

~~73 131/KÖ~~PÖZZÉTÉTELI
PÉLDÁNY *AL*

Kivonat

Rugalmas interfész berendezés és hozzá eljárás

A találmány tárgya rugalmas interfész berendezés és eljárás. A találmány szerinti eljárásnál és berendezésnél a rugalmas interfész magában foglalja tároló egységek valamely sokaságát adat értékek egy folyamának tárolására, ahol minden egyes tároló egység szekvenciálisan tárolja adat értékek vonatkozó halmazainak elemeit. Minden egyes adat értéket egy lokális órajel előre meghatározott számú periódusáig tárolunk. Kiválasztási áramkört csatlakoztathatunk a tároló egységekhez a vonatkozó adat értéknek az adat folyamból való kiválasztására a megfelelő tároló egységben való tároláshoz. Az adatot szekvenciálisan tesszük kimenetre minden egyes tároló egységből a lokális órajellel szinkronban, és a lokális órajel egy cél ciklusán.

(4 *A* ábra)

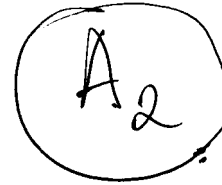
**KÖZZÉTÉTELI
PÉLDÁNY**

73.131/KÖ

S.B.G. & K.
Nemzetközi
Szabadalmi Iroda
H-1062 Budapest, Andrásy út 113.
Telefon: 34-24-950, Fax: 34-24-323

P 02002837

2001. SZEPTEMBER 4.



Rugalmas interfész berendezés és hozzá eljárás

A jelen találmány általánosságban adatfeldolgozó rendszerekkel, szűkebb értelemben valamely adatfeldolgozó rendszerben lévő dinamikus, vagy órajelezett integrált áramköri chippek közötti interfésszel kapcsolatos.

A modern adatfeldolgozó rendszerek megkövetelik adat átvitelét a rendszerben lévő több chipbe ágyazott dinamikus vagy órajelezett áramkörök között. Például szükség lehet adat átvitelére a központi feldolgozó egységek (CPU - central processing unit) között egy többprocesszoros rendszerben, vagy egy CPU és a memória rendszer (ami magában foglalhat egy memória-vezérlőt és chipen kívüli cache-t) között. Az adatátvitel szinkronizáltak, és az adattól azt várjuk, hogy a chipen lévő áramkörhöz egy előre meghatározott rendszer cikluson érkezzen. A CPU sebességek növekedésével a chippek közötti interfész sebessége (a sín ciklus idő) vált a korlátozó tényezővé, amint az interfészen keresztüli késés meghaladja a rendszer órajel periódusát. Annak érdekében, hogy fenntartsuk a rendszer szinkronizációját, a rendszer tervezőjének le kell lassítania a sín sebességét azért, hogy az a ciklus, amelyen az adat érkezik, egyértelmű legyen.

Ezt jobban megérthetjük az 1A ábrára utalva, ahol blokk diagramos formában ábrázolunk egy eddig használatos interfészt



két integrált áramköri chip, a 102 chip és a 104 chip között egy adatfeldolgozó rendszerben. A 102 és 104 chipek mindegyike fogad egy 106 referencia órajelet, amelyet egy 108 fázis igazító hurokhoz (PLL - phase lock loop) csatlakoztatunk. A 108 PLL egy 110 lokális órajelet generál a 102 chipben és 111 lokális órajelet a 104 chipben, ami a 106 referencia órajelhez van igazítva. A 106 referencia órajel egy „nulla idő” referenciát biztosít, és kiadható a 110 és 111 lokális órajelek több periódusához a 108 PLL többszörösségétől függően. A 113 sín órajelet a 106 referencia órajelből származtatjuk oly módon, hogy a 110 lokális órajelet elosztjuk egy előre meghatározott N egésszel egy 112 osztóban. A 102 chiptől a 104 chiphez küldendő adatot eltároljuk az elosztott 111 lokális órajel egy előre meghatározott ágán, és tovább vezetjük a 116 adat vonalhoz a 118 vezetón keresztül. Az adatot fogadjuk a 120 fogadónál (RX) és elfogjuk egy 122 cél tároló eszközbe (latch) a 104 chipben lévő elosztott 110 lokális órajel egy előre meghatározott ágán. A 102 chip és a 104 chip fizikai elkülönülése következtében az adat a 122 cél tároló eszköz 124 bemeneténél az időben késleltetve jelenik meg. (A 120 RX hozzájárulása a késéshez tipikusan kicsi az adatátvitel miatti késleltetéshez képest.) Az időbeli késleltetésre késésként hivatkozunk, és a későbbiekben fogjuk tovább tárgyalni az 1B ábrával kapcsolatban.

Hasonlóan, a 104 chip adatot küld a 102 chipnek a 126 adat vonalon keresztül. A 104 chiptől küldendő adatot eltároljuk a 128 tároló eszközben a 130 osztóból (amely elosztja a 111 lokális órajelet N-nel) származó kimeneti jel egy előre meghatározott ágán. Az adatot a 126 adat vonalra vezetjük a



132 vezetón keresztül, és a 134 cél tároló eszközön fogjuk el a 136 fogadón keresztül. A 102 chip bemeneteként szolgáló adatot a 134 adat tároló eszközbe a 130 osztó (amely szintén elosztja a 110 lokális órajelet N -nel) egy kimenetének egy előre meghatározott ágán fogjuk el.

Az 1B ábrán bemutatunk egy példa időzítési diagramot az 1A ábra 100 interfészéhez, az eddig használatos megközelítéssel összhangban. A 102 chiptől a 104 chiphez küldött 115 adatot a 114 tároló eszközben a 113 sín órajel egy t_1 emelkedő ágán tároljuk el. A 113 sín órajelet oly módon generáljuk, hogy elosztjuk a 110 lokális órajelet N -nel a 112 és 130 osztókban a 102 chipben. Egy késés általi T_1 késleltetést követően a 117 adat megjelenik a 122 cél tároló eszköz egy bemenetén, majd eltároljuk a 123 sín órajel t_2 emelkedő ágán. A 123 sín órajelet úgy generáljuk, hogy elosztjuk a 111 lokális órajelet N -nel a 112 és 130 osztókban a 104 chipben. Így az eddig használatos megközelítésben (összhangban az 1B ábrával) a 125 adat a 102 chipből való kibocsátását követő egy sín ciklussal később jelenik meg a 104 chipben. Az 1B ábrán az eltérés nulla a 113 sín órajel és a 123 sín órajel között.

Ha az 1A ábrán a 100 interfészben a sín órajel sebességet megnöveljük, a késés meghaladhatja az egy sín órajel ciklust. Ez az 1C ábrán ábrázolt példa időzítési diagramot eredményezheti. Csakúgy, mint az előbb, a 115 adatot is a 113 sín órajel t_1 ágán tároljuk el. A 117 adat a 122 cél tároló eszköz 124 bemenetén T_1 késési idő után jelenik meg, amely hosszabb, mint a 113 sín órajel és a 123 sín órajel periódusa. A 117 adatot a 123 sín órajel t_3 ágán tároljuk el a 104 chipben a 125 adat biztosítására a 104 chipen. Ha a 102 és a 104 chipek közötti

100 interfész képviseli valamely adatfeldolgozó rendszeren belül a 102 chip és más chipek sokasága közötti interfészek sokasága közül a leghosszabb késéssel rendelkező interfészt, akkor az 1C ábrán bemutatott kétciklusos késés képviseli a „cél” ciklust az adat chipek (például a 102 chip és a 104 chip) közötti átviteléhez és elfogásához. A cél ciklus az az előre meghatározott ciklus, amely mellett a chip adatot vár. A rövidebb késéssel rendelkező interfészeket tompítanunk kell az eddig használatos megközelítéssel összhangban annak érdekében, hogy biztosítsuk a szinkronizált működést. A tompítás biztosítja, hogy a gyorsabb útvonalak a 100 interfészben nagyobb késésekkel rendelkezzenek, mint egy sín órajel ciklus és kisebbekkel, mint két sín órajel ciklus, ami által az adat szinkronizáció fenntartható.

Ezt jobban megérthetjük, ha most az 1D ábrára utalunk, amely 101 chipek sokaságát, a 102, 103 és 104 chipeket ábrázolja. A 102 chip és a 104 chip egy T_s hosszú késéssel rendelkező 152 „lassú” útvonalon van összekötve. A 103 chip a 102 chiphez egy T_F rövid késéssel rendelkező 154 „gyors” útvonalon keresztül van hozzákötve. A 101 chipek sokaságát a 102-105 chipekhez egy T_M késéssel rendelkező „névleges” útvonal kapcsolja össze, amely késés olyan, mint a 102 chip és a 105 chip közötti 156 útvonalon lévő késés.

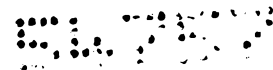
Az 1E ábrán lévő időzítési diagram további részletekkel szolgál. Az 1E ábra az 1C ábrához hasonló időzítési diagramot ábrázol, amelyen a cél ciklus az adatnak egy fogadó chipbe való elfogásához kettő sín ciklus. Az 1E ábrán a T_M névleges késést 1,5 sín ciklusként mutatjuk, a T_F gyors útvonali késést valamivel nagyobbaként ábrázoljuk, mint egy sín ciklus, és a T_s



lassú útvonali késést valamivel kisebbnek mutatjuk, mint két sín ciklus. Ebben az esetben az 1D ábrán lévő 101 chipek sokaságának mindegyike a cél cikluson fogja el az adatot, két sín ciklussal az adat kibocsátása után.

Ha viszont a gyors útvonali késéssel ábrázolt gyors útvonal rövidebb, az adat szinkronizáció elvész. Ebben az esetben az adat hamarabb érkezik meg a 103 chiphez, mint a 103 chip sín órajelének t_2 átmenete, amint azt a 103 chipnél lévő 117 adat pontozott részlete ábrázolja, és eltároljuk a 103 chipbe egy sín ciklus után. Ezt a 125 adat pontozott részletével ábrázoljuk a 103 chipben. Annak érdekében, hogy helyreállítsuk a szinkronizációt, a 102 és 103 chipek közötti 154 gyors útvonalat tompítanunk kellene a gyors útvonali késés növeléséhez, T_s -ről T_F -re. Ennek megfelelően egy ilyen eddig használatos interfész időzítését egy specifikus működési tartományra (egy bizonyos interfész hosszúságra) hangoljuk, és ez csak arra a technológiára érvényes, amelyre a tervezés időzítve és elemezve volt.

Hasonlóan, az 1D ábrán lévő chipek órajel sebességének növelése a szinkronizáció elvesztését eredményezi. Ezt megérthetjük, ha egy konkrét példát tekintünk. A lokális órajel ciklus időt 1 nanoszekundummal (ns) rendelkezőnek vesszük. A sín órajel egy olyan periódussal fog rendelkezni, amely egy rögzített többszöröse (ezt kettőnek fogjuk venni) a lokális órajelnek. Legyen az interfész T_M névleges késése $3ns \pm 99ns$ -os időzítési változással, azaz a legjobb eset, vagyis a T_F gyors útvonal $2ns$ -os és a legrosszabb eset, vagyis a T_s lassú útvonal $4ns$ -os. Az adat két ns után és négy ns előtt fog megérkezni. Ily módon az interfész minden körülmény között működni fog,



vagyis az adat garantáltan megérkezik az első sín ciklus után és a második sín ciklus előtt. Viszont ha a chipek sebességét, 9ns-os ciklus időre növeljük, a sín ciklus idő 1,8ns-ra változik. Annak érdekében, hogy elegendő időt biztosítsunk az adatnak az interfészen való keresztülhaladáshoz rosszabb eseti körülmények között, az adatot nem szabad elfognunk 2,5 sín ciklus, vagyis 4,5ns előtt, mert a kettő sín ciklus kevesebb, mint a T_s lassú útvonal idő, vagyis 4ns. Ezután, annak érdekében, hogy egy 1,8ns-os sín ciklust működtessünk, a leggyorsabban az adat $1,5 \cdot 1,8 = 2,7ns$ alatt érkezhethet meg (egy sín ciklussal korábban) azért, hogy biztosítsuk, hogy az adat ugyanazon a cikluson érkezzon meg minden körülmény között. Viszont a fenti késési számokból következik, hogy a legkorábban az adat a gyors útvonalon érkezhethet meg $3ns - ,99ns = 2,01ns$ -os T_F mellett. Ezért egy 1,8ns-os sín ciklus idő melletti működést nem támogathatunk egy hagyományos szinkronizált elrendezésben. A szinkronizált működés érdekében a sín : processzor arányt le kell csökkentenünk legalább 3 : 1-re és 2,71ns ciklus idő mellett kell működtetnünk ($2,7ns \cdot 1,5$ ciklus = 4,05ns és $2,7ns \cdot ,5$ ciklus = 1,35ns), ami viszont a lokális órajel sebesség növelése ellen dolgozik.

Ezért a tárgykörben szükség van berendezésre és eljárásokra a növekvő órajel sebességekkel rendelkező adatfeldolgozó rendszerben lévő chipek közötti adatátvitelhez való alkalmazkodáshoz. Különösen szükség van eljárásokra és berendezésre az olyan adatfeldolgozó rendszerekben lévő chipek közötti adat szinkronizáció biztosítására, amelyekben az útvonali késések több mint egy sín ciklus fölött változnak, és amelyekben meg-



szűnik a speciális hardver tompítás tervezése iránti szükségesség.

A jelen találmány célja olyan technika biztosítása, amely kiküszöböli a fenti hátrányokat.

A jelen találmánnyal összhangban interfész berendezést biztosítunk, amely magában foglal:

egy első tároló eszközt adat értékek egy első halmazának tárolására;

egy második tároló eszközt adat értékek egy második halmazának tárolására;

az első és második tároló eszközhöz csatlakoztatott áramkört az első tároló eszközből származó első adat érték és a második tároló eszközből származó második adat érték szekvenciális kimenetre tételére legalább egy vezérlő jel hatására, ahol az első és második tároló eszközök egy első órajel előre meghatározott számú ciklusáig tartanak meg adat értékeket, valamint első és második kiválasztási áramkört, amelyek mindegyike az első és második tároló eszköz megfelelő bemeneteihez ebben a sorrendben csatlakoztatott kimenettel rendelkezik, az első és második kiválasztási áramkörök mindegyike rendelkezik első bemenettel adat folyam fogadására és az első és második tároló eszközök megfelelő kimenetéhez csatlakoztatott második bemenetet, ahol az első és második kiválasztási áramkör kiválaszt az első és második bemenetek egyikén lévő jelet kimenetre tételre első és második vezérlő jel hatására.

A jelen találmánnyal összhangban biztosítunk továbbá eljárást integrált áramköri eszközök interfészelésére, amely eljárás magában foglalja a következő lépéseket:



az adat értékek első halmazát az első tároló eszközben tároljuk, ahol az első halmaz minden egyes adat értékét egy első órajel előre meghatározott számú ciklusáig tároljuk;

az adat értékek második halmazát a második tároló eszközben tároljuk, ahol a második halmaz minden egyes adat értékét egy első órajel előre meghatározott számú ciklusáig tároljuk;

szekvenciálisan kimenetre teszünk egy első adat értéket az első tároló eszközből és egy második adat értéket a második tároló eszközből úgy, hogy egy áramkört csatlakoztatunk az első és a második tároló eszközhöz és egy vezérlő jelet biztosítunk az áramkörhöz, ahol a vezérlő jel olyan periódussal rendelkezik, amely az első órajel periódusának egy előre meghatározott többszöröse;

egy adat folyamat fogadunk első és második kiválasztási áramkör mindegyikének első bemeneténél;

az első és második tároló eszközök kimeneti jeleit a vonatkozó első és második kiválasztási áramkör mindegyikének második bemenetéhez biztosítjuk;

az adat értékek első halmazának mindegyikét kimenetre tesszük az első tároló eszközhöz az első kiválasztási áramkörtől egy első kiválasztási jel hatására;

valamint az adat értékek második halmazának mindegyikét kimenetre tesszük a második tároló eszközhöz a második kiválasztási áramkörtől egy második kiválasztási jel hatására.

Az előzőkben leírtak meglehetősen tágan körvonalazták a jelen találmány jellemzőit és technikai előnyeit annak érdekében, hogy a találmány következő részletes leírását jobban megérthessük. Ezután a találmány további jellemzőit és előnyeit



mutatjuk be, amelyek a találmány igénypontjainak tárgyát képezik.

A jelen találmánynak és előnyeinek teljesebb megértése érdekében a kísérő rajzokkal kapcsolatban tekintett következő leírásokra hivatkozunk, amely rajzokon:

Az 1A ábra egy chip interfészt ábrázol, összhangban az eddig használatos megközelítéssel;

Az 1B ábra sematikusan ábrázol egy időzítési diagramot az 1A ábra chip interfészének egy kiviteli példájához, összhangban az eddig használatos megközelítéssel;

Az 1C ábra egy időzítési diagramot ábrázol az 1A ábra chip interfészének egy másik kiviteli példájához, összhangban az eddig használatos megközelítéssel;

Az 1D ábra összekapcsolt chip sokaságát ábrázolja egy adatfeldolgozó rendszerben;

Az 1E ábra sematikusan ábrázol egy időzítési diagramot az 1D ábra összekapcsolt chipjeinek sokasága egy kiviteli példájához;

A 2. ábra blokk diagram formában ábrázol egy reprezentatív hardver környezetet a találmány gyakorlatba való átültetéséhez;

A 3. ábra blokk diagram formában ábrázol egy chip interfészt, összhangban a jelen találmány egy kiviteli példájával;

A 4A ábra blokk diagram formában ábrázol egy rugalmas interfészt, összhangban a jelen találmány egy kiviteli példájával;

A 4B ábra sematikusan ábrázolja a jelen találmány 3A ábráján bemutatott kiviteli példájának egy időzítési diagramját;



Az 5. ábra egy chip interfész egy alternatív kiviteli példáját ábrázolja, összhangban a jelen találmánnyal;

A 6A ábra egy rugalmas interfész egy másik alternatív kiviteli példáját ábrázolja, összhangban a jelen találmánnyal;

A 6B ábra sematikusán ábrázolja az 5A ábra rugalmas interfészének egy időzítési diagramját;

A 7A ábra egy rugalmas interfész egy másik alternatív kiviteli példáját ábrázolja, összhangban a jelen találmánnyal;

A 7B ábra sematikusán ábrázol egy időzítési diagramot a 7A ábra kiviteli példája számára;

A 8A ábra blokk diagram formában ábrázolja a rugalmas interfész egy további alternatív kiviteli példáját, összhangban a jelen találmánnyal; és végül

A 8B ábra sematikusán ábrázol egy időzítési diagramot a 8A ábra kiviteli példája számára.

A jelen találmány rugalmas interfész mechanizmust biztosít, amely adat szinkronizációt valósít meg adatfeldolgozó chipek valamely sokasága között egy adatfeldolgozó rendszerben. Az adat szinkronizációt anélkül érjük el, hogy szükség lenne tompításra, ami máskülönben komplikálja a fizikai huzalozást, és megnöveli a hardver tervezés komplexitását. Az interfész „rugalmassága” ellensúlyozza a rendszerben lévő adatfeldolgozó chipeket összekapcsoló útvonalak közötti fizikai különbségeket. A fogadott adat tároló elemek sokaságába való elfogása és az eltárolt adat szelektív módon való irányítása által biztosítjuk az adat szinkronizációt egy olyan adatfeldolgozó rendszerben, amely egy sín órajel ciklusnál több váltakozású késésekkel rendelkezik. A szinkronizációt dinamikusan építhetjük ki, egy inicializációs elrendezési folyamat



(initialisation alignment procedure) végrehajtásával a bekapcsoláskor, vagy egy újraindítást követően. Ily módon az adatok szinkronizációját elérhetjük a jelen találmány alapelveivel összhangban anélkül, hogy szükséges lenne a lapka elrendezés (board design) időzítési elemzése és a gyors útvonal tompítása.

A következő leírásban számos olyan specifikus részletet ismertetünk, mint sín órajel frekvenciák és szinkronizációs ciklusok, órajel ágak stb. a jelen találmány átfogó megértésének biztosítására. Viszont nyilvánvaló lesz a tárgykörben járatosak számára, hogy a jelen találmány átültethető a gyakorlatba ezen specifikus részletek nélkül is. Más esetekben pedig a jól ismert áramköröket blokk diagram formában mutatjuk annak érdekében, hogy ne zavarjuk össze a jelen találmányt felesleges részletességgel.

Hivatkozunk most a 2-9. ábrákra, ahol az ábrázolt elemeket nem feltétlenül méretarányosan ábrázoljuk, és ahol az azonos vagy hasonló elemeket ugyanazzal a hivatkozási jellel jelöljük néhány részleten végig.

A 2. ábrán a jelen találmánynak a gyakorlatba való átültetéséhez ábrázolunk egy reprezentatív hardver környezetet, amely a 213 adatfeldolgozó rendszer egy tipikus hardver konfigurációját ábrázolja, összhangban a szóban forgó találmánnyal, ahol a rendszer rendelkezik 210 központi feldolgozó egységgel (CPU), például egy hagyományos mikroprocesszorral, és számos más egységgel, amelyeket 212 rendszer sín kapcsol össze. A 213 adatfeldolgozó rendszer tartalmaz 214 közvetlen hozzáférésű memóriát (RAM - random access memory), 216 csak olvasható memóriát (ROM - read-only memory), és 218 bemeneti/kimeneti (I/O



- input/output) csatolót a perifériális eszközöknek, például 220 lemez egységeknek és 240 szalagos meghajtóknak a 212 sínhez való csatlakoztatásához, 222 felhasználói interfész csatolót a 224 billentyűzetnek, a 226 egérnek és/vagy más felhasználói interfész eszközöknek, például érintőképernyős eszköznek (nem mutatjuk) a 212 sínhez való csatlakoztatásához, 234 kommunikációs csatolót a 213 munkaállomásnak egy adatfeldolgozó hálózathoz való kapcsolásához, és 236 megjelenítő csatolót a 212 sínnek a 238 megjelenítő eszközhöz való csatlakoztatásához. A 210 CPU tartalmazhat más, itt nem mutatott áramköröket, amik magukban foglalnak egy mikroprocesszorban általában megtalálható áramköröket, például végrehajtási egységet, sín interfész egységet, aritmetikai-logikai egységet stb. A jelen találmány interfészét magában foglalhatja a 210 CPU. Továbbá, a jelen találmányt beágyazhatjuk tároló eszközökbe, például 214 RAM-ba (ami magában foglalhat itt nem mutatott memória vezérlő áramkört). A 210 CPU elhelyezkedhet egyetlen integrált áramkörön is.

Most a 3. ábrára hivatkozunk, amelyen bemutatunk egy 300 interfészt összhangban a jelen találmánnyal. A 300 interfészt magában foglalja a 302 chip és a 304 chip mindkettője, amelyek egymással adatot egy vonatkozó 300 interfészen keresztül kommunikálnak. (Habár a jelen találmány kiviteli példáját chip - chip interfész szöveggörnyezetében mutatjuk be, a jelen találmány alapelveit megvalósíthatjuk bármely órajeles tároló eszközök egy párja között.) Az adatot a 302 és 304 chipek között egy sín órajel (306 és 308 sín órajelek) által meghatározott sebesség mellett visszük át. A 306 és 308 sín órajelek névlegesen ugyanazon frekvenciájúak, és egy 312 PLL-hez biztosított



310 referencia órajelből származtatjuk őket a 302 és 304 chippek mindegyikében. A jelen találmány egy kiviteli példájában a 310 referencia órajel lehet egy rendszer órajel. A 312 PLL-ek mindegyike kimenetre tesz egy lokális órajelet (314 lokális órajelet a 302 chipben és 316 lokális órajelet a 304 chipben), ami fázis-igazított a 310 referencia órajelhez, és a 310 referencia órajel periódusának egy előre kiválasztott M egész többszöröse lehet. A 314 lokális órajelet a 318 vezető elpuffereli a 302 chiptől származó 306 sín órajel biztosítására. Hasonlóan a 316 lokális órajelet is a 320 vezető elpuffereli a 304 chiptől származó 308 sín órajel biztosítására.

A sín órajelet a chiptől kommunikált adattal együtt biztosítjuk. A 302 chiptől származó 322 adatot eltároljuk a 324 kimeneti tároló eszközbe, és a 326 vezető fogja vezetni és puffereelni. Az adatot a 314 lokális órajel egy előre kiválasztott ágán tároljuk el. Az adatot a 328 multiplexeren (MUX) keresztül fogadjuk. A 328 MUX fogad ezen kívül egy előre meghatározott szinkronizációs mintát az elrendezési folyamattal kapcsolatosan. Ezeket a későbbiekben mutatjuk be alább.

A 322 adatot elpuffereljük a 330 fogadó (RX) által, és egy 332 rugalmas interfész egységhez biztosítjuk. A 322 adattal együtt küldött 306 sín órajelet a 334 RX elpuffereli, amely kimenete alkotja a 336 I/O órajelet, és amit szintén biztosítunk a 332 rugalmas interfészhez. A 304 chiptől a 302 chiphez küldött adatot (a 308 sín órajellel együtt) hasonlóan fogadja a 300 interfész a 302 chipben, és érthető módon a 332 rugalmas eszköz leírása ugyanolyan jól alkalmazható a 304 chiptől származó adat 302 chip általi fogadása esetén is.



A 339 cél ciklus egység beállítja a cél ciklust, amelyen a lokális órajel az adatot eltárolja a fogadó chipben, például a 316 lokális órajel a 304 chipben. A cél ciklust részletesen a 4A/B ábrákkal kapcsolatban fogjuk tárgyalni, amelyek egy kettő rugalmassággal rendelkező 332 interfészt ábrázolnak. Egy E rugalmassággal rendelkező interfész esetén a cél ciklus egység magában foglalhat egy E-vel osztó áramkört. A 339 cél ciklus egység magában foglalhat továbbá egy programozó regisztert az előre meghatározott cél ciklus érték tárolására, amelyet a 341 cél programon keresztül tölthetünk be. A 302 chipben lévő 339 cél ciklus egységben beprogramozott cél ciklus különböző lehet a 304 chipben lévő 339 cél ciklus egységben beprogramozott cél ciklustól. A 339 cél ciklus egység a kimenetére teszi a 343 kiválasztás vezérlőt (amely jelek sokaságát foglalhatja magában), a 332 interfész egység kiviteli példájától és a megfelelő E rugalmasságtól függően. A 343 kiválasztás vezérlőt a későbbiekben tovább tárgyaljuk a 4A-8B ábrákkal kapcsolatban, amelyeken a 332 interfész egység kiviteli példáit ábrázoljuk.

Most a 4A ábrára hivatkozunk, amely egy 332 rugalmas interfész egység egy kiviteli példáját ábrázolja összhangban a jelen találmánnyal. A 332 egység magában foglal egy 404 bemenettel rendelkező 402 MUX-ot, amely adatot fogad a 330 RX-től. A 402 MUX 406 kimenetét csatlakoztatjuk a 408 tároló eszköz (D) adat bemenetéhez. A 408 tároló eszközt a 336 I/O órajel által órajelezzük. A 408 tároló eszköz adatot tárol el a D bemeneténél a 336 órajel emelkedő ágán, és megtartja az adatot a 336 órajel egy következő emelkedő ágáig. A 408 tároló eszköz 410 kimenetét visszacsatlakoztatjuk egy második bemenethez, a 402 MUX 412 bemenetéhez. A 402 MUX kiválaszt a 404 bemenet és

a 412 bemenet közül egyet a 406 kimenetre való kitételhez, a 414 kapu hatására.

A 414 kaput a 306 sín órajelből származtatjuk, és a 306 sín órajel periódusának kétszeresével rendelkezik. A 414 kaput generálhatjuk egy késleltetés igazító hurok (DLL - delay lock loop) használatával. A jelen találmányban használható DLL egy kiviteli példáját írja le a közösen birtokolt, együtt függő (co-pending) „Dynamic Wave Pipelined Interface Apparatus and Method Therefor” című alkalmazás, amelyet itt hivatkozásként alkalmazunk. A 414 kapu fázisát az alább tárgyalt inicializációs elrendezési folyamat közben állítjuk be, és a 414 kapu működését a 4B ábrával kapcsolatban tárgyaljuk tovább.

A 330 RX-ből származó adatot szintén párhuzamosan tápláljuk be egy második MUX-ba (416 MUX) a 418 bemeneten. A 416 MUX 420 kimenetét egy második tároló eszköz (422 tároló eszköz) D bemenetéhez csatlakoztatjuk, amelyet szintén a 336 I/O órajel órajelez, és amely adatot tárol el a 336 I/O órajel egy emelkedő ágán, majd megtartja az adatot az órajel egy következő emelkedő ágáig. A 422 tároló eszköz 424 kimenetét a 416 MUX egy második bemenetéhez (a 426 bemenethez) csatlakoztatjuk.

A 416 MUX választ a 418 bemenet és a 426 bemenet között, a 414 kapu 428 \kapu komplementének hatására. Így, amikor a 402 és 416 MUX-ok egyike kiválasztja a 330 RX-től fogadott adatot, a másik kiválasztja a megfelelő tároló eszközében (a 408 és 422 tároló eszközök egyikében) tartott adatot. Ily módon a 408 és 422 tároló eszközök egyikében megelőzően tárolt bitet a 336 I/O órajel egy további ciklusáig tartjuk meg.



Ily módon két adat folyamat hozunk létre, amelyek mindegyike a 336 I/O órajel két periódusáig érvényes. A 414 kapu és a 428 \kapu közötti fázis megfordítás miatt a két adat folyamat egymástól egy adat érték ideiglenes szélességű eltolásra van egymástól, vagyis a 336 I/O órajel egy ciklusára.

Ezt jobban megérthetjük, ha a 4B ábrára hivatkozunk, amely egy időzítési diagramot ábrázol a 4A ábrán lévő 332 rugalmas interfész egységgel összhangban. Amint azt előzőleg leírtuk, a 324 kimeneti tároló eszközben tartott 325 adatot a 314 lokális órajellel szinkronban bocsátjuk ki a 202 chipben. A 204 chipben a 330 RX-nél fogadott 322 adat késleltetett a 202 és 204 chipek közötti útvonal által képviselt késés miatt, amint azt feljebb az 1D ábrával kapcsolatban tárgyaltuk. A 336 I/O órajel t_1 emelkedő ágán a 322 adat „a” adat értékét elfogjuk a 408 tároló eszköz által a 402 MUX 406 kimenetén keresztül. Mivel a 414 kapu ki van adva, vagy „nyitott”, a 330 RX-től származó adat a 404 bemenetnél ezáltal kiválasztódik kimenetre tételre a 402 MUX által. (Egy kaput nyitottnak fogunk nevezni, amikor a megfelelő MUX a bejövő adat folyamat fogadó inputot választja ki. Bár ehhez a „magas” logikai állapotot társítjuk a 4. ábra kiviteli példájában, érthető módon egy alternatív kiviteli példa lesz az is, amelyben egy nyitott kapunak egy „alacsony” logikai szintet feleltetünk meg.)

Megfordítva, a 428 \kaput negáljuk. Ennek hatására a 416 MUX kiválaszt egy megelőző, a 422 tároló eszközben tartott adat értéket, amelyet visszacsatlakoztatunk a 422 tároló eszköz D bemenetéhez a 416 MUX-on keresztül. Így a 422 tároló eszközben tartott adat értéket visszatartjuk a 336 I/O órajel egy további periódusáig, amely órajel biztosítja az óraütés

jelet a 408 és 422 tároló eszközök mindkettője számára, amint azt a 4A ábrával kapcsolatban leírtuk.

A 336 I/O órajelet a 306 sín órajelből kapjuk meg, amint azt a 3. ábrán mutatjuk. Feltesszük, hogy a kibocsátásnál a 306 sín órajel egy adat érvényességi ablak közepén helyezkedik el, amint azt a 4B ábrán mutatjuk. A sín órajel középre helyezését a közösen birtokolt, együtt függő „Dynamic Wave-Pipelined Interface and Method Therefor” című alkalmazás írja le, amelyet itt hivatkozásként alkalmazunk. A 306 sín órajel késleltetést szenved el az interfészen keresztül, csakúgy, mint az adat. A 306 sín órajelbeli késés a 304 chipnél T_1 -gyel összehasonlítható lehet, és ez tükröződik is a 336 I/O órajelben, amelyet ezáltal a 322 adathoz képest helyezünk középre.

A 414 kaput oly módon generáljuk, hogy a 414 kapu ágai fázis-koherensek legyenek a 336 I/O órajel leeső ágaival. A 336 I/O órajel t_2 ágánál a 414 kapu leesik (t_3 ág). Ennek hatására a 402 MUX kiválasztja a 408 tároló eszköz 410 kimenetét (melyet a 402 MUX 412 bemenetéhez csatlakoztatunk) kimenetre tételhez a 406 kimenetnél. Amint a 414 kaput negáljuk, 428 \kaput adunk ki, ami által a 416 MUX kiválasztja a 420 kimeneten való kimenetre tételhez a 330 RX-től származó adatot a 418 bemeneten. Ezt a 422 tároló eszköz D bemenetéhez csatlakoztatjuk. A 330 RX-től fogadott adat most megfelel a 322 adat „b” adat értékének.

A 336 I/O órajel t_4 ágánál a 408 és 422 tároló eszközök eltárolják az adatot a vonatkozó D bemeneteiknél. A 408 tároló eszközben ez megfelel a megelőzően tartott adat értéknek (a 322 adat a értékének), amelyet aztán megtart a 316 lokális órajel egy további periódusáig. A 422 tároló eszköz a b adat

értéket tárolja el a D bemenetén a 416 MUX-on keresztül a 424 kimenetre.

A 336 I/O órajel következő t_7 átmeneténél a „c” adat értéket fogjuk el. Mivel a t_7 ág nál a 414 kapu ki van adva, a 322 adatnál lévő c adat érték a 402 MUX 406 kimenetén jelenik meg. A b adat értéket visszatartjuk a 422 tároló eszközben, mert a 428 \kapu negált, lévén a 414 kapu komplemente. Mivel az adatok valamely folyamja folyamatosan érkezik a 322 adaton, a 332 rugalmas eszköz ily módon folytatja két adat folyam generálását a 408 és 422 tároló eszközök 410 és 424 kimenetein, ebben a sorrendben. A két adat folyam a 322 adaton érkező bemeneti adat folyam váltakozó részleteit tartalmazza, amelyek a 316 lokális órajel két periódusáig érvényesek, amint azt a 4B ábrán ábrázoljuk.

A bemeneti adat folyam struktúráját úgy állítjuk vissza, hogy felváltva választunk értékeket a két adat folyam valamelyikéről, szinkronban a 316 lokális órajellel. Kiválasztunk egy lokális órajel cél ciklust. A lokális órajel cél ciklus az a ciklus, amely mellett kell az adatokat elfognunk a cél tároló eszközbe a lokális órajel által, például a 3. és 4. ábrákon lévő 316 lokális órajel által. A cél ciklust elemzés által határozzuk meg. A cél ciklusnak időben későbbinek kell lennie, mint a legrosszabb eseti késés az interfészen keresztül. Például, a 4B ábrán bemutatott kiviteli példában a cél ciklust a 316 lokális órajel három periódusára állítottuk be, ami a t_8 ág nak felel meg. A cél ciklust a 310 referencia órajel által meghatározott nulla idő referenciától mérjük, mint ahogyan azt megelőzőleg leírtuk az 1A ábrán lévő 106 referencia órajellel kapcsolatosan. A 4A/B ábra kettő rugalmasságú kiviteli példá-

jában az adat érkezése akár két órajel ciklussal korábban is lehet, mint a cél ciklus. Egy ilyen kiviteli példában a 3. ábrán lévő 339 cél ciklus egység magában foglal egy kettővel osztó áramkört.

A cél ciklus beállításával az adatot kiválasztottuk elfogásra a 430 cél tároló eszközbe, a 316 lokális órajel hatására, a 432 MUX-on keresztül. A 432 MUX egy 434 és 436 bemenetpárral rendelkezik. A 434 bemenetet a 408 tároló eszköz 410 kimenetéhez csatlakoztatjuk, és a 436 bemenetet a 422 tároló eszköz 424 kimenetéhez csatlakoztatjuk. A 432 MUX kiválasztja a 408 és 422 tároló eszközök kimenete által képviselt két adatfolyam egyikét kimenetre tételre, a 343 kiválasztás vezérlő hatására. Amikor a 343 kiválasztás vezérlő egy első logikai állapottal rendelkezik („magas” a 4B ábra kiviteli példájában), a 432 MUX a 434 bemenetnél lévő adatot biztosítja a 430 tároló eszköz D bemenetéhez, és amikor a 343 kiválasztás vezérlő egy második előre meghatározott logikai állapottal rendelkezik („alacsony” a 4B ábra kiviteli példájában), a 432 MUX a 436 bemeneténél lévő adatot biztosítja a 430 tároló eszköz D bemenetéhez. A 343 kiválasztás vezérlő olyan periódussal rendelkezik, amely a kétszerese a 316 lokális órajel periódusának, és fázis-szinkronizált a 316 lokális órajellel oly módon, hogy a 343 kiválasztás vezérlő az első logikai állapottal rendelkezik, és a cél cikluson van középre helyezve. Ily módon a 4B ábrán a 316 lokális órajel t_8 ágánál a 410 kimenetnél lévő a adat értéket a 432 MUX-on keresztül a 430 tároló eszköz D bemenetéhez csatlakoztatjuk, és a 316 lokális órajel t_8 ága által tároljuk el. A 316 lokális órajel következő t_9 pozitív ágánál a továbbított adat folyamata következő részletét tároljuk el a 430



cél tároló eszközbe. A 343 kiválasztás vezérlő a fázisában egy fél periódust haladt előre, és ezért a második logikai állapottal rendelkezik, ami által a 422 tároló eszköz 424 kimenetét csatlakoztatjuk a 430 tároló eszköz D bemenetéhez a 432 MUX-on keresztül. A t_9 ágnál, a 422 tároló eszköz 424 kimeneténél lévő b adat értéket eltároljuk a 430 tároló eszközbe, és a b adat érték megjelenik a 338 adat kimeneten. A 316 lokális órajel következő ciklusaiban a 332 rugalmas interfész visszaállítja a 322 adat folyamat azáltal, hogy felváltva választ a 408 tároló eszköz 410 kimenete és a 422 tároló eszköz 424 kimenete között.

Mielőtt adatot vihetnénk át a 332 rugalmas interfészen keresztül, a 414 kaput (és ezzel együtt a komplement, a 428 \kaput is) inicializálnunk kell. Mivel az interfészen keresztüli késés több mint egy lokális órajel periódussal változhat, a 414 kaput a megfelelő fázissal kell inicializálnunk. Amint azt megelőzően leírtuk, a 414 kapu periódusa kétszerese a 336 I/O órajel és a 316 lokális órajel periódusának. A 414 kapu fázis-szinkronizált a 336 I/O órajellel oly módon, hogy a 414 kapu legfelső lapos (flat) részleteit a 336 I/O órajel előre kiválasztott ágain középen helyezük el (a 4B ábra kiviteli példájában a pozitív ágakon).

A 414 kaput a jelen találmány egy kiviteli példájában inicializálhatjuk oly módon, hogy egy szinkronizációs (sync) mintát továbbítunk a bekapcsoláskor vagy az újraindításkor. Újra a 3. ábrára hivatkozva, az adatfeldolgozó rendszer újraindításának vagy bekapcsolásának hatására 340 inicializációs elrendezési folyamat (IAP - initialisation alignment procedure) mód jelet adunk ki, ami által a 328 MUX kiválaszt



kimenetre tételre egy előre meghatározott sync mintát. A 340 IAP mód jelet kiadhatja a jelen találmány egy kiviteli példájában a 210 CPU, a 216 ROM által tartalmazott BIOS (Basic Input / Output System) utasítások hatására.

A 332 rugalmas interfész egység 4A ábrán ábrázolt kiviteli példájában az adat érkezik egy két ciklus nagyságú ablakban anélkül, hogy szinkronizációs problémákat okozna, amint azt megelőzően leírtuk. A 332 rugalmas interfész egységre a 4A ábrán úgy hivatkozunk, mint kettő rugalmassággal rendelkezőre. Az IAP-ben a szinkronizációs mintától megköveteljük, hogy periodikus legyen egy olyan P periódus szerint, amely egyenlő vagy nagyobb, mint a rugalmas eszköz rugalmassága. Így a 4B ábra 332 rugalmas interfész egysége számára egy alkalmas szinkronizációs minta lenne egy „1” és „0” között váltakozó bitek valamely „1” kezdeti bittel rendelkező sokasága. A kettőtől különböző rugalmassággal rendelkező rugalmas interfészek kiviteli példáit alább írjuk le, és a szinkronizációs mintát ennek megfelelően módosítjuk. Például egy N rugalmassággal rendelkező rugalmas interfész esetén (amelyet alább fogunk tárgyalni a 8A/B ábrákkal kapcsolatban) egy alkalmas minta lenne egy kezdeti „1” bit, amelyet N-1 „0” bit követ, és amely minta azután ismétlődik. Más bit minták szintén használhatóak. Például a fenti példa mintákhoz képest komplementer bit minták alternatívaként használhatóak.

A 414 kapu fázisának beállításakor mintát vételezhetünk a 408 tároló eszköz 410 kimenetéből, miközben a sync mintát küldjük. A 414 kaput egy előre kiválasztott fázissal inicializáljuk. Ha a kettő rugalmassággal rendelkező 332 rugalmas interfész egység számára fentebb leírt sync minta esetén egy

„1”-est detektálunk a 410 kimenetnél, akkor a 414 kapu fázisa, és ezzel együtt a 428 \kapu fázisa is megfelelő. Máskülönben a 414 kapu fázisát, és ennek megfelelően a 428 \kapu fázisát is a 336 I/O órajel periódusának a felével el kell tolnunk. Amint azt megelőzően tárgyaltuk, a 414 kaput generálhatjuk egy DLL használatával, amely DLL kiviteli példáját leírja az együtt függő, közösen kiadott, fentebb hivatkozott „Dynamic Wave Pipelined Interface Apparatus and Method Therefor” című alkalmazás, amelyet itt hivatkozásként alkalmazunk.

A jelen találmány alapelveivel összhangban lévő rugalmas interfész, például a 300 rugalmas interfész működését jobban megértjük, ha most az 5. ábrára hivatkozunk. Az 5. ábrán a 302 chip adatot cserél a 304 chippel, úgy, mint a 3. ábrán, és ezen kívül a 306 chippel is. A 302 chip a 300 interfészből kettőt tartalmaz, amelyek egyike a 302 chipet a 304 chiphez csatlakoztatja a 304 chip által tartalmazott 300 interfészen keresztül, és a második a 302 chipet csatlakoztatja a 306 chiphez, amely szintén tartalmaz egy 300 interfészt, a jelen találmány egy kiviteli példájával összhangban. A 302 és 304 chipeket csatlakoztató 502 útvonal lehet egy gyors útvonal, amely hasonló az 1D ábrán lévő T_F késéssel rendelkező 154 útvonalhoz, és a 302 chipet a 306 chiphez csatlakoztató 504 útvonal lehet egy lassú útvonal, mint az 1D ábrán lévő T_S késéssel rendelkező 152 útvonal. Egy az eddig használatos megközelítéssel összhangban lévő interfészben, ha az 502 és 504 útvonalak közötti késésbeli különbség meghaladja a sín órajel egy periódusát, az 502 útvonalon tompításra lenne szükség annak érdekében, hogy fenntartsuk az adatok szinkronizációját, amint azt megelőzően leírtuk. Viszont a 332 rugalmas interfész egységet

alkalmazó 300 interfészek alkalmazkodnak az 502 és 504 útvonalak közötti késésbeli különbségekhez. A 304 és 306 chipekben lévő 332 rugalmas interfészek mindegyikében a cél ciklusnak ugyanazon ciklusra való beállításával, amint azt fentebb leírtuk, a 302, 304 és 306 chipek közötti adat szinkronitás fenntartható.

A jelen találmány alternatív kiviteli példáit is megvalósíthatjuk. A kettő rugalmassággal rendelkező 332 interfész egység egy alternatív kiviteli példáját ábrázoljuk a 6A ábrán. A 6A ábrán bemutatott 332 rugalmas interfész kiviteli példája a 408 és 422 tároló eszközöket vezető 402 és 416 MUX-okat tartalmaz, mint a 4A ábrán bemutatott 332 rugalmas interfész kiviteli példájában. Viszont a 6A ábra 332 interfész egysége tartalmaz egy 316 lokális órajel által órajelezett második elfogó tároló eszközt (a 628 tároló eszközt) is a 630 tároló eszköz mellett, amely megfelel a 4A ábrán mutatott kiviteli példában lévő 430 tároló eszköznek. Továbbá a 4A ábrán lévő 432 MUX-nak megfelelő kimeneti 632 MUX-ot az elfogó tároló eszközöktől távolabbra mozdítottuk el a 6A ábrán lévő 332 interfész egység kiviteli példájában.

Ez csökkenti a késést magán az interfészen keresztül. A 628 és 630 tároló eszközök D bemeneteit a 422 és 408 tároló eszközök 424 és 410 kimeneteihez csatlakoztatjuk, ebben a sorrendben. A 628 és 630 tároló eszközök D bemeneteinél lévő adatokat a tároló eszközökbe a 316 lokális órajel által órajelezzük. Így az adatot a 316 lokális órajelben a 632 MUX előtt fogjuk el. A 322 adat folyamat oly módon állítjuk vissza, hogy kiválasztjuk a 628 és 630 tároló eszközök 634 és 636 kimeneteinek az egyikét kimenetre tételre a 632 MUX-on keresztül, a



343 kiválasztás vezérlő irányítása alatt. Ez hasonló a kimene-
ti 432 MUX tevékenységéhez a 4A ábra kiviteli példájában, vi-
szont itt a 343 kiválasztás vezérlőt fázisban egy fél periód-
dussal eltoltuk a 4A ábrán lévő 343 kiválasztás vezérlőhöz ké-
pest. A 6A ábrán a 332 interfész kiviteli példájában az adatot
a 316 lokális órajel egy emelkedő ágán tároljuk el. Érthető
viszont egy átlagos jártasságú szakember számára, hogy alter-
natív kiviteli példák az adatot eltárolhatják egy leeső ágon
is, feltéve, hogy más vezérlő jeleket fázisban megfelelően ál-
lítottunk be. Például egy olyan kiviteli példában, amelyben a
628 és 630 tároló eszközök a 316 lokális órajel leeső ágán tá-
rolnak el, a 343 kiválasztás vezérlő fázisa egy fél periódus-
sal lenne eltolva.

A 6A ábrán lévő 332 interfész egység kiviteli példájának
működését jobban megérthetjük, ha most a 6B ábrára hivatko-
zunk, amely ahhoz egy időzítési diagramot ábrázol. Mivel a 402
és 416 MUX-ok, 408 és 422 tároló eszközök, csakúgy, mint a 336
I/O órajel és a 316 lokális órajel közösek a 4A és 4B ábrákon
lévő 332 interfész egység kiviteli példái számára, a 6B ábrán
bemutatott időzítési diagram ezen struktúrákkal kapcsolatos
részleteit nem tárgyaljuk újra a rövidség érdekében.

A 6B ábra időzítési diagramjának a 628 és 630 tároló esz-
közökhöz, illetve a 632 MUX-hoz társított részletére fókuszál-
va, az a adat értéket a 316 lokális órajel t_8' ágán tároljuk el
a 630 tároló eszköz 634 kimenetére. Hasonlóan, a b adat érté-
ket a 316 lokális órajel t_9' ágán tároljuk el a 628 tároló esz-
köz 636 kimenetére. Annak érdekében, hogy az adat a 338 adat
kimeneten a cél ciklusnál jelenjen meg, a 343 kiválasztás ve-
zérlő emelkedő ágát késleltetnünk kell a 316 lokális órajel t_9'

ágáig, amely egybeesik a cél ciklussal. Így a 632 MUX-tól származó a adat érték a 338 adat kimeneten a 343 kiválasztás vezérlő t_{10} ágánál jelenik meg. Ily módon a 343 kiválasztás vezérlő fázis-szinkronizált a 316 lokális órajellel, mivel a legfelső lapos részletei a 316 lokális órajel emelkedő átmenetei között vannak középre helyezve. Hasonlóan, a b adat érték a 338 adat kimeneten a 343 kiválasztás vezérlő t_{11} leeső ágánál jelenik meg, és ezután folytatjuk a 322 adat folyam visszaállítását a 343 kiválasztás vezérlő soron következő átmenetei alapján.

Ezen kívül a 332 rugalmas interfész más előre meghatározott rugalmasságokkal rendelkező kiviteli példáit is megvalósíthatjuk a jelen találmánnyal összhangban. Ezek magukban foglalhatnak feles periódusú rugalmasságokat. A jelen találmány egy 1,5 periódusú rugalmassággal rendelkező kiviteli példáját ábrázoljuk a 7A ábrán. A 322 adatot a 702 és 704 tároló eszközök D bemeneteihez csatlakoztatjuk. A 702 és 704 tároló eszközök „polaritás-tartó”, vagy „átáramló” („flush”) tároló eszközök. Az ilyen tároló eszközökre hivatkozhatunk „átlátszó” tároló eszközökként is. Amikor a 702 tároló eszközben a C órajel egy első előre meghatározott logikai állapottal vagy szinttel rendelkezik, a D bemeneten lévő adat átáramlik rajta a 706 kimenetre. Az órajelnek az első állapotból egy előre meghatározott második logikai állapotba való átmenetekor a 702 tároló eszköz eltárolja a D bemeneten lévő adatot, és a 706 kimeneten lévő adatot megtartjuk az órajelnek a második állapotából az első állapotba való következő átmenetig. (a 7. ábra kiviteli példájában az első állapot egy „alacsony” logikai szintnek felel meg, míg a második állapot egy „magas” logikai szintnek



felel meg, ami által az átmenet egy emelkedő ágból áll. Egy alternatív kiviteli példa viszont a komplementer logikai állapotokkal rendelkezne.) A 704 tároló eszköz átáramoltatja a D bemenetén lévő adatot a 708 kimenetre, ha a C órajele a második logikai szinttel rendelkezik. Az órajel második logikai szintjéről az első logikai szintjére való átmenetkor az adatot eltároljuk, és megtartjuk, amíg az órajel át nem megy a második logikai szintre az első logikai szintre, ami a 7. ábra kiviteli példájában „alacsony”. Az átáramoltató tulajdonság lehetővé teszi az adat számára, hogy elérhetővé váljon anélkül, hogy várnunk kellene addig, amíg egy tároló eszköz órajelvezve nem lesz, ezáltal a 332 interfész egység egy „alacsony késésű - alacsony rugalmasságú” kiviteli példáját valósítjuk meg.

A 7A ábrán bemutatott 332 rugalmas interfész kiviteli példájában a 702 és 704 tároló eszközöket a 336 I/O órajel órajelezi. A 7A ábra 332 rugalmas interfésze egy dupla adat sebességű (DDR - double data rate) eszköz, vagyis az adatot eltároljuk a 702 és 704 tároló eszközök egyikébe a 336 I/O órajel minden egyes átmenetkor, és a 336 I/O órajel periódusa a kétszerese a lokális órajelének (a 314 lokális órajelének és a 316 lokális órajelének) illetve a 306 sín órajelének. A 306 sín órajel egy olyan adat ablak közepére helyezük, amely két lokális órajel periódus széles.

A 7A ábrán lévő 332 rugalmas interfész egység kiviteli példáját jobban megértjük, ha a 7B ábrán mutatott vonatkozó időzítési diagramra hivatkozunk. Amikor az a adat érték a 332 rugalmas interfész egységhez érkezik, átáramlik a 702 tároló eszköz 706 kimenetére, mert a 336 I/O órajel alacsony a 752

„legfelső” lapos részletnél. Más szavakkal, az a adat érték a 702 tároló eszköz 706 kimenetén a 336 I/O órajel t_1 ága előtt jelenik meg.

Az a adat részlet a t_1 átmenetet a 336 I/O órajel egynegyed periódusával előzi meg, ami a 316 lokális órajel fél periódusának felel meg. A 336 I/O órajelet a sín órajelből származtatjuk, és fázisban eltoljuk a sín órajelhez képest a kibocsátáskor a chippek közötti útvonal késésével, amint azt megelőzően leírtuk. Ezen kívül a 336 I/O órajelnek egy negyed periódus fázis előnyt adunk. A t_1 ágnál az a adat értéket eltávololjuk, ami által megtartjuk azt a 316 lokális órajel egy periódusáig.

Hasonlóan, a b adat érték átáramlik a 704 tároló eszköz 708 kimenetére, amikor a 330 RX-től megérkezik a 332 rugalmas interfész egységhez, mert a 336 I/O órajel magas a 754 legfelső lapos részletnél. A b adat értéket ezután a 336 I/O órajel t_2 negatív ága a 708 kimeneten tartja. A b adat értéket a 316 lokális órajel egy periódusáig tartjuk meg. Ily módon az adat értékek a 702 és 704 tároló eszközök 706 és 708 kimenetén, ebben a sorrendben, 1,5 lokális órajel perióduson keresztül jelennek meg, ami a 7A ábrán lévő 332 rugalmas interfész kivitelemi példájának a rugalmassága.

A 322 adat folyamat a 338 kimenetnél a 710 és 712 tároló eszközök, illetve a 714 MUX által rekonstruáljuk. A 706 és 708 kimenetek által képviselt két adat folyamat ebben a sorrendben tárolja el a 316 lokális órajel a 710 és 712 tároló eszközökbe. Az adatot a 316 lokális órajel egy előre meghatározott ágán tároljuk el (ami pozitív a 7. ábra kivitelemi példájában), ahol a cél ciklust beállíthatjuk úgy, hogy az adat érték ru-



galmasságának másfél ciklusán belül bárhol tűnjön fel, amint azt megelőzően leírtuk. Így az a adat értéket, összhangban a 7B ábrán lévő időzítési diagrammal, eltároljuk a 710 tároló eszközbe a 316 lokális órajel t_3 ágán, és a 338 adat kimenetre kapcsoljuk át a 714 MUX-on keresztül, a 343 kiválasztás vezérlő hatására. Az adatot a 710 tároló eszközből választjuk ki, amikor a 343 kiválasztás vezérlő egy első előre meghatározott értékkel rendelkezik, és a 712 tároló eszközből választjuk ki akkor, amikor a 343 kiválasztás vezérlő egy második előre meghatározott értékkel rendelkezik. Az első érték „magas”, a második pedig „alacsony” a 7. ábra kiviteli példájában, viszont látnunk kell, hogy más előre meghatározott értékek is lehetségesek. A 316 lokális órajel t_4 ágánál a b adat értéket eltároljuk a 712 tároló eszközbe, és a 338 adat kimenetre kapcsoljuk át annak a hatására, hogy a 343 kiválasztás vezérlő második értékkel rendelkezik. A következő adat értékeket ezután szekvenciálisan a 338 kimenetre tesszük oly módon, hogy felváltva választjuk ki a kimenetet a 710 és 712 tároló eszközökről a 714 MUX-on keresztül, amint azt a 7B ábrán mutatjuk.

Ezen kívül a rugalmas interfészek a jelen találmány alapelveivel összhangban kinyújthatóak, ami által megnövelhetjük a rugalmasságokat, irányító és tároló elemek hozzáadásával. Egy N-1 sín órajel periódus rugalmassággal rendelkező rugalmas interfész egy kiviteli példáját ábrázoljuk a 8A ábrán. A 332 interfész egység 802 MUX-ok N sokaságát tartalmazza. Minden egyes MUX-ban egy 804 bemenet 322 adat folyamat fogad a 330 RX-től. Egy második 806 bemenet a 808-818 tároló eszközök vonatkozó egyike által kimenetre tett jelet fogad. A 808-818 tároló eszközök mindegyike egy tároló eszköz - párt tartalmaz. A

808, 812 és 816 tároló eszközökben a pár első tagjának belső kimenetét a pár második tagjának belső bemenetéhez csatlakoztatjuk, mester-szolga (master-slave) viszonyban. A 808, 812 és 816 tároló eszközök 820 kimenetet biztosítanak a szolga részlettől a megfelelő 802 MUX egy 806 bemenetéhez. A 808, 812 és 816 tároló eszközök szolga részletei a D bemeneten lévő adatot a 336 I/O órajel egy emelkedő ágán tárolják el. A 808, 812 és 816 tároló eszközök vonatkozó D bemeneteit egy megfelelő 802 MUX kimenetéhez csatlakoztatjuk. A 810, 814 és 818 tároló eszközök a tároló eszköz - pár első tagjától származó 822 kimenetet egy megfelelő 802 MUX 806 bemenetéhez csatlakoztatják. A tároló eszköz - pár első tagja átlátszó, és az első 828 bemeneten lévő adat átáramlik a 822 kimenetre a 336 I/O órajel egy „legfelső” lapos részletén. A 810, 814 és 818 tároló eszközök mindegyikében a 828 bemenetet a tároló eszköz - pár második tagjától származó megfelelő 826 kimenethez csatlakoztatjuk. Ezen kívül a pár második tagjának egy 830 bemenetét a megfelelő 802 MUX egy kimenetéhez csatlakoztatjuk. A tároló eszköz - pár második tagja szintén átlátszó, és az adatot átáramoltatja a 336 I/O órajel legfelső lapos részleténél. A 810, 814 és 818 tároló eszközöket alkotó párok első és második tagja az adatot a 336 I/O órajel ellentétes polaritással rendelkező legfelső lapos részleténél áramoltatják át.

A 802 MUX-ok választanak a 804 és 806 bemeneteken lévő jelek közül egy vonatkozó kapu jel (832-842 kapuk) hatására. A 832-842 kapu jeleket a 8B ábrán bemutatott időzítési diagrammal kapcsolatban fogjuk tovább tárgyalni.

Az adatot a 844 adat tároló eszközökbe a lokális órajelen keresztül tároljuk el, amelyek mindegyike egy kimeneti jelet

fogad a 808-818 tároló eszközök megfelelő egyikétől. Az adatot a 844 adat tároló eszközökbe a 316 lokális órajel által tároljuk el. A 808, 812 és 816 tároló eszközök 824 kimenetét egy megfelelő 844 adat tároló eszköz D bemenetéhez biztosítjuk. A 824 kimeneteket a 808, 812 és 816 tároló eszközök mester részletétől kapjuk meg, amely, amint azt megelőzően leírtuk, átlátszó. A 844 adat tároló eszközök közül a fennmaradóak a D bemeneteiken fogadják a 810, 814 és 818 tároló eszközök megfelelő egyikét alkotó tároló eszköz - párok második tagjának 826 kimenetéről származó jelet. A tároló eszköz - párok ezen második tagja szintén egy átlátszó tároló eszköz, amelyen adat a 336 I/O órajel egy olyan polaritásánál áramlik át, amely ellentétes azzal, amelynél adat a tároló eszköz - párok első tagján áramlik át.

A 846 MUX kiválasztja a 844 adat tároló eszközökben tartott jelek egyikét kimenetre tételre. A 844 adat tároló eszközök mindegyikének egy kimenetét csatlakoztatjuk egy megfelelő bemenethez, a 848-858 bemenetek egyikéhez. A jel kimenet a chip, például a 302 vagy 304 chip számára a 846 MUX 860 kimenetén. A 846 MUX a 848-858 bemenetek közül a 343 kiválasztás vezérlőn keresztül választ. A 343 kiválasztás vezérlő k jelet tartalmaz, ahol 2^k egyenlő N-nel.

Egy N rugalmassággal rendelkező alternatív kiviteli példát megvalósíthatunk a 8A ábrán mutatott áramkör használatával. A 844 adat tároló eszközök megfelelő D bemeneteit a 808, 812 és 816 tároló eszközök megfelelő egyikének 820 kimenetéhez és a 810, 814 és 818 tároló eszközök megfelelő egyikének 826 kimenetéhez (a 8A ábrán mutatott 824 és 822 kimenetek helyett)

csatlakoztatva N rugalmasságot érünk el. A 8A ábrán lévő 332 interfész egység áramköre máskülönben változatlan.

Most a 8B ábrára utalunk, amely egy időzítési diagramot ábrázol a 8A ábrán bemutatott 332 interfész egység számára. Az a adat érték az adat folyamban érkezik a 322 adaton T_1 interfész késés után, és a 802 MUX-ok megfelelő egyikéhez biztosított 832 kapu kiadásának hatására (melyet a 862 „legfelső” lapos részlet képvisel) az a adat érték átadódik a megfelelő 802 MUX-on keresztül a 808 tároló eszköz D bemenetéhez. A 336 I/O órajel t_1 ágán a 808 tároló eszköz megtartja az a adat értéket, amelyet visszacsatlakoztatunk a 808 tároló eszköz 820 kimenetén a 802 MUX-ok megfelelő egyikének 806 bemenetéhez. A 832 kapu t_2 ágánál a megfelelő 802 MUX kiválasztja a jelet a 806 bemeneten, amely az a adat értékkel rendelkezik. A 832 kaput negáljuk a 306 sín órajel N-1 periódusáig, ami által az a adat értéket a 808 tároló eszköz D bemenetén, és ezért a 808 tároló eszköz 824 kimenetén tartjuk. Az a adat értéket a 306 sín órajel egy további ciklusáig megtartjuk a 336 I/O órajel t_3 átmenete által, ami után a 832 kapu t_4 átmenete átkapcsolja a megfelelő 802 MUX-ot az adat folyam kiválasztására a 322 adaton, ami által a 336 I/O órajel t_5 ágánál a 322 adatban lévő (N+2)-edik adat értéket a 808 tároló eszközbe tároljuk el a 336 I/O órajel t_5 ágán. Az a adat értéket a 844 adat tároló eszközbe fogjuk el, amely a 808 tároló eszköz 824 kimenetét fogadja a 316 lokális órajel t_6 ágán, majd az a adat érték megjelenik a 848 kimeneten. Az a adat értéket a 848 kimeneten a 306 sín órajel N periódusáig tartjuk.

A 322 adatban lévő következő adat értéket, a b adat értéket hasonlóan a 306 sín órajel N periódusáig tartjuk a 810 tá-



roló eszközhöz csatlakoztatott 844 adat tároló eszközben. Amikor a b adat érték a 332 egységhez megérkezik, a 834 kapu ki van adva (amelyet a 864 „legfelső” lapos részlet ábrázol), és kiválasztjuk a megfelelő 802 MUX 804 bemenetén lévő 322 adatot. A b adat érték megjelenik a 810 tároló eszköz 830 bemenetén és eltároljuk a 336 I/O órajel t_7 ága által, ami által a b adat érték megjelenik a 810 tároló eszköz 826 kimenetén. A 810 tároló eszköz 826 kimenetét visszatápláljuk a 810 tároló eszköz 828 bemenetéhez. Mivel a 336 I/O órajel negált a t_7 ágat követően, a 810 tároló eszköz 826 kimenetén lévő b adat érték átesik a 810 tároló eszköz 822 kimenetére, ahol visszacsatlakoztatjuk a megfelelő 802 MUX 806 bemenetéhez. A 336 I/O órajel t_8 ágánál a b adat értéket a 810 tároló eszköz 822 kimenetén tartjuk. A 834 kapu t_9 ágánál a megfelelő 802 MUX átkapcsol és a 806 bemenetet választja ki a 810 tároló eszköz 822 kimenetén tartott b adat érték kimenetre tételére a 810 tároló eszköz 830 bemenetébe. A b adat érték ezután átesik a 810 tároló eszköz 826 kimenetére, ahol visszacsatlakoztatjuk a 810 tároló eszköz 828 bemenetéhez, és folytatjuk a b adat érték visszatáplálását a 834 kapu jelet fogadó megfelelő MUX 806 bemenetére. Ily módon a b adat értéket a 810 tároló eszközben a 826 kimeneten a 306 sín órajel $N+1$ periódusáig tartjuk, egy órajel periódussal a 834 kapu t_{10} ágánál lévő átmenete után.

A 836 kapu fázisában a 336 I/O órajel egy periódusával van késleltetve a 834 kapuhoz képest, és ehhez hasonlóan minden egyes következő 838-842 kapu jel fázisban a 336 I/O órajel egy periódusával van késleltetve a láncban a megelőző kapu jelhez képest. Ily módon a soron következő 808-818 tároló eszközök mindegyike a 322 adatban lévő soron következő adat érté-

ket tárolja, és az adat értéket a 306 sín órajel $N+1$ periódusáig tartja. Ezután minden egyes adat értéket a vonatkozó 808-816 tároló eszközben eltárolhatjuk a megfelelő 844 adat tároló eszközben a 316 lokális órajel minden egyes periódusában. Így a b adat értéket a megfelelő 844 adat tároló eszközbe a 316 lokális órajel t_{11} ágán tároljuk el és a 850 kimeneten jelenik meg, és hasonlóképpen, a c adat értéket a neki megfelelő 844 adat tároló eszközbe a 316 lokális órajel t_{12} ágán órajelezzük, és ez a 852 kimeneten jelenik meg. A 818 tároló eszközhöz csatlakoztatott utolsó 844 adat tároló eszköz az $(N+1)$ -edik adat értéket a 316 lokális órajel t_{13} ágán tárolja el.

Az adatot a 846 MUX-ról tesszük kimenetre, a 343 kiválasztás vezérlő hatására, amely k jelet tartalmaz. A 343 kiválasztás vezérlő k jelének mindegyike periodikus. Az $S(0)$ -lal jelölt „nulladik” jel egy fél periódussal rendelkezik, amely egyenlő a 306 sín órajel periódusával. A $(k-1)$ -edik jel egy fél periódussal rendelkezik, amely N sín órajel periódus. Az $S(0)$ és $S(N)$ között szekvenciálisan lévő minden egyes jel olyan periodicitással rendelkezik, amely a megelőző jel periódusának a kétszerese. A 844 adat tároló eszközökben tárolt adat értékeket, amelyek a 846 MUX-ban egy megfelelő 848-858 bemeneten jelennek meg, szekvenciálisan órajelezzük ki a 860 adat kimenetre, a 343 kiválasztás vezérlő hatására. Az a adat értéket a cél ciklusnál órajelezzük ki, $N-1$ rugalmasság mellett, az $S(N)$ t_{14} ágán a 343 kiválasztás vezérlőben. A fennmaradó adat értékeket szekvenciálisan órajelezzük ki, a 343 kiválasztás vezérlőben lévő k jel ciklikus átmenetének hatására. Bár a 343 kiválasztás vezérlőt alkotó jeleket úgy ábrázoltuk, mint amelyek fázis szinkronnal rendelkeznek valamely emelkedő



ágon, egy átlagos tudással rendelkező szakember számára érthető, hogy egy alternatív kiviteli példában a komplementer fá-zist használhatjuk.

A 8. ábrán bemutatott 332 egységet egy IAP közben inicializáljuk, amint azt megelőzően a 4. ábrával kapcsolatosan leírtuk. Egy alkalmas inicializációs minta $(N-1)$ -es periodicitással rendelkezhet egy $(N-1)$ -es rugalmassággal rendelkező kiviteli példa esetén, a 8. ábrán bemutatott 332 egység kiviteli példája rugalmasságának megfelelően. Egy N rugalmassággal rendelkező alternatív kiviteli példa esetén, amint azt fentebb leírtuk, a sync minta N periodicitással rendelkezhet. Az inicializáció közben egy olyan sync minta esetén, amely egy „1”-essel rendelkezik, amit „0”-k sokasága követ, az „1”-est mintavételeznénk a 808 tároló eszközben, amikor a 343 kiválasztás vezérlőben lévő jelek megfelelően vannak sorba rendezve.

Ily módon egy mechanizmust biztosítottunk egy adatfeldolgozó rendszerben lévő interfészekon keresztüli adat szinkronitás fenntartására. A fogadott adatot elfogjuk tároló elemek valamely sokaságába, és szelektív módon elirányítjuk a fogadó chipbe egy előre kiválasztott cél cikluson, amelyet a chip órajellel szinkronizálunk. A kezdeti szinkronizációt dinamikus alakítjuk ki, egy IAP végrehajtása által. A jelen találmány mechanizmusa adat szinkronizációt biztosít egy olyan adatfeldolgozó rendszerben, amely egy sín órajel ciklusnál többel változó késésekkel rendelkezik.

Szabadalmi igénypontok

1. Interfész berendezés, **azzal jellemezve**, hogy magában foglal:

első tároló eszközt adat értékek első halmazának tárolására;

második tároló eszközt adat értékek második halmazának tárolására;

az első és második tároló eszközökhöz csatlakoztatott áramkört az első tároló eszközből származó első adat érték és a második tároló eszközből származó második adat érték szekvenciális kimenetre tételéhez legalább egy vezérlő jel hatására, ahol az első és második tároló eszközök adat értékeket tartanak meg első órajel előre meghatározott számú ciklusáig, valamint első és második kiválasztási áramkört, amelyek mindegyike az első és második tároló eszközök megfelelő bemeneteihez ebben a sorrendben csatlakoztatott kimenettel rendelkezik, amely első és második kiválasztási áramkör mindegyike rendelkezik első bemenettel adat folyam fogadásához és az első és második tároló eszközök vonatkozó kimenetéhez csatlakoztatott második bemenettel, ahol az első és második kiválasztási áramkör működtethető az első és második bemenetek egyikén lévő jel kimenetre tételre való kiválasztásához, első és második vezérlő jelek hatására.

2. Az 1. igénypont szerinti berendezés, azzal jellemezve, hogy a kimenetre tétel második órajellel van szinkronban.

3. A 2. igénypont szerinti berendezés, azzal jellemezve, hogy a második órajel az első órajel periódusával egyenlő periódussal rendelkezik.



4. Az 1 - 3. igénypontok bármelyike szerinti berendezés, azzal jellemezve, hogy az első és második tároló eszközök működtethetőek adat értékek sokaságát magában foglaló adat folyam fogadásához.

5. Az 1 - 4. igénypontok bármelyike szerinti berendezés, azzal jellemezve, hogy az első tároló eszköz adatot tárol el második órajel első előre meghatározott ágán, és a második tároló eszköz adatot tárol el a második órajel második előre meghatározott ágán.

6. Az 5. igénypont szerinti berendezés, azzal jellemezve, hogy a második órajelet küldő eszköztől fogadott óraütésjelből származtatjuk.

7. Az 1 - 6. igénypontok bármelyike szerinti berendezés, azzal jellemezve, hogy az első és második vezérlő jelek mindegyike olyan periódussal rendelkezik, amely az első órajel periódusának kétszerese, és ahol a második vezérlő jel az első vezérlő jel komplemente.

8. Az 1 - 7. igénypontok bármelyike szerinti berendezés, azzal jellemezve, hogy az első és második kiválasztási áramkörök ebben a sorrendben magukban foglalnak első és második multiplexereket.

9. Az 1. igénypont szerinti berendezés, azzal jellemezve, hogy az első és második adat értékek szelektív módon szekvenciálisan való kimenetre tételéhez működtethető áramkör magában foglal multiplexert, amely rendelkezik első bemenettel az első adat érték fogadására és második bemenettel a második adat érték fogadására, ahol a multiplexer az első és második adat értékek egyikét kiválasztja kimenetre tételre olyan periódussal



rendelkező kiválasztás vezérlő jel hatására, amely az első órajel periódusának előre meghatározott többszöröse.

10. A 9. igénypont szerinti berendezés, azzal jellemezve, hogy a kiválasztás vezérlő jel periódusa az első órajel periódusának kétszerese.

11. Az 1. igénypont szerinti berendezés, azzal jellemezve, hogy az első és második adat értékek szelektív módon szekvenciálisan való kimenetre tételéhez működtethető áramkör magában foglal:

első tároló eszközt, amely olyan bemenettel rendelkezik, ami az első adat érték fogadásához működtethető;

második tároló eszközt, amely olyan bemenettel rendelkezik, ami a második adat érték fogadásához működtethető; és

az első tároló eszköz kimenetéhez csatlakoztatott első bemenettel és a második tároló eszköz kimenetéhez csatlakoztatott második bemenettel rendelkező multiplexert, amely multiplexer az első és második adat értékek egyikének kimenetre tételéhez való kiválasztásához működtethető egy olyan periódussal rendelkező vezérlő jel hatására, amely az első órajel periódusának előre meghatározott többszöröse, és ahol az első és második tároló eszköz adatokat tart meg az első órajel első és második előre meghatározott szintjein.

12. A 11. igénypont szerinti berendezés, azzal jellemezve, hogy az első tároló elem magában foglal harmadik tároló eszközt, és a második tároló elem magában foglal negyedik tároló eszközt, amely harmadik tároló eszköz olyan átlátszó tároló eszköz, ami adatot második órajel első előre meghatározott szintjénél tart meg, és amely negyedik tároló eszköz



olyan átlátszó tároló eszköz, ami adatot az első előre meghatározott órajel szint komplementer szintjénél tart meg.

13. A 12. igénypont szerinti berendezés, azzal jellemezve, hogy magában foglal továbbá a harmadik és negyedik tároló eszközökhöz ebben a sorrendben csatlakoztatott kimenetekkel rendelkező harmadik és negyedik multiplexert, amely harmadik és negyedik multiplexerek rendelkeznek első bemenettel adat értékek folyamának fogadására és a harmadik és negyedik tároló eszközök kimenetéhez ebben a sorrendben csatlakoztatott második bemenettel, ahol a harmadik multiplexer működtethető az első és második bemenetek közötti választáshoz első kapu jel hatására, és a negyedik multiplexer működtethető az első és második bemenetek közötti választáshoz olyan második kapu jel hatására, amely az első kapu jel komplementere.

14. Az 1. igénypont szerinti interfész berendezés, azzal jellemezve, hogy magában foglal:

első multiplexerek M sokaságát, amelyek mindegyike rendelkezik első bemenettel adat folyam fogadására, valamint kimenettel;

első tároló eszközök M sokaságát, amelyek mindegyike rendelkezik megfelelő első multiplexer kimenetéhez csatlakoztatott bemenettel, amely első tároló eszközök mindegyike rendelkezik első kimenettel, valamint a megfelelő első multiplexer második bemenetéhez csatlakoztatott második kimenettel;

második multiplexerek M sokaságát, amelyek mindegyike rendelkezik első bemenettel az adat folyam fogadására, valamint kimenettel;

második tároló eszközök M sokaságát, amely második tároló eszközök mindegyike magában foglal tároló eszköz - párt, ahol



a pár első tagjának bemenetét megfelelő második multiplexer kimenetéhez csatlakoztatjuk, és rendelkezik kimenettel, valamint a pár második tagja rendelkezik a pár első tagjának a kimenetéhez csatlakoztatott bemenettel, és a megfelelő második multiplexer második bemenetéhez csatlakoztatott kimenettel;

harmadik tároló eszközök sokaságát, amelyek rendelkeznek az első tároló eszközök megfelelő egyikének az első kimenetéhez csatlakoztatott bemenettel, és harmadik multiplexer megfelelő bemenetéhez csatlakoztatott kimenettel;

negyedik tároló eszközök sokaságát, amelyek rendelkeznek a tároló eszköz - pár első tagjának a kimenetéhez csatlakoztatott bemenettel, és a harmadik multiplexer megfelelő bemenetéhez csatlakoztatott kimenettel, amely harmadik multiplexer működtethető adat értékek sorozatának kimenetre tételéhez kiválasztás vezérlő hatására, amely kiválasztás vezérlő magában foglalja jelek k sokaságát, ahol a jelek k sokaságának első jele olyan fél periódussal rendelkezik, amely egyenlő első órajel periódusával, ahol az első és második tároló eszközök mindegyike működtethető olyan második óraütés jel hatására, ami olyan fél periódussal rendelkezik, amely egyenlő az első órajel periódusával, és ahol a harmadik tároló eszközök mindegyike működtethető az első órajel hatására.

15. A 14. igénypont szerinti berendezés, azzal jellemezve, hogy az első órajel magában foglal lokális chip órajelet, és a második órajel magában foglal bemenet/kimenet (I/O - input/output) órajelet.

16. A 14. igénypont szerinti berendezés, azzal jellemezve, hogy $2(k-1)$ egyenlő M -mel.

17. Eljárás integrált áramköri eszközök interfészelésére, **azzal jellemezve**, hogy magában foglalja a következő lépéseket:

adat értékek első halmazát tárolunk első tároló eszközben, ahol az első halmaz minden egyes adat értékét első órajel előre meghatározott számú ciklusáig tároljuk;

adat értékek második halmazát tárolunk második tároló eszközben, ahol a második halmaz minden egyes adat értékét első órajel előre meghatározott számú ciklusáig tároljuk;

szekvenciálisan kimenetre teszünk az első tároló eszközből származó első adat értéket és a második tároló eszközből származó második adat értéket oly módon, hogy áramkört csatlakoztatunk az első és második tároló eszközökhöz és vezérlő jellet biztosítunk az áramkörhöz, ahol a vezérlő jel olyan periódussal rendelkezik, amely az első órajel periódusának előre meghatározott többszöröse;

adat folyamat fogadunk első és második kiválasztási áramkör mindegyikének első bemeneténél;

az első és második tároló eszközök kimeneti jeleit biztosítjuk a vonatkozó első és második kiválasztási áramkör mindegyikének második bemenetéhez;

az adat értékek első halmazának mindegyikét kimenetre tesszük az első tároló eszközhöz az első kiválasztási áramkörtől első kiválasztási jel hatására; és

az adat értékek második halmazának mindegyikét kimenetre tesszük a második tároló eszközhöz a második kiválasztási áramkörtől második kiválasztási jel hatására.

18. A 17. igénypont szerinti eljárás, azzal jellemezve, hogy az adat értékek első és második halmaza magában foglalja adat folyamban fogadott adat értékek sokaságát.



19. A 17. igénypont szerinti eljárás, azzal jellemezve, hogy az első és második kiválasztási jelek komplementer jelek.

20. A 17. igénypont szerinti eljárás, azzal jellemezve, hogy az első és második kiválasztási jelek olyan periódussal rendelkeznek, amely az első órajel periódusának kétszerese.

21. A 17. igénypont szerinti eljárás, azzal jellemezve, hogy az első és második kiválasztási jelek olyan periódussal rendelkeznek, amely az első órajel periódusának előre meghatározott többszöröse.

22. A 17. igénypont szerinti eljárás, azzal jellemezve, hogy magában foglalja a következő lépést:

inicializáljuk az első és második kiválasztási jeleket.

23. A 22. igénypont szerinti eljárás, azzal jellemezve, hogy az első és második kiválasztási jelek inicializálásának lépése magában foglalja a következő lépéseket:

előre kiválasztott adat folyamot küldünk; és

beállítjuk az első és második kiválasztási jelek mindegyikének fázisát az előre kiválasztott adat folyam hatására.

24. A 23. igénypont szerinti eljárás, azzal jellemezve, hogy a kiválasztási jelek mindegyikének fázisa beállításának a lépése magában foglalja továbbá fázis kiválasztásának a lépését, ahol az első adat folyamban lévő első előre meghatározott adat értéket fogunk el az első tároló elemében, és második előre meghatározott adat értéket fogunk el a második tároló elemében.

25. A 23. igénypont szerinti eljárás, azzal jellemezve, hogy az adat folyam magában foglal szinkronizációs (sync) mintát.

26. Adat feldolgozó rendszer, **azzal jellemezve**, hogy magában foglal:

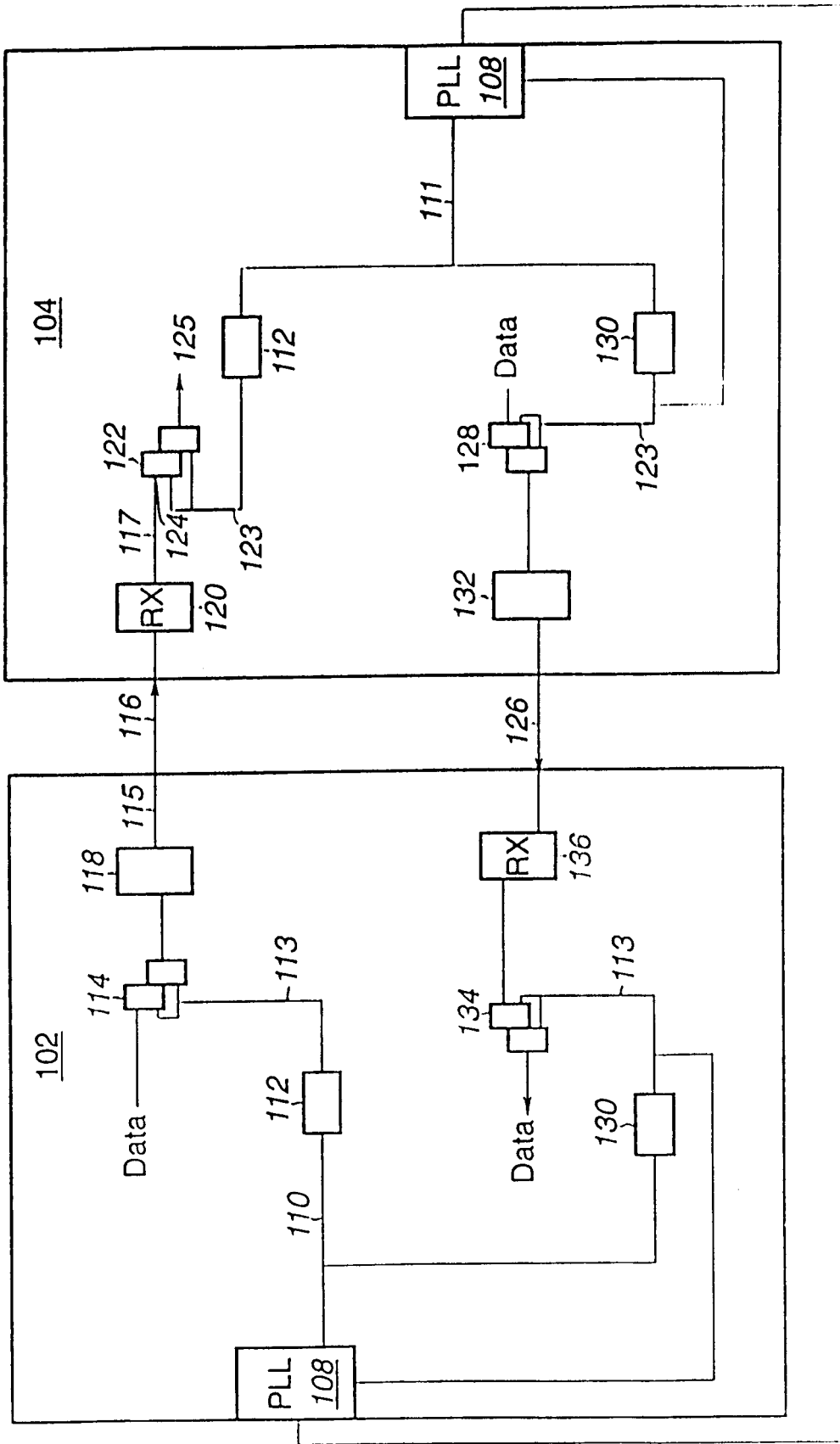
első adatfeldolgozó eszközt; és

az első adatfeldolgozó eszközhöz rugalmas interfészen keresztül csatlakoztatott második adatfeldolgozó eszközt, amely rugalmas interfész magában foglalja az 1 - 13. igénypontok bármelyike szerinti berendezést.

A meghatalmazott



Dr. Köteles Zoltán
szabadalmi ügyvivő
az S.B.G. & K. Szabadalmi Ügyvivői Iroda
tagja
H-1062 Budapest, Andrásy út 113.
Telefon: 461-1000 Fax: 461-1099



100

106

Fig. 1A
(Prior Art)

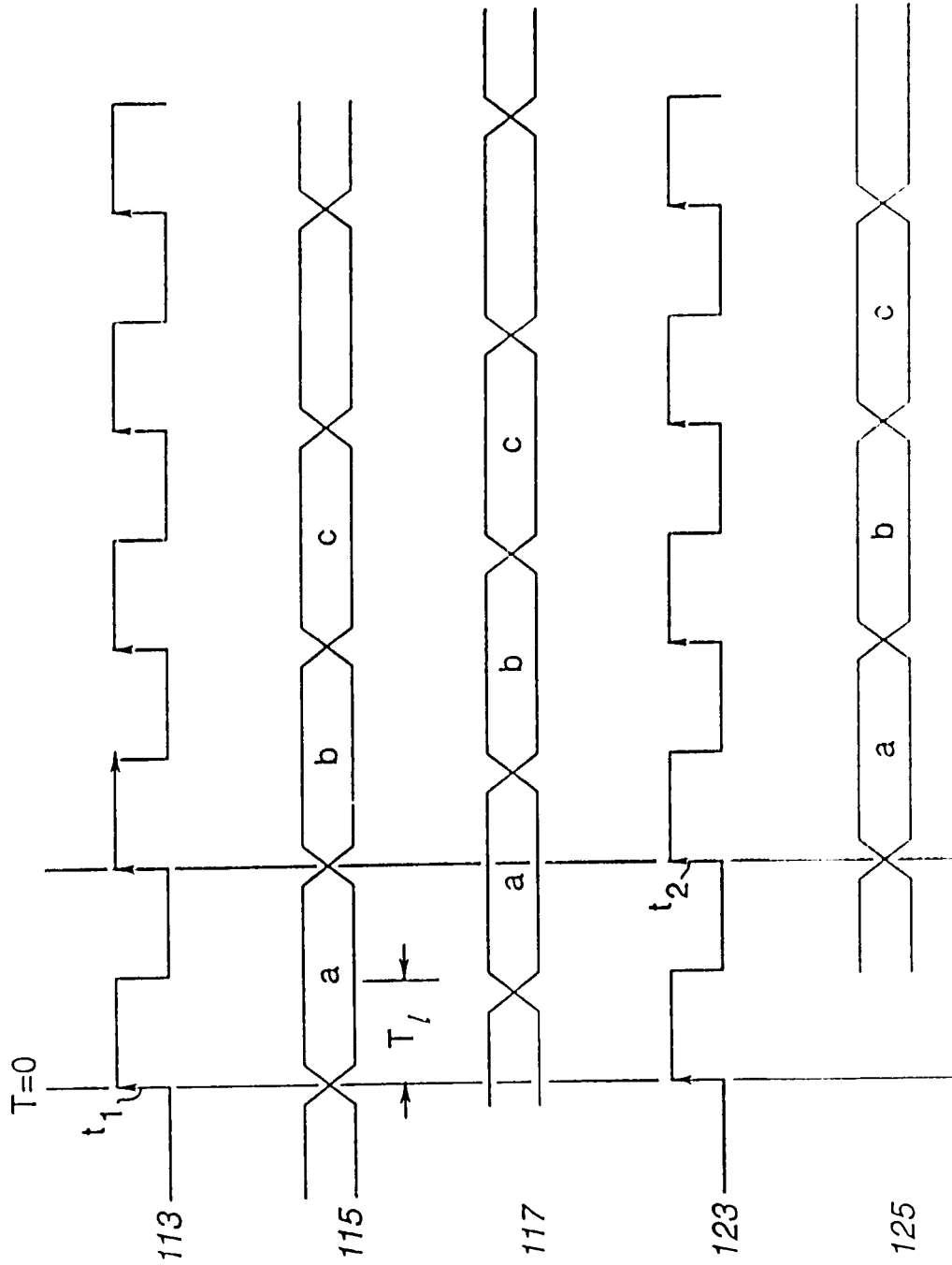


Fig. 1B
(Prior Art)

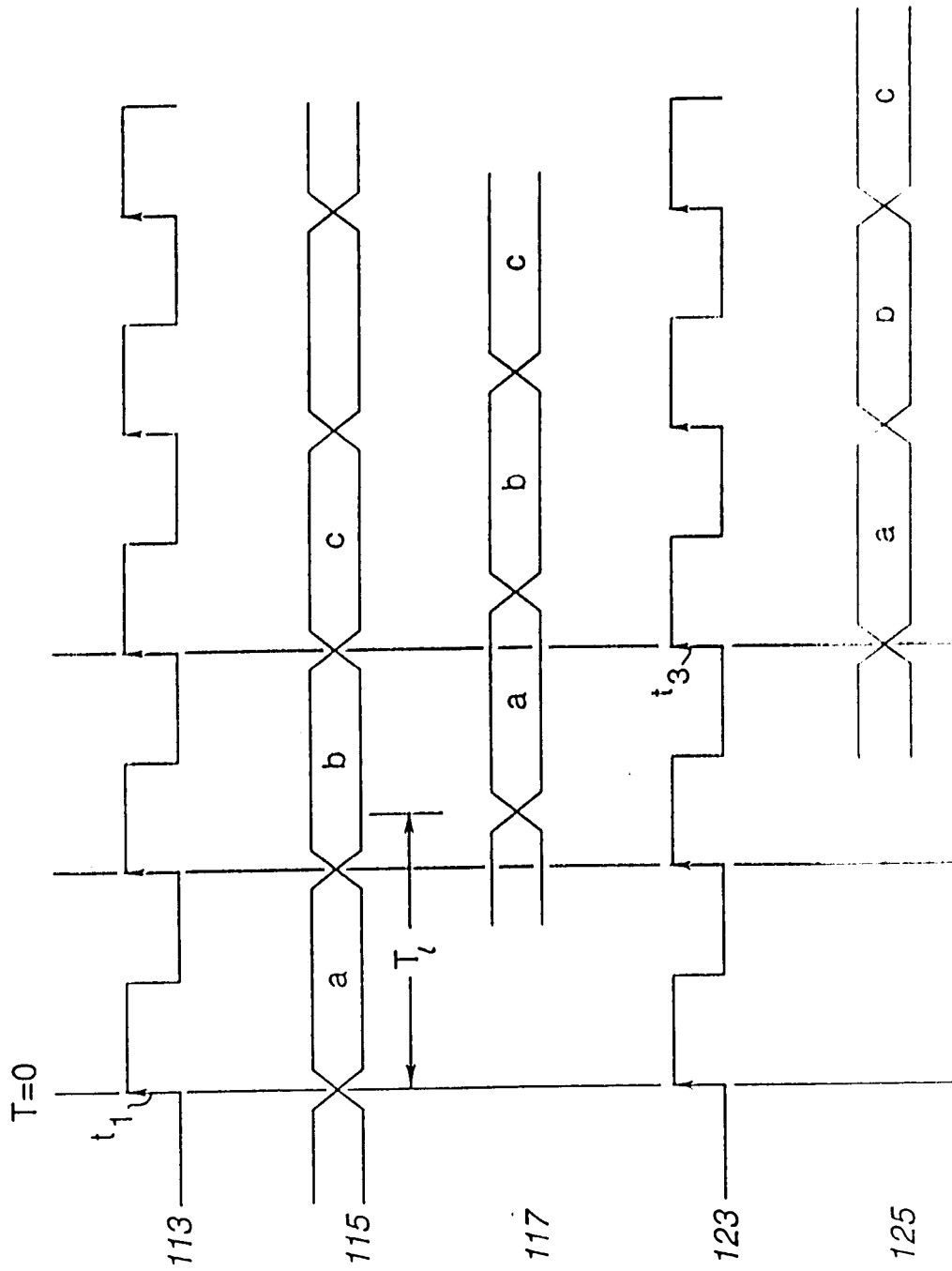


Fig. 1C
(Prior Art)

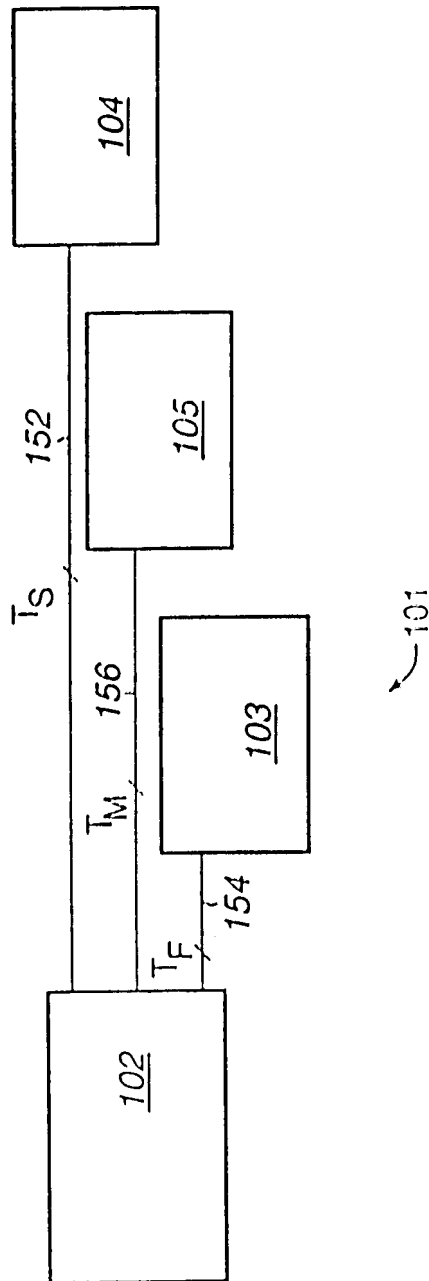


Fig. 1D

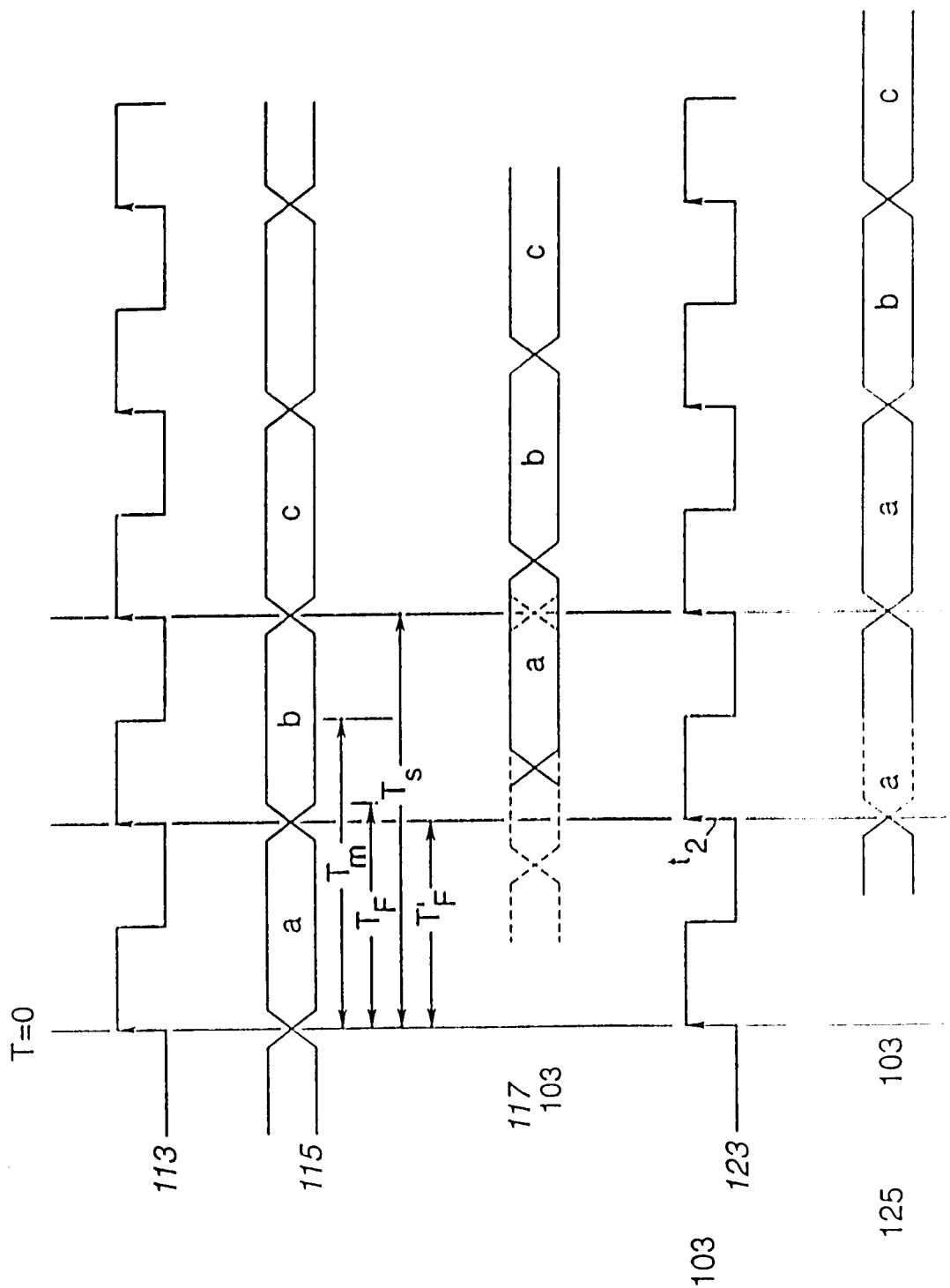


Fig. 1E
(Prior Art)

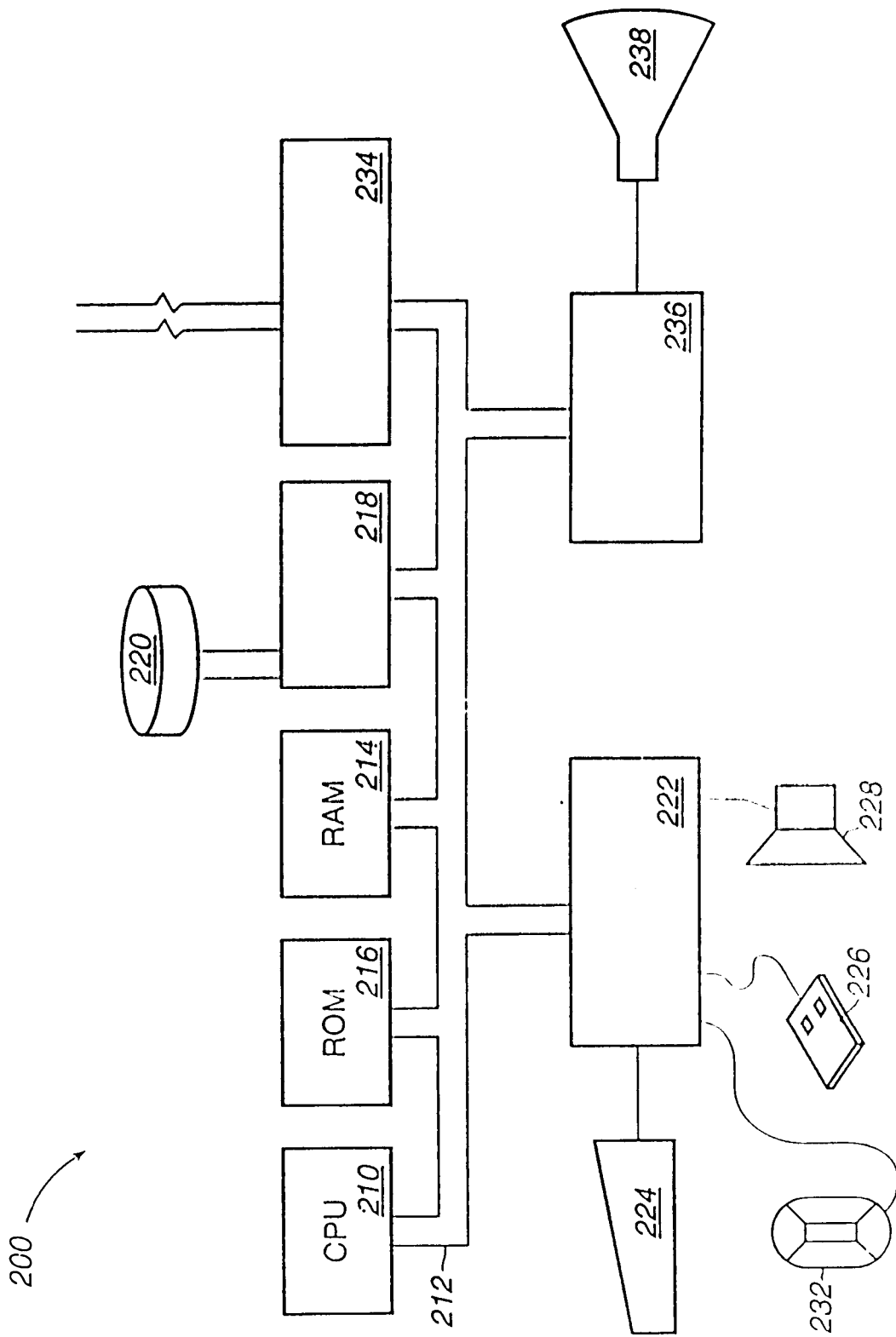
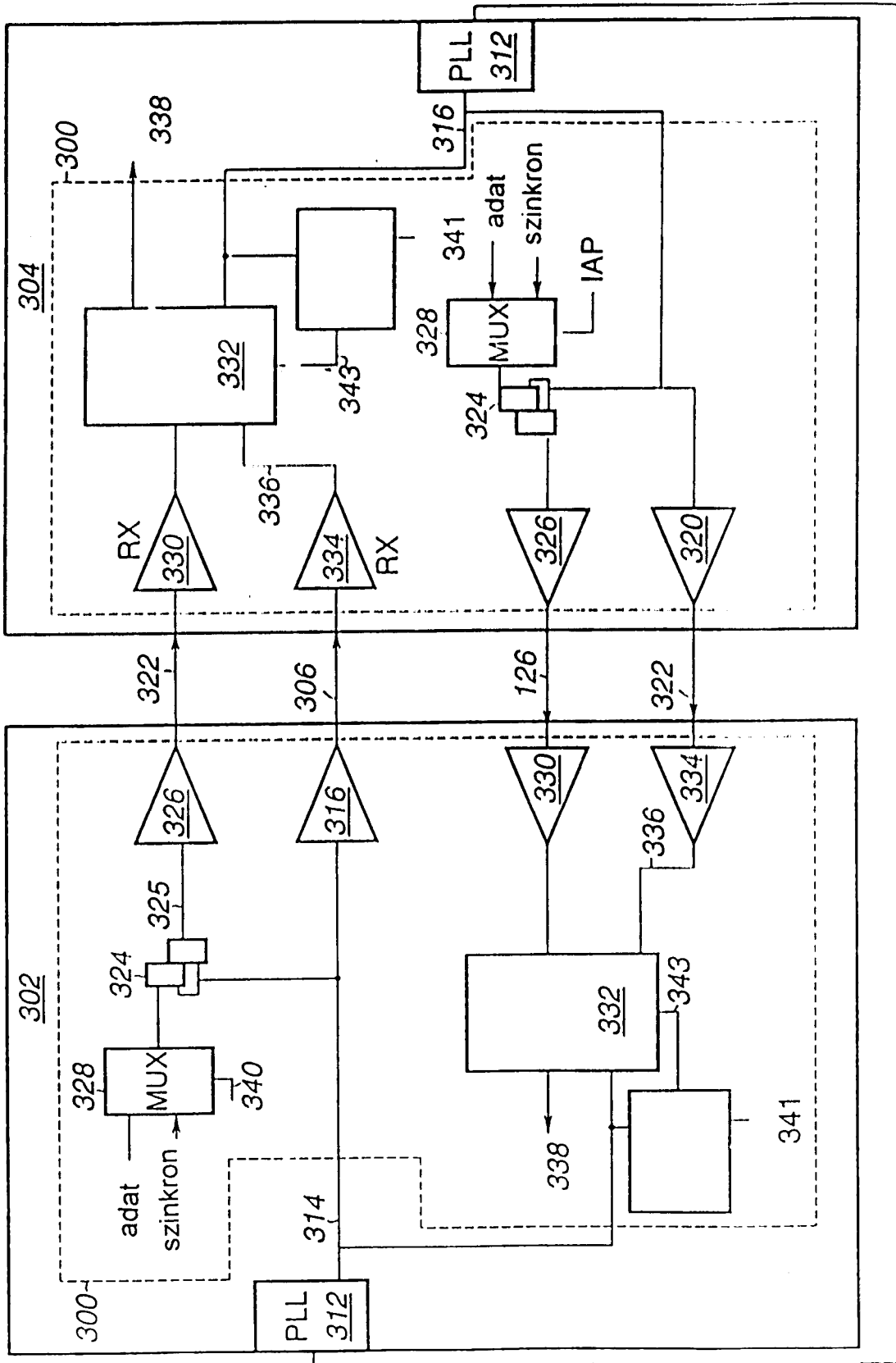


Fig. 2



~310

Fig. 3

KÖZZÉTÉTELI
PÉLDÁNY

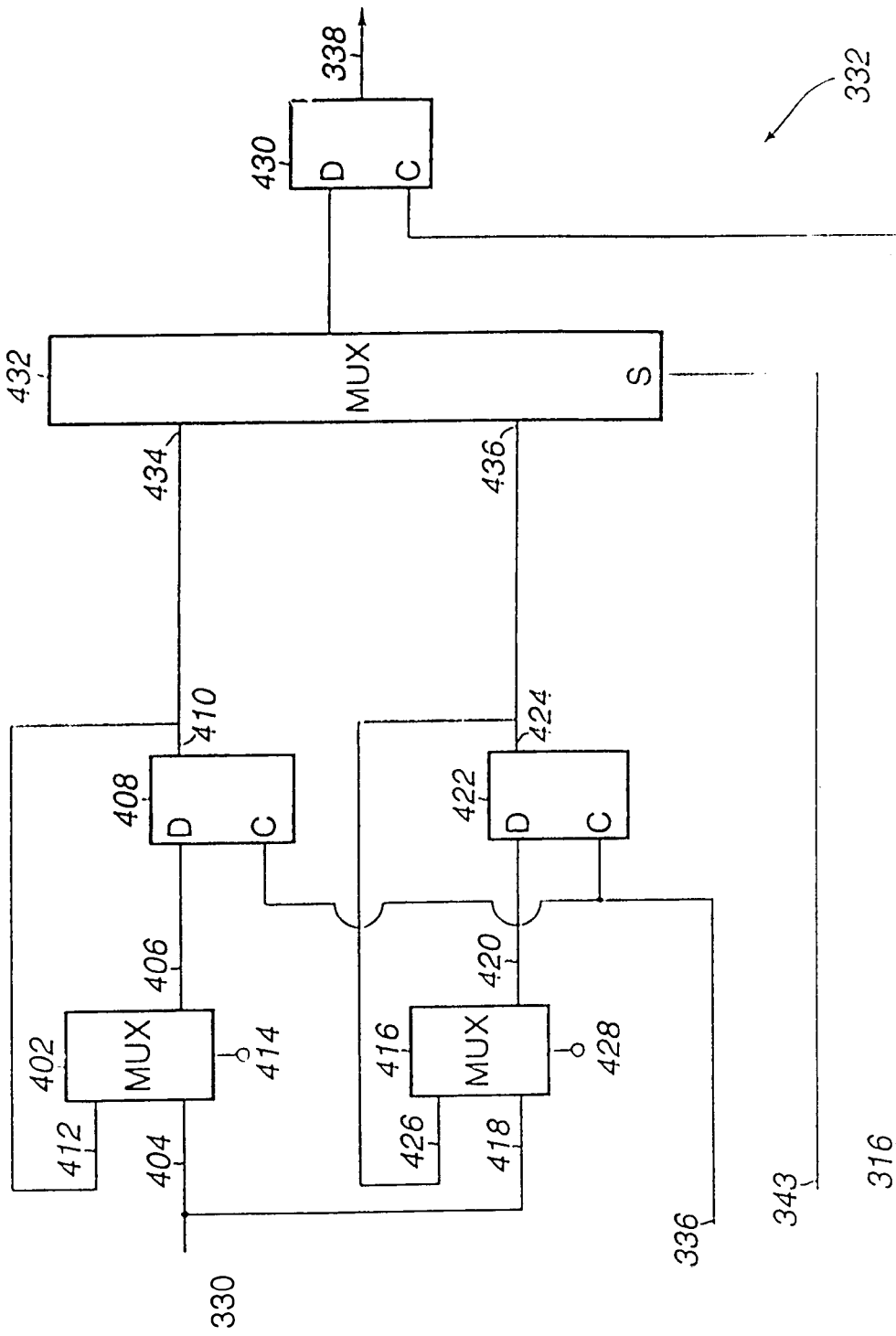


Fig. 4A

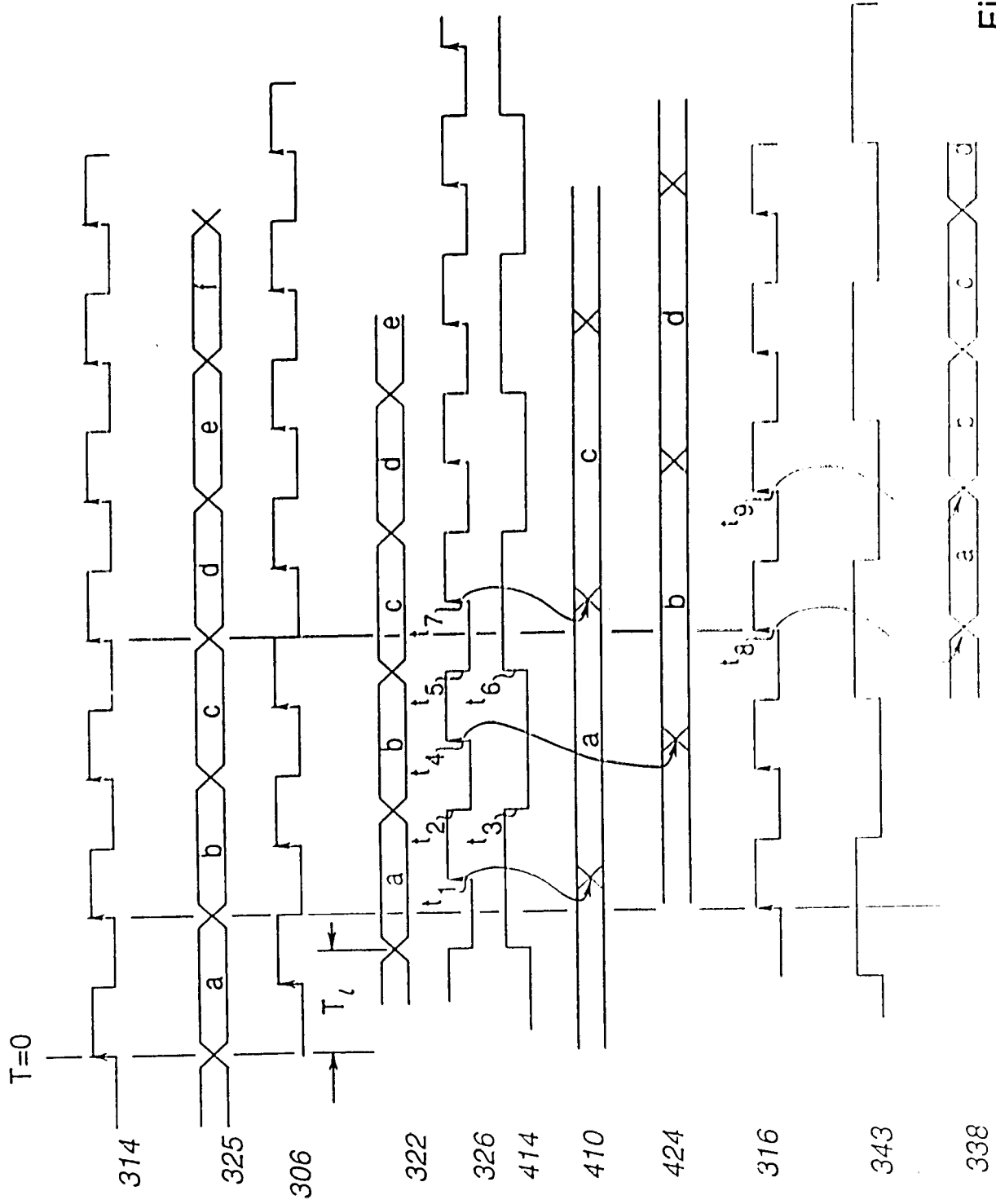


Fig. 4B

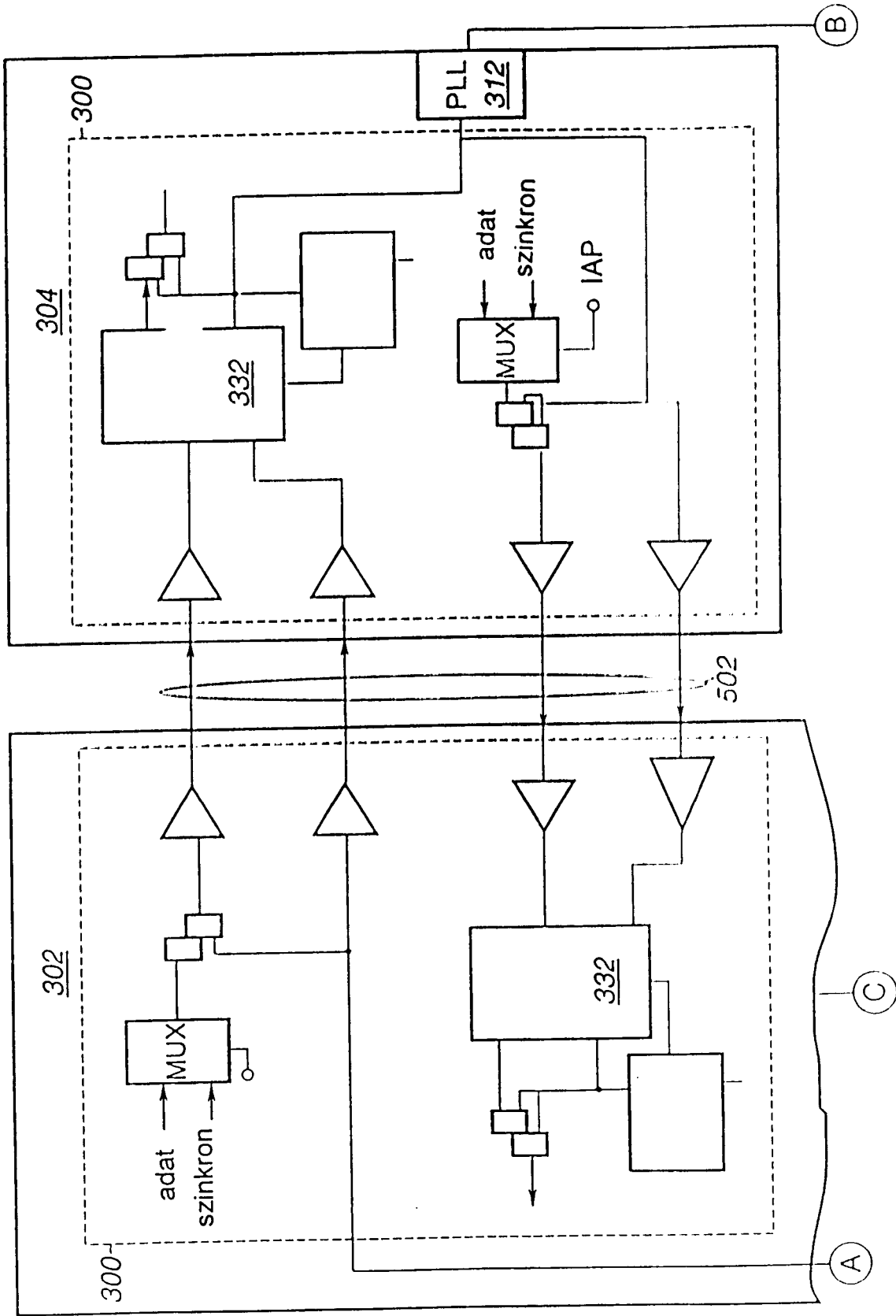


Fig. 5

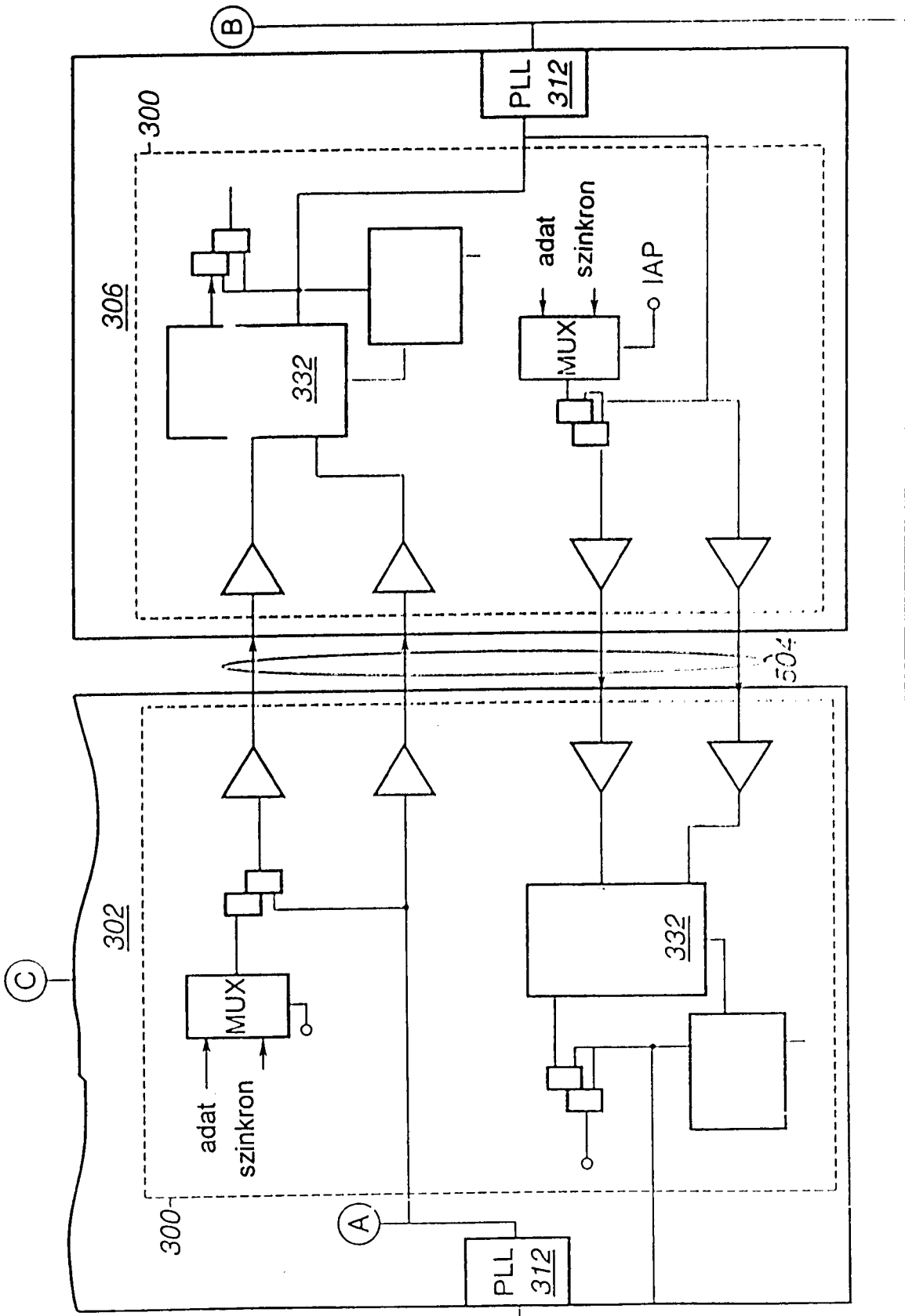


Fig. 5
(continued)

F-310

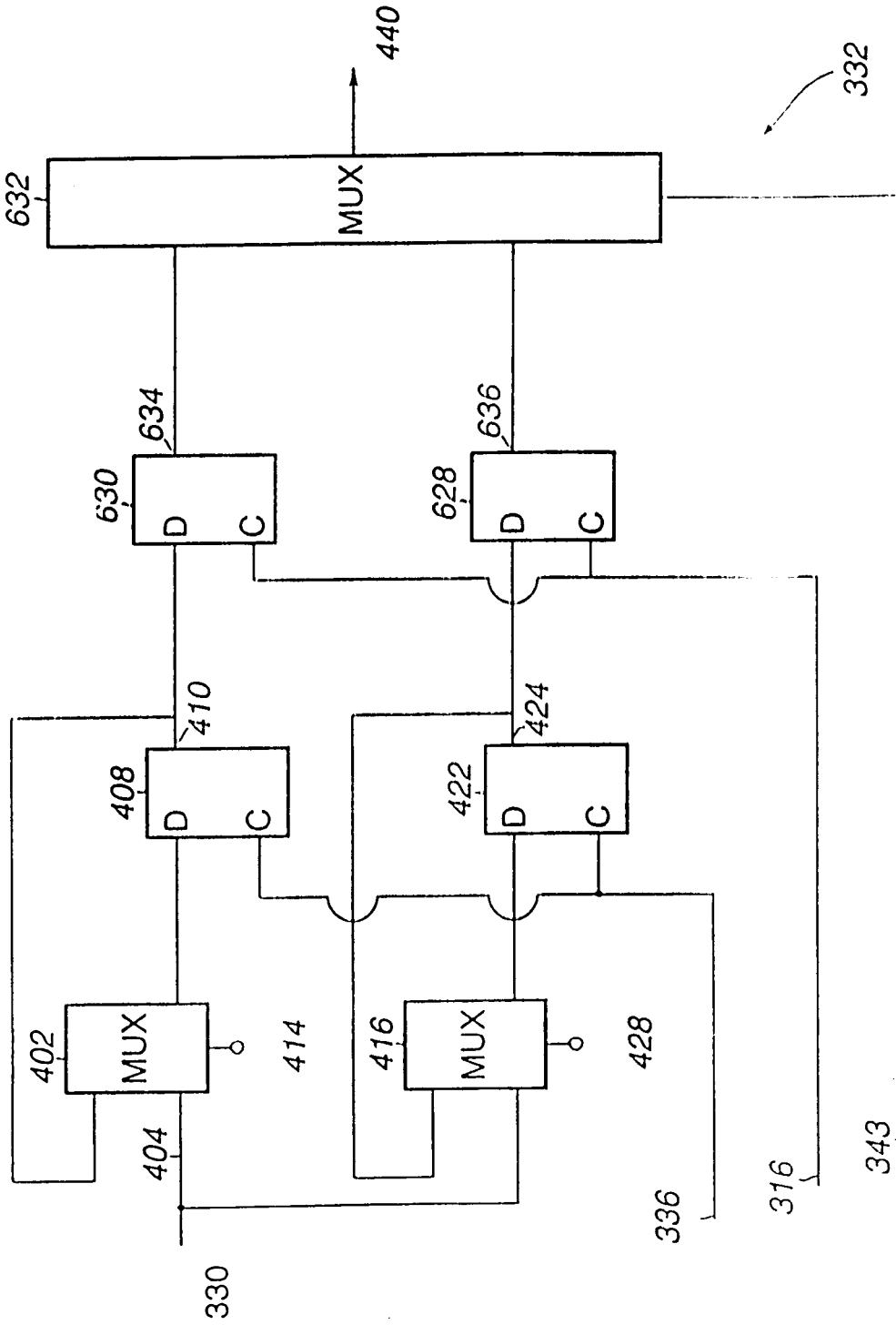


Fig. 6A

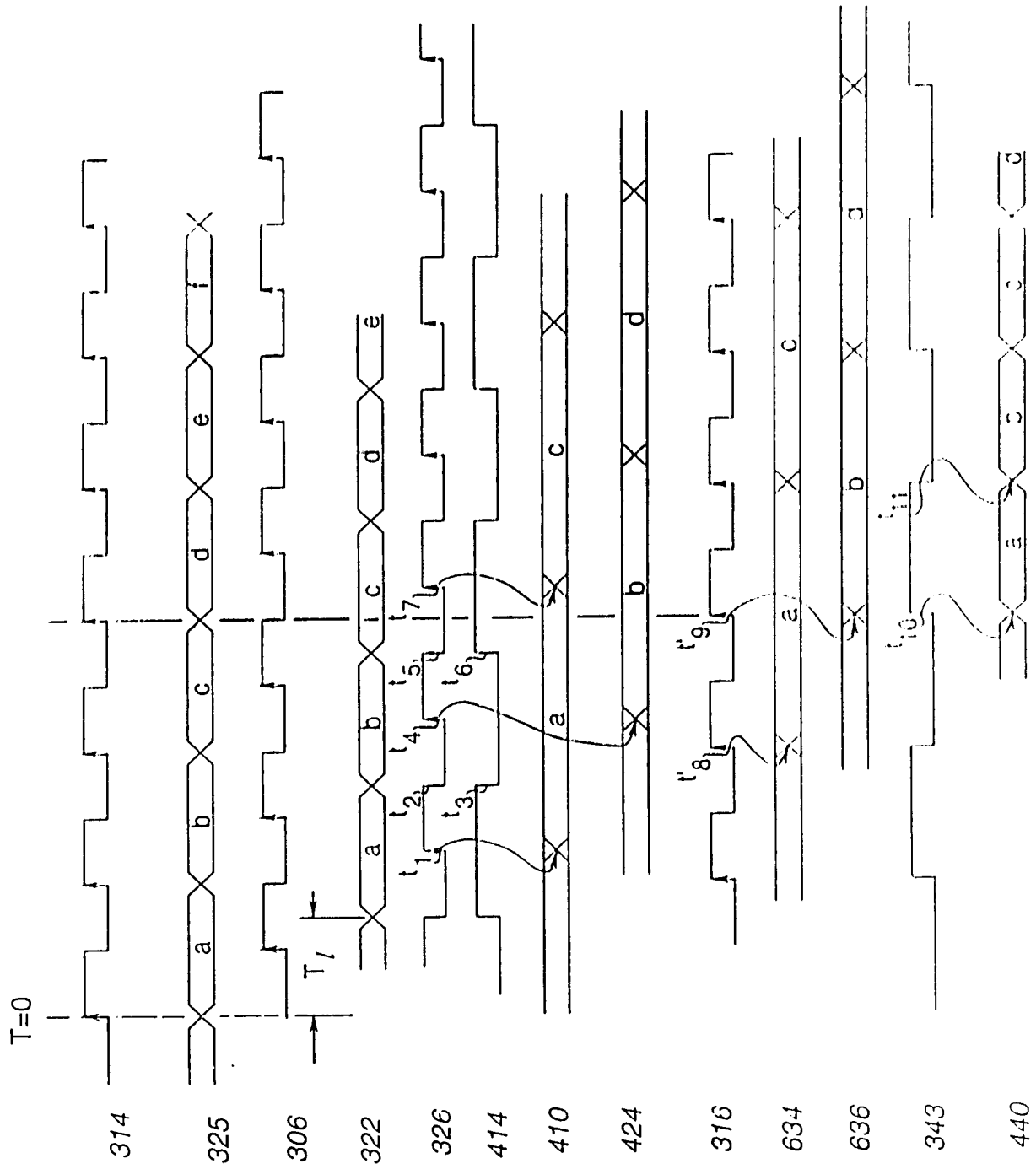


Fig. 6B

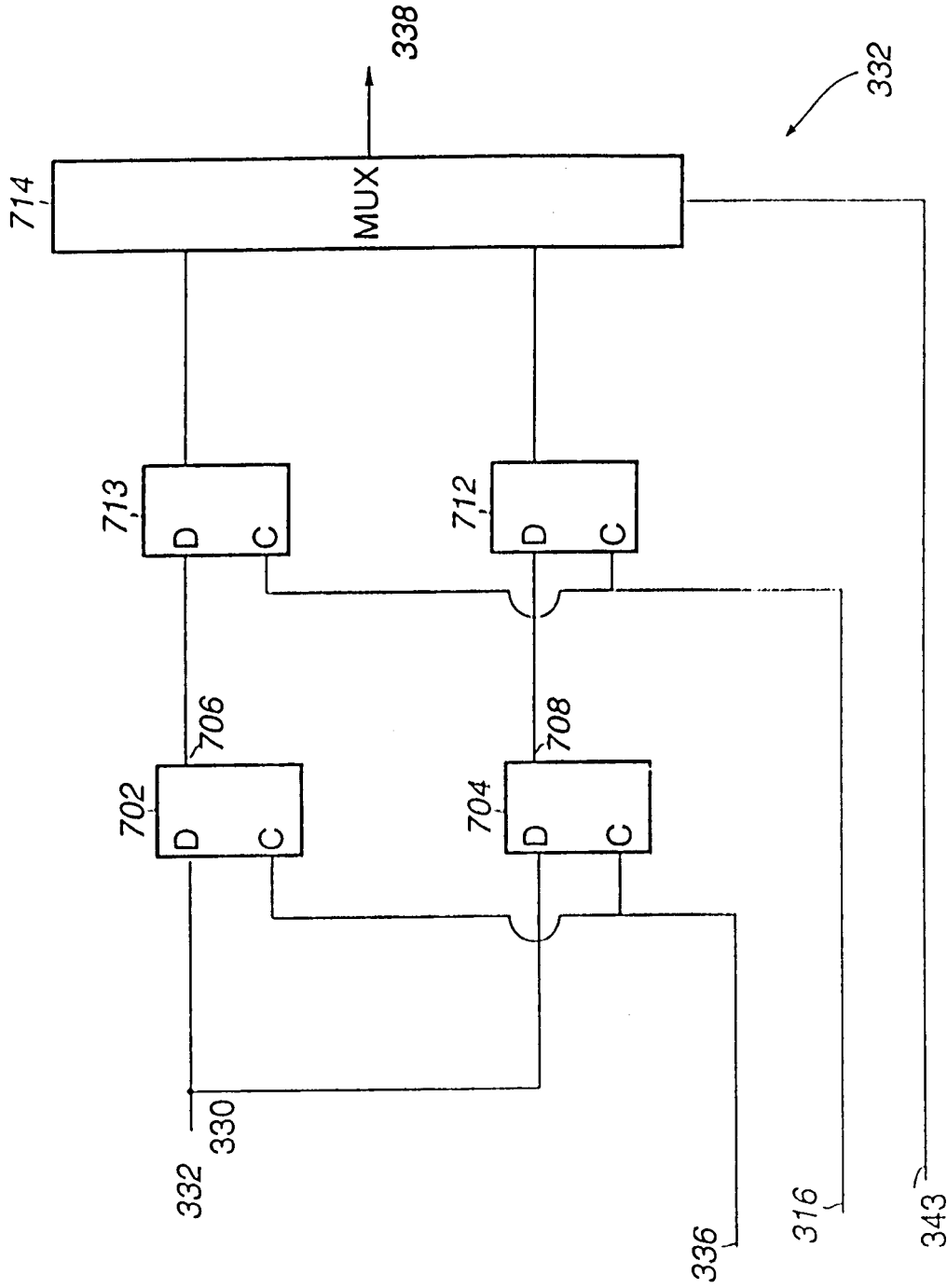


Fig. 7A

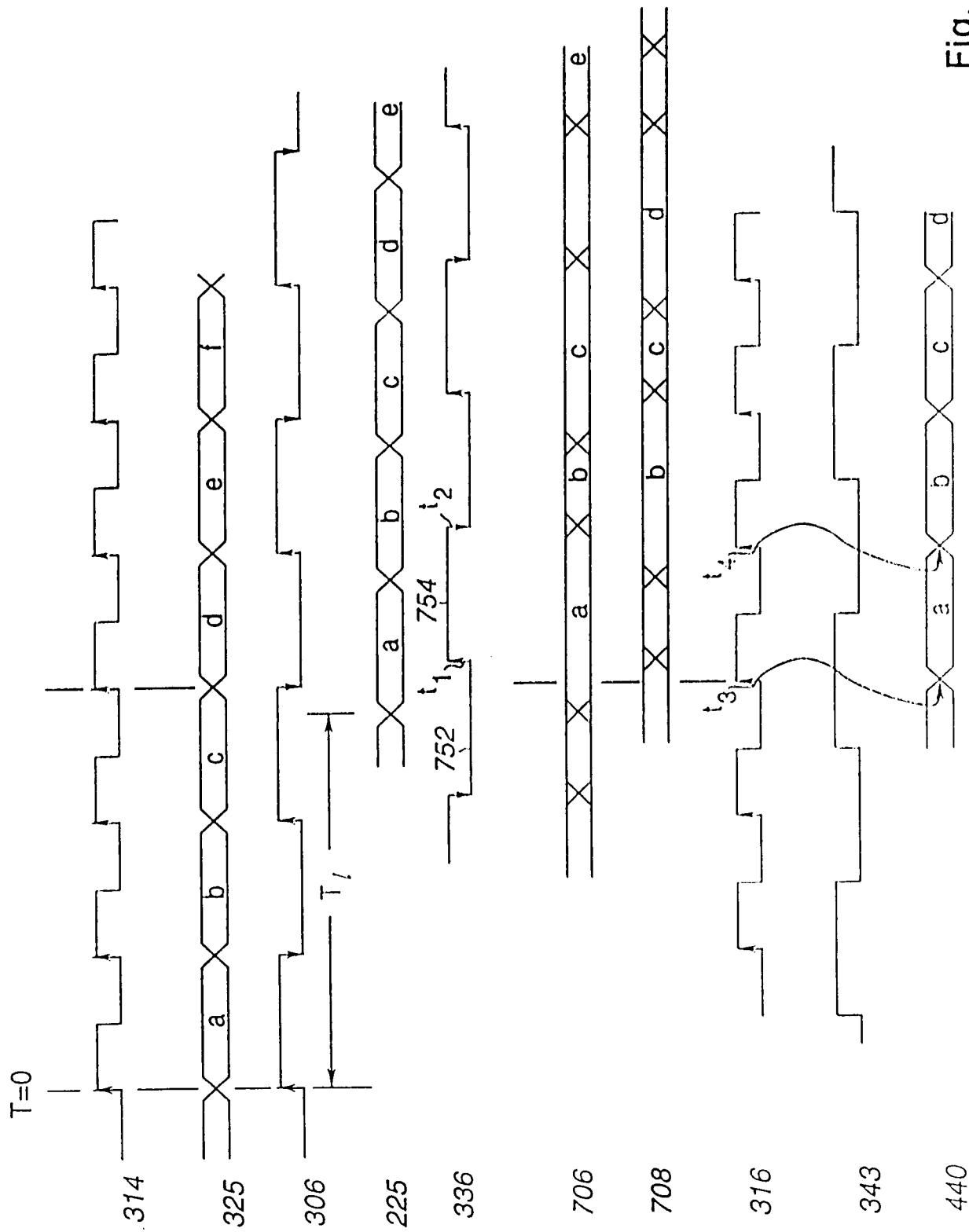


Fig. 7B

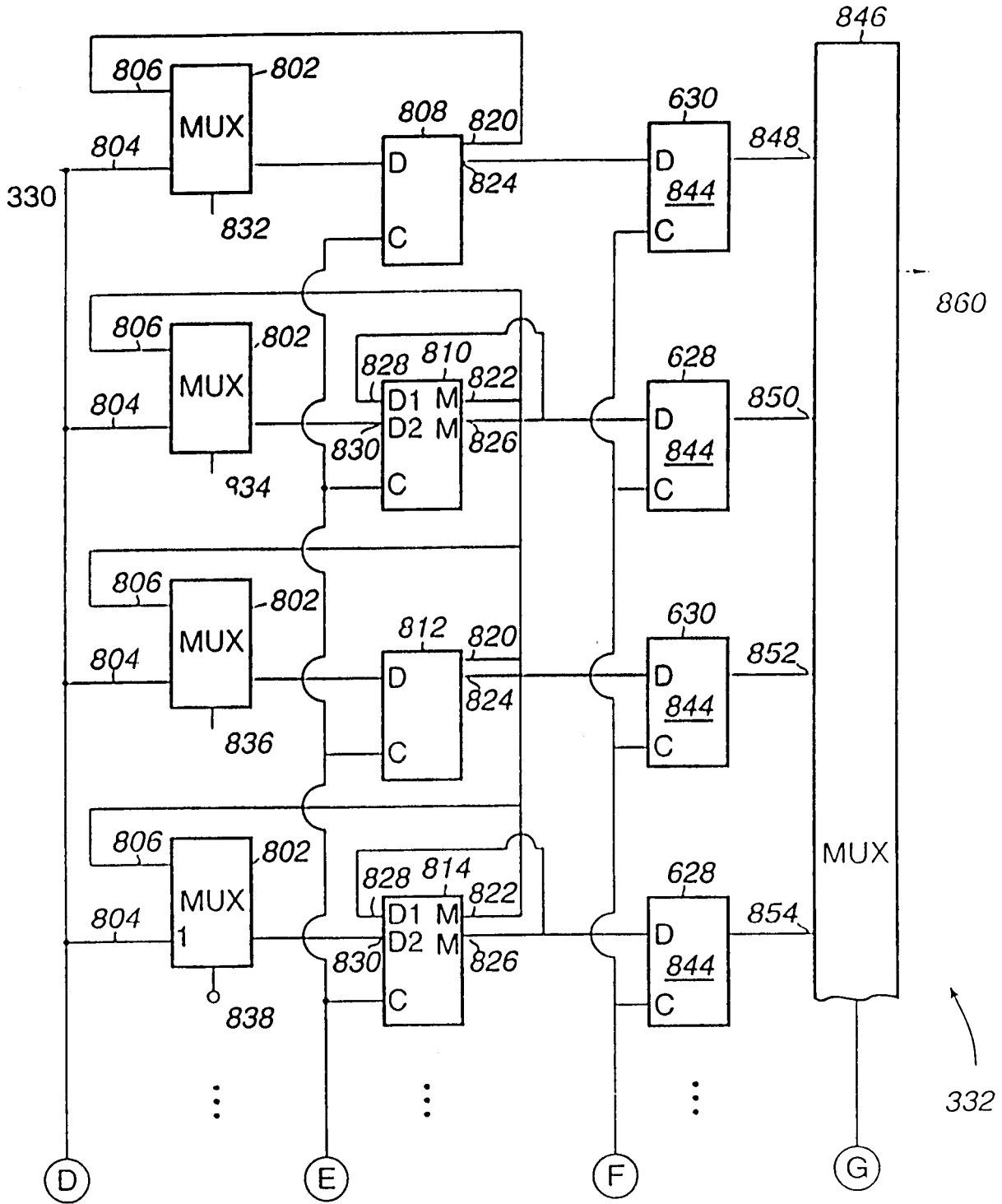


Fig. 8A

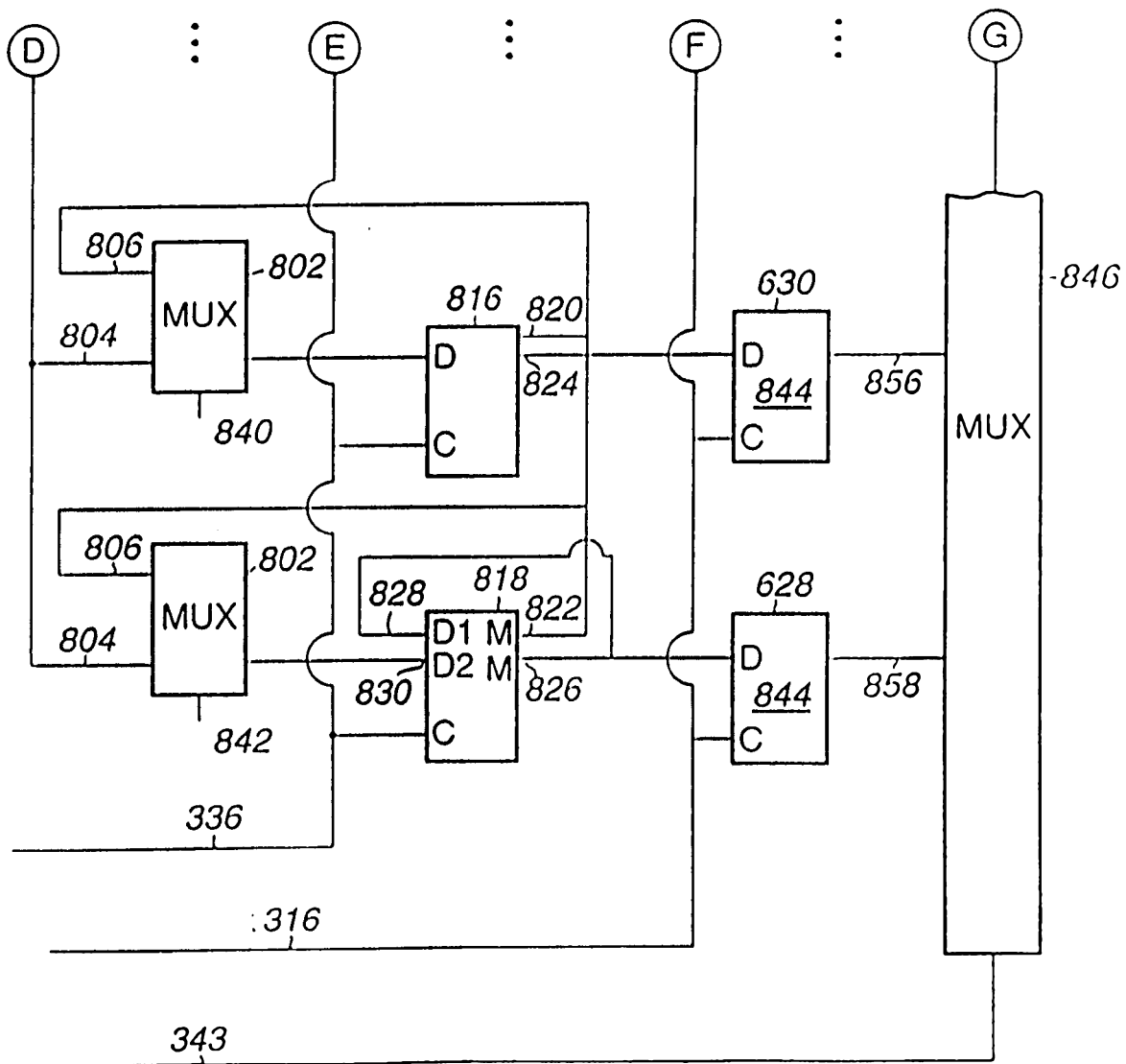


Fig. 8A
(continued)

