

Механизм прекида



Садржај

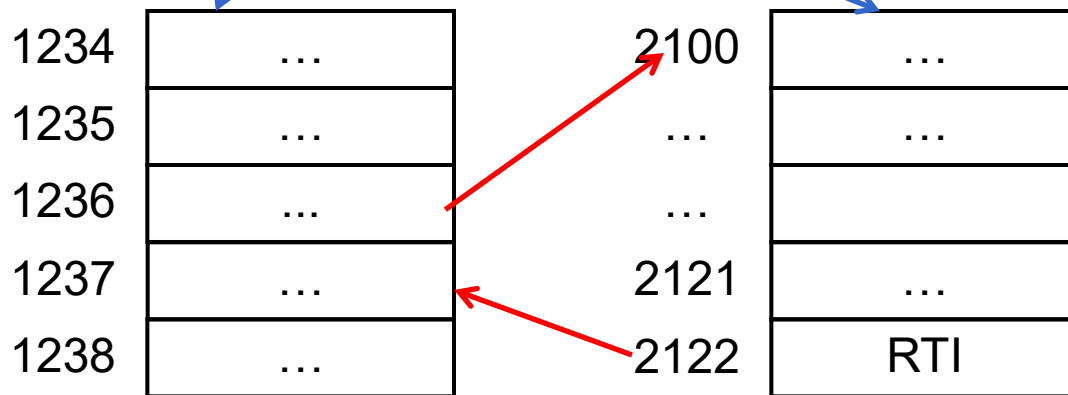
- Увод
- Типови прекида
- Опслуживање захтева за прекид
- Повратак из прекидне рутине
- Приоритети прекида
- Примери



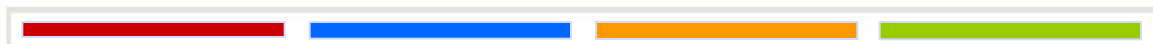
Увод

- Механизам прекида омогућава да се, по генерисању захтева за прекид од стране неког модула рачунара ван процесора или самог процесора, прекине извршавање текућег програма (главни програм), и скочи на нови програм (прекидна рутина).

захтев за прекид →



Извршавања инструкције



Обрада прекида

Када се јави захтев за прекид, текућа инструкција:

- заврши фазу *читање инструкције*, декодовање и *читање операнда*, *извршавање операције* и *сместање резултата* (уколико је то могуће),
- прелази на прихватање захтева за прекид преласком на фазу *обрада прекид* у оквиру које се скаче на прву инструкцију прекидне рутине.



Типови прекида

- Спољашњи прекиди
- Унутрашњи прекиди
- Софтверски прекиди
(понекад се сматрају унутрашњим)



Спољашњи прекиди

1. контролери периферија да би процесору сигнализирани спремност за пренос података (маскирајући прекиди – могу се забранити),
2. уређаји рачунара који контролишу исправност напона напајања, трансфера на магистралама, рада меморије итд.
(немаскирајући прекиди – не могу се забранити),



Унутрашњи и софтверски прекиди

3. процесор, као резултат откривене некоректности у извршавању текуће инструкције (**Fault**) (нелегалан код операције, нелегално адресирање, дељење нулом, page fault, права приступа итд.),
4. процесор, ако је претходно посебном инструкцијом задат такав режим рада процесора, кроз постављање бита *T прекид* после сваке *инструкције* у програмској статусној речи PSW на вредност 1, да се после сваке инструкције скаче на одређену прекидну рутину и
5. процесор, ако је текућа инструкција која се извршава инструкција прекида INT.



Обрада прекида

- Опслуживање захтева за прекид се реализује делом хардверски и делом софтверски.
- Повратак из прекидне рутине софтверски.
- Хардверска реализација дела опслуживања захтева за прекид се остварује извршавањем корака фазе *опслуживање захтева за прекид*.
- Софтверска реализација дела опслуживања захтева за прекид и повратка из прекидне рутине се остварују извршавањем одговарајућих инструкција процесора на почетку и крају прекидне рутине.



Опслуживање захтева за прекид

- Опслуживање захтева за прекид се састоји из:
 - чувања контекста процесора и
 - формирања новог контекста процесора (опционо)

Чување контекста процесора

- Контекст процесора чине програмски бројач PC, програмска статусна реч PSW и преостали програмски доступни регистри.

- Чува се (најчешће на стеку) :

Хардверски

→ програмски бројач PC да би се по повратку из прекидне рутине у главни програм омогућило процесору извршавање главног програма од инструкције на којој се стало и

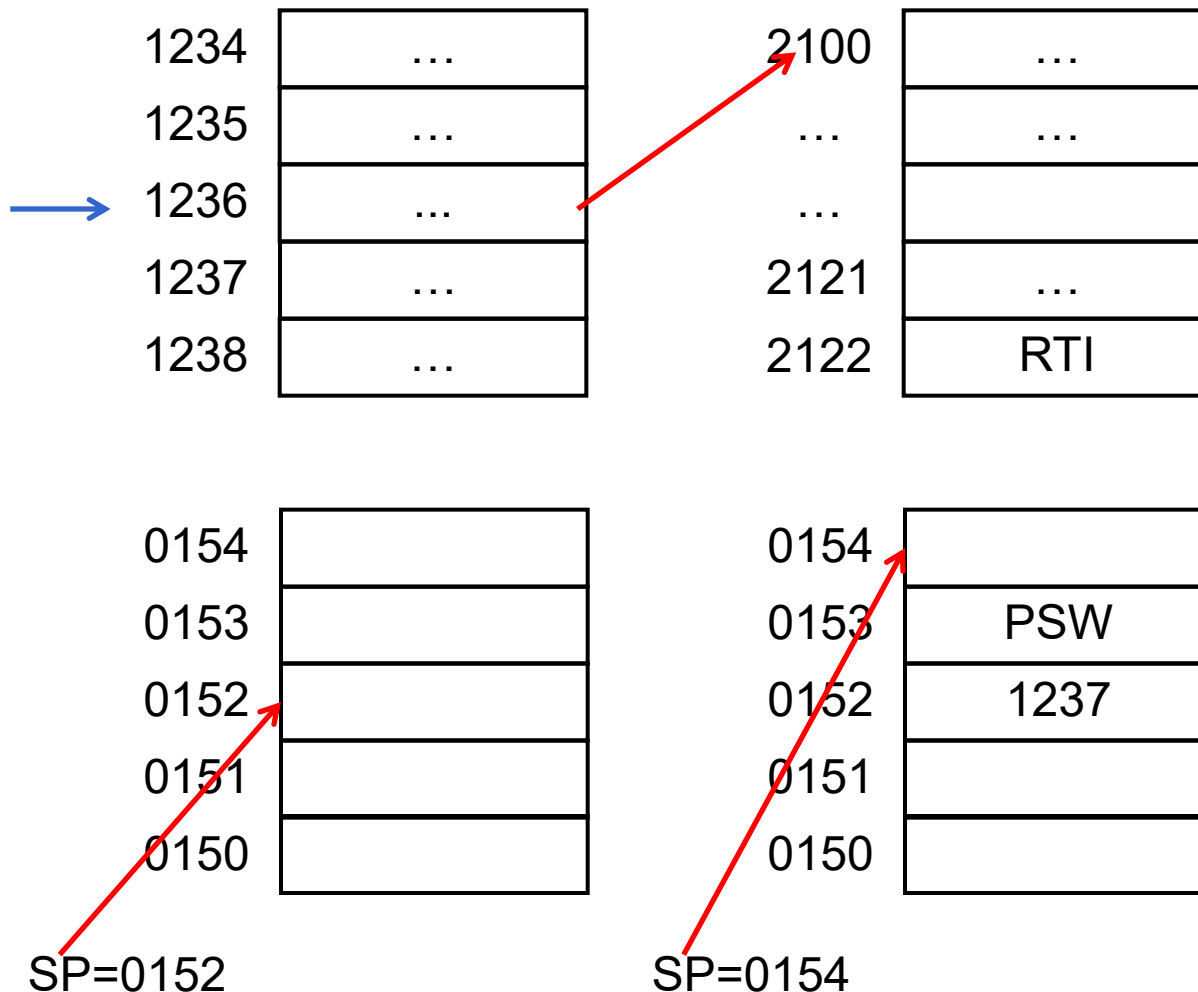
Хардверски најчешће

→ програмска статусна реч PSW и преостали програмски доступни регистри да би се по повратку из прекидне рутине у главни програм у процесору обезбедило исто стање које би било да није било прекида и скока на прекидну рутину.

Софтверски најчешће на почетку прекидне рутине

Чување контекста процесора (пример)

захтев за
прекид



Чување контекста процесора

- Чува се:
 - програмски бројач РС (који??)
- У случају спољашњих маскирајућих (1) и немаскирајућих (2), trap (4) и int (5) чува се адреса прве наредне инструкције након оне прекинуте
- У случају унутрашњих **Fault** прекида (3) чува се адресе инструкције **која је прекинута** како би се сигнализирало оперативном систему

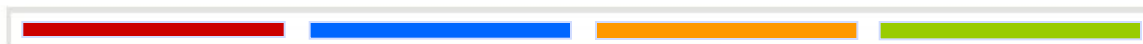


Формирање новог контекста

- Приликом формирања новог контекста процесора обавља се:
 - Утврђивање адресе прекидне рутине
 - Ажурирање програмске статусне речи (PSW)

Утврђивање адресе прекидне рутине

- Утврђивање адресе прекидне рутине се реализује хардверски.
- Утврђивање адресе прекидне рутине се реализује на основу:
 - садржаја табеле адреса прекидних рутина (IV табела)
 - броја улаза у IV табелу.
- IV табела са адресама прекидних рутина за све врсте прекида се налази у меморији.
- Од адресе на коју указује садржај регистра процесора IVTP (*Interrupt Vector Table Pointer*).
- $PC \leftarrow IV[\text{broj_ulaza}]$

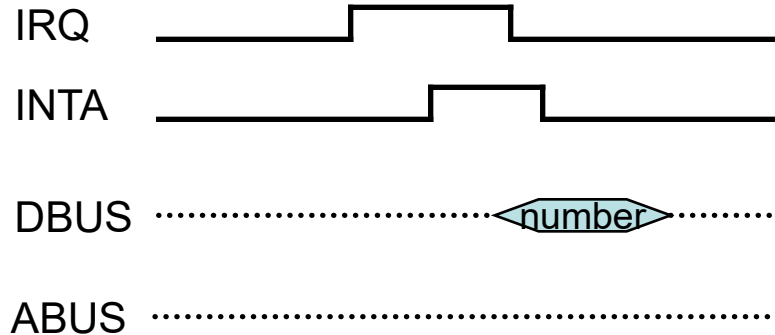
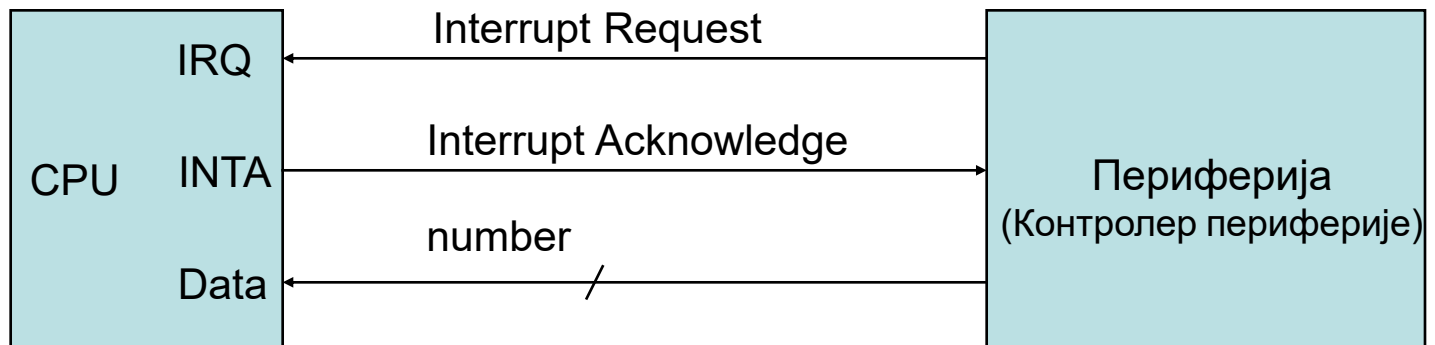


Добијање броја улаза

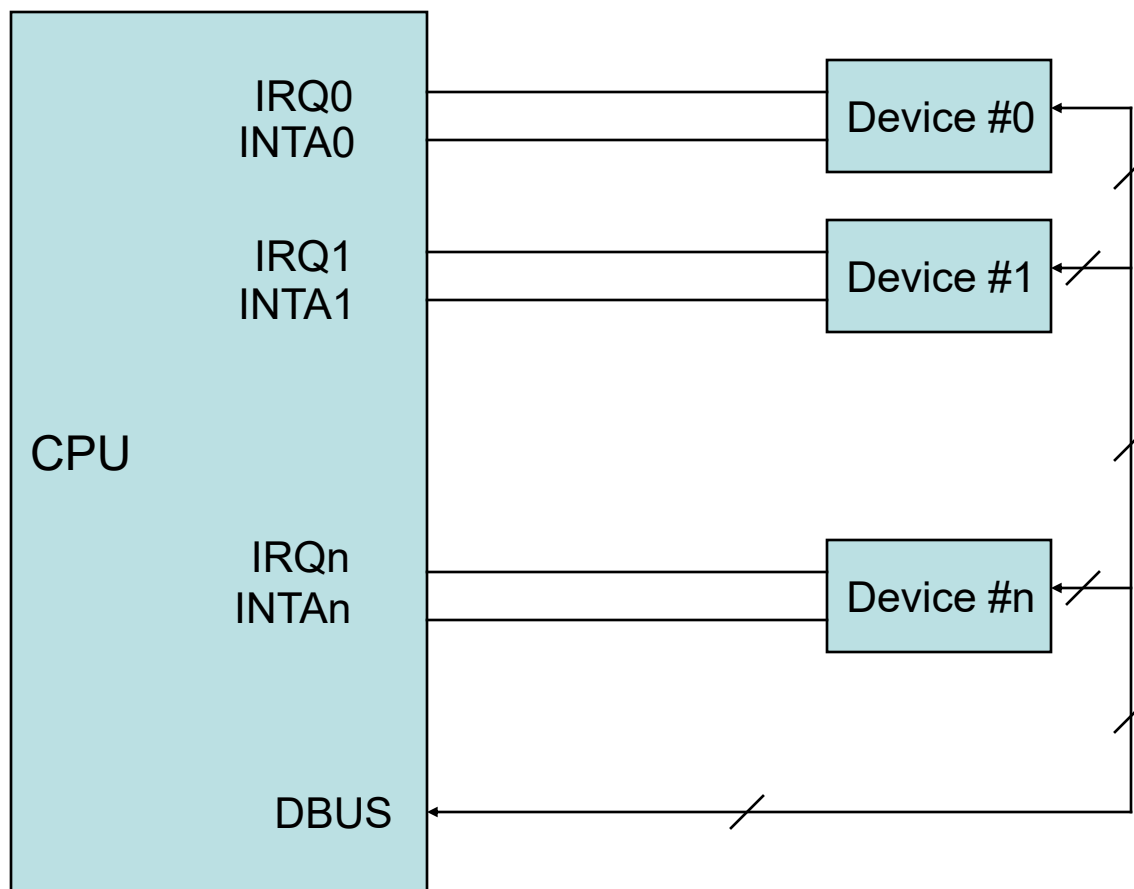
- Процесору их шаљу контролери периферија за прекиде 1, ако улази у IV табелу за маскирајуће прекиде нису фиксни.
- Процесор генерише фиксне вредности за прекиде 1, ако су улази у IV табелу за маскирајуће прекиде фиксни,
- процесор генерише фиксне вредности за прекиде из тачака 2, 3 и 4 и
- процесор генерише вредности на основу адресног дела инструкције INT за прекид из тачке 5.



Добијање броја улаза од контролера



Добијање броја улаза од контролера



Додела броја улаза контролера

- Како контролер периферије зна број улаза?
- Процесор програмским путем иницијализује регистар у коме се чува број улаза IE (*Interrupt Entry*) у контролеру периферије.
- **Пример:** Написати део програма којим се додељују бројеви улаза 4 и 2 периферијама PER1 и PER2, уколико су њихови IE регистри на адресама 65h и ACh, респективно.

LOAD #4

OUT 65h

LOAD #2

OUT ACh

Табела прекидних рутина (пример)

број улаза	прекид	адреса	садржај
4		9	16h
	CODE	8	0Eh
3		7	FDh
	PER1	6	3Ch
2		5	78h
	TRAP	4	54h
1		3	ABh
	PER0	2	65h
0		1	98h
	NMI	0	12h

IVTP=0

Ширина адресе /
адресибилна јединица

На којој адреси почиње прекидна рутине периферије PER1 (улаз број 3)?

$PC = IV[broj_ulaza]$

$PC \leq MEM[IVTP+broj_ulaza*sizeOfAddress]$

PC = 3CFDh или PC = FD3Ch?

За littleendian (нижи бајт на нижој адреси) PC = FD3Ch

Табела прекидних рутина (пример)

број улаза	прекид	адреса	садржај
4		9	16h
	CODE	8	0Eh
3		7	FDh
	PER1	6	3Ch
2		5	43h
	TRAP	4	21h
1		3	ABh
	PER0	2	65h
0		1	98h
	NMI	0	12h

IVTP=0

Иницијализовати улаз број 2 тако да указује на адресу 4321h? Littleendian

LOADB #21h

LOAD #4321h

STOREB 4h

STORE 4h

LOADB #43h

STOREB 5h

Ажурирање програмске статусне речи

- Брисање захтева за прекид у чију прекидну рутину се скаче (спољашњи прекиди)
- Брисање бита за *маскирање свих маскирајућих прекида* и *прекид после сваке инструкције* у програмској статусној речи процесора PSW (код прекида свих врста) - опционо
- Уписивање у бите *текући ниво приоритета* у програмској статусној речи процесора PSW ниво приоритета прекидне рутине на коју се скаче (маскирајућег прекида) - опционо



Регистар PSW – додатни бити

- I—бит који је једнак 1 ако треба да буду дозвољени маскирајући прекиди.
- T—бит који је једнак 1 ако треба да буде дозвољен прекид после сваке инструкције (*Trap*).
- L<n..0>—бити који говоре који је ниво приоритета програма који се тренутно извршава (користи се само за **маскирајуће** прекиде).
- P—бит који је једнак 1 ако је број улаза у табелу прекидних рутина фиксан за (1), 0 ако га периферија шаље процесору.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I	T	L1	L0	P	-	-	-	-	-	-	-	V	C	Z	N

Пример: Структура регистра PSW



Повратак из прекидне рутине

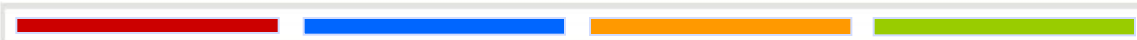
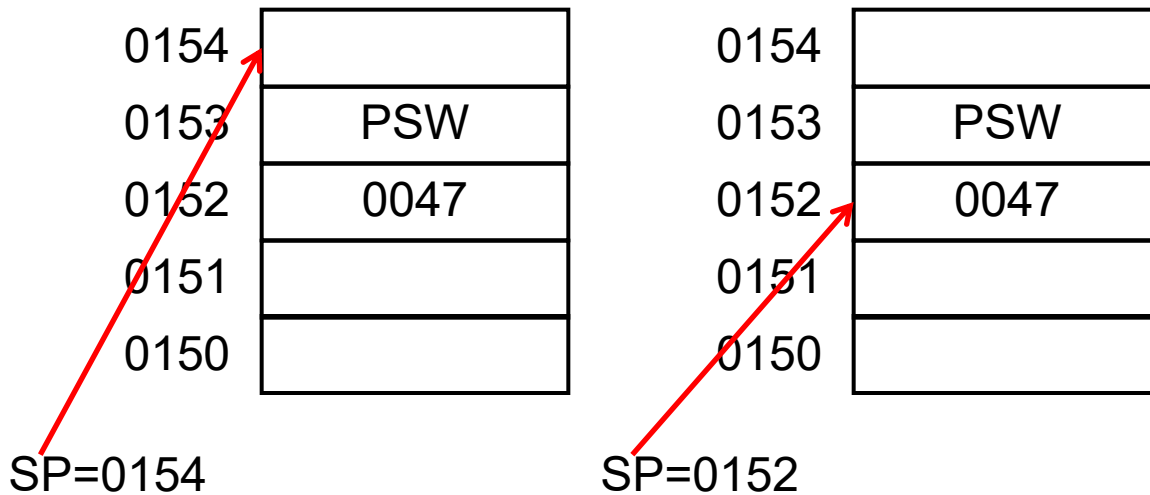
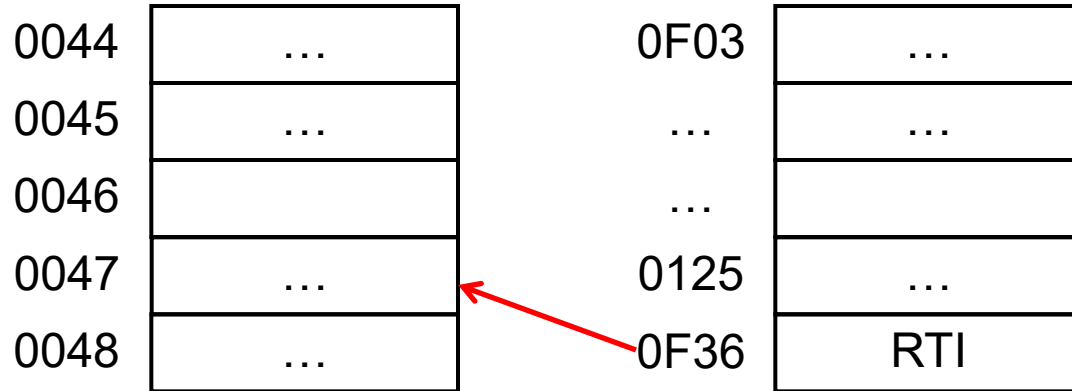
- Реализује се тако што се:
 - При крају прекидне рутине софтверски (са неколико инструкција) рестаурирају вредности оних програмски доступних регистара чије су вредности сачуване софтверски на почетку прекидне рутине
 - Изврши инструкција RTI која са стека рестаурира програмску статусну реч PSW и програмски бројач PC (као последња инструкција прекидне рутине).

Ако су софтверски сачувани приликом уласка у прекидну рутину

Скида се по обрнутом редоследу од постављања

Ако је хардверски сачувана приликом уласка у прекидну рутину

Повратак из прекидне рутине



Приоритети прекида

- У случајевима када се истовремено појавило више прекида они се прихватају по редоследу опадајућих приоритета.
 - INT (5) је највиши
 - Fault (3)
 - NMI (2)
 - IRQ (1)
 - Trap (4) је најнижи

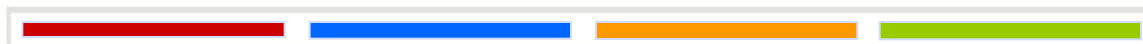
Могућ и другачији поредак!!
(зависи од имплементације, као и све остало)

Маскирање маскирајућих прекида

- Могуће је привремено забранити (маскирати) проверу маскирајућих прекида.
- Маскирање се може обавити на три начина:
 - Маскирање свих маскирајућих прекида
 - Селективно маскирање маскирајућих прекида
 - Маскирање нивоом приоритета

Маскирање свих маскирајућих прекида

- Маскирајући захтеви за прекид могу се **сви** маскирати (тренутно забранити) битом **I** *маскирање свих маскирајућих прекида* у програмској статусној речи процесора PSW.
- Маскирање важи без обзира на то да ли су прекиди селективно маскирани садржајем регистра маске IMR или не.
- Инструкције INTE и INTD за постављање односно брисање бита I



Селективно маскирање маскирајућих прекида

- Могуће је привремено забранити (маскирати) проверу неког од маскирајућих прекида.
- За маскирајуће прекиде постоји у процесору посебан програмски доступан регистар Регистар маске **IMR** (*Interrupt Mask Register*).
- Свакој линији захтева за прекид од контролера периферија придружен је посебан разред регистра маске.
- Захтев за прекид који стиже по одређеној линији у процесору ће бити прихваћен једино уколико се у одговарајућем разреду регистра маске налази вредност 1.



Селективно маскирање - пример

- Написати програм којим се забрањује прекид од периферије повезане на линију IRQ1 а дозвољава прекид од периферија повезаних на IRQ0 и IRQ2.

LOAD #101b

STOREIMR //посебна инструкција за упис у IMR регистар

Селективно маскирање - пример

- Написати програм којим се забрањује прекид од периферије повезане на линију IRQ1 а прекид од периферија повезаних на IRQ0 и IRQ2 не мења.

```
LOADIMR //посебна инструкција за читање IMR регистра  
AND #101b  
STOREIMR
```

Селективно маскирање - пример

- Написати програм којим се дозвољава прекид од периферије повезане на линију IRQ1 а прекид од периферија повезаних на IRQ0 и IRQ2 не мења.

```
LOADIMR  
OR #010b  
STOREIMR
```

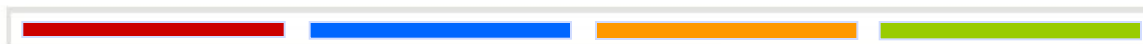

Маскирање нивоом приоритета

- Маскирајући захтеви за прекид могу се маскирати нивоом приоритета текућег програма.
- Уписивањем одговарајуће вредности у бите *текући ниво приоритета* у статусној речи процесора PSW обезбеђује се да се у случају приспећа маскирајућих захтева за прекид прихвате само они који су вишег нивоа приоритета од текућег нивоа приоритета.

или једнаког ако се тако специфицира

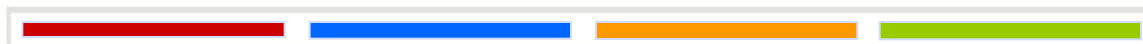
Гнеждење прекида

- Када процесор извршава прекидну рутину може стићи нови захтев за прекид. На овај захтев за прекид може се реаговати на следеће начине:
 - прекида се извршавање текуће прекидне рутине и скаче на нову прекидну рутину или
 - не прекида се извршавање прекидне рутине, већ се захтев за прекид прихвата тек по повратку у главни програм.
- Процесор реагује на оба начина у зависности од ситуације у којој се налази.



Гнеждење прекида

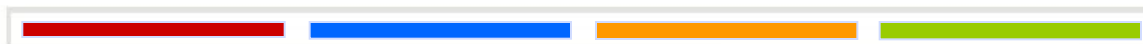
- Процесор одређује да ли прихвата нови прекид или не у зависности од:
 - текућег контекста
 - приспелих захтева за прекид
- Ово важи без обзира да ли се тренутно извршава главни програм или прекидна рутина.
- Процесор **не зна** да ли је у прекидној рутини или главном програму!



Кораци приликом провере прекида

1. Да ли инструкција реагује на прекид?
Ако не реагује онда се даље не проверавају прекиди.
2. Да ли се ради о инструкцији INT?
Ако да прихватити прекид,
број улаза је у инструкцији.
3. Да ли је текућа инструкција успешно извршена?
Ако није прихватити унутрашњи Fault прекид,
бројеви улаза су фиксни.
4. Да ли је приспео захтев за спољашњим немаскирајућим прекидом?
Ако јесте прихвати прекид, број улаза је фиксан.

Најчешће само један NMI!



Кораци приликом провере прекида

5. Да ли је бит I постављен на вредност 1?
Ако није прећи на проверу 8.
6. Да ли постоји захтев за спољашњим маскирајућим прекидом који није маскиран користећи IMR?
Ако не постоји прећи на проверу 8.
7. Да ли је ниво приоритета линије по којој је пристигао захтев виши (или једнак?) од текућег нивоа приоритета у регистру PSW?
Ако није прећи на проверу 8.
Ако јесте прихватити маскирајући прекид, број улаза или шаље периферија или је фиксан.



Кораци приликом провере прекида

8. Да ли је бит Т постављен на вредност 1?
Ако није завршити проверу прекида.
Ако јесте прихватити прекид после сваке инструкције, број улаза је фиксан.



Пример

Посматра се процесор код кога су и подаци и адресе ширине 2 бајта, а млађи бајт се смешта на нижу адресу.

Постоје три линије за спољашње маскирајуће прекиде (IRQ0, IRQ1, IRQ2). При чему је IRQ2 највишег приоритета.

Не прихвата се прекид истог нивоа.

На прекид не реагују инструкције: INTE, INTD, TRPE, TRPD, INT, RTI.

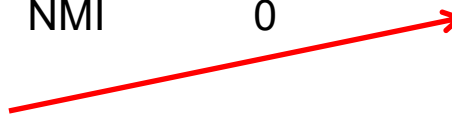
При прекиду на стеку се чува PSW па PC.

Стек расте од виших на нижим адресама, SP указује на заузету локацију.

Пример

број улаза	прекид	адреса	садржај
4		9	16h
	CODE	8	0Eh
3		7	FDh
	PER1	6	3Ch
2		5	78h
	TRAP	4	54h
1		3	ABh
	PER0	2	65h
0		1	98h
	NMI	0	12h

IVTP=0



Ситуација 1

Посматра се инструкција дужине 4 бајта на адреси
100h: ADD #3h

Контекст:

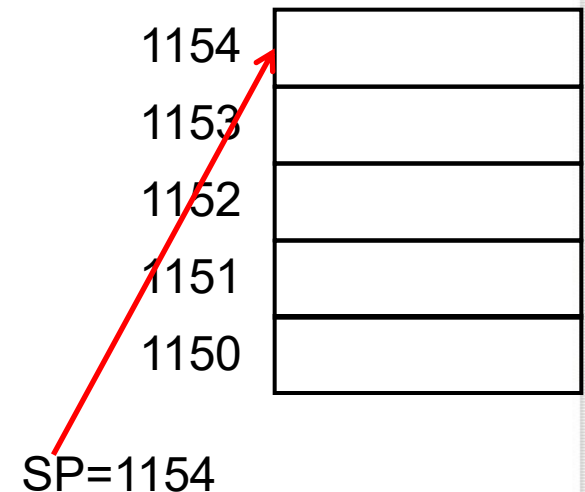
PSWI=0, PSWT=0, PSWL₁₀=01, IMR=111b

IRQ0=0, IRQ1=0, IRQ2=0, IRQN=0

Која је наредна инструкција?

Нема прекида.

Прелази се на 104h.



Ситуација 2

Посматра се инструкција дужине 4 бајта на адреси
100h: ADD #3h

Контекст:

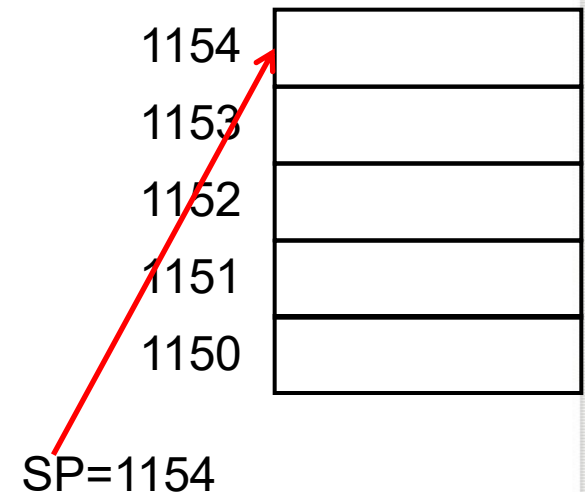
PSWI=0, PSWT=0, PSWL₁₀=01, IMR=111b

IRQ0=1, IRQ1=0, IRQ2=0, IRQN=0

Која је наредна инструкција?

Не прихвата се IRQ0, јер је I=0.

Прелази се на 104h.



Ситуација 3

Посматра се инструкција дужине 4 бајта на адреси
100h: ADD #3h

Контекст:

PSWI=1, PSWT=0, PSWL₁₀=01, IMR=111b

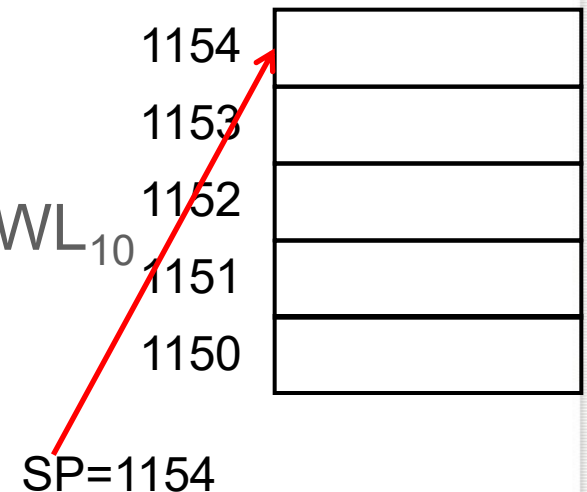
IRQ0=1, IRQ1=0, IRQ2=0, IRQN=0

Која је наредна инструкција?

Не прихвата се IRQ0,

IRQ0 није већег приоритета PSWL₁₀

Прелази се на 104h.



Ситуација 4

Посматра се инструкција дужине 4 бајта на адреси 100h: ADD #3h

Контекст:

PSWI=1, PSWT=0, PSWL₁₀=01, IMR=111b

IRQ0=0, IRQ1=1, IRQ2=0, IRQN=0

Која је наредна инструкција?

Промењено

Прихвата се IRQ1

Прелази се на FD3Ch

PSWI=0, PSWT=0, PSWL₁₀=10, IMR=111b

IRQ0=0, IRQ1=0, IRQ2=0, IRQN=0

1154	
1153	1001xxxxb
1152	xxxxxxxxb
1151	01h
1150	04h

SP=1150



Ситуација 5

Посматра се инструкција дужине 4 бајта на адреси
100h: ADD #3h

Контекст:

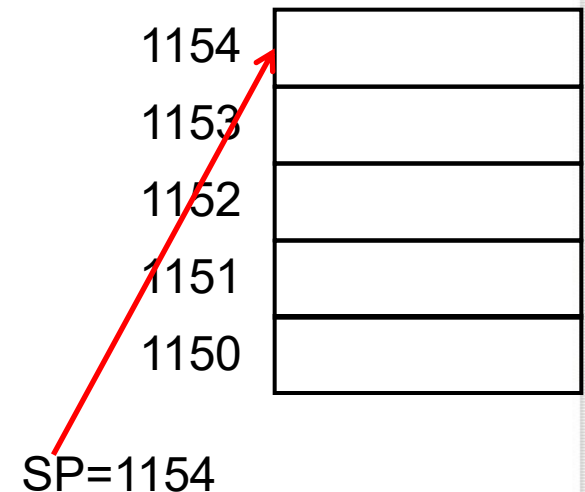
PSWI=1, PSWT=0, PSWL₁₀=01, IMR=001b

IRQ0=0, IRQ1=1, IRQ2=0, IRQN=0

Која је наредна инструкција?

Не прихвата се IRQ1, јер је
IMR₁ = 0

Прелази се на 104h.



Ситуација 6

Посматра се инструкција дужине 4 бајта на адреси 100h: ADD #3h

Контекст:

PSWI=0, PSWT=0, PSWL₁₀=01, IMR=111b

IRQ0=0, IRQ1=0, IRQ2=0, IRQN=1

Која је наредна инструкција?

Не мења се

Прихвата се NMI

Прелази се на 9812h

PSWI=0, PSWT=0, PSWL₁₀=01, IMR=111b

IRQ0=0, IRQ1=0, IRQ2=0, IRQN=0

1154	
1153	0001xxxxb
1152	xxxxxxxxb
1151	01h
1150	04h

SP=1150

Ситуација 7

Посматра се инструкција дужине 4 бајта на адреси 100h: STORE #3h

Контекст:

PSWI=0, PSWT=0, PSWL₁₀=01, IMR=111b

IRQ0=0, IRQ1=0, IRQ2=0, IRQN=0

Која је наредна инструкција?

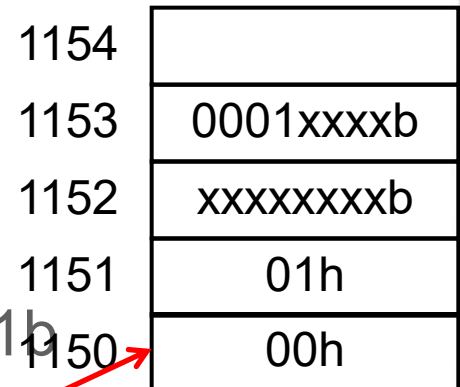
Не мења се

Прихвата се грешка

Прелази се на 160Eh

PSWI=0, PSWT=0, PSWL₁₀=01, IMR=111b

IRQ0=0, IRQ1=0, IRQ2=0, IRQN=0



SP=1150

Прекинута инструкција

Ситуација 8

Посматра се инструкција дужине 4 бајта на адреси
100h: INT #3h

Контекст:

PSWI=0, PSWT=0, PSWL₁₀=01, IMR=111b

IRQ0=0, IRQ1=0, IRQ2=0, IRQN=0

Која је наредна инструкција?

Не мења се

Прихвата се инструкција прекида

Прелази се на FD3Ch (улаз број 3)

PSWI=0, PSWT=0, PSWL₁₀=01, IMR=111b

IRQ0=0, IRQ1=0, IRQ2=0, IRQN=0

1154	
1153	0001xxxxb
1152	xxxxxxxxb
1151	01h
1150	04h

SP=1150



Ситуација 9

Посматра се инструкција дужине 4 бајта на адреси
100h: ADD #3h

Контекст:

PSWI=0, PSWT=1, PSWL₁₀=01, IMR=111b

IRQ0=0, IRQ1=0, IRQ2=0, IRQN=0

Која је наредна инструкција?

Не мења се

Прихвата се trap

Прелази се на 7854h

PSWI=0, PSWT=0, PSWL₁₀=01, IMR=111b

IRQ0=0, IRQ1=0, IRQ2=0, IRQN=0

1154	
1153	0101xxxxb
1152	xxxxxxxxb
1151	01h
1150	04h

SP=1150

Ситуација 10

Посматра се инструкција дужине 1 бајт на адреси
100h: INTD

Контекст:

PSWI=1, PSWT=0, PSWL₁₀=01, IMR=111b

IRQ0=0, IRQ1=0, IRQ2=1, IRQN=0

Која је наредна инструкција?

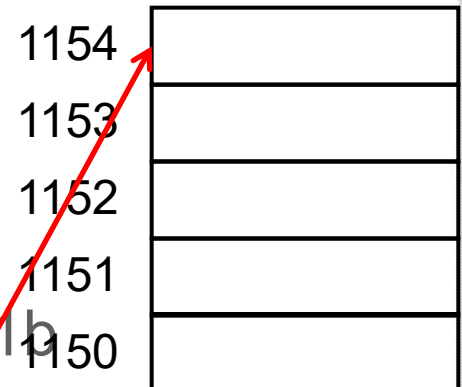
Промењено!

Инструкција не реагује на прекид.

Прелази се на 101h

PSWI=0, PSWT=0, PSWL₁₀=01, IMR=111b

IRQ0=0, IRQ1=0, IRQ2=1, IRQN=0



SP=1154



Ситуација 11

Посматра се инструкција дужине 4 бајта на адреси
100h: ADD #3h

Контекст:

PSWI=1, PSWT=0, PSWL₁₀=01, IMR=111b

IRQ0=1, IRQ1=1, IRQ2=1, IRQN=1

Која је наредна инструкција?

Не мења се

Прихвата се NMI

Прелази се на 9812h

PSWI=0, PSWT=0, PSWL₁₀=01, IMR=111b

IRQ0=1, IRQ1=1, IRQ2=1, IRQN=0

1154	
1153	1001xxxxb
1152	xxxxxxxxb
1151	01h
1150	04h

SP=1150

Ситуација 12

Посматра се инструкција дужине 1 бајт на адреси 9999h: RTI

Контекст пре извршења инструкције:

PSWI=0, PSWT=0, PSWL₁₀=01, IMR=111b

IRQ0=0, IRQ1=1, IRQ2=0, IRQN=0

Ако RTI **реагује** на прекид која је наредна инструкција?

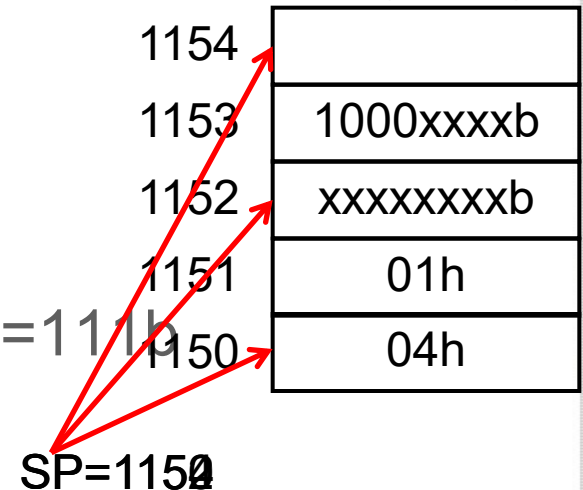
Скинуто са стека!

Прво се изврши операција RTI

PC=104h

PSWI=1, PSWT=0, PSWL₁₀=00, IMR=111b

IRQ0=0, IRQ1=1, IRQ2=0, IRQN=0



Ситуација 12 - наставак

Посматра се инструкција дужине 1 бајт на адреси 9999h: RTI – након извршења операције

PC=104h

PSWI=1, PSWT=0, PSWL₁₀=00, IMR=111b

IRQ0=0, IRQ1=1, IRQ2=0, IRQN=0

Прихвата се IRQ1


Прелази се на FD3Ch

PSWI=0, PSWT=0, PSWL₁₀=10, IMR=111b

IRQ0=0, IRQ1=0, IRQ2=0, IRQN=0

1154	
1153	1000xxxxb
1152	xxxxxxxxb
1151	01h
1150	04h

SP=1150



Питања?

Електротехнички Факултет
Универзитет у Београду

