

Adathozzáférést korlátozó eljárás és berendezés, továbbá vevő és dekódoló berendezés, adatátviteli rendszer, a kettő kombinációja, továbbá adatrögzítő eljárás és berendezés, továbbá eljárás és berendezés adatblokkok mozgatására, továbbá eltárolására

Kivonat

Adathozzáférést korlátozó eljárás memóriával rendelkező vevő és dekódoló berendezésben, illetve digitális műsorszóró rendszerben sugárzott adatokhoz, amelynek során az adatokhoz hozzáférési jogosultság készleteket rendelünk hozzá, ahol minden egyes hozzáférési jogosultság készletet legalább egy félhez társítunk; a vevő és dekódoló berendezés memóriájában az adatokat, a hozzáférési jogosultság készleteket valamint minden egyes félhez tartozó azonosítót eltárolunk; az adatokhoz hozzáférést kérelmező félhez tartozó azonosítót összehasonlítjuk a memóriában tárolt legalább egy azonosítóval; és a vevő és dekódoló berendezés memóriájában hozzárendelt hozzáférési jogosultság készletet a fél rendelkezésére bocsátjuk.

Adathozzáférést korlátozó berendezés, vevő és dekódoló berendezés memóriájában tárolt adatokhoz, az adatokkal hozzáférési jogosultsági készleteket társítunk és minden egyes hozzáférési jogosultság készletet legalább egy félhez hozzárendelünk, és minden egyes félre vonatkozó azonosítót a vevő és dekódoló berendezésben tárolunk, és a berendezésnek az adatokhoz hozzáférést kérelmező fél azonosítóját a vevő és dekódoló berendezés memóriájában tárolt azonosítókkal összehasonlító eszköze, valamint a fél számára a hozzárendelt hozzáférési jogosultság készletet a vevő és dekódoló berendezés memóriájában lévő, hozzárendelt hozzáférési jogosultság készletet rendelkezésre bocsátó eszköze van.

Vevő és dekódoló berendezés, amely adatokat tároló memóriát tartalmaz, és az adatokhoz hozzárendelt hozzáférési jogosultság készleteket, minden egyes félhez tartozó azonosítót, valamint az előbb vázolt, az adatokhoz való hozzáférést korlátozó berendezést foglal magában.

Adatátviteli rendszer, amelynek az adatokhoz hozzáférési jogosultság készleteket társító eszköze van, ahol minden egyes hozzáférési jogosultság készlet legalább egy félhez van hozzárendelve, továbbá a felekhez tartozó hozzáférési jogosultság készleteket és azonosítókat az adatokon belül tároló eszköze, valamint az adatokat is magában foglaló bitfolyamot továbbító eszköze van.

Adatrögzítő eljárás és berendezés fájlok és könyvtárak alakjában tárolt adatok fa struktúrájának rögzítésére, amelyben minden egyes fájl és könyvtár vonatkozásában egy ahhoz tartozó azonosítót tárolunk el, annak a könyvtárnak az azonosítóját, amely az adatok fa struktúrája szerint a fájlt vagy könyvtárat közvetlenül megelőzi. A berendezésnek az adatok fa szerkezetében a fájlt vagy könyvtárat közvetlenül megelőző könyvtár azonosítóját minden egyes fájl és könyvtár vonatkozásában eltároló eszköze van.

Eljárás adatblokkok mozgatására memória lapok között, újabb adatok memóriában történő eltárolásának az elősegítésére, ahol a memória több lapot tartalmaz, és a memória egy lapját érvényes adatot tartalmazó legalább egy érvényes blokkot magában foglaló forrás lapként tekintünk, továbbá érvénytelen adatot tartalmazó legalább egy érvénytelen blokkot, és a memória legalább egy érvénytelen blokkot, és a memória legalább egy lapját átvivő lapként tekintünk, és a forrás lapról az egy vagy mindegyik érvényes blokkot továbbítjuk az átvivő lapra, ahol legalább egy ilyen blokknak az átvivő lapon a forrás lapon elfoglalt helyzetétől különböző pozíciója van, és a forrás lapot töröljük.

Eljárás adatok eltárolására memóriában, ahol a memória több lapot tartalmaz, és a memória egy lapja érvényes adatot tartalmazó legalább egy érvényes blokkot, érvénytelen adatot tartalmazó legalább egy érvénytelen blokkot magában foglaló forrás lapot tartalmaz, és a memória legalább egy lapját kiinduláskor kizárólag szabad blokkokat tartalmazó átvivő lapként tekintjük, és a legalább egy érvényes blokkot a forrás lapról az átvivő lapra másoljuk, ahol legalább egy ilyen blokk az átvivő lapon a forrás lapon elfoglalt helyzetétől különböző helyzetű lesz; a forrás lapot töröljük; és az adatokat az átvivő lapon tároljuk.

Berendezés adatblokkok mozgatására memória lapok között, új adat memóriában történő eltárolásának az elősegítésére, memória lapokkal, és a memória egy lapja érvényes adatot tartalmazó legalább egy érvényes blokkot, és érvénytelen adatot tartalmazó legalább egy érvénytelen blokkot magában foglaló forrás lapként van kijelölve, és a memória legalább egy lapja átvivő lapként van kijelölve, a forrás lapról a legalább egy érvényes blokkot az átvivő lapra másoló eszközt tartalmaz, ahol a legalább egy ilyen blokk átvivő lapon elfoglalt helyzete eltér a blokk forrás lapon elfoglalt pozíciójától, valamint a forrás lapot törölő eszközzel rendelkezik.

Vevő és dekódoló berendezés, amely a fenti berendezést tartalmazza.

(5. ábra)



KÖZZÉTÉTELI PÉLDÁNY

P 0 1 0 4 3 6 3 1
0 1 0 4 3 6 3 1

Adathozzáférést korlátozó eljárás és berendezés, továbbá vevő és dekódoló berendezés, adatátviteli rendszer, a kettő kombinációja, továbbá adatrögzítő eljárás és berendezés, továbbá eljárás és berendezés adatblokkok mozgatására, továbbá eltárolására

A találmány általánosságban adatkezelésre vonatkozik, ezen belül pedig például egy vevő és dekódoló berendezésben történő adatkezelés alábbi szempontjaira:

- memóriában fájlok és mappák formájában tárolt fá struktúrájú adatszerkezet rögzítésére,
- új adatok memóriában történő eltárolásának a lehetővé tétele érdekében adatblokkok memória lapok közötti mozgatására; és
- egy vevő és dekódoló berendezésben az adatokhoz való hozzáférés korlátozására.

A találmány tárgya ezen belül egyrészt egy adathozzáférést korlátozó eljárás memóriával rendelkező vevő és dekódoló berendezésben, másrészt egy adathozzáférést korlátozó eljárás digitális műsorszóró rendszerben sugárzott adatokhoz, harmadrészt egy adathozzáférést korlátozó berendezés, vevő és dekódoló berendezés memóriájában tárolt adatokhoz, negyedrészt egy vevő és dekódoló berendezés, amely adatokat tároló memóriát tartalmaz, ötödrészt egy adatátviteli rendszer, hatodrészt a fent említett vevő és dekódoló berendezés és adatátviteli rendszer kombinációja, hetedrészt adatrögzítő eljárás fájlok és könyvtárak alakjában tárolt adatok fá struktúrájának rögzítésére, nyolcadrészt fájlok és könyvtárak alakjában tárolt adatok fá struktúráját rögzítő adatrögzítő berendezés, kilencedrészt egy ilyen fenti berendezést tartalmazó vevő és dekódoló berendezés, tizedrészt egy eljárás adatblokkok mozgatására memória lapok között, újabb adatok memóriában történő eltárolásának a biztosítására, tizenegyedrészt egy eljárás adatok eltárolására memóriában, ahol a memória több lapot tartalmaz, és a memória egy lapja érvényes adatot tartalmazó legalább egy érvényes blokkot, érvénytelen adatot tartalmazó legalább egy érvénytelen blokkot magában foglaló forrás lapot tartalmaz, és a memória legalább egy lapját kiinduláskor kizárólag szabad blokkokat tartalmazó átvivő lapként tekintjük, tizenkettedrészt egy berendezés adatblokkok mozgatására memória lapok között, új adat memóriában történő eltárolásának az elősegítésére, mely berendezés memóriája lapokat tartalmaz, és a memória egy lapja érvényes adatot tartalmazó legalább egy érvényes blokkot, és érvénytelen adatot tartalmazó legalább egy érvénytelen blokkot magában foglaló forrás lapként van kijelölve, és a memória legalább egy lapja átvivő lapként van kijelölve, és tizenharmadrészt a fenti berendezést magában foglaló vevő és dekódoló berendezés.

A "vevő és dekódoló berendezés" vagy "dekóder" kifejezés akár kódolt, akár kódolatlan jelek vételére alkalmas, például televízió és/vagy rádió jelek vételére alkalmas vevőt jelenthet, ahol a jeleket sugárzás útján vagy más, ismert módon továbbítják. Egy ilyen dekóder lehetséges kiviteli alakjai kialakíthatók úgy, hogy a dekódoló berendezést egybeépítjük a vevővel, az utóbbi által vett jelek dekódolására, például egy set-top box-ban, vagy egy dekóder egy tőle fizikailag különálló vevővel kombinálva működik, vagy a dekódert más készülékekkel, például egy Web böngészővel, egy videomagnetofonnal, vagy egy televízió készülékkel egybe is integrálhatjuk.

Műsorszóró digitális televízió rendszerben a vett jeleket egy vevő és dekódoló berendezésbe, onnan pedig egy televízió készülékbe továbbítják. A "digitális televízió rendszer" leírásunkban bármilyen műholdas, földi sugárzású, kábeles vagy egyéb rendszert magában foglal. A vevő és dekódoló berendezést általában kézi távvezérlővel vezérlik, amely egy megfelelő interfészen keresztül kommunikál a vevő és dekódoló berendezéssel. A vevő és dekódoló berendezést a beérkező bitsorozat feldolgozására használják, és olyan alkalmazás modulokat tartalmaz, amelyek segítségével számos vezérlési és egyéb feladatot lát el.

Az "MPEG" rövidítés egy adatátviteli szabványt jelent, amelyet a Nemzetközi Szabvány Szervezet (ISO, International Standard Organization) "Motion Pictures Expert Group" elnevezésű munkacsoportja fejlesztett ki, és az ő nevéhez fűződik elsősorban de, nem kizárólagosan az MPEG-2 jelzésű szabvány kidolgozása is, amelyet kimondottan a digitális televízió alkalmazások céljára hoztak létre, és az "ISO 13818-1, ISO 13818-2, ISO 13818-3 és ISO 13818-4" jelzésű dokumentumokban rögzítettek. Találmányunk jelen leírásában az MPEG-2 elnevezés alatt az említett szabvány összes változatait, módosításait vagy továbbfejlesztéseit is értjük, melyek alkalmasak a digitális adatátvitel területén valamilyen felhasználásra.

Egy ilyen fent vázolt vevő és dekódoló berendezés számos hozzá kapcsolódó eszközt és egységet tartalmazhat, például egy kártyaolvasót, amellyel a felhasználó egy jogosultságot igazoló kártyát használhat annak igazolására, hogy mely szolgáltatások felhasználására jogosult, továbbá egy kézi távvezérlőt, televízió kijelző egységet, valamint egy második kártyaolvasót, amellyel bankkártyákat használhat a felhasználó, például házi banki szolgáltatások és műveletek elvégzésére. A berendezés természetesen több csatlakozóval lehet ellátva, és például az Internethez történő kapcsolatteremtést és kapcsolattartást biztosító, a házi banki műveleteket lebonyolító modemmel is ellátható.

Az általunk jelenleg általánosan elterjedt, ismert megoldást vázoló 1. ábrán megfigyelhető, hogy egy vevő és dekódoló berendezés központ 50 processzort és azzal ismert módon kapcsolódó 54, 56, 58 memóriát foglal magában. Az 54, 56, 58 memória közvetlenül csatlakoztatható a központi 50 processzorhoz, vagy mint az 1. ábrán látható azzal 52 buszon keresztül állhat kapcsolatban.

Különböző számítógép memória típusokat ismerünk. A különböző memória típusok közötti egyik lényeges különbség a felejtő és a nem felejtő memóriák között található, hiszen a felejtő memóriák tartalmukat csak addig tartják meg, amíg a táplálásukra szolgáló energia is fennáll, és amint táplálásuk megszűnik, tartalmukat rögtön és visszavonhatatlanul elvesztik, míg a nem felejtő memóriák tartalma meghatározatlan ideig megmarad abban az esetben is, ha táplálásukat megszüntetjük. Egy másik lényeges különbség áll fenn az írható memóriák és a csak olvasható memóriák között.

A felejtő memóriákat általában RAM néven ismerjük, a nem felejtő memóriáknak számos típusa van. A RAM általában írható, míg a csak olvasható memóriák leginkább ROM néven ismertek. Ez az utóbbi megkülönböztetés nem feltétlenül végleges. Minden memóriát élete folyamán legalább egyszer írunk kell ahhoz, hogy valamilyen tartalma legyen, de egyes ROM típusú memóriák tartalma, ha nehezen is, de módosítható. Így léteznek olyan memória típusok, mint a PROM, ami nem más, mint programozható (azaz írható) ROM, EEPROM, ami elektromosan törölhető programozható ROM, valamint a Flash ROM-ok.

A különböző memória típusok különböző jellemzőkkel bírnak, így eltérő olvasási idővel és természetesen különböző előállítási költséggel, így igen gyakran előfordul, hogy egyetlen számítógép rendszeren belül a célszerűség miatt különböző memória típusokat kell kombinálnunk, amint az az 1. ábrán megfigyelhető.

A 2. ábrán az adatok egy memória részben történő szervezésének ismert és szokásos módját vázoljuk. Az adatok jól ismert módon fa struktúrájúak, és az ábrán a könyvtárakat egy-egy körbeírt D jellel, az egyes fájlokat pedig egy-egy négyszögbe írt F jellel jelöltük. Mint arra még részletesebben is kitérünk, minden D könyvtár egy cím vagy mutató listát tartalmaz, a vele valamilyen módon társított összes F fájlhoz és/vagy D könyvtárhoz, amely az adatok fa struktúrájú nyomon követését teszi lehetővé.

Az adatszervezés fa struktúráját a könnyebb megértés érdekében összefüggésbe hozhatjuk az általánosan ismert családfákkal, ahol "apák" és "fiúk" szerepelnek. Minden egyes "apa" vagyis minden egyes D könyvtár egy-egy címet tartalmaz mindegyik "fiára" vonatkozóan,

vagyis azokra az F fájlokra és D könyvtárakra vonatkozóan, amelyek ezzel a könyvtárral társítottak.

Például a 2. ábrán nézzük meg D6 könyvtárat, amely ennek megfelelően egy címet tartalmaz fiára, a F6 fájlra vonatkozóan. A D5 könyvtárnak több fia van, így ennek megfelelően több címet is tartalmaz, egyet az F5 fájl, egyet pedig a D6 könyvtár vonatkozásában. A D2 könyvtár mindhárom "fia" azaz a belőle leszármazó D3, D4, D7 könyvtár vonatkozásában tartalmaz egy-egy címet.

A D könyvtárak és F fájlok fa struktúrájú elrendezését a központi 50 processzor folyamatosan újraszervezi, például, ha egy F fájl egy másik D könyvtárba mozgatunk, akkor értelemszerűen annak a címe is módosul. Ha egy fájl, például az F5 fájl a D5 könyvtárból átmozgatjuk a D6 könyvtárba, így az adatszerkezet fa struktúrája is módosul, akkor azt a D könyvtárat, amelyhez az F fájl azt megelőzően közvetlenül tartozott, valamint azt a könyvtárat, amelybe az átmozgatott F fájl most tartozik, vagyis jelen esetben a D6 könyvtárat az F5 fájl megfelelő helyének megtalálása céljából módosítanunk kell.

Ennek megfelelően az adatok fa szerkezetének egy F fájl mozgatása esetén történő megváltoztatását egy kétlépcsős műveletben tudjuk megoldani. A 3(a). ábrán, amely még mindig az ismert megoldást mutatja be, egy memóriakötetben az adatok fa struktúrájának megváltoztatására irányuló első eljárás lépéseit vázoltuk. 1. lépésben a D6 könyvtárba egy címet írunk az F5 fájlra vonatkozóan, 2. lépésben pedig az F5 fájl címét töröljük a D5 könyvtárból. Az ugyancsak a technika állásához tartozó eljárást vázoló 3(b). ábrán egy második folyamatot vázoltunk, amellyel egy memóriakötetben az adatok fa struktúráját módosítjuk. A D5 könyvtárból töröljük az F5 fájl címét, a 2. lépésben pedig az F5 fájl címét beírjuk a D6 könyvtárba.

Mindkét folyamat esetén, ha az 1. és 2. lépés között az operációs rendszer valamilyen összeomlana, akkor az adatok fa struktúrája nem kerülne pontosan rögzítésre. Az első folyamat során az adat struktúra úgy nézne ki, mint a 3(a). ábra (ii) szakaszában, amikor az F5 fájl címe mind a D5, mind pedig a D6 könyvtárba be lenne írva. Más szavakkal ez annyit jelent, hogy az F5 fajlnak két "apja" lenne. A második eljárás során az adat struktúra a 3(b) ábra (ii) esetnek megfelelően kerülne rögzítésre, azaz az F5 fájl sem a D5 könyvtárhoz, sem a D6 könyvtárhoz nem tartozna. Ez szaknyelven azt jelenti, hogy az F5 fájl úgynevezett "fattyú" fájl, és a különböző hibaellenőrző és javító programok az ilyen fájlokat leggyakrabban törléssel javítják.

A vevő és dekódoló berendezés memóriájában tárolt adatokhoz való hozzáférést számos fél kérheti illetve kezdeményezheti. Egy ilyen félnek minősül a szóban forgó adat szerzője, aki például azért szeretne hozzáférni az adathoz, hogy egy abban felfedezett hibát kijavítson, vagy az adatot annak egy felfrissített változatával helyettesítse, és így tovább. Egy másik ilyen fél lehet egy, az adatot használó interaktív alkalmazás szolgáltatója. Míg egy ilyen szolgáltató célszerűen úgy férhet hozzá az adathoz, hogy azt olvasni tudja, és ezzel az interaktív alkalmazás is fel tudja használni az adatot, az adat szerzője megtilthatja a szolgáltató számára, hogy az az adatot saját adatával felülírhasssa. A szolgáltató is megtilthatja például az adathoz való bármilyen hozzáférést a vevő és dekódoló berendezés tulajdonosának vagy gyártójának, fenntartva ezzel az adatok titkosságát a nevezettekkel szemben.

Találmányunk egyik célja egy vevő és dekódoló berendezésben az adatokhoz való hozzáférés egyszerű és eredményes korlátozása.

A kitűzött feladatot egyrészt egy adathozzáférést korlátozó eljárással oldottuk meg, memóriával rendelkező vevő és dekódoló berendezésben, amelynek során javaslatunk értelmében az adatokhoz hozzáférési jogosultság készleteket rendelünk hozzá, ahol minden egyes hozzáférési jogosultság készletet legalább egy félhez társítunk; a vevő és dekódoló berendezés memóriájában az adatokat, a hozzáférési jogosultság készleteket valamint minden egyes félhez tartozó azonosítót eltárolunk; az adatokhoz hozzáférést kérelmező félhez tartozó azonosítót összehasonlítjuk a memóriában tárolt legalább egy azonosítóval; és a vevő és dekódoló berendezés memóriájában hozzárendelt hozzáférési jogosultság készletet a fél rendelkezésére bocsátjuk. A fenti javasolt eljárás egy előnyös foganatosítási módja értelmében a hozzáférés jogosultság készleteket az adatok élőfejében tároljuk el.

Ily módon azzal, hogy az adatoknak még a vevő és dekódoló berendezés memóriájában történő eltárolását megelőzően különböző felek számára különböző hozzáférési jogosultságokat hozzáférési jogosultság készleteket rendelünk vagy társítunk, a vevő és dekódoló berendezésben az adatokat biztonságosan és megfelelő módon tudjuk szétválogatni.

Előnyös a javasolt eljárás olyan foganatosítási módja, amelynek során a felekre vonatkozó azonosítókat az adatok élőfejében tároljuk el. Ezzel az adatokhoz tartozó hozzáférési jogosultságokat megfelelő és könnyen hozzáférhető helyen tudjuk eltárolni.

Előnyös továbbá javaslatunk értelmében, ha az adatokat egy adatátviteli rendszer által továbbított bitfolyamból töltjük le, és az egyes felekhez tartozó hozzáférési jogosultság készleteket és azonosítókat a adatátviteli rendszerben az adatokon belül tároljuk.



A kitűzött feladatot másrészt egy digitális műsorszóró rendszerben sugárzott adatokhoz a hozzáférést korlátozó eljárással oldottuk meg, amelynek során találmányunk értelmében az adó rendszerben az adatokhoz hozzáférési jogosultság készleteket rendelünk hozzá, melyek mindegyikét legalább egy félhez hozzárendeljük; az egyes felekhez tartozó hozzáférési jogosultság készleteket és azonosítókat az adatokban eltároljuk; és az adatokat továbbítjuk; míg egy memóriával rendelkező vevő és dekódoló berendezésben a továbbított adatokat a vevő és dekódoló berendezés memóriájába letöltjük és eltároljuk; az adatokhoz hozzáférést kérelmező fél azonosítóját a memóriában tárolt azonosítókkal összehasonlítjuk; és a fél részére a vevő és dekódoló berendezés memóriájában hozzárendelt hozzáférési jogosultság készletet megadjuk.

A fenti eljárás egy előnyös foganatosítási módja értelmében az adatokat egy digitális bit-folyamban továbbítjuk.

Ugyancsak előnyös a javaslat értelmében, ha további hozzáférési jogosultság készletet rendelünk legalább egy olyan félhez hozzá, akinek az azonosítója nincs a vevő és dekódoló berendezés memóriájában eltárolva, és az adatokhoz hozzáférést kérelmező ilyen felet az így hozzárendelt további hozzáférési jogosultság készlettel látjuk el. Ezzel egy számunkra ismeretlen, vagy előnyben nem részesülő felet meg tudunk gátolni a hozzáférési jogosultságok megfelelő készletének a kiválasztása révén attól, hogy az adatokhoz hozzáférhessen.

A javasolt eljárás egy további előnyös foganatosítási módja értelmében egy meghatározott hozzáférési jogosultság készletet csupán egyetlen félhez, előnyösen az adatok szerzőjéhez rendelünk hozzá. Így az adatok szerzője számára egy különálló hozzáférési jogosultság készletet rendszeresíthetünk, amelynek segítségével a szerző az adatokhoz teljes terjedelemben, korlátozás nélkül hozzáférhet.

Ugyancsak előnyös a javaslat értelmében, ha egy meghatározott hozzáférési jogosultság készletet egy felekből álló csoporthoz rendelünk hozzá, és a csoport minden egyes tagjához tartozó azonosítót tároljuk a vevő és dekódoló berendezés memóriájában. Azáltal, hogy egy hozzáférési jogosultság készletet egy felekből álló csoporthoz rendelünk hozzá, csökkenteni tudjuk a kiosztandó, illetve hozzárendelendő hozzáférési jogosultság készletek számát.

Ugyancsak előnyös a javasolt eljárás olyan foganatosítási módja, amelynek során egy hozzáférési jogosultság készletet használunk annak a megállapítására, hogy egy fél ki van-e zárva az adatok olvasásából.

Ugyancsak előnyös a javasolt eljárás olyan foganatosítási módja, amelynek során egy hozzáférési jogosultság készletet használunk annak a megállapítására, hogy egy fél ki van-e zárva az adatok felülírásából.

A javasolt eljárás egy további előnyös foganatosítási módja értelmében az adatokat a vevő és dekódoló berendezés memóriájában fájlok és könyvtárak alakjában tároljuk.

Ez utóbbi esetben előnyös, ha a fájlok és könyvtárak fa struktúráját a vevő és dekódoló berendezés memóriájában rögzítjük.

Ez utóbbi esetben előnyös továbbá, ha az adat fa struktúrában egy adott esetben egy fájl vagy könyvtárat közvetlenül megelőző könyvtár azonosítóját a fájlal és könyvtárral társítva eltároljuk. Kifejezetten előnyös az utóbbi esetben, ha az azonosítót a fájl vagy könyvtár előfeijében tároljuk el.

A javasolt eljárás egy további előnyös foganatosítási módja értelmében az adatokat a vevő és dekódoló berendezés Flash memóriájában tároljuk el.

A kitűzött feladatot harmadrészt egy vevő és dekódoló berendezés memóriájában tárolt adatokhoz való adathozzáférést korlátozó berendezéssel oldottuk meg, amelynél a javaslat értelmében az adatokkal hozzáférési jogosultsági készleteket társítunk és minden egyes hozzáférési jogosultság készletet legalább egy félhez hozzárendelünk, és minden egyes félre vonatkozó azonosítót a vevő és dekódoló berendezésben tárolunk, és a berendezésnek az adatokhoz hozzáférést kérelmező fél azonosítóját a vevő és dekódoló berendezés memóriájában tárolt azonosítókkal összehasonlító eszköze, valamint a fél számára a hozzárendelt hozzáférési jogosultság készletet a vevő és dekódoló berendezés memóriájában lévő, hozzárendelt hozzáférési jogosultság készletet rendelkezésre bocsátó eszköze van.

A kitűzött feladatot negyedrészt egy vevő és dekódoló berendezéssel oldottuk meg, amely adatokat tároló memóriát tartalmaz, és a találmány értelmében az adatokhoz hozzárendelt hozzáférési jogosultság készleteket, minden egyes félhez tartozó azonosítót, valamint az előbb vázolt, az adatokhoz való hozzáférést korlátozó berendezést foglal magában.

A vevő és dekódoló berendezés ezen túlmenően az egyes felek azonosítóit tároló biztonsági modullal is ellátható.

A vevő és dekódoló berendezés ezen túlmenően az adatokat, a hozzáférési jogosultság készleteket, az azonosítókat magában foglaló bitfolyam vételére alkalmas vevőt, valamint az adatok, a hozzáférési jogosultság készletek és az azonosítók memóriába való letöltésére

alkalmas egyéb eszközöket, például demultiplexert és bitsordekódolót tartalmazhat. A vevő és dekódoló berendezés előnyösen MPEG táblázatok letöltésére alkalmas módon van kialakítva.

A kitűzött feladatot ötödrészt egy adatátviteli rendszerrel oldottuk meg, amelynek a találmány értelmében az adatokhoz hozzáférési jogosultság készleteket társító eszköze van, ahol minden egyes hozzáférési jogosultság készlet legalább egy félhez van hozzárendelve, továbbá a felekhez tartozó hozzáférési jogosultság készleteket és azonosítókat az adatokon belül tároló eszköze, valamint az adatokat is magában foglaló bitfolyamot továbbító eszköze van. A hozzárendelést fogatosító eszköz úgy is megvalósítható, hogy egy további hozzáférési jogosultság készletet társítson azokhoz a felekhez, melyek azonosítói nincsenek a vevő és dekódoló berendezés memóriájában eltárolva. A hozzárendelő eszköz ezen túlmenően egy hozzáférési jogosultság készletet egy felekből álló csoporthoz hozzárendelő eszközként is kialakítható, ahol a csoport egyes tagjainak az azonosítói az adatokban vannak eltárolva.

A puffertár kezelő rendszer különböző funkciói megvalósíthatók hardveresen, például dedikált integrált áramkörrel, ami felgyorsítja a műveleteket. A funkciók legalább némelyikét azonban előnyös, ha szoftveresen valósítjuk meg, például azokon a feldolgozó eszközökön futtatjuk, amelyiken az alkalmazások is futnak, ezzel nagyobb rugalmasságot biztosíthatunk, kevesebb hardver alkatrészre van szükség, és a rendszer is lényegesen könnyebben frissíthető a mindenkori legújabb műszaki szintre.

A kitűzött feladatot hatodrészt a fent említett vevő és dekódoló berendezés és adatátviteli rendszer kombinációjával oldottuk meg.

A találmánnyal célunk ezen kívül egy olyan biztonságos rendszer kialakítása, amellyel a fa adat struktúrát tudjuk nyomon követni egy memória részben, úgy, hogy a fájl műveletek során az adat struktúra frissítéséhez szükséges időt jelentősen csökkenteni tudjuk, mindamelllett a nyomon követés megbízhatóságát kellő szinten tudjuk tartani.

A kitűzött feladatot hetedrészt fájlok és könyvtárak alakjában tárolt adatok fa struktúrájának a rögzítésére vonatkozó eljárással oldottuk meg, amelynek során újszerű módon minden egyes fájl és könyvtár vonatkozásában egy ahhoz tartozó azonosítót tárolunk el, annak a könyvtárnak az azonosítóját, amely az adatok fa struktúrája szerint a fájlt vagy könyvtárat közvetlenül megelőzi. Ezzel a megoldással az adatok fa struktúráját gyorsabban tudjuk rögzíteni, mivel mindegyik fájl vagy könyvtár az őt közvetlenül megelőző könyvtár azono-

sítóját tartalmazza, nem pedig azoknak a könyvtáraknak és fájloknak az azonosítóit, amelyek hagyományos módon a fa struktúrában a könyvtárat követik. Így azoknak a könyvtáraknak vagy fájloknak a száma, amelyeket módosítanunk kell, ha az adat fa struktúra módosul, lényegesen kevesebb lesz, mint a hagyományos megoldások esetében; csupán az egy "fiúban" tárolt "apa" azonosítóját kell egy egylépcsős műveletben megváltoztatnunk, ha a "fiú" címe megváltozik. Ez különösen előnyös olyan memória részekben, ahol az adatok írása erősen időigényes, például a Flash memóriákban.

Mivel a fa struktúra módosítása egy egylépcsős művelet, ha egy fájlt vagy könyvtárat átmozgatunk, a kétlépcsős műveletekhez tartozó problémákat egyszerűen el tudjuk kerülni.

E javasolt eljárás egy előnyös foganatosítási módja értelmében az azonosító a könyvtárhoz hozzárendelt egyedi kódot tartalmaz. Így az adatok fa struktúráját igen gyorsan meg tudjuk változtatni, az új "apához" társított kód olvasása révén, amikor egy fájlt vagy könyvtárat mozgatunk és ennek a kódnak a fájlban vagy könyvtárban történő tárolása révén.

Ugyancsak előnyös a javaslat értelmében, ha az azonosítót a fájl vagy könyvtár élőfejében tároljuk el. Ezzel a fájl vagy könyvtár elhelyezkedésének könnyen hozzáférhető információját tudjuk biztosítani az adatok fa struktúrájában.

Ugyancsak előnyös a javasolt eljárás olyan foganatosítási módja, amelynek során az adatok legalább egy részét, előnyösen egészét Flash memóriában tároljuk el. Ennek segítségével az adatokat két vagy több memória részen, például egy Flash memória részen, valamint egy ROM memória részen is "szét tudjuk teríteni". Az adatokat az előbb említett módon kizárólagosan a Flash memória részben is eltárolhatjuk.

Az előző foganatosítási mód szerint ugyancsak előnyös, ha a Flash memória részben tárolt adatok virtuális fa struktúráját előállítjuk és egy RAM memória részben tároljuk.

Ugyancsak előnyös a javasolt eljárás olyan foganatosítási módja, amelynek során a Flash memória részben tárolt fájl élőfejét a Flash memória rész dedikált blokkjában tároljuk.

Előnyös a javasolt eljárás olyan foganatosítási módja, amelynek során az eltárolt azonosítót egy fájl vagy könyvtár közvetlenül megelőző másik könyvtárba történő mozgatása során módosítjuk. Ezzel egy fájl mozgatása esetén az adat fa struktúráját igen gyorsan tudjuk módosítani.

Előnyös a javaslat értelmében, ha az adatokat egy vevő és dekódoló berendezésben tároljuk.

A kitűzött feladatot nyolcadrészt fájlok és könyvtárak alakjában tárolt adatok fa struktúráját rögzítő adatrögzítő berendezéssel oldottuk meg, amelynek a találmány értelmében az adatok fa szerkezetében a fájlt vagy könyvtárat közvetlenül megelőző könyvtár azonosítóját minden egyes fájl és könyvtár vonatkozásában eltároló eszköze van. A tárolóeszközt egy központi processzor is alkothatja, amelyet úgy állíthatunk be, hogy az azonosítót a fájl vagy könyvtár előfejeiben tárolja el.

A berendezést egy Flash memória rész és egy RAM memória rész kombinációjával is létrehozhatjuk. Ebben az esetben a berendezésnek a Flash memória részben tárolt adatok fa struktúrájának a RAM memória részben történő eltárolására alkalmas eszközei lehetnek, és a rögzítést végrehajtó eszköz arra alkalmas módon a központi processzor is lehet.

A berendezés ezen túlmenően egy fájlnak vagy könyvtárnak az őt közvetlenül megelőző másik könyvtárba történő mozgatása esetén a tárolt azonosítót megváltoztató eszköze lehet, amelyet ugyancsak megfelelő módon a központi processzorral valósíthatunk meg.

A kitűzött feladatot kilencedrészt egy ilyen fenti berendezést tartalmazó vevő és dekódoló berendezéssel oldottuk meg.

A Flash memória leginkább ROM-hoz hasonlít annyiban, hogy nem felejtő tár. Éppen ezért általánosságban ROM-hoz hasonló módon használhatjuk, vagyis olvashatjuk, de nem írhatjuk. A Flash memóriát azonban, még ha kicsit bonyolult módon is, írni is tudjuk. A Flash memória általános esetben lapokra van felosztva, és egy lap jellemzően néhány kilobájt nagyságú. Ha a Flash memória egy lapján egy vagy több bit értékét kívánjuk megváltoztatni, ezt csak úgy tehetjük meg, ha a teljes lapot töröljük. Pontosabban tehát ahhoz, hogy a Flash memória egy blokkját újra felhasználjuk, egy teljes lapot törölnünk kell, hogy abba a blokkba, vagyis arra a lapra új adatot tudjuk írni.

A Flash memóriában az információ célszerű nagyságú blokkokra szervezve található. Egy blokk adatokat, például állandó vagy félig állandó információ táblázatokat, vagy egy programot vagy egy szubrutint tartalmazhat. A blokk méretet általános esetben akkorára választjuk, hogy kisebb legyen, mint a lap nagysága (ha egy blokk nagyobb, mint egy lap, úgy nem kerülhető el, hogy olyan alblokkokra bontsuk, amelyek mindenképpen kisebbek, mint a lap nagysága).

Általában a Flash memória frissítése során a benne lévő információ egy részét meg szeretnénk tartani. Ehhez azonban az szükséges, hogy a frissítendő, vagy felülírandó lapot, pontosabban annak tartalmát beolvassuk a RAM-ba, hogy ott létrehozzuk a lap tükörképét, ezt

a RAM-ban létrehozott tükörképet felfrissítsük, amelynek során azt az új információt is beleírjuk, amelyet szeretnénk, hogy a Flash memória adott lapján megjelenjen. Ezzel egyidejűleg törölhetjük mindazt az információt a lapon, amelyre már nincs többé szükségünk. A felfrissített, új információval ellátott, régi, felesleges információtól megszabadított tükörképet írjuk azután vissza a Flash memóriába.

Annak érdekében, hogy lehetővé tegyünk, hogy a Flash memória blokkjaiban valamilyen fajta név vagy leíró használatával információt tudjunk keresni, valamilyen típusú blokk lokalizáló vagy címző adatstruktúrát kell fenntartanunk. Ahhoz, hogy engedélyezni tudjuk a blokkokban a keresést, ismert megoldás, hogy a Flash memória minden egyes lapjához egy különálló blokk lokalizáló adatstruktúrát tartunk karban. Ezt a blokk lokalizáló adatstruktúrát legalább részben Flash memórián kívül, valamilyen külső memóriában, például egy EEPROM-ban tárolhatjuk.

Találmányunkkal kapcsolatosan ismertük fel azt is, hogy az ilyen blokk lokalizáló adatstruktúra esetén a Flash memória tartalmának bármilyen módosítása, módosulása esetén a Flash memóriát fel kell frissítenünk. Egy új blokk írásához a Flash memóriát frissítenünk kell, egy blokk törléséhez hasonlóképpen frissítésre van szükségünk. Jóllehet arra nincs szükség, hogy a szóban forgó blokkot fizikailag is töröljük, a blokk lokalizáló adatstruktúrát úgy kell frissítenünk, hogy az azt jelezze, hogy a szóban forgó blokk többé már nem érvényes.

A Flash memória frissítésére alkalmas ilyen jellegű rendszer számos hátránnyal rendelkezik. Egyrészt, a Flash memória egy teljes lapját be kell másolnunk a RAM memóriába, hogy lehetővé tegyünk, hogy új adatokat tároljunk el aztán a Flash memóriában. Ezért a RAM memóriában akkora puffertárral kell rendelkezünk, amelynek mérete ugyanakkora, mint a Flash memória egy lapjának a mérete. Másodszor igen lényeges, hogy egy EEPROM memória részünk legyen, amelyben a Flash memóriában tárolt blokkok helyét és állapotát azonosító blokk lokalizáló adatstruktúrát el tudjuk tárolni.

A találmánnyal célunk továbbá egy olyan rendszer létrehozása, melynek segítségével úgy tudjuk egy Flash memória tartalmát frissíteni, hogy ehhez nincs szükségünk ROM vagy RAM memóriára sem.

A kitűzött feladatot tizedrészt adatblokkok memória lapok közötti átvitelére vonatkozó eljárással oldottuk meg, újabb adatok memóriában történő eltárolásának a biztosítására, ahol a találmány értelmében a memória több lapot tartalmaz, és a memória egy lapját érvényes

adatot tartalmazó legalább egy érvényes blokkot magában foglaló forrás lapként tekintünk, továbbá érvénytelen adatot tartalmazó legalább egy érvénytelen blokkot, és a memória legalább egy érvénytelen blokkot, és a memória legalább egy lapját átvivő lapként tekintünk, és az eljárás során a forrás lapról az egy vagy mindegyik érvényes blokkot továbbítjuk az átvivő lapra, ahol legalább egy ilyen blokknak az átvivő lapon a forrás lapon elfoglalt helyzetétől különböző pozíciója van, és a forrás lapot töröljük.

A javasolt eljárás egy előnyös foganatosítási módja értelmében az adatot ezt követően az átvivő lapon a másolt érvényes blokkok szomszédságában tároljuk el. A legalább egy érvényes blokk helyének megváltoztatása révén az érvényes blokkokat össze tudjuk rendezni, például az átvivő lap egyik végén, például a kezdetén. Ezzel elegendő helyet tudunk biztosítani az átvivő lap maradék részén a friss adatok eltárolására.

A kitűzött feladatot tizenegyedrészt adatoknak memóriában történő eltárolására vonatkozó eljárással oldottuk meg, ahol a memória több lapot tartalmaz, és a memória egy lapja érvényes adatot tartalmazó legalább egy érvényes blokkot, érvénytelen adatot tartalmazó legalább egy érvénytelen blokkot magában foglaló forrás lapot tartalmaz, és a memória legalább egy lapját kiinduláskor kizárólag szabad blokkokat tartalmazó átvivő lapként tekintjük, és találmányunk értelmében az eljárás során a legalább egy érvényes blokkot a forrás lapról az átvivő lapra másoljuk, ahol legalább egy ilyen blokk az átvivő lapon a forrás lapon elfoglalt helyzetétől különböző helyzetű lesz; a forrás lapot töröljük; és az adatokat az átvivő lapon tároljuk.

A javasolt eljárás egy előnyös foganatosítási módja értelmében a legalább egy érvényes blokkot úgy másoljuk az átvivő lapra, hogy ezzel friss adatok vételéhez a lehető legnagyobb töredezetlen memória területet hozzuk létre az átvivő lapon.

A javasolt eljárás egy további előnyös foganatosítási módja értelmében a törölt forrás lapot előnyösen közvetlenül annak törlését követően új átvivő lapként tekintjük.

A javasolt eljárás egy további előnyös foganatosítási módja értelmében egy lapot a lap érvénytelen blokkjainak együttes nagysága függvényében tekintünk forrás lapként.

Ez utóbbi esetben előnyös a találmány értelmében, ha a forrás lap legalább egy érvénytelen blokkot tartalmazó lap, ahol a legalább egy érvénytelen blokk együttes nagysága az új adatok méretével azonos, vagy annál nagyobb. Ezáltal biztosíthatjuk, hogy elegendő hely maradjon az adatok eltárolására az átvivő lapon, miután az érvényes blokkoknak az átvivő lapra másolását befejeztük.

A javasolt eljárás további előnyös foganatosítási módja értelmében a blokkok változó méretű blokkok. Ez számos előnyt biztosít a rögzített nagyságú blokkok alkalmazásával szemben. A rögzített nagyságú blokkok általában 5 kB blokkok. Ha egy 3 kB adatot szeretnénk egy ekkora rögzített méretű blokkban eltárolni, úgy 2 kB memóriát tulajdonképpen elvesztegettünk. Ha változó méretű blokkokat használunk, a blokk nagyságát kizárólag az abban tárolt adat nagysága fogja meghatározni, ezzel hatékonyan elő tudjuk segíteni a megfelelő adattárolási kapacitását kihasználását a memóriában. Ha engedélyezzük, hogy az érvényes változó méretű blokkokat máshová átmozgathassuk, amikor egy lapot frissítünk, akkor a blokkokat az átvivő lapon úgy tudjuk újra rendezni, hogy a lap szabad kapacitását a lap egyetlen nem használt területén alakítsuk ki.

Ugyancsak előnyös a javaslat értelmében, ha a memória adat szabad írását gátló memóriát, előnyösen Flash memória részt foglal magában.

Ez utóbbi esetben előnyös, ha az érvényes blokkot a blokkban tárolt egybites jelző értékének megváltoztatásával módosítjuk érvénytelen blokká.

Ez utóbbi esetben előnyös, ha az egybites jelzőt a blokk élőfejében tároljuk. Ha a bit értékét egyről nullára módosítjuk, azaz a bitet töröljük, a Flash memóriában tárolt blokk állapotának a módosítását úgy hajthatjuk végre, hogy ehhez nem kell az egész lapot újraírunk.

A javasolt eljárás egy további előnyös foganatosítási módja értelmében a memória egy vevő és dekódoló berendezés memória részét tartalmazza.

Ez utóbbi esetben előnyös, ha az adatok bitfolyamból letöltött adatok.

Ugyancsak előnyös, ha az adatok MPEG táblázatok alakjában vannak formázva.

A kitűzött feladatot tizenkettedrészlet adatblokkok memória lapok közötti továbbítására alkalmas berendezéssel oldottuk meg, új adat memóriában történő eltárolásának az elősegítésére, mely berendezés memóriája lapokat tartalmaz, és a memória egy lapja érvényes adatot tartalmazó legalább egy érvényes blokkot, és érvénytelen adatot tartalmazó legalább egy érvénytelen blokkot magában foglaló forrás lapként van kijelölve, és a memória legalább egy lapjazz átvivő lapként van kijelölve és a találmány értelmében a forrás lapról a legalább egy érvényes blokkot az átvivő lapra másoló eszközt tartalmaz, ahol a legalább egy ilyen blokk átvivő lapon elfoglalt helyzete eltér a blokk forrás lapon elfoglalt pozíciójától, valamint a forrás lapot törölő eszközzel rendelkezik.

A kitűzött feladatot tizenharmadrészt a fenti berendezést magában foglaló vevő és dekódoló berendezéssel oldottuk meg.

A feltüntetett jellemzők bármilyen megfelelő kombinációban megvalósíthatók. A berendezés jellemzők eljárásoknál is felhasználhatók, és fordítva.

A találmányt az alábbiakban a csatolt rajz segítségével ismertetjük részletesebben, amelyen a javasolt eljárás példakénti fogantatási módját tüntettük fel. A rajzon az

1. ábra egy vevő és dekódoló berendezésen belül memória részek ismert elrendezését tünteti fel tömbvázlat szinten, a
2. ábrán egy memória részben tárolt adatok jelenlegi ismert műszaki szint szerinti fa struktúrájú szervezését vázoltuk, a
- 3(a). ábrán adatok egy memória részben adatok fa struktúrájának megváltoztatására vonatkozó ismert eljárás fő lépései láthatók, a
- 3(b). ábrán egy memória részben az adat fa struktúra módosítására szolgáló másik ismert eljárás vázlata látható, a
4. ábra egy digitális televízió rendszer felépítését mutatja tömbvázlat szinten, az
5. ábrán a 4. ábra szerinti digitális televízió berendezés vevő és dekódoló berendezésének elvi vázlatos felépítése látható, a
6. ábrán egy Flash memória egy lapjának tervezését, felosztását láthatjuk, a
7. ábra egy Flash memória egy lapján lévő blokk előfejét vázolja, a
8. ábra a Flash memóriában tárolt fájl előfej blokkjának tartalmát mutatja, a
9. ábra a Flash memóriában tárolt adat fa struktúrát mutat, a
10. ábrán egy előfej blokkban egy hozzáférési mód mező tartalma látható, a
11. ábra egy érvényes adatblokknak a Flash memória egyik lapjáról egy másik lapjára történő másolásának egy lehetséges fogantatási módját mutatja, a
12. ábrán adatblokkok másolására vonatkozó eljárás folyamatábrája látható, egy Flash memória egyik lapjáról egy másik lapjára történő másolás esetén, és a
13. ábra a 12. ábrán bemutatott eljárás általános hatását mutatja be.

A találmány szerinti 1 digitális televízió rendszer vázlata a 4. ábrán követhető nyomon. A találmány hagyományosan felépített 2 digitális televízió rendszert foglal magában, amely az ismert MPEG-2 tömörítő rendszert használja tömörített digitális jelek kibocsátásához és sugárzásához. Ezt kicsit részletezve, 3 tömörítő, amely szokásosan egy műsorszóró központban helyezkedik el, rendszerint videojel adatáram formájában digitális adatáramot kap. A 3 tömörítő 5 vonalon keresztül 4 multiplexerhez és bitsorkódolóhoz van csatlakoztatva.

A 4 multiplexer és bitsorkódoló multiplexere további bemenőjelet kap, amelyekből egy vagy több szállítási adatáramot állít össze, és a tömörített digitális jelet a műsorszóró központban lévő 6 adóberendezéshez továbbítja további 7 vonalon keresztül, amely természetesen számos módon megvalósítható, beleértve a szokásos távközlési vezetékes kapcsolatokat is. A 6 adóberendezés elektromágneses jeleket továbbít 8 földi állomástól műholdra irányuló adatátviteli kapcsolat segítségével 9 transzponderhez, amely azt elektronikusan feldolgozza, és 10 műholdról földi állomás felé irányuló adatátviteli kapcsolaton át egy földi 12 vevőhöz továbbítja, mely 12 vevő általában egy végfelhasználó tulajdonában lévő, vagy általa bérelt parabola antenna. A 12 vevővel fogadott jelek szintén a végfelhasználó tulajdonában álló, vagy általa bérelt integrált 13 vevő és dekódoló berendezésbe kerülnek, amely a tömörített MPEG-2 jelet olyan televízió jellé alakítja vissza, amely minden további nélkül használható 14 televízió készülékben. Az adatok továbbítására természetesen másfajta szállítási csatornákat is használhatunk, például földi sugárzást, kábeles adatátvitelt, kombinált műholdas-kábeles összeköttetést, telefonhálózatot, és így tovább.

Egy sokcsatornás rendszerben a 4 multiplexer és bitsorkódoló a több egymással párhuzamos forrástól származó audio és video információt kezeli le, és a 6 adóval együttműködve az információt megfelelő számú csatornán kisugározza. Az audiovizuális információn túl az egyes, vagy akár az összes csatornában üzeneteket, vagy alkalmazásokat, vagy bármely másfajta digitális adatot is vezethetünk, amely a továbbított digitális audio és video információra ül rá.

A 4 multiplexerhez és a 13 vevő és dekódoló berendezéshez egy 15 feltételes hozzáférési rendszer kapcsolódik, amely részben a műsorszóró központban, részben a 13 vevő és dekódoló berendezésben van kialakítva. A 15 feltételes hozzáférési rendszer lehetővé teszi a felhasználók számára, hogy egy vagy több műsorszóró szolgáltatótól digitális televízió műsorokat fogadjon. A kereskedelmi ajánlatokra vonatkozó üzenetek dekódolására képes programozható csipkártya (ezek az üzenetek tulajdonképpen a műsorszóró szolgáltató által felkínált egy vagy több televízió programot jelentenek) helyezhető a 13 vevő és dekódoló berendezésbe, és a kettő együttes használatával a végfelhasználó vagy előfizetési módon, vagy helyben fizetési módon (pontosabban "nézd- és fizess", angolul pay-per-view módon) műsorok által megtestesülő eseményeket vásárolhat.

Mint említettük, a rendszer által továbbított programokat a 4 multiplexer és bitsorkódoló bitsorkódolja (ezt leírásunkban az egyszerűség kedvéért gyakran kódolásként is nevezzük), és az egy adott átvitelhez tartozó feltételeket és kódoló kulcsokat a 15 feltételes hozzáférési

rendszer határozza meg. Kódolt adatok ilyen átvitele a fizető televíziós rendszerek területén jól ismert. A kódolt adatokat általában az adatok dekódolásához szükséges ellenőrző szóval együtt továbbítják, ahol az ellenőrző szót magát egy úgynevezett megfejtő kulccsal kódolják, és ebben a kódolt formájában küldik el.

A kódolt (bitsorkódolt) adatokat és a kódolt ellenőrző szót a 13 vevő és dekódoló berendezés veszi, amelynek kellő hozzáférése van a dekódoló berendezésbe helyezett csipkártyán tárolt megfejtő kulcs másolathoz, hogy ezzel dekódolhassa a kódolt ellenőrző szót, majd annak birtokában dekódolja a küldött adatokat. Egy regisztrált előfizető például egy havonta sugárzott jogosultság vezérlő üzenetben megkapja a kódolt ellenőrző szó dekódolásához szükséges megfejtő kulcsot, amely lehetővé teszi az átvitt adatok, azaz műsor megtekintését számára.

Az ugyancsak a 4 multiplexerhez és a 13 vevő és dekódoló berendezéshez csatlakoztatott, és szintén részben a műsorsugárzó központban, részben a 13 vevő és dekódoló berendezésben kialakított 16 interaktív rendszer lehetővé teszi a végfelhasználó számára, hogy egy 17 modemes válaszcatornán keresztül interaktív módon kapcsolatba kerüljön számos alkalmazással. Ezt a 17 modemes válaszcatornát használhatjuk továbbá a 15 feltételes hozzáférési rendszerben alkalmazott kommunikáció céljára is. Ilyen 16 interaktív rendszert használhatunk például arra, hogy a nézőnek, azaz végfelhasználónak lehetővé tegyünk az azonnali kommunikációt a műsorszórás, továbbítást végző központtal, hogy egy meghatározott esemény megnézésére, egy alkalmazás letöltésére, és így tovább, jogosultságot szerezhessen.

Áttérve az 5. ábrára, az ábra segítségével a találmány szerinti terminálban használható 13 vevő és dekódoló berendezés, vagy set-top-box fizikai elemeit, egységeit mutatjuk be. Az egyes elemek a szakterületen ismert és szokásos módon épülnek fel, és megvalósításuk szakember köteles tudása körébe esőnek tekinthető.

A 13 vevő és dekódoló berendezés központi 20 processzort tartalmaz, amelyhez szokásos módon memória van társítva, és 21 soros interfésztől, 22 párhuzamos interfésztől, és (az 1. ábra 17 modemes válaszcatornájához csatlakoztatva) 23 modemtől bemenő adatokat képes fogadni.

A 13 vevő és dekódoló berendezés ezen túlmenően 26 vezérlő egységen keresztül infravörös 25 távvezérlőtől, valamint a 13 vevő és dekódoló és vevő berendezés előlapján elrendezett 24 kapcsolóktól vagy nyomógomboktól is bemeneti adatként értékelhető jeleket ké-

pes fogadni. A 13 vevő és dekódoló és vevő berendezés ezen túlmenően két 27, 28 kártya-olvasóval van ellátva, amelyek egy-egy csipkártya befogadására és olvasására képesek. A csipkártyák egy hitelintézet pénzügyi kártyája, illetőleg egy előfizetői kártya lehetnek. Adatok vihetők be továbbá egy, az ábrán nem látható infravörös kapcsolattal rendelkező billentyűzeten keresztül is. Az előfizetői 30 csipkártya olvasására szolgáló 28 kártyaolvasó a belehelyezett 30 csipkártyával, valamint 29 feltételes hozzáférési egységgel képes arra, hogy a szükséges ellenőrző szót 30 demultiplexerhez és bitsordekódolóhoz továbbítsa, amellyel lehetővé teszi a kódolt sugárzott jel dekódolását. A 13 vevő és dekódoló és vevő berendezés ezen túlmenően hagyományos 31 tunert és 32 demodulátort tartalmaz, amellyel a műholdasan átvitt jeleket veszi és demodulálja, mielőtt azokat a 30 demultiplexer és bitsordekódoló segítségével szűri és demultiplexeli.

A beérkező audio és video jelek esetében – amint ezt később részletesebben is ismertetjük – az ezeket a jeleket tartalmazó MPEG csomagokat demultiplexeljük és megszűrjük, és a valós idejű audio és video adatokat csomagokra osztott elemi adatfolyamok (packetised elementary stream, PES) formájában dedikált, audio és video jeleket feldolgozó processzorokhoz vagy 33, 34 dekódoló egységekhez továbbítjuk. Az audio jeleket feldolgozó 33 dekódoló egység átalakított kimenetét egy 35 előerősítőhöz továbbítjuk, majd a 13 vevő és dekódoló berendezés audio kimenetén továbbítjuk. A video jeleket feldolgozó dekódoló egység átalakított kimenetét egy 36 grafikus processzor és egy PAL/SECAM 37 kódoló egységen keresztül a 13 vevő és dekódoló berendezés video kimenetére továbbítjuk.

A 13 vevő és dekódoló berendezésben az adatok feldolgozását általánosságban a központi 20 processzor végzi. A központi 20 processzor olyan platformot biztosít, amely kellően rugalmas ahhoz, hogy egy alkalmazás különböző eszközökkel tudjon kommunikálni.

Leírásunkban alkalmazás alatt egy számítógépi kódkészletet értünk, amellyel előnyösen a 13 vevő és dekódoló berendezés magas szintű funkcióit vezéreljük. Például, ha a végfelhasználó távirányítóját, pontosabban annak fókuszált sugarát ráirányítja a 14 televíziókészülék képernyőjén látható gomb ikonra, és megnyomja a távirányító érvényesítés gombját, azzal lefuttatja a képernyőn látható nyomógommbal társított utasítás szekvenciát.

Egy interaktív alkalmazás menüket kínál fel, a végfelhasználó kérelmére parancsokat hajt végre és az alkalmazás céljához kapcsolódó és illeszkedő adatokat szolgáltat. Az alkalmazások vagy úgynevezett rezidens alkalmazások, amelyek a 13 vevő és dekódoló berendezés ROM-jában (vagy FLASH, vagy más nem-felejtő tárolójában) vannak eltárolva, vagy a 13

vevő és dekódoló berendezés RAM-jába (vagy FLASH-ROM-jába) kerülnek kisugárzás és letöltés útján.

Az alkalmazásokat a 13 vevő és dekódoló berendezés memóriahelyein tároljuk és ezek erőforrás fájlökként és adatokként jelennek meg. Az erőforrás fájlok grafikus objektum leíró egység fájlkból, változó blokk egység fájlkból, utasítás sorrend fájlkból és alkalmazás fájlkból állnak.

A 13 vevő és dekódoló berendezés dekódere olyan memóriát tartalmaz, amely egy írható olvasható RAM 54 memória részre, egy meghatározott módon frissíthető Flash 58 memória részre, valamint egy csak olvasható ROM 56 memória részre van felosztva, de ez a fizikai szervezés különbözik a logikai szervezéstől. A memóriát továbbá a különböző interfészekkel társított memória részekre oszthatjuk. Ha úgy vesszük, a memóriát a hardver részeként tekinthetjük, más szempontból viszont a memóriát a hardveren kívül eső, a rendszer egészét támogató vagy tartalmazó elemként is vehetjük.

A Flash 58 memória rész 70 lapokra van felosztva, amelyek mérete általában néhány kilobájt. Az adatok a Flash 58 memória részben adatblokkokként vannak eltárolva. A 6. ábrán a Flash 58 memória egy 70 lapjának az elrendezését, jobban mondva felosztását vázoltuk. Ennek a 70 lapnak 12 bájt hosszú 72 élőfeje, és változó hosszúságú 74 blokkjai vannak. Egy 74 blokk nagyságát a benne eltárolásra kerülő adat mérete határozza meg.

A 70 lap 72 élőfeje többek között tartalmazza a 70 lapnak a Flash 58 memória részen belüli 76 számát, valamint a 70 lap 78 állapotát. Egy 70 lap két stabil állapot közül az egyik állapotú lehet:

- i. Üres állapotú, azaz a 70 lap kizárólag üres 74 blokkokat tartalmazhat, és
- ii. Érvényes állapotú, amikor a 70 lap egy vagy több adat 74 blokkot tartalmaz.

A Flash 58 memória rész legalább egy üres állapotú 70 lapot tartalmaz.

Egy 70 lap az egy 70 lap tartalmának egy másik 70 lapra történő, az alábbiakban még részletesebben is bemutatásra kerülő másolása közben három ideiglenes állapot közül vehet fel egy állapotot:

- (i) egy ÍRÁS állapotot, amikor egy 70 lap tartalmát egy másik 70 lapra másoljuk;
- (ii) egy ÚJ állapotot, amikor a 70 lapra történő másolást már befejeztük; és
- (iii) egy ÉRVÉNYTELEN állapotot, amikor a 70 lapról annak tartalmát már átmásoltuk egy másik 70 lapra.

A 7. ábrán azt mutatjuk be, hogy minden egyes 74 blokk 16 bájt hosszú 80 élőfejjel rendelkezik, amely az alábbi mezőkből áll össze:

- (i) egy ahhoz az F fájlhoz hozzárendelt, egyedi számként megadott 82 azonosítót, amelyre a 74 blokk vonatkozik;
- (ii) a 74 blokk 84 nagyságát;
- (iii) a 74 blokk 86 rangsor azonosítóját, és
- (iv) a 74 blokk 88 állapotát.

A 86 rangsor azonosító egy egyedi azonosítót ad a 74 blokknak, ha egy F fájlt több ilyen 74 blokkra bontunk szét, és ilyen esetben minden egyes blokkhoz egy-egy 86 rangsor azonosítót. A 86 rangsor azonosító ezen túlmenően a blokknak az F fájlban belüli helyzetének a megadására is használjuk.

A nulla értékű 86 rangsor azonosítójú blokk tartalmazza az F fájlhoz tartozó élőfej blokkot. Az élőfej blokk nem tartalmaz az F fájl részét képező nyers adatokat, ehelyett fájl attribútumokat foglalja magában.

A 8. ábra egy F fájl 90 élőfej blokkjának a tartalmát mutatja. A 90 élőfej blokk 80 élőfejet tartalmaz, amelyben a 86 rangsor azonosító értéke nullára van beállítva. A 90 élőfej blokknak ezen kívül 92 apa azonosító mezeje van. A 92 apa azonosító mező a Flash 58 memória részben tárolt D könyvtárak és F fájlok fa struktúrájában az F fájl "apa" 82 azonosítóját tartalmazza. A 9. ábra a Flash 58 memória részben tárolt blokkok fa struktúrájára mutat be példát. A blokkok D könyvtár blokkok és élőfej blokkok, valamint az F fájl tényleges, nyers adatait tartalmazó adatblokkok. Egy D könyvtár blokk egy olyan élőfej blokk, például D2 könyvtár blokk, amely nincs közvetlenül F fájllal társítva, de egyéb D könyvtár blokk vagy élőfej blokk számára helymeghatározóként szolgál.

A 13 vevő és dekódoló berendezés bekapcsolását követően a Flash 58 memória részben tárolt adatok fa struktúráját kiolvassuk, és annak vázlatát a RAM 54 részben eltároljuk egy grafikon vagy más megfelelő alakban. A fa struktúrát úgy határozzuk meg, hogy minden egyes adatblokk 82 azonosítóját és 86 rangsor azonosítóját, továbbá minden egyes D könyvtár blokk és élőfej blokk 82 azonosítóját, 86 rangsor azonosítóját és 92 apa azonosítóját kiolvassuk. Például a D1 élőfej blokk esetében a 82 azonosító értéke 1, a 86 rangsor azonosító értéke 0, a 92 apa azonosító értéke pedig 0 (mivel nincs további "apja"). Az F3 fájl blokk 82 azonosítója 1, 86 rangsor azonosítója is 1. A D4 könyvtár blokk 82 azonosítója 4, a 86 rangsor azonosítója 0, és 92 apa azonosítója 1 (mivel az ő "apja" az a D1 élőfej

blokk, amelynek 82 azonosítója 1 értékű). A D5 élőfej blokk 82 azonosítója 5, 86 rangsor azonosítója 0, és 92 apa azonosítója 2.

Ily módon nagyon gyorsan létre tudjuk hozni az adatok fa struktúráját, mivel annak létrehozásához minden egyes blokknak csupán három adatmezejét kell elolvasnunk.

Ezen túlmenően, ha egy élőfej blokkot és az ahhoz tartozó adatblokkokat az adat fa struktúráján belül átmozgatjuk, csupán az élőfej blokk élőfejében kell az új 92 apa azonosítót beírunk. Például, ha a D17 élőfej blokkot és az ahhoz tartozó F20, F21, F22 adat blokkokat úgy mozgatjuk át, hogy a D17 élőfej blokk új apja a D15 élőfej blokk lesz, abban az esetben csak a D17 élőfej blokk 92 apa azonosítóját kell 15-re változtatnunk. Nem kell viszont módosítanunk a D17 élőfej blokk közvetlen elődjének, vagy pedig a D17 élőfej blokk új közvetlen elődjének, azaz a D15 blokknak az élőfejét.

Ha a Flash 58 memóriában tárolt adatok fa struktúrája módosul, a RAM 54 részben tárolt diagramot is felfrissítjük.

Mint korábban utaltunk rá, a 13 vevő és dekódoló berendezésben az adatokat vagy egyetlen memória részben, vagy több memória részen keresztül tudjuk tárolni. Az adatoknak a Flash 58 memóriában történő, fent leírt fa struktúrájú szervezése, amelyben az élőfej blokkok apjukra utalnak vissza, ugyancsak alkalmazható a ROM memóriában, RAM memóriában, vagy a ROM memória, RAM memória és Flash 58 memória bármely kombinációjában tárolt adatok vonatkozásában is. Az újfajta fa struktúrájú adatszervezés által biztosított előnyök leginkább a Flash 58 memóriában tárolt adatokkal kapcsolatosan tűnnek ki, melynek az esetében az abban tárolt adatok újraírásához szükséges idő a legnagyobb. Így a RAM memóriában és a ROM memóriában tárolt adatokat alternatív módon a Flash 58 memóriában tárolhatjuk, a 2. és 3. ábra segítségével az előzőekben már leírt fa struktúrájú elrendezéssel.

Ha ismét rátekintünk a 8. ábrára látható, hogy a 90 élőfej blokk olyan 93 attribútum mezőt tartalmaz, amely azt jelzi, hogy a kérdéses blokk egy D könyvtár, vagy pedig egy élőfej blokk.

A 90 élőfej blokk 94 hozzáférési mód mezőt, 96 tulajdonos mezőt és 98 csoport mezőt tartalmaz. Ez a három mező teszi lehetővé, hogy az F fájlhoz történő hozzáférést a különböző felek számára részlegesen vagy teljességgel letiltsuk.

A 96 tulajdonos mező az F fájl szerzőjének azonosítóját tartalmazza. A 98 csoport mező az



F fájl arra feljogosított felhasználóinak egy csoport azonosítóját tartalmazza. Például, ha az F fájl egy csipkártya olvasóhoz való programhoz tartozik, akkor az F fájl feljogosított felhasználói közé fognak tartozni azoknak az interaktív alkalmazásoknak a szolgáltatói, gyártói is, amelyek ezt a programot használják. A F fájl minden egyes arra feljogosított felhasználójának az azonosítói a 13 vevő és dekódoló berendezésben, egy biztonsági modulban kerülnek eltárolásra.

A 94 hozzáférési mód mező tartalma fogja meghatározni azt a mértéket, amennyire az F fájlhoz az F fájl szerzője hozzáférhet, ahogy az a 96 tulajdonos mezőben szerepel, vagy a 98 csoport mezőben specifikált, az F fájl vonatkozásában feljogosított felhasználó, valamint mások, például a 13 vevő és dekódoló berendezés gyártója, vagy a 13 vevő és dekódoló berendezés egy felhasználója.

A 10. ábrán részletesebben is bemutatjuk az előbb említett 94 hozzáférési mód mezőt. Látható, hogy a 94 hozzáférési mód mező hat bites, ebből két 100, 102 bit a szerzőhöz tartozik, további két 104, 106 bit a feljogosított felhasználókhoz tartozik, a maradék két 108, 110 bit pedig mások számára van fenntartva. A 100, 104 és 108 bitek értéke fogja meghatározni, hogy az F fájl olvasása tiltott-e a szerző, az arra feljogosított felhasználók, illetve mások számára, míg a 102, 106 és 110 bitek értéke határozza meg, hogy a kérdéses F fájl felülírása tiltott-e a szerző, az arra feljogosított felhasználók, vagy más számára. A 10. ábrán bemutatott példában az F fájl az arra feljogosított felhasználók nem írhatják felül, valamint az F fájl mások nem is olvashatják és felül sem írhatják.

A 94 hozzáférési mód mező, 96 tulajdonos mező és 98 csoport mező tartalmát még az F fájlban a 13 vevő és dekódoló berendezésben történő eltárolását megelőzően beállítjuk. Ha egy F fájl a műsorszóró központ hat adójával a 13 vevő és dekódoló berendezéshez kell továbbítanunk, az előbb említett mezők tartalmát még a műsorszóró központban, az elküldést megelőzően a szükséges értékre tudjuk állítani. Alternatív megoldásként ezeknek a mezőknek a tartalmát, vagyis értékét a szerző határozhatja meg, az F fájl írása során illetve közben.

Ha egy F fájlhoz hozzáférésre van szükség, akkor a hozzáférést kérelmező saját azonosítójával azonosítja magát. Ennek alapján az 50 központi processzor meghatározza, hogy a hozzáférést kérelmező azonosítója megfelel-e a 96 tulajdonos mezőben tárolt azonosítónak. Amennyiben megegyezik, úgy a felhasználó az F fájl olyan szerzőjeként kerül azonosításra, akihez a 100 és 102 bitekben eltárolt hozzáférési jogosultság készlet tartozik. A 10.



ábrán bemutatott esetben az F fájl szerző általi olvasását és felülírását az 50 központi processzor engedélyezi.

Ha a hozzáférést kérelmező nem az F fájl szerzője, akkor az 50 központi processzor a biztonsági modulhoz fordul, hogy megállapítsa vajon a hozzáférést kérelmező azonosítója tagja-e az arra feljogosított felhasználóknak a 98 csoportjának. Amennyiben igen, úgy a hozzáférést kérelmező az F fájl vonatkozásában feljogosított felhasználóként kerül azonosításra, akihez a 104 és 106 bitekben eltárolt hozzáférési jogosultság készlet van hozzárendelve. A 10. ábrán látható esetben a jogosult felhasználó számára az 50 központi processzor kizárólag az F fájl olvasását engedélyezte.

Ha egy hozzáférést kérelmező nem tekinthető az F fájl arra feljogosított felhasználójának, akkor a hozzáférés kérelmezőt "egyéb" kategóriába soroljuk, akihez a 108 és 110 bitben tárolt hozzáférési jogosultság készlet van hozzárendelve. A 10. ábrán bemutatott példában az ilyen jellegű, ide tartozó kérelmező számára az F fájlhoz történő bármilyen hozzáférést, azaz olvasást, vagy újírást, megtiltotta az 50 központi processzor.

A 100-110 bitek értékét kizárólag az F fájl felülírásával tudjuk módosítani. A 10. ábrán vázolt példában kizárólag az F fájl szerzője számára engedélyezett az F fájl felülírása. Ily módon a 94 hozzáférési mód mező, a 96 tulajdonos mező és a 98 csoport mező a 90 élőfej blokkban egyszerű, de mégis hatékony biztonsági rendszert képez az adott F fájl számára.

Visszatérve röviden a 8. ábrára, látható, hogy a 90 élőfej blokk olyan 112 verzió mezőt is tartalmaz, amely az F fájl verziószámát tartalmazza, valamint olyan 114 név mezőt tartalmaz, amely az F fájl ASCII kódolású nevét tartalmazza.

Visszatérve kicsit a 7. ábrára látható, hogy a Flash 58 memória egy 70 lapjának egy blokkjának 88 ÁLLAPOT mezeje az alábbi három stabil állapot valamelyikét veheti fel:

- (i) Egy SZABAD állapot, amikor a blokk üres; és egy
- (ii) ÉRVÉNYTELEN állapot, amikor a blokk egy új verzióját a Flash 58 memória egy másik 70 lapjára átmásoltuk, vagy pedig amikor a fájl mezőt töröltük, valamint egy
- (iii) ÉRVÉNYES állapot, ha a blokk előző verzióját ÉRVÉNYTELEN állapotúra változtattuk.

Egy blokk 88 ÁLLAPOT mezeje olyan egybites jelzőt foglal magában, amely azt jelzi, hogy a kérdéses blokk ÉRVÉNYES vagy ÉRVÉNYTELEN. Ha ennek az egybites jelzőnek az értéke 1, akkor a blokk ÉRVÉNYES, és ha az egybites jelző értéke 0, akkor a blokkot ÉRVÉNYTELEN-nek tekintjük. Így, ha egy blokk állapotát ÉRVÉNYES-ről ÉRVÉNYTELEN-né változtatjuk, akkor ezt a nevezett bitet töröljük, vagyis értékét 1-ről 0-ra változtatjuk. Így a Flash



58 memóriában egy blokk állapotának a megváltozását anélkül tudjuk rögzíteni, hogy a Flash 58 memória teljes érintett 70 lapját újra kellene írunk.

Általában csupán a Flash 58 memória 70 lapjának az utolsó blokkja SZABAD állapotú, és az új adatoknak az adott 70 lapon történő eltárolásához rendelkezésre álló szabad memóriateret jelenti.

Egy blokk tartalmának írása közben az alábbi három ideiglenes állapot egyikét veheti fel, vagy pedig a blokk tartalmának a RAM-on keresztül történő másik 70 lpra történő másolása közben (mint azt az alábbiakban még részletesebben is bemutatjuk):

- (i) egy LÉTREHOZÁS állapot, amikor a blokk élőfejét írjuk;
- (ii) ÍRÁS állapot, amikor a blokk maradék tartalmát írjuk; és
- (iii) egy ÚJ állapot, amikor a blokkot befejeztük, de az F fájl többi blokkjának az írását vagy másolását még nem fejeztük be.

A 11. ábrán egy első 70 lapon lévő egyetlen ÉRVÉNYES blokknak egy második 70 lapon lévő SZABAD blokkba történő másolásának a folyamatát vázoltuk fel. 100 lépésben a második 70 lapon lévő SZABAD blokk állapotát LÉTREHOZÁS-ra módosítjuk, míg az ÉRVÉNYES blokk élőfejét a RAM memória segítségével bemásoljuk a blokkba. A 102 lépésben a LÉTREHOZÁS blokk állapotát ÍRÁS-ra módosítjuk, miközben az ÉRVÉNYES blokk tartalmát másoljuk. A 104 lépésben az ÍRÁS blokk állapotát ÚJ-ra módosítjuk, amikor a blokk teljes tartalmának a másolását befejeztük. A 106 lépésben az első 70 lapon lévő másolt ÉRVÉNYES blokk állapotát ÉRVÉNYTELEN-re módosítjuk, majd végül a 108 lépésben a második 70 lapon lévő ÚJ blokk állapotát ÉRVÉNYES-re módosítjuk.

Azáltal, hogy a 104 lépésben az ÍRÁS blokknak egy ÚJ állapotot adunk, nem pedig ÉRVÉNYES állapotot, amikor a blokk tartalmának a másolását befejeztük, biztosítani tudjuk, hogy ugyanabban az időpontban nem létezik két ÉRVÉNYES blokk, amelynek mind az azonosítója, mind a rangsor azonosítója megegyezik. Ezzel meggátoljuk azt a lehetőséget, hogy egyidejűleg két ilyen ÉRVÉNYES blokk létezzen, ha az operációs rendszer a 104 és 106 lépések közötti időben valamiért összeomlik.

Ha a rendszer így összeomlana, miközben egy adatblokkot másolunk a Flash 58 memória egyik 70 lapjáról egy másik 70 lapjára, akkor a rendszerindítást követően az operációs rendszer megvizsgálja, hogy a Flash 58 memóriában talál-e ÚJ blokkot. Ha talál ÚJ blokkot, akkor az operációs rendszer egy olyan ÉRVÉNYES blokkot keres, amelynek ugyanaz az azonosítója és a rangsor azonosítója, mint az előbb megtalált ÚJ blokké. Ha talál ilyen



ÉRVÉNYES blokkot, akkor az előbb megtalált ÚJ blokk állapotát ÉRVÉNYTELEN-re változtatja, ha pedig nem talál, akkor az ÚJ blokk állapotát ÉRVÉNYES-re változtatja meg. Ezzel biztosítjuk, hogy a memóriában nem lesz két ÉRVÉNYES és SZABAD blokk, amelynek ugyanaz az azonosítója és a rangsor azonosítója, anélkül, hogy a Flash 58 memória valamelyik 70 lapját újra íránk.

Ha ugyanolyan azonosítójú több ÉRVÉNYES blokkot kell egy első 70 lapról egy második 70 lapra másolnunk, úgy az első ÉRVÉNYES blokk másolatát felfüggesztjük, miután a 104 lépést befejeztük. Minden egyes rákövetkező ÉRVÉNYES blokk másolását csupán azt követően végezzük el, majd az első 70 lap összes másolt ÉRVÉNYES blokkjának az állapotát ÉRVÉNYTELEN-re változtatjuk, és második 70 lapon lévő ÚJ blokkok állapotát ÉRVÉNYES-re módosítjuk.

Ha a rendszer közben omlik össze, miközben a Flash 58 memória egyik 70 lapjáról adatblokkokat másolunk át egy másik 70 lapjára, akkor a rendszer indítás során az operációs rendszer megvizsgálja, hogy a Flash 58 memóriában található-e egy vagy több ÚJ blokk. Ha több ilyen ÚJ blokkot talál, akkor az operációs rendszer olyan érvényes és érvénytelen blokkokat keres, amelyek azonosítója, illetve rangsor azonosítója megegyezik az ÚJ blokkokéval.

Ha a megfelelő blokkok mindegyike ÉRVÉNYES, akkor az ÚJ blokkok állapotát kivétel nélkül ÉRVÉNYTELEN-re módosítja. Ha nem, vagyis legalább egy megfelelő blokk ÉRVÉNYTELEN állapotú, akkor az összes maradék megfelelő ÉRVÉNYES blokk állapotát ÉRVÉNYTELEN-re változtatja, majd mindegyik ÚJ blokk állapotát ÉRVÉNYES-re módosítja.

Mint korábban jeleztük, a Flash 58 memóriában az egyik 70 lapot üres állapotban tartjuk. Idővel a Flash 58 memória összes maradék 70 lapja megtelik ÉRVÉNYES és ÉRVÉNYTELEN adatblokkokkal, így szükségessé válik az ÉRVÉNYTELEN adatblokkok közül néhány vagy mindegyik törlése, hogy elegendő helyet biztosítsunk a friss adatokat tartalmazó blokkok számára. Azonban az adatblokkok részlegesen nem írhatók felül új adatokkal, a Flash 58 memória egy teljes 70 lapját törölnünk kell ahhoz, hogy azt a 70 lapot újra felhasználhassuk.

Annak érdekében, hogy elegendő helyet biztosítsunk a Flash 58 memóriába beírása kerülő új F fájl számára, úgy kizárólag egy ÉRVÉNYES 70 lapon lévő ÉRVÉNYES blokkokat másolunk a RAM memória segítségével egy ÜRES 70 lapra, az ÉRVÉNYES 70 lap ÉRVÉNYTELEN blokkjait nem másoljuk át. Az ÉRVÉNYES blokkokat úgy másoljuk, hogy a blokkok közül

legalább egy olyan helyzetet vesz fel az ÜRES 70 lapon, amely különbözik az ÉRVÉNYES 70 lapon elfoglalt helyzetétől. Például az ÉRVÉNYES blokkokat az ÜRES 70 lap tetejére másolhatjuk. Ha a másolás során az ÉRVÉNYES blokkokat átrendezzük, létrehozhatjuk az ÜRES 70 lapon a legnagyobb töredezettség mentes memória területet és ezzel elegendő helyet hagyunk, például az üres 70 lap alsó részén az új adatok eltárolására. Az ÉRVÉNYES 70 lap tartalmát töröljük, az ÜRES 70 lapot ÉRVÉNYES 70 lappá minősítjük át, és a megelőző ÉRVÉNYES 70 lap pedig ÜRES 70 lap lesz.

A Flash 58 memória tartalmának a Flash 58 memóriába történő új adatok beírásakor alkalmazott eljárást a 12. és 13. ábrák segítségével részletesebben is bemutatjuk.

S200 lépésben az 50 központi processzorral a Flash 58 memória minden egyes érvényes 70 lapjának minden egyes 74 blokkja 80 élőfejét átnézzük, hogy megtaláljuk azt az ÉRVÉNYES 70 lapot, amelyben az ÉRVÉNYTELEN blokkok együttes nagysága a legnagyobb. S202 lépésben kiválasztjuk az így beazonosított ÉRVÉNYES 70 lapot. S204 lépésben az ÜRES 70 lap állapotát írásra módosítjuk, és a kiválasztott érvényes 70 lap számát eltároljuk az ÍRÁS 70 lap élőfejében. S206 lépésben az 50 központi processzorral megállapítjuk, hogy az ÉRVÉNYES 70 lapon vannak-e ÉRVÉNYES blokkok. Ha az ÉRVÉNYES 70 lapon legalább egy vagy több ÉRVÉNYES blokkot találunk, úgy S208 lépésben az első ÉRVÉNYES blokkot a RAM memória segítségével az ÍRÁS 70 lapra másoljuk. S 210 lépésben az 50 központi processzorral megvizsgáljuk, hogy maradtak-e ÉRVÉNYES blokkok az érvényes 70 lapon. Ha egy második vagy további ÉRVÉNYES blokkot találunk, úgy S212 lépésben ezt a második ÉRVÉNYES blokkot a RAM memórián keresztül bemásoljuk az ÍRÁS 70 lapra, közvetlenül az ezt megelőzően odamásolt ÉRVÉNYES blokk mögé. Az S210 és S212 lépéseket szükség esetén mindaddig ismételjük, amíg az 50 központi processzorral azt állapítjuk meg, hogy az ÉRVÉNYES 70 lapon nem maradt több ÉRVÉNYES blokk. Ebben az esetben S214 lépésben az ÍRÁS 70 lap állapotát ÚJRA módosítjuk. S216 lépésben az ÉRVÉNYES 70 lap állapotát ÉRVÉNYTELEN-re módosítjuk, S218 lépésben az új 70 lap állapotát ÉRVÉNYES-re módosítjuk, és végül S220 lépésben az ÉRVÉNYTELEN 70 lapot töröljük, és a törölt 70 lap állapotát ÜRESRE változtatjuk, így ismét legalább egy üres 70 lapunk lesz a Flash 58 memóriában.

A fent bemutatott eljárás általános alapvető hatását összességében a 13. ábrán mutatjuk be. Ennél az előnyös kiviteli alaknál a Flash 58 memória ÉRVÉNYES 70 lapjának V ÉRVÉNYES blokkjait hatékonyan átmozgattuk az ÜRES 70 lap tetejére, annak érdekében, hogy elegendő tároló kapacitást hozzunk létre az ÜRES 70 lapon, annak alsó részén az új adatok eltárolásához, miután az ÜRES 70 lap állapotát ÉRVÉNYES-re módosítottuk.



A találmány szerinti eljárást leírásunkban valamint a rajzon kizárólag a találmányi gondolat megértetése céljából és az ahhoz szükséges mértékben ismertettük, így attól a találmány igényelt oltalmi körén belül nyilvánvalóan széles körben el lehet térni. A leírásban, az igénypontokban és a rajzokon látható jellemzők mind külön-külön, mind megfelelő kombinációban megvalósíthatók.

Szabadalmi igénypontok

1. Eljárás adathozzáférés korlátozására, memóriával rendelkező vevő és dekódoló berendezésben, **azzal jellemezve**, hogy az adatokhoz hozzáférési jogosultság készleteket rendelünk hozzá, ahol minden egyes hozzáférési jogosultság készletet legalább egy félhez társítunk; a vevő és dekódoló berendezés (13) memóriájában az adatokat, a hozzáférési jogosultság készleteket valamint minden egyes félhez tartozó azonosítót eltárolunk; az adatokhoz hozzáférést kérelmező félhez tartozó azonosítót összehasonlítjuk a memóriában tárolt legalább egy azonosítóval; és a vevő és dekódoló berendezés (13) memóriájában hozzárendelt hozzáférési jogosultság készletet a fél rendelkezésére bocsátjuk.
2. Az 1. igénypont szerinti eljárás, **azzal jellemezve**, hogy a hozzáférés jogosultság készleteket az adatok élőfejében tároljuk el.
3. Az 1. vagy 2. igénypont szerinti eljárás, **azzal jellemezve**, hogy a felekre vonatkozó azonosítókat az adatok élőfejében tároljuk el.
4. Az 1-3. igénypontok bármelyike szerinti eljárás, **azzal jellemezve**, hogy az adatokat egy adatátviteli rendszer által továbbított bitfolyamból töltjük le, és az egyes felekhez tartozó hozzáférési jogosultság készleteket és azonosítókat a adatátviteli rendszerben az adatokon belül tároljuk.
5. Eljárás adathozzáférés korlátozására, egy digitális műsorszóró rendszerben sugárzott adatokhoz, **azzal jellemezve**, hogy az adó rendszerben az adatokhoz hozzáférési jogosultság készleteket rendelünk hozzá, melyek mindegyikét legalább egy félhez hozzárendeljük; az egyes felekhez tartozó hozzáférési jogosultság készleteket és azonosítókat az adatokban eltároljuk; és az adatokat továbbítjuk; míg egy memóriával rendelkező vevő és dekódoló berendezésben (13) a továbbított adatokat a vevő és dekódoló berendezés (13) memóriájába letöltjük és eltároljuk; az adatokhoz hozzáférést kérelmező fél azonosítóját a memóriá-

ban tárolt azonosítókkal összehasonlítjuk; és a fél részére a vevő és dekódoló berendezés memóriájában hozzárendelt hozzáférési jogosultság készletet megadjuk.

6. Az 5. igénypont szerinti eljárás, *azzal jellemezve*, hogy az adatokat egy digitális bitfolyamban továbbítjuk.

7. Az 1-6. igénypontok bármelyike szerinti eljárás, *azzal jellemezve*, hogy további hozzáférési jogosultság készletet rendelünk legalább egy olyan félhez hozzá, akinek az azonosítója nincs a vevő és dekódoló berendezés (13) memóriájában eltárolva, és az adatokhoz hozzáférést kérelmező ilyen felet az így hozzárendelt további hozzáférési jogosultság készlettel látjuk el.

8. Az 1-7. igénypontok bármelyike szerinti eljárás, *azzal jellemezve*, hogy egy meghatározott hozzáférési jogosultság készletet csupán egyetlen félhez, előnyösen az adatok szerzőjéhez rendelünk hozzá.

9. Az 1-8. igénypontok bármelyike szerinti eljárás, *azzal jellemezve*, hogy egy meghatározott hozzáférési jogosultság készletet egy felekből álló csoporthoz rendelünk hozzá, és a csoport minden egyes tagjához tartozó azonosítót tároljuk a vevő és dekódoló berendezés (13) memóriájában.

10. Az 1-9. igénypontok bármelyike szerinti eljárás, *azzal jellemezve*, hogy fél adatok olvasásából kizárt státuszának a megállapítására egy hozzáférési jogosultság készletet használunk fel.

11. Az 1-10. igénypontok bármelyike szerinti eljárás, *azzal jellemezve*, hogy fél adatok felülírásából kizárt státuszának a megállapítására egy hozzáférési jogosultság készletet használunk fel.

12. Az 1-11. igénypontok bármelyike szerinti eljárás, *azzal jellemezve*, hogy az adatokat a vevő és dekódoló berendezés (13) memóriájában fájlok (F) és könyvtárak (D) alakjában tároljuk.

13. A 12. igénypont szerinti eljárás, *azzal jellemezve*, hogy a fájlok (F) és könyvtárak (D) fa struktúráját a vevő és dekódoló berendezés (13) memóriájában rögzítjük.

14. A 13. igénypont szerinti eljárás, *azzal jellemezve*, hogy az adat fa struktúrában egy adott esetben egy fájlt (F) vagy könyvtárat (D) közvetlenül megelőző könyvtár (D) azonosítóját a fájlal (F) és könyvtárral (D) társítva eltároljuk.

15. A 14. igénypont szerinti eljárás, *azzal jellemezve*, hogy az azonosítót a fájl (F) vagy

könyvtár (D) élőfejében tároljuk el.

16. Az 1-15. igénypontok bármelyike szerinti eljárás, *azzal jellemezve*, hogy az adatokat a vevő és dekódoló berendezés (13) Flash memóriájában (58) tároljuk el.

17. Berendezés egy vevő és dekódoló berendezés memóriájában tárolt adatokhoz való adathozzáférés korlátozására, *azzal jellemezve*, hogy az adatokkal hozzáférési jogosultsági készleteket vannak társítva, minden egyes hozzáférési jogosultság készletet legalább egy félhez hozzá van rendelve, és minden egyes félre vonatkozó azonosítót tároló vevő és dekódoló berendezése (13); az adatokhoz hozzáférést kérelmező fél azonosítóját a vevő és dekódoló berendezés (13) memóriájában tárolt azonosítókkal összehasonlító eszköze; valamint a fél számára a hozzárendelt hozzáférési jogosultság készletet a vevő és dekódoló berendezés (13) memóriájában lévő, hozzárendelt hozzáférési jogosultság készletet rendelkezésre bocsátó eszköze van.

18. Vevő és dekódoló berendezés, amely adatokat tároló memóriát tartalmaz, *azzal jellemezve*, hogy az adatokhoz hozzárendelt hozzáférési jogosultság készleteket, minden egyes félhez tartozó azonosítót, valamint az előbb vázolt, az adatokhoz való hozzáférést korlátozó berendezést foglal magában.

19. Adatátviteli rendszer *azzal jellemezve*, hogy az adatokhoz hozzáférési jogosultság készleteket társító eszköze van, ahol minden egyes hozzáférési jogosultság készlet legalább egy félhez van hozzárendelve, továbbá a felekhez tartozó hozzáférési jogosultság készleteket és azonosítókat az adatokon belül tároló eszköze, valamint az adatokat is magában foglaló bitfolyamot továbbító eszköze van.

20. A 19. igénypont szerinti adatátviteli rendszer, *azzal jellemezve*, hogy a 18. igénypont szerinti vevő és dekódoló berendezéssel (13) van kombinálva.

21. Eljárás fájlok és könyvtárak alakjában tárolt adatok fa struktúrájának a rögzítésére, *azzal jellemezve*, hogy minden egyes fájl (F) és könyvtár (D) vonatkozásában egy ahhoz tartozó és az adatok fa struktúrája szerint a fájlt (F) vagy könyvtárat (D) közvetlenül megelőző könyvtár (D) azonosítót tárolunk el.

22. A 21. igénypont szerinti eljárás, *azzal jellemezve*, hogy az azonosító a könyvtárhoz (D) hozzárendelt egyedi kódot tartalmaz.

23. A 21. vagy 22. igénypont szerinti eljárás, *azzal jellemezve*, hogy az azonosítót a fájl (F) vagy könyvtár (D) élőfejében tároljuk el.

24. A 23. igénypont szerinti eljárás, *azzal jellemezve*, hogy az adatok legalább egy részét, előnyösen egészét Flash 58 memóriában tároljuk el.
25. A 24. igénypont szerinti eljárás, *azzal jellemezve*, hogy a Flash memória (58) részben tárolt adatok virtuális fa struktúráját előállítjuk és egy RAM memória (54) részben tároljuk.
26. A 24. vagy 25. igénypont szerinti eljárás, *azzal jellemezve*, hogy a Flash memória (58) részben tárolt fájl (F) előfejét a Flash memória (58) rész dedikált blokkjában tároljuk.
27. A 21-26. igénypontok bármelyike szerinti eljárás, *azzal jellemezve*, hogy az eltárolt azonosítót egy fájl (F) vagy könyvtár (D) közvetlenül megelőző másik könyvtárba (D) történő mozgatása során módosítjuk.
28. A 21-27. igénypontok bármelyike szerinti eljárás, *azzal jellemezve*, hogy az adatokat egy vevő és dekódoló berendezésben (13) tároljuk.
29. Berendezés fájlok és könyvtárak alakjában tárolt adatok fa struktúrájának rögzítésére, *azzal jellemezve*, hogy az adatok fa szerkezetében a fájl (F) vagy könyvtárat (D) közvetlenül megelőző könyvtár azonosítóját minden egyes fájl (F) és könyvtár (D) vonatkozásában eltároló eszköze van.
30. Vevő és dekódoló berendezés *azzal jellemezve*, hogy a 29. igénypont szerinti berendezést tartalmaz.
31. Eljárás adatblokkok memória lapok közötti átvitelére, újabb adatoknak a memóriában történő eltárolásának a biztosítására, *azzal jellemezve*, hogy a memória több lapot (70) tartalmaz, és a memória egy lapját (70) érvényes adatot tartalmazó legalább egy érvényes blokkot magában foglaló forrás lapként (70) tekintünk, továbbá érvénytelen adatot tartalmazó legalább egy érvénytelen blokkot, és a memória legalább egy érvénytelen blokkot, és a memória legalább egy lapját (70) átvivő lapként (70) tekintünk, és a forrás lapról (70) az egy vagy mindegyik érvényes blokkot továbbítjuk az átvivő lapra (70), ahol legalább egy ilyen blokknak az átvivő lapon (70) a forrás lapon (70) elfoglalt helyzetétől különböző pozíciója van, és a forrás lapot (70) töröljük.
32. A 31. igénypont szerinti eljárás, *azzal jellemezve*, hogy az adatot ezt követően az átvivő lapon (70) a másolt érvényes blokkok szomszédságában tároljuk el.
33. Eljárás adatok memóriában történő eltárolására, ahol a memória több lapot tartalmaz, és a memória egy lapja érvényes adatot tartalmazó legalább egy érvényes blokkot, érvény-

telen adatot tartalmazó legalább egy érvénytelen blokkot magában foglaló forrás lapot tartalmaz, és a memória legalább egy lapját kiinduláskor kizárólag szabad blokkokat tartalmazó átvivő lapként tekintjük, **azzal jellemezve**, hogy a legalább egy érvényes blokkot a forrás lapról (70) az átvivő lapra (70) másoljuk, ahol legalább egy ilyen blokk az átvivő lapon (70) a forrás lapon (70) elfoglalt helyzetétől különböző helyzetű lesz; a forrás lapot (70) töröljük; és az adatokat az átvivő lapon (70) tároljuk.

34. A 31-33. igénypontok bármelyike szerinti eljárás, **azzal jellemezve**, hogy a legalább egy érvényes blokkot úgy másoljuk az átvivő lapra (70), hogy ezzel friss adatok vételéhez a lehető legnagyobb töredezetlen memória területet hozzuk létre az átvivő lapon (70).

35. A 31-34. igénypontok bármelyike szerinti eljárás, **azzal jellemezve**, hogy a törölt forrás lapot (70) előnyösen közvetlenül annak törlését követően új átvivő lapként (70) tekintjük.

36. A 31-35. igénypontok bármelyike szerinti eljárás, **azzal jellemezve**, hogy egy lapot (70) a lap (70) érvénytelen blokkjainak együttes nagysága függvényében tekintünk forrás lapként (70).

37. A 36. igénypont szerinti eljárás, **azzal jellemezve**, hogy a forrás lap (70) legalább egy érvénytelen blokkot tartalmazó lap (70), ahol a legalább egy érvénytelen blokk együttes nagysága az új adatok méretével azonos, vagy annál nagyobb.

38. A 31-37. igénypontok bármelyike szerinti eljárás, **azzal jellemezve**, hogy a blokkok változó méretű blokkok.

39. A 31-38. igénypontok bármelyike szerinti eljárás, **azzal jellemezve**, hogy a memória adat szabad írását gátló memóriát, előnyösen Flash memória (58) részt foglal magában.

40. A 39. igénypont szerinti eljárás, **azzal jellemezve**, hogy az érvényes blokkot a blokkban tárolt egybites jelző értékének megváltoztatásával módosítjuk érvénytelen blokká.

41. A 40. igénypont szerinti eljárás, **azzal jellemezve**, hogy az egybites jelzőt a blokk előfejében tároljuk.

42. A 31-41. igénypontok bármelyike szerinti eljárás, **azzal jellemezve**, hogy a memória egy vevő és dekódoló berendezés (13) memória részét tartalmazza.

43. A 42. igénypont szerinti eljárás, **azzal jellemezve**, hogy az adatok bitfolyamból letöltött adatok.

44. A 43. igénypont szerinti eljárás, **azzal jellemezve**, hogy az adatokat MPEG táblázatok

alakjában formázzuk.

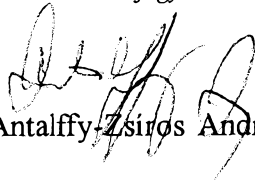
45. Berendezés adatblokkok memória lapok közötti továbbítására, új adat memóriában történő eltárolásának az elősegítésére, mely berendezés memóriája lapokat tartalmaz, és a memória egy lapja érvényes adatot tartalmazó legalább egy érvényes blokkot, és érvénytelen adatot tartalmazó legalább egy érvénytelen blokkot magában foglaló forrás lapként van kijelölve, és a memória legalább egy lapja átvivő lapként van kijelölve, **azzal jellemezve**, hogy a forrás lapról (70) a legalább egy érvényes blokkot az átvivő lapra (70) másoló eszköz tartalmaz, ahol a legalább egy ilyen blokk átvivő lapon (70) elfoglalt helyzete eltér a blokk forrás lapon (70) elfoglalt pozíciójától, valamint a forrás lapot (70) törölő eszközzel rendelkezik.

46. Vevő és dekódoló berendezés, **azzal jellemezve**, hogy a 45. igénypont szerinti berendezést tartalmaz.

A meghatalmazott:

DANUBIA

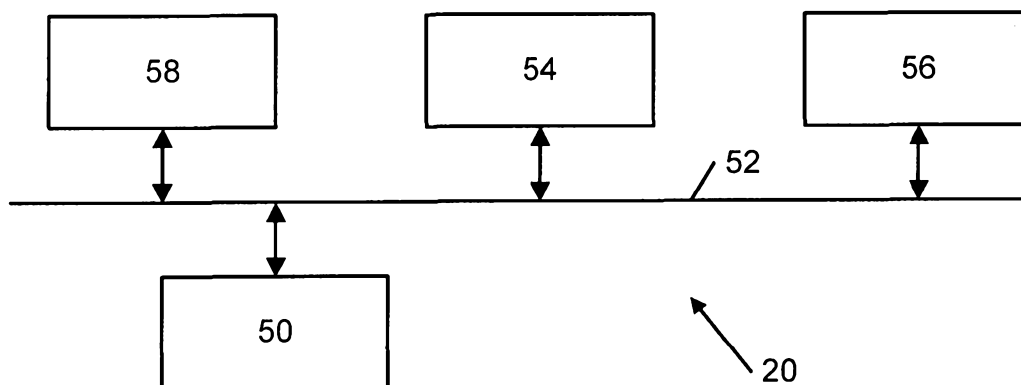
Szabadalmi és Védjegy Iroda Kft.


dr. Antalffy Zsolt András

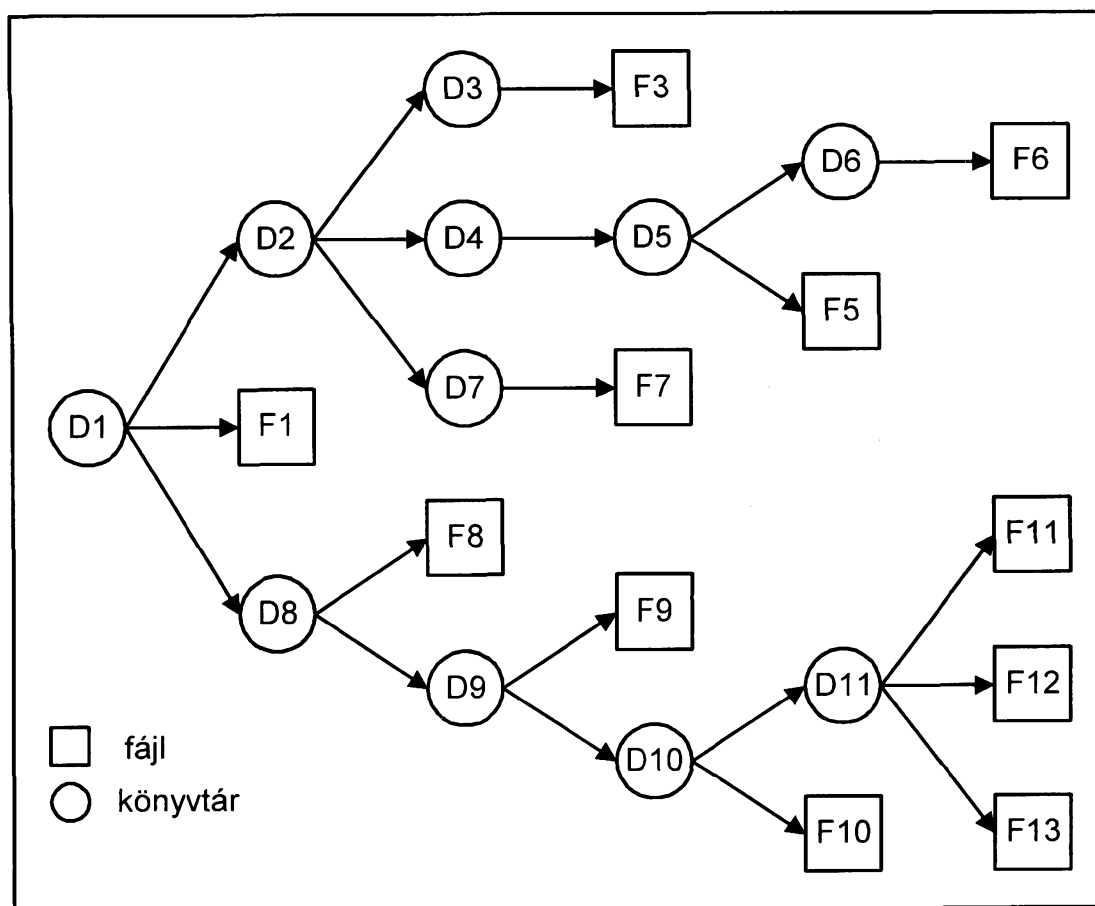
aj. rajz, 14 ábra



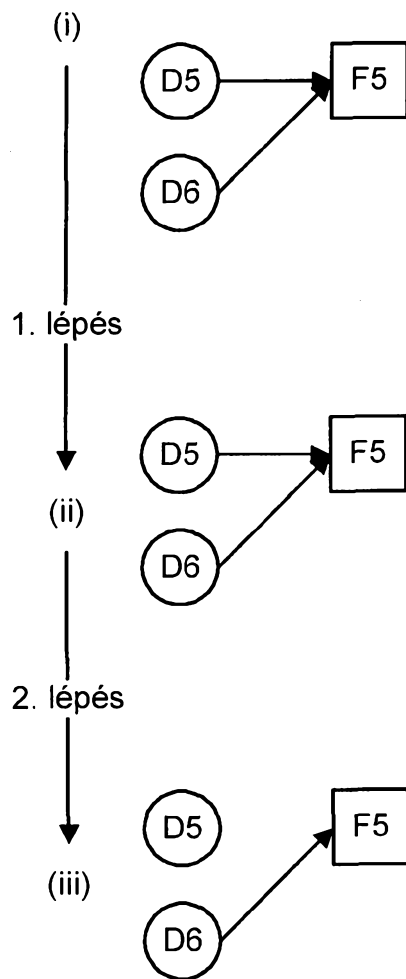
1. ÁBRA



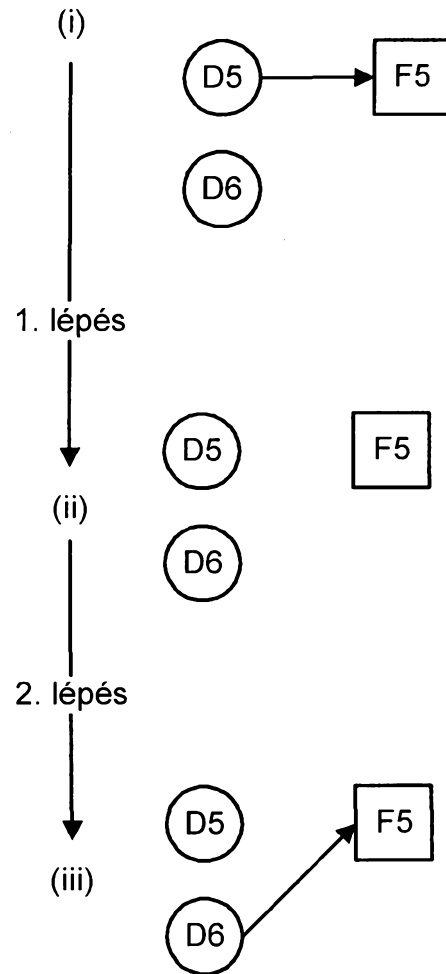
2. ÁBRA



3(A). ÁBRA

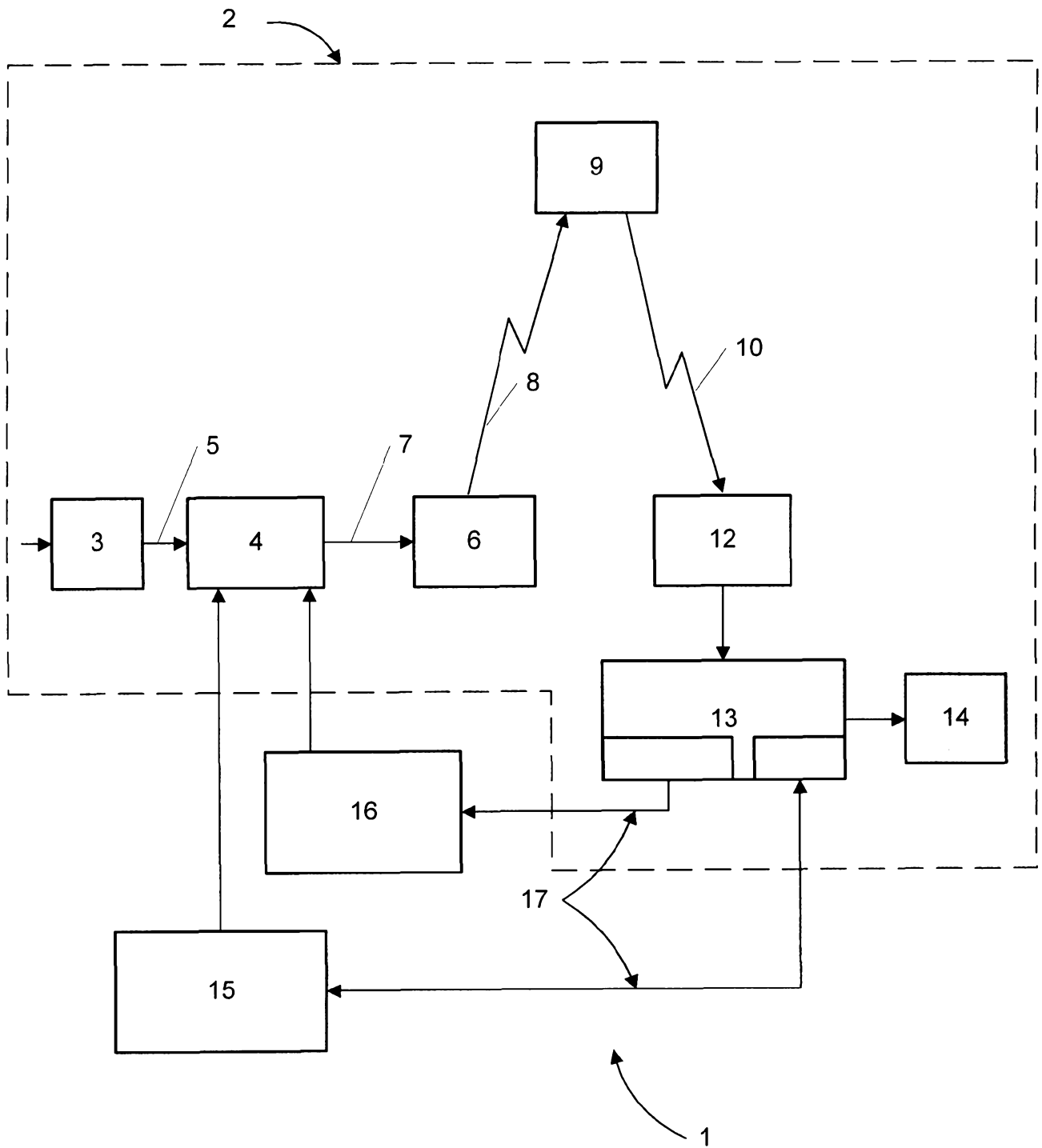


3(B). ÁBRA



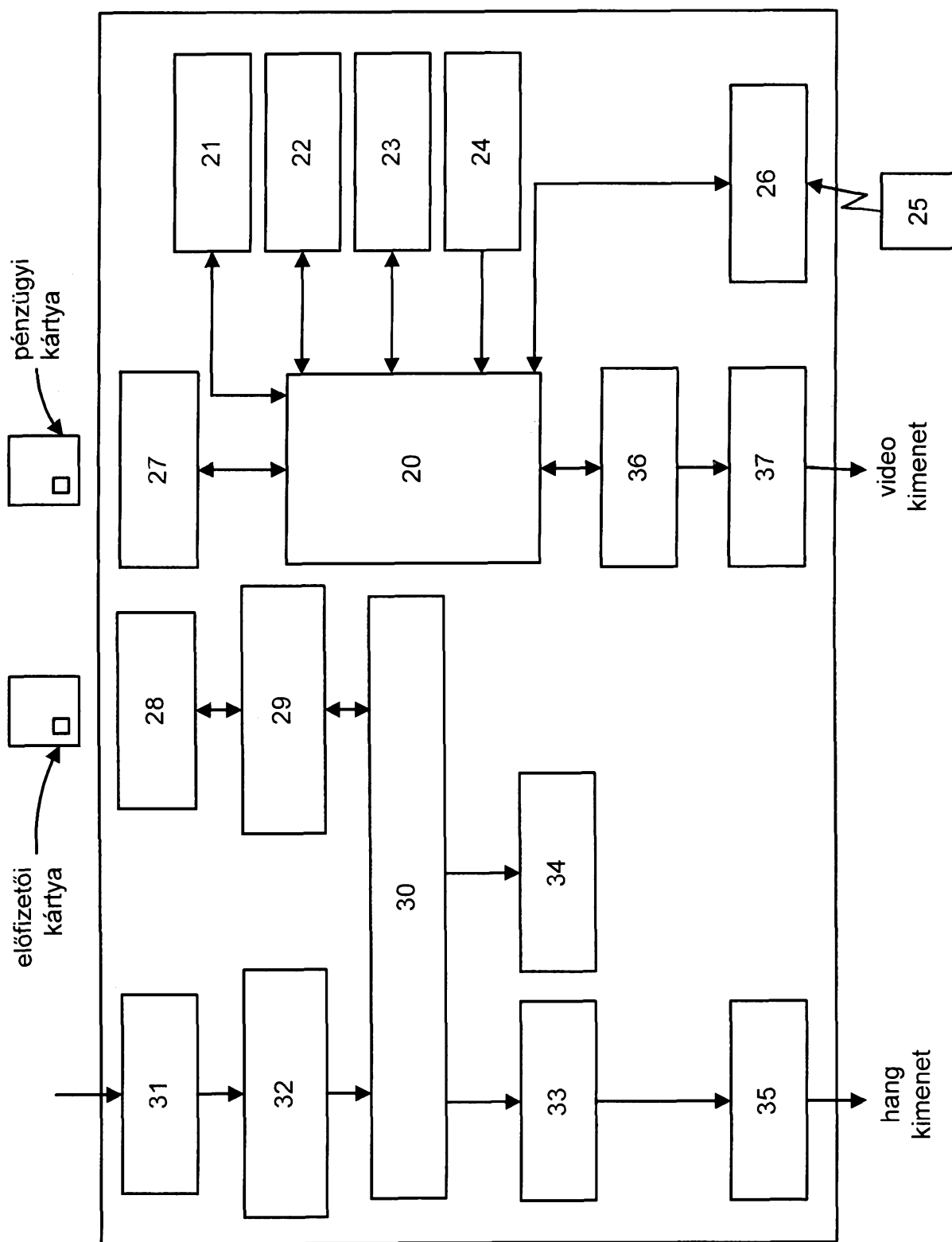
KÖZZÉTÉTELI
 PÉLDÁNY

4. ÁBRA



KÖZZÉTÉTELI PÉLDÁNY

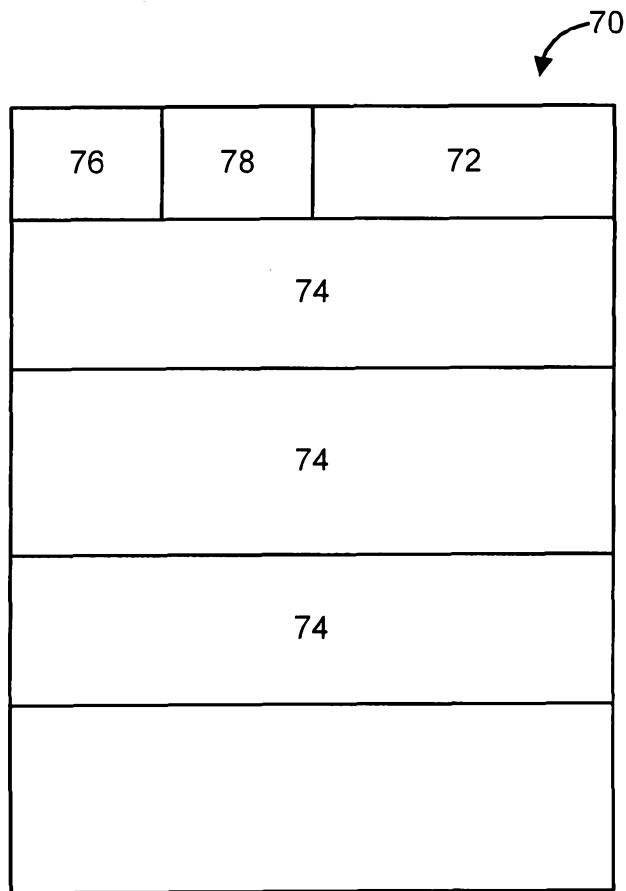
4/9



5. ÁBRA

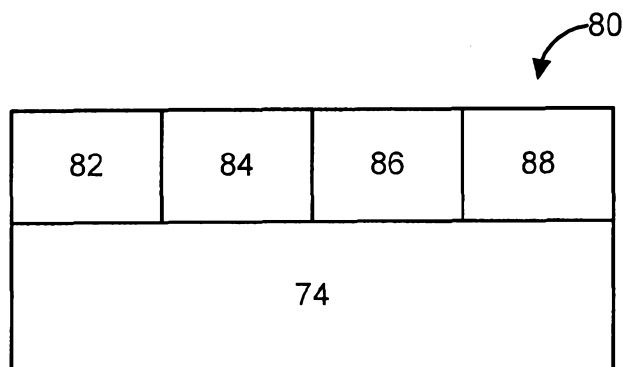
KÖZZÉTÉTELI
PÉLDÁNY

6. ÁBRA



76	78	72
74		
74		
74		

7. ÁBRA



82	84	86	88
74			

KÖZZÉTÉTELI
PÉLDÁNY

8. ÁBRA

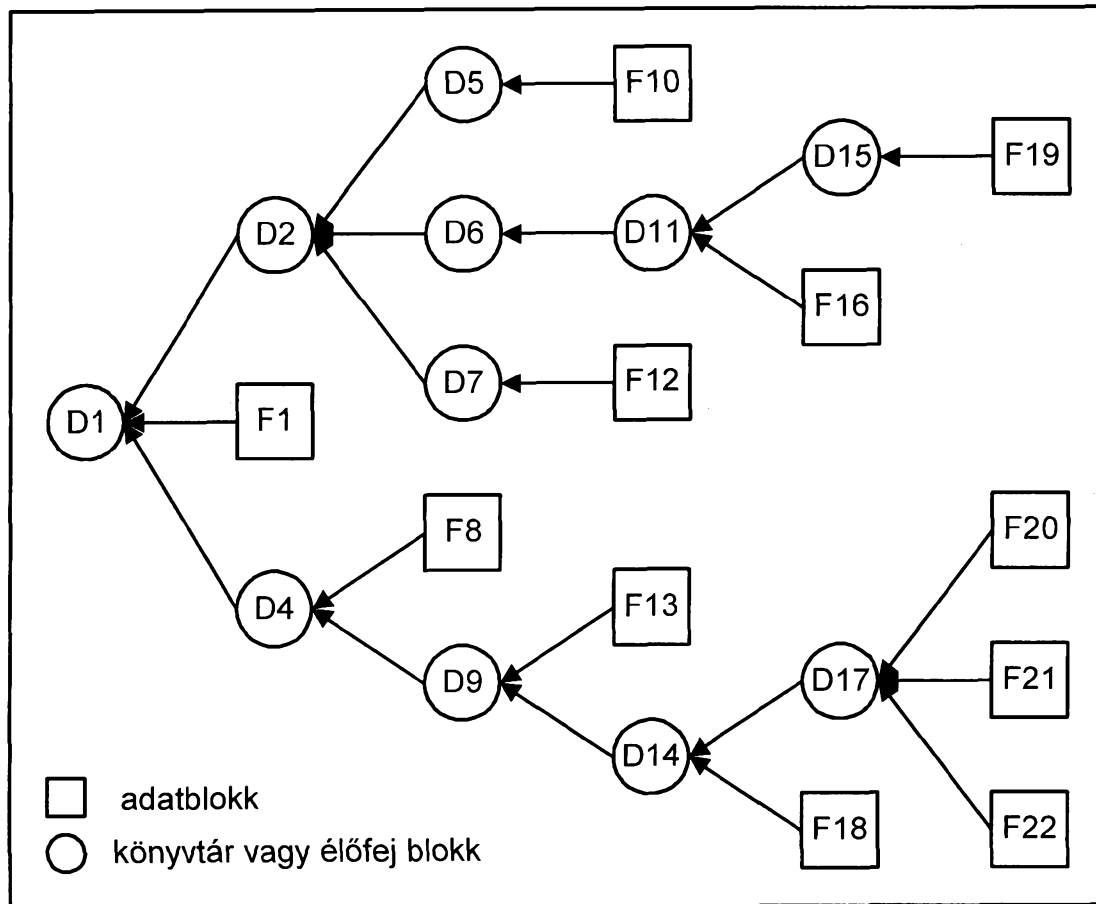
80	92	93	94	96	98	112	114
----	----	----	----	----	----	-----	-----

1 100	1 102	1 104	0 106	0 108	0 110
----------	----------	----------	----------	----------	----------

90

10. ÁBRA

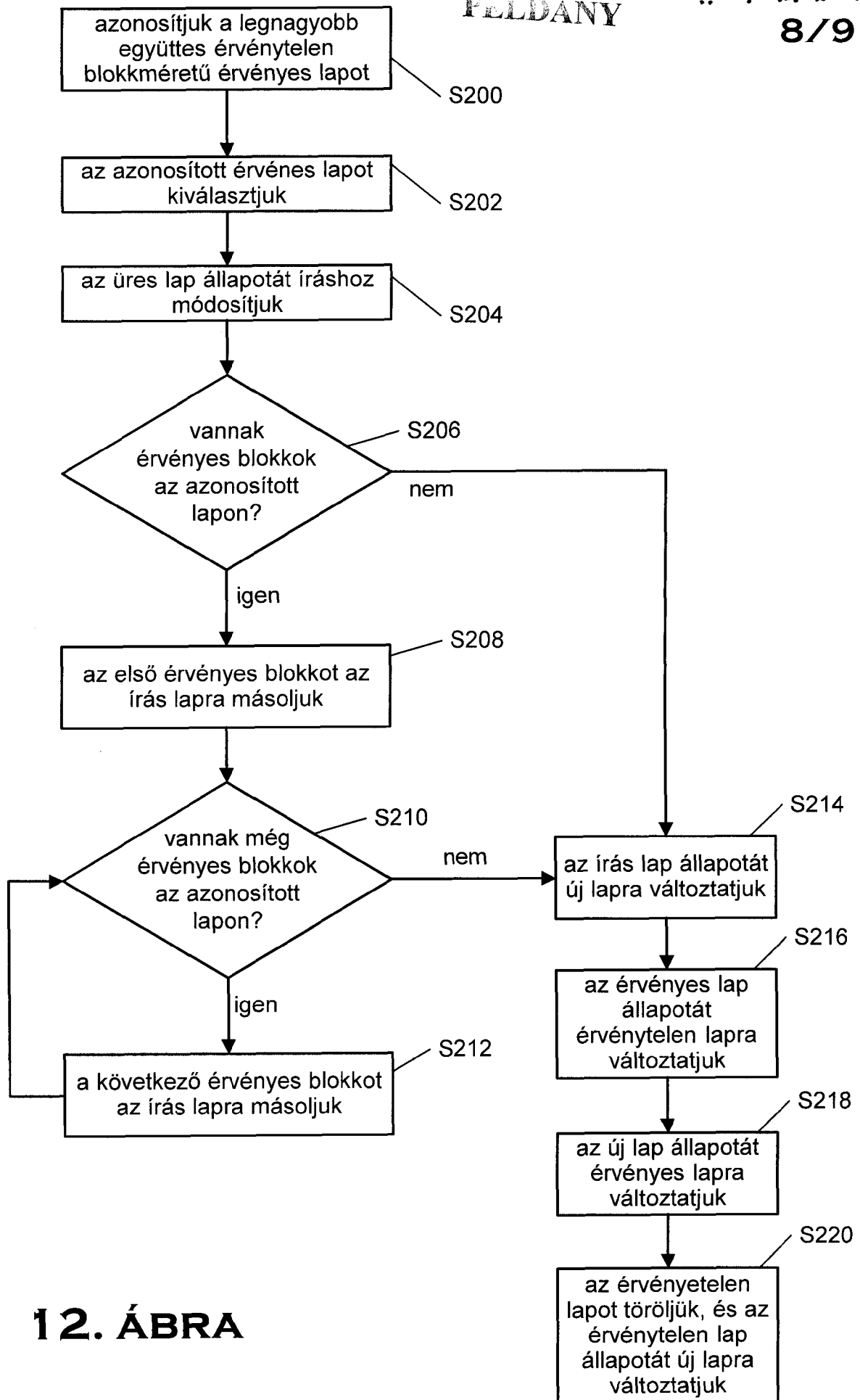
9. ÁBRA



58

	a blokk állapota az első lapon	a blokk állapota a második lapon
100 lépés	érvényes	szabad
102 lépés	érvényes	létrehozás
104 lépés	érvényes	írás
106 lépés	érvénytelen	új
108 lépés	érvénytelen	érvényes

11. ÁBRA



12. ÁBRA

előfej	
V	
V	

 $\geq$ $\geq$

élőfej
I
V
I
I
V
I

Élőfej

 $\geq$

 γ

13. ÁBRA