

**ÇAĞDAŞ MİKROİŞLEMCİLERDE VERİ SAKLAYAN BİRİMLERİN  
SIZDIRMAYA BAĞLI GÜÇ TÜKETİMİNİN AZALTILMASI**

**YUSUF ONUR KOÇBERBER**

**YÜKSEK LİSANS TEZİ  
BİLGİSAYAR MÜHENDİSLİĞİ**

**TOBB EKONOMİ VE TEKNOLOJİ ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ**

**AĞUSTOS 2009**

**ANKARA**

Fen Bilimleri Enstitü onayı

---

Prof. Dr. Ünver KAYNAK

Müdür

Bu tezin Yüksek Lisans derecesinin tüm gereksinimlerini sağladığını onaylarım.

---

Doç. Dr. Erdoğan DOĞDU

Anabilim Dalı Başkanı

Yusuf Onur KOÇBERBER tarafından hazırlanan ÇAĞDAŞ  
MİKROİŞLEMCİLERDE VERİ SAKLAYAN BİRİMLERİN SIZDIRMAYA  
BAĞLI GÜÇ TÜKETİMİNİN AZALTILMASI adlı bu tezin Yüksek Lisans tezi  
olarak uygun olduğunu onaylarım.

---

Yrd. Doç. Dr. Oğuz ERGİN

Tez Danışmanı

Tez Jüri Üyeleri

Başkan :Yrd. Doç. Dr. Murat ÖZBAYOĞLU \_\_\_\_\_

Üye : Yrd. Doç. Dr. Oğuz ERGİN \_\_\_\_\_

Üye : Yrd. Doç. Dr. Ali BOZBEY \_\_\_\_\_

## **TEZ BİLDİRİMİ**

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, ayrıca tez yazım kurallarına uygun olarak hazırlanan bu çalışmada orijinal olmayan her türlü kaynağa eksiksiz atıf yapıldığını bildiririm.

Yusuf Onur KOÇBERBER

**Üniversitesi** : TOBB Ekonomi ve Teknoloji Üniversitesi  
**Enstitüsü** : Fen Bilimleri  
**Anabilim Dalı** : Bilgisayar Mühendisliği  
**Tez Danışmanı** : Yrd. Doç. Dr. Oğuz ERGİN  
**Tez Türü ve Tarihi** : Yüksek Lisans – Ağustos 2009

**Yusuf Onur KOÇBERBER**

## **ÇAĞDAŞ MİKROİŞLEMCİLERDE VERİ SAKLAYAN BİRİMLERİN SIZDIRMAYA BAĞLI GÜÇ TÜKETİMİNİN AZALTILMASI**

### **ÖZET**

Sınırlı pil ömrüne bağlı olarak çalışan taşınabilen aygıtlar düşük güç tüketimi ve uzun bekleme sürelerine ihtiyaç duymaktadırlar. CMOS devreler ise küçülen üretim teknolojileriyle birlikte her geçen gün daha çok sızdırmaya bağlı olarak durağan güç tüketmektedirler. Bu tez çalışmasında mikroişlemcilerin veri saklayan bileşenleri incelenmiştir. SRAM tabloları başarımları, güç tüketimi ve alan açısından ele alınmıştır.

Tezde önerilen teknik, mikroişlemcinin mimari durumunu ve dar değerleri kullanarak, veri saklayan bileşenlere güç ayrıştırılması uygulanmasıdır. Bu fikri uygulamak için yazmaç öbeği seçilmiştir. Yazmaç öbeği satırları mecazı anlamda uyutularak durağan gücün azaltılması sağlanmıştır. Uyutma işlemi mikroişlemcinin mimari durumuna ve dar değerlere bağlıdır. Dar değerler, mikroişlemcinin veri yolu genişliğinden daha az sayıda bit kullanarak gösterilebilen değerlerdir. Yazmaç öbeği satırının üst bitleri darlığa göre uyutulabilirken, mimari duruma bağlı olarak bir satır tamamen kapatılabilir. Bu işlemler yapılırken SRAM tablosunun fiziksel olarak devre serimi, yöntemin uygulanmasının kolaylığını azaltmaktadır.

Durağan enerji tüketimi %42 oranında azaltılırken, bu yöntemin bir yan ürünü olarak devingen enerji tüketimi de %32 azaltılmıştır. Mikroişlemcinin silikonda kapladığı alanı ve başarımları, yöntemin uygulanmadığı bir sistemle aynıdır. Böylece yöntemin getirdiği ek bir yük bulunmamaktadır.

**Anahtar Kelimeler:** Mikroişlemciler, Sızdırma akımı, Güç ayrıştırması, Yazmaç öbeği, SRAM, Dar değerler

**University** : TOBB Economics and Technology University  
**Faculty** : Institute of Natural and Applied Sciences  
**Department** : Computer Engineering  
**Supervisor** : Asst. Prof. Oğuz ERGİN  
**Degree Awarded and Date** : M.Sc. – August 2009

**Yusuf Onur KOÇBERBER**

**REDUCING STATIC ENERGY DISSIPATION OF DATA HOLDING  
COMPONENTS OF MODERN MICROPROCESSORS**

**ABSTRACT**

Mobile devices which are running on limited battery needs low power consumption and long stand-by duration more than ever. However modern microprocessors dissipate more static energy as CMOS process technology shrinks. In this thesis data holding components of the modern microprocessors are examined. SRAM tables are analyzed by means of performance, power consumption and area.

Thesis offers a power gating technique for data holding components which consists of SRAMs by exploiting narrow values and architectural state of the superscalar microprocessor. Register file is selected to prove our idea. Register file rows are literally put to sleep depending of their architectural state and narrowness. Narrowness indicates that a value can be shown by fewer bits than the data path width of the microprocessor. While upper order bits of a SRAM row are put to sleep depending on narrowness, a whole row can be put to sleep depending of the architectural state of the microprocessor. However physical layout of the SRAM tables prevents the straightforward implementation of the sleep decisions.

Static energy dissipation is reduced by %42 and as a byproduct of this technique; dynamic energy is reduced by %32. Silicon area and microprocessor performance is kept constant compared to the baseline microprocessor.

**Keywords:** Microprocessors, Leakage current, Power gating, Register file, SRAM, Narrow values

## **TEŐEKKÜR**

Çalıőmalarım boyunca deęerli yardım ve katkılarıyla beni yönlendiren hocam Yrd. Doç. Dr. Oęuz ERĖİN'e, yine kıymetli tecrübelerinden faydalandığım TOBB Ekonomi ve Teknoloji Üniversitesi Bilgisayar Mühendislięi ve Elektrik Elektronik Mühendislięi Bölümü öğretim üyelerine, benzetimlik kodlamalarında gece gündüz demeden yardımlarını esirgemeyen İlknur Cansu KAYNAK'a, tezimde bulunan şekillere sağladığı katkılarla Meltem ÖZSOY'a, mimari konusunda fikirleri ile aydınlatan Mehmet KAYAALP'e, beni her koşulda destekleyen aileme ve yüksek lisans bursumu karşılayan TÜBİTAK'a teşekkürü bir borç bilirim.

## İÇİNDEKİLER

|  | <b>Sayfa</b> |
|--|--------------|
| ÖZET   | iv           |
| ABSTRACT   | v            |
| TEŞEKKÜR   | vi           |
| İÇİNDEKİLER  | vii          |
| ÇİZELGELERİN LİSTESİ   | viii         |
| ŞEKİLLERİN LİSTESİ   | ix           |
| KISALTMALAR  | x            |
| SEMBOL LİSTESİ   | xii          |
| 1. GİRİŞ   | 1            |
| 1.1. CMOS Devrelerde Güç Tüketimi                              | 1            |
| 1.1.1. Devingen Güç Tüketimi                                   | 1            |
| 1.1.2. Kısa Devre Akımına Bağlı Güç Tüketimi                   | 4            |
| 1.1.3. Durağan Güç Tüketimi                                    | 4            |
| 1.1.4. Toplam Güç Tüketimi                                     | 7            |
| 2. VERİ SAKLAYAN BİLEŞENLER                                    | 9            |
| 2.1. Durağan Rastgele Erişimli Belleklerin Çalışma Prensipleri | 9            |
| 2.2. Rastgele Erişimli Bellek Tasarımı                         | 13           |
| 2.2.1. Bit Hücresi   | 13           |
| 2.2.2. Kod Çözücüler   | 16           |
| 2.2.3. Yazma Sürücüler   | 21           |
| 2.2.4. Ön Doldurucular   | 23           |
| 2.2.5. Fark Algılayıcılar                                      | 25           |

|        |  |    |
|--------|--|----|
| 2.3.   | Başarım, Güç ve Alan Analizi   | 28 |
| 2.3.1. | Başarım  | 28 |
| 2.3.2. | Güç Analizi  | 29 |
| 2.3.3. | Alan Analizi   | 35 |
| 2.4.   | Sızdırmaya Bağlı Durağan Güç Tüketimi Azaltma Yöntemleri                             | 37 |
| 3.     | MODERN MİKROİŞLEMCİLERİN YAPISI ve DAR DEĞERLER                                      | 42 |
| 3.1.   | Buyruk Yakalama ve Dallanma Tahmini  | 43 |
| 3.2.   | Buyruk Çözme, Yeniden Adlandırma, Yayınlama ve Yürütme                               | 44 |
| 3.3.   | Buyrukların Tamamlanması ve Çıkması  | 47 |
| 3.4.   | Dar Değerler   | 47 |
| 4.     | DAR DEĞERLERİ KULLANARAK VERİ SAKLAYAN BİLEŞENLERDEKİ ENERJİ TÜKETİMİNİN AZALTILMASI | 52 |
| 4.1.   | Giriş  | 52 |
| 4.2.   | İlgili Çalışmalar  | 52 |
| 4.3.   | Yazmaç Öbeğindeki Dar Değerler ve Yazmaçların Kullanım Ömürleri                      | 53 |
| 4.4.   | Yazmaç Öbeğine Devingen Olarak Güç Kesintisi Uygulanması                             | 55 |
| 4.5.   | Benzetimlik Ortamı   | 59 |
| 4.6.   | Sonuçlar   | 63 |
| 5.     | SONUÇ  | 71 |
|        | KAYNAKLAR  | 72 |
|        | ÖZGEÇMİŞ   | 76 |



## ÇİZELGELERİN LİSTESİ

| Çizelge  | Sayfa |
|--|-------|
| Çizelge 2.1. Devingen güç hesaplamalarında kullanılan katsayılar                                     | 33    |
| Çizelge 2.2. Durağan enerji hesaplamalarında kullanılan sabitler                                     | 34    |
| Çizelge 2.3. SRAM tablosunun bileşenlerinin fiziksel alanları  | 36    |
| Çizelge 3.1. Tek aşamalı ve iki aşamalı 0 algılayıcı devrelerin enerji tüketimi ve gecikme zamanları | 51    |
| Çizelge 4.1. Güç bloğu kontrol algoritması   | 58    |
| Çizelge 4.2. Benzetim parametreleri  | 59    |
| Çizelge 4.3. Tamsayı denektaşı programları   | 60    |
| Çizelge 4.4. . Kayan nokta denektaşı programları   | 61    |

## ŞEKİLLERİN LİSTESİ

| Şekil  | Sayfa |
|--|-------|
| Şekil 1.1. CMOS mantık devrelerinin genel gösterimi  | 2     |
| Şekil 1.2. CMOS eviricinin parasitik diyotlarını gösteren çizim                              | 5     |
| Şekil 1.3. CMOS eviricinin eşik altı sızdırma akım yolu                                      | 6     |
| Şekil 1.4. Bir mikroişlemcinin üretim teknolojisine göre değişen güç tüketim bileşenleri [6] | 8     |
| Şekil 2.1. 2 portlu SRAM hücresi   | 10    |
| Şekil 2.2. SRAM tablosu  | 12    |
| Şekil 2.3. 12 portlu bit hücresi fiziksel serimi   | 14    |
| Şekil 2.4. 8 portlu bit hücresi fiziksel serimi  | 14    |
| Şekil 2.5. 12 portlu bit hücresinin besleme hatlarının yer değiştirmiş tasarımı              | 15    |
| Şekil 2.6. 4 bitlik Kod çözücü ve kelime seçici hattın sürücüleri                            | 18    |
| Şekil 2.7. Kod çözünün normalize güç tüketimi  | 20    |
| Şekil 2.8. 256 satırlık kod çözücünün tek satırlık parçası                                   | 20    |
| Şekil 2.9. SRAM tablosunun bir satırında bulunan kod çözücüler, sürücüleri ve bit hücreleri  | 21    |
| Şekil 2.10. Yazma sürücüleri   | 22    |
| Şekil 2.11. Yazma sürücüleri ve ön doldurucuların bit hücresine yapılan bağlantıları         | 23    |
| Şekil 2.12. Ön doldurucunun devre şeması   | 24    |
| Şekil 2.13. Ön doldurucu devresinin fiziksel serimi  | 25    |
| Şekil 2.14. İki aşamalı fark algılayıcısı devre şeması                                       | 26    |
| Şekil 2.15. Farklı algılayıcının fiziksel devre serimi                                       | 27    |
| Şekil 2.16. SRAM tablosuna erişim sırasındaki giriş ve çıkış işaretleri                      | 28    |
| Şekil 2.17. 256 satır 64 kolonluk 12 portlu SRAM tablosu elemanlarının güç tüketimi oranları | 30    |
| Şekil 2.18. 32 Satır ve sütunluk 8 portlu SRAM elemanlarının güç tüketim oranları            | 31    |
| Şekil 2.19. Devrelerin uyutulması  | 38    |
| Şekil 2.20. Bit hücresinin zamana karşı harcadığı güç grafiği                                | 39    |
| Şekil 2.21. Bit hücreleri ve uyuma transistörlerinin fiziksel serimi                         | 40    |
| Şekil 3.1. Sırasız yürütüm yapan çok yollu mikroişlemcilerin genel mimarisi                  | 43    |
| Şekil 3.2. Yeniden adlandırma örneği   | 46    |
| Şekil 3.3. Dar değer gösterimleri  | 48    |
| Şekil 3.4. 8 bitlik 0 algılayıcı   | 49    |

|   |    |
|---|----|
| Şekil 3.5. Dar değer algılayıcı   | 49 |
| Şekil 3.6. İki aşamalı 24 bitlik 0 algılayıcı devre   | 50 |
| Şekil 4.1. Spec 2000 programlarının çalıştırılması sırasında yazmaç öbeğine yazılan değerlerin darlık yüzdesi           | 54 |
| Şekil 4.2. Spec 2000 programlarının çalıştırılması sırasında yazmaç öbeğinden okunan değerlerin darlık yüzdesi          | 54 |
| Şekil 4.3. Spec 2000 programlarının yazmaç öbeğindeki yazmaçların kullanım yüzdeleri                                    | 55 |
| Şekil 4.4. Yazmaç öbeğinin güç ayrıştırılması için düzenlenmesi   | 56 |
| Şekil 4.5. 34 Bit darlığa sahip yazmaç öbeği çiftlerinin açık, yarı açık ve kapalı olma yüzdeleri                       | 64 |
| Şekil 4.6. 31 Bit darlığa sahip yazmaç öbeği çiftlerinin açık, yarı açık ve kapalı olma yüzdeleri                       | 64 |
| Şekil 4.7. 16 Bit darlığa sahip yazmaç öbeği çiftlerinin açık, yarı açık ve kapalı olma yüzdeleri                       | 65 |
| Şekil 4.8. Yazmaç öbeğine yazılan ve yazmaç öbeğinden okunan 34 bitlik dar değer oranları                               | 66 |
| Şekil 4.9. Yazmaç öbeğine yazılan ve yazmaç öbeğinden okunan 31 bitlik dar değer oranları                               | 66 |
| Şekil 4.10. Yazmaç öbeğine yazılan ve yazmaç öbeğinden okunan 16 bitlik dar değer oranları                              | 67 |
| Şekil 4.11. 34 bit yapılandırmasında çalışan SPEC denektaşı programlarında sağlanan durağan ve dinamik enerji tasarrufu | 68 |
| Şekil 4.12. 31 bit yapılandırmasında çalışan SPEC denektaşı programlarında sağlanan durağan ve dinamik enerji tasarrufu | 69 |
| Şekil 4.13. 16 bit yapılandırmasında çalışan SPEC denektaşı programlarında sağlanan durağan ve dinamik enerji tasarrufu | 70 |
| Şekil 4.14. Güç tasarrufu uygulanması amacıyla uygulanan tüm yapılandırmaların enerji tasarrufu karşılaştırması         | 70 |

## KISALTMALAR

| <b>Kısaltmalar</b> | <b>Açıklama</b>  |
|--------------------|--|
| <b>CMOS</b>        | Complementary Metal Oxide Semiconductor<br>(Bütünleyici Metal-Oksit Yarıiletken) |
| <b>pMOS</b>        | P-Channel MOSFET<br>(P- kanallı MOSFET)  |
| <b>nMOS</b>        | N-Channel MOSFET<br>(N-kanallı MOSFET)   |
| <b>SRAM</b>        | Static Random Access Memory<br>(Rastgele Erişimli Durağan Bellek)                |
| <b>DRAM</b>        | Dynamic Random Access Memory<br>(Rastgele Erişimli Dinamik Bellek)               |
| <b>SPEC</b>        | Standard Performance Evaluation Corporation                                      |
| <b>RISC</b>        | Reduced Instruction Set Computer   |
| <b>CISC</b>        | Complex Instruction Set Computer   |
| <b>VLSI</b>        | Very Large Scale Integration   |
| <b>UMC</b>         | United Microelectronics Corporation  |
| <b>YSO</b>         | Yazdıktan Sonra Okuma  |
| <b>YSY</b>         | Yazdıktan Sonra Yazma  |
| <b>OSY</b>         | Okuduktan Sonra Yazma  |

## SEMBOL LİSTESİ

Bu çalışmada kullanılmış olan simgeler açıklamaları ile birlikte aşağıda sunulmuştur.

| <b>Simgeler</b> | <b>Açıklama</b>                                       |
|-----------------|---|
| $V_{DD}$        | Transistörlerin savaklarına bağlanan besleme gerilimi |
| $V_{SS}$        | Transistörlerin kaynaklarına bağlanan toprak gerilimi |
| $k$             | Boltzmann sabiti ( $1.38 \times 10^{-23}$ J/K)        |
| $q$             | Elektronik yük ( $1.602 \times 10^{-19}$ C)           |
| $T$             | Periyot   |
| $\alpha$        | Yunan Harfleri  |
| $\beta$         |   |
| $\gamma$        |   |
| $\rho$          |   |
| $\mu$           |   |
| $\phi$          |   |
| $\sigma$        |   |
| $\omega$        |   |
| $\lambda$       |   |

## 1. GİRİŞ

### 1.1. CMOS Devrelerde Güç Tüketimi

CMOS devrelerin güç tüketiminin 3 ana bileşeni vardır. Bunlardan ilki devingen güç tüketimidir. Bu bileşen CMOS devrenin bir mantık durumundan diğerine geçerken devredeki dolan, boşalan sığanın bir sonucudur. Toplam güç tüketiminde en büyük bileşendir.

İkinci güç tüketimi kısa devre akımlarından kaynaklanmaktadır. Kısa devre akımları da devingen güç tüketimine benzer olarak devrenin bir mantık durumundan diğerine geçmesi sırasında meydana gelir. Bu geçiş sırasında devrenin çıkış sığasının dolmasına etki etmeyen güç düğümü ve toprak arasında çok kısa süreli bir kısa devre akımı oluşmaktadır. Anahtarlanma sıklığı göz önüne alındığında bu küçük akım, güç tüketimine kayda değer katkılar yapabilmektedir.

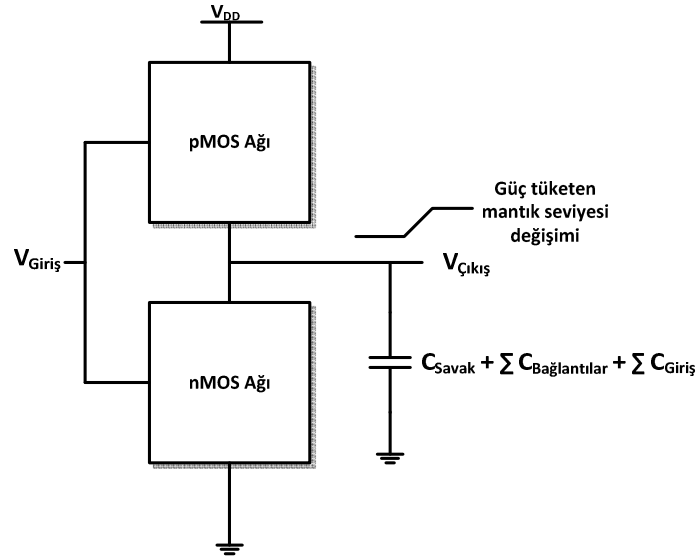
Son güç tüketim bileşeni ise ilk ikisinden farklı olarak durağan güç tüketimidir. Devrelere güç verildiği her durumda sızıntı akımları nedeniyle enerji harcanmasıdır.

#### 1.1.1. Dvingen Güç Tüketimi

Sayısal CMOS devrelerde çıkış düğümündeki sığayı doldurmak için güç kaynağından güç çekildiğinde enerji harcanır. Sığa 0'dan  $V_{dd}$ 'ye kadar dolarken toplam devingen enerjinin yarısı harcanır. Bu sırada kullanılan enerjinin bir kısmı pMOS ağındaki transistörlerde ısı olarak açığa çıkar. Benzer şekilde sığa  $V_{dd}$ 'den 0 Volt'a kadar boşalırken devingen enerjinin diğer yarısı harcanır. Harcanan bu enerji, sığanın boşalması sırasında bağlantılarda ve nMOS ağındaki transistörlerde açığa çıkan ısı enerjisidir.

Sayısal CMOS mantık devreleri genel bir şekilde Şekil 1.1'deki gibi gösterilebilir. pMOS transistörler iletme geçtiklerinde devrenin çıkış düğümünü  $V_{DD}$  gerilim seviyesine çekene kadar akım çekerek çıkış düğümündeki sığayı doldururlar. Bu sebepten bu ağa, yukarı çeken ağ (pull up network) denilmektedir. nMOS transistörler ise iletme geçtiklerinde çıkış sığasını boşaltarak toprak seviyesine yani 0 Volt gerilim seviyesine indirirler. Bu sebepten dolayı bu ağa aşağı çeken ağ (pull down network) adı verilmektedir.

Şekil 1.1'deki devrenin  $V_{Giriş}$  işaretinin periyodunu  $T$ , yükselme ve düşme zamanlarının sıfır olarak kabul ederek, ortalama devingen güç tüketimini hesaplanabilir. Bunun için,



Şekil 1.1. CMOS mantık devrelerinin genel gösterimi

$$C_{yük} = C_{savak} + \sum C_{Bağlantılar} + \sum C_{Giriş} \quad (1.1)$$

Çıkış düğümüne yansıyan tüm sığa denklem 1.1'deki hesaplanır. Burada  $C_{Bağlantılar}$  ile gösterilen sığa, bu devrenin çıkışının bağlandığı diğer devrelerin girişlerine kadar giden tellerin ve bu ağın tüm iç bağlantılarının sığalarının toplamıdır.  $C_{giriş}$  ile gösterilen sığa ise devrenin çıkışının bağlandığı diğer devrelerin girişlerinden görülen sığaların toplamıdır. Sığaların zamana bağlı geriliminden türetilen akım

denklemlerinden ve gerilimin akım ile çarpılmasından yola çıkarak ortalama güç denklem 1.2' deki gibi elde edilir.[1]

$$P_{ort} = \frac{1}{T} \left[ \int_0^{T/2} V_{\text{Çıkış}} \left( -C_{Yük} \frac{dV_{\text{Çıkış}}}{dt} \right) dt + \int_{T/2}^T (V_{DD} - V_{\text{Çıkış}}) \left( C_{Yük} \frac{dV_{\text{Çıkış}}}{dt} \right) dt \right] \quad (1.2)$$

Denklem 1.2'deki integral alındığında çok bilenen şu denklem elde edilir.

$$P_{ort} = \frac{1}{T} C_{Yük} V_{DD}^2 \quad (1.3)$$

Denklem 1.3'de payda da bulunan periyot terimi paya alınırsa,

$$P_{ort} = C_{Yük} V_{DD}^2 f_{saat} \quad (1.4)$$

Denklem 1.4'den de açıkça görülebileceği üzere, ortalama güç tüketimini azaltmak için (i) devredeki toplam sığayı azaltmak, (ii) besleme gerilimi  $V_{DD}$ 'yi azaltmak ve (iii) saat sıklığını azaltmak sonuç verecektir.

Devredeki toplam sığayı azaltmak tasarımcının elinde olan bir tasarım ölçütüdür. Görüldüğü üzere güç denklemlerinde yükselme ve düşme zamanlarıyla ilgili terimler yoktur fakat tasarımlardaki zaman ve alan kısıtlarından dolayı devrelerin gecikmesi önemlidir. Bu sebepten dolayı besleme gerilimi azaltma yöntemi düşük güçlü CMOS mantık devreleri için en genel kullanılan yöntemlerden biridir.  $V_{DD}$ 'yi azaltmak üretim teknolojisine bağlı olabileceği gibi bazı tasarım bloklarına güç anahtarlama yapmak gibi çalışma sırasında uygulanan devingen yöntemlerde olabilir. Besleme gerilimi azaltılırken devrenin gecikmesinin kayda değer miktarda artacağı ve diğer devre bloklarıyla olan uyumunun bozulabileceğini göz önünde bulundurmak gereklidir. Son olarak saat anahtarlama yöntemi ile devre bloklarının gerekmediği zamanlarda çalışmamasını sağlamakta popüler düşük güçlü CMOS tasarımı yöntemlerinden biridir. Bu şekilde sistemin yükü ne olursa olsun bazı bloklar çalışmadığından mantıksal geçiş azaltılır.

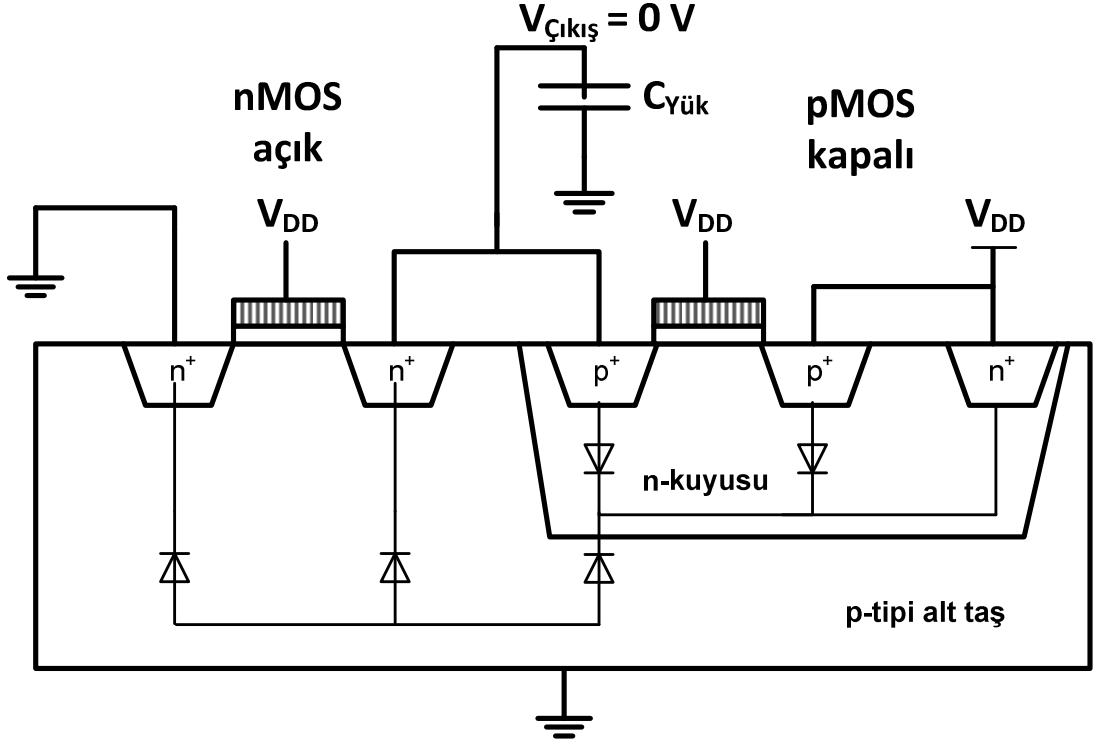


### 1.1.2. Kısa Devre Akımına Bağlı Güç Tüketimi

Bölüm 1.1.1.'de devrenin mantıksal durumunun değişmesi için gerekli olan enerjinin nerelerde tüketildiği açıklanırken, bu enerjinin giriş ve çıkış sinyalinin yükselme ve düşme zamanlarına bağlı olmadığı görülmüştü. Gerçek bir sayısal CMOS devresinde ise bu durum hiçbir zaman böyle olmamaktadır. Sonlu zamanlı yükselme ve düşme zamanına sahip bir giriş sinyalinde nMOS ve pMOS ağlarının aynı anda iletme geçip,  $V_{DD}$  ve toprak arasında düşük dirençli bir yol bulmasından dolayı kaynaktan çekilen akımda kısa süreli ciddi artışlar meydana gelmektedir. Bu akımın devre içindeki hiçbir sığayı doldurmamasından dolayı bu akıma kısa devre akımı denmektedir. Kısa devre akımlarının toplam tüketime oranı %20'den daha küçüktür fakat çıkış sığasının küçük oluşması ve giriş işaretinin yükselme ve düşme zamanlarının çok olması durumunda bu tüketim oranı devingen güç tüketimi ile aynı seviyelere gelebilir.[2]

### 1.1.3. Durağan Güç Tüketimi

Sayısal CMOS devrelerde kullanılan nMOS ve pMOS transistörlerin iletimde olmadığı durumlarda güç tüketmediği düşünülebilir fakat gerçekte bu durum böyle değildir. CMOS devrelerin Şekil 1.2'de görülen PN-kavşaklardan oluşan diyot modeli göze alındığında, savak ve alt taş (substrate) arasında oluşan parasitik diyot ters kutuplandığında sızdırmaktadır.



Şekil 1.2. CMOS eviricinin parasitik diyotlarını gösteren çizim

Şekil 1.2’de eviricinin giriş kapısında mantıksal işaret olarak 1, çıkışında ise 0 vardır. Bu durumda pMOS transistör kapalıdır fakat pMOS’un çıkış kapısında 0 Volt varken, n-kuyusu  $V_{DD}$  ile beslenmektedir. Burada oluşan diyot ters kutuplanmaktadır. Benzer şekilde n-kuyusu ve p-tipi alt taş arasında oluşan diyot, n-kuyusunun  $V_{DD}$ , alt taşın ise toprağa bağlı olmasından dolayı ters şekilde kutuplanmıştır. Şekil 1.2’deki durumun tam tersi olarak, giriş kapısına 0 Volt uygulanıp, çıkış kapısında  $V_{DD}$  gerilimi görüldüğü durumda nMOS transistör kapanır fakat alt taş ile nMOS savağı arasında oluşan diyot ters kutuplanır. PN kavşağının N kısmı nMOS’un savağı, P kısmı ise p tipi alt taştır. Tüm bu parasitik diyotların sızdırması, diyot denkleminde aşağıdaki şekilde ifade edilebilir.[3]

$$I_{ters} = I_s \left( e^{qV/kT} - 1 \right) \quad (1.5)$$

Denklem 1.5 ’de  $I_s$  doyma akımı,  $V$  ise diyota uygulanan ters gerilimdir. Elbette bu akım çok düşük bir değere sahiptir fakat bir yongada milyonlarca transistör olduğu göze alındığında toplam güç tüketimine önemli bir katkıda bulunmaktadır.



artışa karşılık, kapı yalıtkanı sızdırması 2 kat artmaktadır [4]. Mikron altı teknolojilerde her geçen gün daha büyük bir sorun haline gelen bu sızdırma türünü engellemek amacıyla kapı yalıtkanı olarak uzun bir süredir araştırılan Yüksek-K (High-K) dielektriği yalıtkan türü 45nm kapı genişliği teknolojisiyle birlikte sadece güç ve performans değil ticari açıdan da başarılı bir şekilde üretilebilmiştir [5].

Yukarıda bahsedilen etkenlerin tümü bir araya toplandığında sızıntı akımı elde edilmektedir.

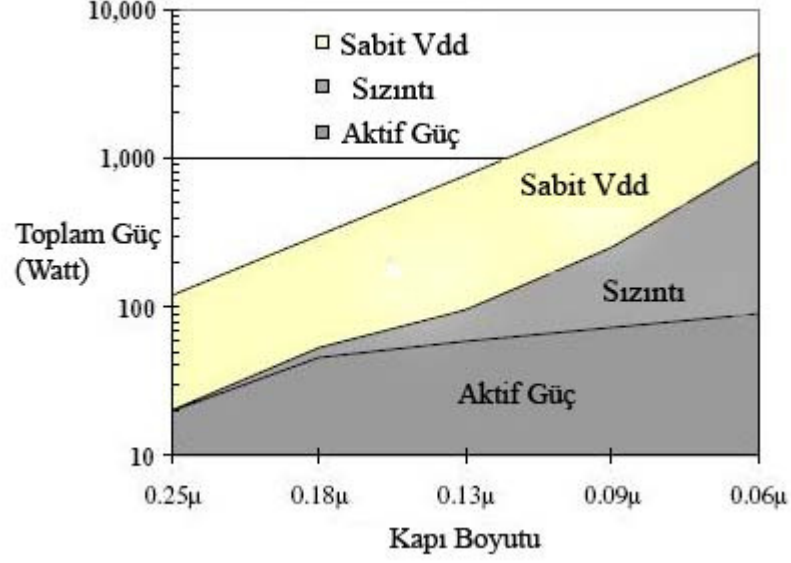
#### 1.1.4. Toplam Güç Tüketimi

CMOS transistörlerin toplam tükettiği güç Bölüm 1.1.1, 1.1.2 ve 1.1.3’de anlatılan tüm bileşenlerin toplamı olarak ifade edilebilir. Buna göre,

$$P_{Toplam} = \alpha_T C_{Yük} V_{DD}^2 f_{Saat} + V_{DD} (I_{kisa\ devre} + I_{sızıntı} + I_{DC}) \quad (1.6)$$

Denklem 1.6’deki  $\alpha_T$  terimi, her bir saat vuruşunda mantıksal durumunu değiştiren düğümlerin oranıdır[1]. Bu şekilde devingen güç doğru bir şekilde hesaplanabilir. En kötü durum için bu terim 1 olarak düşünülebilir. Önceki bölümlerde açıklanmayan bir diğer terim olan  $I_{DC}$  ise kaynaktan çekilen doğru akım bileşenidir.

CMOS devrelerin üretim teknolojisi küçüldükçe sızıntı akımı önemli bir problem haline gelmektedir. Şekil 1.4’de görüldüğü üzere aktif yani devingen güç bir mikroişlemcide en çok güç harcayan bileşendir. Buna rağmen üretim teknolojisinin küçülmesiyle çok büyük değişimler göstermemektedir.



Şekil 1.4. Bir mikroişlemcinin üretim teknolojisine göre değişen güç tüketim bileşenleri [6]

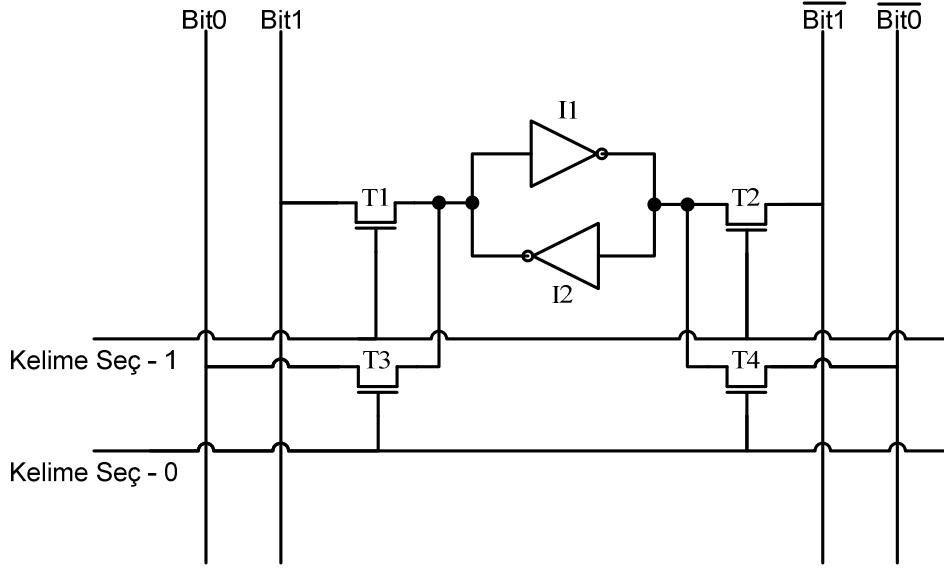
Güç tüketimi denklem 1.6'da görüldüğü üzere  $V_{DD}$ 'nin karesiyle orantılı olduğundan Şekil 1.4'deki sabit  $V_{DD}$  grafiği, küçülen teknoloji karşısında  $V_{DD}$  geriliminin küçülmemesi durumunda diğer tüm güç tüketimi azaltma çabalarının ne kadar yetersiz kalacağını bir göstergesidir. Üretim teknolojisine en bağlı bileşen ise sızıntı akımları ve buna bağlı harcanan güçtür. Mikron altı adı verilen 0,1 mikrondan daha küçük teknolojilerden itibaren sızıntı akımı katlanarak artmaktadır. Moore Yasası'nın devamıyla birlikte gelecek teknolojilerinin en büyük problemi sızıntı olarak gösterilmektedir [4],[7].

## 2. VERİ SAKLAYAN BİLEŞENLER

### 2.1. Durağan Rastgele Erişimli Belleklerin Çalışma Prensipleri

Durağan rastgele erişimli bellekler (SRAM) mikroişlemcilerdeki temel veri saklayan birimlerdir. Yazmaç öbeği, yayın kuyruğu gibi işlemcinin en yoğun birimleri SRAM tablolarından oluşmaktadır. Bu tip belleklerin istenilen yerine sabit bir gecikme ile ulaşılabilindiğinden bu ismi almışlardır. Örneğin, sabit teker gibi veri saklayan başka bir aygıtın belirli bir alanına ulaşılacak istenildiğinde fiziksel olarak sabit tekerin okuyucu kafasının istenilen hücreyi bulması gerekmekte ve erişim zamanı değişken olmaktadır. Devingen rastgele erişimli belleklerden (DRAM) farklı olarak üzerindeki verileri kaybetmemek amacıyla verilerin tazelenmesine ihtiyaç duymaz. Gücü kesilmediği sürece içindeki bilgi kaybolmadığı için bu bellek tipine durağan denmektedir.

SRAM tablolarına erişim hızı sabit ve hızlıdır. Veri saklayan bu tablolara erişim zamanı, işlemcinin hız açısından kritik yollarının üzerinde bulunduğundan dolayı, önemlidir. Tablolara erişim zamanındaki bir artış işlemcinin frekans hızını doğrudan etkilemektedir. Bununla birlikte modern mikroişlemcilerde bir SRAM tablosu olarak bulunan yazmaç öbeği tüm işlemcinin enerji tüketiminin %20'ye yakını tüketmektedir[8]. Bundan dolayı SRAM tablolarının tasarımı çok önemlidir. SRAM'den oluşan belleklerin en temel birimi SRAM hücresidir. Şekil 2.1'de iki portlu bir örneği görülen SRAM hücresine iki amaçla erişmek mümkündür. Bunlardan birincisi içinde saklı olan bilgiyi okumak, ikinci ise var olan yerine yeni bir bilgi yazmaktır. SRAM hücresi sadece 1 bit saklayabilmektedir. Bu hücreler satırlar ve sütunlar halinde yan yana, alt alta dizilerek veri saklayan birimleri oluştururlar.



Şekil 2.1. 2 portlu SRAM hücresi

Şekil 2.1.'de görülen I1 ve I2 eviricileri arka arkaya bağlanmışlardır. Bu sayede saklanacak olan bit I1'in giriş düğümünde I2'nin çıkış düğümünde dururken bitin tersi I1'in çıkışında, I2'nin giriş düğümünde durarak iki eviricinin birbirini sürekli olarak beslemesi sağlanmaktadır. Devreye güç verildiği sürece buradaki veri tazelenmeden saklanabilmektedir. Bu veriye ulaşmak için "Kelime Seç" (Word Select) hatları kullanılmaktadır. Bu hatlara bağlı olan T1,T2 ve T3,T4 nMOS geçiş transistörleri ayrı iki port olarak adlandırılır. Hücreler yan yana dizildiğinde bu hatlar birbirlerine bağlanır. Hücreler alt alta sıralar halinde dizildiğinde ise "Bit hatları" birbirlerine bağlanır. Hücrelere birden çok port koyarak aynı anda farklı satırlardaki hücreleri okumak mümkün olabilmektedir. Örneğin, kelime seç-1 hattından hücreye erişmek istenildiğinde bu hat aktive edilerek T1 ve T2 transistörleri açılır. Bu sayede saklanan veri "Bit1" ve "Bit1 değil" hattına aktarılmış olur. Aynı anda başka bir sıradaki bit hücre dizisinde kelime seç-0 hattı aktive edilerek Bit0 hattına istenilen veri alınabilir.

Bit hücresinin içerisindeki veriyi değiştirmek için, okumaya benzer şekilde hücreye bağlı kelime seç hatlarından yalnızca biri aktive edilir. Okumadan farklı olarak bit hatlarına yazılmak istenilen veri önceden hazırlanır. Kelime seç aktive olur olmaz bu veri bit hatlarından hücrenin içine yazılır. Örneğin, hücreye mantıksal olarak 1 verisi yazılmak isteniyorsa, 2 portlu bir bit hücresinin herhangi bir kelime seç hattı aktive olmadan önce o hata bağlı bit hattına 1 verisi, bit değil hattına ise 0 gönderilir.

Böylece ilgili bit hatlarında 1 ve 0 verisi varken kelime seç hattının geçiş transistörlerini açmasıyla veri hücreye yazılır. Veri yazılımı arka arkaya eviricilerde bulunan aşağı ve yukarı çeken transistörlerin zorlanarak açılması veya kapanması prensibine dayanmaktadır.

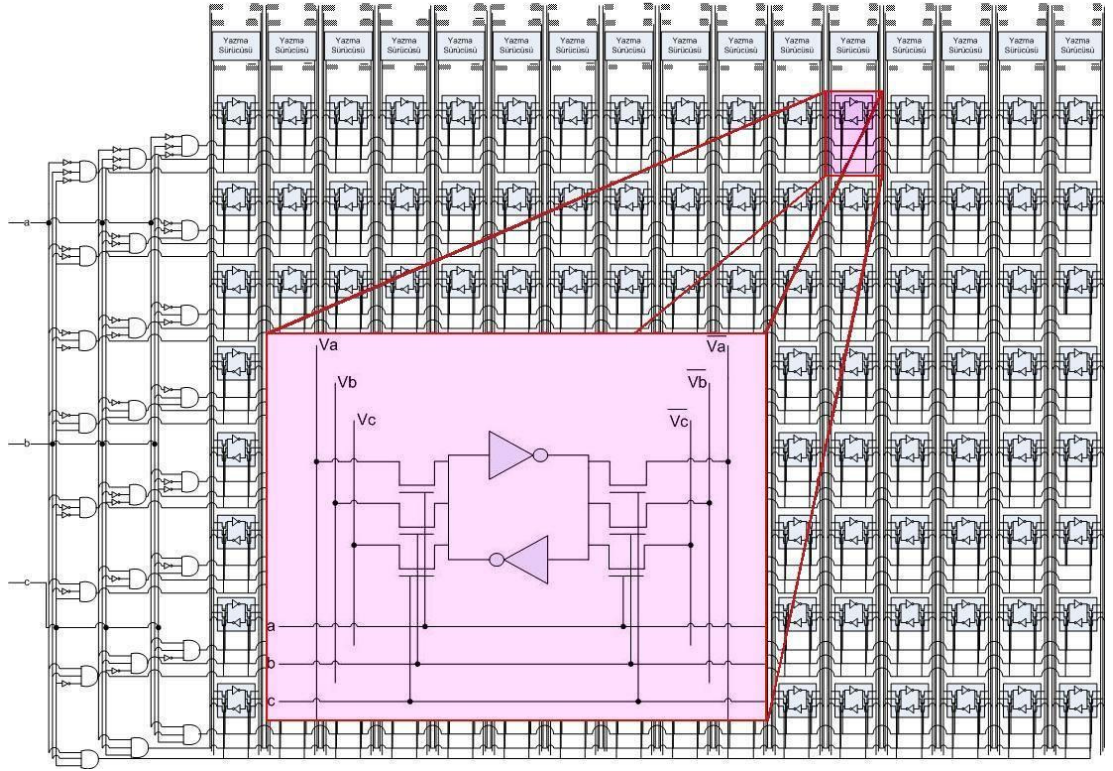
Bit hücrelerine eklenen portlar okuma ve yazma için büyük bir esneklik sağlamaktadır. Örneğin, aynı anda 4 buyruğun işlendiği modern mikroişlemcilerde aynı anda 4 farklı yazmaç değeri okunmak istenmektedir. Eğer bit hücreleri sadece 2 porta sahip olsaydı 4 değeri okuma işlemi diğer saat vuruşuna kalacaktı fakat bu durumda 4 farklı kelime seçme işlemi ile istenilen veriler okunabilmektedir. Bunun yanında yazma işlemleri ve okuma işlemleri için bit hatlarına farklı devre elemanları bağlanmaktadır. Aynı portlardan hem okuma hem yazma yapmak karmaşıklığa yol açacağından sayıca birden çok olan portları okuma ve yazma olarak ayırmak mümkündür. Aynı anda bir hücre birden çok port tarafından okunabilir fakat birden çok port tarafından yazılamaz. Bunun nedeni, aynı anda iki farklı portun yazmak istediği verinin aynı olup olmadığının bilinmemesidir.

Portların sayıca çok olmasının avantajlarının yanında önemli dezavantajları da bulunmaktadır. Her port iki tane geçiş nMOS transistöründen oluşmaktadır bunun yanında fazladan iki tane bit teli ve bu tellerin çalışmasını sağlayan çevresel devrelerde tabloya eklenmektedir. Ayrıca her bir portun nMOS transistörleri bit hücresindeki arka arkaya bağlanmış eviricilerin girişine ve çıkışına fazladan sığa bindirmekte, kelime seçicilerinde benzer bir şekilde sığasını artırmaktadır. Hücrenin fiziksel olarak büyümesine neden olduğundan bit telleri ve kelime seçici teller de uzamaktadır. Portlar SRAM tablosuna erişim zamanını direk olarak etkilemektedir [9], [10].

Şekil 2.2.'de bit hücrelerinden oluşturulan büyük bir tablo gözükmektedir. Tablodaki her bir satır kelime uzunluğunu ifade etmektedir. Modern mikroişlemcilerde kelime uzunluğu 64 bittir fakat bazı bileşenlerinde 32bitlik kısımlar da bulunmaktadır. Tablonun solunda kod çözücüler bulunmaktadır. Bu kod çözücüler gelen adresi çözerek ilgili satırın kelime seçme hattını aktive ederler. Bir tabloda ne kadar çok satır bulunursa kod çözücüler o kadar büyük olur. Bununla birlikte, enerji tüketimi ve gecikme de artar.



Tabloların satırlarının artmasının bir diğer etkisi de bit hatlarında görülmektedir. Yukarıdan aşağıya uzanan bu teller uzadıkça bit hücrelerinin gördüğü sığa artmaktadır. Bu hem güç tüketimini hem de gecikmeyi artırmaktadır. Bu sebepten dolayı bu hatlar tablonun en altlarında fark kuvvetlendiricilere bağlanmaktadır. Fark kuvvetlendiriciler bit hattı ve bit değil hattında oluşan ufak bir farkı hızlıca hissederek veriyi anlamlandırır. Bit tellerinin uzamasının bir diğer kötü etkisi ise yazma sırasında, yazılacak verilerinin bu tellere hızlı bir şekilde verilmesi gerekmektedir. Bundan dolayı yazma sürücüler kullanılmaktadır. Teller uzadıkça sürücüler daha çok güç harcamaktadır.



Şekil 2.2. SRAM tablosu

Görüldüğü üzere bellek tasarımında gecikme ve harcanan güç gibi kritik tasarım ölçütleri bulunmaktadır. Bunlara bir de devrenin fiziksel alanı eklendiği zaman tasarım hedeflerine ulaşmak ciddi bir mühendislik çalışması gerektirmektedir.

## 2.2. Rastgele Erişimli Bellek Tasarımı

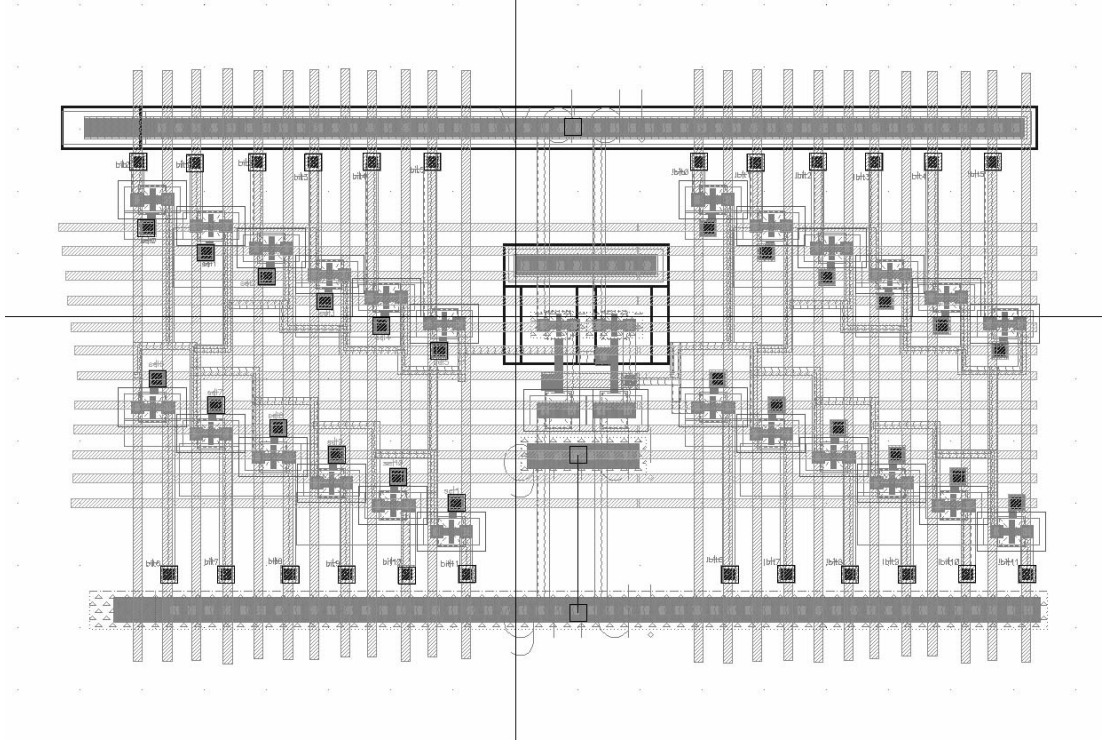
### 2.2.1. Bit Hücresi

Bit hücresi bir SRAM'in en temel elemanıdır. Tabloda sayıca en çok bulunan eleman olarak tablonun alanına en büyük etkiyi yaparlar. Mikron altı teknolojilerde az sayıda porta sahip bit hücrelerini nasıl tasarlanması gerektiği az çok bellidir fakat çok sayıda porta sahip olan ve mikroişlemci içinde kullanılan SRAM tabloları mikroişlemci üreten firmalar tarafından ticari sır olarak saklanmaktadır.

Bit hücresinde bulunan eviricilerin ve portları oluşturan geçiş transistörlerinin boyutları çok önemlidir. Bu elemanların tümünde kullanılan transistörler teknolojinin izin verdiği en küçük kapı boyutunda tasarlanmaya çalışılır.

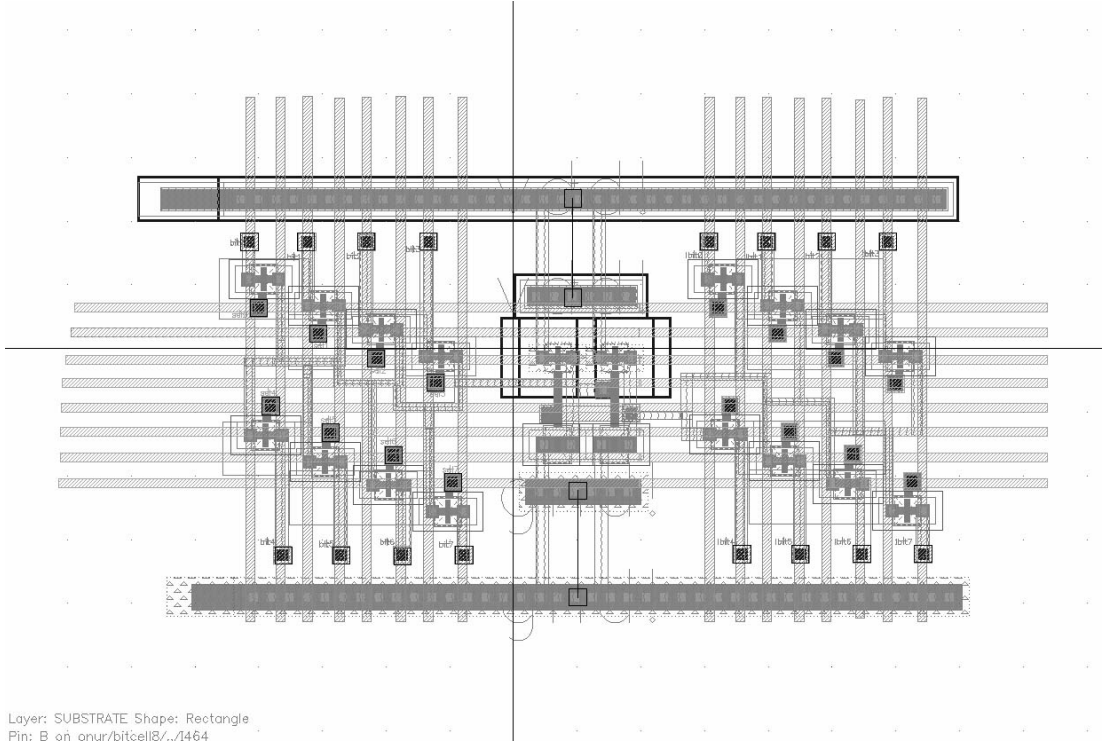
Bit hücresine yazmak veya okumak amacıyla yapılan her erişimde portları oluşturan geçiş nMOS transistörleri ile eviricilerin nMOS transistörleri arasında bir yük paylaşımı olmaktadır. Bu nedenle eviricilerin aşağı çeken nMOS transistörlerinin geçiş transistörlerden en az 2 kat daha büyük olması gerekmektedir. Yukarı çeken transistörlerin ise olabilecek en küçük boyutta olması gerekmektedir. Geleneksel evirici boyutlarının tam tersi olan bu oranlar, tez çalışması sırasında tasarlanan devrelerin benzetim sonuçlarına dayanılarak bulunmuştur. 90nm ile çizilen SRAM hücrelerinde bu boyutların en iyi performansı verdiği rahatlıkla söylenebilir.

Tek portlu SRAM bit hücreleri için birkaç tane fiziksel serim şekli bulunmaktadır fakat çok sayıda port bulunduran hücrelerde kullanılan metal hatlarının bir birlerine ne kadar yakın olabileceğine bağlı olarak hücrenin boyutu belirlenmektedir. Şekil 2.3'de 12 portlu bit hücresinin fiziksel serimi görülmektedir. Sağ ve solda portları oluşturan nMOS transistörler, ortada ise arka arkaya bağlanmış evirici bulunmaktadır. Hücrenin tasarımındaki temel zorluk portların kelime seçme tellerinin birbirlerine tasarım kurallarının izin verdiği en yakın mesafede konumlandırılmasıdır. Bir başka ölçüt ise, yukarıdan aşağıya inen bit ve bit değil tellerinin de birbirlerine minimum mesafe de tutulmasıdır. Gelişmiş CMOS üretim teknolojilerinde yeterli sayıda metal katmanı olduğundan dolayı, hücre seçme telleri ve bit telleri için farklı metal katmanları kullanılmıştır. Bit hatlarının metal temas noktaları hücrenin aşağı ve yukarı noktalarına alınarak yer tasarrufu sağlanmıştır.



Şekil 2.3. 12 portlu bit hücresi fiziksel serimi

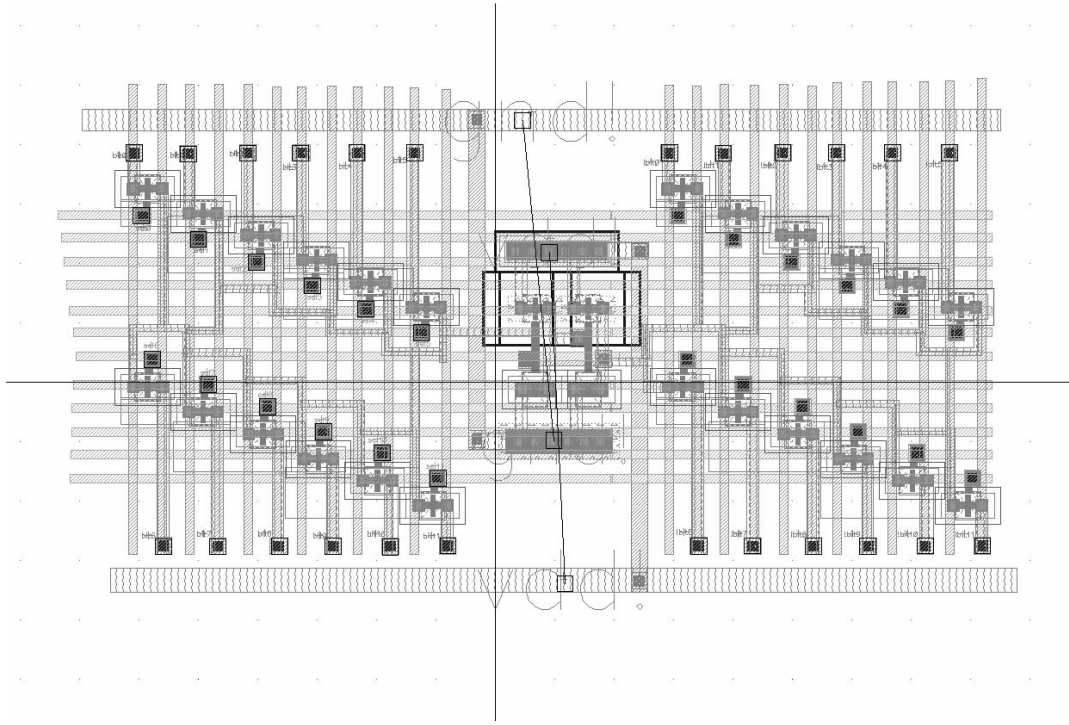
Şekil 2.4'de bit hücresinin 8 portlu fiziksel serimi görülmektedir. Fiziksel serim alanlarına göre 8 portlu bit hücresi 12 portluya göre %39 daha küçük bir alan kaplamaktadır.



Layer: SUBSTRATE Shape: Rectangle  
Pin: B ön onur/bitcell8/./1464

Şekil 2.4. 8 portlu bit hücresi fiziksel serimi

Bit hücreleri alt alta dizilirken yukarıda bulunan  $V_{DD}$  hattı ile aşağıda bulunan Gnd (toprak) hattı problem olmaktadır. Eğer hücreler bu şekilde alt alta dizilecekse, teknolojinin izin verdiği en kısa metal yakınlığı kullanılacak olmasına rağmen arada oluşacak boşluk alan kaybına yol açacaktır. Bu boşluğun oluşmaması için bir bit hücresinin altına aynı bit hücresinin  $V_{DD}$  hattı aşağıda, Gnd hattının ise yukarıda olduğu bir tasarımı kullanılmalıdır. Şekil 2.5’de 12 portlu hücrenin  $V_{DD}$  hattı aşağıdan, Gnd hattı ise yukarıdan geçmektedir. Bit hücresinin güç bağlantıları aynı kalmasına rağmen yerel olarak kullanılan metal-2 katmanı sayesinde güç ve toprak bağlantıları istenilen noktalara taşınmıştır. Bu tasarımdaki bir diğer fark ise  $V_{DD}$  hattının altında bulunan n-kuyusu bağlantısı ve toprak hattında bulunan alt taş bağlantısı yapılmamıştır. Bu sayede hücreler alt alta dizildiği zaman n-kuyusu ve alt taş bağlantıları yukarıdaki hücre ile sağlanacaktır.



Şekil 2.5. 12 portlu bit hücresinin besleme hatlarının yer değiştirmiş tasarımı

Güç ve toprak hatlarının her bir satırda iki bit hücreleri tarafından paylaşılması sayesinde alandan ciddi bir alan tasarrufu sağlanmaktadır. Buna rağmen sızıntı akımını azaltma amacıyla güç kesilmesi gibi yöntemler uygularken, bir güç hattının kesilmesi iki satırında gücünü kesmektedir. Bu tip bir tasarım kriteri göz önünde

bulundurularak devrelerin çalışması sırasında dinamik olarak güç anahtarlanacak ise veri kayıplarının önüne geçilmesi amacıyla mimari düzeyinde değişiklikler yapmak gerekebilir. Bu durum bölüm 4.4.'de uygulanan güç kesme yönteminde göz önüne alınarak hareket edilmiştir.

### 2.2.2. Kod Çözücüler

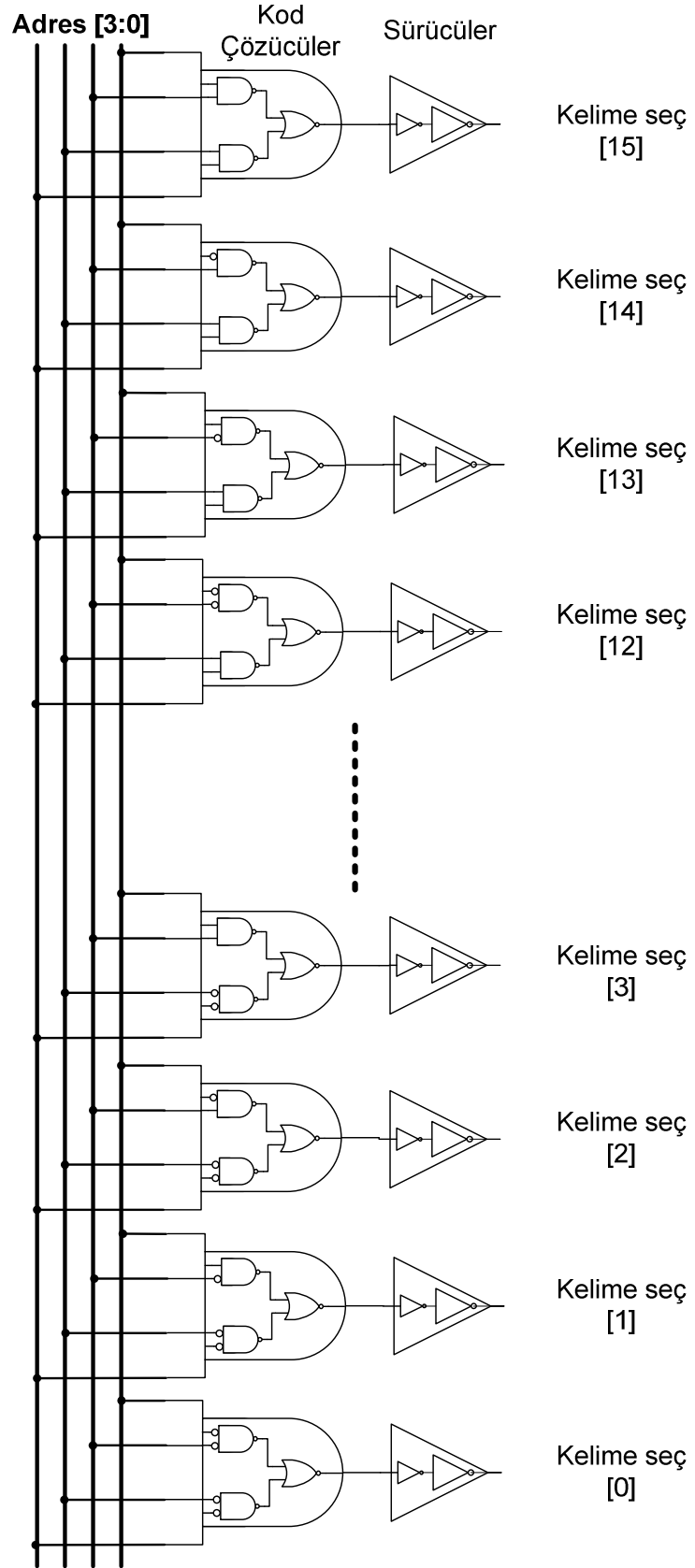
SRAM tablosundan bir kelime okumak için tablonun tüm satırını seçmek gerekmektedir. Bir portun tüm kelime seç telleri birbirine bağlı olduğundan sadece seçilecek satırı saptamak yeterlidir. İlgili satırın kelime seç teline 1 verildiğinde nMOS geçiş transistörleri açılarak bit hücrelerini aktive eder.

SRAM tablosu 0'dan başlayarak adreslenir, ilk satır 0 adresine sahiptir. Bundan sonraki her satırın adresi 1 artarak gider. SRAM tablosuna ulaşacak mikroişlemci ilgili adresi verir, bu adres kod çözücünden geçerek istenilen satırın kelime seçme tellerini aktive eder. Kod çözücüye giriş olarak verilen adresler 2'lik sayı sistemindedir. Örneğin 16 satırı adresleyebilen bir kod çözücüye 4 tane adres teli giriş olarak verilmektedir. Her bir satırın başında o satırın adresini kabul eden kod çözücüler bulunmaktadır.

Kod çözücüler değişik CMOS mantık yöntemleriyle tasarlanabilirler. Alan olarak en çok yer tutan fakat kararlı ve kontrolü kolay olması nedeniyle kombinasyonel mantık devreleri ile bir tasarım yapılmıştır. Bu kod çözücü devreler aslında büyük bir VE kapısı olarak düşünülebilir. Eğer giriş tellerinin hepsi doğru mantık seviyesinde ise bu VE kapısı çıkışına 1 vererek ilgili satırı seçer.

Şekil 2.6.'da 16 satırı adresleyebilen bir kod çözücü mantıksal kapı düzeyinde gösterilmiştir. Her bir satırın başında büyük bir VE kapısı bulunmaktadır. Başarım ölçütleri açısından burada çok girişli büyük bir VE kapısı yerine, aynı işlevi gerçekleştiren arka arkaya bağlanmış VE DEĞİL ve VEYA DEĞİL kapıları kullanılmıştır. 4 girişli bir VE kapısı kullanmak pratikte uygulanmamaktadır. Bunun nedeni ise tasarımın küçük kapılara ve 2 aşamaya bölünerek ölçeklenebilir olması sağlanırken, güç tüketimi ve gecikme açısından da daha iyi sonuç vermesidir. 256 satırı adresleyen büyük bir VE kapısı düşünüldüğünde ölçeklenebilir bir tasarımın önemi daha iyi anlaşılabilir.

Her satırın başındaki kod çözücü kapıların hepsine aynı adres telleri giriş olarak verilmektedir fakat bu kapıların girişleri birbirinden farklıdır. Örneğin, en büyük adresi kabul eden 15 numaralı satırın kod çözücüsünün giriş telleri direk kapıya verilmektedir. 14 numaralı kapının adresi kabul etmesi için giriş bitlerinden en anlamsız olanı evirilerek devreye verilmektedir. Bu sayede 1110 olan 14 numaralı adresin en anlamsız biti 0 evirilerek 1 olacak ve VE kapısını aktive edebilecektir. Benzer şekilde 0 numaralı satırın giriş tellerinin hepsinin evirilerek girişe verildiği Şekil 2.6'da görülmektedir.



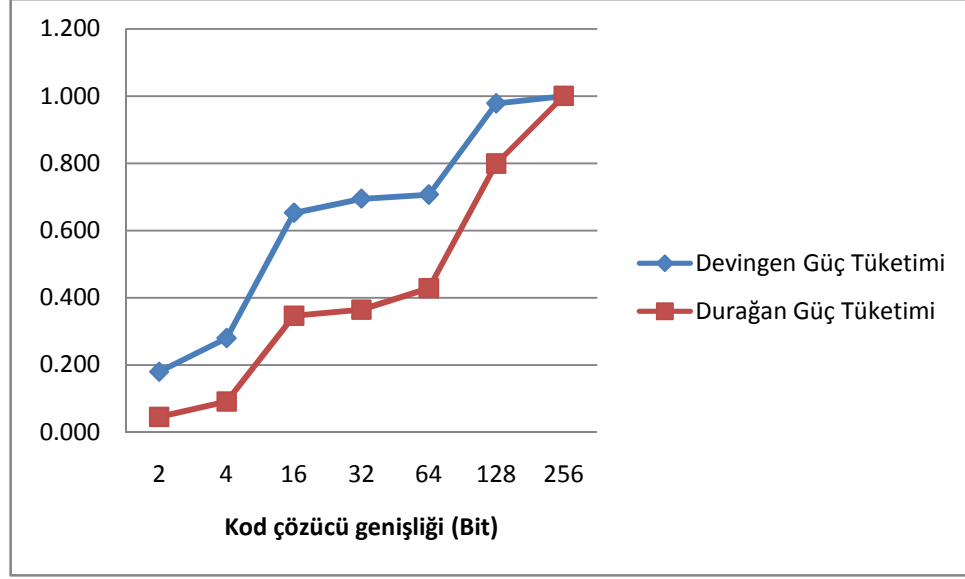
Şekil 2.6. 4 bitlik Kod çözümler ve kelime seçici hattın sürücüleri

SRAM tablolarının kelime uzunluđu yani bir satırın uzunluđu arttıkça, kelime seçme tellerinin de uzunluđu artar. Bununla birlikte tel üzerine daha çok nMOS geçiş transistörü bağlanacağından transistörlerin geçit sığaları da tele eklenir. Kod çözücünün bu teli sürmesi için içindeki transistörlerin boyutlarının artırılması gerekmektedir. Büyüyen transistör boyutu gecikmeyi oldukça arttıracığından bu telleri sürme işi kelime seçme sürücülerine bırakılmıştır.

Kelime seç sürücülerini içersinde arka arkaya bağlanmış iki tane evirici bulunmaktadır. Eviricilerin nMOS ve pMOS oranları en hızlı çalışacak şekilde seçilmiştir fakat boyutları mantıksal olarak işlem yapan eviricilerden daha büyüktür bu sayede uzun telleri küçük gecikmeyle sürebilmektedir. En iyi zamanlamayı yakalayabilmek amacıyla ilk evirici ikincisinden daha küçüktür. Bu sayede kod çözücünün çıkışında görülen sığa azalmakla beraber iki evirme işleminin biri daha hızlı gerçekleştirilmektedir.

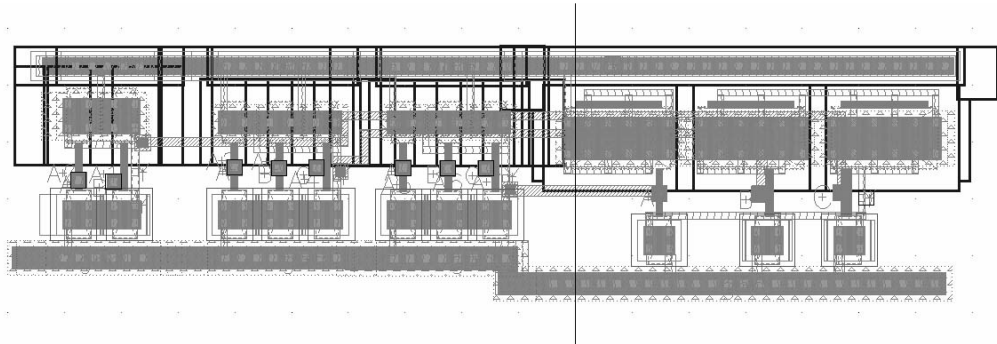
Kod çözücüler tüm satırların başında port sayısı kadar bulunmaları ve sayıca çok transistör ihtiva etmelerinden dolayı güç tüketiminde ciddi bir rol oynarlar. Şekil 2.7'de kod çözücünün deđişen adres genişliklerinde harcadığı devingen ve durađan güç tüketimi görülmektedir. Yazmaç öbeğinde kullanılan 256 satırı adresleyen bir kod çözücü ile yayınlama kuyruğunda kullanılan 32 satırı adresleyen bir kod çözücünün sızdırmaya bađlı güç tüketimi %60'dan fazla artmaktadır. Aynı durumda devingen güç tüketimi de %30 civarında artmaktadır.





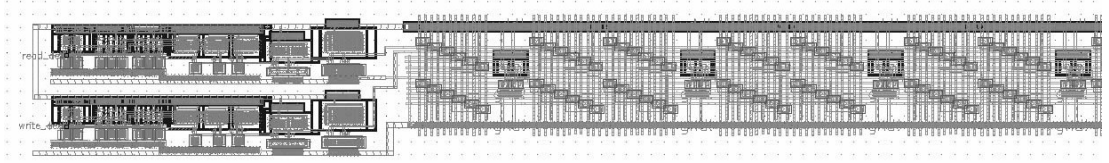
Şekil 2.7. Kod çözünün normalize güç tüketimi

Şekil 2.8’de 256 satırı adresleyebilen 8 girişli bir kod çözücünün fiziksel devre serimi görülmektedir. Görüldüğü üzere 8 girişi sağlayabilmek adına 4 tane 2 girişli VE DEĞİL kapısı yerine 2 tane 3 girişli VE DEĞİL ve 1 tane 2 girişli VE DEĞİL kapısı kullanılmıştır. Bu sayede kod çözücünün ikinci seviye mantık aşaması olan VEYA DEĞİL kapısı 3 girişli olarak tasarlanabilmiş ve tek kapıda gerçekleşmiştir. Bu ve buna benzer kararlar gerek başarımlar gerekse güç açısından alınırken, gerçekten fiziksel bir devre maskesi çizildiği için kod çözücünün SRAM bileşenlerine göre büyüklüğü de önemlidir.



Şekil 2.8. 256 satırlık kod çözücünün tek satırlık parçası

Kod çözücü devreler her satırın başında bulunduğu için, fiziksel devre boyutlarının yükseklik açısından bit hücrelerinden büyük olmamaları önemlidir. Şekil 2.9'de SRAM tablosundaki bit hücrelerine bağlanmış 2 tane 256 satırı adresleyebilen kod çözücü devre ve 64 bit kelime uzunluğundaki SRAM satırını sürebilen kelime seç sürücülerinden 2 tanesi gözükmektedir. 12 portlu bu SRAM hücrelerine 12 tane kod çözücü ve sürücüsü bağlanacak demektir. Bir bit hücresi genişliğine bu elemanlardan 2 tanesi sığabildiği için geri kalan 10 adet kod çözücü ve sürücüsü 2'şerli gruplar halinde sola doğru sıralanacaktır.

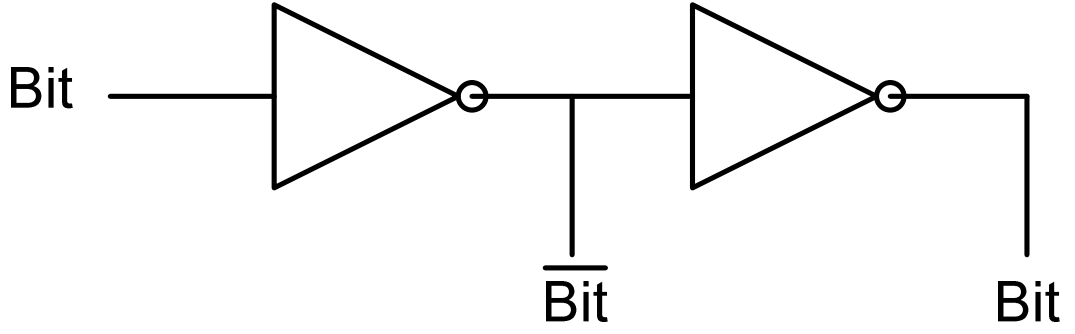


Şekil 2.9. SRAM tablosunun bir satırında bulunan kod çözücüler, sürücüler ve bit hücreleri

Kod çözücü sürücülerinin bit hücrelerinin kelime seç tellerinden ne kadar uzaklaşır ise telli sürmek için gereken enerji artacaktır. Bu sebepten dolayı çok portlu bit hücrelerinde tüm sürücülerin aynı enerjiyi tükettiği söylenemez fakat bu tezde kullanılan hesaplamalarda ortalama bir değer alınarak bu hata payı azaltılmıştır.

### 2.2.3. Yazma Sürücülerini

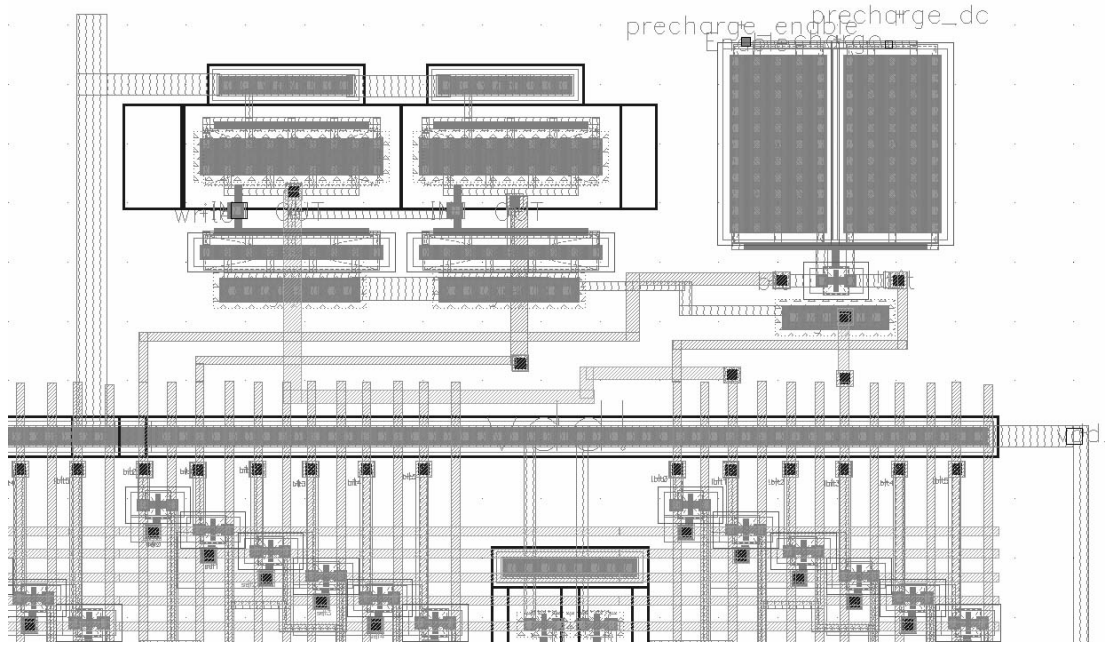
SRAM tablosuna veri yazma amaçlı yapılacak her türlü erişim için verilerin önceden ilgili yazma portunda hazır bulunması gerekmektedir. Bölüm 2.2.2'de anlatılan kelime seç sürücülerine benzer amaçla yazma tellerinin de sürülmesi gerekmektedir. SRAM tablosunda kelime uzunluğu yani sütun sayısından bağımsız olarak satır sayısı arttıkça yazma portlarına gönderilecek veri telleri uzamaktadır. Kelime seç sürücülerinden farklı olarak yazma sürücülerini SRAM tablosunun dikey telleri sürmektedir ve yazılmak istenen verinin tersini de sürmek durumundadır.



Şekil 2.10. Yazma sürücüleri

Yazma sürücüleri Şekil 2.10'da görüldüğü üzere iki tane arka arkaya bağlanmış eviricidir. SRAM tablosu satır olarak arttıkça bit hatlarının üzerine bağlanan portların nMOS transistörlerinin difüzyon sığıması biner [11]. Bunlarla birlikte, iki farklı tel sürmesi ve bu tellerin satır sayıları arttıkça uzamasından dolayı, tabloya yapılacak bir yazma işleminde devingen enerjinin en çok tüketildiği bölümü oluşturmaktadır.

Şekil 2.11'da yazma sürücüleri ve bölüm 2.2.4'te anlatılan ön doldurucularının bit hücrelerinin bir portuna yapılan bağlantıları gösterilmiştir. Arka arkaya bağlanan eviricilerinin her bir birinin dikey alanının fazla artmaması amacıyla, eviriciler çok parmaklı serim (multi-fingered layout) adı verilen bir teknikle çizilmiştir. Bu teknikte bir büyük nMOS ya da pMOS yerine daha küçük boyutlu transistörler birbirlerine paralel olarak bağlanmaktadır. Bu sayede geniş bit hücrelerinden uzaklaşmadan alandan azami derece yararlanılmıştır.



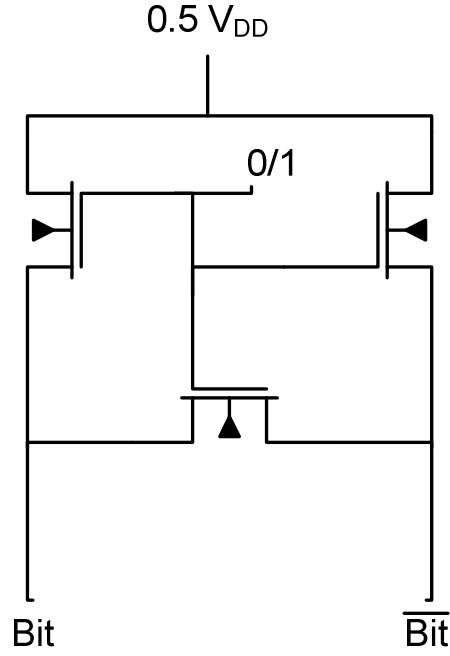
Şekil 2.11. Yazma sürücüleri ve ön doldurucuların bit hücrelerine yapılan bağlantıları

#### 2.2.4. Ön Doldurucular

SRAM tablosundan veri okuma amacıyla kod çözücünün aktive ettiği satırın kelime seç teli ve bağlı olan portları açıldıktan sonra bit hücrendeki eviriciler “Bit” ve “Bit DEĞİL” tellerine hücrenin verisini aktarırlar. Tablodaki satırlar arttıkça bu teller ciddi oranlarda uzarlar. Uzayan bu telleri sürmek, olabilecek en küçük boyutlarda tasarlanmış bit hücresi eviricilerine bırakılamayacak kadar kısa sürmesi gereken bir süreçtir. Bu amaçla ön doldurucu devreler kullanılmaktadır. Bu devreler okuma sinyali öncesinde “Bit” ve “Bit DEĞİL” tellerini  $V_{DD}/2$  gerilim seviyesine çekerler. Bu sayede okunacak veri ne olursa olsun dolup boşalması gereken sığa yarısına iner. Erişim zamanı kısaltıldığı gibi harcanan enerjide azalır.

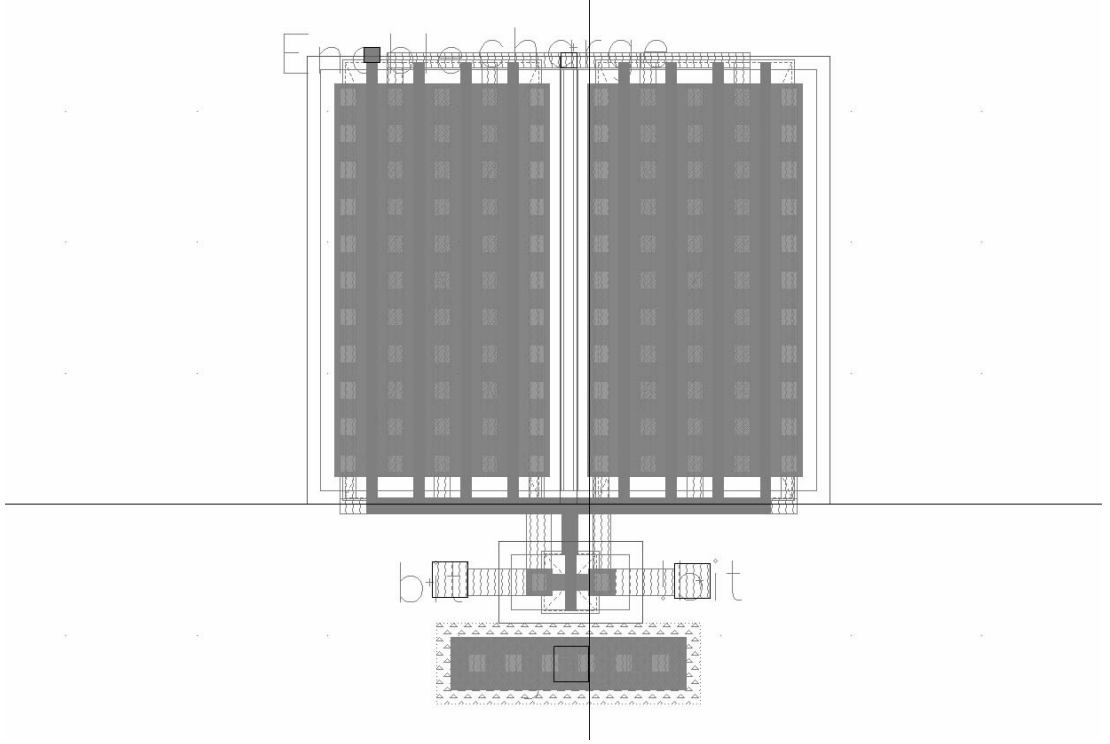
Şekil 2.12’de görülen ön doldurucunun devre şemasında iki adet giriş bulunmaktadır. Bunlardan biri nMOS’ların kapılarına bağlanan 0/1 açma kapama girişidir. Bu sayede okuma öncesinde ön doldurucular çalıştırılarak, telleri istenilen gerileme çekeler. Diğer durumlarda devre kapatılarak enerji tasarrufu sağlanır. Ön doldurucu devrenin diğer girişi ise gerilim girişidir. Örnekte  $V_{DD}/2$  giriş olarak verilmesine

rağmen doldurma işlemi  $V_{DD}$  'ye kadar da yapılabilir. Savak ve kaynak uçları "Bit" ve "Bit DEĞİL" hatlarına bağlanan transistör ise iki telin gerilim seviyesini aynı tutmak için mandal görevi görmektedir.



Şekil 2.12. Ön doldurucunun devre şeması

Şekil 2.11'de bir bit hücresinin portuna yapılan ön doldurucu bağlantıları görülebileceği gibi Şekil 2.13'de tek bir ön doldurucunun devre serimi gözükmektedir. Devre şemasından farklı olarak burada transistör boyutları açık bir şekilde gözükmektedir. İki teli de aynı seviyede tutan nMOS transistörün boyutu ile gerilimi tellere ileten transistörlerin boyutları arasında ciddi bir fark vardır.



Şekil 2.13. Ön doldurucu devresinin fiziksel serimi

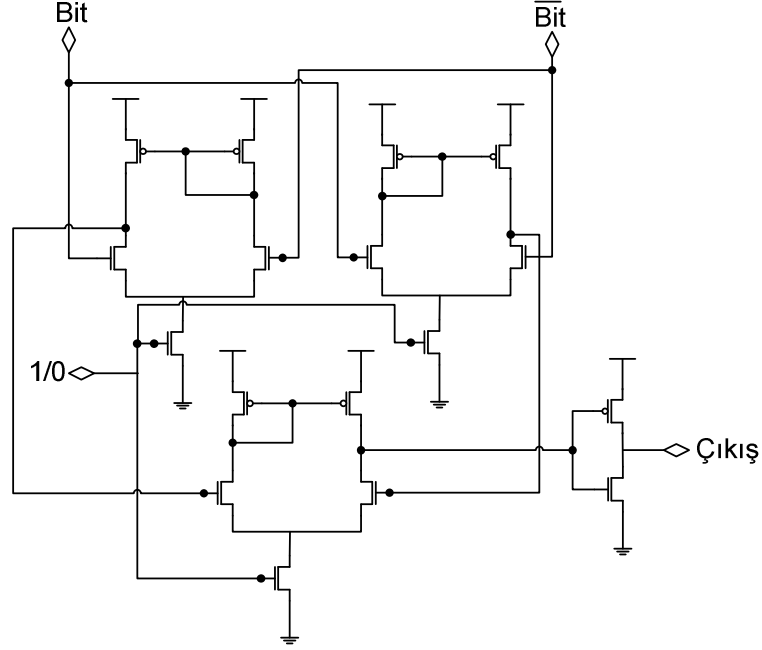
Tıpkı yazma sürücüleri gibi ön doldurucular da tüm satırların üzerinden geçen telleri sürdüğü için çok yüksek sığaları doldurup boşaltmaktadır. Bununla birlikte gerilimi tellere ileten transistörlerin genişlikleri çok artmaktadır. Artan genişlik birden çok portu bulunan bit hücrelerine ön doldurucularının yer planlamasını güçleştirmektedir. Alanı daha verimli kullanmak amacıyla, fiziksel devre seriminde küçük boyutlu paralel nMOS transistörler kullanılmıştır.

Ön doldurucular, diğer SRAM bileşenlerine kıyasla daha az güç tüketirler. Bunun nedeni, yazma veya kelime seçme sürücüleri gibi eviriciler içermediklerinden daha az transistör ihtiva ederler ve sadece okuma öncesi çalıştırılırlar.

### 2.2.5. Fark Algılayıcılar

Fark algılayıcılar (fark kuvvetlendiriciler) kazancı çok yüksek olan yükselticilerdir. İki farklı gerilim seviyesindeki çok ufak bir farkı (mV seviyesinde) hissederek, çıkışlarına  $V_{DD}$  veya toprak gerilim seviyesine çekerler. Fark algılayıcı devreler

SRAM tablosunun diğer tüm bileşenlerden farklı analog bir devredir. Okuma sırasında kullanılan bu devreler “Bit” ve “Bit DEĞİL” tellerinin arasındaki gerilim farkını çok hızlı bir şekilde algılayarak okunan verinin 1 veya 0 olduğunu normalden hızlı bir şekilde belirler. Satır sayısı ile artan bit tellerinin uzunluğu ve artan sığa yüzünden kullanılan fark algılayıcıları, satır sayısının çok az, dolayısıyla tellerin ve sığanın çok küçük olduğu durumlarda kullanılmayabilir.

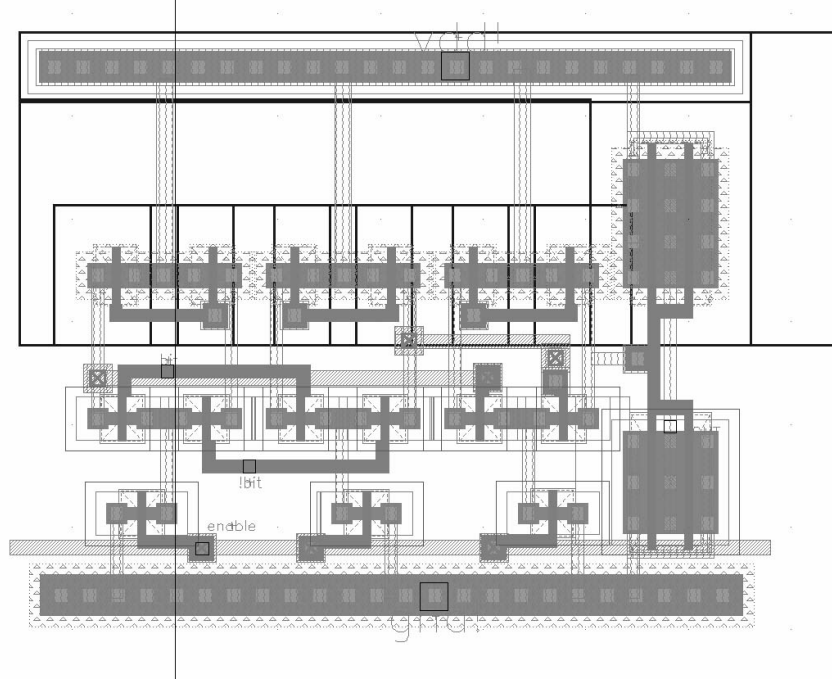


Şekil 2.14. İki aşamalı fark algılayıcısı devre şeması

Şekil 2.14’de iki aşamalı aktif yüklü fark algılayıcısı devre şeması gösterilmiştir. “Bit” ve “Bit DEĞİL” telleri ile “1/0” açma kapama düğümü devresinin girişleri, çıkış ise sonuç çıkışıdır. Fark algılayıcının ilk aşaması iki adet aktif yüklü akım aynasından, ikinci aşaması ise ilk aşamadan çıkan sonuçların giriş olarak verildiği bir tane aktif yüklü akım aynasından oluşmaktadır. Çıkış ise bir eviriciden alınmaktadır. Akım aynaları aktif yüklü olduklarından tüm devreden yüksek bir doğru akım geçmektedir. Bu nedenle SRAM tablosunda sadece okuma sırasında değil, devreye güç verildiği tüm durumlarda hemen hemen aynı miktarda güç harcarlar. Okuma işlemi dışındaki tüm durumlarda devre açma kapama girişinden 0 verilerek kapatılır.

Şekil 2.15’de görülen fark kuvvetlendiricisinin akım aynalarının transistor oranları mümkün olabilen en küçük boyutlarda tasarlanırken gecikme de göz önünde

bulundurulmuştur. İki aşamalı devre, sistemin geneline göre çok hızlı bir şekilde sonucu eviricinin kapısına iletmektedir. Bu noktada tüm fark algılayıcının gecikmesinin artmaması için eviricinin boyutları normalden büyüktür.



Şekil 2.15. Farklı algılayıcının fiziksel devre serimi

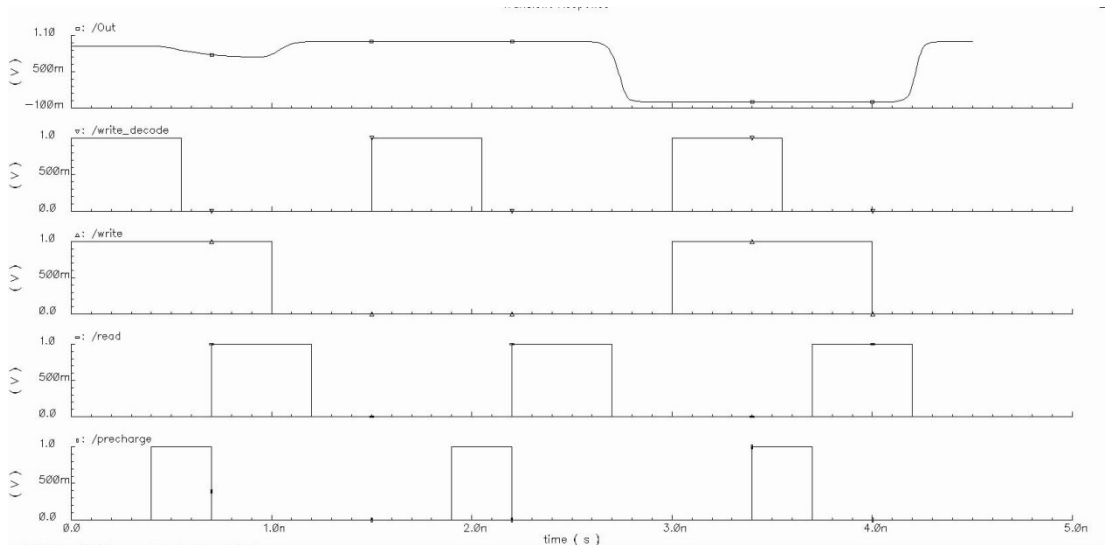


## 2.3. Başarım, Güç ve Alan Analizi

### 2.3.1. Başarım

Bölüm 2.2’de anlatılan SRAM tasarımında devrelerin başarım açısından çok farklı tasarlanabileceği aşikardır. Bu tez çalışmasında benzetim araçlarımıza ve üzerinde çalıştığımız işlemci çekirdeğine uygun olması açısından SRAM’den oluşan veri saklayan bileşenlerin tümüne 2GHz saat sıklığında okuma veya yazma yapılabilecek şekilde erişilmesi gerekmektedir. Bir başka deyişle, bir okuma veya yazmanın toplam süresi 500 piko saniyeyi geçmediği sürece devre üzerinde güç tüketimi ve alanı azaltacak her türlü değişiklik yapılabilir. Devre şeması düzeyinde bu tip değişiklikler tasarım aşamasında başarılı olsa da fiziksel devre serimi seviyesinde genellikle gecikme, alan ve güç tüketiminin en istenen seviyede olması için yeri geldiğinde deneme yanılma tasarımlara dahi başvurulmuştur.

Şekil 2.16’da 120 satırlık, 10 bitlik kelime uzunluğuna sahip bir SRAM tablosunun tek bir bit hücreesine, sırasıyla 1 yazılmış ardından bu veri okunmuş, 0 yazılmış, veri okunmuş ve son olarak tekrar 1 yazılıp bu veri okunmuştur. Toplamda 3 yazma, 3 okuma amaçlı erişim yapılmıştır. En üst satırda fark algılayıcısının zamana karşı gerilim grafiğinden bu erişimlerin sırası takip edilebilir.



Şekil 2.16. SRAM tablosuna erişim sırasındaki giriş ve çıkış işaretleri

SRAM tablosuna yazma amaçlı yapılacak her erişimden önce Şekil 2.16'nın 2. satırındaki yazma portunu kontrol eden kod çözücü ve 3. satırındaki yazma sürücülerine veri gönderen giriş sinyali 1V gerilim seviyesine çekilerek yazma işlemi hazırlanır. Benzer şekilde, okuma işlemi öncesinde Şekil 2.16'nın en alt satırındaki ön doldurucu açma kapama sinyali 1V gerilim seviyesine getirilerek açılır böylece bit hatları doldurulur hemen ardından 4. satırda görülen okuma portlarını kontrol eden kod çözücüye işaret verilerek kelime seç telleri aktive edilir. Görüldüğü üzere kod çözücüye verilen sinyallerden yaklaşık 500ps sonra fark algılayıcısının çıkışındaki işaret değişmektedir (başlangıçta, 1V gerilim seviyesinde başladığı için zaten geçiş olmamaktadır). Tabloya yapılan erişimlerin arasındaki boş zamanlarda fark kuvvetlendiricisinin işaret düşüşü 1ns civarlarında görülebilir.

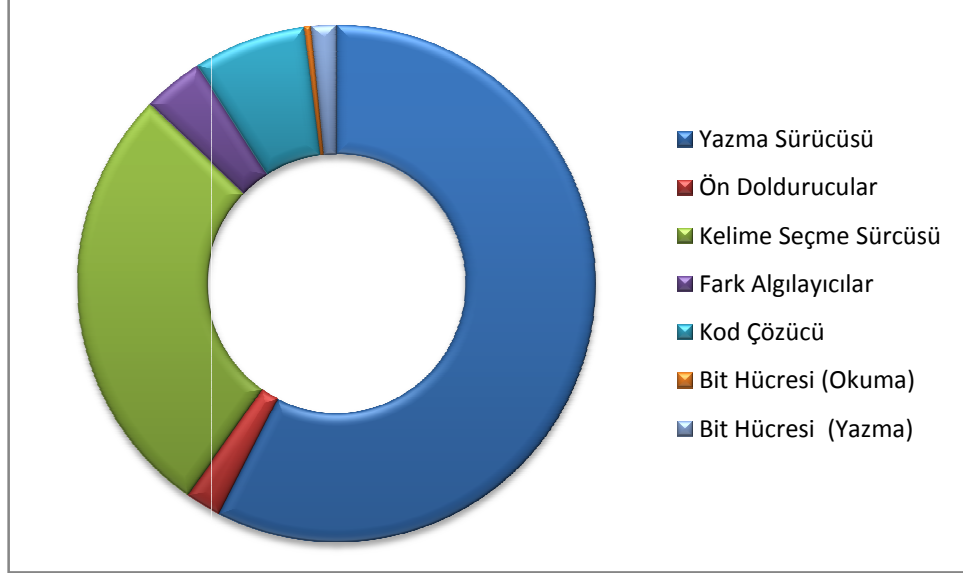
### 2.3.2. Güç Analizi

Modern mikroişlemcilerdeki veri saklayan SRAM tablolarının tamamen benzetimini yapmak çok büyük bir hesaplama gücü gerektirmektedir. Devre benzetimi yazılımları ne kadar alt seviye benzetime inerse o kadar yavaş çalışmaktadır. Bu sebeple bu tezin bir başka amacı da SRAM tabloların güç modellerini çıkararak hem uzun zaman alan fiziksel devre çizimlerini olabildiğince az yapmak ve bu çizimler yapılsa dahi daha kısa sürede sonuç vermesini sağlamaktır.

[12]'a göre SRAM tabloların 3 temel güç tüketim bileşeni bulunmaktadır. Bunlardan birincisi kod çözücüler, sürücüler, bit hücreleri, ön doldurucularda harcanan devingen enerji, ikincisi fark algılayıcılarının tükettiği akım yükseltme kaynaklı enerji ve son olarak tüm tablonun sızdırmaya bağlı durağan enerji tüketimidir. Bu çalışmada da enerji iki bileşene bölünerek fark algılayıcılarının durağan enerjisi durağan enerji hesaplarına, devingen harcadığı enerji de devingen enerji hesaplarına eklenmiştir.

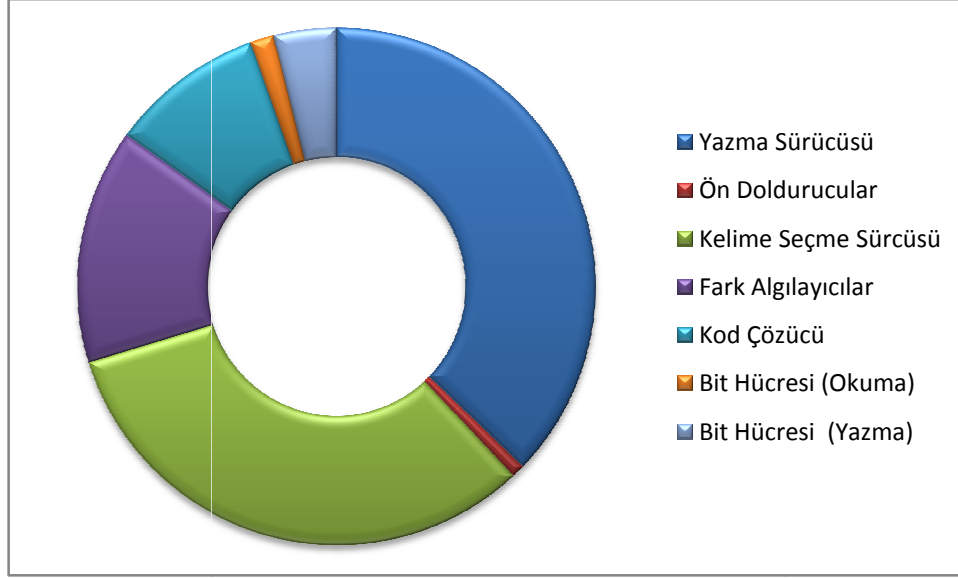
SRAM tablosuna erişimin iki amacı olan yazma ve okuma, güç hesaplarında temel alınmıştır. Yazma ve okuma sırasında kullanılan bileşenlerin devingen ve durağan güç tüketimleri süreçleri yükler altında ölçülerek ayrı ayrı hesaplanmıştır. Bu sayede bir birinden bağımsız bileşenlerin güç tüketimleri toplanarak okuma ve yazma enerjisi olarak iki farklı güç tüketim elemanı elde edilecektir. Elde edilen bu

değerler erişim sayısı ile çarpılarak tablonun toplam devingen güç tüketimi hesaplanacaktır. Benzer şekilde hesaplanacak olan durağan güç tüketimi de toplam saat vuruşu ile çarpılarak durağan güç tüketimi hesaplanacaktır.



Şekil 2.17. 256 satır 64 kolonluk 12 portlu SRAM tablosu elemanlarının güç tüketimi oranları

Şekil 2.17’de 256 satır 64 kolonluk 12 portlu SRAM tablosunun elemanlarının güç tüketim oranları görülmektedir. Buradaki oranlar bir okuma ve yazma sırasında kullanılan toplam enerjiyi göstermektedir. Kelime seçme, kod çözme işlemleri iki kere yapılırken diğer güç elemanları sadece ya yazmada ya da okuma kullanılmıştır. Bellekte bulunan yazma ve kelime seçme sürücülerini en fazla gücü tüketmektedir. Bunun nedeni tablonun satır ve sütun sayısının oldukça büyük olmasıdır. 256 satır ve 64 sütun verileri temel alınırken işlemcinin yazmaç öbeği örnek alınmıştır. İşlemcinin yayın kuyruğunda anlık verileri saklayan 32 satır ve sütunlu 8 portlu SRAM tablosu temel alındığında Şekil 2.18’deki gibi bir durum ortaya çıkar. Burada da yazma ve kelime seçme sürücülerini güç tüketiminde başı çekmektedir fakat diğer bileşenlere oranları ciddi oranda düşmüştür. Analog bir bileşen olan fark algılayıcılar aktif olmadığı durumda da aktif durumuna yakın bir enerji tüketmektedir. Görüldüğü üzere fark algılayıcıların enerji oranı devrede neden bir açma kapama girişi olduğunu açıklamaktadır.



Şekil 2.18. 32 Satır ve sütunluk 8 portlu SRAM elemanlarının güç tüketim oranları

Tez çalışmasında kullanılan modern mikroişlemci çekirdeğinin yazmaç öbeği 256 satır 64 sütundan oluşan bir SRAM belleğidir. Bu belleğin üzerinde çeşitli değişiklikler yaparak hem enerji tüketimi hem de başarımlar üzerinde iyileştirmeler yapılabilir. Bu değişiklikleri yaparken en iyi yapılandırma arandığından birçok deneme yapılmaktadır. Her deneme için SRAM tablosunun tekrar çizilmesi hem iş gücü olarak çok büyük bir yük hem de kullanılan benzetim araçlarının çalıştırılması uzun süreler alan çalışmalardır. Tez çalışması ve akademik çalışmalar sırasında birçok yapılandırma için enerji sayıları toplanmıştır. UMC 90nm teknolojisi ile çizilen devre serimlerinden hesaplanan enerji tüketim sayıları ile yazmaç öbeği ve yayın kuyruğunun enerji tüketimini matematiksel olarak modellemenin mümkün olacağı görülmüştür. SRAM tablosu satır veya sütun olarak genişlediğinde bileşenlerin üzerine binen direnç ve sığalar sabit olduğundan enerji tüketimi belirli katsayılarla artmaktadır. Yeterince sayı olarak ve satır sütun sayısını sınırlı tutarak bu katsayılar bulunmuş ve devingen ve durağan enerji tüketimleri modellenmiştir.

Bir SRAM tablosuna yapılacak bir kelimelik (bir satırdaki sütunların hepsi) okuma ve yazma erişiminin güç tüketimi denklem 2.1’de gösterilmiştir. Burada “x” sütun sayısı “y” ise satır sayısı değişkenleridir.

$$E(x, y)_{okuma} = \text{Kod Çözücü} + \text{Kelime Seçici} + \text{Ön doldurucu} + \text{Bit hücresi}_{okuma} + \text{Fark Algılayıcı} \quad (2.1a)$$

$$E(x, y)_{yazma} = \text{Kod Çözücü} + \text{Kelime Seçici} + \text{Yazma Sürücüsü} + \text{Bit hücresi}_{yazma} \quad (2.1b)$$

Bölüm 2.2’de incelenen bileşenlerin enerji tüketimlerinin sütun sayısına (x değişkenine) veya satır sayısına (y değişkenine) bağlı oldukları açıklanmıştır. Denklem 2.3’de denklem 2.2’deki terimler UMC 90nm teknolojisi kullanarak çizilen devre serimlerinden hesaplanan güç tüketimlerinden elde edilen katsayılar ile değiştirilmiştir.

$$E(x, y)_{Devingen} = g(y) + (x\alpha + \gamma) + x(y\beta + \rho) \quad (2.2)$$

$$g(y) = \begin{cases} 75.1, & 256 \leq y < 128 \\ 73.5, & 128 \leq y < 64 \\ 53.1, & 64 \leq y < 32 \\ 52.2, & 32 \leq y < 16 \\ 49, & 16 \leq y \end{cases} \quad (fJ) \quad (2.3)$$

Çizelge 2.1. Devingen güç hesaplamalarında kullanılan katsayılar

| Katsayılar   | Yazmaç Öbeği (12 port) - Okuma   | Yazmaç Öbeği (12 port) - Yazma | Yayın Kuyruğu (8 port) - Okuma | Yayın Kuyruğu (8 port) - Yazma |
|--|--|--------------------------------|--------------------------------|--------------------------------|
| $\alpha$ -Sütun başına artan kelime seçme enerjisi (fJ/sütun)                            | 3.25   | 3.25                           | 2.6                            | 2.6                            |
| $\beta$ -Satır başına artan ön doldurma ve yazma sürücüsü enerjisi (fJ/satır)            | $\begin{cases} 0.19, & 256 \leq y < 150 \\ 0.35, & 150 \leq y < 85 \\ 0.19, & 85 \leq y \end{cases}$ | 4.74                           | 0.065                          | 4.3                            |
| $\gamma$ -Kelime seçme enerjisi sabiti (fJ)  | 83   | 83                             | 88.7                           | 88.7                           |
| $\rho$ -Bit hücresi okuma yazma, fark algılayıcı ve yazma sürücüsü sabit enerjileri (fJ) | 88.4   | 91                             | 87.7                           | 91.8                           |

Denklem 2.4'de kod çözücünün satırlara bağlı fonksiyonu verilmiştir. Örneğin 150 satırlık bir kod çözücünün devingen enerji tüketimi 71.5 fJ iken 50 satırlık bir kod çözücünün devingen enerji tüketimi 53.1 fJ olmaktadır.

Çizelge 2.1'de denklem 2.3'deki katsayıların anlamları ve değerleri verilmiştir. Örneğin, 32 satır 20 sütuna sahip bir SRAM tablosuna bir kelime okuma işleminin devingen enerjisini hesaplamak için, x ve y sırasıyla 20 ve 32 seçilerek denklemde yerine konmalıdır. Bu durumda tablodan okunacak değerler,  $\alpha=2.6$ ,  $\beta = 0.065$ ,  $\gamma = 88.7$ ,  $\rho = 87.7$  olacaktır. Kod çözücünün enerjisi  $g(y)$ , denklem 2.4'den 57.7 fJ

alınır. Bir tabloya yapılan ve okuma ve yazma erişim enerjileri hesaplandıktan sonra okuma ve yazma sayısı ile çarpılarak toplamda harcanan enerji bulunur.

Durağan enerji ise yapılan işlemler boyunca sızdıran devre elemanlarının ne kadar süre açık kaldığına bağlı olarak hesaplanır. Bu durumda tablodaki bileşenlerin sayısını ve türünü bilmek gerekmektedir. Örneğin 12 portlu SRAM tablosunda her satırda 12 tane kod çözücü kullanılırken 8 portlu SRAM tablosunda 8 tane kullanılır.

Bu nedenle durağan enerji denkleminde, devingen enerji denkleminde bulunan x ve y değişkenine ek olarak okuma port sayısını gösteren “z” değişkeni ve yazma portlarının sayısını gösteren “k” değişkeni eklenecektir.

$$E(x, y, z, k)_{Durağan} = y * (z + k) * f(y) + x * (y * \sigma + k * \varphi + z * \omega) \quad (2.4)$$

Çizelge 2.2. Durağan enerji hesaplamalarında kullanılan sabitler

| Sabit ve Açıklaması                           | Sızdırma (fJ)  |
|---|--|
| $\sigma$ -Bit hücresi başına sızdırma         | 0.045  |
| $\varphi$ -Yazma sürücüsü başına sızdırma     | 0.6  |
| $\omega$ -Fark algılayıcısı başına sızdırma   | 0.68   |
| f(y)-Kod çözücüsü ve sürücülerinin sızdırması | $\begin{cases} 1.35, & 256 \leq y < 128 \\ 1.13, & 128 \leq y < 64 \\ 0.73, & 64 \leq y < 32 \\ 0.66, & 32 \leq y < 16 \\ 0.64, & 16 \leq y \end{cases}$ |

Denklem 2.5’te kullanılan sabitlerin değerleri ve açıklamaları Çizelge 2.2’te verilmiştir. Bu sabitler 80 ° C’ de UMC 90nm teknolojisi ile yapılan fiziksel devre serimlerinden elde edilen 1 saat vuruşu boyunca harcanan sızdırma enerjilerine dayanılarak hesaplanmıştır. Bir SRAM tablosundaki satır ve sütuna bağlı olarak değişen bit hücrelerinin toplam sızdırması ve bununla birlikte okuma ve yazma

portlarına göre deęişen sayıda bulunan yazma sürücüsü, fark algılayıcısı ve kod çözücülerinde aynı şekilde sızdırmaları hesaplanarak katsayılar hesaplanmıştır.

Böylelikle bir SRAM belleğin toplam harcadığı güç bulunmak istendiğinde denklem 2.3 ve 2.5 kullanılarak denklem 2.6 elde edilir.

$$E_{TOPLAM} = (Okuma\ sayısl * E_{Okuma}) + (Yazma\ sayısl * E_{Yazma}) + (Toplam\ Saat\ Vuruşu * E_{Duraęan}) \quad (2.5)$$

Denklem 2.6'daki okuma sayısı, yazma sayısı ve toplam saat vuruşu kavramları mikroişlemcinin üzerinde çalışan bir programın bu tablolara kaç kere yazdığını, okuduğunu ve toplamda kaç saat vuruşunda sonlandığını benzetimlikler vasıtasıyla belirleyerek bulunabilir. Bu denklemlerle elde edilecek sonuçların gerçek enerji tüketimi olarak kullanılmasından çok karşılaştırma verisi olarak kullanılması daha doğrudur.

### 2.3.3. Alan Analizi

Devrelerin maliyetini silikon üzerinde alanları belirler. Mikroişlemcilere yapılacak bir iyileştirme başarım ve güç tüketimi açısından çok iyi olabilir fakat bu iyileştirme silikondaki alanı 2 katına çıkarıyorsa bunun hiçbir yararı olmayabilir. Bu devrenin silikon üzerinde kapladığı alanın her yapılandırma için ne kadar deęiştğine hesaplamak gerekmektedir.

SRAM belleklerin tümünü çizerek alanlarını bulmak iş gücü açısından çok uzun bir uğraştır. Bu nedenle bazı teknolojilerinin hazır bellek sentezleyicileri vardır fakat modern mikroişlemcilerin özel tasarlanmış çok portlu bellekleri ticari olarak genel bir kullanım alanı bulmadığı için bu tip bellekleri sentezleyiciler vasıtasıyla çizmek mümkün deęildir. Bu nedenle belleklerin özel olarak tasarlanması gerekmektedir. Araştırma süreçlerinden onlarca hatta yüzlerce yapılandırma denendiği için her belleğin çizilmesi mümkün olmamaktadır. Bu nedenle alanların hesaplanması için SRAM belleğin içinde birden çok defa kullanılan bileşenlerinin alanlarını hesaplayarak kullanım sayısıyla çarpmak yaklaşımı başarılı sonuç vermektedir. Belleğin bileşenlerinin alanlarını hesaplariken dikkat edilmesi gereken çok önemli bir nokta vardır. Alanı hesaplanan bileşenin tek başına alanı çalışabilir bir bellek için bir şey ifade etmemektedir çünkü tasarım kuralları her tasarım bloğunun bir birine ne



kadar yaklaşabileceğini belirlemektedir. Bu kurallar ihlal edildiğinde bellek fabrikasyona uygun olmamaktadır. Böylece her bileşenin alanı bağlanması gereken yere tasarım kuralları dahilinde bağlandıktan sonra hesaplanmalıdır. Birçok bileşen arasında boşluklar veya bağlantıların geçtiği yollar olacağından alan bileşenlerin tek başlarına alanlarından çok daha büyük çıkmaktadır.

Enerji hesaplarında olduğu gibi satır ve sütun sayısı  $x$  ve  $y$ , okuma portu sayısı  $z$ , yazma portu sayısı ise  $k$  ile gösterildiğinde SRAM belleği toplam alanı denklem 2.7'deki gibi gösterilir.

$$A_{TOPLAM} = x * (k * A + y * B + z * (C + D)) + y * (k + z) * E \quad (2.6)$$

Çizelge 2.3. SRAM tablosunun bileşenlerinin fiziksel alanları

| SRAM Bileşeni                                     | Alan ( $\mu m^2$ ) |       |
|---|--------------------|-------|
| <b>A- Yazma Sürücüsü</b>                          |                    | 28.4  |
| <b>B- Bit Hücresi</b>                             | 12 Portlu          | 143.1 |
|   | 8 Portlu           | 88.61 |
| <b>C- Ön Doldurucu</b>                            |                    | 26.5  |
| <b>D- Fark Algılayıcı</b>                         |                    | 44.08 |
| <b>E- Kod Çözücü ve Kelime Seçme Sürücülerini</b> | 2 Bit              | 40.6  |
|   | 4 Bit              | 55.7  |
|   | 16 Bit             | 89.1  |
|   | 32 Bit             | 96.6  |
|   | 64 Bit             | 97.2  |
|   | 128 Bit            | 115.3 |
|   | 256 Bit            | 115.8 |

Denklem 2.7'ye göre hesaplanan alan ile tüm tablo bileşenlerinin doğru satır sütun ve port sayısına göre tasarım kurallarını ihlal etmeyen fiziksel olarak serimi  $\pm\%10$  hata payı içermektedir. Yapılandırma karşılaştırmaları için yeterli olan bu yaklaşım zamandan ve iş gücünden oldukça tasarruf edilmesini sağlamaktadır.

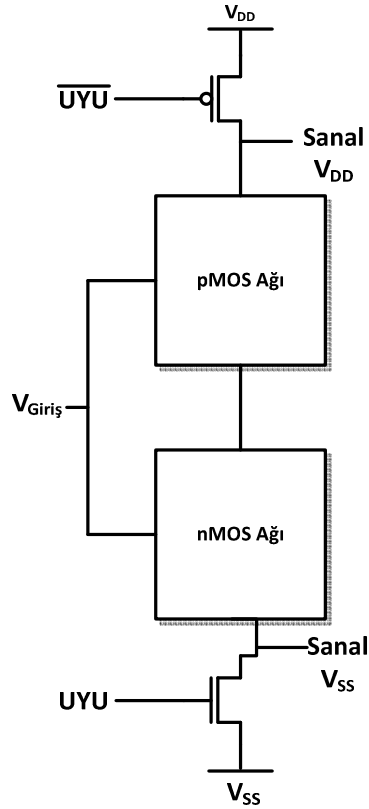
#### **2.4. Sızdırmaya Bağlı Durağan Güç Tüketimi Azaltma Yöntemleri**

Bölüm 1.1 ve bölüm 2.3.2.'de hem CMOS devrelerin hem de SRAM tabloların güç analizleri tüketimin tüm bileşenlerini kapsayacak şekilde analiz edildi. Güç tüketimini tetikleyen birden çok mekanizma ve bileşen olduğundan tüm sistemlere uygulanabilecek bir çözüm yoktur. Azaltılmak istenen güç tüketimi kaynağına göre bir çözüm uygulamak gerekmektedir. Bu yöntemleri uygularken silikon alanının artmamasını da sağlamak önemlidir. Bir ürünün ticari açıdan başarılı olmasının yolu iktisadi olmasından geçmektedir.

Sızdırma akımına bağlı güç tüketimini azaltmanın en kolay yolu devrenin gücünü kesmektir. Güç anahtarlama ya da güç ayrıştırılması (power gating) adı verilen bu yöntemde devreler mecazi anlamda uyutulur. Şekil 2.19'da devrelerin gücünü kesmek amacıyla yerleştirilen uyuma transistörleri gösterilmektedir. P bloğunu besleyen  $V_{DD}$  düğümü bir pMOS transistör ile ayrılarak sanal bir güç hattına sahip olmuştur benzer şekilde n bloğu da nMOS transistör ile  $V_{SS}$  gerilim seviyesinden ayrıştırılmıştır.  $V_{SS}$  0 (toprak) gerilim seviyesi olacağı gibi devrelerin başka bir devre bloğuna bağlı olabileceği hesaba katılarak bu şekilde bir gösterim seçilmiştir. Kullanılan uyutma transistörlerinin eşik değerleri p ve n bloklarının içinde kullanılanlardan yüksek olduğu için eşik altı sızırdırmalardan daha az etkilenmekte ve sızdırma akımlarında daha büyük tasarruflar sağlamaktadır [13]. Üretim teknolojisinin ve tasarım kütüphanesinin mümkün kıldığı farklı eşikli devre elemanları ile uygulanan yöntemlere, çok eşikli CMOS (Multi Threshold CMOS-MTCMOS) denmektedir. 0.13 mikron altı teknolojilerde üreticiler farklı eşik seviyesine sahip aygıtlara sahip kütüphaneleri piyasaya sürmüştür [14]. Buna rağmen çok düşük eşikli aygıtların sızdırma akımlarının ciddi oranlara gelebileceği göz önünde bulundurulmalıdır [15].

UYU sinyali 1 olduğu zaman devre uykudan uyanmakta, 0 olduğu zaman uyumaktadır. Buradaki oluşan sanal gerilim  $V_{DD}$  ve  $V_{SS}$  seviyeleri bölüm 1.1.3’de anlatılan durağan güç tüketimi bileşenlerini büyük oranda engelleyerek sızdırmaya bağlı güç tüketimini azaltmaktadır.

Bu yöntemde iki tane ödünleşim yapılması gereken konu vardır. Bunlarından birincisi kullanılan uyutma transistörlerinin getirdiği silikon alan yüküdür. Son zamanlarda uygulanan yöntemlerde silikonda kaplanan alanı azaltmak amacıyla sadece nMOS ya da pMOS transistörler ile tek bir sanal gerilim seviyesi yaratılarak güç kesimi yapılmaktadır[16],[17],[18]. Bu sayede fazladan kullanılan transistor sayısı yarıya düşerken güç kesimi de sağlanmaktadır. Alan ile ilgili bir başka tartışma konusu da uyku transistörlerinin lokal mi yoksa global olarak mı yerleştirilmesi gerektiğidir. Bu konuyla ilgili çeşitli yaklaşımlar [19] ve[17]’de incelenmiştir.

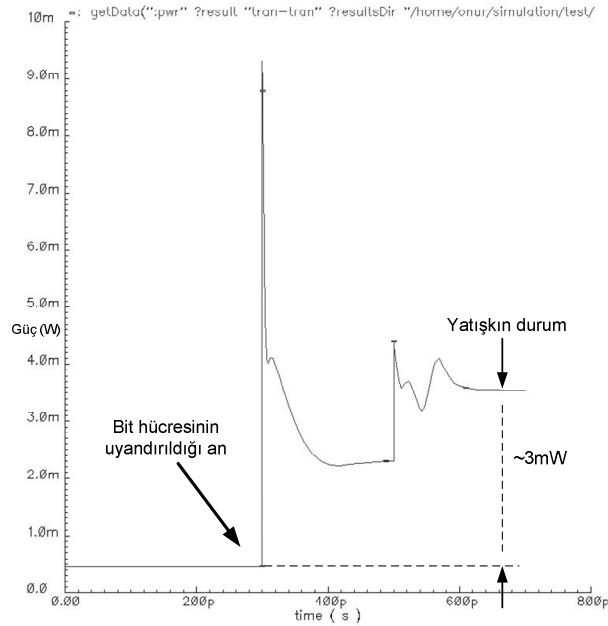


Şekil 2.19. Devrelerin uyutulması

İkinci ödünleşim konusu ise transistörlerin uyanma zamanıdır. Uyanma işareti geldikten sonra çok uzun sürede uyanan devrenin başarımı düşecektir. Çok kısa süre

de uyanan devrelerin ise sızdırmalara artabilir. Bu nedenle uyku transistörlerini paralel ve boyutları değişken bir şekilde bağlamak [16], devreleri değişik uyuma seviyelerine göre uyutmak [20],[21], uyuma transistörlerinin süreceği yükü azaltmak amacıyla zig-zag yöntemiyle bağlamak [22],[21] gibi yaklaşımlar kullanılmaktadır. Farklı uyuma seviyeleri UYU işaretinin yüksek eşikli uyutma transistörlerine farklı seviyelerde verilmesiyle sağlanır. UYU işareti eşik değerine ne kadar yakınsa devre o kadar çabuk uyanır fakat bu durumda sızdırma akımı tamamen azalmaz. Bu sebeple uzun süre kullanılmayacağından emin olunan bloklar derin uykuya alınırken, yakın zamanda erişileceği bilenen devreler hafif uykuya alınırlar. Zig-zag bağlantı ise bazı devre bloklarında gücü kesmek için nMOS, bazılarında pMOS kullanarak yapılmaktadır. Bu sayede bir blokta nMOS varsa diğerinde pMOS gücü kesmek kullanıldığı için bu ismi almıştır. Farklı uyuma seviyeleri ile birleştirilebilen bir yöntemdir.

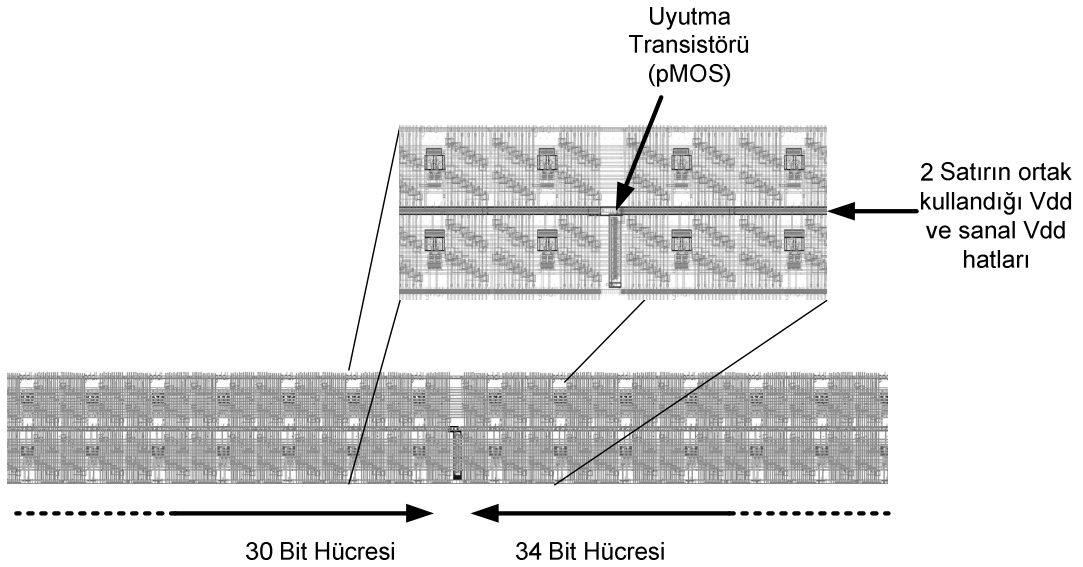
Bu tez çalışmasında SRAM tablosu olan yazmaç öbeğinin satır ve sütunlarına güç kesme yöntemi uygulanmıştır. Satırlar 2'şer, 2'şer, 64 bitlik sütunlar ise en anlamlı 30 biti veya tamamı kapatılabilecek şekilde tasarlanmıştır. 12 portlu bir bit hücrenin uyku durumunda ve aktif durumda harcadığı güç Şekil 2.20'da görülmektedir. Güç grafiği yatışkın duruma ulaştıktan sonra uykudaki bit hücrenin 3mW tasarruf ettiği görülmektedir. Bit hücrenin bir SRAM tablosunun sayıca en çok kullanılan elemanı olduğu göz önüne alındığında bu önemli bir tasarruftur.



Şekil 2.20. Bit hücrenin zamana karşı harcadığı güç grafiği

Şekil 2.20’da ki bir diğer önemli nokta ise bit hücresinin yatışkın duruma ulaşma süresidir. 2Ghz’lik bir saat sıklığında 500 ps içinde açılan bir hücre 1 saat vuruşunda açılıyor kabul edilmektedir. Bir saat vuruşunda bir uyutma transistörünün iki satırda bağlı olan 68 tane (bir satırda 34 hücre bulunmakta) bit hücrelerine güç vermesi gerekmektedir. Bu süre transistörün genişliğine bağlı olarak daha uzun veya kısa olarak ayarlanabilir.

Şekil 2.21’de uyutma transistörünün iki satırın enerjisini nasıl açıp kapadığının fiziksel serimi gözükmektedir. Üst üste farklı metal katmanlarıyla çizilmiş sanal ve gerçek  $V_{DD}$  hattı arasında anahtarlama yapan uyutma transistörü, çok parmaklı ve yatay olarak bir satırdaki bit hücresinin yanına yerleştirilmiştir. Transistör açıldığında 2 satırda bulunan toplam 68 bit hücresinin enerjisini açmaktadır.  $V_{DD}$  hattı tüm satır boyunca devam etmektedir fakat bit hücrelerinin beslendiği sanal  $V_{DD}$  hatları sağdan 34 hücre ve geri kalan 30 hücrenin birleştiği yerde ayrılmaktadır. Bu sayede geri kalan 30 hücre için de bir uyutma transistörü konarak bit hücresinin iki satırı 2 parçalı açılıp kapatılabilmektedir. Yani 2 satırdaki tüm bit hücreleri kapalı olabilir (tamamen kapalı), sağdan ilk 34 hücreyi açık olabilir (yarı açık) veya tüm hücreler açık olabilir (tamamen açık). Donanıma kazandırılan bu yetenek bölüm 4’te mimaride yapılacak iyileştirmeler ile birlikte güç tasarrufu sağlamak amacıyla kullanılmıştır.



Şekil 2.21. Bit hücreleri ve uyutma transistörlerinin fiziksel serimi

Satırların 2'şer yerine tek tek kapanması mimari açılarından daha mantıklı olmasına rağmen, güç hatlarını paylaşan devre serim tarzı alanı oldukça azaltmaktadır. Bununla birlikte eklenen uyutma transistörleri ve oluşan sanal gerilim hatlarının fiziksel olarak tüm SRAM tablosuna nasıl dağıtılması gerektiği gibi konular [23]'da bulunabileceği gibi, bit hücrelerinin gücünü kesmek için kullanılan farklı boyutlardaki uyutma transistörlerinin tartışmaları [13]'da bulunabilir.

### 3. MODERN MİKROİŞLEMCİLERİN YAPISI ve DAR DEĞERLER

Bilgisayarda çalışan yüksek seviyeli dillerde yazılan programlar derlenerek durağan bir makine koduna dönüştürülürler. Bu durağan kodun içinde arka arkaya sıralanmış buyruklar bulunmaktadır. Bu buyruklar donanıma göre derlenmiş olup çok basit ya da karmaşık işlemler olabilir. Mikroişlemciler bu buyrukları ne kadar hızlı çalıştırıp sonlandırırorsa başarımı o kadar artar.

Donanımı verimli kullanmak için boru hattı gibi yöntemler 1950'lerde popüler olsa da 1980'lerin sonlarında aynı anda birden çok buyruğu işleyip sonuçlandırabilen sırasız yürütüm yapabilen (out of order execution) çok yollu (superscalar) işlemciler yüksek oranda başarımlı sistemlere başarılı bir çözüm olmuştur [24]. Bu işlemciler Buyruk Düzeyinde Paralellik (DBP) özelliğini açığa çıkararak aynı anda birden çok buyruğu hesaplarken aralarındaki bağımlılıkları da çözerler. Böylece modern mikroişlemciler tek bir saat vuruşunda bir buyruk yerine birden çok buyruğu, birden çok boru hattı kullanarak çözerek yüksek başarımlı sistemler için genel geçer bir çözüm olmuştur.

Şekil 3.1'de sırasız yürütüm yapan genel bir işlemcinin mimarisi görülmektedir. Bu işlemci de temel aşamaları buyruğun yakalanması, çözülmesi, dağıtılması (ve/veya yayınlanması), işlenmesi (sırasız olarak yürütülmesi), sıraya sokularak tamamlanması ve son olarak işlemciyi terk etmesi olarak 6 temel aşamaya ayrılabilir. Her aşamanın sonucunda elde edilen veriler tamponlara, kuyruklara ve belleklere yazılarak bir sonraki aşamaya aktarılmakta ya da ilgili birimlerde tüketilmektedir.





sonraki doğru adresi belli olmadığı takdirde işlemci durmak zorunda kalabilir veya gideceği yönü tahmin ederek buyruk çekmeye devam edebilir. Eğer tahmin yanlış ise, yanlış yönde çekilen tüm buyruklar işlemciden atılır. Bu sırada saat vuruşu başına işlenen buyruk sayısı düşerek, başarıyı kötü yönde etkiler.

Buyruk çekme düzensizliklerini gidermek, dallanmaya bağlı ve ön belleğin kısıtlı alanından kaynaklanan buyrukların uzun erişim sürelerini kısaltmak için buyruk tamponu (Şekil 3.1'de gösterilmektedir) bir kısım yakalanmış buyruğu saklamaktadır. Bu sayede dallanma tahmini, ön bellek hatası gibi durumlarda işlemcinin durmadan çalışması sağlanmaktadır. Bununla birlikte işlem birimlerinden yapılan işlemler sonucu çıkan dallanma sonuçları, hedef dallanma tamponuna yazılarak, buyruk tamponunun güncellenmesi sağlanır.

### **3.2. Buyruk Çözme, Yeniden Adlandırma, Yayınlama ve Yürütme**

Buyruk çözme, yakalanan buyrukların tek tek tanımlanması, buyruk tiplerinin belirlenmesi ve diğer çekilen fakat henüz yayınlanmayan buyruklarla olan bağılıklarının belirlenmesidir. RISC ( Reduced Instruction Set Computer) tipi mimarilerde durağan makine kodunda belirtilen buyruk tanımlamaları çok kesindir. Buyrukların uzunlukları belli olduğu gibi, birkaç farklı tipe ayrıldıklarından başını ve sonunu belirlemek, işlem kodunu ve anlık değerleri ayrıştırmak kolaydır. CISC (Complex Instruction Set Computer) mimarilerde ise buyruk uzunlukları farklı olduğundan, buyrukların nerede başladıkları ve bittiklerini belirleyerek diğer buyruklar ile karışmaması sağlanmalıdır. Intel Pentium PRO işlemcisinde buyruk belleğine erişme ve buyruk çözme birkaç boru hattı aşamasına ayrılmış ve toplam 5 saat vuruşunda tamamlanmaktadır [25].

Bir buyruğun çözülmesi bittiği zaman yapılacak işlem, kullanılacak değerlerin nereden okunacağı, nereye yazılacağı ve sonucun nerede saklanacağı belli olmaktadır. Bu durumda buyruğun işlenmesi için gerekli kaynaklar adreslerinden okunmalıdır. Buyruk düzeyinde paralelliği sağlamak amacıyla bu buyruklar arasında hiçbir bağımlılık olmaması gerekir.

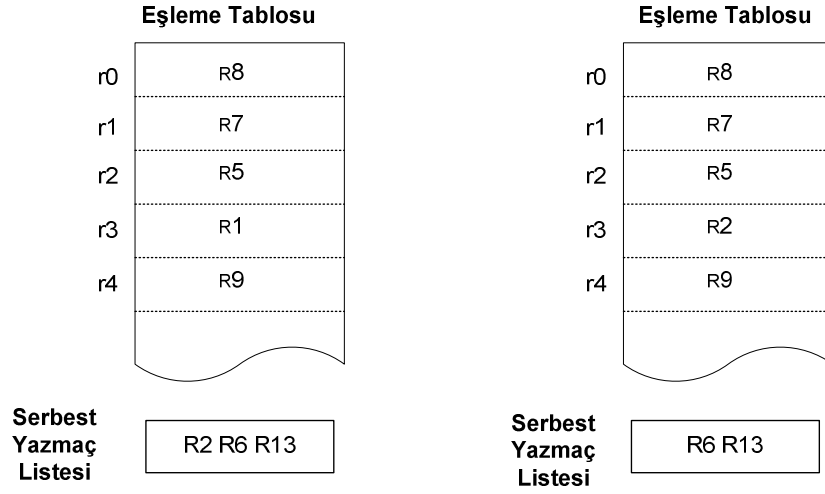
Bir buyruk kendinden bir sonraki buyruğun sonucunu kullanacak ise bu bağımlılığa YSO (Yazdıktan sonra okuma) adı verilir. İşlemci içersinde yürütüm sırası dağıldığı

için sonradan gelen buyruk kaynağa önce ulaşır, kaynaktan eski veriyi okuyabilir. Bu bağımlılığın giderilmesi ancak bir önceki buyruğun işini bitirip sonucunu yazması beklenerek çözülebilir. Bu gerçekten bir bağımlılıktır fakat işlemci içerisinde gerçekten bağımlılık olmayıp yeniden adlandırma tekniği kullanarak çözülebilecek yapay bağımlılıklar vardır. Bunlar YSY (Yazdıktan sonra yazma) ve OSY (Okuduktan sonra yazma) bağımlılıklarıdır. YSY aynı anda işlenen iki buyruğun aynı kaynağa veri yazmasıdır. Bu durumda yazma işlemi doğru sırada yapılmalıdır. Makine kodundan sonradan gelen buyruk o kaynağa sonradan yazmalıdır. OSY bağımlılığı ise bir buyruğun değerini yazmadan önce, o yazmacın eski değerini okuması gereken tüm buyrukların okumasından sonra yazması gerekliliğidir.

Yeniden adlandırma yöntemi var olan yazmaçların işlemci içerisinde başka yazmaçlarla eşlendirilmesi işlemidir. İşlemciye makine kodu yazılırken dışarıdan kullanılabilen mimari yazmaçları vardır fakat fiziksel olarak işlemci içerisinde daha çok yazmaç vardır. Bu şekilde bir mimari yazmaç birden çok fiziksel yazmaca eşlenebilir. Örneğin, MIPS R10000 işlemcisinde 32 mimari yazmaç varken 64 tane fiziksel yazmaç bulunurken [24], Alpha 21264 işlemcisinde tam sayı için kullanılan 80 tane fiziksel yazmacın 40'ı yeniden adlandırmaya ayrılmıştır [25]. Şekil 3.2'de küçük harfle başlayan  $r_0, r_1$  gibi yazmaçlar mimari, büyük harfle başlayan  $R_0, R_1$  gibi yazmaçlar fiziksel yazmaçları göstermektedir. *Add r3, r3, 4* buyruğu, 3 numaralı adresteki değere 4 tamsayısını ekleyerek tekrar 3 numaralı yazmacı yazma işlemi yapmaktadır. Şekil 3.2'ün önce kısmında  $r_3$  yazmacı  $R_1$  fiziksel yazmacına eşlenmiştir. Serbest yazmaç listesinde ise ilk boş yazmaç  $R_2$  fiziksel yazmacıdır. Yeniden adlandırma sonrasında sonucun yazılacağı  $r_3$  yazmacı  $R_2$  fiziksel yazmacına eşlenmiştir böylece bu buyruğun sonucunu okuyacak diğer kaynaklar tabloya baktığında  $R_2$  fiziksel yazmacına gitmesi gerektiğini bilmektedir.  $R_2$  serbest yazmaç listesinden kaldırılmıştır. Kullanılan fiziksel yazmaçlar bu buyruklar işini bitirdiğinde tekrar serbest yazmaç listesine konulacaktır. Böylece sonradan gelen buyruklar aynı yazmaçları kullanabilmektedir.

Önce: add r3,r3,4

Sonra: add R2,R1,4



Şekil 3.2. Yeniden adlandırma örneği

Fiziksel bir yazmacın tekrar serbest yazmaç listesine konulabilmesi için yazmaç ile bir sayaç tutulabilir. Fiziksel yazmacın, mimari yazmacıyla eşleştirildiği her durumda sayacı bir artırmak, ilgili buyruk yayınlandığı zaman bir düşürmek şeklinde bir yöntem ile durumu takip etmek, kullanılabilir basit bir yöntemdir.

Buyrukların birbirlerine bağımlılıkları giderildikten sonra kullanması gereken kaynaklar belli olmuştur fakat işlemci içinde yürütüm sırasız olduğu için bir buyruk için gerekli olan kaynaklar henüz hazır olmayabilir. Bu nedenle buyruklar bir kuyruğa alınır. Bu kuyruğa yayın kuyruğu (issue queue), rezervasyon durağı (reservation station) veya dağıtım tamponu (dispatch buffer) isimleri verilmektedir. Bir buyruğun yayın kuyruğundan çıkıp yürütmeye geçmesi için farklı yöntemler kullanılmaktadır. Buyruğun yayın kuyruğundan çıkması için yürütme birimlerinde yer olması, yazmaç öbeğinin ilgili portlarının boş olması gibi durumlarda kontrol edilmektedir. Her şey hazır olduktan sonra buyruk ilgili işlem birimine gider. Bu işleme “dağıtma” ya da “yayınlama” ismi verilir.

Tüm buyrukların aynı sürede tamamlanmamasından dolayı, yayın kuyrukları işlem tiplerine göre ayrılabilir. Böylece uzun süren buyruklar yüzünden kısa süren buyruklar beklemez. Oluşacak bu küçük yayın kuyruklarına dağıtım, merkezi bir tampondan gerçekleşir. Böyle bir mimaride merkezi tampona, “dağıtım tamponu”, buyrukların yürütme birimlerine göre yayın kuyruklarına dağıtılmasına “dağıtım”

aşaması denirken, yayın kuyruklarından yürütme aşamasına geçilmesinde “yayınlama” denmektedir. Bu durumda “dağıtım” aşaması ve “yayın” aşamasının farklı aşamalar olduğuna dikkat edilmelidir.

### **3.3. Buyrukların Tamamlanması ve Çıkması**

Buyruklar yürütme birimlerinden çıktıklarında, sırasız işlendiği için rastgele zamanlarda sırasız bir şekilde çıkarlar. İşlemcinin mimari durumu ise sıra ile güncellenmelidir. Bu amaçla “yeniden sıralama tamponu (reorder buffer)” kullanılmaktadır. Yürütme biriminden çıkan her buyruk burada tekrar sıraya sokulur. Bir buyruk eğer bellek işlemi yapmayacaksa, yeniden sıralama kuyruğundan çıktığı anda işlemcinin mimari durumunu günceller ve tamamlanmış (complete) sayılır. Eğer bellek işlemi yapılacaksa bellek için saklama sırasını bekler ve daha sonra çıkar (retire). Bellek işlemi olmayan buyruklar için tamamlanma ve çıkma aynı anda olmaktadır. Bu terminoloji işlemciden işlemciye farklılıklar gözükebilir.

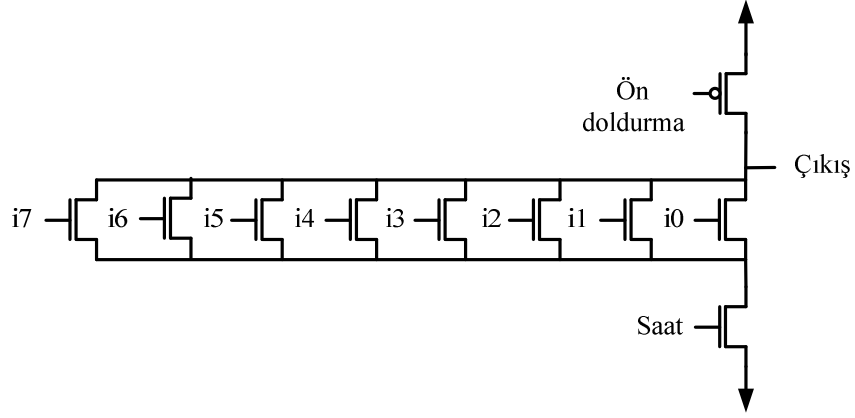
Sırasız yürütüm yapan işlemciler aynı anda birden çok buyruğu işlediği için yazmaç öbeği, yayın kuyruğu gibi veri saklayan bileşenlerin birden çok yazma portu ve okuma portu bulunmaktadır. Örneğin, aynı anda 4 buyruğun işlendiği çok yollu işlemcinin buyruk yapısına göre veri saklayan bileşenlerinin de okuma ve yazma portları belirlenmelidir. Buyruk yapısında 3 işleç olan bir işlemcide 1 yazma 2 okuma gerçekleştirildiği var sayılırsa, 4 buyruğun kaynaklarının aynı anda okunması için 8 okuma, yazılması için 4 yazma portuna ihtiyaç duyulmaktadır.

### **3.4. Dar Değerler**

Çağdaş mikroişlemcilerde kullanılan 64 bitlik kelime uzunlukları bazı programların adreslemeleri için gerekli olmaktadır fakat tam sayı işlemi gerektiren programlar 64 bit uzunluğundaki verilere çok nadiren ihtiyaç duyarlar [26],[27], [28]. İkilik düzende tutulan bu değerlerin üst kısımları gereksiz olacağından daha az bit ile gösterilebilirler. Bu değerlere dar değer adı verilmiştir. Örnek olarak, Şekil 3.3’de gösterilen tümü 0’dan oluşan değer sadece işaret bitiyle gösterilebilir. Önemli olan tekrar eden üst kısımları değerden atmaktır.

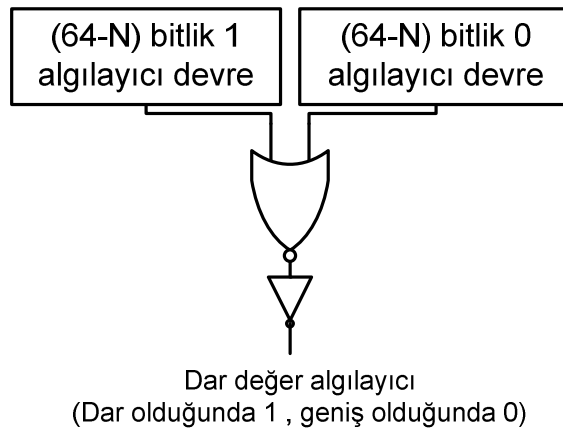


işareti bulunacaktır. Değerlerin herhangi bir tanesi mantıksal olarak 1 seviyesinde ise bağlı olduğu nMOS transistörü açarak çıkış düğümünü 0 mantıksal seviyesine indirecektir. Böylece değer dar olmadığı anlaşılacaktır.



Şekil 3.4. 8 bitlik 0 algılayıcı

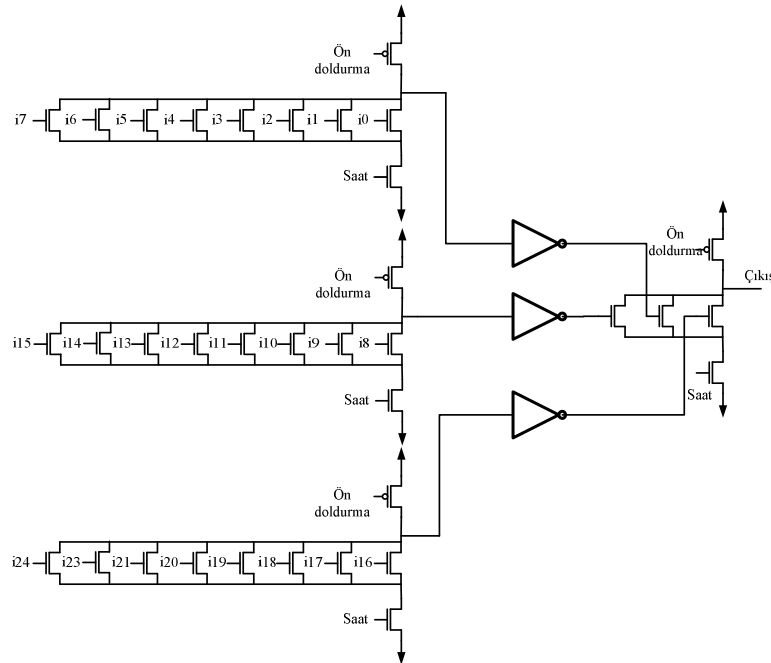
Tekrar eden 1 değerlerini anlamak içinse 0 algılayıcının girişlerini önce tersledikten sonra 0 algılayıcıya giriş olarak vermek gerekmektedir. Bu eviriciler olabildiğince küçük ve hızlı tasarlanmaktadır. Şekil 3.5’de 1 ve 0 algılayıcılarının beraber kullanımı gösterilmiştir. 1 ve 0 algılayan iki devre VEYA kapısından geçirilerek girişler dar değerse 1, geniş ise 0 işaretini çıkışa vermektedir.



Şekil 3.5. Dar değer algılayıcı

Yazmaç öbeğine yazılan anlık değerlerin %75'i sadece 8 bit kullanarak gösterilebilir [31]. Bu durumda 56 bitlik dar değer algılayıcı devreler kullanmak gerekmektedir. Şekil 3.5'te gösterilen 8 bitlik devrede bulunan paralel olarak bağlanmış 8 nMOS yerine 56 tane bağlanarak bu devre yapılabilir fakat çıkış düğümündeki sığa oldukça artmaktadır. Çıkış düğümündeki bu sığa enerji tüketiminde arttırdığı gibi, ön doldurma ve düşme zamanlarını da arttırarak devrenin toplam gecikmesini ciddi miktarlarda artırmaktadır.

Paralel bağlanan nMOS'ların artan gecikme ve enerji tüketimini azaltmak için iki aşamalı 0 algılayıcı devre önerilmektedir [31]. Şekil 3.6'da örnek olarak 24 bitlik dar değer algılayıcı gösterilmiştir. Devrenin ilk basamağında 8 bitlik dar değer algılayıcılar paralel bağlanarak, devrenin ön doldurma ve düşme zamanlarını artıran sığa farklı devre parçalarına bölünmüş, bu sayede devrenin çalışması hızlandırılmıştır. İlk aşamadan çıkan sonuçlar ise küçük eviriciler ile terslenerek aşaması sayısı kadar giriş içeren bir başka paralel nMOS'a bağlanmıştır. Paralel bağlanan nMOS'lar mantıksal olarak VEYA işlemi yaptığından, ilk aşamadan gelen değerlerden herhangi biri geniş işareti verdiyse çıkışı 0 yapmakta, aksi takdirde ön doldurulmuş çıkışı  $V_{DD}$  seviyesinde korumaktadır.



Şekil 3.6. İki aşamalı 24 bitlik 0 algılayıcı devre

Çizelge 3.1’te tek ve iki aşamalı 0 algılayıcının gecikme ve enerji açısından karşılaştırmaları gösterilmiştir. İki aşamalı algılayıcı çok girişli devreleri hızlandırmak için tasarlandığından, büyük dar değer algılama işlemlerinde fark net bir şekilde gözükmemektedir. 60 bit dar değer algılama işlemi yapılacağı zaman tek aşamada 215 ps süren düşme zamanı, iki aşamalı devrede %66 azalarak 74 ps sürmektedir. Benzer şekilde ön doldurma ve harcadıkları enerjilerde sırasıyla %91 ve %19 düşmüştür.

Çizelge 3.1. Tek aşamalı ve iki aşamalı 0 algılayıcı devrelerin enerji tüketimi ve gecikme zamanları

| Bit | Tek Aşamalı 0 Algılayıcı |                  |             | İki Aşamalı 0 Algılayıcı |                  |             |
|-----|--------------------------|------------------|-------------|--------------------------|------------------|-------------|
|     | Düşme zamanı (ps)        | Ön Doldurma (ps) | Enerji (fJ) | Düşme zamanı (ps)        | Ön Doldurma (ps) | Enerji (fJ) |
| 8   | 36.427                   | 18.683           | 44.900      | ---                      | ---              | ---         |
| 16  | 65.471                   | 30.990           | 69.713      | 64.496                   | 3.96             | 127.929     |
| 32  | 117.340                  | 74.720           | 118.477     | 67.09                    | 6.568            | 134.081     |
| 48  | 174.086                  | 77.592           | 162.450     | 72.09                    | 6.83             | 135.301     |
| 56  | 200.639                  | 90.189           | 181.092     | 74.248                   | 7.243            | 137.988     |
| 60  | 214.281                  | 96.564           | 190.516     | 74.337                   | 8.905            | 139.904     |
| 62  | 215.557                  | 97.113           | 195.572     | 74.4                     | 9.04             | 140.034     |



## **4. DAR DEĞERLERİ KULLANARAK VERİ SAKLAYAN BİLEŞENLERDEKİ ENERJİ TÜKETİMİNİN AZALTILMASI**

### **4.1. Giriş**

Günümüzde üretilen mikroişlemcilerin %95'i gömülü sistemlerde kullanılmaktadır [32]. Sınırlı pil ömürleri nedeniyle enerji tüketimi her zamankinden önemli hale gelmektedir. CMOS teknolojisinde sızdırma akımının önemi ve küçülen kapı boyutlarına göre artışı bölüm 1.1.3'te ele alınmıştır. Malzemedeki kaynaklanan ve istenmeyen bu akımın veri saklayan SRAM tablolarındaki analizi bölüm 2.3.2'de yapılırken, önleme yöntemleri bölüm 2.4'te açıklandı. Devre düzeyinde bu sorunu bölüm 3'te anlatılan mimari yöntemleriyle birleştirilerek durağan ve devingen gücü azaltmaya yönelik bir çözüm ise bu bölümde sunulacaktır.

Çağdaş mikroişlemcilere bakıldığında harcanan enerjinin bir noktadan ziyade, bileşenler arasında paylaşıldığı görülmektedir [33]. Yazmaç öbeğinin tükettiği enerji toplamda %25'e kadar çıkabilir[27],[34], [35]. Bununla birlikte 90nm üretim teknolojisi ile üretilmiş bir yongada enerjinin %35'inde yakını sızdırma akımları yüzünden durağan olarak tüketilmektedir [17].

Bu tez çalışmasında güç ayrıştırılması (power gating) yöntemi yazmaç öbeğine uygulanacaktır. Yazmaç öbeğinin kullanmayan satırları ve dar değer bulunan satırlarının üst kısımları dinamik olarak açılıp kapatılacaktır. Bu sırada işlemcinin başarımının düşmemesi ve silikonda kaplanan alanın artmaması esastır.

### **4.2. İlgili Çalışmalar**

Dar değerler üzerinde çok araştırma yapılmış ve önemli kavranmış bir konudur. Araştırmaların genel amacı devingen enerjiden tasarruf olsa da, dar değerler için yapılan modifikasyonlarda küçülen veri saklayan bileşenler durağan enerjiden de tasarruf sağlamaktadır [36]. Yazmaç öbeğini parçalara ayırarak dar değerleri ayrı bir yere yazmakta devingen enerjiden kazanç sağlamaktadır [30],[37],[38]. [39]'da ise yazmaç öbeğinin birçok portun yol açtığı alan genişlemesinin önüne geçmek

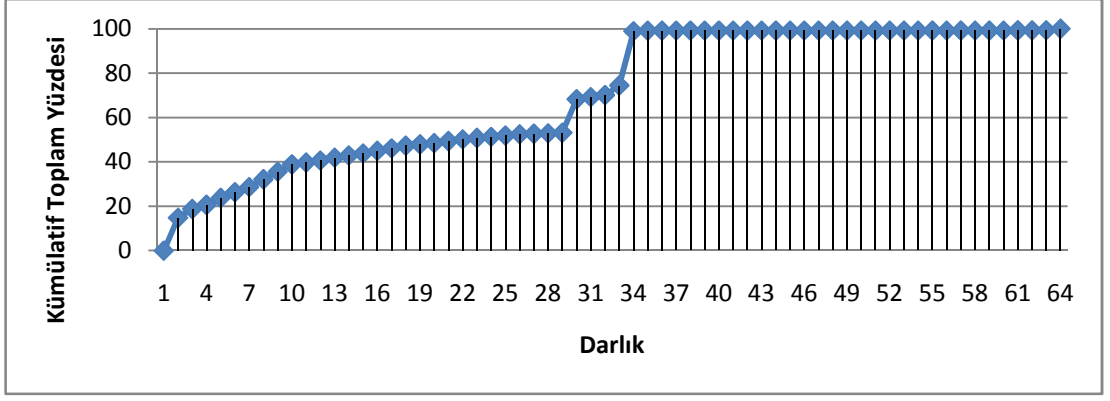
amacıyla, dar değerlerden yararlanarak, verileri dar ve geniş olarak ayırdıktan sonra farklı yerlerden yazıp okuyarak alan ve enerjiden tasarruf sağlanmıştır. [40]'da ise yazmaçların kullanım ömürlerinin kısa olmasından yararlanarak dar değerleri yeniden adlandırma tablosunda adres yerine yazarak, büyük olan yazmaç öbeğine erişimi azaltmaktadır.

Dar değerlerin ortaya çıkardığı diğer yararlar ise birkaç tane dar değeri aynı anda bir işlem biriminde çalıştırmak [26] olabileceği gibi dar değerleri önceden tahmin ederek veri yolunu daha verimli kullanarak başarıyı artırma amaçlı kullanılmıştır [41].

Bununla birlikte güç ayrıştırması yöntemi ve uygulamaları ise pil ömrü gerektiren gömülü sistemler ön bellekleri için [21]'de uygulanmıştır. [13]'da ise SRAM hücrelerinde sızıntı akımını gidermek amaçlı çok eşikli CMOS ile yapılan güç ayrıştırma yönteminden yararlanılmıştır. Bölüm 2.4'de sızıntıyı azaltmak için kullanılan güç ayrıştırması yöntemiyle ilgili daha çok çalışma bulunabilir.

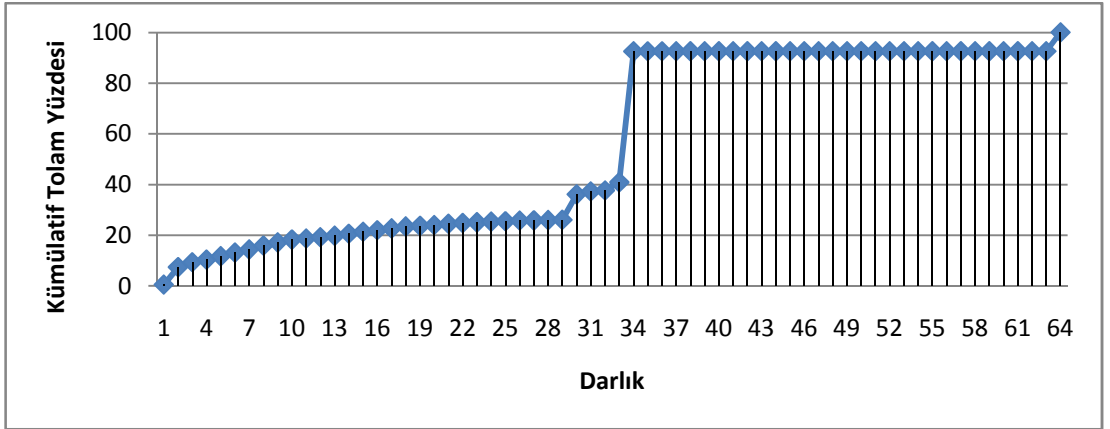
### **4.3. Yazmaç Öbeğindeki Dar Değerler ve Yazmaçların Kullanım Ömürleri**

Dar değerlerden fayda sağlamak için yazmaç öbeğindeki dar değer yoğunluğunun yeterli miktarda olması gerekmektedir. Bunu test etmek amacıyla kar amacı gütmeyen SPEC (Standart Performance Evaluation Corporation) şirketinin denektaşı programları kullanılmıştır. Bu programlar başarımlı ölçümü açısından genel kabul görmüştür [42]. Şekil 4.1'de Spec2000 programlarının çalıştırılması sırasında yazmaç öbeğine yazılan değerlerin yaklaşık %98'inin 34 veya daha fazla bitle gösterilebileceği görülmektedir.



Şekil 4.1. Spec 2000 programlarının çalıştırılması sırasında yazmaç öbeğine yazılan değerlerin darlık yüzdesi

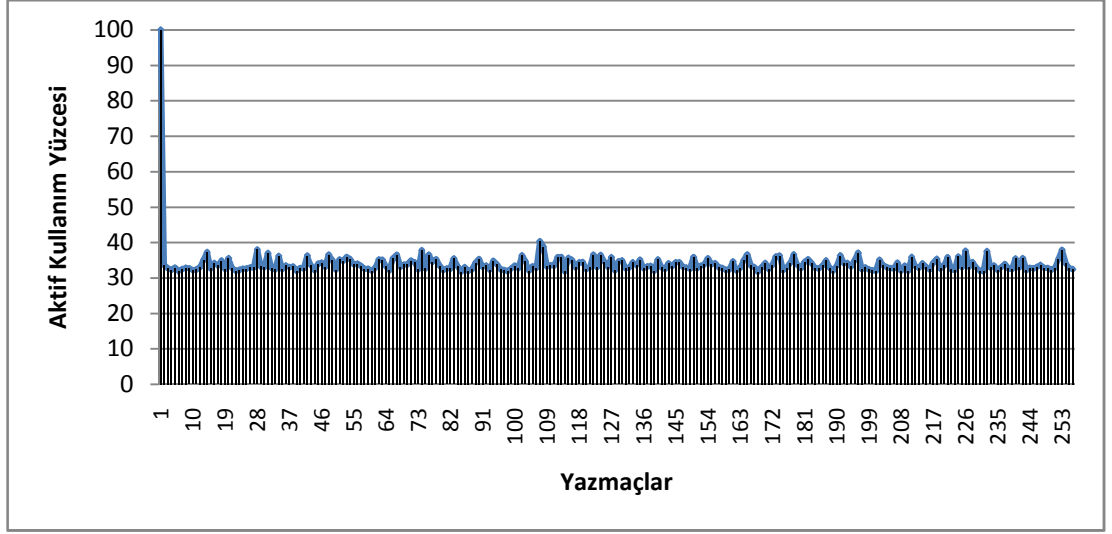
Yazmaç öbeğine yazılan değerlerin birden fazla okunabileceği veya hiç okunmayacağı göz önüne alındığında Şekil 4.2'te SPEC 2000 programları çalışırken yazmaç öbeğinden okunan değerlerin yaklaşık %92'sinin 34 veya daha az bitle gösterilebileceği görülmektedir.



Şekil 4.2. Spec 2000 programlarının çalıştırılması sırasında yazmaç öbeğinden okunan değerlerin darlık yüzdesi

Yazmaçların kullanılmayan üst bitleri olduğu gibi, kullanılmadığı zamanlarda mevcuttur. Yazmaçlar yeniden adlandırma sırasında kullanıma açılırlar ve tamamlanma sırasında bırakılıp, tekrar serbest yazmaç listesine konulurlar. Bu sürenin dışında yazmaç, serbest yazmaç listesinden alınana kadar bir görev üstlenmez. Şekil 4.3'te SPEC 2000 programlarında kullanılan yazmaçların aktif

kullanım yüzdesi görülmektedir. 1 numaralı yazmaç dışında diğer tüm yazmaçlar programların çalışma sürelerinin sadece %33'ünde aktif olarak kullanılmaktadır.



Şekil 4.3. Spec 2000 programlarının yazmaç öbeğindeki yazmaçların kullanım yüzdeleri

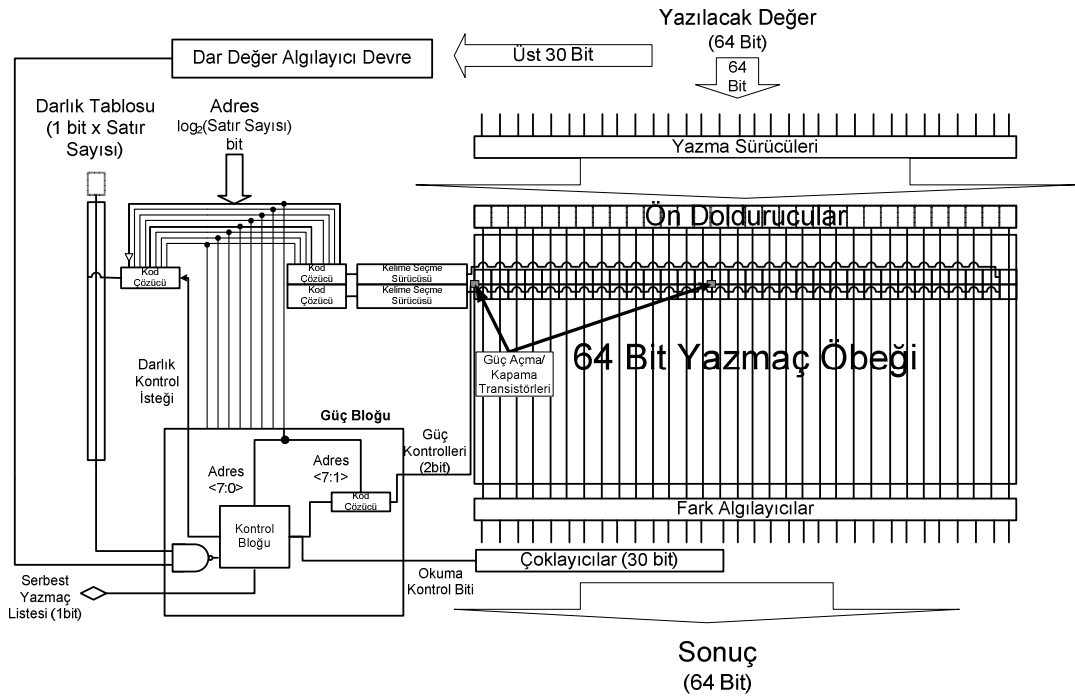
#### 4.4. Yazmaç Öbeğine Devingen Olarak Güç Kesintisi Uygulanması

Yazmaçların kullanım ömürleri ve kullandıkları değerlerin dar oldukları göz önüne alındığında bölüm 2.4'te anlatılan güç kesme yöntemleri yazmaç öbeğine uygulanabilir.

Bit hücrelerinin güç telleri, ilgili yazmaç kullanımında olmadığı zaman kesilebileceği gibi sakladığı değer dar olduğu durumlarda da üst bitleri kapatılabilir. Bu işlemler sırasında sadece mimari olarak değil devre düzeyinde de bazı sınırlayıcı faktörler vardır. Bölüm 2.4'te bir SRAM tablosundaki satırların teker teker kapatılmasının çizim tekniğine bağlı olduğu anlatılmıştı. Alan tasarrufu amacıyla SRAM hücrelerinin  $V_{DD}$  ve toprak düğümleri iki satır tarafından paylaşılmaktadır. Bu sebepten dolayı güç kesileceği zaman bir değil iki satırın aynı anda gücünün kesilmesi söz konusudur. Bununla birlikte bir yazmacın gücü kesilmesi söz konusu olduğunda güç tellerini paylaştığı diğer satırında (komşu) aynı özellikleri paylaşması gerekmektedir. Örneğin, bir satıra yazılacak değer dar olduğunda üst bitlerini kapatmak için komşu satırın ya dar ya da kullanımda olmaması gerekmektedir. Bu

amaçla bir tane bit vektörü tutularak tüm yazmaçların içindeki değerlerin dar veya geniş olduğuna bakılmaktadır.

Şekil 4.4'te yazmaç öbeğinin sızıntı akımlarını azaltmak amacıyla kullanılan bileşenlerin blok şeması görülmektedir. Yazmaç öbeğinin üzerine eklenen güç açma kapama transistörleri bir yazmaç çiftinin üst bitlerini kapamaya ve çifti tamamen uyutma görevini üstlenmektedir. Bu transistörlere uygulanan kontrol bitleri ise "Güç Bloğu" tarafından belirlenmektedir. Bu blok çevresel bileşenleri kullanarak ilgili yazmaç çiftinin kapatılıp kapatılmayacağına karar verir.



Şekil 4.4. Yazmaç öbeğinin güç ayrıştırılması için düzenlenmesi

Şekil 4.4'teki güç bloğuna, dar değer algılayıcı devre, darlık tablosu, serbest yazmaç tablosu ve adresler giriş olarak verilmektedir. Yazmaç öbeğine bir değer yazılacağı zaman, istenen darlık belirlenmesi (örnekte 30 bit) dar değer algılayıcı devre tarafından yapılır. Eğer değer darsa, algılayıcı devreden 1 sinyali çıkar. Aynı zamanda yazılacak satırın komşusunun sakladığı değerın dar olup olmadığına bakılır. Yazmaç öbeğine gelen adresin en anlamsız son biti terslenerek dar değer tablosuna adres olarak verilir. Bu sayede yazılacak satırın komşusu bulunmuş olur. Örneğin, 16 nolu satıra bir değer yazılacak ise 17 nolu adresteki değerın dar olup olmadığına bakılır. 16 sayılı ikilik sistemde (8 bit adres genişliğinde) 00010000 şeklinde

gösterilirken, en anlamsız biti terslendiğinde ikilik düzende 00010001 olarak gösterilir ve ondalık olarak 17 sayısı elde edilir. Benzer şekilde 17 nolu satıra bir yazma işlemi olsaydı, aynı sayının son biti terslenerek 16 nolu satıra komşu olarak bakılır.

Darlık tablosu bir bit vektörüdür. Bu vektör, bir mandal (latch) dizisi veya tek bitlik çok satırlık bir SRAM tablosu olarak tasarlanabilir. Yazmaç öbeğindeki her satırın içindeki bilgiye göre eğer darsa 0, geniş ise 1 biti tutarak yazmaç öbeğine yazma sırasında, güç açma kapama için karar verilmesine yardımcı olur. Yazmaç öbeğinden okuma sırasında ise okunacak değer dar olduğu durumlardaki genişletme işlemi için bilgi sağlar. Yazmaç öbeğine bir değer yazıldıktan sonra bu vektör güç bloğu tarafından güncellenir.

Darlık tablosundan okunan değer ve dar değer algılayıcı devreden gelen değerler güç bloğundaki “VE DEĞİL” kapısından geçirilerek kontrol bloğuna iletilir. İletilen değer 1 ise yazmaç kapatılabilir demektir. Bu durumda doğru satıra doğru kontrol bitleri uygulanması gerekmektedir. Her satır çift çift kapatıldığı için tablodaki satır sayısının yarısı kadar güç kontrol biti vardır. Bu sebepten dolayı, yazmaç öbeğine gelen adres bitlerinin bir sağa kaydırılmış hali güç bloğuna verilerek adresin değeri ikiye bölünmüş olur. Burada bir kaydırıcı kullanmaya gerek yoktur, adres tellerinin en anlamlı 7 biti adres olarak verilerek adres genişliğinin bir bit azaltılması yeterlidir.

Güç bloğunun karar aşamasında önemli verilerden biri olan yazmacın serbest olup olmadığı, yeniden adlandırma sırasında tutulan serbest yazmaç listesinden almaktadır. Yazmacın serbest olup olmadığı da darlık tablosundakine benzer bir bit vektörü ile takip edileceği gibi yeniden adlandırma aşamasındaki listeye erişildiği varsayılmaktadır.

Mimari düzeyinde dikkat edilmesi gereken husus ise, yazmaçların açılma süreleridir. Bir değer yazmaç öbeğine yazma amaçlı bir erişim yapacağında yazmaç kapalı ise bu durum tüm boru hattını etkilemektedir. Bu nedenle yazmacın durumu ne olursa olsun ilgili yazmaç, serbest yazmaç listesinden çıktığı anda gücü açılır.

Satırların gücünü kontrol eden bloğun algoritması Çizelge 4.1’de gösterilmektedir. Karar aşamasının 3 tane farklı durumu bulunmaktadır. İlk adım yazmaç tahsis

edilmesi anında yazmaç çiftinin açılmasıdır. 2. Durum ise geri yazım durumunda yazılacak değer dar ise, komşu yazmaca bakılır eğer komşu yazmaçtaki değer dar ise veya kullanılmıyor ise yazmaç yarı açık duruma getirilir. Burada yarı açık ile kast edilen değer SRAM tablosunda kullanılan dar değer algılayıcı devre ve güç transistörlerinin bir satırı ne kadar kapatabildiğidir. 3. durumda ise yazmacın ömrünü tamamladığı serbest bırakılma anıdır. Bu durumda bırakılan yazmaç geniş bir değere sahip ise bulunduğu yazmaç çiftine güç kesintisi uygulanamamış demektir. Bu yazmacın komşu değerine bakılır ve eğer bu değer darsa yazmaç çifti yarı açık yapılır. Komşu yazmaç kullanılmıyorsa bu yazmaç çifti tamamen uyutulur.

Çizelge 4.1. Güç bloğu kontrol algoritması

|    | Durum                          | Algoritma  |
|----|--------------------------------|--|
| 1. | Yazmaç tahsis edilmesi anı     | Yazmaç çifti açılır.   |
| 2. | Geri yazım (writeback) anı     | <p>Yazılacak değer dar ise:</p> <p>Komşu yazmaç kullanılmıyor ise:</p> <p>-Yazmaç çiftini yarı açık yap.</p> <hr/> <p>Komşu yazmaç kullanılıyor ise:</p> <p>Komşu değer dar ise:</p> <p>-Yazmaç çiftini yarı açık yap.</p>               |
| 3. | Yazmaç serbest bırakılması anı | <p>Komşu yazmaç kullanılıyor ise:</p> <p>Komşu değer dar ise:</p> <p>Yazmaç değeri (serbest bırakılanın) geniş ise:</p> <p>-Yazmaç çiftini yarı açık yap.</p> <hr/> <p>Komşu yazmaç kullanılmıyor ise:</p> <p>-Yazmaç çiftini kapat.</p> |

Uygulanan bu algoritma, Şekil 4.4'teki güç bloğunun içinde bulunan kontrol bloğunda gerçekleşir. Küçük bir durum makinesi ile gerçekleştirilecek bu devrenin sisteme enerji ve alan açısından bir yük getirmediği varsayılmaktadır.

#### 4.5. Benzetimlik Ortamı

Mimari açısından yapılan değişikliklerin performansa olan etkisini ve veri saklayan bileşenlere ne amaçla kaç kere erişildiğini saptamak amacıyla PTLsim [43] benzetim aracı kullanılmıştır. PTLsim x86 ve x86-64 buyruk kümesini kullanan, C++ dilinde yazılmış çok yönlü mikroişlemciler için saat tutarlı tek açık kaynak ve bedava yazılımdır. Intel Pentium 4 sonrası işlemcileri ve AMD Athlon 64 benzeri işlemcilerin benzetimliğini yapmaktadır. Çizelge 4.2'te benzetimlik sırasında kullanılan sanal işlemcinin parametreleri verilmiştir.

Çizelge 4.2. Benzetim parametreleri

| Parametre                     | Yapılandırma   |
|-------------------------------|--|
| Makine Genişliği              | 4 komut getir, 4 komut yayınla, 4 komut sonlandır  |
| Pencere Boyu                  | 16 satır yayın kuyruğu,<br>80 satır yükle/sakla kuyruğu,<br>128 satır YSB,<br>256 satır tamsayı yazmaç öbeği   |
| İşlem Birimleri ve Gecikmeler | Tamsayı AMB (6/1),<br>yükle/sakla birimi (2/2),<br>tamsayı çarpma (2/4),<br>tamsayı bölme (1/32),<br>kayan nokta toplama (2/6),<br>kayan nokta çarpma (2/6),<br>kayan nokta bölme (2/6). |



|                          |   |
|--------------------------|---|
|                          | Toplam 6 tamsayı, 2 kayan nokta işlem birimi                            |
| L1 Komut<br>Önbelleği    | 32 KB, 4 yollu kümeli ilişkili, 64 bayt satır, 1 çevrim isabet zamanı   |
| L1 Veri<br>Önbelleği     | 16 KB, 4 yollu kümeli ilişkili, 64 bayt satır, 2 çevrim isabet zamanı   |
| L2 Tümüleşik<br>Önbellek | 256 KB, 16 yollu kümeli ilişkili, 64 bayt satır, 6 çevrim isabet zamanı |
| L3 Tümüleşik<br>Önbellek | 4 MB, 32 yollu kümeli ilişkili, 64 bayt satır, 14 çevrim isabet zamanı  |
| BTB                      | 1024 satır, 4 yollu kümeli ilişkili                                     |
| Dallanma<br>Tahmin       | 64K satır çift durumlu ve iki aşamalı birleşik                          |
| Bellek                   | 140 çevrim gecikme  |

Benzetimlik üzerinde çalıştırılan SPEC 2000 denektaşları programlarının listesi [42] Çizelge 4.3'te tamsayılar için ve Çizelge 4.4'te kayan noktalar için verilmiştir.

Çizelge 4.3. Tamsayı denektaşları programları

| Program | Yazıldığı Dil | Açıklama                                   |
|---------|---------------|--|
| gzip    | C             | Veri sıkıştırma                            |
| vpr     | C             | FPGA için fiziksel yer ve yol belirlenmesi |

|         |     |                                       |
|---------|-----|---------------------------------------|
| gcc     | C   | C programlama dili derleyicisi        |
| mcf     | C   | Kombinasyonsal iyileştirme            |
| crafty  | C   | Satranç oyunu                         |
| parser  | C   | Kelime işleme                         |
| eon     | C++ | Sanal gerçeklik                       |
| perlbmk | C   | Perl programlama dili                 |
| gap     | C   | Grup teorisi, yorumcu                 |
| vortex  | C   | Nesneye dayalı veritabanı             |
| bzip2   | C   | Veri sıkıştırma                       |
| twolf   | C   | Yerleştirme ve Düzenleme Benzetimliği |

Çizelge 4.4. . Kayan nokta denektaşları programları

| Program | Yazıldığı Dil | Açıklama  |
|---------|---------------|---|
| wupwise | Fortran 77    | Fizik: Kuantum kromo dinamikleri                  |
| swim    | Fortran 77    | Akışkan modelleme                                 |
| mgrid   | Fortran 77    | 3 boyutlu Problem Çözümü                          |
| applu   | Fortran 77    | Parabolik / eliptik kısmi diferansiyel denklemler |
| mesa    | C             | 3 boyutlu grafik kütüphanesi                      |
| galgel  | Fortran 90    | Akışkanlar dinamiği                               |
| art     | C             | Görüntü tanıma / yapay sinir ağları               |
| equake  | C             | Sismik dalga benzetimliği                         |

|          |            |  |
|----------|------------|--|
| facerec  | Fortran 90 | Görüntü işleme: yüz tanıma                             |
| ampp     | C          | Sayısal kimya  |
| lucas    | Fortran 90 | Sayı teorisi   |
| fma3d    | Fortran 90 | Sonlu element çarpışı benzetimliği                     |
| sixtrack | Fortran 77 | Yüksek enerjili nükleer fizik ivmelendiricisi tasarımı |
| apsi     | Fortran 77 | Meteoroloji: Hava kirliliği dağılımı                   |

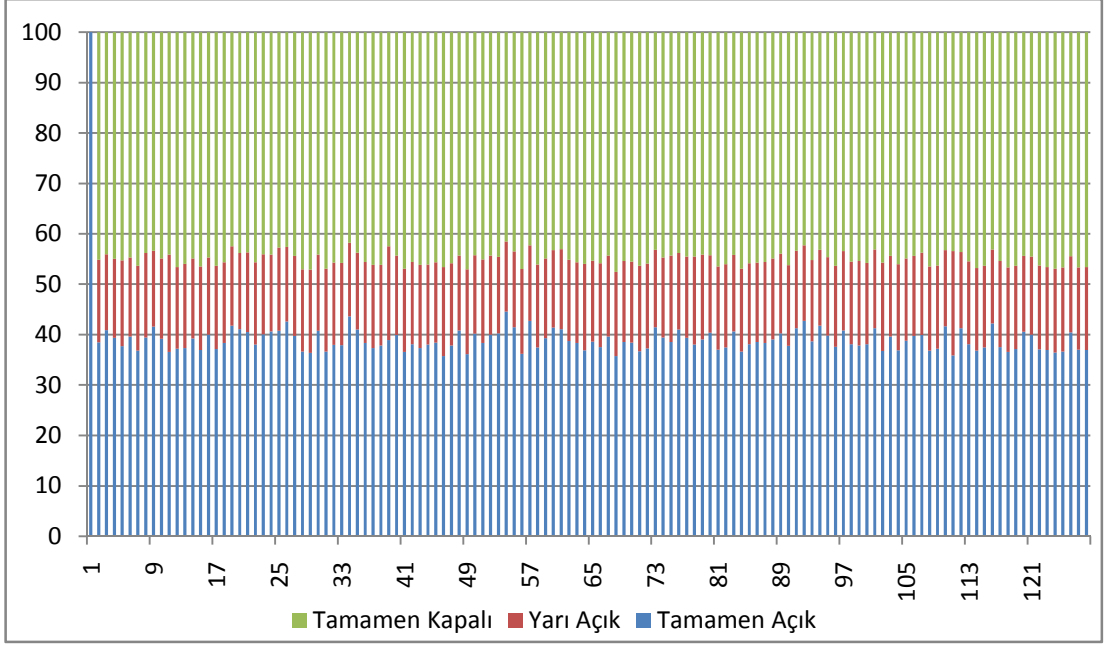
Enerji hesaplamaları UMC 90nm CMOS kütüphaneleri tamamen özel (full custom) fiziksel devre serimi tasarımları yapılarak Cadence IC paket programlarında hesaplanmıştır.

#### 4.6. Sonular

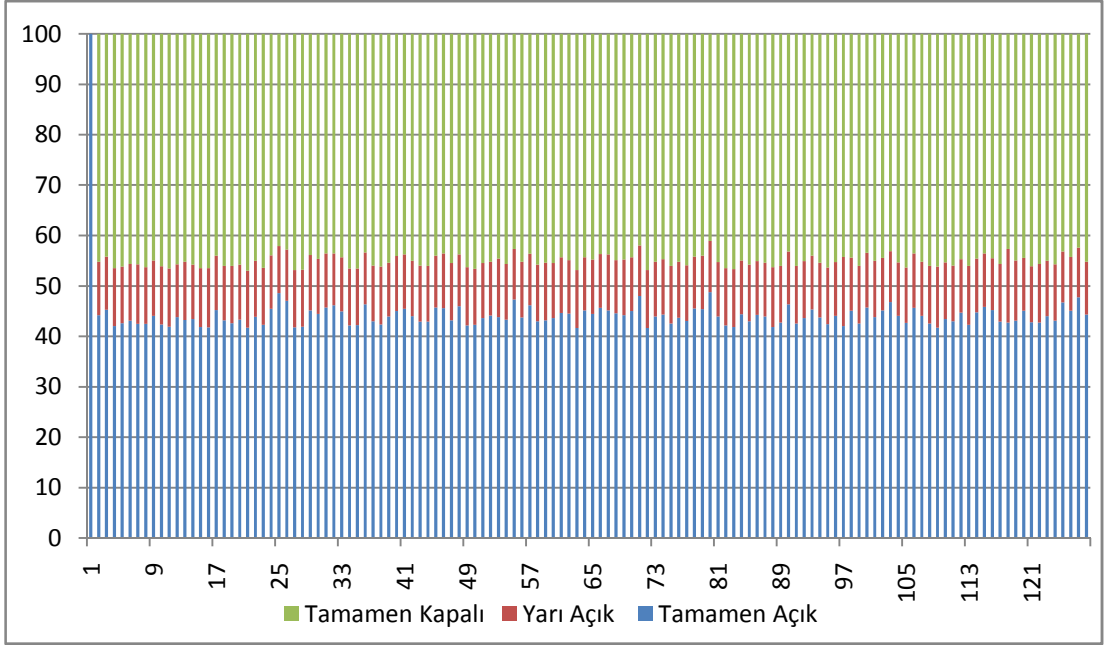
Yazma beđine uygulanan gc tknetimi azaltmaya ynelik yntem iin blm 4.3'teki n verilere de dayanarak 3 farklı yapılandırma denenmiřtir. Her yapılandırmada yazma beđinin bir satırının ve st kısımlarının daha ok kapatılması amalanmıřtır. Buna gre yazma beđine gelecek deđerlerin 34, 31 ve 16 bit dar olacađı var sayılırken buna bađlı olarak dar deđer algılama devrelerinin geniřliđi aynı sırayla 30, 33 ve 48 bitlik olmuřtur.

řekil 4.5'te 34 bit darlıđa sahip yazma beđinin tm SPEC denektařlarındaki durum yzdeleri verilmiřtir. 128 yazma ifti, programların alıřtıđı srenin yaklařık %45'inde tamamen kapatabilirken yaklařık %16'sının da yarı aık duruma getirilip bir iftte bulunan 128 bit hcresinin 60'ının kapanması sađlanmıřtır.

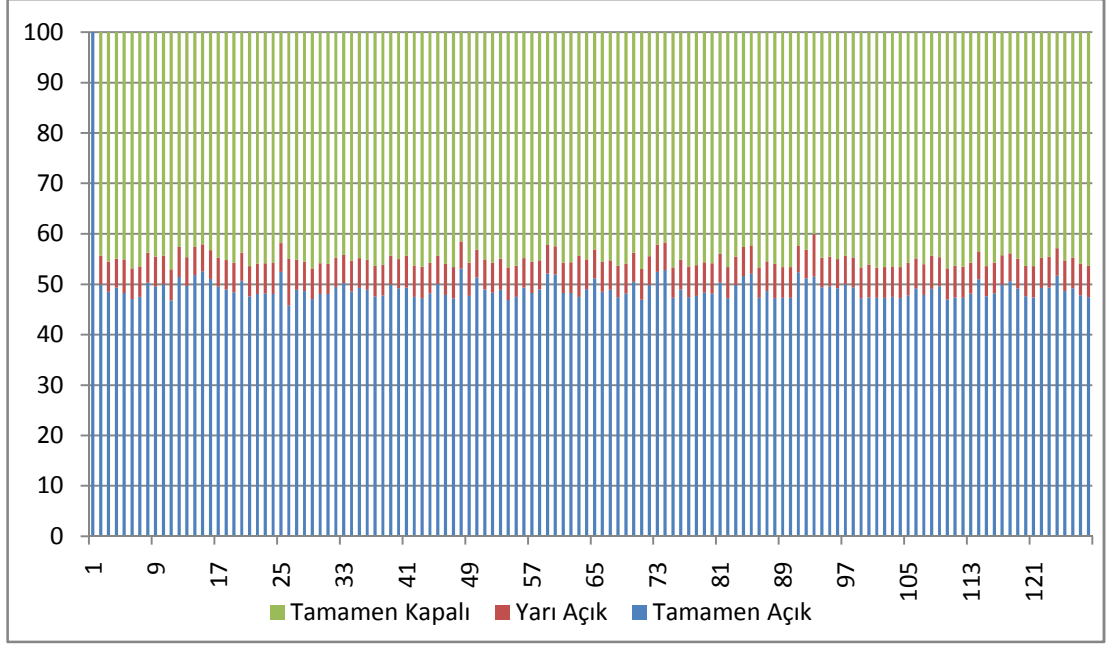
Yazma iftlerinin tamamen kapalı olması belirlenen darlıđa bađlı olmadıđı iin 31 ve 16 bite sahip yapılandırmalarda da yaklařık olarak %45 tamamen kapalıđa ulařılmıřtır. Bunun yanında 31 bitlik yapılandırmada yazma iftleri alıřma mrlerinin %11'inde yarı aık duruma geebilirken, 16 bitlik yapılandırmada bu oran %6'ya dřmektedir. Kaybedilen yarı aık olma yzdeleri, yazma iftlerinin tamamen aık olduđu anlamına gelmektedir. Bu sonular 31 bit iin řekil 4.6'dan ve 16 bit iin řekil 4.7 grlebilir.



Şekil 4.5. 34 Bit darlığa sahip yazmaç öbeği çiftlerinin açık, yarı açık ve kapalı olma yüzdeleri



Şekil 4.6. 31 Bit darlığa sahip yazmaç öbeği çiftlerinin açık, yarı açık ve kapalı olma yüzdeleri



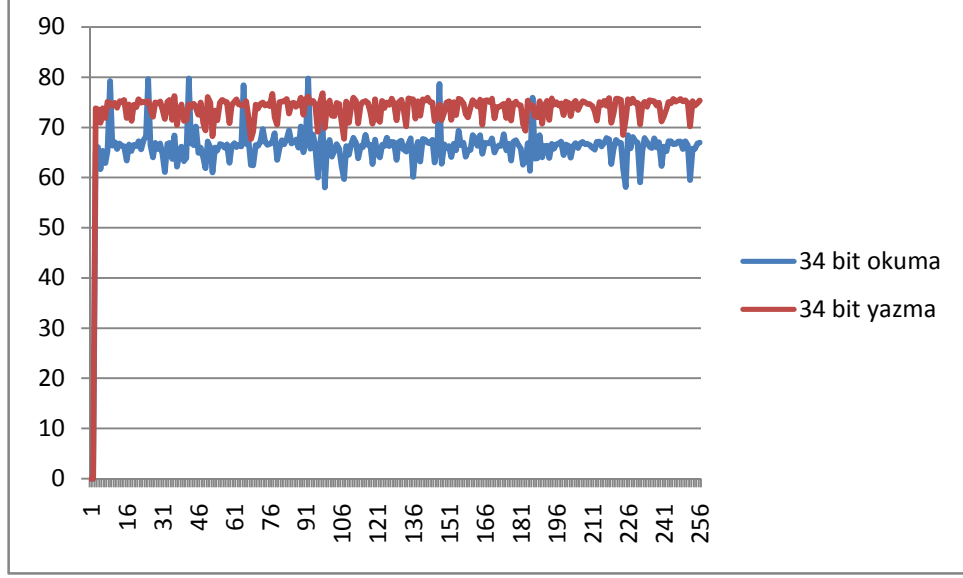
Şekil 4.7. 16 Bit darlığa sahip yazmaç öbeği çiftlerinin açık, yarı açık ve kapalı olma yüzdeleri

Yazmaçların durumunu gösteren Şekil 4.5, Şekil 4.6 ve Şekil 4.7’de 1. yazmaç çifti sürekli açık görülmektedir. Bunun nedeni veri yolu genişliğinde 0 değeri okunacağı ve yazılacağı zaman bu yazmaçtan yararlanılmasıdır.

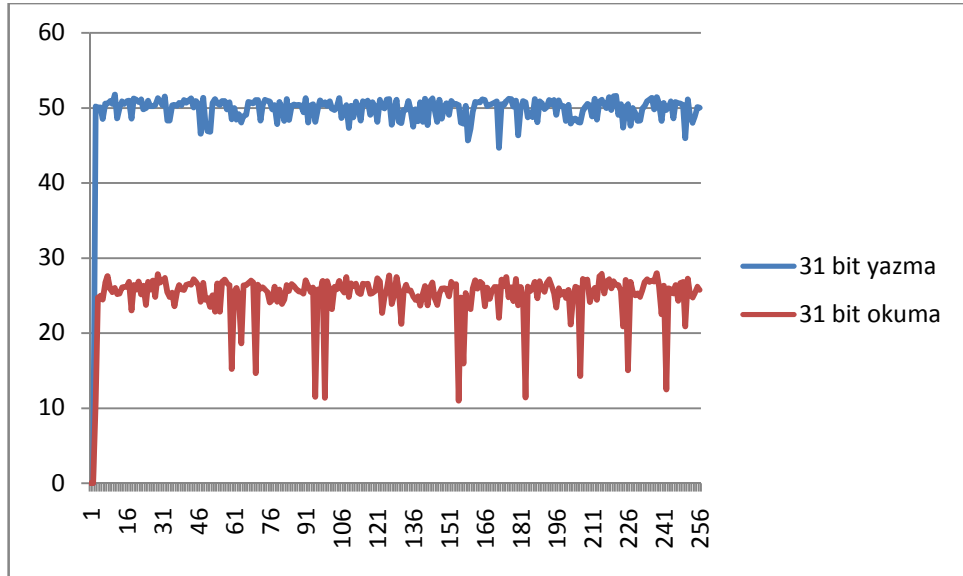
Yazmaçların uyutulması sızıntı akımı nedeniyle harcanan durağan gücü engellemektedir. Dar değerlerin açığa çıkarılan bir yararı ise yazılan ve okunan değerler sırasında üst bitlerin kullandığı SRAM tablosu bileşenlerinin çalıştırılmamasıdır. Örneğin üst kısmı kapatılmış 34 bitlik yazmaç çiftinden herhangi bir değer okunacağı zaman üst bitlerin kullandığı ön doldurucular ve fark algılayıcı devreler kullanılmamaktadır. Dar değerlerin yazıldığı durumda da yazma sürücüleri kullanılmadığından devingen enerji tasarrufu sağlanır.

Şekil 4.8’te 34 bitlik dar değer yapılandırmasına sahip yazmaç öbeğine yazılan dar değerlerin, tüm yazılan değerlerin yaklaşık %73’ü olduğu görülmektedir. Yazmaç öbeğinden yapılan okumaların ise %65’i dar olarak saptanmıştır. Şekil 4.1 ve Şekil 4.2’deki en büyük kırılma noktasının 34 bit darlığı için olmasına rağmen üst kısımları daha çok tekrar eden bir değer seçilerek devingen ve durağan enerjiden daha çok tasarruf etmek mümkündür. Bu nedenle 31 bit ve 16 bit darlıkları için yapılan denemelerde yapılacak değer okuma ve yazmaların sayıca azalması

beklenmesine rağmen tasarruf edilecek üst bitlerin daha çok enerji tasarrufu sağlayıp sağlamayacağına bakılmıştır. 31 bit yapılandırmasında yazılan değerlerin %49'unun 31 bit dar, okunan değerlerin ise ortalama olarak %25 dar olduğu görülmüştür. Şekil 4.9'te bu ortalama değerlerin tüm yazmaç öbeği satırları için dağılımı görülmektedir.

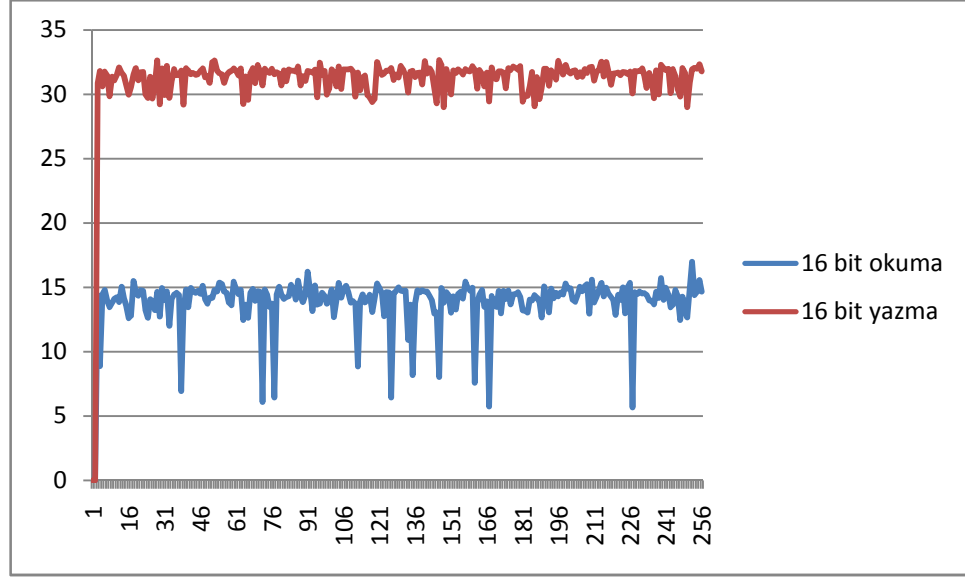


Şekil 4.8. Yazmaç öbeğine yazılan ve yazmaç öbeğinden okunan 34 bitlik dar değer oranları



Şekil 4.9. Yazmaç öbeğine yazılan ve yazmaç öbeğinden okunan 31 bitlik dar değer oranları

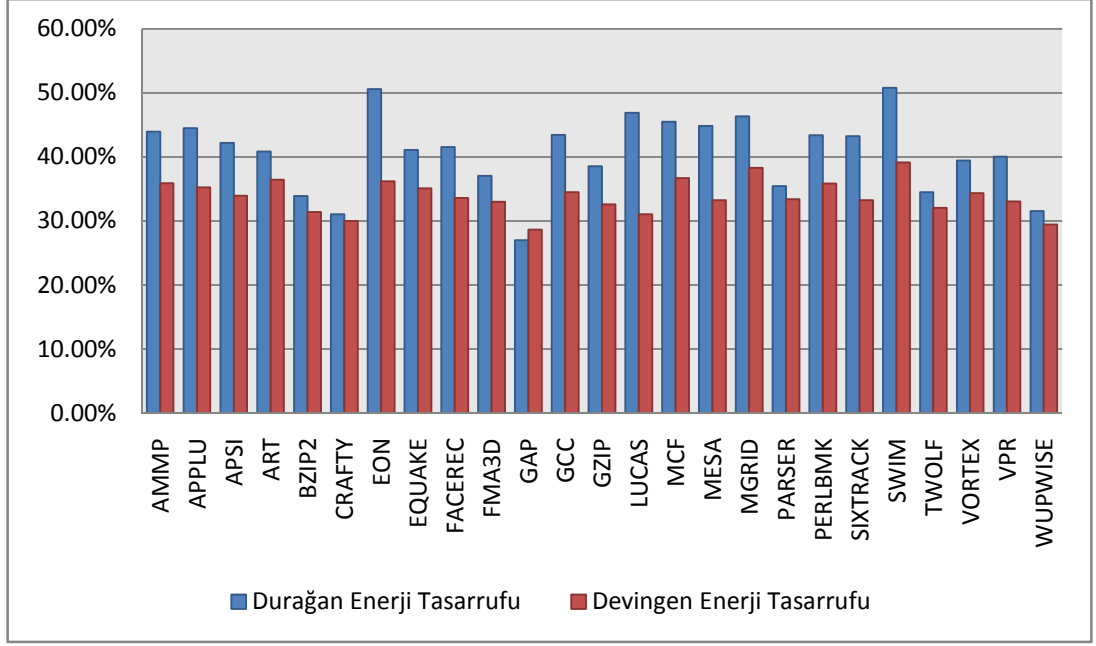
En çok üst bit tasarrufu olan 16 bit yapılandırmaya sahip yazmaç öbeğinde ise yazılan 16 bitlik dar değerlerin ortalaması iyice düşerek sadece %31'lik bir bölümü kapsayabilmiştir. Benzer şekilde yazmaç öbeğinden okunan değerlerin ortalama %13'ünün dar olduğu Şekil 4.10'da görülmektedir.



Şekil 4.10. Yazmaç öbeğine yazılan ve yazmaç öbeğinden okunan 16 bitlik dar değer oranları

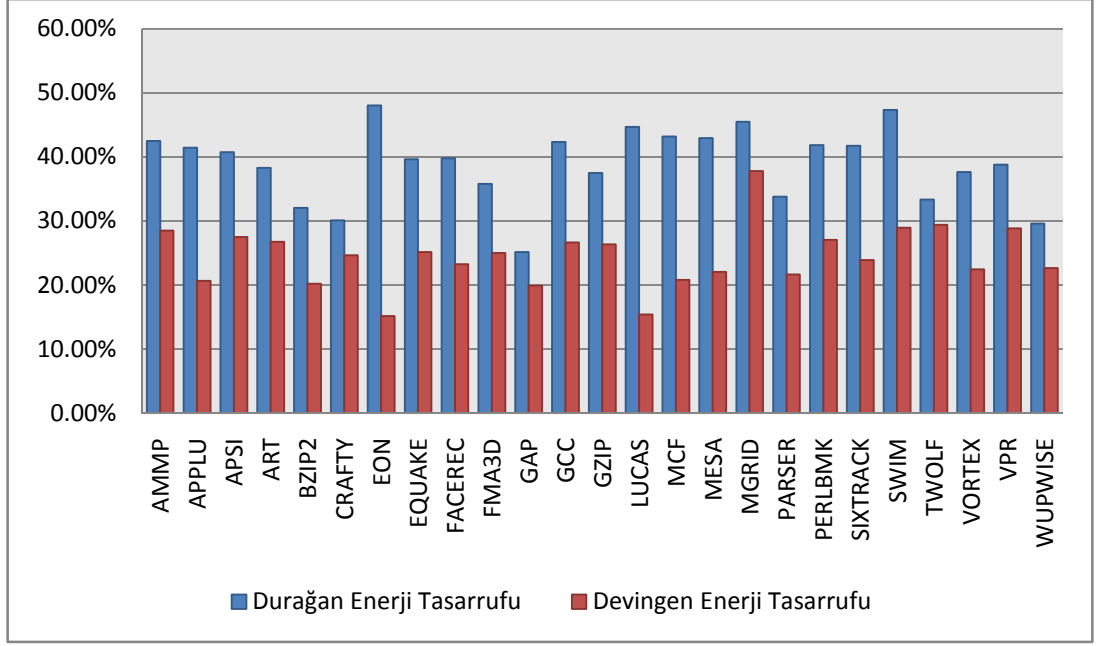
Yazmaç çiftlerinin açık, yarı açık ve kapalı kalma süreleri baz alınarak hesaplanan sızıntı akımına bağlı durağan güç tüketimleri tüm SPEC denektaşları için hesaplanmıştır. Hesaplanan bu değerler, üzerinde hiç değişiklik yapılmamış ve aynı denektaş programların çalıştırılıp durağan güç tüketiminin hesaplandığı temel bir sistemle karşılaştırılarak sağlanan tasarruf bulunmuştur. Temel sistemin durağan enerji tüketiminin nasıl hesaplandığı bölüm 2.3.2'te ele alınmıştır. Yapılandırmalardaki devingen enerji tüketimlerinin dar değer okuma için olanlarında değerlerin çoklanmasında tüketilen enerji ve dar değer yazılması sırasında fazladan olan dar değer anlama devresinin tükettiği enerji hesaplara katılmıştır.





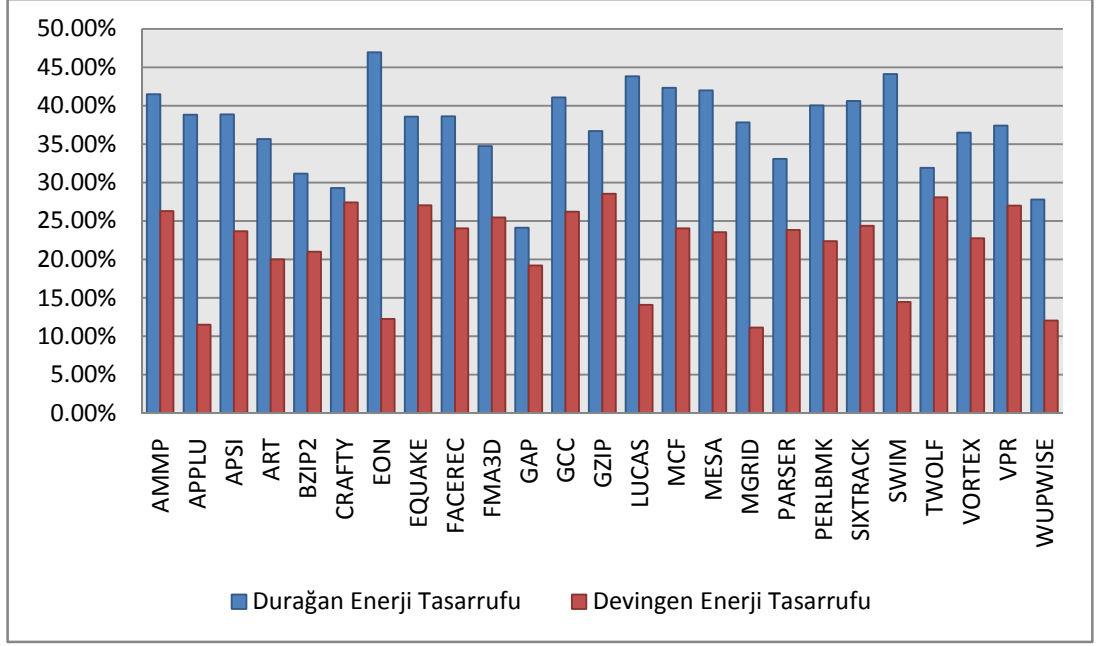
Şekil 4.11. 34 bit yapılandırmasında çalışan SPEC denektaş programlarında sağlanan durağan ve dinamik enerji tasarrufu

34 bit yapılandırmasında sağlanan durağan ve devingen enerji tasarrufları tüm denektaş programları için Şekil 4.11’te karşılaştırılmıştır. Aynı karşılaştırmalar 31 bit yapılandırması için Şekil 4.12’de, 16 bit yapılandırması içinse Şekil 4.13’te yapılmıştır. Yapılandırmalar sonucunda ulaşılan güç tasarrufu oranları fark olsa da durağan enerji için %30-%50 aralığında, devingen enerji içinse %12-%48 aralığında değişmektedir. Böylece uygulanan yöntemin geçerliliği kanıtlanmıştır.



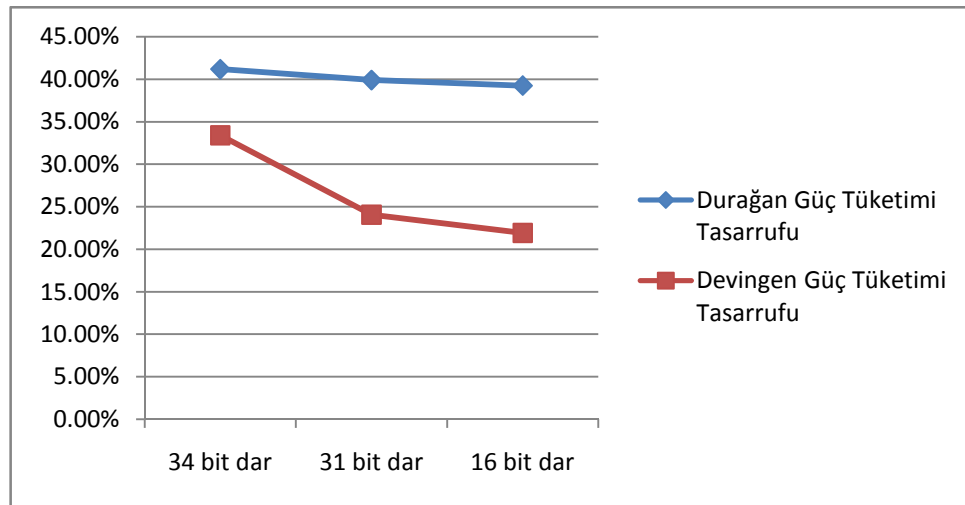
Şekil 4.12. 31 bit yapılandırmasında çalışan SPEC denektaş programlarında sağlanan durağan ve dinamik enerji tasarrufu

Son olarak yapılandırmalarda tüm programların çalıştırılıp güç tüketimleri değiştirilmemiş taban bir sistemle karşılaştırıldığında sağlanan güç tasarrufları Şekil 4.14'te karşılaştırılmıştır. 34 bit yapılandırması %42 durağan güç tüketimi tasarrufu ve %33.5 devingen güç tasarrufu ile en iyi yapılandırma olarak belirlenmiştir. Tüm bu yöntemler uygulanırken mikroişlemcilerinin çevirim başına işlenen buyruk oranı değişmemiştir. Bir başka deyişle işlemcinin başarımı değişmemiştir. Bölüm 2.4'te gösterildiği üzere uygulanan uyuma transistörlerinin alanları çok küçük olduğu için SRAM tablosunun boyutu artmamıştır denebilir. Bu transistörlerin kontrol telleri ise farklı metal katmanları kullanarak üst üste çizildiğinden var olan yollardan gitmektedir. Böylece uygulanan yöntemin silikon alanına herhangi bir yükü bulunmamaktadır.



Şekil 4.13. 16 bit yapılandırmasında çalışan SPEC denektaş programlarında sağlanan durağan ve dinamik enerji tasarrufu

31 bit ve 16 bit yapılandırmalarında sayıca daha çok bit hücresi uykuya alınabilmesine rağmen sağlanabilen dar değer oranı düştüğünden, enerji tasarrufu azalmıştır. Yapılandırmaların darlıklarının belirlenmesi ve çıkan sonuçlar açısından benzer bir yaklaşım [30]'ta yazmaç öbeğine kümelerle ayırırken, [36]'da ise yazmaç öbeğini üst ve alt kısım olarak ikiye bölerken gerçekleştirilmiştir.



Şekil 4.14. Güç tasarrufu uygulanması amacıyla uygulanan tüm yapılandırmaların enerji tasarrufu karşılaştırması

## 5. SONUÇ

Gerçekleştirilen tez çalışmasında çağdaş mikroişlemcilerdeki veri saklayan SRAM tabloları derinlemesine incelenmiştir.

İlk aşamada bu tabloların başarımları, güç tüketimi ve performans açısından analizleri yapılmıştır. 8 ve 12 portlu SRAM tabloları için durağan ve devingen güç tüketimleri matematiksel olarak modellenmiştir. Bununla beraber bu tabloların alanları da fiziksel olarak devre serimi yapılmadan hesaplanabilecek şekilde analiz edilmiştir.

SRAM tablolarında uygulanmak üzere, CMOS devrelerin kapı genişliği azaldıkça artan, sızdırma akımlarını önlemeye yönelik yöntemlerin küçük bir incelemesi yapılarak VLSI devre tasarımı düzeyinde nasıl gerçekleştirilebileceği gösterilmiştir.

Çağdaş mikroişlemcilerin mimari özelliklerinden yararlanarak, bir mikroişlemcinin yazmaç öbeğindeki bit hücreleri devingen olarak açılıp kapatılmıştır. Mikroişlemcinin mimari durumuna göre bazı durumlarda yazmaçların üst bitleri, bazı durumlarda ise yazmaç satırının hepsi uyutulmuştur. Böylece daha sızdırmaya bağlı güç tüketimi büyük oranda engellenmiştir. Uygulanan farklı yapılandırmalar dahilinde en iyi durumda durağan güç tüketiminden %42, devingen güç tüketiminden ise %33.5 oranında tasarruf edilmiştir. Bu tasarruf sırasında SRAM tablosunun silikon üzerinde kapladığı alan artmadığı gibi mikroişlemcinin başarımlarında da herhangi bir azalma olmamıştır. Tüm denektaşları programlarında sağlanan tasarruf oranının standart sapması devingen enerji tasarrufu için %6, devingen enerji için %2 civarındadır. Böylece uygulanan yöntemin genel geçer bir yöntem olabileceği görülmektedir.

Uygulanan bu yöntem ek olarak, yazmaç öbeğine eklenen uyutma transistörlerini farklı seviyelerde uyutarak daha çok enerji tasarrufu sağlanabilir. Böylece uyutma transistörleri daha büyük eşik değeri sahip pMOS'lardan seçilirken, SRAM bileşenleri daha küçük eşikli transistörlerden seçilebilir. Mimaride uygulanan zamanlama özellikleri sağlanabilirse aynı yöntem veri saklayan diğer bileşenlere uygulanabilir.

## KAYNAKLAR

- [1]. Kang, S.-M., Leblebici, Y.: CMOS Digital Integrated Circuits Analysis and Design 3rd edn. McGraw-Hill (2003)
- [2]. Veendrick, H. J. M.: Short-circuit dissipation of static CMOS circuitry and its impact on the design of buffer circuits. Solid-State Circuits, IEEE Journal of 19(4), 468-473 (1984)
- [3]. Weste, N., Eshraghian, K.: Principles of CMOS Vlsi Design A Systems Perspective 2nd edn. Addison-Wesley (1992)
- [4]. Sery, G., Borkar, S., De, V.: Life is CMOS: why chase the life after? In : Proceedings of the 39th Annual Design Automation Conference (DAC), New Orleans,USA, pp.78-83 (2002)
- [5]. Mistry, K., Allen, C., Auth, C., Beattie, B., Bergstrom, D., Bost, M., Brazier, M., Buehler, M., Cappellani, A., Chau, R., Choi, C.-H., Ding, G., Fischer, K., Ghani, T., Grover, R., Han, W., Hanken, D., Hattendorf, M., He, J., Hicks, J., Huessner, R.: "A 45nm Logic Technology with High-k+Metal Gate Transistors, Strained Silicon, 9 Cu Interconnect Layers, 193nm Dry Patterning, and 100% Pb-free Packaging. In : Electron Devices Meeting, pp.247-250 (2007)
- [6]. Tiwari, V., Singh, D., Rajgopal, S., Mehta, G., Patel, R., Baez, F.: Reducing power in high-performance microprocessors. In : Proceedings of the 35th Annual Design Automation Conference (DAC'98), San Fransisco, pp.15-19 (1998)
- [7]. Bernstein, K., Chuang, C., Joshi, R., Puri, R.: Design and CAD Challenges in sub-90nm CMOS Technologies. In : IEEE/ACM international Conference on Computer-Aided Design (ICCAD), Washington,DC, p.129 (2003)
- [8]. Wang, S., Yang, H., Hu, J., Ziavras, S. G.: Asymmetrically Banked Value-Aware Register Files. In : IEEE Computer Society Annual Symposium on VLSI, pp.363-368 (2007)
- [9]. Palacharla, S., Jouppi, N., Smith, J.: Complexity-effective superscalar processors. In : International symposium on Computer architecture (ISCA '97), pp.206-218 (1997)
- [10]. Farkas, K. I., Jouppi, N. P., Chow, P.: Register file design considerations in dynamically scheduled processors. In : International Symposium on High-Performance Computer Architecture, pp.40-51 (1996)

- [11]. Ponomarev, D. V., Kucuk, G., Ergin, O., Ghose, K., Kogge, P. M.: Energy-efficient issue queue design. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 11(5), 789-800 (2003)
- [12]. Amrutur, B. S., Horowitz, M. A.: Speed and power scaling of SRAM's. *IEEE Journal of Solid-State Circuits* 35(2), 175-185 (2000)
- [13]. Powell, M., Yang, S.-H., Falsafi, B., Roy, K., Vijaykumar, T. N.: Gated-Vdd: a circuit technique to reduce leakage in deep-submicron cache memories. In : *International Symposium on Low Power Electronics and Design (ISLPED'00)*, pp.90-95 (2000)
- [14]. Chen, S., Lin, J.: Experiences of low power design implementation and verification. In : *Proceedings of the 2008 Asia and South Pacific Design Automation Conference, Seoul*, pp.742-747 (2008)
- [15]. Halter, J. P., Najm, F. N.: A gate-level leakage power reduction method for ultra-low-power CMOS circuits. In : *Custom Integrated Circuits Conference*, pp.475-478 (1997)
- [16]. Abdollahi, A., Fallah, F., Pedram, M.: A Robust Power Gating Structure and Power Mode Transition Strategy for MTCMOS Design. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 15(1), 80-89 (2007)
- [17]. Shi, K., Howard, D.: Sleep Transistor Design and Implementation - Simple Concepts Yet Challenges To Be Optimum. In : *International Symposium on VLSI Design, Automation and Test, Dallas*, pp.1-4 (2006)
- [18]. Anis, M., Mahmoud, M., Elmasry, M., Areibi, S.: Dynamic and leakage power reduction in MTCMOS circuits using an automated efficient gate clustering technique. In : *Design Automation Conference (DAC'02), New Orleans*, pp.480-485 (2002)
- [19]. Calhoun, B. H., Honore, F. A., Chandrakasan, A. P.: A leakage reduction methodology for distributed MTCMOS. *IEEE Journal of Solid-State Circuits* 39(5), 818-826 (2004)
- [20]. Agarwal, K., Nowka, K., Deogun, H., Sylvester, D.: Power Gating with Multiple Sleep Modes. In : *Proceedings of the 7th international Symposium on Quality Electronic Design, Washington DC*, pp.633-637 (2006)
- [21]. Hodayoun, H., Makhzan, M., Veidenbaum: Multiple sleep mode leakage control for cache peripheral circuits in embedded processors. In : *Proceedings of the 2008 international Conference on Compilers, Architectures and Synthesis For Embedded Systems CASES '08, Atlanta, GA*, pp.197-206 (2008)
- [22]. Paik, S., Shin, Y.: Multiobjective optimization of sleep vector for zigzag power-gated circuits in standard cell elements. In : *Design Automation*

Conference (DAC'08), pp.600-605 (2008)

- [23]. Kosonocky, S., Immediato, M., Cottrell, P., Hook, T., Mann, R., Brown, J.: Enhanced multi-threshold (MTCMOS) circuits using variable well bias. In : International Symposium on Low Power Electronics and Design (ISLPED '01), Huntington Beach, pp.165-169 (2001)
- [24]. Smith, J. E., Sohi, G. S.: The microarchitecture of superscalar processors., 1609-1624 (1995)
- [25]. Shen, J., Lipasti, M.: Modern Processor Design Fundamentals of Superscalar Processors Beta Edition edn. McGraw-Hill (2003)
- [26]. Brooks, D., Martonosi, M.: Dynamically exploiting narrow width operands to improve processor power and performance. In : High-Performance Computer Architecture, Orlando, pp.13-22 (1999)
- [27]. Ergin, O., Balkan, D., Ghose, K., Ponomarev, D.: Register Packing: Exploiting Narrow-Width Operands for Reducing Register File Pressure. In : International Symposium on Microarchitecture (MICRO'04), pp.304-315 (2004)
- [28]. Gonzalez, G., Cristal, A., Ortega, D., Veidenbaum, A., Valero, M.: A Content Aware Integer Register File Organization. In : International Symposium on Computer Architecture, München, p.314 (2004)
- [29]. Osmanlioglu, Y., Kocberber, Y., Ergin, O.: Reducing Parity Generation Latency through Input Value Aware Circuits. In : Great Lakes Symposium on VLSI (GLSVLSI'09), Boston, pp.109-112 (2009)
- [30]. Aggarwal, A., Franklin, M.: Energy efficient asymmetrically ported register files. In : International Conference on Computer Design, pp.2-7 (2003)
- [31]. Kocberber, Y., Osmanlioglu, Y., Ergin, O.: Exploiting narrow values for faster parity generation. *Microelectronics International* 26(3), 22-29 (2009)
- [32]. Folegnani, D., González, A.: Energy-effective issue logic. In : International Symposium on Computer Architecture (ISCA '01), pp.230-239 (2001)
- [33]. Manne, S., Klauser, A., Grunwald, D.: Pipeline gating: speculation control for energy reduction. In : International Symposium on Computer Architecture, Barcelona, pp.132-141 (1998)
- [34]. Balasubramonian, R., Dwarkadas, S., Albonesi, D.: Reducing the complexity of the register file in dynamic superscalar processors. In : International Symposium on Microarchitecture , Austin, pp.237-248 (2001)

- [35]. Azevedo, A., Issenin, I., Cornea, R., Gupta, R., Dutt, N., Veidenbaum, A., Nicolau, A.: Profile-based Dynamic Voltage Scheduling using Program Checkpoints in COPPER Framework. In : Design, Automation and Test in Europe Conference and Exhibition (DATE'02), p.168 (2002)
- [36]. Osmanlioğlu, Y., Hanay, Y., Oğuz, E.: Modifying the Data-Holding Components of the Microprocessors for Energy Efficiency. Journal of Circuits, Systems and Computers 18(6) (2009)
- [37]. Park, I., Powell, M. D., Vijaykumar, T. N.: Reducing register ports for higher speed and lower energy. In : International Symposium on Microarchitecture(MICRO-35), pp.171-182 (2002)
- [38]. Cruz, J.-L., Gonzalez, A., Valero, M., Topham, N. P.: Multiple-banked register file architectures. In : International Symposium on Computer Architecture, pp.316-325 (2000)
- [39]. Kondo, M., Nakamura, H.: A Small, Fast and Low-Power Register File by Bit-Partitioning. In : Proceedings of the 11th international Symposium on High-Performance Computer Architecture, Washington DC, pp.40-49 (2005)
- [40]. Lipasti, M. H., Mestan, B. R., Gunadi, E.: Physical register inlining. In : International Symposium Computer Architecture, pp.325-335 (2004)
- [41]. Loh, G. H.: Exploiting data-width locality to increase superscalar execution bandwidth. In : International Symposium on Microarchitecture (MICRO'02), pp.395-405 (2002)
- [42]. Henning, J. L.: SPEC CPU2000: measuring CPU performance in the New Millennium. Computer 33(7), 28-35 (2000)
- [43]. Yourst, M. T.: PTLsim: A Cycle Accurate Full System x86-64 Microarchitectural Simulator. In : International Symposium on Performance Analysis of Systems & Software, pp.23-34 (2007)



## ÖZGEÇMİŞ

### Kişisel Bilgiler

Soyadı, adı : KOÇBERBER, Yusuf Onur  
Uyruğu : T.C.  
Doğum tarihi ve yeri : 19.04.1987 Ankara  
Medeni hali : Bekar  
Telefon : 0 (312) 292 42 90  
e-mail : [yokocberber@etu.edu.tr](mailto:yokocberber@etu.edu.tr)

### Eğitim

| Derece | Eğitim Birimi                      | Mezuniyet tarihi |
|--------|------------------------------------|------------------|
| Lisans | TOBB-ETÜ /Elektrik Elektronik Müh. | 2004             |

### İş Deneyimi

| Yıl                  | Yer        | Görev        |
|----------------------|------------|--------------|
| Mayıs – Ağustos 2006 | TAI        | Ortak Eğitim |
| Ocak – Nisan 2007    | TAI        | Ortak Eğitim |
| Mayıs – Ağustos 2008 | UPA Makine | Ortak Eğitim |

### Yabancı Dil

İngilizce

### Yayınlar

Ergin, O., Kocberber, Y., Özsoy, M.: Telsiz Bilgisayar Mimarisi: Gömülü Sistemler Sempozyumu (GömSis'08), İstanbul, sayfa 2, (2008)

Osmanlioglu, Y., Kocberber, Y., Ergin, O.: Reducing Parity Generation Latency through Input Value Aware Circuits. : Great Lakes Symposium on VLSI (GLSVLSI'09), Boston, syf.109-112 (2009)

Kocberber, Y., Osmanlioglu, Y., Ergin, O.: Exploiting narrow values for faster parity generation. Microelectronics International 26(3), 22-29 (2009)

Kaynak, İC., Koçberber, Y., Ergin, O.: Reducing The Energy Dissipation Of The Issue Queue By Exploiting Narrow Immediate Operands. (gönderildi)