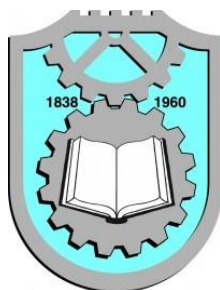


**Факултет инжењерских наука Универзитета у
Крагујевцу**



Назив студијског програма: Машинско инжењерство

Ниво студија: Основне академске студије

Модул: Информатика у инжењерству

Предмет: Архитектура рачунарских система

Број индекса: 102/2014

Ерић Р. Филип

**Динамичка меморија са случајним приступом
завршни рад**

Комисија за преглед и одбрану:

1. Проф. Др Радуловић Јасна

2. _____

3. _____

Датум

одбране: _____

Оцена: _____

У оквиру овог завршног рада кандидат треба да, помоћу предложене и шире литературе, стекне пре свега, основно, а затим и дубље знање о динамичкој рам меморији, принципу рада, карактеристикама, као што су: операција читања и писања, детекција и корекција грешака, архитектуре и структуре система за детекцију и корекцију грешака, пројектовање контролера динамичке рам меморије и типовима динамичке рам меморије. Треба обратити пажњу на тржиште, како се оно мењало током времена и на неке од највећих произвођача динамичке рам меморије

Препоручена литература:

- [1] William Stallings, Организација и архитектура рачунара: Пројекат у функцији перформанси: Превод седмог издања
- [2] Andrew S. Tanenbaum: Архитектура и организација рачунара: Превод петог издања
- [3] Миле К. Стојчев: RISC, CISC i DSP procesori: Електронски факултет у Нишу 1997

Крагујевац,
2018.

Ментор
Проф. Др Јасна Радуловић

САРДЖАЈ

1.0 Уводна разматрања	1
2.0 Основне карактеристике и подела меморије	3
2.1 Унутрашња меморија	3
2.2 Полупроводничка меморија	4
2.3 Хијерархија меморије	5
2.4 Основне карактеристике SRAM меморије	6
3.0 Принцип рада DRAM-а	8
3.1 Операције читања и уписивања и времески дијаграми код DRAM	11
3.2 Освежавање	12
3.3 Кошење сигнала и пропагационо кашњење	13
3.4 Осциловање	14
3.5 Шумови	15
4.0 Разлика између SRAM-а и DRAM-а	16
5.0 Детекција и корекција грешака код DRAM-а	18
5.1 Заштита од грешака	20
5.2 Парност	20
5.3 Детекција и корекција грешака	22
6.0 Архитектуре система за детекцију и корекцију грешака	23
6.1 Придružена EDAC	23
6.2 Активна EDAC	24
6.3 Имплементација EDAC јединице	25
6.4 Структура EDAC	28
7.0 Пројектовање DRAM контролера	29
7.1 Приступ код DRAM-а	32
7.2 Освежавање DRAM-а	33
8.0 Типови DRAM меморије	35
8.1 Синхрона DRAM меморија	35
8.2 Рамбус DRAM меморија	36
8.3 Остали типови DRAM меморија	37
9.0 Тржиште DRAM меморије	38
9.1 Компанија Самсунг на тржишту DRAM-а	40
Закључак	43
Литература	44

Резиме

У овом раду детаљно је описана динамичка меморија са случајним приступом, за шта и како се користи, намене у савременој техници као и све њене мане и предности. Представљен је детаљан принцип рада и карактеристике меморије као што су: операције читања и уписивања са временским дијаграмима, кошење сигнала, шумови, осцилације, детекција и корекција грешака као и архитектура система за детекцију и корекцију грешака. Поред ових обрађених тема, анализирао сам и тржиште динамичке меморије са случајним приступом, указао на кључне факторе који обликују тржиште и на неке од главних произвођача динамичке меморије са случајним приступом.

Кључне речи: динамичка меморија, читање, уписивање, грешке, тржиште

Abstract

In this project, dynamic ram memory is described, it's use, it's purposes in modern technic, and all it's flaws and advantages. The way it operates and all it's characteristics such as: read and write operations with time diagrams, cutting of signals, noises, oscillations, detection and correction of errors and the architecture of the system for detection and correction of errors are described in detail. Along these subjects I have analyzed the dynamic ram memory market, pointed to key factors that make the market what it represents today and also pointed on some of the biggest manufacturers of dynamic ram memory today.

Key words: dynamic memory, read, write, errors, market

1.0 Уводна разматрања

Дигитални рачунар представља савремену електронску машину, која треба да изврши било који задатак који јој се зада. Рачунар је комплексан систем, са хијерархијском организацијом, састављен од великог броја електронских компонената. Хијерархијски систем представља скуп међусобно повезаних подсистема, при чему је сваки од подсистема поново хијерархијски структурно организован, све док се не дође до елементарног подсистема. Два основна појма која се везују за рачунарски систем су: *хардвер* и *софтвер*. Хардвер се састоји од електронских компонената и електромеханичких уређаја, укључујући и системе за напајање и расхлађивање. Како технологија напредује, новији рачунари садрже све већи број електронских компоненти, па су самим тим и комплекснији. Софтвер рачунара се састоји од инструкција и података које рачунар користи како би обављао различите задатке. Секвенца инструкција којом се описује како треба обавити задатак се назива програм. Подаци којима се манипулише у програму чине базу података. Рачунарски систем је направљен од свог хардвера и системског софтвера који омогућава његово коришћење. Системски софтвер је, у суштини, скуп програма који омогућава ефикасније коришћење рачунара. Програми који су садржани у овом скупу се називају оперативни систем. Они се разликују од апликативних програма које пишу програмери у циљу решавања посебних проблема.

Меморија представља део рачунарског система који служи за складиштење програма и података. Информације које се налазе у меморији не трпе никакву промену у току времена, па самим тим можемо рећи да је меморија пасивни део система. Меморијски систем рачунара чине јединице за памћење информација и алгоритама који су намењени за управљање датом информацијом. Како би се омогућио константан рад процесора, пожељно је да он може непосредно приступити меморији и да се тај приступ не прекида. На овај начин омогућава се пренос информација између процесора и меморије брзином којом процесор ради. Меморије чија је брзина рада једнака брзини рада процесора су изузетно скупе и нису практичне. Због тога се у систему информација распоређује по различитим меморијским јединицама које имају различите физичке карактеристике.

Идеја за ДРАМ технологију појавила се релативно рано у временском оквиру полупроводника интегрисаних кола. Ранији облик је пронађен у калкулатору Тошиба који је направљен 1966. године из дискретне компоненте, а потом две године касније развила се идеја о DRAM-у каквог га данас знамо.

Током времена технологија која је примењивана за израду меморијских модула се мењала константно, при чему се неки део и данас користи. Код првих рачунара логички битови су се памтили углавном на механичким уређајима као што су биле картице и папирне траке. Касније долази до промене па се користе фероелектрични материјали у облику језгра и преносивих магнетних медијума. Како је технологија напредовала почиње све већа учесталост у примени полупроводничких компоненти и оптичких дискова.

Као што је наведено у претходном делу, низ прстенасто обликованих феромагнетних петљи био је најчешћа форма складишта са случајним приступом за главну меморију код старих рачунара. Ове петље су често биле називане језгрима (на енглеском *cores*), па се на главну петљу често односио назив језгро. Развојем технологије, тачније микроелектронике дошло је до истискивања магнетних језгара као облика складиштења меморије. Данас је употреба полупроводничких чипова за главну меморију постала скоро универзална.

Динамичка меморија са случајним приступом представља један од кључних компоненти у рачунарском систему, а данас са развојем технологије и у многим другим системима. Развој ове меморије је веома битан јер њена побољшања имају утицај на перформансе система у којима се користи.

2.0 Основне карактеристике и подела меморије

Како би боље разумели како функционише динамичка RAM меморија, неопходно је рећи нешто везано и за меморију уопште, поделу меморије и намену. Када причамо о рачунарима, меморија може бити спољашња или унутрашња. Разлике између ове две су у томе што се спољашња меморија користи за стално чување података, док унутрашња меморија служи за привремено складиштење података, којима процесор може да приступи директно.

Ово је основна подела меморије. Међутим, поред ове постоје и друге поделе по другим критеријумима, а то су:[1]

- 1) Према физичком начину записивања података (електронски, магнетни, оптички)
- 2) Према методи приступа (непосредни приступ, директан приступ, секвенцијални приступ, асоцијативне меморије)
- 3) Према начину организације (адресне, асоцијативне и стек меморије)
- 4) Према трајности података по нестанку напајања (постојане и непостојане)[11]

Карактеристике меморија су:

- 1) Број битова који се могу запамтити у меморији
- 2) Време приступа
- 3) Брзина преноса података – број битова у јединици времена
- 4) Цена – однос цене меморије и перформанси[11]

2.1 Унутрашња меморија

Унутрашњу меморију можемо поделити у:

- 1) RAM (Random Access Memory)
- 2) ROM (Read Only Memory)
- 3) Cache memory (Кеш меморија)
- 4) Buffers (Бафери)[1]

RAM меморија, или полупроводничка главна меморија, је меморија која садржи програм чије је извршавање у току, а садржи и податке који су потребни за извршавање истог. Сваком биту се може приступити независно од претходне меморијске локације. На одређену меморијску локацију се могу уписивати или читати подаци. [1]

Битно је напоменути да се подаци у овој меморији чувају докле год постоји напајање. Када напајање нестане подаци се губе и меморија је од тог тренутка празна.

ROM меморија како име каже, је меморија у којој је могуће само читање. За разлику од претходне меморије, када се изгуби напајање, подаци и даље остају у меморији тј. не губе се. Подаци који се налазе у овој меморији су у главном подаци који су неопходни за рад самог рачунара. [1]

Кеш меморија представља меморију која се налази у процесору. Главна улога ове меморије је да превазиђе разлику између брзине процесора и брзине главне меморије. Како је процесор најбржа компонента у систему, самим тим мора да сачека меморију у процесу достављања података. Овде кеш меморија помаже како би процесор мање чекао на жељене податке.

Бафери служе да ако процесор не може довољно брзо да обрати достављене податке, чува податке који треба да дођу на ред. [1]

2.2 Полупроводничка меморија

Типови меморије наведени у претходном делу спадају у **полупроводничке меморије**.

Основни елемент полупроводничке меморије је меморијска ћелија. Све меморијске ћелије имају неколико заједничких особина које деле а то су:

1) имају два стабилна (или полустабилна) стања, која се употребљавају за представљање бинарне цифре 1 или 0

2) све ћелије су способне да се у њих уписује (бар једном) како би успоставиле стање.

3) из ћелија је могуће читање како би се сазнало њихово стање[1]

Меморијска ћелија има најчешће три функционална терминала који су у стању да носе електрични сигнал. Терминалом за избор, се бира ћелија за читање или уписивање. Да ли је операција читања или уписивања, показује нам управљачки терминал. За уписивање, други терминал обезбеђује електрични сигнал којим се стање

ћелије поставља на 0 или 1. Када је у питању операција читања, тај исти терминал се користи за излаз стања ћелија. Постоји више начина за унутрашњу организацију, функционисање, меморијске ћелије, и они зависе од употребљене технологије интегрисаних кола.

2.3 Хијерархија меморије

Како би се омогућило што ефикасније складиштење, великог броја података, користи се хијерархијско организовање меморије. Први ниво чине регистри процесора којима процесор може приступити пуном брзином. Други ниво чини кеш меморија која може бити величине неколико мегабајта код процесора најновије генерације. Следећа је главна меморија која може бити величина неколико гигабајта. Испод се налазе хард диск, а након тога, магнетне траке које су се некада користиле и оптички дискови за трајно складиштење података.

Ако погледамо овакву хијерархијску поделу можемо уочити три кључна параметра која карактеришу сваки ниво.

Први параметар је време приступа меморији. Ако кренемо од врха, примећује се да се регистрима процесора може приступити за неколико наносекунди, а кеш меморији за нешто више времена. Приступање главној меморији ће трајати неколико десетина наносекунди, што је и даље веома брзо, али у поређењу са регистрима и кешом, уочава се спорији рад. Након главне меморије, време приступа се знатно повећава. Време приступања диску је око десет милисекунди, а приступ тракама и дисковима се мери у секундама.

Други параметар јесте капацитет складиштења. Поново регистри процесора могу чувати најмање, око 128 бајтова. Кеш меморија има нешто већи капацитет и може чувати и до неколико мегабајта, главна меморија може и до неколико гигабајта. Дискови могу чувати и по неколико терабајта. Траке и оптички дискови се могу чувати посебно, па самим тим њихов капацитет зависи од новца.

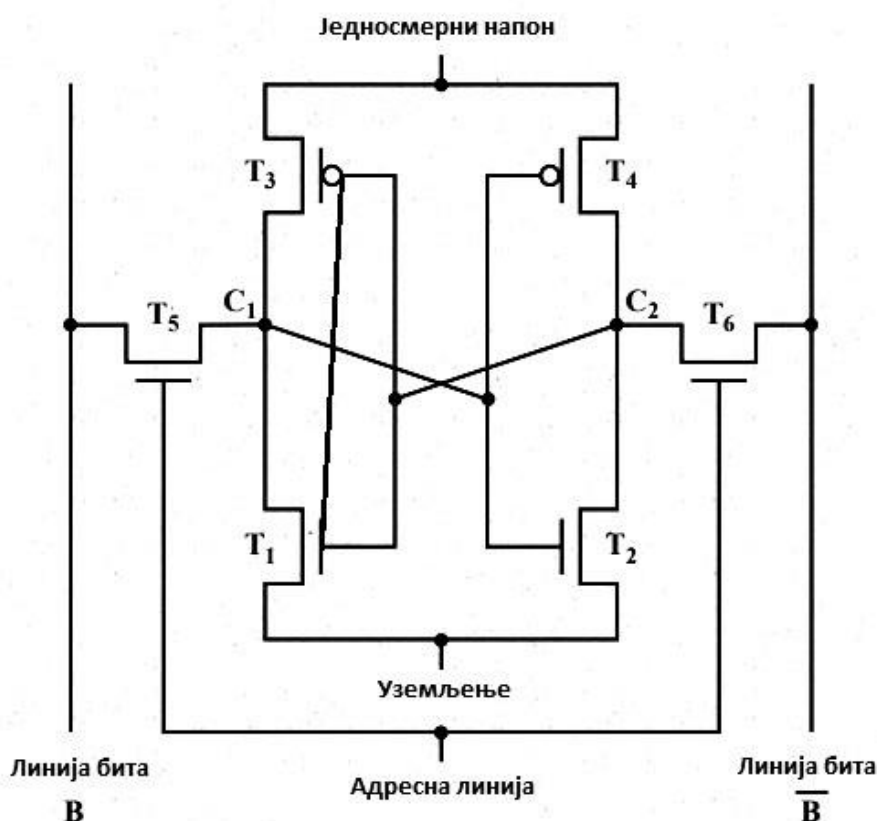
Последњи параметар који посматрамо је број битова меморије који се може купити за долар. Како идемо кроз хијерархију можемо уочити да однос цене и меморије расте низ хијерархију. За главну меморију се мере у доларима по мегабајту, с' тим што у новије време се главна меморија доста развила па можемо гледати и у гигабајтима. Како се спуштамо низ хијерархију све више меморије се може узети за мање новца.

Хијерархијско организовање меморије је битно из неколико разлога, због организовања складиштења података и због што ефикаснијег рада рачунара. Како технологија напредује, организовање меморије омогућаваће бољи, бржи и ефикаснији рад рачунара.[2]

2.4 Основне карактеристике SRAM меморије

Како би се боље разумела разлика између статичке и динамичке меморије, потребно је напоменути и најважније карактеристике статичке рам меморије.

Статичка рам меморија представља дигитални уређај који има исте логичке елементе као и процесор. Бинарне вредности код ове меморије се складиште, коришћењем конфигурације традиционалних флип-флоп логичких кола. Док постоји напајање електричном енергијом, статичка рам меморија ће држати податке.



Слика 1. Структура ћелије SRAM меморије [1]

На слици 1. је приказана класична структура SRAM меморије за појединачну ћелију. Четири транзистора, (T_1 , T_2 , T_3 , T_4) су повезани унакрсно, како би производ тог повезивања довео до стабилног логичког стања. У логичком стању 1, тачка C_1 је на високо а C_2 на ниско: у том стању T_1 и T_4 су искључени а T_2 и T_3 су укључени. У логичком стању 0, C_1 је на ниском а C_2 на високом нивоу: у том стању T_1 и T_4 су укључени а T_2 и T_3 су искључени. Ова два стања су стабилна све док је укључен једносмерни напон. Код DRAM меморије је неопходно освежавање како би се

задржали подаци. Код SRAM меморије није потребно никакво освежавање како би се подаци задржали. На основу слике 1. видимо да је код израде SRAM меморије неопходан већи број логичких елемената од броја истих код DRAM меморије.

И статичка и динамичка меморија су непостојане. То значи да је неопходно трајно напајање електричном енергијом како би се очувале вредности битова, тј. како би се подаци сачували. [1]

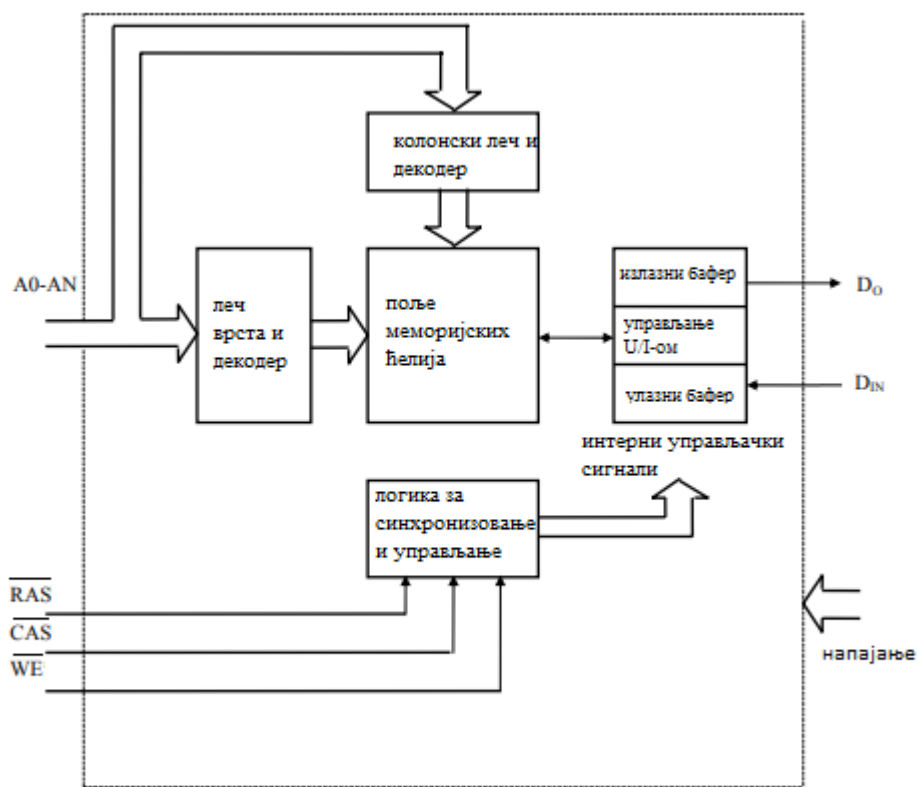
Ћелија динамичке рам меморије је једноставнија и мања од ћелије статичке рам меморије. Дакле DRAM меморија је гушћа (ћелије су мање и број ћелија по јединици површине је већи) и у поређењу са одговарајућом SRAM меморијом је јефтинија. Статичке меморије су мањег капацитета по чипу, а користе се у системима где се захтева већа брзина приступа меморији и мања потрошња струје из извора за напајање. Такође је вероватноћа грешке код статичких меморија мања него код динамичких, тако да се користе у системима где се захтева висока поузданост. Ако гледамо мане, можемо рећи да је мана та што DRAM меморија захтева електронска кола за подршку освежавања. За веће меморије, фиксирана цене кола за освежавање је више него компензована мање променљивом ценом DRAM ћелија. Када је потреба за меморијом велика, све чињенице показују да ће у том случају DRAM меморије бити мање фаворизоване. На крају треба напоменути да су SRAM меморије брже од DRAM меморија. Због наведених карактеристика DRAM меморија се користи као главна меморија а SRAM меморија као кеш меморија. [1]

3.0 Принцип рада DRAM-a

Динамичка RAM меморија је стекла велику популарност из претежно два разлога, а то су:

- а) велики капацитет
- б) релативно мала потрошња [3]

Организација DRAM-a, је најчешће у облику $16\text{ M} \times 1$, $1\text{ G} \times 1$, што значи да су организовани за смештај једног бита по наведеним слотовима. Ова организација се назива 1^* организација. На слици 2 је приказана типична организација оваквих типова DRAM-a.

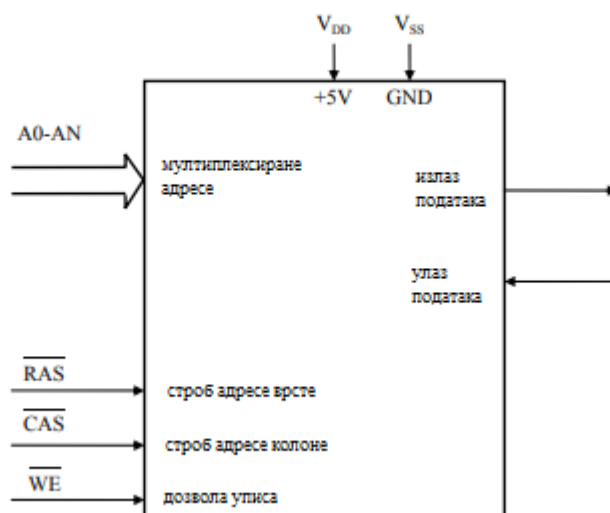


Слика 2 Организација DRAM-a [4]

Поље меморијских ћелија је наважнији а уједно и главни део чипа. У њих се врши смештање нула и јединица. Свака ћелија садржи кондензатор, малог капацитета, и одређен број транзистора. Адресирање ћелија се врши преко адреса врста и адреса колоне. Адресе врста и адресе колоне можемо замислити као координате ћелија. Имајући у виду тежњу да се смањи број спољних интерфејс линија, адресе врсте и колоне се мултиплексирају на истим улазним линијама A_0 - A_N . Адреса врсте се прва поставља на адресним линијама, а одмах за тим се активира сигнал за њихово памћење

RAS (Row Address Strobe). DRAM се одазива памћењем присутних адреса врсте у интерни леч врсте. Затим се врши деактивирање адресе врсте, и након тога се на линијама A0-AN поставља адреса колоне. Долази до активирања сигнала за памћење адреса колоне, CAS (Column Address Strobe). Одазивање DRAM-а се огледа у памћењу адреса колоне у интерни леч колоне. На основу укупне адресе, лоцира се адреса меморијске ћелије на основу декодирања запамћених адреса врсте и адреса колоне. Линија WE (Write Enable) нам показује да ли се приступ односи на операцију уписа или операцију читања. DIN и D0 представљају улазну и излазну линију података DRAM-а. [3,4]

Спољне интерфејс линије стандардног DRAM-а организације *1 приказане су на слици 3.



Слика 3 Спољне интерфејс линије стандардног организација DRAM-а [4]

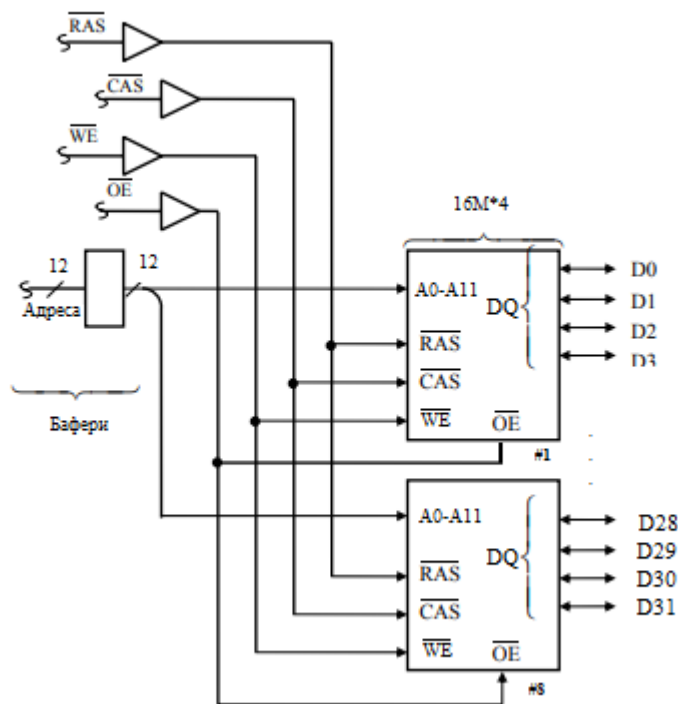
DRAM меморија троши и до десет пута мање снаге, док је у стању приправности (standby), у односу на активно стање. Овим путем се постиже значајно мања средња потрошња.

Што се тиче потребног броја DRAM-ова, које треба уградити код примене меморије микрорачунара, он зависи од:

- а) обима спољне магистрале података CPU-а,
- б) жељеног капацитета меморије,

в) капацитета DRAM-а и његове организације[3]

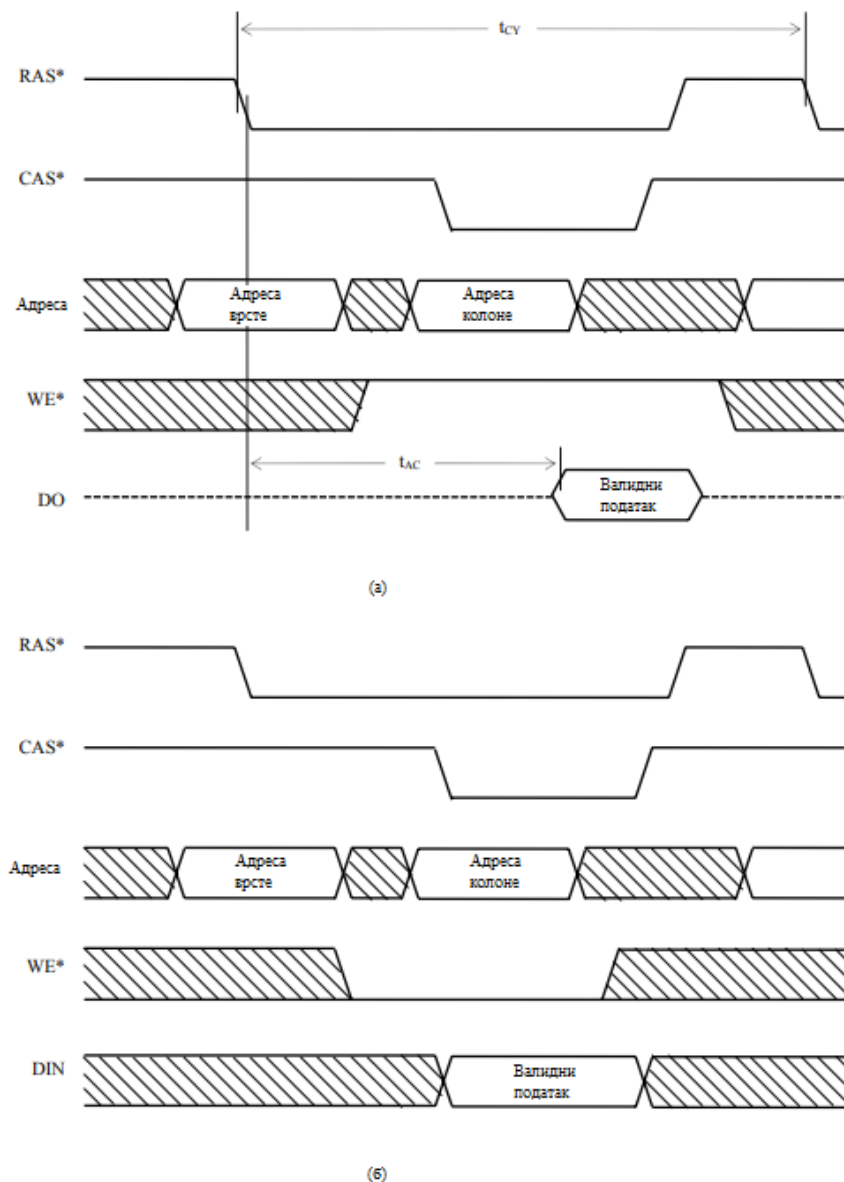
Организација DRAM-а може бити таква да они чувају и више од једног бита по локацији, као што су на пример, 4 М*4, 8 М*8 итд. DRAM чипови од 16 Мб могу бити доступни у обилку 4 М*4 и 8 М*2 организацијама. Овакво алтернативно организовање DRAM-а, у односу на 1* организацију се зове „wide word” организација. Овај тип организације је погодан за меморију малог капацитета. Код 32-битних микрорачунарских система, потребно је само осам 16 М*4 DRAM-а како би се применила меморија капацитета 16 Мб.(Слика 4). Ово се ради због мањег броја извода по чипу. Аналогно томе, и мањег простора којег заузимају на штампаној плочи DRAM меморије већег капацитета „wide word” организације DRAM-а, се израђују у организацији *1. [3]



Слика 4 Организација DRAM-а 16 М*4[3]

3.1 Операције читања и уписивања и временски дијаграми код DRAM

На слици 5 приказан је временски однос између спољних интерфејс сигнала DRAM-а који важи за операције читања и уписа.



Слика 5 Временски однос сигнала за операције читања и уписа [4]

Адреса врсте мора бити стабилна и мора да задовољи захтеве постављања (set up requirements) за дати чип, након чега се активира RAS сигнал. Адреса врсте мора

бити валидна, на улазним линијама, када се активира RAS за период задржавања адресе (address hold time). Период када је RAS сигнал активан, тј. када је на ниско, је лимитиран. Како би се у DRAM-у извршиле неке интерне операције које се тичу пребацивања капацитивног товара, потребно је да RAS сигнал пређе на високо. Укупан минимални временски период када је RAS сигнал на ниско и високо, одређује циклус читања (t_{cu} на слици) DRAM-а. [3,4]

Што се тиче импулса CAS, ограничења која се односе на време постављања и задржавања, и ограничења која се тичу времена трајања импулса су слична. Време приступа (t_{ac}) дефинише се као релативно у односу на предњу ивицу RAS сигнала.

Код операције уписивања секвенцирање сигнала је слично. Ту је WE активни сигнал и из тога се може закључити да се ради о операцији уписивања.

3.2 Освежавање

Како је кондензатор DRAM ћелије мали, оптерећење које се на њему скупи се брзо смањује због цурења струје. Због тога се напон на крајевима кондензатора смањује, до вредности која не приказује реално стање сачуване информације. Како бисе избегао губитак података потребно је:

- а) прочитати вредност напона сваке ћелије
- б) појача
- в) кондензатор поново напунити до почетне вредности

Наведени процес треба понављати периодично и он се назива *освежавање*. У току једне операције DRAM меморија омогућује освежавање свих ћелија које припадају једној врсти. DRAM-ови врше освежавање по 128 врста на 2 ms.

За правилно функционисање освежавања, потребно је уградити тајмер, који ће периодично слати захтеве за освежавање, и бројач на основу ког ће се одређивати редослед адресиране врсте која треба да се освежи. Блок шема на слици 6.



Слика 6 Шема структуре за освежавање DRAM-а [4]

Након примљеног захтева за освежавање следи редослед активности:

- 1) Поставља управљачки сигнал селект (Sel), у такво стање да MUX ка свом излазу усмерава улазе B
- 2) Генерише меморијско-управљачке сигнале који се секвенцирају у сагласности са захтевима циклуса за освежавање
- 3) Након завршетка циклуса, управљачка логика инкрементира бројач. Нови садржај бројача указује на адресу врсте која се наредна освежава.

Када управљачка логика прими два захтева, један захтев се односи на стандардни приступ меморији од стране процесора, а други на освежавање. Један од захтева мора да сачека извршавање. Потребно је пројектовати коло које ће обавити арбитражну функцију. Ако то коло није добро пројектовано може доћи до понашања која се тешко могу логички интерпретирати. Још једна битна ствар код пројектовања ових кола је та, да она у току арбитраже, не смеју изазвати већа кашњења. [4,5,3]

3.3 Кошење сигнала и пропационо кашњење

Када процесор генерише сигнал, који иде до улазне линије DRAM-а, тај сигнал пролази кроз неколико логичких елемената, као на пример гејтова, пре него што стигне до циља. Пре него што стигну до одредишта, адресни сигнали пролазе кроз више MUX-ова и бафера. И подаци који се читају из меморије се морају баферовати па самим тим и они имају одређена кашњења. Ова кашњења продужавају време приступа меморији и доводе до додатних чекања.

Када причамо о временској анализи, морамо узети у обзир још један фактор а то је кошење. Кошење се јавља као последица различитих путева сигнала. Узмимо за пример простирање сигнала преко сличних логичких путева. На основу доступних података можемо одредити само минималну и максималну вредност кашњења за сваки логички елемент. На основу тога не може се тачно предвидети претпостављено кашњење за оба адресна сигнала. Разлика између кашњења која се уноси на оба привидно иста логичка пута зове се *кошење* (skew). Максимална вредност кошења износи

$$\sum d_{imax} - \sum d_{imin}$$

где d_{imax} и d_{imin} представљају максимално и минимално кашњење кроз степен „i“ на логичком путу сигнала.

Смањење кашњења и кошења због логичких елемената постиже се на следеће начине:

а)минимизира се број логичких елемената који се користе на путу сигнала, а могу се користити и кола код којих је пропагационо кашњење мало[3]

б) бирају се логичка кола код којих је разлика између максималног и минималног кашњења мала, самим тим се смањује и кошење

в)користе се логички елементи из истог IC паковања због тога што је разлика између пропагационих кашњења кола на истом чипу у границама од 2 ns.

Кашњење зависи и од температуре. Произвођачи наводе кашњења на температури од око 25 степени. На већим температурама се повећава кашњење за око 30%.

Још један узрок кашњења су и капацитивна оптерећења. Што су она већа, то су времена успона и падова ивица импулса већа па је самим тим и време пропагације сигнала дуже.Класичан случај се јавља код баферских степена за DRAM-ове. Обично се пропагационо кашњење бафера задаје за специфицирано оптерећење. Повећавањем оптерећења повећава се и кашњење.

3.4 Осциловање

На штампаној плочи, везе представљају преносни пут сигнала. Преносна линија је отворена на даљем крају, тако да она рефлектује сигнал назад према крају где је извор сигнала. Ове рефлексије могу довести до осциловања. Осциловање се најлакше може уочити на прелазу из једног логичког стања у друго.

На наведеној слици 7 се види тип осциловања који се јавља код преласка сигнала са високог на ниско. Услед враћања сигнала назад (примарна рефлексција), генерише се напонски подбачај реда -2 V . Како је максимални дозвољени негативни напон на било ком пину DRAM-а, у односу на масу, реда -0.5 V , може доћи до оштећења.



Слика 7 Осциловање сигнала код преласка са високо на ниско [3]

Услед примарне рефлексije сигнал се простире назад према отвореном крају линије. У овом случају се такође рефлектује назад и генерише премештај (секундарна рефлексija). И сада се генерише побачај. Поред опасности по DRAM, осцилације могу изазвати и друге проблеме. Последице осциловања су озбиљније и критичније код тактних сигнала. Ивице такта могу изгледати као вишеструки сигнали, што доводи до погрешног окидања. [3]

3.5 Шумови

Што се тиче динамичких меморија, ситуација са шумовима је знатно компликованија, него код осталих дигиталних система, због тога што меморија у свом склопу има кондензаторе који се пуне и празне у току рада. Треба обратити пажњу на операцију освежавања, јер у току операције долази до истовременог активирања свих меморијских банака. Услед великих струјних краткотрајних оптерећења које (current spikes) на линији за довод напона напајања, V_{cc} , долази до стварања непожељних прелаза који могу изазвати погрешна окидања или постављања логике. Како не би дошло до нежељених појава, ове транзиентне појаве се морају држати под контролом, а то се постиже уградњом филтерских кондензатора. На високим фреквенцијама

филтрирање (оклањање сметњи) се врши коришћењем керамичких или других монолитних кондензатора, који имају малу индуктивност. Због сигурности, препоручује се уграђивање оваквих кондензатора што ближе могуће између пинова Vcc и GND за сваки DRAM.

За отклањање ниско-фреквентних сметњи користе се кондензатори већег капацитета. Најчешће се користе кондензатори фреквенција вредности од 15 μF до 47 μF . Ови кондензатори штите Vcc од падова испод прихватљивог нивоа. Постављање ових кондензатора се врши на улазном доводу Vcc на штампаној плочи што ближе конектору. Ако су плоче веће, врши се распоређивање додатних кондензатора преко целе плоче. [3]

4.0 Разлика између SRAM-а и DRAM-а

Постоји више разлика између SRAM-а и DRAM-а. Приступ подацима је једна од њих. Приступ подацима код DRAM-а је различит у односу на приступ подацима код SRAM-а. Адресне линије се мултиплексирају код DRAM-а, што доводи до смањења броја ножица по чипу. Такође код DRAM-а приступ траженој адреси се врши прво унутрашњим лечовањем адресе врсте уз помоћ сигнала RAS, а након тога адресе колоне помоћу сигнала CAS. Због мултиплексирања обе адресе DRAM од 256 kb се израђује у 16-пинском паковању, а DRAM од 1 Mb у 8-пинском.

Разлику између ова два типа меморије је и у потребама за освежавање и допуњавање (refresh and recharge). У склопу DRAM ћелије се налази кондензатор за памћење. Када се овај кондензатор напуни, он мора да се периодично освежава, како се не би испразнио. DRAM архитектуре симултано освежавају све ћелије за изабрану врсту сваки пут када се приступа врсти. Неки DRAM-ови резервишу MSB као селективит за интерни MUX који селекује податке у две банке поља. Код ових DRAM-ова LS адресни битови приступају меморијским ћелијама обе банке истовремено. Овако се постиже дупло мањи број потребних освежавајућих циклуса.

Још једна специфична карактеристика DRAM-а је допуњење електричног товара на кондензатору за памћење информације. DRAM меморија се мало испразни док је активна операција читања. Зато се после операције читања мора извршити поновни упис у ћелију. Управо та операција, која врши поновни упис у ћелију, се зове допуњење и њу DRAM аутоматски обавља. Време потребно да се обави ова операција назива се време допуњења (precharge period) и оно се мора рачунати када се врши циклус читања DRAM-а.

Најсложенија карактеристика DRAM-а је тајминг дијаграм. У табели 1 су ознаке које ћемо користити[1,3]

Параметар	Скраћеница
Колона или CAS	C
Врста или RAS	R
Ниво сигнала ниско	L
Ниво сигнала високо	H
Време приступа (access times)	a
Време држања (hold times)	h
Време постављања (setup times)	Su
Ширина импулса (puls width)	w
Операција читања	rd
Операција уписа	W
Циклус	c
Адресе	A
Подаци	D

Табела 1 Ознаке које се користе [3]

Минимална ширина импулса, за коју је CAS линија на ниско, означава се са $tw(CL)$. Слово w се односи на ширину (width) импулса, C означава CAS, а L значи ниско (low).

Како би прилагодили рад DRAM-а временским сигнаlima, у меморијски систем се мора уградити контролер, који генерише све управљачке сигнале. Тај контролер ће служити као интерфејс између DRAM-а и системске магистрале. Најчешће се код процесора уграђују спољашња кола која функционишу као DRAM контролери. Он мора да усагласи системски приступ DRAM-у са једне стране, а са друге, приступ логике за освежавање у оквиру задатог времена за освежавање. Можемо рећи да DRAM контролер „вара” процесор тако што са DRAM-ом комуницира као са SRAM-ом.

5.0 Детекција и корекција грешака код DRAM-а

Како је могуће појављивање грешака код DRAM-а, потребно је увођење детекције и корекције грешака. Разлози за увођење детекције и корекције грешака су следећи:

а) код DRAM-а великог капацитета, вероватноћа појаве грешака је велика

б) последице појаве грешака су веома неповољне, па сходно томе, потребно је урадити све како не би дошло до појаве грешака

Узмимо за пример када CPU шаље неку инструкцију. Ако се у опкоду инструкције промени вредност једног бита онда ће процесор слати неки други тип инструкције од намењеног, тј. информацију која није била предвиђена у датом тренутку. Ако се првобитна инструкција промени у инструкцију типа JMP процесор ће стартовати са извршењем неког другог програма и вероватно ће доћи до краха система, што је неприхватљиво за већину апликација. Када се јаве меморијске грешке, оне се морају открити, а след активности мора водити ка томе да се сигнализира CPU да је дошло до грешке. Детекција и корекција грешака се врши помоћу *кодова за детекцију-и-корекцију-грешака*, тј. EDAC (Error-Detection-And-Correction) кодови или ECC (Error Checking and Correction).

У зависности од узрока појаве, меморијске грешке можемо поделити у:

а) перманентне грешке (hard errors) – ове грешке су последице неког физичког оштећења, на пример кратак спој између два пина или дефект DRAM-а. Ове грешке се не могу уклонити софтверским путем, већ је потребно заменити део хардвера.

б) случајне грешке (soft errors) – нису ни у каквој вези са хардверским грешкама и нису предвидљиве. Разни су узроци за појаву оваквих грешака као на пример прекорачење напона напајања, сметње у напону напајања, повећана температура итд. [3]

Главни узрок за појаву непредвидивих грешака су α -честице. DRAM чипови се испоручују у пластичним паковањима. Та пластична паковања садрже у себи капсулиране радиоактивне материјале као што су уранијум и торијум. Ови материјали емитују α -честице. Механизам памћења код DRAM ћелије заснива се на кондензатору, чије стање одређује када је запамћена 0 а када 1. α -честице доводе до јонизације и самим тим неутралишу товар на кондензатору, и тако чине процес одређивања логичке 1 или 0 неодређенијим. Када дође до овакве грешке, она се елиминише корекцијом и врши се поновно уписивање података. Како код SRAM-ова, PROM-ова и EPROM-ова не постоје кондензатори α -честице немају никакав утицај.

Произвођачи за сваки свој склоп одређују *учесталост отказа* (failure rate). Учесталост отказа се дефинише као вероватноћа отказа склопа у % за временски

период од 1000 часова. На пример ако је учесталост отказа код неког DRAM-а 0.14% /1000 часова, то значи да за период рада од 1000 часова вероватноћа да ће се код датог DRAM-а јавити отказ износи 0,0014. Алтернативно, учесталост отказа се изражава у FIT (отказ у времену - failures in time). Један отказ у времену је један отказ за време од 10^9 часова. Што значи да је учесталост отказа од 0.14% /1000 часова еквивалентно $0,0014 \cdot 10^6 = 1400$ FIT-а. Учесталост отказа, без обзира на све, указује на то колико је неки склоп поуздан за коришћење.

Учесталости отказа се морају сабирати. Ако је за једну DRAM организацију $64k \cdot 1$ учесталост отказа 1200 FIT-а, за DRAM меморијско поље организације $64k \cdot 16$ учесталост отказа биће $1200 \cdot 16 = 19200$ FIT-а. Потребно је сабрати перманентне и случајне грешке, како би се одредила укупна, или сложена, учесталост отказа DRAM-а. Ако $64k \cdot 1$ DRAM има перманентну учесталост отказа од 200 FIT-а, а случајну учесталост отказа 1000 FIT-а, онда ће укупна учесталост отказа бити 1200 FIT-а. Како би се одредила интегрална поузданост отказа, морамо познавати учесталост отказа осталих компоненти. У табели 2 приказане су типичне вредности учесталости отказа за неке компоненте.

Тип компоненте	Типична учесталост отказа (FIT)
Отпорник	1
Диода	1
SSI	10
Керамички кондензатор	10
Тнтл кондензатор	20
MSI	50
Штампана плоча	500
$64k \cdot 1$ DRAM	1200

Табела 2 Вредности учесталости отказа [3]

Још један начин за мерење поузданости је *средње-време-између-отказа* (MTBF – Mean Time Between Failures). MTBF нам показује колико дуго у просеку један уређај може да ради без грешке. MTBF је у суштини реципрочна вредност учестаности отказа.

5.1 Заштита од грешака

Заштите од грешака можемо поделити у три категорије:

- а) *генерисање парности и провера* – најједноставнија мера за проверу
- б) *детекција и корекција грешака* – најчешће коришћена мера заштите која се примењује код малих и великих рачунара
- в) *потпуна редундантност* – користи се код високо поузданих система и обично се изводи као трострука

Битно је напоменути да све три категорије користе исти принцип. Подаци се уписују у меморију. Пре него што се упишу у меморију, податку се придружују и редундантни информациони битови. Редундантни битови омогућују детекцију и корекцију грешке. Број грешака који се могу детектовати или детектовати и коректовати, директно је сразмеран броју редундантних битова придружених датом податку.

5.2 Парност

Парност представља једну од најједноставнијих техника за детекцију грешке. Парност се одређује бројем јединица коју садржи реч. Парност може бити парна или непарна. На пример бајтови 00111001 и 01001000 садрже паран, а бајтови 01000000 и 01101000 непаран број јединица. Ако се сви битови *n*-битне речи доведу на улаз дате комбинационе мреже за проверу парности, на излазу ће се генерисати 0 ако је број јединица паран, а 1 ако је број јединица непаран. Суштина технике парности је придруживање само једног бита, названог *бит парности* оригиналном податку. Тај бит парности може имати само две вредности, 1 и 0, у зависности од:

- а) броја јединица у изворном податку
- б) коришћене шеме за парност [3]

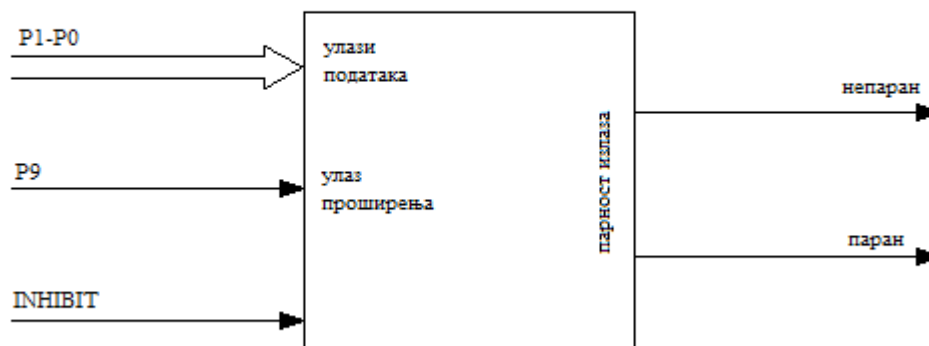
Код шеме – *парна парност* – захтева се да укупни број 1 укључујући и бит парности, буде паран. Ако изворни податак садржи паран број јединица, бит

парности биће додељена вредност 0. Ако изворни податак садржи непаран број јединица, биту парности биће додељена вредност 1. Код шеме – *непарна парност* – потребно је да укупни број 1 буде непаран.

Неопходно је да се податку генерише парност, пре него што се он упише у меморију. То се ради задатом шемом за парност која се усваја. Бит парности који је генерисан се смешта у меморију заједно са податком. Парност податка се одређује када се он чита из меморије. Након тога се врши поређење са почетним битом парности, који се чита из меморије. Ако су битови парности различити сигнализира се грешка CPU-у.

Овом техником је могуће детектовати просте грешке, или вишеструке грешке када је број битова непаран. Двоструке или вишеструке грешке код којих је број битова паран, се не могу детектовати. Како је највећи број грешака које се јављају једнострук, парност је веома ефикасан механизам за детекцију грешака. Двоструке грешке се јављају 50 до 100 пута ређе од једноструких. Могуће је повећати способност детектовања грешака, у случају да када податак који се преноси чине два или већи број бајтова. Ово се може применити ако се по сваком бајту користи један бит парности. На пример код 16-битне речи, ако се за сваки бајт користи по један бит за парност, могу се детектовати двоструке грешке, али само ако се грешке јављају на различитим бајтовима.

Коло које се користи за генерисање бита парности, пре него што се податак упише у меморију, и за проверу парности након читања меморије зове се *parity generator/checker*. Типичан пример овог кола је коло 74820. На слици 8 може се видети спољни интерфејс овог кола. Овај чип може генерисати, за 8-битни податак, парну и непарну парност. [4,3]



Слика 8 Провера парности[3]

Улаз P9 омогућава избор парне или непарне парности. Постоје апликације код којих треба генерисати парност над пољем података већим од 8 битова. Код ових апликација P9 се користи као улаз проширења, тј. повезује се на излаз другог 74820. Када је INHIBIT улаз активан оба излаза парности су забрањена. У одредишту, месту где се проверава парност, се на P9 доводи бит парности, који је генерисан од изворног податка.

Овом техником се могу само детектовати грешке, али се не могу кориговати. У том случају, типично је да систем код ког се јавила грешка поново покрене од почетка. Ако је грешка случајна, већа је вероватноћа да се она неће поновити, па систем може наставити са нормалним радом. Ако је грешка перманентна, након другог покушаја и детекције грешке систем ће престати са радом како не би дошло до неког већег оштећења или других озбиљнијих последица.

5.3 Детекција и корекција грешака

EDAC (Error Correcting And Detecting) технике, представљају проширење једноставне технике парности. Основа технике је коришћење већег броја битова парности, који се генеришу одређеним подскуповима битова података. Овакви битови парности се називају *битови провере* (check bits) и такође се смештају заједно са битовима у меморију. Битови провере се израчунавају сваки пут када се податак уписује у меморију.

Могуће је детектовати и кориговати једноструке грешке EDAC техником. Постоји могућност детектовања двоструких и вишеструких грешака, али се оне не могу кориговати и на њих систем гледа као на грешке парности. Основни Хамингов код захтева $p = \log_2 n$ битова за локацију грешке у оквиру n -битне речи и додатна два бита од којих се један користи за детекцију двоструке, а други за детекцију вишеструке грешке. У суштини, када се обим речи дуплира потребан је један додатни бит провере.

У табели 3 је наведен број неопходних битова провере за детекцију двоструке грешке а корекцију једноструке у случају када је податак 8-, 16-, 32-, 64-битни. За 8-битни податак је потребно 62.5% додатног меморијског капацитета, ако се уведу битови провере. Ово прекорачење пада на 40 % за 16-битне, на 22 % за 32-битне, а на 12.5 % за 64-битне податке.

Дужина податка	Број битова провере	Прекорачење
8	5	62.5%
16	6	37.5%
32	7	21.9%
64	8	12.5%

Табела 3 Битови провере [3]

Битови провере се такође поново израчунавају када се подаци читају из меморије. Изводи се XOR операција између ново генерисаних и меморисаних битова провере, како би се утврдило да ли постоји грешка и ако постоји на којој је бит локацији. Резултат поређења зове се *синдром*. Синдром је нула, у случају када су оба бита провере идентична. Ако постоји грешка синдром је различит од нуле, а у зависности од његове вредности одређује се и локација погрешног бита. Синдром може да идентификује грешке не само у оквиру података него и у оквиру битова провере. Све двобитне и вишебитне грешке је могуће детектовати али не и кориговати. [3]

Како би EDAC техника функционисала коректно, потребно је проширити меморијски капацитет. Такође је потребно и уграђивање додатне хардверске логике, како би било могуће извршавање операција које су неопходне за процес детекције и корекције грешака. За обављање ових операција, које су веома комплексне, користе се специјалне VLSI компоненте које се називају EDAC јединице.

6.0 Архитектуре система за детекцију и корекцију грешака

EDAC има две различите имплементације које су познате као:

- а) *придružена* EDAC (Fly-By EDAC)
- б) *активна* EDAC (Flow-Through EDAC)

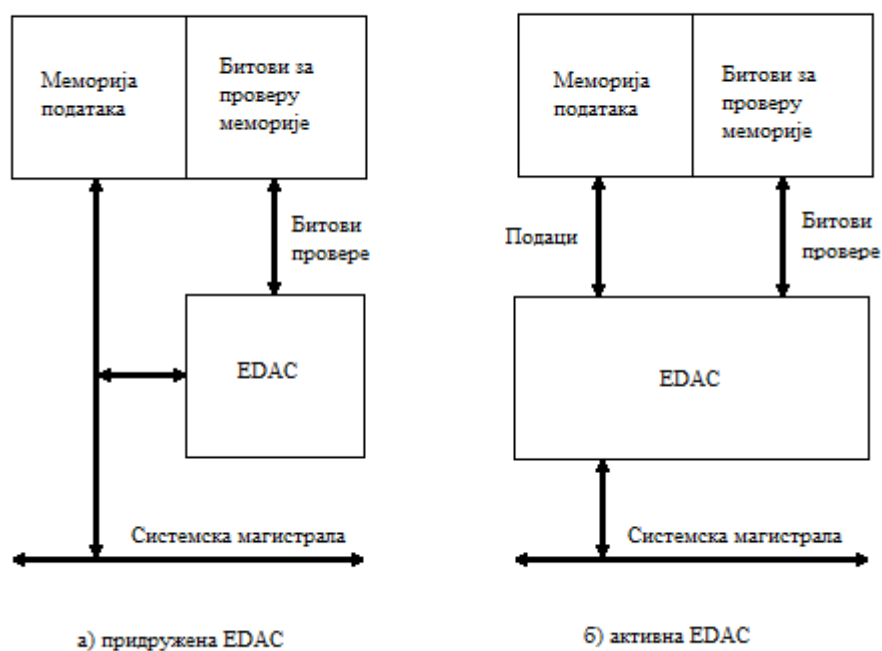
6.1 Придružена EDAC

У енглеском језику овај блок је познат и под називом Check-Only-Configuration. Придružена EDAC јединица има за функцију, да врши проверу грешке код свих речи које се читају из меморије. Грешка која се открије се сигнализира процесору. Након

тога покреће се интерна рутина за исправљање грешке од стране EDAC-а. Поента овог је да се комплементира вредност бита на чијој се позицији нашла грешка. Неважећи податак се након тога уклања а на његово место се поставља нови, важећи. Систем увек претпоставља да не постоји грешка што је у највећем броју случајева тачно. На овај начин се обезбеђује велика брзина приступа меморији. Када се јави грешка активирају се излазне линије EDAC јединице Error или Multiple Error које воде ка процесору. Процесор може да прекине текући меморијски циклус или да га продужи убацивањем стања чекања, што условљава кашњење. Ова одлука важи великим делом од стања на линијама Error или Multiple Error.

6.2 Активна EDAC

Код активне EDAC, врши се постављање између меморије и системске магистрале. Назив ове конфигурације је и Correct Always. Ова конфигурација узима да све речи из меморије имају грешке. Пут података у овом случају је меморија EDAC магистрала и обрнуто (магистрала EDAC меморија). Суштина се огледа у томе што ова јединица претпоставља да све речи имају грешке, и због тога се увек извршава корекциони циклус, што доводи до поједнстављања тајминга кола. Брзина рада EDAC је независна од тога да ли постоји грешка или не, па су зато сви меморијски циклуси спорији. Код микрорачунарских система који не подржавају рестартовање магистралног циклуса пожељна је примена ове технике детекцију грешке (слика 9). [3]



Слика 9 Повезивање EDAC јединице [3]

Један занимљив тип архитектуре за детекцију и корекцију је и *чишћење меморије*. Она врши проверу DRAM-а у току циклуса освежавања. У току сваког циклуса *освежавања-са-чишћењем* проверава се по једна реч DRAM-а, а затим се, ако се идентификује грешка, врши корекција и поновно уписивање у меморију.

Чишћење меморије се обавља на следећи начин. Уколико се јави грешка, активира се операција читања, модификације и уписа (R/M/W). У односу на нормалан циклус читања и уписивања, овај циклус траје дужи временски период. RAS сигнал омогућава приступ информацији врсте у свакој банци својом потврдом сигнала. Ако се детектује грешка, врши се упис у оквиру циклуса освежавања, а самим тим се мора убацити и стање чекања како би се циклус продужио. Како се “soft” грешке јављају у деловима меморије које се не користе често, овим системом је повећана поузданост система. Треба напоменути да ово важи за случај када се истовремено врши освежавање неколико меморијских банки, и тада се корекција врши само над једном.

Освежавање се може вршити и преко RAS линије. Међутим, овим путем се не може вршити чишћење. У овом случају се активирају све RAS линије, а CAS су неактивне. Када је у питању циклус чишћења, активирају се све RAS линије а само једна CAS. Ако се грешке идентификују и коригују у току циклуса освежавања, оне се не пријављују процесору.

6.3 Имплементација EDAC јединице

На слици 10 приказан је дијаграм система за корекцију грешака и ток података који важи у току циклуса упис. Активна је 16-битна EDAC јединица.

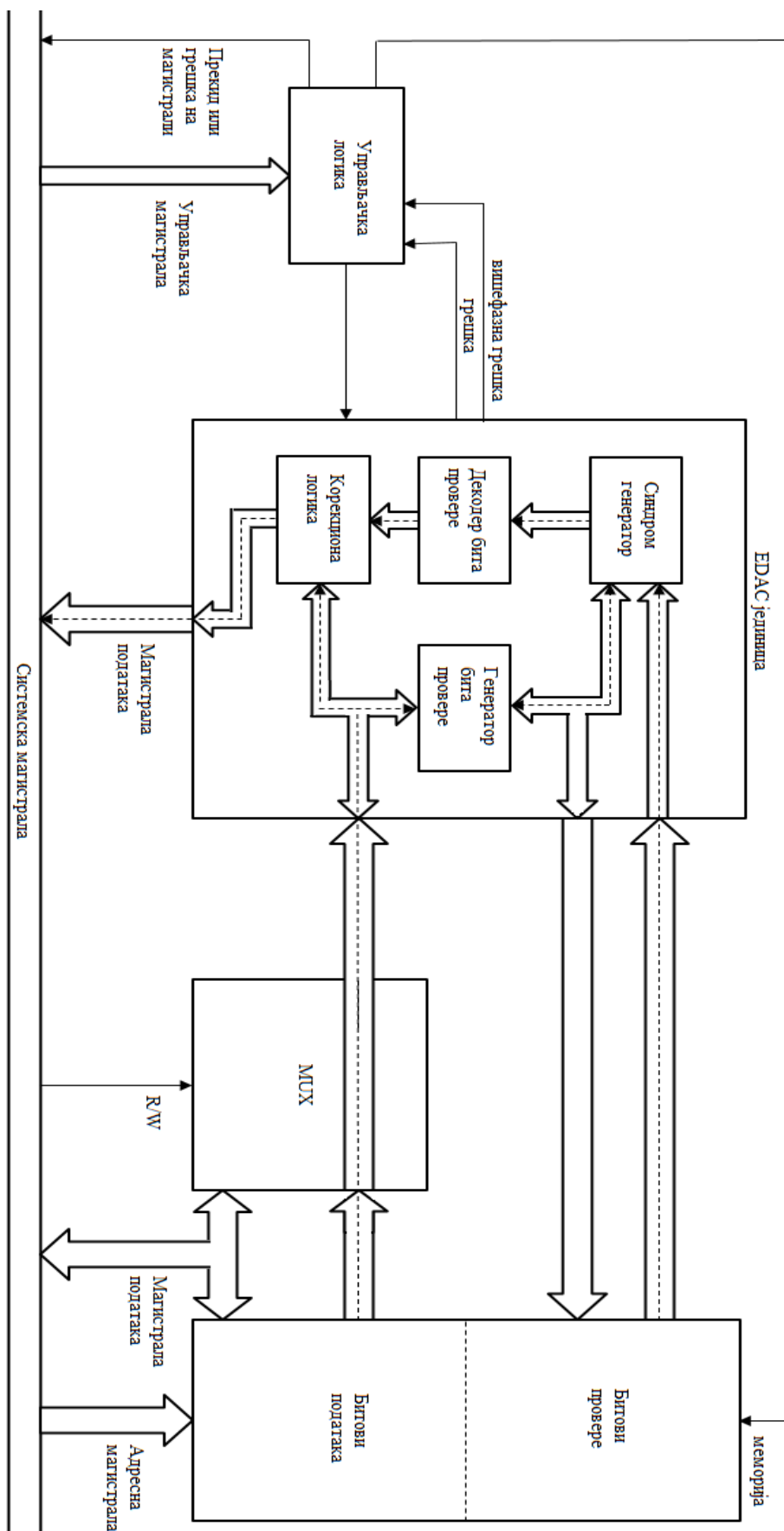
У данашње време, EDAC јединице се израђује као VLSLIC. Пре него што стигну до системске магистрале, подаци пролазе кроз корекциону логику EDAC јединице. Врши се поновно генрисање битова провере, и њихово комбиновање са битовима провере који се читају из меморије. Онда се формира синдром и он се декодира од стране декодера бита грешке. Декодер идентификује погрешан бит који је потребан корекционој логици и активира излазе *грешка* и *вишеструка грешка*. Ако постоји погрешан бит, корекциона логика га инвертује и подаци се прослеђују системској магистрали. Ако се активира излаз вишеструка грешка, онда постоји више погрешних бит позиција и тај податак се не може кориговати. Када се овај сигнал активира долази до прекида циклуса.

Како је раније напоменуто, EDAC јединице су стандардно пројектоване за рад са 16-битним подацима, али могу радити и са бајтовима. Ако је неопходан рад са 32-битним или 64-битним подацима, могуће је повезивање две или више јединица. Остале карактеристике EDAC јединица су:

а) детекција свих 1 и свих 0: ови облици указују на неке главне отказе.

б) маркери грешака: две посебне линије EDAC јединица омогућавају информације о природи грешака, да ли су једноструке или вишеструке. Ако постоји двострука грешка у току читања/корекције, или једнострука грешка у току читања/детекције, рад CPU-а се зауставља. Ако се јави једнострука грешка у току читања/корекције, рад се не може зауставити. Док је прекид активан CPU чита синдром и локацију на чијој адреси је грешка.

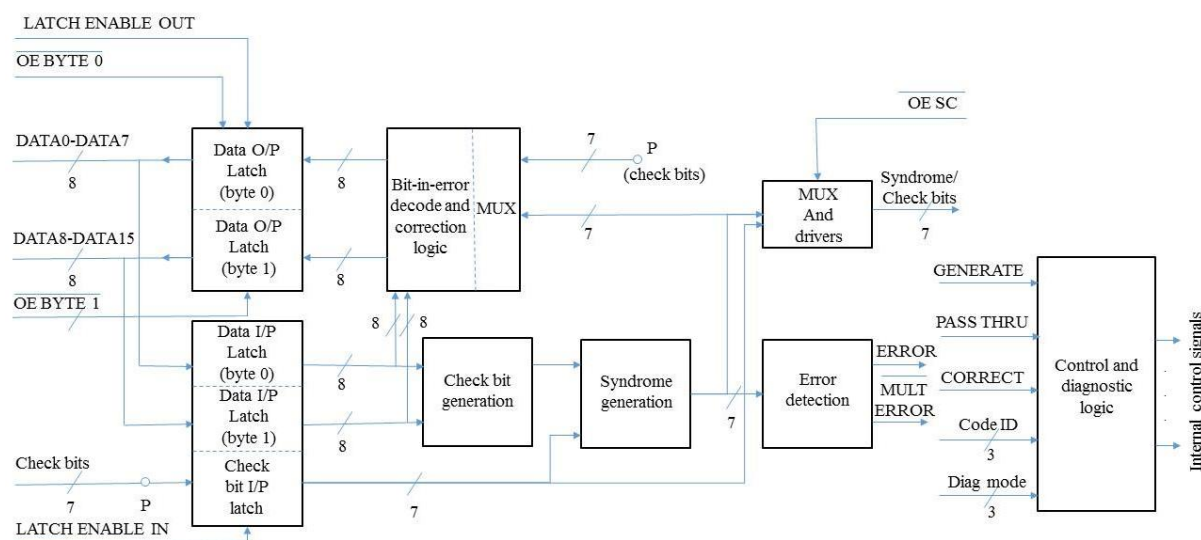
в) доступност синдрому: код EDAC јединица, синдром битови су, најчешће, доступни спољном интерфејсу. CPU налази погрешан бит, читањем и декодирањем синдрома. Овим путем CPU може да држи информацију о грешци због каснијег одржавања. [2,4]



Слика 10 Ток података у току циклуса упис [3]

6.4 Структура EDAC

Структуру EDAC јединице анализираћемо на примеру чипа Am 2960 (слика 11).



Слика 11 Структура EDAC јединице[3]

Линије синдром/провера врше мултиплексовање битове-провера, који се генеришу у току циклуса упис, са синдромом који се издваја у току операције читања. EDAC јединица има два 16-битна леча који се управљају на бајтовском нивоу. Како би се чували битови-провере уграђује се и 7-битни леч. Када се ове јединице споје каскадно, омогућава се директна манипулација и са 32-битним подацима.

GENERATE и CORRECT, су улазне управљачке линије. Оне одговарају постављању начина рада типа у упис и читање/корекција. Подаци које CPU смешта у улазни леч, се доводе на генераторску логику за бит проверу. Битови-провере се преносе преко синдром генераторске логики и мултиплексера. Они су присутни на излазима синдром/провера када OESC активира ове излазе. За то време подаци се као непромењени преносе од улазних лечева преко блока који представља логика за корекцију. Треба напоменути да се садржај излазних лечева података моментално ажурира, онолико дуго колико је LATCH ENABLE OUT линија на високо. Када су битови провере спремни, активирају се излазне линије помоћу којих се врши презентација података за упис у меморију.

Код корекције, из меморије се читају подаци и битови-провере. Активирањем линије LATCH ENABLE IN се смештају у одговарајуће улазне лечеве. Подаци се доводе до логики за бит-проверу, како би се издвојила нова бит провера. Сада, логика

синдром генератора пореди прочитане информације о бит-провери и информације о бит-провери која је новогенерисана. Синдром тада иде ка:

а) логици за детекцију грешке – ако било који синдром бит има вредност 1 активира се излазна линија ERROR. Ако грешка има више од једног бита активира се линија MULT ERROR.

б) мултиплексерима и драјверима како би приступио CPU

в) логици за декодирање и кориговање која коригује податак инвертовањем погрешног бита. Ти подаци се онда уписују у излазним лечевима за податке због предаје CPU. Ако се јави вишеструка грешка, излаз процесора корекције је неодређен.

Am 2960 у раду игнорише садржај излазних лечева за податке. Тада CPU чита исти податак који се из меморије предаје у улазни лек за податке. Када је обим речи већи од 16 битоа користе се линије PASS THRU Code и ID. Чип има и могућност дијагностичког начина рада (Diag Mode Signals). У току овог начина рада се може сам чип испитати од стране CPU-а, због тачности у раду. [3]

7.0 Пројектовање DRAM контролера

Главна компонента на коју треба посебно обратити пажњу код пројектовања DRAM-а је DRAM контролер (DMC).

Пре почетка самог контролера, потребно је извршити анализу, тј. испитати које функције ће обављати и где ће се примењивати сама меморија. Након тога се врши подела на функционалне блокове, а тек онда пројектовање и примена сваког од тих блокова. Како би ови блокови могли да комуницирају међусобно, а то се врши њиховим међусобним повезивањем путем кола. Ова кола одређују интеракцију између блокова.

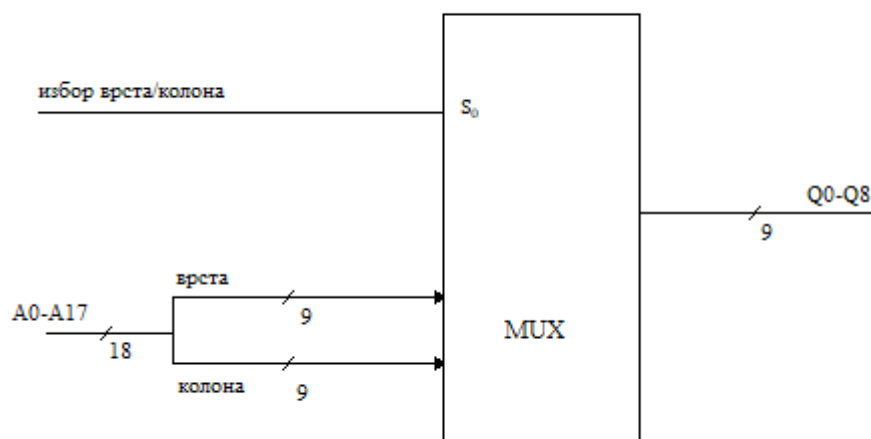
DRAM контролер, има улогу да обезбеди пренос података између CPU-а и DRAM-ова. DMC поставља све врсте и колоне за DRAM, одређује параметре за синхронизацију рада, генерише захтеве за освежавање и одговоран је за секвенцирање начина рада код приступа самој меморији.

Што се тиче грађе DMC-а, он се састоји од функционалних блокова који су повезани преко секвенцијалне машине која извршава супервизорске задатке. [4]

DRAM контролер прима n-битну адресу од CPU-а, и на основу те адресе, поделом саме адресе, добије адресу врсте и адресу колоне DRAM-а. DRAM прихвата адресу врста и колона на истим пиновима, па је потребно у контролер уградити и

систем који ће мултипрексирати више и ниже половине p -битне адресе, на излазној адресној магистрали. (слика 12)

Постоје процесори који имају мултиплексирану магистралу адресе, па је због тога потребно уградити спољна кола која лечују адресе пре него што CPU користи магистралу за пренос података. Код процесора код којих су адресна магистрала и магистрала за податке раздвојене не постоји потреба за уграђивањем додатних лечева.



Слика 12 Мултиплексер раздваја адресе CPU-а на адресе врста и адресе колоне [3]

Контролер има улогу да омогући да се сви DRAM-ови из датог система освежавају у оквиру максималног дозвољеног периода. Типови освежавања могу бити: *освежавање само-RAS*, *RAS-пре-CAS освежавање*, *"бурст" освежавање* итд.

Код освежавања само-RAS и освежавања-чишћењем меморије логика контролера је веома комплексна. Код методе само-RAS, DMC омогућава интервал освежења сваке нове врсте на неколико микросекунди. Код освежавања са чишћењем, DMC у току сваког интервала освежавања мора да генерише адресу врсте и колоне, помоћу бројача. Код методе освежавања са чишћењем потребно је обезбедити да се свакој врсти приступа једном у току освежавања сваке колоне. То се постиже коришћењем битова најмање тежине бројача као адресу врсте. Код других метода освежавања, конструкција контролера је знатно једноставнија. То је због тога што методе користе адресни бројач који постоји у DRAM. У контролер је неопходно уградити тајмер за стартовање циклуса. [3]

Поред ових функција, контролер има функцију и да прихвата управљачке сигнале који стижу од стране CPU-а. То се постиже уградњом функционалног интерфејс блока. Овај блок мора генерисати исте излазне сигнале, без обзира на врсту и облик управљачког сигнала. Како би се прилагодио различитим врстама процесора, могуће је извршити промену интерфејс блока. Ако имамо случај са више CPU-а

уграђује се већи број управљачко-сигналних интерфејс блокова. Ово додатно компликује пројектовање контролера. У случају са више интерфејса, а само једним бројачем за освежавање, у DMC контролер се уграђује арбитар, који врши планирање и контролу захтева који пристижу од различитих CPU-а. Захтев за приступ DRAM-у од стране DMC-а има највиши приоритет. То се дешава зато што освежавање представља важећи захтев. Како је напоменуто раније, DMC треба да изврши освежавање DRAM ћелија у оквиру специфичног периода освежавања. Зато DMC третира освежавање као захтев који је по приоритету виши од свих осталих захтева. У DMC је потребно уградити и блок који повезује све функције, како би се координирала активност DRAM контролера. Ове задатке обавља коначни аутомат FSM (Finite State Machine).

Како DMC независно управља радом неколико банака. Да би се то остварило, неопходно је створити посебне RAS, CAS и WRITE сигнале за сваку банку. То се постиже уградњом демултиплексера који шаљу независне управљачке сигнале, који се бирају помоћу селектора сигнала.

Упркос усавршавању DRAM-меморија, и даље постоји неусклађеност у раду DRAM-а и CPU-а. Самим тим потребно је убацити стања чекања, како би процесор сачекао DRAM, да заврши приступ или освежавање. Док чека CPU не ради ништа. Ако је чекање мање, цео систем ће брже радити. Ова стања се аутоматски убацују. DMC обезбеђује сигнал који говори да је DRAM спреман.

Унутар DMC-а, када CPU хоће да приступи меморији, интерфејс блок, преко арбитра предаје сигнале FSM-у. Када је меморија заузета FSM поставља интерфејс блок на стање чекања (hold). После тога интерфејс блок шаље управљачки сигнал који показује CPU-у да убаци стање чекања.

Ако користе исту тактну побуду, секвенцијална кола у DMC-у, раде синхронно са CPU-ом, или асинхронно, ако користе посебну. Синхрони рад представља бољу опцију, јер се временски усклађује синхронизација различитих команди. Међутим код синхроног рада, CPU не може имати високу тактну фреквенцију због кашњења у обради сигнала. Ако имамо систем са већим бројем CPU-а, који раде различитом тактном побудом, не може се користити синхрони рад DMC-а.

Код пројектовања, користи се и систем проточности, како би се убрзао рад. Код ове технике користи се преклапање извршења различитих операција. Када DMC користи проточност, у току циклуса приступ- меморији раније лечује адресе и податке. Тако су CPU или меморија доступни другим операцијама пре него што се циклус приступа заврши. Када DMC користи ту технику, каже се да користи технику *лечуј и продужи* (latch and go).

Код ове технике, у току циклуса уписа, DMC лечује адресе и податке из CPU-а чим они постану доступни. DMC онда шаље команду која, можемо рећи, „вара”, CPU тако што он „мисли” да је упис завршен. CPU ослобађа магистралу и покреће следећу операцију, док меморија завршава циклус уписа. Овако се спречава појава већих стања

чекања CPU-а који се јављају у току циклуса упис. У току циклуса читања, DMC лечује CPU-ову адресу и податак из меморије, чим податак постане важећи, и онда се предаје CPU-у. Овако се код система са већим бројем CPU-ова, омогућава да спорији CPU-ови лечују податке, а бржи директно приступају меморији. [3,4]

Сада се поставља питање какав DMC треба уградити у систем? Постоји више солуција. Оне су следеће:

а) уградити DMC који је компатибилан са типом CPU-а. Произвођачи нуде DRAM контролере који су компатибилни са форматима сигнала одређеног CPU-а. Ако су у току пројектовања постављени посебни захтеви, рад DMC-а потребно је кориговати како би он био прилагођен задатим условима.

б) коришћење DRAM контролера који је заснован на раду већег броја DMC-ова. Код овајквог приступа постоји већи број LSI чипова који обављају различите DMC функције, што уједно представља и највећи недостатак ове методе.

в) пројектовање ASIC DMC – предност овог решења је што користи предности првог и другог начина.

7.1 Приступ код DRAM-а

Адресе врста а и адресе колона код DRAM-ова се мултиплексирају на истом скупу адресних пинова. Адресе врсте се лечују опадајућом ивицом RAS импулса, а затим се лечују адресе колоне опадајућом ивицом CAS импулса. У зависности од стања сигнала WE зависи да ли је у питању циклус читања или уписа. Ако је сигнал на ниско у питању је операција уписа, а у супротном је читање. Овај начин приступа се зове RAS/MSEL/CAS фундаментални приступ. Како би се скратило време приступа DRAM-у користе се и други типови поред наведеног. Ти типови су:

- а) *странични* (Page mode)
- б) *побољшани странични* (enhanced page mode или fast page mode)
- в) *статичко колонски* (static column mode)
- г) *тактовани* (nibble или ripple mode)

Странични метод приступа почиње довођењем адресе врсте која одговара адреси странице и активирањем RAS сигнала. Затим се доводи адреса колоне и активира се CAS сигнал. На крају приступа CAS се гаси док RAS остаје активан. Код следећих приступа у оквиру странице, доводи се нова адреса колоне на адресним

улазима DRAM-а и потврђује се CAS сигнал. RAS сигнал је потребно деактивирати код приступа новој страници.

Побољшани странични метод приступа. За овај метод је потребан DRAM који ради у страничном начину рада. Овај метод приступа је сличан страничном јер имају исти почетак. Прво се доводи адреса врсте и активира RAS сигнал. Нови приступ може да почне чим се постави колона на адресним линијама DRAM-а, а не од тренутка када се активира CAS. CAS сигнал се користи се лечовање адресе колоне и дозволу рада излазним степенима.

Статичко колонски метод приступа. У овом случају је потребан статичко-колонски DRAM. Као и претходни метод, статичко колонски метод омогућава бржи приступ информацији у оквиру странице којој се претходно приступило. Али ипак је бржи од њега (од побољшаног страничног) јер се у току приступа читање CAS сигнал не мења. Приступ колони почиње одмах након постављања нове адресе на адресним линијама. Адреса колоне се не лечује али је активна у току приступа. RAS сигнал се потврђује само код првог приступа страници. [3,5]

Тактовани метод приступа. Захтева коришћење DRAM-а који користи тактовани начин рада. Овај метод омогућава бржи приступ за 4 редоследне локације по модулу, на основу једне предате адресе, док се остале генеришу у DRAM-у. Овако се ослобађа адресна магистрала у току приступа. Опадајућа CAS ивица иницира наредни приступ. RAS сигнал је активан само у току првог приступа док је за остале приступе активан само CAS сигнал. Како би се приступило наредне 4 локације, потребно је искључити RAS сигнал.

7.2 Освежавање DRAM-а

DRAM меморија се мора освежавати како би се сачувао интегритет података. Освежавање се обавља у току одређеног временског интервала, који је обично 4 ms. Ако се освежи врста DRAM-а, то онда значи да се освежавају све локације у тој врсти. Операција освежавања је иста, али ако се посматра ток операције и сигнали на линијама постоје одређене поделе. На основу метода освежавања деле се на:

- а) *освежавање са преплитањем и пакетно* (interleaved and burst refreshes)
- б) RAS – *само* (RAS – only refresh)
- в) CAS - *пре* – RAS (CAS – before RAS refresh)
- г) *скривено – освежавање* (hidden refresh)
- д) *освежавање – са – чишћењем* (refresh with scrubbing) [5]

Освежавање са преплитањем и пакетно. Код DRAM освежавања је могуће преплитање, са меморијским приступима читање и упис. У овом случају се континуално освежавају локације свих врста. Док се не освеже све врсте циклуси иду у континуитету. Док се не заврши освежавање CPU не може да приступи DRAM-у, па мора да чека дозволу приступа када се освежавање заврши.

RAS – само освежавање. Овај тип је најједноставнији тип освежавања и користи се код свих типова DRAM-а. Адреса врсте се доводи на адресне линије и затим се активира RAS сигнал. Све банке DRAM поља могуће је освежити на овај начин. На адресне улазе поставља се адреса врсте која се лечује RAS сигналом. Време освежавања свих ћелија у једној врсти, једнако је времену читања или уписа. У току освежавања CAS сигнал се не мења. Број циклуса освежавања је исти као број врста. Освежавање је неопходно извршити за неколико милсекунди. Код овог начина освежавања DMC генерише адресе врсте. Укупно време освежавања је $M * t_c$. M представља број врста DRAM-а, а t_c време читања.

CAS – пре – RAS освежавање. Код ове врсте приступа адреса за освежавање врсте се генерише унутар DRAM-а, а не генерише је DMC. Предности овог типа освежавања су то што је једноставан и штеди струју. Код ове врсте приступа неопходно је уградити адресни генератор врсте. Основни принцип ове методе се састоји у томе што се CAS доводи на ниско пре RAS. Крај циклуса је тренутак када RAS буде на ниско. Излаз бројача се активира постављањем улаза CAS улаза на ниско пре RAS.

Скривено освежавање се јавља одмах након операције читања. И за овај тип освежавања се врши уградња адресног генератора. Адресни генератор представља спој одређене логике за управљање и бројача. Бројач служи да показује која се врста наредна треба освежити. Како је одмах након читања CAS сигнал постављен на ниско, а RAS сигнал на високо, активира се освежавање. Опадајућом ивицом RAS-а врши се лечовање врсте.

Освежавање са чишћењем се користи код EDAC технике. Истовремено са освежавањем врши се детекција и корекција грешке. Када имамо 4 DRAM банке, освежавање се врши над одговарајућим врстама 3 банке, а над четвртом банком се врши R/M/W циклус. Овом техником се смањује појава једноструких soft грешака, и спречава се њихово нагомилавање. (Ако дође до нагомилавања, онда се ове грешке не могу кориговати). [5]

8.0 Типови DRAM меморије

8.1 Синхрона DRAM меморија

Један од најшире коришћених облика DRAM меморије је синхрона DRAM меморија (SDRAM). За разлику од традиционалне DRAM меморије која је асинхрона, SDRAM меморија размењује податке са процесором синхронизовано у односу на спољашњи сигнал генератор такта и радећи на пуној брзини процесора/меморије без наметања стања чекања.

У типичној DRAM меморији, процесор представља нивое сигнала адреса и управљања, показујући да скуп података на одређеној локацији у меморији треба да се прочита из DRAM меморије, или у њу упише. После кашњења због приступног времена, у DRAM меморију се уписују, или из ње читају подаци. За време кашњења због времена приступа, DRAM меморија изводи разне унутрашње функције, као што су активирање високе капацитивности линија реда и колоне, одређивање нивоа података и усмеравање података напоље кроз излазне бафере. Процесор просто мора да чека то кашњење, што успорава перформансу система.

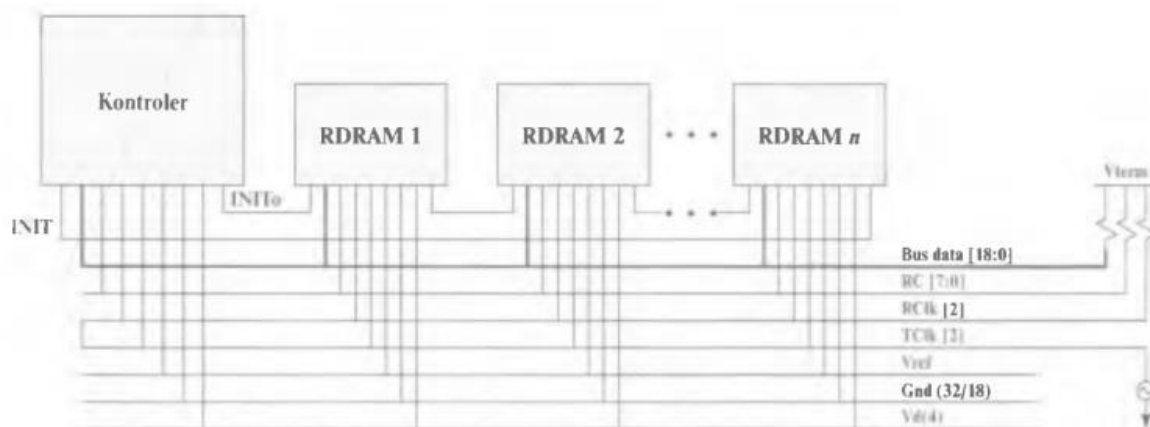
Код синхроног приступа, DRAM меморија помера податке унутра и напоље под контролом системског генератор такта. Процесор, или неки други господар магистрале, издаје информацију инструкције и адресе, коју прихвата DRAM меморија. DRAM меморија онда одговара после постављеног броја циклуса генератора такта. У међувремену, процесор може безбедно да обавља друге задатке, док SDRAM меморија обрађује захтев.

SDRAM латенција суштински није нижа (бржа) од асинхроног DRAM-а. Заиста, рани SDRAM је био нешто спорији него савремени рафал EDO DRAM због додатне логике. Предности SDRAM-овог интерног баферовања долази због његове способности да убаца операције на више банака меморије, чиме се повећава ефикасни пропусни опсег. [1,2,6]

8.2 Рамбус DRAM меморија

RDRAM меморију, коју је развио Рамбус, Интел је усвојио за своје процесоре Пентиум И Итаниум. Она је постала главни конкурент SDRAM меморији. Чипови RDRAM меморије су вертикална паковања, са свим пиновима на једној страни. Чип размењује податке са процесором преко 28 жица, не дужих од 12 центиметара. Магистрала може да адресира до 320 РДРАМ чипова а брзина преноса је 1.6 ГБ/с.

Специјална RDRAM магистрала испоручује адресне и управљачке информације користећи асинхрони протокол оријентисан на блокове. Оно што ту брзину чини могућом је сама магистрала, која веома прецизно дефинише импеданце, тактовање и сигнале. Уместо да се њом управља помоћу експлицитних сигнала RAS, CAS, R/W и CE који се користе у уобичајеним DRAM меморијама, RDRAM меморија добија меморијски захтев преко магистрале велике брзине. Тај захтев садржи жељену адресу, врсту операције и број бајтова у операцији.



Слика 13. Организација RDRAM меморије [1]

На слици 13 је распоред RDRAM меморије. Конфигурација се састоји од контролера и једног броја RDRAM модула који су међусобно повезани преко заједничке магистрале. Контролер је на једном крају конфигурације, а на супротном крају магистрале је паралелни завршетак линија магистрале. Магистрала укључује 18 линија за податке (16 линија стварних података и 2 за парност) које се тактују двоструком брзином генератора такта- то значи да се један бит шаље на предњој, а други на задњој ивици сигнала такта. То резултује брзином сигнала на свакој линији податка од 800 Mb/s. Постоји посебан скуп од 8 линија (RC) које се користе за адресне и управљачке сигнале. Постоји такође и сигнал такта који стартује са супротног краја од контролера, простире се до контролера и онда враћа назад. Модул RDRAM меморије шаље податке контролеру синхроно са генератором такта господара магистрале, а контролер шаље податке RDRAM меморији синхроно са сигналом такта

у супротном смеру. Преостале линије магистрале укључују референтни напон, уземљење и извор напајања електричном енергијом. [1,6]

8.3 Остали типови DRAM меморија

Постоји више типова чипова са динамичком RAM меморијом. Најстарији **FPM (Fast Page Mode) DRAM**, још увек постоји у најстаријим рачунарима. Он је интерно организован у облику матрице битова којој се приступа на основу адресе реда, а после и адресе колоне. Наменски сигнали изазивају рад меморије, тако да она није синхронизована са главним радним тактом.

FPM DRAM замењен је чипом **EDO (Extended Data Output) DRAM**, који омогућава приступ меморији иако претходни циклус читања или уписа није завршен. Оваква једноставна паралелна обрада не убрзава појединачно обраћање меморији, већ се шири њен пропусни опсег, чиме се омогућава читање или уписивање већег броја речи у секунди.

FPM и EDO чипови су прилично добро радили док је циклус меморијских чипова био мањи од 12 нс. Када су процесори постали толико брзи да им је стварно затребала брза меморија, FPM и EDO чипови су замењени **SDRAM (Synchronous DRAM)** чипом, који је био комбинација статичке и динамичке RAM меморије којим управља главни радни такт. Основна предност SDRAM меморије јесте то што радни такт уклања потребу за управљачким сигнаlima- процесор саопштава меморији који број циклуса радног такта треба да изврши. На тај начин, меморија у сваком циклусу испоручује 4, 8 или 16 битова, у зависности од броја излаза.

Након SDRAM меморије дошло је до проналаска **DDR (Double Data Rate)** SDRAM меморија. Она испоручује податке и при узлазној и при силазној ивици сигнала радног такта, чиме се проток података дуплира. Тако је DDR чип ширине 8 битова који ради на учесталости од 200 MHz испоручује две осмобитне вредности 200 милиона пута у секунди, што теоријски даје проток података од 3.2 gb/s. [1,2,6]

9.0 Тржиште DRAM меморије

Када посматрамо тржиште динамичке рам меморије, морамо узети у обзир чињеницу да се произвођачи константно труде да побољшају своје производе, повећају перформансе и смање латенцију, како би придобили што већи део тржишта и самим тим повећали зараду. Динамичка рам меморија је јефтинија за производњу од статичке, па се због тога уграђује и у мобилне телефоне, што отвара нове могућности у смислу тржишта.

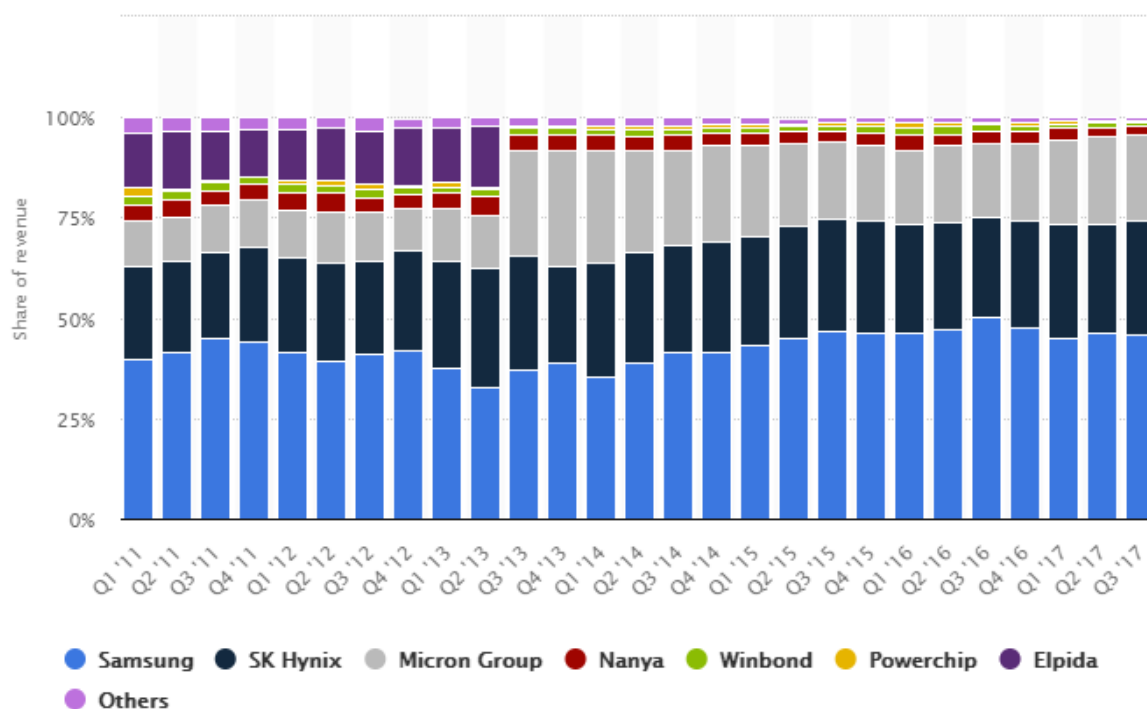
Тржиште DRAM-а можемо да посматрамо и поделимо на два критеријума. Можемо посматрати тржиште гледајући примену и регион. Под применом се мисли на то да ли ће се произведена меморија користити у мобилним уређајима, рачунарима, серверима или ће имати неку специјалну намену. Ако погледамо још дубље, можемо поделити мобилни сегмент на таблете и мобилне телефоне. Меморију која се производи за рачунаре можемо поделити на ону за десктоп и лаптоп рачунаре. Под меморијом која служи за специјалну намену мисли се на меморије које се производе за системе који се уграђују у аутомобиле, графичке картице, смарт телевизоре, гејминг конзоле (XBOX и PlayStation) и друге. Свака подела представља потенцијално тржиште за произвођаче и прилику да се прошири производња. Ако посматрамо географски тржиште DRAM-а се може поделити на Северну Америку, Јужну Америку, Западну Европу, Источну Европу, Блиски исток и Африку, источну Азију без Јапана и Јапан.

Ако се фокусирамо на регионе, може се уочити пораст тржишта динамичке рам меморије због повећане популарности мобилних телефона у Индији, Кини и Бразилу. Предвиђа се да ће то тржиште расти и у наредном периоду (до 2025. године). Источна Азија је била а и даље је регион са најјачим тржиштем динамичких рам меморија а прати је Северна Америка. У предстојећем периоду предвиђа се да ће највећи пораст тржишта имати Блиски исток и Африка. Разлог за то је чињеница да у тим местима слабо заступљено тржиште, што имплицира да ту има највише места за развој.

Са развојем мобилних телефона који имају релативно ниску цену а добре перформансе, расте и потражња за меморијом која има исте такве карактеристике. Данас је број компанија које производе мобилне уређаје највећи у земљама Азије (Индија, Кина, Индонезија). Управо то ће помоћи расту и развоју DRAM тржишта. Још једна ствар која иде у прилог развоју је и та што долази до појаве ултра танких телефона ноутбука и сличних уређаја чија се брзина рада у великој мери ослања на динамичку рам меморију. [7]

Највећи светски произвођачи динамичке рам меморије су Samsung, SK Nuph, Micron group, Nanya, Windbond, Powerchip, Elpida и други. Фокус ових компанија је на производњи меморија које нису скупе како би њихови производи били што приступачнији великом броју купаца. Како би стекли што већу предност у односу на

конкуренцију, компаније сарађују са највећим светским произвођачима мобилних телефона, правећи меморију специфично за њих. [7]



Дијаграм 1 Квартални део глобалног тржишта DRAM-а, за водеће произвођаче, [8]

На графику је приказан квартални део глобалног тржишта DRAM-а, за водеће произвођаче, у периоду од 2010. до 2017. године. Као што се може видети, највећи део тржишта заузима компанија Самсунг, која током свих ових година успела да задржи титулу највећег произвођача. Други по величини произвођач је СК Хинс. Заједно, ове две компаније држе скоро 75 посто тржишта, наравно узимамо у обзир да Самсунг компанија има скоро 50 посто тржишта. Самсунг је до последње четвртине 2014. имао осцилација, али након 2015, када је по трећи пут освојио ЦЕС-ову награду за иновацију, придобио је половину укупног тржишта. Микрон група има значајан удео од последњег квартала 2013. године. Заједно ове 3 компаније имају од 2013. скоро 90 посто тржишта у свом поседу, док је осталих 10 посто распоређено осталим произвођачима. [8]

9.1 Компанија Самсунг на тржишту DRAM-а

Самсунг, једна од највећих компанија, не само по производњи динамичких рам меморија, је главни предводник у иновацији и побољшању квалитета и перформанси меморијских чипова.

У 2009. години Самсунг је представио тада нове чипове капацитета 4 гигабајта. Ови нови чипови су били у могућности да раде на 1,35 V што је било ниже у односу на дотадашњих 1,5 V (референтна вредност). Примарна примена ових чипова је била код серверских рачунара, свакодневних десктоп и лаптоп рачунара. Ови чипови су подржавали dual-die package технологију, што значи да је било могуће саставити два 16 Gb модула са 4 Gb чиповима како би се добио пакет од 32 Gb. Самсунгов нов тип меморије је био и ефикаснији од претходних чипова. Нови модули од 16 Gb су трошили око 40 посто мање енергије у односу на претходну генерацију. Највећа брзина је била 1.6 Gb/s.

Исте године, Самсунг је са компанијом Хинкс, је промовисао нов стандард на пољу меморије који је имао назив Serial Port Memory Technology (SPMT). Главна намена ове нове технологије је била усмерена ка мобилном тржишту, пре свега ка мобилним уређајима високих перформанси, које је тих година било у знатном порасту. SPMT ће омогућити већу пропусну моћ, мањи број пинова, мању потрошњу, и био би јефтинији. Број пинова је био за 40 посто мањи него раније. У зависности од конфигурације, проток би требало да буде од 200 Mb/s до 12.6 Gb/s у зависности од подешавања. [9, 10]

Током 2014. године Самсунг је почео да производи своју 20 nm 6 Gb LPDDR3 меморију за мобилне уређаје. Ова меморија је имала велику густину, што је омогућавало да спакује и до 6 Gb. Један од фокуса је био и побољшавање такозваног мултитаскинга, тј. вршења више задатака истовремено, и рада апликација на екранима великих резолуција. Брзина ове меморије је била за 20 посто већа од претходне LPDDR2. Употреба батерије је смањена за 10 посто у односу на претходну генерацију. Захваљујући бољој компактности, ове меморије су биле мање.

Следеће године, Самсунг је кренуо са производњом DRAM меморије која је била намењена за произвођаче графичких картица, гејминг конзола, суперкомпјутера и других уређаја (GDDR5). Ова меморија је израђивана у 20 nm процесу производње, па је због тога и бржа од свог претходника. Са овим чипом Самсунг је у понуди имао чипове од 4 Gb и 6 Gb. Међутим, када упоредимо ова два чипа, меморија од 6 Gb има већу брзину преноса од 8 Gb/s (чип од 4 Gb је имао брзину од 7 Gb/s). Такође, нова меморија је користила око 25 посто мање струје од претходне GDDR3 меморије.

Пред крај 2015. године Самсунг је најавио да планира да у наредној години започне производњу 18 nm DRAM чипова. Овај потез је оправдаван, тиме што је намеравано да се смањи цена производње и да се самим тим повећа профит.

У 2016. години компанија почиње производњу нових 10 nm DDR4 DRAM-а. Ти чипови су произвођени као 8 Gb чипови. У том тренутку на тржишту није било доста произвођача DDR4 меморије, па је Самсунг овим потезом омогућио бржу транзицију на DDR4 меморију. Технички проблеми при изради меморије су решени коришћењем ArF (argon fluorid) литографије. Брзина трансфера је повећана на 3200 Mb/s, што је било значајно побољшање у односу на 20 nm DDR4 меморију. Ефикасност је побољшана за 20 посто, тако што је смањена волтажа на 1,2 V.

Исте године самсунг је најавио свој најновији LPDDR4 6 GB DRAM чип произведен помоћу 10 nm процесе, коришћен за мобилне уређаје високих перформанси. Предности овог чипа су већа енергетска ефикасност. Самим тим и рок трајања батерије се повећава. Ово је посебно значајно јер су се на тржишту појављивали мобилни уређаји са високим перформансама, а те перформансе су се морале ускладити са радом батерија. Овај чип је касније примењен у изради Самсунговог телефона Note VI.

У овом периоду постајала је све популарнија продаја такозваних NAND флеш меморија. Ове меморије су постајале све популарније због тога што нису захтевале струју како би сачувале податке. Циљ меморије је био да се смањи цена и повећа капацитет. Због тога је самсунг смањио улагања у производњу DRAM меморије како би се фокусирао на производњу NAND меморије.

Након 2014. када је самсунг почео да производи LPDDR3 меморију за мобилне уређаје, компаније је сада најавила производњу нове LPDDR4 меморије од 8 GB. Ови DRAM пакети раде на брзини од 4266 Mb/s. Импресивно је то што је ова брзина скоро два пута бржа од меморије која се користи на десктоп рачунарима. Густина израде је била већа па је поново смањена величина чипа. Функција ових чипова је да пренесе неке могућности које имају десктоп рачунари на мобилне уређаје високих перформанси, као што је на пример 4 K видео резолуција. Једна ствар коју Самсунг није могао да унапреди у односу на LPDDR4 меморију која је произведена уз 20 nm процесе је потрошња, која је била скоро иста. [9,10]

Прошле године Самсунг је одлучио да повећа производњу DRAM меморије отварањем нових фабрика. Сама чињеница да су планиране 2 нове фабрике говори о томе колико се стање побољшало на пољу DRAM меморије. Овај потез ће утицати на целокупно тржиште. Претпоставља се да ће цена меморија променити у другој половини 2018. године. Како поред персоналних рачунара, и мобилни уређаји чине велики део тржишта DRAM-а може се очекивати и промена цена мобилних уређаја. Током 2017. године количина DRAM-а је била релативно ниска па је због тога и цена била у порасту.

Пред крај 2017. самсунг је обелоданио да је почео да производи 8 GB DDR4 DRAM-а процеса 10 nm. Ова генерација меморије има највећу ефикасност и најмање димензије у поређењу са свим конкурентима. Побољшања се огледају у продуктивности која је већа за око 25 посто. Потрошња је мања за око 10 процената.

Радна фреквенција је повећана на 3600 Mb/s. Самсунг је у оквиру компаније развио нове технологије које ће даље помоћи у развоју меморијских чипова следеће генерације (DDR5, LPDDR5, GDDR6). [9,10]

Закључак

У овом раду, смо сазнали како функционише динамичка меморија са случајним приступом (Dynamic Random Access Memory – DRAM). Улога коју врши ова меморија је изузетно важна за рад рачунара. Меморија чува сваки бит података у посебном кондензатору у оквиру интегрисаног кола. Кондензатор може бити пун или празан; ова два стања представљају две вредности бита, уобичајено се зову 0 и 1. Пошто кондензаторима цури набој, информација би на крају нестала ако се не би кондензатор периодично освежавао. Због ових захтева за освежавање, динамичка меморија је супротна у односу на SRAM и друге статичке меморије.

Када говоримо о предностима динамичке у односу на статичку, на првом месту треба поменути једноставну структуру. По биту су потребни један кондензатор и један транзистор, што омогућава велику густину. Примена динамичке рам меморије је изузетно широка. Од десктоп и лаптоп рачунара, конзола, графичких картица мобилних уређаја и других специјализованих примена.

Динамичка рам меморија нема брзину да испрати процесор у смислу извршавања задатака тј. пружања конкретног податка у временским периодима у којим сам процесор то захтева. То доводи до појаве кашњења. Како се не би губило време постоје процеси којима меморија „вара” процесор и избегава да доведе процесор у стање апсолутног чекања. Још једна мана је појава грешака код динамичке рам меморије. Међутим различитим методама детекције и корекције, већина грешака бива спречена, или ако дође до појаве грешке коректована.

Тржиште динамичке рам меморије се мења из године у годину у зависности од потражње. Како последњих година на тржиште доспевају мобилни уређаји високих перформанси, потреба за меморијом која може да испрати те перформансе је све већа. Услед појаве NAND флеш меморија, тржиште динамичке рам меморије доживело је мали пад али упркос томе се опоравља, услед велике потражње за меморијама најновијих генерација (DDR4, GDDR5, LPDDR4). Како време пролази, меморије бивају све брже и произвођачи покушавају да произведу меморију која ће имати брзину што ближе брзини процесора. Највећи светски произвођачи теже да њихове меморије буду што брже али и што јефтиније за израду како би биле доступне што већем броју купаца, а наравно и како би што више повећале профит.

Литература

- [1] William Stallings, Организација и архитектура рачунара: Пројекат у функцији перформанси: Превод седмог издања. Приступ: 04.04.2018.
- [2] Andrew S. Tanenbaum: Архитектура и организација рачунара: Превод петог издања. Приступ: 06.04.2018.
- [3] Миле К. Стојчев: RISC, CISC i DSP procesori: Електронски факултет у Нишу 1997. Приступ: 09.04.2018.
- [4] Настава, Природно математички факултет: Универзитет у Нишу, Одсек за математику и информатику, Рачунарске системе, Главна меморија.
<http://tesla.pmf.ni.ac.rs/Predavanja/racunar%20sist/Knjiga%20-%20Vol2/GLAVA3.pdf>
Приступ: 15.04.2018.
- [5]] Micron: Various methods of DRAM refresh: TN-04-30. Приступ: 21.04.2018.
- [6] https://www.webopedia.com/quick_ref/dram_memory.asp Приступ: 23.04.2018.
- [7] <https://www.futuremarketinsights.com/reports/dynamic-random-access-memory-dram-market> Приступ: 28.04.2018.
- [8] <https://www.statista.com/statistics/271726/global-market-share-held-by-dram-chip-vendors-since-2010/> Приступ: 04.05.2018.
- [9] <https://www.benchmark.rs/pretraga> Приступ: 08.05.2018.
- [10] <http://www.samsung.com/semiconductor/dram/> Приступ: 12.05.2018.
- [11] <https://informatikasg.wordpress.com/category/vrste-memorije-racunara/>
Приступ: 17.05.2018.