

Универзитет у Београду
Електротехнички факултет

Игор М. Икодиновић

**Методологија
за анализу перформанси DRAM меморија
са тачношћу на нивоу циклуса**

Докторска дисертација

Београд, 2016

University of Belgrade
School of Electrical Engineering

Igor M. Ikodinovic

**Methodology
for Cycle-accurate Performance Analysis
of DRAM Memories**

Doctoral Dissertation

Belgrade, 2016

Комисија за преглед и оцену дисертације

Ментор:

др Вељко Милутиновић
редовни професор, Универзитет у Београду, Електротехнички факултет

Чланови комисије:

др Мило Томашевић
редовни професор, Универзитет у Београду, Електротехнички факултет

др Миодраг Живковић
редовни професор, Универзитет у Београду, Математички факултет

Датум одбране:

Захвалнице

Захваљујем се својим колегама са којима сам низ година на различитим пројектима радио на пословима у области дизајна, архитектуре и анализе перформанси рачунарских система у компанијама ATI, AMD и Qualcomm на предусретљивости, отворености у размени мишљења и спремности да помогну када год је то било потребно.

Захваљујем се компанији HDL Design House, што ми је указала поверење и прихватила предлог за пројекат комерцијалне реализације верификационе компоненте за анализу перформанси DRAM меморија, који се базира на овој дисертацији.

Захвалност дугујем и свим својим колегама са Електротехничког факултета Универзитета у Београду за искрену и пријатељску помоћ и подршку која траје од наших првих сусрета до данас.

Посебну захвалност дугујем свом ментору др Вељку Милутиновићу, са којим сам имао задовољство да сарађујем још од својих студентских дана, а касније и као асистент и постдипломац на Електротехничком факултету у Београду. Без његове помоћи и несебичне подршке свих ових година, дисертација би много теже угледала светлост дана.

Срдачно се захваљујем и комисији која је прегледала овај рад и својим корисним предлозима утицала на коначну верзију дисертације.

Хвала мојим родитељима, супрузи и деци на љубави, подршци и разумевању које су ми пружили свих ових година.

Слава и хвала Господу нашем за љубав и снагу коју нам даје да истрајемо, упркос свим изазовима, на своју добробит и на добробит свих око нас.

Резиме

Методологија за анализу перформанси DRAM меморија са тачношћу на нивоу циклуса

Главна меморија је једна од кључних компоненти сваког рачунарског система. У савременим рачунарским системима главна меморија се најчешће имплементира помоћу меморија типа DRAM. Она непосредно утиче на цену, потрошњу енергије и ефикасност система, а посредно и на његову интерну архитектуру и организацију. Из тих разлога се анализи перформанси DRAM меморије посвећује велика пажња приликом пројектовања.

Код мерења и анализе перформанси DRAM меморија тачност је од суштинског значаја. Нетачност смањује поузданост резултата и закључака, што може довести до доношења погрешних одлука приликом пројектовања, а тиме и до значајног губитка времена, труда и новца. Општи трендови у домену технологије израде, архитектуре и организације рачунарских система и развоја системског софтвера воде сталном расту оптерећења главне меморије и њеној све већој виртуелизацији, чиме се проблем тачности додатно погоршава. С обзиром да се не види технологија која би у догледно време могла да замени DRAM, може се очекивати да ће значај овог проблема у будућности само добијати на тежини.

Кључни проблем у анализи перформанси DRAM меморија је немогућност да се непосредно утврди да ли су поједини циклуси на меморијској магистрали слободни или заузети. То ствара проблем са мерењем чак и основних показатеља меморијских перформанси, као што су степен искоришћења и ефикасност. Због тога није јасно како одговорити на фундаментално питање: „Како мерити перформансе DRAM меморије са потребним нивоом тачности?“. Чак и за показатеље перформанси које је могуће тачно и непосредно мерити, као што је

проток података, постоји проблем интерпретације измерених вредности. Теоријски максимуми показатеља меморијских перформанси се континуирано мењају, а не могу се непосредно мерити, па није могуће интерпретирати резултате поређењем измерених и максималних вредности. Стога није јасно како одговорити ни на следећа кључна питања: „Колико су измерене перформансе DRAM меморије добре или лоше?“ и „Како поредити перформансе DRAM меморије измерене у различитим периодима и за различита радна оптерећења?“.

Основни циљ дисертације је проналажење начина да се превазиђу ови суштински проблеми у вези са мерењем и анализом перформанси DRAM меморија. Као резултат рада на тој проблематици створена је нова теоријска основа за мерење и анализу меморијских перформанси и формулисана је одговарајућа методологија која дефинише како да се мерење и анализа спроводе у пракси. Методологија се заснива на тачној и једнозначној карактеризацији меморијских циклуса на слободне (неискоришћени циклуси), активне (заузети циклуси чије стање је опсервабилно) и режијске (заузети циклуси чије стање заузетости није опсервабилно). Најважније компоненте предложене методологије су:

- *Функционални и временски модел DRAM меморије* који DRAM меморију апстрахује у виду генеричког уређаја чији рад се може потпуно описати помоћу параметризованог коначног аутомата и анализирати са жељеним нивоом тачности
- *Модел за мерење и анализу перформанси DRAM меморија* који омогућава прецизну карактеризацију меморијских циклуса
- *Метрика за мерење и анализу перформанси DRAM меморија* која представља нов теоријски основ за рад у овој области
- *Метод за процену максимума перформанси DRAM меморија* који омогућава решавање проблема интерпретације измерених резултата

Методологија специфицира како се наведене компоненте дефинишу, конструишу и параметризују у складу са имплементацијом DRAM меморије и описује све релевантне поступке и процесе који омогућавају мерење и анализу перформанси DRAM меморија у реалном окружењу.

Предложена методологија доноси суштински напредак у односу на постојећа решења. Њене најважније предности са теоријског становишта су:

- максимална тачност
- могућност прецизне процене теоријског максимума перформанси
- могућност идентификације главних узрока субоптималног рада
- комплетност (могућност анализе свих типова DRAM трансакција)

Са становишта практичне примене, најважније предности су:

- независност од архитектуре система на коме се генерише радно оптерећење
- независност од величине радног оптерећења
- преносивост (могућност имплементације на различитим платформама)
- ефикасност (брзо генерисање резултата уз мали напор корисника)
- ниска цена и комплексност имплементације и верификације

Предложена методологија може да замени постојећа решења у свим доменима где је потребно побољшати тачност и ефикасност анализе. Уједно, омогућава се и примена у потпуно новим областима, попут анализе критичних сценарија (анализа секвенци које се јављају спорадично, али имају битан ефекат на рад система), трансакционе анализе система (анализа рада система праћењем појединачних трансакција), компаративне анализе (поређење резултата са различитих плаформи, за различита радна оптерећења или у различитим периодима) и др. Методологија омогућава релативно једноставну примену на инжењерском нивоу уз мали утрошак ресурса и тиме обезбеђује висок ниво ефикасности и употребљивост у пракси.

Као резултат систематизације на чврстим теоријским основама и решавања кључних проблема који су раније постојали у вези са мерењем и анализом перформанси DRAM меморија, направљен је суштински помак у овој области. Тај напредак омогућава прелазак анализе перформанси DRAM меморија из домена инжењерске вештине у домен научно-стручне дисциплине и подизање процеса анализе рада целокупног рачунарског система на квантитативно и квалитативно виши ниво.

Кључне речи: DRAM, перформансе, методологија, анализа, моделовање

Научна област: Техничке науке – Електротехника и рачунарство

Ужа научна област: Архитектура и организација рачунарских система

УДК: 621.3

Abstract

Methodology for Cycle-accurate Performance Analysis of DRAM Memories

Main memory is one of the key components in a computer system. In modern systems, main memory is almost always implemented using DRAM type of memory. Memory has a direct impact on price, power consumption and performance of the system, and an indirect impact on its internal architecture and organization. That is why a lot of attention is paid to DRAM performance analysis during system development.

Accuracy is of utmost importance in measurement and analysis of DRAM performance. Inaccuracy reduces reliability of the results and conclusions, which can lead to wrong architectural or design decisions, and a significant loss of time, effort and money. General trends in manufacturing technology, computer architecture and organization, and system software development lead toward its increasing virtualization and greater utilization, which exacerbates the problem of accuracy. Considering that there is no technology in sight that could replace DRAM in the near future, the importance of this problem will only increase.

The main problem in analysis of DRAM performance is inability to determine which cycles on the memory bus are busy or idle. That makes even basic memory performance parameters, such as utilization or efficiency, difficult to measure. In essence, it is not clear how to answer the fundamental question: “How can DRAM performance be measured with the required level of accuracy?”. Even for performance parameters that can be measured based on the observable signals, such as data bandwidth, there is a problem of interpretation of the measured results. Theoretical maximums of performance parameters continually fluctuate, and they cannot be directly measured, so it is not possible to interpret them by comparing measured and maximum values. It is thus not

clear how to answer the following key questions either: “How good or bad are measured performance results?” and “How can results measured in two different time periods or for different workloads be compared?”.

The main goal of the dissertation was to overcome these fundamental problems. As a result, a new theoretical foundation for DRAM performance measurement and analysis was created, along with a methodology that specifies how to conduct the measurement and analysis in practice. The methodology is based on accurate characterization of memory cycles as idle (cycles not used), active (cycles used, and their state is observable), or overhead (cycle that cannot be used due to DRAM protocol constraints, and their state is not observable). The most important components of the proposed methodology are:

- *Functional and timing model of DRAM memory* that abstracts DRAM memory as a generic device in a form of a state machine parameterized by DRAM device configuration and timing parameters, whose operation can be analyzed with the desired level of accuracy
- *DRAM measurement and performance analysis model* that enables accurate performance characterization of memory cycles
- *DRAM performance metric* that represents a new theoretical foundation for DRAM performance measurement and analysis
- *Method for estimating DRAM performance maximum* that enables solving of the problem of interpretation of results

The methodology specifies how these components are defined, constructed, and parameterized according to a particular DRAM implementation and describes all the relevant processes and procedures that enable measurement and analysis of DRAM performance in real systems.

The proposed methodology makes fundamental advancements over the existing solutions. Its most important advantages, from the theoretical point of view, include:

- guaranteed maximum accuracy
- enabling accurate estimation of theoretical maximum
- enabling root-causing of sub-optimal DRAM performance

- completeness (takes into account all DRAM commands and timing parameters)

The most important advantages from the practical point of view include:

- system agnostic (does not depend on the system that generates workload)
- workload agnostic (does not depend on the size or type of workload)
- portability (can be implemented on any type of system)
- efficiency (generates results fast and with little user effort)
- low implementation and verification complexity and cost

The proposed methodology can replace existing solutions in all domains where accuracy and efficiency are of importance. At the same time, it can be applied in completely new domains, such as analysis of critical scenarios (analysis of sequences that occur sporadically, but have tangible impact on performance), transactional analysis (analysis of system operation by following individual transactions), comparative analysis (comparing results from different platforms, for different workloads, or in different time periods), etc. The methodology enables relatively simple application at the engineering level, with small use of resources, and high level of efficiency.

As a result of systematization on firm theoretical grounds, all key problems in the domain of DRAM performance measurement and analysis were solved. The fundamental improvement made in this domain allows DRAM performance measurement and analysis to be elevated from an engineering art to a scientific method, which enables a quantitative and qualitative leap in computer system analysis.

Keywords: DRAM, performance, methodology, analysis, modeling

Scientific field: Technical sciences – Electrical and computer engineering

Specific scientific field: Computer architecture and organization

UDC: 621.3

Садржај

Захвалнице.....	IV
Резиме.....	V
Abstract	IX
Садржај.....	XII
Предговор.....	XV
Табеле.....	XVIII
Слике и графикони	XIX
Скраћенице	XXI
1 Увод	1
1.1 Врсте меморија	2
1.2 Меморије у рачунарском систему.....	7
1.2.1 Меморијска хијерархија	7
1.2.2 Локација делова меморијске хијерархије у рачунарском систему.....	9
1.2.3 Карактеристике рачунарских меморија	11
1.2.4 Примена рачунарских меморија	13
1.3 Место и значај DRAM меморија	16
1.4 Анализа перформанси DRAM меморија	20
1.4.1 Значај анализе перформанси DRAM меморија	20
1.4.2 Тачност код анализе перформанси DRAM меморија	22
1.4.3 Утицај развојних трендова на анализу перформанси DRAM меморија	23
1.5 Проблеми у вези са анализом перформанси DRAM меморија	24
1.5.1 Фундаментални проблеми.....	24
1.5.2 Применљивост на командну магистралу и магистралу података.....	26
1.5.3 Примена на различита окружења, моделе система и радна оптерећења.....	27
1.5.4 Цена, комплексност и ефикасност у примени	27
1.5.5 Неадекватност постојећих методологија	27

1.6	Предлог решења.....	28
1.7	Терминологија	30
2	DRAM меморија	33
2.1	DRAM меморија у рачунарским уређајима	33
2.2	Структура DRAM меморије.....	36
2.2.1	Меморијска ћелија	36
2.2.2	Меморијска матрица	39
2.2.3	Аналогна кола за приступ меморији	43
2.2.4	Интегрисано коло.....	47
2.2.5	Меморијски чип.....	49
2.2.6	Меморијски модул	52
2.3	Проточна обрада код DRAM меморије	57
2.4	DRAM архитектура	59
2.5	DRAM стандарди.....	63
2.6	DRAM протокол	65
2.7	Организација меморијског система	70
3	Поређење са постојећим решењима	74
3.1	Методологије за мерење и анализу перформанси DRAM меморија	74
3.1.1	Методологије базиране на аналитичком моделовању перформанси	75
3.1.2	Методологије базиране на примени методе симулације	76
3.1.3	Предложена методологија.....	78
3.2	Поређење методологија за мерење и анализу перформанси DRAM меморија	78
4	Моделовање перформанси DRAM меморија.....	83
4.1	Ограничења у DRAM проточној обради	84
4.2	Прелази између DRAM команди.....	85
4.3	Коначни аутомат прелаза између DRAM команди	87
4.4	Карактеризација меморијских циклуса	90
5	Метрика	96
5.1	Метрика за мерење перформанси меморија.....	97
5.2	Теоријски максимум перформанси.....	102
5.2.1	Горња граница перформанси	102
5.2.2	Процена максимума перформанси интерполацијом.....	107
6	Методологија	111
6.1	Ток рада	111

6.2	Снимање DRAM трансакција	113
6.3	Парсирање DRAM трансакција	115
6.4	Одређивање почетног стања	116
6.5	Карактеризација меморијских циклуса	117
6.6	Генерисање резултата	118
6.7	Постпроцесирање	119
6.8	Визуелизација	120
6.9	Анализа	122
7	Примена	126
7.1	Системска анализа	126
7.1.1	Корелација догађаја у систему	128
7.1.2	Компаративна анализа	128
7.2	Архитектурална анализа меморијског контролера	128
7.3	Дебаговање проблема у вези са перформансама	129
7.3.1	Анализа критичних сценарија	129
7.3.2	Трансакциона анализа	129
7.4	Карактеризација радног оптерећења	130
7.5	Анализа потрошње енергије	130
7.6	DRAM верификација	132
7.6.1	Функционална верификација	132
7.6.2	Верификација перформанси	133
7.7	Примена на 3D DRAM меморије	134
7.7.1	Wide I/O меморије	135
7.7.2	HBM меморије	137
7.7.3	HMC меморије	139
8	Закључак	141
	Прилог А – Еквивалентност ante factum и post factum ограничења	144
	Прилог Б – Временски сигнали прелаза између DDR2 команди	146
	Литература	153
	Биографија	158

Предговор

Докторска дисертација припада области архитектуре и организације рачунарских система, односно делу ове области који се бави анализом рада рачунарских система са аспекта перформанси. Тема дисертације је инспирисана вишегодишњим ангажманом на пројектима у овом домену и непосредним увидом у проблеме са којима се стручни тимови сусрећу у пракси. Показало се да је један од већих проблема у анализи рада рачунарских система, како са теоријског, тако и са практичног становишта, недостатак тачности приликом мерења и анализе перформанси DRAM меморија. Решавање тог проблема је постао неопходан услов да би се рад у области анализе рада рачунарских система подигао на виши ниво.

Непосредна мотивација за почетак рада на овој проблематици налази се у једном интересантном примеру из праксе, који уједно сведочи о врсти и размерама проблема који могу да настану услед нетачне анализе перформанси DRAM меморија. Радећи за познатог произвођача интегрисаних графичких кола, приликом пројектовања система са идентичним графичким језгром за два различита клијента, водећа светска произвођача процесора за персоналне рачунаре, утврђено је да је један од њих имао слабије графичке перформансе од другог. Разлог за диспаратитет је била нижа ефикасност DRAM меморије на једној од платформи због различитих адресних образаца генерисаних од стране клијентских процесора. Приликом анализе није се могао утврдити тачан узрок проблема јер је тачност са којом су биле моделоване перформансе DRAM меморије пројектованих система била недовољна. Немогућност да се благовремено открије и отклони узрок проблема је довела до пласмана производа који није био довољно конкурентан, што је резултирало негативним маркетиншким и финансијским ефектима по клијента.

Пожељно је да се перформансе DRAM меморије у различитим окружењима и на различитим моделима рачунарског система мере на методолошки исти начин

како би резултати из различитих фаза развоја, са различитих платформи, за различита радна оптерећења или у различитим периодима могли међусобно да се пореде. Постојеће методологије углавном не могу да задовоље ове захтеве, јер нису лако, или уопште, применљиве у свим ситуацијама. Поред тога, често се не може постићи ни ефикасност и једноставност у примени коју изискује рад на сложеним пројектима. То је био додатни мотив за покушај да се формулише нова методологија за мерење перформанси DRAM меморија којом би се то омогућило.

Рад на изради дисертације захтевао је примену софистицираних софтверских и хардверских алата за пројектовање, имплементацију, верификацију, тестирање, аквизицију података и анализу рада VLSI система. Примена ових алата подразумева и постојање одговарајућег комплексног хардверско-софтверског окружења у коме се радно оптерећење може извршавати на циљној платформи још у фази развоја. Због високе цене и комплексности, оваква окружења и алати су доступни само већим компанијама и истраживачким центрима који се баве пројектовањем интегрисаних кола. У том смислу, сарадња са њима је била истовремено и неопходна и обострано корисна.

Методологија развијена у оквиру дисертације је практично примењена у оквиру развојног пројекта компаније HDL Design House, који је резултирао израдом верификационе IP компоненте за анализу перформанси DRAM меморија. Компонента је у процесу интеграције у комерцијалне IP производе за верификацију DRAM меморија једне од водећих светских компанија у области EDA.

Текст дисертације је писан ћирилицом, а за већину стручних израза за које је то било могуће су коришћене речи нашег језика. На тај начин је дат допринос очувању нашег писма и језика у области у којој доминирају латинично писмо и енглески језик. Ради лакшег праћења и повезивања са терминологијом присутном у иностраној стручној литератури, приликом првог помињања неког израза иза њега је увек наведен и одговарајући термин на енглеском језику. Осим тога, у овој области је присутан велики број скраћеница и акронима, који углавном потичу од речи енглеског језика. Ради лакшег праћења, на почетку дисертације је дат списак свих коришћених скраћеница са одговарајућим значењем.

Дисертација се састоји из осам поглавља. Уводно поглавље даје основне информације о месту и улози DRAM меморија у савременим рачунарским системима и објашњава значај проблематике тачног мерења и анализе њених перформанси. Затим се описују кључни проблеми који постоје у овој области и укратко излаже како их предложена методологија решава. У другом поглављу се детаљно описује DRAM меморија са становишта архитектуре, организације, дизајна и технологије израде и представљају кључни концепти од значаја за анализу перформанси DRAM меморија. У трећем поглављу се даје преглед постојећих решења из домена анализе перформанси DRAM меморија и пореде њихове особине са предложеном методологијом према изабраним критеријумима. Четврто поглавље је посвећено ACDC моделу за мерење перформанси DRAM меморија, у оквиру ког се описује начин његовог креирања и излаже алгоритам за карактеризацију меморијских циклуса који се на том моделу заснива. У петом поглављу се дефинише нова метрика за мерење перформанси DRAM меморија као теоријски основ на коме се базира предложена методологија. У наставку се излаже и метод за тачну процену максимума перформанси DRAM меморија, као кључне компоненте за решавања проблема интерпретације резултата. У шестом поглављу се описује целокупан процес рада у оквиру предложене методологије и објашњава како се анализа спроводи у пракси. Седмо поглавље се бави коришћењем предложене методологије у различитим областима примене, укључујући и коришћење у потпуно новим доменима. У осмом поглављу се разматрају нове меморијске технологије и дају предлози за даље правце истраживања у овој области. На крају дисертације дат је закључак у оквиру ког се износи значај и допринос тезе.

Игор Икодиновић

Београд, 2016

Табеле

Табела 1.1 – Врсте меморија	3
Табела 1.2 – Цена DRAM меморије у оквиру рачунарског система	17
Табела 2.1 – Параметри комерцијалних DRAM чипова	49
Табела 2.2 – Параметри DRAM меморија имплементираних према JEDEC стандардима	64
Табела 2.3 – Кључне команде DRAM протокола	66
Табела 2.4 – Есенцијални временски параметри	68
Табела 2.5 – Изведени временски параметри	69
Табела 3.1 – Поређење са постојећим методологијама	81
Табела 4.1 – DCT FSM за DDR2 SDRAM	92

Слике и графикони

Слика 1.1 – Архитектуре рачунарских система	2
Слика 1.2 – Ћелије непостојаних полупроводничких меморија.....	4
Слика 1.3 – Меморијска хијерархија.....	8
Слика 1.4 – Локација делова меморијске хијерархије у рачунарском систему	10
Слика 1.5 – Карактеристике меморија и њихова улога у рачунарском систему	14
Слика 1.6 – Карактеризација циклуса на меморијској магистрали	25
Слика 2.1 – DRAM меморија у рачунарским уређајима	34
Слика 2.2 – DRAM ћелија.....	38
Слика 2.3 – DRAM меморијска матрица.....	39
Слика 2.4 – Прерасподела наелектрисања код читања из DRAM меморијске ћелије.....	41
Слика 2.5 – Приступ DRAM ћелијама.....	46
Слика 2.6 – DRAM интегрисано коло	47
Слика 2.7 – DRAM чип	50
Слика 2.8 – FBGA паковање	51
Слика 2.9 – SODIMM модул	53
Слика 2.10 – ECC RDIMM модул	54
Слика 2.11 – LRDIMM модул	56
Слика 2.12 – Проточна обрада код DRAM меморије	58
Слика 2.13 – Еволуција DRAM архитектура.....	60
Слика 2.14 – Организација меморијског система	71
Слика 4.1 – Прелаз између две команде у DCT FSM.....	89
Слика 4.2 – Карактеризација DRAM циклуса на меморијској магистрали	94
Слика 5.1 – Графикони метрика перформанси DRAM меморија.....	110
Слика 6.1 – Ток рада према предложеној методологији	112
Слика 6.2 – Имплементација ACDC модела у Excel-у	117
Слика 6.3 – Дистрибуција вредности есенцијалног искоришћења	120
Слика 6.4 – Визуелизација постпроцесираних података.....	121
Слика 6.5 – Спектрална анализа	123
Слика 6.6 – Анализа узрока.....	124

Слика 7.1 – Системска анализа перформанси: Корелација догађаја у систему	127
Слика 7.2 – Универзална компонента за верификацију DRAM протокола.....	133
Слика 7.3 – 2.5D и 3D интеграција меморијских кола.....	134
Слика 7.4 – Wide I/O DRAM	135
Слика 7.5 – Поређење (3D) Wide I/O 2 и (2D) LPDDR3/4 меморија	136
Слика 7.6 – HBM DRAM	137
Слика 7.7 – Освежавање појединачних банки код HBM DRAM.....	138
Слика 7.8 – Поређење пропусне моћи DDR3, GDDR5, Wide I/O и HBM меморија	139
Слика 7.9 – Структура HMC DRAM	140

Скраћенице

ACDC	Accurate Characterization of DRAM Cycles
AL	Additive Latency
AMB	Advanced Memory Buffer
ASIC	Application Specific Integrated Circuit
AP	Auto-Precharge
BC	Burst Chop
BEDO	Burst-mode EDO
BFM	Bus Functional Model
BL	Burst Length (у контексту DRAM протокола)
BL	Bit Line (у контексту структуре DRAM меморије)
BLFC	Bit Lane Failover Correction
BTT	Bus Turnaround Time
BR	Blu-Ray
CAM	Content-Addressable Memory
CAS	Column Address Strobe
CB	Command Bus
CC	Command-to-Command
CCD	CAS-to-CAS Delay
CD	Compact Disc (у контексту врста меморија)
CD	Command-to-Data (у контексту DRAM протокола)
CL	CAS (Read) Latency
CPU	Central Processing Unit
COB	Capacitor Over Bitline
CUB	Capacitor Under Bitline
CWL	CAS Write Latency
DB	Data Bus
DC	Data-to-Command

DCT	DRAM Command Transition
DD	Data-to-Data
DDR	Double Data Rate
DIMM	Dual In-line Memory Module
DRAM	Dynamic RAM
DSP	Digital Signal Processor
DUT	Device Under Test
DVD	Digital Video Disc
eDRAM	Embedded DRAM
ECC	Error Code Checking
EDA	Electronic Design and Automation
EDO	Extended Data Out
EPROM	Erasable PROM
EEPROM	Electrically Erasable PROM
EU	Essential Utilization
FAW	Four-bank Activation Window
FB	Fully Buffered
FBGA	Fine-pitch Ball Grid Array
FF	Flip-Flop
FG	Floating Gate
FIFO	First In First Out
FPGA	Field-Programmable Gate Array
FPM	Fast Page Mode
FSM	Finite State Machine
GDDR	Graphics DDR
HBM	High Bandwidth Memory
HDD	Hard Disk Drive
HMC	Hybrid Memory Cube
HPC	High Performance Computing
IC	Integrated Circuit
IP	Intellectual Property
JEDEC	Joint Electron Device Engineering Council
LPDDR	Low Power DDR

LRDIMM	Load Reduced DIMM
MCM	Multi Chip Module
MT	Magnetic Tape
MTJ	Magnetic Tunnel Junction
MRAM	Magnetoresistive RAM
NC	Non-Characterizable
NV	Non-Volatile
ODT	On-Die Termination
OCM	On-Chip Memory
OE	Output Enable
OS	Operating System
PC	Phase Change
PCM	Phase Change Memory
PLL	Phase-Locked Loop
PRAM	Phase-change RAM
PROM	Programmable ROM
RAM	Random Access Memory
RAD	Request Access Distance
RAS	Row Address Strobe
RC	RAS Cycle
RCD	RAS-to-CAS Delay
RDIMM	Registered DIMM
RFC	Refresh Cycle
RL	Read Latency
RO	Read Only
ROM	Read Only Memory
RP	Row Precharge
RRAM	Resistive RAM (Memristor)
RRD	RAS-to-RAS Delay
RTL	Register Transfer Level
RTP	Read To Precharge
RTW	Read To Write
RW	Read-Write

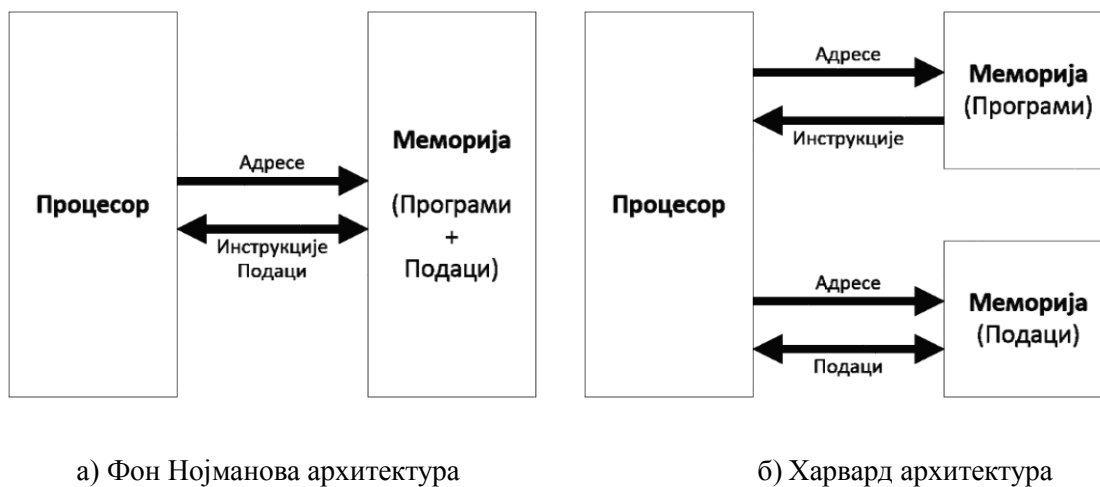
SDR	Single Data Rate
SDRAM	Synchronous DRAM
SEM	Scanning Electronic Microscope
SGRAM	Synchronous Graphics RAM
SOC	System On Chip
SODIMM	Small Outline DIMM
SPD	Serial Presence Detect
SRAM	Static RAM
SSD	Solid State Drive
STT	Spin-Transfer Torque
SV	Spin Valve
TDM	Time Division Multiplex
TSV	Through-Silicon Vias
UDIMM	Unregistered DIMM
UVC	Universal Verification Component
UVM	Universal Verification Methodology
VIP	Verification IP
VLSI	Very Large Scale Integration
WL	Write Latency (у контексту DRAM протокола)
WL	Word Line (у контексту структуре DRAM меморије)
WRT	Write Recovery Time
WTR	Write To Read

1 Увод

Меморија је уређај који може привремено или трајно да складишти информације. Меморија представља једну од кључних компоненти сваког рачунарског система, у ком се непрекидно одвија процес читања података ускладиштених у меморији, њихове обраде и поновног уписа у складу са инструкцијама које систему задаје корисник. Обраду података врши процесор извршавањем процесорских наредби, односно програма, који се најчешће такође налазе ускладиштени у рачунарској меморији.

Готово сви рачунарски системи који се могу срести у пракси су пројектовани у складу са Фон Нојмановом или Харвард архитектуром (Слика 1.1) или представљају неку од њихових модификација. Код ових архитектура меморија у којој се чувају програми и подаци мора бити са произвољним адресирањем, односно мора обезбедити непосредан приступ било ком податку у меморији. Обично се подразумева да је грануларност података један бајт (енг. byte). Таква меморија се назива меморија са случајаним приступом или RAM (енг. Random Access Memory). У савременим рачунарским системима RAM, који се још назива и главна меморија или системска меморија, се најчешће имплементира помоћу меморија типа DRAM.

DRAM меморија се користи у готово свим врстама рачунарских система. Њен значај и распрострањеност се могу илустровати широким распонем платформи код којих се може срести: од носивих (енг. wearable) и мобилних (енг. mobile) уређаја, као што су паметни сатови и мобилни телефони, преко преносних (енг. portable) уређаја попут таблета и лаптопа, до стоних рачунара (енг. desktop), радних станица (енг. workstation), великих серверских платформи и суперрачунара.



Слика 1.1 – Архитектуре рачунарских система

Код Фон Нојманове архитектуре постоји једна меморија у коју се смештају и програми и подаци, док су код Харвард архитектуре програми и подаци смештени у две одвојене меморије. Меморије су повезане са процесором, тј. остатком рачунарског система, преко меморијске магистрале која има адресне линије, линије података и контролне линије (нису представљене на слици).

Разлог тако широке примене DRAM меморија се налази у повољном односу густине складиштења информација, брзине приступа, потрошње енергије и цене у односу на друге врсте меморија.

1.1 Врсте меморија

Постоји велики број различитих врста меморија у зависности од тога како се информације у њима памте и како им се приступа, тј. у зависности од физичких принципа на којима се заснива њихов рад. Међу најважније критеријуме за класификацију меморија спадају: технологија израде, врста меморијских ћелија, постојаност, начин адресирања и врста приступа. Карактеристике релевантних типова меморија које се могу срести у пракси су дате у Табели 1.1.

Технологија израде описује како се физички имплементира меморијски уређај. Меморијски уређај се састоји од низа меморијских ћелија. Меморијске ћелије представљају најмање структуре које могу да ускладиште неку

Меморија	Технологија	Ћелија	Постојаност	Адресирање	Пристап	Примена
DRAM	Полупровод.	1Т1С	Непостојана (Динамичка)	Произвољно	Читање,Упис	Кеш меморија, Физичка меморија
SRAM	Полупровод.	6Т	Непостојана (Статичка)	Произвољно	Читање,Упис	Кеш меморија, Бафери
Регистарска	Полупровод.	FF	Непостојана (Статичка)	Произвољно	Читање,Упис	Регистри
EPROM	Полупровод.	1Т(FG)	Постојана	Произвољно	Читање,Упис*	ROM
EEPROM	Полупровод.	1Т(FG)	Постојана	Произвољно	Читање,Упис	ROM
Флеш,SSD	Полупровод.	1Т(FG)	Постојана	Произвољно	Читање,Упис	Виртуелна меморија, Складишт. података
HDD	Магн. диск	Магн. дипол	Постојана	Секвенцијално	Читање,Упис	Виртуелна меморија, Складишт. података
MT	Магн. трака	Магн. дипол	Постојана	Секвенцијално	Читање,Упис	Архивир. података
CD,DVD,BR	Оптич. диск	Рефл. површ	Постојана	Секвенцијално	Читање	Архивир. података
PRAM	Полупровод.	1Т1Р	Постојана	Произвољно	Читање,Упис	Стална меморија
RRAM**	Полупровод.	1Т1Р	Постојана	Произвољно	Читање,Упис	Стална меморија
MRAM**	Полупровод.	1Т1MTJ/SV	Постојана	Произвољно	Читање,Упис	Физичка меморија

Табела 1.1 – Врсте меморија

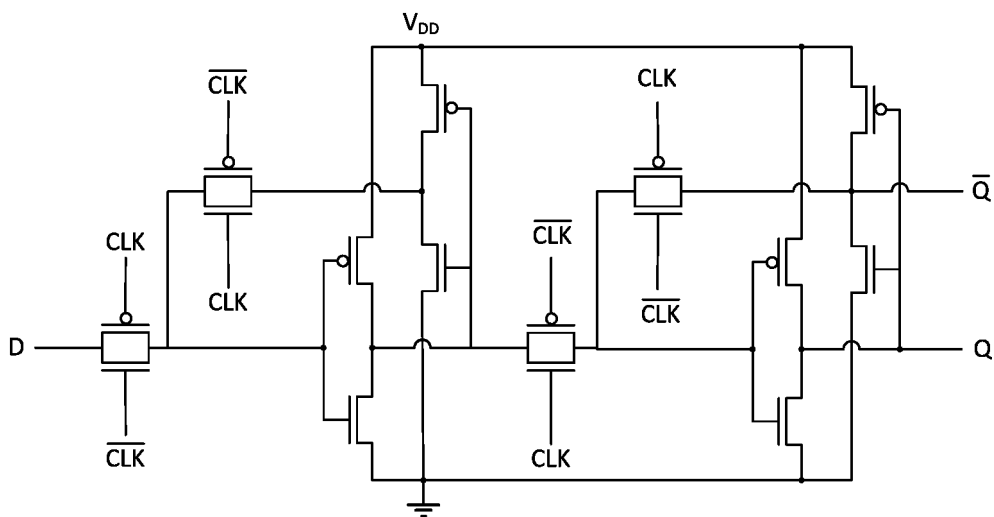
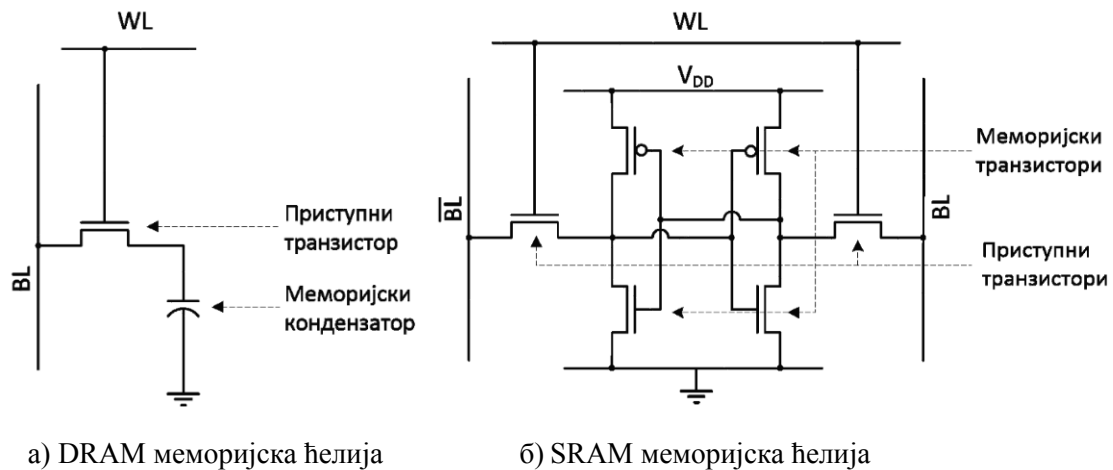
Кључне особине меморија које се користе у савременим рачунарским системима. FF=Flip-Flop, T=Transistor, C=Capacitor, R=Resistor, FG=Floating Gate, P=Phase change cell, MTJ=Magnetic Tunnel Junction, SV=Spin Valve,

*=захтева претходно брисање UV зрацима

**=захтева даљи развој технологије.

информацију. Технологија израде и типови меморијских ћелија које се у датој технологији могу реализовати су међусобно у тесној вези и пресудно утичу и на остале битне особине меморије.

Постојаност дефинише да ли се ускладиштене информације губе приликом прекида напајања, у ком случају се меморија назива непостојана (енг. volatile), или не, у ком случају се назива постојана (енг. non-volatile). У вези са непостојаним меморијама се дефинише и то да ли су ускладиштене информације стабилне (статичке) или их је потребно освежавати у току рада како не би дошло до губитка информација (динамичке). У вези са постојаним меморијама се дефинише појам поузданости (енг. reliability), односно колико су информације у меморији подложне оштећењу услед различитих унутрашњин и спољних фактора. Поузданост има две



Слика 1.2 – Ћелије непостојаних полупроводничких меморија

DRAM меморијске ћелије су знатно мање и троше мање енергије од SRAM меморијских ћелија, док су SRAM меморијске ћелије брже. Флип-флопови су најбрже, највеће и троше највише енергије међу ћелијама непостојаних полупроводничких меморија.

компоненте: динамичку, која се карактерише у виду издржљивости (енг. endurance) и статичку, која се карактерише у виду времена трајања података (енг. data retention). Издржљивост представља број циклуса уписа и читања које меморија може да издржи, а да не дође до оштећења меморијских ћелија, док време трајања одређује колико дуго се једном ускладиштена информација може чувати у меморији без оштећења.

Начин адресирања дефинише да ли је свим подацима унутар меморије могуће приступити непосредно (произвољно), тј. да ли се сваки податак може очитати или уписати са истим кашњењем на задату адресу, или се мора секвенцијално приступати неким деловима меморије пре него што се приступи траженом податку.

Врста приступа дефинише да ли се у стандардним режимима рада меморија може само читати или се у њу може и писати¹.

DRAM спада у класу непостојаних динамичких полупроводничких меморија. Појам полупроводничка меморија подразумева да су меморијске ћелије имплементиране уз помоћ полупроводничких структура, у овом случају транзистора. DRAM меморијска ћелија, на пример, се имплементира помоћу само једног транзистора (Слика 1.2.а), што јој омогућава да има мале димензије и малу потрошњу енергије.

У рачунарским системима се такође користе и непостојане статичке меморије, у које спадају SRAM и регистарске меморије. За реализацију SRAM меморијске ћелије потребно је шест транзистора (Слика 1.2.б), што изискује знатно већу површину (више од десет пута) и потрошњу енергије у односу на DRAM меморијске ћелије. Стога се SRAM користи само за примене где је потребна већа количина брзе меморије, на пример за имплементацију кеш меморија или интерних бафера. Регистарске меморије се реализују помоћу флип-флопова (Слика 1.2.в). Флип-флопови имају време приступа од само једног циклуса, али захтевају 10 или више транзистора за реализацију, у зависности од типа². У просеку флип-флопови

¹ Подразумева се да се и код меморија које су предвиђене само за читање подаци морају најпре уписати (нпр. „нарезати“ оптички диск, „флеш-овати“ EPROM, итд.). У том смислу, под стандардним режимима рада меморија се овде подразумевају примене у оним ситуацијама где упис не утиче на укупне перформансе система, тј. где се упис података дешава само једном, пре употребе меморије.

² Рецимо, ивични D флип-флопови у стандардној библиотеци ћелија типично имају 16 транзистора, али се дизајн флип-флопова може оптимизовати изостављањем појединих транзистора ради балансирања различитих карактеристика, као што су површина, потрошња, брзина, функционалност или поузданост рада. Тако су, на пример, познате имплементације ивичног D флип-флопа са само 12 транзистора, али без комплементарног излаза или са само 10 транзистора, али који могу да раде само у динамичком режиму.

заузимају два до три пута већу површину и имају неколико пута већу потрошњу од SRAM ћелија. Регистарска меморија се користи за примене где су брзина и синхронизација рада суштински важни, на пример за имплементацију регистара, регистарских банки или синхронизационих бафера.

Поред непостојаних, користе се и постојане полупроводничке меморије, од којих су најзаступљеније EPROM, EEPROM и флеш меморија. Ћелије ових меморија су, као и DRAM меморијска ћелија, једнотранзисторске, али другачије транзисторске структуре, са једним неизолованим и једним изолованим каналом (енг. floating gate). Због физичких и технолошких ограничења у изради њихово време трајања и брзина приступа су ограничени. Флеш меморија се данас широко користи пре свега за имплементацију спољашњих меморија за масовно складиштење података (флеш картице, SSD) и као замена за EEPROM, на пример за имплементацију инструкцијске меморије код микроконтролера са Харвард архитектуром.

Постоје и друге постојане полупроводничке меморије, попут фазно-променљивих (PRAM), резистивних (RRAM) и магнето-резистивних (MRAM), које раде на другачијим физичким принципима. Међутим, упркос неким својим предностима, њихова поузданост, перформансе и цена и даље им не омогућавају широку примену у доменима где се користи DRAM меморија. Издржљивост ових меморија је типично знатно већа него код EPROM, EEPROM и флеш меморија и може досећи и неколико милиона циклуса, али је дозвољени број циклуса уписа и даље ограничен. Такво ограничење је неприхватљиво за имплементацију главне меморије, с обзиром да се одговарајући лимит може достићи за само неколико милисекунди рада процесора.

У пракси се користе још и магнетне и оптичке меморије. Њихова мана је што за приступ користе електро-механичке компоненте, чиме се време погоршава за неколико редова величине у односу на полупроводничке меморије, а проток података је ограничен брзином периферијске магистрале. Због тога се ове меморије користе углавном за складиштење података или њихово трајно архивирање.

1.2 Меморије у рачунарском систему

Поред постојаности, начина адресирања и врсте приступа, постоји и низ других особина које су од значаја за дефинисање улоге коју одређена меморија може да има у рачунарском систему. Неке од тих особина су: време приступа, проток података, густина паковања информација, капацитет, потрошња енергије и цена. На пример, брзе меморије се користе за имплементацију унутарпроцесорских меморијских структура попут регистара, бафера и кешева, док се јефтине меморије великог капацитета користе за трајно складиштење велике количине података на екстерним медијумима, као што су оптички дискови и магнетне траке. У овом поглављу ће бити речи о месту и улози различитих врста меморија у рачунарском систему.

1.2.1 Меморијска хијерархија

На почетку поглавља је поменуто да се у архитектури рачунарских система користи једноставан модел рачунарске меморије у коме постоји само главна меморија (RAM) као јединствен линеарни адресни простор у коме је сваком податку могуће директно приступити. Поред главне меморије, подразумева се да постоје и интерни процесорски регистри, као привремене локације у којима се чувају улазне и излазне вредности процесних јединица ради лакше и брже обраде. Овакав модел одражава потребу за што једноставнијом представом меморије коју треба да има корисник приликом писања програма.

У стварности, тако једноставна организација меморије није довољно ефикасна. Због велике разлике у брзини рада процесора и главне меморије, сваки приступ главној меморији би трајао веома дуго. Зато се у пракси прибегава креирању меморијске хијерархије коју чини, идући од главне меморије према процесору, више нивоа сукцесивно све бржих и мањих меморија, кешева (енг. caches), које чувају део садржаја главне меморије са циљем да се време приступа подацима који се често користе што више скрати, користећи временску и просторну локалност података (Слика 1.3). Парадигма приступа меморији притом остаје иста, тј. програми и даље виде меморију у виду јединственог и недељивог адресног



Слика 1.3 – Меморијска хијерархија

Мање и брже меморије се смештају ближе процесору, а веће и јефтиније меморије даље.

простора, односно као RAM. Све што је потребно да би се одржала кохеренција делова меморијске хијерархије се обавља на хардверском нивоу, транспарентно за корисника. Меморија која се може директно адресирати од стране процесора се назива примарна меморија.

Меморија коју процесор не може директно адресирати током извршавања програма служи за складиштење података (програма, слика, база података, и др.) и назива се секундарна меморија. Подаци се у том делу меморије обично складиште у облику датотека (енг. files) којима се приступа помоћу тзв. датотечног система (енг. file system). Сваки тип меморије има датотечни систем који му одговара. На пример, ISO 9660 датотечни систем је дизајниран за приступ оптичким дисковима, FAT16/32 и NTFS се користе као диск датотечни системи у оквиру оперативног система Windows, ext2/3/4 као диск датотечни системи у оквиру оперативног система Linux и сл. Постоје и десетине других датотечних система оптимизованих за складиштење и приступ подацима на магнетним тракама, флеш меморијама и др. који се користе у пракси. Треба нагласити да је могуће и податке у RAM-у организовати тако да им се приступа преко датотечног система. Пример за то је

RAM диск, где се физичка RAM меморија (DRAM) користи као медијум за чување и приступ датотекама. Његов капацитет је ограничен величином RAM-а, али је приступ неколико редова величине бржи него код нпр. HDD и SSD.

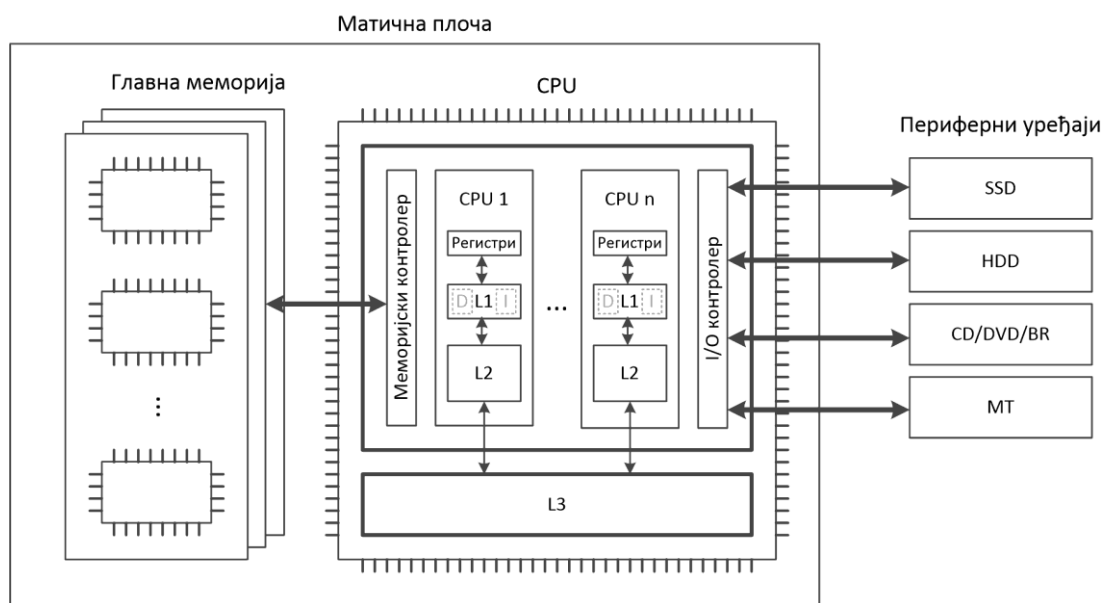
Поред проблема брзине приступа, постоји и проблем капацитета главне меморије. Да би се избегле тешкоће које настају због блокирања рада система када је величина главне меморије мања од потребне, користи се виртуелна меморија. Виртуелна меморија је приступ који омогућава сваком програму да меморију види као неограничен и недељив адресни простор, без обзира на његову физичку величину, попуњеност и расподелу. Функцију управљања виртуелном меморијом у том случају преузима на себе оперативни систем који, уз помоћ одговарајућих хардверских компоненти (енг. memory management unit), има улогу да обавља замену садржаја главне меморије, која се у овом контексту често назива и физичка меморија, са одговарајућим деловима секундарне меморије онда када више нема места у физичкој меморији. Уобичајено је нпр. да се делови HDD-а или SSD-а алоцирају у ове сврхе (енг. swap space). Чак и када секундарна меморија има нпр. секвенцијалан приступ, оперативни систем чини произвољан приступ одговарајућим деловима виртуелне меморије потпуно транспарентним. Са становишта програма, цела виртуелна меморија се понаша као проширење главне меморије, односно као RAM. Може се приметити да виртуелна меморија и RAM диск имају инверзне улоге. Код RAM диска се главној меморији, која има произвољан приступ, приступа коришћењем датотечног система, а код виртуелне меморије се секундарној меморији, којој се приступа преко датотечног система, приступа коришћењем произвољног адресирања.

1.2.2 Локација делова меморијске хијерархије у рачунарском систему

Делови меморијске хијерархије су физички лоцирани на различитим местима у оквиру рачунарског система (Слика 1.4). Регистри и део кеш меморије (типично L1 и L2) готово увек се имплементирају у оквиру истог интегрисаног кола (енг. integrated circuit) као и процесор. То је могуће зато што се и регистарска меморија и SRAM имплементирају коришћењем истог технолошког процеса као и дигитална логичка кола. Највиши ниво кеш меморије (L3 или L4) се због своје

величине некада имплементира и одвојено, као посебно интегрисано коло. Ако је највиши ниво кеша SRAM меморија, то се ради због повећања приноса (енг. yield) силицијумског вафера (енг. silicon wafer) и смањења цене производње, а ако је највиши ниво кеша DRAM меморија (тзв. eDRAM), то се ради из технолошких разлога. У том случају се највиши ниво кеша и процесор повезују у један чип у оквиру истог паковања или један поред другог (енг. multi-chip module) или један изнад другог (енг. stacked die), чиме се ограничава утицај дужине веза и сметњи на пад перформанси који настаје због физичке раздвојености.

Главна меморија се обично имплементира у виду одвојених DRAM интегрисаних кола који се повезују са процесором преко меморијске магистрале. Разлог за то је што производња DRAM меморија захтева различит технолошки процес од производње интегрисаних кола са превасходно дигиталном логиком као што је процесор. Процесорска и DRAM интегрисана кола могу се повезати у оквиру



Слика 1.4 – Локација делова меморијске хијерархије у рачунарском систему

Различити делови меморијске хијерархије су лоцирани на: 1) истом интегрисаном колу заједно са процесором (CPU1..n), 2) у оквиру истог паковања као и процесорско интегрисано коло (CPU), 3) као чипови на матичној плочи (главна меморија) и 4) као периферни уређаји. На слици је приказан пример где је L3 кеш имплементира као засебно интегрисано коло у истом паковању са процесором, а интегрисана кола главне меморије су пакована у засебне чипове и модуле.

истог паковања, тј. у један чип, што је, на пример, чест случај код чипова за играчке конзоле (енг. gaming consoles). Тиме се повећавају перформансе и смањује цена система, јер код њих флексибилност у смислу величине и брзине меморије није потребна с обзиром да су непроменљиви. DRAM интегрисана кола се, ипак, најчешће пакују одвојено, тј. у посебне меморијске чипове. DRAM чипови могу бити директно залемљени на матичној плочи (енг. embedded) или се више DRAM чипова може повезати на посебним штампаним плочицама, меморијским модулима, који се умећу у одговарајуће слотове на матичној плочи. Први приступ омогућава скраћивање веза између меморије и процесора, а тиме и боље перформансе система, док други приступ омогућава кориснику избор типа меморија и произвођача по цену нешто нижих перформанси и веће потрошње.

Флеш меморија (SDD), магнетни дискови (HDD), магнетне траке (MT) и оптички дискови (CD, DVD, BR) се типично повезују са остатком система као периферни уређаји преко периферијских контролера (енг. I/O controllers) и одговарајућих магистрала (нпр. PCIЕ, SATA и др.).

1.2.3 Карактеристике рачунарских меморија

Улога меморије у рачунарском систему, односно њено место у меморијској хијерархији, је одређено њеним физичким и технолошким карактеристикама, перформансама и ценом.

У физичке карактеристике спадају густина паковања информација, максимални капацитет и потрошња енергије.

Густина паковања информација говори о томе колика количина информација се може сместити на одређеном простору. Већина меморија има физички изражену дводимензионалност, тј. информације се смештају по површини меморијског медијума (полупроводничког чипа, магнетног диска или траке, оптичког диска и сл.). Стога се густина паковања информација обично мери у гигабитима по центиметру квадратном (Gb/cm^2). Са развојем 3D полупроводничких технологија и других технологија за тродимензионално складиштење информација (нпр. холографских меморија), густина паковања информација ће почети да се мери и по јединици запремине (Gb/cm^3).

Максимални капацитет меморије одређује количину информација која може да се смести у оквиру једног меморијског уређаја. Капацитет се изражава у бајтовима, са одговарајућим SI префиксима за веће вредности: кило (К), мега (М), гига (Г), тера (Т) итд. Величина регистарске меморије се нпр. типично изражава у КВ, величина кеш меморије у МВ, главне меморије у ГВ, секундарне меморије у ТВ и сл.

Потрошња енергије говори о томе колика енергија је потребна за читавање или упис једног бита информације у меморију. Потрошња енергије се обично мери у пикоџулима по биту (pJ/b). На основу овог податка и максималног протока може се израчунати укупна потрошња енергије меморијског уређаја, а на основу густине паковања информација и термална дисипација по јединици површине (W/cm^2) или запремине (W/cm^3) ради процене потребе за хлађењем.

У технолошке карактеристике спадају информације о томе да ли меморија захтева механичке, оптичке или друге компоненте за приступ, да ли је процес производње и паковања меморије компатибилан са процесом производње и паковања процесора и сл. Тако магнетни дискови и траке нпр. имају механичке компоненте, а оптички дискови механичке и оптичке компоненте, што их чини непогодним за интеграцију у мобилне уређаје. Са друге стране, регистри и SRAM се могу имплементирати у истој технологији као и процесор у оквиру једног интегрисаног кола, SRAM и DRAM интегрисана кола се могу повезати и паковати заједно са процесорским колима у оквиру истог чипа, DRAM чипови се могу лемити на истој плочи са процесорским чипом итд.

Перформансе меморије се дефинишу помоћу две компоненте: времена приступа, односно кашњења (енг. latency), и максималног протока података (енг. bandwidth). Треба имати у виду да перформансе могу битно да варирају у зависности од тога да ли се врши читање или упис, а код меморија са секвенцијалним приступом је од веома великог значаја редослед приступа подацима.

Време приступа означава колико времена мора проћи од тренутка од када процесор затражи приступ податку док се податак не очита из меморије или упише

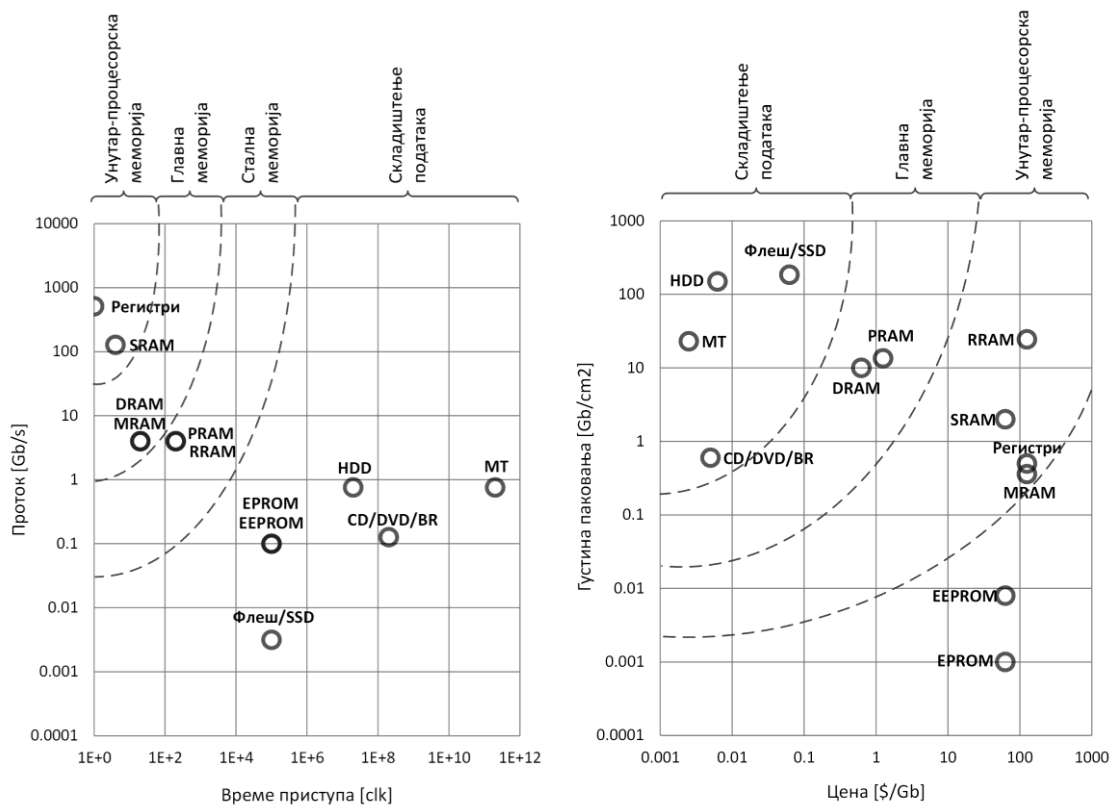
у меморију. Време приступа се мери у секундама (s), уз примену стандардних SI префикса за мање вредности: пико (p), нано (n), микро (μ), мили (m) итд. Алтернативно, време приступа се може мерити и релативно у односу на брзину процесора, преко броја тактова процесора (clk). Време приступа варира између различитих меморија за неколико редова величине и непосредно се одражава на место које меморија има у меморијској хијерархији. Тако се меморије са краћим временом приступа налазе више у хијерархији, односно ближе процесору, и обрнуто.

Проток података означава количину информација којој се може приступити у јединици времена. Обично се означава у гигабитима у секунди (Gb/s). Време приступа и проток не морају бити у корелацији. На пример, код већине меморија са непосредним приступом кашњење је мало, а проток велики, што је адекватно за имплементацију примарне меморије. Са друге стране, код већине меморија са секвенцијалним приступом време приступа је обично велико, док проток варира у зависности од редоследа приступа подацима, што је адекватно за имплементацију секундарне меморије.

Цена меморије показује колико кошта производња меморијског уређаја по јединици ускладиштене информације. Цена се најчешће изражава у \$/Gb. У пракси је важна не само апсолутна цена меморије, већ однос цене и перформанси. Апсолутна цена меморије је од примарног значаја за секундарне меморије, јер оне складиште велику количину информација. Да би цена меморијског уређаја била прихватљива за корисника, потребно је да цена по јединици ускладиштене информације буде довољно ниска. Што је меморија позиционирана више у меморијској хијерархији, то перформансе постају све важније од цене. Тако је, на пример, цена регистара и SRAM меморије за један или два реда величине већа од цене DRAM меморије, а три или више редова величине од цене флеш меморије.

1.2.4 Примена рачунарских меморија

Улога рачунарских меморија у рачунарском систему се дефинише на основу њихових карактеристика (Слика 1.5). Меморије са малим временом приступа и великим протоком (уобичајен назив за њих је „брзе“ меморије), које имају



Слика 1.5 – Карактеристике меморија и њихова улога у рачунарском систему

График приказује основне карактеристике рачунарских меморија (подаци за 2014-2015 годину). Са графика се може закључити да: 1) регистарска меморија и SRAM испуњавају критеријуме за унутар-процесорске меморије, 2) једино DRAM задовољава критеријуме за имплементацију главне меморије, 3) HDD, SSD, MT и CD/DVD/BR задовољавају критеријуме за секундарне меморије, 4) пресек карактеристика за MRAM, PRAM и RRAM [1] сугерише да за њих треба креирати нов ниво у меморијској хијерархији и 5) EPROM и EEPROM немају логично место у меморијској хијерархији, већ се могу користити само за специфичне примене.

произвољно адресирање и дозољавају и читање и упис су природни кандидати за имплементацију примарне меморије, односно за виша места у меморијској хијерархији. Меморије које су јефтине и имају велику густину паковања, али не неопходно и високе перформансе, су погодне за имплементацију секундарне меморије, односно за нижа места у меморијској хијерархији.

Међу кандидатима за реализацију примарне меморије спадају регистарске меморије, SRAM, DRAM, MRAM, PRAM и RRAM. Од њих, само регистарске

меморије и SRAM су довољно брзе да их има смисла имплементирати у оквиру истог интегрисаног кола као и процесор. Од кључне је важности што су технологије њихове израде компатибилне са технологијом израде процесора. За ове меморије се користи израз унутар-процесорске меморије. Цена регистарских меморија је веома висока због велике површине коју сваки флип-флоп заузима у оквиру полупроводничког чипа, али им је време приступа минимално, па се користе за све примене где је брзина рада критична, на пример за имплементацију регистара, регистарских банки или синхронизационих бафера. Цена SRAM меморија је мања од цене регистарских меморија, јер је површина SRAM меморијских ћелија око 3 пута мања. Међутим, време приступа код SRAM меморија је веће, па се оне користе тамо где није неопходно користити брзу и скупу регистарску меморију, на пример за имплементацију кеш меморија или интерних бафера. У прилог коришћењу SRAM меморија иде и чињеница да је њихова потрошња неколико пута мања од потрошње регистарских меморија.

DRAM је једини тип меморија чији пресек карактеристика задовољава критеријуме за имплементацију главне меморије. Ту се пре свега мисли на перформансе, капацитет, цену и потрошњу енергије. У догледној будућности се не види да би нека друга врста меморије могла да заузме место DRAM меморија у овој улози.

MRAM има готово идентичне перформансе као и DRAM, али за ред величине мању густину паковања информација и знатно већу цену. Практична примена ове меморије је условљена даљим развојем технологије израде са циљем повећања густине паковања и смањења цене производње. Уколико се у будућности у оба ова аспекта MRAM меморија приближи карактеристикама DRAM меморије, онда би се могла користити за имплементацију дела главне меморије, која би била постојана.

PRAM и RRAM имају готово идентичне перформансе, али је RRAM знатно скупљи, уз нешто већу густину паковања, која је већа него код DRAM меморије. Међутим, обе ове меморије имају за ред величине лошије перформансе од DRAM меморије, што их чини недовољно брзим за имплементацију главне меморије. Уместо тога, могуће је користити ове меморије за имплементацију дела меморијске

хијерархије који би се налазио између главне меморије (DRAM) и секундарне меморије (флеш), који би био постојан. За овај део примарне меморије у меморијској хијерархији се користи назив стална меморија.

У меморије који су добри кандидати за реализацију секундарне меморије спадају HDD, SSD, MT и CD/DVD/BR. HDD, MT и CD/DVD/BR не подржавају произвољно адресирање јер имају секвенцијални приступ, док SSD, иако подржава произвољно адресирање, има недовољно добре перформансе у том начину рада да би могао да се користи за имплементацију примарне меморије. Међутим, због релативно ниске цене и задовољавајућих перформанси приликом секвенцијалног приступа, које су боље од перформанси магнетних дискова, ове меморије су почеле масовно да се користе за складиштење података.

EPROM и EEPROM имају малу густину паковања и подржавају само читање у стандардном режиму рада, па се зато користе искључиво за имплементацију ROM меморије.

1.3 Место и значај DRAM меморија

Готово сви рачунарски системи који се користе у пракси имају главну меморију имплементирану помоћу DRAM меморија. Једини изузетак су поједине врсте микроконтролера код којих се главна меморија имплементира помоћу SRAM меморије на чипу (енг. on-chip memory), зато што је количина главне меморије потребна за извршавање управљачког софтвера (енг. firmware) у тим системима веома мала. Сходно чињеници да су DRAM меморије присутне готово свуда, спектар уређаја који их користе је изузетно велики и обухвата системе који се знатно разликују по величини, намени, цени и другим карактеристикама.

Да би се у потпуности сагледало место и значај DRAM меморија у рачунарском систему потребно је размотрити њену улогу са више аспеката, укључујући цену, потрошњу енергије, као и утицај на архитектуру, организацију и перформансе система.

Уређај	Врста	DRAM	Цена DRAM	Цена уређаја	%
Apple Watch	„паметни“ сат	512MB LPDDR3	\$5	\$81	5.5%
Apple iPhone 4	моб. телефон	1GB LPDDR	\$14	\$187	7.4%
Apple iPhone 5	моб. телефон	1GB LPDDR2	\$11	\$199	5.3%
Apple iPhone 5s	моб. телефон	1GB LPDDR2	\$11	\$190	5.8%
Apple iPhone 6s Plus	моб. телефон	2GB LPDDR4	\$17	\$231	7.4%
Samsung Galaxy S6 Edge	моб. телефон	3GB LPDDR4	\$27	\$285	9.7%
Microsoft XBOX One	игр. конзола	8GB DDR3	\$60	\$471	12.7%
Sony PlayStation 4	игр. конзола	2GB DDR3 + 8GB GDDR5	\$88	\$372	23.7%
Apple iPad 3	таблет	1GB LPDDR2	\$13	\$347	4.0%
Apple iPad Air 2	таблет	2GB LPDDR3	\$18	\$270	6.7%
Dell Inspiron 15 3000 Series	лаптоп	4GB DDR3	*\$40	\$330	12.1%
Dell Inspiron 17 5000 Series	лаптоп	8GB DDR3	*\$80	\$600	13.3%
Dell Alienware 15 Controller	лаптоп	16GB DDR4 + 2GB GDDR5	\$230	\$1300	17.7%
AMD Radeon HD 6970	акцелератор	2GB GDDR5	\$48	\$363	13.4%
nVidia GTX TITAN X	акцелератор	12GB GDDR5	*\$120	\$1049	11.4%
Dell XPS 8900	десктоп	16GB DDR4 + 4GB DDR3	*\$120	\$1269	9.5%
HP Pavilion 550z	десктоп	8GB DDR3	*\$40	\$400	10.0%
Dell Precision Tower 5000	радна станица	32GB DDR4 RDIMM ECC + 8GB GDDR5	\$580	\$3150	18.4%
HP Z230	радна станица	32GB DDR3 UDIMM ECC + 4GB GDDR5	\$760	\$2697	28.2%
Dell PowerEdge R230	сервер	64GB DDR4 ECC	\$622	\$2315	26.9%
HPE ProLiant DL20 Gen9	сервер	8GB DDR4 UDIMM	*\$80	\$649	12.3%
HP ProLiant DL360 Gen9	сервер	64GB DDR RDIMM	*\$640	\$6270	10.2%
Dell & SDSC – Comet	суперрачунар	253TB DDR4 ECC	*\$1.8 М	\$21.6 М	8.2%
Cray XK7 – Titan	суперрачунар	584TB DDR3 ECC + 110TB GDDR5 ECC	*\$5.2 М	\$97 М	5.3%

Табела 1.2 – Цена DRAM меморије у оквиру рачунарског система

Табела приказује цене уређаја и одговарајуће DRAM меморије уграђене у њих. Подаци су за период 2013-2015 год. Цене су прикупљене из јавно доступних извора, укључујући интернет странице произвођача, специјализоване публикације, маркетиншка истраживања, малопродајне трговине и др. Неке цене су набављачке, а неке малопродајне, али је вођено рачуна да за сваки појединачни уређај буду из истог извора како би се сачувала конзистентност приликом рачунања односа цене DRAM меморије и целог уређаја. У неким случајевима цене уграђене DRAM меморије нису биле доступне, у ком случају је цена процењена на основу стања на тржишту (ове цене су означене са *). Пошто цене DRAM меморије и уређаја временом значајно варирају, то податке треба узети пре свега као индикаторе, а не као апсолутно тачне.

Тржиште DRAM меморија у 2014. години је било вредно 42 милијарде US долара [2], што је чинило 12.6% од целокупног тржишта интегрисанх кола чија је вредност била 333 милијарде US долара. Овај податак сведочи о економском значају DRAM меморија на глобалном нивоу. Тај однос се са глобалног плана пресликава и на ниво појединачних уређаја. У Табели 1.2 је дат преглед цена рачунарских система из различитих сегмената тржишта и цене одговарајућих DRAM меморија које су у њих уграђене. Из табеле се може видети да DRAM меморија у значајном проценту учествује у цени рачунарских система, при чему тај проценат знатно варира у зависности од врсте уређаја, количине и типа DRAM меморије.

Код уређаја код којих је цена осталих компоненти релативно велика, проценат учешћа у цени целог система је нижи. На пример, „паметни“ сатови, мобилни телефони и таблети због релативно високе цене дисплеја, батерије, камере, сензора и др., а релативно мале количине DRAM меморије због смањења потрошње, имају проценат учешћа у укупној цени у распону од 5% до 10%. Ситуација је слична код суперрачунара, где, упркос чињеници да су због високог нивоа перформанси потребне огромне количине DRAM меморије, релативно висока цена процесних елемената, мрежне инфраструктуре и система за напајање и хлађење чине да проценат учешћа у цени целог система буде на истом нивоу како код мобилних уређаја.

На другом крају спектра се налазе уређаји који захтевају релативно велике количине DRAM меморије у односу на величину система како би могли да подрже примене за које су намењени. Пример за то су графички акцелератори и играчке конзоле којима је потребна велика количина меморије за смештање графичких података који се користе за рендеровање слика високе резолуције у реалном времену. Код ових уређаја проценат учешћа у цени система се обично креће у распону од 10% до 20%. Слично важи за сервере, који морају да подрже вишекориснички рад, мултитаскинг (истовремено извршавање више апликација) и виртуелизацију (истовремени рад више независних инстанци оперативних система), неретко у интерактивном режиму рада. Тако се у [2] наводи податак да је 2014. године просечан проценат учешћа DRAM меморије у цени серверског

система био 16%. Још један сличан пример су радне станице намењене професионалном тржишту које морају да буду у стању да обрађују улазне податке великих димензија (нпр. фотографије, аудио и видео снимке високе резолуције, комплексне 3D моделе, велике базе података итд.), код којих проценат учешћа у цени може бити и преко 25%.

Други аспект са кога се може сагледати утицај DRAM меморија на рачунарски систем се односи на потрошњу енергије. Из различитих извора се може доћи до података да удео DRAM меморије у укупној потрошњи енергије у систему варира у распону од 5% до 25%, у зависности од врсте уређаја, количине и типа меморије, оптерећења система и др. Као илустрација се може навести процена да у једном рачунарском центру при радном оптерећењу од 80% на DRAM меморију одлази 11.7% од укупне потрошње енергије [3]. У овом случају у укупну потрошњу је укључена и потрошња у вези са напајањем, хлађењем и мрежном инфраструктуром. Ако се у обзир узме само потрошња полупроводничких компонената, онда је овај проценат нешто већи (15.4%). Из ових примера се види да DRAM у значајном проценту учествује у укупном енергетском буџету рачунарског система. То је разлог и што системи осетљиви на потрошњу, као што су носиви, мобилни и преносни системи, користе мање количине DRAM меморије, делимично на уштрб перформанси и корисничког комфора. Ради смањења потрошње DRAM меморија често се уводе и технике активног управљања деловима меморијског подсистема, као што је стављање у „успавани“ режим и сл.

DRAM меморија има велики утицај и на архитектуру и организацију рачунарског система и његове перформансе. Тај утицај се одражава пре свега на елементе меморијског подсистема због непосредне везе коју имају са DRAM меморијом. На пример, главна меморија утиче на тип, ширину и брзину меморијске магистрале, параметре меморијског контролера, величину и брзину кешева, архитектуру и параметре интерконеционе мреже и сл. Осим тога, постоји и индиректна веза између DRAM меморије и осталих делова рачунарског система који јој приступају. Кроз главну меморију пролази велики број токова података. У неким од тих случајева меморија је на критичној путањи, а у многима је уско грло. У таквим случајевима перформансе меморије заправо одређују максималне

перформансе компоненти на критичној путањи, односно потребну брзину рада и/или величину тих компоненти. Као пример за илустрацију ове тврдње се може узети графички процесор. Графичке перформансе су лимитиране перформансама графичког процесора (енг. engine bound performance) док год је проток података ка меморији мањи од максималног протока података које меморија може да подржи. Повећање пропусне моћи (енг. throughput), нпр. додавањем нових физичких ресурса или повећавање брзине рада графичког процесора повећањем фреквенције, након што се достигне максимум протока података који меморија може да подржи не доводи до даљег повећања перформанси (енг. memory bound performance). Из тога следи да су потребна брзина рада и величина графичког процесора суштински лимитирани перформансама меморије.

1.4 Анализа перформанси DRAM меморија

Пословица каже да је ланац јак само онолико колико је јака његова најслабија карика. У контексту анализе перформанси рачунарског система, ова пословица би се могла парафразирати на следећи начин: анализа перформанси рачунарског система је добра само онолико колико и најлошије урађена анализа перформанси за неку од компоненти система које се налазе на критичној путањи. Пошто се у многим сценаријима главна меморија налази на критичној путањи, то значи да ако се перформансе DRAM меморије не анализирају добро, онда то може постати „слаба карика“ у ланцу.

1.4.1 Значај анализе перформанси DRAM меморија

Већина система се пројектује према задатим критеријумима, од којих се један од најважнијих односи на перформансе. Пошто перформансе система у многоме зависе од главне меморије, перформансама DRAM меморије се посвећује посебна пажња приликом пројектовања.

Пројектни критеријуми који се тичу перформанси се најчешће задају у виду сценарија (енг. use cases) које систем мора да буде у стању да изврши. Пример једног таквог сценарија за систем намењен мобилним уређајима, нпр. телефонима

или таблетима, може да буде: снимање камером у одређеној резолуцији у задатом видео формату уз упис података на флеш меморијску картицу, праћење онога што камера снима на дисплеју и истовремено прављење фото-снимака у задатом формату са постпроцесирањем и уписом резултата на флеш меморијску картицу, при чему све време ради модем како би могао да се прихвати евентуални телефонски или интернет позив или нотификација. У оваквим сценаријима, где постоје захтеви за одржавањем перформанси у реалном времену (енг. real time), што значи да перформансе не смеју пасти испод одређене границе, као што је нпр. случај код камере, дисплеја, аудио и видео процесора, модема, итд., често само неколико процената меморијских перформанси одлучује о томе да ли је систем у стању да испуни постављене захтеве или не. Значај анализе перформанси DRAM меморија се може сагледати кроз разматрање два могућа случаја: једног у ком је меморија уско грло система и другог у ком није.

У првом случају, ако се, нпр. због нетачног мерења или моделирања перформанси DRAM меморије, не установи да је меморија заиста уско грло, анализа лако може отићи у правцу трагања за непостојећим хардверским или софтверским проблемима и тако довести до узалудног губитка труда и времена. Поред тога, пошто су цена и перформансе система пропорционални површини полупроводничог субстрата које систем заузима у интегрисаном колу, то значи да када су перформансе система ограничене могућностима меморије, систем је заправо предимензионисан, а тиме и скупљи него што би морао бити. Предимензионисање има своју цену не само у смислу заузећа веће површине интегрисаног кола, него и у смислу веће потрошње енергије приликом рада система, као и дужег трајања развоја и већег утрошка ресурса приликом развоја. У зависности од архитектуре, повећање перформанси изнад одређеног нивоа често има суперлинеарну зависност у односу на задате параметре пројектовања (површину, потрошњу енергије и брзину рада), што може додатно да повећа цену развоја.

У другом случају, ако се DRAM меморија грешком прогласи за уско грло система, може доћи до маскирања стварних недостатака у архитектури, грешака у дизајну, пропуста у конфигурацији система или мањкавости приликом

пројектовања софтвера. У том случају ће систем имати лошије карактеристике од жељених и бити мање конкурентан на тржишту, што доводи до негативних маркетиншких и финансијских ефеката по произвођача.

Сличне последице наступају и у сценаријима где се мере перформансе делова система који немају ограничења рада у реалном времену, већ раде по принципу максималног могућег учинка (енг. best effort), као што су графика, апликациони процесор, MMU, итд. Једина разлика је у томе што граница неиспуњења постављених критеријума може бити мање оштра.

1.4.2 Тачност код анализе перформанси DRAM меморија

Услед виртуелизације и раста броја меморијских клијената [4], од којих сваки има различите обрасце приступа меморији, меморијски саобраћај постаје све мање предвидљив. То отежава моделовање и анализу перформанси DRAM меморија постојећим методама и повећава нетачност. Трендови истовременог повећања оптерећења DRAM меморије и смањења тачности мерења њених перформанси воде ка томе да, из тачке гледишта анализе перформанси система, DRAM меморија све више почиње да личи на непрозирну „црну кутију”, што отежава правилну анализу рада система. Из тих разлога значај тачности анализе перформанси DRAM меморија постаје све већи и већ сада се намеће не само као пожељан, већ и као неопходан услов за квалитетну анализу рада савремених рачунарских система.

Поред тачности у смислу апсолутног одступања вредности измерених параметара од стварних, за анализу је изузетно битан и аспект тачности који се односи на грануларност, односно резолуцију са којом се перформансе могу мерити. Због динамичке природе система, периоди у којима DRAM представља ограничавајући фактор могу се јављати у неправилним интервалима различитог трајања. Када се анализира период времена који је дужи од трајања таквих интервала, средња вредност измерених перформанси може бити знатно нижа него за време њиховог трајања. Уколико је минимална грануларност мерења ниска, односно ако се перформансе могу мерити само у дужим интервалима, то и поред тачно измерених параметара може довести до стварања погрешне макро-слике рада

система према којој DRAM наизглед никада не представља уско грло, што би довело до погрешних закључака у анализи и онемогућило разумевање праве природе проблема. Такве ситуације се не могу увек лако препознати и могу довести до значајног губитка времена, труда и новца. Једини начин да се перформансе DRAM меморије могу поуздано мерити и анализирати је да се карактеризација релевантних параметара врши са резолуцијом на нивоу појединачних циклуса, независно од трајања периода у ком се резултати касније интерпретирају.

1.4.3 Утицај развојних трендова на анализу перформанси DRAM меморија

За сада се не види технологија која би у догледно време могла да замени DRAM у имплементацији системске меморије. Стога се може очекивати да ће област анализе перформанси DRAM меморија имати велики значај и у будућности. У прилог томе иду и трендови развоја у домену технологије израде, архитектуре и организације рачунарских система и пројектовања системског софтвера, који чине да меморија временом све више постаје уско грло система.

Познато је да напредак у технологији израде полупроводника има већи ефекат на повећање брзине рада елемената дигиталне логике него на повећање брзине рада DRAM меморија. Због минимизовања струја цурења приступни транзистори у DRAM ћелијама имају већи напон прага и веће димензије (дужи канал) у односу на транзисторе у логичким колима. Такође, меморијски кондензатор мора бити довољно велики да држи наелектрисање до наредног циклуса освежавања. Из тих разлога је ефекат смањивања димензија са напретком технологије израде умањен код DRAM меморија. Стога се јаз у перформансама између меморије и остатка система повећава. Тај проблем је познат као „меморијски зид“. Истовремено, стална потреба за повећањем могућности пројектованих система употребом све већег броја апликационих и DSP процесора, мултимедијалних акцелератора и улазно-излазних контролера доводи до пораста броја меморијских клијената, што са једне стране доводи до раста меморијског саобраћаја, чиме се повећава могућност да DRAM постане уско грло, а са друге стране до повећавања конкурентности приликом приступа DRAM меморији, чиме се смањује њена ефикасност и проблем додатно погоршава. Коначно, појава

виртуелизације, где исте хардверске ресурсе истовремено дели већи број клијената, од којих сваки може да извршава више инстанци оперативних система, мултиплицира претходне ефекте и чини да DRAM још брже постаје уско грло система.

1.5 Проблеми у вези са анализом перформанси DRAM меморија

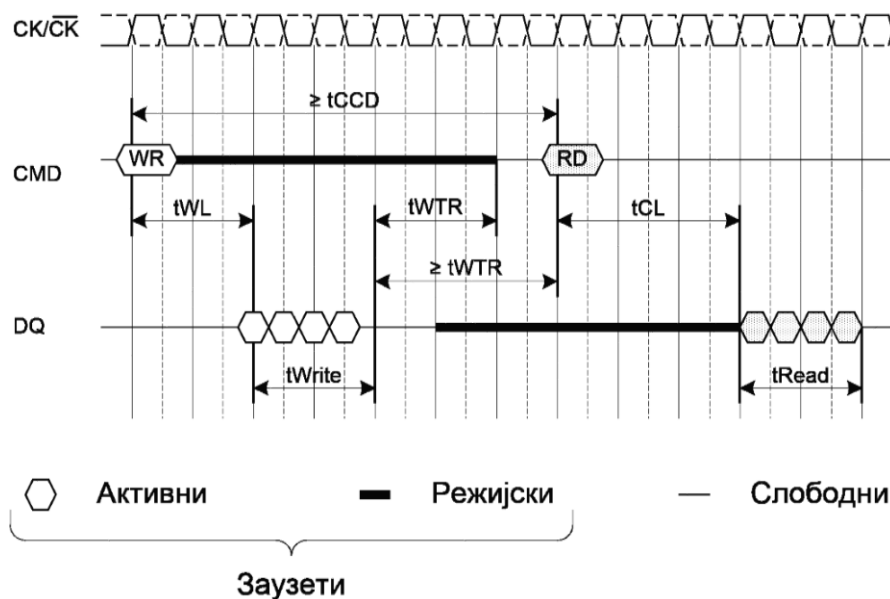
Кључни проблем у анализи перформанси DRAM меморија је неопсервабилност стања свих циклуса на меморијској магистрали. Протокол за комуникацију између DRAM меморије и остатка система не дефинише сигнале на основу којих се може непосредно утврдити да ли су поједини циклуси на меморијској магистрали слободни или заузети. Неки циклуси током којих сигнали на меморијској магистрали нису активни могу да изгледају као да су слободни, али да су, према DRAM протоколу, инхерентно недоступни за коришћење. Такви циклуси се називају режијски (енг. overhead) циклуси и не могу се разликовати од истински слободних циклуса на магистрали (Слика 1.6). Пошто режијски циклуси могу представљати значајан део укупног броја циклуса на меморијског магистрали, њихов утицај на тачност мерења перформанси може бити веома велики.

1.5.1 Фундаментални проблеми

Немогућност да се меморијски циклуси непосредно и једнозначно окарактеришу као слободни или заузети ствара проблем са мерењем чак и основних показатеља меморијских перформанси. То се пре свега односи на степен искоришћења, који се дефинише као однос броја заузетих циклуса и укупног броја циклуса, и ефикасност, која се дефинише као однос броја активних циклуса и заузетих циклуса. Као последица тога јавља се дилема како одговорити на следеће фундаментално питање:

Како мерити перформансе DRAM меморије са потребним нивоом тачности?

Чак и за оне показатеље меморијских перформанси које је могуће непосредно мерити, као што је нпр. проток података, који се дефинише као количина



Слика 1.6 – Карактеризација циклуса на меморијској магистралу

Меморијска магистрала има три главне компоненте: командну магистралу (CMD), магистралу података (DQ) и адресну магистралу (није приказана на слици). Слика илуструје случај када операција читања (RD) следи операцију уписа (WR) приликом приступа истој банки меморије. Минимални број циклуса који мора протећи између команде уписа и читања на командној магистралу или између појаве одговарајућих циклуса преноса на магистралу података јесу режијски циклуси. И режијски и слободни циклуси нису активни, па се не могу међусобно разликовати. На слици су приказана и различита временска ограничења која прописује DDR2 SDRAM спецификација.

података која се пренесе у јединици времена, постоји проблем интерпретације измерених вредности. Због природе DRAM протокола и динамике система, теоријски максимуми показатеља меморијских перформанси се континуирано мењају. На пример, максимум протока на меморијској магистралу ограничен је бројем циклуса који се могу искористити за пренос података у неком временском интервалу, што укључује све не-режијске циклусе из датог интервала. Пошто број режијских и не-режијских циклуса зависи од радног оптерећења, максимални проток варира у времену. Зато и ако би се исти проток података измерио у различитим временским периодима или за различита радна оптерећења, однос стварног и максималног протока, тј. степен искоришћења, у тим случајевима може да буде различит. Имајући у виду да се теоријски максимуми показатеља

меморијских перформанси не могу непосредно мерити, поставља се питање да ли се и како може користити поређење измерених и максималних вредности параметара перформанси ради интерпретације резултата. У том смислу, намеће се дилема како одговорити на следећа кључна питања:

Колико су измерене перформансе DRAM меморије добре или лоше?

Како поредити перформансе DRAM меморије измерене у различитим периодима?

Како поредити перформансе DRAM меморије за различита радна оптерећења?

1.5.2 Применљивост на командну магистралу и магистралу података

У постојећој пракси мерење перформанси DRAM меморија се најчешће своди на бројање трансакција на магистралу података (нпр. рачунање протока података бројањем активних циклуса у датом интервалу), док се анализа перформанси најчешће своди на проучавање трансакција на командној магистралу (нпр. проучавање обрасца читања и уписа у меморију), при чему се временска компонента или не узима у обзир или се узима у обзир само апроксимативно. Разлог за овакав приступ је непостојање довољно доброг модела перформанси DRAM меморије којим би се трансакције на обе магистрале узрочно-последично и аналитички повезале.

Последице приступа где се мерење и анализа перформанси DRAM меморије ради за сваку магистралу посебно су двоструко негативне. Најпре, лако се могу занемарити инхерентне везе које постоје између трансакција на обе магистрале, чиме се систематски смањује тачност анализе, као на пример, ако се не детектују режијски циклуси на магистралу података изазивани неким ограничењима везаним за командну магистралу или обрнуто. Са друге стране, анализа трансакција на командној магистралу се најчешће обавља ручно, апроксимативно или хеуристички, што је често неадекватно или чак потпуно неприменљиво у анализи система који имају већи број меморијских клијената, комплексне адресне обрасце или дуге адресне секвенце. Системска анализа данас захтева знатно робустнији и тачнији приступ анализи перформанси DRAM меморија, који ће истовремено обухватити трансакције на командној магистралу и магистралу података и обезбедити аутоматизацију процеса мерења и анализе перформанси.

1.5.3 Примена на различита окружења, моделе система и радна оптерећења

У разним фазама развоја анализа перформанси се ради у различитим окружењима и на различитим моделима рачунарског система. Тако се у фази дефинисања архитектуре анализа ради на архитектуралном, обично аналитичком или симулационом, моделу система, у фази имплементације анализа се ради на нивоу RTL модела, у фази верификације анализа се обично ради на FPGA прототипу система, а у фази тестирања анализа се ради на одговарајућој платформи за испитивање рада интегрисаних кола. Пожељно је да се перформансе DRAM меморије у свим окружењима и на свим моделима рачунарског система мере на методолошки исти начин како би резултати добијени на различитим платформама, за различита радна оптерећења или у различитим временским периодима могли међусобно да се пореде. Постојеће методологије углавном не могу да задовоље ове захтеве, што може знатно да отежа анализу.

1.5.4 Цена, комплексност и ефикасност у примени

Код методологија за анализу перформанси DRAM меморија посебан проблем представља потреба да се постигне ефикасност и једноставност у примени коју изискује рад на сложеним пројектима. Методологије које захтевају знатне ресурсе, имају велико време извршавања или не гарантују довољну тачност могу драстично утицати на применљивост у пракси. Поред тога, поставља се и питање цене имплементације, као и потребних услова за њихову примену. На пример, методологије базиране на методи симулације захтевају увид у интерну архитектуру меморијског подсистема на коме се извршава радно оптерећење, што некада није могуће било из правних, било из техничких разлога. Такође, верификација алата примењених у оквиру неке методологије може представљати изазов било због њихове комплексности, било због потребних ресурса.

1.5.5 Неадекватност постојећих методологија

Наведени проблеми у вези са мерењем и анализом перформанси DRAM меморија немају адекватно решење. Тако се у пракси данас најчешће срећу два приступа. Један приступ се базира на увођењу различитих претпоставки за поједностављено моделовање рада DRAM меморија. То у неким једноставнијим

случајевима омогућава добијање употребљивих, али никада потпуно тачних резултата. Међутим, као што је раније поменуто, виртуелизација и велики број меморијских клијената у модерним рачунарским системима чине да меморијски саобраћај постаје веома непредвидљив. У таквој ситуацији се понашање DRAM меморије више не може поуздано апроксимирати, тако да грешка мерења постаје неприхватљиво велика. Други приступ се базира на примени методе симулације. У оквиру тог приступа, карактеризација меморијских циклуса на слободне и заузете се врши на основу интерних сигнала модела меморијског контролера који се користи за симулациону анализу. Мане овог приступа у пракси су вишеструке. Он захтева креирање тачног модела меморијског подсистема, као и одговарајуће симулационо окружење. Ако модел у свим елементима не одговара стварном систему, резултујући меморијски саобраћај неће бити веродостојан, па није могуће поуздано мерити ни анализирати перформансе меморије. Такође, генерисање одговарајућег радног оптерећења је често непремостив проблем у пракси, јер се утицаји различитих елемената система, као што је рад оперативног система и симултани рад различитих компоненти, тешко моделирају. Мана је и што, с обзиром да се потребни резултати генеришу динамички у току симулације, анализа перформанси сваке меморијске секвенце захтева ново покретање симулације. То често има неприхватљиву цену узимајући у обзир дужину трајања симулација и ресурсе који се при том ангажују. Коначно, метода симулације сама по себи не омогућава решавање проблема интерпретације добијених резултата, јер не нуди решење за рачунање теоријског максимума.

1.6 Предлог решења

Пошто постојеће методе за мерење и анализу перформанси DRAM меморија не могу да у потпуности задовоље потребе у вези са претходно наведеним критеријумима, постаје нужно да се тај изазов на неки начин превазиђе. Главни циљ ове дисертације је управо проналажење решења за постојеће проблеме и формулација одговарајуће методологије која би дефинисала како да се мерење и анализа спроводе у пракси.

Предложена методологија се заснива на новом моделу за тачно мерење и анализу перформанси DRAM меморија названом ACDC. ACDC модел омогућава прецизну карактеризацију меморијских циклуса на основу утицаја које функционалне зависности и временска ограничења имају на догађаје на меморијској магистралаи. ACDC модел за одређени тип DRAM меморије се може креирати на основу спецификације DRAM уређаја, као што су на пример JEDEC³ DDRx, LPDDRx или GDDRx стандарди, користећи добро дефинисан процес применљив у пракси. Модел има облик коначног аутомата који је параметризован тако да може да подржи све врсте DRAM уређаја, различите конфигурације и вредности временских параметара. Улазне податке аутомата чине DRAM команде и време њихове појаве на меморијској магистралаи, а излазне вредности чине циклуси на меморијској магистралаи једнозначно окарактерисани као слободни, активни или режијски. Поред тога што у потпуности решава проблем тачности, ACDC модел по својој природи карактерише ниска комплексност, мала цена развоја и преносивост на различите платформе.

Пошто се креира на основу DRAM спецификације, ACDC модел се може непосредно применити у анализи реалних DRAM трансакција. Модел не зависи од порекла DRAM трансакција или меморијске архитектуре изворног система, тако да се може користити за обраду трансакција из различитих извора, као што су оне снимљене коришћењем уређаја за аквизицију података са меморијске магистрале (енг. DRAM probe), сигнала снимљених током симулација (енг. waveforms), или текстуалних фајлова са догађајима које су забележили монитори током симулација (енг. event logs), што је веома тешко или немогуће обезбедити коришћењем постојећих методологија. Ове јединствене особине ACDC модела омогућавају његову универзалну применљивост без обзира на радно окружење, модел анализираног система и радно оптерећење. Коначно, ACDC модел чини непотребним извршавање функционалних симулација ради анализе перформанси

³ JEDEC (енг. Joint Electron Device Engineering Council) је инжињерска организација за стандардизацију чији је циљ да усваја отворене стандарде који олакшавају интероперабилност различитих електронских компоненти.

DRAM меморије, што значајно повећава ефикасност и поједностављује употребу у процесу анализе.

Тачна карактеризација појединачних меморијских циклуса коју обезбеђује ACDC модел омогућава формулисање сасвим нове метрике за мерење перформанси DRAM меморије. Ова метрика на природан начин интегрише перформансе магистрале података и командне магистрале, као две кључне компоненте меморијске магистрале. На тај начин се превазилази дихотомија која постоји код других методологија. На основу нове метрике могуће је дефинисати и метод за тачну динамичку процену максимума перформанси DRAM меморије, чиме се у потпуности решава и проблем интерпретације измерених вредности.

ACDC модел моделира перформансе DRAM уређаја, односно на нивоу једног DRAM интегрисаног кола. Детаљи који се односе на организацију читаве DRAM меморије и одговарајућих делова меморијског подсистема, као што су број и конфигурација меморијских канала, број и тип меморијских модула, број DRAM чипова по модулу, број интегрисаних кола у чипу и др., могу се апстраховати на вишем нивоу унутар одговарајућег модела за мерење и анализу перформанси DRAM меморије, у оквиру ког се ACDC модел може користити као основна компонента у потребном броју инстанци.

Имајући у виду особине предложене методологије, очекује се да она омогући квалитативни и квантитативни напредак у односу на постојећа решења, са циљем да их замени у свим доменима где су тачност и ефикасност од примарног значаја, као и да отвори могућност за примену у новим областима где то до сада није било могуће.

1.7 Терминологија

Неке кључне појмове, као на пример „DRAM меморијска архитектура“, је тешко разумети без прецизнијег објашњења, док неки други, као на пример „DRAM“, могу имати различито значење у зависности од контекста у ком се

користе. Све то отежава разумевање текста и може довести до забуне. Стога је корисно да се разјасни значење и употреба тих и других релевантних појмова.

Појмови „RAM“, „главна меморија“, „системска меморија“ и „физичка меморија“ представљају синониме који се односе на меморију у којој се чувају програми и подаци у складу са Фон Нојмановом или Харвард архитектуром. Ови појмови се не односе на део меморијске хијерархије који се уводи ради убрзавања приступа главној меморији, односно кешеве, нити на виртуелну меморију, као проширење концепта RAM-а на секундарну меморију.

Термин „DRAM меморија“ означава (RAM) меморију имплементирану помоћу DRAM чипова. Под DRAM чипом се подразумева полупроводнички уређај који у себи садржи интегрисано коло са DRAM меморијским ћелијама и контролном логиком за приступ њиховом садржају у оквиру истог паковања. У ширем контексту, термин „чип“ означава полупроводнички уређај који се састоји од једног или више интегрисаних кола у оквиру истог паковања. Под интегрисаним колом подразумева се електронски уређај са полупроводничким компонентама на једном комаду силицијумског субстрата (енг. silicon die). Некада се појам „интегрисано коло“ користи и као синоним за чип, али то може довести до забуне и може имати оправдање једино ако чип садржи само једно интегрисано коло унутар паковања. Термин „полупроводнички уређај“ може означавати било какав уређај са полупроводничким компонентама, нпр. чип, интегрисано коло, дискретне полупроводничке компоненте и сл.

Појам „DRAM меморијска архитектура“ подразумева контролну логику унутар DRAM чипова, интерфејс за повезивање чипова са остатком система и одговарајући протокол за комуникацију. DRAM чипови су повезани са остатком система преко меморијске магистрале. Интерфејс преко кога су DRAM чипови и меморијски контролер повезани на меморијску магистралу се назива „DRAM интерфејс“ или „меморијски интерфејс“. За комуникациони протокол који се употребљава приликом размене података преко меморијског интерфејса се користи термин „DRAM протокол“.

Појам „DRAM“ је веома оптерећен. У свом основном значењу он представља врсту непостојане, динамичке меморије са једнотранзисторским меморијским ћелијама. У контексту архитектуре рачунарског система, термин „DRAM“ се често употребљава као синоним за DRAM меморију. Притом се неретко имплицира чак и DRAM меморијска архитектура примењена у имплементацији; на пример, пошто се асинхроне DRAM меморије не користе у савременим рачунарским системима, „DRAM“ може да представља синоним за синхрону DRAM меморију (SDRAM). У контексту имплементације DRAM меморије, термин „DRAM“ се користи да означи DRAM чип имплементиран у складу са одређеним индустријским стандардом, као што је на пример DDR2 SDRAM стандард.

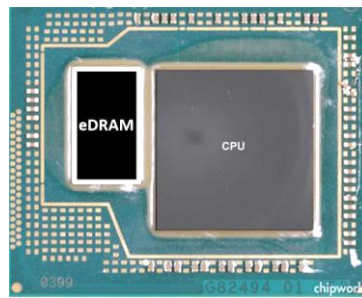
2 DRAM меморија

У оквиру овог поглавља се описује физичка имплементација, технологија израде, принцип функционисања, структура, дизајн, архитектура и организација DRAM меморије. Немогуће је направити потпун преглед ових области у оквиру само једног поглавља. Стога је циљ пре свега да се обезбеде информације са потребним нивоом детаља о раду DRAM меморија како би се могли разумети и решавати проблеми у вези са мерењем и анализом њених перформанси са теоријског и практичног аспекта о којима ће бити речи у наредним поглављима.

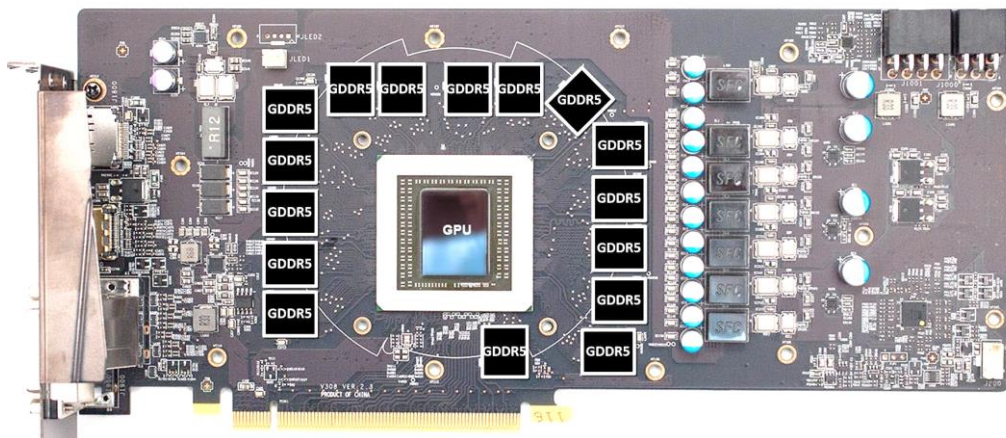
2.1 DRAM меморија у рачунарским уређајима

Пошто се DRAM меморија и дигитална логичка кола производе коришћењем различитих технолошких процеса, DRAM није могуће имплементирати на истој полупроводничкој подлози са процесором. Из тога произилази да је DRAM интегрисано коло физички најмања целина која се може користити за имплементацију DRAM меморије у рачунарским уређајима.

Најтешњи ниво повезивања DRAM интегрисаних кола са остатком система јесте у оквиру истог паковања (Слика 2.1.a). У том случају се за одговарајуће DRAM интегрисано коло користи назив eDRAM (енг. embedded DRAM). Оваква имплементација се обично користи када се DRAM користи као нови, највиши, ниво кеша у меморијској хијерархији. Таква решења се срећу код система који захтевају веома високе перформансе, као што су играчке конзоле, процесори за HPC који имплементирају четврти ниво кеша и сл.



а) Intel Haswell процесор (128MB eDRAM)



б) AMD Radeon R9 390X графичка карта (16x512Mb GDDR5 SDRAM)



в) Supermicro X9DRW-CTF31 сервер (16x32GB ECC RDIMM)

Слика 2.1 – DRAM меморија у рачунарским уређајима

DRAM меморија се интегрисе у систем у три облика: а) eDRAM интегрисано коло у истом паковању, б) DRAM чипови на истој плочи, в) DIMM меморијски модул

Следећи ниво интеграције је када се једно или више DRAM интегрисаних кола смешта у засебно паковање – меморијски чип. DRAM чипови се могу повезати са остатком система лемљењем на истој матичној плочи, што је приступ који се користи код уређаја код којих не постоји потреба да клијент мења капацитет или перформансе меморије. То је, на пример, чест случај код носивих и мобилних уређаја, као што су паметни сатови, мобилни телефони и таблети, или код акцелератора, као што су графичке карте (Слика 2.1.б). Овај приступ је јефтинији, нуди боље перформансе и захтева мање простора од коришћења меморијских модула, али нема флексибилност.

Меморијски модули су штампане плочице на којима се налази више DRAM чипова повезаних у целину. Формат модула одговара физичким, електричним и другим карактеристикама прописаним одговарајућим стандардима, чиме се омогућава коришћење меморија разних произвођача. Меморијски модули се користе у уређајима код којих је битно да клијент може да одређује врсту и капацитет меморије. То је чест случај код рачунарских система као што су персонални рачунари, сервери и суперрачунари (Слика 2.1.в). Ова могућност је посебно важна за систем интеграторе који имају различите линије производа и више добављача.

У пракси се користе два типа модула: DIMM (енг. Dual In-line Memory Module) и SODIMM (енг. Small Outline DIMM). DIMM модули су намењени уређајима стандардних димензија, док су SODIMM модули намењени уређајима код којих је минимизација заузетог простора од великог значаја, као што су лаптоп рачунари, blade сервери и др.

Меморијски модули могу имплементирати и одређене специјалне функције које индивидуални DRAM чипови немају, као што је детекција и корекција грешака (енг. Error Code Correction) и баферовање сигнала са меморијске магистрале. ECC функција изискује уградњу додатног DRAM чипа који служи за чување кодних речи. За кодирање се обично користи модификован (72,64) Хеминговог код на основу ког се могу исправити све једнобитне и детектовати све двобитне грешке. ECC DIMM модули се често користе код серверских платформи због веће поузданости.

Због пораста електричног оптерећења адресних и командних линија које уноси сваки модул који се повезује на меморијску магистралу, истовремено расте и кашњење сигнала, па је број DIMM модула у пракси ограничен, најчешће на четири. То ограничење је критично код серверских плаформи, јер код њих постоји потреба за већом количином меморије од оне коју омогућава стандардни број модула. Проблем се решава тако што се на сваки модул додаје чип који служи да региструје адресе и команде за сваку трансакцију, а они се онда локално разводе до свих меморијских чипова у модулу. Тиме се оптерећење адресних и командних линија меморијске магистрале вишеструко смањује и на тај начин омогућава повезивање већег броја модула. Модули који подржавају овај механизам се зову регистарски модули или RDIMM (енг. Registered DIMM), а они који га не подржавају нерегистарски модули или UDIMM (енг. Unregistered DIMM). Постоје и модули који баферују све сигнале на меморијској магистрали и шаљу их локално до свих меморијских чипова и обратно, а при том обнављају и сигнал такта. Такви модули се називају LRDIMM (енг. Load Reduced DIMM).

2.2 Структура DRAM меморије

DRAM меморија има вишеслојну структуру. Основни елемент DRAM меморије је DRAM меморијска ћелија. Низови меморијских ћелија су организовани у густо паковане дводимензионалне матрице чија оптимална величина зависи од технолошких параметара, жељених перформанси и потрошње. Више меморијских матрица са одговарајућим колима за декодирање адресе и аналогним колима за приступ садржају меморијских ћелија се групише у меморијске банке. Више меморијских банки са контролном логиком за приступ и синхронизацију се имплементира у оквиру једног физичког DRAM интегрисаног кола. Једно или више DRAM интегрисаних кола се може повезати у оквиру истог паковања у засебне чипове. Више чипова се може повезати на шампаној плочици у меморијске модуле.

2.2.1 Меморијска ћелија

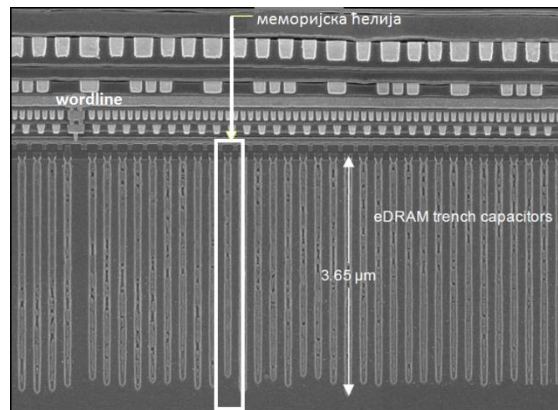
На Слици 2.3 представљена је шема DRAM меморијске ћелије у оквиру меморијске матрице. Информација се унутар меморијске ћелије складишти у виду

наелектрисања на меморијском кондензатору, док се приступ ускладиштеној информацији врши преко приступног транзистора. Када је напон на меморијском кондензатору једнак напону напајања, то представља стање логичке јединице, а када је напон на кондензатору једнак напону уземљења, то представља стање логичке нуле. С обзиром да се памте два стања, количина информације коју складишти једна DRAM меморијска ћелија је један бит⁴.

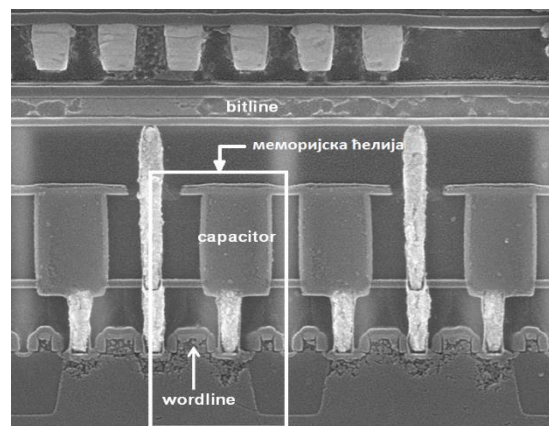
Када је приступни транзистор искључен, тј. напон на линији речи WL (енг. word line) је V_{GND} , кондензатор је електрично изолован и чува уписану информацију. Међутим, с обзиром да приступни транзистор није идеалан прекидач, изолација није потпуна, тако да наелектрисање на меморијском кондензатору није постојано у дужем периоду. Струје цурења приступног транзистора могу да доведу до лаганог пражњења меморијског кондензатора и промене његовог стања, односно да доведу до губитка информације. Да би се информација очувала, потребно је да се наелектрисање повремено освежава. То је разлог због ког се ова врста меморије назива динамичка.

Када је приступни транзистор укључен, меморијски кондензатор се електрично повезује са линијом података BL (енг. bit line). Напон на линији WL којим се активира врста (V_{WL}) мора бити такав да напон на меморијском кондензатору приликом уписа може достићи V_{DD} , што значи да је $V_{WL} > V_{DD} + |V_{TN}|$, где је V_{TN} напон прага NMOS транзистора. Упис се врши тако што се вредност напона на линији BL постави на одговарајућу вредност (на V_{DD} за упис логичке јединице или V_{GND} за упис логичке нуле), чиме се ажурира напон на меморијском кондензатору. Ако се жели извршити читање, онда линију BL треба држати електрично изолованом и измерити промену напона на линији BL која настаје након електричног повезивања са меморијским кондензатором. Ако је промена позитивна, онда се читава логичка јединица, а ако је негативна, онда се читава логичка нула.

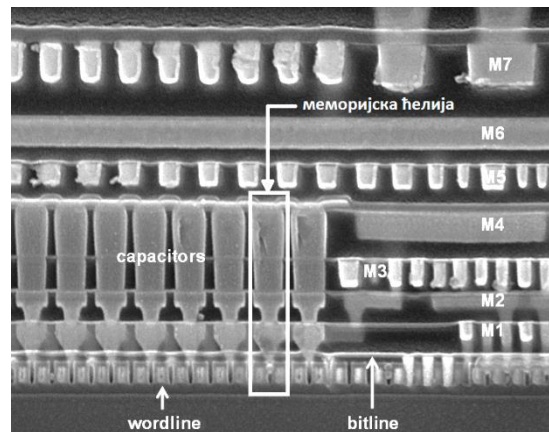
⁴ Код других типова меморијских ћелија, као на пример код ћелија флеш меморије, могуће је складиштити више нивоа напона у оквиру једне ћелије, чиме се постиже да једна ћелија памти више битова информације (нпр. 4 нивоа → 2 бита, 8 нивоа → 3 бита).



а) технологија са укупаним кондензатором



б) технологија са издигнутим кондензатором (CUB)



в) технологија са издигнутим кондензатором (COB)

Слика 2.2 – DRAM ћелија

SEM (Scanning Electronic Microscope) снимци попречног пресека DRAM чипова:

а) Embedded DRAM у IBM Power 7+ процесору (32nm)

б) Embedded DRAM у Microsoft Xbox GPU произведеном од стране TSMC (65nm)

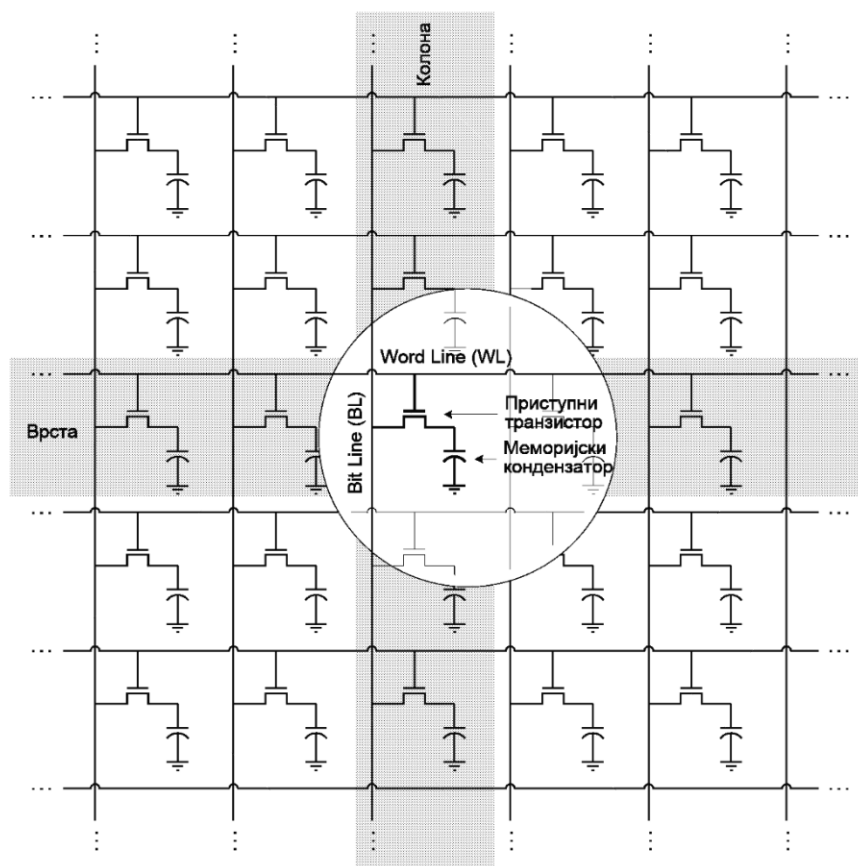
в) Embedded DRAM у Intel Haswell процесору (22nm)

Извор: “Intel’s e-DRAM Shows up in the Wild”, Chipworks, <http://chipworksrealchips.blogspot.com/2014/02/intels-e-dram-shows-up-in-wild.html>, February 2014

Према технологији израде постоје два типа DRAM ћелија: ћелије са укопаним кондензатором (енг. trench capacitor) и ћелије са издигнутим кондензатором (енг. stacked capacitor). Постоје два типа ћелија са издигнутим кондензатором: са кондензатором испод линије података, CUB (енг. Capacitor Under Bitline) и са кондензатором изнад линије података, COB (енг. Capacitor Over Bitline) (Слика 2.2).

2.2.2 Меморијска матрица

Меморијске ћелије су распоређене у матрицу са одређеним бројем врста и колоне. Свака ћелија се налази на пресеку једне врсте и једне колоне (Слика 2.3). Активирањем линије WL укључују се сви приступни транзистори у оквиру врсте,



Слика 2.3 – DRAM меморијска матрица

Меморијске ћелије су организоване у облику матрице чији се елементи адресирају активирањем сигнала за одговарајућу врсту преко линије речи (WL) и колону преко линије података (BL).

повезујући тако кондензаторе из припадајућих меморијских ћелија на линије BL у колонама. Само једна врста сме бити активна у било ком тренутку како не би дошло до мешања наелектрисања меморијских кондензатора из различитих врста на линијама колоне. С обзиром да су колоне међусобно независне, меморијским ћелијама у активираној врсти се може приступати истовремено.

Величина меморијске матрице је одређена са два фактора. Први фактор се односи на потребу да број меморијских ћелија у врстама и колонама буде степен броја два, а други фактор се тиче физичких ограничења који утичу на перформансе и потрошњу.

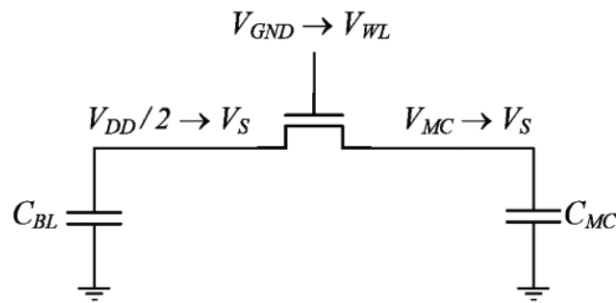
Потреба да број врста и колоне буде степен броја два проистиче из бинарне природе адресирања код DRAM меморија. У меморијској матрици која има 2^{M+N} меморијских ћелија, потребно је $M + N$ битова бинарне адресе да би могло да се изврши једнозначно адресирање сваке ћелије. Матрица се у том случају може организовати у $m = 2^M$ врста и $n = 2^N$ колоне тако да се M битова може узети директно из адресе како би се декодовала врста, а преосталих N битова како би се декодовала колоне. Познато је да су декодери адреса које су степен броја два далеко једноставнији и имају мање кашњење од декодера адреса које то нису. Пошто се декодери адреса налазе на критичним путањама код DRAM меморије, димензионисање меморијске матрице на овај начин повољно утиче на перформансе DRAM меморија.

Број колоне у меморијској матрици је ограничен временом које је потребно да се укључе сви транзистори у врсти како кашњење које настаје том приликом не би било превелико и утицало на укупне перформансе DRAM меморије. Време активирања врсте је одређено временском константом врсте $\tau = RC$, при чему је:

$$R = n \cdot R_S \cdot \frac{d_{BL}}{w_{WL}} \quad (2.1)$$

$$C = n \cdot C'_{WL} \quad (2.2)$$

где је R_S површинска отпорност, d_{BL} растојање између колоне, w_{WL} ширина линије WL и C'_{WL} капацитивност дела WL линије који одговара једној меморијској ћелији.



Слика 2.4 – Прерасподела наелектрисања код читања из DRAM меморијске ћелије

Пре читања, напон на линији BL чија је капацитивност C_{BL} се поставља на $V_{DD}/2$. Напон V_{MC} на меморијском кондензатору капацитивности C_{MC} је или V_{DD} (ако је меморисана логичка јединица) или V_{GND} (ако је меморисана логичка нула). Након активирања врсте (када се напон на линији WL промени са V_{GND} на V_{WL}), приступни транзистор се укључује и долази до прерасподеле наелектрисања између кондензатора, тако да ће напон на оба кондензатора постати исти (V_S).

Ако је $\max(\tau)$ максимално дозвољено кашњење, из једначина (2.1) и (2.2), уз чињеницу да је $n = 2^N$ ($N \in \mathbb{N}$), следи да је максималан број колона одређен са:

$$n \leq 2^{\left\lfloor \frac{1}{2} \log_2 \left(\frac{\max(\tau) \cdot w_{WL}}{R_S \cdot C'_{WL} \cdot d_{BL}} \right) \right\rfloor} \quad (2.3)$$

Максималан број врста у меморијској матрици је ограничен са два фактора. Први фактор се тиче капацитивности линије BL. Приликом читања из меморијске ћелије долази до прерасподеле наелектрисања између линије BL капацитивности C_{BL} и меморијског кондензатора капацитивности C_{MC} (Слика 2.4). Једначине стања пре и после прерасподеле су, респективно:

$$Q_{BL} = \frac{V_{DD}}{2} \cdot C_{BL}, \quad Q_{MC} = V_{MC} \cdot C_{MC} \quad (2.4)$$

$$Q_{BL} + Q_{MC} = V_S \cdot C_{BL} + V_S \cdot C_{MC} \quad (2.5)$$

Решавањем ових једначина се добија да је промена напона на линији BL у односу на референтни напон диференцијалног појачавача $V_{DD}/2$:

$$\Delta V = V_S - \frac{V_{DD}}{2} = \left(V_{MC} - \frac{V_{DD}}{2} \right) \cdot \frac{C_{MC}}{C_{BL} + C_{MC}} \quad (2.6)$$

односно, имајући у виду да је $V_{MC} = V_{GND}$ или $V_{MC} = V_{DD}$:

$$|\Delta V| = \frac{V_{DD}/2}{1 + C_{BL}/C_{MC}} \quad (2.7)$$

Промена напона $|\Delta V|$ која настаје приликом читања из меморијске ћелије мора да буде довољно велика да се може детектовати од стране диференцијалног појачавача на излазу⁵. Из тог услова и једначине (2.7) се одређује граница дозвољене капацитивност линије BL:

$$\max(C_{BL}) = C_{MC} \cdot \left(\frac{V_{DD}/2}{\min |\Delta V|} - 1 \right) \quad (2.8)$$

Капацитивност C_{BL} расте са додатком сваке нове меморијске ћелије у колони. Ако је C'_{BL} капацитивност дела BL линије који одговара једној меморијској ћелији, онда је:

$$C_{BL} = (m - 1) \cdot C'_{BL} \quad (2.9)$$

Из (2.8) и (2.9) се добија да је максималан број врста одређен са:

$$m \leq 1 + \frac{C_{MC}}{C'_{BL}} \cdot \left(\frac{V_{DD}/2}{\min |\Delta V|} - 1 \right) \quad (2.10)$$

Други фактор који ограничава број врста се тиче енергије која је потребна за пуњење и пражњење капацитивности колона током активног периода. Снага која се дисипира у току периода T је пропорционална укупној капацитивности свих колона:

$$P = \frac{V_{DD}^2 \cdot n \cdot (C_{MC} + C_{BL})}{2T} \quad (2.11)$$

⁵ Извођење промене напона на улазу диференцијалног појачавача када је референтни напон једнак напону на некој неактивној ћелији је дато у [5]. Референтни напон је био формиран на овај начин у старијим генерацијама DRAM меморија, али такав приступ има недостатке у односу на новија решења где је референтни напон константан и једнак $V_{DD}/2$, као што је претпостављено у једначинама (2.4)-(2.10).

Ова снага не сме прећи вредност одређену максималном дозвољеном струјом (I_{MAX}):

$$P \leq V_{DD} \cdot I_{MAX} \quad (2.12)$$

Сада се из једначина (2.9), (2.11) и (2.12) може наћи максималан број врста према овом критеријуму:

$$m \leq 1 + \frac{C_{MC}}{C'_{BL}} \cdot \left(\frac{2T \cdot I_{MAX}}{n \cdot V_{DD} \cdot C_{MC}} - 1 \right) \quad (2.13)$$

Из израза се може приметити да максималан број врста није независан од броја колона када је ограничавајући фактор утрошена енергија. Имајући истовремено у виду једначине (2.10) и (2.13) и чињеницу да је $m = 2^M$ ($M \in \mathbb{N}$), израз за максималан број врста се може представити јединственим изразом:

$$m \leq 2^{\left\lfloor \log_2 \left(1 + \frac{C_{MC}}{C'_{BL}} \cdot \left[\min \left\{ \frac{V_{DD}/2}{\min|\Delta V|}, \frac{2T \cdot I_{MAX}}{n \cdot V_{DD} \cdot C_{MC}} \right\} - 1 \right] \right) \right\rfloor} \quad (2.14)$$

2.2.3 Аналогна кола за приступ меморији

Приступ подацима који се чувају у меморијској матрици се обавља помоћу одговарајућих аналогних кола и то у више фаза (Слика 2.5). Код читања, најпре се уз помоћ диференцијалног појачавача (енг. sense amplifier⁶) вредност промене напона која настаје на линији података приликом прерасподеле наелектрисања између ње и меморијске ћелије појачава до потребног нивоа, тј. до нивоа логичке нуле или логичке јединице. Потом се помоћу селектора колоне⁷ (енг. column selector) пропуштају сигнали са одговарајућих линија података меморијске матрице из којих се жели обавити читање. На крају се селектовани сигнали помоћу излазних

⁶ Из историјских разлога се користи овај назив за диференцијални појачавач (енг. differential amplifier) који служи за појачавање читаног сигнала из меморије.

⁷ На Сlici 2.4 су врсте и колоне представљене ротирано у односу на Сliku 2.3 због боље прегледности, али су ознаке врста и колона конзистентне.

драјвера (енг. output drivers) струјно појачавају и шаљу на магистралу података. Код уписа, ток сигнала је супротан. Сигнали са магистрале података се најпре појачавају преко улазних драјвера (енг. input drivers), а потом преко селектора колона доводе до диференцијалних појачавача и одатле преносе на линије података, чиме се ажурира садржај одговарајућих меморијских ћелија.

Промена напона која настаје на линији података BL приликом читања из меморијске ћелије је веома мала у односу на пун напонски опсег од логичке нуле (напон уземљења) до логичке јединице (напон напајања). Да би се ова промена претворила у регуларни дигитални сигнал, потребно је извршити напонско појачање детектованог сигнала. Ову функцију обавља диференцијални појачавач. Диференцијални појачавач ради тако што разлику напона на линији података (BL X) и напона на референтној линији података (BL Y) увећава до нивоа напона логичке јединице, ако је разлика позитивна, односно логичке нуле, ако је разлика негативна. Пре него што се активира одговарајућа врста у меморијској матрици, обе линије података (BL X и BL Y) морају бити доведене (енг. precharged) на исти референтни напон (V_{REF}), који је типично једнак половини напона напајања ($V_{DD}/2$). Довођење напона на линијама података које су повезане на диференцијални појачавач на исту референтну вредност се врши помоћу кола за изједначавање напона (енг. equilibration circuit).

Како би напон на линији података BL Y могао да се користи као референтни приликом читања из колоне BL X, он мора остати непромењен све време од момента када је извршено изједначавање напона. То је једино могуће ако колона BL Y све време буде неактивна, односно ако се не активира ниједна врста којој припадају ћелије колоне BL Y. Из тог разлога парови линија података који улазе у диференцијални појачавач морају бити из међусобно дисјунктних колона, тј. из колона којима не припадају исте врсте.

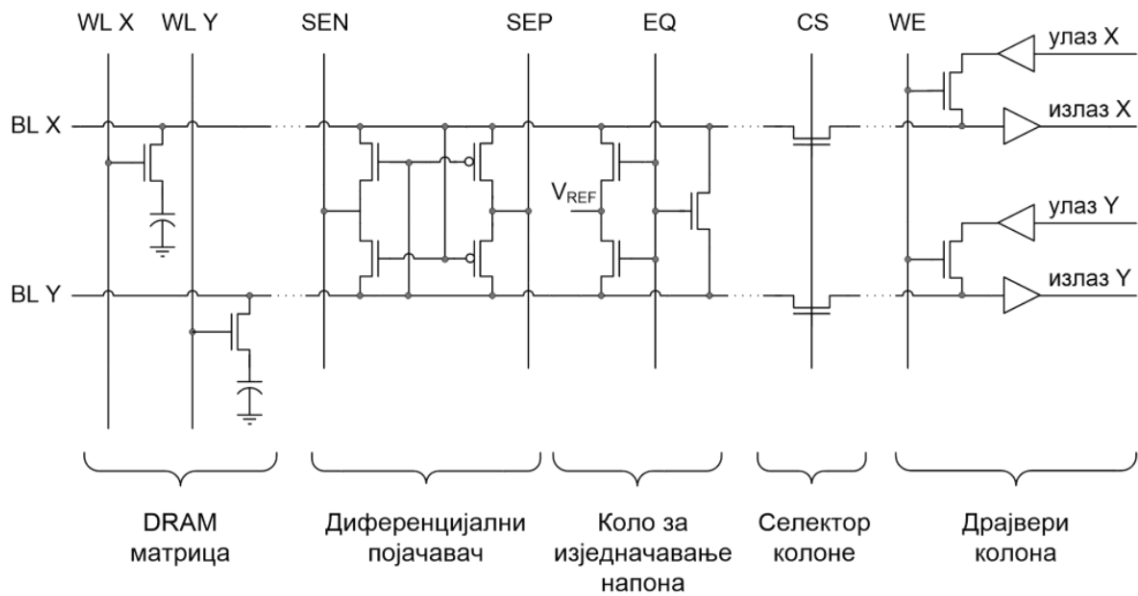
Читање из меморијске ћелије подразумева прераспodelу наелектрисања између меморијског кондензатора и линије података (поглавље 2.2.2). Тиме се мења напон на одговарајућем меморијском кондензатору (Слика 2.4) и доводи на ниво близак референтном напону. На тај начин се ефективно уништава ускладиштена информација и онемогућава поновно читање из те ћелије. Да би се тај проблем

решио, диференцијални појачавач мора да на неки начин обнови оригинално наелектрисање на кондензатору приликом читања. То је могуће спајањем излаза и улаза диференцијалног појачавача, чиме се са једне стране ствара позитивна повратна спрега која убрзава појачавачки процес (зато што се све више повећава разлика у односу на референтни напон), а са друге стране омогућава да појачани сигнал буде доведен и на улаз меморијског кондензатора чиме се његов напон обнавља на првобитни ниво.

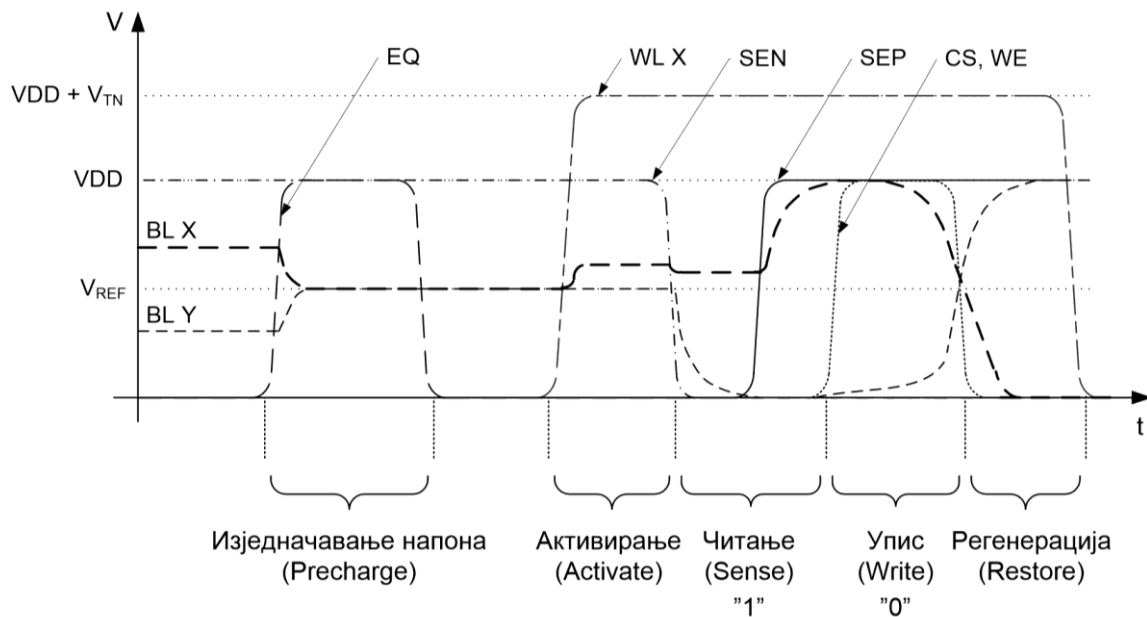
Стварањем позитивне повратне спреге код диференцијалног појачавача формира се структура са меморијским ефектом – лач (енг. latch). У примеру са Сликe 2.5 лач се може идентификовати у облику два унакрсно повезана CMOS инвертора. Као резултат тога, диференцијални појачавачи у DRAM меморијским матрицама уједно представљају и бафере података за активне врсте из којих се подаци могу узастопно читати без потребе да се понављају фазе активирања врсте, изједначавања напона, прерасподеле наелектрисања и регенерације. Код уписа, улаз и излаз диференцијалног појачавача мењају места. Улазни сигнал долази од селектора колоне, а излаз се шаље на линију података. У овом случају диференцијални појачавачи немају првенствено појачавачку улогу као код читања, јер су улазни сигнали претходно појачани, већ пре свега улогу бафера.

Селектори колона повезују диференцијалне појачаваче и одговарајуће улазно/излазне драјвере колона. Приликом активирања врсте (WL X), активирају се истовремено сви меморијски транзистори у тој врсти, којих има колико и колона у матрици (број колона је реда величине 2^7-2^{12}). Селектори колона омогућавају да се на магистралу података повежу само оне колоне којима се жели приступити (реда величине 2^2-2^7) на основу декодоване адресе.

Драјвери колона приликом читања служе као струјни појачавачи сигнала који се потом уписују у излазне бафере и шаљу на магистралу података. Драјвери колона који се користе приликом уписа се активирају одговарајућим сигналом (WE на Слици 2.5) и служе да појачају сигнал који долази из улазних бафера, у које су претходно уписане вредности са магистрале података, како би се побудио улаз диференцијалног појачавача. Више детаља о аналогним колима DRAM меморије се може наћи у [6], [7], [8], [9].



а) Аналогна кола за приступ меморијским ћелијама



б) Контролни сигнали кола за приступ меморијским ћелијама

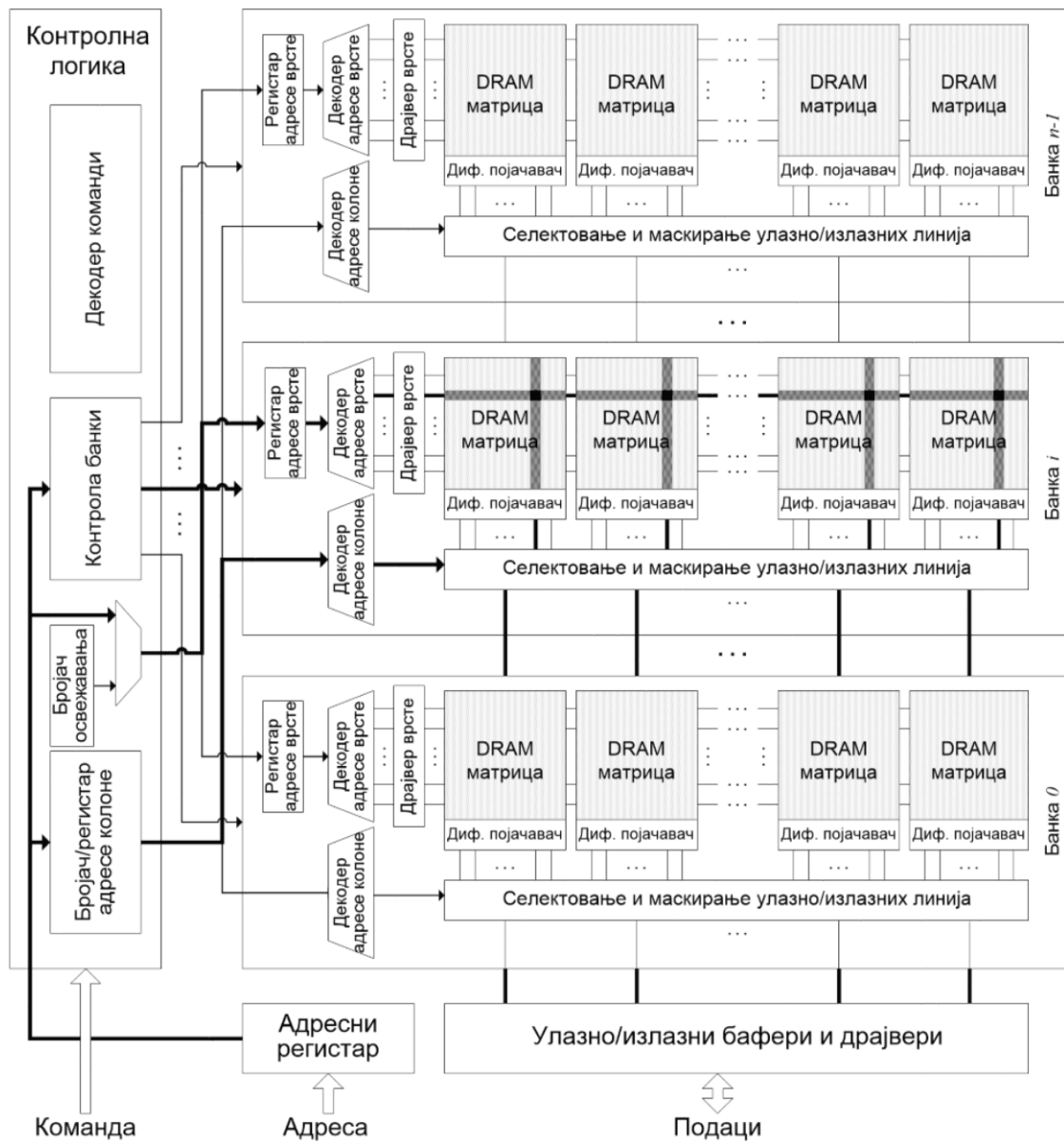
Слика 2.5 – Приступ DRAM ћелијама

Кола која омогућавају приступ ћелијама меморијске матрице: диференцијални појачавач и коло за изједначавање напона имају улогу да појачају очитани сигнал из меморијске ћелије и баферују вредности које се читају или уписују, селектор колоне омогућава да се пропусте само вредности за изабране колоне, а драјвери колоне појачавају сигнал према/са меморијске магистрале приликом читања и уписа. BL=Bit Line, WL=Word Line, SEN=Sense Enable for NMOS transistors, SEP=Sense Enable for PMOS transistors, EQ=Equilibrate, CS=Column Select, WE=Write Enable.

Напомена: На слици су врсте и колоне ротиране због боље прегледности.

2.2.4 Интегрисано коло

С обзиром да је величина DRAM меморијске матрице ограничена (поглавље 2.2.2), интегрисана кола већег капацитета се пројектују тако да садрже већи број DRAM меморијских матрица (Слика 2.6). DRAM интегрисана кола још садрже и



Слика 2.6 – DRAM интегрисано коло

На слици је представљена структура xDDR_x SDRAM интегрисаног кола. Подебљане стрелице означавају ток адресних сигнала, а подебљане линије ток података приликом читања или уписа.

контролна дигитална логичка кола, као и аналогна кола за приступ меморијским ћелијама и комуникацију са меморијском магистралом [9], [10], [11], [12].

Меморијске матрице се у модерним меморијским колима групишу у међусобно независне банке (енг. banks). Оваква организација омогућава смањење времена приступа подацима јер се неке операције, као на пример активирање врсте или читање и упис из и у меморијске ћелије могу обављати симултано унутар различитих банки. DRAM протокол осигурава да притом никада не дође до колизије између банки код приступа заједничким хардверским ресурсима. Коришћењем више меморијских матрица унутар једне банке повећава се и проточност података, јер се свим матрицама из банке може приступати истовремено. Број матрица у једној банки, означимо га са N , представља минималну грануларност приступа меморији у битима.

У једном циклусу се може приступити ка једном или више битова у свакој DRAM матрици. Ако претпоставимо да се свака банка састоји од N DRAM матрица и да се у свакој матрици истовремено чита или пише B битова, то даје укупно $N \cdot B$ битова по циклусу. С обзиром да се у једном циклусу може приступити само једној банки (јер се у току једног циклуса приступа подацима са једне адресе), онда сваки пут мора да се бира банка чијих $N \cdot B$ битова података ће бити повезано на улазно/излазне портове интегрисаног кола. Број улазно/излазних линија M на магистрали података интегрисаног кола, односно њена ширина, се обично у техничкој литератури означава са xM .

С обзиром да је брзина којом се може приступити подацима из меморијске матрице често мања од брзине меморијске магистрале, једини начин да се искористе сви циклуси на магистрали је да се дохвати већа количина података у једном меморијском циклусу, а њихово слање врши у више циклуса на меморијској магистрали – тзв. дохватање података унапред (енг. prefetch). Ако се у једном меморијском циклусу дохвати $N \cdot B$ битова из меморијске матрице, који се на магистралу података ширине M шаљу у P узастопних циклуса на магистрали, онда мора важити релација

$$N \cdot B = M \cdot P$$

2.2.5 Меморијски чип

У Табели 2.1 су дате вредности структурних параметара за различите типове DRAM меморијских чипова. Организација меморијских чипова се обично

Произвођач	Ознака чипа	Тип	Матрица		Банка		Инт. коло				Чип			
			# Врста	# Колона	# Матрица	# Банки	Капацитет	Prefetch	Ширина	# Инт. кола	Капацитет	Ширина	Дубина	Организација
Micron	MT42L256M16D1	LPDDR2	16384	2048	16	8	4 Gb	4	x16	1	4 Gb	x16	256 Mb	256Mx16
Micron	MT42L128M32D1	LPDDR2	16384	1024	32	8	4 Gb	4	x32	1	4 Gb	x32	128 Mb	128Mx32
Micron	MT42L128M64D2	LPDDR2	16384	1024	32	8	4 Gb	4	x32	2	8 Gb	x64	128 Mb	128Mx64
Micron	MT42L256M64D4	LPDDR2	16384	2048	16	8	4 Gb	4	x16	4	16 Gb	x64	256 Mb	256Mx64
Micron	MT42L512M32D4	LPDDR2	16384	2048	16	8	4 Gb	4	x16	4	16 Gb	x32	512 Mb	512Mx32
Micron	EDF8164A1MA	LPDDR3	16384	1024	32	8	4 Gb	8	x32	2	8 Gb	x64	128 Mb	128Mx64
Micron	EDFA164A1MA	LPDDR3	16384	2048	16	8	4 Gb	8	x16	4	16 Gb	x64	256 Mb	256Mx64
Hynix	H5GQ1H24AFR	GDDR5	4096	128	128	16	1 Gb	8	x16	1	1 Gb	x16	64 Mb	64Mx16
Hynix	H5GQ1H24AFR	GDDR5	4096	64	256	16	1 Gb	8	x32	1	1 Gb	x32	32 Mb	32Mx32
Micron	MT41J1G4	DDR3	65536	256	32	8	4 Gb	8	x4	1	4 Gb	x4	1 Gb	1Gx4
Micron	MT41J512M8	DDR3	65536	128	64	8	4 Gb	8	x8	1	4 Gb	x8	512 Mb	512Mx8
Micron	MT41J256M16	DDR3	32768	128	128	8	4 Gb	8	x16	1	4 Gb	x16	256 Mb	256Mx16
Micron	MT40A512M8	DDR4	32768	128	64	16	4 Gb	8	x8	1	4 Gb	x8	512 Mb	512Mx8
Samsung	K4A8G045WB	DDR4	65536	256	32	16	8 Gb	8	x4	1	8 Gb	x4	2 Gb	2Gx4
Samsung	K4A8G085WB	DDR4	65536	128	64	16	8 Gb	8	x8	1	8 Gb	x8	1 Gb	1Gx8

Табела 2.1 – Параметри комерцијалних DRAM чипова

Подаци потичу из јавно доступних произвођачких спецификација за наведене DRAM чипове [13], [14], [15], [16], [17], [18]. Сваки параметар се односи на компоненту назначену у секцији изнад; нпр. #Матрица се налази у секцији „Банка“ и представља број матрица у једној банки, Капацитет у секцији „Инт. коло“ означава капацитет једног интегрисаног кола, а Капацитет у секцији „Чип“ капацитет целог чипа.

Капацитет [Инт. коло] = #Врста · #Колона · #Матрица · #Банки

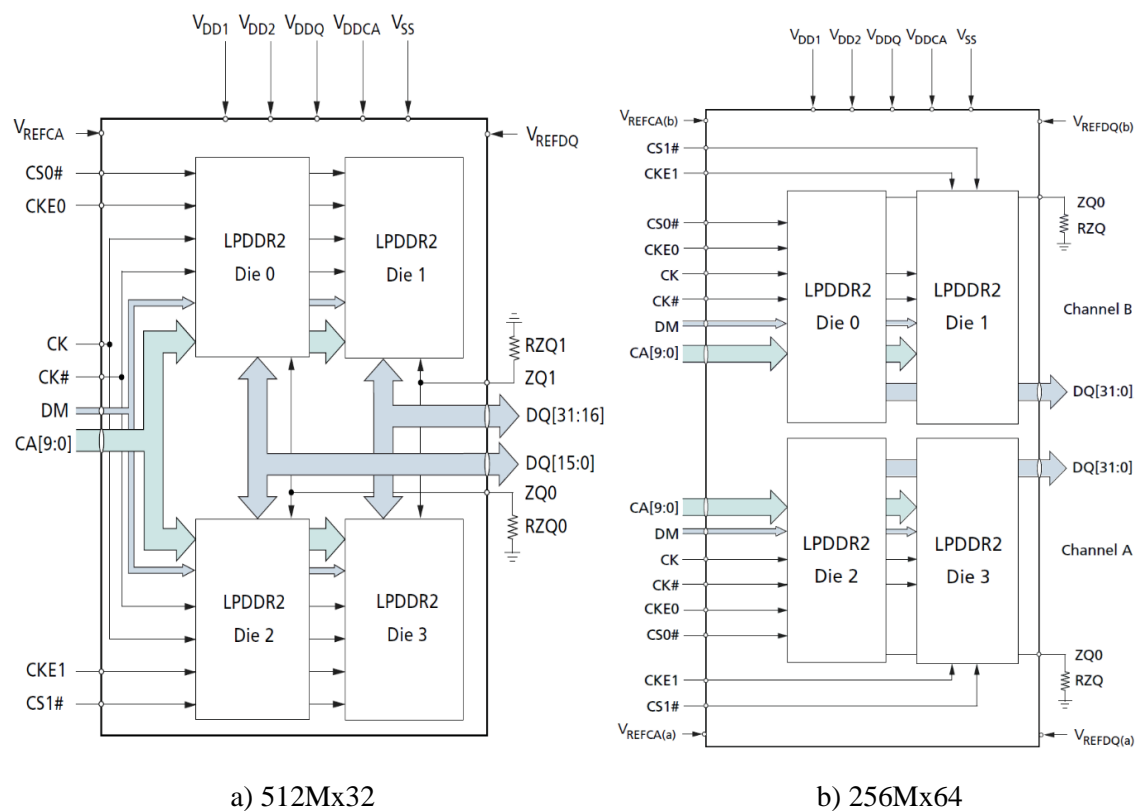
#Матрица · В (број битова који се читају из матрице) = Ширина [Инт. коло] · Prefetch

Капацитет [Чип] = Капацитет [Инт. коло] · #Инт. кола

Дубина [Чип] = Капацитет [Чип] / Ширина [Чип]

обележава са „Дубина x Ширина“, при чему вредност производа даје укупан капацитет чипа.

Меморијски чипови могу садржати једно или више интегрисаних кола у оквиру паковања. Употребом више интегрисаних кола повећава се капацитет чипа, а по потреби се може повећати и ширина магистрале података у односу на ширину магистрале једног интегрисаног кола, као и одредити број независних меморијских канала (Слика 2.7).



Слика 2.7 – DRAM чип

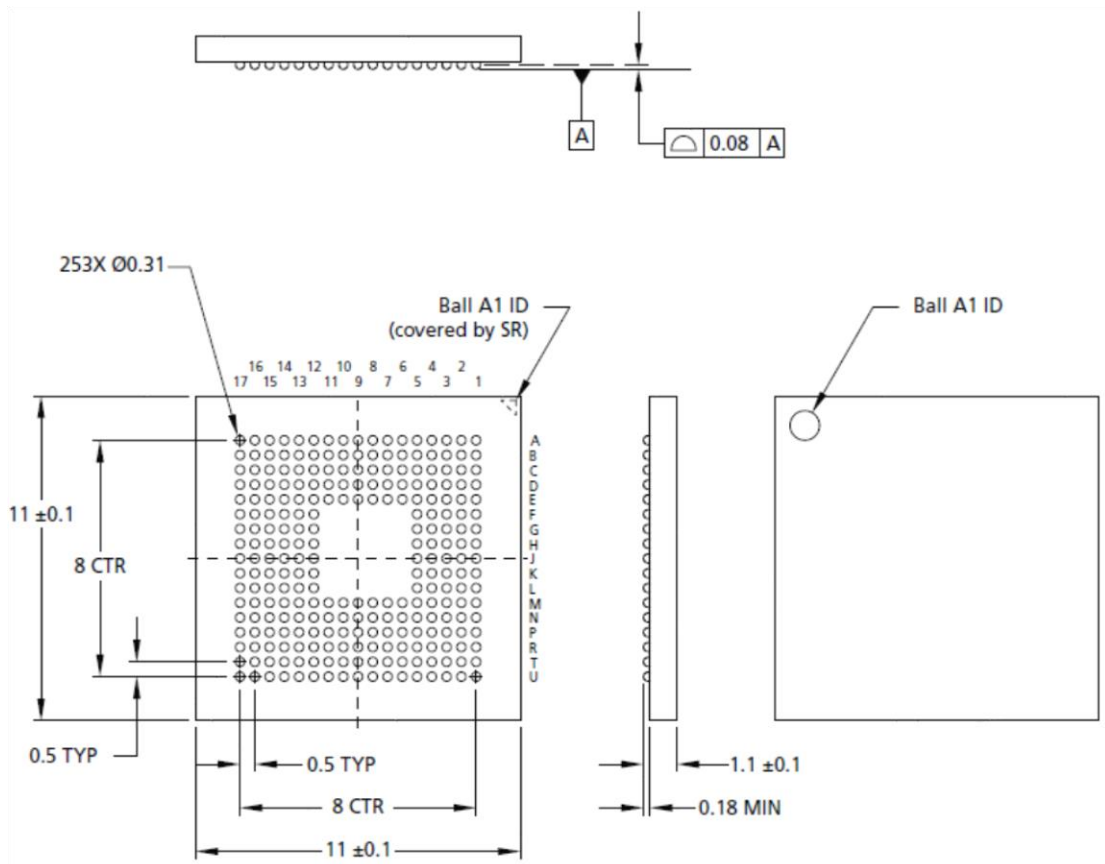
Структура LPDDR2 DRAM чипа капацитета 2GB састављеног од четири интегрисана кола капацитета 4Gb и ширине x16:

а) организација са једним каналом (MT42L512M32D4)

б) организација са два канала (MT42L256M64D4)

CA=Cmd/Addr bus, DQ=Data bus, DM=Data Mask, CK=Clock, CKE=Clock Enable, CS=Chip Select

Извор: Mobile LPDDR2 SDRAM S4 спецификација, Micron Technology Inc., 2011.



Слика 2.8 – FBGA паковање

На слици је приказано паковање чипа MT42L256M64D4 са укупно 253 контактне лоптице у коме се налазе повезана четири LPDDR2 интегрисана кола. Све димензије су дате у милиметрима.

У зависности од физичких димензија интегрисаних кола и намене, меморијски чипови се могу паковати у различите типова кућишта. За DRAM чипове се најчешће користе FBGA паковања (Слика 2.8), јер обезбеђују веома мали профил чипа, чиме се омогућава њихова употреба у великом дијапазону уређаја. Физичке димензије чипа су критичне за примене у носивим, мобилним и преносним уређајима због ограниченог простора, али и у другим системима, као што су сервери и суперрачунари, где је због ограниченог броја места за меморијске модуле на матичној плочи и потребе за што већим капацитетом меморије потребно користити модуле са великом густином паковања, односно са чиповима што мањих димензија.

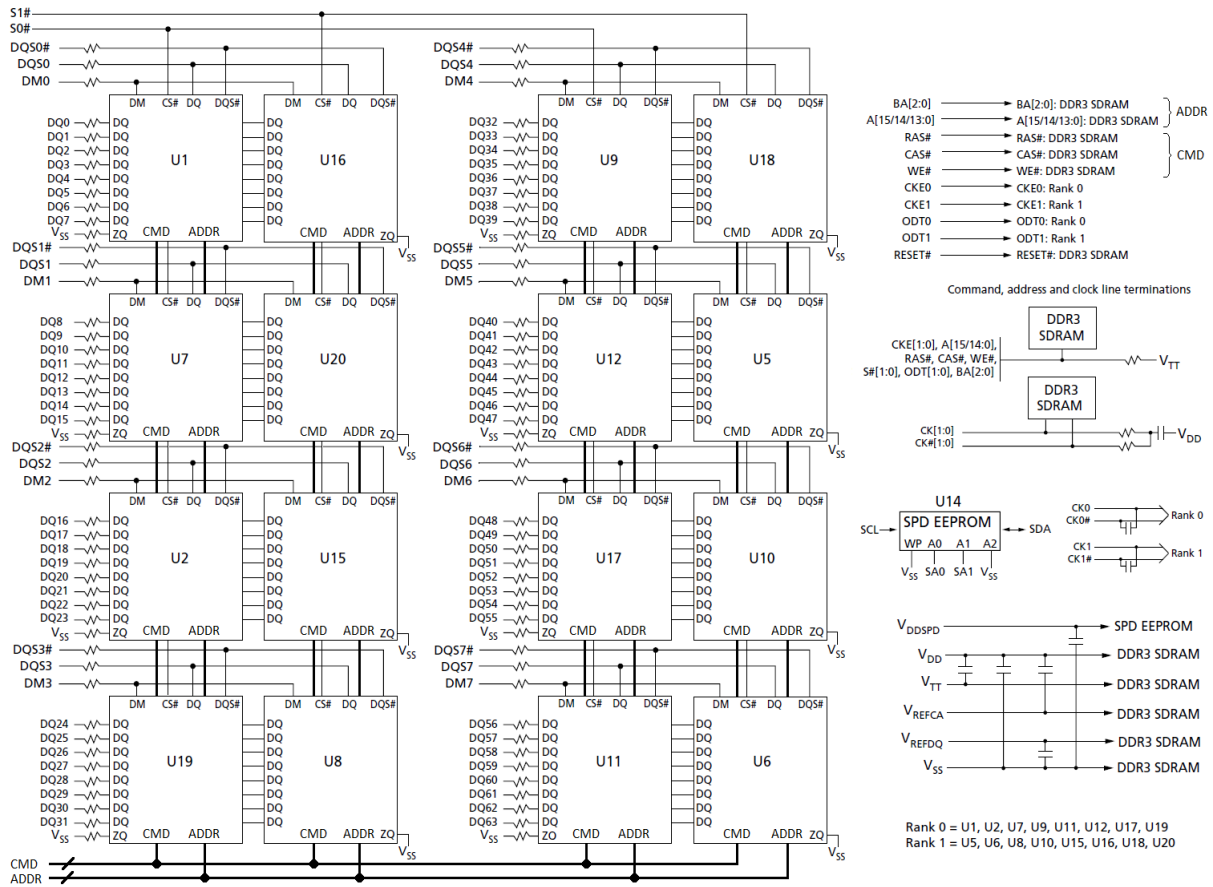
2.2.6 Меморијски модул

Меморијски модул се састоји од већег броја DRAM чипова интегрисаних на штампаној плочици која се може директно повезати на меморијску магистралу преко одговарајућег прикључног места (енг. slot) на матичној плочи. На матичној плочи обично има више прикључних места. Уобичајени број код матичних плоча за лаптопова је два, а код матичних плоча за персоналне рачунаре четири. Матичне плоче за радне станице и сервере могу имати и више прикључних места за меморијске модуле, нпр. осам или шеснаест, у зависности од типа и броја процесорских чипова на плочи, као и врсте меморијских модула који се користе.

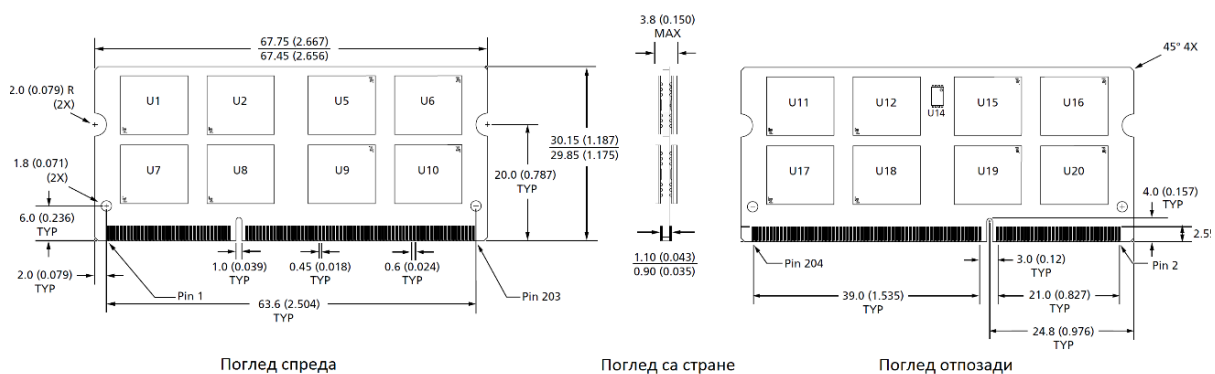
Код DIMM (UDIMM) и SODIMM модула осим DRAM чипова постоји само још SPD (енг. Serial Presence Detect) EEPROM чип са подацима које прописује JEDEC стандард, што укључује тип меморије, број банки, капацитет модула, ширину магистрале података, временске параметре, подржане напоне, подршку за ECC, податке о произвођачу итд. Подаци из SPD EEPROM-а се читају приликом стартовања рачунара и прослеђују BIOS-у који на основу њих конфигурише меморијски контролер. Код ових типова модула се сигнали са меморијске магистрале воде директно до меморијских чипова. Пошто до сваког чипа воде исте адресне, командне и контролне линије (Слика 2.9), оне су вишеструко електрично (капацитивно, индуктивно и резистивно) оптерећене у поређењу са линијама података. Електрично оптерећење линија утиче на кашњење и квалитет сигнала на магистрали, па је број модула који се могу повезати ограничен, најчешће на четири.

Скуп меморијских чипова који се заједно селекују и којима се истовремено приступа зове се ранк (енг. rank). У једном модулу може бити више ранкова, а најчешће их има 1, 2 или 4. На Слици 2.9 приказана је шема везе SODIMM модула са два ранка. Већи број ранкова може да повећа ефикасност приликом приступа DRAM меморији, јер се ефективно повећава укупан број независних банки. Са друге стране, повећава се електрично оптерећење линија података, јер иста линија иде до чипова из свих ранкова.

Модули који имају подршку за детектовање и корекцију грешака (ECC) имају додатне чипове за смештање кодних речи. За детекцију и корекцију грешака



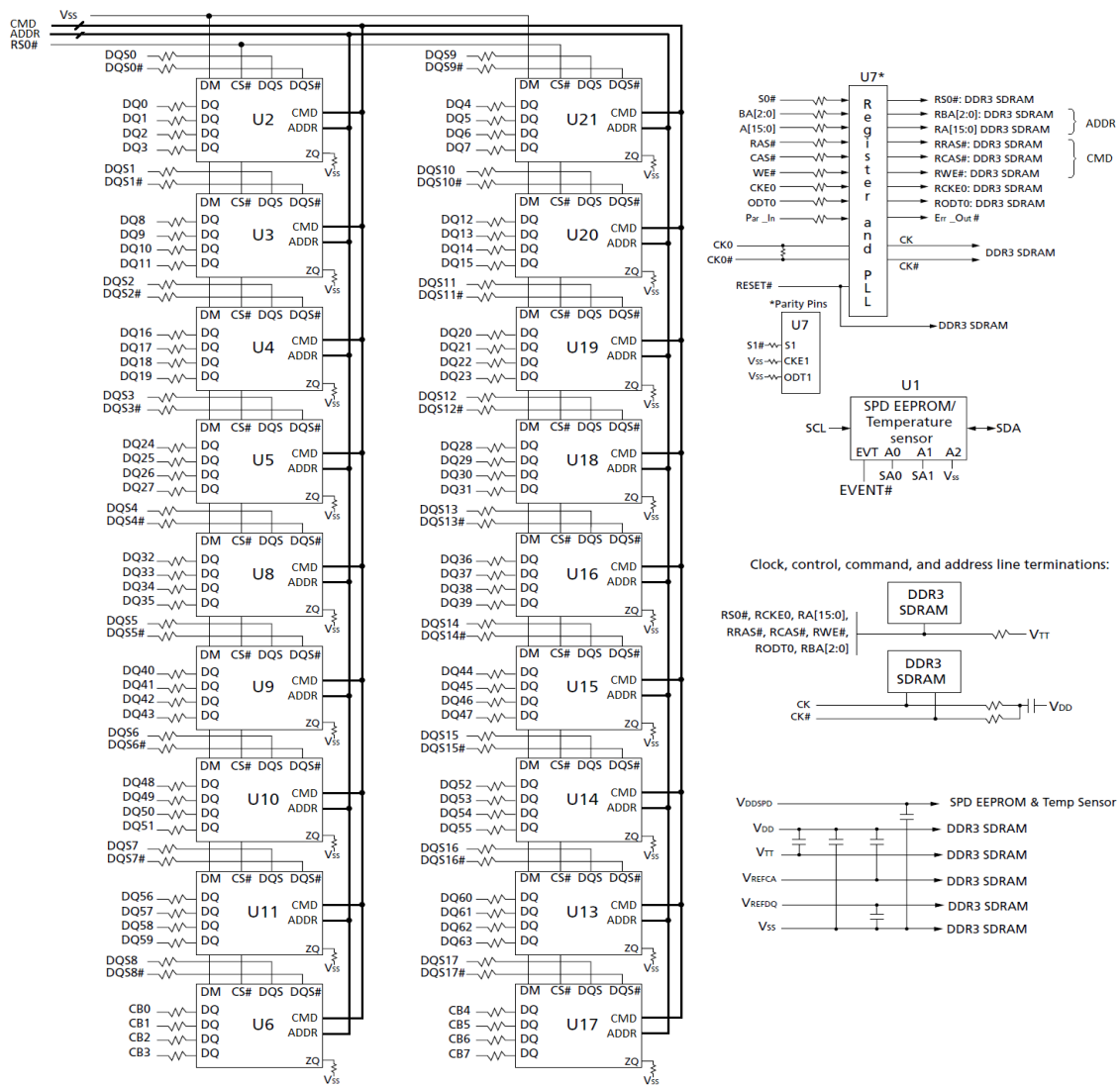
а) шема веза компоненти меморијског модула



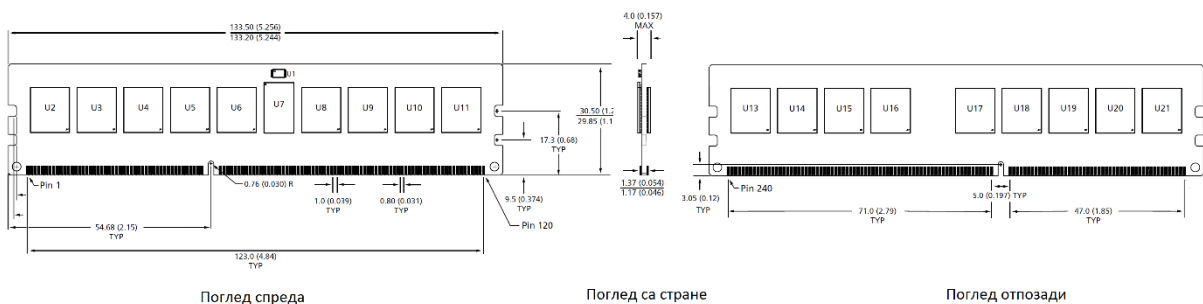
б) физички изглед модула

Слика 2.9 – SODIMM модул

На слици је приказана имплементација 8GB x64 Dual-Rank 204-pin DDR3 SODIMM модула [19]. Сигнали са магистрале се воде директно до сваког појединачног чипа. CMD и ADDR су командни и адресни сигнали са меморијске магистрале, DQx су сигнали података, DQSx су сигнали бирања за DQx сигнале, а Sx су сигнали селекције чипа. Све мере су у милиметрима.



а) шема веза компоненти меморијског модула



б) физички изглед модула

Слика 2.10 – ECC RDIMM модул

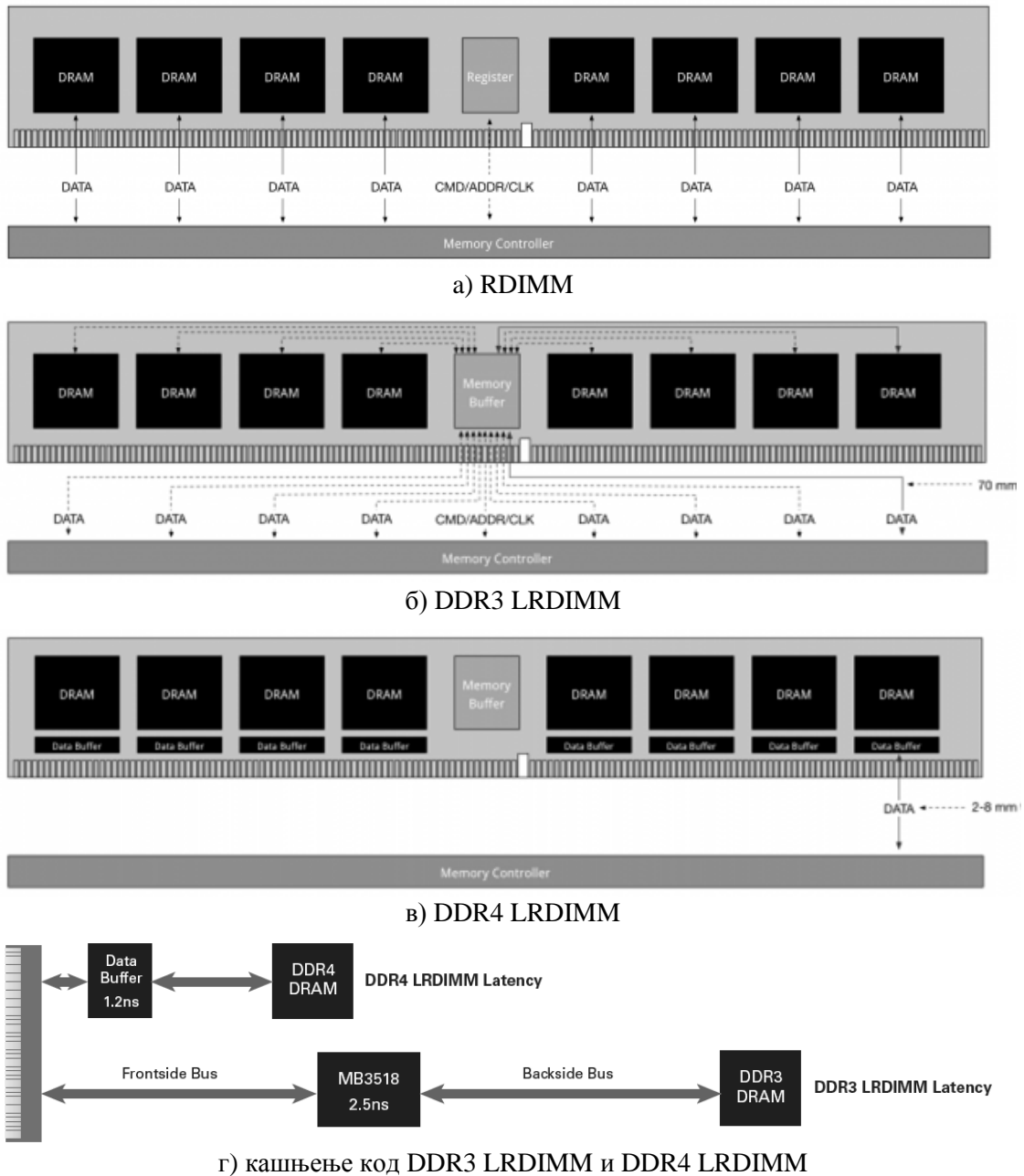
На слици је приказана имплементација 8GB x72 ECC Single-Rank 240-pin DDR3 RDIMM модула [20]. CMD и ADDR су регистровани командни и адресни сигнали са меморијске магистрале, DQx су сигнали података, DQSx су сигнали бирања за DQx сигнале, RSx су регистровани сигнали селекције чипа, а CBx су кодни битови који служе за корекцију и детекцију грешака.

се најчешће користи Хемингов код (127,120) који захтева 7 кодних битова на 120 бита података, али скраћен на 64 бита података и проширен додатним битом парности. То даје (72,64) код, који омогућава детекцију свих двобитних грешака и корекцију свих једнобитних грешака. Пошто овај код користи један кодни бит на сваких осам битова података, меморијски модули који подржавају ЕСС имају један додатни меморијски чип на сваких осам чипова за податке (Слика 2.10).

Да би се смањило електрично оптерећење адресних и командних линија и линије такта меморијске магистрале, код RDIMM модула се између меморијског контролера и меморијских чипова додаје посебан регистарски чип. Регистарски чип региструје сигнале са меморијске магистрале у сваком циклусу, појачава их и разводи до локалних меморијских чипова на модулу, заједно са сигналом такта који се појачава помоћу PLL (енг. Phase-Locked Loop). Тако се повећава максималан број чипова који се могу поставити на једном модулу, као и укупан број модула који се може користити у систему. Уједно се повећава и максимална фреквенција рада, тј. меморијске перформансе.

LRDIMM модули се разликују од RDIMM модула по томе што се код њих не баферују само адресни и командни сигнали, већ се баферују и сви други сигнали са меморијске магистрале, укључујући контролне сигнале и сигнале података. Сигнали података се баферују помоћу бидирекционих драјвера (енг. driver). Пошто се код LRDIMM модула баферују и подаци, повећава се кашњење сигнала података у односу на RDIMM модуле. У првој генерацији (DDR3) LRDIMM модула ово повећање кашњења је било приметно [21]. Разлог за то је што код RDIMM модула линије података иду директно са меморијске магистрале до меморијских чипова (Слика 2.11.а), док су код DDR3 LRDIMM модула линије података рутиране тако да све иду до централног бафер чипа, а одатле даље до меморијских чипова (Слика 2.11.б). Тиме се знатно повећава дужина линија података, што повећава кашњење; нпр. код DDR3 LRDIMM, дужина линија је била и до 70mm. У другој генерацији (DDR4) LRDIMM модула уместо једног већег користи се више мањих бафер чипова и то тако да се сваки бафер физички налази тачно између пинова линија магистрале података и одговарајућег меморијског чипа (Слика 2.11.в). На тај начин је постигнуто да се дужина линија података смањи на само 2-8mm за први ред

меморијских чипова. Уз то, мањи бафер чипови и сами имају мање кашњење у односу на један велики бафер због боље оптимизације (Слика 2.11.г) [22].



Слика 2.11 – LRDIMM модул

DDR3 LRDIMM модули имају централни бафер чип, што изискује рутирање линија података од меморијске магистрале до центра модула, а потом до свих меморијских чипова. Код DDR4 LRDIMM модула има више мањих бафер чипова лоцираних тако да линије података имају знатно краћу путању, слично оној којој имају код RDIMM модула, па је кашњење код DDR4 LRDIMM модула знатно мање него код DDR3 LRDIMM модула.

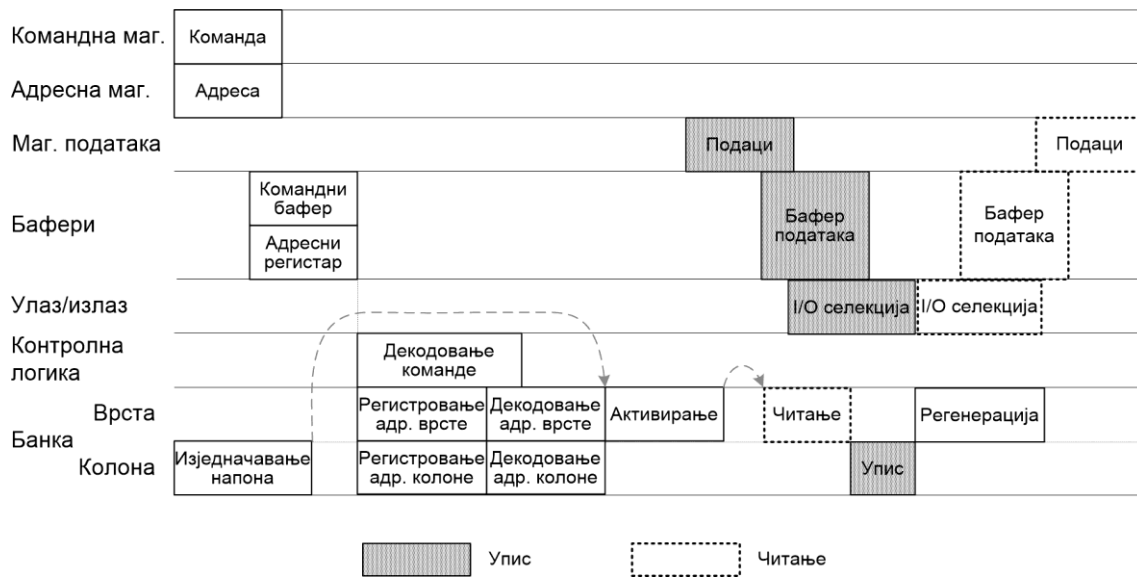
2.3 Проточна обрада код DRAM меморије

Сваки захтев за приступ меморији који долази из рачунарског система се унутар меморијског контролера претвара у секвенцу командних транскација, адресних транскација и транскација преноса података који се шаљу DRAM меморији преко меморијске магистрале. Главне фазе кроз које пролазе ове транскације унутар DRAM интегрисаног кола су:

1. Баферовање команде, адресе и улазних података
2. Декодовање команде
3. Декодовање врсте и колоне на основу адресе
4. Активирање врсте
5. Селекција колоне и читавање садржаја меморијских ћелија (код читања), односно постављање вредности која треба да се меморише (код уписа)
6. Баферовање излазних података и постављање на магистралу података (код читања)

За транскације које су генерисане од стране различитих команди за приступ меморији, фазе 1, 2 и 6 морају да буду серијализоване због коришћења дељених ресурса унутар DRAM интегрисаног кола (види Сliku 2.6). Насупрот томе, фазе 3, 4 и 5 могу бити извршене конкурентно унутар различитих банки. Са тачке гледишта перформанси, приступ меморијским ћелијама у фазама 4 и 5 је најкритичнији, јер је трајање ових фаза ограничено физичким карактеристикама DRAM матрице. Најкритичнији су заправо читање и упис у фази 5, јер иако операција активирања врсте у фази 4 дуже траје, фреквенција операција приступа садржају меморијских ћелија из колоне активне врсте је већа у одосу на фреквенцију операције активирања врсте.

Све фазе проточне обраде код DRAM меморије су дате на Слици 2.12. Изједначавање напона мора да претходи активирању врсте, али не мора да буде извршено непосредно пре. Раздвајање фазе изједначавања напона од осталих операција приступа меморијским ћелијама повећава вероватноћу да се искористе



Слика 2.12 – Проточна обрада код DRAM меморије

Стрелицама је означена функционална зависност између појединих фаза. Фазе које се разликују код читања и уписа су посебно означене на слици.

слободни периоди како би се у току њих обавило изједначавање. Слично томе, активирање врсте не мора да се ради сваки пут када се приступа подацима. Раздвајање активирања врсте од осталих операција приступа меморијским ћелијама може смањити просечно време приступа, јер се може обавити више операција читања и уписа по колонама исте активне врсте. Читање садржаја меморијских ћелија једне врсте може се обавити непосредно по њеном активирању и довољно је да се обави само једном. Очитане вредности ће остати лечоване унутар диференцијалног појачавача све време док је врста активна. Тиме се редукује кашњење приликом читања, јер није потребно читати из меморијских ћелија и ажурирати стање диференцијалних појачавача у сваком циклусу.

Применом свих ових оптимизација, операција читања се своди на трансфер података из диференцијалних појачавача селектованих колона на линије магистрале података, а операција уписа на пренос вредности са магистрале података на одговарајуће диференцијалне појачаваче колона. У том случају, терет

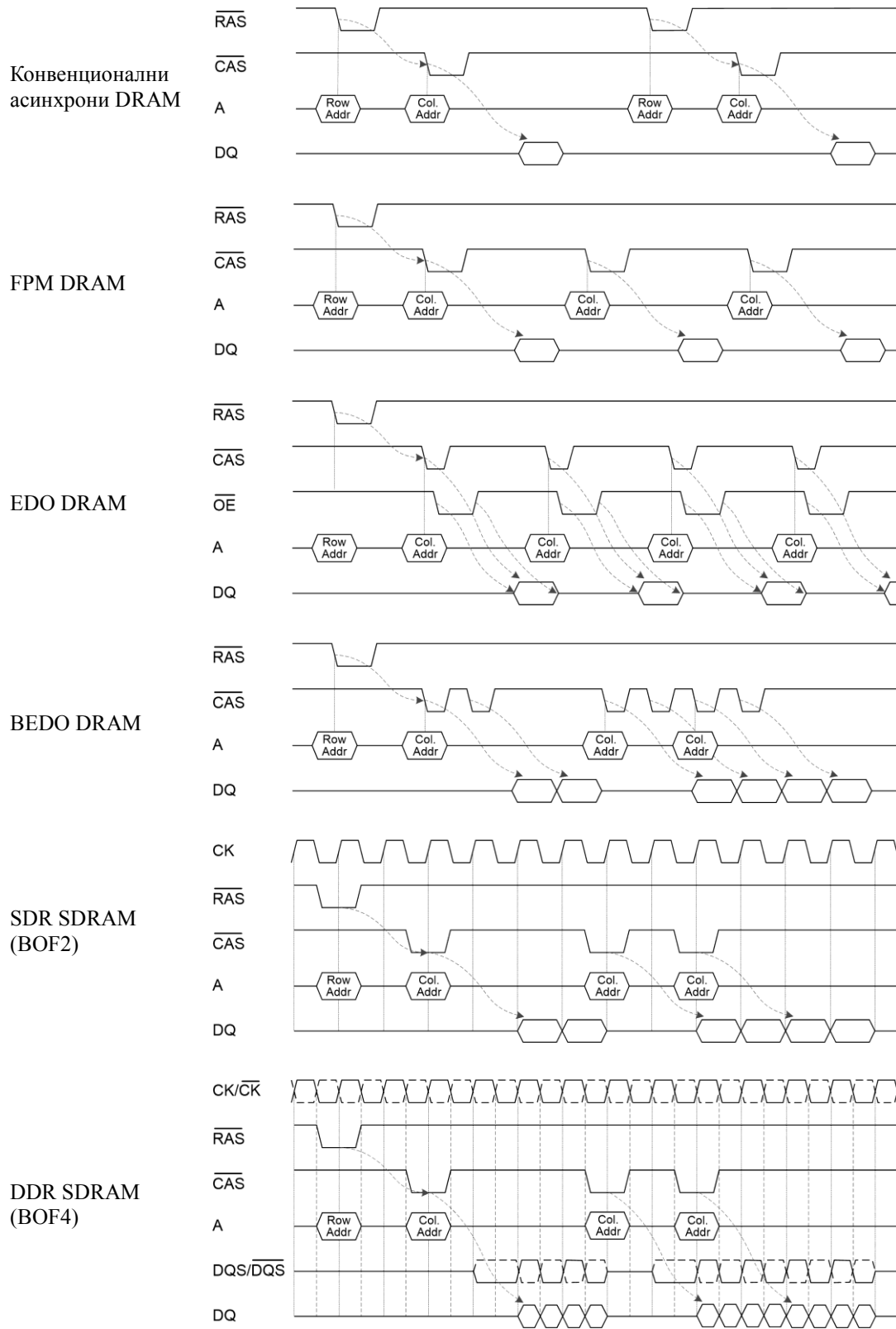
обављања функција изједначавања напона колона, активације врста и правременог читања се пребацује на меморијски контролер.

2.4 DRAM архитектура

Историјски гледано, иновације у домену DRAM архитектура су биле мотивисане потребом да се уклони уско грло између процесора и главне меморије минимизовањем кашњења и повећањем протока података. Проток података се до одређене мере може повећати дохватањем више битова по циклусу из меморијске матрице (коришћењем просторне локалности), док се кашњење може редуковати оптимизацијом проточне обраде приликом приступа подацима у DRAM меморији. Ове оптимизације су постепено увођене у неколико генерација DRAM архитектура (Слика 2.13).

Ране DRAM архитектуре су биле асинхроне. Код асинхроних DRAM архитектура, интерне сигнале DRAM меморија контролише контролна логика смештена на страни меморијског контролера. Постоје два контролна сигнала: ивица RAS (енг. Row Address Strobe) сигнала иницира процес активирања врсте, док ивица CAS (енг. Column Address Strobe) сигнала иницира приступ колонама и пренос података. Адресе врста и колона се шаљу одвојено преко меморијске магистрале, заједно са одговарајућим RAS и CAS сигнаlima. Коришћење исте магистрале за слање адреса врста и колона редукује потребан број линија и пинова на половину. То заузврат смањује цену и комплексност.

Код конвенционалне асинхроне DRAM меморије, приликом сваког приступа подацима пролази се кроз све фазе DRAM проточне обраде. То значи да су перформансе меморије лимитиране пре свега временом активирања врсте, јер она траје најдуже од свих фаза. Код FPM (енг. Fast Page Mode) DRAM меморија, дозвољено је да врста остане активна и након приступа како би се омогућило вишеструко приступање ћелијама исте врсте, избегавајући тако да се сваки пут непотребно врши активирање и изједначавање напона. Код EDO (енг. Extended Data Out) DRAM меморија, уводе се излазни лечеви за податке који се контролишу посебним OE (енг. Output Enable) сигналом који омогућава да излазна вредност



Слика 2.13 – Еволуција DRAM архитектура

Сигнали илуструју сценарио узастопног читања из исте врсте меморијске матрице.

остане валидна и након спуштања CAS ивице. На тај начин се омогућава да се CAS сигнал пошаље раније него код FPM DRAM меморија, чиме се смањује кашњење. Код BEDO (енг. Burst-mode EDO) DRAM меморија, уводи се интерни бројач адресе колоне који се аутоматски инкрементира са сваком узастопном појавом водеће ивице CAS сигнала до специфицираног лимита. Пошто се селекција колоне код овог типа меморија ради локално, а не помоћу екстерних сигнала као у претходним генерацијама DRAM меморија, проблеми са тајмингом се значајно редукују, што омогућава краће циклусе, а тиме и повећање брзине рада. Такође, приступ у више узастопних циклуса помаже да се искористи просторна локалност података која постоји у великом броју случајева.

У модерним системима се користе само синхроне DRAM меморије (SDRAM), јер имају супериорне перформансе у односу на асинхроне DRAM меморије услед мање варијације у времену доласка сигнала такта (енг. clock skew), што је резултат коришћења синхроног интерфејса и синхроних локалних контролних сигнала. RAS и CAS сигнали се и даље користе, али не да директно контролишу интерна DRAM кола као код асинхроних DRAM меморија. Уместо тога, RAS и CAS сигнали су само део интерфејса преко ког меморијски контролер издаје команде DRAM меморији, док је логика која контролише интерне меморијске сигнале смештена на самом DRAM интегрисаном колу. У суштини, меморијски контролер види DRAM меморију као синхрони коначни аутомат. Додатна предност је омогућавање рада са више банки. DRAM може да прихвати нове команде које адресирају различите банке док се старе команде још увек извршавају, чиме се смањује просечно време приступа подацима. Такође, пошто се логика за прихват и декодовање команди налази на самом DRAM интегрисаном колу, може се учинити програмабилном. На тај начин се обезбеђује већа флексибилност у погледу типова DRAM модула који се могу користити у систему.

SDR (енг. Single Data Rate) SDRAM меморије користе само узлазну ивицу сигнала такта за пренос података и слање контролних сигнала. Код DDR (енг. Double Data Rate) SDRAM меморија, магистрала података има двоструко већи проток од адресне и контролне магистрале, јер користи и узлазну и силазну ивицу сигнала такта за пренос података. Магистрала података може да ради на већој

фреквендији од адресне и контролне магистрале јер је електрично оптерећење код ње ниже: адресна и контролна магистрала морају да буду везане на пинове свих меморијских чипова у систему, док су линије података везане на пинове само оних чипова који су у истом ранку (види Сliku 2.9 и Сliku 2.10). Иако је проток на магистрали података код DDR SDRAM двоструко већи у односу на SDR SDRAM, брзина читања података унутар самог DRAM кола није већа, већ је количина података који се читају у једном циклусу двоструко већа. Интерно се ради мултиплексирање тако да се прочитана двострука количина података шаље на магистралу података у два полуциклуса, тј. дупло брже него што се подаци интерно читају. Однос количине података који се читају у току једног циклуса и ширине магистрале података се назива однос дохватања (енг. prefetch ratio). Овај однос одређује минималан број циклуса BL (енг. burst length) колико мора да траје свака трансакција преноса на магистрали података. За однос дохватања величине P користи се још и термин P -битно дохватање (енг. P -bit prefetching). Дужина трајања трансакција преноса података на магистрали података код SDRAM меморија је програмабилна, али мора бити целобројни умножак односа дохватања, јер се читање из меморије може обављати само у целом броју циклуса. То значи да се за сваку трансакцију преноса података на магистрали обавља један или више интерних циклуса читања или уписа у меморији. Ако је број интерних циклуса читања I , онда ће трансакција преноса трајати $BL = I \cdot P$ циклуса. За дужину трајања трансакција преноса података BL користи се и термин „пренос дужине BL “ (енг. burst of BL). У поглављу 2.2.4 било је показано да мора да важи релација $N \cdot B = M \cdot P$, а сада се може извести следећа општа релација:

$$I \cdot N \cdot B = M \cdot BL, \quad BL = n \cdot P, \quad n \in \mathbb{N} \quad (2.15)$$

Код DDR SDRAM меморија се у односу на SDR SDRAM уводи још један сигнал, DQS (енг. data strobe signal), који служи као референтни сигнал за синхронизацију између извора и одредишта. Он обезбеђује да се приликом преноса узорковање (енг. sampling) на одредишту врши сигналом који је синхрон са подацима (енг. source synchronous). Код уписа у меморију, DQS сигнал генерише меморијски контролер и то тако да је ивица овог сигнала на средини „ока“ сигнала

података (његова фаза је померена за $\pi/2$ у односу на сигнал такта), јер се директно користи за узорковање података који се уписују у меморију. Код читања података из меморије, DQS сигнал се генерише помоћу DLL (енг. Delay-Locked Loop) уграђеног у DRAM интегрисано коло, који помера фазу DQS сигнала тако да буде синхрон у односу на улазни сигнал такта.

Анализе су показале да је свака генерација DRAM меморија са новом архитектуром представљала корак унапред у смислу перформанси [23]. Додатна унапређења се још могу постићи на системском нивоу, на пример оптимизацијом времена и редоследа слања DRAM команди од стране меморијског контролера [24] и коришћењем техника за скривање кашњења (енг. latency hiding) као што је преплитање меморијских банки и канала (енг. bank and channel interleaving).

2.5 DRAM стандарди

Сваки DRAM уређај мора да има функционалну, временску и електричну спецификацију која дефинише услове за његов рад и његове карактеристике. С обзиром да постоји велики број произвођача DRAM меморија и да се DRAM меморије користе у великом броју уређаја, неопходна је стандардизација како би се олакшала њихова имплементација и коришћење. У пракси се стандардизацијом DRAM уређаја бави JEDEC организација. Меморијски стандарди дефинишу: 1) физичке карактеристике меморијских уређаја, као нпр. паковање, распоред пинова, итд.; 2) електричне карактеристике, као нпр. напоне, јачине струја на пиновима, дозвољене фреквенције и сл.; и 3) интерфејс и протокол којим се преко њега комуницира.

JEDEC SDR [25], DDR [26], DDR2 [27], DDR3 [28] и DDR4 [29] SDRAM стандарди се користе за имплементацију DRAM меморија у већини модерних персоналних и серверских рачунарских система и великом броју embedded платформи. JEDEC LPDDR [30], LPDDR2 [31], LPDDR3 [32] и LPDDR4 [33] стандарди се користе за примене у системима са ниском потрошњом, као што су носиви, преносиви и мобилни рачунарски системи. LPDDR је базиран на DDR стандарду, LPDDR2 на елементима LPDDR(1) и DDR2/3 стандарда, LPDDR3

Меморија	Густина	Ширина	Prefetch	BL	# Банки/групи	# Група/каналу	# Канала	Интерни такт [MHz]	Такт командне магистрале [MHz]	Максимални проток [Mb/s]	Напајање [V]
SDR	64Mb-512Mb	x4,x8,x16	1	1,2,4,8	4	1	1	66-200	66-200	66-200	3.3
DDR	128Mb-1Gb	x4,x8,x16	2	2,4,8	4	1	1	83-200	83-200	200-400	2.5
DDR2	256Mb-4Gb	x4,x8,x16	4	4,8	4,8	1	1	100-200/266	200-400/533	400-800/1066	1.8
DDR3	512Mb-8Gb	x4,x8,x16	8	BC4,8	8	1	1	100-200/266	400-800/1066	800-1600/2133	1.5
DDR4	2Gb-16Gb	x4,x8,x16	8	BC4,8	4	4	1	100-200/266	800-1600	1600-3200	1.2
LPDDR	64Mb-2Gb	x16,x32	2	2,4,8,16*	4	1	1	100-200/266	100-200/266	200-400/533	1.8
LPDDR2	64Mb-8Gb	x8,x16,x32	2,4	4,8,16	4,8	1	1	100-200/266	200-400/533	400-800/1066	1.2
LPDDR3	4Gb-16/32Gb	x16,x32	8	8	8	1	1	100-200/266	400-800/1066	800-1600/2133	1.2
LPDDR4	4Gb-32Gb	2x16	16	16,32	8	1	2	100-200/266	800-1600/2133	1600-3200/4266	1.1
GDDR	8Mb-16Mb	x32	2	2,4,8	1,2	1	1	166-300	166-300	333-600	2.5
GDDR2	1Gb	x16	4	4,8	8	1	1	200-225	400-500	800-1000	1.8
GDDR3	256Mb-1Gb	x32	4	4,8	4,8	1	1	250-300	500-600	1000-1200	1.8
GDDR4	256Mb-1Gb	x32	8	8	4,8	1	1	300-400	1200-1600	2400-3200	1.5
GDDR5	512Mb-8Gb	x16,x32	8	8	8,16	4	1	200-450	800-1800	3200-7200	1.5
GDDR5X	4Gb-16Gb	x16,x32	8,16	8,16	4	4	1	200	1600	6400-12800	1.35

Табела 2.2 – Параметри DRAM меморија имплементираних према JEDEC стандардима

Време приступа меморијским ћелијама, односно вредност интерног такта меморијског језгра, се није много променило кроз генерације DRAM меморија. Повећање протока код новијих меморија се постиже углавном повећањем броја prefetch битова и пропорционалним повећањем брзине магистрале података, а у новије време интерном паралелизацијом приступа. Новије технологије омогућавају и већу густину паковања, као и нижи напон напајања.

*=optional, BL=Burst Length, BC4=Burst Chop of 4

користи аспекте LPDDR1/2 и DDR2/3 стандарда, док је LPDDR4 креиран коришћењем елемената из DDR2/3/4 и LPDDR1/2/3 стандарда.

Код графичких подсистема, као нпр. код графичких картица или код играчких конзола, користе се embedded меморијски чипови, при чему је број чипова по меморијском каналу мали, тако да је меморијска магистрала краћа, а електрично оптерећење линија ниже. То омогућава меморијским чиповима да раде брже и уз мању потрошњу енергије него чипови системска меморије, што је била мотивација

за стварање JEDEC GDDR [34], GDDR2 [35], GDDR3 [36], GDDR4 [37], GDDR5 [38] и GDDR5X [39] SGRAM стандарда за граfiчке меморије. GDDRx SGRAM стандарди су иницијално били базирани на DDRx SDRAM стандардима. Тако је GDDR базиран на DDR стандарду, GDDR2 и GDDR3 на DDR2 стандарду, а GDDR4 на DDR3 стандарду. Међутим, код GDDR5 се одступило од тог шаблона, па постоји неколико разлика између GDDR5 и DDR3 стандарда. Једна од главних разлика се односи на сигнал такта: команде се и даље региструју на сваку растућу ивицу сигнала CK, адресе се региструју на растуће ивице сигнала CK и \overline{CK} , али су за податке уведени нови сигнали WCK и \overline{WCK} , који имају двоструко већу фреквенцију од CK и \overline{CK} .

Неке од главних карактеристика JEDEC SDRAM меморија су дате у Табели 2.2. Из табеле се види да се време приступа меморијским ћелијама, тј. интерни такт, мало променило од првих генерација DRAM меморија. Повећање протока података се постиже пре свега повећањем броја prefetch битова. Код новијих меморија додатно повећање протока се постиже временским мултиплексирањем читаних вредности, тј. применом TDM (енг. Time Division Multiplex) шеме на више група банки (нпр. код DDR4) или паралелизацијом приступа преко више интерних канала (нпр. код LPDDR4).

О новим стандардима за 3D вишеслојне (енг. stacked die) меморије, као што су Wide I/O, Wide I/O 2, HBM и HMC ће бити више речи у поглављу 7.7.

2.6 DRAM протокол

Као што је наведено у поглављу 2.5, DRAM стандард, између осталог, дефинише интерфејс који се користи за повезивање DRAM интегрисаних кола са остатком система, као и одговарајући протокол за комуникацију, тзв. DRAM протокол. Циљ DRAM протокола је да омогући комуникацију између меморијског контролера и DRAM меморије са циљем да се омогуће све активности које су потребне за обављање функција читања и уписа.

Команда	Опис
Activate	Отвара приступне транзисторе у истој врсти свих матрица које се налазе унутар DRAM банке и иницира процес читања података лечовањем очитане вредности у диференцијални појачавач (назива се и отварање странице)
Precharge	Затвара приступне транзисторе у отвореној врсти свих матрица које се налазе унутар DRAM банке и поставља напон на улазима одговарајућих диференцијалних појачавача на референтни ниво (назива се и затварање странице)
Precharge all banks	Исто као Precharge, али за све DRAM банке
Read	Преноси податке лечоване унутар диференцијалних појачавача селектованих колона на магистралу података
Read with auto-precharge	Исто као Read, осим што се после очитавања аутоматски ради Precharge за дату банку
Write	Доводи податке са магистрале података на улазе диференцијалних појачавача селектованих колона, одакле се уписују у меморијске ћелије
Write with auto-precharge	Исто као Write, осим што се после уписа аутоматски ради Precharge за дату банку
Refresh	Освежава податке у DRAM ћелијама тако што у свакој DRAM банки активира једну по једну врсту свих припадајућих матрица и очитава вредности из њих

Табела 2.3 – Кључне команде DRAM протокола

Као и сваки други протокол, DRAM протокол има своју синтаксу, семантику и синхронизацију. Синтаксу DRAM протокола чине правила за слање и пријем трансакција на меморијској магистрали. Трансакције се састоје од команди, адреса и података. Део синтаксних правила за слање и пријем трансакција су и временска ограничења која дефинишу када се неки сигнали смеју слати, а која су условљена различитим физичким, архитектуралним и другим разлозима. У поглављу 4 ће бити показана улога коју временска ограничења имају у анализи перформанси DRAM меморија.

Семантика трансакција на меморијској магистрали је дефинисана DRAM архитектуром. Када су у питању трансакције које меморијски контролер шаље ка меморији, значење трансакција се одређује на основу декодовања послатих команди, при чему се адресе и подаци узимају као пратећи аргументи. Неке од

транзакција могу, на пример, бити: читање или упис у регистре DRAM уређаја, активирање врсте у меморији, читање или упис на одређене адресе у меморији и сл. Када је у питању слање података од меморије ка меморијском контролеру, транзакције увек представљају одговор на команду читања из регистра или из DRAM меморије коју је меморијски контролер последњу послао.

Синхронизација комуникације се обавља помоћу сигнала такта који се такође шаље преко меморијске магистрале. Ово је значајан аспект DRAM протокола, јер утиче на квалитет и брзину преноса сигнала на меморијској магистрали, а тиме и перформансе DRAM меморије.

DRAM команде се могу класификовати у две групе: 1) команде за иницијализацију, управљање режимом рада и потрошње енергије и *por* (енг. по operation) команде; и 2) команде за пренос података и освежавање меморије. Команде из друге групе су суштински важне за мерење перформанси DRAM меморија и називају се кључне команде (Табела 2.3). Ове команде су исте код DDR и DDR2 протокола, док DDR3 протокол има додатне модове операција читања и уписа који омогућавају да се дужина трајања транзакције на магистрали података може мењати у току рада. LPDDR_x и GDDR_x протоколи имплементирају исти скуп команди као и одговарајући DDR_x протоколи на којима су базирани. Изузетак је GDDR5, код кога је скуп команди проширен у односу на DDR3 протокол командама за тренирање читања и уписа.

Временска ограничења (енг. timing constraints) дефинишу време који мора протећи између релевантних догађаја у меморијском подсистему како би се осигурао поуздан приступ подацима ускладиштеним у DRAM ћелијама и како би се избегао конфликт приликом приступа интерним ресурсима унутар кола или конфликт приликом приступа меморијској магистрали. Ова ограничења се изражавају преко временских параметара (енг. timing parameters) и базирана су на електричним и архитектуралним особинама DRAM кола и меморијске магистрале. Временски параметри се могу класификовати у две групе: 1) есенцијални временски параметри – квантификују физичке и архитектуралне особине DRAM кола и 2) изведени временски параметри – користе се у пракси за дефинисање временских ограничења, али се могу изразити преко есенцијалних временских

Ознака	Параметар	Опис
t_{AL}	Additive Latency	Време након ког се издаје интерна команда читања или уписа ако tRCD ограничење није задовољено
t_{CL}	CAS Latency	Време између интерне команде читања и појаве података на меморијској магистрали
t_{RTP}	Read-to-Precharge Time	Минимално време између интерне команде читања и Precharge команде
t_{WR}	Write Recovery Time	Минимално време између краја трансакције преноса података и Precharge команде
t_{WTR}	Write-to-Read Time	Минимално време између краја трансакције преноса улазних података и команде читања
t_{RCD}	RAS-to-CAS Delay	Минимално време између Activate команде и интерне команде читања или уписа
t_{RRD}	RAS-to-RAS Delay	Минимално време између две Activate команде издате различитим банкама
t_{RAS}	Row Activate Time	Минимално време између Activate и Precharge команде
t_{RP}	Row Precharge Time	Минимално време између Precharge и Activate или Refresh команде
t_{RFC}	Refresh Cycle Time	Минимално време између Refresh команде и Activate или друге Refresh команде
t_{AOND}	ODT Turn-On Delay	Време између регистровања ODT (енг. on-die termination) сигнала и искључења ODT отпорности
t_{AON}	ODT Turn-On Time	Време од када је ODT отпорност у стању високе импедансе па док се ODT отпорност потпуно не укључи
t_{AOFD}	ODT Turn-Off Delay	Време између регистровања ODT сигнала и искључења ODT отпорност
t_{AOF}	ODT Turn-Off Time	Време од укључивања ODT отпорности до уласка у мод високе импедансе
t_{BTT}	Bus Turn-around Time	Минимално време потребно да се промени мастер на магистрали података (да се ресинхронизује DQS)
t_{FAW}	Four-bank Activation Window	Временски интервал у току кога се не сме издати више од четири Activate команде

Табела 2.4 – Есенцијални временски параметри

параметара. Есенцијални и изведени временски параметри за DDR2 SDRAM који се односе на кључне команде су дати у Табели 2.4 и Табели 2.5. Дефиниције временских ограничења за DDR3 SDRAM су сличне као за DDR2, са неколико изузетака који се односе на контролу дужине трајања трансакција преноса података и ODT временске параметре. На пример, код уписа са променљивим трајањем трансакције где је трајање постављено на 4 циклуса (BL4), t_{WR} и t_{WTR} почињу два

Ознака	Параметар	Вредност	Опис
t_{CCD}	CAS-to-CAS Delay	prefetch/DCBCR	Минимално време између две наредбе читања или уписа
t_{CWL}	CAS Write Latency	$t_{CL}-1$	Време између интерне команде уписа и почетка одговарајуће трансакције преноса улазних података на магистралу података
t_{READ}	Read Time	Read BL/DCBCR	Време колико се подаци очитани из меморије налазе на магистралу података
t_{RL}	Read Latency	$t_{AL}+t_{CL}$	Време између издавања Read команде и појаве очитаних података на меморијској магистралу
t_{WRITE}	Write Time	Write BL/DCBCR	Време колико се подаци који треба да се упишу у меморију налазе на магистралу података
t_{WL}	Write Latency	$t_{AL}+t_{CWL}$	Време између Write команде и појаве података за упис на магистралу података
t_{RTW}	Read-to-Write Delay	$t_{RL}-t_{WL}+t_{READ}+t_{BTT}$	Минимално време између Read и Write команде
t_{RC}	RAS Cycle Time	$t_{RAS}+t_{RP}$	Минимално време између две Activate команде
t_{ODT}	ODT Time	$\max(t_{AOND}-t_{AOFD}+t_{AOF}, t_{AOFD}-t_{AOND}+t_{AON})$	Минимално време између две трансакције преноса података издате за различите ранкове
t_{RDRD}	Read-to-Read Delay	$t_{READ}+\max(t_{BTT}, t_{ODT}^*)$	Минимално време између две Read команде издате за различите ранкове
t_{RDWR}	Read-to-Write Delay	$t_{RL}+t_{READ}+\max(t_{BTT}, t_{ODT}^*)-t_{WL}$	Минимално време између Read и Write команде издате за различите ранкове
t_{WRRD}	Write-to-Read Delay	$t_{WL}+t_{WRITE}+\max(t_{BTT}, t_{ODT}^*)-t_{RL}$	Минимално време између Write и Read команде издате за различите ранкове
t_{WRWR}	Write-to-Write Delay	$t_{WRITE}+t_{ODT}^*$	Минимално време између две Write команде издате за различите ранкове

Табела 2.5 – Изведени временски параметри

DCBCR=Data to Command Bus Clock Ratio, BL=Burst Length

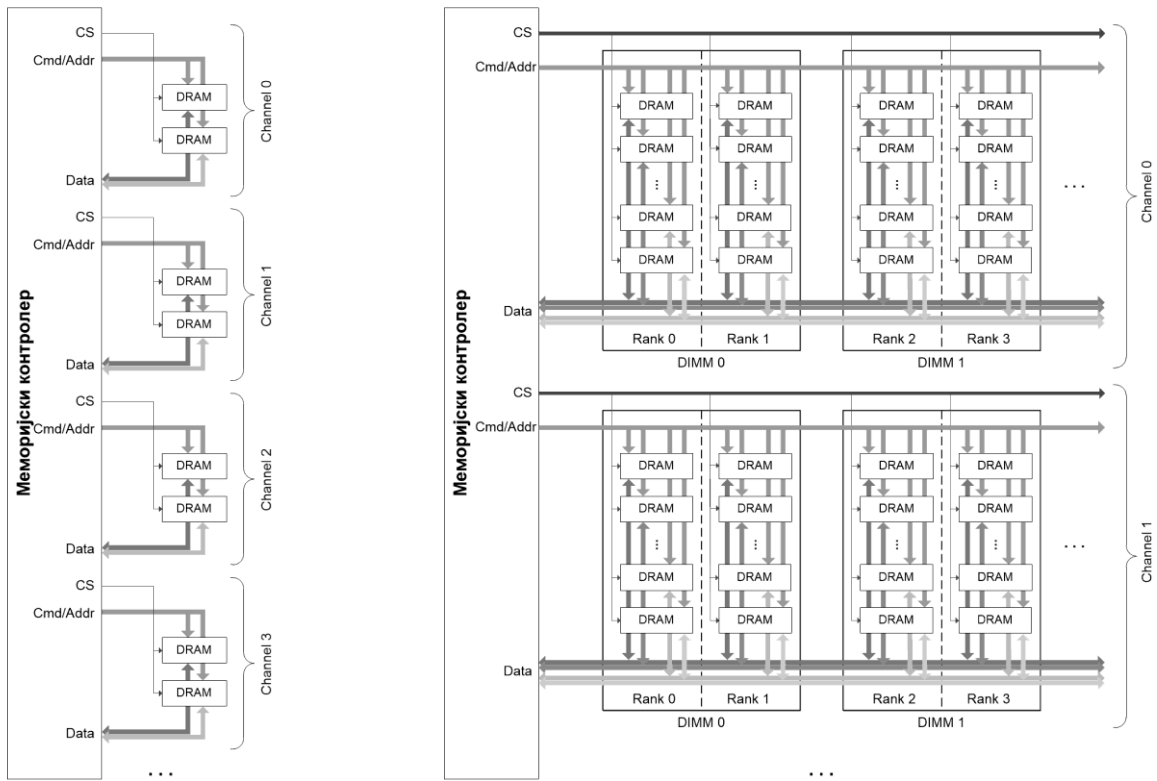
циклуса након почетка трансакције преноса на магистралу података, јер се интерно команда извршава као да је трајање трансакције 8 циклуса (BL8).

2.7 Организација меморијског система

Под организацијом меморијског система се подразумева начин како је DRAM меморија повезана са остатком система. Улогу посредника између остатка рачунарског система и меморије обавља меморијски контролер. Контролер прихвата захтеве за читање и упис у меморију од стране меморијских клијената унутар система, арбитрира који од тих захтева ће бити прихваћен и генерише одговарајуће сигнале на меморијској магистралаи у складу са DRAM протоколом. У систему може постојати више меморијских магистрала. Меморијска магистрала којом меморијски контролер може управљати независно се назива меморијски канал (енг. channel).

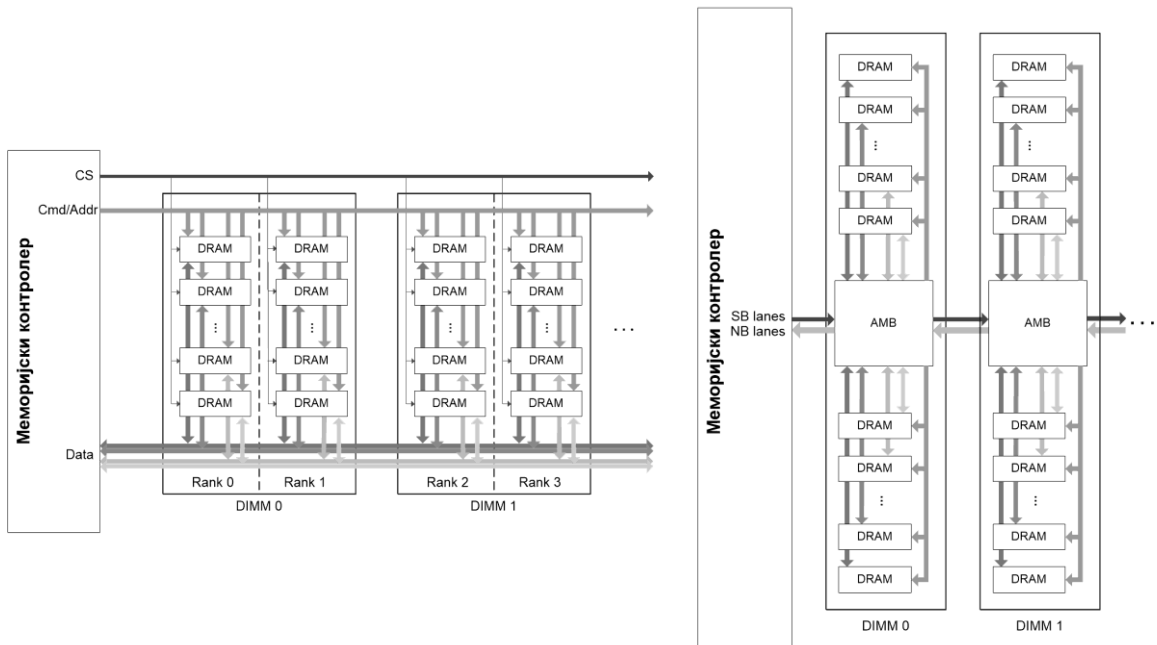
Различите организације меморија се могу класификовати у складу са топологијом веза које меморијски сигнали морају да пређу од меморијског контролера до меморијског кола и обратно приликом приступа меморији. Најчешћи типови организација који се могу срести у пракси су: паралелна, серијска, паралелно/серијска и „од-тачке-до-тачке“ (енг. point-to-point) (Слика 2.14). Паралелна организација се претежно користи за имплементацију локалне меморије у embedded системима или графичким подсистемима, серијска и паралелно/серијска организација се типично користе за имплементацију системске меморије, а организација „од-тачке-до-тачке“ за имплементацију системске меморије у серверским системима.

Циљ сваке организације меморијског система је да омогући што већи капацитет и проток података уз што мање кашњење и потрошњу енергије. Паралелна организација подразумева постојање више меморијских канала, при чему је дужина путања сигнала на свим каналима једнака за сваки DRAM чип. Код серијске организације постоји само један меморијски канал, при чему дужине путања сигнала нису једнаке за сваки DRAM чип, већ она прогресивно расте са физичком удаљеношћу од меморијског контролера. Код паралелно/серијске организације постоји више меморијских канала, при чему су DRAM чипови на сваком каналу организовани серијски. Код организације „од-тачке-до-тачке“ сигнали са магистрале не пропагирају до свих DRAM чипова у једном такту, као



а) Паралелна организација

в) Паралелно/серијска организација



б) Серијска организација

г) Организација „од-тачке-до-тачке“

Слика 2.14 – Организација меморијског система

код претходно наведених организација меморије. У овом случају се сигнали са магистрале у једном такту прослеђују само једној групи DRAM чипова (нпр. у оквиру једног модула), где се баферују и прослеђују наредној групи тек у следећем такту.

Свака организација има своје предности и мане. Паралелна организација омогућава максимално скраћивање дужине линија меморијске магистрале и редуковање броја чипова по каналу, али потреба да меморијски контролер управља са више независних канала захтева додатне ресурсе на чипу, односно повећава заузету површину и потрошњу. Насупрот томе, серијска организација минимизује потребне ресурсе за управљање DRAM меморијом, али на рачун повећаног електричног оптерећења магистрале и смањења перформанси. Паралелно/серијска организација представља компромис између ове две крајности, омогућавајући да се нађе равнотежа између потреба (перформансе) и могућности (површина и потрошња). Организација „од-тачке-до-тачке“ омогућава смањење електричног оптерећења магистрале са циљем повећања укупног капацитета меморије, по цену употребе додатних ресурса у облику бафер чипова, одговарајућег повећања потрошње, као и додатног кашњења сигнала због баферовања.

Пример система са организацијом „од-тачке-до-тачке“ је FB-DIMM (енг. Fully Buffered DIMM) архитектура [40] у оквиру које се уводи AMB (енг. Advanced Memory Buffer) [41] бафер чип између меморијског контролера и DRAM чипова на меморијском модулу. Код FB-DIMM архитектуре се уместо паралелног преноса података преко меморијске магистрале уводи серијализација сигнала магистрале података. На тај начин се омогућава пренос веће количине података коришћењем истог броја пинова, али уз повећање фреквенције рада меморијске магистрале. AMB чип компензује настале сметње на магистрали тако што баферује све сигнале и притом аутономно ради корекцију грешака, не оптерећујући меморијски контролер или процесор. Такође, користи и BLFC (енг. Bit Lane Failover Correction) механизам како би се идентификовале линије које су подложне грешкама и избациле из употребе, чиме се значајно може смањити стопа грешака. Поред тога, пошто се операције читања и уписа баферују, меморијски контролер може да их шаље истовремено, чиме се повећава укупан проток. Постоје и хибридне

организације које имају за циљ да искористе предности две или више различитих организација. Пример за то је хибрид паралелно/серијске и организације „од-тачке-до-тачке“ примењен у серверским системима са RDIMM и LRDIMM модулима.

Други начин да се превазиђе проблем са интегритетом сигнала који настаје због повећавања електричног оптерећења адресне и командне магистрале је да се број циклуса током којих су адресни и командни сигнали активни повећа са један на два. Овај режим је познат и као „2Т“ командни мод. Мана овог приступа је што се проток на командној магистрали преполовљује у односу на „1Т“ командни мод, што може битно да утиче на перформансе.

3 Поређење са постојећим решењима

Методологија за мерење и анализу перформанси DRAM меморија предложена у оквиру дисертације се базира на ACDC моделу перформанси DRAM меморија. Један од кључних концепата у овом ACDC моделу јесте карактеризација индивидуалних циклуса на меморијској магистралу на слободне и заузете помоћу рачунања режиског времена узрокованог временским ограничењима DRAM протокола. Неке од постојећих методологија за мерење и анализу перформанси се такође базирају на идеји идентификације слободних и заузетих циклуса на магистралу, али немају неопходну тачност и општост у карактеризацији, нити довољно добро дефинисану метрику. Друге методологије, пак, рачунају временска ограничења, али не са циљем карактеризације циклуса, већ ради провере усаглашености са DRAM протоколом, док се питање моделирања перформанси посматра одвојено. У оквиру овог поглавља ће предложена методологија бити упоређена са постојећим методологијама по свим релевантним параметрима, а њене предности ће бити детаљно образложене.

3.1 Методологије за мерење и анализу перформанси DRAM меморија

У пракси, мерење и анализа перформанси су базирани на примени две методе: аналитичкој методи и методи симулације. Аналитичке методе се базирају на математичком моделирању система, а методе симулације на креирању извршног модела система који опонаша реални систем у свим његовим битним елементима.

Постоје и хибридне методе, које нпр. користе методу симулације за генерисање резултата који служе као улазни параметри аналитичког модела.

Методологија за мерење и анализу перформанси треба да опише не само како се принципијелно мере перформансе, већ све практичне и теоријске аспекте овог процеса, укључујући: како се прикупљају улазни подаци, како се моделирају перформансе, која метрика се користи за мерење перформанси, како се добијени резултати користе у анализи итд. Већина методологија које се користе у пракси има значајне недостатке у вези са неким од ових аспеката.

3.1.1 Методологије базиране на аналитичком моделовању перформанси

Аналитички модел перформанси је математичка апстракција, најчешће у виду алгебарских формула, која генерише вредности параметара перформанси реалног система на основу улазних података као што су меморијски захтеви који стижу меморијском контролеру, DRAM команде које се шаљу на меморијску магистралу и сл. Због комплексности организације и рада DRAM меморије, за формирање аналитичких модела се морају уводити различите претпоставке које омогућавају поједностављено моделовање перформанси. То у неким случајевима омогућава добијање употребљивих резултата, али никада потпуно тачних. Међутим, као што је поменуто у поглављу 1.5, виртуелизација и велики број меморијских клијената у модерним рачунарским системима чине да меморијски саобраћај постаје веома непредвидљив. У таквој ситуацији грешке услед апроксимација у аналитичким моделима постају неприхватљиво велике.

Примери аналитичких модела су хибридни аналитички модел [42] и RAD (Request Access Distance) аналитички модел [43]. Хибридни аналитички модел као улаз користи меморијске захтеве који су прошли арбитражу унутар меморијског контролера и стигли на улаз секвенцера који генерише DRAM команде. На тај начин се чува оригинални редослед меморијских захтева. За сваки од захтева се симулира генерисање DRAM команди помоћу једноставног интерног алгорита. Заузети циклуси се идентификују тако што се за низ команди читања и уписа између активирања и затварања једне врсте рачуна одговарајућа метрика у виду минималног временског растојања између команди. Сумирањем заузетих циклуса

током приступа адресама унутар једне врсте и заузетих циклуса приликом промене врста за сваку банку се рачуна укупно заузето време. На основу тога се рачуна број заузетих циклуса на магистрали и, коначно, ефикасност.

Основна идеја код RAD аналитичког модела је да се израчуна број циклуса који се мора утрошити на различите режијске послове наспрам циклуса који служе за пренос података. Циљ је да се израчуна ефикасност DRAM меморије, односно одреди колики је максималан могућ проток података, у условима примене разних радних оптерећења, варијације временских параметара, за различите конфигурације система и алгоритме арбитрације и секвенцирања меморијског контролера. У оквиру модела је учињен покушај да се метод за рачунање режијских циклуса формализује. То се постиже тако што се најпре идентификују главни фактори који узрокују појаву режијских циклуса, потом се ти фактори анализирају у контексту интеракције различитих DRAM команди и на крају се њихови ефекти квантификују помоћу RAD метрике, која специфицира минимално време које мора протећи између команди у одговарајућим сценаријима.

3.1.2 Методологије базиране на примени методе симулације

Симулациони модел је извршни модел, најчешће у виду кода писаног у вишем програмском језику или кода у неком језику за опис хардвера, у коме је апстраховано понашање кључних компоненти система у времену. Симулациони модели који се користе за анализу перформанси су обично бихевиорални (енг. behavioral), односно садрже опис функционалности који је релевантан за архитектурални ниво анализе, али не и друге детаље који нису од значаја у том контексту. Ови модели су знатно једноставнији, а тиме и бржи за симулирање, од RTL описа који се користи за имплементацију.

Методологије за анализу перформанси DRAM меморија базиране на методи симулације се ослањају на функционалне моделе DRAM кола и делова меморијског подсистема, пре свега меморијског контролера. Помоћу интерних сигнала модела идентификују се одговарајућа стања и на основу њих се рачунају параметри меморијских перформанси, при чему је сам модел перформанси DRAM меморија независан од симулационог модела.

DRAMsim [44] и његов наследних DRAMsim2 [45] су примери симулационих окружења који омогућавају да се рад меморијског подсистема може симулирати са тачношћу на нивоу циклуса. У оквиру симулационог окружења имплементирани су модели DRAM меморијских кола, као и целог меморијског контролера са секвенцером DRAM команди и командним баферима за сваку банку. Модели подржавају DDR2 и DDR3 протоколе. DRAMsim користи „табелу протокола“ како би се рачунала временска ограничења између парова DRAM команди у циљу верификације сагласности са одговарајућим DRAM протоколом. Међутим, DRAMsim не имплементира неки одређени модел DRAM перформанси и стога не генерише резултате који се могу користити у пракси, осим метрике базиране на директно опсервабилним параметрима, као што је проток на магистрали података.

Ramulator [46] је функционални DRAM симулатор који је пројектован тако да се лако може проширити како би подржао неки нов DRAM протокол. Тренутно подржава већину новијих протокола, укључујући DDR3/4, LPDDR3/4, GDDR5, Wide I/O 1/2 и HBM, као и неке експерименталне протоколе. Аутори наводе да је око 3 пута бржи од DRAMsim2 симулатора. Ни овај симулатор не имплементира одређени модел DRAM перформанси, као ни DRAMsim.

У пракси, методологије базиране на примени методе симулације имају више недостатака. Пре свега, неопходно је креирање тачног модела меморијског подсистема, као и одговарајуће окружење за симулацију радног оптерећења на том моделу. Ако симулациони модел у свим елементима не одговара стварном систему, резултујући меморијски саобраћај неће бити веродостојан, па није могуће поуздано мерити ни анализирати перформансе меморије. Такође, генерисање одговарајућег радног оптерећења је често непремостив проблем, јер се утицаји различитих елемената система, као што је рад оперативног система и симултани рад различитих компоненти, тешко моделирају. Још један потенцијални проблем је што, с обзиром да се потребни резултати генеришу динамички у току симулације, анализа перформанси сваке меморијске секвенце захтева ново покретање симулације. То често има велику цену узимајући у обзир дужину трајања симулација и ресурсе који се при том ангажују.

Коначно, метода симулације сама по себи не дефинише одређени модел DRAM перформанси, метрику за мерење DRAM перформанси, нити омогућава решавање проблема интерпретације добијених резултата. Сва ова питања морају бити решена независно, у оквиру методологије.

3.1.3 Предложена методологија

Предложена методологија се заснива на тачној карактеризацији циклуса на меморијској магистралу на слободне, активне и режијске. Карактеризација се ради помоћу новог модела перформанси DRAM меморија под називом ACDC (поглавље 4). ACDC модел омогућава прецизну карактеризацију на основу временских ограничења специфицираних DRAM протоколом. Модел има облик коначног аутомата који је параметризован тако да може да подржи различите врсте DRAM уређаја, конфигурације меморије и вредности временских параметара, што значајно повећава ефикасност у примени и поједностављује употребу.

У оквиру методологије је предложена потпуно нова метрика за мерење перформанси DRAM меморије, која је применљива и на контролну магистралу и на магистралу података. На основу нове метрике је дефинисан и метод за тачну динамичку процену максимума перформанси DRAM меморије, чиме се решава проблем интерпретације измерених вредности. Увођењем ових нових компоненти, предложена методологија решава кључне проблеме у домену мерења и анализе перформанси DRAM меморија.

3.2 Поређење методологија за мерење и анализу перформанси DRAM меморија

Поређење методологија за мерење и анализу перформанси DRAM меморија се врши према критеријумима који су од највећег значаја са теоријског и практичног аспекта. Упоредне методологије су изабране као репрезентативне на основу примењених метода и модела DRAM перформанси.

Брзина рада у оквиру одређене методологије је у највећој мери одређена природом модела DRAM перформанси на ком се методологија базира. Аналитички

модели су брзи, јер захтевају минимално процесирање за сваку трансакцију, које се обично своди на рачунање одговарајућих формула. Симулациони модели су инхерентно спори, јер је за сваку трансакцију, или још чешће, за сваки циклус, потребно симулирати рад читавог меморијског подсистема (у случају trace-driven симулација) или читавог рачунарског система (у случају execution-driven симулација). Модели базирани на раду коначног аутомата су брзи, јер је за сваку трансакцију довољно само прочитати табелу прелаза.

Улаз RAD аналитичког модела чине Read и Write DRAM команде, као и време појаве команди на контролној магистралу, на онову којих се применом RAD метрике рачуна минимално време које мора протећи између сваке од њих. Пошто модел не укључује друге DRAM команде, чак ни комбиноване команде за читање и упис (Read with Auto-precharge и Write with Auto-precharge), његова примена у многим доменима је лимитирана.

Улаз DRAMsim и Ramulator модела чине захтеви за приступ меморији који стижу меморијском контролеру. Интерни модел меморијског контролера симулира процес арбитража између ових захтева и врши њихово секвенцирање у низ DRAM команди који се шаљу на меморијску магистралу. DRAMsim и Ramulator имплементирају свеобухватније моделе за рачунање режијских циклуса него RAD аналитички модел. DRAMsim модел укључује и Activate, Precharge и Refresh команде, али неке комбиноване команде и даље недостају, као што су Read with Auto-precharge, Write with Auto-precharge и Precharge all banks, док Ramulator подржава цео скуп DRAM команди. Такође, код DRAMsim нису размотрени и неки специјални случајеви, као што је прелаз од Read на Write команду када се адресирају два различита ранка или прекинуте команде читања и писања када је дужина трансакције преноса података једнака 8. За подржане протоколе, симулациони модели су у оба случаја функционално тачни на нивоу циклуса.

DRAMsim и Ramulator не имплементирају ни један одређен модел DRAM перформанси, тако да се метрике које се не добијају директно на основу опсервабилних параметара, као што су искоришћење и ефикасност, не могу мерити. Чак и када би табела протокола била комплетна и тачна, перформансе базиране на измереним (неопсервабилним) режијским циклусима не би биле потпуно тачне. За

то је потребно да и модел перформанси буде тачан и да у себи укључује не само command-command, већ и data-data временска ограничења (види поглавље 4.3).

ACDC модел перформанси је имплементиран у виду коначног аутомата. Оваква имплементација има низак ниво комплексности и захтева знатно мањи напор при коришћењу у односу на методологије базиране на методи симулације. Улазни подаци модела су DRAM трансакције на меморијској магистрали. То је добро са становишта опсервабилности, јер је интерфејс између меморијског контролера и DRAM меморије типично доступан за корисника, како у оквиру развојног окружења, тако и у физичком систему. Слично важи и за опсервабилност DRAM команди које се користе у RAD аналитичком моделу. ACDC модел и RAD модел су независни од архитектуре и дизајна система на коме се генеришу трансакције, па могу да процесирају трансакције из произвољних извора и да буду имплементирани на различитим платформама.

Насупрот томе, интерфејс између меморијског контролера и осталих делова рачунарског система преко кога долазе захтеви за приступ меморији, што је место где се скупљају улазне трансакције у DRAMsim моделу, најчешће није лако доступан. Овај интерфејс може бити интегрални део веће IP целине, као што је CPU или SOC, што га може чинити недоступним у развојном окружењу, нпр. ако изворни код није доступан због заштите интелектуалне својине, или у физичком систему, где нпр. имплементација целог система може бити на једном чипу, тако да се интерним интерфејсима не може физички приступити. Поред тога, због инхерентне зависности која постоји између улазних захтева, дизајна меморијског контролера и DRAM перформанси, симулирање обраде захтева који су генерисани на једној платформи неће дати исте резултате на другој платформи осим ако је меморијски подсистем у оба случаја исти.

Додатна методолошка разлика између RAD и DRAMsim модела са једне стране и ACDC модела са друге стране је што се прва два модела базирају на функционалној и временској анализи генеричког DRAM кола, док се ACDC модел базира на моделу који је дефинисан на основу DRAM стандарда. Овај други приступ има вишеструке предности. Сви физички и имплементациони аспекти DRAM кола који имају утицај на тајминг сигнала већ су апстраховани у оквиру

Модел	RAD	DRAMsim/Ramulator	ACDC
Метод	Аналитички	Симулација	Коначни аутомат
Брзина	Брз	Спор	Брз
Улазни подаци	RW команде	RW захтеви	DRAM трансакције
Комплетност	Некомплетан	Делимично/Комплетан	Комплетан
Опсервабилност параметара	Спољна	Унутрашња	Спољна
DRAM параметри	Генерички DRAM	Генерички DRAM/ DRAM спецификација	DRAM спецификација
Комплексност	Ниска	Висока	Ниска
Напор потребан за коришћење	Средњи	Висок	Низак
Зависност од платформе	Независан	Зависан	Независан
Преносивост	Висока	Ниска	Висока
Тачност	Апроксимативан	Тачан	Тачан
Процена максимума перформ.	Апроксимативан	НEMA	Тачан

Табела 3.1 – Поређење са постојећим методологијама

DRAM стандарда у облику низа ограничења, тако да је ACDC модел по дефиницији минималан, комплетан и директно применљив на анализу DRAM саобраћаја у пракси. Насупрот томе, модели базирани на генеричкој представи DRAM кола су обично некомплетни и имају проблема са компатибилношћу. На пример, неки генерички временски параметри који се користе у DRAMsim окружењу би најпре морали да буду формално интерпретирани, јер се не могу лако изразити преко параметара из JEDEC DRAM спецификација.

Тачност код апроксимативних модела, као што су аналитички, зависи од радног оптерећења. Модели са инхерентно нижом тачношћу могу да дају довољно тачне резултате ако је радно оптерећење такво да не истиче слабости модела. На пример, RAD аналитички модел ће дати резултате веома велике тачности ако се радно оптерећење састоји од дугачких секвенци трансакција за читање и упис. Одговор на питање да ли је тачност неког модела прихватљива зависи од тога каква анализа је потребна, али терет доказивања да је анализа тачна увек лежи на самом

кориснику. Овај проблем није тривијалан и може се потпуно елиминисати само применом модела са тачношћу на нивоу циклуса.

Чак и када су перформансе тачно измерене, ако није могуће наћи одговарајући теоријски максимум, јавља се проблем интерпретације резултата. Од свих методологија, једино методологија базирана на ACDC моделу омогућава тачну процену теоријског максимума на било ком нивоу грануларности. Преглед кључних карактеристика упоређених методологија је дат у Табели 3.1.

4 Моделовање перформанси DRAM меморија

Предложена методологија за анализу перформанси DRAM меморија се базира на новом моделу DRAM перформанси под називом ACDC (енг. Accurate Characterization of DRAM Cycles)⁸. ACDC модел обезбеђује формални механизам за једнозначну карактеризацију циклуса на меморијској магистрали као слободних, активних или режијских. Тиме се решава један од критичних проблема у анализи перформанси DRAM меморија – немогућност да се разликују режијски циклуси од слободних циклуса.

Модел је базиран на фундаменталном запажању да се рад DRAM меморије на нивоу DRAM протокола, односно на нивоу трансакција на меморијској магистрали, може представити помоћу коначног аутомата (енг. finite state machine). Стања овог аутомата одговарају DRAM командама, док су прелази између команди, односно између стања аутомата, условљени тренутним статусом DRAM меморије, адресом наредне команде и временским ограничењима датим у спецификацији DRAM кола. Овај коначни аутомат се може трансформисати у коначни аутомат са истим стањима и истим прелазима између стања, коме је улаз тренутни статус DRAM меморије и адреса наредне команде, а излаз одговарајућа временска ограничења. Помоћу овако трансформисаног коначног аутомата, применом

⁸ Назив је изабран са намером да асоцира на акроним AC/DC (енг. Alternate Current / Direct Current) који се често користи на исправљачким уређајима, тј. уређајима који врше претварање наизменичне у једносмерну струју, јер подсећа на то како ACDC модел претвара „наизменичне“ динамичке трансакције на меморијској магистрали у „једносмерне“ карактеризоване меморијске циклусе.

алгоритма дефинисаног у ACDC моделу, могуће је потом извршити једнозначну карактеризацију циклуса.

4.1 Ограничења у DRAM проточној обради

DRAM проточна обрада има ограничења у виду редоследа догађаја и временских ограничења између одговарајућих догађаја који се морају поштовати како би рад DRAM кола био исправан. Под догађајем се овде подразумева активирање одговарајућег скупа контролних сигнала (попут оних на Слици 2.5), са циљем да се обави нека активност. Догађај се, ради лакшег праћења, може означити и именом одговарајуће активности. На пример:

Precharge : $\overline{EQ}, \overline{WL}, \overline{SEN}, \overline{SEP}, \overline{CS}, \overline{WE}$

Activate : $\overline{EQ}, WL, SEN, \overline{SEP}, \overline{CS}, \overline{WE}$

Sense : $\overline{EQ}, WL, \overline{SEN}, SEP, \overline{CS}, \overline{WE}$

Read : $\overline{EQ}, WL, \overline{SEN}, SEP, CS, \overline{WE}$

Write : $\overline{EQ}, WL, \overline{SEN}, SEP, CS, WE$

Restore : $\overline{EQ}, WL, \overline{SEN}, SEP, \overline{CS}, \overline{WE}$

I/O gating : ...

Data buffer : ...

Data bus : ...

Временска ограничења имају различите функционалне и физичке узроке, као што су: време прерасподеле наелектрисања, максималне вршне струје, избегавања колизије приликом приступа ресурсима итд. (види поглавље 2.3). Ова ограничења се могу изразити у следећем облику:

услов : претходни догађај → следећи догађај

што се може интерпретирати на следећи начин: “прелаз од *претходног догађаја* на *следећи догађај* је дозвољен само ако је задовољен *услов*”. Специфицирањем свих дозвољених прелаза осигурава се исправан редослед и време извршења свих догађаја код DRAM проточне обраде. На пример, ако су A и B меморијске адресе, Δt време између два догађаја, а t_x одговарајући временски параметри, ограничења би се могла формулисати на следећи начин:

$$bank(A) = bank(B), \Delta t \geq t_{Precharge} : Precharge A \rightarrow Activate B$$

$$bank(A) = bank(B), \Delta t \geq t_{Activate} : Activate A \rightarrow Sense B$$

$$bank(A) = bank(B), \Delta t \geq t_{Sense} + t_{I/O\ gating} + t_{Data} : Sense A \rightarrow Sense B$$

$$bank(A) = bank(B), \Delta t \geq t_{Sense} + t_{Restore} : Sense A \rightarrow Sense B$$

$$rank(A) \neq rank(B), \Delta t \geq t_{Data} + t_{Bus\ control} : Sense A \rightarrow Sense B$$

$$bank(A) \neq bank(B), \Delta t \geq t_{Max\ current} : Activate A \rightarrow Activate B$$

...

У листи ограничења се може десити да се више услова односи на исти прелаз. У том случају се одговарајућа ограничења могу спојити у једно ограничење чији услов је конјункција услова индивидуалних ограничења. На пример, треће и четврто ограничење у претходном примеру се може спојити у једно ограничење:

$$bank(A) = bank(B), \Delta t \geq \max(t_{Sense} + t_{I/O\ gating} + t_{Data}, t_{Sense} + t_{Restore}) : Sense A \rightarrow Sense B$$

4.2 Прелази између DRAM команди

Свакој DRAM команди одговара један или више догађаја који се иницирају њеним извршењем. На пример:

Activate : *Sense, Activate, Restore*

Read : *I/O gating, Data buffer, Data bus*

Write : Data bus, Data buffer, I/O gating, Write

Read with autoprecharge : I/O gating, Data buffer, Data bus, Precharge

...

Имајући то у виду, ограничења код прелаза између DRAM команди се могу представити на следећи начин:

услов : претходна команда \rightarrow следећа команда

$$\text{услов} = \bigwedge_{t \in T} \text{услов}(t)$$

$$T = \{e_p \rightarrow e_n \mid \exists(e_p \rightarrow e_n), e_p \in E_p, e_n \in E_n\}$$

$$E_p = \{\text{догађаји уницирани од стране претходне команде}\}$$

$$E_n = \{\text{догађаји уницирани од стране следеће команде}\}$$

$$\text{услов}(\emptyset) = \perp$$

Код правилно дефинисаних DRAM протокола скуп услова за прелазе између DRAM команди садржи сва ограничења DRAM проточне обраде, чинећи ова два скупа еквивалентним у односу на услове за извршавање DRAM операција. Уколико се формулише листа ограничења за прелаз између сваке две DRAM команде, добија се скуп ограничења који у потпуности дефинишу рад DRAM меморије на нивоу DRAM протокола.

Претходна команда и *следећа команда* које су специфициране у оквиру ограничења DRAM команди не морају нужно бити суседне на DRAM командној магистрали. Контекст у коме се посматра прелаз између две команде је одређен *условом*. На пример, *претходна команда* и *следећа команда* можда морају да задовољавају услов да адресирају локације у истој банки или ранку. Ово се може илустровати разматрањем следеће секвенце команди на командној магистрали:

Activate A, Activate B, Read a, Read b, ...

Претпоставимо да важи $bank(a)=bank(A)$, $bank(b)=bank(B)$, $bank(a,A)\neq bank(b,B)$, где су a и b меморијске адресе, а A и B врсте којима припадају подаци са одговарајућих адреса a и b . Познато је да постоје временска ограничења која дефинишу минимално време које мора протећи између две *Activate* команде, између *Activate* и *Read* команде издате истој банки и између две *Read* команде (види Табелу 4.1). Приликом дефинисања ограничења која важе за дату секвенцу команди, не треба разматрати прелазе између DRAM команди у редоследу у ком се оне појављују на командној магистрали. То би довело до разматрања следећих ограничења, што је погрешно:

$$\begin{aligned}
 & bank(A) \neq bank(B), \Delta t \geq t_{RRD} + t_{FAW} : \text{Activate } A \rightarrow \text{Activate } B \\
 & bank(B) \neq bank(a) : \text{Activate } B \rightarrow \text{Read } a \quad (\text{нема временског услова}) \\
 & rank(a) = rank(b), \Delta t \geq \max(t_{CCD}, t_{READ}) : \text{Read } a \rightarrow \text{Read } b \\
 & \dots
 \end{aligned}$$

Уместо тога, прелазе треба разматрати у контексту одговарајућих услова. Ограничења која заправо треба разматрати су:

$$\begin{aligned}
 & bank(A) \neq bank(B), \Delta t \geq t_{RRD} + t_{FAW} : \text{Activate } A \rightarrow \text{Activate } B \\
 & bank(A) = bank(a), \Delta t \geq t_{RCD} - t_{AL} : \text{Activate } A \rightarrow \text{Read } a \\
 & bank(B) = bank(b), \Delta t \geq t_{RCD} - t_{AL} : \text{Activate } B \rightarrow \text{Read } b \\
 & rank(a) = rank(b), \Delta t \geq \max(t_{CCD}, t_{READ}) : \text{Read } a \rightarrow \text{Read } b \\
 & \dots
 \end{aligned}$$

4.3 Коначни аутомат прелаза између DRAM команди

У поглављу 4.2 је показано да је рад DRAM меморије на нивоу DRAM протокола, односно на нивоу трансакција на меморијској магистрали, потпуно дефинисан ограничењима у којима фигуришу прелазе између DRAM команди и одговарајући услови за прелаз. Пошто се у DRAM спецификацијама сва временска ограничења дефинишу управо на нивоу трансакција на магистрали, могуће је

изразити сва DRAM ограничења из DRAM спецификације у облику условних прелаза између команди.

Услови из ограничења се могу класификовати у три категорије: просторни, оперативни и временски. Просторна компонента дефинише да ли претходна и следећа команда код прелаза адресирају исту банку или ранк. Оперативна компонента дефинише оперативне услове под којима се примењује ограничење, као што је дужина трансакције преноса података или статус завршетка претходне команде. Временска компонента дефинише временска ограничења за прелаз. DRAM ограничења се стога могу изразити у следећем облику:

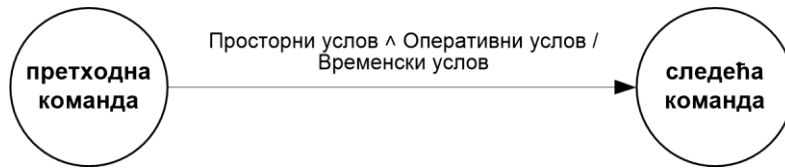
$$\begin{aligned} & \text{просторна услов} \wedge \text{оперативни услов} \wedge \text{временски услов} : \\ & \text{претходна команда} \rightarrow \text{следећа команда} \end{aligned} \quad (4.1)$$

Услови у DRAM ограничењу одређују када се прелаз сме десити. Ова информација је потребна меморијском контролеру када планира DRAM трансакције на меморијској магистрали. Да би се извршила карактеризација DRAM циклуса поребно је учинити супротно – одредити временска ограничења која су важила приликом прелаза након што се он већ десио (тј. *post factum*):

$$\begin{aligned} & \text{просторни услов} \wedge \text{оперативни услов} \wedge \text{претходна команда} \rightarrow \\ & \text{следећа команда} : \text{временски услов} \end{aligned} \quad (4.2)$$

Овај израз се може интерпретирати на следећи начин: „Ако се под *просторним условом* и *оперативним условом* десио прелаз са *претходне команде* на *следећу команду*, онда је важио *временски услов*“. Доказ да су изрази (4.1) и (4.2) логички еквивалентни је дат у Прилог А. Релација (4.2) се може представити на више начина: табеларно, текстуално, у облику графа, аотираних дијаграма временских сигнала итд. JEDEC DRAM спецификације, на пример, користе неколико од ових начина да специфицирају DRAM временска ограничења.

На основу *post factum* репрезентације условних прелаза DRAM команди могуће је конструисати коначни аутомат DCT FSM (енг. DRAM Command



Слика 4.1 – Прелаз између две команде у DCT FSM

Улазе представљају просторни услов (из адресе) и оперативни услов (из команде или регистра), а излаз временски услов који је важио приликом преласка из претходне у следећу команду.

Transition FSM) са следећим особинама: стања одговарају DRAM командама, прелази између стања су условљени просторним и оперативним условима, а излаз се састоји од временских ограничења која су важила приликом преласа (Слика 4.1). Временска ограничења генерисана од стране DCT FSM су потребна за идентификацију режијских циклуса.

Временски услов за прелаз између две DRAM команде представља минимално време које мора протећи између команди тако да одговарајућа DRAM временска ограничења буду задовољена. DRAM временска ограничења се могу класификовати у четири категорије у односу на то да ли се односе на команде (енг. command) или податке (енг. data): CC (command-command) ограничења, CD (command-data) ограничења, DC (data-command) ограничења и DD (data-data) ограничења. Пошто се ограничења из више категорија могу односити на исти прелаз, између две команде могу постојати различите временске путање. Временско ограничење је одређено најдужом од тих путања.

Постоје две врсте CC временских путања: једна садржи само CC ограничења, а друга сва ограничења на CD-DD-DC путањи. Ако су t_{CC} , t_{DC} , t_{CD} и t_{DD} кумулативна CC, CD, DC, and DD DRAM временска ограничења на одговарајућој временској путањи, онда се command-command временски услов може рачунати као:

$$\Delta t_{CC} \geq \max \left[\max_{\forall CC \text{ path}} (t_{CC}), \max_{\forall CD-DD-DC \text{ path}} (t_{CD} + t_{DD} + t_{DC}) \right] \quad (4.3)$$

Поред тога, морају бити задовољена и data-data временска ограничења за прелазе између две команде преноса података (Read, Write, Read with auto-precharge и Write with auto-precharge). Слично као и код command-command временског услова, data-data временски услов се може израчунати као:

$$\Delta t_{DD} \geq \max \left[\max_{\forall DD \text{ path}} (t_{DD}), \max_{\forall DC-CC-CD \text{ path}} (t_{DC} + t_{CC} + t_{CD}) \right] \quad (4.4)$$

Временски услов за прелаз између две произвољне DRAM команде је може рачунати на следећи начин:

$$\Delta t \geq \max(\Delta t_{CC}, \Delta t_{DD}) \quad (4.5)$$

Комплетан DCT FSM за JEDEC DDR2 SDRAM је дат у Табели 4.1. Вредности у табели су изведене на основу временских дијаграма у Прилог Б.

4.4 Карактеризација меморијских циклуса

Први корак у процесу карактеризације циклуса на меморијској магистрали је да се идентификују одговарајући прелаз између команди у DCT FSM за сваку послату команду. То није тривијално, јер, као што је поменуто у поглављу 4.2, претходна команда и следећа команда нису увек суседне на командној магистрали. Да би се идентификовао одговарајући прелаз, поребно је за сваку банку и ранк у систему чувати информације о последњој команди која их је адресирала. Једна од ових команди ће постати „претходна команда“, док ће последња послата команда бити „следећа команда“ у прелазу.

Послата команда се упарује са кандидатима за „претходну команду“ на основу просторних и оперативних услова. Просторни услов се одређује на основу

Претх. команда	Следећа команда	Услов	ACDC (command-command)	ACDC (data-data)
Read	Read	Различит ранк	$\geq t_{\text{READ}} + \max(t_{\text{BTT}}, t_{\text{ODT}})$	$\geq t_{\text{READ}} + \max(t_{\text{BTT}}, t_{\text{ODT}})$
		Исти ранк	$\geq \max(t_{\text{CCD}}, t_{\text{READ}})$	$\geq t_{\text{READ}}$
		Исти ранк/BL=8/ пр.	$2 \cdot t_{\text{CK}}$	$2 \cdot t_{\text{CK}}$
Read	Read with AP	Различит ранк	$\geq t_{\text{READ}} + \max(t_{\text{BTT}}, t_{\text{ODT}})$	$\geq t_{\text{READ}} + \max(t_{\text{BTT}}, t_{\text{ODT}})$
		Исти ранк	$\geq \max(t_{\text{CCD}}, t_{\text{READ}})$	$\geq t_{\text{READ}}$
		Исти ранк/BL=8/ пр.	$2 \cdot t_{\text{CK}}$	$2 \cdot t_{\text{CK}}$
Read	Write	Различит ранк	$\geq t_{\text{RL}} + t_{\text{READ}} + \max(t_{\text{BTT}}, t_{\text{ODT}}) - t_{\text{WL}}$	$\geq t_{\text{READ}} + \max(t_{\text{BTT}}, t_{\text{ODT}})$
		Исти ранк	$\geq t_{\text{RL}} + t_{\text{READ}} - t_{\text{WL}}$	$\geq t_{\text{READ}}$
Read	Write with AP	Различит ранк	$\geq t_{\text{RL}} + t_{\text{READ}} + \max(t_{\text{BTT}}, t_{\text{ODT}}) - t_{\text{WL}}$	$\geq t_{\text{READ}} + \max(t_{\text{BTT}}, t_{\text{ODT}})$
		Исти ранк	$\geq t_{\text{RL}} + t_{\text{READ}} - t_{\text{WL}}$	$\geq t_{\text{READ}}$
Read	Precharge	Иста банка	$\geq t_{\text{AL}} + t_{\text{READ}} - t_{\text{CCD}} + t_{\text{RTP}}$	$\geq t_{\text{READ}}$
Read	Precharge all banks	Исти ранк	$\geq t_{\text{AL}} + t_{\text{READ}} - t_{\text{CCD}} + t_{\text{RTP}}$	$\geq t_{\text{READ}}$
Read	Activate	Иста банка	НЕЛЕГАЛНО	НЕЛЕГАЛНО
Read	Refresh	Исти ранк	НЕЛЕГАЛНО	НЕЛЕГАЛНО
Read with AP	Read	Иста банка	НЕЛЕГАЛНО	НЕЛЕГАЛНО
Read with AP	Read with AP	Иста банка	НЕЛЕГАЛНО	НЕЛЕГАЛНО
Read with AP	Write	Иста банка	НЕЛЕГАЛНО	НЕЛЕГАЛНО
Read with AP	Write with AP	Иста банка	НЕЛЕГАЛНО	НЕЛЕГАЛНО
Read with AP	Precharge	Иста банка	$\geq t_{\text{AL}} + t_{\text{READ}} - t_{\text{CCD}} + t_{\text{RTP}}$	$\geq t_{\text{READ}}$
Read with AP	Precharge all banks	Исти ранк	$\geq t_{\text{AL}} + t_{\text{READ}} - t_{\text{CCD}} + t_{\text{RTP}}$	$\geq t_{\text{READ}}$
Read with AP	Activate	Иста банка	$\geq t_{\text{AL}} + t_{\text{READ}} - t_{\text{CCD}} + t_{\text{RTP}} + t_{\text{RP}} + t_{\text{FAW}}^*$	$\geq t_{\text{READ}}$
Read with AP	Refresh	Исти ранк	$\geq t_{\text{AL}} + t_{\text{READ}} - t_{\text{CCD}} + t_{\text{RTP}} + t_{\text{RP}}$	$\geq t_{\text{READ}}$
Write	Read	Различит ранк	$\geq t_{\text{WL}} + t_{\text{WRITE}} + \max(t_{\text{BTT}}, t_{\text{ODT}}) - t_{\text{RL}}$	$\geq t_{\text{WRITE}} + \max(t_{\text{BTT}}, t_{\text{ODT}})$
		Исти ранк	$\geq \max(t_{\text{CCD}}, t_{\text{CWL}} + t_{\text{WRITE}} + t_{\text{WTR}})$	$\geq t_{\text{WRITE}} + t_{\text{WTR}} + t_{\text{RL}}$
Write	Read with AP	Различит ранк	$\geq t_{\text{WL}} + t_{\text{WRITE}} + \max(t_{\text{BTT}}, t_{\text{ODT}}) - t_{\text{RL}}$	$\geq t_{\text{WRITE}} + \max(t_{\text{BTT}}, t_{\text{ODT}})$
		Исти ранк	$\geq \max(t_{\text{CCD}}, t_{\text{CWL}} + t_{\text{WRITE}} + t_{\text{WTR}})$	$\geq t_{\text{WRITE}} + t_{\text{WTR}} + t_{\text{RL}}$
Write	Write	Различит ранк	$\geq t_{\text{WRITE}} + t_{\text{ODT}}$	$\geq t_{\text{WRITE}} + t_{\text{ODT}}$
		Исти ранк	$\geq \max(t_{\text{CCD}}, t_{\text{WRITE}})$	$\geq t_{\text{WRITE}}$
		Исти ранк/BL=8/ пр.	$2 \cdot t_{\text{CK}}$	$2 \cdot t_{\text{CK}}$
Write	Write with AP	Различит ранк	$\geq t_{\text{WRITE}} + t_{\text{ODT}}$	$\geq t_{\text{WRITE}} + t_{\text{ODT}}$
		Исти ранк	$\geq \max(t_{\text{CCD}}, t_{\text{WRITE}})$	$\geq t_{\text{WRITE}}$
		Исти ранк/BL=8/ пр.	$2 \cdot t_{\text{CK}}$	$2 \cdot t_{\text{CK}}$
Write	Precharge	Иста банка	$\geq t_{\text{WL}} + t_{\text{WRITE}} + t_{\text{WR}}$	$\geq t_{\text{WRITE}} + t_{\text{WR}}$
Write	Precharge all banks	Исти ранк	$\geq t_{\text{WL}} + t_{\text{WRITE}} + t_{\text{WR}}$	$\geq t_{\text{WRITE}} + t_{\text{WR}}$
Write	Activate	Иста банка	НЕЛЕГАЛНО	НЕЛЕГАЛНО
Write	Refresh	Исти ранк	НЕЛЕГАЛНО	НЕЛЕГАЛНО
Write with AP	Read	Иста банка	НЕЛЕГАЛНО	НЕЛЕГАЛНО
Write with AP	Read with AP	Иста банка	НЕЛЕГАЛНО	НЕЛЕГАЛНО
Write with AP	Write	Иста банка	НЕЛЕГАЛНО	НЕЛЕГАЛНО
Write with AP	Write with AP	Иста банка	НЕЛЕГАЛНО	НЕЛЕГАЛНО

Претходна команда	Следећа команда	Услов	ACDC (command-command)	ACDC (data-data)
Write with AP	Precharge	Иста банка	$\geq t_{WL}+t_{WRITE}+t_{WR}$	$\geq t_{WRITE}+t_{WR}$
Write with AP	Precharge all banks	Исти ранк	$\geq t_{WL}+t_{WRITE}+t_{WR}$	$\geq t_{WRITE}+t_{WR}$
Write with AP	Activate	Иста банка	$\geq t_{WL}+t_{WRITE}+t_{WR}+t_{RP}+t_{FAW}^*$	$\geq t_{WRITE}+t_{WR}+t_{RP}+t_{FAW}^*$
Write with AP	Refresh	Исти ранк	$\geq t_{WL}+t_{WRITE}+t_{WR}+t_{RP}$	$\geq t_{WRITE}+t_{WR}+t_{RP}$
Precharge	Read	Иста банка	НЕЛЕГАЛНО	НЕЛЕГАЛНО
Precharge	Read with AP	Иста банка	НЕЛЕГАЛНО	НЕЛЕГАЛНО
Precharge	Write	Иста банка	НЕЛЕГАЛНО	НЕЛЕГАЛНО
Precharge	Write with AP	Иста банка	НЕЛЕГАЛНО	НЕЛЕГАЛНО
Precharge	Precharge	Иста банка	–	–
Precharge	Precharge all banks	Исти ранк	–	–
Precharge	Activate	Иста банка	$\geq t_{RP}+t_{FAW}^*$	–
Precharge	Refresh	Исти ранк	$\geq t_{RP}$	–
Precharge all banks	Read	Исти ранк	НЕЛЕГАЛНО	НЕЛЕГАЛНО
Precharge all banks	Read with AP	Исти ранк	НЕЛЕГАЛНО	НЕЛЕГАЛНО
Precharge all banks	Write	Исти ранк	НЕЛЕГАЛНО	НЕЛЕГАЛНО
Precharge all banks	Write with AP	Исти ранк	НЕЛЕГАЛНО	НЕЛЕГАЛНО
Precharge all banks	Precharge	Исти ранк	–	–
Precharge all banks	Precharge all banks	Исти ранк	–	–
Precharge all banks	Activate	Исти ранк	$\geq t_{RP}+t_{FAW}^*$	–
Precharge all banks	Refresh	Исти ранк	$\geq t_{RP}$	–
Activate	Read	Иста банка	$\geq t_{RCD}-t_{AL}$	–
Activate	Read with AP	Иста банка	$\geq t_{RCD}-t_{AL}$	–
Activate	Write	Иста банка	$\geq t_{RCD}-t_{AL}$	–
Activate	Write with AP	Иста банка	$\geq t_{RCD}-t_{AL}$	–
Activate	Precharge	Иста банка	$\geq t_{RAS}$	–
Activate	Precharge all banks	Исти ранк	$\geq t_{RAS}$	–
Activate	Activate	Различита банка	$\geq t_{RRD}+t_{FAW}^*$	–
		Иста банка	$\geq t_{RC}+t_{FAW}^*$	–
Activate	Refresh	Исти ранк	НЕЛЕГАЛНО	НЕЛЕГАЛНО
Refresh	Read	Исти ранк	НЕЛЕГАЛНО	НЕЛЕГАЛНО
Refresh	Read with AP	Исти ранк	НЕЛЕГАЛНО	НЕЛЕГАЛНО
Refresh	Write	Исти ранк	НЕЛЕГАЛНО	НЕЛЕГАЛНО
Refresh	Write with AP	Исти ранк	НЕЛЕГАЛНО	НЕЛЕГАЛНО
Refresh	Precharge	Исти ранк	$\geq t_{RFC}$	–
Refresh	Precharge all banks	Исти ранк	$\geq t_{RFC}$	–
Refresh	Activate	Исти ранк	$\geq t_{RFC}+t_{FAW}^*$	–
Refresh	Refresh	Исти ранк	$\geq t_{RFC}$	–

Табела 4.1 – DCT FSM за DDR2 SDRAM

DCT FSM у табели је формулисан за 1Т мод. Прелази означени у табели са „–“ немају временска ограничења. Услови „Иста банка“ и „Различита банка“ подразумевају да обе банке припадају истом ранку. Параметри означени са „*“ се користе у случајевима када су применљиви.

банке и ранка из адресе претходне и следеће команде, а оперативни услов се одређује динамички. Потрага за одговарајућим прелазом може резултирати једним од следећих исхода:

1. У DCT FSM не постоји одговарајући прелаз: То значи да не постоје временска ограничења за тај прелаз, па је режијско време нула.
2. У DCT FSM постоји тачно један одговарајући прелаз: За рачунање режијског времена се користи временски услов тог прелаз.
3. У DCT FSM постоји више одговарајућих прелаз: Бира се прелаз који има најстрожи временски услов (јер меморијски контролер мора да поштује најстрожи услов када шаље команду), а режијско време се рачуна на основу тог временског услова.

Финални корак је карактеризација индивидуалних циклуса на контролној магистралаи и магистралаи података као активних, слободних или режијских на основу временског услова добијеног из DCT FSM. На командној магистралаи, само је циклус када се команда издаје активан. Наредни циклуси, све до истека времена специфицираног временским условом, се карактеришу као режијски. Циклуси који следе иза тога, све до појаве нове команде, се карактеришу као слободни. На магистралаи података, циклуси у току којих се преносе подаци се карактеришу као активни. Наредни циклуси, све до истека времена специфицираног временским условом, се карактеришу као режијски. Циклуси након тога, све до појаве нове трансакције преноса података, су слободни.

Активни циклуси генерисани од стране различитих команди се никада не преклапају, док режијски и слободни циклуси који настају као последица више различитих прелаз у DCT FSM могу да се преклопе. Ако се режијски циклуси преклапају са слободним циклусима, треба их карактеризовати као режијске. Ако се активни циклуси преклапају са слободним или режијским циклусима, треба их карактеризовати као активне.

Пример са низом DRAM команди и карактеризованим циклусима на магистралаи је дат на Слици 4.2. На слици се може уочити и „ивични“ ефекат: циклуси на магистралаи података који се налазе на почетку и циклуси на командној

Такт	Команда	Ранк	Банка	Врста	Колона	Циклуси на командној магистрали	Циклуси на магистрали података
1	Read	3	2		02d8	Активан	NC
2						Режијски	NC
3	Read	3	1		02ac	Активан	NC
4						Режијски	Активан
5	Read	3	1		02a4	Активан	Активан
6						Режијски	Активан
7						Режијски	Активан
8						Режијски	Активан
9						Слободан	Активан
10						Слободан	Режијски
11	Write	2	2		02cc	Активан	Слободан
12						Режијски	Слободан
13	Write	2	2		02d8	Активан	Активан
14						Режијски	Активан
15	Write	2	2		02dc	Активан	Активан
16						Режијски	Активан
17	Write	2	2		02c0	Активан	Активан
18						Режијски	Активан
19	Read	3	1		0294	Активан	Активан
20						Режијски	Активан
21	Read	3	2		0330	Активан	Режијски
22						Режијски	Активан
23						Слободан	Активан
24						Слободан	Активан
25	Read	3	2		0324	Активан	Активан
26						Режијски	Слободан
27	Read	3	2		0320	Активан	Слободан
28						Режијски	Активан
29						Слободан	Активан
30	Precharge	3	1			Активан	Активан
31						Режијски	Активан
32						Режијски	Режијски
33	Write	2	2		02c0	Активан	Слободан
34	Activate	3	1	0415		Активан	Слободан
35	Write	2	2		02c4	Активан	Активан
36						Режијски	Активан
37	Write	2	2		02d0	Активан	Активан
38						Режијски	Активан
39	Read	3	1		029c	Активан	Активан
40						Режијски	Активан
41	Read	3	1		035c	Активан	Режијски
42						Режијски	Активан
43						NC	Активан
44						NC	Активан
45						NC	Активан

Слика 4.2 – Карактеризација DRAM циклуса на меморијској магистрали

NC=Non-Characterizable. Timing constraints: BL=2, $t_{AL}=0$, $t_{CCD}=2$, $t_{CL}=3$, $t_{RTW}=3$, $t_{RTP}=2$, $t_{WR}=4$, $t_{WTR}=3$, $t_{RCD}=3$, $t_{RC}=15$, $t_{RRD}=3$, $t_{RAS}=15$, $t_{RP}=3$, $t_{RFC}=28$, $t_{RDRD}=3$, $t_{RDWR}=4$, $t_{WRRD}=2$, $t_{WRWR}=2$

магистралаи који се налазе на крају анализиране секвенце не могу да се карактеришу јер није могуће формирати пар DRAM команди који би чинио валидан прелаз у DCT FSM, у првом случају недостаје „претходна команда“, а у другом случају „следећа команда“.

5 Метрика

Перформансе DRAM меморије се могу дефинисати као количина података којој се може приступити у јединици времена. Због лакше анализе, перформансе се често изражавају и у виду процента од теоријског максимума, који је одређен физичким параметрима меморије, као што су ширина магистрале података, радна фреквенција, итд. и другим факторима, као што су ограничења која намеће DRAM протокол. Најчешће коришћене метрике у анализи перформанси DRAM меморија су искоришћење и ефикасност. Ови параметри се типично дефинишу само у контексту саобраћаја на магистрали података, док се анализа саобраћаја на командној магистрали обично ради одвојено и своди се на разматрање само одређених секвенци, најчешће мануелним путем.

У оквиру методологије предложене у овој дисертацији (поглавље 5.1) уводи се нова метрика која је истовремено применљива и на магистралу података и на контролну магистралу. На тај начин се превазилази дихотомија која је раније постојала у анализи рада DRAM меморија због нужности да се анализа трансакција на командној магистрали и магистрали података ради независно услед непостојања јединственог модела DRAM перформанси који би обухватао обе магистрале. Нова метрика је, насупрот томе, настала на основу модела перформанси DRAM меморије изложеног у поглављу 4, који на исти начин моделира перформансе обе компоненте меморијске магистрале.

На основу нове метрике су потом предложене две методе за процену теоријских граница перформанси DRAM меморије (поглавље 5.2). Једна метода даје горњу границу теоријских перформанси, а друга даје потенцијално реалистичнију процену теоријског максимума интерполацијом. Обе методе су

динамичке и аутоматски узимају у обзир утицај промена узрокованих флукуацијама радног оптерећења. Тиме се решава и други кључни проблем у анализи перформанси DRAM меморије – проблем интерпретације.

5.1 Метрика за мерење перформанси меморија

Код креирања нове метрике потребно је најпре дефинисати релевантне појмове, параметре и ознаке.

Дефиниција 5.1: *Командна магистрала* је део меморијске магистрале преко које меморијски контролер шаље командне трансакције DRAM меморији. *Магистрала података* је део меморијске магистрале преко које меморијски контролер шаље трансакције преноса података приликом уписа у DRAM меморију или преко које DRAM меморија шаље трансакције преноса података који су претходно прочитани ка меморијском контролеру.

Ознака за командну магистралу је $\langle CB \rangle$ (енг. Command Bus), а за магистралу података $\langle DB \rangle$ (енг. Data Bus). Ознака $\langle BUS \rangle$ се користи да се обележи била која од ове две магистрале, односно ова ознака се може заменити било са $\langle CB \rangle$ било са $\langle DB \rangle$.

Дефиниција 5.2: *Активан циклус* је циклус на магистрали података или командној магистрали током кога су одговарајући сигнали података, односно командни сигнали активни.

Уобичајено је да је сигнал активан ако има вредност логичке јединице, али то не мора бити правило. Код неких протокола се поларитет активних сигнала може чак и динамички мењати у току преноса ради смањења потрошње и очувања интегритета сигнала. Пример за то је DBI (енг. Data Bus Inversion) механизам код GGDR4/5, DDR3/4 и LPDDR4 протокола, где се поларитет битова на адресној и командној магистрали мења сваки пут када број битова логичке нуле или јединице (зависно од сигналне логике коју стандард користи) пређе половину. На тај начин се минимизује број битова који узрокују потрошњу енергије на терминалном крају. Број активних циклуса на магистрали се означава са $N_{\langle BUS \rangle ACTIVE}$.

Дефиниција 5.3: *Слободан циклус* је неактиван циклус на меморијској магистрали који је *могао* бити активан.

Разлог зашто неки слободан циклус није активан, иако би могао бити, може да буде то што нпр. није било захтева од стране меморијског контролера. Број слободних циклуса на магистрали се означава са $N_{(BUS) IDLE}$.

Дефиниција 5.4: *Режијски циклус* је неактиван циклус на меморијској магистрали који *није могао* бити активан.

Разлог зашто неактиван циклус није могао бити активан може да буде то што нпр. протокол не дозвољава његово коришћење због одређених ограничења. Број режијских циклуса на магистрали се означава са $N_{(BUS) OVERHEAD}$.

Дефиниција 5.5: *Заузет циклус* је сваки циклус који није слободан.

Број заузетих циклуса на магистрали се означава са $N_{(BUS) BUSY}$.

Дефиниција 5.6: *Искоришћење* је однос броја заузетих циклуса и укупног броја циклуса.

Пошто се активни и режијски циклуси не могу користити за слање нових трансакција, они су, по природи, заузети. Стога се искоришћење може дефинисати на следећи начин:

$$U_{(BUS)} = \frac{N_{(BUS) BUSY}}{N_{(BUS) TOTAL}} \quad (5.1)$$

$$N_{(BUS) TOTAL} = N_{(BUS) BUSY} + N_{(BUS) IDLE}$$

$$N_{(BUS) BUSY} = N_{(BUS) ACTIVE} + N_{(BUS) OVERHEAD}$$

Дефиниција 5.7: *Циклус на командној магистрали је искоришћен* ако је део трансакције слања команде. *Циклус на магистрали података је искоришћен* ако је део трансакције слања података.

Дефиниција 5.8: *Ефикасност* је однос броја циклуса који су искоришћени и броја заузетих циклуса.

Ефикасност се може посматрати и као искоришћење у периодима када је магистрала заузета.

У идеалној DRAM меморији нема режијских циклуса. То значи да осим трансакција за читање и упис података нема других трансакција. Трансакције за читање и упис су есенцијалне за рад меморије без обзира на DRAM архитектуру, а одговарајући циклуси на магистралама се зову есенцијални циклуси.

Дефиниција 5.9: *Есенцијални циклуси на командној магистралама су заузети циклуси који се генеришу услед команди читања и уписа у идеалној DRAM меморији.*

Ови циклуси заправо карактеришу радно оптерећење, а не рад DRAM меморије, јер се појављују без обзира на тренутно стање DRAM меморије и динамику трансакција на магистралама, већ само на основу захтева меморијских клијената у систему.

Дефиниција 5.10: *Функционални циклуси на командној магистралама су неесенцијални заузети циклуси.*

Ови циклуси карактеришу рад DRAM меморије за дато радно оптерећење, јер зависе од тренутног стања DRAM меморије и динамике трансакција на магистралама. Функционални циклуси су неопходни за коректан рад неидеалног DRAM кола. У случају да се есенцијални циклус изазван једном командом преклапа са функционалним циклусом изазваним другом командом, циклус се карактерише као есенцијални.

Дефиниција 5.11: *Есенцијални циклуси на магистралама података су заузети циклуси који се генеришу услед команди читања и уписа, што су, по дефиницији, активни циклуси на магистралама података.*

Дефиниција 5.12: *Функционални циклуси на магистралама података су неесенцијални заузети циклуси, што су, по дефиницији, режијски циклуси на магистралама података.*

Број есенцијалних циклуса на магистралама се означава са $N_{(BUS) ESSENTIAL}$, а број функционалних циклуса са $N_{(BUS) FUNCTIONAL}$.

На основу наведених дефиниција, може се извести следећа релација:

$$N_{(BUS) BUSY} = N_{(BUS) ESSENTIAL} + N_{(BUS) FUNCTIONAL} \quad (5.2)$$

Имајући у виду дефиниције 5.7-5.12 може се успоставити релација једнакости између појмова „искоришћен циклус“ у идеалној DRAM меморији и појма „есенцијални циклус“ у неидеалној меморији. Ефикасност се стога може дефинисати на следећи начин:

$$E_{(BUS)} = \frac{N_{(BUS) ESSENTIAL}}{N_{(BUS) BUSY}} \quad (5.3)$$

Ако је $f_{(BUS)}$ фреквенција рада одговарајућег дела меморијске магистрале⁹, онда је период времена који протекне током $N_{(BUS)}$ циклуса једнак $t_{(BUS)} = N_{(BUS)}/f_{(BUS)}$. Пошто су трансакције на обе магистрале у непосредној вези, то је укупно посматрано време на обе магистрале обично једнако, тј. важи $t_{(CB) TOTAL} = t_{(DB) TOTAL}$. Одатле следи:

$$\frac{N_{(CB) TOTAL}}{f_{(CB)}} = \frac{N_{(DB) TOTAL}}{f_{(DB)}} \quad (5.4)$$

По дефиницији, време које заузимају есенцијални циклуси на командној магистрали и есенцијални циклуси на магистрали података у посматраном периоду је једнако, тј. $t_{(CB) ESSENTIAL} = t_{(DB) ESSENTIAL} = t_{(DB) ACTIVE}$, одакле следи:

$$\frac{N_{(CB) ESSENTIAL}}{f_{(CB)}} = \frac{N_{(DB) ESSENTIAL}}{f_{(DB)}} \quad (5.5)$$

Из (5.3)-(5.5) сада следи:

⁹ Командна магистрала и магистрала података не морају да раде на истој фреквенцији.

$$\begin{aligned}
E_{\langle CB \rangle} &= \frac{N_{\langle CB \rangle ESSENTIAL}}{N_{\langle CB \rangle BUSY}} \\
&= \frac{f_{\langle CB \rangle} / f_{\langle DB \rangle} \cdot N_{\langle DB \rangle ESSENTIAL}}{N_{\langle DB \rangle TOTAL}} \cdot \frac{N_{\langle DB \rangle TOTAL}}{N_{\langle CB \rangle BUSY}} \\
&= \frac{f_{\langle CB \rangle} / f_{\langle DB \rangle} \cdot N_{\langle DB \rangle ESSENTIAL}}{N_{\langle DB \rangle TOTAL}} \cdot \frac{f_{\langle DB \rangle} / f_{\langle CB \rangle} \cdot N_{\langle CB \rangle TOTAL}}{N_{\langle CB \rangle BUSY}} \\
&= \frac{N_{\langle DB \rangle ESSENTIAL}}{N_{\langle DB \rangle BUSY}} \cdot \frac{N_{\langle DB \rangle BUSY}}{N_{\langle DB \rangle TOTAL}} \cdot \frac{N_{\langle CB \rangle TOTAL}}{N_{\langle CB \rangle BUSY}} \\
&= E_{\langle DB \rangle} \cdot U_{\langle DB \rangle} / U_{\langle CB \rangle}
\end{aligned}$$

ОДНОСНО:

$$E_{\langle CB \rangle} \cdot U_{\langle CB \rangle} = E_{\langle DB \rangle} \cdot U_{\langle DB \rangle} \quad (5.6)$$

Вредност производа у изразу (5.6) назива се *есенцијално искоришћење*, EU (енг. essential utilization)¹⁰:

$$EU_{\langle BUS \rangle} = \frac{N_{\langle BUS \rangle ESSENTIAL}}{N_{\langle BUS \rangle TOTAL}} \quad (5.7)$$

Овај параметар показује колики број циклуса од укупног броја циклуса на магистралу је био неопходан искључиво за пренос података. Ово је веома интуитивна метрика, што је, уз чињеницу да истовремено карактерише и командну магистралу и магистралу података, чини веома погодном за карактеризацију перформанси DRAM меморије:

$$EU_{\langle BUS \rangle} = E_{\langle BUS \rangle} \cdot U_{\langle BUS \rangle} \quad (5.8)$$

¹⁰ Згодна случајност је што се акроним EU састоји управо од слова Е и U који су и чиниоци производа који даје вредност тог параметра.

Како DRAM протокол мора да спречи колизије на магистрала података узроковане командама које приступају подацима у меморији, време које је магистрала података заузета мора бити укључено у command-command временска ограничења, а сходно томе и у време заузећа командне магистрале (поглавље 4.2). Стога је $t_{\langle CB \rangle BUSY} \geq t_{\langle DB \rangle BUSY}$, одакле следи:

$$U_{\langle CB \rangle} \geq U_{\langle DB \rangle} \quad (5.9)$$

а последично, из (5.6) и (5.8)-(5.9):

$$E_{\langle CB \rangle} \leq E_{\langle DB \rangle} \quad (5.10)$$

5.2 Теоријски максимум перформанси

Могућност да се тачно квантификује максимум перформанси је суштински важна у анализи перформанси DRAM меморија. Просто коришћење $EU_{\langle BUS \rangle} = 100\%$ као горње границе перформанси занемарује постојање режијских циклуса, што искривљује ефекте узроковане динамичким променама радног оптерећења и чини анализу нетачном. Прецизније границе перформанси DRAM меморије се могу дефинисати коришћењем нове метрике за мерење перформанси DRAM меморија.

5.2.1 Горња граница перформанси

Пошто је $EU_{\langle BUS \rangle} \leq \max(E_{\langle BUS \rangle}) \cdot \max(U_{\langle BUS \rangle})$, из (5.8)-(5.9) следи да се теоријски максимум перформанси DRAM меморије постиже за $U_{\langle CB \rangle} = 100\%$:

$$\max(EU_{\langle BUS \rangle}) = \max(E_{\langle CB \rangle}) \quad (5.11)$$

Ефикасност зависи од редоследа DRAM команди, који зависи од арбитрације и секвенцирања меморијских захтева унутар меморијског контролера. Када је редослед команди за дату секвенцу непроменљив, ефикасност је такође непроменљива.

Ако се одређен број слободних циклуса замени једнаким бројем заузетих циклуса ($\Delta N_{(BUS) BUSY} = -\Delta N_{(BUS) IDLE}$), уз претпоставку да укупан број циклуса остаје непромењен, може се из (5.1)-(5.3) извести да ће промена ефикасности бити:

$$\begin{aligned}
\Delta E_{(BUS)} &= \frac{N_{(BUS) ESSENTIAL} + \Delta N_{(BUS) ESSENTIAL}}{N_{(BUS) BUSY} + \Delta N_{(BUS) BUSY}} - \frac{N_{(BUS) ESSENTIAL}}{N_{(BUS) BUSY}} \\
&= \frac{\Delta N_{(BUS) ESSENTIAL} \cdot N_{(BUS) BUSY} - \Delta N_{(BUS) ESSENTIAL} \cdot \Delta N_{(BUS) BUSY}}{(N_{(BUS) BUSY} + \Delta N_{(BUS) BUSY}) \cdot N_{(BUS) BUSY}} \\
&= \frac{\Delta N_{(BUS) ESSENTIAL} \cdot \left(\frac{\Delta N_{(BUS) ESSENTIAL}}{N_{(BUS) ESSENTIAL}} - \frac{\Delta N_{(BUS) BUSY}}{N_{(BUS) BUSY}} \right)}{N_{(BUS) BUSY} \cdot \left(1 + \frac{\Delta N_{(BUS) BUSY}}{N_{(BUS) BUSY}} \right)} \\
&= E_{(BUS)} \cdot \frac{dN_{(BUS) ESSENTIAL} - dN_{(BUS) BUSY}}{1 + dN_{(BUS) BUSY}} \\
&= E_{(BUS)} \cdot \frac{1 + dN_{(BUS) ESSENTIAL} - (1 + dN_{(BUS) BUSY})}{1 + dN_{(BUS) BUSY}} \\
&= E_{(BUS)} \cdot \left(\frac{1 + dN_{(BUS) ESSENTIAL}}{1 + dN_{(BUS) BUSY}} - 1 \right)
\end{aligned}$$

где је $dN_{(BUS)} = \Delta N_{(BUS)}/N_{(BUS)}$. Посматрајмо добијену релацију:

$$\Delta E_{(BUS)} = E_{(BUS)} \cdot \left(\frac{1 + dN_{(BUS) ESSENTIAL}}{1 + dN_{(BUS) BUSY}} - 1 \right) \quad (5.12)$$

Ефикасност расте ако је $\Delta E_{(BUS)} > 0$, што се своди на услов $dN_{(BUS) ESSENTIAL} > dN_{(BUS) BUSY}$, тј. на услов да је однос додатих есенцијалних циклуса међу замењеним слободним циклусима већи него што је био у остатку посматраног периода. Пошто је (5.12) монотono растућа функција од $dN_{(BUS) ESSENTIAL}$, а тиме и од $\Delta N_{(BUS) ESSENTIAL}$, максимална ефикасност се постиже за $\max(\Delta N_{(BUS) ESSENTIAL})$, па из (5.3) следи:

$$\max(E_{\langle BUS \rangle}) = \frac{N_{\langle BUS \rangle ESSENTIAL} + \max(\Delta N_{\langle BUS \rangle ESSENTIAL})}{N_{\langle BUS \rangle BUSY} + \Delta N_{\langle BUS \rangle BUSY}} \quad (5.13)$$

$\max(\Delta N_{\langle BUS \rangle ESSENTIAL}) = \Delta N_{\langle BUS \rangle BUSY}$, тј. ако су сви слободни циклуси замењени са заузетим циклусима заправо есенцијални. Пошто се теоријски максимум перформанси командне магистрале постиже за $U_{\langle CB \rangle} = 100\%$, што имплицира $\Delta N_{\langle CB \rangle BUSY} = N_{\langle CB \rangle IDLE}$, тј. да се сви слободни циклуси могу заменити заузетим циклусима, горња граница ефикасности командне магистрале постаје:

$$\begin{aligned} \max(E_{\langle CB \rangle}) &= \frac{N_{\langle CB \rangle ESSENTIAL} + \Delta N_{\langle CB \rangle BUSY}}{N_{\langle CB \rangle BUSY} + \Delta N_{\langle CB \rangle BUSY}} \\ &= \frac{N_{\langle CB \rangle ESSENTIAL} + N_{\langle CB \rangle IDLE}}{N_{\langle CB \rangle BUSY} + N_{\langle CB \rangle IDLE}} \\ &= \frac{N_{\langle CB \rangle ESSENTIAL} + N_{\langle CB \rangle IDLE}}{N_{\langle CB \rangle TOTAL}} \\ &= \frac{N_{\langle CB \rangle ESSENTIAL}}{N_{\langle CB \rangle TOTAL}} + \frac{N_{\langle CB \rangle IDLE}}{N_{\langle CB \rangle TOTAL}} \\ &= \frac{N_{\langle CB \rangle ESSENTIAL}}{N_{\langle CB \rangle TOTAL}} + \frac{N_{\langle CB \rangle TOTAL} - N_{\langle CB \rangle BUSY}}{N_{\langle CB \rangle TOTAL}} \\ &= EU_{\langle CB \rangle} + 1 - U_{\langle CB \rangle} \end{aligned}$$

односно:

$$\max(E_{\langle CB \rangle}) = EU_{\langle CB \rangle} + I_{\langle CB \rangle}, \quad I_{\langle CB \rangle} = 1 - U_{\langle CB \rangle} \quad (5.14)$$

Горња граница ефикасности за магистралу података се може извести из (5.3), (5.5), (5.6) и (5.13) узимајући у обзир да је $\max(\Delta N_{\langle DB \rangle ESSENTIAL}) = \Delta N_{\langle DB \rangle ESSENTIAL} \big|_{\Delta N_{\langle CB \rangle ESSENTIAL} = N_{\langle CB \rangle IDLE}} = f_{\langle DB \rangle} / f_{\langle CB \rangle} \cdot N_{\langle CB \rangle IDLE}$:

$$\begin{aligned}
\max(E_{\langle DB \rangle}) &= \frac{N_{\langle DB \rangle ESSENTIAL} + \max(\Delta N_{\langle DB \rangle ESSENTIAL})}{N_{\langle DB \rangle BUSY} + \Delta N_{\langle DB \rangle BUSY}} \\
&= \frac{N_{\langle DB \rangle ESSENTIAL} + \Delta N_{\langle DB \rangle ESSENTIAL} \Big|_{\Delta N_{\langle CB \rangle ESSENTIAL} = N_{\langle CB \rangle IDLE}}}{N_{\langle DB \rangle BUSY} + \Delta N_{\langle DB \rangle BUSY} \Big|_{\Delta N_{\langle CB \rangle BUSY} = N_{\langle CB \rangle IDLE}}} \\
&= \frac{N_{\langle DB \rangle ESSENTIAL} + \Delta N_{\langle DB \rangle ESSENTIAL} \Big|_{\Delta N_{\langle CB \rangle ESSENTIAL} = N_{\langle CB \rangle IDLE}}}{N_{\langle DB \rangle ESSENTIAL} \cdot \frac{N_{\langle DB \rangle BUSY}}{N_{\langle DB \rangle ESSENTIAL}} + \Delta N_{\langle DB \rangle BUSY} \Big|_{\Delta N_{\langle CB \rangle BUSY} = N_{\langle CB \rangle IDLE}}} \\
&= \frac{f_{\langle DB \rangle} / f_{\langle CB \rangle} \cdot N_{\langle CB \rangle ESSENTIAL} + f_{\langle DB \rangle} / f_{\langle CB \rangle} \cdot N_{\langle CB \rangle IDLE}}{f_{\langle DB \rangle} / f_{\langle CB \rangle} \cdot N_{\langle CB \rangle ESSENTIAL} / E_{\langle DB \rangle} + f_{\langle DB \rangle} / f_{\langle CB \rangle} \cdot N_{\langle CB \rangle IDLE}} \\
&= \frac{N_{\langle CB \rangle ESSENTIAL} + N_{\langle CB \rangle IDLE}}{N_{\langle CB \rangle ESSENTIAL} / E_{\langle DB \rangle} + N_{\langle CB \rangle IDLE}} \\
&= \frac{\frac{N_{\langle CB \rangle ESSENTIAL} + N_{\langle CB \rangle IDLE}}{N_{\langle CB \rangle TOTAL}}}{\frac{N_{\langle CB \rangle ESSENTIAL} / E_{\langle DB \rangle} + N_{\langle CB \rangle IDLE}}{N_{\langle CB \rangle TOTAL}}} \\
&= \frac{\frac{N_{\langle CB \rangle ESSENTIAL}}{N_{\langle CB \rangle TOTAL}} + \frac{N_{\langle CB \rangle IDLE}}{N_{\langle CB \rangle TOTAL}}}{\frac{N_{\langle CB \rangle ESSENTIAL}}{N_{\langle CB \rangle TOTAL}} / E_{\langle DB \rangle} + \frac{N_{\langle CB \rangle IDLE}}{N_{\langle CB \rangle TOTAL}}} \\
&= \frac{EU_{\langle CB \rangle} + I_{\langle CB \rangle}}{EU_{\langle CB \rangle} / E_{\langle DB \rangle} + I_{\langle CB \rangle}}
\end{aligned}$$

ОДНОСНО:

$$\max(E_{\langle DB \rangle}) = \frac{\max(E_{\langle CB \rangle})}{U_{\langle DB \rangle} + I_{\langle CB \rangle}} \quad (5.15)$$

Горња граница искоришћења магистрале података се лако изводи на основу дефиниције (5.1) заменом одговарајућих вредности:

$$\begin{aligned}
\max(U_{\langle DB \rangle}) &= \frac{N_{\langle DB \rangle BUSY} + \max(\Delta N_{\langle DB \rangle BUSY})}{N_{\langle DB \rangle TOTAL}} \\
&= \frac{N_{\langle DB \rangle BUSY} + \Delta N_{\langle DB \rangle BUSY} \Big|_{\Delta N_{\langle CB \rangle ESSENTIAL} = N_{\langle CB \rangle IDLE}}}{N_{\langle DB \rangle TOTAL}} \\
&= \frac{N_{\langle DB \rangle BUSY}}{N_{\langle DB \rangle TOTAL}} + \frac{f_{\langle DB \rangle} / f_{\langle CB \rangle} \cdot N_{\langle CB \rangle IDLE}}{N_{\langle DB \rangle TOTAL}} \\
&= \frac{N_{\langle DB \rangle BUSY}}{N_{\langle DB \rangle TOTAL}} + \frac{f_{\langle DB \rangle} / f_{\langle CB \rangle} \cdot N_{\langle CB \rangle IDLE}}{f_{\langle DB \rangle} / f_{\langle CB \rangle} \cdot N_{\langle CB \rangle TOTAL}}
\end{aligned}$$

односно:

$$\max(U_{\langle DB \rangle}) = U_{\langle DB \rangle} + I_{\langle CB \rangle} \quad (5.16)$$

Према (5.6) горња граница перформанси магистрале података је:

$$\max(EU_{\langle DB \rangle}) = \max(EU_{\langle CB \rangle}) = \max(E_{\langle CB \rangle}) \quad (5.17)$$

Дефиниција 5.13: *Маргина перформанси* је разлика између процењеног максимума перформанси и стварних перформанси.

$$\Delta EU_{max} = \max(EU_{\langle BUS \rangle}) - EU_{\langle BUS \rangle} \quad (5.18)$$

У овом случају маргина перформанси је:

$$\Delta EU_{max} = I_{\langle CB \rangle} \quad (5.19)$$

Примећује се да ΔEU_{max} зависи само од броја слободних циклуса. То је последица идеализације да се, у сврху процене горње границе перформанси DRAM меморије, сви слободни циклуси на командној магистрали могу заменити есенцијалним циклусима. У стварности, то није могуће због постојања функционалних циклуса.

5.2.2 Процена максимума перформанси интерполацијом

Пошто се горња граница перформанси DRAM меморија процењује а posteriori, број есенцијалних циклуса се може интерполирати, уместо да се претпоставља његова максимална вредност. Овај приступ даје реалистичнију процену максимума перформанси. Да би се олакшало рачунање перформанси DRAM меморије када се интерполира максималан број есенцијалних циклуса, корисно је извести израз за $\max(E_{(CB)})$ из (5.13) користећи генерализацију $\max(\Delta N_{(CB) ESSENTIAL}) = p_{(CB)} \cdot N_{(CB) IDLE}$, где је $p_{(CB)}$ проценат есенцијалних циклуса који замењују слободне циклусе:

$$\begin{aligned} \max(E_{(CB)}) &= \frac{N_{(CB) ESSENTIAL} + p_{(CB)} \cdot N_{(CB) IDLE}}{N_{(CB) BUSY} + \Delta N_{(CB) BUSY}} \\ &= \frac{N_{(CB) ESSENTIAL} + p_{(CB)} \cdot N_{(CB) IDLE}}{N_{(CB) BUSY} + N_{(CB) IDLE}} \\ &= \frac{N_{(CB) ESSENTIAL} + p_{(CB)} \cdot N_{(CB) IDLE}}{N_{(CB) TOTAL}} \\ &= EU_{(CB)} + p_{(CB)} \cdot I_{(CB)} \end{aligned}$$

односно:

$$\max(EU_{(CB)}) = EU_{(CB)} + p_{(CB)} \cdot I_{(CB)}, \quad p_{(CB)} = \frac{\Delta N_{(CB) ESSENTIAL}}{\Delta N_{(CB) IDLE}} \quad (5.20)$$

Разлика између процењених и стварних перформанси према (5.18) постаје:

$$\Delta EU_{max} = p_{(CB)} \cdot I_{(CB)} \quad (5.21)$$

Формуле (5.11)-(5.13) and (5.15)-(5.17) и даље важе.

Интерполације се може урадити на разне начине, али је најинтуитивнији приступ да се претпостави да се однос броја есенцијалних циклуса и укупног броја

циклуса у замењеним слободним циклусима не мења у односу на остатак посматране секвенце:

$$\Delta N_{\langle CB \rangle ESSENTIAL} : \Delta N_{\langle CB \rangle BUSY} = N_{\langle CB \rangle ESSENTIAL} : N_{\langle CB \rangle BUSY} \quad (5.22)$$

Из овога услова следи $p_{\langle CB \rangle} = E_{\langle CB \rangle}$, а из (5.21):

$$\max(E_{\langle CB \rangle}) = E_{\langle CB \rangle} \quad (5.23)$$

Пошто је према (5.23) максимална ефикасност у овом случају непороменљива, добијена вредност ΔEU_{max} се може интерпретирати као перформансе DRAM меморије које нису искоришћене искључиво због мањег искоришћења командне магистрале:

$$\Delta EU_{max}(U) = \Delta EU_{max}|_{E_{\langle CB \rangle} = const} = E_{\langle CB \rangle} \cdot I_{\langle CB \rangle} \quad (5.24)$$

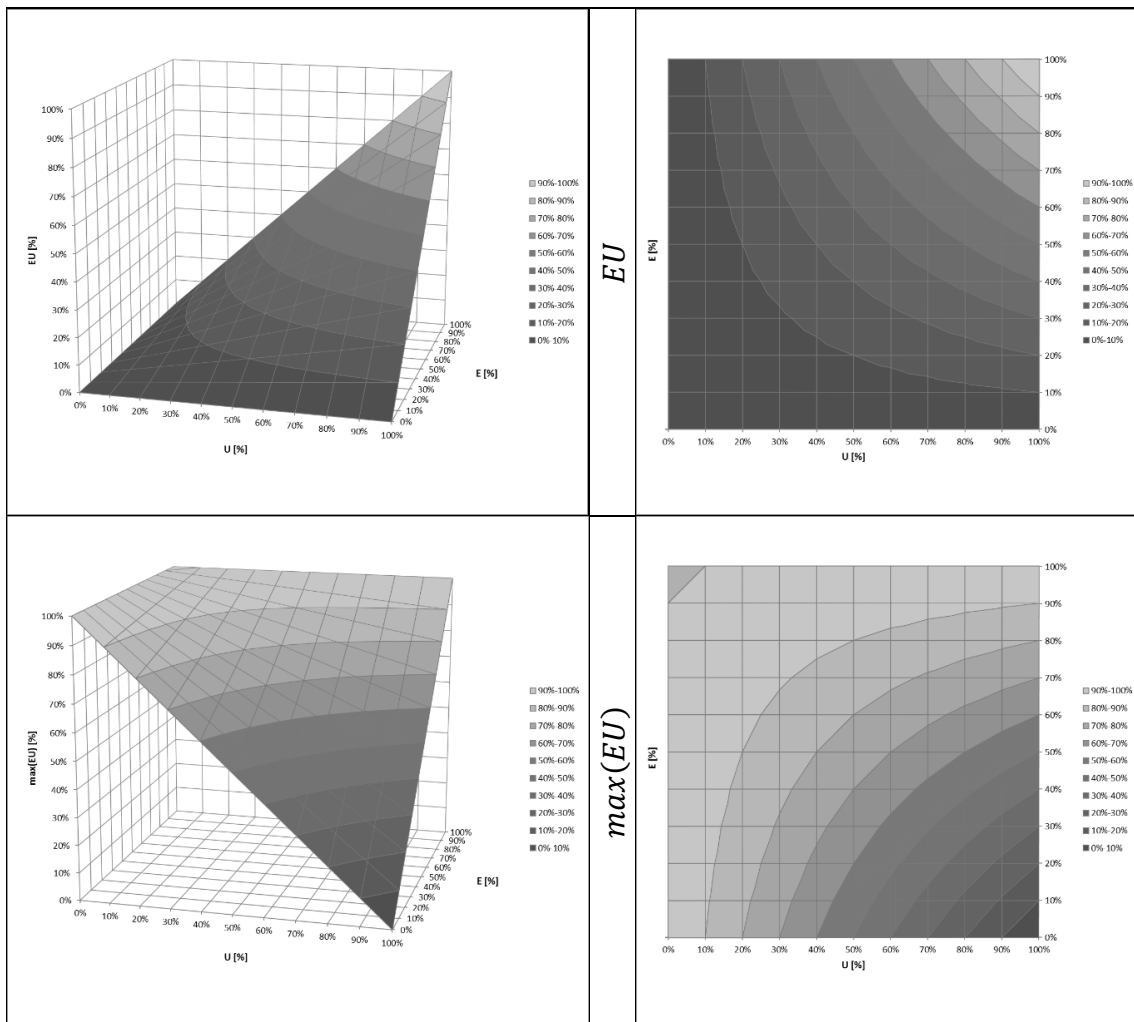
$\Delta EU_{max}(U)$ представља део перформанси које су изгубљене услед појаве слободних циклуса на магистрали података. Узрок томе може бити „изгладњивање“, односно нижа стопа захтева који стижу до меморијског контролера у односу на оно што меморија може да опслужи, или неоптималан тајминг приликом слања команди од стране меморијског контролера, чиме се на магистралу уведе „балончићи“ слободних циклуса. Смањивање $\Delta EU_{max}(U)$ имплицира смањивање $I_{\langle CB \rangle}$, што је могуће јер не захтева промену редоследа DRAM команди.

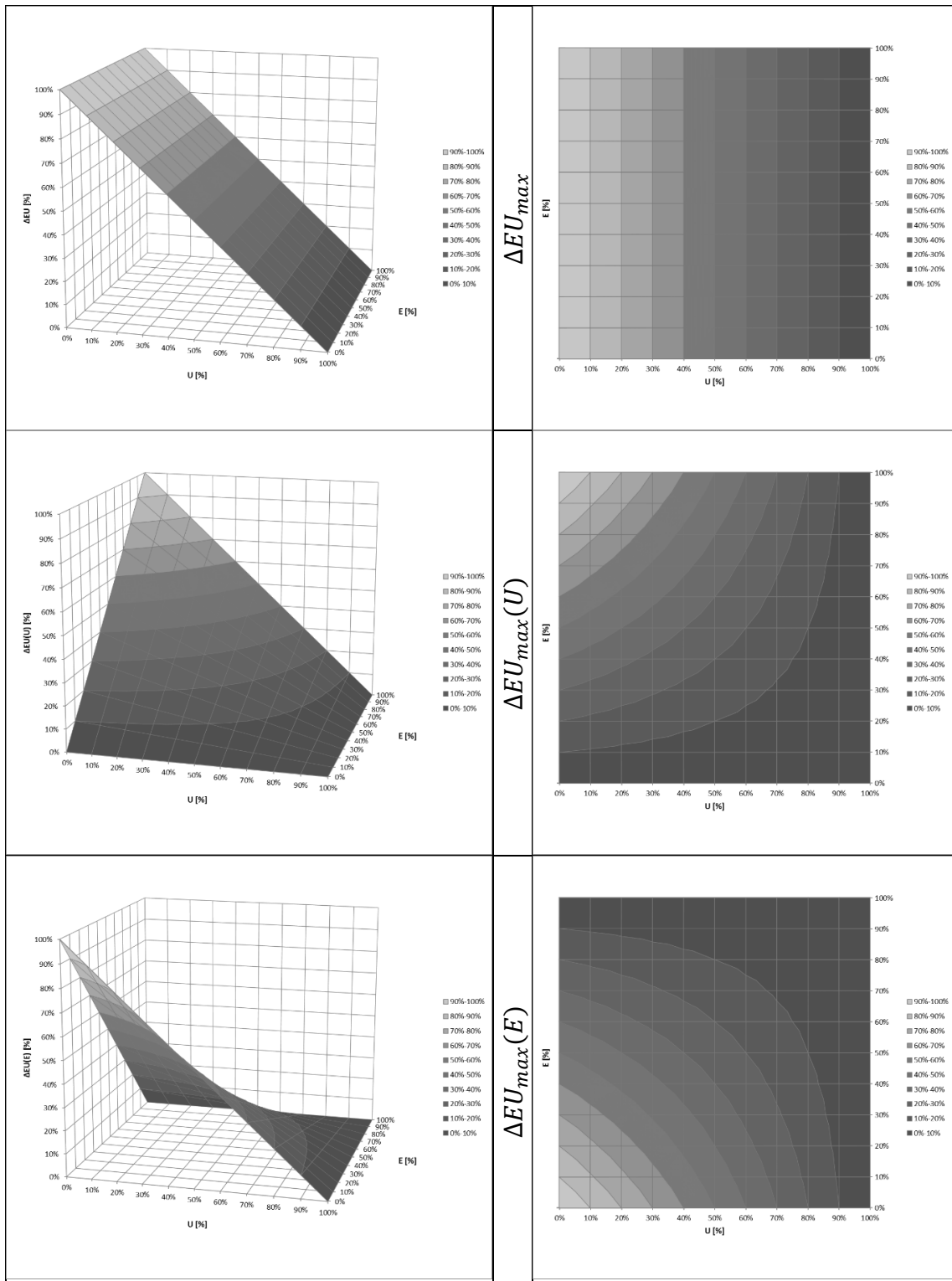
Разлика између (5.19) и (5.24) се може интерпретирати као део перформанси DRAM меморије које нису искоришћене због ниже ефикасности командне магистрале. Пошто се максимум перформанси постиже за $U_{\langle CB \rangle} = 100\%$, то имплицира $E_{\langle CB \rangle} + F_{\langle CB \rangle} = 1$, где је $F_{\langle CB \rangle} = N_{\langle CB \rangle FUNCTIONAL} / N_{\langle CB \rangle BUSY}$, што даје:

$$\Delta EU_{max}(E) = I_{\langle CB \rangle} - \Delta EU_{max}(U) = F_{\langle CB \rangle} \cdot I_{\langle CB \rangle} \quad (5.25)$$

$\Delta EU_{max}(E)$ представља део перформанси које су изгубљене због неоптималног секвенцирања команди од стране меморијског контролера. Пошто се редослед DRAM команди у постојећој секвенци не може мењати, $\Delta EU_{max}(E)$ је непроменљиво.

На Слици 5.1 су дати графикони који илуструју кључне метрике перформанси DRAM меморије.





Слика 5.1 – Графикони метрика перформанси DRAM меморија

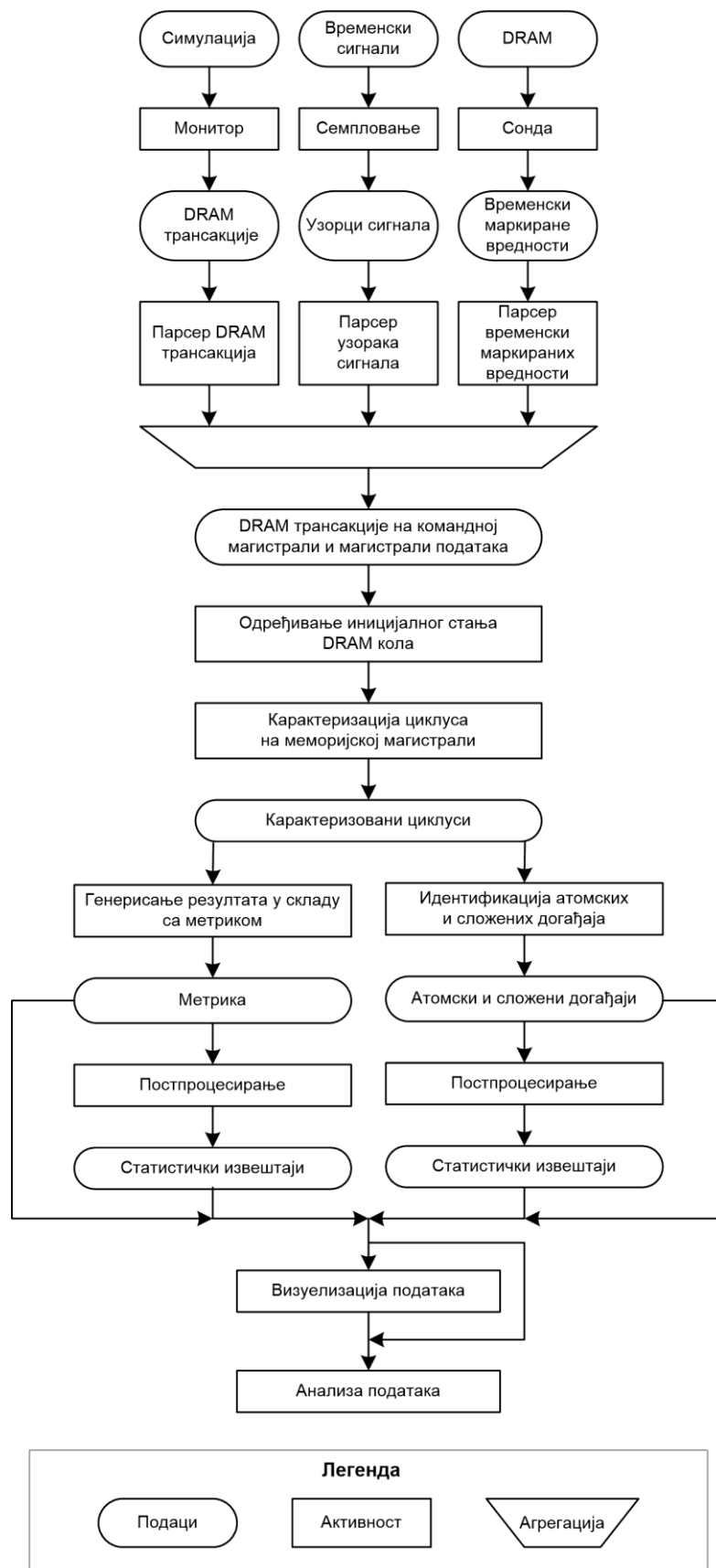
6 Методологија

Методологија за мерење и анализу перформанси DRAM меморија специфицира методе, процесе и алате који се користе за прикупљање података и њихову обраду са циљем проучавања рада DRAM меморије и генерисаног DRAM саобраћаја у рачунарском систему. Методологија предложена у оквиру ове дисертације има за циљ да реши најважније проблеме у овој области у које спадају проблем тачности, проблем интерпретације измерених вредности и проблем компарације резултата (видети поглавље 1.5.1). Предложена методологија уједно нуди низ других предности у односу на постојеће методологије, као што су брзина рада, једноставност употребе, применљивост на различите рачунарске системе, комплетност покривања свих релевантних трансакција, опсервабилност улазних параметара и др.

6.1 Ток рада

Ток рада (енг. workflow) који се користи у оквиру предложене методологије је дат на Слици 6.1. Кључне фазе у оквиру методологије су:

1. Прикупљање/снимање DRAM трансакција у одговарајућем формату
2. Парсирање прикупљених података о меморијским трансакцијама са циљем формирања јединствене репрезентације DRAM трансакција на командној магистралу и на магистралу података без обзира на порекло и формат у коме су подаци оригинално прикупљени



Слика 6.1 – Ток рада према предложеној методологији

3. Одређивање почетног стања меморије како би се могли формирати иницијални прелази у DCT FSM
4. Карактеризација циклуса применом ACDC модела
5. Генерисање резултата
 - i. Метрика: искоришћење, ефикасност, есенцијално искоришћење и маргине перформанси
 - ii. Догађаји: атомски (заузети периоди, слободни периоди, итд.) и сложени (погоци, промашаји и конфликти приликом приступа банки, итд.)
6. Постпроцесирање генерисаних резултата и креирање статистичких извештаја
7. Визуелизација добијених података
8. Анализа података на основу резултата добијених у претходним фазама

У наредним поглављима ће бити детаљно разматрана релевантна питања у односу на сваку од фаза.

6.2 Снимање DRAM трансакција

Улазне податке, на основу којих се врше сва мерења и анализе, чини низ DRAM трансакција на командној магистралу и магистралу података. За трансакције на командној магистралу се бележе следећи подаци:

- време трансакције
- трајање трансакције издавања команде
- DRAM команда
- меморијски канал
- ранк
- банка
- врста

Row	=> <i>врста</i> ,	
Column	=> <i>колона</i> }	
time => { Data	=> <i>податак</i> }	} Један податак у једном циклусу

DRAM сонда уписује податке детектоване на меморијској магистрали у интерну меморију из које се подаци касније могу прочитати и снимити у виду фајла. Сваки запис у фајлу садржи циклус у коме је стање детектовано и одговарајуће вредности на магистрали. По структури и садржају запис је исти као код семплованих временских записа.

6.3 Парсирање DRAM трансакција

Парсирањем се од података снимљених из различитих извора може генерисати јединствена интерна репрезентација DRAM трансакција на командној магистрали и магистрали података. Ова репрезентација има идентичну структуру као и структура коју генерише монитор DRAM трансакција.

Као пример дат је један једноставан парсер у језику Perl који процесира трансакције на командној магистрали снимљене помоћу DRAM сонде. Парсер генерише хеш листу чији елементи су времена трансакција, а сваки елемент показује на одговарајућу структуру са прочитаним параметрима:

```
my %REGEXP;
$REGEXP{Time}      = qr/(\d+)\.\d+\s+ps/;
$REGEXP{Command}   = qr/(\w+)/;
$REGEXP{Channel}   = qr/(ChA|ChB)/;
$REGEXP{Rank}      = qr/RANK=\s*([0-9A-Fa-fxX]+)/;
$REGEXP{Bank}      = qr/BANK=\s*([0-9A-Fa-fxX]+)/;
$REGEXP{Row}       = qr/ROW=\s*([0-9A-Fa-fxX]+)/;
$REGEXP{Column}    = qr/COL=\s*([0-9A-Fa-fxX]+)/;
$REGEXP{Transaction} = qr/$REGEXP{Time}\s+$REGEXP{Command}\s+$REGEXP{Channel}
\s+$REGEXP{Rank}\s+$REGEXP{Bank}\s+$REGEXP{Row}\s+$REGEXP{Column}/;
my $Operations_by_Time = {};
sub Parse {
    my $Time,$Cmd,$Ch,$Rank,$Bank,$Row,$Col;
    my $FILE=open(TRANSACTIONS_LOG, "< $Transactions_Log");
```

```
while (<$FILE>) {
    ($Time,$Cmd,$Ch,$Rank,$Bank,$Row,$Col) = $_ =~ $REGEXP{Transaction};
    push @{$Operations_by_Time->{$Time}}, { Command=>$Cmd,
                                            Channel=>$Ch,
                                            Rank   =>$Rank,
                                            Bank   =>$Bank,
                                            Row    =>$Row,
                                            Column =>$Col }; }

return $Operations_by_Time;
}
```

6.4 Одређивање почетног стања

Анализа DRAM саобраћаја се може радити док се трансакције генеришу (енг. runtime) или након што се трансакције најпре сниме на уређај за складиштење података (енг. postprocessing). У оба случаја је потребно најпре одредити иницијално стање меморије, односно свих DRAM кола. То је неопходно како би се могли формирати парови команди који чине прелаз у DCT FSM. Стање меморије које треба пратити је било раније дефинисано, а састоји се од информација за сваку банку и ранк у систему о томе која је последња команда која их је адресирала.

Уколико иницијално стање меморије није познато или је некомплетно, као нпр. на почетку симулације или уколико трансакције нису праћене све време од момента када је стање било познато, могуће је попунити недостајуће информације тако што ће се пратити DRAM трансакције све до тренутка док се не адресирају свака банка и ранк у систему, при чему се успут бележе недостајуће информације о последњој команди која их је адресирала. На тај начин ће се у једном тренутку добити комплетно стање меморије. Након тога је могуће наставити са карактеризацијом циклуса. Карактеризацију је могуће радити и пре комплетирања стања меморије, када год је могуће формирати прелаз у DCT FSM. Међутим, у том случају карактеризација не мора бити потпуно тачна, јер прелази које није могуће формирати због некомплетног стања меморије могу променити карактер циклуса у случају када постоји преклапање циклуса из два прелаза.

6.5 Карактеризација меморијских циклуса

Карактеризација циклуса се ради помоћу ACDC модела. ACDC модел може бити имплементиран на било којој платформи и у било ком програмском језику. Уколико се трансакције прикупљају у току симулације помоћу DRAM монитора, онда је могуће имплементирати модел у верификационом језику као што је *e* или SystemVerilog. Ако се прикупљене трансакције процесирају накнадно, онда модел може бити имплементиран и у вишем програмском језику опште намене као што су C или Perl. Коначно, ако је потребно анализирати само мање секвенце, онда се може користити и Excel. На Слици 6.2 је дат један такав пример.

CMD	CMD	Rank	Bank	Row	Col	Addr	CB active	CB overhead	CB idle	CB busy	CB essential	CB functional	CB active essential	CB overhead essential	CB active functional	CB overhead functional	DB active	DB overhead	DB idle	DB busy		
21958	Read	3	2		02d8	20abad80	1	0	0	1	1	0	1	0	0	0	0	0	0	1	0	
21959							0	1	0	1	1	0	0	0	0	0	0	0	0	0	1	0
21960	Read	3	1		02ac	20ab6ac0	1	0	0	1	1	0	1	0	0	0	0	0	0	0	1	0
21961							0	1	0	1	1	0	0	0	0	0	0	1	0	0	0	1
21962	Read	3	1		02a4	20ab6a40	1	0	0	1	1	0	1	0	0	0	0	1	0	0	0	1
21963							0	1	0	1	1	0	0	0	0	0	0	1	0	0	0	1
21964							0	1	0	1	0	1	0	1	0	1	0	1	0	0	0	1
21965							0	1	0	1	0	1	0	1	0	1	0	1	0	0	0	1
21966							0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	1
21967							0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	1
21968	Write	2	2		02cc	2002acc0	1	0	0	1	1	0	1	0	0	0	0	0	0	0	1	0
21969							0	1	0	1	1	0	0	0	0	0	0	0	0	0	1	0
21970	Write	2	2		02d8	2002ad80	1	0	0	1	1	0	1	0	0	0	0	1	0	0	0	1
21971							0	1	0	1	1	0	0	0	0	0	0	1	0	0	0	1
21972	Write	2	2		02dc	2002adc0	1	0	0	1	1	0	1	0	0	0	0	1	0	0	0	1
21973							0	1	0	1	1	0	0	0	0	0	0	1	0	0	0	1
21974	Write	2	2		02c0	2002ac00	1	0	0	1	1	0	1	0	0	0	0	1	0	0	0	1
21975							0	1	0	1	1	0	0	0	0	0	0	1	0	0	0	1
21976	Read	3	1		0294	20ab6940	1	0	0	1	1	0	1	0	0	0	0	1	0	0	0	1
21977							0	1	0	1	1	0	0	0	0	0	0	1	0	0	0	1
21978	Read	3	2		0330	20abb300	1	0	0	1	1	0	1	0	0	0	0	0	1	0	0	1
21979							0	1	0	1	1	0	0	0	0	0	0	1	0	0	0	1
21980							0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	1
21981							0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	1
21982	Read	3	2		0324	20abb240	1	0	0	1	1	0	1	0	0	0	0	1	0	0	0	1
21983							0	1	0	1	1	0	0	0	0	0	0	0	0	0	1	0
21984	Read	3	2		0320	20abb200	1	0	0	1	1	0	1	0	0	0	0	0	0	0	1	0
21985							0	1	0	1	1	0	0	0	0	0	0	1	0	0	0	1
21986							0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	1
21987	Prech	3	1			0000	1	0	0	1	0	1	0	0	1	0	0	1	0	0	0	1
21988							0	1	0	1	0	1	0	0	0	1	0	1	0	0	0	1
21989							0	1	0	1	0	1	0	0	0	0	1	0	1	0	0	1
21990	Write	2	2		02c0	2002ac00	1	0	0	1	1	0	1	0	0	0	0	0	0	0	1	0
21991	Activ	3	1	0415			1	0	0	1	1	0	0	0	0	0	0	0	0	0	1	0
21992	Write	2	2		02c4	2002ac40	1	0	0	1	1	0	1	0	0	0	0	1	0	0	0	1
21993							0	1	0	1	1	0	0	0	0	0	0	1	0	0	0	1
21994	Write	2	2		02d0	2002ad00	1	0	0	1	1	0	1	0	0	0	0	1	0	0	0	1
21995							0	1	0	1	1	0	0	0	0	0	0	1	0	0	0	1
21996	Read	3	1		029c	20ab69c0	1	0	0	1	1	0	1	0	0	0	0	1	0	0	0	1
21997							0	1	0	1	1	0	0	0	0	0	0	1	0	0	0	1
21998	Read	3	1		035c	20ab75c0	1	0	0	1	1	0	1	0	0	0	0	0	1	0	0	1
21999							0	1	0	1	1	0	0	0	0	0	0	1	0	0	0	1
22000							0	1	0	1	0	1	0	1	0	1	0	1	0	0	0	1
22001							0	1	0	1	0	1	0	1	0	1	0	1	0	0	0	1
22002							0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	1
22003							0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	1

Слика 6.2 – Имплементација ACDC модела у Excel-у

6.6 Генерисање резултата

Могуће је генерисати два типа резултата. Један тип резултата, назван догађаји, се генерише само на основу активности на меморијској магистралу. За њих није неопходна карактеризација, мада она може олакшати разумевање догађаја. Догађаји се могу класификовати у две групе: атомске и сложене. Атомски догађаји су догађаји који се могу идентификовати само на основу сигнала који су опсервабилни на меморијској магистралу. Типични атомски догађаји су:

- Периоди када је активна одређена команда на командној магистралу
- Периоди када постоји непрекидан низ Read команди (енг. read bursts) или Write команди (енг. write bursts)
- Периоди када се преносе подаци на магистралу података (енг. data bursts)
- Прелазни између одређених команди (нпр. Write→Read прелаз је интересантан јер изазива велики број режијских циклуса)
- Периоди када се приступа одређеном каналу, ранку, банки или врсти
- Периоди када је магистрала неактивна или активна¹¹

Сложени догађаји су догађаји који се могу извести на основу опсервабилних сигнала са меморијске магистрале и тренутног стања меморије. Типични сложени догађаји су:

- Погодак приликом приступа (приступа се податку у врсти која је већ активирана; нема потребе за додатним активностима)
- Промашај приликом приступа (приступа се податку у банки у којој нема активираних врста; мора се урадити Activate врсте)
- Конфликт приликом приступа (приступа се податку у различитој врсти од оне која је активирана; мора се прво урадити Precharge отворене врсте, па Activate нове врсте)

¹¹ Само се карактеризацијом може одредити да ли су циклуси на магистралу слободни или заузети, али ако је период неактивности дуг, онда се може претпоставити да је магистрала слободна.

Други тип резултата који се генерише се базира на карактеризацији. На основу карактеризације циклуса се могу дефинисати следећи догађаји:

- Периоди када је магистрала слободна
- Периоди када је магистрала заузета
- Периоди када се јављају режијски циклуси

Карактеризација омогућава да се генеришу резултати у складу са новоуведеном метриком, са тачношћу на нивоу циклуса. У те резултате спадају:

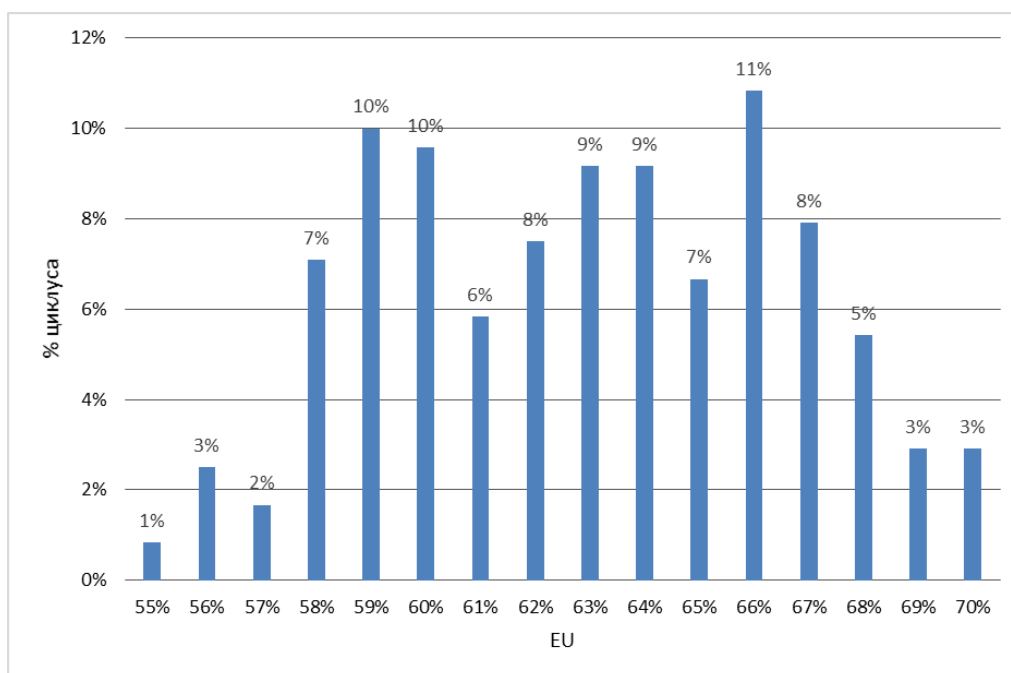
- Искоришћење
- Ефикасност
- Есенцијално искоришћење
- Маргина перформанси
- Маргина перформанси узрокована слободним циклусима
- Маргина перформанси узрокована неоптималним секвенцирањем

Метрика знатно олакшава анализу, јер омогућава разумевање дешавања на меморијској магистрали са произвољним нивоом детаљности.

6.7 Постпроцесирање

Процесирањем генерисаних резултата могу се добити резултати погодни за визуелизацију и анализу. У такве резултате спадају секвенце догађаја, статистички извештаји о укупном броју или проценту вредности одређеног типа, о статистичкој вези (корелацији) догађаја, дистрибуцији вредности итд. Примери резултата који се могу добити постпроцесирањем су:

- Секвенца слободних и заузетих периода на магистрали
- Секвенца погодака, промашаја и конфликта приликом приступа
- Укупан број погодака, промашаја и конфликта за одређени канал, ранк, банку или врсту
- Укупан број режијских циклуса за дати канал, ранк, банку или врсту
- Корелација метрике и броја погодака, промашаја и конфликта



Слика 6.3 – Дистрибуција вредности есенцијалног искоришћења

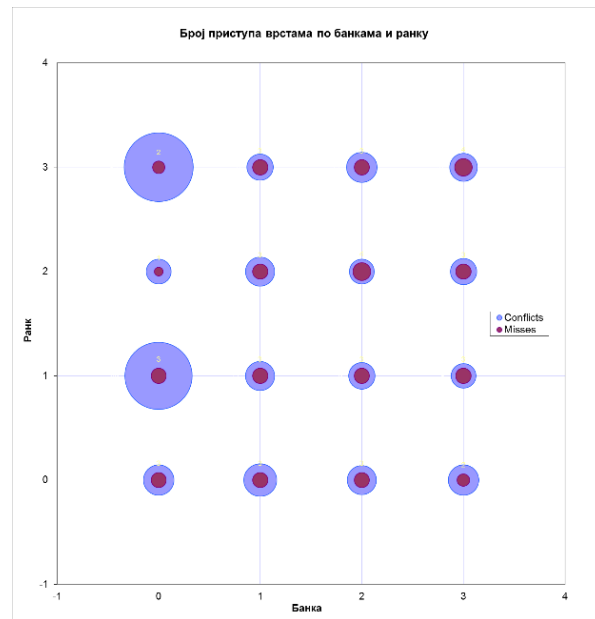
- Дистрибуција вредности параметара метрике: максимум, минимум, средња вредност итд.

На Слици 6.3 је дат пример дистрибуције вредности есенцијалног искоришћења за период од 240 циклуса са прозором од 1 циклуса. На основу ове статистике се може закључити да се ниво перформанси DRAM меморије у 90% случајева налази у опсегу $63\% \pm 5\%$.

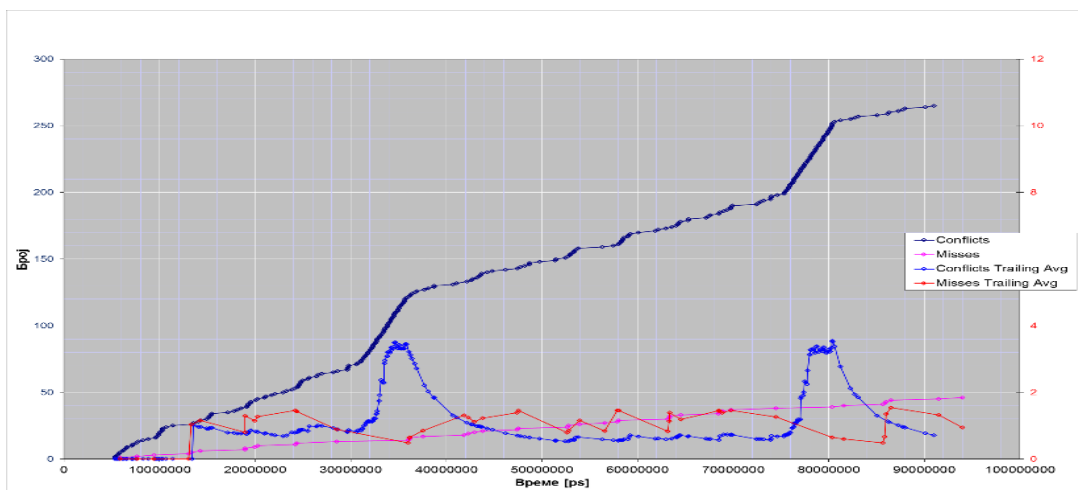
6.8 Визуелизација

Резултати добијени постпроцесирањем се некада лакше, а некада и једино могу разумети након што буду визуелизовани на одговарајући начин. Не постоји „најбољи“ начин за визуелизацију који је применљив на све типове података. Најбољи начин је онај који омогућава да се уоче кључне релације које могу помоћи у анализи.

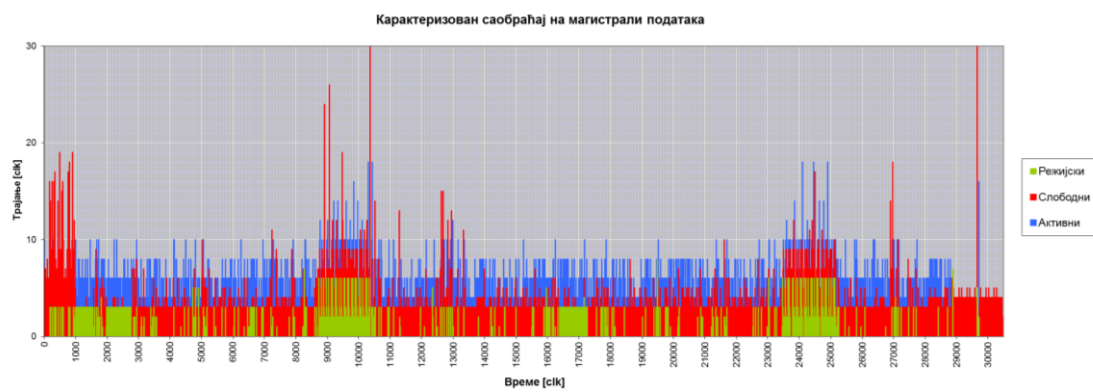
На Слици 6.4 је дат пример три начина визуелизације. У случају на Слици 6.4.а је представљена статистика броја конфликта и промашаја визуелно мапирана



а) Дистрибуција броја конфликта и промашаја по ранку и банки



б) Конфликти и промашаји по ранку и банкама у времену



Карактеризован саобраћај на магистрали података

Слика 6.4 – Визуелизација постпроцесираних података

в)

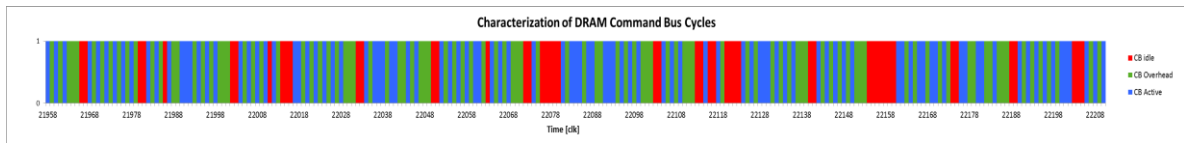
по ранку и банки. Величина кругова је пропорционална броју конфликта, односно промашаја. Овај приказ је добар јер одмах истиче кључну информацију: који ранк и банка имају више конфликта него други. На основу тога се може анализирати зашто долази до тога и утврдити да ли је, рецимо, у питању исти клијент који има неповољну адресну секвенцу или можда меморијски контролер не ради оптимално. На Слици 6.4.б су представљени конфликти и промашаји у времену. Ради лакше анализе уведена је помоћна метрика која показује динамику раста њиховог броја у времену. Овај приказ је добар јер јасно показује да у два периода долази до великог повећања броја конфликта.

На Слици 6.4.в су представљени карактеризовани циклуси на магистрали података. Сваки тип циклуса је приказан у различитој боји. Овај приказ је добар јер визуелно истиче периоде у којима расте број слободних и режијских циклуса.

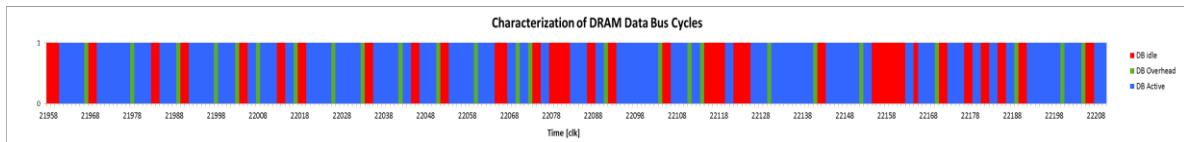
6.9 Анализа

Анализа се ради на основу постпроцесираних и, тамо где за тим има потребе, претходно визуелизованих резултата. На Слици 6.5 су приказане информације које су типично доступне у анализи. На Слици 6.5.а и Слици 6.5.б су визуелизовани циклуси на контролној магистрали и магистрали података који су карактеризовани помоћу ACDC модела. Ови резултати су постпроцесирани како би се креирале секвенце параметара метрике, које су визуелизоване на Слици 6.5.в (временски период, тј. „прозор“ који је коришћен приликом генерисања метрике је 100 циклуса). На слици су подебљани параметри есенцијалног искоришћења (зелено) и маргине есенцијалног искоришћења (љубичасто).

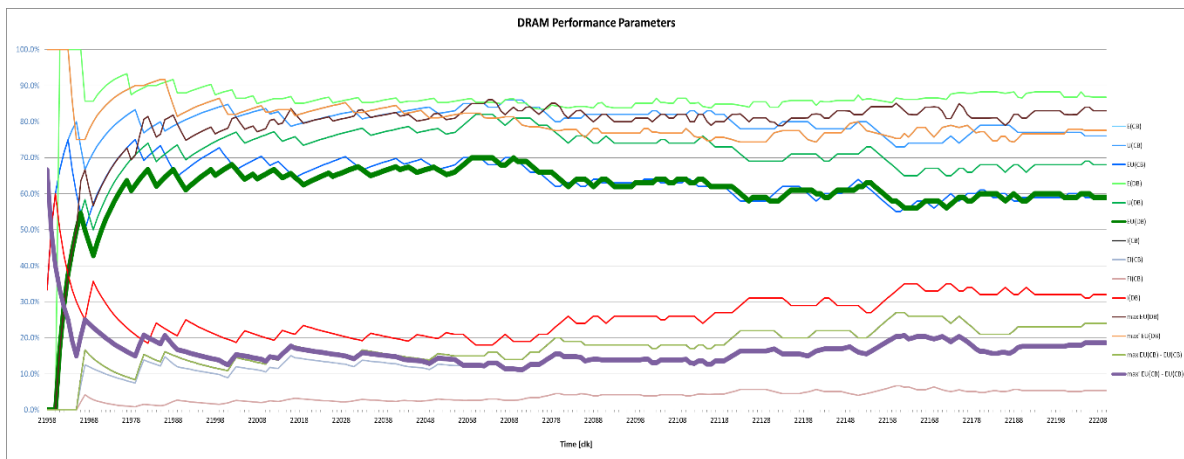
Постпроцесирањем се могу генерисати секвенце за временске периоде различите дужине трајања. У контексту метрике, величина „прозора“ је једнака укупном броју посматраних циклуса $N_{(BUS)TOTAL}$. Мања величина прозора даје већу осетљивост на промене и обрнуто. То се може видети на Слици 6.5.г, где је приказана вредност маргине есенцијалног искоришћења за величине „прозора“ од 5 до 100 циклуса са повећањем од по 5 циклуса (вредност расте по z оси). У пракси, боље је да се величина прозора повећава логаритамски уместо линеарно.



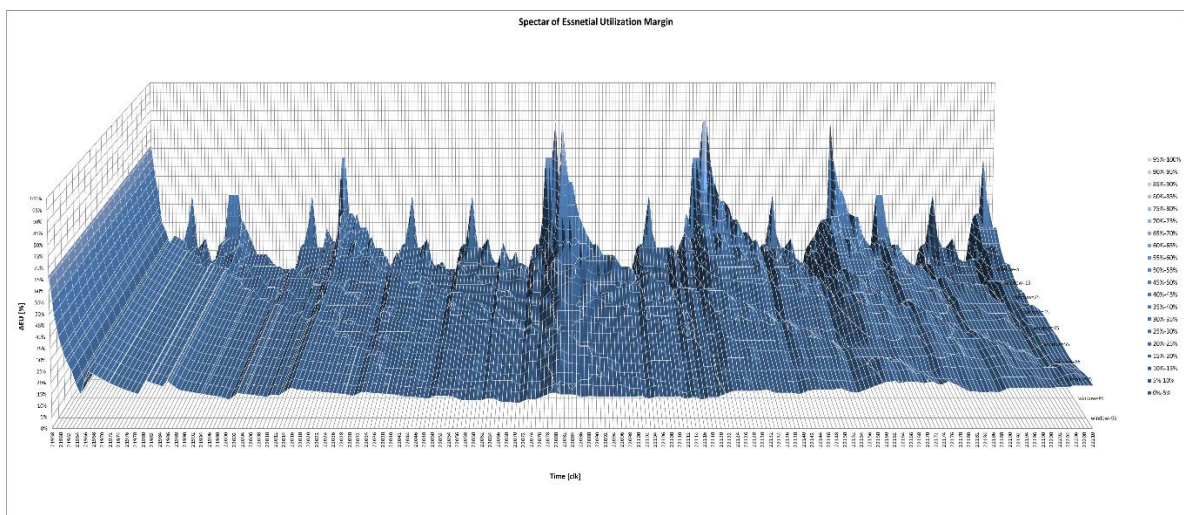
а) Карактеризовани циклуси на командној магистрали



б) Карактеризовани циклуси на магистрали података

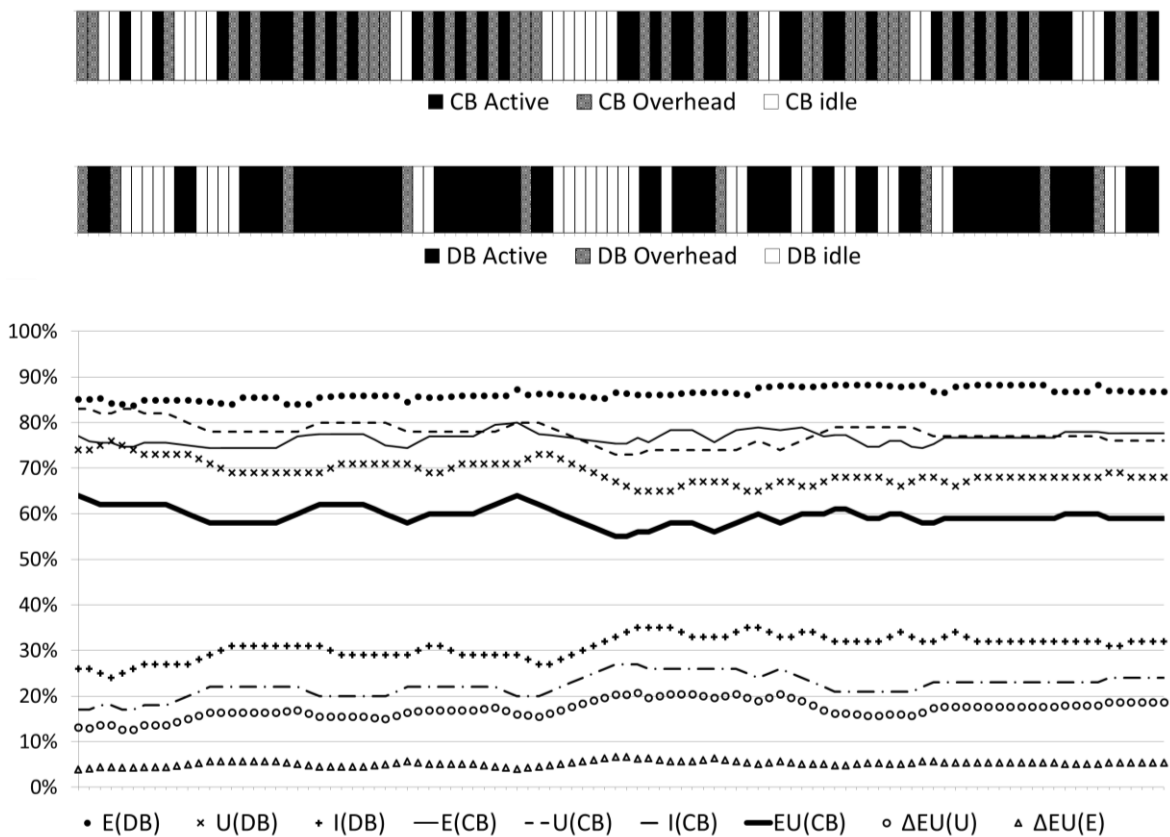


в) Метрика (прозор од 100 циклуса)



г) Спектар маргине есенцијалног искоришћења

Слика 6.5 – Спектрална анализа



Слика 6.6 – Анализа узрока

Спектар вредности маргине есенцијалног искоришћења ΔEU_{max} , као и њене две компоненте $\Delta EU_{max}(U)$ и $\Delta EU_{max}(E)$, за различите величине прозора, омогућавају да се анализира свака појава на меморијској магистрали. Уколико су перформансе у неком периоду ниже, довољно је у спектру наћи нивоу на ком број слободних и режијских циклуса расте и онда анализирати компоненте маргине есенцијалног искоришћења у тим периодима да се утврди узрок.

На Слици 6.6 је дат пример увеличане ситуације са Сlike 6.5.в где су перформансе најниже. Пошто је сваки циклус тачно карактеризован, то је и сва метрика, која се на основу њих рачуна, такође тачна на нивоу циклуса. У случају са слике се види да до пада перформанси долази претежно због $\Delta EU_{max}(U)$. Узрок томе је или повремено изгладњивање због недостатка захтева, што је мање вероватно с обзиром да се ради о губицима који се јављају са учестаношћу од

неколико циклуса, или услед неоптималног рада меморијског контролера који не успева да оптимизује слање команди тако да икористи све доступне циклусе, што је у овом случају вероватније, јер се слање команди ради на нивоу циклуса.

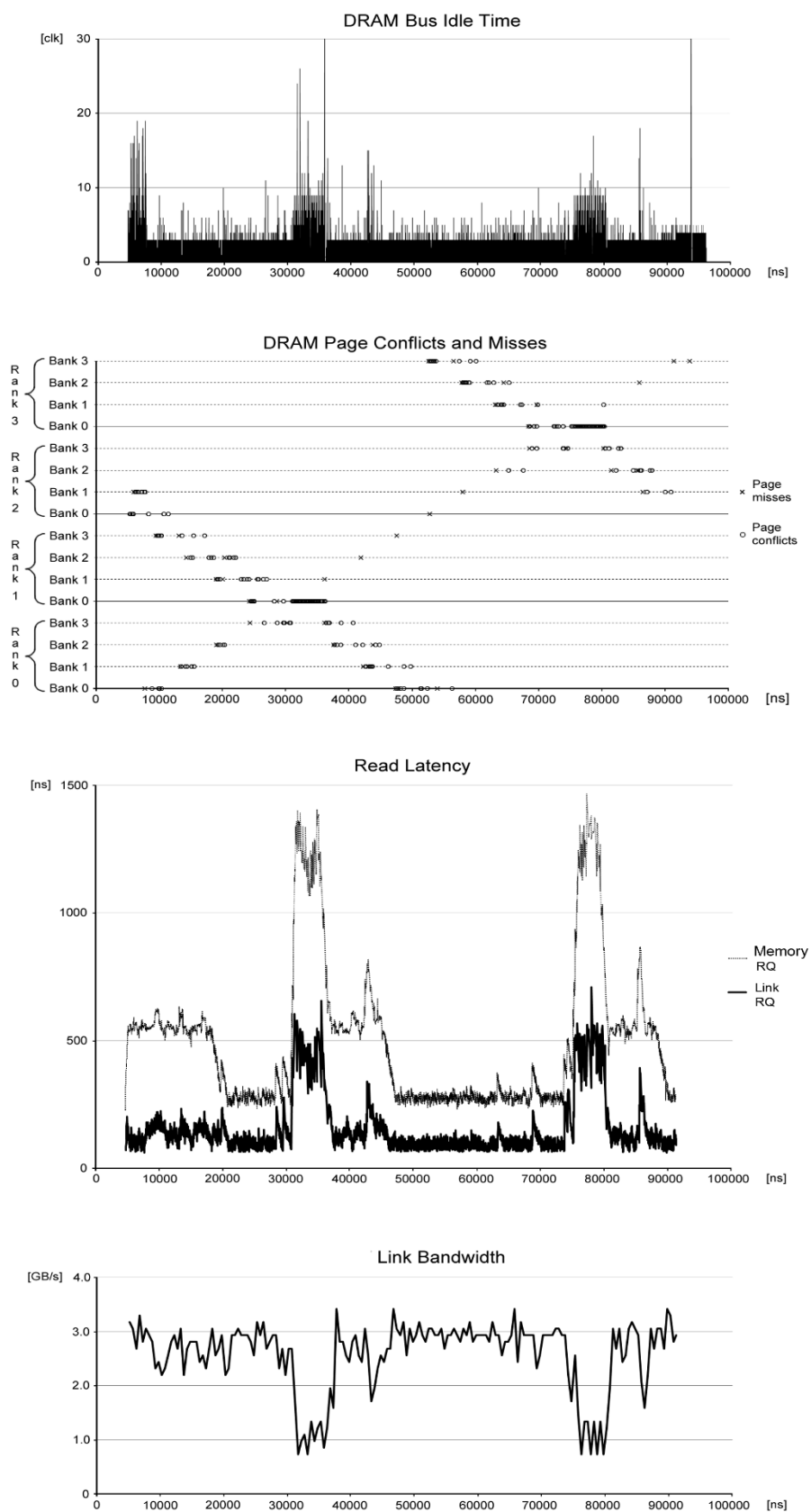
7 Примена

Предложена методологија има примену у великом броју области. Овде ће бити наведене и дискутоване све важније области примене, као и неке нове области где анализа перформанси DRAM меморија није до сада могла да се употреби. Свака од области примене има своје специфичности. Предност ове методологије је што омогућава да применом истог процеса рада може да одговори захтевима употребе у свим овим областима.

7.1 Системска анализа

У системској анализи, односно анализи рада целог рачунарског система, циљ је установити да ли је DRAM меморија уско грло система. Уколико је то тачно, онда се перформансе система могу увећати повећавањем перформанси DRAM меморије. DRAM меморија је уско грло система ако су бафери меморијског контролера који прихватају нове захтеве пуни. Ако је измерена ефикасност меморије на задовољавајућем нивоу, а маргином перформанси доминира $\Delta EU_{max}(U)$, перформансе система се могу поправити повећањем искоришћења на рачун боље оптимизације код слања команди од стране меморијског контролера.

Уколико измерена ефикасност меморије није задовољавајућа, мора се урадити анализа алгоритма арбитрације меморијских захтева и алгоритма за слање DRAM команди унутар меморијског контролера.



Слика 7.1 – Системска анализа перформанси: Корелација догађаја у систему

7.1.1 Корелација догађаја у систему

Корелација догађаја у систему помаже утврђивању узрочно-последичних веза које постоје између њих, а тиме и анализи рада система. Корелацију догађаја у систему са тачношћу на нивоу циклуса омогућава тачна карактеризација меморијских циклуса.

На Слици 7.1 је представљен случај где су дужи периоди слободних циклуса у корелацији са бројем конфликта и промашаја које изазива специфична адресна секвенца коју генерише процесор. То изазива велико кашњење меморијских захтева и смањује проток података на интерном линку, што доводи до пада перформанси система.

7.1.2 Компаративна анализа

Компаративна анализа је поређење перформанси са различитих платформи, за различита радна оптерећења или у различитим периодима времена. Ова врста системске анализе није раније била могућа јер није постојао начин да се резултати пореде због проблема интерпретације измерених вредности.

Применом предложене методологије, решава се проблем флукуације максимума перформанси у зависности од радног оптерећења и извршне платформе, односно решава се проблем интерпретације. Пошто тачна карактеризација обезбеђује и тачно мерење параметара перформанси, а добијени резултат се увек може интерпретирати, поређење је могуће без обзира на то да ли су резултати генерисани на различитим платформама, помоћу различитих радних оптерећења или у различитим периодима времена.

7.2 Архитектурална анализа меморијског контролера

Анализа рада меморијског контролера има за циљ да успостави корелацију између конфигурације контролера и перформанси DRAM меморије за задати скуп радних оптерећења. Као метрика за евалуацију се притом обично користи искоришћење магистрале података. Међутим, коришћење нове метрике може значајно да унапреди овај процес.

Посматрајући промене параметара применом нове метрике много је лакше изоловати прелазе који доводе до појаве режијских и слободних циклуса. У том смислу, $\Delta EU_{max}(U)$ и $\Delta EU_{max}(E)$ могу да пруже информације о томе где треба да буде нагласак у оптимизацији. Код ове врсте анализе потребна је већа осетљивост, тј. мањи временски прозор. Тачност на нивоу циклуса омогућава и да се врши корелација са догађајима као што су конфликти, промашаји и погоци или промене стања као што је промена дужине секвенци или кашњења на магистралама. Просторна дистрибуција параметара (види Сliku 6.4.a) такође може да укаже на неједнак приступ ресурсима унутар DRAM меморије.

7.3 Дебаговање проблема у вези са перформансама

Код ове примене важно је да се избегне „врћење у круг“, односно „јурење за сопственим репом“. Да би то било могуће, перформансе свих компоненти на критичној путањи морају бити добро дефинисане и на најнижим нивоима грануларности. То у случају предложене методологије омогућава тачност карактеризације на нивоу циклуса.

7.3.1 Анализа критичних сценарија

Анализа критичних сценарија је анализа рада, обично кратких, секвенци које се јављају спорадично, али имају битан ефекат на рад система. С обзиром да се код анализе врло кратких секвенци јавља ивични ефекат који утиче на тачност метрике, препоручује се да се у овом случају директно користе резултати карактеризације индивидуалних меморијских циклуса.

7.3.2 Трансакциона анализа

Трансакциона анализа је анализа рада система праћењем појединачних трансакција кроз систем. Такав тип анализе је потребан када се дебагује проблем који из разних разлога није могуће анализирати на други начин. Овај тип анализе је једино могуће извршити применом предложене методологије, јер ни једна друга методологија не омогућава тачност на нивоу циклуса која је потребна код ове врсте примене.

7.4 Карактеризација радног оптерећења

Карактеризација радног оптерећења омогућава процену како ће дато радно оптерећење утицати на перформансе система. Предложена методологија подржава ову врсту примене тако што омогућава разликовање функционалних и есенцијалних циклуса од слободних циклуса.

Радно оптерећење карактеришу команде читања и уписа, тј. есенцијални циклуси, док функционални циклуси карактеришу рад DRAM меморије. Идентификовање есенцијалних циклуса се може вршити тако што се из радног оптерећења избаце све команде осим команди за читање и упис и онда се изврши карактеризација циклуса применом ACDC модела на идеалну меморију. Сви заузети циклуси ће у том случају бити есенцијални (види Дефиницију 5.9).

У реалности није неопходно уклањати неесенцијалне команде из записа и накнадно га процесирати ради карактеризације радног оптерећења. Уместо тога, могуће је истовремено вршити карактеризацију циклуса ради анализе перформанси (узимајући у обзир све команде) и идентификацију есенцијалних циклуса ради карактеризације радног оптерећења (узимајући у обзир само команде за читање и упис).

7.5 Анализа потрошње енергије

Методологија подржава анализу потрошње енергије тако што омогућава тачну идентификацију трајања, као и тачне тренутке прелаза између различитих стања и режима рада DRAM кола. То дозвољава временски тачно моделовање струја и напона који се користе у моделу потрошње.

Постоје три типа модела потрошње енергије DRAM меморије. Један тип модела се базира на тачном рачунању промене потрошње делова DRAM кола у складу са променом режима рада и тренутним стањем. Код ових модела се полази од тога да је структура DRAM кола у потпуности позната, а модел DRAM меморије који се користи за процену потрошње се формира апстракцијом његове структуре.

Пример таквог модела је [47], где се промена енергетског стања кола прати са грануларношћу од једног циклуса, а ефекат на потрошњу се рачуна применом коначног аутомата чија стања одговарају различитим енергетским стањима кола. Овакви модели су веома тачни, али веома захтевни у погледу времена и ресурса потребних за симулацију, јер се утицај на промену енергетског стања кола ради за сваки циклус.

Други тип модела потрошње енергије DRAM кола су емпиријски модели. Код ових модела се полази од претпоставке да структура DRAM меморије није позната. Модел потрошње се добија статистичком обрадом резултата добијених за различите пројектне параметре. Овај приступ захтева коришћење комерцијалних алата, који нису лако доступни и ретко се користи у пракси.

Трећи тип модела је комбинација прва два типа. Код ових модела се захтева одређен ниво познавања структуре DRAM кола, као и емпиријских података о потрошњи у случајевима када се потрошња мења. Модел потрошње се базира на рачуну промене енергетског стања кола приликом промене стања или режима рада DRAM кола. Овај приступ је коришћен у [48] за моделирање потрошње DDR меморије и у [49] за моделирање потрошње DDR2 меморије. У [50] је систематизован приступ из [48] и [49] и предложен је формалан метод за креирање модела потрошње произвољних SDRAM меморија базиран на минималном дозвољеном трајању операција у одређеном режиму рада. Коришћење минималног времена трајања чини да модел по природи даје песимистичну процену потрошње. Стога је у [51] дат предлог да се тај модел унапреди применом тачног времена трајања операција уместо минималних. На основу овог метода је креиран DRAMPower алат за мерење потрошње енергије DRAM меморија [52] који може да прихвати као улаз DRAM команде са меморијске магистрале или меморијске захтеве који стижу на улаз секвенцера DRAM команди, за шта је обезбеђен модел меморијског контролера.

Поред тога што подржава постојеће моделе и методе за анализу потрошње енергије DRAM кола, методологија предложена у овој дисертацији може помоћи да се повећа тачност модела потрошње енергије тиме што омогућава раздвајање стања када су DRAM коло и меморијска магистрала активни (активни циклуси) и

када је DRAM коло активно, али меморијска магистрала није активна (режијски циклуси).

7.6 DRAM верификација

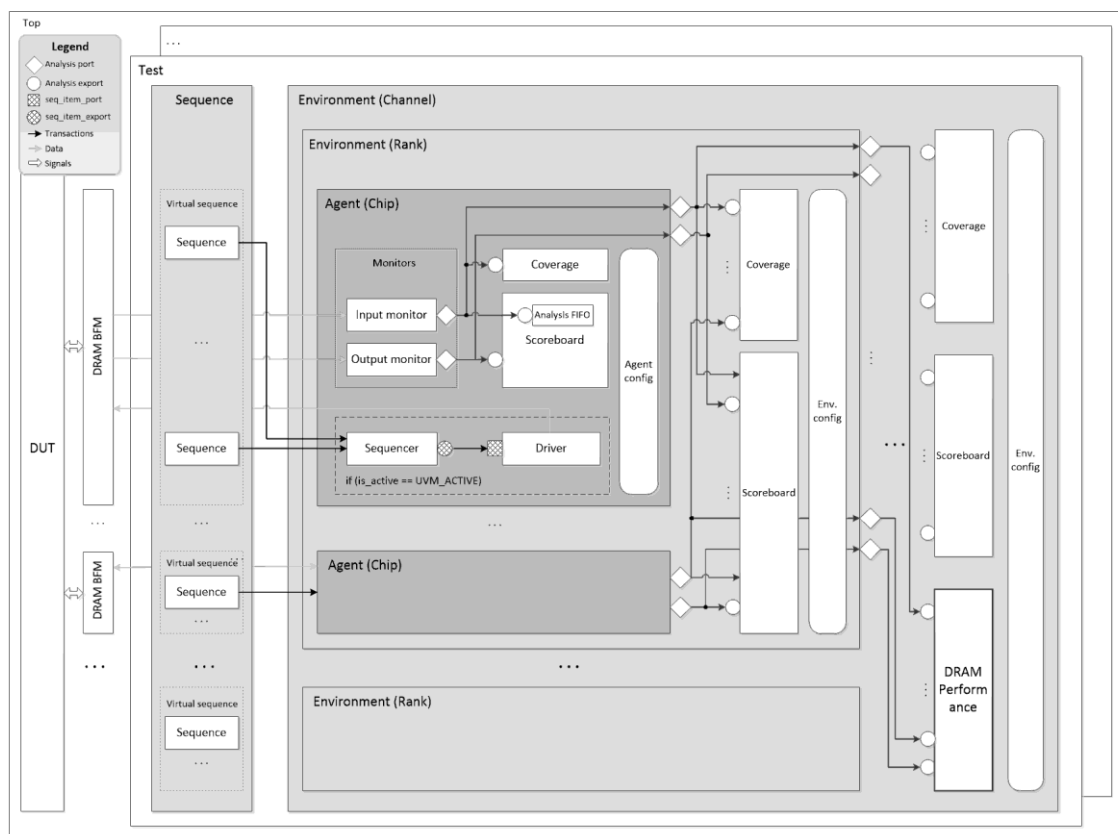
Једна од кључних фаза у методологији за мерење и анализу перформанси DRAM меморија је креирање DCT FSM, коначног аутомата чија улога је да генерише временска ограничења за сваки прелаз између DRAM команди. Временска ограничења су потребна како би се могла извршити тачна карактеризација меморијских циклуса. Генерисана временска ограничења за све прелазе између DRAM команди уједно представљају потребне и довољне услове за проверу да ли DRAM трансакције на меморијској магистрали задовољавају одговарајући DRAM протокол.

7.6.1 Функционална верификација

На основу чињенице да DCT FSM садржи сва временска, просторна и оперативна ограничења која важе код прелаза између DRAM команди, DCT FSM се може користити за ефикасну функционалну верификацију уређаја који имплементирају DRAM протокол.

На Слици 7.2 је приказана структура верификационе компоненте формиране према UVM (енг. Universal Verification Methodology) методологији [53], [54]. Овакве компоненте називају се још и UVC (енг. Universal Verification Component) и садрже све потребне елементе за функционалну верификацију уређаја који имплементирају одређени DRAM протокол. Имплементација верификационе компоненте према UVM се најчешће ради у језику SystemVerilog [55], [56].

Верификација сагласности са DRAM протоколом се обично ради на нивоу индивидуалних DRAM чипова, јер је протокол у DRAM спецификацији дефинисан на том нивоу. Верификација на нивоу целе меморије се имплементира хијерархијски, у складу са UVM методологијом. Најпре се групишу агенти за сваки од чипова у оквиру једног ранка, а затим се ранкови групишу у оквиру једног



Слика 7.2 – Универзална компонента за верификацију DRAM протокола

На слици је представљена верификациона компонента пројектована према UVM методологији која садржи подкомпоненте за функционалну верификацију DRAM протокола и подкомпоненте за верификацију DRAM перформанси. DUT=Device Under Test, BFM=Bus Functional Model.

меморијског канала. Уколико има више независних меморијских канала, потребно је инстанцирати исту верификациону компоненту за сваки канал посебно.

7.6.2 Верификација перформанси

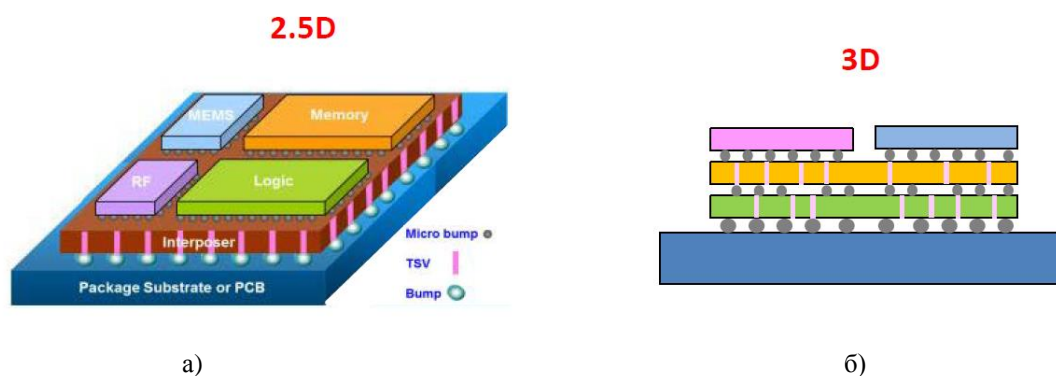
Компонента за мерење и анализу перформанси се може имплементирати на нивоу меморијског канала. Разлог зашто нема потребе да се ова компонента имплементира на нижим нивоима је што све меморијске трансакције иду преко меморијске магистрале, одакле се могу преузети и процесирати у одговарајућој компоненти означеној са „DRAM Performance“ на слици.

Верификација перформанси подразумева додавање нових компоненти за прикупљање података о покривању у вези са перформансама, слично као што

постоје компоненте за функционално покривање. Добијени резултати покривања се могу користити у анализи перформанси. Тако се семпловањем есенцијалног искоришћења, препоручљиво са неколико различитих ширина „прозора“ (најбоље скалираних логаритамски: нпр. 10, 100 и 1000 циклуса), могу на крају симулације аутоматски добити статистички који суштински представљају статичку слику о промени перформанси у времену. Слично се може урадити и са другим параметрима метрике.

7.7 Примена на 3D DRAM меморије

Раст перформанси DRAM меморија у будућности није могуће обезбедити само повећањем броја prefetch бита и пропорционалним повећањем брзине рада DRAM интерфејса. Постојеће планарне технологије достижу свој лимит, тако да проблем „меморијског зида“ поново постаје актуелан [57]. Једно од решења за тај проблем се налази у увођењу нових технологија за повезивање и паковање меморијских чипова са другим компонентама у оквиру SoC базираних на примени посредничког везног слоја – интерпозера (енг. interposer). Овај начин везивања је познат и као 2.5D интеграција (Слика 7.3.а). Друго решење за овај проблем се налази у примени нових технологија за израду DRAM чипова које су базиране на



Слика 7.3 – 2.5D и 3D интеграција меморијских кола

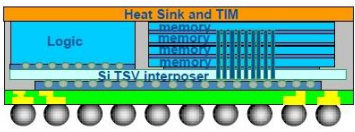
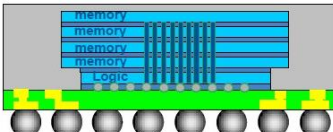
2.5D интеграција подразумева коришћење интерпозера за повезивање са другим компонентама, а 3D интеграција вертикално повезивање полупроводничких слојева помоћу TSV.

тродимензионалном, односно вертикалном, повезивању више слојева DRAM полупроводничког субстрата помоћу TSV (енг. Through-Silicon Vias) у оквиру истог паковања. Овај начин везивања је познат и као 3D интеграција (Слика 7.3.б). Највећи проблем код 3D интеграције је хлађење чипа са дигиталном логиком. Ова два приступа се могу и комбиновати.

Истовремени приступ већем броју DRAM полупроводничких слојева код 3D интеграције омогућава знатно повећање ширине меморијске магистрале, а тиме и већи проток података. То омогућава смањење потребне брзине рада, што, уз инхерентно скраћивање дужине веза, смањује и потребни радни напон. Снижавањем брзине рада и радног напона смањује се и потрошња по јединици површине DRAM меморије у односу на чипове базиране на стандардном дводимензионалном (2D) повезивању.

7.7.1 Wide I/O меморије

Wide I/O [58] и Wide I/O 2 [59] су JEDEC стандарди за DRAM меморије које су намењене за примену у системима са захтевима за великим протоком података

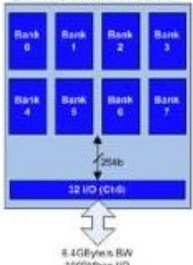
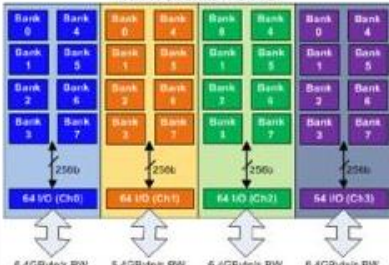
	Computing Wide IO (High Performance)	Mobile Wide IO
Structure Limitation	Thermal	Package Height
Cost	Dependent on design, technology and cooling technique	Serious
Data Band Width (Speed)	≤ 64 GB/s	≤12.8 GB/s
Power	10-150W	2-20W
Interposer	Can be used	Not Used
Structure for Thermal	Use Heat Sink and TIM	-
Structure		

Слика 7.4 – Wide I/O DRAM

Извор: JEDEC

и ниском потрошњом, као и у мобилним системима. У првом случају се због хлађења процесорске логике мора користити 2.5D интеграција, а у другом случају је могућа 3D интеграција због нижих термалних ограничења (Слика 7.4). Полупроводнички DRAM слој се у Wide I/O терминологији назива слајс (енг. slice).

Код Wide I/O, сваки слајс има 4 меморијска канала, а сваки канал има 4 банке. Битови одговарајућих меморијских канала из различитих слајсова заједно чине један физички меморијски канал ширине 128 бита. Укупно постоје 4 физичка канала, тако да је укупна ширина магистрала података за све канале 512 бита. Број слајсова у једном паковању је ограничен висином кућишта чипа и може их бити максимално 4, што уједно лимитира максимални капацитет DRAM меморије у једном паковању. Максимална брзина рада DRAM кола је 200 MHz, што омогућава проток до 12.8 GB/s. Користи се Single Data Rate (SDR) пренос података, а протокол готово у потпуности одговара JEDEC SDR SDRAM стандарду.

	LPDDR3 & LPDDR3E	LPDDR4	Wide IO2
Die Organization	<p>1ch X 8 banks X 32 IO</p>  <p>256b 32 IO (CH0) 6.4GByte/s BW 1000Mbps IO</p>	<p>2ch X 8banks X16 IO</p>  <p>256b 16 IO (CH0) 16 IO (CH1) 6.4GByte/s BW 3200Mbps IO 6.4GByte/s BW 3200Mbps IO</p>	<p>4ch X 8banks X 64 IO</p>  <p>256b 64 IO (CH0) 64 IO (CH1) 64 IO (CH2) 64 IO (CH3) 6.4GByte/s BW 800Mbps IO 6.4GByte/s BW 800Mbps IO 6.4GByte/s BW 800Mbps IO 6.4GByte/s BW 800Mbps IO</p>
Channel #	1	2	4 & 8
Bank #	8	8 per channel (16 per die)	32 per die
Density	4Gb – 32Gb	4Gb – 32Gb	8Gb – 32Gb
Page Size	4KByte	2KByte	4KByte (4ch die), 2KB (8ch die)
Max BW per die	6.4GB/s, 8.5GB/s (overclocking)	12.8GB/s, 17GB/s (overclocking)	25.6GB/s & 51.2GB/s 34GB/s & 68GB/s(overclocking)
Max IO Speed	2133Mbps	4266Mbps	1066Mbps
Signal Pin #	62 per die	66 per die	~430 per die (4ch die), ~850 per die(8ch die)
Package	POP, MCP	POP, MCP	KGD,

Слика 7.5 – Поређење (3D) Wide I/O 2 и (2D) LPDDR3/4 меморија

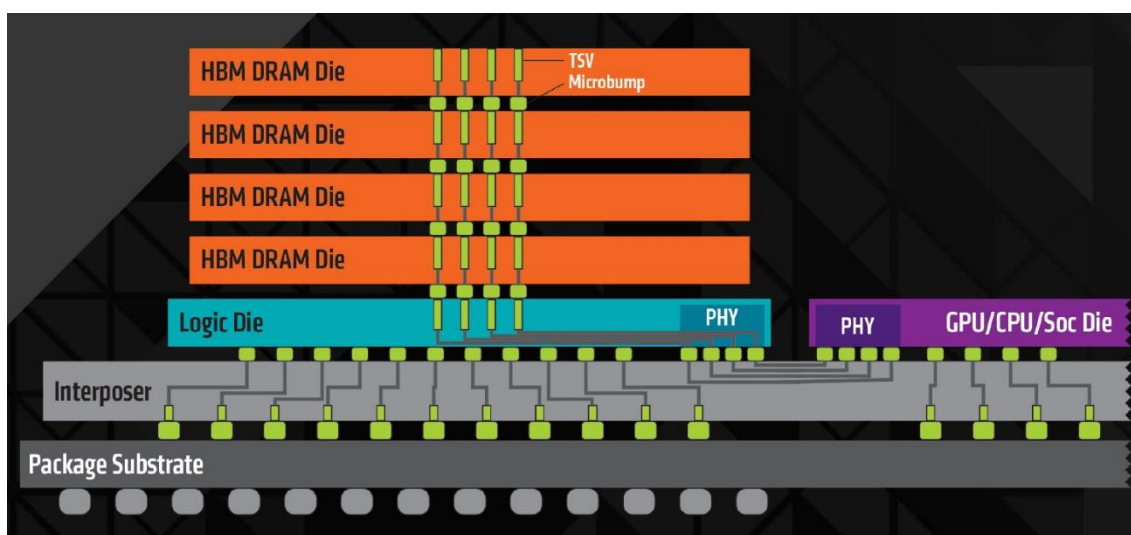
Извор: JEDEC

Код Wide I/O 2, магистрала података је ширине 64 бита. Сваки слајс има 4 или 8 меморијских канала, односно 256 или 512 бита података, а сваки канал има 8 банки. Број слајсова у једном паковању је ограничен на 4. Максимална брзина рада DRAM кола је 200-266 MHz, а користи се 4n prefetch техника. Максимални проток података са 4 канала на 200 MHz је 25.6 GB/s, а са 8 канала 51.2 GB/s. Користи се Double Data Rate (DDR) пренос података, а протокол готово у потпуности одговара JEDEC DDR2 SDRAM стандарду. На Слици 7.5 су упоређене карактеристике Wide I/O 2 3D меморија са LPDDR3 и LPDDR4 планарним меморијама.

С обзиром да су Wide I/O и Wide I/O 2 DRAM протоколи готово идентични постојећим SDR и DDR2 JEDEC стандардима, предложена методологија за мерење и анализу перформанси се може у потпуности применити и на ове типове меморија.

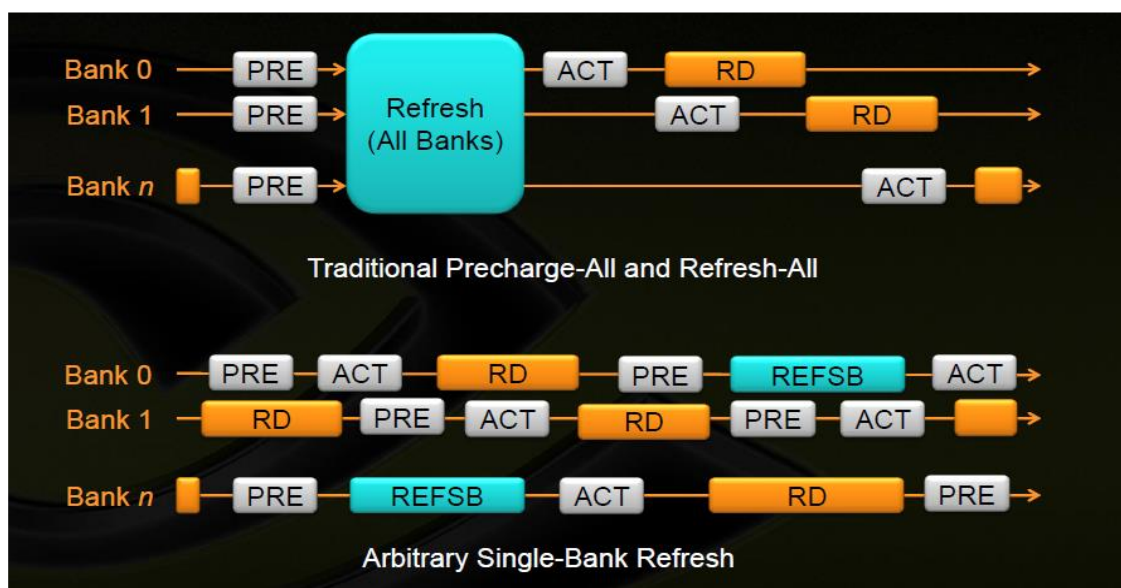
7.7.2 HBM меморије

HBM [60] и HBM 2 [61] су JEDEC стандарди за DRAM меморије које су намењене за примену у системима високих перформанси са захтевима за великим протоком података, као што су графички акцелератори (Слика 7.6).



Слика 7.6 – HBM DRAM

Извор: AMD

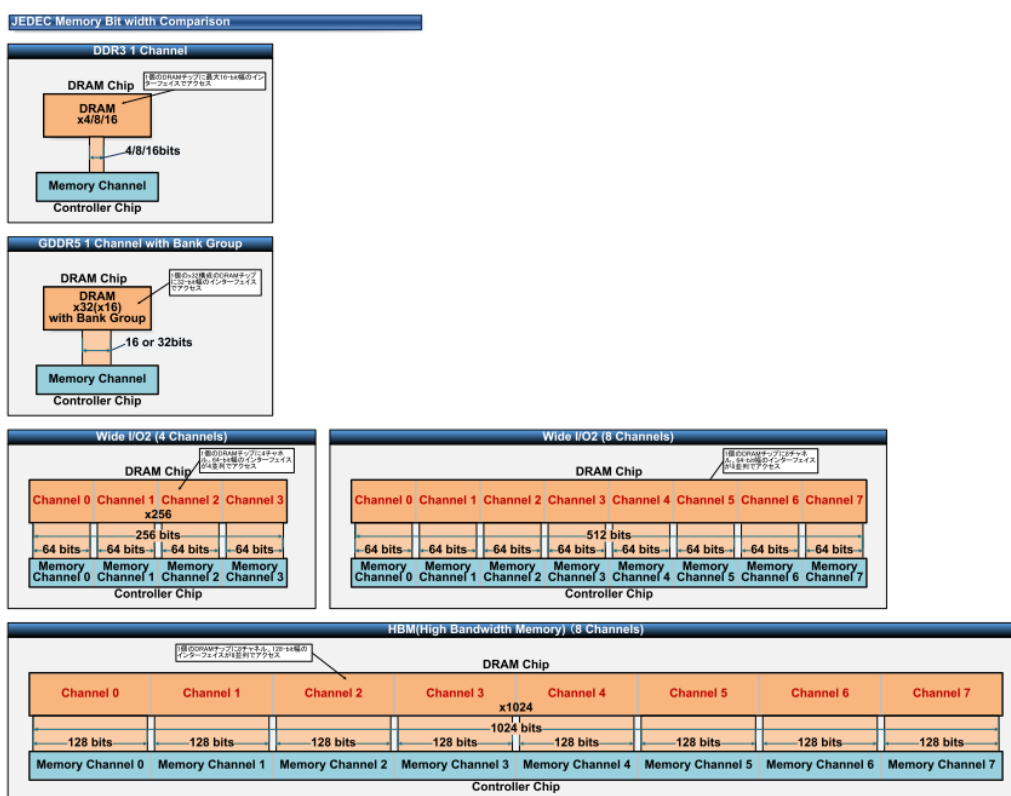


Слика 7.7 – Освежавање појединачних банки код HBM DRAM

Код HBM меморија, сваки DRAM слој има 2 независна меморијска канала, а сваки канал 8 или 16 банки. Магистрала података за сваки канал је ширине 128 бита. Подржано је до 8 слојева по паковању и не више од укупно 8 канала. То даје максималну ширину магистрале података од 1024 бита. Користи се $2n$ prefetch техника и Double Data Rate (DDR) пренос података са burst length величине 2 или 4. Максимални проток података при брзини од 250 MHz је стога 128 GB/s. DRAM протокол готово у потпуности одговара JEDEC DDR SDRAM стандарду, осим што је додата и команда за освежавање појединачних банки (Слика 7.7).

Код HBM 2 меморија, главна разлика у односу на оригиналну HBM спецификацију је у већој брзини рада и увођењу концепта псеудо-канала. Сликвито поређење пропусне моћи DDR3, GDDR5, Wide I/O и HBM меморија је дато на Слици 7.8.

Пошто су HBM и HBM 2 DRAM протоколи суштински слични постојећем DDR JEDEC стандарду, може се закључити да се предложена методологија за мерење и анализу перформанси може применити и на ове типове DRAM меморија.

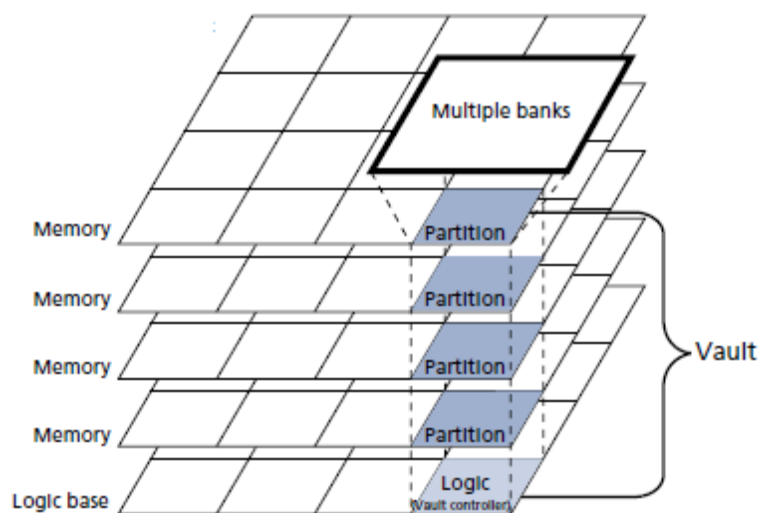


Слика 7.8 – Поређење пропусне моћи DDR3, GDDR5, Wide I/O и HBM меморија

7.7.3 HMC меморије

HMC [62] и HMC 2 [63] су стандарди за DRAM меморије које су намењене за примену у HPC (енг. High Performance Computing) системима, као што су суперрачунари, акцелераторске картице намењене графици или масовним прорачунима. Ови стандарди нису усвојени од стране JEDEC-а, али их подржава већи број компанија попут Intel, Samsung, Micron, Microsoft, ARM, HP, Xilinx и Altera.

Структура HMC DRAM меморије је дата на Слици 7.9. Меморија се састоји од 4 или 8 DRAM слојева и једног логичког слоја. Сваки слој садржи до 64 партиције. Скуп истих партиција из свих слојева чини комору (енг. vault), а свака од комора је потпуно независна и њом управља посебан меморијски контролер. Комуникација је пакетизована и одвија се преко серијског интерфејса са 8 или 16 дуплекс серијских линија по линку. Свака линија има проток од 10-15 Gb/s, а



Слика 7.9 – Структура HMC DRAM

максималан број линкова је 7, што даје максималан могући проток од 320-480 GB/s у пуној конфигурацији.

Иако је екстерна комуникација са HMC меморијом пакетизована, када се пакети декодирају, захтеви за приступ меморији се на основу адресе упућују меморијским контролерима који управљају одговарајућим коморама. Пошто контролери морају управљати DRAM меморијом у складу са неким од DRAM протокола, то значи да је могуће применити предложену методологију за мерење и анализу перформанси на сваку комору појединачно.

8 Закључак

Основни циљ дисертације је био проналажење начина да се превазиђу суштински и реални проблеми који постоје у вези са мерењем и анализом перформанси DRAM меморија. Као резултат тога створена је нова теоријска основа за мерење и анализу меморијских перформанси и формулисана је одговарајућа методологија која дефинише како да се мерење и анализа спроводе у пракси. Најважније компоненте предложене методологије чине:

- Општи функционални модел DRAM меморија у виду коначног аутомата параметризованог временским и функционалним параметрима из DRAM спецификације (DCT FSM).

Модел омогућава да се DRAM меморија дефинише као генерички уређај, при чему се карактеристике сваке конкретне имплементације DRAM меморије могу у потпуности моделирати преко одговарајућих параметара. Функционални модел се може користити не само у оквиру предложене методологије у области системске анализе, већ и независно, у областима као што су пројектовање DRAM контролера, верификација DRAM протокола и друге.

- Метрика за мерење и анализу перформанси DRAM меморија са тачношћу на нивоу циклуса

Предложена метрика поставља нов теоријски основ за рад у области мерења и анализе перформанси DRAM меморија. Помоћу ње се ова област систематизује на чврстим математичким основама и из инжењерске вештине пребацује у домен научно-стручне дисциплине.

- Метод за процену и мерење максимума перформанси DRAM меморија са тачношћу на нивоу циклуса

Метод даје тачну динамичку процену максимума перформанси и на тај начин омогућава решавање проблема интерпретације измерених резултата. Поред тога, омогућава и карактеризацију неискоришћених циклуса према узроку на оне који настају услед недовољног броја захтева (гладовање) и на оне који настају услед неоптималног рада меморијског контролера.

- Модел за карактеризацију меморијских циклуса са тачношћу на нивоу циклуса (ACDC)

Модел служи за тачну карактеризацију меморијских циклуса на слободне и заузете. У крајњој линији, то омогућава да се перформансе DRAM меморије могу мерити и анализирати на највишем нивоу апстракције, без потребе да се разумеју детаљи рада DRAM меморије. На тај начин се знатно олакшава рад у областима као што су системска и архитектурална анализа где комплексност система захтева тачно, али једноставно, моделовање.

Предложена методологија доноси суштински напредак у односу на постојећа решења. Њене најважније предности са теоријског становишта су:

- максимална тачност
- могућност прецизне процене теоријског максимума перформанси
- могућност идентификације главних узрока субоптималног рада
- комплетност (могућност анализе свих типова DRAM трансакција)

Са становишта практичне примене, најважније предности су:

- независност од архитектуре система на коме се генерише радно оптерећење
- независност од величине радног оптерећења
- преносивост (могућност имплементације на различитим платформама)
- ефикасност (брзо генерисање резултата уз мали напор корисника)
- ниска цена и комплексност имплементације и верификације

Предложена методологија може да замени постојећа решења у свим доменама где је потребно побољшати тачност и ефикасност анализе. Уједно,

омогућава се и примена у потпуно новим областима, попут анализе критичних сценарија (анализа секвенци које се јављају спорадично, али имају битан ефекат на рад система), трансакционе анализе система (анализа рада система праћењем појединачних трансакција), компаративне анализе (поређење резултата са различитих платформи, за различита радна оптерећења или у различитим периодима) и др. Методологија омогућава релативно једноставну примену на инжењерском нивоу уз мали утрошак ресурса и тиме обезбеђује висок ниво ефикасности и употребљивост у пракси.

Као резултат систематизације на чврстим теоријским основама и решавања кључних проблема који су раније постојали у вези са мерењем и анализом перформанси DRAM меморија, направљен је суштински помак у овој области. Тај напредак омогућава прелазак анализе перформанси DRAM меморија из домена инжењерске вештине у домен научно-стручне дисциплине и подизање процеса анализе рада целокупног рачунарског система на квантитативно и квалитативно виши ниво.

Прилог А – Еквивалентност ante factum и post factum ограничења

Доказати да важи следећи исказ: Ако важи тврдња „прелаз t (из претходне команде у следећу команду) се може десити ако и само ако су услови $c_1, \dots, c_i, \dots, c_n$ задовољени“, онда важи тврдња „ако се десио прелаз t и услови $c_1, \dots, c_{i-1}, c_{i+1}, \dots, c_n$ су били задовољени, онда је услов c_i морао бити задовољен“.

Доказ исказа се своди на доказивање да је следћи израз таутологија:

$$(c_1 \cdots c_i \cdots c_n \Leftrightarrow t) \Rightarrow (c_1 \cdots c_{i-1} \cdot c_{i+1} \cdots c_n \cdot t \Rightarrow c_i)$$

Применом једнакости $a \Rightarrow b = \bar{a} + b$ и $a \Leftrightarrow b = a \cdot b + \bar{a}$ на дати израз, добија се:

$$\begin{aligned} & (c_1 \cdots c_i \cdots c_n \Leftrightarrow t) \Rightarrow (c_1 \cdots c_{i-1} \cdot c_{i+1} \cdots c_n \cdot t \Rightarrow c_i) \\ &= (c_1 \cdots c_i \cdots c_n \cdot t + c_1 \cdots c_i \cdots c_n \cdot \bar{t}) \Rightarrow (c_1 \cdots c_{i-1} \cdot c_{i+1} \cdots c_n \cdot t + c_i) \\ &= \overline{c_1 \cdots c_i \cdots c_n \cdot t + c_1 \cdots c_i \cdots c_n \cdot \bar{t}} \cdot \overline{c_1 \cdots c_{i-1} \cdot c_{i+1} \cdots c_n \cdot t + c_i} \\ &= \overline{c_1 \cdots c_i \cdots c_n \cdot t} \cdot \overline{c_1 \cdots c_i \cdots c_n \cdot \bar{t}} + \overline{c_1 \cdots c_{i-1} \cdot c_{i+1} \cdots c_n \cdot t + c_i} \\ &= \overline{c_1 \cdots c_i \cdots c_n \cdot t} \cdot (c_1 \cdots c_i \cdots c_n + t) + \overline{c_1 \cdots c_{i-1} \cdot c_{i+1} \cdots c_n \cdot t + c_i} \\ &= (\overline{c_1 \cdots c_i \cdots c_n} + t) \cdot (c_1 \cdots c_i \cdots c_n + t) + \overline{c_1 \cdots c_{i-1} \cdot c_{i+1} \cdots c_n \cdot t + c_i} \end{aligned}$$

$$\begin{aligned}
 &= \overline{c_1 \cdots c_i \cdots c_n} \cdot \bar{t} + c_1 \cdots c_i \cdots c_n \cdot \bar{t} + \overline{c_1 \cdots c_{i-1} \cdot c_{i+1} \cdots c_n} \cdot \bar{t} + c_i \\
 &= (\overline{c_i + c_1 \cdots c_{i-1} \cdot c_{i+1} \cdots c_n}) \cdot \bar{t} + c_i \cdot c_1 \cdots c_{i-1} \cdot c_{i+1} \cdots c_n \cdot \bar{t} + \overline{c_1 \cdots c_{i-1} \cdot c_{i+1} \cdots c_n} \cdot \bar{t} + c_i \\
 &= \overline{c_i \cdot \bar{t} + c_1 \cdots c_{i-1} \cdot c_{i+1} \cdots c_n} \cdot \bar{t} + c_i \cdot c_1 \cdots c_{i-1} \cdot c_{i+1} \cdots c_n \cdot \bar{t} + \overline{c_1 \cdots c_{i-1} \cdot c_{i+1} \cdots c_n} \cdot \bar{t} + c_i \\
 &= \overline{c_i \cdot \bar{t} + c_1 \cdots c_{i-1} \cdot c_{i+1} \cdots c_n} + \bar{t} + c_i \\
 &= \overline{c_i \cdot \bar{t} + c_1 \cdots c_{i-1} \cdot c_{i+1} \cdots c_n} + \bar{t} + c_i \cdot (\bar{t} + \bar{t}) \\
 &= (\overline{c_i + c_i}) \cdot \bar{t} + \overline{c_1 \cdots c_{i-1} \cdot c_{i+1} \cdots c_n} + \bar{t} + c_i \cdot \bar{t} \\
 &= \overline{t + c_1 \cdots c_{i-1} \cdot c_{i+1} \cdots c_n} + \bar{t} \\
 &= \overline{1 + c_1 \cdots c_{i-1} \cdot c_{i+1} \cdots c_n} \\
 &= 1
 \end{aligned}$$

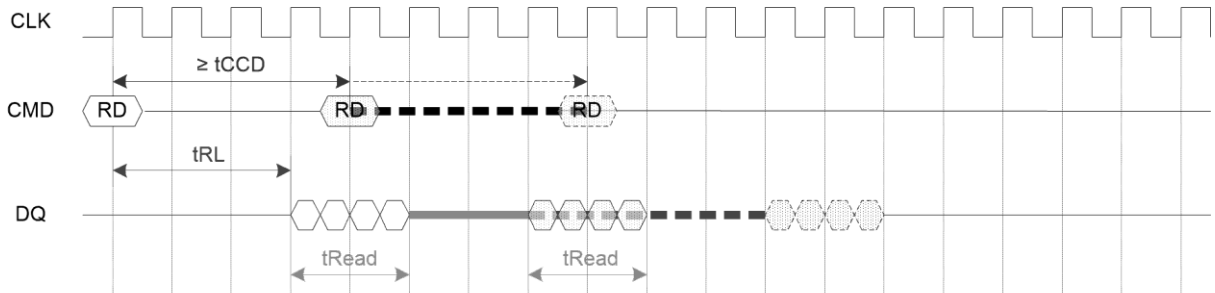
Прилог Б – Временски сигнали прелаза између DDR2 команди

Временски сигнали прелаза између DDR2 команди представљају основу за формирање DCT коначног аутомата. У овом додатку су дати временски сигнали прелаза између свих DRAM DDR2 команди од значаја за анализу перформанси.

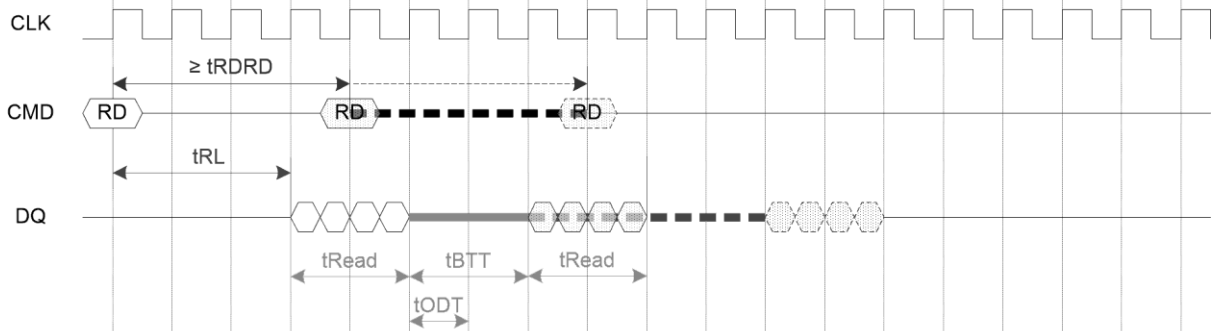
Готово идентични временски сигнали се јављају и код других типова DRAM меморија, јер је скуп релевантних команди готово исти.

Дијаграми временских сигнала показују релевантна временска ограничења, као и то како они утичу на појаву режијских циклуса у случају одређеног прелаза, како за командну магистралу, тако и за магистралу података.

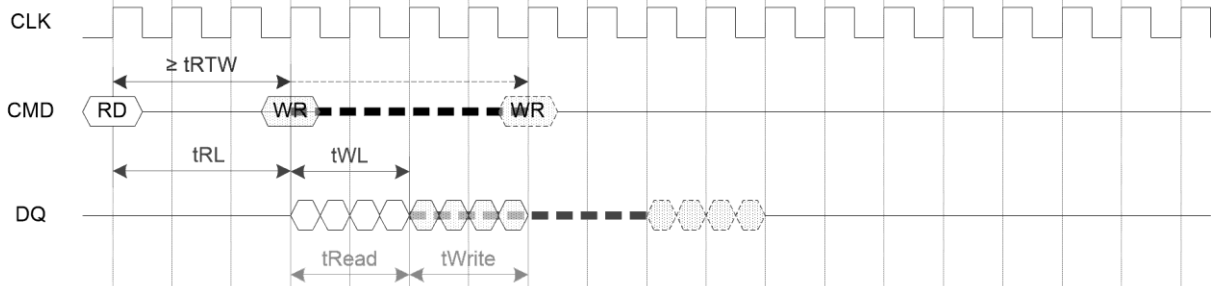
READ → READ
READ → READ with Auto-precharge



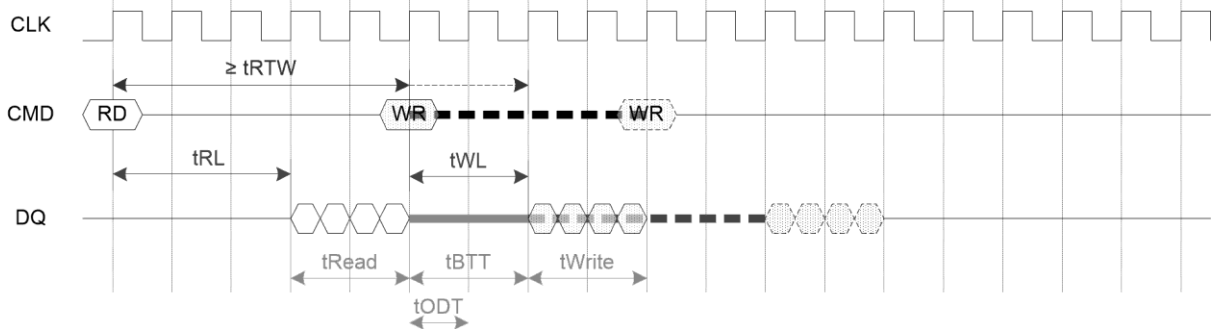
READ → READ (different rank)
READ → READ with Auto-precharge (different rank)



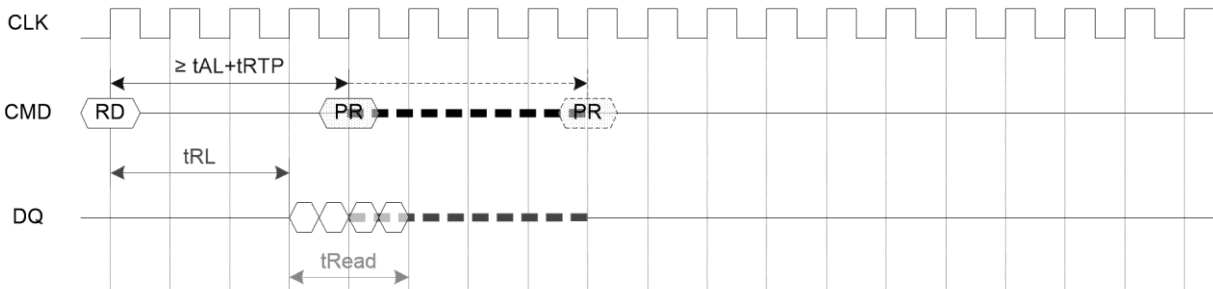
READ → WRITE
READ → WRITE with Auto-precharge



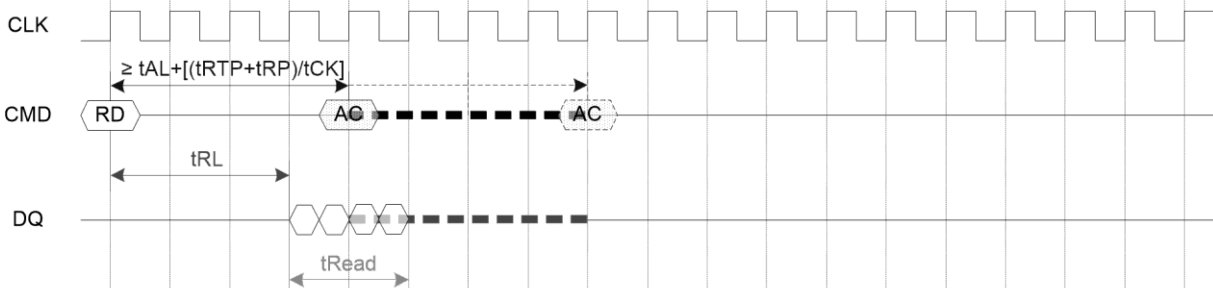
READ → WRITE (different rank)
READ → WRITE with Auto-precharge (different rank)



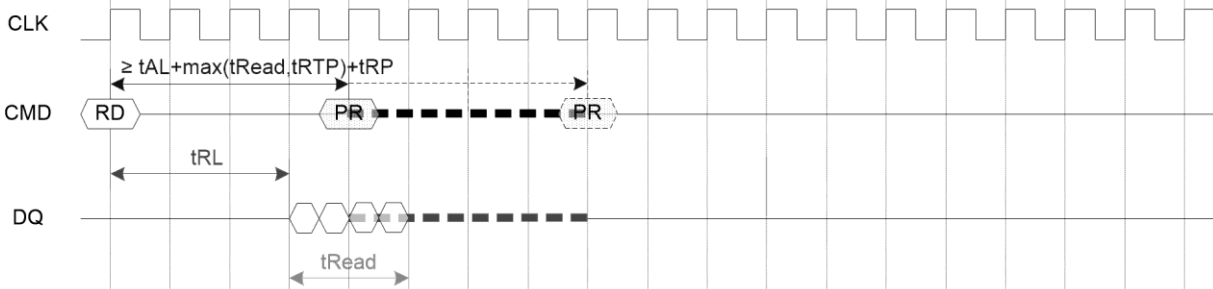
READ → PRECHARGE
READ → PRECHARGE all banks



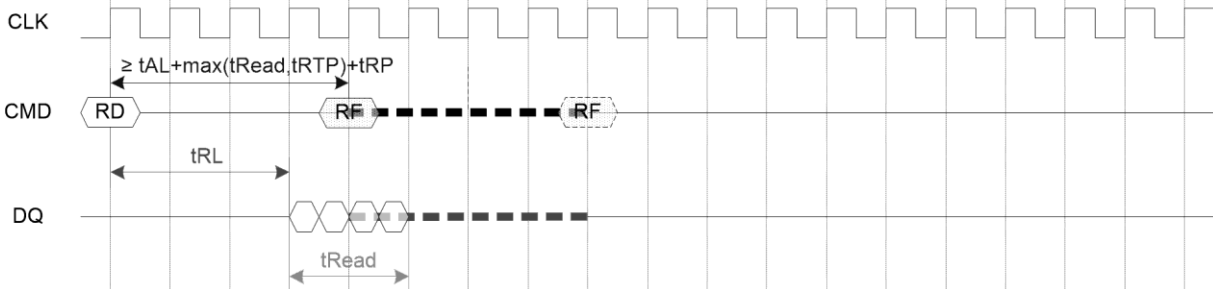
READ with Auto-precharge → ACTIVATE



READ with Auto-precharge → PRECHARGE all banks

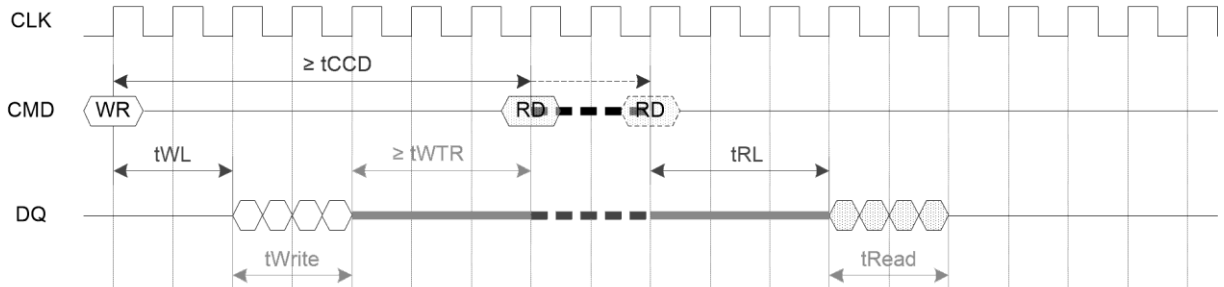


READ with Auto-precharge → REFRESH



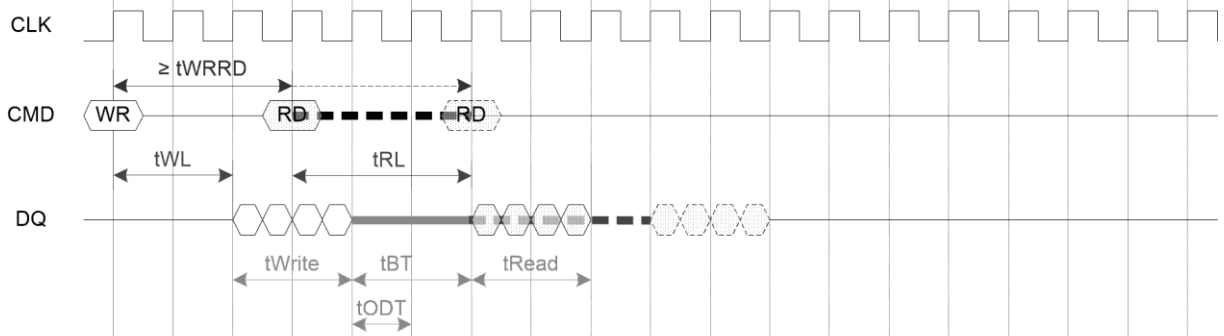
WRITE → READ

WRITE → READ with Auto-precharge



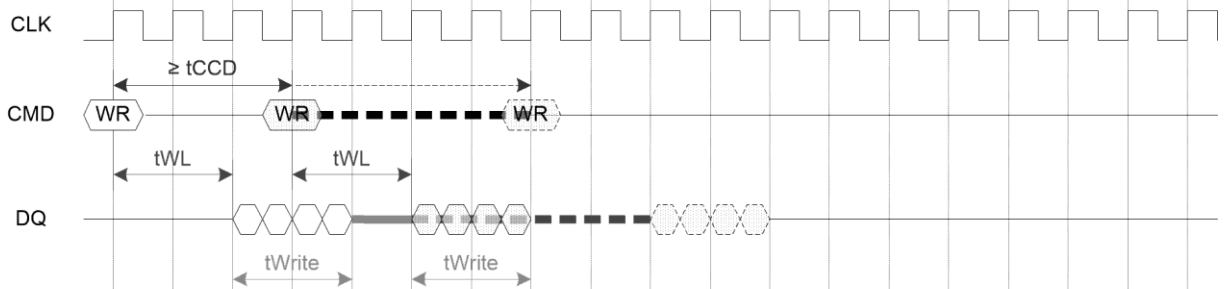
WRITE → READ (different rank)

WRITE → READ with Auto-precharge (different rank)



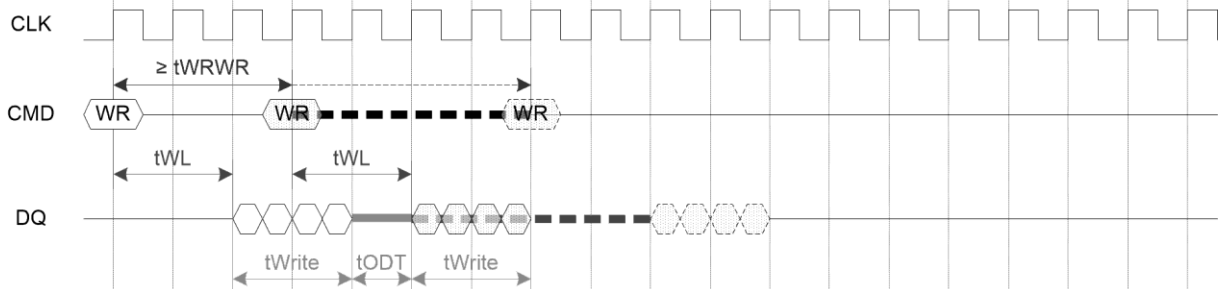
WRITE → WRITE

WRITE → WRITE with Auto-precharge

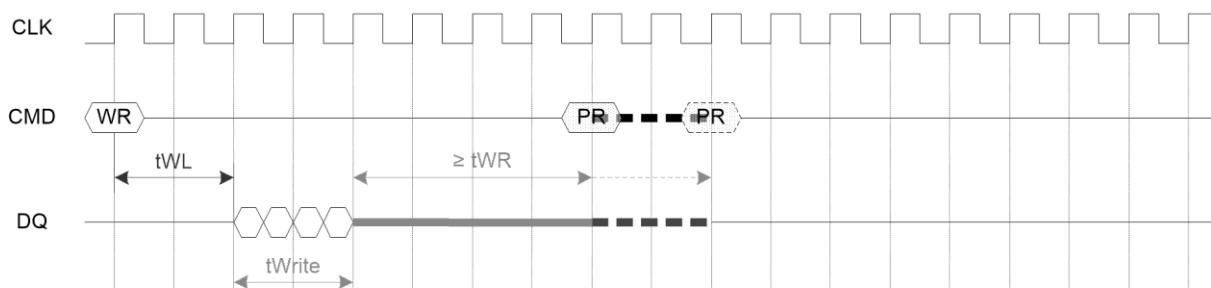


WRITE → WRITE (different rank)

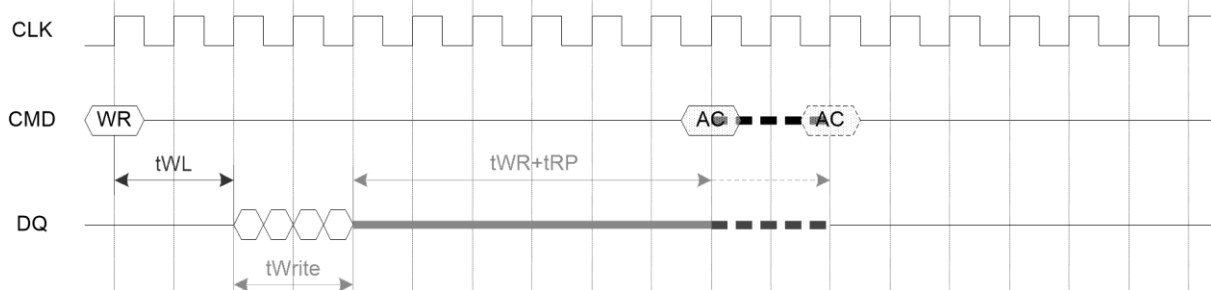
WRITE → WRITE with Auto-precharge (different rank)



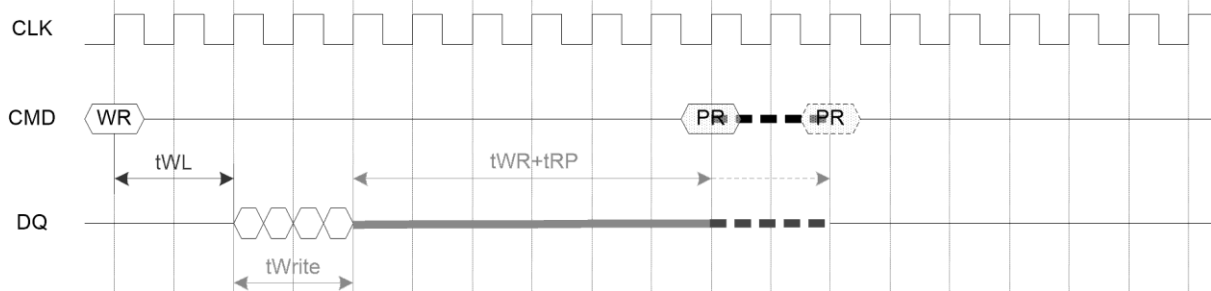
WRITE → PRECHARGE
WRITE → PRECHARGE all banks



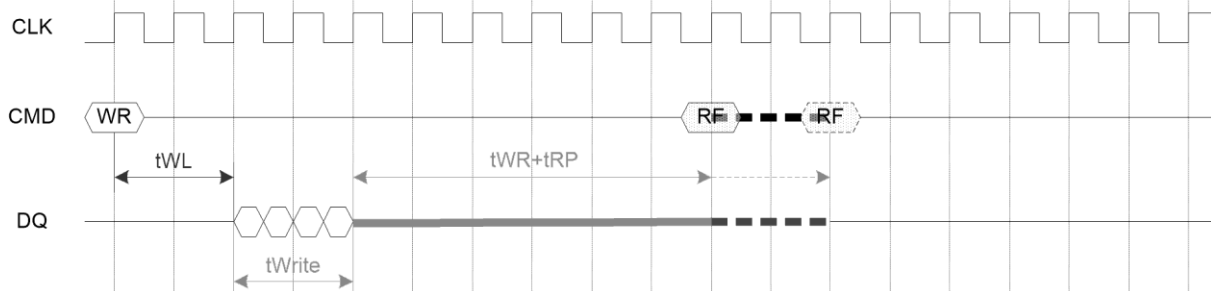
WRITE with Auto-precharge → ACTIVATE



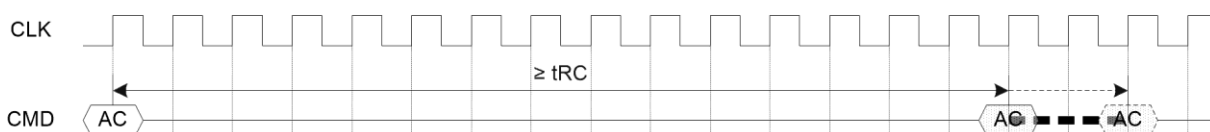
WRITE with Auto-precharge → PRECHARGE all banks



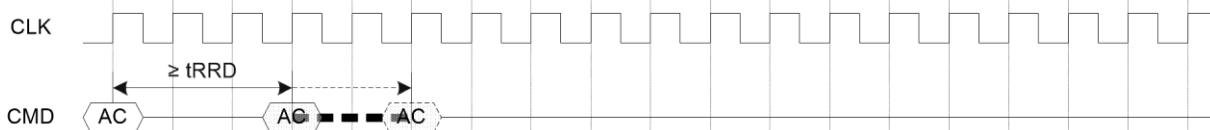
WRITE with Auto-precharge → REFRESH



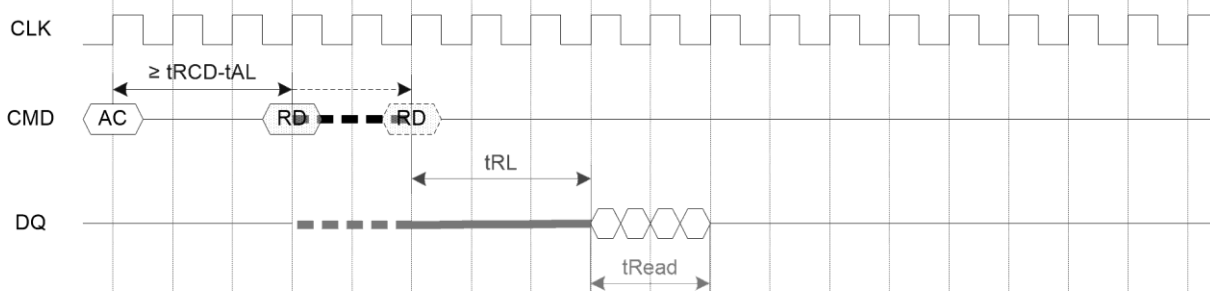
ACTIVATE → ACTIVATE



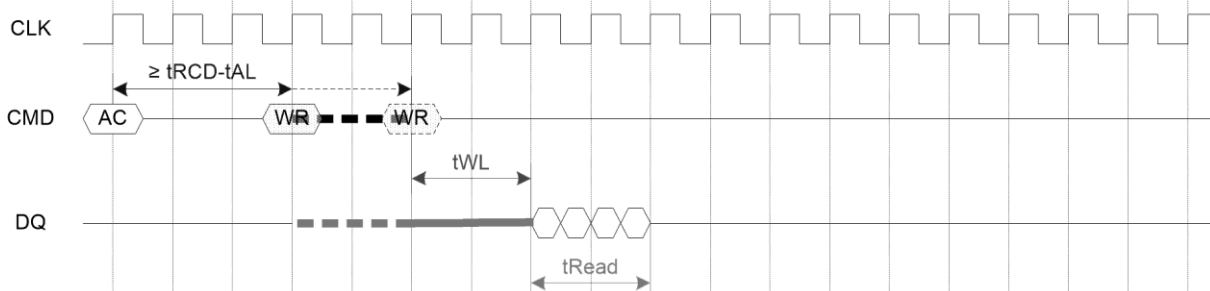
ACTIVATE → ACTIVATE (different bank)



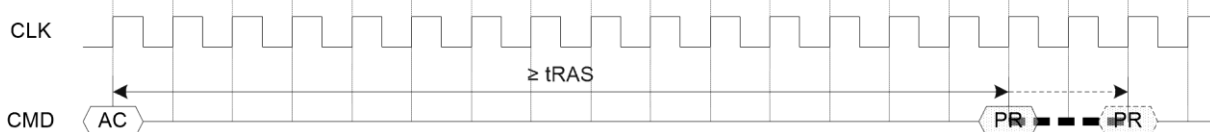
**ACTIVATE → READ
ACTIVATE → READ with Auto-Precharge**



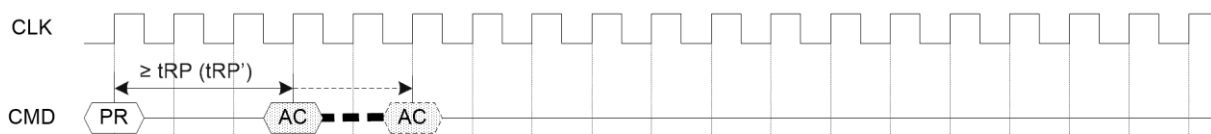
**ACTIVATE → WRITE
ACTIVATE → WRITE with Auto-precharge**



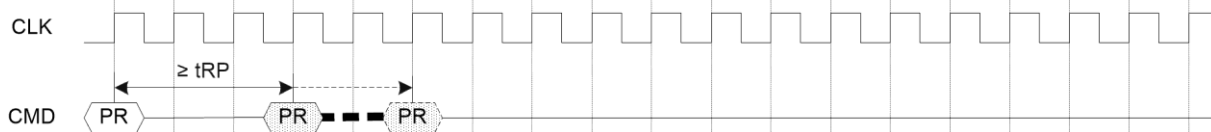
**ACTIVATE → PRECHARGE
ACTIVATE → PRECHARGE all banks**



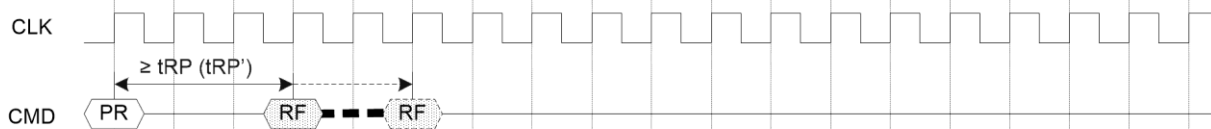
PRECHARGE → ACTIVATE
PRECHARGE all banks → ACTIVATE



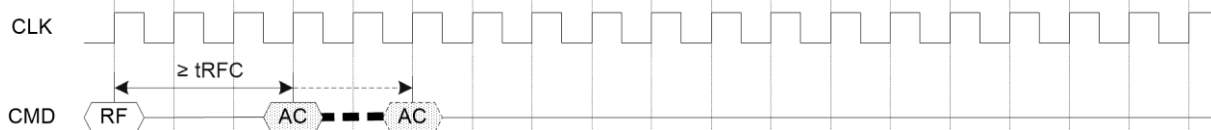
PRECHARGE → PRECHARGE all banks



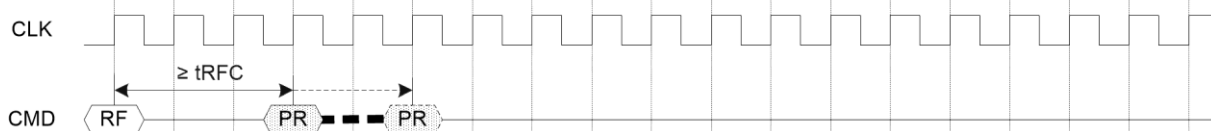
PRECHARGE → REFRESH
PRECHARGE all banks → REFRESH



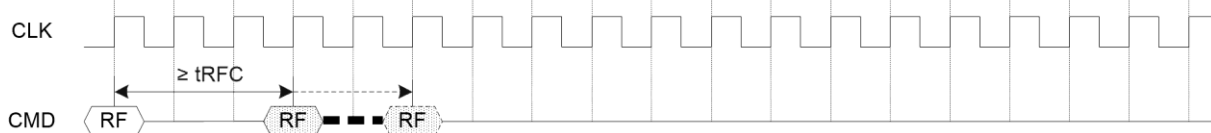
REFRESH → ACTIVATE



REFRESH → PRECHARGE



REFRESH → REFRESH



Литература

- [1] K. Suzuki, S. Swanson, “ A Survey of Trends in Non-Volatile Memory Technologies: 2000-2014“, The IEEE 7th International Memory Workshop, Monterey, CA, pp. 1-4, May 2015
- [2] B. Panzer-Steindel, “Technology, Market and Cost Trends“, CERN, April 12, 2015
- [3] L. A. Barroso, J. Clidaras, U. Hölzle, “The Datacenter as a Computer - An Introduction to the Design of Warehouse-Scale Machines, 2nd Ed”, Synthesis Lectures on Computer Architecture, Jul 2013, Vol. 8, No. 3, pp. 1-154
- [4] “The Industry-Changing Impact of Accelerated Computing”, AMD White Paper, http://sites.amd.com/us/Documents/AMD_fusion_Whitepaper.pdf, 2008
- [5] S. A. McKee, “Reflections on the memory wall”, Proceedings of the 1st conference on Computing frontiers, Ischia, Italy, 2004
- [6] B. G. Streetman, S. J. Banerjee, “Solid State Electronic Devices, Sixth Edition”, Prentice Hall, pp. 486-492, 2006
- [7] P. E. Allen, D. R. Holberg, “CMOS Analog Circuit Design, Second Edition”, Oxford University Press, 2002
- [8] R. J. Baker, “CMOS: circuit design, layout, and simulation, Revised 2nd Edition”, Wiley-Interscience, pp. 433-483, 2008
- [9] T. P. Haraszti, “CMOS Memory Circuits”, Kluwer, 2002
- [10] B. Keeth, R. J. Baker, B. Johnson, F. Lin, “DRAM Circuit Design: Fundamental and High-Speed Topics”, Wiley-IEEE Press, 2007

-
- [11] S-M. Kang, Y. Leblebici, “CMOS Digital Integrated Circuits – Analysis and Design, Third Edition”, McGraw-Hill, pp. 419-438, 2003
- [12] J. M. Rabaey, A. Chandrakasan, B. Nikolic, “Digital Integrated Circuits – A Design Perspective, Second Edition”, Prentice-Hall, pp.672-692, 2008
- [13] “Mobile LPDDR2 SDRAM“ Micron Technology, https://www.micron.com/~media/documents/products/data-sheet/dram/mobile-dram/low-power-dram/lpddr2/u80m_4gb_mobile_lpddr2_s4_sdram.pdf, 2011
- [14] “Mobile LPDDR3 SDRAM”, Micron Technology, https://www.micron.com/~media/documents/products/data-sheet/dram/mobile-dram/low-power-dram/lpddr3/253b_12-5x12-5_2ch_8-16gb_2c0f_mobile_lpddr3.pdf, 2014
- [15] “1Gb (32Mx32) GDDR5 SGRAM H5GQ1H24AFR”, Hynix, <http://www.alldatasheet.com/datasheet-pdf/pdf/333509/HYNIX/H5GQ1H24AFR.html>, 2009
- [16] “DDR3 SDRAM“, Micron Technology, https://www.micron.com/~media/documents/products/data-sheet/dram/ddr3/4gb_ddr3_sdram.pdf, 2009
- [17] “DDR4 SDRAM“, Micron Technology, https://www.micron.com/~media/documents/products/data-sheet/dram/ddr4/4gb_ddr4_sdram.pdf, 2014
- [18] “8Gb B-die DDR4 SDRAM”, Samsung, http://www.samsung.com/semiconductor/global/file/product/2015/08/8GB_DDR4_Samsung_Spec_Rev1.11_Mar.15-2.pdf, 2015
- [19] ”DDR3 SDRAM SODIMM, MT16JTF1G64HZ - 8GB”, Micron Technology Inc., https://www.micron.com/~media/documents/products/data-sheet/modules/sodimm/jtf16c256_512_1gx64hz.pdf, 2011
- [20] ”DDR3 SDRAM RDIMM, MT18JSF1G72PZ - 8GB”, Micron Technology Inc., https://www.micron.com/~media/documents/products/data-sheet/modules/parity_rdimm/jsf18c1gx72pz.pdf, 2011
- [21] D. Malech, “DDR4 LRDIMMs”, IDT White Paper, <http://www.idt.com/document/whp/ddr4-lrdimms-both-memory-capacity-and-speed>, 2014
- [22] D. Malech, S. Kuppahalli, R. Baxter, E. Caward, “DDR4 LRDIMMs Let You Have

- It All”, IDT White Paper, <https://www.idt.com/document/whp/idt-ddr4-lrdimms-let-you-have-it-all>, 2016
- [23] V. Cuppu, B. Jacob, B. Davis, T. Mudge, “High-Performance DRAMs in Workstation Environments”, IEEE Transactions on Computers, vol. 50, no. 11, pp. 1133-1153, November 2001
- [24] S. Rixner et al., “Memory access scheduling”, Proceedings of the 27th annual international symposium on Computer architecture, pp. 128-138, Vancouver, British Columbia, Canada, June 2000
- [25] “General SDRAM Functions”, JEDEC Standard 21-C, pp. 3.11.5.1-1 - 3.11.5.1-20, June 2002
- [26] “JEDEC Standard JESD79F: DDR SDRAM Specification“, February 2008
- [27] “JEDEC Standard JESD79-2E: DDR2 SDRAM Specification“, April 2008
- [28] “JEDEC Standard JESD79-3E: DDR3 SDRAM Specification“, August 2009
- [29] “JEDEC Standard JESD79-4: DDR4 SDRAM Specification“, September 2012
- [30] “JEDEC Standard JESD209B: Low Power Double Data Rate (LPDDR) SDRAM Specification“, February 2010
- [31] “JEDEC Standard JESD209-2D: Low Power Double Data Rate 2 (LPDDR2) Specification“, December 2010
- [32] “JEDEC Standard JESD209-3: Low Power Double Data Rate 3 (LPDDR3) Specification“, May 2012
- [33] “JEDEC Standard JESD209-4: Low Power Double Data Rate 4 (LPDDR4) Specification“, August 2014
- [34] “DDR SGRAM Specification: 144 Pin DDR SGRAM SO–DIMM Family”, JEDEC Standard JESD21-C, pp. 4.5.9–1 - 4.5.9–10, March 1999
- [35] “GDDR2 Specific SGRAM Functions”, JEDEC Standard JESD21-C, pp. 3.11.5.6-1 - 3.11.5.6-3, May 2005
- [36] “GDDR3 Specific SGRAM Functions”, JEDEC Standard JESD21-C, pp. 3.11.5.7-1 - 3.11.5.7-19, May 2005

-
- [37] “GDDR4 Specific SGRAM Functions: GDDR4 SGRAM Specification”, JEDEC Standard JESD21-C, pp. 3.11.5.8-1 - 3.11.5.8-72, November 2005
- [38] “JEDEC Standard JESD212: GDDR5 SGRAM Specification“, December 2009
- [39] “JEDEC Standard JESD232: GDDR5X SGRAM Specification”, November 2015
- [40] “JEDEC Standard JESD206: FB DIMM Architecture and Protocol”, January 2007
- [41] “JEDEC Standard JESD82-20A: FB DIMM Advanced Memory Buffer Specification”, March 2009
- [42] G.L.Yuan, T.M.Aamodt, “A Hybrid Analytical DRAM Performance Model”, Fifth Annual Workshop on Modeling, Benchmarking and Simulation, Austin, Texas, June 2009
- [43] B. Jacob, S. W. Ng, D. T. Wang, “Memory Systems: Cache, DRAM, Disk”, Morgan Kaufmann, 2007
- [44] D. Wang, B. Ganesh, N. Tuaycharoen, K. Baynes, A. Jaleel, B. Jacob, “DRAMsim: A Memory System Simulator“, ACM SIGARCH Computer Arch. News, Vol. 33, No. 4, pp. 100-107, September 2005
- [45] P. Rosenfeld, E. Cooper-Balis, B. Jacob, “DRAMSim2: A Cycle Accurate Memory System Simulator”, IEEE Computer Architecture Letters, Vol. 10, Issue 11, pp. 16-19, 2011
- [46] Y. Kim, W. Yang, O. Mutlu, “Ramulator: A Fast and Extensible DRAM Simulator”, IEEE Computer Architecture Letters, Volume PP, Issue 99, March 2015
- [47] Y. Joo, Y. S. Choi, H. Shim, H. G. Lee, K. Kim, N. Chang,, “Energy Exploration and Reduction of SDRAM Memory Systems”, Proceedings of the 2002 Design Automation Conference, pp 892-897, 2002
- [48] “Calculating Memory System Power for DDR SDRAM”, Micron designline, vol. 10, issue 2, <https://www.ece.umd.edu/class/enee759h.S2005/references/dl201.pdf>, 2001
- [49] “Calculating Memory System Power for DDR2”, Micron, TN-47-04, <https://www.>

- micron.com/~media/documents/products/technical-note/dram/tn4704.pdf, 2004
- [50] A. J. Joshi, S. Sambamurthy, S. Kumar, L. John, “ Power Modeling of SDRAMs“, Technical Report TR-040126-02, University of Texas, 2002
- [51] K. Chandrasekar, B. Akessons, K. Goossens, “Improved Power Modeling of DDR SDRAMs“, Proceedings of the 14th Euromicro Conference on Digital System Design, pp. 99-108, 2011
- [52] K. Chandrasekar, “DRAMPower: Open-Source DRAM Power & Energy Estimation Tool”, <http://www.drampower.info>, 2012
- [53] R. Salemi, “The UVM Primer”, Boston Light Press, 2014
- [54] K. A. Meade, S. Rosenberg, “A Practical Guide to Adopting the Universal Verification Methodology (UVM), Second Edition”, Cadence, 2013
- [55] “IEEE Standard 1800-2012: SystemVerilog - Unified Hardware Design, Specification, and Verification Language”, 2013
- [56] C. Spear, “SystemVerilog for Verification, Second Edition”, Springer, 2008
- [57] H. Reiter, “Tackling the Memory Wall with 2.5D and 3D ICs”, MEMCON, October 2014
- [58] “JEDEC Standard JESD229: Wide I/O Single Data Rate (SDR) Specification”, December 2011
- [59] “JEDEC Standard JESD229-2: Wide I/O 2 Single Data Rate (SDR) Specification”, September 2014
- [60] “JEDEC Standard JESD235: High Bandwidth Memory (HBM) DRAM”, October 2013
- [61] “JEDEC Standard JESD235A: High Bandwidth Memory (HBM) DRAM”, November 2015
- [62] “Hybrid Memory Cube Specification 1.0”, HMC Consortium, 2012
- [63] “Hybrid Memory Cube Specification 2.1”, HMC Consortium, 2015

Биографија

Основну школу, а потом Математичку Гимназију завршио је у Београду са просечном оценом 5.0. Уписао је Електротехнички факултет у Београду као другорангирани на заједничкој пријемној листи за све техничке факултете. Дипломирао је на одсеку за Рачунарску технику и информатику са укупном просечном оценом на студијама 8.93 и просеком оцена 9.45 из предмета са Катедре за рачунарску технику и информатику одбранивши рад под насловом ”Trace-driven симулација мултипроцесорских система”, а магистрирао на смеру Архитектура и организација рачунарских система и мрежа са просечном оценом 10.0 одбранивши магистарску тезу под насловом ”Мултипроцесорски систем за навигацију помоћу глобалног система за позиционирање”.

Током 1999. године био је ангажован на истраживачком пројекту из домена архитектуре мултипроцесорских система које је подржало и финансирало Министарство за науку и технологију Србије. Од 2000. године радио је на Електротехничком факултету у Београду као сарадник по уговору, а од 2001. године као стално запослен у звању асистента приправника при Катедри за рачунарску технику и информатику на предметима Архитектура рачунара, Мултипроцесорски системи, Дистрибуирани рачунарски системи, Програмски језици и Рачунарске основе примене Интернета. Од 2002. до 2006. године радио је у АТI Technologies Inc. (Торонто, Канада), једном од водећих светских произвођача графичких система за персоналне рачунаре, на месту сениор инжењера за перформансе интегрисаних VLSI система. Од 2006. до 2011. године радио је на месту сениор инжењера за ASIC дизајн у Advanced Micro Devices Inc. (AMD), једном од водећих светских произвођача процесорских система за персоналне и

серверске платформе. Од 2011. до 2014. године радио је као виши сениор инжењер за архитектуру и анализу перформанси мултимедијалних подсистема у Qualcomm Inc., водећем светском произвођачу интегрисаних система за мобилне платформе. Од 2014. године ради у компанији HDL Design House као руководилац на пројекатима из области дизајна, верификације и моделирања VLSI система.

Аутор и коаутор је две књиге и неколико радова објављених у домаћим и страним стручним часописима и конференцијама из области архитектуре и организације рачунарских система и области конкурентног програмирања. Редован је члан међународних удружења инжењера електротехнике IEEE и ACM од 2003. године.

Прилог 1.

Изјава о ауторству

Потписани мр Игор Икодиновић

Број индекса _____

Изјављујем

да је докторска дисертација под насловом

„Методологија за анализу перформанси DRAM меморија са тачношћу на нивоу циклуса“

- резултат сопственог истраживачког рада,
- да предложена дисертација у целини ни у деловима није била предложена за добијање било које дипломе према студијским програмима других високошколских установа,
- да су резултати коректно наведени и
- да нисам кршио ауторска права и користио интелектуалну својину других лица

У Београду, 08. 04. 2016. год.

Потпис докторанда



Прилог 2

Изјава о истоветности штампане и електронске верзије докторског рада

Име и презиме аутора мр Игор Икодиновић

Број индекса _____

Наслов рада „Методологија за анализу перформанси DRAM меморија са тачношћу
на нивоу циклуса“

Ментор др Вељко Милутиновић, редовни професор, Универзитет у Београду –
Електротехнички факултет

Потписани мр Игор Икодиновић

Изјављујем да је штампана верзија мог докторског рада истоветна електронској верзији коју сам предао за објављивање на порталу **Дигиталног репозиторијума Универзитета у Београду**.

Дозвољавам да се објаве моји лични подаци везани за добијање академског звања доктора наука, као што су име и презиме, година и место рођења и датум одбране рада.

Ови лични подаци могу се објавити на мрежним страницама дигиталне библиотеке, у електронском каталогу и у публикацијама Универзитета у Београду.

У Београду, 08. 04. 2016. год.

Потпис докторанда



Прилог 3.

Изјава о коришћењу

Овлашћујем Универзитетску библиотеку „Светозар Марковић“ да у Дигитални репозиторијум Универзитета у Београду унесе моју докторску дисертацију под насловом:

„Методологија за анализу перформанси DRAM меморија са тачношћу на нивоу циклуса“

које је моје ауторско дело.

Дисертацију са свим прилозима предао сам у електронском формату погодном за трајно архивирање.

Моју докторску дисертацију похрањену у Дигитални репозиторијум Универзитета у Београду могу да користе сви који поштују одредбе садржане у одабраном типу лиценце Креативне заједнице (Creative Commons) за коју сам се одлучио.

1. Ауторство
2. Ауторство – некомерцијално
3. Ауторство – некомерцијално – без прераде
4. Ауторство – некомерцијално – делити под истим условима
5. Ауторство – без прераде
6. Ауторство – делити под истим условима

(Молимо да заокружите само једну од шест понуђених лиценци, кратак опис лиценци је дат на полеђини листа).

У Београду, 08. 04. 2016. год.

Потпис докторанда

