(19) Országkód

HU

MAGYAR KÖZTÁRSASÁG

MAGYAR SZABADALMI HIVATAL

SZABADALMI LEÍRÁS

(11) Lajstromszám:

215 877 B

(21) A bejelentés ügyszáma: P 95 02569
(22) A bejelentés napja: 1994. 01. 24.
(30) Elsőbbségi adatok:
08/025,538 1993. 03. 03. US
(86) Nemzetközi bejelentési szám: PCT/US 94/00826
(87) Nemzetközi közzétételi szám: WO 94/20904

(51) Int. Cl.[6]

G 06 F 9/46
H 04 M 3/50

(40) A közzététel napja: 1995. 11. 28.
(45) A megadás meghirdetésének a dátuma a Szabadalmi Közlönyben: 1999. 03. 29.

(72) Feltalálók:
Bipin, Patel, San Jose, Kalifornia (US)
Ichnowski, Jeanne, Palo Alto, Kalifornia (US)
Kaminsky, Mark E., Sunnyvale, Kalifornia (US)
Perelman, Roberto, Sunnyvale, Kalifornia (US)
Yuan, Chris, Fremont, Kalifornia (US)

(73) Szabadalmas:
Siemens Rolm Communications Inc., Santa Clara, Kalifornia (US)

(74) Képviselő:
S. B. G. & K. Budapesti Nemzetközi Szabadalmi Iroda, Budapest

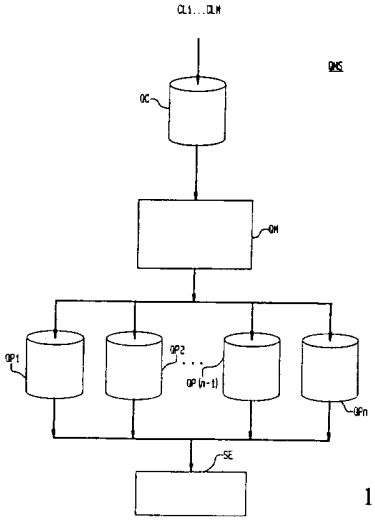
(54) **Eljárás és berendezés sorban állás vezérlésére, valamint üzenettároló és -továbbító rendszer**

KIVONAT

A találmány tárgya sorban állást vezérlő rendszer (QMS) több különböző típusú felhasználó (CL1–CLM) kiszolgálására. A rendszernek a sorban álló felhasználókat (CL1–CLM) előnyösen érkezési sorrendben fogadó és tartó vezérlősora (QC); továbbá a felhasználók (CL1–CLM) fogadására, és a felhasználóktól (CL1–CLM) való időszakonkénti kiürítésre több feldolgozósora (QP1–QPn), valamint a vezérlősorból (QC) a felhasználókat feldolgozósorokba (QP1–QPn) átvivő sorvezérlője (QM) van. A találmány különlegessége, hogy a felhasználók típusainak számánál kevesebb számú, a különböző felhasználók (CL1–CLM) típusához hozzárendelhető feldolgozósora (QP1–QPn), valamint a vezérlősorból (QC) a felhasználókat (CL1–CLM) egy, más felhasználóktól (CL1–CLM) vagy más típusú felhasználóktól (CL1–CLM) mentes feldolgozósorba (QP1–QPn) átvivő, és a feldolgozósor (QP1–QPn) kiürüléséig a feldolgozósort (QP1–QPn) az adott egy vagy több felhasználó (CL1–CLM) típusához hozzárendelő sorvezérlője (QM) van.

A találmány tárgya továbbá eljárás sorban álló, különböző típusú üzenetek feldolgozásának vezérlésére. Ezekhez a különböző típusú üzenetekhez különböző típusú feldolgozóművelet, különösen továbbítási művelet tartozik. A találmányi eljárás során az üzeneteket egy vezérlősorba juttatjuk, az üzenetek típusainál kevesebb számú, több feldolgozósort időszakonként kiürítünk, és a vezérlősorból az üzenetet olyan feldolgozósorba juttatjuk, amely mentes másfajta típusú üzenetektől, és a feldolgozósort annak kiürüléséig egyetlen üzenettípushoz rendeljük.

A találmány tárgya még az eljáráson alapuló üzenettároló és -továbbító rendszer.



1. ábra

A leírás terjedelme 26 oldal (ezen belül 11 lap ábra)

HU 215 877 B

A találmány tárgya eljárás és berendezés, amely célja sorban álló, különböző szolgáltatásokat igénylő felhasználók hozzárendelése olyan kiszolgálókhoz (server), amelyek a kívánt szolgáltatásokat nyújtják. A találmány tárgya különösen olyan eljárás és berendezés, amely rögzített telefonüzenetek sorának kezelésére szolgál, és lehetővé teszi az üzenetek csoportos továbbítását a célállomás felé.

Az ilyen sorban állást vezérlő rendszerek a felhasználók típusától függően különböző feldolgozási igényű felhasználókhoz rendelnek hozzá szolgáltatásokat. Minden egyes, például A, B, C stb. típusú felhasználó számára egy adott osztályú, például A, B, C stb. kiszolgálóegységet rendelnek hozzá, amely képes a felhasználó kiszolgálására. Az is ismert azonban, hogy olyan univerzális kiszolgálóegységet használnak, amely különböző osztályú, például A, B, C stb. osztályú kiszolgálóvá képes átalakulni, a felhasználó igénye szerint.

A legtöbb sorban állást vezérlő rendszer a felhasználókat előre meghatározott sorrendben szolgálja ki, például érkezési sorrendben (FIFO=first-in-first-out). A kiszolgálás két ismert eljárás alapján történik, nevezetesen az egysoros stratégiával vagy a többsoros stratégiával.

Az egysoros stratégiánál az összes felhasználó ugyanabban a sorban várakozik, érkezési sorrendben, és egyenként kerül kiszolgálásra. Ha a sorban legelöl álló felhasználó (client) ugyanolyan típusú, mint az éppen kiszolgálás alatt levő, például A típusú, akkor az A típusú felhasználót kiszolgáló vagy feldolgozó kiszolgálóegység (server) azonnal feldolgozhatja a következő A típusú felhasználót is. Egyébként a rendszernek váltania kell egy másik, a következő felhasználó típusának megfelelő osztályú kiszolgálóra, például B típusnak megfelelőre, vagy a sorban meg kell keresni a következő olyan felhasználót, amelynek típusa megfelel az elsőként feldolgozott, például A típusnak.

Az ismert telefonos üzenettovábbító rendszereknél a felhasználók rögzített telefonüzenetek formájában jelennek meg, amelyeket különböző célállomások kódjával látnak el. Itt az egysoros stratégia a következőképpen működik: Az összes üzenet egy sorban áll, érkezési sorrendben, és kiszolgáló egyenként szolgálja ki és továbbítja az egyes üzeneteket. Ha két egymást követő üzenetnek ugyanaz a célállomáskódja, akkor a rendszer ezeket egyszerre, csoportonként (batch) képes továbbítani. Egyébként a rendszer a célállomást a következő üzenet célállomására változtatja, vagy megkeresi a sorban azokat az üzeneteket, amelyek ugyanarra a célállomásra mennek, mint az első üzenet, és kigyűjt több, közös célállomásra tartó üzenetet, és csoportosan továbbítja azokat.

Az ilyen egysoros rendszerek nyilvánvalóan rossz hatékonysággal működnek. Csak nehézkesen képesek különböző felhasználók párhuzamos feldolgozására.

A többsoros stratégia több sort alkalmaz, és minden egyes sort egy adott típusú felhasználóhoz, és az annak megfelelő osztályú kiszolgálóhoz rendel hozzá. Az ugyanolyan típusú felhasználók ugyanabban a sorban várnak, érkezési sorrendben. A többsoros stratégiát használó telefonos üzenettovábbító rendszereknél a felhasználók különböző célállomásokra szánt rögzített üzenetek. A több sor mindegyike egy meghatározott célállomásra szánt üzenetet tartalmaz. A közös célállomásra tartó üzenetek ugyanabban a sorban várakoznak, érkezési sorrendben. Az ilyen többsoros elrendezéseknél ugyanannyi számú sorra van szükség, amennyi a célállomások száma, amely szám nagyon nagy, vagy akár ismeretlen is lehet. Az egyes üzeneteknek a célállomáshoz rendelt sorhoz kell jutniuk.

Az EP-A-0236013 számú közzétételi irat olyan többsoros sorban állást vezérlő rendszert mutat be, amelyben a különböző felhasználókat olyan feldolgozósorokba terelik, amelyek az adott típusú felhasználókhoz rögzített módon vannak hozzárendelve. Ezért legalább annyi feldolgozósorra van itt is szükség, mint ahány felhasználó van.

A Krzesinski és társai által jegyzett „Product Form Solutions for Multiserver Centres with Hierarchical Concurrency Constraints" című cikkben (Proceedings of the 6th South African Computer Symposium, 1991. július 2., Caledon, Dél-Afrika, 69–82. oldal) olyan általános sorban állást kezelő modellt ismertetnek, amely hierarchikus felépítésű, és amely az egyidejűségi megszorításoknak kitett sorban állási rendszereken alapul. A cikk matematikai módszerekkel elemzi az ilyen rendszerek tulajdonságait. Ennél a modellnél az az alapfeltételezés, hogy az összes felhasználótípus és a hozzájuk tartozó megszorítások előre meghatározottak. Nem ad kitanítást olyan rendszerekről, ahol a különböző típusú sorok dinamikusan változnak, és ezért változnak az egyidejűségi megszorítások is.

A C. H. Chien és társai által publikált „Paradigm: An Architecture for Distributed Vision Processing" című cikk (Proceedings of the 10th International Conference on Pattern Recognition, 1990. június 16., Atlantic City, N.J., USA, 648–653 oldal) ismertet egy olyan rendszert, amelyben különböző processzorokhoz különböző feladatokat osztanak le. Azonban ez a rendszer mesterséges látás feldolgozásával foglalkozik, és a feladatok csoportosítása elsősorban térbeli tulajdonságokon alapul, például közös kétdimenziós részletekkel bíró feladatokat egy ehhez rendelt processzorra oszt le. Nem ismerhető meg belőle egy olyan rendszer, amely új célállomásokat tetszőlegesen hozzáférhetővé váló processzorokhoz rendel hozzá.

A találmány célja a sorban állást vezérlő rendszerek javítása, különösen a hangpostát használó telefonhálózatoknál. A találmány célja még az előbbiekben ismertetett hátrányok kiküszöbölése.

A találmány szerint a fenti célt olyan, a bevezetőben ismertetett típusú sorban állást vezérlő rendszerrel érjük el, amelynek a sorban álló felhasználókat sorrendben fogadó és tartó vezérlősora; továbbá a felhasználók fogadására és a felhasználóktól való időszakonkénti kiürítésre több feldolgozósora van, valamint a vezérlősorból a felhasználókat feldolgozósorokba átvivő sorvezérlőt tartalmaz. A találmány értelmében a rendszernek a felhasználók típusainak számánál kevesebb számú, a különböző felhasználók típusához hozzárendelhető feldol-

gozósora van. Emellett a rendszernek a vezérlősorból a felhasználókat egy, más felhasználóktól vagy más típusú felhasználóktól mentes feldolgozósorba átvivő, és a feldolgozósor kiürüléséig a feldolgozósort az adott egy vagy több felhasználó típusához hozzárendelő sorvezérlője van.

Ha a feldolgozósor valamikor kiürül, akkor a sorvezérlő a kiürített feldolgozósort a következő, a feldolgozósorba továbbított felhasználó típusához rendeli hozzá.

Mivel a sorvezérlő az egyes feldolgozósorokba mindig olyan típusú felhasználót irányít, amilyen típusú felhasználókat a feldolgozósor már tartalmaz, ezért a sorvezérlő önmagától az egyes feldolgozósorokat ahhoz a felhasználótípushoz rendeli hozzá, amilyen típusú felhasználót a sorvezérlő elsőként helyez az adott feldolgozósorba. Ez a hozzárendelés addig tart, amíg a kiszolgáló kiüríti a feldolgozósort. Ha ezt követően a sorvezérlő egy új típusú felhasználót helyez a kiürített feldolgozósorba, akkor a sorvezérlő a feldolgozósort átirányítja a korábbi felhasználótípustól az újabb felhasználótípushoz. Ez az átirányítás és hozzárendelés lehetővé teszi, hogy korlátozott számú, például ötven feldolgozósor bármekkora számú felhasználótípust dolgozzon fel, például ezret vagy annál is többet.

A találmány tárgya még eljárás sorban álló, különböző típusú üzenetek feldolgozásának vezérlésére egy üzenettovábbító rendszerben, különösen hangposta-továbbító rendszerben, ahol a különböző típusú üzenetekhez különböző típusú feldolgozóművelet, különösen továbbítási művelet tartozik. A találmány értelmében az üzeneteket egy vezérlősorba juttatjuk, majd az üzenetek típusainál kevesebb számú, több feldolgozósort időszakonként kiürítünk, és végül a vezérlősorból az üzenetet olyan feldolgozósorba juttatjuk, amely mentes másfajta típusú üzenetektől, és a feldolgozósort annak kiürüléséig egyetlen üzenettípushoz rendeljük.

A találmány egy másik megvalósításánál, telefonos hangposta- vagy üzenettovábbító rendszereknél a felhasználók több különböző célállomáskóddal ellátott rögzített hangpostaüzenetek. Ezért a találmány tárgya még olyan üzenettároló és -továbbító rendszer, amely több cél közül egy meghatározott célba továbbítandó, célállomáskóddal ellátott, üzeneteket fogadó eszközzel, valamint az üzeneteket fogadó, és valamilyen sorrendben, például érkezési sorrendben sorba állító vezérlősorral, és több, az üzeneteket a célállomásokba juttató feldolgozósorral, és a vezérlősorból célállomáskóddal ellátott üzeneteket a feldolgozósorba átvivő sorvezérlővel rendelkezik. A találmány értelmében az üzenettovábbító rendszer a célállomások számánál kevesebb számú feldolgozósorral, a vezérlősorból egy célállomáskóddal ellátott üzeneteket más célállomáskódhoz tartozó üzenetektől mentes feldolgozósorba átvivő, és a feldolgozósor kiürüléséig a feldolgozósort az adott célállomáskóddal ellátott üzenetekhez hozzárendelő sorvezérlővel van ellátva.

Tehát ennél a megoldásnál az összes célállomás felé tartó összes üzenet egy előre meghatározott sorrendben, például érkezési sorrendben sorban áll a vezérlősorban. A sorvezérlő egyenként elosztja a vezérlősorban álló üzeneteket több különböző feldolgozósor között, ahol a feldolgozósorok száma kevesebb, mint a lehetséges célállomások száma. Ennek során az egymást követő, első üzenetek olyan feldolgozósorba kerülnek, amelynek a célállomása megegyezik a feldolgozósorba elsőként kerülő üzenet célállomásával. Ha nincsen ilyen feldolgozósor, akkor a sorvezérlő a sorban első üzenetet egy üres feldolgozósorba továbbítja, ha van ilyen üres feldolgozósor. Ha nincsen üres feldolgozósor, akkor a sorvezérlő visszatartja az üzenetet a vezérlősorban, amíg feldolgozósorok valamelyike kiürül, a benne lévő üzenetek továbbítása következtében. Ezt követően a sorvezérlő ezt a feldolgozósort átirányítja az új célállomáshoz történő továbbításra, és a következő üzenetet az üres feldolgozósorba helyezi.

A találmány lehetővé teszi, hogy korlátozott számú sorokat használjunk, és ugyanakkor párhuzamosan szolgáljunk ki nagyszámú felhasználótípust vagy célállomást. Gyakorlatilag nincsen korlátja a különböző felhasználótípusok vagy célállomások számának. Az ugyanolyan típusú felhasználók vagy az ugyanazon célállomásra tartó üzenetek ugyanabban a feldolgozósorban, várnak. A felhasználók vagy az üzenetek egy sorban rendszerint időrendi sorrendben várakoznak, bár lehetségesek más rendezési szempontok is. Könnyen megoldható a különböző felhasználótípusok vagy üzenet-célállomások párhuzamos feldolgozása is. A rendszer dinamikusan rendeli hozzá a sorokat az egyes felhasználótípusokhoz vagy célállomásokhoz, és megszünteti a hozzárendelést, amikor kiürülnek és új típusú felhasználók, vagy új célállomásra tartó üzenetek érkeznek be. A találmány szerinti rendszer az adott típusú felhasználó vagy adott célállomás feldolgozása szempontjából releváns tulajdonságokat hozzárendeli a megfelelő feldolgozósorhoz a feldolgozás megkönnyítése érdekében.

A különböző típusú felhasználók vagy a különböző célok felé tartó üzenetek feldolgozása hatékonyabb, mint az egysoros stratégia alkalmazásánál, sőt a korábbi többsoros stratégiánál is. A találmány nagy rugalmasságot biztosít a feldolgozáshoz.

A találmány szerinti eljárás és berendezés további előnyeit és működését a mellékelt ábrákon bemutatott, példaképpeni kiviteli alakok segítségével ismertetjük, ahol az

1. ábra a találmány szerinti sorban állást vezérlő rendszer egy kiviteli alakjának a blokkvázlata, a
2. ábra az 1. ábra szerinti sorban állást vezérlő rendszer működésének folyamatábrája, a
3. ábra az 1. ábra szerinti sorban állást vezérlő rendszer egy másik lehetséges működési módjának folyamatábrája, a
4. ábra az 1. ábra szerinti sorban állást vezérlő rendszer kiszolgálóinak működését szemléltető folyamatábra, amint a feldolgozósorokat szolgálják ki, az
5. ábra az 1. ábra szerinti sorban állást vezérlő rendszer kiszolgálóinak egy másik lehetséges működési módját szemléltető folyamatábra, amint a feldolgozósorokat szolgálják ki, a

6. ábra a találmány szerinti rendszert magában foglaló telefonrendszer blokkvázlata, a
7. ábra a 6. ábra szerinti telefonrendszer által adott környezetben működő, találmány szerinti sorban állást vezérlő rendszer blokkvázlata, a
8. ábra a 6. és 7. ábra szerinti, sorban állást vezérlő rendszer működésének lépéseit bemutató folyamatábra, a
9. ábra a 6. és 7. ábra szerinti, sorban állást vezérlő rendszer működésének további lépéseit bemutató folyamatábra, a
10. ábra a 6. és 7. ábra szerinti, sorban állást vezérlő rendszer működésének további lépéseit bemutató folyamatábra, a
11. ábra a 6. és 7. ábra szerinti, sorban állást vezérlő rendszer egy másfajta működését bemutató további folyamatábra.

Az 1. ábrán látható a QMS sorban állást vezérlő rendszer blokkvázlata. Ez a következő részekből áll:

A QM sorvezérlő több különböző sort vezérel, mégpedig a QC vezérlősort és több QP1, QP2, ..., QP(n-1), Qpn feldolgozósort. A QC vezérlősorban sorban állnak a CL1–CLm felhasználók meghatározott sorrendben. A CL1–CLm felhasználók különböző TC1...TCt típusúak lehetnek, és különböző típusú vagy osztályú feldolgozást igényelhetnek ennek megfelelően. A QP1–QPn feldolgozósorok n száma kisebb, mint a CL1–CLm felhasználók TC1...TCt típusainak t száma. Például a QP1–QPn feldolgozósorok n száma 50 lehet, míg a CL1–CLm felhasználók TC1...TCt típusainak t száma lehet például 1000.

A QC vezérlősor minden TC1...TCt típusú CL1–CLm felhasználót fogad, és érkezési sorrendben állítja azokat sorba. A QM sorvezérlő ugyanilyen sorrendben veszi ki a CL1–CLm felhasználókat a QC vezérlősorból. Ennek megfelelően a QC vezérlősor egy úgynevezett FIFO (first-in-first-out) sor. Azonban ez csak egy lehetséges példa, és a QC vezérlősor a CL1–CLm felhasználókat más kívánt sorban is elrendezheti.

A QM sorvezérlő a QC vezérlősorban legelöl elhelyezkedő felhasználót abba a QP1–QPn feldolgozósorba helyezi, amely a felhasználónak megfelelő típushoz van rendelve az adott időpontban. A QM sorvezérlő a hozzárendelést annak a felhasználónak a típusa szerint végzi el, amely felhasználót először helyezett az üres QP1–QPn feldolgozósorba. Ez a folyamat a következőképpen megy végbe:

Tegyük fel, hogy kezdeti állapotban az összes QP1–QPn feldolgozósor üres, és a QC vezérlősorban álló legelső felhasználó TC10 típusú. Ekkor a QM sorvezérlő a felhasználót a QP1 feldolgozósorba teszi át. Ezzel a lépéssel a QP1 feldolgozósor hozzárendelődik a TC10 típushoz. Ezután a QM sorvezérlő kizárólag TC10 típusú felhasználókat helyez át a QP1 feldolgozósorba, addig, amíg az egyik SE kiszolgáló nem üríti ki a QP1 feldolgozósort. Ha a következő, a QC vezérlősorban legelöl álló felhasználó TC143 típusú, akkor a QM sorvezérlő a TC143 típusú felhasználót a QP2 feldolgozósorba helyezi, és ezzel a QP2 feldolgozósort hozzárendeli a TC143 típushoz.

Ha a QC vezérlősorban soron következő első felhasználó típusa TC10, akkor a QM sorvezérlő ezt a felhasználót a másik TC10 típusú felhasználóhoz teszi a QP1 feldolgozósorba, ahelyett, hogy egy másik üres feldolgozósorba tenné. Ezután a QM sorvezérlő a legelső felhasználókat mindig abba a feldolgozósorba helyezi, amelyik feldolgozósort az adott típushoz hozzárendelt, és ha nincsen ilyen hozzárendelt feldolgozósor, akkor a QM sorvezérlő a soron következő felhasználót egy üres feldolgozósorba teszi. Valahányszor egy üres feldolgozósorba egy felhasználótípus kerül, akkor a QM sorvezérlő azt a feldolgozósort ahhoz a felhasználótípushoz rendeli hozzá.

Ha a QM sorvezérlő az összes feldolgozósort kiosztotta vagy hozzárendelte, és a következő kiszolgálandó felhasználó olyan típusú, amelynek a típusához nincsen feldolgozósor hozzárendelve, akkor a QM sorvezérlő felfüggeszti a felhasználók kiosztását a QC vezérlősorból, és megvárja, hogy valamelyik SE kiszolgáló kiürítse valamelyik QP1–QPn feldolgozósort. Az SE kiszolgálók egyenként szolgálják ki a QP1–QPn feldolgozósorokat, és a felhasználókat egyenként vagy csoportosan veszik ki a feldolgozósorokból. A kiszolgálás eredményeképpen a feldolgozósor kiürül, és lehetővé teszi a QM sorvezérlő számára, hogy megszüntesse a feldolgozósor hozzárendelését, és a kiürült feldolgozósort újra hozzárendelje ahhoz a felhasználótípushoz, amelyik a QC vezérlősorban legelöl helyezkedik el.

Egy lehetséges másik kiviteli alaknál, amikor a felhasználókat meghatározott maximális méretű csoportonként (batch) dolgozzák fel, ha a QM sorvezérlő az összes feldolgozósort kiosztotta, és a következő kiszolgálandó felhasználó olyan típusú, amelyhez nincsen hozzárendelve feldolgozósor, akkor a QM sorvezérlő megvizsgálja a QC vezérlősort, és a QC vezérlősorban hátrébb álló olyan felhasználót, amelynek a típusa egyezik valamelyik feldolgozósor típusával, áthelyez abba a feldolgozósorba, feltéve, hogy ez az áthelyezés nem hoz létre egy újabb feldolgozandó csoportot.

A QM sorvezérlő minden feldolgozósorhoz hozzárendeli vagy tárolja a vonatkozó sorinformációkat. Ez a sorinformáció magában foglalja a pillanatnyi hozzárendelt felhasználótípust, és egyéb mezőket is tartalmazhat.

A találmány azon a felismerésen alapul, hogy bár a TC1...TCt felhasználótípusok száma nagyon nagy is lehet, egyidőben rendszerint csak korlátozott számú felhasználó vár feldolgozásra. Ezért az ebbe a részhalmazba eső, feldolgozásra váró felhasználótípusok számának becslésével lehetőség van arra, hogy elegendő QP1–QPn feldolgozósort biztosítsunk a felhasználók átmeneti tárolására.

A QM sorvezérlő nem rendel hozzá állandó jelleggel egy rögzített felhasználótípust egy feldolgozósorhoz, hanem egyszerre csak egy típushoz rendel egy feldolgozósort egy adott időben. Egy QP1...QPn feldolgozósor akkor van igénybe véve, ha a QM sorvezérlő hozzárendeli azt egy felhasználótípushoz. Egyébként a feldolgozósor üres és felhasználható. A feldolgozósor csak akkor használható fel hozzárendeléshez vagy átirányításhoz, ha egy felhasználó sincs abban a feldolgozósorban.

A 2. ábrán az 1. ábra szerinti QMS sorban állást vezérlő rendszer működését szemléltető folyamatábra látható. A 100 logikai lépésben a QM sorvezérlő vár, hogy lehetővé tegye a felhasználók elhelyezését a QC vezérlősorban, egy vagy több forrásból. A folyamat egy várólépéssel indul. Ez rossz hatékonyságúnak tűnhet, de valójában nem az. Mivel a folyamatok ciklikusak, a kezdeti várás után a folyamat úgy folytatódik, mintha a várakozás a folyamat végén lenne. A legtöbb esetben a rendszer elindításakor még nincsenek jelen felhasználók, és ezért ha a folyamat nem várással indulna, akkor üresen lefutna az első várakozásig, anélkül, hogy bármit is végrehajtott volna. Ezért a várással való kezdés általában hatékonyabb. De létezhetnek más megvalósítási módok is. A 104. logikai lépésben a QM sorvezérlő megvizsgálja, hogy a QC vezérlősor üres-e. Ha igen, akkor a QM sorvezérlő visszatér a várakozáshoz a 100 logikai lépésbe. Ha a QC vezérlősor nem üres, hanem felhasználókat tartalmaz, akkor a QM sorvezérlő továbblép a 107 logikai lépésre, és megállapítja az első felhasználótípusát. A 110 logikai lépésben a QM sorvezérlő megvizsgálja, hogy az első felhasználótípusa megfelel-e valamelyik feldolgozósor felhasználótípusának. Más szavakkal, a QM sorvezérlő a tárolt adatai alapján megállapítja, hogy a 107 logikai lépésben talált felhasználótípushoz rendelt-e már hozzá feldolgozósort. Bizonyos értelemben a QM sorvezérlő meggyőződik arról, hogy a QP1–QPn feldolgozósorok valamelyike tartalmaz-e olyan típusú felhasználót, mint amilyen a 107 logikai lépésben került elő.

Ha a 107 logikai lépésben megvizsgált felhasználótípus egyezik valamelyik QP1–QPn feldolgozósorhoz hozzárendelt felhasználótípussal, akkor a QM sorvezérlő továbblép a 114 logikai lépésre, és elveszi a legelső vagy legfelső felhasználót a QC vezérlősorból, és átteszi abba a feldolgozósorba, amelyik ahhoz a felhasználótípushoz van hozzárendelve. Ha a 107 logikai lépésben megvizsgált legelső felhasználótípusa nem egyezik meg a 110 lépésben megvizsgált egy feldolgozósor típusával sem, akkor a QM sorvezérlő továbblép a 117 logikai lépésre, és megkérdezi, hogy létezik-e felhasználható feldolgozósor, nevezetesen egy olyan QP1–QPn feldolgozósor, amelyik üres. Ha van felhasználható, vagyis üres QP1–QPn feldolgozósor, akkor a QM sorvezérlő a 120 logikai lépésben a felhasználható új feldolgozósort ahhoz a felhasználótípushoz rendeli, és a 107 logikai lépésben kiemeli az első felhasználót, és átteszi a feldolgozósorba. Ezek után a QM sorvezérlő visszatér a 104 logikai lépéshez, és megkérdezi, hogy a QC vezérlősor üres-e.

Ha a QP1–QPn feldolgozósorok egyike sem üres, akkor elképzelhető, hogy a feldolgozósorok gyakorlatilag bedugultak. Ekkor a 124 logikai lépésben a QM sorvezérlő meggyőződik arról, hogy az SE kiszolgálók ténylegesen kiszolgálnak-e legalább egy feldolgozósort, vagy ezzel szemben valami eldugítja vagy lefogja az összes feldolgozósort. Ha a felhasználók kiszolgálása nincsen teljesen meggátolva, vagyis legalább egy feldolgozósor feldolgozása sikeresen folyik, akkor a QM sorvezérlő visszatér a felhasználóhoz, és a 100 logikai lépésben várakoztatja, miközben felfüggeszti a teljes folyamat végrehajtását abból a célból, hogy elég időt biztosítson a feldolgozósor kiszolgálására és kiürítésére, azért, hogy felhasználható legyen.

Ha az SE kiszolgálók nem szolgálják ki a felhasználókat eredményesen egy feldolgozósorban sem, vagyis valami teljesen meggátolja vagy akadályozza feldolgozósorok kiszolgálását, akkor a QM sorvezérlő továbblép a 127 logikai lépésre. Ebben a lépésben a QM sorvezérlő kitisztítja az összes QP1–QPn feldolgozósort úgy, hogy kiüríti a feldolgozósorokat, és az ott levő felhasználókat visszaküldi az eredeti forrásukhoz. Ettől a ponttól fogva ezek a felhasználók már nem vesznek részt a folyamatban, bár a források visszaküldhetik a felhasználókat a QC vezérlősorba. Egyéb műveletek is elképzelhetők a torlódást okozó probléma megszüntetésére. Amikor a QM sorvezérlő megtisztította a QP1–QPn feldolgozósorokat, akkor ezeket a feldolgozósorokat hozzárendelheti a legfelső felhasználótípusához, és a legfelső felhasználót átteheti a QC vezérlősorból az üres feldolgozósorba.

A találmány egyik megvalósításánál a QM sorvezérlő a 127 logikai lépésben csak egy vagy több QP1–QPn feldolgozósort tisztít meg. Egy másik lehetséges változatnál a 127 logikai lépésben a QM sorvezérlő a QC vezérlősort is megvizsgálja, és visszaküldi az összes olyan felhasználót, amelyik típusához hozzá volt rendelve valamelyik eldugult feldolgozósor. Egy további, alternatív megvalósításnál a 127 logikai lépésben a felhasználóknak a forrásaikhoz történő visszajuttatása helyett a QM sorvezérlő a felhasználókat a QC vezérlősorba juttatja vissza. Ezeknél a megvalósítási módoknál két kérdésre kell ügyelni ahhoz, hogy ezek a módozatok megfelelően működjenek. Először is, az összes olyan tétel egy adott felhasználótípusból, amely a vezérlősorban van (vagyis még nincsen a feldolgozósorban), és amely típus már hozzá van rendelve egy feldolgozósorhoz, azok után a tételek után kerüljön, amelyek visszajutottak a feldolgozósorból a vezérlősorba. Erre azért van szükség, hogy megmaradjon az adott típusú felhasználók közötti időrendi sorrend. Másodszor, a valamilyen állandó jellegű probléma esetén esetlegesen fellépő végtelen ciklusok elkerülése érdekében, a feldolgozósorból visszakerülő felhasználókhoz visszatérés-számlálót kell rendelni. Meg kell adni a visszatérés-számláló maximális értékét, és ha a felhasználóhoz rendelt visszatérés-számláló elérte ezt a maximális értéket, akkor a felhasználót a forrásához kell visszaküldeni a vezérlősor helyett.

A találmány 2. ábrán szemléltetett megvalósításának egy további módosított változatát a 3. ábra folyamatábrája szemlélteti. Ebben az esetben, amennyiben a válasz a 124 logikai lépésben nemleges, vagyis nincsen eldugulva az összes feldolgozósor, akkor a QM sorvezérlő a 130 logikai lépésben várakozik. Ez a várakozás összemérhető a 100 logikai lépésben végzett várakozással. Ezt követően a QM sorvezérlő közvetlenül a 117 logikai lépésre tér vissza. A 3. ábrán látható művelet hatékonyabb, mint a 2. ábrán látható visszatérés a 100 logikai lépésre, feltéve, hogy QC vezérlősor szigorúan FIFO-

sor. Ebben az esetben semmi nem helyettesítheti a QC vezérlősor legelején álló felhasználót, amelynek a típusa nem fog egyezni egy QP1–QPn feldolgozósorhoz hozzárendelt típussal sem. Ekkor ugyanis a feldolgozás mindig a 104, 107, 110 logikai lépéseken keresztül mindig a 117 logikai lépéshez fog eljutni. A 3. ábrán látható kiviteli alaknál a QM sorvezérlő közvetlenül a 117 logikai lépésre lép. A találmány egy további megvalósításánál a felhasználó forrását az SE kiszolgálók értesítik, ha nem sikeres az adott felhasználó feldolgozása.

A találmány egy további kiviteli alakjánál a QM sorvezérlő más sorrendet állít fel a QC vezérlősorban vagy a QP1–QPn feldolgozósorokban, azaz az időrenditől különböző sorrendet. Egy lehetséges változatnál a sorrendet a prioritás szabja meg. Egy másik változatnál a prioritás és az időrendiség kombinációja használható. Bármilyen skaláris mező használható prioritásként. Eszerint a QM sorvezérlő létrehoz egy sorrendet, amely csökkenő vagy növekvő is lehet, a skaláris mező értékei szerint.

Általános esetben a QP1–QPn feldolgozósorok több, különböző prioritású felhasználót tartalmazhatnak egy adott időben. Ez a helyzet állhat fenn a QC vezérlősorban is, mivel az összes feldolgozósor összes felhasználója először a QC vezérlősoron halad át. Egy lehetséges megvalósításnál a QM sorvezérlő véletlenszerűen választ, vagy csatolt listát alkalmaz. Egy másik változatnál FIFO-sort alkalmaz egy kiegészítő változóval összekapcsolva, amelyik megmondja, hogy az adott időpontban melyik a legmagasabb prioritási érték. Ezek a megvalósítások kompromisszumot jelentenek a felállási idő (setup time) és a következő felhasználó megtalálásának hatékonysága között.

Az SE kiszolgálók a QP1–QPn feldolgozósorokat a 4. és 5. ábrán bemutatott folyamatábráknak megfelelő eljárással szolgálják ki.

A 4. ábra a feldolgozás megkezdését mutatja be, és az 5. ábra a tényleges feldolgozást mutatja. A feldolgozás megkezdése és maga a feldolgozás elválik egymástól, és így az SE kiszolgálók több sort tudnak párhuzamosan feldolgozni. A lépések között van egy „feldolgozás megkezdődött" ellenőrző lépés, ami biztosítja, hogy ugyanazt a sort ne dolgozzák fel többször.

A QM sorvezérlő működésének időzítésétől függetlenül az SE kiszolgálók meghatározott sorrendben kezdik meg a QP1–QPn feldolgozósorok vizsgálatát, és először várnak a 200 logikai lépésben. A folyamat egy várólépéssel indul. Ez rossz hatékonyságúnak tűnhet, de valójában nem az. Mivel a folyamatok ciklikusak, a kezdeti várás után a folyamat úgy folytatódik, mintha a várakozás a folyamat végén lenne. A legtöbb esetben a rendszer elindításakor még nincsenek jelen felhasználók, és ezért ha a folyamat nem várással indulna, akkor üresen lefutna az első várakozásig, anélkül, hogy bármit is végrehajtott volna. Ezért a várással való kezdés általában hatékonyabb. De létezhetnek más megvalósítási módok is. A 204 logikai lépésben az SE kiszolgálók ellenőrzik, hogy a következő felhasználói sor készen áll-e a feldolgozáshoz, és a 207 logikai lépésben megkérdezi, hogy a feldolgozósorban van-e felhasználó. Nemleges válasz esetén az SE kiszolgáló a 210 logikai lépésre megy tovább, és kitakarítja vagy eltávolítja az összes olyan maradék sorinformációt, amely a sorban korábban levő felhasználókra vonatkozott, és megszünteti a feldolgozósor hozzárendeltségét. Ezt követően a kiszolgálók továbblépnek a 214 logikai lépésben a következő sor ellenőrzésére, és visszatérnek a 200 logikai lépésre.

Ha a 207 logikai lépésben a válasz igen, vagyis a kérdéses feldolgozósor tartalmaz felhasználót, akkor az SE kiszolgáló a 220 logikai lépésre lép, hogy megállapítsa, az illető feldolgozósor feldolgozása megkezdődött-e. Ha a válasz nemleges, akkor az SE kiszolgáló továbblép a 224 logikai lépésre, abból a célból, hogy megjelölje a vizsgált feldolgozósort, mint feldolgozásra alkalmast, és hogy megkezdje a feldolgozást. Az SE kiszolgáló a 227 logikai lépésben megállapítja, hogy a feldolgozás sikeresen megkezdődött-e. Ha nem, akkor az SE kiszolgáló visszatér a 200 logikai lépésre. Ha igen, akkor a 234 logikai lépésben az SE kiszolgáló megjelöli a sort mint feldolgozás alatt levőt, és a 214 logikai lépésre lép, mielőtt a 200 lépésre visszatérne.

A feldolgozás megkezdése után a tényleges feldolgozási lépéseket az 5. ábra szemlélteti. Itt a 260 logikai lépésben végrehajtott várakozás után az SE kiszolgálók a 264 lépésben végrehajtják a tényleges feldolgozást. A 267 lépésben megvizsgálják, hogy a feldolgozás sikeres volt-e. Ha igen, akkor az SE kiszolgáló eltávolítja a felhasználót a sorból a 270 logikai lépésben. A 274 logikai lépésben az SE kiszolgálók megállapítják, hogy a feldolgozás alatt lévő sorban vannak-e még felhasználók. Ha a válasz nemleges, akkor a kiszolgálók a 287 logikai lépésre lépnek, ahol a kiszolgálók megjelölik a sort, mint nem feldolgozás alatt levőt, és ezzel véget ér a vonatkozó sor feldolgozása. Ha a válasz igen, akkor a kiszolgálók a 277 logikai lépésben megvizsgálják, hogy feldolgozásra került-e a megengedett legnagyobb számú felhasználó. Egyes kiszolgálók a felhasználókat csoportosan kezelik, amely csoportnak adott esetben maximálva lehet a mérete. Ha a kiszolgáló a felhasználókat csoportosan kezeli, és a kiszolgáló elérte a maximális csoportlétszámot, akkor a kiszolgáló a következő ciklusig felfüggeszti a feldolgozást. Ennek megfelelően, ha a 277 logikai lépésben a válasz igen, akkor a kiszolgáló elérte a maximális csoportlétszámot, és feldolgozta a csoportot. Ha nemleges a válasz, akkor a kiszolgáló a 280 logikai lépésben visszatér a következő felhasználóért, majd visszatér a 264 logikai lépésre. Ha a 267 logikai lépésben a válasz nemleges, akkor az SE kiszolgáló a 284 logikai lépésre vált hibakezelést végezni. A találmány egy lehetséges változata szerint a hibakezelés során a feldolgozósort megjelöljük mint újból feldolgozandót, és megadjuk az újbóli feldolgozásra megszabott időt. A sor feldolgozását vezérlő egység, amikor a 4. ábra szerint megkezdi a feldolgozást, akkor átugorja azokat a sorokat, amelyek újra feldolgozásra vannak megjelölve a megadott újrafeldolgozási idő alatt. A találmány egy megvalósításánál a hibakezelést nem a kiszolgálók végzik.

A 6. ábrán látható blokkvázlat a találmány egy olyan kiviteli alakját szemlélteti, amely egy telefon-

rendszer által adott környezetben működik. Ebben az esetben a 2. ábra szerinti CL1–CLm felhasználók hangpostaüzenetek, amelyeknek a típusa a célállomásukkal együtt változik. A 6. ábrán látható telefonrendszernél az L1 telephelyen kialakított BE1 alállomás köti össze egy állami vagy magán TN telefonhálózat T1, T2, ..., Tk telefonjait, valamint a VM1 hangpostarendszert. A TN telefonhálózat köti össze a BE1 alállomást a többi BE2, ..., BEq alállomással, amelyek LC1, ..., LC h telephelyeken vannak, adott esetben távolabb LC1 telephelytől. Minden egyes BE2–BEq alállomás a TN telefonhálózat több TL telefonjával és a hozzá tartozó VM2...VMq hangpostarendszerrel van összekötve. Az egyes VM1...VMq hangpostarendszerek önmagukban ismertek, amelyek rögzítik a hozzájuk tartozó BE1...BEq alállomások T1–Tk és TL telefonjairól érkező üzeneteket.

A 6. ábra LC1...LCh telephelyei egymástól szét lehetnek szórva térben. Például a BE1 alállomás L1 telephelye és a VM1 hangpostarendszer lehet a California állambeli Santa Claraban, míg a többi telephely lehet a New York állambeli New Yorkban vagy a Massachusetts állambeli Bostonban, vagy például az angliai Nottinghamben.

A VM1 hangpostarendszer tartalmaz egy QMT sorban állást vezérlő rendszert, amelynek részletei a 7. ábra blokkvázlatáról ismerhetők meg. A VM1 hangpostarendszer fogadja és tárolja a BE1 alállomásról érkező üzeneteket, amelyeket az adott L1 telephelyre befutó T1...Tk telefonok vagy a távoli LC1...LCh telephelyekre befutó TL telefonok továbbítanak. Ezek olyan üzenetek, amelyeket a T1...Tk telefonok vesznek, vagy amelyeket a távoli TL telefonokhoz kell továbbítani. Minden egyes üzenet el van látva egy célállomáskóddal, amely egy adott vagy adott számú, több adott célállomásra vonatkozik. A célállomás rendszerint egy másik hangpostarendszer, például a VM2...VMq hangpostarendszerek valamelyike. A QMT sorban állást vezérlő rendszer akkor a leghatékonyabb, ha összegyűjt több, egy célállomásra, például egy másik hangpostarendszerbe szánt üzenetet egy csoportban (batch). Az egyes üzenet kódolása tartalmazza a célhangpostarendszer számát, és az adott cím postaládájának a számát a célként megjelölt hangpostarendszerben. A célállomások száma például akár 1000 is lehet.

A VM1 hangpostarendszerben egy nem ábrázolt kiválasztóegység elválasztja a távoli TL telefonokhoz továbbítandó rögzített üzeneteket, vagyis azokat, amelyek a BE1 alállomás telephelyén kívül eső LC2...LCh telephelyekre befutó telefonokhoz mennek, azoktól az üzenetektől, amelyek a helyi T1...Tk telefonokhoz mennek. A VM1 hangpostarendszer lehetővé teszi, hogy a helyi T1...Tk telefonok hozzáférjenek a nekik szánt üzenetekhez. A QMT sorban állást vezérlő rendszer felhasználóként csak azokat az ME1...MEm üzeneteket fogadja, amelyek a távoli TL telefonokhoz mennek, TC1...TCt célállomásokra, a BE1 alállomás LC1 telephelyén kívüli LC2...LCh telephelyekre.

A 7. ábra a QMR sorban állást vezérlő rendszer részleteit mutatja be. Ez utóbbi az 1. ábra szerinti QMS sorban állást vezérlő rendszert alkotja egy telefonrendszer által adott környezetben, különösen a 6. ábra szerinti hangpostarendszerben. A 7. ábra gyakorlatilag megegyezik az 1. ábrával, azzal a különbséggel, hogy a CL1...CLm felhasználók ebben az esetben ME1...MEm üzenetek, és a TC1...TCt típusok pedig a DE1...DEt célállomásokként jelennek meg. Az SE kiszolgálók ennél a kiviteli alaknál a sorfeldolgozást vezérlő személyi számítógép formájában jelennek meg. Ez a számítógép tárcsázza a távoli célállomást és továbbítja az üzeneteket. Az 1. ábra hivatkozási jelei a 7. ábrán az egyező részeket jelölik.

A 7. ábrán látható kiviteli alaknál a QC vezérlősor fogadja az összes DE1...DEt célállomásra szánt ME1...MEm üzeneteket, és érkezési sorrendben tartja azokat. A QM sorvezérlő ugyanilyen sorrendben távolítja el az üzeneteket a QC vezérlősorból. Ennek megfelelően a QC vezérlősor egy FIFO-sor. Ez azonban csak egy példaképpeni alak, és a QC vezérlősor más kívánt sorrendbe is rakhatja az üzeneteket.

A 7. ábrán a QM sorvezérlő a QC vezérlősorban legfelül álló üzenetet az egyik, az üzenet célállomásához rendelt QP1–QPn feldolgozósorba teszi. A QM sorvezérlő a hozzárendelést annak az üzenetnek a célállomása szerint végzi el, amely üzenetet először helyezett az üres QP1–QPn feldolgozósorba. Ez a folyamat a következőképpen megy végbe:

Tegyük fel, hogy kezdeti állapotban az összes QP1–QPn feldolgozósor üres, és a QC vezérlősorban álló legelső üzenet célállomása DE10. Ekkor a QM sorvezérlő az üzenetet a QP1 feldolgozósorba teszi át. Ezzel a lépéssel a QP1 feldolgozósor hozzárendelődik a DE10 célállomáshoz. Ezután a QM sorvezérlő kizárólag DE10 célállomásra menő üzeneteket helyez át a QP1 feldolgozósorba, addig, amíg a sorfeldolgozást vezérlő számítógép nem üríti ki a QP1 feldolgozósort. Ha a következő, a QC vezérlősorban legelöl álló üzenet DE143 célállomásra megy, akkor a QM sorvezérlő a DE143 célállomásra menő célállomáskódú üzenetet a QP2 feldolgozósorba helyezi, és ezzel a QP2 feldolgozósort hozzárendeli a DE143 célállomáskódhoz.

Ha a QC vezérlősorban soron következő első üzenet célállomáskódja DE10, akkor a QM sorvezérlő ezt a üzenetet a másik DE10 célállomáskódú üzenethez teszi a QP1 feldolgozósorba, ahelyett, hogy egy másik üres feldolgozósorba tenné. Ezután a QM sorvezérlő a legelső üzeneteket mindig abba a feldolgozósorba helyezi, amelyik feldolgozósort az adott célállomáskódhoz hozzárendelte, és ha nincsen ilyen hozzárendelt feldolgozósor, akkor a QM sorvezérlő a soron következő üzenetet egy üres feldolgozósorba teszi. Valahányszor egy üres feldolgozósorba egy üzenet-célállomáskód kerül, akkor a QM sorvezérlő azt a feldolgozósort ahhoz az üzenetnek a célállomáskódjához rendeli hozzá.

A sorfeldolgozást vezérlő számítógép a QM sorvezérlő működésétől független időzítéssel veszi sorba a feldolgozósorokat. Minden egyes feldolgozósornál a sorfeldolgozást vezérlő számítógép tárcsázza az ahhoz a sorhoz rendelt célállomáskódot. Ha a hívás megtörtént, a sorfeldolgozást vezérlő számítógép továbbítja a sor-

feldolgozó rendszerben összeállt üzenetcsoportot az alállomáson keresztül. A csoportban az üzenetek maximális száma attól függ, hogy egy adott időpontban a rendszer egyszerre hány üzenetet továbbíthat, ez utóbbi viszont az üzenetküldő és fogadó protokoll függvénye.

Ha a feldolgozósorban az üzenetek száma kevesebb, mint a maximális szám egy ilyen csoportban, akkor egy ilyen tárcsázás kiüríti a feldolgozósort. Ekkor a QM sorvezérlő megszüntetheti a feldolgozósor hozzárendelését, és újból hozzárendelheti a vezérlősorban legfelül álló üzenet célállomáskódjához. Azonban ha a feldolgozósorban álló üzenetek száma meghaladja a maximális csoportlétszámot, akkor a tárcsázás felfüggeszti a feldolgozást a sorfeldolgozást vezérlő számítógép következő ciklusáig. Ez utóbbi mindig azzal folyatja a műveletet, hogy ellenőrzi a következő sorban az üzenetek elküldésének sikerességét.

Amikor a QM sorvezérlő kiosztotta vagy hozzárendelte az összes feldolgozósort, és a vezérlősorban következő üzenet olyan célállomásra megy, amelyhez nincsen feldolgozósor rendelve vagy kiosztva, akkor a QM sorvezérlő felfüggeszti az üzenetek kiosztását a vezérlősorból, és megvárja a sorfeldolgozást vezérlő számítógép és a BE1 alállomás működésének eredményeképpen valamelyik QP1–QPn feldolgozósor kiürülését, és az abban levő üzenetek elküldését.

A QM sorvezérlő és a sorfeldolgozást vezérlő számítógép minden feldolgozósorhoz hozzárendeli vagy tárolja a vonatkozó sorinformációkat. Ez a sorinformáció magában foglalja a pillanatnyi hozzárendelt célállomáskódot, és egyéb mezőket is tartalmazhat, például arra vonatkozót, hogy a feldolgozósor üres, éppen feldolgozás alatt van, az újrapróbálkozások száma, a feldolgozósor ismételt feldolgozása, a következő újrafeldolgozás ideje stb.

A találmány azon a felismerésen alapul, hogy bár a DE1...DEt üzenet-célállomáskódok száma nagyon nagy is lehet, például akár 1000 vagy több, de egy időben rendszerint csak korlátozott számú üzenet vár feldolgozásra, és ezt korlátozott erőforrással kell feldolgozni. Ezért az ebbe a részhalmazba eső, feldolgozásra váró üzenetek számának becslésével lehetőség van arra, hogy elegendő QP1–QPn feldolgozósort biztosítsunk a üzenetek átmeneti tárolására, például ötvenet.

A QM sorvezérlő nem rendel hozzá állandó jelleggel egy rögzített célállomáskódot egy feldolgozósorhoz, hanem egyszerre csak egy célállomáskódhoz rendel egy feldolgozósort egy adott időben, amíg a feldolgozósor kiürül. Ezután a QM sorvezérlő újra hozzárendeli az üres feldolgozósort. Egy QP1–QPn feldolgozósor akkor van igénybe véve, ha a QM sorvezérlő hozzárendeli azt egy üzenet-célállomáskódhoz. Egyébként a feldolgozósor üres és felhasználható. A feldolgozósor csak akkor használható fel hozzárendeléshez vagy átirányításhoz, ha egy üzenet sincs abban a feldolgozósorban.

A 8. ábrán az 6. és 7. ábra szerinti QMT sorban állást vezérlő rendszer működését szemléltető folyamatábra látható. A működés hasonló a 2. ábrán szemléltetett működéshez, de ezúttal egy telefonos és hangpostakörnyezetben. A 300 logikai lépésben a QM sorvezérlő vár, hogy lehetővé tegye az üzenetek elhelyezését a QC vezérlősorban. A folyamat egy várólépéssel indul. Ez rossz hatékonyságúnak tűnhet, de valójában nem az. Mivel a folyamatok ciklikusak, a kezdeti várás után a folyamat úgy folytatódik, mintha a várakozás a folyamat végén lenne. A legtöbb esetben a rendszer elindításakor még nincsenek jelen üzenetek, és ezért ha a folyamat nem várással indulna, akkor üresen lefutna az első várakozásig, anélkül, hogy bármit is végrehajtott volna. Ezért a várással való kezdés általában hatékonyabb. De létezhetnek más megvalósítási módok is. A 304 logikai lépésben a QM sorvezérlő megvizsgálja, hogy a QC vezérlősor üres-e. Ha igen, akkor a QM sorvezérlő visszatér a várakozáshoz a 300 logikai lépésbe. Ha a QC vezérlősor nem üres, hanem üzeneteket tartalmaz, akkor a QM sorvezérlő továbblép a 307 logikai lépésre, és megállapítja az első üzenet célállomáskódját. A 310 logikai lépésben a QM sorvezérlő megvizsgálja, hogy az első üzenet célállomáskódja megfelel-e valamelyik feldolgozósor üzenet-célállomáskódjának. Más szavakkal, a QM sorvezérlő a tárolt adatai alapján megállapítja, hogy a 107 logikai lépésben talált célállomáskódhoz rendelt-e már hozzá feldolgozósort. Bizonyos értelemben a QM sorvezérlő meggyőződik arról, hogy a QP1–QPn feldolgozósorok valamelyike tartalmaz-e olyan célállomáskódú üzenetet, mint amilyen a 107 logikai lépésben került elő.

Ha a 307 logikai lépésben megvizsgált üzenet-célállomáskód egyezik valamelyik QP1–QPn feldolgozósorhoz hozzárendelt üzenet-célállomáskóddal, akkor a QM sorvezérlő továbblép a 314 logikai lépésre, és elveszi a legelső vagy legfelső üzenetet a QC vezérlősorból, és átteszi abba a feldolgozósorba, amelyik ahhoz a célállomáskódhoz van hozzárendelve. Ha a 307 logikai lépésben megvizsgált legelső üzenet célállomáskódja nem egyezik meg a 310 lépésben megvizsgált egyik feldolgozósor célállomáskódjával sem, akkor a QM sorvezérlő továbblép a 317 logikai lépésre, és megkérdezi, hogy létezik-e felhasználható feldolgozósor, nevezetesen egy olyan QP1–QPn feldolgozósor, amelyik üres. Ha van felhasználható, vagyis üres QP1–QPn feldolgozósor, akkor a QM sorvezérlő a 320 logikai lépésben a felhasználható új feldolgozósort ahhoz a célállomáskódhoz rendeli, és a 307 logikai lépésben kiemeli az első üzenetet, és átteszi a feldolgozósorba. Ezek után a QM sorvezérlő visszatér a 304 logikai lépéshez, és megkérdezi, hogy a QC vezérlősor üres-e.

Ha a QP1–QPn feldolgozósorok egyike sem üres, és olyan sor sincsen, amelyik célállomáskódja egyezik a keresettel, akkor elképzelhető, hogy a feldolgozósorok gyakorlatilag bedugultak. Ekkor a 324 logikai lépésben a QM sorvezérlő megvizsgálja a sorfeldolgozást vezérlő számítógép eredményei alapján, hogy valami eldugítja-e vagy lefogja-e az összes feldolgozósort. Ha a üzenetek továbbítása nincsen teljesen meggátolva, vagyis legalább egy feldolgozósor feldolgozása sikeresen folyik, akkor a QM sorvezérlő az üzenetet a vezérlősorban hagyja, és a 300 logikai lépésben várakoztatja, miközben felfüggeszti a teljes folyamat végrehajtását abból a célból, hogy elég időt biztosítson a feldolgozósor

kiszolgálására és kiürítésére. Ha valami teljesen meggátolja vagy akadályozza feldolgozósorok kiszolgálását, akkor a QM sorvezérlő továbblép a 327 logikai lépésre. Ebben a lépésben a QM sorvezérlő kitisztítja az összes QP1–QPn feldolgozósort úgy, hogy kiüríti a feldolgozósorokat, és az ott levő üzeneteket visszaküldi az eredeti forrásukhoz. Egyéb műveletek is elképzelhetők a torlódást okozó probléma megszüntetésére.

Ez a kitisztítóeljárás az üzeneteket a feladókhoz juttatja vissza. Amikor a QM sorvezérlő megtisztította a QP1–QPn feldolgozósorokat, akkor ezeket a feldolgozósorokat hozzárendelheti a legfelső üzenet célállomáskódjához, és a legfelső üzenetet átteheti a QC vezérlősorból az adott üzenet célállomáskódjához rendelt feldolgozósorba. A találmány egyik másik megvalósításánál a QM sorvezérlő az összes QP1–QPn feldolgozósort megtisztítja.

A 8. ábrán látható esetben a QM sorvezérlő más sorrendet is felállíthat a QC vezérlősorban vagy a QP1–QPn feldolgozósorokban, azaz az időrenditől különböző sorrendet. Bármilyen skaláris mező használható prioritásként. Eszerint a QM sorvezérlő létrehoz egy sorrendet, amely csökkenő vagy növekvő is lehet, a skaláris mező értékei szerint.

Általános esetben a QP1–QPn feldolgozósorok több, különböző prioritású üzenetet tartalmazhatnak egy adott időben. Ez a helyzet állhat fenn a QC vezérlősorban is, mivel az összes feldolgozósor összes üzenete először a QC vezérlősoron halad át.

A sorfeldolgozást vezérlő számítógép a 9. és 10. ábrán bemutatott folyamatábráknak megfelelő eljárással vezérli a BE1 alállomáson az üzenetek továbbítását a megfelelő célállomásokra, a QP1–QPn feldolgozósorokból. A 9. és a 10. ábrán, hasonlóan a 4. és 5. ábrához, a sorfeldolgozást vezérlő számítógép felosztja az üzenetek feldolgozását két szétváló műveletre, és így lehetséges több sort párhuzamosan feldolgozni. A 9. ábra a továbbítás kezdeményezését szemlélteti, és a 10. ábra a 9. ábra szerint kiválasztott üzenetek továbbítását.

A 9. ábra lépései a QM sorvezérlő működésének időzítésétől függetlenek. Itt a sorfeldolgozást vezérlő számítógép meghatározott sorrendben kezdeményezi a QP1–QPn feldolgozósorok vizsgálatát, és az egyenként végzett vizsgálatok között vár a 400 logikai lépésben. A folyamat egy várólépéssel indul. Ez rossz hatékonyságúnak tűnhet, de valójában nem az. Mivel a folyamatok ciklikusak, a kezdeti várás után a folyamat úgy folytatódik, mintha a várakozás a folyamat végén lenne. A legtöbb esetben a rendszer elindításakor még nincsenek jelen üzenetek, és ezért ha a folyamat nem várással indulna, akkor üresen lefutna az első várakozásig, anélkül, hogy bármit is végrehajtott volna. Ezért a várással való kezdés általában hatékonyabb. De létezhetnek más megvalósítási módok is. A 404 logikai lépésben a sorfeldolgozást vezérlő számítógép ellenőrzi, hogy a következő üzenetsor készen áll-e a feldolgozáshoz, és a 407 logikai lépésben megkérdezi, hogy a feldolgozósorban van-e üzenet. Nemleges válasz esetén a sorfeldolgozást vezérlő számítógép a 410 logikai lépésre megy tovább, és kitakarítja vagy eltávolítja az összes olyan maradék sorinformációt, amely a sorban korábban levő üzenetekre vonatkozott, és megszünteti a feldolgozósor hozzárendeltségét. Ezt követően a sorfeldolgozást vezérlő számítógép továbblép a 414 logikai lépésben a következő sor ellenőrzésére, és visszatérnek a 400 logikai lépésre.

Ha a 407 logikai lépésben a válasz igen, vagyis a kérdéses feldolgozósor tartalmaz üzenetet, akkor a sorfeldolgozást vezérlő számítógép a 420 logikai lépésre lép, hogy megállapítsa, hogy a BE1 alállomás már továbbítja-e az üzeneteket a kérdéses feldolgozósorból. Ha igen, akkor a sorfeldolgozást vezérlő számítógép a 414 logikai lépésre lép, a következő sor ellenőrzése céljából. Egy másik eljárásnál, ahol a hibakezelésnél többszöri újrapróbálkozást is használnak, a 414 logikai lépésben át lehet ugrani az olyan feldolgozósorokat, amelyek az újrapróbálkozás közben vannak, és amelyeknél az újrapróbálkozási idő még nem járt le. Ha a válasz nemleges, akkor a sorfeldolgozást vezérlő számítógép továbblép a 224 logikai lépésre, abból a célból, hogy megjelölje a vizsgált feldolgozósort mint üzenettovábbításra alkalmast, és hogy a BE1 alállomástól egy csatornát igényeljen az üzenettovábbításhoz. A sorfeldolgozást vezérlő számítógép a 227 logikai lépésben megállapítja, hogy a csatorna megszerzése sikeresen megtörtént-e. Ha igen, akkor a sorfeldolgozást vezérlő számítógép a 434 logikai lépésben megjelöli a sort feldolgozásra, és a 414 logikai lépésen át visszatér a 404 logikai lépésre.

A tényleges feldolgozási lépéseket az 10. ábra szemlélteti. Itt a 460 logikai lépésben végrehajtott várakozás után a sorfeldolgozást vezérlő számítógép a 464 lépésben utasítja a BE1 alállomást a feldolgozósorban álló üzenetek célállomáskódjának tárcsázására, és a sorban első üzenet továbbítására az így hozzáférhetővé vált csatornán keresztül. A folyamat egy várólépéssel indul. Ez rossz hatékonyságúnak tűnhet, de valójában nem az. Mivel a folyamatok ciklikusak, a kezdeti várás után a folyamat úgy folytatódik, mintha a várakozás a folyamat végén lenne. A legtöbb esetben a rendszer elindításakor még nincsenek jelen üzenetek, és ezért ha a folyamat nem várással indulna, akkor üresen lefutna az első várakozásig, anélkül, hogy bármit is végrehajtott volna. Ezért a várással való kezdés általában hatékonyabb. De létezhetnek más megvalósítási módok is. A 467 lépésben a sorfeldolgozást vezérlő számítógép megvizsgálja, hogy a továbbítás sikeres volt-e. Ha igen, akkor a sorfeldolgozást vezérlő számítógép eltávolítja a üzenetet a sorból a 470 logikai lépésben. A 474 logikai lépésben a sorfeldolgozást vezérlő számítógép ellenőrzi, hogy a feldolgozás alatt lévő sorban vannak-e még üzenetek. Ha a válasz igen, akkor a sorfeldolgozást vezérlő számítógép a 477 logikai lépésben megvizsgálja, hogy a megengedett legnagyobb számú üzenetet továbbította-e a BE1 alállomás. Nemleges esetben a sorfeldolgozást vezérlő számítógép a 480 logikai lépésben visszatér a következő üzenetért, majd visszatér a 464 logikai lépésre, és újraindítja a folyamatot.

A 484 logikai lépés adott esetben mellőzhető is. A 484 logikai lépésre akkor kerül sor, ha a 467 logikai lépésben a BE1 alállomás nem fejezte be a hívást, vagy-

is a feldolgozás sikertelen volt. Ekkor a sorfeldolgozást vezérlő számítógép felfrissíti a sorhoz tartozó sorinformációkat, hogy ezt a helyzetet tükrözze.

Ha a válasz nemleges, akkor a kiszolgálók a 287 logikai lépésre lépnek, ahol a kiszolgálók megjelölik a sort mint nem feldolgozás alatt levőt, és ezzel véget ér a vonatkozó sor feldolgozása.

Ha a 467 logikai lépésben az üzenet továbbítása sikertelen, akkor a találmány szerint a 484 logikai lépésben a hibakezelést különféleképpen lehet végezni. Egy lehetséges változatnál a sorfeldolgozást vezérlő számítógép kiemeli az üzenetet a sorból, és visszajuttatja a forrásához, majd befejezi a feldolgozósor feldolgozását a sorfeldolgozó vezérlő következő ciklusáig. Egy másik változatnál a sorfeldolgozó vezérlő az összes üzenetet eltávolítja a sorból, és visszaküldi azokat a forrásukhoz.

Egy további változatnál a sorfeldolgozó vezérlő az üzenetet a sorban hagyja, de megadja az újrapróbálkozási időt. Az ebből a sorból végzett üzenettovábbítási kísérletet újrapróbálkozásként kezeli. A sorfeldolgozó vezérlő számítógép nyilvántartja az újrapróbálkozások számát, és ezt figyeli a QM sorvezérlő is. A QM a soreldugulás ellenőrzése közben ügyel arra is, hogy a 8. ábrán a 324 logikai lépésben a dugulás ellenőrzése során ne legyen minden sor újrapróbálkozási állapotban.

Ha az üzenet több próbálkozás után sem jut át, akkor a sorfeldolgozást vezérlő számítógép visszajuttatja azt az üzenetet (és valószínűleg az összes többi, arra a célállomásra szánt üzenetet) a forrásához.

A feldolgozósorok feldolgozás kezdeményezése céljából végzett átvizsgálása közben a sorvezérlő átugorja azokat a sorokat, amelyek újrapróbálkozásra vannak megjelölve.

A 484 logikai lépésben végzett hibakezelés után a sorfeldolgozást vezérlő számítógép a 487 logikai lépésre lép, hogy megjelölje a sort, mint ami nincs feldolgozás alatt, és ezzel véget ér az adott sor feldolgozása. Ezt követően a 9. ábra szerint a következő sor feldolgozásának kezdeményezése következik.

Amennyiben a 474 logikai lépésben a sorfeldolgozást vezérlő számítógép megállapítja, hogy nincs több üzenet a sorban, vagy a 477 logikai lépésben azt állapítja meg, hogy a BE1 alállomás már feldolgozta a maximális csoportlétszámú üzenetet, akkor a számítógép a 487 logikai lépésre lép tovább.

Ha a hívás sikeresen lezajlott, akkor a sorfeldolgozást vezérlő számítógép előre meghatározott, maximális csoportlétszámú üzenetcsoportot küld el, és a továbbítás ezzel véget ér. A csoport előre meghatározott száma adott esetben nem mindig teszi lehetővé az összes üzenet elküldését az adott feldolgozósorból. Azonban a sorfeldolgozást vezérlő számítógép a következő ciklusa során újabb üzeneteket továbbíthat, amikor ellenőrzi a feldolgozósorokat.

A 11. ábrán látható folyamat a 9. és 10. ábrán szemléltetett folyamatok egyszerűsített változata. Ennél a kiviteli alaknál az egyik sor feldolgozását be kell fejezni, mielőtt a másik sor feldolgozása megkezdődhet. Ebben az esetben a 11. ábra szerinti lépések szintén függetlenek a QM sorvezérlő működésének az időzítésétől. Ennél a változatnál a sorfeldolgozást vezérlő számítógép kezdeményezi a QP1–QPn feldolgozósorok ellenőrzését, egyenként és előre meghatározott sorrendben, egy időben csak egyet vizsgálva. Az ellenőrzések között az 500 logikai lépésben várakozik. A folyamat egy várólépéssel indul. Ez rossz hatékonyságúnak tűnhet, de valójában nem az. Mivel a folyamatok ciklikusak, a kezdeti várás után a folyamat úgy folytatódik, mintha a várakozás a folyamat végén lenne. A legtöbb esetben a rendszer elindításakor még nincsenek jelen üzenetek, és ezért ha a folyamat nem várással indulna, akkor üresen lefutna az első várakozásig, anélkül, hogy bármit is végrehajtott volna. Ezért a várással való kezdés általában hatékonyabb. De létezhetnek más megvalósítási módok is. Az 504 logikai lépésben a sorfeldolgozást vezérlő számítógép ellenőrzi a következő üzenetsor készültségét, és az 507 megkérdezi, hogy a feldolgozósorban vannak-e üzenetek. Ha a válasz nemleges, akkor a sorfeldolgozást vezérlő számítógép az 510 logikai lépésre megy, és kitisztítja vagy eltávolítja az összes olyan sorinformációt, amely a sorban korábban tartalmazott üzenetekre vonatkozik, és megszünteti a sor hozzárendeltségét. Ezután a sorfeldolgozást vezérlő számítógép az 514 logikai lépésben elkezdi a következő sor ellenőrzését, és visszatér az 500 logikai lépésre. Ha a válasz az 507 logikai lépésben igen, akkor vannak üzenetek a feldolgozósorban, és a sorfeldolgozást vezérlő számítógép a BE1 alállomást utasítja, hogy tárcsázza az üzenetek célállomáskódját, és továbbítsa a feldolgozósorban álló első üzenetet a létrehozott csatornán keresztül. Az 567 lépésben a sorfeldolgozást vezérlő számítógép megvizsgálja, hogy a továbbítás sikeres volt-e. Ha igen, akkor a sorfeldolgozást vezérlő számítógép eltávolítja a üzenetet a sorból az 570 logikai lépésben. Az 574 logikai lépésben a sorfeldolgozást vezérlő számítógép ellenőrzi, hogy a feldolgozás alatt lévő sorban vannak-e még üzenetek. Ha a válasz igen, akkor a sorfeldolgozást vezérlő számítógép az 577 logikai lépésben megvizsgálja, hogy a megengedett legnagyobb számú üzenetet továbbította-e a BE1 alállomás. Nemleges esetben a sorfeldolgozást vezérlő számítógép az 580 logikai lépésben visszatér a következő üzenetért, majd az 520 logikai lépésben újraindítja a folyamatot.

Ha az 574 logikai lépésben megállapítható, hogy nincsen több üzenet a sorban, vagy az 577 logikai lépésben jelzik, hogy a BE1 alállomás maximális számú üzenetet továbbított, akkor a sorfeldolgozást vezérlő számítógép az 514 logikai lépésre megy.

Az 584 logikai lépés adott esetben mellőzhető is. Ha az 567 logikai lépésben a BE1 alállomás nem fejezte be a hívást, vagyis a feldolgozás sikertelen volt, akkor az 584 logikai lépésben felfrissítődnek a sorhoz tartozó sorinformációk, hogy ezt a helyzetet tükrözzék. Az 584 logikai lépésben az üzenet továbbítását ismételten megkísérlik. Ha az üzenet az ismételt többszöri próbálkozás után sem jut át, akkor a sorfeldolgozást vezérlő számítógép visszajuttatja azt az üzenetet (és valószínűleg az összes többi, arra a célállomásra szánt üzenetet) a forrásához.

Ha a hívás sikeresen zajlott, a sorfeldolgozást vezérlő számítógép előre meghatározott számú üzenetből álló csoportot továbbít, és ezzel a továbbítás befejeződik. A csoport előre meghatározott létszáma nem mindig teszi lehetővé a feldolgozósorban álló összes üzenet továbbítását. Azonban a sorfeldolgozást vezérlő számítógép újabb üzeneteket továbbíthat a következő sorellenőrzési ciklus alkalmával.

A találmány azzal az előnnyel jár, hogy biztosítja a hatékony kiszolgálást akkor is, ha a célállomások száma nagyon nagy, például ezer. A találmány azon a felismerésen alapul, hogy egy adott időpontban általában csak korlátozott számú célállomás vár kiszolgálásra. Így viszonylag kisszámú, például ötven QP1–QPn feldolgozósor is elég ahhoz, hogy nagyszámú célállomást szolgáljon ki. A vezérlősor egyben túlcsorduló sorként is szolgál, ha a feldolgozósorok száma elégtelennek bizonyul.

A találmány előnyösen lehetővé teszi, hogy meggátolható legyen az, hogy a leggyakoribb célállomásokra menő üzenetek kisajátítsák a feldolgozósorokat. Ez annak köszönhető, hogy minden egyes üzenetnek ki kell várnia a sorát a QC vezérlősorban.

A találmány szerint a QM sorvezérlő nem rendeli hozzá állandó jelleggel a QP1–QPn feldolgozósorokat egy rögzített célállomáshoz, hanem csak ideiglenesen rendeli hozzá azokat egy adott célállomáshoz. A QP1–QPn feldolgozósorok bármelyike akkor van használva, amikor éppen hozzárendelt állapotban van. Amikor a feldolgozósor kiürült, akkor ismét hozzárendelhetővé válik egy másik célállomáshoz. A QM sorvezérlő a feldolgozás közben is adhat át üzeneteket egy feldolgozósorba.

A találmány egy további ismérve szerint a sorfeldolgozást vezérlő számítógép további információkat fogad annak megállapítása érdekében, hogy mikor kezdeményezze a feldolgozósor működését. Például a továbbítási idő és a továbbítási számláló határozza meg azt, hogy mikor kezdődjön a továbbítás egy adott célállomásra, illetve megkezdődjön-e egyáltalán, illetve jelzi, hogy eddig mennyi próbálkozás történt, és további próbálkozás várható-e.

A találmány egy további kiviteli alakjánál a QC vezérlősor az időrenditől eltérő rendezési elvet alkalmaz a sorokban. Egy változatnál a QC vezérlősorban az üzenetek valamilyen prioritás vagy a FIFO és valamilyen prioritás kombinációjaként adódó sorrendben állnak. A QC vezérlősorban bármilyen skaláris mező használható prioritásként, és a feldolgozás történhet növekvő és csökkenő érték szerint egyaránt. Egy további változatnál a QC vezérlősor több különböző prioritást ír elő az egyes QP1–QPn feldolgozósorokban egy adott időpontban. A jobb hatékonyság érdekében használható a véletlenszerű hozzáférés vagy egy csatolt sorrend a FIFO-sor helyett. Egy másik megoldásnál a FIFO-sor egy kiegészítő változóval együtt kerül alkalmazásra, amely megmondja a pillanatnyi legnagyobb prioritású értéket a sorban. Ezek a megoldások kompromisszumot jelentenek a felállási idő és a következő üzenet megtalálásának hatékonysága között. Egy további lehetséges változatnál a QM sorvezérlő különböző feldolgozósorokat használ az ugyanolyan típusú, de különböző prioritású felhasználók számára.

A találmány egy további kiviteli módjánál a QM sorvezérlő a következőképpen kezeli a QP1–QPn feldolgozósorok bedugulását: A QM sorvezérlő felfüggeszti az üzenetek mozgását a QC vezérlősorból egészen addig, amíg legalább egy QP1–QPn feldolgozósor hozzáférhetővé válik. Az is megvalósítható, hogy a QM sorvezérlő kitisztítja azt a QP1–QPn feldolgozósort, amelynek az egyes sorokhoz tartozó sorinformációk alapján valamilyen feldolgozási problémája van. Ez néhány QP1–QPn feldolgozósort hozzáférhetővé tesz a dugulás közben, és összességében felgyorsítja a feldolgozást.

A dugulások kezelésének egy további módszerénél megvárjuk, hogy az összes QP1–QPn feldolgozósornál feldolgozási probléma merüljön fel, és egyszerre kitisztítjuk az összes QP1–QPn feldolgozósort. Ez egyszerre hozzáférhetővé teszi az összes QP1–QPn feldolgozó sort.

Telefonos vagy hangpostarendszer-környezetben alkalmazva a találmány szerint csak olyan feldolgozósorhoz rendelünk üzenetet, amely mentes egyéb, más célállomásra szánt üzenetektől. Így az azonos célállomásra szánt üzenetek azonos feldolgozósorba kerülnek akkor is, ha gyakorlatilag végtelen számú célállomás lehetséges, de csak korlátozott számú feldolgozósor. A forrásoktól az üzenetek először egy vezérlősorba érkeznek, és a QM sorvezérlő az egyik feldolgozósorba továbbítja azokat további feldolgozásra. Ezzel a QP1–QPn feldolgozósorok ismerete a QM sorvezérlőre és a sorfeldolgozást vezérlő számítógépre korlátozódik. Azok a külső folyamatok, amelyek az üzeneteket a QC vezérlősorba küldik, csak a QC vezérlősort ismerik.

A találmány lehetővé teszi a különböző célállomásra tartó üzenetek egyidejű feldolgozását, és megőrzi a közös célállomásra tartó üzenetek időrendi sorrendjét. A találmány dinamikusan használja a sorokat ugyanolyan típusú, például közös célú üzenetek tartására. A QP1–QPn feldolgozósorok egyike tárolhatja egy időben az egyik célállomásra tartó üzeneteket, míg egy másik időben egy más célállomásra tartó üzeneteket tárolhat. A QM sorvezérlő követi, hogy a feldolgozósor éppen használat közben van, újrapróbálkozási állapotban van, vagy pedig üres és hozzáférhető.

A találmány lehetővé teszi az üzeneteknek az időrendi sorrend mellett vagy helyett prioritási sorrendbe állítását is. A találmány szerinti megoldás korlátozott számú sort használ nagyszámú célállomás egyidejű kiszolgálásához. Gyakorlatilag nincsen korlátja a különböző célállomások számának. Ugyanarra a célállomásra szánt üzenetek ugyanabban a feldolgozósorban várnak. Egy feldolgozósorban az üzenetek általában időrendi sorrendben vannak, de más megközelítés is használható igény szerint. A különböző üzenetek egyidejű feldolgozása könnyen megvalósítható. A rendszer dinamikusan rendeli hozzá a QP1–QPn feldolgozósorokat az egyes célállomásokhoz, és szünteti meg a hozzárendeltséget a célállomástól. A találmány szerint egy adott üzenet feldolgozása szempontjából irányadó, a feldolgozósorral

közös tulajdonságot társít a feldolgozósorhoz a feldolgozás megkönnyítése érdekében.

Több, azonos célállomásra menő üzenet feldolgozása hatékonyabb, mint az egysoros rendszereknél, és a korábbi ismert többsoros rendszereknél is. A találmány nagy rugalmasságot biztosít a feldolgozáshoz.

Ha az üzenet továbbítása nem volt sikeres, akkor a találmány több kiviteli alakja végzi a hibakezelést. Egy lehetséges megoldásnál a sorfeldolgozás-vezérlő vagy a sorvezérlő eltávolítja az üzenetet a sorból, visszaküldi a forrásához, és befejezi a feldolgozást a feldolgozásvezérlő következő ciklusáig. Egy másik kiviteli alaknál a sorfeldolgozás-vezérlő vagy a sorvezérlő az összes üzenetet eltávolítja. Egy újabb további megoldásnál a sorfeldolgozás-vezérlő vagy a sorvezérlő az üzenetet a sorban hagyja, de megad egy időtartamot az újbóli kísérletre. A feldolgozás megkezdése céljából végzett ciklusok során a sorfeldolgozás-vezérlő átugorja a sort a megadott időtartamig.

A találmány egyes kiviteli alakjainak részletes ismertetése mellett a szakember számára nyilvánvaló, hogy a találmánynak további kiviteli módjai is megvalósíthatók a találmányi gondolattól és az oltalmi körtől való eltérés nélkül. Például itt hangpostarendszerben alkalmazott sorvezérlőt ismertettünk, de nyilvánvaló, hogy szakemberek a fent leírt eljárást és berendezést alkalmazhatják egyéb üzeneteket tároló és továbbító rendszerekben is. Például a jelen találmány használható olyan tároló és továbbító rendszerekben, mint például fax, elektronikus levelezés (e-mail), videoüzenetek, valamint mindenfajta multimédiaüzenetek tárolására és továbbítására szolgáló rendszerekben.

SZABADALMI IGÉNYPONTOK

1. Sorban állást vezérlő rendszer (QMS) több különböző típusú felhasználó (CL1-CLm) kiszolgálására, amelynek a sorban álló felhasználókat (CL1–CLm) előnyösen érkezési sorrendben fogadó és tartó vezérlősora (QC), továbbá a felhasználók (CL1–CLm) fogadására, és a felhasználóktól (CL1–CLm) való időszakonkénti kiürítésre több feldolgozósora (QP1–QPn) valamint, a vezérlősorból (QC) a felhasználókat feldolgozósorokba (QP1–QPn) átvivő sorvezérlőt (QM) tartalmaz, *azzal jellemezve,* hogy a felhasználók (CL1–CLm) típusainak számánál kevesebb számú, a különböző felhasználók (CL1–CLm) típusához hozzárendelhető feldolgozósora (QP1–QPn), valamint

a vezérlősorból (QC) a felhasználókat (CL1–CLm) egy, más felhasználóktól (CL1–CLm) vagy más típusú felhasználóktól (CL1–CLm) mentes feldolgozósorba (QP1–QPn) átvivő, és a feldolgozósor (QP1–QPn) kiürüléséig a feldolgozósort (QP1–QPn) az adott egy vagy több felhasználó (CL1–CLm) típusához hozzárendelő sorvezérlője (QM) van.

2. Az 1. igénypont szerinti rendszer, *azzal jellemezve,* hogy a sorvezérlőnek (QM) olyan, egy adott típusú felhasználót (CL1–CLm) a vezérlősorban (QC) tartó eszköze van, amely üres feldolgozósor (QP1–QPn) és egy vagy több, az adott típusú felhasználót (CL1–CLm) tartalmazó feldolgozósor (QP1–QPn) hiányában tartja az adott típusú felhasználót (CL1–CLm) a vezérlősorban (QC).

3. Az 1. igénypont szerinti rendszer, *azzal jellemezve,* hogy a sorvezérlőnek (QM)

a vezérlősorban (QC) álló felhasználó (CL1–CLm) típusát megállapító eszköze,

a megállapított típusú felhasználót (CL1–CLm) tartalmazó feldolgozósor (QP1–QPn) létezését megállapító eszköze, és

a felhasználót (CL1–CLm) a megfelelő feldolgozósorba (QP1–QPn), vagyis a megállapított típusú felhasználót (CL1–CLm) tartalmazó feldolgozósorba (QP1–QPn) juttató eszköze van.

4. A 3. igénypont szerinti rendszer, *azzal jellemezve,* hogy a sorvezérlőnek (QM)

megfelelő feldolgozósor (QP1–QPn) hiányában üres feldolgozósort (QP1–QPn) kereső eszköze, és

üres feldolgozósor (QP1–QPn) létezése esetén a felhasználót (CL1–CLm) az üres feldolgozósorba (QP1–QPn) juttató eszköze van.

5. A 4. igénypont szerinti rendszer, *azzal jellemezve,* hogy a sorvezérlőnek (QM)

üres feldolgozósor (QP1–QPn) hiányában a feldolgozósor (QP1–QPn) kiürítését megvizsgáló eszköze, és

a feldolgozósor (QP1–QPn) kiürítésének hiánya esetén legalább egy feldolgozósort (QP1–QPn) kiürítő és a felhasználót (CL1–CLm) a vezérlősorból (QC) kiürített feldolgozósorba (QP1–QPn) juttató eszköze van.

6. Az 5. igénypont szerinti rendszer, *azzal jellemezve,* hogy a sorvezérlőnek (QM)

az összes feldolgozósor (QP1–QPn) kiosztása és a feldolgozósorok (QP1–QPn) folyamatos kiürítése esetén a felhasználóknak (CL1–CLm) a vezérlősorból (QC) való eltávolításának folyamatát felfüggesztő eszköze van.

7. Az 1. igénypont szerinti rendszer, *azzal jellemezve,* hogy

az egyes feldolgozósorokban (QP1–QPn) a felhasználókat (CL1–CLm) csoportonként feldolgozó eszköze,

az egyes csoportokban megengedett felhasználók (CL1–CLm) legnagyobb számát megállapító eszköze, valamint

ennek a legnagyobb számnak az elérését megállapító eszköze van.

8. A 7. igénypont szerinti rendszer, *azzal jellemezve,* hogy a feldolgozóeszköznek

feldolgozósor (QP1–QPn) felhasználóit (CL1–CLm) ellenőrző eszköze, és

az ellenőrzött feldolgozósorban (QP1–QPn) felhasználók (CL1–CLm) hiánya esetén a következő feldolgozósor (QP1–QPn) felhasználóit (CL1–CLm) ellenőrző eszköze, valamint

az ellenőrzött feldolgozósor (QP1–QPn) (QP1–QPn) feldolgozásának tényét vizsgáló eszköze, és

a feldolgozósor (QP1–QPn) (QP1–QPn) feldolgozásának hiányában a feldolgozást elindító eszköze van.

9. Eljárás sorban álló, különböző típusú üzenetek (ME1–MEm) feldolgozásának vezérlésére egy üzenettovábbító rendszerben (QMT), különösen hangposta-to-

vábbító rendszerben, ahol a különböző típusú üzenetekhez (ME1–MEm) különböző típusú feldolgozóművelet, különösen továbbítási művelet tartozik, *azzal jellemezve,* hogy

az üzeneteket (ME1–MEm) egy vezérlősorba (QC) juttatjuk,

az üzenetek (ME1–MEm) típusainál kevesebb számú, több feldolgozósort (QP1–QPn) időszakonként kiürítünk, és

a vezérlősorból (QC) az üzenetet (ME1–MEm) olyan feldolgozósorba (QP1–QPn) juttatjuk, amely mentes másfajta típusú üzenetektől (ME1–MEm), és a feldolgozósort (QP1–QPn) annak kiürüléséig egyetlen üzenettípushoz rendeljük.

10. A 9. igénypont szerinti eljárás, *azzal jellemezve,* hogy az üzenetet (ME1–MEm) a vezérlősorban (QC) tartjuk vissza, ha nem áll rendelkezésre egy vagy több azonos típusú üzenetet (ME1–MEm) tartalmazó feldolgozósor (QP1–QPn).

11. A 9. igénypont szerinti eljárás, *azzal jellemezve,* hogy az üzeneteknek (ME1–MEm) a feldolgozósorba (QP1–QPn) való juttatása során

megállapítjuk a vezérlősorban (QC) álló üzenet (ME1–MEm) típusát,

megállapítjuk, hogy létezik-e a megállapított típusú üzeneteket (ME1–MEm) tartalmazó feldolgozósor (QP1–QPn), és

az üzenetet (ME1–MEm) egy megfelelő, vagyis a megállapított típussal egyező típusú üzenetet (ME1–MEm) tartalmazó feldolgozósorba (QP1–QPn) juttatjuk.

12. A 11. igénypont szerinti eljárás, *azzal jellemezve,* hogy az üzeneteknek (ME1–MEm) a feldolgozósorba (QP1–QPn) való juttatása során

megfelelő feldolgozósor (QP1–QPn) hiányában üres feldolgozósort (QP1–QPn) keresünk, és

üres feldolgozósor (QP1–QPn) létezése esetén az üzenetet (ME1–MEm) egy üres feldolgozósorba (QP1–QPn) juttatjuk.

13. A 12. igénypont szerinti eljárás, *azzal jellemezve,* hogy az üzeneteknek (ME1–MEm) a feldolgozósorba (QP1–QPn) való juttatása során

üres feldolgozósor (QP1–QPn) hiányában megvizsgáljuk, hogy valamelyik feldolgozósor (QP1–QPn) kiürítése folyamatban van-e, és

ha nincs folyamatban egy feldolgozósor (QP1–QPn) kiürítése sem, akkor legalább egy feldolgozósort (QP1–QPn) kiürítünk, és a vezérlősorból (QC) egy üzenetet (ME1–MEm) egy kiürített sorba juttatunk.

14. A 13. igénypont szerinti eljárás, *azzal jellemezve,* hogy az üzeneteknek (ME1–MEm) a feldolgozósorba (QP1–QPn) való juttatása során

a felhasználók (CL1–CLm) vezérlősorból (QC) való eltávolításának folyamatát felfüggesztjük, ha az összes feldolgozósorhoz (QP1–QPn) van hozzárendelve üzenettípus, és a feldolgozósorok (QP1–QPn) folyamatosan kiürítésre kerülnek.

15. A 9. igénypont szerinti eljárás, *azzal jellemezve,* hogy

az üzeneteket (ME1–MEm) az egyes feldolgozósorokban (QP1–QPn) csoportonként dolgozzuk fel,

megállapítjuk az egyes csoportokban megengedett üzenetek (ME1–MEm) legnagyobb számát, és

megállapítjuk, hogy ezt a legnagyobb számot elértük-e.

16. A 15. igénypont szerinti eljárás, *azzal jellemezve,* hogy az üzenetek (ME1–MEm) feldolgozása során

ellenőrizzük a feldolgozósorban (QP1–QPn) az üzeneteket (ME1–MEm), és

ha az ellenőrzött sorban nincsenek üzenetek (ME1–MEm), akkor a következő feldolgozósor (QP1–QPn) üzeneteit ellenőrizzük.

17. Üzenettároló és -továbbító rendszer, amely

több cél közül egy meghatározott célba továbbítandó, célállomáskóddal ellátott üzeneteket (ME1–MEm) fogadó eszközzel,

az üzeneteket (ME1–MEm) fogadó és célszerűen érkezési sorrendben sorba állító vezérlősorral (QC), és több, az üzeneteket (ME1–MEm) a célállomásokba juttató feldolgozósorral (QP1–QPn) és a vezérlősorból (QC) célállomáskóddal ellátott üzeneteket a feldolgozósorba (QP1–QPn) átvivő sorvezérlővel (QM) rendelkezik, *azzal jellemezve,* hogy

a célállomások számánál kevesebb számú feldolgozósorral (QP1–QPn),

a vezérlősorból (QC) egy célállomáskóddal ellátott üzeneteket (ME1–MEm) más célállomáskódhoz tartozó üzenetektől (ME1–MEm) mentes feldolgozósorba (QP1–QPn) átvivő, és a feldolgozósor (QP1–QPn) kiürüléséig a feldolgozósort (QP1–QPn) az adott célállomáskóddal ellátott üzenetekhez (ME1–MEm) hozzárendelő sorvezérlővel (QM) van ellátva.

18. A 17. igénypont szerinti rendszer, *azzal jellemezve,* hogy a sorvezérlőnek (QM) olyan, egy adott célállomáskóddal ellátott üzenetet (ME1–MEm) a vezérlősorban (QC) tartó eszköze van, amely üres feldolgozósor (QP1–QPn) és egy vagy több, az adott célállomáskóddal ellátott üzenetet (ME1–MEm) tartalmazó feldolgozósor (QP1–QPn) hiányában visszatartja az adott típusú üzenetet (ME1–MEm) a vezérlősorban (QC).

19. A 17. igénypont szerinti rendszer, *azzal jellemezve,* hogy a sorvezérlőnek (QM)

a vezérlősorban (QC) álló üzenet (ME1–MEm) célállomáskódját megállapító eszköze,

a megállapított célállomáskóddal rendelkező üzenetet (ME1–MEm) tartalmazó feldolgozósor (QP1–QPn) létezését megállapító eszköze, és

az üzenetet (ME1–MEm) a megfelelő feldolgozósorba (QP1–QPn), vagyis a megállapított célállomáskóddal rendelkező üzenetet (ME1–MEm) tartalmazó sorba juttató eszköze van.

20. A 19. igénypont szerinti rendszer, *azzal jellemezve,* hogy a sorvezérlőnek (QM)

megfelelő feldolgozósor (QP1–QPn) hiányában üres feldolgozósort (QP1–QPn) kereső eszköze, és

üres feldolgozósor (QP1–QPn) létezése esetén az üzenetet (ME1–MEm) az üres feldolgozósorba (QP1–QPn) juttató eszköze van.

21. A 20. igénypont szerinti rendszer, *azzal jellemezve,* hogy a sorvezérlőnek (QM)

üres feldolgozósor (QP1–QPn) hiányában a feldolgozósor (QP1–QPn) kiürítését megvizsgáló eszköze, és

a feldolgozósor (QP1–QPn) kiürítésének hiánya esetén legalább egy feldolgozósort (QP1–QPn) kiürítő és egy üzenetet (ME1–MEm) a vezérlősorból (QC) a kiürített feldolgozósorba (QP1–QPn) juttató eszköze van.

22. Az 21. igénypont szerinti rendszer, *azzal jellemezve,* hogy a sorvezérlőnek (QM)

az összes feldolgozósor (QP1–QPn) kiosztása és a feldolgozósorok (QP1–QPn) folyamatos kiürítése esetén az üzeneteknek (ME1–MEm) a vezérlősorból (QC) való eltávolításának folyamatát felfüggesztő eszköze van.

23. Az 17. igénypont szerinti rendszer, *azzal jellemezve,* hogy

az egyes feldolgozósorokban (QP1–QPn) az üzeneteket (ME1–MEm) csoportonként feldolgozó eszköze,

az egyes csoportokban megengedett üzenetek (ME1–MEm) legnagyobb számát megállapító eszköze, valamint

ennek a legnagyobb számnak az elérését megállapító eszköze van.

24. A 23. igénypont szerinti rendszer, *azzal jellemezve,* hogy a feldolgozóeszköznek

feldolgozósor (QP1–QPn) üzeneteit (ME1–MEm) ellenőrző eszköze, és

az ellenőrzött feldolgozósorban (QP1–QPn) üzenetek (ME1–MEm) hiánya esetén a következő feldolgozósor (QP1–QPn) üzeneteit (ME1–MEm) ellenőrző eszköze, valamint

az ellenőrzött feldolgozósor (QP1–QPn) feldolgozásának tényét vizsgáló eszköze, és

a feldolgozósor (QP1–QPn) feldolgozásának hiányában a feldolgozást elindító eszköze van.

25. A 9–16. igénypontok szerinti eljárás, *azzal jellemezve,* hogy az üzenetek (ME1–MEm) egy adott típusának több cél közül egy meghatározott célba továbbítandó üzeneteket (ME1–MEm) választjuk, és a különböző üzenettípusokhoz különböző célállomáskódot rendelünk, és a célállomások számánál kevesebb számú feldolgozósort (QP1–QPn) alkalmazunk.

26. A 25. igénypont szerinti eljárás, *azzal jellemezve,* hogy a feldolgozósorokhoz (QP1–QPn) ideiglenesen hozzárendeljük a feldolgozás alatti üzenetek (ME1–MEm) célállomáskódját, és az üzenetet (ME1–MEm) a vezérlősorban (QC) tartjuk vissza, ha nem áll rendelkezésre egy vagy több azonos célállomáskódú feldolgozósor (QP1–QPn).

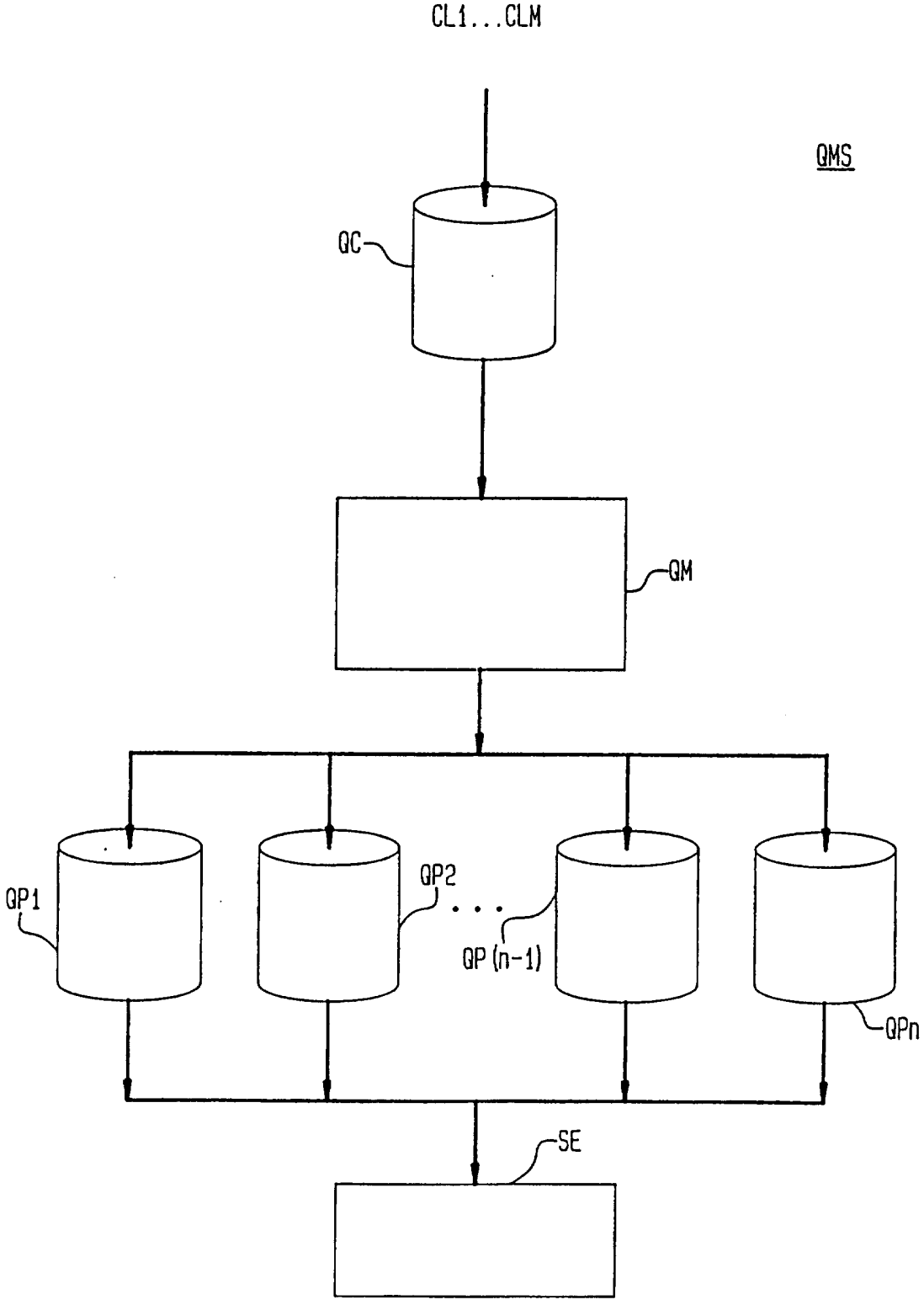
27. Az 5. igénypont szerinti rendszer, *azzal jellemezve,* hogy a legalább egy feldolgozósort (QP1–QPn) kiürítő eszköznek a felhasználót a kiürített feldolgozósorból (QP1–QPn) a vezérlősorba (QC) visszajuttató eszköze van.

28. A 4. igénypont szerinti rendszer, *azzal jellemezve,* hogy a sorvezérlő (QM)

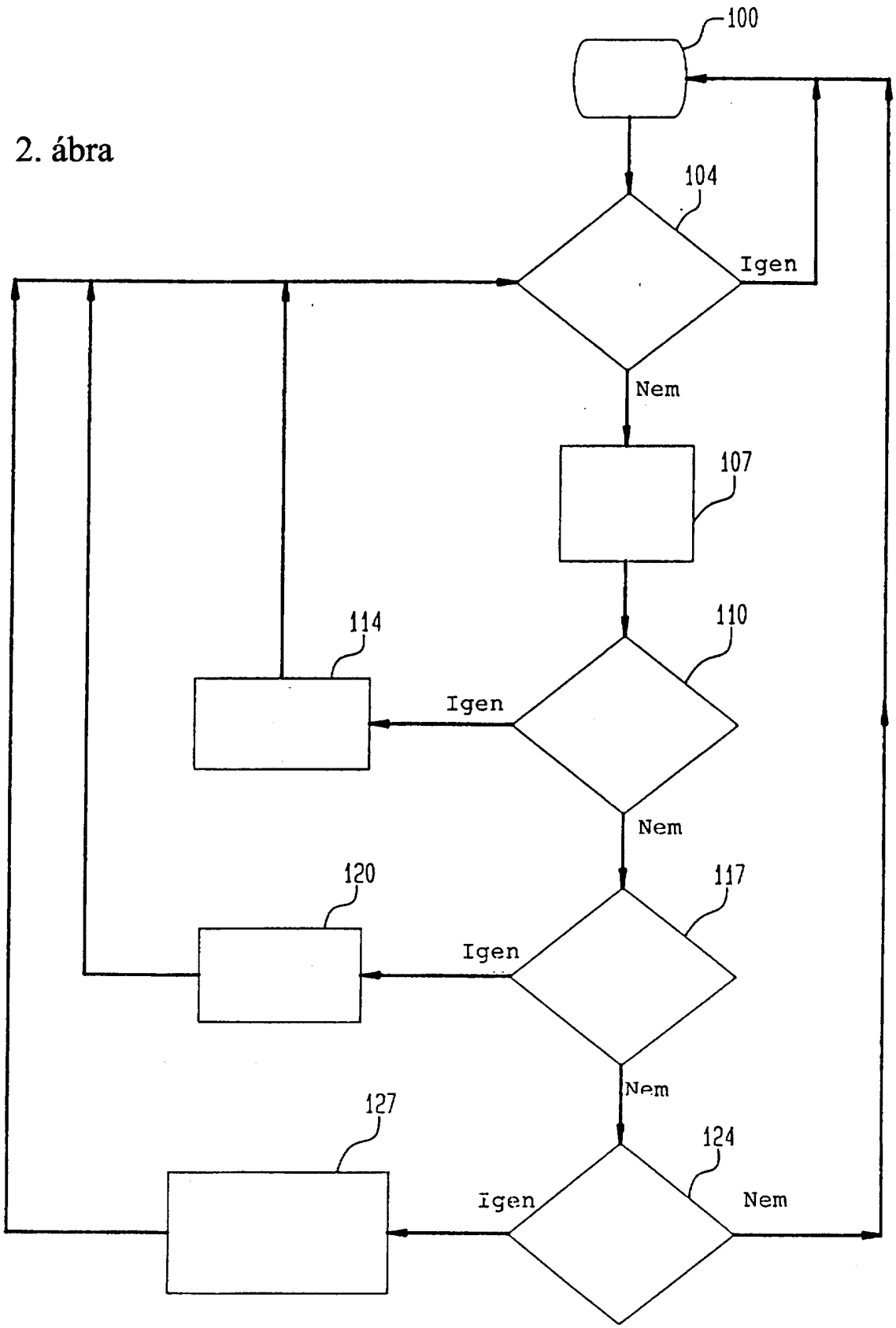
üres feldolgozósor (QP1–QPn) hiányában valamelyik feldolgozósor (QP1QPn) kiürítésének tényét megvizsgáló eszközzel van ellátva, és

feldolgozósor (QP1–QPn) kiürítésének hiányában legalább egy feldolgozósort (QP1–QPn) kiürítő és a kiürített feldolgozósor (QP1–QPn) felhasználóinak típusával megegyező típusú felhasználókat (CL1–CLm) a vezérlősorból (QC) eltávolító eszközzel van ellátva.

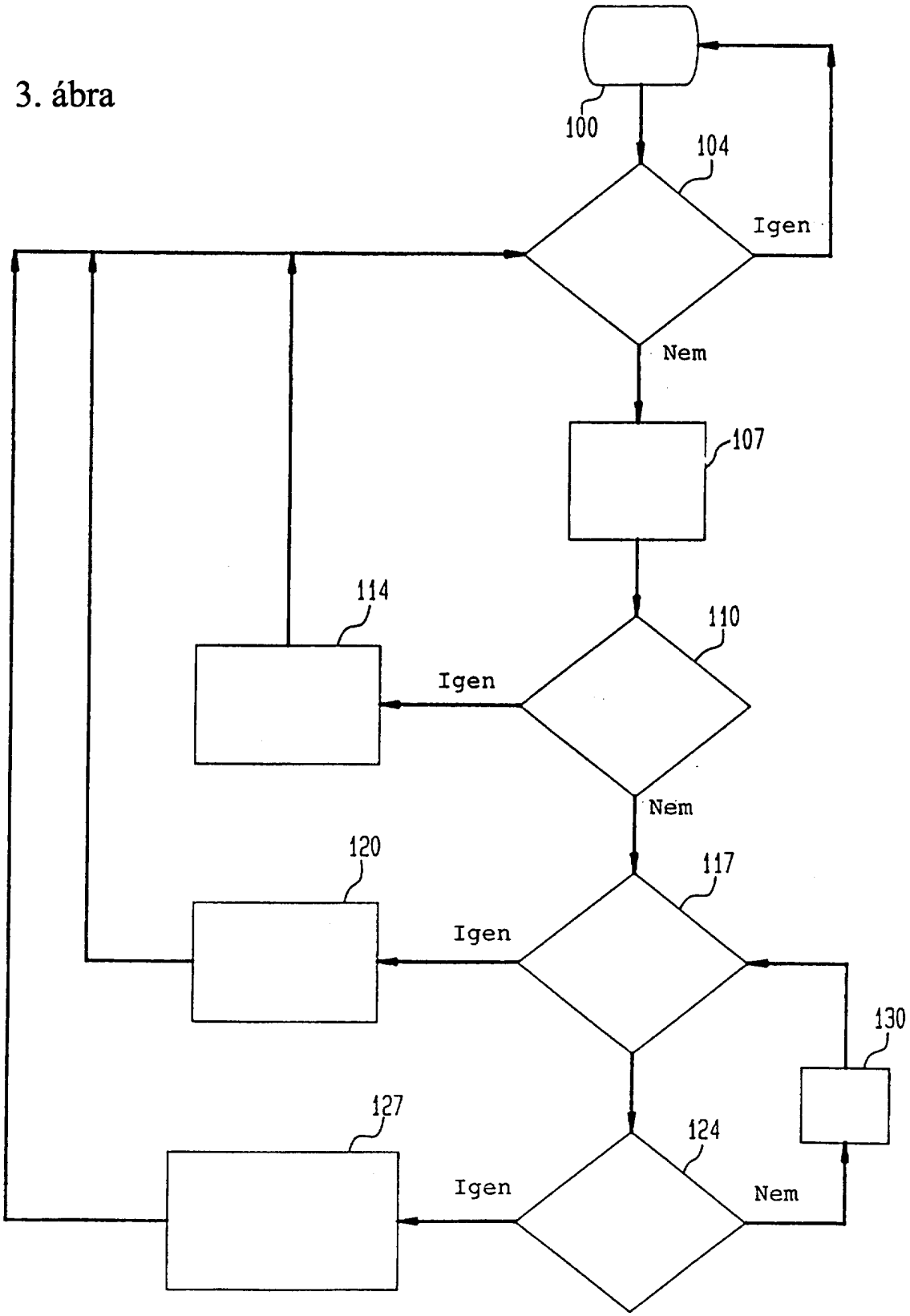
1. ábra



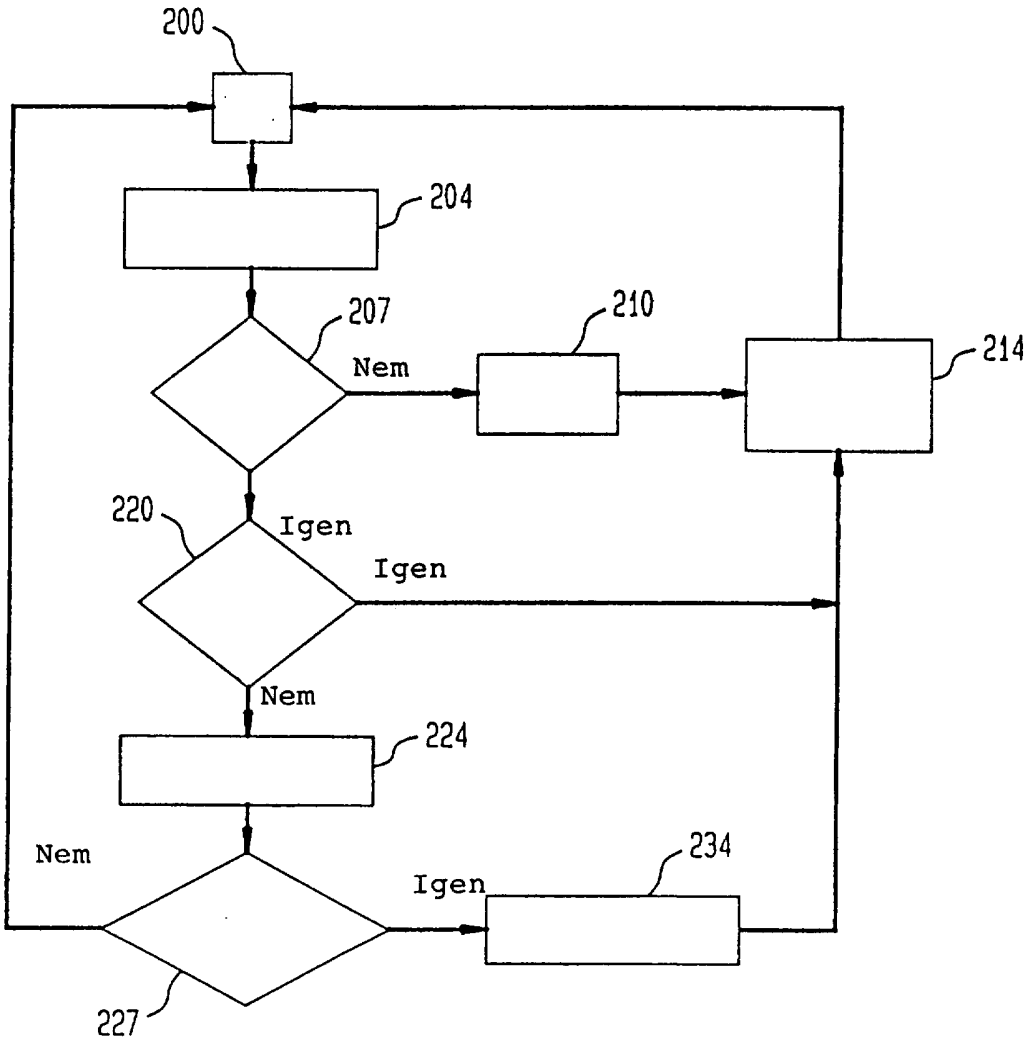
2. ábra


100
104
Igen
Nem
107
114
110
Igen
Nem
120
117
Igen
Nem
127
124
Igen
Nem

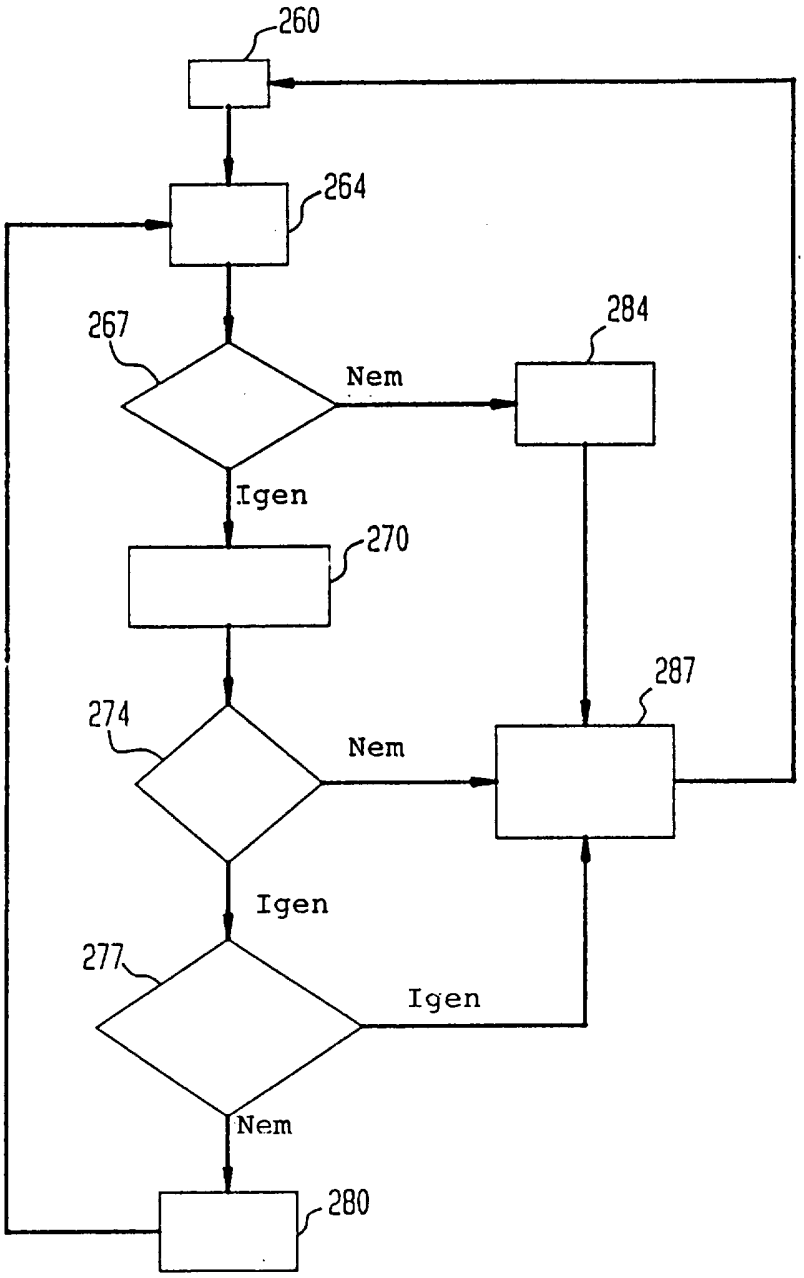
3. ábra


100
104
Igen
Nem
107
114
110
Igen
Nem
120
117
Igen
130
127
124
Igen
Nem

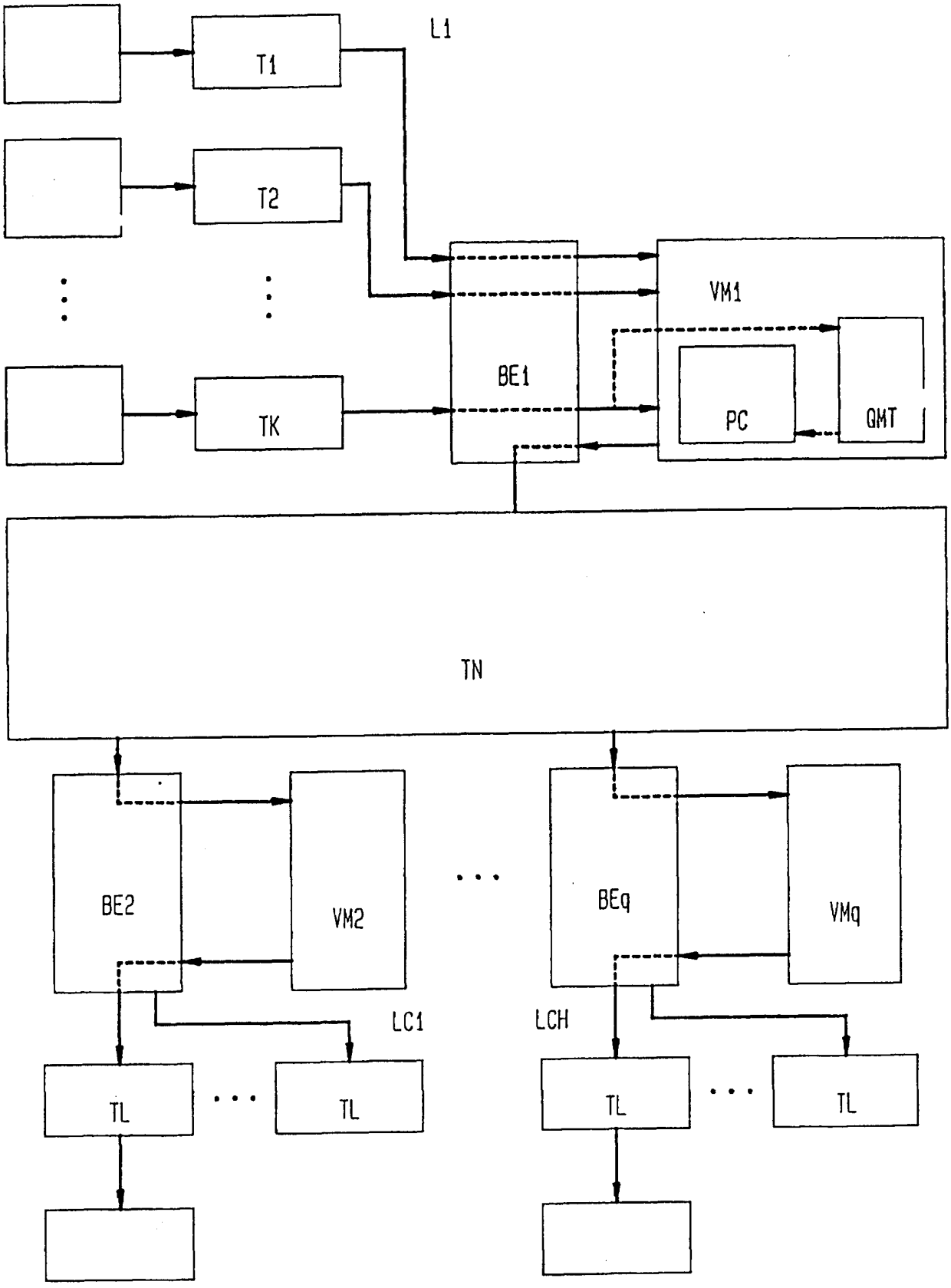
4. ábra



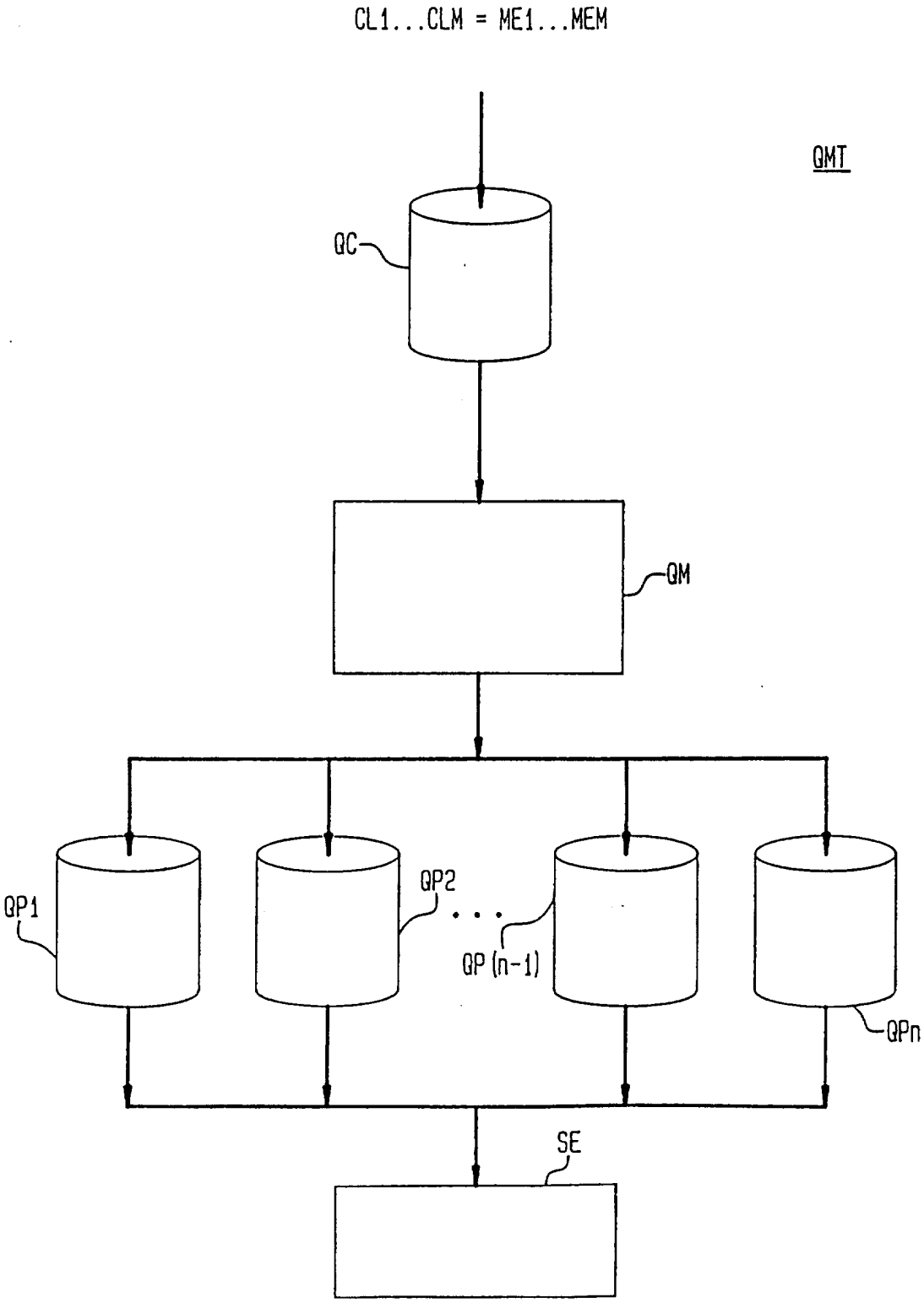
5. ábra



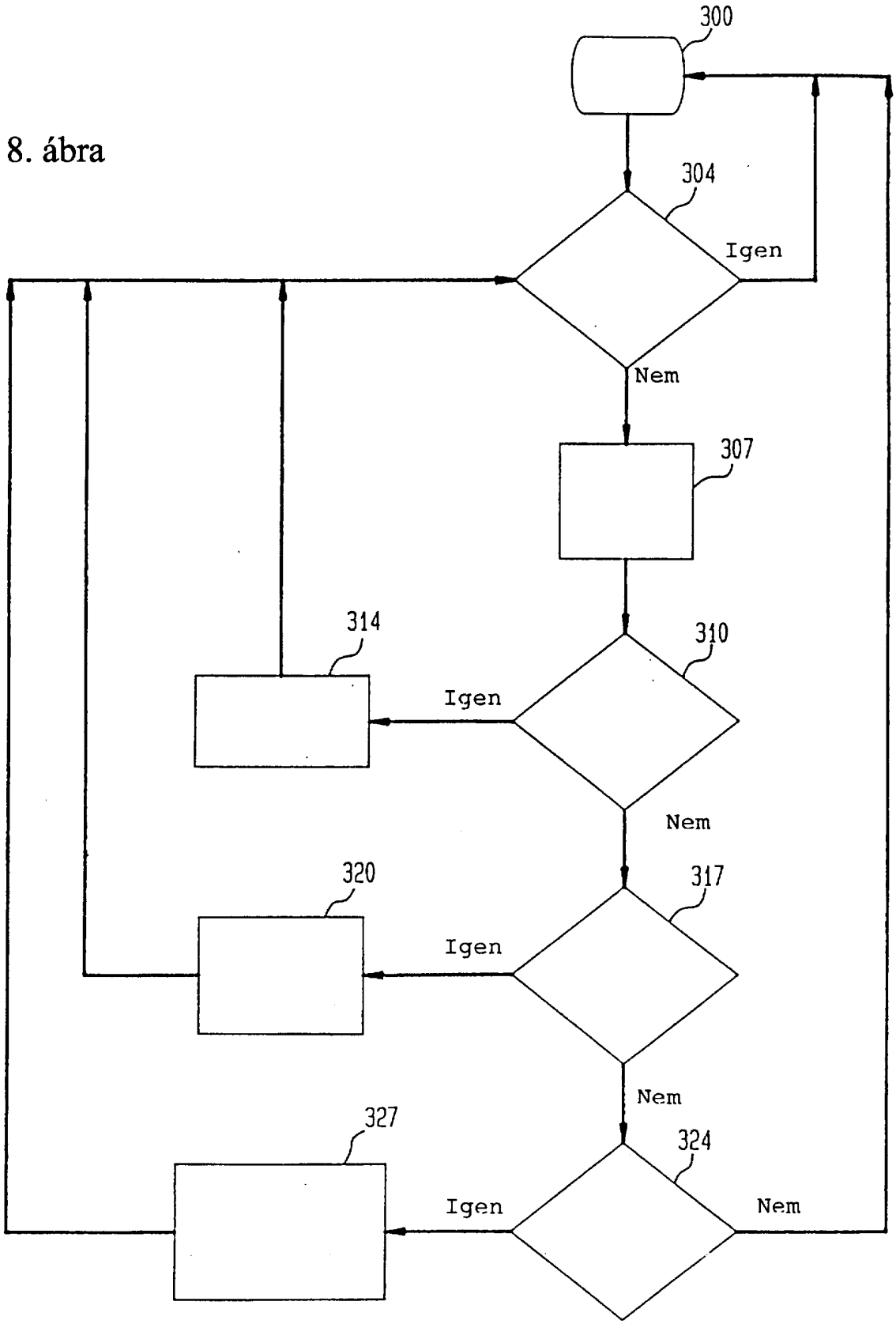
6. ábra



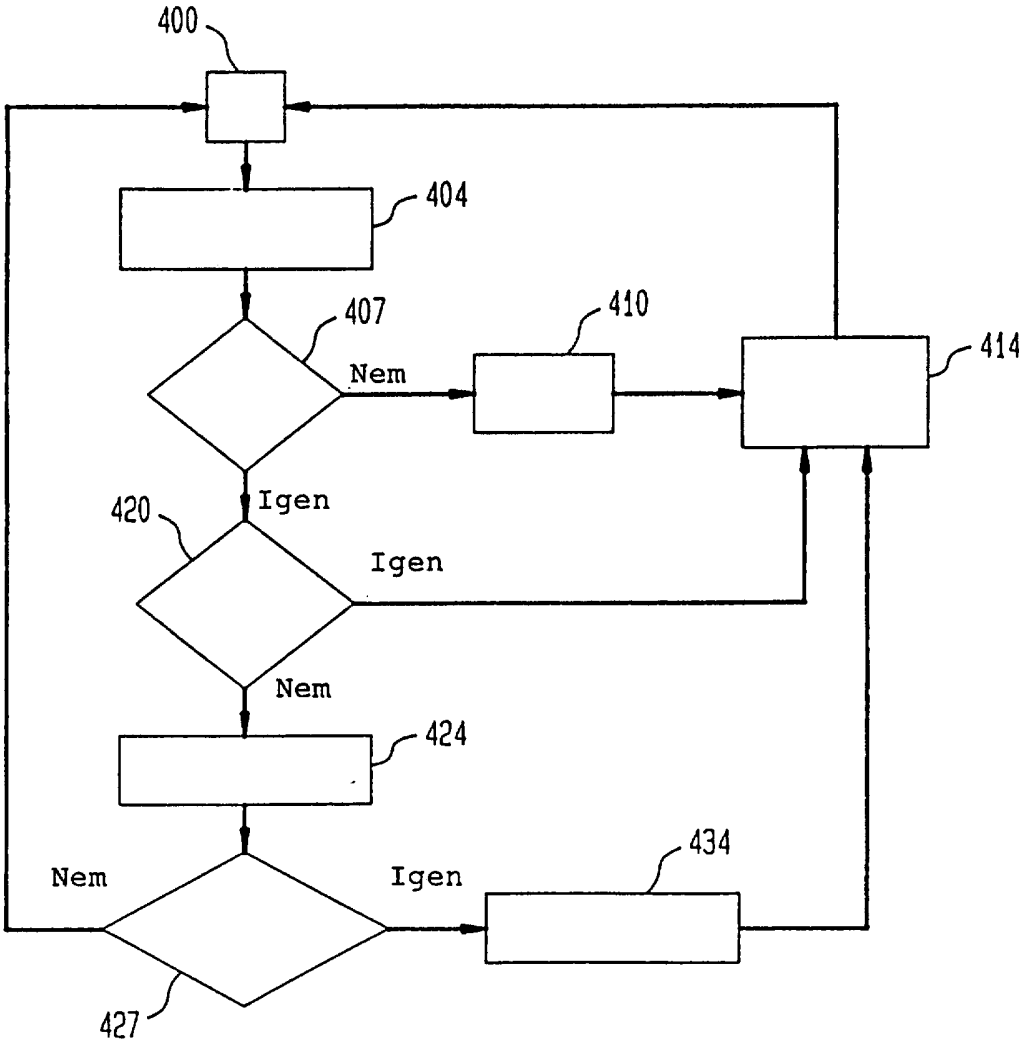
7. ábra



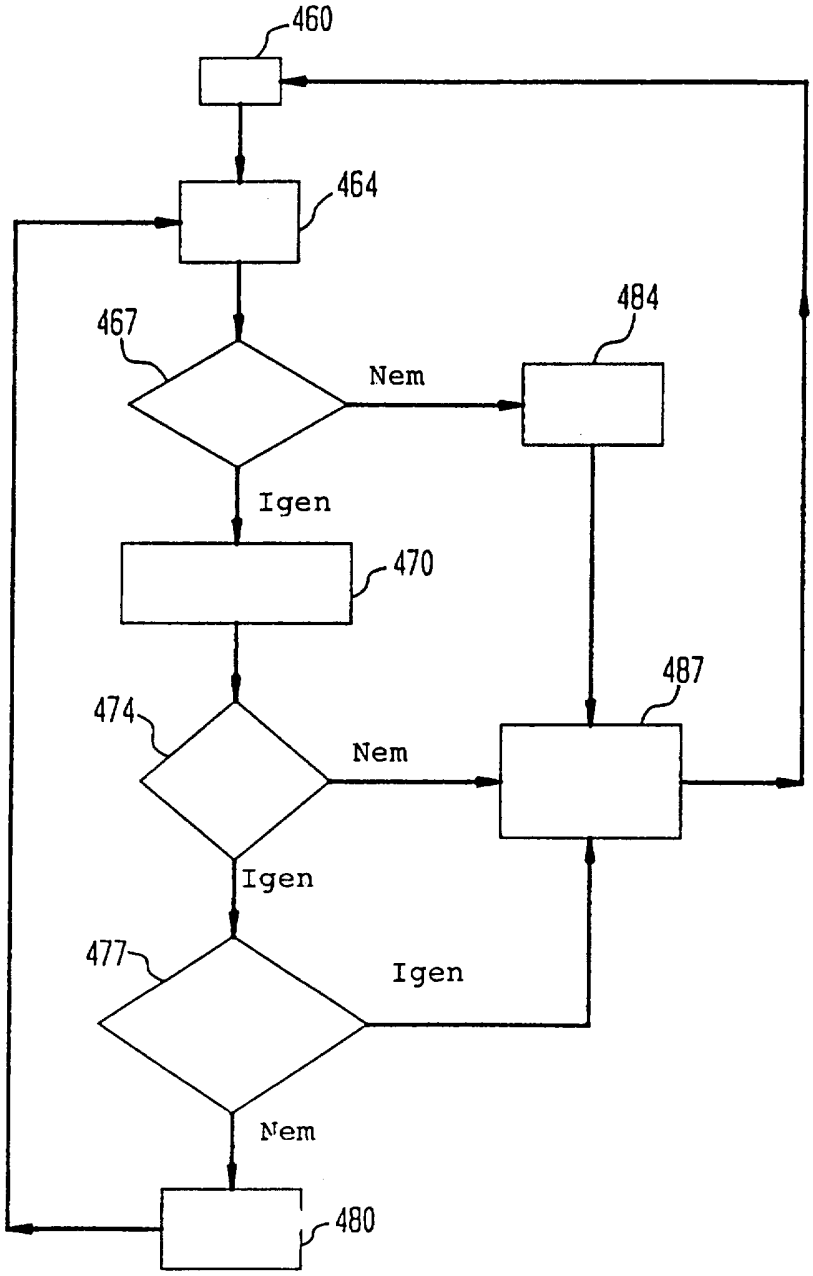
8. ábra


300
304
Igen
Nem
307
314
310
Igen
Nem
320
317
Igen
Nem
327
324
Igen
Nem

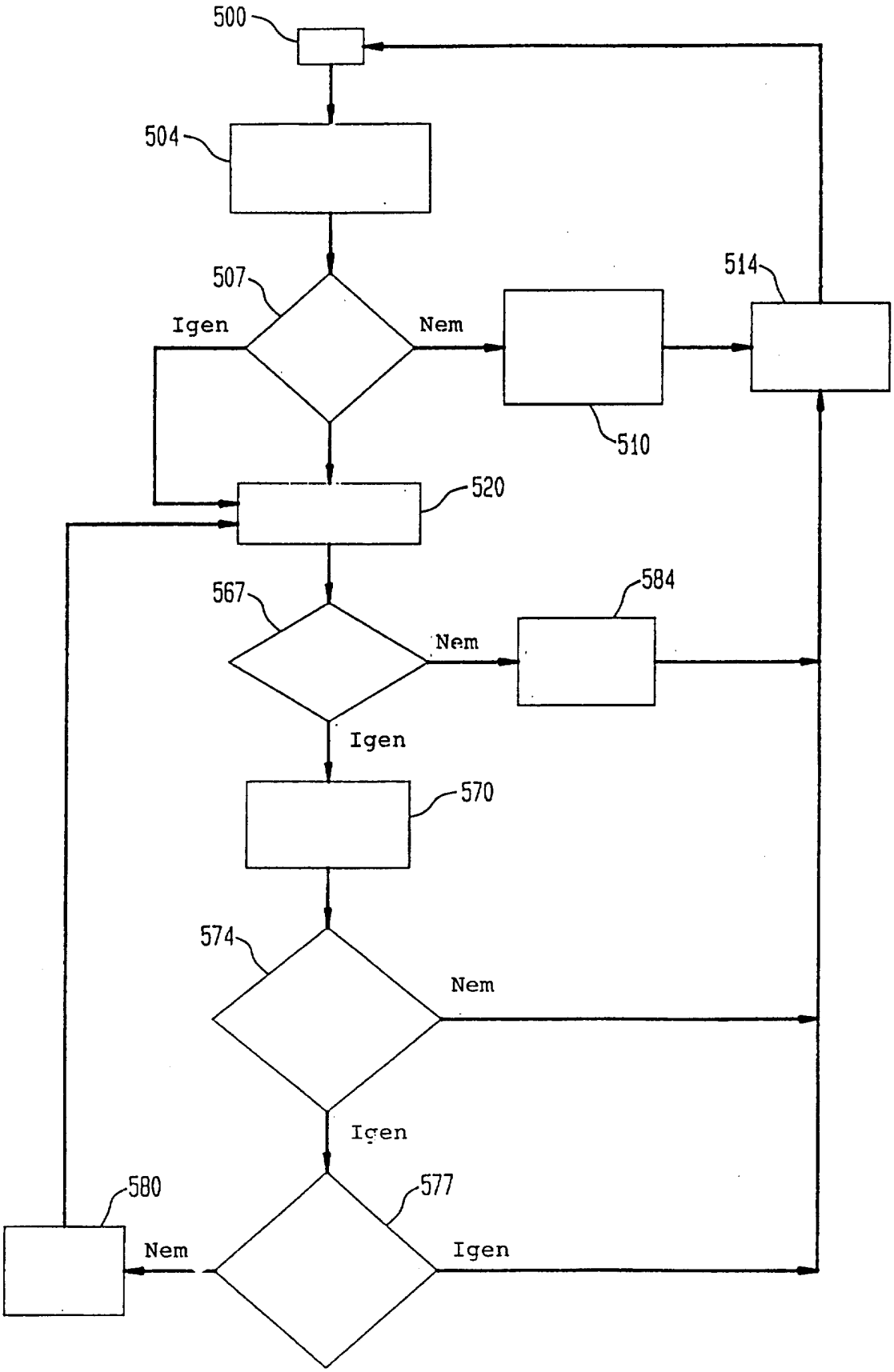
9. ábra



10. ábra



11. ábra


500
504
507
Igen
Nem
514
510
520
567
584
Nem
Igen
570
574
Nem
Igen
580
577
Nem
Igen