

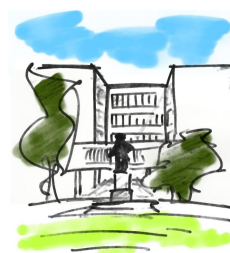
[P220]
Увод у архитектуру
рачунара

5



Саша Малков
Универзитет у Београду
Математички факултет
2013/2014

[P271]
Увод у архитектуру рачунара
Саша Малков



Тема 7
Магистрала
(наставак)

Тип магистрале



- Ширине магистрала утичу на трошкове
 - ако 64-битни процесор има ширине магистрала података и адреса од по 64 бита, онда му је потребно 128 линија (па и пинова) само за ове магистрале
 - ако се користи 128 линија података, онда је то чак 192
- Ради смањивања трошкова се уместо посвећених магистрала података и адреса може употребљавати мултиплексирана магистрала
 - назива се *магистрала адреса и података* (енгл. *AD-bus*)

Мултиплексиране магистрале



- Пример рада – читање из меморије:
 - процесор најпре ставља на магистралу адресу
 - меморијска јединица чита адресу и приступа локацији
 - у међувремену процесор уклања адресу са магистрале
 - меморијска јединица на магистралу поставља прочитан податак
- Пример рада – писање у меморију:
 - процесор најпре ставља на магистралу адресу
 - меморијска јединица чита адресу и приступа локацији
 - процесор уклања адресу са магистрале и поставља податак
 - меморијска јединица чита податак и уписује га у меморији

Мултиплексиране магистрале (2)



- Мултиплексирањем се смањује ефикасност магистрале
 - операције се успоравају због повећавања броја корака
 - (не у случају свих операција)
- Пример мултиплексиране магистрале је *PCI*

Операције магистрале



- Неке од основних операција су
 - читање/писање података меморијеУ/И уређаја
 - пренос блокова података
 - пример: попуњавање кеша процесора
 - читање са мењањем (енгл. *read-modify-write*)
 - корисно у случају више процесорских система
 - атомичном операцијом се обезбеђује искључиви приступ критичним секцијама меморије (катанци и сл.)
 - прекиди
 - прекидима се процесор усмерава на услуге неопходне за подршку улазно/излазних уређаја
 - и друге операције

Синхрона магистрала



- Код синхроне магистрале часовник обезбеђује синхронизацију свих поступака на магистрали
- Промене других сигнала се одвијају у односу на узлазне и силазне рубове часовника

Операције синхроне магистрале



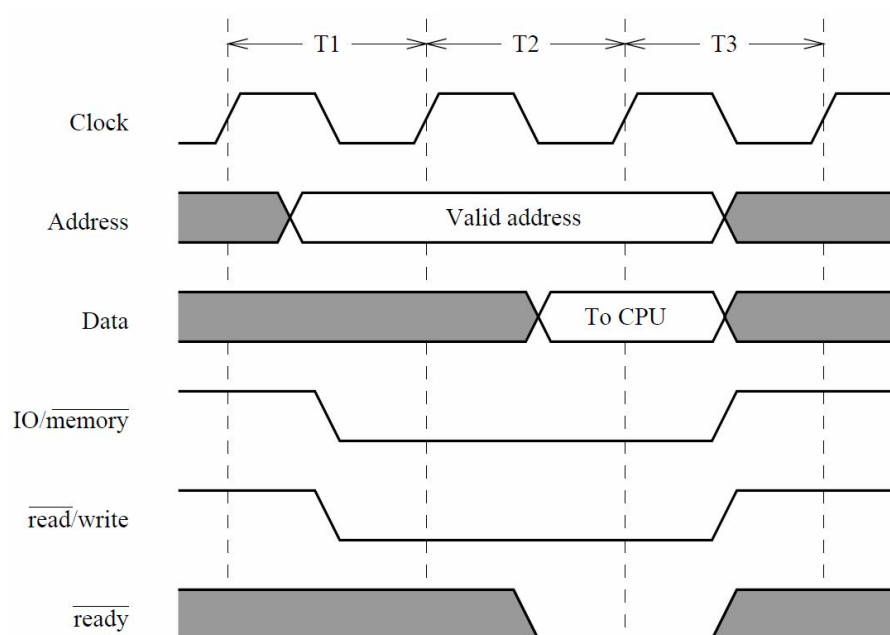
- Разматраћемо начин имплементације операција са меморијом на синхроној магистрали
 - по узору на читање из меморије и писање у меморију код процесора *Intel Pentium*

Операција читања (1)



- Операција читања из меморије се састоји од три основна поступка:
 - процесор поставља захтев за читање
 - меморија извршава операцију читања
 - процесор преузима прочитане податке

Операција читања из меморије без додатних стања чекања



Операција читања (2)



- У циклусу T1 процесор поставља захтев за читање
 - током активног стања циклуса T1 процесор поставља на адресну магистралу исправну адресу меморијске локације са које је потребно читати
 - након тога процесор поставља два контролна сигнала ради идентификовања врсте операције:
 - сигнал "IO/memory" се поставља на ниско стање, што означава меморијску операцију
 - сигнал "read/write" се поставља на ниско стање, што означава операцију читања

Операција читања (3)



- Меморија извршава операцију читања од тренутка постављања контролног сигнала
 - меморија чита адресу са адресне магистрале
 - поставља на магистралу података прочитану вредност
- Меморија завршава операцију читања најраније на силазном рубу циклуса T2
 - ако је операција извршена, поставља ниско стање сигнала "ready"
 - ако је меморија спорија, она означава да операција још није извршена одржавањем активног стања сигнала "ready" све док не постави прочитане податке
 - (потенцијално и више циклуса)

Операција читања (4)

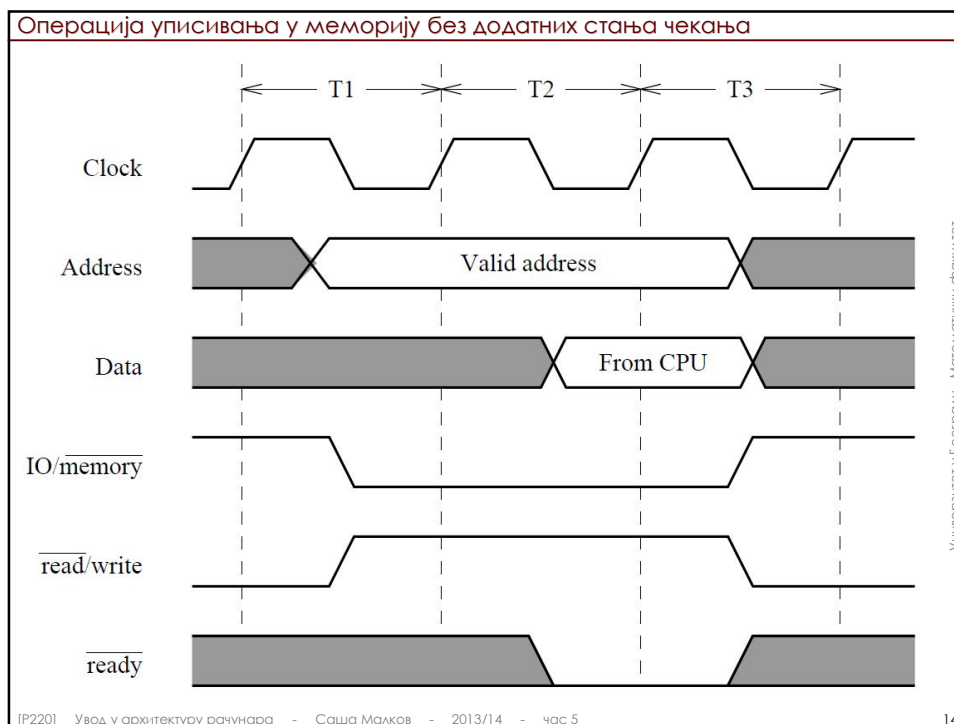


- Процесор преузима прочитане податке
 - почев од неактивног стања циклуса T2 процесор проверава сигнал “ready”
 - ниско стање означава да су подаци прочитани и спремни за преузимање
 - активно стање означава да меморија захтева додатно време (бар још један циклус) да би поставила прочитане податке на магистралу
 - ако су подаци присутни (ниско стање сигнала “ready”)
 - процесор чита податке са магистрале података
 - склања адресу са адресне магистрале
 - деактивира контролне сигнале “IO/memory” и “read/write”
 - операција је завршена најраније на силазном рубу циклуса T3

Операција уписивања (1)



- Слично операцији читања
- Операција уписивања у меморију се састоји од три основна поступка:
 - процесор поставља захтев за писање
 - меморија извршава операцију писања
 - по потврђеном писању, процесор наставља рад



Операција уписивања (2)



- У циклусу T1 процесор поставља захтев за читање
 - током активног стања циклуса T1 процесор поставља на адресну магистралу исправну адресу меморијске локације са које је потребно читати
 - затим поставља два контролна сигнала ради идентификовања врсте операције:
 - сигнал "IO/memory" се поставља на ниско стање, што означава меморијску операцију
 - сигнал "read/write" се поставља на активно стање, што означава операцију писања
 - касније, током циклуса T2, процесор поставља податке на магистралу података

Операција уписивања (3)



- Меморија извршава операцију писања почев од силазног руба циклуса T1
 - меморија чита адресу са адресне магистрале и обавља припреме за уписивање
 - затим чита податке са магистрале података и уписује их на одговарајућој локацији
- Меморија ослобађа магистралу најраније на силазном рубу циклуса T2
 - ако је операција писања извршена (или ће бити извршена током циклуса), поставља ниско стање сигнала "ready" "
 - ако је меморија спорија, она означава да операција још није извршена одржавањем активног стања сигнала "ready" " све док не постави прочитане податке
 - (потенцијално и више циклуса)

Операција уписивања (4)



- Процесор прима потврду о уписивању
 - почев од неактивног стања циклуса T2 (одмах по постављању података) процесор проверава сигнал "ready" "
 - ниско стање означава да ће меморија током тог циклуса завршити уписивање
 - активно стање означава да меморија захтева додатно време (бар још један циклус) да би уписала податке
 - ако је операција извршена (ниско стање сигнала "ready" ")
 - процесор склања адресу са адресне магистрале
 - деактивира контролне сигнале "IO/memory" "и "read/write"
- операција је завршена најраније на силазном рубу циклуса T3

Рад са У/И уређајима

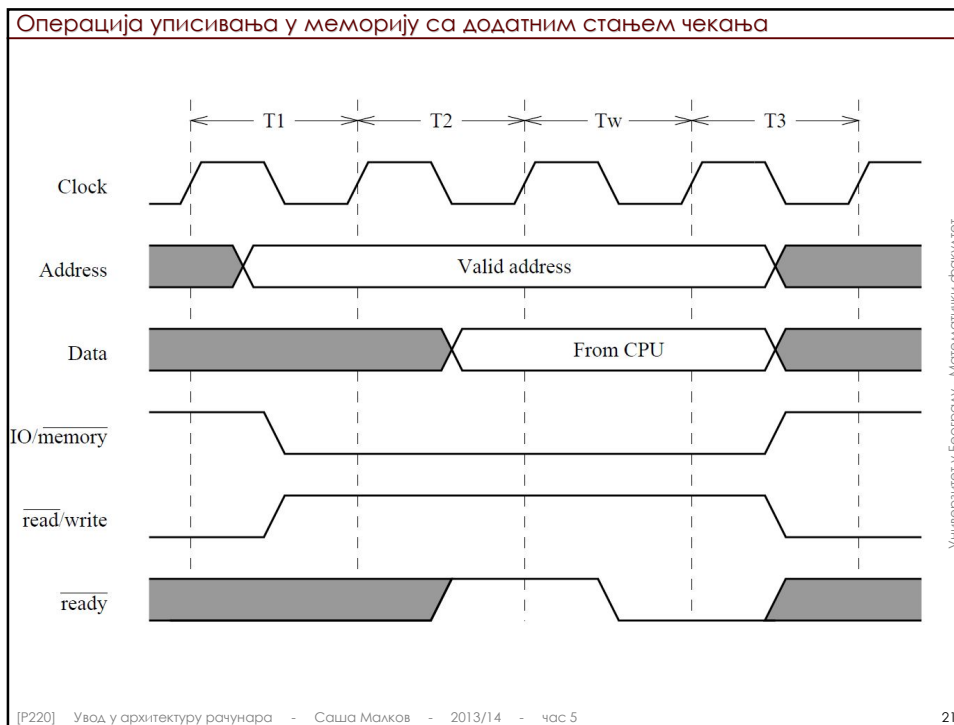
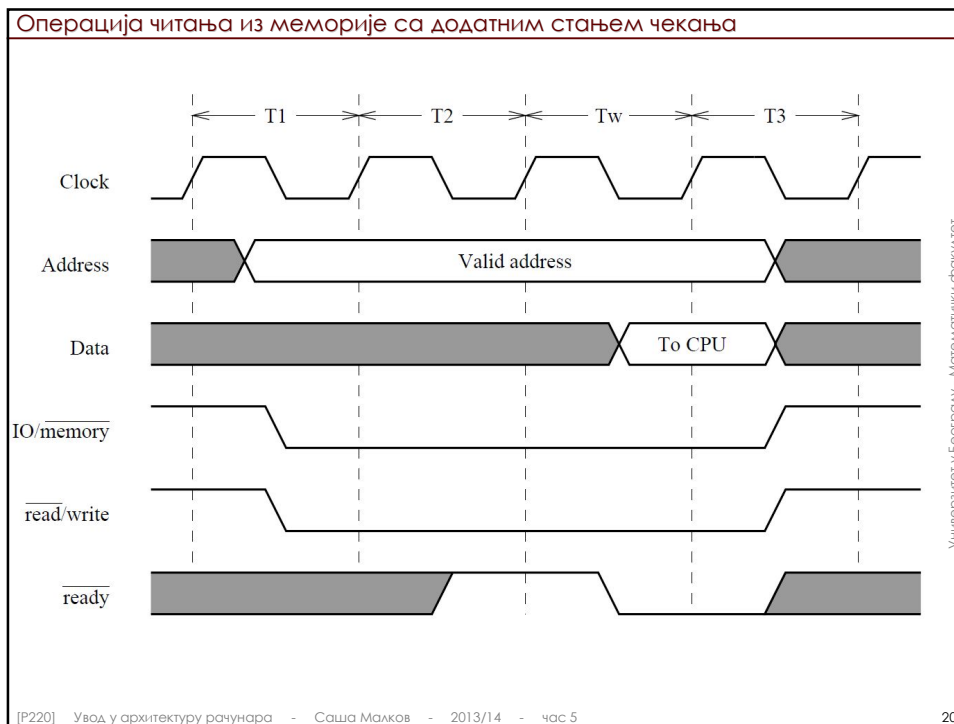


- Читање и писање на У/И уређајима је сасвим слично раду са меморијом
 - ако уређај ради путем пресликавања меморије, операције се одвијају као рад са меморијом
 - сигнал "IO/memory" се поставља на ниско стање, што означава операцију са меморијом
 - ако је у питању изоловани уређај, потребна је додатна У/И линија
 - сигнал "IO/memory" се поставља на активно стање, што означава У/И операцију
- Рад са уређајима ће бити обрађен касније

Стања чекања



- Рад процесора је често сувише брз да би га меморија или У/И уређаји могли пратити
- Због тога се уводе стања чекања и контролни сигнал "ready"
 - Ако је стање сигнала "ready" активно, процесор не сме да претпостави да је уређај извршио операцију
 - У том случају процесор чека један циклус па поново проверава стање сигнала "ready"



Преношење блокова података



- Преношење блокова података подразумева да се једном сложеном операцијом пренеси већа количина података
- Пример примене је попуњавање кеш меморије

Блокови података и *Pentium*



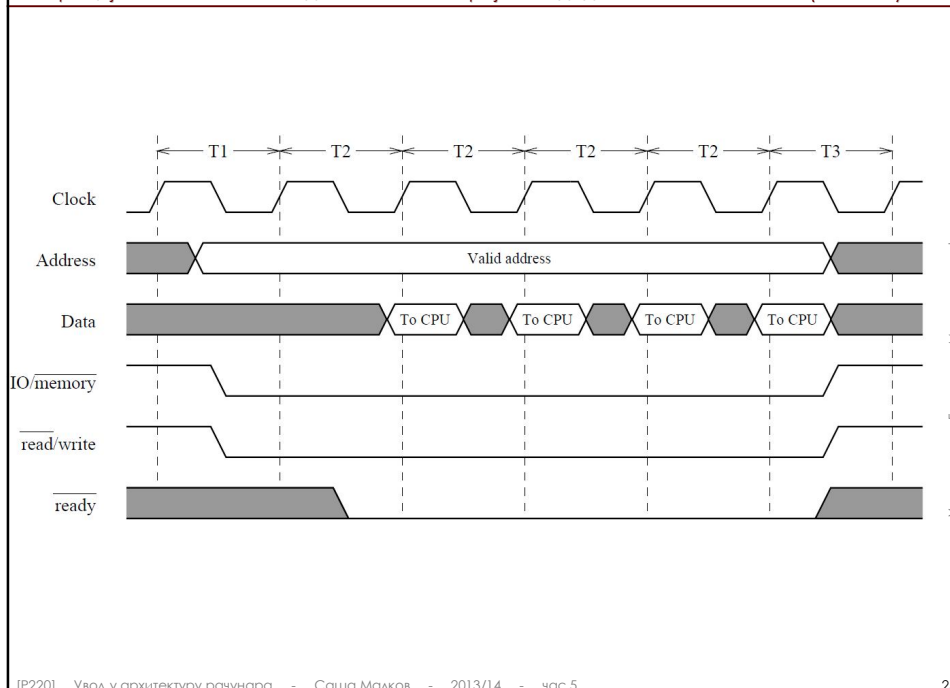
- Размотрићемо читање на примеру процесора *Intel Pentium* и агресивног режима преноса
 - све операције са блоковима код *Pentiuma* се односе на по четири податка (сваки је ширине 64 бита)
 - зато се не преноси податак о величини блока, који би иначе био потребан
 - при операцијама са блоковима адресе морају бити *иоравнише* са величином блокова
 - најнижих 5 битова адресе морају бити 0

Читање блока података



- При постављању захтева
 - процесор поставља на ниско стање додатни контролни сигнал "block", чиме означава да се ради о раду са блоком података
 - код процесора *Pentium* овај сигнал се зове "cache" за операције са кеш меморијом
- Ако нема чекања, претпоставља се да се у сваком циклусу испоручује по један податак
 - у супротном меморија захтева чекање постављањем сигнала "ready"
- Остало је као при појединачном читању

Операција читања блока података из меморије без додатних стања чекања (*Pentium*)



Асинхрона магистрала



- У случају асинхроне магистрале не употребљава се часовник за синхронизацију рада на магистралаи
- Употребљавају се операције руковања и додатни синхронизациони сигнали
- За асинхроне магистрале је уобичајено четворофазно руковање
 - додатни сигнали су
 - главна синхронизација (*MSYN*) и
 - подређена синхронизација (*SSYN*)

Четворофазно руковање



- Обухвата следеће фазе:
 - главни уређај поставља иницијализује комуникацију:
 - поставља све потребне податке на магистралу
 - поставља главни синхронизациони сигнал (*MSYN*)
 - подређени уређај реагује на *MSYN*:
 - чита упућене податке
 - обавља операцију
 - поставља одговор на магистралу
 - поставља подређени синхронизациони сигнал (*SSYN*)
 - главни уређај реагује на *SSYN*:
 - чита одговор са магистрале
 - искључује сигнал *MSYN*
 - подређени уређај реагује на искључивање *MSYN*:
 - искључује сигнал *SSYN*

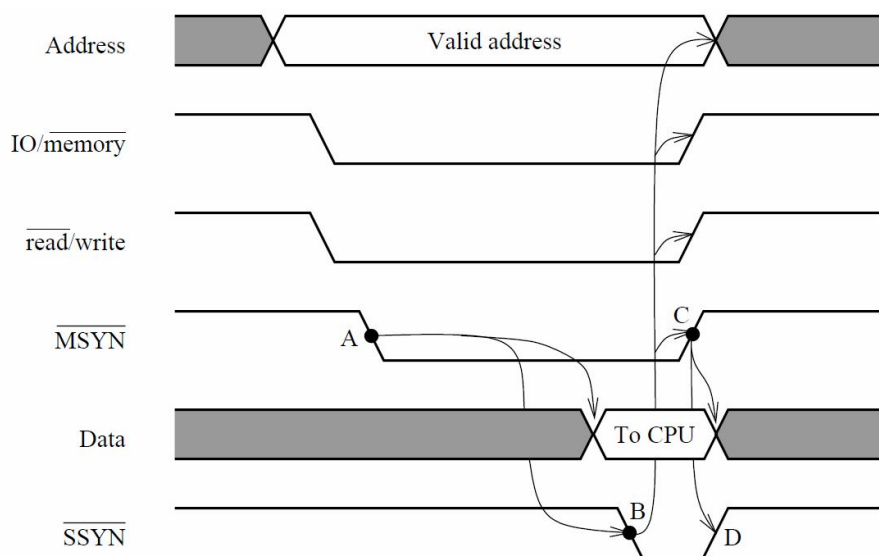
Пример четворофазног руковања



- Операција читања изгледа овако:
 - процесор
 - поставља адресу и контролне сигнале
 - поставља главни синхронизациони сигнал (*MSYN*)
 - меморија
 - чита упућене податке обавља операцију
 - поставља податке на магистралу
 - поставља подређени синхронизациони сигнал (*SSYN*)
 - процесор
 - чита одговор са магистрале
 - поништава адресну магистралу и контролне сигнале
 - искључује сигнал *MSYN*
 - меморија
 - поништава магистралу података
 - искључује сигнал *SSYN*

Универзитет у Београду - Математички факултет

Пример четворофазног руковања – операција читања



Универзитет у Београду - Математички факултет

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 маг.



- 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*

Врсте арбитраже



- Арбитража се може одвијати статички или динамички
 - Статичка арбитража подразумева да се расподела међу главним уређајима одиграва на унапред одређен начин
 - Динамичка арбитража одлучује на основу захтева пристиглих од уређаја
- Већина имплементација почива на динамичкој арбитражи

Статичка арбитража



- Расподела међу главним уређајима одиграва на унапред одређен начин
 - нпр, додељивањем магистрале редом, “у круг”
- Добре стране
 - једноставност
- Недостаци
 - додељивање чак и када уређајима није потребна
 - неефикасност

Динамичка арбитража (1)



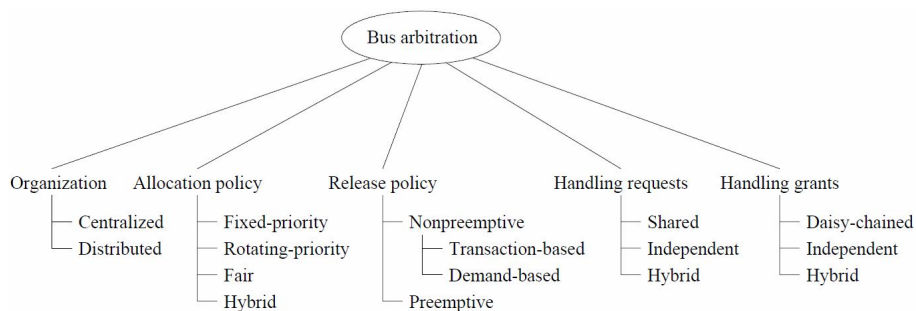
- Потенцијални главни уређај захтева магистралу, а она се арбитражом додељује као одговор на захтев
- Сваки главни уређај има додатне линије за захтевање и одобравање употребе магистрале
 - Главни уређај путем линије за захтевање магистрале (*request line*) да саопшти да му је потребна магистрала
 - Да би започео трансакцију, мора да путем линије за одобравање употребе магистрале (*grant line*) добије дозволу за употребу
- Поступак арбитраже се састоји од
 - политике додељивања магистрале и
 - политике ослобађања магистрале

Врсте динамичке арбитраже



- Динамичке арбитраже се могу разликовати по:
 - организацији (начину имплементације)
 - политици додељивања
 - политици ослобађања
 - обради захтева
 - обради дозвола

Врсте динамичке арбитраже



Политике додељивања



- Постоје четири основна типа политика додељивања магистрале:
 - политике фиксних приоритета
 - политике ротирајућих приоритета
 - равноправне политике
 - хибридне политике

Политике фиксних приоритета



- Сваком главном уређају се додели фиксан приоритет
- Када више главних уређаја захтева магистралу, добија је онај са највишим приоритетом
- Веома је важно да се приоритети пажљиво доделе
 - у супротном уређај са вишим приоритетом може вечито преузимати магистралу од других уређаја (тзв. *изгладњивање*)
 - изгладњивање не би требало да представља проблем ако захтеви не стижу непрекидно
- Ова политика се обично употребљава за У/И уређаје и за услуге *DMA*

Политике ротирајућих приоритета



- У овом случају приоритети главних уређаја нису фиксни већ представљају функцију времена чекања на магистралу
 - што дуже уређај чека, то му је већи приоритет
- Оваквом политиком се избегава изгладњивање
- Подваријанта ове политике је да се уређају који је управо добио магистралу спусти приоритет
 - ако се при томе приоритет увек спусти на најнижи, добија се расподела приоритета “у круг” (*round robin*)
 - еквивалентно распоређивању захтева у ред за чекање

Равноправне политике



- Равноправност је важан критеријум додељивања
 - у основном облику спречава изгладњивање
 - на пример, ротирајуће политике су равноправне
 - равноправне политике не морају да употребљавају приоритете
- Равноправност се може дефинисати на више начина
 - на пример, по класама приоритета уређаја или по класама приоритета захтева
- Примери равноправности:
 - сви захтеви у предефинисаном прозору времена морају бити задовољени пре одобравања захтева у наредном прозору
 - захтев не сме да чека дуже од $M \cdot t_s$
- Пример:
 - *FCF* дефинише равноправност кроз максимално допуштено време чекања

Хибридне политике



- Хибридне политике се заснивају у комбинованој употреби приоритета и правила равноправности
- Називају се и *комбиноване политике*
- Пример:
 - арбитража PCI магистрале користи хибридну политику додељивања

Универзитет у Београду - Математички факултет

Политике ослобађања



- Политике ослобађања магистрале се односе на услове под којима тренутни главни уређај ослобађа магистралу за друге уређаје
- Деле се на:
 - политике без планирања
 - политике са планирањем

Универзитет у Београду - Математички факултет

Политике без планирања



- Главни уређај, који употребљава магистралу, ослобађа магистралу добровољно
- Деле се на:
 - политике засноване на трансакцијама
 - политике засноване на захтевима

Политике зас. на трансакцијама



- Магистрала се ослобађа када се заврши текућа трансакција
- Ако је уређају поново потребна магистрала, он мора да пошаље нови захтев
- Предности
 - најједноставнија политика за имплементацију
 - ослобађањем магистрале после сваке трансакције гарантује се равноправност
- Слабости
 - ако већину захтева шаље само један главни уређај непотребно се уводе арбитраже за сваку трансакцију, чиме се смањују перформансе
 - пример је рад са меморијом у једнопроцесорским системима
 - већина захтева стиже од процесора
 - захтеви за DMA стижу релативно ретко

Политике зас. на захтевима



- Главни уређај ослобађа магистралу на крају текуће трансакције само ако постоје захтеви од других уређаја
- Ако је уређају и даље потребна магистрала, а не постоје захтеви од других уређаја, он може да настави да употребљава магистралу
- Предности
 - ефикаснија употреба у односу на пол. засноване на трансакцијама
- Слабости
 - нешто сложенија имплементација

Политике са планирањем



- У случају политика без планирања један уређај може да задржи магистралу дуже време
 - нпр, пренос великог блока података може да траје и значајно дуго
 - то производи проблеме у случају услуга које морају радити у реалном времену
- Политике са планирањем омогућавају да дугачка трансакција буде принудно прекинута у одређеним случајевима

Организација арбитраже



- Арбитража се имплементира централизовано и дистрибуирано
- У случају централизоване имплементације
 - један централни арбитар прима захтеве од свих главних уређаја
 - на основу политике додељивања арбитар додељује магистралу
 - по завршетку трансакције, главни уређај ослобађа магистралу у складу са политиком ослобађања
- У случају дистрибуиране имплементације
 - хардвер за арбитражу је дистрибуиран по главним уређајима
 - дистрибуиран алгоритам се употребљава за одређивање главног уређаја коме ће се доделити магистрала

Централизована арбитража



- Може да се имплементира на више начина
- Разматрамо три основна механизма
 - Уланчавање
 - Независни захтеви
 - Хибридна схема

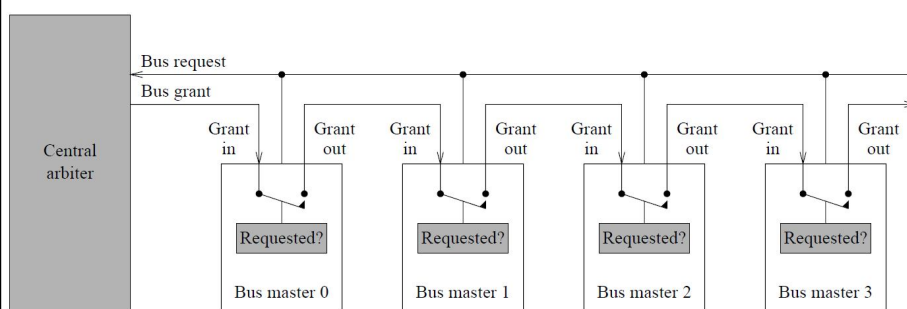
Уланчавање (1)



- Уланчавање користи једну линију за захтеве, коју деле сви главни уређаји
 - Када централни арбитар прими захтев, он шаље одобрење за употребу магистрале првом главном уређају у ланцу
 - Уређај у ланцу прослеђује сигнал ако није захтевао магистралу, а не прослеђује га ако јесте
 - Тако се сигнал за одобравање прослеђује низ ланац све док не дође до неког од уређаја који су захтевали магистралу
 - Први такав уређај у ланцу добија магистралу

Универзитет у Београду - Математички факултет

Уланчавање (2)



Универзитет у Београду - Математички факултет

Уланчавање (3)

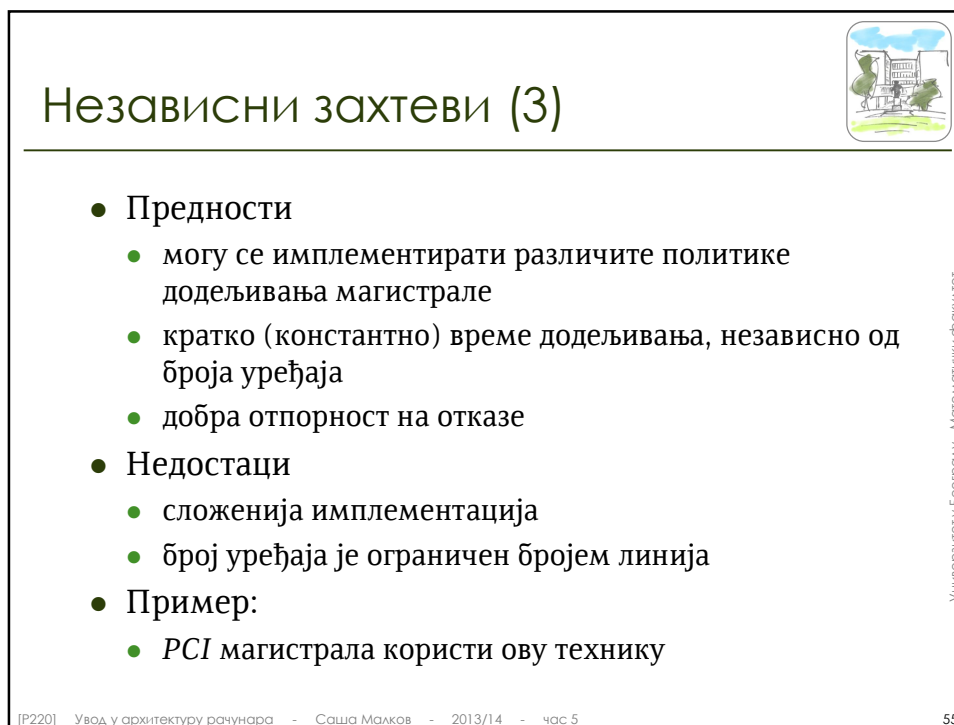


- Предности
 - једноставна имплементација
 - захтева само три контролне линије по уређају
 - арбитар не ограничава број уређаја
 - нити његова имплементација зависи од броја уређаја
- Недостаци
 - имплементира политику фиксних приоритета
 - уређај има виши приоритет ако је ближи арбитру
 - трајање арбитраже је пропорционално броју главних уређаја
 - схема није отпорна на отказивање
 - ако неки уређај откаже, ниједан уређај нижег приоритета не може добити магистралу

Независни захтеви (1)



- Арбитар се повезује са сваким уређајем путем посебних линија за захтеве и одобравање
 - Када главни уређај захтева магистралу, шаље захтев путем своје линије захтева
 - Када арбитар прими захтев, на основу политике додељивања одређује који уређај ће добити магистралу



Хибридна схема (1)



- Претходне две схеме представљају екстреме
 - уланчавање је једноставније али 3 значајне слабости
 - схема независних захтева решава проблеме али је скупља за имплементацију
- Хибридна схема дели главне уређаје на N класа
 - за сваку класу постоје посебне линије за захтеве и одобрења
 - у оквиру сваке класе уређаја примењује се уланчавање

Универзитет у Београду - Математички факултет

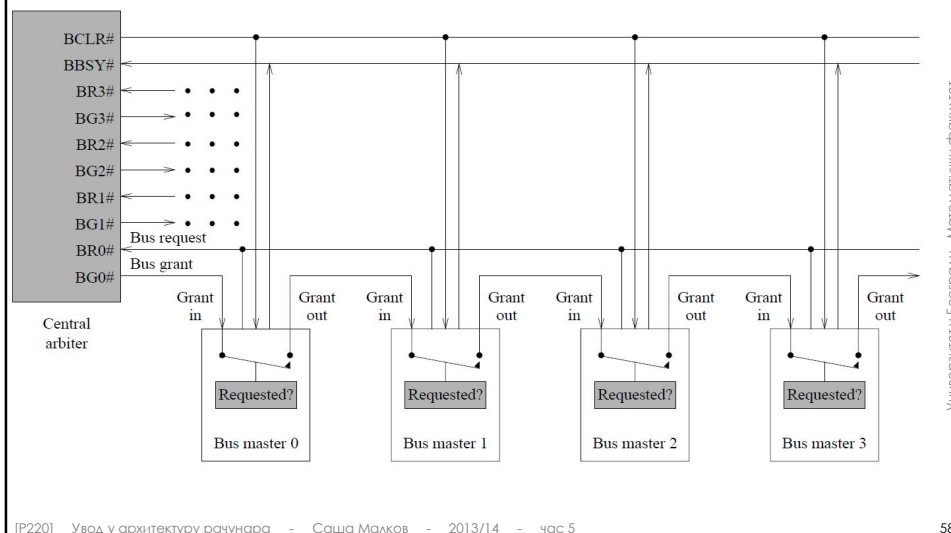
Хибридна схема (2)



- Пример – VME магистрала (фамилија процесора *Motorola 68000*)
 - четири пара линија за захтеве и одобрења ($BR0, BG0, BR1, BG1, BR2, BG2, BR3, BG3$)
 - сви уређаји деле линије заузетости и поништавања ($BBSY, BCLR$)
 - три политике додељивања
 - фиксни приоритети ($BR0$ је најнижи, а $BR3$ највиши)
 - ротирајући приоритети
 - уланчавање (ако се сви уређаји вежу на $BR3$)
 - подразумевана политика ослобађања је без планирања, заснована на трансакцијама
 - у случају фиксних приоритета може да се користи планирано ослобађање, у случају пристизања захтева са вишим приоритетом (арбитар прекида трансакцију путем линије поништавања)

Универзитет у Београду - Математички факултет

Хибридна схема (3)



Дистрибуирана арбитража



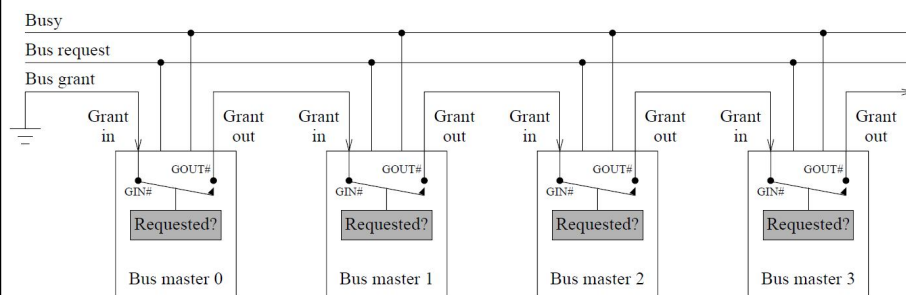
- Главни уређаји сами одређују ко ће добити магистралу за наредну трансакцију
- Хардвер за арбитражу је дистрибуиран између главних уређаја
- Могуће су различите схеме уланчавања и независних захтева, као и у случају централизоване арбитраже

Дистрибуирано уланчавање



- При уланчавању арбитар само иницијализује сигнал за одобравање
- Исто се може постићи и без арбитра
 - линија захтева се повезује тако да има ниско стање акко један или више уређаја зајтевају магистралу
 - тренутни главни уређај одржава стање линије заузетости
 - извор линије одобравања се везује на уземљење, тако да увек има ниско стање
 - све остало је неизмењено

Дистрибуирано уланчавање

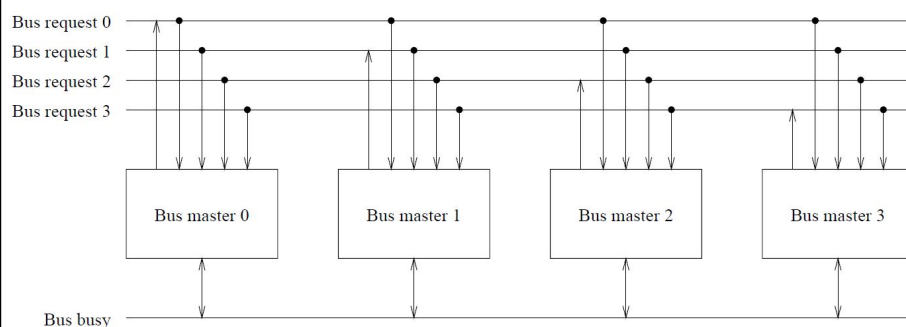


Дистрибуирани независни захтеви



- Употребљавају се независне линије захтева и одобравања
- Поступак је као у случају централизоване арбитраже
- Наредни пример је сличан политици фиксних приоритета
- Изгладњавање се разрешава тако што уређај највишег приоритета који је управо употребљавао магистралу неће поставити нов захтев све док сви уређаји нижег приоритета који су захтевали магистралу не добију одобрење

Дистрибуирани независни захтеви

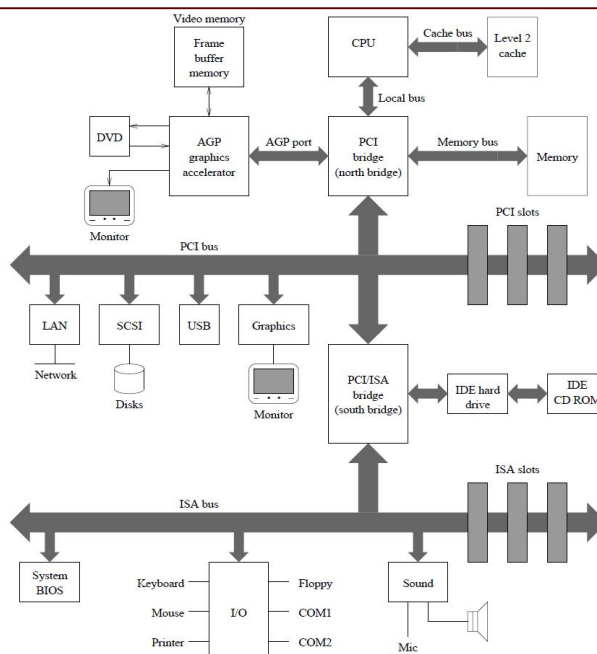


Примери магистрала



- На *PC* рачунарима постоји више врста магистрала
 - магистрала кеша
 - магистрала меморије
 - *PCI*
 - *PCI-X*
 - *ISA*
 - *AGP*
 - и друге

PC рачунар



Магистрала ISA (1)



- Магистрала ISA (енгл. *Industry Standard Architecture*) је настала као основна магистрала рачунара IBM PC
 - 1981.
 - практично је пресликавала линије процесора 8088 на матичну плочу
 - 8088 је јевтинија верзија процесора 8086, са 8-битном магистралом података
 - процесорски зависна
- Универзална - коришћена је како за рад са
 - меморијом
 - свим улазно/излазним уређајима

Магистрала ISA (2)



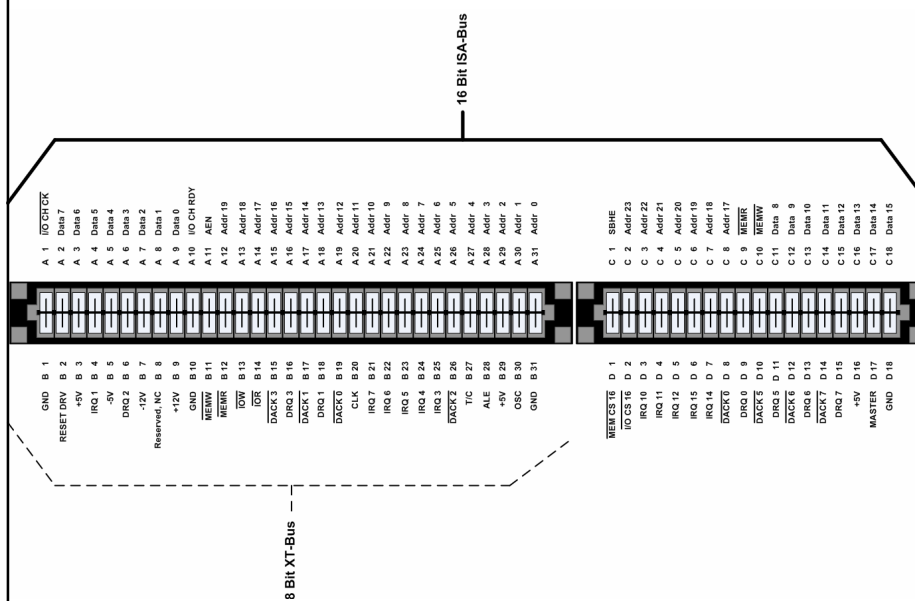
- 20-битна адресна магистрала
- 8-битна магистрала података
- укупно 62 линије:
 - 20 адресних линија
 - 8 линија података
 - 6 линија за прекиде
 - по једна контролна линија за читање из меморије, писање у меморију, читање са У/И и писање на У/И
 - 4 линије за захтеве и 4 за одобравање DMA захтева
 - ...

Магистрала ISA (3)



- Када је представљен *IBM PC AT*, са 16-битним процесором 80286, магистрала је постала уска
 - 1984.
 - 80286 има 24-битне адресе и 16-битне податке
- Магистрала *ISA* је надограђена
 - додато је укупно 36 нових линија
 - 8 линија података
 - 4 линије адресе
 - још 5 линија за прекиде
 - још 4 линије за *DMA*
 - неколико нових контролних сигнала
 - један који означава да ли је комуникација по старом 8-битном или по новом 16-битном протоколу
 - тај додатак је називан “16-битна секција”

Спецификација прикључка за магистралу ISA



Магистрала *ISA* (4)



- Прописана брзина магистрале *ISA* је 8.33 MHz
 - рад са меморијом без стања чекања је захтевао два циклуса (око $2 \times 125\text{ ns} = 250\text{ ns}$)
 - ширина података од 16 бита
 - максимална брзина при раду са меморијом од око 8 MB/s
 - у то време је то било сасвим довољно

Магистрала *ISA* (5)



- Када је представљен процесор 80386, *IBM* је представио магистралу *MCA* (енгл. *Micro Channel Architecture*)
 - није отворена као стандард и није заживела
- Представљала је даље проширење магистрале *ISA*

Магистрала ISA (6)



- Магистрала ISA је употребљавана и након појаве савременијих решења (PCI), као јевтинија магистрала за спорије уређаје
- Престала је да се употребљава када је широко прихваћена магистрала USB

Литература



- *Sivarama Dandamudi, Fundamentals of Computer Organization and Design, Springer, 2002.*
- *Ненад Мишић, Увод у организацију рачунара, Математички факултет, 2009.*