

**KOCAELİ ÜNİVERSİTESİ \* FEN BİLİMLERİ ENSTİTÜSÜ**

**FOTOVOLTAİK SİSTEMLERDE FPGA KULLANIMI**

**YÜKSEK LİSANS TEZİ**

**Teknik Öğretmen Aziz GÜNEROĞLU**

**Ana Bilim Dalı: Elektrik Eğitimi**

**Danışman: Yrd. Doç. Dr. Özcan ATLAM**

**KOCAELİ, 2008**

**KOCAELİ ÜNİVERSİTESİ \* FEN BİLİMLERİ ENSTİTÜSÜ**

**FOTOVOLTAİK SİSTEMLERDE FPGA KULLANIMI**


**YÜKSEK LİSANS TEZİ**

**Teknik Öğretmen Aziz GÜNEROĞLU**

**Tezin Enstitüye Verildiği Tarih : 23 Mayıs 2008  
Tezin Savunulduğu Tarih: 25 Haziran 2008**

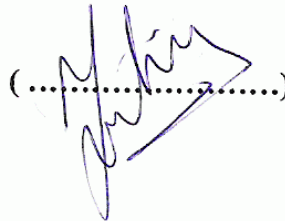
**Tez Danışmanı  
Yrd.Doç.Dr.Özcan  
ATLAM**

(.....)



**Üye  
Yrd.Doç.Dr.Mehmet  
YILDIRIM**

(.....)



**Üye  
Yrd.Doç.Dr.Ferdi  
BOYNAK**

(.....)



**KOCAELİ, 2008**

## ÖNSÖZ ve TEŞEKKÜR

FPGA tabanlı fotovoltaik güç sistemlerinin denetlenmesi diğer işlemcilerle gerçekleştirilen denetleme yöntemleri ile karşılaştırıldığında ilk göze çarpan FPGA üzerinde gerçekleştirilen mimarinin çok amaçlı problemleri çözebilecek devre tasarımlarını içerebilmesidir. Eğer aynı sistem bir ASIC tümleşik devre üzerinde gerçekleştirilmek istenirse tek bir amaca yönelik bir devre yapılabilecekti.

FPGA'ler programlama kabiliyetleri yönüyle DSP ve mikroişlemcilere göre sınırlıdır. Bununla birlikte bir çok CAD (Bilgisayar destekli çizim) aracı FPGA tasarımını desteklemekte ve bu sayede kodlanan devreler, lojik elemanlarıyla bir devre çizimi olarak gösterilmektedir. FPGA ların esnek yapıları ile birlikte VHDL' lin tasarımcılara sağladığı imkanlarla, tasarımlar çok daha çabuk ve etkili bir şekilde gerçekleştirilebilmektedir.

Bu çalışmada FPGA üzerinde gerçekleşen fotovoltaik güç denetimi için maksimum güç noktası izleyicisi algoritmasının tasarımı, simülasyonu ve akış şeması verilmiştir. Tasarımlar Xilinx ISE 8.1i ve simülasyonlar ModelSim SE 6.2e programlarının yardımı ile gerçekleştirilmiştir.

Tez çalışmalarına ayırdığım zamanı anlayışla karşılayıp beni destekleyen eşim Seher SERT GÜNEROĞLU'na ve çalışmalarımı yürütürken az zaman ayırdığım minik Kızım CEYLİN'e ve bu güne kadar öğrettikleriyle bana katkıda bulunan herkese teşekkür ederim. FPGA ve VHDL ile tanışmamı sağlayan ve bu konuda bire bir eğitim vererek bu konuları bana öğreten Öğretim Görevlisi Sayın Suhap ŞAHİN'e ve çalışmalarında bana destek olan sayın Metin SUNAN'a teşekkür ederim.

Öncelikle Beni Yenilenebilir enerji kaynakları üzerinde çalışmamı sağlayan, çalışmam esnasında sahip olduğu tüm kaynakları ve bilgileri ile her türlü desteği sunan, Fotovoltaik Sistemlerde FPGA ve VHDL kullanımı konusunda çalışmaya iten ve bu çalışmalar esnasında fikirleri ile bana yardımcı olan değerli hocam Yrd.Doç.Dr. Özcan ATLAM'a teşekkür ederim.

Mayıs 2008, KOCAELİ

Aziz GÜNEROĞLU

## İÇİNDEKİLER

ÖNSÖZ .....	i
İÇİNDEKİLER .....	ii
ŞEKİLLER DİZİNİ.....	iv
TABLOLAR DİZİNİ .....	vi
SİMGELER.....	vii
ÖZET .....	viii
ABSTARCT .....	ix
1. GİRİŞ .....	1
2. GÜNEŞ ENERJİSİ .....	3
2.1. Güneş Pilleri ( Fotovoltaik Piller ).....	6
2.2. Güneş Pillerinin Yapısı ve Çalışması.....	7
2.3 PV Model Çalışmaları.....	13
3. FPGA İLE PV SİSTEM DENETİMİ VE PV MODELİ.....	15
3.1. FPGA ile PV Sistemlerinin Güç Denetimi.....	15
3.2 Referans Model (Tek Diyot Modeli) .....	16
4. MAKSİMUM GÜÇ NOKTASI İZLEYİCİ .....	20
4.1 Tasarım kriterleri.....	21
5. PROGRAMLANABİLİR LOJİK ELEMANLARIN MİMARİSİ VE PROGRAMLAMA TEKNİKLERİ .....	23
5.1 Programlanabilir Lojik Elemanların Gelişimi.....	25
5.2 Alan Programlamalı Kapı Dizileri (FPGA) .....	28
5.2.1 FPGA'lerin mimarisi.....	29
5.2.2 FPGA'lerin programlama teknolojileri .....	31
5.2.3 FPGA'ların lojik hücre mimarisi .....	34
5.3 FPGA Kullanılarak Gerçekleştirilen Devrelerin Tasarım Süreci.....	35
5.4 VHDL Donanım Tasarım Dili .....	37
5.4.1 Giriş.....	37
5.4.2 VHDL ve donanım tasarımı karşılaştırılması .....	37
5.4.3 VHDL dili mimari yapıları.....	38
5.4.5 Veri türleri ve nesnelere.....	44
5.4.6 Arayüz listeleri .....	46
5.4.7 VHDL dili ana yapıları.....	46
6. UYGULAMA .....	53
6.1 Conductance Incremental Metotlu MPPT Algoritması.....	53
6.2 FPGA Tasarım Mimarisi.....	55
6.3 Veri Gösterimi.....	58
6.3.1 Kayan noktalı sayılar aritmetiği .....	58
6.3.2 Kayan noktalı (Floating Point) sayıların gösterimi: .....	58
6.3.3 Toplama Ve Çıkarma .....	59
6.3.4 Bölme .....	61
6.4 Algoritmanın Uygulanması.....	62
6.4.1 VHDL Kodları .....	62
6.4.2 PV modeli .....	66

6.4.2 Uygulama sonuçları .....	67
7. SONUÇLAR VE ÖNERİLER .....	76
7.1 Veri Duyarlılığı, Performans Niteliği ve Alanın Verimli Kullanılması.....	78
7.2 Çözüm Yöntemi .....	78
7.2.1 Sayısal test ve karşılaştırma .....	79
7.3 Sonuç.....	79
KAYNAKLAR .....	79
ÖZGEÇMİŞ .....	82

## ŞEKİLLER DİZİNİ

Şekil 2.1 Güneş Pili.....	7
Şekil 2.2 :Güneş pilindeki p ve n maddeleri.....	8
Şekil 2.3 : a) PV'deki fotovoltaik enerji dönüşümü.....	9
b) PV prensip eşdeğer devresi.....	10
Şekil 2.4: PV'nin I-V eğrisi.....	11
Şekil 2.5: Farklı ışınım şiddetlerinde I-V eğrileri.....	12
Şekil 2.6: Sıcaklık değişiminin PV'deki etkisi.....	13
Şekil 3.1: PV eşdeğer devresi.....	16
Şekil 3.2: R yükünün PV'deki çalışma noktaları.....	18
Şekil 4.1: Güneş paneli akım-gerilim ve güç-gerilim eğrisi.....	20
Şekil 5.1: PLA lojik devre elemanın iç mimarisi.....	26
Şekil 5.2: PAL lojik devre elemanın iç mimarisi.....	26
Şekil 5.3: CPLD lojik devre elemanın iç mimarisi.....	27
Şekil 5.4: MPGA lojik devre elemanın iç mimarisi.....	28
Şekil 5.5: FPGA mimarisi.....	29
Şekil 5.6: FPGA'yi oluşturan bloklar.....	29
Şekil 5.7: CLB iç yapısı.....	30
Şekil 5.8: CLB'yi oluşturan LUT elemanı.....	30
Şekil 5.9: bağlantı blokları.....	31
Şekil 5.10: Bir Statik RAM hafıza hücresi.....	32
Şekil 5.11. (a)LUT, (b)PIP, (c)Multiplexer.....	33
Şekil 5.12. İki girişli LUT yapısı.....	34
Şekil 5.13. Çoklayıcı tabanlı lojik hücre yapısı.....	35
Şekil 5.14: FPGA kullanılarak yapılan tasarım sürecinin akış diyagramı.....	36
Şekil 5.15: Yarı toplayıcı giriş ve çıkışları.....	42
Şekil 5.16: Tam toplayıcı giriş çıkışları.....	44
Şekil 5.17: Tam toplayıcı iç yapısı.....	44
Şekil 6.1: PV'nin $dP / dV$ değişimini gösteren P – V karakteristik eğrisi.....	54
Şekil 6.2: Incremental Conductance Algoritmasının Akış Diyagram Şeması.....	55
Şekil 6.3: MPPT Algoritmasının ISE programındaki görünümü.....	56
Şekil 6.4: MPPT varlık mimarisi.....	57
Şekil 6.5: VHDL kodunun benzetim çıktıları.....	57
Şekil 6.6: IEEE 754 32 Bit Kayan sayı formatı.....	58
Şekil 6.7: Kayan Noktalı sayıların toplanması.....	60
Şekil 6.8: Kayan Noktalı sayıların bölünmesi.....	61
Şekil 6.9: Incremental Conductance algoritmasının VHDL ile gerçekleştirilmesi.....	62
Şekil 6.10: Sistemin genel blok şeması.....	65
Şekil 6.11: Incremental Conductance algoritmasının lojik devre mimari şeması.....	65
Şekil 6.12: PV modelinin gerçekleştirilmesi.....	66
Şekil 6.13: PV modelinin I-V Eğrisi.....	67
Şekil 6.14: Maksimum gücün takibi.....	68
Şekil 6.15: Maksimum gücün P-V grafiği üzerindeki takibi.....	69
Şekil 6.16: Maksimum güc eğrisinin I-V eğrisi üzerinde karşılaştırılması.....	70

Şekil 6.17: Maksimum güc eğrisinin P-V grafiği üzerinde karşılaştırılması.....	70
Şekil 6.18: Algoritmada kullanılan matematiksel işlemlerin benzetim görüntüsü....	74
Şekil 6.19: FPGA' nın ürettiği referans gerilimin benzetim görüntüsü.....	75

## TABLolar DİZİNİ

Tablo 2.1: Kıtalar bazında güneşlenme miktarları .....	4
Tablo 2.2: Türkiyedeki güneşlenme oranları .....	5
Tablo 2.3: Türkiyenin bölgelerine göre güneş enerjisi potansiyeli .....	5
Tablo 6.1: Sistemin testinde kullanılan akım ve gerilimler .....	71
Tablo 6.2: FPGA Sisteminin ürettiği referans gerilimler .....	72
Tablo 6.3: Sistemin bağıl hataları .....	73



## SİMGELER

$I_g$	: Işık fotonlarıyla üretilen akım veya kısa devre akımı (A)
$I_o$	: Ters sızıntı akımı (A)
$I_d$	: Eklemdeki diyot akımı (A)
$I_{sh}$	: Paralel kol direncinden geçen akım (A)
$R_{sh}$	: Paralel kol direnci (ohm)
$R_s$	: Seri direnç (ohm)
$q$	: Elektron yükü ( $1,6 \cdot 10^{-19}$ c.)
$k$	: Boltzman sabiti ( $1,38 \cdot 10^{-23}$ )
$T$	: Mutlak sıcaklık ( $K^0$ )
$n$	: Diyot faktörü
$I$	: Yük akımı (A)
$V$	: Çıkış gerilimi (V)
$R$	: Yük direnci (ohm)
$P_{PV}$	: PV nin ürettiği güç(W)
$dP_{PV}$	: PV nin ürettiği güçteki değişim(W)
$V_{PV}$	: PV nin ürettiği gerilim(V)
$dV_{PV}$	: PV nin ürettiği gerilimdeki değişim(V)
$I_{PV}$	: PV nin ürettiği akım(A)
$dI_{PV}$	: PV nin ürettiği akımdaki değişim(A)
MPPT	: Maximum Power Point Tracking(Maksimum Güç Noktası İzleyicisi)
FPGA	:Field Programable Gate Array
VHDL	:Very High Speed Integrated Circuit Hardware Description Language
CAD	:Computer Aided Design
ASIC	:Application Specific Integrated Circuit
SRAM	: Static Random Access Memory
PROM	: Programable Read Only Memory
EPROM	: Erasable Programable Read Only Memory
EEPROM	: Electrically Erasable Programable Read Only Memory
PLD	: Programable Logic Device
PAL	: Programable Array Logic
PLA	: Programable Logic Array
SPLD	: Simple Programable Logic Device
CPLD	: Complex Programable Logic Device
MPGA	: Mask Programable Gate Array
LUT	: Look-up Table
EDIF	: Electronik Design Interchange Format
CLB	: Configurable Logic Blok

## FOTOVOLTAİK SİSTEMLERDE FPGA KULLANIMI

Aziz GÜNEROĞLU

**Anahtar Kelimeler:** FPGA ( Field Programmable Gate Arrays), Fotovoltaik Güç Sistemleri, Maksimum Güç Noktası İzleyicisi, Donanım Tanımlama Dilleri, Paralel Programlama, Tekrar Düzenlenebilir tasarımlar.

**ÖZET:** Fotovoltaik güç sistemlerinin FPGA ile gerçekleştirilmesi bu tür sistemlerin verimli ve çevresel değişimlere karşı hızlı tepki verilebilmesini sağlar. Gerçek zamanlı uygulamalar için farklı sistemler kullanılarak çevresel değişikliklerin algılanması ve değişikliklere uygun bir şekilde Fotovoltaik pilin ürettiği gücün maksimum bir verimlilikle yüke aktarılmasına çalışılmaktadır. Bu tür tasarımlarda farklı mikrodenetleyiciler ve elektronik ekipmanlar kullanılarak çoğu kez karmaşık tasarımlar gerçekleştirilmeye çalışılmaktadır.

FPGA ların esnek programlama yapıları, hız, paralel işlem yapabilme yetenekleri bakımından üstündürler. Tekrar düzenlenebilir FPGA programlanması ile özel amaçlı hızlı donanımlar çok geniş uygulamalar için kullanılabilir. FPGA'ların, geleneksel işlemcilerin sahip olmadığı hız, güvenlik ve paralel işlem yapabilme yeteneğine ve ayrıca VLSI teknolojisine sahip olmadığı tekrar düzenlenebilirlik kabiliyetlerine sahip olması nedeniyle Fotovoltaik güç denetim sistemlerinde maksimum güç noktası takibi için kullanılan algoritmaların gerçekleştirilmesine olanak sağlayabilmektedir.

Bu çalışmanın amacı FPGA kullanarak Fotovoltaik sistemlerin güç denetiminde maksimum güç noktasının takibinde (MPPT) kullanılan algoritmasının gerçekleştirilmesidir. Dijital sistem mimarisi, Conductance Incremental Method ("C.I.") olarak bilinen MPPT algoritmasını gerçekleştirmek için tasarlanmıştır. PV nin ürettiği akım ve gerilimler FPGA mimarisi içerisine alınarak VHDL kodlarına adım adım uygulanıp VHDL programının ürettiği yeni referans gerilimleri sisteme uygulayarak sistemin maksimum noktaya ulaşması ve bu noktada kalması sağlanmıştır. PV nin %100 ile % 25 lik farklı ışınım şiddetlerindeki değişimlerde ürettiği gerçek maksimum güç noktaları ile FPGA'nın yakaladığı maksimum güç noktaları arasında ki bağıl hatalar ise %1,6 ile % 4.03 arasında olmaktadır. Tasarım mimarisi VHDL (Very High Speed Integrated Circuits Hardware Description Language ) dili ile tanımlandı.

## USING OF FPGA IN THE PHOTOVOLTAIC SYSTEMS

Aziz GÜNEROĞLU

**Key words:** FPGAs, Photovoltaic Power Systems, Maximum Power Point Tracking, Hardware Description Language, Reconfigurable Designings.

**ABSTARCT:** The FPGAs approach for Photovoltaic Power Systems implementation provides the flexibility in programmable systems. For the Maximum Power Point Tracking Systems Which are used for the Photovoltaic Power Systems based instrument prototype in real time application, conventional specific VLSI chip desing suffer the limitation in easiness, time and cost.

FPGAs have high speed and small size for real time application then the VLSI design. In addition the Maximum Power Point Tracking Systems based on FPGAs has fairly achieved with Photovoltaic Power Systems application.

The programmability of te configurable FPGAs yields the availability of fast special purpose hardware for wide applications. Its programmability, reliability and parallel working could set the conditions to implement the Maximum Power Point Tracking algorithms which are used for Photovoltaic Power Systems Control of a scale that would not be feasible with conventional processor.

The goal of this work is to a hardware implementation of a Maximum Power Point Tracking Algorithm which is known Conductance Incremental Method("C.I.") using Field Programmable Gate Array (FPGA). The Maximum Power Point values are fixed for operating by FPGA. Relative errors between FPGA's maximum power points and real maximum power points data are about 0.1,6 - 4,03% for a 100-25 % irradiance interval. A digital system architecture is described using Very High Speed Integrated Circuits Hardware Description Language (VHDL).

## 1. GİRİŞ

Enerji hızla gelişmekte olan teknolojik dünyamızın vazgeçilemez unsurlarından biridir. Enerji ihtiyacını karşılamaya dönük yaygın olarak kullanılan kömür, petrol ve benzeri yer altı kaynaklarının zamanla tükenmesi, her geçen gün artan maliyetlerinin yanında kullanımları sırasında çevreye verdikleri kirlenme, küresel ısınma vb. olumsuz etkileri göz önüne alındığında güneş, rüzgar gibi doğada doğal olarak bulunan enerji kaynakları daha bir önem kazanmaktadır. Yenilenebilir ya da temiz enerji olarak adlandırılan bu kaynaklarla gerçekleştirilen güç uygulamaları, çalışma performanslarının düşük olmasına rağmen son yıllarda araştırmacıların bu kaynaklara olan ilgileri artmaktadır. Özellikle ülkelerin sahip olduğu ulusal elektrik şebekelerinden uzak bölgeler ve yerel güç uygulamaları için yenilenebilir enerji, ekonomikliği ve kurulum kolaylığı nedeniyle tercih edilmektedir. Bu kaynakların önde geleni olan güneş, kullanımı en kolay ve diğer enerji kaynaklarının dolaylı veya dolaysız temelini oluşturan bir enerji kaynağıdır.

Güneş pili olarak da adlandırılan fotovoltaik güneş panelleri (PV) güneş ışığını doğrudan elektrik enerjisine dönüştürürler. Ürettikleri elektrik enerjisi doğru akım (DA) karakterindedir. Günümüzde üretilen PV verimleri genellikle %5-15 gibi düşük seviyelerde olup ve performansları güneş ışığına bağlı olduğundan sürekli değildir. Bu nedenle görünen kapasitelerine göre PV sistemleri tek başlarına bir alternatif enerji kaynağı olmaktan çok bir ek enerji kaynağı olarak kabul edilebilir. PV panelleri, doğal güneş ışığını kullanırlar. İşletim maliyetleri düşük olmaktadır. Elektrik enerjisi üretim esnasında herhangi bir atık söz konusu değildir. Bu üstünlükleri, düşük verimlerine rağmen onlara olan ilgiyi koruyabilmektedir. Hatta PV üretim maliyetleri, ilk üretildiklerinde çok fazla iken, yarıiletken teknoloji üretiminin gelişimiyle günümüze gelene kadar çok hızlı bir şekilde düşme göstermektedir. Günümüzde maliyetleri düşmesine karşın henüz istenilen düzeye gelmiş bulunmamaktadır. Üretim teknolojilerinin gelişimine paralel olarak maliyetlerinin ileride daha da düşürülmesi hedeflenmektedir.

Tezin konusunu teşkil eden Fotovoltaik güç sistemlerinde güç denetimini FPGA(Field Programmable Gate Arrays) ile gerçekleştirmek hedeflenmiştir. PV lerde üretilen DC enerjinin en verimli şekilde yüke aktarılması ve bu aktarımın dış ortam değişkenleri olarak bilinen atmosferik olaylar göz önünde bulundurularak süreklilik arz etmesi önemli olmaktadır. Bu yüzden PV sistemler her zaman, herhangi bir sıcaklık ve güneş ışınma seviyelerinde maksimum çıkış gücü seviyelerinde çalışacak şekilde tasarlanmalıdır.

Çalışmanın birinci bölümünde yenilenebilir enerji kaynakları, güneş pilleri ve FPGA ile güç denetimi hakkında kısa bilgiler verildi. İkinci bölümde güneş pilleri ve güneş pillerinin yapıları ve bunların model çalışmaları üzerinde duruldu. üçüncü bölümde ise FPGA ile PV sistemlerin güç denetimi ve bu denetimin gerçekleşebilmesi için referans PV modeli çıkarıldı. Dördüncü bölümde Maksimum güç noktası izleyicisi(MPPT) hakkında bilgiler sunuldu ve MPPT tasarım kriterleri belirlendi. Beşinci bölümde ise FPGA mimarisi, gelişimi, yapısal özellikleri tanıtılmış ve donanım tanımlama dili olan VHDL tanıtılmıştır. Altıncı bölümde uygulama tanıtılmış ve uygulamada kullanılan MPPT algoritması açıklanmış ve algoritmada kullanılan matematiksel hesapların aritmetiği açıklanmıştır. Son bölümde ise uygulama sonuçları ve önerilere yer verilmiştir.

Çalışmada PV sisteminden yüke aktarılacak maksimum güç noktalarının tespiti için kullanılan MPPT algoritmalarından Conductance Incremental Method(“C.I.”) algoritması kullanılarak FPGA da gerçekleştirilmiştir. Böylelikle PV nin ürettiği maksimum güç noktasının takip edilmesi ve yüke maksimum gücün aktarılmasına ve verimin artırılmasına çalışılmıştır. FPGA programlamak için yüksek seviyeli dil olan VHDL kullanılmıştır.

## 2. GÜNEŞ ENERJİSİ

Güneş enerjisi, güneşin çekirdeğinde yer alan füzyon süreci ile açığa çıkan ışıma enerjisidir ve güneşteki hidrojen gazının helyuma dönüşmesi şeklindeki füzyon sürecinden kaynaklanır. Dünya atmosferinin dışında güneş enerjisinin şiddeti, aşağı yukarı sabit ve  $1370 \text{ W/m}^2$  değerindedir, ancak yeryüzünde  $0-1100 \text{ W/m}^2$  değerleri arasında değişim gösterir. Bu enerjinin dünyaya gelen küçük bir bölümü dahi, insanlığın mevcut enerji tüketiminden kat kat fazladır. Güneş enerjisinden yararlanma konusundaki çalışmalar özellikle 1970'lerden sonra hız kazanmış, güneş enerjisi sistemleri teknolojik olarak ilerleme ve maliyet bakımından düşme göstermiş, çevresel olarak temiz bir enerji kaynağı olarak kendini kabul ettirmiştir.

Güneşe göre dik açı oluşturmasından dolayı Türkiye'nin de içinde bulunduğu, ekvatora yakın coğrafyaları kapsayan güneş kuşağında, güneş enerjisinden faydalanma daha verimli olabilmektedir. Güneş yılın çoğu zamanında olağanüstü boyutlarda enerji üretmektedir. Örnek vermek gerekirse güneş saniyede  $1 \times 10^{20}$  kilowatsaat'lik enerji sağlar ve bir kilowatsaatlik elektrik enerjisinin 1000 Watt'lık bir lambayı bir saat boyunca çalıştırabildiği düşünüldüğünde bu enerjinin büyüklüğü daha iyi anlaşılır. Güneş ışması sıfır maliyetle dünyada kullanılan enerjinin binlerce kat fazla enerji sağlar. Bu yüksek enerjiye rağmen bu günkü teknolojimiz ile güneş enerjisinden elektrik üretiminde %12-%19 arası verim sağlayan güneş pilleri geliştirilebilmiştir.

Sürdürülebilir enerji kaynaklarının verimini genelde enerji sağlanan kaynak belirler. Diğer sürdürülebilir enerji kaynaklarına kıyasla güneş enerjisinin verimini belirleyen birçok harici etken vardır. Bu etkenlerin bir kısmını doğal koşullar oluşturur. Doğal koşullardan ilk akla geleni güneşin mevsimsel durumudur. Bunun yanı sıra havanın bulutlu, yağışlı ya da çok sıcak olması da verimi olumsuz yönde etkiler. Gün içerisinde belirli bir alana düşen güneş enerjisi miktarı bölgenin yeryüzü üzerindeki enlem değeri, yerel iklim koşulları, yılın hangi mevsiminde bulunduğu ve yerleştirilen güneş panelinin güneşle olan açısı gibi birçok faktöre bağlıdır. Örneğin yatay bir

yüze düşen yıllık ortalama ısıma miktarı Orta Avrupa, Orta Asya ve Kanada'da ortalama 1000 kWh/m<sup>2</sup>, Akdeniz civarındaki bölgelerde 1700 kWh/m<sup>2</sup> ve Afrika'nın Ekvatora yakın bölgeleriyle, Doğu Asya ve Avustralya çöllerinde 2200 kWh/m<sup>2</sup>'a kadar çıkmaktadır. Kısaca, güneş enerjisi uygulamalarında mevsimsel ve coğrafi faktörlerin hepsi önemli rol oynar. Tablo 2.1'de kıtalara göre güneşlenme miktarları verilmiştir. [1]

Tablo 2.1 Kıtalar bazında güneşlenme miktarları

Değişik bölgelerin güneşlenme oranları kWh/m <sup>2</sup> gün cinsinden değerleri (güneye 30 derecelik dikey açıyla)				
	Güney Avrupa	Orta Avrupa	Kuzey Avrupa	Karayipler
Ocak	2,6	1,7	0,8	5,1
Şubat	3,9	3,2	1,5	5,6
Mart	4,6	3,6	2,6	6,0
Nisan	5,9	4,7	3,4	6,2
Mayıs	6,3	5,3	4,2	6,1
Haziran	6,9	5,9	5,0	5,9
Temmuz	7,5	6,0	4,4	6,0
Ağustos	6,6	5,3	4,0	6,1
Eylül	5,5	4,4	3,3	5,7
Ekim	4,5	3,3	2,1	5,3
Kasım	3,0	2,1	1,2	5,1
Aralık	2,7	1,7	0,8	4,8
Yıl Ort.	5,0	3,9	2,8	5,7

Türkiye güneş kuşağında bulunan bir ülkedir. Bu coğrafi konumu nedeniyle sahip olduğu güneş enerjisi potansiyeli açısından birçok ülkeye göre şanslı durumdadır. Türkiye'de toplanabilecek güneş enerjisi miktarının ortalama yıllık toplam süresi 2640 saat (günlük toplam 7,2 saat), ortalama toplam ışınım şiddeti 1311 kWh/m<sup>2</sup>-yıl (günlük toplam 3,6 kWh/m<sup>2</sup>) olduğu tespit edilmiştir. Aylara göre Türkiye güneş enerji potansiyeli ve güneşlenme süresi değerleri ise tablo 2.2'de verilmiştir [1].

Tablo 2.2: Türkiye'deki güneşlenme oranları

<b>Türkiye'nin Aylık Ortalama Güneş Enerjisi Potansiyeli</b>			
<b>AYLAR</b>	<b>AYLIK TOPLAM GÜNEŞ ENERJİSİ</b>		<b>GÜNEŞLENME SÜRESİ (Saat/ay)</b>
	<b>(Kcal/cm<sup>2</sup>-ay)</b>	<b>(kWh/m<sup>2</sup>-ay)</b>	
OCAK	4,45	51,75	103,0
ŞUBAT	5,44	63,27	115,0
MART	8,31	96,65	165,0
NİSAN	10,51	122,23	197,0
MAYIS	13,23	153,86	273,0
HAZİRAN	14,51	168,75	325,0
TEMMUZ	15,08	175,38	365,0
AĞUSTOS	13,62	158,40	343,0
EYLÜL	10,60	123,28	280,0
EKİM	7,73	89,90	214,0
KASIM	5,23	60,82	157,0
ARALIK	4,03	46,87	103,0
TOPLAM	112,74	1311	2640
ORTALAMA	308,0 cal/cm <sup>2</sup> -gün	3,6 kWh/m <sup>2</sup> -gün	7,2 saat/gün

Konumsal olarak Türkiye'de en fazla güneş enerjisi alan bölge Güney Doğu Anadolu Bölgesi olup, bunu Akdeniz Bölgesi izlemektedir. Güneş enerjisi potansiyeli ve güneşlenme süresi değerlerinin bölgelere göre dağılımı da tablo 2.3'te verilmiştir[1].

Tablo 2.3: Türkiyenin bölgelerine göre güneş enerjisi potansiyeli

<b>Türkiye'nin Yıllık Toplam Güneş Enerjisi Potansiyelinin Bölgelere Göre Dağılımı</b>		
<b>BÖLGE</b>	<b>TOPLAM GÜNEŞ ENERJİSİ (kWh/m<sup>2</sup>-yıl)</b>	<b>GÜNEŞLENME SÜRESİ (Saat/yıl)</b>
G.DOĞU ANADOLU	1460	2993
AKDENİZ	1390	2956
DOĞU ANADOLU	1365	2664
İÇ ANADOLU	1314	2628
EGE	1304	2738
MARMARA	1168	2409
KARADENİZ	1120	1971

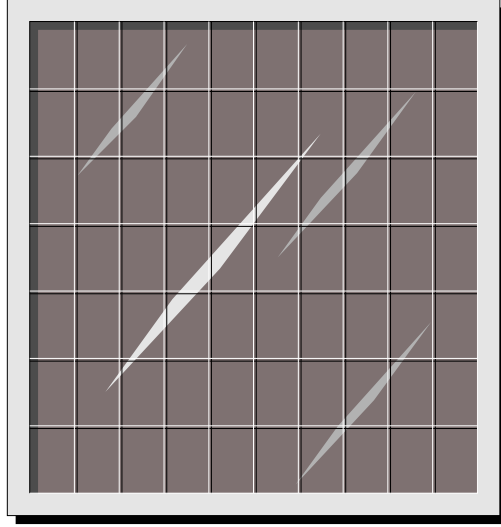


Günümüz teknolojisinde güneş enerjisinden ısı üretimi ve elektrik üretimi olmak üzere iki şekilde faydalanılmaktadır. Güneş enerjisi yenilenebilir kaynaklar içinde kullanımı dünya ülkelerinde en çok yaygınlaşmış kaynaktır. Bunun sebebi güneşin enerjisini dünya geneline yayması ve enerji dönüşümü için büyük ve karmaşık yapılar inşa edilmesi gerekmemesidir. Güneş enerjisinden bireyler kendi imkânları ile faydalanabilme imkânına sahipken, rüzgâr ve jeotermal gibi diğer yenilenebilir temiz enerji kaynaklarından faydalanabilmek için daha çok büyük organizasyonların kurabileceği gelişmiş tesislere ihtiyaç duyulmaktadır.

Ülkemizde hem bölgesel güneşlenme oranlarının hem de enerji maliyetlerinin yüksek oluşu güneş enerjisinin ilk-yatırım masrafının 2-3 sene gibi çok kısa sürelerde çıkmasını sağlayarak, bu yatırımı çok cazip hale getirmektedir. Kuzey Avrupa gibi güneşlenme oranlarının düşük olduğu coğrafyalarda dahi güneş enerjisi sistemleri tek başına sıcak su ihtiyacının % 50-70'ini karşılayabilmektedir. Güney Avrupa'da ise bu oran% 70-90 arasında olmaktadır. [1]

### **2.1. Güneş Pilleri ( Fotovoltaik Piller )**

Güneş pilleri (fotovoltaik piller), yüzeylerine gelen güneş ışığını doğrudan elektrik enerjisine dönüştüren yarıiletken maddelerdir. Yüzeyleri kare, dikdörtgen, daire şeklinde biçimlendirilen güneş pillerinin alanları genellikle 100 cm<sup>2</sup> civarında, kalınlıkları ise 0,2-0,4 mm arasındadır. Bu pillerin seri ve paralel bağlanmaları ile yüksek güce sahip Güneş Panelleri elde edilmektedir. Güneş pilleri fotovoltaik ilkeye dayalı olarak çalışırlar, yani üzerlerine ışık düştüğü zaman uçlarında elektrik gerilimi oluşur. Pilin verdiği elektrik enerjisinin kaynağı, yüzeyine gelen güneş enerjisidir. Güneş enerjisi, güneş pilinin yapısına bağlı olarak % 5 ile % 20 arasında bir verimle elektrik enerjisine çevrilebilir. Güç çıkışını artırmak amacıyla çok sayıda güneş pili birbirine paralel ya da seri bağlanarak bir yüzey üzerine monte edilir, bu yapıya güneş pili modülü ya da fotovoltaik modül adı verilir. Güç talebine bağlı olarak modüller birbirlerine seri ya da paralel bağlanarak bir kaç Watt'tan megaWatt'lara kadar sistem oluşturulur.



Şekil 2.1 Güneş Pili

Güneş pilleri pek çok farklı maddeden yararlanarak üretilebilir.

## 2.2. Güneş Pillerinin Yapısı ve Çalışması

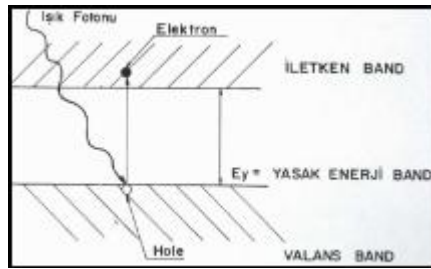
Günümüz elektronik ürünlerinde kullanılan transistörler, doğrultucu diyotlar gibi güneş pilleri de, yarı-iletken maddelerden yapılırlar. Yarı-iletken özellik gösteren birçok madde arasında güneş pili yapmak için en elverişli olanlar, silisyum, galyum arsenit, kadmiyum tellür gibi maddelerdir.

Yarı-iletken maddelerin güneş pili olarak kullanılabilmesi için n ya da p tipi katkılanmaları gereklidir. Katkılama, saf yarıiletken eriyik içerisine istenilen katkı maddelerinin kontrollü olarak eklenmesiyle yapılır. Elde edilen yarı-iletkenin n ya da p tipi olması katkı maddesine bağlıdır. En yaygın güneş pili maddesi olarak kullanılan silisyumdan n tipi silisyum elde etmek için silisyum eriyiğine periyodik cetvelin 5. grubundan bir element, örneğin fosfor eklenir. Silisyum'un dış yörüngesinde 4, fosforun dış yörüngesinde 5 elektron olduğu için, fosforun fazla olan tek elektronu kristal yapıya bir elektron verir. Bu nedenle V. grup elementlerine "verici" ya da "n tipi" katkı maddesi denir.

P tipi silisyum elde etmek için ise, eriyiğe 3. gruptan bir element (alüminyum, indiyum, bor gibi) eklenir. Bu elementlerin son yörüngesinde 3 elektron olduğu için

kristalde bir elektron eksikliği oluşur, bu elektron yokluğuna hol ya da boşluk denir ve pozitif yük taşıdığı varsayılır. Bu tür maddelere de "p tipi" ya da "alıcı" katkı maddeleri denir.

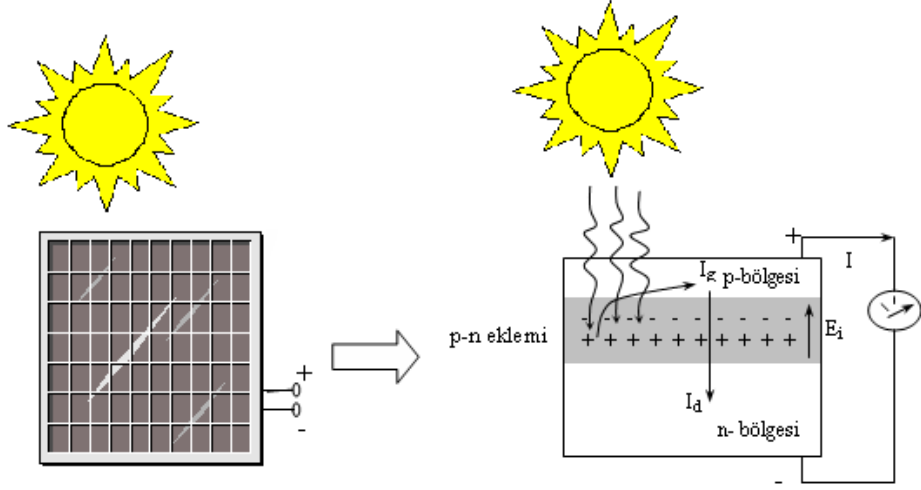
P ya da n tipi ana malzemenin içerisine gerekli katkı maddelerinin katılması ile yarıiletken eklemeler oluşturulur. N tipi yarıiletkende elektronlar, p tipi yarıiletkende holler çoğunluk taşıyıcısıdır. P ve n tipi yarıiletkenler bir araya gelmeden önce, her iki madde de elektriksel bakımdan nötrdür. Yani p tipinde negatif enerji seviyeleri ile hol sayıları eşit, n tipinde pozitif enerji seviyeleri ile elektron sayıları eşittir. PN eklem oluştuğunda, n tipindeki çoğunluk taşıyıcısı olan elektronlar, p tipine doğru akım oluştururlar. Bu olay her iki tarafta da yük dengesi oluşana kadar devam eder. PN tipi maddenin ara yüzeyinde, yani eklem bölgesinde, P bölgesi tarafında negatif, N bölgesi tarafında pozitif yük birikir. Bu eklem bölgesine "geçiş bölgesi" ya da "yükten arındırılmış bölge" denir. Bu bölgede oluşan elektrik alan "yapısal elektrik alan" olarak adlandırılır. Yarıiletken eklemesinin güneş pili olarak çalışması için eklem bölgesinde fotovoltaj dönüşümünün sağlanması gerekir. Bu dönüşüm iki aşamada olur, ilk olarak, eklem bölgesine ışık düşürülerek elektron-hol çiftleri oluşturulur, ikinci olarak ise, bunlar bölgedeki elektrik alan yardımıyla birbirlerinden ayrılır. [2]



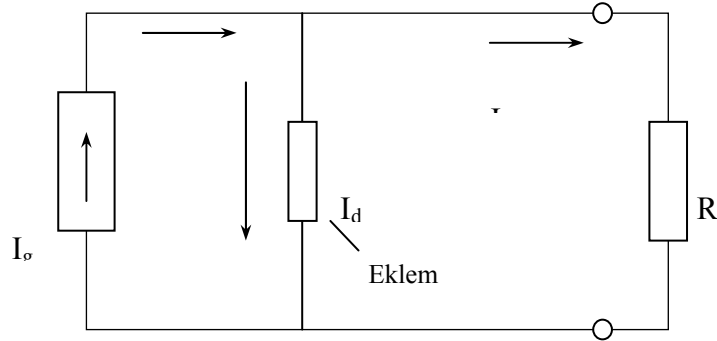
Şekil 2.2 :Güneş pilindeki p ve n maddeleri

Enerji dönüşümü fotovoltaj olaya dayanmaktadır. Fotovoltaj olayda, ışık fotonları özellikle eklem bölgesine ulaştığında elektronlara çarparak serbest yük çiftleri oluşturur. Uyarılan negatif yüklü (-) her elektron, gerisinde pozitif yüklü (+) bir boşluk bırakır. Bu yük taşıyıcıları, eklemle kurulan doğal iç ters elektrik alanla ( $E_i$ ) akım katkısı oluşturmak üzere çoğunlukta oldukları bölgelere sürülür. Doğal  $E_i$  alanı, fotonla enerji kazanan yük taşıyıcılarının hareketlerinde hangi tarafa eğilimli olduklarını ifade eden bir enerji engeli olarak düşünülebilir. Böylece fotonlarla

üretilem (-) yüklü elektronlar n-bölgesinde, (+) yüklü pozitif taşıyıcılar p-bölgesinde toplanarak bir gerilim üretilemektedir[4-9]. Bu durum prensip olarak Şekil2.3 (a) ile verilmektedir.



a)



b)

Şekil 2.3 : a) PV'deki fotovoltajik enerji dönüşümü

b) PV prensip eşdeğer devresi

Işık uyarımı altındaki bir PV, ışınım şiddetine bağılı olarak daha büyük ters sızıntı akımı ( $I_g$ ) üretmektedir. Üretilen  $I_g$  akımı PV uçlarında bir gerilim oluşturmaktadır. Diğer yandan üretilen gerilim, p-n eklemine ileri yönde uyardığından  $I_g$  akımına ters yönlü bir  $I_d$  diyot akımına da neden olmaktadır. Dış devreye aktarılabilecek  $I$  akımı bu iki akımın farkı olmaktadır. Enerji dönüşüm sürecine uygun bir prensip eşdeğer devre (Şekil 2.3.b) ile tanımlanmaktadır.

Bir PV'nin akım-gerilim (I-V) karakteristiği denklem 2.1 ile ifade edilmektedir .[7, 10, 8, 9]

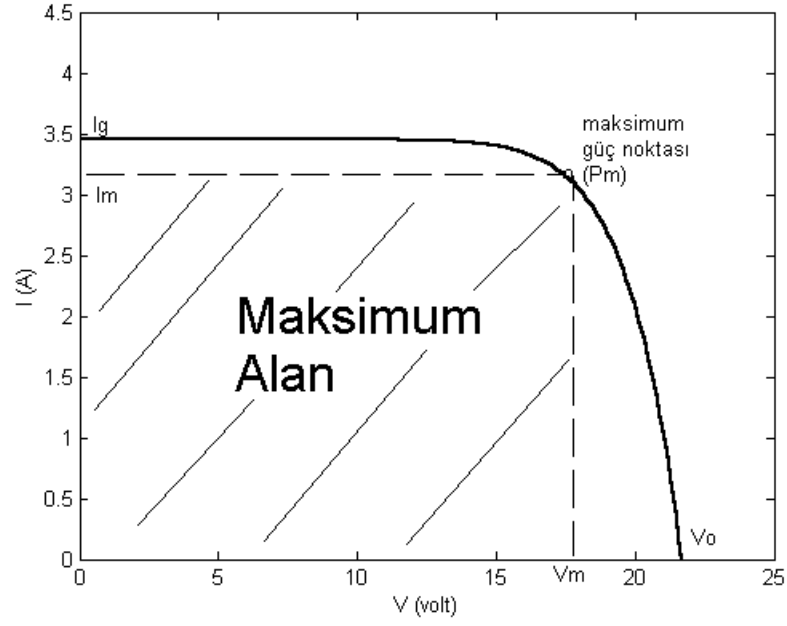
$$I = I_g - I_d = I_g - I_o \cdot \left( e^{\frac{qV}{nkT}} - 1 \right) \quad (2.1)$$

Burada,  $I_g$  ışık fotonlarıyla üretilen akım veya kısa devre akımı (A),  $I_o$  ters sızıntı akımı (A),  $I_d$  eklemdeki diyot akımı (A),  $q$  elektron yükü ( $1,6 \cdot 10^{-19}$  C.),  $k$  Boltzman sabiti ( $1,38 \cdot 10^{-23}$ ),  $T$  mutlak sıcaklık ( $K^0$ ),  $n$  diyot faktörüdür. Denklem 2.1'e göre I-V değişimi doğrusal olmayan bir karakteristiğe sahiptir. Bu yüzden herhangi bir yük direnci için, denklemde ( $V=I.R$ ) bağıntısı düşünüldüğünde I veya V terimi doğrudan çözülememekte ve bir iterasyon çözümü gerekmektedir.

Denklem 2.1'deki  $V=0$ 'da PV kısa devre akımı,  $I=0$ 'da PV açık devre gerilimi ( $V_o$ ) tanımlanabilmektedir. Açık devre gerilimi denklem 2.2 ile verilmektedir.

$$V_o = \left( \frac{nkT}{q} \right) \cdot \ln \left[ \frac{I_g + I_o}{I_o} \right] \quad (2.2)$$

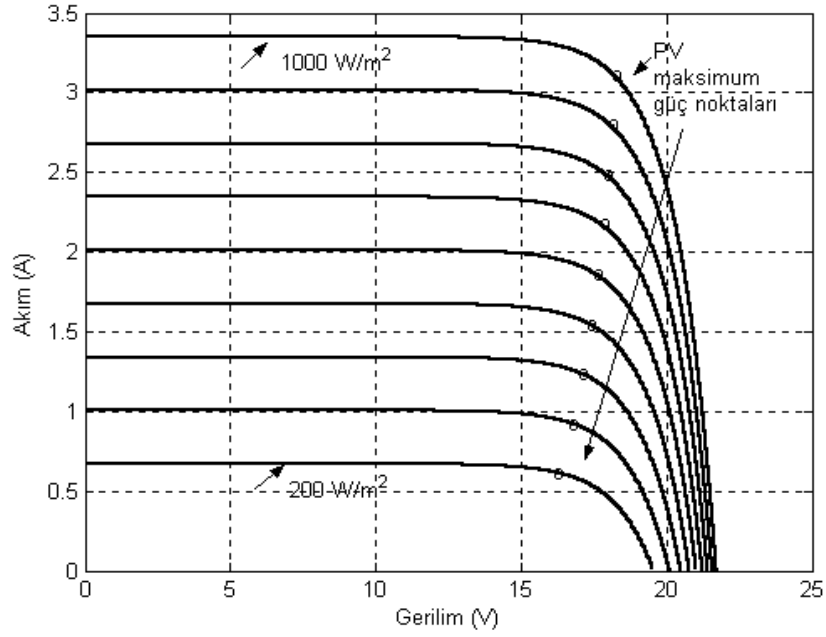
Şekil 2.4, tipik bir PV'nin herhangi bir sıcaklık ve ışınımdaki karakteristik eğrisi olan I-V eğrisidir. I-V eğrisinin altındaki maksimum dikdörtgen alanı PV'nin o anda üertmiş olduğu maksimum gücü ( $P_m$ ) verir ve bu güçteki akım  $I_m$ , gerilim ise  $V_m$  ile gösterilmektedir. Bir PV'nin dolgu faktörü (FF) ise denklem 2.3 ile tanımlanmaktadır.



Şekil 2.4: PV'nin I-V eğrisi

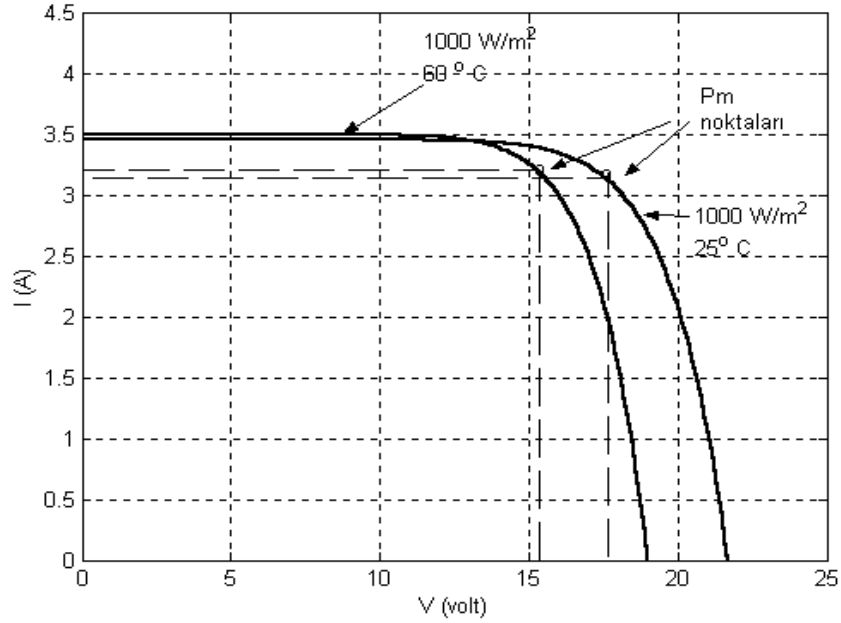
$$FF = \frac{V_m \cdot I_m}{V_o \cdot I_g} \quad (2.3)$$

Bir PV'nin I-V karakteristiği ışınım şiddeti ( $W/m^2$ ) ile değişmektedir (Şekil 2.5). Böylece maksimum güç noktaları da ışınım ile değişmektedir.[3] Bu durumda ışınım şiddeti pilin ürettiği kısa devre akımını doğrudan etkilemektedir. Açık devre gerilimi ise kısa devre akımına oranla daha düşük bir oranda değişmektedir. Değişik ışınım şiddeti altında çalışan bir fotovoltaik pilin performansı fotovoltaik güç sistem tasarımlarında göz ardı edilemeyecek bir öneme sahiptir. [11] Değişken ışınımlardaki PV performans tanımlamaları, PV güç sistem tasarımlarında önemli bir yer teşkil etmektedir.



Şekil 2.5: Farklı ışınım şiddetlerinde I-V eğrileri

Fotovoltaik pillerde ışınım şiddetinin etkisi yanında diğer bir önemli etki ise sıcaklıktır. Belirli bir sıcaklıkta, ışınım ile fotovoltaik pilin kısa devre akımı doğru orantılı olarak artar. Fotovoltaik pillerin çalışma performansına sıcaklık da etki etmektedir. Sabit bir ışınım altında, sıcaklık artışı kısa devre akımının artması yönünde bir etkide bulunurken açık devre geriliminin azalmasına neden olmaktadır[11]. Her ne kadar denklem 2.2'ye göre  $T$  sıcaklığı ile  $I_g$  'nin,  $V_o$ 'ı yükseltebileceği düşünülse de gerçekte sıcaklık,  $I_o$  sızıntı akımını üstel olarak değiştirmekte ve desteklemektedir[6]. Sıcaklığın PV'ye olan etkisi, Şekil 2.6'de prensip olarak verilmektedir.



Şekil 2.6: Sıcaklık değişiminin PV'deki etkisi

### 2.3 PV Model Çalışmaları

PV sistemler verim ve performans açısından uygun bir şekilde tasarlanmalıdır. Çünkü PV'lerin genel verimleri düşük ve performansları güneş ışınımı ile fazlasıyla değişmektedir. PV Sistemleri iyi uygun bir şekilde tasarlanmadığında düşük ışınım şiddetlerinde performansları çok düşmekte veya işlevselliklerini kaybetmektedirler. İstenilen gücü elde edebilmek için çok sayıda PV hücrelerinin bir araya getirildiği düşünülürse, gereksiz maliyet artışları da söz konusu olabilir.

Uygulama alanı bakılmaksızın sistem tasarımı aşamasında öncelikle yapılması gereken PV'lerin performans modelinin tanımlanması önemli bir konudur. Yapısı gereği tipik bir PV'nin ürettiği çıkış gerilimi yükün çektiği akımla doğrusal olmayan bir karakteristik göstermektedir. Işınım şiddeti ve sıcaklık PV çıkış karakteristiğini fazlasıyla etkilemekte ve PV'nin maksimum güç noktaları değişmektedir. Böylece değişen sıcaklık ve ışınım koşullarında PV cevabının tahmin edilmesi gerekmektedir. Genellikle PV'nin performans modeli tek diyot denklemine dayanan bir eşdeğer devre ile tanımlanmaktadır[3-9]. Ancak bu model, PV yapısına bağlı değişen ters sızıntı akımı, diyot faktörü ve iç dirençler gibi bazı özel parametrelerin bilinmesini



gerektirmektedir. Bu parametrelere ulaşmak çoğu kez mümkün olamamaktadır. Ticari amaçlı üretilen PV etiketlerinde özel parametre değerleri bulunmamaktadır. Bu amaçla, deneysel verilerden PV eşdeğer devre modelinin tanımlanması için özel çalışmalar yapılmaktadır [3]. Ayrıca, PV'nin deneysel verilerine göre uygun fonksiyon tanımlamalarıyla (curve-fitting) da yaklaşık performans modeli tanımlanmaktadır [12,13]. Bu tür model yaklaşımları, eldeki bir PV üzerinde örneklenen ışınımlar altındaki deneysel çalışmalarla elde edilen ölçümler ve sonrasında yapılan detaylı analizlerle gerçekleştirilmektedir[11].

### **3. FPGA İLE PV SİSTEM DENETİMİ VE PV MODELİ**

Bu bölümde, tez çalışmasının önemli ve başta gelen elemanı olan FPGA ile PV sistemlerinin güç denetimlerine yönelik çalışmalara ve (PV)'nin elektriksel performans tanımlamalarına yönelik modele yer verilmektedir.

#### **3.1. FPGA ile PV Sistemlerinin Güç Denetimi**

PV'nin üretmiş olduğu gücün maksimum değerini yüke aktarılabilmesi için ışınım şiddeti ve sıcaklık değişimi gibi faktörleri dikkate alacak ve sistemi buna göre adapte edecek kontrol mekanizmalara ihtiyaç duyulmaktadır. PV tabanlı bir sistemin performansı; verilen herhangi bir yükü sürebilecek maksimum noktadaki optimal çalışma noktasını belirleyebilme yeteneğine kuvvetli bir şekilde bağlıdır[14]. Belli sıcaklık ve ışık yoğunluğu altında sadece tek bir maksimum güç noktası mevcuttur. Bu yüzden, PV'nin maksimum güç noktası izleyicisi (MPPT) istenilen sistem verimi kadar önemlidir.

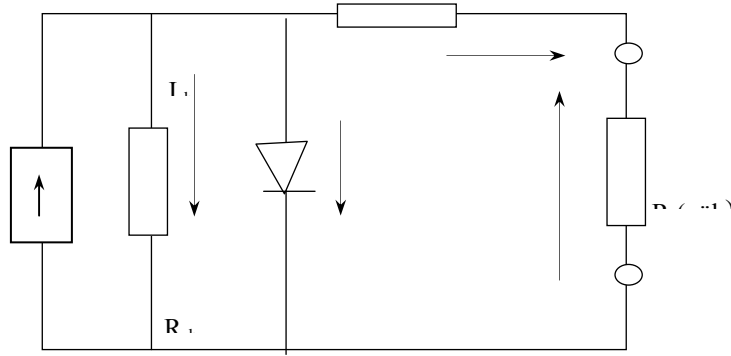
Son yıllarda sayısal işlemci üzerinde gerçekleştirilen birçok MPPT teknikleri önerilmiştir. Bunlar geleneksel metotlarla, örneğin Perturbation and Observation metodu [14] ve sayısal sinyal işlemcisi(DSP) ve RISC mikro denetleyici ile Tepe-Tırmanma metodu önerilmiştir [14]. Bunun yanında daha fazla karmaşık bir yapıya sahip olan yapay zeka metotları RISC Mikro denetleyicilerle Bulanık Mantık denetleyici(FLC) denenmiştir[14]. Farklı bir mikrodenetleyici olarak PIC ile Yapay Sinir Ağları (ANN) yöntemi ile MPPT gerçekleştirilmesi yapılmıştır [14].

Geleneksel yöntemler FPGA ya nispeten daha zayıf bir performans göstermelerine karşın uygulanmaları daha kolaydır. Diğer taraftan yapay zeka metotları daha iyi bir performans göstermelerine karşın yapıları çok daha karmaşık ve nispeten daha yüksek performanslı işlemcilere ihtiyaç duyarlar.

Son zamanlarda, sayısal devrelerin maliyetlerindeki düşüş ve performanslarındaki artış, bu tür sayısal devrelerin güç dönüştürme kontrolü uygulamalarına olanak tanımaktadır[14]. Diğer sayısal sinyal işlemcilerle karşılaştırıldığında, FPGA tabanlı sistemler bir mikrodenetleyici gibi dizinsel makinelerden daha fazla sayısal işlemi aynı zaman içerisinde yapabilmek gibi avantajlar sağlamaktadır[14]. Üstelik eş zamanlı çalışmayla bir DSP den daha hızlı bir şekilde ve sürekli olarak yürütmektedir. Böylece, FPGA cihaz boyutlarını düşürmek için yüksek hızlı anahtarlama devrelerinde uygulanmaktadır[14].

### 3.2 Referans Model (Tek Diyot Modeli)

Bir güneş pilinin(hücre, modül, panel) akım –gerilim(I-V) karakteristiği ve bu eğriden çıkarılabilecek güç tanımlamaları için, yaygın olarak tek diyot (single exponent equation) modeli kullanılmaktadır [3-8, 13] Bu modelin dayandığı eşdeğer devre prensip olarak Şekil 3.1’dedir. Eşdeğer devrede PV, ışığa göre değişen bir akım kaynağı gibi davranmaktadır. I-V ilişkisi ise denklem 3.1 ile verilmektedir.



Şekil 3.1: PV eşdeğer devresi.

$$I = I_g - I_d - I_{sh} = I_g - I_o \left( e^{\frac{q(V+I R_s)}{nkT}} - 1 \right) - \frac{V + I R_s}{R_{sh}} \quad (3.1)$$

Burada,

$I_g$  : Işık fotonlarıyla üretilen akım veya kısa devre akımı (A)

$I_o$  :Ters sızıntı akımı (A)

$I_d$  : Eklemdeki diyot akımı (A)

$I_{sh}$  : Paralel kol direncinden geçen akım (A)

$R_{sh}$  : Paralel kol direnci (ohm)

$R_s$  : Seri direnç (ohm)

$q$  : Elektron yükü ( $1,6 \cdot 10^{-19}$  c.)

$k$  : Boltzman sabiti ( $1,38 \cdot 10^{-23}$ )

$T$  : Mutlak sıcaklık ( $K^0$ )

$n$  : Diyot faktörü

$I$  : Yük akımı (A)

$V$  : Çıkış gerilimi (V)

$R$  : Yük direnci (ohm)

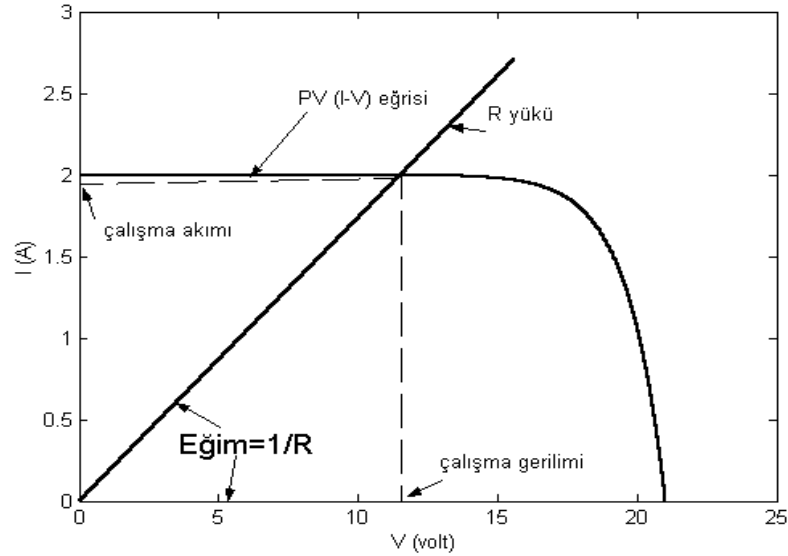
Farklı ışınımlardaki analiz için, STC'deki  $I_g$  terimi, ilgili ışınımın  $1000 \text{ W/m}^2$  'ye olan oranıyla(s) çarpılmaktadır.( $1000 \text{ W/m}^2$  için  $s=1$  birim(br)). Denklem 3.1.'de  $R_s$  ve  $R_{sh}$  parametrelerinin etkisi çoğu zaman ihmal edilerek analiz kolaylığı için denklem basitleştirilmektedir. Bu çerçevede oluşan gerilim ifadesi denklem 3.2'dedir.

$$V = \left( \frac{nkT}{q} \right) \cdot \ln \left[ \frac{I_g \cdot s - I}{I_o} + 1 \right] \quad (3.2)$$

Denklem 3.2 kullanılarak herhangi bir ışınımdaki I-V eğrisi için, I terimi (0-I<sub>g</sub>.s) aralığında belirli adımlar halinde alınmaktadır. Her adım için V değerleri hesaplanır. V=0'da, I kısa devre akımıdır (I=I<sub>g</sub>.s). I=0 'daki V değeri açık devre gerilimini (V<sub>o</sub>) vermektedir.

$$V_o = \left( \frac{nkT}{q} \right) \cdot \ln \left[ \frac{I_{g.s}}{I_o} + 1 \right] \quad (3.3)$$

PV'nin herhangi bir R direnç yükündeki I-V değer çifti, denklem 3.2 için yapılan iterasyonda R=V/I ifadesinin sağlanmasıyla gerçekleşir. Diğer bir deyişle bir R yükündeki PV'nin akım ve gerilimi, eğimi 1/R olan ve orjinden geçen doğrunun, PV'nin I-V eğriyle kesiştiği noktanın koordinatlarıyla tanımlanır.(Şekil 3.2). Bu koordinatlardan, gerilimin yatay ekseninde, akımın ise düşey ekseninde olduğu ön görülmektedir. PV cevabı için, yapılan bu analitik yaklaşım, tamamen denklem 3.2'nin doğrusal olmayan karakteristik özelliğinden kaynaklanmaktadır. Ayrıca bu karakteristik ışınımla da değişmektedir.



Şekil 3.2: R yükünün PV'deki çalışma noktaları.

Herhangi bir ışınım ve yükteki PV gücü (P) ise denklem 3.4 ile tanımlanmaktadır. Maksimum güç ( $P_m$ ) ise oluşan I-V eğrisinin altına sığabilen en büyük dikdörtgen alanıdır ve denklem 3.5 ile verilmektedir.  $P_m$ 'yi veren gerilim  $V_m$  ile, akım ise  $I_m$  ile gösterilmektedir.

$$P = V.I \quad (3.4)$$

$$P_m = V_m \cdot I_m \quad (3.5)$$

Tek diyot modeliyle I-V eğrisini belirlemek için ilgili güneş piline ait özel  $I_0$  ve n parametrelerinin bilinmesi gerekmektedir. Bu parametreler genelde üretici firma kataloglarında yoktur.  $I_0$  sızıntı akımı hassas bir ölçüm olanakları varsa da ölçülebilir. Ayrıca değişen sıcaklıklara göre,  $I_0$  teriminin yeni değerleri, analizde göz önünde tutulmalıdır. Çünkü  $I_0$ , sıcaklıkla üstel olarak değişmektedir [3].

$$I_0 \approx e^{\frac{-E_g}{kT}} \quad (3.6)$$

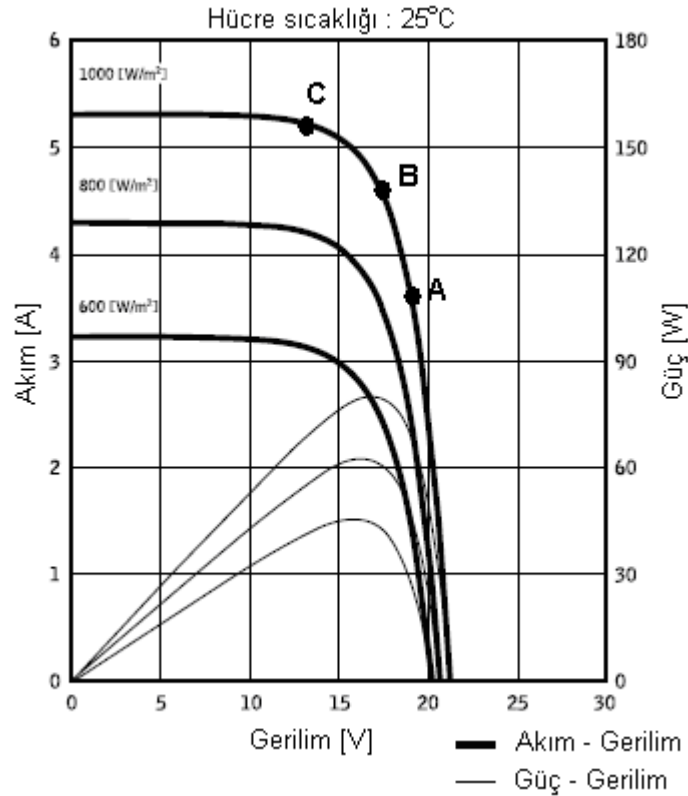
Denklem 3.6'da  $E_g$  terimi, PV yarı iletkeninin enerji aralığını (bant genişliği) göstermektedir.

Parametreleri tek diyot modeline göre tanımlanmış,  $N_s$  kadar seri ve buna  $N_p$  kadar paralel bağlı paket PV'lerden oluşan sistemin akım-gerilim ( $I_p$ - $V_p$ ) ilişkisi denklem 3.7 ile verilmektedir.

$$I_p = N_p \cdot I_g \cdot s - N_p \cdot I_0 \left( e^{\frac{qV_p}{N_s nkT}} - 1 \right) \quad (3.7)$$

#### 4. MAKSİMUM GÜÇ NOKTASI İZLEYİCİ

Güneş panelini kaynak olarak kullanan elektrikli sistemlerde, panellerin yapısından dolayı çekilen akıma göre çıkış geriliminde doğrusal olmayan bir değişim oluşur. Bu değişim Şekil 4.1’de görülmektedir. Panellerin çıkış gerilimi, çekilen akıma bağlı olduğu gibi, panellerin üzerinde düşen güneş ışığı yoğunluğuna ve ortam ısısına bağlı olarak da farklılık göstermektedir. Ticari güneş panellerinin anılan çıkış gücü, birim alana düşen güneş ışığı yoğunluğunun  $1000\text{W/m}^2$  olduğu ortamda, panelin üretebileceği elektrik enerjisi olarak tespit edilmiştir. Anılan çıkış gücünün panelin verebileceği maksimum güç olduğu varsayılır. Güneş ışığı yoğunluğu düştükçe güneş panelinden çekilebilecek akımında azaldığı Şekil 4.1’deki eğrilerde görülmektedir.



Şekil 4.1: Güneş paneli akım-gerilim ve güç-gerilim eğrisi

Şekil 4.1’de  $1000\text{W}/\text{m}^2$  güneş ışığının olduğu bir ortamda çekilen akım  $3.8\text{A}$  iken gerilimin  $19\text{V}$  olduğu görülmektedir (A noktası). Bu durumda A noktasındaki üretilen güç  $3.8\text{A} \times 19\text{V} = 72\text{W}$ ’dır. Yine aynı güneş ışığı ortamında ( $1000\text{W}/\text{m}^2$ ) güneş panelinden daha fazla akım çektiğimizi varsayarak, çekilen akımın  $4.6\text{A}$  olduğunu düşünelim. Bu durumda eğriden çıkış geriliminin  $17\text{V}$ ’a düştüğünü görürüz. (B noktası) Yeni akım ve gerilim değerlerine göre üretilen güç  $4.6\text{A} \times 17\text{V} = 78.2\text{W}$  olmuştur. Aynı durumu C noktası içinde düşünecek olursak, güneş panelinden C noktasında  $5.1\text{A}$  çekilmesine karşılık çıkış geriliminin  $14\text{V}$ ’a düştüğü görülür. Bu durumda C noktasındaki üretilen çıkış gücü  $5.1\text{A} \times 13\text{V} = 66.3\text{W}$ ’a düşmüştür.

Bu değerlerden de anlaşıldığı gibi aynı güneşlenme ortamında, yükün giderek artmasına karşın güneş panelinden alınabilecek güç miktarı doğrusal olmayan bir şekilde farklılık göstermiştir. Bir noktaya kadar üretilen güç yüke bağlı olarak artmış fakat bir noktadan sonra yükün artmasına rağmen üretilen güçte azalma olmuştur. Eğer bu durum kontrol altında tutulmazsa sistem düşük bir verimle çalışacaktır. Maksimum güç noktası takip edici adı verilen sistemler değişim gösterebilen bu çıkış gücünü hep en üst tepe noktasında tutmaya çalışırlar. MPPT devreleri basit bir akım sınırlayıcıdan çok daha öte bir tasarım alt yapısını içermektedirler. Çünkü güneş panellerindeki maksimum güç noktası panelin üzerinde düşen güneş ışınlarının enerjisine göre yer değiştirebilmektedir. Güneş panelindeki bu güç noktasının yeri güneşli bir gündeyken bile değişmektedir. Güç izleyici ise her türlü durumda etkin güç noktasını hemen tespit edip enerji kullanım verimini en üstte tutmaya çalışacaktır [2].

#### **4.1 Tasarım kriterleri**

Maksimum güç noktası izleyicisi bazı hedefleri sağlaması gerekir. Bunlar ; Çevresel koşulların değişmesiyle maksimum güç noktası değiştiğinde sistemin bu noktaya yakın çalışması gerektiği. Yüksek dönüşüm verimliliği sağlaması gerekir. Çevresel koşulların geniş bir ölçekte değişimine karşın izlemeyi düzenleyebilmesi Yük ile uyumlu bir şekilde çıkış ara yüzü sağlayabilmesidir. Bir MPPT devresi tasarlamaya başlamadan önce aşağıdaki kriterlerin göz önünde bulundurulması gerekir:



- Elektriksel karakteristiđi: PV nin verdiđi gerilim aralıđı ve bu gerilim aralıđına karřılık ürettiđi akım deđeri.
- Kazanç: Sistemin en önemli kriterlerden birisi de kazanç olmaktadır. DC-DC çeviricilerde, ideal olmayan elemanlar yüzünden hiçbir zaman giriş gücü, çıkış gücüne eşit deđildir. Gerek bobinin omik direncinde, gerek anahtarlama elemanının iç direncinde ve gerekse de diyotun oluşturduđu ileri yönlü gerilim düşümü ve diđer ısı vs. gibi durumlar mutlaka ekstra bir güç sarfiyatı oluşturacaktır. Önemli olan bu sarfiyatın en aza indirilmesidir[2].
- Algoritmanın esnekliđi ve hızı: MPPT devresi, MPPT algoritması görevini yürüten her türlü deđişken giriş gücüne olabildiđince hızlı bir şekilde tepki vererek maksimum güç noktasını hemen tespit edebilmelidir.
- Elverişsiz ortam şartları: Devre, dış çevreden kaynaklanan bozucu etkilere karřı özellikle de yüksek çevre ısısına ve ortamın ışıınım şiddetindeki deđişimlerine karřı etkin olabilmelidir.
- Maliyet: Ticari MPPT devrelerinin ortalama satış fiyatı 1000-2000\$ arasındadır[2]. Bu fiyat MPPT devresinin gücüne göre büyük oranda artış gösterebilmektedir. Üretilecek güç izleyici devrenin maliyeti de bu miktarları geçmemelidir.

## **5. PROGRAMLANABİLİR LOJİK ELEMANLARIN MİMARİSİ VE PROGRAMLAMA TEKNİKLERİ**

Sayısal işlem sistemleri, sayısal verileri işlemek için tasarlanmış, hızlı donanımlara ve bu donanımlara işlevsellik kazandıracak esnek yazılımlara ihtiyaç duyan yapılardır. Sayısal işlem sistemi oluşturmada, donanım ve yazılım tabanlı olmak üzere iki geleneksel yöntem uygulanır.

Donanım tabanlı yöntem: Sayısal verileri işlemek için ağırlıklı olarak özel tüm devreler (ASIC) kullanılır. Bu tip devreler özel bir fonksiyonu gerçekleştirmek amacıyla üretildiğinden, bu fonksiyonları etkin ve hızlı bir şekilde gerçekleştirebilirler. Ancak işlevleri sınırlıdır ve sadece ilgili oldukları uygulamaya yönelik üretilmişlerdir. ASIC devreler doğru ve hızlı sonuç vermesine rağmen çözüm ürettiği problemin çeşitli türevleri için kullanılamayacaklardır. Yeni problemler için yeni donanımlara ve yeni ASIC yapılara ihtiyaç duyulur. Bu da maliyet artışına ve zaman kaybına neden olur[15].

Yazılım tabanlı yöntem: Bu yöntemde tasarım değişikliklerine daha esnek olan mikroişlemciler kullanılır. Mikroişlemcinin koşturduğu yazılımlar değiştirilerek, hiçbir donanım değişikliğine gidilmeden tasarım ortamına yeni fonksiyonlar eklenebilir. Mikroişlemciler üzerinde koşturan, yazılım uygulamaları aynı anda birçok işlemi yerine getirmek için tek bir işlemcinin genel kaynaklarını kullanırken, yavaş fakat esnek yazılımlar ile çalışırlar. Fakat sıradan bir uygulama için komutların bellekten okunması, onların yorumlanması yerine getirilmesi, sistemin performansı ve hızını oldukça düşürmektedir.

Günümüzde sayısal işlem sistemlerinin kullandığı alandaki hızlı gelişmeler, üretim tamamlandıktan sonra da esnek genel amaçlı olacak şekilde tasarlanan işlem sistemlerini ortaya çıkarmıştır. Özellikle iyi tasarlanmış ve işlemci yükünü azaltan, paylaşan donanımlar, performans artışı için iyi bir çözüm olabilir. Bununla beraber, işlevleri uygulama sırasında değiştirilebilen programlanabilir devre elemanlarının

kullanımı da avantaj getirecektir. Bu devre elemanları ile gerçekleştirilen donanımlar, ASIC gibi devre elemanları ile yapılan klasik donanımlara göre daha işlevseldirler[16].

Tekrar düzenlenebilen işlem sistemleri (Reconfigurable Computing System) olarak ta adlandırılan bu sistemler, esnek ve genel amaçlı yapıları sayesinde yeni bir üretim aşamasına ihtiyaç duymadan, değişen protokollere, sistem özelliklerine ve kullanıcı ihtiyaçlarına kısa sürede cevap verebilirler. Yine bu özellikleri sayesinde tekrar düzenlenebilen işlem sistemleri, donanım tabanlı sayısal işlem sistemlerine göre daha esnek ve yazılım tabanlı sayısal işlem sistemlerine göre daha hızlı sayısal tasarım ortamı oluşturarak, bu iki sistem arasındaki boşluğu doldururlar [16].

Tekrar düzenlenebilir sayısal işlem sistemlerinin, ihtiyaç duyduğu esnek donanımlar, Alan Programlamalı Kapı Dizileri(Field Programmable Gate Array, FPGA) kullanılarak karşılanır [15]. Özellikle SRAM (Statik Rastgele Erişimli Bellek) tabanlı FPGA'ler tekrar düzenlenebilirlik kabiliyetleri ve yüksek performanslı uygulamalardaki yeterlilikleri sayesinde, genel amaçlı sayısal donanımların tasarımında anahtar rol oynarlar.

Yakın tarihten itibaren sayısal işlem sistemleri üzerinde FPGA'ler kadar etkili olan başka bir kavram ise HDL'dir (Donanım tanımlama dili ). Donanım tanımlama dillerinin kullanılması system-on-chip (SoC) teknolojisini beraberinde getirmiştir. Sayısal işlem sistemleri tasarımında system-on-chip (SoC) teknolojisi özellikle yer ve enerji sorunlarının yoğun olarak yaşandığı alanlarda kullanılmaktadır.

SoC teknolojisinde, sistem içerisindeki birimler çoğunlukla bir donanım tanımlama dili yardımıyla ifade edilirler. Sistemin tanımlama aşamasını takiben derleme ve davranışsal benzetim adımları gerçekleştirilir. Sistemden beklenen cevapların elde edilmesiyle sistem üzerinde zamansal benzetim aşamasına geçilir. Bütün birimler, sentezleme ve yerleştirme işlemi sonunda tekrar programlanabilir bir tümleşik devreye aktarılır. Sistemin uygulama aşamasında ise gerek büyük kapasiteleri gerekse de esnek yapılarından dolayı FPGA tümleşik devreleri tercih edilmektedir.

Günümüz teknolojisinde FPGA ve VHDL (Very high speed integrated circuit HDL) gittikçe önem kazanmaya başlamışlardır. VHDL ile SoC teknolojisi kullanılmasıyla tanımlanan ve sentezlenen sistemler FPGA ile gerçekleştirilerek gerçek gücü ortaya koymaktadırlar.

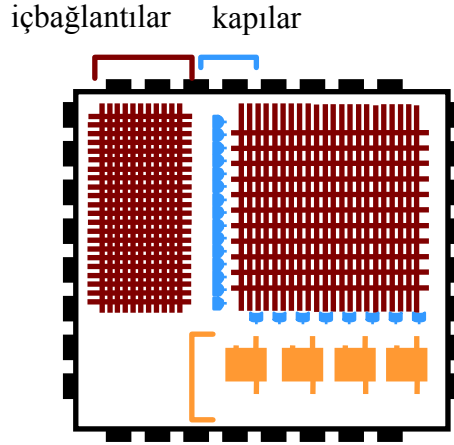
### **5.1 Programlanabilir Lojik Elemanların Gelişimi**

Sayısal tasarımlarda kullanılan ilk programlanabilir lojik devre elemanı salt okunabilir bellek(PROM) elemanıdır. Bu elemanda, belleğin adres girişleri, lojik devrenin girişlerine, adreslenmiş gözdeki bilgiler de, lojik fonksiyonun çıkışlarına karşılık düşer. Genellikle karmaşık lojik fonksiyonlar için verimsizdirler.

Programlanabilir yapıların sonraki türleri PLD'lerdir (Programlanabilir Lojik Devre). PLD'ler; SPLD(Basit Programlanabilir lojik eleman), CPLD(Karmaşık Programlanabilir lojik eleman),MPGA(Maske programlanabilir kapı dizileri), FPGA(Alan Programlamalı kapı dizileri) olmak üzere beş bölümde incelenebilir.

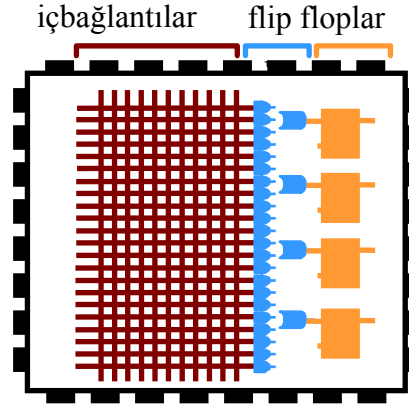
SPLD'ler PLA(Programlanabilir Lojik Dizi) ve PAL (Programlanabilir Dizi Lojiği) olmak üzere iki kısma ayrılırlar.

PLA : PLA mimarisi Şekil 5.1 de görüleceği gibi iki tane programlanabilir düzleme sahiptir. Bu iki programlanabilir düzlem AND (VE) ve OR (VEYA) kapılarının kombinasyonlarıyla oluşmakta ve AND işlemini birçok OR kapısı üzerinde paylaşırma esasına dayanmaktadır. Bu mimari oldukça esnektir fakat iki tane programlanabilir düzleme sahip olması üretim ve yollanma (mapping) gecikmelerini artırmaktadır (Bkz.Şekil 5.1).



Şekil 5.1: PLA lojik devre elemanın iç mimarisi

PAL : Pal mimarisinde ise sadece bir tane programlanabilir düzlem bulunmaktadır. Bu mimaride programlanabilir AND(VE) ve sabit OR(VEYA) matrisleri bulunmaktadır. PAL'ler bu özellikleri ile ucuz ve yüksek hızlı performans sağlamaktadırlar (Bkz.Şekil 5.2).

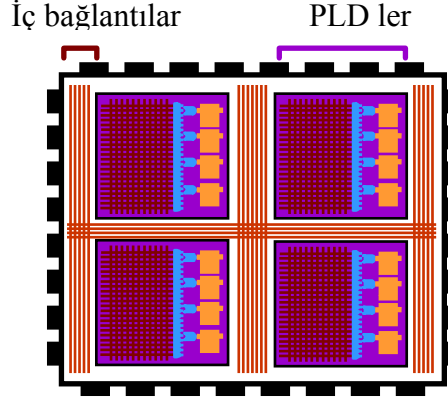


Şekil 5.2: PAL lojik devre elemanın iç mimarisi

PLA ve PAL' ler Basit programlanabilir lojik devreler (SPLD) olarak adlandırılırlar.

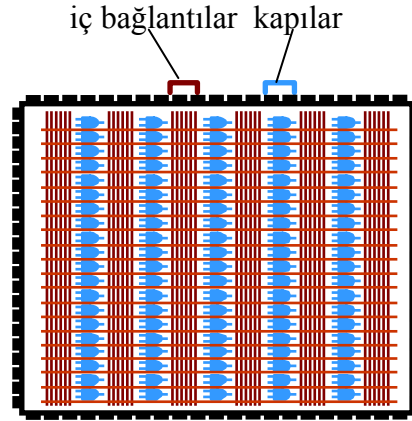
SPLD'lerin sınırlı kapasiteleri daha yüksek kapasiteli programlanabilir devreler olan CPLD'lerin doğmasına neden olmuştur. CPLD'ler, SPLD benzeri birçok bloğun bir araya getirilmesiyle oluşmuş yapılardır.

CPLD sadece bireysel PLD'lerin aynı çip üzerinde toplanmış ve bu bireysel PLD'lerin birbirlerine bağlanmak için arabağlantı yapılarının çip üzerinde düzenlenmiş halidir (Bkz. Şekil 5.3).



Şekil 5.3: CPLD lojik devre elemanın iç mimarisi

Programlanabilir devre elemanlarını gelişim sürecinde CPLD'lerden sonra Maske programlamalı kapı dizileri (Mask Programable Gate Array, MPGA) tanıtılmıştır. MPGA, kullanıcıların lojik devresindeki özelliklere ve bağlantılara göre özel olarak üretilmiş transistor dizilerinden oluşur. Kullanıcının isteği doğrultusunda üretiminden dolayı üretim süreci hem uzun hem de masraflıdır. MPGA ler tam programlanabilir lojik tümdevreler olmasalar da programlanabilir türevleri olan FPGA'ların gelişmesinde ilk aşama olmuşlardır (Bkz. Şekil 5.4).

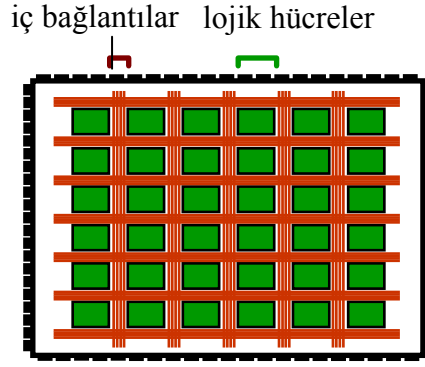


Şekil 5.4: MPGA lojik devre elemanın iç mimarisi

## 5.2 Alan Programlamalı Kapı Dizileri (FPGA)

Alan programlanabilir Kapı Dizinleri (FPGAs), geniş yeteneklere sahip esnek, programlanabilir cihazlardır. Temel yapılarında ayarlanabilir bir bağlantı matrisinde gömülü, evrensel bir dizin, programlanabilir lojik hücreler içermektedir. Hücreler ve bu hücrelerin bağlantıları, tasarım yazılımı ile oluşturulmuş bir veri dosyası kullanılarak, programlanabilir ayarlamalarla kararlaştırılmaktadır. Yazılım tasarımcıya değişik teknolojik soyut düşünceler sunar ve cihaz konfigürasyonu içerisindeki bir tasarım örneğini çevirmek için bu soyut düşünceleri kullanılır.[16 ]

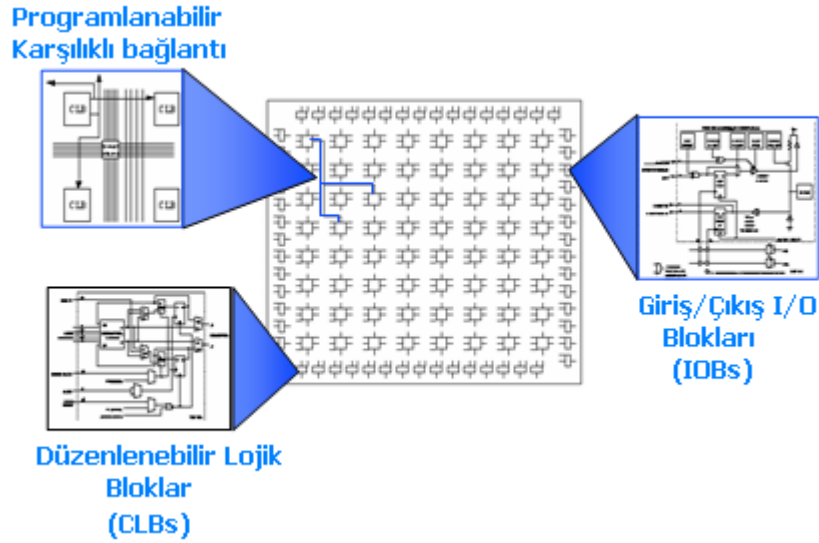
CPLD ve MPGA'ların üstünlüklerini kullanmak ve sakıncalarını ortadan kaldırmak amacıyla Xilinx firması, 1985 yılında, FPGA'leri piyasaya sunmuştur[16]. FPGA'lar programlanabilir lojik bloklar ve ara bağlantılardan oluşur. Kullanıcının tasarladığı devreye göre, FPGA üreticisi tarafından sağlanan bir yazılım sayesinde lojik bloklar ve aralarındaki bağlantılar programlanır. Tasarım sırasında kullanıcıya sağladığı esneklik, düşük maliyet ve hızlı örnek üretme özelliği FPGA'ları sayısal tasarım ortamlarının vazgeçilmezi haline getirmiştir (Bkz. Şekil 5.5).



Şekil 5.5: FPGA mimarisi

### 5.2.1 FPGA'lerin mimarisi

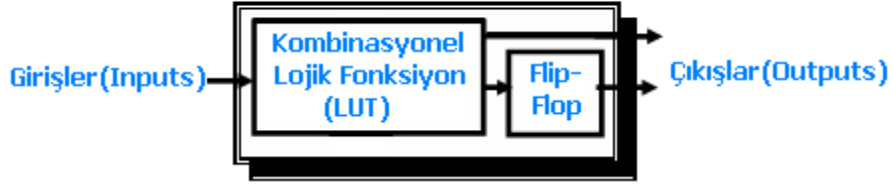
FPGA'lar temelde üç bloktan oluşur, lojik bloklar(CLB), giriş-çıkış blokları(I/O) ve bağlantı blokları.



Şekil 5.6: FPGA'i oluşturan bloklar

**Lojik Bloklar:** Şekil 5.6 ile gösterilen lojik bloklar(CLB) boolean fonksiyonlarının gerçekleştirildiği yapılardır; küçük taneli (fine-grain) ve kaba taneli (coarse-grain) olarak adlandırılan iki sınıfa ayrılırlar. Bu sınıflandırmada; CLB'nin oluşumunda kullanılan transistor sayısı, lojik bloğun gerçekleyebileceği boolean fonksiyon sayısı veya lojik bloğun giriş-çıkış sayısı büyüklük ölçütü olarak kullanılabilir [16].



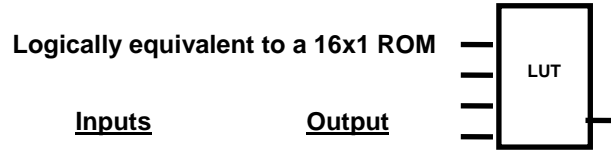


Şekil 5.7: CLB iç yapısı

Şekil 5.7’de iç yapısı verilen CLB, küçük taneli bloklar, genellikle iki girişli bir lojik kapıya veya bir kaç girişli bir çoklayıcıya eşlik eden saklama elemanından oluşur.

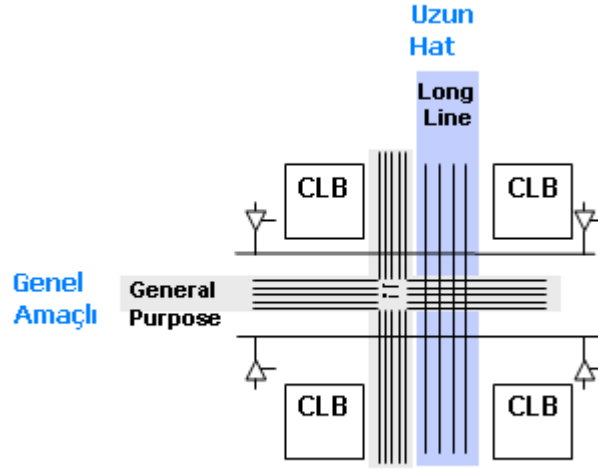
Kaba taneli lojik blokların yapıları çok çeşitlilik göstermelerine karşın yaygın olarak Şekil 5.8 ile gösterilen doğruluk tablosu (LUT) veya çoklayıcı (Multiplexer) gibi daha büyük saklama elemanlarından oluşur.

### Lookup Table



Şekil 5.8: CLB’yi oluşturan LUT elemanı

**Bağlantı Blokları:** Şekil 5.9 da verilen bağlantı blokları lojik bloklarla giriş-çıkış blokları arasındaki bağlantıyı sağlayan yapılardır. Bu yapılar, yönlendirme kanalları ve programlanabilir anahtarlardan oluşur.



Şekil 5.9: bağlantı blokları

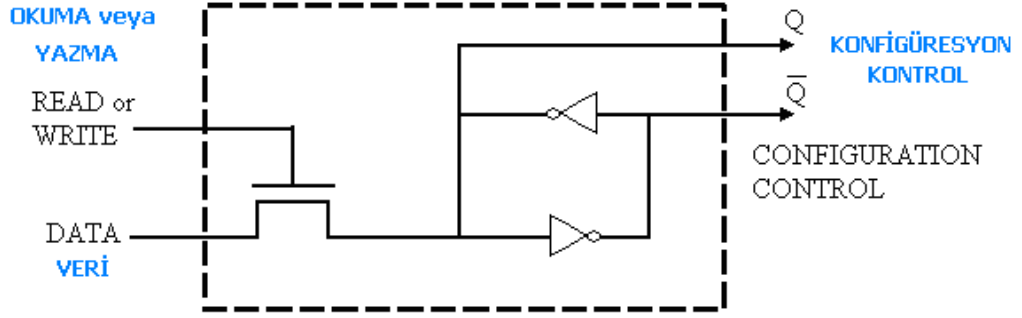
FPGA mimarileri bağlantı kanallarının yapısına göre; Simetrik Dizi Mimarisi (Symmetrical Array), Sıra Tabanlı Mimari (Row-Based Array) ve Kapı Denizi Mimarisi (Sea of Gate) olmak üzere üç ana gruba ayrılır [16].

## 5.2.2 FPGA'lerin programlama teknolojileri

FPGA'ler için üç tip programlama teknolojisi genel olarak kullanılır. Bu üçü de elemanın mimarisinde yansıtılan ilgili alan ve performans etmenlerine bağlı kullanılır. Buna göre FPGA'ler eleman mimarisi ve programlama konfigürasyonuna göre kategorize edilebilir.

### 5.2.2.1 Statik RAM programlama teknolojisi

Bir SRAM programlı FPGA'de, normal işlem boyunca program statik hafıza hücresinde tutulur. Şekil 5.10 Hafıza statik RAM hücrelerinden yapılmıştır ve bunun için çip sık sık "SRAM programlanabilir" olarak ifade edilir.

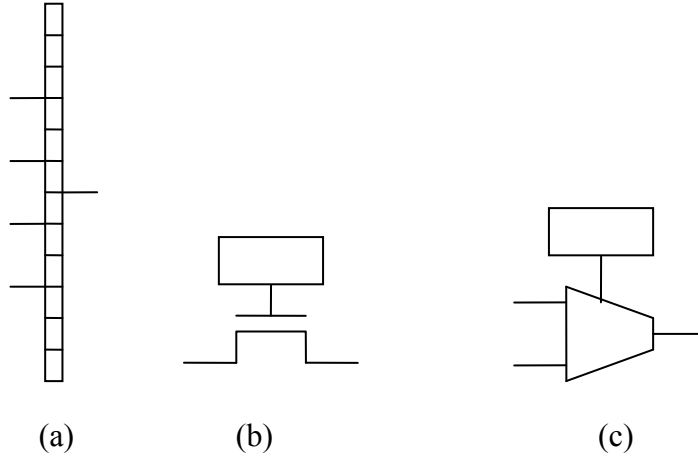


Şekil 5.10: Bir Statik RAM hafıza hücresi

Tümleşik devre üzerinde SRAM hafıza hücreleri için ayrılmış yer yoktur, SRAM hücreleri kontrol ettikleri lojik elemanlar arasında dağıtılmışlardır. SRAM hücreleri ile üç ayrı inşa bloğu kontrol edilir. Look-up tablosu, adres hatlarını kontrol eden fonksiyon girişleriyle hafıza hücrelerinden yapılmış önemli inşa bloklarından biridir. Look-up tablosu Şekil 5.11(a)'da gösterilmiştir. Diğer inşa bloğu programlanabilir ara bağlantı noktası (PIP) diye isimlendirilir ve Şekil 5.11(b)'deki gibidir. PIP bir hafıza hücresinin kontrol ettiği bir geçiş hücresidir. Konfigüre edilebilir bir ara bağlantı olarak kullanılır. Üçüncü inşa bloğu ise Şekil 5.11(c)'de gösterilen bir hafıza hücresi tarafından kontrol edilen multiplexer'dır.

SRAM programlama teknolojisi mimarisinin en büyük dezavantajlarından biri geniş alan kaplamasıdır. Diğerisi ise güç kesilmesiyle SRAM'deki bilgilerin kaybolmasıdır.

Ancak, SRAM programlama teknolojisi iki önemli avantaja sahiptir; sadece standart tümleşik devre teknolojisinin sağladığı hızlı tekrar-programlanabilirlik ve hatta karmaşık lojik devreler için bile düşük güç tüketimidir.



Şekil 5.11. (a)LUT, (b)PIP, (c)Multiplexer

SRAM tabanlı FPGA'ler, tekrar düzenlenebilirlik kabiliyetleri ve yüksek performanslı uygulamadaki yeterlilikleri nedeniyle sayısal işaret işlemede temel devre elemanı olarak yaygın şekilde kullanılmaktadırlar[16]. Ayrıca tasarım tamamlanır tamamlanmaz test etmek mümkün olduğundan uygulamalarda oldukça avantaj sağlamaktadırlar [16].

### 5.2.2.2 Antisigorta-tabanlı programlama teknolojisi

Bu teknolojiye, FPGA programlanmadan önce, yollanma kanalları arasındaki bağlantılar kurulmamış durumdadır. Programlama sırasında uygulanan gerilimle gerekli bağlantılar oluşturulmuş olur. Böylece kullanılan bağlantı sayısının kullanılmayan bağlantı sayısından daha az olması sebebiyle programlama süresi kısalmıştır [16].

Programlama için gerekli gerilimin entegre içine dağıtılmasını sağlayan transistörler yarıiletken yonga üzerinde geniş alan kaplamasına karşın, diğer teknolojilerle karşılaştırıldıklarında antisigorta'lar daha küçük bir alana ihtiyaç duyarlar. Bu teknolojiyi kullanan FPGA'ler bir kez programlanabilme özelliğine sahip olduklarından ilk örnek üretim için pahalı bir çözüm olmaktadır.

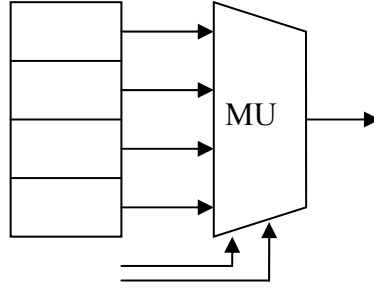
### 5.2.2.3 EPROM ve EEPROM programlama teknolojisi

EPROM ve EEPROM programlamada kullanılan yapı EROM belleklerde kullanılan yapıya benzemektedir. Bu teknolojiyi kullanan FPGA'ler tekrar programlanabilir. Ancak, bu tip yapıları programlamak için özel devreciklere ihtiyac duyulur bu sebeple devre üzerinde programlanamazlar.

### 5.2.3 FPGA'ların lojik hücre mimarisi

#### 5.2.3.1 Doğruluk tablosu tabanlı yapı

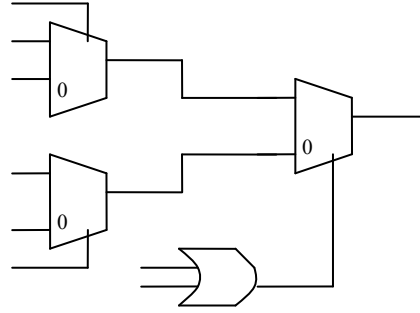
Bu yapının temel bloğu, LUT(Look Up Table) adı verilen ve  $m(m \geq 2)$  deęişkenli her boolean foksiyonunu gerçekleyebilen bir yapıdır (Bkz.Şekil 5.12).



Şekil 5.12. İki girişli LUT yapısı

#### 5.2.3.2 Çoklayıcı tabanlı yapı

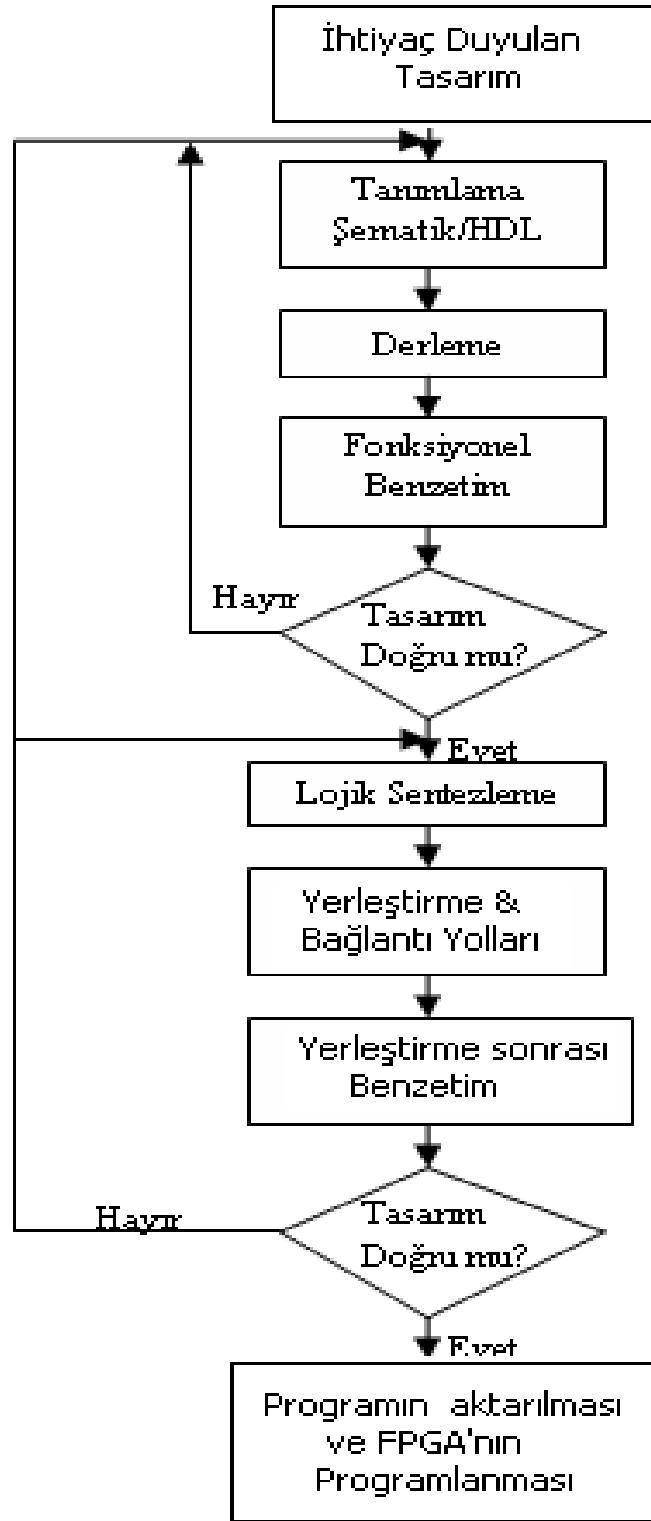
Bu yapının temel bloğu çeşitli konfigürasyonlardan ve olabildiğince az VE ve VEYA gibi lojik kapılardan oluşur. Şekil 5.13



Şekil 5.13. Çoklayıcı tabanlı lojik hücre yapısı

### 5.3 FPGA Kullanılarak Gerçekleştirilen Devrelerin Tasarım Süreci

Tasarım süreci, gerçekleştirilecek devre fonksiyonlarının, sözle veya şematik olarak tanımlanması ile başlar. Sözde tasarımda genellikle yüksek seviyeli donanım tasarım dilleri (Hardware Description Language, HDL) kullanılır. Şematik tanımlamada ise bir çok firma tarafından geliştirilmiş şematik editör programlarından faydalanılır[16]. Dijital sistem tasarımı ihtiyaçların belirlenmesinden başlayan ve ileride incelenecek olan, fiziksel uygulamanın fonksiyonel tasarımını üretmeye doğru devam eden bir yöntemdir. Dijital tasarımların işleyişi için bir ASIC tasarımı göz önüne alınırsa; şekil 5.14'de gösterildiği gibi tasarım esnasında tipik olarak yer alan etkinliklerin sıralanmasına bir örnek olacaktır[17]. Tanımlama ne şekilde olursa olsun, derleme işlemi sonrasında, tüm tanımlamalar, standart bağlantı listesi (netlist) biçimine çevrilir [18]. Yapılan tanımlamaların, istenilen fonksiyonları yerine getirip getirmediği fonksiyonel benzetim (functional simulation) yapılarak test edilir. Benzetim sonucuna göre, tanımlamada gerekli değişiklikler yapılır[16]. Şekil 5.14'de tasarım sürecinin akış şeması verilmiştir.



Şekil 5.14: FPGA kullanılarak yapılan tasarım sürecinin akış diyagramı

## **5.4 VHDL Donanım Tasarım Dili**

### **5.4.1 Giriş**

Lojik devre tasarımının başlangıç dönemlerinde, tasarım lojik kapılar seviyesinde yapıldı. Ancak tasarımların karmaşıklaşması ve bunların lojik kapılar kullanılarak gerçekleştirilme zorluğu nedeniyle çeşitli yazılım dilleri tasarlanmaya başlandı. Bu dillere genel olarak HDL (Hardware Description Languages) adı verilir. Bu dillerden en çok kullanılanı VHDL (Very High Speed Integrated Circuit Hardware Description Language) yazılım dilidir. VHDL yazılım dili kullanılarak, her türlü lojik devre tasarlanabilir. Tasarımın işlevi yazılım programı içerisinde bulunan belirli test bölümleri vasıtasıyla kontrol edilebilir. Bu sayede sistem, donanım yapısı kurulmadan bilgisayar ortamında test edilebilir. Testlerin donanıma ihtiyaç duyulmadan yapılabilmesi tasarım süresinin kısalmasını sağlar[16].

### **5.4.2 VHDL ve donanım tasarımı karşılaştırılması**

VHDL dili kullanılarak tasarım yapılmasının standart donanım tasarım yöntemine göre bazı önemli üstünlükleri vardır.

Tasarım süresi: Teknolojinin hızla gelişimi beraberinde gerçekleşen bir devrenin kullanım ömrü azalmaktadır. Bu, devrenin tasarım zamanının kısıtlanması anlamına gelir. Böyle durumlarda, devrenin optimum tasarlanmış olmasının ötesinde tasarım süresinin kısalığı ön plana çıkar. VHDL dili kullanılarak tasarım yapılması özellikle bu noktada tasarımcıya önemli yararlar sağlar. VHDL dili doğrudan donanım tasarımına göre, tasarım süresi açısından çok daha kısa sürede sonuçlanır.

Tasarım esnekliği: Teknolojideki değişimle birlikte kullanılan elamanların yapıları değişmektedir. Yapı değişikliklerinin daha önce yapılmış tasarımlarda çalışabilmesi için, kullanılan tasarım ortamının buna uygun olabilmesi gerekir. VHDL dili fonksiyon bağımlı olarak çalışır. Dönüştürücü programlar yardımıyla yazılımın



donanım yapısı oluşturulur. Teknoloji deęişimleri durumlarında sadece bu dönüştürücü programların yeni teknolojiye uygun hale getirilmiş olması yeterli olacaktır.

Tasarım kolaylığı: Genel olarak VHDL dili kullanarak yapılan tasarımlarda, klasik donanım tasarımına göre, donanım bilgisine daha az ihtiyaç duyulur.

Yenileme kolaylığı: Teknolojideki hızlı deęişim, gerçekleşen devrelerin deęişim sürelerini azaltmıştır. Bu nedenle yapılacak deęişimlerin hızlı bir şekilde gerçekleştirilebilmesi gerekmektedir. VHDL dili ile yapılan tasarımlarda deęişim esnekliği tasarımcının yazılım gücü ile sınırlıdır.

### **5.4.3 VHDL dili mimari yapıları**

VHDL dili 3 mimari yapıdan oluşur.

#### **5.4.3.1 Davranışsal mimari**

Davranışsal mimaride, sistemin yapması gereken işlemler, ‘process’ yapıları kullanılarak gerçekleştirilir. Ancak tasarımın nasıl gerçekleşeceği konusunda bilgi verilmez. VHDL programı derleyicisi, varsa uyarılarını ya da hatalarını, tasarımın ne kadar yer kaplayacağını kullanıcıya bildirir. Bu nedenle bazı durumlarda programda yapılan küçük deęişimler, gerçekleştirme alanı bakımından büyük deęişiklere neden olabilir. Bunun için tasarım yapılırken sürekli olarak donanım düşünölmelidir.

Davranışsal mimaride ‘process’ ile birlikte duyarlılık ifade eden parametreler parantez içinde yazılır. Verilen parametrelerde deęişim olması durumunda, program yukarıdan aşağıya doğru gerçekleştirilir. Bir mimaride tek ‘process’ olmak zorunluluęu yoktur. Çoklu yapılarda tüm ‘process’ler aynı anda gerçekleşir ancak ‘process’ içindeki işlemler yukarıdan aşağı doğru yapılır. Böylelikle aynı program içerisinde bağımsız bloklar oluşturulabilir.

#### **5.4.3.2 Veri akışı mimarisi**

Veri akışı mimarisinde devre tasarımı; karşılaştırıcılar, toplayıcılar, kod çözücüler ve basit lojik kapılar kullanılarak tasarlanır. Program içerisindeki satırlar tamamen eşzamanlı olarak çalışır. Davranışsal mimariden farklı olarak duyarlılık ifade eden parametreler, program satırlarında ki eşitliklerin sağ tarafında kalan parametrelerdir.

Bu tür mimaride davranışsal mimariye göre dışarıdan bakıldığında yapılacak işlemin anlaşılabilirliği daha azdır. Aynı zamanda yapılacak işlemin gücü büyüdükçe her fonksiyonu standart elemanlarla ifade etme zorluğu nedeniyle, genelde kullanıcı davranışsal mimariyi tercih eder. Ancak davranışsal mimariye göre, devrenin kaplayacağı alan bakımından daha kontrollüdür.

#### **5.4.3.3 Yapısal mimari**

Yapısal mimari temel olarak tamamen kullanıcının kontrolündeki bir tasarım biçimidir. Bu tasarımda tüm bağlantılar yazılımcı tarafından tanımlanır ve işaret isimleri verilerek belirlenir. Ayrıca kullanılacak elemanlar yazılımcı tarafından 'component' olarak oluşturulur ve tasarımda kullanılır.

Yapısal mimari donanım tasarımında yapılacak çizimin, yazılım ile gerçekleşmesi olarak görülebilir. Bu nedenle sistem karmaşıklığı arttıkça bu tasarımın kullanımı çok zorlaşacaktır. Küçük projelerde ya da genelde aynı yapıların tekrar ettiği sistemlerde kullanılabilir. En büyük avantajı tasarımın kaplayacağı alanın kullanıcı kontrolü altında olmasıdır.

Örnek olarak bir karşılaştırıcının VHDL kullanılarak üç farklı mimari ile yazımı aşağıda verilmektedir.

Bildirim bölümü her mimari için aynı şekilde yapılır. Fonksiyonların bildirildiği bölümler farklılık gösterir[16].

```
entity karşılaştırıcı is
port  (X,Y : in std_logic;
C: out std_logic);
end karşılaştırıcı;
--Devre işleyişlerinin verildiği bölümler
--Davranışsal mimari
```

```
architecture davranış of karşılaştırıcı is
begin
process (X,Y)
begin
if (X=Y) then
C<='1';
else
C>='0';
end if;
end process;
end davranış;
```

```
--Veri akış mimarisi
architecture veri_akışı of karşılaştırıcı is
begin
C<=not(A xor B);
end veri_akışı;
```

```
--Yapısal mimari
architecture yapısal of karşılaştırıcı is
component XOR
port (G1,G2:in std_logic;
C:out std_logic);
End component;
component NOT
```

```
port (G: in std_logic;  
C:out std_logic);  
end component;  
signal I:std_logic;  
begin  
U0: XOR port map(X,Y,I);  
U1: NOT port map(I,Z);  
End yapısal;
```

#### **5.4.4 VHDL temel özellikleri**

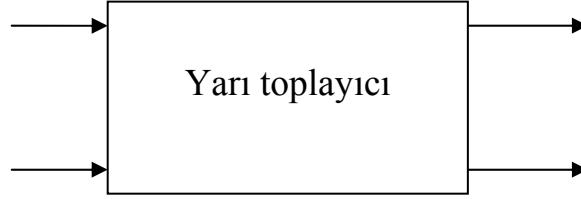
Bu bölümde VHDL dilinin bazı önemli özellikleri ve temel kullanım kuralları verilecektir.

##### **5.4.4.1 Yapısal ve davranışsal tanımlamalar**

VHDL dilinde ‘component’ tanımlamaları ‘entity’ birimleri içerisinde gerçekleştirilir. Her ‘component’ gerçekleştirilmesi iki bölümden oluşur. Birinci bölümde yapının hangi giriş çıkış ucundan oluşacağı bildirimi yapılır. Örnek olarak bir yarı\_toplayıcının bildirim yapısı şu şekilde ifade edilir.

```
entity yari_toplayici is  
port(  
giris1 : in bit;  
giris2: in bit;  
toplama : out bit );  
end yari_toplayici;
```

Gerçeklenen yazılım ile Şekil 5.15’de gösterilen giriş ve çıkışlar oluşturulur.



Şekil 5.15: Yarı toplayıcı giriş ve çıkışları

Bu bildirimde kullanılan port tanımlamaları 4 farklı tipte yapılabilir.

- in : Sadece okunabilir. Giriş için kullanılır.
- out : Sadece yazılabilir. Çıkış için kullanılır.
- buffer : Okunabilir yazılabilir. Devrenin içinden bir süreni olabilir.
- Inout : Okunabilir yazılabilir. Giriş ve çıkış için kullanılabilir.

İkinci bölümde ise ‘component’ davranış biçimi ifade edilir. Örnek yarı toplayıcıya ilişkin davranışsal tanımlama bölümü aşağıdaki gibi yapılabilir.

```
architecture davranissal_tanimlama of yari_toplayici is
begin
process
toplam <= giris1 xor giris2 after 10 Ns;
elde <= giris1 and giris2 after 10 Ns;
wait on giris1,giris2;
end process;
end davranissal_tanimlama;
```

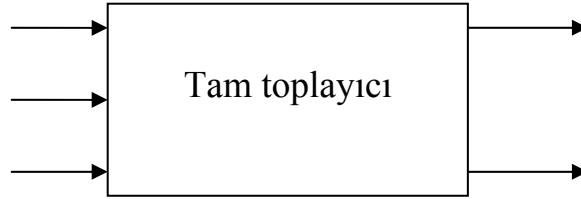
Bu yapı içerisinde kullanılan ‘after’ ifadeleri, gerçekleştirilmede herhangi bir etki yapmaz. Sadece simülasyonlarda gecikme amacıyla kullanılırlar.

Yapısal tanımlamanın belirtilmesi ile yarı toplayıcı oluşturulmuştur. Bu şekilde oluşturulan yapıların başka bir yazılım tarafından kullanılmalarda durumunda ifade edilen port yapıları yeni sistem için, sistem içi işaretleri gösterir ve VHDL dilinde

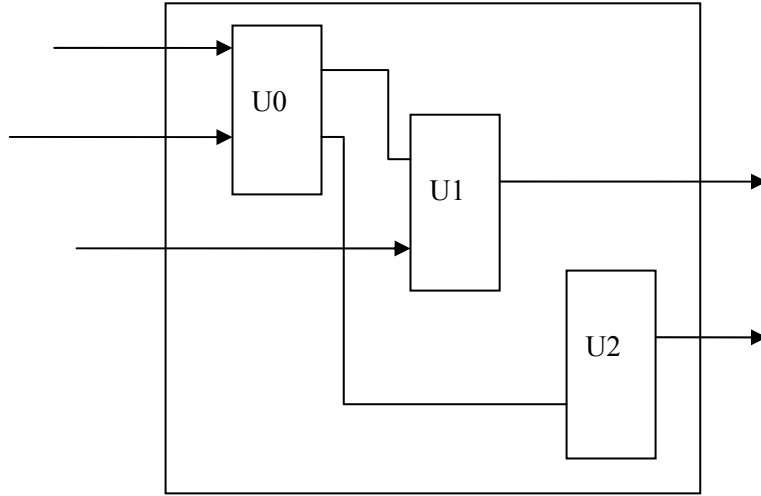
signal olarak ifade edilirler. Aşağıda bununla ilgili olarak bir tam toplayıcı örneği verilmiştir.

```
entity tam_toplayici is
port(
  giris1 : in bit; giris2 : in bit; elde_giris : in bit;
  sonuc : out bit; elde_cikis : out bit);
end tam_toplayici
architecture yapi of tam_toplayici is
  signal gecici_toplam : bit;
  signal gecici_elde1 : bit;
  signal gecici_elde2 : bit;
  component yari_toplayici
  port(giris1 : in bit; giris2 : in bit;
  toplam : out bit; elde : out bit);
  end component;
  component or_kapisi
  port( g1: in bit; g2 : in bit; C1 : out bit);
  end component;
  begin
  U0 : yari_toplayici
  port map(
  giris1 => giris1, giris2 => giris2,
  toplam => gecici_toplam, elde => gecici_elde1);
  U1 : yari_toplayici
  port map(
  giris1 => gecici_toplam, giris2 => elde_girisi,
  toplam => sonuc, elde => gecici_elde2);
  U2 : or_kapisi
  port map(
  g1 => gecici_elde1, g2 => gecici_elde2,
  C1 => elde_cikisi);
  end yapi;
```

Yukarıda örnekte görüldüğü gibi kullanılan her 'component' bir tanımlayıcı ile etiketlenmiştir. Bu etiketten sonra kullanılan 'component' lerin yapı içerisindeki işaret yerleri ve bağlantıları ifade edilir. Tüm bu tanımlamalar sonucunda VHDL dili Şekil 5.16 ile oluşturulan tanımlamayı, Şekil 5.17 ile gösterilen yapı biçiminde gerçekleyecektir.



Şekil 5.16: Tam toplayıcı giriş çıkışları



Şekil 5.17: Tam toplayıcı iç yapısı

#### 5.4.5 Veri türleri ve nesnelere

VHDL dilinde değer tutan her şey bir nesne olarak görülür. Her nesne de kendi yapısını tanımlayan bir türe sahiptir.

### 5.4.5.1 Veri türleri

VHDL’de, sıkça kullanılacağı düşünülen belirli veri türleri yazılım içerisinde oluşturulmuştur. Örneklerde görülen ‘bit’ deyimi bahsedilen türlerden biridir. Ancak VHDL’de bu konu ile ilgili olarak esnek bir yapı mevcuttur. Kullanıcı kendi veri türünü oluşturabilir. Aşağıda bazı veri türü tanımlama örnekleri bulunmaktadır. Bu örnekler genel olarak ‘integer’ yapısında olan tür tanımlamalarını gösteren örneklerdir.

```
type x is range-10 to 10;  
type y is range 40 downto 30;  
type z is range-1E10 to 1E10;
```

‘integer’ türündede tanımlamalarının yanı sıra kayan nokta türünde tanımlamaları da yapılabilir. Bu tanımlamalarda sınır tam sayılardan oluşmaz. Aşağıda bununla ilgili bir örnek mevcuttur.

```
Type k is range -2+0.1E-20 to 2+0.1E20;
```

Bu iki farklı tür tanımlamaları dışında kalan tüm tanımlamalarda 3.gurup tür tanımlamaları olarak düşünülür. 3. gurup tür tanımlamalarına en iyi örnek olarak 3 seviyeli lojik tasarım düşünüldüğünde, ortaya çıkacak 3.seviyeyi tanımlama sorunu cevap verir. Burada 3 seviyeli parametresi, ‘0’,’1’,’Z’ değerlerinden oluşturulacak şekilde tanımlanması durumunda 3 seviyeli yapı tanımlanmış olur. Bununla ilgili örnekler aşağıda verilmiştir.

```
type 3 seviye is(‘0’,’1’,’Z’);  
type mantık is(False,True);
```

### 5.4.5.2 Nesnelere

VHDL’de 3 farklı nesne tanımlamaları bulunmaktadır. Bunlar,işaretler (signals),değişkenler (variables) ve sabitlerdir(constants). Tüm tanımlamaların farklı özellikleri aşağıda sıralanmıştır.



İşaretler donanımsal bağlantıyı ifade eden nesne türleridir. Değişimler bağlı oldukları işaretlerin değişimleri ile gerçekleşir. Doğrudan değerinin değişimine yönelik işlem yapılmaz. Değişkenler, işaretlerden farklıdır. Değerinin değişimi için işlem yapılmalıdır. İşlem sonucunda elde edilen değişim sonucu, bir sonraki yazıma kadar kalır. Son tanımlama biçimi ise sabitlerdir. Bunlar bu iki tanımlamadan farklı olarak bir kez değer alırlar ve bu değeri sürekli olarak korurlar. Bu anlamda bir ROM gibi düşünülebilir[16]. Aşağıda tanımlamalarla ilgili örnekler sunulmuştur.

Constant x:Integer :=16#FFFF# ;

Variable y :Boolean;

Signal z :Bit ;

#### **5.4.6 Arayüz listeleri**

Arayüz listeleri VHDL dili içerisinde 3 farklı yapıda kullanılır. Bunlar ‘entity’ tanımlamaları, ‘block’ ifadeleri ve altprogram tanımlamalarıdır. Tüm tanımlamaları oluşturan elemanlar parantez içerisinde yer alacaktır. Her eleman aşağıdaki verilen 3 vasıfla birlikte tanımlanır.

Nesne sınıfı (signal, constant yada variable)

Verinin akış yönü (in, out, inout yada buffer)

Veri türü (Bit vb...)

Her tanımlama türü için izin verilen nesne sınıfları ve modlar bulunmaktadır.

#### **5.4.7 VHDL dili ana yapıları**

VHDL içerisinde, bazı önemli yapılar sıkça kullanılır. Bunlar aşağıda sıralanmaktadır.

##### **5.4.7.1 ‘Entity’ tanımlamaları**

‘Entity’ tanımlamaları her tasarım içerisinde mutlaka bulunması gereken bir yapıdır. Genel yazım kuralı aşağıdaki gibidir.

entity kimlik is  
generic interface listesi ;  
port interface listesi;  
Tanımlamalar  
begin  
ifadeler  
end kimlik ;

Bu tanımlamalar içerisinde bildirimler yapılır. Bu bildirimler, tür bildirimleri, alttür bildirimleri, sabitlerin bildirimleri, dosya bildirimleri, altprogram bildirimleri ve ‘use’ ifadeleridir.

#### **5.4.7.2 ‘Architecture’ yapıları**

‘Architecture’ yapıları genel yazım biçimi aşağıdaki gibidir.

architecture kimlik of kimlik hedefi is  
bildirimler  
begin  
ifadeler  
end kimlik;

Yapı içerisinde ifade edilen bildirimler bölümü kapsamında temel tanımlamalar, işaret tanımlamaları, altprogram yapısı, nitelik tanımlamaları ve özellikleri, ‘component’ tanımlamaları mevcuttur.

#### **5.4.7.3 Altprogramlar**

Altprogramlar VHDL’ nin önenli yapılarından. Program içerisinde tekraren çağrılabilirler. VHDL dili ‘procedure’ ve ‘function’ olmak üzere iki tür altprogram yapısını destekler. Birinci tür altprogramlar dönüş değeri almazlar. Altprogram tanımlamaları sadece ‘interface’ bilgisini içerir. Altprogram yapısı ise ‘interface’ bilgisi, yerel tanımlamalar ve ifadeleri içerir. Altprogram tanımlaması ile altprogram

yapısı arasındaki fark, ‘entity’ bildirim ve ‘architecture’ yapısı arasındaki fark gibidir. Altprogram bildirimleri aşağıdaki gibi yapılır.

procedure kimlik interface listesi

function kimlik interface listesi return dönüş türü

Bu bildirimlerde, ‘interface’ listelerinin olma zorunluluğu bulunmamaktadır. Yani altprogramlar parametresiz olabilir. Bununla ilgili olarak bazı örnekler verilmiştir.

procedure A ;

function B return byte;

proceture C(x:inout İnteger):

function D(x:İnteger )return byte;

proceture C(variable x:inout İnteger);

function D(constant x:in İnteger )return byte;

Altprogram tanımlamaları aşağıda ifade edildiği gibi yapılabilir. Her iki altprogram türü için aynı yapı geçerlidir.

altprogram özelliği is

bildirimler

begin

ifadeler

end kimlik;

function deneme (X,Y: byte) return byte

begin

if (X>2)

return X;

else

return Y;

end if;

end deneme ;

#### 5.4.7.4 'Package' ve 'Use' yapıları

Tüm yazılım sistemlerinde olduğu gibi birden çok yerde kullanılan elemanları ortak olarak kullanmak için belirli yapılara imkan verilmiştir. VHDL dilinde ortak kullanım oluşturma yapısını 'package' eydana getirir. Tanımlama biçimi ve ilgili bir örnek aşağıda verilmiştir[16]. Prosedürlerin ve fonksiyonların ilişkilendirildiği gruplar bir çok farklı VHDL modülünün karşılıklı olarak paylaşabildiği bir modül içerisinde (birleştirilebilir, toplanabilir ) kümelenirilebilir. Herkesin paylaşabildiği bu tür bir modül bir paket (package) olarak ifade edilir. Fonksiyonların ve prosedürlerin tanımlamalarına ek olarak, paketler kullanıcı tanımlı veri tipleri ve sabitler içerebilir ve kütüphaneler(libraries) içerisine yerleştirilebilirler. Kütüphaneler genellikle tasarım birimleri için depolanırlar ve paketler tasarım birimlerinin bir tipidirler. Genel olarak; prosedürler, fonksiyonlar, paketler ve kütüphaneler VHDL programlarını tekrar kullanmak, modül oluşturmak ve korumak için tasarımcıya olanaklar sağlarlar[19].

```
package kimlik is
tanımlamalar
end kimlik ;
package lojik is
type 3 seviye is ('0','1','Z');
constant bilinmeyen deęer :3 seviye:='0':
function tmleme (
giriř: 3 seviye )
return 3 seviye;
end lojik ;
```

Bu örnekte içerisinde 'fuction' tanımı bulunan bir 'package' bildirim yapısı verilmiştir. Genel yapıları ise

```
package body kimlik is
bildirimler
end kimlik;
```

Şeklinde gösterilebilir. Verilen örneğin yapı yazılımı ise aşağıdaki gibi olabilir.

```
package body lojik is
function tümle(
giriş:3 seviye)
return 3 seviye is
begin
case giriş is
when '0'=>
return '1';
when '1'=>
return '0';
when 'Z'=>
return 'Z';
end case;
end tümle;
end lojik;
```

Bu yapılarının kullanımları 'use' ifadeleri ile gerçekleştirilir. 'use' yazımından sonra kullanılacak 'package' adı yazılır. Hemen ardından nokta ve sonrasında ise yapı içerisinde kullanılacak tanım yazılır. Başka kullanılacak yapılar varsa virgülden sonra aynı mantık içerisinde yazılır. 'package' içerisindeki tüm tanımlamalar kullanılacak ise bu durumda isim yazımı ardından nokta ve sonrasında 'all' ifadesi yazılarak, tüm tanımlamalar alınmış olacaktır. Bununla ilgili olarak yukarıdaki örnekteki yapıyı kullanan bir kod yazılabilir.

```
use lojik.all ; Tüm tanımlamalar alındı.
entity tümle is
port (X : in 3_seviye ; Y : out 3_seviye);
end b tümle;
architecture tümle_yapısı of tümle is
begin
process
```

```
begin  
Y<=tümle(X) 10 ns;  
wait on X;  
end process;  
end tümle_yapısı;
```

#### **5.4.7.5 Tasarım kütüphaneleri**

VHDL’de yazılan tanımlamalar, yazım kurallarına uyduğu doğrulandıktan sonra sentezleme ve simülasyonda kullanılmak üzere kütüphanelerde saklanır. Bir kütüphane tek başına analiz edilebilir. Ancak birbirine bağımlı yapıların belirli bir sırasıyla analiz edilmesi oluşabilecek hataları engeller.

## 6. UYGULAMA

Bu bölümde önerilen MPPT algoritmasının FPGA da tasarlanan sistem mimarisi açıklanacaktır. Öncelikle MPPT algoritmasının çalışma şekli verildikten sonra sistemin mimarisi ve bu mimariyi oluşturan blokların VHDL modelleri açıklanacak ve sistemin benzetim sonuçları verilecektir.

### 6.1 Conductance Incremental Metotlu MPPT Algoritması

Bu algoritma maksimum güç noktasını gözlemleme tekniğine dayanır. Atmosferik koşullardaki değişimleri hızlı bir şekilde algılamak için önerilmiştir[21]. C.I. algoritması PV nin azar azar artan ve anlık iletkenliğinin karşılaştırılması ile maksimum güç noktasının takip edilmesi prensibine dayanır. PV nin ürettiği akım ve gerilim değerlerindeki örnek değişimleri azar azar artan iletkenliği( incremental conductance) de değiştirebilecektir [20]. Bu metot denklem 6.1'e dayanmaktadır. Bu da gerilime bağlı olarak PV gücündeki değişimi görmek ve sonucu sıfıra ayarlamaktır[21]. Bu durum şekil 6.1 de gösterilmektedir.

$$\frac{dP_{PV}}{dV_{PV}} = I_{PV} \frac{dV_{PV}}{dV_{PV}} + V_{PV} \frac{dI_{PV}}{dV_{PV}} = I_{PV} + V_{PV} \frac{dI_{PV}}{dV_{PV}} = 0 \quad (6.1)$$

$$-\frac{I_{PV}}{V_{PV}} = \frac{dI_{PV}}{dV_{PV}} \quad (6.2)$$

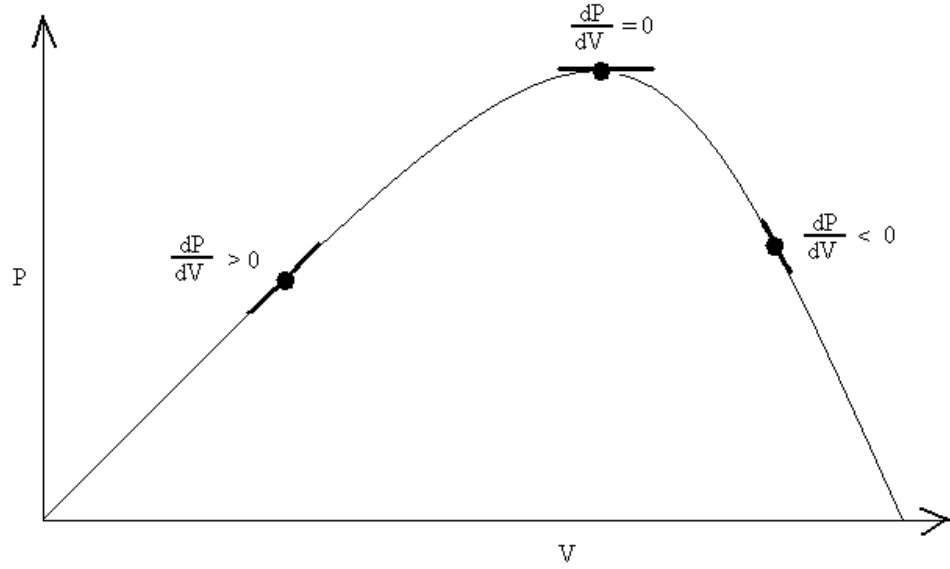
Denklem 6.2 nin sol tarafı anlık iletkenliğin zıtlığını ifade etmektedir. denklem 6.2'nin sağ tarafı ise azar azar artan iletkenliği (incremental conductance) göstermekte ve denklem 6.3 te gösterildiği gibi yazılabilir.

$$G = \frac{dP_{PV}}{dV_{PV}} \quad (6.3)$$

Diğer taraftan incremental değişkenler olan  $dV_{PV}$  ve  $dI_{PV}$  nin her ikisi anlık yapılan ölçümlerle bir önceki yapılmış anlık ölçümlerin karşılaştırılması ile elde edilir [21] ve bunlar sırasıyla denklem 6.4 ve denklem 6.5 te verildiği gibi olmaktadır.

$$dV_{PV}(t_2) = V_{PV}(t_2) - V_{PV}(t_1) \quad (6.4)$$

$$dI_{PV}(t_2) = I_{PV}(t_2) - I_{PV}(t_1) \quad (6.5)$$



Şekil 6.1: PV'nin  $dP / dV$  değişimini gösteren  $P - V$  karakteristik eğrisi

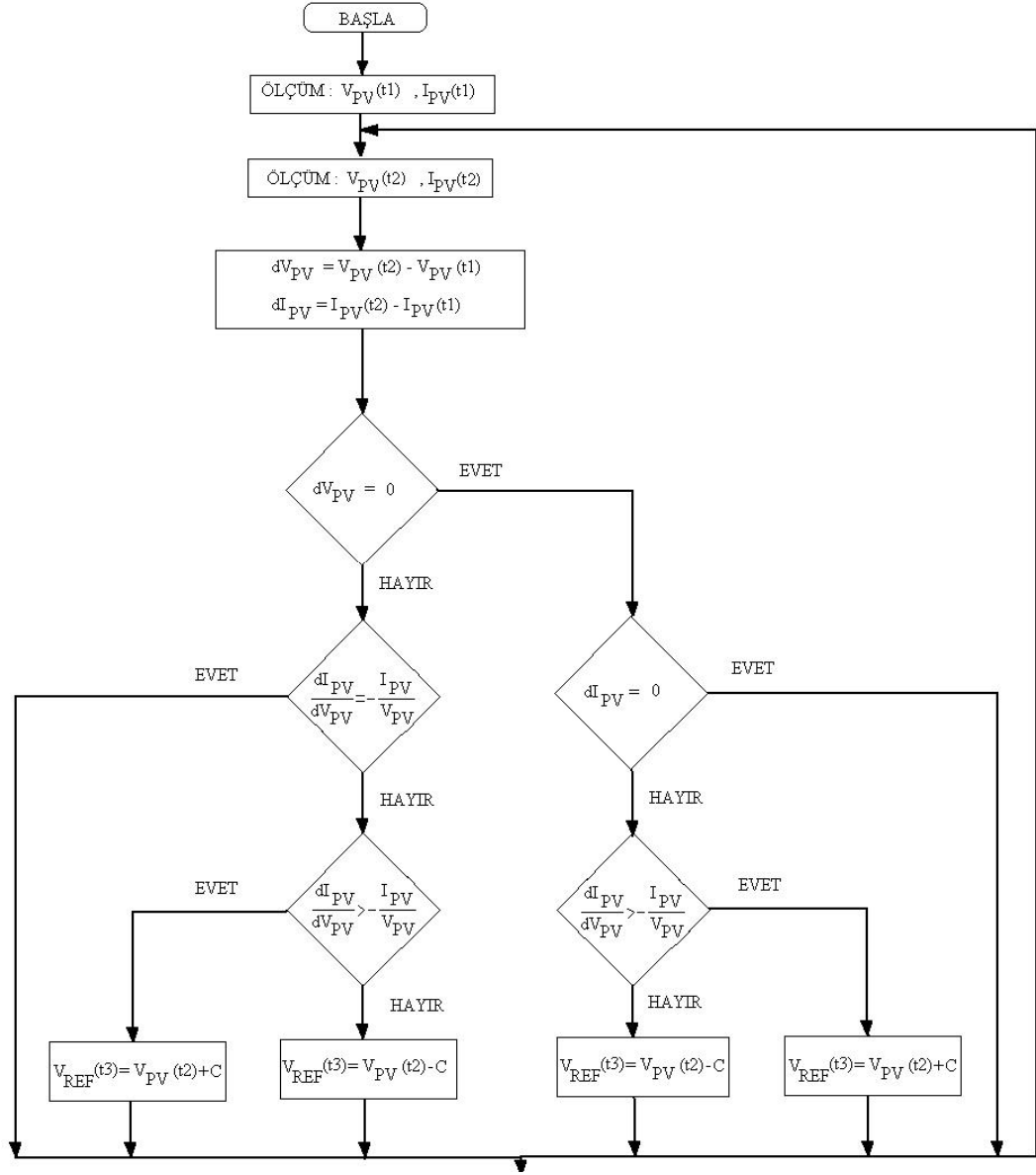
Böylelikle, PV'nin türevinin analiz edilmesi ile PV nin maksimum güç noktasında yada bu noktadan uzak olarak çalıştığı test edilebilir. Denklem 6.6 – 6.8 de algoritmanın çalışma şeklini ve şekil 6.2 de algoritmanın akış diyagramı verilmektedir.

$$\frac{dP_{PV}}{dV_{PV}} > 0 \text{ için } V_{PV} < V_{PMP} \quad (6.6)$$

$$\frac{dP_{PV}}{dV_{PV}} = 0 \text{ için } V_{PV} = V_{PMP} \quad (6.7)$$



$$\frac{dP_{PV}}{dV_{PV}} < 0 \text{ için } V_{PV} > V_{PMP} \quad (6.8)$$



Şekil 6.2: Incremental Conductance Algoritmasının Akış Diyagram Şeması

## 6.2 FPGA Tasarım Mimarisi

Algoritmanın VHDL yazılım tasarımı Xilinx firmasına ait ISE 8.1i Project Navigator programı altında gerçekleştirildi. Yazılım Xilinx firmasının ürettiği Virtex4 serisi FPGA yongasına göre gerçekleştirildi. Yazılımda MPPT

algoritması gerçekleştirilirken algoritmada yer alan matematiksel işlemler için ayrıca algoritmalar kullanıldı. Alınacak akım ve gerilim değerlerinin hassasiyetinin artırılması ve sonuçların daha etkin olması için tasarımda kullanılan veriler 16 bitlik ve Kayan noktalı (Floating Point) sayılar kullanıldı. Tasarım ISE 8.1i programındaki ekran görüntüsü Şekil 6.3' te görülmektedir. Mimarinin genel blok şeması Şekil 6.4'te ve ModelSim SE6.2e programıyla gerçekleştirilen örnek benzetim görüntüsü Şekil 6.5'te verildiği şekilde olmaktadır.

The screenshot shows the Xilinx ISE Design Summary window for the project 'TEZ\_MPPT'. The window is divided into several sections:

- TEZ\_MPPT Project Status:**

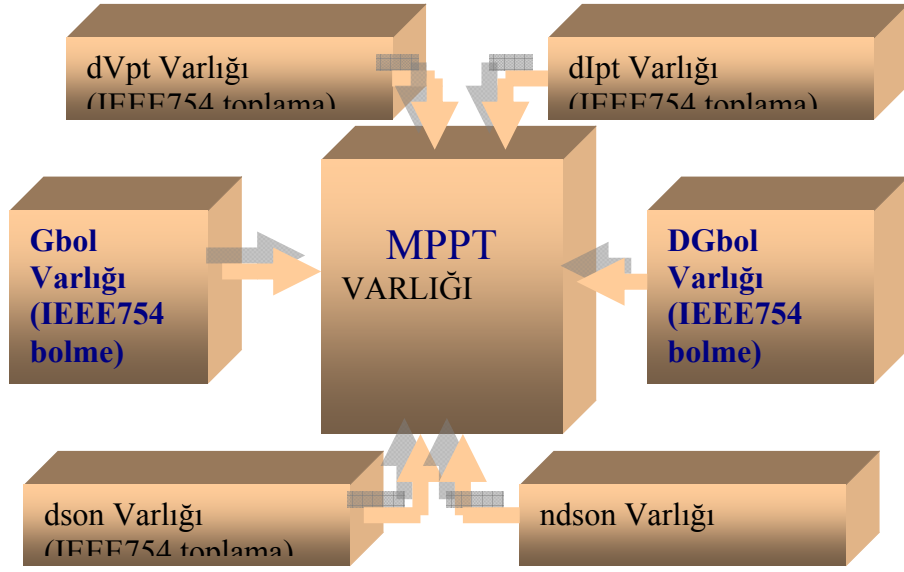
Project File:	tez_mppt.ise	Current State:	Translated
Module Name:	mppt	Errors:	No Errors
Target Device:	xc4vx200-11f1513	Warnings:	66 Warnings (0 new, 0 filtered)
Product Version:	ISE, 8.1i	Updated:	Per 22. May 21:11:21 2008
- Device Utilization Summary (estimated values):**

Logic Utilization	Used	Available	Utilization
Number of Slices	1099	89088	1%
Number of Slice Flip Flops	653	178176	0%
Number of 4 input LUTs	1955	178176	1%
Number of bonded IOBs	65	960	6%
Number of GCLKs	1	32	3%
- Detailed Reports:**

Report Name	Status	Generated	Errors	Warnings	Infos
Synthesis Report	Current	Sat 22. May 15:25:18 2007	0	66 Warnings (0 new, 0 filtered)	12 Infos (0 new, 0 filtered)
Translation Report	Current	Sat 22. May 15:25:26 2007	0	0	0
Map Report					
Place and Route Report					
Static Timing Report					
Bitgen Report					

The console window at the bottom shows the message: "Started : 'Launching Design Summary'".

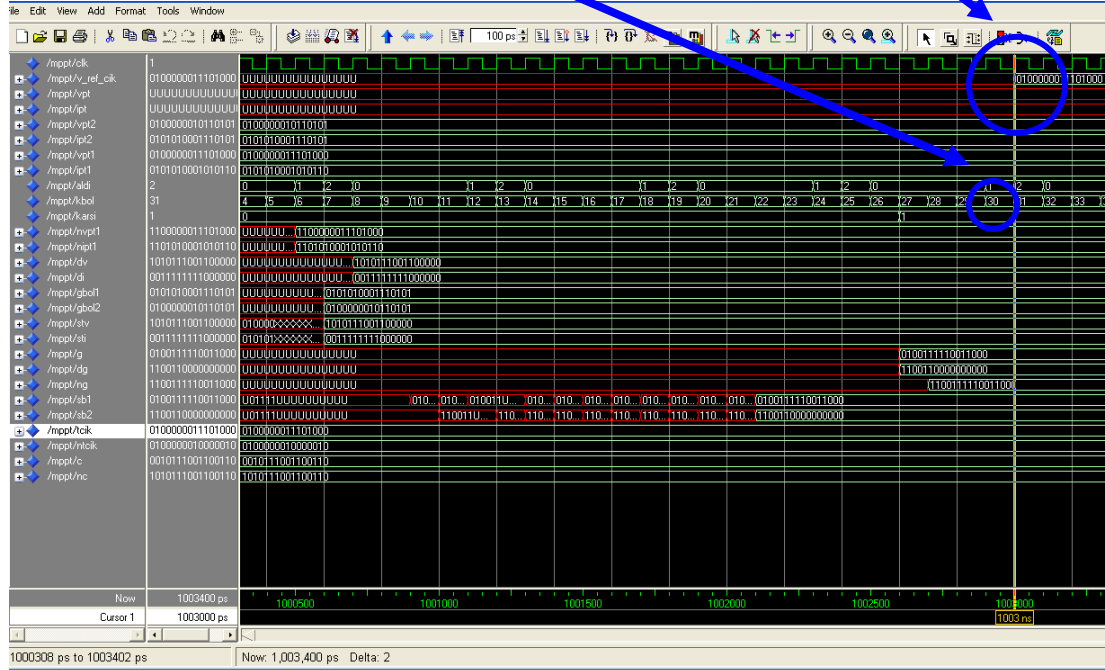
Şekil 6.3: MPPT Algoritmasının ISE programındaki görünümü



Şekil 6.4: MPPT varlık mimarisi

MPPT varlığı çıkışları toplam 31 saat darbesinde üretiyor. 100MHz lik çalışma frekansında 310ns içerisinde sonuç üretilmektedir.

V\_ref\_cik: MPPT referans gerilim çıkışı



Şekil 6.5: VHDL kodunun örnek benzetim çıktıları

## 6.3 Veri Gösterimi

### 6.3.1 Kayan noktalı sayılar aritmetiği

Bir çok algoritma sayı gösterimindeki dinamiklik ve hassasiyetteki işlem yapma kapasitesi sebebiyle kayan noktalı sayıları tercih etmektedir. Birçok avantajına rağmen kayan noktalı sayılar sıradan bir uygulama için bile aşırı derecede donanım kaynağı tüketmektedirler. Algoritmamızda fotovoltaik sistemimizden alacağımız akım ve gerilim değerlerinin virgüllü (Floating Point) değerler olacağından FPGA içerisinde toplama, çıkarma ve bölme gibi matematiksel hesaplamaları gerçekleştirirken bu yöntemi kullandık. Tabii ki bu tür işlemleri FPGA içerisinde gerçekleştirmek başlı başına ayrı bir problemi ortaya çıkardığından bu işlemler için yeni algoritmalar oluşturmaya ihtiyaç duyuldu.

### 6.3.2 Kayan noktalı (Floating Point) sayıların gösterimi:

IEEE Kayan Nokta Aritmetiği Standardı kayan noktalı sayıların gösteriminde en çok kullanılan standarttır. İkilik sistemdeki sayılar bilimsel gösterimle gösterildikten sonra işaret, üst ve anlamlı kısımdan oluşan üç parça şeklinde ifade edilebilirler. Bu gösterime sonsuz, sayı değil ve sıfırın gösterimi dahildir. IEEE 754 standardına göre sayılar tek duyarlı(32 bit) ve çift duyarlı(64 bit) şekilde gösterilebilirler.

i(s)	ü(e)	k(f & m)
------	------	----------

i (sign) : işaret biti, 1 bit ile ifade edilir.(31)  
ü (exponent) : üs bitleri , 8 bit ile ifade edilir.(30-23)  
k (fraction & mantisa) : kesir bitleri, 23 bit ile ifade edilir.(22-0)

Şekil 6.6: IEEE 754 32 Bit Kayan sayı formatı

Gerçel sayıları Şekil 6.6 de görülen 32 bit IEEE format çevirmek için;

$$v = -1^s 2^{(e-127)} (1.f) \quad (6.9)$$

formülü kullanılabilir. Formülde üs(exponent için kullanılan 127 sayısı bias değeridir.

Örneğin, -3.625(ondalık) veya -11.101(ikilik) sayısı floating point formatta aşağıdaki gibi ifade edilir.

$-3.625 = -11.101 = -1.1101 \times 2^1$  burada,  $s = 127 + 1$ ,  $1.f = 1.1101$ ,  $s = 1$  dir.

$$v = -1^1 2^{(128-127)} (1.1101) \quad (6.10)$$

burada;

$s = 1$ (onluk),  $e = 128$ (onluk) ve  $f = 680000$ (onaltılık).

Sonuçta -3,625 sayımız, floating point formatta, C0680000(onaltılık) şeklinde gösterilir.

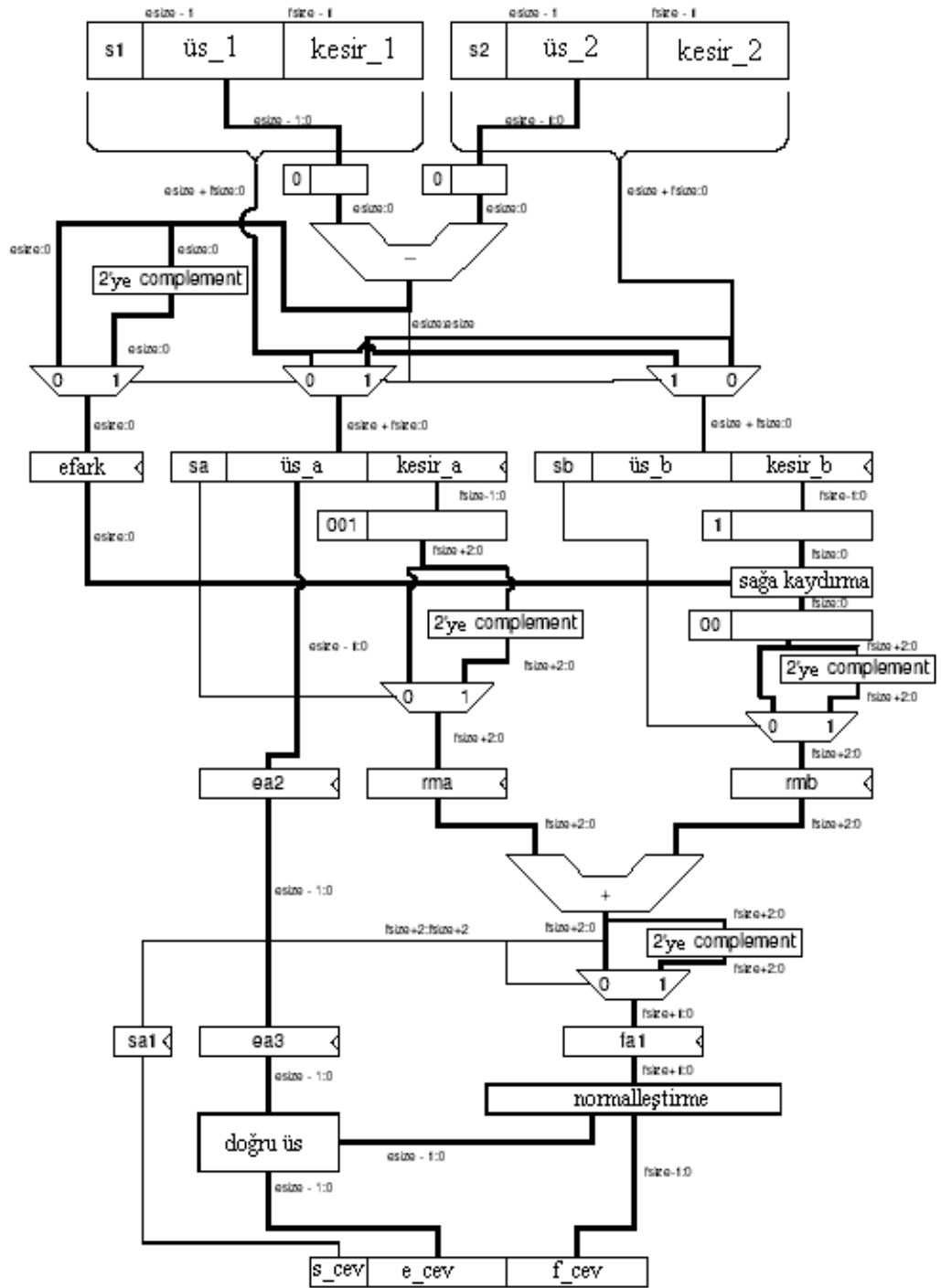
### 6.3.3 Toplama Ve Çıkarma

Toplama veya çıkarmaya işlemine tabi tutulacak sayılar  $F_1$  ve  $F_2$  olsun.

$$F_{toplama} = F_1 + F_2 \quad (6.11)$$

$$F_{minus} = F_1 - F_2 = F_1 + (-F_2) \quad (6.12)$$

şeklinde yazılabilir. Dolayısıyla sadece toplama algoritmasını gerçeklemek çıkarma yapmak içinde yeterli olacaktır. Aşağıda toplama algoritması verilmiştir (Bkz. Şekil6.7).



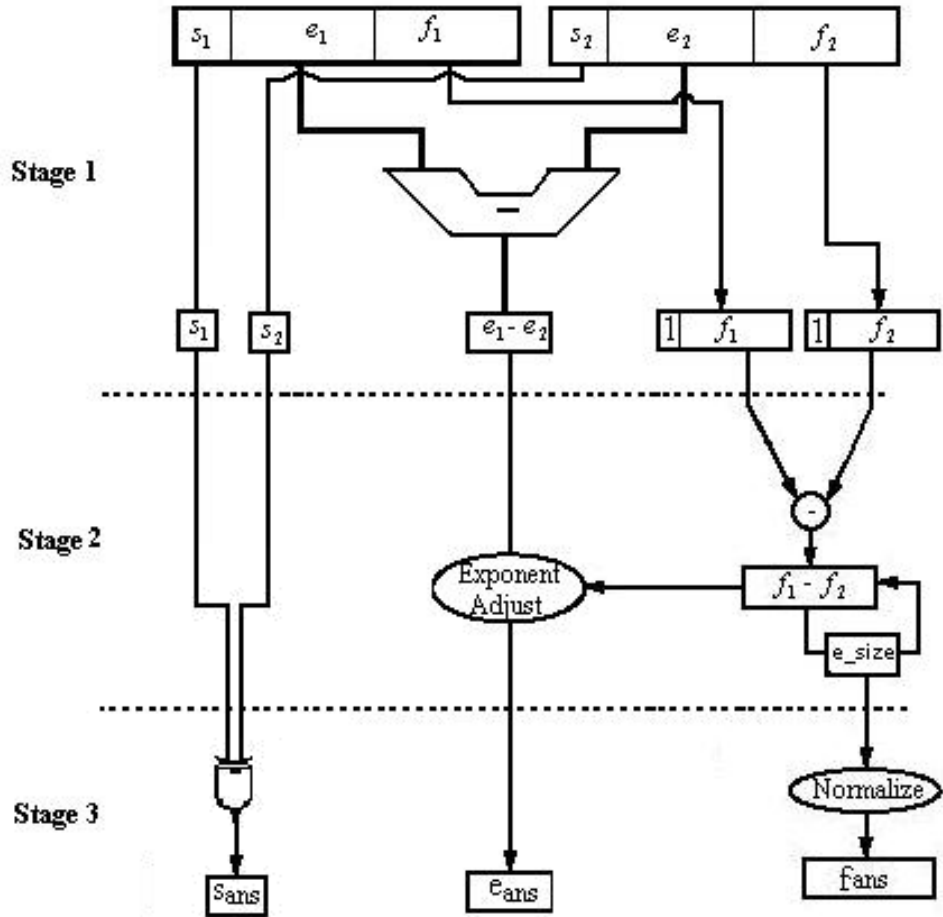
Şekil 6.7: Kayan Noktalı sayıların toplanması

### 6.3.4 Bölme

Bölme işlemine tabi tutulacak sayılar  $F_1$  ve  $F_2$  olsun.

$$F_{bölme} = \frac{F_1}{F_2} \quad (6.13)$$

şeklinde yazılabilir. Bölme algoritması Şekil 6.8 ile verilmiştir[16].



Şekil 6.8: Kayan Noktalı sayıların bölünmesi

## 6.4 Algoritmanın Uygulanması

### 6.4.1 VHDL Kodları

Bölüm 6.1 de anlatılan Incremental Conductance algoritmasının FPGA programlama yapısına göre uygulanması için Şekil 6.9 da gösterilen VHDL kodları geliştirilmiştir. FPGA lerin çalışma prensibine göre tanımlanan modül ve kodların hepsi aynı anda çalışmaktadır. Bu yapı normal işlemcilerin çalışma yöntemlerinde oldukça farklıdır. Çünkü işlemciler tüm komutları sırasıyla yürütmektedir ve hiçbir komutun sonucu kendinden önceki komutların çalışmasını etkilememektedir. Bu karşın FPGA paralel çalışma mimarisi gereği bir komutun ürettiği sonuç aynı zamanda tüm programdaki diğer komutların sonuçlarını da değiştirmektedir. Yazılan VHDL programının bir kısmı belirtilen algoritmayı gerçeklerken, önemli bir kısmı da paralel çalışma sırasında hatalı işlemlerin olmasını engelleyip işlemlerin düzenli çalışmasını sağlamaktadır.

```
-----  
-- Company:  
-- Engineer: AZIZ GUNEROGLU  
-- Create Date: 12:01:31 05/19/2008  
-- Design Name:  
-- Module Name: mppt - Behavioral  
-- Project Name: incremental conductance algorithm  
-- Target Devices: Virtex 4  
-- Tool versions:  
-- Description:  
-- Dependencies:  
-- Revision:  
-- Revision 0.01 - File Created  
-- Additional Comments:  
-----  
library IEEE;  
use IEEE.STD_LOGIC_1164.ALL;  
use IEEE.STD_LOGIC_ARITH.ALL;  
use IEEE.STD_LOGIC_UNSIGNED.ALL;  
entity mppt is  
generic(e : integer := 5; -- 8  
f : integer := 10; -- 23  
s : integer := 1; -- 1  
bias : integer := 15; -- 127  
sifir : std_logic_vector(15 downto 0) := "0000000000000000";  
bir : std_logic_vector(15 downto 0) := "0010111001100110";  
eksibir : std_logic_vector(15 downto 0) := "1010111001100110";  
end entity mppt;
```

Şekil 6.9: Incremental Conductance algoritmasının VHDL kodları ile gerçekleştirilmesi



```

port( clk          : std_logic;
      V_ref_cik    : out std_logic_vector(15 downto 0);
      vpt         : in std_logic_vector(15 downto 0);
      ipt         : in std_logic_vector(15 downto 0));
end mppt;

architecture Behavioral of mppt is

component toplama  --toplama işlemi yapan modül--
  port( Sy1       : in std_logic_vector(s+e+f-1 downto 0);
        Sy2       : in std_logic_vector(s+e+f-1 downto 0);
        Cks       : out std_logic_vector(s+e+f-1 downto 0);
        clk       : in std_logic);
end component;

component bolme  --bölme işlemi yapan modül--
  port( sayi1     : in std_logic_vector(15 downto 0);
        sayi2     : in std_logic_vector(15 downto 0);
        cikis     : out std_logic_vector(15 downto 0);
        clk       : in std_logic);
end component;

-- FPGA iç mimarisinde compenentler arasında dahili bağlantıları temsil eden sinyallerin tanımlanması --
signal vpt2 : std_logic_vector(s+e+f-1 downto 0);
signal ipt2 : std_logic_vector(s+e+f-1 downto 0);
signal vpt1 : std_logic_vector(s+e+f-1 downto 0);
signal ipt1 : std_logic_vector(s+e+f-1 downto 0);
signal aldi,kbol,karsi : integer:=0;
signal nvpt1,nipt1,dv,di: std_logic_vector(s+e+f-1 downto 0);
signal gbol1,gbol2 : std_logic_vector(s+e+f-1 downto 0);
signal stv,sti,G,DG,nG :std_logic_vector(s+e+f-1 downto 0);
signal sb1,sb2,tcik,ntcik: std_logic_vector(s+e+f-1 downto 0);
signal C: std_logic_vector (15 downto 0):="0010111001100110";
signal nC: std_logic_vector (15 downto 0):="1010111001100110";

begin -- FPGA Mimarisinin çalışma biçiminin belirlendiği kısmın başlangıcı -

-- matematiksel işlemlerin yapıldığı modüllere yapılan atamalar ve ilişkilendirme--

dVpt : toplama port map(vpt2,nvpt1,stv,clk);
dIpt : toplama port map(ipt2,nipt1,sti,clk);
Gbol : bolme port map(gbol1,gbol2,sb1,clk);
DGbol: bolme port map(di,dv,sb2,clk);
dson : toplama port map(vpt2,C,tcik,clk);
ndson: toplama port map(vpt2,nC,ntcik,clk);

process(clk)
variable sayac:integer:=0;
begin
if clk'event and clk='1' then
sayac :=sayac+1;
kbol<=kbol+1;

if sayac > 5 then
sayac:=0;
aldi<=1;

--dışarıdan FPGA içerisine alınacak verilerin atanması--

```

Şekil 6.9: Incremental Conductance algoritmasının VHDL kodları ile gerçekleştirilmesi (Devamı)

```

        vpt2<=vpt;
        ipt2<=ipt;

        vpt1<=vpt2;
        ipt1<=ipt2;
        nvpt1 <= not vpt1(15) & vpt1(14 downto 0);
        nipt1 <= not ipt1(15) & ipt1(14 downto 0);
    end if;
    if aldi>0 then
        aldi<=2;

        gbol1<=ipt2;
        gbol2<= vpt2;
        if aldi =2 then
            aldi<=0;
            dv<=stv ;
            di<=sti ;
        end if;
    end if;

    if kbol>25 then

        -- Bölme işleminin yapıldığı modülden elde edilen sonuçların atanması ve
        -- algoritmaya göre karşılaştırılması ve referans çıkışın elde edilmesi--

        G <= sb1 ;
        DG <= sb2 ;
        nG <= not G(15) & G(14 downto 0);
        karsi<= 1;

    end if;

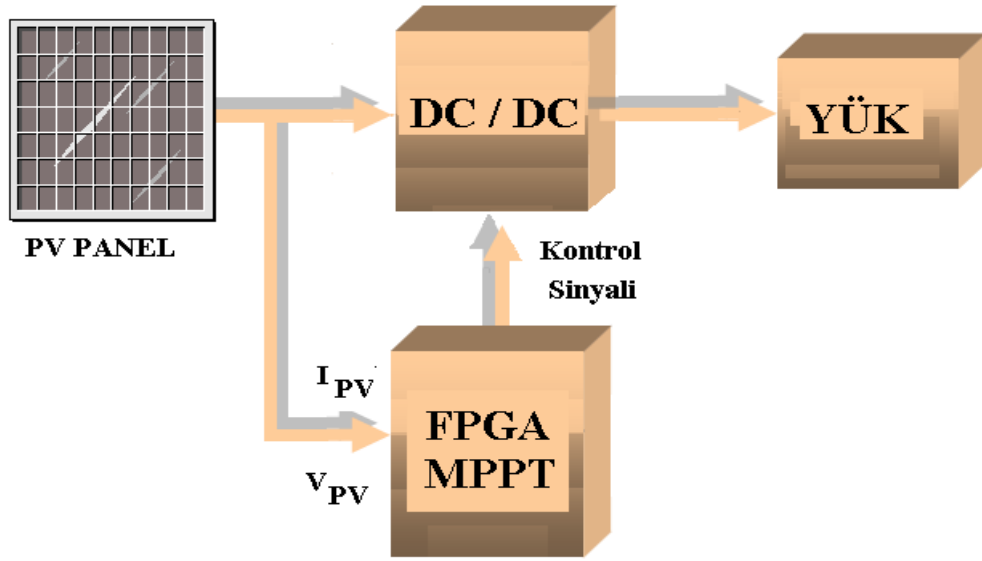
    if karsi=1 then
        if dv = sifir then
            if di = sifir then
                V_ref_cik <=vpt2 after 0.3 ns;
            else
                if DG > nG then
                    V_ref_cik <= tcik after 0.3 ns;
                else
                    V_ref_cik <= ntcik after 0.3 ns;
                end if;
            end if;
        else
            if DG = nG then
                V_ref_cik<=vpt2 after 0.3 ns;
            else
                if DG > nG then
                    V_ref_cik <= tcik after 0.3 ns;
                else
                    V_ref_cik <= ntcik after 0.3 ns;
                end if;
            end if;
        end if;
    end if;

end if;
end if;
end process;
end Behavioral;

```

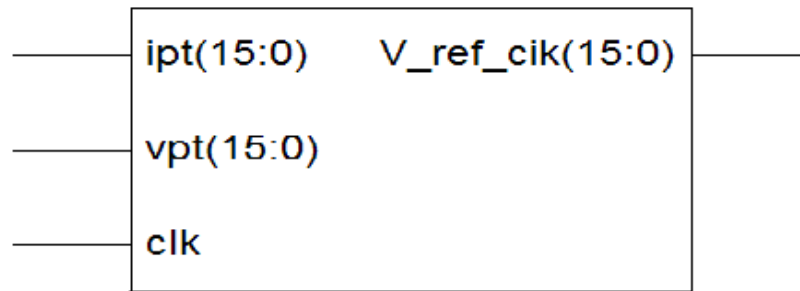
Şekil 6.9: Incremental Conductance algoritmasının VHDL kodları ile gerçekleştirilmesi (Devamı)

Sistemin genel blok şeması şekil 6.10 da verilmektedir. Sistem Panelden aldığı akım ve gerilim değerlerine göre FPGA da gömülü olan MPPT algoritmasına göre sisteme kontrol sinyali üretmekte ve maksimum noktayı yakalamaktadır. Sistemde görülen DC/DC konvertör bu çalışmada üzerinde durulmamıştır sadece FPGA'nın sistemde yüke güç aktarımını maksimum düzeyde tutacak şekilde çalışması ön plana çıkmaktadır.



Şekil 6.10: Sistemin genel blok şeması

VHDL kodlarının Xilinx firmasına ait ISE 8.1i programında tasarlanan MPPT Algoritmasına ait varlığın lojik devre mimari şeması Şekil 6.11 da verilmiştir.



Şekil 6.11: Incremental Conductance algoritmasının lojik devre mimari şeması

## 6.4.2 PV modeli

FPGA mimari tasarımında kullanılan VHDL yazılımının test edilebilmesi için bir güneş pilinin benzetimini sağlayacak bir model kullanıldı. Bu model için Şekil 6.12 de kodları görülen MATLAB yazılımı gerçekleştirilmiştir.

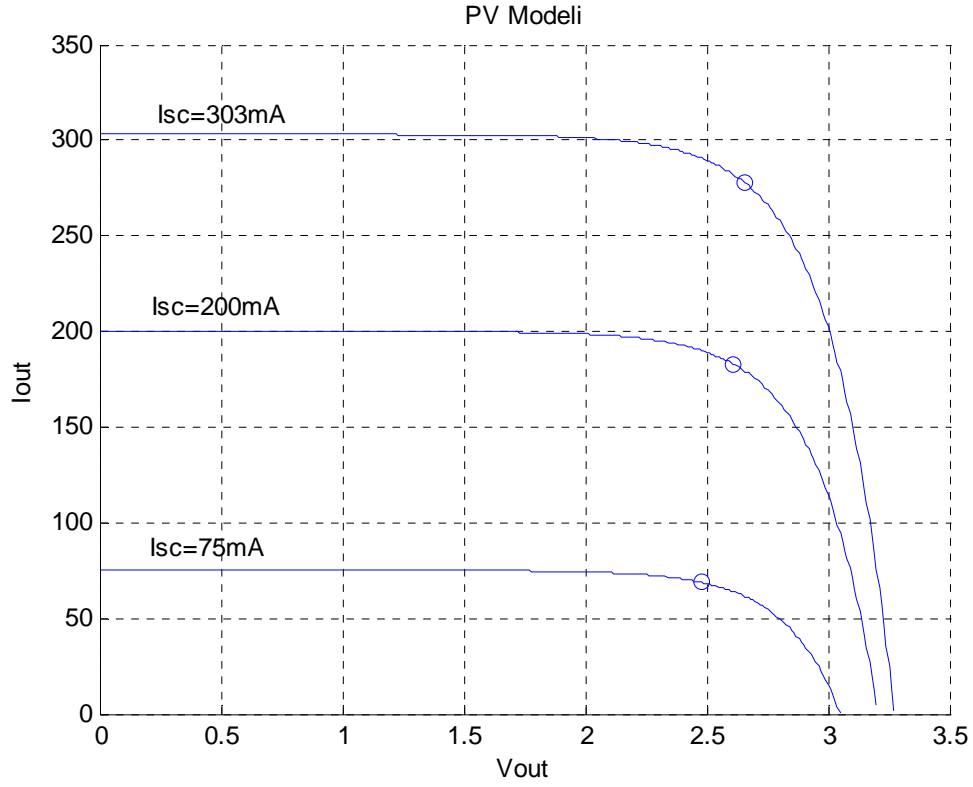
```
clear all; close all;clc;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%PV modellenmesi ve çizimi
vocref=3.27;
iscref=301;
k1=0.1569;
T=37;
dvt=-0.014;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Isc=303;
voc=vocref+k1*log(Isc/iscref)+dvt*(T-37);% 37 C° lık sabit
sıcaklıkta
v1=0:0.01:voc; % giriş için V gerilimi
k2=log(1-244/258)/(2.55/3.27-1); % Ref. İkinci ışınımdan
alındı
i1=Isc-Isc*exp(k2*(v1./voc-1)) % çıkış akımı
[x,y]=max(i1.*v1); % maksimum güçteki akım gerilim
bulunuyor
vm=v1(y)
im=i1(y)
Vmax(1)=vm;
Imax(1)=im;
figure(1)
hold on
grid
plot (v1,i1); % V-I grafiğini çiz
plot (vm,im,'o'); % max noktalar

p=v1.*i1; % çıkış gücünün hesaplanması
[x,y]=max(p); % max. çıkış gücünün bulunması
vm=v1(y)
pm=p(y)
figure(2) % v-p grafiğini çiz
hold on
grid
plot (v1,p);
plot (vm,pm,'o'); % maks. gücü işaretle
```

Şekil 6.12: PV modelinin gerçekleştirilmesi

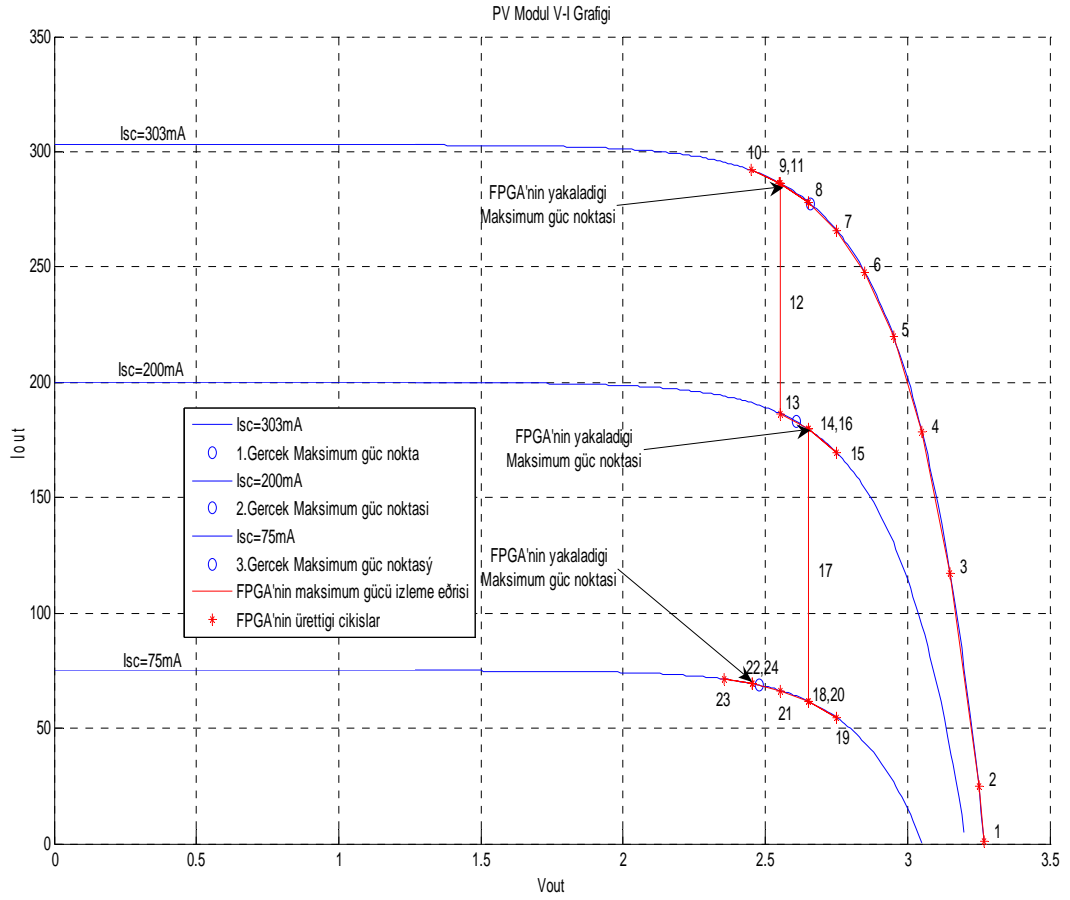
PV modelinin farklı farklı ışınım şiddetlerine göre kısa devre akımlarının da ürettiği I-V eğrileri Şekil 6.13 de gösterilmiştir.



Şekil 6.13: PV modelinin I-V Eğrisi

#### 6.4.2 Uygulama sonuçları

PV modelinin ürettiği akım ve gerilimler FPGA mimarisi içerisine alınarak VHDL kodlarına adım adım uygulanıp VHDL programının ürettiği yeni referans gerilimleri PV modeline uygulayarak sistemin maksimum noktaya ulaşması ve bu noktada kalması sağlanmıştır. Sistem maksimum noktada çalışırken ışınım şiddeti başka bir ifadeyle kısa devre akımı değiştirilmiş ve FPGA yazılımının bu yeni duruma kendini uyarlaması adım adım izlenmiştir. Sistem değişen çevre ortamlarına göre Incremental Conductance algoritmasını sürekli işleterek ve ard arda alınan ölçümlerle iterasyon yapılarak yeni durumdaki maksimum güce ulaşılmaktadır. Şekil 6.14 de VHDL kodlarının PV modelinin I-V grafiği üzerinde adım adım maksimum güç noktasına ulaşması görülmektedir.

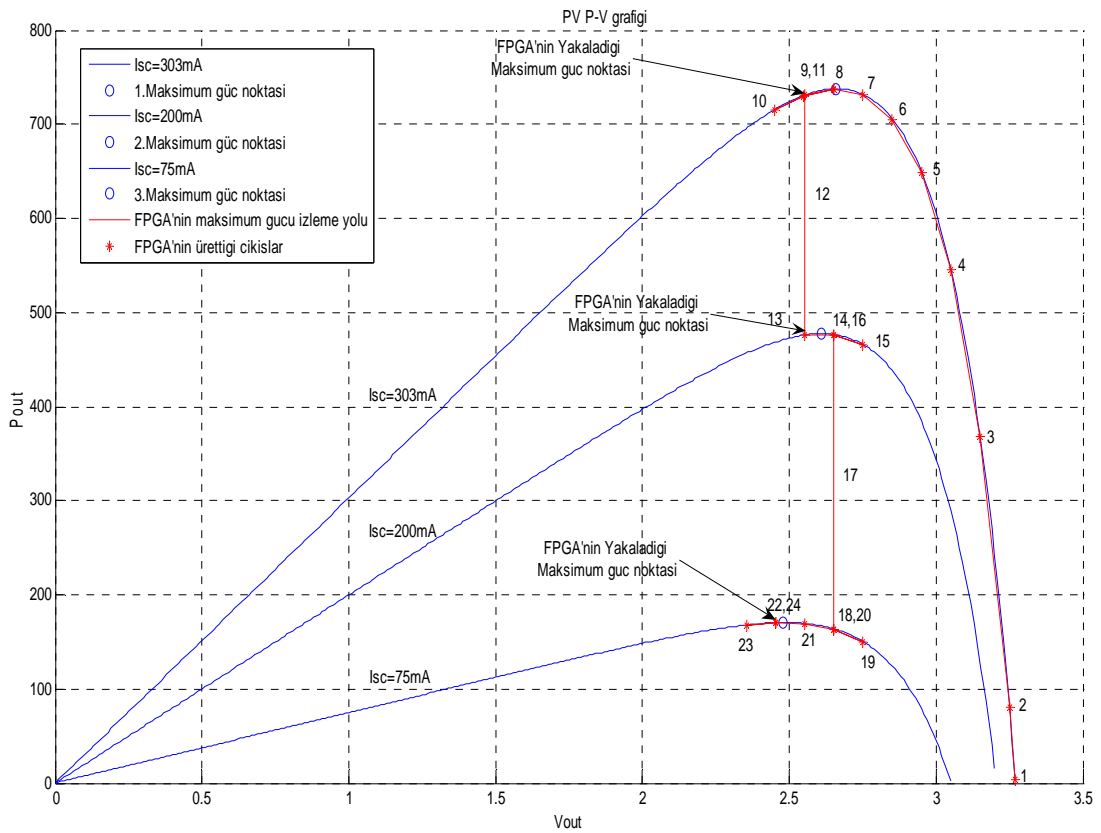


Şekil 6.14 FPGA için gerçekleştirilen MPPT algoritmasına göre sistemin maksimum güce ulaşması adım adım gösterilmiştir. Kodlar yürütülmeye 1 numaralı adımda başlamıştır. Her seferinde dışarıdan alınan akım ve gerilim verileri algoritmaya göre işlenerek maksimum güç noktasına ulaşmak için referans gerilimler üreterek bu noktayı yakalamaya çalışmaktadır. 303mA kısa devre akımına sahip eğrinin maksimum noktası 8 numaralı adımda işaretlidir. Algoritmanın yapısı gereği kodlar önce 9 ve 10 numaralı adımları işletmiş ve tekrar geri dönerek 11. adımda maksimum gücü yaklaşmıştır. Bu noktada güneş piline uygulanan ışınım şiddeti azaltılarak kısa devre akımının 200mA e düşmesi sağlanmıştır. Böylece maksimum güç noktasının yeri değiştirilmiştir. 12. adımda bu olay gözükmemektedir.

Çalışmaya devam eden algoritma maksimum noktaya ulaşabilmek için 14, 15 ve 16. adımları gerçekleştirmiştir. Sistemin maksimum noktaya yaklaştığı bu ışınım da

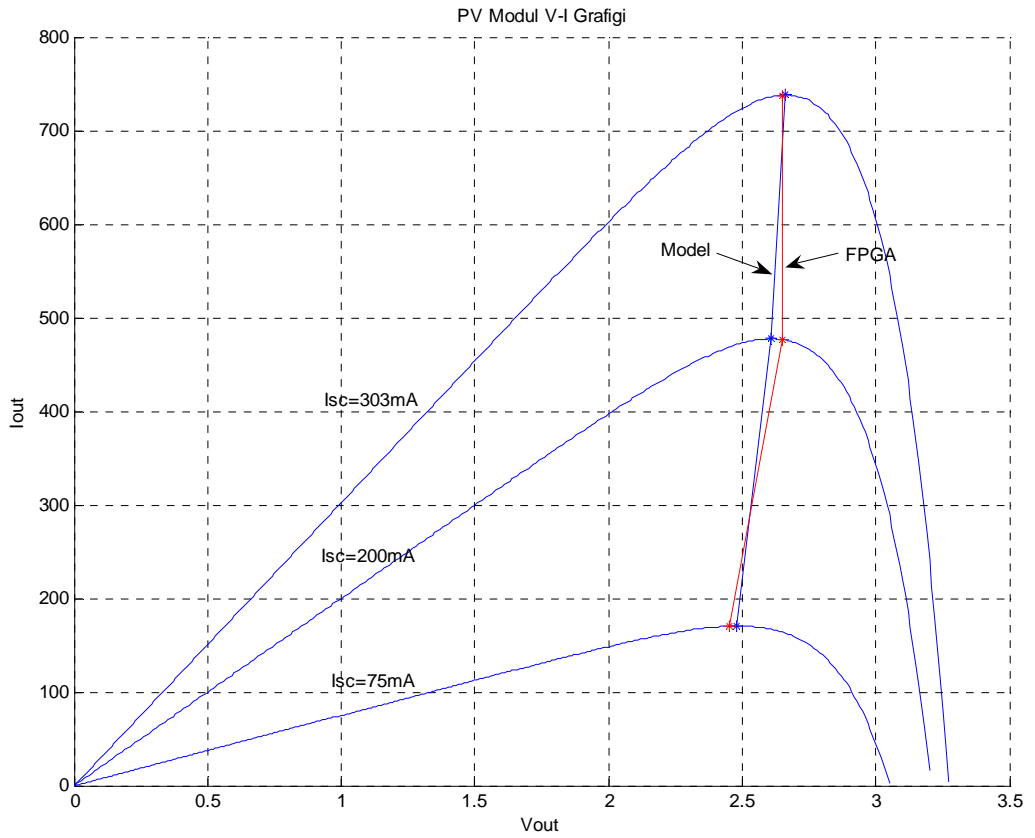
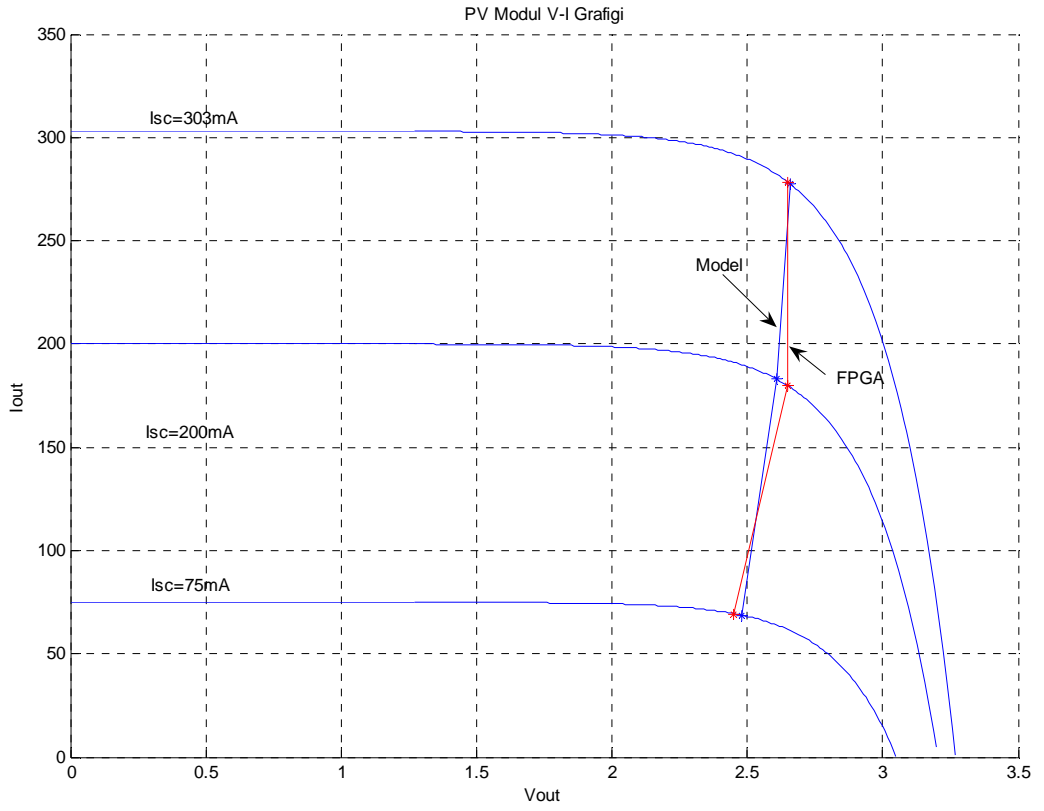
görülmektedir. 17. adımda görüldüğü gibi ışınım tekrar değiştirilmiş ve kısa devre akımı 75mA e düşürülmüştür. Yazılım 18 ve 19. adımlarda maksimum gücün ters yönüne doğru gitmiştir. Çünkü önceki durumla yeni durum arasındaki değişim bu adımın ters olmasına sebep olmuştur. Fakat 20, 21, 22, 23 ve 24. adımlarda görüldüğü gibi yazılım her zaman maksimum gücün etrafında salınım yapmaktadır.

Şekil 6.15 de VHDL yazılımının P-V grafiği üzerinde maksimum güce ulaşması izlenmektedir.



Şekil 6.15: Maksimum gücün P-V grafiği üzerindeki takibi

PV modelinin maksimum güç noktalarının üzerindeki eğri ile VHDL kodlarının tespit ettiği maksimum güç noktalarının eğrisi çakışmıştır. Buradan VHDL kodlarının sistemi maksimum noktada çalıştırdığı anlaşılmaktadır. Bu durum Şekil 6.16 ve Şekil 6.17 de gösterilmiştir.





Sisteminin testi sırasında FPGA tasarım mimarisine girilen V-I değerleri ve sistemin ürettiği referans değerleri Tablo 6.1 ve Tablo 6.2 de gösterilmiştir.

Tablo 6.1: Sistemin testinde kullanılan akım ve gerilimler

<b>Isc=303mA Voc=3.271V</b>					
<b>Adım</b>	<b>V (Volt)</b>	<b>IEEE 754</b>	<b>I (mA)</b>	<b>IEEE 754</b>	<b>P (mW)</b>
	3,2700	0100001010001010	1,2711	0011110100010101	4,1565
1	3,2500	0100001010000000	24,7241	0100111000101110	80,3533
2	3,1500	0100001001001101	117,0248	0101011101010000	368,6281
3	3,0500	0100001000011010	178,7105	0101100110010101	545,0670
4	2,9500	0100000111100111	219,9358	0101101011011111	648,8106
5	2,8500	0100000110110100	247,4872	0101101110111011	705,3385
6	2,7500	0100000110000001	265,9001	0101110000100111	731,2253
7	2,6500	0100000101001110	278,2021	0101110001011000	737,2356
8	2,5500	0100000100011011	286,4296	0101110001111001	730,3955
9	2,4500	0100000011101000	291,9258	0101110010001111	715,2182
10	2,5500	0100000100011011	286,4296	0101110001111001	730,3955
11	2,6500	0100000101001110	278,2057	0101110001011000	737,2451
12	2,5527	0100000100011011	286,4296	0101110001111001	731,1688
<b>Isc=200mA Voc=3.2059V</b>					
13	2,5527	0100000100011011	186,5069	0101100111010100	476,0962
14	2,6523	0100000101001110	179,6441	0101100110011101	476,4700
15	2,7520	0100000110000001	169,2908	0101100101001010	465,8883
16	2,6523	0100000101001110	179,6441	0101100110011101	476,4700
<b>Isc=75mA Voc=3,052V</b>					
18	2,6523	0100000101001110	61,7416	0101001110110111	163,7572
19	2,7520	0100000110000001	54,5791	0101001011010010	150,2017
20	2,6523	0100000101001110	61,7416	0101001110110111	163,7572
21	2,5527	0100000100011011	66,3919	0101010000100110	169,4786
22	2,4531	0100000011101000	69,4111	0101010001010110	170,2724
23	2,3535	0100000010110101	71,3714	0101010001110101	167,9726
24	2,4531	0100000011101000	69,4111	0101010001010110	170,2724

Tablo 6.2: FPGA Sisteminin ürettiği referans gerilimler

Adım	V <sub>cik</sub> (Volt)	IEEE754
	I <sub>sc</sub> =303	
1	3,15	0100001001001101
2	3,05	0100001000011010
3	2,95	0100000111100111
4	2,85	0100000110110100
5	2,75	0100000110000001
6	2,65	0100000101001110
7	2,55	0100000100011011
8	2,45	0100000011101000
9	2,55	0100000100011011
10	2,65	0100000101001110
11	2,5527	0100000100011011
	I <sub>sc</sub> =200	
13	2,5527	0100000100011011
14	2,6523	0100000101001110
15	2,7520	0100000110000001
16	2,6523	0100000101001110
	I <sub>sc</sub> =75	
18	2,6523	0100000101001110
19	2,7520	0100000110000001
20	2,6523	0100000101001110
21	2,5527	0100000100011011
22	2,4531	0100000011101000
23	2,3535	0100000010110101
24	2,4531	0100000011101000

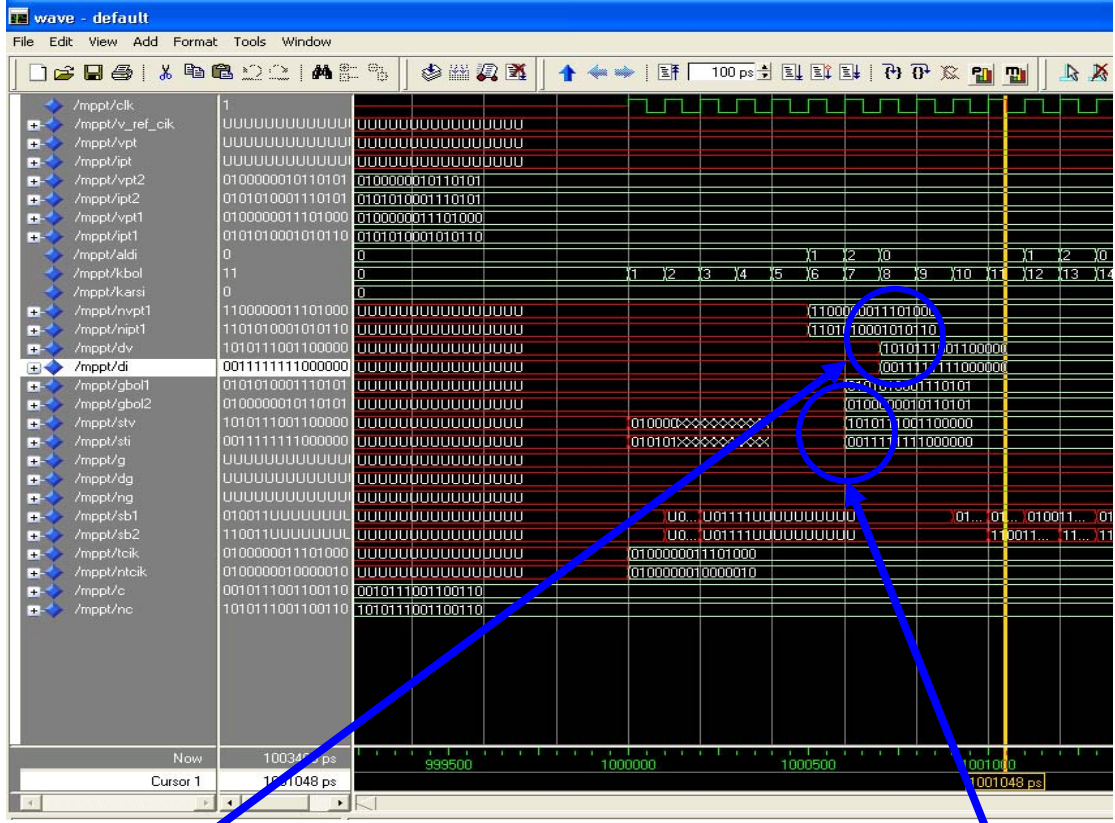
PV nin farklı ışınım şiddetlerinde ürettiği gerçek maksimum güç noktaları ile FPGA'nın yakaladığı maksimum güç noktaları arasında ki bağıl hatalar ise %1,6 ile % 4.03 arasında değişmekte. Sistem ilk anda devreye sokulduğunda FPGA'nın maksimum noktayı tesbit etmek için yaptığı iterasyonlar fazla iken ilk maksimum noktayı yakaladıktan sonra PV sisteminde meydana gelen ışınım şiddetindeki değişimlerde yeni maksimum noktayı tespit etmesi çok daha kısa sürede olabilmektedir. Bağıl hatalar tablo 6.3'te görülmektedir. Maksimum noktanın tespit

edilmesinde algoritmamızda kullandığımız ve kodların her bir yeni durumda sisteme verilecek her yeni referans değerinin hesaplanmasında kullanılan artırım ve azaltım miktarı olan C değeri olarak 0.1 gibi sabit bir değer olarak kullanılmaktadır. Hassasiyetin artırılması ve maksimum noktaya daha bir yaklaşılmak istenmesi durumu bu C değerinin uygun seçilmesine de bağlıdır. C miktarını çok küçük seçilmesi iterasyon sayısını arttıracaktır çünkü FPGA'nın maksimum noktaya tırmanmak için yapacağı ölçüm ve algoritmanın işletilme sayısı arttıracaktır.

Tablo 6.3: Sistemin bağıl hataları

Işınım Şiddeti	Gerçek Maksimum Güç Noktası (V)	FPGA'nın Yakaladığı Maksimum Güç Noktası (V)	Bağıl Hata
303 mA	2.6600	2.5527	0.0403
200 mA	2.6100	2.6520	0.0160
75 mA	2.4800	2.4530	0.027

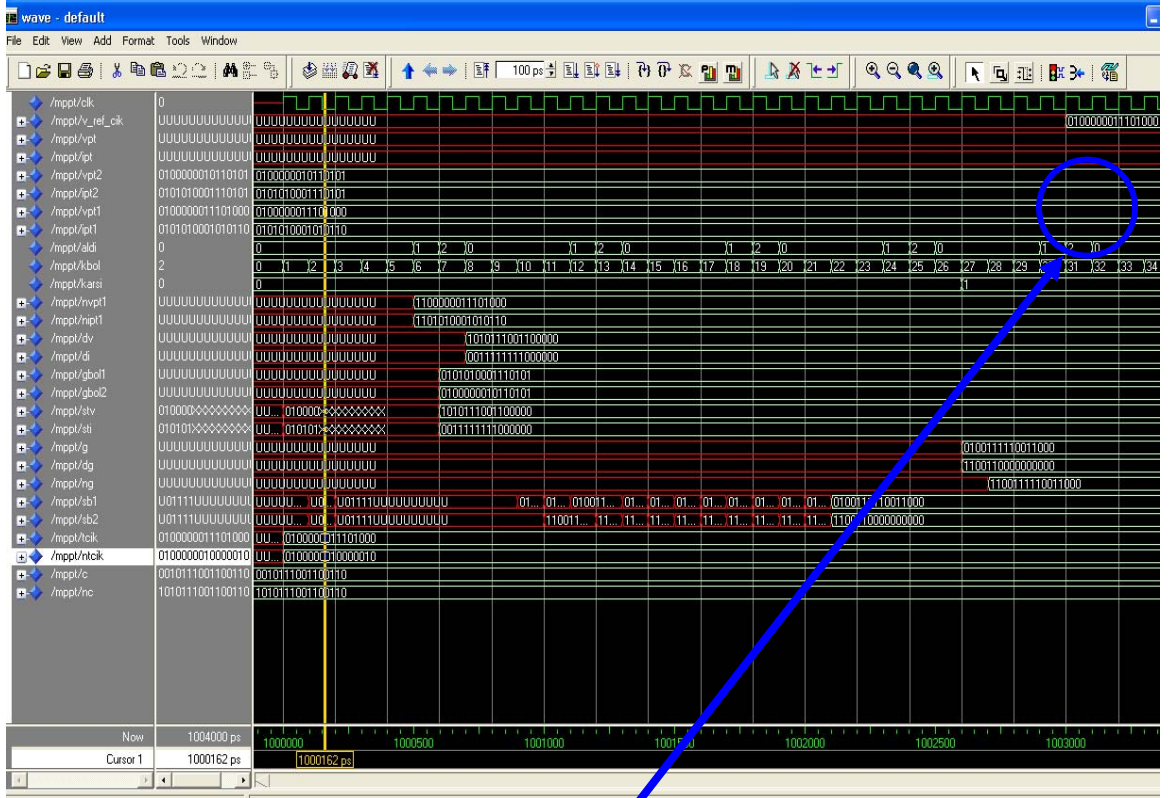
Benzetimde PV den elde edilen akım ve gerilim değerleri algoritma içerisinde matematiksel işlemler ve karşılaştırmalar yapılarak maksimum noktayı yakalayacak değere ulaşana kadar iterasyonlar yaparak referans çıkışı vermektedir. Bu çıkışların benzetim görüntüsüne örnek Şekil 6.18 ve Şekil 6.19 verildiği gibidir.



Güneş pilinden ölçülen Akım (di) ve Gerilimdeki (dv) değişimin hesaplanması 8 saat darbesinden sonra

Algoritmada gerilim ve güç değişimini tespit etmek için kullanılacak  $dv/di$  ve  $I_{(t2)} / V_{(t2)}$  hesaplarının yapılması

Şekil 6.18: Algoritmada kullanılan matematiksel işlemlerin benzetim görüntüsü



MPPT varlığı çıkışları toplam 31 saat darbesinde üretiyor. 100MHz lik çalışma frekansında 310ns içerisinde sonuç üretilmektedir

Şekil 6.19: FPGA' nın ürettiği referans gerilimin benzetim görüntüsü

## 7. SONUÇLAR VE ÖNERİLER

MPPT algoritmaları PV güç sistemleri uygulamalarında kaçınılmaz kontrol mekanizmalarının temelini oluşturur. FPGA ile yapılan algoritma yazılımları son zamanlarda artış göstermeye başladı[14]. MPPT algoritması olarak en çok kullanılanları P&O Algoritması ve I.C. Algoritma teknikleridir[21]. Klasik mikro denetleyicilerle bu algoritmalarından duyarlılığı daha çok olduğu için I.C. algoritması tercih edilmektedir[21]. FPGA'ların kullanımı yaygınlaştıkça bu algoritmaya olan ilgi daha da artış gösterdi[22]. Bununla beraber, FPGA'da bu algoritmayı gerçekleştirme başlangıçta karşılaşılan zorluk matematiksel işlemlerin algoritmalarını oluşturmaktır. Haberleşme alanında daha yaygın olarak kullanılan FPGA'lar için tasarlanmış matematiksel işlem algoritmaları[16] uygulamamızda kullanılacak şekilde VHDL dili ile kodlandı. Daha önce bu konuda yapılan çalışmalarda PV lerin çalışması esnasında çevresel faktörlerin değişimini hızlı bir şekilde algılayacak ve maksimum güç noktasını yakalamaya dönük tepki vermek önemli bir problem idi[23]. FPGA'ların paralel işlem yapabilme kabiliyetleri ve hızlı çalışmaları bu problemi çözmüş görünmektedir[24]. I.C. algoritmasının en temel avantajı, atmosferik koşulların değişmesine hızlı bir şekilde ve iyi bir sonuç vermesi bakımından iyi bir metod sunmasıdır[21]. Bu avantajın yanında FPGA'ların programlanabilme kabiliyetleri ve VHDL dilinin imkanları bir araya getirildiğinde birçok elektronik devre ile tasarlanabilecek böyle bir MPPT devresinin daha küçük ebatlarda tasarlanabilmesi büyük bir olanak olarak ortaya çıkmaktadır.

FPGA'ların tekrar düzenlenebilir sayısal işlem yapabilmelerinin anlamı tasarım aşamasında, diğer fiziksel devrelerde yapılma olasılığının olmadığı, değiştirmeleri daha rahat yapılabilir olmasıdır. FPGA programlanabilen bir lojik elemandır ve yazılım gibi esnek tasarım önerirler. Tekrar düzenlenebilirlik yeteneği üretildiği zamandan itibaren sonsuz kezdır. FPGA'ler donanım tasarımı için ilk örnek olarak kullanılabilirler. Bununla birlikte FPGA teknolojisindeki artış ile birlikte bu elemanların tekrar düzenlenebilirlik isteyen hesaplamalarda gittikçe artan sıklıkta kullanılmaya başlandı.

Uygulamaya başlamadan önce sorulması gereken soru bir PV devresinde bu kadar hızlı çalışan bir elemanın kullanılmasının gerekliliğidir. Çünkü atmosferik değişimler her ne kadar sürekli olsa da FPGA hızında olmamaktadır. Bununla birlikte FPGA dijital işlem tabanlı olduğu için PV güç sistemlerinden alınacak değerlerin analog olması, sistemle FPGA arasındaki iletişimin Analog sayısal dönüştürücülerle yapılması kaçınılmaz olacaktır. Bununla birlikte alınan verilerin işlenmesi ve hassasiyetin sağlanması bakımından FPGA kullanmak daha optimum olacaktır. Hassas hesaplamalar içeren uygulamalar için 32 bit kayan noktalı sayılar ideal duyarlılık olabilir. Fakat uygulamadaki PV sistemin güç denetimi için 16 bit kayan noktalı sayılar yeterli duyarlılıkta olmaktadır. Yüksek hassasiyetli uygulamalar için 32 yada 64 bit kayan noktalı sayılar kullanılabilir. Fakat FPGA'daki sınırlı donanım kaynakları kayan noktalı sayılar yerine daha az donanım kaynağı tüketen sayı sistemlerinin (örneğin sabit noktalı sayılar) kullanımını zorunlu kılar[16]. Sistemin performansının ölçülmesinde örnek PV kullanıldı ve bu PV modelinden elde edilen veriler MPPT algoritmasına göre FPGA da işlenerek referans çıkışlar alınarak bu referans değerlere göre maksimum noktaya artırım miktarına göre sistem düzeltilmektedir. Bu düzeltme işlemi maksimum nokta yakalanana kadar devam etmekte ve bu noktada sistemi tutmaya devam etmektedir. Işınım şiddetindeki değişimler meydana geldiğinde sistem aynı şekilde maksimum noktayı yakalamak için iterasyonlar yapmaktadır. Sistem hızlı çalışmakta ve değişimlere tepki 310 gibi çok kısa bir sürede tepki gösterip maksimum noktaya ulaşabilmektedir. Işınım şiddetindeki değişim miktarı düştükçe maksimum noktayı yakalama sistem için çok daha hızlı olmaktadır.

PV nin farklı ışınım şiddetlerinde ürettiği gerçek maksimum güç noktaları ile FPGA'nın yakaladığı maksimum güç noktaları arasında ki bağıl hatalar ise %1,6 ile % 4.03 arasında değişmektedir. Sistem ilk anda devreye sokulduğunda FPGA'nın maksimum noktayı tesbit etmek için yaptığı iterasyonlar fazla iken ilk maksimum noktayı yakaladıktan sonra PV sisteminde meydana gelen ışınım şiddetindeki değişimlerde yeni maksimum noktayı tespit etmesi çok daha kısa sürede olabilmektedir. Maksimum noktanın tespit edilmesinde algoritmada kullanılan ve kodların her bir yeni durumda sisteme verilecek her yeni referans değerinin hesaplanmasında kullanılan artırım ve azaltım miktarı olan C değeri olarak 0.1 gibi

sabit bir deęer olarak kullanılmaktadır. Hassasiyetin arttırılması ve maksimum noktaya daha bir yaklařılmak istenmesi durumu bu C deęerinin uygun seęilmesine de baęlıdır.

### **7.1 Veri Duyarlılıęı, Performans Nitelięi ve Alanın Verimli Kullanılması**

FPGA tabanlı MPPT uygulamaları için geręekleřtirilecek olan VHDL yazılımlarında ortaya ıkacak yoęunluk dezavantajlarını stesinden gelmenin bir yolu da donanım kaynakların etkin bir Őekilde kullanmaktır. FPGA sahip olduęu alanın en verimli Őekilde kullanılması nem tařımaktadır. Burada alan tasarrufunun anlamı optimal duyarlılıkta donanım alanı kullanımı ile beraber kalite performansından dn vermemektir. Bu yaklařım optimum duyarlılık ve performans kalitesinde minimum alan iřgal etmek anlamına gelmektedir. Tabi buda tasarımcının yazılım kabiliyetine baęlıdır.

Algoritmada ki deęerlerin duyarlılıęı sistemin doęruluk oranını arttıracaktır. Sisteme en uygun dŐecek minimum sayı duyarlılıęı, performans ve alan kullanımındaki verimlilięi kabul edilebilir dzeyde tutacaktır. Bu nedenle donanım için kullanılabilen duyarlılık oranı ve uygun sayısal gsterim genellikle uygulamanın Őekline ve kullanılan algoritmaya gre deęiřecektir. 16 bit duyarlılık FPGA tabanlı maksimum g noktası izleyici sistemi uzay uygulamaları için kullanılmıř ve bu dzeydeki duyarlılık yeterli bulunmuřtur[22].

### **7.2 zm Yntemi**

Sentezleme ařamasında, MPPT algoritmasının geręeklenmesi için kullanılan matematiksel formlleri ieren algoritmaların kullanılabilmesi VHDL'e ait donanım tanımlama ktphanelerinin kullanımı sayesinde olmaktadır. Burada tanımlanan FPGA tabanlı MPPT, PV den alınan akım ve gerilim deęerlerinin kullanılarak PV deki g deęiřimlerini tespit edebilecek hesaplamaları destekleyecek aritmetik HDL ktphanelerini yeterli derecede desteklemektedir[16].



### 7.2.1 Sayısal test ve karşılaştırma

FPGA tabanlı PV güç sisteminin MPPT uygulaması diğer işlemcilerle gerçekleştirilen MPPT'lerle karşılaştırıldığında ilk göze çarpan FPGA üzerinde gerçekleştirilen mimaride yer alan hesaplamaları çok hızlı yapabilmesidir. Sistemde 16 bitlik veri duyarlılığı ölçüm hassasiyetini çok daha üstün noktaya taşımaktadır. FPGA da PV den alınan akım ve gerilim değerlerine bağlı olarak sistemdeki değişimleri tespit etmek için kullanılan matematiksel fonksiyonlar paralel işlemler içerisinde gerçekleştiği için referans çıkışları çok daha etkin olarak alınmaktadır. Bunun yanında tasarımın elektronik devre bazında kapladığı alan ile birlikte maliyet açısından daha üstün özelliklere sahip olduğudur.

### 7.3 Sonuç

FPGA ile PV güç sistemlerinin tasarımı verim, maliyet, hız ve duyarlılık açısından olumlu sonuçlar ortaya çıkarmaktadır. Uygulama benzetiminde sistemin değişimlere cevabı veri girişi işlenmesi ve çıkış verilmesi bakımından 31 saat palsi süresince bu da kullandığımız FPGA Cihazımızın 100 MHz'lik özelliği ile 310 ns lik süre de olmaktadır. Bu bir PV sistemi için oldukça hızlı bir tepki olmaktadır. FPGA'nın yakaladığı maksimum güç noktaları arasında ki bağıl hatalar ise %1,6 ile % 4.03 arasında değişmektedir. Verilerin işlenmesinde kullanılan kayan noktalı aritmetik sistemi ile, sabit değerler (fixed Point) sistemine nazaran çok daha hassas hesaplamalar yapılabilmektedir. Algoritmanın işleyişinde yer alan yüksek yoğunluktaki matematiksel işlemler çok kısa bir sürede işlenmekte ve sonuçlar doğru bir şekilde alınmaktadır. Bunun yanında FPGA PV güç sistemleri ile entegre şekilde çalışabilmesi dışsal donanımlarla güçlendirildiğinde çok daha iyi sonuçlar alınması mümkün olabilmektedir. İleri düzeyde yapılacak çalışmalar açık bir alan olarak görülmektedir. FPGA'ların haberleşme sistemlerinde çok daha yaygın kullandığı bilinmekte fakat yenilenebilir enerji sistemlerinin çalışma verimliliklerinin artırılmasına dönük uygulamalarda henüz istenilen düzeyde olmadığı kanaati taşımaktayız. Donanımsal ve maliyet açısından bakıldığında hem tasarımcı açısından hem de kullanılacak malzeme bakımından daha geniş bir imkan sunmaktadır.

## KAYNAKLAR

- [1] EİE Genel Müdürlüğü, 2001, *Güneş Enerjisi ve Teknolojileri* [online], Elektrik İşleri Etüt İdaresi Genel Müdürlüğü, <http://www.eie.gov.tr/turkce/gunes/gunes.html> (**Ziyaret tarihi : 16.05.2008**)
- [2] SUNAN, Metin, “Güneş Enerjisi İle Çalışan Aracın Elektrik Ve Elektronik Sisteminin Mikrodenetleyiciler İle Tasarımı Ve Uygulaması”, Yüksek Lisans Tezi, *Kocaeli Üniversitesi Fen Bilimleri Enstitüsü*, Kocaeli, 11-14 (2006)
- [3] ATLAM, Özcan, “Alternatif Bir Fotovoltaik Panel (Pv) Modelinin Santrifüj Su Pompa Sistemlerine Uygulanması Ve Performans Gelişimine Yönelik Yaklaşımlar”, Doktora Tezi, *Kocaeli Üniversitesi Fen Bilimleri Enstitüsü*, Kocaeli, 11-14 (2004)
- [4] RAPPAPORT, P., The Photovoltaic Effect and Its Utilazation. Solar Cells” *in C.E. Backus (Editors) from RCA Rev., IEEE Press* , New York., 13, 1959
- [5] SZE, M.S., “Physics of Semiconductors Devices.By John Wiley.sons”, *Library of Congress*, America, Catalog Card Number:69-161321969.
- [6] BACKUS, C.E., “Principles of Photovoltaic Conversion. Solar Energy Technology Handbook” *in W.C. Dickinson and P.N. Cheremisinoff (Editors)*, New York and Basel London, 1980
- [7] TKAHASHİ, K., and KONAGAİ, M., “Amorphous Silicon Solar Cells” *in APPSF.F.R.D.(Translator) North Oxford Academic*, UK., 1986.
- [8] TREBLE, F.C., “Genarating Electricity From The Sun” *Pergamon Press, Oxford*,New York, Seoul, Tokyo, 1991
- [9] GOSWAMI, D.Y, KREITH, F., KREIDER J.F., “Principles Of Solar Engineering”, *ISBN No: 1-56032*, 714-6,UK, 1999
- [10] UYAREL, A. ve ÖZ, E.S.,” Güneş Enerjisi ve Uygulamaları”, *Birsen Yayınevi*, Ankara, 1987
- [11] GÜNEROĞLU, A., SUNAN, M. ve ATLAM, “Fotovoltaik Pillerin Değişken Işınım Ve Sıcaklık Değerlerine Bağlı Olarak Modellenmesi”, *Ulusal Teknik Eğitim Mühendislik ve Eğitim Bilimleri Genç Araştırmacılar Sempozyumu*, Kocaeli Üniversitesi, 205-208 Kocaeli, 20-22 Haziran(2007)

- [12] KIRAN, E. and INAN, D., “An Approximation To Solar Cell Equation For Determination of Solar Cell Parameters”, *Renewable Energy*, Vol.17, 235-241, 1999.
- [13] ALGHUWAINEM, S.M., “A Close –form Solution For The Maximum –Power Operating Point of A Solar Cell Array.”, *Solar Energy Meterials and Solar Cells*, Vol.46, 249-257, 1997.
- [14] KHAEHİNTUNG, N., WİANGTONG, T., SİRİSUK, P. “FPGA imlementation of MPPT using Variable Step-Size P&O Algorithm for PV Applicaions” *ISCIT 2006*
- [15] VILLASENOR, J., WILLIAM, H.M., “Reconfigurable Computing”, *Scientific American*, 1997
- [16] ŞAHİN, S., “FPGA İle Yapay Sinir Ağının Donanımsal Gerçeklenmesi”, Yüksek Lisans Tezi, *Kocaeli Üniversitesi Fen Bilimleri Enstitüsü*, Kocaeli, 25-55 (2004)
- [17] YALAMANÇİLİ, S. “VHDL Starter’s Guide”, *Prentice Hall*, 1-3, (1998)
- [18] NUR, T. “Tekrar Düzenlenebilir İşlem Katı Tasarımı”, Yüksek Lisans Tezi, *İ.T.Ü. Fen Bilimleri Enstitüsü*, İstanbul, 20-23, 2000
- [19] YALAMANÇİLİ, S. “VHDL Starter’s Guide”, *Prentice Hall*, 127-143, (1998)
- [20] HUA, C., LİN, J. and SHEN, C. “ Implementation of a DSP-Controlled Photovoltaic System with peak Power Tracking,” *IEEE Transactions on Industrial Electronics*, 45, 99-107, 1998
- [21] SALAS, V., OLÍAS, E., BARRADO, A., LAZARO, A., “Review of the maximum power point tracking algorithms for stand-alone photovoltaic systems” *Elsevier , Science Direct, Solar Energy Materials & Solar Cells*, Vol. 90, Page 1555-1578, 2006
- [22] PERSEN, T. E., B.S., “Fpga-Based Design Of A Maximum-Power-Point Tracking System For Space Applications” Doktora Tezi *University Of Central Florida*, 2004
- [23] Femia, N., Petrone, G. and Spagnuolo, G., Vitelli, M., “Optimization of Perturb and Observe Maximum Power Point Tracking Method”, *IEEE Transactions on Power Electronics*, Vol.20, 963-973, 2005

[24] GÜNEROĞLU, A., SUNAN, M. ve ATLAM, “FPGA Tabanlı Yenilenebilir Enerji Yönetim Sistemi Tasarımı”, *Ulusal Teknik Eğitim Mühendislik ve Eğitim Bilimleri Genç Araştırmacılar Sempozyumu*, Kocaeli Üniversitesi, 202-204 Kocaeli, 20-22 Haziran(2007)

## **ÖZGEÇMİŞ**

1975 yılında Mardin’de doğdu. İlk, Orta ve Lise öğrenimini Mardin’de tamamladı. 1994 yılında girdiği Kocaeli Üniversitesi Teknik Eğitim Fakültesi Elektrik Öğretmenliği Bölümün’den 1999 yılında mezun oldu. Halen Kocaeli Atatürk Anadolu Teknik, Teknik Lise ve Endüstri Meslek Lisesinde Koordinatör Müdür Yardımcısı olarak görev yapmakta olup; evli ve bir kız babasıdır.