(19) C2 (11) 55445 (13) UA

(98) вул. Пушкінська, 9, кв. 11, м. Київ-34, 01034
(85) 1999-10-11
(74) Пахаренко Антоніна Павлівна, (UA)
(45) [2003-04-15]
(43) [2000-08-15]
(24) 2003-04-15
(22) 1998-02-05
(12) null
(21) 99095020
(46) 2003-04-15

(86) 1998-02-05 PCT/DE98/00319
(30) 197 09 975.0 1997-03-11 DE
(54) МІКРОКОМП'ЮТЕР З ДИСПЕТЧЕРОМ ПАМ'ЯТІ МИКРОКОМПЬЮТЕР С УСТРОЙСТВОМ УПРАВЛЕНИЯ ПАМЯТЬЮ MICROCOM PUTER WITH A MEMORY MANAGEMENT UNIT
(56) US № 5452431, М. кл. G06F 12/14, G06F 12/16, 1995. 2 US № 4087856, М. кл. G06F 13/00, 1978. 2 US № 5325496, М. кл. G06F 1 1/00, G06F 12/00, G06F 12/14, 1994. 2 DE № 3709205, М. кл. G06F 12/14, 1988. 2
(71)
(72) DE Зедлак Хольгер DE Зедлак Хольгер DE Зедлак Хольгер DE Брюкльмайр Франц-Йозеф DE Брюкльмайр Франц-Йоз еф DE Брюкльмайр Франц-Йозеф
(73) DE СІМЕНС АКЦІЄНГЕЗЕЛЬШАФТ DE СИМЕНС АКЦИЕНГЕЗЕЛЬШАФТ DE SIEMENS AKTSIENGEZELSHAFT

В микрокомпьютере, предназначенном для выполнения нескольких программ пользователя, устройство управления памятью обеспечивает то, что ни одна из программ пользователя не имеет доступа к другим программам. Для того чтобы обеспечить возможность использования общих библиотечных программ и одновременно исключить неконтролируемый доступ к этим программам, предусмотрена зона хранения векторного массива, в которую введены начальные адреса библиотечных программ, представленные в виде адресов назначения для команд перехода (векторы 1050, 3000). Для вызова библиотечной программы необходимо ввести номер вектора 0 ... n, по которому устройство управления памятью определяет соответствующий адрес в зоне хранения векторного массива.

В мікрокомп'ютері з диспетчером пам'яті (MMU), в якому має працювати велика кількість програм користувача, за допомогою MMU гарантується, що через жодну програму користувача неможливо здійснити доступ до іншої програми. Щоб все-таки можна було користуватися спільними бібліотечними програмами і в той же час запобігати безконтрольному входженню до них, передбачається векторна область пам'яті, в яку вводять початкові адреси бібліотечних програм, що використовуються з метою переходу (вектори: 1050, 3000). Виклик бібліотечної програми здійснюється шляхом подання векторного номера (0...n), за допомогою якого MMU отримує відповідну адресу в векторній області пам'яті.

In a microcomputer in which a plurality of user programmes are to be run, an MMU ensures that none of the user programmes can access the other programmes. To make possible the use of common library programmes, however, and at the same time to avoid an uncontrolled entry into said programmes, a vector storage region is provided in which the start addresses of the library programmes are entered as branch destinations (vectors: 1050, 3000). To call a library programme, the vector number (0...n) is input, from which the MMU determines the corresponding address in the vector storage region.

1. Мікрокомп'ютер з центральним процесором (CPU), який через диспетчер пам'яті (MMU) з'єднаний з адресною шиною (BUS), до якої підключений принаймні один запам'ятовуючий пристрій (ROM, EEPROM) для програм, що має принаймні одну область пам'яті для програм користувача (A, B),
причому кожній програмі користувача (A або B) в диспетчері пам'яті (MMU) підпорядковано один сегментний дескриптор, в якому зберігаються принаймні початкова адреса (ANFA або ANFB), довжина (LA або LB) та право на доступ (ZRA або ZRB) програми користувача (A або B), та
з принаймні однією іншою областю пам'яті для бібліотечних програм (WRITE, ERASE) та однією векторною областю пам'яті,
причому в диспетчері пам'яті (MMU) сегментний дескриптор описує підпорядкованість векторної області пам'яті та області пам'яті бібліотечної програми,
причому в векторній області пам'яті зберігається принаймні векторний номер (0...n) та підпорядкований йому вектор (1050, 3000),
причому виклик бібліотечної програми (WRITE, ERASE) через програму користувача (A, B) мусить містити в собі принаймні назву сегментного дескриптора MMU, а також векторний номер (0...n), що через диспетчер пам'яті (MMU) підпорядковується вектору, через який здійснюється перехід до викликаної бібліотечної програми (WRITE, ERASE).
2. Мікрокомп'ютер за п. 1, який **відрізняється** тим, що узгодження векторної області пам'яті та області пам'яті бібліотечної програми здійснюється через подання початкової адреси та довжини векторної області пам'яті в сегментний дескриптор MMU, підпорядкований області пам'яті бібліотечних програм.
3. Мікрокомп'ютер за п. 1, який **відрізняється** тим, що узгодження векторної області пам'яті та області пам'яті бібліотечної програми здійснюється через подання початкової адреси та довжини області пам'яті в сегментний дескриптор MMU, підпорядкований векторній області пам'яті.
4. Мікрокомп'ютер за п. 1, який **відрізняється** тим, що узгодження векторної області пам'яті та області пам'яті бібліотечної програми здійснюється шляхом об'єднання обох областей в єдину область пам'яті, яка описується поданням початкової адреси та двох даних довжини в підпорядкованому сегментному дескрипторі MMU.
5. Мікрокомп'ютер за будь-яким з пунктів 1-4, який **відрізняється** тим, що вектор утворений за допомогою адреси переходу.
6. Мікрокомп'ютер за будь-яким з пунктів 1-4, який **відрізняється** тим, що вектор утворений за допомогою адреси команди переходу, яка веде до бібліотечної програми.
7. Мікрокомп'ютер за будь-яким з пунктів 1-6, який **відрізняється** тим, що векторний номер (0...n) отримується з відносної позиції вектора (1050, 3000) в векторній області пам'яті.
8. Мікрокомп'ютер за будь-яким з пунктів 1-6, який **відрізняється** тим, що векторний номер (0...n) складається з декількох байтів, а актуальний вектор отримується шляхом порівняння векторного номера (0...n), що знаходиться в векторній області пам'яті, з векторним номером, що міститься у виклику.

В мікрокомп'ютері працююча на даний момент програма здійснює контроль над комп'ютером або над розміщеними в ньому чи приєднаними до нього запам'ятовуючими пристроями (ЗП) або іншими периферійними приборами. Це поміж іншим означає, що завжди спрацьовує адреса ЗП, яка міститься в програмній команді, незалежно від того, чи повинна область пам'яті, яка містить в собі цю адресу, знаходитись в розпорядженні програми, чи ні.

Оскільки в багатьох випадках це не так - адже таким чином можна було б отримувати інформацію з областей пам'яті з суто секретним вмістом ЗП -, то вживаються запобіжні заходи.

Можливість для таких запобіжних заходів надається застосуванням диспетчера пам'яті (Memory Managment Unit), який надалі позначатиметься як MMU, який, наприклад, має місце в IAPX286 фірми INTEL . Здебільшого він застосовується тоді, коли повинні працювати не лише програми виготовлення (чіпа), але й програми користувача, які можуть бути введені в дію неправильно. MMU розміщують між центральним блоком обробки, який надалі позначатиметься як CPU, комп'ютера та шиною, котра з'єднує його з іншими блоками, такими як ЗП.

Кожне користування отримує введення в MMU, причому фіксується, в якому ЗП знаходиться користування, з якої адреси розпочинається, яку має довжину, та які існують права на доступ. Ці дані користувач мусить вносити в ЗП свого комп'ютера при записі свого користування або ж своєї програми. Лише тоді програма користувача має права,на доступ до областей пам'яті, які знаходяться в рамках, визначених попередньо заданою початковою адресою та довжиною. Отже, введення в MMU описує властивість програми, збереженої в сегменті запам'ятовування даних. Тому область, в якій знаходиться це внесення в MMU, називають сегментним дескриптором.

Кожний виклик адреси через програму перевіряється в MMU, і тільки якщо адреса знаходиться в дозволеній області, виклик приймається, в іншому ж випадку відбувається вихід з циклу програми або сигнал збою.

В тому випадку, коли в ЗП знаходяться програми різних користувачів, для кожного з них створена гарантія, що інші користувачі не зможуть вивчати або навіть змінити його програму, тому що кожна програма користувача може діяти лише в рамках області, визначеної при записі програми самим користувачем.

Документ US 5 452 431 описує логічну мікросхему з CPU, яка через адресну шину підключається до програмного ЗП, що має багато областей для програм користувача. Області пам'яті для програми користувача підпорядкована область пам'яті, в якій в вигляді таблиці відкладаються початкова та кінцева адреси та опорний код програми користувача. В ході роботи програми користувача початкова та кінцева адреса області пам'яті цієї програми порівнюються з адресами актуального виклику, причому викликана адреса, що знаходиться поза областю, визначеною початковою та кінцевою адресою, призводить до виходу програми з циклу. Однак цей запобіжний механізм не виходить за межі захисних можливостей, створених за допомогою MMU.

Документ DE 3 709 205 A1 разом з процитованим в ньому DE 3 533 787 A1 описує запобіжну схему для захисту даних, що знаходяться в області пам'яті. Області пам'яті підпорядковано дескриптор, до якого занесені властивості цієї області. Крім того, він містить преамбулу, до якої занесено, чи про захищені дані йдеться. Крім того, преамбула містить вектор, який при адресуванні цієї області пам'яті відкладається в проміжній пам'яті. Після позитивного результату перевірки права на доступ до захищених даних вектор переноситься до обчислювально-вирішального блоку, який керує пам'яттю і потім викликає програму для обробки захищених даних, місце пам'яті якої вказане вектором. Таким чином гарантується, що захищені дані не можуть оброблятися якимось іншим шляхом, а лише за допомогою цієї визначеної програми.

Зазвичай програми користувача мають ще й підпрограми. При цьому часто буває, що різні користувачі мають потребу в однакових підпрограмах, і тому завдяки описаним вище запобіжним заходам ці підпрограми мають місце в багатьох екземплярах. Для цього потрібен занадто великий обсяг пам'яті.

Тому було б бажано і доцільно передбачити в області пам'яті мікрокомп'ютера бібліотеки підпрограм, до яких могли б мати доступ різні програми користувачів, за певних обставин з застосуванням спеціальних запобіжних заходів, таких, наприклад, як перевірка персонального ідентифікаційного номера.

Але відтак знову виникають описані вище проблеми, які полягають в тому, що якийсь користувач з недобрими намірами може проникнути в будь-яку бібліотечну програму, обминувши рутинну перевірку.

Задача даного винаходу полягає в тому, щоб створити(мікропроцесор, який дозволяв би доступ до бібліотечних програм через програму користувача, але при цьому був захищений від маніпуляцій.

Задача вирішується за допомогою мікрокомп'ютера за пунктом 1 формули винаходу. Інші наділені перевагами варіанти рішення приведені в залежних пунктах формули винаходу.

В мікрокомп'ютері згідно з винаходом безпосередній перехід до бібліотечної програми є неможливим. Замість цього в CALL-команду окрім позначення сегментного дескриптора MMU, який описує бібліотечну програму, вводять векторний номер Позначенням сегментного дескриптора MMU може бути , наприклад, номер або ім'я.

За допомогою MMU перевіряють, чи заданий векторний номер взагалі є наявним та чи належить він до викликаної бібліотечної програми. При позитивному результаті перевірки дозволяється доступ до векторної області пам'яті, початкову адресу та довжину якої містить в собі сегментний дескриптор MMU. Лише в тій векторній області пам'яті, де знаходиться векторний номер, знаходиться і адреса переходу або адреса команди для переходу - тобто вектор - на початкову адресу бібліотечної програми. Таким чином ефективно виключається ситуація, коли користувач може безпосередньо переходити до бібліотечної програми, обминаючи рутинні запобіжні заходи.

В подальшому винахід пояснюється більш детально на основі прикладів виконання за допомогою фігур. Вони зображують: фігура 1 - схематично блок-схему програми мікрокомп'ютера, фігура 2 - схематично розподіл програм користувача серед вмістів сегментних дескрипторів MMU фігура 3 - схематично тип та спосіб виклику бібліотечної програми.

Фігура 1 показує в дуже схематичному вигляді складові частини мікрокомп'ютера. Центральний блок

обробки CPU через адресну шину з'єднується з диспетчером пам'яті MMU. З свого боку MMU з'єднаний з внутрішньою адресною шиною мікрокомп'ютера, до якої підключені блоки пам'яті ROM, RAM і EEPROM, а також блок вводу/виводу I/O. Можуть бути наявними інші звичні для мікрокомп'ютера блоки, які тут не зображені, оскільки не мають відношення до винаходу. Відмовилися також і від зображення шини даних і контролю. В будь-якому випадку мікрокомп'ютер згідно з винаходом може включати в себе всі необхідні для свого функціонування та відомі з рівня техніки складові частини.

Блок CPU передає логічні адреси до MMU, тоді як MMU отримує з них фізичні адреси і передає їх до пам'яті. Для цього MMU, як зображено на лівій частині фігури 2, обладнано комірками пам'яті для сегментних дескрипторів, куди введені підпорядковані програмі користувача A, B початкова адреса, довжина та право на доступ. Крім того, MMU може мати не зображений суматор, призначений для того, щоб можна було з логічної адреси шляхом додавання початкової адреси програми користувача вивести фізичну адресу. Для прикладу зображені сегменти для двох програм користувача A і B, причому програма A починається з адреси 50.000 і має довжину в 3.500 адрес, в той час як програма B починається з адреси 120.000 і має довжину в 5.000 адрес.

В ході програми A в CPU згідно з довжиною програми викликаються адреси від O до 3.499, Ці логічні адреси поступають до MMU, який додає до них початкове значення 50.000 і отриману таким чином фізичну адресу передає на внутрішню адресну шину. MMU заздалегідь перевіряє, чи знаходиться логічна адреса в області адрес, що відповідає збереженій в сегментному дескрипторі MMU довжині. Сегментний дескриптор MMU може бути виконаний, наприклад, в формі запам'ятовуючого регістра. Для такої перевірки в MMU передбачені порівнювальні блоки, до яких з одного боку поступає актуальна адреса, а з іншого боку граничні адреси актуальної програми. В випадку, коли через програму викликається більш висока або більш низька адреса, то відбувається вихід з циклу або має місце сигнал збою або щось подібне.

В іншу область сегментного дескриптора MMU введені права на доступ, завдяки чому може бути встановлено, який доступ може бути здійснений до певних адресних областей - тільки для читання чи для читання і запису.

Для попереднього викладу не має значення, в якому ЗП знаходяться програми користувача - в RAM, в EEPROM чи в якомусь іншому, і до яких комірок пам'яті можна дістати доступ через одну програму.

Фігура 3 зображує вдосконалений згідно з винаходом відомий мікрокомп'ютер. Тут поруч з іншим передбачено область пам'яті для підпрограм, що є доступними для всіх користувачів, тобто область пам'яті для бібліотечних програм. Для цього може бути використаний будь-який ЗП

Фігура 3 як приклад зображує в ЗП бібліотечних програм на адресах 1.050 та 3.000 програму WRITE та ERASE.

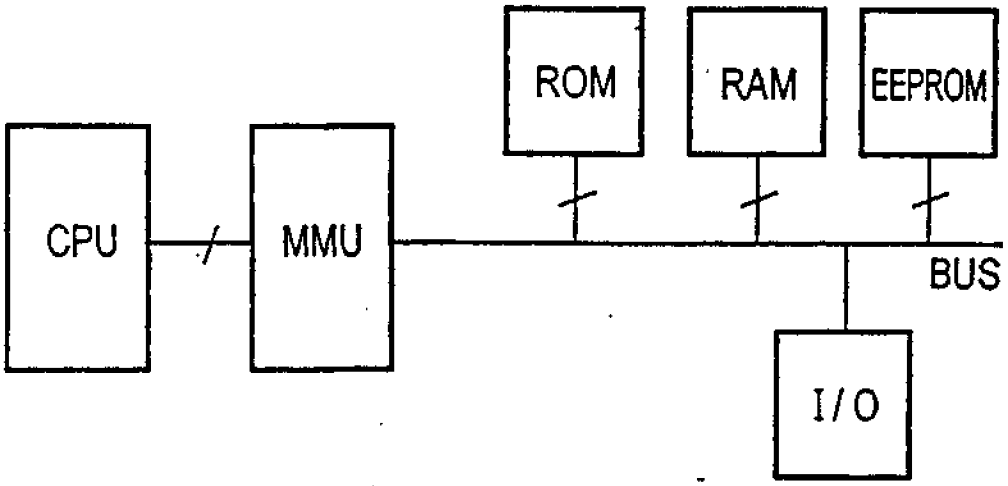
Згідно з винаходом тепер програма користувача не може безпосередньо перейти до цих адрес, інакше можливим став би і необумовлений вхід в ці програми - шляхом обминання запобіжних заходів. Замість цього передбачено векторну область пам'яті, в яку вводяться векторні номери та підпорядковані їм початкові адреси бібліотечної програми як мета переходу (вектори) 1050, 3000. Альтернативно може бути збережена в пам'яті також адреса команди переходу, який веде до підпрограми. Крім того, може бути введена назва підпрограми в вигляді характеристики, як це зображено в фігурі 3. Та це необов'язково.

Користувач не може дізнатися про фактичну, фізичну адресу бібліотечної програми. Окрім захисного аспекту це має ще й ту перевагу, що таку програму при необхідності можна усунути з робочої системи, не призводячи до змін в програмі користувача. Тут потрібно лише змінити мету переходу в векторній області пам'яті.

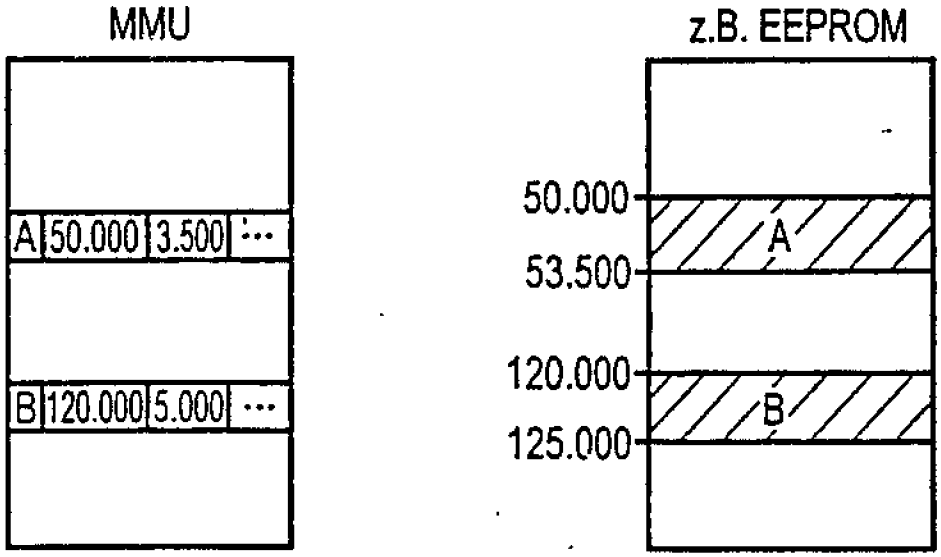
Кожна область пам'яті бібліотечної програми може бути введена в MMU як будь-яка інша програма. Згідно з винаходом кожній області пам'яті бібліотечної програми підпорядковано векторну область пам'яті, де вектори вводяться до бібліотечних програм, що знаходяться в області пам'яті бібліотечної програми. Введення відбувається шляхом подання початкової адреси та довжини векторної області пам'яті.

Альтернативно векторна область пам'яті може бути введена також в сегментний дескриптор MMU, причому в цьому випадку в сегментний дескриптор вводять початкову адресу та довжину області пам'яті бібліотечної програми. Крім того, можна об'єднати векторну область пам'яті та область пам'яті бібліотечної програми і ввести в сегментний дескриптор одну початкову адресу та дві довжини.

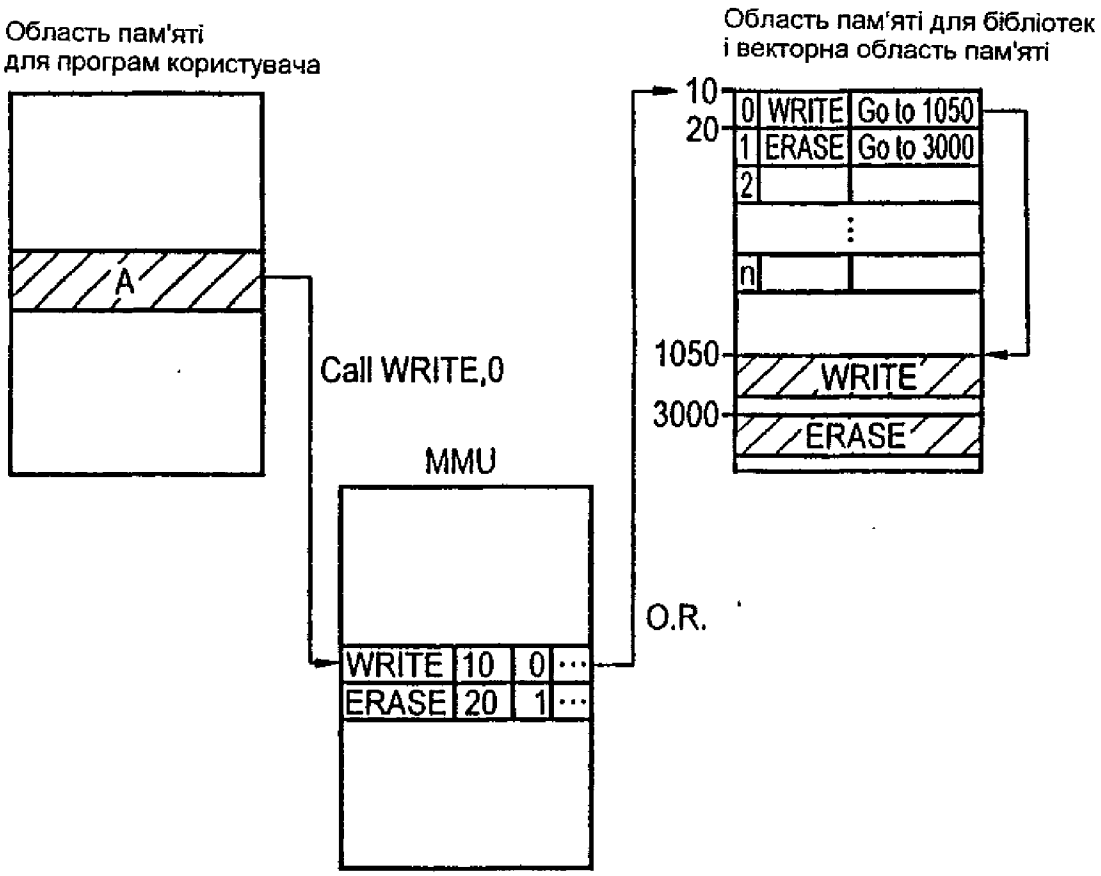
Виклик бібліотечної програми через програму користувача здійснюється шляхом введення назви сегментного дескриптора MMU, такої як ім'я бібліотечної програми або число та номер вектора. Потім MMU перевіряє, чи векторний номер взагалі існує в векторній області пам'яті, і чи співпадає викликана назва програми з введенням, підпорядкованим векторному номеру. Тільки при позитивному результаті перевірки відбувається вибір відповідної адреси в векторній області пам'яті, а вже після цього здійснюється перехід до бібліотечної програми.


ROM
RAM
EEPROM
CPU
MMU
BUS
I/O

Фіг.1


MMU
A
50.000
3.500
B
120.000
5.000
z.B. EEPROM
50.000
53.500
120.000
125.000
A
B

Фіг.2


Область пам'яті
для програм користувача
A
Call WRITE,0
MMU
WRITE 10 0 ···
ERASE 20 1 ···
O.R.
Область пам'яті для бібліотек
і векторна область пам'яті
10
20
0 WRITE Go to 1050
1 ERASE Go to 3000
2
n
1050
WRITE
3000
ERASE

Фіг.3