

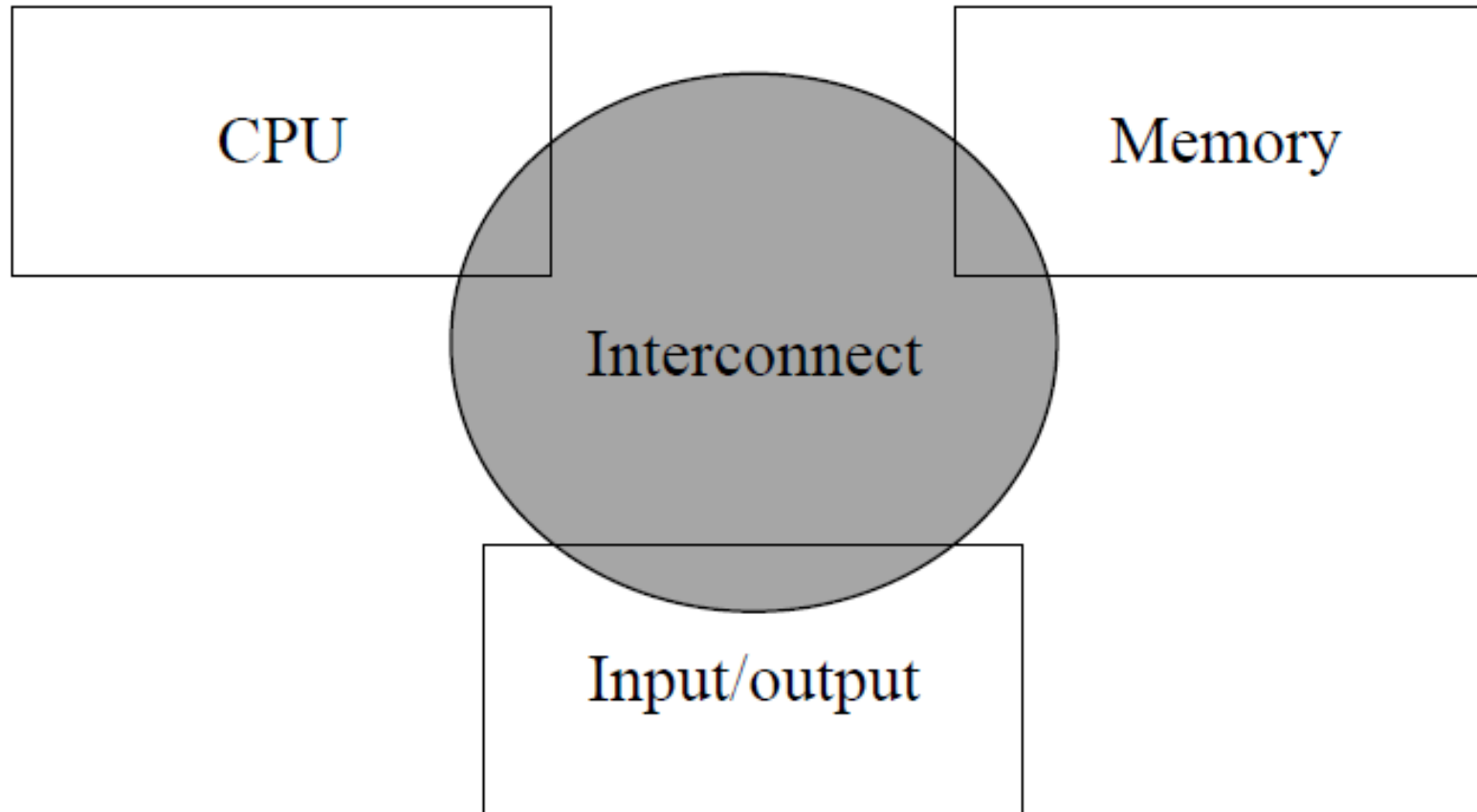
Увод у организацију и архитектуру рачунара 1

Јована Ковачевић

Основне компоненте

- Основне компоненте рачунарског система су:
 - централна јединица за обраду (процесор, *CPU*)
 - меморијска јединица (меморија)
 - улазно/излазни уређаји (*I/O*)
 - њихово повезивање

Основне компоненте



Магистрала

Карактеристике, типови, операције, ...

Магистрала

- Магистрала је подсистем који повезује компоненте рачунарског система
- Може да се састоји од појединачних делова као што су:
 - адресна магистрала
 - магистрала података
 - контролна магистрала

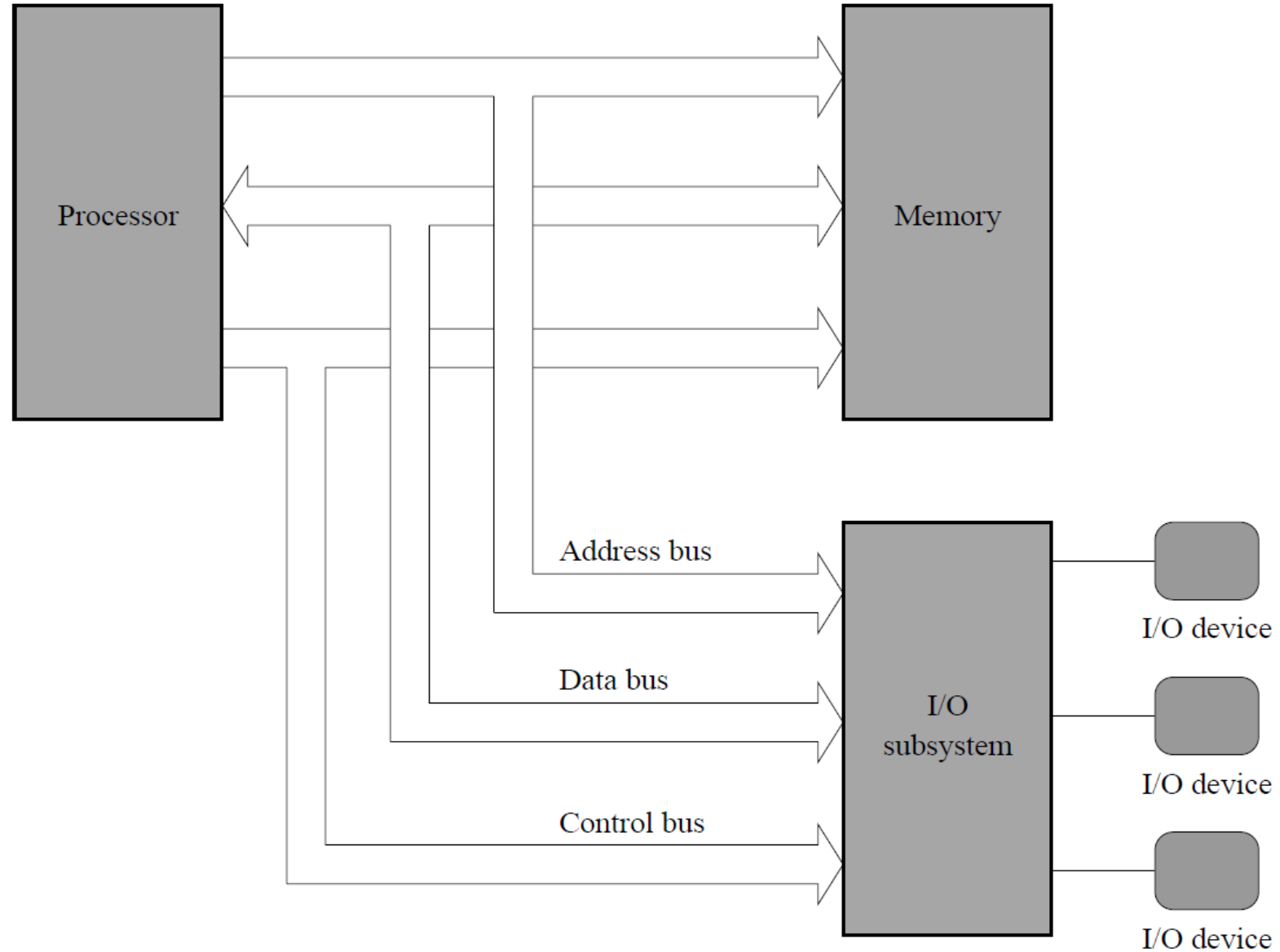
Магистрала (2)

- Адресна магистрала преноси податке о меморијским адресама
 - Њена ширина одређује величину адресног простора
- Магистрала података преноси податке
 - Њена ширина одређује величину података који се преносе
- Контролна магистрала преноси контролне сигнале (кодиране операције)

Системска магистрала

- Налази се унутар процесорског система
- Повезује процесорске јединице са меморијом и улазно/излазним подсистемом
- Употребљава се и термин *интерна (унутрашња) магистрала*

Системска магистрала



Спољашња (екстерна) магистрала

- Повезује уређаје који су ван процесорског система
 - *USB (Universal Serial Bus)*
 - *FireWire*

Дељење магистрале

- Магистрала је дељени ресурс
 - Свака компонента повезана магистралом је корисник магистрале
- При дељењу магистрале постоји могућност истовремених активности на магистрали
 - Истовремена употреба магистрале од стране више компоненти доводи до неисправности

Трансакције магистрале

- Трансакција магистрале (енгл. *bus transaction*) је целовит низ поступака на магистрали
- Примери активности:
 - читање из меморије
 - писање у меморију
 - читање са улазног уређаја
 - писање на улажном уређају
 - ...
- Једна трансакција може да обухвати више операција
 - нпр: агресивно читање (енгл. *burst read*)

Трансакције магистралне (2)

- У оквиру једне трансакције препознају се:
 - главни корисник (енгл. *master*) – започиње трансакцију
 - подређени корисник (енгл. *slave*) – одговара на захтев
- У једном тренутку највише једна трансакција на магистралаи
- Свака трансакција има тачно једног главног корисника
- Неки уређаји могу бити само подређени корисници магистралне
- Други уређаји могу бити главни или подређени (али не у исто време)

Врсте магистрала

- Магистрала може да буде
 - посвећена једној улози
 - нпр. адресна магистрала служи само за преношење адреса
 - има већу пропусност
 - сложенија за имплементацију
 - мултиплексирана магистрала
 - иста магистрала преноси адресе, податке и контролне
 - једноставнија за имплементацију
 - има нижу пропусност

Синхрона и асинхрона магистрала

- Магистрала може бити
 - Синхрона
 - Часовник обезбеђује синхронизацију свих операција
 - Асинхрона
 - Не користи се часовник за синхронизацију
 - Користе се операције руковања и додатни синхронизациони сигнали

Карактеристике магистрале

- Ширина магистрале
 - Односи се на магистрале адресе и података
- Тип магистрале
 - Посвећена или мултиплексирана
- Операције магистрале
 - читање, писање, пренос блокова, читање са мењањем (енгл. *read-modify-write*) и прекиди
- Арбитража
 - може да буде централизована и дистрибуирана
- Подешавање времена
 - може бити синхроно или асинхроно

Ширина магистрале података

- Ширина магистрале података одређује величину података који се преносе магистралом
- Пропусност магистрале представља брзину преношења података и може се мерити бројем пренетих битова у јединици времена или у циклусу часовника
- Ширина магистрала значи већу пропусност али и већу цену због компликованије имплементације
 - На пример, код 32-битног Pentium-а ширина магистрале података је 64 бита
 - Са друге стране, IBM 8086 и јефтинији 8088 су 16-битни али је код 8086 адресна магистрала 16-битна а код 8088 8-битна. То значи да 8086 пропушта 16-битни податак у једном циклусу а 8088 у два

Ширина адресне магистрале

- Ширина адресне магистрале одређује величину адресног простора
 - Ако је ширина магистрале n адресних линија, број адресибилних локација је 2^n
 - Једна адресибилна локација садржи једну меморијску реч
 - Меморијска реч је обично величине 1 бајт, али не мора бити тако
- Основна мотивација за проширивање је повећавање адресног простора
- Основна мотивација за сужавање је смањивање сложености и смањивање трошкова

Особине асинхроне магистрале

- Ослобођене су везивања за часовник
- Флексибилније у погледу трајања операција
 - операције не морају да трају цео број циклуса
 - на пример, ако је меморија мало спорија, код синхроних магистрала се додаје цео циклус чекања, а код асинхроних то може бити и краће време
- Флексибилније су у погледу уређаја
 - брзина рада се прилагођава брзини уређаја
 - за синхроне м. је веома важно добро одређивање брзине часовника, па се захтева одређена хомогеност уређаја
- Сложеније су за имплементацију

Синхроне и асинхроне магистрале

- Системске магистрале су по правилу синхроне
 - делом из историјских разлога
 - раније разлике у брзини рада уређаја нису биле велике као што су данас
 - делом због једноставности
- Пројектанти се чешће одлучују да примене више различитих синхроних магистрала него једну асинхрону
 - на пример, посебне магистрале за меморију, кеш и друге спољне уређаје

Арбитража магистрале

- Магистрале које могу имати више потенцијалних главних уређаја морају имати механизам *арбитраже*
- Механизам арбитраже служи за додељивање магистрале главном уређају
- На системској магистрали је главни уређај најчешће процесор, али то може бити и контролер *DMA*

Меморија

Карактеристике и типови

Меморија

- Меморија је уређај који омогућава чување (записивање) и читање података у рачунару

Основне карактеристике

- Трајање записа
- Тип носиоца
- Капацитет
- Јединица преноса
- Адресибилност
- Цена
- Могући начини приступа
- Перформансе
- Могућност промене садржаја

Трајање записа

- Меморије са сталним записом
- Меморије са привременим записом

Тип носиоца

- Полупроводничке
- Са магнетном површином
- Оптичке

Капацитет

- Капацитет се изражава у бајтовима или речима (адресабилна меморијска локација)
- Уобичајене дужине речи
 - 8, 16, 32, 64, 128 битова
 - 1, 2, 4, 8, 16 бајтова
- Капацитет уобичајено у бајтовима
 - KiB, MiB, GiB, TiB
 - $1\text{KiB} = 1024\text{B}$, $1\text{MiB} = 1024^2\text{B}$, $1\text{GiB} = 1024^3\text{B}$, $1\text{TiB} = 1024^4\text{B}$
 - KB, MB, GB, TB
 - $1\text{KB} = 10^3\text{B}$, $1\text{MB} = 10^6\text{B}$, $1\text{GB} = 10^9\text{B}$, $1\text{TB} = 10^{12}\text{B}$

Јединица преноса

- За унутрашње меморије јединица преноса је обично *реч*
 - 1,2,4,8,16 бајтова
- За спољашње меморије јединица преноса је обично *блок*
 - од 512В до неколико МВ

Адресибилност

- Адресибилна меморија
 - ако се може адресирати свака појединачна меморијска локација (реч)
- Полуадресибилна меморија
 - ако се адресом приступа групи бајтова, која је већа од речи
- Неадресибилна меморија
 - ако се садржају не приступа путем адресе

Цена

- Цена меморије се пореди у односу на одређен капацитет и перформансе

Могући начини приступа

- Секвенцијалан приступ
- Непосредан приступ
- Произвољан приступ
- Асоцијативан приступ

Могући начини приступа

- **Секвенцијалан приступ**

- подаци су организовани у јединице – *слогове*
- слогови се међусобно раздвајају контролним информацијама
- пише се редом
- чита се редом, како је вршено писање
- пример: магнетна трака

- Непосредан приступ

- Произвољан приступ

- Асоцијативан приступ

Могући начини приступа

- Секвенцијалан приступ
- **Непосредан приступ**
 - постоји зависност адресе слога и његове физичке локације (не мора да буде пуна)
 - на основу адресе се приступа непосредно слогу или његовој околини
 - време за приступ није фиксно
 - пример: магнетни диск
- Произвољан приступ
- Асоцијативан приступ

Могући начини приступа

- Секвенцијалан приступ
- Непосредан приступ
- **Произвољан приступ**
 - свака адресибилна локација има механизам приступа подацима
 - време за приступ је фиксно
 - пример: главна меморија рачунара
- Асоцијативан приступ

Могући начини приступа

- Секвенцијалан приступ
- Непосредан приступ
- Произвољан приступ
- **Асоцијативан приступ**
 - подврста меморије са произвољним приступом
 - податку се приступа на основу неког узорка (маске) адресе или податка
 - пример: кеш меморија

Перформансе

- Време приступа
 - трајање операције читања или писања
 - од неколико *ns* до неколико *ms*
- Трајање временског циклуса
 - обухвата време приступа
 - и додатно време за ослобађање магистрале и припрему за наредну операцију
- Брзина преноса
 - време за које већа количина података може да се прочита или упише
 - узима у обзир време приступа али и архитектуру рачунарског система (нпр. преплитање)

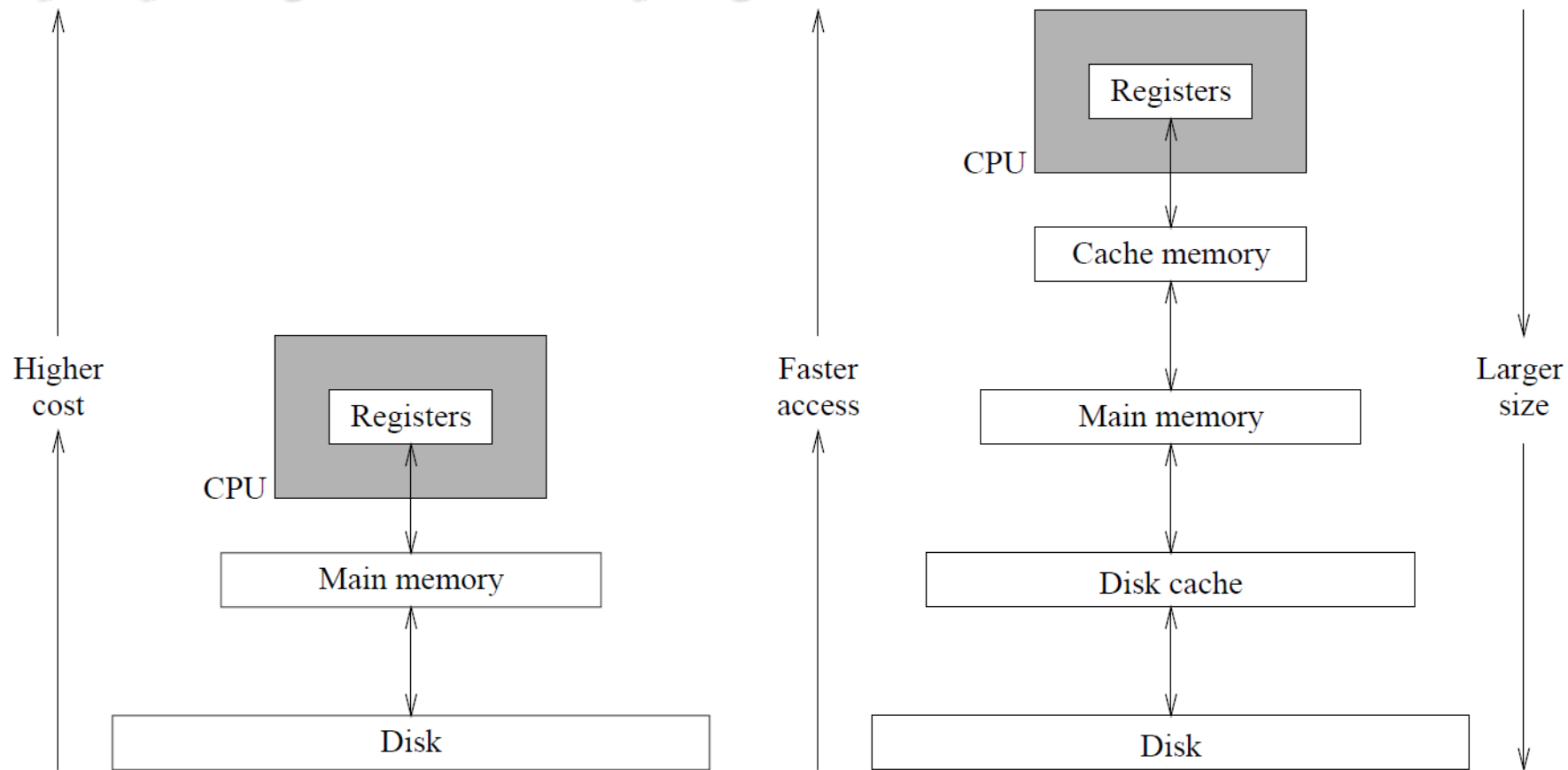
Могућност промене садржаја

- Меморија само за читање
- Меморија за читање и писање

Хијерархија меморија

- За меморију по правилу важи:
 - што је краће време приступа, цена је већа
 - што је већи капацитет, време приступа је дуже
 - што је већи капацитет, цена по биту је нижа
 - нове технологије доносе нижу цену по биту уз очување претходних односа

Хијерархија меморија



Основне врсте меморије

- Два основна типа меморија:
 - меморије само за читање
 - (*read-only memory*)
 - *ROM*
 - меморије за читање и писање
 - (*read-write memory*)
 - *RAM*
 - назив (*random access memory*) је неисправан, зато што и *ROM* и *RAM* омогућавају произвољан приступ

ROM

- *ROM*
 - меморија само за читање
 - (*read-only memory*)
 - не захтева напајање за одржавање садржаја
 - могу чувати податке док је рачунар искључен
 - обично се употребљава за подизање рачунарског система (*boot*)

Врсте ROM-а

- Фабрички програмиран
 - прави се у случају масовне потребе
- Програмибилан
 - *PROM (programmable ROM)*
 - на пример, корисник може да спаљује осигураче по избору
- Вишекратно програмибилан
 - *EPROM (erasable programmable ROM)*
 - излагањем ултраљубичастом светлу се брише садржај *EPROM*-а
- Вишекратно програмибилан са ел. брисањем
 - *EEPROM (electrically erasable programmable ROM)*
 - омогућава да се селективно брише садржај

RAM

- Две основне врсте *RAM* меморија су
 - статички *RAM* и
 - динамички *RAM*

Статички RAM

- *SRAM*
- Имплементира се помоћу резе или флип-флопа
- Не захтева освежавање да би чувао садржај
- Предности:
 - Једноставност употребе
 - Брзина
- Употребљава се за кеш меморије

Динамички RAM

- *DRAM*
- Имплементира се помоћу малих кондензатора
- Захтева периодично освежавање да би чувао садржај
- Читање нарушава садржај
 - неопходно *писање-после-читања*
- Предности
 - Нижа цена
 - Мање загревање
 - Већа густина паковања
- Употребљава се за радну меморију рачунара

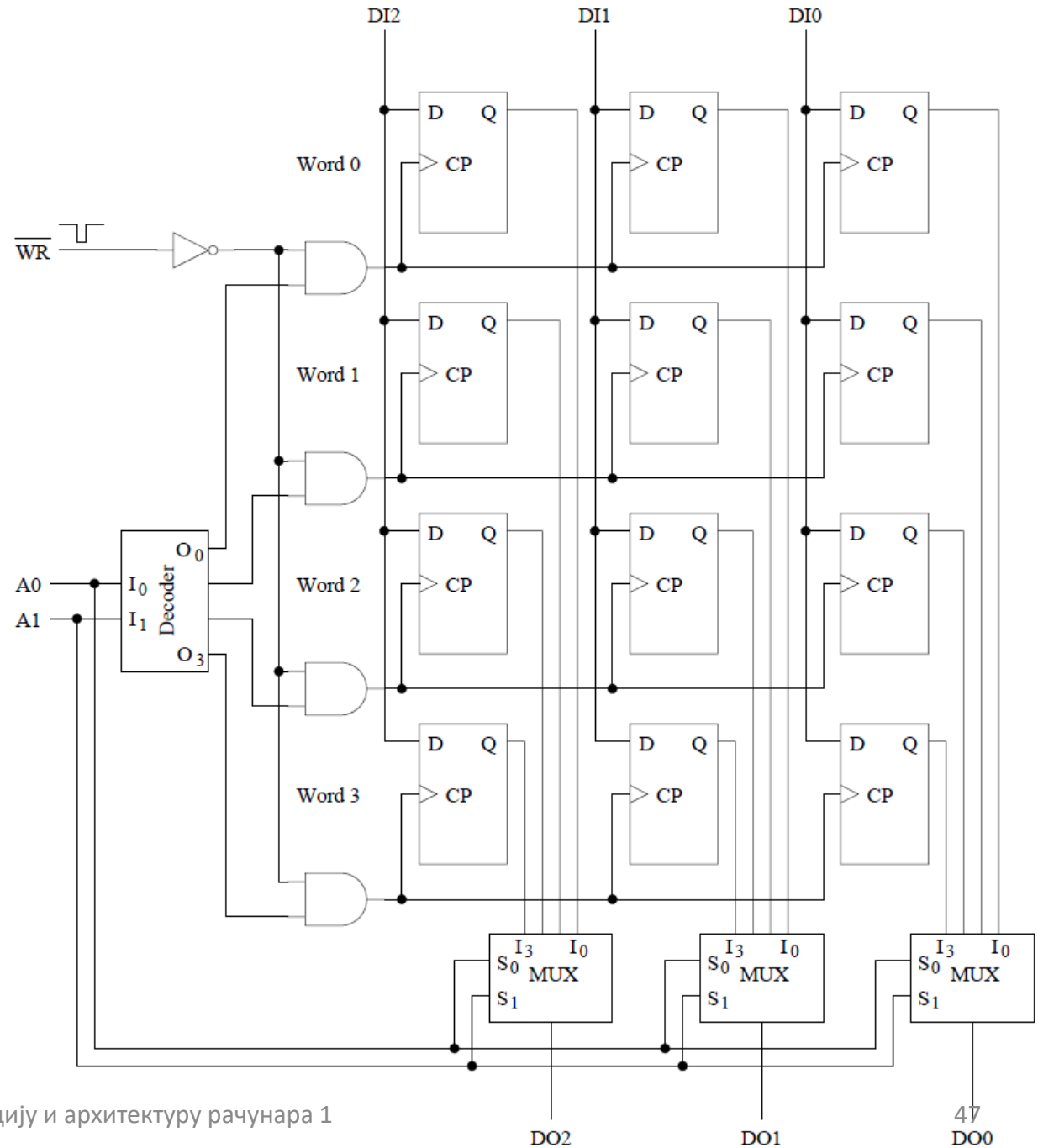
Меморија

Архитектура

Меморија од D флип-флопова

- Употребљава се дводимензиони низ D флип-флопова
 - сваки ред чува једну реч
 - број колона одговара броју битова у речи
 - *хоризонтално ширење* је повећавање број битова у речима
 - број редова одговара броју речи у меморији
 - *вертикално ширење* је повећавање броја речи
 - оба броја су обично неки степени броја 2
 - меморија $M \times N$ има M речи од по N битова

Меморија имплементирана матрицом 4x3 D флип-флопа



Меморија од D флип-флопова (2)

- Декодер одређује тачно један ред на основу улазне адресе
 - адреса је кодирана са две линије
 - декодер са И-елементима гради демултиплексор који усмерава сигнал на одговарајући ред
- Активан сигнал часовника ће добити само изабрани ред и то само у случају писања
- Сви флип-флопови у једној колони добијају исти улазни сигнал
- За читање се употребљава 4-1 мултиплексор
 - адресне линије се користе као селектори

Ограничења и проблеми

- Није прилагођена повезивању на магистралу
 - потребно је да исте линије носе улазне и излазне податке
- Не може да се користи за прављење већих меморија
 - Потребан је додатни селекторски улаз који означава да ли се блок користи или не

Повезивање на магистралу

- Потребно је узети у обзир
 - Исте линије се користе за улаз и за излаз
 - трансфер је двосмеран кроз исте линије
 - не могу се користити различите линије за улаз и излаз
 - Меморијска магистрала је дељена
 - само изабрани уређај сме стављати податке на магистралу података
 - остали се морају понашати као да нису повезани на магистралу
- Постоји више техника које се користе, а ми ћемо обрадити две:
 - Мултиплексоре
 - Бафере са три стања

Употреба мултиплексора

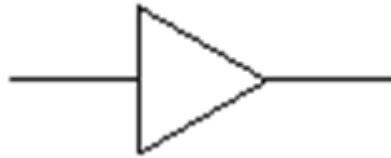
- Представљен пример почива на мултиплексорима
- Мултиплексори су проблематични јер:
 1. Не могу да се непосредно повежу улази и излази података!
 2. Не омогућавају прављење већих меморија, од више меморијских чипова, јер је потребно омогућити везу са CE (chip enable) сигналом!

Употреба бафера са три стања

- Уређаји са три стања имају 3 а не само 2 стања (за разлику од осталих представљаних реза и флип-флопова)
 - Имају додатни контролни сигнал
 - Ако је он активан, излаз је са високом импеданцом (активан) независно од улаза
- Данас уобичајено решење

Бафер

- Најпре да размотримо обичан “бафер”, који наизглед не служи ничему



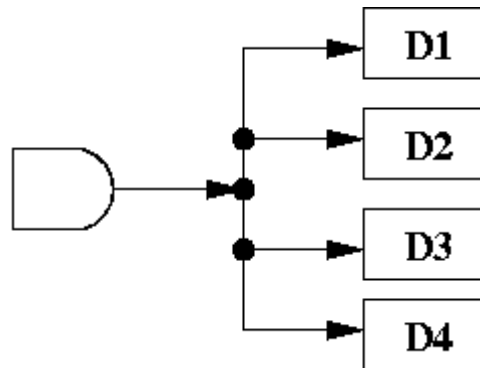
- Међутим, он има функцију
 - Бафер је *активан* елемент

Бафер (2)

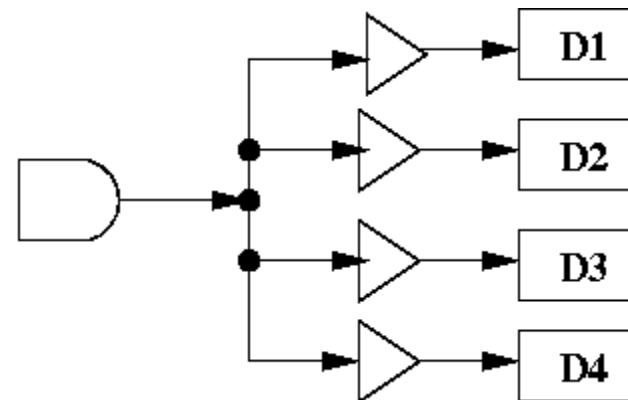
- Бафер је *активан* елемент
 - ако је на улазу *позитиван* потенцијал, он обезбеђује да је на излазу *пун* позитиван потенцијал
 - ако је на улазу *приближно нулти* потенцијал, он обезбеђује да је на излазу *нулти* потенцијал
- Понаша се као појачавач сигнала
 - сваки потенцијал *изнад* прага функционисања транзистора појачава се до пуног интензитета позитивног потенцијала
 - сваки потенцијал *испод* прага функционисања транзистора “појачава” се до нултог потенцијала

Бафер (3)

- У овом примеру, сваки од елемената $D1-D4$ ће добити свега по $\frac{1}{4}$ потребног потенцијала



- Употреба бафера обезбеђује да елементи $D1-D4$ добију пун потребан потенцијал

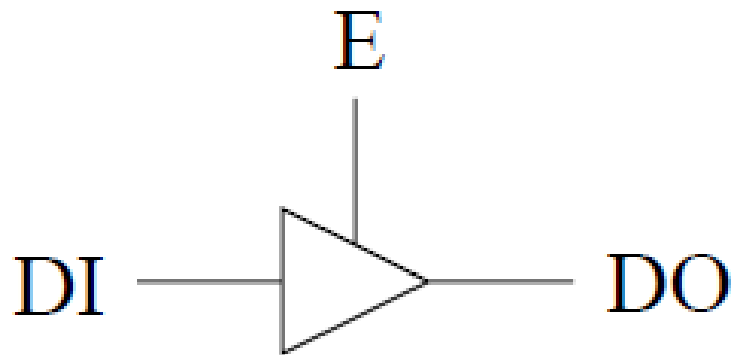


Три стања?

- До сада смо подразумевали да сваки елемент има за излаз једну од две вредности: 0 или 1
- У логичком систему то и јесте тако
- У имплементираним системима имамо потребу за трећим стањем:
 - 0 – на излазу се поставља нулти потенцијал (уземљење) и омогућава проток струје
 - 1 – на излазу се поставља позитиван потенцијал (напајање) и омогућава проток струје
 - **Z – не утиче се на стање на излазу и онемогућава се проток струје**

Бафер са три стања

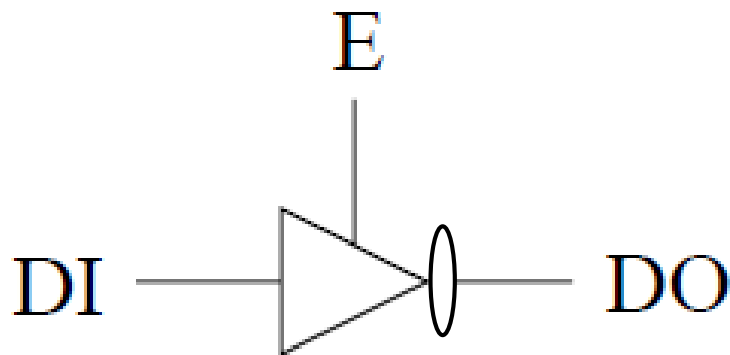
- Бафер са три стања се понаша попут **вентила**
- Ако се на “вентил” E који допушта проток (енгл. *enable*) доведе 0, онда се не утиче на стање на излазу
- Ако се на “вентил” E који допушта проток доведе 1, онда се сигнал са улаза X пропагира на излаз



Inputs		Output
E	DI	DO
1	0	0
1	1	1
0	X	Z

Инвертор са три стања

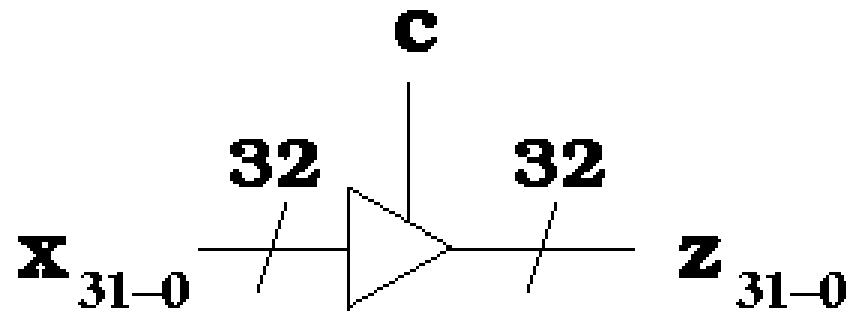
- Инвертор са три стања се понаша као негација са вентилом
- Ако се на “вентил” E који допушта проток (енгл. *enable*) доведе 0, онда се не утиче на стање на излазу
- Ако се на “вентил” E који допушта проток доведе 1, онда се сигнал са улаза X инвертује и пропагира на излаз



Inputs		Output
E	DI	DO
1	0	1
1	1	0
0	X	Z

Бафери са три стања и магистрале

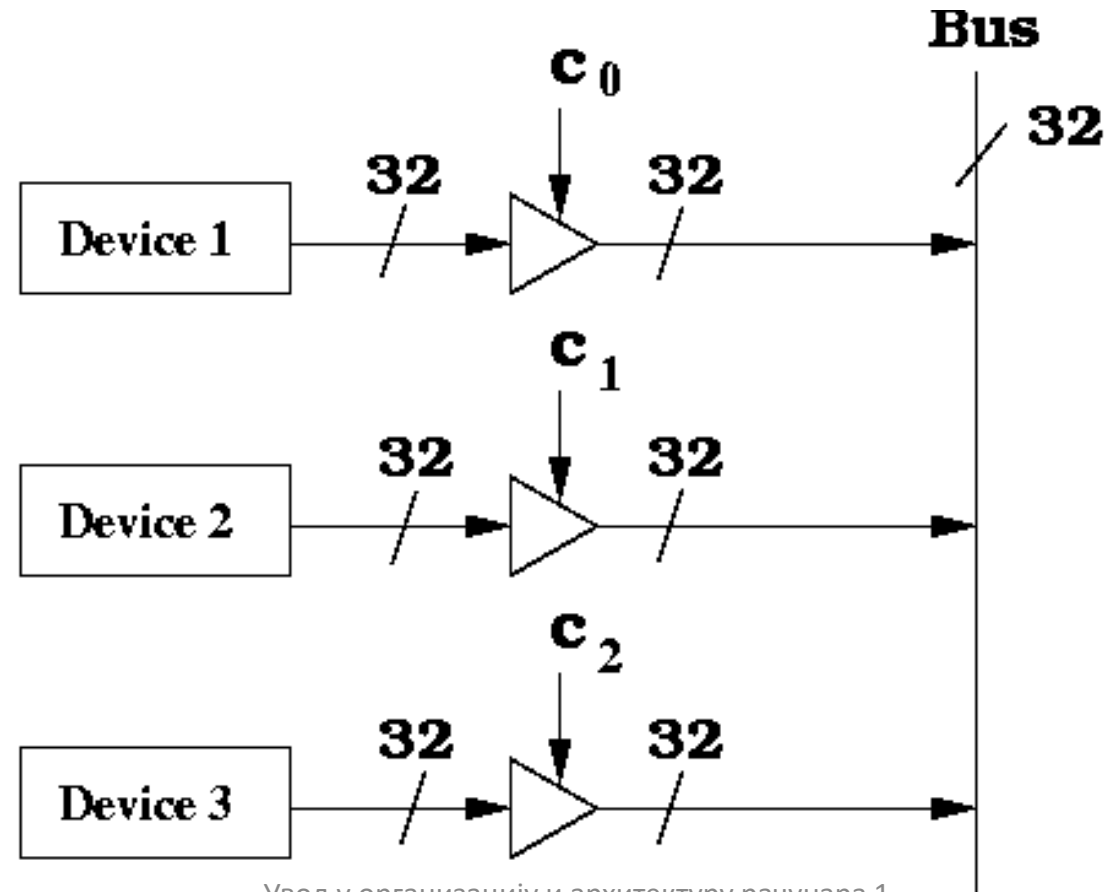
- На магистралу се преко бафера са три стања обично ставља истовремено већи број битова
 - у зависности од ширине магистрале
- Ради једноставнијег представљања често се користи поједностављен симбол за представљање низа бафера са три стања:



**tri-state buffer with
a bus of 32 bits**

Бафери са три стања и магистрале (2)

- Пример везивања уређаја на магистралу
 - ради се о излазним подацима уређаја

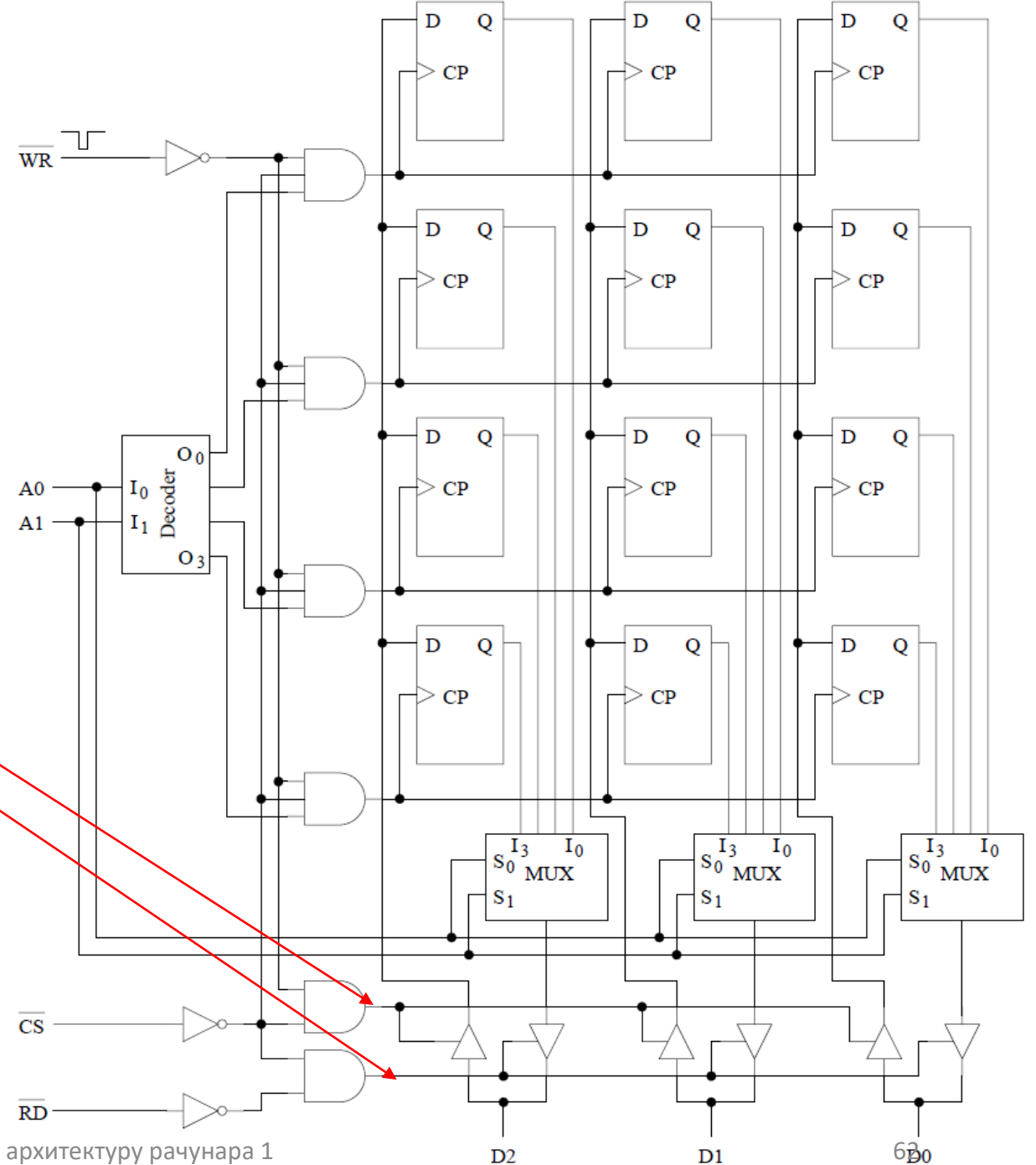


Меморијски блок

- Потребне промене у односу на претходни дизајн:
 - Додајемо селекторски улаз
 - улазни сигнал који одређује да ли се меморијски блок употребљава или не
 - повезује се као улаз на конјункције за избор адресе
 - Спајамо улазне и излазне сигнале података
 - ако се пише, онда помоћу бафера са три стања усмеравамо податке са магистрале на улазе флип-флопова
 - ако се чита, онда помоћу бафера са три стања усмеравамо излазе из флип-флопова на магистралу

Меморија 4x3 D флип-флопа употребом бафера са три стања

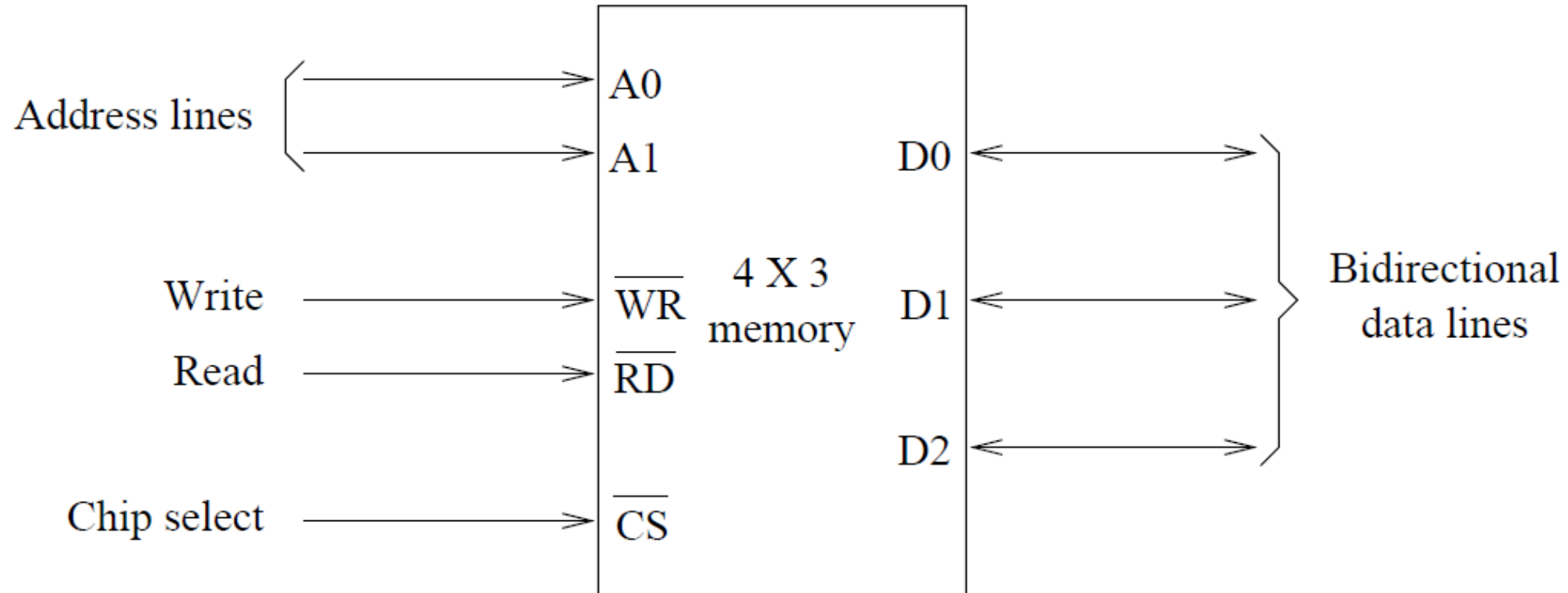
- CS' у конјукцији са WR' или RD' се такође везује као вентил за бафер са три стања
- Ако чип није одабран, излаз ка магистралаи ће бити у стању високе импеданце, па неће реметити рад других уређаја на магистралаи
- Кад су $CS'=0$ и $WR'=0$, биће активан смер писања
- Када су $CS'=0$ и $RD'=0$, биће активан смер читања



Меморијски блок (2)

- Да резимирамо:
- Потребан је додатни контролни сигнал који означава операцију читања
- Селекторски улаз укључује/искључује контролне сигнале за читање и писање
- Баферима са три стања се
 - сигнал са магистрале података пропушта до улаза на флип-флопове, ако су активни и селекторски сигнал и контролни сигнал операције читања
 - сигнал са излаза флип-флопова се пропушта на магистралу података, ако су активни и селекторски сигнал и контролни сигнал операције писања
 - искључени бафер (контролни сигнал 0) има високу импеданцу и не представља сметњу функционисању укључених бафера са истим излазом/улазом

Блок дијаграм меморије 4x3



Прављење већих меморија

- Од меморијских блокова који имају контролне селекторе (CS) могу се правити већи меморијски блокови
- Први корак је прављење независне меморијске јединице која није чврсто везана за специфичне адресе у адресном простору
 - Нешто као већа верзија претходно представљеног меморијског блока
- Други корак је везивање оваквих независних меморијских јединица за конкретан адресни простор

Примери меморијских чипова

- Постоји велики број чипова који се употребљавају за израду већих меморија
 - примери *SRAM* и *DRAM* чипова фирме *Micron*
- *SRAM*
 - 8Mb чип, у три конфигурације:
 - 512K x 18
 - 256K x 32
 - 256K x 36
 - додатни битови служе за препознавање и отклањање грешака
 - време приступа 3.5ns
 - чип 512K x 18 има 19 адресних линија
 - чипови 256K x 32 и 256K x 36 имају по 18 адресних линија

Примери меморијских чипова (2)

- *DRAM*
 - синхрони *DRAM*
 - 256Mb чип, у три конфигурације:
 - 64M x 4, 26 адресних линија
 - 32M x 8, 25 адресних линија
 - 16M x 16, 24 адресне линије
 - трајање циклуса је око $7ns$

Организација чипова

- Иста количина меморије се може различито организовати и имплементирати
 - у време уских магистрала (8-16 битова) *DRAM* се израђивао у ширини 1 бита
 - данас то није практично због велике ширине речи
- Предност широких чипова је што их је потребно мање за веће меморије
 - *Pentium* је 32-битни процесор са 64-битном магистралом података
 - меморија 16М x 64 може се направити
 - од једног реда са 4 чипа 16М x 16
 - али не и од 8 чипова 32М x 8 (добијамо 32М x 64)
 - од уских чипова се ни не могу добити неке мање меморијске јединице

Дизајн већих меморија

- На примеру *DRAM*-а
- Основно питање је да ли је меморијски адресни простор (*memory address space – MAS*) адресибилан на нивоу појединачних бајтова (бајт-адресабилан) или не
 - Већина савремених процесора подржава адресирање на нивоу појединачних бајтова (једна адреса се односи на један бајт)
 - У наредним примерима ћемо претпостављати да је меморијски адресни простор бајт-адресабилан

Дизајн већих меморија (2)

- Затим се одлучује о конфигурацији чипова
 - при изабраној циљној величини и величини чипова, не мења се укупан број чипова, али се мења њихов распоред
 - ако је циљ меморија $M \times N$, а користе се чипови $D \times W$:
 - број колона је N/W
 - број редова је M/D
 - број чипова је $(M \times N)/(D \times W)$
- У примеру
 - правимо меморију од 256Mib
 - циљна конфигурација је $64\text{Mi} \times 32\text{b}$
 - користимо чипове $16\text{Mi} \times 16\text{b}$
 - матрица чипова је 4×2

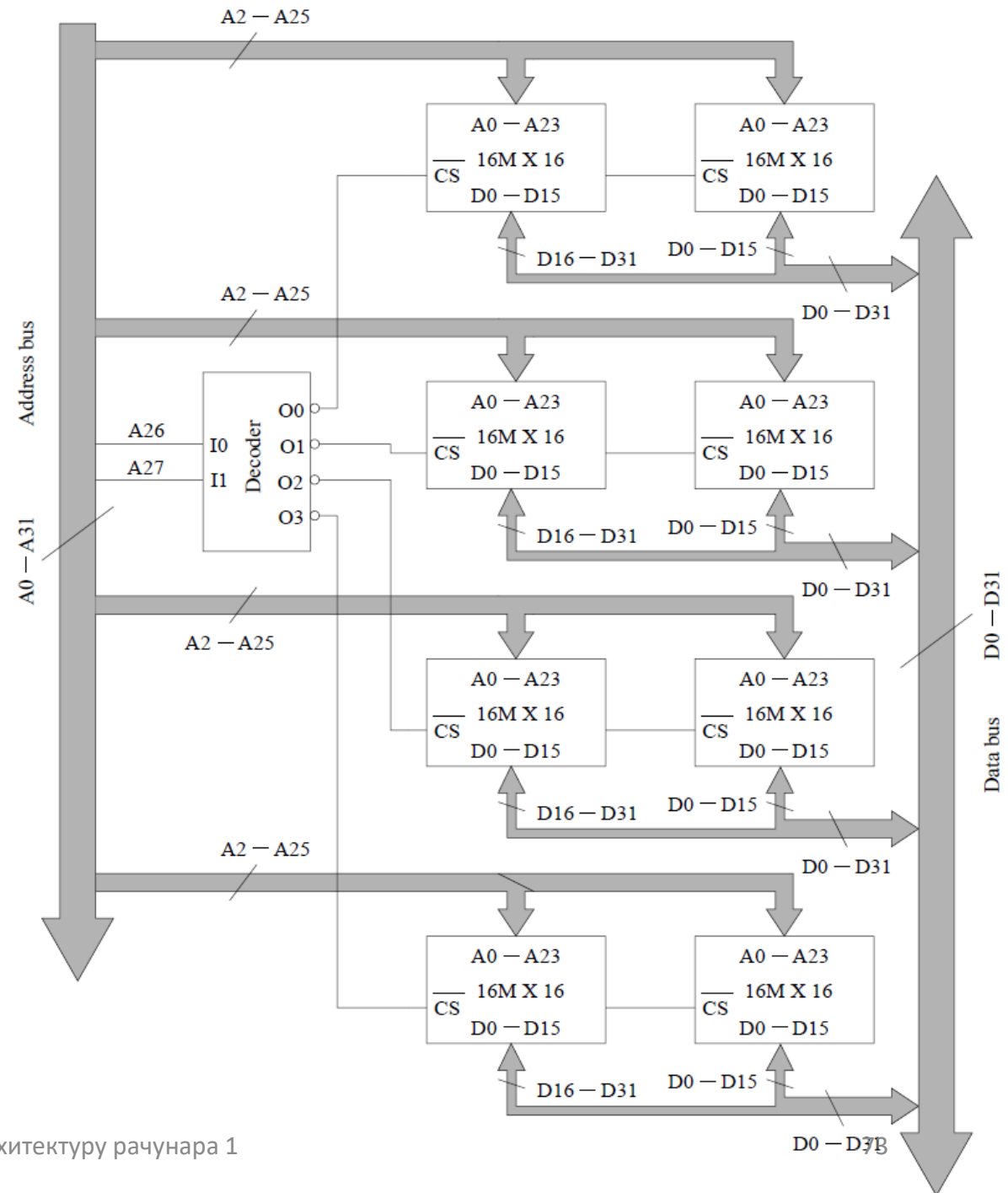
Дизајн већих меморија (3)

- Везивање на магистралу података је непосредно
 - сваком чипу у реду одговара део линија података
- Ако се у једном кораку чита N битова (>8)
 - Z најнижих битови адресе се игноришу
 - $Z = \log_2(N/8)$
- У примеру
 - 256М се адресира са 28 битова адресе
 - $256\text{MB} = 256 \times 2^{20} \text{B} = 2^8 * 2^{20} \text{B} \Rightarrow 2^{28}$
 - један чип има 24 бита адресе (1 чип $16\text{Mi} \times 16\text{b}$)
 - $16\text{Mi} = 16 \times 2^{20} = 2^4 * 2^{20} = 2^{24}$
 - најнижа 2 бита адресе A_0, A_1 се не користе
 - ширина меморије је 32 бита (4 бајта), а адресирање по бајтовима
 - на чипове се везују 24 бита A_2 до A_{25}
 - битови адресе A_{26} и A_{27} се користе за бирање реда чипова

Дизајн већих меморија (4)

- Контролни сигнали за читање и писање свих чипова се повезују међусобно и на контролну магистралу
 - (изостављено из наредног дијаграма ради једноставности)

Дизајн меморије 64М x 32 од чипова 16М x 16



- Слајдови су направљени на основу материјала проф. Саше Малкова и проф. Александра Картеља