

Увод у организацију и архитектуру рачунара 2

Александар Картељ

kartelj@matf.bg.ac.rs

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

Функција система прекида

- Систем прекида представља један од механизма за управљање током рада процесора
 - поред програмских позива и скокова
- У основне примене система прекида улазе
 - комуникација са У/И уређајима
 - употреба основних услуга ОС-а

Начин рада прекида

- Прекиди се могу изазвати хардверски или софтверски
- По наступању прекида, тренутно извршавани програм се прекида и прелази се на контролну процедуру која се назива *рутина за обраду прекида (interrupt service routine)* или *опслужилац прекида (interrupt handler)*
- Када се опслуживање прекида доврши, програм наставља са извршавањем од места на коме је прекинут
 - слично као у случају позивања потпрограма

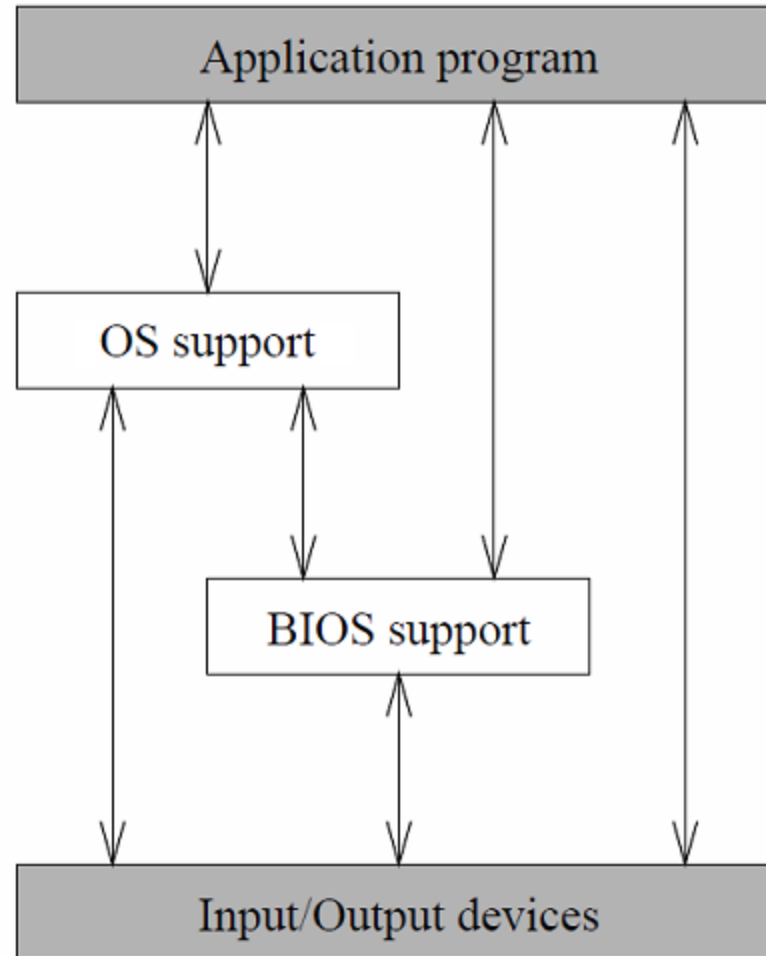
Хардверски прекиди

- Производе се од стране спољашњих уређаја
- Називају се и *неплански* или *асинхрони* прекиди
- Служе за *скретање пажње* процесора на одговарајући уређај
- Постоје две врсте хардверских прекида
 - Маскирајући прекиди
 - Могу да се одложе до неког специфичног тренутка
 - Нпр, ако прекид наступи током обраде претходног прекида
 - Немаскирајући прекиди
 - Одмах се обрађују од стране процесора

Софтверски прекиди

- Називају се и *плански* или *синхрони* прекиди
- Већина процесора подржава софтверске прекиде
 - *Intel x86* – инструкција *int*
 - *PowerPC* – инструкција *sc (system call)*
 - *MIPS* – инструкција *syscall (system call)*
- У основне намене софтверских прекида спадају:
 - приступање У/И уређајима
 - коришћење услуга ОС-а или *BIOS*-а

Комуникација са У/И уређајима



Изузеци

- Посебна подврста хардверских прекида су *изузеци (exceptions)*
- Механизам прекида се употребљава и за обраду препознатих неисправности пи извршавању програма
 - на пример, дељење нулом и сл.
- Неки процесори праве разлику између прекида и изузетака (*Intel x86*) а неки не (*MIPS, PowerPC*)

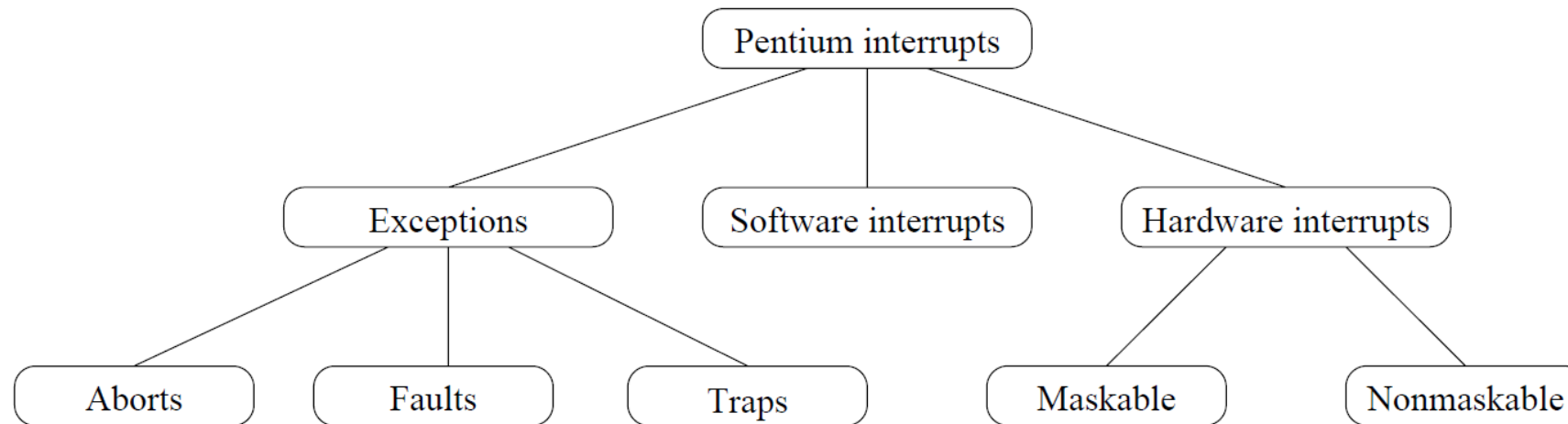
Начин позивања опслужилаца

- Остварује се на два основна начина:
 - Векторски прекиди
 - сваком типу прекида је додељена одговарајућа адреса у меморији
 - у случају наступања прекида, извршавање се наставља од те адресе
 - *Intel x86, PowerPC*
 - Прекиди са узроком
 - прекиди наступају уз означавање узрока прекида
 - сви прекиди се обрађују истим опслужиоцем прекида
 - опслужилац проверава регистар узрока и на основу његове вредности препознаје врсту прекида и преноси контролу одговарајућем коду ОС-а
 - *MIPS*

Пример *Intel x86*

- Систем прекида ће бити описан на примеру фамилије процесора *Intel x86*

Врсте прекида проц. *Intel x86*



Обрада прекида

- Прекиди се означавају бројевима 0-255
- Адресе опслужилаца се налазе у табlici дескриптора прекида (*interrupt descriptor table – IDT*)
 - Ставке табlice су величине по 8 бајтова и представљају 64-битну адресу опслужиоца
 - Таблица се може налазити било где у меморији, а њена адреса мора бити уписана у регистар процесора *IDTR*
 - Постоје посебне инструкције за пуњење и чување табlice
- Број прекида се користи као индекс табlice

Пример софтверског прекида

- Софтверски прекиди се праве инструкцијом *int N*
- И додатним садржајем одговарајућих регистара процесора.
- На пример, инструкција *int 21H* прави прекид који позива одговарајућу функцију ОС-а *DOS*:
 - Поред овога, потребно је навести код операције у регистру *AH*
 - На пример, *AH=06H* означава непосредан конзолни У/И
 - Има две придружене подфункције: унос са тастатуре и приказ унетог карактера
 - Подфункција се спецификује другим регистром: *DL = FFH*
 - проверава се да ли на улазу постоји неки карактер
 - У регистар *AL* се уписује повратна вредност, односно ASCII код унетог карактера

Изузеци (*Intel x86*)

- Деле се на
 - грешке (*fault*)
 - грешке се јављају при стању које је претходило извршавању инструкције која је произвела грешку
 - након обраде прекида, инструкција ће бити поновљена
 - пример: промашај странице или сегмента, јер инструкција мора бити поновљена како би се приступило податку који је сада, након прекида унутар меморије.
 - замке (*trap*)
 - замке се јављају са условима након извршавања инструкције која је произвела грешку
 - након обраде прекида, биће извршена наредна инструкција
 - пример: прекорачење, кориснички прекиди
 - заустављања (*abort*)
 - пријављују озбиљне проблеме, могуће је да не може ни бити препозната инструкција која је произвела прекид
 - једино што опслужилац може да уради јесте да прекине процес
 - пример: хардверске неисправности, неконзистентне вредности системских таблица и сл.

Примери изузетака

- Неки изузеци су унапред дефинисани
 - 0 – грешка дељења
 - 1 – један-корак
 - ако је постављена заставица замке ($TF=1$), после сваке извршене инструкције се прави ова замка
 - користи се за дебаговање
 - 2 – тачка прекида
 - тачка прекида у коду, за дебаговање
 - 3 – прекорачење

Хардверски прекиди (*Intel x86*)

- Немаскирајући прекид се изазива довођењем сигнала на ножицу *NMI* процесора
 - процесор увек одмах реагује на овај прекид
 - он се не може софтверски маскирати
 - његов тип (број) је 2
- Сви остали прекиди су маскирајући
 - изазивају се довођењем сигнала на ножицу *INTR* процесора
 - процесор реагује само ако је заставица *IF = 1* (*interrupt enable flag*)
 - значи, може се софтверски маскирати

Препознавање типа прекида

- Уређај поставља сигнал *INTR*
- Процесор реагује започињањем секвенце прихватања прекида
 - као део те секвенце, шаље сигнал *INTA* уређају
- Уређај реагује на сигнал *INTA* постављањем идентификације (броја, типа) прекида на магистралу података
 - у случају фамилије x86 тип прекида је 8-битни број

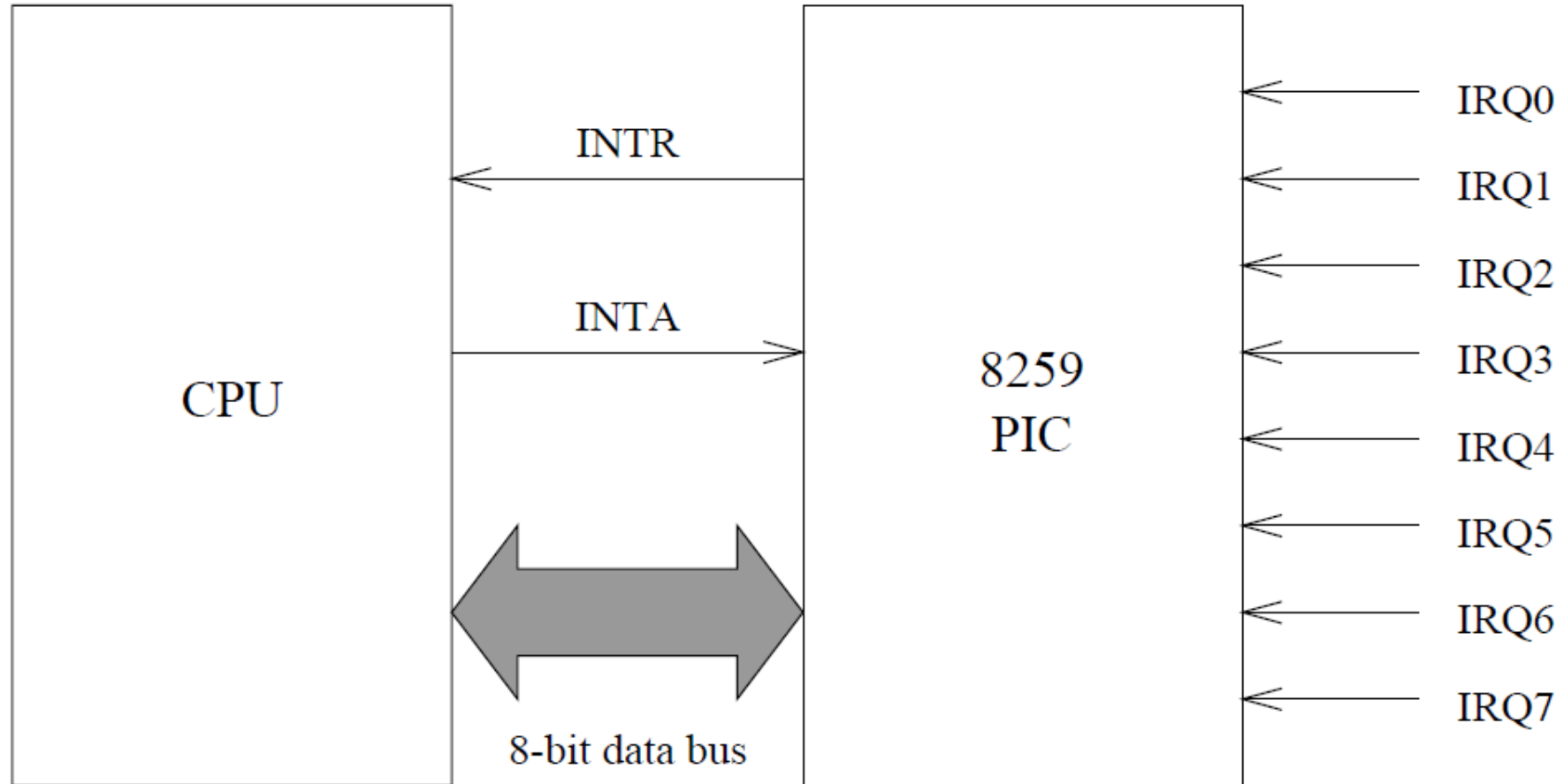
Контролери прекида

- Ако је више уређаја у прилици да изазива прекиде, постоји могућност да више прекида настане у исто време
- Проблем је у томе што се сви прекиди пријављују на исти начин и путем истих сигнала
- То се решава помоћу додатног уређаја – контролера прекида
- Контролер прекида служи као посредник између уређаја и процесора при изазивању прекида

Програмибилни контролер прекида *PIC* 8259

- Пример контролера прекида је чип *Intel PIC 8259*
- Може да опслужује до 8 уређаја
 - сваки уређај има по једну линију према контролеру, којом покушава да изазове прекид
 - са процесором се везује преко линија *INTR* и *INTA* и магистрале података
- У случају да више уређаја истовремено затражи прекид, контролер серијализује те прекиде на основу приоритета

PIC 8259



Програмирање PIC 8259

- PIC 8259 има два регистра
 - 8-битни командни регистар (*ICR*)
 - користи се за програмирање контролера
 - подразумевани приоритети су од навишег: 0,1,...,7
 - 8-битни регистар маске (*IMR*)
 - користи се за допуштање и онемогућавање појединачних прекида
- Програмирање се остварује комуникацијом процесора са контролером
 - путем магистрале података и адреса:
 - 20H – за регистар *ICR*
 - 21H – за регистар *IMR*
- Типови прекида се додељују одређивањем најнижег типа
 - то је тип прекида 0
 - остали типови су у распону +1 до +7

Маскирање прекида и *PIC 8259*

- Блокирање свих прекида се остварује постављањем заставице процесора $IF=0$
- Блокирање појединачних прекида се остварује постављањем регистра *IMR* контролера
 - бит 0 – прекид је омогућен
 - бит 1 – прекид је онемогућен
- На пример, омогућавање само прекида 0 се постиже инструкцијама:

```
mov  AL, 0FEN
out  21H, AL
```

Опслуживање прекида и PIC 8259

- Сваки пут при завршетку обраде прекида који су изазвани посредством контролера, неопходно је јавити контролеру да је обрада завршена
- То се ради посредством командног регистра, слањем кода *20H* (назива се *EOI – end of interrupt*), који означава крај обраде
- Након тога контролер може да настави са изазивањем нових прекида
- Пример:

```
mov  AL, 20H
out  20H, AL
iret
```