

Hafıza İi Hesaplama (In Memory Computing)

ALİ HADİ ALTUNGÖK

GAZİ ÜNİVERSİTESİ MÜHENDİSLİK FAKÜLTESİ

BİLGİSAAR MÜHENDİSLİĞİ

BİLGİSAYAR MİMARİSİ

Prof.Dr. M.ALİ AKÇAYOL

OCAK 2021

İÇİNDEKİLER

Sayfa

İçindekiler.....	1
Özet.....	2
Hafıza İçi Hesaplama (In Memory Computing).....	3
1. In Memory Data Grid (IMDG- Hafıza içi Veri Izgarası).....	4
2. Hafıza İçi Hesaplama Teknolojisi Faydaları.....	5
2.1 Hafıza İçi Hesaplama de Yüksek Hız ve Ölçeklendirme.....	5
2.2 Hafıza İçi Hesaplama de Gerçek Zamanlı Öngörü.....	5
2.3 Hafıza İçi Hesaplama de Çeşitli Kullanılabilirlik.....	6
2.4 Hafıza İçi Hesaplama ile Teknolojinin Gelişimi.....	6
3. Veri Erişim Kısıtına Alternatif Çözümler ve Hafıza içi Hesaplama Teknolojileri.....	6
3.1. Resistive random access memory (RRAM -Dirençli RAM).....	7
3.2. Ferroelectric Random Access Memory (FeRAM).....	8
3.3. Manyetoresistif RAM (MRAM).....	9
3.4. Dynamic Random Access Memory(DRAM-Dinamik RAM).....	9
3.5. Static Random Access Memory (SRAM- Statik-Durağan RAM).....	9
3.6. Deep in-memory accelerator (DIMA-derin bellek içi hızlandırıcı)..	10
Sonuç.....	12
Kazanımlar.....	12
KAYNAKÇA.....	13

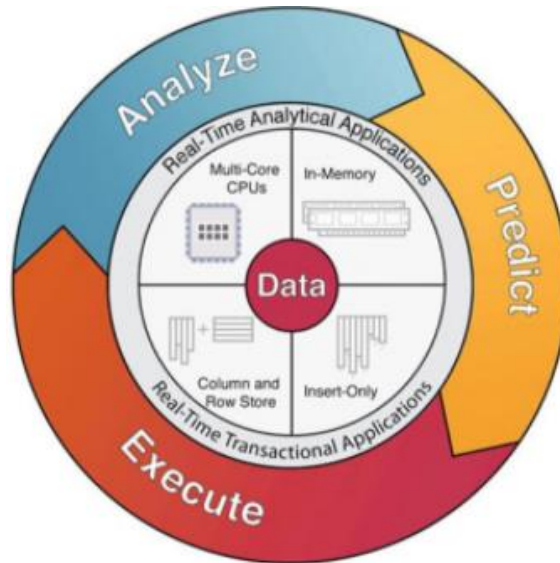
ÖZET

Hafıza içi hesaplama Teknolojisi büyük verilerin daha hızlı bir şekilde işlenmesini sağlamak amacı ile geliştirilmiştir. RAM in yeterli olmadığı durumlar için gerçekleştirilecek olan işlem ve algoritmalara uygun birimler üretilmiştir. Bu araştırma ödevinde bu teknoloji detaylı bir şekilde açıklanmış bu teknolojiye sahip olan cihazlar anlatılmıştır. Hafıza içi hesaplamalardan ve gerçekleştirilmesinden mümkün mertebe elde edilebildiği kadarıyla basedilip ayrıntılarıyla açıklanmıştır. Bu araştırma ödevi Hafıza içi hesaplamaları kapsamakla birlikte bu doğrultuda geliştirilen RRAM, DRAM, SRAM, FeRAM, MRAM ve DIMA yı bunların çalışma mekanizmasını desteklenen algoritmaları ve mimarileri ile ilgili görselleri kapsamaktadır. Aynı zamanda bu teknolojide kullanılan IMDG lerden de bahsedilmiştir. Bu araştırmanın amacı hafıza içi hesaplama (In Memory computing) hakkında detaylı araştırmalar yapılarak geliştirilmiş teknoloji ile birlikte hafızada büyük veriler üzerinde gerçekleştirilen hesaplamaları öğrenmektir. Sonuç bölümünde ise araştırmaların sonucu olarak elde edilmiş bulunan bulgular ele alınıp mümkün bir çıkarımda bulunulmuştur. Kazanımlar bölümünde de araştırma sonucunda öğrenilen bilgiler ve elde edilmiş kazanımlar açıklanmıştır.

Hafıza İçi Hesaplama (In Memory Computing)

İşlemcilerin gelişmesi ile birlikte işlem yapabilme kapasiteleri büyük oranda artmıştır. İşlemcinin veriye ulaşımı ve verinin hafıza birimlerinden işlemciye iletimi, mevcut olan yüksek kapasitenin daha iyi bir seviyede kullanılmasını sınırlandırmıştır. Bunun sonucunda cache bellek (önbellek) teknolojileri, SSD (Solid State Drive) ler ve farklı birçok teknolojilerle üretilmiştir. Gelişen teknoloji, yapay zeka ve büyük veri işlenen uygulamaların artması sonucu veri erişim kısıtına çözüm olmak amacı ile hafıza içi hesaplama (In memory Computing/Computation) teknolojisi gelişmiştir. Bu teknolojiye geliştirilmiş olan ortak hafıza birimleri üzerinde birden fazla işlemcinin çalışması sağlanmıştır. Ve aynı zamanda hafıza birimleri sadece veriyi depolamak için değil bazı algoritmaları destekleyecek şekilde tasarlanmış ve verinin depolanması ile birlikte hafıza biriminde işlenmesi sağlanmıştır.

Büyük ölçekli karmaşık hesaplamalarda işlem hızını arttırmak ve veriye ulaşımın işlem süresini kısıtlamasını engellemek amacı ile geliştirilmiş olunan bu teknolojiye ortak bir RAM de birden fazla bilgisayar yada işlemcinin hızlı bir şekilde işlem yapması sağlanabilmektedir. Normal bilgisayarlarda gerçekleştirilen SSD lere yada harddisklere erişip, oradaki verilerin işlenmesini aksine bu sistemlerde, erişim sonucu ortaya çıkan zaman kaybını önlenerek sadece geliştirilmiş özel RAM ler üzerindeki verilerde çalışılmaktadır ve birden fazla bilgisayar yada işlemci olması durumunda işlem daha küçük görevlere bölünerek her bilgisayara paralel olarak çalışacak şekilde dağıtılır. Aynı zamanda tahin algoritmaları ile hafıza içi veri işleme gerçekleştirilmektedir.

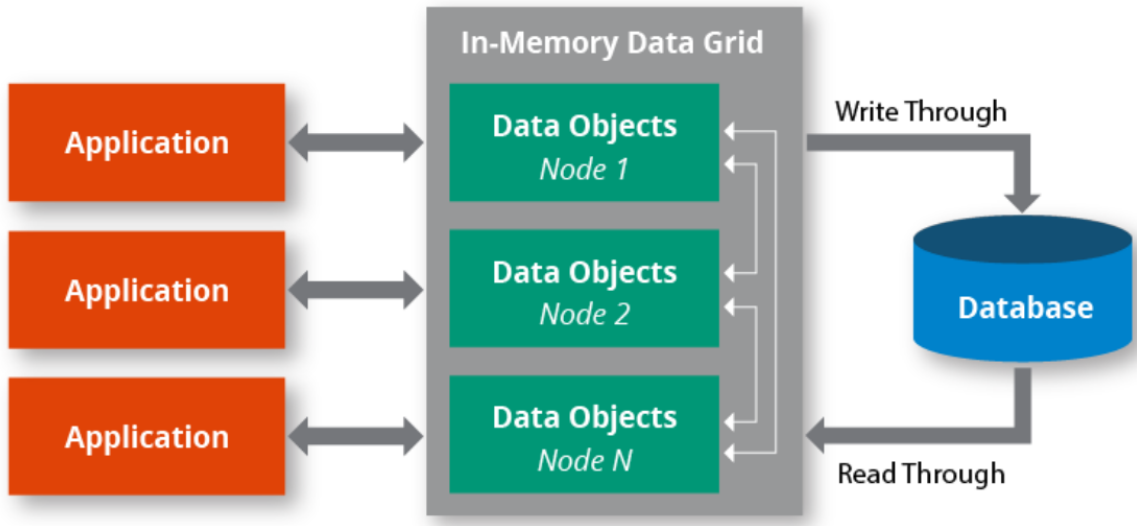


Şekil 1

Bu teknolojide bellek içi işlem (in memory computation) Oracle, DB2, Microsoft SQL server, NoSQL gibi genel veritabanları üzerinden Hazelcast, Infinispan, Oracle Coherence or ScaleOut Software gibi data gridler ile gerçekleştirilebilmektedir.[1]

1. In Memory Data Grid (IMDG- Hafıza içi Veri İzgarası):

In Memory Data Grids (IMDG-hafıza içi veri ızgarası) olarak bilinen teknoloji ise RAM lerini ortak kullanarak verilerini paylaşan aynı ağda kümelenmiş bilgisayarların/işlemcilerin oluşturduğu sistemdir. Yüksek hızlarda veri işlemek için tasarlanmıştır. Mevcut olan tipik tek bilgisayarlı sunucuların aksine daha fazla RAM e ihtiyaç duyan büyük ölçekli uygulamalar için kullanılan sistemleridir. Özellikle büyük veri kümeleri üzerinde kapsamlı paralel işleme yapan uygulamalar için önemlidir. Aşağıdaki şemayı inceleyerek de hafıza içi veri ızgarası (IMDG) diyagramını görmekteyiz.



Şekil 2 IMDG Diyagramı

Bu sistemlerde verilere erişimi kontrol etmek için özel yazılımlar geliştirilmiştir. Her bir bilgisayar hafızada kendine ayrılmış veri yapılarına sahiptir fakat bu veri bütün bilgisayarlar ile paylaşılmaktadır. Veri objeleri genellikle map, list ve queues yapılarındadır. İlkel tipte veri yapıları ise integer ve floating point olarak depolanmaktadır. Bu nesnelerin ve veri türlerinin her biri bir uygulamada değişkenler olarak temsil edilir ve uygulama mantığı bu değişkenlere, uygulamayı çalıştıran aynı bilgisayarda bulunuyorlarmış gibi başvurur. Bu, programlama paradigmasını diğer bellek içi teknolojilerden çok daha basit hale getirir, çünkü geliştiricinin verileri fiziksel olarak almak için kod eklemesine gerek yoktur.

IMDG sistemlerine alternatif olarak IMDB (In Memory Database – Hafıza içi veri tabanı) gibi teknolojiler de mevcuttur. IMDB sistemlerinde farklı olarak verinin daha küçük bir bölümü alınıp işlenip hafızaya geri yazılmaktadır ve genellikle daha çok depolama gereksinimi duyulan uygulamalarda kullanılmaktadır.

In Memory Database teknolojisinin işlevinin her ne kadar da ilk bakışta depolamak için olduğu görülsede burada asıl işlem RAM de yapay zeka uygulamaları gibi büyük hafıza gereksinimi duyulan ve bu büyük verileri gereksinimi olan uygulamaların işlenmesinin ortak bir RAM üzerinde birden fazla işlemci ile hızlıca gerçekleştirilmesidir.

Hafıza içi hesaplama, kaynaklarımıza göre iki ana prensip baz alınarak gerçekleştirilmiştir. Bu prensipler verinin depolanması ve ölçeklendirilmesidir[2]. Ölçekleme, bir sistemin işlem sayısı ve veri sayısının artması durumunda aksaklık yaşanmadan sistemin büyütülebilmesidir. Tam tersi bir durumda ise istek sayısı ve veri sayısında düşüş olduğunda sistemin aksaklık yaşanmadan küçültülebilmesidir. Böylelikle sistemin ağın yada sürecin büyümekte olan datayı idare edebilmesi veya datanın büyümesi durumuna uyum sağlamak için elastik olarak genişleme potansiyeline sahip olması amaçlanmıştır. Bu prensipler sonucu geliştirilmiş olan bu teknolojiye yüksek hız ve ölçeklenebilirlik, gerçek zamanlı öngörüler (Hızlı ve anlık sonuç), çeşitli kullanılabilirlik, ve teknolojinin daha fazla etkinleştirilmesi sağlanmıştır.

2. Hafıza İçi Hesaplama Teknolojisi Faydaları

2.1 Hafıza İçi Hesaplama de Yüksek Hız ve Ölçeklendirme:

Bu sistemlerde bu özelliğin sağlanması verilerin indexlenmesi ortak bir alanda depolanması ile sağlanmakta ve hızlı veri işleyip sorgulama ile gerçekleştirilmektedir. Neticede herhangi bir işlem için en optimum düzeyde performans sağlanmış olunur. Büyük veri için gerçekleştirilmesi gereken veri ölçeklendirilmesi daha öncede anlattığımız gibi ortak bir hafıza üzerinden paralel anlamda işlemcilerin çalışması ile gerçekleştirilmektedir.

2.2 Hafıza İçi Hesaplama de Gerçek Zamanlı Öngörü:

Birçok sistemde bildiğimiz gibi güvenlik amacı ile yada başka amaçlar ile gerçek zamanlı sonuç üretilmesinin sağlanması istenmektedir. Hafıza içi hesaplama ile geçmişte işlenmiş veriler sonucu ile de gerçekleştirilmiş olan yapay zeka uygulaması ile oluşturulmuş

modeller çok daha hızlı karar verebilmekte ve çok daha hızlı sonuç üretebilmektedir. Üstelik bunu gerçekleştirirken her defasında önceden işlemiş olduğu tüm verilere ihtiyaç duyması gibi bir durum yoktur. Birçok işlemicic eşzamanlı olarak çalışıp işlemleri yerine getirmesi ile çok hızlı bir şekilde data analizi yapılabildiği ve tahmin sonucu üretilbildiğinden gerçek zamanlı sonuç üretme sağlanabilmektedir.

2.3 Hafıza İçi Hesaplama de Çeşitli Kullanılabilirlik:

Anlatmış olduğumuz özellikler sonucu bu sistemler birçok alanda kullanılabilmektedirler. Finansal hizmetler, telekomünikasyon, ulaşım gibi alanlarda büyük hacimli verilere sahip olan şirketlerde, bankalarda, taşımacılıkta rota optimizyanunun gerekli olduğu durumlarda, oto pilotlu araçlarda kullanılabilmektedir.

2.4 Hafıza İçi Hesaplama ile Teknolojinin Gelişimi:

Bu teknolojinin entegrasyonu sonucu günümüzde uygulama ve teknolojilerin çoğu gerçekleştirilmiştir. Gerek güvenlik ve sağlık alanında olsun gerek gündelik yaşamda konforun sağlanması amacı ile geliştirilen birçok sistemde bu teknoloji kullanılabilmektedir.

3. Veri Erişim Kısıtına Alternatif Çözümler ve Hafıza İçi Hesaplama Teknolojileri

Von-neumann mimarisini getirdiği darboğaz yani işlemci ile veri arasındaki mesafe ve veri iletim kısıtı sonucu büyük veri işlemede meydana gelen kısıtı kaldırmak amacıyla geliştirilmiş birçok teknoloji mevcuttur. Grafik işleme birimi (GPU-graphics processing unit) birçok işlemcinin kullanımıyla paralel işleme sağlanarak geliştirilen teknolojilerden biridir. Bazı teknolojilerde algoritmalara göre veri akışını hızlandıracak şekilde tasarlanmıştır. Bu teknolojiler arasında tensör işleme birimi (tensor processing unit-TPU) de verilerdeki sinir ağlarının çıkarım aşamasında ana iş yükünü oluşturan multiply accumulate(MAC) işlemlerini hızlandırmak amacı ile geliştirilmiştir.

Araştırma konumuz olan hafıza içi hesaplama teknolojilerini ayrıntılı bir şekilde açacak ve bu teknolojilere örnek verecek olursak: bu teknolojiye dirençli anahtarlama cihazları (resistive switching devices) adı altında static RAM(SRAM) ve dinamik RAM(DRAM)ler mevcuttur. Bu teknolojiler fiziksel olarak cihazların etnegre devrelerinin geliştirilmesi ile

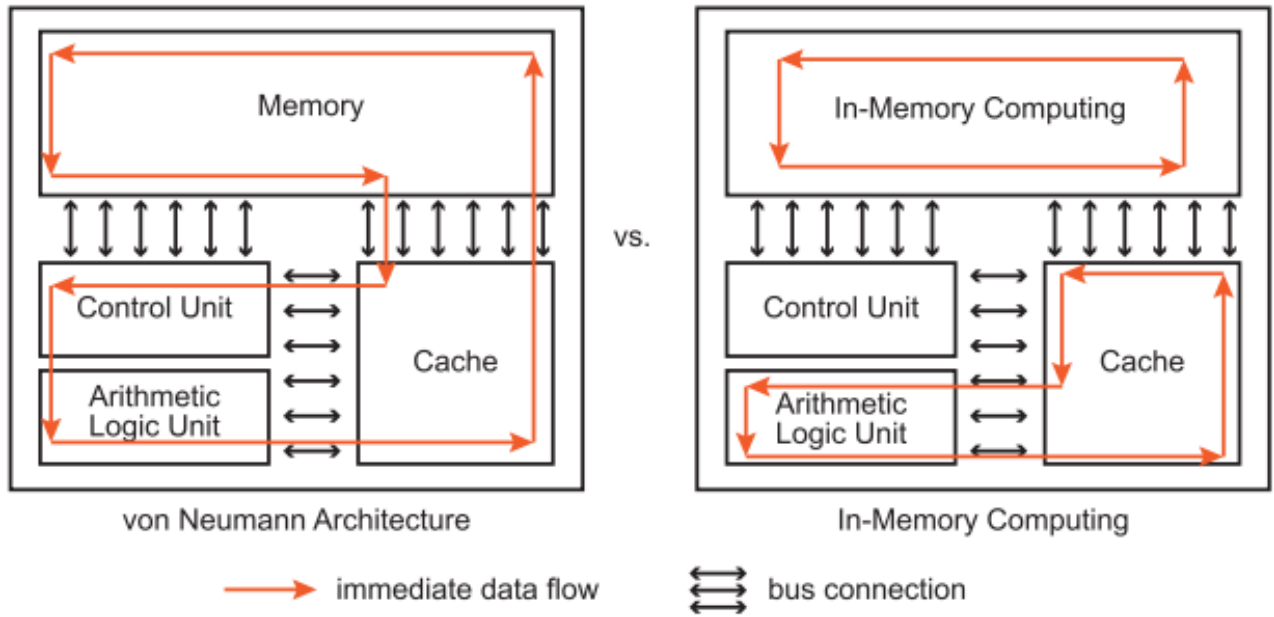
üretmişlerdir. Aynı zamanda dirençli RAM(resistance RAM- RRAM), phase change memory (PCM), magneto-resistive RAM(MRAM) ve ferroelectricRAM(FeRAM) ler de bu cihazlar arasında yer almaktadır. Araştırmaların sonucu başka kaynaklarda DIMA lardan bahsedilmektedir.[3]

Von-neumann dar boğazı kısıtını kaldırmak amacı ile geliştirilen busistemlerde hesaplama işlemi insan beynindeki benzer şekilde gerçekleştirilmeye çalışılmıştır. Hafıza içi teknolojiye bellek duvarındaki gecikme ve enerji yüklerini tamamen ortadan kaldırılmaktadır. Neticede geliştirilmiş olan bu mimarilerde bellekte veri depolanması ile birlikte veri işlemesi gerçekleştirilmektedir. Şimdide kaynaklarımız sonucu bu mimarilerden bahsedip, hafıza içi işlem yapan cihazların fiziksel yapısı ile birlikte algoritma işlemlerini ve devre mimarilerini anlatmaya çalışacağız

3.1 Resistive random access memory (RRAM -Dirençli RAM):

Temel mantık devrelerinin ötesinde vektör matrisi çarpımları gerçekleştirilmektedir. Yapay sinir ağı ve benzeri uygulamalara bağlı matris çarpımı ile ilgili uygulamalara yönelik verimli bir donanımsal çözüm baz alınarak üretilmiştir. Nano ölçekli cihaz olarak mikro mimari katmanlarında paralel işlemler gerçekleştirecek şekilde geliştirilmiştir. Tek bir RRAM cihazı, iki elektrotla sıkıştırılmış bir direnç katmanı içerir. Dirençli katmanlar tipik olarak HfOx, NbOx, TiOx, ve TaOx gibi geçiş metal oksitleridir. RRAM teknolojileri iki kategoriye ayrılabilir: iletken filamanların veya metal iyonlarının kanallarının veya oksijen boşluklarının oluşumuna ve çözünmesine dayanan filamentli RRAM ve heterojen arayüzdeki oksijen boşluklarını RRAM genel direnci değiştirmek için yeniden dağıtan arayüzey.[4]

Programlama sırasında, “SET” işlemi cihaz iletkenliğini artırmakta (veya direnci azaltmakta), “RESET” işlemi ise iletkenliği azaltmaktadır. Normal kullanımdan önce ilk filamentleri oluşturmak için programlamadan çok daha yüksek bir voltaj uygular. Mevcut kararlı direnç durumlarına göre, RRAM iki tipte kategorize edilebilir: analog RRAM, dirençleri en highest resistance state(HRS-yüksek direnç durumu) ile en lowest resistance state(LRS-düşük direnç durumu).[5] RRAM cihazları ile birlikte büyük veri üzerinde işlem yapılması sonucu sistemin genel şemasını aşağıdaki şemayı inceleyerek de anlayabilmekteyiz.



Şekil 3[6]

3.2 Ferroelectric Random Access Memory (FeRAM):

Ferroelektrik rastgele erişim bellekleri (FeRAM'ler), küçük DRAM benzeri hücre boyutunda az voltaj harcayarak hızlı okuma ve yazma gerçekleştirebilmektedir. Özellikle mobil uygulamalar için uygun olduğu söylenmektedir.

Mevcut uygulamalarda $\text{Pb}(\text{Zr},\text{Ti})\text{O}_3$ (PZT) veya $\text{SrBi}_2\text{Ta}_2\text{O}_9$ (SBT) kullanılır. SBT düşük voltajda çalışmasını sağlarken PZT düşük işlem sıcaklığı ve daha yüksek polarizasyon değerleri sağlamaktadır.

Güç kesildiğinde verilerin kaybolmaması için yani uçuculuğun olmaması için dielektrik katman yerine ferroelektrik katman kullanan rastgele erişimli bir bellektir. Flasha göre daha düşük güç kullanımı ve daha hızlı yazma performansı ve yüksek okuma dayanıklılığı gibi avantajları vardır.

	FeRAM ^a	DRAM	Flash	SRAM
Read cycles	10^{12} (10^{15})	10^{15}	10^{15}	10^{15}
Write cycles	10^{12} (10^{15})	10^{15}	10^6	10^{15}
Access time	100ns (20ns)	40-70ns	40-70ns	6-70ns
Write time	100ns (20ns)	ns	μs	Ns
Relative cell size	2x-5x (1x)	1x	1x	> 4x
Data retention	10 years	None	10 years	None

Şekil 4 [7]

3.3 Manyetoresistif RAM (MRAM) :

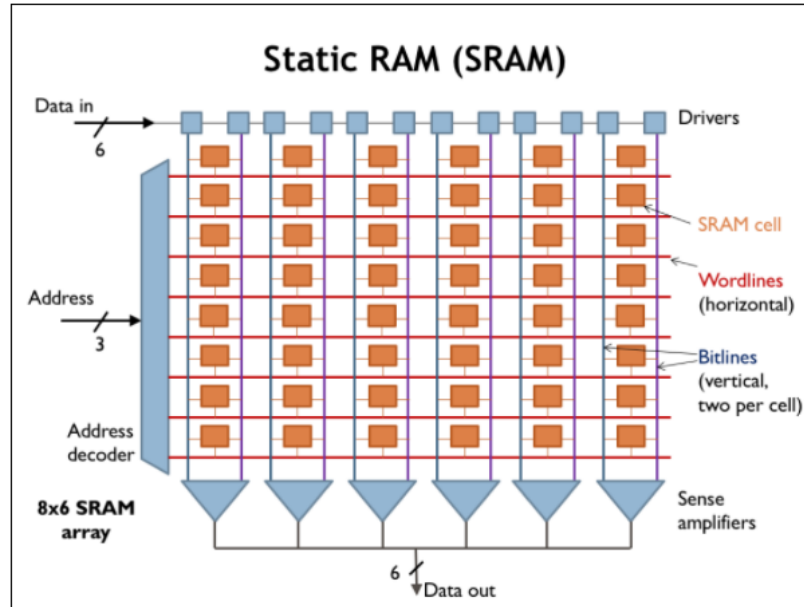
Geçici olmayan, verileri elektrik akımı yerine manyetik birimler ile depolayan rastgele erişimli bellek türüdür. İki ferromanyetik plaka mevcuttur. İki plakadan biri kalıcı bir mıknatıstır. Diğer plakanın mıknatıslanması data depolamak için kullanılmaktadır. Bu durum manyetik tünel bağlantısı olarak adlandırılmaktadır. 1 bit MRAM için en basit yapıdır. Veriler manyetik birimlerde depolandığından uçucu değildir. Güç kesilecek olursa hafıza verileri kaybolmaz. DRAM lerle kıyaslayacak olursak DRAM ler hücre yenileme gereksinimi duyduğundan daha fazla güç tüketirler fakat MRAM ler yenileme gereksinimi duymadıklarından daha az güç tüketirler. Dolayısıyla MRAMler daha düşük güç tüketimi ve sınırsız kullanım ömrü sağlamaktadırlar.

3.4 Dynamic Random Access Memory(DRAM-Dinamik RAM):

DRAM lar da tümleşik devre içinde her bir biti ayrı bir kapasitörde saklanmaktadır. Kapasitörlerin boşalması gereği yenileme devresine ihtiyaç duymaktadırlar. Bu özelliği ile statik RAM lerin zıddı durumdadır. DRAM lar SRAM lara göre yapısal basitliğe sahiptir. Ancak her iki bellekte de güç kaynağı kesildiğinde veriler kaybolmatadır. DRAMlarda Yazma işlemi satırın aktif hale getirilmesiyle ve değerlerin birbirine bağlanarak anlamlı sıraya yazılırken kapasitörleri istenilen değere yüklenmesini sağlayarak yapılır. Belirli bir hücredeki yazma işlemi sırasında bütün satır okutulur. Aynı zamanda VRAM(Video RAM)ler de mevcuttur. VRAMler DRAMların grafik kartlarında kullanılan çift portlu versiyonudur. [8]

3.5 Static Random Access Memory (SRAM- Statik-Durağan RAM):

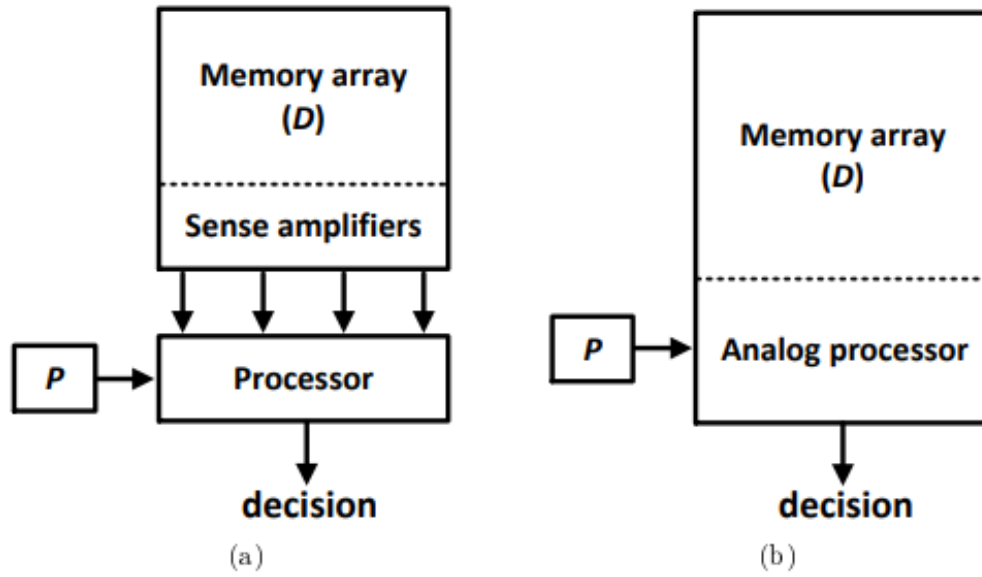
Dinamik RAM'in aksine belleğe güç verildiği sürece içeriği korur ve yenilenme ihtiyacı duymaz. İçindeki her bit iki adet tersleyiciye (inverter) oluşturulan 4lü transistörler üzerine kurulmuştur. SRAM lar 3 ayrı durumda incelenmektedir. Yedek (devre işlemiyorsa), okuma (bilgilerin istendiği), yazma (içeriği güncelliyorken). Bakışımı (simetrik) devre yapısı bir hafıza konumundaki değer DRAM'e göre çok daha hızlı okunmasına olanak tanır. SRAM'in daha hızlı olmasını sağlayan, DRAM'le arasındaki bir başka fark ise tüm adres bitlerini bir kerede alan özel çiplerdir. SRAM ı kontrol etmek DRAM dan daha kolaydır. Ve genellikle diğer modern DRAM türlerinden daha doğru bir rastgele erişime ulaştığı söylenmektedir. SRAM lar genel amaç ürünü olarak kullanılabilirler.[9]



Şekil 5

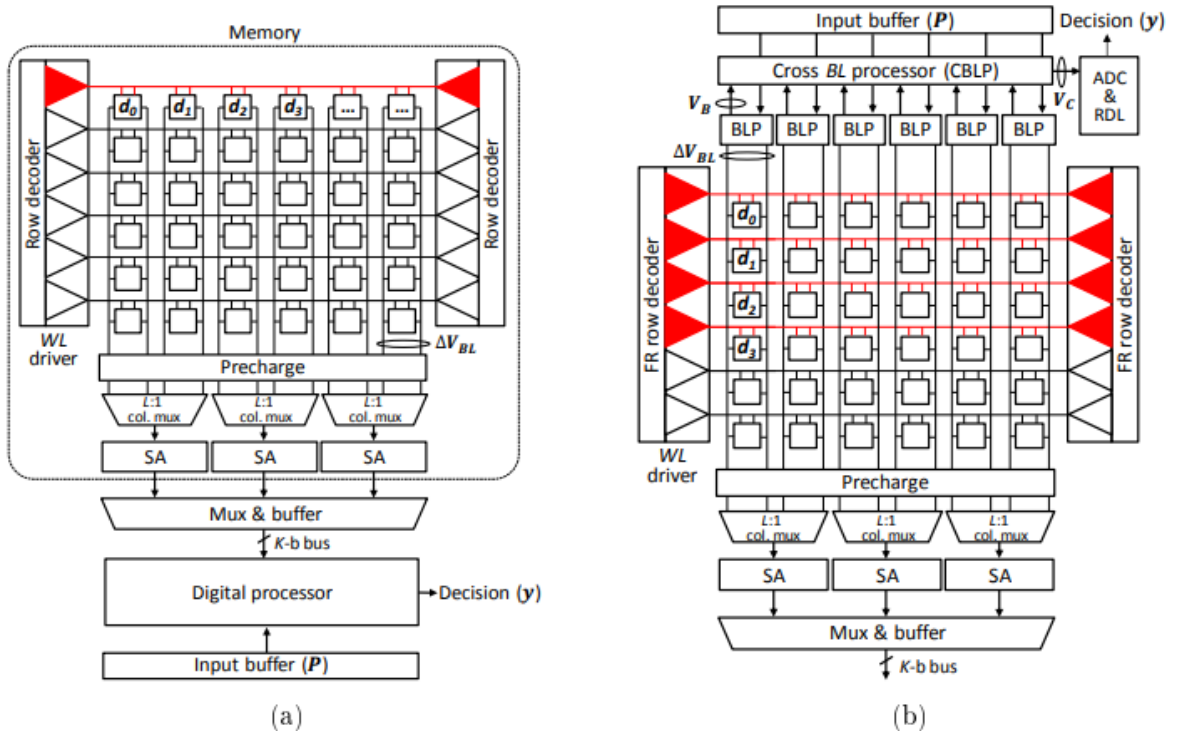
3.6 Deep in-memory accelerator (DIMA-derin bellek içi hızlandırıcı):

DIMA birden çok satıra erişime ve bir seferde işleme amaçlanmıştır. Biyomedikal cihazlarda, otonom araçlarda, robotlar ve nesnelerin interneti gibi sensör açısından zengin platformlarda karar verme yeteneği olan yapay zeka uygulamaları ile kullanılırlar. DIMA iki temel ilkeyi kullanır: ilki birden çok satıra erişim ve tek seferde bellek dizisi işleme ikincisi ise aralık eşlemeli düşük hızlı bitcell dizisinin çevresinde analog işlem gerçekleştirilmedir.[] Kaynağımızda anlatılmış olan DIMA support vector machine (SVM), template matching (TM), k-nearest neighbor (k-NN), and matched filter yapay zeka algoritmaları ile birlikte convolutional neural network (CNN), sparse distributed memory (SDM), and random forest (RF) algoritmalarını destekleyecek şekilde açıklanmıştır. Aşağıdaki şemayı inceleyerek de DIMA mimarisi ile normal bir mimarinin farkını görebilmekteyiz.



Şekil 6 a) Normal mimari b) DIMA mimarisi [10]

Aşağıdaki şekilde de şekli(a) da geleneksel mimariye sahip bir sistemi depolamayı ve kırmızı işaret ile gösterilern driver ları, şekil(b) de ise DIMA mimarisine sahip olan sistemi görmekteyiz.



Şekil 7 a) Normal mimari b) DIMA mimarisi [11]

Sonuç:

Bu araştırma ödevinin sonucu olarak Hafıza içi hesalama (In Memory Computing) teknolojisi büyük verilerin işlenmesine ihtiyaç duyulması, bu doğrultuda uygulamalar geliştirilmesi sonucu ortaya çıkmıştır. Bu teknoloji işlemci ve veri erişimi arasındaki kısını azaltmak amacı ile geliştirilmiştir. Birçok işlemcinin bir arada paralel olarak veri işleme gerçekleştirilmesi ile birlikte bu verilerin depolanması için özel hafıza birimleri geliştirilmiştir. Bu hafıza birimleri özel algoritmaları destekleyecek şekilde tasarlanmışlardır. Bu biremlerde veriler sadece depolanmakla kalmayıp yapay zeka algoritmaları ile veriler işlenmekte, tahmin (predict) tahmin işlemi gerçekleştirilmektedir. Dolayısıyla Gerçek zamanlı sonuç üretilmektedir. Ve bu doğrultuda ihtiyaç duyulan uygulamalardaki işleme ve hafıza sorununa çözüm üretilmiştir. Bu teknoloji için RRAM, DRAM, SRAM, FeRAM, MRAM ve DIMA gibi teknolojiler geliştirilmiştir. Bu cihazlar ve hafıza içi hesaplamaları araştırma ödevinde mümkün mertebe detaylı olarak açıklanmaya çalışılmıştır. DIMA ların k-NN(en yakın komşuluk), random forest(RF), support vector machine(SVM), template matching (TM), convolutional neural network(CNN), sparse distributed memory (SDM) gibi birçok algoritmayı desteklediği ve diğerlerine göre daha güncel olması sonucu diğer hafıza içi hesaplama gerçekleştirilen cihazlardan daha iyi olduğunu, daha yüksek verime sahip olduğunu söyleyebiliriz.

Kazanımlar:

Bu araştırma ödevinde hafıza içi hesaplama teknolojisi ayrıntılı bir şekilde araştırılıp öğrenilmiştir. Bu teknolojinin gerçekleştirilmesi sonucunda ortaya çıkan faydalı uygulamalar ve bu uygulamaların özellikleri ve kullanım alanları öğrenilmiştir. Bu teknolojiye sahip olan cihazlar detaylı bir şekilde araştırılıp fizyolojik mekanizması ile birlikte destekledikleri algoritmalar da incelenmiştir. Bu teknolojiye kullanılan IMDG (In Memory Data Grid - hafıza içi veri ızgarası) teknolojisi ve mimarisi açıklanmıştır. RRAM, DRAM, SRAM, FeRAM, MRAM ve DIMA cihazları anlatılarak işleyişleri açıklanmış ve gerekli görsellerle birlik de daha açıklayıcı hale getirilmiştir. DIMA cihazının diğer cihazlardan dahi verimli olduğu ve hatta yapay zeka algoritmaları arasından kullanışlı olan ve bilinen algoritmaları desteklediği, örnek verecek olursak çok fazla veri olması durumunda k-NN algoritmasının kullanımında çok fazla işlem gerektiği fakat bu teknoloji ile bu işlemlerin birçok işlemcinin aynı anda çalışması ile birlikte daha elverişli bir şekilde gerçekleştirilebildiği öğrenilmiştir.

KAYNAKÇA :

- [1] Wikipedia contributors. In-memory processing. Wikipedia, The Free Encyclopedia. September 8, 2020, 12:05 UTC. Available at: https://en.wikipedia.org/w/index.php?title=In-memory_processing&oldid=977366896. Accessed January 22, 2021.
- [2] Kren Krivaa. November 24, 2019. Available at: <https://www.gigaspace.com/blog/in-memory-computing/#why> Accessed January 22, 2021.
- [3] Ielmini, Daniele, and H-S. Philip Wong. "In-memory computing with resistive switching devices." *Nature Electronics* 1.6 (2018): 333-343. s.333
- [4] Yan, Bonan, et al. "Resistive Memory-Based In-Memory Computing: From Device and Large-Scale Integration System Perspectives." *Advanced Intelligent Systems* 1.7 (2019): 1900068. S.2
- [5] Yan, Bonan, et al. "Resistive Memory-Based In-Memory Computing: From Device and Large-Scale Integration System Perspectives." *Advanced Intelligent Systems* 1.7 (2019): 1900068. S.3
- [6] Yan, Bonan, et al. "Resistive Memory-Based In-Memory Computing: From Device and Large-Scale Integration System Perspectives." *Advanced Intelligent Systems* 1.7 (2019): 1900068. S.3 Figure. 1a
- [7] Nagel, N., Mikolajick, T., Kasko, I., Hartner, W., Moert, M., Pinnow, C., . . . Mazure, C. (2000). An Overview of FeRAM Technology for High Density Applications. *MRS Proceedings*, 655, CC1.1.1. doi:10.1557/PROC-655-CC1.1.1 s.1 Table I
- [8] Wikipedia katılımcıları (2020). DRAM (bilgisayar). *Vikipedi, Özgür Ansiklopedi*. Erişim tarihi 18.35, Ocak 22, 2021 [url://tr.wikipedia.org/w/index.php?title=DRAM_\(bilgisayar\)&oldid=24190011](url://tr.wikipedia.org/w/index.php?title=DRAM_(bilgisayar)&oldid=24190011).
- [9] Wikipedia katılımcıları (2020). SRAM. *Vikipedi, Özgür Ansiklopedi*. Erişim tarihi 18.45, Ocak 22, 2021 <url://tr.wikipedia.org/w/index.php?title=SRAM&oldid=23201908>.
- [10] MINGU KANG. DEEP IN-MEMORY COMPUTING (2017).s4
- [11] MINGU KANG. DEEP IN-MEMORY COMPUTING (2017).s10

Ek Kaynaklar:

<https://www.sciencedirect.com/science/article/pii/S2405896318334098>

<https://www.gigaspace.com/blog/in-memory-computing/#why>

<https://www.eejournal.com/article/in-memory-computing/>

<https://hazelcast.com/glossary/in-memory-computation/>

<https://hazelcast.com/glossary/in-memory-data-grid/>

<https://hazelcast.com/glossary/in-memory-processing/>

<https://dergipark.org.tr/en/download/article-file/236126>