

**Ј. ЂОРЂЕВИЋ, З. РАДИВОЈЕВИЋ, М. ПУНТ,
Б. НИКОЛИЋ, Д. МИЛИЋЕВ, Ј. ПРОТИЋ,
А. МИЛЕНКОВИЋ**

**АРХИТЕКТУРА
И ОРГАНИЗАЦИЈА
РАЧУНАРА**

**ПРЕКИДИ, МАГИСТРАЛА И
УЛАЗ/ИЗЛАЗ**

ЗБИРКА РЕШЕНИХ ЗАДАТАКА

Београд 2013.

САДРЖАЈ

САДРЖАЈ	I
1 ПРЕКИДИ	3
1.1 ЗАДАТАК 1	3
1.2 ЗАДАТАК 2	11
1.3 ЗАДАТАК 3	16
1.4 ЗАДАТАК 4	25
1.5 ЗАДАТАК 5 ZADATAK 1.39 SA STR 140	38
1.6 ЗАДАТАК 6 ZADATAK 1.21 SA STR 58	50
1.7 ЗАДАТАК 7	64
1.8 ЗАДАТАК 8 ЗАДАТАК 1.8 СА СТР 18	86
1.9 ЗАДАТАК 9	94
1.10 ЗАДАТАК 10	97
1.11 ЗАДАТАК 11	100
1.12 ЗАДАТАК 12	110
1.13 ЗАДАТАК 13	120

1 ПРЕКИДИ

1.1 ЗАДАТАК 1

Оперативна меморија неког рачунара је капацитета 64 КВ, а ширина речи меморије је 8 бита. Подаци ширине 16 бита се у меморију смештају у две суседне меморијске локације и то тако да се на нижој адреси налази виших 8 бита, а на вишој адреси нижих 8 бита податка. Механизам прекида је векторисан. Адресе прекидних рутина се чувају у табели адреса прекидних рутина (IV табела - *Interrupt Vector Table*) која почиње на адреси 0h оперативне меморије и има 256 улаза.

Контролери периферија PER1, PER2 и PER3 шаљу процесору спољашње маскирајуће захтеве за прекид преко улазних линија IRQM₁, IRQM₂ и IRQM₃, а од процесора добијају сигнале потврда преко излазних линија INTA₁, INTA₂ и INTA₃. Прекидне рутине периферија PER1, PER2 и PER3 почињу на адресама 1234h, 3456h и 5678h, респективно. Улази 1, 3 и 5 у IV табели су додељени периферијама PER1, PER2 и PER3, респективно. Контролер периферије шаље процесору захтев за прекид *intr* (*INTerrupt Request*) као импулс, и када од процесора прими сигнал потврде *inta* (*INTerrupt Acknowledge*) процесору шаљу свој број улаза у IV табелу.

а) Нацртати део оперативне меморије који се налази на улазима 0 до 5 у IV табели, означити адресе релевантних меморијских локација и попунити их одговарајућим вредностима.

б) Набројати корак по корак шта се све дешава у процесору од тренутка када је стигао захтев за прекид од контролера периферије до тренутка када се у регистар PC (програмски бројач – *Program Counter*) упише почетна адреса прекидне рутине.

в) Објаснити за сваки корак набројан у тачки б) да ли га обавља хардвер или софтвер.

г) Објаснити како се обавља повратак из прекидне рутине и набројати кораке који се том приликом извршавају.

д) Захтеви за прекид преко линија IRQM₁, IRQM₂ и IRQM₃ могу да стигну истовремено од контролера периферија PER1, PER2 и PER3. У неком тренутку само један од захтева за прекид може да се прихвати и у зависности од тога који се захтев за прекид прихвата по једној од линија INTA₁, INTA₂ или INTA₃ се шаље сигнал потврде. Нацртати и објаснити како се утврђује који се од ова три захтева за прекид прихвата и по којој линији се шаље сигнал потврде.

ђ) У неким варијантама реализације процесора не постоје линије за слање сигнала потврде и контролери периферија не шаљу бројеве улаза у IV табелу, већ их сам процесор генеришу. Нацртати и објаснити како се тада генеришу бројеви улаза 1, 3 и 5 додељени у IV табели периферијама PER1, PER2 и PER3, респективно.

Решење:

а) Капацитет оперативне меморије је 64 КВ. Како је ширина речи меморије 8 бита, то оперативна меморија има 2^{16} адресибилних локација које садрже речи меморије ширине 8 бита. Стога су адресе које се користе за адресирање меморијских локација широке 16 бита.

Табела адреса прекидних рутина састоји се од 256 улаза који садрже почетне адресе прекидних рутина. Како су адресе широке 16 бита, а ширина речи меморије је 8 бита,

сваки улаз табеле заузима две суседне локације у меморији. На локацији са нижом адресом се налази виших 8 бита адресе прекидне рутине, а на локацији са вишом адресом се налази нижих 8 бита адресе прекидне рутине. Почетна адреса IV табеле, која се чува у регистру IVTP (*Interrupt Vector Table Pointer*), је 0h.

У општем случају адреса на којој започиње улаз i који одговара некој периферији PER j и у коме се налази адреса прекидне рутине периферије PER j се добија према изразу

$$IVTP + offset_i$$

у коме је IVTP почетна адреса IV табеле, а $offset_i$ је померај улаза i у односу на почетак IV табеле. Померај $offset_i$ се добија према изразу

$$i * size$$

у коме је i број улаза у коме се налази адреса прекидне рутине и $size$ величина улаза изражена као број адресибилних меморијских локација које заузима адреса прекидне рутине која се у њему налази. Обично се узима да величина улаза $size$ буде степен броја два и износи 2^k , јер се тада претварање броја улаза i у померај $offset_i$ реализује једноставно померањем броја улаза i за k места улево.

У датом случају величина улаза у IV табели исказана у адресибилним јединицама је 2, јер је адреса широка 16 бита, а ширина речи меморије која је адресибилна јединица је 8 бита. Ако се број улаза i који одговара некој периферији PER j помножи са 2 добија се померај улаза i у односу на почетак IV табеле. Када се померај дода на садржај регистра IVTP у коме је IVTP почетна адреса IV табеле добија се адреса a на којој је виших 8 бита адресе прекидне рутине периферије PER j , док је на адреси $a+1$ нижих 8 бита адресе прекидна рутина за PER j .

Део IV табеле са улазима од 0 до 5 приказан је на слици 1.а. Улаз је 1 додељен периферији PER1, па се на адреси 0002h налази се виших 8 бита адресе прекидне рутине за PER1(12h), а на адреси 0003h нижих 8 бита адресе прекидне рутине за PER1(34h). Улаз 3 је додељен периферији PER2, па се на адреси 0006h налази виших 8 бита адресе прекидне рутине за PER2(34h), а на адреси 0007h нижих 8 бита адресе прекидне рутине за PER2(56h). Улаз 5 је додељен периферији PER3, па се на адреси 000Ah налази се виших 8 бита адресе прекидне рутине за PER3(56h), а на адреси 000Bh нижих 8 бита адресе прекидне рутине за PER3(78h).

број улаза у IV табелу	меморијска адреса	садржај	опис садржаја
5	000Bh	78h	нижих 8 битова адресе прекидне рутине за PER3
	000Ah	56h	виших 8 битова адресе прекидне рутине за PER3
4	0009h		
	0008h		
3	0007h	56h	нижих 8 битова адресе прекидне рутине за PER2
	0006h	34h	виших 8 битова адресе прекидне рутине за PER2
2	0005h		
	0004h		
1	0003h	34h	нижих 8 битова адресе прекидне рутине за PER1
	0002h	12h	виших 8 битова адресе прекидне рутине за PER1
0	0001h		
	0000h		

Слика 1.а Изглед дела оперативне меморије са улазима од 0 до 5 у IV табели

Напомена: Улазе треба нумерисати почев од 0.

б) Извршавање инструкције се састоји из четири фазе и то *читање инструкције, формирање адресе и читање операнда, извршавање операције и опслуживање прекида*. Захтев за прекид може да се јави у било ком тренутку током извршавања прве три фазе. Процесор не реагује одмах на спољни маскирајући захтев за прекид већ га памти у одговарајућем разреду регистра PRIRQM и продужава са извршавањем инструкције. Тек по комплетирању прве три фазе процесор прелази на фазу *опслуживање прекида*. Фаза *опслуживање прекида* се састоји из корака у оквиру којих се *чува контекст процесора и израчунава адреса прекидне рутине*.

Чување контекста процесора се састоји из корака у оквиру којих се контекст процесора који чине PC, PSW (програмска статусна реч – *Program Status Word*) и програмски доступни регистри чува на стеку.

Израчунавање адресе прекидне рутине се састоји из корака у оквиру којих процесор шаље *inta* сигнал контролеру периферије и брише одговарајући бит регистра PRIRQM у коме је запамћен спољашњи маскирајући захтев за прекид, контролер периферије шаље процесору по линијама магистрале података свој број улаза у IV табелу,

процесор израчуна адресу улаза у IV табелу тако што добијени број улаза претвори у померај и дода на садржај IVTP регистра и

процесор са израчунате адресе улаза у IV табели прочита адресу прекидне рутине и упише у PC.

У оквиру ових корака се веома често извршавају и следећи кораци:

бит I (*Interrupt Enable* – маскирање свих маскирајућих захтева за прекид) у регистру PSW се поставља на 0,

бит T (*Trap Enable* – прекид после сваке инструкције) у регистру PSW се поставља на 0 и

битови L (*Priority Level* – приоритет текућег програма) у регистру PSW се постављају на ниво приоритета прекидне рутине на коју се скаче.

в) Одговор у тачки б) подразумева да се сви кораци изводе хардверски. Међутим, неки од корака у оквиру којих се чува контекст процесора могу да се изведу софтверски. Ово је случај код програмски доступних регистара и понекад код регистра PSW чије вредности могу софтверски да се сачувају на стеку на почетку прекидне рутине. Програмски доступни регистри се обично чувају софтверски ако их је много, док се у супротном случају чувају хардверски. Софтверски се чувају само они регистри чије се вредности мењају у прекидној рутини, при чему се њихове вредности софтверски рестаурирају пре повратка из прекидне рутине. Регистар PSW се мора чувати хардверски уколико се у оквиру корака у којима се израчунава адреса прекидне рутине извршавају и кораци којима се битови I и T у регистру PSW се постављају на 0 и битови L у регистру PSW постављају на ниво приоритета прекидне рутине на коју се скаче. У супротном регистар PSW се може се чувати или хардверски или софтверски. У већини случајева регистар PSW се чува хардверски.

г) Повратак из прекидне рутине се изводи посебном инструкцијом RTI (*ReTurn from Interrupt*). Ова инструкција рестаурира са стека контекст процесора, при чему се

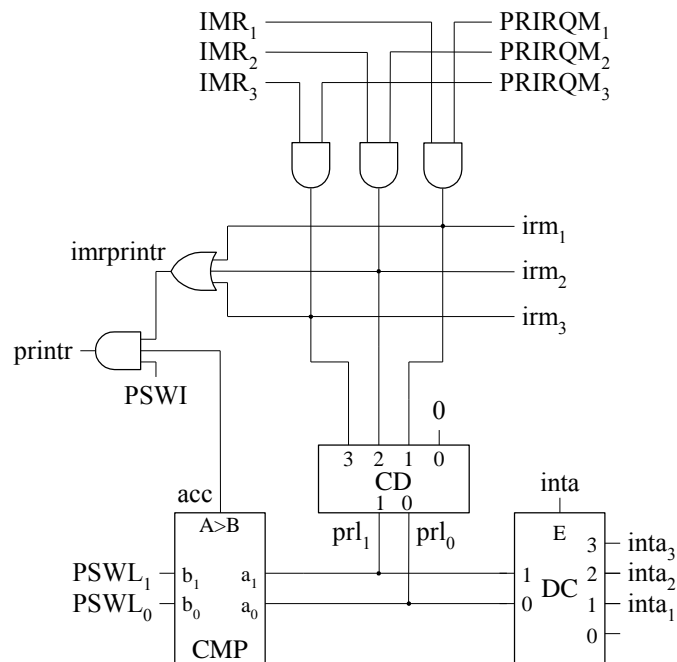
програмски доступни регистри рестаурирају ако су сачувани хардверски у оквиру корака фазе *ослуживање прекида*,

регистар PSW рестаурира ако је сачуван хардверски у оквиру корака фазе *ослуживање прекида* и

регистар PC се увек рестаурира.

Ако се програмски доступни регистри и PSW не рестаурирају инструкцијом RTI, зато што нису хардверски у оквиру корака фазе *ослуживање прекида* сачувани на стеку већ софтверски на почетку прекидне рутине, морају се рестаурирати софтверски при крају прекидне рутине пре инструкције RTI. При томе се софтверски при крају прекидне рутине рестаурирају само они програмски доступни регистри који се софтверски чувају на почетку прекидне рутине.

д) Спољашњи маскирајући захтеви за прекид од контролера периферија PER1, PER2 и PER3 који се повезују на улазне линије процесора IRQM₁, IRQM₂ и IRQM₃ долазе као импулс, па се у процесору памте у флип-флоповима PRIRQM₁, PRIRQM₂ и PRIRQM₃, респективно (слика 1.б). У процесору постоји и регистар маске IMR_{3...1} који се користи за селективно маскирање спољњих маскирајућих захтева за прекид. Стога се коришћењем логичких I елемената од сигнала захтева за прекид PRIRQM₁, PRIRQM₂ и PRIRQM₃, и сигнала маске IMR₁, IMR₂ и IMR₃, формирају сигнали захтева за прекид са маском irm₁, irm₂ и irm₃, респективно.



Слика 1.б Прихватање захтева за прекид и слање сигнала потврде

Да би се остварио механизам којим се

прекида извршававање главног програма уколико се јави било који од спољњих маскирајућих захтева за прекид који није селективно маскиран одговарајућим битом регистра маске и

утврђује који се од захтева за прекид који долазе преко линија IRQM₁, IRQM₂ и IRQM₃ прихвата,

главном програму и прекидним рутинама периферија PER1, PER2 и PER3 се додељују нивои приоритета. У овом случају постоји главни програм и три прекидне рутине, па су

четири бинарне вредности 00, 01, 10 и 11 довољне за кодирање њихових нивоа приоритета. Да би се омогућило прекидало извршавање главног програма када стигне захтев за прекид од било ког контролера периферија PER1, PER2 и PER3, главном програму треба доделити најнижи ниво приоритета 00 а прекидним рутинама периферија PER1, PER2 и PER3 треба некако доделити преостале више нивое приоритета 01, 10 и 11. Уколико се жели да у оквиру периферија PER1, PER2 и PER3 периферија PER1 може да прекида извршавање прекидних рутина периферија PER2 и PER3 и да у оквиру периферија PER2 и PER3 периферија PER2 може да прекида извршавање прекидне рутине периферије PER3, тада периферија PER1 треба да има највиши ниво приоритета, периферија PER2 нижи и периферија PER3 најнижи. У том случају периферијама PER1, PER2 и PER3 треба доделити нивое приоритета 11, 10 и 01, респективно. Међутим, нивое приоритета је могуће и другачије доделити периферијама. Тако је у даљим разматрањима узето да периферија PER3 има највиши ниво приоритета, периферија PER2 нижи и периферија PER1 најнижи, па су периферијама PER3, PER2 и PER1 и додељени нивои приоритета 11, 10 и 01, респективно.

Сигнали irm_1 , irm_2 и irm_3 се даље користе за формирање сигнала спољњег маскирајућег захтева за прекид $imgrirqm$, сигнала највишег нивоа приоритета спољњег маскирајућег захтева за прекид $prl_{1..0}$ и сигнала прихватања спољњег маскирајућег захтева за прекид ass .

Сигнал спољњег маскирајућег захтева за прекид $imgrirqm$ се формира на излазу логичког ILI елемента од сигнала irm_3 , irm_2 и irm_1 .

Сигнали највишег нивоа приоритета спољњег маскирајућег захтева за прекид $prl_{1..0}$ се формирају на излазу кодера приоритета CD од сигнала irm_3 , irm_2 и irm_1 који се у складу са додељеним нивоима приоритетима периферијама PER3, PER2 и PER1 везују на улазе 3, 2 и 1 кодера приоритета CD , респективно.

Ниво приоритета текућег програма се чува у разредима $PSWL_{1..0}$ регистра PSW . У оквиру фазе *опслуживање прекида* хардверски се најпре на стек стави садржај регистра PSW са вредностима разреда $PSWL_{1..0}$ која представља ниво приоритета текућег програма чије се извршавање прекида, а затим се у разреде $PSWL_{1..0}$ уписује вредност сигнала $prl_{1..0}$ који представљају сигнале највишег нивоа приоритета спољњег маскирајућег захтева за прекид на чију се прекидну рутину прелази. На крају сваке прекидне рутине се програмским путем инструкцијом RTI скидањем садржаја са стека и уписивањем у регистре PSW и PC у разреде $PSWL_{1..0}$ враћа вредност нивоа приоритета прекинутог програма чије се извршавање наставља.

Сигнал прихватања спољњег маскирајућег захтева за прекид ass се формира на излазу $A>B$ компаратора CMP уколико је вредност сигнала $prl_{1..0}$ већа од вредности сигнала $PSWL_{1..0}$.

Бит $PSWI$ вредношћу 1 дозвољава прихватање спољњих маскирајућих захтева за прекид.

Сигнал прихватања спољњег маскирајућег захтева за прекид $printr$ добија вредност 1 уколико вредност 1 имају сигнали $imgrirqm$, ass и $PSWI$. То значи да је за прекидање извршавања текућег програма потребно

да постоји барем један спољашњи захтев за прекид који није селективно маскиран одговарајућим разредом регистра маске IMR , па сигнал $imgrirqm$ има вредност 1,

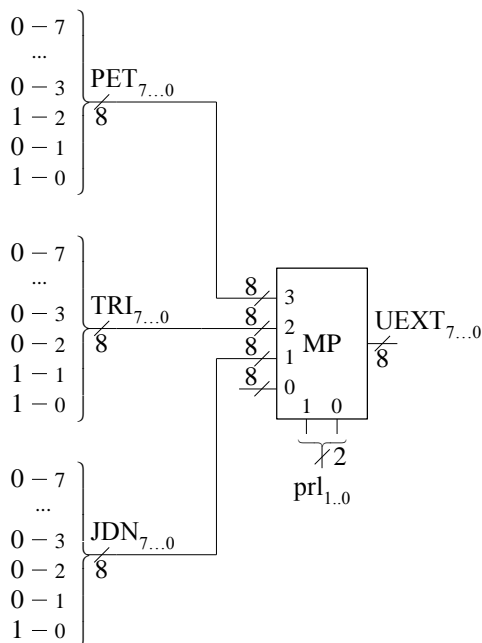
да је његов ниво приоритета већи од нивоа приоритета текућег програма, па сигнал ass има вредност 1 и

да нису маскирани спољњи маскирајући захтеви за прекид, јер сигнал PSWI има вредност 1.

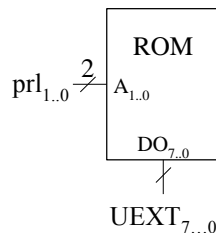
На почетку секвенца корака којом се у оквиру фазе *опслуживање прекида* израчунава адреса прекидне рутине процесор генерише вредност 1 сигнал *inta*, која на излазима декодера DC даје вредност 1 једног од сигнала *inta₃*, *inta₂* или *inta₁*, у зависности од тога да ли сигнали *prl_{1..0}* имају вредност 11, 10 или 01, респективно. Сигнали *inta₃*, *inta₂* и *inta₁* се по линијама INTA₃, INTA₂ и INTA₁ шаљу контролерима периферија PER3, PER2 и PER1, респективно. Поред тога, у зависности од тога да ли сигнали *prl_{1..0}* имају вредност 11, 10 или 01, брише се флип-флоп PRIRQM₃, PRIRQM₂ или PRIRQM₁, респективно.

ђ) Прихватање спољашњег маскирајућег захтева за прекид се највећим делом реализује на начин објашњен у тачкама б) и д). Разлика је само у томе да на почетку секвенце корака којом се у оквиру фазе *опслуживање прекида* израчунава адреса прекидне рутине процесор не шаље сигнал потврде једном од контролера периферија и сходно томе не добија број улаза од одговарајућег контролера периферије, већ број улаза сам генерише (слике 1.в и 1.г).

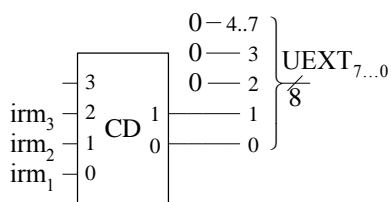
У случају генерисања броја улаза на начин дат на слици 1.в, фиксно формиране вредности бројева улаза 5, 3 и 1 за периферије PER3, PER2 и PER1, означене са PET_{7..0}, TRI_{7..0} и JDN_{7..0}, се воде на улазе 3, 2 и 1 мултиплексера MP, да би се на основу вредности сигнала *prl_{1..0}* једна од њих селектовала и појавила као сигнали броја улаза UEXT_{7..0} на излазима мултиплексера. У случају генерисања броја улаза на начин дат на слици 1.г, вредности бројева улаза 5, 3 и 1 за периферије PER3, PER2 и PER1, су уписане у улазе 3, 2 и 1 меморије ROM. На основу вредности сигнала сигнала *prl_{1..0}* који су везани на адресне линије A_{1..0} меморије EPROM, на излазним линијама података DO_{7..0} се појављује вредност броја улаза UEXT_{7..0}.



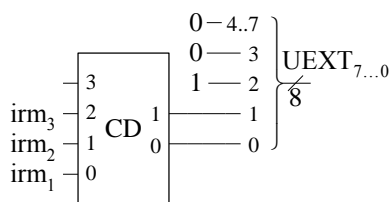
Слика 1.в Генерисање броја улаза мултиплексирањем



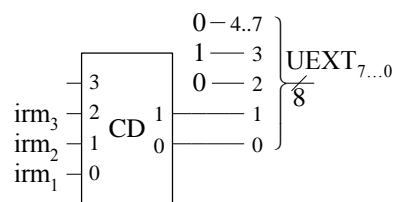
Слика 1.г Генерисање броја улаза читањем из ROM-а



Слика 1.д Генерисање бројева улаза 0 до 2 кодером приоритета



Слика 1.ђ Генерисање бројева улаза 4 до 6 кодером приоритета



Слика 1.е Генерисање бројева улаза 8 до 10 кодером приоритета

У случају процесора са три спољашња маскирајућа захтева за прекид и фиксним генерисања броја улаза, једноставније је генерисање бројеваа улаза уколико се адресе прекидних рутина у IV табели налазе у суседним улазима који почињу на граници 4. Тако на пример уколико се адресе прекидних рутина периферија PER1, PER2 и PER3 налазе у улазима 0, 1 и 2 бројеви улаза се могу генерисати на начин дат на слици 1.д. Уколико се пак адресе прекидних рутина периферија PER1, PER2 и PER3 налазе у улазима 4, 5 и 6 бројеви улаза се могу генерисати на начин дат на слици 1.ђ. На сличан начин се, уколико се адресе прекидних рутина периферија PER1, PER2 и PER3 налазе у улазима 8, 9 и 10, бројеви улаза могу генерисати на начин дат на слици 1.е. Вредности 00, 01 и 10 два најмлађа бита броја улаза се добијају на излазима кодера CD на основу вредности сигнала irm_1 , irm_2 и irm_3 , док се старијих шест бита фиксно постављају на 000000, 000001 и 000010 у зависности од тога да ли се ради о улазима 0 до 2, 4 до 6 или 8 до 10, респективно.

Дискусија:

У свим даљим разматрањима се подразумева да регистар PSW (програмска статусна реч – *Program Status Word*) има битове I, T, L, N, Z, C и V.

Бит I (*Interrupt Enable*) вредношћу 1 дозвољава прихватање спољњих маскирајућих захтева за прекид. Бит I се поставља на вредности 1 и 0 софтверски посебним инструкцијама INTE и INTD, респективно. Бит I се поставља на вредност 0 и хардверски током извршавања фазе опслуживање прекида. Бит I се користи да се хардверски дозволе или маскирају спољњих маскирајућих захтева за прекид.

Бит T (*Trap Enable*) вредношћу 1 дозвољава режим рада прекид после сваке инструкције. Бит T се поставља на вредности 1 и 0 софтверски посебним инструкцијама TRPE и TRPD, респективно. Бит T се поставља на вредност 0 и хардверски током извршавања фазе опслуживање прекида. Бит T се користи да се хардверски генерише прекид после сваке инструкције.

Битови L (*Priority Level*) садрже ниво приоритета текућег програма. Користе се само за главни програм и прекидне рутине спољњих маскирајућих прекида. Број битова L зависи од броја спољњих масакирајућих захтева за прекид и од тога да ли се прихватају само захтеви за прекид вишег нивоа приоритета или се прихватају захтеви за прекид истог и вишег нивоа приоритета. Уколико постоје три спољња масакирајућа захтева за прекид и прихватају се само захтеви за прекид вишег нивоа приоритета довољна су два бита. У овом случају би се четири различите бинарне вредности користиле за четири различита нивоа приоритета главног програм и три прекидне рутине. Уколико постоје три спољња масакирајућа захтева за прекид и прихватају се захтеви за прекид истог и вишег нивоа приоритета, поново су довољна два бита. Међутим, у овом случају би могло да се од

четири различите бинарне вредности користе само три и то једна за исти ниво приоритета главног програм и једне прекидне рутине и две од преостале три за нивое приоритета преостале две прекидне рутине. Битови L се постављају на ниво приоритета прекидне рутине на коју се скаче хардверски током извршавања фазе опслуживање прекида. Битови L се користе да се хардверски током извршавања фазе опслуживање прекида утврди да ли се прихвата неки од спољњих маскирајућих захтева за прекид.

Битови N, Z, C и V се постављају хардверски на основу резултата извршавања одређених инструкција на вредност 1 уколико је резултат негативан, уколико је резултат нула, уколико постоји пренос/позајмица у аритметици целобројних вредности без знака и уколико постоји прекорачење у аритметици целобројних вредности са знаком, респективно. Битови N, Z, C и V се проверавају софтверски инструкцијама условног скока.

Битови I, T, L, N, Z, C и V се постављају и софтверски на вредности очитане са стека приликом извршавања инструкције повратка из прекидне рутине RTI.

1.2 ЗАДАТАК 2

Оперативна меморија неког рачунара је капацитета 8 GB, а ширина речи меморије је 16 бита. Подаци ширине 32 бита се у меморију смештају у две суседне меморијске локације и то тако да се на нижој адреси налази нижих 16 бита, а на вишој адреси виших 16 бита податка. Процесор је једноадресни, акумулатор је дужине 16 бита и све инструкције се извршавају над 16 битним величинама. Улазно/излазни и меморијски адресни простори су раздвојени. Механизам прекида је векторисан. Адресе прекидних рутина се чувају у табели адреса прекидних рутина (IV табела - *Interrupt Vector Table*) која почиње на адреси 0h оперативне меморије и има 64 улаза.

Контролери периферија PER1, PER2 и PER3 шаљу процесору спољашње маскирајуће захтеве за прекид преко улазних линија IRQM₁, IRQM₂ и IRQM₃, а од процесора добијају сигнале потврда преко излазних линија INTA₁, INTA₂ и INTA₃. Прекидне рутине периферија PER1, PER2 и PER3 почињу на адресама 12345678h, 23456789h и 3456789Ah, респективно. Улази 3, 5 и 7 у IV табели су додељени периферијама PER1, PER2 и PER3, респективно. Контролер периферије шаље процесору захтев за прекид *intr* као импулс и када од процесора прими сигнал потврде *inta* процесору шаљу свој број улаза у IV табелу. Адресе 16 битних регистара контролера периферија PER1, PER2 и PER3 у којима се чувају бројеви улаза у IV табелу су 0h, 4h и 8h, респективно.

а) Написати део програма којим се додељују бројеви улаза у IV табелу за периферије PER1, PER2 и PER3.

б) Нацртати део оперативне меморије који заузима првих 8 улаза у IV табели, означити адресе релевантних меморијских локација и попунити их одговарајућим вредностима.

в) Написати део програма којим се попуњава улаз 5 у IV табели.

г) Дати вредност коју контролер периферије PER2 шаље процесору када му процесор пошаље сигнал потврде.

д) Описати и представити програмом поступак којим се и периферији PER2 додељује иста прекидна рутина као и периферији PER3.

ђ) Објаснити колико је меморијских локација неопходно за смештање комплетне IV табеле. Објаснити колико би максимално могло да буде улаза у IV табели у описаној конфигурацији уколико се посматрају спољашњи маскирајући захтеви за прекид. Објаснити колико је меморијских локација тада неопходно за смештање комплетне IV табеле.

Решење:

а) Бројеви улаза у IV табелу чувају се у за то намењеним регистрима контролера периферија PER1, PER2 и PER3 који се налазе на адресама 0h, 4h и 8h, респективно, у улазно/излазном адресном простору. Пошто су улазно/излазни и меморијски адресни простори раздвојени, упис садржаја акумулатора у ове регистре контролера периферија се обавља користећи инструкцију OUT. Да је улазно/излазни адресни простор меморијски пресликан, упис садржаја акумулатора у ове регистре контролера периферија би се обављао користећи инструкцију STORE. Тражени део програма је приказан на слици 2.а. Овај програм извршава оперативни систем у току иницијализације система.

LOAD #3 ; напуни акумулатор непосредном величином 3 која представља број улаза у IV табелу за PER1
 OUT 0h ; пребаци садржај акумулатора у регистар контролера периферије PER1
 LOAD #5 ; напуни акумулатор непосредном величином 5 која представља број улаза у IV табелу за PER2
 OUT 4h ; пребаци садржај акумулатора у регистар контролера периферије PER2
 LOAD #7 ; напуни акумулатор непосредном величином 7 која представља број улаза у IV табелу за PER3
 OUT 8h ; пребаци садржај акумулатора у регистар контролера периферије PER3

Слика 2.а Иницијализација бројева улаза за PER1, PER2 и PER3

б) Капацитет оперативне меморије је 8 GB. Како је ширина речи меморије 16 бита, то оперативна меморија има 2^{32} адресибилних локација које садрже речи меморије ширине 16 бита. Стога су адресе које се користе за адресирање меморијских локација широке 32 бита.

Табела адреса прекидних рутина састоји се од 64 улаза који садрже почетне адресе прекидних рутина. Како су адресе широке 32 бита, а ширина речи меморије је 16 бита, сваки улаз табеле заузима две суседне локације у меморији. На локацији са нижом адресом се налази нижих 16 бита адресе прекидне рутине, а на локацији са вишом адресом се налази виших 16 бита адресе прекидне рутине. Почетна адреса IV табеле, која се чува у регистру IVTP (*Interrupt Vector Table Pointer*), је 0h.

У датом случају величина улаза IV табели исказана у адресибилним јединицама је 2, јер је адреса широка 32 бита, а ширина речи меморије која је адресибилна јединица је 16 бита. Ако се број улаза i који одговара некој периферији PER j помножи са 2 добија се померај улаза i у односу на почетак IV табеле. Када се померај дода на садржај регистра IVTP у коме је IVTP почетна адреса IV табеле добија се адреса a на којој је нижих 16 бита адресе прекидне рутине периферије PER j , док је на адреси $a+1$ виших 16 бита адресе прекидне рутине PER j .

Део IV табеле са улазима од 0 до 7 приказан је на слици 2.б. Улаз 3 је додељен периферији PER1, па се на адреси 00000006h налази нижих 16 бита адресе прекидне рутине за PER1(5678h), а на адреси 00000007h виших 16 бита адресе прекидне рутине за PER1(1234h). Улаз 5 је додељен периферији PER2, па се на адреси 0000000Ah налази нижих 16 бита адресе прекидне рутине за PER2(6789h), а на адреси 0000000Bh виших 16 бита адресе прекидне рутине за PER2(2345h). Улаз 7 је додељен је периферији PER3, па се на адреси 0000000Eh налази нижих 16 бита адресе прекидне рутине за PER3(789Ah), а на адреси 0000000Fh виших 16 бита адресе прекидне рутине за PER3(3456h).

број улаза у IV табелу	меморијска адреса	садржај	опис садржаја
7	0000000Fh	3456h	виших 16 битова адресе прекидне рутине за PER3
	0000000Eh	789Ah	нижих 16 битова адресе прекидне рутине за PER3
6	0000000Dh		
	0000000Ch		
5	0000000Bh	2345h	виших 16 битова адресе прекидне рутине за PER2
	0000000Ah	6789h	нижих 16 битова адресе прекидне рутине за PER2
4	00000009h		
	00000008h		

3	00000007h	1234h	виших 16 битова адресе прекидне рутине за PER1
	00000006h	5678h	нижих 16 битова адресе прекидне рутине за PER1
2	00000005h		
	00000004h		
1	00000003h		
	00000002h		
0	00000001h		
	00000000h		

Слика 2.б Изглед дела оперативне меморије са улазима од 0 до 7 у IV табели

в) Потребно је у улаз 5 IV табеле, који почиње на адреси Ah, уписати адресу прекидне рутине и то у локацију 0000000Ah треба уписати нижих 16 битова адресе (6789h), а у локацију 0000000Bh виших 16 битова адресе (2345h). Све инструкције процесора се извршавају над 16 битним величинама, па се тражена иницијализација мора извести из два преноса 16 битних речи. IV табела се налази у оперативној меморији која је у меморијском адресном простору, па се упис садржаја акумулатора у ове меморијске локације обавља користећи инструкцију STORE. Тражени део програма је приказан на слици 2.в. И овај део програма се извршава током иницијализације система.

```

LOAD      #2345h ; напуни акумулатор непосредном величином 2345h која представља
           виших 16 битова адресе
STORE 0000000Bh ; пребаци садржај акумулатора у меморијску локацију на адреси
           0000000Bh
LOAD      #6789h ; напуни акумулатор непосредном величином 6789h која представља
           нижих 16 битова адресе
STORE 0000000Ah ; пребаци садржај акумулатора у меморијску локацију на адреси
           0000000Ah

```

Слика 2.в Иницијализација улаза 5 у IV табели

г) Контролер периферије шаље процесору свој број улаза у IV табелу када му процесор пошаље сигнал потврде *inta*. Број улаза у IV табелу се шаље по магистрали података и у случају периферије PER2 износи 5.

д) Постоје два начина да се периферији PER2 додели иста прекидна рутина као и периферији PER3.

Први, могући начин да се постигне тражено је да улаз 5 у IV табели за периферију PER2 садржи исту адресу прекидне рутине као улаз 7 у IV табели за периферију PER3. Зато је потребно да се у улаз 5 IV табеле за периферију PER2 упише иста вредност адресе прекидне рутине као и у улаз 7 IV табеле за периферију PER3. То се постиже програмом приказаним на слици 2.г.

```

LOAD      #3456h ; напуни акумулатор непосредном величином 3456h која представља
           виших 16 битова адресе
STORE 0000000Bh ; пребаци садржај акумулатора у меморијску локацију на адреси
           0000000Bh
LOAD      #789Ah ; напуни акумулатор непосредном величином 789Ah која представља
           нижих 16 битова адресе
STORE 0000000Ah ; пребаци садржај акумулатора у меморијску локацију на адреси
           0000000Ah

```

Слика 2.г Упис исте вредности у улаз за PER2 као и у улазу за PER3

Исти ефекат се остварује копирањем садржаја из улаза 7 IV табеле за периферију PER3 у улаз 5 IV табеле за периферију PER2. То се постиже програмом приказаним на слици 2.д.

```

LOAD      0000000Fh      ; напуни акумулатор садржајем меморијске локације са адресе
                                0000000Fh који представља виших 16 битова адресе
STORE 0000000Bh      ; пребаци садржај акумулатора у меморијску локацију на адреси
                                0000000Bh
LOAD      0000000Eh      ; напуни акумулатор садржајем меморијске локације са адресе
                                0000000Eh који представља нижих 16 битова адресе
STORE 0000000Ah      ; пребаци садржај акумулатора у меморијску локацију на адреси
                                0000000Ah

```

Слика 2.д Копирање вредности у улаз за PER2 из улаза за PER3

Други, могући начин да се постигне тражено је да се периферији PER2 додели исти број улаза у IV табелу као и периферији PER3. То се постиже програмом приказаним на слици 2.ђ.

```

LOAD      #7              ; напуни акумулатор непосредном величином 7 која представља
број
                                улаза у IV табелу за PER3
OUT      4h              ; пребаци садржај акумулатора у регистар контролера периферије
PER2

```

Слика 2.ђ Додела истог броја улаза у IV табелу за периферију PER2 као и за периферију PER3

И ови програми се извршавају током иницијализације система.

Напомена: Водеће нуле код децималних и хексадецималних величина није обавезно писати.

ђ) У улазу IV табеле се налази адреса прекидне дужине 32 бита, па су за њено смештање потребне две меморијске речи ширине 16 бита или 4 бајта. На основу величине IV табеле, која износи 2^6 улаза, и величине једног улаза, која износи 2^2 бајта, добија се да је максимална величина IV табеле

$$2^6 \cdot 2^2 \text{ бајта} = 2^8 \text{ бајта} = 256 \text{ бајта.}$$

Број улаза који се додељује некој периферији, у случају маскирајућих прекида, се чува у 16 разредном регистру контролера периферије. С обзиром да IV табела има 64 улаза, бројеви улаза у IV табелу су 6 битне вредности. Због тога би 16 битна вредност броја улаза која се уписује у регистар контролера периферије требало да буде у опсегу 0 до 2^6-1 и да на позицијама виших 10 битова садржи све нуле. Нормално нема никакве заштите од тога да се приликом уписа 16 битне вредности у регистар контролера периферије погрешно упише вредност која је већа од 63. Контролер периферије, када прими од процесора сигнал потврде *inta*, процесору шаље садржај 16 битног регистра са бројем улаза по линијама података магистрале којих има 16. Међутим, процесор од 16 битног податка са линија података магистрале користи само нижих 6 битова док виших 10 битова одбацује.

Вредности које се могу чувати у регистру контролера периферије су дужине 16 бита и спадају у интервал од 0 до $2^{16}-1$, па контролер периферије може процесору да проследи број улаза у IV табелу који је у наведеном интервалу. Уколико не би постојало никакво ограничење на нивоу процесора по питању броја улаза, максимална величина табеле би износила 2^{16} улаза. У том случају би процесор као број улаза у IV табелу користио свих 16

битова податка са линија података магистрале које му шаље контролер периферије када прими од процесора сигнал потврде *inta*.

У улазу IV табеле се налази адреса прекидне дужине 32 бита, па су за њено смештање потребне две меморијске речи ширине 16 бита или 4 бајта. У случају максималне величине IV табеле, који износи 2^{16} улаза, и величине једног улаза, која износи 2^2 бајта, добија се да је максимална величина IV табеле

$$2^{16} \cdot 2^2 \text{ бајта} = 2^{18} \text{ бајта} = 256\text{К бајта.}$$

1.3 ЗАДАТАК 3

Оперативна меморија неког рачунара је капацитета 64 КВ, а ширина речи меморије је 8 бита. Подаци ширине 16 бита се у меморију смештају у две суседне меморијске локације и то тако да се на нижој адреси налази виших 8 бита, а на вишој адреси нижих 8 бита податка. Процесор је једноадресни, акумулатор је дужине 8 бита и све инструкције се извршавају над 8 битним величинама. Улазно/излазни и меморијски адресни простори су раздвојени. Механизам прекида је векторисан. Адресе прекидних рутина се чувају у табели адреса прекидних рутина (IV табела - *Interrupt Vector Table*) која почиње на адреси 0h оперативне меморије и има 64 улаза.

Процесор има 4 улазне линије $IRQM_0$, $IRQM_1$, $IRQM_2$ и $IRQM_3$ на које се повезују спољашњи маскирајући захтеви за прекид. У оквиру спољашњи маскирајућих захтева за прекид највиши ниво приоритета има захтев за прекид који је везан на линију $IRQM_0$, затим захтев за прекид који је везан на линију $IRQM_1$, потом захтев за прекид који је везан на линију $IRQM_2$, док најнижи ниво приоритета има захтев за прекид који је везан на линију $IRQM_3$. Спољашњим маскирајућим захтевима за прекид који се повезују на линије $IRQM_0$, $IRQM_1$, $IRQM_2$ и $IRQM_3$ додељени су фиксно улази 4 до 7 у IV табели, респективно. Контролер периферије шаље процесору захтев за прекид *intr* као импулс.

Процесор има и једну улазну линију $IRQNM$ на коју се повезује спољашњи немаскирајући захтеви за прекид. Процесор може да генерише три унутрашња захтева за прекид и то захтев за прекид због грешке у коду операције, захтев за прекид због грешке у адресирању и захтев за прекид због задатог режима рада прекид после сваке инструкције. Захтеву за прекид због задатог режима рада прекид после сваке инструкције, спољњем немаскирајућем захтеву за прекид, захтеву за прекид због грешке у адресирању и захтеву за прекид због грешке у коду операције додељени су фиксно улази 0 до 3 у IV табели, респективно. Највиши приоритет има унутрашњи захтев за прекид због грешке у коду операције, затим следе унутрашњи захтев за прекид због грешке у адресирању, па спољашњи немаскирајући захтев за прекид и на крају са најнижим приоритетом долази унутрашњи захтев за прекид због задатог режима рада прекид после сваке инструкције.

У процесору постоји регистар *PSW (Program Status Word)* са битовима *I (Interrupt Enable)*, *T (Trap Enable)* и $L_{2...0}$ (*Priority Level*). Бит *I* вредношћу 1 дозвољава прихватање спољашњих маскирајућих захтева за прекид. Бит *T (Trap Enable)* вредношћу 1 дозвољава режим рада прекид после сваке инструкције. Битови $L_{2...0}$ садрже ниво приоритета текућег програма и користе се да се само приликом извршавања главног програма и прекидних рутина спољашњих маскирајућих захтева за прекид реализује механизам којим се прихватају спољашњих маскирајући захтеви за прекид једино уколико су вишег нивоа приоритета од нивоа приоритета текућег програма. Такође постоји и 4 битни регистар маске $IMR_{3...0}$ (*Interrupt Mask Register*) код кога вредност 1 на позицији бита i значи да је дозвољен прекид са линије $IRQM_i$, $i=0,1,2,3$. Овај регистар иницијално садржи све јединице.

На улазне линије процесора $IRQM_1$ и $IRQM_3$ су везани контролери периферија $PER1$ и $PER3$, респективно, док су линије $IRQM_0$ и $IRQM_2$ слободне. На једну од слободних линија $IRQM_0$ и $IRQM_2$ треба везати контролер периферије $PER2$. Везивање треба тако реализовати да се захтев за прекид од контролера периферије $PER2$ може прихватити и у току извршавања неке од прекидних рутина за периферије $PER1$ и $PER3$. Прекидне рутине периферија $PER1$ и $PER3$ почињу на адресама 100h и 300h, респективно.

а) Објаснити на који од два слободна улаза $IRQM_0$ и $IRQM_2$ треба везати сигнал захтева за прекид од контролера периферије PER2.

б) Написати део програма којим се иницијализује улаз периферије PER1 у IV табели.

в) Прекидна рутина периферије PER1 је приказана на слици 3.а.

```
PRPER1:   IN           FF00h
          STORE FF00h
          ...
          RTI
```

Слика 3.а Прекидна рутина периферије PER1

У току извршавања инструкције STORE стиже захтев за прекид од контролера периферије PER2. Објаснити да ли ће се овај захтев за прекид одмах прихватити.

г) Изменити дату прекидну рутину периферије PER1 тако да одговор на претходно питање буде супротан.

д) Објаснити шта треба урадити да извршавање прекидне рутине периферије PER3 не буде прекидано захтевом за прекид од контролера периферије PER1, али да буде прекидано захтевом за прекид од контролера периферије PER2?

ђ) Нацртати шему за генерисање броја улаза у IV табелу за спољашње маскирајуће захтеве за прекид.

е) Нацртати шему за генерисање броја улаза у IV табелу за спољашњи немаскирајући захтев за прекид и за унутрашње прекиде.

ж) На почетку извршавања фазе *опслуживање прекида* сваке инструкције врши се провера вредности сигнала `prekid` да би се утврдило да ли треба продужити са извршавањем ове фазе или треба одмах прећи на први корак фазе *читање инструкције* следеће инструкције. Претпоставити да у процесору постоји сви прекиди дати у опису процесора као и прекид изазван извршавањем инструкције прекида INS. Најпре нацртати и објаснити како се генерише сигнал `prekid` за све прекиде који постоје у датом процесору уз претпоставку да све инструкције процесора реагују на прекид. Затим нацртати и објаснити како се генерише сигнал `prekidnovi` за све прекиде који постоје у датом процесору уз претпоставку да све инструкције процесора реагују на прекид, осим инструкција INTE, INTD, RTI, TRPE и PRTD које не реагују на прекид.

Решење:

а) Да би захтев за прекид од контролера периферије PER2 могао бити прихваћен и у току извршавања неке од прекидних рутина периферија PER1 и PER3, захтев за прекид од контролера периферије PER2 треба да буде везан на ону од две слободне линије $IRQM_0$ или $IRQM_2$ чији је ниво приоритета виши од нивоа приоритета линија $IRQM_1$ и $IRQM_3$ на које су везани контролери периферија PER1 и PER3. Линија $IRQM_0$ је једина слободна линија чији је ниво приоритета виши од нивоа приоритета линија $IRQM_1$ и $IRQM_3$ на које су везани контролери периферија PER1 и PER3, док то није случај са линијом $IRQM_2$. Зато захтев за прекид од контролера периферије PER2 треба да буде везан на линију $IRQM_0$.

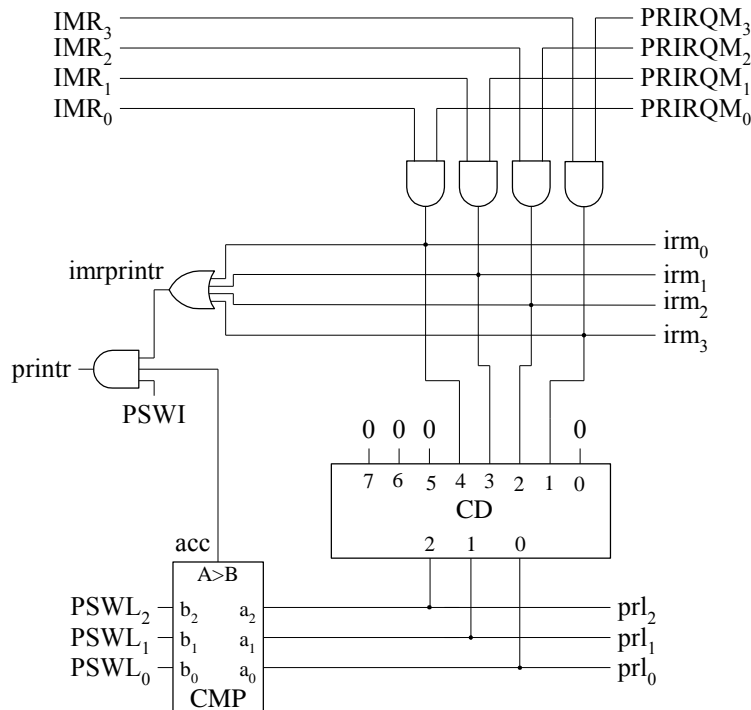
Спољашњи маскирајући захтеви за прекид који се повезују на улазне линије процесора $IRQM_0$, $IRQM_1$, $IRQM_2$ и $IRQM_3$ долазе као импулс, па се у процесору памте у флип-флоповима $PRIRQM_0$, $PRIRQM_1$, $PRIRQM_2$ и $PRIRQM_3$, респективно. У процесору постоји и регистар маске $IMR_{3..0}$ који се користи за селективно маскирање спољашњих маскирајућих захтева за прекид. Стога се коришћењем логичких I елемената од сигнала захтева за прекид $PRIRQM_0$, $PRIRQM_1$, $PRIRQM_2$ и $PRIRQM_3$, и сигнала маске IMR_0 ,

IMR_1 , IMR_2 и IMR_3 , формирају сигнали маскирајућих захтева за прекид са маском irm_0 , irm_1 , irm_2 и irm_3 (слика 3.б).

Сигнали irm_0 , irm_1 , irm_2 и irm_3 се даље користе за формирање сигнала спољашњег маскирајућег захтева за прекид $imrprirqm$, сигнала нивоа приоритета спољашњег маскирајућег захтева за прекид највишег нивоа приоритета $prl_{2..0}$ и сигнала прихватања спољашњег маскирајућег захтева за прекид acc .

Сигнал спољашњег маскирајућег захтева за прекид $imrprirqm$ се формира на излазу логичких ИЛИ елемента

Сигнали нивоа приоритета спољашњег маскирајућег захтева за прекид највишег нивоа приоритета $prl_{2..0}$ се формирају на излазу кодера приоритета CD.



Слика 3.б Прихватање спољашњих маскирајућих захтева за прекид

Да би се остварио механизам којим се

прекида извршавање главног програма уколико се јави било који од спољашњих маскирајућих захтева за прекид који није селективно маскиран одговарајућим битом регистра маске,

прекида извршавање неке прекидне рутине уколико постоји спољашњих маскирајући захтева за прекид који није селективно маскиран одговарајућим битом регистра маске вишег нивоа приоритета од нивоа приоритета текуће прекидне рутине и

обезбеђује да нивои приоритета захтева за прекид са линија $IRQM_0$, $IRQM_1$, $IRQM_2$ и $IRQM_3$ опадају од линије $IRQM_0$ која је највишег нивоа приоритета до линије $IRQM_3$ која је најнижег нивоа приоритета,

главном програму и прекидним рутинама захтева за прекид са линија $IRQM_3$, $IRQM_2$, $IRQM_1$ и $IRQM_0$, додељени су нивои приоритета 000b, 001b, 010b, 011b и 100b, респективно. У складу са тако додељеним нивоима приоритета и уз претпоставку да нивои приоритета кодера приоритета опадају од улаза 7 преко улаза 6, 5, 4, 3, 2 и 1 до улаза 0, сигнали irm_3 , irm_2 , irm_1 и irm_0 се везују на улазе 1, 2, 3 и 4 кодера приоритета CD.

Ниво приоритета текућег програма се чува у разредима PSWL_{2..0} регистра PSW. Током извршавања инструкције и то у оквиру фазе *опслуживање прекида* хардверски се најпре на стек стави садржај регистра PSW са вредностима разреда PSWL_{2..0} које представљају ниво приоритета текућег програма чије се извршавање прекида, а затим се у разреде PSWL_{2..0} уписују вредности сигнала prl_{2..0} који представљају сигнале нивоа приоритета спољашњег маскирајућег захтева за прекид највишег нивоа приоритета на чију се прекидну рутину прелази. На крају сваке прекидне рутине се програмским путем инструкцијом RTI и то у фази *извршавање операције* скидањем садржаја са стека и уписивањем у регистре PSW и PC у разреде PSWL_{2..0} враћа вредност нивоа приоритета прекинутог програма чије се извршавање наставља.

Сигнал прихватања спољашњег маскирајућег захтева за прекид асс се формира на излазу A>B компаратора CMP уколико је вредност сигнала prl_{2..0} већа од вредности сигнала PSWL_{2..0}.

Бит PSWI вредношћу 1 дозвољава прихватање спољашњих маскирајућих захтева за прекид.

Сигнал прихватања спољашњег маскирајућег захтева за прекид printr добија вредност 1 уколико вредност 1 имају сигнали imrprirqm, асс и PSWI. То значи да је за прекидање извршавања текућег програма потребно

да постоји барем један спољашњи захтев за прекид који није селективно маскиран одговарајућим разредом регистра маске IMR, па сигнал imrprirqm има вредност 1,

да је његов ниво приоритета већи од нивоа приоритета текућег програма, па сигнал асс има вредност 1, и

да нису маскирани спољашњи маскирајући захтеви за прекид, јер сигнал PSWI има вредност 1.

б) Капацитет оперативне меморије је 64 KB. Како је ширина речи меморије 8 бита, то оперативна меморија има 2^{16} адресибилних локација које садрже речи меморије ширине 8 бита. Стога су адресе које се користе за адресирање меморијских локација широке 16 бита.

Табела адреса прекидних рутина састоји се од 64 улаза који садрже почетне адресе прекидних рутина. Како су адресе широке 16 бита, а ширина речи меморије је 8 бита, сваки улаз табеле заузима две суседне локације у меморији. На локацији са нижом адресом се налази виших 8 бита адресе прекидне рутине, а на локацији са вишом адресом се налази нижих 8 бита адресе прекидне рутине. Почетна адреса IV табеле, која се чува у регистру IVTP (*Interrupt Vector Table Pointer*), је 0h.

Периферији PER1 додељен је улаз 5 у IV табели и у њега треба уписати адресу прекидне рутине периферије PER1 која износи 100h. Величина улаза у IV табели исказана у адресибилним јединицама је 2, јер је адреса широка 16 бита, а ширина речи меморије која је адресибилна јединица је 8 бита. Када се број улаза 5 који одговара периферији PER1 помножи са 2 добија се да је померај улаза 5 у односу на почетак IV табеле Ah. Када се померај Ah дода на садржај регистра IVTP у коме је IVTP почетна адреса IV табеле и који износи 0h, добија се да на адреси Ah треба да буде смештено виших 8 бита адресе прекидне рутине периферије PER1 (01h), док на адреси Bh треба да буде смештено нижих 8 бита адресе прекидне рутине периферије PER1 (0h). Тражени део програма је приказан на слици 3.в .

LOAD #1h ; напуни акумулатор непосредном величином 1h која представља виших 8 бита адресе прекидне рутине периферије PER1
STORE Ah ; пребаци садржај акумулатор на адресу Ah

LOAD #0h ; напуни акумулатор непосредном величином 0h која представља нижих 8 бита адресе прекидне рутине периферије PER1
 STORE Bh ; пребаци садржај акумулатор на адресу Bh

Слика 3.в Иницијализација улаза 5 у IV табели

в) У оквиру корака фазе *ослуживање прекида* инструкције текућег програма чије се извршавање прекида и прелази на прекидну рутину периферије, хардверски се бит I у регистру PSW (PSWI) поставља на вредност 0, па су по уласку у неку прекидну рутину сви маскирајући захтеви за прекид маскирани и не прихватају се. Зато се у прекидној рутини периферије PER1 чији је ниво приоритета 011b не прихвата пристигли захтев за прекид периферије PER2 чији је ниво приоритета виши и износи 100b (слика 3.б).

г) Да би у току прекидне рутине периферије PER1, чији је ниво приоритета 011b, могао да се прихвати пристигли захтев за прекид периферије PER2, чији је ниво приоритета виши и износи 100b, потребно је на почетку прекидне рутине периферије PER1 софтверски инструкцијом INTE бит I регистра PSW (PSWI) поставити на вредност 1. Измењена прекидна рутина периферије PER1 је приказана на слици 3.г.

```
PRPER1:  INTE                ; уписивање вредности 1 у бит I регистра PSW (PSWI) чиме се
          ; дозвољава прихватање спољашњих маскирајућих захтева за прекид
          IN                  FF00h
          STORE FF00h
          ...
          RTI
```

Слика 3.г Измењена прекидна рутина периферије PER1

д) Да би захтев за прекид од контролера периферије могао да прекида извршавање текућег програма потребно је

да је његов ниво приоритета виши од нивоа приоритета текућег програма,

да спољашњи маскирајући захтеви за прекид нису маскирани вредношћу 0 разреда I регистра PSW и

да дати захтев за прекид није селективно маскиран вредношћу 0 одговарајућег разреда регистра маске IMR.

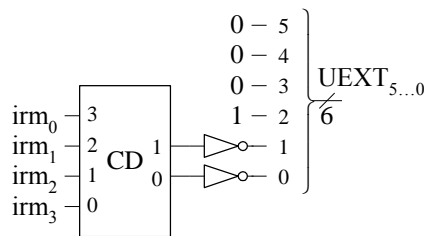
Ниво приоритета прекидне рутине периферије PER3 чији захтев за прекид долази по линији IRQM₃ је 001b, ниво приоритета захтев за прекид периферије PER1 који долази по линији IRQM₁ је 011b и ниво приоритета захтев за прекид периферије PER2 који долази по линији IRQM₀ је 100b. По основу нивоа приоритета извршавање прекидне рутине периферије PER3 би могло да се прекида захтевима за прекид контролера периферија PER1 и PER2.

Међутим, да би извршавање прекидне рутине периферије PER3 могло да се прекида захтевом за прекид контролера периферије PER2, потребно је и да се у разреду I регистра PSW налази вредност 1 и да се у одговарајућем разреду регистра маске IMR налази вредност 1.

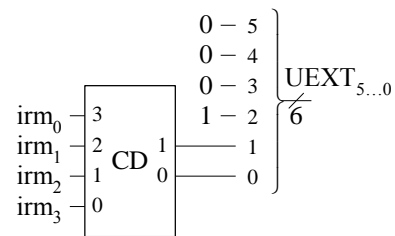
Вредност 1 у разреду I регистра PSW, која је неопходна да би се омогућило да захтев за прекид од контролера периферије PER2 прекида извршавање прекидне рутине периферије PER3, омогућава и да захтев за прекид од контролера периферије PER1 прекида извршавање прекидне рутине периферије PER3. Да извршавање прекидне рутине периферије PER3 не би могло да се прекида захтевом за прекид контролера периферија PER1, потребно је да се у одговарајућем разреду регистра маске IMR налази вредност 0.

Периферији PER3 чији захтев за прекид долази по линији IRQM₃ одговара разред IMR₃ регистра маске, периферији PER1 чији захтев за прекид долази по линији IRQM₁ одговара разред IMR₁ регистра маске и периферији PER2 чији захтев за прекид долази по линији IRQM₀ одговара разред IMR₀ регистра маске. Да би захтевом за прекид од контролера периферије PER3 могло да се из главног програма пређе на прекидну рутину периферије PER3 потребно је да у разреду IMR₃ буде вредност 1, да би извршавање прекидне рутине периферије PER3 могло да се прекида захтевом за прекид од контролера периферије PER2 потребно је да у разреду IMR₀ буде вредност 1, и да извршавање прекидне рутине периферије PER3 не би могло да се прекида захтевом за прекид од контролера периферије PER1 потребно је да у разреду IMR₁ буде вредност 0. Вредност разреда IMR₂ није битна јер по линији IRQM₂ којој одговара разред IMR₂ не долазе захтеви за прекид.

ђ) На слици 3.д је дата шема за генерисање броја улаза у IV табелу за спољашње маскирајуће захтеве за прекид UEXT_{5...0} који се повезују на линије IRQM₀, IRQM₁, IRQM₂ и IRQM₃ и којима су додељени фиксни улази 4, 5, 6 и 7 у IV табели, респективно. Да би се поједноставило генерисање броја улаза за спољашње маскирајуће захтеве за прекид којих може да буде четири, узима се да четири улаза буду груписана у складу са приоритетима који су додељени линијама IRQM₀, IRQM₁, IRQM₂ и IRQM₃ и да заузимају улазе 0 до 3 или улазе 4 до 7 или улазе 8 до 11 итд. У том случају се 6 битна вредност броја улаза UEXT_{5...0} генерише тако што се два нижа бита 1 и 0 генеришу на основу нивоа приоритета линија IRQM₀, IRQM₁, IRQM₂ и IRQM₃ по којима стижу захтеви за прекид, док се четири виша бита 5, 4, 3 и 2 фиксно генеришу и за улазе 0 до 3 износе 0000b, за улазе 4 до 7 износе 0001b, за улазе 8 до 11 износе 0010b итд. На улазе кодера приоритета се доводе сигнали маскирајућих захтева за прекид после маскирања irm₀, irm₁, irm₂ и irm₃ (слика 3.б) јер приликом формирања броја улаза треба узети у обзир само оне захтеве за прекид који нису селективно маскирани одговарајућим разредом регистра маске IMR.



Слика 3.д Генерисање броја улаза у IV табелу за спољашње маскирајуће захтеве за прекид



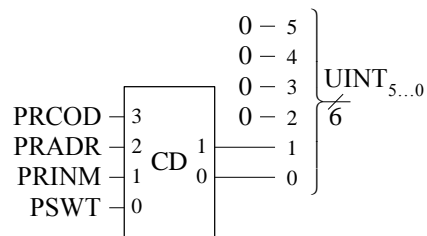
Слика 3.ђ Генерисање броја улаза у IV табелу за спољашње маскирајуће захтеве за прекид

Код формирања два најнижа бита броја улаза треба водити рачуна и о приоритетима сигнала irm₀, irm₁, irm₂ и irm₃ чији приоритети одговарају нивоима приоритета линија IRQM₀, IRQM₁, IRQM₂ и IRQM₃ и опадају од сигнала irm₀ који има највиши ниво приоритета преко сигнала irm₁ и irm₂ којима опадају приоритети до сигнала irm₃ који има најнижи ниво приоритета. Због тога, уз претпоставку да нивои приоритета кодера приоритета опадају од улаза 3 преко улаза 2 и 1 до улаза 0, сигнале irm₀, irm₁, irm₂ и irm₃ треба довести на улазе 3, 2, 1 и 0 кодера приоритета, респективно, а за два најнижа бита броја улаза треба узети инвертоване вредности сигнала са излаза кодера приоритета.

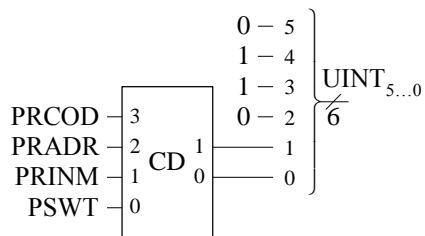
Уз претпоставку да сигнал irm_0 има вредност 1 и без обзира на то које су вредности сигнала irm_1 , irm_2 и irm_3 , на излазу кодера приоритета ће бити 11b, на излазима инвертора ће бити 00b, па ће се за број улаза на линијама $UEXT_{5..0}$ формирати захтевана вредност 000100b. Уз претпоставку да сигнал irm_0 има вредност 0 а сигнал irm_1 вредност 1 и без обзира на то које су вредности сигнала irm_2 и irm_3 , на излазу кодера приоритета ће бити 10b, на излазима инвертора ће бити 01b, па ће се за број улаза на линијама $UEXT_{5..0}$ формирати захтевана вредност 000101b. На сличан начин се формира број улаза 000110b када сигнали irm_0 и irm_1 имају вредност 0, сигнал irm_2 има вредност 1 док вредност сигнала irm_3 није битна и број улаза 000111b када сигнали irm_0 , irm_1 и irm_2 имају вредност 0 и сигнал irm_3 има вредност 1.

Да су спољашњим маскирајућим захтевима за прекид који се повезују на линије $IRQM_0$, $IRQM_1$, $IRQM_2$ и $IRQM_3$ додељени фиксни улази 7, 6, 5 и 4 у IV табели, респективно, а нивои приоритета сигнала irm_0 , irm_1 , irm_2 и irm_3 чији приоритети одговарају нивоима приоритета линија $IRQM_0$, $IRQM_1$, $IRQM_2$ и $IRQM_3$ остали исти и опадају од сигнала irm_0 који има највиши ниво приоритета преко сигнала irm_1 и irm_2 којима опадају приоритети до сигнала irm_3 који има најнижи ниво приоритета, шема за генерисање броја улаза у IV табелу $UEXT_{5..0}$ би била као на слици 3.ђ.

е) На слици 3.е је дата шема за генерисање броја улаза у IV табелу за спољашњи немаскирајући захтев за прекид и за унутрашње прекиде $UINT_{5..0}$. Процесор има и једну улазну линију $IRQNM$ на коју се повезују спољашњи немаскирајући захтеви за прекид, а може да генерише три унутрашња захтева за прекид и то захтев за прекид због грешке у коду операције, захтев за прекид због грешке у адресирању и захтев за прекид због задатог режима рада прекид после сваке инструкције. Захтеву за прекид због задатог режима рада прекид после сваке инструкције, спољашњем немаскирајућем захтеву за прекид, захтеву за прекид због грешке у адресирању и захтеву за прекид због грешке у коду операције додељени су фиксно улази 0 до 3 у IV табели, респективно. Највиши приоритет има унутрашњи захтев за прекид због грешке у коду операције, затим следе унутрашњи захтев за прекид због грешке у адресирању, па спољашњи немаскирајући захтев за прекид и на крају са најнижим приоритетом долази унутрашњи захтев за прекид због задатог режима рада прекид после сваке инструкције.



Слика 3.е Генерисање броја улаза у IV табелу за спољашњи немаскирајући захтев за прекид и за унутрашње прекиде



Слика 3.ж Генерисање броја улаза у IV табелу за спољашњи немаскирајући захтев за прекид и за унутрашње прекиде

Захтев за унутрашњи прекид због грешке у коду операције се генерише тако што процесор у флип-флоп $PRCOD$ упише вредност 1 уколико током извршавања фазе *читање инструкције* открије да се у пољу кода операције појавила вредност која није додељена ни једној операцији. Захтев за унутрашњи прекид због грешке у адресирању се генерише тако што процесор у флип-флоп $PRADR$ упише вредност 1 уколико током извршавања фазе

формирање адресе и читање операнда открије да се непосредно адресирање користи за одредишни операнд. Захтев за спољашњи немаскирајући захтев за прекид се генерише уписивањем вредности 1 у флип-флоп PRINM када се на улазној линији процесора IRQNM појави импулс којим спољашњи уређај који контролише исправност рада делова рачунара сигнализира процесору да је откривена нека нерегуларност. Захтев за унутрашњи прекид због задатог режима рада прекид после сваке инструкције се генерише софтверски у фази *извршавање операције* инструкције TRPE тако што се у бит T регистра PSW (PSWT) упише вредност 1.

Да би се поједноставило генерисање броја улаза за ове захтеве за прекид којих има четири, четири улаза су груписана у складу са приоритетима који су додељени захтевима у флип-флоповима PSWT, PRINM, PRADR и PRCOD. У том случају се 6 битна вредност броја улаза UINT_{5..0} генерише тако што се два нижа бита 1 и 0 генеришу на основу нивоа приоритета прекида PSWT, PRINM, PRADR и PRCOD, док се четири виша бита 5, 4, 3 и 2 фиксно генеришу и за улазе 0 до 3 износе 0000b.

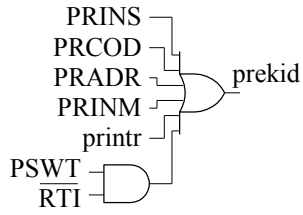
Да су захтеву за прекид због задатог режима рада прекид после сваке инструкције, спољашњем немаскирајућем захтеву за прекид, захтеву за прекид због грешке у адресирању и захтеву за прекид због грешке у коду операције додељени су фиксно улази 12 до 15 у IV табели, респективно, 6 битна вредност броја улаза UINT_{5..0} би се генерисала на начина приказан на слици 3.ж.

ж) На сликама 3.з и 3.и је дато како се генерише сигнал *prekid* за случај процесора у коме све инструкције реагују на прекид и сигнал *prekidnovi* за случај процесора у коме све инструкције реагују на прекид, осим инструкција INTE, INTD, RTI, TRPE и PRTD које не реагују на прекид.

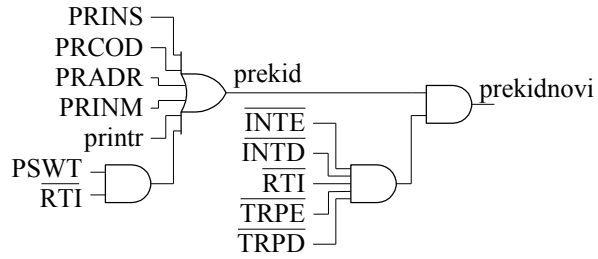
Сигнал *prekid* се генерише као ИЛИ функција сигнала PRINS, PRCOD, PRADR, PRINM, *printr* и $PSWT \cdot \overline{RTI}$. Сигнал PRINS долази са флип-флопа PRINS у који се током фазе *извршавање операције* инструкције прекида INS уписује вредност 1. Генерисање сигнала PRCOD, PRADR, PRINM и PSWT је објашњено под е), генерисање сигнала *printr* је објашњено под њ), док је са RTI означен сигнал операције инструкције повратак из прекидне рутине. Треба уочити да уз сигнал PSWT иде и комплемент сигнала RTI. Тиме током извршавања инструкције RTI, када комплемент сигнала RTI има вредност 0, вредност 1 сигнала PSWT не утиче на то да сигнал *prekid* постане 1, чиме се спречава да инструкција RTI реагује на прекид због задатог режима рада прекид после сваке инструкције. Тиме се омогућује да се инструкцијом RTI реализује повратак из прекидне рутине која одговара прекиду због задатог режима рада прекид после сваке инструкције у прекинути програм. Да то није учињено, у фази *извршавање операције* инструкције RTI би се рестаурирала вредност регистра PSW и том приликом би се у бит PSWT уписала вредност 1, а затим би се у фази *опслуживање прекида* инструкције RTI поново реализовао прелаз на исту прекидну рутину која одговара прекиду због задатог режима рада прекид после сваке инструкције.

Да би се обезбедило да инструкције INTE, INTD, RTI, TRPE и PRTD не реагују на прекид, на почетку извршавања фазе *опслуживање прекида* сваке инструкције треба проверавати уместо вредности сигнала *prekid* вредност сигнала *prekidnovi*. Током извршавања инструкција INTE, INTD, RTI, TRPE и TRPD због вредности 1 једног од сигнала INTE, INTD, RTI, TRPE и TRPD, сигнал *prekidnovi* има вредност 0. Због тога се током извршавања инструкција INTE, INTD, RTI, TRPE и TRPD провером вредности

сигнала *prekidnovi* на почетку фазе *ослуживање прекида* утврђује да не треба продужити са извршавањем ове фазе већ да треба одмах прећи на први корак фазе *читање инструкције* следеће инструкције.



Слика 3.з Генерисање сигнал *prekid* када све инструкције процесора реагују на прекид



Слика 3.и Генерисање сигнал *prekidnovi* када све инструкције процесора реагују на прекид осим инструкција *INTE, INTD, RTI, TRPE* и *PRTD*

1.4 ЗАДАТАК 4

Оперативна меморија неког рачунара је капацитета 16 GB, а ширина речи меморије је 32 бита. Процесор је једноадресни, акумулатор је 32 битни, а механизам прекида је векторисан. Адресе прекидних рутина се чувају у табели адреса прекидних рутина (IV табела - *Interrupt Vector Table*) која почиње на адреси 2h оперативне меморије и има 4 фиксна улаза.

У рачунару могу да се генеришу један спољашњи маскирајући захтев за прекид који долази у процесор преко улазне линије IRQM као импулс, један спољашњи немаскирајући захтев за прекид прекид који долази у процесор преко улазне линије IRQN као импулс, унутрашњи захтев за прекид због задатог режима рада прекид после сваке инструкције и унутрашњи прекид због грешке у адресирању. Спољашњи маскирајући захтев за прекид долази од контролера периферије, спољни немаскирајући захтев за прекид долази од уређаја који контролише исправност рада делова рачунара, унутрашњи захтев за прекид због задатог режима рада прекид после сваке инструкције постоји уколико бит Т (Trap enable) регистра PSW има вредност 1 и унутрашњи прекид због грешке у адресирању се јавља током извршавања инструкције STORE у којој је специфицирано непосредно адресирање. Адресе прекидних рутина за спољашњи маскирајући захтев за прекид, спољашњи немаскирајући захтев за прекид, унутрашњи захтев за прекид због задатог режима рада прекид после сваке инструкције и унутрашњи захтев за прекид због грешке у адресирању се налазе у улазима 0, 1, 2 и 3 IV табеле, респективно. Највиши приоритет има унутрашњи захтев за прекид због грешке у адресирању, затим следе спољашњи немаскирајући захтев за прекид, па спољашњи маскирајући захтев за прекид и на крају са најнижим приоритетом долази унутрашњи захтев за прекид због задатог режима рада прекид после сваке инструкције.

Пошто постоји само једна линија IRQM по којој долази масакирајући захтев за прекид, нема потребе за регистром маске IMR, већ се маскирање тог захтева за прекид реализује битом I (Interrupt enable) регистра PSW који служи за маскирање свих маскирајућих захтева за прекид. Такође, с обзиром да постоји само једна линија по којој долази маскирајући захтев за прекид за који мора да постоји могућност да прекида главни програм, нема потребе да у регистру PSW постоје битови L (Level) за чување приоритета текућег програма.

Битови I (Interrupt enable) и T (Trap enable) регистра PSW се постављају на вредност 0 хардверски током извршавања фазе *опслуживање прекида*. Инструкције INTE, INTD, RTI, TRPE и TRPD не реагују на захтеве за прекид. У оквиру извршавања фазе *опслуживање прекида* на стеку се хардверски чувају PSW и PC тим редом. Стек расте према нижим локацијама, а указивач на врх стека SP указује на задњу заузету локацију. Претпоставити да се у случају унутрашњег прекида због грешке у адресирању поступа на исти начин као и у случају осталих прекида и да програмски бројач PC приликом преласка на одговарајућу прекидну рутину указује на прву следећу инструкцију.

Изглед дела главног програма је дат на слици 4.а, прекидних рутина на слици 4.б и дела меморије почев од адресе 0 на слици 4.в. Инструкција INTE на адреси 0100h означена је као 1. (прва) по редоследу извршавања, а свака следећа инструкција која се извршава означена је следећим редним бројем. У току извршавања 2. инструкције стиже захтев за прекид по линији IRQN, а у току 5. инструкције по линији IRQM. На почетку извршавања главног програма су сви бити регистра програмске статусне речи PSW-а постављени на 0.

Напомена: Код писања 32 битних адреса на сликама 4.а, 4.б. и 4.в. и 32 битних садржаја на слици 4.в. у хексадецималном облику ради прегледности су изостављене четири водеће нуле. Поред тога и у тексту су изостављене све водеће нуле код писања 32 битних адреса и 32 битних садржаја.

адреса	инструкција
0100h	INTE
0101h	LOAD #1h
0103h	INC
0104h	STORE #1h
0106h	DEC
0107h	INTD

Слика 4.а Део главног програма

адреса	инструкција
1000h	INC
1001h	PUSH
1002h	POP
1003h	RTI
1004h	LOAD 1h
1006h	OR #FFh
1008h	STORE 1h
100Ah	RTI
100Bh	POP
100Ch	PUSH
100Dh	RTI

Слика 4.б. Прекидне рутине

адреса	садржај
0000h	1001h
0001h	100Ah
0002h	1000h
0003h	1004h
0004h	1008h
0005h	100Bh

Слика 4.в. Изглед дела меморије почев од адресе 0h

а) Написати на којим адресама започињу прекидне рутине за захтеве за прекид који долазе по линијама IRQM и IRQN, респективно.

б) Нацртати и објаснити како се у датом процесору генерише сигнал од чије вредности зависи да ли ће се на крају фазе *извршавање операције* неке инструкције продужити са фазом *ослуживање прекида* и у оквиру ње реализовати прелаз на одговарајућу прекидну рутину или прећи на фазу *читање инструкције* прве следеће инструкције текућег програма.

в) Нацртати и објаснити како процесор фиксно генерише бројеве улаза у IV табелу.

г) Написати секвенцу адреса инструкција које се редом извршавају почев од адресе 0100h.

д) Навести која ће се вредност налазити на локацији 1h након извршења секвенце инструкција из тачке г).

Решење:

а) Капацитет оперативне меморије је 16 GB. Како је ширина речи меморије 32 бита, то оперативна меморија има 2^{32} адресибилних локација које садрже речи меморије ширине 32 бита. Стога су адресе које се користе за адресирање меморијских локација широке 32 бита.

Табела адреса прекидних рутина састоји се од 4 улаза који садрже почетне адресе прекидних рутина. Како су адресе широке 32 бита, а ширина речи меморије је 32 бита, сваки улаз табеле заузима једну локацију у меморији. Почетна адреса IV табеле, која се чува у регистру IVTP (*Interrupt Vector Table Pointer*), је 2h.

У датом случају величина улаза у IV табели исказана у адресибилним јединицама је 1, јер је адреса широка 32 бита, а ширина речи меморије која је адресибилна јединица је 32 бита. Ако се број улаза i који одговара некој периферији PER j помножи са 1 и та вредност дода на садржај регистра IVTP који износи 2h добија се адреса a на којој је смештена адреса прекидне рутине периферије PER j .

Адресе прекидних рутина за спољашњи маскирајући захтев за прекид који долази у процесор преко улазне линије IRQM и спољашњи немаскирајући захтев за прекид који долази у процесор преко улазне линије IRQN се налазе у улазима 0 и 1 IV табеле, респективно. Када се бројеви улаза 0 и 1 помноже са 1 и добијени помераји 0 и 1 додају на садржај регистра IVTP који износи 2h добија се да се адресе на којима се налазе адресе прекидних рутина за захтеве за прекид који долазе по линијама IRQM и IRQN налазе у меморијским локацијама чије су адресе 2h и 3h, респективно. Са слике 4.в. се добија да садржаји меморијских локација са адреса 2h и 3h чије су вредност 0000 1000h и 0000 1004h представљају адресе прекидних рутина за захтеве за прекид који долазе по линијама IRQM и IRQN, респективно.

б) Спољашњи маскирајући захтеви за прекид који се повезују на улазну линију процесора IRQM долази као импулс, па се у процесору памти у флип-флопу PRIRQM. Спољашњи немаскирајући захтев за прекид се генерише уписивањем вредности 1 у флип-флоп PRIRQN када се на улазној линији процесора IRQN појави импулс којим спољашњи уређај који контролише исправност рада делова рачунара сигнализира процесору да је откривена нека нерегуларност. Унутрашњи захтев за прекид због грешке у адресирању се генерише тако што процесор у флип-флоп PRADR упише вредност 1 уколико током извршавања фазе *формирање адресе и читање операнда* открије да се непосредно адресирање користи за одредишни операнд. Унутрашњи захтев за прекид због задатог режима рада прекид после сваке инструкције се генерише софтверски у фази *извршавање операције* инструкције TRPE тако што се у бит Т регистра PSW (PSWT) упише вредност 1.

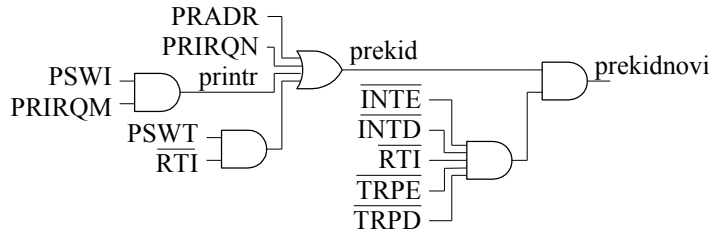
Маскирање масакирајућег захтева за прекид који долази по линији IRQM се реализује битом I (Interrupt enable) регистра PSW који служи за маскирање свих маскирајућих захтева за прекид.

Инструкције INTE, INTD, RTI, TRPE и TRPD не реагују на захтеве за прекид.

Сигнал `prekidnovi` се генерише на начин приказан на слици 4.г. На почетку извршавања фазе *опслуживање прекида* сваке инструкције врши се провера вредности овог сигнала и у зависности од тога да ли има вредност 1 или 0 или продужава са преосталима корацима фазе *опслуживање прекида* којима се реализује прелаз на одговарајућу прекидну рутину или прелази на фазу *читање инструкције* прве следеће инструкције текућег програма.

Генерисање сигнала `prekidnovi` је објашњено у задатку 3.ж. Генерисање сигнала `print` на слици 4.г. је једноставније него у задатку 3.ж. Због тога што постоји само једна линија IRQM по којој долази маскирајући захтев за прекид нема потребе да се као у задатку 3.ж И елементима врши селективно маскирањем маскирајућих захтева за прекид разредима

регистра маске IMR и нема потребе да се кодером приоритета утврђује ниво приоритета маскирајућег захтева за прекид највишег нивоа приоритета и да се компаратором утврђује да ли је он виши од нивоа приоритета текућег програма. Због тога, под условом да није маскиран битом I регистра PSW, маскирајући захтев за прекид увек прекида извршавање текућег програма.

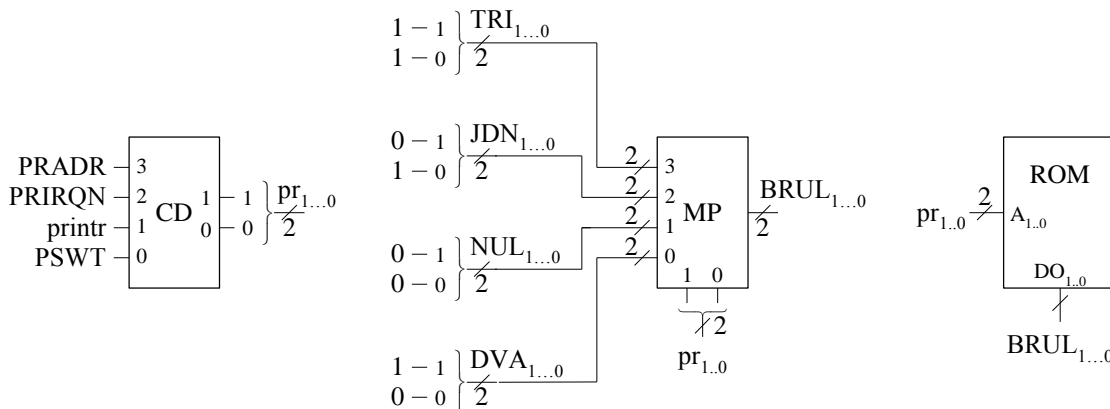


Слика 4.г Генерисање сигнал prekidnovi када све инструкције процесора реагују на прекид осим инструкција INTE, INTD, RTI, TRPE и PRTD

в) На слици 4.д је дата шема за генерисање броја улаза у IV табелу.

Адресе прекидних рутина за спољашњи маскирајући захтев за прекид (printr), спољашњи немаскирајући захтев за прекид (PRIRQN), унутрашњи захтев за прекид због задатог режима рада прекид после сваке инструкције (PSWT) и унутрашњи захтев за прекид због грешке у адресирању (PRADR) се налазе у улазима 0, 1, 2 и 3 IV табеле, респективно. Највиши приоритет има прекид PRADR, затим следе прекид PRIRQN, па прекид printr и на крају са најнижим приоритетом долази прекид PSWT.

Најпре је потребно помоћу кодера приоритета CD формирати сигнале приоритета $pr_{1..0}$ прекида највишег нивоа приоритета, а затим њих користити за формирање броја улаза $BRUL_{1..0}$ (слика 4.д). Један начин за формирање броја улаза $BRUL_{1..0}$ је да се једна од четири вредности броја улаза $TRI_{1..0}$, $JDN_{1..0}$, $NUL_{1..0}$ или $DVA_{1..0}$ селектује сигнаlima $pr_{1..0}$ на излазе мултиплексера MP. Други начин за формирање броја улаза $BRUL_{1..0}$ је да се једна од четири вредности броја улаза чита и појави на излазним линијама података меморије ROM са адресе одређене вредностима сигнала $pr_{1..0}$, при чему су претходно у улазе 0, 1, 2 и 3 уписане вредности броја улаза 10b, 00b, 01b и 11b, респективно.



Слика 4.д Генерисање броја улаза у IV табелу

Одређивање броја улаза би било једноставније да се прекидима PRADR, PRIRQN, printr и PSWT у складу са опадајућим приоритетима додељени улази 3, 2, 1 и 0, респективно. У

том случају би сигнали приоритета $rg_{1..0}$ прекида највишег нивоа приоритета са излаза кодера приоритета CD представљали броја улаза $BRUL_{1..0}$ (слика 4.д).

г) Секвенца адреса инструкција које се редом извршавају, почев од инструкције INTE са адресе 0100h главног програма и закључно са инструкцијом INTD са адресе 0107h главног програма, приказана је на слици 4.ђ, карактеристичне ситуације на стеку су дате на слици 4.е, а објашњења у даљем тексту.

У табели са слике 4.ђ су у посебним колонама дати: Рб – редни број извршене инструкције, Адреса – адреса инструкције у хексадецималном облику (четири водеће нуле су изостављене), Инструкција – симболичка ознака инструкције и начина адресирања, ACC – садржај акумулатора ACC, Стек – ситуација на стеку са слике 4.е, I – вредност бита I регистра програмске статусне речи PSW, T – вредност бита T регистра програмске статусне речи PSW, PRIRQN – спољашњи немаскирајући захтев за прекид, PRIRQM – спољашњи маскирајући захтев за прекид и PRADR - унутрашњи захтев за прекид због грешке у адресирању. У табели је за сваку инструкцију дата једна врста са стањем на крају фазе *извршавање инструкције*, а за инструкције које извршавају и фазу *опслуживање прекида* дата је још једна врста са стањем на крају фазе *опслуживање прекида*.

Почетна ситуација на стеку је дата као Ситуација 1. Регистар SP указује на задњу заузету меморијску локацију на адреси А која представља врх стека и чији је садржај небитан, па је означен са xxxx xxxx. Сви садржаји на стеку су дати у хексадецималном облику, а од садржаја програмске статусне речи PSW су дате само појединачне вредности битова I и T, јер су они једино релевантни.

Инструкција 1: Почетне вредности регистара и индикатора су: PC=100h, ACC=?, I=0, T=0.

Ради се о инструкцији INTE са адресе 100h, која у бит I програмске статусне речи PSW уписује вредност 1.

Инструкција не реагује на прекид и прелази се на извршавање следеће инструкције.

Инструкција 2: Почетне вредности регистара и индикатора су: PC=101h, ACC=?, I=1, T=0.

Ради се о инструкцији LOAD #1 са адресе 101h, која у акумулатор ACC уписује вредност 1. Током извршавања инструкције јавља се спољашњи немаскирајући прекид на линији IRQN којим се сигнализира процесору да се догодио неки значајан догађај који је потребно што пре обрадити, као на пример проблем са напајањем, конфликт на магистралама и слично, па се флип-флоп PRIRQN поставља на 1.

Инструкција реагује на прекид и постоји прекид. На прекиде се реагује по опадајућим приоритетима. Пошто нема унутрашњег прекида због грешке у адресирању PRADR који представља прекид највишег нивоа приоритета, процесор реагује на спољашњи немаскирајући прекид PRIRQN који представља први следећи прекид по опадајућим приоритетима. Стога се у оквиру фазе *опслуживање прекида* најпре се на стеку чува контекст процесора програма чије се извршавање прекида, а затим се формира контекст процесора прекидне рутине на чије се извршавање прелази.

У корацима чувања контекста процесора програма чије се извршавање прекида на стеку се чува прво програмска статусна реч PSW, а потом и програмски бројач PC. Код програмске статусне речи PSW која се чува на стеку бит I има вредност 1 и бит T има вредност 0. Програмски бројач PC има вредност 103h, која показује на прву следећу инструкцију програма. Програмски бројач PC је дужине 32 бита и приликом његовог стављања на стек његов садржај се смешта у једну меморијску локацију јер је ширина

једне меморијске локације 32 бита. У једну меморијску локацију се смешта и садржај програмске статусне речи PSW, јер се из поставке задатка може утврдити да поред уобичајених битова N, Z, C и V (задатак) постоје још само битови I и T и да је дужина PSW мања од 32 бита.

У корацима формирања контекста процесора прекидне рутине на чије се извршавање прелази потребно је поставити нове вредности у програмску статусну реч PSW и програмски бројач PC. У бит I регистра PSW се уписује вредност 0, а у бит T регистра PSW се такође уписује вредност 0. Улази у IV табелу су фиксни, па процесор, пошто се ради о немаскирајућем прекиду PRIRQN, најпре утврђује да број улаза износи 1 (слика 4.д), а затим флип-флоп PRIRQN брише на 0. Адреса улаза 1 у IV табели је 3 ($3 = 2 + 1 \cdot 1$), па процесор најпре са адресе 3 чита вредност 0000 1004h, која представља адресу прекидне рутине, а затим ту вредност уписује у програмски бројач PC и прелази на фазу *читање инструкције* прве инструкције прекидне рутине. На стеку је Ситуација 2 (слика 4.е).

Инструкција 3: Почетне вредности регистара и индикатора су: PC=1004h, ACC=1, I=0, T=0.

Ради се о инструкцији LOAD 1 са адресе 1004h, која у акумулатор ACC уписује садржај са адресе 1, а тамо се налази 100Ah.

Инструкција реагује на прекид, али не постоји прекид, па се прелази на извршавање следеће инструкције.

Инструкција 4: Почетне вредности регистара и индикатора су: PC=1006h, ACC=100Ah, I=0, T=0.

Ради се о инструкцији OR #FFh са адресе 1006h, која реализује логичку ИЛИ операцију садржаја акумулатора 0000 100Ah и непосредне величине 0000 00FFh и у акумулатор уписује вредност 0000 10FFh. Ефекат ове операције је да се најнижих осам бита акумулатора поставља на вредност 1, док се преосталих 14 виших битова акумулатора не мењају.

Инструкција реагује на прекид, али не постоји прекид, па се прелази на извршавање следеће инструкције.

Инструкција 5: Почетне вредности регистара и индикатора су: PC=1008h, ACC=10FFh, I=0, T=0.

Ради се о инструкцији STORE 1 са адресе 1008h, која у меморијску локацију на адреси 1 уписује садржај акумулатора чија је вредност 10FFh. Током извршавања инструкције јавља се спољашњи маскирајући прекид на линији IRQM, па се флип-флоп PRIRQM поставља на 1.

Инструкција реагује на прекид. Међутим, унутрашњег прекида PRADR нема, спољашњег немаскирајућег прекида PRIRQN нема, спољашњи маскирајући прекид PRIRQM постоји али је маскиран вредношћу 0 бита PSWI па сигнал printr има вредност 0 и унутрашњег прекида PSWT нема, па се прелази на извршавање следеће инструкције.

Инструкција 6: Почетне вредности регистара и индикатора су: PC=100Ah, ACC=10FFh, I=0, T=0.

Ради се о инструкцији RTI са адресе 100Ah, која са стека скида контекст који је сачуван на стеку приликом преласка на ову прекидну рутину и уписује у програмски бројач PC и у програмску статусну реч PSW. Регистар PC добија вредност 103h, док у регистру PSW бит I добија вредност 1, а бит T вредност 0.

Инструкција не реагује на прекид, па се прелази се на извршавање следеће инструкције. На стеку је Ситуација 3 (слика 4.е).

Инструкција 7: Почетне вредности регистра и индикатора су: PC=103h, ACC=10FFh, I=1, T=0.

Ради се о инструкцији INC са адресе 103h, која увећава садржај акумулатора ACC за један, па садржај акумулатора постаје 0000 1100h.

Инструкција реагује на прекид и постоји прекид. Унутрашњег прекида PRADR нема, спољашњег немаскирајућег прекида PRIRQN нема, али спољашњи маскирајући прекид PRIRQM постоји и сада није маскиран јер бит PSWI има вредност 1 па и сигнал printg има вредност 1. Стога се у оквиру фазе *опслуживање прекида* најпре се на стеку чува контекст процесора програма чије се извршавање прекида, а затим се формира контекст процесора прекидне рутине на чије се извршавање прелази.

У корацима чувања контекста процесора програма чије се извршавање прекида на стеку се чува прво програмска статусна реч PSW, а потом и програмски бројач PC. Код програмске статусне речи PSW која се чува на стеку бит I има вредност 1 и бит T има вредност 0. Програмски бројач PC има вредност 104h, која показује на прву следећу инструкцију програма чије се извршавање прекида.

У корацима формирања контекста процесора прекидне рутине на чије се извршавање прелази потребно је поставити нове вредности у програмску статусну реч PSW и програмски бројач PC. У бит I регистра PSW се уписује вредност 0, а у бит T регистра PSW се такође уписује вредност 0. Улази у IV табелу су фиксни, па процесор, пошто се ради о маскирајућем прекиду PRIRQM, најпре утврђује да број улаза износи 0 (слика 4.д), а затим флип-флоп PRIRQM брише на 0. Адреса улаза 0 у IV табели је 2 ($2 = 2 + 0 \cdot 1$), па процесор најпре са адресе 2 чита вредност 0000 1000h, која представља адресу прекидне рутине, а затим ту вредност уписује у програмски бројач PC и прелази на фазу *читање инструкције* прве инструкције прекидне рутине. На стеку је Ситуација 4 (слика 4.е).

Инструкција 8: Почетне вредности регистра и индикатора су: PC=1000h, ACC=1100h, I=0, T=0.

Ради се о инструкцији INC са адресе 1000h која увећава садржај акумулатора ACC за један, па садржај акумулатора ACC постаје 0000 1101h.

Инструкција реагује на прекид, али не постоји прекид, па се прелази на извршавање следеће инструкције.

Инструкција 9: Почетне вредности регистра и индикатора су: PC=1001h, ACC=1101h, I=0, T=0.

Ради се о инструкцији PUSH са адресе 1001h, која садржај акумулатора ACC ставља на стек. На стеку је Ситуација 5 (слика 4.е).

Инструкција реагује на прекид, али не постоји прекид, па се прелази на извршавање следеће инструкције.

Инструкција 10: Почетне вредности регистра и индикатора су: PC=1002h, ACC=1101h, I=0, T=0.

Ради се о инструкцији POP са адресе 1002h, која са стека скида једну реч и уписује је у акумулатор ACC. На стеку је Ситуација 6 (слика 4.е).

Инструкција реагује на прекид, али не постоји прекид, па се прелази на извршавање следеће инструкције.

Инструкција 11: Почетне вредности регистара и индикатора су: PC=1003h, ACC=1101h, I=0, T=0.

Ради се о инструкцији RTI са адресе 1003h, која са стека скида контекст који је сачуван на стеку приликом преласка на ову прекидну рутину и уписује у програмски бројач PC и у програмску статусну реч PSW. Регистар PC добија вредност 104h, док у регистру PSW бит I добија вредност 1, а бит T вредност 0. На стеку је Ситуација 7 (слика 4.е).

Инструкција не реагује на прекид, па се прелази се на извршавање следеће инструкције.

Инструкција 12: Почетне вредности регистара и индикатора су: PC=104h, ACC=1101h, I=1, T=0.

Ради се о инструкцији STORE #1 са адресе 104h, која није исправна, јер се користи непосредно адресирање за одредишни операнд, па процесор генерише унутрашњи захтев за прекид због грешке у адресирању и флип-флоп PRADR поставља на 1 и прекида даље извршавање ове инструкције.

Инструкција реагује на прекид и постоји прекид PRADR. Стога се у оквиру фазе *опслуживање прекида* најпре на стеку чува контекст процесора програма чије се извршавање прекида, а затим се формира контекст процесора прекидне рутине на чије се извршавање прелази.

У корацима чувања контекста процесора програма чије се извршавање прекида на стеку се чува прво програмска статусна реч PSW, а потом и програмски бројач PC. Код програмске статусне речи PSW која се чува на стеку бит I има вредност 1 и бит T има вредност 0. Програмски бројач PC има вредност 106h, која показује на прву следећу инструкцију програма чије се извршавање прекида. На стеку је Ситуација 8 (слика 4.е).

У корацима формирања контекста процесора прекидне рутине на чије се извршавање прелази потребно је поставити нове вредности у програмску статусну реч PSW и програмски бројач PC. У бит I регистра PSW се уписује вредност 0, а у бит T регистра PSW се такође уписује вредност 0. Улази у IV табелу су фиксни, па процесор, пошто се ради о унутрашњем прекиду PRADR, најпре утврђује да број улаза износи 3 (слика 4.д), а затим флип-флоп PRADR брише на 0. Адреса улаза 3 у IV табели је 5 ($5 = 2 + 3 \cdot 1$), па процесор најпре са адресе 5 чита вредност 100Bh, која представља адресу прекидне рутине, а затим ту вредност уписује у програмски бројач PC и прелази на фазу *читање инструкције* прве инструкције прекидне рутине.

Инструкција 13: Почетне вредности регистара и индикатора су: PC=100Bh, ACC=1101h, I=0, T=0.

Ради се о инструкцији POP са адресе 100Bh, која са стека скида вредност програмског бројача PC сачувану на стеку приликом преласка на прекидну рутину и која износи 106h и уписује је у акумулатор ACC. На стеку је Ситуација 9 (слика 4.е).

Инструкција реагује на прекид, али не постоји прекид, па се прелази на извршавање следеће инструкције.

Инструкција 14: Почетне вредности регистара и индикатора су: PC=100Ch, ACC=106h, I=0, T=0.

Ради се о инструкцији PUSH са адресе 100Ch, која садржај акумулатора ACC, који вредност програмског бројача PC сачувану на стеку приликом преласка на прекидну рутину и која износи 106h, ставља на стек. На стеку је Ситуација 10 (слика 4.е).

Инструкција реагује на прекид, али не постоји прекид, па се прелази на извршавање следеће инструкције.

Инструкција 15: Почетне вредности регистара и индикатора су: PC=100Dh, ACC=106, I=0, T=0.

Ради се о инструкцији RTI са адресе 100Dh, која са стека скида контекст који је сачуван на стеку приликом преласка на ову прекидну рутину и уписује у програмски бројач PC и у програмску статусну реч PSW. Регистар PC добија вредност 106h, док у регистру PSW бит I добија вредност 1, а бит T вредност 0. На стеку је Ситуација 11 (слика 4.е).

Инструкција не реагује на прекид, па се прелази се на извршавање следеће инструкције.

Инструкција 16: Почетне вредности регистара и индикатора су: PC=106h, ACC=106h, I=1, T=0.

Ради се о инструкцији DEC са адресе 106h која умањује садржај акумулатора ACC за један, па садржај акумулатора ACC постаје 0000 0105h.

Инструкција реагује на прекид, али не постоји прекид, па се прелази на извршавање следеће инструкције.

Инструкција 17: Почетне вредности регистара и индикатора су: PC=107h, ACC=105h, I=1, T=0.

Ради се о инструкцији INTD са адресе 107h, која у бит I програмске статусне речи PSW уписује вредност 0.

Инструкција не реагује на прекид, па се прелази на извршавање следеће инструкције.

Сумарно:

У фази *извршавање операције* 1. инструкције INTE главног програма са адресе 0100h у бит I програмске статусне речи PSW се уписује вредност 1 чиме се дозвољава спољашњи маскирајући прекид PRIRQM. У току извршавања 2. инструкције LOAD #1 главног програма са адресе 0101h стиже спољашњи немаскирајући захтев за прекид PRIRQN који се прихвата и прелази на извршавање 3. инструкције LOAD 1 прекидне рутине за PRIRQN са адресе 1004 h.

У току извршавања 5. инструкције STORE 1 прекидне рутине за PRIRQN са адресе 1008h стиже спољашњи маскирајући прекид PRIRQM који се не прихвата јер бит I програмске статусне речи PSW има вредност 0. У фази *извршавање операције* 6. инструкције RTI прекидне рутине за PRIRQN са адресе 100Ah бит I програмске статусне речи PSW се рестаурира на вредност 1, а програмски бројач PC на вредност 0103h. Како инструкција RTI не реагује на прекид, маскирајући прекид PRIRQM се не прихвата чиме се реализује повратак у прекинути главни програм и то на инструкцију INC са адресе 0103h.

У главном програму 7. инструкција INC са адресе 0103h реагује на прекид па се у њеној фази *опслуживање прекида* маскирајући прекид PRIRQM прихвата и прелази на извршавање 8. инструкције INC прекидне рутине за PRIRQM са адресе 1000h.

У фази *извршавање операције* 11. инструкције RTI прекидне рутине за PRIRQM са адресе 1003h бит I програмске статусне речи PSW се рестаурира на вредност 1, а програмски бројач PC на вредност 0104h. Инструкција RTI не реагује на прекид, а и не постоји прекид, чиме се реализује повратак у прекинути главни програм и то на инструкцију STORE #1 са адресе 0104h.

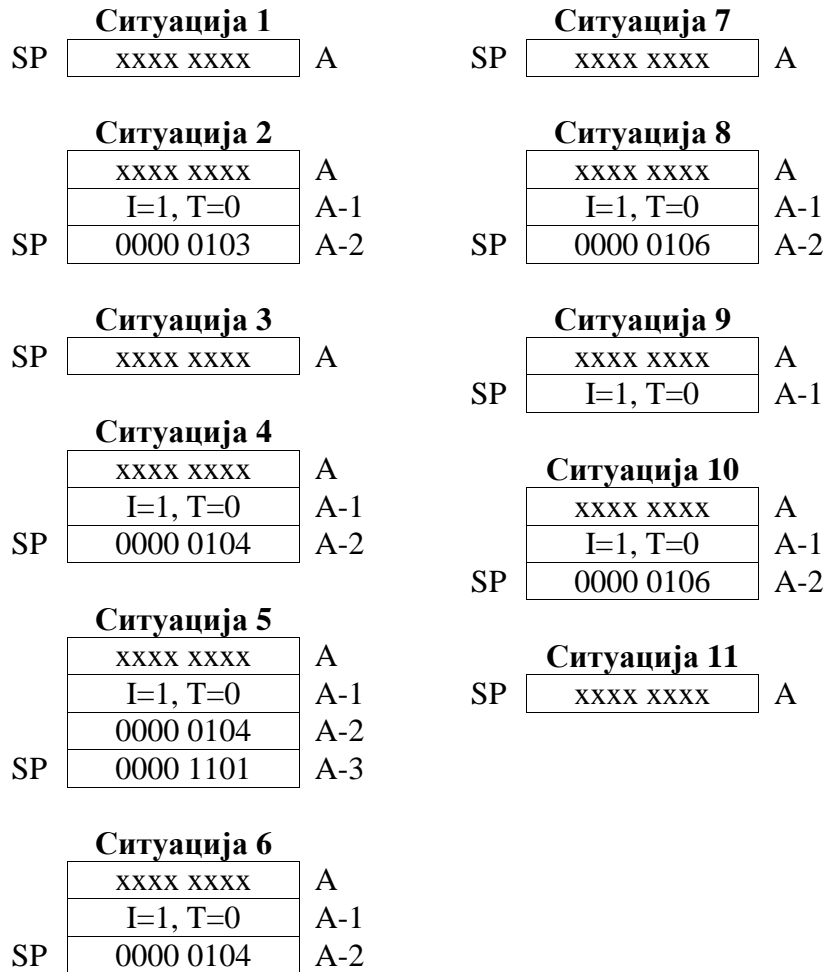
У току извршавања 12. инструкције STORE #1 главном програма са адресе 0104h генерише се унутрашњи прекид PRADR који се прихвата и прелази на извршавање 13. инструкције POP прекидне рутине за PRADR са адресе 100B h.

У фази *извршавање операције* 15. инструкције RTI прекидне рутине за PRADR са адресе 100Dh бит I програмске статусне речи PSW се рестаурира на вредност 1, а програмски бројач PC на вредност 0106h. Инструкција RTI не реагује на прекид, а и не постоји прекид, чиме се реализује повратак у прекинути главни програм и то на 16. инструкцију DEC са адресе 0106h.

У фази *извршавање операције* 17. инструкције INTD главног програма са адресе 0107h у бит I програмске статусне речи PSW се уписује вредност 0 чиме се маскира спољашњи маскирајући прекид PRIRQM.

Рб	Адреса	Инструкција	ACC	Стек	I	T	PRIRQN	PRIRQM	PRADR
0	-	-	?	1	0	0	0	0	0
1	0100	INTE	?	1	1	0	0	0	0
2	0101	LOAD #1	1	1	1	0	1	0	0
			1	2	0	0	0	0	0
3	1004	LOAD 1	100A	2	0	0	0	0	0
4	1006	OR #FFh	10FF	2	0	0	0	0	0
5	1008	STORE 1	10FF	2	0	0	0	1	0
6	100A	RTI	10FF	3	1	0	0	1	0
7	0103	INC	1100	3	1	0	0	1	0
			1100	4	0	0	0	0	0
8	1000	INC	1101	4	0	0	0	0	0
9	1001	PUSH	1101	5	0	0	0	0	0
10	1002	POP	1101	6	0	0	0	0	0
11	1003	RTI	1101	7	1	0	0	0	0
12	0104	STORE #1	1101	7	1	0	0	0	1
			1101	8	0	0	0	0	0
13	100B	POP	0106	9	0	0	0	0	0
14	100C	PUSH	0106	10	0	0	0	0	0
15	100D	RTI	0106	11	1	0	0	0	0
16	0106	DEC	0105	11	1	0	0	0	0
17	0107	INTD	0105	11	0	0	0	0	0

Слика 4.ђ Табела са секвенцом инструкција које се редом извршавају по датом сценарију



Слика 4.е Изглед стека у току извршавања секвенце инструкција

д) Након извршења инструкције STORE 1 на адреси 1008h у прекидној рутини за PRIRQN, вредност на адреси 1h се мења. На крају секвенце под г) MEM[1]=10FFh.

Дискусија1:

У поставци задатка је речено да треба претпоставити да се у случају унутрашњег прекида због грешке у адресирању PRADR поступа на исти начин као и у случају осталих прекида и да програмски бројач PC приликом преласка на одговарајућу прекидну рутину указује на прву следећу инструкцију. Због тога приликом извршавања 12. инструкције STORE #1 са адресе 0104h која генерише прекид због грешке у адресирању PRADR узето је да је вредност програмског бројача PC приликом преласка на одговарајућу прекидну рутину 0106h.

У неким реализацијама процесора узима се да вредност PC приликом преласка на одговарајућу прекидну рутину указују на адресу инструкције на којој се јавио прекид због грешке у адресирању PRADR. Да је усвојена таква реализација процесора вредност програмског бројача PC приликом преласка на одговарајућу прекидну рутину била би 0104h.

Дискусија2:

Прекидна рутина за прекид због грешке у адресирању PRADR је реализована као свака друга прекидна рутина. Међутим, у реалној ситуацији у овој прекидној рутини се одузима процесор процесу у оквиру кога је генерисан овај прекид, шаље порука на којој адреси је инструкција у оквиру које се јавила грешка у адресирању због које је генерисан овај прекид и дати процес терминира. Због тога је погодно да приликом преласка на прекидну рутину вредност програмског бројача PC која се чува на стеку указује на адресу инструкције у оквиру које је генерисан овај прекид, а не на прву следећу инструкцију. После тога се процесор додељује новом процесу тако што се из његовог процес контрол блока формира контекст процесора за процеса коме се додељује процесор и у оквиру тога, између осталог, у регистре SP, PSW и PC уписују нове вредности. Због тога нема као у задатку да се повратком из прекидне рутине за прекид због грешке у адресирању PRADR продужава са извршавањем програма у коме се јавио овај прекид, већ се креће са извршавањем новог програма.

1.5 ЗАДАТАК 5

Оперативна меморија неког рачунара је капацитета 16 GB, а ширина речи меморије је 32 бита. Процесор је једноадресни, акумулатор је 32 битни, а механизам прекида је векторисан. Адресе прекидних рутина се чувају у табели адреса прекидних рутина (IV табела - *Interrupt Vector Table*) која почиње на адреси 2h оперативне меморије и има 4 фиксна улаза.

У рачунару могу да се генеришу два спољашња маскирајућа захтева за прекид који долази у процесор преко улазних линија $IRQM_1$ и $IRQM_2$ као импулс, један спољашњи немаскирајући захтев за прекид који долази у процесор преко улазне линије $IRQN$ као импулс и унутрашњи прекид због задатог режима рада прекид после сваке инструкције. Спољашњи маскирајући захтеви за прекид долазе од контролера периферија, спољашњи немаскирајући захтев за прекид долази од уређаја који контролише исправност рада делова рачунара и унутрашњи захтев за прекид због задатог режима рада прекид после сваке инструкције постоји уколико бит Т (Trap enable) регистра програмске статусне речи PSW има вредност 1. Адресе прекидних рутина за спољашње маскирајуће захтеве за прекид са линија $IRQM_1$ и $IRQM_2$, спољашњи немаскирајући захтев за прекид и унутрашњи захтев за прекид због задатог режима рада прекид после сваке инструкције се налазе у улазима 0, 1, 2 и 3 IV табеле, респективно. Највиши приоритет има спољашњи немаскирајући захтев за прекид, затим следе спољашњи маскирајући прекид са линије $IRQM_2$, па спољашњи маскирајући захтев за прекид са линије $IRQM_1$ и на крају са најнижим приоритетом долази унутрашњи захтев за прекид због задатог режима рада прекид после сваке инструкције.

Спољашњи маскирајући захтеви за прекид са линија $IRQM_2$ и $IRQM_1$ морају да имају више нивое приоритета од нивоа приоритета главног програма, да би се појавом било ког од ова два захтева за прекид извршавање главног програма прекидало и прелазило на извршавање одговарајуће прекидне рутине. Поред тога спољашњи маскирајући захтев за прекид са линије $IRQM_2$ мора да има виши ниво приоритета од спољашњег маскирајућег захтева за прекид са линије $IRQM_1$.

Битови I (Interrupt enable) и Т (Trap enable) регистра PSW се постављају на вредност 0 хардверски током извршавања фазе *опслуживање прекида*. У регистру PSW постоји одређен број бита L (Interrupt level) који одређују ниво приоритета програма који се тренутно извршава. Битови L се користе само за главни програм и за две прекидне рутине за спољашње маскирајуће захтеве за прекид са линија $IRQM_1$ и $IRQM_2$. Битови L регистра PSW се постављају на вредност нивоа приоритета прекидне рутине за спољашњи маскирајући захтев за прекид са линије $IRQM_1$ или $IRQM_2$ на коју се скаче хардверски током извршавања фазе *опслуживање прекида*. Инструкције INTE, INTD, RTI, TRPE и TRPD не реагују на захтеве за прекид. У оквиру извршавања фазе *опслуживање прекида* на стеку се хардверски чувају PSW и PC тим редом. Стек расте према нижим локацијама, а указивач на врх стека SP указује на задњу заузету локацију. Не постоји регистар маске IMR за селективно маскирање спољашњих маскирајућих захтева за прекид са линија $IRQM_1$ и $IRQM_2$.

Изглед дела главног програма је дат на слици 5.а, прекидних рутина на слици 5.б и дела меморије почев од адресе 0 на слици 5.в. Инструкција TRPE на адреси 0100h означена је као 1. (прва) по редоследу извршавања, а свака следећа инструкција која се извршава означена је следећим редним бројем. У току извршавања 2. инструкције стиже захтев за

прекид по линији $IRQM_1$, у току 5. инструкције по линији $IRQN$, а у току 9. инструкције по линији $IRQM_2$. На почетку извршавања главног програма су сви бити регистра програмске статусне речи PSW-а постављени на 0.

адреса	инструкција
0100h	TRPE
0101h	INTE
0102h	LOAD #1h
0104h	INTD
0105h	STORE 1h
0107h	TRPD

Слика 5.а Део главног програма

адреса	инструкција
1000h	INC
1001h	RTI
1002h	DEC
1003h	RTI
1004h	INC
1005h	RTI
1006h	INC
1007h	INC
1008h	RTI

Слика 5.б Прекидне рутине

адреса	садржај
0000h	1002h
0001h	1008h
0002h	1006h
0003h	1004h
0004h	1002h
0005h	1000h

Слика 5.в Изглед дела меморије почев од адресе 0

а) Написати на којим адресама започињу прекидне рутине за захтеве за прекид који долазе по линијама $IRQM_1$, $IRQM_2$ и $IRQN$, респективно

б) Нацртати и објаснити како се у датом процесору генерише сигнал од чије вредности зависи да ли ће се на крају фазе *извршавање операције* неке инструкције продужити са фазом *опслуживање прекида* и у оквиру ње реализовати прелаз на одговарајућу прекидну рутину или прећи на фазу *читање инструкције* прве следеће инструкције текућег програма.

в) Нацртати и објаснити како процесор фиксно генерише бројеве улаза у IV табелу.

г) Написати секвенцу адреса инструкција које се редом извршавају почев од инструкције са адресе 0100h и закључно са инструкцијом са адресе 0107h.

д) Навести која ће се вредност налазити на локацији 1h након извршења секвенце инструкција из тачке г).

Решење:

а) Капацитет оперативне меморије је 16 GB. Како је ширина речи меморије 32 бита, то оперативна меморија има 2^{32} адресибилних локација које садрже речи меморије ширине 32 бита. Стога су адресе које се користе за адресирање меморијских локација широке 32 бита.

Табела адреса прекидних рутина састоји се од 4 улаза који садрже почетне адресе прекидних рутина. Како су адресе широке 32 бита, а ширина речи меморије је 32 бита, сваки улаз табеле заузима једну локацију у меморији. Почетна адреса IV табеле, која се чува у регистру IVTP (*Interrupt Vector Table Pointer*), је 2h.

У датом случају величина улаза у IV табели исказана у адресибилним јединицама је 1, јер је адреса широка 32 бита, а ширина речи меморије која је адресибилна јединица је 32 бита. Ако се број улаза i који одговара некој периферији PER_j помножи са 1 и та вредност дода на садржај регистра IVTP који износи $2h$ добија се адреса a на којој је смештена адреса прекидне рутине периферије PER_j .

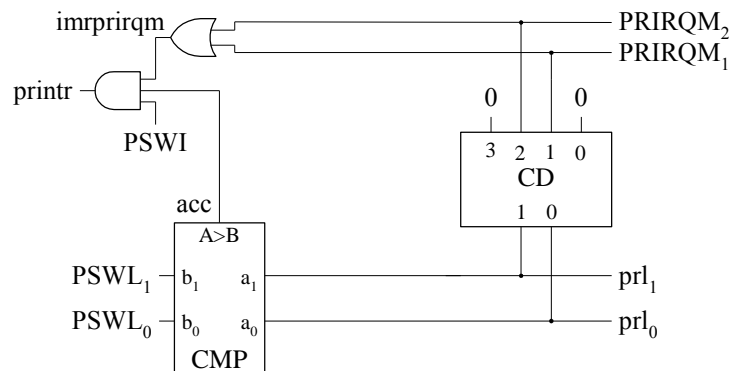
Адресе прекидних рутина за спољашње маскирајуће захтеве за прекид који долазе у процесор преко улазних линија $IRQM_1$ и $IRQM_2$ и спољашњи немаскирајући захтев за прекид који долази у процесор преко улазне линије $IRQN$ се налазе у улазима 0, 1 и 2 IV табеле, респективно. Када се бројеви улаза 0, 1 и 2 помноже са 1 и добијени помераји 0, 1 и 2 додају на садржај регистра IVTP који износи $2h$ добија се да се адресе на којима се налазе адресе прекидних рутина за захтеве за прекид који долазе по линијама $IRQM_1$, $IRQM_2$ и $IRQN$ налазе у меморијским локацијама чије су адресе $2h$, $3h$ и $4h$, респективно. Са слике 5.в. се добија да садржаји меморијских локација са адреса $2h$, $3h$ и $4h$ чије су вредности $0000\ 1006h$, $0000\ 1004h$ и $0000\ 1002h$ представљају адресе прекидних рутина за захтеве за прекид који долазе по линијама $IRQM_1$, $IRQM_2$ и $IRQN$, респективно.

б) Спољашњи маскирајући захтеви за прекид који се повезују на улазне линије процесора $IRQM_1$ и $IRQM_2$ долази као импулс, па се у процесору памти у флип-флоповима $PRIRQM_1$ и $PRIRQM_2$, респективно. Спољашњи немаскирајући захтев за прекид се генерише уписивањем вредности 1 у флип-флоп $PRIRQN$ када се на улазној линији процесора $IRQN$ појави импулс којим спољашњи уређај који контролише исправност рада делова рачунара сигнализира процесору да је откривена нека нерегуларност. Унутрашњи захтев за прекид због задатог режима рада прекид после сваке инструкције се генерише софтверски у фази *извршавање операције* инструкције $TRPE$ тако што се у бит Т регистра PSW ($PSWT$) упише вредност 1.

Сигнали $PRIRQM_1$ и $PRIRQM_2$ се даље користе за формирање сигнала спољашњег маскирајућег захтева за прекид $imgrirqm$, сигнала нивоа приоритета спољашњег маскирајућег захтева за прекид највишег нивоа приоритета $prl_{1..0}$ и сигнала прихватања спољашњег маскирајућег захтева за прекид acc (слика 5.г).

Сигнал спољашњег маскирајућег захтева за прекид $imgrirqm$ се формира на излазу логичког PII елемента

Сигнали нивоа приоритета спољашњег маскирајућег захтева за прекид највишег нивоа приоритета $prl_{1..0}$ се формирају на излазу кодера приоритета CD .



Слика 5.г Прихватање спољашњих маскирајућих захтева за прекид

Да би се остварио механизам којим се

прекида извршавање главног програма уколико се јави било који од спољашњих маскирајућих захтева за прекид PRIRQM₁ или PRIRQM₂,

прекида извршавање прекидне рутине маскирајућег захтева за прекид PRIRQM₁ уколико се јави спољашњих маскирајући захтев за прекид PRIRQM₂, главном програму и прекидним рутинама захтева за прекид PRIRQM₁ и PRIRQM₂, додељени су нивои приоритета 00b, 01b и 010b, респективно. У складу са тако додељеним нивоима приоритета и уз претпоставку да нивои приоритета кодера приоритета опадају од улаза 3 преко улаза 2 и 1 до улаза 0, сигнали PRIRQM₁ и PRIRQM₂ се везују на улазе 1 и 2 кодера приоритета CD.

Ниво приоритета текућег програма се чува у разредима PSWL_{1..0} регистра PSW. Током извршавања инструкције и то у оквиру фазе *опслуживање прекида* хардверски се најпре на стек стави садржај регистра PSW са вредностима разреда PSWL_{1..0} које представљају ниво приоритета текућег програма чије се извршавање прекида, а затим се у разреде PSWL_{1..0} уписују вредности сигнала prl_{1..0} који представљају сигнале нивоа приоритета спољашњег маскирајућег захтева за прекид највишег нивоа приоритета на чију се прекидну рутину прелази. На крају сваке прекидне рутине се програмским путем у фази *извршавање операције* инструкције RTI скида са стека и уписује у регистре PSW и PC, чиме се у разреде PSWL_{1..0} враћа вредност нивоа приоритета прекинутог програма чије се извршавање наставља.

Сигнал прихватања спољашњег маскирајућег захтева за прекид асс се формира на излазу A>B компаратора CMP уколико је вредност сигнала prl_{1..0} већа од вредности сигнала PSWL_{1..0}.

Бит PSWI вредношћу 1 дозвољава прихватање спољашњих маскирајућих захтева за прекид.

Сигнал прихваћеног спољашњег маскирајућег захтева за прекид printr добија вредност 1 уколико вредност 1 имају сигнали imrgrirqm, асс и PSWI. То значи да је за прекидање извршавања текућег програма потребно

да постоји барем један спољашњи маскирајући захтев за прекид PRIRQM₁ или PRIRQM₂, па сигнал imrprirqm има вредност 1,

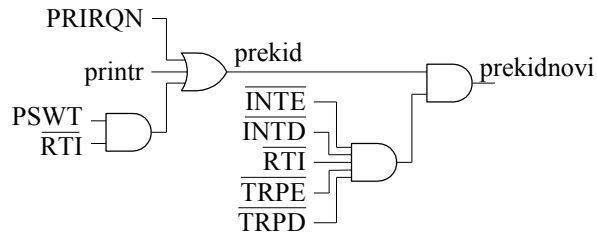
да је његов ниво приоритета prl_{1..0} већи од нивоа приоритета текућег програма PSWL_{1..0}, па сигнал асс има вредност 1, и

да нису маскирани спољашњи маскирајући захтеви за прекид, јер сигнал PSWI има вредност 1.

Инструкције INTE, INTD, RTI, TRPE и TRPD не реагују на захтеве за прекид.

Сигнал prekidnovi се генерише на начин приказан на слици 5.д. На почетку извршавања фазе *опслуживање прекида* сваке инструкције врши се провера вредности овог сигнала и у зависности од тога да ли има вредност 1 или 0 или продужава са преосталима корацима фазе *опслуживање прекида* којима се реализује прелаз на одговарајућу прекидну рутину или прелази на фазу *читање инструкције* прве следеће инструкције текућег програма.

Генерисање сигнала prekidnovi и prekid је објашњено у задатку 3.ж, при чему се овде сигнал prekid генерише као ИЛИ функција само сигнала PRIRQN, printr и PSWI · RTI.

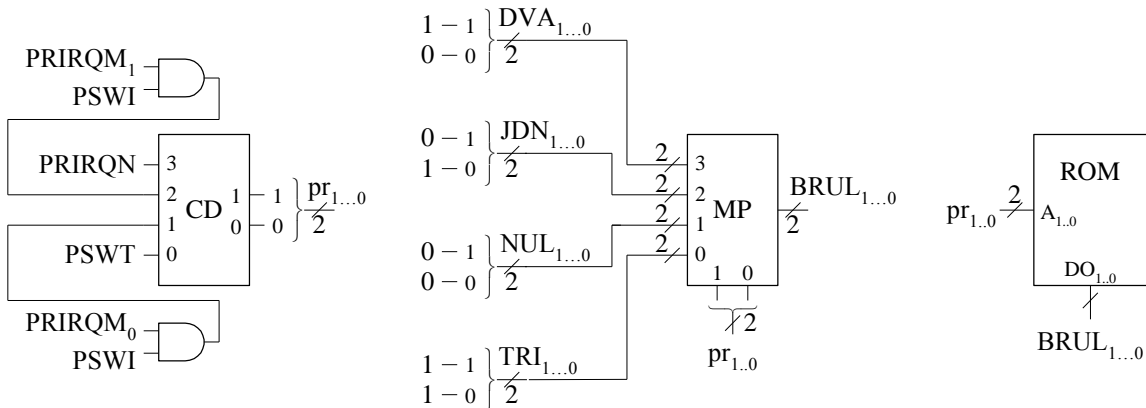


Слика 5.д Генерисање сигнал *prekidnovi* када све инструкције процесора реагују на прекид осим инструкција *INTE*, *INTD*, *RTI*, *TRPE* и *TRPD*

в) На слици 5.ђ је дата шема за генерисање броја улаза у IV табелу.

Адресе прекидних рутина за спољашње маскирајуће захтеве за прекид $PRIRQM_1$ и $PRIRQM_2$, спољашњи немаскирајући захтев за прекид $PRIRQN$ и унутрашњи захтев за прекид због задатог режима рада прекид после сваке инструкције $PSWT$ се налазе у улазима 0, 1, 2 и 3 IV табеле, респективно. Највиши приоритет има прекид $PRIRQN$, затим следе прекиди $PRIRQM_2$ и $PRIRQM_1$ и на крају са најнижим приоритетом долази прекид $PSWT$.

Најпре је потребно помоћу кодера приоритета CD формирати сигнале приоритета $pr_{1..0}$ прекида највишег нивоа приоритета, при чему у формирању ових сигнала сигнали спољашњих маскирајућих захтева за прекид $PRIRQM_2$ и $PRIRQM_1$ учествују једино уколико нису маскирани битом I програмске статусне речи PSW и сигнал $PSWI$ има вредност 1. Сигнале приоритета $pr_{1..0}$ треба затим користити за формирање броја улаза $BRUL_{1..0}$ (слика 5.ђ). Један начин за формирање броја улаза $BRUL_{1..0}$ је да се једна од четири вредности броја улаза $DVA_{1..0}$, $JDN_{1..0}$, $NUL_{1..0}$ или $TRI_{1..0}$ селектује сигнаlima $pr_{1..0}$ на излазе мултиплексера MP. Други начин за формирање броја улаза $BRUL_{1..0}$ је да се једна од четири вредности броја улаза чита и појави на излазним линијама података меморије ROM са адресе одређене вредностима сигнала $pr_{1..0}$, при чему су претходно у улазе 0, 1, 2 и 3 уписане вредности броја улаза 11b, 00b, 01b и 10b, респективно.



Слика 5.ђ Генерисање броја улаза у IV табелу

Одређивање броја улаза би било једноставније да су прекидима $PRIRQN$, $PRIRQM_2$, $PRIRQM_1$ и $PSWT$ у складу са опадајућим приоритетима додељени улази 3, 2, 1 и 0, респективно. У том случају би сигнали приоритета $pr_{1..0}$ прекида највишег нивоа приоритета са излаза кодера приоритета CD представљали број улаза $BRUL_{1..0}$ (слика 5.ђ).

г) Секвенца адреса инструкција које се редом извршавају почев од инструкције TRPE са адресе 0100h главног програма и закључно са инструкцијом TRPD са адресе 0107h главног програма, приказана је на слици 5.е, карактеристичне ситуације на стеку су дате на слици 5.ж, а објашњења у даљем тексту.

У табели са слике 5.е су у посебним колонама дати: Рб – редни број извршене инструкције, Адреса – адреса инструкције у хексадецималном облику (четири водеће нуле су изостављене), Инструкција – симболичка ознака инструкције и начина адресирања, ACC – садржај акумулатора ACC, Стек – ситуација на стеку са слике 5.ж, I – вредност бита I регистра програмске статусне речи PSW, T – вредност бита T регистра програмске статусне речи PSW, L – вредност бита L_{1..0} регистра програмске статусне речи PSW, PRIRQN – спољашњи немаскирајући захтев за прекид, PRIRQM₂ – спољашњи маскирајући захтев за прекид и PRIRQM₁ – спољашњи маскирајући захтев за прекид. У табели је за сваку инструкцију дата једна врста са стањем на крају фазе *извршавање операције*, а за инструкције које извршавају и фазу *опслуживање прекида* дата је још једна врста са стањем на крају фазе *опслуживање прекида*.

Почетна ситуација на стеку је дата као Ситуација 1. Регистар SP указује на задњу заузету меморијску локацију на адреси А која представља врх стека и чији је садржај небитан, па је означен са xxxx xxxx. Сви садржаји на стеку су дати у хексадецималном облику, а од садржаја програмске статусне речи PSW су дате само појединачне вредности битова I, T и L_{1..0}, јер су оне једино релевантне.

Инструкција 1: Почетне вредности регистара и индикатора су: PC=0100h, ACC=?, I=0, T=0, L=00.

Ради се о инструкцији TRPE са адресе 0100h, која у бит T програмске статусне речи PSW уписује вредност 1. Тиме процесор генерише унутрашњи захтев за прекид због задатог режима прекид после сваке инструкције PSWT.

Инструкција не реагује на прекид, мада прекид PSWT постоји, и прелази се на извршавање следеће инструкције.

Инструкција 2: Почетне вредности регистара и индикатора су: PC=0101h, ACC=?, I=0, T=1, L=00.

Ради се о инструкцији INTE са адресе 0101h, која у бит I програмске статусне речи PSW уписује вредност 1 чиме се дозвољавају спољашњи маскирајући прекиди. Током извршавања инструкције јавља се спољашњи маскирајући прекид на линији IRQM₁, па се флип-флоп PRIRQM₁ поставља на 1.

Инструкција не реагује на прекид, мада прекиди PRIRQM₁ и PSWT постоје, и прелази се на извршавање следеће инструкције.

Инструкција 3: Почетне вредности регистара и индикатора су: PC=0102h, ACC=?, I=1, T=1, L=00.

Ради се о инструкцији LOAD #1 са адресе 0102h, која у акумулатор ACC уписује вредност 1.

Инструкција реагује на прекид и прекиди PRIRQM₁ и PSWT постоје. На прекиде се реагује по опадајућим приоритетима. Спољашњег немаскирајућег прекида PRIRQN и спољашњег маскирајућег прекида PRIRQM₂, који представљају два прекида највишег нивоа приоритета нема. Спољашњи маскирајући прекид PRIRQM₁, који представља први следећи прекид по опадајућим приоритетима, постоји. Како је његов ниво приоритета prl_{1..0}, који износи 01b, виши од нивоа приоритета главног програма PSWL_{1..0}, који износи

00b, и како маскирајући прекиди нису маскирани, јер бит PSWI има вредност 1, сигнал прихваћеног спољашњег маскирајућег прекида printg има вредност 1 (слика 5.г). Стога се у оквиру фазе *опслуживање прекида* најпре на стеку чува контекст процесора програма чије се извршавање прекида, а затим се формира контекст процесора прекидне рутине на чије се извршавање прелази.

У корацима чувања контекста процесора програма чије се извршавање прекида на стеку се чува прво програмска статусна реч PSW, а потом и програмски бројач PC. Код програмске статусне речи PSW која се чува на стеку бит I има вредност 1, бит T има вредност 1 и бити L_{1..0} имају вредност 00b. Програмски бројач PC има вредност 0104h, која показује на прву следећу инструкцију програма који се прекида. Програмски бројач PC је дужине 32 бита и приликом његовог стављања на стек његов садржај се смешта у једну меморијску локацију јер је ширина једне меморијске локације 32 бита. У једну меморијску локацију се смешта и садржај програмске статусне речи PSW, јер се из поставке задатка може утврдити да поред уобичајених битова N, Z, C и V (задатак 1 - Дискусија) постоје још само битови I, T и L_{1..0} и да је дужина PSW мања од 32 бита.

У корацима формирања контекста процесора прекидне рутине на чије се извршавање прелази потребно је поставити нове вредности у програмску статусну реч PSW и програмски бројач PC. У бит I регистра PSW се уписује вредност 0, у бит T регистра PSW се такође уписује вредност 0, док се у битове L_{1..0} регистра PSW уписује вредност 01b. Треба уочити да се у битове L_{1..0} регистра PSW уписује нова вредност само приликом преласка на неку од прекидних рутина са спољашње маскирајуће прекиде PRIRQM₁ или PRIRQM₂. Улази у IV табелу су фиксни, па процесор, пошто се ради о маскирајућем прекиду PRIRQM₁, најпре утврђује да број улаза износи 0 (слика 5.в), а затим флип-флоп PRIRQM₁ брише на 0. Адреса улаза 0 у IV табели је 2 ($2 = 2 + 0 \cdot 1$), па процесор најпре са адресе 2 чита вредност 0000 1006h, која представља адресу прекидне рутине, а затим ту вредност уписује у програмски бројач PC и прелази на фазу *читање инструкције* прве инструкције прекидне рутине. На стеку је Ситуација 2 (слика 4.е).

Инструкција 4: Почетне вредности регистра и индикатора су: PC=1006h, ACC=1h, I=0, T=0, L=01.

Ради се о инструкцији INC са адресе 1006h, која увећава садржај акумулатора ACC за један, па садржај акумулатора ACC постаје 0000 0002h.

Инструкција реагује на прекид, али нема прекида. Треба уочити да је претходна инструкција у својој фази *опслуживање прекида* у PSWT уписала вредност 0. Стога се прелази на извршавање следеће инструкције.

Инструкција 5: Почетне вредности регистра и индикатора су: PC=1007h, ACC=2h, I=0, T=0, L=01.

Ради се о инструкцији INC са адресе 1007h, која увећава садржај акумулатора за један, па садржај акумулатора ACC постаје 0000 0003h. Током извршавања инструкције јавља се спољашњи немаскирајући прекид на линији IRQN, па се флип-флоп PRIRQN поставља на 1

Инструкција реагује на прекид и прекид PRIRQN постоји. На прекиде се реагује по опадајућим приоритетима. Спољашњи немаскирајући прекид PRIRQN представља прекид највишег нивоа приоритета, па процесор реагује на овај прекид. Стога се у оквиру фазе *опслуживање прекида* најпре на стеку чува контекст процесора програма чије се извршавање прекида, а затим се формира контекст процесора прекидне рутине на чије се извршавање прелази.

У корацима чувања контекста процесора програма чије се извршавање прекида на стеку се чува прво програмска статусна реч PSW, а потом и програмски бројач PC. Код програмске статусне речи PSW која се чува на стеку бит I има вредност 0, бит T има вредност 0 и бити L_{1..0} имају вредност 01b. Програмски бројач PC има вредност 1008h, која показује на прву следећу инструкцију програма. На стеку је Ситуација 3 (слика 5.ж).

У корацима формирања контекста процесора прекидне рутине на чије се извршавање прелази потребно је поставити нове вредности у програмску статусну реч PSW и програмски бројач PC. У бит I регистра PSW се уписује вредност 0, у бит T регистра PSW се такође уписује вредност 0, док се битови L_{1..0} регистра PSW не мењају, јер се они мењају само када се прелази на неку од прекидних рутину за маскирајуће прекиде PRIRQM₁ или PRIRQM₂, и остају 01b. Улази у IV табелу су фиксни, па процесор, пошто се ради о немаскирајућем прекиду PRIRQN, најпре утврђује да број улаза износи 2 (слика 5.ђ), а затим флип-флоп PRIRQN брише на 0. Адреса улаза 2 у IV табели је 4 (4 = 2 + 2·1), па процесор најпре са адресе 4 чита вредност 0000 1002h, која представља адресу прекидне рутине, а затим ту вредност уписује у програмски бројач PC и прелази на фазу *читање инструкције* прве инструкције прекидне рутине.

Инструкција 6: Почетне вредности регистара и индикатора су: PC=1002h, ACC=3h, I=0, T=0, L=01.

Ради се о инструкцији DEC са адресе 1002h, која умањује садржај акумулатора за један, па садржај акумулатора ACC постаје 0000 0002h.

Инструкција реагује на прекид, али нема прекида, па се прелази на извршавање следеће инструкције.

Инструкција 7: Почетне вредности регистара и индикатора су: PC=1003h, ACC=2h, I=0, T=0, L=01.

Ради се о инструкцији RTI са адресе 1003h, која са стека скида контекст који је сачуван на стеку приликом преласка на ову прекидну рутину и уписује у програмски бројач PC и у програмску статусну реч PSW. Регистар PC добија вредност 1008h, док у регистру PSW бит I добија вредност 0, бит T вредност 0 и битови L вредност 01. На стеку је Ситуација 4 (слика 5.ж).

Инструкција не реагује на прекид, па се прелази се на извршавање следеће инструкције.

Инструкција 8: Почетне вредности регистара и индикатора су: PC=1008h, ACC=2h, I=0, T=0, L=01.

Ради се о инструкцији RTI са адресе 1008h, која са стека скида контекст који је сачуван на стеку приликом преласка на ову прекидну рутину и уписује у програмски бројач PC и у програмску статусну реч PSW. Регистар PC добија вредност 0104h, док у регистру PSW бит I добија вредност 1, бит T вредност 1 и битови L вредност 00. Пошто је у бит T програмске статусне речи PSW уписана вредност 1, процесор поново генерише унутрашњи захтев за прекид због задатог режима прекид после сваке инструкције PSWT. На стеку је Ситуација 5 (слика 5.ж).

Инструкција не реагује на прекид, мада прекид PSWT постоји, и прелази се на извршавање следеће инструкције.

Инструкција 9: Почетне вредности регистара и индикатора су: PC=0104h, ACC=2, I=1, T=1, L=00.

Ради се о инструкцији INTD са адресе 0104h, која у бит I програмске статусне речи PSW уписује вредност 0, чиме су маскирани спољашњи маскирајући захтеви за прекид на

линијама $IRQM_2$ и $IRQM_1$. Током извршавања инструкције јавља се спољашњи маскирајући захтев за прекид на линији $IRQM_2$, па се флип-флоп $PRIRQM_2$ поставља на 1.

Инструкција не реагује на прекид, мада прекид PSWT постоји као и прекид $PRIRQM_2$ који је маскиран вредношћу 0 бита I програмске статусне речи PSW, па се прелази на извршавање следеће инструкције.

Инструкција 10: Почетне вредности регистара и индикатора су: PC=0105h, ACC=2, I=0, T=1, L=00.

Ради се о инструкцији STORE 1 са адресе 0105h, која у меморијску локацију на адреси 1 уписује садржај акумулатора чија је вредност 0000 0002h.

Инструкција реагује на прекид и прекид PSWT постоји. На прекиде се реагује по опадајућим приоритетима. Спољашњег немаскирајућег прекида $PRIRQN$ нема, спољашњи маскирајући прекид $PRIRQM_2$ постоји али је маскиран вредношћу 0 бита PSWI, спољашњег маскирајућег прекида $PRIRQM_1$ нема, па прекид постоји због прекида PSWT. Стога се у оквиру фазе *ослуживање прекида* најпре на стеку чува контекст процесора програма чије се извршавање прекида, а затим се формира контекст процесора прекидне рутине на чије се извршавање прелази.

У корацима чувања контекста процесора програма чије се извршавање прекида на стеку се чува прво програмска статусна реч PSW, а потом и програмски бројач PC. Код програмске статусне речи PSW која се чува на стеку бит I има вредност 0, бит T има вредност 1 и битови L вредност 00. Програмски бројач PC има вредност 0107h, која показује на прву следећу инструкцију програма чије се извршавање прекида. На стеку је Ситуација 6 (слика 5.ж).

У корацима формирања контекста процесора прекидне рутине на чије се извршавање прелази потребно је поставити нове вредности у програмску статусну реч PSW и програмски бројач PC. У бит I регистра PSW се уписује вредност 0, у бит T регистра PSW се такође уписује вредност 0, док се вредност битова $L_{1..0}$ регистра PSW не мењају, јер се они мењају само када се прелази на неку од прекидних рутина за маскирајуће прекиде $PRIRQM_1$ или $PRIRQM_2$, и остају 00b. Улази у IV табелу су фиксни, па процесор, пошто се ради о прекиду PSWT, најпре утврђује да број улаза износи 3 (слика 5.ђ). Адреса улаза 3 у IV табели је 5 ($5 = 2 + 3 \cdot 1$), па процесор најпре са адресе 5 чита вредност 0000 1000h, која представља адресу прекидне рутине, а затим ту вредност уписује у програмски бројач PC и прелази на фазу *читање инструкције* прве инструкције прекидне рутине.

Инструкција 11: Почетне вредности регистара и индикатора су: PC=1000h, ACC=2h, I=0, T=0, L=00.

Ради се о инструкцији INC са адресе 1000h, која увећава садржај акумулатора за један, па садржај акумулатора ACC постаје 0000 0003h.

Инструкција реагује на прекид, прекид $PRIRQM_2$ постоји али је маскиран вредношћу 0 бита PSWI, па се прелази на извршавање следеће инструкције.

Инструкција 12: Почетне вредности регистара и индикатора су: PC=1001h, ACC=3h, I=0, T=0, L=00.

Ради се о инструкцији RTI са адресе 1001h, која са стека скида контекст који је сачуван на стеку приликом преласка на ову прекидну рутину и уписује у програмски бројач PC и у програмску статусну реч PSW. Регистар PC добија вредност 0107h, док у регистру PSW бит I добија вредност 0, бит T вредност 1 и битови L вредност 00. На стеку је Ситуација 7 (слика 5.ж).

Инструкција не реагује на прекид, мада прекид PRIRQM₂ постоји али је маскиран вредношћу 0 бита PSWI, па се прелази на извршавање следеће инструкције.

Инструкција 13: Почетне вредности регистра и индикатора су: PC=0107h, ACC=2, I=0, T=1, L=00.

Ради се о инструкцији TRPD са адресе 0107h, која у бит Т програмске статусне речи PSW уписује вредност 0, чиме се укида захтев за прекид PSW Т.

Инструкција не реагује на прекид, мада прекид PRIRQM₂ постоји али је маскиран вредношћу 0 бита PSWI, па се прелази на извршавање следеће инструкције.

Сумарно:

У фази *извршавање операције* 1. инструкције TRPE главног програма са адресе 0100h у бит Т програмске статусне речи PSW се уписује вредност 1 чиме се задаје режим рада прекид после сваке инструкције. Инструкција TRPE не реагује на прекиде, па се иако постоји прекид PSWT, прекид не прихвата и прелази на следећу инструкцију. У фази *извршавање операције* 2. инструкције INTE главног програма са адресе 0101h у бит I програмске статусне речи PSW се уписује вредност 1 чиме се дозвољавају спољашњи маскирајући прекиди PRIRQM₁ или PRIRQM₂. У току извршавања ове инструкције јавља се спољашњи маскирајући захтев за прекид PRIRQM₁. Инструкција INTE не реагује на прекиде, па се, иако постоје прекиди PSWT и PRIRQM₁, при чему прекид PRIRQM₁ није маскиран јер PSWI има вредност 1 и његов приоритет је виши од приоритета главног програма, прелази на следећу инструкцију. У фази *опслуживање прекида* 3. инструкције LOAD #1 главног програма са адресе 0102h прелази се на извршавање 4. инструкције INC прекидне рутине за PRIRQM₁ са адресе 1006 h.

У току извршавања 5. инструкције INC са адресе 0107h прекидне рутине за PRIRQM₁ јавља се спољашњи немаскирајући прекид PRIRQN који се прихвата и прелази се на извршавање 6. инструкције INC прекидне рутине за PRIRQN са адресе 1002h.

У фази *извршавање операције* 7. инструкције RTI прекидне рутине за PRIRQN са адресе 1003h програмски бројач PC се рестаурира на вредност 0108h.

У фази *извршавање операције* 8. инструкције RTI прекидне рутине за PRIRQM₁ са адресе 1008h битови I и T програмске статусне речи PSW се рестаурирају на вредност 1, а програмски бројач PC на вредност 0104h. Инструкција RTI не реагује на прекид, па се, иако постоји прекид PSWT, реализује повратак у прекинути главни програм и то на инструкцију INTD са адресе 0104h.

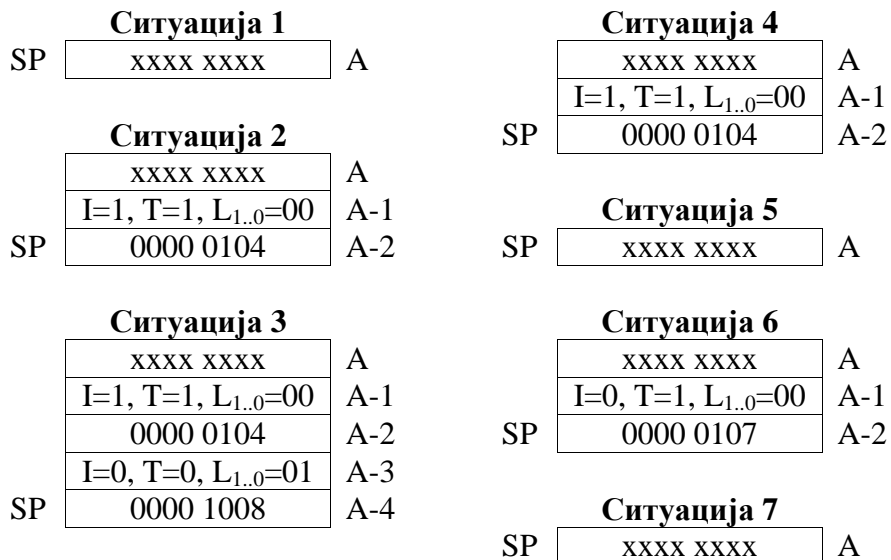
У фази *извршавање операције* 9. инструкције INTD главног програма са адресе 0104h у бит I програмске статусне речи PSW се уписује вредност 0 чиме се реализује маскирање маскирајућих прекида. У току извршавања ове инструкције јавља се спољашњи маскирајући захтев за прекид PRIRQM₂. Инструкција INTE не реагује на прекиде, па се, иако постоје прекиди PSWT и PRIRQM₂, при чему је прекид PRIRQM₂ маскиран јер PSWI има вредност 0, прелази на следећу инструкцију. У главном програму 10. инструкција STORE 1 са адресе 0105h реагује на прекид, па се у њеној фази *опслуживање прекида* прекид PSWT прихвата и прелази на извршавање 11. инструкције INC прекидне рутине за PSWT са адресе 1000h.

У фази *извршавање операције* 12. инструкције RTI прекидне рутине за PSWT са адресе 1001h бит I програмске статусне речи PSW се рестаурира на вредност 0, бит T програмске статусне речи PSW се рестаурира на вредност 1, а програмски бројач PC на вредност 0107h. Инструкција RTI не реагује на прекид, па се, иако постоји прекид PSWT, реализује повратак у прекинути главни програм и то на инструкцију TRPD са адресе 0107h.

У фази *извршавања операције* 13. инструкције TRPE главног програма са адресе 0107h у бит Т програмске статусне речи PSW се уписује вредност 0 чиме се укида режим рада прекид после сваке инструкције и прекид PSWT.

Рб	Адреса	Инструкција	ACC	Стек	I	T	L	PRIRQN	PRIRQM ₂	PRIRQM ₁
0	-	-	?	1	0	0	00	0	0	0
1	100	TRPE	?	1	0	1	00	0	0	0
2	101	INTE	?	1	1	1	00	0	0	1
3	102	LOAD #1	1	1	1	1	00	0	0	1
			1	2	0	0	01	0	0	0
4	1006	INC	2	2	0	0	01	0	0	0
5	1007	INC	3	2	0	0	01	1	0	0
			3	3	0	0	01	0	0	0
6	1002	DEC	2	3	0	0	01	0	0	0
7	1003	RTI	2	4	0	0	01	0	0	0
8	1008	RTI	2	5	1	1	00	0	0	0
9	104	INTD	2	5	0	1	00	0	1	0
10	105	STORE 1h	2	5	0	1	00	0	1	0
			2	6	0	0	00	0	1	0
11	1000	INC	3	6	0	0	00	0	1	0
12	1001	RTI	3	7	0	1	00	0	1	0
13	107	TRPD	3	7	0	0	00	0	1	0

Слика 5.е Табела са секвенцом инструкција које се извршавају



Слика 5.ж Изглед стека у току извршавања секвенце инструкција

д) Након извршења 10. инструкције STORE 1 на адреси 0105h у главном програму, вредност на адреси 1h је промењена, јер је у фази *извршавања операције* инструкције STORE 1 садржај акумулатора ACC, чија је вредност тада 0000 0002h, уписан у меморијску локацију на адреси 0000 0001 h. До краја секвенце под г) ниједна инструкцију не врши уписивање у дату меморијску локацију, па је зато на крају секвенце инструкција MEM[0000 00001]= 0000 0002h.

1.6 ЗАДАТАК 6

Оперативна меморија неког рачунара је капацитета 128KB, а ширина речи меморије је 16 бита. Процесор је једноадресни, акумулатор је дужине 16 бита и све инструкције се извршавају над 16 битним величинама. Улазно/излазни и меморијски адресни простори су раздвојени. Механизам прекида је векторисан. Адресе прекидних рутина се чувају у табели адреса прекидних рутина (IV табела - *Interrupt Vector Table*) која почиње на адреси 2h оперативне меморије и има 256 улаза.

Контролери периферија PER0, PER1 и PER2 шаљу процесору спољашње маскирајуће захтеве за прекид преко улазних линија IRQM₀, IRQM₁ и IRQM₂, а од процесора добијају сигнале потврда преко излазних линија INTA₀, INTA₁ и INTA₂. Улази 2, 3 и 4 у IV табели су додељени периферијама PER0, PER1 и PER2, респективно. Контролер периферије шаље процесору захтев за прекид *intr* као импулс и када од процесора прими сигнал потврде *inta* процесору шаљу свој број улаза у IV табелу. Адресе 16 битних регистара контролера периферија PER0, PER1 и PER2 у којима се чувају бројеви улаза у IV табелу су 10h, 20h и 30h, респективно.

Процесор може да генерише један унутрашњи захтев за прекид због извршавања инструкције прекида INS. Претпоставити да је извршавање инструкције INS тако реализовано да се у фази *извршавање операције* само у флип-флоп PRINS уписује вредност 1 и одмах прелази на фазу *опслуживање прекида*. У оквиру фазе *опслуживање прекида* се прелази на прекидну рутину на исти начин као и за све остале прекиде.

Највиши приоритет има унутрашњи захтев за прекид због извршавања инструкције прекида INS, док ниже приоритете имају спољашњи маскирајући захтеви за прекид. У оквиру спољашњи маскирајућих захтева за прекид највиши ниво приоритета има захтев за прекид који је везан на линију IRQM₀, затим захтев за прекид који је везан на линију IRQM₁, док најнижи ниво приоритета има захтев за прекид који је везан на линију IRQM₂.

У процесору постоји регистар PSW (*Program Status Word*) са битом I (*Interrupt Enable*) и одређен бројем битова L (*Priority Level*). Бит I вредношћу 1 дозвољава прихватање спољашњих маскирајућих захтева за прекид. Овај бит се поставља на вредност 0 хардверски током извршавања фазе *опслуживање прекида*. Битови L садрже ниво приоритета текућег програма и користе се да се само приликом извршавања главног програма и прекидних рутина спољашњих маскирајућих захтева за прекид реализује механизам којим се прихватају спољашњих маскирајући захтеви за прекид једино уколико су вишег нивоа приоритета од нивоа приоритета текућег програма. Ови битови се постављају на вредност нивоа приоритета периферије на чију прекидну рутину се прелази хардверски током извршавања фазе *опслуживање прекида*.

У оквиру извршавања фазе *опслуживање прекида* на стеку се хардверски чувају PSW и PC тим редом. Стек расте према нижим локацијама, а указивач на врх стека SP указује на задњу заузету локацију. Инструкције INTE, INTD и RTI не реагују на спољашње захтеве за прекид.

Изглед дела главног програма је дат на слици 6.а, прекидних рутина на слици 6.б и дела меморије почев од адресе 0 на слици 6.в. Инструкција LOAD #2h на адреси 0100h означена је као 1. (прва) по редоследу извршавања, а свака следећа инструкција која се извршава означена је следећим редним бројем. У току извршавања 2. инструкције стиже захтев за прекид по линији IRQM₂, а у току 5. инструкције по линији IRQM₀. На почетку

извршавања главног програма су сви бити регистра програмске статусне речи PSW-а постављени на 0.

адреса	инструкција
0100h	LOAD #2h
0102h	INTE
0103h	INC
0104h	ADD #2h
0106h	INS #3h
0108h	INTD

Слика 6.а Изглед дела главног програма

адреса	инструкција
1000h	PUSH
1001h	INTE
1002h	LOAD 1h
1004h	INC
1005h	STORE 1h
1007h	POP
1008h	RTI
1009h	INTE
100Ah	INC
100Bh	RTI
100Ch	INC
100Dh	RTI

Слика 6.б Прекидне рутине

адреса	садржај
0000h	0000h
0001h	0001h
0002h	1000h
0003h	1008h
0004h	1000h
0005h	100Ch
0006h	1009h

Слика 6.в Изглед дела меморије

а) Написати на којим адресама започињу прекидне рутине за захтеве за прекид који долазе по линијама $IRQM_0$, $IRQM_1$ и $IRQM_2$, респективно

б) Написати део програма којим се попуњава улаз 3 у IV табели.

в) Написати део програма којим се додељују бројеви улаза у IV табелу за периферије PER0, PER1 и PER2.

г) Нацртати и објаснити како се у датом процесору генерише сигнал од чије вредности зависи да ли ће се на крају фазе *извршавање операције* неке инструкције продужити са фазом *ослуживање прекида* и у оквиру ње реализовати прелаз на одговарајућу прекидну рутину или прећи на фазу *читање инструкције* прве следеће инструкције текућег програма.

д) Захтеви за прекид преко линија $IRQM_0$, $IRQM_1$ и $IRQM_2$ могу да стигну истовремено од контролера периферија PER0, PER1 и PER2, респективно. У неком тренутку само један од захтева за прекид може да се прихвати и у зависности од тога који се захтев за прекид прихвата по једној од линија $INTA_0$, $INTA_1$ или $INTA_2$ се шаље сигнал потврде. Нацртати и

објаснити како се утврђује који се од ова три захтева за прекид прихвата и по којој линији се шаље сигнал потврде.

ђ) У неким варијантама реализације процесора не постоје линије за слање сигнала потврде и контролери периферија не шаљу бројеве улаза у IV табелу, већ их сам процесор генеришу. Нацртати и објаснити како се тада генеришу бројеви улаза 2, 3 и 4 додељени у IV табели периферијама PER0, PER1 и PER2, респективно.

е) Написати секвенцу адреса инструкција које се редом извршавају почев од инструкције са адресе 0100h и закључно са инструкцијом са адресе 0108h.

ж) Навести која ће се вредност налазити на локацији 0001h након извршења секвенце инструкција из тачке е).

Решење:

а) Капацитет оперативне меморије је 128 KB. Како је ширина речи меморије 16 бита, то оперативна меморија има 2^{16} адресибилних локација које садрже речи меморије ширине 16 бита. Стога су адресе које се користе за адресирање меморијских локација широке 16 бита.

Табела адреса прекидних рутина састоји се од 256 улаза који садрже почетне адресе прекидних рутина. Како су адресе широке 16 бита, а ширина речи меморије је 16 бита, сваки улаз табеле заузима једну локацију у меморији. Почетна адреса IV табеле, која се чува у регистру IVTP (*Interrupt Vector Table Pointer*), је 0002h.

У датом случају величина улаза у IV табели исказана у адресибилним јединицама је 1, јер је адреса широка 16 бита, а ширина речи меморије која је адресибилна јединица је 16 бита. Ако се број улаза i који одговара некој периферији PER j помножи са 1 и та вредност дода на садржај регистра IVTP који износи 0002h добија се адреса a на којој је смештена адреса прекидне рутине периферије PER j .

Адресе прекидних рутина за спољашње маскирајуће захтеве за прекид који долази у процесор преко улазних линија IRQM₀, IRQM₁ и IRQM₂ се налазе у улазима 2, 3 и 4 IV табеле, респективно. Када се бројеви улаза 2, 3 и 4 помноже са 1 и добијени помераји 2, 3 и 4 додају на садржај регистра IVTP који износи 0002h добија се да се адресе на којима се налазе адресе прекидних рутина за захтеве за прекид који долазе по линијама IRQM₀, IRQM₁ и IRQM₂ налазе у меморијским локацијама чије су адресе 0004h, 0005h и 0006h, респективно. Са слике 6.в се добија да садржаји меморијских локација са адреса 0004h, 0005h и 0006h чије су вредност 1000h, 100Ch и 1009h представљају адресе прекидних рутина за захтеве за прекид који долазе по линијама IRQM₀, IRQM₁ и IRQM₂, респективно.

б) Потребно је у улаз 3 IV табеле, који се налази на адреси 0005h, уписати адресу прекидне рутине, која износи 100Ch. Све инструкције процесора се извршавају над 16 битним величинама, па се тражена иницијализација може извести из једног преноса 16 битне речи. IV табела се налази у оперативној меморији која је у меморијском адресном простору, па се упис садржаја акумулатора у ову меморијску локацију обавља користећи инструкцију STORE. Тражени део програма је приказан на слици 6.г. Овај део програма се извршава током иницијализације система.

```
LOAD    #100Ch ; напуни акумулатор непосредном величином 100Ch која представља адресу
STORE   0005h ; пребаци садржај акумулатора у меморијску локацију на адреси 0005h
```

Слика 6.г Иницијализација улаза 3 у IV табели

в) Бројеви улаза у IV табелу чувају се у за то намењеним регистрима контролера периферија PER0, PER1 и PER2 који се налазе на адресама 0010h, 0020h и 0030h,

респективно, у улазно/излазном адресном простору. Пошто су улазно/излазни и меморијски адресни простори раздвојени, упис садржаја акумулатора у ове регистре контролера периферија се обавља користећи инструкцију OUT. Да је улазно/излазни адресни простор меморијски пресликан, упис садржаја акумулатора у ове регистре контролера периферија би се обављао користећи инструкцију STORE. Тражени део програма је приказан на слици б.д. Овај део програма се извршава током иницијализације система.

```

LOAD    #2h    ; напуни акумулатор непосредном величином 2 која представља број улаза у
          IV табелу за PER0
OUT 0010h  ; пребаци садржај акумулатора у регистар контролера периферије PER0 на адреси
          0010h
LOAD    #3h    ; напуни акумулатор непосредном величином 3 која представља број улаза у
          IV табелу за PER1
OUT 0020h  ; пребаци садржај акумулатора у регистар контролера периферије PER1 на адреси
          0020h
LOAD    #4h    ; напуни акумулатор непосредном величином 4 која представља број улаза у
          IV табелу за PER2
OUT 0030h  ; пребаци садржај акумулатора у регистар контролера периферије PER2 на адреси
          0030h

```

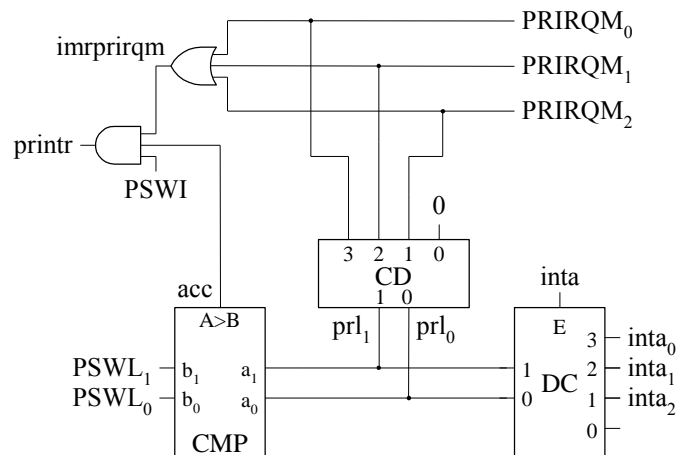
Слика б.д Иницијализација бројева улаза за PER0, PER1 и PER2

г) Спољашњи макирајући захтеви за прекид који се повезују на улазне линије процесора $IRQM_0$, $IRQM_1$ и $IRQM_2$ долази као импулс, па се у процесору памте у флип-флоповима $PRIRQM_0$, $PRIRQM_1$ и $PRIRQM_2$, респективно. Унутрашњи захтев за прекид због инструкције прекида се генерише софтверски у фази *извршавање операције* инструкције INS тако што се у флип-флоп PRINS упише вредност 1.

Сигнали $PRIRQM_0$, $PRIRQM_1$ и $PRIRQM_2$ се даље користе за формирање сигнала спољашњег маскирајућег захтева за прекид $imrprirqm$, сигнала нивоа приоритета спољашњег маскирајућег захтева за прекид највишег нивоа приоритета $prl_{1..0}$ и сигнала прихватања спољашњег маскирајућег захтева за прекид acc (слика б.ђ).

Сигнал спољашњег маскирајућег захтева за прекид $imrprirqm$ се формира на излазу логичког ILI елемента

Сигнали нивоа приоритета спољашњег маскирајућег захтева за прекид највишег нивоа приоритета $prl_{1..0}$ се формирају на излазу кодера приоритета CD.



Слика б.ђ Прихватање захтева за прекид и слање сигнала потврде

Да би се остварио механизам којим се прекида извршавање главног програма уколико се јави било који од спољашњих маскирајућих захтева за прекид и утврђује који се од захтева за прекид који долазе преко линија $IRQM_0$, $IRQM_1$ и $IRQM_2$ прихвата, главном програму и прекидним рутинама периферија PER_0 , PER_1 и PER_2 се додељују нивои приоритета. У овом случају постоји главни програм и три прекидне рутине периферија, па су четири бинарне вредности 00, 01, 10 и 11 довољне за кодирање њихових нивоа приоритета. Да би се омогућило прекидање извршавања главног програма када стигне захтев за прекид од било ког контролера периферија PER_0 , PER_1 и PER_2 , главном програму се додељује најнижи ниво приоритета 00 а прекидним рутинама периферија PER_0 , PER_1 и PER_2 преостали виши нивои приоритета 01, 10 и 11. Уколико се жели да у оквиру периферија PER_0 , PER_1 и PER_2 периферија PER_0 може да прекида извршавање прекидних рутина периферија PER_1 и PER_2 и да у оквиру периферија PER_1 и PER_2 периферија PER_1 може да прекида извршавање прекидне рутине периферије PER_2 , тада периферија PER_0 треба да има највиши ниво приоритета, периферија PER_1 нижи и периферија PER_2 најнижи. У том случају периферијама PER_0 , PER_1 и PER_2 треба доделити нивое приоритета 11, 10 и 01, респективно. У складу са тако додељеним нивоима приоритета и уз претпоставку да нивои приоритета кодера приоритета опадају од улаза 3 преко улаза 2 и 1 до улаза 0, сигнали $PRIRQM_0$, $PRIRQM_1$ и $PRIRQM_2$ се везују на улазе 3, 2 и 1 кодера приоритета CD, респективно.

Ниво приоритета текућег програма се чува у разредима $PSWL_{1..0}$ регистра PSW. Током извршавања инструкције и то у оквиру фазе *опслуживање прекида* хардверски се најпре на стек стави садржај регистра PSW са вредностима разреда $PSWL_{1..0}$ које представљају ниво приоритета текућег програма чије се извршавање прекида, а затим се у разреде $PSWL_{1..0}$ уписују вредности сигнала $prl_{1..0}$ који представљају сигнале нивоа приоритета спољашњег маскирајућег захтева за прекид највишег нивоа приоритета на чију се прекидну рутину прелази. На крају сваке прекидне рутине се програмским путем у фази *извршавање операције* инструкције RTI скида са стека и уписује у регистре PSW и PC, чиме се у разреде $PSWL_{1..0}$ враћа вредност нивоа приоритета прекинутог програма чије се извршавање наставља.

Сигнал прихватања спољашњег маскирајућег захтева за прекид асс се формира на излазу $A > B$ компаратора CMP уколико је вредност сигнала $prl_{1..0}$ већа од вредности сигнала $PSWL_{1..0}$.

Бит PSWI вредношћу 1 дозвољава прихватање спољашњих маскирајућих захтева за прекид.

Сигнал прихваћеног спољашњег маскирајућег захтева за прекид printr добија вредност 1 уколико вредност 1 имају сигнали $imrprirqm$, асс и PSWI. То значи да је за прекидање извршавања текућег програма потребно

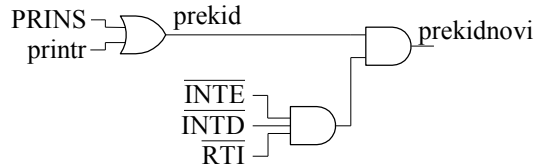
да постоји барем један спољашњи маскирајући захтев за прекид $PRIRQM_0$, $PRIRQM_1$ и $PRIRQM_2$, па сигнал $imrprirqm$ има вредност 1,

да је његов ниво приоритета $prl_{1..0}$ већи од нивоа приоритета текућег програма $PSWL_{1..0}$, па сигнал асс има вредност 1, и

да нису маскирани спољашњи маскирајући захтеви за прекид, јер сигнал PSWI има вредност 1.

Инструкције INTE, INTD и RTI не реагују на захтеве за прекид.

Сигнал *prekidnovi* се генерише на начин приказан на слици 6.е. На почетку извршавања фазе *ослуживање прекида* сваке инструкције врши се провера вредности овог сигнала и у зависности од тога да ли има вредност 1 или 0 или продужава са преосталима корацима фазе *ослуживање прекида* којима се реализује прелаз на одговарајућу прекидну рутину или прелази на фазу *читање инструкције* прве следеће инструкције текућег програма.



Слика 6.е Генерисање сигнал *prekidnovi* када све инструкције процесора реагују на прекид осим инструкција *INTE*, *INTD* и *RTI*

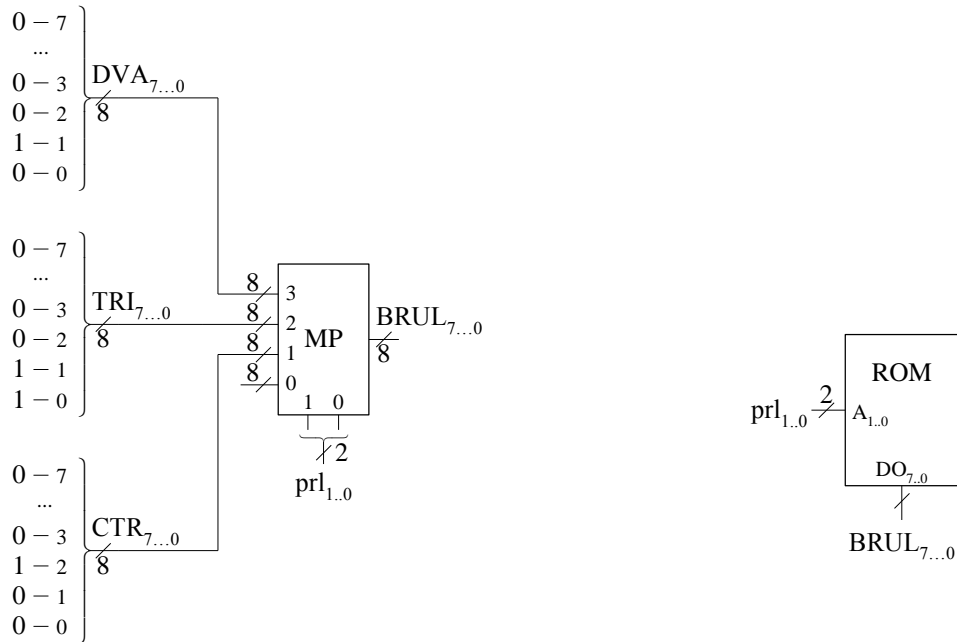
Генерисање сигнала *prekidnovi* и *prekid* је објашњено у задатку 3.ж, при чему се овде сигнал *prekid* генерише као ИЛИ функција само сигнала *PRINS* и *printr*.

д) У тачки г) је објашњено како се спољашњи маскирајући захтеви за прекид који долазе на улазне линије процесора *IRQM₀*, *IRQM₁* и *IRQM₂* као импулс, у процесору памте у флип-флоповима *PRIRQM₀*, *PRIRQM₁* и *PRIRQM₂*, респективно. Такође је објашњено и како се формирају сигнали нивоа приоритета спољашњег маскирајућег захтева за прекид највишег нивоа приоритета *prl_{1..0}* на излазу кодера приоритета *CD* (слика 6.ђ).

На почетку секвенца корака којом се у оквиру фазе *ослуживање прекида* израчунава адреса прекидне рутине процесор генерише вредност 1 сигнал *inta*, која на излазима декодера *DC* даје вредност 1 једног од сигнала *inta₀*, *inta₁* или *inta₃*, у зависности од тога да ли сигнали *prl_{1..0}* имају вредност 11, 10 или 01, респективно (слика 6.ђ). Сигнали *inta₀*, *inta₁* или *inta₀* се по линијама *INTA₀*, *INTA₁* и *INTA₂* шаљу контролерима периферија *PER₀*, *PER₁* и *PER₂*, респективно. Поред тога, у зависности од тога да ли сигнали *prl_{1..0}* имају вредност 11, 10 или 01, брише се флип-флоп *PRIRQM₀*, *PRIRQM₁* или *PRIRQM₂*, респективно.

ђ) На слици 6.ж је дата шема за генерисање броја улаза у *IV* табелу.

Адресе прекидних рутина за спољашње маскирајуће захтеве за прекид *PRIRQM₀*, *PRIRQM₁* и *PRIRQM₂* се налазе у улазима 2, 3 и 4 *IV* табеле, респективно. Највиши приоритет има прекид *PRIRQM₀*, затим следи прекид *PRIRQM₁* и на крају са најнижим приоритетом долази прекид *PRIRQM₂*.



Слика 6.ж Генерисање броја улаза у IV табелу

Најпре је потребно помоћу кодера приоритета CD формирати сигнале приоритета $prl_{1..0}$ спољашњег маскирајућег прекида највишег нивоа приоритета (слика 6.ђ). Сигнале приоритета $prl_{1..0}$ треба затим користити за формирање броја улаза $BRUL_{7..0}$ (слика 6.ж). Један начин за формирање броја улаза $BRUL_{7..0}$ је да се једна од три вредности броја улаза $DVA_{1..0}$, $TRI_{1..0}$ или $CTR_{1..0}$ селекује сигнаlima $prl_{1..0}$ на излазе мултиплексера MP. Други начин за формирање броја улаза $BRUL_{7..0}$ је да се једна од три вредности броја улаза чита и појави на излазним линијама података меморије ROM са адресе одређене вредностима сигнала $prl_{1..0}$, при чему су претходно у улазе 3, 2 и 1 уписане вредности броја улаза 0000 0010b, 0000 0011b и 0000 0100b, респективно.

е) Секвенца адреса инструкција које се редом извршавају почев од инструкције LOAD #2h са адресе 0100h главног програма и закључно са инструкцијом INTD са адресе 0108h главног програма, приказана је на слици 6.е, карактеристичне ситуације на стеку су дате на слици 6.ж, а објашњења у даљем тексту.

У табели са слике 6.е су у посебним колонама дати: Рб – редни број извршене инструкције, Адреса – адреса инструкције у хексадецималном облику, Инструкција – симболичка ознака инструкције и начина адресирања, ACC – садржај акумулатора ACC (три водеће нуле су изостављене), Стек – ситуација на стеку са слике 6.е, I – вредност бита I регистра програмске статусне речи PSW, L – вредност бита $L_{1..0}$ регистра програмске статусне речи PSW, PRINS – унутрашњи захтев за прекид због инструкције прекида INS, PRIRQM₀ – спољашњи маскирајући захтев за прекид, PRIRQM₁ – спољашњи маскирајући захтев за прекид и PRIRQM₂ – спољашњи маскирајући захтев за прекид. У табели је за сваку инструкцију дата једна врста са стањем на крају фазе *извршавање операције*, а за инструкције које извршавају и фазу *опслуживање прекида* дата је још једна врста са стањем на крају фазе *опслуживање прекида*.

Почетна ситуација на стеку је дата као Ситуација 1 (слика 6.ж). Регистар SP указује на задњу заузету меморијску локацију на адреси А која представља врх стека и чији је садржај небитан, па је означен са xxxx. Сви садржаји на стеку су дати у хексадецималном

облику, а од садржаја програмске статусне речи PSW су дате само појединачне вредности битова I и L_{1..0}, јер су оне једино релевантне.

Инструкција 1: Почетне вредности регистра и индикатора су: PC=100h, ACC=?, I=0, L=00.

Ради се о инструкцији LOAD #2 са адресе 100h, која у акумулатор ACC уписује непосредну величину чија је вредност 2.

Инструкција реагује на прекид, али прекид не постоји, па се прелази на извршавање следеће инструкције.

Инструкција 2: Почетне вредности регистра и индикатора су: PC=102h, ACC=2h, I=0, L=00.

Ради се о инструкцији INTE са адресе 102h, која у бит I програмске статусне речи PSW уписује вредност 1. Током извршавања инструкције јавља се спољашњи маскирајући прекид на линији IRQM₂, па се флип-флоп PRIRQM₂ поставља на 1.

Инструкција не реагује на прекид, мада прекид PRIRQM₂ постоји, и прелази се на извршавање следеће инструкције.

Инструкција 3: Почетне вредности регистра и индикатора су: PC=103h, ACC=2h, I=1, L=00.

Ради се о инструкцији INC са адресе 103h, која увећава садржај акумулатора ACC за један, па садржај акумулатора ACC постаје 0003h.

Инструкција реагује на прекид и прекид PRIRQM₂ постоји. На прекиде се реагује по опадајућим приоритетима. Унутрашњег прекида PRINS, који представља прекид највишег приоритета, нема. У оквиру спољашњих маскирајућих прекида, чији је приоритет нижи од приоритета прекида PRINS, највиши ниво приоритета има прекид PRIRQM₀, затим следи прекид PRIRQM₁ и на крају долази са најнижим нивоом приоритета прекид PRIRQM₂. Спољашњих маскирајућих прекида PRIRQM₀ и PRIRQM₁ нема, па се прихвата прекид PRIRQM₂. Како је његов ниво приоритета prl_{1..0}, који износи 01b, виши од нивоа приоритета главног програма PSWL_{1..0}, који износи 00b, и како маскирајући прекиди нису маскирани, јер бит PSWI има вредност 1, сигнал прихваћеног спољашњег маскирајућег прекида printg има вредност 1 (слика 6.ђ). Стога се у оквиру фазе *опслуживање прекида* најпре на стеку чува контекст процесора програма чије се извршавање прекида, а затим се формира контекст процесора прекидне рутине на чије се извршавање прелази.

У корацима чувања контекста процесора програма чије се извршавање прекида на стеку се чува прво програмска статусна реч PSW, а потом и програмски бројач PC. Код програмске статусне речи PSW која се чува на стеку бит I има вредност 1 и бити L_{1..0} имају вредност 00b. Програмски бројач PC има вредност 0104h, која показује на прву следећу инструкцију програма који се прекида. Програмски бројач PC је дужине 16 бита и приликом његовог стављања на стек његов садржај се смешта у једну меморијску локацију јер је ширина једне меморијске локације 16 бита. У једну меморијску локацију се смешта и садржај програмске статусне речи PSW, јер се из поставке задатка може утврдити да поред уобичајених битова N, Z, C и V (задатак 1 - Дискусија) постоје још само битови I и L_{1..0} и да је дужина PSW мања од 16 бита.

У корацима формирања контекста процесора прекидне рутине на чије се извршавање прелази потребно је поставити нове вредности у програмску статусну реч PSW и програмски бројач PC. У бит I регистра PSW се уписује вредност 0, док се у битове L_{1..0} регистра PSW уписује вредност 01b. Треба уочити да се у битове L_{1..0} регистра PSW

уписује нова вредност само приликом преласка на неку од прекидних рутина за спољашње маскирајуће прекиде PRIRQM₀, PRIRQM₁ или PRIRQM₂.

Улази у IV табелу нису фиксни, па процесор, пошто се ради о маскирајућем прекиду PRIRQM₂, најпре по линија INTA₂ шаље контролеру периферије PER2 сигнал потврде и флип-флоп PRIRQM₂ брише на 0, а затим од контролера периферије PER2 добија број улаза чија је вредност 4. Адреса улаза 4 у IV табели је 6 ($6 = 2 + 4 \cdot 1$), па процесор најпре са адресе 6 чита вредност 1009h, која представља адресу прекидне рутине, а затим ту вредност уписује у програмски бројач PC и прелази на фазу *читање инструкције* прве инструкције прекидне рутине. На стеку је Ситуација 2 (слика 6.ж).

Инструкција 4: Почетне вредности регистара и индикатора су: PC=1009h, ACC=3h, I=0, L=01.

Ради се о инструкцији INTE са адресе 1009h, која у бит I програмске статусне речи PSW уписује вредност 1 чиме се дозвољавају спољашњи маскирајући прекиди.

Инструкција не реагује на прекид и прелази се на извршавање следеће инструкције.

Инструкција 5: Почетне вредности регистара и индикатора су: PC=100Ah, ACC=3h, I=1, L=01.

Ради се о инструкцији INC са адресе 100Ah, која увећава садржај акумулатора ACC за један, па садржај акумулатора ACC постаје 0004h. Током извршавања инструкције јавља се спољашњи маскирајући прекид на линији IRQM₀, па се флип-флоп PRIRQM₀ поставља на 1.

Инструкција реагује на прекид и прекид PRIRQM₀ постоји. На прекиде се реагује по опадајућим приоритетима. Унутрашњег прекида PRINS, који представља прекид највишег приоритета, нема, па се спољашњи маскирајући прекид PRIRQM₀, чији је приоритет нижи од приоритета прекида PRINS, али највиши од маскирајућих прекида, прихвата. Како је његов ниво приоритета prl_{1..0}, који износи 11b, виши од нивоа приоритета главног програма PSWL_{1..0}, који износи 01b, и како маскирајући прекиди нису маскирани, јер бит PSWI има вредност 1, сигнал прихваћеног спољашњег маскирајућег прекида printg има вредност 1 (слика 6.ђ). Стога се у оквиру фазе *ослуживање прекида* најпре на стеку чува контекст процесора програма чије се извршавање прекида, а затим се формира контекст процесора прекидне рутине на чије се извршавање прелази.

У корацима чувања контекста процесора програма чије се извршавање прекида на стеку се чува прво програмска статусна реч PSW, а потом и програмски бројач PC. Код програмске статусне речи PSW која се чува на стеку бит I има вредност 1 и бити L_{1..0} имају вредност 01b. Програмски бројач PC има вредност 100Bh, која показује на прву следећу инструкцију програма који се прекида.

У корацима формирања контекста процесора прекидне рутине на чије се извршавање прелази потребно је поставити нове вредности у програмску статусну реч PSW и програмски бројач PC. У бит I регистра PSW се уписује вредност 0, док се у битове L_{1..0} регистра PSW уписује вредност 11b. Треба уочити да се у битове L_{1..0} регистра PSW уписује нова вредност само приликом преласка на неку од прекидних рутина за спољашње маскирајуће прекиде PRIRQM₀, PRIRQM₁ или PRIRQM₂.

Улази у IV табелу нису фиксни, па процесор, пошто се ради о маскирајућем прекиду PRIRQM₀, најпре по линија INTA₀ шаље контролеру периферије PER0 сигнал потврде и флип-флоп PRIRQM₀ брише на 0, а затим од контролера периферије PER0 добија број улаза чија је вредност 2. Адреса улаза 2 у IV табели је 4 ($4 = 2 + 2 \cdot 1$), па процесор најпре са адресе 4 чита вредност 1000h, која представља адресу прекидне рутине, а затим ту

вредност уписује у програмски бројач PC и прелази на фазу *читање инструкције* прве инструкције прекидне рутине. На стеку је Ситуација 3 (слика 6.ж).

Инструкција 6: Почетне вредности регистара и индикатора су: PC=1000h, ACC=4h, I=0, L=11.

Ради се о инструкцији PUSH са адресе 1000h, која садржај акумулатора ACC ставља на стек. На овај начин се чува вредност акумулатора ACC јер ће се унутар прекидне рутине мењати садржај акумулатора, а потребно је да по повратку из ове прекидне рутине вредност акумулатора ACC остане иста као и приликом уласка у прекидну рутину. На стеку је Ситуација 4 (слика 6.е).

Инструкција реагује на прекид, али не постоји прекид, па се прелази на извршавање следеће инструкције.

Инструкција 7: Почетне вредности регистара и индикатора су: PC=1001h, ACC=4h, I=0, L=11.

Ради се о инструкцији INTE са адресе 1001h, која у бит I програмске статусне речи PSW уписује вредност 1 чиме се дозвољавају спољашњи маскирајући прекиди.

Инструкција не реагује на прекид, па се прелази се на извршавање следеће инструкције.

Инструкција 8: Почетне вредности регистара и индикатора су: PC=1002h, ACC=4h, I=0, L=11.

Ради се о инструкцији LOAD 1 са адресе 1002h, која у акумулатор ACC уписује садржај са адресе 0001h, а тамо се налази 0001h.

Инструкција реагује на прекид, али не постоји прекид, па се прелази на извршавање следеће инструкције.

Инструкција 9: Почетне вредности регистара и индикатора су: PC=1004h, ACC=1h, I=1, L=11.

Ради се о инструкцији INC са адресе 1004h, која увећава садржај акумулатора ACC за један, па садржај акумулатора ACC постаје 0002h.

Инструкција реагује на прекид, али не постоји прекид, па се прелази на извршавање следеће инструкције.

Инструкција 10: Почетне вредности регистара и индикатора су: PC=1005h, ACC=2h, I=0, L=11.

Ради се о инструкцији STORE 1 са адресе 1005h, која у меморијску локацију на адреси 0001h уписује садржај акумулатора чија је вредност 0002h.

Инструкција реагује на прекид, али прекид не постоји, па се прелази на извршавање следеће инструкције.

Инструкција 11: Почетне вредности регистара и индикатора су: PC=1007h, ACC=2h, I=1, L=11.

Ради се о инструкцији POP са адресе 1007h, која са стека скида једну реч и уписује је у акумулатор ACC. На овај начин се рестаурира сачувана вредност акумулатора. На стеку је Ситуација 5 (слика 6.ж).

Инструкција реагује на прекид, али прекид не постоји, па се прелази на извршавање следеће инструкције.

Инструкција 12: Почетне вредности регистара и индикатора су: PC=1008h, ACC=4h, I=1, L=11.

Ради се о инструкцији RTI са адресе 1008h, која са стека скида контекст који је сачуван на стеку приликом преласка на ову прекидну рутину и уписује у програмски бројач PC и у програмску статусну реч PSW. Регистар PC добија вредност 100B, док у регистру PSW бит I вредност 1, а бити L вредност 01. На стеку је Ситуација 6 (слика б.ж).

Инструкција не реагује на прекид и прелази се на извршавање следеће инструкције.

Инструкција 13: Почетне вредности регистара и индикатора су: PC=100Bh, ACC=4h, I=1, L=01.

Ради се о инструкцији RTI са адресе 100Bh, која са стека скида контекст који је сачуван на стеку приликом преласка на ову прекидну рутину и уписује у програмски бројач PC и у програмску статусну реч PSW. Регистар PC добија вредност 104h, док у регистру PSW бит I вредност 1, а бити L вредност 00. На стеку је Ситуација 7 (слика б.ж).

Инструкција не реагује на прекид и прелази се на извршавање следеће инструкције.

Инструкција 14: Почетне вредности регистара и индикатора су: PC=104h, ACC=4h, I=1, L=00.

Ради се о инструкцији ADD #2 са адресе 104h, која на почетну вредност акумулатора ACC, која износи 0004h, додаје непосредну величину, чија је вредност 0002h, и у акумулатор ACC уписује резултат, чија је вредност 0006h.

Инструкција реагује на прекид, али прекид не постоји, па се прелази на извршавање следеће инструкције.

Инструкција 15: Почетне вредности регистара и индикатора су: PC=106h, ACC=6h, I=1, L=00.

Ради се о инструкцији INS #3 са адресе 106h, која у фази *извршавање операције* у флип-флоп PRINS уписује вредност 1. Тиме процесор генерише унутрашњи захтев за прекид због извршавања инструкције прекида INS.

Инструкција реагује на прекид и прекид PRINS постоји. На прекиде се реагује по опадајућим приоритетима. Унутрашњи прекид PRINS, који представља прекид највишег приоритета, постоји, па се овај прекид прихвата. Стога се у оквиру фазе *опслуживање прекида* најпре на стеку чува контекст процесора програма чије се извршавање прекида, а затим се формира контекст процесора прекидне рутине на чије се извршавање прелази.

У корацима чувања контекста процесора програма чије се извршавање прекида на стеку се чува прво програмска статусна реч PSW, а потом и програмски бројач PC. Код програмске статусне речи PSW која се чува на стеку бит I има вредност 1 и бити L_{1..0} имају вредност 00b. Програмски бројач PC има вредност 0108h, која показује на прву следећу инструкцију програма који се прекида.

У корацима формирања контекста процесора прекидне рутине на чије се извршавање прелази потребно је поставити нове вредности у програмску статусну реч PSW и програмски бројач PC. У бит I регистра PSW се уписује вредност 0, док се у битова L_{1..0} регистра PSW ништа не уписује и њихова вредност остаје 00b. Треба уочити да се у битове L_{1..0} регистра PSW уписује нова вредност само приликом преласка на неку од прекидних рутину за спољашње маскирајуће прекиде PRIRQM₀, PRIRQM₁ или PRIRQM₂. Број улаза у IV табелу, који је дат као непосредна величина 3, се узима из адресног дела инструкције INS и флип-флоп PRINS брише на 0. Адреса улаза 3 у IV табели је 5 (5 = 2 + 3·1), па процесор најпре са адресе 0005h чита вредност 100Ch, која представља адресу прекидне рутине, а затим ту вредност уписује у програмски бројач PC и прелази на фазу *читање инструкције* прве инструкције прекидне рутине. На стеку је Ситуација 8 (слика б.ж).

Инструкција 16: Почетне вредности регистара и индикатора су: PC=100Ch, ACC=6h, I=0, L=00.

Ради се о инструкцији INC са адресе 100C h, која увећава садржај акумулатора ACC за један, па садржај акумулатора ACC постаје 0007h.

Инструкција реагује на прекид, али не постоји прекид, па се прелази на извршавање следеће инструкције.

Инструкција 17: Почетне вредности регистара и индикатора су: PC=100Dh, ACC=7h, I=0, L=00.

Ради се о инструкцији RTI са адресе 100D, која са стека скида контекст који је сачуван на стеку приликом преласка на ову прекидну рутину и уписује у програмски бројач PC и у програмску статусну реч PSW. Регистар PC добија вредност 108h, док у регистру PSW бит I вредност 1, а бити L вредност 00. На стеку је Ситуација 9 (слика б.ж).

Инструкција не реагује на прекид и прелази се на извршавање следеће инструкције.

Инструкција 18: Почетне вредности регистара и индикатора су: PC=108h, ACC=7h, I=1, L=00.

Ради се о инструкцији INTD са адресе 108h, која у бит I програмске статусне речи уписује вредност 0, чиме се маскирају спољашњи маскирајући прекиди .

Инструкција не реагује на прекид и прелази се на извршавање следеће инструкције.

Сумарно:

У фази *извршавање операције 2.* инструкције INTE главног програма са адресе 0102h у бит I програмске статусне речи PSW се уписује вредност 1 чиме се дозвољавају спољашњи маскирајући прекиди PRIRQM₀, PRIRQM₁ или PRIRQM₂. У току извршавања ове инструкције јавља се спољашњи маскирајући захтев за прекид PRIRQM₂. Инструкција INTE не реагује на прекиде, па се, иако постоји прекид PRIRQM₂, при чему прекид PRIRQM₂ није маскиран јер PSWI има вредност 1 и његов приоритет је виши од приоритета главног програма, прелази на следећу инструкцију. У фази *опслуживање прекида 3.* инструкције INC главног програма са адресе 0103h прелази се на извршавање 4. инструкције INTE прекидне рутине за PRIRQM₂ са адресе 1009h.

У фази *извршавање операције 4.* инструкције INTE прекидне рутине за PRIRQM₂ са адресе 1009h у бит I програмске статусне речи PSW се уписује вредност 1 чиме се дозвољавају спољашњи маскирајући прекиди PRIRQM₀, PRIRQM₁ или PRIRQM₂. У току извршавања 5. инструкције INC са адресе 100Ah прекидне рутине за PRIRQM₂ јавља се спољашњи маскирајући прекид PRIRQM₀ чији је приоритет виши од приоритета прекидне рутине за PRIRQM₂. Стога се у фази *опслуживање прекида* инструкције INC прекид PRIRQM₀ прихвата и прелази на извршавање 6. инструкције PUSH прекидне рутине за PRIRQM₀ са адресе 1000h.

У фази *извршавање операције 12.* инструкције RTI прекидне рутине за PRIRQM₀ са адресе 1008h програмски бројач PC се рестаурира на вредност 100Bh, чиме се наставља извршавање прекидне рутине PRIRQM₂.

У фази *извршавање операције 13.* инструкције RTI прекидне рутине за PRIRQM₂ са адресе 100Bh програмски бројач PC се рестаурира на вредност 0104h, чиме се наставља извршавање главног програма.

У фази *извршавање операције 15.* инструкције INS #3h главног програма са адресе 0106h у флип-флоп PRINS се уписује вредност 1, чиме процесор генерише унутрашњи захтев за прекид због извршавања инструкције прекида INS. У фази *опслуживање прекида*

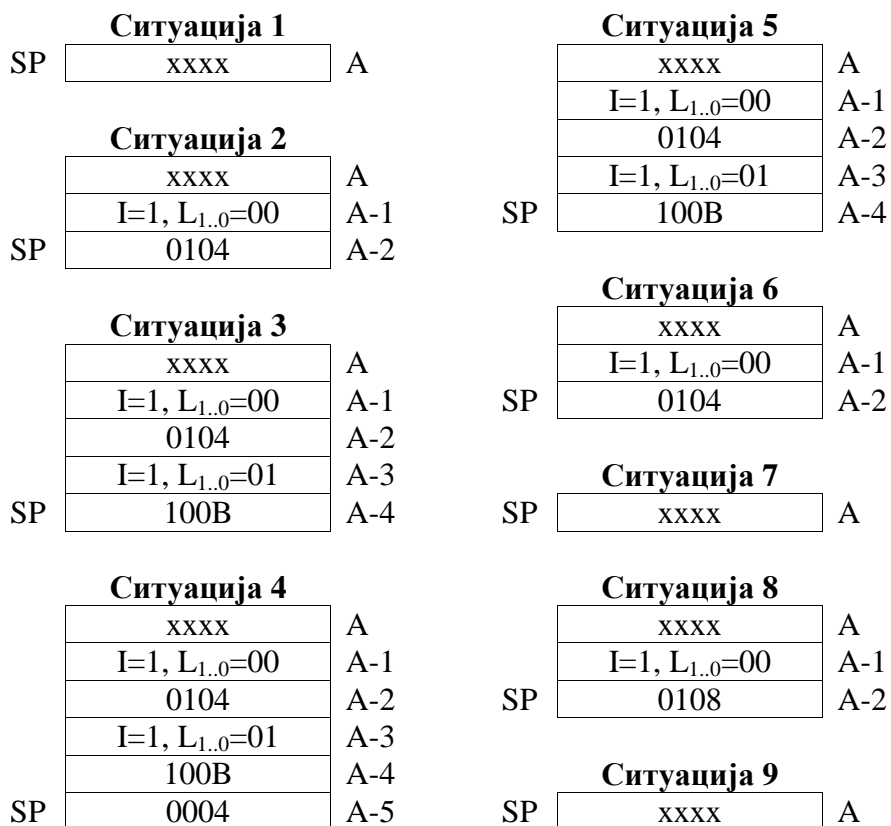
инструкције INS прекид PRINS се прихвата и прелази на извршавање 16. инструкције INC прекидне рутине за PRINS са адресе 100Ch.

У фази *извршавање операције* 17. инструкције RTI прекидне рутине за PRINS са адресе 100Dh програмски бројач PC се рестаурира на вредност 0108h, чиме се наставља извршавање главног програма.

У фази *извршавање операције* 18. инструкције INTD главног програма са адресе 0108h у бит I програмске статусне речи PSW се уписује вредност 0 чиме се маскирају спољашњи маскирајући прекиди PRIRQM₀, PRIRQM₁ или PRIRQM₂..

Pб	Адреса	Инструкција	ACC	Стек	I	L	PRINS	PRIRQM ₀	PRIRQM ₁	PRIRQM ₂
0	-	-	?	1	0	00	0	0	0	0
1	0100	LOAD #2	2	1	0	00	0	0	0	0
2	0102	INTE	2	1	1	00	0	0	0	1
3	0103	INC	3	1	1	00	0	0	0	1
			3	2	0	01	0	0	0	0
4	1009	INTE	3	2	1	01	0	0	0	0
5	100A	INC	4	2	1	01	0	1	0	0
			4	3	0	11	0	0	0	0
6	1000	PUSH	4	4	0	11	0	0	0	0
7	1001	INTE	4	4	1	11	0	0	0	0
8	1002	LOAD 1h	1	4	1	11	0	0	0	0
9	1004	INC	2	4	1	11	0	0	0	0
10	1005	STORE 1h	2	4	1	11	0	0	0	0
11	1007	POP	4	5	1	11	0	0	0	0
12	1008	RTI	4	6	1	01	0	0	0	0
13	100B	RTI	4	7	1	00	0	0	0	0
14	0104	ADD #2	6	7	1	00	0	0	0	0
15	0106	INS #3	6	7	1	00	1	0	0	0
			6	8	0	00	0	0	0	0
16	100C	INC	7	8	0	00	0	0	0	0
17	100D	RTI	7	9	1	00	0	0	0	0
18	0108	INTD	7	9	0	00	0	0	0	0

Слика б.е. Табела са секвенцом инструкција које се извршавају



Слика 6.ж Изглед стека у току извршавања секвенце инструкција

ж) Након извршења 10. инструкције STORE 1 на адреси 0105h у главном програму, вредност на адреси 1h је промењена, јер је у фази *извршавање операције* инструкције STORE 1 садржај акумулатора ACC, чија је вредност тада 0002h, уписан у меморијску локацију на адреси 0001 h. До краја секвенце под г) ниједна инструкција не врши уписивање у дату меморијску локацију, па је зато на крају секвенце инструкција MEM[00001]= 0002h.

1.7 ЗАДАТАК 7

Оперативна меморија неког рачунара је капацитета 16GB, а ширина речи меморије је 32 бита. Процесор је једноадресни, акумулатор је дужине 32 бита и све инструкције се извршавају над 32 битним величинама. Улазно/излазни и меморијски адресни простори су раздвојени. Механизам прекида је векторисан. Адресе прекидних рутина се чувају у табели адреса прекидних рутина (IV табела - *Interrupt Vector Table*) која почиње на адреси 0h оперативне меморије и има 4 улаза.

Контролери периферија PER1, PER2 и PER3 шаљу процесору спољашње маскирајуће захтеве за прекид преко улазних линија IRQM₁, IRQM₂ и IRQM₃, а од процесора добијају сигнале потврда преко излазних линија INTA₁, INTA₂ и INTA₃. Прекидне рутине периферија PER1, PER2 и PER3 започињу на адресама 1000h, 1007h и 1014h, респективно. Контролер периферије шаље процесору захтев за прекид *intr* као импулс и када од процесора прими сигнал потврде *inta* процесору шаљу свој број улаза у IV табелу. Адресе 32 битних регистара контролера периферија PER1, PER2 и PER3 у којима се чувају бројеви улаза у IV табелу су 8h, 20h и 32h, респективно.

Процесор може да генерише један унутрашњи захтев за прекид због извршавања инструкције прекида INS. Претпоставити да је извршавање инструкције INS тако реализовано да се у фази *извршавање операције* само у флип-флоп PRINS уписује вредност 1 и одмах прелази на фазу *опслуживање прекида*. У оквиру фазе *опслуживање прекида* се прелази на прекидну рутину на исти начин као и за све остале прекиде.

Највиши приоритет има унутрашњи захтев за прекид због извршавања инструкције прекида INS, док ниже приоритете имају спољашњи маскирајући захтеви за прекид. У оквиру спољашњи маскирајућих захтева за прекид највиши ниво приоритета има захтев за прекид који је везан на линију IRQM₃, затим захтев за прекид који је везан на линију IRQM₂, док најнижи ниво приоритета има захтев за прекид који је везан на линију IRQM₁.

У процесору постоји регистар PSW (*Program Status Word*) са битом I (*Interrupt enable*) у разреду 2 и два бита L_{1..0} (*priority Level*) у разредима 1 и 0. Бит I вредношћу 1 дозвољава прихватање спољашњих маскирајућих захтева за прекид. Овај бит се поставља на вредност 0 хардверски током извршавања фазе *опслуживање прекида*. Битови L_{1..0} садрже ниво приоритета текућег програма и користе се да се само приликом извршавања главног програма и прекидних рутина спољашњих маскирајућих захтева за прекид реализује механизам којим се прихватају спољашњих маскирајући захтеви за прекид једино уколико су вишег нивоа приоритета од нивоа приоритета текућег програма. Ови битови се постављају на вредност нивоа приоритета периферије на чију прекидну рутину се прелази хардверски током извршавања фазе *опслуживање прекида*.

У оквиру извршавања фазе *опслуживање прекида* на стеку се хардверски чувају ACC, PC и PSW тим редом. Стек расте према нижим локацијама, а указивач на врх стека SP указује на прву слободну локацију. Инструкције INTE, INTD и RTI не реагују на спољашње захтеве за прекид.

Садржај дела меморијског адресног простора почев од адресе 0h дат је на слици 7.а. Главни програм је дата на слици 7.б, а прекидне рутине на слици 7.в. Инструкција на адреси 0100h означена је као 1. (прва) по редоследу извршавања, а свака следећа инструкција која се извршава означена је следећим редним бројем. У току извршавања 1. инструкције стиже захтев за прекид од контролера периферије PER1, у току 5. инструкције од контролера периферије PER2, а у току 11. инструкције од контролера периферије PER3.

На почетку су сви бити програмске статусне речи PSW-а постављени на 0. Почетни садржај акумулатора ACC је 2h, локације 13h је 6h, а локације 23h је 1h.

Напомена: Код писања 32 битних адреса на сликама 7.а, 7.б. и 7.в. и 32 битних садржаја на слици 7.а. у хексадецималном облику ради прегледности су изостављене четири водеће нуле. Поред тога и у тексту су изостављене све водеће нуле код писања 32 битних адреса и 32 битних садржаја.

адреса	садржај
0000h	1000h
0001h	1007h
0002h	1014h
0003h	100Dh
0004h	1004h
0005h	1002h

Слика 7.а Изглед дела меморије

адреса	инструкција
0100	INTE
0101	INC
0102	AND #6
0104	OR #6

Слика 7.б Изглед дела главног програма

адреса	инструкција
1000h	INS #3
1001h	INTE
1002h	INC
1003h	INC
1004h	STORE 15h
1006h	RTI
1007h	LOAD 13h
1009h	INC
100Ah	STORE 13h
100Ch	RTI
100Dh	POP
100Eh	AND #FFFFFF8h
1010h	OR 23h
1012h	PUSH
1013h	RTI
1014h	INC
1015h	RTI

Слика 7.в Прекидне рутине

- а) Навести по којим линијама периферија PER1 шаље број улаза у IV табелу.
- б) Навести која вредност се налази на магистрала података када у тачки а) периферија PER1 шаље број улаза у IV табелу
- в) Написати секвенцу инструкција којом се периферији PER3 додељује број улаза у IV табелу.
- г) Написати секвенцу адреса инструкција које се редом извршавају почев од инструкције са адресе 0100h и закључно са инструкцијом са адресе 0104h. Резултат дати табеларно тако да табела садржи редни број интрукције, адресу на којој започиње инструкција, симболичку ознаку инструкције, садржај акумулатора након извршења

инструкције, ситуацију на стеку, вредности бита I и L програмске статусне речи PSW и вредности флип-флопова PRINS, PRIRQM₃, PRIRQM₂ и PRIRQM₀ који указују да ли постоје прекиди због инструкције прекида и спољашњи маскирајући прекиди.

д) Поновити поступак под г) уколико почетни садржај локације 23h има вредност 2.

Решење:

а) Контролер периферије PER1 шаље процесору спољашњи маскирајући захтев за прекид преко улазне линије IRQM₁, а од процесора добија сигнале потврде преко излазне линије INTA₁ током извршавања фазе *опслуживање прекида* оне инструкције процесора у оквиру које се прихвата захтев за прекид периферије PER1. Вредношћу 1 сигнала INTA₁ процесор као газда започиње на магистралу циклус прихватање броја улаза у оквиру кога контролер периферије PER1 као слуга шаље процесору преко магистрале података садржај регистра броја улаза у који је током иницијализације система уписан број улаза у IV табелу у коме се налази адреса прекидне рутине периферије PER1.

Треба уочити да је циклус прихватање броја улаза сличан циклусу читања. Разлика је у томе да процесор, уместо да шаље адресу регистра броја улаза контролера периферије PER1 на адресну магистралу и генерисања сигнала читања на управљачкој магистралу, шаље контролеру периферије PER1 по посебној линији INTA₁ индикацију да треба да пусти на магистралу података садржај свог регистра броја улаза.

б) На магистралу података се налази број улаза у IV табелу који припада PER1, која је повезана на линију IRQM₁.

Капацитет оперативне меморије је 16 GB. Како је ширина речи меморије 32 бита, то оперативна меморија има 2³² адресибилних локација које садрже речи меморије ширине 32 бита. Стога су адресе које се користе за адресирање меморијских локација широке 32 бита.

Табела адреса прекидних рутина састоји се од 4 улаза који садрже почетне адресе прекидних рутина. Како су адресе широке 32 бита, а ширина речи меморије је 32 бита, сваки улаз табеле заузима једну локацију у меморији. Почетна адреса IV табеле, која се чува у регистру IVTP (*Interrupt Vector Table Pointer*), је 0h.

Како се једино на адреси 0h у IV табели налази вредност 1000h која представља адресу прекидне рутине периферије PER1, следи да је 0h тражени број улаза IV табелу који се налази на магистралу података.

в) Регистар контролера периферије PER3 у који треба уписати број улаза у IV табелу периферије PER3 налази се на адреси 32h у улазно/излазном адресном простору који је раздвојен од меморијског адресног простора. Како се једино на адреси 2h у IV табели налази вредност 1014h која представља адресу прекидне рутине периферије PER3, утврђује се да је 2h број улаза у IV табелу периферије PER3. Тражени програм је приказан на слици 7.г.

```
LOAD    #2      ;napuni akumulator konstantom 2
OUT 32h  ;prebaci sadržaj akumulatora na adresu 32h u ulazno/izlaznom adresnom prostoru
```

Слика 7.г Додела броја улаза у IV табели за PER3

г) Секвенца адреса инструкција које се редом извршавају, почев од инструкције INTE са адресе 0100h главног програма и закључно са инструкцијом OR #6 са адресе 0104h главног програма, приказана је на слици 7.д, карактеристичне ситуације на стеку су дате на слици 7.ђ, а објашњења у даљем тексту.

У табели са слике 7.д су у посебним колонама дати: Рб – редни број извршене инструкције, Адреса – адреса инструкције у хексадецималном облику (четири водеће нуле су изостављене), Инструкција – симболичка ознака инструкције и начина адресирања, АСС – садржај акумулатора АСС, Стек – ситуација на стеку са слике 4.ђ, I – вредност бита I регистра програмске статусне речи PSW, L– вредност бита L_{1..0} регистра програмске

статусне речи PSW, PRINS – унутрашњи захтев за прекид због инструкције прекида INS, PRIRQM₃ – спољашњи маскирајући захтев за прекид, PRIRQM₂ – спољашњи маскирајући захтев за прекид и PRIRQM₁ – спољашњи маскирајући захтев за прекид. У табели је за сваку инструкцију дата једна врста са стањем на крају фазе *извршавање инструкције*, а за инструкције које извршавају и фазу *опслуживање прекида* дата је још једна врста са стањем на крају фазе *опслуживање прекида*.

Унутрашњи захтев за прекид због инструкције прекида се генерише софтверски у фази *извршавање операције* инструкције INS тако што се у флип-флоп PRINS упише вредност 1. Спољашњи маскирајући захтеви за прекид који се повезују на улазне линије процесора IRQM₃, IRQM₂ и IRQM₁ долази као импулс, па се у процесору памте у флип-фловима PRIRQM₃, PRIRQM₂ и PRIRQM₁, респективно.

Да би се остварио механизам којим се

прекида извршавање главног програма уколико се јави било који од спољашњих маскирајућих захтева за прекид и

утврђује који се од захтева за прекид који долазе преко линија IRQM₃, IRQM₂ и IRQM₁ прихвата,

главном програму и прекидним рутинама периферија PER3, PER2 и PER1 се додељују нивои приоритета. У овом случају постоји главни програм и три прекидне рутине периферија, па су четири бинарне вредности 00, 01, 10 и 11 довољне за кодирање њихових нивоа приоритета. Да би се омогућило прекидање извршавања главног програма када стигне захтев за прекид од било ког контролера периферија PER3, PER2 и PER1, главном програму се додељује најнижи ниво приоритета 00 а прекидним рутинама периферија PER3, PER2 и PER1 преостали виши нивои приоритета 01, 10 и 11. Пошто се жели да у оквиру периферија PER3, PER2 и PER1 периферија PER3 може да прекида извршавање прекидних рутина периферија PER2 и PER1 и да у оквиру периферија PER2 и PER1 периферија PER2 може да прекида извршавање прекидне рутине периферије PER1, тада периферија PER3 треба да има највиши ниво приоритета, периферија PER2 нижи и периферија PER1 најнижи. У складу са тим периферијама PER3, PER2 и PER1 треба доделити нивое приоритета 11, 10 и 01, респективно. Ниво приоритета текућег програма се чува у разредима PSWL_{1..0} регистра PSW.

Почетна ситуација на стеку је дата као Ситуација 1 (слика 7.ђ). Регистар SP указује на прву слободну меморијску локацију на адреси А која представља врх стека и чији је садржај небитан, па је означен са xxxx xxxx. Сви садржаји на стеку су дати у хексадецималном облику, а од садржаја програмске статусне речи PSW су дате само појединачне вредности битова I и L, јер су они једино релевантни.

Инструкција 1: Почетне вредности регистара и индикатора су: PC=100h, ACC=2, I=0, L=00.

Ради се о инструкцији INTE са адресе 0100h, која у бит I програмске статусне речи PSW уписује вредност 1 чиме се дозвољавају спољашњи маскирајући прекиди. Током извршавања инструкције јавља се спољашњи маскирајући прекид на линији IRQM₁, па се флип-флоп PRIRQM₁ поставља на 1.

Инструкција не реагује на прекид, па се, мада прекид PRIRQM₁ постоји, прелази на извршавање следеће инструкције.

Инструкција 2: Почетне вредности регистара и индикатора су: PC=101h, ACC=2, I=1, L=00.

Ради се о инструкцији INC са адресе 101h, која увећава садржај акумулатора ACC за један, па садржај акумулатора ACC постаје 0003h.

Инструкција реагује на прекид и прекид PRIRQM₁ постоји. На прекиде се реагује по опадајућим приоритетима. Унутрашњег прекида PRINS, који представља прекид највишег приоритета, нема. У оквиру спољашњих маскирајућих прекида, чији је приоритет нижи од приоритета прекида PRINS, највиши ниво приоритета има прекид PRIRQM₃, затим следи прекид PRIRQM₂ и на крају долази са најнижим нивоом приоритета прекид PRIRQM₁. Спољашњих маскирајућих прекида PRIRQM₃ и PRIRQM₂ нема, па се прихвата прекид PRIRQM₁. Како је његов ниво приоритета $rg_{1..0}$, који износи 01b, виши од нивоа приоритета главног програма PSWL_{1..0}, који износи 00b, и како маскирајући прекиди нису маскирани, јер бит PSWI има вредност 1, сигнал прихваћеног спољашњег маскирајућег прекида rg_{intg} има вредност 1. Стога се у оквиру фазе *опслуживање прекида* најпре на стеку чува контекст процесора програма чије се извршавање прекида, а затим се формира контекст процесора прекидне рутине на чије се извршавање прелази.

У корацима чувања контекста процесора програма чије се извршавање прекида на стеку се чува прво вредност акумулатора ACC, затим програмски бројач PC и на крају програмска статусна реч PSW. Акумулатор ACC има вредност 3. Програмски бројач PC има вредност 0102h, која показује на прву следећу инструкцију програма који се прекида. Код програмске статусне речи PSW која се чува на стеку бит I има вредност 1 и бити L_{1..0} имају вредност 00b. Акумулатор је дужине 32 бита и приликом његовог стављања на стек његов садржај се смешта у једну меморијску локацију јер је ширина једне меморијске локације 32 бита. Исто важи и за програмски бројач PC чија је дужина такође 32 бита. У једну меморијску локацију се смешта и садржај програмске статусне речи PSW, јер се из поставке задатка може утврдити да поред уобичајених битова N, Z, C и V (задатак 1 - Дискусија) постоје још само битови I и L_{1..0} и да је дужина PSW мања од 32 бита.

У корацима формирања контекста процесора прекидне рутине на чије се извршавање прелази потребно је поставити нове вредности у програмску статусну реч PSW и програмски бројач PC. У бит I регистра PSW се уписује вредност 0, док се у битове L_{1..0} регистра PSW уписује вредност 01b. Треба уочити да се у битове L_{1..0} регистра PSW уписује нова вредност само приликом преласка на неку од прекидних рутина за спољашње маскирајуће прекиде PRIRQM₃, PRIRQM₂ или PRIRQM₁.

Улази у IV табелу нису фиксни, па процесор, пошто се ради о маскирајућем прекиду PRIRQM₁, најпре по линија INTA₁ шаље контролеру периферије PER1 сигнал потврде и флип-флоп PRIRQM₁ брише на 0, а затим од контролера периферије PER1 добија број улаза чија је вредност 0. Адреса улаза 0 у IV табели је 0h ($0 = 0 + 0 \cdot 1$), па процесор најпре са адресе 0h чита вредност 0000 1000h, која представља адресу прекидне рутине, а затим ту вредност уписује у програмски бројач PC и прелази на фазу *читање инструкције* прве инструкције прекидне рутине. На стеку је Ситуација 2 (слика 7.ђ).

Инструкција 3: Почетне вредности регистара и индикатора су: PC=1000h, ACC=3h, I=0, L=01.

Ради се о инструкцији INS #3 са адресе 1000h, која у фази *извршавање операције* у флип-флоп PRINS уписује вредност 1. Тиме процесор генерише унутрашњи захтев за прекид због извршавања инструкције прекида INS.

Инструкција реагује на прекид и прекид PRINS постоји. На прекиде се реагује по опадајућим приоритетима. Унутрашњи прекид PRINS, који представља прекид највишег приоритета, постоји, па се овај прекид прихвата. Стога се у оквиру фазе *опслуживање*

прекида најпре на стеку чува контекст процесора програма чије се извршавање прекида, а затим се формира контекст процесора прекидне рутине на чије се извршавање прелази.

У корацима чувања контекста процесора програма чије се извршавање прекида на стеку се чува прво вредност акумулатора АСС, затим програмски бројач РС и на крају програмска статусна реч PSW. Акумулатор АСС има вредност 3. Програмски бројач РС има вредност 0101h, која показује на прву следећу инструкцију програма који се прекида. Код програмске статусне речи PSW која се чува на стеку бит I има вредност 0, бити L_{1..0} имају вредност 01b, док вредности осталих битова нису битне.

У корацима формирања контекста процесора прекидне рутине на чије се извршавање прелази потребно је поставити нове вредности у програмску статусну реч PSW и програмски бројач РС. У бит I регистра PSW се уписује вредност 0, док се у битова L_{1..0} регистра PSW ништа не уписује и њихова вредност остаје 01b. Треба уочити да се у битове L_{1..0} регистра PSW уписује нова вредност само приликом преласка на неку од прекидних рутина за спољашње маскирајуће прекиде PRIRQM₃, PRIRQM₂ или PRIRQM₁. Број улаза у IV табелу, који је дат као непосредна величина 3, се узима из адресног дела инструкције INS и флип-флоп PRINS брише на 0. Адреса улаза 3 у IV табели је 3 (3 = 0 + 3·1), па процесор најпре са адресе 0000 0003h чита вредност 0000 100Dh, која представља адресу прекидне рутине, а затим ту вредност уписује у програмски бројач РС и прелази на фазу *читање инструкције* прве инструкције прекидне рутине. На стеку је Ситуација 3 (слика 7.ђ).

Инструкција 4: Почетне вредности регистара и индикатора су: PC=100Dh, ACC=3, I=0, L=01.

Ради се о инструкцији POP са адресе 100Dh, која скида садржаја са стека и уписује у акумулатор АСС. Садржај који је скинут са стека и уписан у акумулатор АСС представља садржај програмске статусне речи PSW која је сачуван на стеку у оквиру фазе *опслуживања прекида* инструкције INS #3 са адресе 1000h којом се прешло на ову прекидну рутину. Стога се у акумулатору у разреду 2, који одговара биту I регистра PSW, налази вредност 0, у разредима 1 и 0, који одговарају битовима L_{1..0} регистра PSW, вредност 01b, док преостали битови нису битни. Садржај акумулатора АСС је сада x...x001b, где су са x означени битови 31 до 3 чије вредности нису битне. На стеку је Ситуација 4 (слика 7.ђ).

Инструкција реагује на прекид, али прекид не постоји, па се прелази на извршавање следеће инструкције.

Инструкција 5: Почетне вредности регистара и индикатора су: PC=100Eh, ACC=x...x001b, I=0, L=01.

Ради се о инструкцији AND #FFFFFFF8h са адресе 100Eh, која реализује логичку И операцију садржаја акумулатора АСС који износи x...x001b и непосредне величине 1...1000b (FFFFFFF8h) и резултат x...x000b уписује у акумулатор АСС. Резултат ове операције је да се у разреде 2, 1 и 0 регистра АСС уписује вредност 000b, док се разреде 31 до 3 акумулатора АСС, у којима се налазе вредности које нису битне, не мењају.

Треба уочити да је у акумулатору АСС сада садржај који одговара садржају регистра PSW скинутог са стека, али тако модификован да се у разредима 2, 1 и 0 акумулатора АСС, који одговарају битовима I и L_{1..0} регистра PSW, уместо вредности 001b налази вредност 000b. Током извршавања инструкције јавља се спољашњи маскирајући прекид на линији IRQM₂, па се флип-флоп PRIRQM₂ поставља на 1.

Инструкција реагује на прекид и спољашњи прекид $PRIRQM_2$ постоји, али су сви спољашњи маскирајући прекиди маскирани вредношћу 0 бита I регистра PSW, па се прелази на извршавање следеће инструкције.

Инструкција 6: Почетне вредности регистара и индикатора су: $PC=1010h$, $ACC=x...x000b$, $I=0$, $L=01$.

Ради се о инструкцији OR 23h са адресе 1010h, која реализује логичку ИЛИ операцију садржаја акумулатора ACC који износи $x...x000b$ и садржаја меморијске локације 23h који износи $0...0001b$ (1h) и резултат $x...x001b$ уписује у акумулатор ACC. Резултат ове операције је да се у разред 2, 1 и 0 регистра ACC уписује вредност 001b, док се разреди 31 до 3 акумулатора ACC, у којима се налазе вредности које нису битне, не мењају. Треба уочити да је у акумулатору ACC сада поново садржај који одговара садржају регистра PSW скинутог са стека, јер је резултат извршених логичких операција И и ИЛИ такав да се у разредима 2, 1 и 0 акумулатора ACC, који одговарају битовима I и $L_{1..0}$ регистра PSW, поново налази вредност 001b, док вредности разреди 31 до 3 акумулатора ACC, у којима се налазе вредности које нису битне, нису ни мењане.

Инструкција реагује на прекид и спољашњи прекид $PRIRQM_2$ постоји, али су сви спољашњи маскирајући прекиди маскирани вредношћу 0 бита I регистра PSW, па се прелази на извршавање следеће инструкције.

Инструкција 7: Почетне вредности регистара и индикатора су: $PC=1012h$, $ACC=x...x001b$, $I=0$, $L=01$.

Ради се о инструкцији PUSH са адресе 1012h, која садржај регистра ACC ставља на стек. Садржај акумулатора ACC који се ставља на стек представља оригинални садржај програмске статусне речи PSW који је сачуван на стеку у оквиру фазе *опслуживање прекида* инструкције INS #3 са адресе 1000h којом се прешло на ову прекидну рутину. Овај садржај је инструкцијом POP скинут са стека и уписан у акумулатор ACC, али је после извршавања инструкција AND и OR остао непромењен. На стеку је Ситуација 5 (слика 7.ђ).

Инструкција реагује на прекид и спољашњи прекид $PRIRQM_2$ постоји, али су сви спољашњи маскирајући прекиди маскирани вредношћу 0 бита I регистра PSW, па се прелази на извршавање следеће инструкције.

Инструкција 8: Почетне вредности регистара и индикатора су: $PC=1013h$, $ACC=x...x001b$, $I=0$, $L=01$.

Ради се о инструкцији RTI са адресе 1013h, која са стека скида контекст који је сачуван на стеку приликом преласка на ову прекидну рутину и уписује у програмску статусну реч PSW, програмски бројач PC и у акумулатор ACC. У регистру PSW се поново уписује у бит I вредност 0 и у битове L вредност 01b, регистар PC добија вредност 1001h и акумулатор ACC вредност 3. На стеку је Ситуација 6 (слика 7.ђ).

Инструкција не реагује на прекид, а и спољашњи маскирајући прекид $PRIRQM_2$ који постоји је маскиран вредношћу 0 бита I регистра PSW, па се прелази на извршавање следеће инструкције.

Инструкција 9: Почетне вредности регистара и индикатора су: $PC=1001h$, $ACC=3$, $I=0$, $L=01$.

Ради се о инструкцији INTE са адресе 1001h, која у бит I програмске статусне речи PSW уписује вредност 1 чиме се дозвољавају спољашњи маскирајући прекиди.

Инструкција не реагује на прекид, па се, мада постоји прекид PRIRQM₂, прелази на извршавање следеће инструкције.

Инструкција 10: Почетне вредности регистра и индикатора су: PC=1002h, ACC=3, I=1, L=01.

Ради се о инструкцији INC са адресе 1002h, која увећава садржај акумулатора за један, па садржај акумулатора ACC постаје 0000 0004h.

Инструкција реагује на прекид и прекид PRIRQM₂ постоји. На прекиде се реагује по опадајућим приоритетима. Унутрашњег прекида PRINS, који представља прекид највишег приоритета, нема. У оквиру спољашњих маскирајућих прекида, чији је приоритет нижи од приоритета прекида PRINS, највиши ниво приоритета има прекид PRIRQM₃, затим следи прекид PRIRQM₂ и на крају долази са најнижим нивоом приоритета прекид PRIRQM₁. Спољашњег маскирајућег прекида PRIRQM₃ нема, па се прихвата прекид PRIRQM₂. Како је његов ниво приоритета prl_{1..0}, који износи 10b, виши од нивоа приоритета прекидне рутине периферије PER1, који износи 01b, и како маскирајући прекиди нису маскирани, јер бит PSW има вредност 1, сигнал прихваћеног спољашњег маскирајућег прекида printr има вредност 1. Стога се у оквиру фазе *опслуживање прекида* најпре на стеку чува контекст процесора програма чије се извршавање прекида, а затим се формира контекст процесора прекидне рутине на чије се извршавање прелази.

У корацима чувања контекста процесора програма чије се извршавање прекида на стеку се чува прво вредност акумулатора ACC, затим програмски бројач PC и на крају програмска статусна реч PSW. Акумулатор ACC има вредност 4. Програмски бројач PC има вредност 1003h, која показује на прву следећу инструкцију програма који се прекида. Код програмске статусне речи PSW која се чува на стеку бит I има вредност 1, бити L_{1..0} имају вредност 01b, док вредности осталих битова нису битне.

У корацима формирања контекста процесора прекидне рутине на чије се извршавање прелази потребно је поставити нове вредности у програмску статусну реч PSW и програмски бројач PC. У бит I регистра PSW се уписује вредност 0, док се у битове L_{1..0} регистра PSW уписује вредност 10b. Треба уочити да се у битове L_{1..0} регистра PSW уписује нова вредност само приликом преласка на неку од прекидних рутина за спољашње маскирајуће прекиде PRIRQM₀, PRIRQM₁ или PRIRQM₂.

Улази у IV табелу нису фиксни, па процесор, пошто се ради о маскирајућем прекиду PRIRQM₂, најпре по линија INTA₂ шаље контролеру периферије PER2 сигнал потврде и флип-флоп PRIRQM₂ брише на 0, а затим од контролера периферије PER2 добија број улаза чија је вредност 1. Адреса улаза 1 у IV табели је 1 (1 = 0 + 1·1), па процесор најпре са адресе 0000 0001h чита вредност 0000 1007h, која представља адресу прекидне рутине, а затим ту вредност уписује у програмски бројач PC и прелази на фазу *читање инструкције* прве инструкције прекидне рутине. На стеку је Ситуација 7 (слика 7.ђ).

Инструкција 11: Почетне вредности регистра и индикатора су: PC=1007h, ACC=4, I=0, L=10.

Ради се о инструкцији LOAD 13h са адресе 1007h, која у акумулатор ACC уписује садржај са адресе 13h, а тамо се налази 6h. Током извршавања инструкције јавља се спољашњи маскирајући прекид на линији IRQM₃, па се флип-флоп PRIRQM₃ поставља на 1.

Инструкција реагује на прекид и спољашњи прекид PRIRQM₃ постоји, али су сви спољашњи маскирајући прекиди маскирани вредношћу 0 бита I регистра PSW, па се прелази на извршавање следеће инструкције.

Инструкција 12: Почетне вредности регистара и индикатора су: PC=1009h, ACC=6, I=0, L=10.

Ради се о инструкцији INC са адресе 1009h, која увећава садржај акумулатора ACC за један, па садржај акумулатора ACC постаје 0000 0007h.

Инструкција реагује на прекид и спољашњи прекид PRIRQM₃ постоји, али су сви спољашњи маскирајући прекиди маскирани вредношћу 0 бита I регистра PSW, па се прелази на извршавање следеће инструкције.

Инструкција 13: Почетне вредности регистара и индикатора су: PC=100Ah, ACC=7, I=0, L=10.

Ради се о инструкцији STORE 13h са адресе 100Ah, која у меморијску локацију на адреси 13h уписује садржај акумулатора чија је вредност 0007h.

Инструкција реагује на прекид и спољашњи прекид PRIRQM₃ постоји, али су сви спољашњи маскирајући прекиди маскирани вредношћу 0 бита I регистра PSW, па се прелази на извршавање следеће инструкције.

Инструкција 14: Почетне вредности регистара и индикатора су: PC=100Ch, ACC=7h, I=0, L=10.

Ради се о инструкцији RTI са адресе 100Ch, која са стека скида контекст који је сачуван на стеку приликом преласка на ову прекидну рутину и уписује у програмску статусну реч PSW, програмски бројач PC и у акумулатор ACC. У регистру PSW се уписује у бит I вредност 1 и у битове L вредност 01b, регистар PC добија вредност 1003h и акумулатор ACC вредност 4. На стеку је Ситуација 8 (слика 7.ђ).

Инструкција не реагује на прекид, па се, мада постоји прекид PRIRQM₃, прелази на извршавање следеће инструкције.

Инструкција 15: Почетне вредности регистара и индикатора су: PC=1003h, ACC=4h, I=1, L=01.

Ради се о инструкцији INC са адресе 1003h, која увећава садржај акумулатора за један, па садржај акумулатора ACC постаје 0000 0005h.

Инструкција реагује на прекид и прекид PRIRQM₃ постоји. На прекиде се реагује по опадајућим приоритетима. Унутрашњег прекида PRINS, који представља прекид највишег приоритета, нема. У оквиру спољашњих маскирајућих прекида, чији је приоритет нижи од приоритета прекида PRINS, највиши ниво приоритета има прекид PRIRQM₃, па се прихвата прекид PRIRQM₃. Како је његов ниво приоритета prl_{1..0}, који износи 11b, виши од нивоа приоритета прекидне рутине периферије PER1, који износи 01b, и како маскирајући прекиди нису маскирани, јер бит PSWI има вредност 1, сигнал прихваћеног спољашњег маскирајућег прекида printr има вредност 1. Стога се у оквиру фазе *опслуживање прекида* најпре на стеку чува контекст процесора програма чије се извршавање прекида, а затим се формира контекст процесора прекидне рутине на чије се извршавање прелази.

У корацима чувања контекста процесора програма чије се извршавање прекида на стеку се чува прво вредност акумулатора ACC, затим програмски бројач PC и на крају програмска статусна реч PSW. Акумулатор ACC има вредност 5. Програмски бројач PC има вредност 1004h, која показује на прву следећу инструкцију програма који се прекида. Код програмске статусне речи PSW која се чува на стеку бит I има вредност 1, бити L_{1..0} имају вредност 01b, док вредности осталих битова нису битне.

У корацима формирања контекста процесора прекидне рутине на чије се извршавање прелази потребно је поставити нове вредности у програмску статусну реч PSW и

програмски бројач PC. У бит I регистра PSW се уписује вредност 0, док се у битове L_{1..0} регистра PSW уписује вредност 11b. Треба уочити да се у битове L_{1..0} регистра PSW уписује нова вредност само приликом преласка на неку од прекидних рутина за спољашње маскирајуће прекиде PRIRQM₁, PRIRQM₂ или PRIRQM₃.

Улази у IV табелу нису фиксни, па процесор, пошто се ради о маскирајућем прекиду PRIRQM₃, најпре по линија INTA₃ шаље контролеру периферије PER3 сигнал потврде и флип-флоп PRIRQM₃ брише на 0, а затим од контролера периферије PER3 добија број улаза чија је вредност 2. Адреса улаза 2 у IV табели је 2 (2 = 0 + 2·1), па процесор најпре са адресе 0000 0002h чита вредност 0000 1014h, која представља адресу прекидне рутине, а затим ту вредност уписује у програмски бројач PC и прелази на фазу *читање инструкције* прве инструкције прекидне рутине. На стеку је Ситуација 9 (слика 7.ђ).

Инструкција 16: Почетне вредности регистара и индикатора су: PC=1014h, ACC=5h, I=0, L=11.

Ради се о инструкцији INC са адресе 1014h, која увећава садржај акумулатора за један, па садржај акумулатора ACC постаје 0000 0006h.

Инструкција реагује на прекид, али прекид не постоји, па се прелази на извршавање следеће инструкције.

Инструкција 17: Почетне вредности регистара и индикатора су: PC=1015h, ACC=6h, I=0, L=11.

Ради се о инструкцији RTI са адресе 1015h, која са стека скида контекст који је сачуван на стеку приликом преласка на ову прекидну рутину и уписује у програмску статусну реч PSW, програмски бројач PC и у акумулатор ACC. У регистру PSW се уписује у бит I вредност 1 и у битове L вредност 01b, регистар PC добија вредност 1004h и акумулатор ACC вредност 5. На стеку је Ситуација 10 (слика 7.ђ).

Инструкција не реагује на прекид, па се прелази на извршавање следеће инструкције.

Инструкција 18: Почетне вредности регистара и индикатора су: PC=1004h, ACC=5, I=1, L=01.

Ради се о инструкцији STORE 15h са адресе 1004h, која у меморијску локацију на адреси 15h уписује садржај акумулатора чија је вредност 0005h.

Инструкција реагује на прекид, али прекид не постоји, па се прелази на извршавање следеће инструкције.

Инструкција 19: Почетне вредности регистара и индикатора су: PC=1006h, ACC=5, I=1, L=01.

Ради се о инструкцији RTI са адресе 1006h, која са стека скида контекст који је сачуван на стеку приликом преласка на ову прекидну рутину и уписује у програмску статусну реч PSW, програмски бројач PC и у акумулатор ACC. У регистру PSW се поново уписује у бит I вредност 1 и у битове L вредност 00b, регистар PC добија вредност 0102h и акумулатор ACC вредност 3. На стеку је Ситуација 11 (слика 7.ђ).

Инструкција не реагује на прекид, па се прелази на извршавање следеће инструкције.

Инструкција 20: Почетне вредности регистара и индикатора су: PC=102h, ACC=3, I=1, L=00.

Ради се о инструкцији AND #6h са адресе 102h, која реализује логичку И операцију садржаја акумулатора ACC, који износи 0...00011b, и непосредне величине, која износи 0...0110b (6h), и резултат, који износи 0...0010b, уписује у акумулатор ACC. Треба уочити

да непосредна величина има вредност 1 на битовима 2 и 1 и вредност 0 на битовима 31 до 3 и биту 0. Због тога је резултат логичке И операције такав да у разредима 2 и 1 акумулатора АСС остају непромењене вредности 0 и 1, респективно, док се у разреде 31 до 3 и разред 0 акумулатора АСС уписује вредност 0.

Инструкција реагује на прекид, али прекид не постоји, па се прелази на извршавање следеће инструкције.

Инструкција 21: Почетне вредности регистара и индикатора су: PC=104h, ACC=2, I=1, L=00.

Ради се о инструкцији OR #6h са адресе 104h, која реализује логичку ИЛИ операцију садржаја акумулатора АСС, који износи 0...00010b, и непосредне величине, која износи 0...0110b (6h), и резултат, који износи 0...0110b, уписује у акумулатор АСС.

Треба уочити да непосредна величина има вредност 1 на битовима 2 и 1 и вредност 0 на битовима 31 до 3 и биту 0. Због тога је резултат логичке И операције такав да се у разреде 2 и 1 акумулатора АСС уписује вредност 1, док у разредима 31 до 3 и разреду 0 акумулатора АСС остају непромењене вредности.

Инструкција реагује на прекид, али прекид не постоји, па се прелази на извршавање следеће инструкције.

Сумарно:

У фази *извршавање операције* 1. инструкције INTE главног програма са адресе 0100h у бит I програмске статусне речи PSW се уписује вредност 1 чиме се дозвољавају спољашњи маскирајући прекиди PRIRQM₃, PRIRQM₂ или PRIRQM₁. У току извршавања ове инструкције јавља се спољашњи маскирајући захтев за прекид PRIRQM₁. Инструкција INTE не реагује на прекиде, па се, иако постоји прекид PRIRQM₁, при чему прекид PRIRQM₁ није маскиран јер PSWI има вредност 1 и његов приоритет је виши од приоритета главног програма, прелази на следећу инструкцију. У фази *опслуживање прекида* 2. инструкције INC главног програма са адресе 0101h прелази се на извршавање 3. инструкције INS #3 прекидне рутине за PRIRQM₁ са адресе 1000h.

У фази *извршавање операције* 3. инструкције INS #3h прекидне рутине за PRIRQM₁ са адресе 1000h у флип-флоп PRINS се уписује вредност 1, чиме процесор генерише унутрашњи захтев за прекид због извршавања инструкције прекида INS. У фази *опслуживање прекида* инструкције INS прекид PRINS се прихвата и прелази на извршавање 4. инструкције POP прекидне рутине за PRINS са адресе 100Dh.

У току извршавања 5. инструкције AND #FFFFFF8h са адресе 100Eh прекидне рутине за PRINS јавља се спољашњи маскирајући прекид PRIRQM₂ који се не прихвата током извршавања прекидне рутине за PRINS јер PSWI има вредност 0, чиме су маскирани сви маскирајући захтеви за прекид. У фази *извршавање операције* 8. инструкције RTI прекидне рутине за PRINS са адресе 1013h програмски бројач PC се рестаурира на вредност 1001h, чиме се наставља извршавање прекидне рутине PRIRQM₁.

У фази *извршавање операције* 9. инструкције INTE прекидне рутине за PRIRQM₁ са адресе 1001h у бит I програмске статусне речи PSW се уписује вредност 1 чиме се дозвољавају спољашњи маскирајући прекиди PRIRQM₃, PRIRQM₂ или PRIRQM₁. Захтев за прекид PRIRQM₂ није више маскиран, али се не прихвата јер инструкција INTE не реагује на прекид. У фази *опслуживање прекида* 10. инструкције INC прекидне рутине за PRIRQM₁ прекид PRIRQM₂, чији је приоритет виши од приоритета прекидне рутине за PRIRQM₁, се прихвата и прелази на извршавање 11. инструкције LOAD 13h прекидне рутине за PRIRQM₂ са адресе 1007h.

У току извршавања 11. инструкције LOAD 13h прекидне рутине за PRIRQM₂ са адресе 1007h. јавља се спољашњи маскирајући прекид PRIRQM₃ који се не прихвата током извршавања прекидне рутине за PRIRQM₂ јер PSWI има вредност 0, чиме су маскирани сви маскирајући захтеви за прекид. У фази *извршавање операције* 14. инструкције RTI прекидне рутине за PRIRQM₂ са адресе 100Ch програмски бројач PC се рестаурира на вредност 1003h, чиме се наставља извршавање прекидне рутине PRIRQM₁.

У фази *опслуживање прекида* 15. инструкције INC прекидне рутине за PRIRQM₁ прекид PRIRQM₃, чији је приоритет виши од приоритета прекидне рутине за PRIRQM₁, се прихвата и прелази на извршавање 16. инструкције INC прекидне рутине за PRIRQM₃ са адресе 1014h.

У фази *извршавање операције* 17. инструкције RTI прекидне рутине за PRIRQM₃ са адресе 1015h програмски бројач PC се рестаурира на вредност 1004h, чиме се наставља извршавање прекидне рутине PRIRQM₁.

У фази *извршавање операције* 19. инструкције RTI прекидне рутине за PRIRQM₁ са адресе 1006h програмски бројач PC се рестаурира на вредност 102h, чиме се наставља извршавање главног програма.

Рб	Адреса	Инструкција	ACC	Стек	I	L	PRINS	PRIRQM ₃	PRIRQM ₂	PRIRQM ₁
1	0100	INTE	2	1	1	00	0	0	0	1
2	0101	INC	3	1	1	00	0	0	0	1
			3	2	0	01	0	0	0	0
3	1000	INS #3	3	2	0	01	1	0	0	0
			3	3	0	01	0	0	0	0
4	100D	POP	x..x001b	4	0	01	0	0	0	0
5	100E	AND #F..F8h	x..x000b	4	0	01	0	0	1	0
6	1010	OR 23h	x..x001b	4	0	01	0	0	1	0
7	1012	PUSH	x..x001b	5	0	01	0	0	1	0
8	1013	RTI	3	6	0	01	0	0	1	0
9	1001	INTE	3	6	1	01	0	0	1	0
10	1002	INC	4	6	1	01	0	0	1	0
			4	7	0	10	0	0	0	0
11	1007	LOAD 13h	6	7	0	10	0	1	0	0
12	1009	INC	7	7	0	10	0	1	0	0
13	100A	STORE 13h	7	7	0	10	0	1	0	0
14	100C	RTI	4	8	1	01	0	1	0	0
15	1003	INC	5	8	1	01	0	1	0	0
			5	9	0	11	0	0	0	0
16	1014	INC	6	9	0	11	0	0	0	0
17	1015	RTI	5	10	1	01	0	0	0	0
18	1004	STORE 15h	5	10	1	01	0	0	0	0
19	1006	RTI	3	11	1	00	0	0	0	0
20	0102	AND #6	2	11	1	00	0	0	0	0
21	0104	OR #6	6	11	1	00	0	0	0	0

Слика 7.д Табела са секвенцом инструкција које се извршавају

Ситуација 1		Ситуација 5		Ситуација 8	
SP	xxxx xxxx A	0000 0003 A	0000 0102 A-1	0000 0003 A	0000 0102 A-1
Ситуација 2		I=1, L _{1..0} =00 A-2	0000 0003 A-3	SP	I=1, L _{1..0} =00 A-2
	0000 0003 A	0000 1001 A-4	0000 1001 A-4		xxxx xxxx A-3
	0000 0102 A-1	I=0, L _{1..0} =01 A-5	I=0, L _{1..0} =01 A-5	Ситуација 9	
	I=1, L _{1..0} =00 A-2	xxxx xxxx A-6	xxxx xxxx A-6	0000 0003 A	0000 0102 A-1
SP	xxxx xxxx A-3	Ситуација 6		I=1, L _{1..0} =00 A-2	0000 0005 A-3
Ситуација 3		0000 0003 A	0000 0102 A-1	0000 1004 A-4	I=1, L _{1..0} =01 A-5
	0000 0102 A-1	0000 0102 A-1	I=1, L _{1..0} =00 A-2	I=1, L _{1..0} =01 A-5	xxxx xxxx A-6
	I=1, L _{1..0} =00 A-2	SP	xxxx xxxx A-3	SP	xxxx xxxx A-6
	0000 0003 A-3	Ситуација 7		0000 0003 A	Ситуација 10
	0000 1001 A-4	0000 0003 A	0000 0102 A-1	0000 0102 A-1	0000 0003 A
	I=0, L _{1..0} =01 A-5	0000 0102 A-1	I=1, L _{1..0} =00 A-2	I=1, L _{1..0} =00 A-2	0000 0102 A-1
SP	xxxx xxxx A-6	0000 0004 A-3	0000 0004 A-3	SP	I=1, L _{1..0} =00 A-2
Ситуација 4		0000 1003 A-4	0000 1003 A-4		xxxx xxxx A-3
	0000 0003 A	I=1, L _{1..0} =01 A-5	I=1, L _{1..0} =01 A-5	Ситуација 11	
	0000 0102 A-1	SP	xxxx xxxx A-6	SP	xxxx xxxx A
	I=1, L _{1..0} =00 A-2	Ситуација 11			
	0000 0003 A-3				
	0000 1001 A-4				
SP	xxxx xxxx A-5				

Слика 7.ђ. Изглед стека у току извршавања секвенце инструкција

д) Секвенца адреса инструкција које се редом извршавају, почев од инструкције INTE са адресе 0100h главног програма и закључно са инструкцијом OR #6 са адресе 0104h главног програма, приказана је на слици 7.е, карактеристичне ситуације на стеку су дате на слици 7.ж, а објашњења у даљем тексту. Сlike 7.е и 7.ж имају исту структуру и садрже исти тип информација као слике 7.д и 7.ђ, респективно. Почетна ситуација на стеку је дата као Ситуација 1 (слика 7.ж).

Секвенца првих пет инструкција на слици 7.е је идентична као и на слици 7.д. Разлике се јављају почев од шесте инструкције.

...

Инструкција 6: Почетне вредности регистара и индикатора су: PC=1010h, ACC=x...x000b, I=0, L=01.

Ради се о инструкцији OR 23h са адресе 1010h, која реализује логичку ИЛИ операцију садржаја акумулатора ACC који износи x...x000b и садржаја меморијске локације 23h који износи 0...0010b (2h) и резултат x...x010b уписује у акумулатор ACC. Резултат ове операције је да се у разред 2, 1 и 0 регистра ACC уписује вредност 010b, док се разреди 31 до 3 акумулатора ACC, у којима се налазе вредности које нису битне, не мењају. Треба уочити да је у акумулатору ACC сада садржај који одговара садржају регистра PSW скинутог са стека, али тако модификован да се у разредима 2, 1 и 0 акумулатора ACC, који одговарају битовима I и L_{1..0} регистра PSW, уместо вредности 001b налази вредност 010b.

Инструкција реагује на прекид и спољашњи прекид $PRIRQM_2$ постоји, али су сви спољашњи маскирајући прекиди маскирани вредношћу 0 бита I регистра PSW, па се прелази на извршавање следеће инструкције.

Инструкција 7: Почетне вредности регистара и индикатора су: PC=1012h, ACC=x...x010b, I=0, L=01.

Ради се о инструкцији PUSH са адресе 1012h, која садржај регистра ACC ставља на стек. Садржај акумулатора ACC који се ставља на стек представља модификовани садржај програмске статусне речи PSW који је сачуван на стеку у оквиру фазе *опслуживање прекида* инструкције INS #3 са адресе 1000h којом се прешло на ову прекидну рутину, па се у разредима 2, 1 и 0 акумулатора ACC, који одговарају битовима I и L_{1..0} регистра PSW, уместо вредности 001b налази вредност 010b. На стеку је Ситуација 5 (слика 7.ж).

Инструкција реагује на прекид и спољашњи прекид $PRIRQM_2$ постоји, али су сви спољашњи маскирајући прекиди маскирани вредношћу 0 бита I регистра PSW, па се прелази на извршавање следеће инструкције.

Инструкција 8: Почетне вредности регистара и индикатора су: PC=1013h, ACC=x...x001b, I=0, L=01.

Ради се о инструкцији RTI са адресе 1013h, која са стека скида контекст који је сачуван на стеку приликом преласка на ову прекидну рутину и уписује у програмску статусну реч PSW, програмски бројач PC и у акумулатор ACC. У регистру PSW се поново уписује у бит I вредност 0 док у битове L вредност 10b, регистар PC добија вредност 1001h и акумулатор ACC вредност 3. Треба уочити да се у битове L уписује не вредност 01b која је сачувана на стеку приликом преласка на прекидну рутину већ модификована вредност 10b. Као резултат у наставку извршавања прекидна рутина $PRIRQM_1$ ће имати ниво приоритета 10b уместо 01b. На стеку је Ситуација 6 (слика 7.ж).

Инструкција не реагује на прекид, а и спољашњи маскирајући прекид $PRIRQM_2$ који постоји је маскиран вредношћу 0 бита I регистра PSW, па се прелази на извршавање следеће инструкције.

Инструкција 9: Почетне вредности регистара и индикатора су: PC=1001h, ACC=3, I=0, L=10.

Ради се о инструкцији INTE са адресе 1001h, која у бит I програмске статусне речи PSW уписује вредност 1 чиме се дозвољавају спољашњи маскирајући прекиди.

Инструкција не реагује на прекид, па се, мада постоји прекид $PRIRQM_2$, прелази на извршавање следеће инструкције.

Инструкција 10: Почетне вредности регистара и индикатора су: PC=1002h, ACC=3, I=1, L=10.

Ради се о инструкцији INC са адресе 1002h која увећава садржај акумулатора за један, па садржај акумулатора ACC постаје 0000 0004h.

Инструкција реагује на прекид. Прекид $PRIRQM_2$ постоји и није маскиран јер се у биту I програмске статусне речи PSW налази вредност 1. Међутим, како је његов ниво приоритета $prl_{1..0}$, који износи 10b, једнак нивоу приоритета прекидне рутине периферије PER1 који је у прекидној рутини PRINS модификован са 01b на 10b, сигнал прихваћеног спољашњег маскирајућег прекида $printg$ има вредност 0. Због тога се, иако прекид $PRIRQM_2$ постоји, прелази на извршавање следеће инструкције.

Инструкција 11: Почетне вредности регистара и индикатора су: PC=1003h, ACC=4, I=1, L=10.

Ради се о инструкцији INC са адресе 1003h која увећава садржај акумулатора за један, па садржај акумулатора ACC постаје 0000 0005h. Током извршавања инструкције јавља се спољашњи маскирајући прекид на линији IRQM₃, па се флип-флоп PRIRQM₃ поставља на 1.

Инструкција реагује на прекид и прекиди PRIRQM₃ и PRIRQM₂ постоје. На прекиде се реагује по опадајућим приоритетима. Унутрашњег прекида PRINS, који представља прекид највишег приоритета, нема. У оквиру спољашњих маскирајућих прекида, чији је приоритет нижи од приоритета прекида PRINS, највиши ниво приоритета има прекид PRIRQM₃, затим следи прекид PRIRQM₂ и на крају долази са најнижим нивоом приоритета прекид PRIRQM₁. Због тога се прихвата спољашњи маскирајући прекид PRIRQM₃. Како је његов ниво приоритета prl_{1..0}, који износи 11b, виши од модификованог нивоа приоритета прекидне рутине периферије PER1, који износи 10b, и како маскирајући прекиди нису маскирани, јер бит PSWI има вредност 1, сигнал прихваћеног спољашњег маскирајућег прекида printr има вредност 1. Стога се у оквиру фазе *опслуживање прекида* најпре на стеку чува контекст процесора програма чије се извршавање прекида, а затим се формира контекст процесора прекидне рутине на чије се извршавање прелази.

У корацима чувања контекста процесора програма чије се извршавање прекида на стеку се чува прво вредност акумулатора ACC, затим програмски бројач PC и на крају програмска статусна реч PSW. Акумулатор ACC има вредност 5h. Програмски бројач PC има вредност 1004h, која показује на прву следећу инструкцију програма који се прекида. Код програмске статусне речи PSW која се чува на стеку бит I има вредност 1, бити L_{1..0} имају вредност 10b, док вредности осталих битова нису битне.

У корацима формирања контекста процесора прекидне рутине на чије се извршавање прелази потребно је поставити нове вредности у програмску статусну реч PSW и програмски бројач PC. У бит I регистра PSW се уписује вредност 0, док се у битове L_{1..0} регистра PSW уписује вредност 11b. Треба уочити да се у битове L_{1..0} регистра PSW уписује нова вредност само приликом преласка на неку од прекидних рутина за спољашње маскирајуће прекиде PRIRQM₃, PRIRQM₂ или PRIRQM₀.

Улази у IV табелу нису фиксни, па процесор, пошто се ради о маскирајућем прекиду PRIRQM₃, најпре по линија INTA₃ шаље контролеру периферије PER3 сигнал потврде и флип-флоп PRIRQM₃ брише на 0, а затим од контролера периферије PER3 добија број улаза чија је вредност 2. Адреса улаза 2 у IV табели је 2 (2 = 0 + 2·1), па процесор најпре са адресе 0000 0002h чита вредност 0000 1014h, која представља адресу прекидне рутине, а затим ту вредност уписује у програмски бројач PC и прелази на фазу *читање инструкције* прве инструкције прекидне рутине. На стеку је Ситуација 7 (слика 7.ж).

Инструкција 12: Почетне вредности регистара и индикатора су: PC=1014h, ACC=5h, I=0, L=11.

Ради се о инструкцији INC са адресе 1014h, која увећава садржај акумулатора за један, па садржај акумулатора ACC постаје 0000 0006h.

Инструкција реагује на прекид и спољашњи прекид PRIRQM₂ постоји, али су сви спољашњи маскирајући прекиди маскирани вредношћу 0 бита I регистра PSW, па се прелази на извршавање следеће инструкције.

Инструкција 13: Почетне вредности регистара и индикатора су: PC=1015h, ACC=6h, I=0, L=11.

Ради се о инструкцији RTI са адресе 1015h, која са стека скида контекст који је сачуван на стеку приликом преласка на ову прекидну рутину и уписује у програмску статусну реч PSW, програмски бројач PC и у акумулатор ACC. У регистру PSW се уписује у бит I вредност 1 и у битове L вредност 10b, регистар PC добија вредност 1004h и акумулатор ACC вредност 5h. На стеку је Ситуација 8 (слика 7.ж).

Инструкција не реагује на прекид, па се, мада постоји прекид PRIRQM₂, прелази на извршавање следеће инструкције.

Инструкција 14: Почетне вредности регистара и индикатора су: PC=1004h, ACC=5h, I=1, L=10.

Ради се о инструкцији STORE 15h са адресе 1004h, која у меморијску локацију на адреси 15h уписује садржај акумулатора чија је вредност 0005h.

Инструкција реагује на прекид. Прекид PRIRQM₂ постоји и није маскиран јер се у биту I програмске статусне речи PSW налази вредност 1. Међутим, како је његов ниво приоритета prl_{1..0}, који износи 10b, једнак нивоу приоритета прекидне рутине периферије PER1 који је у прекидној рутини PRINS модификован са 01b на 10b, сигнал прихваћеног спољашњег маскирајућег прекида printr има вредност 0. Због тога се, иако прекид PRIRQM₂ постоји, прелази на извршавање следеће инструкције.

Инструкција 15: Почетне вредности регистара и индикатора су: PC=1006h, ACC=5, I=1, L=10.

Ради се о инструкцији RTI са адресе 1006h, која са стека скида контекст који је сачуван на стеку приликом преласка на ову прекидну рутину и уписује у програмску статусну реч PSW, програмски бројач PC и у акумулатор ACC. У регистру PSW се поново уписује у бит I вредност 1 и у битове L вредност 00b, регистар PC добија вредност 0102h и акумулатор ACC вредност 3. На стеку је Ситуација 9 (слика 7.ж).

Инструкција не реагује на прекид, па се, мада постоји прекид PRIRQM₂, прелази на извршавање следеће инструкције.

Инструкција 16: Почетне вредности регистара и индикатора су: PC=102h, ACC=3, I=1, L=00.

Ради се о инструкцији AND #6h са адресе 102h, која реализује логичку И операцију садржаја акумулатора ACC, који износи 0...00011b, и непосредне величине, која износи 0...0110b (6h), и резултат, који износи 0...0010b, уписује у акумулатор ACC. Треба уочити да непосредна величина има вредност 1 на битовима 2 и 1 и вредност 0 на битовима 31 до 3 и биту 0. Због тога је резултат логичке И операције такав да у разредима 2 и 1 акумулатора ACC остају непромењене вредности 0 и 1, респективно, док се у разреде 31 до 3 и разред 0 акумулатора ACC уписује вредност 0.

Инструкција реагује на прекид и прекид PRIRQM₂ постоји. На прекиде се реагује по опадајућим приоритетима. Унутрашњег прекида PRINS, који представља прекид највишег приоритета, нема. У оквиру спољашњих маскирајућих прекида, чији је приоритет нижи од приоритета прекида PRINS, највиши ниво приоритета има прекид PRIRQM₃, затим следи прекид PRIRQM₂ и на крају долази са најнижим нивоом приоритета прекид PRIRQM₁. Спољашњег маскирајућег прекида PRIRQM₃ нема, па се прихвата прекид PRIRQM₂. Како је његов ниво приоритета prl_{1..0}, који износи 10b, виши од нивоа приоритета главног програма, који износи 00b, и како маскирајући прекиди нису маскирани, јер бит PSWI има

вредност 1, сигнал прихваћеног спољашњег маскирајућег прекида `printr` има вредност 1. Стога се у оквиру фазе *опслуживање прекида* најпре на стеку чува контекст процесора програма чије се извршавање прекида, а затим се формира контекст процесора прекидне рутине на чије се извршавање прелази.

У корацима чувања контекста процесора програма чије се извршавање прекида на стеку се чува прво вредност акумулатора ACC, затим програмски бројач PC и на крају програмска статусна реч PSW. Акумулатор ACC има вредност 2. Програмски бројач PC има вредност 0104h, која показује на прву следећу инструкцију програма који се прекида. Код програмске статусне речи PSW која се чува на стеку бит I има вредност 1, бити L_{1..0} имају вредност 00b, док вредности осталих битова нису битне.

У корацима формирања контекста процесора прекидне рутине на чије се извршавање прелази потребно је поставити нове вредности у програмску статусну реч PSW и програмски бројач PC. У бит I регистра PSW се уписује вредност 0, док се у битове L_{1..0} регистра PSW уписује вредност 10b. Треба уочити да се у битове L_{1..0} регистра PSW уписује нова вредност само приликом преласка на неку од прекидних рутина за спољашње маскирајуће прекиде PRIRQM₃, PRIRQM₂ или PRIRQM₁.

Улази у IV табелу нису фиксни, па процесор, пошто се ради о маскирајућем прекиду PRIRQM₂, најпре по линија INTA₂ шаље контролеру периферије PER2 сигнал потврде и флип-флоп PRIRQM₂ брише на 0, а затим од контролера периферије PER2 добија број улаза чија је вредност 1. Адреса улаза 1 у IV табели је 1 ($1 = 0 + 1 \cdot 1$), па процесор најпре са адресе 0000 0001h чита вредност 0000 1007h, која представља адресу прекидне рутине, а затим ту вредност уписује у програмски бројач PC и прелази на фазу *читање инструкције* прве инструкције прекидне рутине. На стеку је Ситуација 10 (слика 7.ж).

Инструкција 17: Почетне вредности регистара и индикатора су: PC=1007h, ACC=2, I=0, L=10.

Ради се о инструкцији LOAD 13h са адресе 1007h, која у акумулатор ACC уписује садржај са адресе 13h, а тамо се налази 6h.

Инструкција реагује на прекид, али прекид не постоји, па се прелази на извршавање следеће инструкције.

Инструкција 18: Почетне вредности регистара и индикатора су: PC=1009h, ACC=6, I=0, L=10.

Ради се о инструкцији INC са адресе 1009h, која увећава садржај акумулатора ACC за један, па садржај акумулатора ACC постаје 0000 0007h.

Инструкција реагује на прекид, али прекид не постоји, па се прелази на извршавање следеће инструкције.

Инструкција 19: Почетне вредности регистара и индикатора су: PC=100Ah, ACC=7h, I=0, L=10.

Ради се о инструкцији STORE 13h са адресе 100Ah, која у меморијску локацију на адреси 13h уписује садржај акумулатора чија је вредност 0007h.

Инструкција реагује на прекид, али прекид не постоји, па се прелази на извршавање следеће инструкције.

Инструкција 20: Почетне вредности регистара и индикатора су: PC=100Ch, ACC=7h, I=0, L=10.

Ради се о инструкцији RTI са адресе 100Ch, која са стека скида контекст који је сачуван на стеку приликом преласка на ову прекидну рутину и уписује у програмску статусну реч

PSW, програмски бројач PC и у акумулатор ACC. У регистру PSW се уписује у бит I вредност 1 и у битове L вредност 00b, регистар PC добија вредност 1000h и акумулатор ACC вредност 2. На стеку је Ситуација 11 (слика 7.ж).

Инструкција не реагује на прекид, па се прелази на извршавање следеће инструкције.

Инструкција 21: Почетне вредности регистара и индикатора су: PC=104h, ACC=2, I=1, L=00.

Ради се о инструкцији OR #6h са адресе 104h, која реализује логичку ИЛИ операцију садржаја акумулатора ACC, који износи 0...00010b, и непосредне величине, која износи 0...0110b (6h), и резултат, који износи 0...0110b, уписује у акумулатор ACC.

Треба уочити да непосредна величина има вредност 1 на битовима 2 и 1 и вредност 0 на битовима 31 до 3 и биту 0. Због тога је резултат логичке И операције такав да се у разреду 2 и 1 акумулатора ACC уписује вредност 1, док у разредима 31 до 3 и разреду 0 акумулатора ACC остају непромењене вредности.

Инструкција реагује на прекид, али прекид не постоји, па се прелази на извршавање следеће инструкције.

Сумарно:

У фази *извршавање операције 1.* инструкције INTE главног програма са адресе 0100h у бит I програмске статусне речи PSW се уписује вредност 1 чиме се дозвољавају спољашњи маскирајући прекиди PRIRQM₃, PRIRQM₂ или PRIRQM₁. У току извршавања ове инструкције јавља се спољашњи маскирајући захтев за прекид PRIRQM₁. Инструкција INTE не реагује на прекиде, па се, иако постоји прекид PRIRQM₁, при чему прекид PRIRQM₁ није маскиран јер PSWI има вредност 1 и његов приоритет је виши од приоритета главног програма, прелази на следећу инструкцију. У фази *опслуживање прекида 2.* инструкције INC главног програма са адресе 0101h прелази се на извршавање 3. инструкције INS #3 прекидне рутине за PRIRQM₁ са адресе 1000h.

У фази *извршавање операције 3.* инструкције INS #3h прекидне рутине за PRIRQM₁ са адресе 1000h у флип-флоп PRINS се уписује вредност 1, чиме процесор генерише унутрашњи захтев за прекид због извршавања инструкције прекида INS. У фази *опслуживање прекида* инструкције INS прекид PRINS се прихвата и прелази на извршавање 4. инструкције POP прекидне рутине за PRINS са адресе 100Dh.

У току извршавања 5. инструкције AND #FFFFFFF8h са адресе 100Eh прекидне рутине за PRINS јавља се спољашњи маскирајући прекид PRIRQM₂ који се не прихвата током извршавања прекидне рутине за PRINS јер PSWI има вредност 0, чиме су маскирани сви маскирајући захтеви за прекид. У фази *извршавање операције 8.* инструкције RTI прекидне рутине за PRINS са адресе 1013h програмски бројач PC се рестаурира на вредност 1001h, чиме се наставља извршавање прекидне рутине PRIRQM₁. Треба уочити да је као резултат извршавања прекидне рутине PRINS, садржај на врху стека, који представља садржај регистра PSW сачуван приликом преласка из прекидне рутине PRIRQM₁ у прекидну рутину PRINS, модификован, па се по повратку из прекидне рутине PRINS у прекидну рутину PRIRQM₁, извршавање прекидне рутине PRIRQM₁ наставља не са оригиналним нивоом приоритета 01b, већ са модификованим нивоом приоритета 10b.

У фази *извршавање операције 9.* инструкције INTE прекидне рутине за PRIRQM₁ са адресе 1001h у бит I програмске статусне речи PSW се уписује вредност 1 чиме се дозвољавају спољашњи маскирајући прекиди PRIRQM₃, PRIRQM₂ или PRIRQM₁. Захтев за прекид PRIRQM₂ није више маскиран, али се не прихвата јер инструкција INTE не реагује на прекид. У току извршавања 10. инструкције INC прекидне рутине за PRIRQM₁

са адресе 1002h, прекид PRIRQM₂ се не прихвата, јер је његов ниво приоритета, који износи 10b, једнак модификованом нивоу приоритета прекидне рутине за PRIRQM₁, који такође износи 10b. У току извршавања 11. инструкције INC прекидне рутине за PRIRQM₁ са адресе 1003h, јавља се спољашњи маскирајући прекид PRIRQM₃. У фази *ослуживање прекида* инструкције INC, прекид PRIRQM₃, чији је приоритет, који износи 11b, виши од модификованог нивоа приоритета прекидне рутине за PRIRQM₁, који износи 10b, се прихвата и прелази на извршавање 12. инструкције INC прекидне рутине за PRIRQM₃ са адресе 1014h.

У фази *извршавање операције* 13. инструкције RTI прекидне рутине за PRIRQM₃ са адресе 1015h програмски бројач PC се рестаурира на вредност 1004h, чиме се наставља извршавање прекидне рутине PRIRQM₁.

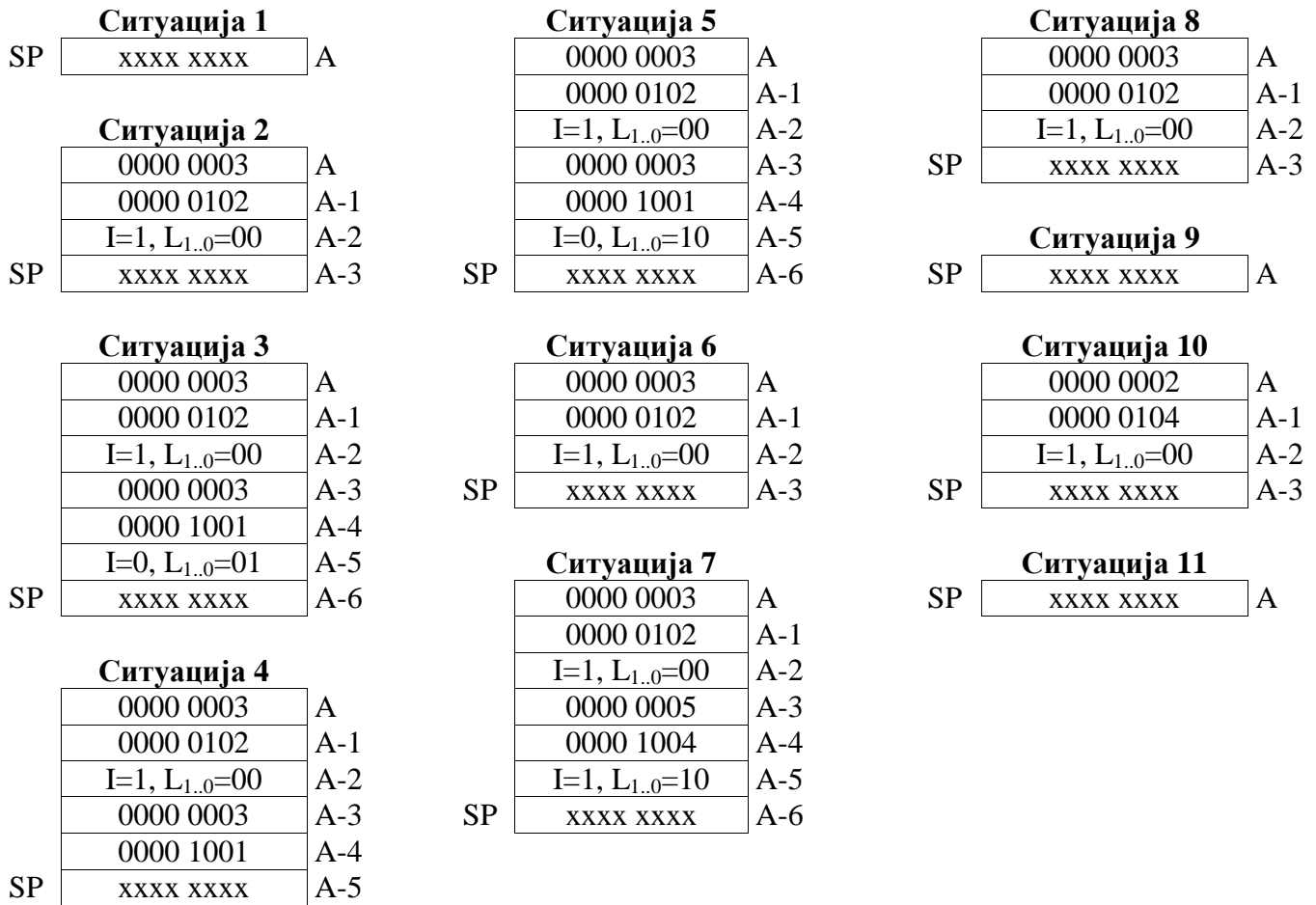
У фази *извршавање операције* 15. инструкције RTI прекидне рутине за PRIRQM₁ са адресе 1006h, програмски бројач PC се рестаурира на вредност 102h, чиме се наставља извршавање главног програма.

У фази *ослуживање прекида* 16. инструкције AND #6h главног програма са адресе адресе 102h, прекид PRIRQM₂, чији је приоритет, који износи 10b, виши од приоритета главног програма, који износи 00b, се прихвата и прелази на извршавање 17. инструкције LOAD 13h прекидне рутине за PRIRQM₂ са адресе 1007h.

У фази *извршавање операције* 20. инструкције RTI прекидне рутине за PRIRQM₂ са адресе 100Ch, програмски бројач PC се рестаурира на вредност 104h, чиме се наставља извршавање главног програма.

P6	Адреса	Инструкција	ACC	Стек	I	L	PRINS	PRIRQM ₃	PRIRQM ₂	PRIRQM ₁
1	0100	INTE	2	1	1	00	0	0	0	1
2	0101	INC	3	1	1	00	0	0	0	1
			3	2	0	01	0	0	0	0
3	1000	INS #3	3	2	0	01	1	0	0	0
			3	3	0	01	0	0	0	0
4	100D	POP	x...x001b	4	0	01	0	0	0	0
5	100E	AND #F..F8h	x...x000b	4	0	01	0	0	1	0
6	1010	OR 23h	x...x010b	4	0	01	0	0	1	0
7	1012	PUSH	x...x010b	5	0	01	0	0	1	0
8	1013	RTI	3	6	0	10	0	0	1	0
9	1001	INTE	3	6	1	10	0	0	1	0
10	1002	INC	4	6	1	10	0	0	1	0
11	1003	INC	5	6	1	10	0	1	1	0
			5	7	0	11	0	0	1	0
12	1014	INC	6	7	0	11	0	0	1	0
13	1015	RTI	5	8	1	10	0	0	1	0
14	1004	STORE 15h	5	8	1	10	0	0	1	0
15	1006	RTI	3	9	1	00	0	0	1	0
16	0102	AND #6	2	9	1	00	0	0	1	0
			2	10	0	10	0	0	0	0
17	1007	LOAD 13h	6	10	0	10	0	0	0	0
18	1009	INC	7	10	0	10	0	0	0	0
19	100A	STORE 13h	7	10	0	10	0	0	0	0
20	100C	RTI	2	11	1	00	0	0	0	0
21	0104	OR #6	6	11	1	00	0	0	0	0

Слика 7.е Табела са секвенцом инструкција које се извршавају



Слика 7.ж. Изглед стека у току извршавања секвенце инструкција

1.8 ЗАДАТАК 8

Оперативна меморија неког рачунара је капацитета 64 КВ, а ширина речи меморије је 8 бита. Подаци ширине 16 бита се у меморију смештају у две суседне меморијске локације и то тако да се на нижој адреси налази нижих 8 бита, а на вишој адреси виших 8 бита податка. Механизам прекида је векторисан. Адресе прекидних рутина се чувају у табели адреса прекидних рутина (IV табела - *Interrupt Vector Table*), а почетна адреса IV табеле у регистру IVTP (*Interrupt Vector Table Pointer*).

Процесор може да генерише унутрашњи захтев за прекид због задатог режима рада прекид после сваке инструкције (прекид *trap*) коме је додељен 2 у IV табели. У процесору постоји регистар PSW (*Program Status Word*) са битом Т (*Trap enable*). Бит Т вредношћу 1 дозвољава режим рада прекид после сваке инструкције и поставља се на вредност 0 хардверски током извршавања фазе *опслуживање прекида*. Инструкције RTI, TRPE и TRPD не реагују на захтеве за прекид.

У оквиру извршавања фазе *опслуживање прекида* на стеку се хардверски чувају PSW и PC тим редом. Стек расте према нижим локацијама, а указивач на врх стека SP указује на задњу заузету локацију.

Процесор је двоадресни и све инструкције се извршавају над 8 битним величинама. Регистри опште намене (R_i) и регистар PSW су 8 битни.

Главни програм је дат на слици 8.а, *trap* прекидна рутина на слици 8.б, и део меморије почев од адресе 0 на слици 8.в.

адреса	инструкција
FF00h	TRPD
FF01h	MOV R0, #2
FF03h	TRPE
FF04h	DEC R0
FF05h	JNZ FF03h
FF08h	TRPD

Слика 8.а Део главног програма

адреса	инструкција
00A0h	POP R1
00A1h	PUSH R1
00A2h	RTI

Слика 8.б *Trap* прекидна рутина

адреса	садржај
0000h	00h
0001h	A0h
0002h	00h
0003h	0Ah
0004h	00h
0005h	A0h
0006h	00h
0007h	00h
0008h	0Ah
0009h	00h
000Ah	AAh
000Bh	0Ah
000Ch	A0h

Слика 8.в Део меморије почев од адресе 0

а) Уколико је првим бајтом инструкције дат код операције и начин адресирања, а другим и трећим адреса на коју се скаче, која се вредност налази на адреси FF07h?

б) Која се вредност налази у регистру IVTP (та вредност је већа или једнака од 0, а мања од 7)?

в) Написати секвенцу адреса наредби које се редом извршавају, почев од FF00h, закључно са FF08h.

г) Која се вредност налази у регистру R1 непосредно пре првог извршавања наредбе RTI на адреси 00A2h?

д) Написати секвенцу адреса наредби које се редом извршавају, почев од FF00h, све док су адресе познате, ако се уместо *trap* рутине са слике 8.б користи рутина са слике 8.г.

адреса	инструкција
00A0h	POP R1
00A1h	PUSH #2
00A3h	RTI

Слика 8.г Прекидна рутина

Решење:

а) Локације на адресама FF05h, FF06h и FF07h заузима инструкција JNZ FF03h. Како се у овој инструкцији користи меморијско директно адресирање, сигурно је да бар два бајта заузима запис саме адресе FF03h. То су свакако друга два бајта инструкције, на адресама FF06h и FF07h, јер први бајт сигурно садржи код операције. Како се 16 битни бројеви смештају тако да је на вишој адреси виши бајт, следи да је на адреси FF07h садржај FFh.

б) Како је *trap* прекиду фиксно додељен улаз 2, а адреса прекидне рутине је 00A0h, потребно је одредити почетак IV табеле према условима из задатка. Потребно је у датом сегменту меморије најпре пронаћи редом бајтове A0h и 00h, јер се 16 битне величине смештају у редоследу нижи па виши бајт. Тражене вредности налазе се на два места: почев од адресе 1 и почев од адресе 5. Како сваки улаз заузима 2 бајта, адреса почетка улаза у IV табелу одређује се према формули: $a = IVTP + 2 \cdot i$, где је i број улаза. Улаз 2 почиње са померајем 4 у односу на IVTP. Како важи да је $IVTP \geq 0$ и померај улаза 2 је 4, улаз 2 не може да буде на адреси 1 већ на адреси 5. Одавде следи да је IVTP једнако 1.

в) Пре почетка извршавања датог програмског сегмента важе следећи почетни услови: регистар PC има вредност FF00h, стек је празан, у програмској статусној речи бит T има вредност 0, а почетно стање регистара R0 и R1 није познато.

Инструкција 1: Почетне вредности регистара и индикатора су: PC=FF00h, R0=?, R1=?, T=?.

Ради се о инструкцији TRPD са адресе FF00h, која у бит T програмске статусне речи уписује вредност 0.

Инструкција не реагује на прекид и прелази се на извршавање следеће инструкције.

Инструкција 2: Почетне вредности регистара и индикатора су: PC=FF01h, R0=?, R1=?, T=0.

Ради се о инструкцији MOV R0, #2 са адресе FF01h, која у регистар R0 уписује вредност 2. Инструкција реагује на прекид, али бит T има вредност 0 и прекид *trap* не постоји, па се прелази на извршавање следеће инструкције.

Инструкција 3: Почетне вредности регистара и индикатора су: PC=FF03h, R0=2, R1=?, T=0.

Ради се о инструкцији TRPE са адресе FF03h, која у бит Т програмске статусне речи уписује вредност 1, чиме се генерише унутрашњи прекид *trap* због задатог режима рада прекид после сваке инструкције.

Инструкција не реагује на прекид и прелази се на извршавање следеће инструкције. Треба уочити да би инструкција TRPE могла да се реализује тако да реагује на прекид. У том случају би се прешло на извршавање фазе *опслуживање прекида* инструкције TRPE у оквиру које би се реализовао прелаз на прекидну рутину *trap*.

Инструкција 4: Почетне вредности регистра и индикатора су: PC=FF04h, R0=2, R1=?, T=1.

Ради се о инструкцији DEC R0 са адресе FF04h, која умањује садржај регистра R0 за 1, тако да регистар R0 добија вредност 1. Инструкција реагује на прекид. Пошто је бит Т постављен на вредност 1 прекид *trap* постоји, па се прелази се на фазу *опслуживања прекида*.

У фази *опслуживања прекида* прво се чува контекст процесора, а након тога се припрема нови контекст процесора. У корацима чувања контекста на стеку се чува програмска статусна реч и вредност програмског бројача. Код програмске статусне речи бит Т има вредност 1, а програмски бројач има вредност FF05h, која показује на прву следећу инструкцију програма. Приликом стављања програмског бројача на стек треба водити рачуна да виши бајт (FFh) иде на виши адресу, а да нижи бајт (05h).

Након корака чувања контекста прелази се на припрему новог контекста процесора. Приликом формирања новог контекста потребно је поставити нове вредности у програмску статусну реч и програмски бројач. У програмској статусној речи се модификују бити задужени за прекид (I, T, као и одређен број L бита). Пошто се у наведеном примеру користи само бит Т у њега се уписује вредност 0. Постављање програмског бројача се обавља тако што се из одговарајућег улаза табеле прекидних рутина прочита вредност која одговара датом прекиду. Пошто се овде ради о унутрашњем прекиду после сваке инструкције број улаза је фиксан и износи 2. Након одређивања броја улаза приступа се адресама које одговарају датом улазу како би се прочитала адреса од које почиње прекидна рутина. Број сукцесивних меморијских локација којима се приступа је одређен односом ширине адресе и адресибилне јединице. Процесор са адреса 5 и 6 чита адресу прве инструкције прекидне рутине ($5 = 1 + 2 * 2$). Прочитане су вредности A0h и 00h, што одговара адреси 00A0h. Ова вредност се уписује у регистар PC и прелази на фазу *читање инструкције* прве инструкције прекидне рутине *trap*.

Инструкција 5: Почетне вредности регистра и индикатора су: PC=00A0h, R0=1, R1=?, T=0.

Ради се о инструкцији POP R1 са адресе 00A0h, која са стека скида један бајт и уписује га у регистар R1. Прочитана је вредност 5. Ова вредност одговара нижем бајту повратне адресе. Проверава се да ли има прекида, констатује се да нема и прелази се на читавање следеће инструкције.

Инструкција 6: Почетне вредности регистра и индикатора су: PC=00A1h, R0=1, R1=5, T=0.

Ради се о инструкцији PUSH R1 са адресе 00A1h, која на стек ставља један бајт из регистра R1. Уписана је вредност 5. Овде треба водити рачуна о томе да ће се ова вредност приликом повратка из прекидне рутине тумачити као нижи бајт повратне адресе. Грешка која би се овде могла појавити би се огледала у томе да би програм могао да се врати да се извршава почев од неке адресе, која не одговара адреси где је програм прекинут. Проверава се да ли има прекида, констатује се да нема и прелази се на читавање следеће инструкције.

Инструкција 7: Почетне вредности регистара и индикатора су: PC=00A2h, R0=1, R1=5, T=0.

Ради се о инструкцији RTI са адресе 00A2h, која са стека скида контекст који се уписује у програмски бројач, као и у програмску статусну реч. Подаци се са стека скидају у обрнутом редоследу од оног у коме су постављани. Прво се скида програмски бројач, и то бајт по бајт, зато што је адресибилна јединица два бајта а програмски бројач два бајта. Након читавања повратне адресе (FF05h) читава се програмска статусна реч. Вредности које се мењају се разликују од оних које су биле у прекидној рутини. Бит Т добија вредност 1.

Не проверава се да ли је било прекида или не јер инструкција не реагује на прекид и прелази се на извршавање следеће инструкције. Ова инструкција би могла да се реализује тако да реагује на прекид, али треба водити рачуна да инструкција реагује на све прекиде осим на прекид после сваке инструкције. Ово је потребно урадити да се не би упало у бесконачну петљу извршавања ове прекидне рутине.

Инструкција 8: Почетне вредности регистара и индикатора су: PC=FF05h, R0=1, R1=5, T=1.

Ради се о инструкцији JNZ FF03h са адресе FF05h, која уколико је бит Z програмске статусне речи вио 0 прелази на адресу FF03h. Пошто бит Z има вредност 0 у програмски бројач се уписује вредност FF03h.

Инструкција реагује на прекид и прелази се на кораке фазе *опслуживања прекида*. Ови кораци су исти као код инструкције 4. На стеку се чува програмска статусна реч код које је Т постављено не 1, и програмски бројач са верношћу FF03h. У бит Т се уписује вредност 0, а у програмски бројач вредност 00A0h. Прелази се на прву инструкцију прекидне рутине.

Инструкција 9: Почетне вредности регистара и индикатора су: PC=00A0h, R0=1, R1=5, T=0.

Ради се о инструкцији POP R1 са адресе 00A0h, са стека скида један бајт (3) и уписује га у регистар R1.

Проверава се да ли има прекида, констатује се да нема и прелази се на читавање следеће инструкције.

Инструкција 10: Почетне вредности регистара и индикатора су: PC=00A1h, R0=1, R1=3, T=0.

Ради се о инструкцији PUSH R1 са адресе 00A1h, која на стек враћа бајт из регистра R1 (3).

Проверава се да ли има прекида, констатује се да нема и прелази се на читавање следеће инструкције.

Инструкција 11: Почетне вредности регистара и индикатора су: PC=00A2h, R0=1, R1=3, T=0.

Ради се о инструкцији RTI са адресе 00A2h, која са враћа на инструкцију на адреси FF03h.

Не проверава се да ли је било прекида или не.

Инструкција 12: Почетне вредности регистара и индикатора су: PC=FF03h, R0=1, R1=3, T=1.

Ради се о инструкцији TRPE са адресе FF03h, која у бит Т програмске статусне речи уписује вредност 1.

Не проверава се да ли је било прекида или не.

Инструкција 13: Почетне вредности регистара и индикатора су: PC=FF04h, R0=1, R1=3, T=1.

Ради се о инструкцији DEC R0 са адресе FF04h, која умањује садржај регистра R0 за 1, тако да регистар R0 добија вредност 0, а индикатор Z вредност 1.

Инструкција реагује на прекид и прелази се на кораке опслуживања прекида. Ови кораци су исти као код инструкције 4. На стеку се чува програмска статусна реч код које је Т постављено на 1, и програмски бројач са вредношћу FF05h. У бит Т се уписује вредност 0, а у програмски бројач вредност 00A0h. Прелази се на прву инструкцију прекидне рутине.

Инструкција 14: Почетне вредности регистара и индикатора су: PC=00A0h, R0=0, R1=3, T=0.

Ради се о инструкцији POP R1 са адресе 00A0h, са стека скида један бајт (5) и уписује га у регистар R1.

Проверава се да ли има прекида, констатује се да нема и прелази се на читавање следеће инструкције.

Инструкција 15: Почетне вредности регистара и индикатора су: PC=00A1h, R0=0, R1=5, T=0.

Ради се о инструкцији PUSH R1 са адресе 00A1h, која на стек враћа бајт из регистра R1 (5).

Проверава се да ли има прекида, констатује се да нема и прелази се на читавање следеће инструкције.

Инструкција 16: Почетне вредности регистара и индикатора су: PC=00A2h, R0=0, R1=5, T=0.

Ради се о инструкцији RTI са адресе 00A2h, која са враћа на инструкцију на адреси FF05h.

Не проверава се да ли је било прекида или не.

Инструкција 17: Почетне вредности регистара и индикатора су: PC=FF05h, R0=0, R1=5, T=1.

Ради се о инструкцији JNZ FF03h са адресе FF05h. Пошто бит Z има вредност 1, не скаче се, а у програмском бројачу остаје адресе прве следеће инструкције FF08h.

Инструкција реагује на прекид и прелази се на кораке опслуживања прекида. Ови кораци су исти као код инструкције 4. На стеку се чува програмска статусна реч код које је Т постављено на 1, и програмски бројач са вредношћу FF08h. У бит Т се уписује вредност 0, а у програмски бројач вредност 00A0h. Прелази се на прву инструкцију прекидне рутине.

Инструкција 18: Почетне вредности регистара и индикатора су: PC=00A0h, R0=0, R1=5, T=0.

Ради се о инструкцији POP R1 са адресе 00A0h, са стека скида један бајт (8) и уписује га у регистар R1.

Проверава се да ли има прекида, констатује се да нема и прелази се на читавање следеће инструкције.

Инструкција 19: Почетне вредности регистара и индикатора су: PC=00A1h, R0=0, R1=8, T=0.

Ради се о инструкцији PUSH R1 са адресе 00A1h, која на стек враћа бајт из регистра R1 (8).

Проверава се да ли има прекида, констатује се да нема и прелази се на читавање следеће инструкције.

Инструкција 20: Почетне вредности регистара и индикатора су: PC=00A2h, R0=0, R1=8, T=0.

Ради се о инструкцији RTI са адресе 00A2h, која са враћа на инструкцију на адреси FF08h. Не проверава се да ли је било прекида или не.

Инструкција 21: Почетне вредности регистара и индикатора су: PC=FF08h, R0=0, R1=8, T=1.

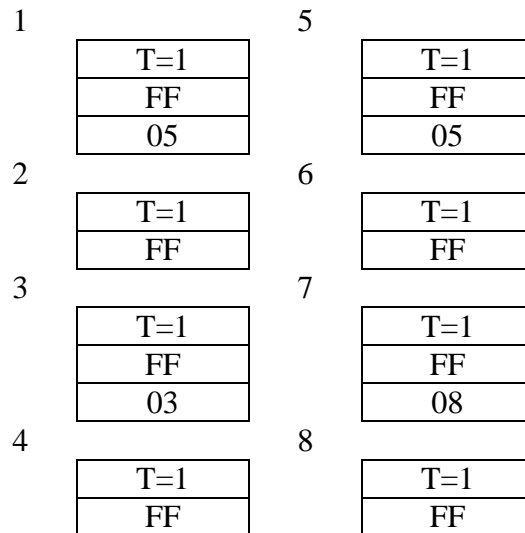
Ради се о инструкцији TRPD са адресе FF08h, која у бит Т програмске статусне речи уписује вредност 0.

Инструкција не реагује на прекид и прелази се на извршавање следеће инструкције.

Овим се завршава дати програмски сегмент. Секвенца адреса инструкција је дата на слици 8.д (све вредности су хексадецималне).

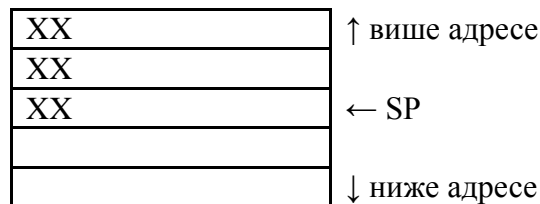
P6	Адреса	Инструкција	R0	R1	Стек	Z	T
0	-	-	?	?	-	?	0
1	FF00	TRPD	?	?	-	?	0
2	FF01	MOV R0, #2	2	?	-	0	0
3	FF03	TRPE	2	?	-	0	1
4	FF04	DEC R0	1	?	-	0	1
			1	?	1	0	0
5	00A0	POP R1	1	5	2	0	0
6	00A1	PUSH R1	1	5	1	0	0
7	00A2	RTI	1	5	-	0	1
8	FF05	JNZ FF03h	1	5	-	0	1
			1	5	3	0	0
9	00A0	POP R1	1	3	4	0	0
10	00A1	PUSH R1	1	3	3	0	0
11	00A2	RTI	1	3	-	0	1
12	FF03	TRPE	1	3	-	0	1
13	FF04	DEC R0	0	3	-	1	1
			0	3	5	1	0
14	00A0	POP R1	0	5	6	0	0
15	00A1	PUSH R1	0	5	5	0	0
16	00A2	RTI	0	5	-	1	1
17	FF05	JNZ FF03h	0	5	-	1	1
			0	5	7	1	0
18	00A0	POP R1	0	8	8	0	0
19	00A1	PUSH R1	0	8	7	0	0
20	00A2	RTI	0	8	-	1	1
21	FF08	TRPD	0	8	-	1	0

Слика 8.д. Табела са секвенцом инструкција које се извршавају

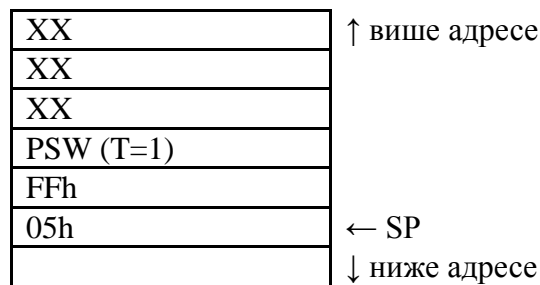


Слика 8.ђ. Изглед стека у току извршавања секвенце инструкција

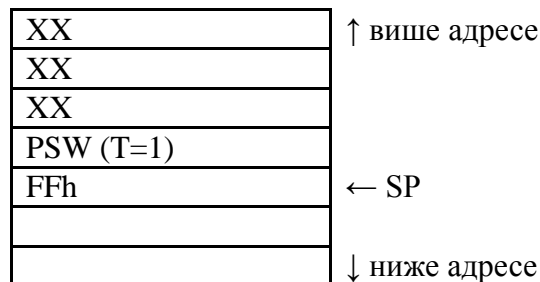
г) Изглед стека на почетку главног програма представљен је на слици 8.е. Стек расте према нижим локацијама, а претпостављено је да SP показује на последњу заузету локацију. Са XX представљен је садржај који се налазио на стеку пре почетка извршавања инструкције FF00h главног програма, тај садржај нам је непознат и није од значаја за задатак. Пре преласка на *trap* прекидну рутину, на стек се ставља PSW, па затим PC. Изглед стека након стављања PSW и PC приказан је на слици 8.ж. Како је PC 16 битни (јер су адресе 16 битне), на врх стека се стављају ова два бајта. Стек расте према нижим локацијама, а на нижу адресу се смешта нижи бајт 16 битне речи, на врху стека, при уласку у *trap* рутину, налази се нижи бајт адресе инструкције која је следећа по реду за извршавање у главном програму. То је вредност 5 (инструкција на адреси FF05h је следећа). Управо ова вредност се скида са стека у *trap* рутини инструкцијом POP и смешта у R1, а затим враћа на стек ради регуларног повратка у главни програм. Због тога је тражена вредност регистра R1 једнака 5. Изглед стека након инструкције POP приказан је на слици 8.з.



Слика 8.е. Изглед стека на почетку главног програма



Слика 8.ж. Изглед стека непосредно пре преласка на *trap* прекидну рутину



Слика 8.з. Изглед стека након инструкције POP, у регистру R1 се налази 05h

д) Првих 5 инструкција програма који се извршава су исте као и у претходном примеру.

Инструкција 6: Почетне вредности регистара и индикатора су: PC=00A1h, R0=1, R1=5, T=0.

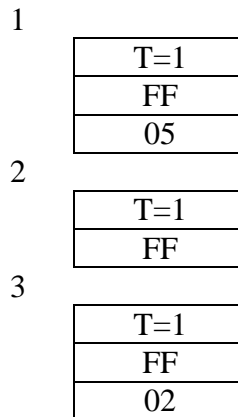
Ради се о инструкцији PUSH #2 са адресе 00A1h, која на стек ставља као један бајт вредност 2. Овде треба водити рачуна о томе да ће се ова вредност приликом повратка из прекидне рутине тумачити као нижи бајт повратне адресе. Сада се у *trap* рутини, уместо вредности 5, на врх стека ставља вредност 2. Због тога се, при повратку из ове рутине, у РС смешта вредност FF02h. То је адреса следеће инструкције која се извршава. На овој адреси налази се део кода инструкције MOV R0, #2, тачније њен други бајт, у коме је вероватно непосредна вредност 2 (или чак и део кода операције и начина адресирања). У сваком случају, не зна се како ће процесор тумачити вредност која се налази на адреси FF02h, када је дохвати као први бајт инструкције.

Проверава се да ли има прекида, констатује се да нема и прелази се на учитавање следеће инструкције.

Секвенца адреса је тако: FF00h, FF01h, FF03h, FF04h, 00A0h, 00A1h, 00A3h, FF02h, даље непознато.

Рб	Адреса	Инструкција	R0	R1	Стек	T
0	-	-	?	?	-	0
1	FF00	TRPD	?	?	-	0
2	FF01	MOV R0, #2	2	?	-	0
3	FF03	TRPE	2	?	-	1
4	FF04	DEC R0	1	?	-	1
			1	?	1	0
5	00A0	POP R1	1	5	2	0
6	00A1	PUSH #2	1	5	3	0
7	00A3	RTI	1	5	-	1
8	FF02	???				

Слика 8.и. Табела са секвенцом инструкција које се извршавају



Слика 8.ј Изглед стека у току извршавања секвенце инструкција

1.9 ЗАДАТАК 9

Адресни простор неког рачунара је величине 128 KB (кило бајта). Адресибилна јединица је 16 битна реч. Процесор је једноадресни, акумулатор је 16 битни, стек расте према нижим адресама, а показивач на врх стек показује на последњу заузету локацију, улазно-излазни и меморијски адресни простори су раздвојени, а механизам прекида је векторисан. IV табела (Interrupt Vector табела) почиње од адресе на коју указује регистар IVTP, чија је вредност 2h.

Инструкције RTI и INTE не реагују на спољашње прекиде. У PSW постоји бит I (Interrupt Enable) који се брише у фази *опслуживање прекида*, као и потребан број L бита. Регистар PSW је 16 битни. Процесор поседује три улаза за спољашње маскирајуће прекиде различитих приоритета, IRQM₀, IRQM₁ и IRQM₂, при чему је IRQM₀ највишег приоритета, као и линију IRQN за спољашње немаскирајуће прекиде. На линију IRQM₀ је везана периферија PER0, а на линију IRQN периферија PERN. Не постоји селективно маскирање прекида, а прекиди истог приоритета се прихватају.

Процесор може да генерише један унутрашњи захтев за прекид због извршавања инструкције прекида INS. Претпоставити да је извршавање инструкције INS тако реализовано да се у фази *извршавање операције* само у флип-флоп PRINS уписује вредност 1 и одмах прелази на фазу *опслуживање прекида*. У оквиру фазе *опслуживање прекида* се прелази на прекидну рутину на исти начин као и за све остале прекиде.

У кораку за обраду прекида на стеку се хардверски чувају PSW и PC тим редом. Адреса 16 битног регистра контролера периферије PER0 у коме се чува број улаза у IV табелу за PER0 је 80h. Линији IRQN одговара фиксан улаз 0 у IV табелу.

Дат је део главног програма на слици 9.а., прекидна рутина за PER0 дата је на слици 9.б, а за PERN на 9.в, изглед дела меморије почев од адресе 0 дат је на слици 9.г. Инструкција INTE на адреси 0100h означена је као 1. (прва) по редоследу извршавања, а свака следећа инструкција која се извршава означена је следећим редним бројем. У току извршавања 3. инструкције стижу захтеви за прекид од PER0 и од PERN. На почетку су сви бити PSW-а постављени на 0.

адреса	инструкција
0100	INTE
0102	LOAD #2h
0104	INC
0105	DEC
0107	INC

Слика 9.а Део главног програма

адреса	инструкција
X	INTE
X + 1	INS #5
X + 2	RTI

Слика 9.б Прекидна рутина за PER0

адреса	инструкција
Y	PUSH
Y + 1	INC
Y + 2	POP
Y + 3	RTI

Слика 9.в. Прекидна рутина за PERN

адреса	садржај
0000 0001h	
0001 00A1h	
0002 00BBh	
0003 0000h	
0004 00CDh	
0005 0000h	
0006 0000h	
0007 00CFh	

Слика 9.г. Изглед дела меморије почев од адресе 0

- Написати део програма којим се периферији PER0 додељује број улаза 2 у IVT.
- На којим адресама започињу прекидне рутине за периферије PER0 и PERN.
- Дати секвенцу адреса инструкција које се извршавају по датом сценарију, све док су оне познате.

Решење:

- Тражени програм приказан је на слици 9.д.
LOAD #2
OUT 80h

Слика 9.д. Део програма којим се периферији PER0 додељује број улаза 2 у IVT

б) Адресни простор процесора је величине 128 KB, односно 2^{16} речи, пошто је адресибилна јединица 16 битна реч. Одатле следи да су адресе 16 битне и да сваки улаз IV табеле заузима једну локацију у меморији. IV табела почиње од адресе на коју указује регистар IVTP, а вредност регистра IVTP је 2. Периферији PER0 је под а) додељен број улаза 2 у IV табели, а периферија PERN је везана на линију IRQN којој одговара фиксан улаз 0 у IV табели. То значи да ће се адресе на којима започињу прекидне рутине за периферије PER0 и PERN налазити на меморијским локацијама 4h (IVTP+2) и 2h (IVTP+0) респективно.

Због тога прекидна рутина за периферију PER0 почиње на адреси 00CDh, а за периферију PERN на адреси 00BBh. Поред тога адреса прекидне рутине за прекид PRINS за инструкцију INS #5 са адресе 00CEh у прекидној рутини за PER0 се налази у улазу 5 IV табели и износи 00CFh.

адреса	инструкција
00CDh	INTE
00CEh	INS #5
00CFh	RTI

Слика 9.б Прекидна рутина за PER0 и PRINS

адреса	инструкција
00BBh	PUSH
00BCh	INC
00BDh	POP
00BEh	RTI

Слика 9.в. Прекидна рутина за PERN

- Секвенца адреса наредби које се редом извршавају, почев од адресе 0100h приказана је на слици 9.ђ. Коментар је дат само за 3. инструкцију.

Инструкција 3: Почетне вредности регистара и индикатора су: PC=104h, ACC=2h, I=1, T=0, L=00.

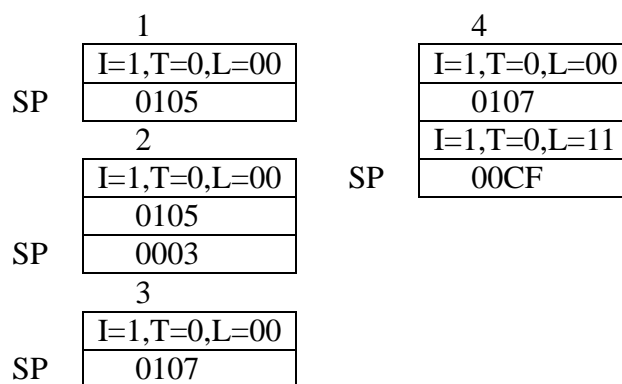
Ради се о инструкцији INC са адресе 104h, која увећава садржај акумулатора за један. Током извршавања инструкције стижу захтеви за прекид PRIRQM₀ и PRIRQN.

Инструкција реагује на прекид и прекиди PRIRQM₀ и PRIRQN постоје. Прекид PRIRQN је вишег приоритета од прекида PRIRQM₀, па се прекид PRIRQN прихвата. На стеку се прво чува програмска статусна реч PSW (I=1, T=0, L=00) па програмски бројач PC (105h). У бит I програмске статусне речи PSW се уписује вредност 0, а у бит T се уписује вредност 0, док се L бити не мењају, јер спољашњи немаскирајући прекиди не утичу на L бите. Процесор генерише фиксно број улаза у IV табелу који за спољашње немаскирајуће прекиде приспеле по линији IRQN износи 0. Брише се PRIRQN у коме је запамћен немаскирајући захтев за прекид. Процесор са адресе 2 чита адресу прве инструкције прекидне рутине (2 = 2 + 0*1), која износи Bbh, уписује у програмски бројач PC и прелази се на фазу *читање инструкције* следеће инструкције.

Треба приметити да је за спољашњи немаскирајући прекид број улаза у IV табелу фиксан, док за спољашње немаскирајуће прекиде контролер периферије шаље број улаза IV табелу када добије сигнал потврде.

Рб	Адреса	Инструкција	ACC	Стек	I	T	L	PRIRQN	PRIRQM ₀	PRIRQM ₁	PRIRQM ₂
0	-	-	?	-	0	0	00	0	0	0	0
1	0100	INTE	?	-	1	0	00	0	0	0	0
2	0102	LOAD #2	2	-	1	0	00	0	0	0	0
3	0104	INC	3	-	1	0	00	1	1	0	0
			3	1	0	0	00	0	1	0	0
4	00BB	PUSH	2	2	0	0	00	0	1	0	0
5	00BC	INC	4	2	0	0	00	0	1	0	0
6	00BD	POP	2	1	0	0	00	0	1	0	0
7	00BE	RTI	3	-	1	0	00	0	1	0	0
8	0105	DEC	2	-	1	0	00	0	1	0	0
			2	3	0	0	11	0	0	0	0
9	00CD	INTE	2	3	1	0	11	0	0	0	0
10	00CE	INS #5	2	4	0	0	11	0	0	0	0
11	00CF	RTI	2	3	1	0	11	0	0	0	0
12	00CF	RTI	2	-	1	0	00	0	0	0	0
13	0107	INC	3	-	1	0	00	0	0	0	0

Слика 9.ђ. Табела са секвенцом инструкција које се извршавају



Слика 9.е. Изглед стека у току извршавања секвенце инструкција

1.10 ЗАДАТАК 10

Адресни простор процесора је величине 8 GB, адресибилна јединица је 16 битна реч, а 32 битни бројеви се смештају тако да је на нижој адреси нижа реч. Процесор је једноадресни са раздвојеним улазно-излазном адресним простором, а механизам прекида је векторисан. Табела прекидних рутина (IV табела) има 4 фиксних улаза и почиње од адресе 2h.

Процесор има три улазне линију $IRQM_1$, $IRQM_2$ и $IRQM_3$ за спољашње маскирајуће прекиде, при чему је $IRQM_1$ највишег приоритета а $IRQM_3$ најнижег. На те линије су повезане периферије PER1, PER2 и PER3, којима су придружени улази 3, 2, и 0 у IV табели, респективно. Са #@%^~! је означена инструкција код које постоји грешка у коду операције. За унутрашњи прекид због грешке у коду операције (PRCOD) је додељен улаз 1 IV табели, а а прекидна рутина PRCOD је на адреси 1002h.

У PSW-у постоје бити I (*Interrupt enable*) и T (*Trap enable*), који се бришу фази *опслуживање прекида*, као и одређен број L бита. У кораку за обраду прекида на стеку се хардверски чувају ACC, PSW и PC тим редом. Стек расте према нижим локацијама, а указивач на врх стека SP указује на задњу заузету локацију. Акумулатор је 16 битни. Инструкције INTE, INTD, RTI, TRPE и TRPD не реагују на спољашње прекиде.

Не прихвата се прекид истог нивоа приоритета. Не постоји регистар маске IMR. Прекидне рутине периферија PER1, PER2 и PER3 почињу на адресама 1006h, 1008h и 100Ch, респективно.

Дат је део главног програма на слици 10.а, прекидне рутине на слици 10.б. Инструкција на адреси 0100h означена је као 1. (прва) по редоследу извршавања, а свака следећа инструкција која се извршава означена је следећим редним бројем. У току извршавања 2. инструкције стиже захтев за прекид по линији $IRQM_1$, у току 4. по линији $IRQM_3$, а у току 5. по линији $IRQM_2$. На почетку су сви бити PSW-а постављени на 0.

адреса	инструкција
0100h	INTE
0101h	LOAD #1h
0102h	INC
0103h	INTD
0104h	STORE 1h
0105h	TRPD

Слика 10.а Део главног програма

адреса	инструкција
1000h	INC
1001h	RTI
1002h	POP
1003h	INC
1004h	PUSH
1005h	RTI
1006h	INC
1007h	RTI
1008h	#@%^~!
1009h	LOAD #1h
100Ah	TRPE
100Bh	RTI
100Ch	PUSH
100Dh	INC
100Eh	POP
100Fh	RTI
1010h	INC
1011h	RTI

Слика 10.6 Прекидне рутине

а) Написати секвенцу адреса наредби које се редом извршавају, почев од адресе 0100h.

Решење

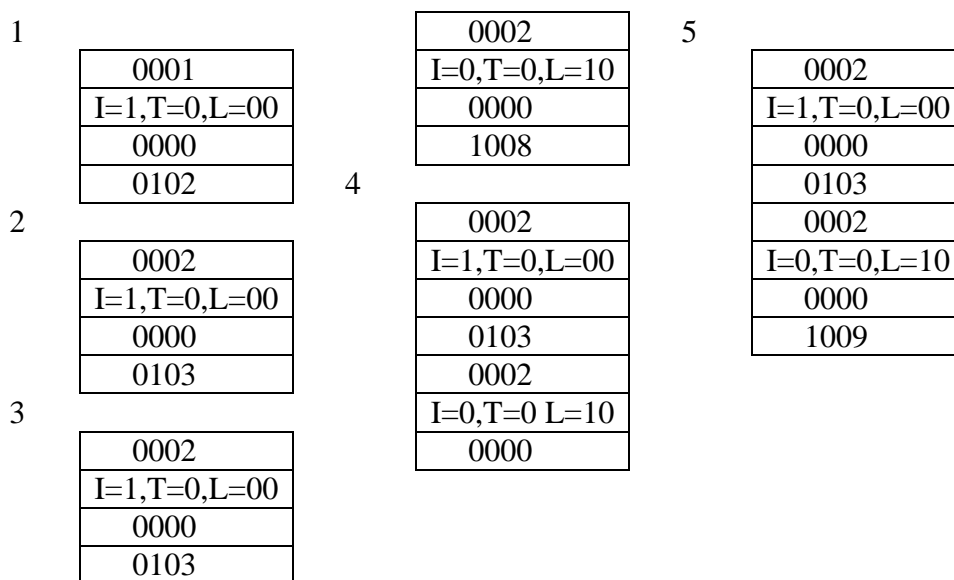
а)

Инструкција 6: Почетне вредности регистара и индикатора су: PC=1008h, ACC=2h, I=0, T=0, L=10.

Ради се о меморијском садржају који не одговара ни једној инструкцији што је представљено са #@%^~!. Ово одговара грешки у коду операције, и прелази се на кораке за његово опслуживање. На стеку је потребно сачувати РС инструкције **која је прекинута**, а не прве следеће. У корацима чувања контекста на стеку се чува прво акумулатор (ACC=2), па програмска статусна реч (I=0, T=0, L=10), а потом вредност програмског бројача (PC=1008h). Након корака чувања контекста прелази се на припрему новог контекста процесора. У бит I се уписује вредност 0, а у бит T се такође уписује вредност 0, L бити се не мењају. Постављање програмског бројача се обавља тако што се из одговарајућег улаза табеле прекидних рутина прочита вредност која одговара датом прекиду. Пошто су улази у табелу прекидних рутина фиксни процесор одређује да се ради о улазу број 0, који се користи у свим осталим случајевима. Процесор са адреса 4 и 5 чита адресу прве инструкције прекидне рутине ($4 = 2 + 1*2$). Прочитана је вредност 1002h. Ова вредност се уписује у регистар PC и прелази се на фазу читавања нове инструкције.

Рб	Адреса	Инструкција	ACC	Стек	I	T	L	PRIRQM1	PRIRQM2	PRIRQM3
0	-	-	?	-	0	0	00	0	0	0
1	100	INTE	?	-	1	0	00	0	0	0
2	101	LOAD #1	1	-	1	0	00	1	0	0
			1	1	0	0	11	0	0	0
3	1006	INC	2	1	0	0	11	0	0	0
4	1007	RTI	1	-	1	0	00	0	0	1
5	102	INC	2	-	1	0	00	0	1	1
			2	2	0	0	10	0	0	1
6	1008	#@%^~!	2	3	0	0	10	0	0	1
7	1002	POP	1008	4	0	0	10	0	0	1
8	1003	INC	1009	4	0	0	10	0	0	1
9	1004	PUSH	1009	5	0	0	10	0	0	1
10	1005	RTI	2	2	0	0	10	0	0	1
11	1009	LOAD #1	1	2	0	0	10	0	0	1
12	100A	TRPE	1	2	0	1	10	0	0	1
13	100B	RTI	2	-	1	0	00	0	0	1
14	103	INTD	2	-	0	0	00	0	0	1
15	104	STORE 1	2	-	0	0	00	0	0	1
16	105	TRPD	2	-	0	0	00	0	0	1

Слика 10.в. Табела са секвенцом инструкција које се извршавају



Слика 10.г Изглед стека у току извршавања секвенце инструкција

1.11 ЗАДАТАК 11

Адресни простор процесора је величине 16 МВ, адресибилна јединица је 16-битна реч, а вишечрени бројеви се смештају тако да је на нижој адреси нижа реч. Процесор је једноадресни, а механизам прекида је векторисан. Табела прекидних рутина (IV табела) има 8 фиксних улаза и почиње од адресе 32h.

Процесор има две улазне линије $IRQM_0$ и $IRQM_1$ за спољне маскирајуће прекиде, при чему је $IRQM_1$ вишег приоритета, и једну улазну линију $IRQN$ за спољне немаскирајуће прекиде. Њима су придружени улази 1, 2 и 5 у IV табели, респективно. Улаз 4 се користи у случају прекида после сваке инструкције (*trap*), а улаз 0 у случају унутрашњих прекида. Прекидне рутине започињу на следећим адресама: 1006h, 1009h, 1013h, 1000h, 101Ch респективно.

Процесор може да генерише један унутрашњи захтев за прекид због извршавања инструкције прекида *INS*. Претпоставити да је извршавање инструкције *INS* тако реализовано да се у фази *извршавање операције* само у флип-флоп *PRINS* уписује вредност 1 и одмах прелази на фазу *опслуживање прекида*. У оквиру фазе *опслуживање прекида* се прелази на прекидну рутину на исти начин као и за све остале прекиде.

У *PSW*-у постоје бити *I* (*Interrupt Enable*) и *T* (*Trap*) који се бришу у кораку за обраду прекида, као и одређен број *L* бита. У кораку за обраду прекида на стеку се хардверски чувају *PSW* и *PC* тим редом. Стек расте према нижим локацијама, а указивач на врх стека *SP* указује на задњу заузету локацију. Акумулатор је 32-битни. Инструкције *RTI*, *INTE*, *INTD*, *TRPE* и *TRPD* не реагују на спољашње прекиде. Не прихвата се прекид истог нивоа приоритета. Не постоји регистар маске *IMR*.

Дат је део главног програма на слици 11.а и прекидне рутине на слици 11.б. Инструкција на адреси 0100h означена је као 1. (прва) по редоследу извршавања, а свака следећа инструкција која се извршава означена је следећим редним бројем. У току извршавања 4. инструкције стиже захтев за прекид по линији $IRQM_0$, у току 6. по линији $IRQN$, а у току 8. по линији $IRQM_1$. На почетку су сви бити *PSW*-а постављени на 0. Почетни садржај локације 1h је 10h.

адреса	инструкција
0100h	TRPE
0101h	INTE
0102h	LOAD #1h
0105h	INC
0106h	STORE 1h
0109h	INTD

Слика 11.а Део главног програма

адреса	инструкција
1000h	POP
1001h	STORE 1h
1004h	PUSH
1005h	RTI
1006h	INTD
1007h	INS #2
1008h	RTI
1009h	LOAD 1h
100Ch	XOR #1h
100Fh	STORE 1h
1012h	RTI
1013h	LOAD 1h
1016h	INC
1017h	INTE
1018h	STORE 1h
101Bh	RTI
101Ch	POP
101Dh	ADD #3
1020h	PUSH
1021h	RTI

Слика 11.6 Прекидне рутине

а) Нацртати изглед првих 8 улаза у табели прекидних рутини, означити адресе релевантних локација и уписати садржаје у њих.

б) Написати секвенцу адреса наредби које се редом извршавају, почев од адресе 0100h.

в) Приказати садржај свих познатих локација на врху стека након извршавања 11. инструкције. За сачувану вредност PSW дати само вредност бита I, T и L. Назначити у коме смеру расте стек.

г) Која ће се вредност налазити на локацији 1h након извршења секвенце под б)?

Решење

а) Адреса на којој започиње улаз i који одговара некој периферији PER_j се добија према једначини: $IVTP + i * \text{sizeof}(\text{adrptr})$, где је adrptr показивач на меморијску локацију, а функција $\text{sizeof}(\text{adrptr})$ враћа дужину показивача исказану у адресибилним јединицама. У овом задатку имамо да је $\text{sizeof}(\text{adrptr})$ износи 23 бита то јест 2 речи тако да се добија да улази 0, 1, 2, 4 и 5 започињу на адресама: 32h, 34h, 36h, 3Ah и 3Ch респективно. Посматра се улаз 0 коме одговара прекидна рутина на адреси 0000 101Ch. Пошто се ради о процесору код кога виша реч иде на вишу адресу, а нижа на нижу за улаз 0 који се налази на адресама 32h и 33h адреса прекидне рутине се смешта на следећи начин: На нижу адресу (32h) се смешта нижа реч 101Ch а на вишу адресу (33h) се смешта виша реч 0000h. На исти начин се попуњавају и преостали улази у табели.

број улаза у IV табелу	меморијска адреса	садржај
7	0041h	
	0040h	
6	003Fh	
	003Eh	
5	003Dh	0000h
	003Ch	1013h
4	003Bh	0000h
	003Ah	1000h
3	0039h	
	0038h	
2	0037h	0000h
	0036h	1009h
1	0035h	0000h
	0034h	1006h
0	0033h	0000h
	0032h	101Ch

Слика 11.в Изглед свих 8 улаза у IV табели

б)

Инструкција 1: Почетне вредности регистара и индикатора су: PC=100h, ACC=?, I=0, T=0, L=00.

Ради се о инструкцији INTE са адресе 100h, која у бит I програмске статусне речи уписује вредност 1. Инструкција не реагује на прекид и прелази се на извршавање следеће инструкције.

Инструкција 2: Почетне вредности регистара и индикатора су: PC=101h, ACC=?, I=1, T=0, L=00.

Ради се о инструкцији TRPE са адресе 101h, која у бит T програмске статусне речи уписује вредност 1. Инструкција не реагује на прекид и прелази се на извршавање следеће инструкције.

Инструкција 3: Почетне вредности регистара и индикатора су: PC=102h, ACC=?, I=1, T=1, L=00.

Ради се о инструкцији LOAD #1h са адресе 102h, која у акумулатор уписује вредност 1. Проверава се да ли има прекида, констатује се да нема унутрашњих и спољашњих немаскирајућих, и немаскирајућих прекида. Пошто је бит T постављен на вредност 1 констатују се да су испуњени услови за прихватање захтева за прекидом после сваке инструкције. Прелази се на кораке чувања контекста процесора. На стеку се прво чува програмска статусна реч (I=1, T=1, L=00), па програмски бројач (105h) који заузима две адресибилне јединице. У бит I се уписује вредност 0, у бит T вредност 0, а бити L се не мењају. Процесор фиксно генерише број улаза (4). Процесор са адреса 3Ah и 3Bh чита адресу прве инструкције прекидне рутине (3Ah = 32h + 4*2). Прочитана је вредност 1000h. Ова вредност се уписује у регистар PC и прелази се на фазу читавања нове инструкције.

Инструкција 4: Почетне вредности регистара и индикатора су: PC=1000h, ACC=1h, I=0, T=0, L=00.

Ради се о инструкцији POP са адресе 1000h, са стека скида сачувану вредност ниже речи програмског бројача и уписује је у акумулатор (105h). Проверава се да ли има прекида, констатује се да нема унутрашњих и спољашњих немаскирајућих, а пошто бит I има вредност 0 не проверавају се немаскирајући прекиди, и пошто бит T има вредност 0 нема прекида после сваке инструкције, прелази се на читавање следеће инструкције.

Инструкција 5: Почетне вредности регистара и индикатора су: PC=1001h, ACC=105h, I=0, T=0, L=00.

Ради се о инструкцији STORE 1h са адресе 1001h, која садржај акумулатора уписује на меморијску локацију 1, вредност 105h. Проверава се да ли има прекида, констатује се да нема унутрашњих и спољашњих немаскирајућих, а пошто бит I има вредност 0 не проверавају се немаскирајући прекиди, и пошто бит T има вредност 0 нема прекида после сваке инструкције, прелази се на читавање следеће инструкције.

Инструкција 6: Почетне вредности регистара и индикатора су: PC=1004h, ACC=105h, I=0, T=0, L=00.

Ради се о инструкцији PUSH са адресе 1004h, која на стек враћа прочитану вредност. Инструкција реагује на прекид и пошто нема унутрашњих прекида проверавају се спољашњи немаскирајући прекиди. Пошто је периферија PERN процесору упутила захтев за прекидом по линији IRQN који је запамћен у регистру PRIRRN процесор прихвата захтева за прекидом. Прелази се на кораке чувања контекста процесора. На стеку се прво чува програмска статусна реч (I=0, T=0, L=00), па програмски бројач (1005h) који заузима две адресибилне јединице. У бит I се уписује вредност 0, у бит T вредност 0, а бити L се не мењају. Пошто су улази фиксни периферија PERN не шаље број улаза у табелу прекидних рутина већ га процесор генерише (5), процесор брише регистар PRIRRN. Процесор са адреса 3Ch и 3Dh чита адресу прве инструкције прекидне рутине (3Ch = 32h + 5*2). Прочитана је вредност 1013h. Ова вредност се уписује у регистар PC и прелази се на фазу читавања нове инструкције.

Инструкција 7: Почетне вредности регистара и индикатора су: PC=1013h, ACC=105h, I=0, T=0, L=00.

Ради се о инструкцији LOAD 1h са адресе 1013h, која у акумулатор уписује вредност са адресе 1, а то је 105h. Проверава се да ли има прекида, констатује се да нема унутрашњих и спољашњих немаскирајућих, а пошто бит I има вредност 0 не проверавају се немаскирајући прекиди, и пошто бит T има вредност 0 нема прекида после сваке инструкције, прелази се на читавање следеће инструкције.

Инструкција 8: Почетне вредности регистара и индикатора су: PC=1016h, ACC=105h, I=0, T=0, L=00.

Ради се о инструкцији INC са адресе 1016h, која увећава садржај акумулатора за један. Проверава се да ли има прекида, констатује се да нема унутрашњих и спољашњих немаскирајућих, а пошто бит I има вредност 0 не проверавају се немаскирајући прекиди, и пошто бит T има вредност 0 нема прекида после сваке инструкције, прелази се на читавање следеће инструкције.

Инструкција 9: Почетне вредности регистара и индикатора су: PC=1017h, ACC=106, I=0, T=0, L=00.

Ради се о инструкцији INTE са адресе 1017h, која у бит I програмске статусне речи уписује вредност 1. Инструкција не реагује на прекид и прелази се на извршавање следеће инструкције.

Инструкција 10: Почетне вредности регистара и индикатора су: PC=1018h, ACC=106h, I=1, T=0, L=00.

Ради се о инструкцији STORE 1h са адресе 1018h, која садржај акумулатора уписује на меморијску локацију 1, вредност 105h. Инструкција реагује на прекид и пошто нема унутрашњих прекида као ни спољашњих немаскирајућих прекида проверава се да ли је активан спољашњи немаскирајући прекид пошто бит I има вредност 1. Бит I је постављен на вредност 1 и пошто су периферије PERM0 и PERM1 процесору упутиле захтев за прекидом по линијама IRQM0 и IRQM1 који су запамћени у регистру PRIRRM, процесор проверава да ли је ниво приоритета линије вишег приоритета по којој је пристигао захтев за прекидом (линија IRQM1, приоритет 11) виши од текућег нивоа приоритета (00). Констатују се да су испуњени услови за прихватање захтева за прекидом. Прелази се на кораке чувања контекста процесора. На стеку се прво чува програмска статусна реч (I=1, T=0, L=00), па програмски бројач (101Bh) који заузима две адресибилне јединице. У бит I се уписује вредност 0, у бит T вредност 0, а у бит L вредност 11 која одговара приоритету линије IRQM1. Периферији PERM1 је придружен улаз 2 у табели прекидних рутина, процесор брише регистар PRIRRM1. Процесор са адреса 36h и 37h чита адресу прве инструкције прекидне рутине (36h = 32h + 2*2). Прочитана је вредност 1009h. Ова вредност се уписује у регистар PC и прелази се на фазу читавања нове инструкције.

Инструкција 11: Почетне вредности регистара и индикатора су: PC=1009h, ACC=106h, I=0, T=0, L=11.

Ради се о инструкцији LOAD 1h са адресе 1009h, која у акумулатор уписује вредност са адресе 1, а то је 106h. Проверава се да ли има прекида, констатује се да нема унутрашњих и спољашњих немаскирајућих, а пошто бит I има вредност 0 не проверавају се немаскирајући прекиди, и пошто бит T има вредност 0 нема прекида после сваке инструкције, прелази се на читавање следеће инструкције.

Инструкција 12: Почетне вредности регистара и индикатора су: PC=100Ch, ACC=106h, I=0, T=0, L=11.

Ради се о инструкцији XOR #1h са адресе 100Ch, која комплентира најнижи бит акумулатора. Проверава се да ли има прекида, констатује се да нема унутрашњих и спољашњих немаскирајућих, а пошто бит I има вредност 0 не проверавају се немаскирајући прекиди, и пошто бит T има вредност 0 нема прекида после сваке инструкције, прелази се на читавање следеће инструкције.

Инструкција 13: Почетне вредности регистара и индикатора су: PC=100Fh, ACC=107h, I=0, T=0, L=11.

Ради се о инструкцији STORE 1h са адресе 100Fh, која садржај акумулатора уписује на меморијску локацију 1, вредност 107h. Проверава се да ли има прекида, констатује се да нема унутрашњих и спољашњих немаскирајућих, а пошто бит I има вредност 0 не проверавају се немаскирајући прекиди, и пошто бит T има вредност 0 нема прекида после сваке инструкције, прелази се на читавање следеће инструкције.

Инструкција 14: Почетне вредности регистара и индикатора су: PC=1012h, ACC=107h, I=0, T=0, L=11.

Ради се о инструкцији RTI са адресе 1012h, која са стека скида контекст. Регистар PC добија вредност 100Bh, а програмска статусна реч I=0, T=0, L=00. Инструкција не реагује на прекид и прелази се на извршавање следеће инструкције.

Инструкција 15: Почетне вредности регистара и индикатора су: PC=100Bh, ACC=107h, I=0, T=0, L=00.

Ради се о инструкцији RTI са адресе 100Bh, која са стека скида контекст. Регистар PC добија вредност 1005h, а програмска статусна реч I=0, T=0, L=00. Инструкција не реагује на прекид и прелази се на извршавање следеће инструкције.

Инструкција 16: Почетне вредности регистара и индикатора су: PC=1005h, ACC=107h, I=0, T=0, L=00.

Ради се о инструкцији RTI са адресе 1005h, која са стека скида контекст. Регистар PC добија вредност 105h, а програмска статусна реч I=1, T=1, L=00. Инструкција не реагује на прекид и прелази се на извршавање следеће инструкције.

Инструкција 17: Почетне вредности регистара и индикатора су: PC=105h, ACC=107h, I=1, T=1, L=00.

Ради се о инструкцији INC са адресе 105h, која увећава садржај акумулатора за један. Инструкција реагује на прекид и пошто нема унутрашњих прекида као ни спољашњих немаскирајућих прекида проверава се да ли је активан спољашњи немаскирајући прекид пошто бит I има вредност 1. Бит I је постављен на вредност 1 и пошто је периферија PERM0 процесору упутила захтев за прекидом по линији IRQM0 који су запамћени у регистру PRIRRM0, процесор проверава да ли је ниво приоритета линије IRQM0 (приоритет 10) виши од текућег нивоа приоритета (00). Констатују се да су испуњени услови за прихватање захтева за прекидом. Прелази се на кораке чувања контекста процесора. На стеку се прво чува програмска статусна реч (I=1, T=1, L=00), па програмски бројач (106h) који заузима две адресибилне јединице. У бит I се уписује вредност 0, у бит T вредност 0, а у бит L вредност 10 која одговара приоритету линије IRQM0. Периферији PERM0 је придружен улаз 1 у табели прекидних рутина, процесор брише регистар PRIRRM0. Процесор са адреса 34h и 35h чита адресу прве инструкције прекидне рутине (34h = 32h + 1*2). Прочитана је вредност 1006h. Ова вредност се уписује у регистар PC и прелази се на фазу читавања нове инструкције.

Инструкција 18: Почетне вредности регистара и индикатора су: PC=1006h, ACC=108h, I=0, T=0, L=10.

Ради се о инструкцији INTD са адресе 1006h, која у бит I програмске статусне речи уписује вредност 0. Инструкција не реагује на прекид и прелази се на извршавање следеће инструкције.

Инструкција 19: Почетне вредности регистара и индикатора су: PC=1007h, ACC=108h, I=0, T=0, L=10.

Ради се о инструкцији INS #2 са адресе 1007h, која представља инструкцију софтверског прекида. Инструкција на стеку се прво чува програмска статусна реч (I=0, T=0, L=10) па програмски бројач (1008h) који заузима две адресибилне јединице. У бит I се уписује вредност 0, у бит T вредност 0, док се L бити не мењају. Процесор генерише број улаза на основу адресног дела инструкције, који износи 2. Процесор са адреса 36h и 37h чита адресу прве инструкције прекидне рутине (36h = 32h + 2*2). Прочитана је вредност 1009h. Ова вредност се уписује у регистар PC и прелази се на фазу читавања нове инструкције.

Инструкција 20: Почетне вредности регистара и индикатора су: PC=1009h, ACC=108h, I=0, T=0, L=10.

Ради се о инструкцији LOAD 1h са адресе 1009h, која у акумулатор уписује вредност са адресе 1, а то је 107h. Проверава се да ли има прекида, констатује се да нема и прелази се на читавање следеће инструкције.

Инструкција 21: Почетне вредности регистара и индикатора су: PC=100Ch, ACC=107h, I=0, T=0, L=10.

Ради се о инструкцији XOR #1h са адресе 100Ch, која комплентира најнижи бит акумулатора. Проверава се да ли има прекида, констатује се да нема и прелази се на читавање следеће инструкције.

Инструкција 22: Почетне вредности регистара и индикатора су: PC=100Fh, ACC=106h, I=0, T=0, L=10.

Ради се о инструкцији STORE 1h са адресе 100Fh, која садржај акумулатора уписује на меморијску локацију 1, вредност 106h. Проверава се да ли има прекида, констатује се да нема и прелази се на читавање следеће инструкције.

Инструкција 23: Почетне вредности регистара и индикатора су: PC=1012h, ACC=106h, I=0, T=0, L=10.

Ради се о инструкцији RTI са адресе 1012h, која са стека скида контекст. Регистар PC добија вредност 1008h, а програмска статусна реч I=0, T=0, L=10. Инструкција не реагује на прекид и прелази се на извршавање следеће инструкције.

Инструкција 24: Почетне вредности регистара и индикатора су: PC=1008h, ACC=106h, I=0, T=0, L=00.

Ради се о инструкцији RTI са адресе 1008h, која са стека скида контекст. Регистар PC добија вредност 1012h, а програмска статусна реч I=1, T=1, L=00. Инструкција не реагује на прекид и прелази се на извршавање следеће инструкције.

Инструкција 25: Почетне вредности регистара и индикатора су: PC=106h, ACC=106h, I=1, T=1, L=00.

Ради се о инструкцији STORE 1h са адресе 106h, која садржај акумулатора уписује на меморијску локацију 1, вредност 106h. Проверава се да ли има прекида, констатује се да нема унутрашњих и спољашњих немаскирајућих, а пошто бит I има вредност 0 не проверавају се немаскирајући прекиди. Пошто је бит T постављен на вредност 1 констатују се да су испуњени услови за прихватање захтева за прекидом после сваке инструкције. Прелази се на кораке чувања контекста процесора. На стеку се прво чува програмска статусна реч (I=1, T=1, L=00), па програмски бројач (109h) који заузима две адресибилне јединице. У бит I се уписује вредност 0, у бит T вредност 0, а бити L се не мењају. Процесор фиксно генерише број улаза (4). Процесор са адреса 3Ah и 3Bh чита адресу прве инструкције прекидне рутине ($3Ah = 32h + 4*2$). Прочитана је вредност 1000h. Ова вредност се уписује у регистар PC и прелази се на фазу читавања нове инструкције.

Инструкција 26: Почетне вредности регистара и индикатора су: PC=1000h, ACC=106h, I=0, T=0, L=00.

Ради се о инструкцији POP са адресе 1000h, са стека скида сачувану вредност ниже речи програмског бројача и уписује је у акумулатор (109h). Проверава се да ли има прекида, констатује се да нема и прелази се на читавање следеће инструкције.

Инструкција 27: Почетне вредности регистара и индикатора су: PC=1001h, ACC=109h, I=0, T=0, L=00.

Ради се о инструкцији STORE 1h са адресе 1001h, која садржај акумулатора уписује на меморијску локацију 1, вредност 109h. Проверава се да ли има прекида, констатује се да нема и прелази се на читавање следеће инструкције.

Инструкција 28: Почетне вредности регистара и индикатора су: PC=1004h, ACC=109h, I=0, T=0, L=00.

Ради се о инструкцији PUSH са адресе 1004h, која на стек враћа прочитану вредност. Проверава се да ли има прекида, констатује се да нема и прелази се на читавање следеће инструкције.

Инструкција 29: Почетне вредности регистара и индикатора су: PC=1005h, ACC=109h, I=0, T=0, L=00.

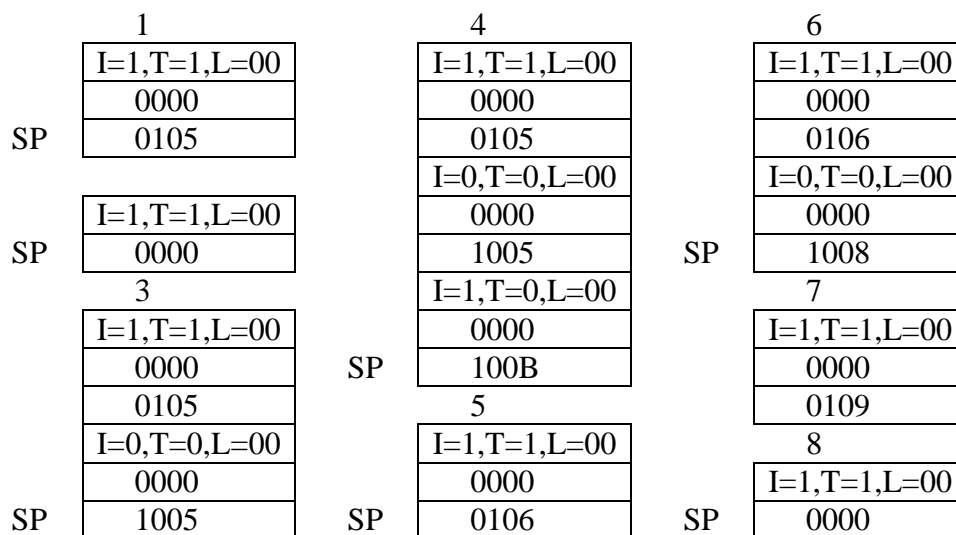
Ради се о инструкцији RTI са адресе 1005h, која са стека скида контекст. Регистар PC добија вредност 109h, а програмска статусна реч I=1, T=1, L=00. Инструкција не реагује на прекид и прелази се на извршавање следеће инструкције.

Инструкција 30: Почетне вредности регистара и индикатора су: PC=109h, ACC=109h, I=1, T=1, L=00.

Ради се о инструкцији INTD са адресе 109h, која у бит I програмске статусне речи уписује вредност 0. Инструкција не реагује на прекид и прелази се на извршавање следеће инструкције.

Pб	Адреса	Инструкција	ACC	Стек	I	T	L	PRINS	PRIRQN	PRIRQM ₁	PRIRQM ₀
0	-	-	?	-	0	0	00	0	0	0	0
1	0100	TRPE	?	-	0	1	00	0	0	0	0
2	0101	INTE	?	-	1	1	00	0	0	0	0
3	0102	LOAD #1	1	-	1	1	00	0	0	0	0
			1	1	0	0	00	0	0	0	0
4	1000	POP	0105	2	0	0	00	0	0	0	1
5	1001	STORE 1	0105	2	0	0	00	0	0	0	1
6	1004	PUSH	0105	1	0	0	00	0	1	0	1
			0105	3	0	0	00	0	0	0	1
7	1013	LOAD 1	0105	3	0	0	00	0	0	0	1
8	1016	INC	0106	3	0	0	00	0	0	1	1
9	1017	INTE	0106	3	1	0	00	0	0	1	1
10	1018	STORE 1	0106	3	1	0	00	0	0	1	1
			0106	4	0	0	11	0	0	0	1
11	1009	LOAD 1	0106	4	0	0	11	0	0	0	1
12	100C	XOR #1	0107	4	0	0	11	0	0	0	1
13	100F	STORE 1	0107	4	0	0	11	0	0	0	1
14	1012	RTI	0107	3	1	0	00	0	0	0	1
15	101B	RTI	0107	1	0	0	00	0	0	0	1
16	1005	RTI	0107	-	1	1	00	0	0	0	1
17	0105	INC	0108	-	1	1	00	0	0	0	1
			0108	5	0	0	10	0	0	0	0
18	1006	INTD	0108	5	0	0	10	0	0	0	0
19	1007	INS #2	0108	6	0	0	10	1	0	0	0
			0108	6	0	0	10	0	0	0	0
20	1009	LOAD 1	0107	6	0	0	10	0	0	0	0
21	100C	XOR #1	0106	6	0	0	10	0	0	0	0
22	100F	STORE 1	0106	6	0	0	10	0	0	0	0
23	1012	RTI	0106	5	0	0	10	0	0	0	0
24	1008	RTI	0106	-	1	1	00	0	0	0	0
25	0106	STORE 1	0106	-	1	1	00	0	0	0	0
			0106	7	0	0	00	0	0	0	0
26	1000	POP	0109	8	0	0	00	0	0	0	0
27	1001	STORE 1	0109	8	0	0	00	0	0	0	0
28	1004	PUSH	0109	7	0	0	00	0	0	0	0
29	1005	RTI	0109	-	1	1	00	0	0	0	0
30	0109	INTD	0109	-	0	1	00	0	0	0	0

Слика 11.г Табела са секвенцом инструкција које се извршавају



Слика 11.д Изглед стека у току извршавања секвенце инструкција

в) Након завршене секвенце на адреси 1h ће се наћи вредност 109h

1.12 ЗАДАТАК 12

Адресни простор процесора је величине 8 GB, адресибилна јединица је 16-битна реч, а вишеречни бројеви се смештају тако да је на вишој адреси виша реч. Процесор је једноадресни са раздвојеним меморијским и У/И адресним простором, механизам прекида је векторисан, IV (*Interrupt Vector*) табела почиње од адресе на коју указује регистар IVTP (*Interrupt Vector Table Pointer*), а регистар IVTP има вредност 2h.

Процесор има три улазне линије IRQM₀, IRQM₁ и IRQM₂ за спољне маскирајуће прекиде, при чему је IRQM₀ највишег, а IRQM₂ најнижег приоритета и једну улазну линију IRQN за спољне немаскирајуће прекиде на које су везане периферије PER0, PER1, PER2 и PERN респективно. Њима треба доделити улазе 2, 4, 1 и 5 у табели прекидних рутина, и којима одговарају прекидне рутине на адресама 1000h, 1011h, 1016h и 1006h, респективно. Адресе 8-битних регистара у којима се чувају бројеви улаза су 27h, 32h, F1h и C0h, респективно.

Процесор може да генерише један унутрашњи захтев за прекид због извршавања инструкције прекида INS. Претпоставити да је извршавање инструкције INS тако реализовано да се у фази *извршавање операције* само у флип-флоп PRINS уписује вредност 1 и одмах прелази на фазу *опслуживање прекида*. У оквиру фазе *опслуживање прекида* се прелази на прекидну рутину на исти начин као и за све остале прекиде.

У PSW-у постоје бити I (*Interrupt enable*) и T (*Trap enable*) који се бришу у оквиру фазе *опслуживање прекида*, као и одређен број L бита. У фази *опслуживање прекида* на стеку се хардверски чувају PSW и PC тим редом. Стек расте према нижим локацијама, а указивач на врх стека SP указује на задњу заузету локацију. Акумулатор је 16 битни. Инструкције INTE, INTD, TRPE и TRPD не реагују на спољашње прекиде. Инструкција RTI реагује на прекиде. Не прихвата се прекид истог нивоа приоритета. Не постоји регистар маске IMR.

Дат је део главног програма на слици 12.а и прекидне рутине на слици 13.б. Инструкција на адреси 0100h означена је као 1. (прва) по редоследу извршавања, а свака следећа инструкција која се извршава означена је следећим редним бројем. У току извршавања 1. инструкције стиже захтев за прекид по линији IRQ1, у току 5. по линији IRQN, а у току 7. по линији IRQ0. На почетку су сви бити PSW-а постављени на 0. Почетни садржај акумулатора је 0h, а меморијске локације 1h је 12h.

адреса	инструкција
0100h	INC
0101h	LOAD 1h
0104h	INS #1h
0105h	INTE
0106h	INC
0107h	OR #1h
0109h	STORE 1h
010Ch	INTD

Слика 14.а Део главног програма

адреса	инструкција
1000h	POP
1004h	PUSH
1001h	STORE 1h

1005h	RTI
1006h	PUSH
1007h	LOAD 1h
100Ah	ADD #3h
100Ch	STORE 1h
100Fh	POP
1010h	RTI
1011h	INC
1012h	STORE 1h
1015h	RTI
1016h	LOAD 1h
1019h	INC
101Ah	INTE
101Bh	RTI

Слика 14.б Прекидне рутине

а) Нацртати изглед првих 8 улаза у табели прекидних рутини, означити адресе релевантних локација и уписати садржаје у њих.

б) Написати део кода којим се периферијама додељују бројеви улаза.

в) Написати секвенцу адреса наредби које се редом извршавају, почев од адресе 0100h. Резултат дати табеларно тако да табела садржи редни број инструкције, адресу на којој започиње инструкција, саму инструкцију, садржај акумулатора након извршења инструкције, вредности свих познатих бита унутар програмске статусне речи.

г) Која ће се вредност налазити на локацији 1h након извршења секвенце под в)?

Решење

а)

број улаза у IV табелу	меморијска садржај адреса	
7	0011h	
	0010h	
6	000Fh	
	000Eh	
5	000Dh	0000h
	000Ch	1006h
4	000Bh	0000h
	000Ah	1011h
3	0009h	
	0008h	
2	0007h	0000h
	0006h	1000h
1	0005h	0000h
	0004h	1016h
0	0003h	
	0002h	

Слика 14.в Изглед свих 8 улаза у табели прекидних рутини

б)

Бројеви улаза чувају се у за то намењеним регистрима периферије. То су наведени регистри на адресама 27h, 32h, F1h и Ch. Ови регистри су дужине 8 бита и налазе се у улазно/излазном адресном простору, па им се приступа OUTB инструкцијама. Тражени део програма је приказан на следећој слици.

LOAD #02h	;napuni akumulator konstantom 2—broj ulaza za PER0
OUTB 27h	;prebaci bajt akumulatora u registar periferije PER0
LOAD #04h	;napuni akumulator konstantom 4—broj ulaza za PER1
OUTB 32h	;prebaci bajt akumulatora u registar periferije PER1
LOAD #01h	;napuni akumulator konstantom 1—broj ulaza za PER2
OUTB F1h	;prebaci bajt akumulatora u registar periferije PER2
LOAD #05h	;napuni akumulator konstantom 5—broj ulaza za PERN
OUTB C0h	;prebaci bajt akumulatora u registar periferije PERN

Слика 14.г Додела бројева улаза периферијама

в)

Инструкција 1: Почетне вредности регистара и индикатора су: PC=100h, ACC=0h, I=0, T=0, L=00.

Ради се о инструкцији INC са адресе 100h, која увећава садржај акумулатора за један. Проверава се да ли има прекида, констатује се да нема унутрашњих и спољашњих немаскирајућих, а пошто бит I има вредност 0 не проверавају се немаскирајући прекиди, и пошто бит T има вредност 0 нема прекида после сваке инструкције, прелази се на читавање следеће инструкције.

Инструкција 2: Почетне вредности регистара и индикатора су: PC=101h, ACC=1h, I=0, T=0, L=00.

Ради се о инструкцији LOAD 1h са адресе 101h, која у акумулатор уписује вредност са адресе 1, а то је 12h. Проверава се да ли има прекида, констатује се да нема унутрашњих и спољашњих немаскирајућих, а пошто бит I има вредност 0 не проверавају се немаскирајући прекиди, и пошто бит T има вредност 0 нема прекида после сваке инструкције, прелази се на читавање следеће инструкције.

Инструкција 3: Почетне вредности регистара и индикатора су: PC=104h, ACC=12h, I=0, T=0, L=00.

Ради се о инструкцији INT #1h са адресе 104h, која представља инструкцију софтверског прекида. На стеку се прво чува програмска статусна реч (I=0, T=0, L=00), па програмски бројач (105h) који заузима две адресибилне јединице. У бит I се уписује вредност 0, у бит T вредност 0, а бити L се не мењају. Процесор генерише број улаза на основу адресног дела инструкције, који износи 1. Процесор са адреса 4h и 5h чита адресу прве инструкције прекидне рутине ($4h = 2h + 1*2$). Прочитана је вредност 1016h, која одговара и прекиду периферије PER2. Ова вредност се уписује у регистар PC и прелази се на фазу читавања нове инструкције.

Инструкција 4: Почетне вредности регистара и индикатора су: PC=1016h, ACC=12h, I=0, T=0, L=00.

Ради се о инструкцији LOAD 1h са адресе 1016h, која у акумулатор уписује вредност са адресе 1, а то је 12h. Проверава се да ли има прекида, констатује се да нема унутрашњих и спољашњих немаскирајућих, а пошто бит I има вредност 0 не проверавају се немаскирајући прекиди, и пошто бит T има вредност 0 нема прекида после сваке инструкције, прелази се на читавање следеће инструкције.

Инструкција 5: Почетне вредности регистара и индикатора су: PC=1019h, ACC=12h, I=0, T=0, L=00.

Ради се о инструкцији INC са адресе 1019h, која увећава садржај акумулатора за један. Инструкција реагује на прекид и пошто нема унутрашњих прекида проверава се да ли је активан спољашњи немаскирајући прекид. Пошто је захтев периферије PERN приспео по линији IRQN активан процесор прихвата захтев за прекидом. Прелази се на кораке чувања

контекста процесора. На стеку се прво чува програмска статусна реч ($I=0$, $T=0$, $L=0$), па програмски бројач (101Ah) који заузима две адресибилне јединице. У бит I се уписује вредност 0, у бит T вредност 0, а бити L се не мењају. Процесор по линији INTAN јавља да прихваћан захтев за прекидом. Периферија PERN процесору шаље број улаза у табелу прекидних рутина (5), процесор брише регистар PRIRRN. Процесор са адреса Ch и Dh чита адресу прве инструкције прекидне рутине ($Ch = 2h + 5 \cdot 2$). Прочитана је вредност 1006h. Ова вредност се уписује у регистар PC и прелази се на фазу читавања нове инструкције.

Инструкција 6: Почетне вредности регистара и индикатора су: PC=1006h, ACC=13h, I=0, T=0, L=00.

Ради се о инструкцији PUSH са адресе 1006h, која на стек смешта акумулатор. Проверава се да ли има прекида, констатује се да нема унутрашњих и спољашњих немаскирајућих, а пошто бит I има вредност 0 не проверавају се немаскирајући прекиди, и пошто бит T има вредност 0 нема прекида после сваке инструкције, прелази се на читавање следеће инструкције.

Инструкција 7: Почетне вредности регистара и индикатора су: PC=1007h, ACC=13h, I=0, T=0, L=00.

Ради се о инструкцији LOAD 1h са адресе 1007h, која у акумулатор уписује вредност са адресе 1, а то је 12h. Проверава се да ли има прекида, констатује се да нема унутрашњих и спољашњих немаскирајућих, а пошто бит I има вредност 0 не проверавају се немаскирајући прекиди, и пошто бит T има вредност 0 нема прекида после сваке инструкције, прелази се на читавање следеће инструкције.

Инструкција 8: Почетне вредности регистара и индикатора су: PC=100Ah, ACC=12h, I=0, T=0, L=00.

Ради се о инструкцији ADD #3h са адресе 100Ah, која увећава садржај акумулатора за 3h. Проверава се да ли има прекида, констатује се да нема унутрашњих и спољашњих немаскирајућих, а пошто бит I има вредност 0 не проверавају се немаскирајући прекиди, и пошто бит T има вредност 0 нема прекида после сваке инструкције, прелази се на читавање следеће инструкције.

Инструкција 9: Почетне вредности регистара и индикатора су: PC=100Ch, ACC=15h, I=0, T=0, L=00.

Ради се о инструкцији STORE 1h са адресе 100Ch, која садржај акумулатора уписује на меморијску локацију 1, вредност 15h. Проверава се да ли има прекида, констатује се да нема унутрашњих и спољашњих немаскирајућих, а пошто бит I има вредност 0 не проверавају се немаскирајући прекиди, и пошто бит T има вредност 0 нема прекида после сваке инструкције, прелази се на читавање следеће инструкције.

Инструкција 10: Почетне вредности регистара и индикатора су: PC=100Fh, ACC=15h, I=0, T=0, L=00.

Ради се о инструкцији POP са адресе 100Fh, са стека скида сачувану вредност акумулатора и уписује је у акумулатор. Проверава се да ли има прекида, констатује се да нема унутрашњих и спољашњих немаскирајућих, а пошто бит I има вредност 0 не проверавају се немаскирајући прекиди, и пошто бит T има вредност 0 нема прекида после сваке инструкције, прелази се на читавање следеће инструкције.

Инструкција 11: Почетне вредности регистара и индикатора су: PC=1010h, ACC=13h, I=0, T=0, L=00.

Ради се о инструкцији RTI са адресе 1010h, која са стека скида контекст. Регистар PC добија вредност 101Ah, програмска статусна реч I=0, T=0, L=00. Пошто инструкција **реагује** на прекид проверава се да ли има прекида, констатује се да нема унутрашњих и спољашњих немаскирајућих, а пошто бит I има вредност 0 не проверавају се немаскирајући прекиди, и пошто бит T има вредност 0 нема прекида после сваке инструкције, прелази се на читавање следеће инструкције.

Инструкција 12: Почетне вредности регистара и индикатора су: PC=101Ah, ACC=13h, I=0, T=0, L=00.

Ради се о инструкцији INTE са адресе 101Ah, која у бит I програмске статусне речи уписује вредност 1. Инструкција не реагује на прекид и прелази се на извршавање следеће инструкције.

Инструкција 13: Почетне вредности регистара и индикатора су: PC=101Bh, ACC=13h, I=0, T=0, L=00.

Ради се о инструкцији RTI са адресе 101Bh, која са стека скида контекст. Регистар PC добија вредност 105h, програмска статусна реч I=0, T=0, L=00. Пошто инструкција реагује на прекид проверава се да ли има прекида, констатује се да нема унутрашњих и спољашњих немаскирајућих, а пошто бит I има вредност 0 не проверавају се немаскирајући прекиди, и пошто бит T има вредност 0 нема прекида после сваке инструкције, прелази се на читавање следеће инструкције.

Инструкција 14: Почетне вредности регистара и индикатора су: PC=105h, ACC=13h, I=0, T=0, L=00.

Ради се о инструкцији INTE са адресе 105h, која у бит I програмске статусне речи уписује вредност 1. Инструкција не реагује на прекид и прелази се на извршавање следеће инструкције.

Инструкција 15: Почетне вредности регистара и индикатора су: PC=106h, ACC=13h, I=1, T=0, L=00.

Ради се о инструкцији INC са адресе 106h, која увећава садржај акумулатора за један. Инструкција реагује на прекид и пошто нема унутрашњих прекида као ни спољашњих немаскирајућих прекида проверава се да ли је активан спољашњи немаскирајући прекид пошто бит I има вредност 1. Бит I је постављен на вредност 1 и пошто су периферије PER0 и PER1 процесору упутиле захтев за прекидом по линијама IRQ0 и IRQ1 који су запамћени у регистру PRIRR, процесор проверава да ли је ниво приоритета линије вишег приоритета по којој је пристигао захтев за прекидом (линија IRQ0, приоритет 11) виши од текућег нивоа приоритета (00). Констатују се да су испуњени услови за прихватање захтева за прекидом. Прелази се на кораке чувања контекста процесора. На стеку се прво чува програмска статусна реч (I=1, T=0, L=00), па програмски бројач (107h) који заузима две адресибилне јединице. У бит I се уписује вредност 0, у бит T вредност 0, а у бите L вредност 11 која одговара приоритету линије IRQ0. Периферија PER0 по линијама магистрале података шаље број улаза у табелу прекидних рутина (2), процесор брише регистар PRIRR0. Процесор са адреса 6h и 7h чита адресу прве инструкције прекидне рутине (6h = 2h + 2*2). Прочитана је вредност 1000h. Ова вредност се уписује у регистар PC и прелази се на фазу читавања нове инструкције.

Инструкција 16: Почетне вредности регистара и индикатора су: PC=1000h, ACC=14h, I=0, T=0, L=11.

Ради се о инструкцији POP са адресе 1000h, са стека скида сачувану вредност млађе речи програмског бројача и уписује је у акумулатор (107h). Проверава се да ли има прекида,

констатује се да нема унутрашњих и спољашњих немаскирајућих, а пошто бит I има вредност 0 не проверавају се немаскирајући прекиди, и пошто бит T има вредност 0 нема прекида после сваке инструкције, прелази се на читавање следеће инструкције.

Инструкција 17: Почетне вредности регистара и индикатора су: PC=1001h, ACC=107h, I=0, T=0, L=11.

Ради се о инструкцији STORE 1h са адресе 1001h, која садржај акумулатора уписује на меморијску локацију 1, вредност 107h. Проверава се да ли има прекида, констатује се да нема унутрашњих и спољашњих немаскирајућих, а пошто бит I има вредност 0 не проверавају се немаскирајући прекиди, и пошто бит T има вредност 0 нема прекида после сваке инструкције, прелази се на читавање следеће инструкције.

Инструкција 18: Почетне вредности регистара и индикатора су: PC=1004h, ACC=107h, I=0, T=0, L=11.

Ради се о инструкцији PUSH са адресе 1004h, која на стек смешта акумулатор. Проверава се да ли има прекида, констатује се да нема унутрашњих и спољашњих немаскирајућих, а пошто бит I има вредност 0 не проверавају се немаскирајући прекиди, и пошто бит T има вредност 0 нема прекида после сваке инструкције, прелази се на читавање следеће инструкције.

Инструкција 19: Почетне вредности регистара и индикатора су: PC=1005h, ACC=107h, I=0, T=0, L=11.

Ради се о инструкцији RTI са адресе 1005h, која са стека скида контекст. Регистар PC добија вредност 107h, програмска статусна реч I=1, T=0, L=00. Инструкција **реагује** на прекид и пошто нема унутрашњих прекида као ни спољашњих немаскирајућих прекида проверава се да ли је активан спољашњи немаскирајући прекид пошто бит I има вредност 1. Бит I је постављен на вредност 1 и пошто је периферија PER1 процесору упутила захтев за прекидом по линији IRQ1 који је запамћен у регистру PRIRR1, процесор проверава да ли је ниво приоритета линије вишег приоритета по којој је пристигао захтев за прекидом (линија IRQ1, приоритет 10) виши од текућег нивоа приоритета (00). Констатују се да су **испуњени** услови за прихватање захтева за прекидом. Прелази се на кораке чувања контекста процесора. На стеку се прво чува програмска статусна реч (I=1, T=0, L=00), па програмски бројач (107h) који заузима две адресибилне јединице. У бит I се уписује вредност 0, у бит T вредност 0, а у бите L вредност 10 која одговара приоритету линије IRQ1. Периферија PER1 по линијама магистрале података шаље број улаза у табелу прекидних рутина (4), процесор брише регистар PRIRRN. Процесор са адреса Ah и Bh чита адресу прве инструкције прекидне рутине ($Ah = 2h + 4*2$). Прочитана је вредност 1011h. Ова вредност се уписује у регистар PC и прелази се на фазу читавања нове инструкције.

Инструкција 20: Почетне вредности регистара и индикатора су: PC=1011h, ACC=107h, I=0, T=0, L=10.

Ради се о инструкцији INC са адресе 1011h, која увећава садржај акумулатора за један. Проверава се да ли има прекида, констатује се да нема и прелази се на читавање следеће инструкције.

Инструкција 21: Почетне вредности регистара и индикатора су: PC=1012h, ACC=108h, I=0, T=0, L=10.

Ради се о инструкцији STORE 1h са адресе 1012h, која садржај акумулатора уписује на меморијску локацију 1, вредност 108h. Проверава се да ли има прекида, констатује се да нема и прелази се на читавање следеће инструкције.

Инструкција 22: Почетне вредности регистара и индикатора су: PC=1015h, ACC=108h, I=0, T=0, L=10.

Ради се о инструкцији RTI са адресе 1015h, која са стека скида контекст. Регистар PC добија вредност 107h, програмска статусна реч I=1, T=0, L=00. Проверава се да ли има прекида, констатује се да нема и прелази се на читавање следеће инструкције.

Инструкција 23: Почетне вредности регистара и индикатора су: PC=107h, ACC=108h, I=1, T=0, L=00.

Ради се о инструкцији OR #1h са адресе 107h, која на позицију 0 акумулатора уписује вредност 1. Проверава се да ли има прекида, констатује се да нема и прелази се на читавање следеће инструкције.

Инструкција 24: Почетне вредности регистара и индикатора су: PC=109h, ACC=109h, I=1, T=0, L=00.

Ради се о инструкцији STORE 1h са адресе 109h, која садржај акумулатора уписује на меморијску локацију 1, вредност 109h. Проверава се да ли има прекида, констатује се да нема и прелази се на читавање следеће инструкције.

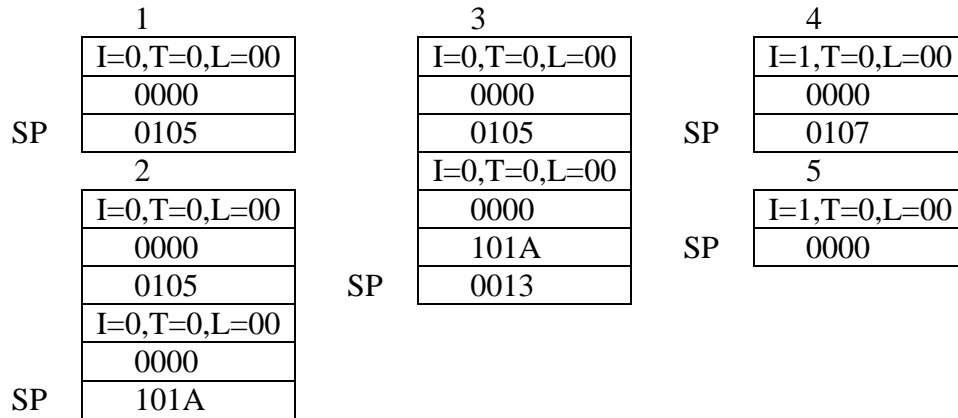
Инструкција 24: Почетне вредности регистара и индикатора су: PC=10Ch, ACC=109h, I=1, T=0, L=00.

Ради се о инструкцији INTD са адресе 10Ch, која у бит I програмске статусне речи уписује вредност 0. Инструкција не реагује на прекид и прелази се на извршавање следеће инструкције.

Рб	Адреса	Инструкција	ACC	Стек	I	T	L	PRINS	PRIRQN	PRIRQ0	PRIRQ1	PRIRQ2
0	-	-	0	-	0	0	00	0	0	0	0	0
1	100	INC	1	-	0	0	00	0	0	0	1	0
2	101	LOAD 1h	12	-	0	0	00	0	0	0	1	0
3	104	INS #1h	12	1	0	0	00	1	0	0	1	0
			12	1	0	0	00	0	0	0	1	0
4	1016	LOAD 1h	12	1	0	0	00	0	0	0	1	0
5	1019	INC	13	1	0	0	00	0	1	0	1	0
			13	2	0	0	00	0	0	0	1	0
6	1006	PUSH	13	3	0	0	00	0	0	0	1	0
7	1007	LOAD 1h	12	3	0	0	00	0	0	1	1	0
8	100A	ADD #3h	15	3	0	0	00	0	0	1	1	0
9	100C	STORE 1h	15	3	0	0	00	0	0	1	1	0
10	100F	POP	13	2	0	0	00	0	0	1	1	0
11	1010	RTI	13	1	0	0	00	0	0	1	1	0
12	101A	INTE	13	1	1	0	00	0	0	1	1	0
13	101B	RTI	13	-	0	0	00	0	0	1	1	0
14	105	INTE	13	-	1	0	00	0	0	1	1	0
15	106	INC	14	-	1	0	00	0	0	1	1	0
			14	4	0	0	11	0	0	0	1	0
16	1000	POP	107	5	0	0	11	0	0	0	1	0
17	1001	STORE 1h	107	5	0	0	11	0	0	0	1	0
18	1004	PUSH	107	4	0	0	11	0	0	0	1	0
19	1005	RTI	107	-	1	0	00	0	0	0	1	0
			107	4	0	0	10	0	0	0	0	0
20	1011	INC	108	4	0	0	10	0	0	0	0	0
21	1012	STORE 1h	108	4	0	0	10	0	0	0	0	0
22	1015	RTI	108	-	1	0	00	0	0	0	0	0
23	107	OR #1h	109	-	1	0	00	0	0	0	0	0

24	109	STORE 1h	109	-	1	0	00	0	0	0	0	0
25	10C	INTD	109	-	0	0	00	0	0	0	0	0

Слика 14.д Табела са секвенцом инструкција које се извршавају



Слика 14.ђ Изглед стека у току извршавања секвенце инструкција

г) Након завршене секвенце на адреси 1h ће се наћи вредност 109h.

1.13 ЗАДАТАК 13

Адресни простор процесора је величине 8GB, адресибилна јединица је 16 битна реч, а вишечни бројеви и адресе се смештају у меморију тако да се на вишој адреси смешта нижа реч. Процесор је једноадресни са раздвојеним меморијским и У/И адресним простором, механизам прекида је векторисан, табела прекидних рутина (*Interrupt Vector*) почиње од адресе на коју указује регистар IVTP (*Interrupt Vector Table Pointer*), а регистар IVTP има вредност 8h.

Процесор има три улазне линије IRQM₀, IRQM₁ и IRQM₂ за спољне маскирајуће прекиде, при чему је IRQM₀ највишег, а IRQM₂ најнижег приоритета и једну улазну линију IRQN за спољне не маскирајуће прекиде, на које су везане периферије PER0, PER1, PER2 и PERN респективно. Њима треба доделити улазе 2, 3, 1 и 5 у табели прекидних рутина, и којима одговарају прекидне рутине на адресама 1008h, 1000h, 1014h и 100Eh, респективно. Адресе 16 битних регистара у којима се чувају бројеви улаза су FF00h, EF12h, BC15h и ABCDh, респективно.

У регистру PSW постоје бити I (*Interrupt Enable*) и T (*Trap*) који се бришу у кораку за обраду прекида, као и два L_{1.0} бита. У кораку за обраду прекида на стек се хардверски стављају ACC, PSW и PC, тим редом. Стек расте према нижим локацијама, а указивач на врх стека SP указује на задњу заузету локацију. Акумулатор ACC и PSW су 16 битни. Инструкције INTE, INTD, TRPE, TRPD и RTI не реагују на прекиде. Не прихвата се прекид истог нивоа приоритета. Не постоји регистар маске IMR.

Дат је део главног програма на слици 15.а и прекидне рутине на слици 15.б. Инструкција на адреси 100h означена је као 1. (прва) по редоследу извршавања, а свака следећа инструкција која се извршава означена је следећим редним бројем. У току извршавања 1. инструкције стиже захтев за прекид по линији IRQM₂, у току 3. по линији IRQM₁, у току 6. по линији IRQN, а у току 8. по линији IRQM₀. На почетку су сви бити регистра PSW постављени на 0. Почетни садржај акумулатора је 0h, а меморијске локације 1h је 1234h.

адреса	инструкција
0100h	LOAD 1
0103h	INC
0104h	INTE
0105h	STORE 1
0108h	INC
0109h	INTD

Слика 15.а Део главног програма

адреса	инструкција
1000h	LOAD 1
1003h	DEC
1004h	STORE 1
1007h	RTI
1008h	POP
1009h	PUSH
100Ah	STORE 1
100Dh	RTI
100Eh	INTE
100Fh	INC

1010h STORE 1
 1013h RTI
 1014h RTI

Слика 15.6 Прекидне рутине

- а) Написати део програма којим се додељују бројеви улаза наведеним периферијама.
 б) Нацртати изглед првих 8 улаза у табели прекидних рутина, означити адресе релевантних локација и уписати садржаје у њих.
 в) Написати секвенцу адреса наредби које се редом извршавају почев од адресе 100h. Резултат дати табеларно тако да табела садржи редни број инструкције, адресу на којој започиње инструкција, саму инструкцију, садржај акумулатора након извршења инструкције, вредности свих познатих бита унутар програмске статусне речи, и изглед стека. Резултат дати након фазе извршења инструкције и уколико је у фази опслуживања прекида прихваћен прекид и након фазе опслуживања прекида.
 г) Написати део програма којим се периферији PER2 додељује иста прекидна рутина као и периферији PER1.

Решење

- а)
 LOAD #2h
 OUT FF00h
 LOAD #3h
 OUT EF12h
 LOAD #1h
 OUT BC15h
 LOAD #5h
 OUT ABCDh

Слика 15.в Додела бројева улаза периферијама

- б) На исти начин како је објашњено у претходним задацима добијају се вредности које је потребно уписати у табелу прекидних рутина која је дата на следећој слици.

број улаза у IV табелу	меморијска садржај адреса	
7	17	
	16	
6	15	
	14	
5	13	100E
	12	0000
4	11	
	10	
3	F	1000
	E	0000
2	D	1008
	C	0000
1	B	1014

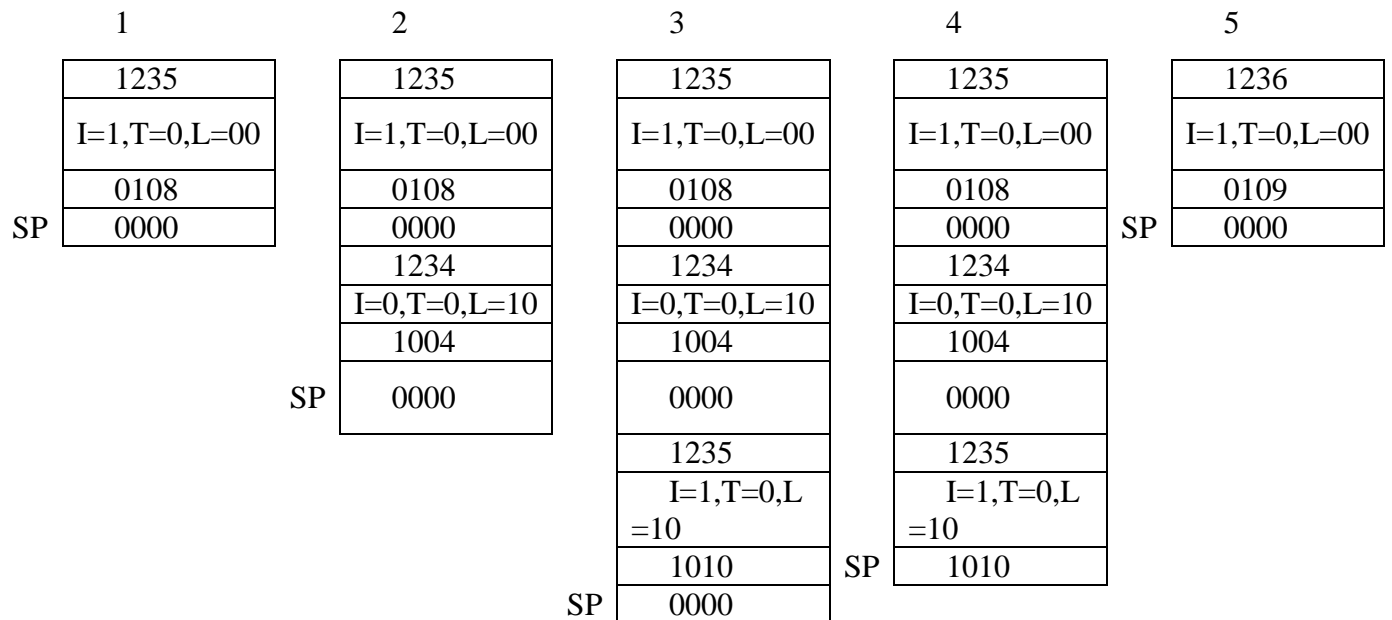
	A	0000
	9	
0	8	

Слика 15.г Изглед свих 8 улаза у табели прекидних рутини

В)

Рб	Адреса	Инструкција	ACC	Стек	I	T	L	PRIRQN	PRIRQM ₀	PRIRQM ₁	PRIRQM ₂
0	-	-	0	-	0	0	00	0	0	0	0
1	100	LOAD 1	1234	-	0	0	00	0	0	0	1
2	103	INC	1235	-	0	0	00	0	0	0	1
3	104	INTE	1235	-	1	0	00	0	0	1	1
4	105	STORE 1	1235	-	1	0	00	0	0	1	1
				1	0	0	10	0	0	0	1
5	1000	LOAD 1	1235	1	0	0	10	0	0	0	1
6	1003	DEC	1234	1	0	0	10	1	0	0	1
				2	0	0	10	0	0	0	1
7	100E	INTE	1234	2	1	0	10	0	0	0	1
8	100F	INC	1235	2	1	0	10	0	1	0	1
				3	0	0	11	0	0	0	1
9	1008	POP	0	4	0	0	11	0	0	0	1
10	1009	PUSH	0	3	0	0	11	0	0	0	1
11	100A	STORE 1	0	3	0	0	11	0	0	0	1
12	100D	RTI	1235	2	1	0	10	0	0	0	1
13	1010	STORE 1	1235	2	1	0	10	0	0	0	1
14	1013	RTI	1234	1	0	0	10	0	0	0	1
15	1004	STORE 1	1234	1	0	0	10	0	0	0	1
16	1007	RTI	1235	-	1	0	00	0	0	0	1
17	108	INC	1236	-	1	0	00	0	0	0	1
				5	0	0	01	0	0	0	0
18	1014	RTI	1236	5	0	0	01	0	0	0	0
19	109	INTD	1236	-	0	0	00	0	0	0	0

Слика 15.ђ Табела са секвенцом инструкција које се извршавају



Слика 15.е Изглед стека у току извршавања секвенце инструкција

д)

```
LOAD #0h
STORE Ah
LOAD #1000h
STORE Bh
или
LOAD Eh
STORE Ah
LOAD Fh
STORE Bh
или
LOAD #3h
OUT BC15h
```