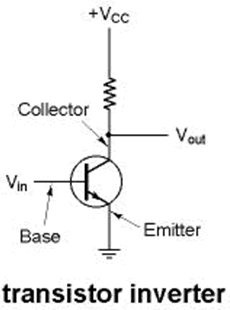
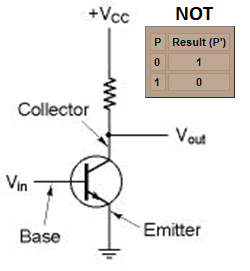
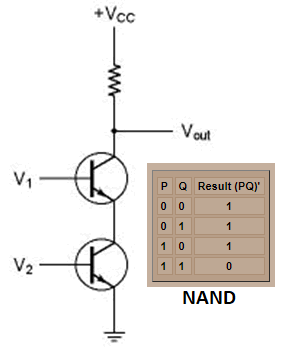
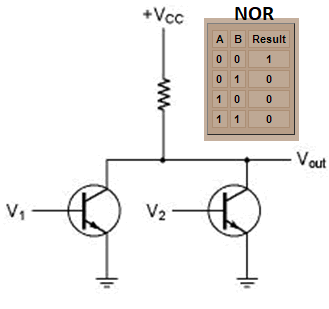
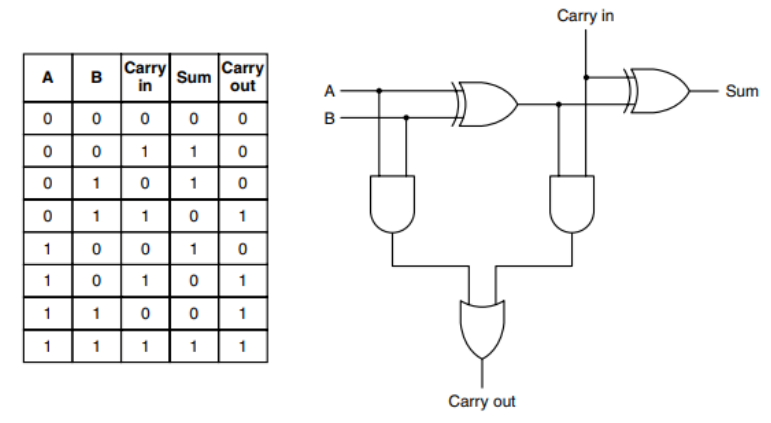
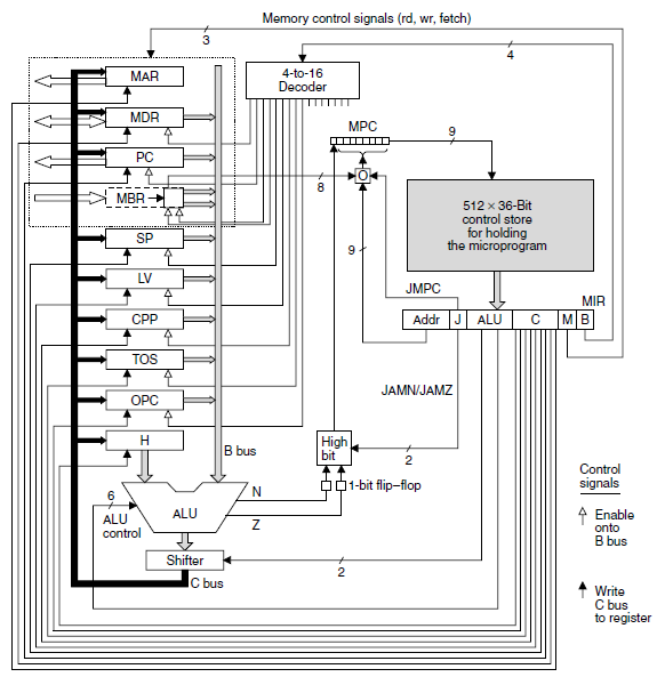
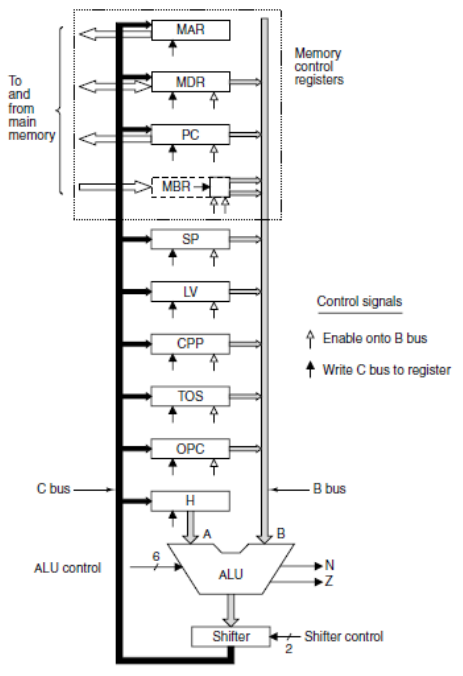
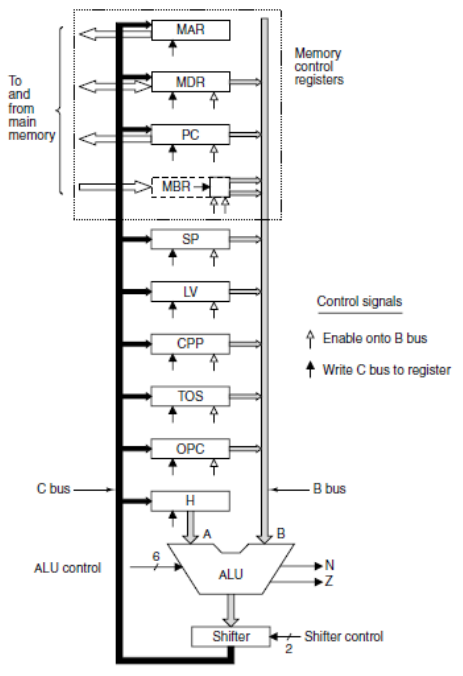
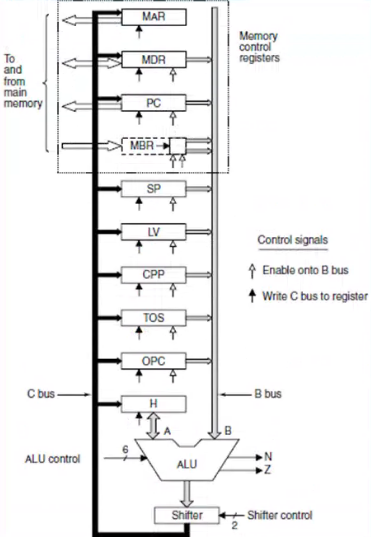
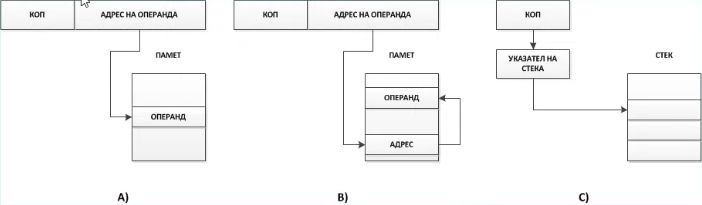
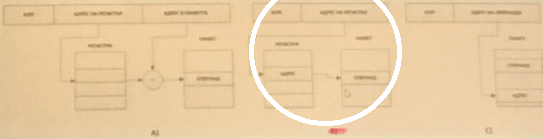
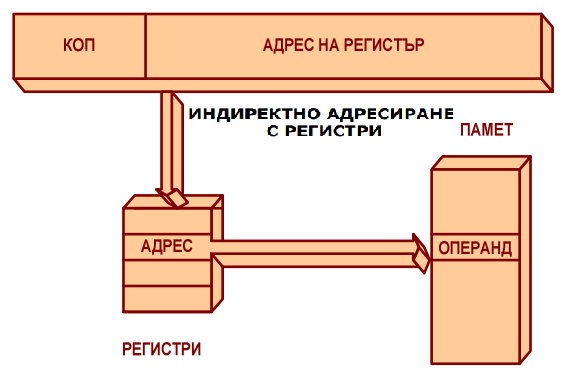
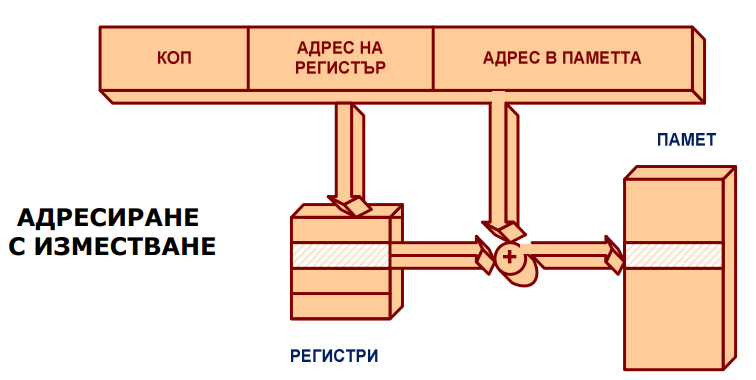
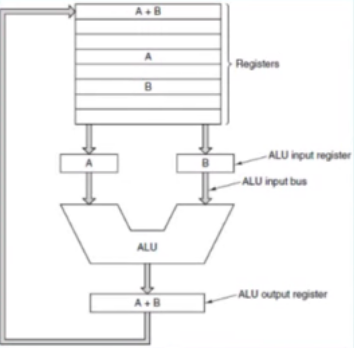
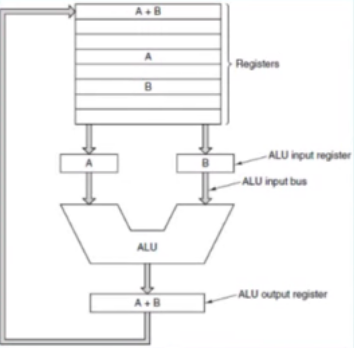
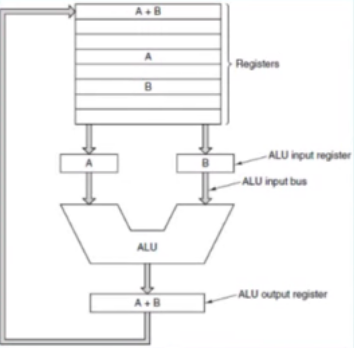
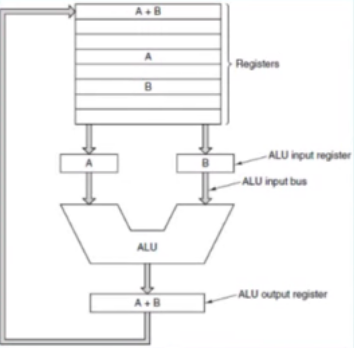
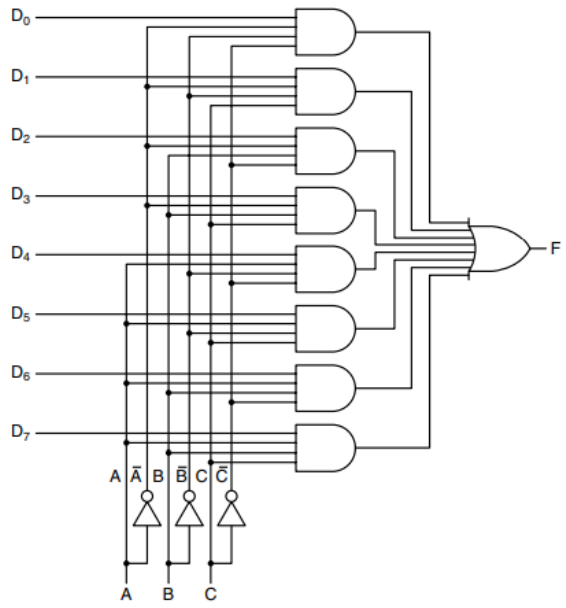
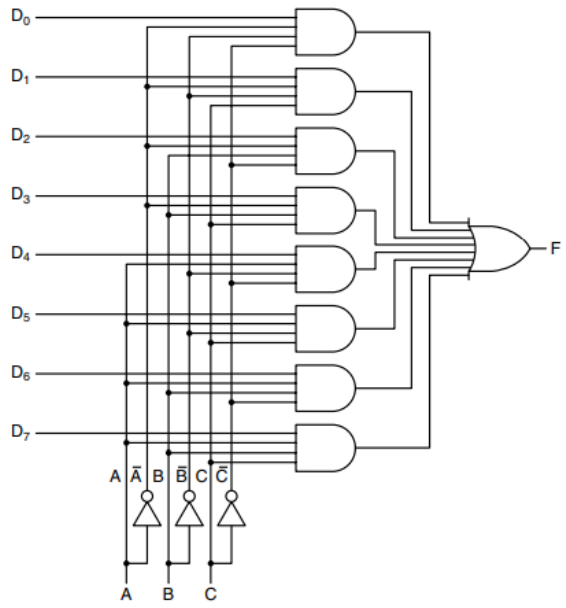
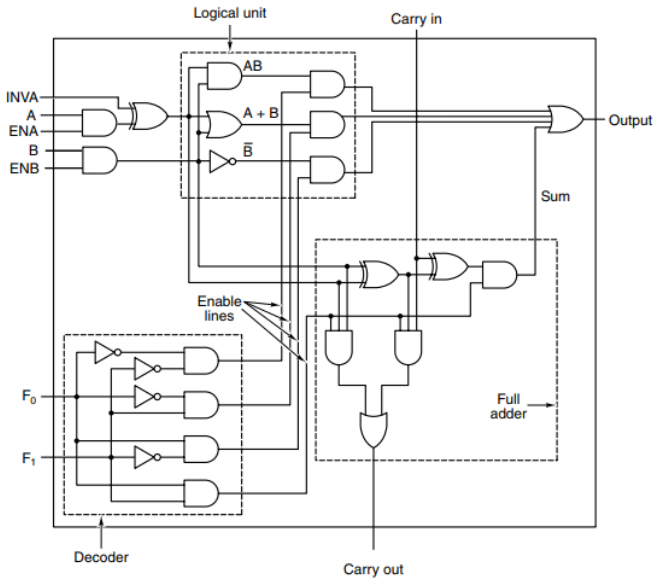
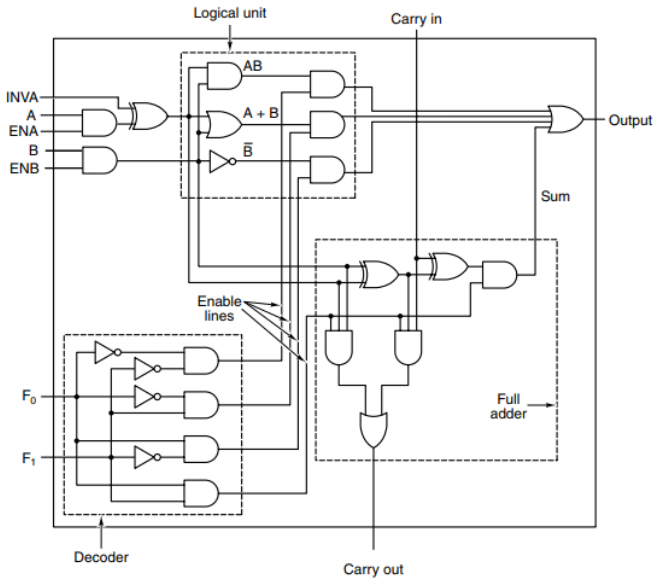
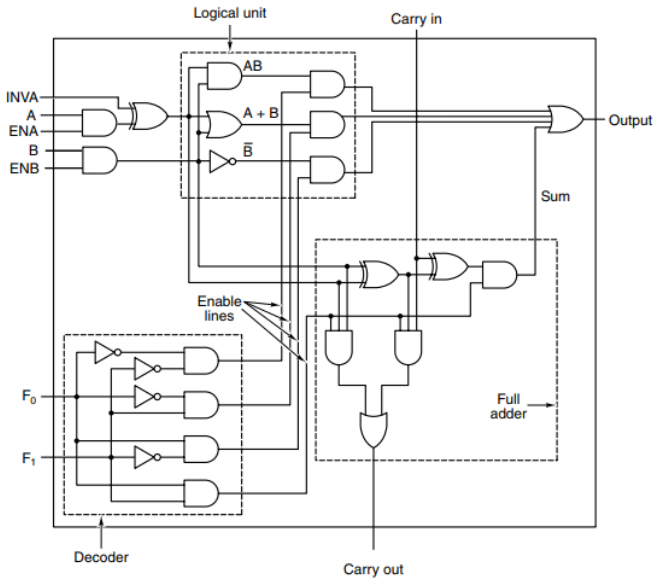
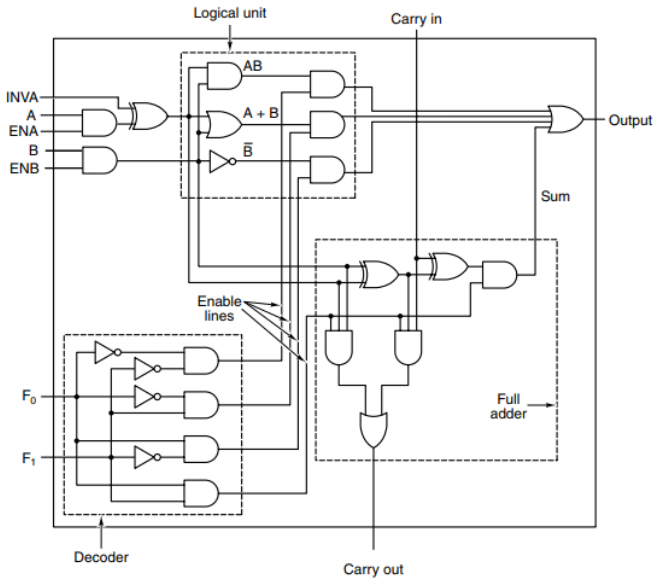
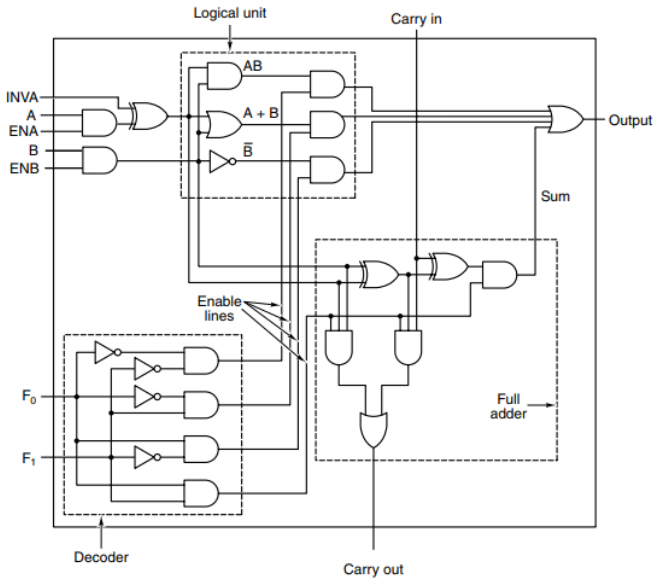
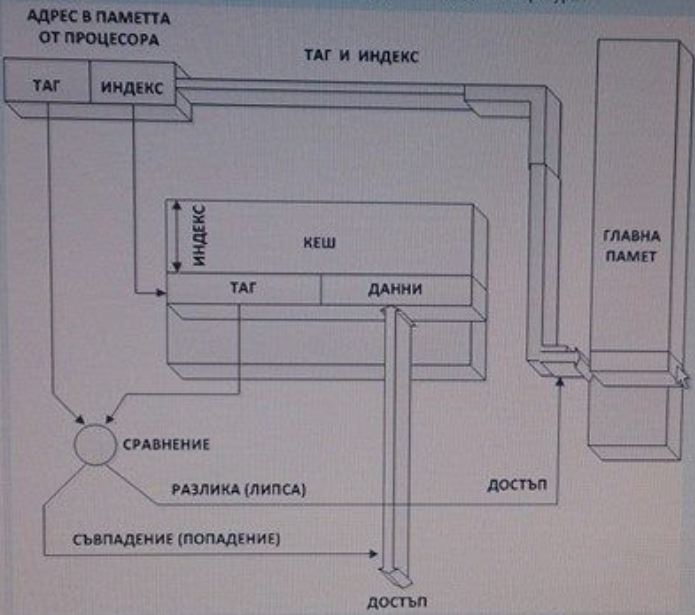
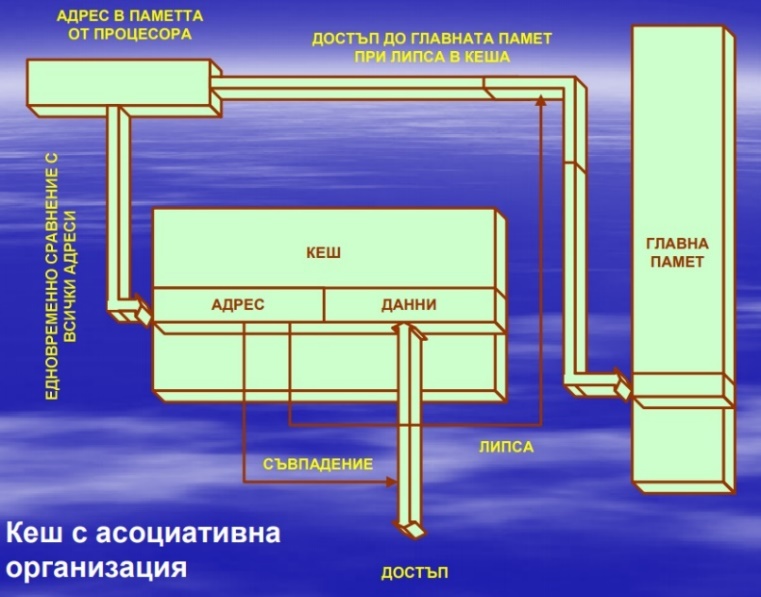
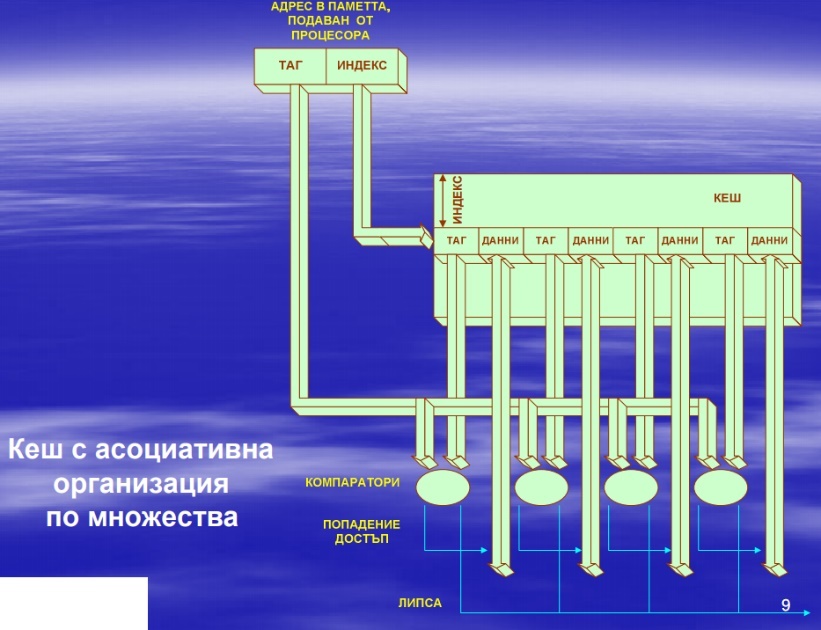
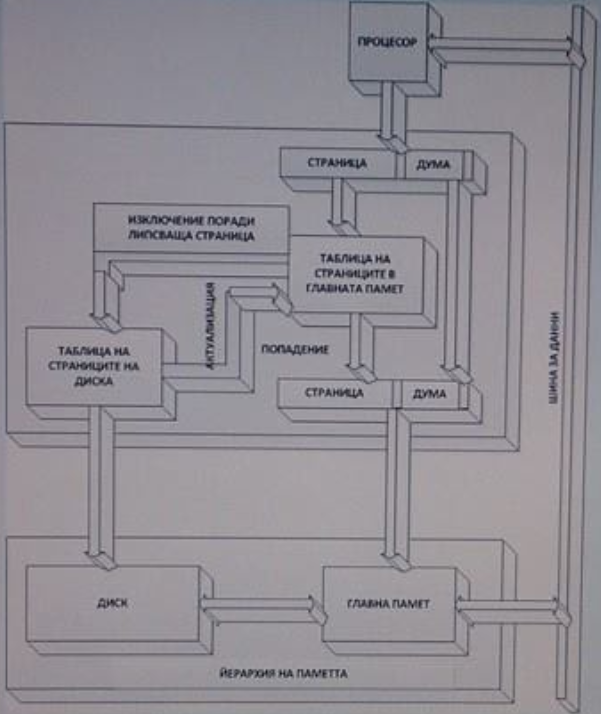
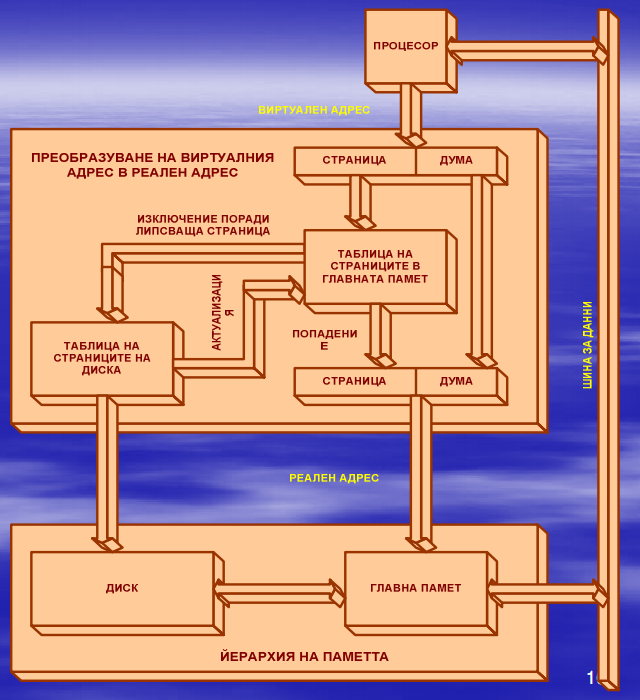
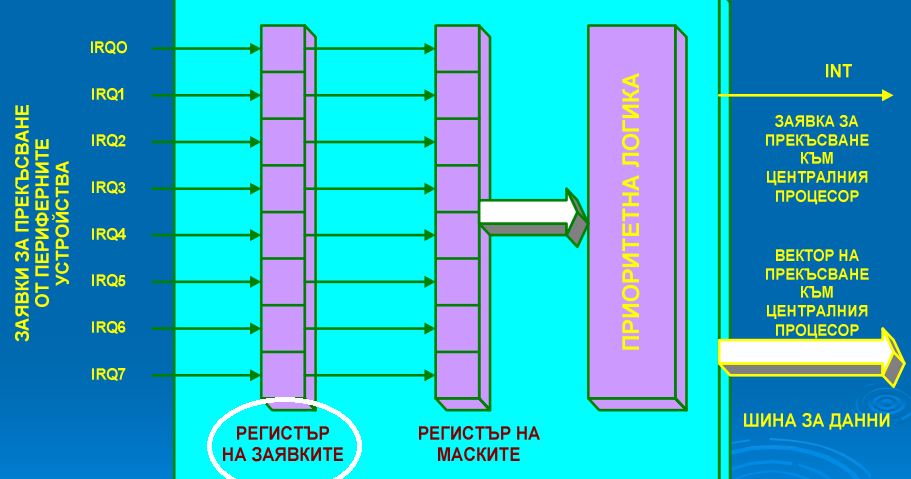
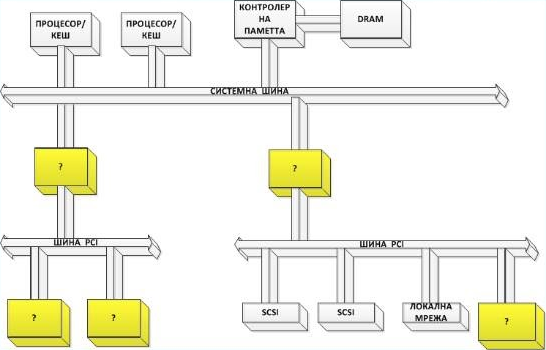
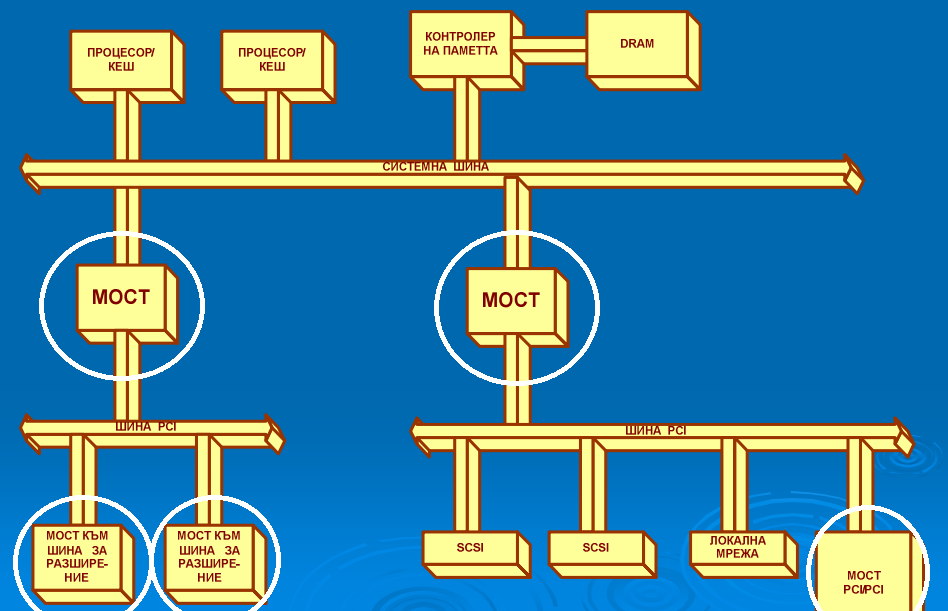
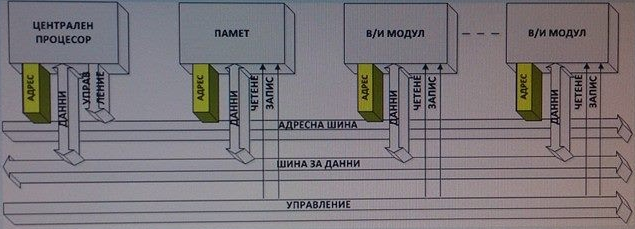
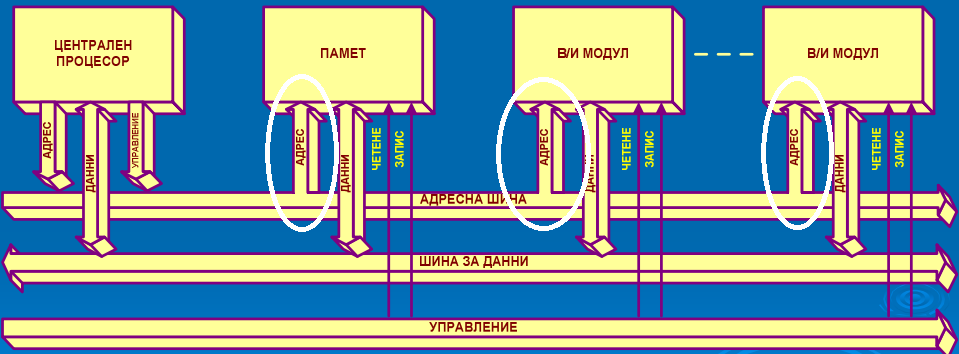
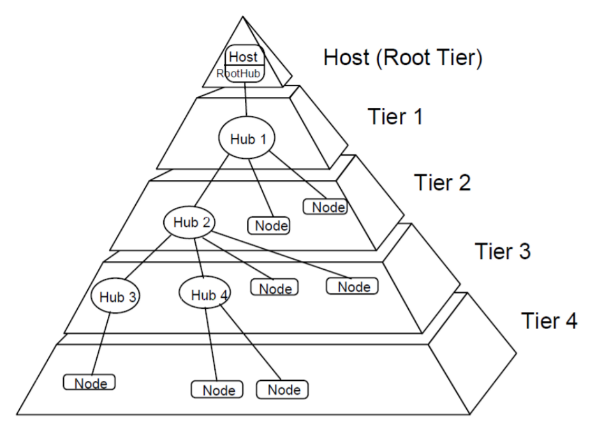
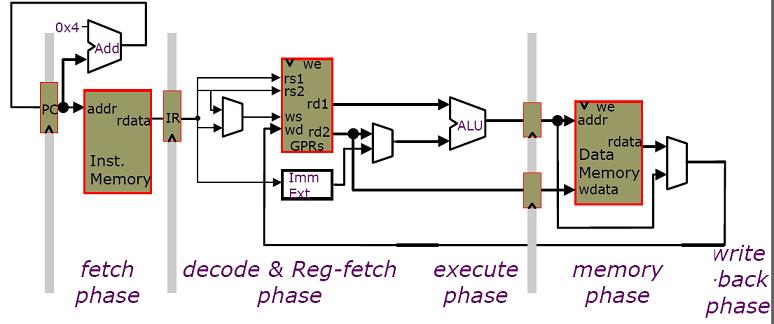
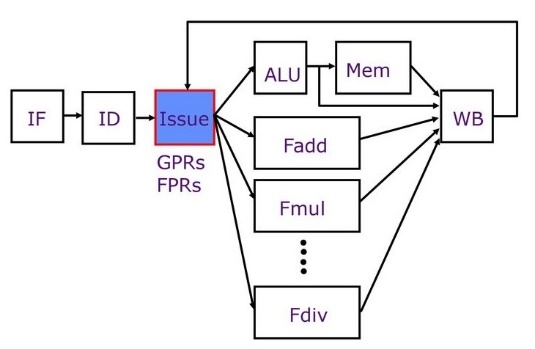
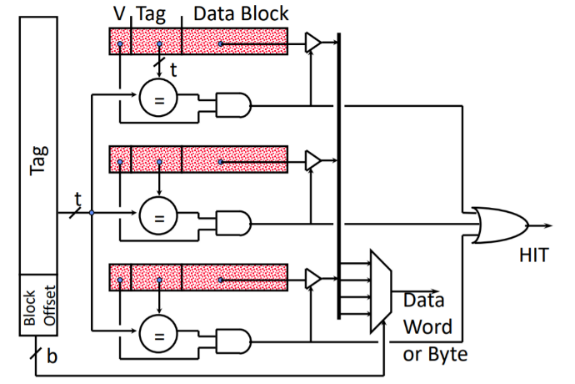
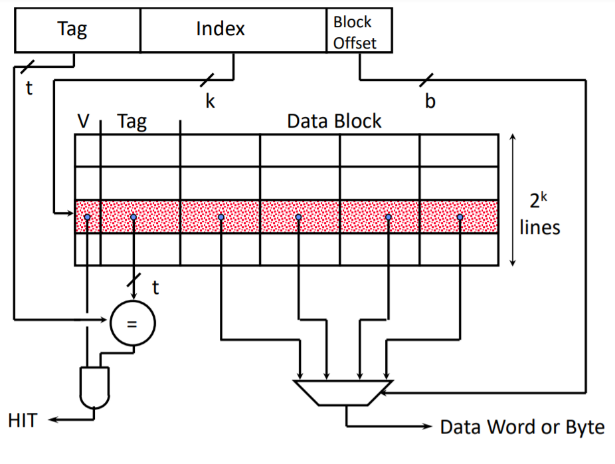
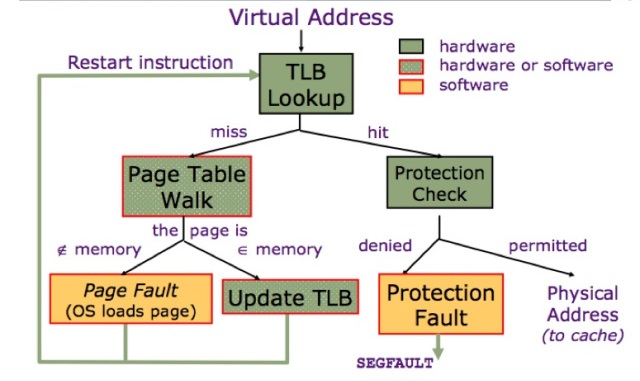
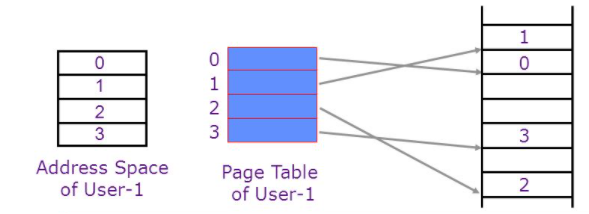
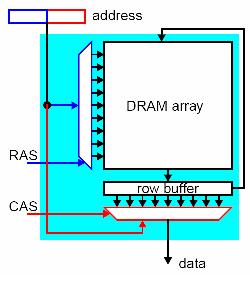
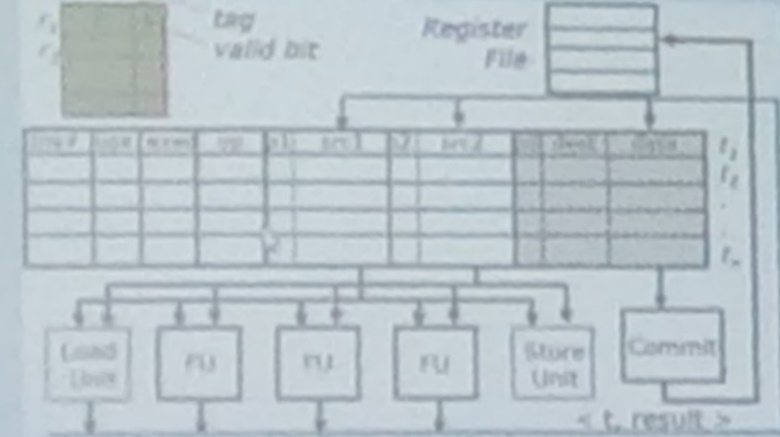
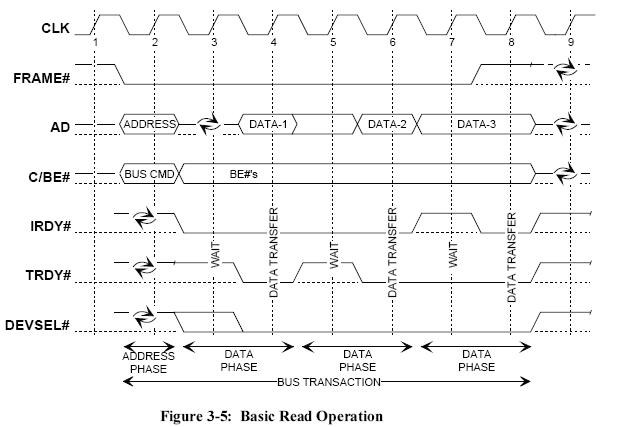
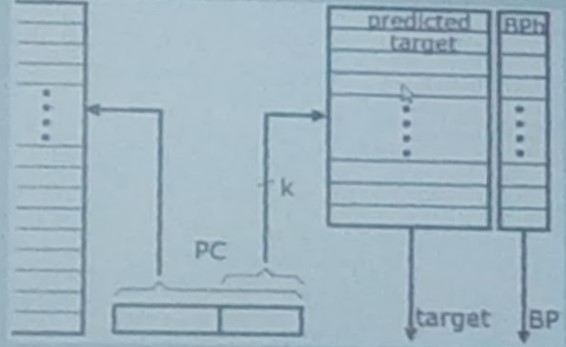
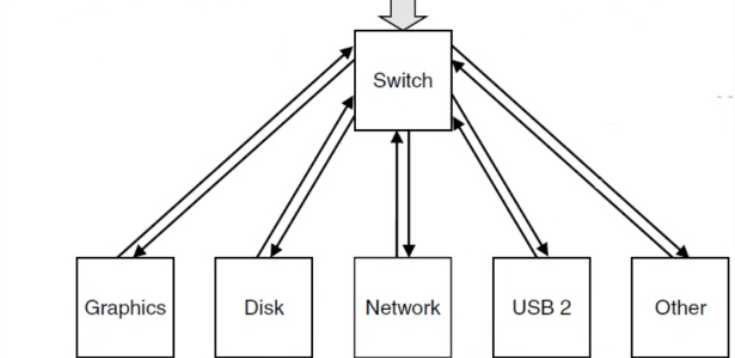
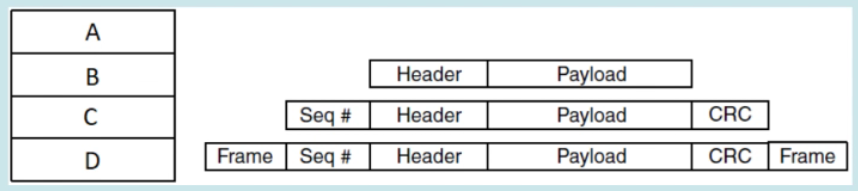
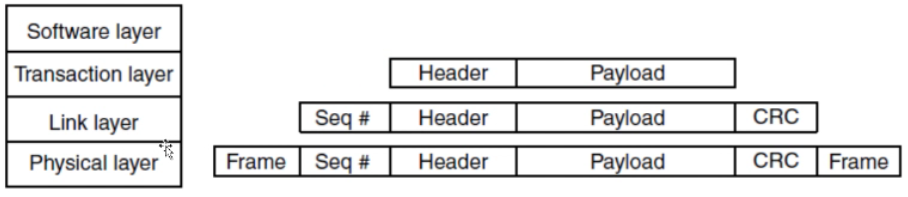
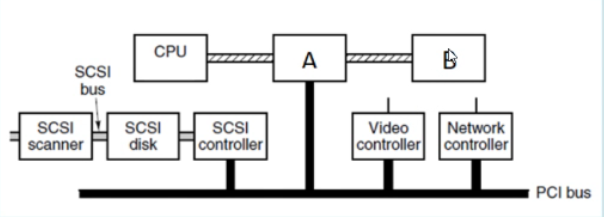
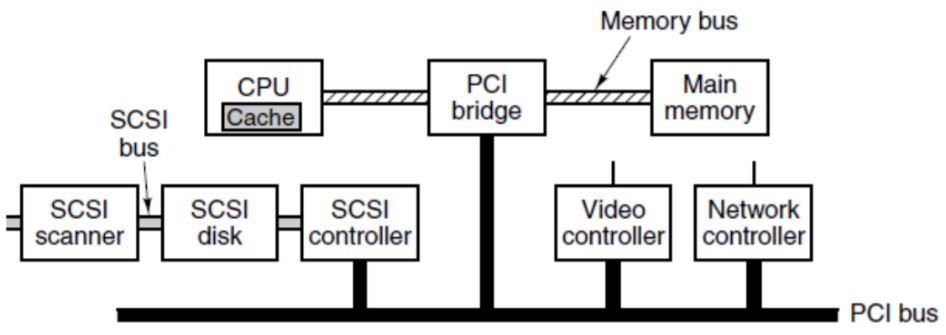
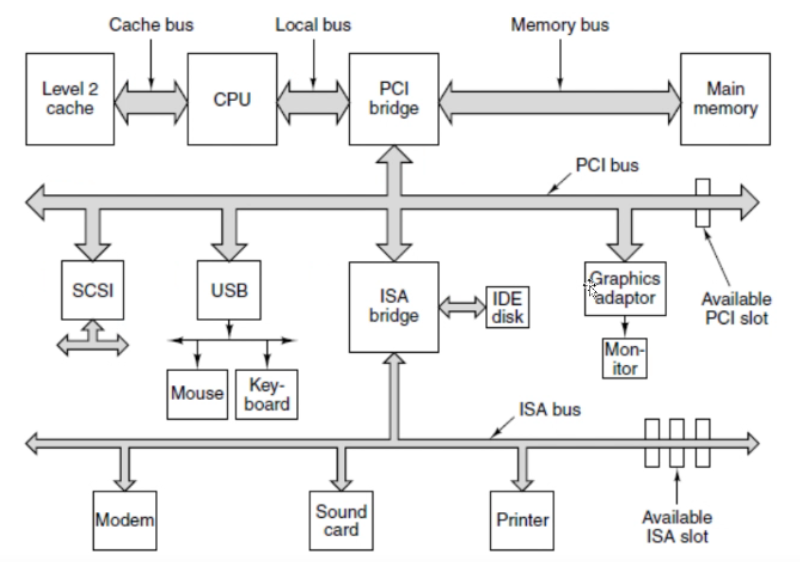
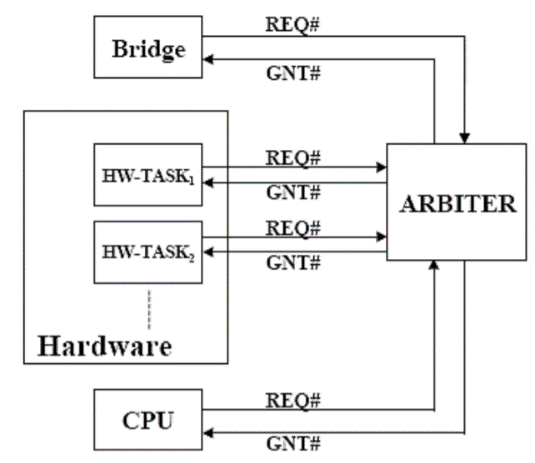
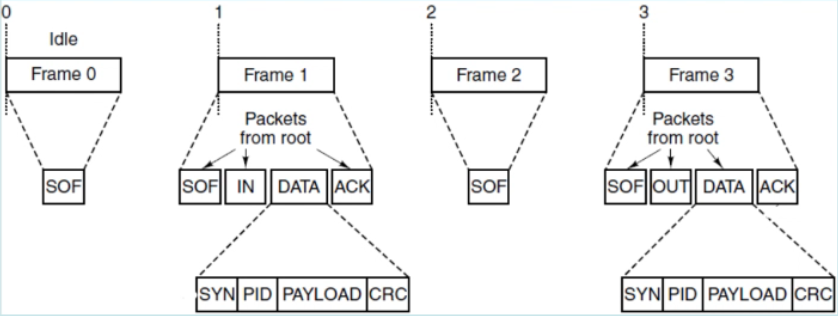
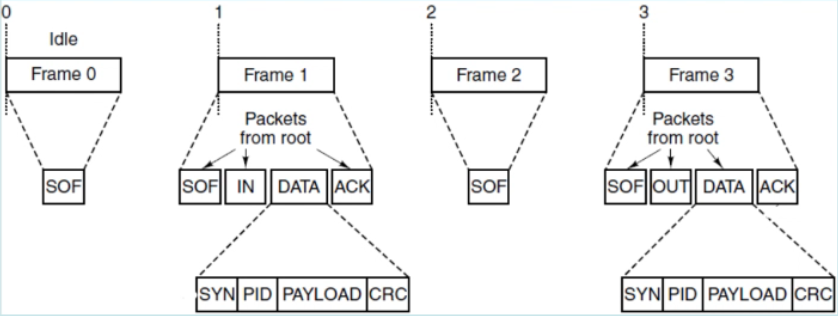
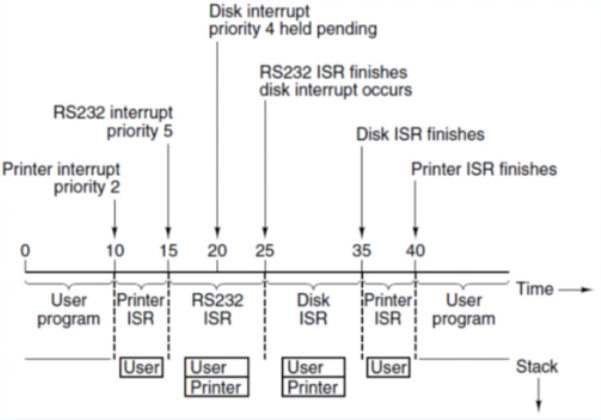
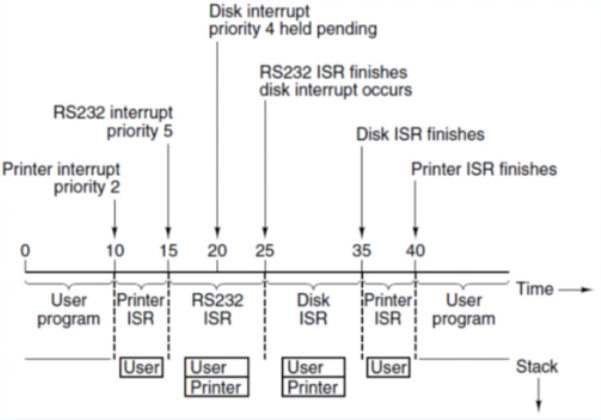
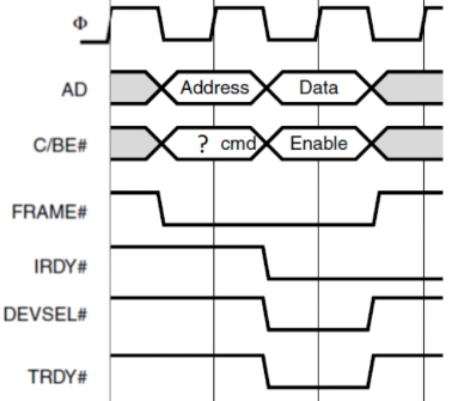
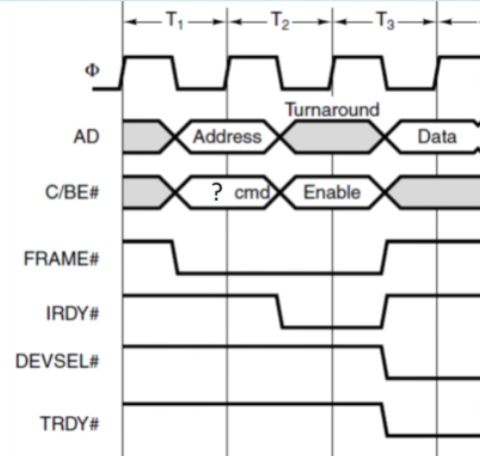
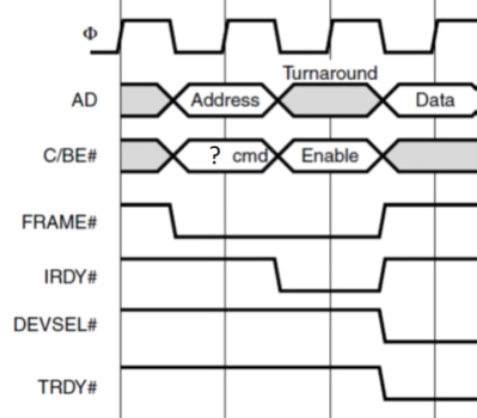
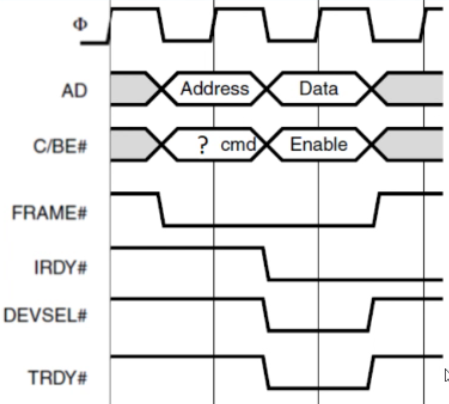
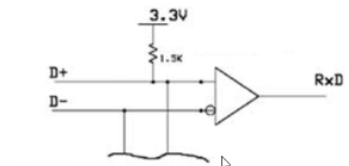
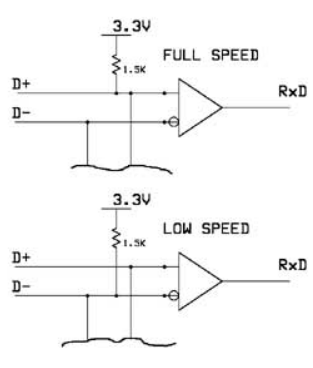
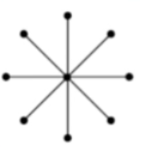
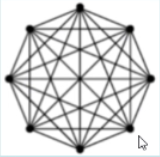
1. Основни производители на x86 архитектурата са
   * **IBM**, Sun Microsystems, **Hewlett-Packard**, Advanced Micro Device, **Intel**
2. От кои фактори зависи производителността на компютърните системи?
   * от времето за достъп до регистриране, **архитектурата и технологията**, от тактовата честота
3. В режим на директен достъп до паметта управлението на обмена се поема от:

* **устройството за вход/изход**, процесор, памет;

1. Коя/кои от изброените шини не са част от системната шина?
   * шина за данни, **шина за логически сигнали**, контролна шина, адресна шина;
2. Шината PCI в процесорната архитектура се явява мост между:
   * **шина EISA и периферията**, системна шина на микропроцесор и шина EISA, системна шина на микропроцесор и основна памет, микропроцесор и кеш памет L2;
3. Кои от изброените подходи способстват за намаляване на латентността (закъснението) при зареждане на данни в кеш паметта?
   * Едновременен запис, **зареждане при поискване (on demand)**, обратен запис, **спекулативно зареждане**, **отложен запис на данни**
4. Единицата за измерване на производителността MIPS за съпоставяне на еднотипни процесорни архитектури се основава на
   * Брой операции с операнди от виртуалната памет, Брой операции с операнди от кеш-паметта, **Брой операции с операнди от главната памет**, Брой операции с операнди от флаш-паметта
5. За какво служи програмният брояч?
   * За съхраняване на резултата от аритметичните операции изпълнявани в ALU-то, **За указване на адреса на следващата инструкция за изпълнение**, За отчитане броя на изпълненията на дадена инструкция, За отчитане на процента памет заеман от изпълняващата се програма
6. Кое от изброените **не е** вярно за кеш паметта?
   * **Намира се в адресното пространство на процесора и може да се адресира с помощта на машинните инструкции.**, Осигурява бърз достъп до интензивно използваните данни чрез тяхното предварително зареждане., Съгласува интерфейса на процесора и на контролера на паметта.
7. Дадена е машина с побайтово адресируема основна памет 224 байта и с кеш за данни с директна организация и с капацитет 64K и 32 байтови блокове. Колко бита са необходими за таг, блок и отместване?
   * **8 бита за таг, 11 бита за блок, 5 бита за уникална дума или байт от блок от основната памет**, 8 бита за таг, 5 бита за блок, 11 бит за уникална дума или байт от блок от основната памет, 11 бита за таг, 8 бита за блок, 5 бит за уникална дума или байт от блок от основната памет
8. Ако даден компютър има честота 10 MHz, колко е времето за един такт (clock time)?
   * 1 ms, 200 ns, **100 ns**, 10 ps, 20 ns
9. Ако даден компютър има честота 2 GHz, колко е времето за един такт (clock time)?
   * 200 ns, **500 ps**, 20 ns, 10 ps, 1 ms
10. Ако даден компютър има честота 500 MHz, колко е времето за един такт (clock time)?
    * **2 ns**, 1 ms, 10 ps, 20 ns, 200 ns
11. Методът, при който запис на данни се извършва само в кеш паметта, а състоянието на кеш-блока се отбелязва като обновено, а управляващото поле с признак, наречен „update“ или „dirty“ се нарича:
    * **обратен запис (write back)**, едновременен запис (write through)
12. Кеш памет с асоциативна организация е с капацитет 64 линии, разделени в множества с по четири линии всяко. Основната памет съдържа 4K блока с по 128 думи всеки. Посочете адресния формат, който и съответства:
    * (TAG-SET-WORD): 8-4-7, 14-8-4, **14-8-7**, 4-8-7
13. Кеш памет с асоциативна организация е с капацитет 64 линии, разделени в множества с по 16 линии всяко. Основната памет съдържа 4K блока с по 128 думи всеки. Посочете адресния формат, който и съответства:
    * (TAG-SET-WORD): 4-8-7, 14-8-4, **8-2-7**, 14-8-7
14. Дадена е машина с побайтово адресируема основна памет и 2-кратен множествено асоциативен кеш. Кеш-логиката интерпретира адреса от паметта както следва: 14 бита за таг, 8 бита за множество, 2 бита за адресиране на дума. Колко на брой са блоковете в главната памет?
    * 28, 224, не може да се определи, **222**, 22, 214
15. Дадена е машина с побайтово адресируема основна памет с размер 216 байта и размер на блока 8 байта. Кеш-паметта е с директна организация и се състои от 32 линии. Какъв е максималният брой байтове, които могат да бъдат разположени в кеш-паметта?
    * 1024B, 512B, **256B**, 128B, 64B
16. LFU е алгоритъм за:
    * **заместване на най-неизползваната страница в кеш паметта**, заместване на някои от използваните страници в кеш паметта, периодично изчистване (нулиране) признаците за обръщение към най-рядко използвани страници в кеш паметта, заместване на най-често използваната страница от кеш паметта
17. Как се преодоляват хазартите в конвейера при достъп до общи ресурси в инструкционния поток?
    * Чрез пренареждане на инструкционния поток от компилатора, чрез предсказване на преходите в инструкционния поток, чрез дублиране на критичните фази, **чрез блокиращи техники на конвейера за определено време и последователна обработка**, чрез прескачане на критичните фази, **чрез въвеждане на закъснения и модифициране главната таблица на заетост на конвейера**
18. На каква концепция се базират инструкционните конвейери?
    * **Паралелизъм на ниво инструкции**, Паралелизъм на ниво данни, Паралелизъм на ниво заявки, Паралелизъм на ниво нишки
19. Какъв е формата на линията при кеш паметта с **асоциативна организация**?
    * таг, линия, дума, **таг, дума**, таг, множество, дума
20. Какъв е формата на линията при кеш паметта с **пълна асоциативност**?
    * таг, линия, дума, **таг, дума**, таг, множество, дума
21. Какъв е формата на линията при кеш паметта с **множествена асоциативност**?
    * таг, линия, дума, таг, дума, **таг, множество, дума**
22. Методът, при който обновяване на главната памет настъпва след премахване на дума от кеша се нарича:
    * cache-write, **write-back**, защитен запис, write-through
23. Методът, при който обновяване на главната памет настъпва **асинхронно** след премахване на дума от кеша се нарича:
    * cache-write, **write-back**, защитен запис, write-through
24. Методът, при който обновяване на главната памет настъпва **синхронно** след премахване на дума от кеша се нарича:
    * cache-write, write-back, защитен запис, **write-through**
25. Кое от твърденията относно размера на страниците е невярно?
    * **по-големият размер на страниците увеличава времето за входно-изходни операции (четене/запис на страници) по време на изпълнение на програмите**, по-големият размер води до въвеждане на излишна информация, която може и да не е необходима, по-малкият размер на страница води до намаляване на вътрешната фрагментация, която е 1/2 от последната страница
26. При кой/кои от посочените методи на запис в кеша се генерира значителен трафик по шината?
    * write-through, **write-back**, нито един от двата, и двата метода
27. Кой/кои от изброените етапи е част от инструкционния цикъл?
    * Пренареждане, **Извличане**, **Изпълнение**, **Декодиране**
28. Кой/кои от изброените етапи не е част от инструкционния цикъл?
    * **Пренареждане**, Извличане, Изпълнение, Декодиране
29. Кой вид памети са по-бързи от кеш-паметите?
    * **регистрите**, flash паметите, главната памет – DRAM
30. Колко проводна може да бъде USB платината?
    * **4**, 6, **9**
31. Каква система инструкции имат различните фамилии мултипроцесорни архитектури?
    * Еднаква, **Съвпадаща отчасти**, **Изцяло различна**
32. При кой от трите метода на разполагане блокове от операционната памет в кеша е най-малко вероятно два блока да бъдат записани на една и съща линия?
    * **пълна асоциативност**, директно (пряко съпоставяне), множествена асоциативност
33. Отбележете невярното твърдение:
    * главната (DRAM) памет е по-бавна от SRAM паметта, DRAM консумира повече енергия от SRAM, кеш-паметта е малка, бърза и SRAM базирана, **SRAM е по-евтина DRAM**
34. Каква е корелацията капацитет (размер) и бързодействие при компютърните памети?
    * пропорционална, няма корелация, **обратнопропорционална**
35. Примитивите за ресурсен мениджмънт на многоядрени платформи са:
    * имплементирани хардуерно, **универсални**, специализирани
36. Какво е предназначението на механизмите за защита на паметта?
    * да ограничават достъпа до съхраняваната в системата информация, да откриват грешки при адресирането преди извършването на неразрешен достъп, **да ограничават достъпа до определени сегменти или страници**, да не допускат модифициране на системния софтуер от приложения
37. Как се определя производителността на централния процесор?
    * **Чрез времетраенето на цикъла на процесора, чрез честотата на използване на програмата от работния товар**, чрез броя процесорни цикли за изпълнение на програмата, чрез средния брой процесорни цикли за инструкции, чрез средния брой инструкции, изпълнени за един цикъл, чрез времето за изпълнение на програмата от работния товар
38. Как се указва ефективния адрес на операнда в паметта при непосредствен режим на адресация?
    * чрез адрес на регистър на микропроцесора, чрез стековия указател, **чрез адресното поле на операнда в кода на инструкцията**
39. Терминът „подобряване на производителността“ означава:
    * намаляване на потребителското процесорно време, намаляване на системното процесорно време, **намаляване на времето за изпълнение**, **увеличаване на пропускателната способност**
40. Суперскаларен процесор обработва два инструкционни потока (от прости и сложни инструкции). Коя от изброените фази на разработка е обща и за двата потока?
    * **извличане**, изпълнение, запис на резултата, декодиране
41. При коя от изброените структурни схеми за управление на паметта логическото и физическото адресно пространство се разделят на блокове с еднакъв обем (кратен на степените на 2)?
    * сегментно-странична организация, сегментна организация, **странична организация**
42. За коя от изброените компютърни архитектури се отнасят следните особености: Дължината на инструкциите е стотици битове. В рамките на процесора функционират паралелно множество функционални устройства. Всички функционални устройства споделят огромен общ регистров файл.
    * потокова, суперскаларна, **VLIW**, EPIC
43. Кои от изброените събития се определят като „изключения“?
    * **маскируеми прекъсвания**, **програмни прекъсвания**, **немаскируеми прекъсвания**, **капани**, повреди, грешки
44. В инструкционния конвейер се изпълняват едновременно:
    * празни инструкции, записи на данни от регистрите към процесора, **инструкции в различни фази**, инструкции за условен и безусловен преход
45. Кое от изброените не прави хипернишковата технология по-добра в сравнение с мултипроцесорите?
    * производителността, цената, стената на паметта, **консумираната мощност**, по-високите тактови честоти на работа на процесорите, термалната стена
46. Кои от изброените техники се използват при разрешаването на проблеми при конвейерното изпълнение на инструкции с преход?
    * дублиране на конвейерните ресурси, **множествено предварително изпълване на инструкции**, Prediction Look-up Table, използване на специализирани функционални устройства за четене на операциите или запис на резултатите в паметта, BTB buffer
47. Механизмите за ресурсен мениджмънт на многоядрени платформи осигуряват:
    * цялостни решения, имплементирани в софтуера, **примитиви за изграждане на стратегиите за мениджмънт**, специализирани подпрограми за ресурсен мениджмънт
48. Как се преодоляват конфликтите за памет при конвейеризираното изпълнение на инструкционния поток?
    * чрез въвеждане на закъснения и модифициране изходната таблица на заетост на конвейера, **чрез предсказване на преходите в инструкционния поток**, чрез пренареждане на инструкционния поток от компилатора, чрез използване на специализирано функционално устройство за четене на операциите или запис на резултати в паметта
49. Посочете кои от изброените фактори са причина за понижаване на потоковата скорост при конвейерната обработка
    * **зависимости по данни**, предсказване на прехода в информационния поток, темп на инициализация на конвейера, **конфликти при достъп на общи ресурси**, патентност на конвейерната обработка, **анти-зависимости по данни**, използваемост на функционалните устройства по конвейера, **преходи в информационния поток, засягащи модифицирането на програмния брояч**
50. Кои от изброените методи се използва за решаване на проблема с инструкционните хазарти?
    * презареждане на конвейера, увеличаване размера на буферната памет в отделните функционални устройства, опростяване формата на инструкциите използвани от програмата, разделяне на инструкционния цикъл на допълнителни фази на обработка, **пренареждане на инструкции от компилатора**
51. Кои от изброените алгоритми за управление на заместването на блокове при кеш-паметите се основават на оценка на вероятността за обръщение към блока в бъдеще време?
    * алгоритъм за случайния избор RC (random choice), алгоритъм FIFO, Алгоритъм LFU (Least Frequently Used), Алгоритъм LRU (Least Recently Used), **нито един от посочените**
52. От какво се определя времето за изпълнение на програмата в процесора? 1. алгоритъма, 2. програмния език, 3. компилатора, 4. системата процесорни инструкции
    * **1 и 3**, 3 и 4, 1, 2, 3 и 4, 1 и 2, 2 и 3, 2, 3 и 4
53. Кои процесори от изброените процесорни фамилии се характеризират с RISC-архитектура?
    * AMD, **SPARC**, ALPHA, CYRIX, INTEL, **ARM**, **MIPS-R4000**
54. На какво се дължат процедурните зависимости?
    * единствено на инструкции за безусловен преход, единствено на инструкции за условен преход, **инструкции за безусловен преход и инструкции с осъществен условен преход в инструкционния поток на програмата**, едновременни заявки за един и същ ресурс
55. Кой от посочените методи не е метод за преобразуване на виртуални адреси?
    * Метод на странична организация, метод на сегментна организация, **метод на локалността (по време и пространство)**, метод на сегментно-странична организация
56. Колко е максималният брой страници с размер на страниците 4k при 32-битов адрес?
    * **1M**, 64k, 5k, 20k
57. Колко е максималният брой страници с размер 8 при 8-битов адрес?
    * 16, **32**, 64, 512, 4k
58. При размер на страниците 8 и 8-битов адрес, кой ред на таблицата съдържа адреса на страницата?
    * **3**, 5, 7, 10, 12
59. При размер на страниците 8 и 8-битов адрес, преобразувайте логическия адрес 18 във физически?
60. Какво съдържа TLB-буфера при странична организация на паметта?
    * **моментна карта на състоянието на страниците на оперативната памет, принадлежащи на даден процес**, таблица с номера на физически страници, принадлежащи на даден процес, таблица с номера на виртуални страници, принадлежащи на даден процес, таблица на страниците на диска
61. Какво е предназначението на TLB-буфера?
    * **да минимизира загубата на бързодействие при търсене на физическото разположение на данните**, да преобразува виртуалните странични адреси във физически, да намали производителността на хардуерна имплементация
62. Коя от изброените характеристики не описва таблицата на страниците?
    * таблицата на страниците се състои от редове, всеки ред от таблицата съответства на номер на виртуална страница, таблицата на страниците е допълнена с бит за присъствие, показващ дали съответната страница се намира в свободната памет, таблицата на страниците се пази някъде в оперативната памет, **таблицата на страниците съдържа информация за активните процеси**
63. Какви типове памет се използват за съхранение и поддържане на таблиците на страниците при статична организация на паметта?
    * кешова с директна организация, **асоциативни кешове**, външна дискова памет, **множество асоциативни кешове**, главна памет
64. Посочете при конвейерната обработка на коя от дадените последователности от инструкции възниква хазарт от типа write-after-read:
    * ADD R3, R2, R1; R3 = R2 + R1, **ADD R3, R2, R1; R3=R2+R1 SUB R2, R3, 1; R2=R3-1 ADD R3, R2, R5; R3 = R2+R5**, ADD R3, R4, R5 SUB R2, R2, 1 BEQ R2, R0, L1
65. Посочете при конвейерната обработка на коя от дадените последователности от инструкции възниква хазарт от типа write-after-read:
    * **ADD R3, R2, R1; SUB R2, R3, 1; ADD R3, R2, R5**, ADD R3, R4, R5; SUB R2, R2, 1; BEQ R2, R0, L1, ADD R3, R2, R1; SUB R2, R3, 1
66. Знаейки, че ускорението е правопропорционално на броя фази на конвейера, защо реално полученото ускорение е по-малко в сравнение с теоретично достижимото?
    * **заради хазартите**, заради по-високата патентност за индивидуалните инструкции, **заради времето, необходимо за запълване на конвейера**, заради еднаквото време за изпълнение на инструкции
67. Разделянето на инструкциите на отделни фази на обработка е характерно за:
    * **конвейерната обработка**, системната обработка, **поддържането на паралелизъм на ниво инструкции**, паралелната обработка, разпределената обработка
68. Колко машинни цикъла ще са необходими за конвейерна обработка на 1500 инструкции в 5-фазен линеен инструкционен конвейер?
    * 7500, 1505, 18, **1504**
69. Колко машинни цикъла ще са необходими за конвейерна обработка на 9200 инструкции в 7-фазен линеен инструкционен конвейер?
    * 64400, **9206**, 9153, 9207
70. Какво означава абревиатура ‘CISC’?
    * Control Independent System Computer, **Complex Instruction Set Computers**, Continual Input System Computer
71. Коя е правилната дефиниция за термина RISC?
    * **Reduced Instruction Set Computing**, Random Index System Call, Risk Instruction Stack Call, Real Invested Space Capacity
72. От какво се предизвикват ресурсните конфликти?
    * недобро планиране на ресурсите на инструкционния конвейер, **едновременни заявки за едни и същи ресурси**, от възникнали мехури в конвейера
73. Как се разрешават ресурсните конфликти?
    * **чрез дублиране на ресурсите**, само от инструкции за условен преход, чрез инструкции за безусловен преход, чрез инструкции за безусловен и условен преход
74. Кои от изброените видове памет се използват за изграждането на кеш памети?
    * SIMM, SDRAM, DIMM, DRAM, **SRAM**, EEPROM
75. Кой от трите метода на разполагане блокове от оперативната памет се имплементира най-лесно?
    * **директно (пряко) съпоставяне**, пълна асоциативност, множествена асоциативност
76. Посочете за кой от изброените режими на адресация се отнася показаната фигура?
    * |КОД|ОПЕРАНД|: индексна, косвена, вложена, **непосредствена (пряка)**
77. Как се оказва стойността на операнда при регистров режим на адресация?
    * **с номер на регистъра**, задава се в кода на инструкцията, задава се чрез константно отместване спрямо началото на програмния сегмент
78. В зависимост от режима на адресация времето на изпълнение на една процесорна инструкция може да варира в значителна степен. При коя от изброените подредби на адресни режими времето ще се изменя от много кратко до много дълго?
    * директна, косвена, непосредствена, непосредствена, косвена, директна, **непосредствена, директна, косвена**, косвена, непосредствена, директна
79. При кой от изброените методи на адресация в адресната част на инструкцията е записан адресен указател?
    * **индиректна адресация**, индиректна адресация с регистри, директна адресация, непосредствена адресация, индексна адресация
80. Къде е местоположението на операнда при регистрово косвена адресация?
    * в регистър, **в основната памет**, във вторичната памет, в кода на инструкцията
81. От какъв тип е следната инструкция. Събира съдържанието на адрес 5 и 15 и записва резултата в адрес 20 (ADD, 5, 15, 20)?
    * VLIW, RISK, **CISC**, EPIC
82. При събиране на floating point числа, кое действие се извършва първо с мантисата?
    * нормализира се, **измества се на дясно**, събира се, умножава се
83. При умножение на floating point числа, какво действие се извършва с експонентите.
    * умножават се, **събират се**, изместват се наляво, изместват се надясно, нулират се, изваждат се, наместват се взаимно
84. По какво се различават CISC от RISC?
    * **всички инструкции се характеризират с опростен формат**, инструкциите преобладаващо са от тип памет-памет, всички инструкции се характеризират с една и съща дължина, сложните инструкции се синтезират от по-простите, операциите с паметта са само от типа "Load" и "store"
85. Кои от изброените характеризира архитектурната концепция EPIC?
    * **оползотворяване на явен паралелизъм на ниво инструкция**, оползотворяване паралелизъм на ниво инструкция чрез изпращане на няколко инструкции на обикновения инструкционен поток в различни функционални устройства на конвейера, **оползотворяване на скрития в програмния код на ниво компилатор и апаратна поддръжка**, едновременно зареждане за изпълнение на няколко инструкции и динамично планиране, **използване на широка шина за зареждане на инструкция и дълги конвейери с голяма задръжка**
86. Кои от изброените техники служат за увеличаване степента на паралелизъм по време на изпълнение на програмата?
    * нито едно от изброените, **прогнозиране посоката на преходите, едновременно зареждане за изпълнение на няколко инструкции и динамично планиране, откриване и премахване на зависимости по данни при компилиране, реорганизация на циклите по такъв начин, че всяка итерация в получения код да се състои от инструкции, които са избрани от различни итерации на първоначалния цикъл (loop unrolling)**
87. Коя е правилната дефиниция за термина SIMD?
    * **Single Instruction, Multiple Data**, Single Interrupt, Multiple Distribution, Single Input, Multiple Distributions, Single Integration, Multiple Dynamics
88. При Signed – 2’s complement кодиране, какъв е резултатът от операцията 3 – 5?
    *   
      **1110**, 1001, 0001, 1000
89. При Signed – 2’s complement кодиране, какъв е резултатът от операцията 0 – 1?
    *   
      1001, **1111**, 1000, 0001
90. Коя логическа функция се имплементира чрез следната схема?
    *   
      NOR, **NOT**, NAND
91. При Vin=0, каква стойност ще се появи на изхода Vout?
    *   
      **0**, не може да се определи, 1
92. При V1=0 и V2=0, каква стойност ще се появи на изхода Vout?
    *   
      0, не може да се определи, **1**
93. При V1=0 и V2=1, каква стойност ще се появи на изхода Vout?
    *   
      **0**, не може да се определи, 1
94. При стойности на входовете A=0 и B=0 и Carry in = 0, колко стойността на изхода sum?
    *   
      **0**, не може да се определи, 1
95. За даденото съдържание на инструкционния буфер и разместено изпълнение (out of order), посочете коя/кои от съдържащите се в него инструкции е готова за изпълнение.
    * Картина, която съдържа маса

      Описанието е генерирано автоматично  
      Opcode 3, 4 и 5, Opcode 1, 3 и 4, **Opcode 4**, Opcode 1, 2 и 5, **Opcode 3**, Всички, Нито една
96. Каква е функцията на битовете ALU от регистъра MiR?
    *   
      посочват адреса на следващата инструкция, задържат информация за състояние за процесора, **избират функцията на ALU и Shifter**, определят адреса на операндите
97. Защо шината MaR с бял цвят е еднопосочна?
    *   
      защото регистъра е само за четене, **защото трябва да изберете адрес от паметта**, за да се пести памет, защото ALU е 8 битово, защото регистъра е инвестиран
98. Защо шината PC с бял цвят е еднопосочна?
    *   
      защото регистъра е само за четене, **защото трябва да изберете адрес от паметта**, за да се пести памет, защото ALU е 8 битово, защото регистъра е инвестиран
99. Защо шина A към ALU е двупосочна?
    *   
      защото регистърът е инвестиран, защото ALU е 8 битово, защото от шина B се предават към шина A, защото изхода се праща към регистър H, **защото е объркана схемата**
100. Коя от показаните схеми илюстрира директен режим на адресация?
     *   
       **A**, B, C
101. Посочете за коя от показаните схеми на адресация илюстрира режим на адресация с регистри?
     * 
     *   
       A, **B**, C
102. Посочете за кой от изброените режими на адресация се отнася дадената фигура?
     *   
       **адресация с изместване**, индиректна адресация с регистри, вложена адресация, адресиране на регистри
103. За компютърна система с дадената архитектура е известно, че зареждането на данните във входните регистри отнема 5 ns, изпълнението на АЛО-то 10 ns и записването на резултата 5 ns. Колко е максимално достижимата производителност, измерена в MIPS?
     *   
       100, 1, 10, **50**, 200
104. За компютърна система с дадената архитектура е известно, че зареждането на данните във входните регистри отнема 10 ns, изпълнението на АЛО-то 20 ns и записването на резултата 20 ns. Колко е максимално достижимата производителност, измерена в MIPS?
     *   
       100, 200, 10, 1, **20**
105. За компютърна система с дадената архитектура е известно, че зареждането на данните във входните регистри отнема 20 ns, изпълнението на АЛО-то 30 ns и записването на резултата 20 ns. Колко е максимално достижимата производителност измерена в MIPS?
     *   
       **100**, 1, 10, 20, 200
106. За компютърна система с дадената архитектура е известно, че зареждането на данните във входните регистри отнема 3 ns, изпълнението на АЛО-то 4 ns и записването на резултата 3 ns. Колко е максимално достижимата производителност, измерена в MIPS?
     *   
       10, 100, 1, 200, 20
107. При зададени стойности на управляващите сигнали A = 0, B = 0 и C = 0, кой от входовете ще се свърже към изхода?
     *   
       D7, **D0**, D1, D3
108. При зададени стойности на управляващите сигнали A = 0, B = 1 и C = 1, какъв резултат ще се появи на изхода при D1 = 1, D2 = 0, D3 = 1, D4 = 1?
     *   
       0, не може да се определи, **1**
109. При F0 = 0 и F1 = 0, коя функция се активира?
     *   
       **AB**, B, A+B, Sum
110. При F0 = 0 и F1 = 1, коя функция се активира?
     *   
       AB, B, **A+B**, Sum
111. При активирана функция A + B и INVA = 0, A = 0, ENA = 1, B = 0, ENB = 1, каква е стойността на изхода?
     *   
       **0**, не може да се определи, 1
112. При активирана функция AB и INVA = 0, A = 0, ENA = 1, B = 0, ENB = 1, каква е стойността на изхода?
     *   
       **0**, не може да се определи, 1
113. При активирана функция AB(F0 = F1 = 0) и INVA = 0, А = 1, ENA = 1, B = 0, ENB = 1, каква е стойността на изхода?
     *   
       0, не може да се определи, **1**
114. Какъв тип организация на кеш памет е илюстриран на дадената фигура?
     * 
     *   
       **кеш с директна организация**, кеш с множествено асоциативна организация, кеш с асоциативна организация
115. Какъв тип организация на кеш памет е илюстриран на дадената фигура?
     *   
       кеш с директна организация, кеш с множествено асоциативна организация, **кеш с асоциативна организация**
116. Какъв тип организация на кеш памет е илюстриран на дадената фигура?
     *   
       кеш с директна организация, **кеш с множествено асоциативна организация**, кеш с асоциативна организация
117. Кои от показаните на фигурата адресни полета подлежат на преобразуване при преобразуване на виртуалния адрес в реален?
     * 
     *   
       полетата и на страницата и на линията в адресите, полето на страницата в адресите, полето на линията в адресите
118. Посочете, какъв блок трябва да има на мястото на жълтия правоъгълник ако представената схема илюстрира вътрешната архитектура на програмируем В/И контролер за прекъсване.
     * Картина, която съдържа текст, табло

       Описанието е генерирано автоматично
     *   
       брояч на данните, регистър за данните, регистър за адресите, регистър на маските, **регистър на заявките**
119. Посочете, какви блокове трябва да има на мястото на жълтите правоъгълници в дадена схема на шинна архитектура за сървърна система.
     * 
     *   
       мрежов адаптер, **мост към шина PCI**, **мост**, шинен арбитър, **мост към шина за разширение**
120. Каква е посоката на предаване по адресните линии, свързващи модулите памет и вход-изход с адресната шина?
     * 
     *   
       **еднопосочно, от адресната шина към модулите**, двупосочно, еднопосочно, от модулите към адресната шина
121. Обяснете схемата и основните принципи на действие (USB):
     * 
     * На схемата е изобразена йерархичната структура на USB шината. Шината поддържа свързване на до 127 устройства – хъбове (hubs) или крайни устройства (nodes). Хъбовете са устройства, в които имаме вход и повече от един изход. При USB шината устройствата се идентифицират от хоста (централен хъб), т.е. когато се включи ново устройство, хостът му дава адрес от 1 до 127.
122. Обяснете схемата и основните принципи на действие (Pipelining)
     * 
     * На схемата е изобразена структурата на инструкционния конвейер. Конвейерът прави по-бързо изпълнението на група инструкции – изпълняват се едновременно инструкции в различни фази. Между обектите в различните стъпки няма взаимно влияние, като времето за минаване през всеки етап е едно и също за всички. При конвейера от схемата инструкциите минават през 3 етапа: извличане, дешифриране и изпълнение и записване и write-back.
123. Обяснете схемата и основните принципи на действие (complex pipelining):
     * 
     * На схемата е изобразен пример за инструкционен конвейер. С цел да се повиши производителността и да се постигне едно и също време за изпълнение на всички операции, в процесора присъстват множество устройства за пресмятане на различни операции. Въвежда се едно допълнително стъпало („Issue“), което предава инструкциите към функционалните блокове и следи за това кое устройство работи и кое е празно. ALU-то работи по-бързо от устройствата с плаваща запетая и конвейера трябва да предвиди възможността 2-та инструкция да прескочи първата и това не трябва да се отрази на резултата.
124. Обяснете схемата и основните принципи на действие (fully associative cache):
     * 
     * На схемата е показана структурата на кеш памет с пълна асоциативност - няма индекси, тага стои едновременно във всички линии. При този вид кеш всеки блок от ОП може да бъде свързан към всеки фрейм от кеша. За да бъде намерен точният блок обаче се налага да се претърси паметта на целия кеш. Следователно, колкото по-голяма е кеш паметта, толкова по-голямо е забавянето, преди нужният байт да стане достъпен за процесора. Предимството на този вид кеш е, че нещо се маха от кеша, чак когато той е пълен.
125. Обяснете схемата и основните принципи на действие (direct mapped):
     * 
     * На дадената схема е изобразена структурата на кеш памет с директно съответствие. При този вид кеш всеки блок може да е в точно определена линия, което се определя по индекси (имаме множество от индексирани фреймове и блокът се поставя на точно определеното място). Голям плюс на този вид кеш е, че потенциалните положения на всеки блок са значително редуцирани и съответно броят на търсенията е много по-малък от този при други типове кеш. Недостатъкът обаче е потенциалното получаване на колизия – когато на процесорът са му нужни блокове в едни и същи линии. Тогава ефективността от кеширането рязко спада.
126. Обяснете схемата и основните принципи на действие (2 way set cache):
     * 
     * На схемата е изобразена структурата на двупътно асоциативен кеш. Този вид кеш памет е подобен на кеша с директно съответствие. Тук имаме няколко индексирани множества с равен брой фреймове. За това къде трябва да се постави новият блок се ориентираме по индекса на множеството.
127. Обяснете схемата и основните принципи на действие (Translation):
     * 
     * На схемата е изобразен процесът по преобразуването на виртуалния адрес в реален физически адрес. Таблицата на страниците показва в кой реален адрес се намира виртуалната страница. Започваме с търсене в TLB. Ако е имало такава транслация (hit), то направо се генерира реалният адрес, няма забавяне. Ако няма направена такава транслация (miss), то трябва да се проверят таблиците в паметта и т.н. Накрая ОС обновява всички таблици.
128. Обяснете схемата и основните принципи на действие (address and page of user):
     * 
     * Класическата виртуална памет има таблица на страниците, която се пази някъде в реалната ОП. PT Root е фиксиран регистър в процесора, в който се държи адресът на началото на таблицата. Съдържанието на този регистър може да бъде променяно. Page Table-a се състои от редове – всеки ред от таблицата съответства на номер на виртуална страница.
129. Обяснете схемата и основните принципи на действие (DRAM array):
     * 
     * DRAM паметта се представя като матрица (с равни редове и колони). Aдресът е разделен на 2 части, като първо се чете редът, а после колоната. Частта от адреса, която отговаря за реда, се подава на RAS. Прочита се целия ред и се записва в буфера на реда. Чака се да се върне отговор от RAS, за да се изпълни CAS - който прочита частта от адреса, която отговаря за колоната.
130. Обяснете схемата и основните принципи на действие (wb-buf):
     * Картина, която съдържа часовник, знак, метър, улица

       Описанието е генерирано автоматично
     * Write-back buffer - Буфер за писане, който позволява да се извърши заместване преди writeback. Ако нещо липсва в кеша, влизането в буфера WB се брои за случило се. WB буфера е част от йерархията на кеша. Накрая кеша се обновява, а буфера се изпразва.
131. Обяснете схемата и основните принципи на действие (RenamingTable):
     * 
     * Renaming table - За да се реши проблема с WAR (write after read) и WAW (write after write) се създава един реордер буфер, в който се преименуват регистрите. Той събира резултатите от изпълнението на функционалните блокове и ги пуска за commit. За програмиста процесора разполага с 16 регистъра. Предложението е вътре в процесора да има 2-3 пъти повече (вътрешни регистри). Това прави невъзможно появата на различните рискове. В буфера се слагат инструкциите и диспечера ги пуска ако са им готови операндите независимо дали и е дошъл реда или не. Колко инструкции могат да бъдат в конвейера зависи от: 1) броя на регистрите, 2) контролните инструкции.
132. Обяснете схемата и основните принципи на действие (Basic Read Transaction):
     * 
     * По време на фронта на 2-я такт всичко е стабилно. IRDY# чака DEVSEL#. IRDY става активен(пада надолу) когато усети, че има DEVSEL. По време на втория такт нищо не става, понеже TRDY# го няма. Мастера ще ги прочете при 4. На 6 минава втора шина данни, след което мастера докато ги обработва прави пасивен сигнал, и таргета е извадил трета шина и понеже няма IRDY ги държи. Фрейма е свален още на 7, но DEVSEL още не е завършило работа. Има една адресация и три фази данни. Read-транзакциите отнемат минимум 3 такта.
133. Обяснете схемата и основните принципи на действие:
     * 
     * Устройство за предсказване на преходите – това устройство позволява да се предвиди коя инструкция ще се изпълни при прехода още преди неговото действително изпълнение. За предсказване на прехода се използват различни стратегии за предсказване на преходите (branch prediction), които могат да бъдат статични и динамични. Основната разлика е, че при динамичните стратегии за предсказване на прехода се следи и се взима предвид поведението на програмата до стартирането на конкретната инструкция за условен преход.
     * Най-често използваните статични стратегии за предсказване на прехода се основават на следното:
     * • Приема се, че преходът никога не се осъществява и се извличат следващите инструкции;
     * • Приема се, че преходът винаги се осъществява и се извличат инструкцията-цел на прехода и последователността от инструкциите след нея;
     * • Решението зависи от операционния код на инструкцията за условен преход.
134. Какво представлява и какви са предимствата на пакетираните SIMD типове данни? - В един регистър могат да се запишат няколко инстанции от данни. Пример: MMS инструкциите за графична обратботка. Наричат се също векторна обработка. Покачва се производителността. ММХ типът е разработен, за да ускори мултимедийните програми и програмите за комуникация като включват нови инструкции и типове данни които позволяват на програмите да достигат ново ниво на производителност. Основните факти за този тип са: една инструкция, много информация (Single Instruction, Multiple Data (SIMD)); Четирите ММХ типа данни са: пакетиран байт (Packed byte) – 8 байта пакетирани в 64 бита; пакетирана дума (Packed word) – 4 16-битови думи пакетирани в 64 бита; пакетирана двойна дума (Packed doubleword) – 2 32-битови двойни думи пакетирани в 64 бита; четворна дума (Quadword) – 64 бита. XMM-тип :Типовете пакетирани данни за XMM са: 4 32-битови single-precision числа с плаваща запетая; 2 64-битови double-precision числа с плаваща запетая; 16 8-битови байтове или символи; 8 16-битови short integer; 4 32-битови integer; 2 64-битови integer.
135. Опишете сегментните регистри и тяхното предназначение - Тези специални регистри позволяват на дизайнерите на системния софтуер да използват различни модели на организация на паметта. Тези шест регистъра определят в определено време кои сегменти на паметта са адресируеми в дадения момент. Флагов сегмент ,Кодов сегмент. Сегментните регистри са четири типа: CS -кодов сегмент; DS -сегмент за данни; SS -стеков сегмент и ES - допълнителен сегмент за данни. Същественото предимство на образуването на сегменти е възможността за преместване на програмите и данните в паметта. Това свойство се нарича преместваемост - relocatability.
136. Какво е характерно за регистрите с общо предназначение при 64-битов режим? - Регистрите за общо предназначение на IA-32 и Intel-64 архитектурата са EAX, EBX, ECX, EDX, EBP, ESP, ESI и EDI. Тези регистри се използват за съхранението на операндите на логическите и аритметични операции. Те също така могат да бъдат използвани за операнди на изчисленията на адрес (освен че ESP не може да бъде използван като индексен операнд)
137. Приложения на управляващи регистри - CR0, CR1, CR2, CR3, CR4 са 32-битови регистри и се използват за задаване на различни режими на работа на микропроцесора, базови адреси на таблиците с каталога на страниците при странична организация на паметта и някои разширени функционални възможности на микропроцесорите с архитектура IA-32 и Intel-64. CR0 съдържа системни флагове, които контролират или указват условията, които се отнасят към системата като цяло, а не към отделна задача. Регистърът CR2 се използва за обработка на странични прекъсвания, при условие че страницирането е задействано. В такъв случай CR2 съдържа адресът, от който се е получило страничното прекъсване. Регистърът CR3 също като CR2 се използва само, ако страницирането е задействано. В такъв случай, CR3 позволява на процесора да намери PTD (Page Table Directory) за текущата задача. Регистърът CR4 се използва в защитен режим на работа за контролиране на редица операции, като поддръжка на режим на виртуален 8086, задействане на входно/изходни прекъсвания и други.
138. Какви полета съдържа локалната дискрипторна таблица? - Потребителската програма се разполага на части в оперативната памет и за всяка такава част се създава един ред от една таблица – Локална Дескрипторна Таблица (ЛДТ). Може да има максимум 8152 части (реда). Всеки ред от тази таблица се нарича „дескриптор”. Операционната памет има една, т. нар. „Глобална Дескрипторна Таблица”, всеки ред на която сочи началото на всеки ЛДТ на тази програма. Тоест, ако има 500 програми, ще има 500 ЛДТ и 500 реда в ГДТ. Всеки елемент от ДТ се нарича „дескриптор” (8 байта). полето Base задава адреса на началото на сегмента. То е разделено на 3 части; полето Limit задава размерът на сегмента; то е 20 бита, така че максималният размер на сегмент е 1 000 000 единици; битът DB задава размерът на операндите и адресите за този сегмент; битът P е бит за наличност (present); битовете DPL са битове за нивото на привилегии на дескриптора - от 0 до 3; битът DT определя типа на дескриптора
139. Каква информация се съдържа в task register - Task Register (TR) - съдържа 16 битов сегментен селектор, базов адрес (32 бита в защитен и 64 бита в IA-32e режим), размер на сегмента и атрибути на Task State Segement (TSS) на текущия процес. Селекторът реферира TSS дескриптор в глобалната таблица GDT. Когато започне изпълнението на нов процес (Task) се зарежда базовия адрес и размера на съответния сегмент. Едва когато се смени процесът, то тогава се променя и сегментния регистър. По този начин, при следващо обръщение, не е необходимо да се достъпва глобалната таблица. Когато обаче се смени сегментния селектор е необходимо отново да се заредят данните от съответния дескриптор и да се запишат в скритата част (Segment Descriptor Registers).
140. За какво и как се ползва префиксът? - Чрез префикса може да се променя поведението на инструкцията, като например да се сменя сегмента, размера на думата, да се управлява цикличното изпълнение и използването на шината. Инструкцията може да има по един префикс за всяка от четирите групи или общо четири байта. ∙ Group 1: LOCK, REPE/REPZ, REP, REPNE/REPNZ ∙ Group 2: CS, DS, ES, FS, GS, SS, Branch hints ∙ Group 3: Operand-size override (16 bit vs. 32 bit) ∙ Group 4: Address-size override (16 bit vs. 32 bit)
141. Поведение на REX prefix? - В 64 битовия режим, към формата на инструкцията, се добавя REX префикс. Не всички инструкции в 64 битов режим изискват REX префикс. Той е необходим единствено, ако инструкцията реферира някой от разширените регистри или използва 64 битови операнди. REX префикса съдържа битове, които разширяват ModR/M полетата. По този начин ще работят и старите и новите програми, които използват повече регистри. Полето reg е за индексния xрегистър. Разширяват се също полетата на SIB.
142. Обяснете преобразуването на логическия адрес във физически при сегментно страничната организация на паметта - При защитен режим сегментният селектор се взема от сегментния регистър. Този сегментен селектор съдържа индекс в главната дескрипторна таблица, а съответният дескриптор съдържа адреса на сегмента. Към този адрес се добавя ефективния адрес от инструкцията (отместването) и се получава линейния адрес. Ако е включено страничното преобразуване, линейният адрес се дели на три части. В най-лявата е адресът в каталога на таблиците, който съдържа 1024 таблици. Всяка от тези таблици описва по 1024 страници по 4Kb. Към началото на страницата се добавя отместването и се получава физическия адрес.
143. Предназначения на сегментните селектори - Сегментните селектори са 16 битови идентификатори на сегмента. Старшите 13 бита се използват за определяне на отместването в глобалната или локалната дескрипторни таблици. Трите младши бита имат по-специфично предназначение. В зависимост от флага TI се адреси глобална или локална таблица. Полето RPL съдържа необходимото ниво на привилегия, като за да бъде позволен достъп, стойността на (DPL — Descriptor privilege level) трябва да е по-малка или равна от RPL. Всеки сегментен селектор има видима и скрита част. При първото обръщение към дескриптора се извлича информацията и се записва в скритата част на сегментния регистър. След това се използва, докато не се наложи да се смени сегмента.
144. Опишете полетата на флаговия регистър - Флагов регистър (EFLAGS (program status and control) register) – съдържа информация за статуса, т.е. изпълнението на програмите, които се изпълняват и позволява ограничен контрол върху процесора от страна на потребителските програми. Той се състои от еднобитови флагове, които се установяват на 0 или 1 от процесора, в зависимост от резултата от различни операции.
145. Опишете полетата, обособяващи формата на инструкциите - Инструкцията съдържа: незадължително поле префикс, код на операцията (до три байта), спецификатор формиращ адрес (не е задължителен) и съдържа ModR/M байт и понякога а SIB (Scale-Index-Base) байт, отместване (незадължително) и поле „непосредствен“ (незадължително). Максималният размер на инструкцията е ограничен от механизмите за защита до 15 байта. Инструкция с по-голяма дължина вероятно няма да се изпълни и ще генерира грешка.
146. Каква информация се съдържа в IDT? - Interrupt Descriptor Table (IDT) e таблицата на векторите на прекъсванията. Тя има 256 елемента. Векторите на прекъсванията не са дескриптори на сегменти, а са входове към такива.
147. Каква е фунцкията на челната част? - Челната част извлича и декодира инструкциите. Декодираните инструкции се транслират до микрооперации (μops). Основната задача на тази част е да осигурява непрекъснат поток от микрооперации на изпълняващата част. Челната част се състои от fetch/decode unit execution trace cache
148. Как работи изпълняващата част? - Изпълняващата част пренарежда микрооперациите по такъв начин, че тези чиито операнди са готови (и има налични изчислителни ресурси) да се изпълнят възможно по- скоро. Изпълняващата част може да обработи (issue) няколко микрооперации за един цикъл.
149. Каква е ролята на завършващата част? - Завършващата част се грижи за това, резултатът от изчисленията да съответства на реда на постъпване на инструкции и че състоянието на системата се обновява правилно.
150. Обяснете групите, на които се делят изключенията - aborts, faults, traps - Капанът (trap) e изключение, при което, след като се изпълни съответната програма за обработка, броячът на инструкциите се увеличава и сочи следващата инструкция. Т.е. текущата програма се прекратява и се продължава със следващата. По този начин например, когато стартирате програма, в която се появи изключение от този тип, то операционната система си връща управлението. Изключенията от тип грешки (fault) като цяло могат да бъдат коригирани и за това след като се случат и бъде изпълнена обработващата програма, се стартира отново същата програма (която е генерирала грешката), като се счита, че грешката вече е коригирана (от обработващата програма). За изключенията от тип abort не винаги се знае коя инструкция ги е предизвикала и не позволяват рестартиране на програмата или процесът, които ги причиняват. Използват се за регистриране на хардуерни грешки или недопустими стойности в системни таблици.
151. По какво се различават изключенията от тип faults от тип aborts? - За faults( и traps) се знае коя инструкция ги е причинила и програмата или процесът могат да бъдат коригирани и за това след като се случат и бъде изпълнена обработващата програма, могат да бъдат рестартирани без загуба, докато при aborts това може да не е ясно и следва прекратява на програмата или процесът.
152. Какво представлява и как работи предварителното извличане? - Предварителното извличане на данните в кеша (Cache prefetching) е техника, която увеличава производителността, като копира данните от бавна памет в по-бърза, още преди да са необходими. Най-често се използват потокови буфери (Stream buffers), които се основават на последователното изпълнение на инструкциите и извличане на данните. Предварителното извличане може да хване както нарастващи, така и намаляващи потоци. Когато се установи, че адресите (които процесорът вече е поискал) нарастват, то се извличат следващите няколко в последователността, още преди още преди да са поискани.
153. Как се осъществява превключването на задачите? - Когато дадена задача бъде извикана от друга такава, то тя се нарича вложена. Превключването към дадена задача се осъществява в един от следните четири случая: • Текущата програма, задача или процедура изпълнява JMP или CALL инструкция към TSS дескриптора в GDT на дадената задача. • Текущата програма, задача или процедура изпълнява JMP или CALL инструкция към TSS-gate дескриптор в GDT или текущата LDT на дадената задача. • Векторът на прекъсване или изключение сочи към TSS-gate дескриптор в IDT. • Текущата задача изпълни IRET инструкция, при условие че флагът NT в регистъра EFLAGS има стойност 1. При превключването към друга задача, състоянието на средата, в която се е изпълнявала текущата задача (състояние на задачата или още контекст на задачата) се запазва в TSS и изпълнението на задача се прекратява за някакъв период от време. След това информацията за предстоящата за изпълнение задача се зарежда в процесора и нейното изпълнение започва от инструкцията, сочена от вече зареденият регистър EIP на текущата задача.
154. Опишете режимите на адресация и ролята на modR полето - Това поле специфицира кои регистри или адреси от паметта да се използват като операнди на инструкцията. Полето Mod определя методът за адресация на операндите от паметта. По подразбиране, полето R/M съдържа първия операнд, а Reg/Opcode втория. Ако инструкцията изисква само един операнд, то полето R/M съдържа допълнителен код на операцията, вместо адрес на регистър.
155. Какво е характерно за защитния режим? - В защитен режим, всички обръщения към паметта се осъществят чрез глобалната или локалната дескрипторни таблици. Тези таблици съдържат входни точки (референции) към сегментните дескриптори. Всеки дескриптор се асоциира със селектор. Т.е. таблиците се намират в паметта, но адресите им са записани в регистри на процесора. В защитен режим, всяко изпълнение на програма се осъществява като задача.В защитен режим, винаги се използва някакъв вид сегментация и тя няма възможност да се забрани, докато използването на страницирането не е задължително и може да се конфигурира.
156. Обяснете структурата на задачата при IA 32 архитектура - Задачата се състои от пространство за изпълнение (execution space) и сегмент за състоянието на задачата (Task State Segment TSS). Пространството за изпълнение се състои от сегмент за кода, стека и няколко сегмента за данни. Най-общо казано, TSS съхранява в паметта информацията, необходима за изпълнението (execution space) на дадена задача. С други думи, в TSS (Task State Segment) се съхранява състоянието/контекста на задачата/процеса. Тази структура описва сегментите, управляващият регистър и общите регистри на задачата. Всяка задача, се определя от селектор към съответния и сегмент за състоянията TSS. Когато дадена задача бъде заредена в процесора за изпълнение, сегментния селектор, размера на сегмента и други атрибути на TSS се зареждат в регистъра на задачата (task register).
157. Какви класове инструкции знаем? - FPL инструкции (плаваща запетая), SSE - Streaming SIMD Extensions, MMX - целочислени инструкции, част от x86 процесорната архитектура, SSE2, SMXI
158. Адресиране на операнди при IA-32 и INTEL-64 ? - Операндите в паметта се реферират чрез сегмент и отместването в него. При 32 битовия режим адресът на операнда се състои от 16 битов сегментен селектор и 32 битово отместване в сегмента. При 64 битовия режим, сегментния селектор също е 16 бита, но отместването може да бъде до 64^2. В зависимост от самата инструкция и режимите на адресация, някои операнди могат да бъдат да бъдат специфицирани явно, а други скрито, като данните за източника или дестинацията могат да се намират в: ∙ в самата инструкция (непосредствен операнд) ∙ в регистър ∙ адрес в паметта ∙ входно/изходен порт Операндите в паметта се реферират чрез сегмент и отместването в него. При 32 битовия режим адресът на операнда се състои от 16 битов сегментен селектор и 32 битово отместване в сегмента. При 64 битовия режим, сегментния селектор също е 16 бита, но отместването може да бъде до 64^2. Отместването (Offset) може да бъде специфицирано като статична стойност (displacement) или да се изчисли с използване на следните компоненти ∙ Displacement — An 8-, 16-, or 32-bit value. ∙ Base — The value in a general-purpose register. ∙ Index — The value in a general-purpose register. ∙ Scale factor — A value of 2, 4, or 8 that is multiplied by the index value.
159. Формат на инструкциите при IA-32? - IA-32e се явява 64 битово разширение на IA-32 архитектурата. Процесорите с 64 битово разширение могат да работят в реален (real 8086) режим, IA32, или IA-32e режими. Режимът IA-32 позволява на процесора да работи в 64 битов режим и в режим на съвместимост (compatibility mode), което от своя страна позволява едновременното изпълнение на 32 и 64 битови програми. Т.е. режимът IA-32 включва два подрежима: 64 битов режим – позволява на 64 битови операционни системи да изпълняват 64 битово приложение Режим на съвместимост – Позволява 64 битови ОС да изпълняват повечето 32 битови програми. 64 битовия режим, към формата на инструкцията, се добавя REX префикс. Не всички инструкции в 64 битов режим изискват REX префикс. Той е необходим единствено, ако инструкцията реферира някой от разширените регистри или използва 64 битови операнди.
160. Обяснете принципа на работа на кеш-паметта и протоколите за процесори - Първо, по индекса (Index) се определя множеството, в което да се търси, след което в него се търси по тагът (Tag) и ако има съвпадение (hit), по отместването (Offset) се определят данните. Времето за достъп, което се измерва в машинни цикли, се различава в порядък. За да се запази консистентността на кеш паметта, архитектурата IA-32 поддържа протокол на кеша MESI. Той се използва както в едноядрени, така и в многоядрени процесори. При този протокол се проверяват четири дефинирани състояния на всяка линия на кеша – (Modified, Exclusive, Shared, Invalid).
161. Какво представлява сегментирането на паметта? Предимства и недостатъци - В някои случаи е удобно размерът на единиците памет да се определя от програмиста. Така по-лесно може да се структурира програмата чрез определяне размера на информационната единица. Основна информационна единица е сегментът (С). Сегментите са с променлива дължина, която се определя от самия програмист. В архитектурите IA-32 и Intel-64 са вградени два механизма за организация на виртуалната памет – сегментация (segmentation) и странициране (paging). При сегментацията адресът се получава от 16-битов сегментен регистър и 32-битово отместване. Сегментните регистри в архитектурите IA-32 и Intel-64 са следните: 1) CS – кодов сегментен регистър – показва началото на сегмента на кода на програмата; 2) SS – стеков сегментен регистър – показва началото на сегмента на стека; 3) DS, ES, FS, GS – сегментни регистри за данни – показват началото на четири сегментна за данни.
162. Как се изчислява адресът? - За да се адресира байт в даден сегмент, е необходим логически адрес, който се състои от сегментен селектор и отместване. Селекторът е идентификатор на сегмента и дава отместването в таблица (Global Descriptor Table, GDT), съдържащата структура наречена сегментен дескриптор, в която е описан сегмента.
163. Опишете последователността от стъпки поради прекъсване от изключения? - Прекъсванията и изключенията карат процесора да скочи (jump) в програма, която обработва прекъсването (interrupt handler), след като запази контекста (за да може прекъснатата програма да продължи). Прекъсванията и изключенията имат идентификатор (ID), наречен вектор, който определя адреса на програмата за обработка на прекъсването. Тези програми (interrupt handler) се извикват през таблица на дескрипторите на прекъсванията (Interrupt Descriptor Table).
164. Представете хсема на паметта при big endian и little endian записване на данните. - Big-endian – Писане на десетично число като 123 на хартия – най-висшият digit ще бъде записан най-вляво, писането започва от ляво на дясно, т.е ще се прочете/запише 123.Little-endian – Точно обратното, най-висшият ще остане най-вдясно, писане отдясно наляво – 321.
165. Какво е характерно за адресното пространство (при някоя конкретна архитектура)? - При плоския модел цялото адресно пространство е линейно от адрес 0 до адрес MaxBytes-1.
166. Механизъм на обновяване на страница от диск - При използване на страницирането, всеки сегмент се разделя на страници (обикновено по 4KB всяка), които могат да се намират във физическата памет или на диска.
167. Опишете режимите на адресация при IA-32 и Intel-64 - При адресацията на IA-32 процесорен набор инструкции една инструкция може да оперира върху нула или повече операнди. Пример за инструкция без операнди е NOP (no operation). Един операнд може да бъде във всяка от следните позиции: В инструкцията (immediate operand), В регистър, В паметта, Във входно изходен порт. Достъпът до операндите в инструкцията и в регистрите може да бъде осъществяван много по-често от операндите в паметта, понеже операндите в паметта трябва първо да бъдат извлечени от паметта. - x86-64 процесор се държи като IA-32 процесор когато е включен в реален или защитен режим. Това са режими които се поддържат когато процесора не е в long режим. Докато големината на регистрите е увеличена до 64-бита в сравнение с предишната x86 архитектура, адресирането на паметта все още не е увеличено до пълните 64 бита.
168. Защо D+ и D- са усукани?
     * За по-добра гъвкавост на кабела, за по-голяма здравина, **за по-добра шумоустойчивост**, за повишаване на скоростта
169. Защо има по една стрелка във всяка посока?
     * 
     * Защото не достига честотната лента, за да се постигне по-висока скорост, **всички са верни**, защото са диференциални двойки
170. Защо се налага използването на повече линии?
     * Защото не достига честотната лента, за да се постигне по-висока скорост, **всички са верни**, защото са диференциални двойки
171. С каква цел се използва инструкцията CMOVZ?
     * **Редуциране на условните преходи**, предотвратяване на хазартите, редуциране на обръщенията до паметта, намаляване на закъснението при четене или запис
172. Как се случват сигналите REQ и GNT във времето?
     * Получава се GNT, след което се получава REQ, **Получава се REQ, след което се получава GNT**, един машинен цикъл след активиране на REQ се активира и GNT
173. Кое е означеният с „A“ слоя в PCI протоколния стек?
     * 
     * 
174. Кой е елементът обозначен с буквата A?
     * 
     * 
175. Кой е елементът означен с „A“?
     * 
     * 
176. Кой е елементът означен с буквата A на схемата?
     * 
     * 
177. Кои от изброените абревиатури са имена на шини?
     * PWM, I2C; CAN, ADC, UART; DAC, USB, CAN; **USB, CAN, I2C**
178. Кой фрейм е за четене?
     * 
     * **Frame 1**, Frame 2, Frame 3, Нито един от изброените
179. Кой фрейм е за запис?
     * 
     * Frame 1, Frame 2, **Frame 3**, Нито един от изброените
180. Какъв е приоритетът на прекъсванията представени на фигурата (От най-висок към най-нисък)?
     * 
     * Printer > RS232 > Disk, RS232 > Printer > Disk, Printer > Disk > RS232, Disk > RS232 > Printer, **RS232 > Disk > Printer**
181. Какъв е приоритетът на прекъсванията представени на фигурата (От най-нисък към най-висок)?
     * 
     * Printer > RS232 > Disk, RS232 > Printer > Disk, **Printer > Disk > RS232**, Disk > RS232 > Printer, RS232 > Disk > Printer
182. Кой установява сигналите за данни на шината?
     * 
     * **Master**, Slave, Arbiter
183. По времето между T3 и T4, кой установява сигналите за данни на шината.
     * 
     * Master, **Slave**, Arbiter
184. Според представената на фигурата времедиаграма на PCI сигналите се осъществява:
     * 
     * **Четене**, Запис, Празен запис, Обратен запис, Изпреварващо четене
185. Според представената на фигурата времедиаграма на PCI сигналите се осъществява:
     * 
     * Четене, **Запис**, Празен запис, Обратен запис, Изпреварващо четене
186. При USB комуникация, за какво служат резисторите представени на схемата?
     * 
     * 
     * За повишаване на скоростта, за по-добра шумоустойчивост, **за определяне на скоростта**, за повишаване на обхвата
187. За какво служат резисторите?
     * **За създаване на електрическо съпротивление**, за по-бърза връзка с паметта, за защита от удари, за ускоряване на връзката между шините и процесора
188. Какво е предназначението на Predicate?
     * **Да опише даден процес**, да оптимизира операция, да намали латентността, да ускори работата с процесора
189. Кой от посочените компоненти осъществява връзка с вътрешна шина?
     * USB, LPT, Com-port, **PCI**, **ISA**, Ethernet
190. Кой от посочените компоненти осъществява връзка с външна шина?
     * **USB**, **LPT**, **Com-port**, PCI, ISA, **Ethernet**
191. Какво означава CRC?
     * **Cyclic Redundancy Check**, Complete Register Content, Connection Regulating Chip, Create Read Clear
192. Какво е DMA?
     * **Direct Memory Access**, Digital Machine Accumulator, Digital Memory Access, Detector of Masked Attacks
193. Какво е SOF?
     * **Start of Frame**, Superior orbital fissure, Sequence of Files, Single Offset Flag
194. Коя шина има посока от uC-Host към Ethernet?
     * **TXD**, RXD
195. Коя шина има посока от Ethernet към uC-Host?
     * TXD, **RXD**
196. Колко проводна е USB3 шината?
     * **9**, 3, 6, 1
197. Колко проводна е USB2 шината?
     * **4**, 2, 8, 1
198. Дадена е мултипроцесорна компютърна система с споделена шина и без кеш. Една от всички четири инструкции прави обръщение към паметта, като по време на своето изпълнение тази инструкция заема шината. Ако шината е заета, то заявилият я процесор се поставя в опашка за изчакване. Колко по-бърза би била такава компютърна система с 64 процесора в сравнение с 1?
     * 2, **4**, 8, 16, 32
199. Дадена е мултипроцесорна система с n процесора използващи обща шина. Вероятността всеки един от процесорите да използва шината в даден цикъл е p. Какъв е шансът шината да е свободна (да има 0 заявки за ползване)
     * p/n, (p\*n)^2, **(1 - p)^n**, 1/(p\*n), (1 - p)\*n
200. За показаната топология определете нивото на отказоустойчивост (Fault tolerance) определена като броят връзки, които могат да бъдат загубени, без това да доведе до разпадане на мрежата на две.
     * 
     * **0**, 1, 2, 3, 4, 6
201. За показаната топология определете нивото на отказоустойчивост (Fault tolerance) определена като броят връзки, които могат да бъдат загубени, без това да доведе до разпадане на мрежата на две.
     * 
     * 0, 1, 2, 3, 4, **6**
202. Какво е предназначението на механизмите за защита на паметта?
     * Да предпазват паметта от претоварване, да ограничават достъпа до съхраняваната в системата информация, **да ограничават достъпа до определени сегменти или страници (“Най-вероятно е това” SMAIL)**, да не допускат модифициране на системния софтуер от приложения
203. Ако даден процес използва 8 страници, колко реда има в таблицата на страниците?
     * 2, **8**, 16, 23
204. Колко е максималния брой страници с размер страниците 4K при 32 битов адрес?
     * 5K, 20K, **1M (На преговора казва 20K, ама този въпрос го е имало май и за Тест 1 и там е казал 1M, а и в Драйва е 1М тей че нз СМАЙЛ)**, 64K
205. При размер на страниците 8 байта и 8 битов адрес, колко е максималния брой страници?
     * 16, **32**, 64, 512, 4K
206. При размер на страниците 8 байта, на кой ред в таблицата на страниците се съдържа адресът на страницата, която съдържа адрес 12 от паметта? (Редовете се броят от 1)
     * 1, **2**, 3, 5, 7
207. При размер на страниците 8 байта, на кой ред в таблицата на страниците се съдържа адресът на страницата, която съдържа адрес 18 от паметта?
     * 1, 2, **3**, 5, 7
208. Какво е предназначението на TLB?
     * Да преобразува виртуалните странични адреси във физически, **да минимизира загубата на бързодействие при търсене на физическото разположение на данните**, да намали производителността на хардуерната виртуализация
209. Какво съдържа TLB при странична организация на паметта?
     * Таблица с номера на физически страници, принадлежащи на даден процес, Таблица с номера на виртуални страници, принадлежащи на даден процес, **Моментна карта на съответствието на страниците от оперативната памет, принадлежащи на даден процес**, Таблица на страниците на диска
210. Коя от изброените характеристики не описва таблицата на страниците?
     * Таблица на страниците се пази някъде в реалната оперативна памет, **таблицата на страниците съдържа информация за активните процеси**, таблицата на страниците се състои от редове – всеки ред от таблицата съответства на номер на виртуална страница, таблицата на страниците е допълнена с бит за присъствие, показващ дали съответната страница се намира в основната памет
211. Кой от посочените методи не е метод за преобразуване на виртуални адреси?
     * **Метод на локалността (по време и пространство)**, метод на странична организация, метод на сегментна организация, метод на сегментно-странична организация
212. Колко инстанции на GDT (Global Descriptor Table) съществуват в IA-32 архитектурата?
     * **Една за всички процеси**, по една за всеки процес, една или няколко за всеки процес
213. Колко инстанции на LDT (Local Descriptor Table) съществуват в IA-32 архитектурата?
     * Една за всички процеси, **една за всеки процес**, една или няколко за всеки процес
214. За коя от изброените процесорни архитектури се отнасят следните особености: Дължината на инструкциите е стотици битове, В рамките на процесора функционират паралелно множество функционални устройства, Всички функционални устройства споделят огромен обем общ регистров файл
     * Потокова, Суперскаларна, **VLIW**, EPIC
215. Кои от изброените особености характеризират предимствата на VLIW архитектурите пред останалите?
     * **Имплементациите са машинно-зависими**, Нито един от изброените, **Заемането на функционалните устройства се планират от компилатора според позицията в инструкционните пакети**, **Компилаторът генерира изцяло планирани изпълним код, в който липсват хазарти**, **Зависимостите между инструкциите се определят от компилатора и се използват при планирането в съответствие с латентностите на функционалните устройства**

ОТВОРЕНИ ВЪПРОСИ

1. Какви класове инструкции знаем?
   * FPL инструкции (плаваща запетая), SSE, MMX - Matrix Math Extensions, SSE2, SMXI.
2. При Intel архитектурата, какво е характерно за регистрите с общо предназначение при 64-битов режим ?

* Регистри с общо предназначение (General - purpose registers) – съхраняват операнди.64 битовият режим позволява работа с 32 и 64 битови операнди, като по подразбиране се използват 32 битови операнди. При работа с 32 битови операнди се използват регистрите EAX, EBX, ECX, EDX, EDI, ESI, EBP, ESP, R8D -R15D.При работа с 64 битови операнди, пред обикновената (32 битова) инструкцията се поставя REX префикс и регистрите са съответно RAX, RBX, RCX, RDX, RDI, RSI, RBP, RSP, R8- R15 Използването на REX префикса се разглежда детайлно във въпрос.

1. Какво е характерно за адресното пространство (при някоя конкретна архитектура) ? //тоя въпрос не го разбирам ама май тия долу за различни разновидности на условието????
   * режим на работа
   * protected mode
   * сегментация  
     1. "Flat" моделът
   * При "flat" моделът на организация на паметта, програмистът вижда един масив от до 2^(32) байта (4 GB). Физическата памет може да съдържа до 4GB, но обикновено е много по-малка. Процесорът изобразява 4-те GB flat пространство върху физическото адресно пространство чрез транслации, които ще описваме във някоя тема по-нататък. Програмистът не се нуждае от детайлите по това изобразяване. Pointer-ът на това flat адресно пространство е 32-битово число, което може да варира от 0 до 2^(32) -1.  
     1. Сегментния модел
   * При сегментния модел на организация на паметта, адресното пространство се разглежда като логическо адресно пространство което то е с големина до 2^(46) байта (64 TB). Процесорът изобразява 64 TB-овото логическо адресно пространство на физическото адресно пространство (до 4GB) посредством механизми за транслация на адреси. Програмистът вижда логическото адресно пространство на архитектурата като колекция от подпространства. Всяко от тези линейни подпространства е наречено сегмент. Големината на един сегмент може да варира от един байт до 2^(32) байта (4 Gb).

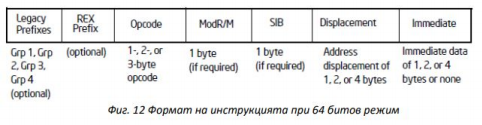
Пълен pointer в това адресно пространство се състои от две части:

# сегментен селектор, който е 16-битово поле, което идентифицира сегмента.

# Отместване(offset), който е 32-битово число, което адресира байт нивото в сегмента.

* + 1. "Real-Address Mode" моделът
  + При Real-Address Mode Model цялата памет се разделя на равни по размер сегменти. В CPU за всеки един от тези от тези сегменти има по един регистър, който сочи началото на съответния сегмент. Програмистът има достъп и може да променя съдържанието на кой да е от тези регистри. Промяната на съдържанието автоматично води до преместване на съответния сегмент в друга начална точка. Съществуват и други регистри с общо предназначение, които също са програмно достъпни. Те служат за задаване на отместването на адресната клетка от началото. За останалите три сегмента може да се използва кой да е от останалите регистри. Следователно адресът на клетката се задава винаги чрез двойка регистри – сегментен регистър и регистър-отместване.

1. Как се изчислява адресът?
   * Операндите в паметта се реферират чрез сегмент и отместването в него. При 32 битовия режим адресът на операнда се състои от 16 битов сегментен селектор и 32 битово отместване в сегмента. При 64 битовия режим, сегментния селектор също е 16 бита, но отместването може да бъде до 64^2. Отместването (Offset) може да бъде специфицирано като статична стойност (displacement) или да се изчисли с използване на следните компоненти : Displacement , Base, Index, Scale factor. Всеки от 8-те регистри с общо предназначение могат да се използват като за база. Регистри с общо предназначение, с изключение на стековия, могат да се използват като index при индексна адресация.
2. Опишете режимите на адресация при IA-32 и Intel-64?
   * Операндите в паметта се реферират чрез сегмент и отместването в него. При 32 битовия режим адресът на операнда се състои от 16 битов сегментен селектор и 32 битово отместване в сегмента. При 64 битовия режим, сегментния селектор също е 16 бита, но отместването може да бъде до 64^2.
3. Опишете сегемнтните регистри и тяхното предназначение
   * Сегментните регистри са 6 и остават 16 битови. Тяхното предназначение е да съхраняват сегментен селектор. Има различни сегменти. CS – (Code Segment) – съдържа базовия адрес на сегментът, в който се съхранява програмата. DS – (Data Segment) – съдържа адреса на сегмента, в който се съхраняват данните. SS – (Stack Segment) ES – (Extra Data Segment) – допълнителен сегмент за данни. FS - Допълнителен сегмент за данни. GS - Допълнителен сегмент за данни.
4. Опишете полетата на флаговия регистър
   * AF (Auxiliary Carry Flag) - междинен пренос между младшата и страшата тетрада. Смисълът от този флаг е някоя от следващите инструкции да провери този флаг. Използва се при двоично кодиране на десетични числа.
   * SF(Sign flag) – най-левия бит на резултата (знака).
   * OF (Overflow flag) - за препълване ако и двата са нула или и двата са единица нямаме препълване. Ако се различават то имаме такова.
   * DF (Direction flag) – обикновено при сравнение на низове (обръщане на посоката на прочитане на данните). Този флаг може да се постави от приложната програма (разрешен е за пипане от всяка програма).
   * TF (Trap Flag) – когато се сложи в единица се изпълнява една инструкция и се прави трап. Режим на трасировка.
   * IF (Interrupt Enable Flag) – за да се случи прекъсване този флаг трябва да е единица.
   * IOPL (I/O Privileg Level) – ниво на привилегия на входно изходните устройства. От това се определя дали да се случи едно входно-изходно прекъсване или не.
   * AC (Alignment Check) – когато се сложи в единица се задължава процесора да прави проверка за подравняването на границите на операндите. Ако е в нула процесора не се интересува къде са границите.
   * ID (ID flag) – идея всеки процесор да си има собствен идентификатор. Това не е този идентификатор, но при този флаг в единица се позволява да се ползва този идентификатор.
   * Status Flags - Статус флаговете на EFLAGS регистъра позволяват резултатите от една инструкция да влияят на следващи инструкции. Аритметичните инструкцииизползват OF, SF, ZF, AF, PF и CF. SCAS (Scan String), CMPS (Compare String) и LOOP инструкците използват ZF за да сигнализират, че са свършили работата си.
   * Control Flag - Контролният флаг DF от регистъра EFLAGS контролира инструкциите с низове.
5. Опишете основните типове данни
   * Байт, дума, двойна дума, четворна дума и двойна четворна дума са фундаменталните типове данни. Integer - Знакова двоична числова стойност; Unsigned Integer - Беззнакова двоична числова стойност; Floating point – Формат с плаваща запетая; Near pointer - 32-битов логически адрес; Far pointer - 48-битов логически адрес от два компонента: 16-битов сегментен селектор и 32-битово отместване; Bit field - Последователност от битове. Битово поле може да започне от всяка позиция и всеки байт и може да съдържа до 32 бита; BCD - кодиране на всяка десетична цифра на число като набор от четири бита. При това всеки байт на числото съдържа една или две десетични цифри в така наречения двоично-десетичен код (BCD — Binary-Coded Decimal); SIMD data types
6. Приложения на управляващи регистри
   * Управляващите регистри се използват от операционната система. Те са 32 битови във всички 32 битови режими. В 64 битов режим системните регистри са 64 битови. CR0 - Съдържа флагове контролиращи режимите на работа и състоянието на процесора. CR1 — Запазен, но не се използва. CR2 – Съдържа линейния адресът, който е причина за изключение от тип „page fault“. CR3 (PDBR) - Съдържа физическия адрес на таблицата със страници на текущия процес, както и флаговете PCD и PWT. CR4 - Съдържа група от флагове позволяващи използването на различни разширения.
7. Обяснете преобразуването на логическия адрес във физически при сегментно страничната организация на паметта
   * В режим на сегментация и странициране получаваме 32-битов линеен адрес. Първите 10 бита от него се умножават по ширината на полето на каталога (4 bytes) и се добавят към началния адрес на каталога. Ако старшият бит на реда в каталога е 1, значи че съществува каталог в паметта, който се сочи от реда в каталога. В такъв случай се взимат младшите 20 бита от реда в каталога, добавят им се 12 нули отдясно и се прибавя стойността на полето PN, умножена по широчината на реда в каталога. В реда в каталога (в младшите му20 бита) е записан адресът на търсената от нас страница, ако тя съществува. Последните 20 бита на реда в каталога съдържат линейния адрес на страницата, а физическият адрес на нейното начало се получава, като отново към тези 20 бита прибавим 12 нули отдясно и се прибави 12-битовият offset.
8. Какви полета съдържа локалната дискрипторна таблица?
   * Локалната дескрипторна таблица (Local Descriptor Table LDT) съдържа описание на сегменти, които са частни за дадена програма, докато глобалната съдържа глобални сегменти.
9. Обяснете структурата на задачата при IA 32 архитектура
   * Задачата се състои от пространство за изпълнение (execution space) и сегмент за състоянието на задачата (Task State Segment TSS). Пространството за изпълнение се състои от сегмент за кода, стека и няколко сегмента за данни. Най-общо казано, TSS съхранява в паметта информацията, необходима за изпълнението (execution space) на дадена задача.
10. Обяснете принципа на работа на кеш-паметта и протоколите за процесори
    * Кеш-паметта представлява структура, състояща се от множество от рамки (frames). Всяка рамка се състои от данни, етикет (tag) и състояние. Етикетът е характеристика на рамката, която е съществена при търсене в данните в кеша. Състоянието е два бита – бит valid, който показва дали данните в кеша са валидни и бит dirty, който показва дали в рамката са били писани данни. Търсенето и зареждането на рамки в кеша става по следния начин: проверява се дали входящият етикет съвпада с етикета на рамката. Ако не съвпадат, рамката не се намира (cache-miss) и в такъв случай блокът данни в рамката се изхвърля и се заменя с блок от по-долното ниво в йерархията, а след това блокът се подава на предното ниво в йерархията.
    * Протокол на кеша MESI: използва се както в едноядрени, така и в многоядрени процесори. При този протокол се проверяват четири дефинирани състояния на всяка линия на кеша – (Modified, Exclusive, Shared, Invalid).
11. Какво представлява и как работи предварителното извличане?
    * Предварителното извличане на данните в кеша (Cache prefetching) е техника, която увеличава производителността, като копира данните от бавна памет в по-бърза, още преди да са необходими. Най-често се използват потокови буфери (Stream buffers), които се основават на последователното изпълнение на инструкциите и извличане на данните.
12. Каква е ролята на завършващата част (Retirment) на вътрешната архитектура ?
    * Завършващата част се грижи за това, резултатът от изчисленията да съответства на реда на постъпване на инструкции и че състоянието на системата се обновява правилно. След оттеглянето (Retirement) на инструкциите (микрооперациите –μops) за запис, данните се записват в кеш паметта от първо ниво. Следва да се отбележи, че за един цикъл може да се оттеглят (Retire) до три микроинструкции.
13. Какво е характерно за защитения режим на процесора ?
    * Това е нормалният режим на работа. Той позволява използването на сегментиране, виртуална памет и други техники за защита. В него се влиза при зареждането на операционната система и изобщо е създаден за да позволи управлението на ресурсите от страна на операционната система.Например, когато даден процес има нужда от допълнително памет, трябва чрез системно извикване да се обърне към операционната система.
14. Обяснете механизма за обновяване на страница от диска.
    * Мениджърът на паметта използва информацията от таблиците за да намери неизползвана страница физическа памет (такава, към която скоро не е имало обръщения) и я записва в резервирана област на диска, наречена swap файл. После уведомява централния процесор за необходимостта да прочете изискваната страница от по-бавната памет (файл на диска или swap файл). Накрая MMU установява съответствието между виртуалната и физическата страница и обновява таблицата с тази информация.
15. Опишете последователността от стъпки, която се изпълнява при прекъсване в IA32 архитектурата
    * Прекъсванията и изключенията карат процесора да скочи (jump) в програма, която обработва прекъсването (interrupt handler),след като запази контекста (за да може прекъснатата програма да продължи).
16. Обяснете групите, на които се делят изключенията.
    * Изключенията се делят на faults, traps, и aborts в зависимост от начина по който се отчитат и дали инструкцията, която ги е причинила може да бъде рестартирана. За faults и traps се знае коя инструкция ги е причинила и програмата или процесът могат да бъдат рестартирани без загуба, докато при aborts това може да не е ясно и следва прекратява на програмата или процесът.
17. Опишете причините за възникване и стратегиите за преодоляване на "Мехури" в инструкционните конвейри.
    * Причиняват се от преодоляването на хазарти. Стратегия за преодоляване са да се ползват инструкции, които имат по-малко или никакви зависимости върху регистри, както и заместването на условните преходи с условни премествания (CMOV).
18. Обяснете използването на регистъра GDTR.
    * съдържа базовия адрес (32 бита в защитен и 64 бита в IA-32e режим) и размера (16 бита) на глобалната дескрипторна таблица GDT. Инструкциите LGDT и SGDT се използват съответно за зареждане и записване стойността на регистъра. Глобалната дескрипторна таблица, която е с 8 байтови елементи, описващи къде се намират сегментите се съхранява в оперативната памет.
19. Какво е предназначението на сегментните селектори?
    * Сегментните селектори са 16 битови идентификатори на сегмента.Старшите 13 бита се използват да определяне на отместването в глобалната или локалната дескрипторни таблици. Трите младши бита имат по-специфично предназначение. В зависимост от флага TI се адресира глобална или локална таблица. Полето RPL съдържа необходимото ниво на привилегия, като за да бъде позволен достъп, стойността на (DPL —Descriptor privilege level) трябва да е по-малка или равна от RPL.
20. Какво представлява сегментната преадресация? Посочете предимства и недостатъци.
    * Сегментната предадресация представлява разделянето на паметта на сегменти, т.е. на много адресни пространства. Предимствата са че това позволява защита на паметта, на всеки процес може да се дадат отделни сегменти и така той ще вижда свое собствено адресно пространство. Недостатък е че това увеличава сложността на достъпа до паметта, което забавя работата.

1. Какво представлява изпълняващата част (Execution Core)?
   * Изпълняващата част пренарежда микрооперациите по такъв начин, че тези чиито операнди са готови (и има налични изчислителни ресурси) да се изпълнят възможно по-скоро. Изпълняващата част може да обработи (issue) няколко микрооперации за един цикъл.
2. Обяснете приликите и разликите между прекъсване и изключение.
   * Изключенията са специален клас прекъсвания и ползват същите механизми. Докато в общия случай прекъсванията не зависят от програмата, която се изпълнява в момента, а от хардуера, то изключенията се причиняват от самата програма и в един смисъл са предвидими. Те се случват при грешки в самата програма – аритметични, неразрешен достъп до паметта и т.н.
3. Как се осъществява превключването между задачите при Intel архитектурите ?
   * При превключването към друга задача, състоянието на средата, в която се е изпълнявала текущата задача (състояние на задачата или още контекст на задачата) се запазва в TSS и изпълнението на задача се прекратява за някакъв период от време. След това информацията за предстоящата за изпълнение задача се зарежда в процесора и нейното изпълнение започва от инструкцията, сочена от вече зареденият регистър EIP на текущата задача).
4. Какво представляват и какви са предимствата на пакетираните типове данни?
   * Те са много примитивни типове, съхранявани на едно място – например една 128-битова „данна“ може да съдържа 2 64 битови числа, 4 32 битови и т.н. Предимството е, че процесорите поддържат КАК (Single Instruction Multiple Data) инструкции, с които се постига по-висока производителност.
5. Какво представлява виртуалната памет и как се поддържа при архитектурата Intel64?
   * Виртуалната памет представлява механизъм за „увеличаване“ на наличната памет, като се ползва и диска. Също така преодолява получаването на неизползваеми дупки. Това става с мапване на виртуални адреси, с които работи процесора, към физически. При Intel 64 се поддържа странициране (paging) с размер на страниците 4 KB, 1 MB, 2 MB и 4 МB, като превода на адреси се осъществява от хардуера.
6. Каква информация се съдържа в IDT (Interrupt Descriptor Table)?
   * Interrupt Descriptor Table (IDT) e таблицата на векторите на прекъсванията. Тя има 256 елемента. Векторите на прекъсванията не са дескриптори на сегменти, а са входове към такива.
7. По какво се различават изключенията от тип fault и aborts ?
   * За faults и traps се знае коя инструкция ги е причинила и програмата или процесът могат да бъдат рестартирани без загуба, докато при aborts това може да не е ясно и следва прекратява на програмата или процесът.
8. Каква информация се съдържа в Task Register и за какво се използва ?
   * Съдържа 16-битов сегментен селектор, както и допълнителна част, в която автоматично се зарежда съдържанието на съответния дескриптор. Това включва базов адрес (32 бита в защитен и 64 бита в IA-32e режим), размер на сегмента и атрибути на Task State Segment (TSS ) на текущия процес. Селекторът реферира TSS дескриптор в глобалната таблица GDT. Когато започне изпълнението на нов процес (Task) съдържанието на дескриптора се зарежда в скритата час и едва когато се смени процесът, то тогава се променя и сегментния регистър. По този начин, при следващо обръщение (в рамките на същата задача), не е необходимо да се достъпва глобалната таблица. Когато обаче се смени сегментния селектор е необходимо отново да се заредят данните от съответния дескриптор и да се запишат в скритата част (Segment Descriptor Registers).
9. Какво представлява и каква е функцията на челната част (Front End) ?
   * Челната част извлича и декодира инструкциите. Декодираните инструкции се транслират до микрооперации (μops). Основната задача на тази част е да осигурява непрекъснат поток от микрооперации на изпълняващата част.
10. При процесорна компютърна система със споделена шина, какво ще се случи, ако два от процесорите се опитат да достъпят глобалната памет по едно и също време ?
    * Това не е проблем, тъй като паметта може да бъде достъпвана само от 1 устойство по едно и също време, т.е. един от процесорите ще досъпи паметта първи.От там нататък имаме последователен достъп.
11. Опишете полетата обособяващи формата на инструкциите в режим IA32.

* 1) Instruction prefixes – Чрез префикса може да се променя поведението на инструкцията, като например да се сменя сегмента, размера на думата, да се управлява цикличното изпълнение и използването на шината   
  2) Opcode – Кодът на операцията се кодира с 1, 2 или 3 байта. Допълнително може да се използват и 3 бита от полето ModR/M. Използването на тези полета зависи в голяма степен от типа операцията.

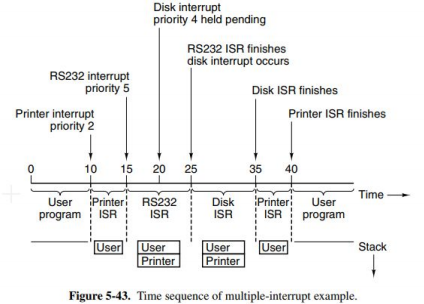
3) ModR/M байт – не задължителен и понякога може да съдържа част от opcode-а.

4) SIB – Scaled Index Base, не е задължителен

5) Displacement – Размерът на отместването се определя от полето Mod на ModR/M байта

6) Immediate - не е задължителен компонент и се използва като числова стойност от вариращи размери от байтове(byte, word, long).

1. Какъв е приоритета на прекъсванията RS232, DISK?

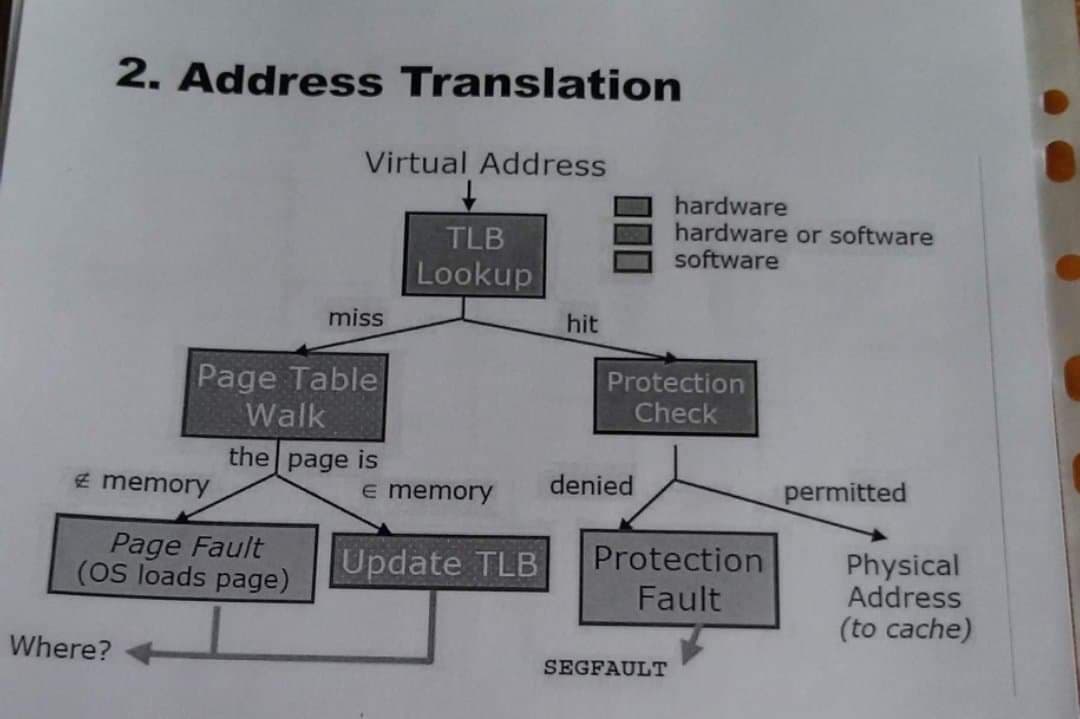


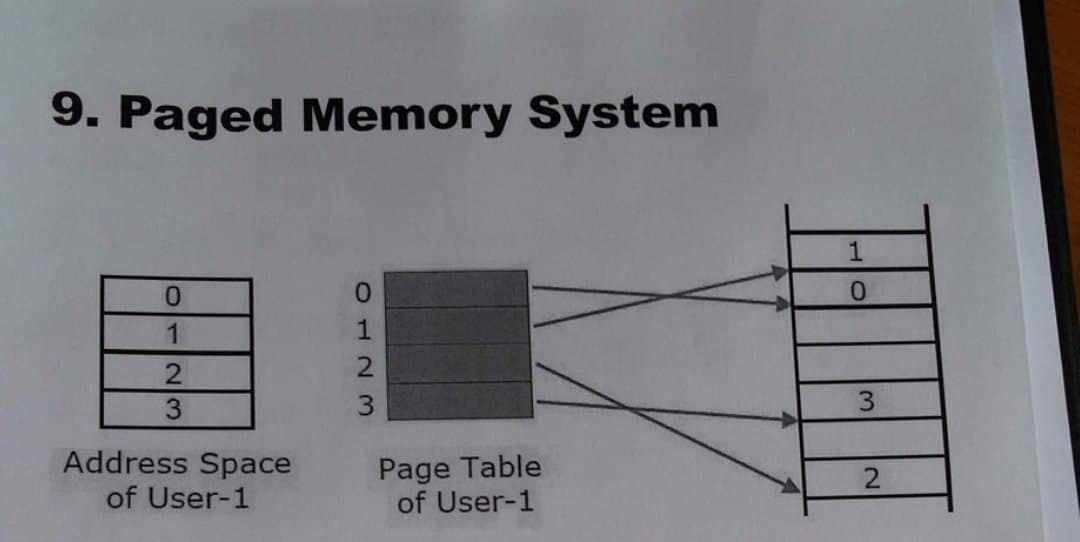
* RS232 се изпълнява преди DISK

Пояснение: Ако настъпи прекъсване с по-висок приоритет се влиза в него

1. Обяснете схемата и основните   
   принципи на действие.

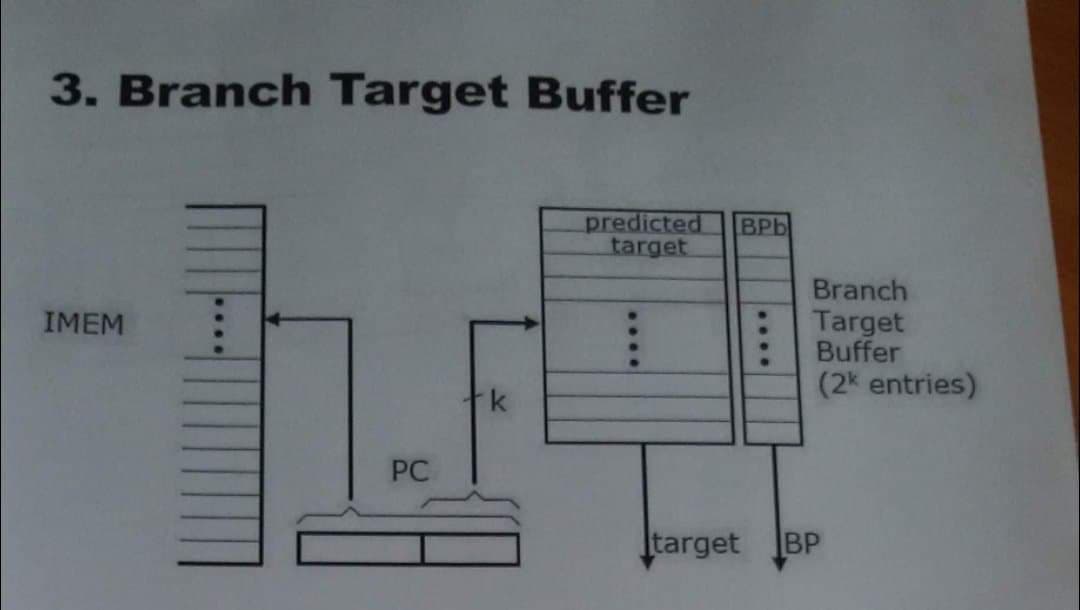
* FRAME# определя кога започва и кога заръвшва транзакцията. По време на адресната фаза се идектифицират типът и целта на транзакцията. През data phase(s) информацията се предава между двата компонента, свързани с магистралата.

1. Обяснете схемата и основните принципи на действие.  
   * TLB буферът (Transaction Lookaside Buffer) се използва за намаляване на времето за достъп до паметта. Той съхранява най-скорошните транслации от виртуална към физическа намет. При транслирането на виртуален адрес към физически се прави първо проверка в TLB буфера.
2. Обяснете схемата и основните принципи на действие.



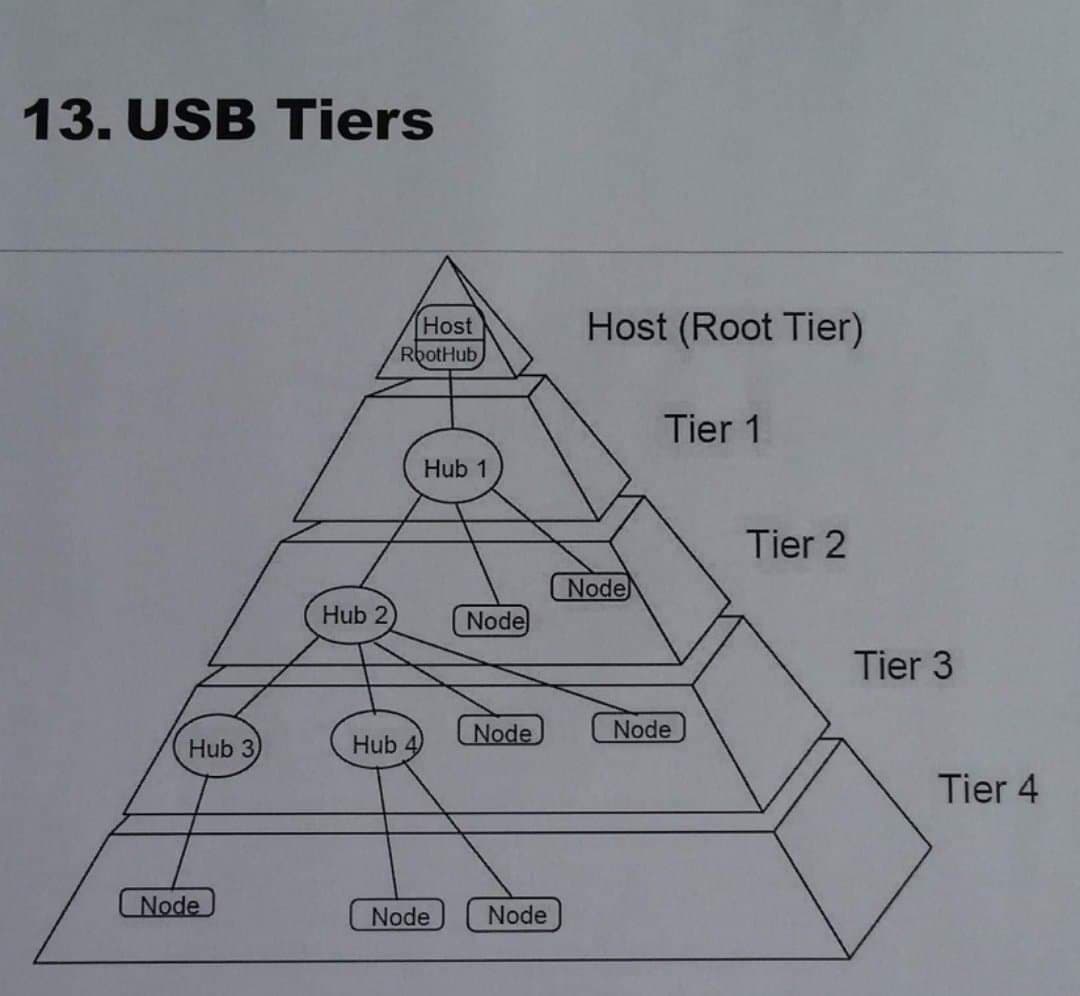
* Виртуалната памет има таблица на страниците, която се пази в оперативната памет. Всеки ред от таблицата съответства на номер на виртуална страница. При 32 битова архитектура, размерът е 32 бита за ред.

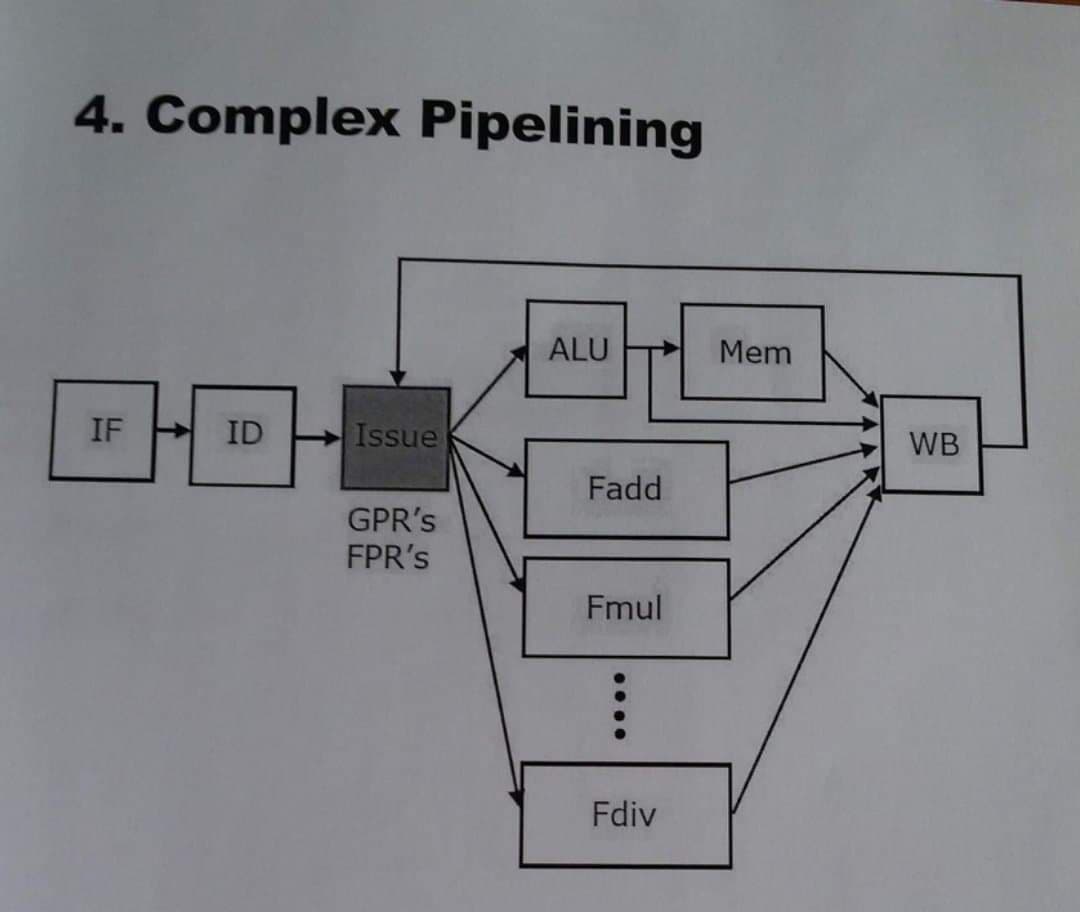
1. Обяснете схемата и основните принципи на действие.

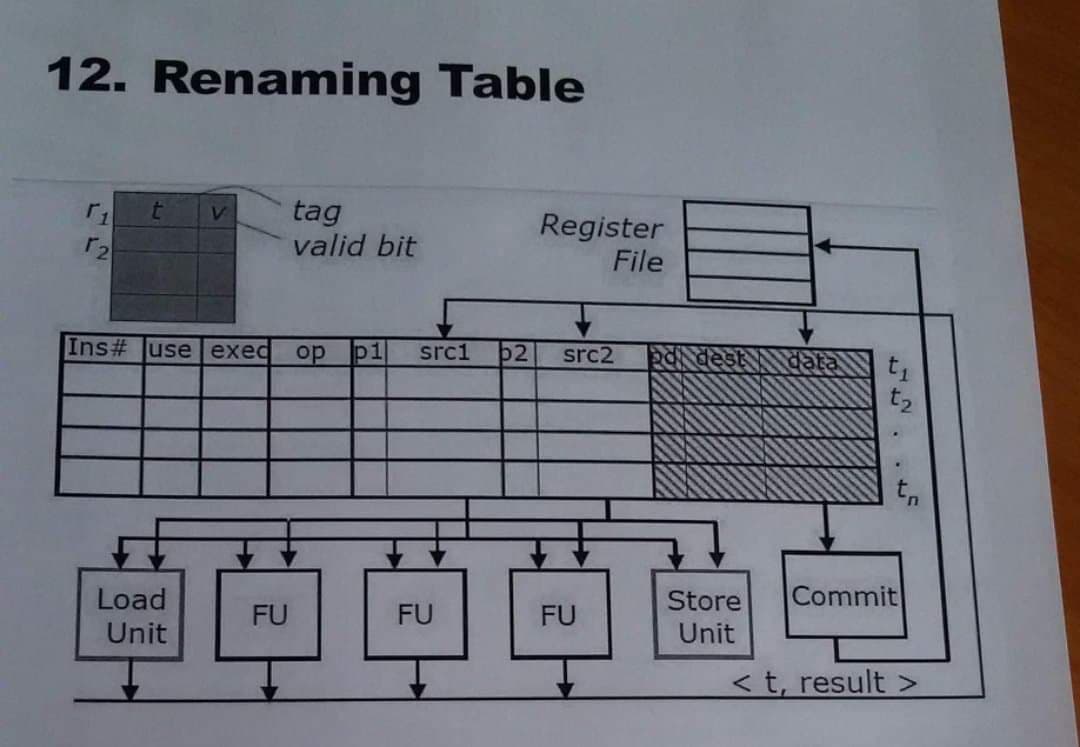


* BTB е таблица, индексирана от текущия програмен брояч, която съдържа branch target адреси. Някои branch target методи като jump, могат да бъдат изпълнени за 0 цикли. И Branch Target Buffers и Branch Prediction Buffers могат да се използват за повишаване на производителността.

1. Обяснете схемата и основните принципи на действие.

* Коренът на дървото се нарича централен хъб и играе ролята на контролер на шината, а по дървото могат да бъдат включвани до 127 хъбове или node-ове (крайни у-ва). Не се позволяват директни връзки между node-овете (P2P връзки). Данните се предават  
  импулсно.

1. Обяснете схемата и основните принципи на действие.
   * За постигане на едно и също време на изпълнение на инструкциите се използват отделно функционални блокове (instruction-level parallelism).

1. Обяснете схемата и основните принципи на действие.
   * Преименуването на регистрите е техника, която позволява да се намали ефектът на фалшивите зависимости, чрез използване на голям брой физически регистри, вместо ограничен брой архитектурни регистри. Например х86 съвместимите Intel P6 процесорни архитектури съдържат 40 физ. регистъра.