

Увод у организацију и архитектуру рачунара 2

Александар Картељ
kartelj@matf.bg.ac.rs

Напомена: садржај ових слајдова је преузет од проф. Саше Малкова

Архитектура рачунара

Процесор, магистрала, меморија, ...

Појам архитектуре рачунара

- Архитектура рачунара
 - *Опис рачунарског система на логичком нивоу, из угла “програмера” (тј. техничког корисника)*
 - Обухвата различите аспекте који се тичу “програмера”
 - Основне функционалне јединице
 - процесор
 - меморија
 - ...
 - њихово функционисање
 - нпр. скуп инструкција, начин адресирања меморије,...
 - начини повезивања

Појам организације рачунара

- Организација рачунара
 - Односи се на начин имплементације и повезивања хардверских компоненти у циљу остваривања архитектуром одређених својстава

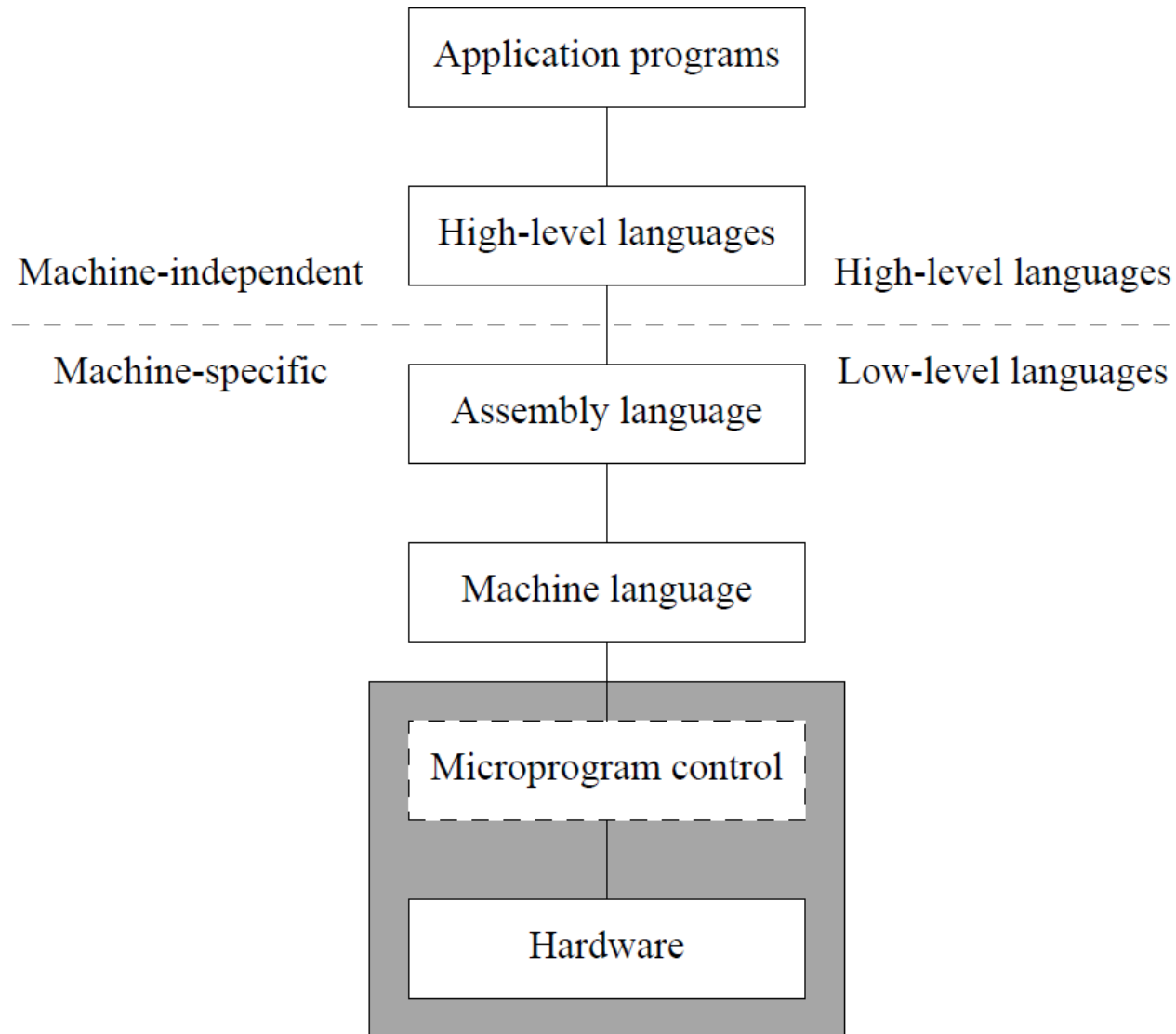
Појам пројектовања рачунара

- Пројектовање рачунара
 - Поступак који преводи архитектурне одреднице система у имплементацију применом конкретних организационих решења
 - Пројекат рачунара (*computer design*) се назива и имплементација рачунара

Појам програмирања рачунара

- Програмирање рачунара
 - Изражавање конкретних проблема на језику који рачунар може да разуме
 - Од интереса су само језици ниског нивоа, који се непосредно ослањају на конкретну архитектуру и организацију:
 - машински језик
 - асемблер

Нивои програмирања рачунара



Архитектура скупа инструкција

- Архитектура скупа инструкција
(енгл. *Instruction Set Architecture – ISA*)
 - Списак функционалности које рач. систем може да ради
 - Апстракција архитектуре скупа инструкција
 - Представља интерфејс између хардвера и најнижег нивоа софтвера

Асемблерски језици

- Два основна мотива за употребу:
 - уштеда меморијског простора
 - временска ефикасност

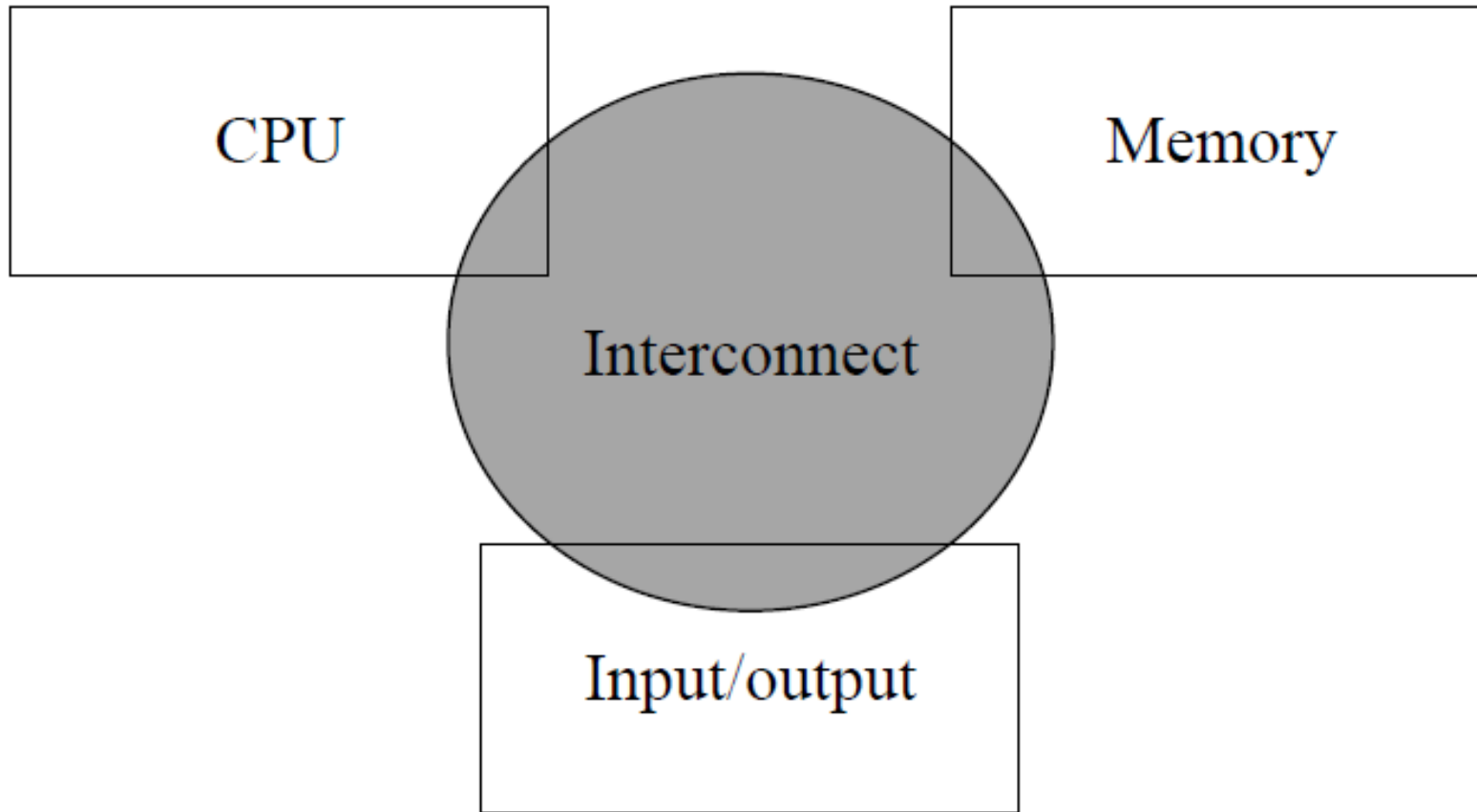
Из угла архитектуре

- Пројекат (имплементација) рачунара се посматра са вишег нивоа апстракције
 - Значајне су основне компоненте и њихови односи
 - Нису важни детаљни аспекти имплементације (нпр. којим логичким колом се имплементира нека функција АЛУ)

Основне компоненте

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

Основне компоненте



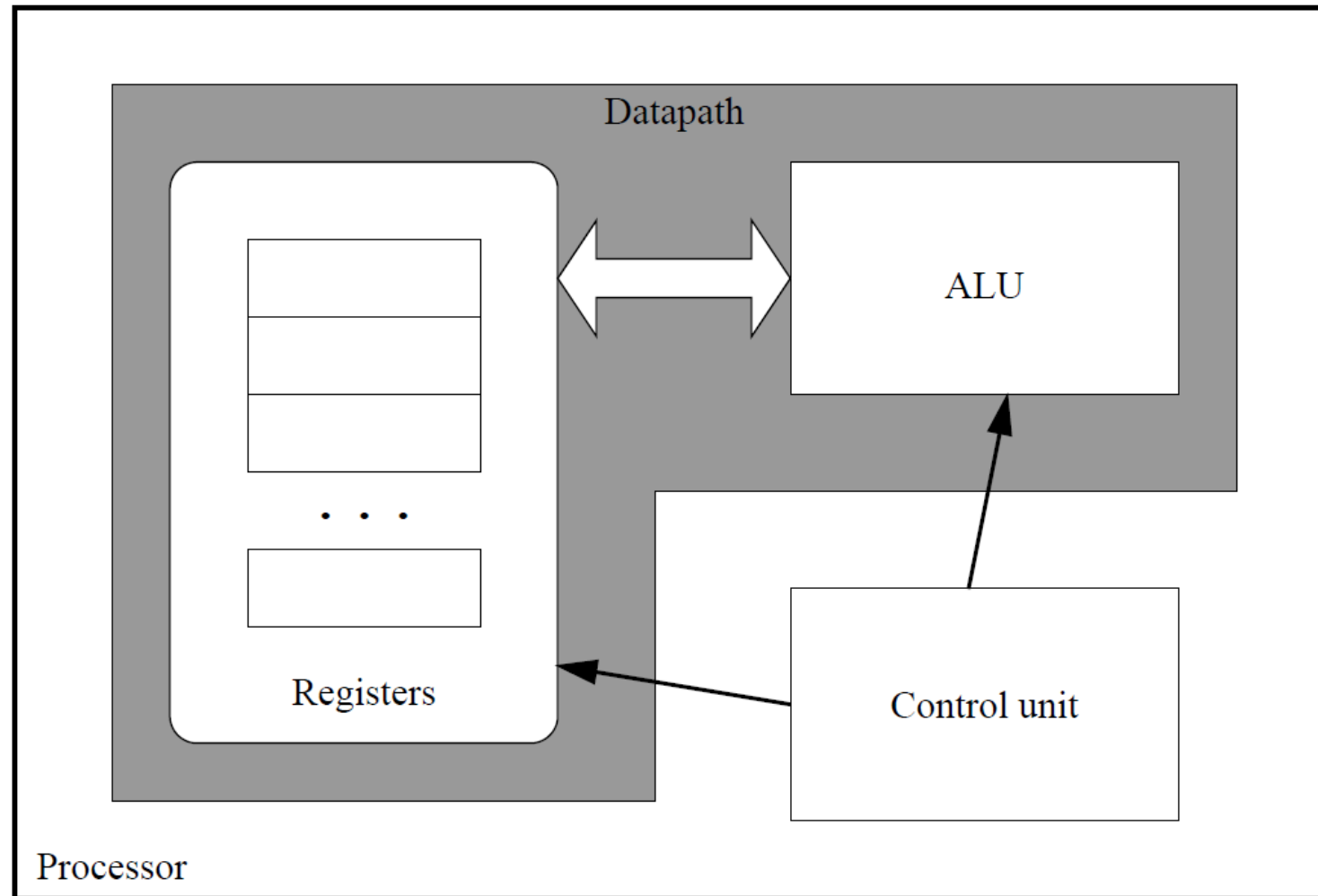
Из угла имплементатора

- Имплементатори раде на нивоу дигиталне логике
- Имплементирају логичким колима функције прописане архитектуром

Компоненте процесора

- Имплементатори виде три основне компоненте процесора:
 - контролна јединица
 - чита инструкције из главне меморије
 - декодира их и распознаје тип
 - управља радом процесора
 - регистри
 - локални меморијски простор процесора
 - начелно су сви исте величине
 - аритметичко логичка јединица (једна или више)
 - имплементација конкретних аритметичких и логичких операција

Компоненте процесора



Магистрала

Карактеристике, типови, операције, ...

Магистрала

- Магистрала је подсистем који повезује компоненте рачунарског система
- Може да се састоји од компоненти, 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)

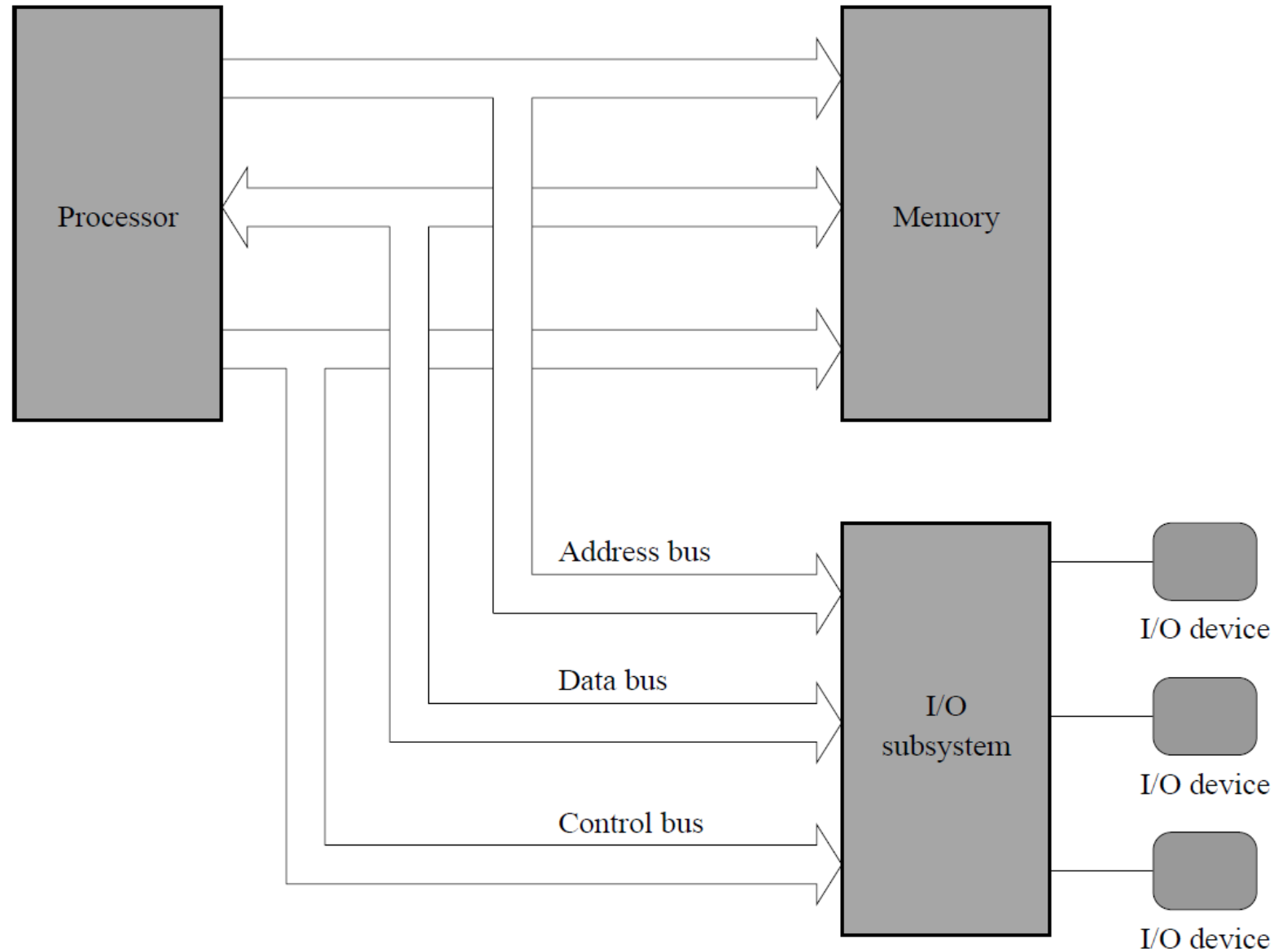
Системска магистрала

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

Системска магистрала (2)

- Мрежа која повезује компоненте рачунарског система назива се *системска магистрала* (енгл. *system bus*)
- Системска магистрала се састоји од три основне компоненте:
 - адресне магистрале
 - магистрале података и
 - контролне магистрале

Системска магистрала (3)



Спољашња (екстерна) магистрала

- Повезује уређаје који су ван процесорског система
 - *USB*
 - *FireWire*
 - серијски интерфејс
 - паралелни интерфејс

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 улазном уређају
 - ...
- Једна трансакција може да обухвати више операција
 - нпр: агресивно читање (енгл. *burst read*)

Трансакције магистрале (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 магистрали
- Свака трансакција има тачно једног главног корисника
- Неки уређаји могу бити само подређени корисници магистрале
- Други уређаји могу бити главни или подређени (али не у исто време)

Посвећене магистрале

- 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

Контролни сигнали

- Радом магистрале се управља посредством контролних сигнала путем
 - посвећене контролне магистрале или
 - мултиплексиране магистрале

Контролни сигнали (2)

- *Memory Read, Memory Write*
 - Означавају да је трансакција једна од операција са меморијом
- *I/O Read, I/O Write*
 - Означавају да трансакција обухвата улазно/излазну операцију
- *Ready*
 - Овај сигнал обично поставља подређена компонента
 - Обавештава главну компоненту да је потребно још времена
 - Главна операција обично реагује преласком у стање чекања

Контролни сигнали (3)

- *Bus request* - пре сваке трансакције компоненте најпре морају захтевати да *добују* магистралу
- *Bus grant* - арбитар магистрале бира ко ће да добије магистралу и шаље му сигнал
- *Clock* - сигнал који служи за синхронизацију рада свих компоненти рачунарског система
- *Reset* - сигнал који иницијализује рад система

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
 - Основна мотивација за проширивање је подизање пропусности магистрале, а тиме и перформанси
 - Основна мотивација за сужавање је смањивање сложености и смањивање трошкова

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

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 (жица)
 - ако се користи 128 линија података, онда је то чак 192
- Ради смањивања трошкова се уместо посвећених магистрала може се употребљавати мултиплексирана магистрала
 - назива се *магистрала адреса и података* (енгл. *AD-bus*)

Мултиплексиране магистрале

- Пример рада – читање из меморије:
 1. процесор најпре ставља на магистралу адресу
 2. меморијска јединица чита адресу и приступа локацији
 3. у међувремену процесор уклања адресу са магистрале
 4. меморијска јединица на магистралу поставља прочитан податак
- Пример рада – писање у меморију:
 1. процесор најпре ставља на магистралу адресу
 2. меморијска јединица чита адресу и приступа локацији
 3. процесор уклања адресу са магистрале и поставља податак
 4. 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
 - прекиди
 - 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

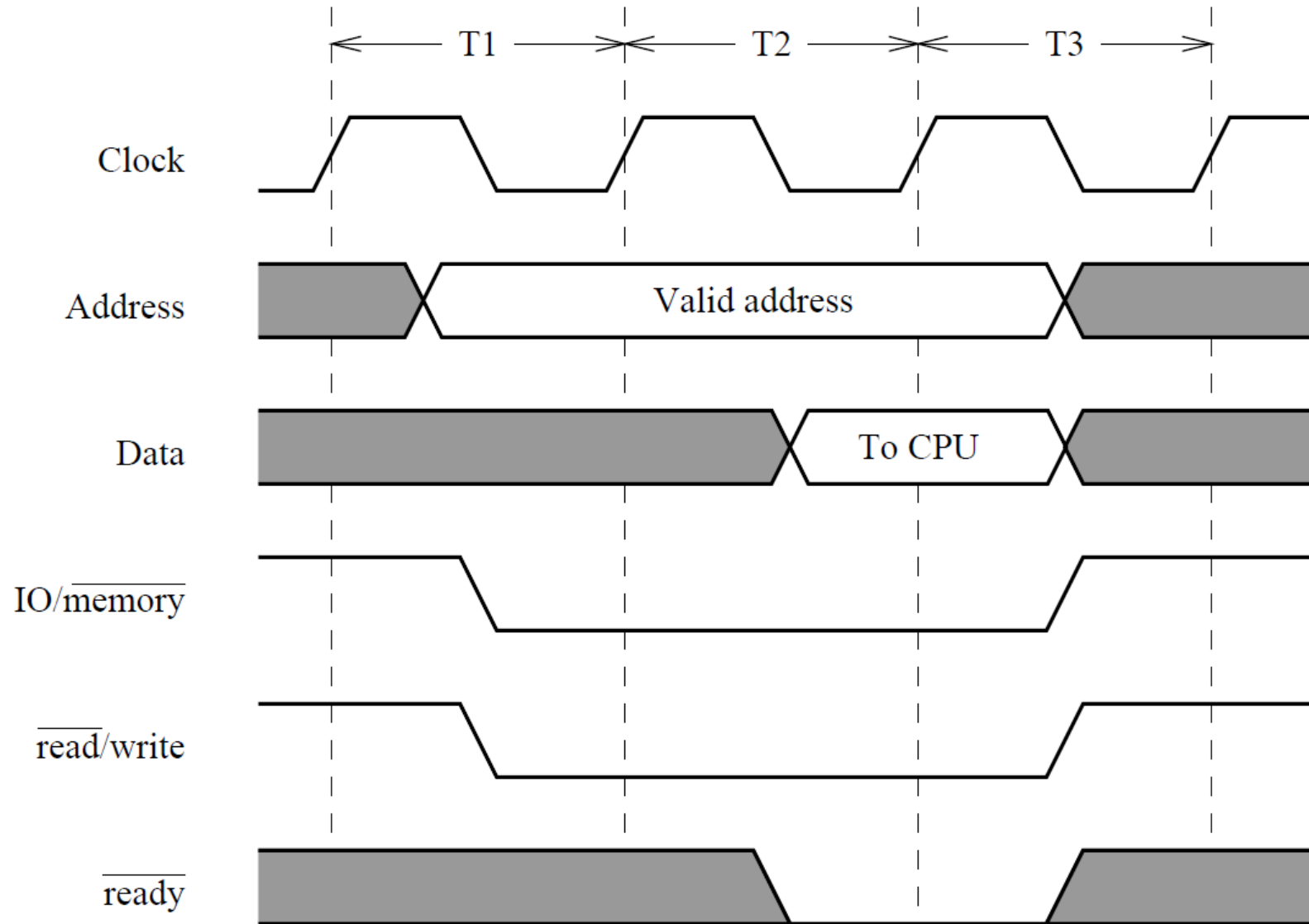
Операције синхроне магистрале

- 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

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

Операција читања (2)



Операција читања (3)

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

Операција читања (4)

- 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 податке

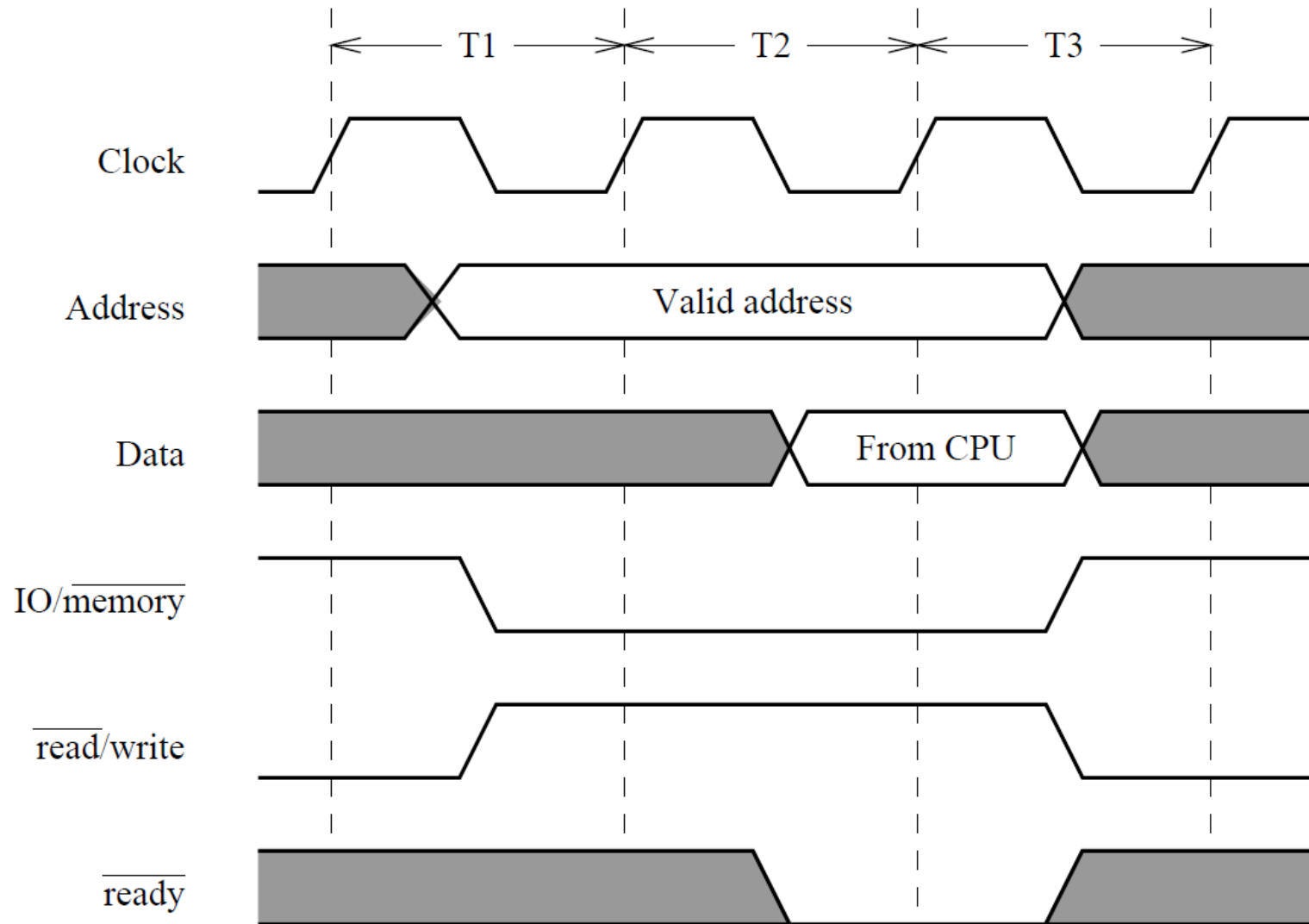
Операција читања (5)

- 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

- 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)



Операција уписивања (3)

- У циклусу T1 процесор поставља захтев за читање
 - током активног стања циклуса T1 процесор поставља на адресну магистралу исправну адресу меморијске локације са које се чита
 - затим поставља два контролна сигнала:
 - 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 (4)

- 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 податке

Операција уписивања (5)

- 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

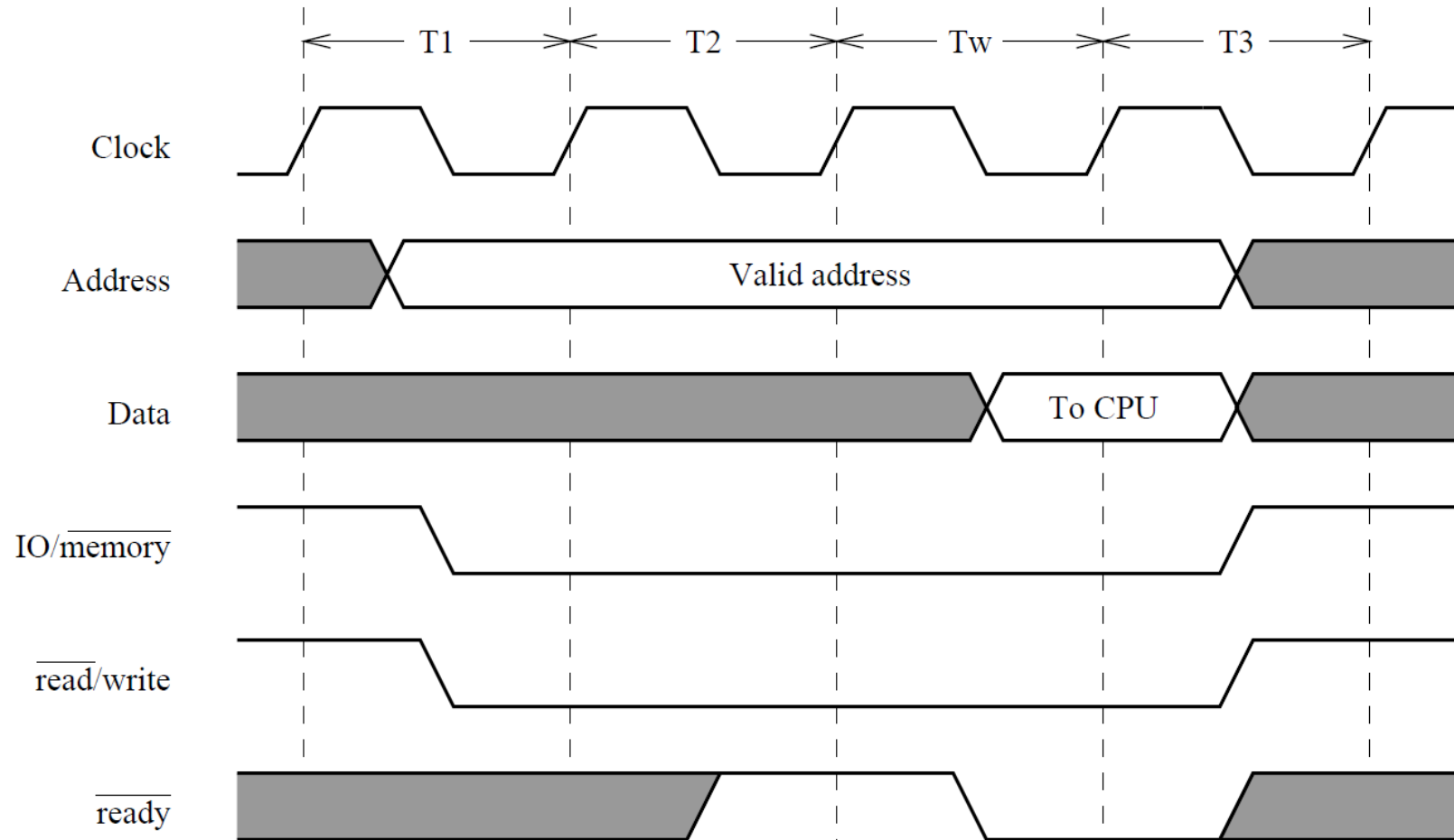
Рад са У/И уређајима

- 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

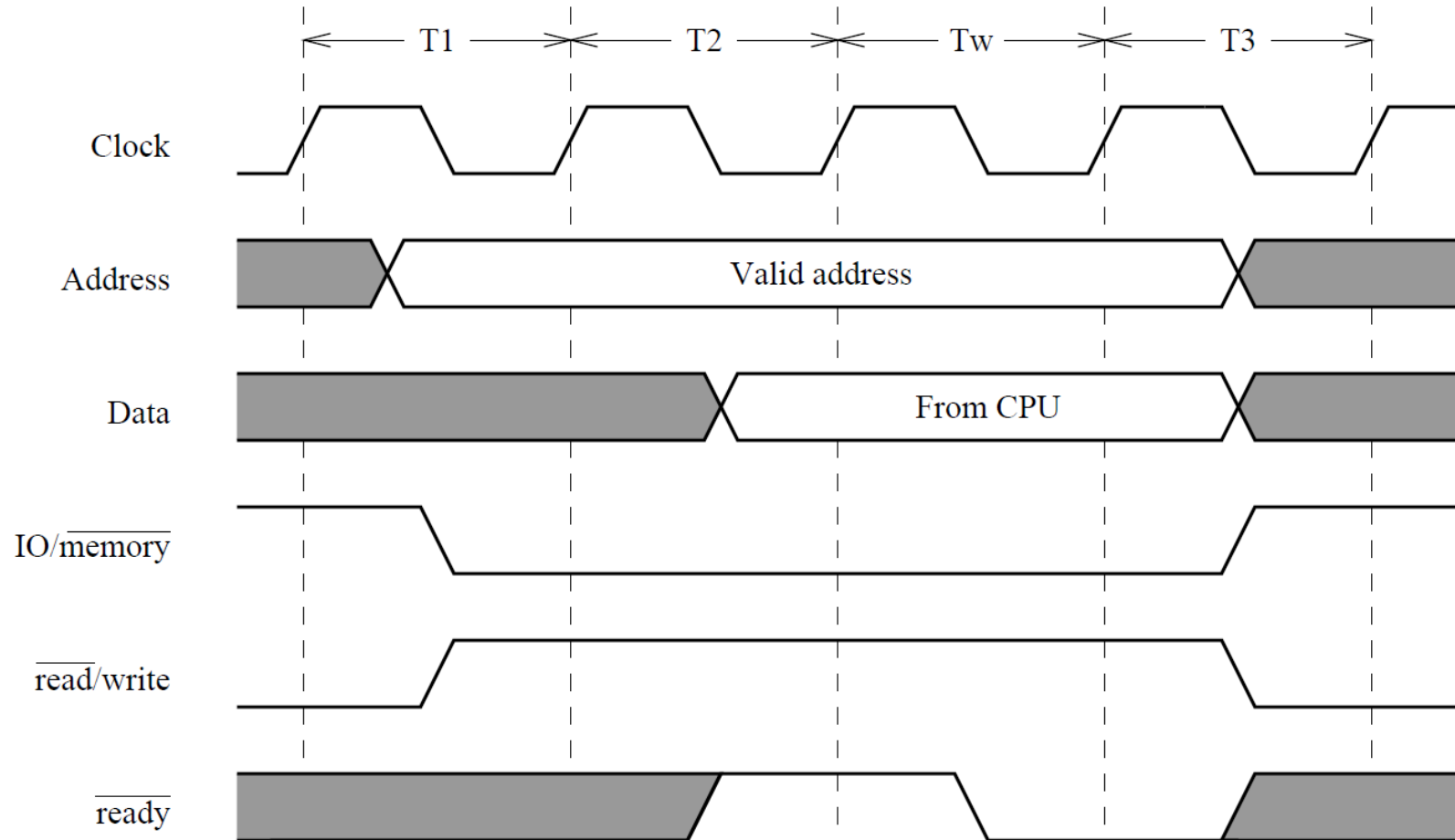
Стања чекања

- 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*”

Операција читања са стањем чекања



Операција читања са стањем чекања (2)



Преношење блокова података

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

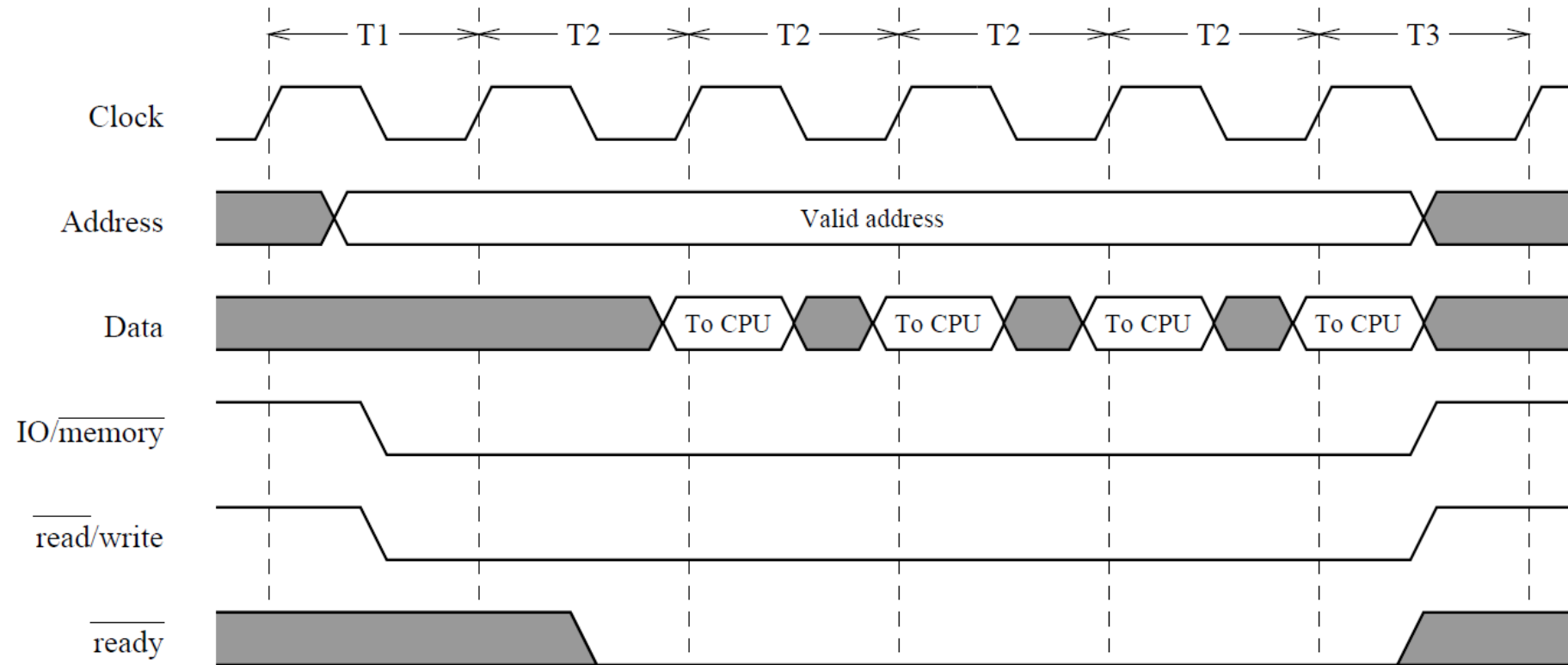
Блокови података и *Pentium*

- 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 *поравнате* 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



Асинхрона магистрала

- У случају асинхроне магистрале не употребљава се часовник за синхронизацију рада на магистрали
- 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

1. Главни уређај поставља иницијализује комуникацију:
 - поставља све потребне податке на магистралу
 - поставља главни синхронизациони сигнал (*MSYN*)
2. Подређени уређај реагује на *MSYN*:
 - чита упућене податке
 - обавља операцију
 - поставља одговор на магистралу
 - поставља подређени синхронизациони сигнал (*SSYN*)
3. Главни уређај реагује на *SSYN*:
 - чита одговор са магистрале
 - искључује сигнал *MSYN*
4. Подређени уређај реагује на искључивање *MSYN*:
 - искључује сигнал *SSYN*

Четворофазно руковање – операција читања

1. процесор

- поставља адресу и контролне сигнале
- поставља главни синхронизациони сигнал (*MSYN*)

2. меморија

- чита упућене податке обавља операцију
- поставља податке на магистралу
- поставља подређени синхронизациони сигнал (*SSYN*)

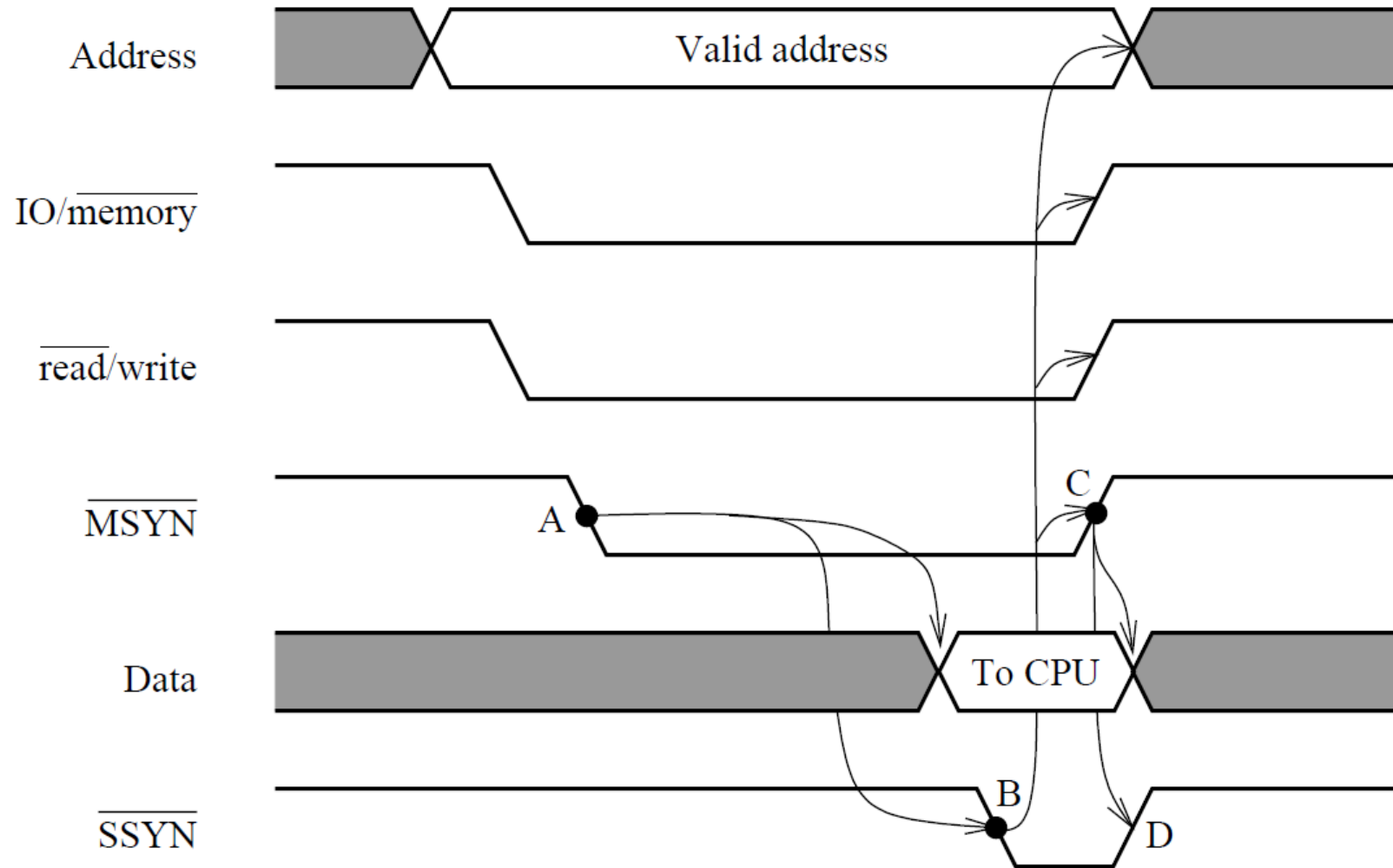
3. процесор

- чита одговор са магистрале
- поништава адресну магистралу и контролне сигнале
- искључује сигнал *MSYN*

4. меморија

- поништава магистралу података
- искључује сигнал *SSYN*

Четворофазно руковање – операција читања (2)



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

Магистрала

Арбитража магистрале

Арбитража магистрале

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

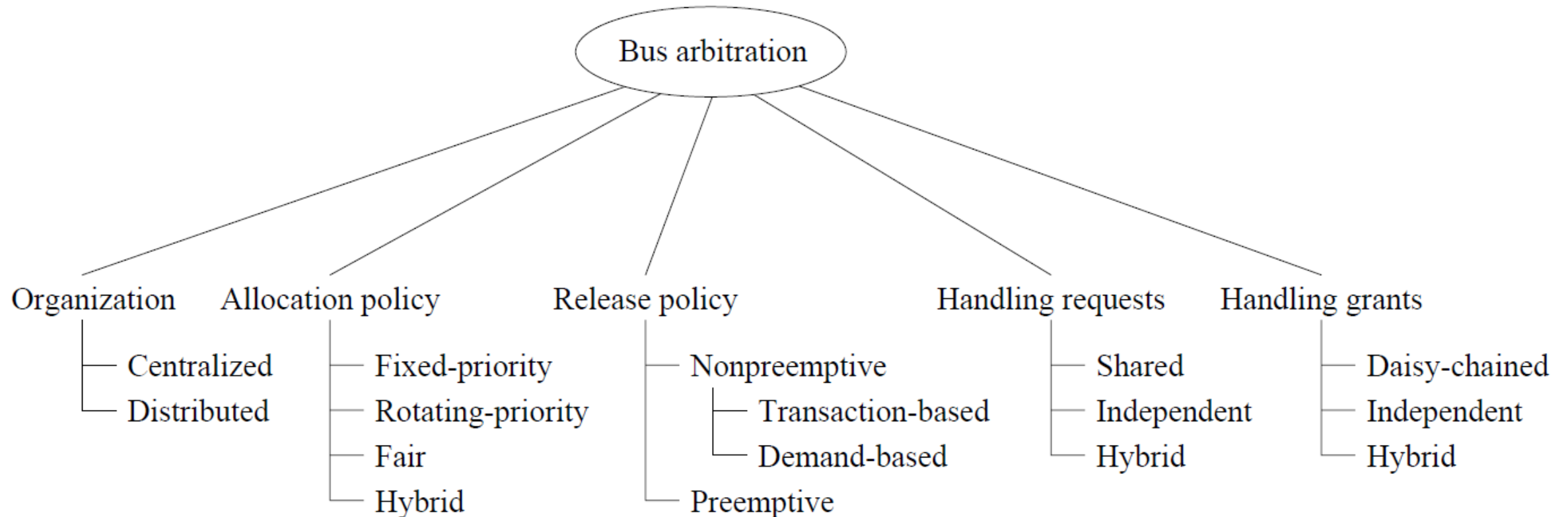
Динамичка арбитража

- 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
 - *Request line* - саопштава да му је потребна магистрала
 - *Grant line* – добија дозволу да може да користи магистралу
- Поступак арбитраже се састоји од
 - политике додељивања магистрале
 - и политике ослобађања магистрале

Врсте динамичке арбитраже

- Динамичке арбитраже се могу разликовати по:
 1. организацији (начину имплементације)
 2. политици додељивања
 3. политици ослобађања
 4. обради захтева
 5. обради дозвола

Врсте динамичке арбитраже (2)



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*)
- 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*)

Равноправне политике

- Равноправност је важан критеријум додељивања
 - у основном облику спречава изгладњивање
 - на пример, ротирајуће политике су равноправне
- 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 $M ms$

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

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

Политике без планирања

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

Политике засноване 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 уређај смањене су перформансе
 - пример је рад са меморијом у једнопроцесорским системима

Политике засноване 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 наставља да користи магистралу
- Предности
 - ефикаснија употреба у односу на политике 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

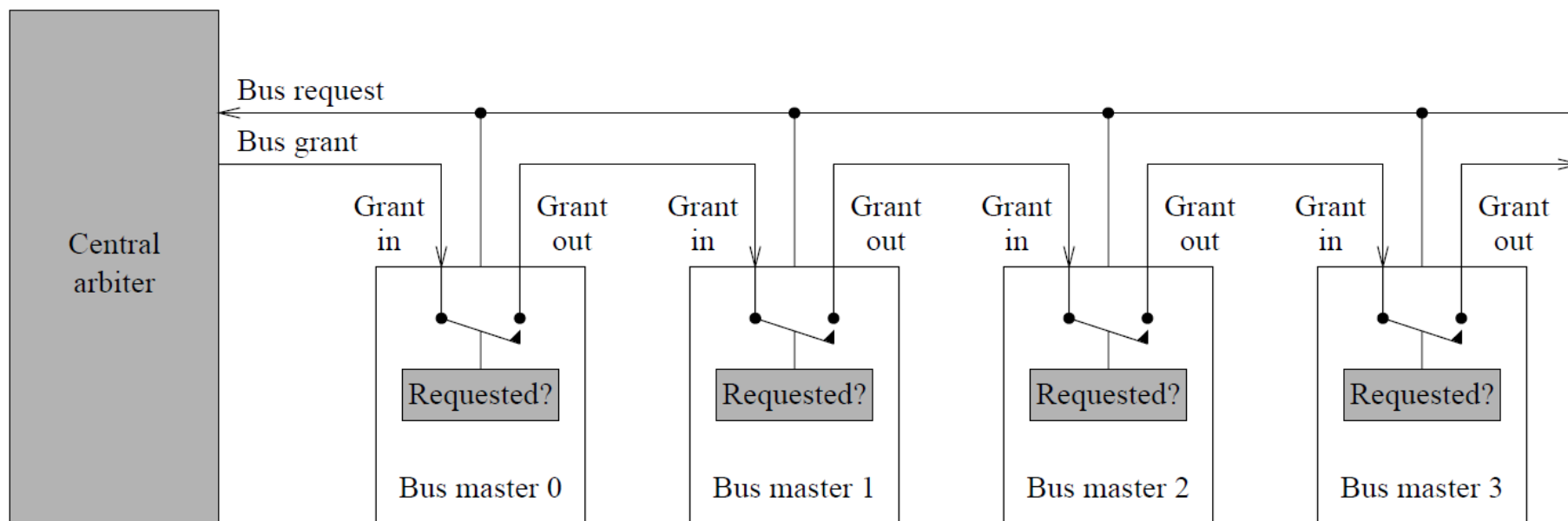
Централизована арбитража

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

Уланчавање

- 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

Уланчавање (2)



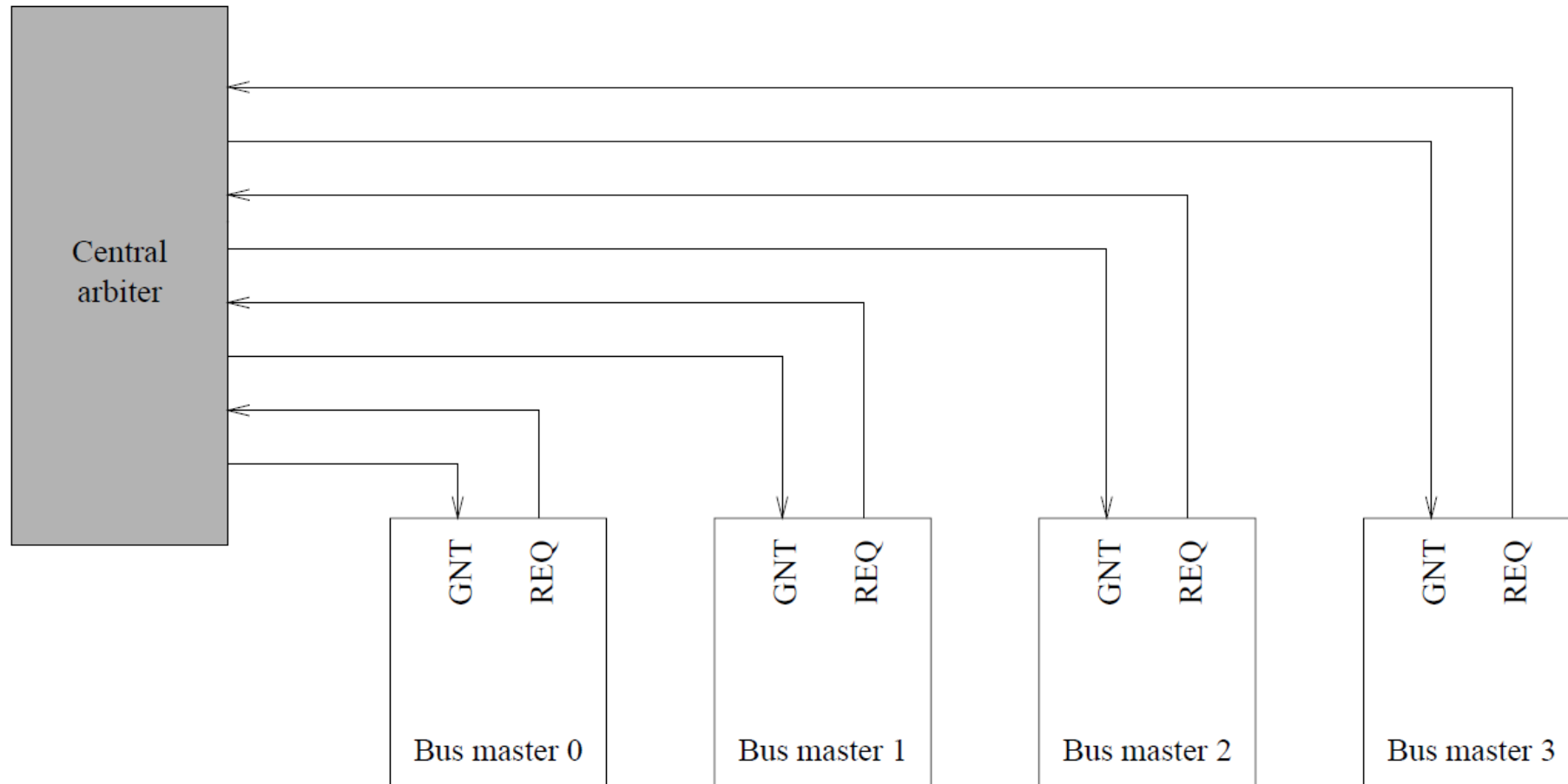
Уланчавање (3)

- 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

- 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

Независни захтеви (2)



Независни захтеви (3)

- Предности
 - могу се имплементирати различите политике додељивања магистрале
 - кратко (константно) време додељивања, независно од броја уређаја
 - добра отпорност на отказе
- Недостаци
 - сложенија имплементација
 - број уређаја је ограничен бројем линија
- Пример:
 - *PCI* магистрала користи ову технику

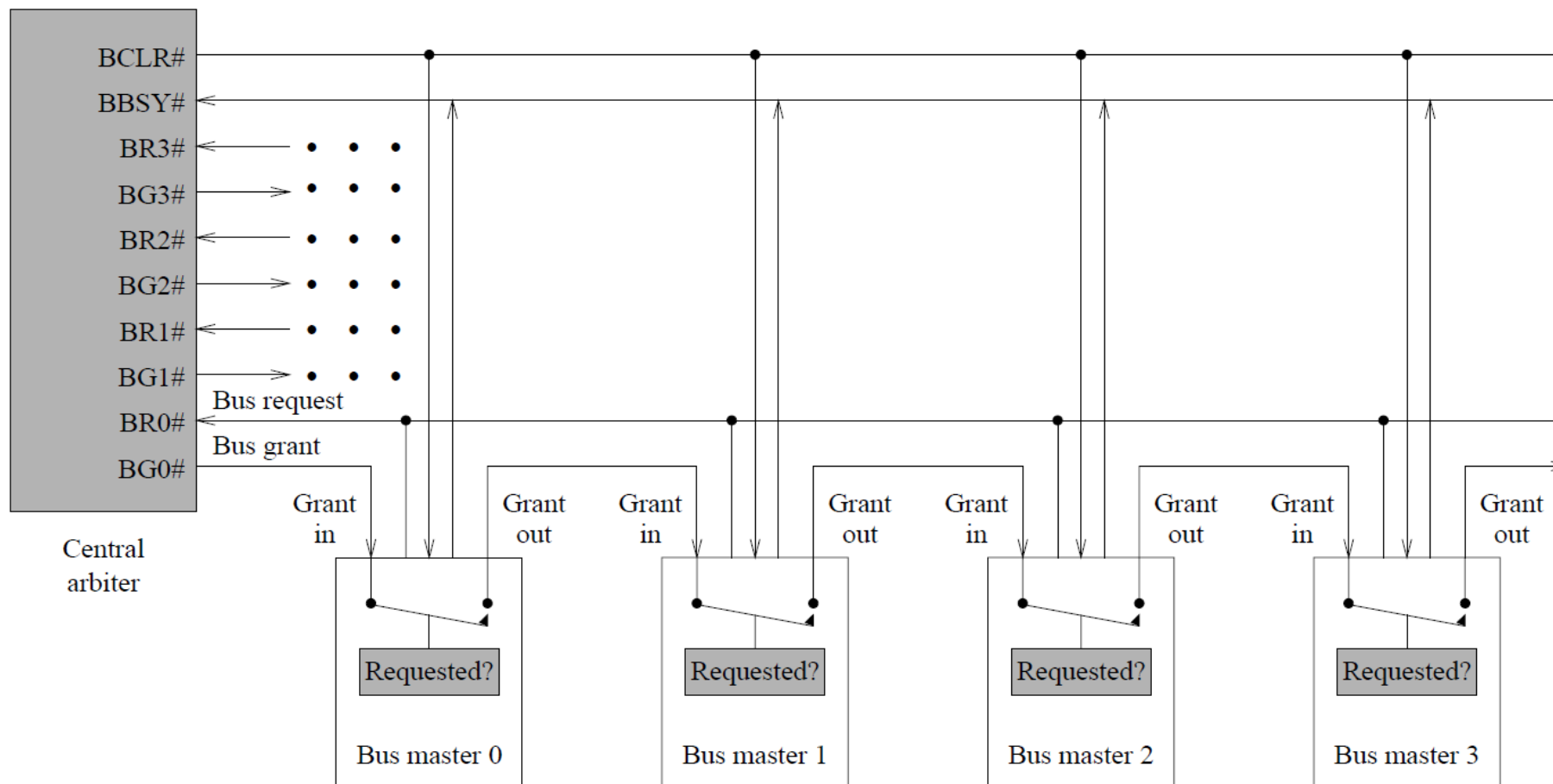
Хибридна схема

- 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

Хибридна схема (2)

- Пример:
- 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 без планирања са трансакцијама

Хибридна схема (3)



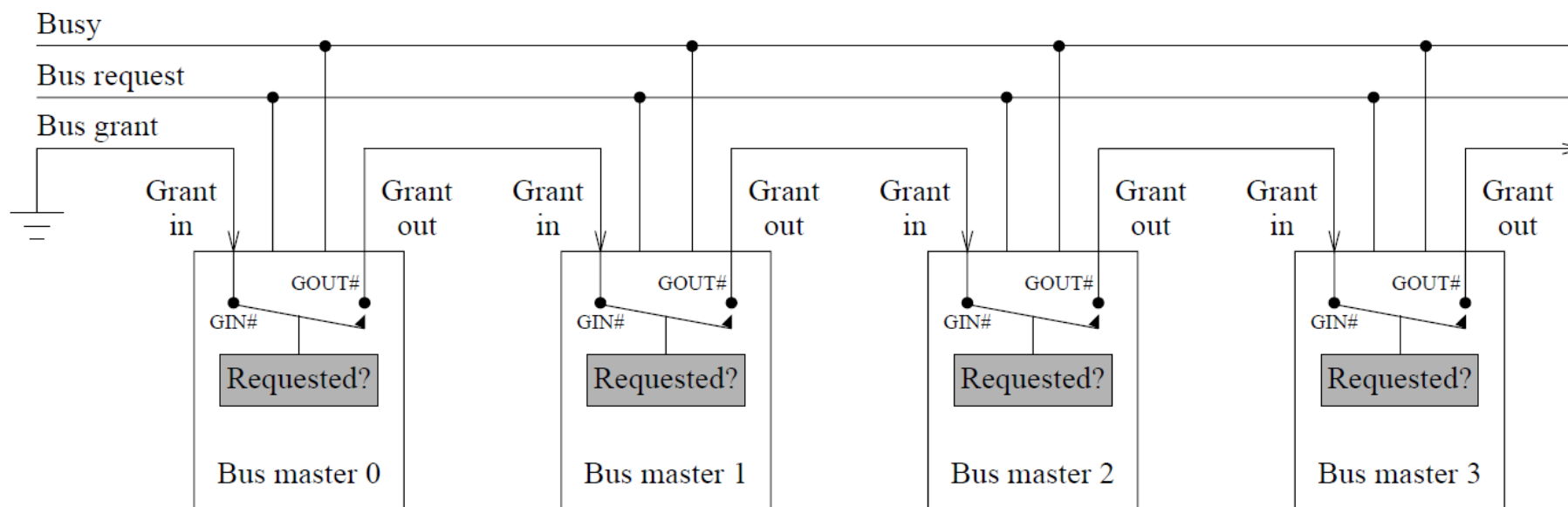
Дистрибуирана арбитража

- 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 има ниско стање
 - све остало је неизмењено

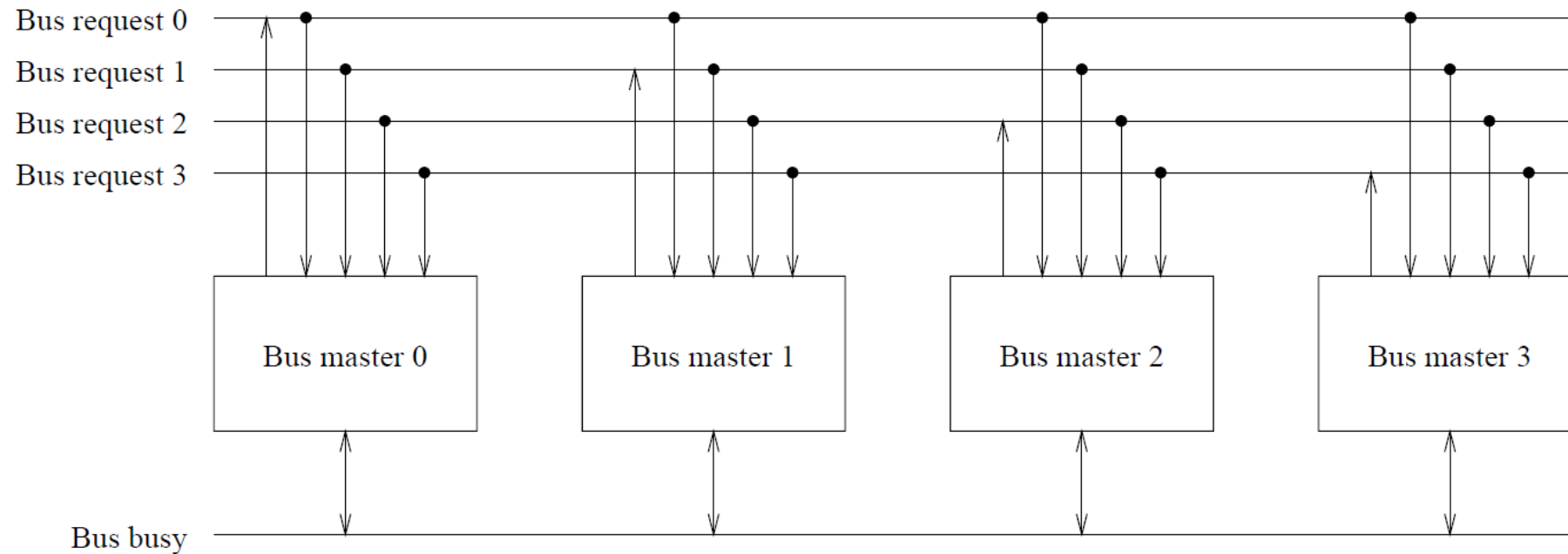
Дистрибуирано уланчавање (2)



Дистрибуирани независни захтеви

- Употребљавају се независне линије захтева и одобравања
- Поступак је као у случају централизоване арбитраже
- Наредни пример је сличан политици фиксних приоритета
- Изгладњивање 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 (2)



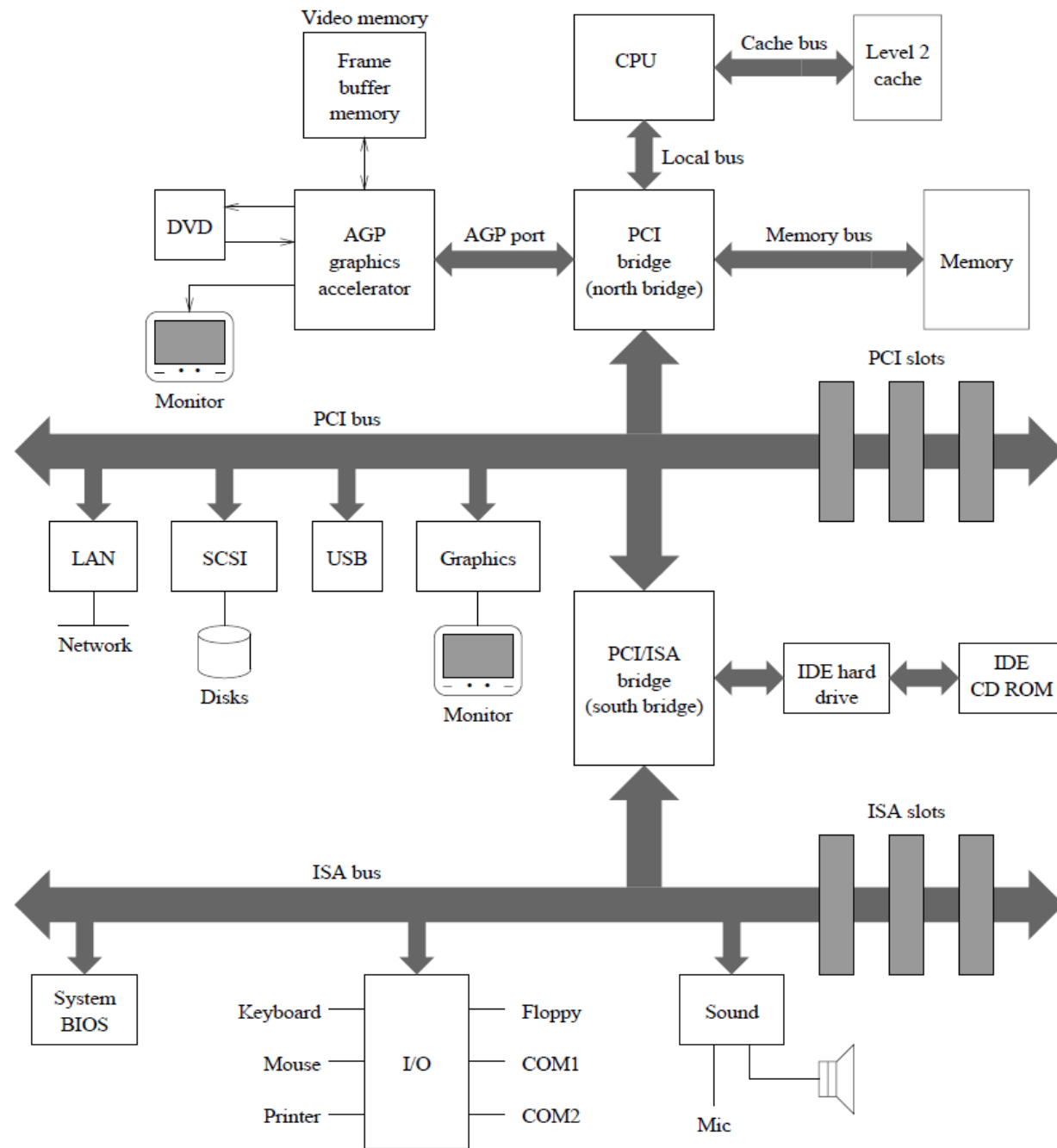
Магистрала

Неке битније магистрале

Примери магистрала

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

Магистрале



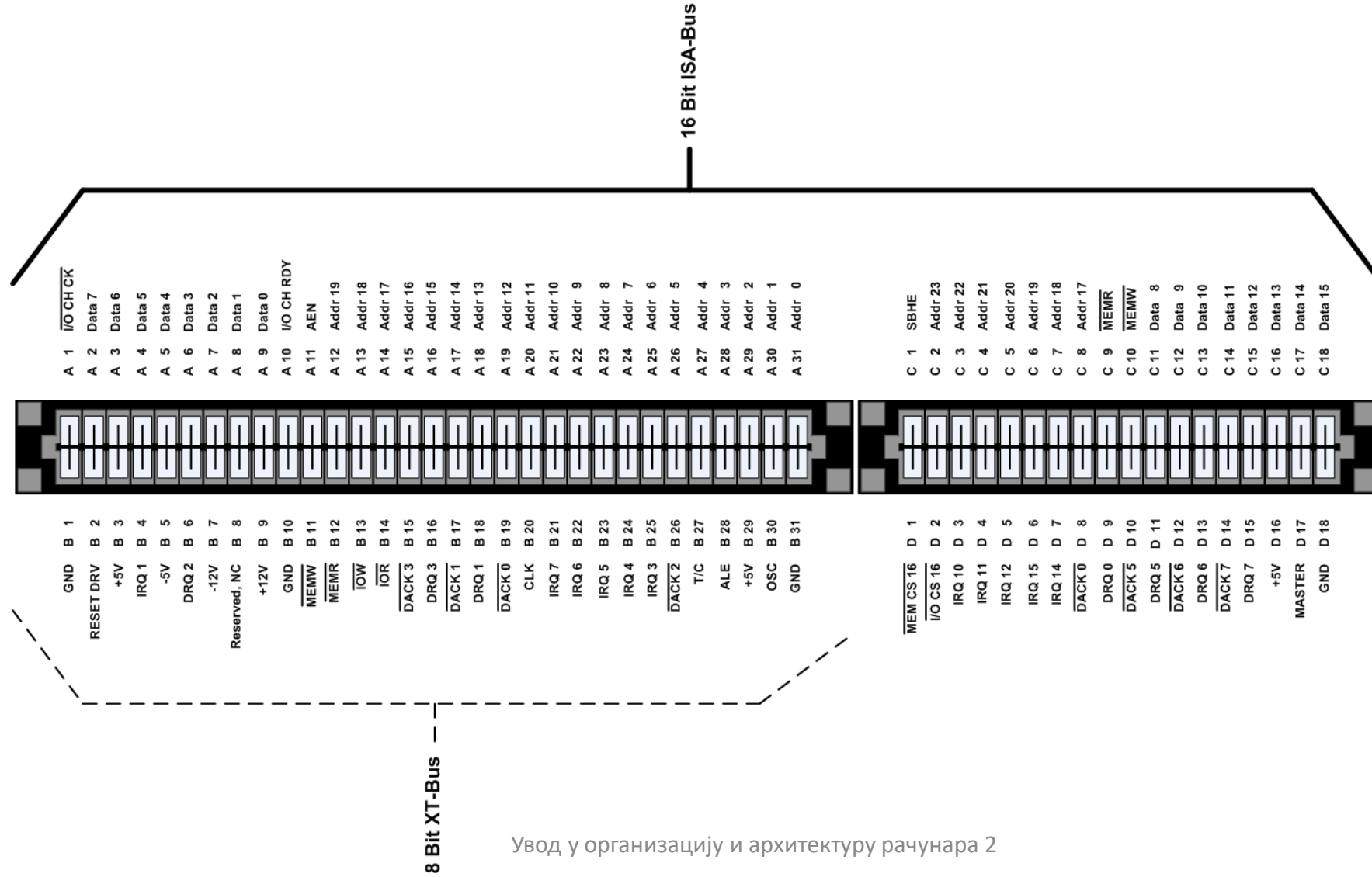
Магистрала *ISA*

- Магистрала *ISA* (енгл. *Industry Standard Architecture*) је настала као основна магистрала рачунара *IBM PC*
 - 1981.
 - практично је пресликавала линије процесора 8088 на матичну плочу
 - 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)



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)

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

Магистрала *PCI*

- Рад на магистралама *PCI* је започео *Intel* 1990. године
 - *Peripheral Component Interconnect (PCI)*
 - сви патенти су објављени у јавном власништву, ради ширег прихватања
 - оригинална спецификација је названа *V1.0*
 - верзија *V2.0* је објављена 1993. године
 - верзија *V2.1* је објављена 1995. године

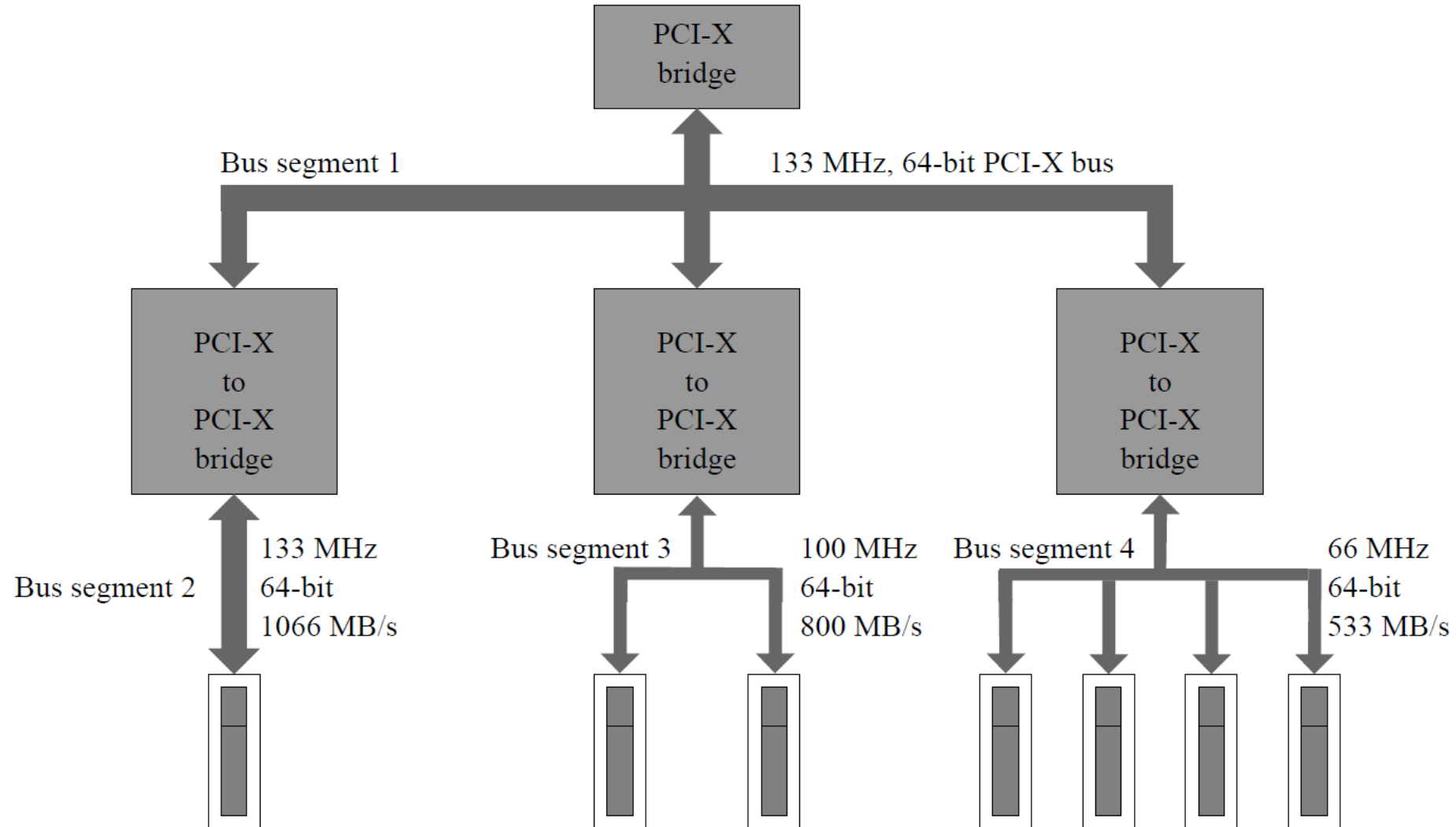
Магистрала *PCI* (2)

- независна од процесора
- изворно 32-битна, на 33.33 *MHz*, пропусност 133 *MB/s*
 - касније 64-битна, на 66 *MHz*, пропусност 528 *MB/s*
 - најчешће имплементирана као 64-битна, на 33.33 *MHz*, пропусност 266 *MB/s*
- синхрона
- мултиплексирана (адресе и подаци)
 - довољно 64 линије за 64-битне адресе и 64-битне податке
- централизовани арбитар
 - са независним линијама захтева
- старији стандард на 5V и новији на 3.3V

Магистрала *PCI-X*

- 64-битна
- 133 *MHz* (постоје и режими на 66 и 100 *MHz*)
- 1064 *MB/s*
 - у режимима *DDR* и *QDR* и до 4264 *MB/s*
- Поддржава до 256 сегмената
 - Сваки од сегмената може да има своју брзину рада
- Компатибилна наниже са магистралом *PCI*
 - на нивоу сегмената – сваки сегмент ради илу у режиму *PCI* или у режиму *PCI-X*

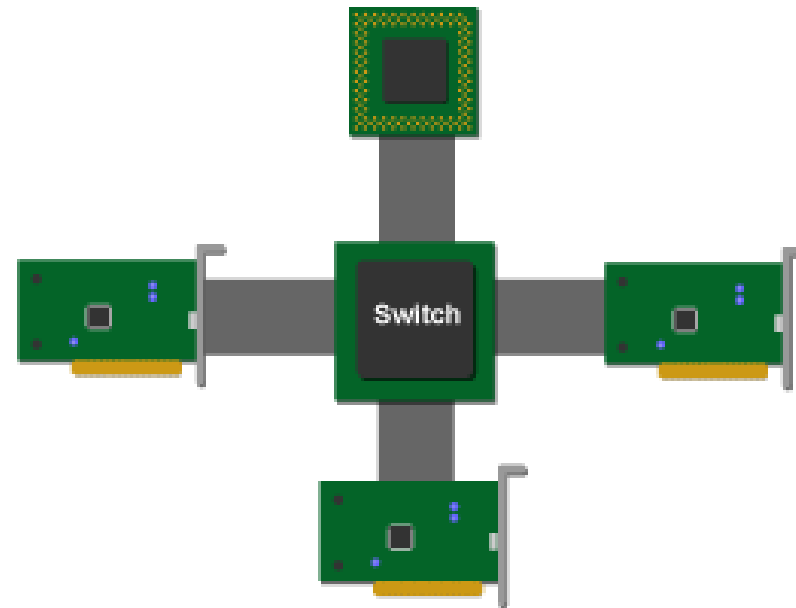
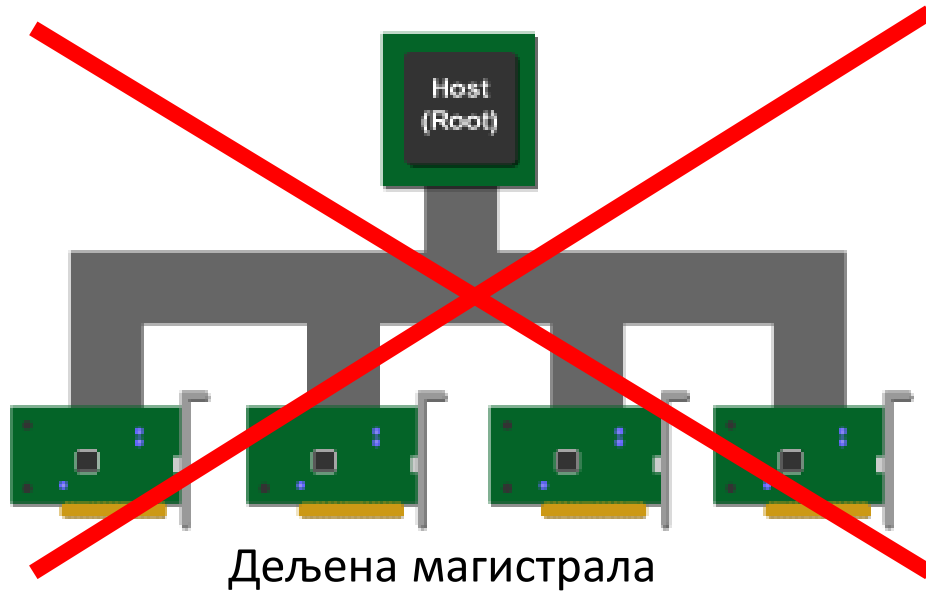
Магистрала PCI-X – пример сегмената



Магистрала *PCI Express*

- Основне карактеристике магистрале *PCI Express*:
 - серијска архитектура
 - комуникација се, у основи, одвија серијски, а не паралелно
 - уместо дељене магистрале, сваки уређај има сопствену везу са прекидачем (*switch*)
 - као да сваки уређај има своју посвећену магистралу

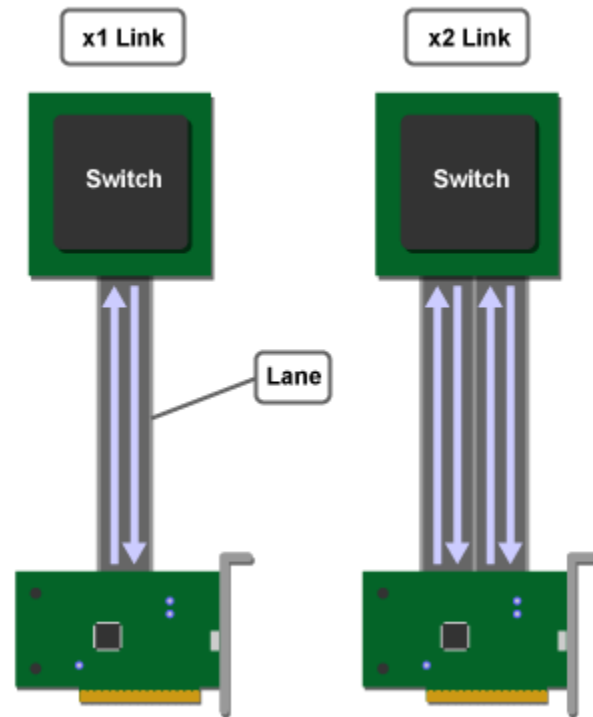
Магистрала *PCI Express* (2)



Магистрала *PCI Express* (3)

- Карактеристични елементи архитектуре:
 - свака веза уређаја и прекидача се састоји од једног или више канала (*lane*)
 - канал је основни носилац комуникације
 - један канал се састоји од две једносмерне линије
 - једна за пренос сигнала од прекидача према уређају
 - једна за пренос сигнала од уређаја према прекидачу
 - дуплекс
 - омогућена је истовремена комуникација у оба смера

Магистрала *PCI Express* (4)



Канали и линије

Магистрала *PCI Express* (5)

- Основне карактеристике магистрале *PCI Express*:
 - 2.5 GHz
 - до 256 MB/s кроз један канал у једном смеру
 - до 512 MB/s кроз један канал у два смера (дуплекс)
 - брзина се може подизати додавањем канала
 - x1, x2, x4, x8, x16, x32 (до 16 GB/s)
- условна компатибилност са *PCI* магистралом
 - мост према магистралаи *PCI* се повезује као један уређај на магистралаи *PCI Express*