

МЕЃУНАРОДЕН ЦЕНТАР ЗА СЛАВЈАНСКА ПРОСВЕТА - СВЕТИ НИКОЛЕ

«МЕЃУНАРОДЕН ДИЈАЛОГ: ИСТОК - ЗАПАД»
(ЕКОНОМИЈА, БЕЗБЕДНОСНО ИНЖЕНЕРСТВО,
ИНФОРМАТИКА)

СПИСАНИЕ
на научни трудови

**ДВАНАЕСЕТТА МЕЃУНАРОДНА
НАУЧНА КОНФЕРЕНЦИЈА
„МЕЃУНАРОДЕН ДИЈАЛОГ: ИСТОК - ЗАПАД“
МЕЃУНАРОДЕН СЛАВЈАНСКИ УНИВЕРЗИТЕТ
„ГАВРИЛО РОМАНОВИЧ ДЕРЖАВИН“
СВЕТИ НИКОЛЕ - БИТОЛА**

Година VIII

Број 1

Април 2021

- СВЕТИ НИКОЛЕ, Р. СЕВЕРНА МАКЕДОНИЈА -
- 2021 -

Издавач: Меѓународен центар за славјанска просвета - Свети Николе

За издавачот: м-р Михаела Ѓорчева, директор

Наслов: «МЕЃУНАРОДЕН ДИЈАЛОГ: ИСТОК - ЗАПАД» (ЕКОНОМИЈА, БЕЗБЕДНОСНО ИНЖЕНЕРСТВО, ИНФОРМАТИКА)

Организационен одбор:

Претседател: проф. д-р Јордан Ѓорчев

Заменик претседател: д-р Стромов Владимир Јуревич, Русија

Член: м-р Борче Серафимовски

Член: м-р Милена Спасовска

Уредувачки одбор:

Проф. д-р Ленче Петреска - Република Северна Македонија

Проф. д-р Александар Илиевски - Република Северна Македонија

Проф. д-р Мирослав Крстиќ - Република Србија

Проф. д-р Момчило Симоновиќ - Република Србија

Проф. д-р Тодор Галунов - Република Бугарија

Проф. д-р Даниела Тасевска - Република Бугарија

Доц. д-р Хаџиб Салкиќ - Република Босна и Херцеговина

Проф. д-р Татјана Осадчаја - Руска Федерација

Доц. д-р Вера Шунаева - Руска Федерација

Уредник: проф. д-р Јордан Ѓорчев

Компјутерска обработка и дизајн: Адриано Панајотов, Маја Маријана Панајотова, Благој Митев

ISSN (принт) 1857-9299

ISSN (онлајн) 1857-9302

Адреса на комисијата: ул. Маршал Тито 77, Свети Николе, Р. Северна Македонија

Контакт телефон: +389 (0)32 440 330

Уредувачкиот одбор им се заблагодарува на сите учесници за соработката!

Напомена:

Уредувачкиот одбор на списанието «МЕЃУНАРОДЕН ДИЈАЛОГ: ИСТОК-ЗАПАД» не одговара за можните повреди на авторските права на научните трудови објавени во списанието. Целосната одговорност за оригиналноста, автентичноста и лекторирањето на научните трудови објавени во списанието е на самите автори на трудовите.

Секој научен труд пред објавувањето во списанието «МЕЃУНАРОДЕН ДИЈАЛОГ: ИСТОК-ЗАПАД» е рецензиран од двајца анонимни рецензенти од соодветната научна област.

Печати: Печатница и книжарница „Славјански“, Свети Николе

Тираж: 100

МЕЃУНАРОДЕН ДИЈАЛОГ

ИСТОК - ЗАПАД

ЕКОНОМИЈА, БЕЗБЕДНОСНО ИНЖЕНЕРСТВО,
ИНФОРМАТИКА

ОБЛАСТ
ИНФОРМАТИКА

Доц. д-р Жанета Сервини

МСУ „Г. Р. Державин“ Свети Николе – Битола
Р. Северна Македонија

М-р Јани Сервини

СОТУ „Ѓорѓи Наумов“ – Битола
Р. Северна Македонија

ОПШТ МИКРОПРОЦЕСОР - ФАЗИ НА ИЗВРШУВАЊЕ НА ИНСТРУКЦИИ

АПСТРАКТ: Идентификуван е проблем на непостоење на единствен генерален модел на општ МКП којшто и од хардверски и од софтверски аспект е близок до некој од реалните МКП.

Во овој труд воведуваме оригинална 8-битна дигитална компонента како посебен модел на општ микропроцесор (ОМП), која и од хардверски и од софтверски аспект претставува комбинација на заедничките карактеристики на три реални 8-битни микропроцесори: Intel 8080, Intel 8085 и Zilog Z-80, но и нивно упростување.

Поконкретно, во овој труд се фокусираме на извршувањето на инструкциите во фази, при што во поголеми детали ја објаснуваме секоја од фазите низ која поминува тоа извршување.

КЛУЧНИ ЗБОРОВИ: општ микропроцесор, извршување на инструкција, фазите на извршување на инструкција

GENERAL MICROPROCESSOR - STAGES OF INSTRUCTION EXECUTION

ABSTRACT: A problem of non-existence of a single general model of general microprocessor has been identified, which, from a hardware and software aspect, is close to some of the real microprocessor. In this paper we introduce the original 8-bit digital component as a special model of general microprocessor, which, in terms of both hardware and software, is a combination of common features of three real 8-bit microprocessors: Intel 8080, Intel 8085 and Zilog Z-80, but also their simplification. More specifically, in this paper we focus on the stages of instruction execution, and explain them in details.

KEYWORDS: general microprocessor, instruction execution, stages of instruction execution

ВОВЕД

Новововедениот ОМП е разработен во неколку различни трудови и тоа од трите клучни аспекти кои се неопходно потребни за разбирање на неговото функционирање и употреба за формирање на МКП-систем:

- » од внатрешен аспект – неговата интерна архитектура: функционалната блок-шема и пин конфигурација,
- » од аспект на поврзување со полупроводнички мемориски интегрирани

- кола, како и со различни влезно/излезни порти, но и
- » од софтверски аспект за негово програмирање во асемблерски јазик.

Сличноста на овој ОМП со трите наведени реални МКП-и во однос на нивните заеднички хардверски и софтверски особини е навистина голема. Токму заради тоа, воведувањето на ваков оригинален модел на општ микро-процесор, разјаснет од страна на сите три аспекти, овозможува поедноставување на изучувањето и разбирањето на базичните принципи за функционирање на реалните МКП-и.

Освен научната јавност, бенефит од резултатите на овој труд ќе имаат и студентите од една страна, бидејќи им се олеснува совладувањето и разбирањето на базичните концепти во овој домен, а од друга страна професорите, бидејќи им се олеснува трансферот на знаења. На овој начин, студентите поедноставно ги стекнуваат неопходно потребните основи за архитектонската структура, функционирањето, поврзувањето во систем, како и програмирањето на реалните микропроцесори, кои би ги изучувале во понатамошната едукација.

Поконкретно, во овој труд се фокусираме на функционалната блок шема на ОМП со осврт на базичните градбени блокови, внатрешни и надворешни сигнали, аритметичко-логичката единица со придружните регистри, потоа регистрите за општа и специјална намена, како и на единицата за контрола на прекините и единицата за сериски влез/излез на податоци.

1. ИЗВРШУВАЊЕ НА ИНСТРУКЦИЈА

Како што е познато, МКП го изведува програмот на тој начин што редоследно ги извршува сите асемблерски инструкции кои тој ги содржи, а потекнуваат од множеството на инструкции со кои тој процесор располага. При тоа, секоја асемблерска инструкция, во општ случај, задолжително содржи мнемоник. Покрај мнемоникот, инструкцијата може да содржи еден или два операнди, кои следуваат со мнемоникот. Сепак, постојат и инструкции без операнд/и, и тие содржат само мнемоник бидејќи операндите им се испуштени. Мнемоникот претставува кратенка со одредено значење која потсетува на тоа која операција треба да ја изврши инструкцијата. Од друга страна, операндот се однесува на податокот со кој таа операција треба да се изврши.

Во врска со претходното, за да изврши некоја инструкция, ОМП прво од меморијата го презема првиот бајт којшто е *код на операција (КОП)*, а симболички е претставен со мнемоникот на таа инструкцијата. Потоа преземениот КОП процесорот внатрешно го декодира со инструкцискиот декодер и со тоа ОМП „дознава“ дали инструкцијата има и втор или трет бајт. Ако инструкцијата нема друг бајт, или бајти, ОМП ја извршува операцијата која е зададено со тој КОП. Но, ако инструкцијата покрај КОП има уште еден бајт како операнд, тогаш процесорот уште еднаш му пристапува на меморијата и го чита вториот бајт на инструкцијата кој претставува или директен податок кој треба да се обработи, или адреса на некоја влезна порта од каде тој податок ќе биде добиен. Ако пак, инструкцијата покрај КОП содржи уште два бајти, тогаш процесорот уште два пати чита од меморијата и со тоа добива директно податок кој е дво-бајтен, или добива адреса на мемориска локација каде се наоѓа бараниот податок и истиот го чита. Дури после тоа ОМП завршува со извршувањето на инструкцијата и истата ја комплетира.

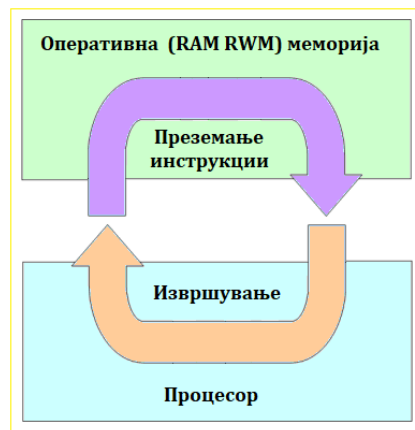
Заклучокот се наметнува сам по себе: секоја инструкция како прв бајт има еднозначно определен КОП. Покрај едно-бајтните инструкции, постојат и инструкции со должина два или најмногу три бајти. Кај нив вториот и/или третиот

бајт содржи податок кој треба да биде обработен со операцијата содржана во КОП, или пак содржи адреса каде тој податок се наоѓа.

2. ФАЗИ НА ИЗВРШУВАЊЕ НА ИНСТРУКЦИЈА

Како и секој реален процесор, така и нашиот ОМП има свое посебно инструкциско множество, а тоа се сите негови инструкции во асемблерски, односно машински јазик, со кои тој располага и може да ги изврши. Процесорот секоја инструкција ја извршува, т.е. процесира во временски период дефиниран како инструкциски циклус, и тоа во две фази според сл.1 а):

- I. фаза на преземање на инструкцијата,
- II. фаза на извршување на инструкцијата.



Сл.1 а) Фази на извршување на инструкција

Глобално гледано, секоја од двете фази на извршување дополнително се дели на потфази, како што е прикажано на сл.1 б)



Сл. 1 б) Потфази за извршување на инструкција

Фазите и потфазите на извршување на една инструкција се:

I) Фазата на преземање се реализира во две потфази кои се исти за секоја инструкција, а тие се: читање на првиот бајт на инструкцијата кој го претставува нејзиниот КОП, декодирање на КОП.

1. читање на првиот бајт на инструкцијата кој го претставува нејзиниот КОП,
2. декодирање на КОП.

II) Фазата на извршување на инструкцијата содржи три потфази кои се различни за секоја инструкција.

3. земање на операндот од меморијата,
4. извршување на инструкцијата,
5. запамтување на резултатот.

3. ОБЈАСНУВАЊЕ НА ФАЗИТЕ НА ИНСТРУКЦИЈАТА

Во првата фаза, секогаш задолжително се реализираат и првата и втората потфаза при што прво се презема КОП после што следува негово декодирање. Доколку инструкцијата нема потреба од податок, таа се извршува во следната, втората фаза, и тоа само во нејзините четврта и петта потфаза. Имено, кај овие инструкции нема потреба од дополнително обраќање на ОМП кон меморијата или кон некоја влезна порта бидејќи нема потреба од податок кој се наоѓа надвор од процесорот. Исто така нема потреба од комуникација и тогаш кога инструкцијата има потреба од податок, но тој се наоѓа во регистрите на процесорот. Ваквите инструкции извршуваат само внатрешни машински операции кои може да бидат аритметички, логички, тестирање на некој услов, операции за внатрешен пренос на податоци меѓу регистрите и сл. и се завршуваат без третата под-фаза.

Следните три под-фази, се реализираат тогаш кога после декодирањето на КОП, се утврди дека инструкцијата има потреба од податок и ако тој податок се наоѓа надвор од процесорот, во некоја од неговите периферни компоненти како на пр. меморијата или некоја В/И порта. Кога извршува вакви инструкции ОМП ќе треба да реализира екстерни, надворешно ориентирани машински операции. При тоа, ако податокот се наоѓа во меморијата или во некој В/И модул, ОМП ја извршува третата под-фаза на земање на податок. Тогаш ОМП ќе изврши екстерна работна операција со која ќе прочита или прифати еднобајтен податок. Тој податок може да биде содржина на некоја мемориска локација, или да бил запамтен во стек магацинската меморија, или да е еден бајт добиен од некоја влезна компонента или порта. Потоа, имајќи го потребниот влезен податок, ОМП ќе премине на четвртата под-фаза извршување на инструкцијата.

Меѓутоа, ако дополнително во инструкцијата, покрај наведеното читање на влезен податок, се бара и запишување на резултантен податок во меморијата или тоа е податок наменет за во стек – меморијата, или некој бајт кој треба да се пренесе до адресирана излезна порта, ОМП ќе премине на петтата под-фаза на запамтување на резултатот.

Значи, инструкцијата ќе биде извршена од страна на МКП само ако КОП на инструкцијата биде пре/земен од главната RAM меморија, а заедно со него, доколку постои, биде прочитан и податокот од меморијата или од В/И порта, кој ќе биде со неа обработен. Од кажаното станува јасно дека за секоја инструкција

времето што е потребно за нејзино извршување варира и тоа е различно од инструкција до инструкција.

3.1. Фаза– 1: Пре/земање на кодот на операција од оперативната меморија

Процесорот ја става вредноста на програмскиот бројач PC во меморискиот адресен баферски регистер MAR, во него ја задржува и со тоа ја става на адресната магистрала со што адресира одредена мемориска локација од оперативната RAM меморија. Содржината на адресираната мемориска локација е кодот на операција на инструкцијата, кој е нејзин прв бајт што процесорот го чита и го презема преку податочната магистрала сместувајќи во својот мемориски податочен бафер регистер MDR. Добиениот КОП понатаму се пренесува до инструкцискиот регистер IR каде што се сместува и привремено се задржува за да може да биде декодиран од страна на декодерот на инструкции.

Едновремено со пренесувањето на содржината на програмскиот бројач во MAR, процесорот ја зголемува содржината на програмскиот бројач за 1 како би можел да ја чита следната мемориска локација.

3.2. Фаза– 2: Декодирање на кодот на операција

Инструкцискиот декодер IR го декодира кодот на операција на инструкцијата и врз основа на тоа кодира машински циклуси кои треба да бидат реализирани заради нејзино извршување и ја активира единицата за контрола и временска синхронизација која испраќа соодветни внатрешни и надворешни контролни сигнали до сите внатрешни функционални единици и надворешни компоненти за да го комплетира започнатото извршување на инструкцијата. Доколку инструкцијата е едно бајтна, т.е. содржи само код на операција следното читање од меморијата ќе биде пре/земање на код на операција на следната инструкција. Меѓутоа, ако инструкцијата содржи два или три бајти, ОМП преминува во третата фаза на земање на потребниот податок за комплетирање на инструкцијата.

3.3. Фаза– 3: Земање на податок од меморијата

Во оваа фаза се чита операндот на инструкцијата кој е нејзин втор бајт и кој може да биде податок кој ќе биде обработен со операцијата која се извршува, или адреса на мемориска локација или В/И порта каде што се наоѓа тој податок. Прочитаниот бајт се сместува во некој од регистрите на процесорот, што зависи од конкретната инструкција што се извршува, но никако во инструкцискиот регистер. Доколку инструкцијата има и трет бајт, кој исто така е операнд, после вториот, и овој бајт процесорот ќе го прочита од следната мемориска локација. По ова ОМП може да премине на следната фаза во процесирањето на инструкцијата, а тоа е фазата на нејзино извршување.

3.4. Фаза– 4: Извршување на инструкцијата

За да влезе во ова фаза, кодот на операција од IR е веќе декодиран од страна на декодерот на инструкции и од него веќе се испратени соодветни сигнали до контролната единица. Врз основа на добиените декодирани сигнали контролната единица генерира и испраќа секвенца (низа) од дополнителни внатрешни и/или надворешни контролни сигнали кон внатрешните логички и секвенцијални кола и/или надворешните компоненти: мемориските интегрирани кола или В/И порти. Тоа се сигнали испратени до релевантните

функционални единици на процесорот, за да ги превземат оние активности кои се потребни за извршување на инструкцијата. Тоа се регистрите и/или АЛЕ кои треба да се вклучени на пр. за извршување на некоја аритметичка или логичка операција со внесениот податок, или читање на вредности од регистри, или нивно префрлување во други, или пренесување на вредности до АЛЕ, или запишување на добиени резултати од АЛЕ во други регистри, и тн. Доколку врз основа на извршената операција од страна на АЛЕ, кога е таа вклучена, се исполнил некаков услов како на пр. резултат кој е нула, или е негативен, или има пренос, до контролната единица ќе биде испратен соодветен сигнал т.н. знаменце кој покажува каква е состојбата по завршената операција.

3.5. Фаза– 5: Запомнување на резултатот

По извршувањето на било која аритметичка или логичка операција ОМП резултатот добиен од неа го пренесува во акумулаторот. Токму заради тоа што резултатот се сместува во акумулаторот, кај ОМП може да се земе дека 4-тата и 5-тата фаза може да се третираат заеднички како единствена 4-та фаза – фаза на извршување во која се подразбира дека извршувањето опфаќа и внесување на резултатот од операцијата во акумулаторот. Заради ова, за општиот микропроцесор може да се каже дека е акумулаторски ориентиран.

По завршувањето на оваа последна фаза процесорот циклично ги повторува сите фази повторно почнувајќи од првата фаза кога зема код на операција од следната мемориска адреса бидејќи програмскиот бројач е веќе зголемен за 1.

Доколку ОМП извршува инструкција која проверува исполнување на одреден услов, тој го тестира резултатот добиен по операцијата извршена од страна на АЛЕ кој е сместен во акумулаторот. Ако при тоа се исполнил тестираниот услов, на пр. резултатот е нула или има пренос, тогаш во програмскиот бројач ќе се внесе нова адреса на онаа мемориска локација каде што е сместена бараната инструкција соодветна за исполнување на поставениот услов. И во овој случај се повторуваат сите фази почнувајќи од првата кога процесорот го чита и презема кодот на операција на новоадресираната инструкција.

Од претходното може да се заклучи дека должината на инструкциите изразена во бајти, што се нарекува и формат на инструкција, може да биде еден, два, или три бајти. Покрај тоа е јасно дека брзината или времето на извршување на инструкциите со различен формат ќе биде исто така различна затоа што:

- » еднобајтните инструкции ѝ пристапуваат само еднаш на меморијата кога процесорот го зема единствениот бајт, а тоа е кодот на операција на инструкцијата,
- » двобајтните инструкции во два наврати ѝ пристапуваат на меморијата,
- » три-бајтните инструкции бараат уште едно – трето дополнително обраќање кон меморијата.

4. ПРИМЕР ЗА ИЗВРШУВАЊЕ НА ИНСТРУКЦИЈА

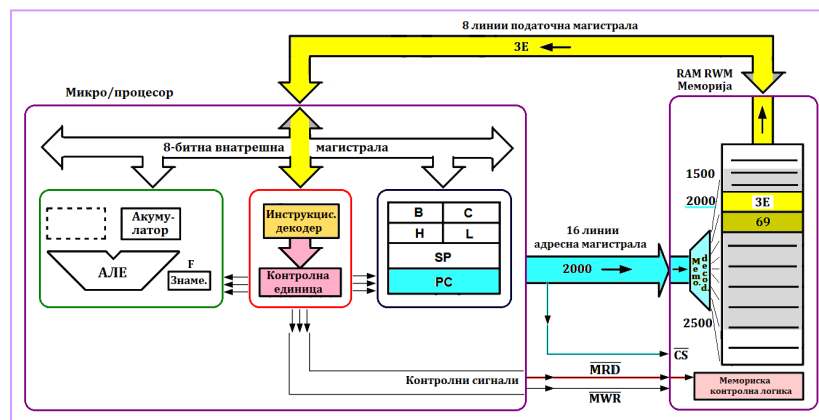
Како едноставен пример ќе ја анализираме извршување на една асемблерска инструкција од некој програм. Да земеме дека тоа е дво-бајтната асемблерската инструкција MVI A, 69H со која ОМП треба податокот 69H да го внесе во акумулаторскиот регистар. На сликите сл. 3 а) б) и сл. 4 а) б) на два различни начини е претставено извршувањето на двете фази на оваа инструкција (MVI A, 69H).

- » првиот бајт на анализираната инструкција е нејзиниот код на операција 3E кој е запамтен во мемориска локација со адреса 2000H, додека
- » вториот бајт на инструкцијата е непосредниот податок 69H, кој треба да биде со неа обработен и е сместен веднаш под КОП, како содржина на следната мемориска локација со адреса 2001H.

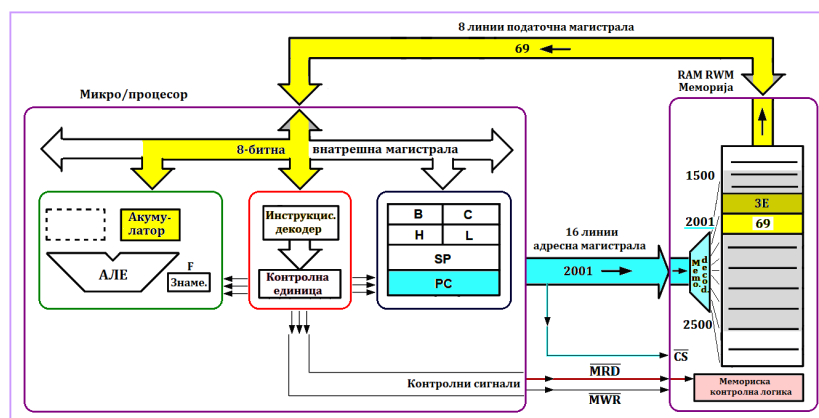
Во практиката меѓутоа, МКП извршува и тро-бајтни инструкции кои покрај кодот на операција како прв бајт содржат уште два бајти со кои се специфицира податокот кој ќе биде обработен со тој КОП. За да може нив да ги изврши процесорот освен првото обраќање кон оперативната меморија кога го чита КОП, на меморијата и пристапува уште два пати.

Покрај тробајтни инструкции во множеството на инструкции на еден МКП има и едно-бајтни инструкции. Тоа се такви инструкции кои се составени само од кодот на операција и кои ОМП ги извршува само со едно читање на меморијата кога го презема КОП на инструкцијата.

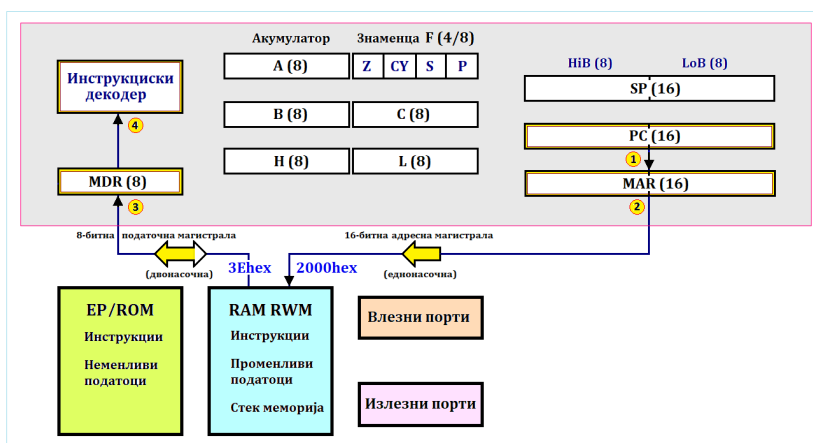
Овој принцип на работа се однесува и на извршувањето на секоја друга инструкција од програмот кој е запамтен во оперативната меморија. На овој начин, извршувајќи инструкција-по-инструкција од програмот кој го работи, МКП практично го извршува целиот програм.



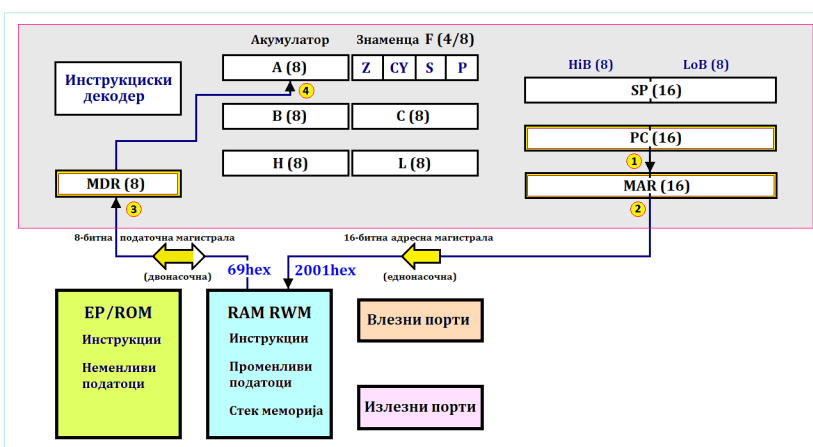
Сл. 3 а) Прва фаза на извршување на инструкцијата MVI A, 69H-детално



Сл. 3 б) Втора фаза на извршување на инструкцијата MVI A, 69H-детално



Сл. 4 а) Прва фаза на извршување на инструкцијата MVI A, 69H - едноставно



Сл. 4 б) Втора фаза на извршување на инструкцијата MVI A, 69H - едноставно

ЗАКЛУЧОК

Со овој труд воведовме оригинален модел на 8-битен општ микропроцесор (ОМП), која претставува комбинација од заедничките хардверски и софтверски карактеристики со истовремено поедноставување на некои од нив, на три реални 8-битни микропроцесори: Intel 8080, Intel 8085 и Zilog Z-80.

Поконкретно, во овој труд се фокусираме на извршувањето на инструкциите во фази и тоа во поголеми детали објаснувајќи ја секоја од фазите низ кои поминува тоа извршување.

ЛИТЕРАТУРА

1. D. K. Kaushik. (2010) AN INTRODUCTION TO MICROPROCESSOR 8085. Dhanpat Rai Publishing co., New Delhi, India. https://www.researchgate.net/publication/264005162_An_Introduction_to_Microprocessor_808
2. Intel Corporation. (1980). 8080/8085 MACRO ASSEMBLER OPERATOR'S MANUAL, Intel Corporation, Santa Clara, California, USA. http://www.bitsavers.org/pdf/intel/ISIS_II/9800292-04D_ISIS_II_8080_8085_Macro_Assembler_Operators_Guide_Aug81.pdf
3. Intel Corporation. (1978). 8080/8085 ASSEMBLY Language Programming Manual, Intel Corporation, Santa Clara, California, USA. http://bitsavers.trailing-edge.com/components/intel/8085/9800301C_8080_8085_Assembly_Language_Programming_Manual_Nov78.pdf
4. John Crisp. (2004, 2nd Edition.). Introduction to Microprocessors and Microcontrollers. Newnes Elsevier. Oxford, Massachusetts, USA. <https://www.pdfdrive.com/introduction-to-microprocessors-and-microcontrollers-d158887842.html>
5. Microprocessor Fundamentals.pdf <https://www.pdfdrive.com/microprocessor-fundamentalspdf-d41485263.html>
6. MICROPROCESSOR 8085 (Highlited extracts) <https://inspirit.net.in/books/academic/8085%20Microprocessor%20-%20Ramesh%20Gaonkar.pdf>