

Eljárás adatok letöltésére vevő és dekódoló berendezésbe, továbbá vevő és dekódoló berendezés, továbbá átviteli rendszer, ezek kombinációja, valamint jel adatok vevő és dekódoló berendezésbe töltésére

Kivonat

A találmány tárgya egyrészt egy eljárás adatok letöltésére egy adó és vevő berendezésbe, amelynek során a vevő és dekódoló berendezésben az adatokat magában foglaló bitfolyamot veszünk, a bitfolyamból az adatok betöltéséhez a vevő és dekódoló berendezésbe letöltünk egy ^{adat}betöltőt, majd az adatokat a bitfolyamból a letöltött adatbetöltő felhasználásával letöltjük.

A találmány tárgyát képezi továbbá egy vevő és dekódoló berendezés, amely adatokat magában foglaló bitfolyam vételére alkalmas vevőt, tárolóeszközt, valamint a bitfolyamból az adatoknak a vevő és dekódoló berendezésbe történő letöltéséhez szükséges adatbetöltőt a tárolóeszközbe letöltő letöltő eszközt tartalmaz.

A találmány tárgyát képezi továbbá egy átviteli rendszer, amelynek adatokat egy vevő és dekódoló berendezésbe letöltő legalább egy betöltőt magában foglaló bitfolyam, valamint a legalább egy adatbetöltővel társított adatok továbbítására szolgáló eszköze, valamint a legalább egy adatbetöltőt modulokra osztó, és a legalább egy adatbetöltővel társított adatokat megfelelő számú modulra osztó eszközzel rendelkezik a moduloknak a továbbító eszközzel történő továbbítására.

A találmány tárgyát képezi továbbá egy olyan jel, amelynek adatokat egy vevő és dekódoló berendezésbe letöltő legalább egy adatbetöltőt, továbbá a legalább egy adatbetöltővel társított adatot foglal magában, ahol a legalább egy adatbetöltő több modulra van felosztva, és a legalább egy adatbetöltővel társított adat megfelelő számú modulra van felosztva; valamint olyan rendszer, amely a javasolt vevő és dekódoló berendezésből átviteli rendszerből van összeépítve.

~~(ábra nélkül)~~

(—)

VÁRHEGYI C.

Eljárás adatok letöltésére vevő és dekódoló berendezésbe, továbbá vevő és dekódoló berendezés, továbbá átviteli rendszer, ezek kombinációja, valamint jel adatok vevő és dekódoló berendezésbe töltésére

A találmány tárgya eljárás adatnak vagy egy alkalmazás legalább egy részének, vagy több adatmodulnak a letöltésére egy vevő és dekódoló berendezésre, továbbá vevő és dekódoló berendezések az eljárások megvalósítására, valamint egy átviteli rendszer.

A "vevő és dekódoló berendezés" kifejezés alatt leírásunkban egyrészt olyan vevőberendezést értünk, amellyel kódolt vagy kódolatlan jeleket, például televízió és/vagy rádió jeleket vehetünk, amelyeket hagyományos műsorszórással vagy más módon juttatnak el a végfelhasználókhoz. Ez a kifejezés a vett jelek dekódolására szolgáló dekódoló berendezést is magában foglalja, ismert módon az ilyen vevő és dekódoló berendezések tartalmazhatnak a vevővel egybeépített, a vett jelek dekódolására szolgáló dekódert, amelyet akár egy "Set-Top-Box"-ban, vagyis a többi ismert vételi modultól különállóan, de azok közelében helyeznek el. Egy ilyen dekóder egy tőle fizikailag különálló vevővel együtt működhet, de tartalmazhat más, járulékos funkciókat is, például az Internet használatát jelentő és lehetővé tevő Web böngészőt, videomagnetofont, vagy televízió készüléket.

A digitális jelátviteli rendszerek fejlődése lehetővé tette ezen rendszerek más célokra történő felhasználását is. Egy ilyen lehetséges felhasználás interaktív kapcsolatot tesz lehetővé a végfelhasználóval. Leírásunkban a "digitális jelátviteli rendszer" kifejezés alatt minden olyan jel átvitelére alkalmas rendszert értünk, amellyel lehetséges például elsődlegesen audiovizuális vagy multimédiás digitális adatok átvitele vagy műsorszórása. Jóllehet, mint jeleztük, találmányunk elsősorban műsorszóró digitális televízió rendszerekben használható, a találmány hasonlóképpen eredményesen alkalmazható vezetékes telekommunikációs hálózatokban, multimédiás Internet alkalmazás céljára, zártláncú televízió rendszerekben, és így tovább.

Ennek egyik lehetséges megoldása, hogy egy alkalmazást futtatunk azon a vevő és dekódoló berendezésen, amelyen keresztül a televízió jelet vesszük. Az alkalmazás kódja permanens módon eltárolható a vevő és dekódoló berendezésben, ez azonban eléggé behatárolja a lehetőségeket, ezért előnyösen a vevő és dekódoló berendezést úgy célszerű kialakítani, hogy képes legyen egy kívánt alkalmazáshoz az ahhoz tartozó kódot letölteni. Ezen

a módon lényegesen több variációt hozhatunk létre, és az alkalmazásokat is bármikor frissíteni tudjuk a felhasználó részéről bármilyen beavatkozás vagy művelet igénye nélkül.

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.

Az MPEG táblázatok letöltéséhez szükséges szoftvert állandó jelleggel el kell tárolnunk a vevő és dekódoló berendezésben. Annak érdekében, hogy adatokat tölthessünk le, ilyen alkalmazás kódra, vagy egy futás alatti algoritmus (run time engine) frissített verziójára, továbbá összetett szoftverre van szükség, ami jellemző módon igen nagy memóriahelyet foglal el. Ezt a szoftvert azonban csupán szórványosan használjuk, ha egyáltalán használatára sor kerül, így igen nagy méretű memóriát foglalunk le egy olyan szoftver számára, amely igen hosszú időn keresztül adott esetben redundáns módon megtalálható.

A vevő és dekódoló berendezésben tárolt, a bitfolyamból adatok letöltéséhez szükséges szoftvert általános elnevezéssel betöltő programnak (bootstrap loader) nevezzük. A betöltő program célszerűen a legtöbb adattípus, adatforma letöltésére alkalmassá van téve, beleértve a bitfolyamból letöltött szoftver, például a vevő és dekódoló berendezés flash memória kötetében történő eltárolását is. Ily módon a betöltő program egyre inkább bizonyos "alapvető" szerkezetet kezd ölteni, minimális funkcionalitás mellett, hogy az összes lehetséges szoftver típust le tudjuk tölteni.

A betöltő programot általában a vevő és dekódoló berendezés ROM kötetében tároljuk kitörölhetetlen formában. Mivel a betöltő programot nem tudjuk módosítani, ha egyszer már beírtuk a ROM kötetbe, azokat a feldolgozási hibákat, amelyek adott esetben felléphetnek, és a betöltő program működését károsan befolyásolják, többé nem tudjuk korrigálni. Ezen túlmenően a betöltő program működési lehetőségei "be vannak betonozva", ha egyszer beírtuk a ROM kötetbe; így például nem tudjuk többé felfrissíteni oly módon, hogy csökkentjük a bitfolyamból az adatok letöltéséhez szükséges időtartamot. Ennek köszönhetően azo-

kat a szoftvereket, amelyek továbbfejlesztett, vagy új szerkezetűek, és ennek következtében a betöltő program számára ismeretlenek, a bitfolyamból nem tudjuk letölteni.

Ha a vevő és dekódoló berendezésben tárolt adatok egy része hibássá válik, a betöltő programot használhatjuk fel arra, hogy a meghibásodott adatok teljes és hibátlan verzióját letöltsük. Ha az adatoknak csupán csak nagyon kis része vált használhatatlanná, akkor a teljes verzió letöltése következtében jelentős időn keresztül töltjük le azt az adatrészt, amely nem vált hibássá.

Találmányunkkal erre, valamint más, vázolt problémákra keresünk megoldást.

A kitűzött feladatot egyrészt egy eljárással oldottuk meg, adatok letöltésére egy adó és vevő berendezésbe, . Ezt a találmány értelmében úgy fejlesztettük tovább, hogy a vevő és dekódoló berendezésben az adatokat magában foglaló bitfolyamot veszünk, a bitfolyamból az adatok betöltéséhez a vevő és dekódoló berendezésbe letöltünk egy betöltőt, majd az adatokat a bitfolyamból a letöltött adat betöltő felhasználásával letöltjük.

Egy lehetséges és előnyös megvalósítás értelmében a letöltött adat betöltő egy adat betöltő programot tartalmaz. Az adat betöltőnek legalább egy része, előnyösen legnagyobb része, vagy akár az egész adat betöltő natív kódként valósítható meg. Leírásunkban "natív kód" kifejezés alatt olyan hardverspecifikus kódot értünk, amely a vevő és dekódoló berendezés egy meghatározott hardver platformjára jellemző, vagy olyan kódra, amely nem interpretáló, és/vagy olyan kódra, amelyet a vevő és dekódoló berendezés mikroprocesszora közvetlenül végre tud hajtani. Így az egy vevő és dekódoló berendezés által letöltendő natív kód rész struktúrája a vevő és dekódoló berendezés hardver platformában felhasznált meghatározott elemektől fog függeni. Ez ellentétben áll az "interpretáló kóddal", amelyet a felhasználók más néven "p-kód" néven ismernek, amely igényli a vevő és dekódoló berendezésben eltárolt szoftveres interpretálást ahhoz, hogy a mikroprocesszor végre tudja hajtani, és amely éppen ezért szélesebb hardver platform tartományban működőképes. Az adat betöltő által letöltött adatok lehetnek natív kódú, p-kódú, vagy bármely más alkalmas formájú kódok, akár adattáblázatok.

A fentiek segítségével az adatoknak a bitfolyamból történő letöltéséhez szükséges adat betöltőt letöltjük az adatáramból, és - előnyösen csak ideiglenesen - a vevő és dekódoló berendezés RAM-jában tároljuk el. A letöltést követően a kívánt adatokat az adatbetöltő segítségével letölthetjük az adatáramból, és előnyösen a letöltött adat betöltőt a vevő és de-



kódoló berendezésből töröljük, miután az adatokat letöltöttük a bitfolyamból, így ha a korábban letöltött adat betöltő feladatát már betöltötte, a RAM tárolási kapacitását hatékonyan meg tudjuk növelni arra az időre, amíg nincs szükségünk ismét adatok letöltésére.

Természetesen az adat betöltő törlése nem feltétlenül szükséges az összes adat bitfolyamból történő letöltését követően. Ezzel ellentétben, az adat betöltőt ezt követően a vevő és dekódoló berendezés valamely nemfelejtő tárában, például flash memória kötetében is eltárolhatjuk. Ez elősegíti, hogy a vevő és dekódoló berendezéssel további adatokat tudjunk viszonylag gyorsan letölteni, hiszen az ehhez szükséges adat betöltőt nem kell ismét letöltenünk, mivel azt korábban eltároltuk, és ez adott esetben jelentősen lerövidítheti az adatok letöltésére fordított időt. Akár több különböző adat betöltőt is tárolhatunk egyidejűleg a vevő és dekódoló berendezésben.

Mivel az egy meghatározott adat letöltéséhez szükséges, és külön ahhoz megírt adat betöltőt a bitfolyamból szükség esetén a vevő és dekódoló berendezéssel tudjuk letölteni, biztosítottuk annak javíthatjuk üzemképességét, mivel frissített vagy átalakított adatokat tudunk letölteni és eltárolni a vevő és dekódoló berendezésben, mely adatok szerkezete vagy jellemzői lényegesen eltérnek azokatól az adatoktól, amelyet a program betöltővel tudunk csak letölteni. Egy további előnyös megvalósítás értelmében az adatok letöltését a letöltött adat betöltővel hajtjuk végre. Így az adatok letöltésének az idejére a program betöltőt hatékonyan, de csak ideiglenesen helyettesítjük, felcseréljük a letöltött adat betöltővel, lehetővé téve ezzel, hogy a vevő és dekódoló berendezésben egy felfrissített, vagy más módon feljavított adat betöltőt használhassunk.

Egy előnyös fogantatási mód értelmében a vevő és dekódoló berendezésben tárolt adatoknak csupán egy részét helyettesítjük a letöltött adat betöltővel letöltött megfelelő adatrésszel. Például, ha az eltárolt adatok egy része megsérül, vagy időszerűtlenné válik, annak az adatrésznek egy hibátlan, vagy frissített verzióját tölthetjük le a vevő és dekódoló berendezéssel, amelynek során az azt megelőzően letöltött adat betöltő "belevarázsolja" a letöltött adatrészt a korábban letöltött és a vevő és dekódoló berendezésben tárolt adatok közé, a megfelelő helyre. Így a letöltött adat betöltőnek nem kell a vevő és dekódoló berendezésben tárolt adat teljes változatát ismét letöltenie. Ez lényegesen lecsökkentheti az eltárolt adatok kijavításához vagy felfrissítéséhez szükséges időt, mivel az adatok hibátlan részét egyszerűen nem kell újra letöltenünk. Egy alternatív megoldás érdekében a vevő és

dekódoló berendezésben tárolt adatok egy részét például a letöltött adat betöltővel együtt kisugárzott adat szekció egy megfelelő részével is helyettesíthetjük.

Előnyösen a legalább egy adat betöltőt magában foglaló bitfolyam esetében az adatokat adó rendszerben a legalább egy adat betöltőt modulokra osztjuk, továbbá a legalább egy adat betöltőhöz az adatokat is megfelelő számú modulokra osztjuk, ahol minden meghatározott számú adatmodult meghatározott számú adat letöltő modullal társítunk.

Így a bitfolyam több adat betöltőt, valamint ezekhez tartozó társított adatot tartalmazhat. Ez lehetővé teszi, hogy a különböző hardver platformon működő vevő és dekódoló berendezések az adat betöltők és az azokhoz tartozó adatok számukra megfelelő verzióit tölthessék le. Az adatokat, például egy alkalmazást, szokásos módon és egyszerűen tudjuk modulokból is felépíteni, melyeket akár külön-külön is letölthetünk, és szükség esetén futtathatunk.

Ugyancsak előnyös a javasolt eljárás olyan foganatosítási módja, amelynek során az adó rendszerben a legalább egy adat betöltőhöz mindegyik modult olyan táblázatként alakítjuk ki, amelyeknek ugyanazt a táblázat azonosítót (TID), és különböző táblázat azonosító bővítményeket (TID bővítmények) adunk, és a legalább egy adatmodulhoz minden egyes adatmodult egy-egy olyan táblázatként alakítunk ki, amelyek táblázat azonosítója megegyezik a velük társított adat betöltő modulok táblázatainak a táblázat azonosítójával, de különböző táblázat azonosító bővítményeket tartalmaznak.

Előnyös, ha ennek során MPEG protokollt használunk, és ilyen esetben a letöltési lépések során MPEG táblázat modulokat tölthetünk le.

A találmány szerinti eljárás egy további előnyös foganatosítási módja értelmében a letöltés során az azonos táblázat azonosítójú modul táblázatokat töltjük le.

Ugyancsak előnyös, ha a táblázatok egy előre meghatározott táblázat azonosító bővítménytől eltérő táblázat azonosító bővítménnyel rendelkeznek, továbbá az adó rendszerben az ugyanolyan táblázat azonosítójú modulokhoz egy-egy könyvtár táblázatot állítunk elő, amelynek vagy amelyeknek az előre meghatározott táblázat azonosító bővítménnyel azonos táblázat azonosító bővítménye van, valamint azonos táblázat azonosítója van, és a könyvtár táblázat minden egyes modul részére tartalmazza a modul nevét, valamint az ahhoz tartozó táblázat azonosító bővítményt.

Ugyancsak előnyös a javasolt eljárás olyan fogantatosítási módja, amelynek során a vevő és dekódoló berendezésben az előre meghatározott táblázat azonosító bővítménnyel rendelkező egyik táblázat letöltésével egy könyvtár táblázatot letöltünk, a könyvtár táblázat tartalmából meghatározzuk a könyvtár táblázat azonosítójával azonos táblázat azonosítójú modul táblázatok táblázat azonosító bővítményét, és a letöltés során azokat a modul táblázatokat töltjük le, amelyek táblázat azonosítója megegyezik a letöltött könyvtár táblázat táblázat azonosítójával, és amelyek táblázat azonosító bővítményeit a letöltött könyvtár táblázatból határoztuk meg.

Ezen tulajdonságok révén a könyvtár táblázatot könnyen beazonosíthatjuk, hiszen annak egy meghatározott táblázat azonosító bővítménye lesz, és ha egyszer letöltöttük, lehetővé teheti a vevő és dekódoló berendezés számára, hogy az adat betöltő modul táblázatait azok táblázat azonosító bővítményei alapján azonosítsa be.

Ugyancsak előnyös a találmány értelmében, ha a az adó rendszerben előre meghatározott táblázat azonosítóval rendelkező, valamint egy vevő és dekódoló berendezés minden egyes verzió azonosítójához egy-egy hozzárendelt táblázat azonosítót tartalmazó könyvtár táblázatot állítunk elő.

Ez utóbbi esetben ugyancsak előnyös a javasolt eljárás olyan fogantatosítási módja, amelynek során a vevő és dekódoló berendezésben az előre meghatározott táblázat azonosítójú könyvtár táblázatot letöltjük, a vevő és dekódoló berendezés verzió azonosítóját meghatározzuk, valamint egy könyvtár táblázat letöltése során azt a könyvtár táblázatot töltjük le, amely a vevő és dekódoló berendezés verzió számával társított táblázat azonosítóval, és az előre meghatározott táblázat azonosító bővítménnyel rendelkezik.

Előnyös, ha a verzió azonosító a vevő és dekódoló berendezés verziójára jellemző kódot, valamint a vevő és dekódoló berendezés gyártójára jellemző kódot tartalmaz.

Természetesen a vevő és dekódoló berendezéseket számos különböző gyártó tervezi, gyártja és forgalmazza. Minden egyes gyártó több különböző vevő és dekódoló berendezés verziót gyárthat, és a mindennapi életben gyárt is. A vevő és dekódoló berendezések ezért igen különböző hardver felépítést mutatnak, jóllehet természetesen mindegyikük funkcionálisan ugyanannak a specifikációnak felel meg. Éppen ezért igen lényeges, hogy az adatok, például az alkalmazások minden egyes vevő és dekódoló berendezésben ugyanúgy vi-

selkedjenek, ugyanazt a hatást váltsák ki, és ezzel együtt az is, hogy egy adott vevő és dekódoló berendezés minden alkalmazást ugyanazon a helyes módon hajtson végre.

Annak biztosítása érdekében, hogy az adatok kompatibilisek egy vevő és dekódoló berendezés meghatározott verziójával, a bitfolyam a vevő és dekódoló berendezés minden egyes verzió azonosítója részére egy-egy adat betöltőt és ahhoz hozzárendelt adatot tartalmazhat, és az előre meghatározott táblázat azonosítójú könyvtár táblázat lehetővé teszi, hogy a vevő és dekódoló berendezés minden egyes verzió azonosítójához tartozó adat betöltő és adatmodulok táblázat azonosítóit könnyen beazonosíthassuk.

A javasolt eljárás egy további előnyös fogantatási módja értelmében, az adó rendszerben minden egyes elküldött könyvtár táblázatba egy arra jellemző könyvtár verzió azonosítót illesztünk, és a vevő és dekódoló berendezésben meghatározzuk, hogy az éppen elküldött könyvtár táblázat könyvtár verzió azonosítója fiatalabb-e, mint egy korábban letöltött, az éppen elküldött könyvtár táblázat táblázat azonosítójával azonos táblázat azonosítójú könyvtár táblázat, és amennyiben nem, úgy az adatok letöltését megszakítjuk.

A program betöltőt például egy alkalmazás arra utasíthatja, hogy meghatározott időközönként töltsen le a könyvtár táblázatot, hogy megállapítsa, hogy az azt megelőzően letöltött könyvtár táblázat könyvtár verzió azonosítója változott-e időközben. Ezzel biztosíthatjuk, hogy a vevő és dekódoló berendezés a bitfolyamból haladéktalanul le fog tölteni minden frissített adatot.

Annak érdekében, hogy a vevő és dekódoló berendezésben tárolt rezidens adatokat az ezekkel azonos kapott adatokkal felülírjuk, egy, a rezidens adatok frissítését igénylő alkalmazás számára lehetővé tesszük az adatok letöltésének a megszakítását, ha egy táblázat könyvtár azonos azzal, amelyet a rezidens adatok azt megelőző frissítése során használtunk.

Előnyösen a modul táblázatok legalább egyikét olyan szekciók sokaságaként alakítjuk ki, amelyeket külön-külön továbbítunk a bitfolyamban, és mindegyik szekció egy előre meghatározott részében a szekcióra vonatkozó azonosítót tartalmaz a táblázatban, valamint a szekciók számának jelzését tartalmazza egy táblázatban.

Az eljárás során az adó rendszerben a táblázatokat rendszeres időközönként leadhatjuk a bitfolyamban.

Ugyancsak előnyös a javasolt eljárás olyan fogantatási módja, amelynek során az adó rendszerben a bitfolyamba az adatokra vonatkozó adat verzió azonosítót illesztünk, és a vevő és dekódoló berendezésben meghatározzuk, hogy a vett adatok adatverzió azonosítója fiatalabb-e, mint az eltárolt adatok adatverzió azonosítója, és amennyiben igen, úgy az adatokat a bitfolyamból letöltjük.

Ezen tulajdonság révén a letöltést még azelőtt megszakíthatjuk, mielőtt sor kerülne a rezidens szoftver törlésére, és/vagy a vett adatok letöltését folytathatjuk, ha a vett adatok adatverzió azonosítója ugyanaz, mint a vevő és dekódoló berendezésben tárolt rezidens adatok adatverzió azonosítója.

Annak a meghatározását, hogy a vett vagy kapott adatok adatverzió azonosítója fiatalabb-e, mint a vevő és dekódoló berendezésben tárolt adatok adatverzió azonosítója, célszerűen annak a meghatározását követően hajtjuk végre, hogy egy éppen továbbított könyvtár táblázat adatverzió azonosítója fiatalabb-e, mint a korábban letöltött, és az éppen továbbított könyvtár táblázat táblázat azonosítójával megegyező táblázat azonosítójú könyvtár táblázat adatverzió azonosítója.

Egy további előnyös fogantatási mód értelmében a letöltött adat betöltővel úgy módosítjuk a vevő és dekódoló berendezésben az adat betöltő letöltése céljából tárolt eszközöket, hogy az adatokat a módosított letöltő eszközzel tudjuk letölteni. Így a letöltő eszközt kényelmesen módosítani tudjuk a bitfolyamból letöltött adat betöltővel, úgy, hogy például egy eltérő struktúrájú adatot is le tudunk tölteni a letöltő eszköz segítségével.

Előnyös a javasolt eljárás olyan fogantatási módja, amelynek során az adó oldalon egy második adat betöltőt küldünk el a bitfolyamba illesztve, és a vevő és dekódoló berendezésben a második adat betöltőt letöltjük, és az elsőként említett adat betöltőt és az adatokat letöltjük, amelynek során a második adat betöltővel hajtjuk végre az elsőként említett egyik adat betöltő, valamint az adatok letöltését.

Egy lehetséges megvalósításban a második adat betöltőt egy másik adat betöltő program bocsátja a rendelkezésre, és a második adat betöltőnek legalább egy része szintén natív kód formájában van megvalósítva.

Ez lehetővé teszi, hogy egy a bitfolyamból korábban letöltött, különböző adat betöltő alkalmazásával elkerüljük egy meghatározott adat betöltő letöltését. Így tehát nem lesz szükség egy adat betöltő letöltésére a bitfolyamból minden alkalommal, amikor friss, vagy mó-

dosított adatokat kívánunk letölteni, ha a korábban letöltött adat betöltő is alkalmas arra, hogy az adatokat olyan hatékonyan tudja letölteni, mint az új, le nem töltött adat betöltő. Ezzel jelentősen le tudjuk csökkenteni azt az időt, amely az új vagy módosított adatok bitfolyamból történő letöltéséhez szükséges. A második adat betöltő további funkciókat nyújthat az elsőként említett adat betöltőn túlmenően, például a második adat betöltő alkalmas lehet számítógép programok letöltésére is.

A kitűzött feladatot ezen túlmenően egy olyan vevő és dekódoló berendezéssel oldottuk meg, amely adatokat magában foglaló bitfolyam vételére alkalmas vevőt, tárolóeszközt, valamint a bitfolyamból az adatoknak a vevő és dekódoló berendezésbe történő letöltéséhez szükséges adat betöltőt a tárolóeszközbe letöltő letöltő eszközt tartalmaz.

Egy előnyös kiviteli alak értelmében a letöltő eszközt a vevő és dekódoló berendezésben tárolt indító program biztosítja.

Előnyös, ha a letöltött adat betöltőt az adatoknak a letöltését követően a bitfolyamból a vevő és dekódoló berendezésből törlő eszközt tartalmaz. A törlő eszközt egy központi processzor, valamint ahhoz tartozó, a vevő és dekódoló berendezésben tárolt program alkotja.

A vevő és dekódoló berendezést alkalmassá tehetjük táblázatok letöltésére. Ilyen esetben a letöltő eszköz úgy van kialakítva, hogy egy táblázat azonosítót és egy előre meghatározott táblázat azonosító bővítményt tartalmazó táblázat letöltésével egy könyvtár táblázatot töltsön le, továbbá, hogy meghatározza a könyvtár táblázat tartalmából a könyvtár táblázat táblázat azonosítójával azonos táblázat azonosítójú modul táblázatok táblázat azonosító bővítményeit, és hogy letöltse azokat a modul táblázatokat, amelyek táblázat azonosítója megegyezik a letöltött könyvtár táblázat táblázat azonosítójával, és táblázat azonosító bővítményeik a letöltött könyvtár táblázatból meghatározott táblázat azonosító bővítmények, és ezzel betöltse az adat betöltőt.

Egy további előnyös kiviteli alak értelmében a letöltő eszköz úgy van kialakítva, hogy egy előre meghatározott táblázat azonosítóval rendelkező könyvtár táblázatot töltsön le, amely egy vevő és dekódoló berendezés minden egyes verzió azonosítója részére egy ahhoz hozzárendelt táblázat azonosítót tartalmaz, továbbá, hogy meghatározza a vevő és dekódoló berendezés verzió azonosítóját, valamint, hogy a vevő és dekódoló berendezés verzió azo-

nosítójához tartozó táblázat azonosítójú, az előre meghatározott táblázat azonosító bővítéssel rendelkező könyvtár táblázatot töltsön le.

Ugyancsak előnyös a találmány értelmében, ha a letöltő eszköz úgy van kialakítva, hogy meghatározza, vajon az éppen továbbított könyvtár táblázat könyvtár verzió azonosítója fiatalabb-e egy korábban letöltött, az éppen továbbított könyvtár táblázat táblázat azonosítójával azonos táblázat azonosítójú könyvtár táblázat könyvtár verzió azonosítójánál, és amennyiben nem, úgy az adat betöltő letöltését megszakítsa.

A vevő és dekódoló berendezés ezen túlmenően párhuzamos porttal vagy soros porttal rendelkezhet, amely a legalább egy táblázatként formattált adatok vételére alkalmasan van kialakítva.

Ugyancsak előnyös, ha a letöltő eszköz a bitfolyamba illesztett adatokba beépített, az elsőként említett adat betöltő és az adatok letöltéséhez szükséges második adat betöltőt letöltő eszközként van kialakítva.

Előnyösen a letöltött adat betöltőt a bitfolyamból való letöltését követően eltároló nemfelejtő tárat tartalmaz.

Ez esetben előnyös és célszerű, ha a nemfelejtő tár a vevő és dekódoló berendezés flash memóriája.

Előnyös a találmány értelmében, ha a letöltött adat betöltő adatoknak a bitfolyamból történő letöltését végrehajtó betöltő.

Ebben az esetben különösen előnyös, ha a letöltött adat betöltő a vevő és dekódoló berendezésben tárolt adatoknak a letöltött megfelelő adattérrel csupán egy részét helyettesítő adat betöltő.

A kitűzött feladatot ezen túlmenően átviteli rendszerrel oldottuk meg, amely adatokat egy vevő és dekódoló berendezésbe letöltő legalább egy betöltőt magában foglaló bitfolyam, valamint a legalább egy adat betöltővel társított adatok továbbítására szolgáló eszköze, valamint a legalább egy adat betöltőt modulokra osztó, és a legalább egy adat betöltővel társított adatokat megfelelő számú modulra osztó eszközzel rendelkezik a moduloknak a továbbító eszközzel történő továbbítására.

Az átviteli rendszer egy előnyös kiviteli alakja értelmében a legalább egy adat betöltő minden egyes modulját egy-egy olyan táblázatként kialakító eszköze van, ahol a legalább egy

adat betöltő táblázatai azonos táblázat azonosítóval, és egymástól eltérő táblázat azonosító bővítményekkel rendelkeznek, valamint a legalább egy adat betöltővel társított adat modulok mindegyikét egy-egy olyan táblázatként kialakító eszköze van, ahol az adat modulok táblázatainak a táblázat azonosítója megegyezik a velük társított adat betöltő modulok táblázatainak a táblázat azonosítóival, és különböző táblázat azonosító bővítményekkel vannak ellátva.

A modulok formázását végző eszközként az adatszervert is felhasználhatjuk.

Ugyancsak előnyös a találmány értelmében, ha a táblázatoknak egy előre meghatározott táblázat azonosító bővítménytől eltérő, különböző táblázat azonosító bővítményük van, továbbá a rendszer az azonos táblázat azonosítójú modul vagy modulok mindegyike számára egy-egy könyvtár táblázatot előállító eszközzel rendelkezik, ahol minden egyes könyvtár táblázatnak ugyanaz a táblázat azonosítója, valamint az előre meghatározott táblázat azonosító bővítménye van, és a könyvtár minden egyes modul részére tartalmazza a modul nevét és táblázat azonosító bővítményét. Egy további lehetséges és előnyös kiviteli alak értelmében előre meghatározott táblázat azonosítójú, és egy vevő és dekódoló berendezés minden egyes verzió azonosítója számára egy azzal társított táblázat azonosítót tartalmazó könyvtár táblázatot előállító eszközzel van ellátva.

Ugyancsak előnyös, ha az átviteli rendszer minden egyes továbbított táblázatba egy arra vonatkozó verzió azonosítót beépítő eszközt tartalmaz.

A fent vázolt eszközök mindegyikét szokásos és ismert módon az adatszerverrel is megvalósíthatjuk.

Ugyancsak előnyös a javasolt átviteli rendszer olyan kiviteli alakja, ahol a legalább egy adat betöltő a vevő és dekódoló berendezés hardver felépítésére jellemző kód alakjában van megvalósítva.

A kitűzött feladatot ezen túlmenően egy jellel oldottuk meg, amely a találmány értelmében adatokat egy vevő és dekódoló berendezésbe letöltő legalább egy adat betöltőt, továbbá a legalább egy adat betöltővel társított adatot foglal magában, ahol a legalább egy adat betöltő több modulra van felosztva, és a legalább egy adat betöltővel társított adat megfelelő számú modulra van felosztva.

A találmányt az alábbiakban a csatolt rajz segítségével ismertetjük részletesebben, amelyen az

- 1. ábra egy digitális televízió rendszer felépítésének elvi vázlata, a
- 2. ábra az 1. ábra szerinti digitális televízió rendszer interaktív rendszerének felépítését mutatja, a
- 3. ábra az 1. és 2. ábrán látható digitális televízió rendszer részét képező vevő és dekódoló berendezés interfészeinek vázlata, a
- 4. ábrán a digitális televízió rendszerben használt távirányító egy lehetséges kialakításának vázlatos rajza látható, az
- 5. ábra egy interaktív vevő és dekódoló berendezés memóriájába letöltött modulon belül a fájlok elrendezését mutatja, a
- 6. ábra egy MPEG jeláram különböző összetevői közötti kölcsönhatást mutat be, a
- 7. ábrán azt vázoltuk, miképpen építhető fel egy alkalmazás olyan modulokból illetőleg táblázatokból, amelyek különböző szekciókból vannak felépítve, a
- 8. ábra egy MPEG táblázat jogosítását mutatja be, a
- 9. ábra a digitális televízió rendszer vevő és dekódoló berendezésében lévő memória különböző területeit vázolja, a
- 10. ábrán egy paraméter mező látható, a
- 11. ábrán egy hardver könyvtár táblázat látható, a
- 12. ábrán egy adat betöltő könyvtár táblázat látható, és a
- 13. ábra az adatok letöltésének műveletét mutatja be.

Az 1. ábrán a találmány szerinti 1000 digitális televízió műsorszóró és vételi rendszer vázlatát tüntettük fel az áttekintés elősegítésére. Az 1000 digitális televízió műsorszóró hagyományosan felépített 2000 digitális televízió rendszert foglal magában, amely az ismert MPEG-2 tömörítő rendszert használja tömörített digitális jelek kibocsátásához. Egy kicsit részletesebben, a 2002 MPEG-2 tömörítő, amely egy műsorszóró központban helyezkedik el, digitális adatáramot kap (amely rendszerint video jel adatáram). A 2002 MPEG-2 tömörítő 2006 vonalon keresztül 2004 multiplexer és bitsorkódolóhoz van csatlakoztatva. A 2004 multiplexer és bitsorkódoló számos további bemenőjelet kap, ezekből egy vagy több

szállítási adatáramot állít össze, és a tömörített digitális jelet a műsorszóró központban lévő 2008 adóberendezéshez továbbítja egy további 2010 vonalon keresztül, amely természetesen számos módon megvalósítható, beleértve a szokásos távközlési vezetékes kapcsolatok is. A 2008 adóberendezés egy 2012 földi állomástól műholdra irányuló adatátviteli kapcsolat segítségével elektromágneses jeleket továbbít 2014 transzponderhez, amely azt elektronikusan feldolgozza, és 2016 műholdról földi állomás felé irányuló adatátviteli kapcsolaton át földi 2018 vevőhöz továbbítja, mely 2018 vevő általában egy végfelhasználó tulajdonában álló, vagy általa bérelt parabola antenna. A 2018 vevővel fogadott jelek szintén a végfelhasználó tulajdonában álló vagy általa bérelt integrált 2020 vevő és dekódoló berendezésbe kerülnek, amely a tömörített MPEG-2 jelet olyan televízió jellé alakítja vissza, amely minden további nélkül használható 2022 televízió készülékben.

A 2004 multiplexer és bitsorkódolóhoz és a 2020 vevő és dekódoló berendezéshez egy 3000 feltételes hozzáférési rendszer kapcsolódik, amely részben a műsorszóró központban, részben a 2020 vevő és dekódoló berendezésben van kialakítva. Ez a 3000 feltételes hozzáférési rendszer lehetővé teszi a felhasználó 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 2020 vevő és dekódoló berendezésbe, és a kettő együttes használatával a végfelhasználó vagy előfizetéses módon, vagy fizetős módon (pontosabban "nézd-és-fizess", pay per view módon) műsorok által megtestesülő eseményeket vásárolhat.

Az ugyancsak a 2004 multiplexer és bitsorkódolóhoz és a 2020 vevő és dekódoló berendezéshez kapcsolódó 4000 interaktív rendszer szintén egyrészt a műsorszóró központban, másrészt a 2020 vevő és dekódoló berendezésben van kialakítva, és lehetővé teszi a végfelhasználó számára, hogy számos alkalmazással egy modemes 4002 válaszcsatormán keresztül interaktív módon álljon kapcsolatban.

A 2. ábrán a találmány szerinti 1000 digitális televízió műsorszóró 4000 interaktív rendszerének általános felépítését vázoltuk.

A 4000 interaktív rendszer például lehetővé teszi egy végfelhasználó számára, hogy különböző árucikkeket vásároljon képernyőn megjelenő katalógusokból, helyi híreket és időjárás térképeket "fogyasszon" kívánsága szerint és 2022 televízió készüléke segítségével játszhasson.

A 4000 interaktív rendszer négy fő részből áll:

- a) a műsorszóró központban vagy valahol máshol telepített 4004 alkotói eszköztárat, amellyel a műsorszóró szolgáltató alkalmazásokat állíthat elő, fejleszthet, tesztelhet és követhet nyomon;
- b) a műsorszóró központban elhelyezett 4006 alkalmazás- és adatszervert, amely a 4004 alkotói eszköztárral áll kapcsolatban és lehetővé teszi a műsorszóró szolgáltató számára, hogy alkalmazásokat és adatokat készítsen elő, jogosítsa fel és alakítsa, amelyeket aztán a 2004 multiplexer és bitsorkódolóhoz továbbíthat, hogy beillesse az MPEG-2 szállító jeláramba (jellemzően annak privát szekciójába), amelyet a végfelhasználónak sugároz ki;
- c) egy 4008 végrehajtás alatt futó főalgoritmust magában foglaló virtuális gép, amely a végfelhasználó által bérelt, vagy a tulajdonát képező 2020 vevő és dekódoló berendezésben installált végrehajtható kód, és amely lehetővé teszi a végfelhasználó számára, hogy a 2020 vevő és dekódoló berendezés memóriájába végrehajtás céljából alkalmazásokat vegyen jogosultságokat szerezzen, dekomprimáljon és betöltsön. A 4008 végrehajtás alatt futó főalgoritmust magában foglaló virtuális gép rezidens általános célú alkalmazásokat is futtat, és független a hardvertől és az operációs rendszertől.
- d) A 2020 vevő és dekódoló berendezés és a 4006 alkalmazás- és adatszerver között elhelyezkedő 4002 modemes válaszcsatorna, amely lehetővé teszi a 4006 alkalmazás- és adatszervert a jelek segítségével arra utasítani, hogy a végfelhasználói kérelemre az MPEG-2 jeláramba adatokat és alkalmazásokat illesszen.

A 4000 interaktív rendszer olyan "alkalmazások" felhasználásával működik, amelyek a 2020 vevő és dekódoló berendezés valamint az abban lévő különböző eszközök egyes funkcióit vezérlik. Az alkalmazások a 4008 végrehajtás alatt futó főalgoritmust magában foglaló virtuális gépben "erőforrás fájlként" vannak jelen. Egy "modul" nem más, mint ilyen erőforrás fájlból és adatokból álló készlet. A 2020 vevő és dekódoló berendezés "memória kapacitása" az említett modulok számára rendelkezésre álló tárolóhely. A modulokat a 2020 vevő és dekódoló berendezésbe az MPEG-2 jeláramból tölthetjük le.

Az adatok letöltésére a 2020 vevő és dekódoló berendezés fizikai interfészeit használjuk. A 3. ábrán látható, hogy a 2020 vevő és dekódoló berendezés például hat különböző letöltő eszközt tartalmaz: 40028 MPEG áram tuner, 4030 soros interfészt, 4032 párhuzamos interfészt, és két 4036 kártyaolvasót.

Leírásunkban alkalmazás alatt egy számítógépi kód készletet értünk, amellyel előnyösen a 2020 vevő és dekódoló berendezés magas szintű funkcióit vezéreljük. Például, ha a végfelhasználó 2026 távirányítóját (mint az a 4. ábrán jobban megfigyelhető), pontosabban annak fókuszált sugarát ráirányítja a 2022 televízió készülék képernyőjén látható gomb objektumra, és megnyomja a 2026 érvényesítés gombját, azzal lefuttatja a képernyőn látható nyomógombbal 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 2020 vevő és dekódoló berendezés ROM-jában (vagy Flash, vagy más nem-felejtő tárolójában) vannak eltárolva, vagy a 2020 vevő és dekódoló berendezés RAM-jába (vagy Flash-ROM-jába) kerülnek kisugárzás és letöltés útján.

Az alábbiakban nem kizárólagosan, csupán példaképpen felsorolunk néhány lehetséges alkalmazást:

- Inicializáló alkalmazás. A 2020 vevő és dekódoló berendezés egy rezidens inicializáló alkalmazással van ellátva, amely a 2020 vevő és dekódoló berendezést az MPEG-2 környezetben azonnal működőképesé tevő modulok illeszthető, adaptálható gyűjteménye (ezt a kifejezést később még részletezzük). Az alkalmazás kulcsfontosságú jellemzőket nyújt, amelyet szükség esetén a műsorszóró szolgáltató módosítani tud. Ezen kívül interfészt képez a rezidens alkalmazások valamint a letöltött alkalmazások között.
- Indító alkalmazás. Az indító alkalmazás teszi lehetővé bármely alkalmazás számára, legyen az letöltött vagy rezidens alkalmazás, hogy fusson a 2020 vevő és dekódoló berendezésen. Ez az alkalmazás egy betöltő rutinként szolgál, amely egy szolgáltatás megérkezésekor kerül végrehajtásra, annak érdekében, hogy az alkalmazás elindulhasson. Az indító alkalmazás a 2020 vevő és dekódoló berendezés RAM-jába kerül letöltésre, így könnyen frissíthető. Az indító alkalmazás úgy konfigurálható, hogy bármely csatornán rendelkezésre álló interaktív alkalmazásokat tudunk kiválasztani és futtatni, vagy közvetlenül a letöltést követően, vagy egy előzetes letöltés utáni későbbi időpontban. Ez utóbbi esetben az alkalmazás a 2020 vevő és dekódoló berendezés 2024 memóriájába töltődik, és az indító alkalmazás igény szerint fogja aktiválni.

- Programkalauz. A programkalauz egy olyan interaktív alkalmazás, amely a programozásra nyújt teljes körű információt. Például információt adhat mondjuk egy digitális televízió csomag egyes csatornáin egy egész hét során sugárzásra kerülő televíziós műsorokról. Ha a végfelhasználó a 2026 távirányítón megnyom egy megfelelő gombot, a 2022 televízió készüléken éppen látható esemény képernyőre rányúló további képernyőt kap, amely nem más, mint egy olyan böngésző, amely információt nyújt a digitális televízió csomag egyes csatornáinak éppen futó aktuális és rákövetkező műsor eseményeiről. Ha a végfelhasználó a 2026 távirányítón most egy másik gombot nyom meg, akkor egy olyan alkalmazást indít el, amely kiad neki egy információs listát egy teljes hét eseményeiről. A végfelhasználó ilyen módon egyszerű és testreszabott feltételekkel keresheti és válogathatja ki a számára érdekes eseményeket, de éppígy közvetlenül hozzáférhet egy kiválasztott csatornához is.
- Nézd-és-fizess alkalmazás. A nézd-és-fizess alkalmazás egy olyan interaktív szolgáltatás, amely a digitális televízió csomag minden egyes nézd-és-fizess csatornáján rendelkezésre áll a 3000 feltételes hozzáférési rendszerrel együttműködésben. A végfelhasználó ezt az alkalmazást egy televízió műsorfűzet vagy csatorna böngésző felhasználásával tudja indítani. Ezen túlmenően az alkalmazás automatikusan megindul abban a pillanatban, amint a nézd-és-fizess csatornán a műsorszóró szolgáltató egy nézd-és-fizess eseményt megindít. A végfelhasználó ekkor ezt az aktuális eseményt, azaz műsort meg tudja venni, vagy a saját gyermek 3020 programozható csipkártyáján, vagy a 3022 kommunikációs szerveren keresztül (modem, telefon és DTMF kódok, vagy MINITEL vagy hasonló rendszer használatával, természetesen). Ez az alkalmazás vagy rezidensként a 2020 vevő és dekódoló berendezés ROM-jában található, vagy pedig annak RAM-jába letölthető.
- Internet böngésző alkalmazás. Az Internet böngésző alkalmazás egy megvalósítási példájában a végfelhasználótól kiinduló utasításokat, mint például egy meghatározott URL-lel rendelkező Web-lap megtekintésére vonatkozó kérelmet a felhasználó a 2026 távirányító segítségével tud bevinni, és a kérelmét tartalmazó adat a 4002 modemes vonalon keresztül jut a 4006 alkalmazás- és adatszerverhez. A kért Web-lapot a műsorszóró központból induló jelátvitelbe beépítik és azt a 2020 vevő és dekódoló berendezés a 2012 földi állomáson, a 2014 transzponderen és a 2016 műholdon keresztül meg-

kapja, majd továbbítja a 2022 televízió készülékhez, amelynek képernyőjén a végfelhasználó a Web-lapot megtekintheti.

Az alkalmazásokat a 2020 vevő és dekódoló berendezés memóriahelyein tároljuk és ezek erőforrás fájlakként és adatokként jelennek meg. Az erőforrás fájlok grafikus objektum leíró egység fájlkból, változó blokk egység fájlkból, utasítás sorrend fájlkból, alkalmazás fájlkból és adatfájlkból állnak.

A 4014 grafikus objektum leíró egység fájl a különböző képernyőket írja le, vagyis tulajdonképpen az alkalmazás ember-gép közötti interfészét. A 4016 változó blokk egység fájl az alkalmazás által megszólított és "kezelt" adat struktúrákat írja le, a 4018 utasítás szekvencia fájl az alkalmazások feldolgozási folyamatait írja le, míg a 4012 alkalmazás fájl az alkalmazásokhoz tartozó egyes belépési pontokat definiálja.

Az ily módon felépített alkalmazások adat fájlakat, mint amilyenek az ikon könyvtár fájlok, képfájlok, betűtípus fájlok, színtáblázat fájlok és tiszta szöveges (ASCII) fájlok, képesek használni. Egy interaktív alkalmazás bemenetek és/vagy kimenetek felhasználásával online adatokat is nyerhet.

A 4008 végrehajtás alatt futó főalgoritmust magában foglaló virtuális gép csupán azokat az erőforrás fájlakat hívja be a memóriába, amelyekre egy adott időpontban éppen szükség van. Ezeket az erőforrás fájlakat a 4008 végrehajtás alatt futó főalgoritmust magában foglaló virtuális gép a 4014 grafikus objektum leíró egység fájlkból, 4018 utasítás szekvencia fájlkból, 4012 alkalmazás fájlkból és 4016 változó blokk egység fájlkból olvassa ki, amelyek a memóriában vannak eltárolva, méghozzá egy a modulok betöltésére szolgáló eljárás meghívását követően, és az erőforrás fájlok mindaddig ott maradnak, amíg egy, a modulok kitöltésére, eltávolítására szolgáló eljárást meg nem hívunk.

Az 5. ábrán látható, hogy egy 4010 modul, mint amilyen a részletesebben is bemutatásra kerülő vásárlási modul, erőforrás fájlok és adatok készleteként értelmezendő, amely az alábbiakat tartalmazza:

- egy egyetlen 4012 alkalmazás fájlt,
- meghatározatlan számú 4014 grafikus objektum leíró egység fájlt,
- meghatározatlan számú 4016 változó blokk egység fájlt,
- meghatározatlan számú 4018 utasítás szekvencia fájlt, és

- ahol szükséges 4020 adat fájlokat, például ikon könyvtár fájlokat, képfájlokat, betű-típus fájlokat, színtáblázat fájlokat és ASCII szöveges fájlokat.

A 4010 modulokat használó koncepció együtt egy kód csupán kis részeinek a letöltését célzó koncepcióval lehetővé teszi a különböző alkalmazások egyszerű kifejlesztését vagy továbbfejlesztését. Ezeket a kis kódrészeket a 2020 vevő és dekódoló berendezés FLASH memóriájába akár rezidens szoftverként is le tudjuk tölteni, vagy kisugározzuk őket annak érdekében, hogy csak akkor töltődjenek le egy 2020 vevő és dekódoló berendezés RAM-jába, ha a végfelhasználónak, ezen keresztül a 2020 vevő és dekódoló berendezésnek valóban szüksége van rájuk.

MPEG áram esetében egyetlen 4010 modult egyetlen MPEG táblázatban továbbítunk. Abban az esetben, ha modulokat továbbítunk a 4028 MPEG áram tunerhez, a hosszú MPEG-2 formátumot használjuk, hosszú élőfejjel és CRC kóddal. Ugyanez a helyzet akkor is, ha az öt másik interfészt használjuk (vagyis a 4030 soros interfészt, a 4032 párhuzamos interfészt, a 4034 modemet és a két 4036 kártyaolvasót), azzal az eltéréssel, hogy itt a "rövid" MPEG-2 formátumot alkalmazzuk, rövidebb élőfejjel, és CRC kód nélkül.

Áttérve a 6. ábrára, mint ismert, az MPEG-2 bitfolyam 10 program hozzáférési táblázatot foglal magában, amelynek 0 csomag azonosítója van. A program hozzáférési táblázat referenciákat tartalmaz egy nagyobb számú program 12 program térkép táblázatainak a csomag azonosítóihoz. Minden egyes program térkép táblázat egy referenciát tartalmaz 14 audio MPEG táblázatok és 16 video MPEG táblázatok adatáramjainak a csomag azonosítóihoz egy adott programhoz. Egy nullás csomag azonosítóval rendelkező csomag, vagyis maga a 10 program hozzáférési táblázat alkotja a belépési pontot az összes MPEG hozzáférés számára.

Alkalmazások, valamint az alkalmazásokhoz szükséges adatok letöltéséhez két új adatáram típust definiálunk, és a releváns program térkép táblázat referenciákat tartalmaz 18 alkalmazás MPEG táblázatok (vagy azok szekciói) áramainak, valamint 20 adat MPEG táblázatok (vagy azok szekciói) áramainak a csomag azonosítóihoz.

Áttérve a 7. ábrára látható, hogy egy 22 alkalmazás letöltéséhez a 22 alkalmazást egy-egy MPEG táblázatként kialakított 24 modulokra osztottuk fel, melyek közül néhányat egyetlen 18 szekció alkot, a többit pedig több 18 szekció képezi. Egy jellegzetes 18 szekciónak olyan 26 élőfeje van, amely egy 1 bájtos 28 táblázat azonosítót foglal magában, továbbá az

adott szekciónak a táblázatban elfoglalt 30 szekció számát, a szekciók 32 teljes számát a táblázatban, valamint egy 2 bájtos 34 táblázat azonosító bővítményt. Minden egyes 18 szekciónak ezen kívül 36 adatrésze, valamint 38 CRC része van. Egy meghatározott 24 modul esetében a 24 modult alkotó összes 18 szekciónak ugyanaz a 28 táblázat azonosítója, és ugyanaz a 34 táblázat azonosító bővítménye. Egy meghatározott 22 alkalmazás esetében az azt alkotó összes 24 modulnak ugyanaz lesz a 28 táblázat azonosítója, de különböző 34 táblázat azonosító bővítményekkel fognak rendelkezni.

Az alábbiakban a 8. ábra segítségével egy MPEG táblázat feljogosítását mutatjuk be. A 40 MPEG táblázat 42 adatokat, 44 kulcs azonosítót és 46 rejtjelezett területet tartalmaz. A 42 adatok jellemzően 26 élőfejet, 28 táblázat azonosítót, 34 táblázat azonosító bővítményt és 36 adatrészt foglalnak magukban. A 44 kulcs azonosító a blokk kódolásához használandó meghatározott privát kulcs 1 bájtos azonosítóját tartalmazza. A 46 rejtjelezett terület 96 bájt méretű adatblokkot tartalmaz, amelynek első 48 bájtja 0 értékű. Egy 16 bájt nagyságú 50 aláírás az első bájtot követően tipikusan 0-31 bájtnyi eltolással kezdődik. Az 50 aláírást az ismert MD5 aláírás létrehozó eljárással állítjuk elő a 42 adatokból. Az első 48 bájt és az 50 aláírás közé 52 áladatot illesztettünk és a blokkot egy ismert kódoló eljárással kódoltuk, valamint felhasználtuk a 44 kulcs azonosító által megadott privát kulcsot is.

Ha több 40 MPEG táblázatot kell jogosítással ellátnunk, akkor a vivőjelbe a 40 MPEG táblázatok nevét, valamint a 40 MPEG táblázatok 50 aláírásait felsoroló könyvtárat helyezzünk el. MPEG áram esetében ezt a könyvtárat egyetlen 40 MPEG táblázatban továbbítjuk, amelynek a 34 táblázat azonosító bővítménye jellemző módon 0 értékű. A könyvtár táblázatot a fent vázolt módon hitelesítjük. Miután a könyvtárat letöltöttük a hordozójelből, lehetőség nyílik arra az alkalmazás számára, hogy a könyvtárban felsorolt egy vagy több 40 MPEG táblázatot letöltsön.

A következőkben a 2020 vevő és dekódoló berendezés működését mutatjuk be, egy alkalmazás letöltése során az aláírásokkal és a dekódolással kapcsolatos tevékenységére koncentrálva. A 9. ábrán megfigyelhető, hogy a 2020 vevő és dekódoló berendezésnek 68 EEPROM-ja, 69 flash memóriája, 70 ROM-ja és 72 RAM-ja van. A 68 EEPROM olyan 74 védett területtel rendelkezik, amelyet egy virtuális gép használ, és ahová kizárólag a virtuális gép tud adatot beírni, közönséges alkalmazás nem. A 74 védett terület 16 vagy 256 bit nagyságú 76 kulcs érvényesítő bittérképet, valamint 32 bit nagyságú 80 eltolás bittérképet tartalmaz. Egy előnyös kiviteli alaknál a 70 ROM tizenhat 82 nyilvános kulcsot tartalmaz,

mely esetben 16 bites 76 kulcs érvényesítő bittérképet használunk, más kiviteli alak esetében viszont 256 nyilvános kulcsot tartalmaz, mely esetben értelemszerűen egy 256 bites 76 kulcs érvényesítő bittérképet használunk. A nyilvános kulcsokat azoknak a 70 ROM-ban való fizikai elhelyezkedése azonosítja, de alternatív megoldásként egy kereső táblázatba is foglalhatjuk őket, amely esetben egy meghatározott kulcs azonosító vezet el a megfelelő nyilvános kulcshoz. A 72 RAM-ot 84 ideiglenes kulcs tárolására használhatjuk.

Ha egy alkalmazást töltünk le, először az ahhoz az alkalmazáshoz tartozó előre meghatározott táblázat azonosítójú, valamint egy 0 értékű táblázat azonosító bővítményű könyvtár táblázatot töltünk le. A 44 kulcs azonosítót ezután ebből a könyvtár táblázatból vonjuk ki, és a 74 védett területen a 76 kulcs érvényesítő bittérképet leellenőrizzük, hogy a kivont és megkapott 44 kulcs azonosítónak megfelelő bit aktív állapotú-e. Ha nem, akkor az alkalmazás további letöltését megszakítjuk. Ha a megfelelő bit be van állítva, azaz aktív, akkor a 70 ROM-ból kiválasztunk a kivont 44 kulcs azonosítónak megfelelő 82 nyilvános kulcsot. Ezt a kiválasztott 82 nyilvános kulcsot, valamint egy ismert dekódolási műveletet használunk ezt követően arra, hogy a dekódolt blokk előállításához a 40 könyvtár táblázatban a 46 kódolt blokkot dekódoljuk. A 74 védett területen lévő 80 eltolás bittérképben lévő eltolás értéket megnézzük, vagy, ha egynél több eltolási bit van aktívra állítva, úgy minden egyes eltolás bitet megnézzünk, és a megkeresett eltolás értéktől kezdődően a dekódolt blokkból 16 bájtnyi adatot kivonunk. Az egy vagy több megkeresett eltolás érték vonatkozásában ezek a 16 bájtnyi adatok tekinthetők úgy, mint a 40 könyvtár táblázattal együtt elküldött aláírások. A 40 könyvtár táblázatban lévő 42 könyvtárban található bejegyzések aláírását az ismert MD5 eljárás segítségével számoljuk ki, és ezt a kiszámított aláírást összehasonlítjuk a dekódolt blokkból kivont aláírással. Ha az egy vagy több megkeresett eltoláshoz tartozó két aláírás nem egyezik meg egymással, akkor az alkalmazás letöltését szintén megszakítjuk. Ha a két aláírás azonban megegyezik egymással, akkor folytatjuk a 42 könyvtárban megadott modulok letöltését. Mint azt már jeleztük, annak érdekében, hogy egy megadott modult letölthessünk, az ahhoz a modulhoz tartozó táblázat azonosító bővítményt a 42 könyvtárból előbb meg kell kapnunk, és letöltjük a 40 könyvtár táblázat 28 táblázat azonosítójával egyező 28 táblázat azonosítójú, a megkapott 34 táblázat azonosító bővítményű MPEG 24 modulokat vagy 18 szekciókat. Ha a 24 modult letöltöttük, a 2020 vevő és dekódoló berendezéssel kiszámítjuk a letöltött táblázat aláírását, ismét az ismert MD5 eljárást alkalmazva, majd az így kiszámított aláírást összehasonlítjuk a könyvtár

bejegyzésben lévő aláírással. Ha a két aláírás megegyezik, a 24 modult elfogadjuk, de ha nem egyezik meg, akkor a 24 modult elvetjük.

Az alkalmazás minden egyes modulját úgy tölthetjük le, mint az előzőekben azt bemutat-
tuk, majd magát az alkalmazást futtathatjuk a 2020 vevő és dekódoló berendezésben.

A következőkben a 9-13. ábrák segítségével az adatoknak a 2020 vevő és dekódoló beren-
dezésbe történő letöltését mutatjuk be részletesebben. A 2020 vevő és dekódoló berendezés
100 program betöltőt tartalmaz, amelyet elsődlegesen arra használunk, hogy a szoftver le-
töltéséhez szükséges betöltőt töltsünk le, például gyártó főrmvert vagy a 4008 futás alatti
algoritmust vagy alkalmazásokat, amelyek az MPEG adatáramban vannak jelen, és a 2020
vevő és dekódoló berendezés 69 flash memóriájában tárolandók el. A 100 program betöltő
szintén a 2020 vevő és dekódoló berendezés 69 flash memóriájában van, jellemző módon
kitörölhetetlen módon eltárolva. A 100 program betöltő a 2020 vevő és dekódoló berende-
zés hardverének és az abban tárolt szoftvernek a vezénylete alatt működik.

A 2020 vevő és dekódoló berendezésben tárolt szoftver beírását illetve felfrissítését végez-
hetjük:

- a 2020 vevő és dekódoló berendezés felhasználójának a kezdeményezésére;
- a 2020 vevő és dekódoló berendezésben tárolt alkalmazás kezdeményezésére; vagy
- ha a 2020 vevő és dekódoló berendezésben már korábban eltárolt szoftver (amit leírá-
sunkban ekvivalens módon "rezidens" szoftvernek is nevezünk) megsérül.

Annak a megállapítására, hogy a 2020 vevő és dekódoló berendezésben tárolt rezidens
szoftver megsérült-e, a 2020 vevő és dekódoló berendezés gyártója által írt és abban tárolt
szoftverrel a rezidens szoftver adatokon ellenőrző összeget számítunk, és ezt összehason-
lítjuk a rezidens szoftverbe eleve beírt ellenőrző összeggel. Ha ezek értéke nem egyezik
meg egymással, az azt jelenti, hogy a rezidens szoftver megsérült, és minden valószínűség
szerint használhatatlan.

A 2020 vevő és dekódoló berendezés 69 flash memóriája és 68 EEPROM-ja olyan para-
métereket tartalmaz, amelyek lehetővé teszik a 100 program betöltő számára, hogy a bit-
folyamból natív kód formájában egy adat betöltőt töltsön le. A paramétereket tárolhatjuk a
100 program betöltőben magában, vagyis a 69 flash memóriában vagy a 68 EEPROM-ban.
A 69 flash memóriában eltárolható paraméterekre néhány, korántsem teljeskörű példát ho-
zunk fel:

- az a frekvenciaérték, amelyre 2014 transzponder be van hangolva,
- a 2020 vevő és dekódoló berendezés által demodulálendő jel különböző karakterisztikái, jellemzői;
- a szoftver továbbítását befolyásoló csomag azonosító;
- a felhatalmazás és a jogosultság ellenőrzése során használandó nyilvános kulcs készlet (előnyösen három nyilvános kulcsból);
- a 40 könyvtár táblázatoknak az MPEG bitfolyamból való letöltése során mértékadó időtúllépés;
- a 100 program betöltő verziószáma; és
- egy N bájtból álló ellenőrző összeg, amelyet arra használunk, hogy ellenőrizzük vele a rezidens szoftver integritását, ennek az ellenőrző összegnek az értékét a 2020 vevő és dekódoló berendezés gyártója határozza és adja meg.

A 68 EEPROM-ban tárolt, és a 2020 vevő és dekódoló berendezésben tárolt alkalmazással felfrissíthető paraméterekre példa:

- a 2020 vevő és dekódoló berendezés által demodulálendő jel további karakterisztikái, és
- az írásról illetve felfrissítésről szóló riport kompilálását lehetővé tevő paraméterek.

Ezeket a paramétereket a 69 flash memória vagy a 68 EEPROM megfelelő paraméter mezőiben tároljuk. A 10. ábrán látható, hogy minden egyes 400 paraméter mező 402 hosszúságú, és 404 tartalék bájtot, 406 paraméter készletet, valamint longitudinal redundancy code (LCR) 408 ellenőrző összeget foglal magában. A 408 ellenőrző összeg 410 CRL-t tartalmaz, amely a 400 paraméter mező bájtjainak kizárólagos VAGY értéke, valamint 412 NCRL-t tartalmaz, amely a 410 CRL 1-s komplementere. Ha a 2020 vevő és dekódoló berendezésben tárolt alkalmazás a 400 paraméter mezőben tárolt paramétereket fel kívánja frissíteni például azért, hogy frissítse a csomag azonosító értéket, ehhez az adott mezőre vonatkozó LRC 408 ellenőrző összeget számol ki, és azt összehasonlítja a mezőben eltárolt 410 CRL-lel. Ha a két érték egyező, az azt jelenti, hogy a paraméter mező frissíthető, ha nem, akkor a 400 paraméter mező frissítése nem engedélyezett. A 2020 vevő és dekódoló berendezésbe letöltendő adatokat tartalmazó MPEG bitfolyam natív kódot tartalmaz, amely legalább részben egy olyan további, járulékos betöltőt foglal magában, amelyet a 100 program betöltőtől eltérően "folyam" betöltőnek nevezünk. A 100 program betöltő ezt a folyam betöltőt az MPEG bitfolyamból tölti le a 2020 vevő és dekódoló berendezés 72 RAM-jába, és ezt követően ez a folyam betöltő lesz az, amely gondoskodik az adatoknak az MPEG

bitfolyamból történő letöltésére, például azért, hogy a 2020 vevő és dekódoló berendezésben lévő rezidens szoftvert frissítse.

A 2020 vevő és dekódoló berendezés 69 flash memóriájába letöltött szoftver is tartalmazhat egy betöltőt, amelyet a megkülönböztethetőség céljából "rezidens" betöltőnek hívunk. Ennek a betöltőnek legalább arra alkalmasnak kell lennie, hogy az MPEG bitfolyamból kapott szoftver írását vagy frissítését elvégezze, de ezen kívül egyéb jellemzőkkel is rendelkezhet, például frissítéseket fogadhat a helyi portoktól, és engedélyezheti az MPEG bitfolyamban lévő video és audio adatok dekódolását. A rezidens betöltő egy alkalmazástól származó kérelemre kerül letöltésre a bitfolyamból, például azért, hogy kiegészítse azt a betöltőt, amely végrehajtja a folyam betöltő letöltését, vagy, hogy a bitfolyamból adatokat töltsön le. Például, ha a 2020 vevő és dekódoló berendezésben tárolt alkalmazás írás vagy frissítés kérelmet bocsát ki, és a rezidens szoftver nem sérült, akkor ezt a rezidens betöltőt használjuk arra, hogy ezt a felfrissítést végrehajtsuk, nem pedig a folyam betöltőt. Ezzel csökkenteni tudjuk azt az időt, amely a 2020 vevő és dekódoló berendezésben lévő szoftver frissítéséhez szükséges. A rezidens betöltő is legalább részben natív kód alakú.

A 11. és 12. ábra segítségével azokat a különböző MPEG táblázatokat mutatjuk be, amelyek az MPEG bitfolyamba ágyazva lehetővé teszik a 2020 vevő és dekódoló berendezés számára, hogy a keresett szoftvert megtalálja és letölthesse.

Az MPEG bitfolyam legalább egy 200 hardver könyvtár táblázatot és több 300 betöltő könyvtárat foglal magában.

Egy 200 hardver könyvtár táblázat lehetővé teszi a 100 program betöltő számára, hogy az megtalálja a folyam betöltő megfelelő verzióit, valamint a 2020 vevő és dekódoló berendezés különböző verzióihoz igazodó, letölthető szoftvert. A 11. ábrán látható, hogy a 200 hardver könyvtár táblázat D0 értékű 202 táblázat azonosítót, valamint 0000 értékű 204 táblázat azonosító bővítményt tartalmaz, mely értékeket azt megelőzően már eltároltuk a 2020 vevő és dekódoló berendezés 68 EEPROM-jába, például azért, hogy lehetővé tegyünk a 100 program betöltő számára a 200 hardver könyvtár táblázat megtalálását és letöltését.

A 200 hardver könyvtár táblázat az alábbiakat tartalmazza:

- a 200 hardver könyvtár táblázat 206 verziószámát, amelyet minden esetben megnövelünk, ha a 200 hardver könyvtár táblázat tartalmában valamilyen változás lép fel;
- a 200 hardver könyvtár táblázatban lévő folyam betöltő leírások 208 számát;

- minden egyes 2020 vevő és dekódoló berendezés verzióhoz:

- a 2020 vevő és dekódoló berendezés verziószámának 210 azonosítóját;
 - a 210 azonosítóval társított 300 betöltő könyvtárhoz használt MPEG táblázatok 212 táblázat azonosítóját;
 - 214 redundáns bájtot;
 - a 210 azonosítóval társított 300 betöltő könyvtárhoz használt MPEG táblázat egy szekciójának 216 legnagyobb szekcióméretét;
 - a 210 azonosítóval társított 300 betöltő könyvtár betöltésénél mértékadó 218 időtúllépés értéket; és
 - a 210 azonosítóval társított 300 betöltő könyvtár 220 aláírás értékét;
- a 200 hardver könyvtár táblázat jogosultságának megállapításához használt 222 privát kulcs azonosítót; és
- 224 kódolt területet, amely 200 hardver könyvtár táblázat olyan 226 aláírását tartalmazza, amely 228 aláírás eltolás értékkel el van tolva a 224 kódolt terület kezdetétől.

A 2020 vevő és dekódoló berendezés 210 azonosítója 4 bájttal hosszúságú. Ebből 1 bájttal jövőbeli használatra tartalékoltunk, 2 bájttal tartalmazza a 2020 vevő és dekódoló berendezés hardverének verziószámához tartozó kódot, és 1 bájttal a 2020 vevő és dekódoló berendezés gyártójának kódját tartalmazza. Ezáltal a 100 program betöltő azt a folyamat betöltő verziót tudja letölteni, amely kompatibilis a 2020 vevő és dekódoló berendezés hardver platformjával.

A 200 hardver könyvtár táblázat letöltését követően a 100 program betöltővel azt végigkutatjuk, és a 2020 vevő és dekódoló berendezés 210 azonosítójának megfelelő bejegyzést keresünk. Ha nem találunk ilyen bejegyzést, a letöltést megszakítjuk. Ha megtaláljuk a keresett bejegyzést, a 100 program betöltővel a 200 hardver könyvtár táblázatból beazonosítjuk azt a 212 táblázat azonosítót, amely a 2020 vevő és dekódoló berendezés 210 azonosítójával, a folyamat betöltővel és a letöltendő szoftverrel társított 300 betöltő könyvtárhoz lett hozzárendelve.

A 12. ábrán látható, hogy a 2020 vevő és dekódoló berendezés 210 azonosítójával társított minden egyes 300 betöltő könyvtár az alábbiakból áll:

- a 300 betöltő könyvtár 302 verziószámából, amelyet ugyancsak minden alkalommal megnövelünk, ha változás áll be a 300 betöltő könyvtár tartalmában;

- a folyam betöltő MPEG táblázatainak a 304 számát;
- a folyam betöltő 306 verziószámát;
- a letöltendő szoftver MPEG táblázatainak 308 számát;
- a letöltendő szoftver 310 verziószámát;
- a folyam betöltő minden egyes MPEG táblázatára érvényesen:
 - az MPEG táblázat 312 azonosítóját;
 - az MPEG táblázat 314 táblázat azonosító bővítményét,
 - két 316 redundáns bájtot,
 - az MPEG táblázat MPEG szekciójának legnagyobb 318 szekció méretét;
 - az MPEG táblázat betöltésénél mértékadó 320 időtúllépés értéket; és
 - az MPEG táblázat 322 aláírását;
- a szoftver minden egyes MPEG táblázata vonatkozásában:
 - az MPEG táblázat 324 azonosítóját;
 - az MPEG táblázat 326 táblázat azonosító bővítményét;
 - két 328 redundáns bájtot;
 - a szóban forgó MPEG táblázat MPEG szekciójának legnagyobb 330 szekcióméretét;
 - az MPEG táblázat betöltésénél mértékadó 332 időtúllépés értéket; és
 - az MPEG táblázat 334 aláírását;
- a 300 betöltő könyvtár jogosultságának a megállapításánál használt privát kulcs 336 azonosítóját; és
- 338 kódolt területet, amely olyan 300 betöltő könyvtár 340 aláírást tartalmaz, amely 342 aláírás eltolásnyival el van tolvaa a 338 kódolt terület kezdetétől.

A frissítés alatt egy riportot képzünk és kompilálunk, amely többek között tartalmazza az írási vagy felfrissítési eljárás minden egyes lépésének a részleteit, továbbá azt, hogy a lépés vagy művelet sikeresen befejeződött-e, vagy sem, mely utóbbi esetben be tudjuk azonosítani azt a lépést, amelynek során az írási vagy frissítési eljárás megszakadt vagy hibát jelzett. A riport például tartalmazhatja az alábbiakat:

- a 200 hardver könyvtár táblázat 206 verziószámát;
- ha a 200 hardver könyvtár táblázat feldolgozása során hiba lépett fel, akkor ennek a hibának a típusjelzését, valamint a 200 hardver könyvtár táblázat MPEG táblázatának azt a táblázat azonosító bővítményét, amelyiknél a hiba fellépett,
- a 300 betöltő könyvtár 302 verziószámát;
- ha a 300 betöltő könyvtár feldolgozása során hiba lépett fel, akkor a hiba típusának a jelzését, valamint a 300 betöltő könyvtár MPEG táblázatának azt a táblázat azonosító bővítményét, amelyiknél a hiba fellépett; és
- ha hiba lépett fel a folyam betöltő feldolgozása közben, akkor a hiba típusának a jelzését, valamint a folyam betöltő MPEG táblázatának azt a táblázat azonosító bővítményét, amelyiknél a hiba fellépett; és
- ha a szoftver feldolgozása során lépett fel hiba, akkor a hiba típusának a jelzését, valamint a szoftver MPEG táblázatának azon táblázat azonosító bővítményét, amelyiknél a hiba jelentkezett.

A riport tartalmazza azt az okot is, amiért az írást vagy frissítést el kellett végeznünk, például, hogy az írás vagy frissítés egy alkalmazás kezdeményezésére történt, vagy esetleg a szoftverben hiba vagy bizonytalanság lépett fel, akkor a szoftverben lévő hibák számát, vagy éppen a frissítési kísérletek, pontosabban a meghiúsult felfrissítések számát.

Ha a 2020 vevő és dekódoló berendezésben lévő rezidens szoftvert egy alkalmazás kérelmére frissítjük fel az MPEG bitfolyamban lévő szoftverrel, a 2020 vevő és dekódoló berendezést úgy állítjuk be, hogy összehasonlítsa a frissen letöltött 300 betöltő könyvtár táblázatban azonosított szoftver 310 verziót a benne tárolt, rezidens szoftver 310 verziószámával. Ha a 310 verziószám újabb, akkor a rezidens szoftverhez tartozó modulokat töröljük a 2020 vevő és dekódoló berendezés 69 flash memóriájából, és a frissített szoftver modulokat letöltjük és telepítjük.

A 2020 vevő és dekódoló berendezés 4038 LED kijelzőjét, amely a 2020 vevő és dekódoló berendezés előlapján nyer legtöbbször elhelyezést, úgy alakíthatjuk ki, hogy a felhasználó számára továbbítsa a 2020 vevő és dekódoló berendezéstől eredő üzeneteket az adatok letöltése közben. Például a 2020 vevő és dekódoló berendezés 69 flash memóriájában tárolt 400 paraméter mezőben az alábbi négy üzenetet definiálhatjuk:

- "LETÖLTÉS" üzenet, amely azt jelzi, hogy az írás vagy frissítés normálisan zajlik, vagyis egy alkalmazás kezdeményezésére történik;
- "NATÍV" üzenet, amely jelzi, hogy a frissítés úgynevezett natív állapotban zajlik, valószínűleg azért, mert a 2020 vevő és dekódoló berendezés rezidens szoftverre meghibásodott,
- egy "000" üzenet, amely azt jelzi, hogy a 100 program betöltő nem tudja végrehajtani az írás vagy frissítés műveletet, mivel nem képes érvényes vagy megfelelő paramétereket felkutatni (mint amilyen az a frekvencia, amelyre a 4028 MPEG tunert be kell állítani, vagy az MPEG bitfolyam csomag azonosítója), a 2020 vevő és dekódoló berendezés memóriájában; és
- egy "HIBA" üzenet, amely azt jelzi, hogy az írás vagy frissítés alatt a "000" üzenettel kapcsolatosan meghatározottól eltérő hiba lépett fel.

A 2020 vevő és dekódoló berendezés segítségével az előbbi, statikus üzenetek helyett animált rajzos üzenetek is kiadhatók.

A 13. ábra segítségével példaképpen a 2020 vevő és dekódoló berendezésben lévő rezidens szoftver frissítésének egy lehetséges megvalósítását mutatjuk be főbb folyamatábra szintjén.

S101 lépésben a 2020 vevő és dekódoló berendezésben lévő szoftverrel megvizsgáljuk valamely rezidens szoftver integritását, úgy, hogy ellenőrző összeg számítást végzünk, és annak eredményét összehasonlítjuk a rezidens szoftverben tárolt összehasonlító összeg értékével. Ha a két érték különböző, akkor a felfrissítést natív állapotban, natív formában folytatjuk, ha a két érték megegyezik, vagy nem találunk rezidens szoftvert, akkor a frissítést a szokásos módon folytatjuk.

Ebben a natív állapotban ezt követően megvizsgáljuk, S102 lépésben, hogy függőben van-e még esetleg egy előző frissítési kérelem. Ha valamilyen alkalmazástól egy ilyen frissítési kérelem van még függőben, azt a kérelmet S103 lépésben töröljük, majd az S102 lépést megismételjük. Ha nem találunk ilyen függő felfrissítési kérelmet, S104 lépésben töröljük az előző frissítésről szóló riportot, és a most következő frissítésre vonatkozó események jegyzőkönyvezéséhez szükséges kezdeményezést elvégezzük. Az ennek eredményeként létrejövő riport tartalmazni fogja a frissítési kérelem okát, ami jelen esetben egy sérült szoftver cseréjének az igénye.

S105 lépésben a 2020 vevő és dekódoló berendezés 4038 LED kijelzőjén megjelenítjük az előbb vázolt "NATÍV" üzenetet.

S106 lépésben megvizsgáljuk a 68 EEPROM és a 69 flash memória paraméter mezőiben tárolt paramétereket. Ha úgy találjuk, hogy a hangolási paraméterek és/vagy a csomag azonosító paraméter nincs meghatározva, a 2020 vevő és dekódoló berendezés 4038 LED kijelzőjén az "000" üzenetet jelenítjük meg, és a frissítést megszakítjuk.

Ha ezek a paraméterek definiálva vannak a 400 paraméter mezőben, a frissítési eljárást S107 lépésben folytatjuk, amelyben a 100 program betöltővel a 4028 MPEG áram tunert a 2014 transzponderre hangoljuk, a 400 paraméter mezőben tárolt paraméterekkel összhangban. Ha a hangolás sikertelen a frissítést megszakítjuk, és a 2020 vevő és dekódoló berendezés 4038 LED kijelzőjén a "HIBA" üzenetet jelenítjük meg.

Ha a hangolás sikeres, akkor a 100 program betöltővel S108 lépésben letöltjük a 200 hardver könyvtár táblázatot, és azt ellenőrizzük. Ha a 200 hardver könyvtár táblázat letöltése nem fejeződik be az időtúllépés értéken belül, vagy ha úgy találjuk, hogy a 200 hardver könyvtár táblázat nem hiteles, (például, mert a letöltés közben hiba lépett fel), a frissítést megszakítjuk, és a 2020 vevő és dekódoló berendezés 4038 LED kijelzőjén szintén a "HIBA" üzenetet jelenítjük meg.

Ha mind a letöltés, mind a hitelesítés eredményes, a 100 program betöltővel a 2020 vevő és dekódoló berendezés verziószámának megfelelő 210 verziószámot keresünk, ez egyébként ugyancsak a 400 paraméter mezőben van megadva. Ha nem találunk 210 verziószám értéket, a frissítést szintén megszakítjuk és a 2020 vevő és dekódoló berendezés 4038 LED kijelzőjén a "HIBA" üzenetet jelezzük ki.

Ha viszont találunk ilyen bejegyzést, akkor a 100 program betöltővel kiolvassuk azoknak az MPEG táblázatoknak a 212 táblázat azonosítóit, amelyeket a 210 verziószámmal, a folyam betöltővel és letöltendő szoftverrel társított 300 betöltő könyvtár táblázathoz használtunk, és S109 lépésben letöltjük és hitelesítjük a megfelelő, korrekt 300 betöltő könyvtár táblázatot. Ha a 300 betöltő könyvtár táblázat letöltése nem sikerül a megadott időtúllépés értéken belül, vagy a 300 betöltő könyvtár táblázat nem hiteles (például, mert letöltés közben hiba lépett fel), a frissítést megszakítjuk és a 2020 vevő és dekódoló berendezés 4038 LED kijelzőjén a "HIBA" üzenetet jelezzük ki.

Ha a letöltés és a hitelesítés eredményes, akkor S110 lépésben a 100 program betöltővel az MPEG bitfolyamból letöltjük a folyam betöltőt a 2020 vevő és dekódoló berendezés 72 RAM-jába. Ha a folyam betöltő letöltése nem fejeződik be a megadott időtúllépés értéken belül, vagy úgy találjuk, hogy a folyam betöltő nem hiteles (például, mert hiba lépett fel a letöltés során) a frissítést megszakítjuk, és a 2020 vevő és dekódoló berendezés 4038 LED kijelzőjén a "HIBA" üzenetet jelezzük ki.

Ha a folyam betöltőt eredményesen letöltöttük és hitelesítettük, akkor S111 lépésben végrehajtjuk a folyam betöltőt, majd S112 lépésben a 2020 vevő és dekódoló berendezésben lévő rezidens szoftver meghibásodott részét kitöröljük, és a letöltendő szoftver szegmenseket a folyam betöltővel letöltjük, hitelesítjük, majd a 69 flash memória megfelelő címerire beírjuk. Ha a szoftver letöltését nem tudjuk befejezni a megadott időtúllépés értéken belül, vagy úgy találjuk, hogy a szoftver nem hitelesíthető (például, mert a letöltés során valamilyen hiba lépett fel, vagy hiba lép fel a szoftver szegmenseknek a 69 flash memóriájába írása során), a frissítést megszakítjuk, és a 2020 vevő és dekódoló berendezés 4039 LED kijelzőjén a "HIBA" üzenetet jelezzük ki.

Ha a rezidens szoftver frissítését eredményesen befejeztük S113 lépésben leállítjuk a riport írást, és a 2020 vevő és dekódoló berendezést alaphelyzetbe állítjuk, hogy adott esetben egy további frissítést tudjon végrehajtani.

Minden olyan lépés esetében, amikor jeleztük, hogy a frissítést megszakítjuk, azt a lépést, amennyiben szükséges és lehetséges, előre meghatározott számban megismételhetjük, amíg a lépést eredményesen be tudjuk fejezni, vagy amíg az adott lépés végrehajtására megadott időtúllépés értéket el nem érjük.

Ha a frissítést normál állapotban kell elvégeznünk, akkor S201 lépésben a 100 program betöltővel megállapítjuk, hogy van-e már függőben valamilyen, egy alkalmazástól eredő frissítési kérelem. Ha nincs, akkor a frissítést a szokásos módon végezzük, de ha van ilyen függő kérelem, akkor először ezt a függő kérelmet teljesítjük.

S202 lépésben töröljük az előző frissítésre vonatkozó riportot, és a most következő frissítés nyomon követése érdekében egy újabb riportot inicializálunk. Ez a riport tartalmazni fogja a frissítési kérelem közvetlen kiváltó okát, például, hogy a frissítést egy alkalmazás kezdeményezte, valamint az összes olyan frissítési opciót és paramétert, amelyet az alkalmazás kiválasztott vagy meghatározott.

Ezt követően S203 lépésben megvizsgáljuk, hogy a 2020 vevő és dekódoló berendezés 69 flash memóriájában található-e valamilyen rezidens betöltő eltárolva. Ha a 2020 vevő és dekódoló berendezés 69 flash memóriájában találunk ilyen betöltőt, akkor S204 lépésben megvizsgáljuk, hogy ezt a rezidens betöltőt a 2020 vevő és dekódoló berendezésben tárolt szoftvertől származó parancsra válaszként végrehajtottuk-e. Ha a rezidens betöltő végrehajtása megtörtént, akkor a rezidens betöltővel ezt követően azokat a lépéseket hajtjuk végre a frissítési eljárás során, amelyet normális esetben a 100 program betöltővel hajtottunk volna végre.

Ha a rezidens betöltőt nem találjuk, vagy nem lett végrehajtva, akkor viszont a 100 program betöltőt használjuk. A 2020 vevő és dekódoló berendezésben tárolt szoftver számára az is lehetséges, hogy a 100 program betöltőt kényszerítse a frissítési eljárás folytatására abban az esetben is, ha a 2020 vevő és dekódoló berendezés 69 flash memóriájában van rezidens betöltő eltárolva.

A 2020 vevő és dekódoló berendezés 4038 LED kijelzőjén ebben az esetben a "LETÖLTÉS" üzenetet jelenítjük meg.

S206 lépésben megvizsgáljuk a 69 flash memória valamint a 68 EEPROM 400 paraméter mezőiben tárolt paramétereket. Ha úgy találjuk, hogy vagy a hangolási paraméterek, és/vagy a csomag azonosító paraméterek nincsenek megadva, a 2020 vevő és dekódoló berendezés 4038 LED kijelzőjén a "000" üzenetet jelenítjük meg, és a frissítést megszakítjuk.

Ha ezek a paraméterek a paraméter mezőben definiálva vannak, a frissítési eljárásban az S207 lépést hajtjuk végre, amelynek során vagy a 100 program betöltővel vagy a rezidens betöltővel a 2020 vevő és dekódoló berendezés 4028 MPEG áram tunerjét ráhangoljuk a 2014 transzponderre a 400 paraméter mezőben tárolt paramétereknek megfelelően. Ha a hangolás sikertelen, a frissítést megszakítjuk, és a 2020 vevő és dekódoló berendezés 4038 LED kijelzőjén a "HIBA" üzenetet jelezzük ki. Ha a hangolás eredményes, akkor a 100 program betöltővel, vagy a rezidens betöltővel S208 lépésben letöltjük és hitelesítjük a 200 hardver könyvtár táblázatot. Ha a 200 hardver könyvtár táblázat letöltése nem sikerül az előre meghatározott időtúllépés értéken belül, vagy a 200 hardver könyvtár táblázat nem hitelesíthető (például, mert a letöltés során valamilyen hiba lépett fel), vagy a frissítést kezdeményező alkalmazás által kiválasztott opció függvényében az ugyanolyan 206 verziószámú 200 hardver könyvtár táblázat alkalmazásával hajtottunk végre egy eredményes

frissítést, akkor a frissítést megszakítjuk, és a 2020 vevő és dekódoló berendezés 4038 LED kijelzőjén a "HIBA" üzenetet jelezzük ki.

Ha a letöltés és hitelesítés eredményes, és a frissítést az alkalmazás engedélyezi, a 100 program betöltővel, vagy a rezidens betöltővel a 2020 vevő és dekódoló berendezésnek a 400 paraméter mezőben megadott verziószámának megfelelő 210 verziószámot keresünk. Ha nem találunk ilyen 210 verziószámot, a frissítést megszakítjuk és a 2020 vevő és dekódoló berendezés 4038 LED kijelzőjén a "HIBA" üzenetet jelezzük ki.

Ha találunk ilyen keresett bejegyzést, akkor a 100 program betöltővel vagy a rezidens betöltővel a 210 verziószámmal, a folyam betöltővel és a letöltendő szoftverrel társított 300 betöltő könyvtár táblázathoz használt MPEG táblázatok 212 táblázat azonosítóit olvassuk, majd S209 lépésben letöltjük és hitelesítjük a helyes 300 betöltő könyvtár táblázatot.

Ha a 300 betöltő könyvtár táblázat letöltése nem fejeződik be a megadott időtúllépés értéken belül, vagy ha a 300 betöltő könyvtár táblázat nem hitelesíthető (például, mert a letöltés során valamilyen hiba lépett fel), vagy a frissítést kezdeményező alkalmazás által kiválasztott opciótól függően, ha az azonos 306 verziószámú 300 betöltő könyvtár táblázat használatával eredményes frissítést hajtottunk végre, akkor a frissítést megszakítjuk, és a 2020 vevő és dekódoló berendezés 4038 LED kijelzőjén a "HIBA" üzenetet jelenítjük meg.

Ha mind a letöltés mind a hitelesítés eredményes, és az alkalmazás engedélyezi a frissítést, akkor a 100 program betöltővel az MPEG bitfolyamból letöltjük a folyam betöltőt a 2020 vevő és dekódoló berendezés 70 RAM-jába az S210 lépésben. Ha a folyam betöltő letöltését nem fejezzük be a megadott időtúllépés értéken belül, vagy a folyam betöltő nem hitelesíthető (például, mert a letöltés során valamilyen hiba lépett fel), a frissítést megszakítjuk, és a 2020 vevő és dekódoló berendezés 4038 LED kijelzőjén a "HIBA" üzenetet jelenítjük meg.

Ha a folyam betöltőt eredményesen betöltöttük és hitelesítettük, akkor S211 lépésben végrehajtjuk és S212 lépésben az MPEG bitfolyamban lévő szoftver 310 verziószámát összehasonlítjuk a rezidens szoftver verziószámával.

Ha a verziószámok azonosak, akkor a szoftvert nem írjuk bele a 69 flash memóriába, és az alkalmazás frissítés kérelmet töröljük. Ha viszont a verziószámok eltérnek egymástól, akkor a rezidens szoftvert töröljük, és a letöltendő szoftver szegmenseket a folyam betöltő segítségével letöltjük, hitelesítjük, majd S213 lépésben beírjuk a 69 flash memóriába.

Ha a szoftver letöltését nem tudjuk befejezni a megadott időtúllépés értéken belül, vagy ha a szoftver nem hitelesíthető, (például, mert a letöltés közben valamilyen hiba lépett fel), vagy ha valamilyen hiba lép fel a szoftvernek a 69 flash memóriába történő beírása közben, a frissítést megszakítjuk, és a 2020 vevő és dekódoló berendezés 4038 LED kijelzőjén a "HIBA" üzenetet jelezzük ki. Ha a rezidens szoftvert eredményesen felfrissítettük, a riport írást S214 lépésben leállítjuk, a függő frissítési kérelmet töröljük, és a 2020 vevő és dekódoló berendezést alaphelyzetbe állítjuk, hogy az készen álljon egy következő frissítési műveletre.

Hasonlóképpen a napi állapotban végrehajtott frissítéshez, minden olyan lépést, amelynek során a frissítést valamilyen oknál fogva meg kellett szakítanunk, újból elvégezhetjük mindaddig, amíg a lépést eredményesen be nem fejezzük.

Kihangsúlyozzuk, hogy a találmányt csupán a megértéséhez szükséges mértékben, példák segítségével mutattuk be, és a találmánynak a szabadalmi igénypontokkal körülbástyázott oltalmi körén belül számos részletbeli módosítás tehető.

A leírásból, igénypontokból és rajzokból kivehető minden egyes jellemző akár külön, akár tetszőleges kombinációban létrehozható. A találmány bemutatásánál leírt előnyös kiviteli alakoknál a találmány néhány, lényeges és nem annyira lényeges jellemzőjét számítógép program segítségével valósítottuk meg, azonban szakember számára nyilvánvaló és egyértelmű, hogy ezek a jellemzők kivétel nélkül hardveresen is megvalósíthatók. Ezen túlmenően nyilvánvaló, hogy a hardveresen, szoftveresen és hasonló módon megvalósított funkciókat villamos és elektronikus jelek felhasználásával hajtjuk végre.



Szabadalmi igénypontok

1. Eljárás adatok letöltésére egy adó és vevő berendezésbe, **azzal jellemezve**, hogy a vevő és dekódoló berendezésben (2020) az adatokat magában foglaló bitfolyamot veszünk, a bitfolyamból az adatok betöltéséhez a vevő és dekódoló berendezésbe letöltünk egy betöltőt, majd az adatokat a bitfolyamból a letöltött adat betöltő felhasználásával letöltjük.
2. Az 1. igénypont szerinti eljárás, **azzal jellemezve**, hogy a letöltött adat betöltőt a vevő és dekódoló berendezésből töröljük, miután az adatokat letöltöttük a bitfolyamból.
3. Az 1. igénypont szerinti eljárás, **azzal jellemezve**, hogy a letöltött adat betöltőt azt követően a vevő és dekódoló berendezés (2020) nemfelejtő tájában eltároljuk.
4. A 3. igénypont szerinti eljárás, **azzal jellemezve**, hogy nemfelejtő tárként a vevő és dekódoló berendezés (2020) memória kötetét használjuk.
5. Az 1-4. igénypontok bármelyike szerinti eljárás, **azzal jellemezve**, hogy az adatok letöltését a letöltött adat betöltővel hajtjuk végre.
6. Az 1-5. igénypontok bármelyike szerinti eljárás, **azzal jellemezve**, hogy a vevő és dekódoló berendezésben (2020) tárolt adatoknak csupán egy részét helyettesítjük a letöltött adat betöltővel letöltött megfelelő adatrésszel.
7. Az 1-6. igénypontok bármelyike szerinti eljárás, **azzal jellemezve**, hogy a legalább egy adat betöltőt magában foglaló bitfolyam esetében az adatokat adó rendszerben a legalább egy adat betöltőt modulokra osztjuk, továbbá a legalább egy adat betöltőhöz az adatokat is megfelelő számú modulokra osztjuk, ahol minden meghatározott számú adatmodult meghatározott számú adat letöltő modullal társítunk.
8. A 7. igénypont szerinti eljárás, **azzal jellemezve**, hogy az adó rendszerben a legalább egy adat betöltőhöz mindegyik modult olyan táblázatként alakítjuk ki, amelyeknek ugyanazt a táblázat azonosítót (TID), és különböző táblázat azonosító bővítményeket (TID bővítmények) adunk, és a legalább egy adatmodulhoz minden egyes adatmodult egy-egy olyan táblázatként alakítunk ki, amelyek táblázat azonosítója megegyezik a velük társított adat betöltő modulok táblázatainak a táblázat azonosítójával, de különböző táblázat azonosító bővítményeket tartalmaznak.
9. A 8. igénypont szerinti eljárás, **azzal jellemezve**, hogy a letöltés során az azonos táblázat azonosítójú modul táblázatokat töltjük le.

10. A 9. igénypont szerinti eljárás, **azzal jellemezve**, hogy a táblázatok egy előre meghatározott táblázat azonosító bővítménytől eltérő táblázat azonosító bővítménnyel rendelkeznek, továbbá az adó rendszerben az ugyanolyan táblázat azonosítójú modulokhoz egy-egy könyvtár táblázatot állítunk elő, amelynek vagy amelyeknek az előre meghatározott táblázat azonosító bővítménnyel azonos táblázat azonosító bővítménye van, valamint azonos táblázat azonosítója van, és a könyvtár táblázat minden egyes modul részére tartalmazza a modul nevét, valamint az ahhoz tartozó táblázat azonosító bővítményt.

11. A 10. igénypont szerinti eljárás, **azzal jellemezve**, hogy a vevő és dekódoló berendezésben (2020) az előre meghatározott táblázat azonosító bővítménnyel rendelkező egyik táblázat letöltésével egy könyvtár táblázatot letöltünk, a könyvtár táblázat tartalmából meghatározzuk a könyvtár táblázat azonosítójával azonos táblázat azonosítójú modul táblázatok táblázat azonosító bővítményét, és a letöltés során azokat a modul táblázatokat töltjük le, amelyek táblázat azonosítója megegyezik a letöltött könyvtár táblázat táblázat azonosítójával, és amelyek táblázat azonosító bővítményeit a letöltött könyvtár táblázatból határoztuk meg.

12. Az 1-11. igénypontok bármelyike szerinti eljárás, **azzal jellemezve**, hogy az adó rendszerben előre meghatározott táblázat azonosítóval rendelkező, valamint egy vevő és dekódoló berendezés (2020) minden egyes verzió azonosítójához egy-egy hozzárendelt táblázat azonosítót tartalmazó könyvtár táblázatot állítunk elő.

13. A 12. igénypont szerinti eljárás, **azzal jellemezve**, hogy a verzió azonosító a vevő és dekódoló berendezés (2020) verziójára jellemző kódot, valamint a vevő és dekódoló berendezés (2020) gyártójára jellemző kódot tartalmaz.

14. A 12. vagy 13. igénypont szerinti eljárás, **azzal jellemezve**, hogy a vevő és dekódoló berendezésben (2020) az előre meghatározott táblázat azonosítójú könyvtár táblázatot letöltjük, a vevő és dekódoló berendezés (2020) verzió azonosítóját meghatározzuk, valamint egy könyvtár táblázat letöltése során azt a könyvtár táblázatot töltjük le, amely a vevő és dekódoló berendezés (2020) verzió számával társított táblázat azonosítóval, és az előre meghatározott táblázat azonosító bővítménnyel rendelkezik.

15. A 10-14. igénypontok bármelyike szerinti eljárás, **azzal jellemezve**, hogy az adó rendszerben minden egyes elküldött könyvtár táblázatba egy arra jellemző könyvtár verzió azonosítót illesztünk, és a vevő és dekódoló berendezésben (2020) meghatározzuk, hogy az

éppen elküldött könyvtár táblázat könyvtár verzió azonosítója fiatalabb-e, mint egy korábban letöltött, az éppen elküldött könyvtár táblázat táblázat azonosítójával azonos táblázat azonosítójú könyvtár táblázat, és amennyiben nem, úgy az adatok letöltését megszakítjuk.

16. Az 1-15. igénypontok bármelyike szerinti eljárás, **azzal jellemezve**, hogy az adó rendszerben a bitfolyamba az adatokra vonatkozó adat verzió azonosítót illesztünk, és a vevő és dekódoló berendezésben (2020) meghatározzuk, hogy a vett adatok adatverzió azonosítója fiatalabb-e, mint az eltárolt adatok adatverzió azonosítója, és amennyiben igen, úgy az adatokat a bitfolyamból letöltjük.

17. Az 1-16. igénypontok bármelyike szerinti eljárás, **azzal jellemezve**, hogy a letöltött adat betöltő legalább egy része a vevő és dekódoló berendezés (2020) hardver felépítésére jellemző kód alakjában van megvalósítva.

18. Az 1-17. igénypontok bármelyike szerinti eljárás, **azzal jellemezve**, hogy az adó oldalon egy második adat betöltőt küldünk el a bitfolyamba illesztve, és a vevő és dekódoló berendezésben (2020) a második adat betöltőt letöltjük, és az elsőként említett adat betöltőt és az adatokat letöltjük, amelynek során a második adat betöltővel hajtjuk végre az elsőként említett egyik adat betöltő, valamint az adatok letöltését.

19. A 18. igénypont szerinti eljárás, **azzal jellemezve**, hogy a második adat betöltő legalább egy részét a vevő és dekódoló berendezés (2020) hardver felépítésére jellemző kód alakjában képezzük ki.

20. Egy vevő és dekódoló berendezés, **azzal jellemezve**, hogy adatokat magában foglaló bitfolyam vételére alkalmas vevőt, tárolóeszközt, valamint a bitfolyamból az adatoknak a vevő és dekódoló berendezésbe (2020) történő letöltéséhez szükséges adat betöltőt a tárolóeszközbe letöltő letöltő eszközt tartalmaz.

21. A 20. igénypont szerinti vevő és dekódoló berendezés, **azzal jellemezve**, hogy a letöltött adat betöltőt az adatoknak a letöltését követően a bitfolyamból a vevő és dekódoló berendezésből törlő eszközt tartalmaz.

22. A 20. igénypont szerinti vevő és dekódoló berendezés, **azzal jellemezve**, hogy a letöltött adat betöltőt a bitfolyamból való letöltését követően eltároló nemfelejtő tárat tartalmaz.

23. A 22. igénypont szerinti vevő és dekódoló berendezés, **azzal jellemezve**, hogy a nemfelejtő tár a vevő és dekódoló berendezés (2020) flash memóriája.

24. A 20-23. igénypontok bármelyike szerinti vevő és dekódoló berendezés, **azzal jellemezve**, hogy a letöltött adat betöltő adatoknak a bitfolyamból történő letöltését végrehajtó betöltő.

25. A 20-24. igénypontok bármelyike szerinti vevő és dekódoló berendezés, **azzal jellemezve**, hogy a letöltött adat betöltő a vevő és dekódoló berendezésben (2020) tárolt adatoknak a letöltött megfelelő adatrésszel csupán egy részét helyettesítő adat betöltő.

26. A 20-25. igénypontok bármelyike szerinti vevő és dekódoló berendezés, **azzal jellemezve**, hogy táblázatok letöltését végző vevő és dekódoló berendezésként (2020) van kialakítva.

27. A 26. igénypont szerinti vevő és dekódoló berendezés, **azzal jellemezve**, hogy a letöltő eszköz úgy van kialakítva, hogy egy táblázat azonosítót és egy előre meghatározott táblázat azonosító bővítményt tartalmazó táblázat letöltésével egy könyvtár táblázatot töltsön le, továbbá, hogy meghatározza a könyvtár táblázat tartalmából a könyvtár táblázat táblázat azonosítójával azonos táblázat azonosítójú modul táblázatok táblázat azonosító bővítményeit, és hogy letöltse azokat a modul táblázatokat, amelyek táblázat azonosítója megegyezik a letöltött könyvtár táblázat táblázat azonosítójával, és táblázat azonosító bővítményeik a letöltött könyvtár táblázatból meghatározott táblázat azonosító bővítmények, és ezzel betöltse az adat betöltőt.

28. A 26. vagy 27. igénypont szerinti vevő és dekódoló berendezés, **azzal jellemezve**, hogy a letöltő eszköz úgy van kialakítva, hogy egy előre meghatározott táblázat azonosítóval rendelkező könyvtár táblázatot töltsön le, amely egy vevő és dekódoló berendezés (2020) minden egyes verzió azonosítója részére egy ahhoz hozzárendelt táblázat azonosítót tartalmaz, továbbá, hogy meghatározza a vevő és dekódoló berendezés (2020) verzió azonosítóját, valamint, hogy a vevő és dekódoló berendezés (2020) verzió azonosítójához tartozó táblázat azonosítójú, az előre meghatározott táblázat azonosító bővítménnyel rendelkező könyvtár táblázatot töltsön le.

29. A 27. vagy 28. igénypont szerinti vevő és dekódoló berendezés, **azzal jellemezve**, hogy a letöltő eszköz úgy van kialakítva, hogy meghatározza, vajon az éppen továbbított könyvtár táblázat könyvtár verzió azonosítója fiatalabb-e egy korábban letöltött, az éppen továbbított könyvtár táblázat táblázat azonosítójával azonos táblázat azonosítójú könyvtár



táblázat könyvtár verzió azonosítójánál, és amennyiben nem, úgy az adat betöltő letöltését megszakítsa.

30. A 20-29. igénypontok bármelyike szerinti vevő és dekódoló berendezés, **azzal jellemezve**, hogy az adat betöltő a vevő és dekódoló berendezés (2020) hardver felépítésére jellemző kód alakjában van kiképezve.

31. A 20-30. igénypontok bármelyike szerinti vevő és dekódoló berendezés, **azzal jellemezve**, hogy a letöltő eszköz a bitfolyamba illesztett adatokba beépített, az elsőként említett adat betöltő és az adatok letöltéséhez szükséges második adat betöltőt letöltő eszközként van kialakítva.

32. Átviteli rendszer, **azzal jellemezve**, hogy adatokat egy vevő és dekódoló berendezésbe (2020) letöltő legalább egy betöltőt magában foglaló bitfolyam, valamint a legalább egy adat betöltővel társított adatok továbbítására szolgáló eszköze, valamint a legalább egy adat betöltőt modulokra osztó, és a legalább egy adat betöltővel társított adatokat megfelelő számú modulra osztó eszközzel rendelkezik a moduloknak a továbbító eszközzel történő továbbítására.

33. A 32. igénypont szerinti átviteli rendszer, **azzal jellemezve**, hogy a legalább egy adat betöltő minden egyes modulját egy-egy olyan táblázatként kialakító eszköze van, ahol a legalább egy adat betöltő táblázatai azonos táblázat azonosítóval, és egymástól eltérő táblázat azonosító bővítményekkel rendelkeznek, valamint a legalább egy adat betöltővel társított adat modulok mindegyikét egy-egy olyan táblázatként kialakító eszköze van, ahol az adat modulok táblázatainak a táblázat azonosítója megegyezik a velük társított adat betöltő modulok táblázatainak a táblázat azonosítóival, és különböző táblázat azonosító bővítményekkel vannak ellátva.

34. A 33. igénypont szerinti átviteli rendszer, **azzal jellemezve**, hogy a táblázatoknak egy előre meghatározott táblázat azonosító bővítménytől eltérő, különböző táblázat azonosító bővítményük van, továbbá a rendszer az azonos táblázat azonosítójú modul vagy modulok mindegyike számára egy-egy könyvtár táblázatot előállító eszközzel rendelkezik, ahol minden egyes könyvtár táblázatnak ugyanaz a táblázat azonosítója, valamint az előre meghatározott táblázat azonosító bővítménye van, és a könyvtár minden egyes modul részére tartalmazza a modul nevét és táblázat azonosító bővítményét.



35. A 32-34. igénypontok bármelyike szerinti átviteli rendszer, *azzal jellemezve*, hogy előre meghatározott táblázat azonosítójú, és egy vevő és dekódoló berendezés (2020) minden egyes verzió azonosítója számára egy azzal társított táblázat azonosítót tartalmazó könyvtár táblázatot előállító eszközzel van ellátva.

36. A 32-35. igénypontok bármelyike szerinti átviteli rendszer, *azzal jellemezve*, hogy minden egyes továbbított táblázatba egy arra vonatkozó verzió azonosítót beépítő eszközt tartalmaz.

37. A 32-36. igénypontok bármelyike szerinti átviteli rendszer, *azzal jellemezve*, hogy a legalább egy adat betöltő a vevő és dekódoló berendezés (2020) hardver felépítésére jellemző kód alakjában van megvalósítva.

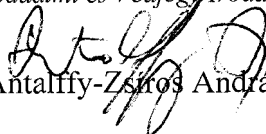
38. Rendszer, *azzal jellemezve*, hogy a 20-31. igénypontok bármelyike szerinti vevő és dekódoló berendezésből (2020) és 32-37. igénypontok bármelyike szerinti átviteli rendszerből van összeépítve.

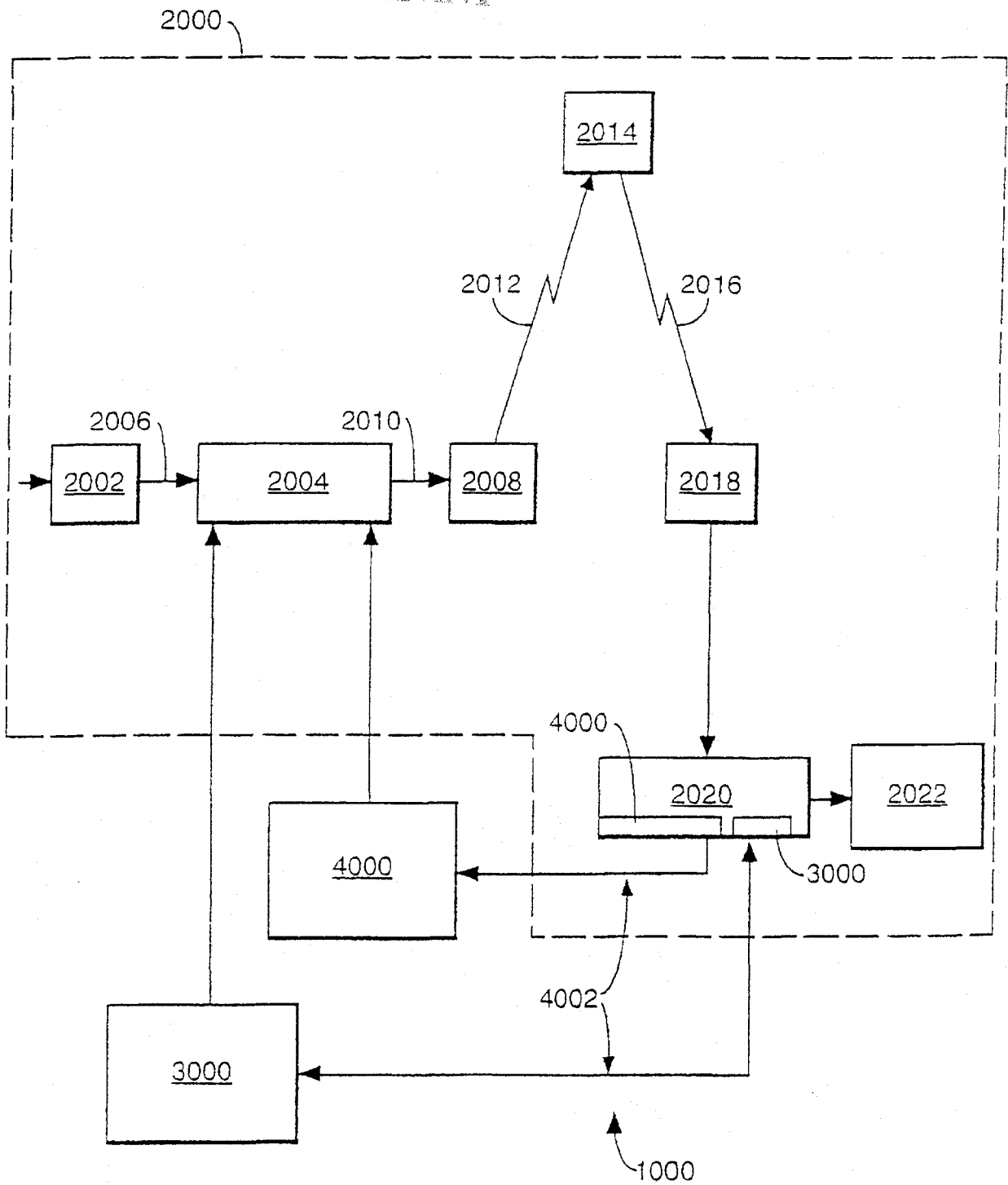
39. Jel, *azzal jellemezve*, hogy adatokat egy vevő és dekódoló berendezésbe (2020) letöltő legalább egy adat betöltőt, továbbá a legalább egy adat betöltővel társított adatot foglal magában, ahol a legalább egy adat betöltő több modulra van felosztva, és a legalább egy adat betöltővel társított adat megfelelő számú modulra van felosztva.

VÁRHEGYI C.

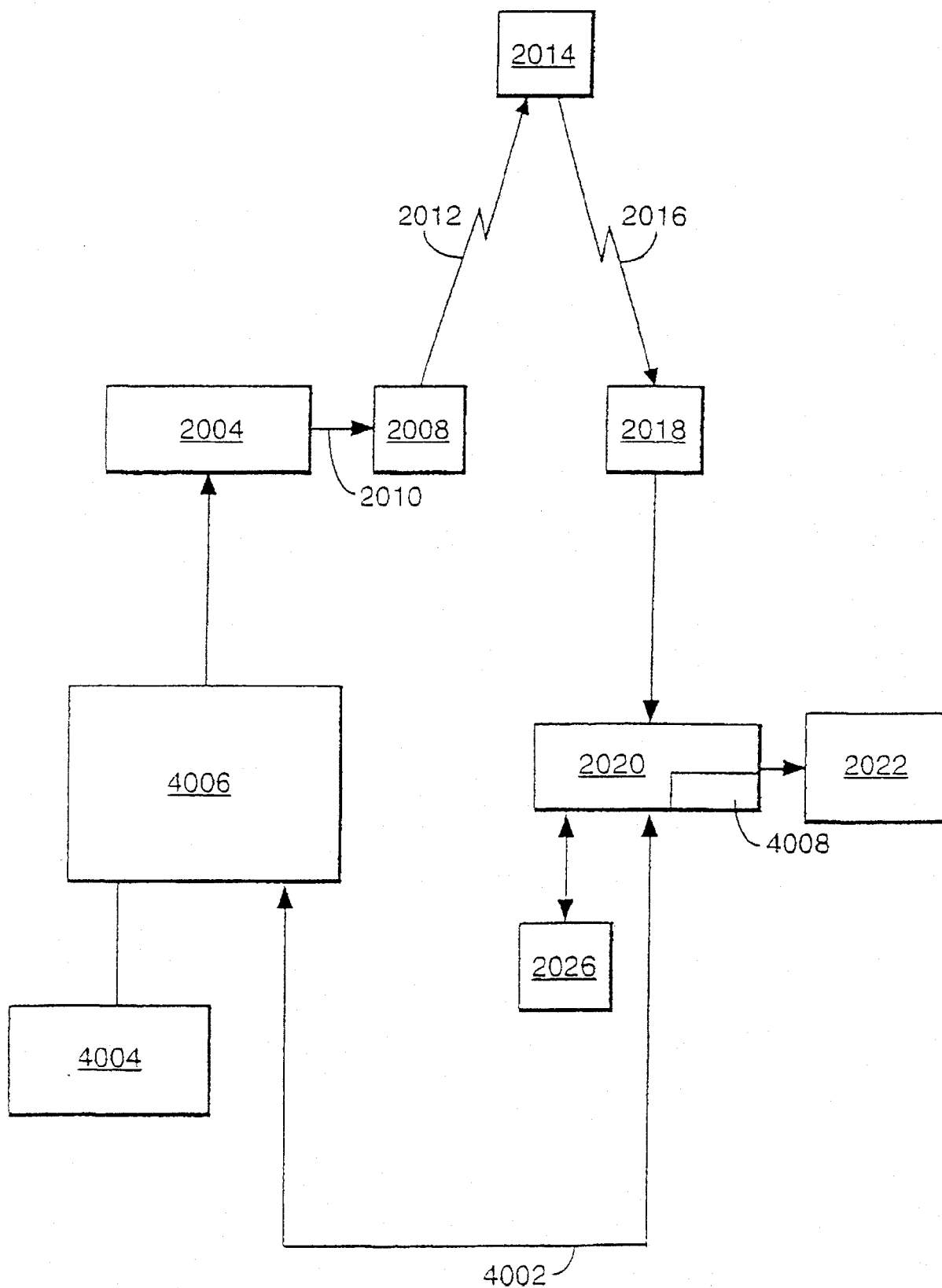
A meghatalmazott:

DANUBIA
Szabadalmi és Védjegy Iroda Kft.


Antalffy-Zsuzsanna András



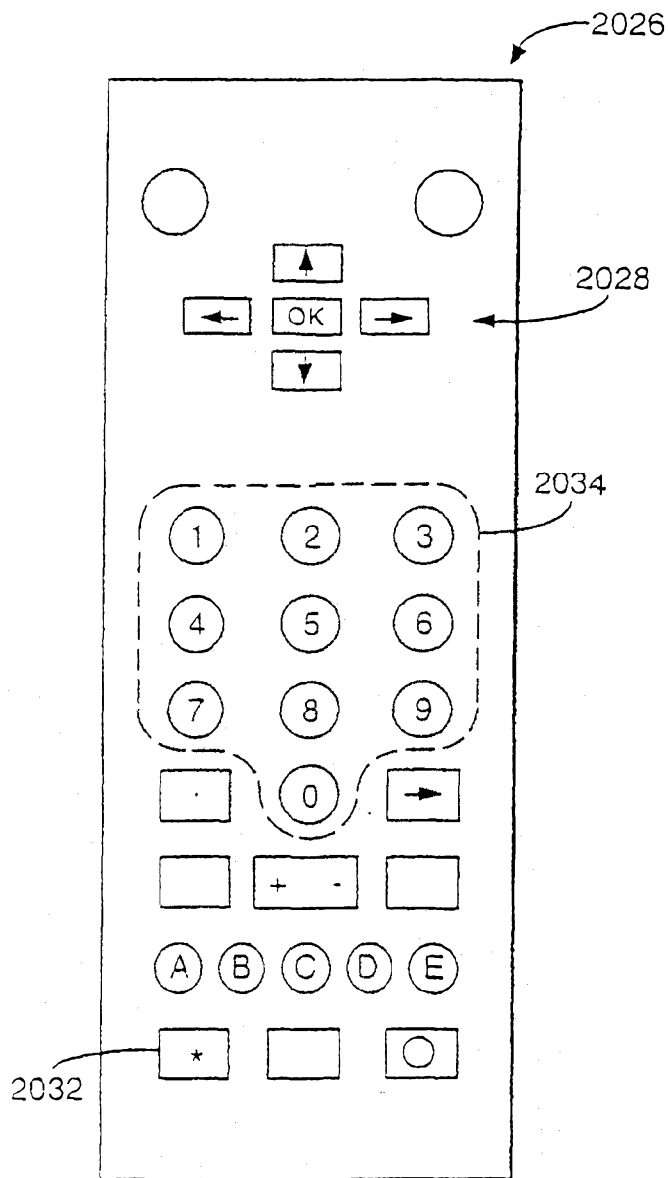
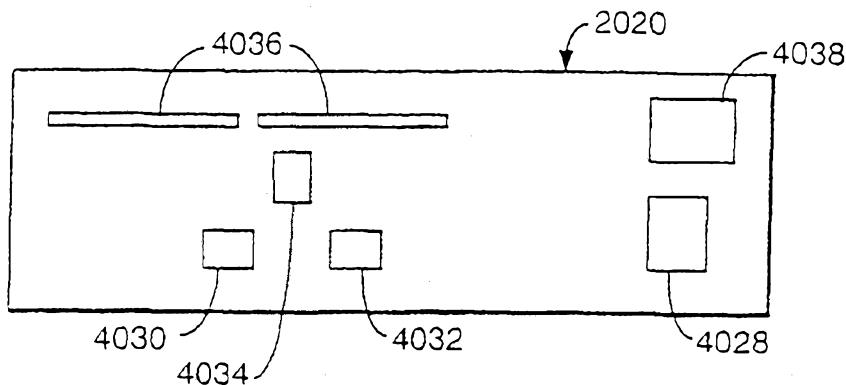
1. ÁBRA



2. ÁBRA

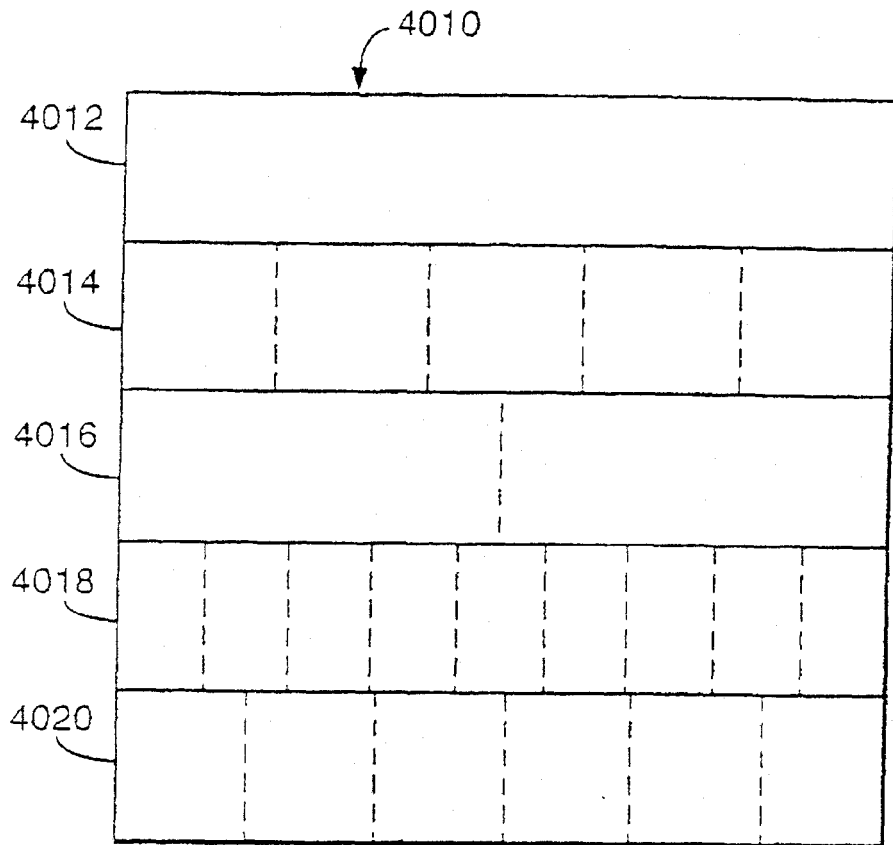
KÖZZETÉLJI
PÉLDÁNY

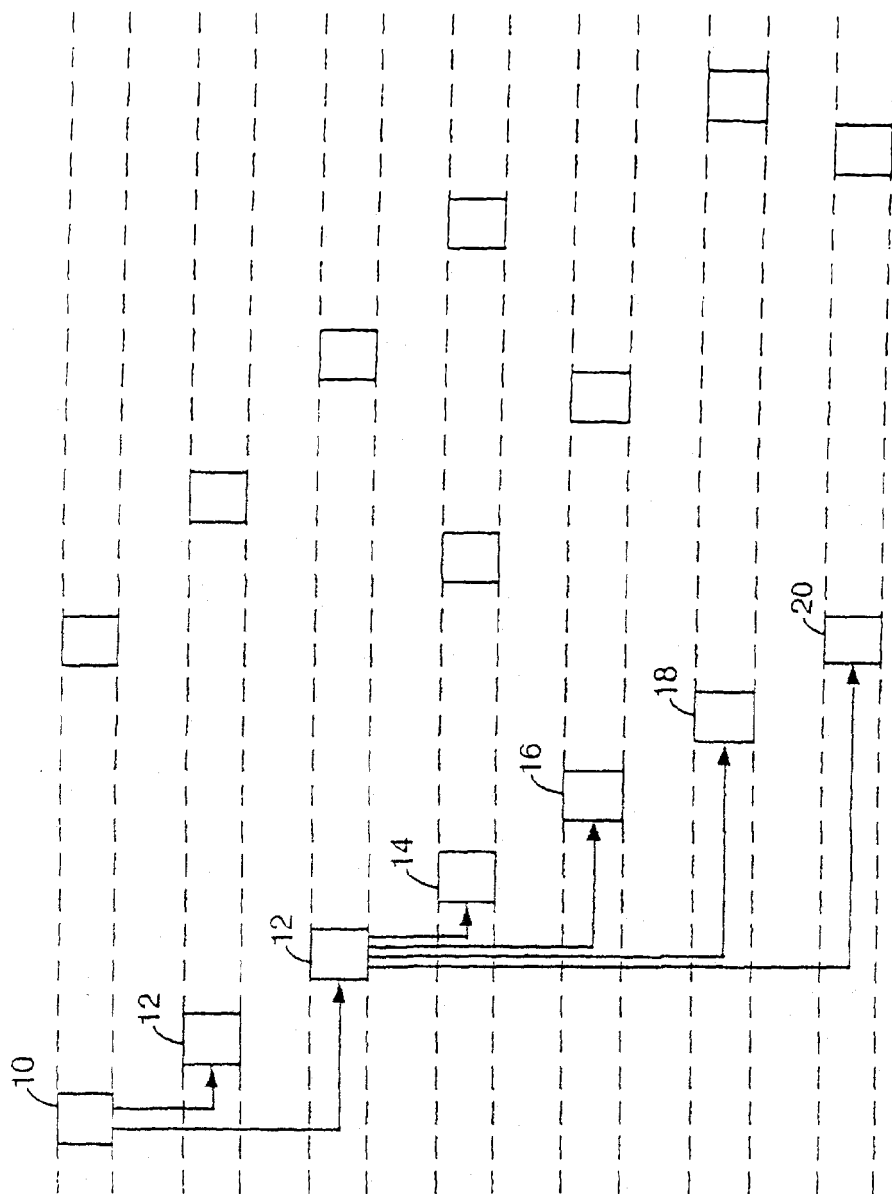
3. ÁBRA



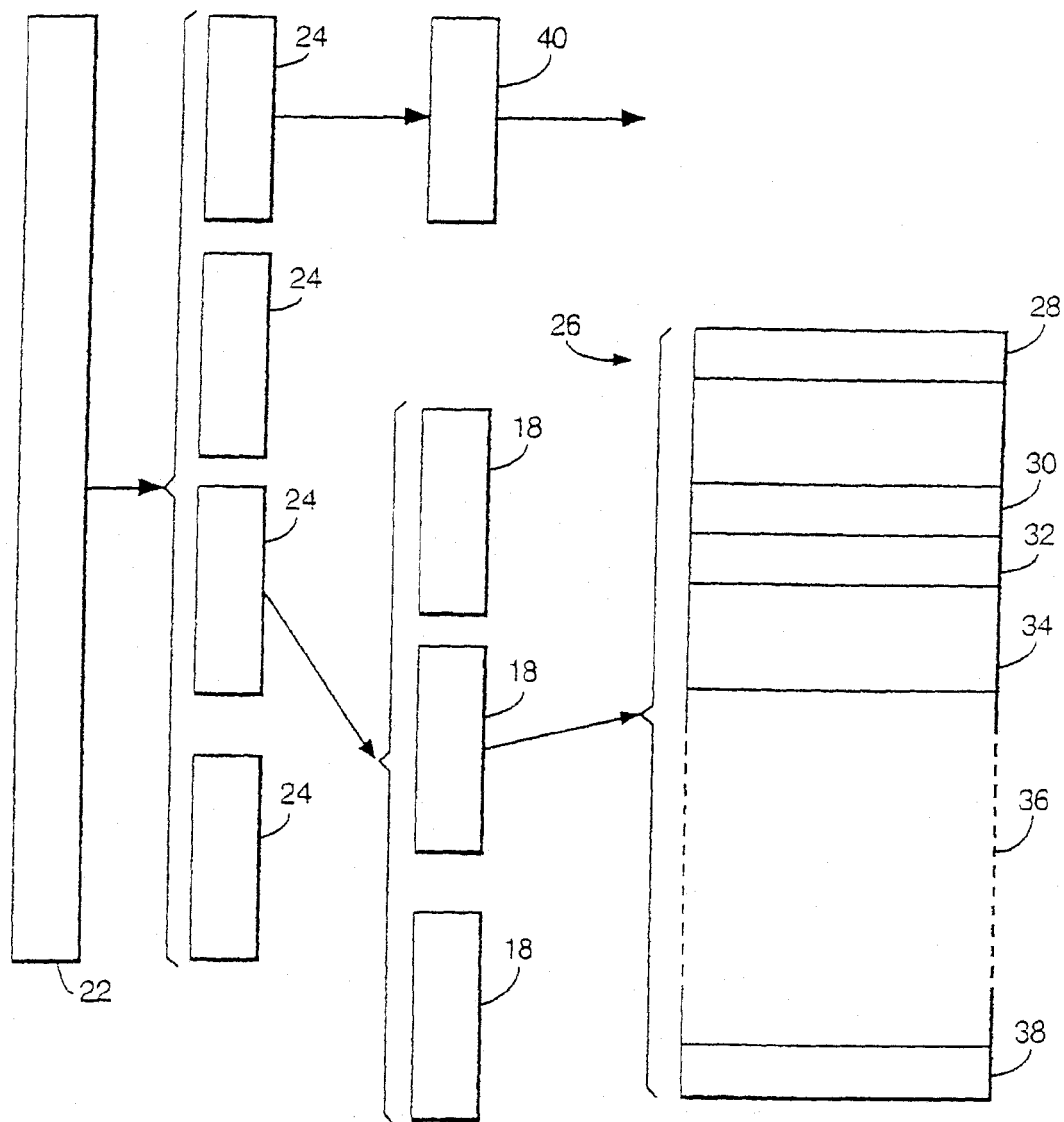
4. ÁBRA

5. ÁBRA

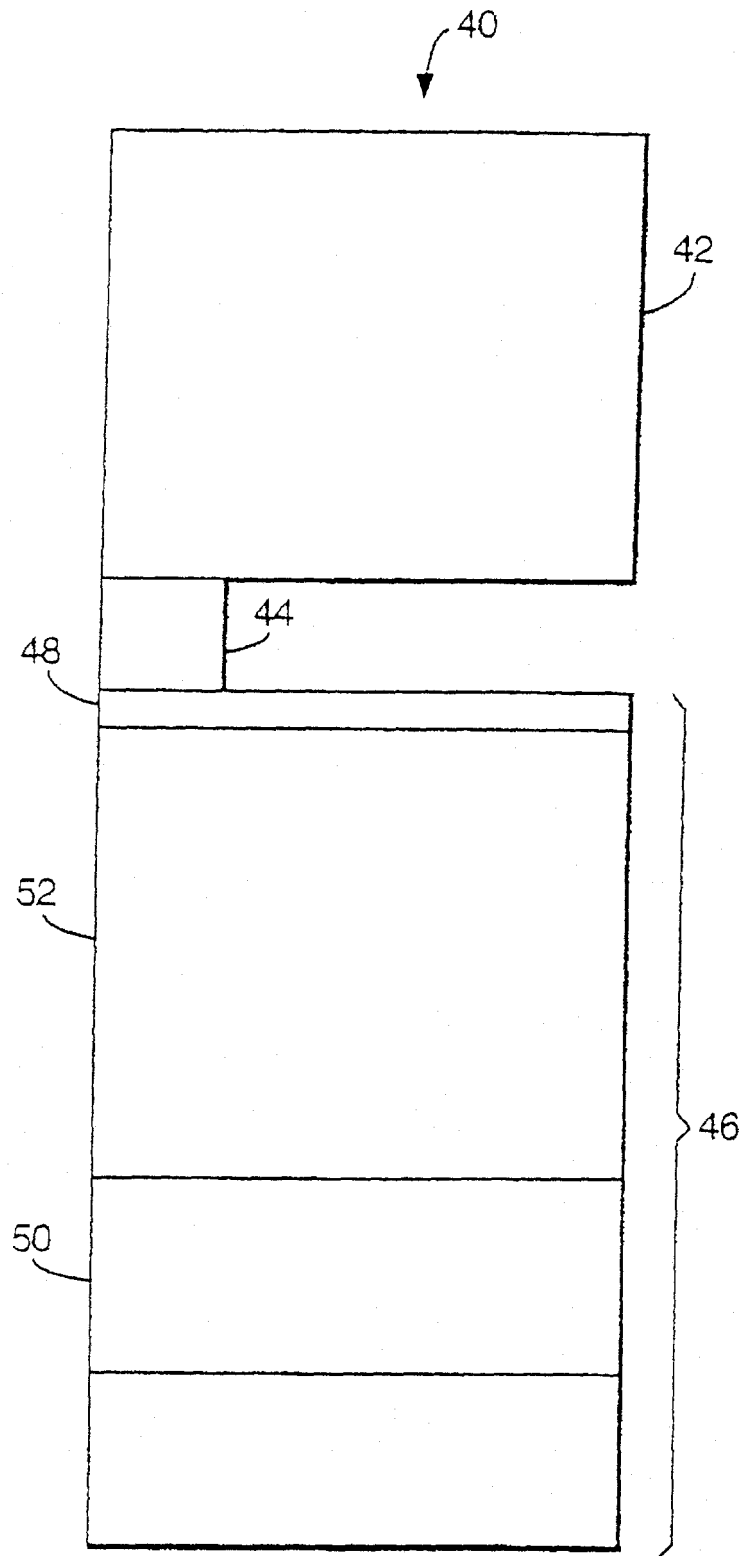




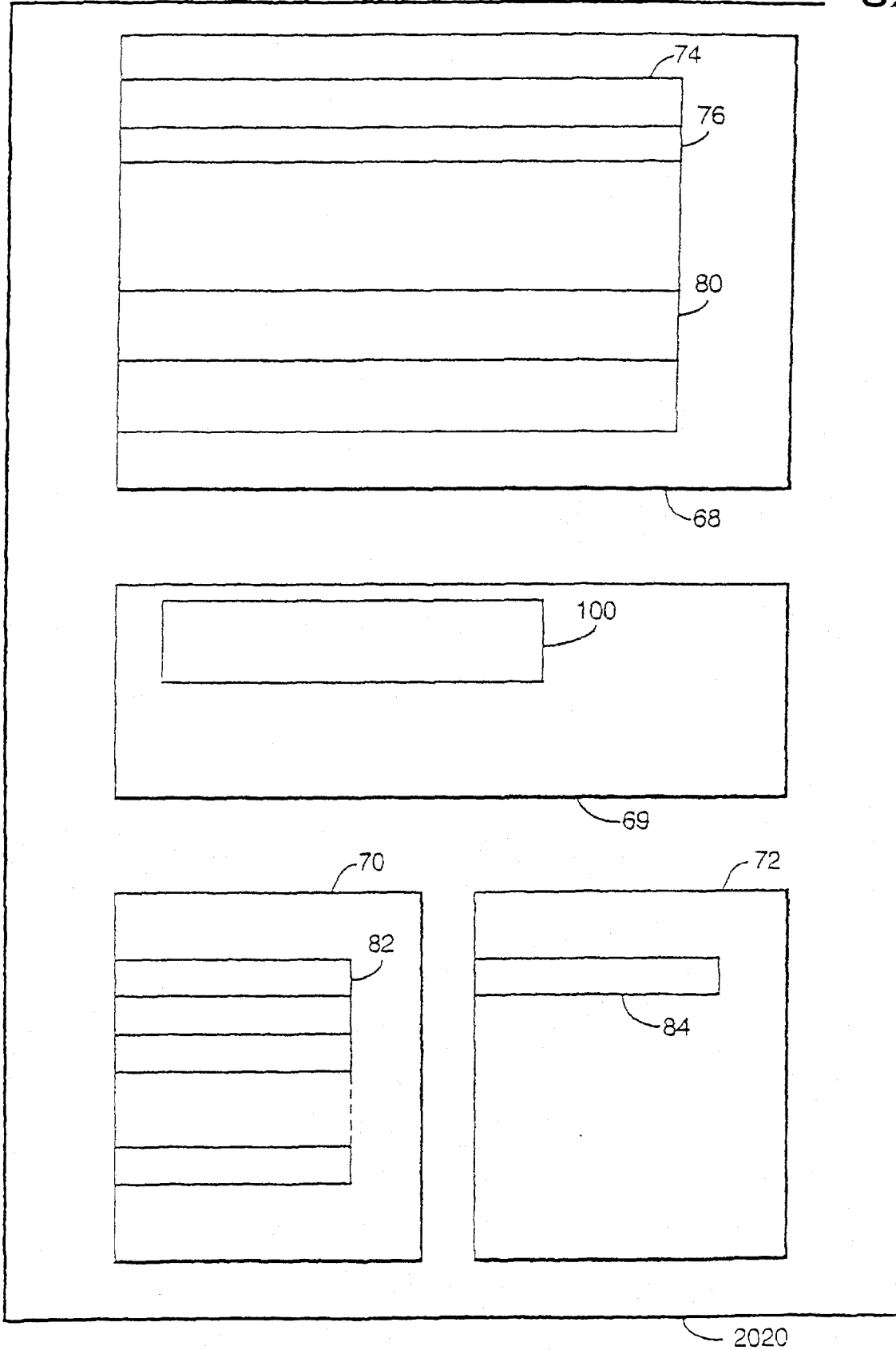
6. ÁBRA



7. ÁBRA

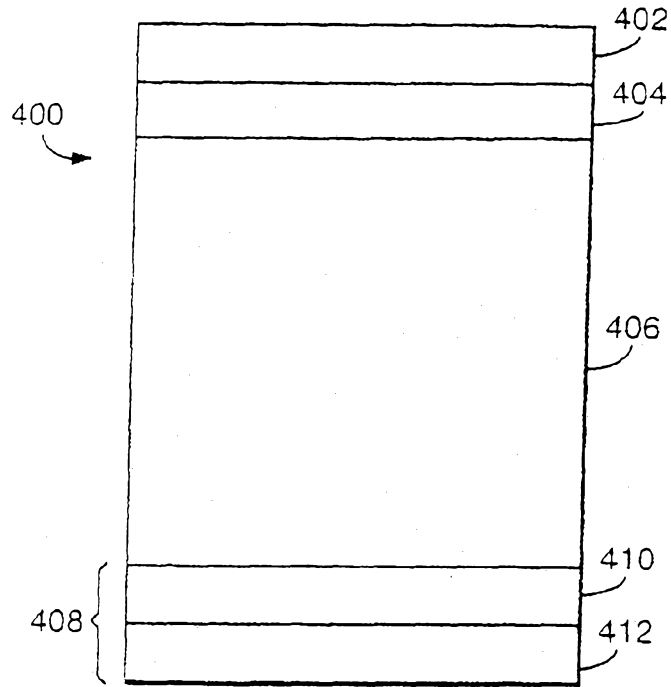


8. ÁBRA

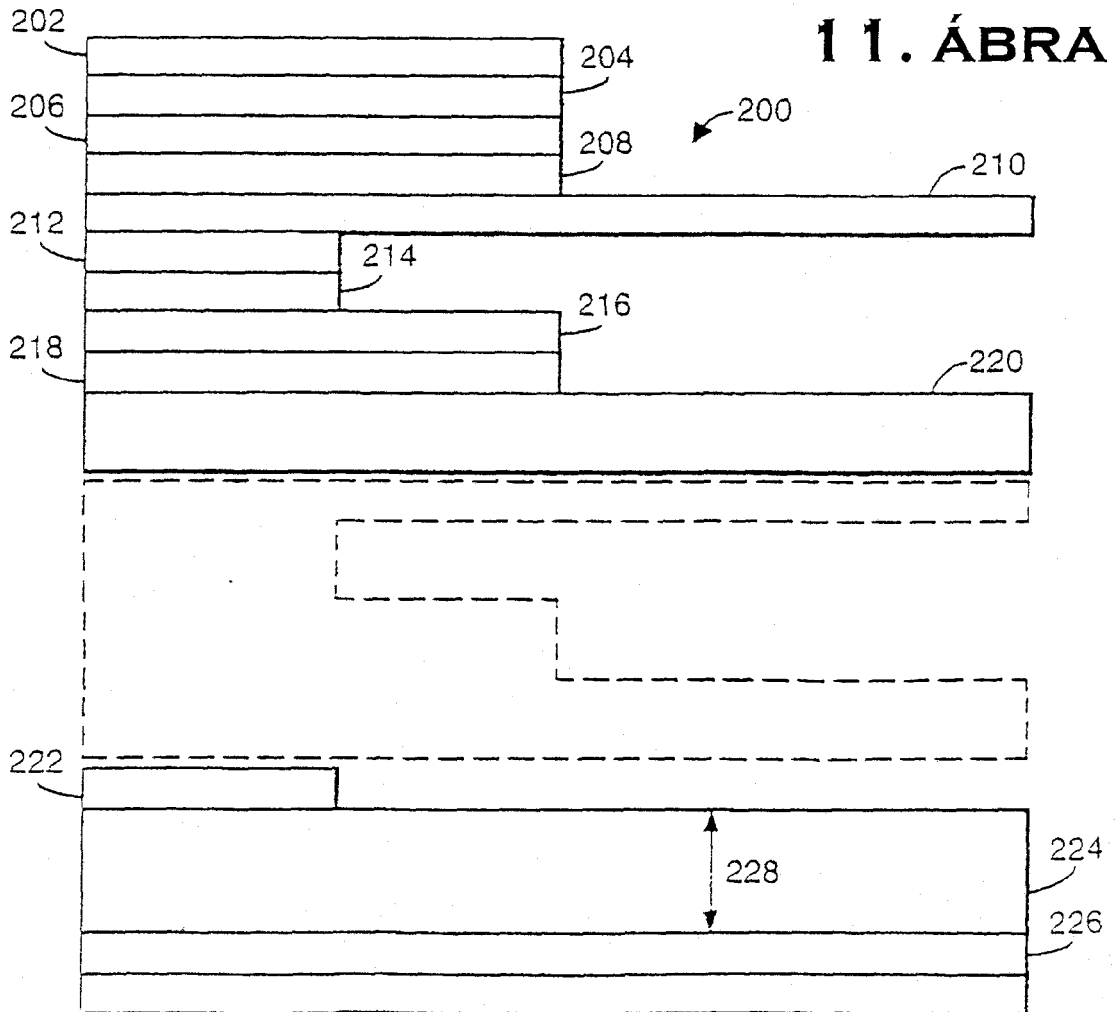


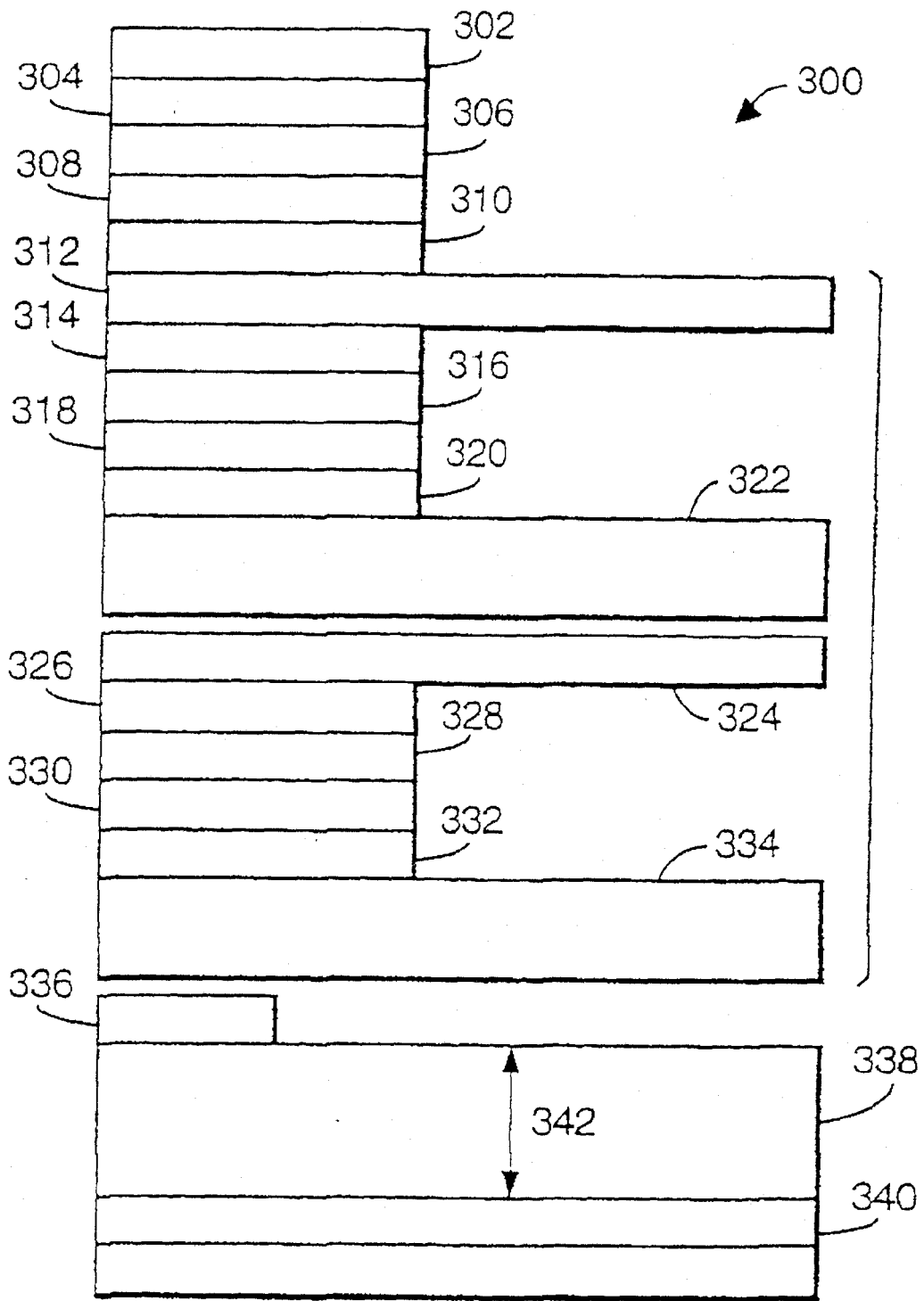
9. ÁBRA

10. ÁBRA

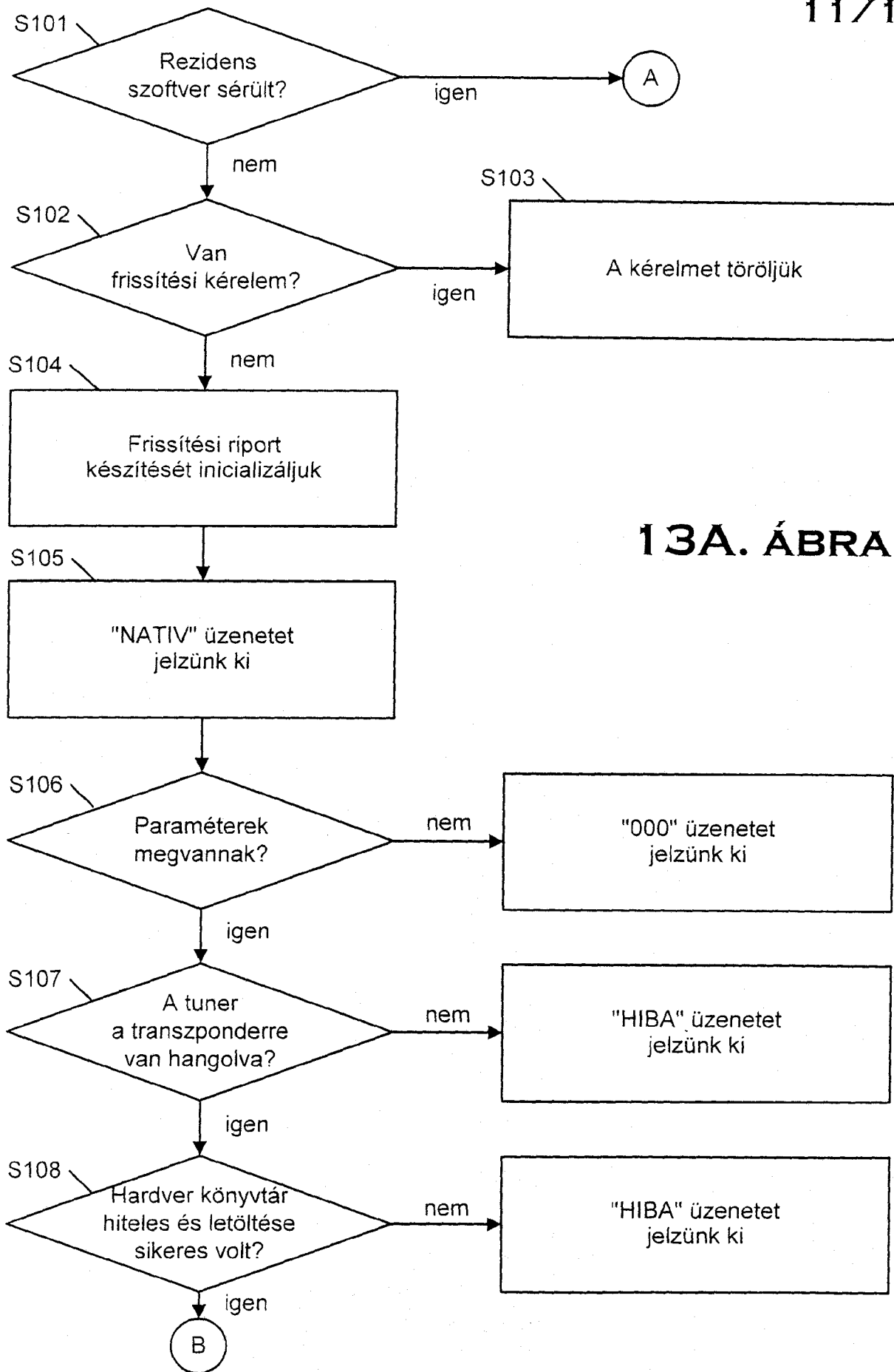


11. ÁBRA

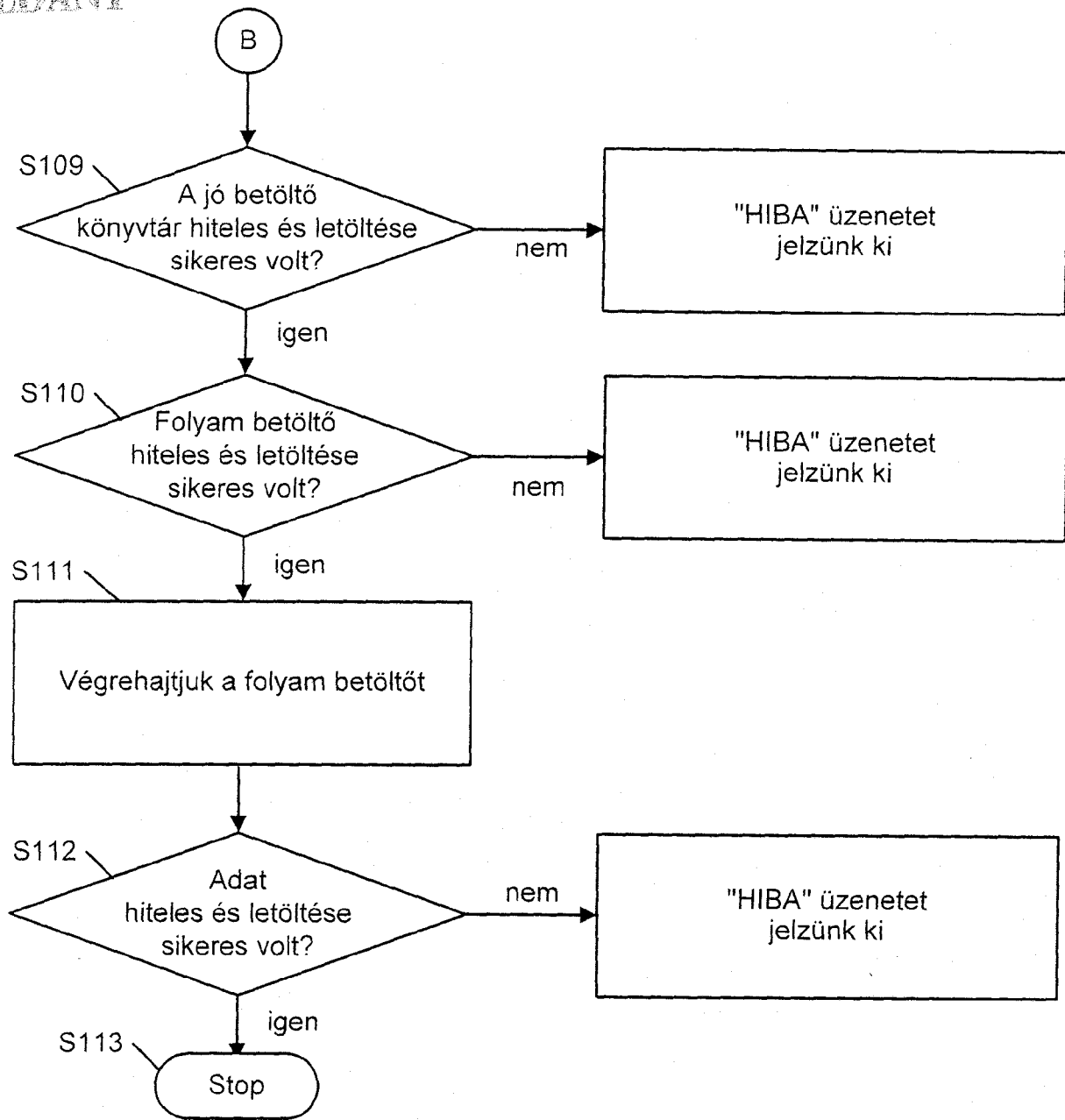




12. ÁBRA

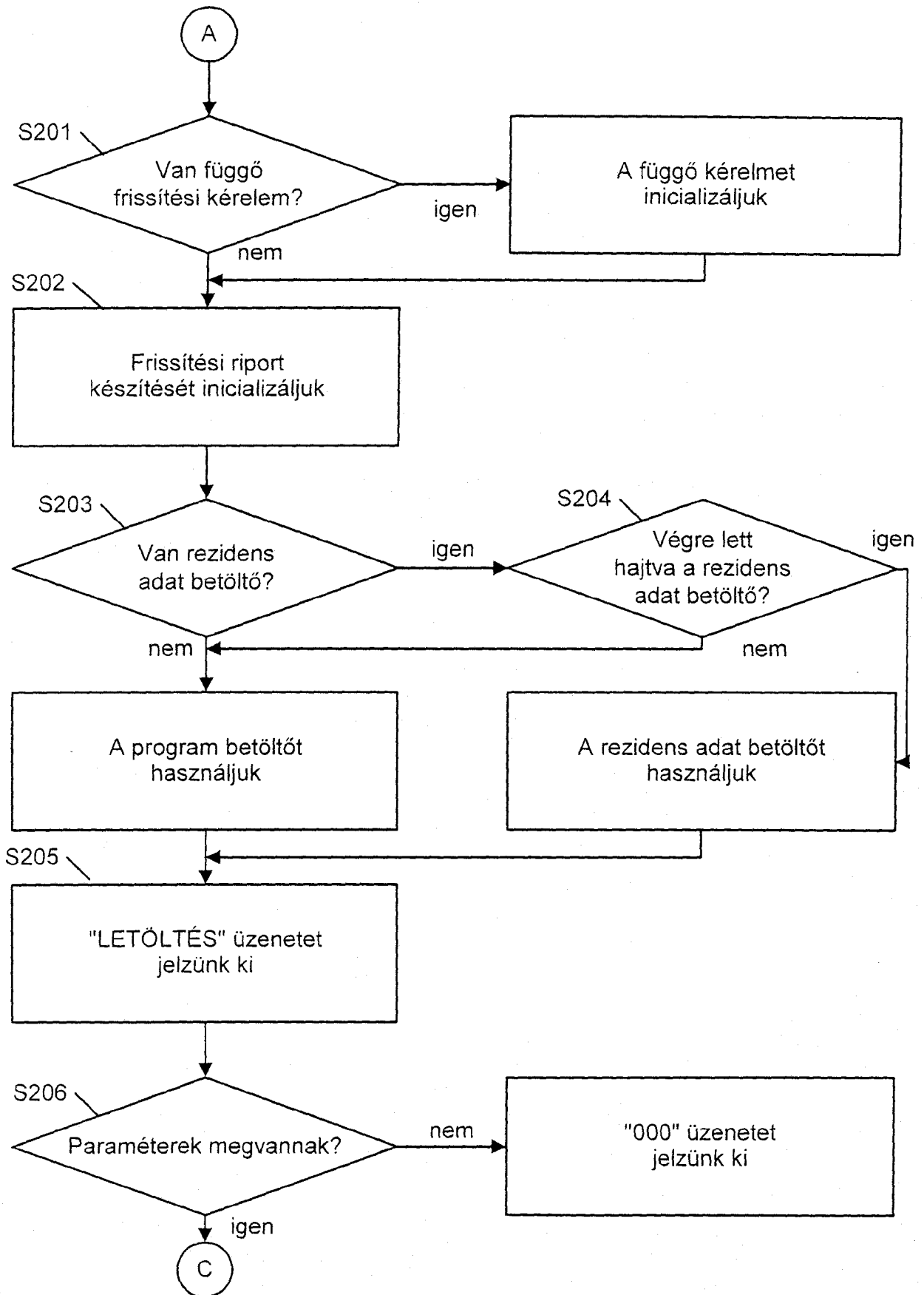


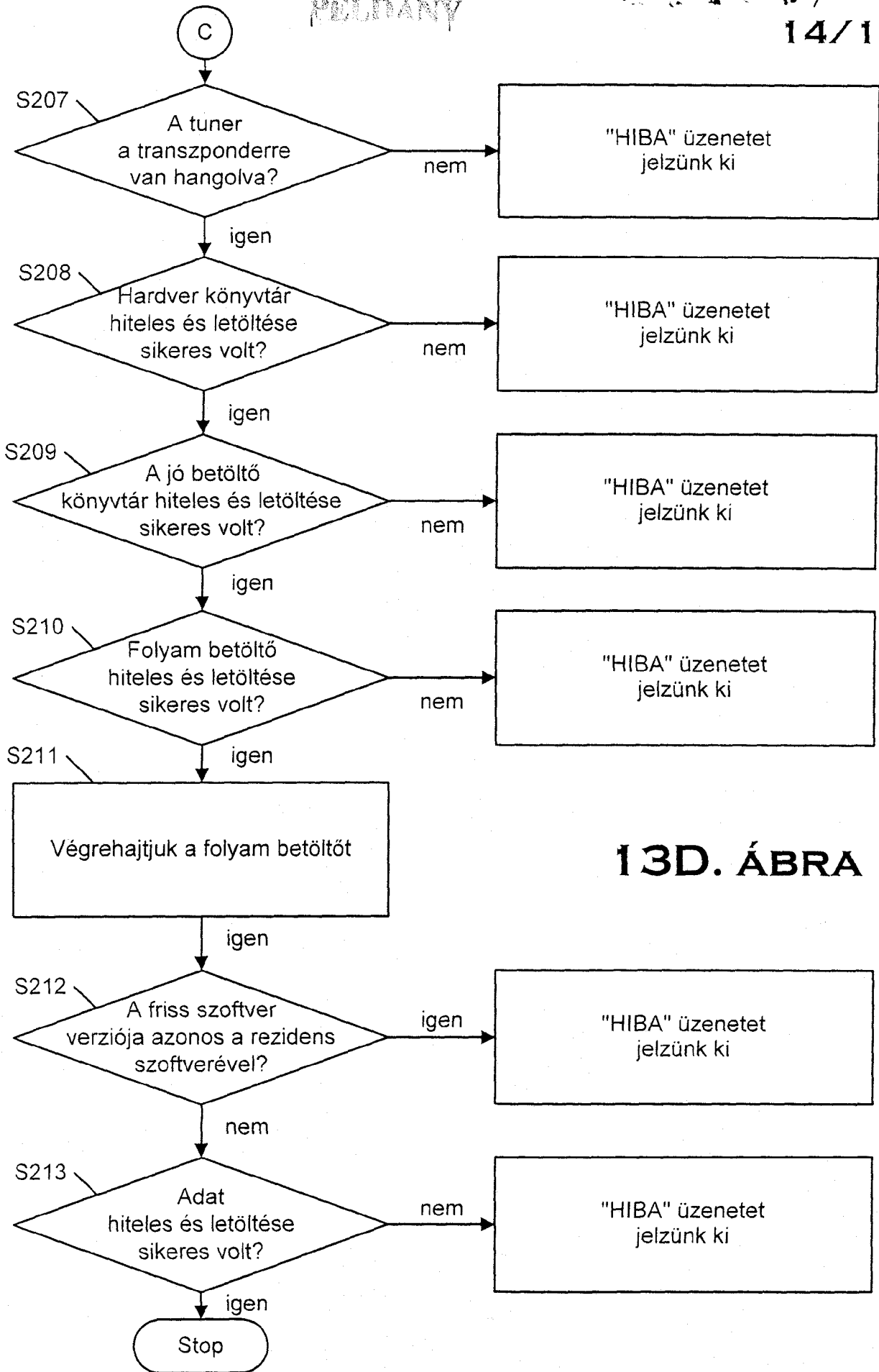
13A. ÁBRA



13B. ÁBRA

13C. ÁBRA





13D. ÁBRA