

Даний винахід відноситься до забезпечення інтерфейсу між програмними прикладними програмами і фізичними (периферійними) пристроями, зокрема, але не виключно, застосовно до приймачів/декодерів для систем цифрового телебачення.

Поява систем передачі цифрових даних уможливила використання таких систем для інших цілей. Однією з них є забезпечення інтерактивної роботи з кінцевим користувачем. Термін "система передачі цифрових даних", що використовується тут, включає будь-яку систему передачі даних для передачі або віщання, наприклад, аудіовізуальних або мультимедійних цифрових даних. Незважаючи на те, що даний винахід застосовується, зокрема, у віщальній системі цифрового телебачення, він може бути застосований в телекомунікаційній мережі з фіксованою структурою для мультимедійних прикладних Інтернет-програм, в замкненій телевізійній мережі і т.д. Термін "система передачі цифрових даних", що використовується тут, включає, наприклад, будь-яку супутникову, наземну, кабельну або іншу систему.

Даний винахід знаходить конкретне застосування у віщальній системі цифрового телебачення, в якій сигнали, що приймаються, подаються через приймач в приймач/декодер і далі в телевізор. Термін "приймач/декодер", що використовується тут, може означати приймач для прийому або закодованих, або не закодованих сигналів, наприклад, телевізійних і/або радіосигналів, віщання або трансляція яких може здійснюватися деякими іншими засобами. Цей термін може також означати декодер для декодування сигналів, що приймаються. Реалізації таких приймачів/декодерів можуть включати декодер, об'єднаний з приймачем для декодування сигналів, що приймаються, наприклад, в приставці, або декодер, який функціонує в поєднанні з фізично окремим приймачем, або декодер, що містить додаткові функції, такі як веб-браузер, пристрій для відеозапису або телевізор.

Приймач/декодер декодує ущільнений сигнал типу MPEG в телевізійний сигнал для телевізора. Він керується ручним пристроєм дистанційного управління через інтерфейс в приймачі/декодері, званому також приставкою або STB (Set-Top-Box). Термін MPEG відноситься до стандартів передачі даних, розроблених робочою групою Motion Pictures Experts Group (Група Експертів по Рухомих Зображеннях) Міжнародної організації по стандартах і, зокрема, але не виключно, стандарту MPEG-2, розробленому для прикладних програм цифрового телебачення і викладеному в документах ISO 13818-1, ISO 13818-2, ISO 13818-3 і ISO 13818-4. У контексті цієї патентної заявки цей термін включає всі варіанти, модифікації або розробки форматів MPEG, застосовних в області передачі цифрових даних.

Одним зі способів надання описаного вище інтерактивного режиму роботи є виконання прикладної програми в приймачі/декодері, за допомогою якого приймається телевізійний сигнал. Бажано забезпечити можливість взаємодії різних прикладних програм з різними фізичними пристроями в прозорому режимі. У наших заявках РСТ/EP97/02115 і РСТ/EP97/02116, що розглядаються водночас з цією заявкою, описуються системи, в яких приймачем/декодером можуть бути завантажені одна або декілька прикладних програм, які можуть здійснювати обмін інформацією з фізичними пристроями в приймачі/декодері, такими як паралельний і послідовний інтерфейси і пристрої зчитування смарт-карт, за допомогою драйвера пристрою для кожного пристрою і загального менеджера пристроїв. Термін "смарт-карта", що використовується тут, включає, але також не виключно, будь-який пристрій типу карти на основі мікросхеми або об'єкт з аналогічними функціями і характеристиками, що містить, наприклад, мікропроцесор і/або засоби пам'яті. У цей термін включені також пристрої, що мають альтернативні по відношенню до карти фізичні форми, наприклад, пристрої в формі ключа, подібні тим, що часто використовуються в системах TV декодерів.

Згідно даного винаходу було запропоновано забезпечити приймачеві/декодеру можливість обміну даними з іншим аудіо/відео обладнанням, наприклад, пристроєм цифрового відеозапису, через високошвидкісний цифровий інтерфейс. Нещодавно розроблений стандарт IEEE 1394 надає перспективний і гнучкий протокол обміну, що забезпечує швидкість послідовного обміну даними близько 100Мбіт/с і вище.

Проблема використання інтерфейсу IEEE 1394 полягає в тому, що шина інтерфейсу може бути скинена, або параметри можуть бути змінені, не приймачем/декодером, а підключеним до шини іншим пристроєм, і це може призвести до проблем для прикладної програми. Як наслідок, можуть бути потрібні більший обсяг пам'яті і більша обчислювальна потужність для виконання більш складних прикладних програм, здатних працювати з цим інтерфейсом. Це збільшило б як вартість кожного приймача/декодера, так і вартість розробки і налагоджування прикладних програм.

Аспекти даного винаходу направлені на усунення проблем обміну даними прикладних програм з такими інтерфейсами. Хоч більшість переваг, що надаються винаходом, відноситься до забезпечення інтерфейсу приймача/декодера з IEEE 1394 або подібним йому інтерфейсом, очевидно, що даний винахід може бути використаний для забезпечення інтерфейсу прикладних програм з інтерфейсами, параметри яких можуть змінюватися засобами, які не є підконтрольними прикладній програмі.

У першому аспекті даного винаходу пропонується спосіб обміну даними через драйвер пристрою між прикладною програмою і інтерфейсом, що має щонайменше один засіб, якому присвоєний інтерфейсний ідентифікатор, при цьому вказаний один або кожний інтерфейсний ідентифікатор може бути схильний до зміни після щонайменше однієї події, що передбачає для згаданого щонайменше одного засобу збереження відповідного логічного ідентифікатора, подачу згаданого логічного ідентифікатора прикладній програмі для здійснення асоційованого з відповідним засобом обміну даними між драйвером пристрою і прикладною програмою, і підтримка відповідності між згаданим одним або кожним логічним ідентифікатором і згаданим одним або кожним засобом незалежно від інтерфейсного ідентифікатора, присвоєного згаданому одному або кожному засобу, так що обмін даними між прикладною програмою і драйвером пристрою, здійснюваний з використанням деякого даного логічного ідентифікатора, залишається асоційованим з відповідним даним засобом після зміни присвоєного цьому засобу відповідного інтерфейсного ідентифікатора.

Таким чином, незважаючи на те, що відповідність між інтерфейсними ідентифікаторами і засобами може час від часу змінюватися, ці зміни будуть практично прозорі для прикладної програми, яка, отже, може бути простішою.

Переважно обмін даними між інтерфейсом і драйвером пристрою здійснюється на основі згаданого одного

або кожного інтерфейсного ідентифікатора; цим полегшується обмін даними з інтерфейсом.

Логічні ідентифікатори можуть бути присвоєні тільки тим засобам, які вказуються однією або декількома прикладними програмами. Таким чином можна зменшити необхідну кількість логічних ідентифікаторів.

У альтернативному варіанті драйвер пристрою може бути виконаний з можливістю формування списку логічних ідентифікаторів і відповідних інтерфейсних ідентифікаторів для всіх згаданих засобів, або для всіх засобів, що задовольняють заздалегідь заданим критеріям, і переважно з можливістю оновлення цього списку кожний раз, коли відбувається додання, видалення або зміна засобу, або при зміні будь-якого інтерфейсного ідентифікатора.

Хоч запропонований спосіб усуває необхідність в знанні прикладною програмою інтерфейсного ідентифікатора, переважно драйвер пристрою виконаний з можливістю передачі прикладній програмі, за запитом, інтерфейсного ідентифікатора, відповідного певному логічному ідентифікатору. Цим істотно спрощується тестування системи, оскільки високорівнева прикладна програма отримує можливість, при необхідності, перевірки правильності роботи інтерфейсу і відповідного йому драйвера пристрою.

Переважно драйвер пристрою виконаний з можливістю прийому запитів від прикладної програми на встановлення з'єднань між фізичними пристроями, підключеними до шини, з використанням щонайменше одного логічного ідентифікатора замість інтерфейсного ідентифікатора. Цим можна спростити управління з'єднаннями прикладною програмою.

Переважно прикладна програма виконана з можливістю обміну даними з драйвером пристрою за допомогою менеджера пристроїв. Менеджер пристроїв дозволяє здійснювати централізоване управління обміном даними, так що декілька прикладних програм можуть обмінюватися даними з декількома пристроями без виникнення конфліктів.

У першій переважній реалізації щонайменше один згаданий інтерфейсний засіб включає в себе периферійний пристрій, підключений до інтерфейсу, і відповідний інтерфейсний ідентифікатор включає в себе фізичну адресу (іноді званий також адресою вузла), присвоєну цьому периферійному пристрою, і логічний ідентифікатор включає в себе логічну адресу (який може також бути названий логічним ідентифікатором периферійного пристрою), присвоєну цьому периферійному пристрою. Таким чином, прикладна програма за допомогою певної даної логічної адреси може продовжувати обмін даними з периферійним пристроєм (наприклад, пристроєм цифрового відеозапису) навіть якщо фізична адреса периферійного пристрою змінилася (наприклад, внаслідок підключення до шини іншого периферійного пристрою і подальшого скидання шини).

У такому випадку переважно підтримка відповідності включає опитування кожного периферійного пристрою, якому присвоєна логічна адреса, з метою визначення фізичної адреси, присвоєної периферійному пристрою після згаданої однієї або кожної події, наприклад, скидання шини. Цим забезпечується оновлення присвоєних адрес після будь-якої вірогідної зміни фізичної адреси.

Також в цьому випадку особливі переваги забезпечуються у випадку, коли операція передачі інтерфейсного ідентифікатора для заданого периферійного пристрою включає передачу фізичної адреси (або адреси вузла) периферійного пристрою і також включає передачу додаткового ідентифікатора периферійного пристрою, наприклад, унікального ідентифікатора вузла, що містить додаткову інформацію, що ідентифікує периферійний пристрій. Унікальний ідентифікатор вузла може вказувати номер виробника і/або постачальника і/або моделі периферійного пристрою і може включати серійний номер. Переважно унікальний ідентифікатор вузла має розмір щонайменше 4 байти, більш переважно — 8 байтів.

Згідно з другою переважною реалізацією, щонайменше один згаданий інтерфейсний засіб включає в себе канал із заданими параметрами, доступний через інтерфейс, і відповідний інтерфейсний ідентифікатор містить номер інтерфейсного каналу (або так званий ідентифікатор каналу), при цьому логічний ідентифікатор містить логічний ідентифікатор каналу. Таким чином, для прикладної програми усувається необхідність у відстеженні номерів інтерфейсних каналів, які можуть змінюватися. Канали переважно є ізохронними каналами, що мають задану пропускну спроможність.

Переважно драйвер пристрою виконаний з можливістю прийому від прикладної програми запиту на виділення каналу із заданими параметрами (наприклад каналу, що має деяку задану максимальну пропускну спроможність) і повернення логічного ідентифікатора каналу, якщо канал був виділений успішно. Хоч для прикладної програми немає необхідності знати номер інтерфейсного каналу, переважно драйвер пристрою виконаний з можливістю прийому номера переважного інтерфейсного каналу, для виділення згаданого переважного інтерфейсного каналу, якщо він доступний, і виділення вільного каналу, якщо переважний інтерфейсний канал недоступний, або якщо переважний інтерфейсний канал не вказаний. Надання можливості вказування інтерфейсних каналів може полегшити управління і перевірку інтерфейсу відповідним чином створеною прикладною програмою, без необхідності в розрізненні номерів інтерфейсних каналів обов'язково всіма прикладними програмами. Переважно драйвер пристрою виконаний з можливістю прийому ідентифікатора переважного інтерфейсного каналу і розпізнавання заздалегідь заданого коду замість реального номера інтерфейсного каналу, як вказівки на відсутність переважного інтерфейсного каналу, і повідомлення прикладній програмі про помилку, якщо вказані інші некоректні номери інтерфейсних каналів; це може сприяти налагоджуванню прикладних програм.

Переважно також драйвер пристрою виконаний з можливістю передачі прикладній програмі ідентифікатора інтерфейсного каналу, і переважно також інших параметрів, переважно включаючи хоч би один з наступних: максимальна виділена для каналу швидкість передачі даних, можлива в даний момент швидкість передачі даних, кількість з'єднань, що використовують даний канал, і ідентифікатори кожного зі з'єднань, що використовують даний канал. Цим забезпечується можливість "інтелектуального" управління відповідним чином створеною прикладною програмою обміном даними, і при цьому не потрібно, щоб всі прикладні програми при використанні інтерфейсу мали справу з такими параметрами.

У найбільш переважному варіанті перша і друга переважні реалізації використовуються разом, драйвер пристрою виконаний з можливістю прийому від прикладної програми запитів на встановлення одного або

декількох з'єднань між периферійними пристроями, підключеними до інтерфейсу, шляхом посилання на логічні адреси і логічні ідентифікатори каналів. Подібне поєднання двох реалізацій забезпечує синергетичні переваги, що полягають в тому, що прикладна програма може встановлювати з'єднання без необхідності відстеження ані-якої інформації про фізичні адреси задіяних периферійних пристроїв або інтерфейсного каналу, через який встановлюється з'єднання. Переважно драйвер пристрою виконаний з можливістю встановлення хоч би одного з наступних видів з'єднань: двохточкові з'єднання між конкретними периферійними пристроями і широкомовні з'єднання.

Протягом тих деяких подій, коли параметри інтерфейсу зазнають змін, таких як скидання шини, обмін даними може бути перерваний. Хоч драйвер пристрою може обробляти деякі події без необхідності у вхідній інформації від прикладної програми, переважно драйвер пристрою виконаний з можливістю повідомлення прикладній програмі про одну або декілька подій (у разі запиту прикладною програмою), де згадані події переважно включають хоч би одну з наступних: скидання шини (переважно про початок і закінчення скидання шини сигналізують окремі події), зміна параметрів топології шини, параметрів каналу або з'єднання.

У другому аспекті даного винаходу пропонується драйвер пристрою для здійснення обміну даними між прикладною програмою і інтерфейсом, що має щонайменше один засіб, якому присвоюється інтерфейсний ідентифікатор, при цьому згаданий один або кожний інтерфейсний ідентифікатор може бути змінений внаслідок щонайменше однієї події, згаданий драйвер пристрою містить засоби для зберігання щонайменше одного логічного ідентифікатора, відповідного відповідному інтерфейсному ідентифікатору, засоби для подачі прикладною програмою згаданого логічного ідентифікатора для здійснення обміну даними, асоційованого з відповідним засобом, між драйвером пристрою і прикладною програмою, і засобу для підтримки відповідності між згаданим одним або кожним логічним ідентифікатором і згаданим одним або кожним засобом незалежно від інтерфейсного ідентифікатора, присвоєного згаданому одному або кожному засобу, так що обмін даними між прикладною програмою і драйвером пристрою, здійснюваний з використанням деякого даного логічного ідентифікатора, залишається асоційованим з відповідним даним засобом після зміни присвоєного цьому засобу відповідного інтерфейсного ідентифікатора.

Драйвер пристрою може бути реалізований апаратно, наприклад, в спеціалізованій інтегральній схемі; це може забезпечити підвищену швидкість функціонування. Однак переважно принаймні частина драйвера пристрою реалізована програмно, і переважно виконується обчислювальним засобом, що виконує прикладну програму; це забезпечує велику гнучкість, вимагає менше компонентів і дозволяє простіше оновлювати драйвер пристрою.

У третьому аспекті даного винаходу пропонується система обробки даних, що містить підсистему реального часу для виконання прикладної програми, інтерфейс для з'єднання з щонайменше одним пристроєм, який має щонайменше один засіб, якому присвоюється інтерфейсний ідентифікатор, при цьому згаданий один або кожний інтерфейсний ідентифікатор може бути змінений внаслідок щонайменше однієї події, і драйвер пристрою, що містить засоби для зберігання щонайменше одного логічного ідентифікатора, відповідного відповідному інтерфейсному ідентифікатору, засоби для подачі прикладній програмі згаданого логічного ідентифікатора для здійснення обміну даними, асоційованого з відповідним засобом, між драйвером пристрою і прикладною програмою, і засобу для підтримки відповідності між згаданим одним або кожним логічним ідентифікатором і згаданим одним або кожним засобом незалежно від інтерфейсного ідентифікатора, присвоєного згаданому одному або кожному засобу, так що обмін даними між прикладною програмою і драйвером пристрою, здійснюваний з використанням деякого даного логічного ідентифікатора, залишається асоційованим з відповідним даним засобом після зміни присвоєного цьому засобу відповідного інтерфейсного ідентифікатора.

Переважні особливості першого аспекту можуть бути застосовані до другого і третього аспектів.

Переважно система обробки даних реалізована в приймачі/декодері (наприклад, в приставці), який містить засоби для прийому віщальних даних (через супутник або по кабелю), причому інтерфейс переважно виконаний з можливістю з'єднання з пристроєм цифрового відеозапису, або з цифровим пристроєм відображення, або з комп'ютером, для відображення або збереження щонайменше частини прийнятих даних. Драйвер пристрою переважно виконаний з можливістю взаємодії з засобами пристрою для модифікації потоку даних, що приймається, з метою отримання зміненого потоку даних для передачі на згаданий інтерфейс.

Інтерфейс переважно відповідає стандарту IEEE 1394 або його модифікації, удосконаленню або варіації. Дані можуть передаватися згідно з стандартом IEEE 1883.

Прикладна програма переважно виконується на мові, що інтерпретується, а драйвер пристрою переважно скомпільований.

Даний винахід використовується із забезпеченням найбільших переваг в приймачі/декодері для забезпечення обміну даними по шині IEEE 1394 між прикладною програмою і, наприклад, пристроєм цифрового відеозапису. Драйвер пристрою може бути виконаний з можливістю передачі команд прикладної програми для управління пристроєм цифрового відеозапису, і/або з можливістю прийому даних, що відносяться до інформації, що зберігається в пристрої цифрового відеозапису; інтерактивна прикладна програма, що виконується в приймачі/декодері, таким чином може управляти записом і відтворенням програм або інших даних. Дані, що передаються, переважно є даними формату MPEG (під яким розуміється будь-який варіант або розвиток базового формату MPEG), але можуть використовуватися і інші формати.

Нижче будуть описані, виключно у вигляді прикладу, переважні особливості даного винаходу з використанням прикладених малюнків, на яких:

Фіг. 1 є схематичною діаграмою інтерфейсів приймача/декодера;

Фіг. 2 є функціональною схемою приймача/декодера;

На фіг. 3 більш детально показані деякі компоненти віртуальної машини і підсистеми реального часу;

Фіг. 4 є схематичною діаграмою для пояснення обміну даними між прикладною програмою і віддаленим периферійним пристроєм через драйвер пристрою; і

Фіг. 5 схематичною діаграмою, що ілюструє деякі компоненти драйвера пристрою.

Основні відомості про приймач/декодер

Щоб полегшити розуміння пристрою, спочатку надамо короткий опис переважної платформи, на якій функціонує цей пристрій, - нашого цифрового супутникового приймача/декодера.

На фіг. 1 схематично зображений приймач/декодер 2020 або приставка для використання в інтерактивній системі цифрового телебачення, в якій має бути інстальований драйвер пристрою системи, що пропонується. Докладний опис відповідної інтерактивної системи цифрового телебачення може бути знайдений в заявках РСТ/EP97/02106-02117, що одночасно розглядаються, зміст яких використовується тут шляхом посилання. Для спрощення посилань елементи, описані більш детально в згаданих вище описах, звичайно позначаються числовими позиціями, використаними в цих описах.

Як детально описано в згаданих вище описах, згідно з фіг. 1, приймач/декодер 2020 містить декілька інтерфейсів; зокрема, тюнер 4028 для потоку сигналів MPEG, послідовний інтерфейс 4030, паралельний інтерфейс 4032 і два пристрої зчитування карти 4036, один для смарт-карти, що є частиною системи, і один для банківської карти (що використовується для здійснення платежів, банківських розрахунків на дому і т.д.) або іншої смарт-карти. Приймач/декодер містить також інтерфейс зворотного модемного каналу 4002 для зв'язку з провайдером телевізійного сигналу, так що користувач може вказати свої уподобання і т.п. провайдеру телевізійного сигналу (програми). Приймач містить також підсистему реального часу (RTE - Real Time Engine) 4008, менеджер пристроїв 4068 і множину пристроїв 4062 для виконання однієї або декількох прикладних програм 4056.

Застосовно до даного опису, прикладна програма - це фрагмент комп'ютерного коду для управління високорівневими операціями приймача/декодера 2020. Наприклад, коли кінцевий користувач позиціонує фокус пристрою дистанційного управління на об'єкті кнопки, видимому на екрані телевізора 2022, і натискає клавішу підтвердження, виконується послідовність інструкцій, відповідна цій кнопці.

Інтерактивна прикладна програма забезпечує меню і виконує команди по відповідному запиту кінцевого користувача і надає дані відповідно до призначення даної прикладної програми. Прикладні програми можуть бути або резидентними, тобто такими, що зберігаються в ПЗП (або у FLASH пам'яті, або іншій енергонезалежній пам'яті) приймача/декодера 2020, або такими, що передаються шляхом віщання і що завантажуються в ОЗП (або FLASH пам'ять) приймача/декодера 2020.

Прикладами декількох прикладних програм, описаних більш детально в попередніх заявках, є:

- Прикладна програма ініціалізації. Прикладна програма, яка є набором модулів, що адаптується, і забезпечує готовність приймача/декодера 2020 до роботи в середовищі MPEG-2.
- Прикладна програма запуску дозволяє виконуватися в приймачі/декодері 2020 будь-якій прикладній програмі, як такої, що завантажується, так і резидентній.
- Керівництво по програмах - це інтерактивна прикладна програма, яка надає повну інформацію по програмах.
- Прикладна програма PPV (PPV - Pay Per View, оплата за перегляд, оплата здійснюється за кожну переглянену передачу) - це інтерактивна послуга, що є на кожному PPV каналі "букета" цифрового ТБ, що дозволяє користувачеві купити поточну телевізійну програму.
- Прикладна програма завантаження в ПК дозволяє кінцевому користувачеві завантажувати комп'ютерне програмне забезпечення з використанням прикладної програми завантаження в ПК.
- Прикладна програма перегляду журналів, що містить циклічне віщання відео зображень, по яким користувач може переміщатися за допомогою кнопок, що видаються на екран.
- Прикладна програма дистанційних покупок дозволяє транслювати в приймач/декодер пропозиції до продажу товарів і потім виводити їх на екран телевізора 2022, а також вибирати для покупки будь-який необхідний товар.

Прикладні програми зберігаються в елементах пам'яті приймача/декодера 2020 і представляються у вигляді файлів ресурсів. Під файлами ресурсів розуміють файли бібліотек описів графічних об'єктів, файли бібліотек блоків змінних, файли послідовностей команд, файли прикладних програм і файли даних, як більш детально описано в згаданих вище описах заявок.

У потоці даних MPEG кожний модуль містить групу MPEG таблиць. Кожна MPEG таблиця може бути відформатована у вигляді деякого числа секцій. У потоку даних MPEG кожна секція має "розмір" до 4 кбайт. Для передачі даних через, наприклад, послідовний і паралельний порт, модулі точно так само розділяються на таблиці і секції, причому розмір секції варіюється в залежності від середовища передачі.

Модулі передаються в потоці даних MPEG у вигляді пакетів даних, звичайно по 188 байтів, в потоках даних відповідних типів, наприклад, потоків відео даних, потоках аудіо даних і потоках даних телетексту. Кожному пакетові передують ідентифікатор пакету (PID) розміром 13 бітів, по одному PID для кожного пакету, що передається в потоці даних MPEG. Таблиця відображення програм (PMT) містить список різних потоків даних і визначає вміст кожного потоку даних у відповідності з відповідним PID. PID може оповістити пристрій про присутність прикладної програми в потоці даних, і цей PID ідентифікується з використанням таблиці PMT.

Декодер містить пам'ять, розділену на том оперативної пам'яті, том флеш-пам'яті і том ПЗП, але фізична організація пам'яті відрізняється від логічної організації. Пам'ять може бути додатково розділена на томи пам'яті, асоційовані з різними інтерфейсами. З однієї точки зору, пам'ять може розглядатися як частина апаратних засобів; з іншої точки зору, пам'ять може розглядатися як підтримуюча або така, що містить, всю систему, показану окремо від апаратних засобів.

Система може розглядатися як така, що засновується на підсистемі реального часу 4008, що є частиною віртуальної машини 4007. Вона пов'язана, з одного боку (сторона "верхнього рівня"), з прикладними програмами, і, з іншого боку (сторона "нижнього рівня"), через різні проміжні логічні пристрої, що описуються нижче, з апаратними засобами 4061 приймача/декодера. Апаратні засоби приймача/декодера можуть розглядатися як такі, що містять різні порти або інтерфейси, як описано вище (інтерфейс 2030 для пристроєм дистанційного управління 2026, інтерфейс потоку MPEG 4028, послідовний інтерфейс 4030, паралельний інтерфейс 4032, інтерфейси для пристроїв зчитування карт 4036 і інтерфейс 4034 для зворотного модемного

каналу 4002).

Як показано на фіг. 2, різні прикладні програми 4056 пов'язані з пристроєм 4007; деякі прикладні програми, що найчастіше використовуються, можуть бути більш або менш безперервно резидентними в системі, як показано позицією 4057, тоді як інші будуть за потребою завантажуватися в систему, наприклад, з потоку даних MPEG або через інші порти.

Блок 4007 містить, в доповнення до підсистеми реального часу 4008, резидентну бібліотеку функцій 4006, яка містить набір інструментальних засобів 4058. Бібліотека містить різноманітні функції на мові Сі, що використовуються підсистемою 4008. У їх число входять функції обробки даних, такі як ущільнення, декомпресія або порівняння структур даних, малювання ліній і т.д. Бібліотека 4006 містить також інформацію про мікропрограмне забезпечення (firmware) 4060 в приймачі/декодері 2020, таку як номери версій апаратного і програмного забезпечення і наявний обсяг оперативної пам'яті, і функцію, що використовується при завантаженні нового пристрою 4062. Функції в бібліотеку можуть завантажуватися, із збереженням у флеш-пам'яті або оперативній пам'яті.

Підсистема реального часу 4008 пов'язана з менеджером пристроїв 4068, який з'єднаний з множиною пристроїв 4064, з'єднаних з драйверами пристроїв 4060, які, в свою чергу, пов'язані з портами або інтерфейсами. У більш загальному значенні, драйвер пристрою може розглядатися як такий, що визначає логічний інтерфейс, так що два різних драйвери пристроїв можуть бути пов'язані із спільним фізичним портом. Пристрій звичайно пов'язаний з кількома драйверами пристроїв; якщо пристрій пов'язаний з одним драйвером пристрою, цей пристрій звичайно проектується таким, що повністю задіює функціональність, що забезпечує обмін інформацією, так що необхідність в окремому драйвері пристрою відсутня. Деякі пристрої можуть обмінюватися інформацією один з одним.

Як буде описано нижче, існують три способи передачі інформації від пристроїв 4064 в підсистему реального часу: за допомогою змінних, буферів і подій, які вміщуються в декілька черг подій.

Кожна функція приймача/декодера 2020 представляється як пристрій 4062. Пристрої можуть бути локальними або віддаленими. Локальні пристрої 4064 включають в себе smart-карти, SCART-роз'єм, модеми, послідовний і паралельний інтерфейси, MPEG відео- і аудіопротравач і пристрій виділення MPEG секцій і таблиць. Віддалений пристрій 4066, що функціонує у віддаленому місці, відрізняється від локального пристрою тим, що порт і протокол мають бути визначені розробником або проектувальником системи, а не пристроєм або драйвером пристрою, наданим і спроектованим виробником приймача/декодера.

Коли створюється новий пристрій 4062, він може бути інстальований в існуючому приймачі/декодері 2020 шляхом завантаження відповідної прикладної програми 4056 з центра віщання. Це завантаження виконується в приймачі/декодері 2020 прикладною програмою 4056, яка перевіряє версії апаратного і програмного забезпечення і, якщо все правильно, завантажує модуль програмного забезпечення, що представляє новий пристрій 4062, і викликає процедуру з бібліотеки 4006 для інсталяції коду нового пристрою як частини мікропрограмного забезпечення (у флеш-пам'яті). Це може забезпечити гнучку і безпечну інсталяцію нових функцій в приймачі/декодері 2020 без аніякого впливу на інше програмне забезпечення.

Менеджер пристроїв 4068 являє собою загальний програмний інтерфейс між прикладною програмою 4056 і конкретною функцією приймача/декодера 2020. Менеджер пристроїв 4068 управляє доступом до пристроїв 4062, повідомляє про прийом несподіваної події і управляє пам'яттю, що спільно використовується.

Підсистема реального часу 4008 виконується під управлінням мікропроцесора і загального програмного інтерфейсу прикладних програм. Вони інстальуються в кожному приймачі/декодері 2020, так що всі приймачі/декодери 2020 з точки зору прикладної програми ідентичні.

Підсистема 4008 виконує прикладні програми 4056 в приймачі/декодері 2020. Вона виконує інтерактивні прикладні програми 4056 і приймає події, зовнішні по відношенню до приймача/декодера 2020, видає графічні зображення і текст, здійснює виклики пристроїв для надання тих або інших сервісів і використовує функції бібліотеки 4006, підключеної до підсистеми 4008, для спеціальних обчислень.

Підсистема реального часу 4008 являє собою код, що виконується, інстальований в кожному приймачі/декодері 2020, і містить інтерпретатор для інтерпретації та виконання прикладних програм. Підсистема 4008 може бути адаптована до будь-якої операційної системи, включаючи однозадачну операційну систему (таку як MS-DOS). Підсистема 4008 заснована на контролерах послідовності обробки (які обробляють різні події, такі як натиснення клавіші, для виконання різних дій), і містить свій власний планувальник для управління чергами подій від різних апаратних інтерфейсів. Вона управляє також відображенням графіки і тексту. Контролер послідовності обробки містить множину груп дій. Кожна подія призводить до того, що контролер послідовності обробки переміщається від поточної групи дій до іншої групи дій в залежності від природи події і виконує дії нової групи дій.

Підсистема 4008 містить завантажувач коду для завантаження прикладної програми 4056 в пам'ять приймача/декодера 2028. У оперативну пам'ять або флеш-пам'ять завантажуються тільки необхідний код, таким чином забезпечуючи ефективність використання пам'яті. Дані, що завантажуються, перевіряються засобами перевірки автентичності з метою запобігання будь-яким змінам прикладної програми 4056 або виконання якоїсь невідомої прикладної програми. Підсистема 4008 містить додатково декомпресор. Оскільки код прикладної програми (один з видів проміжного коду) є ущільненим з метою економії простору і швидкості завантаження з транспортного потоку MPEG або через інтегрований модем приймача/декодера 2020, перед завантаженням в оперативну пам'ять необхідно зробити декомпресію коду. Підсистема 4008 містить також інтерпретатор проміжного коду для інтерпретації коду прикладної програми, для оновлення значень різних змінних і визначення змін стану, а також блок контролю помилок.

Перед використанням сервісів будь-якого пристрою 4062 програма (така як послідовність команд прикладної програми) має бути оголошена як "клієнт", тобто як логічний шлях доступу до пристрою 4062 або менеджера пристроїв 4068. Менеджер пристроїв привласнює клієнту номер клієнта, на який проводиться посилання при всіх зверненнях до пристрою. Пристрій 4062 може мати декілька клієнтів, кількість клієнтів для кожного пристрою 4062 задається в залежності від типу пристрою 4062. Клієнт "представляється" пристрою

4062 за допомогою процедури "Device: Open Channel". Ця процедура присвоює клієнту номер клієнта. Клієнт може бути видалений зі списку клієнтів менеджера пристроїв 4068 за допомогою процедури "Device: Close Channel".

Доступ до пристроїв 4062, що надається менеджером пристроїв 4068, може бути або синхронним, або асинхронним. Для синхронного доступу використовується процедура "Device: Call". Цей засіб для доступу до даних, які доступні на даний момент, або до функцій, які не передбачають очікування до отримання бажаної відповіді. Для асинхронного доступу використовується процедура "Device: I/O". Цей засіб для доступу до даних, які передбачають очікування до отримання відповіді, наприклад, сканування частот тюнера або виділення таблиці з потоку MPEG. Коли необхідний результат отриманий, в чергу підсистеми вміщується подія, щоб повідомити про свою появу. Ще одна процедура, "Device: Event", надає засоби управління несподіваними подіями.

Як відмічалось вище, основний цикл підсистеми реального часу пов'язаний з різними контролерами послідовності обробки, і коли основний цикл зустрічає відповідну подію, управління тимчасово передається одному з контролерів послідовності обробки.

Як показано на фіг. 3, менеджер пристроїв містить чергу 100, в яку події від пристроїв передаються для тимчасового зберігання. Через відповідні інтервали часу віртуальна машина посилає в цю чергу сигнал на відбір з неї першого елемента. Цей елемент-подія пересилається в структуру черг 101 у віртуальній машині. У залежності від рівня пріоритету елемента-події, вона вміщується в одну з 5 черг з відповідним пріоритетом від 0 до 4. Елементи-події вибираються з структури черг 101 пристроєм вибірки 102 під управлінням підсистеми реального часу.

Коли подія вибирається з структури черг 101, вона передається в підсистему контролерів послідовності обробки 104, яка складається з драйвера контролерів послідовності обробки 105 і множини контролерів послідовності обробки 106. Кожний контролер послідовності обробки являє собою набір взаємопов'язаних груп дій, так що кожний крок з однієї групи дій в наступну групу дій залежить, в загальному випадку, від поточної групи дій і від природи події. Різні контролери послідовності обробки мають різні розміри і складність, включаючи той, в якому "наступна" група дій, наприклад, група дій, до якої система переходить у відповідь на подію, залежить тільки від природи події, але не залежить від поточної групи дій. Крім цього, як показано в правій частині блоку контролерів послідовності обробки, може існувати декілька копій контролера послідовності обробки, тобто декілька ідентичних контролерів послідовності обробки, що мають справу, наприклад, з декількома окремими потоками даних з використанням ідентичних протоколів для одного порту.

Коли подія відбирається з структури черг 101, вона передається у відповідний контролер послідовності обробки. За допомогою нього вибирається відповідний вихід з поточної групи дій контролера послідовності обробки. Це приводить до вибору відповідної наступної групи дій, а також виконання дій в цій групі дій, включаючи, наприклад, відсилання повідомлення менеджеру пристроїв або виконання послідовності команд. Групи дій в контролері послідовності обробки можуть також посилати повідомлення про події в інші контролери послідовності обробки.

Якщо вибрана послідовність команд, ідентифікатор цієї послідовності команд відсилається в пристрій вибірки послідовності команд 107. Він отримує бажану послідовність команд з пам'яті послідовностей команд 108 і передає її в інтерпретатор послідовностей команд 109, який виконує дану послідовність команд.

Система також містить фільтр 110, в який завантажуються типи подій, наприклад, від контролерів послідовності обробки 106. Коли елемент-подія передається з черги 100 менеджера пристроїв в структуру черг 101 у віртуальній машині, тип або характер (природа) події перевіряється на відповідність списку в фільтрі 110, і, якщо ця подія типу, який не впізнаний, вона відкидається. Цим гарантується, що у випадку, якщо менеджер пристроїв або клавіатура генерує події такого типу, з яким віртуальна машина працювати не може, то ці події не передаються в структуру черг 101 (якщо події такого роду передавалися б в структуру черг 101, то вони або нагромаджувалися б в цій структурі черг, або могли б викликати збої в підсистемі контролерів послідовності обробки 104).

Таким чином, як видно, наша базова платформа приймача/декодера надає значну гнучкість в забезпеченні обміну даними прикладної програми з різними пристроями.

Драйвер пристрою для шини IEEE 1394

Як показано на фіг. 4, драйвер шини IEEE 1394 функціонує відповідно до описаної вище схеми для забезпечення обміну даними між прикладною програмою і периферійним пристроєм, таким як пристрій цифрового відеозапису, підключеним до шини IEEE 1394.

Для високошвидкісного обміну даними, наприклад, для запису MPEG даних в реальному часі, традиційні послідовні і паралельні інтерфейси, які є відносно просто керованими прикладною програмою, можуть виявитися недостатньо швидкими. Драйвер пристрою, що описується нижче, має декілька нових особливостей, які забезпечують прикладній програмі можливість ефективного використання шини IEEE 1394 і забезпечують можливість управління з допомогою відносно простої прикладної програми, наприклад, підключеним до шини пристроєм цифрового відеозапису.

Драйвер пристрою може розглядатися як такий, що складається з деякого числа функціональних блоків, званих далі командами, до кожного з яких прикладна програма може звертатися окремо. Кожна команда взаємодіє з прикладною програмою під управлінням менеджера пристроїв 4068 за допомогою однієї з трьох стандартних процедур, згаданих вище, спільних і для інших пристроїв. Дані можуть передаватися між прикладною програмою і пристроєм за допомогою таблиць параметрів. Для спрощення опису коротко охарактеризуємо ці три основні процедури:

1) Device: Call. Ця команда може бути використана прикладною програмою для виконання команд або передачі даних в синхронному режимі. Виконання прикладної програми переривається доти, доки не буде повернене управління після завершення роботи драйвера пристрою; цим забезпечується можливість надійного управління виконанням операцій, які повинні виконуватися в суворій послідовності.

2) Device: I/O. Ця команда забезпечує асинхронне функціонування. Це означає, що прикладна програма

може посилати драйверу пристрою запит на передачу даних або виконання деякої конкретної функції, і виконання прикладної програми може продовжуватися, в той час як драйвер пристрою виконує передачу даних або функцію.

3) Device: Event. Ця перехоплююча події функція дозволяє пристрою повідомляти про подію прикладній програмі, а прикладній програмі - робити відповідні дії у відповідь на подію незалежно від коду, який виконує прикладна програма в той момент часу, коли повідомляється про подію; фактично прикладна програма переривається. Подіям можуть бути призначені пріоритети. Події можуть застосовуватися для того, щоб повідомляти про події, що відбуваються на інтерфейсі, такі як скидання шини.

Нижче будуть описані команди, передбачені в драйвері пристрою, що реалізовує даний винахід. Прикладна програма може звертатися до кожної команди шляхом передачі ідентифікатора команди як параметру через одну з процедур - Device: Call або Device: I/O. Не обов'язково всі описані нижче команди мають бути реалізовані в пристрої, і функції команд можуть бути змінені. Хоч команди можуть бути реалізовані або модифіковані незалежно одна від іншої, певні синергетичні переваги можуть бути отримані від спільного використання функціональних можливостей, що забезпечуються командами, що описуються.

Команди будуть описані в термінах особливостей і функцій, що забезпечуються кожною командою, як їх бачить прикладна програма, в тому числі факультативних особливостей і особливостей, яким віддається перевага. За наявності наведених інформації і специфікацій, фактична реалізація цих особливостей буде абсолютно очевидна для фахівця, тому конкретні деталі залишені на розсуд розробника. Наприклад, кожна команда може бути реалізована програмно, переважно написана на мові програмування C і переважно скомпільована для виконання на процесорі, який використовується для виконання прикладної програми; проте драйвер пристрою може виконуватися на окремому процесорі, а деякі або всі команди можуть реалізовуватися спеціалізованими апаратними засобами. Використовуючи згадані команди Call і I/O, драйвер пристрою може повідомляти інформацію або передавати параметри назад прикладній програмі шляхом установки значень в таблиці параметрів, що зберігається в пам'яті, адреса якої передана драйверові пристрою.

Зрозуміло, що функціональні можливості команд, що описуються нижче, передбачають реалізацію драйвером пристрою деяких базових функцій, наприклад, для роботи з логічними ідентифікаторами периферійних пристроїв і логічними ідентифікаторами каналів, при цьому драйвер пристрою містить засоби ведення відповідних таблиць логічних ідентифікаторів периферійних пристроїв і логічних ідентифікаторів каналів, забезпечуючи їх кореляцію відповідним інтерфейсним засобом (фізична адреса або номер інтерфейсного каналу відповідно). Крім цього, драйвер пристрою виконується з можливістю, у разі виникнення такої події, як скидання шини, з'ясування нових фізичних адрес і номерів каналів і оновлення згаданих таблиць, так що перехід є відносно непомітним для прикладної програми.

Додатково, природно, драйвер пристрою містить засоби для реального здійснення обміну даними з інтерфейсом і для виконання необхідних службових задач, таких як виділення пам'яті і звільнення пам'яті. Деякі з цих функцій схематично зображені на фіг. 5. Деталі реалізації цих функцій будуть залежати від конкретних фізичних апаратних засобів, що використовуються, але вони є очевидними для середнього фахівця в даній області, виходячи з інформації, наведеної в даному описі, і з урахуванням відповідних частин документації по стандарту IEEE 1394 (вміст яких включений в даний опис шляхом посилання), і тому тут не описуються.

Команда: Bus 1394 Set

Ця команда дозволяє прикладній програмі задавати основні параметри інтерфейсу, переважно розмір буфера для прийому даних, що він має бути виділений, і кількість спроб обміну даними, що мають бути зроблені при передаванні через інтерфейс асинхронних команд. Ці параметри могли б задаватися заздалегідь, і ця команда могла б бути випущена, але надання даної команди забезпечує оптимізацію обміну даними для різних прикладних програм. Хоч ці параметри можуть бути легко задані в асинхронному режимі, краще все ж звертатися до даної команди за допомогою процедури Call, так що після установки параметрів пристрою виконуються тільки подальші команди прикладної програми. Переважно команда повідомляє прикладній програмі про помилку, якщо драйвер пристрою знаходиться в процесі прийому даних від якогось периферійного пристрою.

Команда: Bus 1394 Info

Ця команда повертає в прикладну програму основну інформацію про топологію шини. Оскільки вона менш критична за часом, доступ до неї здійснюється переважно в асинхронному режимі за допомогою процедури I/O.

Переважно ця і всі, або хоч би деякі, асинхронні команди реалізовані з можливістю передачі максимального часу (наприклад, в мілісекундах), необхідного для відповіді (або коду, наприклад, нуля, що означає, що максимальний час не визначений); це забезпечить драйверу пристрою можливість ранжування запитів за пріоритетом.

Переважно команда повертає інформацію про максимальну швидкість передачі даних по шині, про швидкість передачі даних, можливу на момент виклику (тобто з урахуванням з'єднань через шини, що вже мають місце), про кількість фізично підключених до шини периферійних пристроїв і відповідних ним логічних ідентифікаторів (як буде описано нижче), а також про те, які логічні канали вільні на момент виклику.

При використанні шини IEEE 1394 кожному підключеному до шини периферійному пристроєві присвоюється фізична адреса, яка може змінюватися з перебігом часу.

Зрозуміло, що хоч надання цієї команди є необов'язковим, бажано, щоб драйвер пристрою вів таблицю логічних адрес (званих також логічними ідентифікаторами периферійних пристроїв), які постійні для кожного периферійного пристрою (протягом даного сеансу для даного прикладної програми; логічні адреси можуть змінюватися при скиданні приймача/декодера), так що при кожному сеансі виконання прикладна програма може використовувати одну логічну адресу для унікальної і однозначної ідентифікації відповідного периферійного пристрою. Номери каналів, що присвоюються каналам, також можуть змінюватися, тому

драйвер пристрою веде також таблицю логічних номерів каналів. У такому випадку драйвер пристрою може відповідати на запити на інформацію за допомогою перегляду даних відповідної таблиці.

Переважаю інформація про наявність вільних каналів передається в бінарній формі у вигляді бітового масиву переважно розміром 8 байтів, в якому кожний біт показує, чи вільний один з 64 логічних каналів (наприклад, "0" означає, що канал вже виділений, "1" означає, що канал вільний для використання).

Команда: Bus 1394 Info Periph

Ця команда виконана з можливістю прийому параметра, який визначає логічний ідентифікатор периферійного пристрою, і повернення двобайтової фізичної адреси (відомої також як ідентифікатор вузла), відповідної фізичній адресі, присвоєній згаданому периферійному пристрою на інтерфейсі, і переважно також повернення унікального ідентифікатора вузла, переважно такого, що однозначно ідентифікує периферійний пристрій в глобальному значенні, або принаймні ідентифікує постачальника або номер моделі периферійного пристрою. Цим для спеціальним чином виконаних "інтелектуальних" прикладних програм забезпечується можливість визначення, наприклад, особливих можливостей обладнання, виходячи з інформації, що персонально ідентифікує периферійні пристрої.

Переважаю команда виконана з можливістю повідомлення про помилку, якщо інтерфейс фізично не підключений до функціональної шини IEEE 1394, або якщо логічний ідентифікатор периферійного пристрою є заданим некоректно (наприклад, більший за заздалегідь задане максимальне число, переважно 63), а також для повідомлення про скидання шини, що має місце на даний момент, про помилку у випадку, коли вказаний ідентифікатор периферійного пристрою невідомий, або якщо пристрій не відповідає протягом заданого часу.

Звернення до цієї команди виконується переважно в асинхронному режимі за допомогою процедури Device: Call, і сигнал успішного завершення або відмови передається за допомогою блоку параметрів.

Команда: Bus 1394 Alloc Channel

Ця команда виконана з можливістю прийому запиту на виділення каналу, переважно із вказуванням бажаної швидкості обміну даними, і переважно також із вказуванням певного бажаного для використання інтерфейсного каналу. Заздалегідь заданий код (наприклад, OFFh) може використовуватися для позначення відсутності потреби в певному конкретному інтерфейсному каналі, і в цьому випадку, або у випадку, коли бажаний інтерфейсний канал зайнятий, драйвер пристрою виділяє будь-який вільний канал.

Якщо команда виконана успішно, вона повертає логічний ідентифікатор виділеного каналу; команда переважно повідомляє про помилку у випадках, описаних вище застосовно до команди Bus_1394_Info_Periph, або якщо вільні канали відсутні, або якщо необхідна швидкість обміну даними більша за максимально можливу швидкість.

У спрощеній реалізації драйвера пристрою, наприклад, при використанні дуже обмеженої кількості каналів, ця команда, а також пов'язані з нею дві наступні команди, що описуються нижче, можуть бути випущені за рахунок деякої втрати гнучкості.

Переважаю звернення до цієї команди виконується в асинхронному режимі, за допомогою процедури Device: I/O, і сигнал успішного завершення або відмови передається за допомогою блоку параметрів.

Команда: Bus 1394 Info Channel

Ця команда виконана з можливістю повернення в прикладну програму інформації, що стосується характеристик заданого логічного каналу. Переважаю команда повертає виділену для каналу максимальну швидкість обміну даними (в Кбіт/с), швидкість обміну даними для каналу на момент виклику, реальний ідентифікатор каналу (тобто ідентифікатор, присвоєний інтерфейсом, а не драйвером пристрою), кількість з'єднань, що використовують даний канал, і логічні ідентифікатори кожного зі з'єднань, що використовують цей канал.

Дана команда переважно повідомляє про помилку, якщо канал зі вказаним номером не виділений, якщо ідентифікатор заданий некоректно, якщо в даний момент відбувається скидання шини, або якщо інтерфейс фізично не підключений.

Переважаю звернення до цієї команди виконується в асинхронному режимі, за допомогою процедури Device: I/O, і сигнал успішного завершення або відмови передається за допомогою блоку параметрів.

Команда: Bus 1394 Free Channel

Ця команда звільняє канал для обміну даними шляхом розриву з'єднань логічного каналу, вказаного як параметр (але переважно без скинення ідентифікаторів з'єднань). Переважаю команда функціонує в асинхронному режимі і повідомляє за допомогою події про те, що обмін даними в заданому каналі все ще має місце.

Команда: Bus 1394 Open Connect

Ця команда виконана з можливістю прийому запиту, яким зазначається логічний ідентифікатор каналу і переважно тип з'єднання - чи ініціювати двохточкове з'єднання між двома пристроями або широкомовне з'єднання, в залежності від вказаного типу з'єднання. Коли задається двохточкове з'єднання, драйверові пристрою мають бути також передані логічні ідентифікатори згаданих двох периферійних пристроїв. Хоч варіанти цієї команди могли б оперувати фізичними адресами і реальними інтерфейсними номерами каналів, засноване на логічних параметрах функціонування надає переваги, що полягають в згаданому вище спрощенні роботи прикладної програми.

При успішному виконанні команда повертає логічний ідентифікатор з'єднання.

У спрощених реалізаціях може бути випущена можливість завдання певного двохточкового з'єднання; в типових системах може бути тільки один пристрій, підключений до шини, такий як пристрій цифрового відеозапису, тому широкомовного з'єднання може виявитися досить.

У деяких реалізаціях драйвера пристрою встановлення якого-небудь з'єднання може автоматично приводити до перемаршрутизації інших потоків даних в приймачі/декодері. Наприклад, відкриття широкомовного з'єднання може спричинити автоматичне відключення зовнішнього клієнта від входу демультіплексора, так що демультіплексор може здійснювати обробку даних, що приймаються по шині IEEE 1394.

Переважно ця команда повідомляє про помилку при досягненні максимального числа з'єднань, або в інших застосовних випадках, згаданих вище відносно інших команд.

Переважно звернення до цієї команди виконується в асинхронному режимі, за допомогою процедури Device: I/O, і сигнал успішного завершення або відмови передається за допомогою події.

Команда: Bus 1394 Close Connect

Ця команда приймає логічний ідентифікатор з'єднання і перериває обмін даними через це з'єднання, звільняючи таким чином ідентифікатор з'єднання для повторного використання.

Якщо при встановленні з'єднань в приймачі/декодері автоматично проводиться перемаршрутизація потоків даних, пристрій переважно відновлює з'єднання в попередній стан після закриття з'єднання або після закриття останнього такого з'єднання. Наприклад, вхід демультимплексора може бути знов з'єднаний із зовнішнім клієнтом по перериванню останнього широкомовного з'єднання.

Переважно звернення до цієї команди виконується в асинхронному режимі, за допомогою процедури Device: I/O, і сигнал успішного завершення або відмови передається за допомогою події.

Команда: Bus 1394 List Connect

Ця команда повертає список активних з'єднань, які відносяться до самого приймача/декодера, доступних на час виклику, переважно в формі переліку, що містить кількість з'єднань і для кожного з'єднання - логічний ідентифікатор з'єднання і прапорець, який вказує тип з'єднання (двохточкове, широкомовне вхідне, широкомовне вихідне).

Ця команда і/або команда, що описується нижче, в простих реалізаціях пристрою може, якщо встановлюються тільки прості з'єднання, бути випущена. Однак надання таких команд дозволяє прикладній програмі здійснювати моніторинг не тільки тих з'єднань, які вона сама встановила, але також і з'єднань, встановлених іншими прикладними програмами, якщо в будь-який момент часу драйвер пристрою може використовуватися більш ніж одною прикладною програмою, а також здійснювати моніторинг того, чи не було яке-небудь з'єднання несподівано закрито.

Звернення до цієї команди здійснюють переважно в синхронному режимі, за допомогою процедури Device: Call, оскільки з'єднання схильні до частих змін, і прикладна програма, в іншому випадку, може намагатися управляти обміном даними виходячи із застарілої інформації, або, інакше, буде потрібне опитування драйвера пристрою.

Команда: Bus 1394 Info Connect

Ця команда приймає логічний ідентифікатор з'єднання і повертає логічний номер каналу, по якому встановлене з'єднання. Переважно також команда повертає вказівку на тип з'єднання і, у разі двухточкового з'єднання, повертає логічні адреси задіяних периферійних пристроїв.

Як і для команди List_Connect, до команди звертаються переважно в синхронному режимі.

Команда: Bus 1394 Reset

Ця команда ініціює процедуру скидання шини або повертає помилку, якщо скидання шини відбувається в даний момент. Ця команда може використовуватися для того, щоб дозволити прикладній програмі захопити управління шиною IEEE 1394 відразу ж після скидання шини; до команди звертаються переважно в синхронному режимі. Переважно драйвер пристрою повідомляє про завершення скидання шини за допомогою події, як буде описано нижче.

Команда: Bus 1394 Send FCP

Ця команда особливо може бути випущена або реалізована іншим чином. Наступний опис являє собою опис прикладу реалізації для передачі даних в асинхронному режимі по шині IEEE 1394.

Ця команда приймає блок параметрів, що містить повідомлення для асинхронної передачі по шині IEEE 1394, як інструкцію або відповідь периферійному пристрою. Переважно блок параметрів містить вказівку на тип повідомлення, розмір буфера, який повинен бути виділений для відповіді, логічний ідентифікатор віддаленого периферійного пристрою, розмір повідомлення і саме повідомлення.

Команда переважно повідомляє про успішне завершення або повідомляє про помилку, якщо передача не була успішною після заздалегідь заданого числа спроб, або у випадках, застосовних до описаної вище команди Info_Periph.

Оскільки можуть передаватися великі обсяги даних, звернення до команди виконується переважно в асинхронному режимі, дозволяючи прикладній програмі продовжувати виконання поки здійснюється передача.

Переважно команда виконана з можливістю широкомовлення всім периферійним пристроям у випадку, коли вказаний певний заздалегідь заданий логічний ідентифікатор периферійного пристрою дорівнює, наприклад, 63.

У спрощених реалізаціях драйвера пристрою ця команда може бути обмежена передачею повідомлень фіксованого розміру, наприклад, 32 байти, чого досить для передачі інструкцій пристрою цифрового відеозапису.

Переважно драйвер пристрою здатний приймати і передавати декілька запитів квазіпаралельно і повідомляти про декілька відповідей. У той же час спрощені реалізації можуть забезпечувати можливість роботи тільки з одиночними послідовними запитами.

У доповнення до згаданих команд, які дозволяють прикладній програмі посилати команди драйверу пристрою, драйвер пристрою реалізовується з можливістю повідомлення прикладній програмі про події за допомогою функцій обробки подій менеджера пристроїв. Драйвер пристрою реалізовує наступні події:

Подія: Bus 1394 Rcv FCP

Ця подія повідомляє про прийом кадру FCP від периферійного пристрою і забезпечує блок параметрів, що містить логічну адресу периферійного пристрою-джерела, тип, розмір і вміст повідомлення.

Подія: Bus 1394 Channel

Ця подія повідомляє про виділення або звільнення каналу і передає список, який показує, які канали виділені, переважно закодований в бінарній формі як описано вище для команди Info.

Подія: Bus 1394 Config

Ця подія сигналізує про встановлення або розрив з'єднання із зовнішнім пристроєм і надає список, що містить кількість периферійних пристроїв, що беруть участь в з'єднаннях, і їх логічні адреси.

Зрозуміло, що зміни на інтерфейсі, що стосуються даної події і раніше описаної події Channel, повинні підлягати моніторингу з боку драйвера пристрою для того, щоб підтримувати оновленою таблицю відповідності між логічними ідентифікаторами і інтерфейсними ідентифікаторами, навіть якщо драйвер пристрою не повідомляє прикладній програмі про такі події.

Подія: Bus 1394 Connect

Ця подія використовується для повідомлення про переривання з'єднання, і надає прикладній програмі логічний ідентифікатор перерваного з'єднання, і переважно також список, що містить додаткову інформацію, що стосується перерваного з'єднання, в форматі, аналогічному описаному вище для команди Info_Connection.

Подія: Bus 1394 Lo Events

Ця подія може повідомляти про одну або декілька помилок інтерфейсу низького рівня, наприклад, про захоплення периферійним пристроєм управління шиною на неприпустимо тривкий час, помилки даних або помилки CRC, несподівані транзакції, невідомі номери каналів і коди транзакцій і т.д. Ця подія насамперед корисна для налагоджування, і в спрощених реалізаціях драйвера може бути випущена.

Подія: Bus 1394 Hi Events

Ця подія може повідомляти про один або декілька високорівневих станів шини, що включають щонайменше одну (переважно обидві) з подій початку і закінчення скидання шини, а також такі події, як обрив кабелю живлення, виявлення замикання в шині або поява неусувної помилки, після якої драйвер пристрою не може відновитися самостійно після декількох спроб.

Подія: Bus 1394 Off

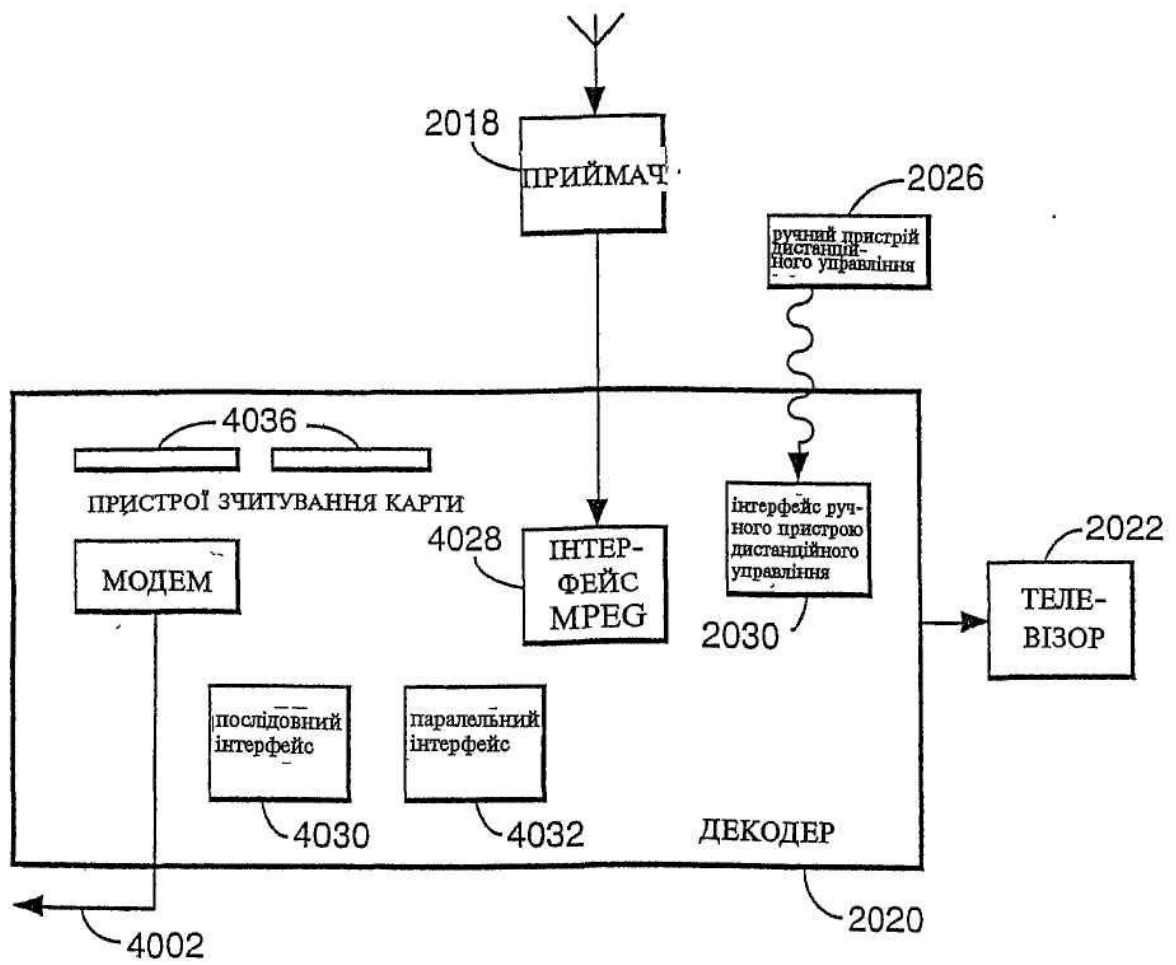
Як додаткова подія, дана подія може використовуватися для повідомлення про внутрішні помилки самого драйвера пристрою, такі як відсутність вільного буфера, в якій можна було б зберегти прийняте повідомлення.

Описані вище команди і події є виключно ілюстративними, і винахід може бути реалізований різними способами, і, зокрема, деякі команди можуть бути об'єднані з іншими, що виконують схожі функції, або деякі з них можуть бути випущені в спрощених реалізаціях. Апаратні і програмні реалізації кожної з функцій можуть вільно комбінуватися, як відносно різних команд, так і в межах однієї команди; апаратні реалізації можуть працювати швидше і вивільняти обчислювальні потужності, в той час як програмні реалізації можуть бути простіше оновлені. Зрозуміло, що функції, що виконуються апаратними засобами, програмним забезпеченням і подібними ним засобами, виконуються з використанням електричних і подібних ним сигналів. Програмні реалізації можуть зберігатися в ПЗП або флеш-пам'яті, або можуть оновлюватися у флеш-пам'яті.

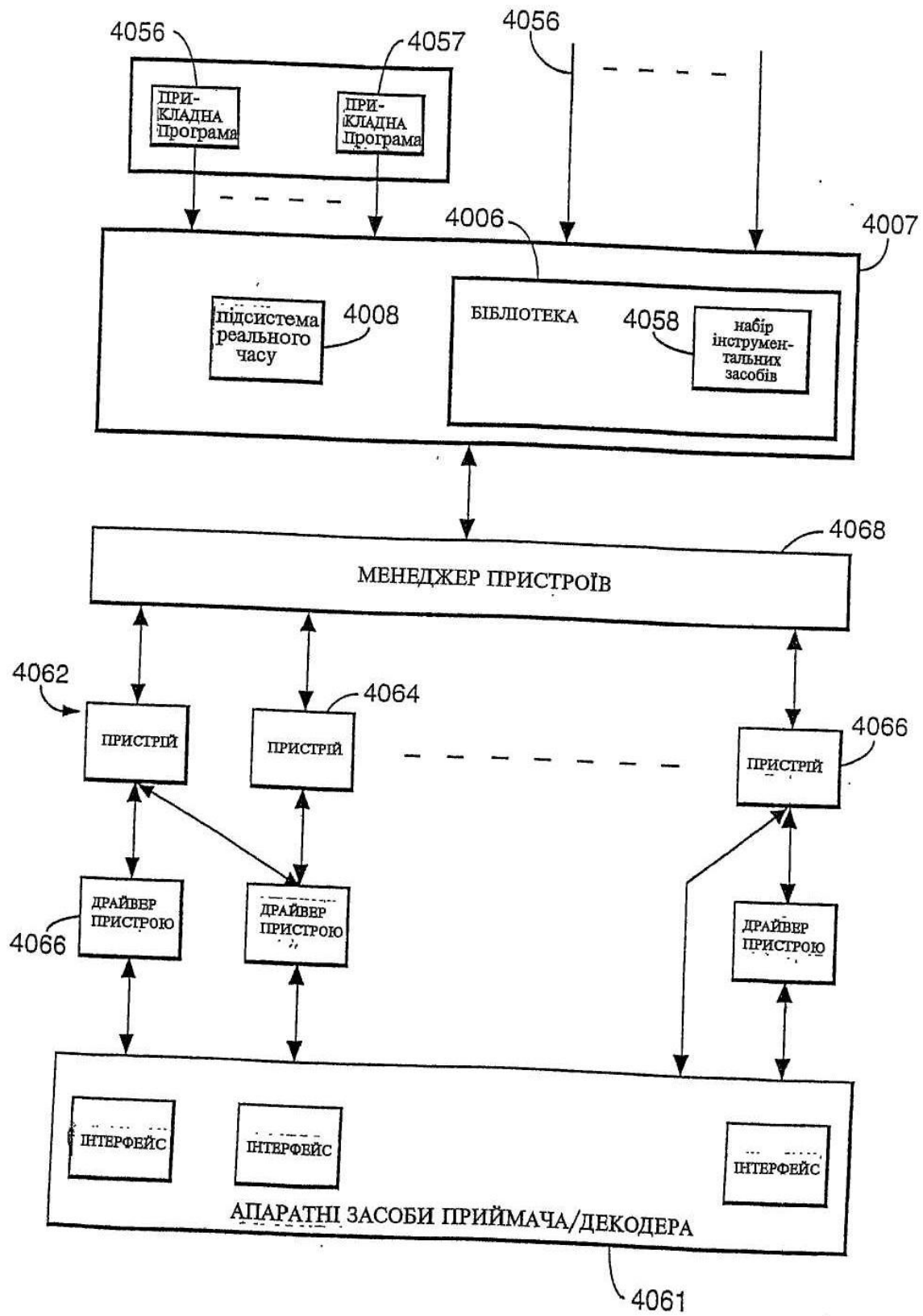
Зрозуміло, що винахід, що пропонується, викладений виключно у вигляді прикладу, і окремі деталі можуть бути змінені в межах обсягу винаходу.

Кожна викладена в описі особливість, а також (де можливо) пункти формули винаходу і малюнки можуть бути надані незалежно або в будь-якому необхідному поєднанні.

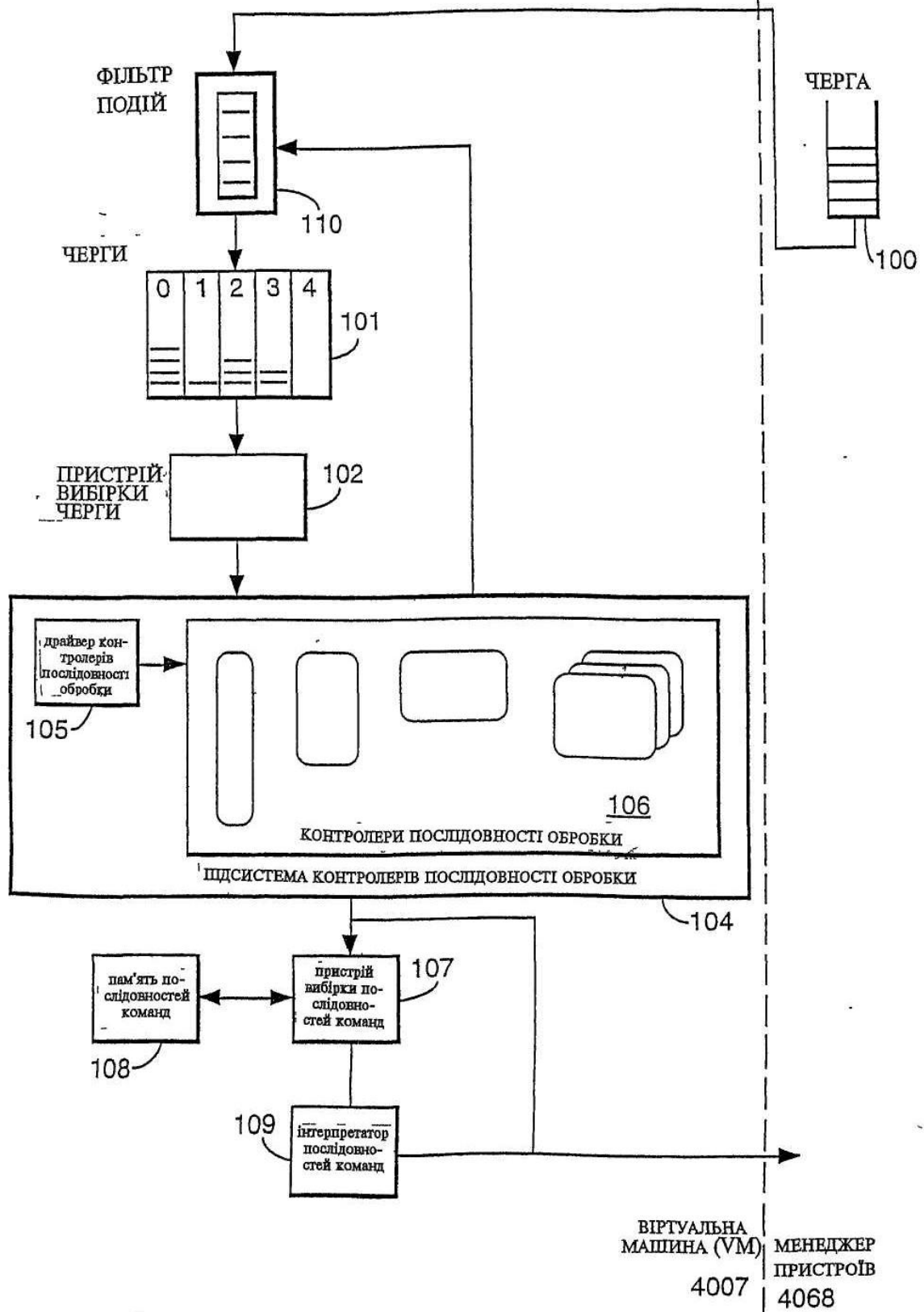
ФІГ. 1



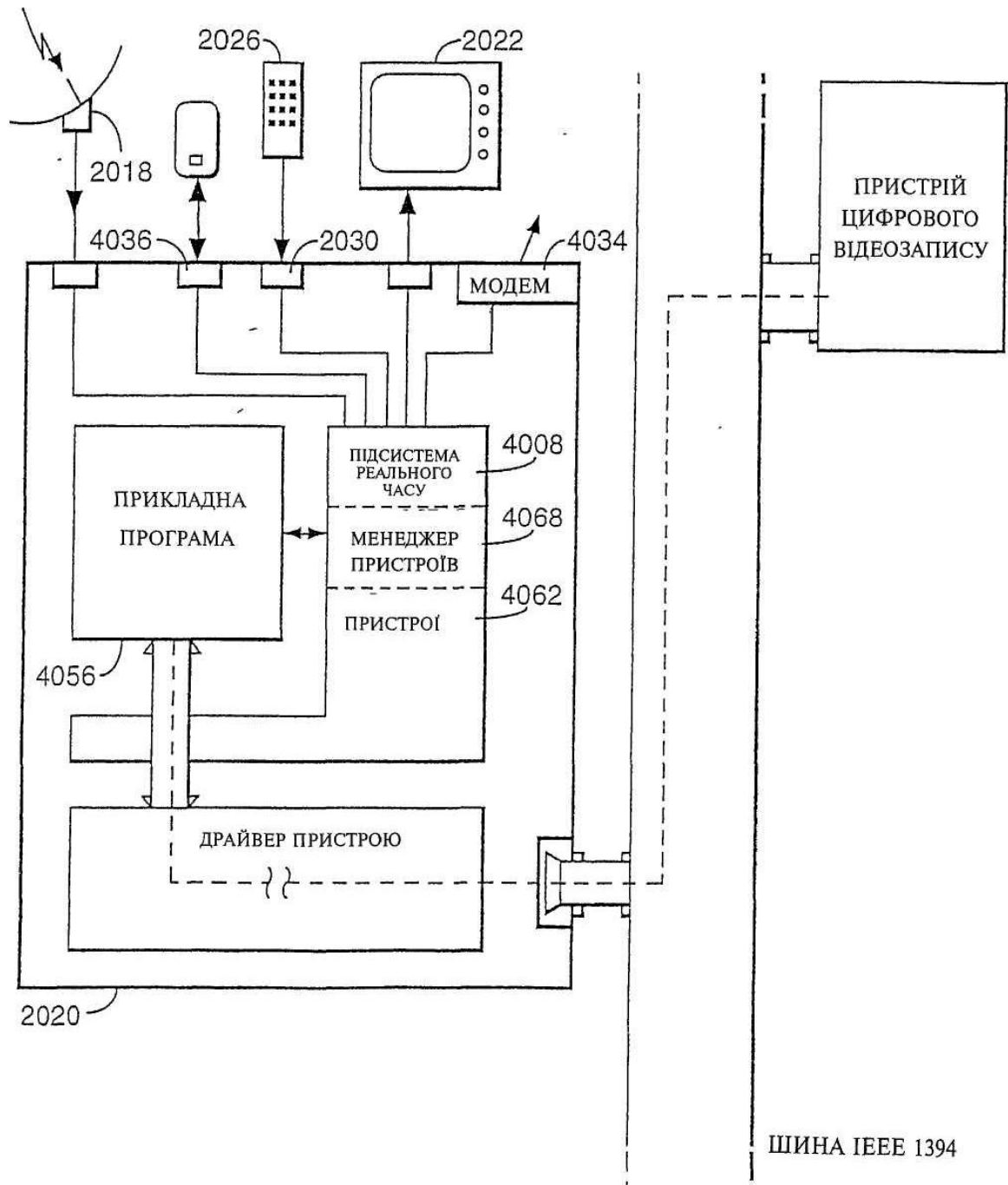
ФІГ. 2



ФІГ. 3



ФІГ. 4



ФІГ. 5

