Кеш меморија

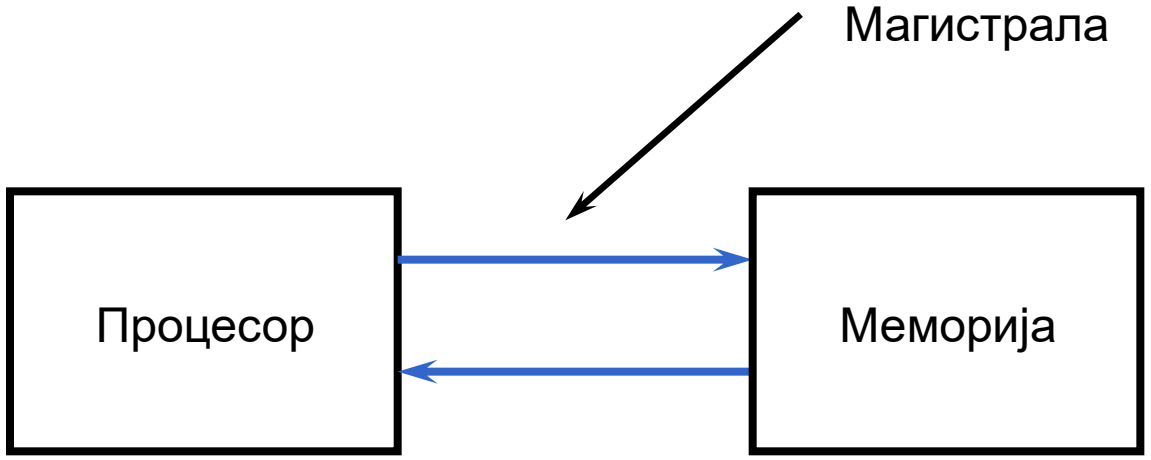
Садржај

- Увод
- Кеш меморија
- Технике пресликавања
- Алгоритми замене
- Ажурирање оперативне меморије
- Реализација
- Перформансе

Структура рачунара

- Рачунари су електронски уређаји у којима се решавање одређених проблема реализује извршавањем одређеног скупа аритметичких, логичких и померачких операција.

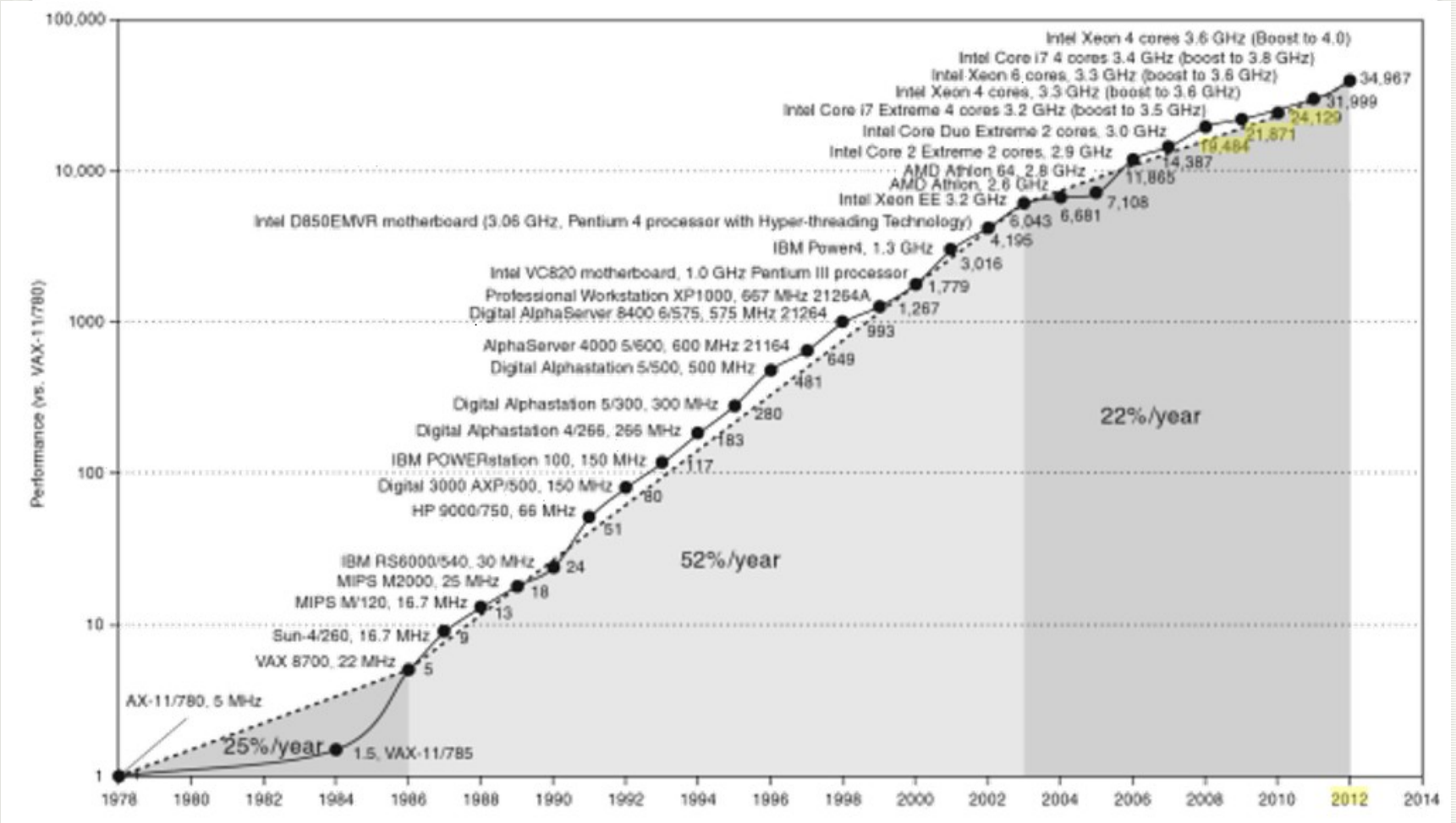
Једноставна организација



Процесор

- Операције које се у рачунару извршавају се представљају помоћу бинарних речи које се називају инструкције, команде или наредбе.
- Скуп операција које рачунар може да извршава је такав да било који проблем који треба да се решава у рачунару може да се разложи на уређени низ инструкција рачунара који се назива програм.
- Подаци над којима се операције извршавају се, такође, представљају помоћу бинарних речи које се називају операнди.

Процесор

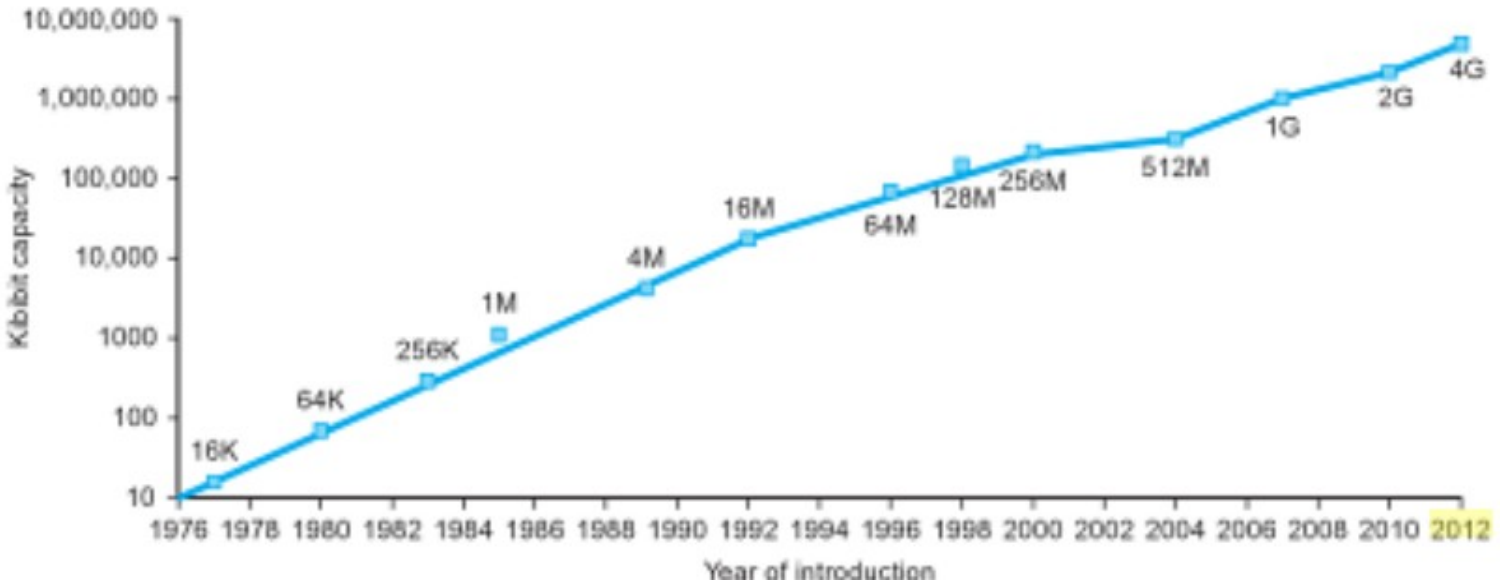


Patterson, Hennessy - Computer Organization and Design: The Hardware-Software Interface, 5Th edition.

Меморија

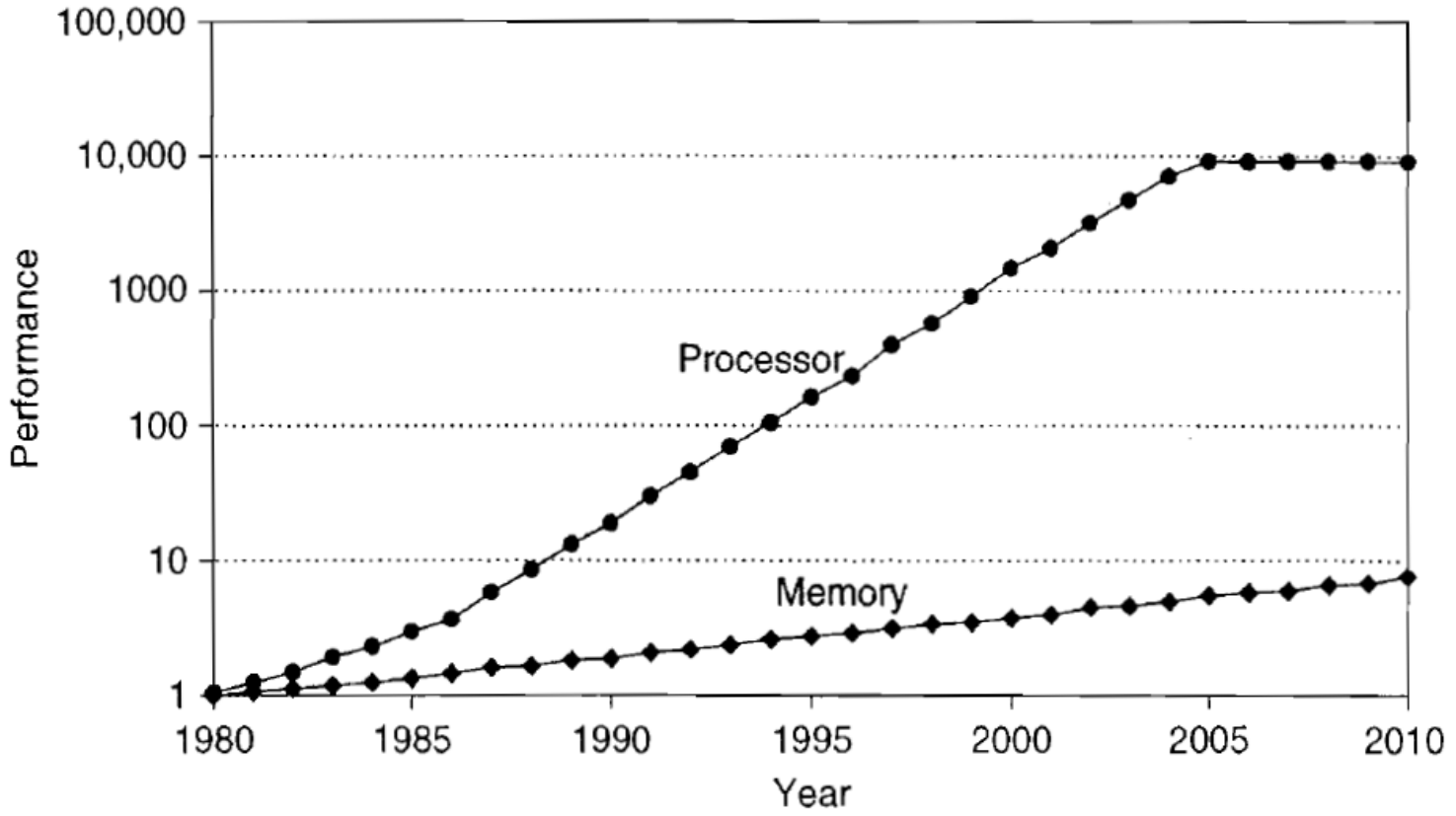
- За складиштење бинарних речи користи се модул рачунара који се назива меморија.
- Чување програма и података
- Организована на нивоу:
  - Бита
  - Бајтова = 8 бита
  - Речи = (типично 4 бајта)

Меморија



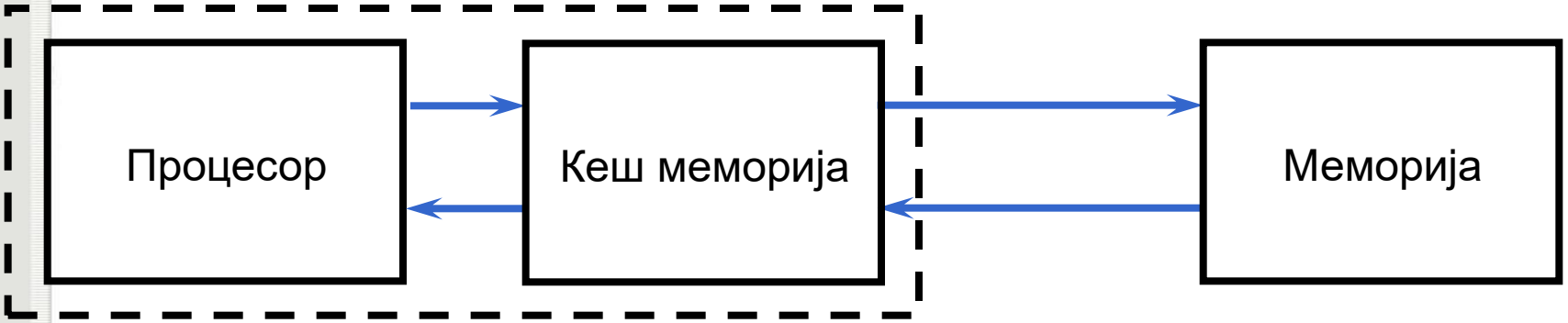
Patterson, Hennessy - Computer Organization and Design: The Hardware-Software Interface, 5Th edition.

Процесор/Меморија перформансе



Patterson, Hennessy - Computer Organization and Design: The Hardware-Software Interface, 5Th edition.

Једноставна организација 2



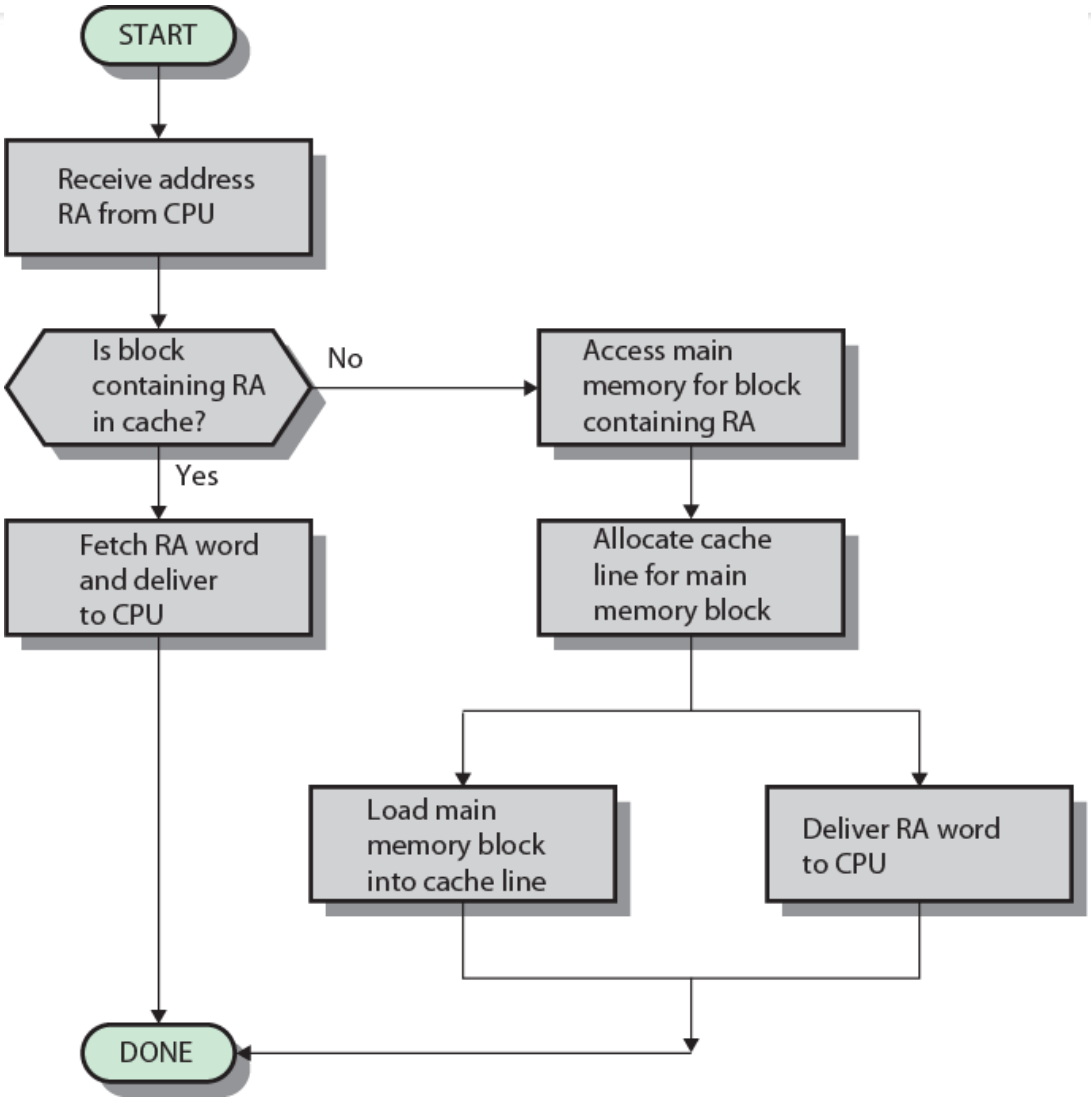
Кеш меморија

- Механизам кеш меморије подразумева да у процесору постоји посебна компонента - кеш меморија
- Кеш меморија се реализује са меморијским компонентама чије је време приступа:
  - **далеко мање** од времена приступа меморијских компонената оперативне меморије и
  - **веома блиско** времену преноса података између регистара.
- Због тога је цена по биту кеш меморије далеко већа од цене по биту оперативне меморије => Капацитет кеш меморије далеко мањи од капацитета оперативне меморије

Алгоритам - упрошћен

- При сваком генерисању адресе оперативне меморије од стране неког дела процесора ради читања или уписа:
  - врши се провера да ли се садржај са генерисане адресе налази у кеш меморији
  - уколико садржај није у кеш меморији онда се садржај са дате и неколико суседних адреса пребацује из оперативне у кеш меморију (блок)
  - садржај из кеш меморије се чита и прослеђује оном делу процесора који је адресу генерисао или уписује у кеш меморију садржај из оног дела процесора који је адресу генерисао

Алгоритам - упрошћен



Кеш меморија

- На почетку рада за неколико генерисаних адреса оперативне меморије вероватно ће се утврђивати да се садржаји не налазе у кеш меморији, па ће се одговарајући блокови довлачити из оперативне меморије у кеш меморију
- Повећавање броја довучених блокова повећава **вероватноћу** да се за неке од адреса утврди да се садржаји налазе у кеш меморији
- У свакој таквој ситуацији:
  - садржај ће се или читати из кеш меморије **уместо** из оперативне меморије или
  - уписивати у кеш меморију **уместо** у оперативну меморију.

Перформансе кеш меморије

Зависе од:

- времена приступа кеш меморији при сагласности (*hit time*),
- процента промашаја (*miss rate*) и
- просечног губитка времена при промашају (*miss penalty*)

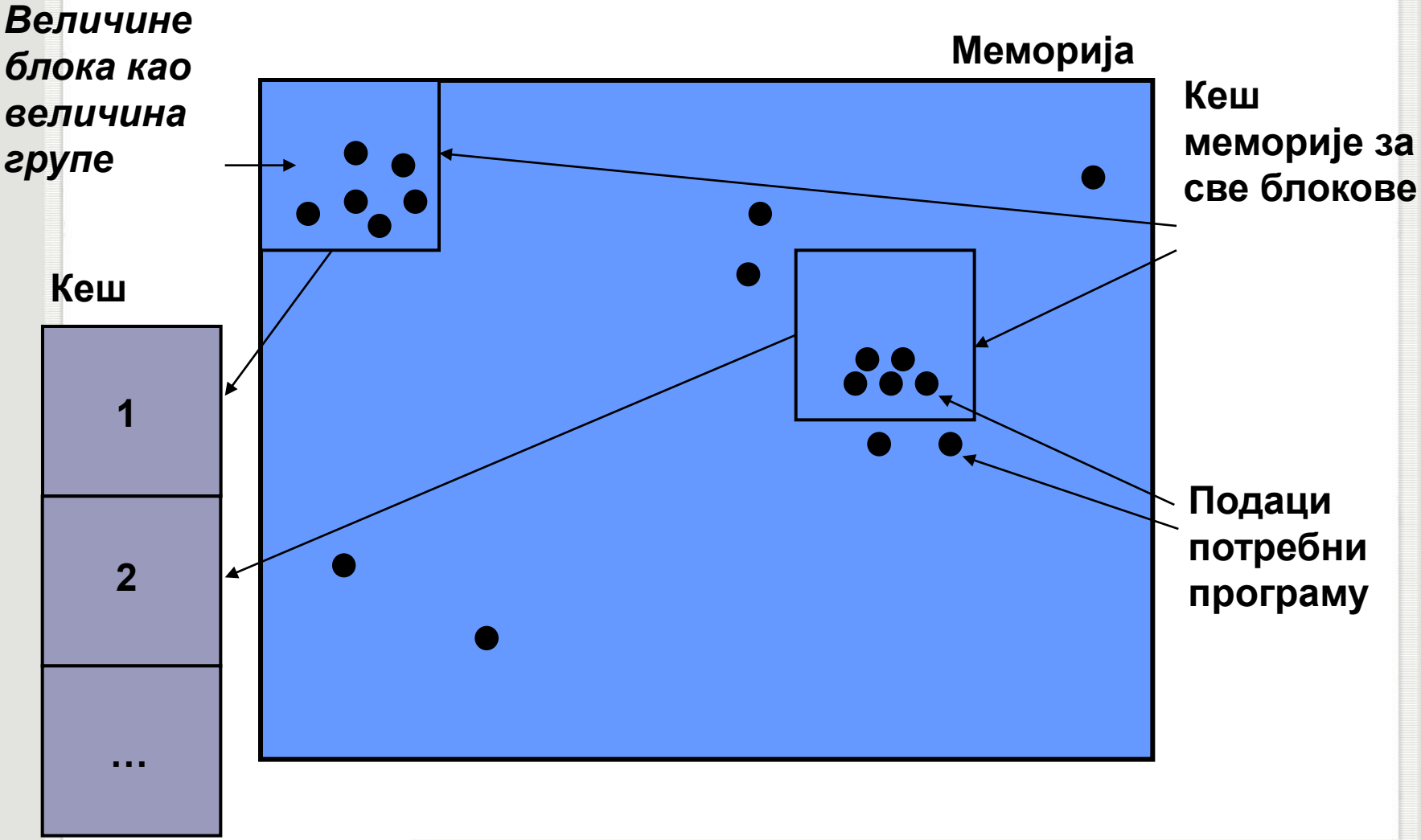
и дато је формулом:

$$t_{average\ memory\ access\ time} = t_{hit\ time} + miss\ rate * t_{miss\ penalty}$$

Кеш меморија

- Два ефекта који директно утичу на ефикасност механизма кеш меморије.
  - Једанпут генерисана адреса обично после тога још неколико пута се генерише. Овај ефекат се назива **временски локалитет** програма и јавља се код генерисања адреса инструкција и скаларних величина у петљи
  - После неке генерисане адресе веома често генеришу се адресе које следе секвенцијално. Овај ефекат се назива **просторни локалитет** програма и јавља се код секвенцијалног извршавања инструкција и секвенцијалног приступа подацима који представљају елементе вектора

Кеш меморија



Кеш меморија

- Механизам кеш меморије је "скривен" од програмера и програмер не може програмским путем да утиче на кеш меморију
- Провера да ли се садржај са генерисане адресе налази у кеш меморији, евентуално довлачење блока података из оперативне меморије у кеш меморију, читање податка из кеш меморије и упис податка у кеш меморију реализују се комплетно хардверски
- Због тога механизам кеш меморије не припада архитектури већ **организацији** процесора

Кеш меморија

- Капацитет кеш меморије је мањи од капацитета оперативне меморије и постоји потреба да се у кеш меморији води евиденција о томе који се блокови оперативне меморије налазе у кеш меморији и где се налазе у кеш меморији. Ово вођење евиденције се назива **техника пресликавања**.

Кеш меморија

- Капацитет кеш меморије је мањи од капацитета оперативне меморије, па ће се после одређеног времена дешавати да се генеришу адресе са којих се садржаји не налазе у кеш меморији, а кеш меморија је попуњена. Тада постоји потреба да се одлучи који ће се блок избацити из кеш меморије да би се у њој створио простор за довлачење блока из оперативне меморије коме припада генерисана адреса.
  Ово одлучивање се реализује према неком од **алгоритама замене**.

Кеш меморија

- Приликом операција уписа и утврђивања да у кеш меморији постоји блок коме припада генерисана адреса, упис ће се реализовати у кеш меморију. Тиме се јавља разлика у вредности копије садржаја са одређене адресе у кеш меморији и у оперативној меморији. Начини реализације зависе од усвојене технике **ажурирања садржаја** оперативне меморије.

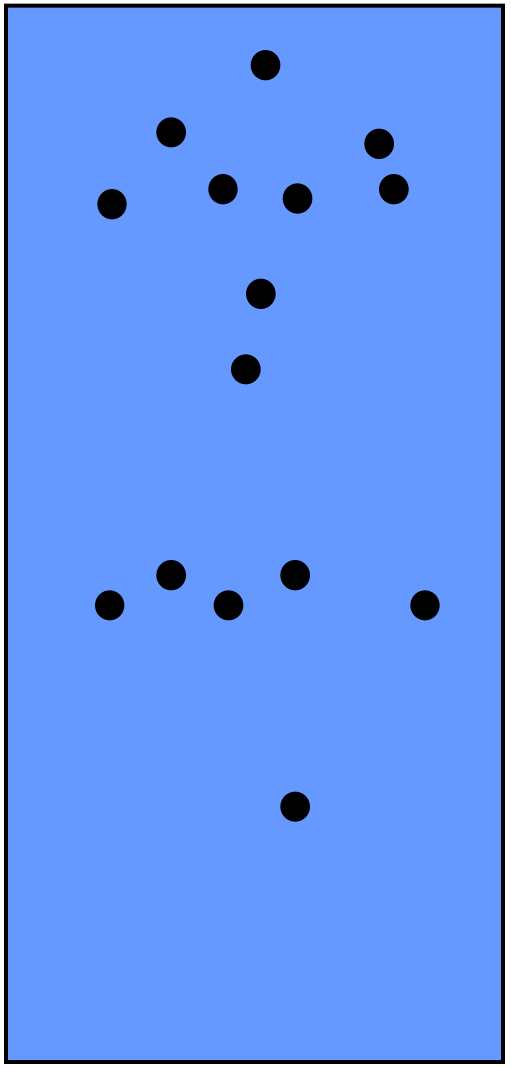
Технике пресликавања

- Техника пресликавања одређује начин вођења евиденције о томе који се блокови оперативне меморије налазе у појединим блоковима кеш меморије.
- Користе се три технике пресликавања и то:
  - асоцијативно,
  - директно и
  - сет-асоцијативно.

Асоцијативно пресликавање

Оперативна меморија

Генерисана адреса

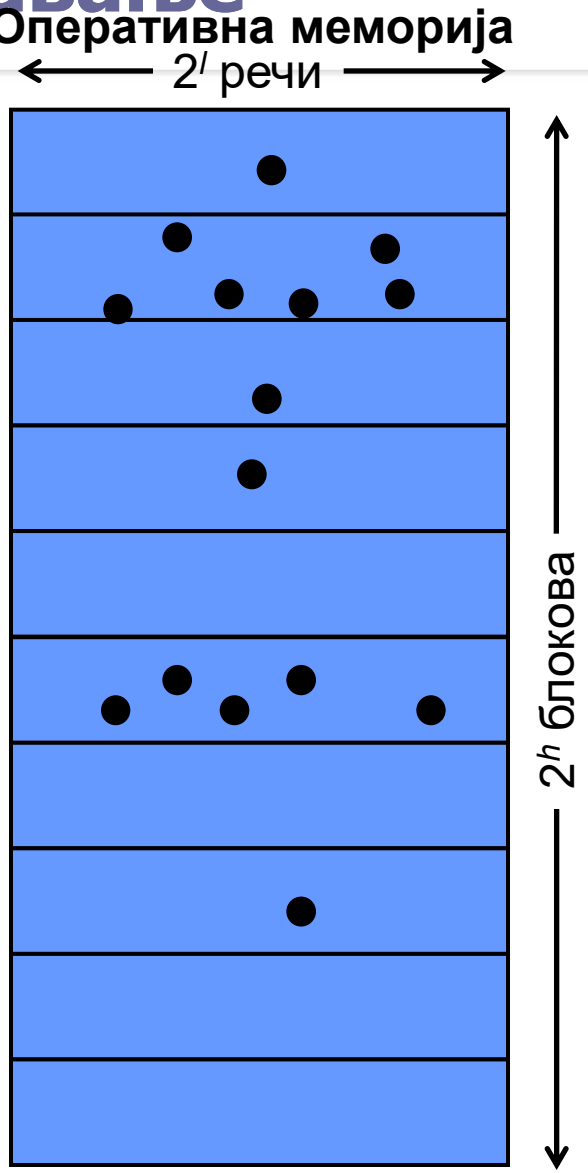


Асоцијативно пресликавање

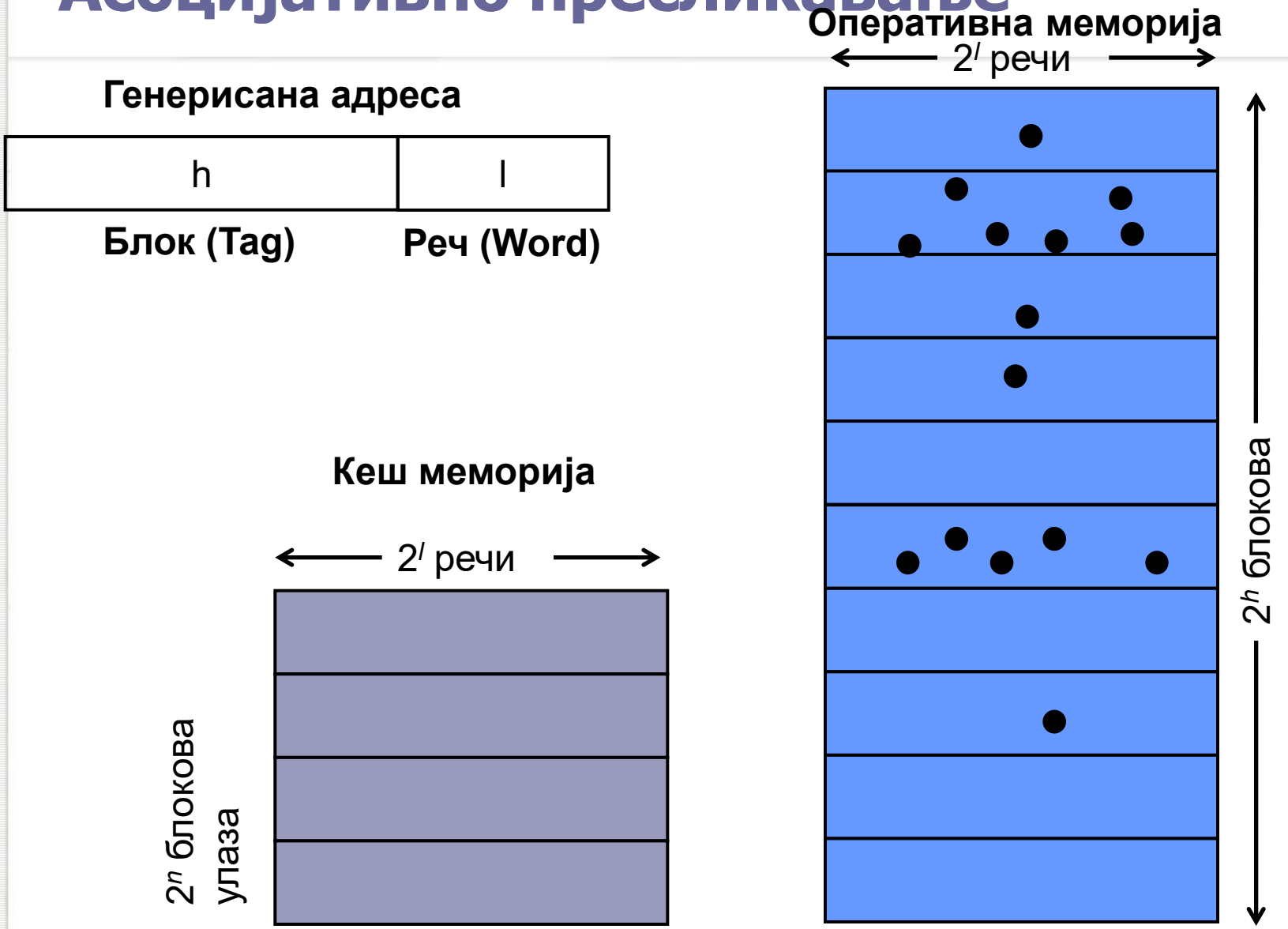
Генерисана адреса

Блок (Tag) **Реч (Word)**

Подела на блокове фиксне величине на основу најнижих l бита адресе

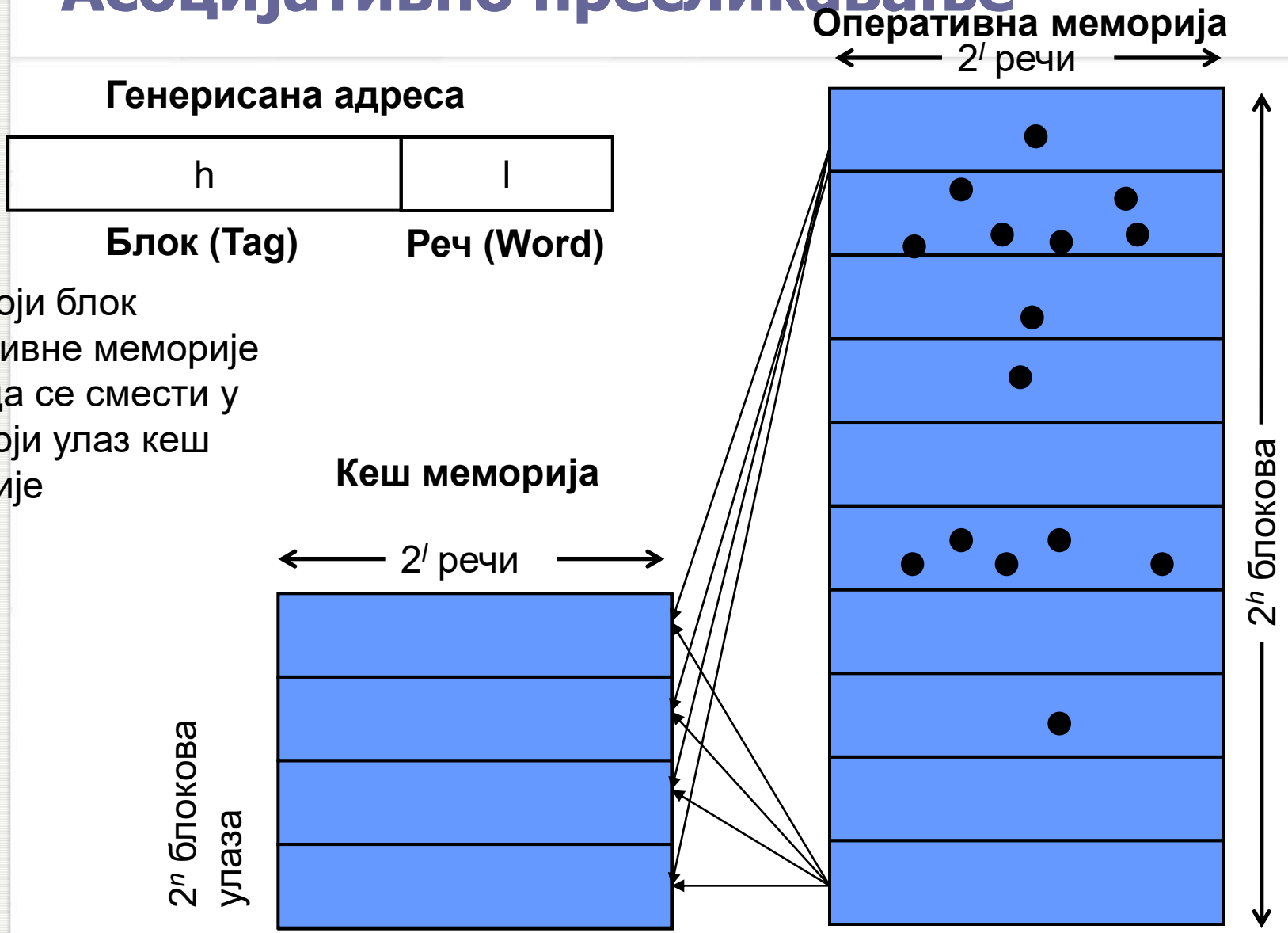


Асоцијативно пресликавање

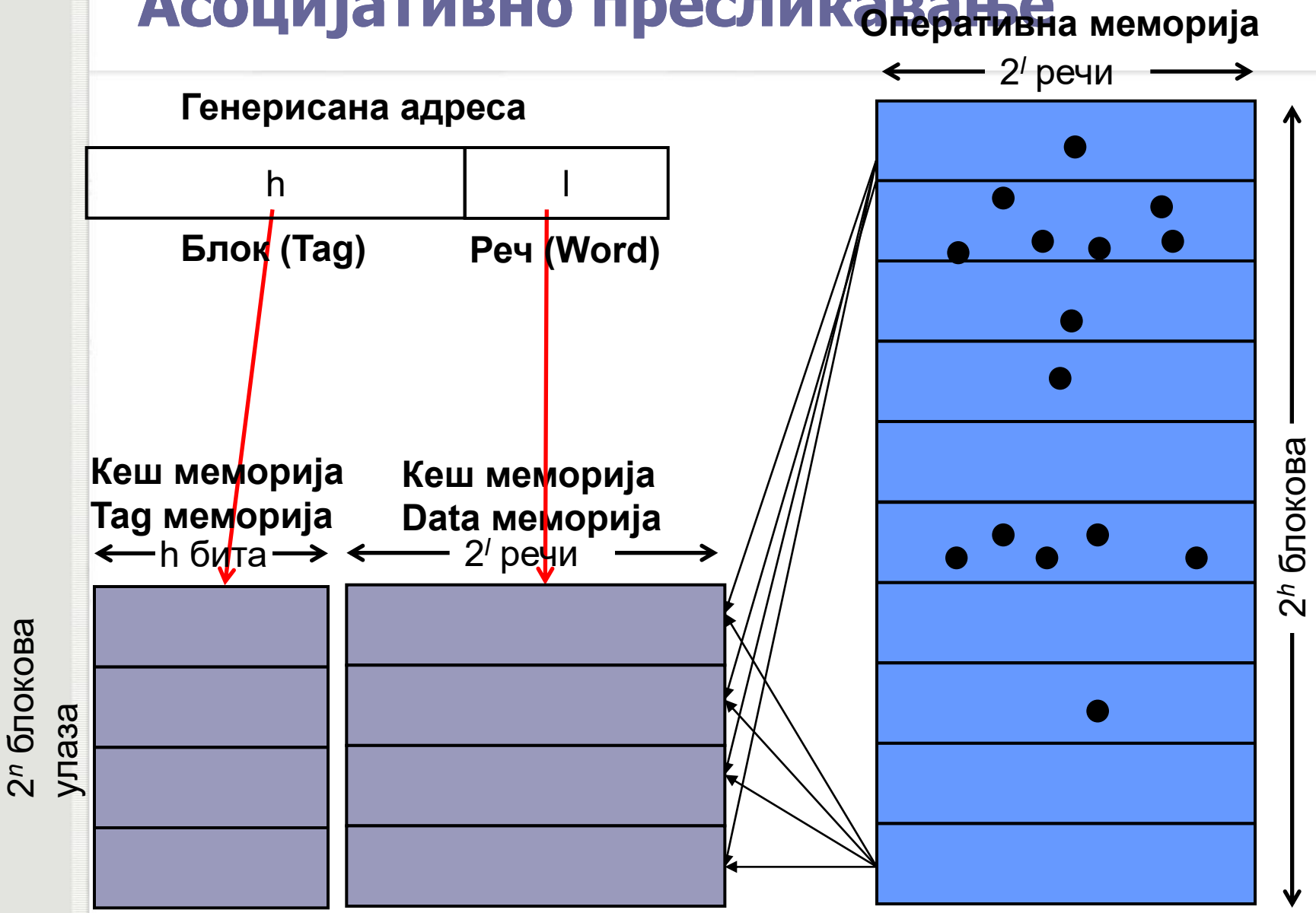


Асоцијативно пресликавање

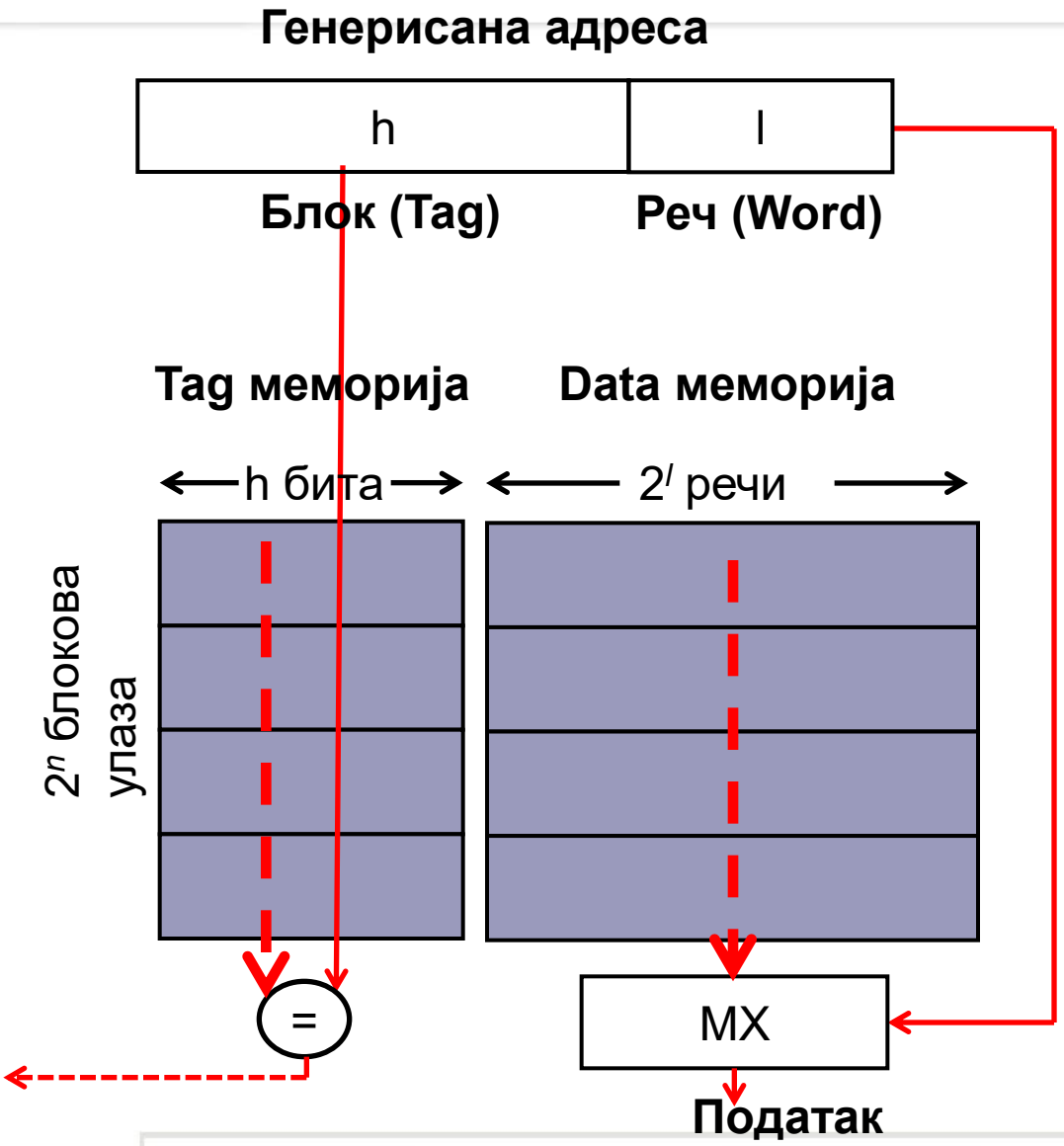
Било који блок оперативне меморије може да се смести у било који улаз кеш меморије



Асоцијативно пресликавање

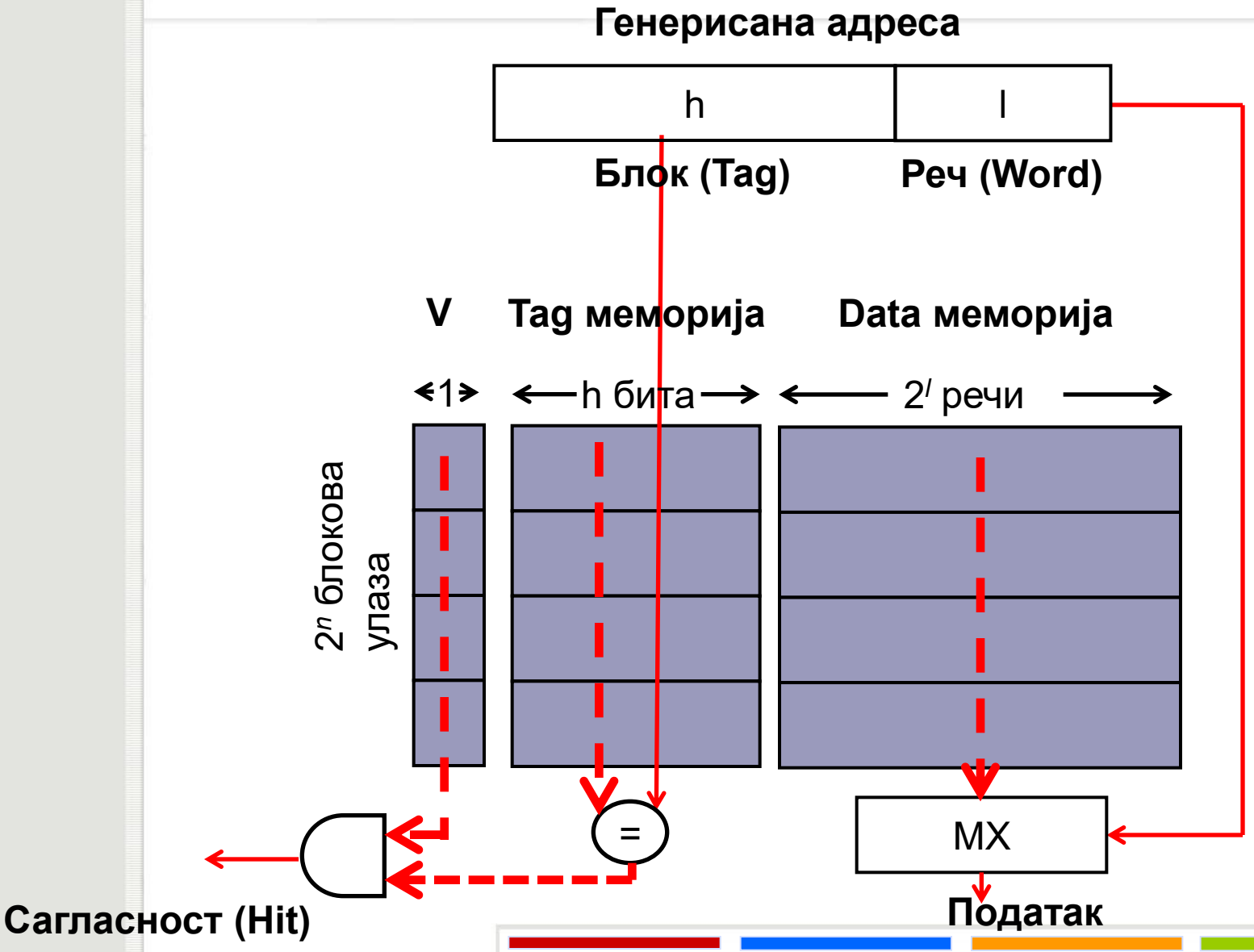


Асоцијативно пресликавање



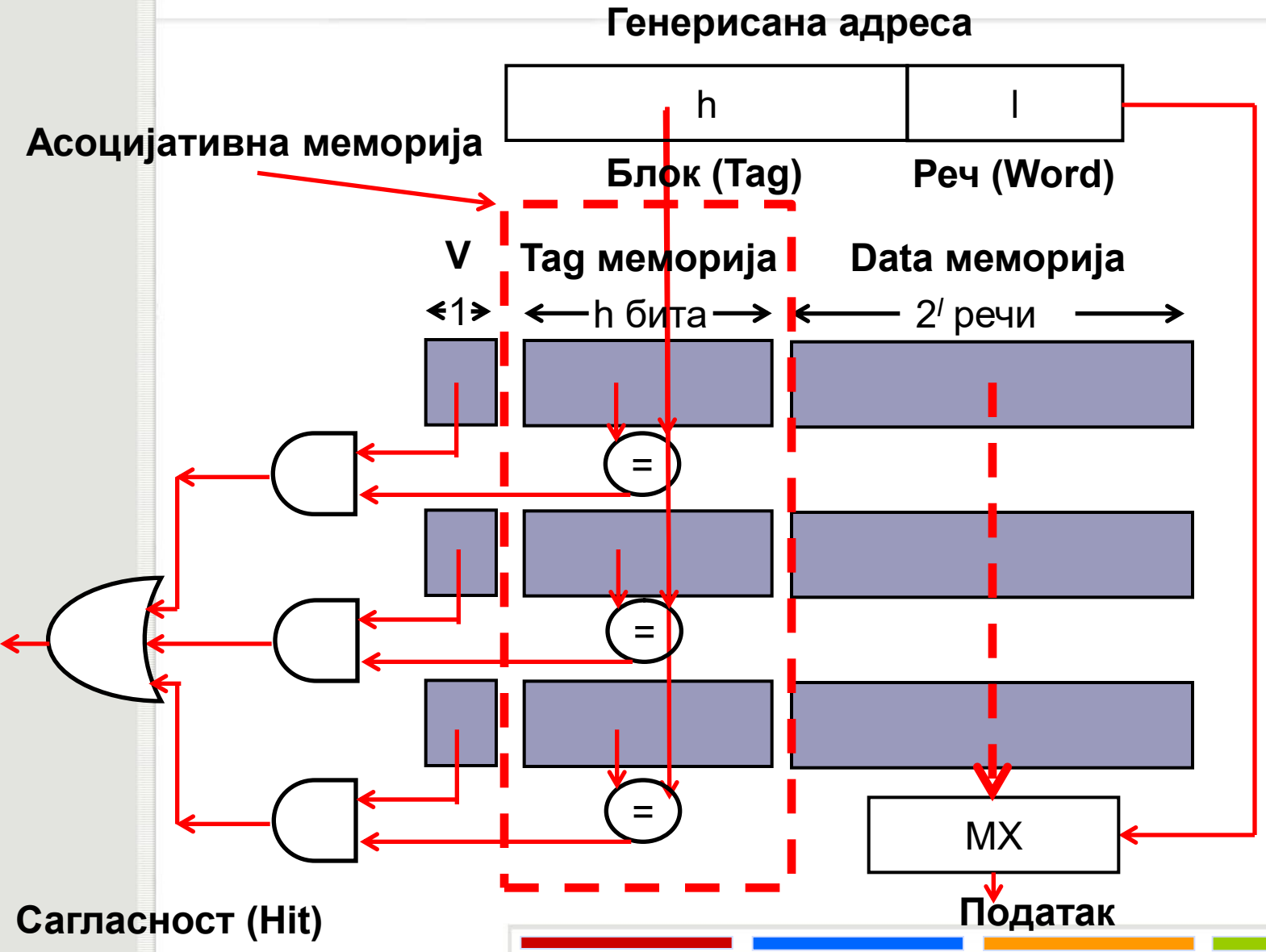
Сагласност (Hit)

Асоцијативно пресликавање

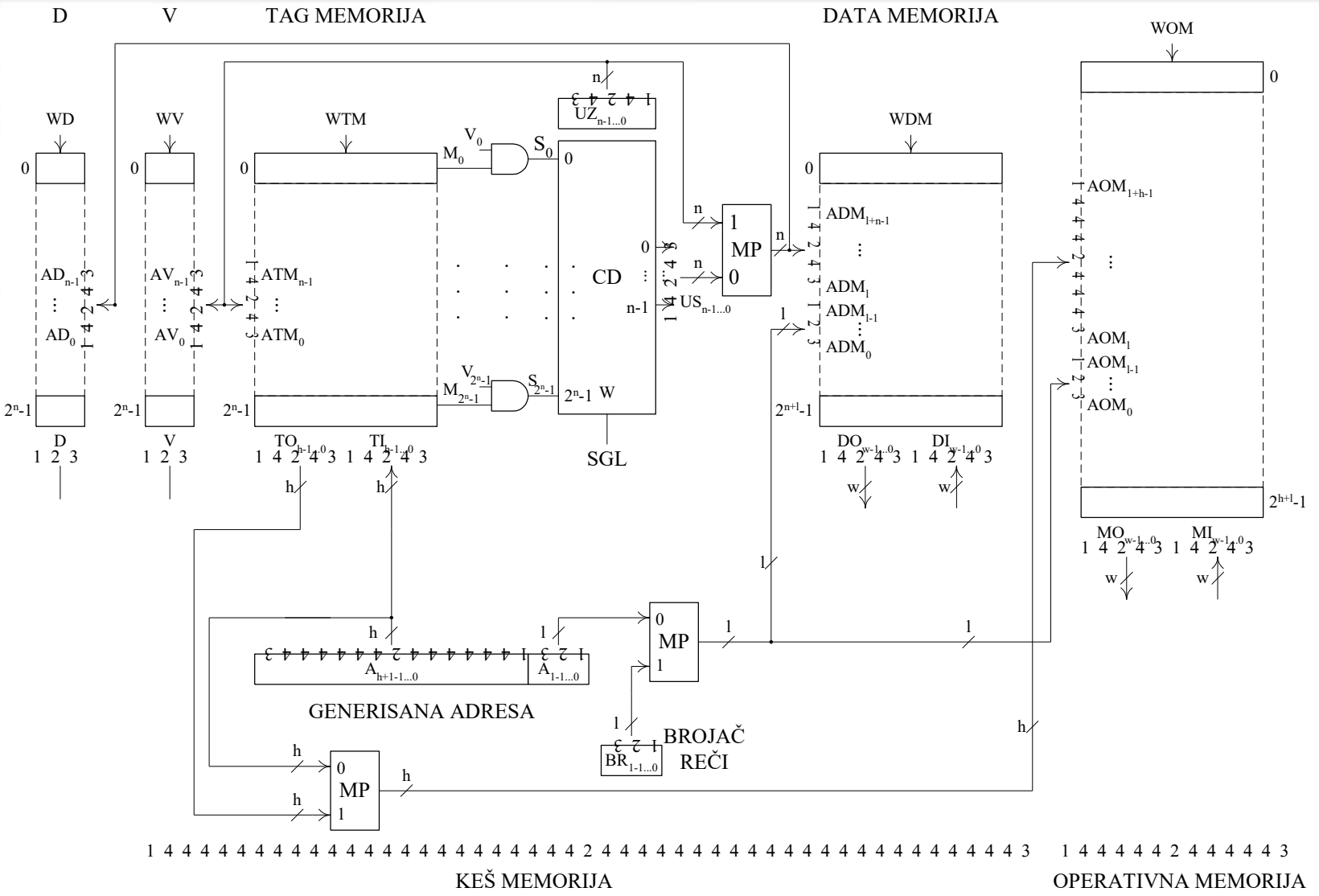


Одабира реч унутар блока, **не одабира** сам блок!

Асоцијативно пресликавање



Асоцијативно пресликавање



Асоцијативно пресликавање-алгоритам

Читање:

- Виших h битова адресе води се на $TI_{h-1...0}$
- **Истовремено** се упоређују са садржајима свих 2^n улаза TAG меморије
- Уколико се пронађе једнакост у неком улазу онда сигнал M_x добија вредност 1
- Уколико је и индикатор V_x датог улаза 1 онда је откривена сагласност важећа=>S_x је 1, односно
- Сигнал сагласности SGL (HIT) је такође 1
- Бинарна вредност броја улаза у коме је откривена сагласност је одређена са *n* бита са излаза кодера CD на основу сигнала S_0 до S_{2^n-1}.

Асоцијативно пресликавање-алгоритам

Читање, постоји сагласност:

- Уколико постоји сагласност, са *n* битова са излаза кодера CD и l нижих битова генерисане адресе, адресира се реч DATA меморије и обавља читање.

Асоцијативно пресликавање-алгоритам

Читање, нема сагласности:

- Довлачење блока из оперативне у кеш меморију у улаз за замену ($UZ_{n-1...0}$).
- Провера да ли се у улазу за замену налази модификован блок на основу индикатора D
  - Уколико је D=1, блок је модификован, па прво треба дати блок вратити у оперативну меморију па тек онда довући жељени блок.
  - Уколико је индикатор D=0, блок није модификован, па се жељени блок одмах довлачи.

Асоцијативно пресликавање-алгоритам

- Приликом враћања блока одабраног за замену 2^l речи датог блока се чита из DATA меморије са адреса формираних од вредности $UZ_{n-1...0}$ која даје *n* старијих битова адресе и вредности $BR_{l-1...0}$ која је даје *l* млађих битова адресе. Дате речи се уписују у оперативну меморију на адресама формираних од вредности на линијама $TO_{h-1...0}$ која даје *n* старијих битова адресе и вредности $BR_{l-1...0}$ која је даје *l* млађих битова адресе. Вредност на линијама $TO_{h-1...0}$ прочитана је из TAG меморије са адресе одређене вредношћу $UZ_{n-1...0}$.

Асоцијативно пресликавање-алгоритам

- Приликом довлачења жељеног блока 2^l речи датог блока се уписује у DATA меморију на адресама формираним од вредности $UZ_{n-1...0}$ која даје *n* старијих битова адресе и вредности $BR_{l-1...0}$ која је даје *l* млађих битова адресе. Дате речи се читају из оперативне меморије са адреса формираних од вредности битова $A_{h+l-1...l}$ генерисане адресе која даје *n* старијих битова адресе и вредности $BR_{l-1...0}$ која је даје l млађих битова адресе. Поред тога битови $A_{h+l-1...l}$ генерисане адресе се уписују у улаз TAG меморије чија је адреса одређена вредношћу $UZ_{n-1...0}$. Индикатори V и D улаза адресираних вредношћу $UZ_{n-1...0}$ постављају се на 1 и 0, респективно.

Асоцијативно пресликавање-алгоритам

Упис:

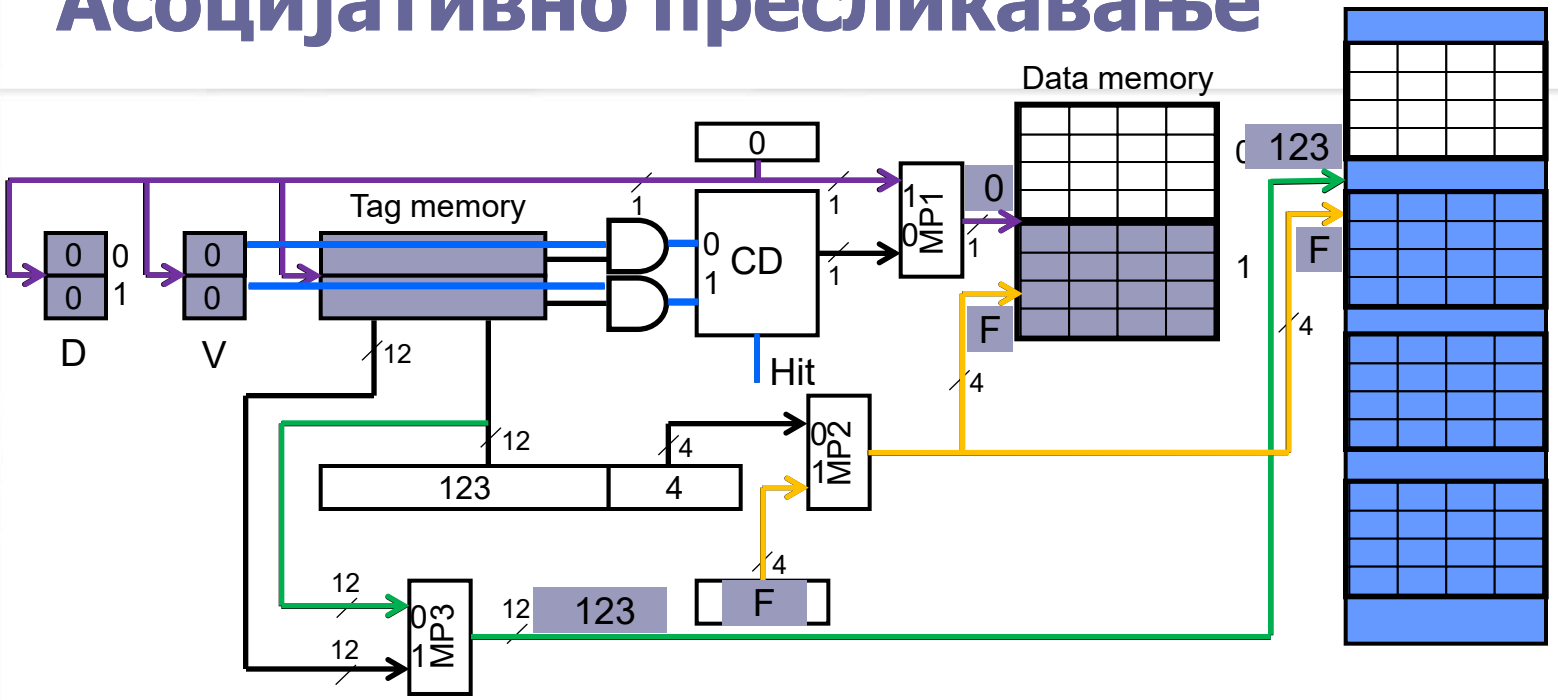
- При генерисању захтева за упис од стране процесора, на исти начин се испитује сагласност са садржајем кеш меморије као у случају операције читања. Уколико постоји сагласност, на исти начин се адресира реч DATA меморије и врши упис, при чему се сада индикатор модификованог улаза D, адресиран вредношћу $US_{n-1...0}$ са излаза кодера CD и која представља број улаза у коме је откривена сагласност, поставља на 1.

Асоцијативно пресликавање-алгоритам

Упис:

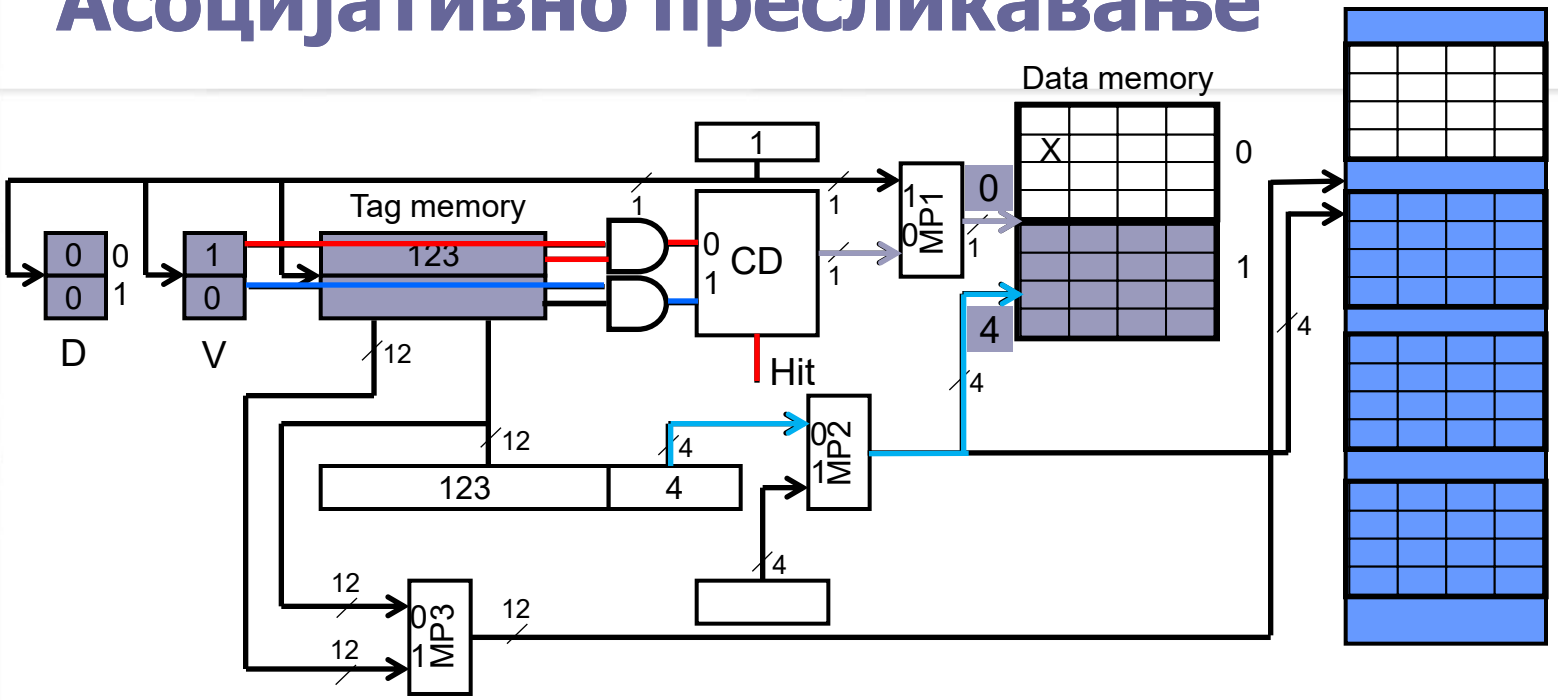
- Ако сагласност не постоји, на идентичан начин као и за операцију читања, се, најпре, блок из улаза кеш меморије, одређеног вредношћу $UZ_{n-1...0}$, враћа у оперативну меморију, уколико је модификован, а затим, у исти улаз кеш меморије, довлачи нови блок из оперативне меморије. Потом се поново, на већ описани начин, врши провера да ли постоји сагласност, утврђује да постоји сагласност и реализује упис.

Асоцијативно пресликавање



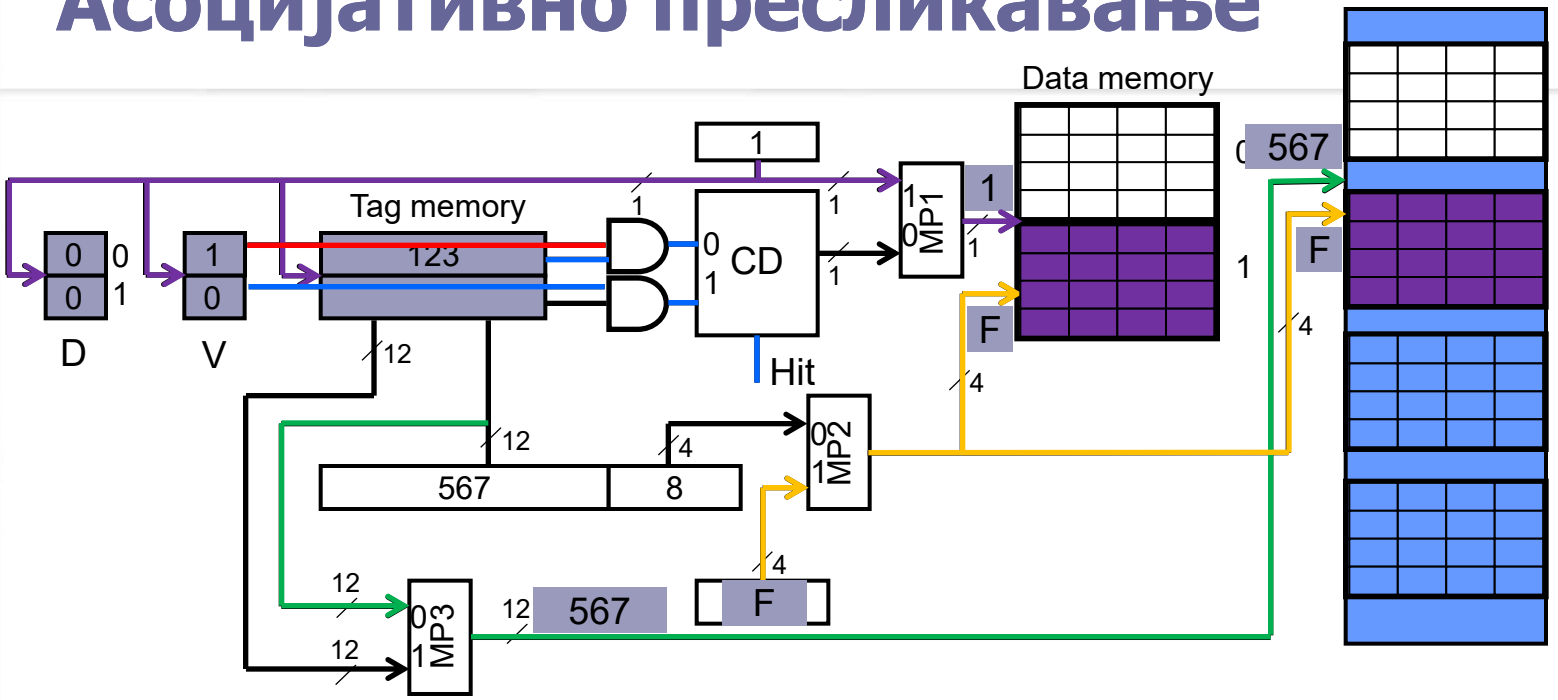
Адреса	Тип	Tag	Word	Hit?	Улаз КМ	ОМ адресе	КМ адресе

Асоцијативно пресликавање



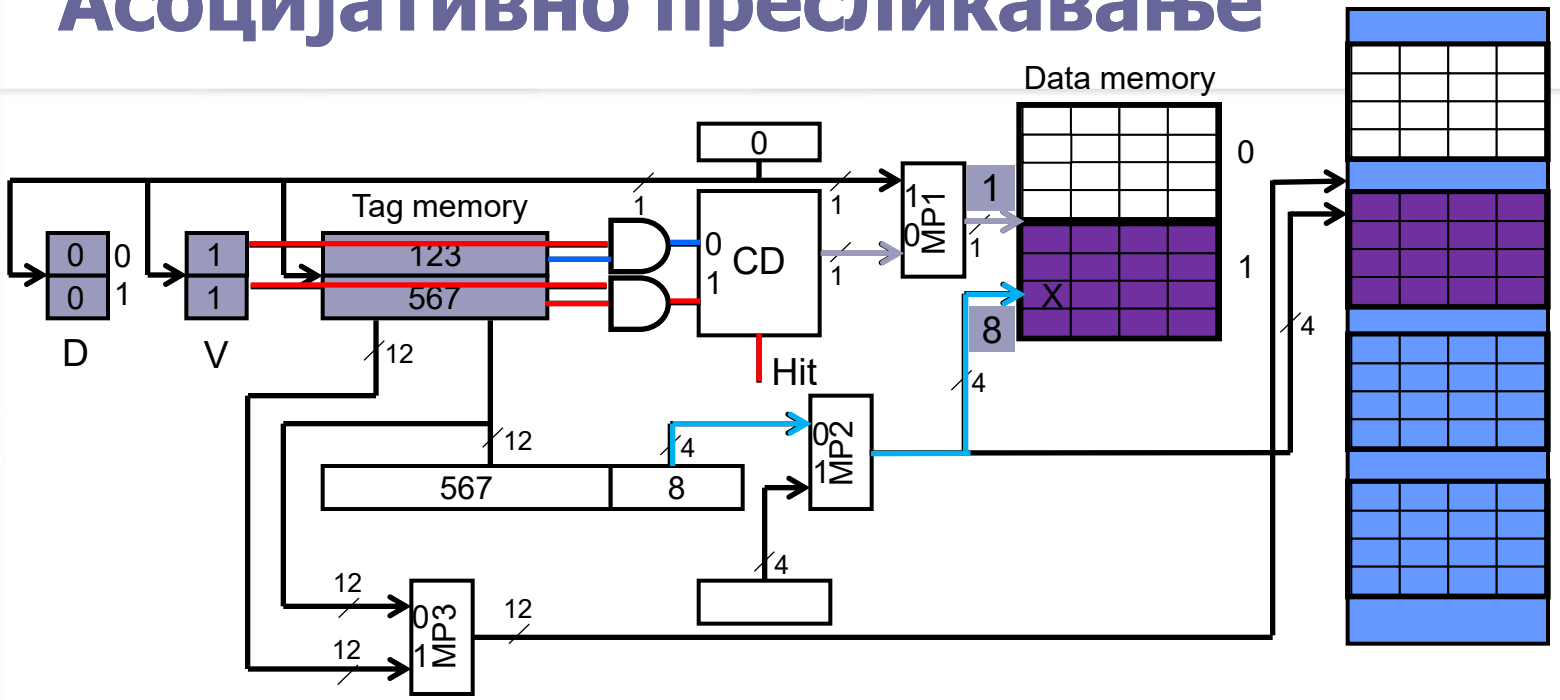
Адреса	Тип	Tag	Word	Hit?	Улаз КМ	ОМ адресе	КМ адресе
1234	Rd	123	4	Miss	0	[1230-123F]	[00-0F]

Асоцијативно пресликавање



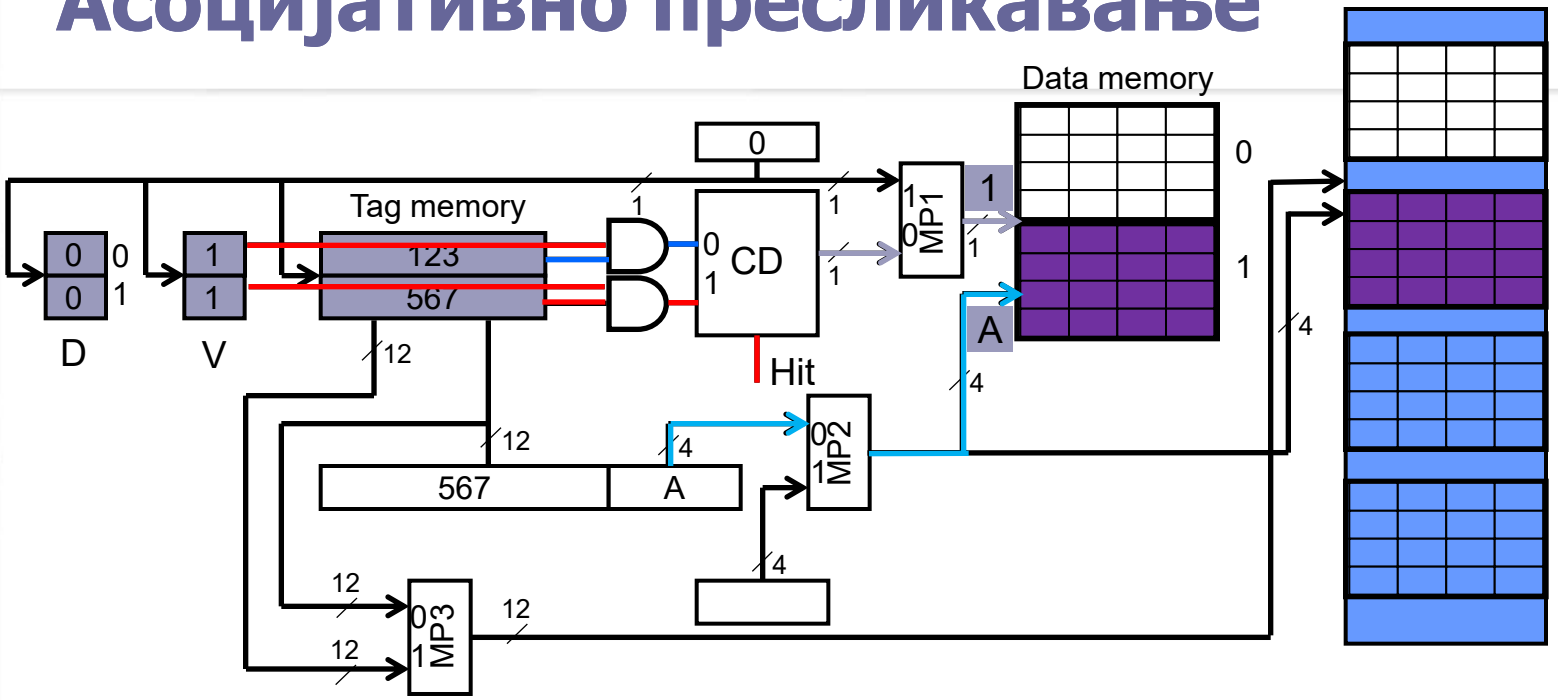
Адреса	Тип	Tag	Word	Hit?	Улаз КМ	ОМ адресе	КМ адресе
1234	Rd	123	4	Miss	0	[1230-123F]	[00-0F]

Асоцијативно пресликавање



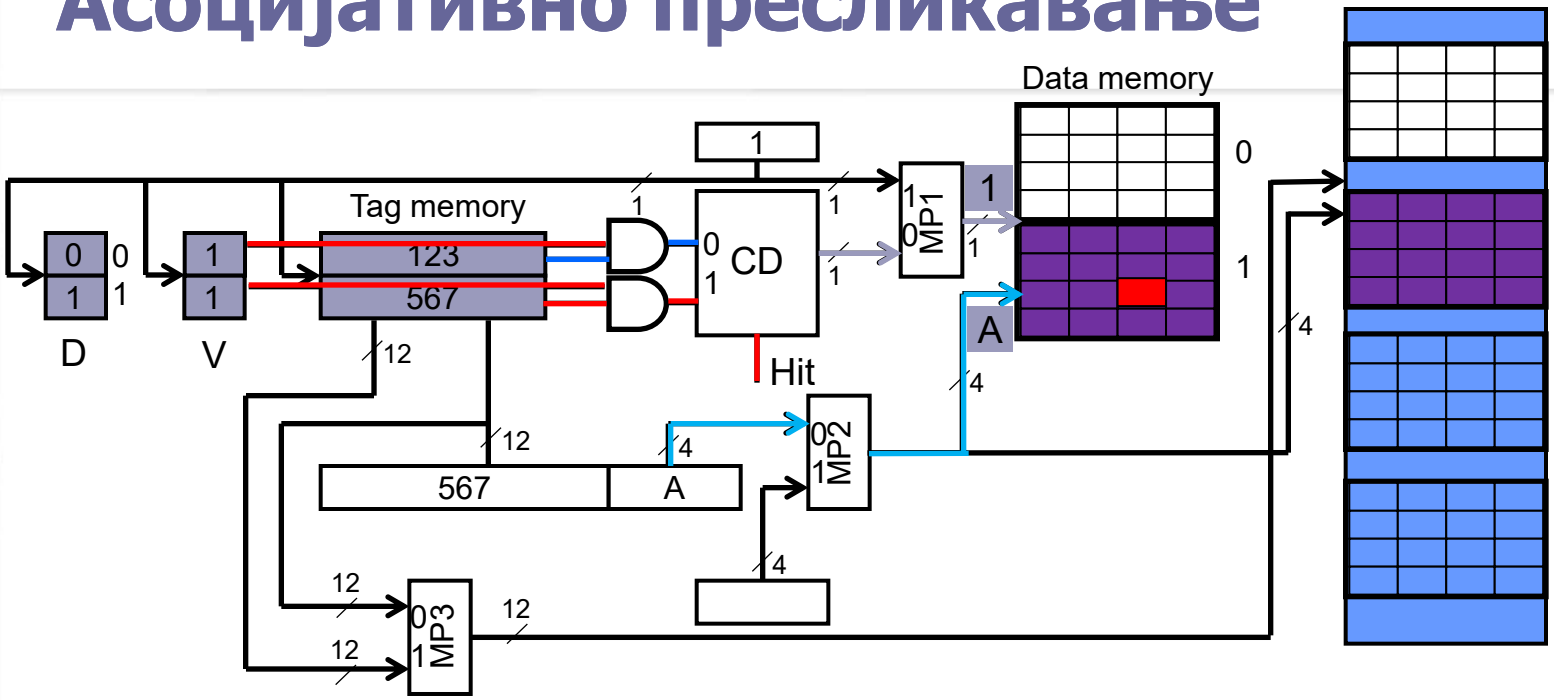
Адреса	Тип	Tag	Word	Hit?	Улаз КМ	ОМ адресе	КМ адресе
1234	Rd	123	4	Miss	0	[1230-123F]	[00-0F]
5678	Rd	567	8	Miss	1	[5670-567F]	[10-1F]

Асоцијативно пресликавање



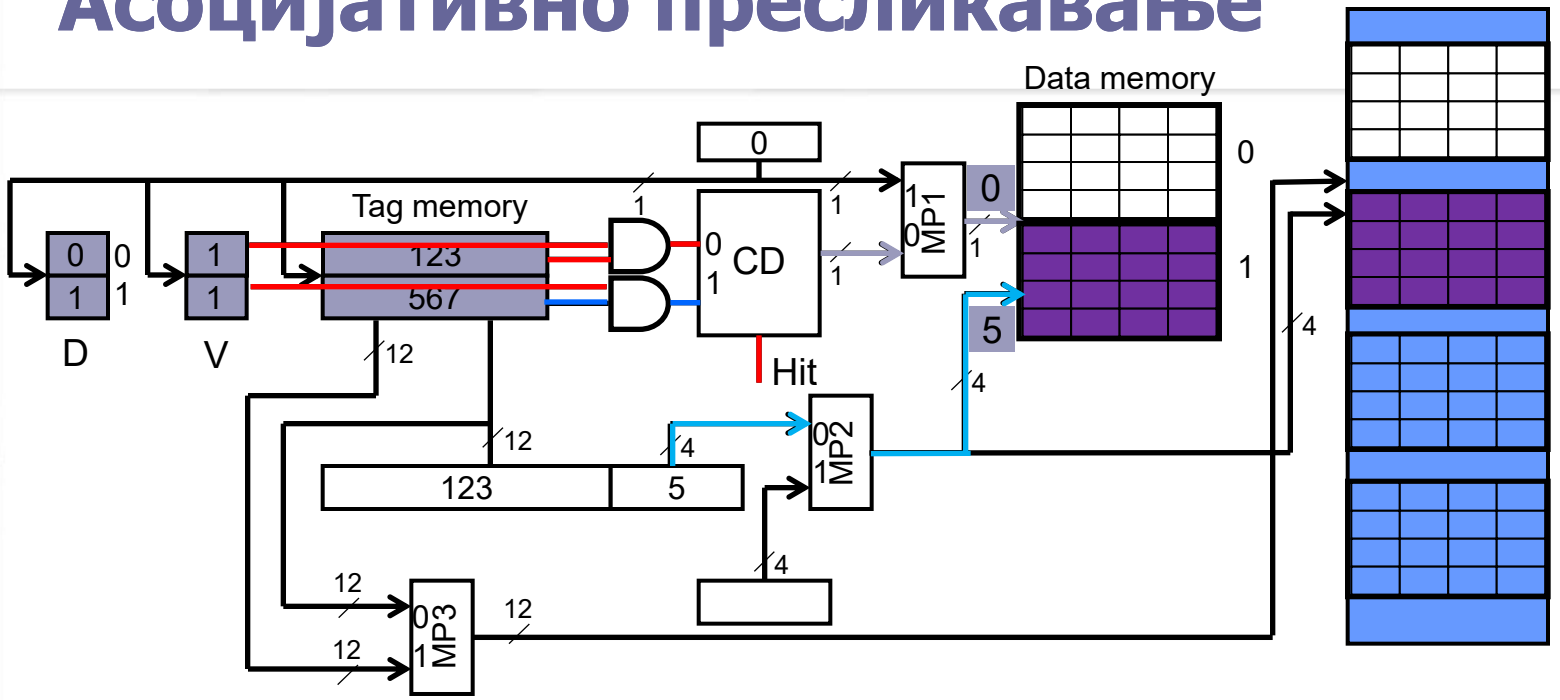
Адреса	Тип	Tag	Word	Hit?	Улаз КМ	ОМ адресе	КМ адресе
1234	Rd	123	4	Miss	0	[1230-123F]	[00-0F]
5678	Rd	567	8	Miss	1	[5670-567F]	[10-1F]

Асоцијативно пресликавање



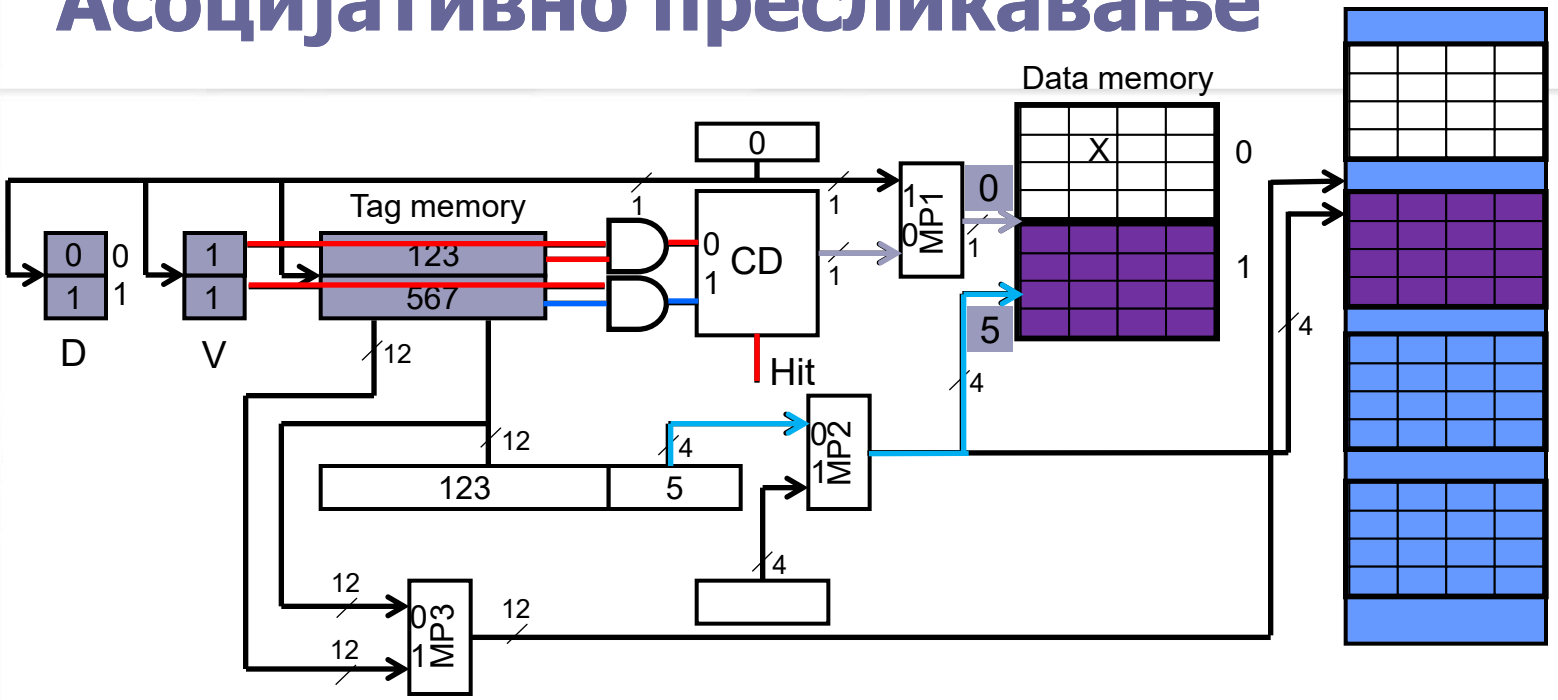
Адреса	Тип	Tag	Word	Hit?	Улаз КМ	ОМ адресе	КМ адресе
1234	Rd	123	4	Miss	0	[1230-123F]	[00-0F]
5678	Rd	567	8	Miss	1	[5670-567F]	[10-1F]
567A	Wr	567	A	Hit	1	-	1A

Асоцијативно пресликавање



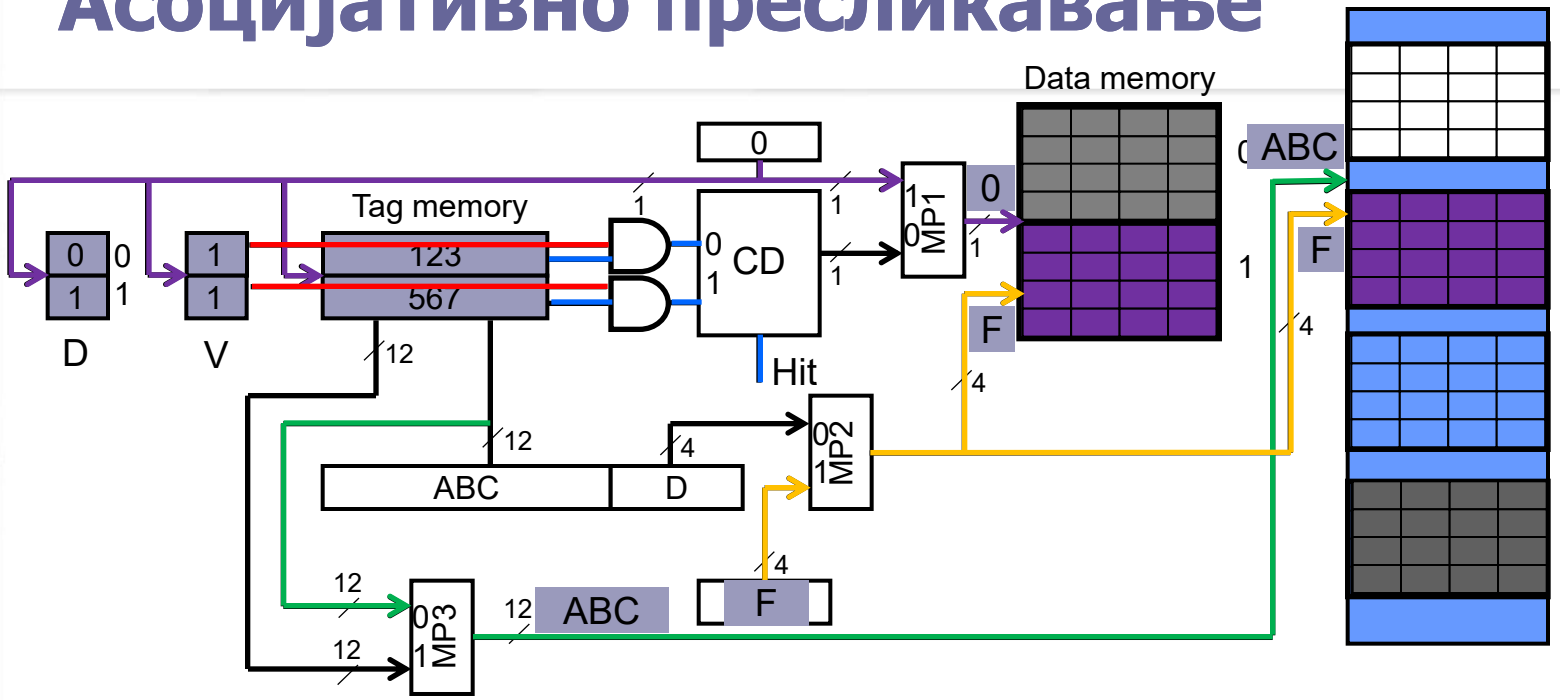
Адреса	Тип	Tag	Word	Hit?	Улаз КМ	ОМ адресе	КМ адресе
1234	Rd	123	4	Miss	0	[1230-123F]	[00-0F]
5678	Rd	567	8	Miss	1	[5670-567F]	[10-1F]
567A	Wr	567	A	Hit	1	-	1A

Асоцијативно пресликавање



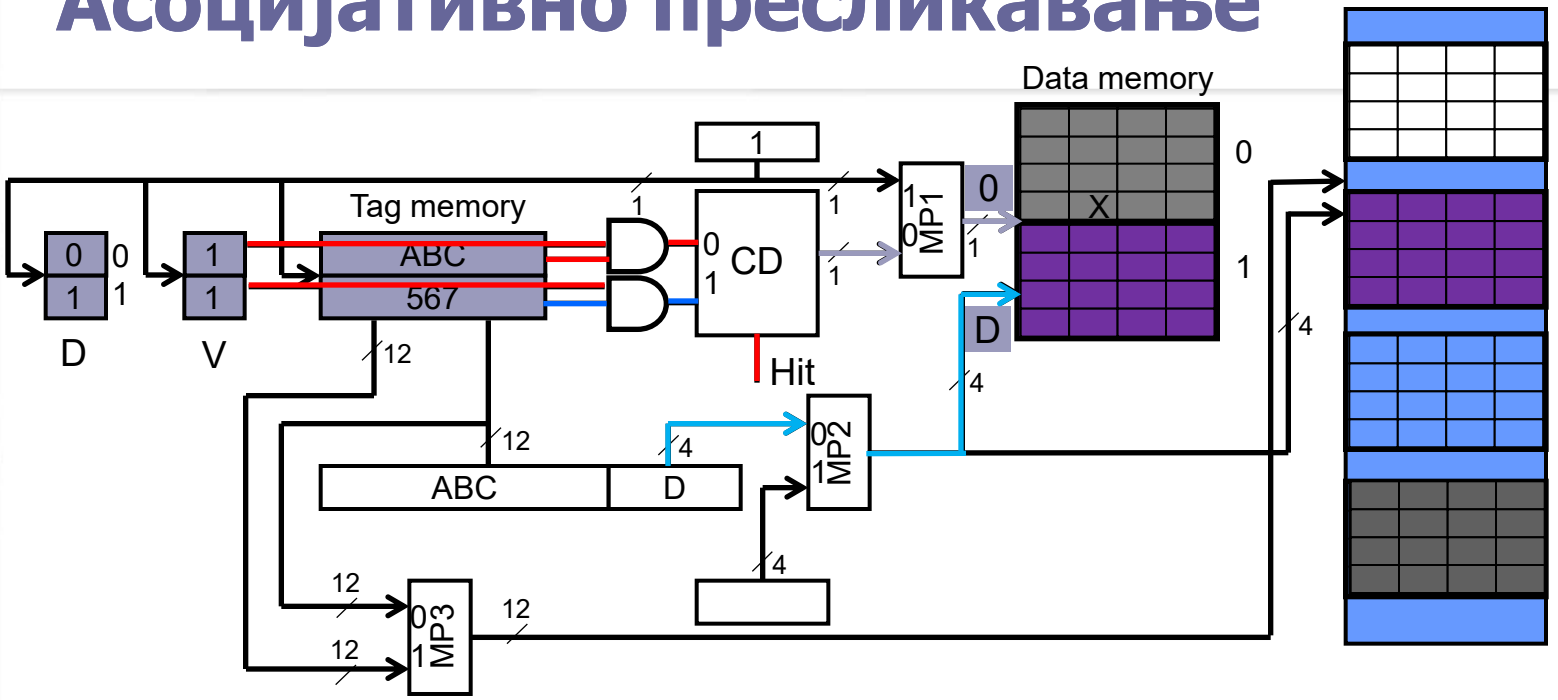
Адреса	Тип	Tag	Word	Hit?	Улаз КМ	ОМ адресе	КМ адресе
1234	Rd	123	4	Miss	0	[1230-123F]	[00-0F]
5678	Rd	567	8	Miss	1	[5670-567F]	[10-1F]
567A	Wr	567	A	Hit	1	-	1A
1235	Rd	123	5	Hit	0	-	05

Асоцијативно пресликавање



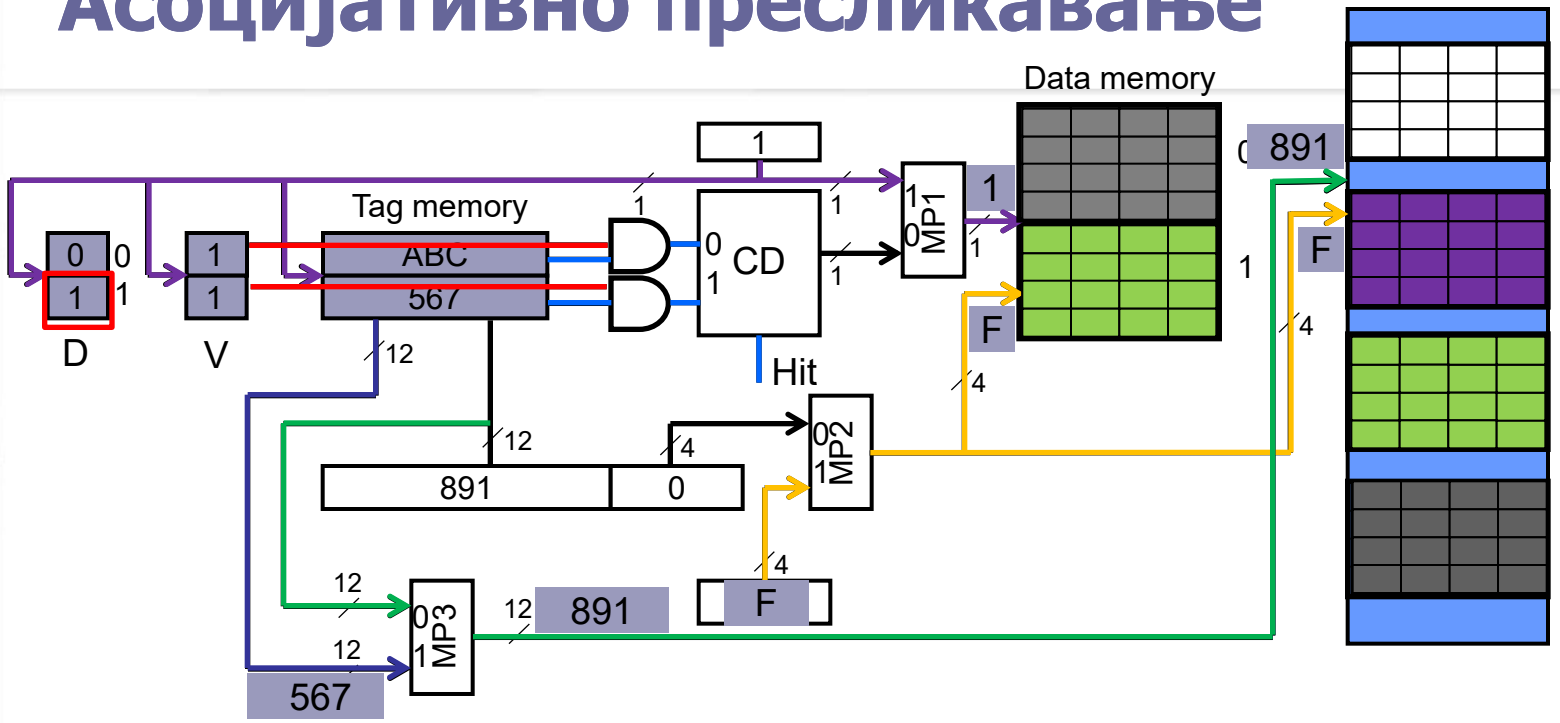
Адреса	Тип	Tag	Word	Hit?	Улаз КМ	ОМ адресе	КМ адресе
1234	Rd	123	4	Miss	0	[1230-123F]	[00-0F]
5678	Rd	567	8	Miss	1	[5670-567F]	[10-1F]
567A	Wr	567	A	Hit	1	-	1A
1235	Rd	123	5	Hit	0	-	05

Асоцијативно пресликавање



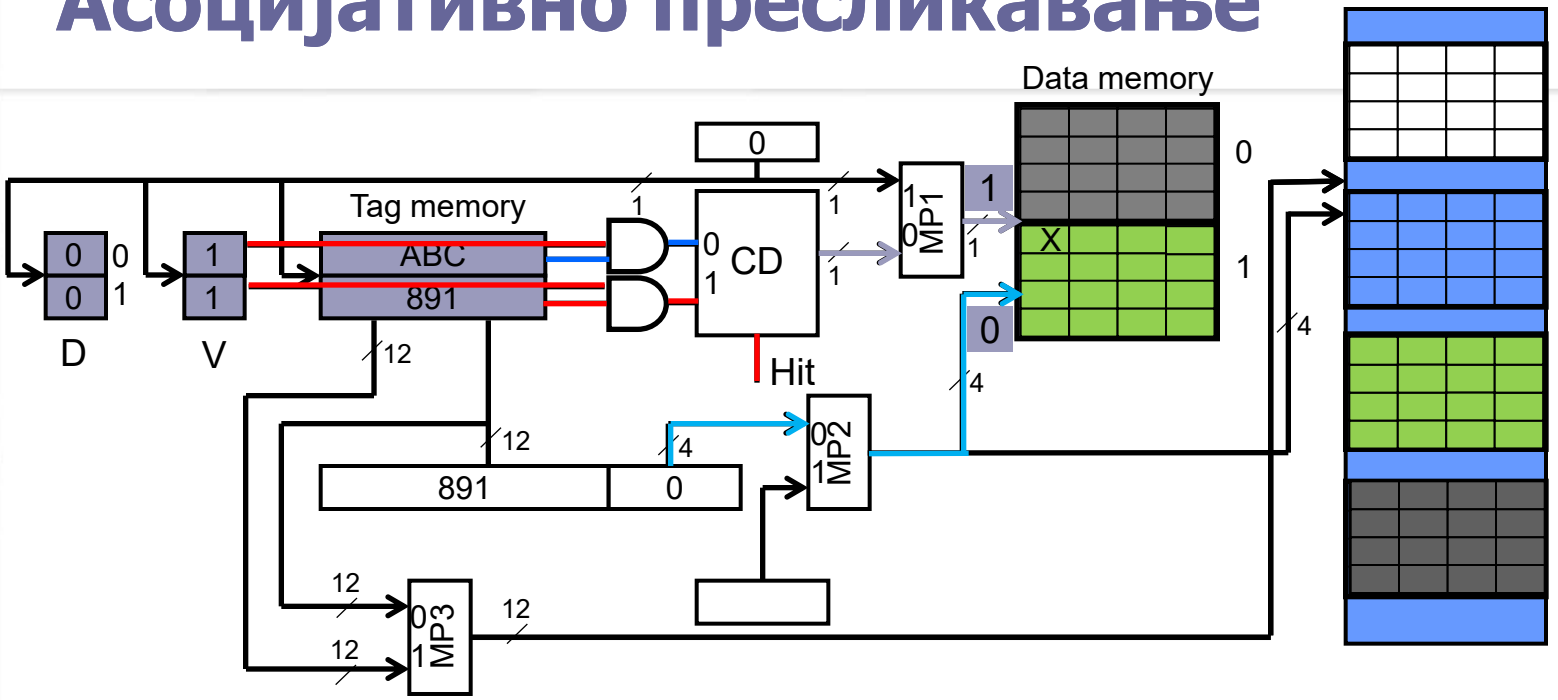
Адреса	Тип	Tag	Word	Hit?	Улаз КМ	ОМ адресе	КМ адресе
1234	Rd	123	4	Miss	0	[1230-123F]	[00-0F]
5678	Rd	567	8	Miss	1	[5670-567F]	[10-1F]
567A	Wr	567	A	Hit	1	-	1A
1235	Rd	123	5	Hit	0	-	05
ABCD	Rd	ABC	D	Miss	0	[ABC0-ABCF]	[00-0F]

Асоцијативно пресликавање



Адреса	Тип	Tag	Word	Hit?	Улаз КМ	ОМ адресе	КМ адресе
1234	Rd	123	4	Miss	0	[1230-123F]	[00-0F]
5678	Rd	567	8	Miss	1	[5670-567F]	[10-1F]
567A	Wr	567	A	Hit	1	-	1A
1235	Rd	123	5	Hit	0	-	05
ABCD	Rd	ABC	D	Miss	0	[ABC0-ABCF]	[00-0F]

Асоцијативно пресликавање



Адреса	Тип	Tag	Word	Hit?	Улаз КМ	ОМ адресе	КМ адресе
1234	Rd	123	4	Miss	0	[1230-123F]	[00-0F]
5678	Rd	567	8	Miss	1	[5670-567F]	[10-1F]
567A	Wr	567	A	Hit	1	-	1A
1235	Rd	123	5	Hit	0	-	05
ABCD	Rd	ABC	D	Miss	0	[ABC0-ABCF]	[00-0F]
8910	Rd	891	0	Miss	1	[5670-567F][8910-891F]	[10-1F] [10-1F]

Асоцијативно пресликавање

- Комплетан поступак ажурирања кеш меморије се обавља хардверски
- Водити рачуна да алгоритам рада кеш меморије може да зависи од технике **ажурирања садржаја** оперативне меморије!

Асоцијативно пресликавање

- Добра страна асоцијативног пресликавања је **велика вероватноћа** да се податак нађе у кеш меморији (блок може да се смести у произвољни улаз кеш меморије).
- Лоша страна је **велико време** које је потребно да се прочита податак из кеш меморије уколико постоји сагласност (компаратор, и коло, или коло, кодер, мултиплексер, приступ меморији).

Асоцијативно пресликавање

- CacheSize = BlockNumC * BlockSize
- CacheSize = $2^n * 2^l$ AU

- MemSize = BlockNumM * BlockSize
- MemSize = $2^h * 2^l$ AU

Замена блокова кеш меморије

- Разматрају се четири алгоритма замене и то
  - RANDOM,
  - FIFO,
  - LRU и
  - PSEUDO LRU
- При избору алгоритма треба водити рачуна:
  - Алгоритам треба да обезбеди **минималну вероватноћу** да ће блок који је одабран за замену и враћен из кеш у оперативну меморију убрзо морати поново да се довуче из оперативне у кеш меморију.
  - Други је да **цена хардвера** потребног за његову реализацију буде што је могуће нижа.

Ажурирања садржаја оперативне меморије

- Ажурирање оперативне меморије одређује како се код операције уписа мења садржај у оперативној меморији.
- Код захтева за упис, могу јавити две ситуације:
  - прва је да у кеш меморији постоји сагласност,
  - друга је да нема сагласности.

Ажурирања садржаја оперативне меморије

- За случај када је у кеш меморији откривена сагласност, постоје два приступа и то
  - *упиши скроз* (***write through*** или ***store through***) и
  - *врати назад* (***write back*** или ***copy back***).
- Код приступа ***упиши скроз***, при сваком захтеву за упис истовремено се врши упис и у кеш меморију и у оперативну меморију.
- Код приступа ***врати назад***, при сваком захтеву за упис врши се упис само у кеш меморију, па одговарајући садржај у оперативној меморији није ажуран.

Ажурирања садржаја оперативне меморије

- Предност приступа *упиши скроз* је у томе да је оперативна меморија увек ажурна чиме је обезбеђена конзистентност садржаја оперативне и кеш меморије.
- Недостатак овог приступа је у обраћању оперативној меморији при сваком упису у кеш меморију, чиме се беспотребно оптерећује магистрала уписивањем међурезултата у оперативну меморију.

Нема потребе за D битима!

Ажурирања садржаја оперативне меморије

- Предност приступа *врати назад* је у томе што се оперативној меморији и магистрали приступа само онда када се блок враћа из кеш меморије у оперативну меморију што резултује у мањем саобраћају на магистрали.
- Недостатак овог приступа је потреба да се блок који се избацује из кеш меморије мора најпре вратити у оперативну меморију, па тек онда довући нови, што знатно успорава одзив кеш меморије у случају промашаја.

Ажурирања садржаја оперативне меморије

- За случај када у кеш меморији није откривена сагласност, постоје два приступа и то
  - *довуци блок* (*write allocate*) и
  - *не довлачи блок* (*no write allocate*).
- Код приступа *довуци блок*, блок се довлачи из оперативне у кеш меморију, чиме се обезбеђује да се сада у кеш меморији открива сагласност. (ажурирање садржаја оперативне меморије користи приступ *упиши скроз* или *врати назад*)
- Код приступа *не довлачи блок*, блок се не довлачи из оперативне у кеш меморију, већ се упис врши само у оперативну меморију.

Ажурирања садржаја оперативне меморије

- Обично се уз приступ *врати назад* (*write back*) користи приступ *довуци блок* (*write allocate*),
- док се уз приступ *упиши скроз* (*write through*) користи приступ *не довлачи блок* (*no write allocate*).

Питања?

Електротехнички Факултет
Универзитет у Београду