TOBB EKONOMİ VE TEKNOLOJİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

DÜŞÜK GÜÇ TÜKETİMİ VE YÜKSEK BAŞARIM İÇİN ÖZGÜN UYARLANABİLİR GÖMÜLÜ SİSTEM VE BELLEK TASARIMLARI

DOKTORA TEZİ

Fahrettin KOÇ

Bilgisayar Mühendisliği Anabilim Dalı

Tez Danışmanı: Prof. Dr. Oğuz ERGİN

AĞUSTOS 2022

TEZ BİLDİRİMİ

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, alıntı yapılan kaynaklara eksiksiz atıf yapıldığını, referansların tam olarak belirtildiğini ve ayrıca bu tezin TOBB ETÜ Fen Bilimleri Enstitüsü tez yazım kurallarına uygun olarak hazırlandığını bildiririm.

Fahrettin KOÇ

İMZA

ÖZET

Doktora Tezi

DÜŞÜK GÜÇ TÜKETİMİ VE YÜKSEK BAŞARIM İÇİN ÖZGÜN UYARLANABİLİR GÖMÜLÜ SİSTEM VE BELLEK TASARIMLARI

Fahrettin KOÇ

TOBB Ekonomi ve Teknoloji Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Tez Danışmanı: Prof. Dr. Oğuz ERGİN

Tarih: Ağustos 2022

Modern gömülü sistemler ve bilgisayarlarda düşük güç tüketimi sağlamak için bu sistemlerin en önemli parçası olan bellek yapılarında enerji kayıplarını azaltan çözümlere ihtiyaç vardır. Ancak bu çözümlerin başarımda istenen seviyeyi düşürmemesi ve hoş görülemez alan kaybına neden olmaması beklenir. Çağdaş bilgisayar mimarilerinde en çok kullanılan bellek yapılarından biri, *Dinamik Rasgele Erişimli Bellek* (DRAM)'lerdir. DRAM'i oluşturan bit hücreleri, belirli bir süre herhangi bir erişim olmaksızın veri saklayabilmekte ancak belirli süreden sonra erişim yapılmazsa sızdırma akımları nedeniyle veri kaybı olmaktadır, bu nedenle periyodik olarak DRAM hücrelerine erişilmesi ve yenilenmesi (Refresh) gerekmektedir. Bu işlem ise, hem güç tüketimi hem de başarım açısından oldukça maliyetlidir. Tez kapsamında, farklı koşullar/girdilere göre DRAM'in devre parametrelerini (besleme gerilimi veya alttaş kutuplama gerilimi) kendisinin değiştirilebildiği özgün Uyarlamalı DRAM (Adaptive DRAM) tasarımları (Geliştirdiğim üç tasarımdan ikisi; 2019/17243 ve 2019/13677 patent numarası ile tescillenmiştir, üçüncüsü; 2019/10444, tescil sürecindedir.) önerilmektedir. Önerilen tasarımların herhangi biri, DRAM'e kıyasla en az %21 daha düşük güç tüketimi sağlamaktadır, ve sadece %10'dan daha az gecikmeye neden olmaktadır. Ayrıca, özgün ADRAM tasarımlarımız, girdilere göre, ihtiyaç duyulan toplam yenileme sayısında %34 ile %81,8 aralığında düşüş sağlayabilmektedir.

Durağan Rasgele Erişimli Bellek (SRAM) diğer bir önemli bellek birimidir. SRAM için sızdırma akımları küçülen transistör boyutları (kanal genişliği, ısıl yükler vb.) nedeniyle büyüyen bir problemdir. Bu problemi çözmek için, birden fazla hücre içeriği uyarlamalı ve bu uyarlamayı birden fazla hücreye dağıtan *Multi-contents Aware SRAM* (MASRAM) tasarımı önerilmektedir. MASRAM, 64 bit gruplu hücre öbeği için en az %74 ihtimalle %35' e varan durağan enerji kaybı düşüşü sağlayabilmektedir (15. ve 47. bit'lere göre alttaş kutuplama gerilimi 64 hücreye uygulandığında), ve sadece %1'lik bir alan artışına neden olur.

Gömülü sistemlerden uç cihazlara, hava savunmadan yapay zeka uygulamalarına, *Alanda Programlanabilir Kapı Dizileri* (FPGA) kullanımı, yeniden programlanabilir yapısı nedeniyle yaygınlaşmaktadır, ve FPGA'lerde güç tüketiminin önemi de artmaktadır. Düşük güç tüketimi için önerilen çözümlerden biri, FPGA'lerde "gerilim düşürme"'dir.Ancak, bu yöntem güvenilirlik endişesi oluşturmamalı, ve istenen doğruluk seviyesini garanti etmelidir. Tez kapsamında, FPGA tabanlı *Evrişimsel Sinir Ağları* (CNNs) hızlandırıcılar için gerilim düşürmeye yönelik şu çalışmalar gerçekleştirilmiştir: İlk çalışmada; farklı FPGA'lerde, farklı frekanslarda, farklı CNN denektaşları için gerilim düşürme ile doğruluk ilişkisi araştırılır. İkinci çalışma, -40 ile 50 °C arasındaki her sıcaklıkta, 4 farklı nem koşulunda (ilk kez bir FPGA için), farklı gerilimlerde CNNs koşturularak; gerilim düşürmenin doğruluklara etkisinin farklı zorlu şartlar altında karakterizasyonu sağlanır. Ayrıca, FPGA tabanlı CNN hızlandırıcıların güç verimliliğinde; temel tasarıma kıyasla %65 artış sağlayan, 3 özgün güvenilir gerilim düşürme tasarımı önerilmiştir. Son çalışmada ise, ilk kez, şu 2 etki keşfedilmiştir: CNN hızlandırıcı FPGA'lerde belirli bir düşük voltajda artan sayıda CNN iterasyonu ile doğrulukların azalması (DIE), ve o voltajda yineleme devam ederken geçici olarak yüksek gerilim uygulamanın DIE'a karşı iyileştirici etkisi (RE). Bu etkileri kullanarak, istenen doğruluğu koruyarak en az %43 güç verimliliği artışı sağlayan 3 özgün FPGA gerilim düşürme tasarımı önerilmiştir.

Anahtar Kelimeler: Düşük güç tüketimli bellek, DRAM, SRAM, VLSI tasarım, Gerilim ölçekleme, Uyarlamalı alttaş kutuplama, Durağan enerji kaybı, Sızdırma azaltma, Saklama zamanı, FPGA, FPGA tabanlı gömülü sistemler, Donanım hızlandırıcı, Derin öğrenme, Evrişimsel sinir ağı, Gerilim düşürme.

ABSTRACT

Doctor of Philosophy

NOVEL ADAPTIVE EMBEDDED SYSTEM AND MEMORY DESIGNS FOR LOW POWER CONSUMPTION AND HIGH PERFORMANCE

Fahrettin KOÇ

TOBB University of Economics and Technology
Institute of Natural and Applied Sciences
Department of Computer Engineering

Supervisor: Prof. Dr. Oğuz ERGİN

Date: August 2022

To ensure low power consumption in modern embedded systems and computers, solutions that reduce energy dissipation are needed in Memory structures, which are the most critical part of these systems. However, these solutions are expected not to reduce the intended performance level and not cause an intolerable area cost. One of the most widely used memory structures in contemporary computer architectures is Dynamic Random Access Memory (DRAM). The bit cells that make up the DRAM can store data without access for a certain period. Still, if access is not made after a certain period of time, data is lost due to leakage currents, so it is necessary to periodically access and refresh. This process is very costly in terms of both power consumption and performance. In the scope of the thesis work, novel adaptive DRAM (Adaptive DRAM) designs (Two of the three techniques I developed; are registered with patent numbers 2019/17243 and 2019/13677, and the third one is in the registration process 2019/10444) in which DRAM can change its own circuit parameters (supply voltage or body biasing voltage) according to different conditions/inputs are proposed. Any of our proposed designs provide at least 21% lower power consumption than DRAM and only cause latency of less than 10%. In addition, our different ADRAM designs can achieve a 34% to 81.8% reduction in the total number of refreshes needed, depending on the inputs.

Static Random Access Memory (SRAM) is another important branch of memory. Leakage currents in an SRAM are a growing problem due to shrinking transistor sizes (channel width, thermal loads, etc.). To solve this problem, it is proposed to design Multi-contents Aware SRAM (MASRAM), which adapts multiple cell contents and distributes this to multiple cells. MASRAM can provide a static energy dissipation reduction of up to 35% with a probability of at least 74% for a grup of cells with 64 bits (when the body biasing voltage relative to the 15th and 47th bits is applied to 64 cells), and causes an area increase of only 1%.

From embedded systems to edge devices, from defense to AI applications, Field Programmable Gate Array (FPGAs) is spreading due to their reprogrammable structure, and the importance of power consumption in FPGAs is also growing. One recommended solution for low power consumption is "undervolting" in FPGAs. However, this method should not raise a reliability concern and should guarantee the intended levels of accuracy. In the scope of the thesis, the following studies were carried out for FPGA-based Convolutional Neural Networks (CNNs) accelerators: In the first study, we inspect the undervolting accuracy relationship for CNN benchmarks on different FPGAs at different frequencies. The second study is on characterizing the effect of undervolting on accuracies at different voltages under four different humidity conditions (for the first time for an FPGA), at any temperature between -40°and 50 °C, under different harsh conditions. Moreover, we propose three novel reliable voltage reduction designs proposed for FPGA based CNN accelerators that provide a 65% increase in power efficiency compared to the baseline design. In the final study, for the first time, we discover the two effects: an increasing number of CNN iterations at a low voltage decreases the accuracy (DIE), and the rejuvenating effect against DIE by temporarily applying high voltage while iteration continues at that voltage (RE). Exploiting these effects, we proposed three novel FPGA undervolting designs providing at least a 43% power efficiency increase while preserving the desired accuracy.

Keywords: Low-power memory, DRAM, SRAM, VLSI design, Voltage scaling, Adaptive body biasing, Static energy dissipation, Leakage reduction, Retention time, FPGA, FPGA-based embedded systems, Hardware accelerator, Deep learning, Convolutional neural network, Undervolting.

TEŞEKKÜR

Çalışmalarım boyunca değerli yardım ve katkılarıyla beni yönlendiren Prof.Dr. Oğuz ERGİN'e, kıymetli tecrübelerinden faydalandığım TOBB Ekonomi ve Teknoloji Üniversitesi Bilgisayar Mühendisliği ve Elektrik Elektronik Bölümü öğretim üyelerine, Makina, Fizik ve diğer bölümlerimizden çok değerli hocalarıma, Bölüm, Enstitü ve Üniversite idari personeline, patentlerim süresince destek sağlayan personele, Z10 zamanından beri diğer bir yuvam haline gelen Kasırga Labına ve Kasırga ailesi arkadaşlarıma, Barcelona Supercomputing Center'da birlikte çalıştığım Adrian C. Kestelmen ve Osman Ünsal hocalarıma, çalıştığım kuruma, iş arkadaşlarıma, ve özellikle Bozok, Göktuğ ve Siper projesinde gece gündüz demeden omuz verdiğimiz destekleriyle her zaman yanımda olan arkadaşlarıma, yöneticilerime, ülkeye faydalı birçok kişinin yetiştirilmesine ve birçok faydalı işin yapılmasına vesile olan rahmetli babama, her zaman hep yanımda olan anneme ve aileme, saydığım ve sayamadığım üzerimde emeği ve desteği olan herkese, birlikte çalışma desteği sağlayan Hipeac'e, araştırma desteği sağlayan Barcelona Supercomputing Center'a ve sunduğu tüm imkanlar için TOBB Ekonomi ve Teknoloji Üniversiteme teşekkürü bir borç bilirim.

İÇİNDEKİLER

ŞEKİL LİSTESİ

KISALTMALAR

AADRAM : Access Aware Dynamic Random Access Memory
ADRAM : Adaptive Dynamic Random Access Memory
ASIC : Application Specific Integrated Circuit
CNN : Convolutional Neural Network
CPU : Central Processing Unit
CRU : Contrained Rejuvenating Undervolting
DDR : Double Data Rate
DIE : Destructive Iteration Effect
DPU : Deep Learning Processing Unit
DRAM : Dynamic Random Access Memory
FPGA : Field Programmable Gate Array
GPU : Graphical Processor Unit
IP : Intellectual Property
IU : Iteration-aware Undervolting
MCSRAM : Multi Content-aware Static Random Access Memory
MUX : Multiplexer/Çoklayıcı
NMOS : N-type metal-oxide-semiconductor
ORU : Optimal Rejuvenating Undervolting
PADRAM : Process-variation Aware Dynamic Random Access Memory
PMOS : p-type metal-oxide-semiconductor
PL : Programming Logic
PS : Processing System
RE : Rejuvenating Effect
SRAM : Static Random Access Memory
TADRAM : Temperature Aware Dynamic Random Access Memory
VLSI : Very Large Scale Integrated-circuits

SEMBOL LİSTESİ

Bu çalışmada kullanılmış olan simgeler açıklamaları ile birlikte aşağıda sunulmuştur.

Simgeler	**Açıklama**
Vrej	Gençleştirici/Rejuvenating Voltaj
Vcrv	Kritik Bölge Voltajı
Vmin	Guardband bölgesindeki en düşük voltaj gerilimi
NoI	İterasyon Sayısı
VCCINT	Internal Voltage
Vth	Eşik Değer Voltajı
Vbram	BRAM Voltaj (hattı)
α	Alpha
T_N	N. sıcaklık limiti
P	Power
V	Gerilim/Voltaj
I	Akım
W	Watt
Gops	Giga Operations
Vdd	Besleme Gerilimi
Vbias	Alttaş Kutuplama Gerilimi
δ	Tellerde oluşan küçük fark

1. GİRİŞ

1.1 Tezin Amacı

Savunma sanayi için geliştirilmiş bir askeri gömülü sistem, nesnelerin interneti olarak tasarlanan bir cihaz, yapay zeka uygulamalarına sahip bir drone veya tablet gibi son kullanıcıya yönelik taşınabilir ürünler... Günümüz elektroniğinin önemli bir kısmı, doğrudan kendisi veya dahil olduğu platform üzerinden, kısıtlı bir güç kaynağına ihtiyaç duymaktadır. Dolayısıyla da güç tüketimi devre ve sistem tasarımı için en önemli kısıt ve problem olmaktadır. Güç tüketiminin azaltılması batarya kritik sistemlerin bağımsız taşınabilirlik kabiliyetlerinin artması anlamına gelir. Ayrıca, enerji kayıplarının azaltılması batarya kritik olmayan bir sistem için de önemlidir, çünkü yüksek enerji kayıpları güvenilirliğin (reliability) azalmasına da neden olabilmektedir, ve enerji kayıplarıyla birlikte artan soğutma ihtiyacı, sık erişim ihtiyacı gibi problemler başarımı da etkilemektedir [1–3]. Tezin amacı, istenen başarımdan ödün vermeden, hoş görülemez alan maliyetine sebep olmadan, güç tüketimini azaltacak doktora çalışmaları kapsamında geliştirilmiş özgün uyarlamalı donanım yaklaşımı ve bu yaklaşıma dayalı çözümleri sunmaktır.

Belirli bir amaç; örneğin bir bit hücresinde "0" tutulurken güç tüketimini azaltmak, için eniyilenmiş tasarımlar, farklı durumlarda (örneğin mantık "0" saklarkenki durum için) aynı etkiyi (güç tüketimi azalışı) oluşturmadığı veya hatta daha kötü etki (gecikme) oluşturduğu için sunduğu iyileştirme kısıtlı kalabilmekte, ve hatta farklı uygulamalar için denendiğinde toplamda daha kötü etkiye neden olabilmektedir [1, 4, 5]. Bu yüzden doktora çalışmaları kapsamında, günümüz elektronik tasarımlarında en sık kullanılan mimari yapılar üzerinden gidilerek (bu yapılar için anaçizgi tasarımlar gerçeklenmiş ve üzerinde tasarım değişiklikleri ve analizler gerçekleştirilmiştir) farklı durum setlerinde (ortam sıcaklığı, uygulama ve donanım karakterizasyonu, üretim kaynaklı farklılıklar, saklanan veri vb.) ilgili yapının kendi devre seviyesi parametrelerini durum girdilerine bağlı olarak kendi kendine değiştirebildiği özgün akıllı tasarımlar geliştirilmiştir. Bu yapılar için kullanılan "Uyarlamalı Donanım Tasarımı (Adaptive Hardware Design)" yaklaşımımızın diğer donanım (farklı FPGA tipleri, farklı bellek çeşitleri, veya GPU vb. farklı donanımlar) ve uygulamalarda (diğer öğrenme ve yapay zeka uygulamaları, veya herhangi başka bir yazılım) için de uygulanabileceği değerlendirilmektedir, ve tez kapsamında bu yüzden (tezin temel amaçlarından biri, Uyarlamalı Donanım Tasarımı yaklaşımını tanıtmak ve yaygınlaşmasını sağlamaktır.) farklı tekniğin bilinen en iyi durumundaki (state-of-the-art) mimari yapılar üzerine uyarlamalı donanım tasarımı

yaklaşımı uygulanmıştır.

Tez kapsamında, sayısal elektronikte en yaygın kullanılan DRAM (Devingen Rasgele Erişimli Bellek) ve SRAM (Durağan Rasgele Erişimli Bellek)'ler için, ve gömülü sistemler ve yapay zeka uygulamalarının gerçeklenmesi/hızlandırılması amacıyla sıkça tercih edilen (hem programlanabilir yapısı hem de yazılımsal ve donanımsal işlemleme kabiliyeti nedeniyle) FPGA (Alan Programlanabilir Kapı Dizileri) yapıları için düşük güç tüketimi sağlayan uyarlanabilir özgün tasarımlar geliştirilmiştir. Bu üç mimari yapının her birine yönelik geliştirilen özgün tasarımlar, farklı bölümlerde detaylı olarak ele alınmaktadır. İlgili tasarımın dayanağı olan gözlemler ve motivasyon kaynağı olan mevcut durumdaki problemlerin aktarımı, tasarımın çalışma mantığı ve işleyişi, ve o tasarıma ait yoğun deney veya test sonuçları, tezde ayrı ayrı, tasarımın anlatıldığı bölümün içerisinde sunulmaktadır.

1.2 Motivasyon

1.2.1 DRAM

Bit hücreleri istemsiz sızdırma akımları (Leakage Currents) nedeniyle durağan durumda enerji kaybetmektedir. Üstelik DRAM bit hücreleri, sürekli güç beslemesi olmadığı için belirli bir süre ardından bu sızdırma akımları nedeniyle veri kaybı yaşanmaktadır [1, 2, 6]. Bunu önlemek için de belirli periyotta (saklama zamanı) bir bit hücresine erişilmesi (yenileme, refreshing) gerekir. Ancak yenileme işlemi yapılırken okuma ve yazma yapılamaz. Dolayısıyla erişim işlemleri hem doğrudan güç tüketimi açısından, hem de erişim sırasında yazma ve okuma işlemleri yapılamadığı için başarım açısından oldukça maliyetlidir. Diğer taraftan, tüm bit hücreleri açısından sızdırma aynı oranda gerçekleşmez, özellikle üretim süreçlerindeki farklılık (process variation) nedeniyle farklı bit hücrelerinin saklama zamanları da aynı olmamaktadır. Son olarak, sızdırma akımları elektronların mobilitesini artırdığı için sıcaklıkla da artış göstermektedir. Bu yüzden de belirli sıcaklığın üstüne çıkılması durumunda DRAM bit hücrelerinin daha sık yenilenmesi gerekmektedir. İşte bu nedenlerle, tüm bu farklı durum ve koşulları sağlayacak şekilde, DRAM üreticileri standartlar gereği (Örn: JEDEC DDR3 veya DDR4) tüm hücreler için ortak bir yenileme sıklığı belirlemektedir [1, 7, 8].

DRAM üzerine gerçekleştirdiğim çalışmalar ve önerdiğim uyarlamalı DRAM (ADRAM) tasarımları için takip eden şu temel unsurlar motivasyon kaynağı olmuştur:

- Mevcut DRAM'ler için belirlenen erişim sıklığı bit hücrelerinin büyük bir çoğunluğu için gereksizdir, gereksiz erişim maliyetine neden olmaktadır.

- Bilinen kadarıyla DRAM'lerde sızdırma akımlarının azaltılmasını sağlamak için mimari seviye çözümleri de kullanan uyarlamalı bir devre yoktur.
- DRAM bit hücrelerinin olduğu satırlar en düşük saklama zamanına sahip bit hücresine göre profillenebilmektedir. Bunu uygulayarak yenileme işlemlerinin sayısı azaltılmaya çalışılmaktadır, ancak sorunun asıl sebebi olan sızdırma akımlarının azaltılmasına veya güç tüketiminin düşürülmesine yönelik devre seviyesi çözümlerle birlikte denenmemektedir.
- Devre parametrelerinin (alttaş kutuplama gerilimi veya besleme gerilimi) sıcaklığa uyarlamalı değiştirilmesini sağlayan mekanizmalar, bilindiği kadarıyla, daha önce önerilmemiştir.
- DRAM hücrelerine yönelik yazma ve okuma örüntüleri incelendiğinde, sıklıkla erişilen bir satıra daha sonra erişilmesi ihtimali daha yüksektir.

Özetle, mevcut DRAM'lerde tüm hücreler için gerekli olmayan, güç tüketimi ve başarım açısından külfetli erişimler (yenileme için) yapılmaktadır. Ancak; eğer uyarlamalı olarak, saklama zamanı düşük hücreler için sızdırma akımları azaltılabilirse, veya eğer yüksek sıcaklıklarda artan sızdırma akımları nedeniyle yenileme sıklığını artırmak yerine uyarlamalı olarak sızdırma akımları azaltılabilirse, veya eğer erişim örüntüsüne uyarlamalı olarak sızdırma akımları azaltılabilirse, bahsedilen gereksiz erişim sayısı etkin şekilde azaltılmış olur. İşte tez kapsamındaki DRAM'lere yönelik çalışma ve önerilen tüm özgün tasarımların hedefi; bu koşulları sağlayabilen uyarlamalı ve akıllı mekanizmalar kullanarak, hem güç tüketimini düşürmek hem de başarımı artırmaktır.

1.2.2 SRAM

Transistör boyutları küçüldükçe sızdırma akımları daha da artmaktadır [6]. Durağan enerji kayıpları devingen güç tüketimi kadar önem kazanmaktadır. SRAM bit hücreleri için anahtarlama olmadığı durumda bile istenmeyen sızdırma akımları nedeniyle güç tüketimi yaşanmaktadır. Bunun için, fabrika seviyesinde mantık 1 saklama durumunda daha az sızdıracak (veya tersi) asimetrik ve sabit çözümler bulunmaktadır. Ancak bu çözümler durum değişikliklerinde tam tersi etkiye neden olmaktadır: Toplamda güç tüketimine sağladıkları iyileştirici etki azalmakta veya daha fazla güç tüketimine neden olabilmektedirler, ve aynı zamanda gecikmelere yol açabilmektedirler. İşte bu yüzden bir SRAM bit hücresinin içinde tuttuğu veri mantık 0 veya 1 iken veriye bağlı olarak alttaş kutuplama yapılması ve uyarlamalı alttaş kutuplaması ile Vth kontrolü gibi yöntemler daha önce önerilmiştir [2, 4, 5, 9–13]. Ancak doğrudan SRAM hücresine

uygulanan bu tip çözümler hücre başına yüksek alan maliyetine neden olmaktadır. Diğer taraftan çok fazla hücre için bir hücrenin verisine göre uyarlama yapıldığında ise sağlanan enerji kayıplarındaki kazanç üzerinde düşüş görülmektedir [5, 14].

Özetle, eğer bit hücrelerindeki sızdırma akımlarını tek bir girdi yerine birden fazla girdiye göre azaltacak bir tasarım bulunabilirse ve bu tasarım; girdilerin karakterizasyonu kullanılarak, diğer hücreleri en çok yansıtacak şekilde gerçeklenebilirse, hoş görülebilir alan maliyetiyle düşük güç tüketimi açısında önemli kazançlar sağlanmış olur. İşte tez kapsamındaki çalışma ve önerilen tasarımların diğer bir hedefi bu kazanımı sağlayacak akıllı tasarımı ortaya çıkarmaktır.

1.2.3 CNN hızlandırıcı olarak FPGA

FPGA üreticileri nominal besleme gerilimlerini güvenilir tarafta kalarak belirlemektedirler. Gömülü sistemler ve yapay zeka uygulamaları açısından yeniden programlanabilir donanım kabiliyeti nedeniyle (hem yazılımsal hem donanımsal işlemleme kabiliyeti de sunmasıyla da) FPGA'ler yaygın kullanımdadırlar. Bu yüzden de, FPGA'lerin güç tüketimlerinin azaltılması gerekmektedir. Bu amaçla gerilim düşürme (undervolting) yöntemi en bilinen yöntemler arasındadır [15–17]. Yapay zeka ve görüntü işleme uygulamalarının temel araçlarından biri evrişimsel sinir ağlarıdır (CNNs) [18–23], CNN algoritmaları için donanım hızlandırıcı olarak kullanılan bir FPGA'de gerilim düşürmenin doğruluklar açısından etkisi karakterize edilmelidir. Tez kapsamındaki çalışmalardan biri, bu karakterizasyon hedefi doğrultusunda gerçekleştirilmiştir [17, 24].

FPGA'ler savunma sanayii uygulamalarından sürücüsüz araçlara kadar çok farklı alanlarda kullanılmaktadır, bu durum ise FPGA'ler açısında çok çeşitli ve zorlayıcı ortam koşullarında çalışabilme gereksinimi ortaya çıkarmaktadır. Tez kapsamındaki çalışmalardan bir diğeri de, bu zorlayıcı ortam koşullarında gerilim düşürme yöntemine güvenilir mi sorusuna cevap aramak için gerçekleştirilmiştir.

CNN hızlandırıcı olarak FPGA'lerde gerilim düşürme çalışmalarım sırasında, bilindiği kadarıyla literatürde ilk kez; uygulanan bazı gerilimlerde uygulama iteratif çalıştırılmaya devam ederken, doğrulukların ilk iterasyona göre azalmaya başladığı gözlemlenmiştir. İterasyonun bu yıkıcı etkisinin karakterizasyonu gerilim düşürme tasarımları açısından oldukça önemlidir, çünkü bunun dikkate alınmadığı önceki tasarımlarda güvenli (doğruluktan ödün vermeyen) olarak belirlenen minimum gerilim düşürme geriliminin aslında güvenli olmadığını gözlemliyoruz. Ayrıca, yine literatürde ilk kez; uygulama iteratif çalıştırılmaya devam ederken, yıkıcı etkinin görüldüğü herhangi bir gerilimle birlikte daha yüksek bir gerilim seviyesi uygulanacak olursa

bunun doğruluk düşüşünü iyileştirebildiğini gözlemledik. Tez kapsamında bu iki gözlem ve bu gözlemlere ait kapsamlı deneysel çalışmalar gerçekleştirilmiştir. Ayrıca bu gözlemlerden faydalanarak, güvenilir (reliable) özgün gerilim düşürme (undervolting) tasarımları geliştirilmiştir.

1.3 Katkı ve Hedefler

Tezin takip eden bölümlerinde bahsedilecek olan, doktora çalışmaları kapsamında gerçekleştirilmiş kapsamlı benzetim, analiz ve deneylere dayalı temel gözlemler, bu gözlemler kullanılarak geliştirilen özgün tasarımlar ve akademik çıktı şu şekilde özetlenebilir:

- En çok tercih edilen yapay zeka ve görüntü işleme uygulamalarının temelini oluşturan CNN'ler için hızlandırıcı olarak bir FPGA gerçeklenmiş, gerçek zamanlı güç ve doğruluk sonuçlarının toplanabildiği ve farklı fikirlerin denenebileceği ortam kurulmuştur. Tezde, temel güç verimliliği teknikleri ve özellikle gerilim düşürme tanıtılmış, bunlar üzerine güvenilir tasarımlar önerilmiştir. Doktora çalışmaları kapsamında FPGA'lere yönelik bulunan fikir ve çözümler, uluslararası saygın kurumlardan olan HIPEAC (European Network on High-performance Embedded Architecture and Compilation) tarafından desteğe layık görülmüş ve "collaboration grant" kazanılmıştır, ayrıca birlikte çalışılan Barcelona Supercomputing Center (BSC) tarafından da "visiting scholar" desteğine layık görülmüştür. Tekniğin bilinen durumunun tanıtılması açısından da tezin içeriği önemlidir, ayrıca geliştirilen özgün farklı tasarımlar da tez kapsamında anlatılmaktadır. Bu tasarımların bir kısmından ise akademik çıktılar (bildiri, makale, patent vb.) elde edilmiştir, elde edilmektedir, tez sonrasına da bu akademik çıktıların giderek daha da artması hedeflenmektedir.

- Bilinen kadarıyla literatürde daha önce olmayan şu iki gözlem ortaya konmuştur: Belirli gerilim seviyelerinde, bir düşük gerilimde CNN uygulaması ilk kez çalıştırıldığında elde edilen doğruluk, aynı uygulama çalıştırılmaya devam ettikçe düşmektedir. Buna iterasyonun yıkıcı etkisi (DIE) adı verilmiştir. İkinci gözlem ise; DIE görülen bir gerilim seviyesinde uygulama iteratif çalıştırılmaya devam ederken araya yüksek gerilim uygulanırsa bu durumda doğruluklardaki düşüşte azalma yaşanmaktadır, buna da "iyileştirici etki" (RE) (veya rejuvanating etkisi, veya gençleştirici etki) adı verilmiştir. Yoğun ve kapsamlı deneylerle, bu etkilerin karakterizasyonu sağlanmıştır. Ayrıca, bu etkilere dayalı 3 özgün gerilim düşürme yöntemi geliştirilmiştir. Bu tasarımlar sayesinde istenen doğruluklardan ödün vermeden en az %43

seviyesinde güç verimliliği artışı sağlanmıştır (Bu çalışma kapsamında bir bildiri gönderilmiştir.).

- Zorlayıcı çevre koşullarında, FPGA tabanlı CNN hızlandırıcılar için gerilim düşürme yönteminin güvenilirliği çalışılmış ve farklı koşullarda doğruluk ve gerilim seviyesi ilişkisi karakterizasyonu sağlanmıştır. FPGA tabanlı CNN uygulamaları için, ilk kez bir çalışmada bu kadar geniş sıcaklık aralığında deneyler gerçekleştirilmiştir, gerçek zamanlı güç verileri CNN denek taşları koşarken toplanmıştır, ve nem etkisi analiz edilmiştir. Ayrıca, deney sonuçlarına dayalı 3 özgün gerilim düşürme tasarımı önerilmiştir, bu tasarımlar sayesinde güç verimliliğinde en az %65 kazanç elde edilebilmiştir (Bu çalışma kapsamında 1 dergi makalesi yayınlanmıştır.). CNN hızlandırı olarak kullanılan FPGA'ler için gerilim düşürme yöntemi uygulanmış ve bu sayade güç verimliliğinin 3 kata kadar artırılabildiği görülmüştür (1 bildiri olarak sunulmuştur).

- Çoklu içerik uyarlamalı ve bu uyarlamanın çoklu hücreye uygulandığı özgün MCSRAM (Multi Content Aware SRAM) tasarımı geliştirilmiştir. Bu tasarım sayesinde, alan maliyeti ihmal edilebilir seviyeye kadar düşürülmüştür, ve %74 ihtimalle 64 bit hücre için %35'e (mimari benzetimlere göre) kadar durağan enerji tüketiminde düşüş sağlanmıştır (1 bildiri olarak sunulmuştur, kapsamlı 1 bildiri hazırlanmaktadır).

- Gömülü sistemlerden mobil cihazlara, FPGA'lerden bilgisayarlara tüm elektronik sistemlerde en çok yer alan bellek birimleri olan DRAM'ler dünyada sayılı üretici tarafından üretilebilmektedir. SRAM'e kıyasla DRAM tasarımının yapısı ve özellikle parametreleri ticari sır (Örneğin, açık kaynaktan ulaşılabilen model kütüphaneleri parametre değişimine ve tasarımı göstermeyecek şekilde şifrelenmiştir.) olarak tutulmaktadır, ve özellikle yenileme zamanı gibi bazı parametreler standartlara uygun olması beklenmektedir. Doktora çalışmaları kapsamında DRAM için standartlara uygun zamanlama ve başarımda olduğu benzetimlerle gösterilen bir DRAM (DDR3) devresi ayağa kaldırılmış ve temel tasarım olarak bu devre kullanılmıştır. Ayrıca, en yaygın kullanılan bellek yapılarından olan DRAM'ler için tekniğin bilinen durumunun, çalışma mekanizması ve temel sorunları da bu tez kapsamında sunulmaktadır. Son olarak, DRAM'ler için birçok fikir ve özgün tasarım geliştirilmiştir, geliştirilen özgün farklı tasarımlar da tez kapsamında anlatılmaktadır. Bu tasarımların bir kısmından ise akademik çıktılar (bildiri, patent vb.) elde edilmiştir, elde edilmektedir, tez sonrasına da bu akademik çıktıların giderek daha da artması hedeflenmektedir.

- Sıcaklık uyarlamalı, erişim örüntüsü uyarlamalı ve üretim işlem farklılığı

uyarlamalı 3 özgün adaptif DRAM tasarımı (ve her biri için 2 farklı gerçekleme yöntemi ile) geliştirilmiştir. Bu tasarımlar sayesinde toplam yenileme sayesinde %34 ile %81.8 arasında azaltma sağlanabilmiş, ve durağan enerji kayıplarında %55'e kadarlık bir düşüş ve toplam güç tüketiminde en az %21'lik düşüş elde edilebilmektedir. Ayrıca, sıcaklık uyarlamalı ADRAM tasarımı, sıcaklık limitinin üzerine çıkıldığı durumda yenileme sıklığının 2 katına çıkmasını önleyebilmiş, bu sayede bu sıcaklık üzerinde mevcut yenileme sayısı kadar kazanç sağlanmış olunmaktadır (DRAM üzerine gerçekleştirilen tasarımlar için patent başvuruları gerçekleştirilmiştir: 2 patent tescil almış, üçüncü patent için tescil süreci devam etmektedir, ve farklı bildiriler hazırlanmıştır: 1 bildiri olarak sunulmuştur, 2 bildiri/makale ve 1 patent daha hedeflenmektedir.). Geliştirilen uyarlamalı tasarımlar ayrıca uyarlamalı donanım tasarımı yaklaşımına da temel oluşturmaktadır, farklı donanım ve uygulamalar için veya donanım alt seviyesinde farklı parametreler için de geliştirilen uyarlamalı karar mekanizmaları önerilmektedir ve böylece donanımların üretim sonrasında farklı koşul ve durumlarda kendi devre/donanım parametrelerini kendi kendine değiştirebildiği bir yaklaşımının yaygınlaşması hedeflenmektedir.

1.4 Organizasyon

Tez için bölümlendirme planı şu şekilde kurgulanmıştır: Bölüm 2, gerçekleştirilen çalışma, gözlem ve önerilen tasarımlarda kullanılan özet temel bilgiler ve en bilinen tekniklerden ve çözümlerin geliştirildiği/denendiği mimari yapılardan bahsetmektedir. Daha sonra sırayla uyarlamalı donanım tasarımı yaklaşımının uygulandığı farklı mimari yapılar için gerçekleştirilen çalışmalar ayrı bölümlerde ele alınmaktadır. Uyarlamalı DRAM tasarımları, benzetim ve analiz sonuçları Bölüm 3'de, Uyarlamalı SRAM tasarımları, benzetim ve analiz sonuçları Bölüm 4'de, FPGA bazlı CNN hızlandırıcılar için uyarlamalı gerilim düşürme, ve bu gerilim düşürmenin güvenilir olup olmadığına yönelik sonuç ve değerlendirmeler de Bölüm 5 içinde anlatılmaktadır. İterasyonun yıkıcı etkisi ve yüksek gerilim seviyelerinin iyileştirici etkisi gözlemleri ve bunları kullanarak geliştirilen güvenilir gerilim düşürme tasarımları ise Bölüm 6 ile sunulmaktadır. Bu bölümlerde gerektiğinde ayrıca, bahsedilen çalışmalar için izlenen metodoloji ve geliştirilen tasarımlarla ilgili literatür çalışmaları da her bölümün kendi içinde aktarılmaktadır. Son olarak, tüm çalışmalardaki sonuçlar ve adreslenen gelecek çalışmalar Bölüm 7 ile paylaşılmaktadır.

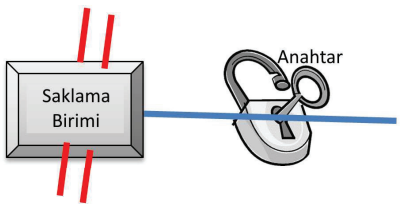
2. TEMEL BİLGİLER VE ANAÇİZGİ

2.1 DRAM Bit Hücesi ve Bellek Mimarisi

Bellek hücresi (bit hücresi, bitcell, cell) çok temel olarak, yük tutmak üzere kullanılan bir saklama birimi ve yüke erişim için kullanılmak üzere bir veya birden fazla erişim anahtarından (transistör) oluşmaktadır. Temsili gösterim Şekil 2.1'de yer almaktadır. Birçok bit hücresi biraraya gelerek yoğun bellek dizi (array)'leri oluşur. Bu bellek dizileri içerisinde okuma ve yazma işlemleri çoklu bit hücresine aynı anda gerçekleştirilir. Bit hücreleri içinde saklanan veriye (mantık "0", mantık "1") veya yüke erişim için anahtarlamayı kelime seç teli (word select line) gerçekleştirmekte ve erişim sağlandıktan sonra saklanan yükün aktarılması (okuma) veya yük aktarmak (yazma) için ise bit telleri (bitline) kullanılmaktadır [1].

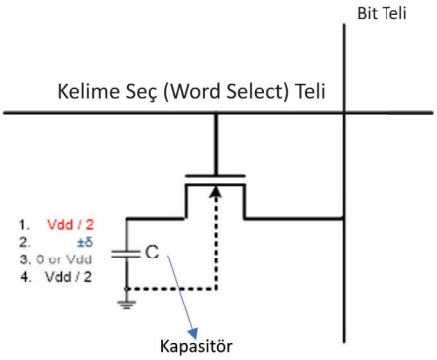
Çağdaş Bilgisayarlarda (ana bellek olarak), gömülü sistemlerde (embedded systems), mobil cihazlarda vb. birçok elektronik tasarımda ve sistemde en yaygın kullanılan bellek yapısı devingen rasgele erişimli bellek (DRAM, Dynamic Random Access Memory)'lerdir (Tez içerisinde DRAM olarak kullanılacaktır). DRAM mimarisinin en temel bileşeni bit hücreleridir ("DRAM bitcells" yada "cells"). DRAM'lerin bu kadar yaygın kullanılmasının sebeplerinden biri, birim alana daha yoğun bit hücresi yerleşimi yapılabilmesidir. Ayrıca, serim maliyeti diğer bellek birimlerine kıyasla (örnek olarak, SRAM bit hücrelerine göre) daha ucuzdur [1, 2].

DRAM bit hücreleri için saklama birimi yaygın olarak bir kapasitördür. Ancak, farklı kapasitör çeşitleri ve serim yöntemleri de bulunmaktadır, ayrıca transistörler ile de geçerlendiği uygulamalar bilinmektedir. Bir DRAM bit hücresi'nin yalın bir anlatımla çalışma mantığı şu şekildedir: DRAM bit hücresinde yazma yapılacağında erişim transistörü kelime seç (word select) teli üzerinden açılır, bit teli üzerinden yük kapasitöre doldurulur ve böylece bir dahaki erişime katar yük bu kapasitörde saklanır.



Şekil 2.1 : Bellek bit hücresi temsili gösterimi.

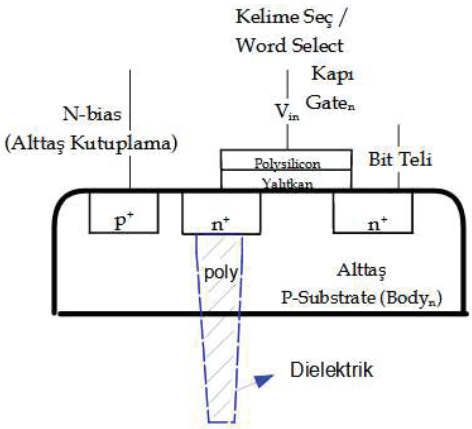
Aynı şekilde okuma yapılacağında ise; yine bu kapasitördeki yük erişim transisörü açıldığında bit teli üzerine akar ve sonra ilgili çevre birimler ve işlem adımları sayesinde okuma yapılır. DRAM hücrelerinde kapasitörler (veya kapasite oluşturacak birimler) sayesinde sürekli güç beslemesi yapmak gerekmez. Ancak devre elemanları sızdırdıkları (leakage) bir süre sonra kapasitör üzerinde tutulan veri kaybolabilir ve bu yüzden tutulan veriye göre kapasitörün belirli bir zamanda bir doldurulması (refresh) gerekir [1]. Kapasitör ve erişim transistörü ile birlikte hem kelime seç teli (word select line) hem de bit teli'nin olduğu DRAM bit hücresine ait temsili gösterim Şekil 2.2'de yer almaktadır.



Şekil 2.2 : DRAM bit hücresi.

Güncel endüstriyel uygulamalarda kapasitörü veya kapasitör yığınını transistörün terminali içerisinde bit telleri üzerine katmanlı yapıda (Burried Digitline) veya bit teli altına veya derin trench kapasitör olarak layout içerisine gömme şeklinde farklı tekniklerle kapasitör üretimleri gerçekleştirilmektedir. Bit teli altında kapasitör serimi ile bit teli üzerinde kapasitör serimi arasında, kapasitör altında bit teli silikon alana daha yakın olmasından dolayı erişim kolaylığı avantajı bulunmaktadır. Fakat, 2 yöntemde de transistör terminali üzerinde farklı ve büyük bir katman ve alan kullanılarak kapasitör oluşturulmaktadır. Bu katmanı oluşturmak yerine transistörde alttaş/substrate/body üzerine oyuk açarak kapasitörü burada oluşturmak ise alan açısından avantaj sağlamakta ve ek katman oluşturmaktan kurtarmaktadır. Ayrıca, kapasitör dolayısıyla bit telleri arası kontak erişimi zorlaştığı için katmanlı yapılardan daha erişimi kolaydır. İşte trench kapasitör olarak adlandırılan bu yöntem mevcut endüstriyel uygulamalarda sıkça kullanılmaktadır. Trench kapasitörle oluşturulan bir bit hücresini yansıtan temsili kesit görüntüsü Şekil 2.3 ile sunulmaktadır. DRAM hücresine erişimi sağlayan transistörün kapı terminali kelime seç (word select) hattına bağlıdır, kapasitördeki verinin akacağı veya kapasitöre verinin yükleneceği tel ise bit teli (bitline) hattına bağlıdır.

Kapasitörün verimliliği alttaş/subtrate/body alanın katkılaması ile kontrol



Şekil 2.3 : DRAM bit hücresi temsili kesit gösterimi.

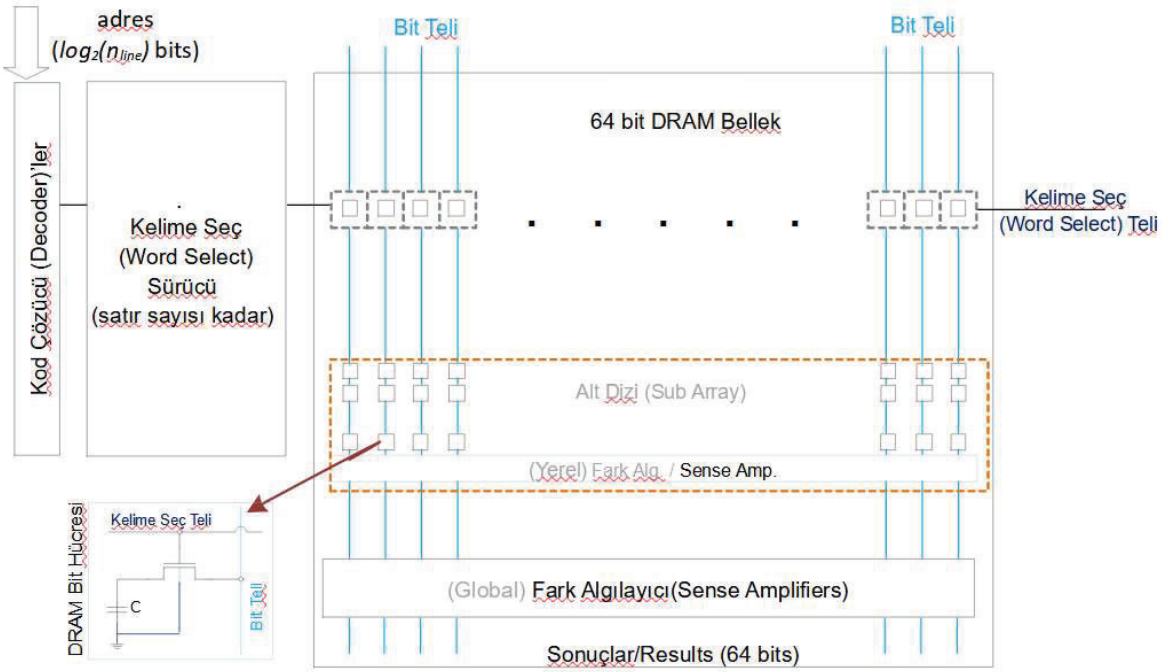
edilebilmekte ve bu yolla sızdırma akımları azaltılmaya çalışılmaktadır. Trench kapasitörün dezavantajı ise alttaşa açılan derin oyukların üretimde derinliğinin tam ayarlanaması kaynaklı üretim farklılıkları ve zorluklarıdır. Hem kapasitör, hem transistör, hem de bellek yapılarının üretimlerinde farklı yöntemler de uygulanmaktadır, ayrıca sürekli yeni yöntemler denenmekte ve gelişmeler olmaktadır. Örneğin; DRAM hücrelerinde alan maliyetini azaltmak için üç boyutlu serim istifleme (3D die stacking) gibi yöntemler de önerilmektedir [25]. Tez kapsamında önerilen çözüm ve tasarımlar DRAM ve SRAM için üretim tekniklerinden bağımsız her durumda uyarlanabilecek şekilde tasarlanmıştır. Eğer bir DRAM veya SRAM bit hücresinde sızdırma varsa (herşey sızdırır), erişim için transistör kullanılıyorsa, üretim yöntemi farketmeksizin bu çalışmada önerilmekte olan bu çözümler katkılarına devam eder.

Bir DRAM mimarisi gereksinim duyulan veri boyutuna bağlı olarak tasarlanmaktadır, ve buna göre bit hücresi sayısı belirlenir. Çok sayıda DRAM bit hücresinin biraraya gelmesiyle oluşan mimari yapı DRAM dizisi (DRAM array) olarak adlandırılır. DRAM bit hücreleri bir dizi (a DRAM array) halinde çevre devrelerle birlikte DRAM küme (a DRAM bank) yapısını oluşturur. DRAM yapısında, çok sayıda bit hücresine aynı anda okuma, yazma ve erişim (access) işlemi gerçekleştirilmesi gerekir. Birden fazla sayıda DRAM dizisinden oluşan bir DRAM'de aynı anda birden fazla dizi üzerindeki çoklu bit hücrelerine erişim sağlanabilmektedir. Genel DRAM mimarisine devam etmeden önce dizi yapısını açıklayalım. DRAM dizisinde: Yan yana aynı kelime seç teli üzerinde çok sayıda hücre bir DRAM sırasını/satırını (a DRAM row) oluşturur. Bu satırlar/sıralar içerisinde farklı bit hücreleri grupları bulunur (kelimeler vb.), ve okuma ve yazma işlemleri DRAM arayüz kabiliyetlerine göre bir veya birden fazla bu grupların birleştiği bloklar halinde gerçekleşebilir. Benzer şekilde bir DRAM tasarımının sahip olması gereken veri boyutuna göre belirli sayıda DRAM satırı yerleştirilir. Böylece "DRAM satır sayısı x DRAM satırındaki hücre sayısı" kadar bit

boyutunda bir DRAM dizisi dolayısıyla bir DRAM kümesi ortaya çıkar.

DRAM dizisinde bit hücreleri bloklar halinde okunur ve yazılır ancak aynı anda sadece bir satıra erişim sağlanır ve bu satırdaki bit hücrelerine erişim yapılabilir. Bir DRAM satırındaki hücrelere erişim sağlanabilmesi için çevre devreler/yapılar gerekmektedir. Bu çevre yapıları ile birlikte tüm DRAM dizisi ise bir DRAM bank (küme) oluşturur. Çevre devre yapıları ve satırlar halinde bit hücrelerinden oluşan bir DRAM kümesi kavramsal gösterimi Şekil 2.4 ile sunulmaktadır. Şekilde belirtilen çevre yapılar (sürücüler, fark algılayıcılar ve çözücü) farklı bellek yapıları (örn: SRAM) için benzerdir. Ancak bit hücresi yapısı ve birim alana sığan bit hücresi sayısı (tasarımı ve kullanımı farklı olduğu için) farklı olduğu için yapıların tasarımları (sürücülerin boyutu, devre karakterleri vb.) da buna göre değişiklik gösterir, haricinde bu çevre devrelerin çalışma ve kullanım mantığı benzerdir.

Şekil 2.4'de gösterilen ve bir DRAM kümesinde bir satırın seçilmesi, o satırdaki bit hücrelerine erişilmesi ve çoklu okuma ve yazma yapılabilmesi için kullanılan DRAM kümesi yapıları şunlardır [1, 14].



Şekil 2.4 : DRAM küme (bank) yapısı temsili gösterimi.

- Bit hücresi: Bir bit veri saklayan DRAM mimarisi en temel yapı taşı.
- Kod Çözücü (decoder): DRAM'de erişim sağlanacak bit hücrelerine satırlar üzerinden ulaşılır. Bit hücrelerine ait her satırın bir etiketi veya adresi vardır. Uygulama tarafından hangi bit hücrelerine ne işlem yapılacağına dair bilgi

DRAM'e gelir. Önce hangi DRAM dizisi üzerindeki satıra işlem yapılacağı bilgisi çözülür, daha sonra da DRAM dizisine gelen satır adresi çözülür. İşte bu adres çözme işlemi yapıp hangi satırın aktif edileceğini bulan yapılar kod çözücü olarak adlandırılır. Kod çözücüler sıklıkla "ve değil" ve "veya değil" kapıları ile gerçeklenir. Satır sayısına göre kodlanarak giriş sayısı belirlenir.

- Kelime Seç (Word Select) Sürücü: Hangi satırın aktif edileceği donanımsal olarak çözüldükten sonra, o satır üzerindeki tüm bit hücrelerinin kapı terminallerine açacak voltajı beslemek gerekmektedir. Ancak aynı anda aktifleştirilecek (okuma veya yazma vb. için) bit hücresi sayısı çok fazladır, ve bu işlemin gecikmesinin çok az olması gerekir. Bu yüzden de, aynı anda bir satırda sürülecek bit hücresi sayısına göre ve maksimum gecikme gereksinimini sağlayan bir sürücü yapısı gerekmektedir, bu sürücü word select veya kelime seç sürücü olarak adlandırılır. Bu sürücü sıklıkla arka arkaya eviricilerle gerçeklenir. Eviricilerin boyutları ile, gecikme maliyetleri arasında getiri götürü vardır, bunu en iyilemeye yönelik çalışmalar gerçekleştirilmektedir [14, 26].

- Bit teli (Bitline) Sürücü: Kelime seç sürücü ile benzer ihtiyacı karşılamak için tasarlanmış bileşendir; amacı bit telleri üzerinden bit hücrelerine gerçekleştirilen işlemlerde gecikmeyi azaltmaktır.

- Ön doldurucu/yükleyici (precharge): Çok sayıda bit hücresi ve bu hücrelerin tümüne yükü aktaran bit teli bulunmaktadır. Ancak mantık 0'dan mantık 1'e yazma işlemi vakit alır ve bir gecikmesi vardır. Özellikle çok sayıda hücreyi sürmek gerektiğinde bu gecikme daha da artmaktadır. Gecikmeyi azaltmak için sürücüler kullanılır, ancak sürücülerin de alan maliyeti bulunmaktadır. İşte bu problemi çözebilmek için bit telleri mantık 1 (Vdd) yerine okuma veya yazma yapmadan önce Vdd/2'ye sürülür. Bu işleme ön doldurma adı verilmektedir.

- Fark algılayıcı (sense amplifier): Sayısal devrelerde mantık 0 ve mantık 1 için besleme gerilimine veya toprağa belirli bir eşiği geçene kadar çıkılması veya düşülmesi gerekir. Bu bağlamda yarı doldurulan bit tellerine bir yük geldiğinde bunun teli aşağı yada yukarı çekmesi beklenir. Bir bit hücresi içindeki yük ve bir sürü hücreyi bağlayan bir bit telini düşününce bu işlemin (örneğin Vdd/2 den Vdd'ye) ön doluma rağmen gecikme açısından problem oluşturmaktadır. Bunun çözümü bit telinde Vdd/2'den bir fark olduğunda tam Vdd veya 0 olmasını beklemeye gerek kalmadan, sadece Vdd/2'den aşağıda veya yukarda olduğuna karar verecek bir devre kullanmaktır. İşte bu devreler fark algılayıcı olarak adlandırılır. Bahsedilen farklar analog devrelerle ancak algılanıp adlandırılabilir, ve akım aynaları veya mandallar ile gerçeklenebilir. Akım aynalarının güç

tüketimini azaltmak için SRAM ve DRAM özelinde çalışmalar (Sadece belirli durumlarda çalışmasını sağlayacak çözümler vb.) yapılmaktadır [14, 27, 28].

Özetle, DRAM kümeleri için işlevsel akış şu şekildedir: Erişim yapılacak hücrelere ait satır adresi bir DRAM kümesine geldiğinde, kod çözücü tarafından çözülür. İlgili satır bulunur ve o satır için kelime seç/word select sürücü aktif hale gelir ve tüm satırdaki hücreleri aktifleştirir. O satırdaki bit hücrelerinden veya bit hücrelerine okuma veya yazma işlem tipine göre bit telleri üzerinden yük aktarımı tamamlanır, böylece yazma gerçekleşir. Okuma işleminde ise bu aktarım, bit telleri üzerinden fark algılayıcılar ile algılanarak sonuç elde edilir.

DRAM bit hücresi seviyesinde ise aktarılan çevre yapıları kullanarak bit hücrelerine erişimin işlevsel akışı ise şu şekildedir (bu adımlar aynı sırayla Şekil 2.2'de numaralarla, 4'e kadar, temsil edilmektedir.):

- Bir DRAM hücresine erişim için o hücrenin yer aldığı satıra erişmek gerekmektedir. Bu bağlamda, bellek kontrolcü birimi tarafından satır ve sütun adreslerinin olduğu komutlar gönderilir ve sonrasında bu komutlar kod çözücü tarafından çözülerek ilgili satır aktif edilir. Bu işleme Aktifleştirme (Activate) denilmektedir.

- Satır seçildikten sonra erişim transistörü açılır ve kapasitör üzerinde saklanan yük bit telleri üzerine aktarılır. Bit telleri çok uzun olduğu için kapasitörün bu telleri sürmesi beklenemez, bu sebeple bit telleri vdd/2 seviyesine çekilir, bu işleme ise precharge/öndoldurma denilmektedir.

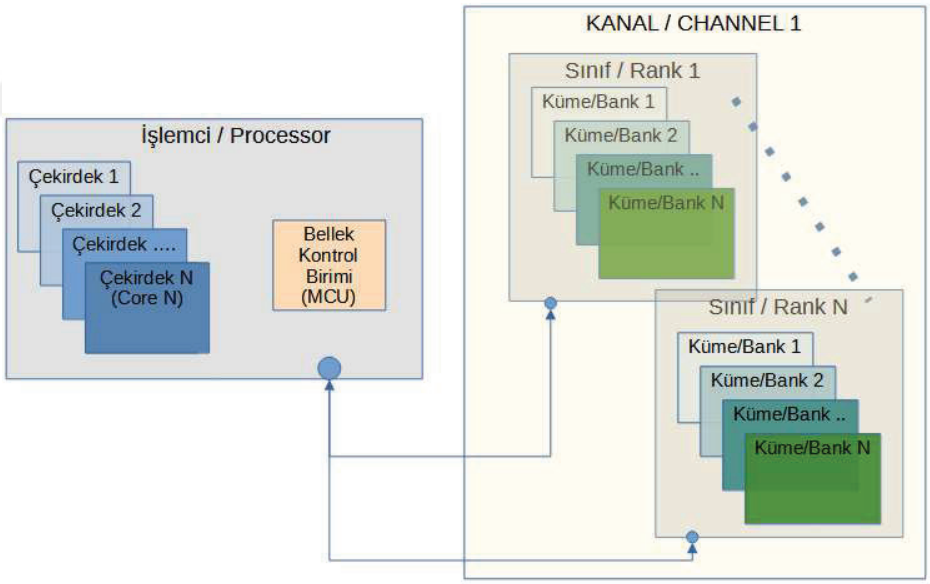
- Erişim için için satır açıldıktan sonra okuma işleminde kapasitörün yükü bit telleri üzerine aktarılır ve bu yük aktarımı ile Vdd/2 seviyesinden aşağı veya yukarı yönde epsilon miktarda gerilim farkı oluşur. Bu gerilim farkı ise fark algılayıcılar tarafından algılanır. Eğer bir DRAM mimarisinde yerel fark algılayıcılar varsa, alt dizin seviyesinde paralellik söz konusu olur [29], ve her hücrenin verisi o hücrenin ait olduğu alt dizin fark algılayıcı tarafından algılanır. Böylece okuma sağlanır.

- Yazma yapılacaksa, ilgili bit hücresine bit telleri üzerinden yazma gerçekleştirilir.

- En sonunda yazma veya okuma gerçekleştirildikten hemen sonra bit telleri tekrardan vdd/2 seviyesine çekilir veya Önyükleme (Precharge-4) yapılır. Bu sayede o satırdaki bit telleri bir sonraki erişim için hazır hale getirilir, bu şekilde bir sonraki işleme hazırlanmış olur ve gecikme azaltılmış olur.

DRAM mimarilerinde, aynı anda birden fazla satıra erişim sağlanabilmesi ve bu işlemlerin paralellenerek başarımın artırılması amacıyla, birden fazla küme tasarımı kurgulanır. Buna küme/bank seviyesi paralelleştirme denir [29]. DRAM kümeleri ise birleşerek sınıfları (rank) oluşturur ve bu sınıflar için ayrı giriş/çıkış arayüzü bulunur. DRAM için hangi satırlara ne veri yazılacağı işlemci veya dış dünya üzerinden bu arayüz ile sağlanır. Birden fazla sınıf/rank içeren bellek yapısı ise kanal (channel) olarak adlandırılır. DRAM'ler için her bir kanal bir DRAM yongası (chip) anlamına gelmektedir. DRAM mimari yapıları arasındaki hiyerarşi bu şekildedir, ve bunu betimleyen bir görsel Şekil 2.5'de sunulmaktadır.

DRAM hiyerarşisinde, tüm okuma yazma ve erişim işlemleri ve zamanlamaları işlemci veya işlem birimleri tarafındaki bellek kontrol birimi (memory controller unit veya memory controller, MCU) tarafından sağlanır. Veri aktarımı çift yönlüdür, ve her sınıfta bulunan arayüz üzerinden, "DRAM Komutları" aracılığıyla MCU tarafından kontrol edilir ve gerçekleştirilir. Aynı şekilde erişim sonrası, eğer erişim tipi okuma ise, okunan veri bu arayüz üzerinden iletilir [1]. Bir DRAM komutunda boşta bırakılmış (reserved) alanlar bulunmaktadır. Farklı uygulamalar için gerekmesi durumunda boşta bırakılan bu alanlar kullanılabilir.



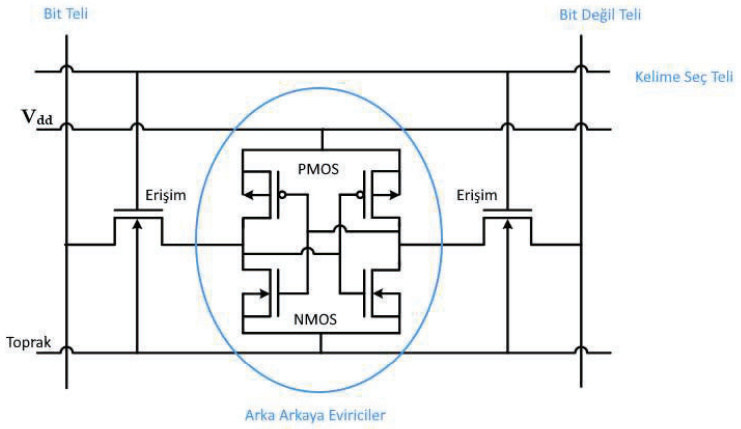
Şekil 2.5 : DRAM mimarisi temsili gösterimi.

2.2 SRAM Bit Hücesi ve Bellek Mimarisi

SRAM bit hücresi ve dizi yapısı ile ilgili temel bilgiler bu bölümde yer almaktadır. (Bu bölümde SRAM ile ilgili temel bilgileri anlatırken SRAM serim (layout) görselleri ve bazı hazır veriler için yüksek lisans tezimden [14] faydalanılmaktadır. Tez kapsamında SRAM için önerilen özgün fikirler için de anaçizgi tasarımları mümkün olduğunca

yüksek lisans çalışmalarımdan gelen önceki tasarımlarım değerlendirilmiş, doktora çalışmaları kapsamında SRAM için (sadece SRAM için; yüksek lisansta DRAM ve FPGA üzerine çalışma zaten yapılmamıştır.) önerilen özgün fikir ve tasarımlar bu temel tasarımlar üzerinden geliştirilmiştir, ayrıca karşılaştırma değerlendirmesi de sunulmuştur.)

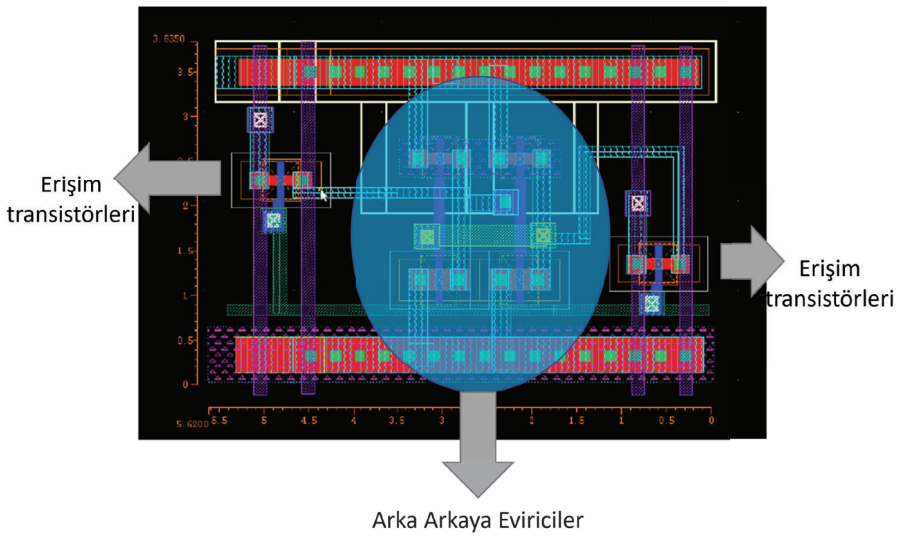
Bilgisayarlarda önbellek, gömülü sistemlerde yazmaç ve birçok elektronik sistemde bellek olarak kullanılan diğer bir mimari yapı ise durağan rasgele erişimli bellek (SRAM, olarak anılacaktır)'lerdir. SRAM bellek mimarisinin en temel bileşeni SRAM bit hücreleridir. SRAM bit hücreleri, DRAM bit hücreleri gibi erişim transistörleri içermektedir, ancak farklı olarak kapasitör yerine veri saklamak için de, sıklıkla, transistörleri kullanmaktadırlar [2, 14]. Transistörlerden oluşan bir SRAM bit hücresi gösterimi Şekil 2.6 ile verilmektedir. DRAM bit hücresine benzer mantıkla SRAM bit hücrelerine kelime seç/word select teli üzerinden erişilir, bit telleri üzerinden de veri aktarımı sağlanır.



Şekil 2.6 : SRAM bit hücresi şematiği.

Sıklıkla bir NMOS ve bir PMOS transistörü ile evirici elde edilmektedir [2, 14]. Bir evirici (Inverter) girişine mantık "0" verilirse çıkışta mantık "1" (veya tam tersi) elde edilir. İşte bu eviricilerin arka arkaya birbirlerine (bir eviricinin girişi diğerinin çıkışına ve aynı şekilde diğer eviricinin ikinci eviricinin çıkışı da diğer eviricinin girişine) bağlanarak bir SRAM bit hücresi için veri saklayan bir yapı elde edilmiş olur. Şekil 2.6 içerisinde yuvarlak şekil ile belirtilen transistörler arka arkaya eviricileri göstermektedir. Bahsedilen NMOS transistörler her bir evirici için şeklin alt kısmında, solda ve sağda yer alır, PMOS transistörler ise her bir evirici için şeklin üst kısmında, solda ve sağda yer alır. Arka arkaya eviriciler transistörlerden oluştuğu için sadece

veri yollarındaki parazitik kapasitanslar haricinde veri saklamak için tasarlanmış bir veri sığası yoktur ve dolayısıyla saklanan verinin kaybolmaması için transistörlerin sürekli olarak beslenmesi (Vdd) gerekmektedir. Bit hücrelerinde Vdd ve toprak hatları da bu şekilde gösterilmektedir. SRAM bit hücresini oluşturan arka arkaya eviriciler nedeniyle bir SRAM bit hücresinde bit ve bit değil telleri ve bunlar için ayrı erişim transistörler bulunmaktadır (daha az veya fazla sayıda transistör kullanılan SRAM bit hücresi tasarımları da mevcuttur, ancak tez kapsamında önerilecek bu tasarım detayında değişiklikten etkilenmez ve uyarlanabilirdir, o yüzden yaygın kullanılan tasarımlar temel tasarım veya ana çizgi olarak kullanılmaktadır.). Dolayısıyla, bit hücreleri için yük aktarım işlemi yani okuma yazma işlemleri için bit ve bit değil telleri kullanılmaktadır. Şekil 2.6'de bu teller ve erişim transistörleri de yer almaktadır. Şematiği temsili gösterilen SRAM bit hücresinin serim (layout) görüntüsü de Şekil 2.7'de yer almaktadır (Cadence Virtuoso ile).

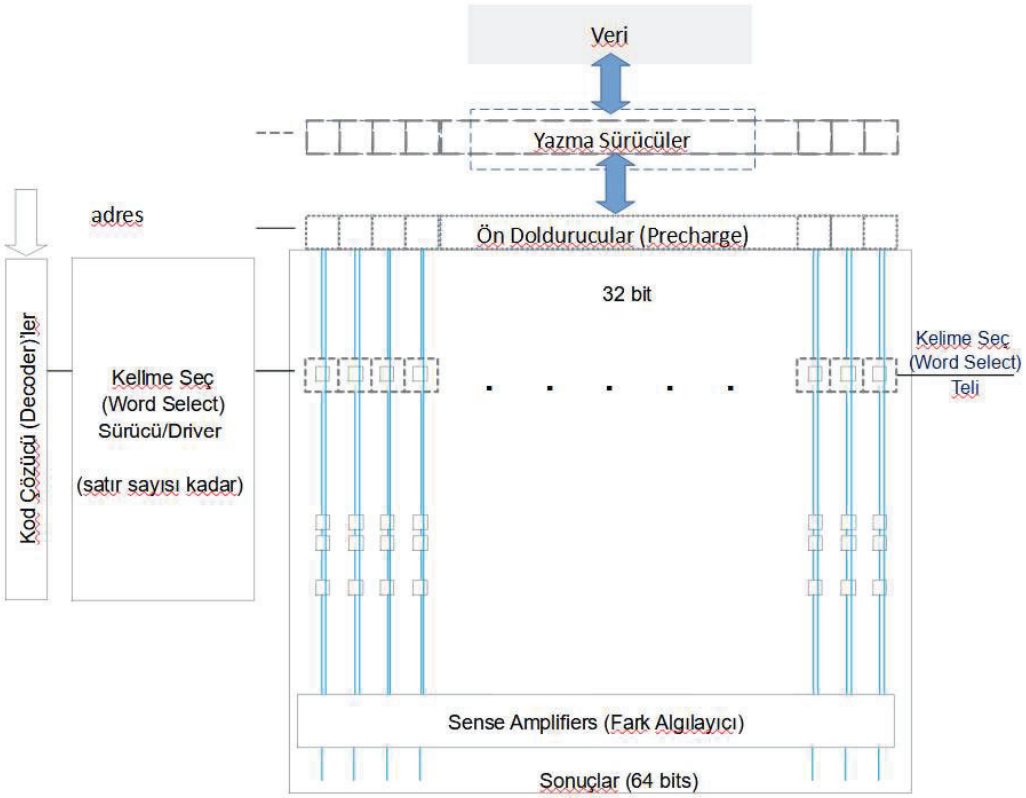


Şekil 2.7 : SRAM bit hücresi serim görüntüsü.

Şekil 2.7'de orta kısımda eviriciler (açık mavi şekille gösterilmektedir), yanlarda (okla gösterilmektedir) ise bit ve bit değil telleri için erişim transistörleri yer almaktadır, üstte ve alttaki açık kırmızı kısımlar ise Vdd ve toprak hatlarıdır.

SRAM bit hücreleri biraraya gelerek SRAM dizi (SRAM array) yapılarını oluşturur. Çoklu okuma ve yazma işlemleri bu yapılar üzerinden sağlanır, ve bu işlemleri sağlamak için DRAM'de olduğu gibi çevre devre tasarımlarına ihtiyaç duyar. SRAM bit hücreleri satırlar halinde bulunur ve gereksinimde belirtilen veri boyutu ihtiyacına göre satır sayısı ve o satırdaki hücre sayısı belirlenerek tasarım yapılır. Böylece (DRAM'deki gibi) bir SRAM dizisinin veri boyutu kapasitesi: "Bir satırdaki SRAM bit hücresi sayısı x satır sayısı" olur. SRAM satırındaki tüm hücrelerin erişim transistörlerinin kapı terminalleri kelime seç teline (word select line) bağlıdır. Bu tele,

kelime teli/hattı (word line)'da denilebilir. Ayrıca, satır seçme (row select) teli olarak da adlandırılabilir, çünkü bir satır üzerindeki hücrelerin kapıları bu tele bağlıdır, ancak bir satır birden fazla alt satıra veya hücre grubuna bölünebildiği için genelde kelime seç/word select teli olarak kullanılmaktadır [30]. Bir bit hücresine erişileceğinde bu tel üzerinden erişim transistörü açılır. SRAM dizisinin ve çevre yapılarının temsili gösterimi Şekil 2.8 ile sunulmaktadır.



Şekil 2.8 : SRAM mimarisi temsili gösterimi.

SRAM mimarisini oluşturan yapılar şunlardır (erişim işlemlerinin işlevsel akış sırasına uygun şekide anlatılmaktadır.):

- Kod Çözücü (decoder): SRAM dizisinde hangi hücreye erişilmek isteniyorsa, o hücrenin olduğu satırın seçilmesi gerekir, DRAM ile benzer mantıkla, bu seçim işlemi kod çözücüler tarafından gerçekleştirir. Kod çözücüler, SRAM dizisine gelen adresi çözer, ve o adresdeki satıra ulaşılır. (Kod çözücü ve aşağıdaki diğer yapıların doğrudan kendi devre tasarımlarına yönelik güç tüketimi azaltmak ve başarım için iyileştirmeler ve teknikler çalışılmaktadır [14], ancak tez kapsamında önerilen fikirler daha çok DRAM dizini ve bit hücrelerine yönelik olduğu için bu tasarım detaylarına gerekmedikçe yer verilmemektedir.)
- Satır Seçme (Row Select) Sürücüsü veya Kelime Seç (Word Select) Sürücü: Ulaşılan satırdaki erişilecek hücrelerin aktif hale getirilmesi gerekmektedir,

bunun için kelime seç/word select teli üzerinden hücrelerin erişim transistörleri açılır. Ancak DRAM'deki problem burada da geçerlidir, çok fazla sayıda bit hücresine erişim sağlanması gerekmektedir, ancak bunun gecikmeye neden olmaması gerekir, aynı zamanda da alan maliyetine dikkat edilmelidir. Bunu sağlayacak çözüm olarak sürücüler kullanılır. Kelime seç teli üzerindeki bit hücrelerini gecikmeyi aza indirerek açmak için ve teli sürmek için kelime seç (word select) sürücü kullanılır (telde olduğu gibi; bu sürücü satır seçme sürücüsü (row select driver) olarak da adlandırılabilir, ancak satırlarda kelimeler halinde sürme ihtiyacı olabileceği için ve literatürdeki yaygın kullanım nedeniyle kelime seç sürücü olarak anılmaktadır).

- Yazma sürücüler: Kelime seç sürücüye benzer mantıkla çalışır, yazma telini sürmek için tasarlanır (birbirlerine bağlı olmayan farklı kanal genişliğindeki transistörlerden oluşan arka arkaya evirici kullanmak vb.) ve kullanılırlar.

- Bit teli (bit line) Sürücü: Kelime seç sürücü ile benzer ihtiyacı karşılamak için tasarlanmış bileşendir; amacı bit telleri üzerinden bit hücrelerine gerçekleştirilen işlemlerde gecikmeyi azaltmaktır.

- Ön doldurucu/yükleyici (precharge): Bir SRAM bit teli üzerinde çok sayıda bit hücresi vardır, ve özellikle devam eden kısımlarda anlatılacak olan erişim kapısı (port) sayısı da eklendiğinde hem bit hücreleri hem de bu bit hücrelerinin serimdeki alanlarından kaynaklı tel uzunluğu artışı bu telin sürülmesini zorlaştırmaktadır, bunun için bit teli sürücü kullanılmasına rağmen, bir bit hücresinin yükünü buraya aktarması ve özellikle tel üzerinden 0'dan mantık 1 okunacak seviyeye kadar sürmesi çok uzun bir işlemedir. Bunun yerine teller (DRAM'lerde olduğu gibi) Vdd/2'ye çekilir, bu işlem ön doldurma işlemidir ve bu işlemler öndoldurucular tarafından sağlanır. Bir SRAM dizisinde ön doldurucular hem bit hem de bit değil tellerini Vdd/2 seviyesine çekmektedir.

- Fark algılayıcı (sense amplifier): Hem sürücüler kullanıldı, hem de hatlar yarısına (Vdd/2'ye) çekildi, ancak yine de yarı doldurulan bit tellerine bir yük geldiğinde bunun teli aşağı yada yukarı çekmesi gecikmeye ve güç tüketimine neden olacak bir durumdur. İşte bunu daha da verimli hale getirmek için fark algılayıcılar kullanılır. Bir fark algılayıcı, bit telinde Vdd/2'den bir fark olduğunda tam Vdd veya 0 olmasını beklemeye gerek kalmadan, sadece Vdd/2'den aşağıda veya yukarda olduğuna karar verecek yapıdır. Fark algılayıcılar, bu farkı algılayabilecek akım aynaları veya mandallar gibi analog devrelerden oluşur [14, 27].

Özet olarak, hangi bit hücrelerine erişim olacaksa sırayla ilgili hücrenin olduğu

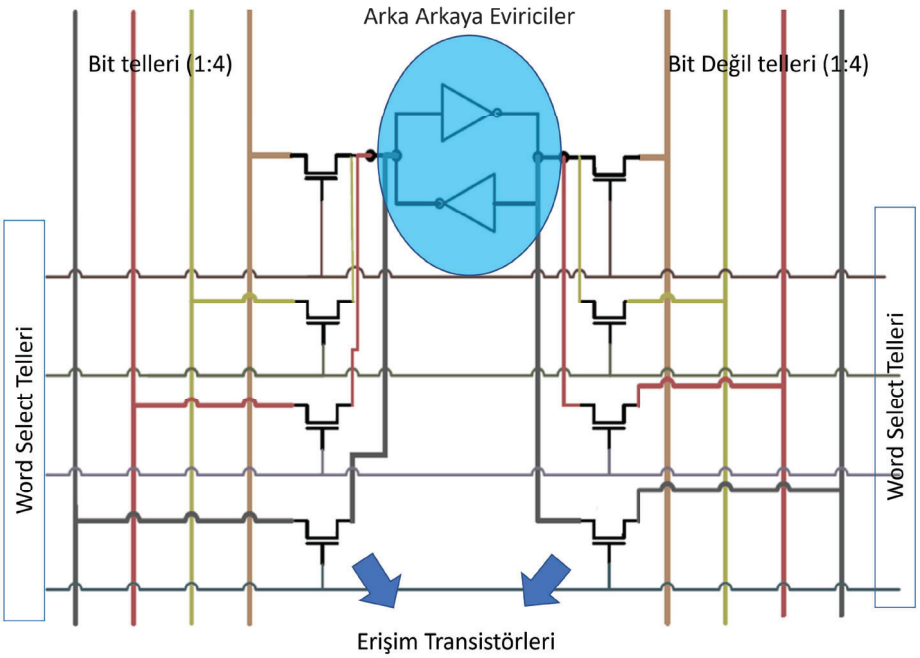
satırın adresi iletilir, bu adres kod çözücüyle çözülür, ilgili satıra ulaşılır ve sürücüler yardımıyla o satırdaki hücrelerin transistörleri üzerinden hücreler açılır, bit teli ve bit değil teli üzerinden hücre içindeki yüke bağlı olarak Vdd/2'nin altına veya üstüne fark oluşur ve fark algılayıcılar bunu algılayarak okuma yapılır, veya yazma sürücüler vasıtasıyla erişim için açılan bit hücrelerine veri yazılır.

Bir bit hücresinde aynı anda birden fazla okuma yapılması için "erişim kapısı (port)" olarak adlandırılan tasarım yöntemi uygulanır. Bu yöntemde, bit hücresinde veri tutan tek bir arka arkaya evirici bloğuna bağlı birden fazla erişim transistörü (bit ve bit değil için ayrı ayrı birer erişim transistörü kullanılır) eklenir. Bir bit hücresinde veri saklayan yapı tek olduğu için çoklu erişim kapısı (port) mantığı sadece okumak için paralelleştirme sağlar, aynı anda bir hücreye çoklu yazma işlemi veri kaybına neden olur.

SRAM bit hücresinde normal durumda (tek erişim kapısı (port)) arka arkaya eviriciler için 4 transistör, erişim için ise 1x2 transistör kullanılır. Erişim kapısı (port) sayısı artırılması gerekirse; her eklenecek erişim kapısı (port) için "erişim kapısı (port) sayısı x 2" kadar da erişim transistörü eklenir. Örneğin; 4 erişim kapısı (port) içeren bir SRAM bit hücresi toplamda; arka arkaya eviriciler için 1x4, bit teli erişim transistörleri için 1x4 ve bit değil teli erişim transistörleri için 1x4 olmak üzere 12 adet transistörden oluşmaktadır. Arka arkaya transistörler haricinde 8 adet transistör eklenir, 4 erişim kapısı (port) için. Ayrıca 4 erişim kapısı (port) kadar bit ve bit teli de eklenmesi gerekmektedir. Buna bağlı olarak da fark algılayıcı gibi çevre devrelerin yapısı da değişmektedir. Örnek seçilen, 4 erişim kapısı (port) içeren bir SRAM bit hücresine ait temsili gösterim Şekil 2.9'da yer almaktadır.

Mavi kısım arka arkaya eviricileri gösterir, her bir bit ve bit değil teli farklı erişim kapısı (port) için farklı renkle gösterilir, erişim transistörleri ise okla belirtilmektedir. Hangi erişim kapısı (port) açılacağının belirlenmesi de yine kelime seç (word select) teli üzerinden yapılacağı için bu tasarıma göre erişim kapısı (port) sayısı kadar kelime seç teli de eklenir. Bu teller, transistörler ve yapı için farklı gerçekleme yöntemleri de bulunmaktadır, ancak yaygın kullanılan tasarım ve yöntemler anaçizgi tasarımı olarak tercih edilmiştir.

SRAM'ler farklı uygulama alanlarında ve çeşitli amaçlar için kullanılabilmektedirler. Örneğin, yazmaç öbeği olarak, veya bir bilgisayarda önbellek olarak SRAM kullanımı yaygındır [31]. Kullanım amacına göre farklı erişim kapısı (port) sayıları, farklı satır ve hücre sayıları belirlenerek farklı SRAM tasarımları oluşturulur. Ayrıca SRAM kullanılarak geliştirilen mimari yapılar da kendi içlerinde kullanım yerine göre farklı boyut ve tasarımda olabilmektedir [14]. Doktora kapsamında SRAM için önerilen tasarım çözümü tüm bu farklılıklara rağmen geçerlidir ve uyarlanabilirdir.



Şekil 2.9 : SRAM 4 erişim kapısı (port) içeren bit hücresi.

2.3 Devingen ve Durağan Güç Tüketimi

Tez çalışmaları kapsamında DRAM'ler, SRAM'ler ve FPGA'ler için özgün fikir ve tasarımlar geliştirilmiştir, ve bu önerilen tasarımların temel amacı başarımı istenen seviyede tutarak güç tüketimini düşürmek ve bunu donanımın kendi kendine yapabilir olmasıdır. Bu bölümde, bu donanımlar için güç tüketimini oluşturan devingen ve durağan güç tüketimi anlatılmaktadır.

Devingen Güç Tüketimi (Dynamic Power Consumption)

Donanım ve devrelerde güç tüketimini hesaplayabilmenin veya anlayabilmenin en kolay yolu bir devreyi sığa (kapasitans) ve doldurma (charge) için gerekli besleme gerilimi seviyesine taşımaktır. FPGA'ler adından da anlaşılacağı üzere, programlanabilir kapılar kullanılarak tasarlanır, ve sonuç olarak transistörlerden oluşmaktadır. DRAM hücreleri transistör serimine ilave edilen yüksek sığa alanı (kapasitör) ve bir erişim transistöründen oluşmaktadır, önceki bölümde detaylı anlatıldığı üzere, ve SRAM bit hücreleri veriyi saklayan ve erişim sağlayan transistörlerden oluşmaktadır. Bellek mimarisinde okuma ve yazma işlemleri için kullanılan çevre devreler de, işlemciler ve işlem birimleri de benzer şekilde aslında çok sayıda transistörün biraraya gelmesiyle oluşan devrelerdir (VLSI: Very Large Scale Integrated Circuits). Dolayısıyla, bu donanımları transistör ve teller (güç ve veri hatları) seviyesine indirgeyebiliriz. Bu yalınlıkta bakıldığında, devrelerde 2

temel sığa vardır, transistörün terminalleri (source/kaynak, drain/savak, gate/kapı ve body/substrate/alttaş) arasında oluşan sığalar ve telin/hattın (çok geniş ölçekli tümleşik devrelerdeki veri ve güç iletim hatları) sığası [2, 32–34]. Devreler aktifken, transistörlerin belirli sıklıkla anahtarlanması (açılıp kapanması) sırasında, sığanın (yükün) doldurulup boşalması nedeniyle gerçekleşen güç tüketimi, Devingen Güç Tüketimi (Dynamic Power Consumption)'dir.

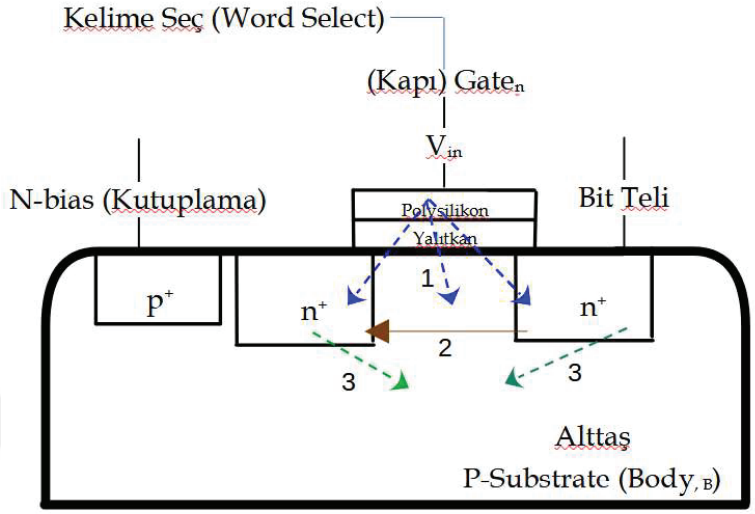
Bir devre için (veya donanım için), devingen güç tüketimi; anahtarlama sıklığına, sığaya (transistörler kaynaklı ve teller kaynaklı), ve sığayı süren besleme gerilimlerine bağlıdır. Devingen güç tüketimi (P_L); devre sığası (C_{devre}), anahtarlama frekansı/sıklığı (f) ve besleme gerilimi (V_{dd}) üzerinden denklem 2.1 ile belirtildiği gibi hesaplanır [2, 35].

$$P_{devingen} = f \; x \; C_{devre} \; x \; Vdd^2 \tag{2.1}$$

Bir donanımda (örneğin; FPGA, veya Bellek) farklı frekanslarda, farklı sığalarda ve farklı besleme gerilimlerinde devreler olacaktır. Bu durumda, sürülen hatlar ayrıştırılarak güç tüketimleri hesaplanabilir (Örnek olarak, FPGA için bu çalışmada VCCINT hattı üzerinden güç düşürülmesi ve sonuçlarının elde edilmesi sağlanmıştır.) [36]. Temelde, bu denklem bu çalışmada önerilen tüm tasarımlar için şu şekilde kullanılmaktadır: "Besleme gerilimi artarsa devingen güç tüketimi gerilimin karesiyle orantılı olarak artar", ve "Bir devreye erişim beklendiği üzere güç tüketimini artırır". (Devingen güç tüketiminin detay seviyede bir kaynağı da anahtarlama sırasında gerçekleşen kısa devre akımlarıdır [37], bunları da bu tez kapsamında yukarda belirtilen "f" sıklık altına dahil ediyoruz. Çünkü sıklık arttıkça anahtarlama da artacak ve dolayısıyla anahtarlama sırasında gerçekleşen güç tüketimi artacaktır, bizim tasarımlarımızda kullanmamız gereken prensip erişimi azaltmak olduğu için bunu da dikkate almaktadır.)

Durağan Güç Tüketimi (Static Power Consumption) ve Sızdırma Akımları Herşey sızdırır, transistör, kapı, bellek hücresi veya VLSI devresi. Herhangi bir anahtarlama, mantık işlemi, erişim veya hesaplama yapılmasa bile istenmeyen akımlar nedeniyle enerji kaybedilmektedir. İşte bu fenomene "leakage" veya sızdırma, bu akımlara da "leakage currents" veya sızdırma akımları denilir. Bir devrede sızdırma akımları kaynaklı durağan enerji kayıpları yaşanmaktadır, ve devre aktif değilken de "durağan güç tüketimi" gerçekleşmeye devam eder. Devingen güç tüketiminde izlenen yaklaşımı durağan güç tüketimi için de uygulayabiliriz ve devreleri transistör seviyeye indirgeyebiliriz. Bir transistörde temel olarak 4 terminal bulunmaktadır: Gate (Kapı), Drain (savak), Source (kaynak) ve Body (alttaş). Bu terminaller bağlı olduğu alanların

ismini almışlardır. Sızdırma akımları transistör içindeki bu bölgeler arasında durağan durumda iken bile gerçekleşmeye devam ederler. Şekil 2.10'da temsili bir transistör kesit görüntüsü üzerinden terminal bölgeleri ve sızdırma akımları, numaralandırılarak, gösterilmektedir [2, 14].



Şekil 2.10 : NMOS üzerinde sızdırma akımları temsili gösterimi.

Transistörler için üç temel sızdırma akımı kaynağı bulunmaktadır (bu tez kapsamında, sızdırma akımlarına önerilen tasarımlar tarafından kullanılacak etkiler özellikle aktarılmaktadır, [14] içinde daha kapsamlı anlatımı yer almaktadır.).

- Kapı sızdırma akımı (Şekil 2.10, 1 numaralı oklar): Kapı terminaline doğru elektron geçişi veya ters yönde oluşan akımdır. Normalde kapı üzerindeki yalıtkan malzeme baraj görevi görmektedir, ancak bu yalıtkanın gelişen teknolojiyle giderek kalınlığının azalması nedeniyle diğer üç terminalden kapı terminaline elektron geçişi yaşanmaktadır [14, 38, 39].Kapı sızdırma akımını etkileyen önemli parametreler: Kanal genişliği (W), yalıtkan kalınlığı (t_{ox}), savak kaynak arası gerilim (Vds'nin karesiyle orantılıdır). Kullanılacak etki: Kapı sızdırma akımı gerilimin karesiyle orantılı artmaktadır.

- Alt eşik değerinde sızdırma akımı (Şekil 2.10, 2 numaralı ok): Transistörün girişine (Kapı) 0'dan yüksek ama eşik değer, Vth, den daha düşük bir besleme yapılıyorsa bu durumda transistör kapalı olmasına rağmen besleme gerilimiyle (Vds) artan bir sızdırma akımı oluşur. Alt eşik değerinde sızdırma akımını etkileyen önemli parametreler: Eşik değer voltajı (Vth), Sıcaklık (V_T), Kanal genişliğinin boyuna oranı (W/L), ikinci dereceden etkili besleme gerilimi. Kullanılacak etki: Eşik değer voltajı artırılabilirse, sızdırma akımları düşer. Alt eşik değerinde sızdırma akımı sıcaklıkla orantılı, besleme gerilimiyle dolaylı orantılı artmaktadır.

- Eklem sızdırma akımı (Şekil 2.10, 3 numaralı oklar): Normalde ters yönde akım geçirmeyecek şekilde oluşturulan eklem (p-n junction) diyotunda ortaya çıkan ters yönde sızdırma akımıdır.

Özet olarak, bir VLSI devre veya bir donanımın; artan sıcaklıkla (1) ve besleme gerilimleriyle (2) sızdırma akımları ve dolayısıyla durağan güç tüketimi artmaktadır (Ayrıca transistörler küçüldükçe de sızdırma artar). Eşik değer voltajının (3) artmasıyla ise durağan enerji kayıpları azalır. Doktora çalışmaları kapsamında,

- (2) kullanılarak: FPGA için önerilen tasarımlar, geliştirilen akıllı mekanizmalarla başarımın istenen seviyesi koruyarak güç tüketimini azaltmak için Vdd'nin uyarlamalı düşürülmesini sağlar (Bölüm 5, Bölüm 6).
- (1), (2) ve (3) kullanılarak: DRAM için önerilen tasarımlar, düşük güç ve yüksek başarım için; uyarlamalı (sıcaklığa göre de) olarak eşik değer gerilimin yükseltilmesini ve gerilimin dinamik ölçeklenmesini sağlar.
- (3) SRAM için önerilen tasarımlarda başarım korunarak, güç tüketimini düşürmek için, alan maliyeti iyileştirilmiş içeriğe uyarlamalı mekanizmalarla eşik değer geriliminin yükseltilmesi sağlanır.

2.4 Alttaş Kutuplama ile Sızdırma Azaltma ve Gerilim Ölçekleme

VLSI devrelerde durağan güç tüketimini ve sızdırma akımlarını azaltmak için sıcaklığın azaltılması, fabrika seviyesinde kanal parametrelerinin, yalıtkan kalınlığının vb. değiştirilmesi ve eşik değerin artırılması gerekir. Ancak, fabrika seviyesinde sızdırma akımlarını azaltacak bu çözümler, aynı zamanda gecikmelerin artmasına ve gürültüye karşı hassasiyetin yükselmesine neden olmaktadır, ve aynı zamanda alan maliyetine de neden olabilmektedir. Örneğin eşik değer geriliminin (Vth: Threshold Voltage) artırılması sızdırma akımlarını azaltacaktır, ancak transistörde anahtarlama gecikmelerine neden olacaktır. Bu yüzden de, fabrika seviyesinde tüm transistörlerin eşik değer gerilimlerini yükseltecek şekilde üretim ve tasarım mümkün olsa da, başarımı ve güvenilirliği azaltacağı için tercih edilmez.

Eşik değer gerilimini tüm transistörler için artırmak başarımı ve güvenilirliği düşüreceği için bunun yerine sadece belirlenen transistörler için fabrika seviyesindeyken yüksek değer atanıp, buna göre üretilmesine yönelik bazı tasarımlar önerilmiştir daha önce. Örnek olarak, [4] çalışmalarında buna yönelik bit hücreleri için asimetrik hücre tasarımı, "Asimetrik SRAM (ASRAM)" tasarımı önerilmiştir. Buna göre mantık "1" tutan bir SRAM bit hücresinde, mantık "1"in uygulandığı

PMOS kapısı ve mantık "0" uygulanan NMOS kapısının eşik değerlerinin fabrika seviyesinde yüksek olması kurgulanmıştır. Bu sayede, bu transistörler kapalıyken, durağan durumdalarken, daha az sızdırır hale gelmeleri sağlanmıştır. Aynı şekilde, erişim transistörlerinden de açık olan PMOS transistöründen bit teline sızdırma olmaması için ilgili olanına sadece yüksek vth olacak şekilde ayarlama yapılmıştır. Evet, bu sayede sızdırma akımları azalmış ve böylece durağan enerji kayıpları düşürülmüştür. Peki, mantık "0" tutulma oranı daha fazla olursa? Bu durumda, mantık "0"'a göre yüksek Vth olacak transistörler belirlenir. Ancak, bu durum değişkense, ki uygulamadan uygulamaya değişmesi mümkündür.

Eğer "1" tutmak için optimize edilmiş bit hücrelerine, çoğunlukla "0" gelirse bu durumda dinamik değişiklik yapılamadığı için sızdırma akımları yeniden eski yüksek seviyelerine dönecektir. Bu problemin çözümü, eşik değer voltajını uyarlamalı olarak değiştirmenin bir yolunu bulmaktır.

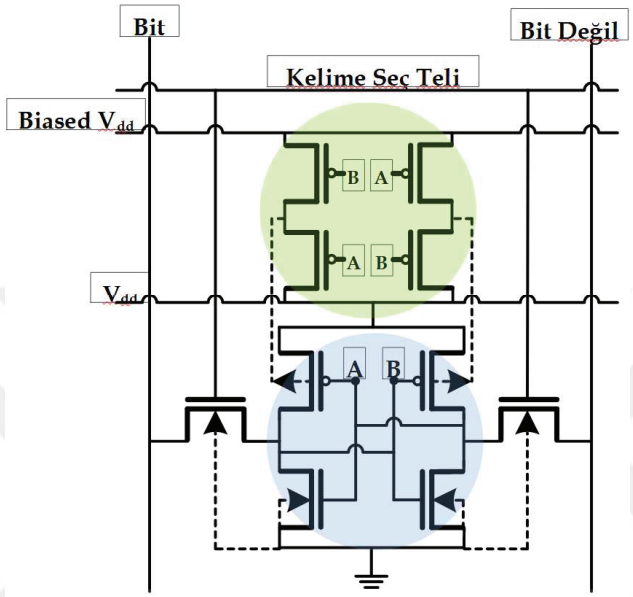
Transistörlerde eşik değer gerilimi fabrika seviyesinde (üretim sürecinde) atanır, ancak bu gerilimde sonradan değişiklik oluşturabilecek parametreler de vardır. Bir transistör için eşik değer geriliminin hesaplanması Denklem 2.2 ile sunulmaktadır.

$$V_{th} = V_{th_0} + \gamma * (\sqrt{(2*\theta + |V_{SB}|)} - (\sqrt{(2*\theta)}) \tag{2.2}$$

Denkleme göre fabrika seviyesinde belirlenen eşik değer voltajı, Vth_0 olmaktadır. Ancak, bunun üretimden sonra da sunulan bu parametrelere göre değişebileceği görülmektedir. Buna göre, V_{SB} gerilimi (Kaynak ile alttaş arası gerilim farkı) eğer 0'dan farklı ise, veya bu gerilimi ayarlayarak eşik değer gerilimi fabrika seviyesinde belirlenen Vth_0 değerinden farklılaşabilmektedir. İşte bu yönteme "body biasing", alttaş kutuplama yöntemi denilmektedir. Alttaş kutuplama yönteminde, NMOS'lar için kutuplama/bias gerilimi 0'dan aşağıda, PMOS'lar için bu gerilim besleme geriliminden yukarıya (veya kaynak geriliminden) bir değere getirilerek eşik değeri güncellenebilmiş olmaktadır. Ancak, burada şöyle bir problem bulunmaktadır; NMOS transistörler için eksi gerilim uygulama ortamında devreye karmaşıklık ve alan maliyeti getirmektedir [14]. PMOS transistörler için besleme geriliminin üstünde gerilim uygulamak ise, hali hazırda sıklıkla kullanılan ve kolay bir yöntemdir. Böylece sızdırma akımlarının azaltılabilmesi için Vth düşürmek amacıyla alttaş kutuplama yönteminin kullanılabileceği görülmektedir. Denklemde yer alan "γ", alttaş kutup katsayısıdır, "θ" ise Fermi potansiyelini gösterir, ancak burada asıl odaklanılan kısım; kaynakla alttaş arası kutuplama geriliminin Vth_0 ile Vth arasında fark oluşturabilmesidir.

Fabrika seviyesinden asimetrik olarak yüksek eşik değer voltajı belirlemek yerine,

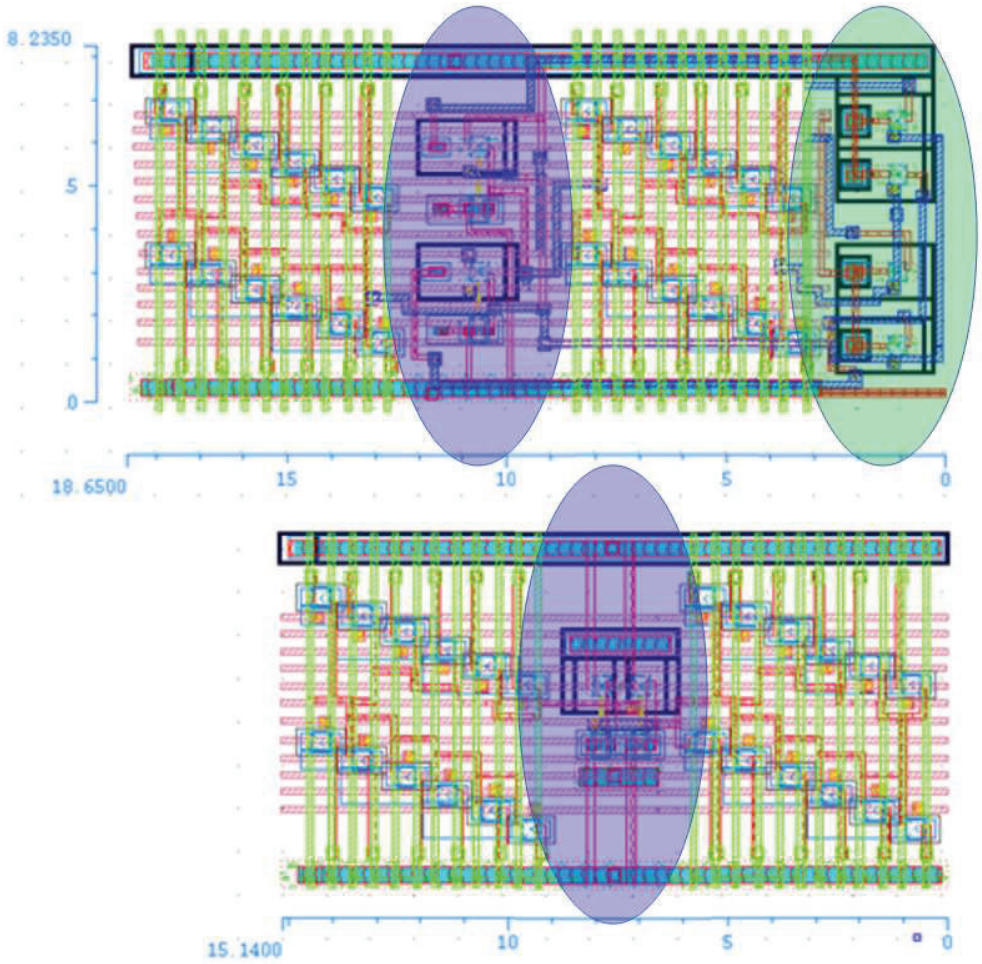
eşik değer voltajını içeriğe uyarlı değiştirmek mümkündür ve önceki çalışmamda ve tezimde [5, 14], alttaş kutuplamayı dinamik ve uyarlamalı olarak transistörlere uygulayabilen ve eşik değer voltajını belirli bir girdiye göre donanımın kendi kendine değiştirebildiği; "Content-aware SRAM (CSRAM)" tasarımı önerilmiştir. İçerik uyarlamalı bit hücresi tasarımına ait şematik gösterimi ise Şekil 2.11 ile sunulmaktadır [5, 14].



Şekil 2.11 : İçerik uyarlamalı bit hücresi tasarımı.

Şekil 2.11'de yeşil daire içindeki alan alttaş kutuplamayı, mavi daire içindeki alan ise bit hücresini göstermektedir. Buna göre, bit hücresinde mantık "1" tutulurken soldaki NMOS ve sağdaki PMOS kapalı durumdadır ve az sızdırma olması için PMOS'a yüksek gerilim uygulanır. Aynı şekilde "0" tutulurken de soldaki PMOS'a yüksek kutuplama/bias gerilimi verilir. Böylece, hem 0 hem 1 tutarken durağan durumda olan transistörlerin kutuplama/bias gerilimleri yüksek uygulanarak sızdırma akımları düşürülmüş olur.

12 erişim kapısı (port) içeren bir içerik uyarlamalı bit hücresi serim görüntüsü ise Şekil 2.12'de yer almaktadır, ayrıca şeklin alt kısmına baseline tasarım yani 12 erişim kapısı (port) içeren normal SRAM bit hücresi serim görüntüsü eklenmiştir [5, 14]. Bu şekilde yeşil elips içine alınan kısımlar, alttaş kutuplama devreleridir, mavi elips içine alınan kısımlar ise arka arkaya evirici devreleridir. Kalan kısımlar ise güç hatları ve erişim kapısı (port) sayısı kadar bit ve bit değil telleri için erişim transistörleridir. Bu şekilden de görüleceği üzere, bit hücresi içinde alttaş kutuplama devresinin alan maliyeti yüksektir. Bu yüzden, yine önceki çalışmalarımda birden fazla hücreyi bir kutuplama devresine bağlama çözümünden bahsedilmiştir, ancak bu durumda halen alan maliyeti çözülmüş olmamaktadır.



Şekil 2.12 : İçerik uyarlamalı ve anaçizgi bit hücresi serimleri.

Alan maliyeti problemini çözmek için, çok daha fazla sayıda bit hücresini bir kutuplama devresine bağlamak gerekir. Bu durumda ise, başka bir problemle, ASRAM tasarımındaki benzer durumla karşılaşmış oluruz: Farklı durumlar için uyarlama yapısı kaybolmuş olur. Çünkü tek bitin içeriği diğer bit hücrelerindeki tutulan veriyi yansıtmadığı durumda sızdırma akımları tekrar eski seviyesine yükselmeye başlar. Bu tez kapsamında, doktora çalışmalarının bir parçası olarak bu problemlere çözüm bulan özgün bir SRAM bit hücresi tasarımı önerilmektedir.

Bu tezde ayrıca, bilindiği kadarıyla ilk kez, DRAM'ler için de sızdırma akımlarının uyarlamalı kontrol edilebildiği DRAM hücresi, DRAM satırı ve DRAM dizini özgün tasarımları önerilmektedir. Bu özgün tasarımlar sayesinde DRAM farklı içerik ve girdilere göre kendi kendine alttaş kutuplama yapmakta, ve bu sayede sızdırma akımlarını azaltarak daha düşük güç tüketimi ve daha yüksek başarım sağlayabilmektedir.

VLSI devrelerde devingen güç tüketimi (Bölüm 2.3) besleme gerilimiyle (karesiyle orantılı olarak) artmaktadır. Diğer taraftan da, durağan güç tüketimi ise; farklı sızdırma

akımlarının besleme gerilimiyle (üssel ve doğrusal orantı söz konusudur farklı sızdırma akımı kaynakları için) artmaktadır. Özet olarak hem devingen hem de durağan güç tüketimini azaltmak için gerilimin düşürülmesi gerekmektedir. SRAM ve DRAM'lerde düşük güç tüketimi için voltajın düşürülmesine yönelik çalışmalar mevcuttur [40–42], ve bu yöntem gerilim ölçekleme (Voltage scaling) olarak adlandırılmaktadır. Ancak bu yöntemin uygulanması, örneğin bir DRAM bit hücresine düşük besleme gerilimiyle erişim yapılması, bu erişimin gecikmesine neden olmaktadır. Bu tez kapsamında, DRAM'ler için uyarlamalı olarak gerilim ölçekleme yapabilen tasarımlar önerilmektedir. Ayrıca, FPGA'ler için de benzer prensiple, gerilim ölçeklemesi yapılan, bu sayede düşük güç tüketimi sağlanan, ancak bunu yaparken de güvenilirliği azaltmayan özgün tasarımlar önerilmiştir.

2.5 DRAM için Saklama Zamanı ve Yenileme

DRAM hücrelerinde kapasitörde saklanan veri sızdırma akımları nedeniyle erişim transistörü üzerinden boşalmaktadır. Bir süre sonra ise eğer erişim sağlanmamış olursa içinde sakladığı veriyi kaybeder. Veri kayıplarını önlemek için DRAM bit hücrelerine belirli bir zamanda bir erişim yapılması gerekmektedir. Bir DRAM hücresi için veri kaybı yaşanmaksızın içinde tutulan veriyi saklayabileceği en uzun zamana "saklama zamanı" (retention time) denilir. Bu saklama zamanından önce DRAM hücresine erişilmesi gerekir, bu işleme de yenileme (refresh) denilir.

DRAM hücreleri üretim sürecinden gelen farklılıklar içermektedir fiziksel olarak, ve bu farklılıklar nedeniyle her bit hücresinin saklama zamanı birbirlerinden farklılaşabilmektedir [43]. Bu nedenle DRAM üreticileri tarafından oldukça güvenli tarafta kalacak şekilde, tüm DRAM bit hücreleri için ortak bir yenileme sıklığı ve bunun tersi oranda yenileme zamanı belirlenir. Bu ortak yenileme zamanı (refresh period) için yaygın kullanılan bir değer 64 ms'dir [44]. Ancak bu değer bit hücrelerinin büyük çoğunluğu açısından gereksiz yenileme anlamına gelmektedir.

DRAM hücrelerinin yenilenmesi için her erişim demek aslında o sırada okuma ve yazma işlemlerinin yapılamması anlamına gelmektedir. Dolayısıyla DRAM'in asıl işlemlerinin yenileme sırasında bekletilmesi gerekmektedir [29, 45]. Bu çakışma başarım açısından kayıp anlamına gelmektedir. Üstelik her erişim işlemi güç tüketimi anlamına da gelmektedir. Özetle, mevcut hazır raf ürünü DRAMlerde DRAM bit hücrelerinin büyük bir kısmı için gerekmediği halde yapılan erişim nedeniyle istenmeyen güç tüketimi ve başarım düşüşü gerçekleşmektedir. Üstelik sıcaklıkla sızdırma akımları arttığı için, belirli bir sıcaklık limitinin üzerine çıkıldığında yenileme sıklığı (refresh frequency) 64 ms'den 32 ms'ye çekilmektedir, bu da bit hücrelerinin

yarı zamanda bir yenilenmeleri anlamına gelmektedir. Ve gereksiz başarım kaybı ve güç tüketimi daha da artmaktadır.

DRAM hücrelerinin yenileme sıklığını tüm hücreler için aynı yapmak verimsizdir. Bunu iyileştirmek için statik çözümler önerilmiştir. Örneğin statik olarak DRAM hücrelerinin saklama zamanları çıkartılıp profillenmesi ve buna göre yenileme yapılması gibi çözümler denenebilir. Ancak yenileme zamanı sadece fabrika seviyesindeki farklılıklarla değişmez bit hücreleri arasında. Aynı zamanda bit hücrelerinin içinde sakladığı veriye göre birbirleriyle, bit telleri ile veya hücreler arasında çapraz etkileşime girerek de farklılıklar doğabilmektedir. Bu yüzden dinamik profilleme yapılması gerekmektedir, ve bu kapsamda yapılan bir çok çalışma bulunmaktadır [8, 25, 46–49]. Örneğin, [48] çalışmasında her satır için bit hücrelerinin saklama zamanları profillenerek satır bazında yenileme sıklığının yapılması önerilmektedir, buna göre belirli yenileme zamanı sınıfları oluşturulur. O sınıfa giren satırlar belirlenir, ve bundan sonra artık hangi satır hangi yenileme sıklığı sınıfındaysa ona göre yenileme yapılır.

İşte bu tezde; doktora çalışmaları kapsamında geliştirilen bit hücrelerinin yenileme sıklığı ihtiyacını azaltabilen özgün uyarlamalı DRAM tasarımları anlatılmaktadır. Bu tasarımlar; bit hücrelerinin erişim zamanlarına, veya yenileme zamanı profillerine veya sıcaklığa bağlı olacak şekilde gerilim ölçekleme ve alttaş kutuplama yapabilmektedir. Geliştirilen bu uyarlamalı devre tasarımları sayesinde, hem düşük güç tüketimi hem de yüksek başarım sağlanabilmektedir. Ayrıca, tez kapsamında tüm bu DRAM tasarımlarının denenebileceği (daha önce bahsedilen zorluklara rağmen) bir temel (anaçizgi) tasarım kurulması başarılmıştır. Bu temel tasarım tam olarak 80-85 °C sıcaklıktan daha düşük sıcaklıklarda 64 ms ve daha yüksek sıcaklıklarda 32 ms'de yenileme yapılacak şekilde tasarlanabilmiştir (Devre analizleri için Cadence tasarım platformu ve Analog Design Environment analiz ortamı kullanılmıştır).

2.6 FPGA-tabanlı Evrişimsel Sinir Ağları Hızlandırıcı için Gerilim Düşürme

Otonom sürücüsüz bir araç veya savunma sanayiine yönelik gömülü bir sistem içerisinde veya bunlar gibi bağımsız çok çeşitli kullanım alanında görüntü işleme ve yapay zeka uygulamaları kullanılmaktadır. Görüntü/video/örüntü işleme uygulamalarının yararlandığı en temel araç ise, derin öğrenme ve yapay sinir ağlarıdır. Derin yapay sinir ağları'nın en yaygın temsilcisi de "Evrişimsel Yapay Sinir Ağları" (Convolutional Neural Networks, CNNs) algoritmalarıdır. Tez içerisinde sıklıkla CNNs veya CNN olarak anılacaktır.

CNN algoritmalarının temeli evrişimlerdir (convolution). Aslında her bir

evrişim/convolution bir filtre (kernel) anlamına gelmektedir. Bir input (örneğin bir resim) farklı filtrelerden geçirilerek evrişimler hesaplanır, filtreler dediğimiz ise temelde matrislerdir ve matris olarak alınan farklı boyuttaki bir girdi bu filtre olan matrislerle çarpılır, özetle çok sayıda matris çarpımı yapılmaktadır. Daha sonra bu matrisler üzerinden alt örnekleme (subsampling) veya havuzlama (pooling) gibi işlemler yapılarak matris boyutları küçültülür. Sonra tekrar farklı filtrelerden geçirilerek yeni evrişimler oluşturulur. Ağ mimarisine bağlı bu tekrarlı işlemlerin (layers) ardından en sonunda tam bağlı (fully connected) katmanına ulaşılır. Bu katmanda tüm matrisler tek sütundan oluşan bir vektör haline getirilir. Böylece artık sınıflandırma işlemi yapılabilir. Bu işlem ile CNN algoritmalarında (klasik bir CNN uygulamasında) olasılık hesabıyla farklı sınıfların değerleri, olasılıkları, çıkartılır. Buna göre tespit edilen olasılıkla girdinin ne olduğu söylenir. Çok temel seviyede CNN algoritması katmanları ve işlevsel akışı bu şekilde özetlenebilir. Farklı CNN algoritma çeşitleri bulunmaktadır; bu algoritmalardan googlenet, vggnet ve resnet bu çalışmada kullanılan tekniğin bilinen en ilerisi denek taşlarıdır (benchmarks) [21, 22, 50].

CNN algoritmalarında aslında en sık yapılan işlemler matris çarpımları ve bellek erişim işlemleridir. Seçilen CNN mimarisine göre hangi katmanda ne işlemlerin yapılacağı da belirlidir. İşte bu karakterizasyon kullanılarak CNN algoritmalarının hızlandırılması işlemi gerçekleştirilebilir, ve bu sayede güç tüketiminin düşürülmesi ve başarımın artırılması hedeflenir. CNN algoritmalarını hızlandırmak için farklı donanım tipleri kullanılabilmektedir, temel CNN hızlandırıcı dononanımları (CNN Accelerators): Grafik işlem birimleri (GPUs) [51], Uygulamaya Özgü Tümleşik Tasarım Devreleri (ASICs) [52], ve Alanda Programlanabilir Kapı Dizini (FPGAs) olarak örneklendirilebilir [53, 54].

Güç tüketimi açısından en ileri CNN hızlandırıcı donanımları ASIC'lerdir. Ancak bu donanımlar uygulamaya özel olarak tasarlanmaktadır, ve CNN uygulamaları çok farklı alanlarda kullanılabildiği için kullanım çeşitliliği açısından esneklik sağlamamaktadırlar. Bu anlamda, kullanım esnekliğine en uygun donanım GPU'lardır, ancak GPU'ların da güç tüketimi maliyetleri fazladır. Güç tüketimi ve ölçeklenebilirlik açısından birlikte düşünüldüğünde en etkin çözüm FPGA hızlandırıcılardır [54, 55].

CNN hızlandırıcılar mobil cihazlar gibi batarya kritik sistemlerde de kullanılabilmektedir, otonom sürücüsüz araç sistemleri gibi doğruluk kritik sistemlerde de kullanılabilmektedir. Dolayısıyla güç tüketimi oldukça önceliklidir, ancak bunu sağlarken de başarıma ve güvenilirliğe dikkat edilmesi beklenmektedir. CNN hızlandırıcılar için SRAM ve DRAM ler için uygulanan düşük güç tüketimine yönelik transistör seviye çözümler oldukça etkin olacaktır [4, 5, 11, 56, 57], ancak

bu yöntemler transistör seviye tasarım değişikliği içerdiği için üretici tarafında gerçeklenebilirler. Bu kapsamda en etkin çözümler var olan donanım kabiliyetlerini kullanarak güç tüketiminin düşürülmesidir.

CNN hızlandırıcı donanımlarda ve özellikle FPGA'lerde, var olan donanım kabiliyetleri kullanılarak güç tüketimini azaltma tekniklerinin en temel ve yaygın kullanımdaki temsilcisi Gerilim Düşürme (undervolting, voltage scaling) yöntemidir. Son kullanıcı seviyesindeki hazır bir ürün için vendor/üretici tarafından güvenilir tarafta kalarak farklı hatlar için korunumlu besleme gerilimi değeri ayarlanır, ancak pratikte bu gerilimlerin altına düşülmesi mümkündür. İşte bu nominal değerin altındaki voltajlarda CNN hızlandırıcılar çalıştırıldığında daha düşük güç tüketir hale gelmektedir. Bu tezde FPGA tabanlı CNN hızlandırıcılar üzerine geliştirilen özgün gerilim düşürme tasarımları, yoğun ve kapsayıcı deneyler ve karakterizasyon çalışmaları sunulmaktadır.

3. ADRAM: UYARLAMALI DRAM TASARIMLARI

3.1 Amaç ve Motivasyon

DRAM'ler için en önemli problemlerden biri yenileme ihtiyacından kaynaklanmaktadır. Bit hücreleri saklama zamanlarının ardından tuttukları veriyi kaybederler. Bu yüzden de yenilenmeleri veya bit hücrelerine erişim yapılması gerekmektedir. Normalde yazma ve okuma yapılan hücreler için erişim ihtiyacı giderilmiş olur. Fakat tüm hücrelere okuma ve yazma yapılmadığı için kalan hücrelerin yenilenmesi gerekir. İşin kötü yanı bit hücreleri yenilenirken diğer faydalı işlemler bekletilmiş olur, yenileme erişimleri ile okume veya yazma erişimleri çakışmış olur. Bu başarımı kötü yönde etkiler ve iyileştirilmesi gerekir. Üstelik önceki bölümlerde de bahsedildiği üzere her erişim demek okuma veya yazma olmadan güç tüketmek anlamına gelmektedir. Üstelik veriyi kaybetmenin asıl sebebi olan sızdırma akımları demek, aslında durağan güç tüketimi demektir. Tüm bunlar DRAM açısından yenilemenin ne kadar maliyetli olduğunu göstermektedir.

DRAM'lerde bit hücreleri fabrika seviyesinde üretim kaynaklı farklılıklar içermektedir. Örneğin bir transistörün kanal parametreleri bit hücresinden bit hücresine farklılık gösterir, ve bu kanal parametrelerinin, Bölüm 2.3 içerisinde anlatılmaktadır, sızdırma akımlarını etkilediği bilinmektedir. Üstelik DRAM bit hücreleri ve bit tellerinin çapraz etkileşimleri yüzünden de içerde tutulan veriler ve sızdırma akımları değişebilmektedir. Bu nedenle, DRAM bit hücrelerinin statik olarak üretildikten sonra profillenmesi ve buna göre yenileme sıklıklarının belirlenmesi problemlere neden olacaktır, ve DRAM üreticileri standartlar gereği tüm hücreler için korunumlu bir DRAM sıklığı belirler ve bu sıklığı da sıcaklıkla tekrar artırırlar. DRAM bit hücrelerinin çoğunluğu için bunun gereksiz olduğundan bahsedilmişti, bu yüzden de zaten okuma ve yazma gibi bir işlem yapılmayan erişimler üstüne üstelik birçok hücre için gerekmediği halde yapılır hale gelmektedir. Bunun iyileştirilmesi için dinamik profilleme yaparak yenileme periyodunun belirlenmesine yönelik birçok çalışma bulunmaktadır (örn:[48]), ancak bu çalışmalar devre seviyesinde en azından parametrik bir değişim bile uygulamamaktadırlar ve ayrıca problemin asıl sebebi olan bit hücrelerindeki yenileme ihtiyacını iyileştirmeye odaklanmamaktadırlar, sadece mimari seviye çözümlerler, var olan problemin azaltılmadan mevcut uygulamadaki verimsizlikleri iyileştirmiş olurlar.

Doktora çalışmaları kapsamında, bilindiği kadarıyla ilk kez, DRAM'lerin temel

problemi olan yenileme ihtiyacını azaltmak için DRAM'in belirlenen bir içeriğe uyarlamalı olarak kendi kendine alttaş kutuplama ve gerilim ölçekleme yapabildiği DRAM tasarımları önerilmiştir (bir kısmı için; tescil almış veya hazırlanacak, yayınlanmış veya hazırlanacak patent ve çıktılar girişte sunulmuştur.). Bu tasarımlara ortak olarak ADRAM (Adaptif DRAM) adını verdik. ADRAM tasarımlarının motivasyon kaynağı olan şu üç problem, ADRAM tasarımları sayesinde çözülmüş olmaktadır:

- Sızdırma akımları azaltılarak veya gerilim ölçeklemesi uyarlamalı hale getirilerek yenileme zamanı uzatılmış olur, böylece daha az sıklıkla yenileme yeter hale getirilmiştir, ve toplam yenileme için erişim sayısı azaltılmış olur.
- Yenileme ihtiyacı azaltıldığı için yenileme işlemleri sırasında bekleyen okuma ve yazma işlemleri çakışmaları azaltılmış olur, böylece başarım artışı sağlanmıştır.
- Sızdırma akımları azaltıldığı tasarımlarda durağan enerji kayıpları azaltılmış olur.
- Toplam yenileme sayısı azalması ile; **başarılan toplam yenileme sayısındaki düşüş * yenileme için harcanan güç** kadar güç tüketiminde tasarruf elde edilmiş olur.
- Güvenilirlik iyileştirilmiş olur.

Doktora çalışmaları kapsamında 4 özgün uyarlamalı DRAM tasarım fikri tarafımca önerilmiştir ve tasarımları yoğun benzetimlere dayanmaktadır (farklı özgün fikir ve tasarımlar üzerine de çalışılmaktadır, bu tez kapsamında önerilen bu fikir ve uyarlamalı yapı, ve üstelik bu tasarımların denenebileceği ortam kurulması birçok akademik çalışma için katkı sağlamış ve kapı açmıştır.). Bu ADRAM tasarımları (birer cümle tanıtımlarıyla) şunlardır, her biri ilerleyen alt bölümlerde detaylı anlatılmakta ve sonuçları sunulmaktadır:

- **CADRAM (Hücre içeriği Uyarlamalı DRAM):** Bİt hücresinde tutulan içeriğe uyarlamalı devre parametresini kendi kendine değiştirebilen özgün DRAM tasarımıdır.
- **PADRAM (Üretim farklılığı Uyarlamalı DRAM):** DRAM satırlarının saklama zamanı kabiliyetleri sınıflandırılır, ve bu sınıflandırmaya bağlı olarak DRAM o satırın devre parametresini kendi kendine değiştirebilir, bu özgün tasarımı PADRAM olarak adlandırdım.

- **TADRAM (Sıcaklık Uyarlamalı DRAM):** DRAM'lerde sıcaklık bilgisine göre DRAM'in tüm hücreleri için (DRAM dizini veya DRAM alt dizini) devre parametresinin kendi kendine değiştirilebildiği özgün DRAM tasarımıdır.

- **AADRAM (Erişim Uyarlamalı DRAM):** DRAM'lerde, satırlara erişim örüntüsüne uyarlamalı olarak, o satırdaki hücrelerin devre parametrelerinin kendi kendine değiştirilebildiği özgün DRAM tasarımıdır.

Önerilen tasarımlarda kullanılan yöntem yani karar mekanizması karar aldıktan sonra devre parametresinin değiştirilmesi tarafı (devreleri), TADRAM, PADRAM ve AADRAM için mümkün olduğunca benzerdir, uygulanacak birim; hücre, satır veya dizin, değişmektedir. Bu sayede hibrit uygulanabilmeleri veya uygulamaya göre birden fazla gerçekleme ile farklı durumlara göre birlikte veya ayrı çalışabilirlerdir.

3.2 Metodoloji

Bu kısımda özet olarak kullanılan tasarım ve benzetim ortamları anlatılmaktadır, tasarım ve benzetim araçları için sonuçların alındığı ayarlanan konfigurasyon da yine bu bölümde sunulmaktadır. Oluşturulan devre, DRAM DDR3 standardını sağlayacak şekilde tasarlanmıştır [7, 44]. Mimari ve devre benzetim ortamları diğer araştırmacılar tarafından da ulaşılabilirdir, ve bu ortamlarda sonuçları almak için atanan konfigurasyonlar ve metodoloji takip edilerek benzer sonuçların alınması amaçlanmaktadır. Bu kısımda anlatılan metodoloji aksi belirtilmedikçe tüm ADRAM tasarımları için ortaklanmıştır. Sonuçlar ve değerlendirmesi ise her tasarım için farklıdır ve ayrıca ileriki bölümlerde sunulmaktadır.

ADRAM tasarımları için devre seviyesi tasarım ve benzetimler, model tabanlı tüm donanım için benzetimler ve mimari seviye benzetimler gerçekleştirilmiştir. Devre seviyesi tasarımlar Cadence platformu üzerinden 45nm CMOS (UMC tasarım kütüphanesi) teknolojisi ile gerçekleştirilmiştir, kullanılan temel Vdd değeri 1.4 V olarak alınmıştır. Bununla birlikte, daha düşük ve daha yüksek gerilimler uygulanarak dav gerilim ölçekleme yapılmıştır. Benzetimler için Analog Design Environment devre analiz aracı kullanılmış, bu şekilde devre seviyesinde tasarım ve analizler koşturularak sonuçlar elde edilebilmiştir. DRAM için DDR3 ve DDR4 DRAM standartlarını sağlayacak şekilde bir DRAM temel tasarımı ortaya konulmuştur [7, 44]. Karşılaştırmalarımızı rafta hazır ürün olan Micron 1GB DDR3 DRAM'e yapıyoruz, ve temel tasarımımız da buna karşılık gelecek şekildedir. Daha sonra bu ana çizgi tasarım üzerine bu tezde anlatılan farklı uyarlamalı DRAM tasarımları denenmiş ve benzetimleri koşturulmuştur. DRAM'ler birçok mimari yapıdan oluşmaktadır Bölüm

ref'de anlatıldığı üzere, ve tüm DRAM mimarisini devre seviyesinde tasarlamak mümkün değildir, bunun yerine devre seviyesinde yaptığım tasarımların sonuçlarına göre modeldeki hazır parametrelerin ilgili olanlarını değiştirip, donanıma yönelik bir üst seviye maliyet analizi yapmak gerekmektedir. Örneğin, alttaş kutuplama ile elde edilen saklama zamanının donanımın tümü için herkes tarafından erişilebilen bir modelle güç tüketiminin hesaplanması ve bunun güç tüketiminde sağladığı kazancın baseline tasarımla karşılaştırılması gibi. İşte bu amaçla, Cacti 6.5 ve 7 gecikme ve güç model tabanlı devre benzetim aracı kullanılmıştır. Devre seviyesindeki tasarımlar 45nm ile gerçekleştirildiği için, Cacti'da da seçilen modellerin 45nm teknolojide olmasına dikkat edilmiştir [58].

Mimari benzetimleri ise Ramulator olarak adlandırılan [59]; döngü doğrulukta DRAM benzetimleri sağlayabilen işlemci ve DRAM konfigurasyonlarını seçebildiğimiz ve tekniğin en ileri durumundaki denek taşlarını kullanarak sonuç elde edebildiğimiz platform üzerinden gerçekleştirilmiştir. CPU trace driven modunda sonuçlar alınmıştır. Erişim örüntüsü ve saklama zamanı profilleme işlemleri de bu ortama entegre edilmiştir. Tüm karşılaştırma, getiri götürü analizleri ve benzetimlerin gerçeklendiği konfigurasyon devam eden maddelerde sunulmaktadır.

- DRAM: DDR3 kullanılmıştır [7, 44].
- DRAM mimarisi: Her kanalda bir sınıf/rank, her sınıf'da ise 8 küme/bank bulunmaktadır. Bu kümeler, kendi içinde 64K satır içermektedir. Her bir satır ise 2KB boyuttadır.
- İşlemci: 8 çekirdekli, 4 GHz, 128 girişli komut pencereli, her çekirdek için kayıp oranı tutan birden fazla yazmaç
- Bellek kontrol birimi: İlk hazır olan ilk gelen ilk servis politikası

Saklama zamanı sınıflandırma için saklama zamanının normal dağılımla hücreler arasında dağılım gösterdiğini varsayıyoruz [43]. Simulator/benzetimci tarafından sağlanan denek taşlarını (benchmarks) kullanıyoruz, ve karşılaştırmaları her biri için sonuç alarak sağlıyoruz. Tez kapsamında önerilen erişim uyarlamalı DRAM ve üretim farklılığı uyarlamalı DRAM tasarımlarını temel çizgi tasarım'ı ile bu mimari benzetim aracı kullanarak karşılaştırıyoruz. Ayrıca, bu araca satırların erişim örüntülerini kaydeden FIFO (ilk giren ilk çıkar) mantığıyla çalışan tablolar ekliyoruz. Son olarak, literatürdeki tekniğin bilinen en iyi durumunu yansıtan Raidr çalışmasını da karşılaştırma için gerçekledik ve mimari karşılaştırmalarımıza bunu da ekledik. Bu çalışma dinamik profilleme yapan ama donanımsal değişiklik yapmadığı için asıl probleme çözüm sağlamayan bir DRAM çözümüdür, ancak dinamik profilleme ile

toplam yenileme sayısında önemli ölçüde düşüş sağlayabilen en bilinen çözümlerden birisidir. Tez kapsamında önerilen tasarımlar sadece temel DRAM tasarımıyla değil, bu çözümle de karşılaştırılmaktadır ve sonuçları verilmektedir. Benzetimlerde ayrıca 25 °C ile 125 °C arasında sıcaklıklarda analizler koşturularak, en kötü koşullarda çözümün çalışmasına bakılmıştır. Zaten sıcaklık uyarlamalı DRAM tasarımında da bu değerlere göre tasarım yapılmaktadır. Özetle, gerçek bir DRAM erişim, okuma/yazma yapıyorken güç ve başarım ölçümü yapılıyor gibi bir deney tasarımı kurulması hedeflenmiştir.

3.3 Sıcaklık Uyarlamalı DRAM, TADRAM

3.3.1 TADRAM mimarisi ve devre tasarımı

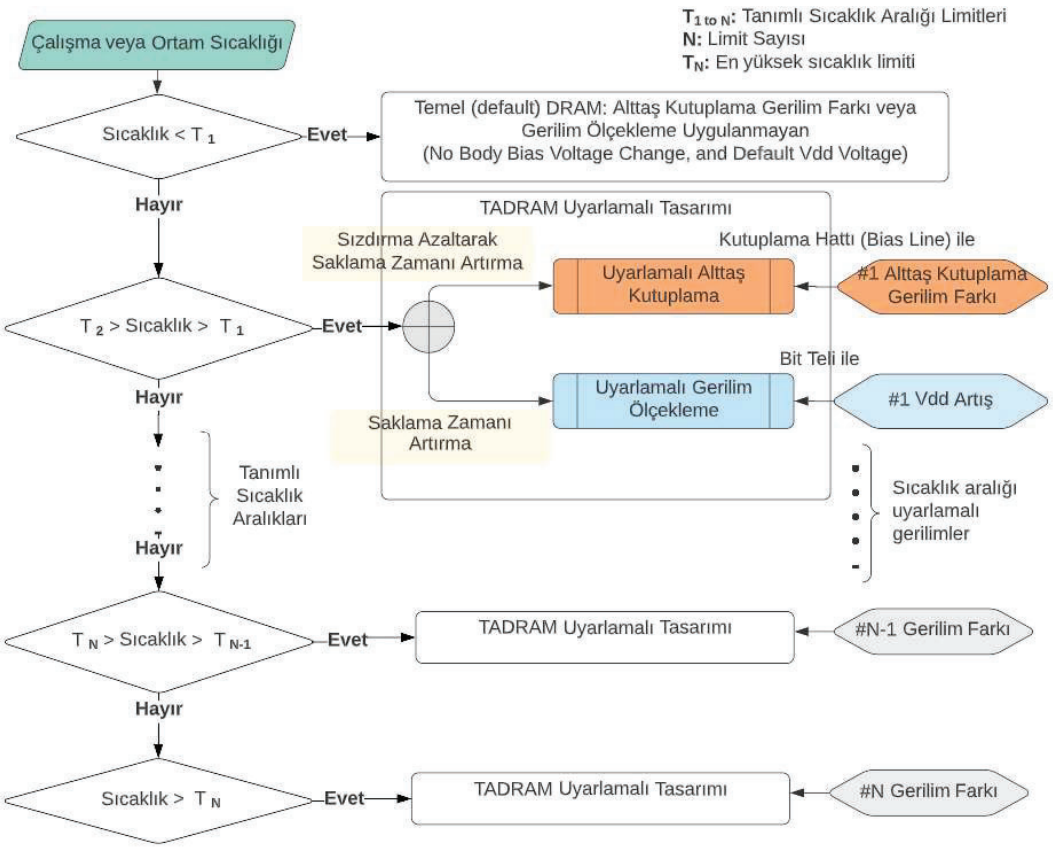
DRAM hücrelerinin sızdırma akımları artan sıcaklıkla arttığı için ve artık bazı hücreler daha sık yenilemeye ihtiyaç duyarlar, bu nedenle belirli bir sıcaklığın üzerinde DRAM üreticileri DRAM hücrelerinin tümü için uygulanan yenileme sıklığını arttırırlar (bu tezde referans alınan yaygın kullanımdaki DDR3 için 85 °C ve yukarısında 2 katına çıkarılmaktadır, 64 ms olan yenileme periyodu 32 ms'ye düşmektedir.). Zaten birçok hücre için gereksiz olan ve her açıdan maliyetli olan yenileme işlemlerinin sayısı daha da artmaktadır. Sıcaklık uyarlamalı DRAM tasarımı ise sıcaklık belirli bir limitin üzerine çıkarsa (veya DDR5 gibi farklı DRAM teknolojilerinde tek bir sıcaklık limiti değil, birden fazla limit belirlenmektedir) devre parametrelerini o durumda değiştirerek saklama zamanı artan DRAM hücrelerini kullanır. TADRAM olarak adlandırılan bu tasarımın temeli olan bu DRAM hücreleri sayesinde bir hücre en düşük saklama zamanına sahip olsa bile ve en kötü (en yüksek sıcaklık) sıcaklık durumunda çalışırken, yenileme sıklığının artmadığı durumda veri kaybının olmayacağını garanti eder. Böylece TADRAM sayesinde sıcaklığa bağlı artacak yenileme sayısı kadar güç ve başarım kazancı sağlanmış olur.

TADRAM iki şekilde (veya hibrit uygulamalı) gerçeklenir:

1. Alttaş kutuplama yöntemini uyarlamalı uygulayan tasarım,
2. Besleme gerilimlerini uyarlamalı uygulayan tasarım.

Her iki yöntemde de karar mekanizması aynıdır, sıcaklık verisi tüm DRAM hücreleri veya satırlar veya belirli satır grupları için geldiğinde bu veriye uyarlamalı olarak besleme gerilimi ve/veya alttaş kutuplama gerilimi uygulanır. Bu sayede sıcaklık yükseldikçe saklama zamanı kabiliyeti düşen hücrelerin saklama zamanlarının artırılması sağlanır. Asıl amaç ise bu artış sayesinde saklama zamanı diğer hücrelere

göre düşük olan bit hücrelerinin sıcaklık yükseldikçe daha sık yenilenme ihtiyaçlarını ortadan kaldırmaktır. Her iki yöntemle gerçeklenen farklı TADRAM tasarımları için işlevsel akış Şekil 3.1'de yer almaktadır.



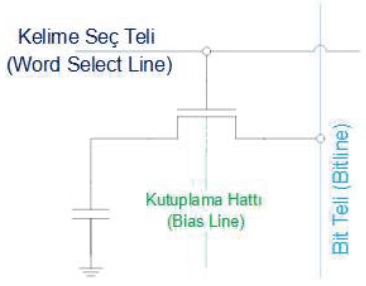
Şekil 3.1 : Farklı gerçekleme yöntemleri için TADRAM işlevsel akışı.

Şekil 3.1'e göre; önce DRAM için çalışma sıcaklığı (iç duyargaları sayesinde) veya DRAM'e dışardan gelen (bellek kontrolcü birimi üzerinden dış duyarga bilgileriyle) ortam sıcaklığı bilgisi alınır. Bu bilgi sınıflandırılmış olarak gelmişse direkt buna göre devre parametreleri uyarlamalı ayarlanır. Doğrudan sıcaklık bilgisi gelmişse bu işlevsel akış takip edilir. Buna göre, önceden sıcaklık aralığı tanımlanmalıdır, bu eğer hazır ürün olan bir DRAM için uygulanacaksa bu durumda o DRAM için belirtilen specler dikkate alınır; yenileme sıklığı değişim sıcaklıklarına bakılır. Örneğin, birkaç kez bahsedilen DDR3 Micron 1 GB DRAM'i için DDR3 JEDEC standardı da takip edilerek 64 ms'de bir yenileme yapılırken, 85 °C üstünde yenileme sıklığı artırılarak 32 ms'de bir yenileme yapılır. TADRAM bu bilgiye göre tek bir limit belirler, Şekil 3.1 için N=1 olur, $T_1 = 85$ °C olur ve sadece T_1'e bakılır.

Eğer sıcaklık T_1'den düşükse bu durumda daha yüksek sıcaklık limiti zaten olmadığı için default durumda kalınır. Eğer sıcaklık T_1'den yüksekse bu durumda bakılır, daha yüksek sıcaklık limiti var mı diye, DDR3 durumu için olmadığı için, T_1'e göre uyarlama başlatılır. Eğer sızdırma azaltarak saklama zamanı artırmak odaklı

tasarım gerçeklenmişse, buna gör alttaş kutuplama yapılır (defaulttan farklı değer sürülür). Eğer gerilimle saklama zamanı artacak gerçeklenme varsa bu durumda da sıcaklık 85 °C üstüne çıktığında DRAM bit hücrelerine sürülen besleme gerilimlerini artırır. Burada önemli olan DRAM bit hücresi kapasitöründe tutulan yükü yüksek değere çekmektir ki, sızdırma devam etse de daha uzun zamanda veri kaybı olmadan saklasın. Eğer hazır ürün kullanıyor olsaydık ve bu ürün birden fazla sıcaklık kullanıyor olsaydı veya uygulamaya özgü DRAM tasarlanıyor olsaydı ve belirli sıcaklık limitleri var olsaydı bu durumda aynı şekilde akışa devam edilirdi. Daha yüksek sıcaklıklara çıkıldıkça daha yüksek besleme gerilimi veya daha uygun (yüksek veya düşük olması transistöre bağlıdır) kutuplama/bias gerilimi uygulanır. Bölümün devamında: Uyarlamalı alttaş kutuplama tabanlı ve uyarlamalı gerilim ölçekleme tabanlı TADRAM devre tasarımı anlatılmaktadır, hemen ardından da tasarımlarla ilgili sonuç ve değerlendirme sunulmaktadır.

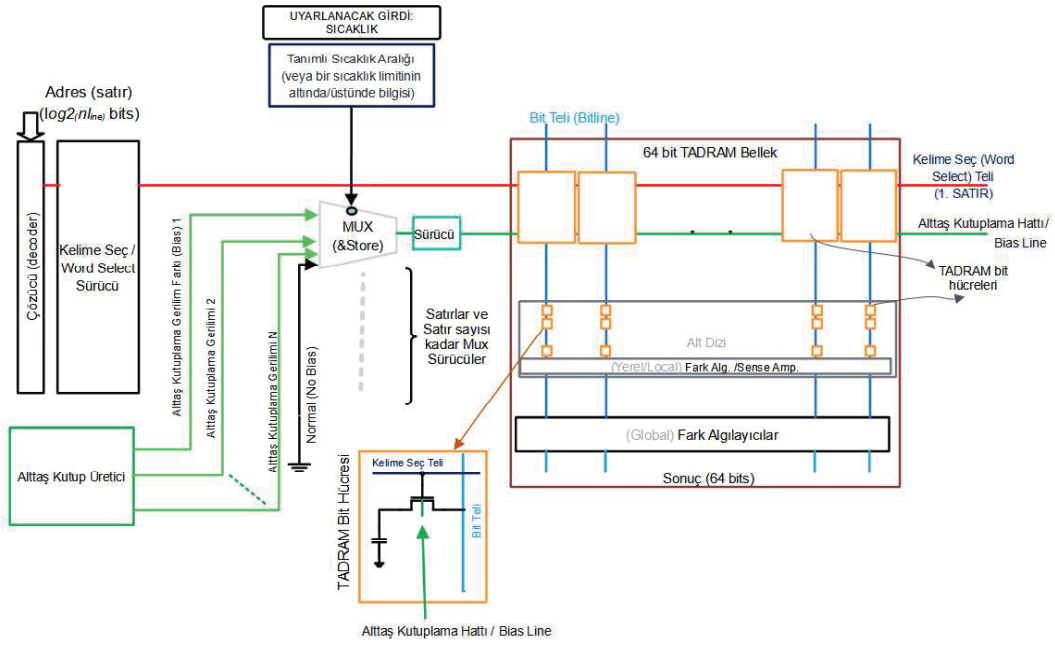
Sıcaklık Uyarlamalı DRAM (TADRAM) bit hücresinin gösterimi temelde DRAM bit hücresi ile çok benzerdir, Şekil 3.2'de sunulmaktadır. TADRAM bit hücresinin normal DRAM bit hücresinden tek farkı, alttaş kutuplama için kullanılacak alttaş veya body terminalinin doğrudan kaynak terminaline bağlı olmamasıdır. Bu terminal, (alttaş) kutuplama hattına (bias line) bağlıdır (Uyarlamalı gerilim ölçekleme tabanlı TADRAM bit hücresi için ise, hücre içinde herhangi bir eklenti ve ilave bir kutuplama hattı bulunmasına gerek yoktur. Uyarlamalı gerilim ölçeklemesi ise, var olan bit telleri sayesinde gerçekleştirilir). Her iki gerçekleme yöntemine dayalı TADRAM bit hücresinin, serim görüntüsü açısından DRAM bit hücresiyle (Şekil 2.3'de yer alan) bir farkı bulunmamaktadır. Sadece hücre başına bir kutuplama/bias hattı eklenmektedir. Temel DRAM tasarımında alttaşın toprağa bağlandığı varsayılmıştır (anlatımı yalınlaştırmak için). Çünkü doğrudan kaynak/source terminaline bağlandığı durumda yük olacağı ve bu yük de değişken olacağı için kutuplama/bias geriliminin sonuçlarını kıyaslamak mümkün olmayacaktır. Ancak pratikte NMOS için negatif kutuplama/bias veya PMOS için besleme gerilimi üstü bir kutuplama/bias her durumda kutuplamayı sağlayacaktır, ve bu sadece sıcaklık belirli limitin üstüne çıktığında devreye girmektedir.



Şekil 3.2 : Sıcaklık Uyarlamalı DRAM (TADRAM) bit hücresi.

Sadece yüksek sıcaklıklarda limitlere göre kutuplama/bias gerilimi uygulanacağı için bu gerilimin en yüksek olduğu tasarım kurgusunda bile TADRAM, sıcaklıkla yenileme sayısını artırmakla kaybedilecek güç tüketimiyle kıyaslanmayacak şekilde güç tüketimi oluşturacaktır. Ayrıca, sızdırma da azaldığı için durağan güç tüketimini de azaltmaktadır, devam eden kısımda kazanç oranları belirtilmektedir.

TADRAM dizi yapısı, Şekil 3.2'de gösterilen, bit hücrelerinden oluşmaktadır, ve tüm hücrelerin alttaş kutuplaması sıcaklığa uyarlamalı olarak (alttaş) kutuplama hatları (bias line) üzerinden gerçekleştirilir. TADRAM küme/bank yapısının kavramsal tasarım temsili gösterimi Şekil 3.3 ile sunulmaktadır. DRAM dizisi ve çevre devreleri aynen kullanılır, farklı olarak ise kelime seç (word select) sürücüler gibi, kutuplama/bias sürücüleri vardır kutuplama hatlarını sürmek için ve TADRAM'a girdi olarak gelen sıcaklık verisine (sıcaklık doğrudan gelmeyip, sıcaklık aralığı bellek kontrolcü birimi tarafından da gönderilebilir) bağlı olarak hangi kutuplama/bias gerilimini uygulayanacağını seçen Çoklayıcı/Seçici (MUX) birimi kullanılmaktadır. DDR3 için sıcaklık verisi; "sıcaklık 85 °C altında" veya "üstünde" şeklinde bir bitlik veridir. Sıcaklık 85 °C altında ise, kutuplama/bias gerilimi default duruma yakın hale getirilir (veya erişim gecikmelerini azaltacak seviyeye de getirilebilir, böylece ileriki çalışmalarda hedeflenen başarım artışı da sağlanmış olur.). Sıcaklık 85 °C üstünde ise (veya gelen bir bit veriye göre sadece), tüm DRAM hücrelerine kutuplama/bias gerilimi uygulanır. Şekil 3.3'de turuncu blokların herbiri bit hücresidir. Yeşil çizgiler kutuplama hatlarıdır. DRAM'den farklı olarak, TADRAM'da bir de, alttaş kutup üretici bloğu yer alır.

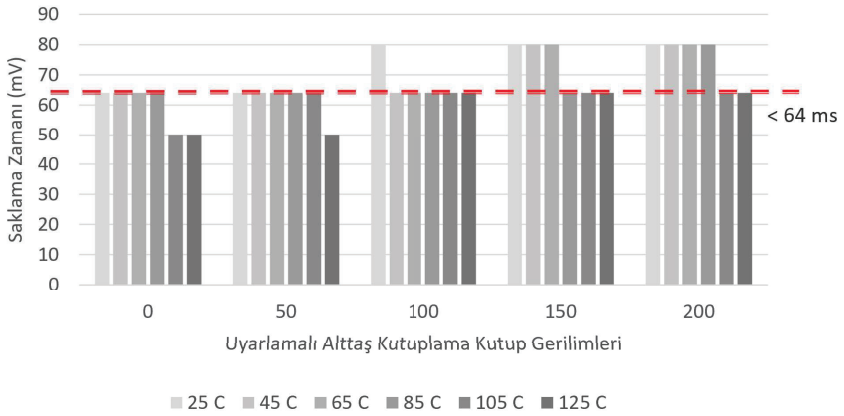


Şekil 3.3 : TADRAM küme/bank yapısı temsili gösterimi.

Kutuplama hatlarına çoklayıcı tarafından seçilen gerilim seviyesi bu kutup üretici tarafından üretilir [60], yada bu üretici yerine dış bir arayüzden kutuplama gerilim beslemesi de yapılabilir. Farklı DRAM küme/bank'larına tüm DRAM mimarisi için ortak kutuplama hattı uygulanabilir. Bu tasarımda kutuplama/bias tellerinin satır satır sürüleceği bir tasarımın temsili görseli bulunmaktadır. DRAM'de belirli satırların olduğu bölgelerin (alt dizi olarak bölünmüşse bunların) sıcaklığına uyarlama yapmak daha etkin sonuç verecektir. Ancak böyle bir ısı dağılımı zamanla değişkense ve farklı satır veya alt diziler için sıcaklık bilgisi alınamıyorsa bu durumda tüm DRAM dizini hücreleri için ortak sıcaklık uyarlamalı alttaş kutuplaması yapılması önerilmektedir. Bu durumda telleri tek tek sürme maliyeti de azaltılmış olacaktır. Kısaca, MUX tüm satırlar için ortaklanmış olacaktır, sadece kutuplama/bias hattı sürücüsünün tüm satırları sürecek büyüklükte tasarlanması gerekmektedir. Bu verimsiz olacaksa, veya ortak sürücü tasarımı daha çok alan maliyetine neden olacaksa, Mux üzerinden karar alındıktan sonra, satır gruplarına veya alt dizinlere farklı özdeş sürücülerle kutuplama yapılması da önerilmektedir.

3.3.2 TADRAM tasarımı benzetim ve analiz sonuçları

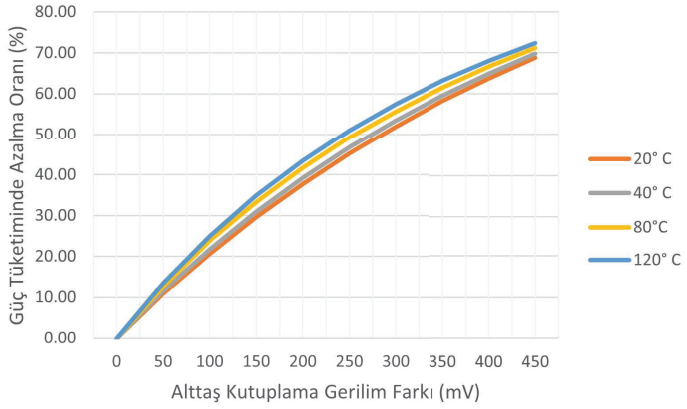
Farklı alttaş kutuplama gerilimleri için farklı çalışma sıcaklıklarının her birinde çoklu DRAM bit hücrelerinin herhangi birisi için saklama zamanının nasıl değiştiğini gösteren devre analiz sonuçları Şekil 3.4 ile verilmektedir. Bu sonuçlar için, 50 mV çözünürlükle, 6 farklı sıcaklıkta temel tasarımı 64 ms saklama (standarda ve referans tasarıma uygun olarak) zamanı olan DRAM bit hücreleri üzerine alttaş kutuplama yapılarak benzetimler iteratif olarak tekrar edilmiştir. Bit hücresinde; saklanan veri kaybı yaşandığı durumla, ilk kapasitörün dolduğu an (kapasitörün doğrudan üst terminali üzerinden ölçüm alınarak) arasındaki süre, saklama zamanı olarak hesaplanmıştır.



Şekil 3.4 : Farklı kutuplama gerilimlerinde saklama zamanı.

Saklama zamanı süresi, daha yüksek çözünürlükteki benzetim zamanı açısından beklenmiştir. Çünkü farklı sıcaklık ve gerilimlerle artık 64 ms noktası artmakta veya azalmaktadır. Şekil 3.4'deki sonuçlara göre; öncelikle gerçeklenen temel tasarımın referans alınan hazır ürünle uyumlu olduğu görülmektedir: 85 °C sıcaklık altından saklama zamanı 64 ms, üstünde ise bu zaman düşmektedir. TADRAM açısından ise sonuçlar şu şekildedir; 85 °C ve üzerinde düşen saklama zamanlarını tekrar 64 ms'ye çekebilen en düşük kutuplama/bias gerilimi 100 mV olarak gözükmektedir. Bu kutuplama/bias gerilimi en kötü sıcaklık koşulunu da sağlamaktadır. TADRAM'ın temel amacı artan sıcaklıkla daha sık yenileme yapılması ihtiyacını ortadan kaldırmaktır, ve bu örnek durum için başarılmış olur. Artık DDR3 bir DRAM için 85 °C üstünde sıcaklık artsa hatta en kötü sıcaklığa çıkılsa bile uyarlamalı kutuplama ile saklama zamanı 64 ms'de tutulması başarılmıştır.

Varsayalım ki, bir DDR3 DRAM de bir uygulama bir iterasyon çalışma süresi boyunca bellek okuma ve yazma işlemleri ve yenileme erişimleriyle birlikte çalıştığında toplamda N kere yenileme yapılmak zorunda olsun (N times refresh). Ve normalde 64 ms iken sıcaklık 85 °C üstüne çıktığında 32 ms olsun yenileme periyodu, standart bir DDR3de yenileme zamanı. Son olarak da, her bir yenileme işleminin maliyeti $P_{yenileme}$ olsun. Yüksek sıcaklıkta iterasyon döndüğünde yaklaşık kabaca 2N kere yenileme yapılmak zorunda olduğu anlaşılır, 85 °C üzerinde. TADRAM sayesinde ise 85 °C üzerinde saklama zamanı artırıldığı için (100 mV ile) sıklığın artırılmasına ihtiyaç kalmaz ve 85 °C üstünde devreye giren uyarlamalı alttaş kutuplama mekanizması sayesinde N kere yenileme yapılması yeterlidir. Böylece bu tip bir örnek durum için kaba bir hesapla, TADRAM N x $P_{yenileme}$ kadar güç kazancı sağlar denilir. Üstelik bu durumda N sayıda yenileme yapılmadığı için çakışma nedeniyle bekleyecek okuma ve yazma işlemleri de yapılabilir hale gelir, başarım da artar. Son olarak ise; TADRAM bit hücreleri artık daha az sızdırdıkları için durağan durumda güç tüketiminde de azalma olur ve bu azalma oranları Şekil 3.5 ile verilmektedir.

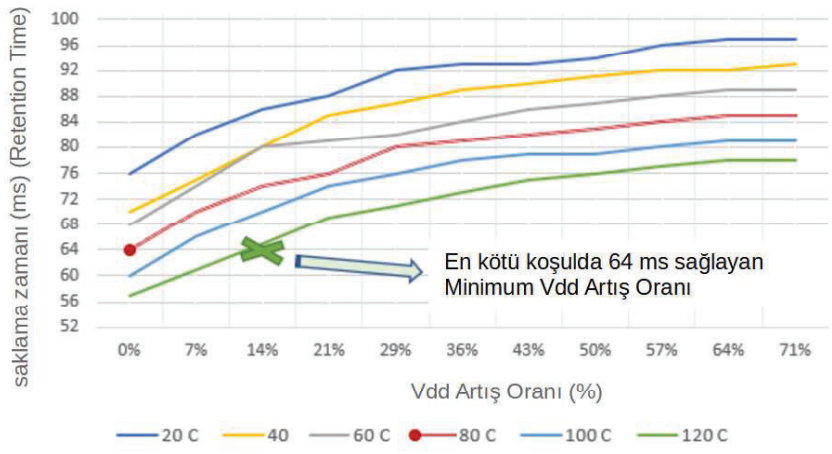


Şekil 3.5 : Alttaş kutuplamanın güç tüketimine etkisi.

Şekil 3.5'de durağan güç tüketimindeki temel tasarıma göre azalma oranları çok yakın çıktığı için orta sıcaklık değerleri gösterilmemektedir, onlar da aynı eğilimdedirler. Bu şekle göre; DDR3 için yenileme sayısının artmasını engelleyen bir tasarım sayesinde, durağan güç tüketiminde azalma oranı %20 seviyelerini bulabilmektedir.

Özetle uyarlamalı alttaş kutuplama ile TADRAM hem devingen hem de durağan güç tüketiminde kazanç sağlarken, aynı zamanda da başarım artar. Üstelik TADRAM'ın alan maliyeti de kabul edilebilir seviyededir. TADRAM için, uyarlamalı mekanizma bit hücresi başına kutuplama gerilim hattı, kutuplama/bias hattı, kadar küçük bir alan maliyetine neden olmaktadır. DRAM mimarisi açısından ise çoklayıcı, sürücü ve gerilim üretici eklemeleri bulunmaktadır, bunları da ilave bir öndoldurucu sürücü, öndoldurucu, ve sadece tek satırlık bir kodçözücü için bir alan maliyeti ile yaklaşık bir maliyet getireceğini varsayabiliriz. Ancak bu maliyet çok sayıda bit hücresi tarafından ortak maliyet olacaktır.

TADRAM için diğer bir gerçekleme yöntemi ise gerilim Ölçekleme'dir. Besleme gerilimlerinin sıcaklıkla uyarlamalı olarak DRAM hücrelerinin kendi kendisi tarafından değiştirilmesini sağlayan tasarıma ait sonuçlar diğer yöntemdeki sonuçlarla benzer methodla alınmıştır. Vdd değeri default tasarım için 1.4 V olarak kabul edilmiştir. Farklı sıcaklıkların her biri için farklı besleme gerilimlerinde saklama zamanı değişimi hesaplanarak sonuçlar elde edilmiştir. Bu tasarımın sonuçları alınırken farklı olarak, her bir test adımında ilgili besleme geriliminden ne kadar sürede veri kaybı noktasına geleceği gerilim izlenerek hesaplanmış ve saklama zamanı olarak kaydedilmiştir. İlgili sonuçlar Şekil 3.6 içinde yer almaktadır. Şekil 3.6'ya bakıldığında sıcaklıkla saklama zamanının aynı gerilimde arttığı görülebilmektedir. Ayrıca tüm sıcaklıklarda, gerilimin artmasıyla (Vdd artışıyla) beklendiği üzere saklama zamanının arttığı görülmektedir.

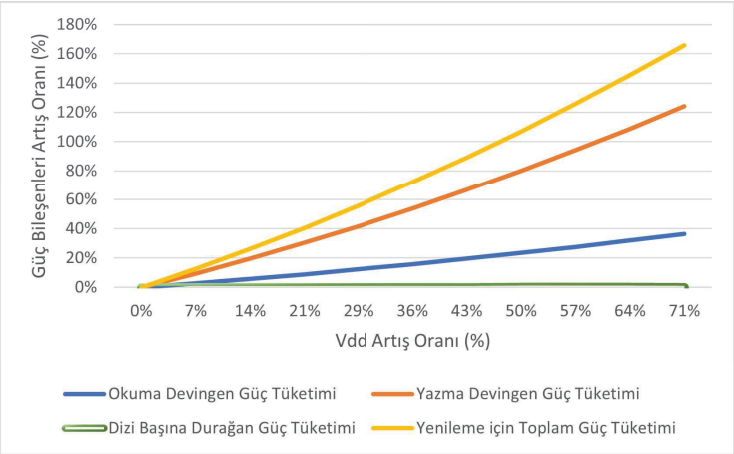


Şekil 3.6 : Farklı sıcaklıklarda, farklı gerilimler için saklama zamanı.

Sonuçlara bakıldığında sıcaklık artışına rağmen en kötü ortam sıcaklığında default yenileme periyodunu sağlayan Vdd artış oranı %14 olmaktadır. DDR3 için 85 °C üstünde yenileme periyodu 32 ms yapılmakta iken, %14'lük bir Vdd artışı (1.4 üzerinden yaklaşık 1.55-1.6 V) sayesinde artık 2N sayıda değil N sayıda yenileme yapılması yeterli olacağı için N sayıda yenileme kadar kazanç sağlanmış olacaktır.

İlk TADRAM gerçekleme yönteminden farklı olarak, bu tasarımda sızdırma akımları azaltılmadan saklama zamanı artırıldığı için ve gerilimin artırılması demek güç tüketimi açısından (dinamik güç tüketimi açısından karesiyke orantılı artar) artış anlamına geldiği için, N tane yenileme işlemi kadar güç tasarrufu yapılırken, Vdd'nin karesiyle orantılı olacak şekilde ise devingen güç tüketimi artar. Dolayısıyla devingen güç tüketimini üç açıdan ele almak gerekir: Yazma/Write işlemi güç tüketimi, okuma/read işlemi güç tüketimi ve refresh/yenileme için güç tüketimi. Bunlar devingen güç tüketimi bileşenleridir, ayrıca durağan güç tüketimi bileşenini de unutmamak gerekir.

Tüm bu güç bileşenleri açısından Vdd artış oranıyla güç bileşenlerindeki artış oranları Şekil 3.7 ile sunulmaktadır. Ancak dikkat edilmelidir ki bu artışlar sadece sıcaklık belirli bir seviyenin üstüne çıktığında uyarlamalı devre parametreleri değiştiğinde gerçekleşmektedir. Yenileme sayısının artmamasını sağlayan Vdd artışında yazma ve yenileme güç tüketimlerinde yaklaşık %20'ye yakın artış olmaktadır. Okuma güç tüketiminde ise, güç tüketim artışı %10'un altındadır. Devingen güç tüketimi açısından artışlar bu şekildedir, durağan güç tüketiminde ise nerdeyse yok denecek kadar az bir artış yaşanmaktadır.



Şekil 3.7 : Farklı besleme gerilimlerinde güç bileşenleri değişimi.

Diğer taraftan, toplam yenileme sayısı iki katına çıkacakken bu önlenmiş olduğu için mevcut yenileme sayısı kadar yaklaşık güç tüketimi kazancı da sağlanmış olur. Bunu da dahil ederek toplam güç tüketimi okuma, yazma ve yenileme sayısına bağlı değişecektir. Ancak şu şekilde bir varsayımla toplam güç tüketimindeki değişimi

yorumlayabiliriz: Diyelim ki, bir uygulama iterasyonunda; N kere okuma, N kere yazma ve N kere de refresh yapılacak olsun. Güçler ise; P_{read}, P_{write}, $P_{refresh}$ olsun. Ve çıkan sonuçlara göre, şekildeki sonuçlardan da görüleceği üzere, okuma güç tüketimi yazmanın 2.5-3 katı, refresh ise 3-3.5 katı olmaktadır. Durağan güç tüketimi ise epsilon olmaktadır. Bu durumda: Vdd artışına rağmen, TADRAM için toplam güç tüketimi (okuma, yazma ve yenileme güçleri toplamı) kabaca hesapla (Denklem 3.1) yaklaşık 8.3 birim okuma gücü kadardır ($P_{toplam} = 8.3xP_{read}$).

$$P_{TADRAM} = P_{read}x2.5x\frac{120}{100} + P_{read}x3.5x\frac{120}{100}x32/32 + P_{read}x1x\frac{110}{100} \tag{3.1}$$

Eğer TADRAM olmasaydı ise, DRAM için toplam güç tüketimi (okuma, yazma ve yenileme güçleri toplamı) kabaca hesapla (Denklem 3.1); yaklaşık 10.5 birim okuma gücü kadardır ($P_{toplam} == 10.5xP_{read}$).

$$P_{DRAM} = P_{read}x2.5x\frac{100}{100} + P_{read}x3.5x\frac{100}{100}x64/32 + P_{read}x1x\frac{100}{100} \tag{3.2}$$

Kısaca; TADRAM sayesinde, böyle bir örneklem (okuma, yazma ve yenilemenin eşit oranlarda yapıldığı durumda ve belirtilen artış oranlardaki kazançlarla) için yaklaşık %21 toplam güç tüketiminde kazanç sağlanmış olur. Üstelik, yenileme sayısının sıcaklıkla iki katına çıkmaması sağlanarak başarım artışı da sağlanmış olur.

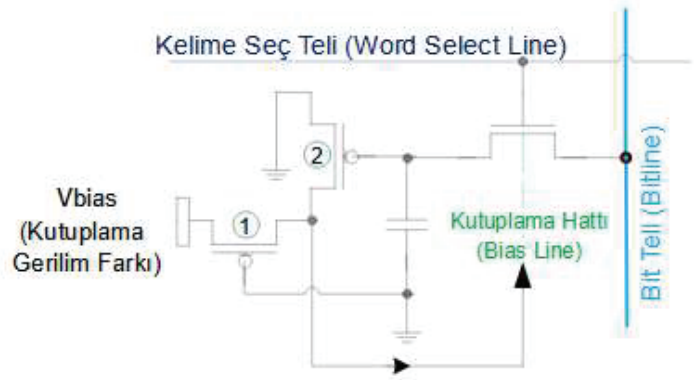
Alan maliyeti için ise, ilk gerçekleme yönteminden farklı olarak gerilim ölçekleme yöntemiyle gerçeklenen TADRAM bit hücresine ekstra bir kutuplama hattı da çekmeye gerek yoktur. Bunu süren sürücü veya seçici ihtiyacı da bulunmamaktadır. Kutuplama/bias gerilimi oluşturma ve gerilim üretici ile ilgili alan kayıpları da bulunmamaktadır. Uygulama açısından ve alan maliyeti açısından uyarlamalı gerilim ölçekleme tabanlı TADRAM tercih edilebilirdir, ancak güç tüketiminin daha da azalması isteniyorsa ve özellikle gelişen transistör teknolojisiyle durağan güç tüketiminin oranı daha da arttığı için ise uyarlamalı alttaş kutuplama yöntemi tercih edilebilir.

Özetle, her iki gerçekleme yönteminde de TADRAM, sıcaklık arttığında uyarlamalı olarak devreye girer, ve bu sayede hücrelerin saklama zamanlarını artırarak sıcaklıkla daha çok yenilenme ihtiyacını kaldırır. Bu sayede herhangi bir DRAM tasarımında bir sıcaklık limitinden sonra yapılacak yenileme sayısı artışı da önlenebilir olmaktadır. Alan maliyeti açısından kritiklik söz konusuysa ise ikinci gerçekleme yönteminin yok sayılabilecek bir alan maliyeti vardır. Gecikme maliyetleri ise %10'un altında olmaktadır, ancak çakışma çözüldüğü için başarım gecikme kayıplarına rağmen artış göstermektedir.

3.4 Hücre İçeriği Uyarlamalı DRAM, CADRAM

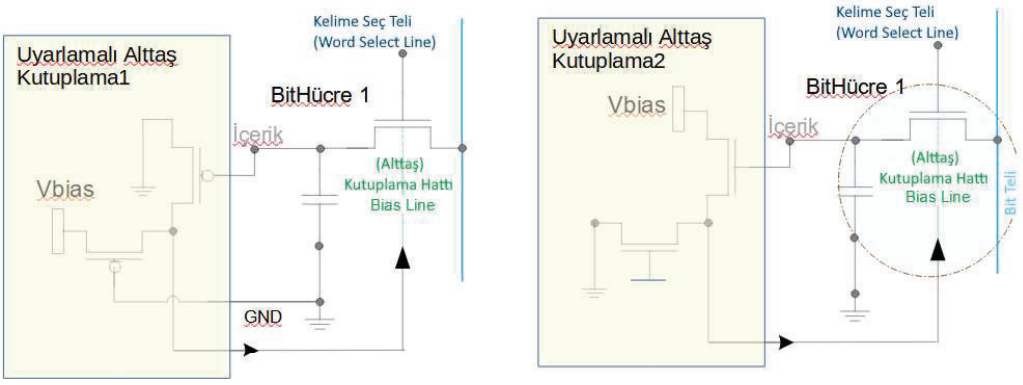
3.4.1 CADRAM mimarisi ve devre tasarımı

Bit hücresi içeriği uyarlamalı DRAM, CADRAM, tasarımında bit hücresindeki erişim transistörüne hücrede tutulan veriye göre alttaş kutuplama yapılır veya normal (kutuplama/bias olmadan) halde bırakılır. Önerdiğimiz bu özgün DRAM tasarımı sayesinde, eşik değer voltajı artırılarak, sızdırma akımlarının azaltılması ve bit hücresinin sakladığı veriği daha uzun süre koruyabilmesi amaçlanmaktadır. CADRAM bit hücresinde normal bir DRAM bit hücresine ilave olarak; 2 geçiş transistörü ve kutuplama hattı eklenir. Bu geçiş transistörlerinin girişleri hücrede tutulan veriye bağlıdır, bu transistörlerin farklı uçlarında biri kutuplama/bias gerilimine biri ise toprağa bağlıdır, diğer uçları ise ortaktır. Bu ortak uç ise kutuplama hattı üzerinden bit hücresinin alttaş terminaline bağlanarak, hücre içinde saklanan veriye bağlı olarak alttaş kutuplama yapabilen DRAM tasarımı sağlanmış olur. Önerdiğimiz hücre içeriği uyarlamalı DRAM tasarımına ait kavramsal görünüm Şekil 3.8'de sunulmaktadır. Bu fikir, farklı tasarımlara da uyarlanabilirdir.



Şekil 3.8 : Hücre içeriği uyarlamalı DRAM (CADRAM) bit hücresi.

Bu tasarımın serim tasarımı veya üretime yönelik tasarımı farklılaşabilecektir. Fikrin özü; kendi tuttuğu bit değerine göre alttaş kutuplama yaparak sızdırma akımlarını azaltabilen DRAM tasarımıdır. Bu tez kapsamında fikrin kavramsal tasarımı sunulmaktadır (as a proof of concept). Bu tasarımın farklı yöntemlerle de gerçeklenmesi mümkündür; Şekil 3.9 buna örnek vermek için konulmuştur. Kutuplama/bias gerilimini aktarmak için kullanılan geçiş transistörleri gerçekte üretim için daha elverişli ise o şekilde seçilir, veya parametreleri veya sayısı değişir. Hatta tez kapsamında yapılan benzetimlerde kutuplama/bias gerilimleri doğrudan sinyal üretecine bağlandığı durumlar da olmuştur, gerçekte ise bunu üreten bir üretici de olacaktır. Veya DRAM tasarımın kendisi de değişebilir, örneğin farklı bir erişim transistörü de eklenebilir.

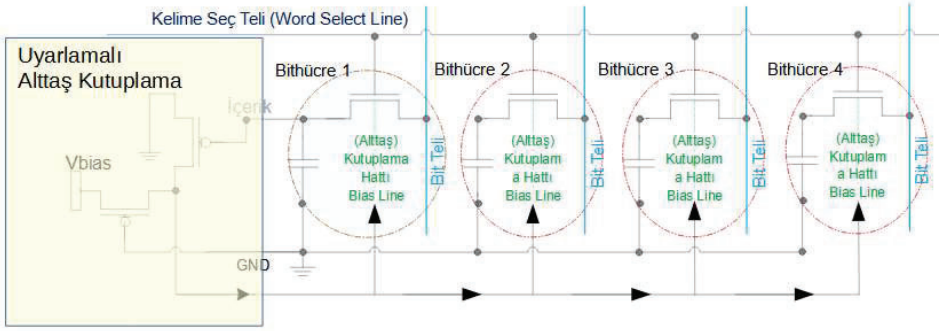


Şekil 3.9 : CADRAM için alternatif bit hücresi tasarımı (kavramsal).

Sunulan temsili gösterimler fikir vermek ve anlatım açısından daha çok yerleştirilmiştir, üretim açısından farklı tasarımlar gerekebilir, ancak bu tez kapsamında önerilen fikrin, örğenin içerik uyarlamalı DRAM hücresi tasarımını geçersiz kılmaz, tam tersi daha verimli kılar. Fikirler bu sayede tam ölçeklenebilirdir.

Önerilen CADRAM bit hücresi yapısı özellikle DRAM gibi alan maliyeti etkin yapılarda alan maliyeti açısından problem oluşturabilecektir. Çünkü bit hücresi başına en az iki transistör ve kutuplama hattı eklenmesi tolere edilebilir bir alan maliyeti olmayabilir. İşte bu sebeple bir hücre içeriğinin birden fazla hücreye uyarlanması çözümü geliştirilmiştir. Şekil 3.10 işte bu çözümü görselleştirerek anlatmak için konulmuştur. Buna göre 1. bit hücresinin içeriğine göre uyarlamalı alttaş kutuplama yapılır, ve buradaki parametre değişikliği ise 1. hücrenin komşuluğundaki 4 bit hücresine de uygulanır. Bu sayede birim hücre başına eklenmekte olan transistörler, 4 hücre için eklenir hale gelmektedir.

Bu yöntemle, alan maliyeti azaltılabilmektedir. Ancak, içeriği uyarlanan hücre, içeriğin uyarlandığı hücrelerin tuttuğu verileri düşük oranda temsil ettiği durumda, yenileme sayısı ve güç tüketiminde sağlanan başarımda yok olmaya başlar.



Şekil 3.10 : Tek içeriğin çok hücreye uyarlandığı CADRAM tasarımı.

3.4.2 CADRAM tasarımı benzetim sonuçları ve değerlendirme

Alan maliyeti en azından yakın komşuluktaki hücrelere içerik uyarlanmasıyla eğer tolere edilebiliyorsa, CADRAM yakın komşuluktaki hücrelerin aynı veriyi tuttukları varsayımda en az %65 oranda toplam yenileme sayısının düşmesini sağlamaktadır (mimari seviye benzetimler ve devre seviyesi saklama zamanı ile alttaş kutuplama gerilimi arasındaki sonuçlar kullanılarak hesaplanmıştır. ADRAM için gecikme, güç tüketimi ve saklama zamanı sonuçları erişim uyarlamalı DRAM ve üretim farklılığı uyarlamalı DRAM tasarımlarının birim hücre devre analiz sonuçları ve mimari benzetim sonuçlarıyla benzerdir ve o yüzden tekrar olmaması için burada verilmemiştir. Çünkü AADRAM ve PADRAM'de ve hatta TADRAM'de karar mekanizmaları farklıdır, ama sonuçta hücrelerin kutuplama/bias gerilimi hattı üzerinden uyarlamalı bias gerilimi yada kutuplamasız (no bias) uygulanır. Kutuplama/bias uygulandığındaki gecikmeler erişim transistörleri aynı olduğu için benzerdir, veya sızdırma azaltma oranları çok yakın kutuplama/bias aynı transistöre eklendiği için aynı kabul edilebilir. Mimari açıdan ise toplam yenileme sayısı ayrıca hesaplanmıştır, burada yenileme sayısında sağlanan düşüş belirtilmektedir, ayrıca ADRAM tasarımlarının tümüyle birlikte Ramulator üzerinden tüm denek taşları için alınan sonuçlarla birlikte görsel içinde paylaşılacaktır).

ADRAM'in bu önerdiğimiz tasarımı için alan maliyeti açısından çoklu hücreye uyarlama denenmesine rağmen problemleri devam etmektedir, içerik uyarlamayı uyguladığımız hücre arttıkça da içerik uyumu azalıp verim düşmektedir. Bu nedenle bu tasarım SRAM'deki gibi çoklu içeriğin çoklu hücreye uyarlandığı bir tasarıma ihtiyaç duymaktadır. Bu tasarım gelecekte çalışılması planlanan çalışmalardandır.

3.5 Üretim Süreci Farklılıkları Uyarlamalı DRAM, PADRAM

Bu bölümde üretim süreci farklılıkları uyarlamalı DRAM, PADRAM, tasarımına başlanmadan önce üretim farklılıkları nedir, neden DRAM için sorun olmaktadır, motivasyonumuz ne ve problemle ilgili mevcut çalışmalar nedir gibi sorulara cevap aranacaktır (bu nedenle ayrıca alt alt bölüm açılmıştır).

3.5.1 Motivasyon ve ilgili çalışmalar

"DRAM bit hücreleri özdeş değillerdir" ve "DRAM hücreleri de transistörlerden oluşan tümleşik devreler gibi sızdırırlar". Üretim süreci farklılıkları uyarlamalı DRAM (PADRAM) temelde bu iki prensibe dayanmaktadır.

1. Sızdırma: DRAM bit hücrelerinde, sızdırma akımları transistörlerin küçülmesiyle ve üretim teknolojileri geliştikçe doğrudan artar ve birim alana daha çok transistör gelmesiyle artan ısıl problemler de bu artışı hızlandırır [6].

2. Üretim Süreci Farklılıkları: Üretimden kaynaklı olarak DRAM bit hücrelerinin bazı temel parametrelerinde farklılık vardır ve bunların başında saklama zamanı farklılığı gelmektedir. Bir DRAM serimindeki DRAM bit hücrelerinden bazıları diğerlerine göre daha az sızdırır ve saklama zamanları daha uzun sürer. Bu hücrelere "güçlü" hücreler denilir. Diğer taraftan bazı hücreler de bu hücrelere kıyasla daha çok sızdırır ve saklama zamanları da bu nedenle daha kısadır. Bu hücrelere de "zayıf hücreler" denilir. Buradaki problem ise daha önce de anlatıldığı üzere tüm hücrelere DRAM üreticileri tarafından üretilen hazır ürünlerde [44] aynı saklama zamanı varmış gibi yenileme yapılmasıdır, ve bu yenileme sıklığı en zayıf hücreye ve en kötü duruma göre üzerine pay konularak belirlenir. Bu durumda da güçlü hücreler için anlamsız yenileme (yenilemenin maliyeti Bölüm 2.5'de anlatılmaktadır ama özetlenecek olursa; yenileme sırasında okuma ve yazma yapılamaması kaynaklı başarım kaybına neden olur [29, 45], ve her yenileme işlemi güç tüketiminin artması anlamına gelir.) gerçekleşmiş olmaktadır.

İlgili Çalışmalar: DRAM için yenileme problemdir, bu problemin kaynağı da sızdırma akımlarıdır. Sızdırma akımlarının azaltılmasına yönelik transistör seviye birçok çalışma önerilmiştir [5, 9, 10, 12, 61]. Ancak bu çalışmalar çoğunlukla SRAM ve transistör tabanlı çalışmalardır. DRAM üzerine de sızdırma akımlarını azaltmak üzerine çalışmalar yer almaktadır, ancak bu çalışmalar sadece transistör seviyede kalmaktadır ve saklama zamanının artırılması ve uyarlamalı olması gibi temel çözümler olmaksızın katkıları sınırlı olabilmektedir [62–64]. Bu tasarımlar daha çok doğrudan güç tüketimi düşürmek üzerine odaklanmışlardır.

Diğer taraftan, doğrudan yenileme işlemlerinin sayısını azaltmaya yönelik mimari seviye çalışmalar da yapılmaktadır [8, 46, 48, 65–67]. Bu çalışmaların bir kısmı zayıf ve güçlü hücrelerin satır veya DRAM birimleri açısından sınıflandırılmasına dayanmaktadır, bu sayede en azından tüm hücrelere aynı yenileme sıklığı uygulanmamış olmaktadır ve yenileme sayısı da düşmektedir. Bazı çalışmalar da yenileme sayısını en azından belirli bir amaç için kullanmayı önermişlerdir, örneğin kritik data olan yerlere sık erişim diğerlerine daha az sıklıkla erişim gibi. Ancak, mimari seviye bu çalışmalar da bu sefer sorunun asıl kaynağı olan saklama zamanını artırmaya ve sızdırmayı azaltmaya yönelik bir çözüm önermemekte ve donanım seviyesinde uyarlamalı bir çözüm sunmamaktadırlar.

İşte ADRAM tasarımları (TADRAM, PADRAM, CADRAM ve AADRAM) ve özellikle PADRAM'ın amacı farklı girdilere göre uyarlamalı olarak DRAM

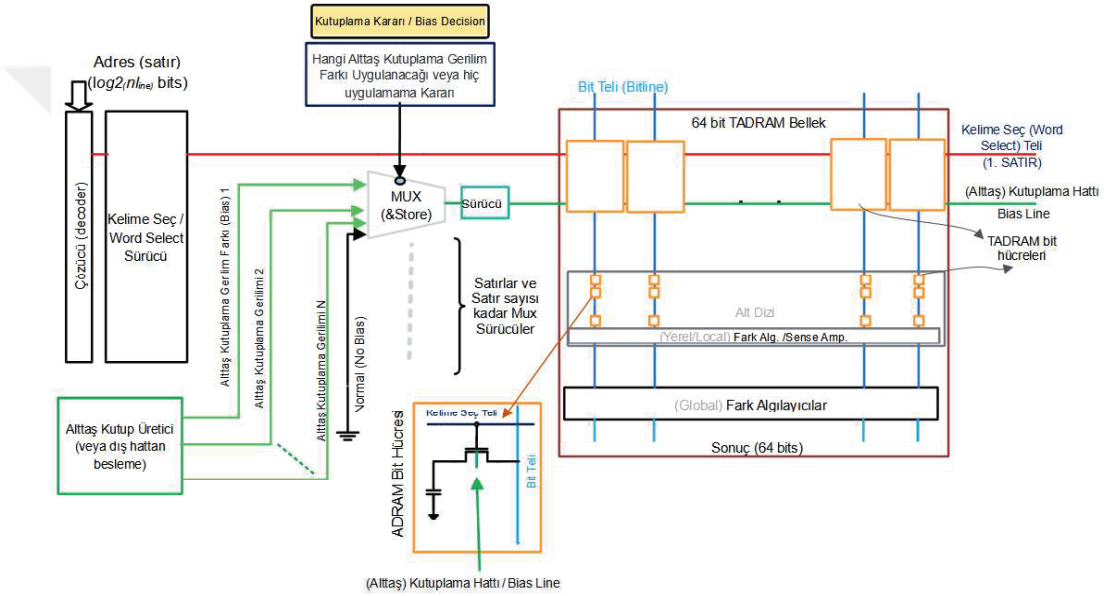
hücrelerinin donanım parametrelerinin değiştirilebildiği bir tasarımla hücrelerin saklama zamanını artırmak ve toplam yenileme sayısını düşürerek hem güç tüketiminde düşüş hem de başarımda artış sağlamaktır [68]. Not: Doktora çalışmalarım kapsamında DRAM üzerine çalışma ve tasarımlarımdan, PADRAM ile ilgili olanlarından bir kısmı yayın haline getirilmiş ve ITC'19'dan kabul alıp, sunulmuştur. Bu yayınımızdan sonra çalışmalar devam ettiği için tezde kullanılan sonuçlar daha detaylı ve kapsamlı halde sunulmaktadır, ama mimari tasarımda (örn, Şekil 3.11) değişiklik yoktur, ve sonuçların bir kısmı da ortaklanmıştır.

3.5.2 Önerilen PADRAM mimarisi ve devre tasarımı

PADRAM bu amaçla, mimari seviyedeki tekniklerden en bilineni seçilerek [48], satır bazında yenileme periyodu profilleme/gruplama tekniğini uyarlama mekanizmasına girdi olarak kullanır. Bu çalışmadakine benzer mantıkla satırlar için yenileme zamanı gruplarına göre etiketleri oluşturur. Hangi satır hangi gruptaysa bu kaydedilir. Örneğin; 2 gruplu bir profilleme yaptığımızı varsayalım. Varsayım şöyle devam etsin; n. satırda en düşük saklama zamanı 70 ms olsun, ve m. satırda ise en düşük saklama zamanı 144 ms olsun, ve 128 ms altındakiler bir grup (64 ms) üstündekiler de bir grup (128) şeklinde tanımlama olsun. Bu durumda n. satır 64 ms grubunda, m. satır 128 ms grubunda olur. İşte bu bilgiyi tüm satırlar için kaydettikten sonra, PADRAM tasarımı bu bilgiyi devre parametrelerini o satıra uygun şekilde uyarlamak için kullanır. Temel amaç, zayıf hücrelerin olduğu satırlardaki hücrelere uyarlama yaparak devre parametrelerini değiştirerek saklama zamanını artırmaktır. Böylece bir satırın bir üst gruba geçmesi mümkündür, böylece daha az yenileme sayısı sağlanmış olur o satır için ve bu yöntemle bir üst gruba geçebilen tüm satırlar için. Sonuçlar bir sonraki alt bölümde sunulmaktadır.

PADRAM 2 şekilde gerçeklenebilir: Sızdırma akımlarını azaltarak saklama zamanlarını artırmaya yönelik uyarlamalı alttaş kutuplama yapan PADRAM, ve doğrudan saklama zamanlarını artırmaya yönelik uyarlamalı gerilim ölçekleme yapan PADRAM. Uyarlamalı alttaş kutuplama yapan PADRAM'ın bit hücresi tasarımı uyarlamalı alttaş kutuplama tabanlı TADRAM bit hücresi tasarımıyla aynıdır. Çünkü TADRAM için sıcaklık, PADRAM için ise satırların saklama zamanı uyarlama için kullanılır, ve bu farklılık haricinde devre parametrelerini uyarlayacak mekanizma bit hücrelerinde aynıdır (Şekil 3.2). Uyarlamalı gerilim ölçekleme yapan PADRAM'ın bit hücresi tasarımı da uyarlamalı gerilim ölçeklemesi tabanlı TADRAM ile aynıdır. Alttaş kutuplama yapılmayacağı ve kutuplama/bias hattı artık normal/default olacağı için ayrıca hat kullanılmaz.

PADRAM için küme (bank) yapısını anlatan temsili kavramsal tasarım gösterimi Şekil 3.11 ile verilmektedir. Bu şekil, TADRAM ve AADRAM ile benzerdir, sadece karar destek mekanizmasına girdi değişmektedir (sıcaklık, erişim örüntüsü, saklama zamanı bilgisi). Kutuplama hattı için sürücü, çoklayıcı her satır için ayrı olabileceği gibi çok satıra bir tane de olabilir. Onun haricinde DRAM dizi yapısı geçerlidir, sadece DRAM bit hücrelerinde bahsedildiği üzere kutuplama hattı eklemesi yapılmıştır, satır boyunca bu hat devam eder. Gerilim ölçeklemeli PADRAM'da ise kutup üretici, çoklayıcı, kutuplama gerilimi sürücü ve kutuplama hattı yoktur. DRAM dizisi ve DRAM hücresiyle özdeştir, sadece bit teli üzerinden kelime seç teliyle aktif edilen gereken hücrelere daha yüksek vdd veya default vdd basılır. Zaten var olan Vdd hattı, daha yüksek gerilim ölçeklemesi için kullanılmış olur.



Şekil 3.11 : PADRAM küme/bank yapısı temsili gösterimi.

PADRAM işlevsel akışı şu şekilde gerçekleşir: Bit hücresinde sızdırma akımlarını azaltmak üzere satırlar için saklama zamanına bağlı olarak uyarlama yapılır. Bunun için DRAM ilk çalıştırılma anında bir kez tüm satırlar için saklama zamanı grupları belirlenir. En düşük saklama zamanına sahip hücreye göre satırlar o gruptaki alt limitin saklama zamanına sahip kabul edilir. PADRAM bu noktada düşük saklama zamanı grubundaki satırlara alttaş kutuplama veya gerilim ölçekleme uygular. Yüksek saklama zamanı grubundaki hücreleri ise default durumda bırakır. Uygulanan alttaş kutuplama gerilimi veya diğer yöntemde uygulanan yüksek voltaj sayesinde zayıf gruptaki satırların saklama zamanları artar. Bu o gruplar için aslında uygulanan gerilime bağlı olarak katsayı ile tüm hücrelerin saklama zamanlarının çarpılması ve tekrar hesaplanması anlamına gelir, katsayı da uygulanan gerilimlerle saklama zamanlarındaki artış demektedir. Daha sonra tekrardan tüm satırlar için profilleme işlemi tekrarlanır, ve hangi satır hangi saklama zamanı grubunda belirlenir. Buna

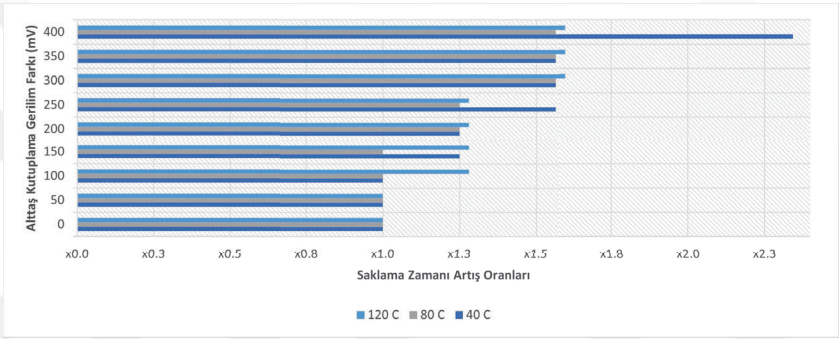
göre de DRAM çalışmasına devam edilir, ve o sırada da belirlenen saklama zamanı atamalarına göre her satır için yenileme gerçekleştirilir. Bir satır eğer PADRAM sayesinde bir üst saklama zamanına geçerse bu sayede o satırdaki tüm hücreler için daha az yenilenme sağlanmış olur.

Motivasyon kısmında belirtilen örnekle devam edecek olursak, n. satırda en düşük saklama zamanlı hücre 81 ms olduğu için satır 64 ms grubunda ve m. satırda en düşük saklama zamanlı hücre 144 ms olduğu için satır 128 ms grubunda idi. PADRAM'ın alttaş kutuplama yöntemi kullandığımızı ve uygulanan gerilimle saklama zamanının %60 artırılabildiğini varsayalım, bu durumda katsayı 1.6 olur tüm satırlar için. 2 gruplu olduğunu düşündüğümüzde yüksek gruba zaten kutuplama/bias gerilimi uygulanmaz ve katsayı ile çarpılmaz. Düşük saklama zamanı grubuna ise 1.6 ile çarpımı uygulandığında, 81 ms olan en düşük hücrenin saklama zamanı artık 129 ms olmaktadır. Böylece artık n. satır bir üst gruba yani 128 ms grubuna geçer. Sonuç olarak da, bu şekildeki tüm satırlar ve hücreler için yüzde yüz yenileme sayısında azalma sağlanır. Daha sonra grup değişimi olan satırlar kaydedilir ve sadece bunlara uyarlama yapılır. Bu bilgi, Şekil 3.11'de gösterilen kutuplama kararı/bias decision olarak DRAM dizisine ulaştığında n. satır için alttaş kutuplama gerilimi uygulanmaya başlar, ve diğer bu şekilde grup değiştiren satırlara uygulanır. Böylece toplam yenileme sayısı önemli ölçüde azaltılmış olur. Bir PADRAM mimarisinde uygulanacak gerilimler belirlenecek grup sayısına göre değişir. Profilleme sırasında bu örnekte 2 grup kullandık, bu 4 grup olsaydı 3 farklı kutuplama/bias gerilimi ve 1 tane de default gerilim uygulayacaktır. Bu şekilde, grup sayısına bağlı gerilim çeşitliliği de artacaktır. Şekil 3.11'de tek bir hat özellikle bu örneğe hitaben konulmuştur, grup sayısı arttıkça TADRAM dizi görselindeki gibi farklı kutuplar üretilecek ve kutuplama hatları üzerinden çoklayıcıya gönderilecektir, çoklayıcı da ilgili satır için uygun gerilimi uyarlayıp sürücü tarafından sürülecektir. Benzer mantık, gerilim ölçekleme için de geçerlidir.

3.5.3 PADRAM tasarımı benzetim ve analiz Sonuçları

PADRAM'ın iki farklı gerçekleme yöntemi için devre seviyesi ve mimari seviye benzetim sonuçları bu kısımda anlatılmaktadır. Burada gecikme, alan maliyeti de tartışılmaktadır, ancak asıl olarak; uygulanan besleme gerilimiyle veya uygulanan alttaş kutuplama gerilimiyle sağlanabilecek saklama zamanı artışları katsayı olarak belirlemek için elde edilmektedir/sunulmaktadır, sonrasında farklı grup sayılarına ve farklı gerilimlere göre bu katsayılar kullanılarak tüm denek taşları üzerinden mimari benzetimlerin sonuçları; toplam yenileme sayılarının değişimi elde edilmektedir/sunulmaktadır.

PADRAM bit hücresinde sızdırma akımlarının azaltılması için alttaş kutuplama uygulanır. Uygulanan gerilimle, sızdırma akımlarının azalması, durağan enerji kayıplarının azalması, ve saklama zamanının artması hedeflenmektedir. Şekil 3.12'de alttaş kutuplama gerilim farkı ile (body bias voltage change) saklama zamanı (retention time) artış oranlarının değişimi gösterilmektedir. Burada yüzde yerine direkt artış oranı verilir ki, katsayılar olarak toplam yenileme sayısını hesaplamak için mimari benzetimlerde kullanılabilsin. Bu şekilde, farklı sıcaklıklarda kutuplama/bias gerilimi saklama zamanı ilişkisi sunulmaktadır. Yaklaşık olarak, 200 mV kutuplama gerilimi 1.3 kat, 300 mV kutuplama gerilimi 1.6 kat, ve 400 mV kutuplama gerilimi 2 kat kadar saklama zamanını artırmaktadır. Kutuplama uygulanmayan temel DRAM tasarımı, default, ise 1 kat kabul edilir. Artan alttaş kutuplama gerilimleriyle güç tüketimi de düşmektedir, ve bu tüm sıcaklıklar için de geçerlidir.



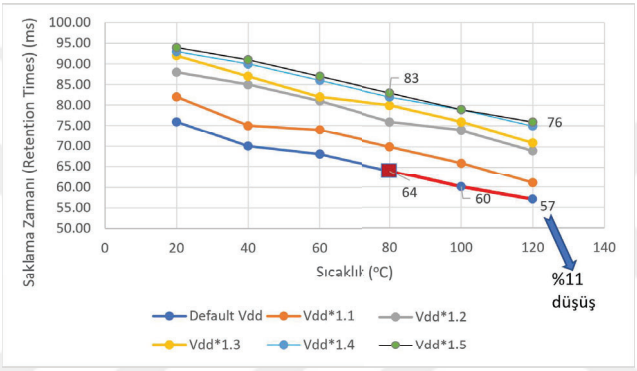
Şekil 3.12 : Farklı sıcaklıklarda, alttaş kutuplama ve saklama zamanı.

Farklı sıcaklıklarda artan kutuplama gerilimleriyle güç tüketiminde sağlanan azalma oranlarıyla ilgili sonuçlar Şekil 3.5'de yer almaktadır. Alan maliyeti ise, TADRAM ile benzer şekilde, her satır için bir sürücü, çoklayıcı ve bir kutuplama hattı kaynaklı oluşur. Kabaca PADRAM, her satır için; kelime seç sürücü ve hattı ile, birkaç satırlık bir kod çözücü kadar alan maliyetine neden olmaktadır. Temel tasarımdan farklı olarak ayrıca dışardan bir kutup üretici, veya tüm satırlar için ortak kutuplama üretici de vardır.

Eğer güç tüketimi kritikse bu uyarlamalı alttaş kutuplama yapan PADRAM, alan maliyeti kritikse diğer uyarlamalı gerilim ölçekleme yapan PADRAM tasarımının kullanılması tavsiye edilmektedir.

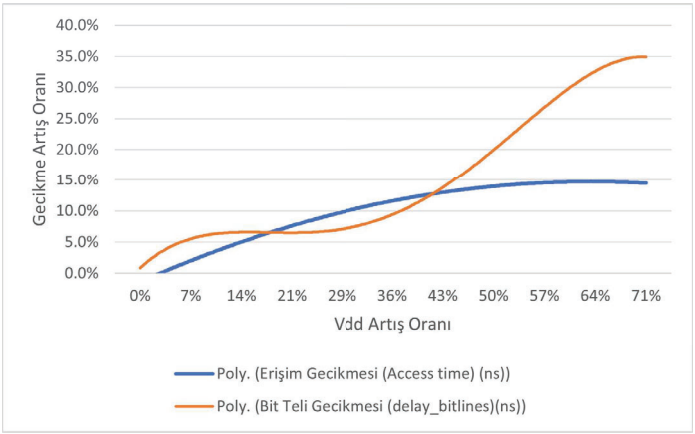
Gerilim ölçekleme yapan PADRAM tasarımında ayrıca kutuplama hattı yoktur, ve zaten var olan Vdd besleme yapısı aynen kullanılabilir. Bit telleri üzerinden beslenen Vdd yerine aynı hattan daha yüksek gerilim uygulanır. Bu nedenle, satır başına sürücü ve seçici ihtiyacı da yoktur, ancak uyarlama yapılan girdi grup sayısına bağlı olarak hangi hücrelere hangi besleme gerilimlerinin seçileceğine karar veren bir yapı gerekmektedir, var olan sürücü aynen kullanılabilecektir.

Besleme gerilimleri arttıkça saklama zamanı artmaktadır. Bunu sıcaklık eksenli hale çevirirsek, artan sıcaklıkla azalan yenileme zamanları artan Vdd artış oranlarıyla artmaktadır. Gerilim yükseldikçe, sıcaklıkla değişen saklama zamanı eğrisi bir üst zaman noktasından başlamaktadır. İlgili sonuçlar Şekil 3.13'de verilmektedir. Farklı sıcaklıklar için gerilimlerin saklama zamanlarındaki artış oranı az da olsa değişmektedir, ancak 1.3 kat Vdd ile; 1.4 V (default) yerine 1.8 V (yüksek) Vdd uygulamak, saklama zamanını yaklaşık %30'a kadar artırmaktadır (kutuplama gerilimindeki 200 mV kutuplama gerilimine karşılık gelmektedir.). Dolayısıyla katsayı olarak düşünürsek, 1.3 kat Vdd, kabaca 1.3 kat saklama zamanı artışı demektir. Ancak bu artış oranı gerilim arttıkça birebir oranda artmamaktadır, artan gerilimden kaynaklı yenileme zamanı artış oranı gerilim arttıkça azalmaktadır.



Şekil 3.13 : Gerilim ölçeklemeyle saklama zamanı değişimi.

Gerilim artışının diğer bir etkisi de, erişim zamanlarında ve bit tellerinin artan gerilim nedeniyle daha yüksek gerilime daha uzun sürede sürülebilmesine neden olmasıdır. 2 gecikme tipi için de Vdd artış oranıyla artan gecikme artış oranları Şekil 3.14'de sunulmaktadır. Örneğin; 1.3 kat bir Vdd ise normal Vdd'ye göre yüzde onun altında bir ilave gecikmeye neden olmaktadır. Erişim gecikmesi azalma oranı Vdd arttıkça azalmaktadır.



Şekil 3.14 : Gerilim ölçeklemenin gecikmelere etkisi.

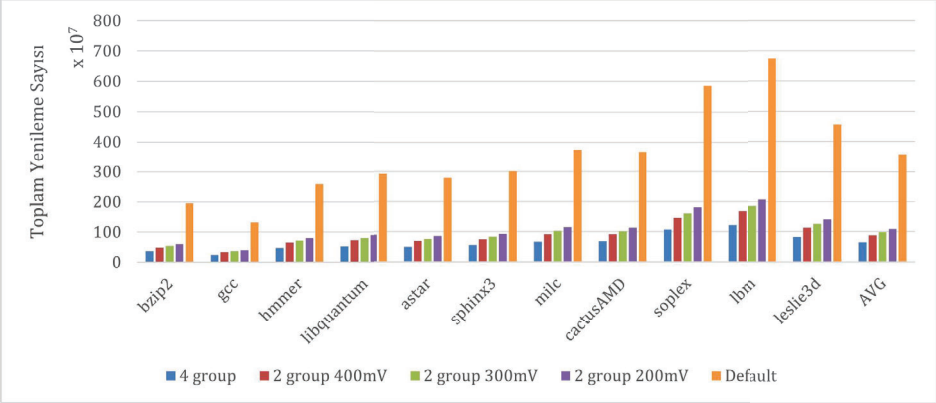
Toplam yenileme sayılarında sağlanan kazançla ilgili, kutuplama gerilimi (veya gerilim ölçekleme) ile elde edilen saklama zamanı artış katsayıları kullanılarak, mimari benzetimlerde 5 farklı tasarım için tüm denek taşlarıyla (benchmarks) koşum gerçekleştirilmiştir. Bu tasarımlar:

- Temel/Default DRAM: Tüm hücreler için 64 ms yenileme periyodu
- 2 grup 200 mV: Saklama zamanı profillemesi yapılarak satırlar 2 gruba adreslenir. Satırlar gruplara 127 ms ve altında, 128 ms ve üstünde olacak şekilde dağıtılır. 127 ms ve altında olan gruptaki hücrelerin saklama zamanları, 200 mV alttaş kutuplama gerilimi uygulandığındaki elde edilen saklama zamanı artış katsayısı ile çarpılır. Diğer grup ise temel konfigurasyonda bırakılır, saklama zamanları aynı kalır.
- 2 grup 300 mV: 127 ms ve altında olan gruptaki hücrelerin saklama zamanları, 300 mV alttaş kutuplama gerilimi uygulandığındaki elde edilen saklama zamanı artış katsayısı ile çarpılır. Diğer grup ise temel konfigurasyonda bırakılır, saklama zamanları aynı kalır.
- 2 grup 400 mV: Satırlar 2 gruba 127 ms ve altında, 128 ms ve üstünde olacak şekilde dağıtılır. 127 ms ve altında olan gruptaki hücrelerin saklama zamanları, 400 mV alttaş kutuplama gerilimi uygulandığındaki elde edilen saklama zamanı artış katsayısı ile çarpılır. Diğer grup ise temel konfigurasyonda bırakılır, saklama zamanları aynı kalır.
- 4 grup: Satırlar 4 gruba dağıtılırlar. İlk grup 128 ms altındaki satırlardır, ikinci grup 128 ms ile 192 ms arasındaki gruptur, üçüncü gruba 192 ms ile 256 ms arasındaki gruptur ve son grup da 256 ms üzerindeki gruptur. 1. gruba 400 mV, 2. gruba 300 mV, 3. gruba 200 mV alttaş kutuplama gerilimi ile elde edilen saklama zamanı artış katsayısı çarpımı uygulanır ve son grup ise temel konfigurasyonda bırakılır, saklama zamanları aynı kalır.

Önce saklama zamanı profillemesi yapılarak gruplara dağıtılan satırlardaki hücrelerin saklama zamanları, daha sonra ilgili grup için belirlenen saklama zamanı artış katsayısı ile çarpılır. Daha sonra ise tekrardan tüm satırlar için saklama zamanı profillemesi işlemi gerçekleştirilir. Saklama zamanına göre belirlenen grubu bir üst gruba geçen satırlar için o üst gruptaki saklama zamanına göre yani daha az yenileme yapılması yeterli olacaktır artık, bunu sağlamak için de o satırlara da katsayısı uygulanan alttaş kutuplama gerilimi (veya besleme gerilimi) uygulanır ve devam eder.

Her bir denek taşı, Ramulator kullanılarak bu yöntemle boot sırasında profilleme, katsayı uygulama, profilleme tekrarı işlemlerinin ardından bir üst saklama zamanı

grubuna çıkan, aynı kalan veya çarpım uygulanmayan tüm satırlar için ilgili grubundaki saklama zamanlarına göre yenileme gerçekleştirilir, okuma ve yazma yapılır, gerçek bir DRAM donanımında bu uygulamalar çalıştırılıyor gibi sayılar elde edilir. Farklı grup sayılarına, bu gruplar için farklı kutuplama gerilimlerine göre toplam yenileme sayıları default temel DRAM tasarımına kıyaslanmıştır ve elde edilen sonuçlar her bir denek taşı için Şekil 3.15'de sunulmaktadır. Buna göre grup sayısı arttıkça toplam yenileme sayısındaki sağlanan azalma oranı artmaktadır. Aynı grupta besleme gerilimi arttıkça da toplam yenileme sayısı tüm denek taşları için azalmaktadır. Asıl temel sonuç ise baseline DRAM ile kıyaslandığında ortaya çıkmaktadır, en yalın 2 gruplu ve en düşük uygulanan besleme gerilimli PADRAM tasarımı sayesinde DRAM baseline tasarımına göre en az %65 toplam yenileme sayısında düşüş sağlanmış olmaktadır.



Şekil 3.15 : Farklı denek taşlarına göre toplam yenileme sayıları.

3.6 Erişim Uyarlamalı DRAM, AADRAM

3.6.1 AADRAM mimarisi ve devre tasarımı

Uyarlamalı DRAM tasarımlarının sonuncusu Erişim Uyarlamalı DRAM (AADRAM) tasarımıdır. Bu tasarımın motivasyonu şu şekilde özetlenebilir: Tüm devreler sızdırır [2, 6]. Bir DRAM satırına yakın zamanda erişim olmuşsa tekrar o satıra erişim olma ihtimali yüksektir [69]. Yenileme maliyetlidir. Yenilemeye yönelik sadece mimari seviye çözümler, yada sızdırmaya yönelik sadece devre seviyesi çözümler eksik kalmaktadır; iki çözümü de sunan uyarlamalı yapılara ihtiyaç duyulur. Bu motivasyonla geliştirilen AADRAM tasarımları, satırlar için erişim örüntüsünü devre parametrelerini kendi kendine değiştirmek için kullanır.

Bir AADRAM bit hücresi tasarımı ile TADRAM ve PADRAM bit hücresi tasarımları aynıdır (bit hücresi görseli Şekil 3.2, DRAM hücresinden tek farklı yanı kutuplama

yapılarak gerçeklenen tasarım için "bias line" yani kutuplama hattı eklentisidir. NMOS transistörler için toprağa bağlı olduğu default tasarım için varsayılan konfigurasyon bu tasarımlar için uyarlamalı olarak default değerine yani toprağa yada bias gerilimiyle sürülür. Serim açısında bakıldığında ise bu 3 farklı ADRAM tasarımının bit hücresi serim görüntüsü ile DRAM bit hücresi serim görüntüsü aynı gözükmektedir (Serim görüntüsü Şekil 2.3'de yer almaktadır).

PADRAM ve TADRAM ve AADRAM birbirlerine oldukça benzer tasarımlardır, PADRAM üretim süreci farklılığından doğan saklama zamanı uyarlamalı olarak, TADRAM ortam veya DRAM çalışma sıcaklığına uyarlamalı olarak ve AADRAM ise satırların erişim zamanları uyarlamalı olarak DRAM hücrelerinin devre parametrelerini, alttaş kutuplama gerilimleri veya besleme gerilimleri, kendi kendine değiştiren DRAM tasarımlarıdır (ancak bu 3 tasarımda uyarlanan girdiği üretirken oldukça farklı yöntemler izler, dolayısıyla uyarlanan girdi farklı şekilde üretilir ve mekanizmalar farklıdır, ancak uyarlama mekanizmaları aynıdır.). Aynı şekilde, yine bu 3 tasarımın bank tasarımları (AADRAM bank kavramsal gösterimi Şekil 3.11'de sunulmaktadır.) da benzerdir, tek farklılık her bir tasarıma uyarlama için aktarılan girdidir (bias decision: Sıcaklık veya Erişim veya Saklama zamanı olmaktadır, farklı 3 ADRAM tasarımına göre). DRAM'den farkı ise daha önce de belirtildiği üzere uyarlamalı kutuplama tabanlı tasarım için kutuplama üretici, çoklayıcı ve sürücüdür.

AADRAM'de diğer tasarımlarda olduğu gibi hem uyarlamalı alttaş kutuplama yöntemiyle hem de uyarlamalı gerilim ölçekleme yöntemiyle gerçeklenebilmektedir. Bahsedilen farklılıklar gerilim ölçekleme yöntemine dayalı AADRAM tasarımı için geçerli değildir, ne bank tasarımında ne de bit hücresi tasarımında DRAM temel tasarımıyla aynı kabul edilebilir. Bu yönteme dayalı AADRAM'in farkı ise, bit telleri üzerinden sürülen yüksek besleme gerilimini uyarlayacak ve hangi besleme gerilimini seçeceğine karar veren mekanizmadır. Zaten sürücü ve bit hatları DRAMde mevcut halleriyle kullanılırlar, DRAM bit hücresi ise bu yüzden birebir DRAM bit hücresi (Şekil 2.2'de yer almaktadır) ile birebir aynıdır denilebilir (Vdd artış oranı limitlendiği için aynı tasarımın geçerli olacağı varsayılmıştır, ancak üretim aşamasında bunun geçerli olup olmayacağına bakılarak uygulanacak en yüksek Vdd'ye göre kapasitör ve sürücünün tasarımının veya parametrelerinin kontrol edilmesi beklenir.).

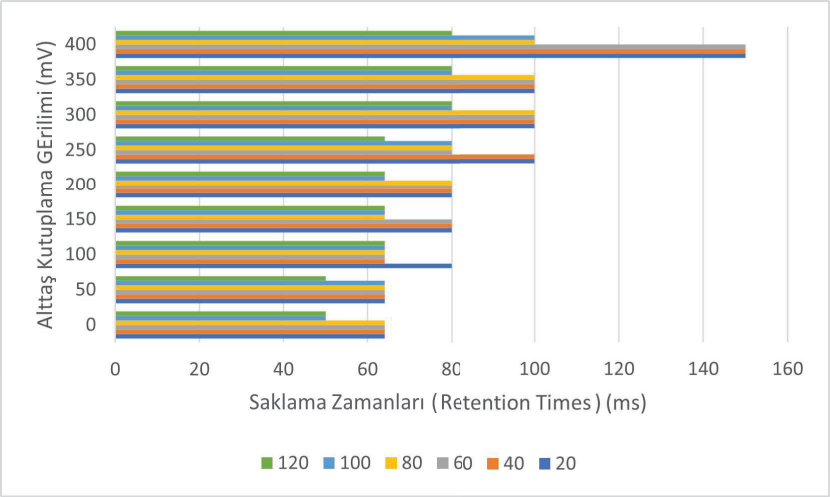
AADRAM yakın zamanda erişilen satırların tekrar erişilme ihtimalinin yakın zamanda erişilmemiş olan satırlardan fazla olması prensibini kullanır, ve bu prensibe göre satırların erişim örüntülerinin tutulması gerekir. Bunun için profilleme mantığı verimli değildir, bunun yerine AADRAM için erişim için tablo çözümünü geliştirmiştir. Yeni erişilen satırlar, belirli tekrarlı erişimden sonra bu tabloya yazılırlar. O yüzden tablo tabanlı (table based) tasarım olarak da isimlendiriyoruz bu tasarımı. Örnek bir durum

için, 256 girişli bir tablo düşünelim ve tabloya da en az 10 erişimden sonra bir satırı yazdığımızı düşünelim. Bu şekilde bir satıra 11. kez erişiliyorsa artık bu tabloya kaydedilir, ve bu tablodaki satırlara ve hücrelere herhangi bir alttaş kutuplama (veya gerilim ölçekleme) uygulanmaz. Bu satır dışındaki satırlara ise uyarlamalı olarak alttaş kutuplama (veya gerçekleme yöntemine göre gerilim ölçekleme) yapılır.

AADRAM tasarımındaki amaç, tablo dışındaki satırların ve bu satırlardaki hücrelerin yakın zamanda erişilmeme ihtimalleri yüksek olduğu için bu satırların devre parametrelerini değiştirip saklama zamanlarını artırmak ve güncel satır saklama zamanlarına göre yenileme yaparak toplam yenileme sayısını düşürmektedir. Unutulmamalıdır ki, AADRAM dahil hiçbir ADRAM tasarımı birebir gerçek hayata uyarlanacak tasarımda olmayabilir, eksik, hata veya güncelleme ihtiyacı olabilir. Ancak bu tez kapsamında kavramların ispatları yapılmaya çalışılmıştır. Ve temel amaçlar ve fikirler ise her zaman geçerlidir. Örnek verecek olursak, bu tip bir tablonun boyutu üretilecek bir DRAM için farklı olabilir, veya tablo yerine başka bir yöntem daha gerçeklenirdir, veya tablo dışındaki tüm satırlara uyarlama yapmanın üretim açısından farklı bir yöntemi veya limiti olabilir, ... Ancak buradaki fikir tasarımda veya sonuçlarında eksik/hata olsa bile geçerlidir, DRAM'de satırların erişim örüntüsü vardır, buna göre sık erişilenler ve erişimi görece daha seyrek olan satırlar vardır. Buna göre de erişimi seyrek olan satırların bir kısmının bir grubunun veya tümünün ve bu satırlardaki hücrelerin devre parametreleri uyarlamalı olarak değiştirilerek saklama zamanları artırılır. Böylece daha uzun sürede bir erişilecekleri için, saklama zamanlarının artması veri kaybını önleyecek veya yenileme sıklığının düşürülebilmesini sağlayacaktır.

3.6.2 AADRAM tasarımı benzetim ve analiz sonuçları

AADRAM tasarımının her iki gerçekleme yöntemi için sonuçlar bu kısımda anlatılmaktadır. Alttaş kutuplama yöntemiyle gerçeklenen AADRAM tasarımında, bit hücresi için farklı sıcaklıklarda, artan kutuplama gerilimi ile artan saklama zamanlarına dair sonuçlar Şekil 3.16 ile gösterilmektedir. AADRAM tasarımı için bu saklama zamanlarındaki artış katsayı olarak kullanılmaktadır. (Bu alt bölümde, karşılaştırma amaçlı, hem alttaş kutuplama hem de gerilim ölçekleme yöntemleri için bir örnek seçilip toplam yenileme sayılarının olduğu sonuçlarla birlikte etkileri değerlendirilecektir. Bu örnek %25 saklama zamanı artışı görülen, 80 $\pm$ 5 °C sıcaklıkta, 200 mV alttaş kutuplama gerilimidir.). Farklı sıcaklıklarda artan kutuplama gerilimleriyle güç tüketiminde sağlanan azalma oranlarıyla ilgili sonuçlar Şekil 3.5'de yer almaktadır.

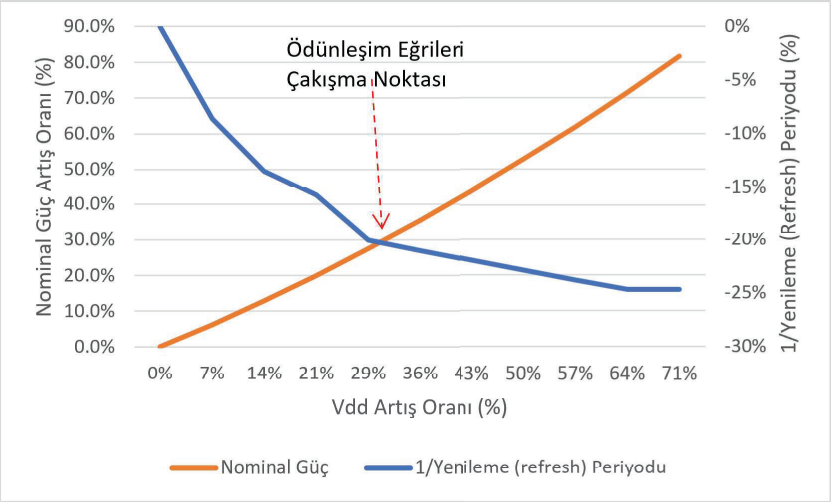


Şekil 3.16 : Farklı alttaş kutuplama gerilimleri ve saklama zamanları.

Uyarlamalı gerilim ölçekleme tabanlı AADRAM tasarımı, erişim örüntüsüne göre erişim tablosu dışındaki hücreler için yüksek besleme gerilimi uygulanarak saklama zamanının artmasını amaçlar. Besleme gerilimi artmasıyla saklama zamanı artar, ve bundan kaynaklı her yenileme işlemi için harcanan güçten tasarruf edilip hem de okuma ve yazmayla çakışma engellenerek başarım artışı sağlanmış olur. Ancak diğer taraftan da besleme gerilimi artışının doğrudan güç tüketimine de etkisi bulunmaktadır. Bu iki amaç arasında ödünleşim bulunmaktadır.

80 $\pm$ 5 °C sıcaklık koşulu altında, Vdd artış oranıyla değişen güç tüketimi oranı ve yenileme sıklığı (1/yenileme periyodu) sonuçları Şekil 3.17 ile sunulmaktadır. Sonuçlara göre, Vdd arttıkça 1/yenileme periyodu oranı 0 dan eksi değerlere düşmekte yani azalmaktadır. Diğer ifadeyle saklama zamanı artmaktadır (1/yenileme periyodu şeklinde yazılmasının sebebi, ödünleşim sırasında eğrilerin çakışan noktasını ödünleşim noktası varsaymak içindir). Diğer taraftan da güç tüketimi artışı Vdd artışıyla artmaktadır (ancak burada toplam yenileme sayısının düşmesi kaynaklı güç tüketimi azalması daha baskın olmaktadır.). Bu eğrilerin çakıştığı nokta ise yaklaşık %29-%30 Vdd artışına karşılık gelir. Bu nokta için saklama zamanı %25-%30 artmaktadır (diğer AADRAM gerçekleme yönteminde de saklama zamanını aynı oranda artıran 200 mV değeri seçilmiştir. Bu gerçekleme yöntemindeki karşılığı ile uyumlu ve karşılaştırılabilir olması amaçlanmaktadır.).

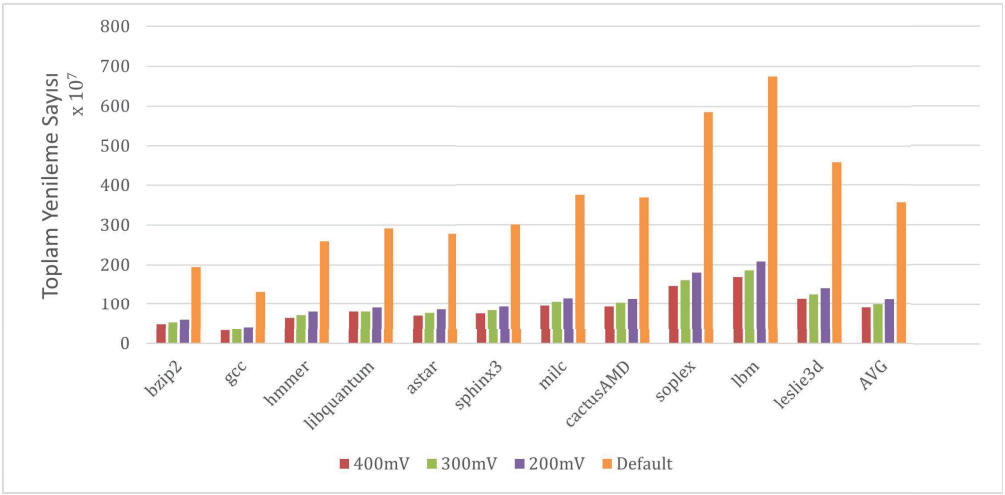
AADRAM tasarımında tablo dışında kalan satırlar ve bu satırlardaki hücrelerle, tablo içinde kalan satır ve hücreler iki gruba ayrılmaktadır. Aslında birden fazla tablo yapmak veya tabloyu büyütmek mümkündür ancak tablonun boyutunun artmasının erişim gecikmelerine ve alan maliyetine etkisi olacaktır. Örnek olarak bu tez kapsamında 256 girdilik tablo seçilmiştir. 10 erişim limiti konulmuştur. Bu şartlar altında Ramulator üzerinden tüm denek taşları üzerinden şu tasarımlar için ihtiyaç



Şekil 3.17 : Vdd artışıyla değişen güç tüketimi ve yenileme sıklığı.

duyulan toplam yenileme sayılarına bakılmıştır: 1. Default: Herhangi bir uygulama yapılmayan, tüm hücreler için 64 ms default yenileme periyodu uygulanan DRAM tasarımıdır. 2. Tablo dışındaki satırlara 200 mV alttaş kutuplama gerilim farkı uygulanan tasarımdır. 3. Tablo dışındaki satırlara 300 mV alttaş kutuplama gerilim farkı uygulanan tasarımdır. 4. Tablo dışındaki satırlara 400 mV alttaş kutuplama gerilim farkı uygulanan tasarımdır. Elde edilen sonuçlar Şekil 3.18'de sunulmaktadır. Bu sonuçlar tasarımın en düşük gerilimde bile temel tasarım DRAM'e göre ne kadar toplam yenileme sayısında düşüş sağladığını göstermektedir. Üstelik bu kazanım farklı gerçek uygulamaları örnekleyen yaygın kullanılan farklı denek taşları (Şekil 3.18'de sonuçları sunulan) için de geçerliliğini sürdürmektedir.

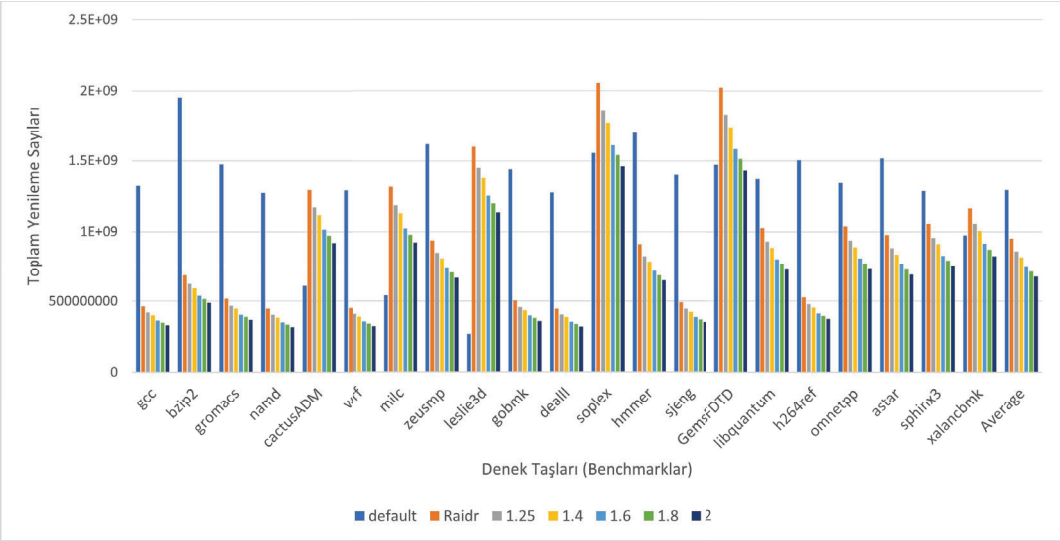
Aynı varsayımlarla Ramulator üzerinden tüm denek taşları için ayrıca şu tasarımlar için ihtiyaç duyulan yenileme sayıları için mimari seviye benzetimler koşturulmuştur:



Şekil 3.18 : Farklı AADRAM ve DRAM tasarımları yenileme sayıları.

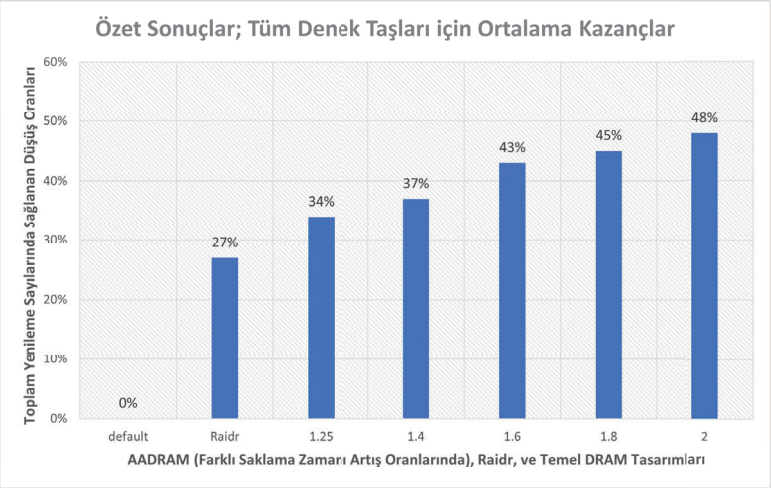
- DRAM (default): Tüm hücreler için 64 ms yenileme periyodu uygulayan temel DRAM tasarımıdır.
- RAIDR [48]: Tekniğin bilinen iyi durumundaki ilgili bir çalışma ile karşılaştırmak için seçilen, mimari seviye saklama zamanı profilleme mekanizması tabanlı çalışmadır.
- (1.25): Tablo dışındaki hücrelere Vdd'nin 1.25 katı uygulandığı AADRAM tasarımıdır. Bu tasarım, Vdd artışıyla saklama zamanı artışı ve güç tüketimi ödünleşiminin sonucunda tespit edilen noktayı temsil eder, bu noktanın diğer tasarımdaki karşılığı 200 mV olmaktadır.
- (1.4), (1.6), (1.8), (2) tasarımları: 4 farklı Vdd artış oranını temsil eden, tablo dışındaki hücrelere bu artış oranlarında besleme yapılan tasarımlardır.

Elde edilen sonuçlar Şekil 3.19'da sunulmaktadır. Denek taşından denek taşına, farklı tasarımların toplam yenileme sayısı ihtiyacı açısından sonuçları değişiyor olsa da, sonuçlara bakıldığında gerilim ölçeklemeli AADRAM tasarımı, temel DRAM tasarımına göre yenileme sayısında etkin düşüş sağlamaktadır. Ayrıca, karşılaştırılan literatürdeki bilinen ilgili çalışmalardan raidr tasarımına göre de ([48]), önerilen AADRAM tasarımlarının toplam yenileme sayısında sağladığı düşüş daha fazladır.



Şekil 3.19 : Gerilim ölçeklemeli AADRAM ve yenileme sayıları.

Her bir tasarım alternatifi için tüm denek taşları üzerinden elde edilen toplam yenileme sayısının ortalaması ve bu ortalamaların DRAM temel tasarımına göre oranlamasına ait özet sonuçlar ise Şekil 3.20'de verilmektedir. Buna göre, Vdd oranında %25 artış sağlayan, ödünleşim eğrilerinin çakıştığı noktadaki, AADRAM tasarımı temel DRAM tasarımına göre %34 toplam yenileme sayısında düşüş sağlamaktadır. Ayrıca,

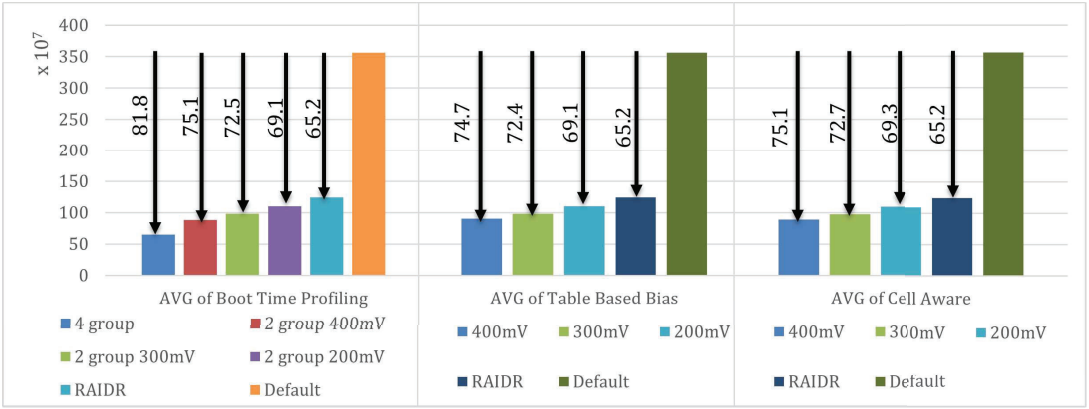


Şekil 3.20 : Farklı AADRAM tasarımları için karşılaştırma sonuçları.

bu oran Vdd artış oranı arttıkça daha da yükselmektedir. Diğer taraftan, tablo dışındaki hücrelere uyarlamalı olarak %25 fazla Vdd besleyen bu AADRAM tasarımı, tekniğin bilinen örneklerinden Raidr tasarımına kıyasla [48], yaklaşık %25 daha fazla toplam yenileme sayısında kazanç sağlamaktadır.

3.7 Sonuç ve Değerlendirme: Uyarlamalı DRAM (Adaptive DRAM)

Uyarlamalı DRAM (Adaptive DRAM) için tez kapsamında alttaş kutuplama ve gerilim ölçekleme ile gerçeklenebilen 3 farklı tasarım önerilmiştir; TADRAM, PADRAM ve AADRAM. 1 adet de içerik uyarlamalı/cell aware CADRAM tasarımı önerilmiştir. Alttaş kutuplama ile gerçeklenen tüm ADRAM tasarımları, temel DRAM tasarımı, ve seçilen literatür çalışmasının (Raidr tasarımı, [48]), karşılaştırmalı toplam yenileme sayıları sonuçları Şekil 3.21'de özet olarak sunulmaktadır (hücre içerik uyarlamalı (CADRAM) tasarımın sonuçları, tek hücre içeriğinin tek hücreye uygulandığı durum içindir).



Şekil 3.21 : Farklı ADRAM tasarımları için karşılaştırma sonuçları.

CADRAM tasarımında, birim hücre başına transistör eklendiği için, alan maliyeti problem olmaktadır. Bunu çoklu hücreye yaymak alan maliyetini düşürmektedir, ancak aynı zamanda başarımı da düşürmektedir (gelecekte çoklu içerik uyarlaması üzerine çalışma yapılacaktır), bu yüzden mevcut haliyle diğer tasarımlar önerilmektedir.

Diğer 3 tasarım (TADRAM, PADRAM, ve AADRAM) için; eğer alan maliyeti güç tasarrufu kadar önemliyse, gerilim ölçekli yöntem, eğer güç tasarrufu öncelikli ise alttaş kutuplama yöntemi ile gerçekleme önerilir. Çünkü, gerilim ölçekleme yönteminde artan gerilimden kaynaklı sağlanan toplam güç tüketimi düşüşü, alttaş kutuplama yöntemine kıyasla daha az seviyededir. Fakat, alttaş kutuplama tabanlı ADRAM tasarımlarının kutuplama ile ilgili eklentileri nedeniyle, gerilim ölçekleme yöntemine kıyasla alan maliyetleri daha fazladır.

Şekil 3.21'deki sonuçlara göre, herhangi bir ADRAM tasarım alternatifi en düşük etkili olanı bile DRAM tasarımının toplam yenileme sayısının %65'ine kadar düşüş sağlar. Aynı şekilde, herhangi bir tasarım literatürdeki ilgili çalışmaya kıyasla toplam yenileme sayısında daha fazla düşüş sağlar.

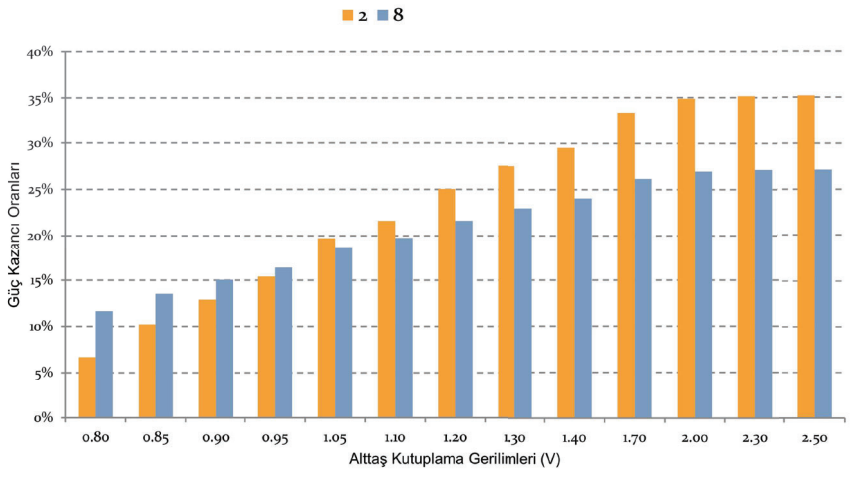
Dikkat edilmelidir ki bu tasarımlar sadece kavramın ispatı (proof of concept) niteliğindedir. Varsayımlara ve modellere bağlıdır, tasarımların üretime geçişi durumunda eksik/hata veya güncelleme gerekebilecektir.

4. MCSRAM: UYARLAMALI SRAM TASARIMLARI

4.1 Amaç, Motivasyon ve İlgili Çalışmalar

Güç tüketimi tüm VLSI devreler için en önemli problemlerdendir [2], yaygın kullanımdaki mimari yapılar olan bellekler için özellikle batarya kritik uygulamalar açısından düşük güç tüketimli tasarım arayışları devam etmektedir 2.3. Bu kapsamda devre seviyesinde hem durağan hem de devingen güç tüketimine odaklanan birçok çalışma gerçekleştirilmiştir [9–11, 61, 70]. Özellikle transistör üretim teknolojilerinin ilerlemesiyle SRAM'ler için durağan güç tüketiminin de oranı giderek daha da artmaktadır. SRAM için düşük güç tüketimi çalışmalarından biri de CASRAM [5], bit hücresi içeriği uyarlamalı SRAM tasarımıdır. Bu tasarımımız sayesinde hem sızdırma akımları hem de güç tüketimi azaltılabilmektedir. Ancak bu tasarımın alan maliyeti problemi bulunmaktadır.

Alan maliyetini düşürebilmek için, içeriğin birden fazla hücreye yayılması önerilmiştir. Ancak, bunun da problemi; içeriğin uygulandığı hücre sayısı arttıkça, devrenin başarımının düşmesidir ([17]'daki çalışmamdan faydalanılarak elde edilen Şekil 4.1 ile gösterilmektedir.). İşin kötü yanı 8 hücreden sonra çok hücreye uygulama mantığı çökmeye başlamaktadır, çünkü tekrarlı işaret bitleri genelde 8 bitten sonra görülmez. Bu da verilerin daha çok ayrışacağı anlamına gelmektedir. Diğer bir problem ise, bu çalışmamızda 12 erişim kapısı (port) için bu kıyaslamaları yaptık, ancak önbelleklerde erişim kapısı (port) sayısı düştükçe transistör sayısı düşeceği için ilave transistörlerin alan maliyeti oransal olarak artacaktır.

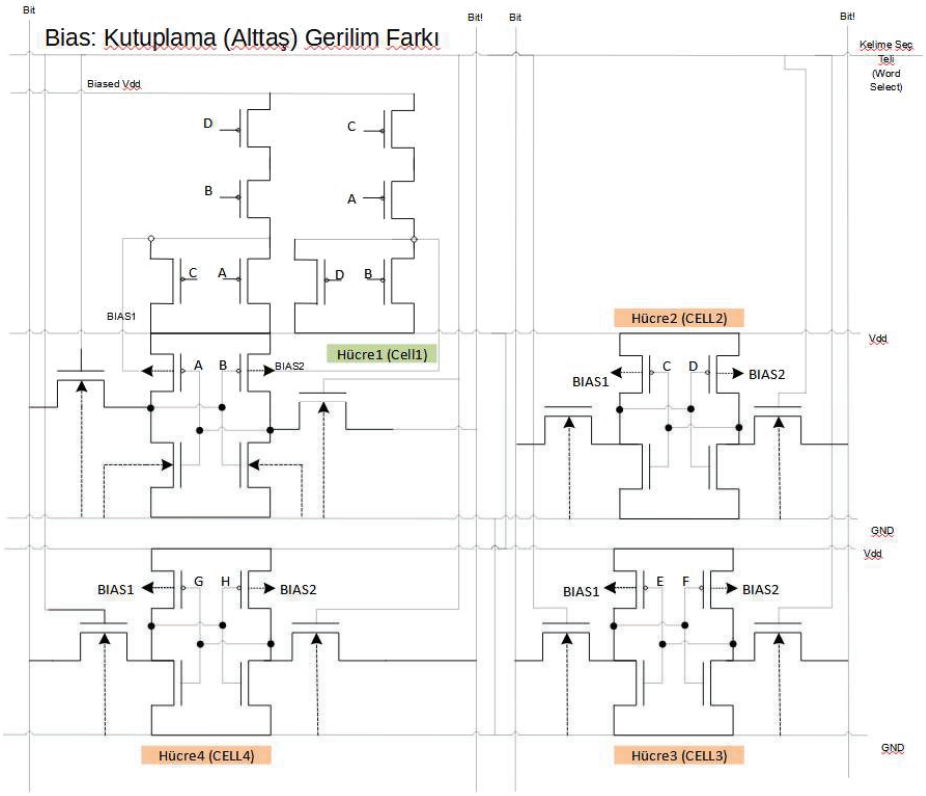


Şekil 4.1 : 2 ve 4 bit hücresi için güç kazanç oranları.

İşte, bu 2 gözleme dayanarak; tek içeriğin çoklu hücreye uyarlanması yerine, çoklu içeriğin çoklu hücreye uyarlanması tasarımı önerilmektedir. Doktora çalışmaları kapsamında geliştirilen bu tasarım; çoklu içerik uyarlamalı SRAM, MCSRAM, olarak adlandırılmaktadır. MCSRAM tasarımı sayesinde, çok sayıda hücreye uyarlama yapılarak alan maliyeti düşürülürken, birden fazla hücrenin içeriğine göre uyarlama yapıldığı için karar mekanizması başarımı artmaktadır. Çünkü, uyarlama kararı, diğer hücre içeriklerini daha yüksek ihtimalle temsil eder hale gelmektedir. Tasarımın nasıl gerçeklenebileceğine yönelik kavramsal devre tasarımı, ilgili açıklamalar ve sonuçlar, tezin devam eden alt bölümlerinde sunulmaktadır (Ayrıca, ilgili tasarım detayı ve sonuçların bir kısmı derlenerek, tezden türetilmiş bildiride sunulmuştur [71].).

4.2 MCSRAM Devre Tasarımı

MCSRAM'de birden fazla bit hücresinin içeriğini uyarlayarak (hangi bias gerilimi uygulanacağına karar vererek) bunu birden fazla hücreye uygulayan (ilgili komşuluktaki hücrelerin kutuplama hattına sürerek) devre tasarımı geliştirilmiştir. Bu tez kapsamında için kavram ispatı olarak önerilen 2 hücre içeriğini bir mantık kapısıyla uyarlayıp, 4 hücreye (PMOS transistörlerinin alttaş kutuplama gerilimlerine) uygulayan MCSRAM devre şematik tasarımı (temsili) Şekil 4.2'de gösterilmektedir.



Şekil 4.2 : Çoklu İçerik Uyarlamalı SRAM (MCSRAM) tasarımı.

MCSRAM şu şekilde çalışmaktadır: CELL1 ve CELL2 hücrelerinin içerikleri (A ve C, B ve D halinde) sol üstteki bloktaki uyarlama devresine gönderilir (aslında telle bağlıdır, görseli anlaşılır kılmak adına erişim kapıları/port'lar eklenmiştir). Bu devre içeriğe bağlı karar veren veya uygulanacak gerilimi seçen mantık kapısıdır. Daha sonra karar verildikten sonra yani 1. ve 2. hücrelerin içinde tuttukları "0" ve "1" durumlarına göre bias gerilimi veya default bias seçildikten sonra 4 komşu hücreye bu gerilim bias hatları üzerinden uygulanır. Burada arka arkaya çeviriciler olduğu için, 1 ve 0 tutarken farklı transistörlere farklı bias gerilimleri verilmektedir, biri bias biri default örneğin. BIAS 1 ve BIAS 2 bunu temsil etmektedir. 1 tutarken kapalı olan PMOS transistöre bias gerilimi uygulanarak düşük sızdırması sağlanır, aynı anda diğer PMOS açık olduğu için ona gerilim uygulanmaz default bırakılır. 0 tutarken de tersi geçerlidir. Bu örnek gösterimde, 2 hücrenin içeriği 4 hücreye uyarlanmıştır, ancak bu sadece anlatım içindir. Sonuçlar alınırken 64 bit okuma ve yazma konfigurasyonu uyarlanmıştır.

4.3 Metodoloji

Farklı mantık kapıları ve farklı bit hücreleri üzerinden, GEM5 açık kaynaklı sistem seviyesi ve işlemci benzetimcisi kullanılarak 64 bit önbellek mimari konfigurasyonunda SPEC denek taşlarıyla okuma ve yazma yapılarak sonuçlar alınmıştır.

4.4 Sonuç ve Değerlendirme

MCSRAM sayesinde 64 bit ve daha üstü bit hücresine çoklu içerik uyarlaması uygulanabilmektedir. Böylece hem alttaş kutuplamanın güç kazancındaki etkisindeki düşüşler engellenebilmiş olmakta, hem de alan maliyeti ihmal edilebilir seviyeye getirilebilmektedir. Çünkü bir kapı ve geçiş transistörleri ile 64 hücredeki tüm transistörlerin alttaş kutuplaması yapılabilmektedir önerilen tasarım sayesinde. Ancak, rasgele 2 bitin içeriğinin tüm hücrelere bağlanması yeterince iyi çözüm vermemektedir. Bu yüzden farklı kapılar için farklı bitlerin içeriğiyle okuma ve yazma işlemlerindeki 64 bite farklı denek taşları için "1" ve "0" yazılan/okunan bit örüntüleri çıkarılmıştır. Daha sonra tüm farklı konfigurasyonlar arasından başarımı en yüksek konfigurasyonlar tespit edilerek, MCSRAM tasarımına dahil edilmiştir. Başarımı yüksek ile kasıt, hangi bit ikilisininde diğer 64 bitte tutulan verileri daha çok temsil ettiğidir. Ayrıca, bu durumun gerçek uygulamalarda yaşanma mimari benzetimlerle bakılmıştır.

Mimari benzetimlerin koşturularak 3 farklı mantık ("VE" ve "VEYA" ve "EĞER (N. bit M. bit)) ile birlikte çok sayıda 2 bitin kombinasyonları denenmiştir. Denenen bit

kombinasyonlarından bazıları ise şu şekildedir:

- 0 ve 63. bitler (ilk ve son bitler)
- 0 ve 16. bitler (ilk ve ara değerdeki bit)
- 0 ve 32. bitler (ilk ve ortadaki)
- 31 ve 32. bitler
- 32 ve 63. bitler
- 15 ve 47. bitler (64 bitin iki yarısını ikiye bölen bitler)
- 16 ve 48. bitler

Tüm yoğun benzetimlerin ardından en iyi sonucun 15. ve 47. bitler ile çıktığı görülmüştür.

SONUÇ: Benzetim sonuçlarına göre, MCSRAM tasarımı sayesinde; en az %74 ihtimalle 64 bit hücresinin durağan güç tüketimi en az %35 düşürülebilir. Bunun alan maliyeti ise, 64 bit hücresine uyarlama yapıldığı için %1'in altına kadar düşürülebilmektedir,

Gelecek çalışmalarda bu fikrin DRAM ve diğer belleklerde de denenmesi planlanmaktadır.

5. ZORLU ORTAM KOŞULLARINDA GÜVENİLİR GERİLİM DÜŞÜRME

5.1 Amaç, Motivasyon ve İlgili Çalışmalar

Yapay zeka ve görüntü işleme uygulamaları, uç cihazlardan veri merkezlerine birçok modern kullanım alanına sahiptir. Görüntü işleme ve video işleme algoritmaları temelde derin öğrenme algoritmalarını kullanır ve bu algoritmaların en bilineni Evrişimsel Sinir Ağları (CNNs) dır. CNN'leri hızlandırmak için grafik işlemci birimi (GPU), FPGA (alanda programlanabilir kapı dizisi) ve ASIC (uygulamaya özgü tümleşik devre) donanımları kullanılabilmektedir [24, 72]. GPU'lar daha esnektirler; program çeşitliliği açısından uyarlanabilirdirler, ancak güç tüketimleri yüksektir. Güç tüketimi açısından, en uygun donanım, ASIC'lerdir. Ancak onlar da uygulama çeşitliliği açısından uygun değildir. Uygulama çeşitliliği ve güç tüketimi amaçları açısından, ara çözüm olarak en çok FPGA'ler tercih edilir. Ancak FPGA'ler için güç tüketimi halen ASIC'lere göre oldukça fazladır ve azaltılması gerekir. Özellikle batarya kritik mobil sistemler için bu daha da önem kazanır. Diğer taraftan CNNler aynı zamanda başarım kritik hassas uygulamalarda da çalışır, örneğin yolcu taşıyan sürücüsüz bir aracın kaza yapmasını kimse istemez [73]. Bu yüzden güç tüketimi için önerilecek tasarımların güvenilirliğe dikkat etmesi gerekir.

Güç tüketimini düşürmek adına en verimli yöntemlerden biri gerilim düşürme (undervolting) yöntemidir. Bu yöntemi uygulayan; FPGA'ler de dahil olmak üzere farklı donanımlar üzerine, birçok gerilim düşürme ve güç verimliliğini artırma çalışması bulunmaktadır ([15, 17, 24, 74–77]).

Gerilim düşürme, güç verimliliği açısından etkin bir çözümdür. Ancak, gerilim düşürmenin zamanlama hatalarına, gürültüye karşı dirençsizliğe ve devre gecikmelerine neden olduğu da bilinen bir olgudur [15, 41, 78]. Flip-floplar [79], bellekler [16], işlem birimleri ve diğer transistör tabanlı yapılar [57, 78] ise; zamanlama (timing) hataları oluştuğu durumda, doğru çalışmamaya başlarlar, veya gerilim düştükçe, gürültüye karşı daha dirençsiz hale gelmektedirler. Bu durumda, eğer frekans da gerilime göre ölçeklenmiyorsa, devrelerdeki, kapılardaki ve/veya iletimdeki gecikmeler nedeniyle hesaplamalarda, saklanan verilerde, iletilen veride, okuma veya yazma işlemlerinde ve anahtarlamalarda zamanlama uyumsuzlukları ve gecikmeler nedeniyle hatalar görülür. Benzer şekilde, derin öğrenme ve evrişimsel sinir ağları uygulamaları için de gerilim düşürmenin zamanlama hatalarına ve dolayısıyla doğruluklardaki kayıplara neden olduğu birçok çalışma ile gösterilmektedir [15]. Bu

çalışmaların en bilinenlerinden olan; [15] çalışmasında, derin öğrenme uygulamaları için uygulamaya özgü bir donanım kullanılarak, hata olasılığının besleme gerilimi düşüşüyle arttığı ispatlanmaktadır. Üstelik farklı girdi kümeleri (input data/image set) ve farklı uygulamaları için de bu düşüşün geçerli olduğu da ortaya konulmuştur. Dolayısıyla, gerilim düşürme ile güç verimliliği artarken hatalar, ve güvenilirlik endişeleri de artmaktadır. Bu kapsamda, iki temel yöntem izlenmelidir: Gerilim düşürülürken, hata tespit eden mekanizmalarla hatanın tespiti ([15] çalışmasındaki gibi) ve frekans ölçekleme, hata doğrultma kodları ([80] vb. tekniklerle önlenmesi, veya gerilim düşürülürken gerilimin ortam şartı ve uygulamaya özgü sınırlar dahilinde güvenilir seviyede ayarlanması.

Tez kapsamında; CNN hızlandırıcı FPGA'ler için gerilim düşürme ile doğrulukların nasıl değiştiğine yönelik deney sonuçları, bu sonuçların zorlayıcı koşullar altında değişimi ve bu sonuçlar dikkate alınarak geliştirilen güvenilir gerilim düşürme tasarımları sunulmaktadır. Sunulan sonuçlar, gerçekleştirilen deney ve geliştirilen tasarımlar derlenerek 2 akadamik çıktı ortaya konmuştur [17, 24]. (Not: Doktora çalışmalarım kapsamındaki çalışmalarımdan bir kısmının biraraya geldiği [17] yayını, içindeki sonuçlarla birlikte, ilgili doktora çalışmalarımın sonuçlarını anlatmak üzere tezin bu bölümünde sunulmaktadır [17]. Diğer akademik çıktıdan ([24]) bir sonuç görseli de karşılaştırma ve gerilim bölgelerinin anlaşılması için tez kapsamında aktarılmaktadır.) [24]'de sunulan çalışmada; CNN hızlandırıcı FPGA'lerde gerilim düşürmenin doğruluklar üzerine etkisine bakılmıştır. Bu çalışma, doktora çalışmalarım kapsamında parçası olduğum ortak çalışmanın bir çıktısıdır. Bu çalışmada, gerilim düşürmenin haricinde, farklı mimari seviye eniyileme teknikleri de denenmiştir. Bunlardan biri, nicemleme ("quantization") yani kayan nokta çözünürlüğünü tam sayı çözünürlüğüne indirmektir. Farklı çözünürlüklerde, CNN uygulama doğruluklarının değişimi gözlemlenmek istenmiştir. Örnek olarak, 8 bit nicemleme çözünürlüğünde doğruluklarda kayıp olmadığı görülmüştür. Bu sonuç doğrultusunda ve güç tüketimi açısından daha verimli tasarımlara ulaşabilmek adına, devam eden çalışmalarda da bu eniyileme konfigurasyonu denenmiştir.

Tez kapsamında, bir kısmı derlenerek [17]'de sunulan çalışmada ise, farklı zorlayıcı ortam koşullarında gerilim düşürmenin doğruluklara etkisi karakterize edilmiştir. Bunun için, kapsamlı, uzun süren ve kontrollü deney tasarımı ile gerçekleştirilen deneyler gerçekleştirilmiştir. IEEE Micro tarafından kabule layık görülen bu çalışmamızda, önceki çalışmamızdan, [24], farklı olarak; hassas bir iklim kabininde -40 ve 50 °C arasında kontrollü sıcaklık testleri yapılmıştır. Bu testler için her bir sıcaklıkta kabinin ve test biriminin şartlanması beklenir ve koşumlar tekrar tekrar yapılarak güç ve doğrulkuk sonuçları alınmıştır. Ayrıca, [24] çalışmamızda; farklı gerilimlerde farklı denek taşları için CNN'leri koşturup güç ve doğruluk sonuçlarını

alırken güç sonuçları izlenerek sonuçlar toplanmıştır. Diğer taraftan, [17] çalışmasında, bu sonuçlar gerçek zamanlı I2C arayüzü üzerinden kaydedildiği (logging) için daha kesin sonuçlar elde edilebilmiştir. Daha önceki çalışmamızda [24], herhangi bir sıcaklık kabini olmadan fanı tak çıkar yaparak oda sıcaklığı ve üstü sıcaklıklarda kontrolsüz testler yapılarak etkinin kaba çıkarımı yapılmıştır, [17] çalışmamızda ise; hassasiyeti yüksek bir kabinle, daha geniş bir sıcaklık aralığında ve test ortamı şartlandırılarak, "karakterizasyon" yapılmıştır. Ayrıca, [17] çalışmamızda; bilindiği kadarıyla literatürde ilk kez, CNN hızlandırıcı olarak kullanılan bir FPGA için farklı nem koşullarında, gerilim düşürmenin doğruluklar üzerindeki etkisine bakılmıştır. Son olarak, önemli farklarından biri olarak, [17] çalışmasında; 3 farklı özgün, sıcaklık uyarlamalı, güvenilir gerilim düşürme tasarımı da [17] ile ilk kez sunulmuş ve sonuçları tartışılmıştır. (Bu iki çalışma ve çıktılar haricinde; tezde sonraki bölümde ele alınacak olan son çalışmamızda ise; aynı sıcaklıkta ve aynı gerilimde uygulamayı iteratif çalıştırmanın etkisine, ve bu iteratif çalıştırma sırasında geçici olarak aralarda farklı gerilim uygulamanın etkisine bakılmaktadır. Yoğun deneyler ve kontrollü deney ortamında bu etkiler karakterize edilmektedir, ve bu karakterizasyon sonuçlarına dayalı güvenilir gerilim düşürme tasarımları önerilmektedir.)

Özetle; tezin bu bölümünde şu soru sorulmaktadır, "evet, güç verimliliği önemli ama güç tüketimini düşürürken güvenilirlik riske girer mi?" CNN hızlandırıcı FPGA'ler için gerilim düşürme yöntemine güvenilirlik açısından güvenebilir miyiz? Devam eden alt bölümlerde, bu sorulara cevap olarak gerçekleştirilen kapsamlı deneyler için deney tasarımı yaklaşımı, karakterizasyonu sağlayan deney sonuçları, ve üstelik bu karakterizasyon çalışmalarına dayalı önerilen özgün güvenilir gerilim düşürme tasarımları anlatılmaktadır.

5.2 Metodoloji

Tez kapsamında, CNN uygulamalarında gerilim düşürmenin doğruluklar üzerindeki etkisine bakarken, CNN uygulamalarının çıkarım/iterasyon/yineleme ("inference/iteration") kısmına odaklanılmaktadır, çünkü bu kısım sürekli tekrar eder. Tekrar eden bir kısımda gücü düşürmek etkin sonuçlar verebilecektir. Ancak, öğrenme kısmı için de önerilen tasarımlar uygulanabilirdir. Ayrıca, önerilen gerilim düşürme tasarımlarında sıcaklık uyarlamalı en uygun gerilim seviyesinin belirlenebilmesi için ön koşumlar gerçekleştirilmelidir. Bu ön koşumlar, öğrenme sırasında veya öğrenmeden sonra iterasyonlar başlamadan hemen önce yapılabilir.

CNN uygulamalarını FPGA'de çalıştırmak için derin yapay sinir ağları tasarım kiti, DNNDK [81], kullanılır. FPGA için ise gerilim ölçekleme yapabildiği için Zynq

tabanlı ZCU102 FPGA kartı kullanılır [36]. Hem doğruluk hem de güç tüketimi sonuçları, uygulama (CNN) çalışırken gerçek zamanlı alınabilmektedir. Bu erişim, I2C arayüzü üzerinden sağlanmaktadır.

Deneylerin gerçekleştirildiği, Xilinx ZCU 102 kartı içerisinde gerilim düşürme uygulanabilen 2 temel hat; VBRAM ve VCCINT hatlarıdır. Tez kapsamında, toplam FPGA güç tüketimine kıyasla oldukça düşük güç tüketimine sahip (güç verimliliğini artıran birçok tekniğin hali hazırda uygulanıyor olması sayesinde) BRAM yapıları yerine, doğrudan PL tarafı hesaplama birimlerini (flip floplar, mantık kapıları ve hesaplama birimleri) besleyen VCCINT hattı üzerinden gerilim düşürmeye odaklanılmıştır. Beklendiği üzere, bu hat üzerinde gerilim düşürme zamanlama hatalarına ve dolayısıyla CNN uygulamaları açısından doğruluklarda kayıplara neden olmaktadır, ve tez kapsamında istenen doğrulukları sağlayacak güvenilir gerilim düşürme tasarımları hedeflenmektedir.

Önerilen tasarımların ve paylaşılan gözlemlerin, tek bir uygulama için geçerli olmadığından emin olmak gerekmektedir. Uygulamadan uygulamaya gerilim düşürmenin etkisi değişecektir, gözlem ve tasarımların seviye ve başarımları da farklılık gösterecektir. Ancak, diğer uygulamalarda da gözlemlerin/fenomenlerin geçerliliğinin devam etmesi, tasarımların uyarlanabilir olması beklenmektedir. Ayrıca, önerilen gözlem ve tasarımların herhangi başka bir araştırmacı tarafından da tekrarlanabilir olması hedeflenmektedir. Bu doğrultuda, tez kapsamında gerilim düşürme çalışmaları için en bilinen (state-off-the-art) CNN denek taşlarından olan; GoogleNet [22], VggNet [50], ve ResNet [21], ile deneyler gerçekleştirilmiştir. Bu denek taşları, veya CNN algoritmaları, her sene düzenlenen ImageNet Geniş Ölçekli Görsel Algılama Yarışması (ILSVRC) tarafından en başarılı görülen algoritmalardandır. Bu algoritmalar, farklı katman sayıları ve mimari yapıda tasarlanmaktadır. Bu algoritmaları, birbiri ile kıyaslamak için de öğrenme ve test resim/image kümesi/seti tanımlanmaktadır.

Algoritmalar, standart, hazır ve herkes tarafından ulaşılabilir girdiler üzerinden çalıştırılarak, uygulama doğruluklarına bakılmaktadır. Örneğin, GoogleNet ve VggNet, Cifar-10 olarak adlandırılan veri kümesine göre farklı doğruluk başarımına sahiptir. GoogleNet için literatürdeki (top-1) doğruluk yaklaşık %91 iken, VggNet için bu oran %87 olmaktadır. Bu doğruluklar, şu şekilde hesaplanmaktadır: Veri seti için belirlenen kaç sınıf varsa, test girdi kümesi ile gerçekleştirilen CNN çıkarımlarında elde edilen sınıfların doğru bilinenlerinin sayısı ile, toplam sayı oranlanır. Eğer yeni bir girdi için; örneğin bir kedi resmi olsun, kedi sınıfında olduğunu bilirse algoritma başarılı olmuş olur, eğer başka bir sınıfta; örneğin araba, sınıfında derse de o resim/girdi için başarısız olmuş olur. Cifar-10 veri kümesi için toplamda 10

sınıf/kategori bulunmaktadır, ve algoritmalar girdileri bu sınıflara adresler.

GoogleNet ve VggNet farklı katman sayılarında, aynı input seti üzerinden farklı doğruluklar veren iki denek taşıdır. Peki, önerdiğimiz tasarımlar ve aktardığımız gözlemler farklı girdilere göre nasıl değişmektedir? İşte bu soruya cevap verebilmek için de, üçüncü denek taşımız olan ResNet'i kullanıyoruz. ResNet ile ILSVRC'2012 veri kümesi üzerinden doğruluklar hesaplanır, ve literatürde bu değer yaklaşık olarak %76 olarak çıkar. ILSVRC'2012 veri kümesi, Cifar-10 veri setinden daha kapsamlı bir veri setidir, ve toplamda 1000 sınıf/kategoriden oluşmaktadır. Dolayısıyla, hem farklı mimari ve katman sayısına sahip, hem farklı girdiler üzerinden referans doğruluk sağlayabilen 3 farklı CNN uygulaması seçilmiş olunmaktadır.

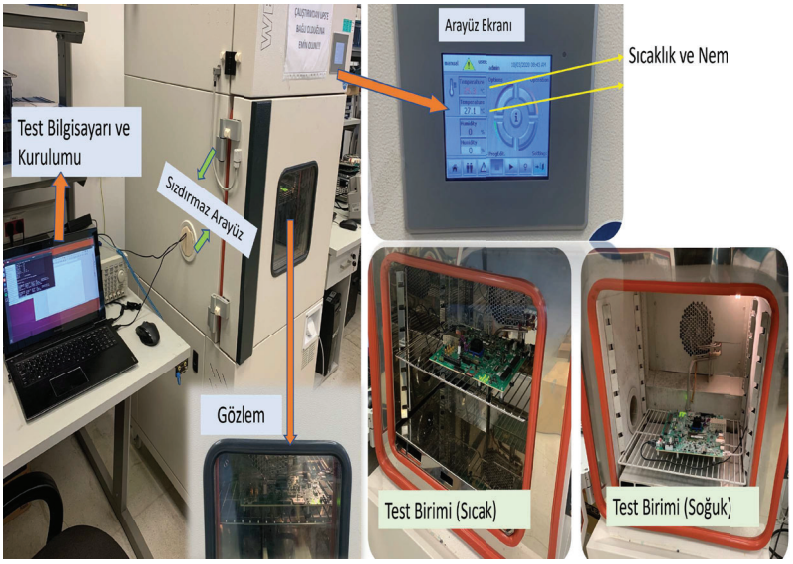
Önerilen tasarım ve aktarılan gözlemlerden önce, referans doğrulukların elde edildiğinden emin olunmaktadır. Örneğin, GoogleNet için 850 mV'da %91 doğruluk alınmaktadır. Tüm denek taşları için doğruluklardan kayıp olmadan veya benchmarkın en yüksek doğruluk seviyesi demek, literatürdeki bu referans doğruluklardır. Tez kapsamındaki çalışmalarda bu denek taşları için elde edilen doğruluk değerleri ile literatürde bildirilen değerler aynıdır (Bu paragrafda anlatılan açıklamalar, bir sonraki bölümdeke anlatılan çalışmalar için de geçerlidir.). Son olarak, deney sonuçları ilerleyen alt bölümlerde verilecektir, ancak ön değerlendirme olarak; gerilim düşürmenin etkisi ve zorlayıcı koşullarda bu etkinin değişimi bu denek taşları ile denendiğinde, geçerliliğini koruduğu görülmüştür. Önerilen tasarımlar da, bu uygulamalar arasında uyarlanabilirdir.

Önerilen tasarım ve aktarılan gözlemlerin tek bir karta özgü olmadığını ispatlamak için, tüm deneyleri ayrıca birden fazla özdeş FPGA kartla (ZCU102) da tekrarlanmıştır. Buna göre, aktarılan gözlemlerin bozulmadığı ve önerilen tasarımların geçerli olduğu görülmektedir.

Son olarak, farklı sıcaklıklarda test için kullandığımız hassas ve kalibre edilmiş sıcaklık ve nemi kontrollü olarak ayarlayabilen iklimlendirme kabini ve test kurgusu Şekil 5.1 ile verilmektedir. Sıcaklık için -40 ve 50 °C arası, nem için ise (50%, 60%, 70%, 80%) nem seviyelerinde testler gerçekleştirilmiştir.

5.3 Gerilim Düşürme Tasarımları ve Deneysel Sonuçlar

Sonuçlara başlamadan önce, gerilim düşürme (undervolting) çalışmalarının dayandığı gerilim bölgeleri anlaşılmalıdır. Öncelikle FPGA üreticileri FPGA'lerin çok farklı ortamlarda ve çeşitli uygulamalarda çalıştırılabilmesi için ve üstelik güvenilir tarafta da kalarak bir nominal besleme gerilimi belirler. Örneğin, [36] FPGA kartı için bu

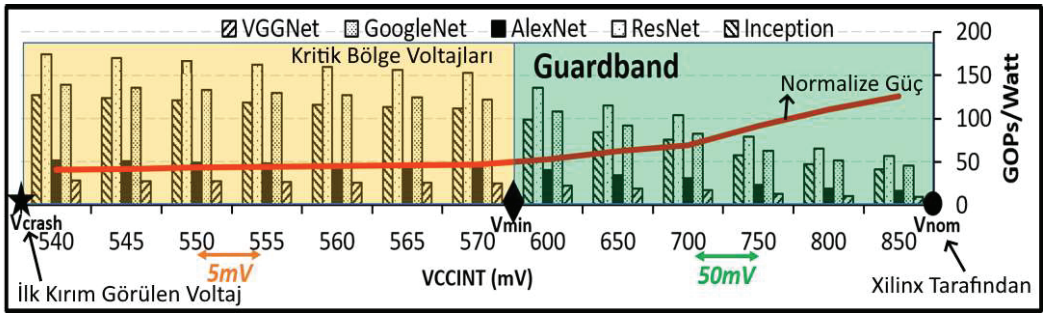


Şekil 5.1 : Test kabini ve sıcaklık ve nem test kurulumu.

VCCINT hattı için 850 mV olarak belirlenmiştir. Ancak aslında bu gerilim seviyesi en kötü durumda bile oldukça yüksek kalabilmektedir. Bu nedenle gerilim düşürme yapılır, ancak bu gerilim düşürme seviyesinin de bir limiti vardır. Çünkü bu limitten daha düşük bir gerilim uygulanması durumunda doğruluklarda bu sefer kayıplar yaşanacaktır.

Doğrulukların kayıp görmediği, güvenli gerilim bölgesine "guardband region" denir. Bu bölgedeki voltajlara da guardband voltajları denilebilmektedir. Guardband region içerisindeki en düşük limit voltaj ise "Vmin" olarak adlandırılır. Diğer taraftan, Vmin altında gerilim düşürülmeye devam edebilir. Bunun için doğruluklarda azalma yaşandığı bilinmelidir ve kabul edilebiliyorsa bu seviye altına inilebilir. Bu seviyenin altında gerilimler düştükçe doğruluklar azalır, ancak bir süre sonra artık kırım/crash görülmeye başlar. İşte Vmin ile ilk kırım/crash görülen bu bölgeyede kritik voltaj bölgesi denir. Doğruluklarda düşüş istenmiyorsa guardband bölgesindeki voltajlar, biraz düşüş kabul edilebiliyorsa kritik bölge voltajlarına gerilim düşürme yapılabilir. Bu sayede güç verimliliği (Giga ops per Watt: Watt başına Giga operations/işlem) artırılmaktadır. Guardbandde doğruluk düşmezken güç verimliliği azalan gerilimle artmaktadır. Kritik bölgede ise kırım olana kadar doğruluklar gerilim azaldıkça azalmaktadır, ancak bu bölgede de azalan gerilimle güç veirmliliği artmaktadır. Buna dair gerçek donanımla; ZCU 102 FPGA kartı üzerinden, farklı CNN denek taşları ile gerçekleştirilen deneylere ait sonuçlar Şekil 5.2'de sunulmaktadır (bu sonuç/şekil [24] çalışmamızda yer almaktadır).

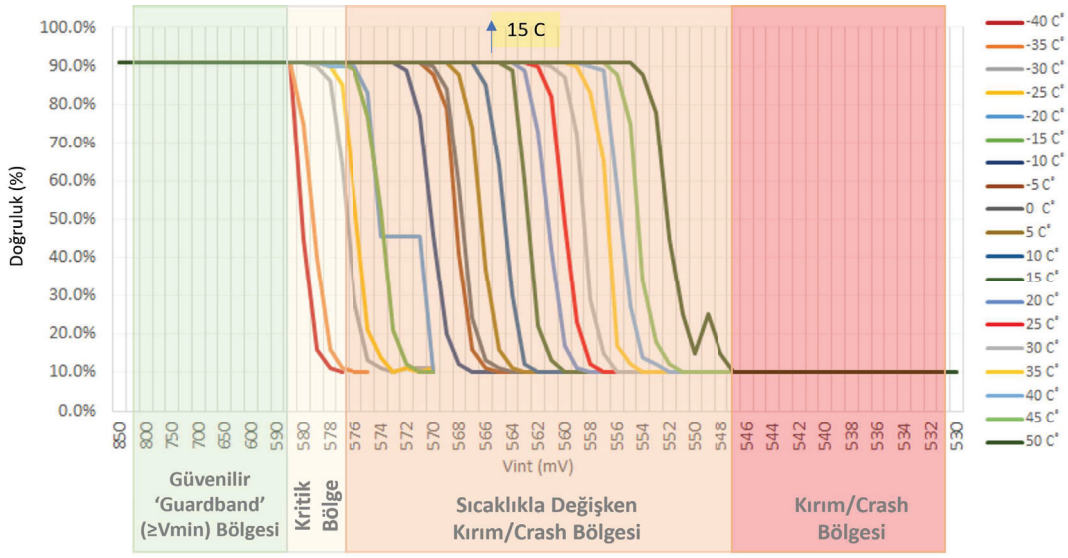
Not: Tezde bu bölümde anlatılanların bir kısmı derlenerek, doktora çalışmaları kapsamında ve tezin çıktısı olarak; [17] makalesi ve [24] bildirisi yayınlanmıştır, ve ayrıca da HPCA 2023'e bir bildiri gönderilmiştir.



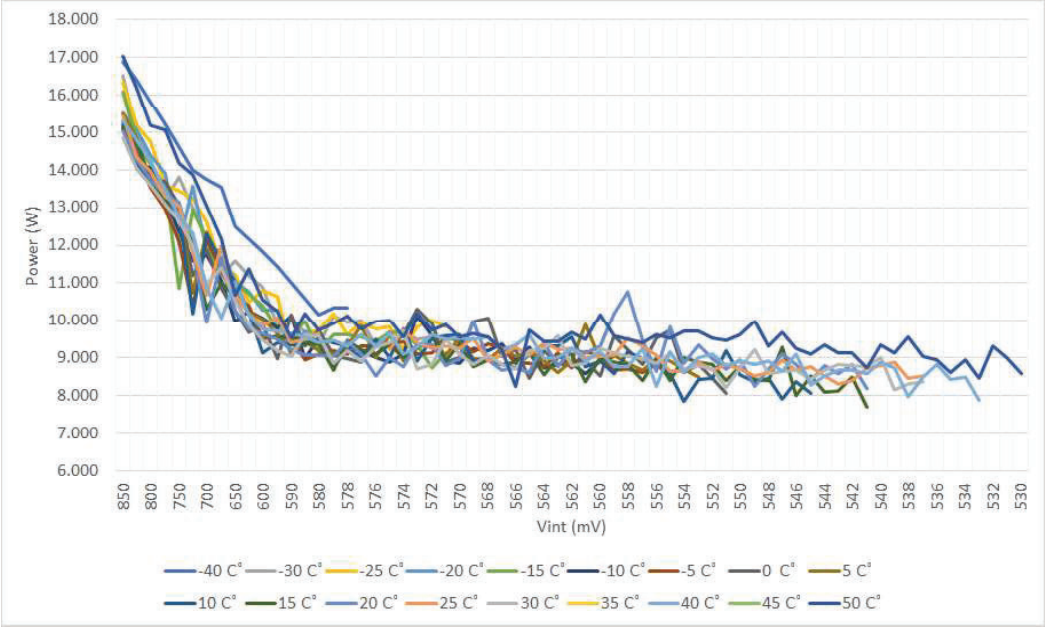
Şekil 5.2 : Gerilim düşürme ile CNN'lerde güç verimliliği.

Şekil 5.2'ye bakıldığında 850 mV olan nominal/temel tasarım geriliminin 600 mV ve altına kadar indilebildiği (Vmin'e kadar) görülmektedir, ve VMin ve Vcrashözellikle işaretlenmiştir. Nominal ve Vmin arasındaki yeşil bölge ise Guardband dediğimiz bölgedir. Vmin altı, sarı bölge ise Vcrash'e kadar kritik bökgedir.

Farklı sıcaklıklarda (-40 ile 50 °C arasında 5 °C adım aralığı ile) ve bu sıcaklıkların her biri için; farklı gerilim seviyelerinde (850 mV ile 530 mV'a kadar) ve bu gerilimlerinde her birinde; CNN uygulaması (GoogleNet denek taşı [22]) gerçek FPGA (ZCU102 [36]) üzerinde koşturularak doğruluklar ve ayrıca güç değerleri (her CNN koşumundaki maksimum görülen güç tüketimleri) elde edilmiştir. Doğruluklar için sonuçlar Şekil 5.3'de gösterilmektedir [17]. Her bir sıcaklık koşulunda hem kabinin hem de FPGA'in belirlenen sıcaklık koşuluna gelmesi ve şartlanması için ayrıca beklenmiştir, sonra koşumlara başlanmıştır (bu deneyler scriptlere rağmen gerçekten uzun süren ve yorucu deneylerdir, nem sonuçları için de benzer durum geçerlidir, doktora kapsamında başarıyla tamamlanmış ve ortak çalışılan hocalar tarafından da bu başarı dile getirilmiştir, zaten de alanda saygın bir yerden kabul alabilmiştir). Güç tüketimi için sonuçlar ise; Şekil 5.4'de gösterilmektedir [17].



Şekil 5.3 : Farklı sıcaklıklarda gerilim düşürme ile CNN doğrulukları.



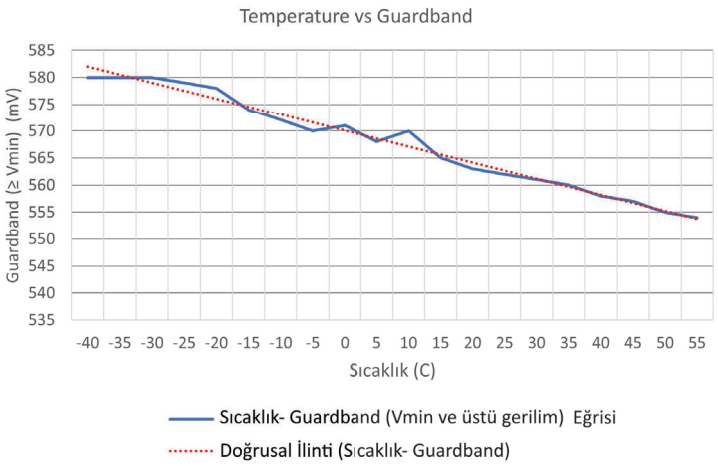
Şekil 5.4 : Farklı sıcaklıklarda gerilim düşürme ile güç tüketimi.

Şekil 5.3'de kırmızı bölge kesin kırım bölgesidir, ve en iyi sıcaklık durumunda bile 548 mV sonrasında crash region başlamaktadır. Turuncu ile gösterilen bölge ise sıcaklığa bağlı olarak kırım görülebilen, dolayısıyla sıcaklığa bağlı kırım bölgesi olarak adlandırılır, bölgedir. Bunun üzerinde ve 585 mV civarına kadar en kötü sıcaklık (en düşük sıcaklık) için kritik bölge olan açık turuncu bölge vardır. Vmin üzerinde ise, yani yeşil bölge ise, güvenilir guardband bölgesidir.

Şekil 5.3 incelendiğinde; güvenilir guardband ($\geq Vmin$) bölgesi/region haricindeki bölgelerde tüm sıcaklıklarda gerilim düşüşüyle doğrulukların azalması gerçeği gözlemlenebilmektedir. Ayrıca, sıcaklığın aynı gerilimde olmasına rağmen, CNN uygulama doğruluğunu etkilediği görülmektedir; sıcaklık azaldıkça doğruluklar düşebilmektedir. Bu nedenle de, sıcaklıkla gerilim bölgelerinin aralıkları değişmektedir. Örneğin guardband regionın Vmin voltajı veya guardband voltajı, 15 °C sıcaklık için 566 mV civarında olurken, eğer sıcaklık -15 °C'ye düşerse aslında o sıcaklık için geçerli guardband bölgesinin ve Vmin'in yukarı kaydığı ve 576 mV seviyelerinde olduğu görülmektedir. Aynı şekilde Vmin'den sonra da kritik bölge başladığı için, kritik voltaj bölgesinin de sıcaklıkla değiştiği gözlemlenmektedir.

Şekil 5.4'deki güç tüketimi sonuçlarına bakıldığında, gerilimin azalmasıyla beklendiği üzere (Bölüm 2.3'de gerilimin hem durağan hem de devingen güç tüketimini artırdığı ve açıklamalar bahsedilir.) güç tüketimi de düşmektedir. Ancak doğruluklarda olduğu gibi sıcaklığın güç tükeminin gerilimle değişimine doğrusal bir etkisi gözlemlenememektedir, daha çok bir dalgalanma şeklinde gerçekleşmektedir.

Sıcaklıkla guardband voltajı veya Vmin'in değişimi Şekil 5.5'de sunulmaktadır [17].

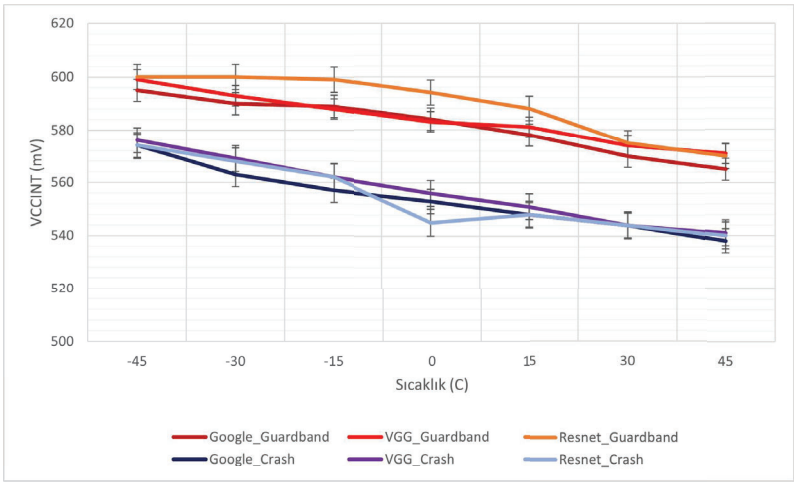


Şekil 5.5 : Sıcaklıkla 'Guardband' ($\geq Vmin$) değişimi.

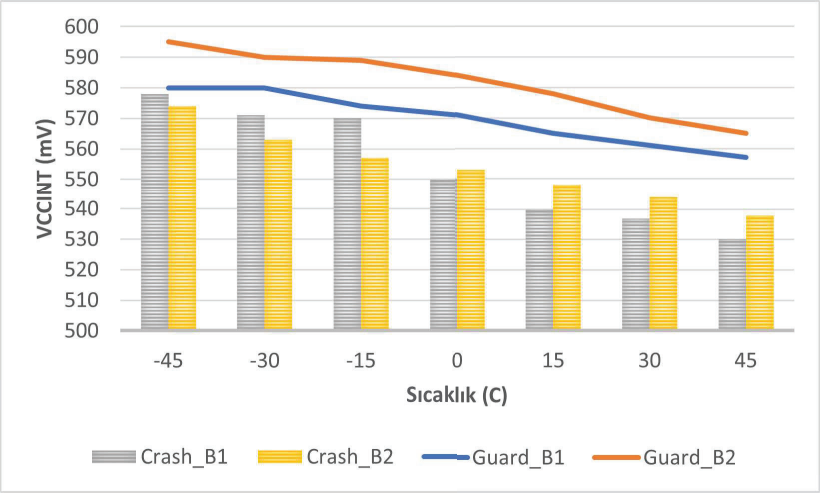
Her bir sıcaklık için Vmin seviyesi yukardaki sonuçlara göre elde edilmiş ve sıcaklıkla bu Vmin seviyeleri arasındaki ilişki bu eğri ile gösterilmiştir. Bu eğri üzerine model oturtulabilmektedir ve bu modelin fonksiyonu ($AAV = \alpha T + \beta = -1.483T + 583.4$) olarak çıkmaktadır. Vmin şu yüzden önemlidir, bizim hangi seviyeye kadar gerilim düşürme yapabileceğimizi söyler, o yüzden Şekil 5.5 ve bu fonksiyon oldukça önemlidir. Hangi sıcaklıkta hangi gerilim düşürme seviyesine kadar gelebileceğimizi bu fonksiyonla hesaplayabilir hale geliriz.

Peki sıcaklığın gerilim düşüşü ile doğruluklar arasındaki ilişkiye etkisi ve Vmin seviyelerine etkisi bu kuruluma özgü müdür? Yoksa, diğer denek taşları ve boardlar için de sıcaklığın gerilim düşürme üzerindeki etkisi benzer ve geçerli midir?

İşte bu sorunun cevabı, Şekil 5.6 ve Şekil 5.7 ile verilmektedir. Bu soruya cevap bulabilmek için; 3 farklı en bilindik CNN denek taşıyla, her bir denek taşı için farklı sıcaklıklarda, her bir sıcaklık için farklı gerilimlerde her bir gerilim için koşum



Şekil 5.6 : Farklı CNN uygulamaları için sıcaklık ve gerilim düşürme.



Şekil 5.7 : Farklı özdeş FPGA'ler için sıcaklık ve gerilim düşürme.

yapılarak doğruluklar elde edilmiştir. Deneylerde koşturulan CNN denek taşları; GoogleNet [22], VggNet [50], ve Resnet'dir [21]. Farklı CNN uygulamalarında, sıcaklığın gerilim düşürme üzerindeki etkisine yönelik sonuçlar Şekil 5.6'da yer almaktadır. Ayrıca, yine sorunun cevabı için, birden fazla özdeş FPGA kartı ([36]) ile de deneyler tekrarlanmıştır. Sonuçlar, Şekil 5.7'de gösterilmektedir [17]. Bu iki şekilde yer alan sonuçlara göre, üstteki sorunun cevabı evettir. Eğrilerin başladığı nokta (Y ekseni, gerilim) denek taşları arasında farketse de (Vmin ve Vcrash açısından), eğrilerin trendi benzerdir. Sıcaklıkla; Vmin seviyeleri de, Vcrash seviyeleri de düşmektedir. Üstelik, farklı girdi seti kullanılan ResNet için de bu trendin geçerli olduğu görülmektedir. Benzer şekilde, farklı özdeş FPGA'lerde (B1 ve B2: 2 özdeş FPGA kartıdır) de sıcaklığın gerilim düşürme üzerindeki etkisinin geçerli olduğu görülmektedir; ve artan sıcaklıkla Vmin ve Vcrash seviyeleri her iki FPGA kartında da benzer eğilimde düşüş göstermektedir.

Bu gözlemleri kullanarak; doktora çalışmaları kapsamında, CNN hızlandırıcı olarak FPGA'lerde güvenilir olarak gerilim düşürmeyi sağlayacak şekilde güç tüketimini azaltabilen 3 özgün gerilim düşürme tasarımı önerilir:

- En kötü duruma göre gerilim düşürme
- Uygulamaya özgü gerilim düşürme
- Akıllı veya uyarlamalı gerilim düşürme

En kötü duruma göre gerilim düşürme: Bu gerilim düşürme tasarımımızda, bir FPGA için kullanım alanı çeşitliliği veya potansiyeli düşünülerek belirlenen en kötü sıcaklığa göre (en soğuk koşul) guardband voltajı veya Vmin belirlenir, ve bu seviyenin

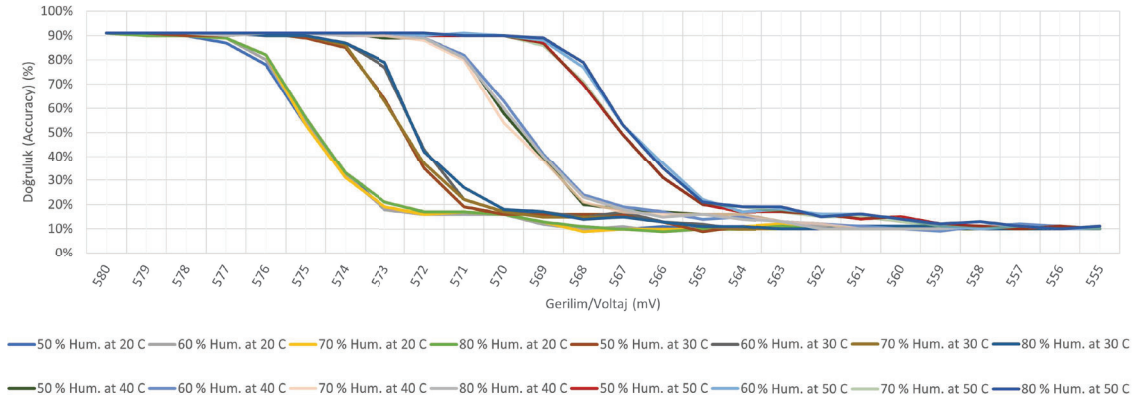
altına inilmeden bir voltajda gerilim düşürme yapılır. Tez kapsamındaki örnekte, Şekil 5.3'de gösterilen -40 °C sıcaklık en kötü sıcaklıktır. Bu tasarım -40 °C sıcaklık için doğrulukların etkilenmediği minimum voltajı bulur (Vmin) ve bunu uygular, böylece en kötü duruma göre tasarlandığı için diğer tüm sıcaklıklarda doğrulukların düşmediğini garanti eder. Hem bunu sağlarken hem de güç tüketimi verimliliğini artırır, bu sonuçlara göre; -40 °C sıcaklık için Vmin: 590 mV olduğunda, temel tasarıma göre (gerilim düşürülmeyen, nominal voltajlı) %65 artış sağlamaktadır.

Uygulamaya özgü gerilim düşürme: Bu gerilim düşürme tasarımımızda, tüm uygulamaları ve ortamları düşünerek en kötü duruma göre gerilim düşürme uygulamak yerine, eğer bir CNN hızlandırıcı FPGA sadece belirli bir ortamda çalıştırılıyorsa, uygulamanın çalışacağı ortamda sıcaklığın belirli bir alt ve üst limiti varsa bu limite göre gerilim düşürme uyarlanır. Örneğin; bir veri merkezinde sıcaklığın belirli aralıklarda tutulması durumu vardır. Diyelim ki bu sıcaklık en kötü 15 °C olsun [82], bu durumda Şekil 5.3'de gösterilen 15 °C sıcaklık için Vmin değeri bulunur ve uygulanır. Bu sayede, temel tasarıma göre %79-%80 civarında güç verimliliği artışı sağlanır, aynı zamanda da doğruluklarda kayıp olmayacağını bu koşullar altında garanti eder. Çünkü zaten o uygulamaya özgü en kötü şarta uyarlandığı için daha iyi şartlarda yani sıcaklıklarda zaten doğrulukların düşmeyeceği garanti altına alınmış olur.

Uyarlamalı gerilim düşürme: Bu gerilim düşürme tasarımımızda, belirli bir sabit koşula veya en kötü duruma göre gerilim düşürme uygulamak yerine tamamen değişken koşullara göre dinamik uyarlama yapabilen akıllı gerilim düşürme uygulanır. Bu tasarım, farklı sıcaklıklarda gerilimin doğruluklar üzerindeki etkisinin değişiminin veya guardband gerilimilerinin sıcaklıkla değişiminin modellenmesine dayalıdır.

Örneğin, Şekil 5.5 ve bu sonuçlardan elde edilen fonksiyon veya modeli kullanır. Bir uygulama çalıştırılacağı ortamda hangi sıcaklıkta ise o sıcaklık için gerekli Vmin'i veya guardband voltajı bulur ve uygular, böylece bir uygulama örneğin oda sıcaklığında çalışıyorsa ve bu iklimsel olarak 25 C altına inmeyecekse başka bir Vmin uyarlar, veya dış ortamda kullanılacak bir otonom araçta kullanılacaksa eksi bir sıcaklık geçerlidir, ona göre Vmin'i günceller, uyarlar. Bu tasarımımız, diğer iki tasarımdan daha fazla güç verimliliği elde etmek için ortaya konulmuştur. Ortam şartlarına (sıcaklığa) uyarlamalı olarak sağladığı güç verimliliği değişkendir, ve hedeflenen doğruluklara zarar vermeden bu güç verimliliği kazançlarını sağlamaktadır.

Farklı nem koşullarında; 3 farklı nem koşulunun her biri için farklı sıcaklıklarda; 4 farklı sıcaklığın her biri için farklı gerilimlerin her birinde CNN uygulaması koşturularak doğruluklar elde edilmiştir. Elde edilen sonuçlar Şekil 5.8'de yer almaktadır [17]. Bilinen kadarıyla ilk kez bu şekilde ve CNN hızlandırıcı FPGA'lerde nemin CNN doğruluklarına bakılmıştır, hatta FPGA'lerdeki diğer uygulamalar ve CNN



Şekil 5.8 : Farklı sıcaklık ve nem koşullarında gerilim düşürme.

hızlandırılar için de ilk olmaktadır.

Şekil 5.8 incelendiğinde; farklı sıcaklıkların her birinde nemin doğruluklar üzerinde iyileştirici etkisi olduğu ancak bunun çok küçük bir etki olduğu görülmektedir; bu iyileşme etkisi tüm sıcaklıklarda da görülebilmektedir. Bunun nemin sebep olduğu ısı sığası artışından kaynaklı olduğu düşünülmektedir. Tersten düşündüğümüzde de farklı nem koşullarında, sıcaklığın gerilim düşürme üzerindeki etkisinin devam ettiği ve geçerli olduğu görülmektedir. Hatta sıcaklığın etkisi oldukça büyüktür, nemin etkisi ise oldukça ihmal edilebilir seviyelere yakındır.

5.4 Sonuç ve Değerlendirme

Doktora çalışmalarının bir parçası olan bu çalışmalar (bir kısmı [17]'da yayınlanan ve burada anlatılan) Güvenilir (Reliable) bir Gerilim Düşürme (Undervolting) tasarımı koyabilmek için gerçekleştirilmiştir. Çok yoğun ve kapsamlı deneyler sonucunda, zorlayıcı ve çok geniş aralıktaki ortam koşullarında bu deneylerin tekrarı ile geniş bir sonuç seti elde edilmiştir. Bu sonuçlar tezin bu kısmında aktarılmıştır, ve bu sonuçlar kullanılarak; hızlandırıcı olarak kullanılan FPGA'lerde düşük güç tüketimi sağlarken hedeflenen başarımı da garanti eden 3 özgün gerilim düşürme tasarımı ortaya konmuştur.

Önerilen tasarımlar, uyarlamalı donanım tasarımlarıdır, ve tezin amacı olan "kendi kendine devre parametreleri uyarlayabilen donanım tasarımı" yaklaşımı bu tasarımlarda sergilenmiş kullanılmıştır. Bu yaklaşıma göre tasarlanmış 3 tasarımın herhangi biri ile, en az %65 ve şartlara bağlı olarak daha fazlası, güç verimliliği artışı sağlanmaktadır. Bu kazanç sağlanırken de, CNN uygulama doğruluklarının belirlenen koşullar altında düşmediği garanti edilmektedir.

Seçilen ve deneyler yapılan denek taşları sayesinde; tasarımların farklı girdi setlerine

uyarlanabilir olduğu da, farklı uygulamalarda geçerli olduğu da gösterilmektedir. Ayrıca, bunlar haricindeki bir girdi seti, veya uygulama için de tasarımların uyarlanabilir/uygulanabilir olduğu değerlendirilmektedir. Bunun için, herhangi bir girdi seti, veya donanım veya uygulama için özetle şu yaklaşım önerilmektedir: "İlk olarak, uygulamaya özgü bir sıcaklık limiti varsa belirle. Bu limite göre, bu kısımda anlatılan şekilde, gerilim düşürme seviyesini deneylerle (veya benzetimlerle) tespit et. CNN çıkarımları için artık bu gerilim düşürme seviyesinde uygulama yap. Eğer, belirli bir sıcaklık limiti yoksa, ya en kötü sıcaklık koşuluna göre gerilim düşürme seviyesini belirle ve uygulamaya başla, ya da tüm sıcaklıklar için gerilim düşürme seviyesini belirleyip, sıcaklığa göre uygun olan gerilim düşürme seviyesini uygula." Bu yaklaşım sayesinde, farklı sıcaklıklarda doğruluklardan ödün vermeden gerilim düşürerek güç tüketimi verimliğinide artış sağlanmış olmaktadır.

6. UYARLAMALI FPGA GERİLİM DÜŞÜRME TASARIMLARI

6.1 Amaç, Motivasyon ve İlgili Çalışmalar

Hava hedeflerinin teşhisi, veya havaalanlarında güvenlik amaçlı yüz tanıma, veya nesne tanıma gibi birçok uygulama için derin öğrenme algoritmaları ve bunun da temel tekniklerinden olan Evrişimsel Yapay Sinir Ağları (CNNs olarak anılacaktır) kullanılmaktadır, ve giderek daha da yaygın hale gelmektedir [18–23]. Bu nedenle de hem hassas kullanım alanlarında hem de mobil isterlerin öncelikli olduğu alanlarda da kullanılabilmektedirler, ve başarım ve güç tüketimi birbirinin önüne geçmeyecek iki hedef haline gelmektedir.

Güç tüketimini düşürmek için (Bölüm 2.3, 2.10) çeşitli çalışmalar gerçekleştirilmektedir [4, 5, 11, 56, 57]. Örneğin dinamik eşik değer gerilimi uygulamak, veya bunu uyarlamalı yapmak ve böylece durağan güç tüketimini azaltmak veya besleme gerilimilerini ölçeklemek veya kullanımda olmayan devrelerin anahtarlanması gibi. Bu tip devre veya transistör seviye çözümler güç tüketimi açısından oldukça etkin gözükse de, bu tip tasarım değişikliklerini rafta hazır ürünlerde sadece üretici gerçekleyebilir. Bu yüzden, FPGA'ler gibi ticari ve hazır ürünlerde düşük güç tüketimi sağlamak için hazır devre kabiliyetlerini kullanan teknik ve tasarımlar daha çok tercih edilir.

CNN uygulamalarının başarım ve güç tüketimi açısından hızlandırıcı donanımlarla desteklenmesi gerekir ve bunun için çeşitli donanımlar kullanılmaktadır.

- ASIC: CNN hızlandırıcı olarak kullanılmaktadır [83, 84], ve güç tüketimi açısından da oldukça etkindirler. Ancak, bu kadar uygulama çeşitliliği olduğu durumda ASICler uygulamaya özgü tasarlandıkları için verimsizliğe neden olmaktadırlar [85].

- GPU: ASIC'lere kıyasla uygulama çeşitliliğine uyarlama açısında oldukça esnek oldukları için CNN hızlandırıcı olarak tercih edilirler [54], ancak diğer taraftan da güç tüketimi açısından oldukça verimsizdirler.

- FPGA: FPGA'ler tam iki donanımın arasındadır, ASIC'lerden daha fazla esnektirler; yeniden konfigure edilebilirler. GPU'lardan ise daha az güç tüketirler, ve hızlandırıcı olarak giderek daha fazla kullanımda görülmektedirler [53–55].

CNN'ler için önemli olan güç tüketimi ve başarım CNN hızlandırıcı olarak kullanılan FPGA'ler için de geçerlidir ve gereklidir. Bölüm 2.3'de bahsedildiği üzere güç tüketiminde en önemli parametrelerden biri gerilimdir, çünkü hem durağan hem de devingen güç tüketimini etkiler. İşte anlatılan bu sebeplerden, gerilim ölçekleme özelliği bulunan ve CNN hızlandırıcı olarak kullanılan bir FPGA için en bilinen düşük güç tüketimi yöntemi gerilim düşürmedir (önceki bölümde de bahsedilmektedir, anlam bütünlüğü için burada yer almalıdır). CNN hızlandırıcı donanımlarında Gerilim düşürme uygulayan çeşitli çalışmalar bulunmaktadır: CPUs [86], GPUs [77], ASICs [87], and DRAMs [40]. Ayrıca doğrudan FPGA tabanlı CNN hızlandırıcılarda gerilim düşürme üzerine de birçok çalışma yapılmıştır [15–17, 24]. Bu yöntem güç tüketimini düşürmektedir, ancak Vdd'nin düşmesinin devre tasarımı açısından da etkileri bulunmaktadır, gecikmelere ve bu nedenle hatalara neden olabilir ve aynı zamanda gürültüye karşı dayanıklılık azalmış olur, gibi. Bu nedenle gerilim düşürmenin başarımı etkilemediğinden veya CNN özelinde konuşacak olursak hedeflenen doğrulukları bu yöntemin garanti etmesi beklenir.

CNN'ler aslında görüntü işleme vb. işlemler için her yeni görüntü veya değişiklikte tekrar tekrar çağrılmaktadırlar [18, 55, 72], buna inference de denilmektedir. İteratif çalıştırıldıkları için CNN'lere yönelik önerilecek herhangi bir çözümün bu iteratif çalışmaya dikkat etmesi beklenir. Gerilim düşürme için de benzer durum geçerlidir, gerilim düşürme doğrulukları bu iterasyonlara rağmen garanti edebilmelidirler, ancak biz doktora kapsamında yapılan ve tezin bu bölümünde anlatılan bu çalışmaların başında burada bir hata kaynağı tespit ettik.

Bir önceki alt bölümde bahsedildiği üzere, güvenilir bir gerilim düşürme (undervolting) yapılabilmesi için uygulanabilecek bir limit vardır, bu limit "guardband voltaj" olarak veya "Vmin of guardband region" olarak adlandırılmaktadır. Vmin'in altında doğrulukların düşmeye başladığı bilinmektedir. İşte doktora çalışmalarım sırasında keşfedilen fenomen şudur:

"Bir CNN uygulaması aynı düşük gerilimde ve aynı sıcaklık koşulundayken eğer iteratif olarak çalıştırılmaya devam ederse, ilk koşumda doğrulukta düşüş görülmese bile, iterasyon devam ettikçe doğruluklarda düşüş gözlemlenmeye başlamıştır". Bu gözlem, iterasyonun yıkıcı etkisi (DIE) olarak adlandırılmıştır. Bilinen kadarıyla, literatürde ilk kez, CNN uygulamaları için bu gözlem tanımlanmaktadır, karakterize edilmektedir, ve üstelik bu etkiyi indirebilecek çözümler sunulmaktadır. (Not: Tezin bu bölümünde anlatılan çalışmalar derlenerek bildiri olarak gönderilmiştir.)

Gözlemlenen diğer bir fenomen ise şu şekilde özetlenebilir;

"İterasyonun yıkıcı etkisi görülen aynı düşük bir voltajda, bir CNN uygulaması iteratif

koşturulmaya devam ederken, eğer geçici olarak uygulanan voltajdan daha yüksek bir voltaj uygulanacak olursa bu DIE kaynaklı doğruluklardaki düşüşü azaltabilmektedir." Bu fenomen de "onarıcı etki" (RE) olarak adlandırılır bu çalışma kapsamında.

Bu 2 keşif farklı özdeş FPGA kartları ve farklı CNN denek taşlarıyla ile yapılan deneylerde de gözlemlenmeye devam edilmektedir ve geçerlidir. Bu gözlemleri ve bunu ortaya koyan karakterizasyon sonuçlarını kullanarak ise 3 farklı özgün güvenilir gerilim düşürme tasarımı önerilmektedir tez kapsamında. Bu tasarımlar bu etkileri kullanan ve uyarlamalı gerilim düşürme yapabilen tasarımlardır. Tezin amacı olan "bir girdiye bağlı devre parametrelerini kendi kendine değiştirebilir bir uyarlamalı donanım tasarımı" yaklaşımını başarıyla uygulayabilmektedirler.

Uyarlamalı ve akıllı tasarım sayesinde; bu üç tasarımın herhangi biri, en az %43 güç verimliliği artışı (GOPS/W cinsinden; Watt başına giga işlem) sağlamaktadır, ve bunu CNN uygulamalarının iterasyonlarına rağmen ve DIE etkisine rağmen CNN uygulama doğruluklarında herhangi bir düşüşe sebep olmadan başarabilmektedirler. Ayrıca bu tasarımlar, farklı CNN hızlandırıcı donanımları, farklı FPGA'ler veya farklı CNN uygulamaları açısından da uyarlanabilirdir, bu şekilde tasarlanmışlardır ve tezdeki metodoloji de bu doğrultuda tekrar herhangi başka bir araştırmacı tarafından uygulanabilecek şekilde sunulmaktadır.

6.2 Metodoloji

CNN uygulamalarını çalıştırmak için Xilinx tarafından sağlanan Derin Öğrenme İşlemci Birimi (DPU) IP'sini kullanıyoruz Vivado ortamında. DPU'lar FPGA'de programlama mantığı tarafında yer alırlar, işlemleme sistemi tarafıyla da direkt konuşabilirler, ve bir DPU CNN uygulamasını çalıştırmak için eniyilenmiştir [88]. DPU'lar BRAM'leri kullanırlar ve kullanılacak BRAM boyutlarının seçilebilen konfigurasyonları bulunur, bu çalışmada en geniş olanını seçtik [88]. DPU'lar yine Xilinx tarafından sağlanan Derin Yapay Sinir Ağları Tasarımı Kiti ile çalışırlar. Bu kit, hesaplamalardan ve evrişimlerden sorumludur.

Bir derin öğrenme uygulaması için önce güç tüketimi ve başarım açısından iyileştirici tedbirler uygulanır, bunlardan biri quantizasyondur. İşlemler sırasında hesaplamaların kayan nokta (floating point) değil, tam sayı (integer) olarak hesaplanmasına dayalıdır [89]. Daha sonra, öznitelik/özellik çıkarımı (feature extraction) yapılır, bu aslında evrişimlerle yapılır, yani filtreleme ve matris çarpım işlemleri anlamına gelir. Daha sonra da sınıflandırma veya tanıma işlemleri başlar, bu kısım artık iteratif devam eder. Çıkan alternatiflerin ihtimalleri hesaplanarak en yüksek ihtimalli seçenek çıktı olarak sunulur. Tüm bu işlemler DNNDK tarafından sağlanır [81].

DNNDK, tüm donanımlarla uyumlu değildir, ancak bu çalışmalarda kullandığımız Xilinx ZCU102 FPGA [36] ile ise uyumludur. Bu donanım, hem bu yüzden hem de gerilim ölçekleme kabiliyeti nedeniyle seçilmiştir.

CNN parametreleri, ağırlıklar vb. ve girdiler FPGA içerisinde DDR4 DRAM bellek üzerinde bulunurlar. PL tarafındaki işlemler bellek üzerinden aktarılan verilere ve komutlara göre gerçekleşir. Gerilim düşürme yöntemi FPGA üstünde, PL tarafındaki işlemleri gerçekleyen birimler (sayısal sinyal işlemci, flip-floplar, bufferler vb.)'in güç besleme hattı olan VCCINT'e yapılmaktadır. Diğer bir hat olan BRAM besleme hattı ise, BRAM'in güç tüketiminin toplam güç tüketimine oranla oldukça az olması nedeniyle uygulamaya gerek görülmemiştir. Çünkü, daha önce de bahsedildiği üzere; BRAM'ler için zaten güç tüketimini düşürmeye yönelik birçok teknik ve tasarım uygulanmaktadır. Deneyler sırasında her CNN koşumu sırasında gerçek zamanlı olarak güç tüketimi değerleri ve koşum neticesinde ise doğruluk sonuçları alınır, kaydedilir, bu işlemler için i2c arayüzü kullanılır.

Seçtiğimiz FPGA; ZCU 102 Xilinx [36], 16nm transistör teknoloji noktasındadır, nominal olarak 850 mV gerilim beslenmektedir, DNNDK ile çalışabilmektedir ve gerilim ölçeklemesi yapabilmektedir. Bu çalışma kapsamında frekans ölçeklemesi yapılmamıştır, sabit önceden tanımlı frekansta gerçekleştirilmiştir tüm deneyler. Farklı sıcaklık koşullarında DIE ve RE etkilerini gözleyebilmek ve karakterize edebilmek için hassas ve kontrollü sıcaklık kabini kullanılmaktadır 5.1. En kötü (-45 °C) ve en iyi (45 °C), ve ara değer sıcaklıklarında (-15 °C ve 15 °C), her bir gözlem noktası veya test adımında, CNN uygulamaları çok sayıda iterasyon ile çalıştırılmaktadır.

Koşumlar için özellikle farklı CNN denek taşları kullanılmaktadır [21, 22, 50]. Bu uygulamalar herkes tarafından ulaşılabilen ve koşturulabilen ve en bilinen CNN denek taşlarıdır. Bu denek taşlarının, girdi/input kümeleri de herkes tarafından bilinen ve herhangi bir CNN algoritması için denenebilen girdilerdir. Koşumlarda alınan en yüksek doğruluk değerleri literatürdeki ile aynıdır. Googlenet için %91, Vgg için %87, ve Resnet için %76 dır. Bu çalışma için uygulama çeşitliliğinin etkisini görebilmek amacıyla, ilgili üç denek taşı; özellikle farklı büyüklükte, farklı girdi setlerinde çalışan ve farklı katman sayısında tasarlanmış olan denek taşları seçilmiştir.

Önerilen herhangi bir gerilim düşürme tasarımı için DIE etkisini hesaba katan ve eğer kullanıyorsa RE'yi sağlayan, ve aynı zamanda bu 2 etkiyi gözlemlemeyi de sağlayan genelleştirilmiş bir algoritma sözde kod olarak sunulmuştur. Bu sayede bu algoritmayı uygulayarak bu çalışmada önerilen tasarımlar farklı bir donanım ve uygulamaya da uyarlanabilirdir. Bu nedenle de, algoritma bilinçli olarak parametrik bırakılmıştır. Güvenilir gerilim düşürme tasarımları veya mevcut tasarımların analizi için deneysel akışı görselleştiren algoritma sözde kod olarak Algoritma 1 ile verilmektedir.

Algoritma 1 Güvenilir gerilim düşürme için sözde kod

```
i2c arayüzünü ayağa kaldır, gerilim düşürme ortamını ilklendir
Parametrelere atama yap
for Vrej = Vrej_max → Vrej_min do
    NoI_Vre ← numberofCNNIterations at Vrej
  for Vcrv = Vcrv_max → Vcrv_min do
    NoI_Vcrv ← numberofCNNIterations at Vcrv
      for NoI_Vre = Max_NoI_Vrej → Min_NoI_Vrej do
          for NoI_Vcrv = Max_NoI_Vcrv → Min_NoI_Vcrv do
              for RPN = 1 → PatternRepeatNumber do
                  if Vrej ≠ Vcrv then
                      for i_rej = 1 → NoI_Vrej do
                          logEnergy
                          denek taşı koştur
                          if fail then Vrej ← Vnominal
                              break
                          else if success then logAccuracy
                          end if
                          öteleme süresi
                      end for
                  end if
                  for i_crv = 1 → NoI_Vcrv do
                      logEnergy
                      run benchmark
                      if fail then V_crv ← Vnominal
                          break
                      else if success then logAccuracy
                      end if
                      öteleme süresi
                  end for
              end for
          end for
      end for
  end for
end for
Her gerilimde doğruluk ve güç değerlerini ayrıştır/kaydet
```

Algoritmada kullanılan terimler tasarım ve deney akışı açısından anlamlarıyla şu şekildedir:

- Vrej: Gençleştirici (Rejuvenating) gerilim, Die görülen voltajda uygulama iterate ederken geçici uygulanan yüksek voltajdır. Bu voltajın maksimum ve minimum limitleri vardır, bu limit aralığında voltajlar değişerek, bu voltaj aralığındaki her voltajda NoI kadar CNN iterasyonu döner.

- Vcrv: İterasyonun yıkıcı etkisi gözlemlenen gerilim bölgesindeki herhangi bir voltajdır. Kritik bölge voltajı olarak adlandırılır. Bu voltajın maksimum ve minimum limitleri vardır, bu limit aralığında voltajlar değişerek, bu voltaj aralığındaki her voltajda NoI kadar CNN iterasyonu döner.

- NoI (NumberofCNNIterations): Iterasyon sayısıdır. NoI of Vrej, Vrej geriliminde kaç iterasyon CNN çalışacağını belirler. NoI of Vcrv ise, Vcrv geriliminde kaç iterasyon CNN çalışacağını belirler. Her iki NoI tipi için de max ve min limitleri vardır, bu limit aralıklarında iç döngüler döner. Örneğin maksimum 3 ise NoI of Vcrv için, bu durumda önce 3 kere iterasyon döndüğündeki durum, sonra 2 sonra 1 olacak şekilde iç içe döngüler çalışır.

- Örüntü/Pattern: Bir ardışık gerilim dizisidir. Bir örüntü/pattern 3 bileşenden oluşur bu çalışma için: 1. Vcrv, 2. Vrej, 3. NoI of Vcrv. Her bir RE örüntüsü/pattern'i aslında bu parametrelerle gençleştirici etkinin ne kadar baskın uygulanacağını belirler. Çünkü, Vrej artarsa gençleştirici etki artar ancak güç tüketimi verimliliği azalır. Ödünleşimi bu parametre belirler. Bir RE örüntüsü/pattern'i şu şekildedir: Vrej x Vcrv x NoI of Vcrv, ve şöyle örneklenebilir: "850&575x2". Bu şu anlama gelir; önce 850 mV'da bir CNN iterasyonu, ardından 575 mV ile 2 CNN iterasyonu, ardından 850 mV da bir CNN iterasyonu 575 mV da 2 CNN iterasyonu diye devam ederek bu örüntü/pattern tekrar eder.

- PatternRepeatNumber: Bir örüntünün/pattern'in kaç kere tekrar edeceğini ifade eder. Örneğin bir üst maddede örüntü/pattern 2 kere tekrar etmiş oldu.

Bu algoritma ve parametrelerle DIE karakterizasyonu, RE karakterizasyonu ve bu iki keşfe dayalı 3 tasarım için sonuçların alınıp en uygun gerilim düşürme limitinin belirlenmesi sağlanır, ve bu herhangi bir tasarım, donanım ve uygulama için de gerçeklenebilirdir.

6.3 İterasyonun Yıkıcı Etkisi, DIE (Keşif-1)

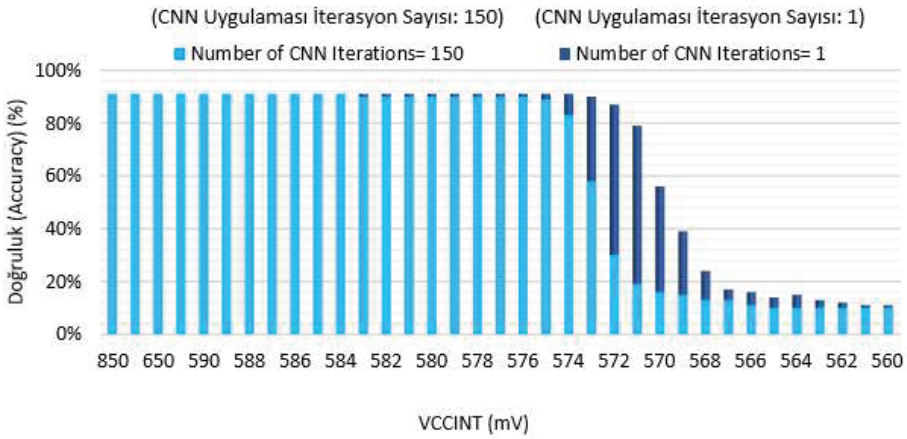
Bu etkiyi karakterize etmek için ilk akla gelen soru; bu etkinin tüm voltajlarda mı yoksa belirli voltajlarda mı meydana geldiğidir. Bunu cevaplamak için, Algoritma 1'de şu değişiklikler gerçekleştirilerek tekrarlı ve kapsamlı deneyler gerçekleştirilmiştir:

- Vrej: 850 mV olarak ayarlanmıştır. Vrej max ve min: Eşit olarak 850 mV olarak atanmıştır, kısaca Vrej sabit tutulacak anlamına gelmektedir.
- Vcrv: Maksimumu 850 mV ve minimumu 560 mV olarak belirlenmiştir.
- NoI of Vrej: 100 olarak belirlenmiştir. Aslında deneyin amacı Vcrv'de DIE etkisi gözlemlemek, ancak bu döngü sırasında Vcrv voltajlarındaki doğrulukların önceki Vcrv voltajından etkilenmemesi için her döngü sonunda 850 mV nominal voltajda koşullandırma yapılmak istenmektedir.
- NoI of Vcrv: 150 olarak belirlenmiştir. Dolayısıyla, her Vcrv geriliminde 150 kere CNN iterasyonu çalıştırıldığında doğrulukların değişimi gözlemlenmek istenmektedir.

Böylece belirlenen aralıkta her Vcrv geriliminde 150 kere CNN iterasyonu koşup sonra koşullandırma için Vrej'de de 100 kere CNN iterasyonu devam edecek, ve bir sonraki Vcrv gerilimine geçilerek bu şekilde iç içe döngüler algoritma 1'deki gibi devam edecektir.

Bu yöntemle gerçek donanım üzerinden iteratif deneylerle elde edilen sonuçlar, Şekil 6.1'da sunulmaktadır. Bu şekilde, her bir voltaj (VCCINT) için iterasyon sayısı 1 (NoI=1) olduğunda ve 150 (NoI=150) olduğunda doğrulukların değişimi gösterilmektedir.

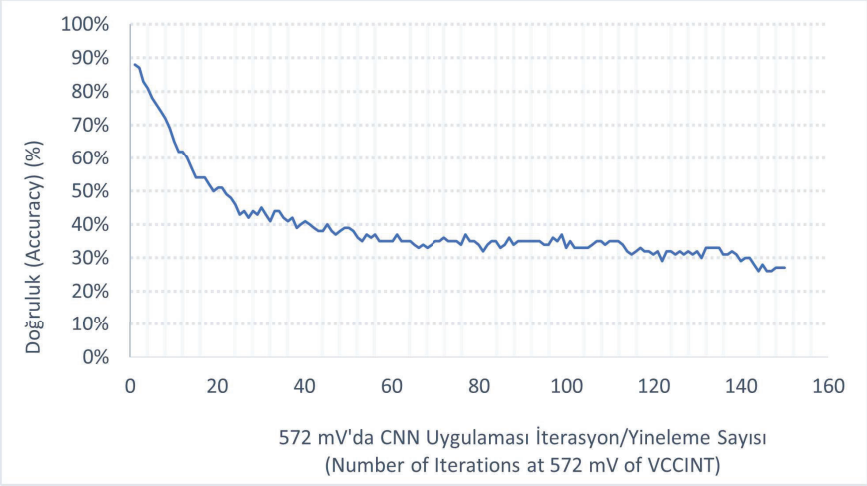
Şekil 6.1 incelendiğinde; iterasyonun yıkıcı etkisinin her gerilim seviyesinde görülmediği gözlemlenmektedir. 585 mV ve altında DIE kendini göstermeye başlamaktadır, bundan daha yüksek gerilimlerde iterasyonun yıkıcı etkisi görülmemektedir denilebilir. CNN algoritması için kullanılan veri kümesi için elde edilebilen en yüksek doğruluk, iterasyona rağmen 586 mV için halen elde edilmektedir. Bu sonuç, iterasyona rağmen belirlenecek bir gerilim seviyesinde gerilim düşürme yapılabileceğini göstermektedir. Ayrıca, yineleme sırasında görülen doğruluk kayıplarının, doğruluk hesaplamalarında yapılan bir yanlışlıkla olmadığı da ortaya çıkmaktadır. Diğer taraftan, gerilim azaldıkça da etkinin yıkıcılığı artmaktadır. Etkinin en çok görülebilir olduğu gerilim 572 mV'dur, bu gerilim seviyesinde 1 kere CNN iterasyonu gerçekleştiğinde neredeyse tepe seviye doğruluk alınırken, CNN



Şekil 6.1 : Farklı gerilimlerde CNN iterasyonunun doğruluklara etkisi.

iterasyonları devam ettikçe doğruluğun azaldığı ve 150. iterasyonda ise doğruluğun üçte birine düştüğü görülebilmektedir.

İterasyonun yıkıcı etkisini en belirgin gözlemlediğimiz 572 mV için doğrulukları incelediğimizde, doğrulukların iterasyon sayısı/number of iterations ile azaldığı ancak belirli bir iterasyondan sonra artık iterasyonun yıkıcı etkisinin azalmaya başladığı görülmüştür. Buna ait görsel Şekil 6.2'de sunulmaktadır. Bu şekilde paylaşılan sonuçlara göre, doğruluk ilk iterasyondan 20. iterasyona kadar yaklaşık yüzde doksanlardan yüzde kırklara kadar düşmüştür, sonraki iterasyon adımlarında ise bu düşüş hızı yavaşlamış ve durma noktasına gelmiştir, ve 20. ve 150. iterasyonlar arasında sadece %10 daha düşmüştür. Bu düşüş modellenebilir mi diye diğer gerilimler de incelenmiş ancak, düşüşün eğrisinin ve hızının gerilimden gerilime farkettiği gözlemlenmiştir. Dolayısıyla, bir gerilim seviyesi için modelleme yapılabilirdir, ancak farklı sıcaklık durumları hesaba katıldığında tüm gerilimler için ortak modelleme yapılabilir olmadığı görülmüştür.



Şekil 6.2 : İterasyon sayısı (NoI: number of iterations) ve doğruluklar.

Doğruluklar alınırken güç tüketimi sonuçları da elde edilmiştir. Güç tüketimi ile ilgili sonuçlar incelendiğinde; beklendiği üzere, gerilimle güç tüketiminin arttığı görülmüştür. Ancak iterasyon sayısıyla güç tüketimi arasında doğrudan bir ilinti kurulamamıştır.

DIE fenomeni ve bunun belirli gerilim aralıklarında gerçekleştiği deneysel sonuçlarla ortaya çıkarılmıştır. Peki, bu etki neden gerçekleşmektedir? Öncelikle, tez kapsamındaki çalışmalar ve önerilen tasarımlar, DIE'ı bir amaç için kullanmayı önermemektedir. Aksine, DIE görülmeyen güvenilir bir gerilim seviyesinin seçilmesini (veya DIE'a karşı bir sonraki alt bölümde anlatılacak olan çözümlerin denenmesini) önermektedir. Ayrıca, bu fenomenin gerçekleşmeye başladığı gerilim seviyesi, farklı uygulama veya donanıma göre değişeceği için, bu farklılıklara da uyarlanabilir şekilde tasarımlar ortaya konulmuştur. Dolayısıyla, DIE'ın neden gerçekleştiğinden ziyade, gerçekleştiği ispatlanan (farklı girdi resim/veri kümesinde çalışan farklı uygulamalarda ve farklı özdeş FPGA kartlarında da denenerek) bu fenomeni dikkate alarak, gerilim düşürme tasarımının yapılması gerektiği aktarılmaktadır. Sorunun cevabı ise şu şekilde değerlendirilmektedir:

- Bölüm 2.3'de de bahsedildiği üzere, devrelerin üzerinde transistörler ve yollar kaynaklı sığa olduğunu, veya devreleri besleme gerilimi açısından yük olduğunu düşünebiliriz. Bu iki anlama gelmektedir; eğer devreleri besleyen güç hattına mevcut durumdan daha düşük bir gerilim uygulanırsa, hemen uygulanmaya başlar başlamaz, sığa (devre) nedeniyle devreler o gerilim seviyesine ulaşmaz. Belirli bir süre geçer, boşalarak o seviyeye gelir (capacitor discharging). Veya, tam tersi de geçerli; daha yüksek bir gerilim uygulandığında o yük seviyesine doğru dolmaya başlar, belirli bir süre geçince o seviyeye ulaşır.

- ZCU102 FPGA kartına, gerilim düşürme sırasında hangi gerilim uygulanırsa uygulansın, gücü kesilip tekrar açıldığında, nominal gerilimi olan 850 mV'a geri döner. Dolayısıyla, ilk açılış gerilimi 850 mV olmaktadır. Daha sonra gerilim düşürme uygulandıkça, literatürde ispatlandığı ve kendi çalışmalarımızda da gösterildiği üzere, CNN uygulama doğruluğu düşmektedir.

- Gerilim düşürme uygulanan VCCINT hattı üzerindeki tüm devreleri sığa kaynağı olarak yorumlarsak, ilk açılışta 850 mV olarak beslenen ve 850 mV'u gören devreler, daha düşük gerilimde sürülmeye başlansa bile hemen yükünü boşaltmamaktadır. CNN uygulamasının iterasyon sayısı arttıkça, devreler, devrelere uygulanan gerilim düşürme seviyesine gelmeye başlar ve durur (o seviye geçerli olmuş olur).

- Benzer mantıkla, düşük gerilim seviyesindeki bir devreye eğer yüksek gerilim

atılırsa, bu da devreleri belirli bir süre sonra yukarı sürecek, ve tekrar eski gerilim seviyesine dönülse bile etkisi devam edecektir (bir sonraki alt bölümde anlatılmakta olan, RE ve bunu kullanarak geliştirilen tasarımlar da buna dayanmaktadır.).

Bu değerlendirmemizin geçerliliğini gözlemlemek amacıyla; kritik bölge voltajında (572 mV gerilim seviyesinde), bir CNN uygulaması (GoogleNet) için iterasyon sayısını 1000'e kadar çıkardık ve doğrulukları kaydettik. Buna göre, önceki kısımlarda da açıklandığı üzere 20 iterasyona kadar hızlı düşüş, 20-150 arası giderek yavaşlayan düşüş görüyoruz doğruluklarda. 150-1000 arasında ise artık düşüş değil, nerdeyse kararlı bir eğri gözlemliyoruz. Çünkü, bu aralıktaki artan iterasyon sayısı ile doğrulukların değişimi düşüş değil, çizgi üzerinde standart sapma davranışı göstermektedir. 150-1000 arasında ise; iterasyon sayısı ile değişen doğrulukların standart sapması yaklaşık %1.9 olmaktadır. Devreler artık gerçekten sürülmek istenen gerilim seviyesini görmüş denilebilir. Tüm kritik bölge voltajlarında, düşüş eğrisi 572 mV gerilim seviyesindeki gibi olmamaktadır, ancak iterasyonun sayısıyla belirli bir seviyeden sonra düşüş görülmediği gözlemlenmiştir. Değerlendirmenin geçerliliğini daha net gösterebilmek adına ve aynı zamanda DIE'ın farklı uygulamalarda (CNN harici) geçerli olduğunu da gösterebilmek için, gelecek çalışmalar arasında FPGA üzerinde kendi tanımladığımız hesaplamaları iteratif tekrarlarken gerilim düşürmenin etkisine yönelik deneyler gerçekleştirilmesi de planlanmaktadır.

Önceki bölümde sıcaklığın gerilim düşürme üzerinde oldukça etkili olduğunu göstermiştik, peki iterasyonun yıkıcı etkisi sıcaklıkla nasıl değişiyor? Bunu gözlemlemek için, Algoritma 1 üzerinde şu değişikliklerle yinelemeli deneyler gerçekleştirilerek sonuçlar alınmıştır:

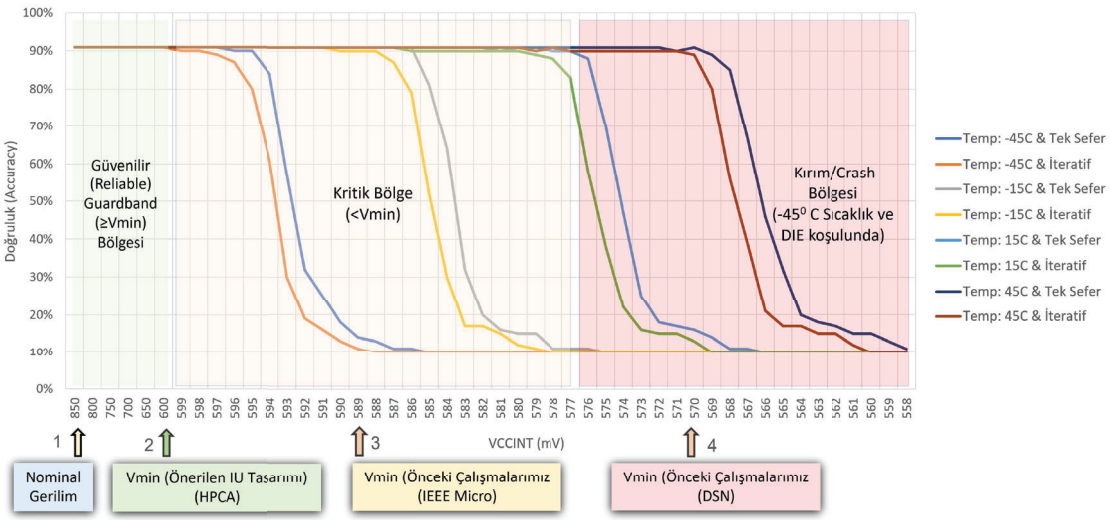
- Vrej: 850 mV olarak ayarlandı. Vrej max ve min: Eşit olarak 850 mV olarak atandı, kısaca Vrej sabit tutulacak anlamına gelmektedir.
- Vcrv: Maksimumu 600 mV ve minimumu 560 mV olarak belirlenmiştir.
- NoI of Vrej: 60 olarak belirlenmiştir. Aslında deneyin amacı Vcrv'de DIE etkisi gözlemlemek, ancak bu döngü sırasında Vcrv voltajlarındaki doğrulukların önceki Vcrv voltajından etkilenmemesi için her döngü sonunda 850 mV nominal voltajda koşullandırma yapılmak istenmektedir.
- NoI of Vcrv: 40 olarak belirlenmiştir. Dolayısıyla, her Vcrv geriliminde 40 kere CNN iterasyonu çalıştırıldığında doğrulukların değişimi gözlemlenmek istenmektedir.

Bu yöntemle elde edilen sonuçlar, Şekil 6.3'de sunulmaktadır. Sonuçlar incelendiğinde; guardband bölgesinin sıcaklık azaldıkça daraldığı ve Vmin voltajının yukarı kaydığı gözlemlenmektedir. Aynı şekilde iterasyonun yıkıcı etkisinin ortaya çıktığı ilk nokta da sıcaklıkla değişmektedir. Bu şekil üzerinde ayrıca; bu tezde de bahsedilen önceki çalışmalarımdaki Vmin gerilimleri de işeratlenmiştir, ki "eğer iterasyonun yıkıcı etkisi göz önüne alınmadan bir gerilim düşürme yapılırsa bu hataya neden olur" olgusunu ortaya çıkarılabilsin. Buna göre, [24](4)'deki çalışmamızda hata olmadan inilebilecek diye belirlenen Vmin gerilim seviyesinde (oda sıcaklığı ve üstü sıcaklıklara göre ve iterasyon etkisi ihmal edilerek belirlendiği için) iterasyosyona devam edildiğinde kırım bile görülebildiği gözlemlenmektedir. Aynı şekilde, çok geniş aralıkta sıcaklık koşulları altında sonuç alsak ve buna göre gerilim düşürme tasarımı koysak da, bir önceki belirtilen tasarımımızda [17] (3), bazı sıcaklıklarda yineleme etkisiyle doğruluklarda kayıp olabileceğini gözlemliyoruz.

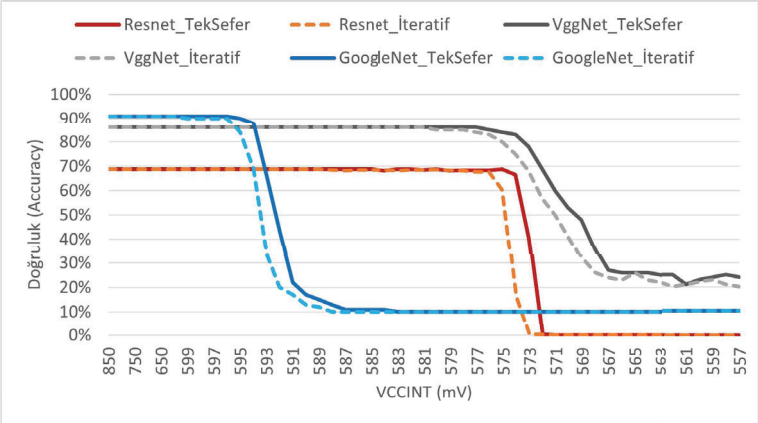
Son olarak, bu sonuçlara bakılarak; 850 mV (1) nominal voltajda kalmaktansa, 600 mV (2) ve yukarısında belirlenecek bir guardband bölgesi veya Vmin'in 600 mV yapılması, bir gerilim düşürme tasarımı için iterasyonun yıkıcı etkisine rağmen ve en soğuk koşulda en kötü duruma rağmen doğruluklarda düşüş olmamasını sağlamaktadır.

İterasyonun yıkıcı etkisinin farklı CNN uygulamalarıyla nasıl değiştiğini gözlemlemek için, bir önce uyguladığımız deney konfigurasyonu ile Algoritma 1 üzerinden deneyler farklı CNN denek taşları için tekrar edilir. Bu yöntemle elde edilen sonuçlar Şekil 6.4'de sunulmaktadır.

Şekil 6.4'de yer alan sonuçlar incelendiğinde; tüm denek taşları için iterasyonun yıkıcı etkisinin geçerli olduğu ve görülebildiği gözlemlenmektedir. Sadece bu etkinin görülebildiği yer ve guardband voltajı denek taşından denek taşına biraz



Şekil 6.3 : Farklı sıcaklıklarda, iterasyonun yıkıcı etkisi.



Şekil 6.4 : Farklı denek taşları için iterasyonun yıkıcı etkisi.

değişebilmektedir. Eğer 599 mV üstü bir gerilim (600 mV) Vmin olarak seçilirse, bu seçim; tüm denek taşları ve tüm sıcaklıklar için, iterasyonun yıkıcı etkisine rağmen, doğruluklardaki kayıpları önleyebilir gözükmektedir.

6.4 Gençleştirici (Rejuvenating) Etki, RE (Keşif-2)

Bu alt bölümde, iterasyonun yıkıcı etkisine karşı bir gerilim düşürme yöntemi tarafından yararlanılabilecek "Gençleştirici Etki (Rejuvenating Effect), RE" için yapılan deneyler ve sonuçlarından bahsedilmektedir. RE, DIE görülen herhangi bir gerilim seviyesinde CNN iterasyonları koşturulurken/devam ederken geçici olarak o gerilimden daha yüksek bir gerilimin uygulanmasıyla sağlanır. Algoritma 1'de bu etkiyi gözlemleyebilmek için şu değişiklikleri yaparak deneyler gerçekleştirilmiş, ve hem güç verimliliği, hem de doğruluklar için sonuçlar alınmıştır:

- Vrej: 850 mV olarak ayarlanmıştır. Vrej maksimumu; 850 mV olarak, Vrej mimimumu ise; 572 mV olarak ayarlanmıştır. Böylece, en düşük voltajda Vcrv ile eşitlenmiş; RE olmaksızın, 572 mV da CNN iterasyonu devam etmiş olacaktır.

- Vcrv: 572 mV olarak, maksimumu ve minimumu da 572 mV, kısaca Vcrv, 572 mV da sabit tutulmak istenmektedir.

- NoI of Vrej: 1 olarak belirlenmiştir, ardışık olarak Vrej ve Vcrv olması için.

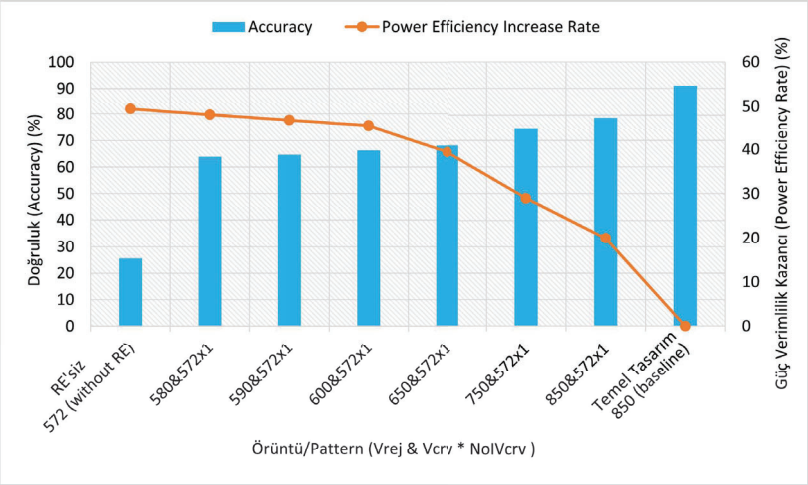
- NoI of Vcrv: 1 olarak belirlenmiştir, ardışık olarak Vrej ve Vcrv olması için.

- Pattern repeat Number: 10 olarak belirlenmiştir, yukardaki ardışık tekrar 10 kere tekrar edecek şekilde.

Bu yöntemle alınan sonuçlar Şekil 6.5'de gösterilmektedir. Bu şekilde, farklı RE örüntüleri (patterns) için, temel tasarım (gerilim düşürülmeyen), ve sıradan gerilim düşürme (572 mV'da iterasyonların devam ettiği, RE uygulanmayan) tasarımı için sonuçlar sunulmaktadır. Bir RE örüntüsü, geçici olarak uygulanan Vrej (Gençleştirici/Rejuvenating gerilimi), Vcrv (hangi kritik bölge voltajında iterasyon yapıldığını), ve NoIofVcrv (belirlenen kritik bölge voltajında kaç kere iterasyon yapıldığını) tanımlamaktadır. Bir örüntü şu şekilde gösterilir; örneğin: 580&572x1 (Vrej=580 mV, Vcrv=572 mV, NoIofVcrv=1). Bundan sonraki sunulan sonuçlar da, bu örüntüler ve tasarımlar üzerinden aktarılmaktadır.

Güç verimliliği artış oranları, belirtilen RE örüntüsüne/patternine uygun olarak gerilim düşürme yapıldığı durumda elde edilen güç verimliliğinin, gerilim düşürme yapılmayan temel tasarımın güç verimliliğine göre farkının yani kazancının, temel güç verimliliğine oranı olarak hesaplanmıştır. Dolayısıyla temel (baseline) tasarımın güç verimliliği kazanç oranı 0 gözükmektedir. Turuncu renkli eğri güç kazanç oranlarını gösterir. Sağdaki dikey eksen güç verimliliği artış oranlarının yüzdeler halinde neye karşılık geldiğini göstermektedir. Bundan sonraki şekiller için de güç verimliliği artış oranı (power efficiency increase rate) bu şekilde hesaplanmış ve bu formatta gösterilecektir. Mavi renkli barlar ise CNN uygulamasının doğruluklarını (accuracy) göstermektedir. Sol dikey eksen doğrulukların yüzde olarak neye karşılık geldiğini göstermektedir.

Yatay eksen ise oluşturulan/tanımlanan RE örüntülerini (patternleri) göstermektedir. İlave olarak; kıyaslama amacıyla, gerilim düşürme yapılmayan, 850 mV nominal besleme yapılan temel (baseline) tasarım, ve "tek bir değer"de CNN iterasyonları yapıldığı (iterasyonun yıkıcı etkisinin görüldüğü ama gençleştirici etkinin olmadığı) sıradan gerilim düşürme tasarımı gösterilmektedir.



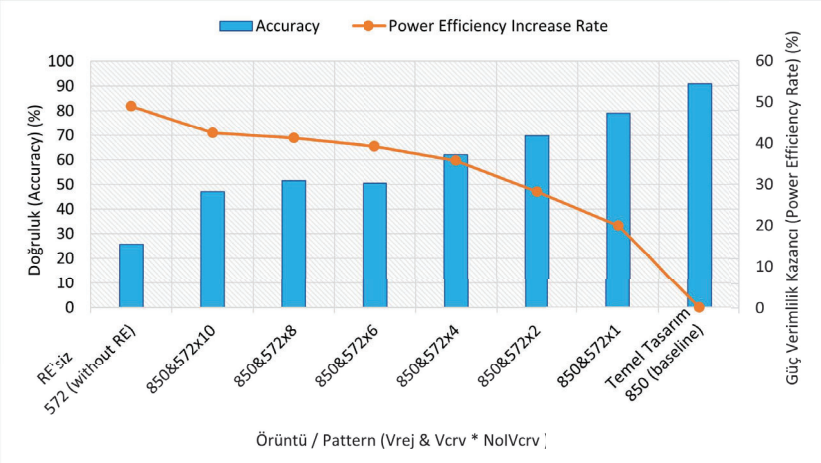
Şekil 6.5 : Farklı Vrej gerilimlerinde Gençleştirici/Rejuvenating Etki.

Şekil 6.5'deki sonuçlara bakıldığında, 850&572x1 örüntüsünde; en yüksek gençleştirici/rejuvenating voltaja (850 yani nominal besleme gerilimi) rağmen ve en sık RE uygulanıyor olmasına rağmen (Vrej ve Vcrv ardışık uygulandığı durum), temel tasarıma göre en az %20'lerde bir güç verimliliği artışı sağlanabilmektedir. Diğer taraftan, yine bu örüntüde; sadece Vcrv uygulandığında doğruluk %25 lere kadar yaklaşık düşerken, doğruluk %80'lerde olmaktadır. Bu gençleştirici/rejuvenating etkinin ne kadar faydalı ve etkili olduğunu göstermektedir. Diğer önemli çıkarım ise, gençleştirici/rejuvenating voltaj düştükçe güç verimliliğini beklendiği üzere arttığı ancak bu durumda doğrulukların azalmaya başladığıdır.

Vrej seviyesinin etkisini gözlemledik, peki NoI'nin etkisi nedir? RE sıklığı ile, RE iyileştirici etkisi arasındaki ilişki nedir? Bu sorulara cevap bulmak Algoritma 1 üzerinde şu değişiklikleri yaparak deneyler gerçekleştirilmiş ve hem güç verimliliği hem de doğruluklar için sonuçlar alınmıştır:

- Vrej: 850 mV olarak, maksimumu ve minimumu da 850 mV, kısaca Vrej'nin 850 mV da sabit olması istenmektedir.
- Vcrv: 572 mV olarak, maksimumu ve minimumu da 572 mV, kısaca Vcrv'nin 572 mV da sabit olması istenmektedir.
- NoI of Vrej: 1 olarak seçilmiştir, ardışık olarak 1 Vrej 1 Vcrv olması için.
- NoI of Vcrv: 1 ile 10 arasında değişecek şekilde belirlenmiştir, böylece sadece RE sıklığı değiştiğinde, RE'nin iyileştirici etkisi değişimi gözlemlenmek istenmiştir.

Bu yöntemle alınan sonuçlar Şekil 6.6'da gösterilmektedir. Sonuçlar incelendiğinde; Vcrv geriliminde koşturulacak CNN iterasyon sayısı arttıkça Vrej kaynaklı



Şekil 6.6 : Farklı sıklıklarla Gençleştirici/Rejuvenating Etki (RE).

gençleştirici/rejuvenating etkinin zayıfladığı görülmektedir, ve doğruluklar azalmaktadır. Ancak her durumda, en seyrek RE örüntüsünde bile tek Vcrv uygulanmasından çok daha fazla doğruluk sağlanmakta, hem de güç tüketimi açısından oldukça az bir fark gözükmektedir. RE sıklığı ("Rejuvenating Effect Frequency"), yani Vrej ile Vcrv'nin ardışıklık oranı arttığında ise doğruluklar artmakta, ancak beklendiği üzere bu sefer de güç verimliliği kazancı düşmektedir.

Özet olarak, bu sonuçlara göre, Vcrv'de koşacak CNN iterasyonu sayısı arttıkça RE etkisi zayıflamakta, doğruluklar azalmakta ama güç verimliliği de artmaktadır. Vrej gerilim seviyesi için de; gerilim arttıkça, RE etkisi artmakta, doğruluklardaki kayıplar azalmakta ama güç verimliliği de azalmaktadır. Dolayısıyla, doğruluklar ile güç verimliliği arasında ödünleşim vardır. Bu yüzden de, sonuçlara gözle bakarak eniyilenmiş çözümü bulmak mümkün değildir. Akla iki çözüm gelmektedir: 1. İki amaçtan biri; örneğin doğruluk, için belirli bir alt limit koyup bunu sağlayan ve diğer amaç için en yüksek değeri oluşturan örüntüyü/pattern'i bulup uygulamak. 2. Bu iki amaca göre en iyiyi/optimumu bulan bir amaç fonksiyonu kullanmak.

6.5 İterasyon Uyarlamalı Gerilim Düşürme Tasarımı, IU

Madem ki, CNN iterasyonu arttıkça doğruluklar azalıyor, bu durumda en kötü iterasyon sayısına göre ve en düşük sıcaklıkta doğrulukların azalmadığı en düşük gerilim belirlenip, Vmin olarak atanırsa ve buna göre gerilim düşürme yapılırsa, bu durumda DIE veya sıcaklıktan etkilenmeyen, hedef doğruluktan ödün vermeyen ve aynı zamanda güç verimlilik artışı sağlayan bir tasarım elde edilmiş olur. İşte bu tasarımımızı iterasyon uyarlamalı gerilim düşürme tasarımı (IU) olarak adlandırıyoruz. IU tasarımı, Şekil 6.3'de sunulan sonuçlara göre (ve hatta farklı denek taşları için sonuçlarımızı da hesaba katarak) Vmin'i belirler ve buna göre gerilim düşürme uygular. Buna göre, 600 mV seviyesinde gerilim düşürme uygular IU tasarımı ve bu sayede doğruluktan bu koşullar altında taviz vermemeyi garanti ederken, aynı zamanda da %43'lük bir güç verimlilik artışı sağlar. Farklı donanım ve uygulama için de; IU kendini uyarlayıp, aynı yöntemle Vmin'i belirleyip buna göre gerilim düşürme yapabilirdir.

6.6 Kısıtlı Gençleştirici Gerilim Düşürme Tasarımı, CRU

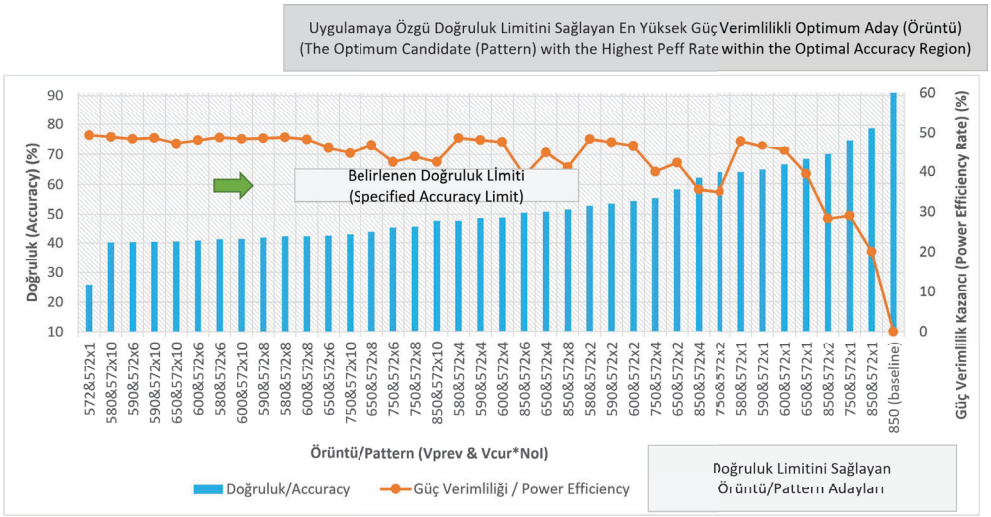
Kısıtlı gençleştirici gerilim düşürme tasarımımız, CRU, belirli bir doğruluk limiti belirlenebilen uygulamlar için, bu doğruluk değerini sağlayan en yüksek güç verimliliğindeki örüntüyü/pattern'i bulur ve uygular. Bu sayede de o doğruluk limiti

korunurken, düşük güç tüketimi de sağlanmış olur. Bazı uygulamalarda doğruluktan ödün verilebilir, bazı uygulamalar ise asla doğruluk kaybını tolere edemez. Eğer bir uygulama için doğruluktan belirli bir seviyeye kadar ödün verilebiliyorsa, bu durumda CRU tasarımının kullanılmasını öneririz.

Amaç; burada tek bir Vmin değeri bulmak değildir, o doğruluk değerini sağlayan güç verimliliği açısından en iyi/optimum sonucu bulmaktır. Peki; Vcrv'deyken iterasyon sayısı mı, yoksa Vrej'in seviyesi mi güç verimliliği açısından daha etkindir? Bu soruyu cevaplamak için (önceki 2 sonuçtan birinde Vcrv'deki iterasyon sayısının RE'ye ve güce etkisine bakılmıştı, diğerinde de Vrej seviyesinin RE'ye ve güce etkisine bakılmıştı. Şimdi ise iki durum için de örüntülerin/patternlerin birarada olduğu sonuçlara bakmalıyız.) Algoritma 1'de şu değişiklikleri yaparak deneyler gerçekleştirilmiş ve hem güç verimliliği hem de doğruluklar için sonuçlar alınmıştır (hem Vrej hem NoIofVcrv serbest bırakılır):

- Vrej: 850 mV olarak, maksimumu 850 ve minimumu da 572 mV olarak ayarlanmıştır.
- Vcrv: 572 mV olarak, maksimumu ve minimumu da 572 mV belirlenmiştir, kısaca Vcrv'nin 572 mV da sabit olması istenmektedir.
- NoI of Vrej: 1 olarak belirlenmiştir
- NoI of Vcrv: 1 ile 10 arasında değişecek şekilde belirlenmiştir, böylece sadece RE sıklığı değiştiğinde RE'nin gençleştirici etkisi değişimi gözlemlenmek istenmektedir.

Sonuçlar Şekil 6.7'de sunulmaktadır, ve bu sonuçlara bakıldığında eğer DIE



Şekil 6.7 : Değişen Vrej ve NoIofVcrv değerlerinde RE etkinliği.

dikkate almazsak ve RE'de kullanmazsak, doğrulukların %20'lere kadar düşebildiği görülmüştür. Eğer RE'yi kullanırsak ise, en zayıf RE örüntüsü/pattern'i için bile doğrulukları %40'tan daha fazla seviyede tutmayı başarmıştır. Ancak bu sonuçlardan, sorumuza cevap bulamadığımızı gözledik, kısacası NoIofVcrv'nin artması veya Vrej'in düşmesi, biri birinden daha çok artırıyor güç verimliliğini diyemeyiz, ve güç verimliliği artarken doğrulukta düştüğü için hangi durum en iyiyi/optimumu sağlar onu da bulamayız. Bu durumda, yukarda da bahsedildiği üzere ya uygulamaya özgü bir limit atamalıyız, herhangi bir amaç için, yada en iyiyi/optimumu bulacak fonksiyon kullanmalıyız.

CRU tasarımına geri dönülecek olunursa; CRU tasarımı, tam da bu noktada devreye girer, ve uygulamaya özgü bir limit varsa onu bulur uygular. Örneğin; doğruluk alt limitimiz %60 olsun, ve IU tasarımı, sonuçlar üzerinden buna göre, %60 limitini sağlayan tüm örüntüleri/pattern'leri ayırır (yeşil bölgedeki örüntüler/pattern'ler) ve bunlar arasından en yüksek güç verimliliğine sahip olanı bulur, 580&572x1. Bulunan bu örüntü/pattern sayesinde de, CRU tasarımı en az %45 oranında güç verimliliği artışı sağlar. Burada, doğruluk limiti düşük seçildiği ve kritik bölge aralığı dar olduğu için tasarımlar arası kazanç oranları benzerdir, ancak kritik bölge aralığı genişledikçe kazanç oranları da farklılaşacaktır.

6.7 En İyi Gençleştirici Gerilim Düşürme Tasarımı, ORU

En İyi (Optimal) Gençleştirici (Rejuvenating) Gerilim Düşürme (Undervolting), ORU, tasarımı; DIE'a karşı, en iyi gençleştirici/rejuvenating örüntüyü/pattern'i bulur, bu sayede diğer gerilim düşürme tasarımlarından daha aşağı seviyelere kadar gerilim düşürme yapabilmeyi amaçlar. Eğer uygulamaya özgü bir limit amaç (doğruluk veya güç verimliliği) belirlenemiyorsa, ve eğer doğruluktan taviz verilebiliyorsa, bu durumda ORU tasarımının kullanılması önerilmektedir. Örüntüler/pattern'ler için şimdiye kadar elde edilen sonuçlara göre, eğer doğruluğu artıran bir pattern varsa bu pattern güç verimliliğini düşürmekte, veya tam tersi yaşanmaktadır ve ödünleşim vardır. İşte, bu sorunu çözmek ve ödünleşim yaparak en iyi (optimal) noktayı bulabilmek için, ORU tasarımı bir amaç (objective) fonksiyonu görevlendirir.

Amaç fonksiyonu, çoklu amacı tek bir amaca dönüştürme işini üstlenir. ORU tasarımı için bu çalışmada; aşağıda aktarılan, 3 farklı amaç fonksiyonu (objective function) metodu ile gerçekleme yapılmıştır; bu amaç/objective fonksiyonları bilinen en yaygın ve en temel fonksiyonlardır [90–92].

- Eşit ağırlıkta toplama fonksiyonu (EWS): 2 amaç, doğruluk ve güç verimliliği, için normalize edilmiş puanları (normalize edildikten sonra bir pattern için

geçerli olan normalize güç verimliliği ve normalize doğruluklar) hiçbirine katsayı uygulamadan toplar ve puanı o örüntü/pattern için kaydeder. Bu şekilde tüm örüntü/pattern'ler için puan tutar, ve en son en yüksek puanı sağlayan örüntüyü/pattern'i en iyi (optimal) kabul eder.

- Eşit ağırlıkta çarpma fonksiyonu (EWP): 2 amaç, doğruluk ve güç verimliliği, için normalize edilmiş puanları hiçbirine katsayı uygulamadan çarparak kaydeder o örüntü/pattern için, toplamak yerine. Bu şekilde tüm örüntüler/pattern'ler için puan tutar, ve en son en yüksek değeri sağlayan örüntüyü/pattern'i en iyi (optimal) kabul eder.

- Doğruluk öncelikli toplama fonksiyonu (ACS): 2 amaç, doğruluk ve güç verimliliği, için normalize edilmiş puanları; bu sefer doğruluk puanını 2 ile çarparak ve güç verimliliği normalize puanını aynen bırakarak, toplar ve kaydeder her örüntü/pattern için. Bu şekilde, tüm örüntüler/pattern'ler için puan tutar, ve en son en yüksek değeri sağlayan örüntüyü/pattern'i en iyi (optimal) kabul eder.

Bu amaç fonksiyonları oldukça temel seviyededir, ve bu çalışmadaki örnekler için yeterlidir, ancak çok daha karmaşık sonuçlar için veya önceliklendirmenin karmaşık şekilde olduğu durumlarda başka amaç fonksiyonları da denenebilir.

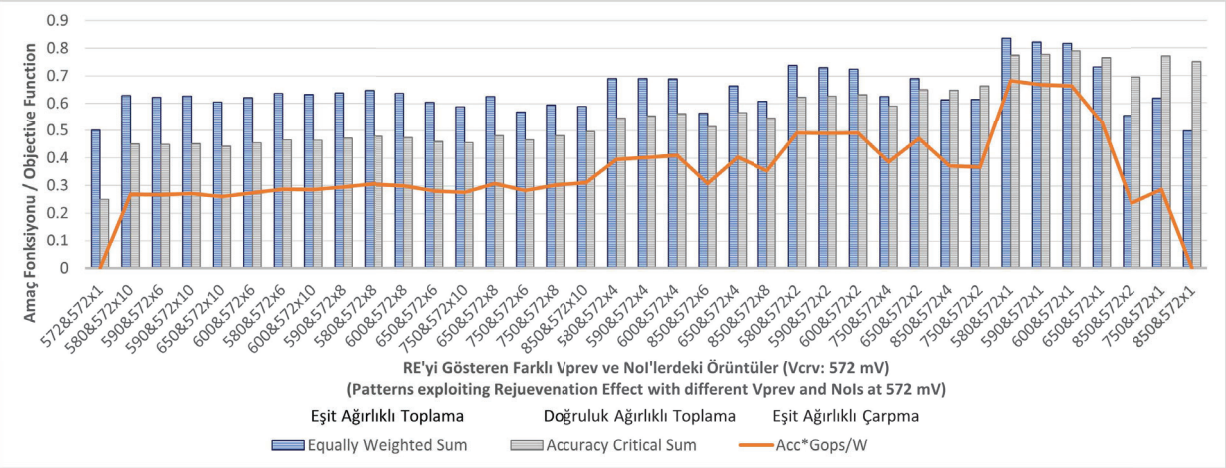
Şekil 6.7'deki sonuçlar için Algoritma 1'de yaptığımız değişikliklerle Algoritma 1'i çalıştırdığımızı ve sonuçları aldığımızı varsayalım. ORU tasarımı önce bu sonuçları alır her pattern için, sonra bu sonuçları normalize eder. Sonra normalize değerler üzerinden amaç/objective fonksiyonları tatbik eder. Bu her örüntü/pattern için amaç (objective) fonksiyonuna göre değişen puan anlamına gelmektedir (Normalde bir amaç fonksiyonu görevlendirilir, bu çalışmada karşılaştırma için 3 farklı amaç fonksiyonu kullanılmıştır). Daha sonra, en yüksek puana göre en iyi/optimal örüntüyü/pattern'i bulur.

Tüm örüntülerin güç ve doğruluk sonuçları normalize edildikten sonra seçilen amaç fonksiyonuna göre her bir örüntü/pattern için bulunan fonksiyon değeri, üç farklı amaç fonsiyon için Şekil 6.8'de sunulmaktadır. Bu sonuçlara göre; ORU tasarımının,

- EWP'ye göre en iyi gerilim düşürme örüntüsü/pattern'i: 580&572x1'dir.

- ACS'ye göre en iyi örüntü/pattern: 600&572x1'dir (doğruluğa ağırlık atadığımız için RE'yi artıran daha yüksek Vrej'li bir örüntü/pattern bulmuştur)

- EWS'ye göre en iyi gerilim düşürme örüntüsü/pattern'i: EWP ile aynıdır.

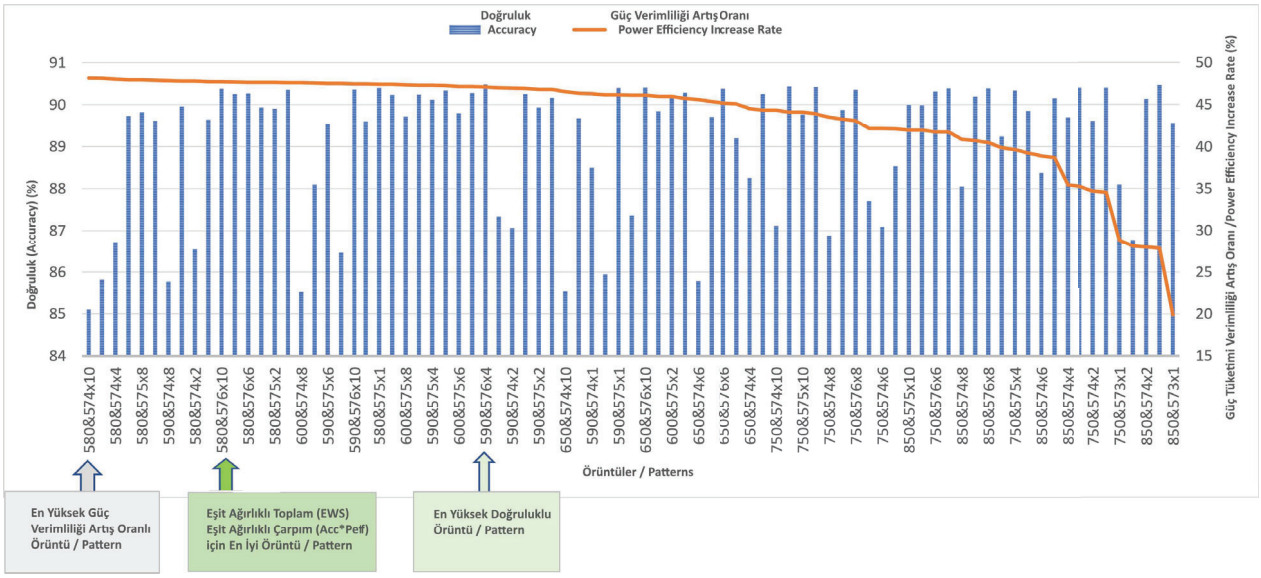


Şekil 6.8 : Farklı örüntüler için farklı amaç fonksiyonları çıktıları.

Şimdiye kadar, hep 572 mV Vcrv'de sonuçlar alınmıştır. Sırayla, RE'yi farklılaştırarak gözlemlemek için, NoIofVcrv ile Vrej belirlenen aralıkta değişecek şekilde serbest bırakılmıştır. Bu sonuçlar kullanılarak da; önce CRU, sonra da ORU için geçerli örüntüler elde edilmiş, ve bunlar üzerinden getiri götürü değerlendirmesi yapılmıştır. Şimdi ise; Vcrv'yi de serbest bırakarak, ORU tasarımının daha kapsamlı değerlendirilmesi hedeflenmektedir. Bu amaçla; Algoritma 1'de şu değişiklikleri yaparak deneyler gerçekleştirilmiş ve hem güç verimliliği hem de doğruluklar için sonuçlar alınmıştır (hem Vrej, hem NoI of Vcrv, ve hem de Vcrv serbest bırakılır):

- Vrej: 850 mV olarak, maksimumu 850 ve minimumu da 560 mV olarak ayarlanmıştır.
- Vcrv: 560 mV olarak, maksimumu 590 ve minimumu da 560 mV olarak belirlenmiştir, kısaca Vcrv'nin bu aralıkta değişmesi istenmektedir.
- NoI of Vrej: 1 olarak belirlenmiştir, NoI of Vcrv: 1 ile 10 arasında değişecek şekilde belirlenmiştir

Sonuçlara geçmeden önce; deneylerimizde, farklı iterasyonlar için aynı voltajın güç tüketim sonuçlarının standart sapmasının ortalama değerden %2 olduğunu gördük. Bu nedenle de, ayrı ayrı sonuçları tutmak yerine ortalama değer kullanılmıştır. Özellikle bu son deneyler çok büyük veri seti oluşturduğu için, bu verinin analizi için bu bilgiden faydalanılmaktadır. Elde edilen sonuçlar Şekil 6.9'da sunulmaktadır. Sonuçların tümü, tek bir şekil içerisinde yansıtılamaz olduğu için, belirli bir limitin üzerinde doğruluk sağlayabilen örüntüler/pattern'ler gösterime konulmuştur. Ancak, amaç fonksiyonları tüm örüntüler/pattern'ler için çalıştırılarak, her biri için sonuçlar (fonksiyon çıktıları) elde edilmiştir. Amaç fonksiyonları tarafından bulunan en iyi



Şekil 6.9 : Farklı amaç fonksiyonları için en iyi örüntüler

(optimal) örüntülerin/pattern'lerden, hali hazırda zaten doğruluk limitinin altında kalan herhangi bir en iyi (optimal) örüntü/pattern olmadığı da görülmüştür.

Şekil 6.9'da yer alan sonuçlar incelendiğinde; ORU tasarımının bu daha kapsamlı örüntü/pattern kümesi içinden bulduğu,

- EWP'ye göre en iyi gerilim düşürme örüntüsü/pattern'i: 580&576x10'dir.
- ACS'ye göre en iyi örüntü/pattern: 580&577x10'dir (doğruluğa ağırlık atadığımız için RE'yi artıran daha yüksek Vrejli bir örüntü bulmuştur).
- EWP'ye göre en iyi gerilim düşürme örüntüsü/pattern'i: 580&577x10'dir.

Özetle; elde edilen örüntüler/pattern'ler ve üç amaç fonksiyonundan herhangi biri ile gerçeklenen ORU tasarımı; hem %85 doğruluk limitini sağlarken, hem de "en iyi gençleştirici gerilim düşürme (optimal rejuvenating undervolting)" sayesinde %47'lik güç verimliliği artışı sağlamaktadır.

6.8 Sonuç ve Değerlendirme

Tezin bu bölümünde anlatılan ve doktora kapsamında gerçekleştirilen bu çalışma, iterasyonun yıkıcı etkisi (DIE) ve gençleştirici etki (RE) fenomenlerini tanıtan literatürdeki ilk çalışmadır. Bu çalışma ile; "bir CNN uygulamasında aynı gerilimde ve sıcaklıkta olmasına rağmen, artan iterasyon sayısı ile doğrulukların azalması", ve "yinelemenin yıkıcı etkisi (DIE) olarak adlandırdığımız bu etkiyi iyileştirebilmek için

bu etkinin görüldüğü voltajda iterasyonlar devam ederken geçici olarak daha yüksek bir gerilimin uygulanmasının "gençleştirici etkisi (RE)" keşifleri gerçekleştirilmiştir. Keşiflerin ardından, gerçek donanımla gerçekleştirilen yoğun deneylerle, bu etkiler karakterize edilmiştir.

Ayrıca, tüm bu karakterizasyon sonuçlarından yola çıkarak; keşfettiğimiz bu iki etkiyi kullanan, FPGA'ler için 3 farklı özgün güvenilir gerilim düşürme tasarımı geliştirilmiştir. Bu tasarımlar için de benzer şekilde kapsamlı deneyler gerçekleştirilerek sonuçlar alınmış ve gerekli ödünleşim çalışmaları yapılmıştır. Geliştirilen tasarımların tümü, tezin amacı olan kendi devre parametrelerini uyarlayabilen donanım tasarımı yaklaşımını yansıtmaktadır.

Önerilen tasarımlardan herhangi biri sayesinde, en az %43'lük bir güç verimliliği artışı sağlanır, ve bu kazanç yanında, doğrulukların hedeflenen veya uygulamaya özgü konulan seviyeden daha aşağı düşmemesi garanti edilir. Son olarak, tüm önerdiğimiz tasarımlar için parametrik ve bir başkası tarafından da gerçeklenebilecek bir metodoloji sunulmaktadır. Böylece, tasarımlar başka donanım, uygulama ve girdi seti için de gerçeklenebilir ve uyarlanabilirdir.

7. SONUÇ VE ÖNERİLER

Tez kapsamında SRAM için 1, DRAM için 4 (gerçekleme yöntemi ile 8), FPGA için 6 (3+3) farklı özgün uyarlamalı donanım tasarımı önerilmiştir. Bu tasarımlar sayesinde, hem güç verimliliği artırılmış hem de başarımdan ödün verilmediği garanti edilmiştir. Ayrıca farklı keşif ve gözlemler sunulmuştur. Devre seviyesinde, mimari seviyede, ve model seviyesinde benzetim ve analizler yapılmıştır, ayrıca gerçek FPGA donanımı ile de deneyler gerçekleştirilmiştir. Tasarımlara yönelik kapsamlı sonuçlar tez içinde aktarılmıştır. Temel tasarımlar için ayrı bölümlerde sonuç ve değerlendirme kısımlarında kazançlar ve ödünleşim anlatılmaktadır.

Uyarlamalı DRAM tasarımları, ADRAM, 4 farklı özgün tasarımı içermektedir, ve her biri en az %21 güç tüketimi kazancı sağlamaktadır, ve bu tasarımlara göre, ve gerçekleme yöntemine (alttaş kutuplama tabanlı, veya gerilim ölçekleme tabanlı) bağlı olarak, ve girdilere bağlı olarak %34 ile %81.8 aralığında toplam yenileme sayısında düşüş sağlamaktadırlar.

Uyarlamalı FPGA tasarımları; toplamda 6 (3 sıcaklık uyarlamalı, 3 iterasyon ve sıcaklık uyarlamalı) özgün gerilim düşürme tasarımından oluşmaktadır. FPGA bazlı CNN hızlandırıcılar için önerilen bu tasarımlar; tasarıma bağlı değişmek üzere, en az %43 oranında güç tüketimi verimlilik artışı sağlarken, doğrulukları da hedeflenen veya uygulamaya özgü seçilen limitte tutmayı garanti ederler.

Uyarlamalı SRAM tasarımları, 1 özgün tasarım içermektedir, ve %74 ihtimalle %35'e varan durağan güç tüketiminde düşüş sağlamaktadır (çalışmada seçilen durum için), ve bunu %1'in altında bir alan maliyetiyle başarmaktadır.

Doktora kapsamında önerilen tasarımların tamamı, tezin amacı olan "kendi kendine devre parametrelerini değiştirebilen ve bu sayede düşük güç tüketimi sağlarken, başarımı koruyabilen uyarlamalı donanım tasarımı" yaklaşımını yansıtmaktadır ve uygulamaktadır. Önerilen tasarımlar, farklı donanım ve uygulamalara ölçeklenebilir ve uyarlanabilirdir. Örneğin; CNN yerine farklı bir uygulama için de; önerilen metodoloji izlenerek önerilen tasarımlar uyarlanabilir. Bunu gösterebilmek amacıyla; iterasyonun yıkıcı etkisinin kök sebebini de gösterecek şekilde, CNN hızlandırıcı olarak çalıştığımız FPGA üzerinde CNN uygulamaları yerine, kendi tanımladığımız temel hesaplamaları koşturmak ve bu hesaplamalar sırasında gerilim düşürmenin sonuçlara etkisini analiz etmek için çalışmalara başlanmıştır. Benzer şekilde, farklı donanımlar için de tasarımların ölçeklenebilmesi mümkündür. Örneğin; FPGA'ler için

önerilen tasarımlar, GPU'lar için de uyarlanabilirdir. Gelecek çalışmalar kapsamında bu tasarım uyarlama alternatiflerinin denenmesi ve değerlendirilmesi planlanmaktadır.

KAYNAKLAR

[1] **Keeth, B., Baker, R., Johnson, B., Lin, F.** DRAM Circuit Design: Fundamental and High-Speed Topics. Wiley-IEEE Press, USA, 2007.

[2] **Weste, N., Harris, D.,** CMOS VLSI Design: A Circuits and Systems Perspective, 4th ed. Addison-Wesley Publishing Company, USA, 2010.

[3] **Rabaey, Jan M., A. C. B. N.,** Digital Integrated Circuits: A Design Perspective. Prentice-Hall, Inc., USA, 2002.

[4] **Moshovos, A., Falsafi, B., Najm, F., Azizi, N.**, A case for asymmetric-cell cache memories. IEEE Transactions on Very Large Scale Integration (VLSI) Systems 13, 7 (2005), 877–881.

[5] **Koc, F., Simsek, O. S., Ergin, O.,** Using content-aware bitcells to reduce static energy dissipation. In 2011 IEEE 29th International Conference on Computer Design (ICCD) (2011), pp. 51–56.

[6] **Borkar, S.,** Design challenges of technology scaling. IEEE Micro, vol. 19, no. 4, pp. 23-29, July-Aug. 1999

[7] **Jedec Solid State Technology Association,** DDR3 SDRAM, JEDEC STANDARD,. Arlington, VA 22201-2107, Nov. 2008.

[8] **Qureshi, M. K., Kim, D.-H., Khan, S., Nair, P. J., Mutlu, O.,** AVATAR: A variable-retention-time (VRT) aware refresh for DRAM systems. In 2015 45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, IEEE.

[9] **Assaderaghi, F., Sinitsky, D., Parke, S. A., Bokor, J., Ko, P. K., Hu, C.** Dynamic threshold-voltage MOSFET (dtmos) for ultra-low voltage VLSI. IEEE Tran. on Electron Devices 44, 3 (Mar. 1997).

[10] **Bonnoit, A., Herbert, S., Marculescu, D., Pileggi, L.,** Integrating dynamic voltage frequency scaling and adaptive body biasing using test-time voltage selection. In Proceedings of the 14th ACM/IEEE international symposium on Low power electronics and design - ISLPED09 (2009).

[11] **Martın, S. M., Flautner, K., Mudge, T., Blaauw, D.,** Combined dynamic voltage scaling and adaptive body biasing for lower power microprocessors under dynamic workloads. In IEEE/ACM International Conference on Computer Aided Design, 2002. ICCAD 2002., IEEE.

[12] **Mıyazakı, M., Ono, G., Ishibashi, K.,** A 1.2-GIPS/w microprocessor using speed-adaptive threshold-voltage CMOS with forward bias. In IEEE Journal of Solid-State Circuits, vol. 37, no. 2, pp. 210-217, Feb. 2002.

[13] **Tschanz, J. W., Kao, J. T., Narendra, S. G., Nair, R., Antoniadis, D. A., Chandrakasan, A. P., De, V.,** Adaptive body bias for reducing impacts of die-to-die and within-die parameter variations on

microprocessor frequency and leakage. In IEEE Journal of Solid-State Circuits, vol. 37, no. 11, pp. 1396-1402, Nov. 2002.

[14] **Koc, F**., Yüksek Lisans Tezi, "Durağan enerji kaybına karşı geliştirilen içerik uyarlamalı bit hücreleri ile özgün sram tasarımı: CSRAM", TOBB ETU. 2013. Ankara, Türkiye.

[15] **Zhang, J., Rangineni, K., Ghodsi, Z., Garg, S.** Thundervolt: Enabling aggressive voltage underscaling and timing error resilience for energy efficient deep learning accelerators. In 2018 55th ACM/ESDA/IEEE Design Automation Conference (DAC) (2018), pp. 1–6.

[16] **Salami, B., Unsal, O., Cristal, A.,** Fault characterization through fpga undervolting. In 2018 28th International Conference on Field Programmable Logic and Applications (FPL) (2018), pp. 85–853.

[17] **Koc, F., Salami, B., Ergin, O., Unsal, O., Kestelman, A. C.**, Can we trust undervolting in fpga-based deep learning designs at harsh conditions? IEEE Micro 42, 3 (2022), 57–65.

[18] **Lecun, Y., Bengio, Y.,** Convolutional Networks for Images, Speech, and Time Series. MIT Press, Cambridge, MA, USA, 1998, p. 255–258.

[19] **Lecun, Y., Bengio, Y., Hinton, G.,** Deep learning. Nature 521, 7553 (May 2015), 436–444.

[20] **Krizhevsky, A., Sutskever, I., Hinton, G. E.,** Imagenet classification with deep convolutional neural networks. Commun. ACM 60, 6 (may 2017), 84–90.

[21] **He, K., Zhang, X., Ren, S., Sun, J.,** Deep residual learning for image recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770-778.

[22] **Szegedy, C., Wei Liu, Yangqing Jia, Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Abinovich, A.,** Going deeper with convolutions. In 2015 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (June 2015), pp. 1–9.

[23] **Bengio, Y.,** Learning deep architectures for ai. Found. Trends Machine Learn. 2, 1 (jan 2009), 1–127.

[24] **Salami, B., Onural, E. B., Yuksel, I. E., Koc, F., Ergin, O., Kestelman, A. C., Unsal, O. S., Sarbazi-Azad, H., Mutlu, O.,** An experimental study of reduced-voltage operation in modern fpgas for neural network acceleration. In IEEE/IFIP DSN (2020), pp. 138–149.

[25] **Ghosh, M., Lee, H.-H. S.** Smart refresh: An enhanced memory controller design for reducing energy in conventional and 3d die-stacked DRAMs. In 40th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO 2007), IEEE.

[26] **Itoh, K., Nakagome, Y., Kimura, S., Watanabe, T.**, Limitations and challenges of multigigabit dram chip design. IEEE Journal of Solid-State Circuits 32, 5 (1997), 624–634.

[27] **Wıcht, B., Larguier, J.-Y., Schmitt-Landsiedel, D.**, A 1.5v 1.7ns 4k /spl times/ 32 sram with a fully-differential auto-power-down current

sense amplifier. In 2003 IEEE International Solid-State Circuits Conference, 2003. Digest of Technical Papers. ISSCC. (2003), pp. 462–508 vol.1.

[28] **Nambu, H., Kanetani, K., Yamasaki, K., Higeta, K., Usami, M., Fujimura, Y., Ando, K., Kusunoki, T., Yamaguchi, K., Homma, N.,** A 1.8-ns access, 550-mhz, 4.5-mb cmos sram. IEEE Journal of Solid-State Circuits 33, 11 (1998), 1650–1658.

[29] **Kim, Y., Seshadri, V., Lee, D., Liu, J., Mutlu, O.,** A case for exploiting subarray-level parallelism (SALP) in DRAM. ACM SIGARCH Computer Architecture News 40, 3 (sep 2012), 368.

[30] **Wu, B., Stine, J. E., Guthaus, M. R.,** Fast and area-efficient sram word-line optimization. In 2019 IEEE International Symposium on Circuits and Systems (ISCAS) (2019), pp. 1–5.

[31] **Hennessy, J. L., Patterson, D. A.,** Computer Architecture, Fifth Edition: A Quantitative Approach, 5th ed. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2011.

[32] **Suzuki, K**., Parasitic capacitance of submicrometer mosfet's. IEEE Transactions on Electron Devices 46, 9 (1999), 1895–1900.

[33] **Chawla, B., Gummel, H.,** Transition region capacitance of diffused p-n junctions. IEEE Transactions on Electron Devices 18, 3 (1971), 178–195.

[34] **Bohr, M.**, Interconnect scaling-the real limiter to high performance ulsi. In Proceedings of International Electron Devices Meeting (1995), pp. 241–244.

[35] **Kang, S.-M. S., Leblebici, Y.,** CMOS Digital Integrated Circuits Analysis amp; Design, 3 ed. McGraw-Hill, Inc., USA, 2002.

[36] **Xilinx**. Zynq ultrascale+ mpsoc zcu102 evaluation kit, https:// www.xilinx.com/products/boards-and-kits/ek-u1-zcu102-g.html 2019.

[37] **Yoo, C.**, A CMOS buffer without short-circuit power consumption. IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing 47, 9 (2000), 935–937

[38] **Shiue, W.-T.,** Leakage power estimation and minimization in vlsi circuits. In ISCAS 2001. The 2001 IEEE International Symposium on Circuits and Systems (Cat. No.01CH37196) (2001), vol. 4, pp. 178–181 vol. 4.

[39] **Rohrer, N., Lichtenau, C., Sandon, P., Kartschoke, P., Cohen, E., Canada, M., Pfluger, T., Ringler, M., Hilgendorf, R., Geissler, S., Zimmerman, J. A,** 64-bit microprocessor in 130-nm and 90-nm technologies with power management features. IEEE Journal of Solid-State Circuits 40, 1 (2005), 19–27.

[40] **Chang, K. K., Yaglıkçı, A. G., Ghose, S., Agrawal, A., Chatterjee, N., Kashyap, A., Lee, D., O'connor, M., Hassan, H., Mutlu, O.,** Understanding reduced-voltage operation in modern dram devices: Experimental
characterization, analysis, and mechanisms. Proc. ACM Meas. Anal. Comput. Syst. 1, 1 (jun 2017).

[41] **Howard, J., Dighe, S., Vangal, S. R., Ruhl, G., Borkar, N., Jain, S., Erraguntla, V., Konow, M., Riepen, M., Gries, M., Droege, G., Lund-Larsen, T., Steibl, S., Borkar, S., De, V. K., Van Der Wijngaart, R.,** A 48-core ia-32 processor in 45 nm cmos using on-die message-passing and dvfs for performance and power scaling. IEEE Journal of Solid-State Circuits 46, 1 (2011), 173–183.

[42] **Ahmed, I., Zhao, S., Meijers, J., Trescases, O., Betz, V.,** Automatic bram testing for robust dynamic voltage scaling for fpgas. In 2018 28th International Conference on Field Programmable Logic and Applications (FPL) (2018), pp. 68–687.

[43] **Hamamoto, T., Sugiura, S., Sawada, S.**, On the retention time distribution of dynamic random access memory (DRAM), in *IEEE Transactions on Electron Devices*, vol. 45, no. 6, pp. 1300-1309, June 1998.

[44] **Kim, Lee, Lee, Nob, Nam, Park, Kim, Kim, Kim, Park, Lee, Lee, Moon, Choi, Park, Lee.** 1Gb X4 X8 X16 DDR3 SDRAM, Micron, 1997.

[45] **Liu, J., Jaiyen, B., Kim, Y., Wilkerson, C., Mutlu, O.,** An experimental study of data retention behavior in modern DRAM devices. ACM SIGARCH Computer Architecture News 41, 3 (jul 2013), 60.

[46] **Lee, D., Kim, Y., Pekhimenko, G., Khan, S., Seshadri, V., Chang, K., Mutlu, O.,** Adaptive-latency DRAM: Optimizing DRAM timing for the common-case. 2015 IEEE 21st International Symposium on High Performance Computer Architecture (HPCA), 2015, pp. 489-501.

[47] **Lee, D., Kim, Y., Seshadri, V., Liu, J., Subramanian, L., Mutlu, O.,** Tiered-latency DRAM: A low latency and low cost DRAM architecture. 2013 IEEE 19th International Symposium on High Performance Computer Architecture (HPCA), 2013, pp. 615-626.

[48] **Liu, J., Jaiyen, B., Veras, R., Mutlu, O.,** Raidr: Retention-aware intelligent dram refresh. In Proceedings of the 39th Annual International Symposium on Computer Architecture (USA, 2012), ISCA '12, IEEE Computer Society, p. 1–12.

[49] **Bhati, I., Chang, M.-T., Chishti, Z., Lu, S.-L., Jacob, B.,** DRAM refresh mechanisms, penalties, and trade-offs. IEEE Transactions on Computers 65, 1 (jan 2016), 108–121.

[50] **Simonyan, K., Zisserman, A.,** Very deep convolutional networks for large-scale image recognition, 2014. arXiv 1409.1556.

[51] **Zhang, X., Zhou, X., Lin, M., Sun, J.,** Shufflenet: An extremely efficient convolutional neural network for mobile devices. In 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (2018), pp. 6848–6856.

[52] **Andri, R., Cavigelli, L., Rossi, D., Benini, L. Yodann**, An ultra-low power convolutional neural network accelerator based on binary weights. In 2016 IEEE Computer Society Annual Symposium on VLSI (ISVLSI) (2016), pp. 236–241.

[53] **Suda, N., Chandra, V., Dasika, G., Mohanty, A., Ma, Y., Vrudhula, S., Seo, J.-S., Cao, Y.,** Throughput-optimized opencl-based fpga accelerator

for large-scale convolutional neural networks. In Proceedings of the 2016 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays (New York, NY, USA, 2016), FPGA '16, Association for Computing Machinery, p. 16–25.

[54] **Boutros, A., Yazdanshenas, S., Betz, V.,** You cannot improve what you do not measure: Fpga vs. asic efficiency gaps for convolutional neural network inference. ACM Trans. Reconfigurable Technol. Syst. 11, 3 (dec 2018).

[55] **Guo, K., Zeng, S., Yu, J., Wang, Y., Yang, H.,** [dl], A survey of fpga-based neural network inference accelerators. ACM Trans. Reconfigurable Technol. Syst. 12, 1 (mar 2019).

[56] **Flautner, K., Kim, N. S., Martin, S., Blaauw, D., Mudge, T.,** Drowsy caches: simple techniques for reducing leakage power. In Proceedings 29th Annual International Symposium on Computer Architecture (2002), pp. 148–157.

[57] **Kuroda, T., Fujita, T., Mita, S., Nagamatsu, T., Yoshioka, S., Suzuki, K., Sano, F., Norishima, M., Murota, M., Kako, M., Kinugawa, M., Kakumu, M., Sakurai, T.,** A 0.9-v, 150-mhz, 10-mw, 4 mm/sup 2/, 2-d discrete cosine transform core processor with variable threshold-voltage (vt) scheme. IEEE Journal of Solid-State Circuits 31, 11 (1996), 1770–1779.

[58] **Hplabs.** Cacti: An integrated cache and memory access time, cycle time, area, leakage, and dynamic power model, https://www.hpl.hp.com/research/cacti/.

[59] **Kim, Y., Yang, W., Mutlu, O.**, Ramulator: A fast and extensible dram simulator. IEEE Computer Architecture Letters 15, 1 (2016), 45–49.

[60] **Meijer, R. M.** Body bias aware digital design : a design strategy for areaand performance-efficient CMOS integrated circuits. Technische Universiteit Eindhoven. Phd Thesis, 2011, Electrical Engineering, Technische Universiteit Eindhoven.

[61] **Chen, T., Naffziger, S.,** Comparison of adaptive body bias (ABB) and adaptive supply voltage (ASV) for improving delay and leakage under the presence of process variation. 888–899. Publisher: Institute of Electrical and Electronics Engineers (IEEE).

[62] **Iwasaki, T., Yamamoto, K., Sakamoto, K., Morita, T., Tomita, R., Morikuni, H., Fujishiro, A., Nakayama, T., Moritoki, M.**, Fabrication technique for ultra low leakage embedded DRAM cell transistor. In 11th International Workshop on Junction Technology (IWJT). 2011, pp. 17-18.

[63] **Cheng C. C., Chin, A**., Low-Leakage-Current DRAM-Like Memory Using a One-Transistor Ferroelectric MOSFET With a Hf-Based Gate Dielectric, in IEEE Electron Device Letters, vol. 35, no. 1, pp. 138-140, Jan. 2014, doi: 10.1109/LED.2013.2290117.

[64] **Schmid, J. R., Parks, H. G., Craigin, R., Schrimpf, R. D.,** Estimating the effect of contamination-induced leakage current in view of DRAM

architectural trends. In Proceedings of 1994 IEEE/SEMI ASMC, IEEE.

[65] **Venkatesan, R. K., Herr, S., Rotenberg, E.,** Retention-aware placement in DRAM (RAPID): Software methods for quasi-non-volatile DRAM. In the Twelfth International Symposium on High-Performance Computer Architecture, 2006., 2006, pp. 155-165.

[66] **Ohsawa, T., Kai, K., Murakami, K.,** Optimizing the DRAM refresh count for merged DRAM/logic LSIs. In Proceedings of the 1998 international symposium on Low power electronics and design. (IEEE Cat. No.98TH8379), 82-87.

[67] **Liu, S., Pattabiraman, K., Moscibroda, T., Zorn, B. G.,** Flikker saving dram refresh-power through critical data partitioning. In Proceedings of the sixteenth international conference on Architectural support for programming languages and operating systems (ASPLOS XVI). Association for Computing Machinery, New York, NY, USA, 213–224.

[68] **Koc, F., Ergin, O.,** ADRAM: Yenileme sıklığı iyileştirilmiş düşük güç tüketimli adaptif dram mimarisi, İslemci Tasarımı Çalıştayı, ITC'2019.

[69] **Hassan, H., Pekhimenko, G., Vijaykumar, N., Seshadri, V., Lee, D., Ergin, O., Mutlu, O.,** ChargeCache: Reducing DRAM latency by exploiting row access locality. In 2016 IEEE International Symposium on High Performance Computer Architecture (HPCA), 2016, pp. 581-593

[70] **Kubo, M., Hori, R., Minato, O., Sato, K.**, A threshold voltage controlling circuit for short channel MOS integrated circuits. In 1976 IEEE International Solid-State Circuits Conference., IEEE.

[71] **Koc, F., Ergin, O.,** Multi-contents aware adaptive memory design. In HIPEAC 16th Int. Summer School on Advanced Computer Architecture and Compilation for High-performance Embedded Systems (online).

[72] **Sze, V., Chen, Y.-H., Yang, T.-J., Emer, J. S.,** Efficient processing of deep neural networks: A tutorial and survey. Proceedings of the IEEE 105, 12 (2017), 2295–2329.

[73] **Schlosser, J., Chow, C. K., Kira, Z.,** Fusing lidar and images for pedestrian detection using convolutional neural networks. In 2016 IEEE International Conference on Robotics and Automation (ICRA) (2016), pp. 2198–2205.

[74] **Whatmough, P. N., Lee, S. K., Lee, H., Rama, S., Brooks, D., Wei, G.-Y.,** 14.3 a 28nm soc with a 1.2ghz 568nj/prediction sparse deep-neural-network engine with 0.1 timing error rate tolerance for iot applications. In IEEE ISSCC (2017), pp. 242–243.

[75] **Chandramoorthy, N., Swaminathan, K., Cochet, M., Paidimarri, A., Eldridge, S., Joshi, R. V., Ziegler, M. M., Buyuktosunoglu, A., Bose, P.**, Resilient low voltage accelerators for high energy efficiency.

In 2019 IEEE International Symposium on High Performance Computer Architecture (HPCA), 2019, pp. 147-158,

[76] **Swaminathan, K., Chandramoorthy, N., Cher, C.-Y., Bertran, R., Buyuktosunoglu, A., Bose, P.,** Bravo: Balanced reliability-aware voltage optimization. In 2017 IEEE International Symposium on High Performance Computer Architecture (HPCA) (2017), pp. 97–108.

[77] **Zou, A., Leng, J., He, X., Zu, Y., Gill, C. D., Janapa Reddi, V., Zhang, X.,** Voltage-stacked gpus: A control theory driven cross-layer solution for practical voltage stacking in gpus. In 2018 51st Annual IEEE/ACM International Symposium on Microarchitecture (MICRO) (2018), pp. 390–402.

[78] **Tajalli, A., Leblebici, Y.,** Design trade-offs in ultra-low-power digital nanoscale cmos. IEEE Transactions on Circuits and Systems I: Regular Papers 58, 9 (2011), 2189–2200.

[79**] Blaauw, D., Chopra, K., Srivastava, A., Scheffer, L.,** Statistical timing analysis: From basic principles to state of the art. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 27, 4 (2008), 589–607.

[80] **Salami, B., Unsal, O. S., Kestelman, A. C.,** Evaluating built-in ecc of fpga on-chip memories for the mitigation of undervolting faults. In 2019 27th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP) (2019), pp. 242–246.

[81] **Xilinx**. Ultrascale architecture memory resources, https://docs.xilinx.com /v/u/en-US/ug573-ultrascale-memory-resources, 2019.

[82] **ASHRAE**., ASHRAE Technical Committee 9.9., Thermal Guidelines for Data Processing Environments, Fourth Edition, 2015.

[83] **Jouppi, N. P., Young, C., Patil, N., Patterson, D., Agrawal, G., Bajwa, R., Bates, S., Bhatia, S., Boden, N., Borchers, A., Boyle, R., Cantin, P.-L., Chao, C., Clark, C., Coriell, J., Daley, M., Dau, M., Dean, J., Gelb, B., Ghaemmaghami, T. V., Gottipati, R., Gulland, W., Hagmann, R., Ho, C. R., Hogberg, D., Hu, J., Hundt, R., Hurt, D., Ibarz, J., Jaffey, A., Jaworski, A., Kaplan, A., Khaitan, H., Killebrew, D., Koch, A., Kumar, N., Lacy, S., Laudon, J., Law, J., Le, D., Leary, C., Liu, Z., Lucke, K., Lundin, A., Mackean, G., Maggiore, A., Mahony, M., Miller, K., Nagarajan, R., Narayanaswami, R., Ni, R., Nix, K., Norrie, T., Omernick, M., Penukonda, N., Phelps, A., Ross, J., Ross, M., Salek, A., Samadiani, E., Severn, C., Sizikov, G., Snelham, M., Souter, J., Steinberg, D., Swing, A., Tan, M., Thorson, G., Tian, B., Toma, H., Tuttle, E., Vasudevan, V., Walter, R., Wang, W., Wilcox, E., Yoon, D. H**., In-datacenter performance analysis of a tensor processing unit. In 2017 ACM/IEEE 44th Annual International Symposium on Computer Architecture (ISCA) (2017), pp. 1–12.

[84] **Reagen, B., Whatmough, P., Adolf, R., Rama, S., Lee, H., Lee, S. K., Hernández-Lobato, J. M., Wei, G.-Y., Brooks, D.**, Minerva: Enabling low-power, highly-accurate deep neural network

accelerators. In 2016 ACM/IEEE 43rd Annual International Symposium on Computer Architecture (ISCA) (2016), pp. 267–278.

[85] **Nurvitadhi, E., Sheffield, D., Sim, J., Mishra, A., Venkatesh, G., Marr, D.,** Accelerating binarized neural networks: Comparison of fpga, cpu, gpu, and asic. In 2016 International Conference on Field-Programmable Technology (FPT) (2016), pp. 77–84

[86] **Papadimitriou, G., Chatzidimitriou, A., Gizopoulos, D.,** Adaptive voltage/frequency scaling and core allocation for balanced energy and performance on multicore cpus. In 2019 IEEE International Symposium on High Performance Computer Architecture (HPCA) (2019), pp. 133–146.

[87] **Pandey, P., Basu, P., Chakraborty, K., Roy, S.,** Greentpu: Predictive design paradigm for improving timing error resilience of a near-threshold tensor processing unit. IEEE Transactions on Very Large Scale Integration (VLSI) Systems 28, 7 (2020), 1557–1566.

[88] **Xilinx**. Zynq dpu ip product guide, https://docs.xilinx.com/v/u/3.1-English/pg338-dpu, 2019.

[89] **Han, S., Liu, X., Mao, H., Pu, J., Pedram, A., Horowitz, M. A., Dally, W. J.,** EIE: efficient inference engine on compressed deep neural network. 2016 ACM/IEEE 43rd Annual International Symposium on Computer Architecture (ISCA), 2016, pp. 243-254.

[90] **Triantaphyllou, E.,** Multi-Criteria Decision Making Methods, vol. 44. Springer New York, NY, 01 2000.

[91] **Kim, I. Y., De Weck, O. L.,** Adaptive weighted sum method for multiobjective optimization: a new method for pareto front generation. Structural and Multidisciplinary Optimization 31, 2 (Feb 2006), 105–116.

[92] **Marler, R. T., Arora, J. S.,** The weighted sum method for multi-objective optimization: new insights. Structural and Multidisciplinary Optimization 41, 6 (Jun 2010), 853–862.