

Лекция 8. Көп процессорлы жүйелер

Мультипроцессор-екі немесе одан да көп орталық процессорлар ортақ жедел жадқа толық қол жеткізе алатын компьютерлік жүйе. Кез-келген орталық процессорда жұмыс істейтін бағдарлама әдеттегі виртуалды кеңістікті көреді (әдетте беттік ұйымы бар). Бұл жүйеге тән жалғыз ерекше қасиет-орталық процессор жад сөзіне қандай да бір мағынаны жазып, содан кейін сол сөзді санап, басқа мағынаны ала алады (өйткені басқа орталық процессор оны өзгертті). Тиісті ұйымдастырумен бұл қасиет процессораралық деректер алмасудың негізін құрайды: бір орталық процессор жадқа кейбір деректерді жазады, екіншісі оны жадтан оқиды.

Көп жағдайда мультипроцессорлық операциялық жүйелер әдеттегіден айтарлықтай ерекшеленбейді. Олар жүйелік қоңырауларды өңдейді, жадты басқарады, файлдық жүйені қамтамасыз етеді және енгізу-шығару құрылғыларын басқарады. Дегенмен, олардың бірегей ерекшеліктері бар бірқатар аймақтар бар. Бұл бағыттарға процесті синхрондау, ресурстарды басқару және жоспарлау кіреді. Мультипроцессорлық жабдықтың қысқаша шолуын қарастырамыз.

Мультипроцессорлық аппараттық құрал

Барлық мультипроцессорлардың әрбір орталық процессорға бүкіл жад кеңістігіне қол жеткізуге мүмкіндік беретін қасиеті болғанымен, кейбір мультипроцессорлардың тағы бір қасиеті бар: әрбір жад сөзін кез келген басқа жад сөзі сияқты жылдам оқуға болады. Мұндай машиналар *UMA* мультипроцессорлары деп аталады (*UniformMemoryAccess* – жадқа біркелкі қол жеткізу). Керісінше, *NUMA*-мультипроцессорлар (*NonuniformMemoryAccess* – жадқа гетерогенді қол жетімділік) бұл қасиетке ие емес. Әрі қарай, неге мұндай айырмашылық бар екені белгілі болады. Алдымен *UMA* мультипроцессорлары, содан кейін *NUMA* мультипроцессорлары қарастырылады.

UMA-Шина архитектурасы бар мультипроцессорлар

Қарапайым мультипроцессорлар (8.1-сурет, а) Жалпы шинаны қолдануға негізделген.

Екі немесе одан да көп орталық процессорлар және бір немесе одан да көп жад модульдері деректерді ортақ пайдалану үшін бірдей шинаны пайдаланады. Орталық процессор жад сөзін оқуы керек болғанда, ол алдымен шинаның жұмысын тексереді. Егер шина бос болмаса, орталық процессор оған қажетті сөздің мекен-жайын қояды, бірнеше басқару сигналдарын береді және жад шинаға қажетті сөзді ашқанша күтеді.

Егер шина бос емес болса, онда сөзді санау немесе оны жадқа жазу керек орталық процессор шинаның босатылуын күтеді. Дәл осы архитектураның проблемасы. Екі немесе үш орталық процессор болған кезде, шина даулары толығымен басқарылатын болады, бұл 32 немесе 64 процессор туралы айту мүмкін емес.

Жүйе шинаның өткізу қабілеттілігімен толығымен шектеледі және Орталық процессорлардың негізгі бөлігі көп жағдайда бос болады.

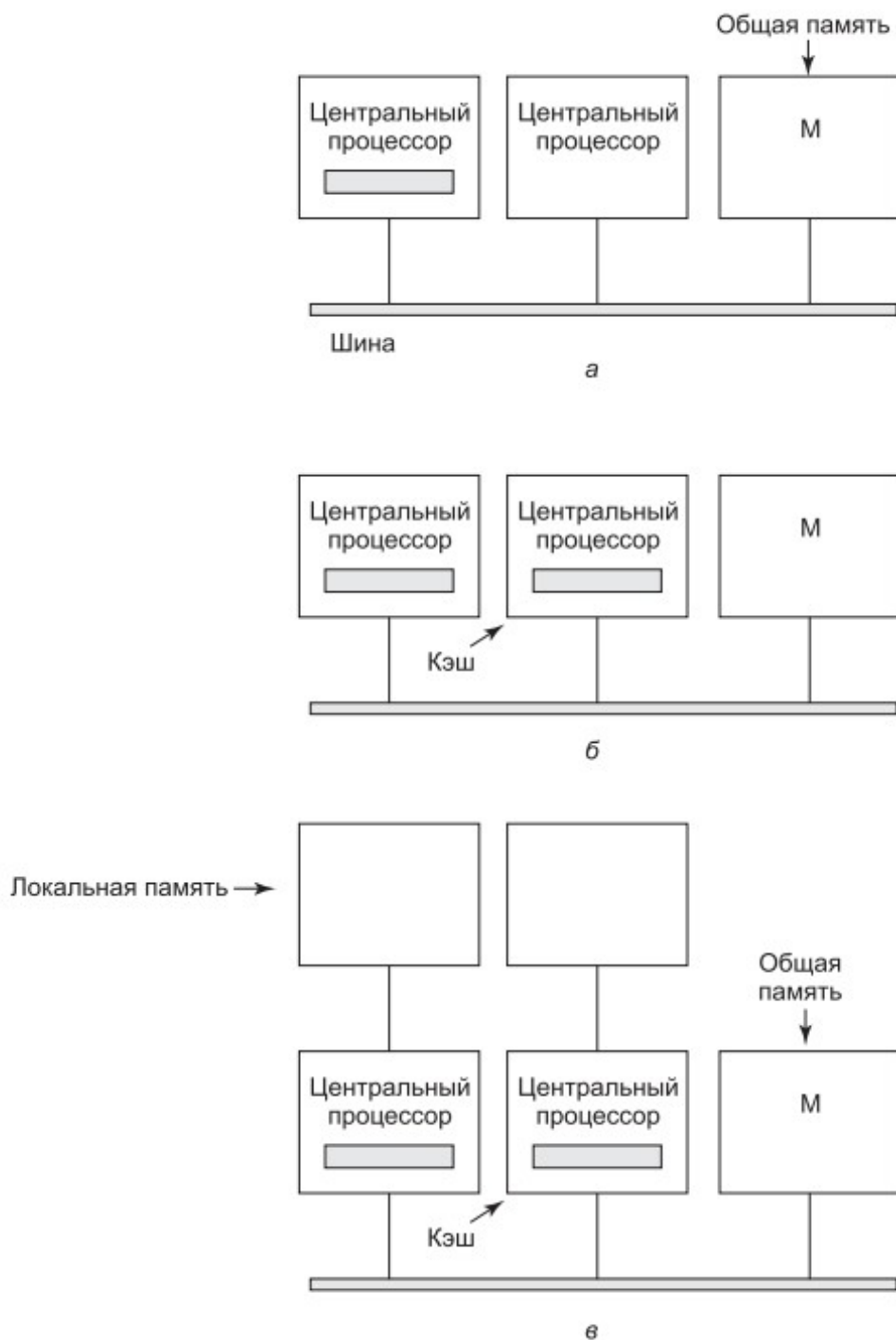
Бұл мәселенің шешімі көрсетілген 8.1-сурет, б және әрбір орталық процессорға кэшті қосудан тұрады. Бұл жад орталық процессор чіпінің ішінде орналасуы мүмкін, сол чіпке іргелес болуы мүмкін, жалпы тақтада болуы мүмкін немесе тізімделген үш опцияның тіркесімімен ұсынылуы мүмкін. Көптеген оқу операциялары енді жергілікті кэш арқылы қанағаттандырылуы мүмкін болғандықтан, шина арқылы берілетін деректер көлемі айтарлықтай азаяды және жүйе орталық процессорлардың көбірек санын қолдай алады.

Әдетте, кэштеу сөзбе-сөз емес, 32 немесе 64 байт блоктарда жүзеге асырылады. Сөзге жүгінген кезде, орталық процессордың кэшіне кэш жолы (*cacheline*) немесе кэш жолы деп аталатын бүкіл блок шығарылады.

Әрбір кэш блогы тек оқуға арналған деп белгіленеді (бұл жағдайда ол бір уақытта бірнеше кэште болуы мүмкін) немесе оқуға және жазуға арналған (бұл жағдайда ол басқа кэштерде болмауы мүмкін). Егер орталық процессор бір немесе бірнеше қашықтағы кэштердегі сөзді жазуға тырысса, Шина аппаратурасы жазбаны анықтайды және барлық басқа кэштерді жазу туралы хабардар ететін

шинаға сигнал береді. Егер басқа кэштерде жадтағы блок мазмұнына сәйкес келетін өзгертілмеген көшірмелер болса, олар бұл көшірмелерді жай ғана қабылдамай, жазуды жүзеге асыратын процессорға оны өзгертпес бұрын кэштегі блокты жадтан шығаруға мүмкіндік береді. Егер басқа кэштерде өзгертілген көшірме болса, олар жаңа жазбаны жасамас бұрын оны жадқа қайта жазуы керек немесе оны шина арқылы тікелей жазып жатқан процессорға жіберуі керек еді. Бұл ережелер жинағы кэш когеренттілігін қолдау хаттамасы (cache-coherenceprotocol) деп аталады және бұл көптеген хаттамалардың бірі ғана.

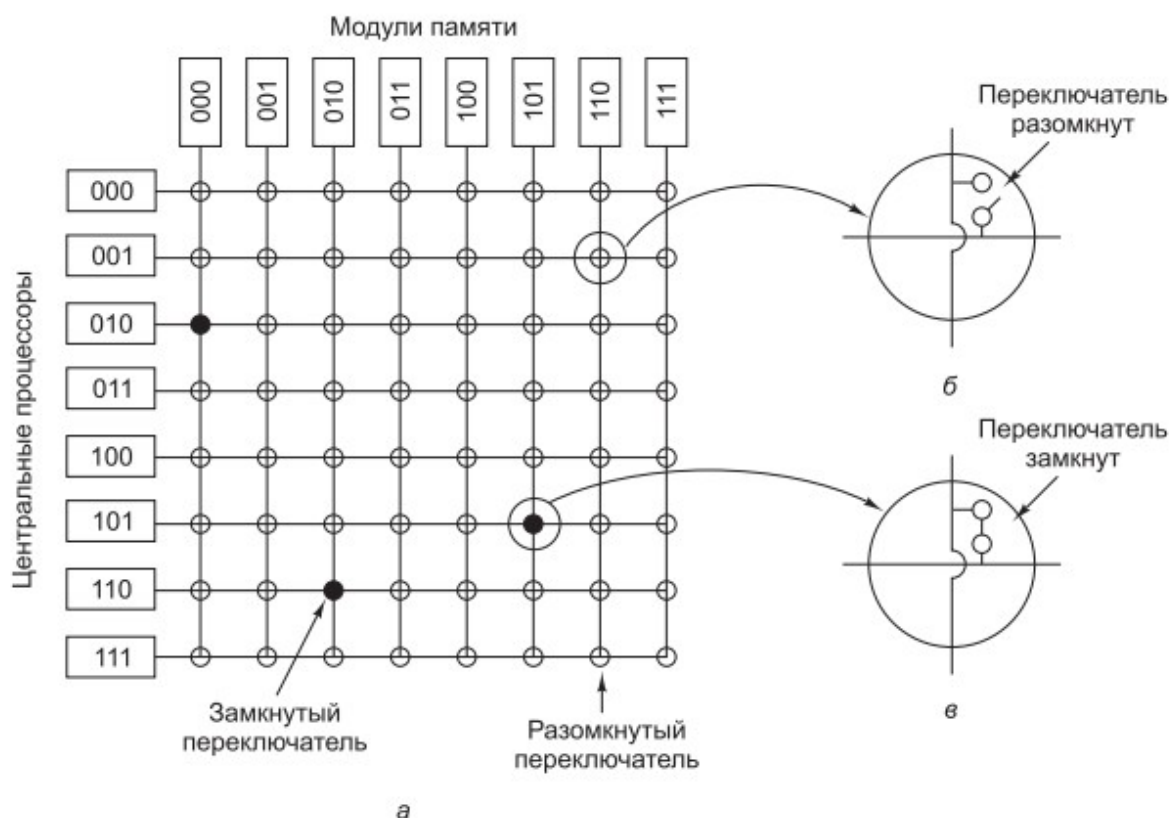
Мұнда әр процессордың кэші ғана емес, сонымен қатар жергілікті жеке жады бар, оған арнайы жеке шина арқылы қол жеткізуге болады. Бұл архитектураны оңтайлы пайдалану үшін компилятор бағдарламаның барлық мәтінін, барлық жолдарды, тұрақтыларды және тек оқуға арналған басқа деректерді, стектер мен жергілікті айнымалыларды жергілікті жад модульдеріне орналастыруы керек. Содан кейін ортақ жад тек өзгертілетін жалпы айнымалылар үшін қолданылады. Көп жағдайда мұндай ұқыпты орналастыру шина арқылы берілетін деректер көлемінің айтарлықтай төмендеуіне әкеледі, бірақ компилятордың белсенді көмегін қажет етеді.



Сурет 8.1. Ортақ шинасы бар мультипроцессорлар:
 а-кэш жады жоқ; б – кэш жады бар; в-кэш жады және жеке жады бар процессор
Ута-координаталық қосқыштарды қолданатын мультипроцессорлар

Ең сәтті кэштеу кезінде де бір шинаны пайдалану ұта мультипроцессорларын тек 16 немесе 32 орталық процессорға дейін азайтады. Бұл кедергіні жеңу үшін деректер алмасу желісінің басқа түрі қажет. N орталық процессорларын k жад модульдеріне қосудың қарапайым схемасы-координаталық қосқыш (crossbar switch) (8.2-сурет). Координаталық коммутатор ондаған жылдар бойы кіріс желілерінің тобын Шығыс желілерінің жиынтығымен еркін байланыстыру үшін телефон байланысын ауыстыру үшін қолданылған.

Көлденең (кіріс) және тік (шығыс) сызықтардың әр қиылысында коммутация элементі, электр сигналын жіберуі немесе өткізбеуі мүмкін шағын қосқыш, яғни көлденең және тік сызықтар қосылуы немесе қосылмауы керек немесе қосылмауы керек. 8.3-суретте және "орталық процессор – жад модулі" жұптарына қосылуға мүмкіндік беретін бір уақытта қосылған үш элемент көрсетілген": (001, 000), (101, 101) және (110, 010). Сіз көптеген басқа комбинацияларды жасай аласыз. Шын мәнінде, комбинациялардың саны шахмат тақтасында сегіз қырлы орналастырудың әртүрлі тәсілдерінің санына тең, онда бұл фигуралардың ешқайсысы екіншісінің соққысына ұшырамас еді.



Сурет 8.2. Координаталық коммутатор: а-бір уақытта қосылған үш элемент; б-ашық коммутация элементі; в-жабық коммутация элементі

Координаталық коммутатордың бір керемет қасиеті бар: ол бұғатталмайтын желі (nonblockingnetwork), яғни ешқандай орталық процессор қандай да бір коммутация элементінің немесе желінің бос болуына байланысты қажетті қосылымнан ешқашан бас тартпайтын (қажет жад модулінің өзі бос деп есептесек). Бірақ бұл керемет қасиет барлық ішкі қосылыстарда бола бермейді. Сонымен қатар, ұзақ мерзімді жоспарлаудың қажеті жоқ. Жеті ерікті қосылым орнатылған болса да, қалған орталық процессорды қалған жад модуліне қосу мүмкіндігі әрқашан болады.

Әрине, екі орталық процессор бір уақытта бір жад модуліне қол жеткізуді қажет еткенде пайда болатын жадқа қосылу бәсекелестігі жоққа шығарылмайды. Дегенмен, 8.2-суретте көрсетілген модельмен салыстырғанда, жадты N модульдерге бөлу арқылы бәсекелестік n есе азаяды.

Негізгі әдебиеттер: [1,2,3].

Қосымша әдебиеттер: [8,11].

Бақылау сұрақтары:

1. Мультипроцессор-бұл...
2. UMA мультипроцессоры қандай машиналар деп аталады?
3. Кэш когеренттілігін қолдау протоколы деп не аталады? Мәнін түсіндіріңіз.
4. Координаталық қосқыштың жұмыс принципін сипаттаңыз.
5. Координаталық қосқыштың қандай қасиеттері бар?