

Улаз/излаз



Садржај

- Основни појмови
- Структура улазно/излазног уређаја
- Приступ подацима на уређају
- DMA контролер



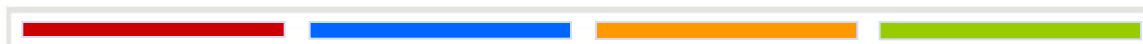
Типови уређаја

- Уређаји са поједначним приступом
 - Миш, неке тастатуре, џоистик
- Уређаји са блоковским приступом
 - Диск, трака, CD, DVD
 - мрежа
- Тајмери
 - интерни, екстерни
- Графички
 - GUI, игре
- Мултимедија
 - Аудио, видео
- Остали
 - Сензори, контролери

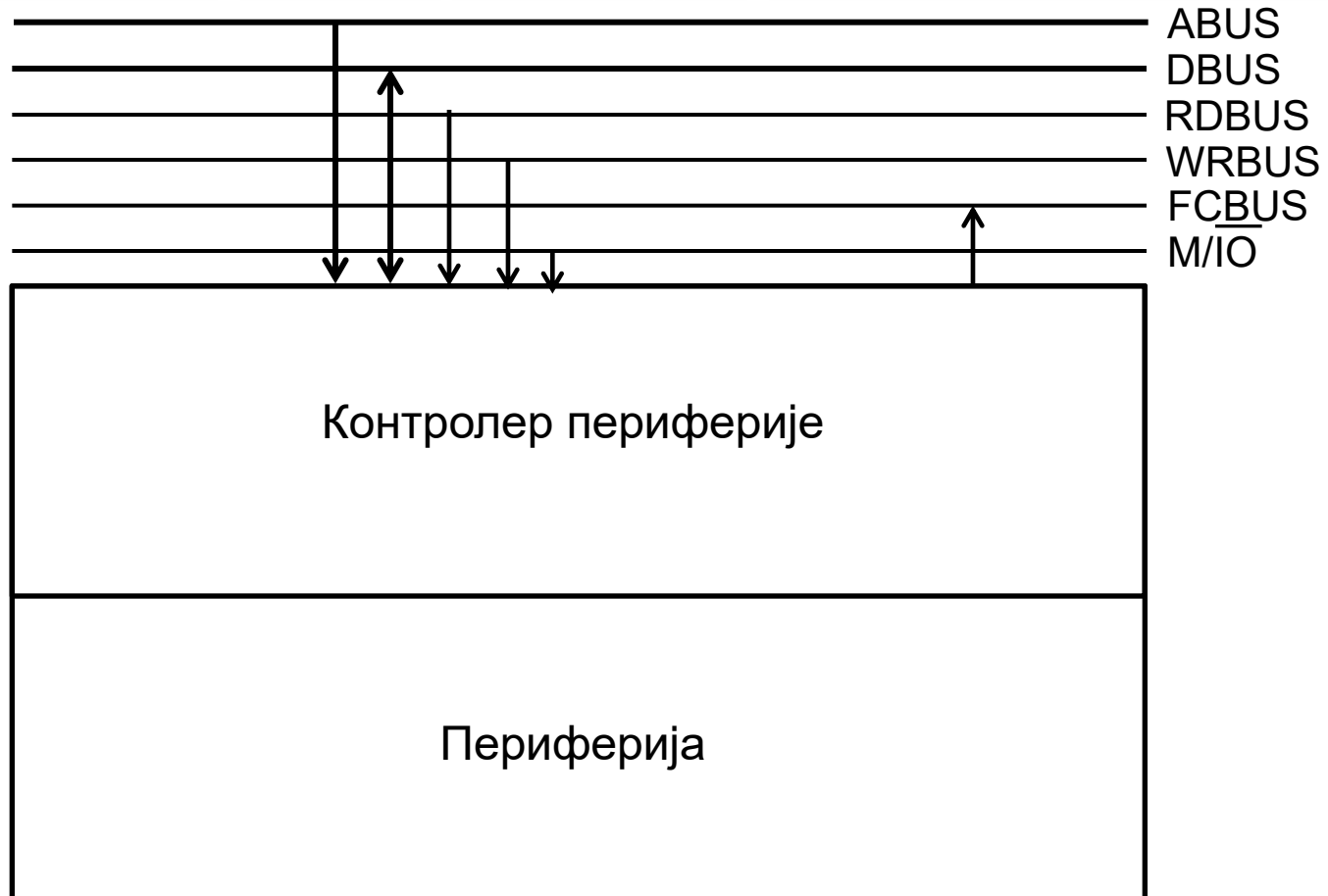


Основни појмови

- Постоји велики број различитих улазних и излазних уређаја. Они се разликују:
 - по томе како се информације у њима смештају,
 - по начинима како са њих долазе и
 - по начинима како се у њих шаљу информације и
 - по брзини са којом се то ради.
- Рад са улазно/излазним уређајима се реализује **једнообразно**, без обзира на то о којој се врсти улазног или излазног уређаја ради.
- Да би се то омогућило уређаји се тако реализују да се састоје од:
 - контролера периферије и
 - периферије.



Основни појмови



Регистри контролера периферије

- Контролери периферија садрже програмски доступне регистре и управљачку логику.
- Програмски доступни регистри служе да се комплетна организација улаза/излаза на програмском нивоу сведе на уписивање у ове регистре и читање ових регистара.
- Управљачка логика служи да се на основу садржаја програмски доступних регистара организује читање података из улазне периферије и упис података у излазну периферију.



Регистри контролера периферије

- Програмски доступни регистри се, према својој функцији, могу сврстати у четири групе и то:
 - управљачки регистар,
 - регистар податка,
 - статусни регистар и
 - регистар броја улаза.

Може да их буде више,
не мора само по један



Управљачки регистар - Control

- Управљачки регистар служи да се уписивањем одговарајућих вредности **програмским путем** у овај регистар изврши:
 - иницијализација,
 - стартовање и
 - заустављање контролера периферије.

Процесор,
на основу програма
- Управљачка логика на основу садржаја одређених битова управљачког регистра креће са преносом података из улазне периферије у регистар податка или из регистра податка у излазну периферију.



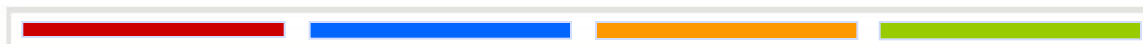
Управљачки регистар - битови

- start—бит који означава да ли је контролер периферије стартован (1) или је заустављен (0).
- i/o—бит који одређује да ли дати улазно/излазни контролер треба да ради у режиму **улаза** или **излаза**.
- enable—бит који означава да контролер треба да генерише прекид како би означио спремност за операцију (1) или да ће процесор програмским путем проверити спремност (0).

Излаз (упис) на периферију, не са ње

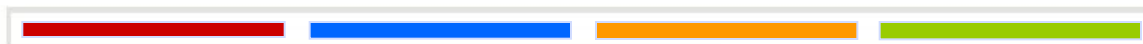
Улаз (читање) са периферије, не на њу

Гледа се у односу на процесор



Управљачки регистар - битови

- Поред наведених битова *start*, *i/o* и *enable*, управљачки регистар *Control* може да има и друге битове.
- Број и значење тих битова зависи од карактеристика периферије и могућих режима рада са периферијом.
- Они се користе да се приликом стартовања контролера периферије уписивањем одговарајућих вредности у битове управљачког регистра *Control* зада и жељени режим рада са периферијом.
- Бит *start* не мора да постоји већ је могуће да се приликом сваке операције она задаје!



Регистар података - Data

- Регистар у који се смешта податак који се пребацује између периферије и процесора.
- Уколико је периферија покренута као улазна онда периферија уписује вредност у овај регистар, а процесор чита регистар.
- Уколико је периферија покренута као излазна онда процесор уписује вредност у овај регистар, а периферија чита регистар.
- Нема унутрашњу структуру



Статусни регистар - Status

- Статусни регистар је регистар контролера периферије који означава које је тренутно стање периферије.
- У овај регистар само контролер периферије уписује одговарајућу вредност.
- Процесор само чита овај регистар, не може да уписује.



Статусни регистар - битови

- `ready`—бит који означава да је контролер спреман за задату операцију.

Ситуације: покренута улазна периферија

1. `ready=0`. Пренос податка из периферије у регистар `Data` контролера периферије је у току.
Процесор, чека, не сме да чита податак из *Data* регистра
2. `ready=1`. Пренос податка из периферије у регистар `Data` контролера периферије је завршен.
Процесор сме да чита податак из *Data* регистра

Статусни регистар - битови

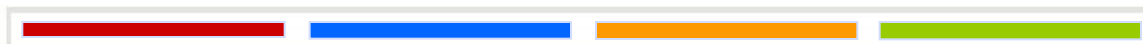
Ситуације: покренута излазна периферија

1. $ready=0$. Пренос претходног податка из *Data* регистра контролера на периферије је у току. Процесор, чека, не сме да упише нови податак у *Data* регистра
2. $ready=1$. Пренос претходног податка из *Data* контролера периферије на периферију је завршен. Процесор сме да упише нови податак у *Data* регистра



Статусни регистар - битови

- Поред наведеног бита *ready* статусни регистар *Status* може да има и друге битове.
- Број и значење тих битова зависи од броја и врсте статусних сигнала којима периферија даје информације контролеру о могућим нерегуларностима у раду периферије, као што је информација да нема папира, да напајање периферије није укључено и слично.
- Управљачка јединица повремено чита вредности статусних сигнала који долазе из периферије и њихове вредности уписује у одговарајуће разреде статусног регистра *Status*.

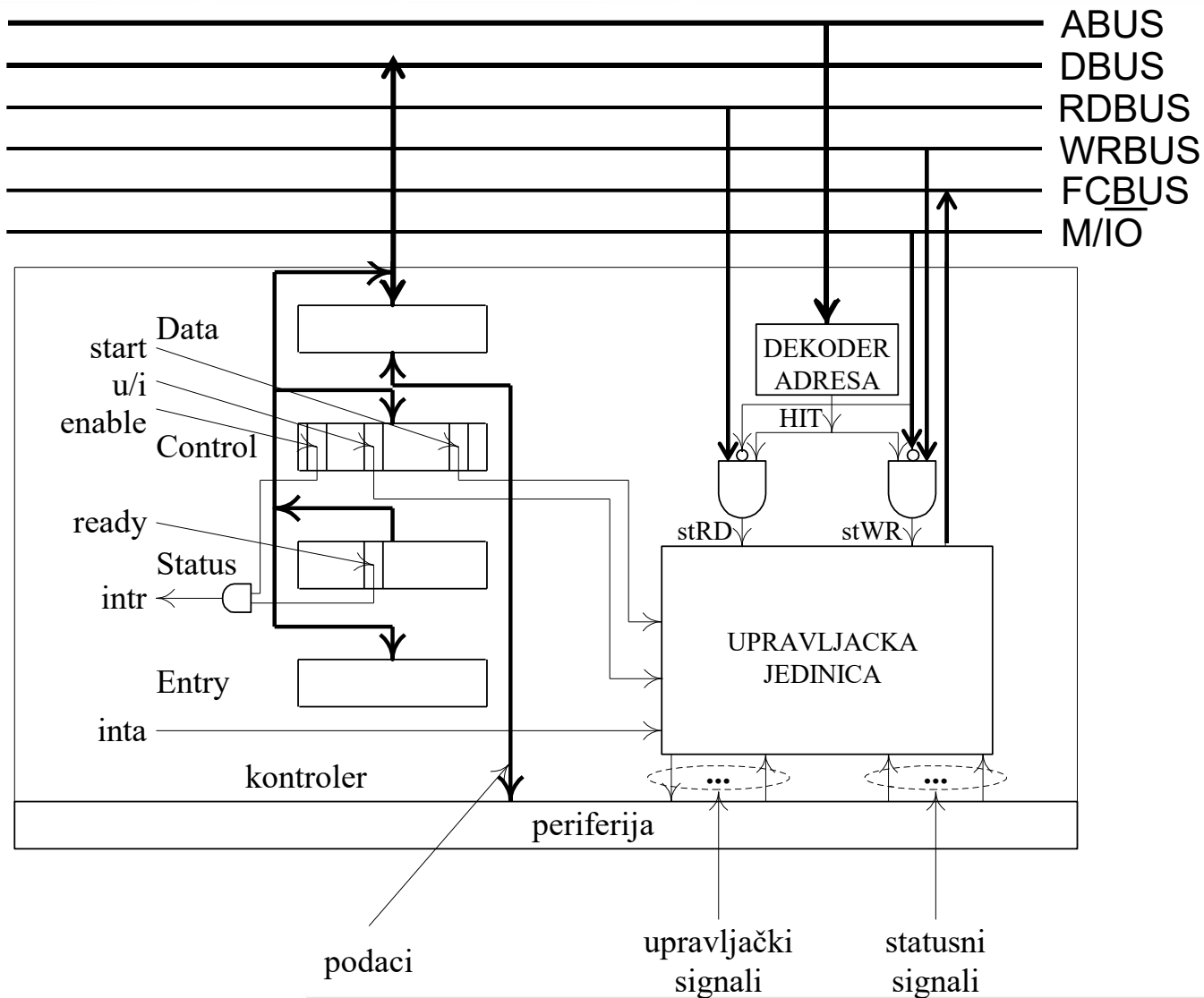


Регистар броја улаза података - *Entry*

- Регистар у коме се чува број улаза у табелу прекидних рутина.
- Приликом иницијализације система програмским путем се у регистар *Entry* уписује број улаза у IV табелу у коме се налази адреса прекидне рутине дате периферије.
- Као реакција на сигнал прекида **intr** процесор у оквиру опслуживања прекида шаље контролеру вредност 1 сигнала потврде **inta** који се користи у контролеру да се процесору као број улаза у IV табелу пошаље садржај регистра *Entry*.
- Нема унутрашњу структуру



Контролер без директног приступа меморији



Технике рада са IO уређајима

- Програмирани улаз/излаз
 - Испитивање бита спремности
 - Коришћење прекида
- Коришћење DMA контролера



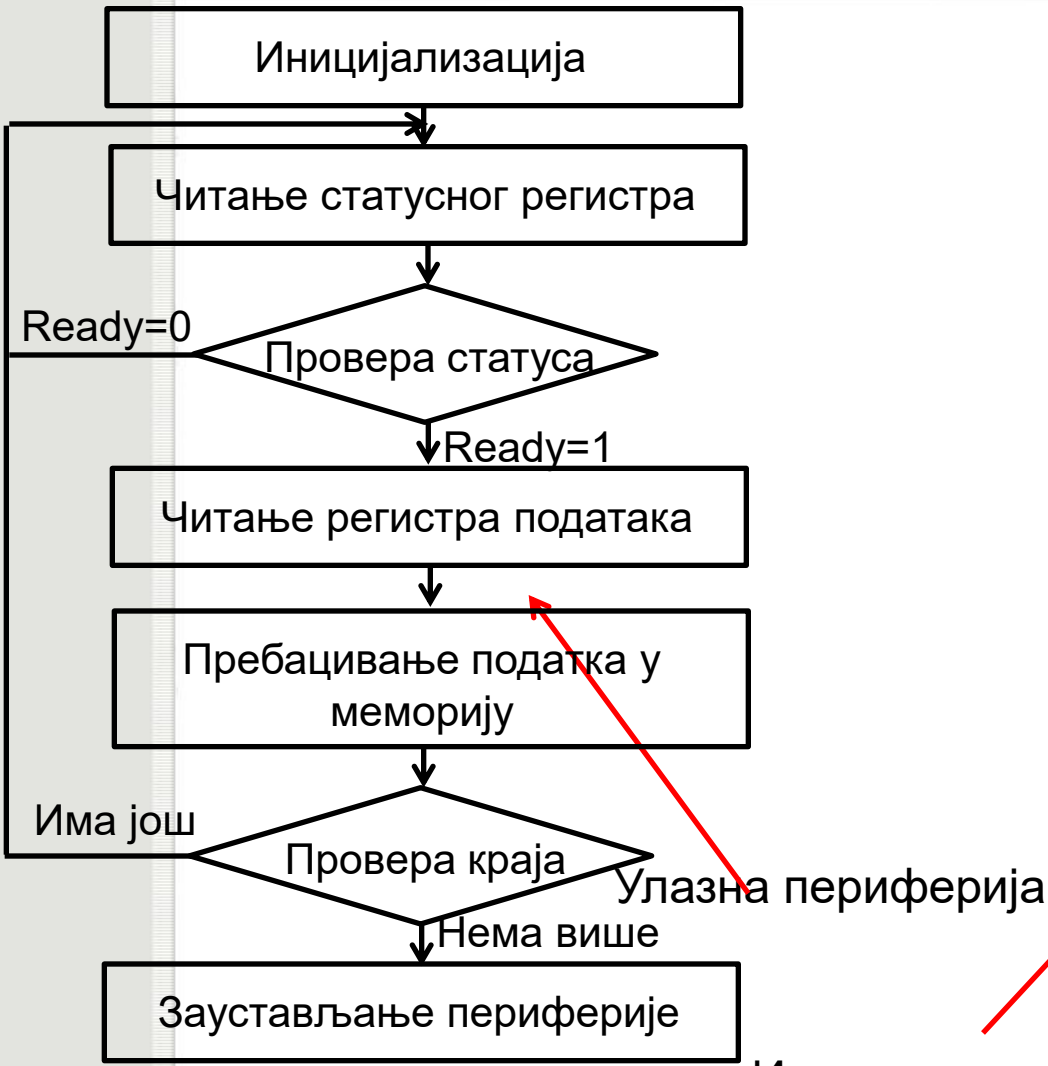
Испитивање бита спремности

- Основни алгоритам рада се састоји из корака:
 - Сваки пут када је процесору потребан податак приступа периферији и проверава да ли је податак доступан.
 - Уколико је податак доступан чита га.
 - Уколико податак није доступан чека.

Разматра се улазна периферија,
слично је и за излазну

- Добра страна је што процесор има пуну контролу када је податак доступан.
- Лоша страна је што троши доста времена на проверу да ли је податак доступан

Испитивање бита спремности



Иницијализација

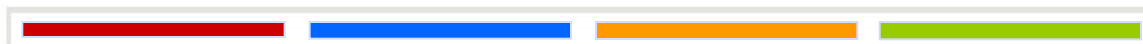
Приликом иницијализације потребно је поставити:

- Бројач који каже колико је података потребно пребацити
- Адресу у меморији почев од које се смештају учитани подаци
- Поставити мод рада периферије:
 - start=1 (укључење периферије)
 - i/o=режим улаза
 - enable=0 (нема прекида)



Читање статусног регистра

- Сваком од регистара контролера периферије се додељује адреса помоћу које му се приступа.
- Уколико се ради о меморијски мапираном улазно/излазном адресном простору приступа се користећи инструкције:
 - LOAD – читање из регистра
 - STORE – упис у регистар
- Уколико се ради о раздвојеном меморијском и улазно/излазном адресном простору приступа се користећи инструкције:
 - IN – читање из регистра
 - OUT – упис у регистар



Провера статуса

- Приликом провере статуса периферије проверава се да ли је периферија припремила нови податак, што је индицирано постављањем бита Ready на вредност 1.
- Да би се ово проверило потребно је из очитаног регистра издвојити само бит Ready од осталих бита.
- Издвајање једног бита се обавља користећи инструкцију AND (TST) и непосредну величину која има вредност 1 на оној позицији на којој се налази бит Ready, а на свим осталим вредност 0.
- Издвајање је неопходно јер остали бити статусног регистра могу имати различита значења.

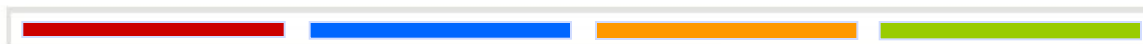
Читање регистра података

- На исти начин као и статусни регистар чита се садржај регистра података и пребацује се у неки од регистара процесора.
- Читањем податка сигнализира се контролеру и периферији да пређу на генерисање новог податка.



Пребацивање податка у меморију

- Податак се из регистра процесора пребацује на прву наредну слободну адресу
- Помера се показивач на следећу слободну адресу
- Умањује се бројач који каже колико је података још потребно пребацити
(или се увећава бројач пребачених података)



Провера краја

- Проверава се да ли су сви подаци прочитани са периферије.
- Ово се постиже провером да ли је бројач преосталих података доспео до 0.
- Уколико није доспео до 0 прелази се на проверу статусног регистра како би се прихватио нови податак.



Заустављање периферије

- Заустављање периферије се постиже постављањем мода рада на:
 - `start=0` (искључење периферије)
 - `i/o` не мора да се мења
 - `enable` не мора да се мења

Главни програм - in

```
MOV #blockadr, R1
MOV #blockcount, R2
MOV #modestart, R3
OUT R3, Control
LOOP: IN Status, R3
      AND R3, #ready
      JZ LOOP
      IN Data, R3
      MOV R3, (R1)
      INC R1
      DEC R2
      JNZ LOOP
      MOV #modestop, R3
      OUT R3, Control
```

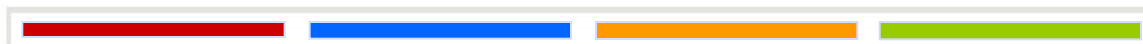
Главни програм - out

```
MOV #blockadr, R1
MOV #blockcount, R2
MOV #modestart, R3
OUT R3, Control
LOOP: IN Status, R3
AND R3, #ready
JZ LOOP
MOV (R1), R3
OUT R3, Data
INC R1
DEC R2
JNZ LOOP
MOV #modestop, R3
OUT R3, Control
```

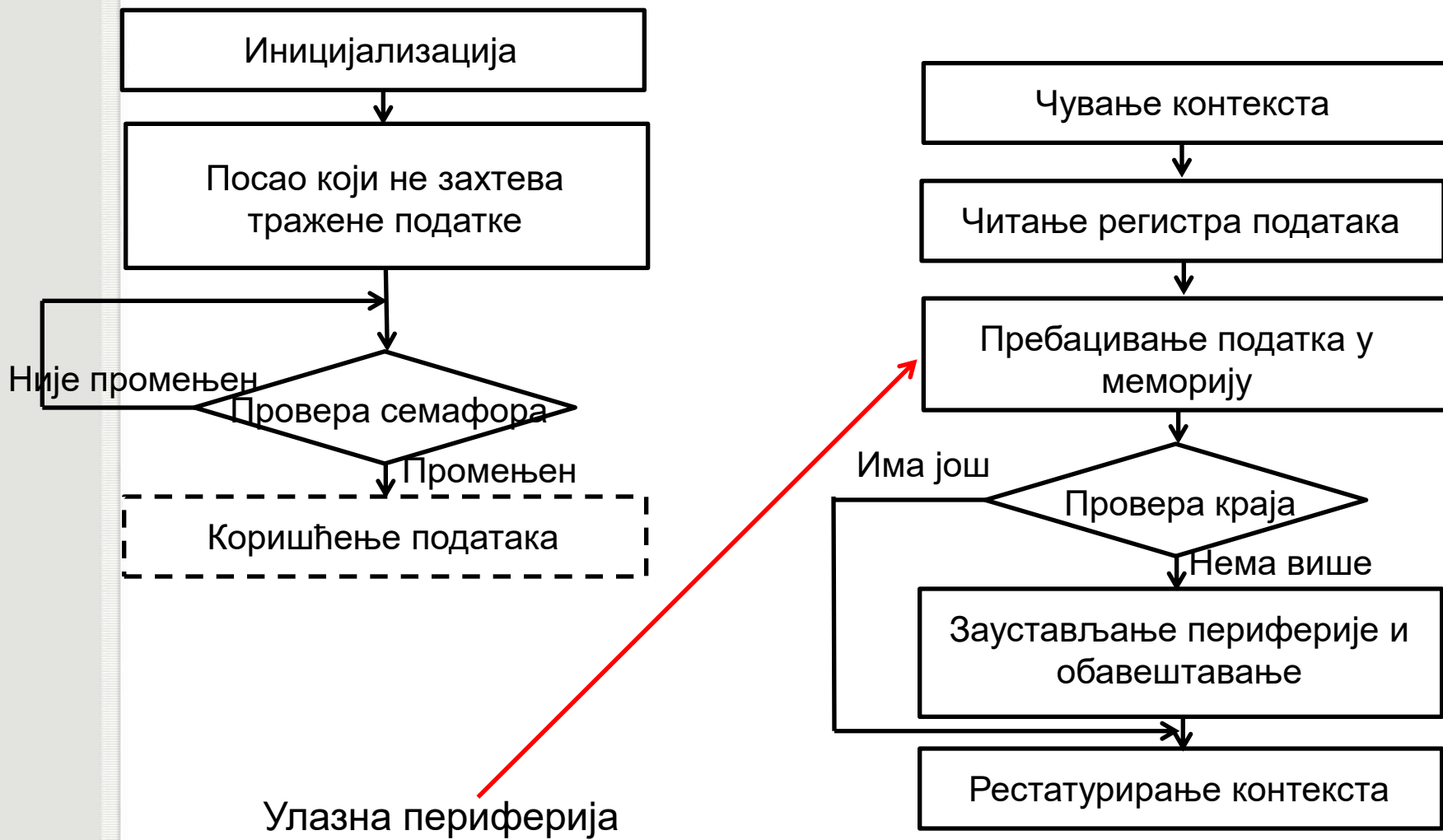
Разлика!

Улаз/излаз генерисањем прекида

- Основни принцип рада се састоји у томе да се покрене контролер периферије који ће процесор унутар прекида да обавештава када буде доступан за операцију, а процесор може нешто друго да ради у међувремену.
- Добра страна је што процесор не мора да троши време на сталну проверу већ може да ради нешто друго у међувремену.
- Лоша страна је што се троши време за чување и рестатурирање контекста приликом скока и повратка из прекидне рутине



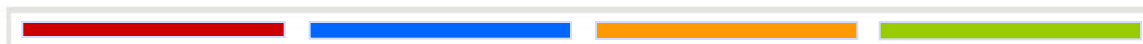
Улаз/излаз генерисањем прекида



Иницијализација

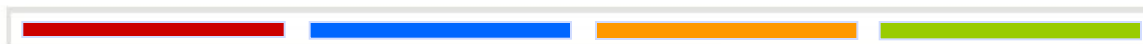
Приликом иницијализације у главном програму потребно је поставити:

- Бројач који каже колико је података потребно пребацити
- Адресу у меморији почев од које се смештају учитани подаци
- **Семафор преко кога се јавља да су обрађени сви подаци**
- Поставити мод рада периферије:
 - start=1 (укључење периферије)
 - i/o=режим улаза
 - enable=1 (има прекида)



Провера семафора

- Приликом провере семафора у **главном програму** проверава се да је дошло до промене вредности на коју је семафор иницијализован.
- Ако је дошло до промене онда су у прекидној рутини обрађени сви подаци
- Провера да ли је вредност иста као иницијална се постиже користећи инструкцију SUB (CMP) и непосредну величину којом је семафор иницијализован.



Чување контекста

- Приликом чувања контекста на почетку прекидне рутине потребно је сачувати све регистре опште намене који ће бити промењени унутар прекидне рутине како би се приликом повратка из прекидне рутине прекинути програм наставио без икаквих последица и знања да је прекинут.



Читање регистра података

- На исти начин као код испитивања бита спремности чита се садржај регистра података и пребацује се у неки од регистара процесора. Ово се ради унутар прекидне рутине.
- Читањем податка сигнализира се контролеру и периферији да пређу на генерисање новог податка.



Пребацивање податка у меморију

- Податак се из регистра процесора пребацује на прву наредну слободну адресу
- Помера се показивач на следећу слободну адресу
- Умањује се бројач који каже колико је података још потребно пребацити
(или се увећава бројач пребачених података)

Провера краја

- Проверава се да ли су сви подаци прочитани са периферије.
- Ово се постиже провером да ли је бројач преосталих података доспео до 0.
- Уколико је доспео до 0 прелази се на заустављање периферије.
- Уколико није доспео до 0 прелази се на рестаурирање контекста.



Заустављање периферије и обавештавање

- Заустављање периферије се постиже постављањем мода рада на:
 - `start=0` (искључење периферије)
 - `i/o` не мора да се мења
 - `enable` не мора да се мења
- Обавештавање се обавља уписивањем вредности у променљиву семафор која је различита од иницијалне вредности.

Рестатурирање контекста

- На крају прекидне рутине потребно је да се са стеку скину сви регистри који су тамо били сачувани на почетку прекидне рутине.

Главни програм - in

MOV #blockadr, mem1

MOV #blockcount, mem2

MOV #modestart, R3

OUT R3, Control

MOV #startVal, sem

! Програм у коме се не користе

! подаци из блока меморије у који

! се уносе подаци са периферије

LOOP: CMP sem, #startVal

JZ LOOP

! Програм у коме се користе

! подаци из блока меморије у који

! су унети подаци са периферије



Прекидна рутина - in

```
PUSH R3
IN Data, R3
MOV R3, (mem1)
INC mem1
DEC mem2
JNZ BACK
MOV #endVal, sem; !startVal <> endVal
MOV R3, #modestop
OUT R3, Control
BACK: POP R3
RTI
```

Главни програм - out

```
MOV #blockadr, mem1
```

```
MOV #blockcount, mem2
```

```
MOV #modestart, R3
```

```
OUT R3, Control
```

```
MOV #startVal, sem
```

```
! Програм у коме се не користи
```

```
! Периферија
```

```
LOOP: CMP sem, #startVal
```

```
JZ LOOP
```

Различита вредност



Прекидна рутина - out

PUSH R3

MOV (mem1), R3

OUT R3, Data

Разлика!



INC mem1

DEC mem2

JNZ BACK

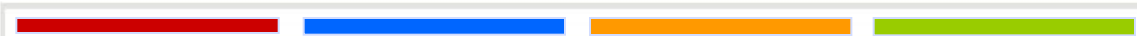
MOV #endVal, sem; !startVal <> endVal

MOV R3, #modestop

OUT R3, Control

BACK: POP R3

RTI



DMA (Direct Memory Access) контролер

- Уређај који омогућава директан трансфер података између периферије и меморије без потребе да се подаци прво пренесу у процесор.
- Подаци који се пребацују најчешће чине блокове.
- Приликом пребацивања сваког податка DMA контролер излази на магистралу и обавља одговарајуће циклусе на магистрали, при чему поставља адресу и остале сигнале на магистрали.
- DMA контролер замењује процесор приликом трансфера података.



DMA контролер

- При преносу из периферије у меморију:
 - чита податак из периферије и уписује у регистар података контролера,
 - податак из регистра контролера уписује у меморијску локацију на адреси одређеној садржајем адресног регистра контролера
- При преносу из меморије у периферију:
 - податак прво чита из меморије са адресе одређене садржајем адресног регистра контролера и уписује у регистар података контролера,
 - податак из регистра података контролера уписује у периферију.



Регистри DMA контролера

- Програмски доступни регистри се, према својој функцији, могу сврстати у групе и то:
 - управљачки регистар,
 - адресни регистар (један или два),
 - бројачки регистар,
 - статусни регистар,
 - регистар податка, и
 - регистар броја улаза.

Иста функција као код контролера периферије

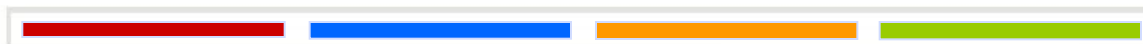
Управљачки регистар - битови

- start—бит који означава да ли је контролер периферије стартован (1) или је заустављен (0).
 - i/o—бит који одређује да ли контролер треба да ради у режиму улаза или излаза.
 - enable—бит који означава да контролер треба да генерише прекид након комплетног трансфера.
 - burst—бит који означава да ли контролер за трансфер сваког податка треба да тражи магистралу (Cycle Stealing) или да једном тражи магистралу коју враћа тек на крају читавог трансфера (Burst)
 - mem—бит који означава да ли се ради трансфер типа меморија-меморија
- опционо



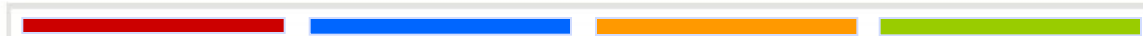
Управљачки регистар - i/o бит

- У случају да контролер ради у режиму улаза контролер дохвата податак са периферије и уписује га у меморију на адресу на коју указује адресни регистар контролера *Adr*.
- У случају да контролер ради у режиму излаза контролер дохвата податак из меморије са адресе на коју указује адресни регистар контролера *Adr* и смешта га на периферију.
- Пренос се реализује онолико пута колика је вредност *Count* регистра.



Управљачки регистар - burst бит

- У случају рада са веома брзом и временски критичном периферијом да би се избегло да контролер тражи дозволу за коришћење магистрале и да тек по добијању дозволе реализује циклус на магистрали контролер се може, уписивањем вредности 1 у бит *burst* управљачког регистра, програмирати да пренос блока података реализује у пакетском режиму рада.
- У овом режиму рада контролер, по добијању дозволе коришћења магистрале, држи магистралу заузету све време док не пренесе цео блок података и тек по завршетку преноса укида захтев за коришћење магистрале.



Појединачни режим-Cycle Stealing

- Пре сваког трансфера контролер најпре постављањем сигнала *hreq* на 1 тражи од процесора дозволу да користи магистралу.
- Тек када му процесор постављањем сигнала *hask* на 1 да дозволу контролер обавља трансфер.
- Све време док траје трансфер контролер држи вредност 1 сигнала *hreq*, а процесор вредност 1 сигнала *hask*.
- По завршетку трансфера контролер укида захтев за коришћење магистрале тако што сигнал *hreq* поставља на 0, а процесор укида дозволу коришћења магистрале тако што сигнал *hask* поставља на 0.

Пакетски режим-Burst

- Пре првог трансфера контролер најпре постављањем сигнала *hreq* на 1 тражи од процесора дозволу да користи магистралу.
- Тек када му процесор постављањем сигнала *hask* на 1 да дозволу контролер обавља трансфер **читавог** блока.
- Све време док траје трансфер контролер држи вредност 1 сигнала *hreq*, а процесор вредност 1 сигнала *hask*.
- По завршетку **читавог** трансфера контролер укида захтев за коришћење магистрале тако што сигнал *hreq* поставља на 0, а процесор укида дозволу коришћења магистрале тако што сигнал *hask* поставља на 0.

Управљачки регистар - *mem* бит

- Уколико бит *mem* има вредност 0, реализује се пренос у зависности од вредности бита *i/o*.
- Уколико бит *mem* има вредност 1, контролер реализује пренос података из меморије у меморију и тада вредност бита *i/o* није битна.
- Постоје два адресна регистра: изворишни (*Adrs*) и одредишни (*Adrd*).
- За сваки трансфер на магистрали се реализују два посебна циклуса и то циклус читања са адресе *Adrs* и циклус уписа на адреси *Adrd*.
- Пренос из меморије у меморију се реализује онолико пута колика је вредност *Count* регистра.

Увећавају се након сваког трансфера

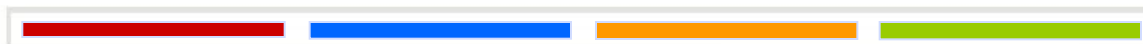


Управљачки регистар - *memset* бит

- Уколико бит *memset* има вредност 1, контролер реализује пренос једног податка из меморије на више адреса у меморију и тада вредност бита *mem* и *i/o* није битна.
- Постоје два адресна регистра: изворишни (*Adrs*) и одредишни (*Adrd*).
- За сваки трансфер на магистрали се реализују два посебна циклуса и то циклус читања са адресе *Adrs* и циклус уписа на адреси *Adrd*.
- Пренос се реализује онолико пута колика је вредност *Count* регистра.

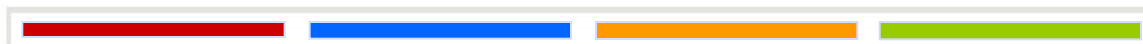
Увећава се након сваког трансфера

Не мења се након сваког трансфера

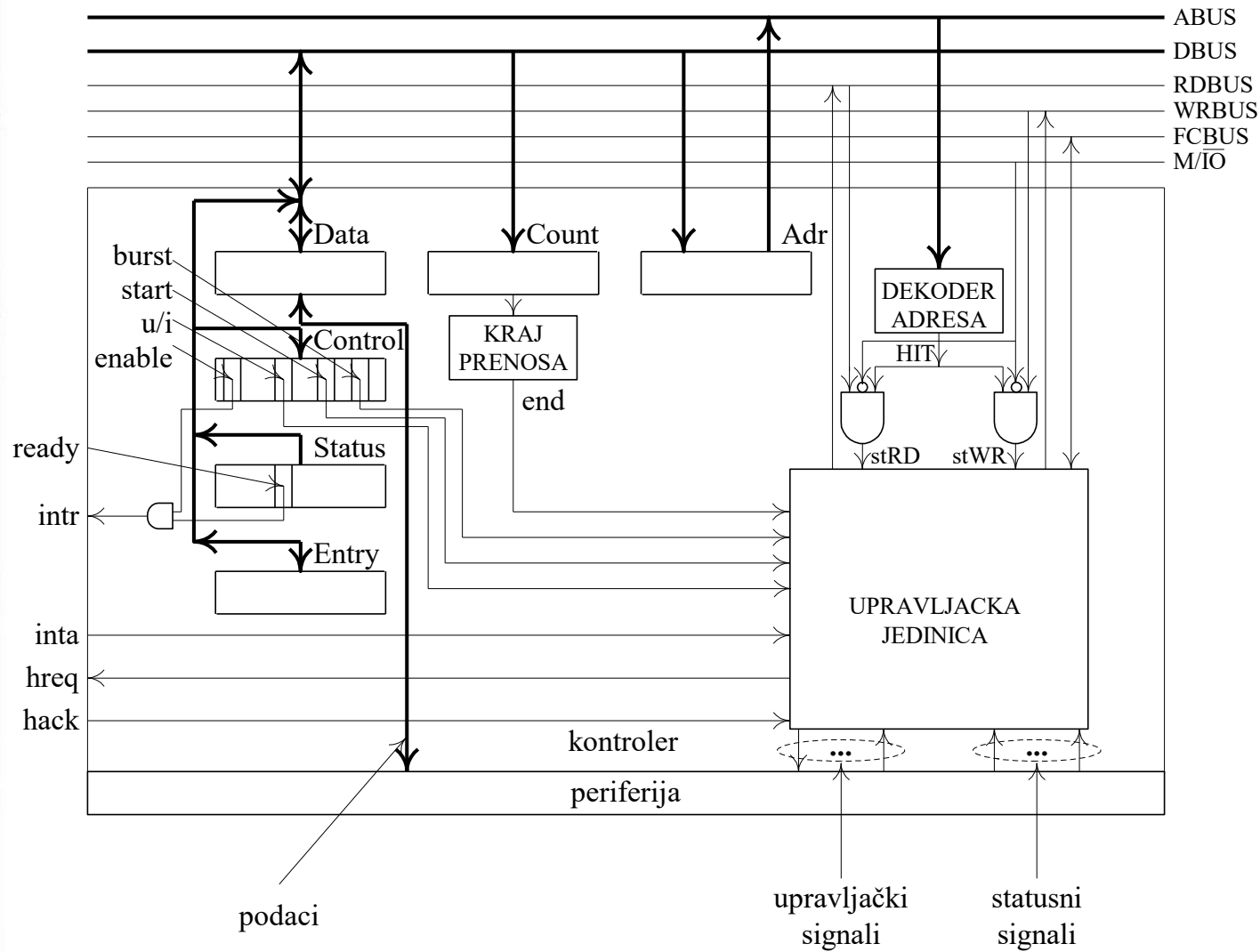


Управљачки регистар - *direction* бит

- Постоји и варијанта контролера код које се може додати режим преноса према вишим и нижим меморијским локацијама.
- Постоји посебан бит *direction* у управљачком регистру *Control*, па се вредношћу 0 овог бита одређује да садржај адресног регистра *Adr*, или адресних регистара *Adrs* и *Adrd*, треба инкрементирати, а вредношћу 1 да треба декрементирати.
- Ово је могуће и за преносе између периферије и меморије и за преносе из меморије у меморију.



DMA контролер

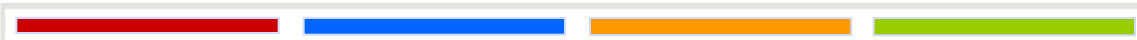
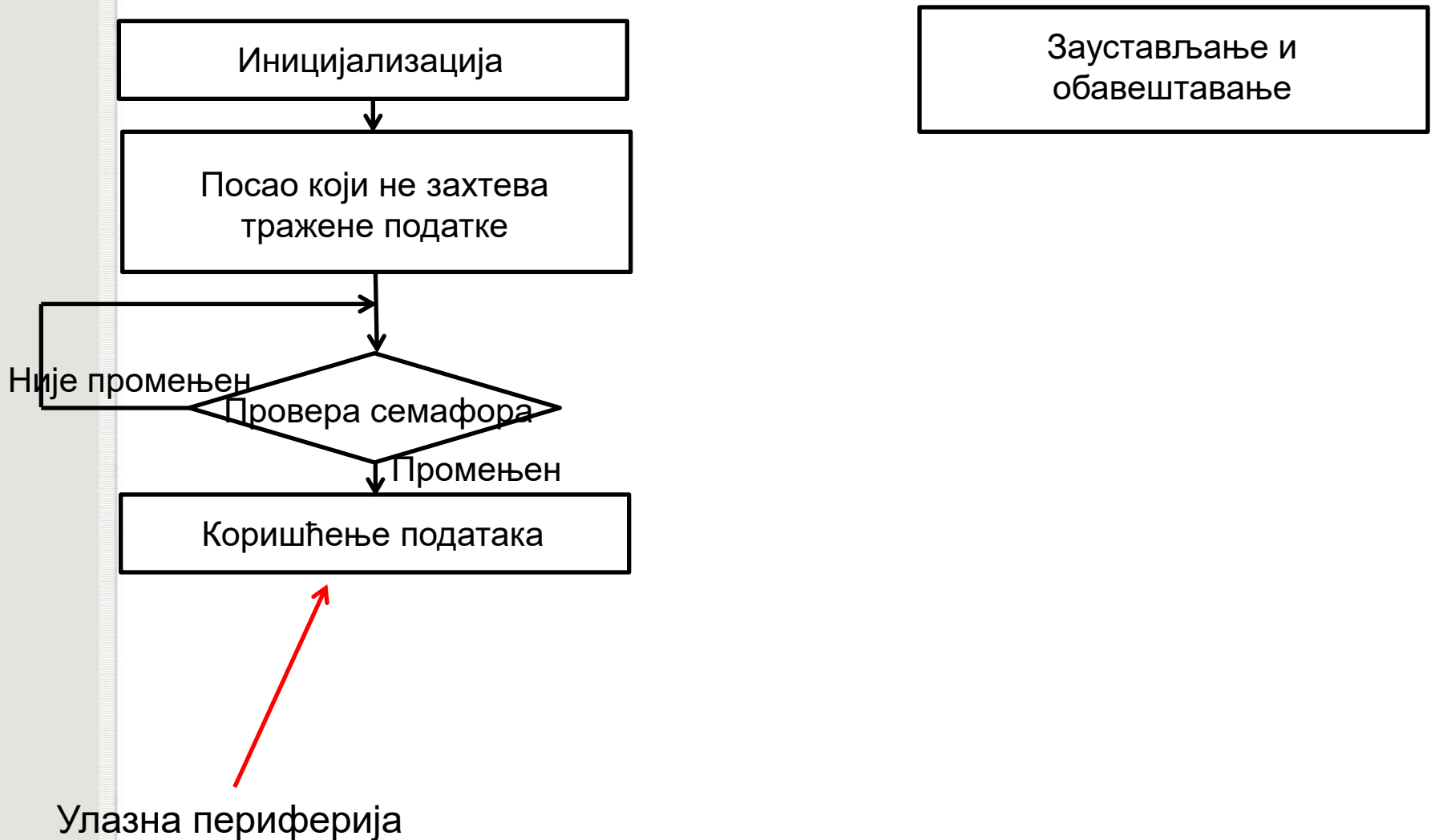


Коришћење DMA контролера

- Основни принцип рада се састоји у томе да се процесор растерети трансфера тако што се покрене DMA контролер који ће обављати трансфер хардверски, а процесор може нешто друго да ради у **паралели**.
- Добра страна је што процесор не мора да троши време на трансфер података већ може да ради нешто друго.
- Лоша страна је што процесор мора да се надмеће за магистралу са DMA контролером што захтева додатни хардвер.



Коришћење DMA контролера



Иницијализација

Приликом иницијализације у главном програму потребно је поставити:

- *Count* регистар који каже колико је података потребно пребацити
- *Adr* регистар са адресом у меморији почев од које се налазе подаци
- Семафор преко кога се јавља да су обрађени сви подаци
- Поставити мод рада DMA контролера:
 - start=1 (укључење периферије)
 - i/o=режим улаза
 - enable=1 (скоро увек јавља прекидом)
 - burst=жељени режим

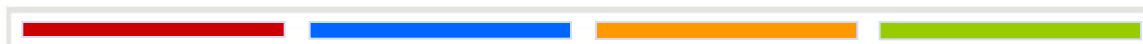
Иницијализација

- Поставити мод рада контролера периферије:
 - start=1 (укључење периферије)
 - i/o=режим улаза
 - enable=0



Провера семафора

- Приликом провере семафора **у главном програму** проверава се да је дошло до промене вредности на коју је семафор иницијализован.
- Ако је дошло до промене то значи да се извршила прекидна рутина у којој је обавештен главни програм да је трансфер завршен.
- Провера да ли је вредност иста као иницијална се постиже користећи инструкцију SUB (CMP) и непосредну величину којом је семафор иницијализован.



Заустављање и обавештавање

- Заустављање периферије се постиже постављањем мода рада DMA контролера на:
 - start=0 (искључење DMA контролера)
 - i/o, enable, burst: не морају да се мењају
- Потребно је поставити и мод контролера периферије:
 - start=0 (искључење периферије)
 - i/o, enable: не морају да се мењају
- Обавештавање се обавља уписивањем вредности у променљиву семафор која је различита од иницијалне вредности.



Главни програм - in/out

MOV #blockadr, R1

OUT R1, Adr

MOV #blockcount, R1

OUT R1, Count

MOV #startVal, sem

MOV #modestart, R1

OUT R1, Control

MOV #modestartDMA, R1

OUT R1, ControlDMA

! Програм у коме се не користе

! подаци из блока меморије у који

! се уносе подаци са периферије

LOOP: CMP sem, #startVal

JZ LOOP



Прекидна рутина – in/out

PUSH R3

MOV #endVal, sem; !startVal <> endVal

MOV R3, #modestop

OUT R3, Control

MOV R3, #modestopDMA

OUT R3, ControlDMA

POP R3

RTI

Питања?

Електротехнички Факултет
Универзитет у Београду

