T.C.
İSTANBUL ÜNİVERSİTESİ-CERRAHPAŞA
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

YÜKSEK LİSANS TEZİ

FARKLI NANOMETRE TEKNOLOJİLİLERİ İLE ALU (ARİTMETİK MANTIK BİRİMİ) TASARIMI

Merve ÖZCAN

DANIŞMAN
Prof. Dr. Fırat KAÇAR

Elektrik-Elektronik Mühendisliği Anabilim Dalı

Elektrik-Elektronik Mühendisliği Programı

İSTANBUL, 2023

TEZ KABUL VE ONAYI

Merve ÖZCAN tarafından, **Prof. Dr. Fırat KAÇAR** danışmanlığında hazırlanan " Farklı Nanometre Teknolojilileri İle Alu (Aritmetik Mantık Birimi) Tasarımı " başlıklı bu çalışma, jürimiz tarafından **31/04/2023** tarihinde yapılan sınav sonucunda **oy birliği** ile başarılı bulunarak **Yüksek Lisans Tezi** olarak kabul edilmiştir.

Tez Jürisi

		İmza	**Sonuç**
DANIŞMAN	Prof. Dr. Fırat KAÇAR İstanbul Üniversitesi Cerrahpaşa Elektrik-Elektronik Anabilim Dalı		☒ Kabul ☐ Ret
ÜYE	Doç. Dr. Yasin ÖZCELEP İstanbul Üniversitesi Cerrahpaşa Elektrik-Elektronik Anabilim Dalı		☒ Kabul ☐ Ret
ÜYE	Doç. Dr. Yunus BABACAN Erzincan Binali Yıldırım Üniversitesi Elektrik-Elektronik Anabilim Dalı		☒ Kabul ☐ Ret

BEYAN

Bu tez çalışmasının kendi çalışmam olduğunu, tezin planlanmasından yazımına kadar bütün aşamalarda etik dışı davranışımın olmadığını, bu tezdeki bütün bilgileri akademik ve bilimsel etik kuralları içinde elde ettiğimi, bu tez çalışmasıyla elde edilmeyen bütün bilgi ve yorumlara kaynak gösterdiğimi ve bu kaynakları da kaynaklar listesine aldığımı, yine bu tezin çalışılması ve yazımı sırasında patent ve telif haklarını ihlal edici bir davranışımın olmadığını ve her türlü hukuki sorumluluğu aldığımı kabul ederim.

Merve ÖZCAN

(İmza)

Aileme ithaf ediyorum…

BÜTÇE DESTEKLERİ

FARKLI NANOMETRE TEKNOLOJİLİLERİ İLE ALU (ARİTMETİK MANTIK BİRİMİ) TASARIMI

Bu tez çalışması için herhangi bir kurumdan bütçe desteği alınmamıştır.

TEŞEKKÜR

Yüksek Lisans eğitimim ve tez çalışmam sırasında hiçbir zaman desteğini esirgemeyen ve lisansüstü eğitimimde büyük katıları bulunan Sn. Prof. Dr. Fırat KAÇAR'a teşekkürlerimi sunarım.

Mart 2023 Merve ÖZCAN

İÇİNDEKİLER

Sayfa No

TEZ KABUL VE ONAYI ii
BEYAN iii
BÜTÇE DESTEKLERİ v
TEŞEKKÜR vi
İÇİNDEKİLER vii
ŞEKİL LİSTESİ ix
TABLO LİSTESİ xii
KISALTMA LİSTESİ xiii
ÖZET xv
ABSTRACT xvii
1. GİRİŞ 1
2. KAVRAMSAL ÇERÇEVE 3
2.1. VLSI TEKNOLOJİSİNİN TARİHSEL GELİŞİMİ 3
2.1.1. Moore Yasası 5
2.2. VLSI – TRANSİSTÖR ARASINDAKI BAĞLANTI 6
2.3. TRANSİSTÖR TEKNOLOJİSİNİN GELİŞİMİ 7
2.3.1. Transistörlerin Hızlı Yükselişi 7
2.3.2. Transistörlerin Küçülmesindeki Süreçler 9
2.3.3. Standart Hücreler 11
2.3.4. Transistör Teknolojisinde Türkiye 13
2.3.5. VLSI TASARIM AKIŞI 14
2.3.6. Davranışsal (Lojik) Tasarım 16
2.3.7. Fiziksel Tasarım 18
2.4. FPGA NEDİR? 20
2.4.1. FPGA Programlama Dilleri 20
2.4.2. Tasarımları Doğrulama 21
2.5. MİKROİŞLEMCİ 22
2.5.1. Kontrol Birimi 24

2.5.2. Aritmetik Mantık Birimi (ALU) 24
2.5.3. Bellek Hiyerarşisi 27
3. YÖNTEM 28
3.1. QUARTUS PROGRAMI – LOGİC TASARIM 28
3.1.1. ALU Tasarımı 30
3.2. MODELSİM – SİMÜLASYON 34
3.3. Qflow PROGRAMI – FİZİKSEL SENTEZ 35
3.3.1. QFlow Kurulum 37
3.3.2. QFlow Kullanım 43
3.4. IRSIM – SİMÜLASON 47
3.4.1. Kurulum 48
3.4.2. Kullanım 48
4. BULGULAR 54
4.1. FARKLI BİT UZUNLUKLARINDA LOJİK TASARIM 54
4.1.1. 8 Bit ALU Lojik Tasarım 54
4.1.2. 16 Bit ALU Lojik Tasarım 56
4.1.3. 32 Bit ALU Lojik Tasarım 57
4.2. FARKLI TEKNOLOJİ DOSYALARI İLE FİZİKSEL SENTEZLEME 59
4.2.1. 180nm Teknoloji ile Sentezleme 59
4.2.2. 350nm Teknoloji İle Sentezleme 61
5. TARTIŞMA 65
6. SONUÇ VE ÖNERİLER 68
KAYNAKLAR 74
İNTİHAL RAPORU İLK SAYFASI 79
ÖZGEÇMİŞ 80

ŞEKİL LİSTESİ

Sayfa No

Şekil 2.1: Moore Yasası Transistör sayısı 1970-2020 [8]. 6
Şekil 2.2: 1970-2000 yıllarındaki entegre devrelerin boyutlarındaki değişim [6]. 7
Şekil 2.3: Transistör mimarisinin değişimi [10]. 8
Şekil 2.4: Transistör mimarisindeki öngörülen dikey mimari değişimi [10]. 9
Şekil 2.5: CMOS teknolojisi boyutlarındaki değişim [11]. 9
Şekil 2.6: Teknoloji ölçeklendirme [13]. 10
Şekil 2.7: Boole lojik işlemlere örnekler [10]. 11
Şekil 2.8: Genel tasarım akışı [23]. 15
Şekil 2.9: ACIS ayrıntılı tasarım adımları. 16
Şekil 2.10: Genel kapı sentez akış diyagramı. 17
Şekil 2.11: Fiziksel tasarım akış diyagramı. 19
Şekil 2.12: İşlemci yapısı ve çevresel birimleri. 23
Şekil 2.13: Kontrol birimi şematik [26]. 24
Şekil 2.14: ALU biriminin çalışma diyagramı. 25
Şekil 3.1: Quartus II giriş ekranı. 29
Şekil 3.2: ALU tasarım mimarisi. 30
Şekil 3.3: 16 bit ALU tasarımı verilog kodu. 31
Şekil 3.4: Compilation sayfası. 32
Şekil 3.5: 16 bit ALU tasarımının RTL modeli. 33
Şekil 3.6: 16 bit ALU tasarımının testbench kodu. 34
Şekil 3.7: ModelSim simülasyon sonucu. 35
Şekil 3.8: QFlow yapısal şeması. 36

Şekil 3.9: WMware ile Ubuntu16.2.4 tanıtım. 38
Şekil 3.10: Qflow Gui terminalden açılması. 41
Şekil 3.11: Qflow kullanıcı arayüzü. 42
Şekil 3.12: Qflow ile sentezleme – 1. 44
Şekil 3.13: Qflow ile sentezleme – 2. 44
Şekil 3.14: Qflow ile sentezleme – 3. 45
Şekil 3.15: Qflow ile sentezleme – 4. 45
Şekil 3.16: Qflow ile sentezleme – 5. 46
Şekil 3.17: Qflow ile sentezleme – 6. 47
Şekil 3.18: Qflow ile fiziksel sentezleme sonucu tasarım yerleşimi. 49
Şekil 3.19: tkcon 2.3 main sayfası. 49
Şekil 3.20: .ext dosya içeriği. 50
Şekil 3.21: ALU.sim uzantılı dosya içeriği. 51
Şekil 3.22: Komut sayfasında simülasyon çıktısı. 52
Şekil 3.23: IRSIM simülasyon analizi. 52
Şekil 4.1: 8 bit ALU tasarımı verilog kodu. 54
Şekil 4.2: 8 Bit ALU Quartus II sentez çıktısı. 55
Şekil 4.3: 8 Bit ALU RTL şeması. 55
Şekil 4.4: 8 Bit ALU ModelSim ile simülasyonu. 55
Şekil 4.5: 16 bit ALU tasarımı verilog kodu. 56
Şekil 4.6: 16 bit ALU Quartus II sentez çıktısı. 56
Şekil 4.7: 16 bit ALU RTL şeması. 57
Şekil 4.8: 16 bit ALU ModelSim ile simülasyonu. 57
Şekil 4.9: 32 bit ALU tasarımı verilog kodu. 57
Şekil 4.10: 32 bit ALU Quartus II sentez çıktısı. 58
Şekil 4.11: 32 bit ALU RTL şeması kesiti. 58
Şekil 4.12: 32 bit ALU ModelSim ile simülasyonu. 58

Şekil 4.13: Qflow sentezlemesi ile oluşan Log kayıtları. 59

Şekil 4.14: 8 bit ALU tasarımı 180nm Qflow sentezlemesi. 59

Şekil 4.15: 16 bit ALU tasarımı 180nm Qflow sentezlemesi. 60

Şekil 4.16: 32 bit ALU tasarımı 180nm Qflow sentezlemesi. 60

Şekil 4.17: 8 bit ALU tasarımı 350nm Qflow sentezlemesi. 61

Şekil 4.18: 16 bit ALU tasarımı 350nm Qflow sentezlemesi. 61

Şekil 4.19: 32 bit ALU tasarımı 350nm Qflow sentezlemesi. 62

Şekil 6.1: ModelSim ile 8 bit ALU tasarımı simülasyonu – Lojik Sentez. 72

Şekil 6.2: IRSIM ile 8 bit ALU tasarımı simülasyonu – Fiziksel Sentez. 72

TABLO LİSTESİ

Sayfa No

Tablo 2.1: IC buluşlarının tarihsel akışı [5]. 4
Tablo 2.2: Entegre devrelerde mantık karışımının evrimi [5]. 5
Tablo 2.3: Teknoloji ölçeklendirme [13]. 10
Tablo 2.4: Qflow aracı transistör teknolojisi dizini [15]. 12
Tablo 2.5: Türkiye'de yarı iletken üretim tesisleri [22]. 14
Tablo 2.6: 2000'li yıllara Intel firmasın ait işlemci bilgileri [26]. 23
Tablo 2.7: Piyasadaki işlemcilerin değerlendirilmesi [34]. 26
Tablo 3.1: Quartus II sürümü özellikleri. 28
Tablo 3.2: ALU işlem listesi. 31
Tablo 3.3: Qflow kurulumu öncesi olması gereken kurulumlar. 37
Tablo 3.4: Qflow alt programları ve işlevsellikleri. 39
Tablo 3.5: Qflow checklist işlevsellikleri. 42
Tablo 4.1: 180 - 350 nm teknolojisindeki place.log verileri. 63
Tablo 4.2 : 180 - 350 nm teknolojisindeki sta.log verileri. 63
Tablo 4.3: 180 nm teknolojisindeki synth.log verileri 63
Tablo 4.4: 350 nm Teknolojisindeki synth.log verileri. 64
Tablo 6.1: Farklı bit uzunluklarında ALU tasarımlarının standart hücre sayısı. 69
Tablo 6.2: Farklı bit uzunluklarında ALU tasarımlarının kapladığı alan. 70
Tablo 6.3: 180 nm teknolojisinde pre ve post STA süreleri. 70
Tablo 6.4: 350 nm teknolojisinde pre ve post STA süreleri. 71
Tablo 6.5 : ModelSim ve IRSIM sonuçlarının değerlendirilmesi. 72

KISALTMA LİSTESİ

Kısaltmalar	Açıklama
ALU	: Aritmetik Mantık Birimi (Aritmetic Lojik Unit)
ASICs	: Uygulamaya Özel Tümleşik Devreler (Application Specific Integrated Circuits)
CAD	: Bilgisayar Destekli Tasarım (Computer Aided Design)
CMOS	: Bütünleyici Metal Oksit Yarı İletken (Complementary Metal Oxide Semiconductor)
CPU	: Merkez İşlem Birimi (Central Processing Unit)
FPGA	: Alanda Programlanabilir Kapı Dizisi (Field Programmable Gate Array)
GDS	: Grafik Tasarım Sistemi (Graphic Design System)
GPL	: Genel Kamu Lisansı (General Public License)
HDL	: Donanım Tasarlama Dili (Hardware Description Language)
IC	: Entegre Devrelerin (Integration Circuit)
IEEE	: Elektrik ve Elektronik Mühendisleri Enstitüsü (Institute of Electrical and Electronics Engineers)
ILA	: Entegre Mantık Analizörü (Integration Lojik Analizör)
IRDS	: Uluslararası Cihazlar ve sistemler için Yol Haritası (International Roadmap for Devices and Systems)
ITRS	: Yarı İletkenler için Uluslararası Teknoloji Yol Haritası (International Technology Roadmap for Semiconductors)
LSI	: Geniş Seviye Entegreler (Level Stage Integrated)
MOSFET	: Metal Oksit Yarı İletkenli Alan Etki Transistörü (Metal Oxide Semi-conductor Field Effect Transistör)
MSI	: Orta seviye Entegreler (Intermediate Integrateds)
NMOS	: N Kanal Mosfer
RAM	: Rastgele Erişimli Bellek (Random Access Memory
RISC	: Azaltılmış Komut Setli Bilgisayar ()
ROM	: Sadece Okunur Bellek (Read-Only Memory)
RTL	: Tasarım Kapı Düzeyinde (Register Transfer Level)
SRAM	: Durağan Rastgele Erişim Bellek (Statik Randam Access Memory)
TÜBİTAK	: Türkiye Bilimsel ve Teknik Araştırma Kurumu

ULSI : Ultra Geniş Seviyeli Entegreler (Ultra Wide-Level Integrateds)

VHDL : Çok Yüksek hızlı entegre devre Açıklama Dil (Very High Speed Integrated Circuit Description Language)

VLSI : Çok Geniş Ölçekli Tümleşim (Very large-scale Integration)

YİTAL : Yarı İletken Teknolojileri Araştırma Laboratuvarı

ÖZET

YÜKSEK LİSANS TEZİ

FARKLI NANOMETRE TEKNOLOJİLİLERİ İLE ALU (ARİTMETİK MANTIK BİRİMİ) TASARIMI

Merve ÖZCAN

İstanbul Üniversitesi-Cerrahpaşa

Lisansüstü Eğitim Enstitüsü

Elektrik-Elektronik Mühendisliği Anabilim Dalı

Elektrik-Elektronik Mühendisliği Programı

Danışman : Prof. Dr. Fırat KAÇAR

Günümüzde VLSI teknolojisine gösterilen önem artmaktadır. Öyle ki, bu alanda gerçekleştirilen çalışmalar ülkelerin stratejik konular arasında yer almaktadır. Tez kapsamında VLSI tasarım süreçlerinin aşamaları olan dijital tasarım ve fiziksel tasarım aşamalarına yer verilmiştir. Bu iki aşama sonrası simülasyonlar gerçekleştirilerek tasarımların doğruluğu kontrol edilmiştir. Bu alanda yetişmiş iş gücümün de yaygınlaştırılması için açık kaynak kodlu yada ücretsiz sürümleri kullanılarak çalışma gerçekleştirilmiştir. Dijital Tasarım için Quartus II aracı ve simülasyon için ModelSim, Fiziksel Tasarım için qflow ve simülasyon için Irsim aracı kullanılmıştır. VLSI teknolojisi ile transistör teknolojisinin gelişimleri birbirini desteklemektedir. Bu çalışmada 180 nm ve 350 nm teknoloji dosyaları ile farklı bit uzunluklarına sahip ALU tasarımları sentezlenmiştir. Aynı bit uzunluğundaki tasarımda transistör teknolojisin küçülmesi tasarımı oluşturan standart hücre sayısında belirgin bir değişikliğe sebep olmamaktadır. Standart hücre sayısı bir uzunluğu ile doğru orantılı bir şekilde artmaktadır. Teknolojinin küçülmesi ile tasarımın kapladığı alan küçülmektedir. Aynı zamanda teknolojisinin küçülmesi ile Statik Time Analizi (STA) azalmaktadır. Bu çalışmada aynı

zamanda dijital tasarım ve fiziksel tasarım amalarında simülasyon çalışmaları gerçekleştirilmiştir. Farklı teknoloji dosyalarında gerçekleştirilen sentezleme sonuçları simülasyonlar ile de doğrulanmıştır.

Mart 2023 , 80 sayfa.

Anahtar kelimeler: VLSI, Dijital Tasarım, Fiziksel Tasarım, Açık Kaynak Kod, Quartus II, Qflow, Modelsim, Irsim

ABSTRACT

M.Sc. THESIS

DİFFERENT NANOMETER TECHNOLOGİES ON ALU (ARİTMETİC LOGİC UNİT) DESİGN

Merve ÖZCAN

İstanbul University-Cerrahpaşa

Institute of Graduate Studies

Department of Electrical and Electronics Engineering

Electrical and Electronics Engineering Program

Supervisor : Prof. Dr. Fırat KAÇAR

Today, the importance given to VLSI technology is increasing. So much so that the studies carried out in this field are among the strategic issues of the countries. Within the scope of the thesis, digital design and physical design stages, which are the stages of VLSI design processes, are included.After these two stages, simulations were carried out and the accuracy of the designs was checked. In order to expand the workforce trained in this field, the study was carried out using open source code or free versions. Quartus II tool for Digital Design and ModelSim for simulation, qflow for Physical Design and Irsim tool for simulation. Developments in VLSI technology and Transistör technology support each other. In this study, 180 nm and 350 nm technology files and ALU designs with different bit lengths were synthesized.The shrinkage of the Transistör technology in the same bit-length design does not cause a significant change in the number of standard cells that make up the design.The number of standard cells increases in direct proportion to a length.With the shrinking of technology, the area covered by the design is getting smaller.At the same time, with the shrinking of technology, Static Time Analysis (STA) is decreasing. In this study, simulation studies were also carried out in the digital design

and physical design stages.Synthesis results performed in different technology files were also verified by simulations.

March 2023, 80 pages.

Keywords: VLSI, Digital Design, Physical Design, Open Source Code, Quartus II, Qflow, Modelsim, Irsim

1. GİRİŞ

Günümüzde yüksek teknolojili ürünlerin vazgeçilmezi noktasına gelen Transistör teknolojisinde ki gelişmelerin başlangıç noktasında 2 adet Transistör olduğu bilinmektedir. Bu sayı günümüzde ortalama olarak milyonlar ile ifade edilmektedir. Transistör teknolojisinde gelişmeler ile ışığında entegre devre kullanımları yaygınlaşmıştır. Alanda programlanabilir array dizilerinin (FPGA) kullanımı birden fazla kez programlanabildikleri ve düşük güç tüketimleri ile tercih sebebi ürünler olmaktadırlar.

Yaygınlaşan VLSI teknolojisi avantajlı olduğu kadar pahalı ve uzun süreli üretim süreçlerine sahip bir teknolojidir. Temelde tasarım aşaması iki bölümden oluşmaktadır; davranışsal tasarım ve fiziksel tasarım. VLSI teknolojisi ile ilgilenen firmaların bazıları bu sürecin sadece davranışsal tasarımını bazısı ise sadece fiziksel tasarımı yapmakta iken sadece tasarımların üretimini gerçekleştiren firmalarda bulunmaktadır.

VLSI tasarımlarının lisanslı Bilgisayar Destekli Tasarım (Computer Aided Design) CAD programları ile profesyonel olarak yapılabildiği gibi açık kaynak kodlu yazılımlar ile de yapılabilmektedir. Lisanlı kullanılan araçlar; Cadence, Synopsys, Mentor [1]. Lisansa ihtiyaç duymayan araçlar; qFlow, OpenLANE, Coriolis, VTR, SymbiFlow [2].

Mikroişlemci kullanımları hem endüstriyel uygulamalarda hem de askeri alanda kullanımı yaygındır. Mikroişlemcilerin temel yapı taşları Aritmetik Lojik Unitler (ALU)'dir. ALU'ların kapasiteleri de mikroişlemcilerin büyüklüğü ile artmıştır.

Tez çalışması kapsamında temel işlevsellikte bir ALU'nun dijital entegre devre tasarımı sürecinde yer alan davranışsal tasarım ve fiziksel tasarım aşamaları gerçekleştirilmiştir. Davranışsal tasarım, Quartus II ile Verilog dilinde yapılarak ModelSim aracı ile simülasyonu gerçekleştirilmiştir. Fiziksel sentezleme sürecinde açık kaynak kodlu qflow aracı kullanılmış ve simülasyon aracı olarak IRSIM aracı kullanılmıştır.

8 bit, 16 bit ve 32 bit ALU tasarımlarının davranışsal tasarımları 180nm ve 350nm Transistör teknolojilerinde fiziksel sentezlemesi yapılarak standart hücre sayısı, kapladıkları alan ve sabit süre analizi (STA) değerleri değerlendirilmiştir. Literatüre ek olarak hem davranışsal

modelleme sonrasında ModelSim ile tasarımın simülasyonu yapılmış hem de fiziksel sentezleme sonrasında IRSIM aracı kullanılarak fiziksel sentezlemenin doğruluğu kontrol edilmiştir. Bu iki aşamadaki simülasyonları da hem farklı bit uzunluklarına hem de farklı transistör teknolojilerdeki sentezlemeler ile karşılaştırılmıştır. Bu simülasyonların karşılaştırılması ile transistör teknolojisindeki küçülmenin tasarımın işlevselliğine bozucu bir etki oluşturmadığı gözlemlenmiştir.

Tez çalışmasının sonuçlarında, tasarımdaki bit uzunluğu arttığında kullanılan hücre sayısı bit uzunluğu ile doğru orantılı olarak artmıştır. Sentezlemede kullanılan transistör teknolojisindeki küçülmenin tasarımın hücre çeşitliliğini etkilememektedir. Transistör teknolojisinde ki ilerleme yani Transistörlarin küçülmesi ile birim alana sığdırılan Transistör sayısı artmaktadır. Bu değişimim daha önce yapılan çalışları ile uyuşmaktadır.

Transistör teknolojisindeki küçülme ile tasarımın kapladığı alan bit uzunluğu artsa dahi azalmaktadır. Tasarımın kapladığı alanın azalması sonucunda tasarımını yerleştirirken bir hatanın oluşup oluşmadığı da fiziksel tasarım aşamasından sonra yapılan IRSIM simülasyonu ile doğrulanmıştır [3 – 4].

Farklı bit uzunluğundaki tasarımları ve iki farklı transistör teknolojisi ile sentezleme yaparak tasarımda ki gecikme süresi üzerindeki etkileri gözlenmiştir. Transistör teknolojisindeki küçülmenin bit uzunluğu artsa dahi gecikme süresini azaltmaktadır. Ayrıca açık kaynak kodlu bir yazılımda fiziksel sentezleme sonrasında kapasitif ve rezistif değerleri dahil ederek oluşturduğu gecikme süresi üzerindeki değişim de gözlemlenmiştir. Elde edilen değerler literatürde yer alan çalışmaları desteklemektedir.

Tez çalışması, ücretsiz sürümü olan bir araç ile davranışsal tasarımı ve açık kaynak kodlu olan Qflow aracı ile VLSI tasarım süreçlerinin yaşam döngüsünü kapsamaktadır. Bu yönüyle VLSI tasarım ve sentezleme alanında çalışmak isteyenler için destekleyici bir içerik oluşturmaktadır. Ayrıca, farklı bit uzunluklarında ve iki farklı transistör teknolojisinde dijital tasarımı ve fiziksel tasarımı yapılmış ve sentezlemeler sonrasında gerçekleştirilen simülasyon çalışmaları ile de süreçlerin doğruluğu teyit edilmiştir.

VLSI alanında açık kaynak kodlu yazılımların geliştirilmesi ve bu araçlar ile yapılacak çalışmaların lisanslı yazılımlar ile sonuçlarının karşılaştırılarak açık kaynak kodlu araçların zayıf yönleri geliştirilmelidir.

2. KAVRAMSAL ÇERÇEVE

2.1. VLSI TEKNOLOJİSİNİN TARİHSEL GELİŞİMİ

Çok Geniş Ölçekli Tümleşim (VLSI) teknolojisinin gelişiminin temelleri 1904 – 1947 yıllarındaki gelişmeler ile başlamıştır. Bu yıllar arasında elektron lambalarının kullanımı önem kazanmıştır. J.A. Fleming tarafından ilk diyot lamba 1904 yılında yapılmıştır. Lee De Forest, 1906 yılında üçüncü elektrotu diyot lambaya ilave ederek triyot lambayı geliştirmiştir.

Entegre Devrelerin (IC) tarihsel gelişimi 1949'lu yıllara dayanmaktadır. Alman mühendis Werner Jacobi tarafından yapılan 2 aşamalı amplifikatör düzenlemesinde ortak bir alt tabaka üzerinde beş Transistörü bulunduran IC benzeri bir yarı iletken yükseltme cihazı yapmıştır. Jacobi, bu yükseltme cihazı için patent başvurusu da yapmıştır. Jacobi yaptığı tasarım ile işitme cihazlarının küçük ve ucuz olabileceğini göstermiştir.

Vakum tüplerinin işlevini yarı iletkenlerin gelebileceğini gösteren deneysel keşifler ve 20. yüzyılın ortalarında yarı iletken cihaz imalatındaki teknolojik gelişmeler sayesinde entegre devrelerin kullanımıyla mümkün olmuştur. Elektronik bileşenlerin manuel olarak montajlanması yerine, çok sayıda Transistörun küçük bir çipe entegrasyonu önemli bir kilometre taşı olarak yer almaktadır [5-6-7].

IC'ler Transistörler, dirençler ve kapasitelerden oluşan küçük bir yarı iletken tabanlı elektronik cihazlardır. IC'ler gelişen teknoloji ile elektronik cihaz ve ekipmanın yapı taşı olarak görülmektedir. IC'ler çip ve mikroçip olarak da bilinmektedir.

Tablo 2.1: IC buluşlarının tarihsel akışı [5].

Mucit/Bulan Kişi	Yıl	Tasarım	Açıklama
Fleming	1904 1906	Vakum tüplü diyot Vakum triyot	Büyük pahalı, güç aç, güvenilmez
William Shockley (Bell labs)	1945	Yarı iletken değiştirme vakum tüpü	-
Bardeen and Brattain and Shockley (Bell labs)	1947	Nokta İletişim transferi direnç cihazı "BJT"	Büyümenin itici faktörü VLSI teknolojisi
Werner Jacobi (Siemens AG)	1949	IC içeren yükseltici Cihaz iki aşamalı amplifikatör	Ticari kullanım bildirilmedi
Shockley	1951	Kavşak (Junction) Transistörü	"Pratik formu Transistör"
Jack Kilby (Texas Instruments)	Temmuz 1958	Entegre Devreler F/F 2-T Germanyum dilim ve altın teller	IC tasarımının babası
Noyce Fairchild Semiconductor	1958	Entegre Devreler Silikon	"Silikon Vadisi Başkanı"
Kahng Bell Lab	1960	İlk MOSFET	Yarı iletken endüstrisi için yeni dönemin başlangıcı
Fairchild Semiconductor And Texas	1961	İlk Ticari IC	-
Frank Wanlass (Fairchild Semiconductor)	1963	CMOS	-
Federico Faggin (Fairchild Semiconductor)	1968	Silikon kapı IC teknolojisi	Daha sonra 1970'te ilk CPU Intel 4004'ü 9 mm'de 2300 T'ye yönlendirmek için Intel'e katıldı
Zarlink Semiconductors	Günümüz	Endoskopi için M2A kapsül	2 sn sindirim sisteminin fotoğrafı çekilmiş

Günümüzde veri çeşitliliğinin artması ile birlikte daha karmaşık işlevler yapabilecek IC'lere ihtiyaç artmıştır. Bir çipteki mantık kapılarının sayısı ile ölçülen entegrasyon düzeyi yaklaşık otuz yıldır istikrarlı bir şekilde yükselmektedir.

Tablo 2.2'de tümleşik devrelerdeki mantık karmaşıklığının evriminin son otuz yılki gelişimi görülmektedir [5].

Tablo 2.2: Entegre devrelerde mantık karışımının evrimi [5].

Dönem	**Tarih**	**Karmaşıklık Çip İçerisindeki Lojik Bloklar**
Tek Transistör	1958	<1
Birim Mantığı (one gate)	1960	1
Çok İşlevli (multi-function)	1962	2-4
Karmaşık İşlevli	1964	5-20
Orta Seviye Entegreler (MSI)	1967	20-200
Geniş Seviye Entegreler (LSI)	1972	200-2.000
Çok Geniş Seviyeli Entegreler (VLSI)	1978	2.000-20.000
Ultra Geniş Seviyeli Entegreler (ULSI)	1989	20.000-?

Entegre Devrelerin avantajları;

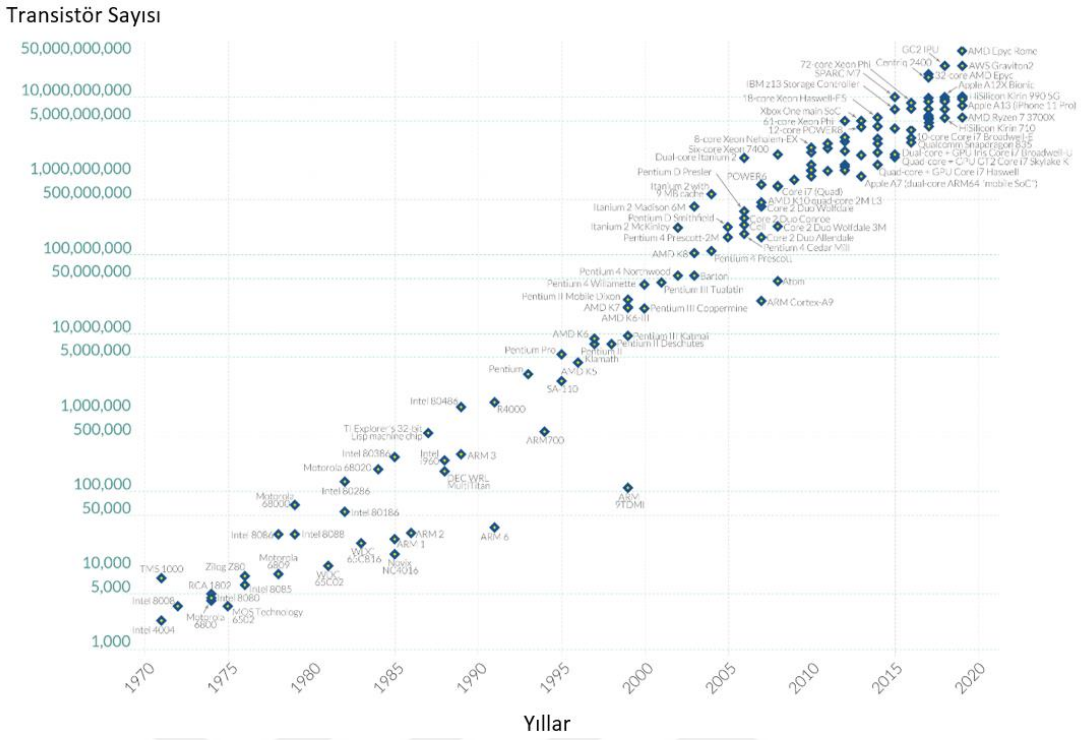
- Alandan ve hacimden tasarruf sağlar
- Daha az güç tüketimi
- Sistem bütününde daha az test gereksinimi
- Çip üzerindeki ara bağlantılar sayesinde daha yüksek güvenlik
- Azaltılmış ara bağlantılar sayesinde daha yüksek hız
- Maliyet tasarrufu

Teknolojik gelişmeler ile birlikte, Transistör teknolojisinin gelişmesi desteklenmiştir. Bu gelişmeler Transistörün kanal uzunluğu ya da çip üzerinde yerleşimi sonrasındaki ara bağlantılarının genişliği sürekli olarak azaltılmasına imkan sağlamaktadır [5]. Bu azalmanın en önemli dezavantajı ise entegreye ait sıcaklık artışı olmuştur.

2.1.1. Moore Yasası

Inten firmasının kurucusu olan Gordan Moore tarafından ortaya atılan ve adı ile anılan yasadır. Bu yaklaşım "*Her 18 ayda, bir tümleşik devre üzerine yerleştirilebilecek bileşen sayısı iki katına çıkarken, üretim maliyetleri aynı kalır, hatta düşme eğilimi gösterir*." ifade etmektedir [8,9].

Dünyada bazı istisnalar haricinde Moore Yasasının takip edilmektedir. Şekil 2.1 günümüzü göstermektedir.



Şekil 2.1: Moore Yasası Transistör sayısı 1970-2020 [8].

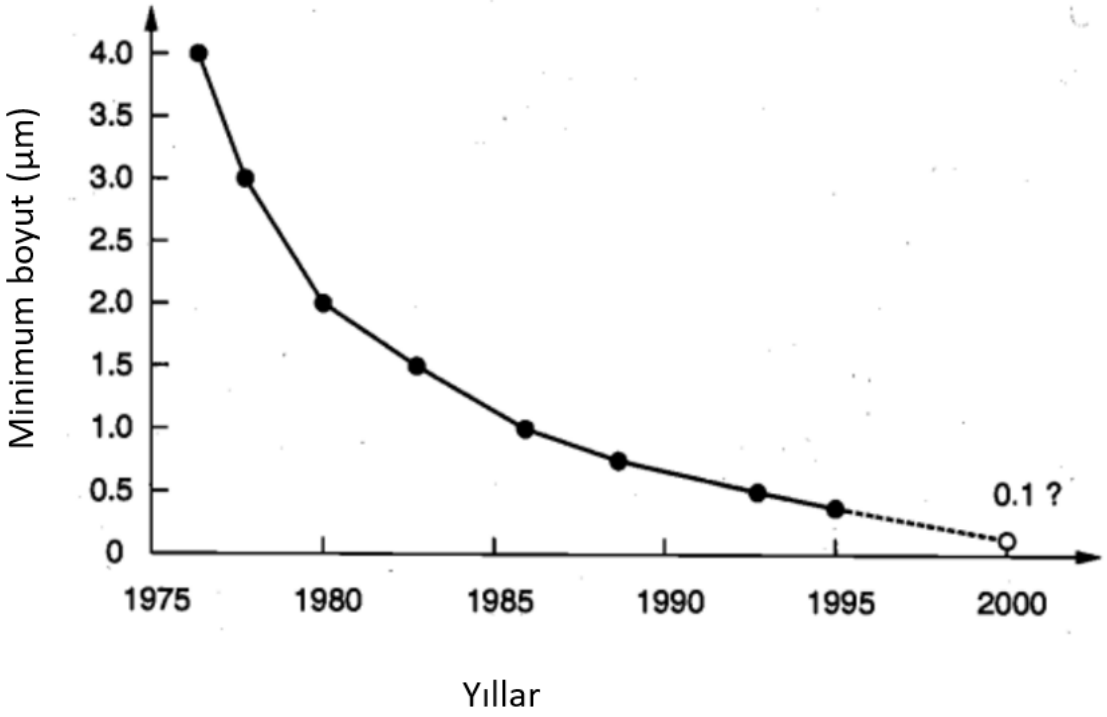
2.2. VLSI – TRANSİSTÖR ARASINDAKI BAĞLANTI

Elektronik endüstrinde en yaygın kullanılan Transistörler mosfetlerdir. Mosfetler, VLSI yongalarındaki temel bileşendir. 1980'lerde talebinin artması ile NMOS tümleşik devrelerde güç tüketimi sorunları ortaya çıkmıştır. Güç tüketim sorunun çözüm olarak CMOS teknolojisi geliştirilmiştir.

CMOS'lar (Complementary MOS) yüksek giriş empedansı, gürültülere karşı dayanıklı ve çift yönlü anahtarlama özelliğine sahip anahtarlama elemanlarıdır. CMOS'lar n-kanallı ve p-kanallı mosfetlerin birleşimiyle oluşmaktadır. Düşük güç tüketimi ve kolay üretim sürecine sahip olması öne çıkan özellikleridir [5].

VLSI teknolojisinin hızlı büyümesini destekleyen çalışmalardan biri de VLSI tasarım süreçlerini destekleyen çalışmalardır. Teknolojideki hızlı gelişmeler sayesinde entegrelerin boyutları küçülürken birim alanda ki yoğunluk artmıştır [5,6].

Şekil 2.2'de 1970'lerden 2000'li yıllara kadarki entegre devrelerdeki Transistörlerin boyutundaki değişimi göstermektedir.



Şekil 2.2: 1970-2000 yıllarındaki entegre devrelerin boyutlarındaki değişim [6].

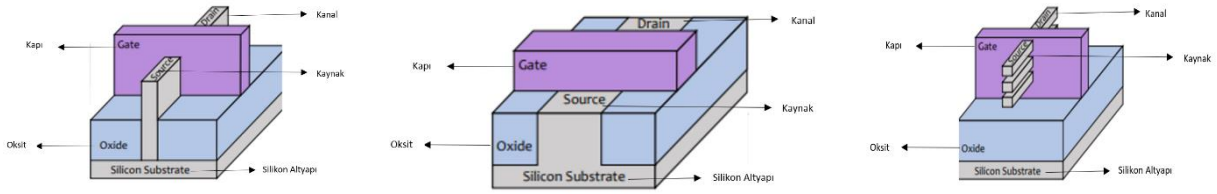
2.3. TRANSİSTÖR TEKNOLOJİSİNİN GELİŞİMİ

2.3.1. Transistörlerin Hızlı Yükselişi

Yarı iletken teknolojisindeki gelişmeler sürekli bir önceki sürecin küçültülmesini sağlamıştır. Bu gelişmeler sayesinde, Transistörleri ölçeğindeki küçülmeler IC'lerdeki daha fazla Transistör yoğunluğu elde edilmesini olanak vermiştir. Bir IC üzerindeki Transistör sayısı her iki yılda bir iki katına çıkmasının mümkün olduğunu belirten öngörü Moore Yasası'dır. Bu öngörü ile Transistör teknolojisinde kullanılan malzemeleri ve fiziksel boyutlarını sınırlarını zorlamaktadır [10].

MOSFET (Metal-okside-semiconductor field-effect Transistör), iletken ve yarı iletken malzemelerin sandviç modelini andıran yapısı ile ticari olarak kullanılan en başarılı yarı iletken cihazlardır. Transistörün çalışması elektrik alanları ile kontrol edildiğinden bu cihazlara alan etkili Transistör (FET) denir. Mosfetleri kullanarak lojik fonksiyonlar oluşturmak için CMOS'lar (Complementary MOS'lar) kullanılmıştır. MOSFET'lerin ilk geliştirme aşamasında yerleşimi düzlemsel yapılmıştır. 1950'lerde ilk MOSFET başarıyla üretilmiştir [10].

2000'lı yılların başında, Dennard Boyutlandırma methodolojisi çökmüştür. Dannard methodolojisi, Transistörlerin fiziksel boyutları azaltıkça güç yoğunluğunun sabit kaldığı bilgisi bozulmuştur. Boyutlar küçüldükçe sızıntı akımları problemi ve ısı problemleri ortaya çıkmıştır. Bu sorunlara çözüm olarak Transistör mimarisinde yenileme yapılmıştır. Kaçakları azaltmak için kanal kontrolü artırılması ve düşük güçlerde çalışma için düzlemsel modelden vazgeçilmiştir. Transistör mimarisinin değişimi Şekil 2.3'de gösterilmektedir [10].

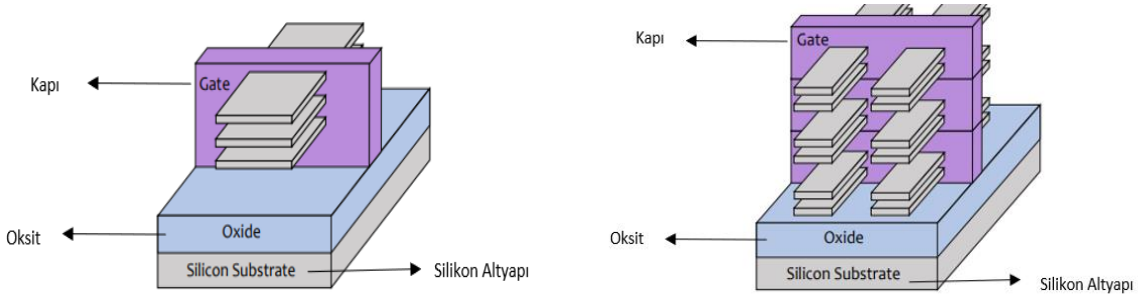


(a) Düzlemsel FET (b) FinFET (c) GAAFET

Şekil 2.3: Transistör mimarisinin değişimi [10].

2001 yılında yayınlanan Yarı İletkenler için Uluslararası Teknoloji Yol Haritası (ITRS) raporuna göre, CMOS teknolojisindeki ölçeklendirme sınırlamasının ortadan kaldırılması için FinFET teknolojisini tavsiye etmiştir. FinFET teknolojisi dökümhanelerde 2010'da beri kullanılmaktadır. 14nm, 10nm ve 7nm teknolojileri FinFET teknolojisi içermektedir. Bununla birlikte, Elektrik ve Elektronik Mühendisleri Enstitüsü (IEEE) Uluslararası Cihazlar ve sistemler için Yol Haritası (IRDS) 2020'deki raporunda FinFET teknolojisinin 2025'e kadar devam edebileceğinden bahsetmiştir [11,12].

2020 de yayınlanan IEEE IRDS'ye göre bir sonraki transistör kalıbındaki değişiklik GAAFETler ile 2025 sonrasında olacağını raporlamıştır. Önümüzdeki 15 yıllık süreçte, mimarideki öngörülen değişiklik dikey olarak yaşanması beklenmektedir. Öngörülen dikey mimari değişimi Şekil 2.4'de gösterilmiştir [11,12].



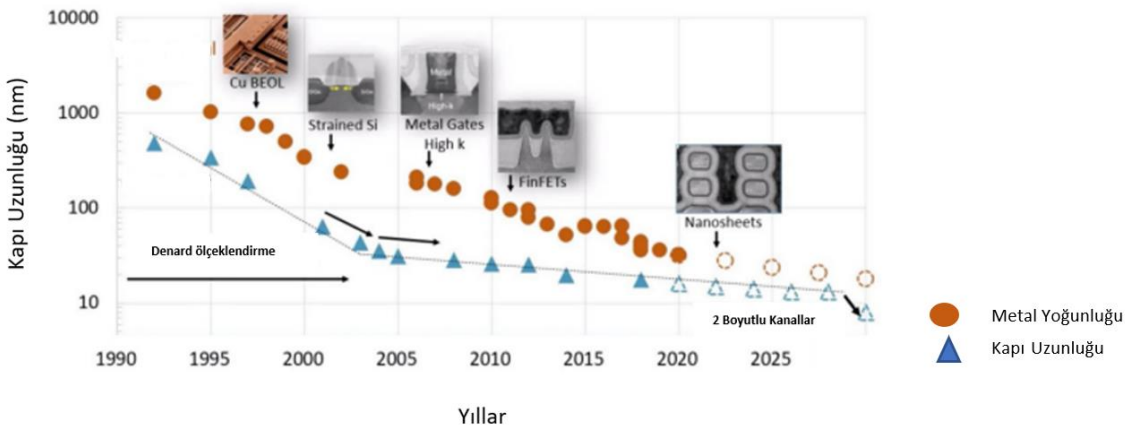
Şekil 2.4: Transistör mimarisindeki öngörülen dikey mimari değişimi [10].

2.3.2. Transistörlerin Küçülmesindeki Süreçler

Transistörlerin üretim teknolojisi, Transistörün boyutunu belirten bir düğüm adıyla isimlendirilir. Ancak son zamanlarda, özellikle FinFET teknolojisi kullanımıyla düğüm aralığı ile isimlendirme anlamını yitirmeye başlamıştır. Düğüm Adı; geçit uzunluğu veya metal yarım adım gibi Transistörün fiziksel özelliklerini temsil etmek için kullanılmaktadır [10].

Node Range (Düğüm Aralığı) isimlendirmesi daha önceden belirli bir boyuttaki transistör boyutunu ve teknolojisini belirtmekteydi. Fakat artık FinFET teknolojisi kullanımı ile herhangi bir Transistörün geometrisini temsil etmemektedir [10].

2020'de yayınlanan IEEE IRDS kapsamında 2022'den 2034'e kadar öngörülen düğüm isimleri "3", "2.1", "1.5", "1.0 eq" and "0.7 eq"dir. Sayılar sürekli olarak küçülüyor görünmesine rağmen fiziksel kapı uzunluğu sürekli azalmaz. 5nm düğümünde beklenen CMOS kapı uzunluğu 18nm ile başlar 1.5nm düğümünde bu değer yanlızca 12nm'ye düşmektedir [11].

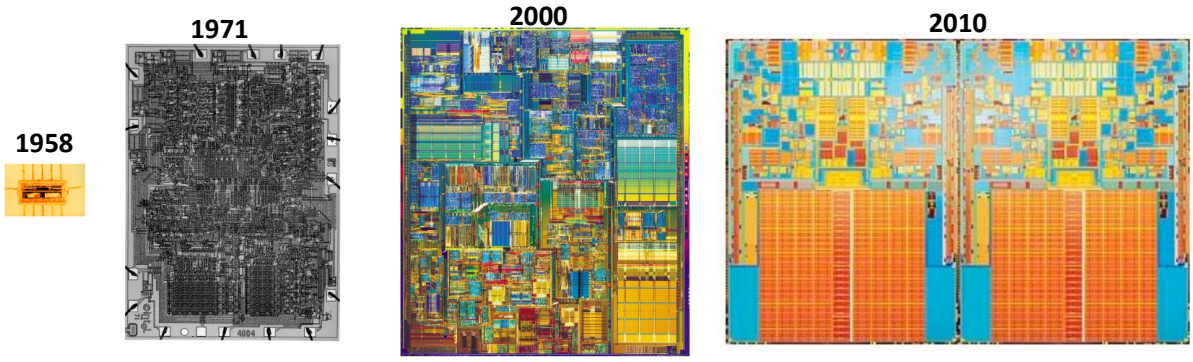


Şekil 2.5: CMOS teknolojisi boyutlarındaki değişim [11].

Transistör teknolojisinde gelinen noktada fiziksel kısıtlamalar nedeniyle ölçeklendirme zorlaşmaktadır. Yeni tasarlanan GAAFET'ler ile gelecekteki kısıtlara çözüm olması beklenmektedir [10].

Tablo 2.3: Teknoloji ölçeklendirme [13].

Transistör Teknolojisi Adı	Yıl
10 µm	1971
6 µm	1974
3 µm	1977
1.5 µm	1982
1 µm	1985
800 nm	1989
600 nm	1994
350 nm	1995
250 nm	1997
180 nm	1999
130 nm	2001
90 nm	2004
65 nm	2006
45 nm	2008
32 nm	2010
22 nm	2012
14 nm	2014
10 nm	2017
7 nm	~2018
5 nm	~2020



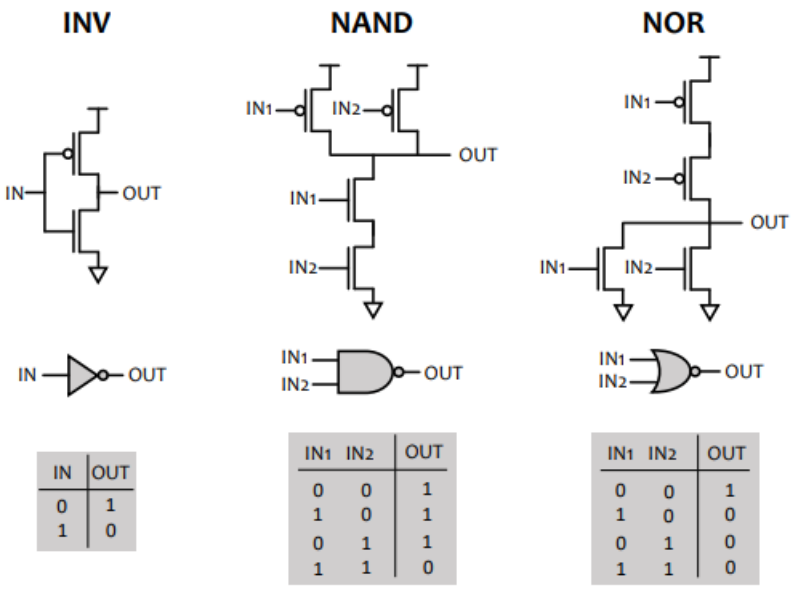
Şekil 2.6: Teknoloji ölçeklendirme [13].

Transistör teknolojisinde yaşanan fiziksel gelişmelerin yanı sıra IC'lerdeki performansı artırmak için tasarım da optimizasyonların yolları aranabilir. Var olan teknolojiler ile IC tasarım

metotlarında iyileştirmeler yaparak performans iyileştirmeleri yapılabilir. Transistörlar büyük bir hiyerarşi oluşturmak için kullanılan temel bileşenlerdir. Tasarım birinci adımında başlayarak üretimdeki son aşamaya kadar mevcuttaki süreçlerdeki hataları minimize ederek performans iyileştirilmesi yapılabilir [10].

2.3.3. Standart Hücreler

Standart hücre bir fonksiyonu yapabilmek üzere tasarlanmış bloktur. Bu hücreler, dijital ortamda IC tasarımı yapabilmek için tanımlanmıştır. Genellikle çip üreticileri tarafından sağlanan ve üretimi için önceden kalifiye edilmiş olan kitaplıklarda (kütüphane) dağıtılır. Standart bir hücre kitaplığı geniş bir Boole mantığı (değil, ve, veya vb.) ve depolama (latch) veya kaydırma (shift) işlemlerini sunar. Böylece üreticinin sağladığı standart hücreler ile evrensel tasarım özellikleri karşılanabilir ve genel işlevsellik sağlanabilir.



Şekil 2.7: Boole lojik işlemlere örnekler [10]**.**

Standart hücrelerin işlevselliğinin sağlanmasının yanı sıra, yongaları bantlarken fiziksel seviyeden türetilen kısıtlamaları veya gereksinimleri de aşılamalıdır. Bu nedenle üreticiler, sürücü gücünden birden fazla opsiyon ile her bir standart hücre işlevini oluşturur ve ayrıca kütüphanede yönlendirme izi (routing track), eşik voltajı (Vth) ve besleme voltajı (Vdd) konusunda da birden fazla versiyon sağlamaktadır [10].

Tüm kütüphanesi için, standart hücre yapılarının oluşturulmasında fiziksel yerleşimde kaç tane yol kanalının kullanılabileceğinin bilgisi de kütüphanede yer almaktadır. Eşik voltajı (Vth),

Transistörü çalıştırabilmek için gereken Transistör kapısında ki minimum voltajdır. Dokümhaneler genellikle dijital IC'lerde kaçak gücü etkin bir şekilde hesaplayabilmek için standart Vth (SVT), yüksek Vth (HVT) ve düşük Vth (LVT) bilgileri kütüphanede yer almaktadır.

Standart hücre kütüphanesi dijital tasarım sürecinin temel yapı taşıdır. Sektörde kullanılan lisanslı CAD programları X-Fab, IBM yada TSMC gibi IC üretim fabrikasyonu olan firmalara ait standart hücre kütüphaneleri kullanmaktadırlar. Bu lisanslı kütüphaneler dağıtılamaz ve ücretsiz kullanımına izin verilmemektedir [14].

IC teknolojisi alanında yetişmiş insan gününün artırılması ve bu sektörün yaygınlaşmasının sağlanması için birkaç açık kaynak kodlu standart hücre kütüphanesi bulunmaktadır. Bu kütüphaneler ölçeklenebilir CMOS kurallarına dayanmaktadır. Bu kurallara dayanarak oluşturulan fiziksel düzen yapısı açık kaynak kod olarak dağıtılabilir. Aynı zamanda kuralların tasarlandığı süreçlerde üretim gerçekleştirilebilir [14].

Tez çalışması kapsamında kullanılan açık kaynak kodlu fiziksel tasarım yapmaya imkan sağlana qflow aracı Oklahome State University (OSU) kütüphanesini kullanmaktadır. Çalışmamda OSU018 ve OSU035 standart hücre kütüphaneleri kullanılmıştır. Standart hücre kütüphanesi ait içeriğinde dizin yapısı Tablo 2.4 kapsamında gösterilmiştir [15].

Tablo 2.4: Qflow aracı transistör teknolojisi dizini [15].

INDEX (Dizin)	
AND2X1	INVX8
AND2X2	LATCH
AOI21X1	MUX2X1
AOI22X1	NAND2X1
BUFX2	NAND3X1
BUFX4	NOR2X1
CLKBUF1	NOR3X1
CLKBUF2	OAI21X1
CLKBUF3	OAI22X1
DFFNEGX1	OR2X1
DFFPOSX1	OR2X2
DFFSR	PADINC
FAX1	PADINOUT
HAX1	PADOUT
INVX1	TBUFX1
INVX2	TBUFX2
INVX4	XNOR2X1

2.3.4. Transistör Teknolojisinde Türkiye

Dünyada çip üreticilerinden önde gelen ülkeler Çin, ABD, Tayvan ve Japonya'dır. Tayvan dünyadaki yarı iletken ihtiyacını yaklaşık %60 oranını karşılamakta iken en yakın rakibi olan Güney Kore yaklaşık %20 oranını alabilmektedir. Dünya çapında çip üreticisi olan firmalar; Texas Instruments, Nvidia Corp, Micron Technology Inc. (MU), Qualcomm Inc.(QCOM), Intel, Samsung Elektronik, Tayvan Yarı İletken İmalat A.Ş (TSMC), SK Hynic Inc ve Broadcom Inc. [16,17].

Türkiye çip sektöründe OTDÜ MEMS ve Bilkent UNAM başta olmak üzere akademik çalışmalar ile hem çip tasarımı hem de üretimi konusunda önemli çalışmalar gerçekleştirmektedir. Bununla birlikte TÜBİTAK YİTAL (Yarı İletken Teknolojileri Araştırma Laboratuvarı) denetiminde ASELSAN, YONGATEK ve ERMEKSAN firmaları Türkiye'de hem üretim hem de satış gerçekleştirmektedir [18].

YİTAL, TÜBİTAK BİLGEM UEKAE bünyesinde yer alan bir yarıiletken araştırma laboratuvarıdır. Kuruluşu 1983 yılında gerçekleştirilmiştir. YİTAL'ın sahip olduğu güncel üretim teknolojileri 0,25 µm 5 metal CMOS ve 0.25 µm 5 metal SiGe teknolojileridir. Yarı İletken teknolojisinde Türkiye'nin kritik uygulamalarında kullanılan askeri standartları destekleyen ürünleri yer almaktadır. Bununla birlikte kâğıt üzerinde bir tasarımı kılıflanmış ve test edilmiş bir IC'ye milli olarak dönüştürebilen yetkinliğe sahip bir kurumdur [19].

ASELSAN ve Bilkent Üniversitesi ortaklığı 2014 yılında başlatılan ile MikroNano projesi başlatılmıştır. Bu projenin hedefi Galyum Nitrat temelli çiplerin üretimi ile stratejik konular olan savunma sanayi alanında ve iletişim (4G/5G) alanlında çalışmalar yürütülmektedir. Firma ilk prototipini 2016 yılında üretmiştir [18].

YONGATEK firması Malezyalı Silterra Malaysia Sdn. Bhd. ile yerli çip üretimi için işbirliğinde çalışmalarını sürdürmektedir. IC tasarımı ve doğrulama hizmeti, video ve görüntü işleme çözümleri sunmaktadır [18,20].

ERMAKSAN firması öz sermayesi ile yarı iletken teknolojileri üretim ve araştırma yatırımı yapan Türkiye'nin ilk özel sektör firmasıdır. Firma mevcut üretiminde yer alan lazer çip ile güç kaynağı, aydınlatma teknolojileri, savunma sanayi, uzay sanayi ve sağlık sektöründe üretimler gerçekleştirmektedir [18,21].

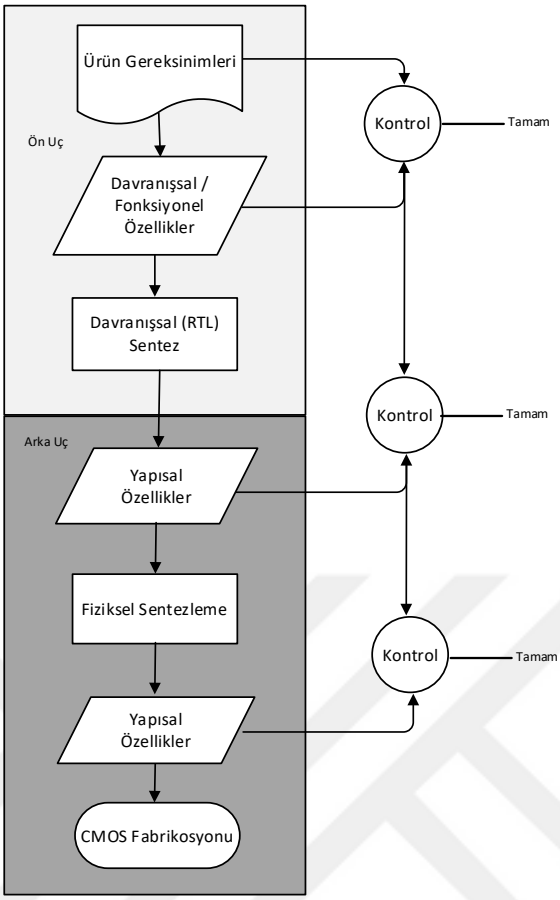
Tablo 2.5: Türkiye'de yarı iletken üretim tesisleri [22].

ÜRÜN	ÜRETİM TESİSİ	ÜRETİM YERİ
CMOS 250nm	YİTAL Araştırma Merkezi	Gebze
CMOS / SiGe 130nm	YİTAL Araştırma Merkezi /ASELSAN	Gebze/Ankara
CMOS 90u-65nm	YİTAL A.Ş / ASELSAN	Gebze/Ankara
MEMS (Ded., İvm., Dön.)	ODTÜ-MEMS / ASELSAN	Ankara
GaN (Yüksek hızlı tr.)	AB Mikro Nano / ASELSAN	Ankara
GaAs (Kızılötesi ded.)	ODTÜ KANAL, ASELSAN	Ankara
InGaAs (Kızılötesi ded.)	ODTÜ KANAL, CÜNAM, ASELSAN	Ankara
HgCdTe (Kızılötesi ded.)	ODTÜ KANAL, ASELSAN	Ankara
Yüksek güçlü lazer (500W CW)	ERMAKSAN	İstanbul

2.3.5. VLSI TASARIM AKIŞI

Elektronik dünyasındaki gelişmeler ve yarı iletken teknolojisinin ilerlemesi sayesinde çip üretimi gelişmiştir. Çip üretimi uzun süreler ve yüklü maliyet gerektiren bir süreçtir. Bu sebepten dolayı, tasarımdan üretime gidiş sürecin de bir dizi özel prosedür yer almaktadır.

Tasarım akışı temelde iki bölümden oluşmaktadır. Tasarım öncelikli olarak davranışsal düzeyde tasarlanır ardından yapısal düzeye geçirilerek devam eder. Aynı, donanım tasarım öncelikli olarak işlemi yapacak şematik tasarımın yapılıp sonrasında layout tasarımının yapılması sürecidir [10-23].

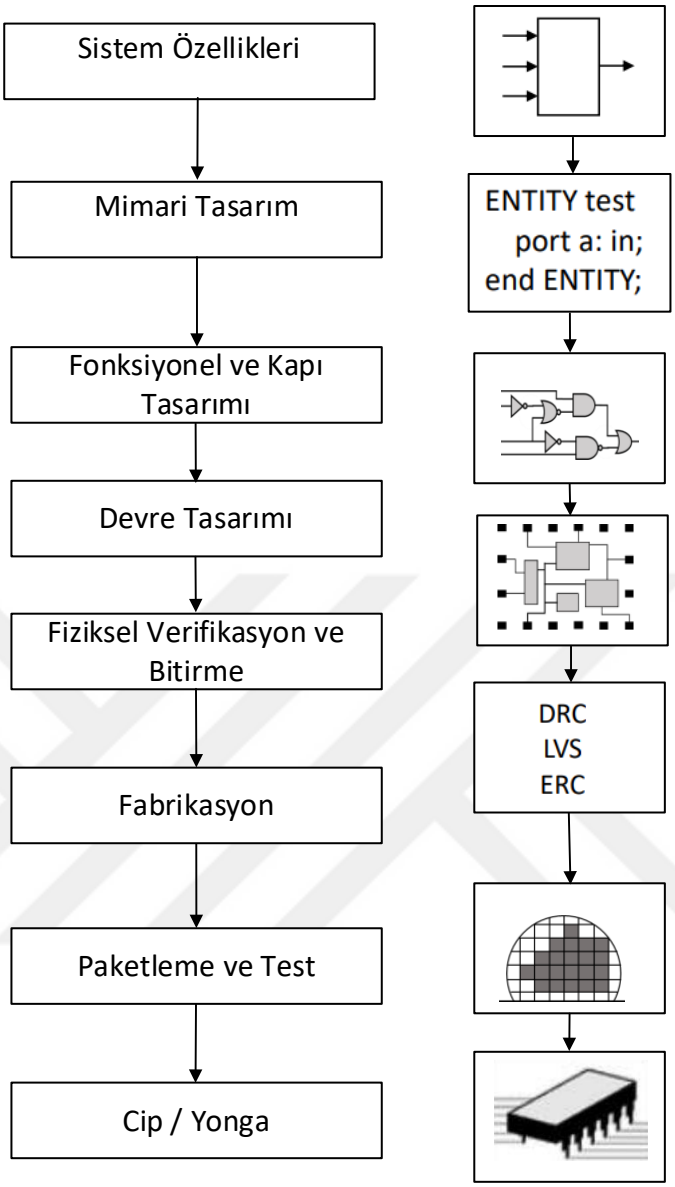


Şekil 2.8: Genel tasarım akışı [23].

İlk aşama olan Davranışsal Sentez tasarımın kapı düzeyindeki (RTL) çalışma durumu sağlanmaktadır. Daha sonraki aşamada, tasarımın çip boyutuna gelmesi sürecindeki fiziksel aşamalardaki işlemleri gerçekleştirilmektedir. Bu aşamaya da Fiziksel Sentez olarak isimlendirilmektedir. Bu süreç bütün olarak Uygulamaya Özel Tümleşik Devreler (ASICs) oluşturmak içinki işlemlerdir.

ASIC'lerin tasarımı ve üretiminin ikiye ayrılmasında etken sebeplerden işgücü ve üretim maliyetinin yüksek olmasıdır. Şirketlerde yatırımlarını ve etki alanları ASIC tasarımı yapan yada üretimini gerçekleştiren olarak ayrılmaktadır.Ticari alanda sentezlemede lisanslı olarak yaygın olan araçlar; Synopsys, Cadence Design Systems, Mentor Graphics ve Synplicity'dir.

Davranışsal Sentez ve Fiziksel Sentez adımlarının alt adımları Şekil 2.9'de gösterilmiştir.

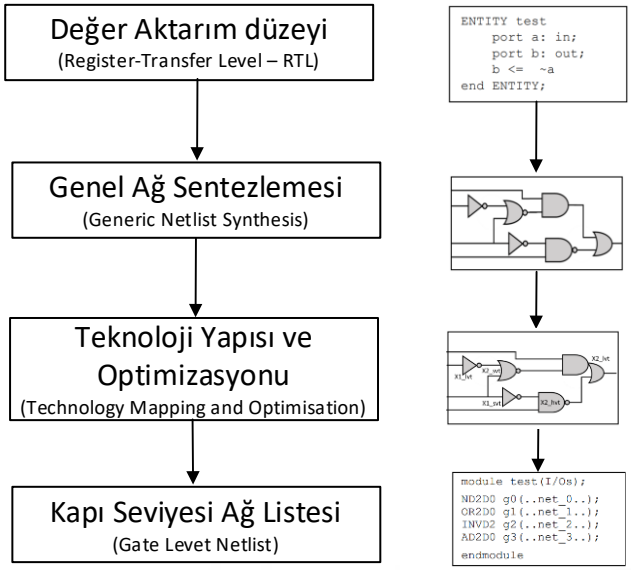


Şekil 2.9: ACIS ayrıntılı tasarım adımları.

2.3.6. Davranışsal (Lojik) Tasarım

ASIC'de Davranışsal Tasarım Ön Uç tasarım (Front-end) olarak isimlendirilmektedir. Temelde 4 adımda gerçekleştirilmektedir [2]. Lojik tasarım işlemi, işlevsel davranışı tanımlayan bir HDL kullanarak RTL elde etmektedir. Tasarım adımlarının tamamlanması sonrasında ayrı ayrı tüm RTL modülleri simüle edilerek doğrulukları kontrol edilmelidir.

Ayrıca, bir IC tasarım sadece mantık tasarımlarından oluşmamaktadır. Tasarımda temelde bulunması gereken makrolarda yer almalıdır. Temelde yer alan makrolara örnek olarak bellek blokları, analog devreler ve giriş/çıkış blokları verilebilir. Mantıksal tasarım yapılır iken bir de zaman ayarlaması ve güç modülleri de karakterize edilmektedir [10].



Şekil 2.10: Genel kapı sentez akış diyagramı.

Davranışsal tasarım süreçlerine ait akış diyagramı Şekil 2.10 kapsamında gösterilmiştir. Her bir adımımda yapılan işlemler sırasıyla açıklanmıştır.

2.3.6.1. RTL - Design Entry – Tasarıma Giriş

Tasarımın bir Donanım Tasarlama Dili (HDL) ile ifade edilmesi aşamasıdır. Bu aşamada Verilog HDL yada VHDL programlama dilleri kullanılabilir.

2.3.6.2. Generic Netlist Synthesis – Genel Ağ Sentezlemesi

HDL olarak yazılan yazılımdan mantık hücreleri ve hücreler aralarındaki bağlantıların kurulması aşamasıdır. Mantık Sentezi aşaması; tansistör teknolojiden bağımsız olarak kapıların optimizasyonu ve davranışsal modelden yapısal modele çevirme adımlarını kapsamaktadır. Bu adımda kullanılan kütüphane zamanlama ve güç için ihtiyaç duyduğu bilgileri .lib dosyasından layout bilgilerini (.lef) dosyalarındaki önceden tanımlanan bilgilerden almaktadır [10].

2.3.6.3. Technology Mapping and Optimisation – Teknoloji Yapısı ve Optimizasyonu

Tasarım programının desteklediği teknolojiye ait standart hücre kütüphanesinden mantık kapıların eşleşmesine göre haritalanır.

2.3.6.4. Gate Level Netlist – Kapı Seviyesi Ağ Listesi

Tasarımın fiziksel yapısını oluşturacak hücreler ve aralarındaki bağlantı listelerinin oluşturulduğu aşamadır.

Doğrulama aşamasında hata tespit edildi ise sentezleme aracı ile tasarım ölçütleri iyileşene kadar mantık blokları yeniden eşlenir yada kapıları yeniden boyutlandırma yapılarak optimizasyon yapılmaktadır.

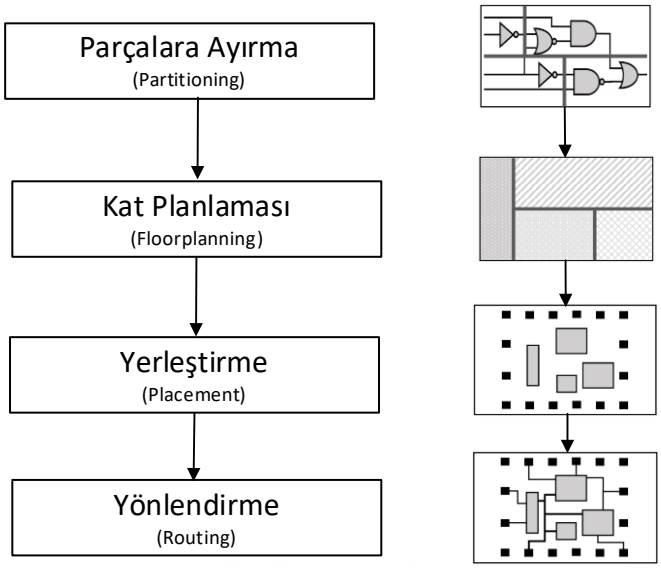
RTL tasarımda tasarımın sahip olduğu zamanlama konusunda fazlaca hatalar ile karşılaşılmaktadır. Bu hatalar VLSI mühendisleri tarafından en iyi optimizasyon ile düzeltilmektedir. Bu düzenleme bazen tüm sentezin yenilenmesi sebep olacak kadar köklü olmaktadır [10].

2.3.7. Fiziksel Tasarım

ASIC'de Fiziksel Tasarım Arka Uç tasarım (Back-end) olarak isimlendirilmektedir. Temelde 4 adım içermektedir [2].

Bir tasarım için elde edilen kapı seviyesinde sentezlenmiş çıktının son sürüne dönüşmesi için gerekli olan ayrıntılı geometrik yerleşiminin yapılması sürecidir. Bu süreç aynı zamanda dijital VLSI akışının önemli bölümü olan fiziksel tasarım olarak isimlendirilmektedir [10]. Fiziksel yerleşim sırasında, tasarımı oluşturan parçalara bir konum atanır ve katmanlı imalat teknolojisi ile metal katmanlar üzerinden yada arasından ara bağlantılar ve yönlendirmeler yapılmaktadır. Fiziksel tasarım sonucunda Grafik Tasarım Sistemi (GDSII) akışı tamamlandığında elde olan kazanım, üretim için bir dizi talimat listesidir.

Günümüzde kullanılan fiziksel üretim araçları tüm süreci otomatik hale getirmektedir. Böylece insandan kaynaklana hatalar minimize edilmektedir. Bu süreç profesyonel alanlarda kullanılan lisanlı fiziksel tasarım araçlarının yanı sıra kullanımı yaygınlaşan açık kaynak kodlu araçlarla da yapılmaktadır.



Şekil 2.11: Fiziksel tasarım akış diyagramı.

Fiziksel tasarım süreçlerine ait akış diyagramı Şekil 2.11 kapsamında gösterilmiştir. Her bir adımımda yapılan işlemler sırasıyla açıklanmıştır.

2.3.7.1. Partitioning - Bölümleme

Tasarımı oluşturan modülleri veya blokların en küçük alt bileşenlerine indirgenerek tasarımda bölümleme yapılır. Bu parçalara ayrıma işlemi tasarımdaki karmaşıklığı azaltmak için yaygın kullanılan bir tasarım stratejisidir. Bu şekilde her bir blok ayrı tasarlanabildiği gibi ayrı olarak analizde edilebilmektedir. Fakat bu durumda bloklar arasında yer alan bağlantılar göz önünde bulundurulmalıdır. Aksi durumda, yanlış yada eksik bağlantılar sonucunda performans kayıpları yaşanabilir [5-10].

2.3.7.2. Floor Planning – Kat Planlaması

Her blok yapısı sabit yada esnek şekle sahiptir. Sert blokların sabit boyutları ve alanları vardır, esnek blokların sabit bir alanı vardır ancak en boy oranları değişkenlik gösterebilmektedir. Tasarımın içerdiği sert ve esnek blokların düzlemde yerleşim kurallarına göre yapılandığı aşamadır [5-10].

2.3.7.3. Placement – Yerleşim

Tasarımda yer alan blokların içerisindeki hücrelerin yerleşimine karar verilmesi aşamasıdır. Tüm yerleşimler önceden belirli alanlara yerleştirilmek üzere yapılmaktadır. Bu adım sırasında, zamanlama kısıtları ile karşılaşılmaktadır. Yönlendirilebilmeyi korurken hücreler arasındaki mesafe veya ara bağlantıları en aza indirmek için optimizasyon gerekmektedir.

Kötü bir yerleşim, daha büyük alan kaplar ve aynı zamanda daha çok zamanlama hatası yaşamaya sebep olmaktadır. Ayrıca, yerleştirme adımı sırasında zaman optimizasyonu öncelikli hedef olur ise alandan tasarruf edilebilmektedir [5-10].

2.3.7.4. Routing – Yönlendirme

Tasarımdaki hücrelerin ve blokların arasındaki bağlantıların yapıldığı aşamadır. Yönlendirme süreci aşamasının, standart hücreler ve ara bağlantılar arasındaki yolların topolojilerin belirlemesi gerekmektedir. Eşzamanlı yönlendirme optimizasyonun hedefi, toplam bağlantıların uzunluğun en aza indirmek ve üst düzey zamanlama çıkartmaktır [5-10].

2.4. FPGA NEDİR?

FPGA (Field Programmable Gate Array, Alanda Programlanabilir Kapı Dizisi); bir veya daha fazla mantık işlemlerini gerçekleştirebilecek mantık kapılarını arasında bağlantıya imkan sağlayan donanımlardır.

Alanda programlanabilmeleri, paralel işle yapabilmeleri, tekrar programlanabilmeleri, simülasyon imkanı sağlaması ve düşük güç tüketim tercih sebepleridir. FPGA'lerde bir saat sinyali içerisinde paralel işlem yapabilmeyi desteklemektedirler. Böylece diğer entegrelerden daha hızlı işlem yapma özelliğine sahiptirler [25,26].

Tasarım geliştirme süreçlerinde simülasyon imkanı sunmaktadır. Tasarlanan donanım FPGA'a yüklenerek üretim öncesinde doğruluk kontrolü yapılabilmektedir. ILA (Entegre Mantık Analizörü) sayesinde hatalar bulunarak doğrulamalar sağlanmaktadır [25].

FPGA'lerin yapısından birde sayısal tümleşik devreler bulunmaktadır. Durağan Rastgele Erişim Bellek (SRAM), giriş/çıkış blokları (I/O'lar), programlanabilir bloklar, yapılandırılabilir mantık blokları ve yönlendirilebilir ara bağlantılardan oluşmaktadır.

2.4.1. FPGA Programlama Dilleri

FPGA programlamada yaygın kullanımda olan iki programlama dili vardır. Programlama dilleri; Verilog HDL ve VHDL'dir. Uygulamanın ihtiyaçlarına göre hangi yazılım dilinin kullanılacağı tercih edilmektedir.

2.4.1.1. Verilog HDL

Verilog yazılım dili ilk 1984 yıllarında Gateway Design Automation şirketi tarafından donanım modelleme için kullanılmaya başlanmıştır. O yıllarda yaygın kullanılan bilgisayar dillerinden

olan C dilinden etkilenmiştir. Verilog simülatörü 1987'li yıllarda kullanımı etkinleşmiştir. Cadence Design System, verilogun geliştiricisi olan Gateway Design Automation'u satın almıştır ve 1990 yılında verilog yazılım dilini sanayinin kullanımının yanı sıra genel kullanıma da açmıştır. Teknolojideki gelişmeler ile yıllar içerisinde gelişmeye devam etmiştir. Son güncellemesi Verilog 2001 olarak isimlendirilmektedir. 2001 versiyonu ile Verilog 1995'te bulunan çeşitli hatalar çözümlenmiştir.

Verilog HDL (Hardware Description Langugel) donanım tanımlama dilidir. Donanımı dijital sistemlerde tanımlayabilmek üzere kullanılmaktadır. Basit bir mantık işleminden gelişmiş bir mikroişlemcinin donanımın tanımlanmasına imkan sağlamaktadır. Verilog 3 farklı soyutlama şekline sahiptir. Davranışsal Seviye (Behavioral Level), Yazmaç Transfer Seviyesi (Register – Transfer Level, RTL) ve Kapı Seviyesi (Gate Level)'dır. Soyutlama modelin üst seviyeden en alt seviyeye verimli olarak inmesine olanak sağlamaktadır [27].

2.4.1.2. VHDL

VHDL (Very High speed ıntegrated circuit Description Language), yüksek hızlı tümleşik devreler için yazılım ile donanım tasarlamak için kullanılan yazılım dilidir. ABD Savunma Bakanlığı tarafında 1981 yılında donanım tanımlama dili olarak tavsiye edilmiştir. Kullanımı yaygınlaşması sonrasında IEEE tarafından 1986 yılında standartlaştırılmıştır. 1987,1993 ve 2002 yıllarında standartta güncelleme gerçekleştirilmiştir.

VHDL kullanılarak programlama da üç temel yapıya sahiptir. Yapılar; Varlık (entity), Mimari (architecture) ve işlem (process)'dir.

- Varlık: Tasarıma ait giriş ve çıkış arayüzlerinin tanımlandığı bölümdür.
- Mimari: Tasarımın işlevinin belirlendiği bölümdür.
- İşlem: Tasarımın işlevinin davranışın tanımlandığı bölümdür.

2.4.2. Tasarımları Doğrulama

Tasarım adımlarında, tasarımın doğrulaması yapılmaktadır. Simülasyon aşamasında, tasarım test sinyalleri ile devrenin lojiğinin ve zamanlamasının doğruluğu test edilmektedir. Programlanabilir lojiğe ait bazı temel test yapıları bulunmaktadır. Yerleştirme ve yönlendirme yapılmadan önce tasarıma ait lojiğin doğruluğu kontrol edilmektedir. Simülasyon aşaması ile

zamanlamasının tam testinin yerleştirme ve yönlendirmeden sonra yapılması daha doğru bir adımdır [28].

Simülasyon aşaması her zaman tavsiye edilmektedir. Davranışsal tasarım sonrasındaki simülasyon, Fiziksel Tasarımdan sonra yapılan simülasyondan daha az tutulabilir. Asıl olan Fiziksel Tasarım sonrasındaki simülasyondaki doğruluktur [28].

Simülasyonlar öncelikli olarak bilgisayar ortamında yapılmakta ardından ya üretim gerçekleştiril çip üzerinde yada hazır modüller üzerinde doğruluk testleri yapılmaktadır.

FPGA üreticiler olan Xilinx, Intel (Altera), Actel, Lattice ve diğerleri arasında pazar payı en yüksek olan iki üretici Xilinx ve Intel (Altere)dır. Bu iki firmanın pazar payının büyük olmasının sebebi ise FPGA tasarı kartlarına kendilerinin sunduğu uygulamalardır. Bu önde gelen firmaların kullandığı çeşitli simülasyon uygulamaları vardır [29].

2.5. MİKROİŞLEMCİ

Mikroişlemciler, milyonlarca Transistörden oluşturulmuş bir ekosistemdir. 1971 yılında Intel firması, silikon çip üzerine binlerce Transistöru yerleştirmeyi başarmıştır. Bu başarı sonrasında bilgisayarların fiziksel boyutları küçükmüş ve erişilebilir bir sistem olmaya başlamıştır.

1950'lı yıllarda iki Transistör ile başlayan serüven günümüzde milyonlarca Transistörün birleşmesi ile oluşan programlanabilir cihazlara dönüşmüştür. Sektörün öncülerinden olan Intel firmasına ait işlemcilerin ve özelliklerinin listesi Tablo 2.6 özelinde görülmektedir.

Tablo 2.6: 2000'li yıllara Intel firmasın ait işlemci bilgileri [26].

	Tarih	Transistör	Mikron	Saat Frekansı	Veri Genişliği	MIPS
8080	1974	6,000	6	2 MHz	8 bit	0,64
8088	1979	29,000	3	5 MHz	16 bit-8 bit	0,33
80286	1982	134,000	1,5	6 MHz	16 bit	1
80386	1985	275,000	1,5	16 MHz	32 bit	5
80486	1989	1,200,000	1	25 MHz	32 bit	20
Pentium	1993	3,100,000	0,8	60 MHz	32 bit-64 bit	100
Pentium II	1997	7,500,000	0,35	233 MHz	32 bit-64 bit	~300
Pentium III	1999	9,500,000	0,25	450 MHz	32 bit-64 bit	~510
Pentium IV	2000	42,000,000	0,18	1,5 MHz	32 bit-64 bit	~1,700

Mikroişlemcilerin yapısında Merkez İşlem Birimi (CPU), kontrol devresi, Aritmetik Mantık Birimi (ALU), kaydediciler ve adres/program sayıcısı bulunmaktadır. Mikroişlemci yapısına ait blok diyagram Şekil 2.12'de yer almaktadır.



Şekil 2.12: İşlemci yapısı ve çevresel birimleri.

Mikroişlemciler programlanabilen transistörlerden oluşmaktadır. Bu transistörlerin durumlarına göre elektriksel işaret transistörler yoluyla iletilmektedir. Böylece işlemcinin yaptığı tüm karmaşık işlemleri toplama, çıkarma, çarpma ve bölme gibi temel seviye

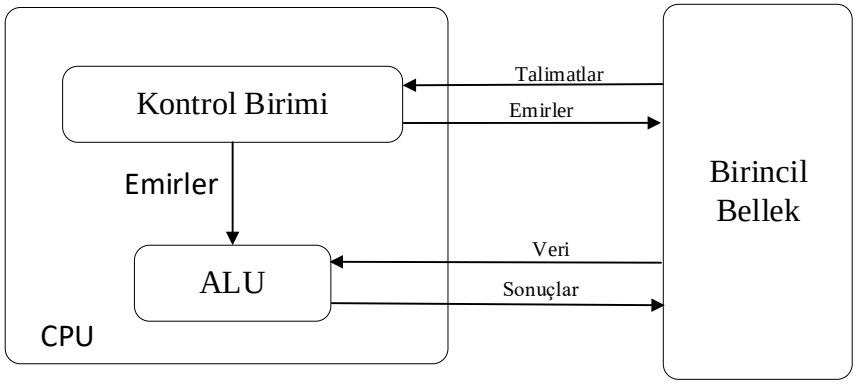
matematiksel işlemlere dönüştürmektedir. İşlemciler 2 bitlik sayma sistem üzerine kurulmaktadır. İşlemci dilinde "1" sinyalin varlığını yani yüksek gerilim (high voltage), "0" ise sinyalin olmadığını yani düşük gerilim (low voltage) olarak ifade edilmektedir [30,31,32].

2.5.1. Kontrol Birimi

İşlemci içerisinde bulunan Kontrol Birimi tüm sistemdeki işlevselliğin sağlanmasından sorumlu olan birimdir. Birimi ait blok diyagram Şekil 2.13'de yer almaktadır. Sistemi oluşturan alt sistemlerin tamamının kontrolünü sağlamak için sinyal üreten birimdir. Ayrıca, sistemde bulunan ALU için gerekli olan sinyalleri üretmektedir. Aynı zamanda bilgisayar sisteminde var olan veri akışını da kontrol sinyalleri ile kontrol etmektedir.

Kontrol Biriminin en yakın çalıştığı birim ALU'dir. ALU'ya bellekten aldığı işlem talimatlarının getirilmesi, gelen talimatların çözülüp yürütülmesi işlevselliğinin kazandırılmasını sağlamaktadır. Aynı zamanda, ALU'dan işlemler sonrasında çıkan bilgilerin bellekte depolanması için Kontrol Birimi sinyal üretmektedir.

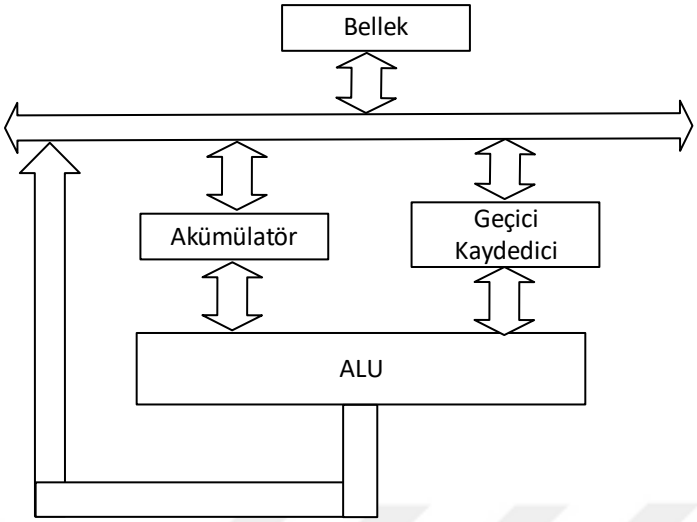
İhtiyaç duyulması halinde işlemcinin durdurulması ve sistemin işlevselliğinin senkronizasyonun sağlanması Kontrol Biriminin ürettiği sinyaller ile sağlanır.



Şekil 2.13: Kontrol birimi şematik [26].

2.5.2. Aritmetik Mantık Birimi (ALU)

Merkez işlem biriminin veri işleme merkezine ALU denir. ALU bir mikroişlemcide aritmetik ve mantık işlemlerinin yapıldığı en önemli birimlerden biridir. Aritmetik işlemler; toplama, çıkarma, çarpma ve bölme ilk başta gelmekte iken mantık işlemleri olarak ve, veya, değil gibi işlemler sıralanmaktadır. Farklı işlemcilerde farklı aritmetik mantık birimleri olması kaçınılmazdır fakat temelde her ALU sıralanan temel mantık işlemlerini içermektedir [33].



Şekil 2.14: ALU biriminin çalışma diyagramı.

ALU'ya ait blok diyagram Şekil 2.14'de yer almaktadır. ALU kapsamındaki alt parçalara dair açıklamalara alt başlıkta yer verilmiştir.

2.5.2.1. Akümülatör

İşlenmek üzere gelen ve işlem sonrası oluşan bilgileri depolamak için akümülatörler kullanılır. Akümülatör yerine veri kaydedici (data reg) kullanılan mikroişlemciler de bulunmaktadır [33].

2.5.2.2. Geçici Kaydedici (Temporary Register)

Bellekten alınan bilgilerin geçici olarak depolandığı ilk alandır [18].

2.5.2.3. Bayrak Kaydedici (Flag Register)

Mikroişlemcilerde ALU'nun yaptığı işlemin sonucunun gösterildiği ve sonuçların değerlendirildiği alanlardır. Bazen hatalar içeride çözülebildiği gibi bazı durumlarda bayrak kaldırılarak dışarıdan düzenleme de mümkün olmaktadır.

Son zamanlarda piyasadaki işlemcilerde çok güçlü ve karmaşık ALU'lar yer almaktadır. Bazı işlemcilerin yapılarında ALU bir aritmetik birime bir de mantıksal birime bölünerek tasarlanmıştır [33].

2.5.2.4. Mikroişlemci Performansı

Mikroişlemcilerin performansı makine diline çevrimin etkinliği ile belirlenir. Makine diline çevrimi belirleyen farklı özellikler bulunmaktadır. Bunlar; çalışma hızı (clock speed), sözcük boyutu (word size), yol genişliği (bus width), iş hattı (pipelining), RISC (reduced instruction set computer) işlem, çoklu işlem ve çok çekirdeklilik gibi özelliklerdir [34].

Mikroişlemcinin çalışma hızı olarak ifade edilen durum, temel işlemleri gerçekleştirme hızıdır.

Sözcük boyutu, makine döngüsüne işleci yoluyla aktarılan veri yada görev miktarıdır. 8 bit bir işlemci bir seferde 8 bit (1 byte) veri aktarabilir iken, 64 bit bir işlemci bir seferde 4 byte veri aktarabilmektedir. Piyasada yer alan işlemcilerinin bazı performans kriterlerine ait bilgilerin karşılaştırması Tablo 2.7 kapsamında yer almaktadır.

Tablo 2.7: Piyasadaki işlemcilerin değerlendirilmesi [34].

Mikroişlemci	**Transistör**	**Saat Hızı**	**Sözcük Boyutu**
Intel Pentium IV	42 milyon	2.4 GHz	32 Bit
AMD Athlon64	105.9 milyon	2.2 GHz	32 Bit
AMDK10	758 milyon	1.6 GHz	64 Bit
Intel i7 Quad	731 milyon	2.93 GHz	64 Bit
Intel Atom Z540	47 milyon	1.86 GHz	64 Bit
Intel i7 Six Core	2.2 milyon	3.9 GHz	64 Bit

Yol genişliği, işlemci ile bellek arasındaki veri yolu genişliğidir. Bu yolda iki veri yolu bağlantısı bulunmaktadır. Bunlar, veri yolu ve adres veri yoludur. Bellek ve işlemci arasındaki verilerin taşındığı hatların tamamıdır. Adres veri yolu ise bellekteki verilen konumları hakkındaki bilgileri bulunduran elektronik yoldur.

Önde gelen mikroişlemci üreticilerinin Intel ve IBM gibi, işlemci hızını daha üste çıkartmak için ek teknikler kullanmaktadırlar. İş hattı özelliği de ek tekniklerden biridir. İş hattı, bir işlem talimatı alınıp işlem başladığı anda başka bir işlemin komutunu da alıp onu da başlatma işlemdir [26].

Azaltılmış Komut Setli Bilgisayar (RISC) tekniği ile üretilen işlemcilerde hız bu teknik uygulanmadan üretilenlere göre daha hazladır [26].

Çoklu işlem yeteneği de yine performans artıran özelliktir. Sistem kartında bulunan komutları aynı anda yürütebilen çoklu işlemlerin birleşimi işlemidir [26].

Çok çekirdekli işlemciler, günümüzde kişisel bilgisayarda daha da yaygınlaşmaktadır. Mikroişlemcilerde işlem kapasitesini artırmak için önceleri Transistör sayısı artırılarak iyileştirmeler yapılmıştır. Bu iyileştirme ile çip üzerindeki ısı miktarı artığından dolayı hedeflenen performans artışına ulaşılamamıştır. Bununla birlikte üretici firmalar ısı miktarını azaltırken işlem hızı artırmak için başka yöntem arayışlarına girmişlerdir. Çalışmalar neticesinde çok çekirdekli mimarı fikri kabul görmüştür. Bu mimaride her çekirdek içerisinde kendine ait ALU birimi ve önbellekleri yer almaktadır.

CPU'larda Transistör sayısını artırmak için başka iyileştirmeler yapmışlardır. Yapılan bu değişiklikler neticesinde çip üzerindeki ısıyı artmış ve mevcut bilgi-işlem talepleri için gerekli performans kazanımlarını elde edememişlerdir. Üreticiler, ısıyı düşürürken ve enerji verimliliğini artırırken işlem hızını arttırmak için farklı bir strateji arayışına girmiş ve sonucunda çok çekirdekli mimari ortaya koyulmuştur. Çok çekirdekli mimaride CPU içerisinde bulunan her çekirdeğin ayrı ALU birimleri ve önbellekleri bulunmaktadır [26].

2.5.3. Bellek Hiyerarşisi

Bellek mikroişlemci sistemi içerisindeki bilgilerin saklandığı birimdir. İki bölüme ayrılmıştır; ROM ve RAM. İlki programları saklar, diğer bölüm ise işlemcinin gerek duyduğu bilgileri yada verileri tutmak için kullanılır [26,35].

Sadece Okunur Bellek (ROM), sahip olduğu bilgiler sadece üzerinden kopyalanabilir. Üreticisi yada kullanıcısı tarafından ROM belleğe kaydedilen hiçbir bilgi değiştirilemez ve veri kaydı yapılamaz.

Tek seferlik programlanmış ve geçici olmayan hafızaya sahip bellektir [26,35].

Rastgele Erişimli Bellek (RAM), üzerinde yazma ve okuma işlemlerinin yapılabildiği bellektir. Başka bir ifade ile, değişken veri saklayabilen ve rastgele erişilebilen bellekttir. RAM'lar bir elektronik devre olduğu için besleme gerilime ihtiyacı duyarlar, Bundan dolayı güç kesintisi yaşandığında saklanan veriler kaybolabilir [26,35].

Bunlardan başka bellek türleri ve yer almaktadır. Örnek vermek gerekir ise, EPROM, silinebilir programlanabilir ROM'dur.

3. YÖNTEM

3.1. QUARTUS PROGRAMI – LOGİC TASARIM

Quartus II programı Intel (Altera) firması tarafından mantık devrelerinin tasarlanması ve benzetilmesi amacıyla geliştirdiği araçtır. Quartus II modeli güncel donanım tanımlama dilleri olan VHDL ve Verilog dillerini desteklemektedir [36].

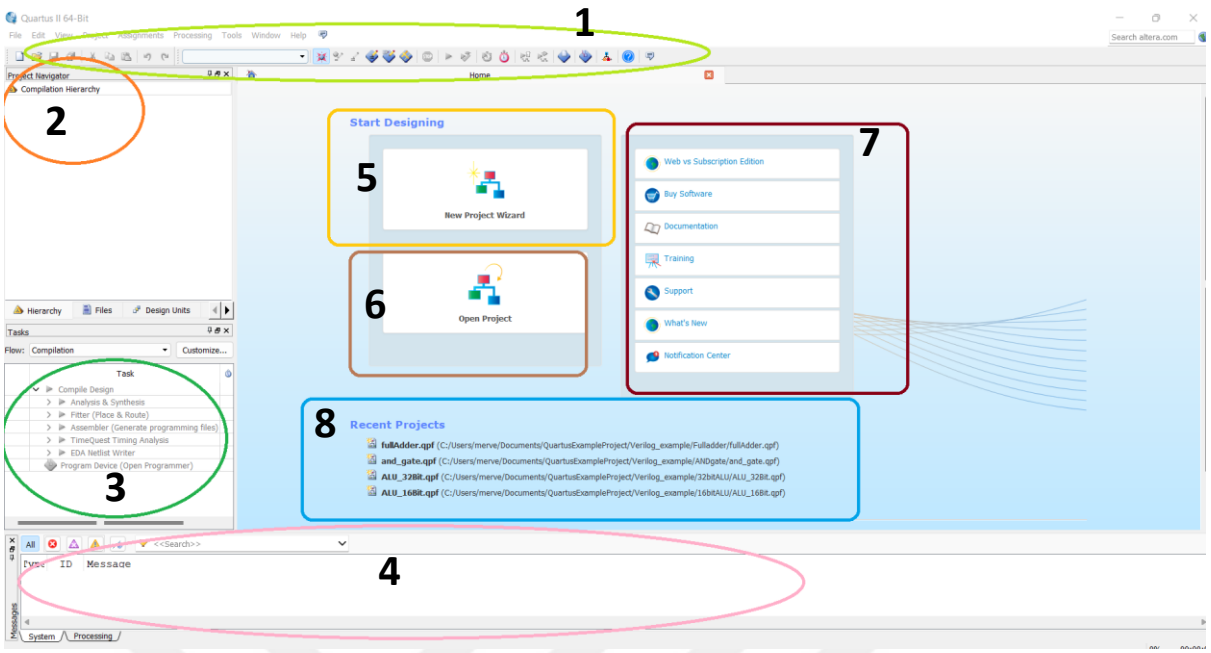
Quartus II'nın ücretsiz sürümü ve full sürümü bulunmaktadır. Web-Edition Quartus II ücretsiz sürümü olarak Quartus II Subscription – Edition ücretli sürümüdür. İki sürümü arasındaki özellikler Tablo 3.1 da karşılaştırılmıştır [37].

Tablo 3.1: Quartus II sürümü özellikleri.

Özellikler	Lisanslı Ürün	Internet Sürümü
İşletim Sistemi	32-64 bit Windows, Linux	32 bit Windows
Desteklenen Aygıtlar	Tüm aygıtlara destek	Bazı aygıtlara destek
Çoklu işlemci Desteği	Var	Yok
Hızlı Derleme	Var	Yok
Kademeli Derleme	Var	Yok
IP Desteği	Sınırlı Ücretsiz	Ücretli
Maliyet	Ücretli	Ücretsiz

Quartus II programı kullanılarak şematik ve donanım tanımla dilleri ile dizayn oluşturulabilir, oluşturulan dizayn sentezlenebilir. Sentezleme sonrasında tasarımın yerleştirilmesini ve yönlendirme işlemlerini gerçekleştirebilmek üzere özelliklere sahiptir. Sentezleme sonrasında zaman ve güç analizi bilgisini de sunmaktadır [37]. Tüm bunlar aslında Quartus II yazılımının davranışsa tasarım yapabilen uygun bir araç olduğunun kanıtıdır.

Yazılımın ücretsiz sürümünün kurulumu Intel firmasının resmi web sitesinden ücretsiz olarak kayıt olunarak erişim sağlanmaktadır. Windows ve Linux işletim sistemleri için kurulumları mevcuttur [38].



Şekil 3.1: Quartus II giriş ekranı.

Kurulum yapıldığında açılan ilk sayfa Şekil 3.1'de tanıtılmıştır. Açılan pencere numaralandırılarak tanıtımı yapılacaktır.

1 ile numaralandırılan bölüm; araç çubuğudur. Buradan sentezle işlemi simülasyon işlemi, pin tanımlama ve atama işlemleri, yazılım yükleme işlemi ve görünümü değiştirme işlemleri yapılabilmektedir.

2 ile numaralandırılan bölüm; Project Navigator'dur. Bu bölümden aktif çalışılan proje ismi (Entity), kodlama sırasında oluşan alt dosya isimleri ve uzantılarını görüntülemektedir. Ayrıca, projede yer alan çeşitli alt birimleri de bulundurmaktadır.

3 ile numaralandırılan bölüm, projenin sentezlenmesi aşamasının yüzdelikli olarak görüntülendiği bölümdür. Projenin sentezlenmesi işleminin bağlantılarını denetler, sentezleme sonrasında zaman analizini görüntülendiği bölümdür.

4 ile numaralandırılan bölüm, projenin üzerinde çalışırken, sentezlerken yada yükleme esnasında uyarıların yada hatalarının görüntülendiği penceredir. Hata kodlarını kırmızı ile uyarıları ise sarı ile renklendirmektedir.

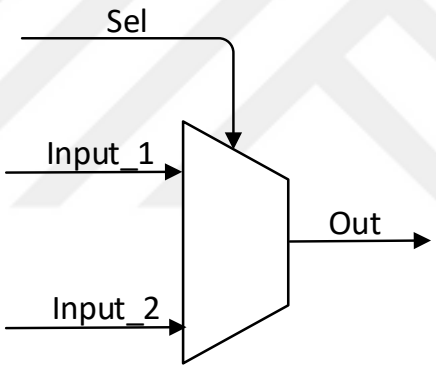
5 ile numaralandırılan bölüm yeni proje oluşturmak için olan bölümdür. Bu bölümden istenilen lojik tasarım projelendirilebilmektedir. 6 ile numaralandırılan bölümden daha önce oluşturulmuş projelere erişim sağlanabilmektedir.

7 ile numaralandırılan bölümden ise üretici firma tarafından kullanılan yazılım hakkında tanıtıcı bilgilere erişim sağlanmaktadır.

8 ile numaralandırılan bölümde en son erişim sağlanan projeye hızlı erişim linkinin bulunduğu bölümdür.

3.1.1. ALU Tasarımı

ALU tasarımı Verilog dilinde kodlanmıştır. Tasarım temel mantık işlemlerini ve aritmetik işlemleri kapsamaktadır. Verilog dilinde davranışsal (behaviral style) tarzda tasarım yapılmıştır.



Şekil 3.2: ALU tasarım mimarisi.

Davranışsal modelleme “always” bloklarının içeriğinde yer alan “if-else” yada “case” yapıları ile ifade edilmektedir. “Always” bloğu içerisinde sadece reg tipi sinyallere değer ataması yapılmaktadır.

Tasarımı yapılan ALU bloğunda yer alan mantıksal işlemler ve aritmetik işlerinin kısa tanıtımı Tablo 3.2 kapsamında yapılmıştır. Tasarımda yer alan girişlere Input_1 ve Input_2 isimlendirmesi yapılmıştır. Sel ALU tasarımın işlem seçim operatörüdür. Out ise çıkış olarak tanımlanmaktadır.

Tablo 3.2: ALU işlem listesi.

ALU İşlemi	Sel Ataması
Toplama İşlemi	0
Çıkartma İşlemi	1
Tersini Alma İşlemi	2
VeDeğil İşlemi	3
VeyaDeğil işlemi	4
Ve İşlemi	5
Veya İşlemi	6
Özelveya İşlemi	7

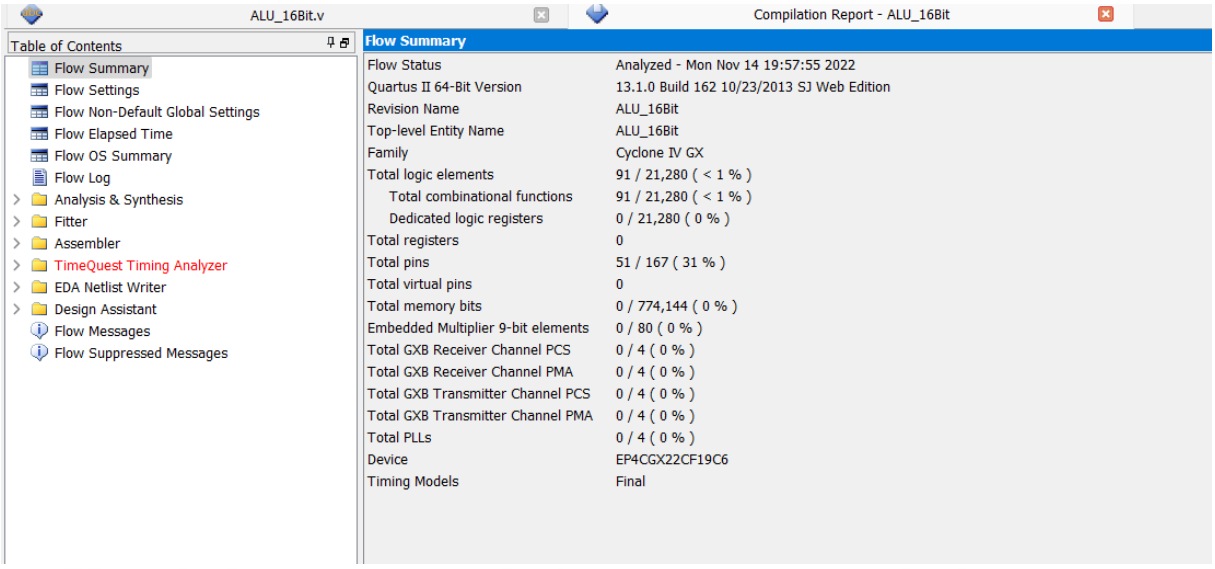
```
module ALU_16Bit(
    Input_1,
    Input_2,
    Sel,
    Out );
    //giriş ve çıkışların tanımlanması
    input [15:0] Input_1,Input_2;
    input [2:0] Sel;
    output [15:0] Out;
    wire [15:0] Reg1,Reg2;
    reg [15:0] Reg3;

    //İşlemleri yapmak için girişlerin dahili değişkenlere atayın
    assign Reg1 = Input_1;
    assign Reg2 = Input_2;
    //Çıkışın değişkene atanması
    assign R = Reg3;

    //Always bloğunun girişlere hassasiyetinin tanımlanması
    always @(Op or Reg1 or Reg2)
    begin
        case (Op)
         0 : Reg3 = Reg1 + Reg2;  //addition
         1 : Reg3 = Reg1 - Reg2; //subtraction
         2 : Reg3 = ~Reg1;  //NOT gate
         3 : Reg3 = ~(Reg1 & Reg2); //NAND gate
         4 : Reg3 = ~(Reg1 | Reg2); //NOR gate
         5 : Reg3 = Reg1 & Reg2;  //AND gate
         6 : Reg3 = Reg1 | Reg2;  //OR gate
         7 : Reg3 = Reg1 ^ Reg2; //XOR gate
        endcase
    end
endmodule
```

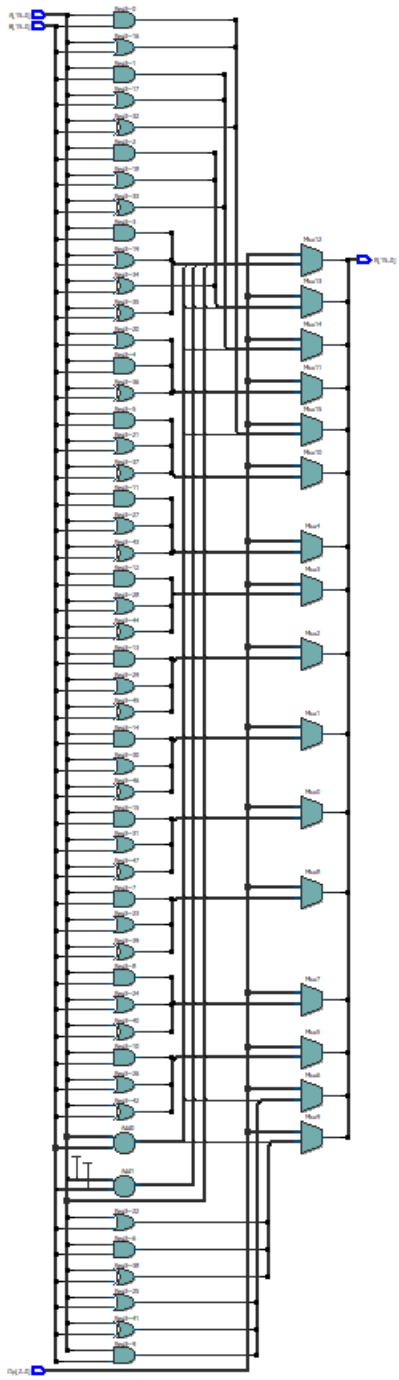
Şekil 3.3: 16 bit ALU tasarımı verilog kodu.

Verilog dilinde tasarım tamamlanmasından sonra kodun sentezlene bilirliğini kontrol etmek için Şekil 3.1'de 1 ile numaralandırılan bölümde " ▸ " ikonu ile belirtilen Start Compilation butonuna basılarak kodumuz sentezlenir. Sentezleme işleminin başarıyla tamamlanması sonrası "Compilation Report" sayfası açılmaktadır. Açılan pencere Şekil 3.4'de gösterilmektedir.



Şekil 3.4: Compilation sayfası.

Başarılı sentezleme sonrasında tasarımın RTL modellemesini oluşmaktadır. Şekil 3.1'de 3 ile numaralandırılan bölümde Compile Design/Analysis&Synthesis/Netlist Viewer/RTL Viewer sekmesinden erişilmektedir. Tasarıma ait RTL model görüntüsü Şekil 3.5 ile gösterilmektedir.



Şekil 3.5: 16 bit ALU tasarımının RTL modeli.

Tasarım sentezlemesinden sonra simülasyon adımı yer almaktadır. Bu adım için tekrar Verilog kodu oluşturulur. Bu sefer simüle edilmek üzere kodlanır. Test benche ait yazılan kod Şekil 3.6 özelinde gösterilmektedir. Kodun yazımının doğrulunu kontrol etmek için Şekil 3.1 kapsamında belirtilen 1 ile numaralandırılan bölümde “ ▶ “ ikonu ile belirtilen Start Compilation butonuna basılarak Test bench kodu sentezlenir.

```
module tb_ALU_16Bit;
    // Inputs
    reg [15:0] Input_1;
    reg [15:0] Input_2;
    reg [2:0] Sel;

    // Outputs
    wire [15:0] Out;

    // Instantiate the Unit Under Test (UUT)
    ALU_16Bit uut (
        .Input_1(Input_1),
        .Input_2(Input_2),
        .Sel(Sel),
        .Out(Out)
    );
    initial begin
        // Apply inputs.

        Input_1 = 16'b0011001100110011;
        Input_2 = 16'b0000000100000001;
        Sel = 0; #100;
        Sel = 1; #100;
        Sel = 2; #100;
        Sel = 3; #100;
        Sel = 4; #100;
        Sel = 5; #100;
        Sel = 6; #100;
        Sel = 7; #100;
    end

endmodule
```

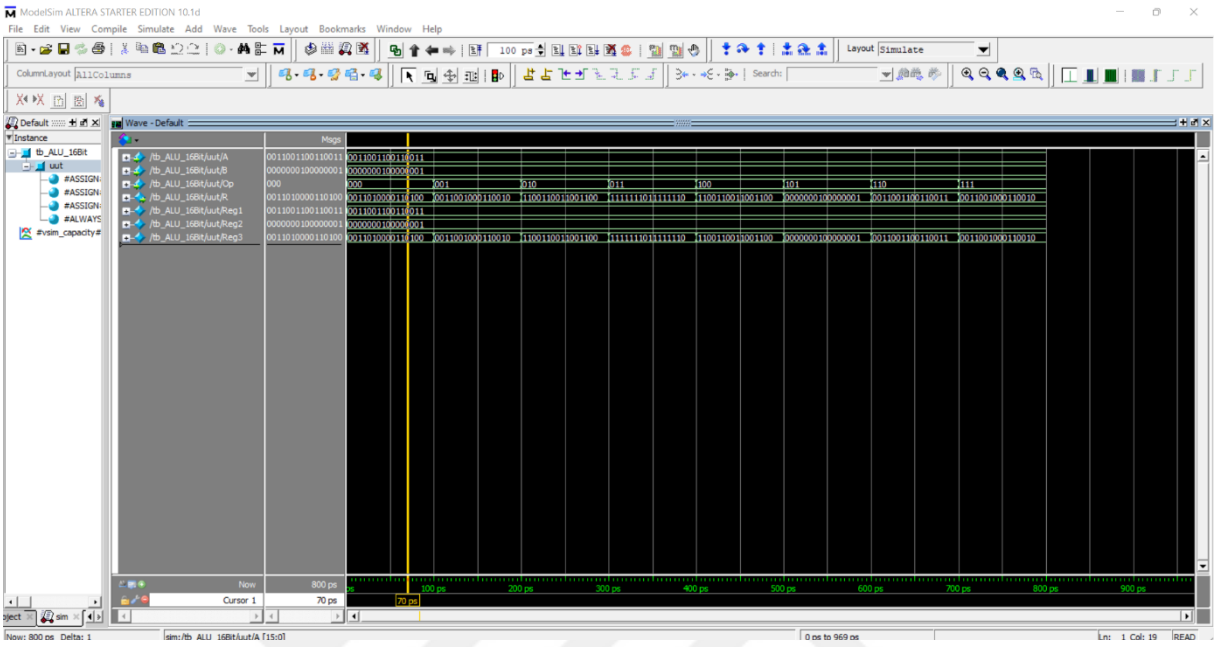
Şekil 3.6: 16 bit ALU tasarımının testbench kodu.

3.2. MODELSİM – SİMÜLASYON

Quartus II programının HDL kodlarının doğrulanması ve simülasyonu için kullandığı program Modelsim programıdır. Ücretsiz olarak kullanılabilen üretici destekli simülasyon aracıdır. Kurulumu hızlı ve kolay olarak yapılmaktadır [39].

Test bench'i oluşturulan modeli ModelSim ile simülasyonunun yapılması için kodun simülasyona atılarak simüle edilmesi gerekmektedir. Simülasyonu yapılacak olan tasarıma ait Test Bench dosyası "Work" kütüphanesinin altına eklenmesi gerekmektedir. "Work" kütüphanesi seçilerek Araç Çubuğunda yer alan Compiler sekmesine basılarak tasarım ve test bench çalışması yapılan dosya dizinine gidilir. İlgili dosyalar açılan pencerede önce "Compile" ardından "Done" butonlarına basarak eklenti sağlanır. Böylece çalışma kütüphanesine

simülasyonu yapacağımız dosya eklenmiş olmaktadır. ModelSim ile çalıştırılan test bench koduna ait çıkış grafiği Şekil 3.7'de görülmektedir.



Şekil 3.7: ModelSim simülasyon sonucu.

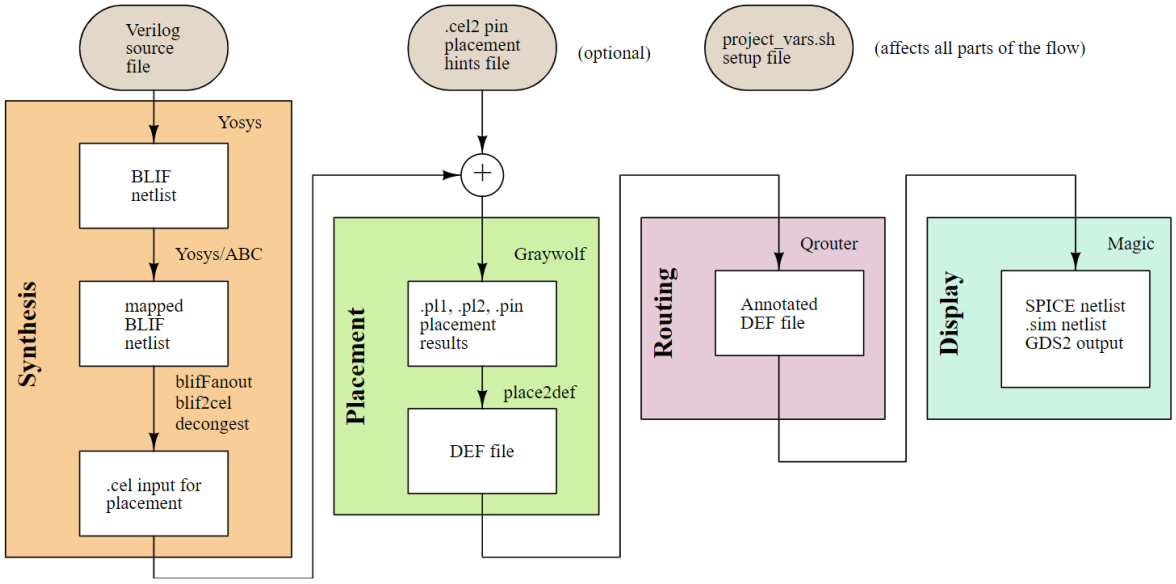
3.3. Qflow PROGRAMI – FİZİKSEL SENTEZ

Açık kaynak kodlu Qflow aracı, RTL tasarımı Verilog dilinde yapılmış olan tasarımın, belirli bir üretim teknolojisinde ve sürecinde dijital devresinin sentezlenmesini simule eden araçlar zinciridir.

Ticari olarak, çip tasarımında yaygın ve efektif olarak kullanılan dijital sentez araçları Cadance ve Synopsys gibi yazılımlardır. Dijital sentezleme algoritmalarının tümü kapalı kaynaklı yazılımlara bağlıdır ve geliştirilmeleri birkaç EDA yazılım satıcıları tarafından kontrol edilmektedir. Bu yazılımların fiyatları küçük ve orta ölçekli dijital sentezleme işlemleri için oldukça yüksek kalmaktadır.

qFlow 1.4 yazılımı, RTL modellemesi yapılan tasarımı GDS II düzenine dönüştürmek için kullanılan bilgisayar destekli fiziksel dizayn (CAD) programıdır. Bu yazılım General Public License (GPL) yani açık kaynak kodlu bir yazılımdır. Bu CAD R.Timothy Edward tarafından ilk olarak 2013 yılında geliştirilmiştir. Son sürümü 10.06.2021 tarihinde yayınlanmıştır. Qflow'u oluşturan alt yazılımlar da mevcuttur. Bunlar; Magic, Graywolf ve Yoys gibi harici araçlardır [45].

Programın sentezleme sürecinde kullandığı araçları, mantıksal denklik kontrolü (LEC) ve Düzene karşı şematik (LVS) araçlarının şematik gösterimi Şekil 3.8'de verilmiştir.



Şekil 3.8: QFlow yapısal şeması.

- Mantık Sentez aşamasında Yosys[40] ve Odin II [41] alt yazılımları kullanılmaktadır. Bu işlem sonucunda netlist bağlantıları .BLİF formatında oluşmaktadır.

- Yerleşim ve Rota aşamasında graywolf [42] alt programı ilk blok yerleşimleri ve yönlendirmede, qrouter [43] alt programı ise spesifik yönlendirme adımında kullanılan alt programlardır.

- Statik Zaman Analizi (STA) için OpenSTA [44] yada vesta [45] alt programları kullanılmaktadır.

- Tasarım Kontrolü (DRC) için Magic [46] alt programı kullanılmaktadır.

- LEC ve LVC işlemleri için sırasıyla Yosy ve Netgen [47] alt programları kullanılmaktadır.

Standart hücre kitaplığı olarak açık kaynak kodu bulunan MOSIS'ın "ölçeklenebilir CMOS kurallarına "dayanmaktadır. Qflow, Oklahome State University (OSU) [48] açık kaynak kodlu hücre kitaplığını kullanmaktadır. OSU 180-500 nm ve FreePDF 45nm teknoloji hücre kütüphaneleri içermektedir. Sektörün önde gelen isimlerinden X-Fab, IBM yada TSMC gibi

üretim tesisi bulunan grupların standart hücre kitaplıklarını paylaşmaları olası görülmemektedir.

3.3.1. QFlow Kurulum

Qflow kurulumu Linux işletim sistemi üzerinde çalışacak şekilde tasarlanmıştır. Bu durumda 2 altenatif oluşmaktadır. Bunlar sırasıyla işletim sistemi Linux bir PC ile çalışmak yada Windows işletim sistemine sanal makine kurup içerisine Ubuntu işletim sistemi kurmaktır. Tez kapsamındaki çalışmalar Windows işletim sistemine sanal makineye kurulan Ubuntu 16.2.4 işletim sistemi kurulumu yapılmıştır. Sanal ortamı sağlayan yazılım WMware WorkStation'dır.

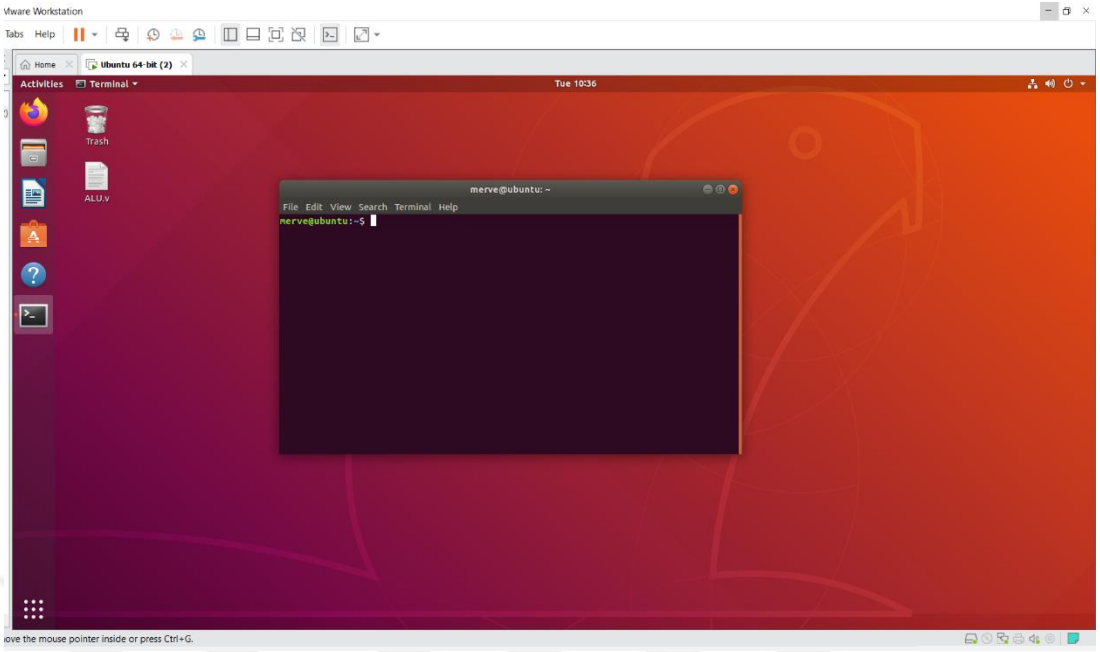
3.3.1.1. Qflow Kurulumu

Windows işletim sistemi olan bir makinede, WMware WorkStatiıon sanallaştırma ortamı üzerine Ubuntu işletim sistemi kurulmuştur. Qflow programının kurulumu için komut setleri listelenmiştir.

Qflow programının çalışabilmesi için bağlı olduğu bazı paketler vardır. Öncelikli olarak bunların kurulumları yapılmalıdır [49].

Tablo 3.3: Qflow kurulumu öncesi olması gereken kurulumlar.

Paket Adı	**Versiyon**	**Kullanıldığı Yer**
python	3 (2.7 uygun değil)	Qflow GUI arayüzü
Tclsh ve wish	8.6	Magic,Netgen,Qrouter ve diğerleri
tcsh	6.18 (yada diğer versiyonları)	Bazı qflow temel scriptleri



Şekil 3.9: WMware ile Ubuntu16.2.4 tanıtım.

Açılan komut sayfasına yazılacak komutlar [50];

```
>_
1  sudo apt update                    // Ubuntunun paket yöneticisi güncellenmesi
2
3  sudo apt install csh tcsh -y       //ihtiyaç duyulan paketlerin yüklemesi
4
5  sudo apt install wget -y
6  sudo apt install curl -y              // ihtiyaç duyulan paketlerin yüklemesi
7
8  sudo apt install build-essential git -v   // dosya indirme için ihtiyaç duyulan paketle
9
10 sudo apt install build-essential git -v // zipli dosyaları kullanabilmesi için
11
12 sudo apt install checkinstall zlib1g-dev libssl-dev -y    // ihtiyaç duyulan diğer kurulumla
13 sudo apt install libgsl-dev -y                            // yüklenmesi
14 sudo apt install libx11-dev -y
15
16 sudo apt install tcl-dev tk-dev -y              // Qflow gui arayüzü için python
17 sudo apt install python3-tk -y                  // yüklenmesi
```

Qflow programını oluşturan birden fazla alt program vardır. Bunlar; yoys, graywolf, qrouter ve magic ve netgen'dir. Ayrıca, Qflow programının kullanıcı kolaylığı için GUI arayüzü bulunmaktadır. Bunun kullanımı için de qflow kurulumu gerekmektedir. Sırası ile alt programların kurulumları ve işlevsellikleri belirtilmiştir.

Tablo 3.4: Qflow alt programları ve işlevsellikleri.

Aracın Adı	**Görevi**
Yoys	Verilog Kodunun Ayrıştırılması/Sentezlenmesi/Optmizasyonu/Doğrulaması
Graywolf	Atama
Qrouter	Detaylı Yönlendirme
Magic	Görüntüleme, Çıkartma, DRC kontrolleri, GDS Üretimi
Netgen	LVS

- Yoys Yüklemesi

```
>_
1  sudo apt install yosys -y     // paketin yüklenmesi
2  yosys -V                 // yoys versiyon bilgisi sorgulama
```

- Graywolf Yüklenmesi

```
>_
1  # masaüstü dizininle olunmalı (~)
2
3  git clone https://github.com/rubund/graywolf.git
4  cd graywolf
5  git checkout 0.1.6  // yoys versiyon bilgisi sorgulama
6  mkdir build
7  cd build
8  cmake ..
9  make
10 sudo make install
```

- Qrouter Yüklenmesi

```
>_
1  # masaüstü dizininle olunmalı (~)
2
3  git clone https://github.com/RTimothyEdwards/qrouter.git
4  cd qrouter
5  git checkout qrouter-1.4 // yoys versiyon bilgisi sorgulama
6  ./configure
7  make
8  sudo make install
```

- Magic yüklenmesi

```
>_
1  # masaüstü dizininle olunmalı (~)
2
3  git clone https://github.com/RTimothyEdwards/magic.git
4  cd magic
5  git checkout magic-8.3 // yoys versiyon bilgisi sorgulama
6  ./configure
7  make
8  sudo make install
```

- Netgen yüklenmesi

```
>_
1  # masaüstü dizininle olunmalı (~)
2
3  git clone https://github.com/RTimothyEdwards/netgen.git
4  cd netgen
5  git checkout netgen-1.5 // yoys versiyon bilgisi sorgulama
6  ./configure
7  make
8  sudo make install
```

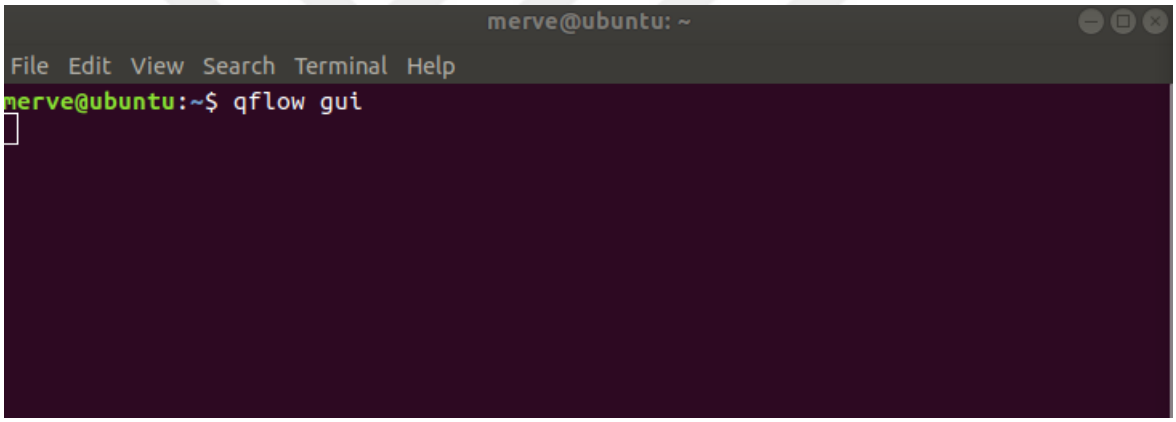
- Qflow Yüklenmesi

```
# masaüstü dizininle olunmalı (~)

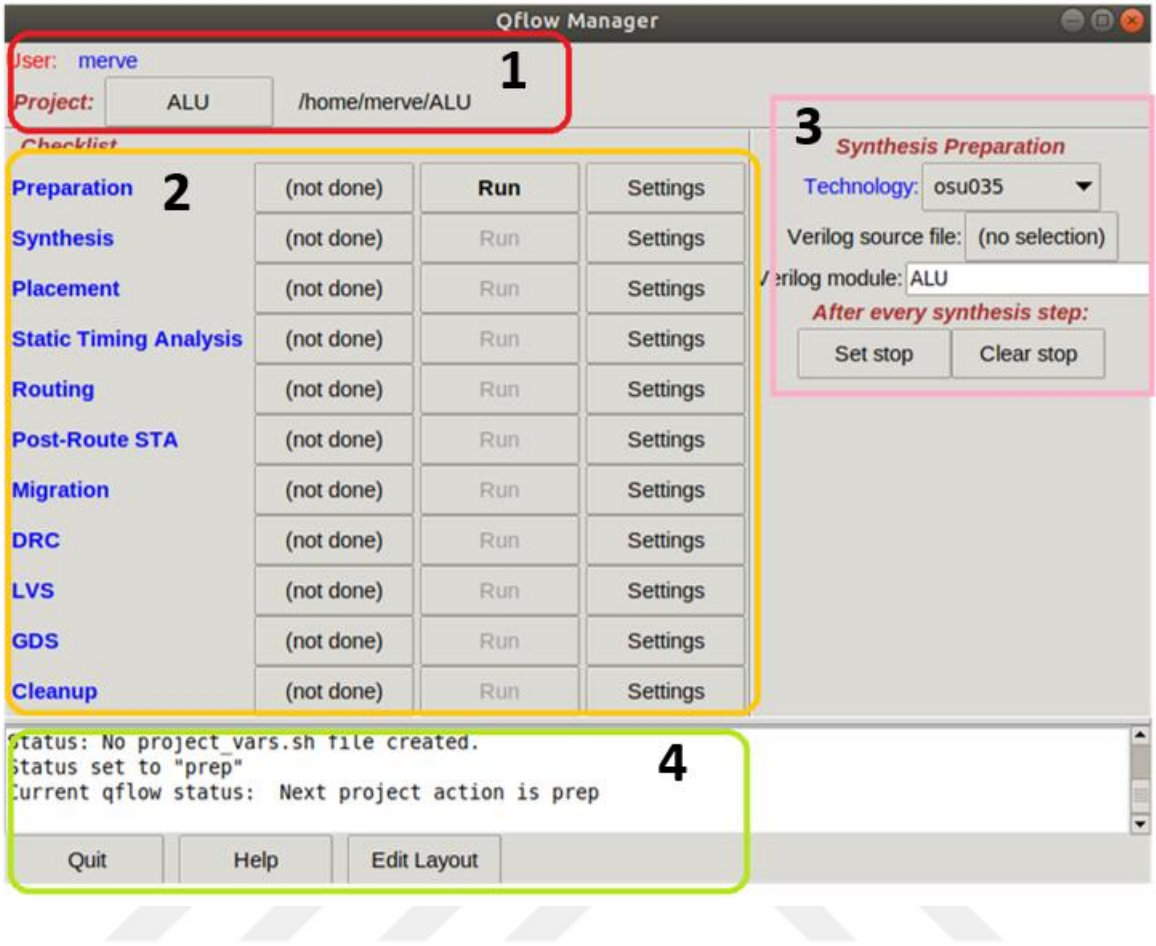
git clone https://github.com/RTimothyEdwards/qflow.git

cd qflow
git checkout qflow-1.4 // yoys versiyon bilgisi sorgulama
./configure
make
sudo make install
```

Kurulumu tamamlanmasından sonra Qflow programının kullanıcı arayüzünü terminal sayfasından ulaşılabilmektedir. İlgili komut satırına ait görsel Şekil 3.10'de gösterilmektedir.



Şekil 3.10: Qflow Gui terminalden açılması.



Şekil 3.11: Qflow kullanıcı arayüzü.

1 numara ile numaralandırılan bölüm; projenin olduğu dosya yolunu belirtmektedir.2 numara ile numaralandırılan bölüm; fiziksel sentez sırasında yapılan adımları listelemektedir. Satırların yanlarında bulunan 3 kutucukta sırasıyla işlem adımının tamamlanıp tamamlanmadığını, İşlemi başlatma ve işlem adımını ilgilendiren ayarlar için kutucuklar bulunmaktadır. Her işlem adımında üçüncü kutucukta ki ayarlar farklılık göstermektedir.

Tablo 3.5: Qflow checklist işlevsellikleri.

İşlem Adı	**Görevi**
Preparation	Sentezlenecek dosya kontrolü ve sentez yapılacak teknolojiyi kontrolü sağlanır.
Synthesis	Mantık sentezi, standart hücre eşleme ve yük dengeleme, yayılım azaltma ve tıkanıklık giderme.

Placement	Tampon ağacı optimizasyonu ile standart hücrelerin yerleştirilmesi
Static Timing Analysis	Yönlendirmelerden kaynaklı gecikmeler dikkate alınmadan yapay olarak hesaplanan veri
Routing	Detay yönlendirme
Post-Route STA	Yönlendirmeler sonrasında hesaplanan veri
Migration	Düzen ve soyut görünümlerin oluşturulması ve netlist çıkarma
DRC	DRC (Düzen tasarım kuralı kontrolü) analizi
LVS	LVS (Yerleşime karşı şematik) analizi
GDS	GDS maskesi verilerinin oluşturulması
Cleanup	Geçici çalışma dosyalarının kaldırılması

3 numara ile numaralandırılan bölüm; sentezde kullanılacak verilog uzantılı dosyanın adını ve sentezleme işleminin hangi teknoloji dosyası ile yapılacağının seçildiği bölümdür. Qflow aracının desteklediği transistör teknoloji dosyaları osu018, osu35, osu050 ve gscl45nm'dir. Bunlardan fiziksel sentezde kullanılan metal katmanları tam olan osu035 ve osu018'dir. Bundan dolayı tez kapsamında bu teknoloji dosyaları ile sentezleme yapılmıştır.

4 numara ile numaralandırılan bölüm, sentez adımlarında bilgi notlarının yazdığı bölümdür.

3.3.2. QFlow Kullanım

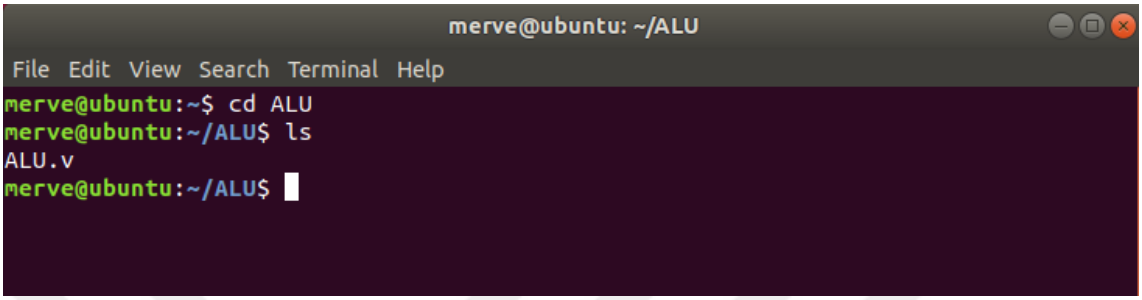
Qflow programının kullanıcı dostu bir GUI arayüzü bulunmaktadır. Arayüzü kullanılarak bir fiziksel sentez yapımının adımları belirtilmiştir.

Sentezleme işleminin yapılabilmesi için öncelikle .v uzantılı kaynak dosyamız olmalıdır. Sentezleme sırasında birden fazla dosya oluşmaktadır. Bunların bir klasör yapısı içerisinde olmasında fayda vardır. Bundan dolayı öncelikle sentezleme işleminin yapılacağı bir dosya oluşturulur ve o dosya içerisine kaynak olan .v uzantılı dosya kayıt edilir.

Örnek olarak sentezin yapılacağı klasörün adı ALU ve sentezlenen kaynak dosyamızın adı ALU.v dosyasıdır. Terminalde bu klasör dizinine erişim sağlanmalıdır.

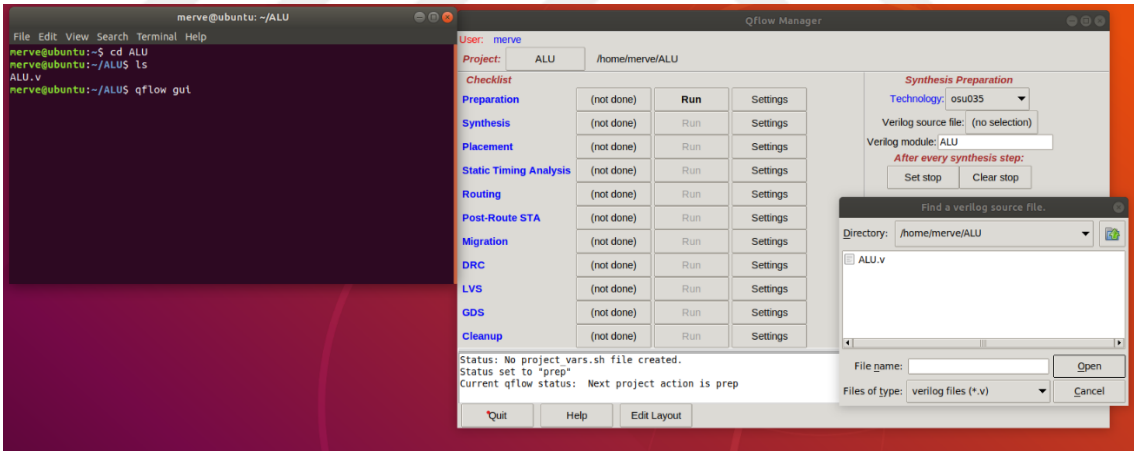
“cd “komutu ile terminalden hangi klasöre gidileceği seçilmiştir.

“ls” komutu ile klasör içerisinde dosyalar listelenmiştir. Klasör içerisinde görüldüğü üzere sentezlenmek için yerleştirilmiştir. .v uzantılı ALU dosyası yer almaktadır. İlgili görsel Şekil 3.12’de gösterilmiştir.



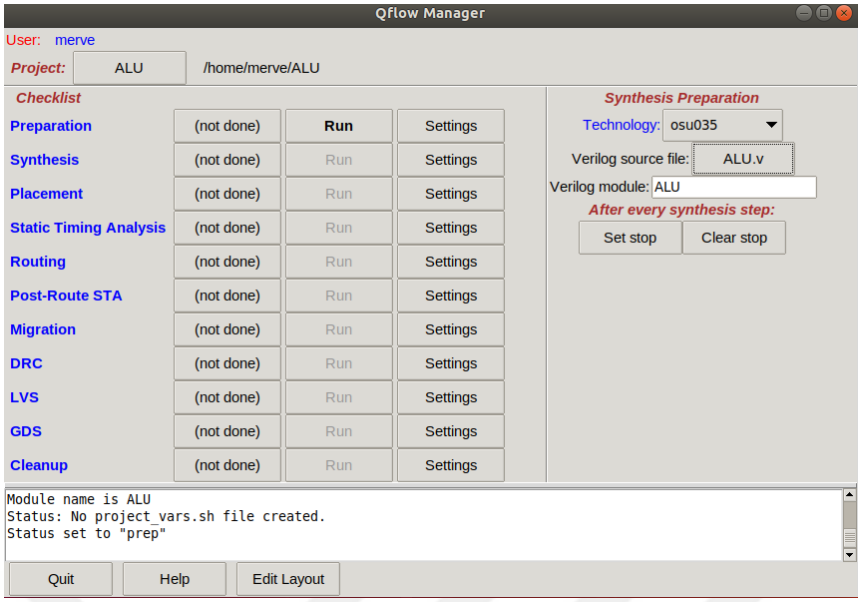
Şekil 3.12: Qflow ile sentezleme – 1.

Terminalde sentezlenecek dosya dizininde iken “qflow gui” komutu çalıştırılır ve gui penceresi otomatik açılmaktadır. Pencerede bir işlem yapılmadığı durumda aktif işlem adımı bulunmamaktadır.



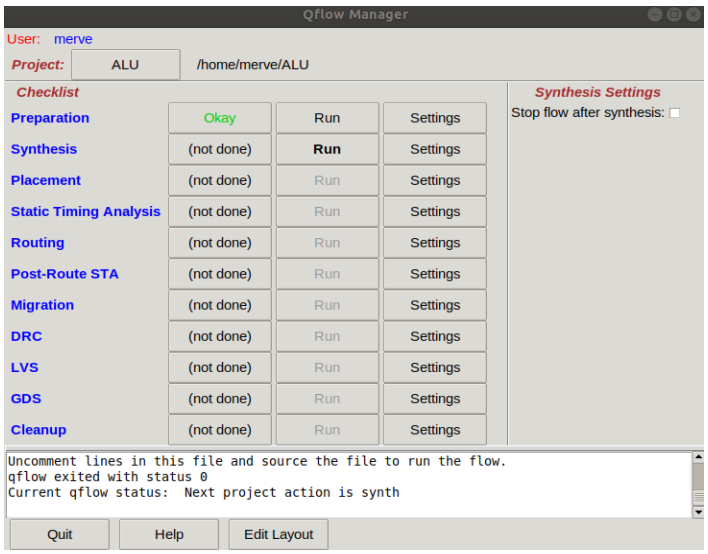
Şekil 3.13: Qflow ile sentezleme – 2.

Açılan gui penceresinde yapılması gereken ilk adım Şekil 3.13’te 3 numara ile isimlendirilen bölümde yer alan “Verilog Source File” kutucuğuna tıklanarak sentezlemesi yapılacak .v uzantılı dosya seçilmelidir. Aynı zamanda yine aynı bölümde yer alan “Technology” bölümünden istenilen transistör teknolojisi seçilebilmektedir. Açılış ayarlarında transistör teknoloji dosyası “osu35” olarak gelmektedir. Bu bölümde seçim işlemleri tamamlanmıştır. Artık Şekil 3.13’de 2 ile numaralandırılan bölümde sentezleme adımları sırası ile yapılacaktır.



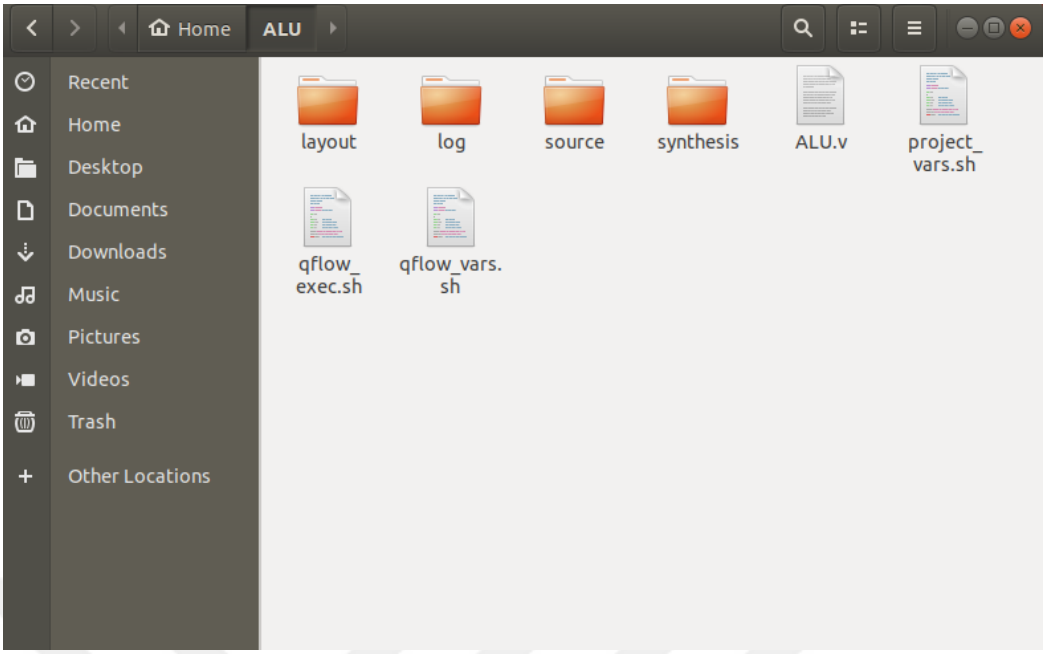
Şekil 3.14: Qflow ile sentezleme – 3.

2 ile numaralandırılan bölümdeki sentezleme adımlarına başlamadan önce görülen gui penceresi Şekil 3.14'da belirtilmiştir. Sentezleme adımları işlem satırlarının ortasında bulunan "Run" kutucuğuna basarak başlatılmaktadır. İşlem tamamlandığında satırda ilk durumu "Not done" olan kutucuk adım başarı ile "Okay" başarısız ise "Fail" uyarısı almaktadır. İlgili işlem adımına ait görsel Şekil 3.16 kapsamında gösterilmiştir.



Şekil 3.15: Qflow ile sentezleme – 4.

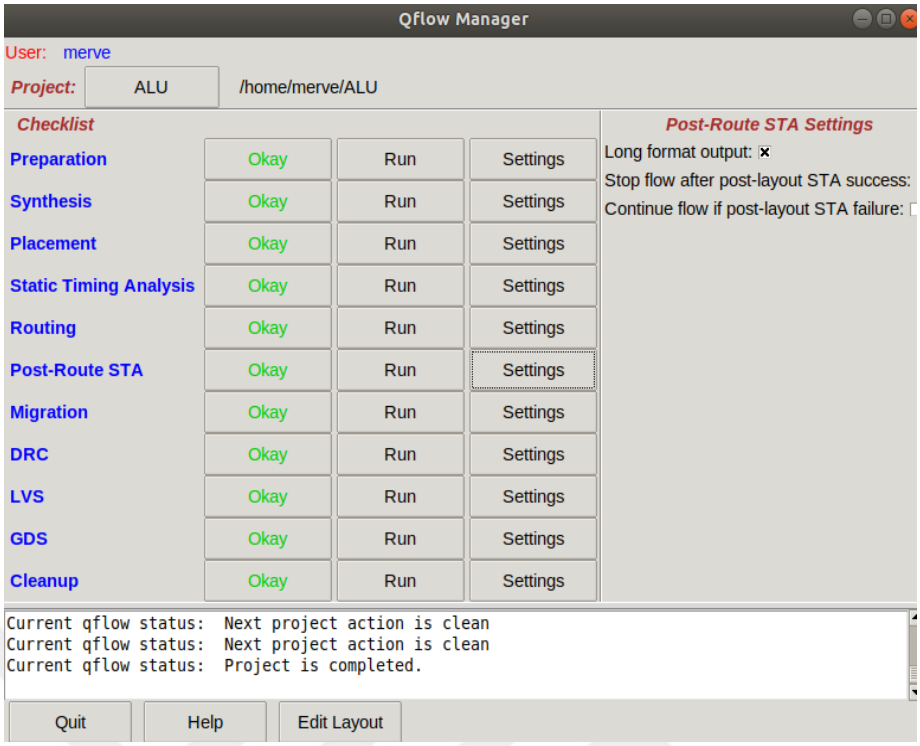
Aynı zamanda ilgili işlem adımının sentezleme sırasında bulunduğumuz klasör içerisinde Qflow log kayıtlarını klasörlere otomatik oluşturmaktadır. Buna dair ilgili görsel Şekil 3.16'de belirtilmiştir.



Şekil 3.16: Qflow ile sentezleme – 5.

İkinci adım olan "Synthesis" adımı "Run" kutucuğun basarak başlatılır. Tamamlandığında satırdaki ilk kutucuk "Okay" durumuna geçmesi adımın başarıyla tamamlandığını ifade etmektedir. Bu adımla birlikte eğer sentezlemede bir hata yok ise tüm adımlar sırası ile otomatik olarak devam etmektedir.

Sentezlemenin başarıyla tamamlandığını tüm işlem adımları "Okay" durumuna geçmesi ile anlaşılmaktadır. Herhangi bir adımda "Fail" alınması durumunda ilgili satırdaki "Fail" kutucuğuna basarak hata kaydına erişilebilmektedir. Başarılı bir sentezleme yapılabildiğini Şekil 3.17'da görülmektedir.



Şekil 3.17: Qflow ile sentezleme – 6.

3.4. IRSIM – SİMÜLASON

Açık kaynak kodlu olarak yayınlanan elektronik tasarım otomasyonu (EDA) başka bir araç IRSIM'dır. IRSIM aracı anahtarlama seviyesi simülasyon aracıdır. Böylece Transistörleri ideal anahtarlar olarak modellemektedir. Sentezlemede çıkarılan kapasite ve direnç değerleri ile göreceli bir zamanlama hesaplamak için RC zaman sabitini kullanmaktadır. Bu hesaplama ile anahtarları idealden ziyade daha gerçekçi modellenerek işlem yapılmaktadır [51].

Açık kaynak kodlu araçlar; Magic, qflow, Netgen, IRSIM, PCB, and XCircuit. Bu araçlar GNU Kamu Lisansı (GPL) veya benzer açık kaynak lisansı kapsamında ücretsiz olarak kullanılmaktadır. Ayrıca, ticari amaç ile kullanılan yazılımlar ile rekabet edebilecek işlevsellikte olmaya çalışmaktadırlar [52]. IRSIM aracı ile tasarımımızın fiziksel seviyedeki yerleşimini ve yollarının doğruluğunu üretim öncesinde simülasyon ile kontrolü yapılacaktır.

Dijital tasarım süreçlerinin uzun olması ve üretim maliyetlerinden dolayı üretim öncesinde simülasyon oldukça önem içermektedir. Tez kapsamında açık kaynak kodlu yazılım olan IRSIM aracı eklentisi ile fiziksel sentez sonrasında simülasyon gerçekleştirilmiştir.

3.4.1. Kurulum

Ubuntu işletim sisteminde açılan bir terminale yazılan kod dizileri ile IRSIM kurulumu gerçekleştirilmektedir.

```
>_
# masaüstü dizininde olunmalı (~)

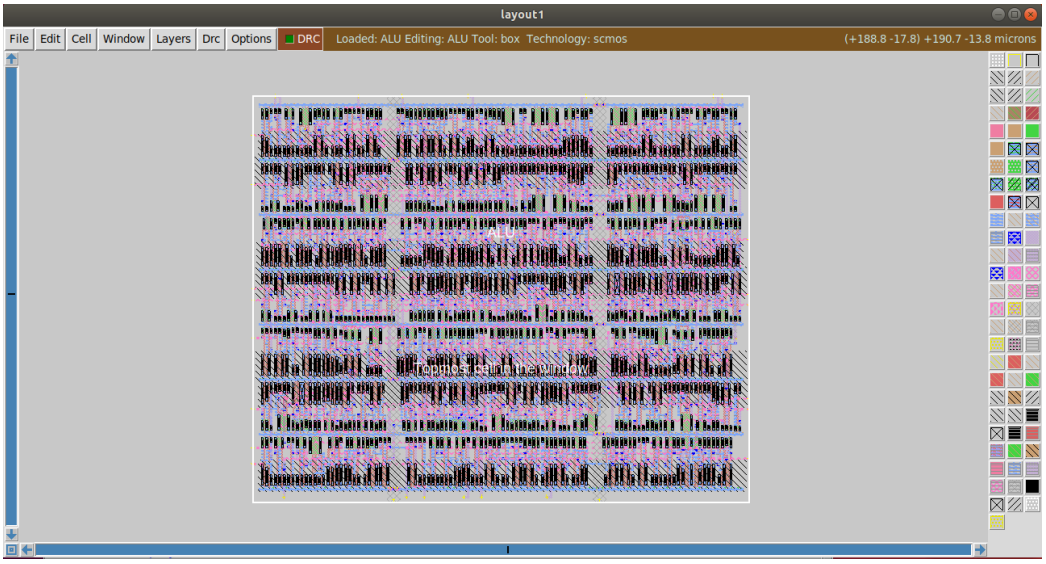
# depoaya kopyalama
git clone https://github.com/RTimothyEdwards/irsim.git
cd irsim

# versiyon bilgisin kontrolü
git checkout irsim-9.7 // irsim versiyon bilgisi sorgulama
./configure
make
sudo make install
```

3.4.2. Kullanım

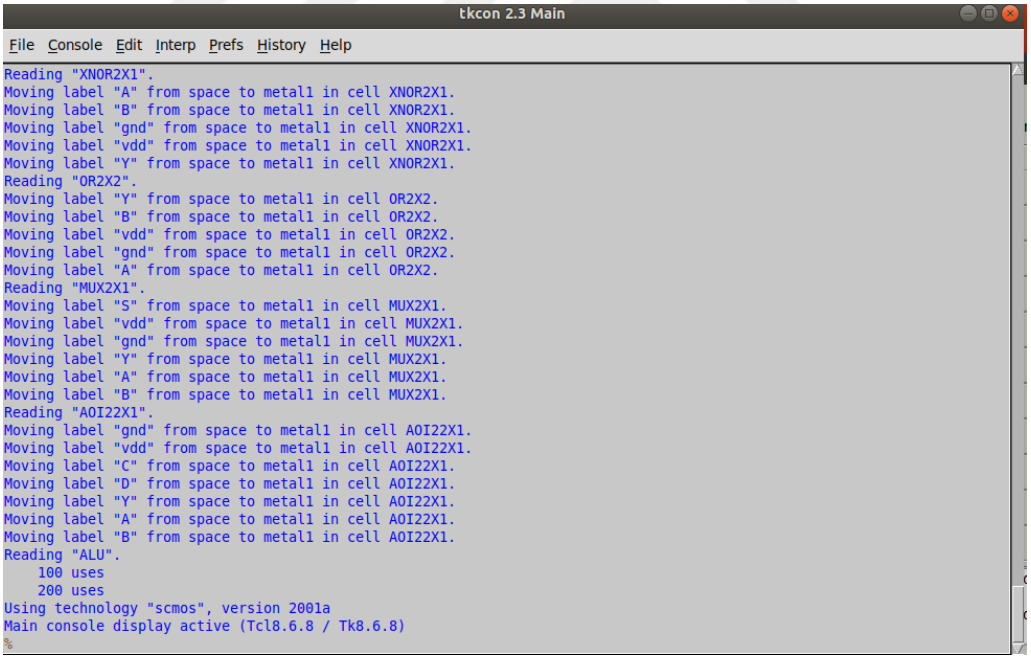
IRSIM ile simülasyon adımına geçilebilmesi için öncelikle seçilen transistör teknolojisindeki dosya ile Qflow sentezinin tüm adımlarının "Okay" alarak tamamlanması gerekmektedir.

IRSIM aracı Transistör seviyesideki yerleşim ile bağlantı kuracağından dolayı çalışma klasörü "Layout"tur. Çalışmaya başlamak için Şekil 3.11 üzerinde 4 ile numaralandırılan bölümde yer alan "Edit Layout" butonuna tıklanmalıdır. Açılan iki pencere bulunmaktadır. Bunlara ilki Şekil 3.18'da seçilen teknoloji dosyasına göre yerleşimi yapılmış halidir. Burada girişler, çıkışlar, GND ve VDD yapıları görülebilmektedir. Ayrıca yine tasarımı oluşturan standart hücre yapıları da yerleşimde yer almaktadır. Buradaki yerleşimde magic dosyasında yer alan bilgiler kullanılmaktadır.



Şekil 3.18: Qflow ile fiziksel sentezleme sonucu tasarım yerleşimi.

IRSIM ile simülasyon çalışması için açılan ve komutları yazacağımız pencere Şekil 3.19'de yer almaktadır.

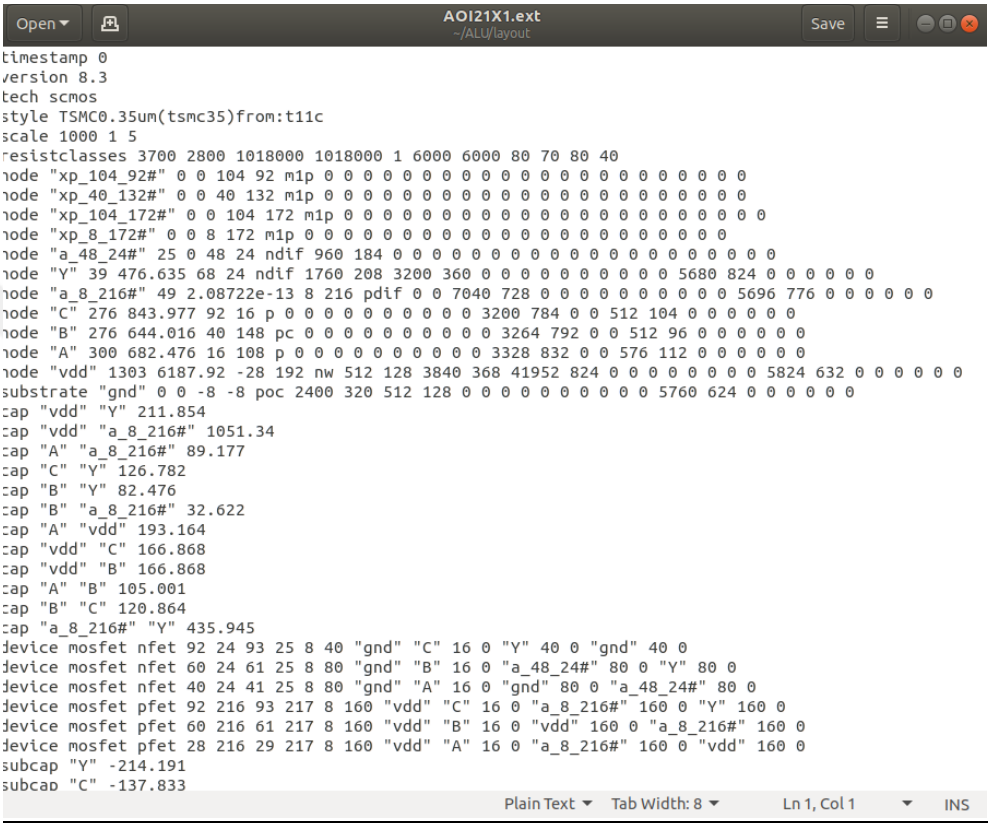


Şekil 3.19: tkcon 2.3 main sayfası.

Çalıştırılan ilk komut;

- >> extract all

Sentezlenme ile oluşan tüm alt modüller ve top modül "layout" klasörünün içerisine çıkartılır. Çıkartılan dosyalar .ext uzantılıdır. Bu dosyalar modüllerin yerleşim düzenindeki yerini belirtmektedir. Dosya içerisinde matris ifadesi yer almaktadır ve modülün vdd, gnd, input ve output ile olan ilişkisini göstermektedir. Örnek olması için .ext uzantılı dosya içeriği Şekil 3.20'de belirtilmiştir.



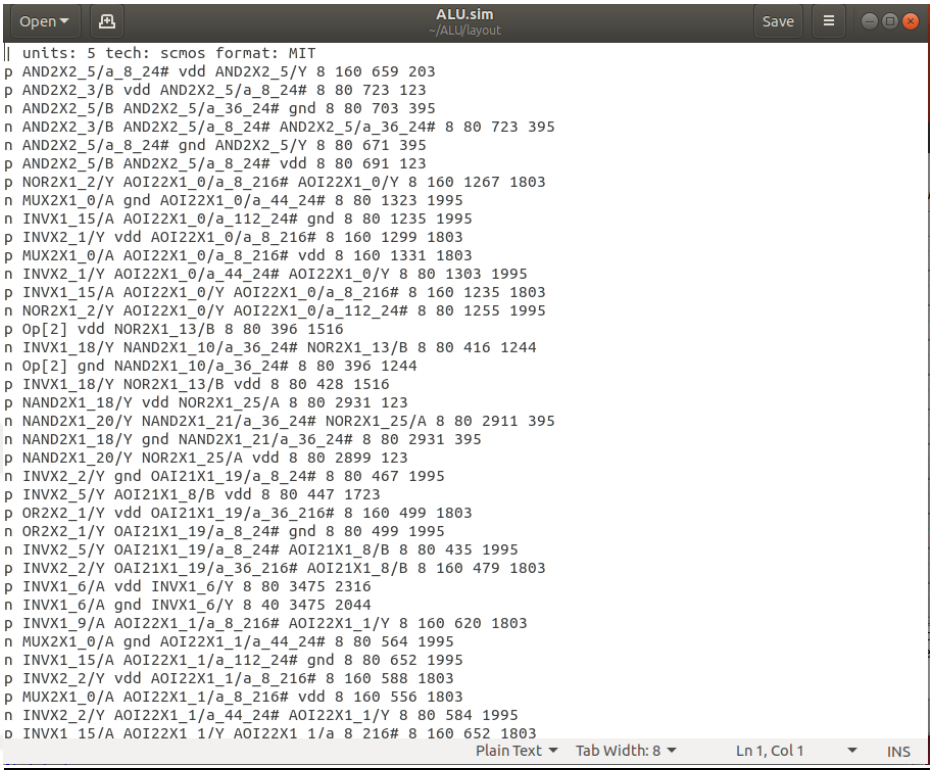
Şekil 3.20: .ext dosya içeriği.

Daha sonra çalıştırılacak komut;

- >> ext2sim ALU.ext

Simülasyon için ana modülün simülasyon için hazırlanması adımıdır. Komut çalıştırıldıktan sonra işlem adımının tamamlandığını "exttosim finished" uyarısı ile anlaşılmaktadır. "Layout" klasörü içerisinde ALU.sim uzantılı dosya oluşmuştur.

ALU.sim dosyası ile simülasyonu yapılacak tasarıma ait p ve n junction mosfetler ile yerleşimdeki bağlantılara referans verilerek bağlantıları oluşturulmuştur. .sim uzantılı dosya içeriği Şekil 3.21'te belirtilmiştir.



Şekil 3.21: ALU.sim uzantılı dosya içeriği.

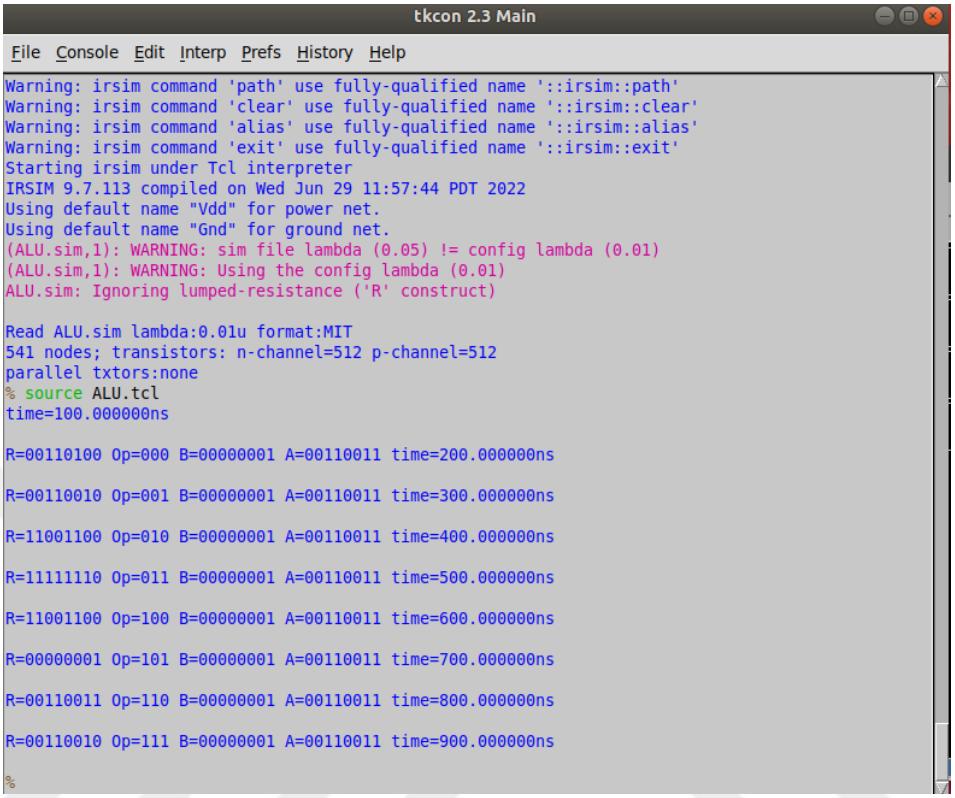
Bu adımdan sonra IRSIM ile simülasyonun çalıştırmak için "Layout" klasörüne sentezleme yaptığımız transistör teknoloji dosyasına ait .prm uzantılı dosyanın kopyalanması gerekmektedir. Teknoloji dosyasına ait parametreleri içeren .prm uzantılı dosya /usr/local/share/qflow/tech/osu35 uzantılı adrestedir. Kopyalama işlemi sonrasında simülasyonda çalıştırılacak test dosyası oluşturulmalıdır. Bu dosya .tcl uzantılı olmalıdır. Ardından IRSIM simülasyonu ile çalışma gerçekleştirilir.

Daha sonra çalıştırılacak komut;

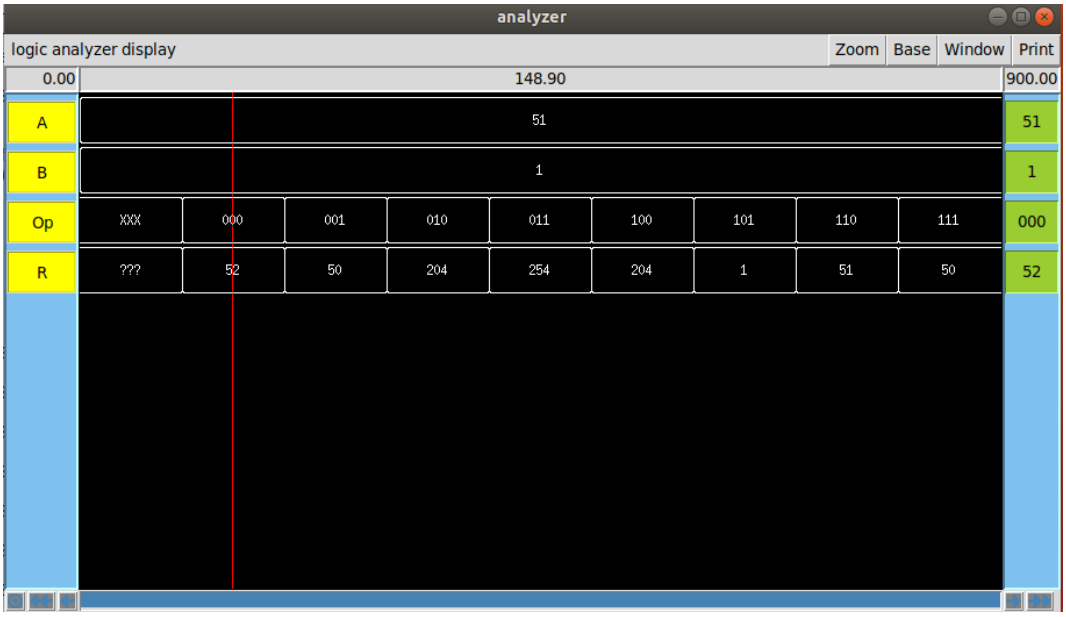
- >> irsim osu035.prm ALU.sim
- >> source ALU.tcl

Açılan pencerede set edilen giriş değerlerine göre tasarımın çıkarması gereken değerler görüntülenmektedir. Fiziksel olarak tasarım yapıldığında doğruluğu gözlemlenebilmektedir.

IRSIM simülasyon sonuçları Şekil 3.22 kapsamında sayışlar veriler üzerinden teyit edilebilir iken Şekil 3.23 özelinde de veriler çizdirilerek gösterim sağlanmaktadır.



Şekil 3.22: Komut sayfasında simülasyon çıktısı.



Şekil 3.23: IRSIM simülasyon analizi.

3.4.2.1. IRSIM Test Dosyasının Oluşturulması

Simülasyon adımında giriş değerlerine göre tasarımın çıkışlarının beklenen değerlere olup olmadığını değerlendirmek için kullanılacak test dosyasıdır. Dosya uzantısı .tcl'dir. Örnek tasarım 8 bitlik bir ALU tasarımına aittir. A ve B girişler OP işlem seçme için ve R de çıkış olarak ifade edilmektedir.Anahtar seviyesinde yapılacağından dolayı GND ve VDD değerleri Low ve High olarak tanımlanmıştır. Tasarımın girişleri ve çıkış bitleri vektör olarak tanımlanmaktadır. Başlangıç değeri olarak referans değerler atanmaktadır.

"ana" olarak ifade edilen adımda analiz edilecek girişler ve kayıt edilecek çıkış belirtilmiştir.

"s" bekleme süresini ifade etmektedir. Ardından OP'nın durumuna göre çıkıştaki işlemler "w" yazan satır ile sıralı şekilde yazdırılmakta ve görüntülenmektedir.

"vector" tanımlamaları ile girişlerin bitleri tanımlanmıştır.

"setvector" tanımlaması ile girişlere ilk değer atamaları yapılmıştır. Artından OP tanımlaması ile oluşacak tüm durumlar simülasyonu gerçekleştirilmiştir.

```
1  l gnd
2  h vdd
3  vector A {A[7]} {A[6]} {A[5]} {A[4]} {A[3]} {A[2]} {A[1]} {A[0]}
4  vector B {B[7]} {B[6]} {B[5]} {B[4]} {B[3]} {B[2]} {B[1]} {B[0]}
5  vector Op {Op[2]} {Op[1]} {Op[0]}
6  vector R {R[7]} {R[6]} {R[5]} {R[4]} {R[3]} {R[2]} {R[1]} {R[0]}
7
8  setvector A 00110011
9  setvector B 00000001
10 ana A B Op R
11 s
12 #durum 0
13 setvector Op 000
14 w A B Op R
15 s
16 #durum 1
17 setvector Op 001
18 w A B Op R
19 s
20 #durum 2
21 setvector Op 010
22 w A B Op R
23 s
24 #durum 3
25 setvector Op 011
26 w A B Op R
27 s
```

4. BULGULAR

Farklı transistör teknolojileri 180nm ve 350nm teknolojileridir. Farklı bit uzunluklarındaki ALU tasarımlarının sentezlenmesi gerçekleştirilmiştir. Farklı bit uzunluklarındaki ALU tasarımları 8 bit, 16 bit ve 32 bit uzunluklarında tasarımlar gerçekleştirilmiştir.

Lojik Tasarımlar Quartus II ortamında Verilog dilinde yapılmıştır. Lojik tasarımın simülasyonu için Modelsim Altera 10.1d kullanılmıştır. Lojik tasarım sonrasında fiziksel tasarım ortamı için açık kaynak kodlu Qflow1.4 yazılımı kullanılmıştır. Ayrıca fiziksel tasarımın doğrulunun teyit edilmesi için de IRSIM simülasyon ortamı ile tasarım doğruluğu teyit edilmiştir.

4.1. FARKLI BİT UZUNLUKLARINDA LOJİK TASARIM

4.1.1. 8 Bit ALU Lojik Tasarım

ALU tasarımı için Verilog kodu ile tasarım gerçekleştirilmiştir. Tasarımın sentezlenmesi ile RTL model oluşturulmuş ve tasarımın doğrulanması ModelSim ile gerçekleştirilmiştir.

8 bit ALU tasarımının lojik tasarım olarak yazılan kodu Şekil 4.1'de belirtilmiştir.

```
2  module ALU(
3      A,
4      B,
5      Op,
6      R   );
7
8      //inputs,outputs and internal variables declared here
9      input [7:0] A,B;
10     input [2:0] Op;
11     output [7:0] R;
12     wire [7:0] Reg1,Reg2;
13     reg [7:0] Reg3;
14
15     //Assign A and B to internal variables for doing operations
16     assign Reg1 = A;
17     assign Reg2 = B;
18     //Assign the output
19     assign R = Reg3;
20
21     //Always block with inputs in the sensitivity list.
22     always @(Op or Reg1 or Reg2)
23     begin
24         case (Op)
25          0 : Reg3 = Reg1 + Reg2;  //addition
26          1 : Reg3 = Reg1 - Reg2; //subtraction
27          2 : Reg3 = ~Reg1;  //NOT gate
28          3 : Reg3 = ~(Reg1 & Reg2); //NAND gate
29          4 : Reg3 = ~(Reg1 | Reg2); //NOR gate
30          5 : Reg3 = Reg1 & Reg2;  //AND gate
31          6 : Reg3 = Reg1 | Reg2;  //OR gate
32          7 : Reg3 = Reg1 ^ Reg2; //XOR gate
33         endcase
34     end
35 endmodule
```

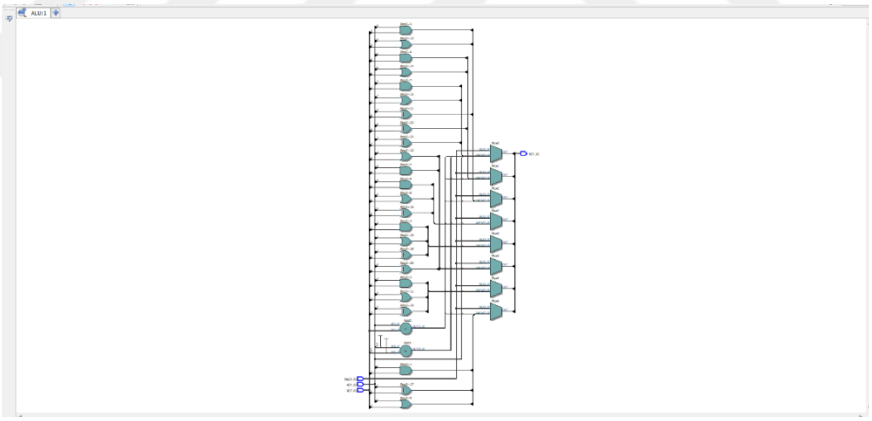
Şekil 4.1: 8 bit ALU tasarımı verilog kodu.

Kodun başarılı sentezlenmesi sonrasındaki çıktısı Şekil 4.2'de belirtilmiştir.

Fitter Summary	
Fitter Status	Successful - Wed Nov 23 21:38:12 2022
Quartus II 64-Bit Version	13.1.0 Build 162 10/23/2013 SJ Web Edition
Revision Name	ALU
Top-level Entity Name	ALU
Family	Cyclone IV GX
Device	EP4CGX15BF14C6
Timing Models	Final
Total logic elements	47 / 14,400 (< 1 %)
Total combinational functions	47 / 14,400 (< 1 %)
Dedicated logic registers	0 / 14,400 (0 %)
Total registers	0
Total pins	27 / 81 (33 %)
Total virtual pins	0
Total memory bits	0 / 552,960 (0 %)
Embedded Multiplier 9-bit elements	0
Total GXB Receiver Channel PCS	0 / 2 (0 %)
Total GXB Receiver Channel PMA	0 / 2 (0 %)
Total GXB Transmitter Channel PCS	0 / 2 (0 %)
Total GXB Transmitter Channel PMA	0 / 2 (0 %)
Total PLLs	0 / 3 (0 %)

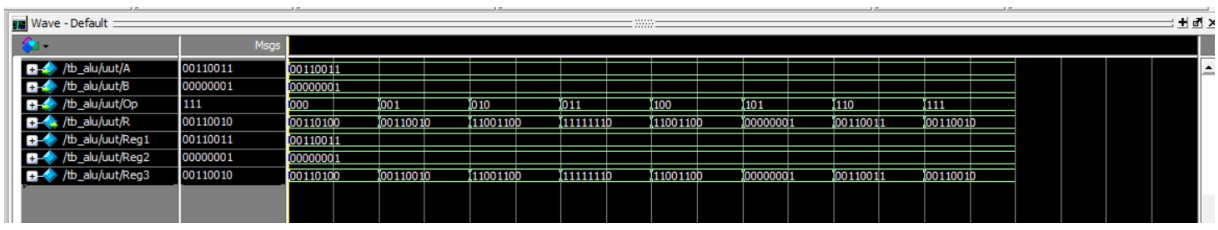
Şekil 4.2: 8 Bit ALU Quartus II sentez çıktısı.

8 bit ALU lojik tasarımına ait RTL modeli çıktısı Şekil 4.3 özelinde belirtilmiştir.



Şekil 4.3: 8 Bit ALU RTL şeması.

8 bit ALU lojik tasarımına ait ModelSim çıktısı Şekil 4.4'de belirtilmiştir.



Şekil 4.4: 8 Bit ALU ModelSim ile simülasyonu.

4.1.2. 16 Bit ALU Lojik Tasarım

16 bit ALU tasarımının lojik tasarım olarak yazılan kodu Şekil 4.5'de belirtilmiştir.

```
7  module ALU_16Bit(
8      A,
9      B,
10     Op,
11     R   );
12
13     //inputs,outputs and internal variables declared here
14     input [15:0] A,B;
15     input [2:0] Op;
16     output [15:0] R;
17     wire [15:0] Reg1,Reg2;
18     reg [15:0] Reg3;
19
20     //Assign A and B to internal variables for doing operations
21     assign Reg1 = A;
22     assign Reg2 = B;
23     //Assign the output
24     assign R = Reg3;
25
26     //Always block with inputs in the sensitivity list.
27     always @(Op or Reg1 or Reg2)
28     begin
29         case (Op)
30          0 : Reg3 = Reg1 + Reg2;  //addition
31          1 : Reg3 = Reg1 - Reg2; //subtraction
32          2 : Reg3 = ~Reg1;  //NOT gate
33          3 : Reg3 = ~(Reg1 & Reg2); //NAND gate
34          4 : Reg3 = ~(Reg1 | Reg2); //NOR gate
35          5 : Reg3 = Reg1 & Reg2;  //AND gate
36          6 : Reg3 = Reg1 | Reg2;  //OR gate
37          7 : Reg3 = Reg1 ^ Reg2; //XOR gate
38         endcase
39     end
40 endmodule
```

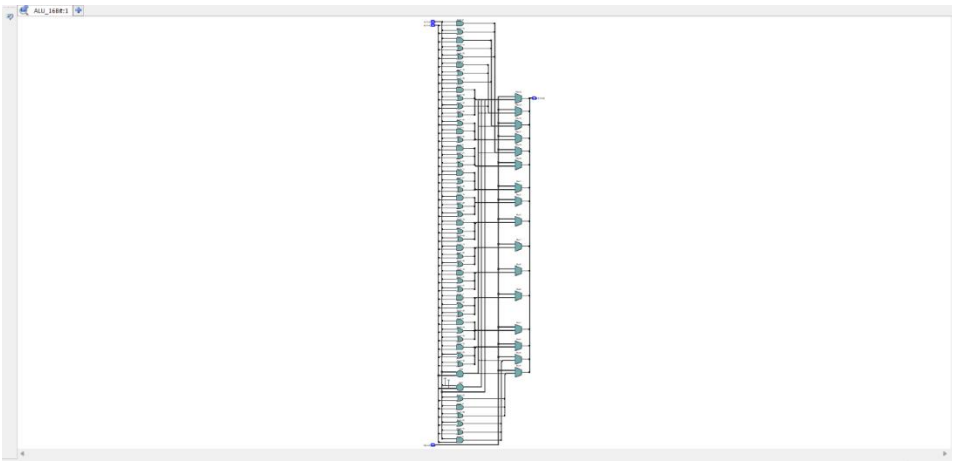
Şekil 4.5: 16 bit ALU tasarımı verilog kodu.

Kodun başarılı sentezlenmesi sonrasındaki çıktısı Şekil 4.6'de belirtilmiştir.

Fitter Summary	
Fitter Status	Successful - Wed Nov 23 21:46:08 2022
Quartus II 64-Bit Version	13.1.0 Build 162 10/23/2013 SJ Web Edition
Revision Name	ALU_16Bit
Top-level Entity Name	ALU_16Bit
Family	Cyclone IV GX
Device	EP4CGX22CF19C6
Timing Models	Final
Total logic elements	91 / 21,280 (< 1 %)
Total combinational functions	91 / 21,280 (< 1 %)
Dedicated logic registers	0 / 21,280 (0 %)
Total registers	0
Total pins	51 / 167 (31 %)
Total virtual pins	0
Total memory bits	0 / 774,144 (0 %)
Embedded Multiplier 9-bit elements	0 / 80 (0 %)
Total GXB Receiver Channel PCS	0 / 4 (0 %)
Total GXB Receiver Channel PMA	0 / 4 (0 %)
Total GXB Transmitter Channel PCS	0 / 4 (0 %)
Total GXB Transmitter Channel PMA	0 / 4 (0 %)
Total PLLs	0 / 4 (0 %)

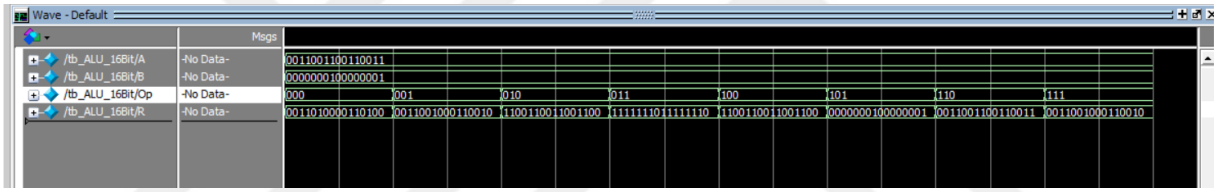
Şekil 4.6: 16 bit ALU Quartus II sentez çıktısı.

16 bit ALU lojik tasarımına ait RTL modeli çıktısı Şekil 4.7 özelinde belirtilmiştir.



Şekil 4.7: 16 bit ALU RTL şeması.

16 bit ALU lojik tasarımına ait ModelSim çıktısı Şekil 4.8'de belirtilmiştir.



Şekil 4.8: 16 bit ALU ModelSim ile simülasyonu.

4.1.3. 32 Bit ALU Lojik Tasarım

32 bit ALU tasarımının lojik tasarım olarak yazılan kodu Şekil 4.9'de belirtilmiştir.

```
7  module ALU_32Bit(
8      A,
9      B,
10     Op,
11     R   );
12
13     //inputs,outputs and internal variables declared here
14     input [31:0] A,B;
15     input [2:0] Op;
16     output [31:0] R;
17     wire [31:0] Reg1,Reg2;
18     reg [31:0] Reg3;
19
20     //Assign A and B to internal variables for doing operations
21     assign Reg1 = A;
22     assign Reg2 = B;
23     //Assign the output
24     assign R = Reg3;
25
26     //Always block with inputs in the sensitivity list.
27     always @(Op or Reg1 or Reg2)
28     begin
29         case (Op)
30           0 : Reg3 = Reg1 + Reg2;  //addition
31           1 : Reg3 = Reg1 - Reg2; //subtraction
32           2 : Reg3 = ~Reg1;  //NOT gate
33           3 : Reg3 = ~(Reg1 & Reg2); //NAND gate
34           4 : Reg3 = ~(Reg1 | Reg2); //NOR gate
35           5 : Reg3 = Reg1 & Reg2;  //AND gate
36           6 : Reg3 = Reg1 | Reg2;  //OR gate
37           7 : Reg3 = Reg1 ^ Reg2; //XOR gate
38         endcase
39     end
40 endmodule
```

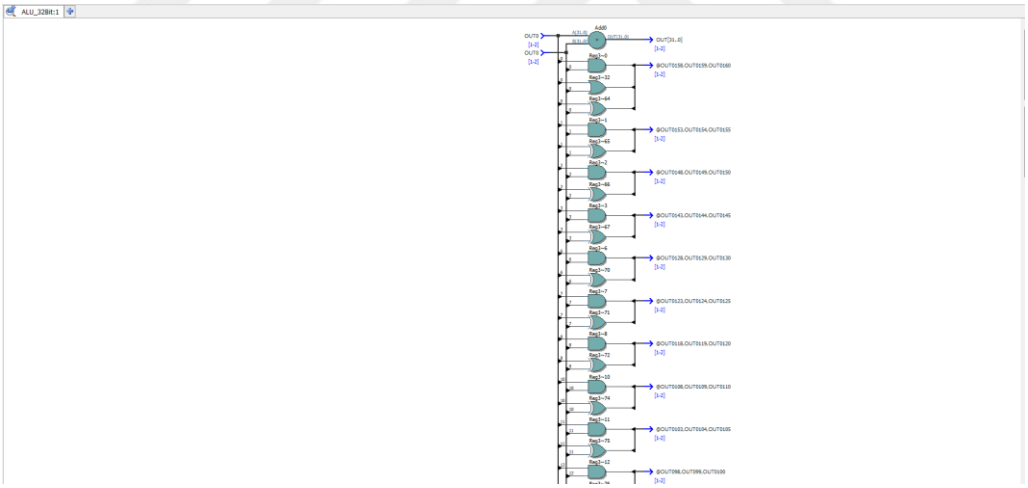
Şekil 4.9: 32 bit ALU tasarımı verilog kodu.

Kodun başarılı sentezlenmesi sonrasındaki çıktısı Şekil 4.10'de belirtilmiştir.

Fitter Summary	
Fitter Status	Successful - Wed Nov 23 21:48:26 2022
Quartus II 64-Bit Version	13.1.0 Build 162 10/23/2013 SJ Web Edition
Revision Name	ALU_32Bit
Top-level Entity Name	ALU_32Bit
Family	Cyclone IV GX
Device	EP4CGX22CF19C6
Timing Models	Final
Total logic elements	179 / 21,280 (< 1 %)
Total combinational functions	179 / 21,280 (< 1 %)
Dedicated logic registers	0 / 21,280 (0 %)
Total registers	0
Total pins	99 / 167 (59 %)
Total virtual pins	0
Total memory bits	0 / 774,144 (0 %)
Embedded Multiplier 9-bit elements	0 / 80 (0 %)
Total GXB Receiver Channel PCS	0 / 4 (0 %)
Total GXB Receiver Channel PMA	0 / 4 (0 %)
Total GXB Transmitter Channel PCS	0 / 4 (0 %)
Total GXB Transmitter Channel PMA	0 / 4 (0 %)
Total PLLs	0 / 4 (0 %)

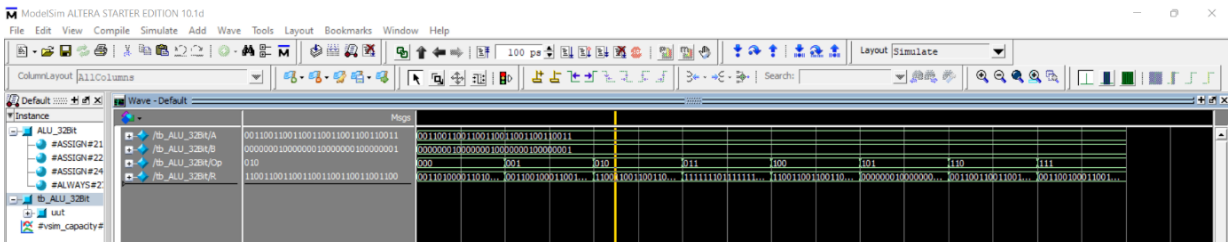
Şekil 4.10: 32 bit ALU Quartus II sentez çıktısı.

32 bit ALU lojik tasarımına ait RTL modeli çıktısından bir kesit Şekil 4.11 özelinde belirtilmiştir.



Şekil 4.11: 32 bit ALU RTL şeması kesiti.

32 bit ALU lojik tasarımına ait ModelSim çıktısı Şekil 4.12'de belirtilmiştir.

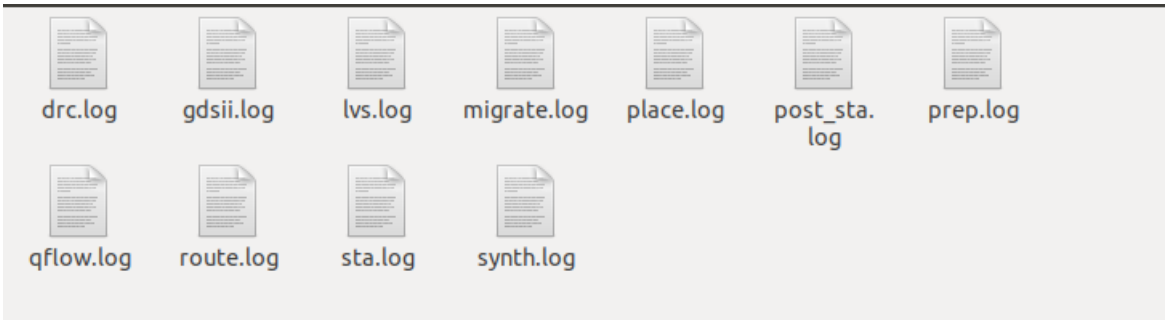


Şekil 4.12: 32 bit ALU ModelSim ile simülasyonu.

4.2. FARKLI TEKNOLOJİ DOSYALARI İLE FİZİKSEL SENTEZLEME

Fiziksel tasarımı Verilog dilinde yapılan tasarımların .v uzantılı dosya formatı oluşturmaktadır. Bu dosya açık kaynak kodlu Qflow aracı ile sentezlenmiştir. Sentezleme işlemi sonrasında IRSIM aracı ile de yine simülasyon işlemi yapılmıştır.

Qflow sentezlemesi sonucunda oluşan log dosyaları bulunmaktadır. Farklı bit uzunluklarındaki ALU tasarımlarında aynı teknoloji ile sentezlenmesindeki etkiler gözlemlenebilecektir. Sentezlenmesi sonrasında oluşan log klasöründe yer alan log dosyaları Şekil 4.13'te gözlemlenmektedir. synth.log, sta.log, place.log dosyalarındaki bilgiler gözlemlenmiştir.

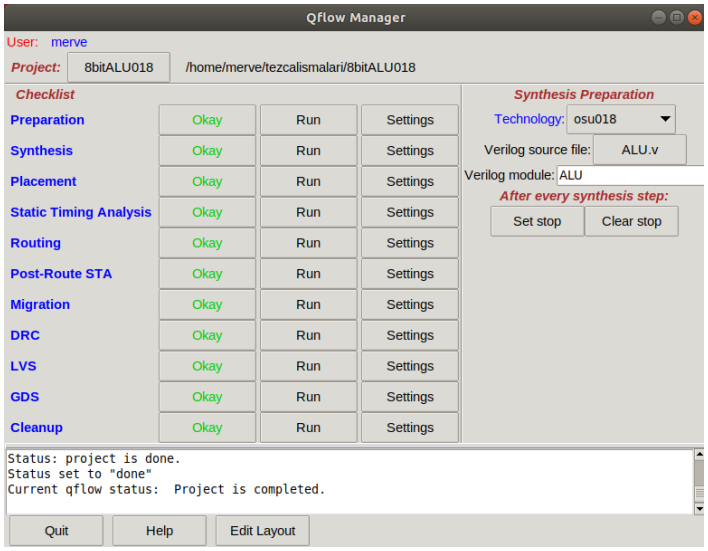


Şekil 4.13: Qflow sentezlemesi ile oluşan Log kayıtları.

4.2.1. 180nm Teknoloji ile Sentezleme

4.2.1.1. 8 Bit ALU Sentezlenmesi

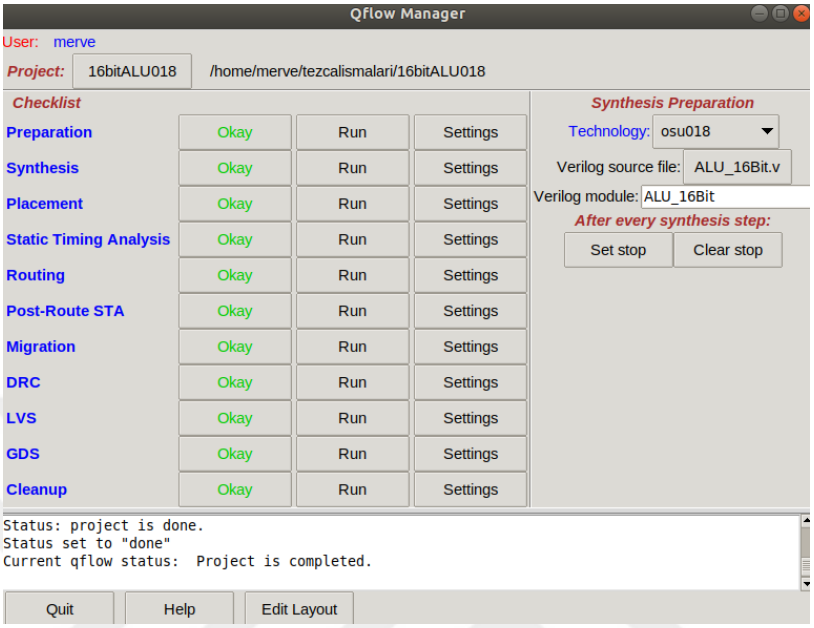
180nm transistör teknolojisindeki fiziksel sentezleme, 8 bit ALU tasarımının Qflow aracında osu018 seçilerek başarılı sentezlemesine ait görsel Şekil 4.14'de görülmektedir.



Şekil 4.14: 8 bit ALU tasarımı 180nm Qflow sentezlemesi.

4.2.1.2. 16 Bit ALU Sentezlenmesi

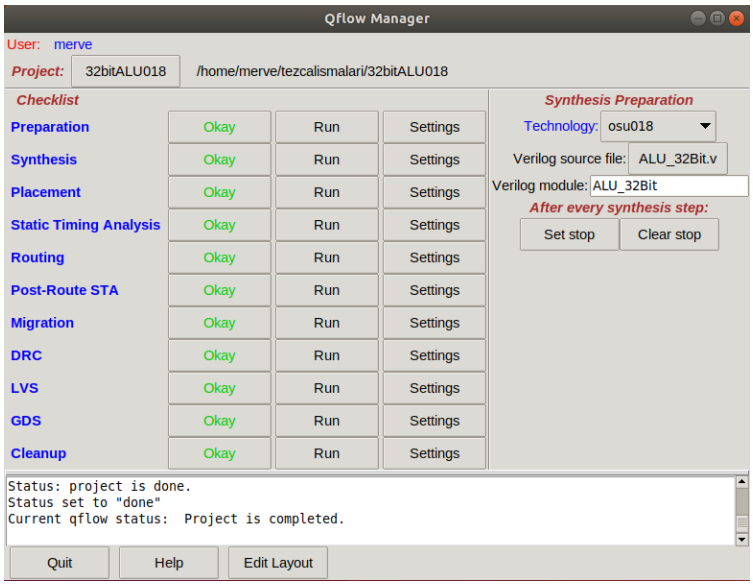
180nm transistör teknolojisindeki fiziksel sentezleme, 16 bit ALU tasarımının Qflow aracında osu018 seçilerek başarılı sentezlemesine ait görsel Şekil 4.15'de görülmektedir.



Şekil 4.15: 16 bit ALU tasarımı 180nm Qflow sentezlemesi.

4.2.1.3. 32 Bit ALU Sentezlenmesi

180nm transistör teknolojisindeki fiziksel sentezleme, 32 bit ALU tasarımının Qflow aracında osu018 seçilerek başarılı sentezlemesine ait görsel Şekil 4.16'de görülmektedir.

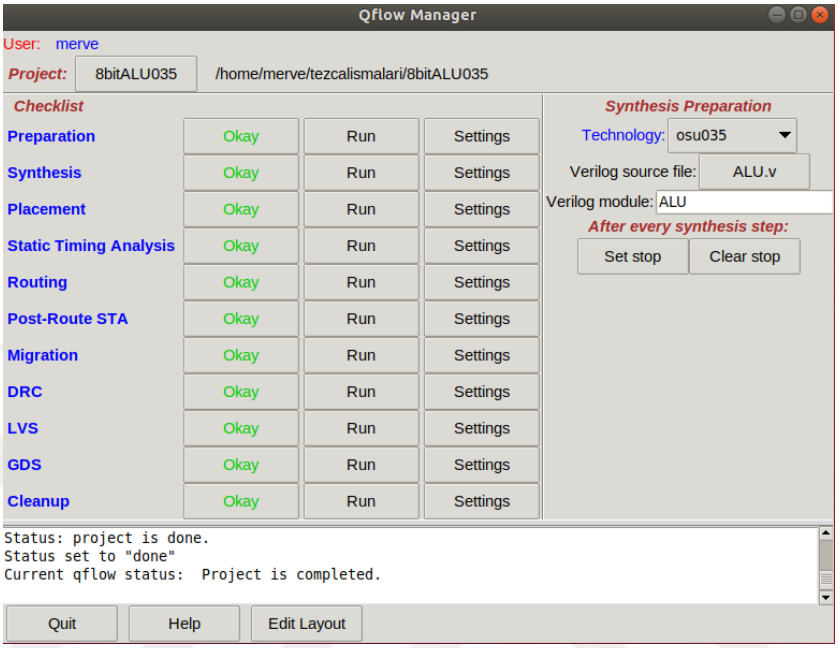


Şekil 4.16: 32 bit ALU tasarımı 180nm Qflow sentezlemesi.

4.2.2. 350nm Teknoloji İle Sentezleme

4.2.2.1. 8 Bit ALU Sentezlenmesi

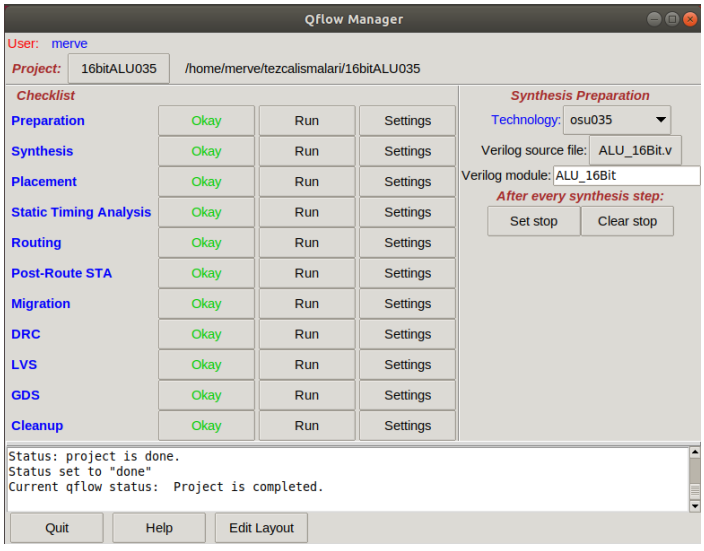
350nm transistör teknolojisindeki fiziksel sentezleme, 8 bit ALU tasarımının Qflow aracında osu018 seçilerek başarılı sentezlemesine ait görsel Şekil 4.17'de görülmektedir.



Şekil 4.17: 8 bit ALU tasarımı 350nm Qflow sentezlemesi.

4.2.2.2. 16 Bit ALU Sentezlenmesi

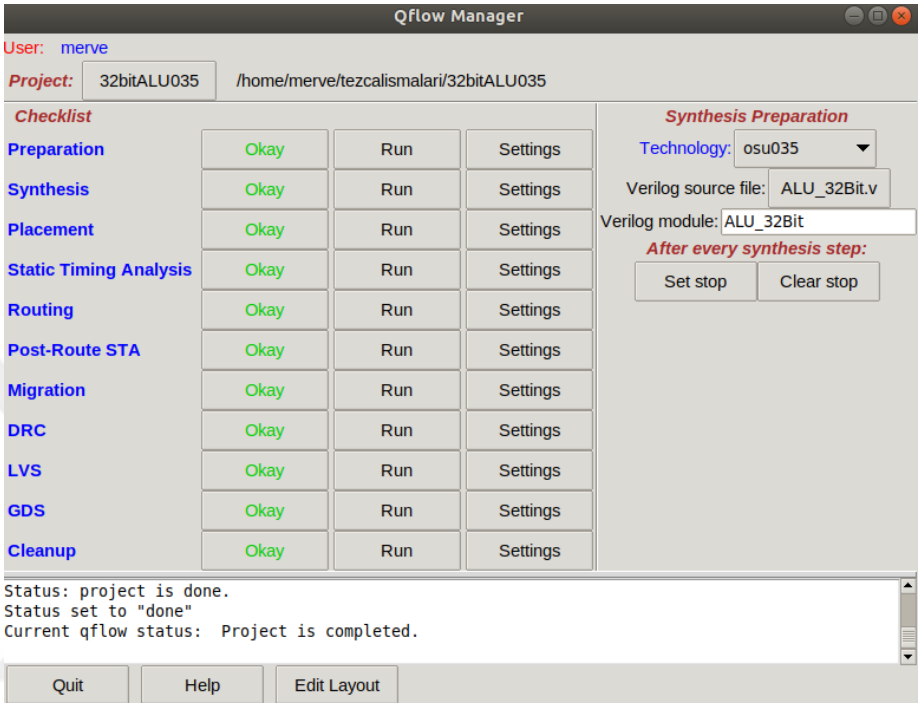
350nm transistör teknolojisindeki fiziksel sentezleme, 16 bit ALU tasarımının Qflow aracında osu018 seçilerek başarılı sentezlemesine ait görsel Şekil 4.14'de görülmektedir.



Şekil 4.18: 16 bit ALU tasarımı 350nm Qflow sentezlemesi.

4.2.2.3. 32 Bit ALU Sentezlenmesi

350nm transistör teknolojisindeki fiziksel sentezleme, 32 bit ALU tasarımının Qflow aracında osu018 seçilerek başarılı sentezlemesine ait görsel Şekil 4.19'de görülmektedir.



Şekil 4.19: 32 bit ALU tasarımı 350nm Qflow sentezlemesi.

180nm ve 350nm teknolojilerinde yapılan Qflow sentezlemesi sonucunda IRSIM için testbenchler oluşturulmuş ve simülasyonlar ile tasarımlar doğrulanmıştır. Oluşan log klasöründeki yerleşime ait veriler farklı teknoloji dosyaları için Tablo 4.1 yer almaktadır. Oluşan log klasöründeki statik time analiz (sta) ait ait veriler farklı teknoloji dosyaları için Tablo 4.2 yer almaktadır. Oluşan log klasöründeki yerleşime ait veriler farklı teknoloji dosyaları için Tablo 4.3 ve Tablo 4.4 yer almaktadır.

Tablo 4.1: 180 - 350 nm teknolojisindeki place.log verileri.

	8 Bit ALU	**16 Bit ALU**	**32 Bit ALU**
180 nm	Total stdcells:211	Total stdcells:456	Total stdcells:970
	Total cell width:5.89e+04	Total cell width:1.29e+05	Total cell width:2.74e+05
	Total cell height:2.11e+05	Total cell height:4.56e+05	Total cell height:9.70e+05
	Total cell area:5.89e+07	Total cell area:1.29e+08	Total cell area:2.74e+08
	Total core area:5.89e+07	Total core area:1.29e+08	Total core area:2.74e+08
	Average cell height:1.00e+03	Average cell height:1.00e+03	Average cell height:1.00e+03
350 nm	Total stdcells :210	Total stdcells :437	Total stdcells :973
	Total cell width :1.16e+05	Total cell width :2.45e+05	Total cell width :5.36e+05
	Total cell height :4.20e+05	Total cell height :8.74e+05	Total cell height :1.95e+06
	Total cell area :2.31e+08	Total cell area :4.89e+08	Total cell area :1.07e+09
	Total core area :2.31e+08	Total core area :4.89e+08	Total core area :1.07e+09
	Average cell height:2.00e+03	Average cell height:2.00e+03	Average cell height:2.00e+03

Tablo 4.2 : 180 - 350 nm teknolojisindeki sta.log verileri.

	8 Bit ALU	**16 Bit ALU**	**32 Bit ALU**
180 nm	Top 8 maximum delay paths:1114.15 ps	Top 16 maximum delay paths:1482.58 ps	Top 20 maximum delay paths: 1769.47 ps
	Top 8 minimum delay paths:187.453 ps	Top 16 minimum delay paths:187.453 ps	Top 20 minimum delay paths:187.453 ps
350 nm	Top 8 maximum delay paths:1925.81 ps	Top 16 maximum delay paths:2451.97 ps	Top 20 maximum delay paths:3041.15 ps
	Top 8 minimum delay paths:264.201 ps	Top 16 minimum delay paths:264.201 ps	Top 20 minimum delay paths:264.201 ps

Tablo 4.3: 180 nm teknolojisindeki synth.log verileri.

	8 Bit ALU		**16 Bit ALU**		**32 Bit ALU**	
180 nm	Number of wires	200	Number of wires	429	Number of wires	876
	Number of wire bits	230	Number of wire bits	491	Number of wire bits	1002
	Number of public wires	200	Number of public wires	429	Number of public wires	876
	Number of public wire bits	230	Number of public wire bits	491	Number of public wire bits	1002
	Number of memories	0	Number of memories	0	Number of memories	0
	Number of memory bits	0	Number of memory bits	0	Number of memory bits	0
	Number of processes	0	Number of processes	0	Number of processes	0
	Number of cells	211	Number of cells	456	Number of cells	935
	AND2X2	5	AND2X2	18	AND2X2	30
	AOI21X1	18	AOI21X1	38	AOI21X1	82
	AOI22X1	6	AOI22X1	14	AOI22X1	22
	BUFX2	8	BUFX2	16	BUFX2	32
	INVX1	37	INVX1	71	INVX1	154

	8 Bit ALU		16 Bit ALU		32 Bit ALU	
	MUX2X1	2	MUX2X1	2	MUX2X1	6
	NAND2X1	28	NAND2X1	59	NAND2X1	102
	NAND3X1	15	NAND3X1	26	NAND3X1	51
	NOR2X1	34	NOR2X1	68	NOR2X1	152
	OAI21X1	42	NOR3X1	1	OAI21X1	257
	OAI22X1	3	OAI21X1	125	OAI22X1	4
	OR2X2	6	OAI22X1	3	OR2X2	17
	XNOR2X1	4	OR2X2	4	XNOR2X1	12
	XOR2X1	2	XNOR2X1	6	XOR2X1	14
			XOR2X1	5		

Tablo 4.4: 350 nm Teknolojisindeki synth.log verileri.

	8 Bit ALU		16 Bit ALU		32 Bit ALU	
350 nm	Number of wires	199	Number of wires	410	Number of wires	880
	Number of wire bits	229	Number of wire bits	472	Number of wire bits	1006
	Number of public wires	199	Number of public wires	410	Number of public wires	880
	Number of public wire bits	229	Number of public wire bits	472	Number of public wire bits	1006
	Number of memories	0	Number of memories	0	Number of memories	0
	Number of memory bits	0	Number of memory bits	0	Number of memory bits	0
	Number of processes	0	Number of processes	0	Number of processes	0
	Number of cells	210	Number of cells	437	Number of cells	939
	AND2X2	6	AND2X2	19	AND2X2	31
	AOI21X1	20	AOI21X1	44	AOI21X1	104
	AOI22X1	5	AOI22X1	15	AOI22X1	27
	BUFX2	8	BUFX2	16	BUFX2	32
	INVX1	36	INVX1	64	INVX1	159
	MUX2X1	3	MUX2X1	1	MUX2X1	5
	NAND2X1	30	NAND2X1	50	NAND2X1	120
	NAND3X1	23	NAND3X1	31	NAND3X1	52
	NOR2X1	28	NOR2X1	69	NOR2X1	145
	OAI21X1	44	NOR3X1	1	OAI21X1	231
	OAI22X1	3	OAI21X1	103	OAI22X1	14
	OR2X2	2	OAI22X1	9	OR2X2	13
	XNOR2X1	2	OR2X2	11	XNOR2X1	6
			XNOR2X1	3		
			XOR2X1	1		

5. TARTIŞMA

Günümüzde çip teknolojisine olan ihtiyacın artması ile bu alanda yetişmiş insan gücü ihtiyacı da artmaktadır. Ülkemizde bu alan ile ilgilenmek isteyen bireylerin farkındalığını arttırmak ve aynı zamanda konu üzerine çalışmalarını desteklemek üzere kamu destekli yarışmalar düzenlenmektedir.

Bu durumun yanı sıra VLSI tasarım ve sentezleme alanlarında kullanılan araçların lisanslı olması sebebi ile kullanıcılar erişim sorunu yaşanmaktadır. Kullanıcıların yaşadığı bu sorunların aşılabilmesi için açık kaynak kodlu yazılımlar geliştirilerek VLSI sentezleme çalışmaları yaygınlaştırılmaktadır.

Tez kapsamında lojik tasarım aracı olarak kullanılabilen ve ücretsiz sürüm olan Quartus II ile birlikte lojik sentezleme aşamalarına ayrıntılı olarak yer verilmektedir. Ardından fiziksel sentez için açık kaynak kodlu yazılım ile sentezlemeler yapılarak bu aşamadaki işlem adımlarından da ayrıntılı olarak bahsedilmektedir.

Mikroişlemcilerde, boyutları gelişen Transistör teknolojisi ile işlem yapabilme kabiliyetinde artış yaşanırken kullanılan birim alanda da azalma yaşanması beklenmektedir. Bu çalışma Transistör boyutlarında küçülmenin lojik tasarım süreçlerindeki etkisini ve farklı bit uzunluklarındaki tasarımlarda, transistör boyutunun etkisini gözlemleyebilmek amacıyla gerçekleştirilmiştir. Farklı bit uzunluklarındaki ALU tasarımı, 180 nm ve 350 nm boyutlarındaki transistör teknolojileri ile sentezlenerek kullanılan standart hücreleri, tasarımın kapladığı alan ile tasarımın sahip olduğu maximum ve minimum gecikme süreleri değerlendirilmiştir.

8 bit ALU tasarımının 180 nm transistör teknolojisine göre fiziksel sentezinin yapılması sonucunda standart hücre sayısı 211, hücrelerin genişliği 5.89e+04 µm, hücrelerin yüksekliği 2.11e+05 µm, tasarımın kapladığı alan 5.89e+07 µm² ve ortalama hücre yüksekliği 1.00e+03 µm olarak hesaplanmıştır. Ayrıca bu tasarımın sahip olduğu maximum geçişme süresi 1114.15 ps ve minimum gecikme süresi 187.453 ps olarak hesaplanmıştır.

16 bit ALU tasarımının 180 nm transistör teknolojisine göre fiziksel sentezinin yapılması sonucunda standart hücre sayısı 456, hücrelerin genişliği 1.29e+05 µm, hücrelerin yüksekliği 4.56e+05 µm, tasarımın kapladığı alan 1.29e+08 µm² ve ortalama hücre yüksekliği 1.00e+03 µm olarak hesaplanmıştır. Ayrıca bu tasarımın sahip olduğu maximum geçişme süresi 1482.58 ps ve minimum gecikme süresi 187.453 ps olarak hesaplanmıştır.

32 bit ALU tasarımının 180 nm transistör teknolojisine göre fiziksel sentezinin yapılması sonucunda standart hücre sayısı 970, hücrelerin genişliği 2.74e+05 µm, hücrelerin yüksekliği 9.70e+05 µm, tasarımın kapladığı alan 2.74e+08 µm² ve ortalama hücre yüksekliği 1.00e+03 µm olarak hesaplanmıştır. Ayrıca bu tasarımın sahip olduğu maximum geçişme süresi 1769.47 ps ve minimum gecikme süresi 187.453 ps olarak hesaplanmıştır.

8 bit ALU tasarımının 350 nm transistör teknolojisine göre fiziksel sentezinin yapılması sonucunda standart hücre sayısı 210, hücrelerin genişliği 1.16e+05 µm, hücrelerin yüksekliği 4.20e+05 µm, tasarımın kapladığı alan 2.31e+08 µm² ve ortalama hücre yüksekliği 2.00e+03 µm olarak hesaplanmıştır. Ayrıca bu tasarımın sahip olduğu maximum geçişme süresi 1925.81 ps ve minimum gecikme süresi 267.201 ps olarak hesaplanmıştır

16 bit ALU tasarımının 350 nm transistör teknolojisine göre fiziksel sentezinin yapılması sonucunda standart hücre sayısı 437, hücrelerin genişliği 2.45e+05 µm, hücrelerin yüksekliği 8.74e+05 µm, tasarımın kapladığı alan 4.89e+08 µm² ve ortalama hücre yüksekliği 2.00e+03 µm olarak hesaplanmıştır. Ayrıca bu tasarımın sahip olduğu maximum geçişme süresi 2451.97 ps ve minimum gecikme süresi 267.201 ps olarak hesaplanmıştır

32 bit ALU tasarımının 350 nm transistör teknolojisine göre fiziksel sentezinin yapılması sonucunda standart hücre sayısı 973, hücrelerin genişliği 5.36e+05 µm, hücrelerin yüksekliği 1.95e+06 µm, tasarımın kapladığı alan 1.07e+09 µm² ve ortalama hücre yüksekliği 2.00e+03 µm olarak hesaplanmıştır. Ayrıca bu tasarımın sahip olduğu maximum geçişme süresi 3041.15 ps ve minimum gecikme süresi 267.201 ps olarak hesaplanmıştır.

Dijital tasarımın ürüne dönüşmesi için gereken zaman ve maliyet fazla olduğundan üretim öncesinde hataları minimize etmek için simülasyonlar yapılmaktadır. Bu çalışmada lojik tasarım ve fiziksel tasarım sonrasında simülatörler ile tasarımın doğruluğu kontrol edilmiştir. ModelSim ve Irsim sonuçları birbirleri ile örtüşmektedir. Transistör teknolojisinin değişmesi ya da tasarımdaki bit uzunluğunun değişmesi simülasyonlarda bir hata oluşturmamıştır. 8 bit

ALU tasarımının fiziksel sentezleme sonrasında yapılan ModelSim çalışması ile 180nm transistör teknolojisi ile sentezlenmesi sonrasında yapılan Irsim çalışması aynı çıktıyı verdiği gözlemlenmiştir. Gerçekleştirilen tasarımın doğruluğu lojik tasarımda yapılan simülasyon ve fiziksel tasarımda yapılan simülasyon karşılaştırılarak teyit edilmiştir. Çalışma bu yönüyle alana katkı sunmaktadır.

6. SONUÇ VE ÖNERİLER

Çalışmaya konu olan farklı bit uzunluklarındaki ALU tasarımının farklı teknolojiler ile açık kaynak kodlu bilgisayar destekli program ile analizi değerlendirilmiştir. Bu çalışma ile dijital tasarım ve fiziksel tasarım aşamalarından sonra tasarımın doğruluğunu kontrol etmek için simülasyon yapılmaktadır. Yapılan simülasyonlar ile aynı bit uzunluğundaki tasarımların farklı transistör teknolojilerinde sentezlenmesinin doğruluğu kontrol edilmektedir. Aynı zamanda, dijital tasarım sonrasında yapılan simülasyon ile uyumluluğu gözlemlenerek fiziksel tasarımının doğruluğu teyit edilmektedir. Çalışma bu yönleriyle diğer çalışmalardan farklılaşarak alana katkı sunmaktadır.

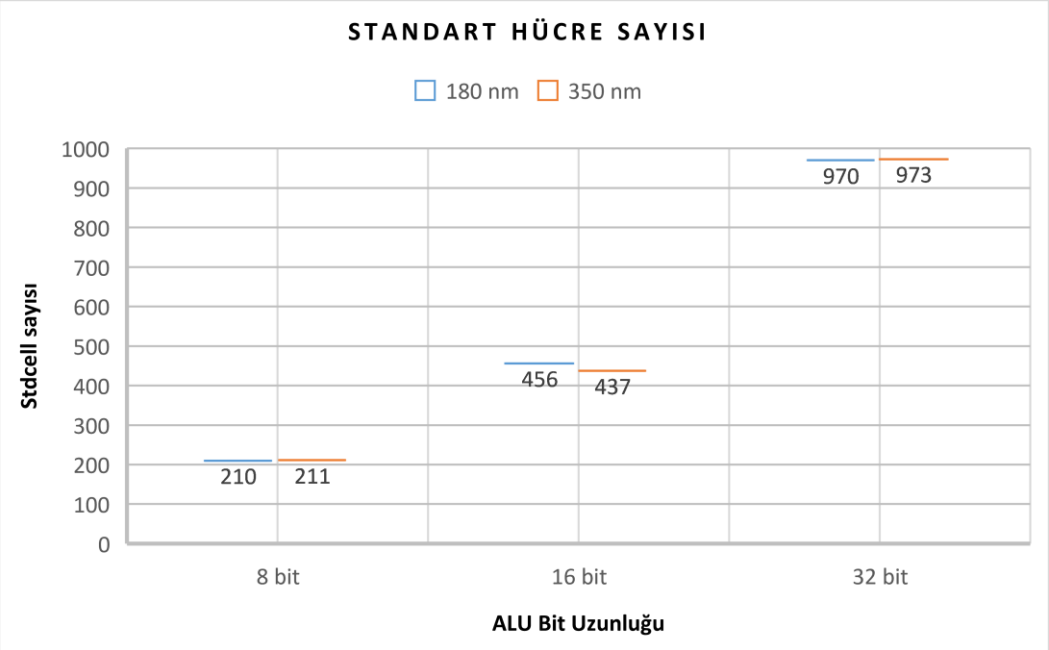
Tez kapsamında yapılan çalışmada dijital tasarımın süreçleri olan lojik tasarım ve fiziksel tasarım süreçleri gerçekleştirilmiştir. Lojik tasarım için donanım tanımlama dillerinden olan Verilog ile programlama yapılmıştır. Fiziksel tasarım için açık kaynak kodlu qflow aracı kullanılmıştır. Literatürde yer alan açık kaynak kodlu araçlar; qFlow, OpenLANE, Coriolis. VTR ve SymbiFlow olarak karşımıza çıkmaktadır [2]. Tez çalışmasında qflow aracı tercih edilmesinin nedeni farklı transistör teknoloji dosyalarını destekleyebilmekte ve diğer araçlara göre sentezlenecek tasarımı daha basit ve düzenli olarak yerleştirmektedir [2].

Aynı bit uzunluğuna sahip ALU tasarımlarının farklı transistör teknolojileri ile sentezlenmesi sonucunda transistör teknolojisindeki değişimin standart hücre sayılarına olan etkisi gözlemlenmiştir. Standart hücre sayıları Qflow aracının kayıtlarında yer alan place.log dosyasında tutulmaktadır. 8 bit, 16 bit ve 32 bit ALU tasarımlarının 180nm ve 350nm transistör teknolojisiyle sentezlenmesi sonrasında elde edilen hücre sayısı bilgileri Tablo 6.1'de gözlemlenmektedir. Teknoloji boyutu küçülse de tasarımı oluşturan standart hücre sayısında belirgin bir değişiklik olmamaktadır. Böylece dijital tasarımımızın sadeleştirilerek yapıldığı gözlemlenmiştir. Çalışma bu yönüyle literatür ile örtüşmektedir [4].

Farklı bit uzunluğuna sahip ALU tasarımlarının aynı transistör teknolojisinde sentezlenmesi sonucunda bit uzunluğunun standart hücre sayısına olan etkisi gözlemlenmiştir. 180 nm transistör teknolojileri ile 8 bit, 16 bit ve 32 bit ALU tasarımlarının sentezlenmesi sonucunda standart hücre sayısına ait bilgi Qflow aracının kayıtlarından biri olan place.log dosyasında tutulmaktadır. Elde edilen bilgiler Tablo 6.1'de gözlemlenmektedir. Tasarımın bit uzunluğu

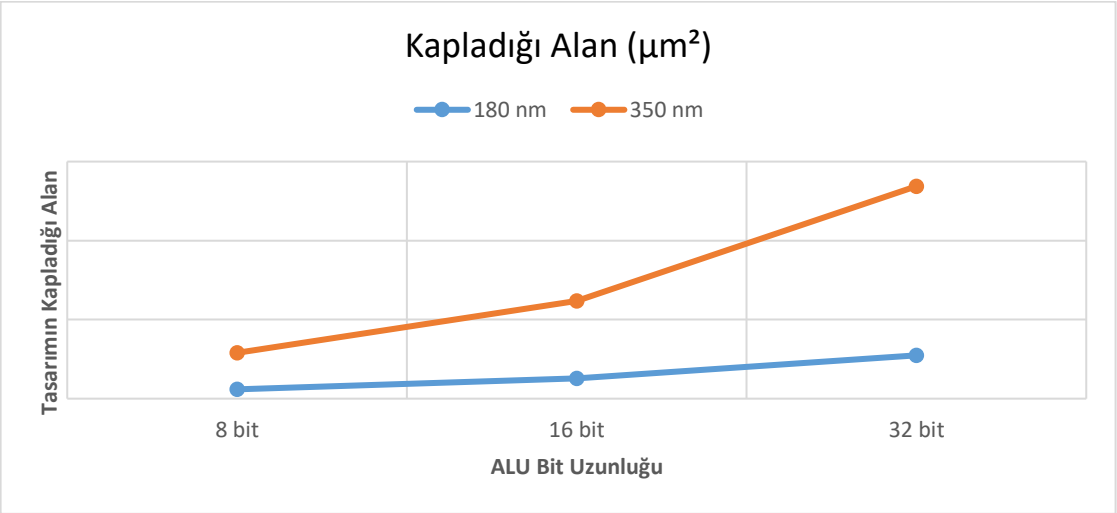
arttıkça tasarımı oluşturan hücre sayısı artışı ile bit uzunluğundaki artış doğru orantılı olarak gözlemlenmektedir. Bu durum literatür ile uyuşmaktadır [4].

Tablo 6.1: Farklı bit uzunluklarında ALU tasarımlarının standart hücre sayısı.



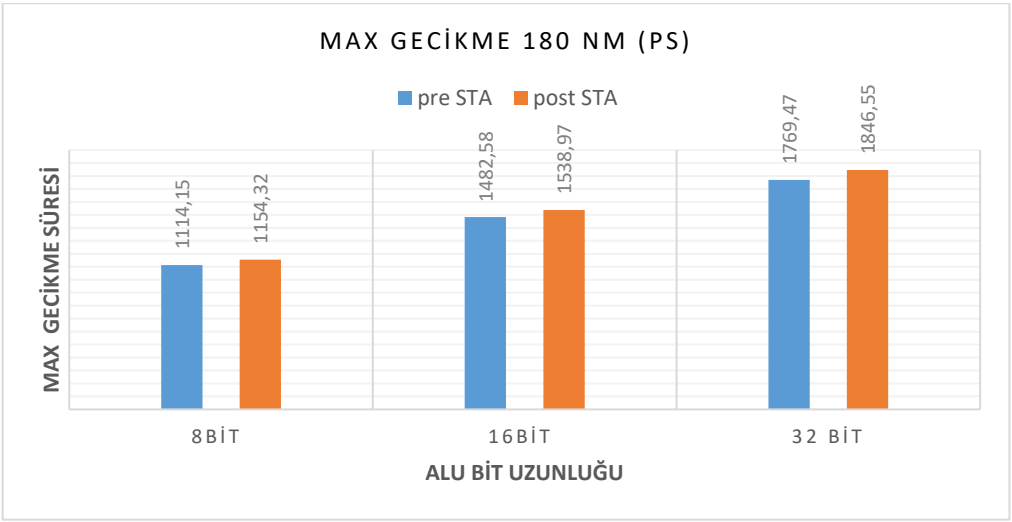
Aynı bit uzunluklarındaki ALU tasarımlarının farklı transistör teknolojileri ile sentezlenmesi sonucunda transistör teknolojisindeki değişimin, tasarımın kapladığı alan üzerindeki etkisi gözlemlenmiştir. Qflow ile yapılan başarılı sentezleme sonrasında place.log dosyasında tasarımların kapladığı alan bilgisi tutulmaktadır. 180nm transistör teknolojisi ile 8 bit, 16 bit ve 32 bit ALU tasarımların kapladığı alan bilgisi Tablo 6.2'de gösterilmiştir. Aynı tabloda 350nm transistör teknolojisi ile yapılan sentezleme sonuçları da yer almaktadır. Fiziksel sentezlemede kullanılan transistör teknolojisinin küçülmesi ile aynı bit uzunluğuna sahip olan tasarımların kapladığı alan da azalmaktadır. Bununla birlikte aynı transistör teknolojisinde sentezlenen tasarımların bit uzunluğunun artması ile tasarımın kapladığı alan da artmaktadır. Elde edilen sonuçlar literatürde yer alan çalışmalardaki sonuçlar ile tutarlılık göstermektedir [3].

Tablo 6.2: Farklı bit uzunluklarında ALU tasarımlarının kapladığı alan.



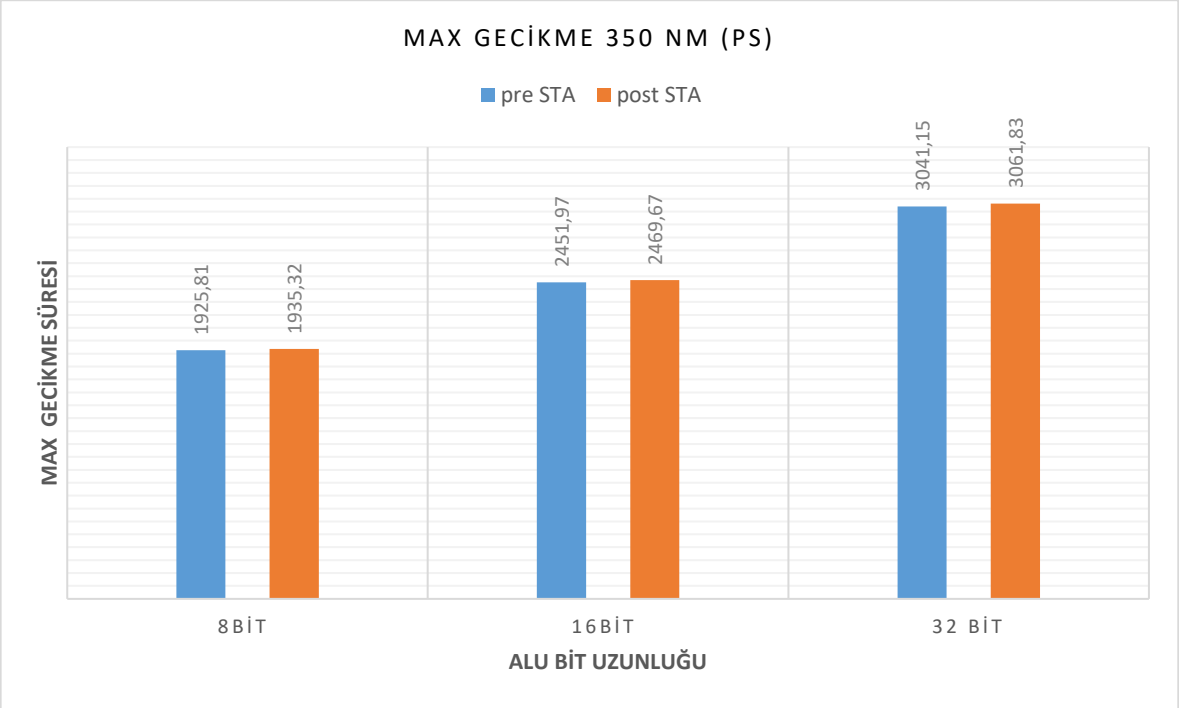
Farklı bit uzunluklarındaki ALU tasarımlarının aynı transistör teknolojileri ile sentezlenmesi sonucunda tasarımın sahip olduğu gecikmeler gözlemlenmiştir. Qflow ile yapılan başarılı sentezleme sonrasında sta.log dosyasında tasarımların sahip olduğu gecikme bilgisi yer almaktadır. 180nm transistör teknolojisi ile 8 bit, 16 bit ve 32 bit ALU tasarımların sahip olduğu pre ve post STA değerleri Tablo 6.3'de yer almaktadır. Tabloda yer alan preSTA değeri fiziksel sentez sırasında önce kabaca yerleşim yapılarak hesaplan değerdir. Ardından kapasitif etkilerin de devreye girmesi ile oluşan post STA hesaplanması ile gecikmeleri de açık kaynak kodlu yazılım ile gözlemlenmiştir. Tabloda da görüldüğü üzere 8 bit ALU tasarımının sahip olduğu post STA değeri pre STA değerinden büyük olması ve bit uzunluğu artıkça sahip olunan gecikme süresinin artıyor olması literatürde yer alan çalışmalar ile örtüşmektedir [3].

Tablo 6.3: 180 nm teknolojisinde pre ve post STA süreleri.



350 nm transistör teknolojisi ile 8 bit, 16 bit ve 32 bit ALU tasarımlarının da fiziksel sentezlemesi yapılarak tasarımın sahip olduğu gecikme süreleri gözlemlenmiştir 350 nm transistör teknolojisi ile gerçekleştirilen sentezleme sonrasındaki sta.log dosyasından sta.log dosyasından alınan sonuçlar Tablo 6.4 kapsamında yer almaktadır. 180nm transistör teknolojisine göre 350 nm transistör teknolojisinde hem pre hem de post STA değerlerinin artığı olduğu gözlemlenmiştir. Bu değerlendirme transistör teknolojisindeki küçülmenin tasarımlarının sahip olduğu işlem sürelerinin kısalttığını destekleyen bir sonuçtur.

Tablo 6.4: 350 nm teknolojisinde pre ve post STA süreleri.



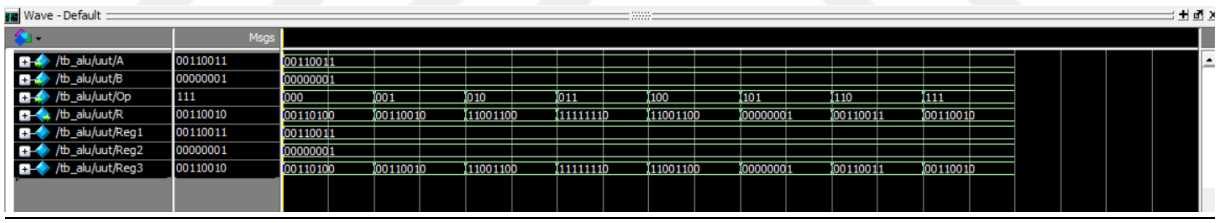
Tez çalışasında lojik tasarım ve fiziksel tasarım aşamalarının doğruluğunu kontrol etmek için simülasyonlar gerçekleştirilmiştir. Bu simülasyonlar 8 bit, 16 bit ve 32 bit ALU tasarımlarının hem 180nm hem de 350nm transistör teknolojilerinde sentezlenmesinden sonra tekrarlanmıştır. Elde edilen sonuçlar lojik tasarım simülasyonlar ile fiziksel tasarım simülasyonları birbiri ile örtüştüğü görülmüştür. ALU tasarımında yer alan işlemler ve referans alınan başlangıç değerlerine karşılık elde edilmesi gereken simülasyon sonuçları Tablo 6.5 özelinde yer almaktadır. 8 bit ALU tasarımının kontrolünün yapılması için gerçekleştirilen ModelSim simülasyonu ve sonrasında 180nm transistör teknolojisinde fiziksel sentezinin yapılmasının ardından gerçekleştirilen IRSIM simülasyon sonucu çıktıları Şekil 6.1 ve Şekil 6.2 de gösterilmektedir.

Tablo 6.5 : ModelSim ve IRSIM sonuçlarının değerlendirilmesi.

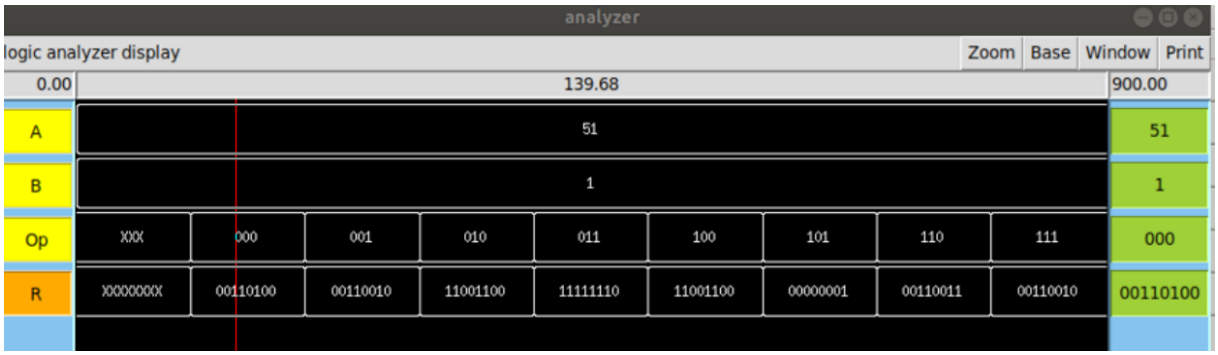
Op	ALU İşlemi	Reg 1 / A	Reg2 / B	ModelSim	İrsim
0	Toplama İşlemi	00110011	0000001	00110100	00110100
1	Çıkartma İşlemi	00110011	0000001	00110010	00110010
2	Tersini Alma İşlemi	00110011	0000001	11001100	11001100
3	VeDeğil İşlemi	00110011	0000001	11111110	11111110
4	VeyaDeğil işlemi	00110011	0000001	11001100	11001100
5	Ve İşlemi	00110011	0000001	00000001	00000001
6	Veya İşlemi	00110011	0000001	00110011	00110011
7	Özelveya İşlemi	00110011	0000001	00110010	00110010

Şekil 6.1'de yer alan ModelSim çıktısında ve Şekil 6.2'de yer alan IRSIM çıktısında, işleme tabi olan 8 bir değerler sırası ile binary olarak 00110011 (hex 51) ve 0000001 (hex 1) OP değerinin 000 olası durumunda olması gereken çıktı 00110100'dır. Bu sonucu iki simülasyon gözlemlenmektedir. Aynı şekilde diğer işlemlerde de Tablo 6.5 ile uyumluluk sağlanmaktadır.

Şekil 6.1: ModelSim ile 8 bit ALU tasarımı simülasyonu – Lojik Sentez.



Şekil 6.2: IRSIM ile 8 bit ALU tasarımı simülasyonu – Fiziksel Sentez.



Bu çalışma ile fiziksel sentez sonrasında açık kaynak kodlu bir araç ile simülasyon çalışması gerçekleştirilmiştir. Ayrıca, lojik sentez ve fiziksel sentez sonrasında yapılan simülasyon

karşılaştırmaları ile de tasarımların doğrulukları kontrol edilmiştir. Transistör teknolojisinde yaşanan küçülmenin açık kaynak kodlu bir araç ile farklı bit uzunluklarındaki standart hücre sayısına etkisini, tasarımın kapladığı alana olan etkisi ve tasarımın sahip olduğu gecikmeye etkisi gözlenmiştir.

Tez kapsamında transistör teknolojisindeki değişiminin tasarımın bit uzunluğuna göre etkisi açı kaynak kodlu yazılım ile gözlemlenmiştir. Transistör teknolojisin küçülmesi tasarımı oluşturan standart hücre sayısına belirgin bir değişiklik göstermemiştir. Gelişen transistör teknolojisi sayesinde transistör teknolojisindeki küçülme tasarımdaki bit uzunluğu artsa dahi tasarımın kapladığı alanı küçültmüştür. Fiziksel sentez sonrasında tasarımın sahip olduğu gecikme süresi transistör teknolojisi küçülmesi ile azalmıştır.

Bu çalışma ile dijital tasarım aşamasından ve fiziksel tasarım aşamasından sonra yapılan simülasyon çalışmaları ile tasarımın doğruluğu kontrol edilmiştir. Açık kaynak kodlu araç ile farklı transistör teknolojisindeki tasarımların simülasyon kontrolü ile doğruluğunun kontrolü literatüre yapılan katkıdır.

VLSI tasarım süreçlerinin yaygınlaşması açısından açık kaynak kodlu yazılımların geliştirilmesi ve kullanımının yaygınlaştırılması gerekmektedir. Bu sayede VLSI tasarım alanında ki çalışmalar yaygınlaştırılarak yetişmiş iş gücü artırılmasına imkan sağlanabilecektir.

KAYNAKLAR

[1]. https://en.wikipedia.org/wiki/List_of_EDA_companies, websitesi ziyareti: Ocak 2023

[2]. Chupilko,M. , Kamkin A. ve Smolov, S., Survey of Open-source Flows for Digital Hardware Design, *2021 Ivannikov Memorial Workshop (IVMEM)*, Nizhny Novgorod, Russian Federation, 2021, pp. 11-16

[3]. T. Gaurav, K. Patel ve R. Parekh, RTL to GDSII Implementation of RADIX-4 Booth Multiplier, *2022 IEEE International Conference on Nanoelectronics*, Nanophotonics, Nanomaterials, Nanobioscience & Nanotechnology (5NANO), Kottayam, India, 2022, pp. 1-5

[4]. S. Purohit, P. Laddha ve R. Parekh, Implementation and Physical Design of 8/4-Bit Signed Divider, *2021 8th International Conference on Signal Processing and Integrated Networks (SPIN)*, Noida, India, 2021, pp. 829-834

[5]. KANG, S. ve LEBLEBECI, Y. Yayın yılı, Introductıon, CMOS Digital Integrated Circuits: Analysis and Deisgn, In: TATA MCGRAW HILL (ed), Chapter 1 , Yayın evi, Basın evi , ISBN-10 : 0070530777, ISBN-13 : 978-0070530775, 1-8

[6]. Essentials of VLSI Circuits and Systems, Kamran Eshraghian, Eshraghian Dougles, A. Pucknell, 2005, PHI

[7]. https://ferhatbagis.blogspot.com/2017/05/vlsi-devre-tasarm.html, websitesi ziyareti: Ocak 2023

[8]. https://ourworldindata.org/technolojikal-change, Websitesi Ziyareti: Ocak 2023

[9]. https://tr.lambdageeks.com/vlsi-design-very-large-scale-integration/, Websitesi Ziyareti: Aralık 2022

[10]. Cao, L., 2021, Multi-objective Digital VLSI Design Optimisation, Doctor of Philosophy, University of York

[11]. “International roadmap for devices and systems (irds),” https://irds.ieee.org/, 2020, Websitesi Ziyareti: Kasım 2022

[12]. “International technology roadmap for semiconductors (itrs),” http://www.itrs2. net/, 2001, Websitesi Ziyareti: Kasım 2022

[13]. http://people.rennes.inria.fr/Olivier.Sentieys/?page_id=95, Websitesi Ziyareti: Kasım 2022

[14]. http://opencircuitdesign.com/qflow/welcome.html, Websitesi Ziyareti: Kasım 2022

[15]. https://vlsiarch.ecen.okstate.edu/flows/stdcell_datasheet/, Websitesi Ziyareti: Kasım 2022

[16]. https://www.elektrikde.com/dunyada-cip-ureten-ulkeler/, Websitesi Ziyareti: Kasım 2022

[17]. https://ungo.com.tr/2022/08/dunyanin-en-buyuk-cip-ureticileri/, Websitesi Ziyareti: Kasım 2022

[18]. https://www.kalkinmakutuphanesi.gov.tr/assets/upload/dosyalar/ankara-ili-ultrason-islevini-gerceklestirecek-akilli-cip-uretimi-on-fizibilite-raporu.pdf, Websitesi Ziyareti: Aralık 2022

[19]. https://bilgem.tubitak.gov.tr/tr/urunler/yital-yari-iletken-teknolojileri-arastirma-laboratuvari, Websitesi Ziyareti: Kasım 2022

[20]. https://www.defensehere.com/tr/yongatekin-yerli-cip-uretiminde-yol-haritasi, Websitesi Ziyareti: Kasım 2022

[21]. https://www.ermaksan.com.tr/tr-TR/, Websitesi Ziyareti: Kasım 2022

[22]. https://www.sbb.gov.tr/wp-content/uploads/2020/04/ElektronikSanayiiCalismaGrubuRaporu.pdf, Websitesi Ziyareti: Kasım 2022

[23]. Weste, N. ve Harris, D.,2009, CMOS VLSI Design-A Circuits and Systems Perspectiv, Pearson,3rd Edn, Ayan Banerjee, ISBN 10: 0-321-54774-8

[24]. https://en.wikipedia.org/wiki/Physical_design_%28electronics%29, Websitesi Ziyareti: Kasım 2022

[25]. Aydın, M., 2022, *FPGA Tabanlı Gerçek Zamanlı Görüntü İşleme ile Baskı Devre Kartı Hatalarınının Tespit Edilmesi*, Thesis (PhD), İstanbul Üniversitesi-Cerrahpaşa

[26]. Bedir, N.,2018, *FPGA Tabanlı 64 Bit Aritmetik Mantık Birimi Tasarımı*, Thesis (PhD), İstanbul Üniversitesi-Cerrahpaşa

[27]. Başak, S., 2011, *FPGA Tabanlı Sentezlenebilir İşlemci Tasarımı*, Thesis (PhD), Yıldız Teknik Üniversitesi

[28]. Sırmaçek, B., 2007, *FPGA ile Mobil Robot için Öğrenme Algoritması Modellenmesi*, Thesis (PhD), Yıldız Teknik Üniversitesi

[29]. Bakacak, M., 2021, *Eğitim Amaçlı Temel Bir Mikroişlemci Tasarımı ve Gömülü Sistem Uygulaması*, Thesis (PhD), Kırıkkale Üniversitesi

[30]. Çömlekçi, M., Tüzel, S., PC Donanımı Herkes İçin, Alfa BasımYayım Dağıtım, Nisan, 2005

[31]. Henkoğlu Türkay, Modern Donanım Mimarisi, Pusula Yayıncılık, Şubat, 2005

[32]. Pala, Z., Bilgisayar Donanımı, Türkmen Kitapevi, 2005

[33]. https://tr.wikipedia.org/wiki/Aritmetik_mant%C4%B1k_birimi, Websitesi Ziyareti: Kasım 2022

[34]. http://samples.jbpub.com/9781449688394/88394_CH03_Savage.pdf, Websitesi Ziyareti: Kasım 2022

[35]. Yaşar, İ., 2009, Otonom Araç için DA Motor Sürücü Sistemi ve Denetleyicinin Gerçeklenmesi, Thesis (PhD), Karadeniz Teknik Üniversitesi

[36]. https://www.intel.com/content/www/us/en/products/programmable.html, Websitesi Ziyareti: Kasım 2022

[37]. Başa, B., 2014, FPGA Yapıları ile Dijital Osiloskop Gerçeklemesi, Thesis (PhD), Sakarya Üniversitesi

[38]. https://www.intel.com.tr/content/www/tr/tr/products/details/fpga/development-tools/quartus-prime/resource.html, Websitesi Ziyareti: Kasım 2022

[39]. https://www.intel.com/content/www/us/en/software/programmable/quartus-prime/questa-edition.html?wapkw=modelsim, Websitesi Ziyareti: Kasım 2022

[40]. Yosys Open SYnthesis Suite, https://yosyshq.net/yosys/, Websitesi Ziyareti: Kasım 2022

[41]. Odin II, https://docs.verilogtorouting.org/en/latest/odin/ , Websitesi Ziyareti: Kasım 2022

[42]. rubund/graywolf, https://github.com/rubund/graywolf, Websitesi Ziyareti: Aralık 2022

[43]. Qrouter 1.3 (Stable) and 1.4 (Development), http://opencircuitdesign.com/qrouter, Websitesi Ziyareti: Aralık 2022

[44]. OpenSTA engine, https://github.com/The-OpenROAD-Project/OpenSTA, Websitesi Ziyareti: Aralık 2022

[45]. qflow/vesta.c, https://github.com/RTimothyEdwards/qflow/blob/master/src/vesta.c, Websitesi Ziyareti: Aralık 2022

[46]. Magic VLSI Layout Tool, http://opencircuitdesign.com/magic, Websitesi Ziyareti: Aralık 2022

[47]. Netgen 1.5, http://opencircuitdesign.com/netgen, Websitesi Ziyareti: Aralık 2022

[48]. https://vlsiarch.ecen.okstate.edu/flows/, Websitesi Ziyareti: Aralık 2022

[49]. Qflow 1.4, http://opencircuitdesign.com/qflow/download.html, Websitesi Ziyareti: Aralık 2022

[50]. https://www.muchen.ca/documents/ELEC402/t0-qflow.html#installing-yosys, Websitesi Ziyareti: Aralık 2022

[51]. OpenCircuitDesign, http://opencircuitdesign.com/index.html, Websitesi Ziyareti: Aralık 2022

[52]. IRSIM 9.7, http://opencircuitdesign.com/irsim/index.html, Websitesi Ziyareti: Aralık 2022

İNTİHAL RAPORU İLK SAYFASI

ORİJİNALLİK RAPORU

%12	%10	%3	%6
BENZERLİK ENDEKSİ	İNTERNET KAYNAKLARI	YAYINLAR	ÖĞRENCİ ÖDEVLERİ

BİRİNCİL KAYNAKLAR

1	Submitted to The Scientific & Technological Research Council of Turkey (TUBITAK) Öğrenci Ödevi	%3
2	acikbilim.yok.gov.tr İnternet Kaynağı	%2
3	www.muchen.ca İnternet Kaynağı	%1
4	www.kalkinmakutuphanesi.gov.tr İnternet Kaynağı	%1
5	github.com İnternet Kaynağı	%1
6	Tanya Gaurav, Krutarthkumar Patel, Rutu Parekh. "RTL to GDSII Implementation of RADIX-4 Booth Multiplier", 2022 IEEE International Conference on Nanoelectronics, Nanophotonics, Nanomaterials, Nanobioscience & Nanotechnology (5NANO), 2022 Yayın	%1