
Şekil 3.16. Dördüncü grup için ML+PBL ile öğretim planı .....	41
Şekil 3.17. Tipik CAD akışı.....	42
Şekil 3.18. Mühendislik eğitimini geliştirme yöntemi.....	44
Şekil 3.19. Dört eğitim döneminde çalışma planı .....	46
Şekil 3.20. FPGA teknolojisi, PBL ve ML ile CPU tasarımı.....	48
Şekil 3.21. MUX 4-1'in yapısı.....	48
Şekil 3.22. Çoklayıcı 4-1'i doğrulama .....	49
Şekil 3.23. Bir aritmetik birimin yapısı.....	50
Şekil 3.24. Aritmetik komutların doğrulanması.....	50
Şekil 3.25. Mantık biriminin yapısı.....	51
Şekil 3.26. Mantıksal komutları doğrulama .....	52
Şekil 3.27. Sekiz bitlik bir ALUSHIFT için bit kaydırma örnekleri.....	53
Şekil 3.28. Kayma ve döndürme birimin yapısı.....	53
Şekil 3.29. Kaydırma ve döndürme talimatlarını doğrulama .....	54
Şekil 3.30. Doğrulama kaydı (bir bayt).....	55
Şekil 3.31. ALUSHIFT doğrulama komutları .....	56
Şekil 3.32. Salt okunur belleğin (ROM) doğrulanması.....	58
Şekil 3.33. Ana belleği (RAM) doğrulama .....	59
Şekil 3.34. Aritmetik komutlarla kontrol birimi doğrulaması .....	60
Şekil 3.35. Mantıksal komutlarla kontrol birimi doğrulama.....	61
Şekil 3.36. Kaydırma ve döndürmeyle kontrol birimi doğrulaması.....	61
Şekil 3.37. Kayıt yeri ile hafıza konumuna göre kontrol birimi doğrulaması.....	61
Şekil 3.38. Kayıtlı giriş birimi ile kontrol birimi doğrulaması .....	61
Şekil 3.39. Programı sayacını doğrulama .....	64
Şekil 3.40. Sistem veriyolu ve CPU gösterimi.....	65

Şekil 3.41. Komut döngüsü.....	66
Şekil 3.42. Veri akışı ve getirme döngüsü .....	66
Şekil 3.43. CPU doğrulama.....	67
Şekil 3.44. Çalışma metodolojinin akış şeması.....	68
Şekil 4.1. Dağılım histogramı ve öğretim faktörünün normal Q-Q grafiği.....	75
Şekil 4.2. Öğrenim yöntemlerinin öğretim faktörüne göre ortalamaları.....	77
Şekil 4.3. Dağılım histogramı ve çekicilik faktörün normal eğrisi .....	80
Şekil 4.4. Çekicilik faktörüne göre öğretim yöntemleri .....	82
Şekil 4.5. Dağılım histogramı ve geçerlik faktörünün normal grafiği .....	84
Şekil 4.6. Geçerlik faktörüne göre öğretim yöntemleri.....	86
Şekil 4.7. Dağılım histogramı ve yeterlik faktörü normal grafiği.....	89
Şekil 4.8. Yeterlik faktörü açısından öğretim yöntemleri .....	90
Şekil 4.9. Dağılım histogramı ve karmaşıklığa izin verme .....	93
Şekil 4.10. Karmaşıklık faktörüne göre öğretim yöntemleri.....	94
Şekil 4.11. Ev ödevlerine (1, 2 ve 3) ait dağılım histogramı .....	97
Şekil 4.12. Ödev 1 (hom1) için ML ve ML+PBL arasındaki fark.....	97
Şekil 4.13. Ödev 2 (hom2) için ML ve ML+PBL arasındaki fark.....	98
Şekil 4.13. Ödev 3 (hom3) için ML ve ML+PBL arasındaki fark.....	98
Şekil 4.15. Dağılım histogramı ve ara sınavın normal dağılımı .....	99
Şekil 4.16. Ara sınav sonuçlarına göre öğretim yöntemleri.....	100
Şekil 4.17. Dağılım histogramı ve final sınavının normal dağılımı .....	102
Şekil 4.18. Final sınavına göre öğretim yöntemleri .....	103
Şekil 4.19. Öğretim faktörü açısından ortalamalar .....	105
Şekil 4.20. Çekicilik faktörü açısından ortalamalar .....	106
Şekil 4.21. Geçerlik faktörü açısından ortalamalar .....	107
Şekil 4.22. Yeterlik faktörü açısından ortalamalar.....	108
Şekil 4.23. Karmaşıklığa izin verme faktörü açısından ortalamalar .....	109
Şekil 4.24. Ev ödevleri için ML ve ML+PBL yöntemleri .....	110
Şekil 4.25. Yöntemlerin ara sınav başarı verileri.....	111
Şekil 4.26. Final sınavı için yöntemlerin başarısı .....	112

## TABLolar DİZİNİ

	<b>Sayfa</b>
Tablo 3.1. Ödev seviyeleri.....	30
Tablo 3.2. FPGA ve ML yöntemlerinin etkileşimine ait kodlar ve anlamları...	31
Tablo 3.3. Kullanılan semboller ve anlamları .....	32
Tablo 3.4. Müfredat ve ders kodları .....	33
Tablo 3.5. Semboller ve anlamları .....	37
Tablo 3.6. Kullanılan kodlar.....	41
Tablo 3.7. Çoklayıcı 4-1 .....	48
Tablo 3.8. Aritmetik birim.....	50
Tablo 3.9. Mantık birimi .....	51
Tablo 3.10. Kaydırma ve döndürme birimi.....	54
Tablo 3.11. ALUSHIFT biriminin opcode tablosu .....	55
Tablo 3.12. Kontrol biriminin kontrol sinyali.....	62
Tablo 3.13. Adresleme komutları.....	67
Tablo 3.14. Projede kullanılan yöntemler ve testler .....	70
Tablo 3.15. Çalışmada incelenen faktörler .....	71
Tablo 3.16. Faktörlerin Cronbach Alfa güvenilirlik katsayısı .....	72
Tablo 4.1. Öğretim Faktörü açısından Normallik testi.....	75
Tablo 4.2. Öğretim faktörü Madde-Toplam istatistik değerleri .....	76
Tablo 4.3. Öğretim faktörü için tanımlayıcı istatistikler .....	77
Tablo 4.4. Öğretim faktörü ANOVA Analizi.....	78
Tablo 4.5. Öğretim faktörüne göre farklılıkların homojenliği testi.....	78
Tablo 4.6. Çoklu Karşılaştırmalar- Games-Howell Testi.....	79
Tablo 4.7. Çekicilik Faktörü için Normallik testi.....	80
Tablo 4.8. Çekicilik faktörü güvenilirlik analizi .....	80
Tablo 4.9. Çekicilik faktörü için madde-toplam istatistiği verileri .....	81
Tablo 4.10. Çekicilik faktörünün tanımlayıcı istatistikleri .....	82
Tablo 4.11. Çekicilik faktörü, ANOVA verileri .....	82
Tablo 4.12. Çekicilik faktörüne göre varyansların homojenliği testi.....	83
Tablo 4.13. Çekicilik faktörün çoklu karşılaştırmaları - Tukey testi. ....	83
Tablo 4.14. Geçerlik faktörü için normallik testi verileri .....	84
Tablo 4.15. Geçerlik faktörü için madde-toplam istatistiği .....	85
Tablo 4.16. Geçerlik faktörünün tanımlayıcı istatistikleri .....	86
Tablo 4.17. Geçerlik faktörü için ANOVA testi .....	87
Tablo 4.18. Geçerlik faktörü, varyansların homojenliği testi .....	87
Tablo 4.19. Geçerlik faktörünün çoklu karşılaştırmaları - Games-Howell testi. ....	88
Tablo 4.20. Yeterlik faktörüne göre Normallik testi.....	88
Tablo 4.21. Yeterlik faktörü için Madde-Toplam İstatistikleri.....	89
Tablo 4.22. Yeterlik faktörünün tanımlayıcı istatistikleri.....	90
Tablo 4.23. Yeterlik faktörü ANOVA verileri.....	91
Tablo 4.24. Yeterlik faktörü, varyansların homojenliği testi.....	91
Tablo 4.25. Yeterlik faktörünün çoklu karşılaştırmaları - Games-Howell testi..	92
Tablo 4.26. Karmaşıklığa izin verme faktörü için Normallik testi .....	92
Tablo 4.27. Karmaşıklığa izin verme faktörünün Madde-Toplam istatistiği ....	93
Tablo 4.28. Karmaşıklığa izin verme faktörü tanımlayıcı istatistikleri .....	94

Tablo 4.29. Karmaşıklıđa izin verme faktörü, ANOVA analzi verileri.....	95
Tablo 4.30. Karmaşıklık faktörü - homojenlik testi.....	95
Tablo 4.31. Karmaşıklıđa izin vereme faktörü - çoklu karşılařtırmalar (Tukey HSD).....	96
Tablo 4.32. Ev ödevleri için Normallik Testi verileri.....	96
Tablo 4.33. Ev ödevleri tanımlayıcı istatistikleri.....	98
Tablo 4.34. Ev ödevleri ANOVA verileri.....	99
Tablo 4.35. Ara sınav için Normallik testi.....	99
Tablo 4.36. Ara sınavın tanımlayıcı istatistikleri.....	100
Tablo 4.37. Ara sınav ANOVA verileri.....	100
Tablo 4.38. Ara sınav için varyansların homojenliđi.....	101
Tablo 4.39. Ara sınav verileri çoklu karşılařtırmaları.....	101
Tablo 4.40. Final sınavı için normallik testi.....	102
Tablo 4.41. Final sınavının tanımlayıcı istatistikleri.....	103
Tablo 4.42. Final sınavının ANOVA verileri.....	103
Tablo 4.43. Final sınavı için varyansların homojenliđi testi.....	103
Tablo 4.44. Final sınavının çoklu karşılařtırmaları (Tukey HSD).....	104
Tablo 5.1. Öğretim yöntemlerinin faktörlere göre durumları.....	114
Tablo 5.2. Öğretim yöntemlerinin ev ödevlerine göre durumları.....	114
Tablo 5.3. Öğretim yöntemlerinin sınavlara göre durumları.....	114

## 1. GİRİŞ

Geleneksel eğitim, bilgisayar mimarisi tasarımı için yeterli bir model değildir çünkü gerçek bir CPU tasarımı ve onun artan karmaşıklığı ile başa çıkmak için gerekli esnekliğe sahip değildir. Bu durum, temel projeler ve projelerin testleriyle gerçek dünyadaki uygulanabilirlikleri incelenerek ele alınabilir. Oysaki günümüz tasarımları çok sayıda elektronik kapağa ve elektronik devrelere sahiptir. Bu nedenle, bu tür projelerin geleneksel laboratuvarlarda uygulanması oldukça zor ve sıkıntılı bir süreçtir. Bilgisayar mimarisi gelişiminin gerisinde kalmamak ve onu yakından takip etmenin yanı sıra öncü atılımlar gerçekleştirebilmek için öğrencilerin tasarım yaklaşımlarını, geleneksel yaklaşımlarla mümkün olmayan artımlı karmaşıklıkla gerçekleştirmeleri gerekmektedir.

Bilgisayar mühendisliği eğitiminde bilgisayar mimarisi çok önemlidir ve vasıflı dijital mühendisleri yetiştirmek, CPU tasarım süreçlerine katkı sağlayarak bilgisayar mimarisinin gelişimini hızlandıracak ve bunun neticesinde etkin bir iş gücünü ortaya çıkaracaktır. Bunu sağlayabilmek için, kaliteli CPU tasarım eğitimi için özel olarak tasarlanmış uygulama projeleriyle etkili ve hızlı bir öğrenme sürecine ihtiyaç vardır. Bu amaç doğrultusundaki birçok girişim, öğrencilerin performanslarını ve etkinliklerini geliştirmek için umut verici alternatif eğitim stratejilerinin geliştirilmesini sağlamıştır. Proje temelli öğrenim ve mikro öğrenim teknikleri gibi yeni öğrenme yaklaşımları buna örnek olarak verilebilir. Ayrıca, Alanda Programlanabilir Kapı Dizisi (FPGA) teknolojisi de model tasarımı ve CPU mimarisi modellemesi için kullanılan ve yeniden yapılandırma ve benzetim açısından fayda sağlayan köklü bir yöntemdir. Bununla birlikte, öğrencilerin tatmin edici olmayan akademik performansları ve yavaş düşünme süreçleri, eğitim sistemlerinin tüm dünyada karşılaştığı en genel zorluklar arasındadır. Bunun yanı sıra, düşük performanslı öğrencilerin öğrenmesine yardımcı olmak için standartlardan taviz vermek de kısır bir döngü oluşturmaktadır. Öğrencilerin devam eden başarısızlık döngüsünden kurtulmalarına yardımcı olmak için öğretim metodolojilerinin geliştirilmeye ihtiyacı olduğu açıktır.

Bu çalışma, sürekli gelişen dünyada, giderek karmaşıklaşan eğitim ihtiyaçlarını karşılamak için geleneksel eğitim/öğretim sorunlarını çözmek amacıyla proje merkezli eğitim ve mikro eğitim yöntemlerini kullanarak daha hızlı, daha fayda sağlayıcı, test edilebilir ve geliştirmeye açık bir öğretim modelini ortaya koymayı amaçlamaktadır. Ayrıca, bu çalışma kapsamında üniversitelerde genel olarak uygulanmakta olan söz konusu yerleşik/geleneksel öğrenme süreci de incelenmiş ve irdelenmiştir.

Özet olarak, bu çalışma birçok geleneksel eğitim sorununu çözmeye odaklanmaktadır. Örneğin, öğrenciler dersleri sınırlı bir süre içinde ele almanın yanı sıra ders içeriğini de anlamak/özümsemek zorundadırlar. Bu kapsamda, çalışmanın ilk basamağında, Proje Tabanlı Öğrenme (PBL) ve Mikro Öğrenme (ML) yöntemleri incelenmiş ve bilgisayar bilimleri eğitiminde FPGA teknolojiyle nasıl entegre olarak uygulanabilecekleri tartışılmıştır. Akabinde, geleneksel yöntemlerin bir sonucu olarak ortaya çıkan öğrenim sorunlarının üstesinden gelebilmek için PBL ve ML yöntemleri birleştirilerek yeni bir öğretim uygulaması geliştirilmiştir. Ayrıca ödevler ve projeler kullanılarak geleneksel eğitim yaklaşımlarının yanı sıra ML, PBL ve ML+PBL yöntemlerinin öğrenci öğrenimi üzerine etkilerinin sonuçları analiz edilerek öğretim yöntemlerinin eksiklikleri ve pozitif yönleri ortaya konmuş ve yeni yöntemler için bir zemin hazırlanmıştır.

Bu çalışmanın bir diğer amacı ise öğrencilerin bir bilgisayar mimarisi dersinde yazılım ve donanım kullanmaları vasıtasıyla bazı donanım tasarımı ve öğretim yöntemlerinin bütünleştirilmesi yoluyla yapılan simülatöre dayalı yaklaşımı vurgulamaktır. Buna ek olarak, hedef, CPU tabanlı VHDL (Çok yüksek hızlı entegre devre donanım tanımlama dili) tasarımı yardımıyla FPGA teknolojisini de kullanarak PBL ve ML öğretim yöntemlerinin bir kombinasyonu ile öğrencilerin daha iyi ve hızlı öğrenme gerçekleştirmeleri hedeflenmiştir. Öğrencilerin geri dönütlerinin analizi neticesinde, söz konusu üst düzey, çok değişkenli sürecin, öğrencilerin karmaşık proje sorunlarını çözmelerine ve çok versiyonlu işlemci tasarımı gibi daha karmaşık projeleri gerçekleştirmelerine olanak sağladığı sonucuna varılmıştır.

Tez çalışması 6 bölüme ayrılmıştır:

*Birinci bölüm:* Çalışmanın süreci hakkında bilgi verir, ön bilgi gereksinimini belirler, problem tanımını yapar, ilgililiği belirler, amaçları ortaya koyar ve araştırma metodolojisini belirler.

*İkinci bölüm:* Literatüre genel bakışı, teorik çerçeveyi ve çalışılan modeller arasındaki ilişkileri içerir.

*Üçüncü bölüm:* Çalışmanın amaçlarına ulaşmak için benimsenen araştırma ve tasarım yöntemlerini açıklar. Araştırma araçları, araştırma yöntemleri ve veri toplama prosedürlerini içerir.

*Dördüncü bölüm:* Anket, ev ödevi, ara sınav ve final sınavından toplanan araştırma sonuçlarını ve verileri sunmaktadır.

*Beşinci Bölüm:* Sonuçlarla ilgili tartışma ve yorumları içerir.

*Altıncı bölüm:* Yüksek mühendislik eğitiminin gelişimi sürecine katkı sağlayabilecek önerileri içerir.

## **1.1. Arka Plan**

Bilgisayar mühendisliğinin ileri yöndeki evrimi, öğrencilerin daha etkili bir biçimde öğrenmelerini sağlamak amacıyla dijital tasarımın gerekliliklerini yerine getirmek için artan gereksinimlere ve öğrenme aşamalarını destekleyen yeni öğrenme yöntemlerinin geliştirilme zorunluluğunu ortaya koymuştur (Kiray, Demir ve ark., 2013). Mevcut birçok çalışma dijital elektroniği daha verimli ve hızlı bir şekilde öğretmek için ilgi çekici ve faydalı örnekler geliştirmenin önemi üzerinde durmaktadır. Ayrıca, iyi tasarlanmış uygulamaların, öğrencilerin performanslarını arttırdığı gözlenmiştir. Bununla birlikte, ML ve PBL öğrenme yöntemlerinin FPGA ile kullanımı, farklı üniversitelerin birçok elektronik mühendisliği ve bilgisayar mühendisliği bölümlerinde başarılı sonuçlar vermiştir (Said ve Çavuş, 2018).

Gerek elektronik gerekse bilgisayar bilimlerinde tasarım, öğrenme için temel şarttır. Tasarım eğitimi, teknolojinin daha etkili ve çok yönlü kullanımına olanak sağlar. Bu

nedenle multidisipliner projeler, genellikle, öğrencilerin temel kavramları anlamalarının ve uygulama ilkelerini öğrenmelerinin yanı sıra bilgiye erişme konusunda derin ve bütünlük bir anlayışa sevk etmektedir (Seinauskas, 1997). Mevcut öğrenim sürecinde, lisans öğrencileri “sistem tabanlı” anlamlı bir eğitime ihtiyaç duymaktadırlar. SOC (System on Chip) cihazlarının artan karmaşıklığı ile birlikte, sistem tabanlı tasarım becerisi kazanımı kaçınılmaz bir gerekliliktir (Wirthlin, 2005). Ayrıca, öğrencilerin mühendislik inovasyonunu sağlayan ve çok disiplinli mühendislik becerileri kazandıran pratik tasarım deneyimine de ihtiyaçları vardır.

Bilgisayar mühendisliği mezunları, sistem tabanlı tasarımlar geliştirme, tasarımlarını fonksiyonel cihazlar yardımı ile uygulama ve bu cihazları gerçek yaşam koşullarında test etme becerisine sahip olmalıdır. Profesyonel verimlilik sağlamak için, öğrencilerin bir dizi donanım tasarımı denemesi ve projesi ile çokça meşgul olmaları gerekmektedir (Mahmoud, 2006).

Mühendislik eğitimi, teknolojik gelişmelerin yüksek hızından dolayı çağa ayak uydurma ve teknolojiyi geliştirme çabası nedeniyle genellikle bir baskı altındadır. Elektrik, elektronik ve bilgisayar mühendisliği disiplinlerinde, müfredat tasarımcılarının, en yeni mesleki konuları müfredata dahil etmek ve mühendislik temellerini daha derinlemesine öğretmek arasında seçim yapmaları gerekmektedir. Zaman ve kredi saatleri, dersler, projeler ve diğer öğrenme metodolojileri yalnızca belirli bir bilgi dizisinin verilmesine olanak sağlayabilmektedir (Gonzalez-V ve Loya-Hernandez, 2007).

Daha yüksek eğitilmiş/bilgili öğrenci katılımı, etkili öğrenmenin arkasındaki kilit bir faktör olarak yaygın şekilde kabul edilmiştir. Eğitim uzmanları, geleneksel derslerin öğrencilere istenilen bir konuyu/bilgiyi/kavramı hızlı bir şekilde öğretmek için yetersiz olduğunu, çünkü mühendislik eğitiminde teorik olarak alınan bilginin içselleştirilmesinde deneylerin önemli bir unsur olduğunu kabul etmişlerdir (Araujo ve Alves, 2008).

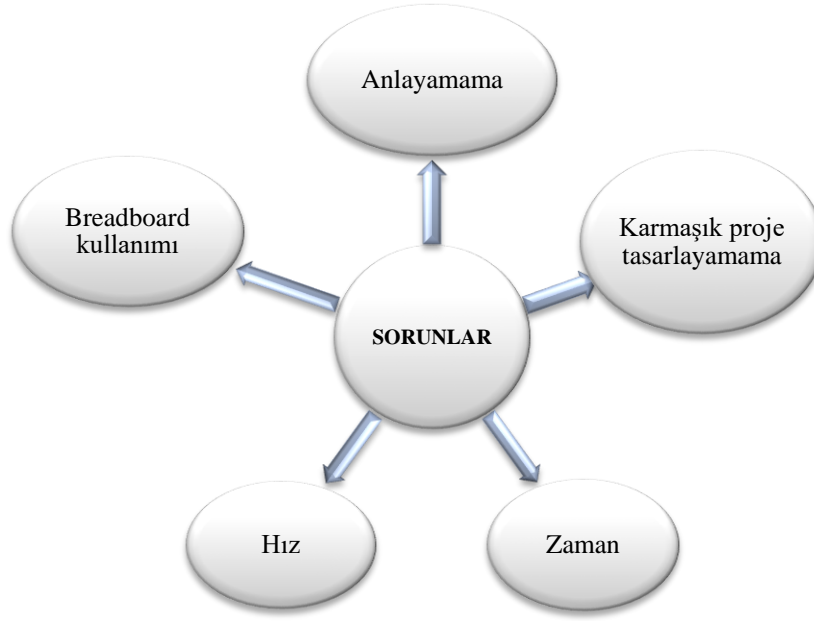
Öğretmenler, ülkelerin çoğunda, eğitim sistemlerinin öğrencilerin çoğunluğu için yetersiz olduğunu, çünkü yetenekleri ortaya çıkarıp geliştirmediğini, öğrenmeyi eğlenceli hale getirmede yetersiz kaldığını iddia edebilmektedirler (Friedman ve Friedman, 2018). Bunun yanısıra birçok kurum, kendisine gelir elde etmek için öğrencileri bir müşteri olarak görmektedir (Guilbault, 2018). Bununla birlikte, teknik üniversiteler, dünya çapında dijital tasarım dersleri vermektedirler, ancak, onları organize etmek için yoğun bir araştırmaya ve çabaya ihtiyaç vardır. Ayrıca, her ne kadar laboratuvar çalışmaları mühendislik derslerinin pratik bir bölümü olsa da çoğu durumda sadece benzetim (simülasyon) tabanlı yaklaşımlar takip edilmektedir (Sudnitson, 2010).

Çağımız dijital sistemlerinin giderek artan karmaşıklığı, günümüzün mühendislik eğitimi için büyük bir zorluktur. Teknolojik topluluk, akademik kurumlardan daha hızlı hareket ettiğinden, eğitim ve öğretim ile mühendislik uygulamalarındaki ticari ve profesyonel uygulamalar arasındaki fark giderek artmaktadır (Donzellini ve Ponta, 2013). Örneğin, yarı iletken endüstrisinin hızlı büyümesi, sonuçta birkaç sistemin performansını artıran bütünleşmiş (entegre) devrelerin geniş çaplı kullanımıyla sonuçlandı. Dijital entegre devre (IC) kontrol cihazları tıbbi cihazların performansını iyileştirmek için kullanıldı. Bunların bir sonucu olarak programlanabilir kapı matrisi (FPGA) platformlarının kullanımını öğretmek amacıyla çok sayıda eğitim kursu ve laboratuvarı tasarlanmıştır (Chen, Lai ve ark., 2013).

## **1.2. Geleneksel Eğitimin Sorunları**

Öğrencilerin geleneksel eğitim yöntemlerine ilişkin problemleri temel olarak beş ana kategoriye ayrılabilir (Şekil 1.1).

- Anlayamama.
- Karmaşık projeler tasarlayamama.
- Kavramanın uzun zaman alması.
- Eğitim gelişiminin hızlandırılmaması.
- Laboratuvar da breadboard kullanımı.



Şekil 1.1. Geleneksel eğitimde karşılaşılan sorunlar.

Geleneksel eğitimde karşılaşılan sorunları daha ayrıntılarıyla ortaya koyacak olursak:

- (a) Öğrenciler genellikle birkaç bileşenli küçük bir devre kullanırken daha az problemlerle karşı karşıya kalırlar ya da devre daha karmaşık hale geldiğinde sık sık büyük sorunlar ortaya çıkar.
- (b) Birçok projede yüzlerce dijital mantık ve binlerce kapı ve devre bulunur; bu nedenle bunların geleneksel laboratuvarlarda takibi çok zor ve de yetersizdir. Ayrıca, teorik bilgiyi test etmek ve istenen çıktıyı sağlamak için mevcut karmaşıklığı korurken projeler tasarlamak zor bir iştir. Tipik bir endüstriyel problemin çözümü için farklı devrelerin birbirine bağlanması sıkıntılı bir süreç haline gelebilir.
- (c) Geleneksel eğitim, dersi zorlaştıran, çekici ve pratik olmayan bir yapıya sahiptir. Geleneksel eğitim, öğretmenin gerçek dünya projelerindeki en son gelişmeleri dersine uyarlaması durumunda, zaman ve para gibi iki önemli değişken, öğretmene ve kuruma bir yük olabilir.

- (d) Eğitim laboratuvarlarında genellikle breadboardlar kullanılır. Bazı durumlarda popüler bir teknik olan ancak parazit yaratan, görünürlük sorunları olan ve takibi zor, kötü temaslı olan tel sargılı breadboardların kullanımı hem öğretici hem de öğrenen açısından büyük problemler yaratabilmektedir.
- (e) Öğrenciler özellikle orta ve büyük ölçekli entegrasyon cihazlarını içeren karmaşık devrelerin geliştirilmesini ve hata ayıklanmasını zor bulmaktadırlar. Sonuç olarak, laboratuvar ortamında öğrenciler, gerçek dijital tasarıma kıyasla bağlantı hata ayıklama tekniklerine daha fazla odaklanırlar.
- (f) Projelerin kullanılabilirliği ve karmaşıklığı arasında bir çatışma durumu mevcuttur. Ayrıca, projenin süresi ve proje maliyeti de öğretim sürecini olumsuz olarak etkileyen faktörler arasındadır.
- (g) Öğrenciler, tasarımlarını donanımda uygulama veya çalıştırma şansı bulamazlar.
- (h) FPGA, uygulamada avantajlar sağlamakla birlikte, öğretmenlik yapan mühendislerin uygulamaları açısından sınırlıdır ve hala müfredatlarla ilgili sorunlar bulunmaktadır.

### **1.3. Araştırmanın Motivasyonu**

Bu çalışma, geleneksel, ML ve PBL gibi öğretim yöntemlerini inceleme ve karşılaştırmanın yanı sıra, öğrenci odaklı, daha karmaşık projelerde daha iyi sonuçlar elde etmeyi kolaylaştıran yeni bir öğretim yöntemi olarak proje tabanlı öğrenme ve mikro öğrenmenin gelişmiş bir kombinasyonunu ortaya koymaktadır.

### **1.4. Çalışmanın Amacı ve Önemi**

Geleneksel eğitim, öğrencilerin karmaşık projeleri anlama ve çözme taleplerini yerine getirememektedir. Bu noktadan hareketle, Bu çalışmanın önemini detaylandırarak olursak:

- Bu çalışmanın sonuçları, yükseköğretimin bugünü ve geleceğini geliştirmek için önemlidir. Bu çalışmanın, eğitimde yeni öğretim yöntemleri ve bilgiye ulaşma yollarını teşvik edebilecek yeni uç noktalar yaratması muhtemeldir.
- Bu çalışma, geleneksel eğitimdeki öğrenciler için zorluk oluşturan birçok sorunu çözmeyi amaçlamaktadır. Örneğin, öğrenciler kendilerine verilen konuları anlamamakta ve çoğu zaman müfredat istenilen sürede tamamlanamamaktadır. Bu nedenle, proje tabanlı öğrenme ve mikro öğrenme, geleneksel yöntemlerle ilgili bahsedilen problemlerle başa çıkmak için yeni öğretim araçları olarak kullanılmıştır.
- Bu çalışmanın bir diğer önemli amacı, gerçek dünyadaki eğitim ihtiyaçlarını karşılamak için geleneksel eğitim problemlerini çözmek ve gelişmelere ayak uydurmaktır.
- Bilgisayar mimarisi eğitiminde daha etkili, daha hızlı ve kaliteli bir öğrenme süreci sağlamak için yeni eğitim, yöntem ve teknolojiler test edilmiştir. Ek olarak, tasarım sürecinin üstesinden gelmek için öğrenme sürecini daha etkili, çekici, öğretici ve çözüm odaklı hale getirmeye odaklanılmıştır.
- Bu çalışmanın amaçlarına ulaşmak için, konunun müfredatını kapsayacak şekilde çekici ve öğretici projeler ve örnekler oluşturulmuştur. Geliştirilmiş öğretici ve çekici örnekler, etkili ve ilgi çekici bilgisayar mimarisi eğitimi için donanım ve yazılımı bir araya getirmiştir.
- Bu çalışma, benzer çalışmalar yürüten ve yeni değişkenler kullanan araştırmacılara fayda sağlayacaktır. Araştırmacıların, yeni fikirler ve müfredat geliştirme kavramlarını yeni bir bakış açısıyla, bir beyin fırtınası yapmaları için yeni yollar açacaktır.

### **1.5. Çalışmanın Hipotezi**

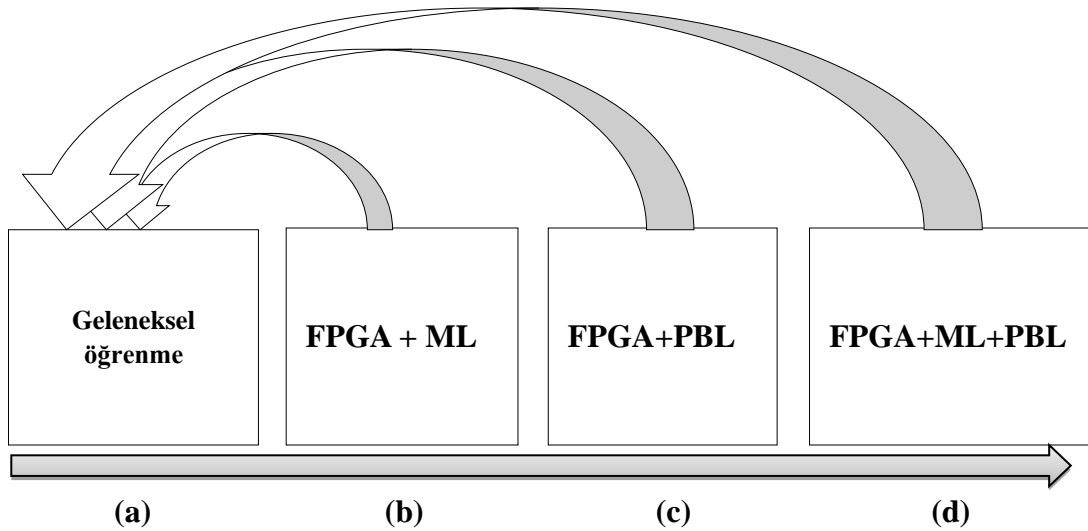
Bu çalışmanın ana hipotezi “FPGA destekli, PBL ve ML yöntemlerinin geliştirilmiş bir kombinasyonunun öğrenme sürecini yönetmesi, geleneksel öğrenme yöntemine kıyasla daha etkili, daha hızlı ve daha pratik bir öğrenme sunar.” şeklinde özetlenebilir.

FPGA teknolojisiyle birlikte PBL ve ML yöntemlerini kullanmak (projeler ve örneklerle) geleneksel öğrenme yöntemlerine kıyasla öğrenciler için daha öğretici ve çekici olmasının yanı sıra öğrencilerin daha karmaşık tasarım projelerine adaptasyonuna olanak sağlar. Bu tür projelerin uygulanması, öğrencilerin, gerçek hayattaki projelere benzer bir deneyim kazanmaları için çok önemlidir.

## 1.6. Çalışma Grupları ve Yaklaşımlar

Bu çalışma dört farklı öğrenci gurubuna uygulanmıştır (Şekil 1.2):

- *İlk grup*, uygulamalı FPGA teknolojilerini, PBL ve ML yöntemleri uygulanmaksızın bir bilgisayar mimarisi dersi aldı.
- *İkinci grup*, uygulamalı FPGA teknolojilerini ve ML yöntemini (PBL yöntemi uygulanmaksızın çekici ve öğretici örnekleri inceleyerek) kullandı.
- *Üçüncü grup*, uygulamalı FPGA teknolojilerini ve PBL yöntemini (ML yöntemini uygulamaksızın çekici ve öğretici bir proje inceleyerek) kullandı.
- *Dördüncü grup*, sözü edilen örnek ve projeleri gerçekleştirmek için FPGA teknolojisi kullanılarak, PBL ve ML yöntemlerinin bir kombinasyonunun uygulandığı bilgisayar mimarisi dersini aldı.



Şekil 1.2. Çalışma grupları.

## 1.7. Araştırma Metodolojisi

Bu araştırma, yaygın olarak uygulanan tanımlayıcı analitik yaklaşımı izlemiştir ve özellikle bu araştırma için tasarlanmış anketler aracılığıyla birincil kaynaklardan veri toplanmasına dayanan uygulamalı araştırma olarak sınıflandırılmıştır. Anketler, çalışılan örneğe/projeye göre hedeflenmiştir. Anketler her dönem sonunda dağıtılmış ve sonuçları incelenmiştir.

Çalışmanın pratik uygulaması şu prosedüre dayalı olarak gerçekleştirildi: Karmaşık bir proje tasarlanmış ve kullanışlı ve çekici örneklere ayrılmıştır. Bundan sonra, konunun tüm içeriğine PBL ve ML yöntemleri uygulanmıştır. Bu faaliyetler aşağıdaki gibi özetlenebilir:

- Birden fazla donanım ve yazılımı birleştiren bir proje geliştirmek.
- ALU (Aritmetik mantık birimi) ve CPU'yu VHDL ile tasarlama.
- FPGA teknolojisini kullanma.
- PBL ve ML eğitim yöntemlerinin bir kombinasyonunu kullanma.

Çalışmanın bir bölümünde, VHDL kullanılarak temel mantık kapıları ve bileşenleri dijital tasarım ele alınmıştır. Öğrencilerin bu yöntemi anlamaları ve kursun başında daha hızlı ve daha verimli olmaları için aşağıdaki adımlar izlenmiştir:

- Dersler, kolaydan karmaşığa doğru üç farklı karmaşıklık seviyesine ayrılmıştır.
- Üç saatlik bir süre boyunca bu ders için kullanılan yeni teknolojinin (yani FPGA'nın) ve öğretim yöntemlerinin (PBL ve ML) kullanımını gösteren bir sunum yapılmıştır.
- İlk haftanın iki saati içinde dijital CPU tasarımı ele alınmıştır.
- Deneysel tasarımın detayları, proje başlamadan önce gösterilmiş ve açıklanmıştır.

Çalışmanın sonuçları, önerilen ve uygulanan dört modelin, *öğretim faktörü*, *çekicilik (ilgi çekicilik) faktörü*, *geçerlik faktörü*, *yeterlik faktörü* ve *karmaşıklığa izin verme faktörü* açısından bir karşılaştırması yapılarak analiz edilmiştir.

## 1.8. Çalışmanın Özeti ve Önemi

Bu çalışmanın başarısı, dijital tasarımı öğrenmek için dört öğretim modelinin incelenmesine ve uygulanma biçimine dayanmaktadır.

- i. Geleneksel öğrenme modeli.
- ii. FPGA teknolojisi ile mikro öğrenme yöntemi.
- iii. FPGA teknolojisi ile proje tabanlı öğrenme yöntemi.
- iv. FPGA teknolojisi kullanılarak proje tabanlı mikro öğrenme yöntemi.

Bu dört öğretim modeli, karmaşık projelerin geliştirilmesinde daha başarılı ve kullanışlı olan yeni bir öğretim yöntemini ortaya koymak amacıyla, üst düzey bir bilgisayar mühendisliği dersinde dijital tasarımın öğretilmesi için uygulanmıştır.

## 1.9. Örneklem

Uygulamalar, Turgut-Özal Üniversitesi ve Kastamonu Üniversitesi'nde okuyan öğrenciler üzerinde gerçekleştirilmiştir. Söz konusu üniversitelerde sözü edilen öğretim yöntemlerinin uygulandığı toplam öğrencisi sayısı 162'dir. Bilgisayar Mühendisliği ve Elektronik Mühendisliği Bölümlerinin her birinden iki sınıf araştırmaya katılmıştır. Bu çalışmanın metodolojilerini uygulamak için dört grubun tümü test edilmiştir. Birinci grupta 46, ikinci grupta 43; üçüncü grupta 39, dördüncü grupta ise 34 öğrenci vardır. Çalışma, Nisan 2015 – Haziran 2018 tarihleri arasında gerçekleştirilmiştir.

## 1.10. Verilerin Toplanması ve Analizi

Bahsedilen teknikler ve öğretim yöntemleri dört yarıyıldan kullanılmıştır:

- ✓ 1. Yarıyıl: FPGA + ML+PBL kullanılmadan aritmetik mantık kayma ve dönme talimatlarının VHDL kullanılarak tasarlanması.
- ✓ 2. Yarıyıl: FPGA + ML kullanarak VHDL ile aritmetik mantık kayma ve dönme talimatlarının tasarlanması.
- ✓ 3. Yarıyıl: FPGA + PBL kullanarak VHDL ile CPU tasarlanması.

- ✓ 4. Yarıyıl: Aynı anda FPGA ve ML ve PBL kullanarak VHDL ile CPU tasarlanması.

Öğretim yöntemlerinin sonuçları aşağıdaki testlerle değerlendirilmiştir:

- i. *Anket*: Bu çalışma için özel bir anket tasarlanmıştır.
- ii. *Ödev*: İkinci grup ve dördüncü grupta üç seviyeye ayrılmış üç ev ödevi verilmiştir.
- iii. *Ara Sınav*: Dönemim ortalarında bir ara sınav yapılmış ve değerlendirilmiştir.
- iv. *Final Sınavı*: Dönem sonunda bir final sınavı yapılmış ve değerlendirilmiştir.

### 1.10.1. Anket

Bu çalışma kapsamında özel olarak tasarlanan anketler, mühendislik eğitiminin önemli yönlerini, FPGA teknolojisi, PBL ve ML'yi kullanarak incelemek amacıyla aşağıdaki sorulara odaklanmıştır:

- Yeni eğitim yöntemlerinin uygulanmasına yol açan faktörler nelerdir?
- Yeni eğitim yöntemlerini uygulamanın faydaları nelerdir?

Anket için aşağıdaki yargılara ait geri dönütler incelenmiştir:

- 1) Kurs, dijital elektronik gereksinimlerini karşılamak için yüksek performans elde etmeme yardımcı oldu.
- 2) Kurs, dijital elektroniği daha verimli ve hızlı bir şekilde öğrenmeme yardımcı oldu.
- 3) Kurs, bana proje tasarlama ve dijital elektroniği öğrenme konusunda yeterli güven verdi.
- 4) Kurs, bana devreleri ve karmaşık dijital sistemleri tasarlama yeteneği verdi.
- 5) Kurs, bana uzun vadede başarılı bir profesyonel olma fırsatı verdi.
- 6) Kurs, bana herhangi bir elektronik projede çalışabilme yeteneği kazandırdı.
- 7) Kurs, araştırma çalışmalarını yürütme yeteneğini ve tutumunu kazanmama yardımcı oldu.
- 8) Kurs, projeleri analiz etme ve tasarlama yeteneğimi geliştirmeme teşvik etti.

- 9) Kurs, tasarım arayüzünü çevre aygıtlarında kullanmama yardımcı oldu.
- 10) Kurs, bilgiyi pratikte uygulamama yardımcı oldu.
- 11) Kurs, bilgiyi ve zamanı yönetmeme yardımcı oldu.
- 12) Kurs, bağımsız olarak çalışabilme yeteneği edinmeme yardımcı oldu.
- 13) Kurs, başkalarıyla etkili iletişim kurmama ve takım çalışması yapmama yardımcı oldu.
- 14) Kurs, mühendislik problemlerini çözme yeteneği kazanmamda yardımcı oldu.
- 15) Kurs, disiplinlerarası takımların bir üyesi olarak çalışmamda bana yardımcı oldu.
- 16) Kurs, öğrenmeye odaklandı ve önemli kavramları anlamama yardımcı oldu.
- 17) Kurs, dijital tasarımı hızlı ve ayrıntılı bir şekilde öğrenmemi sağladı.
- 18) Kurs etkiliydi ve yenilikçi öğrenmeyi teşvik etti.
- 19) Kurs, öğretici ve çekiciydi.
- 20) Kurs, olumlu yönde, beklentilerimin ötesinde öğrenmeye çalışmamı sağladı.
- 21) Görevler ilginç ve zorluydu ve proje önemliydi. Elbette yaratıcı olmak için birçok fırsat sağladı.
- 22) Bu ders, öğrenilen kavramlarla maksimum pratik yapılmasını sağladı.
- 23) Kurs bana tasarım projeleri konusunda yeterince güven veriyor.
- 24) Kurstaki yazılım araçlarının kullanımı kolaydı.
- 25) Kurs, beni, çağdaş dijital tasarım araçlarına maruz bıraktı.
- 26) Kurs, bana, temel bileşenleri tasarlama ve bunları büyük projelerde kullanma yeteneği kazandırdı.
- 27) Kurs, programlanabilir cihazlardaki karmaşık dijital devreleri ve sistem tasarımlarını anlamamı sağladı.
- 28) Nasıl tasarım yapılacağını öğrenmek için yeterli kaynak vardı.
- 29) Çok iyi bir kurs projesi tasarladım.
- 30) Kurs projesinin deneyimi beni sektördeki işler için hazırladı.
- 31) Kurs projesindeki deneyimimden memnunum.
- 32) Kurs, artan karmaşıklıkla başa çıkma esnekliği kazandırdı.
- 33) Kurs projesi kursta öğrenilen tüm dersleri içeriyordu.
- 34) Kurs projesi aşamalı olarak kolaydan zora farklı karmaşıklık düzeylerinde gelişti.
- 35) Kursun sonunda, nispeten karmaşık projeler uygulayabiliyorum.

- 36) Kursun gerek dnya projelerinin tasarım karmaşıklığı aısından gereksinimleri karşıladığını dşnyorum.
- 37) Grev karmaşıklığına rağmen, ğrendiğim konuları çoğaltmak ve birleştirmek iin şansım oldu.
- 38) Gelişmiş ve karmaşık rneklerin geliştirilmesi, daha iyi bir ğrenme eğrisi olduğunu dođrulamıştır.
- 39) Bu projelerin karmaşıklığının gelecekteki karmaşık gerek projelerle baş etmek iin gereken deneyimi kazanmayı sağlayacağını dşnyorum.
- 40) Kurs sresince meslektaşlarımla etkin bir şekilde alıştım.
- 41) Verilen zaman izelgesinde kurs hedeflerine başarıyla ulaştım.
- 42) Kurs boyunca profesyonelce bir mhendis gibi dşnebildim.
- 43) Kurs tamamlandığında, fikirler daha da geliştirilebilir.

## 2. ÖNCEKİ ÇALIŞMALAR VE KURAMSAL ÇERÇEVE

### 2.1. Mikro Öğrenme (ML)

Öğrenme ortamlarının çoğunda, öğrencilere çok çeşitli faydalar sağlayabilecek "mikro öğrenme" terimi kullanılmaktadır. Bu kullanım, öğrencilere mikro öğrenme yoluyla mümkün olan bilgi ve becerileri sağlayabilmesinden kaynaklanmaktadır. Bu terim, hızlı bir şekilde yükseköğretimdeki en önemli trendlerden biri haline gelmiştir (Dessi, Fenu ve ark., 2019).

Zaman baskısı ve motivasyon eksikliği genellikle ciddi öğrenme sorunları olarak kabul edilir. Bu problemlerle başa çıkmak için eğitim yaklaşımları elektronik cihazların yardımıyla geliştirilmelidir. Bilişim teknolojileri (BT) kullanımı şimdilerde eğitim ve mesleki eğitimin önemli bir parçası haline gelmiştir (Gstrein ve Hug, 2006). Ayrıca, öğrenme yöntemi, esas olarak imalat ve iş ihtiyaçlarına göre değiştirilmiş olan müfredatları içermektedir ve tartışılan zorluklar iç ve dış şirket ortamları ile eşleşmektedir (Job ve Ogalo, 2012).

Mikro öğrenme, müfredatı takip etme açısından en iyi sonucu veren daha küçük adımlarla öğrenme metodolojisini izler. Mikro öğrenme etkinlikleri tipik olarak öğrencilere gerekli bilgileri sağlamak için tasarlanan kısa vadeli dersleri, projeleri veya kursları içerir. Örneğin, bir öğrenciye aynı anda geniş bir konuyu öğretmeye çalışmak yerine, ders, daha küçük parçalara veya küçük proje bölümlerine ayrılır.

Mikro-öğrenmenin iyi bir yönü, öğrencilerin neyi ve ne zaman öğrendiklerinin kontrolünü hissetmeleridir (Charteris, Smardon ve ark., 2017). Mikro öğrenme alıştırmaları, en iyi şekilde, öğrencinin doğrudan bir bilgiye ihtiyaç duyduğu noktada gerçekleştirilir çünkü öğrenci bu bilgiye daha duyarlıdır. Mikro öğrenme öğrencilere, küçük ve çok özel hamlelerle içerik öğretme ve sunmanın bir yoludur (Gross, Rusin ve ark., 2019). Örneğin, iki sayının nasıl toplandığına dair çevrimiçi bir video, gerçek hayatta mikro öğrenme alıştırmalarının ideal örneği olarak görülmektedir.

### **2.1.1. Faktörlerin ML ile İlişkisi**

Bu kısımda mikro öğrenmenin bazı faktörler açısından değerlendirmesi ele alınacaktır.

#### **2.1.1.1. Öğretim Faktörü - ML İlişkisi**

Mikro öğrenme, öğrenci etkileşimini geliştirir; işbirliği ve anlayışı, öğrenci merkezli öğrenmeyi, çok boyutlu öğrenmeye olan ilgiyi ve genel öğrenme ortamını önemli ölçüde iyileştirir (Liu, Wei ve ark.,2016).

Üniversite araştırmacıları, öğrencilerden konusunu öğrenmeye yüksek ilgi gösterdiklerinde olumlu geribildirim almışlardır. Oysaki gelişen dünya, verilen konuların içerik miktarının önceki yıllara göre artmasına ve geri dönütlerin de önceki yıllara göre zayıflamasına neden olmaktadır (Zhamanov ve Zhamapor, 2013).

Mikro öğrenme, öğrencilere belirli küçük bölümlerle içerik sağlayan bir süreçtir. Mikro öğrenme, mikro içeriğe dayanır. Mikro öğrenme, daha küçük adımlarla öğrenmeyi içerir ve geleneksel öğrenme süreçleriyle birlikte kullanılır. Mikro öğrenme, öğrencilere alt blok bileşenleriyle başa çıkma, her bölüm için modüller oluşturma ve modüllerin eğitsel anlamda üretkenliğini artırmalarında yardımcı olur (Said ve Çavuş, 2018). Yeni mühendislik eğitimi metodolojileri uygulandığında bazı araştırmacılar kişiselleştirilmiş öğrenme içeriği oluşturmak için bilgi boşluklarını dolduran ve daha fazla esneklik kazanmalarını sağlayan cesaret verici sonuçlar bulmuşlardır (Kovachev, Cao ve ark., 2011)..

#### **2.1.1.2. Çekicilik Faktörü - ML İlişkisi**

Mikro öğrenme, öğrenme kavramları ve modelleri dahil olmak üzere, çeşitli fenomenlerin mikro yönlerine odaklanarak birçok yolla anlaşılabilir (Dillenbourg, 1999). Mikro öğrenme, heyecan verici olabilecek ilginç veya faydalı bilgiler de sağlayabilir (Gassler ve ark., 2004). Öğrenme süreçleri gözlem ve veri toplamaya dayandığından, mikro öğrenme, yeni bir bilgi yaratmaya yardımcı nitelik taşıır (Giuliani ve Bell, 2005).

Mikro öğrenme, içeriği sınıf dışında paylaşmak için çekici öğrenme yöntemleri bulan etkili bir öğrenme yöntemidir. Bu, hızlı tempolu öğrenme ile mikro eğitim arasındaki ilişkiyi daha da güçlendirir ve genel anlamda bir farkındalık yaratır (Job ve Ogalo, 2012).

#### ***2.1.1.3. Geçerlik Faktörü - ML İlişkisi***

Mikro öğrenme, mikro içerikten doğar; bu, öğrencilerin öğrenmelerini geliştirmek için etkin biçimde kullanılan çok az dijital bilgi anlamına gelir. Öğrenme çıktısı, öğrenciler arasındaki eğitimsel etkileşime de bağlıdır (Bonwell ve Eison, 1991).

#### ***2.1.1.4. Yeterlik Faktörü - ML İlişkisi***

Mikro öğrenme teknolojisi, öğrencilere gerçek hayattaki çalışma ortamlarındaki performanslarını artıran alaka düzeyi, özerklik ve yeterlilik gibi temel ihtiyaçları sağlar (S.A. Nikou, 2018).

#### ***2.1.1.5. Karmaşıklığa İzin Verme Faktörü - ML İlişkisi***

Temel mikro eğitim ilkesi, bir projenin küçük birimlere bölünmesi ve öğrenmenin küçük adımlarla yapılması halinde, öğrenmenin hem verimlilik hem de zorlukla başa çıkma açısından geliştirilebileceğini öngörür (Zhen-Ting ve Jiang-zhao, 2010). Birçok çalışma, mikro eğitim yönteminin öğrencilerin karmaşık projeler geliştirmelerine yardımcı olacak yüksek özgüven kazanmalarına yardımcı olduğunu ortaya koymaktadır (Esteves, 2009). Öğrencilerin proje süreçlerinde karmaşıklığı arttırması önemlidir; bu, öğrenme sürecini uygun öğretim teknikleriyle kullanmanın bir sonucudur. Birçok araştırma, öğrencilerin küçük bir eğitimle yüksek düzeyde öğrenme elde ettiğini göstermektedir (Sun, Tsai ve ark., 2008).

Bazı öğrenci araştırmacılar, proje geliştirmedeki etkinlik ve karmaşıklık arasındaki bağlantıyı incelediklerinde, bilgi boşluklarını doldurabilecekleri ve iyi bir eğitim içeriği oluşturmak için daha fazla esneklik kazanacakları sonucuna varmışlardır (Vidal ve Marle, 2008).

### **2.1.2. Mikro Öğrenmenin Avantajları**

Mikro öğrenme, öğrencilere daha küçük formlarda bilgi toplama yeteneği verir ve bu da onların ilgili konuyu daha etkili bir şekilde almalarına yardımcı olur. Mikro öğrenme, öğrencilerin,

- Bağımsız ve birbirine bağlı öğrenme becerilerini geliştirir.
- Problem çözme, eleştirel düşünme ve yaratıcı düşünme becerilerini geliştirir.
- Öz değerlendirme becerilerini geliştirir.
- Bütünleşik ve geniş açılı düşünme becerilerini geliştirir.
- Mühendislik projelerinde bilgiyi uygulama becerilerini geliştirir.
- Deney tasarlama, deney yapma, veri analizi ve yorumlama becerilerini geliştirir.

### **2.2. Proje Tabanlı Öğrenme (PBL)**

Proje tabanlı öğrenme, öğrencilerin karmaşık bir problemi araştırmak ve sorgulamak için uzun bir süre boyunca bilgi ve beceri kazanmalarına yardımcı olan bir öğretim yöntemidir (Blumenfeld, Soloway ve ark.,1991).

Sorunlar ve zorluklar daha derin bilgi edinmeyi gerektirir. Öğrencilerin sınıfta öğrenmelerini çevrelerindeki dünyaya bağlarken derin düşünmeye ve bağımsız olarak öğrenmeye teşvik eder. Tüm öğrenciler için aynı kurallar verildiğinde, istenen sonuçları elde etmek için kendi benzersiz fikirlerini, tasarımlarını ve seçimlerini kullanabilirler (Krajcik ve Blumenfeld, 2006).

#### **2.2.1. Faktörlerin PBL ile İlişkisi**

PBL, zaman içerisinde mühendislik eğitimi topluluğunun büyük ilgisini çekmiştir. PBL'nin en büyük yararı, öğrenme sürecine öğrencinin katılımının artırılmasıdır (Sokiç ve Ahık-Djokiç, 2008). Araştırma sonuçları, bu öğrencilerin geleneksel öğretim yöntemleriyle eğitim almış öğrencilerden daha başarılı olduklarını göstermektedir (Baş ve Beyhab, 2017).

Bu kısımda PBL ile faktörler arasındaki ilişki üzerinde kısaca durulacaktır.

#### ***2.2.1.1. Öğretim Faktörü - PBL İlişkisi***

PBL eğitim teorisi temel olarak öğrencilerin bilişsel becerilerinin öğrenme ve problem çözme için nasıl kullanılacağı ile ilgilidir (Chandler ve Sweller, 1992). PBL yoluyla öğrenme, öğrenciler arasında işbirlikli öğrenme ortamı yarattığı için öğrencilerin iletişim yeteneklerini de ayrıca geliştirir (Tse ve Chan, 2003). PBL, öğrencilerin düşünme yetkinliğini doğrudan arttırdığından ve esnek bir öğrenme ortamı yarattığından dolayı oldukça popüler bir yaklaşımdır (Doppelt, 2003).

#### ***2.2.1.2. Çekicilik Faktörü - PBL İlişkisi***

PBL yönteminde, son ürün akılda tutularak öğrenim sürecine başlanılır. Projeyi çekici kılan özel bilgiler, temel bilgiler ve anahtar kavramların öncelikle belirlenmesi gereklidir. Bu, kavramları ve istenen bilgileri daha verimli öğrenme ve anlama yolunu açar (Blumenfeld, Soloway ve ark., 1991).

#### ***2.2.1.3. Geçerlik Faktörü - PBL İlişkisi***

Proje temelli öğrenme, hedef belirleme, proje yönetimi becerileri, izleme, proje danışmanlığı ve geri bildirim gibi bazı aktiviteler etkili bir şekilde yapıldığında başarılı olur (English ve Kitsantas, 2013).

#### ***2.2.1.4. Yeterlik Faktörü - PBL İlişkisi***

Proje tabanlı öğrenme, derin öğrenmeye ihtiyaç duyar çünkü gerçek problemleri çözme, karar verme ve sorgulama gibi becerilere dayanan önemli zorluklarla ilgilenir (Thomas, 2000).

### **2.2.1.5. Karmaşıklığa İzin Verme Faktörü – PBL İlişkisi**

Proje tabanlı eğitim, mühendislik eğitimi dünyasında araştırmacıların ve uzmanların dikkatini çekmektedir. Proje yönetimi gelişmeleri, araştırmacıların önemli olan belirli değerlendirme ve proje karmaşıklığı yönetimi konularını ele almalarını sağlamıştır (Qureshi ve Kang, 2015). PBL, öğrencilerin kendi kendine öğrenmesini ve aktif öğrenmeyi teşvik ederek onların genel öğrenme sürecine katılımını artırır.

Ayrıca, farklı öğrenme yöntemleri altında çalışırken iletişim becerilerini geliştirir ve karmaşık problemleri çözmek için düşünme becerilerini daha üst seviyelere taşır. PBL, bir mühendisin sahip olması gerektiğı düşünölen kişisel becerilerin kazandırılmasında pozitif etkilere sahiptir (Hadim ve Esche, 2002). PBL, öğrencilere yükseköğrenime devam ederken derin kavramlar kazanma fırsatları sağlar. Bu, onların karmaşık problemleri daha hızlı, stressiz ve daha verimli olacak şekilde çözmelerine olanak verir (Hmelo-Silver, 2004).

### **2.2.2. Proje Tabanlı Öğrenmenin Avantajları**

PBL yöntemi aşağıdaki avantajlara sahiptir:

- Öğrencilerin farklı konulara ilgisinin artmasını sağlar.
- Projelerle, ele alınan konunun özömsenmesi sağlanır.
- Pratik mühendislik projeleri tasarlayarak öğrenme mühendisliğı ilkelerini daha kolay ve daha eğlenceli hale getirir.
- Kişisel becerileri ve takım çalışmasını teşvik eder.
- Sistem, süreç veya bileşen tasarımı yeteneklerini geliştirir.
- Öğrencilerin çok disiplinli takımlarda çalışma becerilerini geliştirir.
- Öğrencilerin mühendislik sorunlarını tanımlamasını, formüle etmesini ve çözmelerini sağlar.
- Öğrencilerin mühendislik becerilerini artırır; temel mühendislik araçlarının kullanımının gelişmesini ve modern tekniklerin öğrenimini sağlar.

### **2.3. Geleneksel Öğrenme**

Geleneksel öğrenme yöntemi, doğrudan öğretmen-öğrenci etkileşiminin olduğu bir sınıfta öğrenmeye dayanır ve öğretmen bir bilgi kaynağı olarak hareket eder. Geleneksel öğretim yöntemleri genellikle öğrencilere veya derslere dayanırken, öğrenciler sadece pasif dinleyicilerdir (Davcev, Stojkoska ve ark., 2016). Bununla birlikte başarılı öğrenmenin önemli bir ihtiyacı ise düzenli olarak öğrenme etkinlikleri yapmaktır. Öğrenciler genellikle gelişmiş sistemleri kullanacak kadar motive olmadıklarında kendilerini geliştirmek için modern öğrenmeden kaçınırlar (Gassler ve ark., 2004) .

Geleneksel öğretim yöntemleri, mühendislik mezunlarının mesleki taleplerini karşılamak için gereken temel beceri seti, bilgi ve tutumları kazandırmada yeterli gelmemektedir. Kapsamlı bir şekilde test edilmiş alternatif yöntemler, mühendislik eğitimi açısından daha iyi sonuçlar vermektedir (Rugarcia, Felder ve ark., 2000). Örneğin, mantıksal simülatörler, dijital sistem tasarım eğitimi ve araştırması için yaygın olarak uygulanmıştır. Programlanabilir cihazlar, karmaşık dijital sistemlerin ve devrelerin simüle edilmesine yardımcı olur. Emülatör üzerindeki temel devrenin kullanılması, geleneksel dijital model simülasyonundan ziyade yeni öğretim metodolojilerinde gerçekleştirilmektedir. Bu işlem, FPGA cihaz tedarikçileri tarafından sağlanan çok sayıda ticari programlanabilir yazılım geliştirme kartıyla kolaylaştırılmıştır (Trost ve Zajc, 2011). Her ne kadar geleneksel öğretim yöntemlerinin etkisiz olduğunu açıkça ortaya koyan bir çok çalışma olsa da ve her ne kadar öğrenme üzerine yenilikçi bir çok araştırma ve uygulamalar mevcutsa da, mühendislik eğitimi için baskın pedagoji sadece “tebeşir ve konuşma”dır (Mills ve Treagust, 2003).

### **2.4. Alanda Programlanabilir Kapı Dizisi (FPGA)**

Bir FPGA, karmaşık tasarımları tek bir IC'ye entegre eden yüzlerce RAM ve mantık kapısından oluşur. İki boyutlu flip-flop dizileri ve bağlı programlanabilir mantık blokları vardır (Trimberger, 1993). Hassas, hızlı ve tekrarlanabilir bir teknolojidir. GPU (grafik işlemci birimi) ve CPU'nun güç tüketimi açısından daha iyi performans

göstermesine yardımcı olur (Svab, Krajnik ve ark., 2009). FPGA'lar eğitimde farklı uygulamaları olan çeşitli karmaşık elektronik sistemlerin temelini oluşturur (Quintans, Valdes ve ark., 2005). FPGA mantıksal işlevleri uygulamak için kullanılır, fonksiyonelliği güncellenebilir ve birçok uygulama için avantajlara sahiptir (Panda, Rajput ve ark., 2012). Ucuz, güçlü ve uyarlanabilirlerdir çünkü konfigürasyonları soyut bir donanım tanımlama dilinde özelleştirilmiştir. FPGA'nın birçok avantajlarının yanı sıra, gelişmiş işlemcili mimari oluşturmak için kullanılabilmesini gösteren çalışmalar da vardır (Joost ve Salomon, 2005).

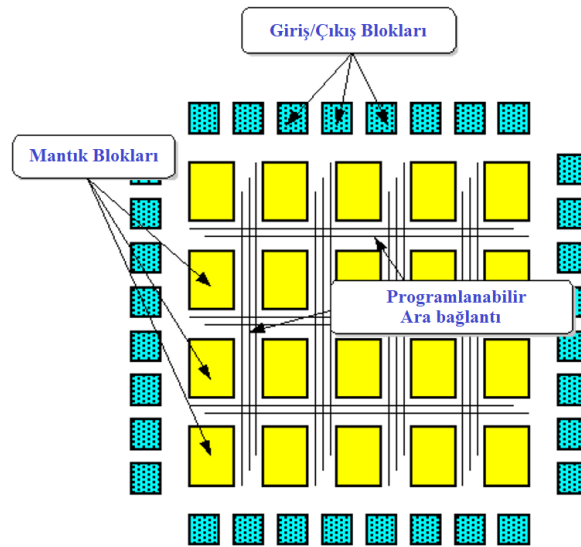
Kontrol uygulamaları, yüksek hızlı kısıtlamaları ve FPGA gibi yüksek yoğunluklu programlanabilir mantık cihazlarını arayüzlemek için gerçek zamanlı bir işlem gerektiren fiziksel sistemlerden faydalanır (Monmasson ve Cirstea, 2007). FPGA, kullanıcının ürün özelliklerini ve işlevlerini programlamasına, donanımı yeniden yapılandırmasına ve alana ürün monte edildiğinde bile yeni standartlara adapte olmasına izin veren programlanabilir yarı iletken bir cihazdır; bu nedenle "alanda programlanabilir" olarak adlandırılır (Sulaiman, Obaid ve ark., 2009).

FPGA'lar dijital elektronik tasarımlarında ve prototip projelerde yaygın olarak kullanılır. Yeniden yapılandırma ve serbest simülator (free-simulator) programlanabilirlik özellikleri de önemli avantajlar sağlar. FPGA'lar mühendislik uygulamalarına ek olarak üniversitelerin birçok elektronik ve bilgisayar mühendisliği bölümlerinde de kullanılmaktadır. FPGA elektronik/bilgisayar mühendisliği eğitimi için önemlidir ve FPGA kullanımı, hem zamandan hem de paradan tasarruf sağlayarak tasarım karmaşıklığının artmasına izin verir (Trimberger, 1993).

#### **2.4.1. FPGA'nın Yapısı**

FPGA mimarisi iki bileşene ayrılmıştır: Mantık bloğu mimarileri ve yönlendirme mimarileri. Mantık blokları, ayrıntı derecelerine göre sınıflandırılır ve piyasada bulunan FPGA'larda birkaç mantık bloğu kullanılır (Rose, El Gamal ve ark., 1993). FPGA'lar, her bir mantık bloğunu diğer mantık bloklarına bağlayan, programlanabilir, birbirine bağlı bir yapı üzerinden bağlanabilen mantık bloklarını içerir (Young, Chaudhary ve ark., 1999).

Bir FPGA, yerel ve ekspres veriyolu hatları ağı ve bir dizi sıfırlama hattı ile karakterize edilen I/O pedleri ile birbirine bağlanabilen, programlanabilir mantık hücrelerinin bir satır ve sütun matrisine sahiptir. Bu satırlar ana sıfırlama hatları (main reset lines), sütun sıfırlama hatları (column reset lines) ve sektör sıfırlama hatlarını (sector reset lines) içerir. Ana sıfırlama hatlarının her biri farklı bir sıfırlama sinyali alır. Sütun sıfırlama hatlarının her biri, matrisin belirli bir mantık hücreleri sütunu ile ilişkilidir. Sıfırlama hatlarının (lines) her sütunu, seçilen bir sıfırlama sinyali alan ana sıfırlama hatlarından herhangi birine isteğe bağlı olarak bağlanabilir. Sektör sıfırlama hatlarının her biri, Şekil 2.1'de gösterildiği gibi, bir sütundaki mantık hücrelerinin bir alt kümesine bağlanır. Sütun sıfırlama hatları, ilgili sütunlardaki mantık hücrelerine, sütun sıfırlama hatlarına bağlanabilir olan sektör sıfırlama hatlarının yardımı ile seçici bir şekilde bağlanabilir (Furtek, Mason ve ark., 2001).



Şekil 2.1. FPGA'nın temel yapısı.

## 2.5. Quartus Yazılımı

Bu çalışma kapsamında Quartur II yazılımı kullanılmıştır. Quartus II yazılımı, derleme ve bindirme (fitting) sonuçlarının analizine yardımcı birçok araca sahiptir ve bir tasarımın cihazda ve sistemde çalıştığı gibi incelenmesine olanak tanır. Quartus yazılımı, tasarım doğrulama yeteneğine ek olarak bir tasarımın işlevsel işlemlerini ve iç zamanlamasını test edebilir. Quartus simülatörü, tek veya çok cihazlı proje modellemesi üzerinde esneklik ve kontrol sağlamaya olanak tanır. Simülatör, bir

proje için fonksiyonel, zamanlama veya çoklu cihaz simülasyonu gerçekleştirmek için derleme sırasında oluşturulan simülasyon ağ listelerini kullanır (Hamblen, Hall ve ark., 2006).

Quartus simülatörü, bir projenin sentezlenmeden önce mantıksal çalışmasını test etmek için işlevsel simülasyonu destekler; bu, tasarımcının mantıksal hataları hızlıca tanımlamasını ve düzeltmesini sağlar. Quartus II dalga biçimi düzenleyicisi, işlevsel simülasyonun sonuçlarını görüntüler ve birleştirme işlevleri de dahil olmak üzere projedeki tüm düğümlere kolay erişim sağlar. Quartus, tasarımcıya aynı anda birkaç cihazı simüle etme imkanı verir ve farklı Altera cihaz ailelerinin cihazları aynı projede kullanılabilir.

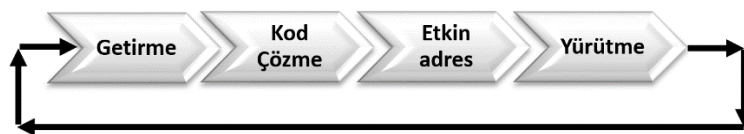
## 2.6. CPU Mimarisi ve Organizasyonu

Bilgisayar programları, bellekte depolanan talimat dizisinden oluşur. CPU, programın her bir talimatını önceden tanımlanmış bir adıma göre yürütür. CPU'nun kontrol ünitesi bu tanımlanmış adımları talimat formatına ve bilgisayar mimarisine göre belirler (Carpinelli, 2000). Aşağıda bazı temel bilgiler verilmiştir:

**Komut kodları:** Komut kodu, bilgisayara belirli bir işlemi gerçekleştirmesini söyleyen bir bit grubudur; üç bölüme ayrılmıştır (Carpinelli, 2000).

- *Operasyon Kodu:* ADD, SUB, vb. Bit sayısı, işlem sayısını belirler, örneğin, 4 bitlik boyut 16 Opcod üretir.
- *Adres Kodu:* İki işlenenin (operand) adresi.
- *Adres Değiştirme Kodu:* İşlenenlerin depolandığı konum.

**Bilgisayar döngüsünün genel formu:** Komut döngüsünün genel bir yapısı Şekil 2.2'de gösterilmiştir.



Şekil 2.2. Komut döngüsü diyagramı.

- Fetch (Getir): Bellekten bir talimat okur.
- Decode (Kod Çözme): Talimatın kodunu çözer; operasyon kodunu ve operand adresini edinir.
- Effective Address (Geçerli Adres): İşlenenlerin adresini bulur.
- Execute (Yürütme): Gerekli işlemi gerçekleştirir ve sonucu hafızaya veya kayıt defterine kaydeder.

### 3. MATERYAL VE METOD

Bu bölümde farklı öğrenme modelleri/yöntemleri, çalışmanın kapsamı, modelleri uygulamak ve test etmek için izlenen prosedürler ve çalışılan değişkenler arasındaki ilişki ortaya konulacaktır.

Bu çalışma kapsamında aşağıdaki öğrenme yöntemleri/modelleri kullanılmıştır:

- i. Geleneksel öğrenme yöntemi (TR).
- ii. Mikro öğrenme yöntemi (ML).
- iii. Proje tabanlı öğrenme yöntemi (PBL).
- iv. Proje tabanlı öğrenme ve mikro öğrenme yöntemlerinin birleşimi (ML+PBL).

Ayrıca, VHDL kullanılarak CPU tasarımının nasıl yapıldığı kısaca özetlendi. FPGA'nın yapısı ve FPGA simülasyon programı VHDL yardımıyla CPU tasarımı için kullanılan grafik arayüz yazılımı Quartus II hakkında da kısa bir bilgi verildi.

Bu çalışmada, FPGA teknolojisi ile PBL ve ML yöntemlerinin bir kombinasyonu kullanılarak öğrencilerin karmaşık projeleri kısa sürede başarıyla gerçekleştirme becerilerinin kazandırılması amaçlanmıştır. CPU tasarımı, PBL ve ML yöntemlerinin kombinasyonunu kullanarak, karmaşık projeler gerçekleştirmeyi mümkün kılan VHDL kullanılarak gerçekleştirmiştir.

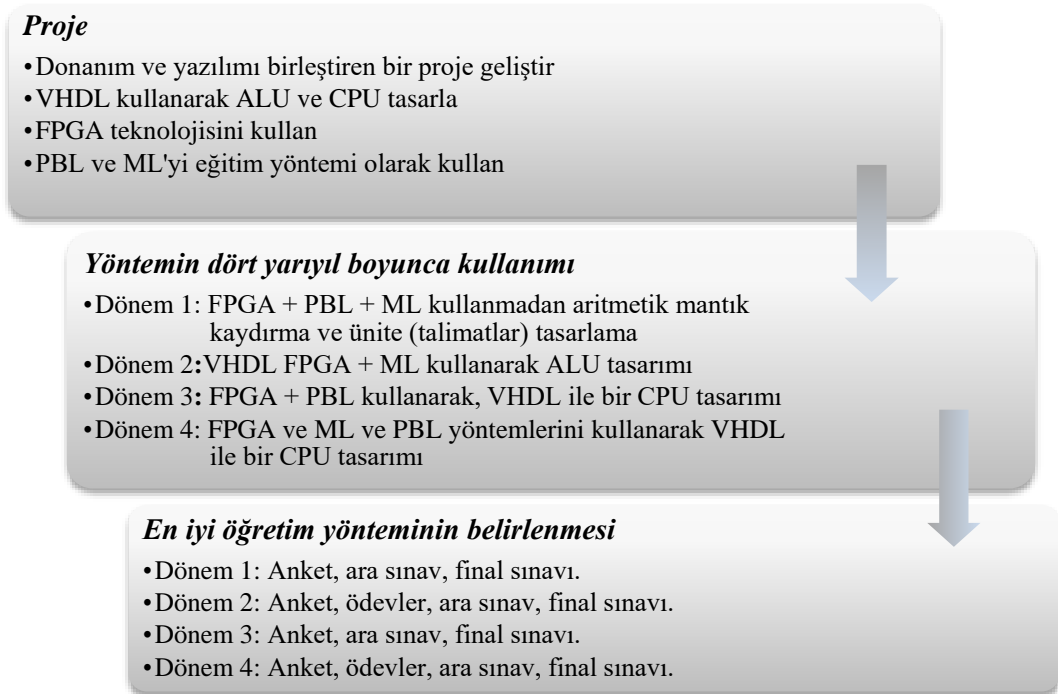
Söz konusu PBL ve ML yöntemleri, öğrencilerin pratik projelere ilişkin problemleri, özellikle de teorik olarak çözülemeyen problemleri çözmelerini sağlar. Ayrıca, bu yöntemler öğrencilerin karmaşık tasarım problemlerini ele alma ve çözme, istenen sonuçları elde etmelerine ve değiştirilebilir ya da esnek tasarım sağlamalarına yardımcı olma becerilerini geliştirir. Bu noktadan hareketle bu yöntemlerin etkin bir kombinasyonu ve yardımcı araçlarla öğrenim sürecinin verimliliğinin artırılması hedeflenmiştir.

Turgut Özal Üniversitesi ve Kastamonu Üniversitesi'nde uygulaması yapılan bu çalışmada, PBL ve ML yöntemlerinin öneminin vurgulanmasının yanı sıra, PBL ve

ML yöntemlerinin geleneksel öğrenme yöntemiyle olan kombinasyonlarının öğrenme çıktıları üzerindeki verimliliği de incelenmiştir. Araştırma anketlerinden veri toplamak için nicel araştırma metodolojisi kullanılmıştır.

Çalışmanın pratik uygulaması şu şekilde gerçekleştirildi: Karmaşık bir proje tasarlandı ve faydalı ve ilgi çekici örnekler yardımıyla bir öğretim planı yapıldı. Bundan sonra, tüm konu içeriğine PBL ve ML yöntemleri uygulandı.

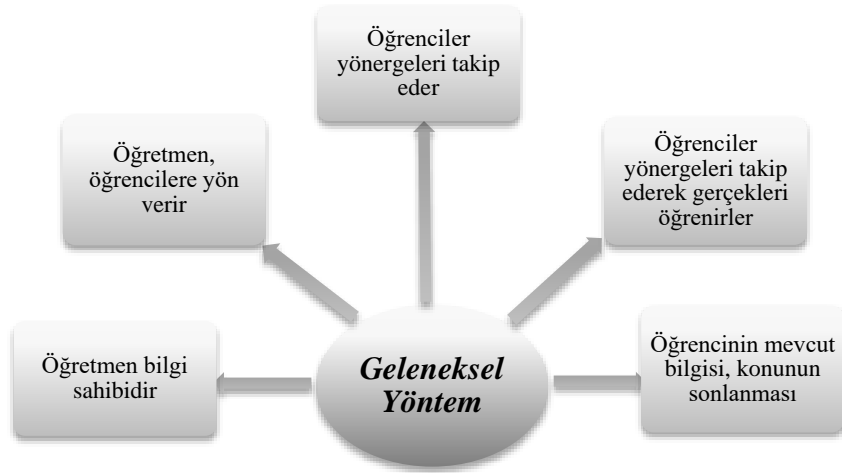
Bu yaklaşımın organizasyon planı Şekil 3.1'de belirtilmiştir.



Şekil 3.1. Öğretim yöntemlerinde veri toplama planı.

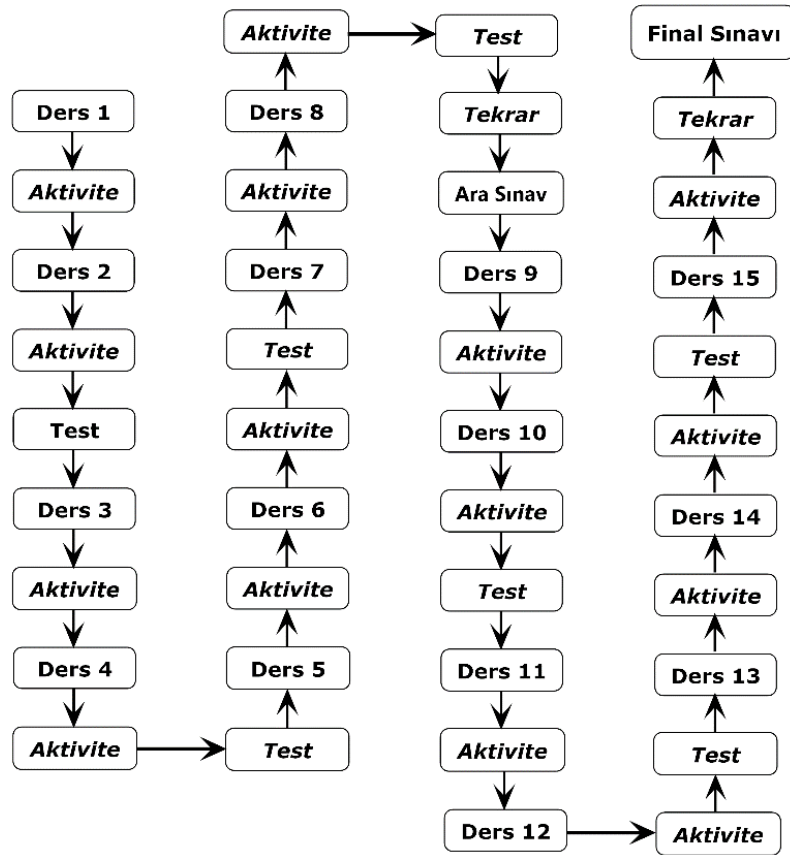
### 3.1. Geleneksel Öğrenme

Geleneksel öğrenme süreci, öğretmenin öğrenciler için bilgi kaynağı olarak hareket ettiği öğrenci-öğretmen etkileşimi yoluyla sınıf içinde öğrenmeye dayanır; Şekil 3.2'den de anlaşılacağı gibi, öğrenme işlemi tamamen öğretmen ve tahtaya bağlıdır.



Şekil 3.2. Geleneksel öğrenme yöntemi.

FPGA teknolojisi, ML ve PBL öğrenme yönteminin bir uyumlu kombinasyonu kullanılmaksızın gerçekleştirilen bir öğrenmede sorunlarla karşılaşılacaktır. Karşılaşılan öğrenme sorunları ilerleyen kesimde daha detaylı olarak ele alınacaktır. Geleneksel öğrenme için bir dönem çizelgesi Şekil 3.3'te verilmiştir.



Şekil 3.3. Geleneksel öğrenim gören birinci grup için öğretim planı.

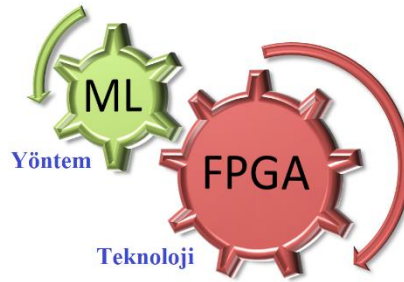
### 3.2. Mikro Öğrenme (ML)

Mikro öğrenmenin önemli özelliklerinden bazıları aşağıda verilmiştir:

- Süre çok kısa ve belirlenebilirdir.
- İçerik çok küçük (mikro içerikli) ve basit bir şekilde düzenlenmiştir.
- İçerik, müfredatın küçük bir parçasıdır.
- Süreç, bir dizi seviyeye bölünmüş ayrı bir faaliyettir.
- Kurstan önce yazılı belgeler ve elektronik ortamlar bulunur.
- Süreci açıklayan kısa süreli video sağlanmalıdır.

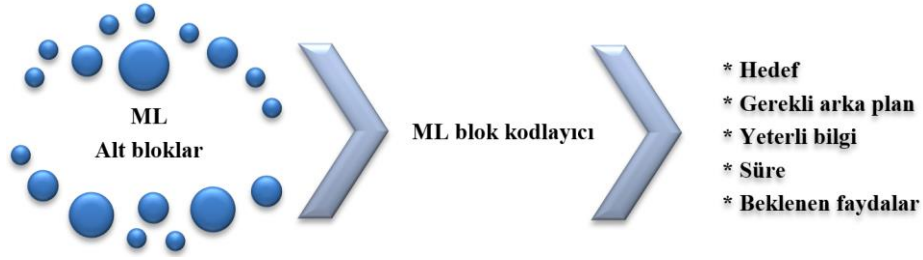
Mikro öğrenme süreci ile öğrenen öğrenciler, alt blok bileşenleriyle belirli zaman aralıklarında etkileşimli olarak karşı karşıya kalırlar. ML, öğrencilere küçük öğrenme birimlerinin yardımıyla ve kısa süreli öğrenme etkinlikleri oluşturarak öğretmeye odaklanır. Bilgisayar mimarisi kursu planlarına daha küçük alt bloklara bölerek daha uzun örnekler ve ödevler eklemek önemlidir.

Kanımızca, derste iyi bir proje üretmek için FPGA teknolojisi ve mikro öğrenme yöntemi birleştirilmelidir. Dolayısıyla, Şekil 3.4'te gösterildiği gibi, öğrencilerin karmaşık tasarım sorunlarını çözme becerileri geliştirilecektir.



Şekil 3.4. FPGA - ML etkileşimi.

Örneğin, aritmetik mantıksal kaydırma ve döndürme talimatları (ALUSHIFT) tasarımını öğretilmek isteniyorsa, Şekil 3.5'e gösterildiği gibi yeterli bilgiyi içeren kısa bir giriş yapılmalıdır.



Şekil 3.5. ML öğretim kademeleri.

Başlangıç olarak, VHDL'yi öğretmek için yeni ve özel olarak tasarlanmış bir örnek hazırlandı. Örnek, kursun tüm konularını kullanmak için hayali bir ALUSHIFT ile ilişkilendirildi. Bu noktada aşağıdaki sorular dikkate alınmıştır:

- ALUSHIFT ünitesinin tasarlanması ve uygulanmasının amacı nedir?
- ALUSHIFT ünitesi çalışmanın gerekli arka planı nedir?
- Hangi tür bilgiler yeterli sayılmalıdır? Bir örnekle açıklayın?
- VHDL ile uygun tasarım yöntemi nedir?
- Öğrencilerden beklenen sorular nelerdir?
- Bu konuyu öğretmek için gereken zaman nedir?
- Öğrencilerin not etmesi gereken belirli şeyler nelerdir?

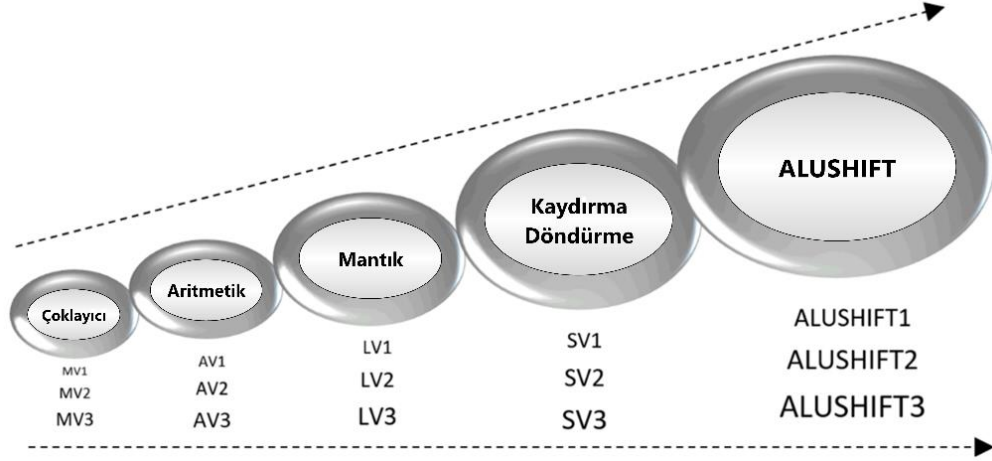
Öğrencilerin verilen bilgiyi anlamalarını geliştirmek için kullanılan adımlar (V1, V2, V3), her ders için Tablo 3.1'de gösterildiği gibi belirlenmiştir.

Tablo 3.1. Ödev seviyeleri.

Kod	Kod adı	Tanım
V1	Birinci versiyon	Ödev I, ders boyunca aynı konu öğretildi.
V2	İkinci versiyon	Ödev II, ara dönem projesi.
V3	Üçüncü versiyon	Ödev III, karmaşık proje.

ALUSHIFT tasarımına başlarken ilk adım, projenin karmaşıklığı ile başa çıkmak için projeyi birçok parçaya bölmektir. Bu, Şekil 3.6'da gösterildiği gibi, öğrencilerin anlama ve nihai proje tasarımını gerçekleştirmesine yardımcı olan ML yöntemi kullanılarak gerçekleştirilmiştir. ML kullanırken, nihai tasarıma erişim, doğrudan

PBL kullanmaktan daha kolaydır. Her dersin içeriği, Tablo 3.2'de de gösterildiği gibi kolaydan zora doğru üç seviyeye ayrılmıştır.



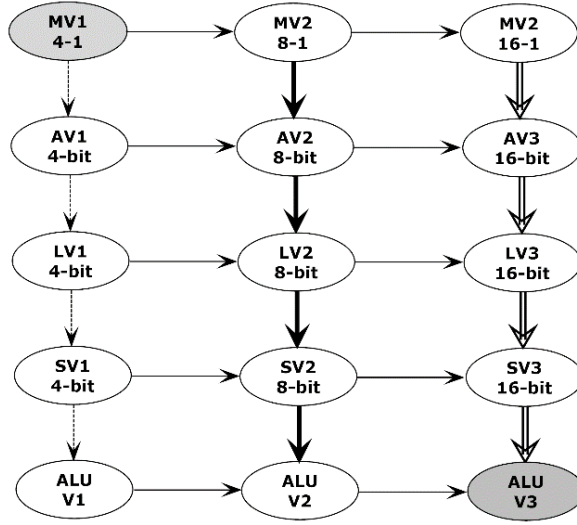
Şekil 3.6. Basitten karmaşığa alt blok basamakları.

Şekil 3.6'da kullanılan kısaltma ve sembollerin açıklamaları Tablo 3.2'de verilmiştir.

Tablo 3.2. FPGA ve ML yöntemlerinin etkileşimine ait kodlar ve anlamları.

Kod	Açıklama
MV1	VHDL diline göre çoklayıcı (Multiplexer) 4-1.
PMV2	VHDL diline göre çoklayıcı 8-1.
MV3	VHDL diline göre çoklayıcı 16-1.
AV1	VHDL diline göre 4-bit aritmetik komutlar.
AV2	VHDL diline göre 8-bit aritmetik komutlar.
AV3	VHDL diline göre 16-bit aritmetik komutlar (ADD için).
LV1	VHDL diline göre 4-bit mantık komutları tasarımı.
LV2	VHDL diline göre 8-bit mantık komutları tasarımı
LV3	VHDL diline göre 16-bit mantık komutları tasarımı (NAND veya NOR için)
SV1	4-bit tasarım, VHDL diline göre kaydırma ve döndürme komutları.
SV2	8-bit tasarım, VHDL diline göre kaydırma ve döndürme komutları.
SV3	16-bit tasarım, VHDL diline göre kaydırma ve döndürme komutları.
ALUSHIFT 1	VHDL diline göre 4-bit aritmetik mantık kaydırma ve döndürme.
ALUSHIFT 2	VHDL diline göre 8-bit aritmetik mantık kaydırma ve döndürme.
ALUSHIFT 3	VHDL diline göre 16-bit aritmetik mantık kaydırma ve döndürme.

Tasarım, diyagramda gösterilen adımlarla ML kullanılarak daha kolay seviyelerden karmaşık olanlara ilerleyecek şekilde tasarlanmıştır. Örneğin, bir öğrenci MV1 4-1'den ALUSHIFT 16-bit'e kadar başlangıç noktasından nihai hedefe, Şekil 3.7'de gösterildiği gibi, tasarımda herhangi bir yolu takip edebilir.



Şekil 3.7. ML kullanılarak hazırlanan tasarım yolları.

Müfredatın her bir başlığı, ML bloğu adı verilen belirli bir kodla kendi bloğuna sahiptir. Bazı müfredatların sembolleri Tablo 3.3'de verilmiştir.

Tablo 3.3. Kullanılan semboller ve anlamları.

Kod	Tanım	Kod	Tanım
T-ML-1	Sayı sistemleri	T-ML-15	DE çoklayıcılar
T-ML-2	Elektronik sinyaller ve anahtarlar	P-ML-10	DE çoklayıcılar
T-ML-3	Aritmetik işlemler ve devreler	T-ML-16	Karşılaştırıcı
T-ML-4	Boolean cebiri ve indirgeme teknikleri	P-ML-11	Karşılaştırıcı
T-ML-5	Karnough Haritası	T-ML-17	flip-flop
T-ML-6	Mantık kapıları	P-ML-12	flip-flop
P-ML-1	Mantık kapıları	T-ML-18	Kayıt Defterleri
T-ML-7	Değişiklik kayıtları	P-ML-13	Kayıt Defterleri
P-ML-2	Değişiklik kayıtları	T-ML-19	Sayaçlar
T-ML-8	Yarı toplayıcı	P-ML-14	Sayaçlar
P-ML-3	Yarı toplayıcı	T-ML-20	SM
T-ML-9	Tam toplayıcı	P-ML-15	SM
P-ML-4	Tam toplayıcı	T-ML-21	Aritmetik birim
T-ML-10	Yarı çıkarma	P-ML-16	Aritmetik birim
P-ML-5	Yarı çıkarma	T-ML-22	Mantık birimi
T-ML-11	Tam çıkarma	P-ML-17	Mantık birimi
P-ML-6	Tam çıkarma	T-ML-23	Kayan ve dönen birim
T-ML-12	Enkoder	P-ML-18	Kayan ve dönen birim
P-ML-7	Enkoder	T-ML-24	ALUSHIFT
T-ML-13	Dekoder	P-ML-19	ALUSHIFT
P-ML-8	Dekoder		
T-ML-14	Çoklayıcı		
P-ML-9	Çoklayıcı		

P: pratik, T: Teorik

Öğrenciler, başlangıçta bu proje hakkında bazı temel sorulara sahip olabilirler:

- Bu proje hakkında ne biliyorum?
- Tasarım hakkında ve bu projenin etkili bir şekilde nasıl ele alınacağı hakkında bilmem gerekenler nelerdir?
- Sorunumun nedenlerini ve çözümünü belirlemek için hangi kaynakları alabilirim?

Her ne kadar yeni bilgiler elde edildiğinde ve anlaşıldığında bu değişecek olsa da, bu aşamada, bazı soruları dile getirme ve bu projedeki bazı önemli noktaları açıklığa kavuşturma ihtiyacı vardır. Bazı müfredatların sembolleri Tablo 3.4'de verilmiştir..

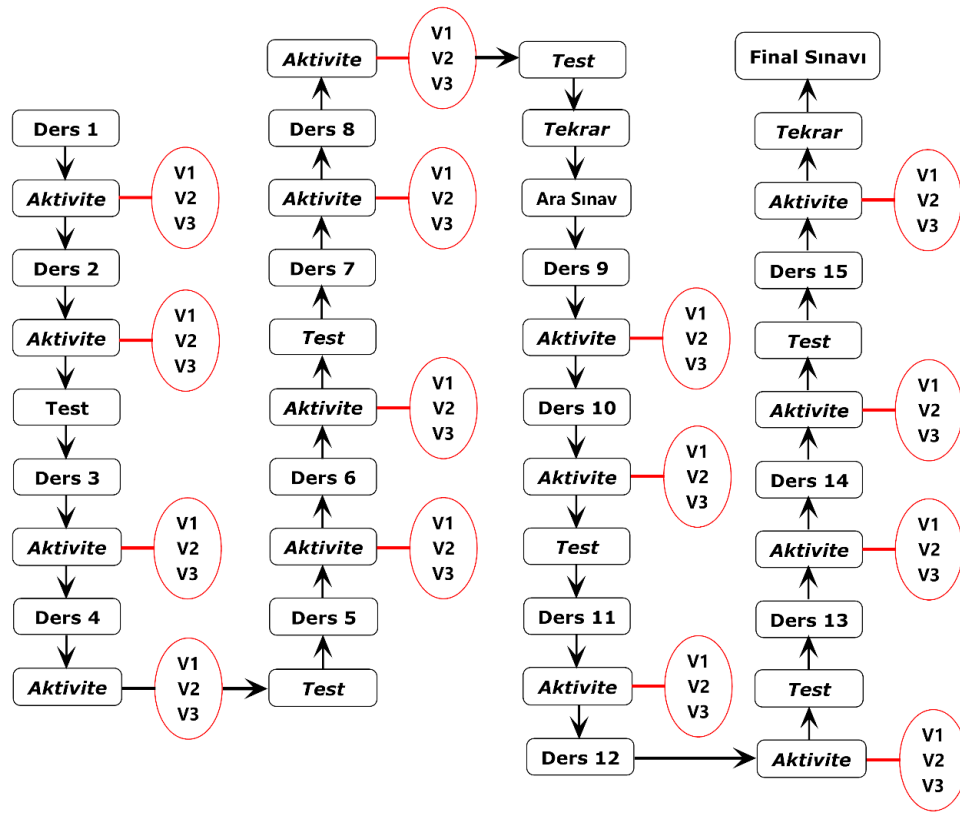
Tablo 3.4. Müfredat ve ders kodları.

	Kod	Öğrenme süresi	Gerekli arka plan	İçerik / Müfredat	Kurstaki yer	ML Blok	Video Meteriyal kullanımı
<b>ALUSHIFT ünitesi</b>	T-ML-24	30 dk	T1, T3, T4-T11, T14, P1-P6, T18, T21-T23, P9, P13, P16-P18	Teorik, pratik	İkinci hafta	T35, P30	Aritmetik birim (2 dk.)
<b>Öğrenim çıktısı</b>							<ul style="list-style-type: none"> <li>- Mikroişlemcinin tüm aritmetik işlemleri aritmetik mantık kaydırma ve döndürme ünitesinde (ALUSHIFT) gerçekleştirilir.</li> <li>- Kapı ve VHDL kombinasyonunun kullanılması.</li> <li>- Gerçekleştirilecek işlem kontrol ünitesinden gelen sinyallerle belirlenir.</li> <li>- İşlemlerin gerçekleştirildiği veriler bellekten veya harici bir girişten gelebilir.</li> <li>- Veriler bir şekilde akümülatörün içeriği ile birleştirilebilir ve sonuçlar tipik olarak akümülatöre yerleştirilir, oradan hafızaya veya bir çıkış ünitesine aktarılabilir.</li> </ul>
<b>Sorular</b>							<ul style="list-style-type: none"> <li>- ALUSHIFT ünitesini tasarlama ve uygulama amacı nedir?</li> <li>- ALUSHIFT ünitesini incelemek için gerekli arka plan nedir?</li> <li>- VHDL ile tasarım için uygun yöntem nedir?</li> <li>- 16-bit ALUSHIFT ünitesini nasıl tasarlayabilirsiniz?</li> <li>- Örneğin, karmaşık kısım (sıfır ve sıfır) gibi diğer bölümleri nasıl tasarlayabilirsiniz?</li> </ul>

Her bir dersin istenilen gereksinimleri ve nedenlerini aşağıdaki gibi özetleyebiliriz:

- *İsim:* Deneyin adı verilmek istenilen konuya uygun olarak seçilir.
- *Kod:* Her deneyimin kendine özgü bir kodu vardır.
- *Öğrenme süresi:* Her deneyin belirli bir zamanı vardır.
- *Gerekli arka plan:* Herhangi bir dersi öğretmeden önce, kurs yöneticisi veya eğitmeni konuları iyice anlamalıdır, çünkü her kurs için bir zaman sınırı vardır.
- *İçerik/müfredat:* Her deney hem teorik hem de pratik olarak öğretilir, böylece ML yöntemini zahmetsizce uygulayabiliriz.

- *Kurstaki Yeri*: Her deneyin/projenin planında belirtilen bir tamamlama süresi vardır. Bu, her deneyi/projeyi gerçekleştirmek için belirli bir zaman ve haftanın ayrıldığı anlamına gelir.
- *MLB (Mikro Öğrenme Bloğu)*: Her deneyin, üzerinde uğraşılan nihai karmaşık projede gerçel bir yeri vardır.
- *Video-animasyon kullanımı*: Her deneyde, öğrencilerin karmaşık projenin nasıl çalıştığını anlamalarına yardımcı olmak için iki dakikalık bir video kullanılır.



Şekil 3.8. İkinci grup için mikro öğrenme ders akış diyagramı.

Öğrenme sürecinde her ders, kolaydan zora doğru üç farklı aktivite seviyesine ayrılmıştır. Bu, Şekil 3.8'de gösterilmiştir.

### 3.3. Proje Tabanlı Öğrenme (PBL)

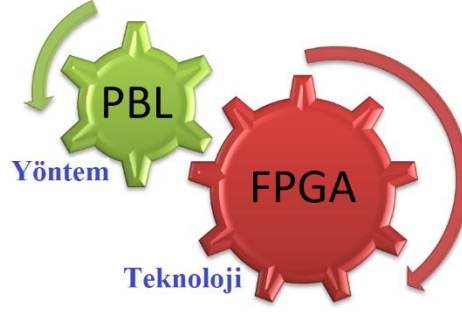
Sentezlenen PBL, öğrencilerin projelerini tamamlama sürecinde bir dizi konu öğrenmeleri ve beceri geliştirmelerini sağlayan bir başka eğitim yöntemidir. Bu

öğretim yöntemini kullanmak, öğrencilerin iletişim becerilerini ve ekip çalışmasını geliştirmelerine yardımcı olur. Ayrıca, öğrencilerin öğrenme deneyimlerinin sorumluluğunu üstlenmeleri ve pasif öğrenme kalıplarını aktif kalıplara dönüştürmeleri teşvik edilir.

Proje tabanlı öğrenme, öğrencilerin gerçek dünyada karşılaşılabilecekleri zorluklar ve problemlere yönelik projelerle, onlara bilgi ve beceri geliştirme fırsatı vermek için tasarlanmış bir öğretim yaklaşımıdır. PBL öğrenme yöntemi öğrencileri, derin ve sürekli dikkat gerektiren karmaşık sorunlar veya zorluklara dahil eden bir yapıya sahiptir. PBL modeli temel olarak şu yedi özellikten oluşur:

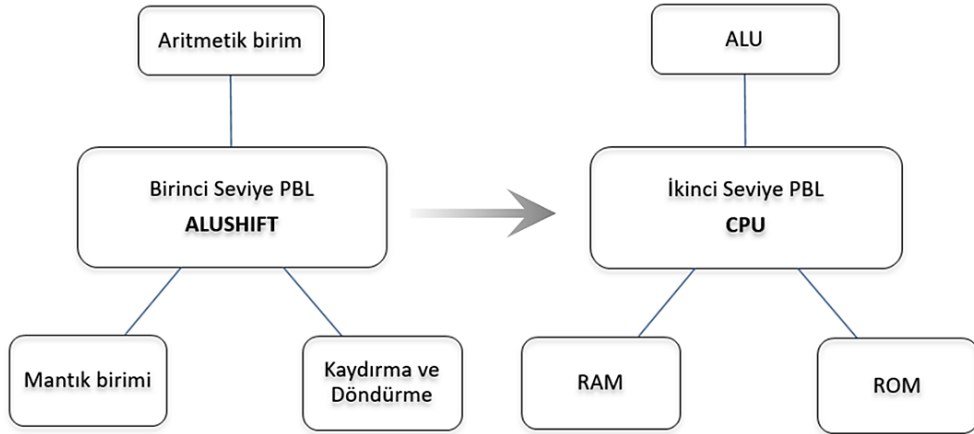
- Öğrenciyi, araştırması ve cevap vermesi ve/veya çözmesi için büyük, açık uçlu bir soru, zorluk veya problem üzerine odaklar.
- Öğrencilerin neyi bilmeleri, anlamaları gerektiğini ve ne yapabileceklerini anlamalarını sağlar.
- Sorgulama tabanlıdır.
- Eleştirel düşünme, iletişim, işbirliği ve yaratıcılık gibi becerilerini kullanır ve geliştirir.
- Öğrencinin süreç içerisinde kendi seçimlerini uygulama olanağı vardır.
- Plan ve projenin geri bildirim ve revizyonu için fırsatlar sağlar.
- Öğrencilerin sorunlarını, araştırma sürecini, yöntemlerini ve sonuçlarını ortaya koymalarını gerektirir.

PBL'nin her ne kadar başarılı yönleri olsa da, öğrencilerin iyi bir projeyi başarmalarına yardımcı olmak için FPGA teknolojisi ve PBL öğrenme yöntemlerinin birleştirilmesinin öğrenme verimliliğine etkisinin iyi yönde olacağı düşüncesi tarafımızca ele alınmıştır. Şekil 3.9'da gösterildiği gibi, böyle bir yaklaşım öğrencilerin karmaşık tasarım sorunlarını daha kolay çözmelerini sağlayacaktır.



Şekil 3.9. FPGA ve PBL'nin etkileşimi (Kurs 3 için).

Bilgisayar mimarisi kurs planlarına ana örnekler eklemek ve daha sonra bunları her bir alt bloğun bilgisayar mimarisine ilişkin belirli bir konuyu ele alması gereken şekilde daha küçük alt bloklara bölmek önemlidir. Ayrıca, bu bloklar yalnızca öğrencileri eğitmek için oluşturulmalıdır; PBL kullanılarak nihai tasarım aşamasına nasıl erişilebileceği Şekil 3.10'de gösterilmektedir.



Şekil 3.10. Proje tabanlı öğrenmenin seviyeleri.

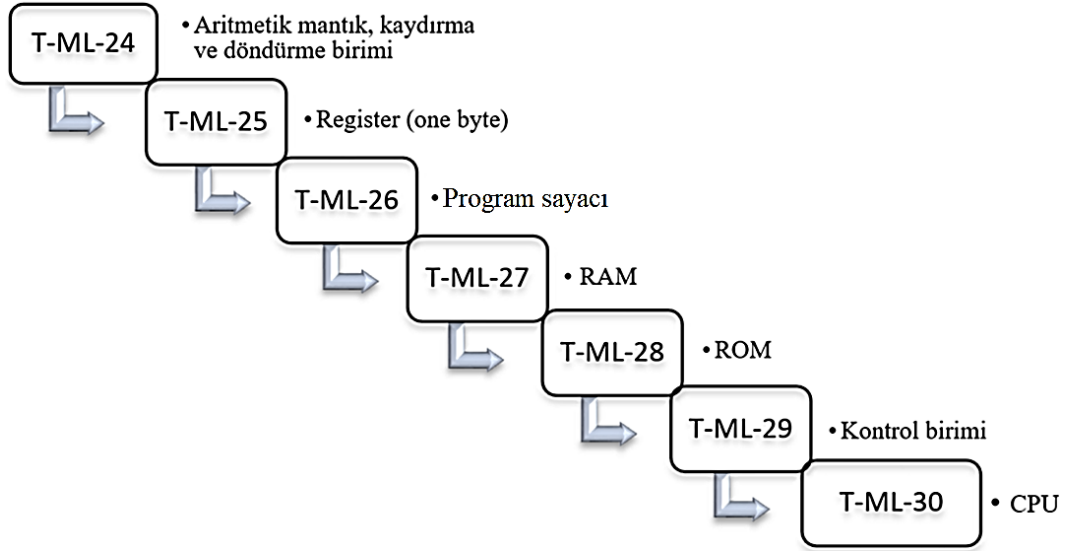
Müfredat tabanlı bir proje tasarlarken, tasarım daha küçük alt bloklara veya adımlara ayrılabilir. Bu, Şekil 3.11 ve Tablo 3.5'te gösterilmiştir. Daha önceki kullanılan testlere nazaran PBL kullanıldığında nihai tasarıma erişmenin daha kolay olduğu görülmektedir.

Her derste yer alan her bir başlık, Şekil 3.11'de gösterildiği gibi bölünmüştür.

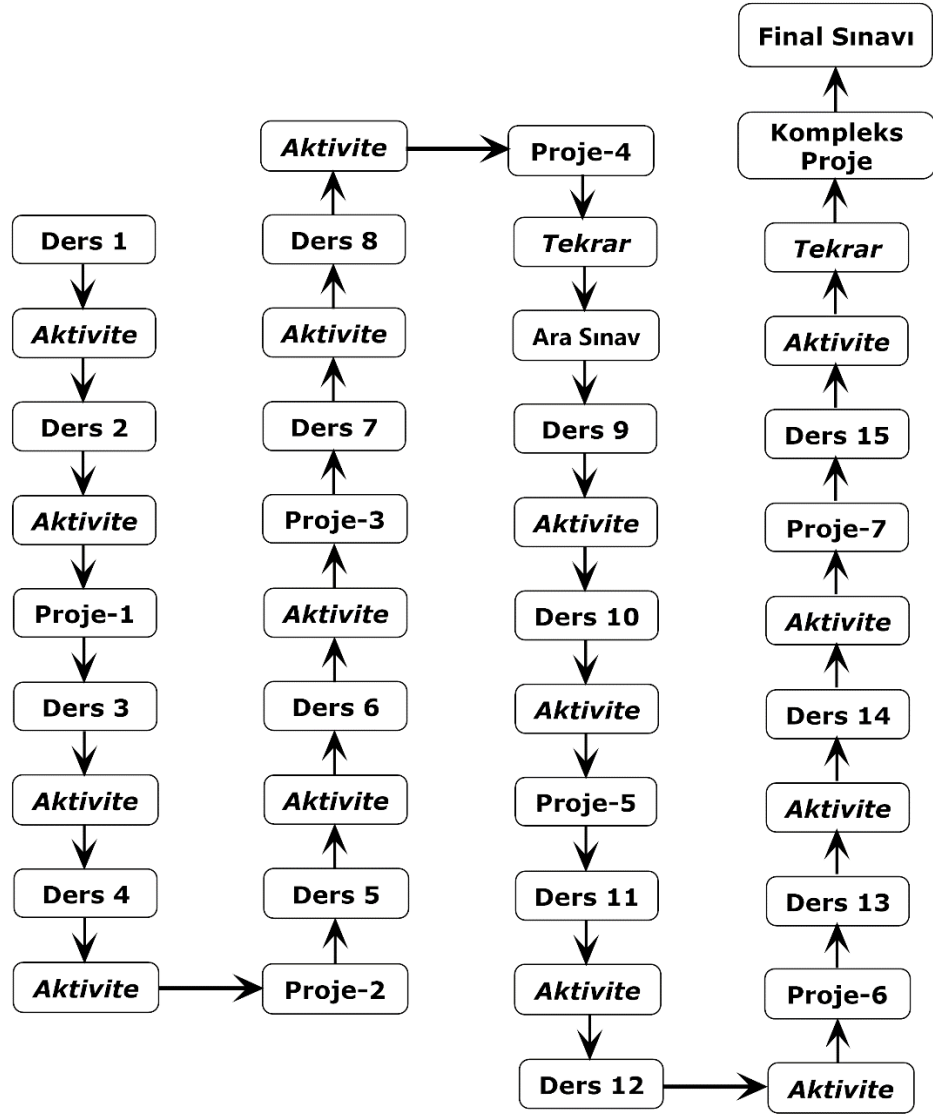
Tablo 3.5. Semboller ve anlamları.

Kod	Tanım
T-ML-24	Aritmetik mantık, kaydırma ve döndürme birimi
P-ML-19	Aritmetik mantık, kaydırma ve döndürme birimi
T-ML-25	Kayıt ol (bir bayt)
P-ML-20	Kayıt ol (bir bayt)
T-ML-26	Program sayıcı
P-ML-21	Program sayıcı
T-ML-27	Ana hafıza-RAM
P-ML-22	Ana hafıza-RAM
T-ML-28	ROM
P-ML-23	ROM
T-ML-29	Kontrol ünitesi
P-ML-24	Kontrol ünitesi
T-ML-30	İşlemci
P-ML-25	İşlemci

Her proje, derslerde ele alınan konuları birleştirecek şekilde tasarlanmıştır: Örneğin, ilk proje birinci ve ikinci dersi birleştirirken dördüncü proje, Şekil 3.12'te gösterildiği gibi, birinciden sekizinciye kadar olan dersleri birleştirmektedir. Bu kümülatif yaklaşım PBL ile öğrenmenin başarısını daha da pekiştirmektedir.

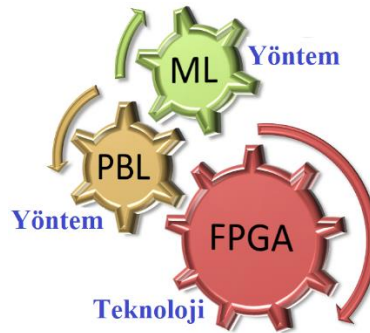


Şekil 3.11. Küçük alt bloklara bölünmüş CPU projesi.



Şekil 3.12. Derslerin, üçüncü grup için PBL ile düzenlenmesi.

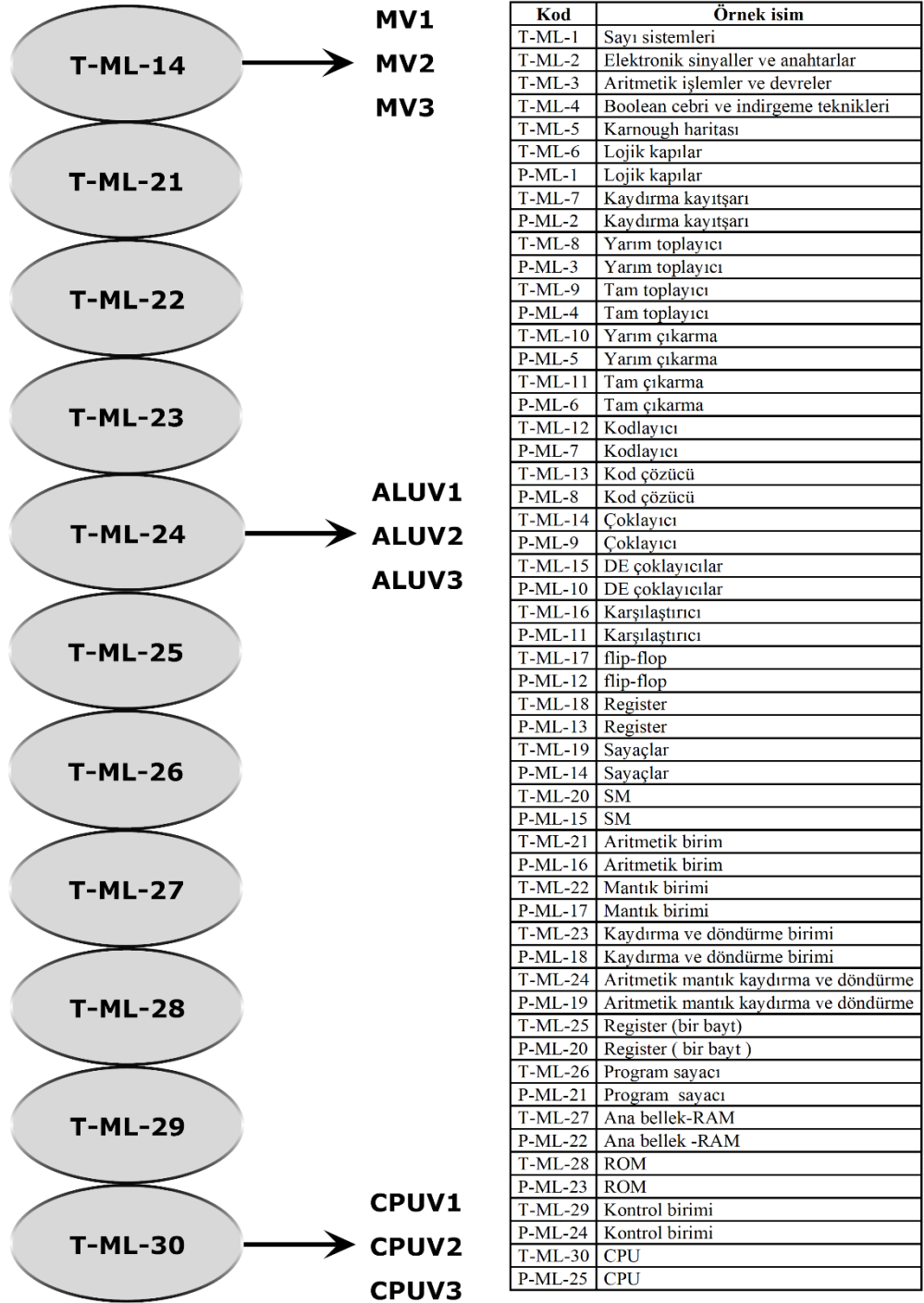
### 3.4. PBL ve ML'nin Kombinasyonu



Şekil 3.13. FPGA ile ML-PBL etkileşimi.



Bir dizi örnek kullanarak bir proje oluşturmak mümkündür ve proje diğer daha karmaşık projelerin tasarımlarına örnek olarak hizmet edebilir. Elde edilen test sonuçlarına göre, ML+PBL kombinasyonunun kullanılması, sadece PBL kullanımından daha iyi sonuçlar vermiştir. Derslerde yer alan her bir konu, Şekil 3.15'da gösterildiği gibi, basitten zora üç seviyeye ayrılmıştır.



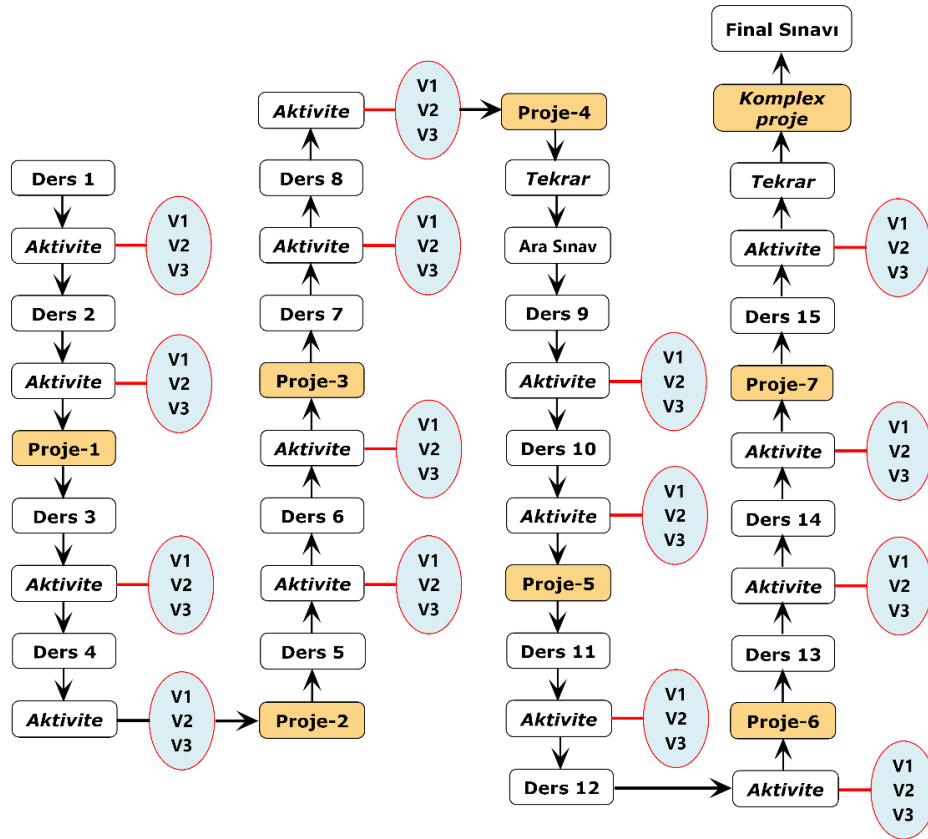
Şekil 3.15. CPU projesinin basitten karmaşığa küçük alt bloklara bölünmesi.

CPU tasarımına başlarken ilk adım, projenin karmaşıklığı ile başa çıkmak için projeyi birçok parçaya bölmektir. Bu, Tablo 3.6'da gösterildiği gibi, öğrencilerin nihai proje tasarımlarını anlamalarına ve gerçekleştirmelerine yardımcı olan ML yöntemini kullanarak mümkündür.

Tablo 3.6. Kullanılan kodlar.

Kod	Tanım
<b>MV1</b>	VHDL ile çoklayıcı 4-1.
<b>MV2</b>	VHDL ile çoklayıcı 8-1.
<b>MV3</b>	VHDL ile çoklayıcı 16-1.
<b>ALUV1</b>	VHDL ile 4-bit aritmetik mantık kaydırma ve dönme komutları.
<b>ALUV2</b>	VHDL ile 8-bit aritmetik mantık kaydırma ve dönme komutları.
<b>ALUV3</b>	VHDL ile 16-bit aritmetik mantık kaydırma ve dönme komutları.
<b>CPUV1</b>	VHDL ile 4-bit CPU tasarlayın.
<b>CPUV2</b>	VHDL ile 8-bit CPU tasarlayın.
<b>CPUV3</b>	VHDL ile 16-bit CPU tasarlayın.

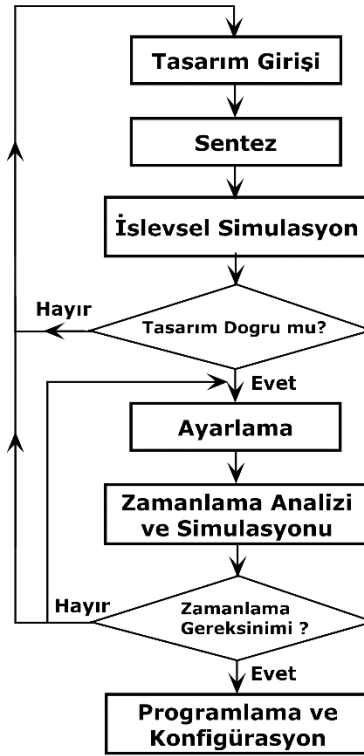
Daha önce de belirtildiği gibi her proje, daha önce verilen dersleri birleştirecek şekilde tasarlanmıştır. İlgili proje şeması Şekil 3.16'de verilmiştir.



Şekil 0.6. Dördüncü grup için ML+PBL ile öğretim planı.

### 3.5. Alanda Programlanabilir Kapı Dizisi (FPGA)

FPGA, üretimden sonra gereken işlevlere veya uygulamaya göre programlanabilen veya yeniden programlanabilen entegre bir devredir. FPGA'ların önemli karakteristikleri arasında düşük karmaşıklık, yüksek hız, farklı hacimli tasarımlar ve programlanabilir işlevler sayılabilir. Teknolojik ilerlemelerin ortaya koyduğu yeni tasarımlar için FPGA kullanılması şimdilik uygun bir seçim olarak karşımıza çıkmaktadır. Bilgisayar Destekli Tasarım (CAD) yazılımı, FPGA yongası gibi programlanabilir bir mantık cihazı kullanarak istenen bir mantık devresinin uygulanmasını kolaylaştırır. Tipik bir FPGA CAD akışı, Şekil 3.17'de gösterilmiştir.



Şekil 0.17. Tipik CAD akışı.

Tipik bir CAD akışı aşağıdaki temel adımları içerir:

- i. *Tasarım Girişi*: İstenilen devre, Verilog veya VHDL gibi bir donanım tanımlama dili kullanılarak veya şematik bir diyagramla belirtilir.

- ii. *Sentez*: CAD sentez aracı, devreyi mantıksal elemanların (LE'ler) devreyi ve LE'ler arasındaki bağlantıları gerçekleştirmesini sağlayan bir ağ listesine dönüştürür.
- iii. *İşlevsel Simülasyon*: Sentezlenen devre, işlevsel doğruluğunu doğrulamak için test edilmiştir. Simülasyon herhangi bir zamanlama meselesini dikkate almamaktadır.
- iv. *Ayarlama (Fitting)*: CAD montaj aracı, net listede tanımlanmış LE'lerin gerçek bir FPGA yongasına yerleştirilmesini belirler; ayrıca, belirtilen LE'ler arasında gerekli bağlantıları kurmak için çip içindeki yönlendirme tellerini seçer.
- v. *Zamanlama Analizi*: Takılan devredeki çeşitli yollar boyunca yayılma gecikmeleri, devrenin beklenen performansı hakkında bilgi sağlamak için analiz edilir.
- vi. *Zamanlama Simülasyonu*: Takılan devre hem işlevsel doğruluğunu hem de zamanlamayı doğrulamak için test edilir.
- vii. *Programlama ve Konfigürasyon*: Tasarlanan devre, LE'leri konfigüre eden ve gerekli kablo bağlantılarını kuran konfigürasyon anahtarlarını programlayarak fiziksel bir FPGA yongasına uygulanır.

### 3.6. Araştırmanın Amacı

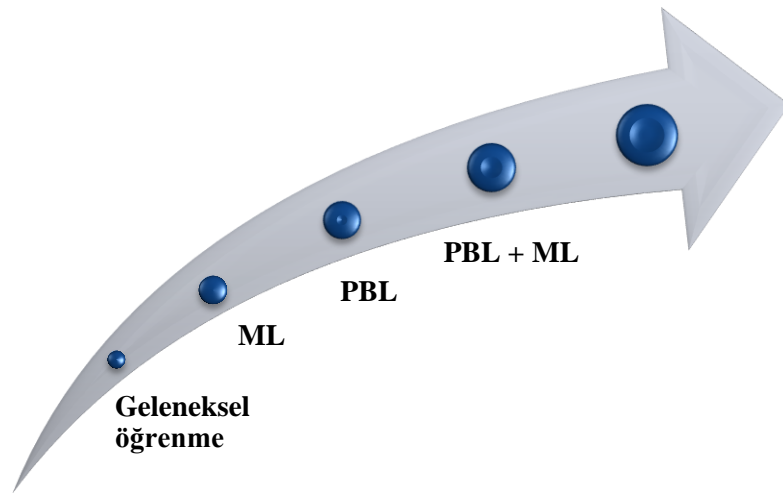
Bu çalışmanın amaçlarına ulaşmak için öncelikle aşağıdaki soruların cevaplanması gerekir:

- Öğrenciler öz-yönelimli öğrenenler olma konusunda nasıl motive edilebilir?
- Öğrencilerin birçoğunun kendi başlarına öğrenmek ya da öğrenmeleri için daha fazla sorumluluğa olan dirençlerinin üstesinden gelmelerine nasıl yardım edilebilir?
- Fakültenin ya da ilgili bölümün/birimin sınıfta yeni bir şey denemek konusundaki isteksizliğini nasıl aşabiliriz?

Geleneksel eğitim, eğitimin temelidir ve geride bırakılamaz; ancak, geleneksel eğitimin geliştirilmesi için yeni yöntemler, teknolojiler, uygulamalar vs. oluşturma gereksinimi de göz ardı edilmez bir gerçektir.

Geleneksel öğrenmeyi geliştirmenin bir yolu, Şekil 3.18'da gösterildiği gibi, yeni ve geliştirilmiş proje tabanlı öğrenme ve mikro öğrenme yöntemlerini birleştiren ayrı bir ders şeklindeki proje tabanlı öğrenmedir.

Çalışmanın amacı, öğrencileri gerçek ve karmaşık örnekler kullanarak konulara çekmek, onları iyi tasarımcılar olmaya hazırlamak, tasarım konusunda güven ve yetenek vermektir. Bunun yanında, mühendislik öğrencilerinin eğitim problemlerini “kendi kendine yönlendirilen (öz yönelimli) bağımsız öğrenme” ile çözmektir. Ayrıca, önerilen müfredat projesi gerçek dünyadaki bir projenin geliştirilmesine dayanmaktadır. Geliştirilen öğretici ve ilgi çekici örnekler, daha etkili bir bilgisayar mimarisi eğitimi için donanım ve yazılımı bir araya getirmiştir. Bu çalışmanın bir diğer amacı ise başarılı bir öğrenme sürecinde bulunması gereken birçok önemli faktörü göz önüne alarak geleneksel öğrenme modeli, mikro öğrenme modeli ve proje tabanlı öğrenme modeli arasında kapsamlı bir karşılaştırma yapmaktır.



Şekil 0.28. Mühendislik eğitimi geliştirme yöntemi.

Bu çalışmada hazırlanan proje, öğrencilere karmaşık proje tasarımını öğretmek için heyecan verici, ilgi çekici ve zorlayıcı bir şekilde tasarlanmıştır. Bu çalışma özellikle bilgisayar/elektronik mühendisliği üniversite için yeni bir eğitim tasarımı

sunmaktadır. Öğretim tasarımı, proje tabanlı öğrenmeyi kullanan bazı karmaşık ve açık araştırma projelerine yöneliktir; ancak bu çalışmada hem PBL hem de ML birleştirilerek yeni bir öğretim yöntemi kullanılmıştır.

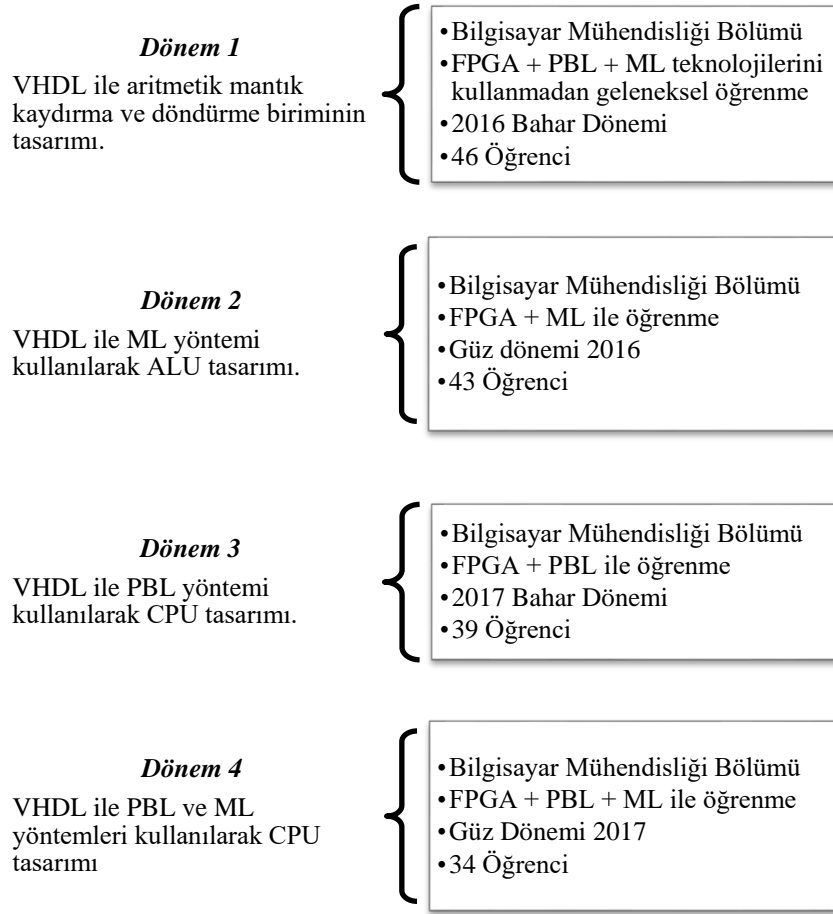
### 3.7. Çalışmanın Kapsamı

Bu çalışmanın amaçlarından biri aşağıdaki yöntemleri uygulayarak bilgisayar mimarisini öğretmek için kullanılan öğretim yöntemlerini karşılaştırmaktır:

- 1) FPGA teknolojisini kullanmadan geleneksel öğrenme.
- 2) FPGA teknolojisini kullanarak mikro öğrenme yöntemi.
- 3) FPGA teknolojisini kullanarak proje tabanlı öğrenme yöntemi.
- 4) FPGA teknolojisi kullanılarak proje tabanlı öğrenme modeli ve mikro öğrenme modellerinin birleşimi.

Karşılaştırma faktörleri, mikro öğrenme modelini, proje temelli öğrenme modelini ve proje tabanlı öğrenme modelini ve mikro öğrenme modelini bir araya getiren FPGA teknolojisinin kullanılmasının etkisine dayanmaktadır.

Karşılaştırma faktörleri, *öğretim faktörü*, *çekicilik faktörü*, *geçerlik faktörü*, *yeterlik faktörü* ve *karmaşıklığa izin verme faktörü* olmak üzere beş kısımda incelenmiştir. Çalışma sırasında nicel araştırma metodolojisi uygulanmıştır. Bu çalışma, pratik projelere dayalı öğrenme yöntemlerinin önemini ortaya koymaktadır. Ayrıca, proje tabanlı öğrenme ile geleneksel öğrenme yöntemiyle mikro öğrenme arasında bir karşılaştırma yapılmıştır. Çalışma kapsamında (Şekil 3.19) bilgisayar mühendisliği bölümünden dört öğrenci grubu seçilmiştir. Bu gruplar için dört yarıyılık eğitim planı aşağıdaki gibidir:



Şekil 3.19. Dört eğitim döneminde çalışma planı.

**Birinci yarıyıl:** VHDL ile aritmetik mantık kaydırma ve döndürme biriminin tasarımı geleneksel öğrenme yöntemiyle (FPGA + ML+PBL kullanılmadan) yapıldı (Bahar Dönemi-2016). Sınıfın öğrenci sayısı 46'dır.

**İkinci yarıyıl:** FPGA + ML kullanılarak VHDL ile aritmetik mantık kaydırma ve döndürme biriminin tasarımı öğretildi (Güz Dönemi-2016). Sınıfın öğrenci sayısı 43'tür.

**Üçüncü yarıyıl:** CPU tasarımı VHDL ile PBL öğrenme yöntemi kullanılarak öğretildi. FPGA + PBL kullanıldı (Bahar Dönemi-2017). Sınıfın öğrenci sayısı 39'dur.

**Dördüncü yarıyıl:** CPU tasarımı VHDL ile FPGA + ML+PBL teknolojileri kullanılarak gerçekleştirildi (Güz Dönemi-2017). Sınıfın öğrenci sayısı 34'tür.

### 3.7.1. Çalışma Projesi Örnekleri

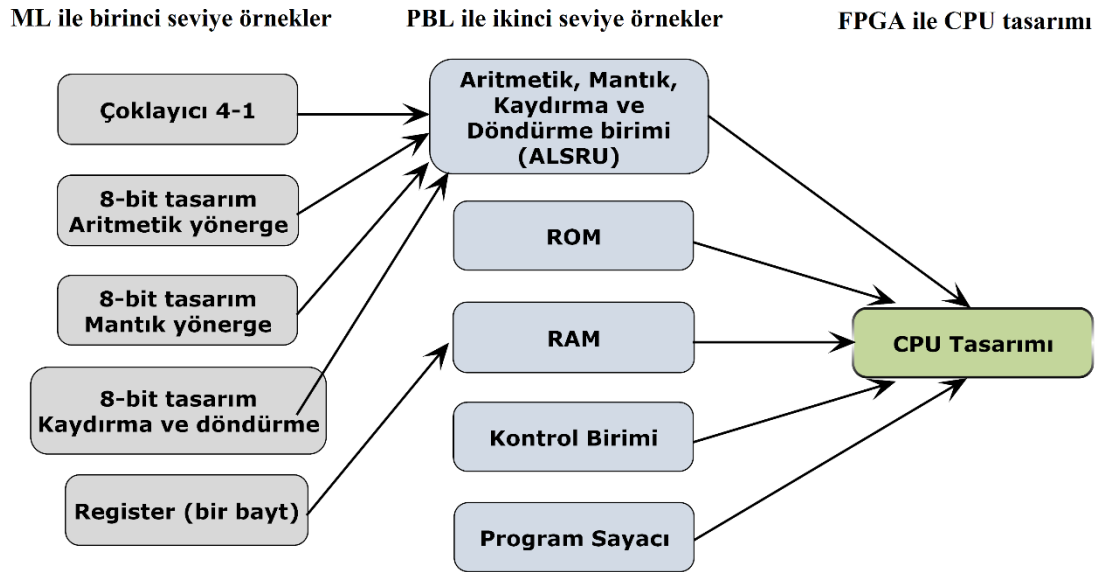
Bu deney iki seviyeye ayrılmıştır; birinci seviye ML'ye, ikinci seviye ise, Şekil 3.20'de gösterildiği gibi, PBL'ye bağlı olarak tasarlanmıştır.

**Birinci seviye:** FPGA teknolojilerini ve ML yöntemini kullanan 5 örnek.

- i. VHDL ile çoklayıcı 4-1.
- ii. VHDL ile 8-bit aritmetik komut tasarımı.
- iii. VHDL ile 8-bit lojik komut tasarımı.
- iv. VHDL ile 8-bit tasarım kaydırma ve döndürme talimatları.
- v. VHDL ile kayıt (bir bayt).

**İkinci seviye:** FPGA teknolojilerini ve PBL yöntemini kullanan altı proje.

- i. Aritmetik mantık kaydırma ve döndürme komutlarının tasarımı.
- ii. VHDL ile ROM yapısı.
- iii. VHDL ile ana hafıza.
- iv. VHDL ile kontrol ünitesi.
- v. VHDL ile program sayacı.
- vi. VHDL ile CPU tasarımı



Şekil 0.30. FPGA teknolojisi, PBL ve ML ile CPU tasarımı.

### 3.7.1.1. Birinci Seviye: FPGA Teknolojisini ve ML Yöntemini Kullanma

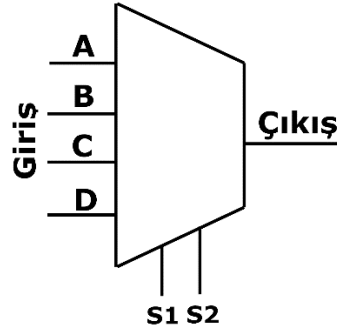
#### 3.7.1.1.1. VHDL ile Çoklayıcı 4-1

Bir çoklayıcı (MUX), birkaç girişi ve yalnızca bir çıkışı olan bir birleşik mantık bileşenidir. MUX, girişlerden birini bir kontrol bit sözcüğü (seçim satırı) kullanarak çıkış hattına yönlendirir. Çoklayıcı aşağıdakileri içerir:

- $2^N$  veri girişi.
- N seçim girişi.
- Tek bir çıktı.

Seçim girişi, çıkışa bağlanması gereken girişi belirler. Çoklayıcıya bazen veri seçici de denir. Çoklayıcı, farklı seçeneklerden birini seçen elektronik bir anahtar gibi çalışır.

Bu örnekte, Şekil 3.21 ve Tablo 3.7'de gösterildiği gibi, S1 ve S2 anahtarları bağlı (switch) iken, 4 giriş sinyali ve 1 seçilmiş çıkış vardır.

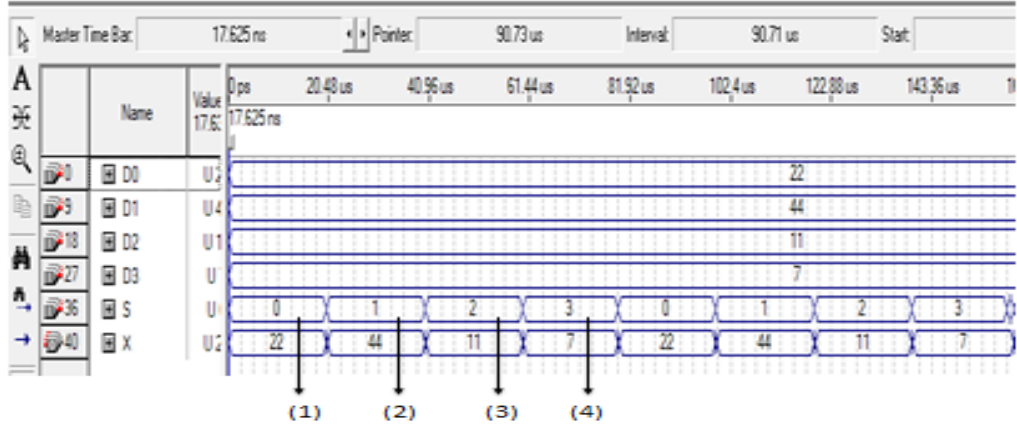


Şekil 3.21. MUX 4-1'in yapısı.

Tablo 3.7. Çoklayıcı 4-1.

S1	S2	İletilen
0	0	A
0	1	B
1	0	C
1	1	D

Multiplexer 4-1 tasarımı girilip başarıyla derlendiğinde, tasarımın doğrulanması gerekir. Simülatör, bu girişleri programlanmış bir cihazın aynı koşullar altında üretebileceği çıkış sinyallerini oluşturmak için kullanır. Çoklayıcı 4-1'in doğrulanması Şekil 3.22'de gösterilmiştir.



Şekil 0.22. Çoklayıcı 4-1'i doğrulama.

- $S_2 = S_1 = S_0 = 0$  (D0) ise verilerin kaydedici X'e (recorder X) aktarıldığı anlamına gelir.
- $S_2 = S_1 = 0, S_0 = 1$  (D1) ise verilerin kaydedici X'e aktarıldığı anlamına gelir.
- $S_2 = S_0 = 0, S_1 = 1$  (D2) ise verilerin kaydedici X'e aktarıldığı anlamına gelir.
- $S_2 = 0, S_1 = S_0 = 1$  (D3) ise verilerin kaydedici X'e aktarıldığı anlamına gelir

### 3.7.1.1.2. VHDL ile 8 Bit Aritmetik Komutlar

Aritmetik birim genellikle kısa süreli sayıların depolanması için birkaç kayıttan/yazmaçtan (register), toplayıcılardan (adders), sayılar üzerinde temel işlemlerin gerçekleştirilmesi için mantık devrelerinden ve bilgisayarın merkezi kontrol ünitesinden bir işlemi gerçekleştirmek için bir komut alan ve işlemi yapan bir yerel kontrol ünitesinden oluşur ve komutların gerekli olan dizinini işler.

Elektronik dijital bilgisayarların temel birimleri doğrudan aritmetik işlemleri gerçekleştirir. Birimdeki aritmetik işlemin performansı temel olarak bir dizi temel

işlem veya mikro işlemlere indirgenir. Bunlar arasında, herhangi bir aritmetik birim bloğu hanesinin 0'a ayarlanması, bir sayı kodunun veya tek tek hanesinin kabul edilmesi, kod çıktısının bulunması, sayı kodunun tersinin (karşılıklı) alınması, kodların eklenmesi ve kodun düşük veya daha büyük haneli bir sayının yan tarafına kaydırılması yer alır. Aritmetik ünite, Şekil 3.23 ve Tablo 3.8'de gösterildiği gibi sayısal verilerin işlenmesi için hesaplama işlemimi gerçekleştirir.

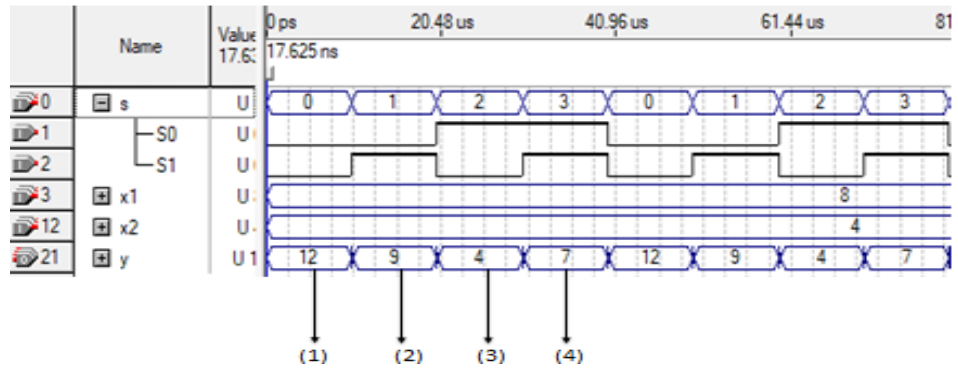


Şekil 3.23. Bir aritmetik birimin yapısı.

Tablo 3.8. Aritmetik birim.

Yönerge	S0	S1
ADD	0	0
SUB	1	0
INC	0	1
DEC	1	1

Aritmetik komut tasarımı girilip sonuç verdiğinde, daha önce de yapıldığı gibi tasarımın doğrulanması gerekir. Simülatör, Şekil 3.24'de gösterildiği gibi, tasarımın doğrulanmasından sonra üreteceği çıkış sinyallerini oluşturmak için kullanır.



Şekil 3.24. Aritmetik komutların doğrulanması

- $S1 = S0 = 0$  olduğunda, ADD işlemi başlar ve X1 ve X2 yazmaçlarının (register) eklenmesi gerçekleşir ve sonuç Y'de tutulur/saklanır (stored).
- $S1 = 1$  ve  $S0 = 0$  olduğunda, X1 registerının değerini birer arttıran INC işlemi gerçekleşir ve sonuç Y'de tutulur/saklanır.
- $S1 = 0$  ve  $S0 = 1$  olduğunda, SUB işlemi çalışır ve X1 yazmacının X2'den çıkarılması gerçekleşir, ardından sonuç Y'de tutulur/saklanır.
- $S1 = S0 = 1$  olduğunda DEC işlemi gerçekleştirildiğinde, X1 yazmacının değerini bir azaltır, ardından sonuç Y'de tutulur/saklanır.

### 3.7.1.1.3. VHDL ile 8 Bit Mantıksal Komutlar

Bir bilgisayarın içinde, mantıksal işlemleri gerçekleştirebilecek (örneğin, AND, OR, Ex-OR, Invert) bir Mantıksal Birim (LU) vardır. LU, bir bilgisayar sisteminin merkezi işlem biriminin ana bileşenidir. Komutları yürütmek için mantıksal işlemlerle ilgili tüm işlemleri yapar. Bazı mikroişlemci mimarileri Şekil 3.25 ve Tablo 3.9'da gösterilmiştir.

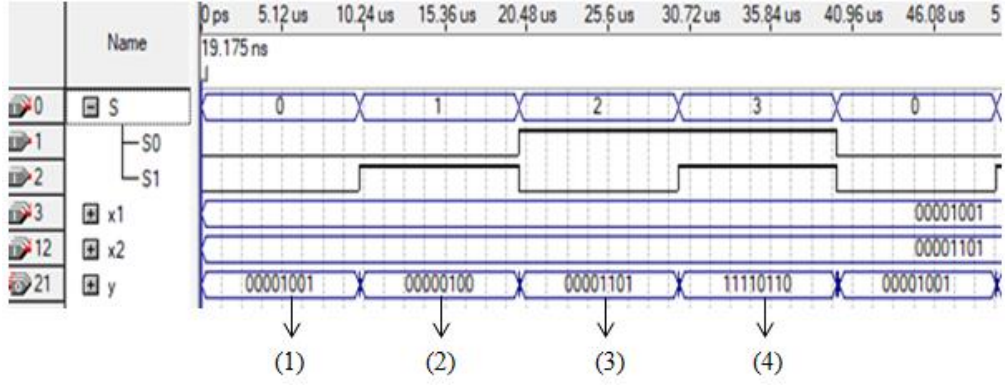


Şekil 3.25. Mantık biriminin yapısı

Tablo 3.9. Mantık birimi.

Yönerge	S0	S1
AND	0	0
OR	1	0
XOR	0	1
NOT	1	1

Mantıksal komut tasarımına girdikten ve derlendikten sonra, tasarım doğrulanır ve simülasyon yardımıyla çıkış sinyalleri kontrol edilir. Mantıksal talimatların doğrulanması Şekil 3.26'da gösterildiği gibi yapılır.



Şekil 3.26. Mantıksal komutları doğrulama.

- $S1 = S0 = 0$ , AND işlemi gerçekleştiğinde ve X1 kaydedicideki değer, X2 kaydedicideki değer ile karşılaştırılır ve sonuç Y konumunda tutulur/saklanır.
- $S1 = 1, S0 = 0$  olduğunda, XOR işlemi gerçekleşir ve X1 kaydedicideki değer, X2 kaydedicideki değerle karşılaştırılır ve sonuç Y konumunda tutulur/saklanır.
- $S1 = 0, S0 = 1$ , OR işlemi gerçekleştiğinde ve X1 kaydedicideki değer, X2 kaydedicideki değerle karşılaştırılır ve sonuç Y konumunda tutulur/saklanır.
- $S1 = S0 = 1$  olduğunda, NOT işlemi gerçekleşir ve X1 kaydedicideki değer, X2 kaydedicideki değerle karşılaştırılır ve sonuç Y konumunda tutulur/saklanır.

#### 3.7.1.1.4. VHDL ile 8 Bit Kaydırma ve Döndürme Tasarımı

*Bit Değiştirme İşlemleri:* Bitlerin konumlarını ya çarpma ya da bölme işlemi olan ya sağa ya da sola doğru belirli sayıda yerlere kaydırmaya yöneliktir.

ALUSHIFT kaydırma işlemleri A (veya B) işlecinin sola veya sağa (opcode'a bağlı olarak) kaymasına neden olur ve kaydırılan işlenen (operand) Y'de görünür. Basit ALUSHIFT tipik olarak işlemciyi sadece bir bit kaydırır, daha karmaşık ALUSHIFT ise işlemciyi tek bir işlemde rastgele sayıda bit ile değiştirmelerini sağlayan "barrel shifters" kaydırıcıları kullanır. Tüm tek bit kaydırma işlemlerinde, işlenenden çıkarılan bit, yürütülürken görünür ve bitin değeri, Şekil 3.27'de görüldüğü gibi, kaydırma türüne bağlı olarak işlenene geçer.

*Aritmetik kaydırma:* İşlenen, ikinin tamamlayıcı tamsayısı olarak değerlendirilir; bu, en önemli bitin bir "işaret" biti olduğu anlamına gelir ve korunur.

*Mantıksal kayma:* İşlenene bir mantık sıfır kaydırılır. Bu, işaretsiz tamsayıları kaydırmak için kullanılır.

*Döndürme:* İşlenen bitlerden oluşan dairesel bir tampon olarak işlenir; bu nedenle, en az ve en önemli bitlere sahiptir.

Tip	Sola Kayma	Sağa Kayma
Aritmetik		
Mantık		
Döndürme		

Şekil 3.27. Sekiz bitlik bir ALUSHIFT için bit kaydırma örnekleri.

**Kaydırma ve döndürme komutları:** X1 yazmacının içeriğini bir bit sağa veya sola Y yazmacına aktarır (Şekil 3.28 ve Tablo 3.10).



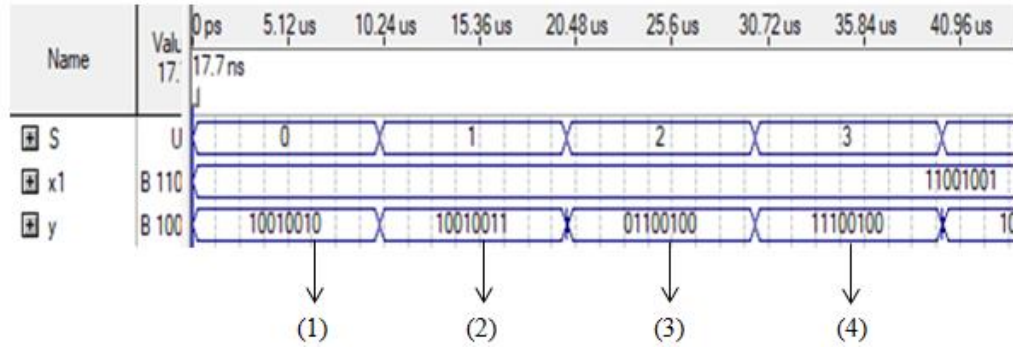
Şekil 3.28. Kayma ve döndürme birimin yapısı.

Kaydırma işleminden sonra, RI'lerin talimat tasarımı girilir ve başarıyla derlenir, bundan sonra tasarımın doğrulanması gerekir. Simülatör, yine, bu girişleri, programlanmış bir cihazın aynı koşullar altında üreteceği çıkış sinyallerini oluşturmak için kullanır.

Tablo 3.10. Kaydırma ve döndürme birimi.

Yönerge	S0	S1
AND	0	0
OR	1	0
XOR	0	1
NOT	1	1

Doğrulama, kaydırma ve RI talimatları, Şekil 3.29'da gösterilmiştir.

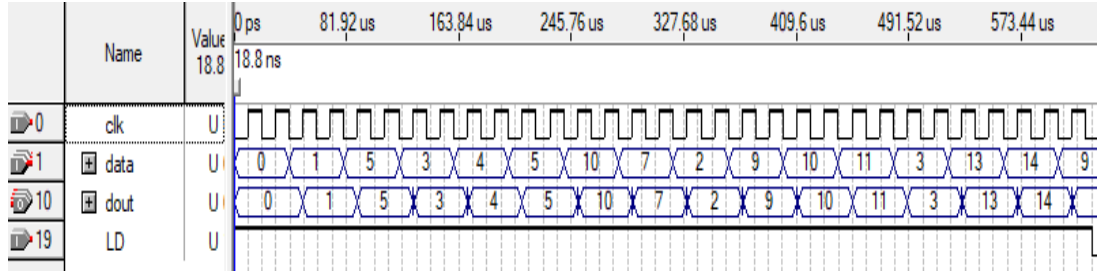


Şekil 3.29. Kaydırma ve döndürme talimatlarını doğrulama.

- $S1 = S0 = 0$  olduğunda SL işlemi yapılır, X1 kaydedicideki değer bir bit sola kaydırılır ve sonuç Y konumunda tutulur/saklanır.
- $S1 = 0, S0 = 1$ , ROL işlemi gerçekleştiğinde, X1 kaydedicideki değer bir bit sola kaydırılır ve sonuç Y konumunda tutulur/saklanır.
- $S1 = 1, S0 = 0$ , SR işlemi gerçekleştiğinde, X1 kaydedicideki değer bir bit sağa kaydırılır ve sonuç Y konumunda tutulur/saklanır.
- $S1 = S0 = 1$  olduğunda, ROR işlemi gerçekleşir ve X1 kaydedicideki değer bir bit kadar sağa döndürülür ve sonuç Y konumunda tutulur/saklanır.

### 3.7.1.1.5. VHDL ile Kayıt (Bir Bayt)

Bir register, bir bit dizisini tutabilen işlemcinin bir parçasıdır. Genellikle veriler birkaç bayttan oluşur. Örneğin, Bayt - sekiz bit; Word - iki bayt (16 bit) ve Double word - dört bayt (32 bit) şeklindedir. Kayıt tasarımı girilip başarıyla derlendiğinde tasarım doğrulanır. Simülatör, daha önce olduğu gibi çıkış sinyallerini oluşturmak ve doğrulama kaydı oluşturmak için kullanır (Şekil 3.30).



Şekil 3.30. Doğrulama kaydı (bir bayt).

LD aktif olduğunda, giriş (veri) ikinci işlemden (clk) sonra çıkış (dout) vasıtasıyla görülebilir.

### 3.7.1.2. İkinci Seviye: FPGA Teknolojisini ve ML Yöntemini Kullanma

#### 3.7.1.2.1. ALUSHIFT'in VHDL ile Tasarımı ve Simülasyonu

Çeşitli devreler, verileri aritmetik-mantık birimi veya ALUSHIFT olarak adlandırılan tek bir birleştirilmiş birim olan bir işlemci kullanarak işler. Her bir ünitenin test edilmesi iki aşamada gerçekleştirilir:

Tablo 3.11. ALUSHIFT biriminin opcode tablosu.

Yönerge	S0	S1	S2	S3	S4	Yönerge biçimi
<b>ADD</b>	0	0	0	0	0	Aritmetik
<b>SUB</b>	1	0	0	0	0	
<b>INC</b>	0	1	0	0	0	
<b>DEC</b>	1	1	0	0	0	
<b>AND</b>	0	0	1	0	0	Mantık
<b>OR</b>	1	0	1	0	0	
<b>XOR</b>	0	1	1	0	0	
<b>NOT</b>	1	1	1	0	0	
<b>SL</b>	0	0	0	1	0	Kaydırma
<b>ROL</b>	1	0	0	1	0	
<b>SR</b>	0	1	0	1	0	Döndürme
<b>ROR</b>	1	1	0	1	0	

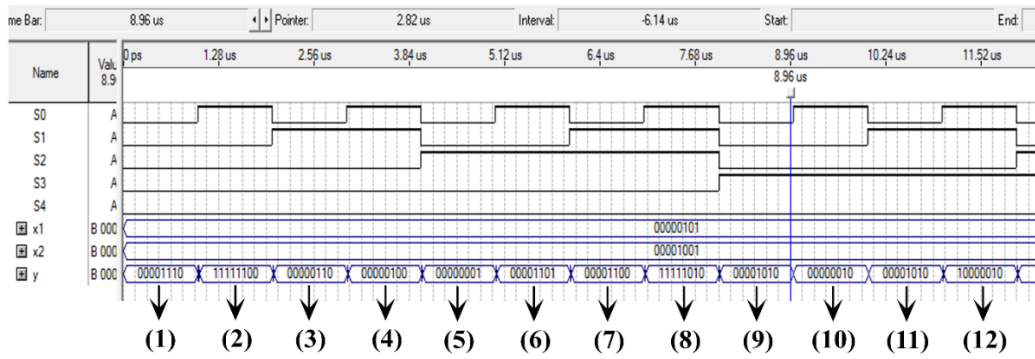
S: Sinyal kontrolü

Birinci aşamada, bu üniteye her bir ayrı blok test edilir. İkinci aşamada blok ara bağlantısı test edilir. Testler, test edilen parçaların düzgün çalıştığı stratejisine dayanır ve bu bileşenlerin birbirine bağlantısı düzgün çalışıyorsa, test edilen bloğun tamamı düzgün çalışıyor demektir. Seçilen testler ALUSHIFT ünitesi için en sık

kullanılan işlem grubunu temsil eder. Hafıza ve genel kayıtlar arasındaki ALUSHIFT içerik biriminin kodlanması Tablo 3.11'deki gibi opcode'lar üzerinden gerçekleştirilir.

ALUSHIFT komut tasarımı tamamen girildiğinde ve başarıyla derlendiğinde, tasarımın doğrulanır. Simülatör, benzer şekilde, çıkış sinyallerini oluşturmak için kullanır ve ALUSHIFT talimatlarının doğrulanması gerçekleştirilir (Şekil 3.6).

Testler yoluyla elde edilen sonuçlar Şekil 3.31'da gösterilmiştir. Seçilen testlerde test ünitesinin tüm parçalarının verilen spesifikasyonlara göre yapıldığı gösterilmiştir.



Şekil 3.31. ALUSHIFT doğrulama komutları.

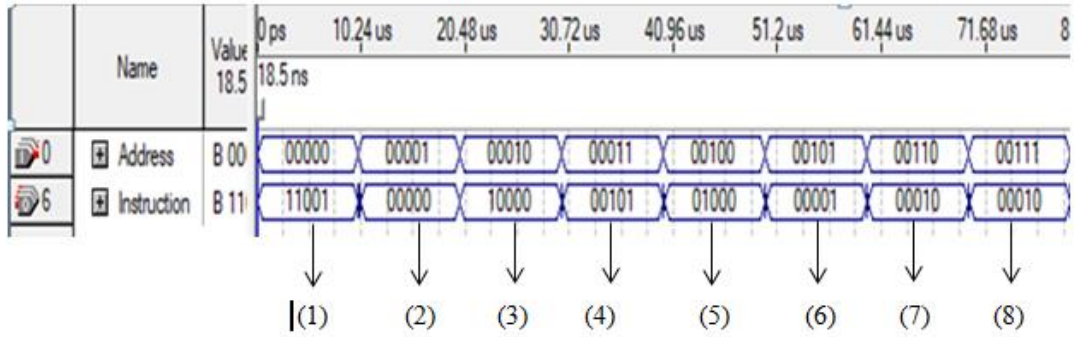
Şekil 3.31, ALUSHIFT biriminin simülasyon sonuçlarını göstermektedir. Bu testler ALUSHIFT ünitesi için en sık kullanılan işlem grubunu temsil eder. Bazı testlerin açıklaması aşağıdaki gibidir:

- $S2 = S3 = S4 = 0$  olduğunda, aritmetik birim çalışır ve  $S0 = S1 = 0$  olduğunda, ADD işlemi gerçekleştirir, bu, X1 kaydındaki (registrar) değer, X2 kaydındaki değer ile birleştirildiği anlamına gelir. Sonuç Y konumunda saklanır.
- $S2 = S3 = S4 = 0$  olduğunda, aritmetik birim çalışır ve  $S0 = 1, S1 = 0$  olduğunda, SUB işlemi gerçekleştirilir, bu, X1 kaydındaki değer, X2 kaydındaki değerden çıkarıldığı anlamına gelir. Sonuç Y konumunda saklanır.
- $S2 = S3 = S4 = 0$  olduğunda, aritmetik birim çalışır ve  $S0 = 0, S1 = 1$  olduğunda, INC işlemi gerçekleştirilir, bu, X1 kaydındaki değerin birer birer arttığı anlamına gelir. Sonuç Y konumunda saklanır.

- $S2 = S3 = S4 = 0$  olduğunda, aritmetik birim çalışır durumdadır ve  $S0 = S1 = 1$  olduğunda DEC işlemi gerçekleştirilir, bu, X1 kaydedicideki değer bir azaldığı anlamına gelir. Sonuç Y konumunda saklanır.
- $S2 = 1, S3 = S4 = 0$  olduğunda, mantıksal birim çalışıyor ve  $S0 = S1 = 0$ , AND işlemi gerçekleştiğinde, bu, X1 kaydedicideki değer, kaydedici X2'deki değerle karşılaştırıldığı ve sonuç Y konumunda saklanır.
- $S2 = 1, S3 = S4 = 0$  olduğunda, mantıksal birim çalışır ve  $S0 = 1, S1 = 0$ , OR işlemi başlatıldığında, bu, X1 kaydedicideki değer, X2 kaydedicideki değerle karşılaştırıldığı anlamına gelir. Sonuç Y konumunda saklanır.
- $S2 = 1, S3 = S4 = 0$  olduğunda, mantıksal birim çalışır ve  $S0 = 0, S1 = 1$  olduğunda, XOR işlemi gerçekleşir, bu, X1 kaydedicideki değer, X2 kaydedicideki değerle karşılaştırıldığı anlamına gelir. Sonuç Y konumunda saklanır.
- $S2 = 1, S3 = S4 = 0$  olduğunda, mantıksal birim çalışır ve  $S0 = S1 = 1$  olduğunda NOT işlemi gerçekleşir, bu, X1 kaydedicisinin, kaydedici X1'teki aynı değerle NOT işlemiyle karşılaştırıldığı anlamına gelir. Sonuç Y konumunda saklanır.
- $S2 = S4 = 0, S3 = 1$  olduğunda, birimnin değiştirilmesi ve döndürülmesi ve  $S0 = S1 = 0$  olduğunda SL işlemi gerçekleşir, bu, X1 kaydedicideki değer bir bit sola doğru kaldırıldığı anlamına gelir. Sonuç Y konumunda saklanır.
- $S2 = S4 = 0, S3 = 1$  olduğunda, kaydırma ve döndürme birimi çalışır ve  $S0 = 1, S1 = 0$  olduğunda, SR işlemi çalışır ve X1 kaydedicideki değer bir bit sağa doğru çıkarılır. Sonuç Y konumunda saklanır.
- $S2 = S4 = 0, S3 = 1$  olduğunda, kaydırma ve döndürme işlevleri ve  $S0 = 0, S1 = 1$  olduğunda, ROL işlemi X1 kaydedici değerini bir bit sola doğru döndürür. Sonuç Y konumunda saklanır.
- $S2 = S4 = 0, S3 = 1$  olduğunda, kaydırma ve döndürme birimi gerçekleştirir ve  $S0 = S1 = 1$  olduğunda, X1 kaydedici değerini bir bit sağa döndüren ROR işlemi gerçekleşir. Sonuç Y konumunda saklanır.

### 3.7.2. VHDL ile ROM

ROM, yalnızca okuyabildiğimiz ancak yazamadığımız salt okunur belleği ifade eder. Bu tür bir bellek kalıcı değildir. Bilgi, işlem sırasında bu hafızalarda depolanmaktadır. ROM, bir bilgisayarı başlatmak için gerekli olan talimatları saklar. ROM, CPU'nun bunları önceliğine göre uygulamalarını sağlamak için tüm talimat kodlarını kaydettiği anlamına gelir. Salt okunur belleğin (ROM) tasarımı girilip başarıyla derlendiğinde, tasarımın doğrulanması gerekir. Simülasyon, benzer şekilde salt okunur belleğin (ROM) doğrulanmasını yapmak için kullanılır (Şekil 3.32).



Şekil 3.32. Salt okunur belleğin (ROM) doğrulanması.

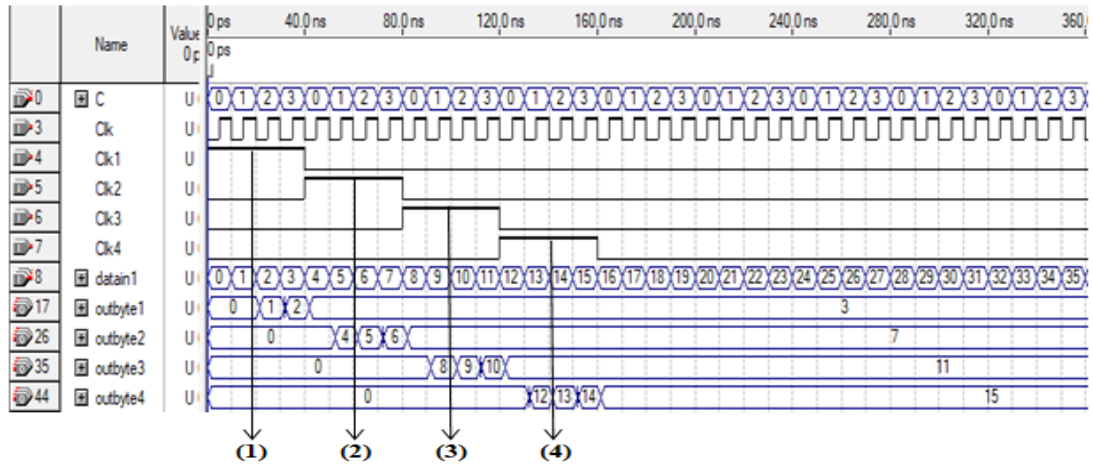
Şekil 3.7’de verilen numarandırmalar aşağıdaki gibidir:

- 00000 numaralı ilk adres, bu işlemin çalıştığı anlamına gelirken, 11001, klavyeden okuma anlamına gelir, ardından depolama X2 kaydındadır.
- İkinci adres 00001, bu işlemin çalıştığı anlamına gelirken, 00000, ADD işleminin gerçekleştiği anlamına gelir.
- Üçüncü adres (00010) bu işlemin çalıştığı anlamına gelirken, 10000 bellekten (bayt bir) X1 yazmacına kayan yükleme verileridir.
- Dördüncü adres 00011, bu işlemin çalıştığı, 00101 ise OR işleminin gerçekleştiği anlamına gelir.
- Beşinci adres 00100, bu işlemin çalıştığı ve 01000, SL işleminin gerçekleştiği anlamına gelir.
- Altıncı adres 00101, bu işlemin devam ettiği anlamına gelirken, 00001, SUB işleminin gerçekleştiği anlamına gelir.

- Yedinci adres 00110, bu işlemin çalıştığı, 00010 kodu ise INC işleminin gerçekleştiği anlamına gelir.
- Sekizinci adres 00111, bu işlemin çalıştığı anlamına gelirken, 00010 kodu, INC işleminin gerçekleştiği anlamına gelir.

### 3.7.3. VHDL ile Ana Bellek (RAM)

Bir bellek birimi olan RAM, depolama hücrelerine sahip, depoya bilgileri aktarmak için gereken ilgili devreler topluluğudur. Ana bellek (RAM) tasarımı girilip başarıyla derlendikten daha önce de yapıldığı gibi tasarımın doğrulanması gerekir ve simülatör yardımı ile doğrulama yapılır (Şekil 3.33).



Şekil 3.33. Ana belleği (RAM) doğrulama.

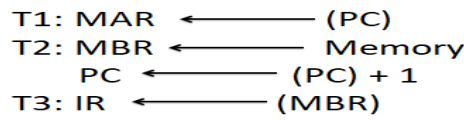
- Clk1 aktifken, veriler birinci bayta kaydedilir.
- Clk2 aktifken, veri ikinci bayta kaydedilir.
- Clk3 aktifken, veriler üçüncü bayta kaydedilir.
- Clk4 aktifken, veriler dördüncü bayta kaydedilir.

### 3.7.4. VHDL ile Kontrol Birimi

Kontrol birimi, bir bilgisayarın merkezi işlem biriminin (CPU) bir bileşenidir ve işlemcinin işlemlerini yönlendirir. Giriş ve çıkış cihazları arasındaki iletişimi ve koordinasyonu kontrol eder. Talimatları okur, yorumlar ve veri işleme sırasını

belirler. Bir CPU'nun işlevini belirlemek için gerekli olan üç temel yapı, *işlem kodu* (opcodes), *register* ve *bellek modülü* arayüzüdür.

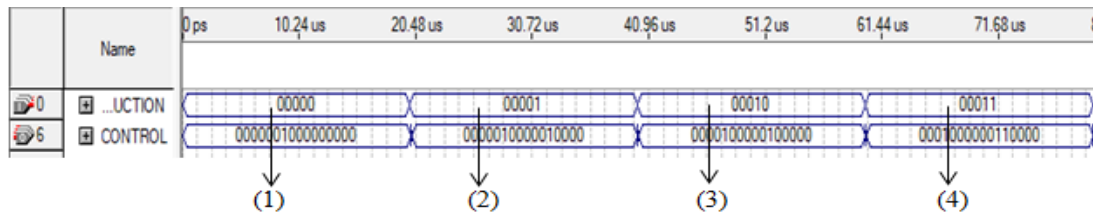
Basit bir getirme (fetch) döngüsü aslında üç adım ve dört mikro işlemden oluşur. Her bir mikro işlem, bir kaydın içine veya dışına verilerin hareketini içerir; bu nedenle, bu hareketler birbirine müdahale etmediği sürece, zamandan tasarruf sağlayan çok sayıda işlem aynı anda gerçekleşebilir. Bu olay dizisini aşağıdaki şekilde gösterebiliriz.



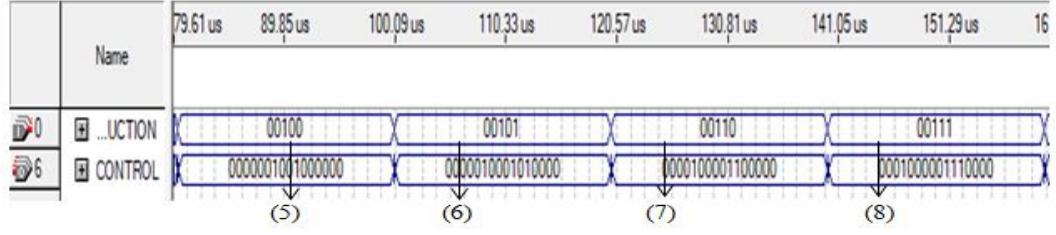
Burada T1, T2 ve T3 zamanlama sinyalleridir ve aşağıdaki görevleri üstlenirler:

- *Birinci zaman birimi T1*: PC içeriğini MAR'a taşır.
- *İkinci zaman birimi T2*: belirtilen bellek konumunun içeriğini MAR'dan MBR'ye taşır. PC'nin içeriği 1 arttırılır.
- *Üçüncü zaman birimi T3*: MBR'nin içeriğini IR'ye taşır.

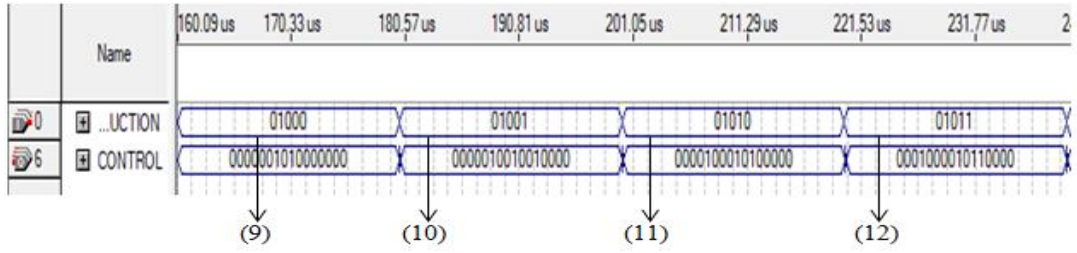
Kontrol birimi tasarımı girilip başarıyla derlendiğinde, tasarımın yine doğrulanması gerekir ve bu işlem simülatör yardımıyla kolaylıkla yapılabilir. Şekil 3.34'ten 3.38'e aritmetik, mantıksal, kaydırma ve döndürme, kayıt yeri ve kayıt girişlerinin simülasyon doğrulamaları gösterilmiştir.



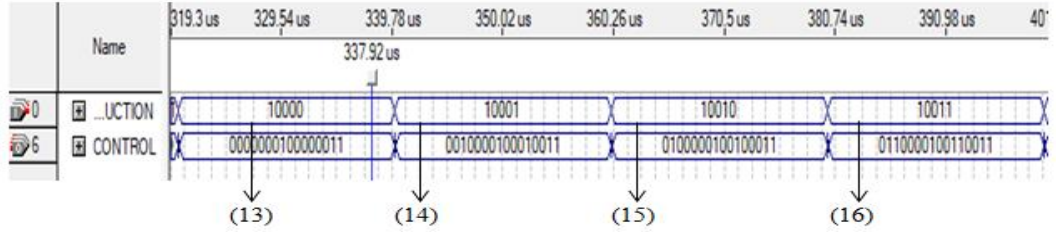
Şekil 3.34. Aritmetik komutlarla kontrol birimi doğrulanması.



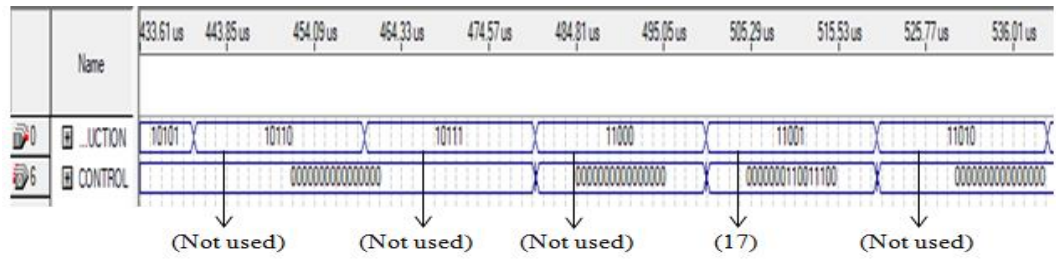
Şekil 3.35. Mantıksal komutlarla kontrol birimi doğrulama.



Şekil 3.36. Kaydırma ve döndürmeyle kontrol birimi doğrulaması.



Şekil 3.37. Kayıt yeri ile hafıza konumuna göre kontrol birimi doğrulaması.



Şekil 3.38. Kayıtlı giriş birimi ile kontrol birimi doğrulaması.

Tablo 3.12. Kontrol biriminin kontrol sinyali.

Kontrol Sinyali	M[4]			M[3]		M[2]		M[1]		Yönerge				X2		X1	
	C15	C14	C13	C12	C11	C10	C9	C8	C7	C6	C5	C4	C3	C2	C1	C0	
ADD	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	
SUB	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	
INC	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	
DEC	0	0	0	1	0	0	0	0	0	0	1	1	0	0	0	0	
AND	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	
OR	0	0	0	0	0	1	0	0	0	1	0	1	0	0	0	0	
XOR	0	0	0	0	1	0	0	0	0	1	1	0	0	0	0	0	
NOT	0	0	0	1	0	0	0	0	0	1	1	1	0	0	0	0	
SL	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	
SR	0	0	0	0	0	1	0	0	1	0	0	1	0	0	0	0	
RL	0	0	0	0	1	0	0	0	1	0	1	0	0	0	0	0	
RR	0	0	0	1	0	0	0	0	1	0	1	1	0	0	0	0	
M[0]→X1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	
M[1]→X1	0	0	1	0	0	0	0	1	0	0	0	1	0	0	1	1	
M[2]→X1	0	1	0	0	0	0	0	1	0	0	1	0	0	0	1	1	
M[3]→X1	0	1	1	0	0	0	0	1	0	0	1	1	0	0	1	1	
KEY→X2	0	0	0	0	0	0	0	1	1	0	0	1	1	1	0	0	

Temel sonuçlar aşağıdaki şekildedir (ayrıca Tablo 3.12'ye bak):

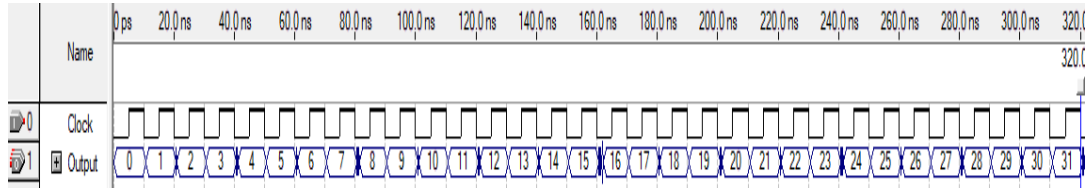
- $C4 = C5 = C6 = C7 = C8 = 0$  olduğunda, bu ADD işleminin gerçekleşeceği ve X1 kayıtcısındaki değerin X2 kayıtcısındaki değerle birleştirildiği ve  $C9 = 1$  olduğunda sonuç 1 konumundaki hafızaya kaydedilmelidir.
- $C4 = 1$  ve  $C5 = C6 = C7 = C8 = 0$  olduğunda, SUB işleminin gerçekleştirileceği, X1 kaydedicideki değerin X2 kaydedicideki değerden çıkarıldığı ve  $C10 = 1$  olduğunda sonucun çıkarıldığı anlamına gelir. 2 numaralı konumda hafızaya kaydedilmelidir.
- $C5 = 1$  ve  $C4 = C6 = C7 = C8 = 0$  ise, bu INC işleminin gerçekleşeceği anlamına gelir, X1 kaydedicideki değer birer birer artar ve  $C11 = 1$  olduğunda sonuç cihazda saklanır. 3. konumdaki hafıza.
- $C4 = C5 = 1$  ve  $C6 = C7 = C8 = 0$  olduğunda, DEC işleminin gerçekleşeceği, X1 kaydedicideki değerin bir puan azalacağı ve  $C12 = 1$  olduğunda sonucun 4. konumdaki hafıza depolanacağı anlamına gelir.
- $C6 = 1$  ve  $C4 = C5 = C7 = C8 = 0$  olduğunda, NAD işleminin gerçekleşeceği, X1 kaydedicideki değerin X2 kaydedicideki değerle karşılaştırılacağı ve sonuç  $C9 = 1$  olduğunda sonuç 1 konumundaki hafızaya kaydedilmelidir.
- $C4 = C6 = 1$  ve  $C5 = C7 = C8 = 0$  olduğunda, OR işleminin gerçekleşeceği anlamına gelir, X1 kaydedicideki değer, X2 kaydedicideki değer ile karşılaştırılır ve  $C10 = 1$  sonuç 2 konumundaki hafızaya kaydedilmelidir.

- $C5 = C6 = 1$  ve  $C4 = C7 = C8 = 0$  olduğunda, bu, XOR işleminin gerçekleşeceği, X1 kaydedicideki değerin, X2 kaydedicideki değer ile karşılaştırıldığı ve  $C11 = 1$  ise, sonuç 3 konumundaki hafızaya kaydedilecektir.
- $C4 = C5 = C6 = 1$  ve  $C7 = C8 = 0$  olduğunda, NOT işleminin gerçekleşeceği anlamına gelir, X1 kaydedicideki değer, NOT işleminde X1 kaydedicideki değer ile aynı değerle karşılaştırılır.  $C12 = 1$ , sonuç 4 konumunda hafızaya kaydedilmelidir.
- $C7 = 1$  ve  $C4 = C5 = C6 = C8 = 0$  olduğunda, SL işleminin gerçekleşeceği anlamına gelir, X1 kaydedicideki değer bir bit sola doğru çıkarılır ve sonuç  $C9 = 1$  olduğunda, 1 konumunda bellekte saklanır.
- $C4 = C7 = 1$  ve  $C5 = C6 = C8 = 0$  olduğunda, SR işleminin gerçekleşeceği, X1 kaydedicideki değerin bir bit sağa doğru kaydırıldığı ve sonucun  $C10 = 1$  olduğunda 2 numaralı yerde hafızaya alınması gerektiği anlamına gelir.
- $C5 = C7 = 1$  ve  $C4 = C6 = C8 = 0$  olduğunda, işlem RL demektir, X1 kaydedicideki değer bir bit sola doğru döndürülür ve  $C11 = 1$  olduğunda sonuç 3. konumdaki hafızada saklanmalıdır.
- $C4 = C5 = C7 = 1$  ve  $C6 = C8 = 0$  olduğunda, işlem RR demektir, X1 kaydedicideki değer bir bit sağa doğru döndürülür ve  $C12 = 1$  olduğunda, sonuç 4. konumdaki hafızada saklanmalıdır.
- $C8 = 1$  ve  $C4 = C5 = C6 = C7 = 0$  ise, işlem  $M [0] \rightarrow X1$  olur ve ilk konumun belleğindeki değer X1 yazmacına aktarılır, çünkü  $C0 = C1 = 1$ .'dir
- $C4 = C8 = 1$  ve  $C5 = C6 = C7 = 0$  olduğunda, işlem  $M [1] \rightarrow X1$  olur ve ikinci konumun belleğindeki değer X1 yazmacına aktarılır, çünkü  $C0 = C1 = 1$ .'dir
- $C5 = C8 = 1$  ve  $C4 = C6 = C7 = 0$  ise, işlemin  $M [2] \rightarrow X1$  olduğu ve üçüncü konumun belleğindeki değerin, X1 kayıt defterine aktarıldığı anlamına gelir, çünkü  $C0 = C1 = 1$ .'dir
- $C4 = C5 = C8 = 1$  ve  $C6 = C7 = 0$  ise, işlemin  $M [3] \rightarrow X1$  olduğu ve dördüncü konumun hafızasındaki değerin, X1 kayıtçısına aktarıldığı anlamına gelir çünkü  $C0 = C1 = 1$ .'dir

- $C8 = C7 = C4 = 1$  ve  $C6 = C5 = 0$  olduğunda, işlem  $KEY \rightarrow X2$  demektir, bu okuma klavyeden yapılır ve sonuçlar  $X2$  kaydedicide saklanır, çünkü  $C2 = C3 = 1$ 'dir.

### 3.7.5. VHDL ile Program Sayacı

Program hafızasına kaydedilen talimatları getirmek (fetch) için kullanılır. Komutlar, yapılacak işlemi ve işlenecek verileri gösterir. Adresleme modu, veri işlenenin konumunu belirlemek için kullanılan yöntemdir. Talimatlarda belirtilen işlenenler; durum kodu, anlık tarih veya kayıt, dosya, program hafızası veya veri hafızasındaki bir konum olabilir. Her komut bir sonraki komutun noktasına getirildiğinde, PC bir sonraki adıma geçer. Program sayacı tasarımı girilip başarıyla derlenir ve smülasyon ile doğrulanır (Şekil 3.39).



Şekil 3.39. Programı sayacını doğrulama.

### 3.7.6. VHDL ile CPU Tasarımı

#### 3.7.6.1. İşlemci

İşlemci, bilgisayar sisteminin bir parçasıdır ve işlemcinin bir kısmı işlem birimini oluşturur. İşlemcinin organizasyonunu anlamak için işlemcinin yapması gerekenler aşağıdaki şekilde özetlenebilir.

*Getirme (fetch) talimatı:* CPU, bellekten talimatları okumalıdır.

*Talimatı yorumla:* Hangi işlemin gerekli olduğunu belirlemek için talimatın kodu çözülmelidir.

*Verilerin alınması:* Bir talimatın yürütülmesi bellekten veya giriş birimi modülünden veri okunmasını gerektirebilir.

*Proses verileri:* Bir talimatın uygulanması, veriler üzerinde aritmetik veya mantıksal veya kaydırma ve döndürme işlemi yapılmasını gerektirebilir.

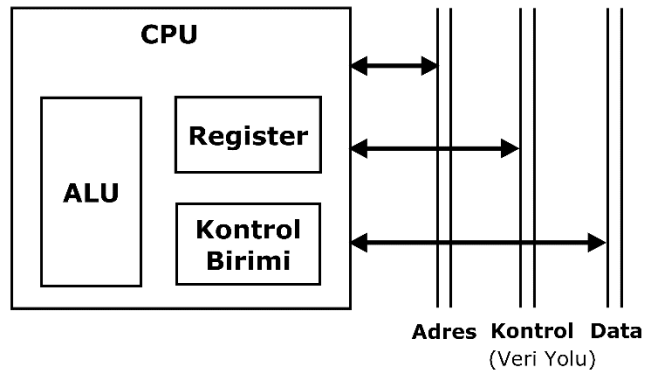
*Veri yazma:* Uygulama sonuçları belleğe veri yazılmasını gerektirebilir.

Bunları yapabilmek için CPU'nun bazı verileri geçici olarak saklaması gerektiği açıktır. Bir sonraki talimatın nereden alınacağını bilmesi için son komutun yerini hatırlamalıdır. Bir komut yürütülürken komutu ve verileri geçici olarak saklaması gerekir. Başka bir deyişle, CPU küçük bir dahili belleğe de ihtiyaç duyar (Dumas II, 2016).

### 3.7.6.2. İşlem Birimi

Bir CPU'nun sistem veriyolu üzerinden sistemin geri kalanıyla olan bağlantısının basit bir gösterimi Şekil 3.26'da verilmiştir. Benzer bir ara bağlantı, ara bağlantı yapılarının herhangi biri için de gereklidir ve CPU'nun ana bileşenlerinin aritmetik, mantık ve kontrol birimleri olduğu akılda tutulmalıdır.

ALU, aslında verileri hesaplar veya işler. Kontrol birimi, verilerin ve talimatların CPU içine ve dışına hareketini kontrol etmenin yanı sıra ALU'nun çalışmasını da kontrol eder. Şekil 3.40, kayıt/yazmaç (register) denilen bir dizi saklama konumundan oluşan minimal bir dahili belleğin temel yapısını göstermektedir.



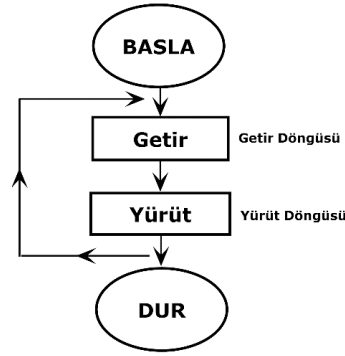
Şekil 3.40. Sistem veriyolu ve CPU gösterimi.

### 3.7.6.3. Kontrol Döngüsü

Şekil 3.41'de CPU komut döngüsünün yapısı basit bir görsel ile verilmiştir. Ayrıca bir komut döngüsü aşağıda verilen iki temel işlevi yerine getirir:

*Getir (Fetch)*: Bir sonraki talimatı bellekten CPU'ya okur.

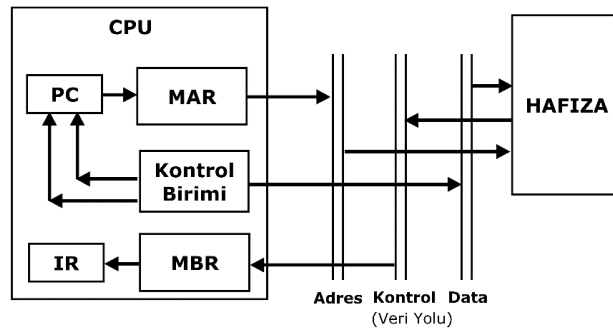
*Çalıştır (Execute)*: Opkodu yorumlar ve belirtilen işlemi yapar.



Şekil 3.41. Komut döngüsü.

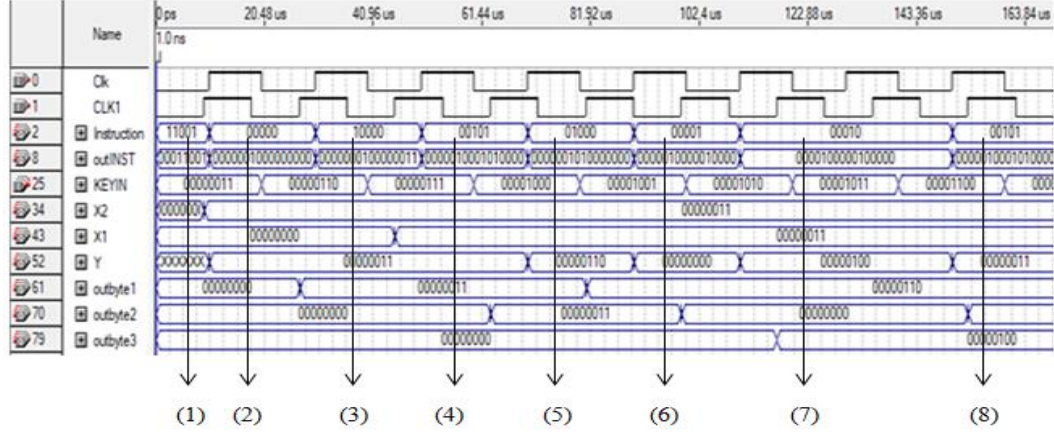
### 3.7.6.4. Veri Akışı

Bir komut döngüsü sırasındaki olayların tam sırası CPU tasarımına bağlıdır. Şekil 3.42, bu döngüdeki veri akışını göstermektedir. Getirme işlemi sırasında, bir talimat bellekten okunur. PC, alınacak bir sonraki talimatın adresini içerir. Bu adres MAR'ye taşınır ve adres yoluna yerleştirilir. Kontrol birimi hafıza okumasını talep eder ve sonuç veri yoluna yerleştirilir, MBR'ye kopyalanır ve daha sonra IR'ye taşınır. Bu arada, PC bir sonraki adıma geçer.



Şekil 3.42. Veri akışı ve getirme döngüsü.

CPU tasarımına başarıyla derlendikten sonra, tasarımın doğrulanması gerektiğini önceki kesimlerde de vurgulamıştık. Benzer şekilde simülör spesifik CPU doğrulamasını oluşturmak için kullanır. İlgili doğrulama girdi-çıkı verileri Şekil 3.43 ve Tablo 3.13'te verilmiştir.



Şekil 3.43. CPU doğrulama.

Tablo 3.13. Adresleme komutları.

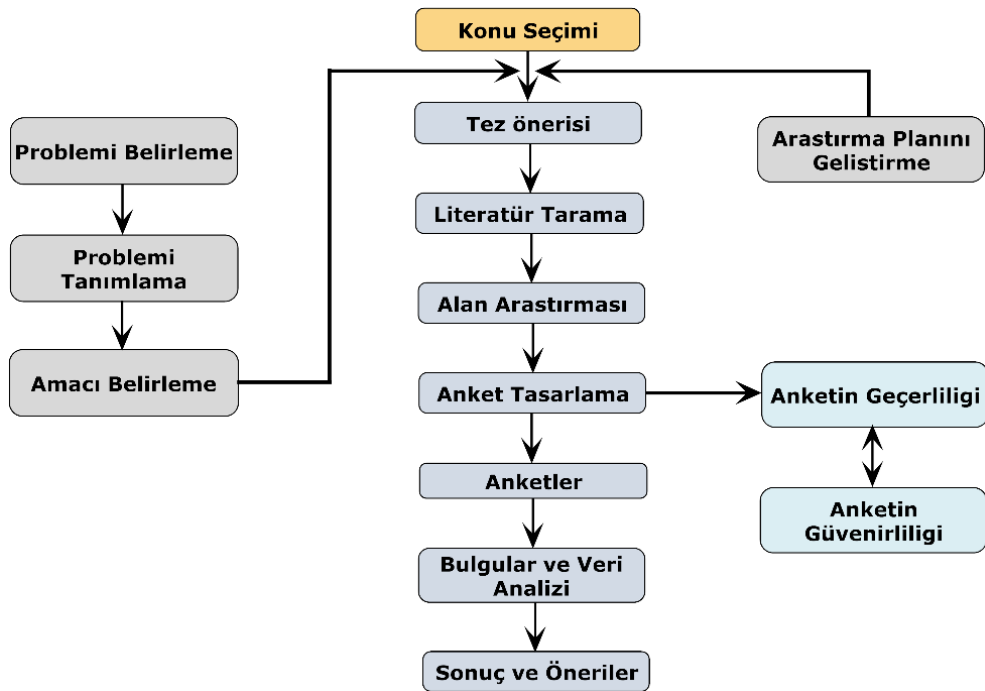
Adresleme komutları	Talimatın detayları
11001	Klavyeden okur ve ondan sonra X2 registerında saklar
00000	ADD: X1 ve X2 registerlerinin eklenmesi, bundan sonra sonucu bellekte saklar (bayt bir).
10000	Bellekten (bayt bir) veriyi X1 registerına yükler.
00101	OR: mantıksal işlem X1 ve X2 register bitlerine uygulanır ve bundan sonra sonucu bellekte depolar (bayt iki).
01000	SL: (Sola kaydırma) talimatı, X1 registerının içeriğini soldaki bir bit'e (bayt bir) kaydırır.
00001	SUB: X1 ve X2 registerlerinin çıkarır ve bundan sonra sonucu bellekte saklar (bayt iki).
00010	INC, X1 registerının değerini birer birer artırır ve bundan sonra sonucu hafızaya kaydeder (byte üç).
01001	SR (sağa kaydırma) komutu, X1 registerının içeriğini bir bit sağa (bayt iki) kaydırır.

### 3.8. Araştırma Süreci

Araştırmanın tasarım aşamaları şu şekilde özetlenebilir:

- Çalışmanın ilk aşaması, problemleri tanımlamayı ve çalışmanın hedeflerini belirlemeyi ve geliştirme planını yapmayı içerir.
- Çalışmanın ikinci aşaması, kapsamlı bir literatür taraması özeti içermektedir.
- Çalışmanın üçüncü aşaması, 162 öğrenciden oluşan bir saha çalışmasını içermektedir.
- Çalışmanın dördüncü aşaması, anketin en uygun biçimde oluşturulması ve pilot çalışma için dağıtılması ve verilerin analizini kapsamaktadır. Pilot çalışmanın amacı, ankette sorulan soruların, çalışmanın amaçlarına ulaşılmasına yardımcı olacak şekilde cevaplamanın kolay olup olmadığını test etmek ve anlamaktır. Anket pilot çalışmadan elde edilen veriler ile en verimli ve uygun şekilde revize edilmiştir.
- Çalışmanın beşinci aşaması, her dönemin bitiminden sonra anketlerin dağıtılması ve anket sonuçlarından elde edilen verilerin analizi üzerine kuruludur.
- Son aşama, sonuçları ve önerileri içerir.

Şekil 3.44, çalışma hedeflerine ulaşmak için kullanılan metodolojinin akış şemasını göstermektedir.



Şekil 3.44. Çalışma metodolojinin akış şeması.

### **3.9. Araştırma Stratejisi**

Simülasyon temelli yaklaşım, öğrencilerin bilgisayar mühendisliği dersi için donanım programlarında pratik eğitim almalarını sağlamak için bazı donanım tasarımı ve öğretim yöntemlerini entegre ederek geliştirilmiştir. Ayrıca, bir CPU, VHDL dili kullanılarak tasarlanmış ve simüle edilmiş ve bu süreç FPGA'ya PBL ve ML yöntemlerini uygulayarak gerçekleştirilmiştir.

Bu çalışma için araştırma soruları tümdengelimli bir bakış açısına sahiptir. Bu çalışma için en mantıklı stratejinin zaman kısıtlamaları ve düşük bütçe nedeniyle nicel araştırma yöntemi olduğu sonucuna ulaşılmış ve çalışma planı bu doğrultuda hazırlanmıştır.

### **3.10. Veri Toplama**

Farklı maliyet, hız ve içerik kapsamı kombinasyonları sunan çeşitli veri toplama yöntemleri mevcuttur; dolayısıyla, en uygun araştırma yöntemini seçmek çok önemlidir. Öğrencilerin kafasında oluşan, anket, ev ödevi, ara sınav ve final sınavı gibi başlangıçta içeriği net olmayan bu geri bildirim (ya da veri toplama) yöntemleri hakkında bilgiler almak için çeşitli yollar mevcuttur. Öte yandan, veri toplama yöntemleri araştırma felsefesinin benimsenmesine, araştırma yaklaşımına, araştırma stratejisine, genel olarak araştırmanın amaçlarına veya hedeflerine dayanırken, veri toplamak için kullanılan en yaygın yöntemler anketler ve görüşmelerdir. Öğrencilerin notları, ödevleri, ara sınavları ve final sınavları sonuçları araştırma sorularını yanıtlamak için kullanıldı. Anketler ise diğer değişkenlere veri toplamak için kullanılmıştır.

### **3.11. Veri Analizi**

Bu çalışmada nicel yaklaşım kullanılmıştır. Veri toplama kaynakları olarak öğrencilerin notları ve anket formları değerlendirilmiştir. İstatistiksel hesaplamalar için katılımcıların verileri istatistiksel programlarla analiz edilmiştir. Verilerin araştırılmasında grafikler ve çıkarımsal ölçümler kullanılmıştır. Bu incelemedeki aydınlatıcı görüşler, en düşük ve en uç değer anlamına gelen frekans dağılımlarını,

yüzdeleri, standart sapmayı, ortalamaları, karşılaştırmalı analizleri ve post-hoc analizini içermektedir.

### 3.12. Projede Kullanılan Yöntemler ve Testler

Projede kullanılan yöntem ve test türleri Tablo 3.14'te gösterilmektedir. Görüldüğü üzere geleneksel yöntemde, küçük bir proje, karmaşık bir proje ve ev ödevi kullanılmamıştır.

İkinci yöntemde (FPGA+ML), hem küçük hem de karmaşık bir proje kullanılmamıştır.

Üçüncü yöntemde (FPGA+PBL) sınav ve ödev kullanılmamıştır. Son olarak dördüncü yöntemde (FPGA+ML+PBL) sadece quiz yapılmamıştır.

Tablo 3 14. Projede kullanılan yöntemler ve testler.

Ölçme biçimi	TR	ML	PBL	ML+PBL
<i>Quiz</i>	EVET	EVET	HAYIR	HAYIR
<i>Küçük proje</i>	HAYIR	HAYIR	EVET	EVET
<i>Karmaşık proje</i>	HAYIR	HAYIR	EVET	EVET
<i>Ev ödevleri</i>	HAYIR	EVET	HAYIR	EVET
<i>Faktörler</i>	EVET	EVET	EVET	EVET
<i>Ara sınav</i>	EVET	EVET	EVET	EVET
<i>Final sınavı</i>	EVET	EVET	EVET	EVET

### 3.13. Çalışmada İncelenen Faktörler

Bu çalışma kapsamında uygulanan yeni öğretim yöntemlerinin değerlendirilmesinde “öğretim faktörü, çekicik faktörü, geçerlik faktörü, yeterlik faktörü ve karmaşıklık faktörüne izin verme” olarak adlandırılan beş faktör analiz edilmiştir.

Tablo 3.15’da bu faktörlerin yöntemlerle olan kombinasyonları verilmiştir.

Tablo 3.15. Çalışmada incelenen faktörler.

Faktör	Tanım	Yöntemler (Karşılaştırmalı)
<i>Öğretim</i>	Bir öğrencinin kısa veya uzun vadeli eğitim hedeflerine ulaşma derecesi	
<i>Çekicilik</i>	Öğrencinin yeteneğine güvenmesinin değerlendirilmesi	
<i>Geçerlik</i>	İlginçlik, motive edicilik, heyecan vericilik ve sıkıcı olmama özelliği	TR, ML, PBL, ML+PBL
<i>Yeterlik</i>	Planlama, çözme veya uygulama becerisi	
<i>Karmaşıklığa izin verme</i>	Etkili, hızlı öğrenme ve karmaşık projelerle başa çıkma becerisi	

### 3.14. Çalışmaya Katılanlar

Bu çalışma yayınlanmış araştırmalar, makaleler, belgeler ve ilgili diğer literatür çalışmalarının derin bir analizi sonucunda, özellikle bilgisayar bilimlerinde uygulana gelen geleneksel eğitim sistemindeki eksikliklerin giderilmesi yönünde yeni bir yöntem arayışının zorunluluğu olarak ortaya çıkmıştır. Bu çalışmanın öğrenciler üzerindeki uygulamaları Turgut Özal Üniversitesi ve Kastamonu Üniversitesi'nde yapılmıştır. Araştırmaya katılanlar: Bilgisayar Mühendisliği Bölümü, Bahar dönemi-2016 (46 öğrenci) ve Güz dönemi-2016 (43 öğrenci) öğrencileri ve de İleri Elektronik Dijital Tasarım Dersi Elektrik Mühendisliği Yüksek Lisans öğrencileri, Bahar dönemi-2017 (39 öğrenci) ve Güz dönemi-2017 (34 öğrenci).

### 3.15. Çalışmadaki Karşılaştırmalar

Bu çalışma kapsamında elde edilen veriler için dört kısımda karşılaştırma yapılmıştır: İlk karşılaştırma türü ankete ve 5 puanlık Likert skalasındaki puanlamaya göre (1 ile 5 arasında değişmektedir; burada 5 en iyi (kesinlikle katılıyorum) ve 1 ise en kötü (katılmıyorum) şeklindedir) yapılmıştır. Kullanılan anket bu araştırmaya özel olarak tasarlanmıştır. Sorular analiz edilecek beş faktörü de kapsayacak şekilde hazırlanmıştır ve her faktörün kendine özgü bir soru seti vardır:

- Öğretim faktörü (26 Soru).
- İlgi Çekicilik faktör (24 Soru).
- Geçerlik faktörü (24 Soru).

- Yeterlik faktörü (35 Soru).
- Karmaşıklık faktörüne izin verme (27 Soru).

İkinci karşılaştırma ödevlere dayandırıldı: İkinci grup ve dördüncü grup üç ödev ve üç seviye için değerlendirildi.

- Ödev I; Aynı konu derste de gösterildi.
- Ödev II; Yazılım ve ara dönem projesini (orta seviye) içermektedir.
- Ödev III; Karmaşık proje içermektedir.

Üçüncü karşılaştırma ara sınav verilerine göre yapıldı (Bir yarıyılıda bir arasınava yapılmıştır).

Dördüncü ve son olarak dönem sonunda yapılan final sınavı verilerine göre karşılaştırmalar yapılmıştır.

### 3.16. Çalışmanın Güvenilirliği

Bu araştırmada, her bir soru ve bütün arasındaki korelasyon, yani madde-toplam korelasyonu (item-total correlation) hiçbir soru için düşük/zayıf çıkmamıştır; bu nedenle bu çalışma kapsamında kullanılan ölçek uygulanabilir. Diğer bir ifadeyle, Cronbach alfa katsayısı yüksek olup iç tutarlılık derecesinin iyi olduğu görülmüştür. Çalışmanın faktörlerinin (anket üzerinden çalışılan faktörler) Cronbach Alfa güvenilirlik katsayıları 0.834 ile 0.889 arasında hesaplanmıştır (Tablo 3.16).

Tablo 3.16. Faktörlerin Cronbach Alfa güvenilirlik katsayısı.

Faktörler	Öğrenci sayısı	Cronbach Alfa katsayısı
<i>Öğretim</i>	26	0.889
<i>Çekicilik</i>	24	0.834
<i>Geçerlik</i>	35	0.839
<i>Yeterlik</i>	35	0.842
<i>Karmaşıklığa izin verme</i>	30	0.839

### **3.17. Altera ile Tasarım**

Altera tasarım ortamı, tasarımcılara, çeşitli platformlarda kullanılan farklı mimarilere sahip aygıtları destekleyen kullanımı kolay bir arayüz sağlayan tasarım giriş yöntemlerini ve doğrulama araçlarını kullanma özgürlüğü veren tamamen entegre bir tasarım ortamıdır. Altera tasarım süreci, kısaca bu bölümde sunulan tasarım girişi, tasarım derlemesi, tasarım doğrulaması ve cihaz programlamasından oluşur.

### **3.18. Quartus II**

Bir tasarımda *doğrulama* gerekli bir süreçtir, çünkü başarılı bir derleme bir cihazın programlanması için bir dosyanın yaratılabileceğini garanti eder. Simülasyon, üzerinde uğraşılan projenin doğru şekilde çalışacağını doğrulamanın en kolay yoludur. Simülasyon, tamamen test edilmiş bir tasarıma izin verir ve bir cihazın programlamadan önce olası her duruma doğru tepki vermesini sağlar. Bu çalışmada Quartus II simülatörü kullanılmıştır.

## **4. BULGULAR**

Bu çalışma, sorularla test, ev ödevleri yoluyla test (V1, V2, V3), ara sınavla test ve final sınavları ile test olmak üzere öğrencilere dörk farklı test uygulamıştır. Testlerden elde edilen veriler istatistik analiz yöntemiyle değerlendirilmiş ve yorumlanmıştır.

İstatistiksel analizde, test yönteminin türüne karar vermek için normallik testinin uygulanmasına ihtiyaç vardır. Normal dağılım için çeşitli faktörlerin uygunluğunun incelenmesi gerekir; çarpıklık (skewness) ve basıklık (kurtosis) değerleri yapılan testler ve ödevler için hesaplanmıştır. Bu kısımda faktörlerin analizleri ve verilerin değerlendirmesi yapılacaktır.

### **4.1. Test Faktörleri**

Bu çalışmada ortaya konulan öğretim yöntemlerinin analizinde kullanılan test faktörleri daha önceki kesimlerde de söylendiği üzere öğretim, çekicik, geçerlik, yeterlik ve karmaşıklığa izin verme faktörleridir. Bu kesimde uygulanan eğitim yöntemlerinin analizi için bu faktörlerin bir incelemesi yapılacaktır.

#### **4.1.1. Öğretim Faktörü**

##### **4.1.1.1. Normallik Testi**

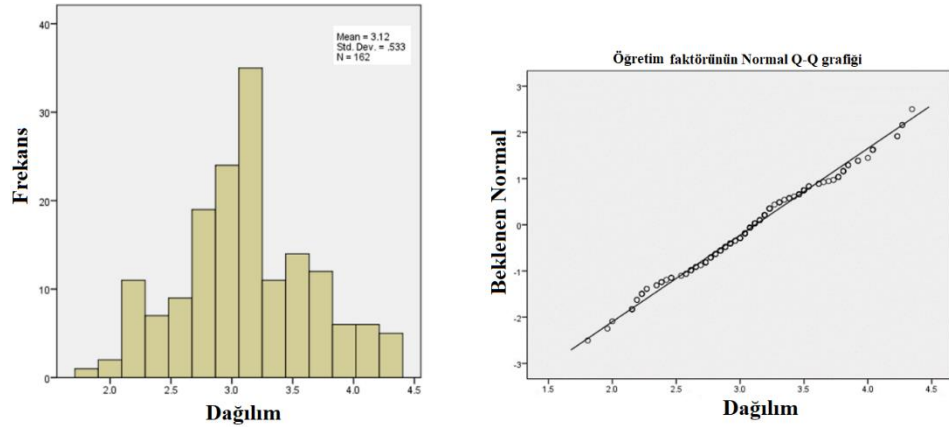
İstatistiksel teknik, numune verilerinin bir normal dağılım eğrisi oluşturup oluşturamayacağını araştırmak için kullanılmıştır. Tablo 4.1'den görüldüğü üzere, mevcut veri seti normal olarak dağılmıştır. Ayrıca, Tablo 4.1, Shapiro-Wilk'in sıfır hipotezini (null hypothesis) destekleyen  $0.150 > 0.05$  olasılık değerine sahip olduğunu göstermektedir. Sıfır hipotezi (ki, örnek normal dağılmış) burada geçerlidir (Ruppert, 2014). Bu aynı zamanda örnek verilerin normal popülasyon verilerinden istatistiksel olarak farklı olmadığı anlamına gelir. Normallik testi bulguları, parametrik test istatistiklerinin söz konusu hipotezleri test etmek için uygulanabileceğini göstermektedir. Ayrıca, normal dağılım durumunda basıklık

(kurtosis) ve çarpıklık (skewness) verilerin normal dağılımı durumu için -2 ile +2 arasında olmalıdır (George ve Mallery 2016). Tablo 4.1, hem eğriliğin hem de kurtozis değerlerinin -2 ile +2 aralığında olduğunu göstermektedir. Çarpıklık ve basıklık değerleri (0.047, -0.259) oldukça normal dağılıma sahip verileri göstermektedir.

Tablo 4.1. Öğretim Faktörü açısından Normallik testi.

Shapiro-Wilk		
İstatistik	df	Sig.
0.987	162	0.150
Çarpıklık	0.047	
Basıklık	-0.259	

Öğretici faktör verilerinin normal grafiği ve dağılım histogramı normal dağılım histogramına oldukça yakın bulunmuştur (Şekil 4.1).



Şekil 4.1. Dağılım histogramı ve öğretim faktörünün normal Q-Q grafiği.

#### 4.1.1.2. Güvenirlilik Analizi

Güvenirlilik testleri, öğretim ölçeği ve Tablo 4.3'te belirtilen Cronbach alfa hesaplaması kullanılarak gerçekleştirilmiştir. Korelasyon, ölçeğin kalan sorularına ilişkin değerleri hesaplamak için "Düzeltilmiş Madde-Toplam Korelasyon" işlemi aracılığıyla öğretim ölçeğine göre (Field, 2009) hesaplanmıştır. Bir soru için madde-toplam korelasyonunun düşük olması durumunda, soru, çalışma kavramını ölçemez. Mevcut araştırma, madde toplam korelasyonunun hiçbir soru için düşük olmadığını

göstermektedir; bu nedenle, mevcut ölçek uygundur.

Öğretim faktörünün derecesini bulmak için bir ifadeler koleksiyonu kullanılmıştır. Katılımcılardan, uyum düzeylerini Likert ölçeği ile belirtmeleri istenmiştir. Tablo 4.2’den de görüldüğü gibi ankete katılanlara yirmi altı soru sorulmuş ve öğretim faktörü hesaplanmıştır. Her bir soruya verilen cevapların ortalama ve standart sapma değerleri de Tablo 4.3’te verilmiştir. Hesaplamalar, Soru 1’in ortalama 2.93 (aynı fikirde olmayan %41.4) ve 1.154 standart sapma ile en düşük dereceye ulaştığını göstermektedir (Tablo 4.2). Soru 11, ortalama 3.40, standart sapma 1.077 ve %68 aynı fikirde olma yüzdesi ile en üst sırada yer almıştır.

Tablo 4.2. Öğretim faktörü Madde-Toplam istatistik değerleri.

Soru	Düzeltilmiş Madde-Toplam Korelasyonu	Cronbach Alfa (Soru silinirse)	N (number of studens)	Ortalama	Stn. Sapma
Q1	0.446	0.885	162	2.93	1.154
Q2	0.428	0.885	162	2.99	1.045
Q3	0.544	0.883	162	2.96	0.974
Q4	0.401	0.886	162	3.19	1.099
Q5	0.494	0.884	162	3.18	1.131
Q6	0.392	0.886	162	2.98	1.033
Q7	0.548	0.882	162	3.06	1.250
Q8	0.419	0.885	162	3.01	1.012
Q9	0.366	0.887	162	3.18	1.142
Q10	0.470	0.884	162	3.25	0.978
Q11	0.471	0.884	162	3.40	1.077
Q12	0.466	0.884	162	3.20	1.039
Q13	0.510	0.883	162	3.22	1.102
Q14	0.414	0.886	162	3.20	1.097
Q15	0.312	0.888	162	3.19	1.165
Q16	0.376	0.886	162	3.14	1.021
Q17	0.408	0.886	162	3.04	1.063
Q18	0.390	0.886	162	3.07	1.058
Q19	0.368	0.887	162	3.24	1.032
Q20	0.453	0.885	162	3.21	1.012
Q21	0.446	0.885	162	3.04	0.965
Q22	0.422	0.885	162	3.13	1.081
Q23	0.430	0.885	162	3.07	1.013
Q24	0.359	0.887	162	2.98	1.066
Q25	0.477	0.884	162	3.10	1.149
Q26	0.536	0.883	162	3.17	1.197

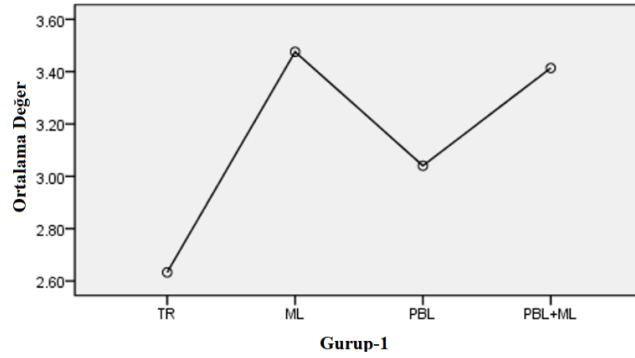
#### 4.1.1.3. ANOVA Analizi

Tek yönlü varyans analizi (ANOVA), üç veya daha fazla bağımsız (ilişkisiz) grubun ortalamaları arasında istatistiksel olarak anlamlı bir fark olup olmadığını belirlemek için kullanılır. Çalışma kapsamında uygulanan öğretim yöntemleri arasındaki ilişki ANOVA analizi ile değerlendirilmiştir.

TR, ML, PBL ve ML+PBL öğretim yöntemleri arasında anlamlı farklılık olup olmadığını bulmak için ANOVA analizi yapılmıştır. ML+PBL ve ML'nin 3.41 ve 3.47 ile en yüksek ortalamalara sahip olduğu hesaplanırken geleneksel (TR) yönteminin 2.63 ile en düşük ortalamaya sahip olduğu görülmüştür. PBL ise bu yöntemler arasında 3.04 ortalamaya sahiptir (Tablo 4.3, Şekil 4.2).

Tablo 4.3. Öğretim faktörü için tanımlayıcı istatistikler.

Yöntem	N	Ortalama	Stn. Sapma
TR	46	2.63	0.404
ML	43	3.47	0.316
PBL	39	3.04	0.314
ML+PBL	34	3.41	0.575



Şekil 4.2. Öğrenim yöntemlerinin öğretim faktörüne göre ortalamaları.

Tablo 4.4, ANOVA tablosu kullanılarak doğrulanan TR, ML, PBL ve ML+PBL teknikleri arasındaki ana farkları gösterirken, ortalama farklar %5 anlamlılık düzeyinde anlamlı bulunmuştur.

Tablo 4.4. Öğretim faktörü ANOVA Analizi

	Kareler Toplamı	df	Ortalama Kare	F	Sig.
Gruplar Arası	19.543	3	6.514	39.334	0.000
Grup İçi	26.168	158	0.166		
Toplam	45.711	161			

#### 4.1.1.4. Varyans Eşitliği - Levene Testi

Bu test, örneklerin eşit farklılık gösterip göstermediğini belirlemek için kullanılır. Bağımsız ölçümler gibi istatistiksel testler, varyansların örnekler arasında eşit olmasını gerektirir. Levene testi, bu gereksinimin karşılandığını kontrol etmenizi sağlar.

Levene testi ML, TR, ML+PBL ve PBL'nin eşit/eşit-olmayan varyansa sahip olup olmadığını anlamak için yapılmıştır. Tablo 4.5, Levene'nin testinin  $F(3,158) = 11.728$ ,  $p < 0.01$ 'daki önemini göstermektedir. Bu nedenle, sıfır hipotezi reddedilmiştir, ki bu da PBL, ML, TR ve ML+PBL'nin önemli ölçüde farklı varyanslara sahip olduğu anlamına gelir. Bu durumda, varyansların homojenliği yoktur. Bu sonuçlar karşısında, Levene'in buradaki test değeri için kullanışlı olan Games-Howell testi olarak adlandırılan (Allen, 2017), benzer olmayan varyans için alternatif prosedürler kullanıldı.

Tablo 4.5. Öğretim faktörüne göre farklılıkların homojenliği testi.

Levene İstatistik	df1	df2	Sig.
11.728	3	158	0.000

Games-Howell testi, Tablo 4.6'de sırasıyla ML, ML+PBL ve PBL'nin (ortalamalar sırasıyla 3.47, 3.41 ve 3.04, Tablo 4.3' bak) daha büyük ortalama değerlere sahip olduğunu ve ortalama değerlerin %0.05 anlamlılık düzeyinde ve TR ortalamasından daha yüksek olduğunu ortaya koymuştur. ML ve ML+PBL'ye ait ortalamalar, PBL ortalamasına göre daha yüksektir. Ayrıca ML yönteminin diğer yöntemlere göre öğreticilik faktörü açısından daha başarılı olduğu görülmektedir. ML ile ML+PBL yöntemleri arasında öğreticilik faktörü açısından kayda değer bir fark da görülmemiştir ( $I-J = 0.062$ ).

Geleneksel eğitimin zayıf olduğu her ne kadar orta da olsa da PBL yönteminin de

diğer yöntemlere nazaran pek başarılı olduğu söylenemez. Bunun nedeninin PBL yönteminde öğrencilerin bireysel anlamda daha fazla çalışmak zorunda kalmaları, daha fazla zamana ihtiyaç duymaları ve geleneksel öğrenme biçimlerinin dışına çıkmaya olan direç gösterilebilir.

Tablo 4.6. Çoklu Karşılaştırmalar-Games-Howell Testi.

Grup-1 (I)	Grup-1 (J)	Ortalama Farkı (I-J)	Stn. Hata	Sig.
TR	ML	-0.843*	0.077	0
	PBL	-0.407*	0.078	0
	ML+PBL	-0.781*	0.115	0
ML	TR	0.843*	0.077	0
	PBL	0.435*	0.070	0
	ML+PBL	0.062	0.110	0.942
PBL	TR	0.407*	0.078	0
	ML	-0.435*	0.070	0
	ML+PBL	-0.374*	0.111	0.008
ML+PBL	TR	0.781*	0.115	0
	ML	-0.062	0.110	0.942
	PBL	0.374*	0.111	0.008

\* Ortalama farkı 0.05 seviyesinde anlamlıdır.

#### 4.1.2. Çekicilik Faktörü

##### 4.1.2.1. Normallik Testi

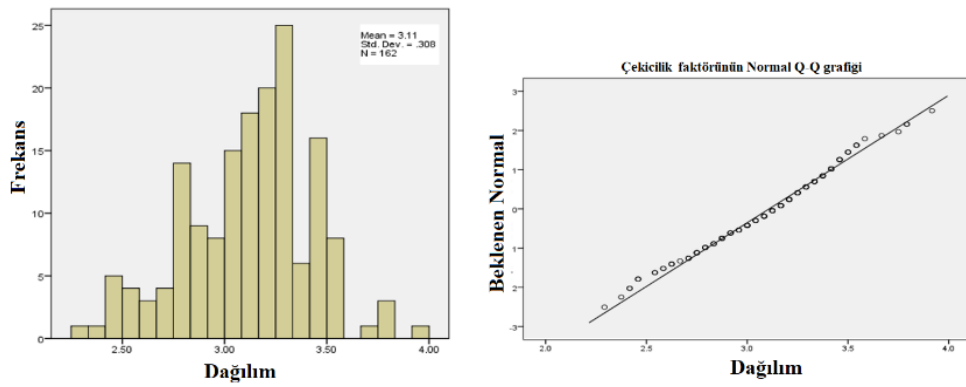
Tablo 4.7'te görüldüğü gibi, mevcut veri seti normal olarak dağılmıştır. Shapiro-Wilk'in, sıfır hipotezi destekleyen  $0.140 > 0.05$  olasılık değerine sahip olduğu görülmüştür ve sıfır hipotezi burada geçerlidir. Normallik testi bulguları, parametrik test istatistiklerinin söz konusu hipotezleri test etmek için uygulanabileceğini göstermektedir.

Ayrıca, normal dağılım durumunda basıklık ve çarpıklık değerleri daha önce de ifade edildiği gibi -2 ile +2 arasında olmalıdır, ki bulunan çarpıklık ve basıklık değerleri (-0.275, -0.031) oldukça normal dağılım gösteren verileri göstermektedir; böylece, temel parametrik test varsayımı yerine getirilmiştir.

Tablo 4.7. Çekicilik Faktörü için Normallik testi.

Shapiro-Wilk		
İstatistik	df	Sig.
0.987	162	0.140
Çarpıklık	-0.275	
Basıklık	-0.031	

İlgi çekicilik faktörü verilerinin normal grafiği ve dağılım histogramı normal dağılım histogramına oldukça yakın bulundu (Şekil 4.3).



Şekil 0.1. Dağılım histogramı ve çekicilik faktörün normal eğrisi.

#### 4.1.2.2. Güvenlilik Analizi

Cronbach alfa katsayısı çekicilik ölçeği 0.834 olarak hesaplanmıştır (Tablo 4.8). Alfa değeri yüksek iç tutarlılık göstermektedir ve bu da ölçeğin uygulanabilir olduğu anlamına gelir.

Tablo 4.8. Çekicilik faktörü güvenlilik analizi.

Güvenlilik İstatistikleri	
Cronbach Alfa	Soru Sayısı
0.834	24

Güvenlilik testleri, çekicilik ölçeği ve Tablo 4.8'da belirtilen Cronbach alfa hesaplaması kullanılarak yapılmıştır. Korelasyon, ölçeğin kalan sorularına ilişkin değerleri hesaplamak için "Düzeltilmiş Madde-Toplam Korelasyon" işlemi yoluyla İlgi Çekicilik ölçeğine göre hesaplanmıştır.

İlgi çekicilik faktörün derecesini bulmak için katılımcılardan, uyum düzeylerini Likert ölçeği ile belirtmeleri istenmiştir. Tablo 4.9’da da verildiği gibi, katılımcıların çekicilik faktörüne ilişkin tutumlarını ölçmek için yirmi dört soru sorulmuştur. Her bir soruya verilen cevaplar kullanılarak ortalama ve standart sapma hesaplanmıştır.

Hesaplamalar, Soru 8 için ortalamanın 4,81 (yüzde %43,8 kabul etmeme/olumsuz), standart sapmanın ise 0,962 olarak en düşük değerleri verdiğini göstermiştir (Tablo 4.9). Bunun yanısıra Soru 6’nın ortalama değeri 3.25, standart sapması da 1.111 (%65 olumlu) ile en yüksek değerleri almıştır.

*Tablo 4.9. Çekicilik faktörü için madde-toplam istatistiği verileri.*

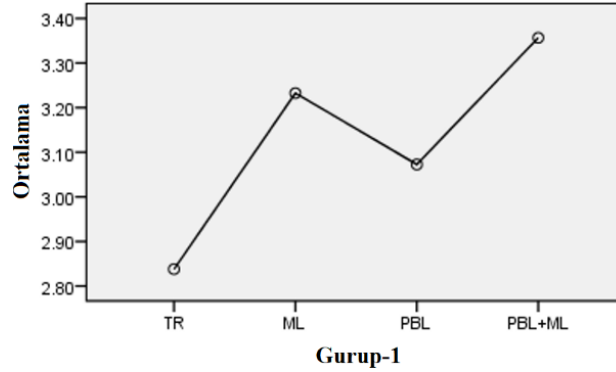
Soru	Düzeltilmiş Madde-Toplam Korelasyonu	Cronbach Alfa (Soru silinirse)	N	Ortalama	Stn. Sapma
Q1	0.383	0.828	162	2.95	0.983
Q2	0.458	0.825	162	2.95	0.989
Q3	0.369	0.828	162	3.03	1.060
Q4	0.368	0.828	162	3.12	1.054
Q5	0.326	0.830	162	3.09	0.968
Q6	0.346	0.829	162	3.25	1.111
Q7	0.322	0.830	162	3.20	1.002
Q8	0.290	0.831	162	2.81	0.962
Q9	0.388	0.827	162	2.98	1.097
Q10	0.481	0.824	162	3.18	0.925
Q11	0.416	0.826	162	3.14	0.975
Q12	0.329	0.829	162	3.05	0.957
Q13	0.362	0.828	162	2.93	1.000
Q14	0.426	0.826	162	3.25	1.041
Q15	0.304	0.831	162	3.08	1.142
Q16	0.313	0.830	162	3.08	0.991
Q17	0.349	0.829	162	2.93	1.046
Q18	0.322	0.830	162	3.09	1.054
Q19	0.447	0.825	162	3.08	0.997
Q20	0.382	0.828	162	3.11	1.003
Q21	0.363	0.828	162	2.91	0.981
Q22	0.391	0.827	162	3.10	1.007
Q23	0.359	0.828	162	3.10	1.055
Q24	0.250	0.832	162	2.91	0.890

#### 4.1.2.3. ANOVA Analizi

TR, ML, PBL ve ML+PBL yöntemleri arasında anlamlı bir farklılık olup olmadığını bulmak için ANOVA analizi yapılmıştır. ML+PBL ve ML ortalamaları 3.35 ve 3.23'tür, TR ve PBL ise sırasıyla 2.83 ve 3.07 ortalamalarına sahiptirler (Tablo 4.10 ve Şekil 4.4). Bu sonuçlar TR, ML, PBL ve ML+PBL yöntemleri arasında önemli farklar olduğunu göstermektedir.

Tablo 4.10. Çekicilik faktörünün tanımlayıcı istatistikleri.

Yöntem	N	Ortalama	Stn. Sapma
TR	46	2.83	0.266
ML	43	3.23	0.218
PBL	39	3.07	0.255
ML+PBL	34	3.35	0.206
Toplam	162	3.10	0.307



Şekil 4.4. Çekicilik faktörüne göre öğretim yöntemleri.

Tablo 4.11, ANOVA tablosu kullanılarak doğrulanan ML, TR, ML+PBL ve PBL teknikleri arasındaki ana farkları gösterirken, ortalama farklar %5 anlamlılık düzeyinde anlamlı bulunmuştur.

Tablo 4.11. Çekicilik faktörü, ANOVA verileri.

	Kareler Toplamı	df	Ortalama Kare	F	Sig.
Gruplar Arası	6.174	3	2.058	35.877	0.000
Grup İçi	9.064	158	0.057		
Toplam	15.238	161			

#### 4.1.2.4. Varyans Eşitliği - Levene Testi

TR, ML, PBL ve ML+PBL'nin eşit (ya da eşit olmayan) varyanslara sahip olup olmadığını anlamak için Levene testi yapılmıştır. Levene'nin  $F(3,158) = 1.855$ ,  $p > 0.01$  olarak hesaplanmıştır (Tablo 4.12), ki bu durumda sıfır hipotezi kabul edilir ve TR, ML, PBL ve ML+PBL varyansları eşittir. Bu nedenle, Levene testinin değeri anlamsız olduğundan ve homojenlik varyansı varsayımı ihlal edilmediğinden, Tukey Testi'nin kullanımı (Tablo 4.13) uygundur.

Tablo 4.12. Çekicilik faktörüne göre varyansların homojenliği testi.

Levene İstatistik	df1	df2	Sig.
1.855	3	158	0.140

df1: aktif olayan öğrenci, df2: aktif öğrenci

ML+PBL ve ML ortalama değerleri (sırasıyla 3.35 ve 3.23), %0.05 anlamlılık düzeyinde TR ve PBL ortalamalarından daha yüksek hesaplanmıştır. İlgili çekicilik açısından ML+PBL yönteminin daha başarılı sonuçlar verdiği görülmüştür. Öğretim faktörü sonuçlarına benzer olarak TR ve PBL yöntemlerinin daha az ilgi çekici olduğu görülmüştür. İlgili çekicilik açısından ML ile ML+PBL arasındaki fark çok küçük olsa da ( $I - J = 0.12406$ , sig. = 0.013) diğer yöntemlerle olan fark verilen anlamlılık düzeyinde göze çarpmaktadır.

Tablo 4.13. Çekicilik faktörünün çoklu karşılaştırmaları - Tukey testi.

Grup-1 (I)	Grup-1 (J)	Ortalama Farkı (I-J)	Stn. Hata	Sig.
TR	ML	-0.39470*	0.050	0.000
	PBL	-0.23479*	0.052	0.000
	ML+PBL	-0.51876*	0.054	0.000
ML	TR	0.39470*	0.050	0.000
	PBL	0.15991*	0.052	0.015
	ML+PBL	-0.12406	0.054	0.113
PBL	TR	0.23479*	0.052	0.000
	ML	-0.15991*	0.052	0.015
	ML+PBL	-0.28397*	0.056	0.000
ML+PBL	TR	0.51876*	0.054	0.000
	ML	0.12406	0.054	0.113
	PBL	0.28397*	0.056	0.000

\* Ortalama fark 0.05 düzeyinde anlamlıdır.

Proje bazlı uygulamaların ML ile birleştirilmesinin öğrencinin daha fazla ilgisini çektiği görülmekle birlikte öğrencilerin proje çalışmalarına olan isteksizliği yine dikkat çekicidir.

### 4.1.3. Geçerlik Faktörü

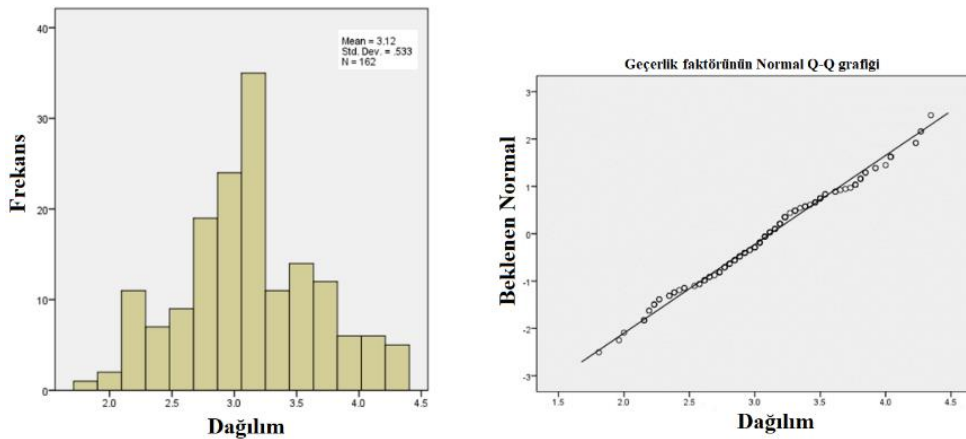
#### 4.1.3.1. Normallik Testi

Tablo 4.14'te görüldüğü gibi, mevcut veri seti normal olarak dağılmıştır. Shapiro-Wilk'in olasılık değeri  $0.304 > 0.05$  ile sıfır hipotezini desteklemektedir ve burada uyulanabilir. Ayrıca, normal dağılım durumunda basıklık ve çarpıklık değerleri normal şekilde dağıldığını kanıtlamak için -2 ile +2 arasında olmamalıdır. Hesaplamalarımıza göre bulunan çarpıklık ve basıklık değerleri (-0.010, -0.491) oldukça normal dağılım gösteren verileri göstermektedir.

Tablo 4.14. Geçerlik faktörü için normallik testi verileri.

Shapiro-Wilk		
İstatistik	df	Sig.
0.990	162	0.304
Çarpıklık	-0.010	
Basıklık	-0.491	

Geçerlik Faktörü verilerinin normal grafiği ve dağılım histogramı, normal dağılım histogramına oldukça yakın bulunmuştur (Şekil 4.5).



Şekil 4.5. Dağılım histogramı ve geçerlik faktörünün normal grafiği.

#### 4.1.3.2. Güvenirlilik Analizi

Cronbach alfa katsayısı (efektif ölçek) 0.839 olarak hesaplanmıştır ve yüksek iç tutarlılık göstermektedir, bu da uygulanan ölçeğin yeterli olduğu anlamına gelir.

Tablo 4.15. Geçerlik faktörü için madde-toplam istatistiği.

Soru	Düzeltilmiş Madde-Toplam Korelasyonu	Cronbach Alfa (Soru silinirse)	N	Ortalama	Stn. Sapma
Q1	0.292	0.836	162	2.73	0.819
Q2	0.425	0.833	162	2.93	1.049
Q3	0.415	0.833	162	3.07	1.052
Q4	0.330	0.835	162	3.00	0.905
Q5	0.460	0.832	162	2.98	0.916
Q6	0.414	0.833	162	3.00	0.984
Q7	0.223	0.838	162	3.14	0.936
Q8	0.228	0.838	162	2.99	0.997
Q9	0.078	0.842	162	3.04	0.977
Q10	0.389	0.834	162	3.20	0.947
Q11	0.375	0.834	162	3.15	0.982
Q12	0.240	0.838	162	2.96	1.005
Q13	0.255	0.837	162	2.96	0.980
Q14	0.304	0.836	162	2.87	0.966
Q15	0.181	0.839	162	2.78	0.984
Q16	0.286	0.837	162	3.09	0.948
Q17	0.236	0.838	162	2.93	0.930
Q18	0.049	0.843	162	3.02	0.949
Q19	0.258	0.837	162	3.25	0.993
Q20	0.402	0.833	162	3.14	1.068
Q21	0.191	0.839	162	3.03	0.942
Q22	0.258	0.838	162	3.12	1.050
Q23	0.326	0.836	162	3.02	1.000
Q24	0.314	0.836	162	2.98	0.987
Q25	0.337	0.835	162	2.99	0.936
Q26	0.462	0.832	162	3.01	1.051
Q27	0.435	0.833	162	3.06	0.950
Q28	0.421	0.833	162	2.92	1.009
Q29	0.450	0.832	162	2.95	0.938
Q30	0.486	0.831	162	3.10	1.076
Q31	0.161	0.840	162	3.14	0.984
Q32	0.156	0.840	162	2.94	0.989
Q33	0.327	0.835	162	2.99	0.959
Q34	0.359	0.835	162	3.02	0.942
Q35	0.446	0.832	162	3.07	1.066

Güvenirlik testleri, etkili ölçek ve Cronbach alfa katsayısı hesaplamaları kullanılarak gerçekleştirildi. Mevcut araştırma, madde toplam korelasyonunun hiçbir soru için düşük olmadığını ve bu nedenle uygun olduğunu ortaya koymuştur.

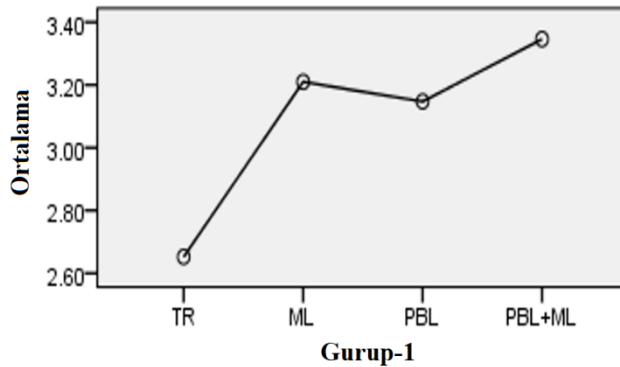
Hesaplamalar, Soru 1'in ortalama 4,73 (yüzde %45.4 aynı fikirde olmama) ve standart sapma 0,819 ile en düşük değerleri verdiğini göstermektedir (Tablo 4.15). Soru 19 ise ortalama 3.25 ve standart sapma 0.993 (%65 aynı fikirde olma yüzdesi) ile en yüksek sıralamayı aldı.

#### 4.1.3.3. ANOVA Analizi

TR, ML, PBL ve ML+PBL teknikleri arasında anlamlı farklılık olup olmadığını bulmak için, daha önceki kesimde de yapıldığı gibi, ANOVA analizi yapılmıştır. Hesaplanan değerler Tablo 4.16'de verilmiştir. ML+PBL ve ML'nin daha yüksek ortalama değerlere sahip olduğu görülmüştür (sırasıyla 3.34 ve 3.21). TR ve PBL'nin ise daha düşük (sırasıyla 2.65 ve 3.14) ortalama değerlerini verdiği görülmüştür (Tablo 4.18, Şekil 4.6).

Tablo 4.16. Geçerlik faktörünün tanımlayıcı istatistikleri.

Yöntem	N	Ortalama	Stn. Sapma
TR	46	2.65	0.244
ML	43	3.21	0.244
PBL	39	3.14	0.312
ML+PBL	34	3.34	0.380
Toplam	162	3.06	0.397



Şekil 4.6. Geçerlik faktörüne göre öğretim yöntemleri.

Tablo 4.17, ANOVA tablosu kullanılarak doğrulanan ML, TR, ML+PBL ve PBL teknikleri arasında büyük farklılıklar olduğunu gösterirken, ortalama farklar %5 düzeyinde anlamlı bulunmuştur.

Tablo 4.17. Geçerlik faktörü için ANOVA testi.

	<b>Kareler Toplamı</b>	<b>df</b>	<b>Ortalama Kare</b>	<b>F</b>	<b>Sig.</b>
Gruplar Arası	11.693	3	3.898	44.905	0.000
Grup İçi	13.714	158	0.087		
Toplam	25.407	161			

#### 4.1.3.4. Varyans Eşitliği - Levene Testi

Levene'in  $F(3,158) = 5.174$ ,  $p < 0.01$  dir (Tablo 4.18) ve bu nedenle sıfır hipotez reddedilmiştir. ML, TR, ML+PBL ve PBL önemli ölçüde farklı varyanslara sahiptir ve bu durumda varyansların homojenliği yoktur.

Tablo 4.18. Geçerlik faktörü, varyansların homojenliği testi.

<b>Levene İstatistik</b>	<b>df1</b>	<b>df2</b>	<b>Sig.</b>
5.174	3	158	0.002

Games-Howell Testi (Tablo 4.19), sırasıyla ML+PBL, ML ve PBL'nin daha büyük ortalama değerlere sahip olduğunu göstermektedir: 3.34, 3.21 ve 3.14, ortalama değerleri (Tablo 4.16), %0.05 anlamlılık düzeyinde TR ortalamasından daha yüksektir (Tablo 4.21). Aynı şekilde, ML+PBL, ML ve PBL arasındaki ortalama farklar %0.05 düzeyinde anlamlıdır. ML+PBL ve ML araçları, PBL ortalamasına göre daha yüksektir.

Geçerlik faktörü yöntemin etkililiği ile doğrudan ilgili olup ML+PBL yönteminin diğer yöntemlere zayıf bir üstünlüğü söz konusu olmuştur. Bu faktör açısından da ML+PBL en başarılı sonuçları alırken (Tablo 4.19, satır 4, ML+PBL'nin diğer yöntemlerle olan karşılaştırma verilerine bak) onu, daha önceki faktörlere benzer olarak, sırasıyla ML ve PBL yöntemleri takip etmiştir. Geleneksel yöntem geçerlilik faktörü açısından da en zayıf öğretim yöntemi olarak ortaya çıkmaktadır.

Tablo 4.19. Geçerlik faktörünün çoklu karşılaştırmaları - Games-Howell testi.

Grup-1 (I)	Grup-1 (J)	Ortalama Farkı (I-J)	Stn. Hata	Sig.
TR	ML	-0.55779*	0.051	0.000
	PBL	-0.49508*	0.061	0.000
	ML+PBL	-0.69388*	0.074	0.000
ML	TR	0.55779*	0.051	0.000
	PBL	0.06271	0.062	0.747
	ML+PBL	-0.13608	0.075	0.281
PBL	TR	0.49508*	0.061	0.000
	ML	-0.06271	0.062	0.747
	ML+PBL	-0.19880	0.082	0.084
ML+PBL	TR	0.69388*	0.074	0.000
	ML	0.13608	0.075	0.281
	PBL	0.19880	0.082	0.084

#### 4.1.4. Yeterlik Faktörü

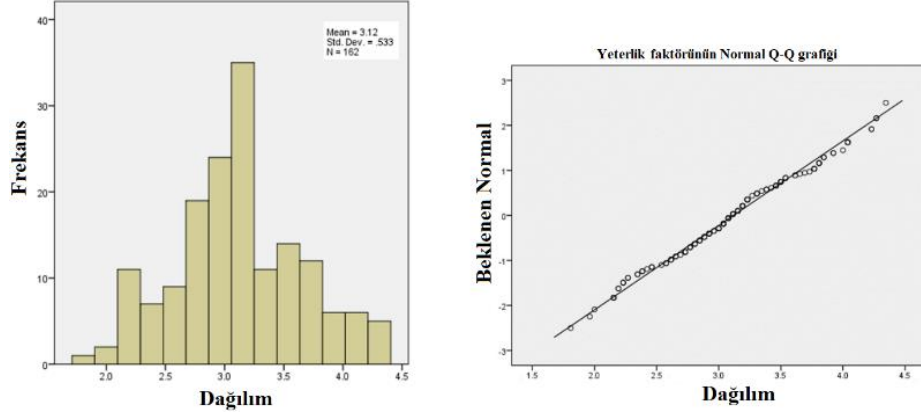
##### 4.1.4.1. Normallik Testi

Tablo 4.20'den görüleceği üzere, mevcut veri seti normal olarak dağılmıştır ve de Shapiro-Wilk  $0.317 > 0.05$  olasılık değerine sahiptir ve sıfır hipotezi desteklenir. Çarpıklık ve basıklık değerleri (0.130, 0.155) oldukça normal dağılım gösteren verileri göstermektedir.

Tablo 4.20. Yeterlik faktörüne göre Normallik testi.

Shapiro-Wilk		
İstatistik	df	Sig.
0.990	162	0.317
Çarpıklık	0.130	
Basıklık	0.155	

Yeterlik faktörü verilerinin normal grafiği ve dağılım histogramı normal dağılım histogramına oldukça yakın bulunmuştur (Şekil 4.7).



Şekil 4.7. Dağılım histogramı ve yeterlik faktörü normal grafiği.

#### 4.1.4.2. Güvenirlilik Analizi

Öğrencilerin yeterlik faktörü ile ilgili tutumlarını ölçmek için otuz beş soru sorulmuştur. Cronbach alfa katsayısı 0,842 olarak hesaplanmıştır. Elde edilen veriler neticesinde yapılan hesaplama sonuçları Tablo 4.21’te verilmiştir. Ortalama ve standart sapma açısından en alt ve yüksek değerler sırasıyla Soru 1 ve Soru 26 için hesaplanmıştır.

Tablo 4.21. Yeterlik faktörü için Madde-Toplam istatistikleri.

Soru	Düzeltilmiş Madde- Toplam Korelasyon	Cronbach Alfa (Soru silinirse)	N	Ortalama	Stn. Sapma
Q1	0.402	0.836	162	2.66	1.047
Q2	0.341	0.838	162	2.86	0.942
Q3	0.340	0.838	162	2.95	0.964
Q4	0.383	0.836	162	2.75	0.948
Q5	0.351	0.837	162	2.86	1.027
Q6	0.392	0.836	162	2.96	0.948
Q7	0.297	0.839	162	2.88	0.938
Q8	0.307	0.838	162	2.86	0.945
Q9	0.256	0.840	162	2.81	1.013
Q10	0.303	0.839	162	2.95	0.924
Q11	0.378	0.836	162	2.99	1.015
Q12	0.240	0.840	162	2.82	1.009
Q13	0.245	0.840	162	2.78	0.959
Q14	0.320	0.838	162	2.67	0.946
Q15	0.209	0.841	162	2.73	1.091
Q16	0.266	0.839	162	2.96	0.971
Q17	0.264	0.840	162	2.85	0.982

Tablo 4.21'in devamı.

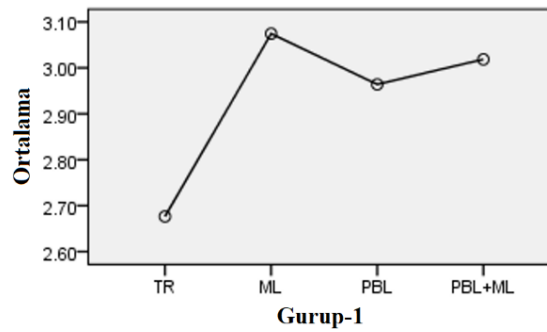
Q18	0.215	0.841	162	2.88	1.056
Q19	0.320	0.838	162	2.97	0.981
Q20	0.399	0.836	162	3.01	1.012
Q21	0.214	0.841	162	2.75	0.978
Q22	0.278	0.839	162	2.97	1.006
Q23	0.376	0.837	162	2.89	0.959
Q24	0.308	0.838	162	2.82	1.015
Q25	0.309	0.838	162	2.99	1.015
Q26	0.348	0.837	162	3.15	1.072
Q27	0.344	0.837	162	3.01	1.057
Q28	0.409	0.836	162	2.98	1.033
Q29	0.281	0.839	162	2.94	0.960
Q30	0.365	0.837	162	3.01	1.015
Q31	0.264	0.839	162	2.97	0.942
Q32	0.283	0.839	162	2.81	0.927
Q33	0.381	0.837	162	2.91	0.869
Q34	0.411	0.836	162	3.00	1.009
Q35	0.455	0.834	162	2.92	1.027

#### 4.1.4.3. ANOVA Analizi

Yeterlik faktörüne ait istatistik verileri Tablo 4.22’te verilmiştir. Ayrıca, uygulanan yöntemler arasındaki farklılığı ortaya koymak amacıyla ANOVA analizi yapılmıştır. Elde edilen veriler Tablo 4.23’da verilmiştir. Test sonuçları daha öncekilere benzer elde edilmiştir (Şekil 4.8).

Tablo 4.22 Yeterlik faktörünün tanımlayıcı istatistikleri.

Yöntem	N	Ortalama	Stn. Sapma
TR	46	2.67	0.189
ML	43	3.07	0.230
PBL	39	2.96	0.293
ML+PBL	34	3.01	0.217
Toplam	162	2.92	0.282



Şekil 4.8. Yeterlik faktörü açısından öğretim yöntemleri.

ANOVA verileri, ortalama farklar için %5 anlamlılık düzeyinde anlamlı sonuç vermiştir.

Tablo 4.23. Yeterlik faktörü ANOVA verileri.

	<b>Kareler Toplamı</b>	<b>df</b>	<b>Ortalama Kare</b>	<b>F</b>	<b>Sig.</b>
<b>Gruplar Arası</b>	4.159	3	1.386	25.249	0.000
<b>Grup İçi</b>	8.676	158	0.055		
<b>Toplam</b>	12.835	161			

#### 4.1.4.4. Varyans Eşitliği - Levene Testi

Tablo 4.24'den de görüleceği üzere Levene  $F(3,158) = 3.704$ ,  $p > 0.01$  ile sıfır hipotezi kabul edilmiştir. Yöntemlerin varyansları eşittir ve Tukey testi kullanılmıştır (Tukey çoklu karşılaştırma testi, bir dizi ortalama arasında hangi ortalamaların diğerlerinden farklı olduğunu belirlemek için kullanılacak birkaç testten biridir ve Tukey'in HSD'si de denir).

Tablo 4.24. Yeterlik faktörü, varyansların homojenliği testi.

<b>Levene İstatistik</b>	<b>df1</b>	<b>df2</b>	<b>Sig.</b>
3.704	3	158	0.013

Tablo 4 28 sırasıyla ML, ML+PBL ve PBL'nin daha büyük ortalama değerlere sahip olduğunu göstermektedir: Sırasıyla 3.07, 3.01 ve 2.96, ortalama değerleri (Tablo 4.22) %0.05 anlamlılık seviyesinde TR ortalamasından daha yüksektir. Aynı şekilde, ML, ML+PBL ve PBL arasındaki ortalama farklar %0.05 seviyesinde anlamlı değildir. Yeterlik faktörü açısından daha ilginç sonuçlar ortaya çıkmıştır. ML yönteminin sırasıyla ML+PBL ve PBL yöntemlerine olan üstünlüğü söz konusu olsa da bu üstünlük verilen düzeyde pek anlamlı değildir (sig. değerleri 0.726 ve 0.756 olarak hesaplandı, Tablo 4.25).

Şöyle ki, öğrencilerin daha alışık oldukları öğrenme yöntemi olan mikro öğrenmeye olan bağlılığı ML+PBL yöntemine karşı kırılındır, yani bu yeni yöntemle uyum sağlamaları, bu veriler ışığında, belki de tahmin edilenden çok daha kolay ve hızlı olacaktır. Bu, dikkate alınması gereken bir durumdur.

Tablo 4.25. Yeterlik faktörünün çoklu karşılaştırmaları - Games-Howell testi.

Tukey HSD				
Grup-1 (I)	Grup-1 (J)	Ortalama Farkı (I-J)	Stn. Hata	Sig.
TR	ML	-0.39802*	0.049	0.000
	PBL	-0.28771*	0.051	0.000
	ML+PBL	-0.34209*	0.053	0.000
ML	TR	0.39802*	0.049	0.000
	PBL	0.11032	0.051	0.148
	ML+PBL	0.05593	0.053	0.726
PBL	TR	0.28771*	0.051	0.000
	ML	-0.11032	0.051	0.148
	ML+PBL	-0.05438	0.054	0.756
ML+PBL	TR	0.34209*	0.053	0.000
	ML	-0.05593	0.053	0.726
	PBL	0.05438	0.054	0.756

\* Ortalama farkı 0.05 seviyesinde anlamlıdır.

#### 4.1.5. Karmaşıklığa İzin Verme Faktörü

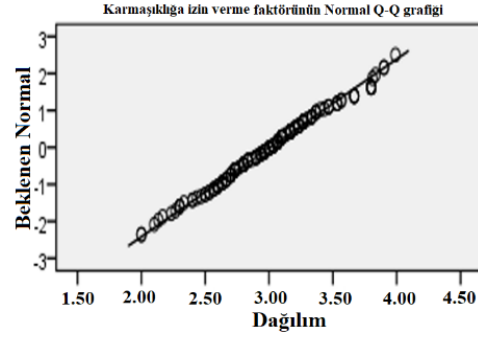
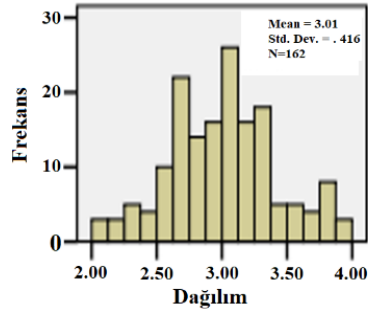
##### 4.1.5.1. Normallik Testi

Tablo 4.26'un gösterdiği gibi, mevcut veri seti normal olarak dağılmıştır. Shapiro-Wilk'in olasılık değeri  $0.297 > 0.05$  olup sıfır hipotezini desteklemektedir. Sıfır hipotezi (örnek normal dağılmıştır) burada geçerlidir. Çarpıklık ve basıklık değerleri (0.062, -0.128) normal dağılım gösteren verileri göstermektedir.

Tablo 4.26. Karmaşıklığa izin verme faktörü için Normallik testi.

	Shapiro-Wilk		
	İstatistik	df	Sig.
	0.990	162	0.297
Çarpıklık		0.062	
Basıklık		-0.128	

Karmaşıklığa izin verme faktörü verilerinin dağılım histogramı normal dağılım eğrisi vermiştir (Şekil 4.9).



Şekil 4.9. Dağılım histogramı ve karmaşıklıkça izin verme.

#### 4.1.5.2. Güvenirlilik Analizi

Cronbach alfa katsayısı (karmaşıklık ölçeğine izin verme) 0.839 olarak hesaplanmıştır. Alfa değeri yüksek iç tutarlılık göstermektedir, bu da ölçeğin, karmaşıklıkça değerlendirmek için yeterli olduğu anlamına gelir. Katılımcılardan, uyum düzeylerini Likert ölçeği ile belirtmeleri istenmiştir. Tablo 4.27'den de görüleceği üzere, katılımcıların izin verilen karmaşıklık faktörüyle ilgili tutumlarını ölçmek için otuz soru sorulmuştur. Her bir soruya verilen cevapların örnekleminde ortalamalar ve standart sapmalar hesaplanmıştır. Hesaplamalar, Soru 1'in ortalama 2.73 ve %45.4 katılmama yüzdesi ve de 0.819 standart sapma ile en düşük dereceyi aldığını göstermektedir. Soru 10 ise, ortalama 3.25, standart sapma 0.928 ve %65 olumlu kabul yüzdesi ile en yüksek dereceyi almıştır.

Table 0.27. Karmaşıklıkça izin verme faktörünün Madde-Toplam istatistiği.

Soru	Düzeltilmiş Madde-Toplam Korelasyonu	Cronbach Alfa (Soru silinirse)	N	Ortalama	Stn. Sapma
Q1	0.299	0.836	162	2.73	0.819
Q2	0.462	0.831	162	2.98	1.012
Q3	0.378	0.834	162	3.11	1.028
Q4	0.337	0.835	162	2.99	0.916
Q5	0.425	0.832	162	2.98	0.936
Q6	0.387	0.833	162	3.02	0.975
Q7	0.237	0.838	162	3.13	0.947
Q8	0.285	0.837	162	2.98	1.042
Q9	0.083	0.843	162	3.06	1.007
Q10	0.385	0.834	162	3.25	0.928
Q11	0.426	0.832	162	3.16	0.971
Q12	0.240	0.838	162	3.00	1.009
Q13	0.268	0.837	162	3.01	1.000
Q14	0.334	0.835	162	2.91	0.970

Tablo 4.27'nin devamı.

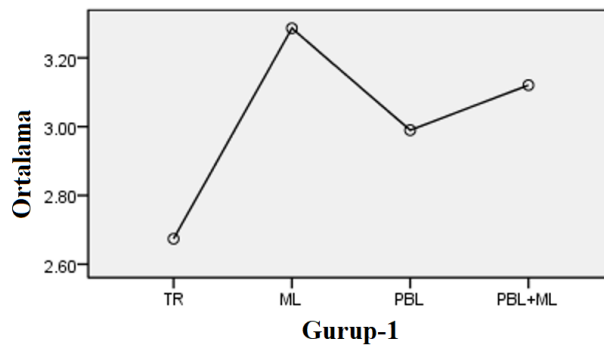
Q15	0.212	0.839	162	2.84	0.984
Q16	0.315	0.836	162	3.03	0.987
Q17	0.231	0.838	162	2.93	0.947
Q18	0.123	0.841	162	2.99	0.994
Q19	0.296	0.836	162	3.21	1.030
Q20	0.409	0.833	162	3.11	1.086
Q21	0.253	0.837	162	2.99	0.956
Q22	0.302	0.836	162	3.12	1.048
Q23	0.427	0.832	162	2.96	1.030
Q24	0.379	0.834	162	2.94	0.973
Q25	0.301	0.836	162	2.99	0.932
Q26	0.484	0.830	162	3.01	1.060
Q27	0.463	0.831	162	3.04	0.961
Q28	0.443	0.832	162	2.92	1.009
Q29	0.420	0.832	162	2.94	0.950
Q30	0.472	0.830	162	3.09	1.085

#### 4.1.5.3. ANOVA Analizi

TR, ML, PBL ve ML+PBL teknikleri arasında anlamlı farklılık olup olmadığını bulmak için daha önceki yapılanlara benzer olarak ANOVA analizi yapılmıştır. ML+PBL ve ML yöntemlerinin daha yüksek ortalamalara (sırasıyla 3.12 ve 3.28) sahip olduğu görülmüştür. TR ve PBL ise daha düşük ortalama değerleri almıştır (Tablo 4.28 ve Şekil 4.10).

Tablo 4.28. Karmaşıklıkta izin verme faktörü tanımlayıcı istatistikleri.

Yöntem	N	Ortalama	Stn. Sapma
TR	46	2.67	0.256
ML	43	3.28	0.321
PBL	39	2.98	0.448
ML+PBL	34	3.12	0.348
Toplam	162	3.00	0.416



Şekil 4.10. Karmaşıklık faktörüne göre öğretim yöntemleri.

Tablo 4.29, ANOVA tablosu kullanılarak doğrulanan ML, TR, ML+PBL ve PBL yöntemleri arasındaki ortalama farklar %0.05 anlamlılık düzeyinde anlamlı bulunmuştur

Tablo 4.29. Karmaşıklığa izin verme faktörü, ANOVA analzi verileri.

	<b>Kareler Toplamı</b>	<b>df</b>	<b>Ortalama Kare</b>	<b>F</b>	<b>Sig.</b>
Gruplar Arası	8.902	3	2.967	24.712	0.000
Grup İçi	18.972	158	0.120		
Toplam	27.874	161			

#### 4.1.5.4. Varyans Eşitliği - Levene Testi

Tablo 4.30, Levene  $F(3,158) = 3.153$ ,  $p > 0.01$  olduğu görülmüştür. Sıfır hipotezi kabul edilmiştir ve varyansların homojenliği varsayımı ihlal edilmemiştir. Daha önce de olduğu gibi burada Tukey testi uygulanmıştır.

Tablo 4.30. Karmaşıklık faktörü - homojenlik testi.

<b>Levene İstatistik</b>	<b>df1</b>	<b>df2</b>	<b>Sig.</b>
3.153	3	158	0.027

Tablo 4.31, sırasıyla ML, ML+PBL ve PBL'nin daha büyük ortalama değerlere sahip olduğunu göstermektedir (sırasıyla 3.28, 3.12 ve 2.98), ve de ortalama değerleri, %0.05 anlamlılık düzeyinde, TR ortalamasından daha yüksektir. Bununla birlikte, ML+PBL ve PBL arasındaki ortalama farkları %0.05 düzeyinde anlamlı değildir. Ayrıca bu düzeyde ML ile PBL arasında ML yönünde anlamlı bir fark görülmüştür.

Karmaşıklığa izin verme özelliği açısından ML yöntemi her ne kadar daha başarılı sonuçlar elde etmiş olsa da ML+PBL ile arasında anlamlı bir fark gözlenmemiştir ( $I-J = 0.16546$ , sig. = 0.164). Bu sonuç, öğrenciler her ne kadar projelere dayalı bir öğrenim görmeyi tercih etmek istemeseler de mikro öğrenmeyle birlikte verilen proje çalışmalarının öğrencileri daha karmaşık problemleri çözebilecekleri yönünde cesaretlendirdiğini ortaya koymaktadır.

Tablo 4.31. Karmaşıklığa izin vereme faktörü - çoklu karşılaştırmalar (Tukey HSD).

Grup-1 (I)	Grup-1 (J)	Ortalama Farkı (I-J)	Stn. Hata	Sig.
TR	ML	-0.61213*	0.073	0.000
	PBL	-0.31583*	0.075	0.000
	ML+PBL	-0.44668*	0.078	0.000
ML	TR	0.61213*	0.073	0.000
	PBL	0.29630*	0.076	0.001
	ML+PBL	0.16546	0.079	0.164
PBL	TR	0.31583*	0.075	0.000
	ML	-0.29630*	0.076	0.001
	ML+PBL	-0.13084	0.081	0.376
ML+PBL	TR	0.44668*	0.078	0.000
	ML	-0.16546	0.079	0.164
	PBL	0.13084	0.081	0.376

\* Ortalama farkı 0.05 düzeyinde anlamlıdır.

## 4.2. İkinci Bölüm: Ev Ödevlerine Dayalı Test

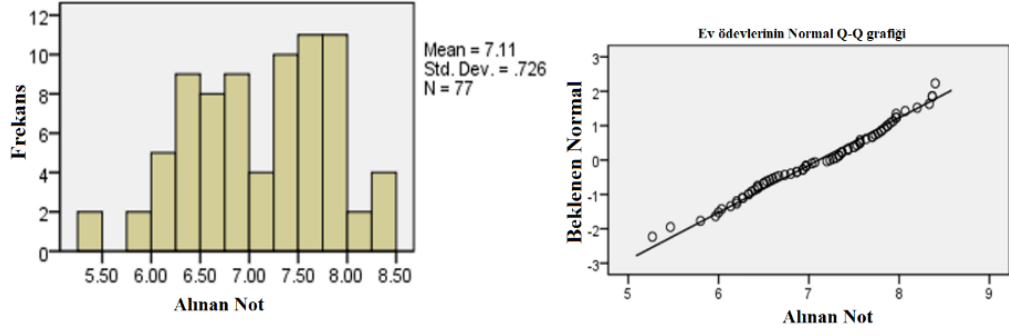
Ödevlere dayalı tüm metodolojilerin performansını test etmek için testler yapıldı. Ödev testi, ML ve ML+PBL yöntemleri arasında bir karşılaştırmayapmak açacıyla tasarlanmıştır.

### 4.2.1. Normallik Testi

Tablo 4.32'da görüldüğü gibi, mevcut veri seti normal olarak dağılmıştır. Shapiro-Wilk 0.168 > 0.05 olasılık değerine sahiptir ve sıfır hipotezi desteklenir. Çarpıklık ve basıklık değerleri (-0.271, -0.596) normal dağılım gösteren verileri göstermektedir (Şekil 4.11).

Tablo 4.32. Ev ödevleri için Normallik testi verileri.

	Shapiro-Wilk		
	İstatistik	df	Sig.
	0.977	77	0.168
Ev Ödevi 1, 2, 3	Çarpıklık		-0.271
	Basıklık		-0.596



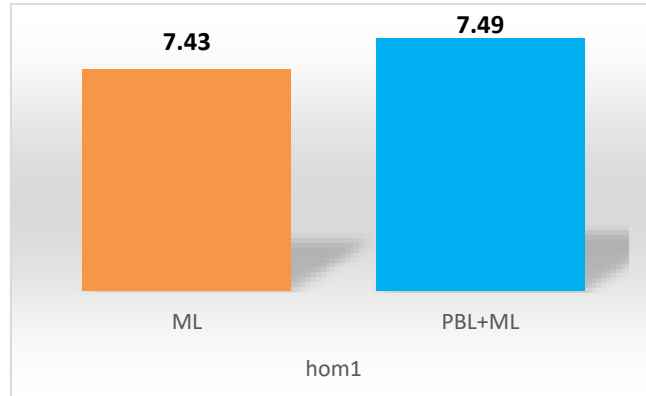
Şekil 4.11. Ev ödevlerine (1, 2 ve 3) ait dağılım histogramı.

#### 4.2.2. ANOVA Analizi

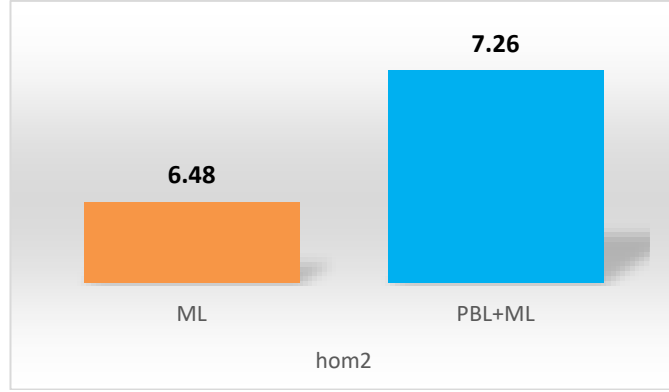
Burada ML ve ML+PBL arasındaki farklılıkları ölçmek için ANOVA analizi yapılmıştır. Ödevlerin ayrı ayrı analizleri yapıldığında (Tablo 4.33) yöntemler arasında düşük de olsa anlamlı bir fark olduğu görülmüştür.

Özellikle, Ev ödevi 1 ile ilgili olarak, ML+PBL'nin ML ile karşılaştırıldığında diğer ödevlerde olduğundan daha düşük bir fark olduğu görülmüştür.

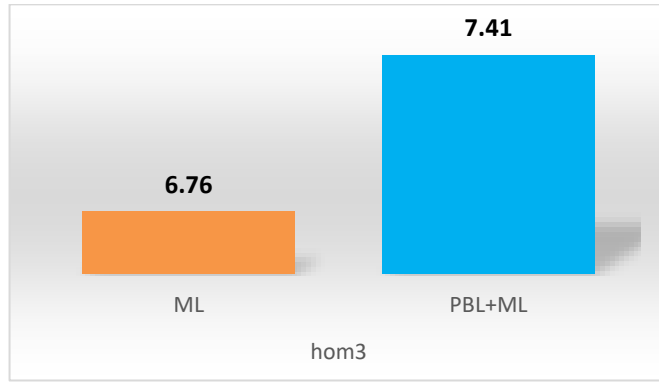
Ev ödevlerine ilişkin ortalama verilerinin karşılaştırması Şekil 4.12-14'te verilmiştir.



Şekil 4.12. Ödev 1 (hom1) için ML ve ML+PBL arasındaki fark.



Şekil 4.13. Ödev 2 (hom2) için ML ve ML+PBL arasındaki fark.



Şekil 0.14. Ödev 3 (hom3) için ML ve ML+PBL arasındaki fark.

Tablo 4.34'e göre, ev ödevi 1 ile ilgili olarak ML ve ML+PBL arasında anlamlı bir fark yoktur. Ancak, ANOVA verileri, ev ödevleri 2 ve 3 ile ilgili olarak ML ve ML+PBL arasındaki ortalama farkı 0.05 düzeyinde anlamlıdır.

Tablo 4.33. Ev ödevleri tanımlayıcı istatistikleri.

		N	Ortalama	Stn. Sapma
<b>E.Ö. 3</b>	ML	43	6.76	1.344
	ML+PBL	34	7.41	1.023
	Toplam	77	7.05	1.249
<b>E.Ö. 2</b>	ML	43	6.48	1.868
	ML+PBL	34	7.26	0.850
	Toplam	77	6.82	1.547
<b>E.Ö. 1</b>	ML	43	7.43	0.961
	ML+PBL	34	7.49	1.043
	Toplam	77	7.46	0.992

E.Ö. : Ev ödevi

Tablo 4.34. Ev ödevleri ANOVA verileri.

		Kareler Toplamı	df	Ortalama Kare	F	Sig.
<b>E.Ö. 1</b>	Gruplar Arası	0.064	1	0.064	0.064	0.801
	Grup İçi	74.686	75	0.996		
	Toplam	74.750	76			
<b>E.Ö. 2</b>	Gruplar Arası	11.476	1	11.476	5.052	0.028
	Grup İçi	170.347	75	2.271		
	Toplam	181.823	76			
<b>E.Ö. 3</b>	Gruplar Arası	8.039	1	8.039	5.460	0.022
	Grup İçi	110.432	75	1.472		
	Toplam	118.471	76			

E.Ö. : Ev ödevi

### 4.3. Üçüncü Bölüm: Test Bazlı Ara Sınav

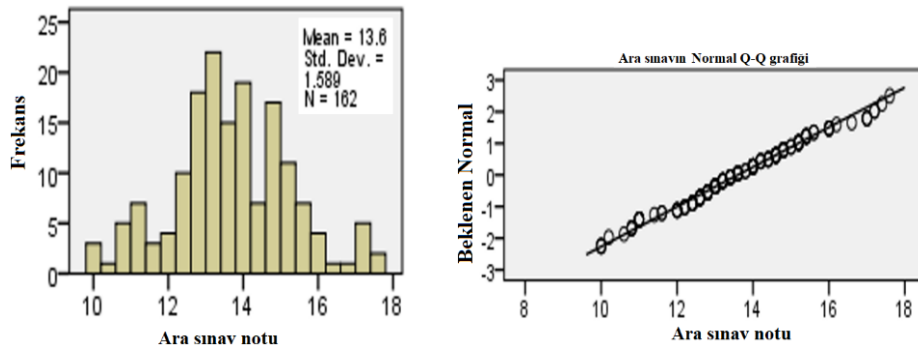
#### 4.3.1. Normallik Testi

Shapiro-Wilk olasılık değeri  $0,085 > 0,05$  olup sıfır hipotezi desteklenir (Tablo 4.35). Çarpıklık ve basıklık değerleri (0.071, 0.025) normal dağılım gösteren verileri göstermektedir.

Tablo 4.35. Ara sınav için Normallik testi.

Shapiro-Wilk		
İstatistik	df	Sig.
0.985	162	0.085
Çarpıklık	0.071	
Basıklık	0.025	

Ara sınav verilerinin dağılım histogramı normal dağılım histogramına oldukça yakın bulunmuştur (Şekil 4.15).



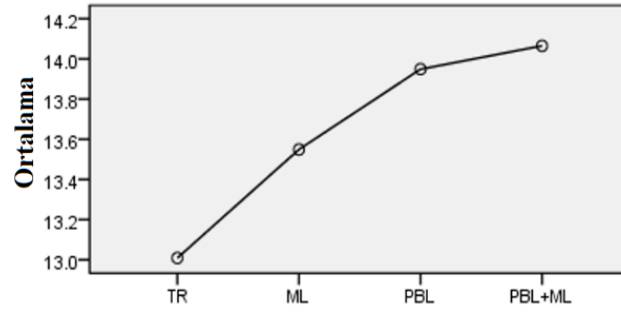
Şekil 4.15. Dağılım histogramı ve ara sınavın normal dağılımı.

### 4.3.2. ANOVA Analizi

TR, ML, PBL ve ML+PBL teknikleri arasında anlamlı farklılık olup olmadığını bulmak için ANOVA analizi yapılmıştır. Tablo 4.36'ta, ML+PBL ve PBL'nin daha yüksek ortalamalara (sırasıyla 14.06 ve 13.95) sahip olduğu, TR ve ML'nin sırasıyla 13.01 ve 13.55 ortalama değerlerine sahip oldukları görülmüştür (Şekil 4.16).

Tablo 4.36. Ara sınavın tanımlayıcı istatistikleri.

Yöntem	N	Ortalama	Stn. Sapma
TR	46	13.01	1.536
ML	43	13.55	1.723
PBL	39	13.95	1.487
ML+PBL	34	14.06	1.384
Toplam	162	13.60	1.589



Şekil 4.16. Ara sınav sonuçlarına göre öğretim yöntemleri.

Tablo 4.37'deki ANOVA verilerine göre ML, TR, ML+PBL ve PBL yöntemleri arasındaki ortalama farkları %5 anlamlılık düzeyinde anlamlı bulunmuştur.

Tablo 4.37. Ara sınav ANOVA verileri.

	Kareler Toplamı	df	Ortalama Kare	F	Sig.
Gruplar Arası	28.281	3	9.427	3.939	0.010
Grup İçi	378.119	158	2.393		
Toplam	406.400	161			

### 4.3.3. Varyans Eşitliği - Levene Testi

, Levene  $F(3,158) = 1.168$ ,  $p > 0.01$  bulunmuş (Tablo 4.38) ve sıfır hipotezi kabul edilmiştir. Daha önceki kesimlerde olduğu gibi Tukey testi uygulanmıştır.

Tablo 4.38. Ara sınav için varyansların homojenliği testi.

Levene İstatistik	df1	df2	Sig.
1.168	3	158	0.324

Tablo 4.39, sırasıyla ML+PBL, PBL ve ML'nin daha büyük ortalama değerlere sahip olduğunu göstermektedir: Bu yöntemlerin sırasıyla 14.06, 13.95 ve 13.55 ortalama değerleri (Tablo 4.36), %0.05 anlamlılık düzeyinde, TR ortalamasından daha yüksektir. Aynı şekilde, ML+PBL, PBL ve ML arasındaki ortalama farkları %0.05 düzeyinde anlamlı değildir.

Tablo 4.39. Ara sınav verileri çoklu karşılaştırmaları.

Tukey HSD				
ID (I)	ID (J)	Ortalama Farkı (I-J)	Stn. Hata	Sig.
TR	ML	-0.540	0.328	0.356
	PBL	-0.940*	0.337	0.030
	ML+PBL	-1.056*	0.350	0.016
ML	TR	0.540	0.328	0.356
	PBL	-0.400	0.342	0.647
	ML+PBL	-0.516	0.355	0.468
PBL	TR	0.940*	0.337	0.030
	ML	0.400	0.342	0.647
	ML+PBL	-0.116	0.363	0.989
ML+PBL	TR	1.056*	0.350	0.016
	ML	0.516	0.355	0.468
	PBL	0.116	0.363	0.989

\* Ortalama farkı 0,05 düzeyinde anlamlıdır.

Arasınav sonuçları öğrencilerin verilen konuları ne kadar anladığını bir göstergesi olarak ML+PBL yönteminin küçük farkla da olsa sırasıyla PBL ve ML'den daha iyi sonuçlar verdiğini ortaya koymuştur.

İlginç bir nokta, proje temelli öğrenme gerçekleştiren öğrencilerin ML yöntemine kıyasla daha iyi sonuçlar elde etmesidir. Her ne kadar faktörler açısından PBL öğrenciler tarafından pek tercih edilmese de sınav başarıları söz konusu olduğunda PBL'nin ML'den daha iyi sonuçlar sergilediği görülmüştür (Tablo 4.39).

#### 4.4. Dördüncü Bölüm: Final Sınavına Dayalı Test

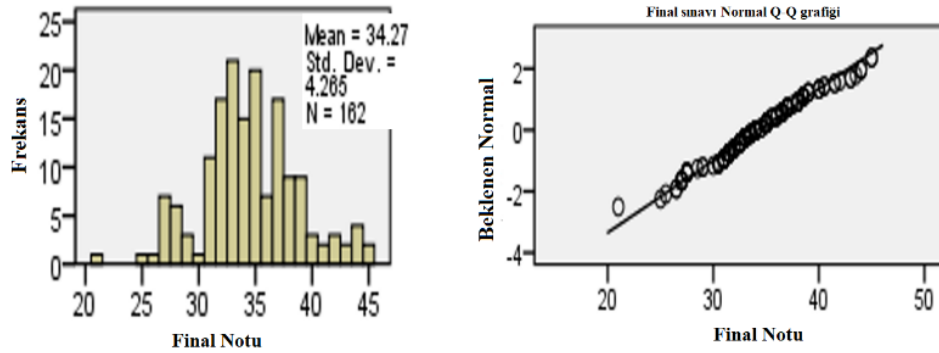
##### 4.4.1. Normallik Testi

Shapiro Wilk'in olasılık değeri  $0.061 > 0.05$  olup sıfır hipotezi desteklenir. Çarpıklık ve basıklık değerleri (0.094, -0.394) normal dağılıma sahip verileri göstermektedir (Tablo 4.40).

Tablo 4.40. Final sınavı için normallik testi.

Shapiro-Wilk		
İstatistik	df	Sig.
0.984	162	0.061
Çarpıklık	0.094	
Basıklık	0.394	

Öğretim faktörü verilerinin dağılım histogramı normal dağılım histogramına oldukça yakın çıkmıştır (Şekil 4.17).



Şekil 4.17. Dağılım histogramı ve final sınavının normal dağılımı.

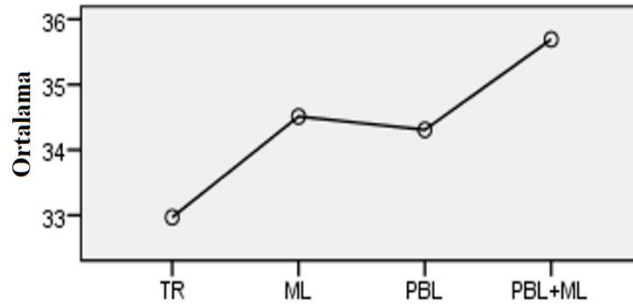
##### 4.4.2. ANOVA Analizi

ANOVA analizi, ML+PBL, ML ve PBL'nin, TR (32.97) ile karşılaştırıldığında yüksek ortalamalara (sırasıyla 35.69, 34.51 ve 34.31) sahip olduğunu göstermektedir (Tablo 4.41, Şekil 4.18).

Bu noktada da geleneksel yöntemin başarısının düşük olduğu görülmüştür.

Tablo 4.41 Final sınavının tanımlayıcı istatistikleri.

Yöntem	N	Ortalama	Stn. Sapma
TR	46	32.97	3.416
ML	43	34.51	5.272
PBL	39	34.31	3.830
ML+PBL	34	35.69	4.004
Toplam	162	34.27	4.265



Şekil 4.18. Final sınavına göre öğretim yöntemleri.

TR, ML, PBL ve ML+PBL arasındaki farklılıklar ANOVA tablosu ile doğrulanmıştır ve de farklılıklar 0.05 düzeyinde anlamlıdır (Tablo 4.42).

Tablo 4.42. Final sınavının ANOVA verileri.

	Kareler Toplamı	df	Ortalama Kare	F	Sig.
Gruplar Arası	149.289	3	49.763	2.830	0.040
Grup İçi	2778.760	158	17.587		
Toplam	2928.049	161			

#### 4.4.3. Varyans Eşitliği - Levene Testi

Tablo 4.43’de görüldüğü gibi, Levene  $F(3,158) = 2.778$ ,  $p > 0$  bulunmuş ve sıfır hipotezi kabul edilmiştir. TR, ML, PBL ve ML+PBL varyansları eşittir, bu nedenle, varyansların homojenliği varsayımı ihlal edilmemiştir. Böylece, Tukey testi uygulanmıştır.

Tablo 4.43. Final sınavı için varyansların homojenliği testi.

Levene İstatistik	df1	df2	Sig.
2.778	3	158	0.043

Tablo 4.44'e göre ortalamalar arasındaki fark 0,05 düzeyinde anlamlıdır ve de ML, PBL ve ML+PBL arasında anlamlı bir fark gözlenmemiştir. Bununla birlikte, TR yöntemi diğer yöntemlerden ayrılarak en düşük değerleri almıştır.

Final sınavı sonuçlarında da ML+PBL yönteminin az da olsa diğer yöntemlere bir üstünlüğünün ortaya çıktığı görülmüştür. Arasınavdan farklı olarak final sınavında ML öğretim gurubunun PBL öğretim gurubundan daha iyi sonuçlar verdiği görülmektedir (I-J = 0.204, sig. = 0.996), ki bu sonuç proje temelli öğrenimin zamanla öğrencileri çalışmaya teşvikte zorlandığı düşüncesini doğrulamaktadır. Yani öğrenme sürecinin ML yardımıyla öğrencileri sürekli proje stresinden uzaklaştırarak daha verimli bir çalışma dağılımına izin verdiği öngörülebilir.

Tablo 4.44. Final sınavının çoklu karşılaştırmaları (Tukey HSD).

ID (I)	ID (J)	Ortalama Farkı (I-J)	Stn. Hata	Sig.
TR	ML	-1.544	0.890	0.309
	PBL	-1.340	0.913	0.459
	ML+PBL	-2.724*	0.948	0.024
ML	TR	1.544	0.890	0.309
	PBL	.204	0.927	0.996
	ML+PBL	-1.180	0.962	0.612
PBL	TR	1.340	0.913	0.459
	ML	-.204	0.927	0.996
	ML+PBL	-1.383	0.984	0.497
ML+PBL	TR	2.724*	0.948	0.024
	ML	1.180	0.962	0.612
	PBL	1.383	0.984	0.497

\* Ortalama farkı 0.05 seviyede anlamlıdır.

#### 4.5. Bulgular ve Tartışma

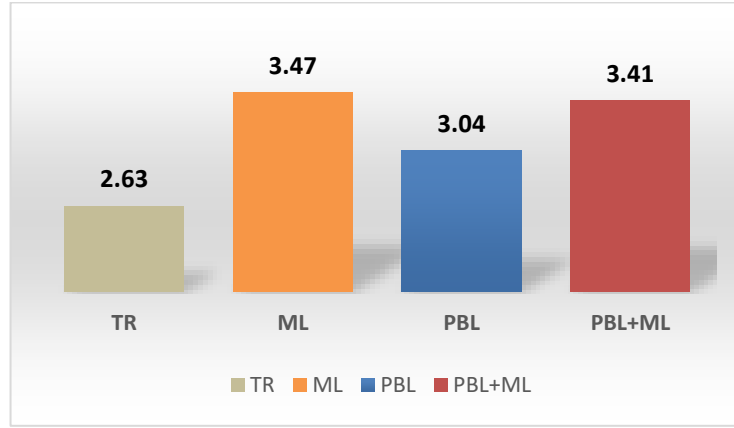
Hatırlanacağı üzere bu çalışmanın veri toplamak için kullanılan testleri: Anket testi, ödevlerle test (V1, V2, V3), ara sınav ile test ve de final sınavı ile test şeklindeydi.

Bu bölümde, bu çalışmanın ana sorularına cevap bulmak için öğretim metodolojileri arasında karşılaştırmalar yapılacak, sonuçlar gözden geçirilecek ve planlanan amaçlara ulaşıp ulaşılamadığı tartışılacaktır.

## 4.5.1. Anket Test Faktörleri

### 4.5.1.1. Öğretim Faktörü

Öğretim faktörü bu çalışmanın ilk ve önemli değişkenlerindedir. Yapılan anket çalışmasında verilen cevaplar, mikro öğrenme tekniğini uygulayarak öğrenen öğrencilerin (ortalama = 3.47, %69.4) puanla daha yüksek düzeyde bir eğitim aldıklarını kabul ettiklerini göstermiştir. Mikro öğrenme ve Proje tabanlı öğrenmenin birlikte uygulandığı yönteme ait anketin öğrenci puanlaması küçük bir farkla ML'den geride kalmıştır (ortalama = 3.41, yüzde 68.2). Ankete verilen yanıtlarda en düşük seviye geleneksel öğrenme içindir (ortalama = 2.63, %52.6). Şekil 4.19'da uygulanan yöntemlerin öğretim faktörü açısından ortalamaları verilmiştir.



Şekil 4.19. Öğretim faktörü açısından ortalamalar.

Eğitici faktör analizi, mikro öğrenme modelinin eğitim faktörü açısından geleneksel öğrenmeye göre %16.8 daha başarılı olduğunu, proje tabanlı öğrenmeye göre %8.6, ML+PBL'ye göre ise %1,2'lik küçük bir farkla daha yüksek değerlere sahip olduğunu ortaya koymuştur. Öğretim sisteminde büyük bir yeri olan geleneksel öğrenimin, öğretim faktörü açısından en az etkiye sahip olduğu açıkça görülmektedir.

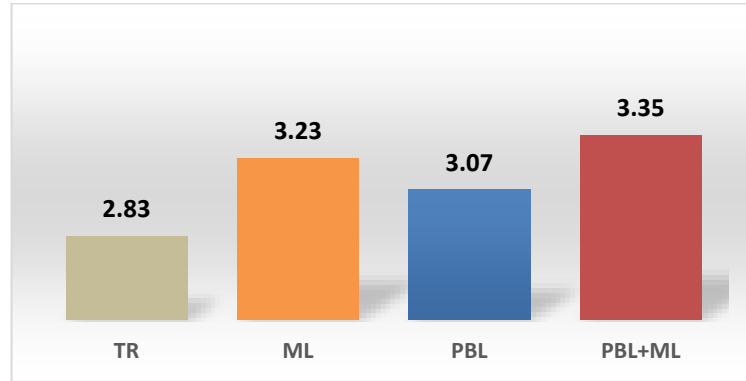
Öğretim faktörü verileri, öğrencilerin bilgisayar mimarisini incelemek için PBL ve TR öğrenme modellerine göre ML'yi tercih ettiklerini göstermektedir. Bununla birlikte, ML+PBL yönteminin öğretim faktörü de oldukça yüksek hesaplanmıştır, ki

bunların farklı dönemlerde aynı konuların değişik düzeylerine uygulanmaları öğrencilerin edinimleri üzerinde daha büyük etkilere sahip olabilir.

#### 4.5.1.2. Çekicilik Faktörü

Bu çalışmada göz önünde bulundurulması gereken ikinci faktör çekicilik faktördür ve unutulmamalıdır ki bu faktör öğrencilerin bilgi seviyelerine hassas bağlıdır.

Öğrenciler genel olarak ML+PBL yönteminin daha ilgi çekici olduğu yönünde bir tutum sergilemişlerdir (ortalama = 3.35, %67). Bununla birlikte ML yöntemi de tek başına uygulandığında ilgi çekicilik açısından hemen ML+PBL'nin arkasında yer almaktadır (ortalama = 3.23, %64.6). Geleneksel öğrenme ise öğrenciler tarafından ilgi çekiciliği en az bulunan yöntem olarak kabul edilmiştir (ortalama = 2.83, %56.6) (Şekil 4.20).



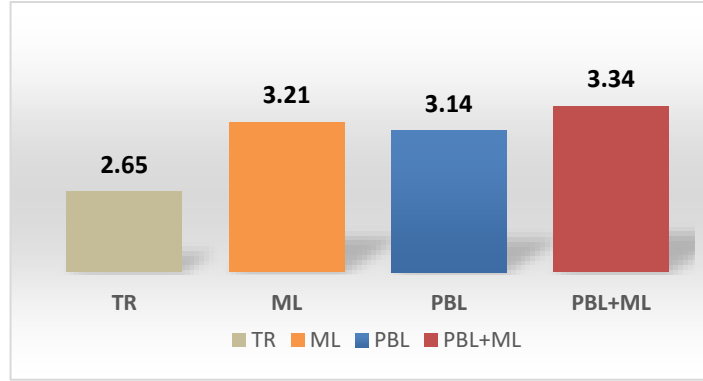
Şekil 4.20. Çekicilik faktörü açısından ortalamalar.

İlgi çekicilik faktörü analizi, ML+PBL modelinin geleneksel öğrenmeye göre %10.4, PBL'ye göre %5.6, ML'ye göre ise %2.4 daha fazla ilgi çektiğini ortaya koymuştur.

#### 4.5.1.3. Geçerlik Faktörü

Öğrencilerden alınan geri dönütler ML+PBL yönteminin geçerlik faktörü açısından (ortalama = 3.34, %66.8) da en çok tercih edilen yöntem olduğunu ortaya koymuştur. Arkasından ML grubunda olan öğrenciler de bu öğretim metodolojisine büyük bir güven göstermişlerdir ve ML yönteminin geçerlik faktörünün diğer geri kalan yöntemlerden daha yüksek değerler aldığı görülmüştür (ortalama = 3.21, %64.2).

Geleneksel öğretim yöntemi geçerlilik faktörü açısından da en düşük değerleri almıştır (Şekil 4.21).



Şekil 4.21. Geçerlilik faktörü açısından ortalamalar.

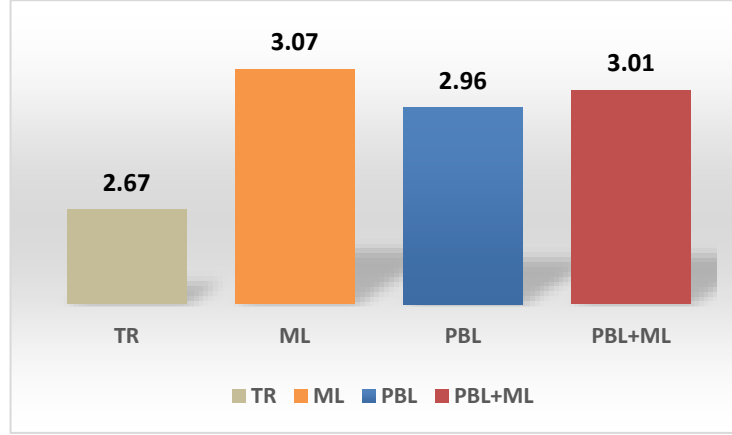
Geçerlilik faktör analizi, ML+PBL modelinin geleneksel öğrenmeye göre %13.8, proje tabanlı öğrenime göre %4 ve ML'ye göre ise %2.6 daha fazla tercih edildiğini göstermiştir. Daha önceki faktör sonuçlarına paralel olarak geleneksel öğrenmenin ise en az etkili olduğu görülmüştür.

Sonuçlar göstermektedir ki, öğrencileri derse çekme ve küçük içerikleri öğrenmelerini sağlama konusunda ML ve PBL'nin birlikte kullanımı, öğrencilerin bu teknikleri benimseme ve onları karşılaştıkları problemleri çözmede gerçek bir araç olarak görme ve kariyerlerini keşfetmelerine yardımcı olma konusunda teşvik etmektedir. Bu yaklaşım (ML+PBL), sadece, dersi öğrenciler için daha çekici kılmakla kalmayıp, aynı zamanda gömülü sistemlerin gelişimini desteklemek için gereken konuları incelemeye de yönlendirebilir.

#### 4.5.1.4. Yeterlilik Faktörü

ML ve ML+PBL yöntemleri yeterlilik faktörü açısından yine en yüksek değerleri almışlardır. Burada ML yönteminin yalnız kullanımı yeterlilik faktörü açısından (ortalama = 3.07, %61.4), öğrenciler tarafından daha fazla tercih edilmiştir. Bu durum uygulana gelen geleneksel öğretim yöntemlerinin öğrenciler üzerindeki etkisinden kaynaklanabilir. Doğal olarak geleneksel eğitim sisteminde eğitilen öğrenciler her ne kadar farklı yöntemlere ilgi duysalar da sahip oldukları alışkanlıkları bırakarak yeni yöntemlere adapte olmakta zorluk çekebilirler. ML öğrenme, geleneksel eğitim yöntemi içerisinde kısmen de olsa

kullanılan/kullanılabilen bir yöntem olması dolayısıyla öğrenciler tarafından daha çabuk kabul görebilir. Bununla birlikte ML yöntemi PBL ile kullanıldığında da yeterlik faktörü diğer yöntemlere göre daha yüksek değerler almıştır (ortalama = 3.01, %60.2). Öğrencilerin, yeni yöntemleri geleneksel yöntemlere göre daha etkili bulduğu elde edilen sonuçlardan da açıkça görülmektedir (Şekil 4.22).



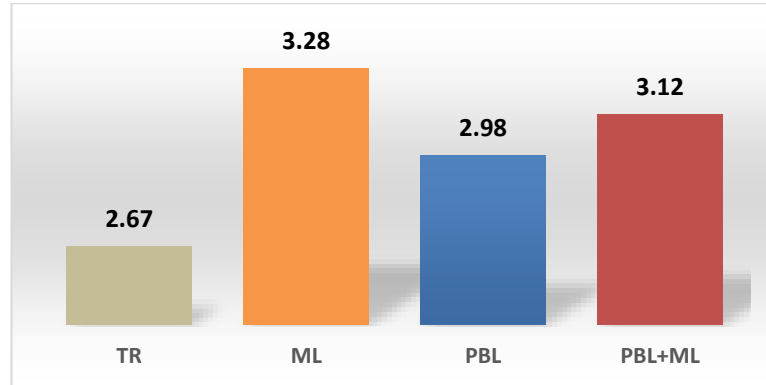
Şekil 4.22. Yeterlik faktörü açısından ortalamalar.

#### 4.5.1.5. Karmaşıklığa İzin Verme Faktörü

Bu çalışmada özellikle dikkate alınacak önemli bir değişken karmaşıklık faktörüdür, ki bu faktör, öğrencilerin öğrenimleri sırasında kolaydan karmaşığa olan öğrenme basamağında öğretim yönteminin karmaşıklığa ne kadar izin verdiğinin bir ölçüsüdür.

Karmaşıklığa izin verme faktörüne göre mikro öğrenme öğrenciler tarafında en fazla desteği görmüştür yani ML karmaşıklığa izin verme açısından en çok tercih edilen yöntem olmuştur (ortalama 3.28, %65.6 olumlu), ki yeterlik faktörü açısından da benzer bir durum ortaya çıkmıştı. Yine benzer bir durum olarak ML+PBL yönteminin ML'den sonra en çok tercih edilen yöntem olduğu görülmüştür (ortalama = 3.12, %62.4 olumlu) (Şekil 4.23). İlginç bir nokta, proje temelli bir öğretim yönteminin karmaşıklığı aşmada öğrenciler tarafından pek tercih edilmeyişidir. Bu durumun kaynağı olarak öğrencilerin geleneksel öğretim yönteminde daha çok teorik problemlerle karşı karşıya kalmaları ve pratiklerinin az olması dolayısıyla yalnızca PBL yöntemine dayalı bir öğretim sisteminden kaçınmaları gösterilebilir. Bu proje

kapsamında belirgin bilgi düzeyine sahip üst sınıf öğrencileri araştırıldığından bu öğrencilerin geleneksel sisteme alt sınıflara göre daha fazla adapte oldukları ve bu nedenle yalnızca PBL ile öğretim yöntemine de daha fazla direç gösterdikleri söylenebilir. Bununla birlikte ML ve PBL'nin birlikte uygulandığı bir yöntem teori ile pratik arasında bir köprü kurduğundan öğrencilerin bu öğretim sistemine uyumunun daha kolay olduğu öngörülebilir.

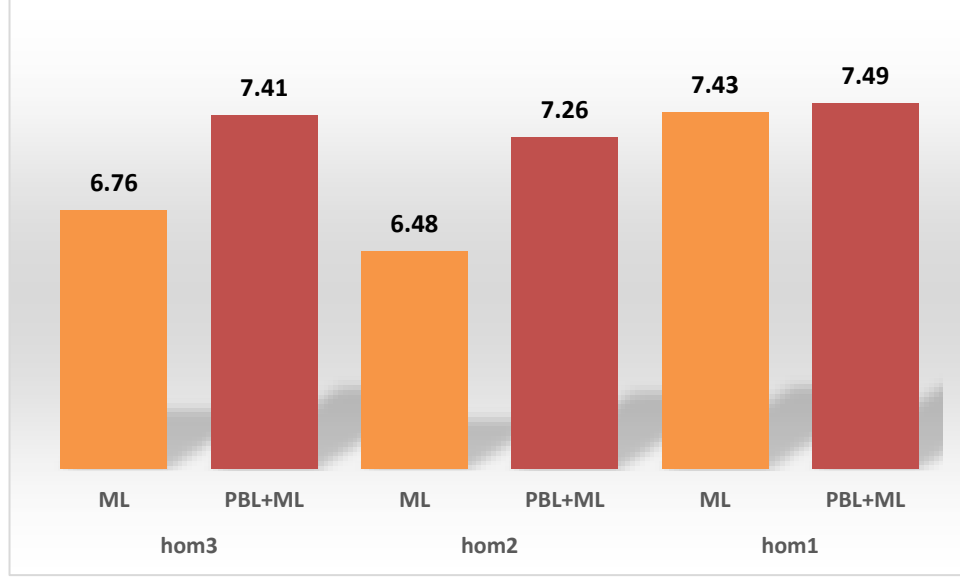


Şekil 4.23. Karmaşıklığa izin verme faktörü açısından ortalamalar.

#### 4.5.2. Ev Ödevleri Testi (V1, V2, V3)

Çalışmanın bu basamağında ML ile ML+PBL yöntemleri arasında bir kıyaslama yapabilmek amacıyla, verilen ev ödevlerine ait geri dönütler analiz edilmiştir.

İlk versiyon (V1) bir sınıf görevidir, ancak ikinci versiyon (V2) orta düzeyde karmaşıklığa sahip bir ev ödevi görevidir. V3 ise diğer ödevlere göre çok daha fazla çalışmayı gerektiren daha karmaşık bir proje ödevidir. Ödevlerin öğrenci skorları incelendiğinde (Şekil 4.24) bazı göze çarpan durumlar gözlenmiştir. Bir sınıf ödevi olan V1 en yüksek skora sahiptir, ki doğal olarak bilginin taze olması ve öğrencilerin öğretmenler tarafından sınıf ortamında anlık olarak yönlendirilebilmeleri bu skorun yüksek olmasında büyük bir etkidir. İlginç olan nokta en karmaşık proje olan V3'ün daha kolay olan V2'ye göre daha yüksek skorlar almasıdır. Burada dikkati çeken bir diğer durum daha önceki kısımda birçok faktör açısından öğrenciler tarafından daha çok tercih edilen ML yönteminin proje bazlı ödevler söz konusu olduğunda ML+PBL yöntemine kıyasla başarı açısından daha zayıf olduğudur.

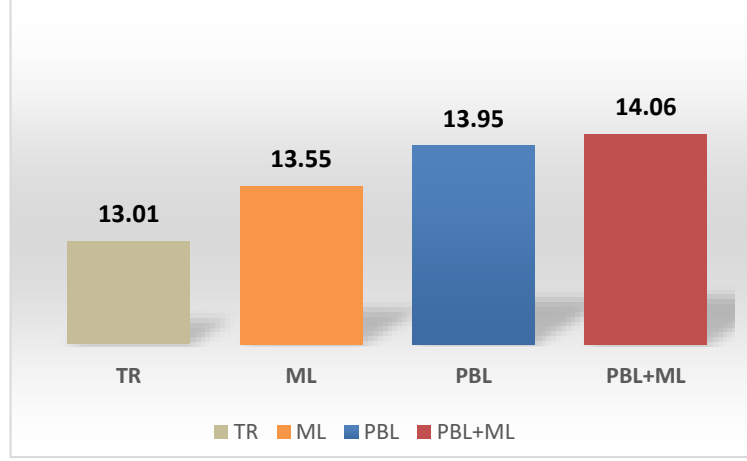


Şekil 4.24. Ev ödevleri için ML ve ML+PBL yöntemleri.

Şekil 4.24, aynı zamanda öğrencilerin öğretim yöntemlerine adaptasyon süreçlerinin bir göstergesi olarak da karşımıza çıkmaktadır. Öyle ki, birinci ödev öğretim yöntemlerinin uygulamasının ilk basamağına denk gelmektedir ve öğrencilerin derse ilgi düzeylerinin daha yoğun olduğu bir dönemdir. Küçük bir proje bazlı ödev V2 gelenekselin dışına çıkmakta ve öğrencilerin başarı ve ilgilerinin düşüşüyle sonuçlanmaktadır. Doğal olarak yöntemlere adaptasyon belli bir zaman alacaktır, ki bu zaman sonunda öğrencilerin daha zor olan uygulamalı projeleri daha rahat yaptıkları görülmektedir. Öyle ki, daha karmaşık bir proje ödevi olan V3 için geri dönütlerin V2'ye kıyasla daha yüksek olması bu görüşü desteklemektedir.

#### 4.5.3. Ara Sınav Testi

Yapılan bir arasınava, sözü edilen yöntemlerle ders görmüş öğrencilerin başarıları karşılaştırılarak yöntemlerin öğrenciler üzerindeki etkililiği incelenmiştir. Sınav sonucundan elde edilen bulgular Şekil 4.25'de verilmiştir.

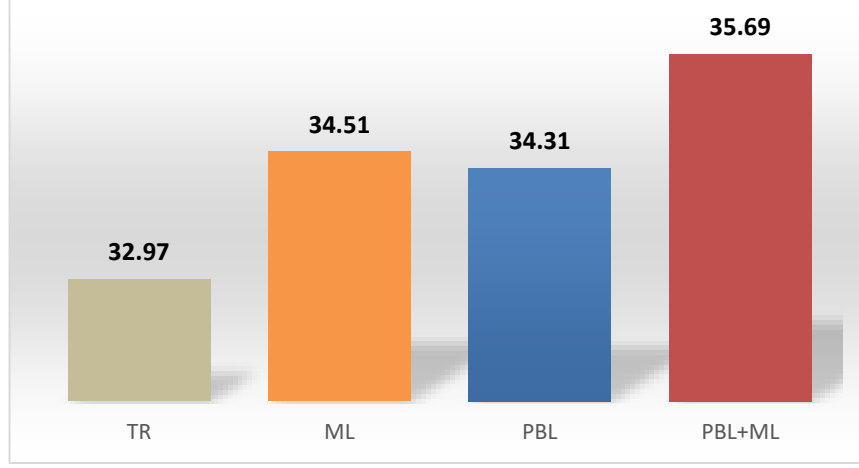


Şekil 4.25. Yöntemlerin ara sınav başarı verileri.

Ara sınav notları, ML+PBL modelini uygulayarak öğrenen öğrencilerin daha iyi bir öğrenme gerçekleştirdiğini göstermiştir (ortalama = 14.06, %70.3). Başarı sıralaması açısından, arkasından PBL (ortalama 13.95, %69.7) ve ML yöntemleri gelmektedir. Geleneksel yöntem (ortalama = 13.01, %65) ise en düşük skorlara sahiptir. ML+PBL yöntemi arasınnav sonuçlarına göre geleneksel yöntemden %5.25 daha başarılı sonuçlar vermiştir. Diğer yöntemler için bu yüzde fark daha düşüktür. Geleneksel öğrenme ve yeni yöntemler arasındaki başarı farkının pek de kayda değer olmaması açıklanmaya ihtiyaç duyulan bir durumdur. Öyle ki arasınnav, içerisinde bir uygulama bulundurmayan, teorik temelli bir sınavdır, yani öğrencinin hafızası ve ezber kabiliyeti gibi unsurlar öğrencinin başarı potansiyelinde ön plana çıkmaktadır. Geleneksel eğitime alışagelmiş öğrenciler için dönem ortalarına gelen bir teorik arasınnav, öğretim yöntemlerinde köklü değişiklikler olsa da, arasınnavın geleneksel yapısı nedeniyle başarı açısından pek bir farklılık yaratmamıştır. Öğretim yöntemleri arasındaki farklılığı, öğrencilerin yeni yöntemlere alışma süreleri ve proje tabanlı problemlerle karşılaştıklarında çözüm üretebilme kabiliyetleri belirleyecektir.

#### 4.5.4. Final Sınavı Testi

Öğrencilene uygulanan son test bir final sınavı olarak tasarlanmıştır ve öğretim yöntemlerinin etkililik kıyaslamasında önemli bir yere sahiptir.



Şekil 4.26. Final sınavı için yöntemlerin başarısı.

Final sınavının notları, PBL +ML yönteminin (ortalama = 35.69, %71.38) yine en iyi öğretim yöntemi olduğunu ortaya koymaktadır. Onu sırasıyla ML ve PBL takip etmektedir (Şekil 4.26). Öteyandan final sınavında da geleneksel yöntem ile ML+PBL arasındaki başarı farkı %5.44 gibi çok da büyük sayılamayacak bir orandır. Bununla birlikte final sınavı ve arasınava sonuçları bir noktada birbirinden çok büyük bir farklılık göstermektedir. Arasınava PBL yöntemi ML'den daha iyi sonuçlar verse de final sınavında PBL, ML ile öğretim gören öğrencilerin gerisine düşmüştür. Bunun sebebinin final sınavının teorik olması ve PBL öğrencilerinin yavaş yavaş teoriden uzaklaşıp uygulama ağırlıklı deneyimlere adapte olmaları olduğunu söyleyebiliriz. Bununla birlikte mikro öğrenmenin başarısı yapılan testler süresince yadsınmaz. ML, geleneksel yöntemi de içerisine andığından dolayı öğrencilerin daha kolay adapte olabildikleri bir sistemdir. Bu sistemin PBL ile birleşmesi hem teorik hem de pratik becerilerin daha hızlı ve verimli gelişmesi sonucunu doğurmuştur.

## 5. SONUÇLAR VE ÖNERİLER

Bu çalışma, bilgisayar mimarisini öğretmek için mikro ve proje tabanlı öğrenme yöntemlerinin kullanımının etkisine ilişkin önemli soruları ışıktutmaktadır. Dört ders dönemi boyunca 162 üst sınıf öğrencisi üzerinde gerçekleştirilen öğrenim yöntemleri, bu araştırma ile, öğrenim yöntemleri ve onların öğrenci tarafından özümsemesi, başarıya olan etkileri vs gibi bir çok açıdan incelenmiştir. ML, PBL, ML+PBL kombinasyonu ve TR arasındaki karşılaştırmanın geçerliliği için bu yöntemlerin her biri aynı bilgisayar mimarisi dersine alınan farklı öğrenci gruplarına ayrı ayrı uygulanmıştır. Ayrıca, ML ve PBL'nin bir araya getirilmesi, mikro öğrenme ve proje tabanlı öğrenmenin ayrı ayrı uygulanması gibi yeni teknik eğitim stratejilerinin FPGA teknolojisi kullanılarak etkisi test edilmiş ve bulguları karşılaştırılmıştır.

Bu çalışma kapsamında mikro öğrenme ve proje temelli öğrenme yöntemlerinin bir kombinasyonu tasarlanmış ve FPGA teknolojisi kullanılarak kolaydan karmaşık projelere kadar bir uygulama gerçekleştirilmiştir. Başarının ölçülendirilmesi amacıyla kullanılan faktörler açısından oldukça başarılı bir etki göstermiştir. Mikro öğrenme ve proje tabanlı öğrenme yöntemlerinde, bilgisayar mimarisini öğretmek için FPGA'nın kullanılması geleneksel metodolojiyle karşılaştırıldığında daha iyi sonuçlar vermiştir.

Elde edilen bulgular kısaca derlendiğinde şu sonuçlara ulaşılmıştır: Faktörlerin karşılaştırmasına ait sonuçlar en etkili/başarılı olandan en az başarılı olana olacak şekilde (En etkili *A* ve en az etkili *D* olmak üzere) Tablo 4.45'te verilmiştir. Yöntemlerin incelenen faktörler açısından başarıları göz önüne alınırsa ML ve ML+PBL yöntemlerinin diğer yöntemlere göre daha başarılı sonuçlar verdiği görülmüştür. Çekicilik ve geçerlik faktörleri açısından ML+PBL yöntemleri tercih edilirken öğretim, yeterlik ve karmaşıklığa izin verme faktörleri açısından ML tercih edilmiştir. Bu faktörler öğrencilerin alışageldikleri öğretim sistemlerine olan bağlılıklarını ve değişik değişik yöntemlere karşı bir direnç gösterdiklerini de ortaya koymaktadır.

Tablo 5.1. Öğretim yöntemlerinin faktörlere göre durumları.

<b>Faktör</b>	<b>Yöntem</b>			
	<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>
Öğretim	ML	ML+PBL	PBL	TR
Çekicilik	ML+PBL	ML	PBL	TR
Geçerlik	ML+PBL	ML	PBL	TR
Yeterlik	ML	ML+PBL	PBL	TR
Karmaşıklığa izin verme	ML	ML+PBL	PBL	TR

Bununla birlikte, öğrencilerin kendilerine verilen konuları öğrenme düzeyleri uygulanan yöntemlerin başarılarını kıyaslamada önemli bir rol oynamaktadır. Özellikle ML ve ML+PBL yöntemleri arasında görülen öğrenci tercihi, ev ödevi ve sınav başarıları ML+PBL yönteminin diğer yöntemlere kıyasla daha etkili bir yöntem olduğunu göstermektedir (Tablo 5.2 ve Tablo 5.3). Özellikle bir arasınava için PBL yönteminin ML'den daha iyi sonuçlar verdiği görülmüştür, ki bu sonuçlar, birbirine çok yakın olmakla birlikte, öğrencilerin proje temelli yaklaşımlara olan ilgisini göstermektedir. Öğrencilerin final sınavları için diğer derslere de çalışma süresi ayırma zorunluluğu nedeniyle bu ilginin dönem sonuna doğru azalma gösterdiği kuvvetle muhtemeldir.

Tablo 5.2. Öğretim yöntemlerinin ev ödevlerine göre durumları.

<b>Faktör</b>	<b>Yöntem</b>	
	<b>A</b>	<b>B</b>
Ev Ödevi 1	ML+PBL	ML
Ev Ödevi 2	ML+PBL	ML
Ev Ödevi 3	ML+PBL	ML

Tablo 5.3. Öğretim yöntemlerinin sınavlara göre durumları.

<b>Faktör</b>	<b>Yöntem</b>			
	<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>
Arasınava	ML+PBL	PBL	ML	TR
Final sınavı	ML+PBL	ML	PBL	TR

Bu çalışmanın bulguları göstermiştir ki, bilgisayar eğitiminde hedeflenen amaçlara ulaşabilmek, bilgisayar mimarisi eğitimini daha çekici ve öğretici hale getirmek, öğrencilerin karmaşık projeler tasarlama ve uygulama kapasitelerini geliştirmek, pratik projeleri uygulama konusunda güvenlerini arttırmak ve aldıkları eğitim ile

memnuniyet düzeylerini üst seviyelere çıkarmak için ML ve PBL yöntemlerinin FPGA gibi laboratuvar teknolojilerinin yardımıyla birlikte kullanılması istenilen sonuçlara daha kısa zamanda ve daha yüksek bir verimle ulaşılmasını sağlayacaktır. ML ve PBL metodolojilerinin kombinasyonu (ML+PBL), söz konusu yöntemlerin ayrı ayrı uygulandığı durumlara kıyasla önemli ölçüde iyileştirilmiş sonuçlar göstermiştir.

Sonuç olarak, mikro öğrenme ve proje tabanlı öğrenmenin birleşik bir uygulaması, öğrenim verimliliğini, incelenen faktörler açısından geleneksel yönteme kıyasla önemli ölçüde iyileştirmiştir. Bu kombinasyon diğer mühendislik konuları ve projeleri için de test edilmeli ve üzerinde daha fazla çalışma ve araştırma yapılmalıdır.

Bu çalışmanın, akademik dünyayı, daha iyi bilgi, teknik derslerde karmaşıklık yönetimi ve yüksek öğrenim performansını artıracak en etkili öğretim yöntemini bulmaya daha da yakınlara gireceğini öngörüyoruz. Uygulanan kombinasyon öğrencileri gerçek dünyadaki zorluklarla ve zorlu profesyonel ortamlarla yüzleşmeye hazırlayacaktır. Çalışmamız, mikro öğrenme ve proje tabanlı öğrenme kombinasyonunun, bilgisayar mimarisini öğretmek için mevcut tüm seçeneklerden en etkili yöntem olduğu sonucuna varmıştır

Bu çalışma, proje temelli öğrenme ile birlikte mikro öğrenme uygulamasının bilgisayar mimarisi öğrenimini artırdığına dair yeterli kanıt ve tavsiye sağlamıştır; bu nedenle, gelecek nesilleri daha iyi bir şekilde eğitmek ve farklı etkilerini bulmak için aynı araştırmanın diğer teknolojilerde ve teknik konularda da yapılmasını tavsiye ediyoruz.

Bu çalışma, üniversitelerin ve eğitim araştırmacılarının, öğrencilerin daha iyi öğrenmelerine yardımcı olacak daha kapsamlı araştırmalar yapmaları için yeterli nedenlerin olduğunu göstermektedir. Gelişimin bir bir değişmezi olarak, üniversiteler ve araştırmacılar mevcut çalışmamızda olmayan bazı yeni bulgular keşfedeceklerdir.

Bu araştırma için uygulanan modellerle ilgili farklı üniversitelerde farklı görüşler olabilir. Bu çalışma, veri toplama için farklı anketlerle ve yeni verilere dayanan

bulgularla tekrarlanmalıdır. Mevcut alıřmanın sonularına dayanarak, dięer lkelerde yrtlen daha ileri alıřmaların, mevcut eęitim sistemini geliřtirmeye teřvik edecek ve katkıda bulunacaklarını umuyoruz.

## KAYNAKLAR

- Ab Rahman, A. A.-H. & Z. M. Yusof "Altera Quartus II Tutorial for Introductory Digital Electronics."
- Allen, R. (2017). *Statistics and Experimental Design for Psychologists: A Model Comparison Approach*, World Scientific Publishing Company.
- Araujo, A. J. & J. C. Alves (2008). A project driven digital design course using FPGAs. EAEEIE Annual Conference, 2008 19th, IEEE.
- Baş, G. & Ö. Beyhab (2017). "Effects of multiple intelligences supported project-based learning on students' achievement levels and attitudes towards English lesson." *International Electronic Journal of Elementary Education* 2(3): 365-386.
- Blumenfeld, P. C., E. Soloway, R. W. Marx, J. S. Krajcik, M. Guzdial & A. Palincsar (1991). "Motivating project-based learning: Sustaining the doing, supporting the learning." *Educational psychologist* 26(3-4): 369-398.
- Bonwell, C. C. & J. A. Eison (1991). *Active Learning: Creating Excitement in the Classroom*. 1991 ASHE-ERIC Higher Education Reports, ERIC.
- Carpinelli, J. D. (2000). *Computer systems organization and Architecture*, Addison-Wesley Longman Publishing Co., Inc.
- Chandler, P. & J. Sweller (1992). "The split-attention effect as a factor in the design of instruction." *British Journal of Educational Psychology* 62(2): 233-246.
- Charteris, J., D. Smardon & E. Nelson (2017). "Innovative learning environments and new materialism: A conjunctural analysis of pedagogic spaces." *Educational Philosophy and Theory* 49(8): 808-821.
- Chen, S.-L., Y.-K. Lai, W.-C. Hu & W.-Y. Chung (2013). Case-Based Instruction of Digital Integrated Circuit Design Courses for Non-major Undergraduates. *Learning and Teaching in Computing and Engineering (LaTiCE)*, 2013, IEEE.
- Davcev, D., B. Stojkoska, S. Kalajdziski & K. Trivodaliev (2016). "Project based learning of embedded systems." arXiv preprint arXiv:1606.07498.
- Dessì, D., G. Fenu, M. Marras & D. R. Recupero (2019). "Bridging learning analytics and Cognitive Computing for Big Data classification in micro-learning video collections." *Computers in Human Behavior* 92: 468-477.
- Dillenbourg, P. (1999). *What do you mean by collaborative learning?*, Oxford: Elsevier.

- Donzellini, G. & D. Ponta (2013). From gates to FPGA: Learning digital design with Deeds. 2013 3rd Interdisciplinary Engineering Design Education Conference, IEEE.
- Doppelt, Y. (2003). "Implementation and assessment of project-based learning in a flexible environment." *International journal of technology and design education* 13(3): 255-272.
- Dumas II, J. D. (2016). *Computer architecture: fundamentals and principles of computer design*, CRC Press.
- English, M. C. & A. Kitsantas (2013). "Supporting student self-regulated learning in problem-and project-based learning." *Interdisciplinary journal of problem-based learning* 7(2): 6.
- Esteves, M. (2009). "Construction and development of the professional competences of teachers." *Educational sciences journal* 80(2): 33-44.
- Field, A. (2009). *Discovering statistics using SPSS*, Sage publications.
- Friedman, H. H. & L. W. Friedman (2018). "Does Growing the Number of Academic Departments Improve the Quality of Higher Education?" *Psychosociological Issues in Human Resource Management* 6(1): 96-114.
- Furber, S. B. (2017). *VLSI RISC architecture and organization*, Routledge.
- Furtek, F. C., M. T. Mason & R. B. Luking (2001). *FPGA structure having main, column and sector reset lines*, Google Patents.
- Gassler, G., T. Hug & C. Glahn (2004). "Integrated Micro Learning—An outline of the basic method and first results." *Interactive Computer Aided Learning* 4: 1-7.
- George, D. & P. Mallery (2016). *IBM SPSS statistics 23 step by step: A simple guide and reference*, Routledge.
- Giuliani, E. & M. Bell (2005). "The micro-determinants of meso-level learning and innovation: evidence from a Chilean wine cluster." *Research policy* 34(1): 47-68.
- Gonzalez-V, J. L. & J. E. Loya-Hernandez (2007). Project-based learning of reconfigurable high-density digital systems design: An interdisciplinary context based approach. *Frontiers In Education Conference-Global Engineering: Knowledge Without Borders, Opportunities Without Passports, 2007. FIE'07. 37th Annual*, IEEE.
- Gross, B., L. Rusin, J. Kiesewetter, J. M. Zottmann, M. R. Fischer, S. Prückner & A. Zech (2019). "Microlearning for patient safety: Crew resource management training in 15-minutes." *PloS one* 14(3): e0213178.

- Gstrein, S. & T. Hug (2006). Integrated micro learning during access delays: A new approach to second-language learning. *User-centered computer aided language learning*, IGI Global: 152-176.
- Guilbault, M. (2018). "Students as customers in higher education: The (controversial) debate needs to end." *Journal of Retailing and Consumer Services* 40: 295-298.
- Hadim, H. A. & S. K. Esche (2002). Enhancing the engineering curriculum through project-based learning. *32nd Annual Frontiers in Education, IEEE*.
- Hamblen, J. O., T. S. Hall & M. D. Furman (2006). *Rapid Prototyping of Digital Systems: Quartus® II Edition*, Springer Science & Business Media.
- Hmelo-Silver, C. E. (2004). "Problem-based learning: What and how do students learn?" *Educational psychology review* 16(3): 235-266.
- Job, M. A. & H. S. Ogalo (2012). "Micro learning as innovative process of knowledge strategy." *International journal of scientific & technology research* 1(11): 92-96.
- Joost, R. & R. Salomon (2005). Advantages of FPGA-based multiprocessor systems in industrial applications. *31st Annual Conference of IEEE Industrial Electronics Society, 2005. IECON 2005.*, IEEE.
- Kiray, V., S. Demir & M. Zhaparov (2013). Improving Digital Electronics Education with FPGA technology, PBL and Micro Learning methods. *Proceedings of 2013 IEEE International Conference on Teaching, Assessment and Learning for Engineering (TALE)*, IEEE.
- Kovachev, D., Y. Cao, R. Klamma & M. Jarke (2011). Learn-as-you-go: new ways of cloud-based micro-learning for the mobile web. *International Conference on Web-Based Learning*, Springer.
- Krajcik, J. S. & P. C. Blumenfeld (2006). Project-based learning, na.
- Lee, D. G., S. M. Jung & M. S. Lim (2007). System on chip design of embedded controller for car black box. *2007 IEEE Intelligent Vehicles Symposium*, IEEE.
- Liu, Z., L. Wei & X. Gao (2016). "A study on self-regulated micro-course learning and implicitly layered flipped classroom." *Theory and Practice in Language Studies* 6(4): 870.
- Mahmoud, W. H. (2006). Work in progress: Incorporating system-level design tools into digital design and capstone courses. *Frontiers in Education Conference, 36th Annual*, IEEE.

- Mills, J. E. & D. F. Treagust (2003). "Engineering education—Is problem-based or project-based learning the answer." *Australasian journal of engineering education* 3(2): 2-16.
- Monmasson, E. & M. N. Cirstea (2007). "FPGA design methodology for industrial control systems—A review." *IEEE transactions on industrial electronics* 54(4): 1824-1842.
- Panda, A. K., P. Rajput & B. Shukla (2012). "Design of Multi Bit LFSR PNRG and Performance comparison on FPGA using VHDL." *International Journal of Advances in Engineering & Technology* 3(1): 566.
- Quintans, C., M. D. Valdes, M. J. Moure, L. Fernandez-Ferreira and E. Mandado (2005). Digital electronics learning system based on FPGA applications. *Proceedings Frontiers in Education 35th Annual Conference, IEEE*.
- Qureshi, S. M. & C. Kang (2015). "Analysing the organizational factors of project complexity using structural equation modelling." *International Journal of Project Management* 33(1): 165-176.
- Ridgeway, D. J. (2000). Bus structure for modularized chip with FPGA modules, Google Patents.
- Rose, J., A. El Gamal & A. Sangiovanni-Vincentelli (1993). "Architecture of field-programmable gate arrays." *Proceedings of the IEEE* 81(7): 1013-1029.
- Rugarcia, A., R. M. Felder, D. R. Woods & J. E. Stice (2000). "The future of engineering education I. A vision for a new century." *Chemical Engineering Education* 34(1): 16-25.
- Ruppert, D. (2014). *Statistics and Finance: An Introduction*, Springer New York.
- S.A. Nikou , A. A. E. (2018). "Mobile-Based micro-Learning and Assessment: Impact on learning performance and motivation of high school students." *Journal of Computer Assisted Learning, John Wiley & Sons Ltd* 34(3): 10.
- Said, I. & M. Çavuş "ALU DESIGN BY VHDL USING FPGA TECHNOLOGY AND MICRO LEARNING IN ENGINEERING EDUCATION." *British Journal of Computer, Networking and Information Technology* 1(1): 18.
- Said, I. & M. Çavuş, (2019) " Utilizing VHDL for Teaching CPU Design in Engineering Education: An Approach to Build Students' Capacities to Understand and Develop Complex Projects by Applying FPGA, PBL, and ML Technologies", *International Journal of Science and Research (IJSR)* 8(3):1177-1187.
- Seinauskas, R. (1997). A distance laboratory for computer-aided design. *Proceedings of International Conference on Microelectronic Systems Education, IEEE*.

- Sokic, E. & M. Ahic-Djokic (2008). Simple computer vision system for chess playing robot manipulator as a project-based learning example. 2008 IEEE International Symposium on Signal Processing and Information Technology, IEEE.
- Sudnitson, A., D. Mihhailov & M. Kruus (2010). Advanced topics of FSM design using FPGA educational boards and web-based tools. Design & Test Symposium (EWDTS), 2010 East-West, IEEE.
- Sulaiman, N., Z. A. Obaid, M. Marhaban & M. Hamidon (2009). "FPGA-based fuzzy logic: design and applications-a review." International Journal of Engineering and Technology 1(5): 491.
- Sun, P.-C., R. J. Tsai, G. Finger, Y.-Y. Chen & D. Yeh (2008). "What drives a successful e-Learning? An empirical investigation of the critical factors influencing learner satisfaction." Computers & education 50(4): 1183-1202.
- Svab, J., T. Krajnik, J. Faigl & L. Preucil (2009). FPGA based speeded up robust features. 2009 IEEE International Conference on Technologies for Practical Robot Applications, IEEE.
- Thomas, J. W. (2000). "A review of research on project-based learning.", [http://bie.org/research/study/review\\_of\\_project\\_based\\_learning\\_2000](http://bie.org/research/study/review_of_project_based_learning_2000).
- Trimberger, S. (1993). "A reprogrammable gate array and applications." Proceedings of the IEEE 81(7): 1030-1041.
- Trost, A. & B. Zajc (2011). Logic emulators in digital systems education. 2011 IEEE EUROCON-International Conference on Computer as a Tool, IEEE.
- Tse, W. & W. Chan (2003). "Application of problem-based learning in an engineering course." International Journal of Engineering Education 19(5): 747-753.
- Vidal, L.-A. & F. Marle (2008). "Understanding project complexity: implications on project management." Kybernetes 37(8): 1094-1110.
- Wirthlin, M. J. (2005). Senior-Level Embedded Systems Design Project Using FPGAs. null, IEEE.
- Young, S. P., K. Chaudhary & T. J. Bauer (1999). FPGA repeatable interconnect structure with hierarchical interconnect lines, Google Patents.
- Zhamanov, A. & M. Zhamapor (2013). Computer Networks teaching by microlearning principles. Journal of Physics: Conference Series, IOP Publishing.

Zhen-Ting, L. & Z. Jiang-zhao (2010). "Micro-learning: A new way for adult education [J]." *Adult Education* 7: 35-37BSTS / *Biyoloji Terimleri Sözlüğü* 1998.

# **EKLER**

<b>EK 1</b>	<b>Anketler</b>
<b>EK 2</b>	<b>Birinci Seviye Proje Örneđi</b>
<b>EK 3</b>	<b>İkinci Seviye Proje Örneđi</b>
<b>EK 4</b>	<b>CPU</b>

## EK 1. Anketler

Bu anket FPGA teknolojisi ve mikro öğrenme, proje tabanlı öğrenme yöntemleri kullanarak mühendislik eğitiminin önemli yönlerini incelemeyi amaçlamaktadır. Her soruya verdiğiniz yanıt bize gerekli olan değerli bilgileri sağlayacaktır. Lütfen cevabınızı en iyi kararlarla veriniz.

Listeyi, programın müfredatına dahil edilecek bilgi, beceri ve tutumları önem derecesini değerlendirerek belirlemek için kullanın. Lütfen tüm ölçeği aşağıdaki tabloda gösterildiği gibi 1 = En az önemli olandan 5 = En önemli olanına kullanın. (Lütfen her birine √ işareti koyun).

1	En az önemli
2	Daha az önemli
3	Önemli
4	Daha önemli
5	En önemli

FPGA tekniği kullanılarak yapılan mühendislik eğitimi ve proje temelli öğrenme ve mikro öğrenme (ML+PBL) 'nin yeni öğretim yöntemiyle ilgili bu eğitim araştırması anketi, Öğretim faktörü ölçmek için, lütfen doğru fikrinizi belirleyen sayıyı belirtiniz.

Ad : .....

No: .....

NO.	Sonuç	
1	Kurs dijital elektronik gereksinimlerini karşılamak için yüksek performans elde etmeme yardımcı oldu.	1-□, 2-□, 3-□, 4-□, 5-□
2	Bu ders Dijital Elektronik'i daha verimli ve hızlı bir şekilde öğrenmeme yardımcı oldu.	1-□, 2-□, 3-□, 4-□, 5-□
3	Kurs bana proje tasarlama ve dijital elektroniği öğrenme konusunda yeterli güven verdi.	1-□, 2-□, 3-□, 4-□, 5-□
4	Kurs bana uzun vadeli bir öğrenci ve başarılı bir profesyonel olma fırsatı verdi.	1-□, 2-□, 3-□, 4-□, 5-□
5	Kurs, projeleri analiz etme ve tasarlama yeteneğimi teşvik etti.	1-□, 2-□, 3-□, 4-□, 5-□
6	Kurs, pratikte bilgiyi uygulamama yardımcı oldu.	1-□, 2-□, 3-□, 4-□, 5-□
7	Kurs, başkalarıyla etkili iletişim kurmamda ve takım çalışması yapmama yardımcı oldu.	1-□, 2-□, 3-□, 4-□, 5-□
8	Kurs mühendislik problemlerini çözmeye yeteneği kazanmamda bana yardımcı oldu.	1-□, 2-□, 3-□, 4-□, 5-□
9	Kurs disiplinlerarası takımların bir üyesi olarak çalışmamda bana yardımcı oldu.	1-□, 2-□, 3-□, 4-□, 5-□
10	Kurs öğrenmeye odaklandı ve önemli kavramları anlamama yardımcı oldu.	1-□, 2-□, 3-□, 4-□, 5-□

<i>Anketin devamı.</i>		
<b>11</b>	Kurs materyali hızlı ve ayrıntılı bir şekilde öğrenmemi sağladı.	1- <input type="checkbox"/> , 2- <input type="checkbox"/> , 3- <input type="checkbox"/> , 4- <input type="checkbox"/> , 5- <input type="checkbox"/>
<b>12</b>	Kurs etkiliydi ve yenilikçi öğrenmeyi teşvik etti.	1- <input type="checkbox"/> , 2- <input type="checkbox"/> , 3- <input type="checkbox"/> , 4- <input type="checkbox"/> , 5- <input type="checkbox"/>
<b>13</b>	Kurs, öğretici ve çekici idi.	1- <input type="checkbox"/> , 2- <input type="checkbox"/> , 3- <input type="checkbox"/> , 4- <input type="checkbox"/> , 5- <input type="checkbox"/>
<b>14</b>	Kurs olumlu yönde beklentilerimin ötesinde bir şeyler öğrendi.	1- <input type="checkbox"/> , 2- <input type="checkbox"/> , 3- <input type="checkbox"/> , 4- <input type="checkbox"/> , 5- <input type="checkbox"/>
<b>15</b>	Bu ders öğrenilen kavramlara maksimum pratik maruz kalma sunar.	1- <input type="checkbox"/> , 2- <input type="checkbox"/> , 3- <input type="checkbox"/> , 4- <input type="checkbox"/> , 5- <input type="checkbox"/>
<b>16</b>	Kurs beni çağdaş dijital tasarım araçlarına maruz bıraktı.	1- <input type="checkbox"/> , 2- <input type="checkbox"/> , 3- <input type="checkbox"/> , 4- <input type="checkbox"/> , 5- <input type="checkbox"/>
<b>17</b>	Kurs bana temel bileşenleri tasarlama ve bunları büyük projelerde kullanma yeteneği verdi.	1- <input type="checkbox"/> , 2- <input type="checkbox"/> , 3- <input type="checkbox"/> , 4- <input type="checkbox"/> , 5- <input type="checkbox"/>
<b>18</b>	Kurs, programlanabilir cihazlardaki karmaşık dijital devreleri ve sistem tasarımlarını anlamamı sağladı.	1- <input type="checkbox"/> , 2- <input type="checkbox"/> , 3- <input type="checkbox"/> , 4- <input type="checkbox"/> , 5- <input type="checkbox"/>
<b>19</b>	Kurs projesindeki deneyimimden memnunum.	1- <input type="checkbox"/> , 2- <input type="checkbox"/> , 3- <input type="checkbox"/> , 4- <input type="checkbox"/> , 5- <input type="checkbox"/>
<b>20</b>	Kursun sonunda, nispeten karmaşık projeler uygulayabiliyorum.	1- <input type="checkbox"/> , 2- <input type="checkbox"/> , 3- <input type="checkbox"/> , 4- <input type="checkbox"/> , 5- <input type="checkbox"/>
<b>21</b>	Kurs gerçek dünya projelerinin tasarımın karmaşıklığı açısından gerekliliğini karşıladığını düşünüyorum.	1- <input type="checkbox"/> , 2- <input type="checkbox"/> , 3- <input type="checkbox"/> , 4- <input type="checkbox"/> , 5- <input type="checkbox"/>
<b>22</b>	Görev karmaşıklığına rağmen, öğrendiğim konuları çoğaltmak ve birleştirmek için şansım oldu.	1- <input type="checkbox"/> , 2- <input type="checkbox"/> , 3- <input type="checkbox"/> , 4- <input type="checkbox"/> , 5- <input type="checkbox"/>
<b>23</b>	Gelişmiş ve karmaşık örneklerin geliştirilmesi, daha iyi bir öğrenme eğrisi olduğunu doğrulamıştır.	1- <input type="checkbox"/> , 2- <input type="checkbox"/> , 3- <input type="checkbox"/> , 4- <input type="checkbox"/> , 5- <input type="checkbox"/>
<b>24</b>	Bu projelerin karmaşıklığının gelecekteki karmaşık gerçek projelerle baş etmek için gereken deneyimi kazanmaya başladığımı düşünüyorum.	1- <input type="checkbox"/> , 2- <input type="checkbox"/> , 3- <input type="checkbox"/> , 4- <input type="checkbox"/> , 5- <input type="checkbox"/>
<b>25</b>	Kurs boyunca profesyonelce bir mühendis gibi düşünebildim.	1- <input type="checkbox"/> , 2- <input type="checkbox"/> , 3- <input type="checkbox"/> , 4- <input type="checkbox"/> , 5- <input type="checkbox"/>
<b>26</b>	Kurs tamamlandığında, fikirler daha da geliştirilebilir.	1- <input type="checkbox"/> , 2- <input type="checkbox"/> , 3- <input type="checkbox"/> , 4- <input type="checkbox"/> , 5- <input type="checkbox"/>

## EK 2. Birinci Seviye Proje

- 1) Tasarım 8-bit Aritmetik komutları VHDL ve simülasyonu ile tasarlayın.--  
VHDL'nin aritmetik birim tasarımı aşağıda verilmiştir:

```
LIBRARY ieee ;
USE ieee.std_logic_1164.all ;
USE ieee.std_logic_unsigned.all ;
ENTITY ArithmeticUnit IS
    PORT (
        S0:IN STD_LOGIC;
        S1:IN STD_LOGIC;
        x1 : IN STD_LOGIC_VECTOR (7 DOWNTO 0);
        x2 : IN STD_LOGIC_VECTOR (7 DOWNTO 0);
        y : OUT STD_LOGIC_VECTOR (7 DOWNTO 0));
END ArithmeticUnit;
architecture beh of ArithmeticUnit is
begin
process (x1,x2,S0,S1)
begin
If S0='0' and S1='0' Then
    y<=x1+x2;
End IF;
If S0='1' and S1='0' Then
    y<=x1-x2;
End IF;
If S0='0' and S1='1' Then
    y<=x1+1;
End IF;
If S0='1' and S1='1' Then
    y<=x1-1;
End IF;
END PROCESS ;
end beh;
```

## 2) 8 bit Mantıksal talimatları VHDL simülasyonu

--VHDL tarafından lojik birim tasarımı aşağıda verilmiştir.

```
LIBRARY ieee ;
USE ieee.std_logic_1164.all ;
USE ieee.std_logic_unsigned.all ;
use ieee.numeric_std.all;
ENTITY shiftunit IS
PORT ( S0:IN STD_LOGIC;
      S1:IN STD_LOGIC;
      x1 : IN STD_LOGIC_VECTOR (7 DOWNT0 0);
      y : OUT STD_LOGIC_VECTOR (7 DOWNT0 0));
END shiftunit;

architecture beh of shiftunit is
begin
process (x1,S0,S1)
begin
If S0='0' and S1='0' Then
    y<=x1(6 downto 0) & '0';--SHL
End IF;
If S0='1' and S1='0' Then
    y<= '0' & x1(7 downto 1);--SHR
End IF;
If S0='0' and S1='1' Then
    y<=x1(6 downto 0) & x1(7);--ROTL
End IF;
If S0='1' and S1='1' Then
    y<=x1(0) & x1(7 downto 1);--ROTR
End IF;
END PROCESS ;
end beh;
```

- **VHDL ile 8 bit kaydırma ve RI'leri tasarlayın.**

- VHDL tarafından kaydırma ve RI birim tasarımı aşağıda verilmiştir:

```
LIBRARY ieee ;
USE ieee.std_logic_1164.all ;
USE ieee.std_logic_unsigned.all ;
use ieee.numeric_std.all;
ENTITY shiftunit IS
PORT ( S0:IN STD_LOGIC;
      S1:IN STD_LOGIC;
      x1 : IN STD_LOGIC_VECTOR (7 DOWNT0 0);
      y : OUT STD_LOGIC_VECTOR (7 DOWNT0 0));
END shiftunit;
```

```

architecture beh of shiftunit is
begin
process (x1,S0,S1)
begin
If S0='0' and S1='0' Then
    y<=x1(6 downto 0) & '0';--SHL
End IF;
If S0='1' and S1='0' Then
    y<= '0' & x1(7 downto 1);--SHR
End IF;
If S0='0' and S1='1' Then
    y<=x1(6 downto 0) & x1(7);--ROTL
End IF;
If S0='1' and S1='1' Then
    y<=x1(0) & x1(7 downto 1);--ROTR
End IF;
END PROCESS ;
end beh;

```

## EK 3. İkinci Seviye Proje

### 1) Kontrol Birimi

- VHDL ile kontrol birimi tasarımı aşağıda verilmiştir:

```
LIBRARY ieee ;
USE ieee.std_logic_1164.all ;
USE ieee.std_logic_unsigned.all ;
ENTITY CU IS
    PORT (
        INSTRUCTION : IN STD_LOGIC_VECTOR (4 DOWNTO 0);
        CONTROL : OUT STD_LOGIC_VECTOR (15 DOWNTO 0));
END CU;
architecture beh of CU is
begin
process (INSTRUCTION)
begin
CASE INSTRUCTION IS
    WHEN "00000" =>
        CONTROL<="0000001000000000";--ADD
    WHEN "00001" =>
        CONTROL<="0000010000010000";--SUB
    WHEN "00010" =>
        CONTROL<="0000100000100000";--INC
    WHEN "00011" =>
        CONTROL<="0001000000110000";--DEC
    WHEN "00100" =>
        CONTROL<="0000001001000000";--AND
    WHEN "00101" =>
        CONTROL<="0000010001010000";--OR
    WHEN "00110" =>
        CONTROL<="0000100001100000";--XOR
    WHEN "00111" =>
        CONTROL<="0001000001110000";--NOT
    WHEN "01000" =>
        CONTROL<="0000001010000000";--SL
    WHEN "01001" =>
        CONTROL<="0000010010010000";--SR
    WHEN "01010" =>
        CONTROL<="0000100010100000";--RL
    WHEN "01011" =>
        CONTROL<="0001000010110000";--RR
    WHEN "10000" =>
        CONTROL<="0000000100000011";--loadX1<-M[0]
    WHEN "10001" =>
```

```
        CONTROL<="0010000100010011";--loadX1<-M[1]
WHEN "10010" =>
        CONTROL<="0100000100100011";--loadX1<-M[2]
WHEN "10011" =>
        CONTROL<="0110000100110011";--loadX1<-M[3]
WHEN "11001" =>
        CONTROL<="0000000110011100";--x2<--KEY
WHEN OTHERS =>
        CONTROL<="0000000000000000";
End CASE;
END PROCESS ;
end beh;
```

## EK 4. CPU

--CPU design by VHDL is given below:

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
LIBRARY work;
ENTITY cpuu IS
    PORT
    (
        clk : IN STD_LOGIC;
        CLK1 : IN STD_LOGIC;
        KEYIN : IN STD_LOGIC_VECTOR(7 DOWNTO 0);
        Instruction : OUT STD_LOGIC_VECTOR(4 DOWNTO 0);
        outbyte1 : OUT STD_LOGIC_VECTOR(7 DOWNTO 0);
        outbyte2 : OUT STD_LOGIC_VECTOR(7 DOWNTO 0);
        outbyte3 : OUT STD_LOGIC_VECTOR(7 DOWNTO 0);
        outbyte4 : OUT STD_LOGIC_VECTOR(7 DOWNTO 0);
        outINST : OUT STD_LOGIC_VECTOR(15 DOWNTO 0);
        outMEMORY1 : OUT STD_LOGIC_VECTOR(7 DOWNTO 0);
        X1 : OUT STD_LOGIC_VECTOR(7 DOWNTO 0);
        X2 : OUT STD_LOGIC_VECTOR(7 DOWNTO 0);
        Y : OUT STD_LOGIC_VECTOR(7 DOWNTO 0)
    );
END cpuu;
ARCHITECTURE bdf_type OF cpuu IS
    COMPONENT rom
        PORT(Address : IN STD_LOGIC_VECTOR(4 DOWNTO 0);
            Instruction : OUT STD_LOGIC_VECTOR(4 DOWNTO 0)
        );
    END COMPONENT;
    COMPONENT programcounter_v
        PORT(Clock : IN STD_LOGIC;
            Output : OUT STD_LOGIC_VECTOR(0 TO 4)
        );
    END COMPONENT;
    COMPONENT alu
        PORT(S0 : IN STD_LOGIC;
            S2 : IN STD_LOGIC;
            S1 : IN STD_LOGIC;
            S3 : IN STD_LOGIC;
            S4 : IN STD_LOGIC;
            x1 : IN STD_LOGIC_VECTOR(7 DOWNTO 0);
            x2 : IN STD_LOGIC_VECTOR(7 DOWNTO 0);
            y : OUT STD_LOGIC_VECTOR(7 DOWNTO 0)
        );
    END COMPONENT;
    COMPONENT cu
```

```

        PORT(INSTRUCTION : IN STD_LOGIC_VECTOR(4 DOWNTO 0);
              CONTROL : OUT STD_LOGIC_VECTOR(15 DOWNTO 0)
        );
END COMPONENT;
COMPONENT memxx_v
    PORT(Clk : IN STD_LOGIC;
          Clk1 : IN STD_LOGIC;
          Clk2 : IN STD_LOGIC;
          Clk3 : IN STD_LOGIC;
          Clk4 : IN STD_LOGIC;
          C : IN STD_LOGIC_VECTOR(14 DOWNTO 13);
          datain1 : IN STD_LOGIC_VECTOR(7 DOWNTO 0);
          MAINout : OUT STD_LOGIC_VECTOR(7 DOWNTO 0);
          outbyte1 : OUT STD_LOGIC_VECTOR(7 DOWNTO 0);
          outbyte2 : OUT STD_LOGIC_VECTOR(7 DOWNTO 0);
          outbyte3 : OUT STD_LOGIC_VECTOR(7 DOWNTO 0);
          outbyte4 : OUT STD_LOGIC_VECTOR(7 DOWNTO 0)
    );
END COMPONENT;
COMPONENT regx1
    PORT(clk : IN STD_LOGIC;
          LD : IN STD_LOGIC;
          w : IN STD_LOGIC;
          data : IN STD_LOGIC_VECTOR(7 DOWNTO 0);
          dout : OUT STD_LOGIC_VECTOR(7 DOWNTO 0));
END COMPONENT;
SIGNAL      C : STD_LOGIC_VECTOR(15 DOWNTO 0);
SIGNAL      outMEMORY_ALTERA_SYNTHESIZED1 : STD_LOGIC_VECTOR(7
DOWNTO 0);
SIGNAL      SYNTHESIZED_WIRE_0 : STD_LOGIC_VECTOR(0 TO 4);
SIGNAL      SYNTHESIZED_WIRE_1 : STD_LOGIC_VECTOR(7 DOWNTO 0);
SIGNAL      SYNTHESIZED_WIRE_2 : STD_LOGIC_VECTOR(7 DOWNTO 0);
SIGNAL      SYNTHESIZED_WIRE_3 : STD_LOGIC_VECTOR(4 DOWNTO 0);
SIGNAL      SYNTHESIZED_WIRE_4 : STD_LOGIC_VECTOR(7 DOWNTO 0);
BEGIN
Instruction <= SYNTHESIZED_WIRE_3;
X1 <= SYNTHESIZED_WIRE_1;
X2 <= SYNTHESIZED_WIRE_2;
Y <= SYNTHESIZED_WIRE_4;
b2v_inst1 : rom
PORT MAP(Address => SYNTHESIZED_WIRE_0,
          Instruction => SYNTHESIZED_WIRE_3);
b2v_inst2 : programcounter_v
PORT MAP(Clock => clk,
          Output => SYNTHESIZED_WIRE_0);
b2v_inst3 : alu
PORT MAP(S0 => C(4),

```

```

        S2 => C(6),
        S1 => C(5),
        S3 => C(7),
        S4 => C(8),
        x1 => SYNTHESIZED_WIRE_1,
        x2 => SYNTHESIZED_WIRE_2,
        y => SYNTHESIZED_WIRE_4);
b2v_inst4 : cu
PORT MAP(INSTRUCTION => SYNTHESIZED_WIRE_3,
        CONTROL => C);
b2v_inst6 : memxx_v
PORT MAP(Clk => CLK1,
        Clk1 => C(9),
        Clk2 => C(10),
        Clk3 => C(11),
        Clk4 => C(12),
        C => C(14 DOWNTO 13),
        datain1 => SYNTHESIZED_WIRE_4,
        MAINout => outMEMORY_ALTERA_SYNTHESIZED1,
        outbyte1 => outbyte1,
        outbyte2 => outbyte2,
        outbyte3 => outbyte3,
        outbyte4 => outbyte4);
b2v_inst7 : regx1
PORT MAP(clk => CLK1,
        LD => C(1),
        w => C(0),
        data => outMEMORY_ALTERA_SYNTHESIZED1,
        dout => SYNTHESIZED_WIRE_1);
b2v_inst8 : regx1
PORT MAP(clk => CLK1,
        LD => C(2),
        w => C(3),
        data => KEYIN,
        dout => SYNTHESIZED_WIRE_2);
outINST <= C;
outMEMORY1 <= outMEMORY_ALTERA_SYNTHESIZED1;
END bdf_type;

```

## ÖZGEÇMİŞ

Adı Soyadı : İsmail İbrahim SAID  
Doğum Yeri ve Yılı : Trablus-LİBYA / 15.10.1967  
Medeni Hali : Evli  
Yabancı Dili : Arapça, İngilizce  
E-posta : ismailsaid482@yahoo.com



### Eğitim Durumu

Lise : Alzaf Alakhder School  
Lisans : Higher Diploma in Computer Maintenance Engineering Major Field “Computer Engineering” at The Higher Institute of Civil Aviation- LİBYA  
Yüksek Lisans : Electro Technical of Sciences in Field of On-Board Computer Design at The Faculty of Electrical Engineering, Belgrade, Serbia. MsC Title: “A Processor for a Dedicated Computer”

### Mesleki Deneyim

- Aug 2010 to Sept 2013, at the general ministry for education and scientific research Suk El-Gumma High Institute, One of the teaching staff.
- Oct 2003 to july 2010, I worked in central apparatus for electronic research engineering, computer department, I worked in this department as a designer for input/output cards, also I worked as a lecturer in the University of Tripoli- the faculty of teaching.
- Sept 2001 to sept 2003. Belgrade University, masters degree.
- Dec 1991 to sept 2001. I worked in the central apparatus for electronic research engineering, at the computer network department, I worked as an engineer in the section to supervise the maintenance of PCs and the design of networks for the company.

## **Yayınlar**

1. Ismail Said and M. S. Çavuş, “Alu Design by VHDL Using FPGA Technology and Micro Learning in Engineering Education,” *British Journal of Computer, Networking and Information Technology*, Vol.1, No.1, pp.1-18, 2018.
2. Ismail Said, M. S. Çavuş, “Utilizing VHDL for Teaching CPU Design in Engineering Education: An Approach to Build Students' Capacities to Understand and Develop Complex Projects by Applying FPGA, PBL, and ML Technologies,” *International Journal of Science and Research (IJSR)*, Volume 8 Issue 3, March 2019..