

公告本

321811

321811

申請日期	84.11.22
案 號	84112430
類 別	H04L29/02, H04M1/2

A4
C4

321811

Int. C16

(以上各欄由本局填註)

發明專利說明書

一、發明 名稱	中 文	安全接通電話分機系統與方法
	英 文	Secure Access Telephone Extension System and Method
二、發明 人	姓 名	(1) 大衛 B · 巴托羅紐 (2) 阿爾瑪 K · 蘇里 (3) A · 雷 · 艾維
	國 籍	(1) ~ (3) 美 國
	住、居所	(1) 美國猶他州西谷市克里大道5047號 (2) 美國猶他州奧倫市東北第600街472號 (3) 美國猶他州奧倫市西南第400街1174號
三、申請人	姓 名 (名稱)	美商 · 豐克斯公司
	國 籍	美 國
	住、居所 (事務所)	美國猶他州密德瓦勒市高科技道6952號
	代 表 人 姓 名	尼可拉斯 A · 史密斯

裝

訂

線

321811

(由本局填寫)

承辦人代碼：

A6

大類：

B6

I P C分類：

本案已向：

國(地區) 申請專利，申請日期：

案號：

，☐有 ☐無主張優先權

美

1993.03.22.

08/034,086

有關微生物已寄存於：

，寄存日期：

，寄存號碼：

(請先閱讀背面之注意事項再填寫本頁各欄)

裝

訂

線

經濟部中央標準局員工消費合作社印製

五、發明說明 (1)

I · 發明背景

A · 發明領域

本發明係有關於電力線路載波電流電話分機系統的領域。詳言之，本發明係有關於供線路載波電話分機系統用的安全系統控制協定，其提供了多分機仲裁邏輯體系、私密性、安全性、及其它電話發訊選項，此種技術稱為安全接達 (Secure Access, S A)。

B · 背景技術

已有人多方嘗試利用建築物之電力線路供電話通訊之用，且各種嘗試之成功程度各異。線路載波電話分機系統係利用：(1) 每一條中央局線有一個單一的基座單元（用以介接線路載波信號與中央局線路）；以及(2) 一或更個分機單元，每一個分機單元係供每一部分機電話機用（供介接該等線路載波信號與該等分機電話機）。全雙工通訊通常係使用兩種載波頻率來嘗試，每一方向用一種載波頻率。而通常在每一部基座台與分機台中亦包括有一個發射器與接收器，其在通訊功能正進行時係同時運作。基於經濟上的考量，常常係運用以聲頻（語音）信號對載波作 F M 調變的方式，美國專利第 3,949,172 及 4,701,945 號即為其實例。線路載波通訊系統的實體範圍經常被侷限在電力線路變壓器二次側的領域以內，且視載波頻率及負載狀況而定，會遭致嚴重雜訊與衰減量變異。

儘管雜訊與衰減問題可藉著提高發射器功率準位或增加接收器靈敏度略加克服，但這卻會增加與使用類似系統

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明(2)

的鄰居發生干擾的可能性，而如此會使習知技術的線路載波系統所顯現的私密性與安全性問題更為惡化。因此，當在同一電力配送系統上或是在有相同線路載波裝置的同一棟公寓大廈內的其它使用者，可以在另一使用者的線路上盜打長途電話及偷聽其通話內容時，即產生私密性與安全性的嚴重漏洞。但正在車內開車的人們則無法像他們用無線電話 R F 傳輸系統那樣地掃描及偷聽線路載波信號。

這些問題的衝擊已在無線電話工業中被證實，此種工業也擁有線路載波工業的一些相同限制。防止第三者盜用線路的安全性在無線電話工業中已有人用安全性編碼協定來提供之；但即便如此，擁有掃描器或無線電話接收機的人還是可以偷聽鄰居的無線電話通話內容。而且，過去也未有人提供使用同一基座的多部無線分機電話系統；同樣地，在線路載波系統中，過去也一直不允許一組以上的系統使用同一建築物中的相同電力線路。

本發明之主體內容係與本案發明人於1991年10月7日所申請之第07/773,009號美國專利申請案“多通接達電話分機系統與方法”這一件前案的主體內容略有關聯。此一前案主要係有關於諸如分頻多通(FDMA)及分碼多通(CDMA或展頻)等等射頻(RF)傳輸技術的一種結合體系；而本案則係有關於聲頻及次聲頻頻率系統控制邏輯鑰匙碼的運用，此等邏輯鑰匙碼解開電話信號發出功能、RF與語音傳輸功能，諸如獲取中央局線路、讓分機電話機振鈴、與在基座與諸分機單元間以全雙工通話等等。

五、發明說明 (3)

利用附有安全鑰匙碼之系統控制邏輯訊息，可以非同時地讓多個人使用同一電力配送系統，而仍具有免遭偷聽的私密性以及安全性。此系統控制邏輯亦防止正在進行的一通電話遭另一使用者加以中斷及終止。一些其它的多種分機仲裁邏輯功能被提供來讓一部分機可以把一通電話設於保留狀態並可把這通電話轉至另一部分機單元。使用 F D M A 技術配合安全性／控制鑰匙碼實現了以低成本提供真正多通接達及安全性的目的。

此種安全接達線路載波電話分機技術的商業應用領域包括有難以把電話線與絞線對裝配到既有建物結構內的付費收視 (pay-per-view, P P V) 系統以及瓦斯泵／安全辦公室系統等用的電腦數據機介面。

由於居住在共用電力線路之出租公寓及出售公寓的人數與日俱增，且因為須把傳輸功率與接收機靈敏度提高俾克服衰減及雜訊問題，故一套電話分機系統要是能以一種獨特而精湛的方式利用私密性與安全性編碼技術來容許進行可接受之操作，是十分重要的。

II · 發明概述

許多習知技術的限制均可如同本發明所運用者藉由一些安全接達技術的適當應用，而順利加以克服。安全接達 (S A) 技術包括了使用以聲頻 F S K 速率操作之一個數位資料傳輸頻道，此種數位資料傳輸頻道在基座與分機單元控制器之間交換訊息，而此等訊息係由一個同步位元組、一個獨特安全鑰匙碼、以及一個系統控制鑰匙碼所組成

五、發明說明(4)

。S A 技術需要在聲頻、R F、與系統控制路徑上加上只有藉由接收適當控制鑰匙碼方能加以連接(打開)的硬體“鎖”。訊息可經曼徹斯特(Manchester)編碼處理，以消除在調變與解調時加諸於訊息的任何純D C成份。而諸如曼徹斯特編碼法等等資料傳輸技術已描述於霍華W·山姆斯圖書公司1972年出版之I T T“無線電工程師參考資料”第五版中，是以該書之相關內容特併供本案參考。

在訊息之第一部分的一個同步位元組(8個位元)可讓接收控制器判斷某一訊息正被發送，及在諸鑰匙碼中伴隨之數位式的1與0在某些時槽中將為有效。因此，此同步位元組極迅速地將接收控制器中之一個數位鎖相迴路初始化，且有助於非同步通訊作業。對於此種通訊作業而言，重要的是為對大部分電話使用者服務成透通狀之線路載波分機系統，在振鈴命令方面頗快速地將脈衝撥號、及接聽／轉接鍵(flash)信號初始化。在另一方面，若把資料速率提升得過高，將會使資料易於因處理增益之損失而遭到雜訊與干擾破壞。較實在的資料速率大約為1至4KB。

安全鑰匙碼尾隨於同步位元組之後，且可具任何實用長度，諸如可提供65,536種碼的16個位元(2個位元組)。一旦接收到一個有效的安全碼，便啟動隨後的控制鑰匙碼。此等安全碼可由控制器隨機選取，或是由使用者可取用之開關作人工選取。針對基座與分機單元提供有一套安全碼匹配程序，以共用同一個碼，以及因而可以接受彼此

五、發明說明(5)

之控制鑰匙碼。

控制鑰匙碼尾隨於安全碼之後，且可為1個或2個位元組長，視系統控制邏輯所要求之複雜度而定。例如，假如第一個位元組為“脈衝”命令，且第二個位元組指定該脈衝之期間長短，則一個脈衝命令可能需要2個位元組，因而區分脈衝撥號脈衝與供通話等待用的“轉接鍵”脈衝等等。職是之故，諸電話控制信號會依控制鑰匙碼之請求被基座或分機控制器重新合成，以供透通作業之用。

為完成安全接達所需之系統需求，一些硬體鎖（即一些受數位式控制之類比開關）必須被設置於在接收到適當安全性暨控制鑰匙碼時被系統控制器所操作的所有基本電話信號與控制路徑中。

由於習知技術的線路載波電話分機系統並未結合安全接達技術之諸基本元件，故它們自不能以一種可接受的商用方式運作。

緣是，本發明的一個目的即是要提供安全接達（S A）通訊的一種有效方法，其可在單一個通訊媒體上提供接達多個信號的機能。

另一個目的是要提供一種線路載波電話通訊方法與系統，其運用S A（安全接達）來提供高度安全性以防止他人未經許可地取用用戶線路，且其在通話方面可提供隔絕第三者的私密性，又其可在諸分機電話機與中央局線路之間提供近乎透通之互連功能。

再一個目的是要提供一種多分機仲裁邏輯方法與系統

（請先閱讀背面之注意事項再填寫本頁）

裝

訂

結

五、發明說明(6)

，俾為同一用戶線路提供不會相互干擾的多部分機。

又一個目的是要提供一種線路載波電話通訊方法與系統，其利用 F D M A (分頻多通) 配合 S A (安全接達) 來防止相當靠近之相鄰傳輸系統間或是在同一系統內之同組傳輸作業之間的相互干擾，以及為至少兩條電話線路提供雙工信號之多通接達 (同時傳輸)。

更一個目的是要提供一種安全接達無線電話通訊方法與系統，其應用相同的安全接達技術，來獲取和線路載波電話分機系統與方法相同的優點。

本發明之系統與方法的此等以及其它目的和優點，將可在配合附圖閱讀下文之詳細說明時變得更為明顯。

Ⅲ · 圖式簡介

第 1 圖係為服務一條單一用戶線路與多部分機電話機之一組線路載波電話分機系統的一幅方塊圖。

第 2 圖係為在具有一套共用 A C 電力配送系統的一棟公寓大樓中服務多條用戶線路與多部分機電話機之一組線路載波電話分機系統的一幅方塊圖。

第 3 圖係為供一個基座單元系統控制器用之一幅狀態圖。

第 4 圖係為供一個分機單元系統控制器用之一幅狀態圖。

第 5 圖係為一套安全接達線路載波電話分機系統之一個基座單元的一幅電氣方塊圖，此基座單元介接一條用戶線路與一棟建築物之 A C 電力配送系統。

五、發明說明(7)

第 6 圖係為一套安全接達線路載波電話分機系統之一個分機單元的一幅電氣方塊圖，此分機單元介接一部分機電話機到一棟建築物之 A C 電力配送系統上。

IV · 較佳實施例詳述

首先請參見第 1 圖中所繪示之一套線路載波電話分機系統，此套線路載波電話分機系統負責服務一條單一用戶線路 1、以及多部分機電話機 6 與 9。用戶線路 1 插入基座單元 2 內，此基座單元 2 提供諸用戶線路信號（包括振鈴、接聽、及雙工聲頻信號等）所需、以及經諸 A C 電力線路 3 載送通過一棟建築物而到達分機介面單元 4 與 7 之諸安全接達電力線路載波信號所需的介面電路和協定。分機單元 4 與 7 係為單一的線路分機，且提供了將安全接達線路載波信號分別經由電話線 5 與 8 介接至各單一線路電話機 6 與 9 的裝置。

如第 1 圖所繪示者的一套線路載波系統必須以對分機電話機使用者而言呈透通的一種方式來運作。因此，對於分機電話機 6 而言，藉由基座單元 2、A C 電力線路 3、與分機單元 4，一個來話振鈴信號即在基座單元中被檢測、被編碼成安全接達線路載波信號、經電力線路傳送、在分機單元中被解碼、及被轉換為同等長度的一個振鈴信號。當一位使用者拿起一部分機電話機之受話器（話筒）時，該部分機電話機即呈接聽狀態，此狀態在分機單元 4 中被檢知、被該分機單元 4 編碼成安全接達線路載波信號、並經由諸條電力線路 3 而被傳送到基座單元 2，該等安全

（請先閱讀背面之注意事項再填寫本頁）

裝

訂

結

五、發明說明(8)

接達線路載波信號即在此基座單元 2 中被解碼，且用戶線路便被接聽電路取用。雙工聲頻電路被解鎖且被連接到線路上，此時語音及／或撥號音與 D T M F 之雙向傳輸作用即在由諸語音信號調變線路載波信號之情形下進行，此等線路載波信號經由諸條電力線路 3 傳送並在基座與分機單元中被解碼／解調回去成為語音。此處所牽涉到之特定子系統及對應功能將會在下文參照第 5 與 6 圖予以說明。

第 1 圖亦可用於闡述多分機仲裁邏輯。例如，當某一中央局透過用戶線路 1 發送一個振鈴信號進入基座單元 2 時，此基座單元 2 即讓此振鈴信號具有合適的頻率（15 至 110 Hz）以及振幅（40 Vrms min.），並發送啓始與停止振鈴訊息至二分機單元 4 與 7，此等振鈴訊息被檢知及被轉換為標準振鈴信號，然後這些標準振鈴信號即會以與原始中央局振鈴信號之韻律大致相同之韻律使二具電話機 6 與 9 發出鈴聲。假使有一位使用者以電話機 # 2（9）接聽，此一接聽動作在對應的分機單元 # A 2（7）中被檢知，此分機單元 # A 2（7）便發送一個接聽請求訊息到基座單元 2，而此基座單元 2 則：(1) 將安全碼與接聽訊息適格化、(2) 獲取用戶線路、以及 (3) 發送一個接聽確認訊息到二分機單元，此一接聽確認訊息只允許分機單元 # A 2（7）在它自己 7 和電話機 # 2（9）之間連接聲頻與發信電路，而使該使用者得以進行一般的電話通訊。而於電話機 # 2（9）正使用時，若有在電話機 # 1（6）處之一位第二位使用者把他的受話器拿起變成接聽狀態，分機

（請先閱讀背面之注意事項再填寫本頁）

裝

訂

五、發明說明(9)

單元 # A 1 (4) 中之系統控制邏輯電路即會針對此電話機 # 1 (6) 產生一個忙線音，且不允許此第二位使用者聽到或插入正進行中之通話內容，因為系統控制邏輯電路察覺到這位第二位使用者並非第一個作出接聽請求者，且其電話機在第一個接聽確認訊息被接收到時仍處於掛斷狀態。然而，如果該第一位使用者於此時在電話機 # 1 (6) 之第二位使用者拿起話筒時把電話機 # 2 (9) 掛斷，分機單元 # A 2 (7) 將發送一個切斷電話訊息至基座單元 2，此基座單元 2 即回送一個“未確認”訊息至所有的分機單元（但不會之即切斷電話）；分機單元 # A 1 (4) 一接到此未確認訊息，即發送一個接聽請求訊息至基座單元 2，並在收到此基座單元傳回之一個確認訊息時接上聲頻電路。該基座單元中之邏輯電路會在出現一個切斷電話訊息之後暫時只在一段充份時間內不切斷連至中央局之通路，以容許第二部分機在出現來自基座單元之未確認訊息之後請求使用該線路。

多分機仲裁邏輯容許使用者藉著快速按壓其分機電話機 # 1 或 2 上之開關鉤鍵兩次，而把一通電話置於保留狀態；而於對應之分機單元 4 或 7 中產生之一個音調會確認此保留請求狀態。分機單元發送一個保留訊息到基座單元 2，因此，基座單元即：(1) 在仍然抓住線路之情形下令用戶線路聲頻成為靜音狀態，以及(2) 每隔 45 秒鐘左右對所有分機單元發送一組 3 個短振鈴韻律之訊息共經 10 個周期，隨後，此基座單元再產生振鈴兩次，並且若使用者仍未接

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

終

五、發明說明 (10)

聽電話，即在切斷線路之前再等待最後20秒鐘。於此段時間內，包括硬佈線之電話機在內的任何分機電話機可由使用者拿起其受話器使其成為接聽狀態，並發信號通知處於呆留狀態之基座單元解除保留狀態；電話機#1或2發出接聽請求至該基座單元，使此基座單元把聲頻路徑解鎖；而若有一部硬佈線電話機之話筒被拿起成接聽狀態，基座單元即檢知用戶線路電壓上之壓降，並把線路抓取使用權釋出給予該具硬佈線電話機。此系統控制邏輯已在第3與4圖中更詳盡地示出。

第2圖繪示出安全接達與多通接達技術的結合應用設計，其中有數棟公寓共用一套共同AC電力配送系統且配有多條用戶電話線路以及線路載波分機延伸線。由於所有的“A”單元均使用相同的線路載波頻率，故即須用安全接達技術來提供公寓#1線路1與公寓#2線路15之間的私密性及安全性。公寓#1中的使用者可藉著先在基座單元#A1(2)上按下一個按鈕，及接著在一分鐘內在分機單元#A1(4)與#A2(7)上按下對應按鈕，而在其單元內調配密碼。按下按鈕之動作把諸單元置入學習模式，此時諸分機單元送出一個請求來要求基座單元中之隨機選定安全鑰匙碼，基座單元即把此鑰匙碼送回諸分機單元。因為公寓#2內的使用者並不知公寓#1內的使用者究竟何時會調配其安全碼，故其單元A3即會有一個不同密碼。如此便以一種非同時的原則提供了讓一些AC電力線路被數個A頻率單元作多通使用的機能。基座單元#

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

結

五、發明說明(11)

B (11) 及其相對應分機單元 # B (12) 使用另一組不同的多通接達傳輸頻率，此即為線路 # 2 (10) 提供了真正同時使用 A C 線路媒體之能力。線路 2 可供數據機、傳真機、付費收視電視服務、或其它電話裝置利用。

第 3 圖與第 4 圖提供了除了電源開啓程序以外供上文針對第 1 與 2 圖所揭示之諸系統控制協定用的一套邏輯狀態圖。若在一“電源開啓”重新設定程序期間，某一單元檢查非依電性記憶體中是否有一組有效的安全碼且並未發現有任何有效的安全碼時，此單元接著便提供一個內定安全碼；反之，若發現有一組有效的安全碼，此單元即採用記憶體內所保有的該組安全碼。而由於諸單元上並未設置開啓／關閉 (on/off) 之開關，所以當 A C 線路之電力開啓時，諸單元便跟著開啓電源。倘若發生了電源故障或某一單元未插接妥當之狀況，當時所用之安全碼即被儲存於非依電性記憶體中，是以當電力恢復時，該安全碼即維持不變。諸如摩托羅拉 6805 系列或奇洛格 (Zilog) Z 8 系列的任何數目之商用低價位微控制器均可被用於此一應用中。如同本發明之一實施例中所用者的供此應用所用之工作商用程式碼的一份列表，已列示於文末之附錄 A 中，其內容特併供本文參考；而由於其內容已可為熟於相關技藝人士所了解，故在此不再予以贅述。除此以外，亦可採用其它程式碼，且仍在本發明之範圍以內。

第 5 圖係為取自第 1 與 2 圖所示系統中之一個安全接達基座單元 2 的一套階層體系概示圖的主頁。依據階層體

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

終

五、發明說明 (12)

系概示程序，其中之方塊5-2至5-12等方塊的完整電氣示意圖已個別繪示於文末所附之附錄B中；故附錄B之內容乃併供本文參考；而由於其內容俱可為熟於此技者了解，並不再予以贅述。尚有一些可供具體實施本案發明技術內容之其它電氣概要架構亦可採用，且意欲包含於本發明之範圍以內。電話線路1被連接於COLIC 5-2，而AC電力線路3則與電源供應器5-12介接；特別要請注意微控制器5-10以及其伴隨之I/O線路20至29。"unreg" 20輸入提供第3與4圖之電源開啓／關閉重新設定信號。在電源開啓時，要注意的是，要在電力供應與時鐘29穩定之後才啓動控制器；而在電源關閉時，須在時鐘與電力供應變成不穩定之前停止控制器。來自（諸）分機4之資料透過方塊5-12、5-7、5-8、及5-11接收和鑑別，並經由線路21進入微控制器5-10，此等資料即在微控制器5-10中被認證和分類成一些系統控制指令。對於在線路1上之硬佈線電話機而言，其它的硬體輸入尚包括有電話線路1振鈴信號檢測埠24、與接聽狀態檢測埠25。微控制器5-10之邏輯電路依據此等輸入作動，以提供適當之系統控制回應動作。在埠24之振鈴信號於資料輸出埠27產生振鈴訊息，此等振鈴訊息經由方塊5-5、5-6、5-12、與線路3而被傳輸至（諸）分機單元。當某一分機單元透過埠21作出一個中央局請求時，基座控制器5-10即透過控制埠25把線路1取為接聽狀態、發送一個確認訊息回到該分機單元、透過控制埠26與28取消發射與接收聲頻之靜音狀態，且該分機單元便可與

（請先閱讀背面之注意事項再填寫本頁）

裝

訂

裝

五、發明說明 (13)

一位來話者進行通話、或是在聽見一個中央局撥號音調時發送 D T M F 信號或脈衝撥號訊息俾撥出一個電話號碼。脈衝撥號訊息係透過控制埠 22 加以處理，以求與中央局有最佳之相容性。如同在第 5 圖中已標示者的其它聲頻與 R F 信號之流程，將可為熟於此技者由圖中清楚看出，故不再贅述之。

第 6 圖係為一個安全接達分機單元 4 (或 7、12、或 17) 的一套階層體系概示圖的主頁。依據階層體系概示程序，其中之方塊 6-2 至 6-12 等方塊的完整電氣示意圖已個別繪示於附錄 B 中。由於其內容俱可為熟於此技者了解，並不再予以贅述。電話機線 5 被連接於 SLIC 6-2，而 A C 電力線路 3 則與電源供應器 6-12 介接；特別要請注意微控制器 6-10 以及其伴隨之 I / O 線路 30 至 39。unreg 埠 30 輸入提供第 3 與 4 圖之電源開啓／關閉重新設定信號。在電源開啓時，要注意的是，要在電力供應與時鐘 39 穩定之後才啓動控制器；而在電源關閉時，須在時鐘與電力供應變成不穩定之前停止控制器。來自基座 2 之資料透過方塊 6-12、6-7、6-8、及 6-11 接收和鑑別，並經由線路 31 進入微控制器 6-10，此等資料即在微控制器 6-10 中被認證和分類成一些系統控制指令。其它的硬體輸入尚包括有分機電話機 6 之接聽狀態檢測埠 33。微控制器 6-10 之邏輯電路依據此等輸入作動，以提供適當之系統控制回應動作。來自基座單元之振鈴訊息被認證並透過埠 34 送至 SLIC 6-2，而此 SLIC 6-2 便據此而令分機電話機 6 振鈴。當某一使用者把

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

終

五、發明說明(14)

分機電話機受話器舉離鈎鍵成為接聽狀態時，此一指示在埠33使控制器在埠34切掉高電壓振鈴、檢查系統之狀態以確定有無其它分機正在使用中，且假如全部均經清除，即經由埠38打開發射器、以訊息通知基座單元2可取用中央局線路1、接收及認證透過埠31送回的一個“C0確認”訊息、透過埠36取消諸聲頻路徑之靜音狀態，且該分機單元便可與一位來話者進行通話、或是在聽見一個中央局撥號音調時發送出DTMF信號或脈衝撥號訊息俾撥出一個電話號碼。分機單元亦可向被連接來取代電話機6的某些電話附屬設備，藉由埠35以及在電話機線5上產生一個較低線路電壓的SLIC 6-2中之硬體，指出線路1正被硬佈線電話機使用中。如同在第6圖中已標示者的其它聲頻與RF信號之流程，將可為熟於此技者由圖中清楚看出，不再贅述之。

由本文先前所呈現之論述中，可輕易看出第1至6圖中所繪示之該系統可有許多項變化。其中一項明顯可見的變化是，把本文所述之安全接達技術應用在一套無線分機電話系統上，而其中須以一種RF載波取代該電力線路載波線，並以一支天線取代該電力線路介面。

另一種變化則是以一種CVSD（連續可變斜率三角調變器）及FSK調變術取代類比音幅縮伸器（compandor）與FM調變技術。一個CVSD調變器亦可被視為是具有音幅縮伸力的一個單一位元A/D轉換器；而這麼一個CVSD需要在發射器與接收器上用到16kB至32kB的一個

（請先閱讀背面之注意事項再填寫本頁）

裝

訂

五、發明說明 (15)

F S K 速率，這對某些商用接收器晶片已達實用階段。而其中藉由提供加密技術，便可為數位資料傳輸功能提供所要求之安全性。

有許多接收器晶片也提供有 F M 解多工能力，這為聲頻及控制發信工作提供了數項額外途徑。例如，與下旁帶聲頻同時地，安全接達控制發信動作可在一個上旁帶上傳送。如此可藉著在某些發信操作動作期間不把聲頻設定於靜音狀態，而增強其作業之透通性。此外，運用一種副聲頻調變技術，則可實現一條獨立的資料頻道。

某些變化型態可針對安全接達訊息結構及其同步技術、系統控制指令、與安全碼結構／長度施行。由於發射器與接收器之間呈頻率鎖定關係，是以同步資料傳輸技術在此亦屬可行，而可簡化資料鑑別作業。

儘管上文已針對若干特定實施例描述及闡示了本發明，但熟於此技者將可體認到，有許多種變化與修正均可在不悖離本案所闡示、描述、及請求之本發明技術原理之前提下為之。

本發明亦可採用其它特定型式予以具體實施，而仍不會偏離其精神或基本特性。上文所描述之諸實施例在各方面均應被視為只供例示說明之用，而毫無限制意味。職是之故，本發明之範圍係由後附之申請專利範圍所指出，而非由前文之說明指定。落於此等申請專利範圍請求項之意義與等效範圍內之所有變化，均將被涵蓋於其範疇內。

五、發明說明(16)

專 有 名 詞 列 表

1、10、15	用戶線路
2、16	基座單元
3	AC電力線路
4、7、12、17	分機單元
5、8、13、18	電話線
6、9、14、19	分機電話機
5-2	COLIC
5-3、6-3	混成部
5-4、6-4	壓縮器
5-5、6-5	發射振盪器
5-6、6-6	發射器
5-7、6-7	接收器
5-8、6-8	聲頻濾波器
5-9、6-9	擴展器
5-10、6-10	微控制器
5-11、6-11	資料鑑別器
5-12、6-12	電源供應器
20~39	微控制器輸出／入控制埠
6-2	SLIC

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

裝

五、發明說明 (17)

附 錄 A

安全接達軟體列表

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

系

五、發明說明 (18)

```
;Intelejak Z86C08 control software (c) 1993 Phonex Corporation
;Software Revision:
AAitjlv0 ;Released on 2/3/93
.z8regdefs
```

```
;----- Hardware assignments for the I/O ports -----
;..... PORT 0 .....
muteo .equ 00000001b ;p00
msgfo .equ 00000010b ;p01
ledon .equ 00000100b ;p02
bp0dad = 00000101b ;base p0 data default
ep0dad = 00000101b ;ext p0 data default
```

```
;..... PORT 2 .....
boffhooko .equ 00000001b ;p20
eppvon .equ 00000001b ;p20
bpulseo .equ 00000010b ;p21
exmiton .equ 00000010b ;p21
pshbtnti .equ 00000100b ;p22
bringi .equ 00001000b ;p23
ehvon .equ 00001000b ;p23
msgo .equ 00010000b ;p24 (is sometimes an input)
bsei .equ 00100000b ;p25
testi .equ 01000000b ;p26
stopi .equ 10000000b ;p27
; (direction controls: 0 = output; 1 = input)
bp2ddd = 1111100b ;data direction default for the base
bp2dad = 00000010b ;default data for port 2 base
ep2ddd = 1110100b ;data direction default for the extension
ep2dad = 0000101b ;default data for port 2 extension
```

```
;..... PORT 3 .....
bunholdi .equ 00000010b ;p31
eoffhooki .equ 00000010b ;p31
msgi .equ 00000100b ;p32
;p33 used as reference input
```

```
;port macro definitions:
```

```
pset .macro pm1,pm2 ;set port bits
or pm1, # pm2
.endm
```

```
pcir .macro pm1,pm2 ;clear port bits
and pm1, #~c pm2
.endm
```

```
p2set .macro pm1 ;set port2 bits
or p2i, # pm1
ld r2, p2i
.endm
```

```
p2clr .macro pm1 ;clear port2 bits
and p2i, #~c pm1
ld r2, p2i
.endm
```

```
p2out .macro pm1 ;setup direction of port2 to outputs
and p2mi, #~c pm1
ld p2m, p2mi
.endm
```

```
p2in .macro pm1 ;setup direction of port2 to inputs
or p2mi, # pm1
ld p2m, p2mi
.endm
```

```
; Power fail indicator bytes
CPWR1 .EQU 05Ah ;BATTERY BACK UP DETECTORS they get
CPWR2 .EQU 0A5h ;written out and checked for upon power on
```

```
DEFKEY1 = 00Ah ; DEFAULT KEY ONE
DEFKEY2 = 05Fh ; DEFAULT KEY TWO
```

```
; IRQ Identifiers
```

```
TOIRQ .EQU 010h ; irq4
TIIRQ .EQU 020h ; irq5
MSGIRQ .EQU 009h
```

```
; Message communication constants
```

```
STR .EQU 00000101b ;Preamble for transmitting key and status info
DATALEN .EQU 5 ;Number of bytes sent: 1-START 2-KEY 1-STATUS 1-pulse length
NUMBITS .EQU DATALEN * 8 ;Number of bits sent in key sequence
NUMGOOD .EQU (DATALEN-1) * 8 ;MIN number of bits IN 'GOOD' DATA
```

```
; Parametrics of the message data stream
```

```
; based upon the half bit time of 105 spaced multiples of 4th root of 2 apart
```

```
BIT_T_MIN .EQU 84 ;88 computed
BIT_T_H .EQU 105 ;half bit time
BIT_T_MAX .EQU 125
BIT_T_OH .EQU 148 ; 1.5 pulse time for stop indication
BIT_T_2MIN .EQU 177
;the full bit time is: 210
BIT_T_2MAX .EQU 252 ;250 computed
```

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

名

五、發明說明 (19)

```

; message type identifiers
ID_ACK .EQU 01h ;requests and acks audio path
ID_UNACK .EQU 02h ;indicates the pending release of the line
ID_RING .EQU 04h
ID_FLASH .EQU 08h ;pulse the co line for .5 sec
ID_KEYREQ .EQU 10h
ID_HANG .EQU 20h
ID_HOLD .EQU 40h ; put the line on hold
ID_PULSE .EQU 80h ; identifies pulse count
ID_UNRING .EQU 80h ; identifies to exts to stop ringing

RINGMAX = 90 ; t0 cycles in 15 hz signal
RINGMIN = 18 ; t0 cycles in 70 hz signal
RINGCY = 3 ; cycles counted on the line to detect!

```

```

;Data register locations
; r0: port 0
; r1: port 1
; r2: port 2
; r3: port 3
; r4: temp variable used all over but in interrupts
; r5:
temp0 = r4
temp1 = r5
; r6: Used by the receive interrupt routines
; r8-9 used by send_msg routine as params
; r8,9 used by ext pulse routine
trf10 = r7 ;time ring signal fell last
trf11 = r8
trr10 = r9 ;time ring signal rose last
trr11 = r10
rcc = r11 ;ring cycle counter
temp2 = r12
temp3 = r13 ; ext
pgmt = r13 ; base program indicator timer
time1 = r14 ; t0 cycle counter ho
time0 = r15 ; t0 cycle counter lo
; start the data variables above the working registers
msgio .equ 10h ; shift register for receive and transmit
; reserve 5 bytes for above
pwr1 .equ 15h ; power indicator byte 1
key .equ 16h ; the key is saved here
; reserve 2 bytes for key
p2mi .equ 18h ; "image" register for p2m
p2i .equ 19h ; "image" register for p2
rngwas = 1ah ; ring level was
good .equ 1bh ; good data flag/counter
; lch is available
; ldh is available
pwr2 .equ 1eh ; power indicator byte 2
ind .equ 1fh ; eight bits for flags datawas,middle,etc.

```

```

;IND POINTERS
PGM .EQU 001h
MIDDLE .EQU 004h ;MIDDLE OF BIT INDICATOR for receive routine
MAIL .EQU 020h

```

```

;*****
;** Beginning of ROM
;** Interrupt vectors come first, then code begins as 0Ch
;** Set up the stack and register pointers.
;** Test the base/extension input and jump to the appropriate place.
;*****

```

```

vectors
.org 00h
.word ivec0
.word ivec1
.word ivec2
.word ivec3
.word ivec4
.word ivec5

.org 0Ch ; execution starts at 0Ch
;This is where the reset begins execution.

; initialize the control registers
di
ld p0lm, #06dh ; default value upon reset (Internal Stack - P0 inputs)
ld sph, #0 ; set the stack pointer
ld spl, #080h ; stack at the top of free memory
srp #0 ; use the lower memory as registers
; ports are now working registers, i.e. R0 = P0, R1 = P1, R2 = P2, R3 = P3
; this will not be changed throughout the program
; interrupt priority
ld ipr, #01eh ; interrupt priority register
; irq # - highest priority first
; 2 - stop1
; 0 - msg1

```

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

各

五、發明說明 (20)

```

; 5 - t1 (timer 1)
; 3 - no connect (P30 - Sin)
; 4 - t0 (timer 0) (Sout)
; 1 - no connect (P33)

; interrupt mask
ld    imr, #089h    ; Enable 0 and 3 for msgi
ld    irq, #00      ; clear any pending

; set up the timers
; latest operating frequency = 5.48 Mhz
; timer freq = Fosc / 2 / 4 / pre / tn
; 1.338 kHz = 5.48M / 2 / 4 / 2 / 256
; period = 747.45 us
PRESC = 2
TOPRE = PRESC * 2 - 1    ; continuous mode
TIPRE = PRESC * 2 - 2    ; internal, single pass mode

ld    pre0, #TOPRE
ld    prel, #TIPRE
ld    t0, #0             ; start modulo 256 t0 interrupts
ld    t1, #0             ; used for period measurement
; timer mode register
ld    tmr, #0fh          ; start as down counters load and enable

; test for base or extension
ld    p2m, #0ffh         ; make sure all p2 is inputs
tm    r2, #bsei          ; are we a base?
jp    z, extension       ; jump if not
;===== BASE ROUTINE =====
; Initialize the I/O ports
ld    r0, #bp0dad        ; make port 0 an output + internal stack
ld    p0lm, #06ch
; 'Image' registers are used because read modify write instructions don't
; give correct results for P2M and P2 (and others) ie: or p2m, #80h
ld    p2mi, #bp2ddd      ; set up the port 2 mode image register
ld    p2m, p2mi           ; set up port 2 mode to match
ld    p2l, #bp2dad       ; set up the port two latch image register
ld    r2, p2l            ; set up the latch for port 2
ld    p3m, #03h          ; port three input set to analog
; port two pull-ups active
clr    ind               ; clear all of the indicators
ei                      ; allow the interrupts to appear in irq see pg 130 of z8 tech man
; (testmode calls debounce routine that uses timer0)
; check to see if should go to test mode (user pressing button on pwr on)
tm    r2, #pshbtni
jp    nz, btestmode

; Check the power indicators to see if still there
cp    pwr1, #CPWR1       ; is first power indicator there?
jr    ne, pwrfail        ; no - jump to start up routine
cp    pwr2, #CPWR2       ; second?
jr    eq, main1          ; yes, skip program routine

pwrfail
ld    key, #DEFKEY1      ; set up default key
ld    key+1, #DEFKEY2
ld    pwr1, #CPWR1       ; save the power fail indicator
ld    pwr2, #CPWR2
jr    main1

setkey ; jump in here if pgm button pressed later on
pset  r0, ledon          ; turn off the led
ld    key, t0            ; Get the first number from the timer
call  debou

pb2l
tm    r2, #pshbtni       ; wait for the switch to release
jr    nz, pb2l
ld    key+1, t0          ; Get the second number from the timer
call  debou

pgmr
or    ind, #PGM          ; in pgm mode
pcir  r0, ledon          ; turn on the led
; set up FOR 06 xx xxh cycles or five minutes in three bytes
ld    pgmt, #006h
clr   time0
clr   time1

main1
tm    r2, #bsei          ; restart if stumbled onto the wrong code
jp    z, start
ld    rngwas, r2          ; save the condition of the ring line
and   rngwas, #bring1
p2set bpulseo
ei

main00
ld    rcc, #RINGCY
mainw ; loop here waiting for something to do
tm    irq, #TOIRQ        ; t0 counted out yet
jr    z, mainw02         ; no, jump
and   irq, #c TOIRQ      ; clear the interrupt
djnz  time0, mainw02     ; update the time

```

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

裝

五、發明說明 (21)

```

dlnz    timel, mainw02
tm      ind, #PGM
jr      z, mainw02
dlnz    pgmt, mainw02
pset    r0, ledon      ; turn off the led
and     ind, #~c PGM

mainw02
tm      r2, #stopi     ; check for reset
jp      nz, stop
tm      r2, #pshbtnti  ; check for the button
jr      nz, setkey
tm      ind, #MAIL     ; any messages?
jp      nz, bparse

ld      temp0, r2      ; ringing?
and     temp0, #bringi ; mask the pin
ld      temp1, temp0
xor     temp0, rngwas  ; changed?
jr      z, mainw      ; n- loop more
ld      rngwas, temp1  ; save the new value for later
tm      temp1, #bringi ; rizing?
jr      z, main10     ; n- go
ld      temp0, trr10
ld      temp1, trr11
ld      trr10, time0
ld      trr11, timel
jr      main20

main10
ld      temp0, trf10
ld      temp1, trf11
ld      trf10, time0
ld      trf11, timel

main20
sub     temp0, time0
sbc     temp1, timel
cp      temp0, #RINGMAX
jr      ugt, main00
cp      temp0, #RINGMIN
jr      ult, main00
cp      temp1, #0
jr      ne, main00
dec     rcc
jr      nz, mainw

ringthing
ld      r8, #ID_RING
ld      temp0, #03
call    send_msg      ; send it three times
;wait for ring to quit
rngwq
ld      rngwas, r2      ; save the condition of the ring line
and     rngwas, #bringi
ld      temp2, #RINGMAX ; count cycles for maximum wait

rngwql
tm      ind, #MAIL     ; did someone answer?
jr      nz, rngmail    ; y- go do it
ld      temp0, r2      ; ringing?
and     temp0, #bringi ; mask the pin
xor     temp0, rngwas  ; changed?
jp      nz, rngwq      ; y- start wait again
tm      irq, #TOIRQ
jr      z, rngwql
and     irq, #~c TOIRQ
dlnz    temp2, rngwql
ld      r8, #ID_UNRING
ld      temp0, #03
call    send_msg      ; send it three times
jr      main1

rngmail
cp      msgio+3, #ID_ACK ; ack sent?
jr      eq, bparse
ld      r8, #ID_UNRING
ld      temp0, #03
call    send_msg      ; send it three times
jr      bparse ;(just fall through for now)
;----- Base Message Parser -----
bparse ; see if the key matches ours (only expecting key requests or line requests)
cp      msgio+1, key+0 ; see if their key matches ours
jr      ne, isnw       ; jump if no match
cp      msgio+2, key+1 ; second byte
jr      ne, isnw       ; jump if match
cp      msgio+3, #ID_ACK ; Ack request?
jr      eq, bpack

isnw
cp      msgio+3, #ID_KEYREQ ; extension asking for key?
jr      ne, bpexitm      ; collect the garbage and throw it away
bpggm  ;check for an key request message
call    get_mail        ; wait for ext to send all 3
tm      ind, #PGM       ; is this in program mode
jr      z, bpexit       ; no, go back
ld      r8, #ID_KEYREQ

```

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明 (22)

```

ld    temp0, #03      ; send it three times
call  send_msg
jp    bpexit

bpack
call  get_mail
ld    r8, #ID_ACK      ; return the ack
ld    temp0, #03      ; send it three times
call  send_msg
p2set boffhooko        ; go off hook
pcir  r0, ledon        ; turn on led
pcir  r0, muteo        ; enable the audio
ei    ; allow for more messages

bpackl ;here we hold the line
tm    r2, #stopi      ; check for reset
jp    nz, stop
tm    ind, #MAIL
jr    z, bpackl
cp    msgio+3, #ID_PULSE ; pulse requested?
jr    ne, ishang

bppulse ; pulse the switch hook time of pulse in message
pset  r0, muteo        ; disable the audio
p2clr bpulseo          ; turn off audio response control
ld    temp0, msgio+4    ; get the value
or    tmr, #01         ; restart at a new load
pset  r0, ledon        ; turn off led
p2clr boffhooko        ; on hook
call  dlyl            ; off hook
p2set boffhooko        ; off hook
pcir  r0, ledon        ; turn on led
pcir  r0, muteo        ; enable the audio
p2set bpulseo          ; turn on audio response control
and   ind, #'c MAIL    ; say we got the message
ei

ishang
cp    msgio+1, key+0    ; key matches so tones won't kill the line
jr    ne, bnogood      ; jump if no match
cp    msgio+2, key+1    ; second byte
jr    ne, bnogood      ; jump if no match
cp    msgio+3, #ID_HOLD
jr    eq, bphold
cp    msgio+3, #ID_FLASH
jr    eq, bpflash
cp    msgio+3, #ID_HANG
jr    eq, bphang
; throw away unrecognized message
bnogood
and   ind, #'c MAIL    ; say we got the message
ei
jp    bpackl

bphang
pset  r0, muteo        ; disable the audio
call  get_mail
ld    r8, #ID_UNACK
ld    temp0, #03      ; send it three times
call  send_msg        ; enables the receiver when done
call  dlychk          ; wait for a message to appear
tm    ind, #MAIL      ; did one arrive?
jr    z, bphangup      ; n- hang up the CO line
cp    msgio+3, #ID_ACK ; got another ack?
jr    eq, bpack        ; y- go back

bphangup
p2clr boffhooko        ; hang up
pset  r0, ledon        ; turn off led

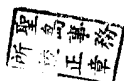
bpexitm
call  get_mail

bpexit
ei    ; allow for more messages
jp    mainl

bphold
pset  r0, muteo        ; disable the audio
call  get_mail
ld    r8, #ID_HOLD    ; tell the exts
ld    temp0, #03      ; send it three times
call  send_msg        ; enables the receiver when done
clr   temp0
clr   temp1
ld    temp2, #10      ; wait many minutes

bphold10
tm    ind, #MAIL      ; wait for more mail for many minutes
jr    nz, bphold20
tm    r3, #bunhold1    ; unhold?
jr    nz, bphold14    ; y- see if for real
tm    r2, #stopi      ; check for reset
jp    nz, stop
tm    irq, #TOIRQ      ; t0 counted out yet?
jr    z, bphold10      ; no, loop more
and   irq, #'c TOIRQ
djnz  temp0, bphold10

```



(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明 (23)

```

djnz    temp1, bphold10
ld      r8, #ID_HOLD
ld      temp0, #3
call    send_msg
djnz    temp2, bphold10
ld      temp1, #12      ; wait 1 sec
clr     temp0
call    dlychk2
tm      ind, #MAIL      ; if mail then see what it is
jr      nz, bphold20
ld      r8, #ID_HOLD
ld      temp0, #3
call    send_msg
ld      temp1, #157     ; wait 30 seconds
clr     temp0
call    dlychk2
tm      ind, #MAIL      ; if mail then see what it is
jr      nz, bphold20
p2clr   boffhooko      ; hang up
pset    r0, ledon      ; turn off led
jp      main1

bphold14
ld      temp0, #7      ; deglitch for 5 ms
bphold16
; unhold pin went high
tm      r3, #bunhold1  ; check for gone already?
jr      z, bphold10    ; if gone then it was a glitch!-- go back
tm      r2, #stop1     ; check for reset
jp      nz, stop
tm      irq, #TOIRQ    ; t0 counted out yet?
jr      z, bphold16    ; no, loop more
and     irq, #c TOIRQ
djnz    temp0, bphold16
p2clr   boffhooko      ; hang up
pset    r0, ledon      ; turn off led
jp      main1

bphold20
cp      msgio+3, #ID_ACK ; got another ack?
jp      eq, bpack      ; y- go back
p2clr   boffhooko      ; hang up
pset    r0, ledon      ; turn off led
jp      main1          ; go pick up the mail elsewhere

bflash
; pulse for .75 sec = 1003 tics -03 ebh
p2clr   bpulseo        ; turn off audio response control
p2clr   boffhooko      ; go on-hook
pset    r0, ledon      ; turn off led
ld      temp0, #0ebh
ld      temp1, #04     ; inc this one
call    dly2
p2set   boffhooko      ; go off-hook
pclr    r0, ledon      ; turn on led
p2set   bpulseo        ; turn on audio response control
call    get_mail
ei
jp      bpack1

```

```

;*****
;***** EXTENSION ROUTINES *****
;*****

```

```

extension    ; initialize the I/O ports
ld          r0, #ep0dad
ld          p01m, #06ch      ; make port 0 an output

; 'image' registers are used because read modify write instructions don't
; give correct results for P2M and P2 (and others) ie: or p2m, #80h
ld          p2m1, #ep2ddd    ; load ext dat dir def
ld          p2m, p2m1
ld          p21, #ep2dad
ld          r2, p21
ld          p3m, #03h        ; port three always input set to analog
clr         ind              ; clear all of the indicators
ei          ; allow the interrupts to appear in irq

; (testmode calls debounce routine that uses the interrupts)
; first check to see if in test mode
tm          r2, #pshbnt1
jp          nz, etestmode

; Check the power indicators to see if still there
cp          pwr1, #CPWR1     ; see if first power indicator is ram
jr          ne, epwrfail     ; no - jump to start up routine
cp          pwr2, #CPWR2     ; second flag present
jr          eq, emain        ; yes, go to main

epwrfail
ld          key, #DEFKEY1    ; set up the default keys
ld          key+1, #DEFKEY2
ld          pwr1, #CPWR1     ; show have keys
ld          pwr2, #CPWR2
jr          emain

eblink

```

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

名

五、發明說明 (24)



```

ld    pwr1, #0      ; destroy pwr ind if jumped into
tm    r2, #pshbtni  ; port 2 mask the push button
jr    nz, epb1      ; button pushed branch pbl
djnz  temp0, eblink ; not time to blink yet
djnz  temp1, eblink ; not time to blink yet
xor    r0, #ledon   ; change the light
ld    r2, p2i
ld    temp1, #12h   ; set blink rate
jr    eblink

epb1   pset    r0, ledon      ; turn off the led
      call    debou

epb2l  tm    r2, #pshbtni    ; wait for the switch to release
      jr    nz, epb2l      ; button pushed branch pbl
      call    debou

egutkey ; Request the key from the base
p2clr  exmiton      ; turn on the transmitter
call   delay        ; wait for .5 sec don't check receive
call   delay        ; wait for .5 sec don't check receive
ld     r8, #ID_KEYREQ ; send the key request
ld     temp0, #3     ; ask 3 times
and    ind, #c MAIL  ; remove any junk messages
call   send_msg
p2set  exmiton      ; turn off the transmitter

egkl   call   dlychk      ; wait for return of message
tm     ind, #MAIL      ; see if msg was sent
jr     z, eblink      ; n- go back
call   get_mail        ; sent multiple times -- throw rest away
cp     msgio+3, #ID_KEYREQ ; is this a key request return
jr     ne, eblink      ; n- go back
ld     key+0, msgio+1   ; save the keys
ld     key+1, msgio+2   ; save the keys
ld     pwr1, #CPWR1    ; show have keys
ld     pwr2, #CPWR2    ; show have keys
pclr   r0, ledon      ; turn on the light (indicate we got it!)
call   delay
pset   r0, ledon      ; turn off the light

emain  pset    r0, muteo    ; turn off the audio
      p2in    msgo         ; make message input (hi-z)
      p2set   ehvon        ; turn off ring voltage
      pset    r0, ledon    ; turn off light
      p2set   exmiton      ; turn off xmit
      tm     r2, #bsei     ; restart if wrong unit
      jp     nz, start

emainl tm     r2, #stopi    ; check for stop
      jp     nz, stop
      tm     r2, #pshbtni  ; check for button
      jr     nz, egetkey   ; check for messages
      tm     ind, #MAIL    ; check for off hook
      jp     nz, eparse
      tm     r3, #coffhooki ; check for off hook
      jr     z, emainl     ; this time jump if not present

epoh   ; Extension PHONE IS OFF HOOK enters from ring routine also
pset   r0, muteo        ; disable audio (if come from ring)
p2clr  exmiton          ; turn on the transmitter
call   delay            ; wait for .5 sec don't check receive

epuhfp ld     r8, #ID ACK    ; REQUEST AN ACK
      ld     temp0, #3      ; ask 3 times
      call   send_msg       ; enables mail
      call   dlychk        ; check for mail
      tm     ind, #MAIL
      jr     z, epoh0
      cp     msgio+3, #ID_ACK ; ack returned?
      jr     eq, eonward
      call   get_mail       ; throw it away

epoh0  ld     r8, #ID ACK    ; REQUEST AN ACK again
      ld     temp0, #3      ; ask 3 times
      call   send_msg
      call   dlychk
      tm     ind, #MAIL
      jr     z, epoh11
      cp     msgio+3, #ID_ACK ; ack returned?
      jr     eq, eonward
      call   get_mail       ; throw it away

epoh11 ld     r8, #ID HANG   ; in case base saw us
      ld     temp0, #3      ; ask 3 times
      call   send_msg       ; (base returns an unack)
      p2set  exmiton        ; turn off the transmitter
      ld     temp1, #5      ; chirp four times and wait for on-hook

epoh05 ld     temp0, #40
      pclr   r0, muteo      ; turn on the audio

```

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明 (25)

```

epoh10 : (beep)
tm      r3, #eoffhooki ; are we still off hook?
jp      z, epoh35       ; n- go try again
tm      irq, #TOIRQ     ; t0 counted out yet?
jr      z, epoh10       ; no, loop more
and     irq, #c TOIRQ   ; change the light (make audio tone)
xor     r0, #ledon
ld      r2, p2i
djnz   temp0, epoh10
ld      temp0, #60

epoh20 : (pause beep)
tm      r3, #eoffhooki ; are we still off hook?
jr      z, epoh35       ; n- go try again
tm      irq, #TOIRQ     ; t0 counted out yet?
jr      z, epoh20       ; no, loop more
and     irq, #c TOIRQ
djnz   temp0, epoh20
djnz   temp1, epoh05    ; outter loop
pset   r0, muteo       ; turn off the audio

epoh30
tm      r2, #stopi      ; check for stop
jp      nz, stop
tm      r3, #eoffhooki ; are we still off hook?
jr      nz, epoh30      ; n- stick around

epoh35
call   get_mail        ; also debounces
ei
jp     emain

eo::ward
call   get_mail        ; throw away extra messages

es::p0
pcir   r0, muteo       ; turn on the audio
pcir   r0, ledon       ; turn on the light

eswpulse
tm      r2, #stopi      ; check for stop
jp      nz, stop
tm      r3, #eoffhooki ; is the phone still off hook
jr      nz, eswpulse    ; yes, wait

; phone just went on-hook
clr     r9              ; set up counter
eswpul2 : remove bounce and deglitch
tm      r2, #stopi      ; check for stop
jp      nz, stop
tm      irq, #TOIRQ     ; counted out yet
jr      z, eswpul2      ; no, wait for counter
and     irq, #c TOIRQ   ; clear the irq
inc     r9
cp      r9, #54         ; at least 40 ms?
jr      ult, eswpul2    ; no, wait some more
tm      r3, #eoffhooki ; still on hook?
jr      nz, eswpulse    ; n- go back and wait (just a glitch)
pset   r0, muteo       ; turn off audio
pset   r0, ledon       ; turn off the light

; measure time up to 90 ms for dial pulses (60 ms. USA, 67 ms others)
eswpulsel
tm      r3, #eoffhooki ; is the phone still on hook
jr      nz, dialdone    ; no, tell the base how long off
tm      r2, #stopi      ; check for stop
jp      nz, stop
tm      irq, #TOIRQ     ; counted out yet
jr      z, eswpulsel    ; no, wait for counter
and     irq, #c TOIRQ   ; clear the irq
inc     r9
cp      r9, #120        ; less than 90 ms?
jr      ult, eswpulsel  ; no, wait some more

; check for 'flash', 'hold' or 'hangup'
ld      temp0, #4       ; if gone longer than 3/4 sec hangup
eswp20 : (still on hook for first time)
tm      r3, #eoffhooki
jr      nz, eswp50      ; jump out of here on first full click v"
tm      r2, #stopi      ; check for stop
jp      nz, stop
tm      irq, #TOIRQ
jr      z, eswp20
and     irq, #c TOIRQ
inc     r9
jr      nz, eswp20
djnz   temp0, eswp20

; send msg to base that we are hanged up
ld      r8, #ID_HANG
ld      temp0, #3       ; snd msg 3 times
call   send_msg         ; (base returns an unack)
p2set  exmiton          ; turn off xmtr
call   dlychkm          ; pick up the unack
call   get_mail
ei
jp     emain            ; turn on the receiver
dialdone
ld      r8, #ID_PULSE   ; takes long enough to debounce the thing

```

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明 (26)

```

ld    temp0, #1          ; ask 1 time
call  send_msg           ; send the message
jp    eswp0
; could be flash or hold (now off hook)
eswp50
clr   r9
ld    temp0, #4          ; if not back in 3/4 sec then flash
; remove bounce
eswp52
tm    r2, #stopi        ; check for stop
jp    nz, stop
tm    irq, #TOIRQ       ; counted out yet
jr    z, eswp52         ; no, wait for counter
and   irq, #~c TOIRQ    ; clear the irq
inc   r9
cp    r9, #120          ; 90 ms?
jr    ult, eswp52       ; no, wait some more
; (still off hook)
eswp60
tm    r3, #eoffhooki    ; check for stop
jr    z, eswp70
tm    r2, #stopi
jp    nz, stop
tm    irq, #TOIRQ
jr    z, eswp60
and   irq, #~c TOIRQ
inc   r9
jr    nz, eswp60
djnz  temp0, eswp60
ld    r8, #ID_FLASH     ; if falls through assume flash
ld    temp0, #3         ; ask 3 times
call  send_msg           ; send the message
jp    eswp0
; hold wanted (on hook again)
eswp70
ld    r8, #ID_HOLD
ld    temp0, #3
call  send_msg           ; long enough to debounce it
p2set exm1on            ; turn off xmtr
call  dlychk; pick up the unack
call  get_mail
ei    ; turn on the receiver
clr   temp0
ld    temp1, #5         ; set up for 3/4 sec
; wait for off-hook within 3/4 sec
eswp80
tm    r3, #eoffhooki
jr    nz, eswp85
tm    r2, #stopi
jp    nz, stop
tm    irq, #TOIRQ
jr    z, eswp80
and   irq, #~c TOIRQ
djnz  temp0, eswp80
djnz  temp1, eswp80     ; fall through if hung up for good
; turn on the audio
eswp85
call  debou
pcir  r0, muteo
; wait for on-hook (guy hangs up his phone)
eswp90
tm    r3, #eoffhooki
jr    z, eswp100
tm    r2, #stopi
jp    nz, stop
tm    irq, #TOIRQ
jr    z, eswp90
and   irq, #~c TOIRQ
xor   r0, #ledon        ; change the light (make audio tone)
ld    r2, p21
jr    eswp90
; turn off the audio
; turn off the light
; now quit
eswp100
pset  r0, muteo
pset  r0, ledon
call  debou
jp    emain

;***** Extension Parse MESSAGES ROUTINE *****
eparse ; see if the key matches ours
cp    msgio+1, key+0    ; compare the 2 bytes of the stored to rec.
jp    ne, epexitm      ; jump if no match
cp    msgio+2, key+1    ; second byte
jp    ne, epexitm      ; jump if no match
cp    msgio+3, #ID_RING ; is it a ring
jr    eq, epring
cp    msgio+3, #ID_HOLD ; is it holding? (signal the user)
jr    eq, ephold
cp    msgio+3, #ID_ACK
jp    eq, epack
; could be key request, unack, unring...
jp    epexitm          ; enables ints
; *** ring routines ***
; ring routine 50% DUTY 25 HZ
ptime .equ 25          ; 20ms pulse width
ringsub
pset  r0, muteo        ; turn off the audio (used for caller-id)

```



(請先閱讀背面之注意事項再填寫本頁)

裝

訂

名

五、發明說明 (27)

```

rsloop    p2clr    eppvon    ; turn off lsource
          ld      temp0, #2
          call    dly1
          p2clr    ehvon     ; turn on hv
          ld      temp0, #ptime
          call    dly1
          p2set    ehvon     ; hv = off
          ld      temp0, #2
          call    dly1
          p2set    eppvon    ; turn on lsource
          ld      temp0, #ptime
          call    dly1
          tm      r3, #eoffhook1 ; check if the phone is off hook
          jr      nz, rs10
          tm      ind, #MAIL
          jr      nz, rs10
          djnz    temp2, rsloop ; finished with ring? no - ring more
rs10      ret
; * ring the extension phone
rtime     .equ    52 ;ring time = 2.0 sec++
epring    call    get_mail
          ei      ;allow for unring commands
          ld      temp2, #rtime
          call    ringsub
          tm      r3, #eoffhook1 ; check if the phone is off hook
          jp      nz, epoh
          tm      ind, #MAIL ; did the mail bump us out?
          jp      nz, eparse
          jp      epexit ; finished
hldrtime  .equ    3 ;ring for short bursts
epioid    call    get_mail
          ld      temp3, #3 ; ring the phone 3 times
ep:oi     ld      temp2, #hldrtime
          call    ringsub
          tm      r3, #eoffhook1 ; check if the phone is off hook
          jp      nz, epoh
          ld      temp0, #0eh ; pause for stime * ptime * 2
          ld      temp1, #02 ; inc this one
          call    dlychk2
          tm      r3, #eoffhook1 ; check if the phone is off hook
          jp      nz, epoh
          djnz    temp3, ephol ; ring a few times
          jr      epexit ; when finished exit
epack     ; if here the phone is on hook and an ack received--wait for unack
          call    get_mail ; pick up the rest of the messages
          ei      ; allow for more to come
epack05   pclr     r0, muteo ; allow for audio tone
          ld      temp0, #050h
          ld      temp1, #002h ; inc this byte
epack10   tm      ind, #MAIL ; mail?
          jr      nz, epack20 ; y- go on
          tm      r2, #stopi ; check for stop
          jp      nz, stop
          tm      irq, #TOIRQ ; t0 counted out yet?
          jr      z, epack10 ; no, loop more
          and     irq, #c TOIRQ
          xor     r0, #ledon ; change the light (make audio tone)
          ld      r2, p2i
          djnz    temp0, epack10
          djnz    temp1, epack10
          pset    r0, muteo ; turn off the audio
          ld      temp0, #050h
          ld      temp1, #002h ; inc this byte
epack12   ;cycle some with out any tone
          tm      ind, #MAIL ; mail?
          jr      nz, epack20 ; y- go on
          tm      r2, #stopi ; check for stop
          jp      nz, stop
          tm      irq, #TOIRQ ; t0 counted out yet
          jr      z, epack12 ; no, jump
          and     irq, #c TOIRQ
          djnz    temp0, epack12
          djnz    temp1, epack12
          jr      epack05
epack20   pset     r0, muteo ; turn off the audio
          pset    r0, ledon ; turn off the light
          jp      eparse ; go back and see what was sent
epexitm   call    get_mail
epexit

```

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明 (28)

```

ei                ;allow for more mail
jp      emain

;----- Subroutines used by both BASE and EXTENSION -----
;*** STOP routine
stop
    ld      p2m, #0ffh    ;make all p2 inputs
    ld      p0im, #06dh   ;make all p0 inputs
    nop
    stop
;generic single variable wait
div1    ld      temp1, #1
        jr      dly2
; * Wait for a while 0.5s = 669 t0 cycles = 0290h
delay
    ld      temp0, #09dh
    ld      temp1, #003h   ;inc this byte
dly2    ;generic two variable wait enters here
        tm      r2, #stop1  ; check for stop
        jp      nz, stop
        tm      irq, #TOIRQ ; t0 counted out yet
        jr      z, dly2     ; no, jump
        and     irq, #~c TOIRQ
        djnz    temp0, dly2
        djnz    temp1, dly2
        ret

; * Wait for 0.5++ second if there is not any mail
-----
dlychk1 ld      temp0, #0adh ; 16 more for those waiting
        ld      temp1, #003h ;inc this byte
dlychk2 tm      ind, #MAIL    ; Mail?
        jr      nz, dlychkend ; yes, end
        tm      r2, #stop1  ; check for stop
        jp      nz, stop
        tm      irq, #TOIRQ ; t0 counted out yet
        jr      z, dlychk2  ; no, jump
        and     irq, #~c TOIRQ
        djnz    temp0, dlychk2
        djnz    temp1, dlychk2
dlychkend
        ret

;*-generic delay checking the hook status
dlychkh2 tm      r3, #eoffhook1 ; Off hook?
        jr      nz, dlyhend   ; yes, end
        tm      r2, #stop1  ; check for stop
        jp      nz, stop
        tm      irq, #TOIRQ ; t0 counted out yet
        jr      z, dlychkh2  ; no, jump
        and     irq, #~c TOIRQ
        djnz    temp0, dlychkh2
        djnz    temp1, dlychkh2
dlyhend
        ret

; * Wait for extra keys
; * Wait for 88 tofs = 058h = 66 ms
get_mail
    ld      temp0, #88
    call    dly1
    and     ind, #~c MAIL    ; say we got the mail
    ret

; * debounce routine
; * Wait for .1 second
dehou
    ld      temp0, #134
    call    dly1
    ret

; * BASE TEST MODE
; * turn on the xmitter, audio, and led
btestmode
    pcir    r0, ledon        ; turn on led
    pzset   boffhooko       ;C.O. line off-hook
    pcir    r0, muteo       ; enable the audio
    btm10   jr      btm10

; * EXTENSION TEST MODE
; * turn on the xmitter, audio, and led

```

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明 (29)

```

etestmode
    p2clr    exmiton    ; turn on xmtr
    pclr     r0, ledon   ; turn on led
    pclr     r0, muteo   ; enable the audio
etm10
    jr       etm10

*****
                SEND_MSG
*****
; Take the Start, Key, and Type bytes and send them out the keyo pin
; The accumulator holds the MESSAGE TYPE.
; Note: parameter temp0 (number of times sent) is set by the calling routine
; param r8, r9 are the message mostly r8
;
; Also since the receive routines would ruin the timing the receiver is
; disabled when sending messages
;
; Uses: temp0, temp1, msgio, and T0 in continuous mode
*****
send_msg
    di                    ; disable interrupts from messing timing or vars
    p2out    msgo         ; turn on the output
    and      tmr, #0ch     ; turn off timer0
    ld       t0, #BIT_T_H ; 1/2 bit time
    or       tmr, #03h     ; load & start timer0

sk11
    ld       msgio+0, #STR ; initialize the key sequence
    ld       msgio+1, key+0 ; put in the keys
    ld       msgio+2, key+1 ; put in the keys
    ld       msgio+3, r8    ; move the id byte into the message
    ld       msgio+4, r9
    ld       temp1, #NUMBITS ; transmit all of the bits

sk12
    rlc      msgio+4
    rlc      msgio+3
    rlc      msgio+2
    rlc      msgio+1
    rlc      msgio    ; carry bit used later too - do not destroy!
    jr       nc, skout0 ; if zero clear image bit
    or       p2i, #msgo ; else set the bit
    jr       skoutd

skout0
    and      p2i, #~c msgo ; clear the bit

skoutd
    tm       irq, #TOIRQ    ; time out yet?
    jr       z, skoutd      ; no, jump skoutd
    ld       r2, p2i        ; output the bit
    and      irq, #~c TOIRQ ; reset the interrupt
    xor      p2i, #msgo     ; change the bit for next half

skoutd1 ; wait for the counter time
    tm       irq, #TOIRQ    ; time out yet?
    jr       z, skoutd1     ; no, jump skoutd
    ld       r2, p2i        ; output the bit
    and      irq, #~c TOIRQ ; reset the interrupt
    djnz     temp1, sk12    ; done with message? no, sk12

    xor      p2i, #msgo     ; change the bit for next half
    ld       t0, #BIT_T_OH ; change to end bit 1.5 bit time

skoutd2 ; wait for the counter time
    tm       irq, #TOIRQ    ; time out yet?
    jr       z, skoutd2     ; no, jump skoutd
    ld       r2, p2i        ; output the bit
    and      irq, #~c TOIRQ ; reset the interrupt
    xor      p2i, #msgo     ; change the bit for next half

skoutd3 ; wait for the counter time
    tm       irq, #TOIRQ    ; time out yet?
    jr       z, skoutd3     ; no, jump skoutd
    ld       r2, p2i        ; output the bit
    and      irq, #~c TOIRQ ; reset the interrupt
    ld       t0, #BIT_T_H   ; restore back to 1/2 bit time
    djnz     temp0, sk11    ; send msg multiple times

    p2in     msgo          ; make input
    ld       t0, #00       ; reset t0 for full loop
    ei       ; allow interrupts again
    ret

***** Interrupt RECEIVE ROUTINES *****
; 'Owns' R6, msgio, T1, good Irq0 and Irq3
; disabled by transmitt routine and when message is received
; lasts about 60 uS.
***** Falling message line routine *****
ivec0
    ;p32 F edge msgi
    pset     r0, msgfo      ; output the new value
    ld       r6, t1         ; get the time
    or       tmr, #0ch     ; restart timer 1
; See if the transition fits within the allotted times:
    com      r6             ; count time = -(down count time)
    cp       r6, #BIT_T_2MIN ; smaller than a two?

```

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

系

五、發明說明 (30)

```

jr      ult, rcvcsf      ; y- check for stop or data
cp      r6, #BIT_T_2MAX ; smaller than 2 * max
jr      ugt, rcvbadf     ; n - to big for data

rcvlongf
inc      good
or       ind, #MIDDLE    ; set the middle indicator
scf      ;use for upright data
;       rcf              ;use for inverted data
rlc      msgio+4         ; move the bit into data stream
rlc      msgio+3
rlc      msgio+2         ; shift in the received bits
rlc      msgio+1
rlc      msgio+0
jr      rcvexitf

rcvcsf
cp      r6, #BIT_T_MAX   ; is it a stop?
jr      ugt, rcvstopf    ; y- goto stop routine
cp      r6, #BIT_T_MIN   ; it is a short?
jr      ult, rcvbadf     ; n- too small for data
tm      ind, #MIDDLE     ; if was at edge then goto the middle routine
z, rcvlongf              ; else was at middle go to edge
and     ind, #c MIDDLE   ; clear middle indicator
jr      rcvexitf

rcvbadf
clr      good            ; reset the data counter
jr      rcvexitf

rcvstopf
cp      good, #NUMGOOD   ; Enough to be a message?
jr      ult, rcvbadf     ; n- it's bad!
di      ; no more mail till this one is gone
or      ind, #MAIL       ; say we have something
jr      rcvexitf

and     irq, #c MSGIRQ
iret

```

----- Rising message line routine -----

```

ivcc3   ;p32 R edge msgi
pcir    r0, msgfo        ; output the new value
ld      r6, t1           ; get the time
or      tmr, #0ch        ; restart timer 1
; See if the transition fits within the allotted times:
com      r6              ; count time - -(down count time)
cp      r6, #BIT_T_2MIN ; smaller than a two?
jr      ult, rcvcsf      ; y- check for stop or data
cp      r6, #BIT_T_2MAX ; smaller than 2 * max
jr      ugt, rcvbadf     ; n - to big for data

rcvlongr
inc      good
or       ind, #MIDDLE    ; set the middle indicator
rcf      ;use for upright data
;       scf              ;use for inverted data
rlc      msgio+4         ; move the bit into data stream
rlc      msgio+3
rlc      msgio+2         ; shift in the received bits
rlc      msgio+1
rlc      msgio+0
jr      rcvexitr

rcvcsr
cp      r6, #BIT_T_MAX   ; is it a stop?
jr      ugt, rcvstopr    ; y- goto stop routine
cp      r6, #BIT_T_MIN   ; it is a short?
jr      ult, rcvbadr     ; n- too small for data
tm      ind, #MIDDLE     ; if was at edge then goto the middle routine
z, rcvlongr              ; else was at middle go to edge
and     ind, #c MIDDLE   ; clear middle indicator
jr      rcvexitr

rcvbadr
clr      good            ; reset the data counter
jr      rcvexitr

rcvstopr
cp      good, #NUMGOOD   ; Enough to be a message?
jr      ult, rcvbadr     ; n- it's bad!
di      ; no more mail till this one is gone
or      ind, #MAIL       ; say we have something
jr      rcvexitr

and     irq, #c MSGIRQ
iret

; not expecting any of these interrupts -- all just return
ivcc1   ;p33 F edge
ivcc2   ;p31 F edge
ivcc4   ;t0
ivcc5   ;t1

iret
.end

```

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

系

五、發明說明 (31)

附 錄 B

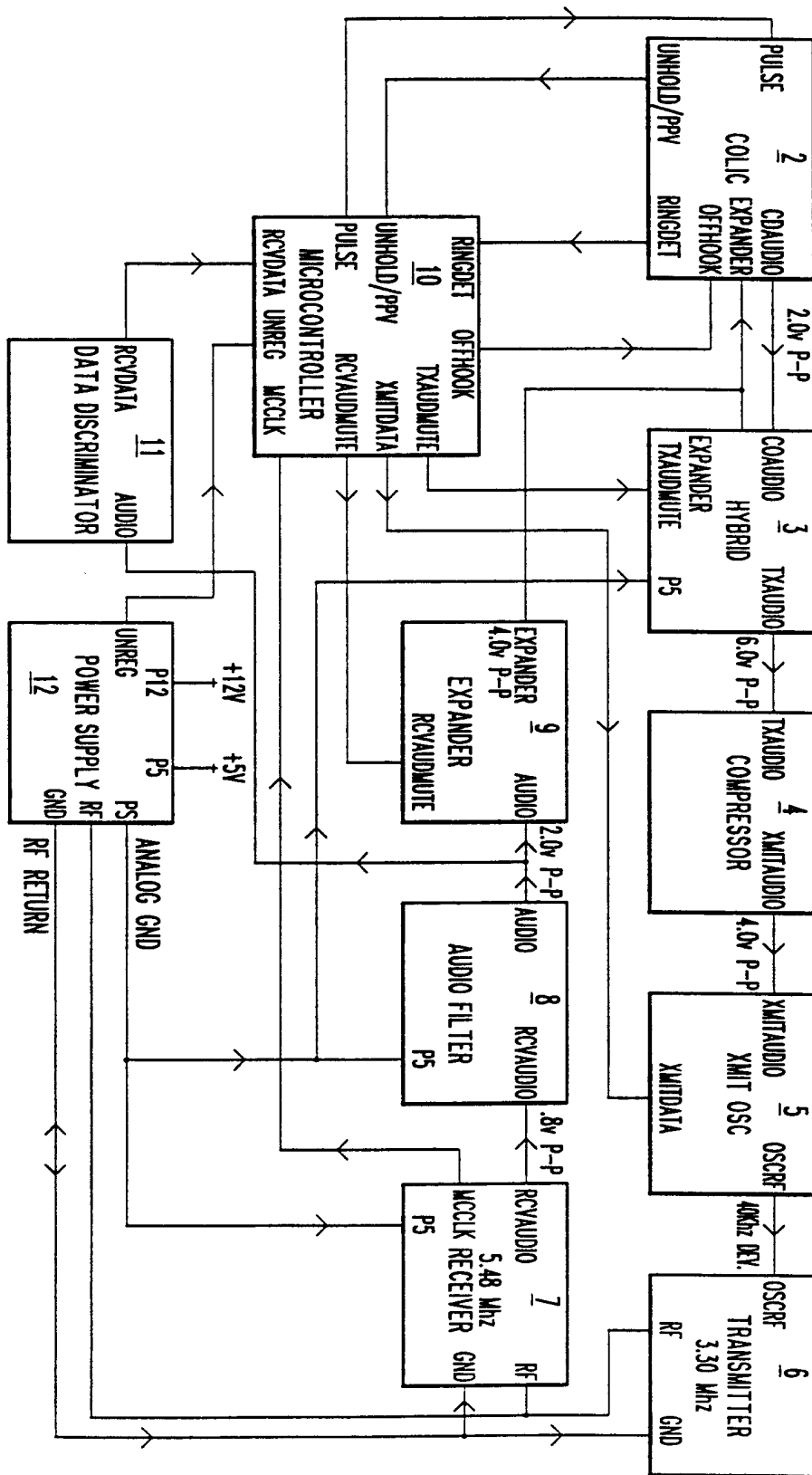
(請先閱讀背面之注意事項再填寫本頁)

裝

訂

系

五、發明說明 (32)



第 7 圖

(請先閱讀背面之注意事項再填寫本頁)

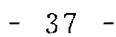
裝

訂

裝

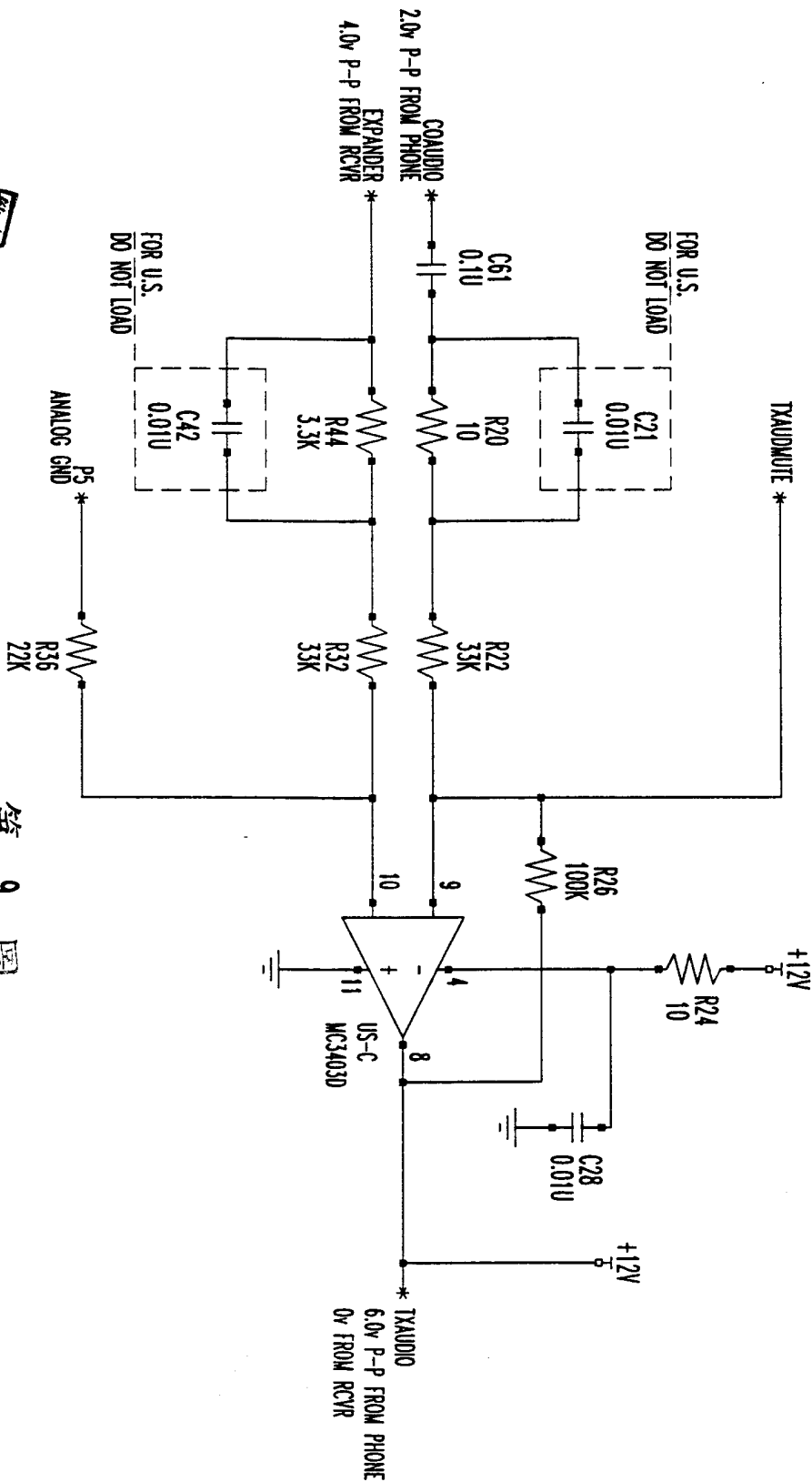
系

第 8 圖



五、發明說明 (34)

所聖
校島正
務事



第 9 圖

(請先閱讀背面之注意事項再填寫本頁)

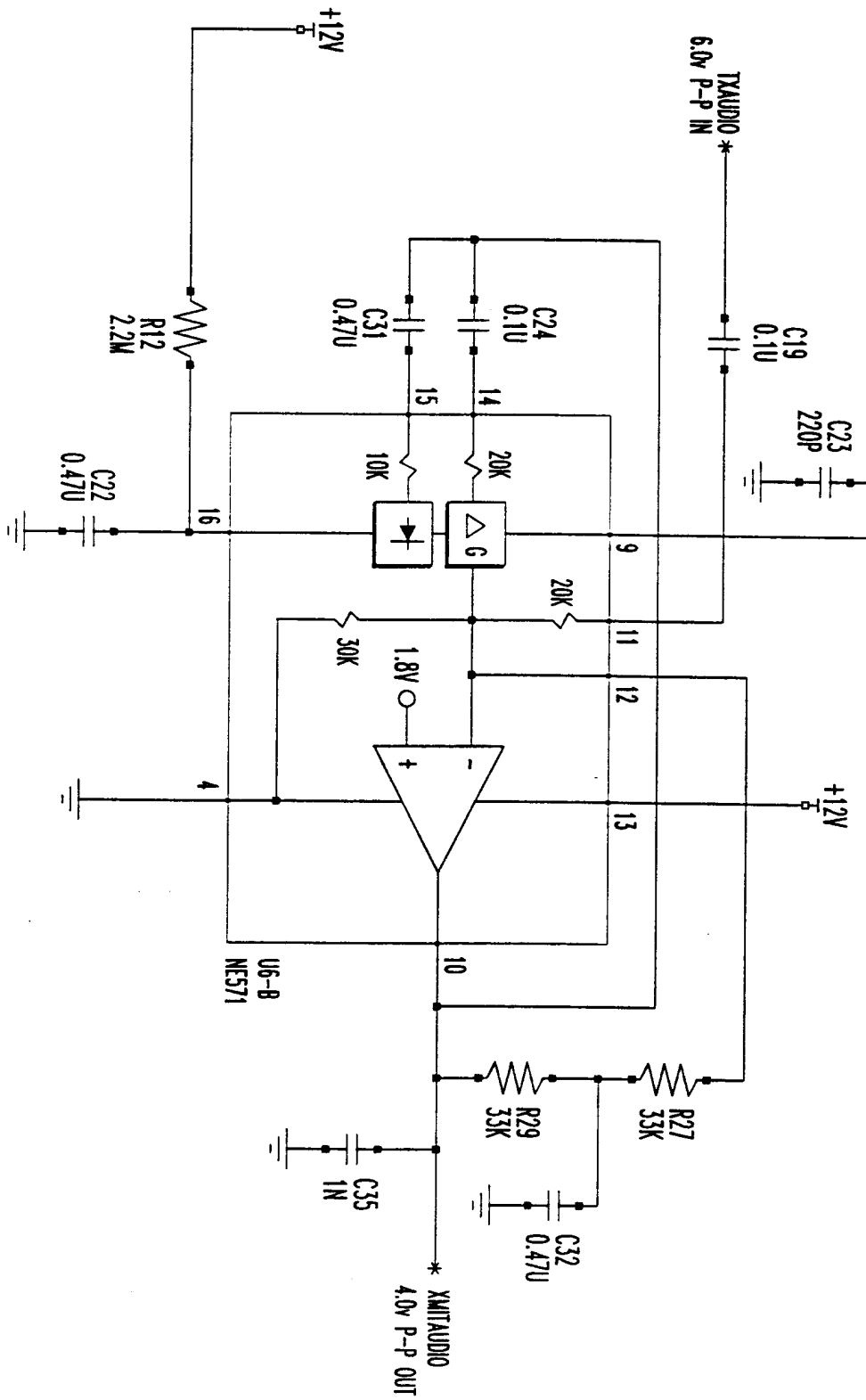
裝

訂

系

(請先閱讀背面之注意事項再填寫本頁)

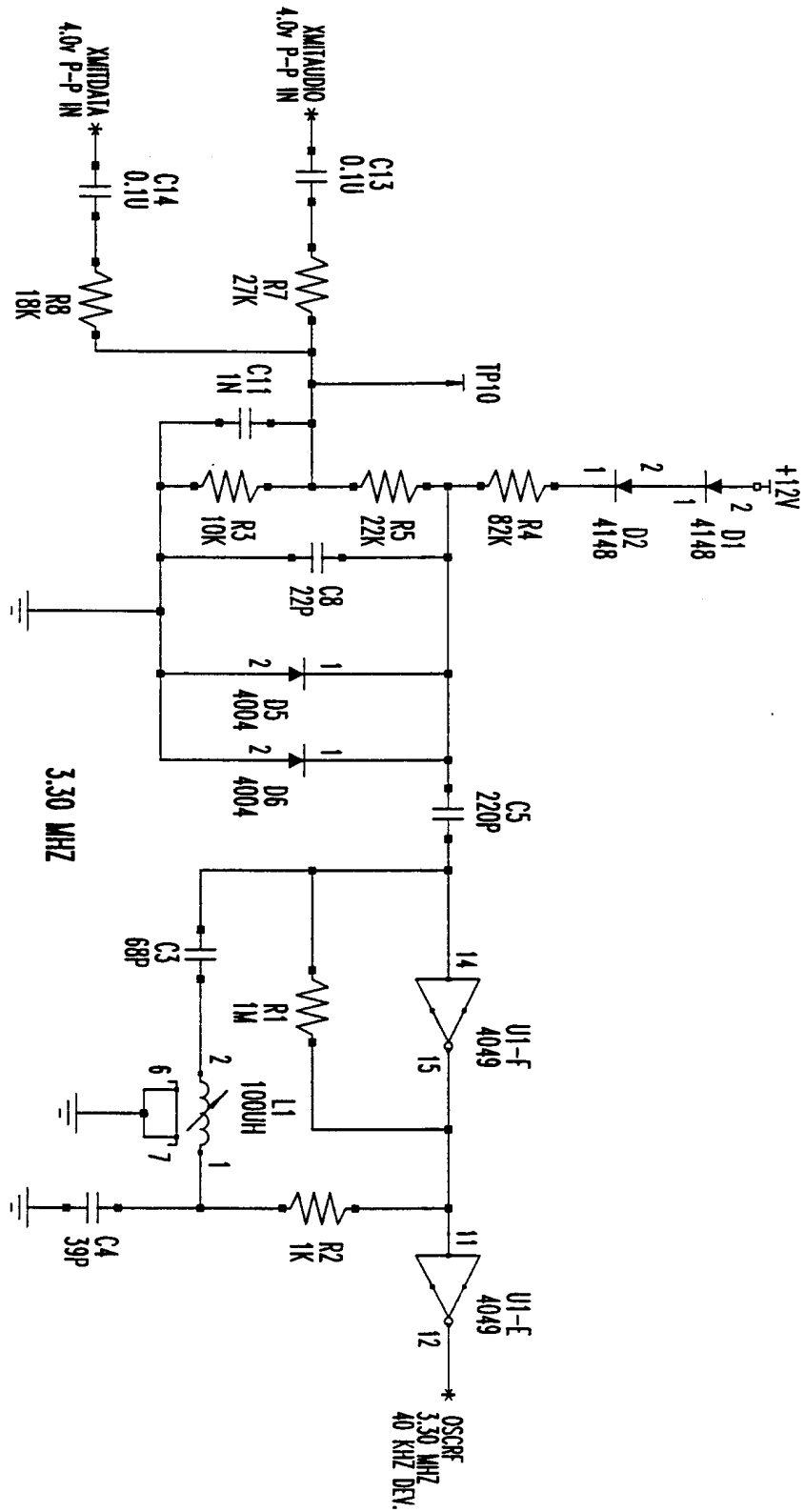
五、發明說明 (35)



第 10 圖

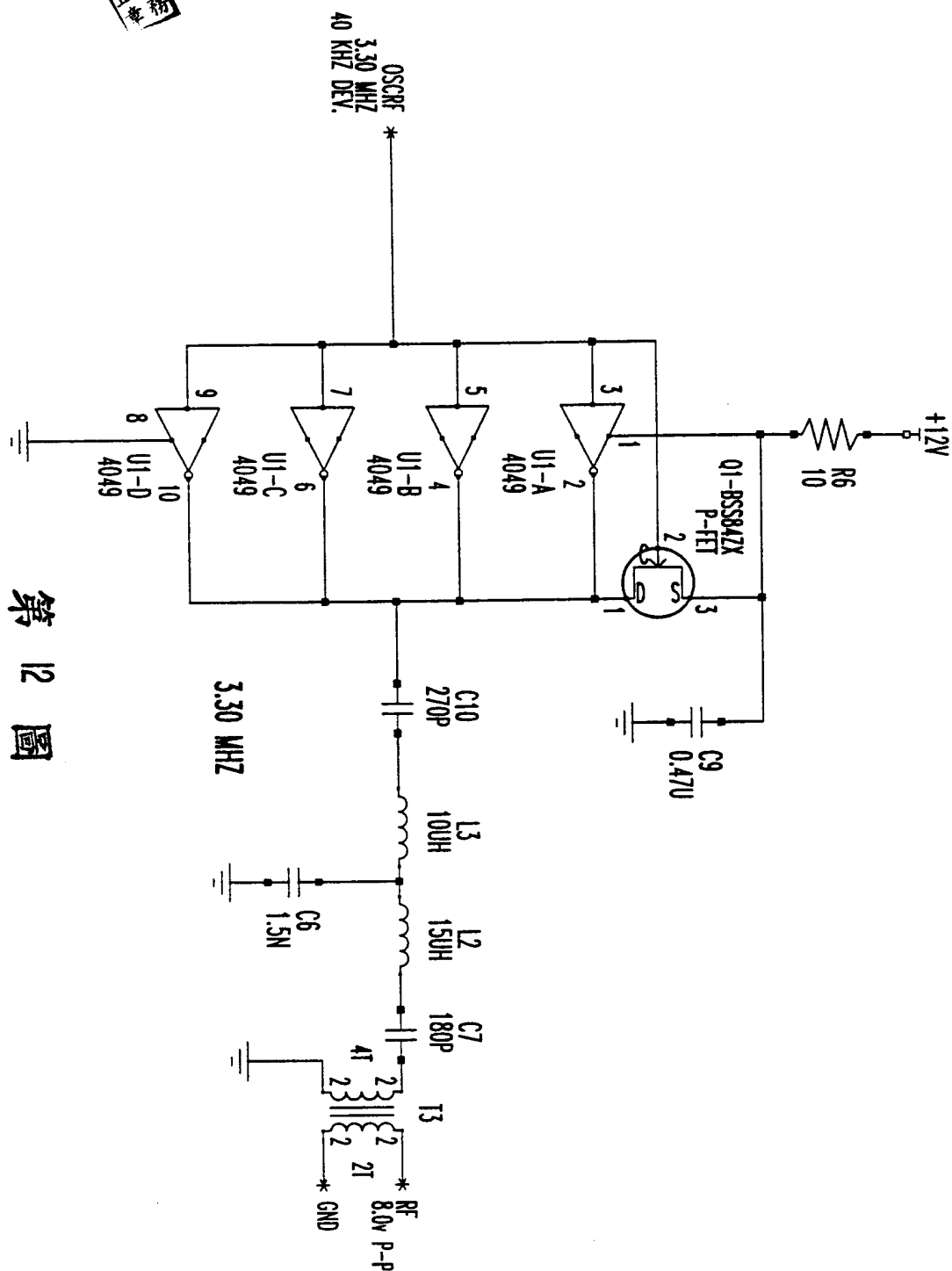
(請先閱讀背面之注意事項再填寫本頁)

五、發明說明 (36)



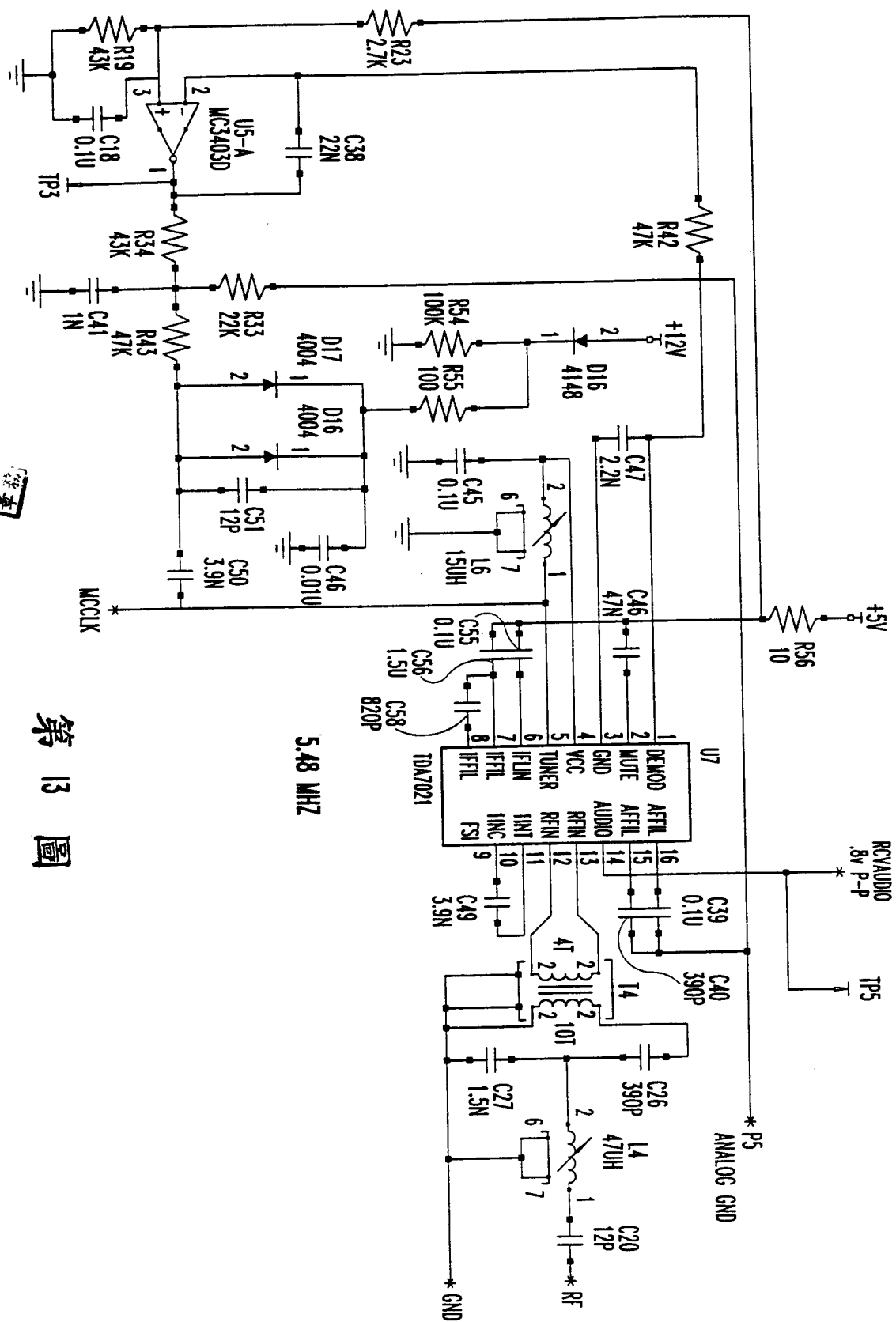
第 二 圖

五、發明說明 (37)



第 12 圖

五、發明說明 (38)



第 13 圖

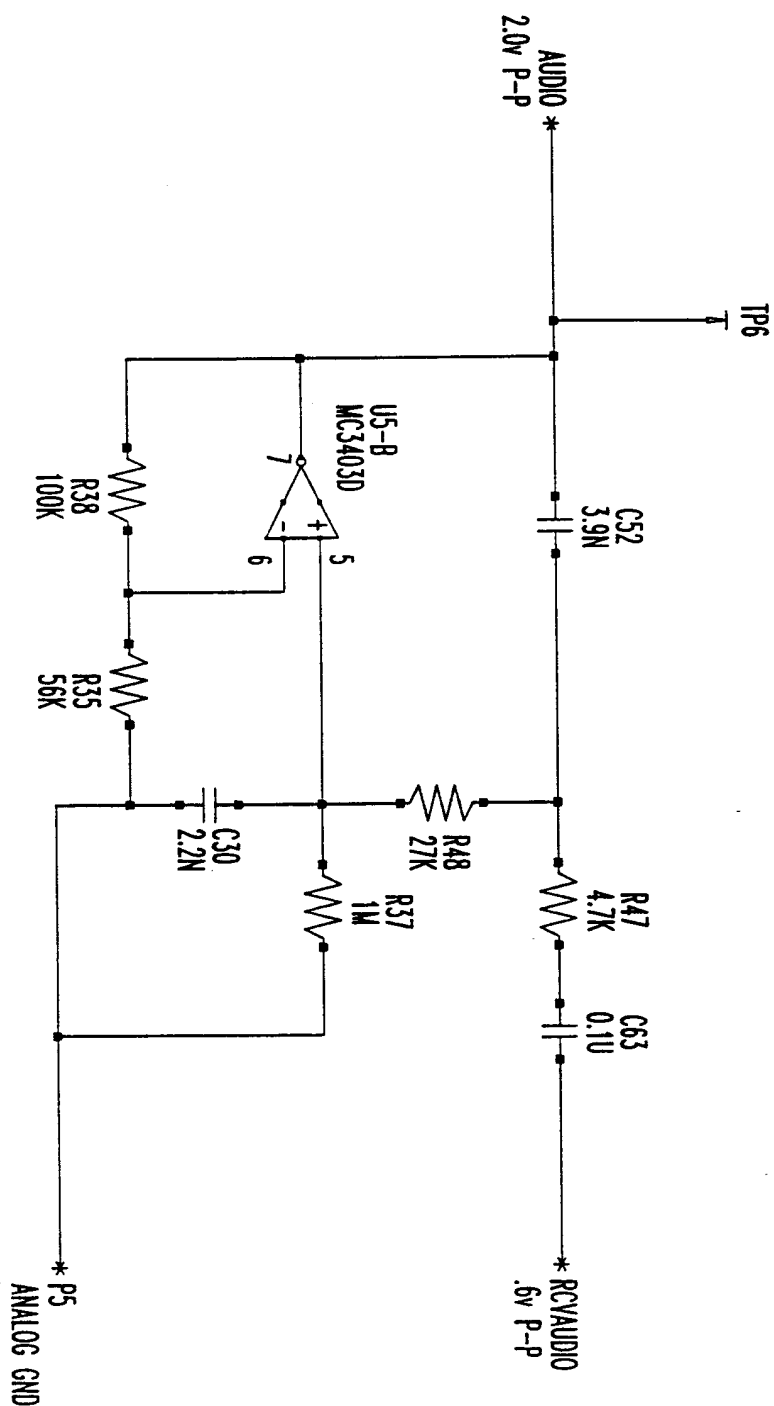
(請先閱讀背面之注意事項再填寫本頁)

裝

訂

後

五、發明說明 (39)



(請先閱讀背面之注意事項再填寫本頁)

裝

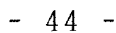
訂

矣

4 KHZ LOW PASS
第 14 圖

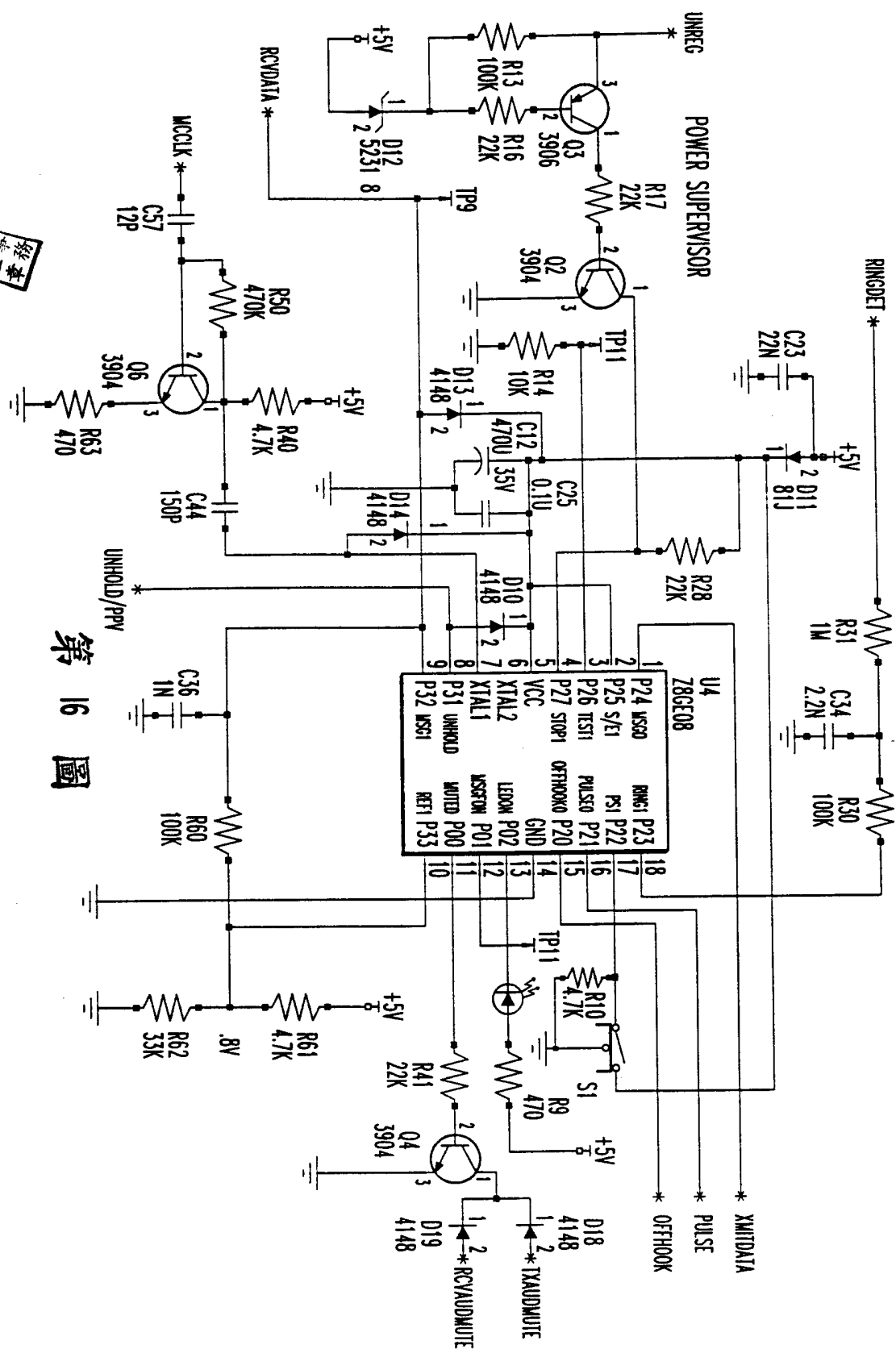
經濟部中央標準局員工消費合作社印製

第 15 圖



本紙張尺度適用中國國家標準 (CNS) A4規格 (210×297公釐)

第 16 圖



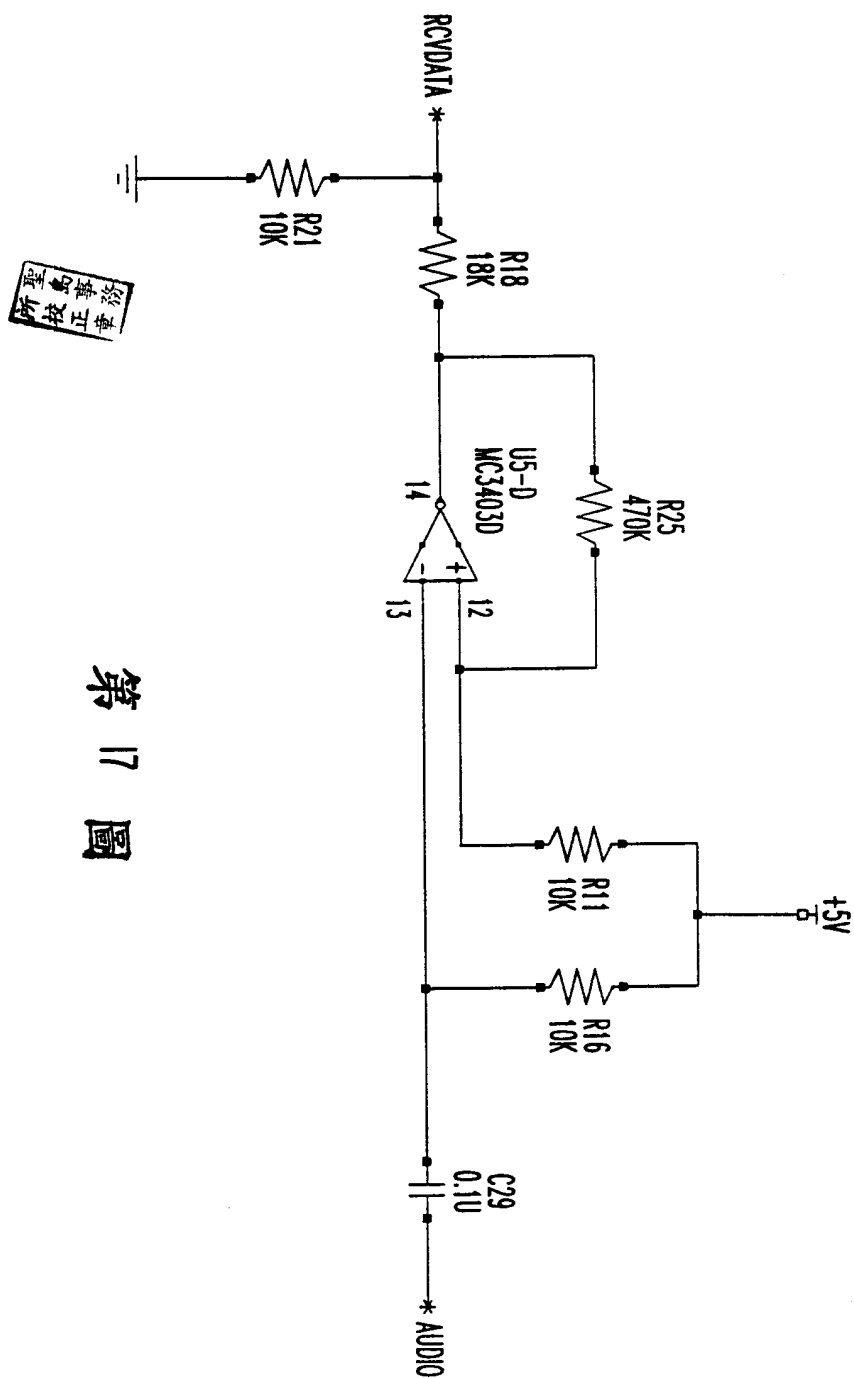
五、發明說明 (42)

(請先閱讀背面之注意事項再填寫本頁)

裝

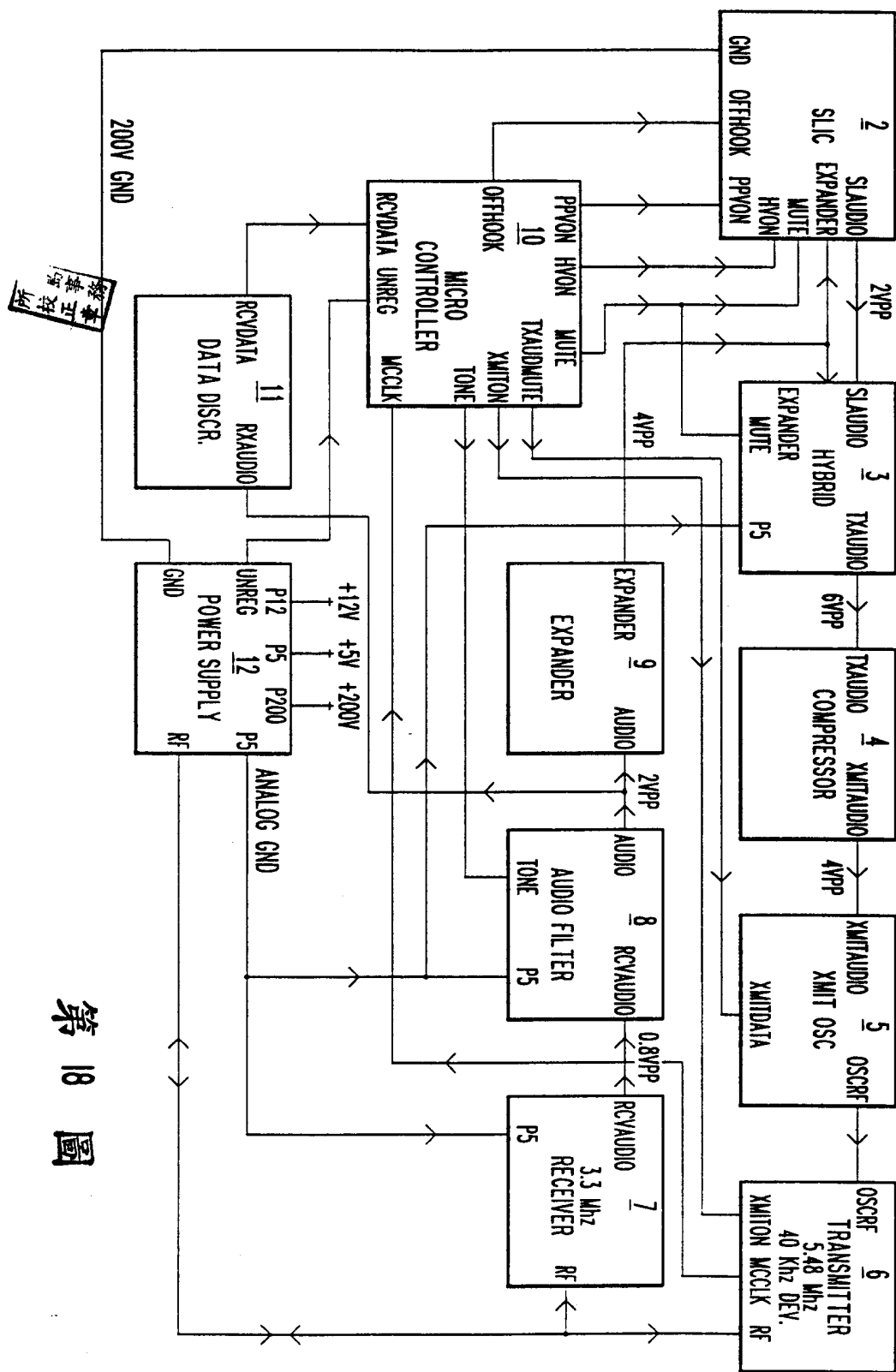
訂

系



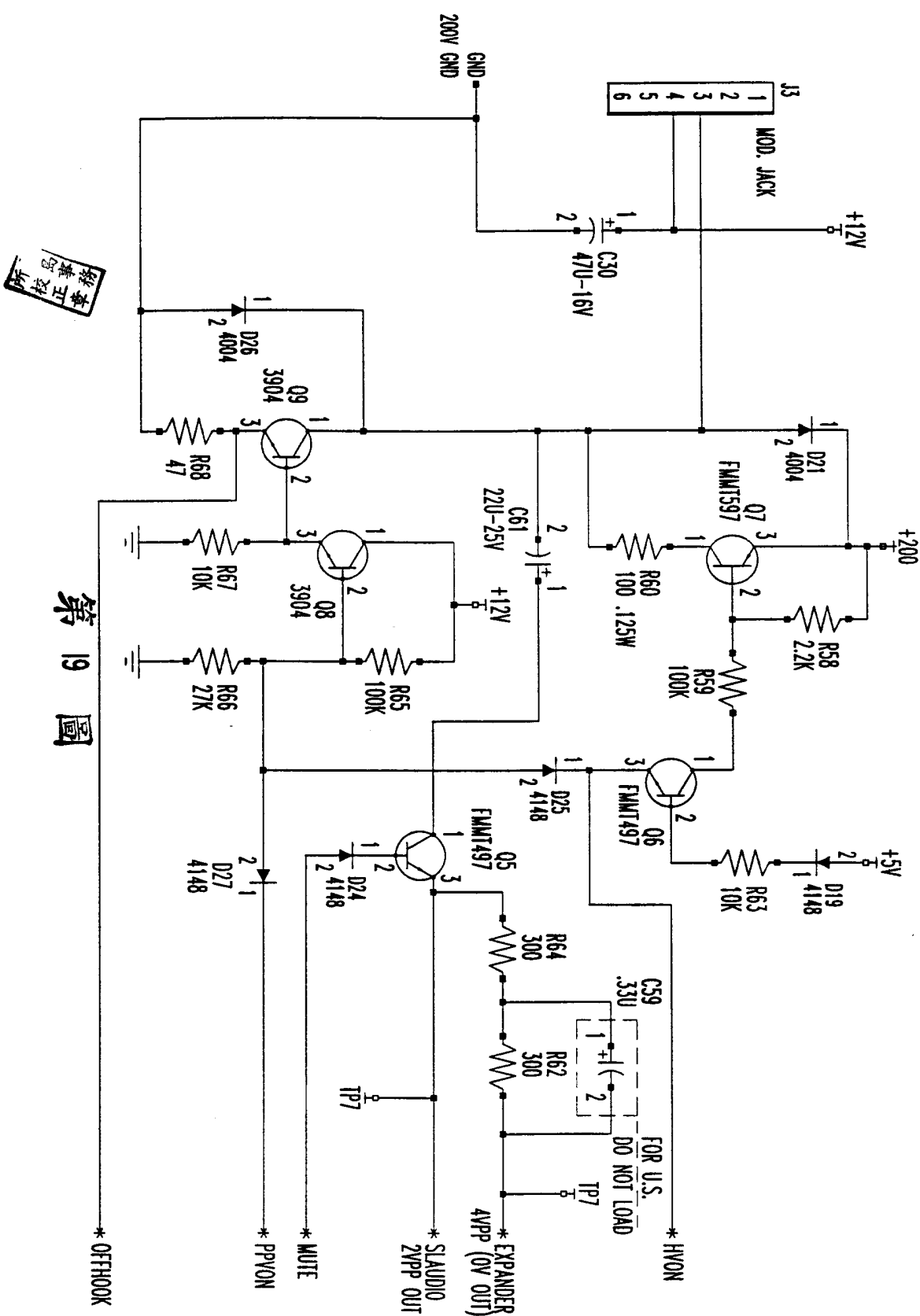
第 17 圖

五、發明說明 (43)

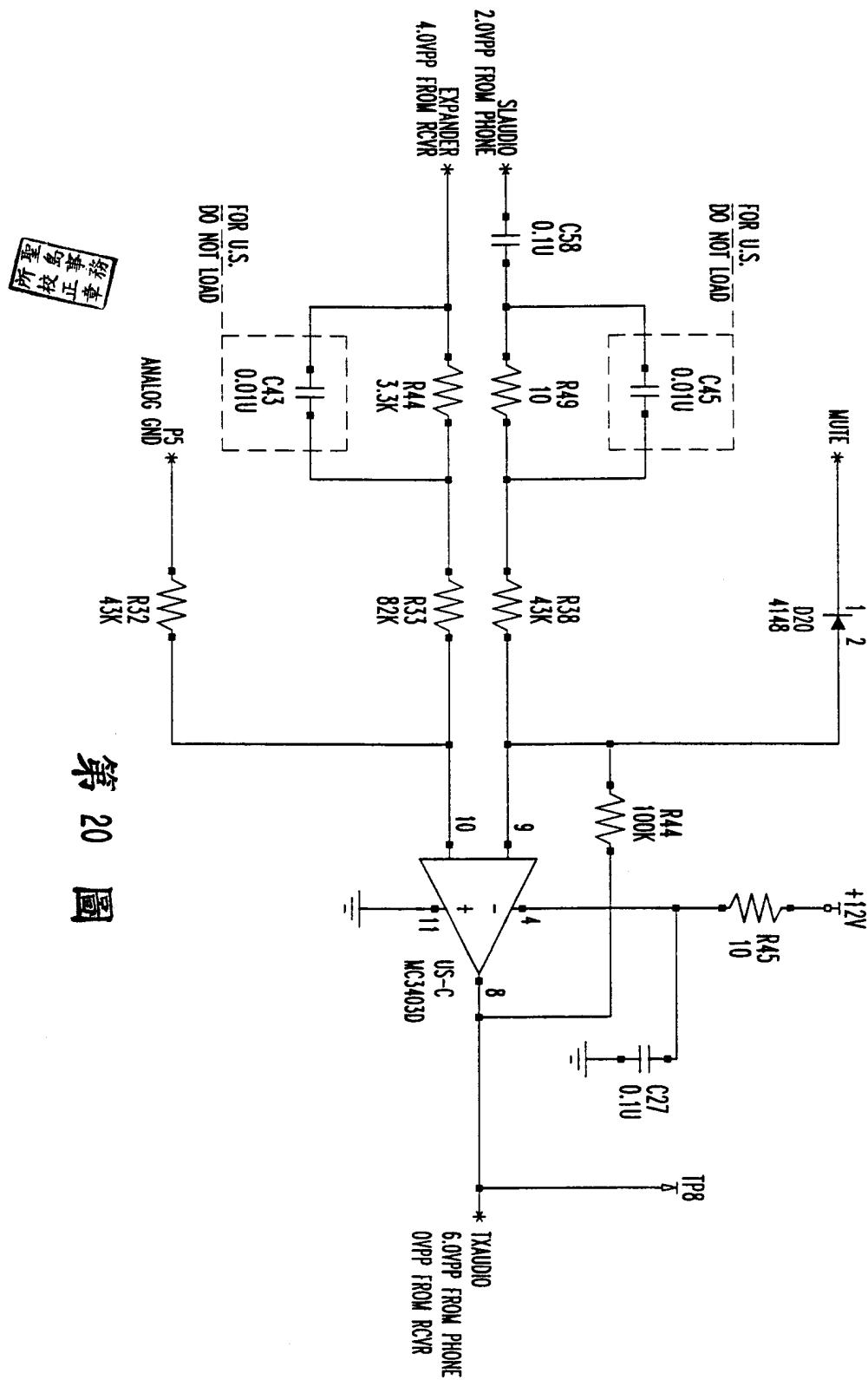


第 18 圖

五、發明說明 (44)



五、發明說明 (45)

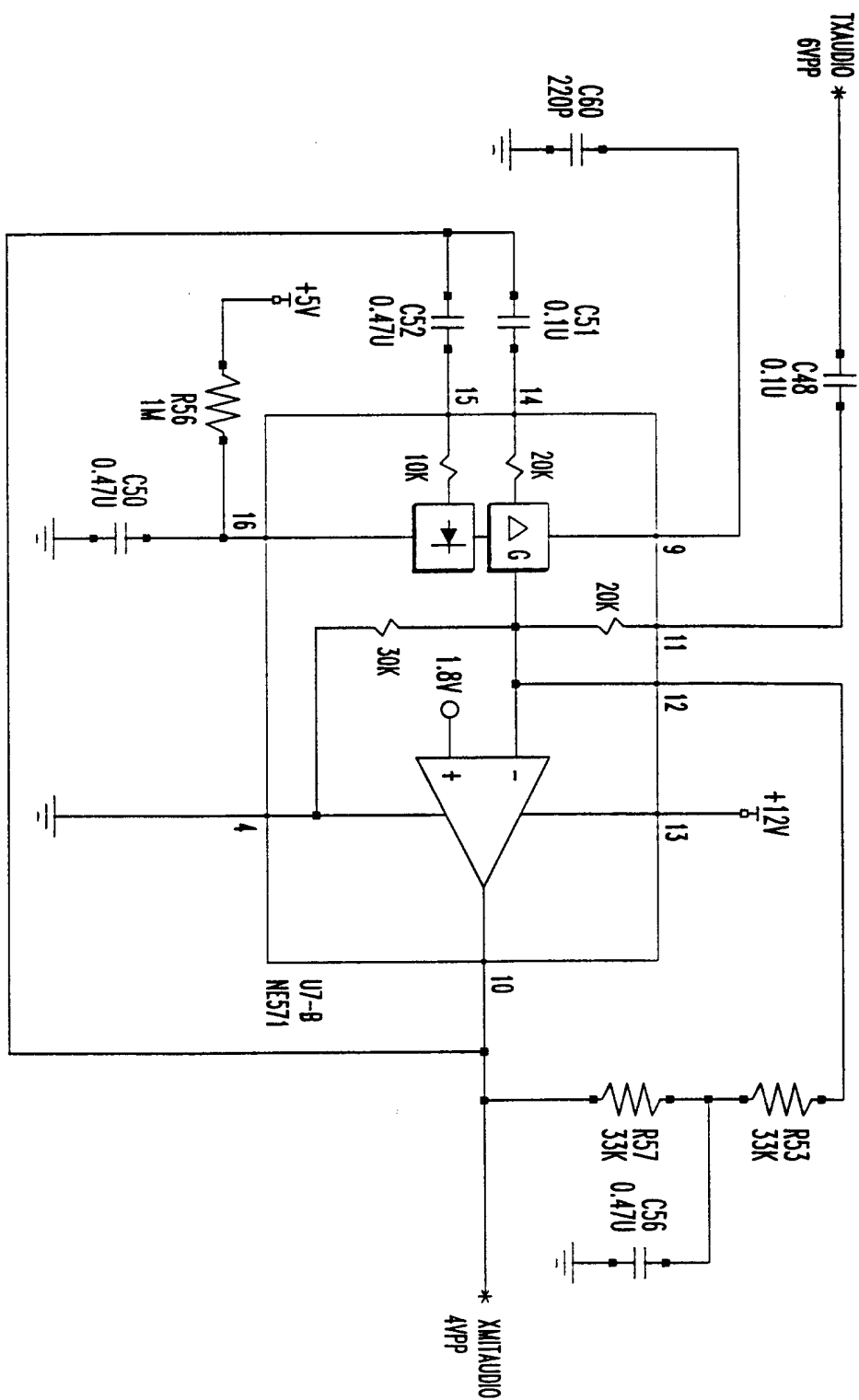


第 20 圖

五、發明說明 (46)

所聖校高正華務

第 21 圖



(請先閱讀背面之注意事項再填寫本頁)

裝

訂

後

- 51 -

第 22 圖



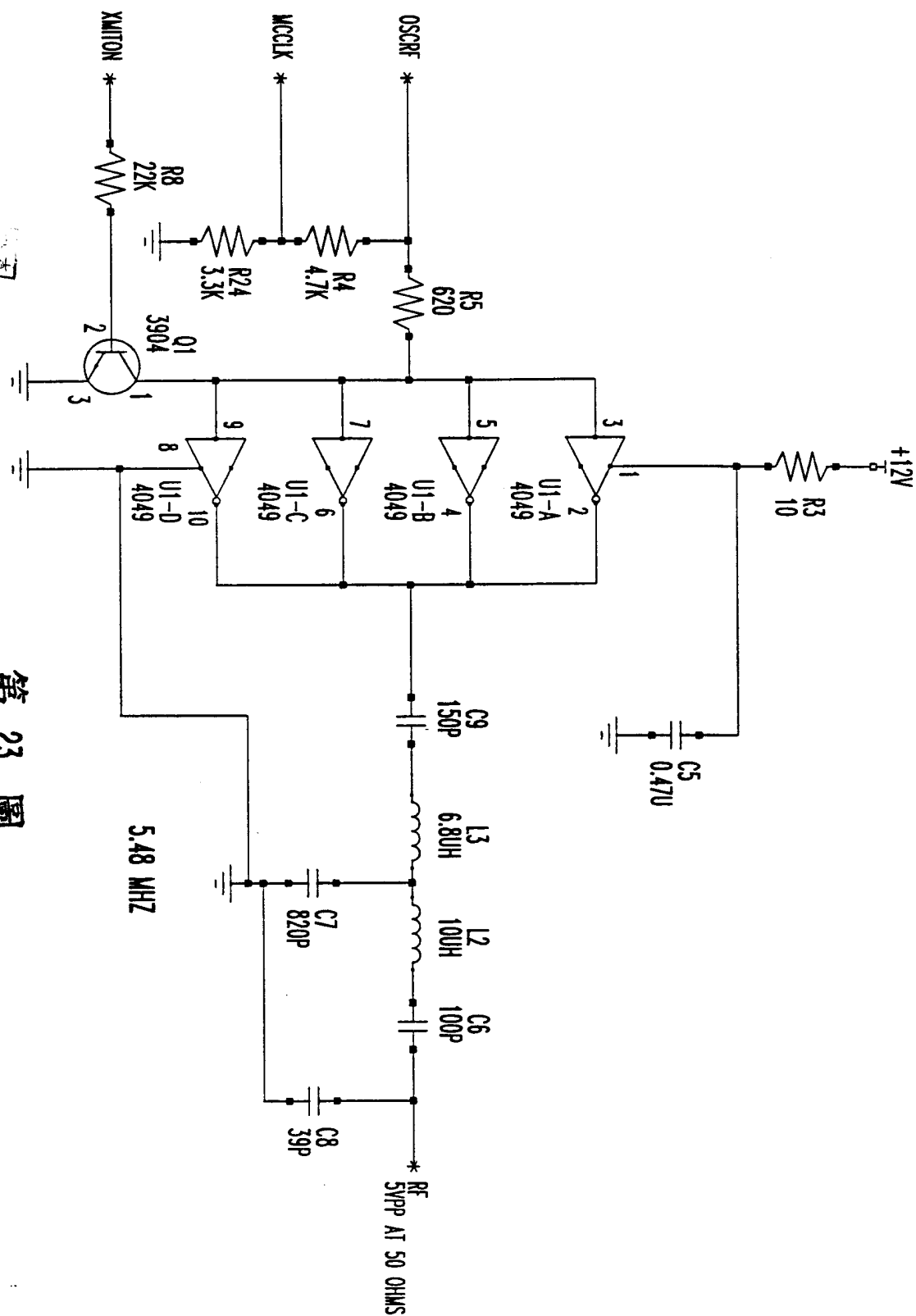
五、發明說明 (48)

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

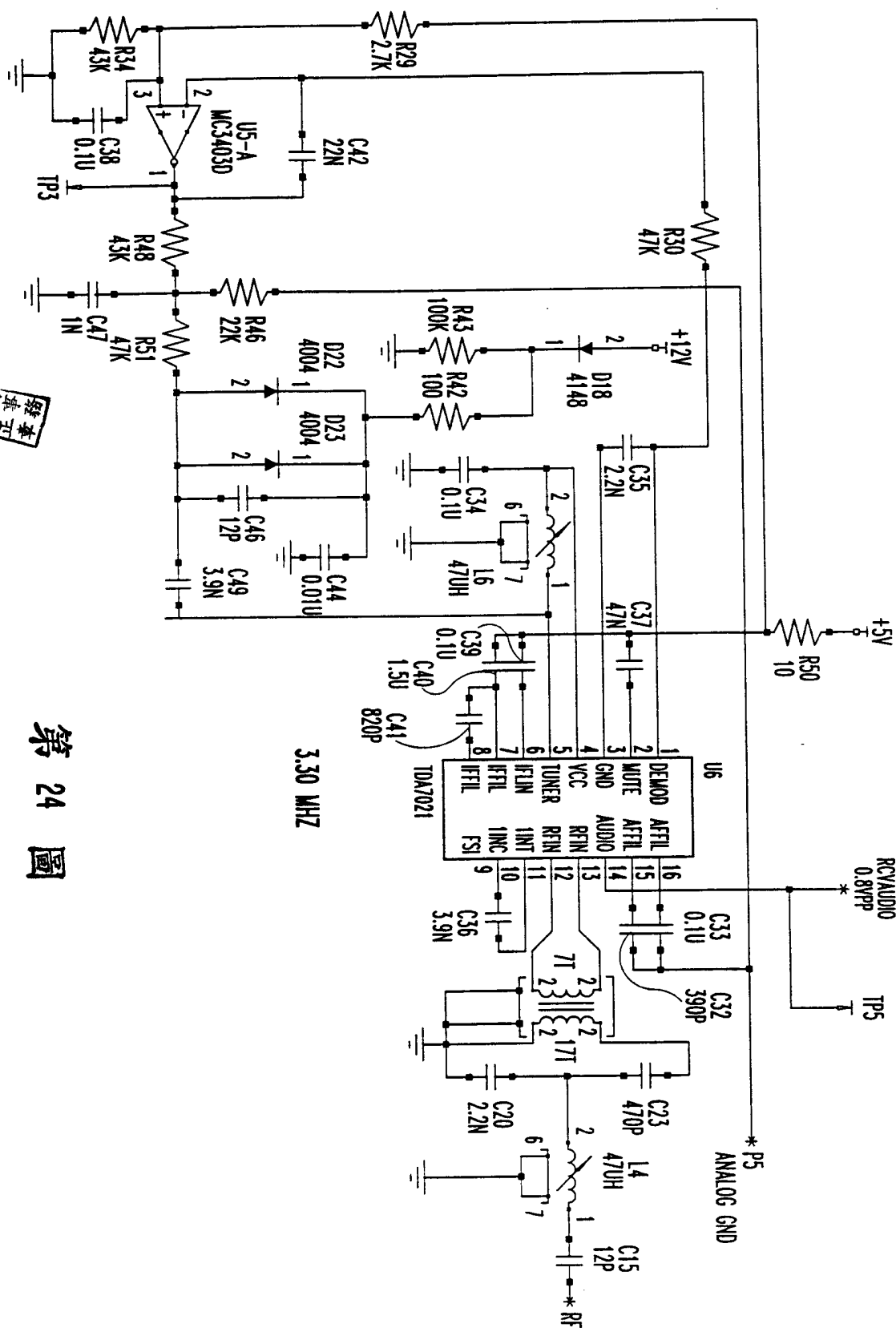
矣



第23圖

(請先閱讀背面之注意事項再填寫本頁)

五、發明說明 (49)

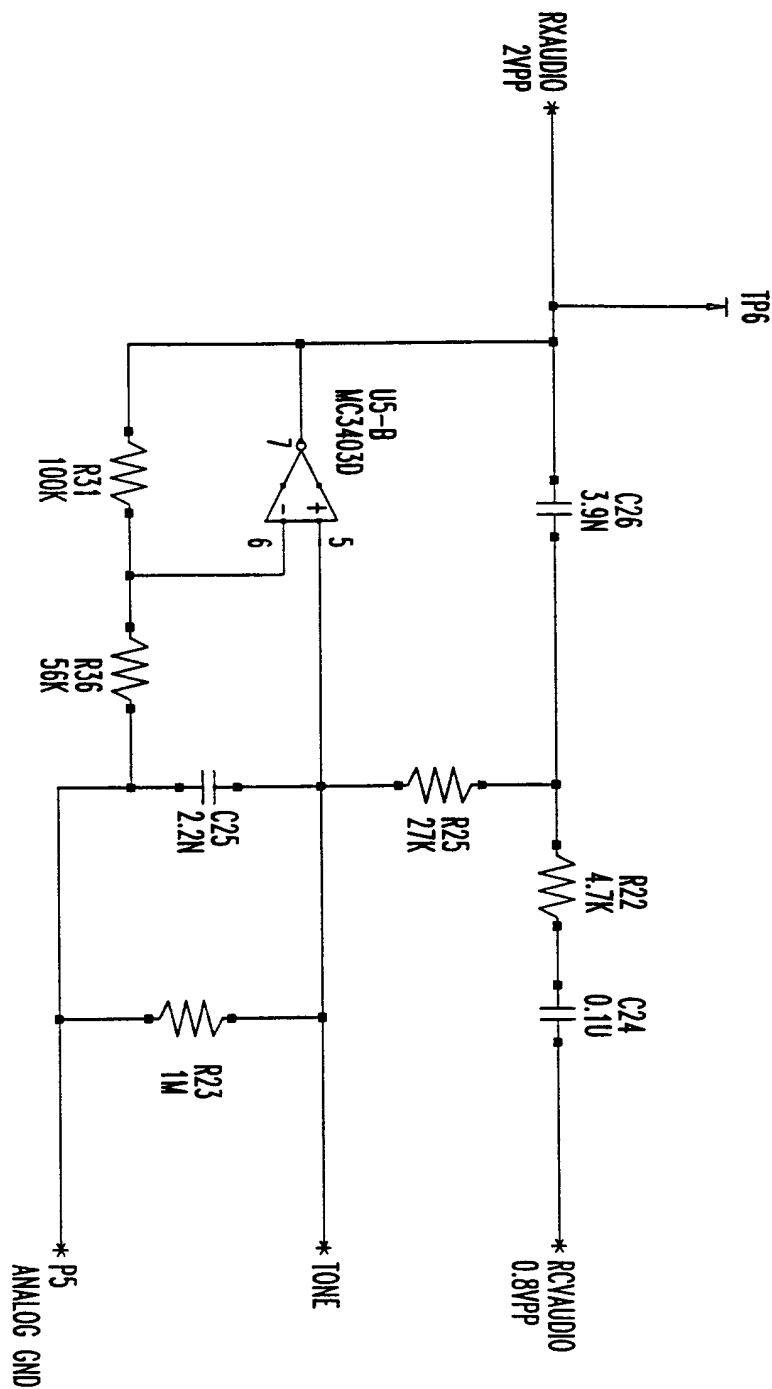


第24圖

五、發明說明 (50)



4 KHZ LOW PASS
第 25 圖



(請先閱讀背面之注意事項再填寫本頁)

裝

訂

系

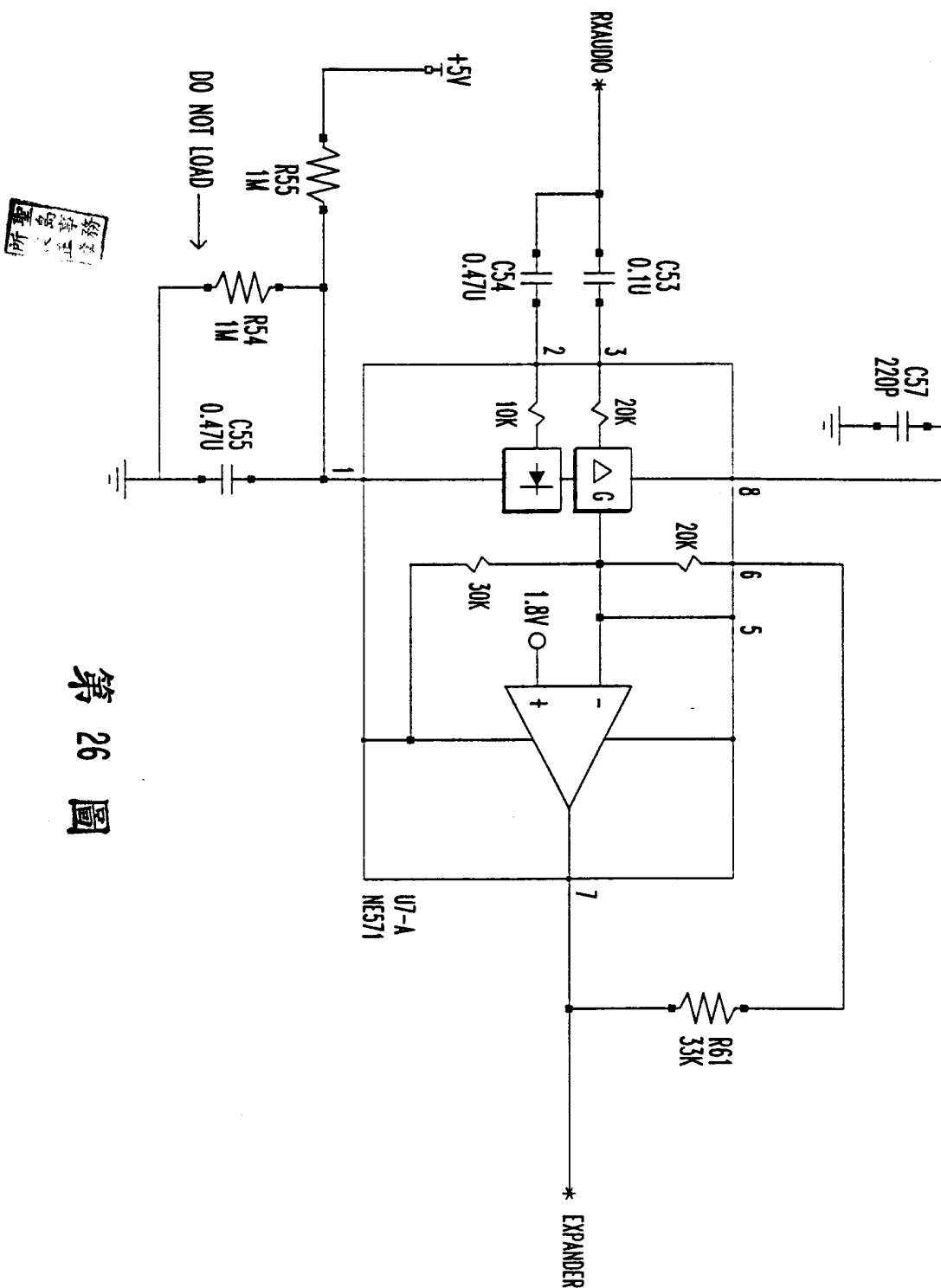
五、發明說明 (51)

裝

訂

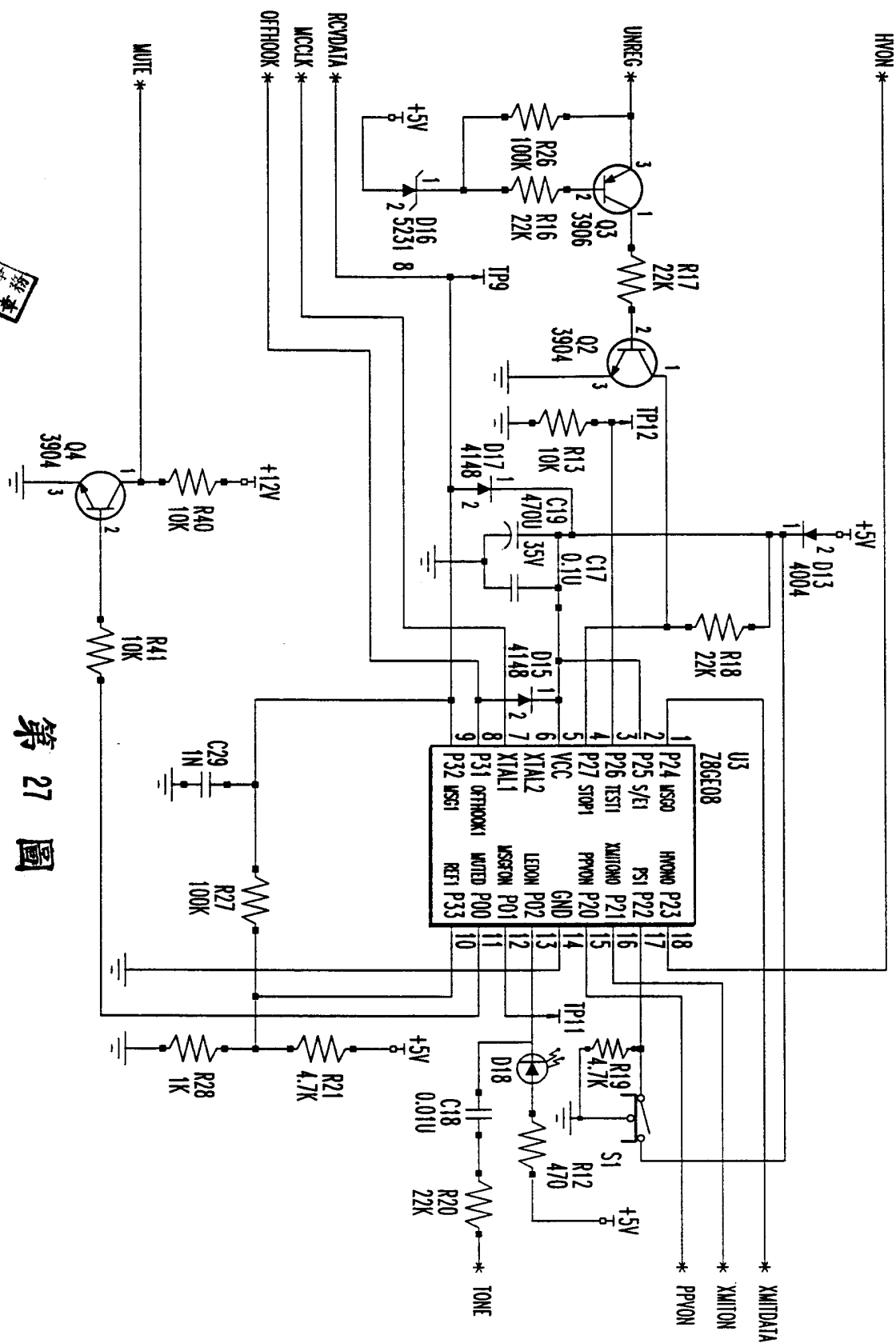
(請先閱讀背面之注意事項再填寫本頁)

※



第 26 圖

五、發明說明 (52)



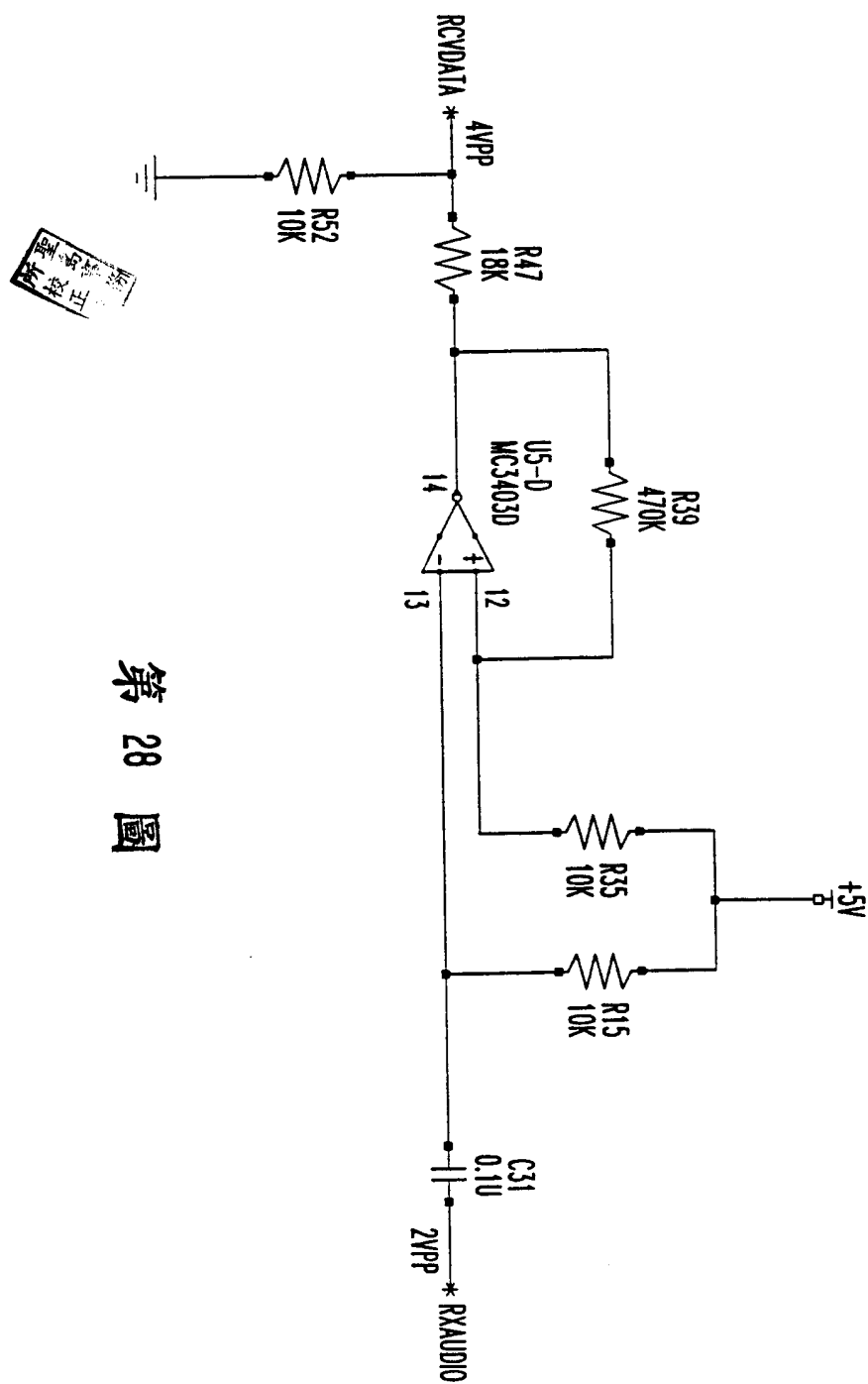
第 27 圖

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明 (53)



第 28 圖

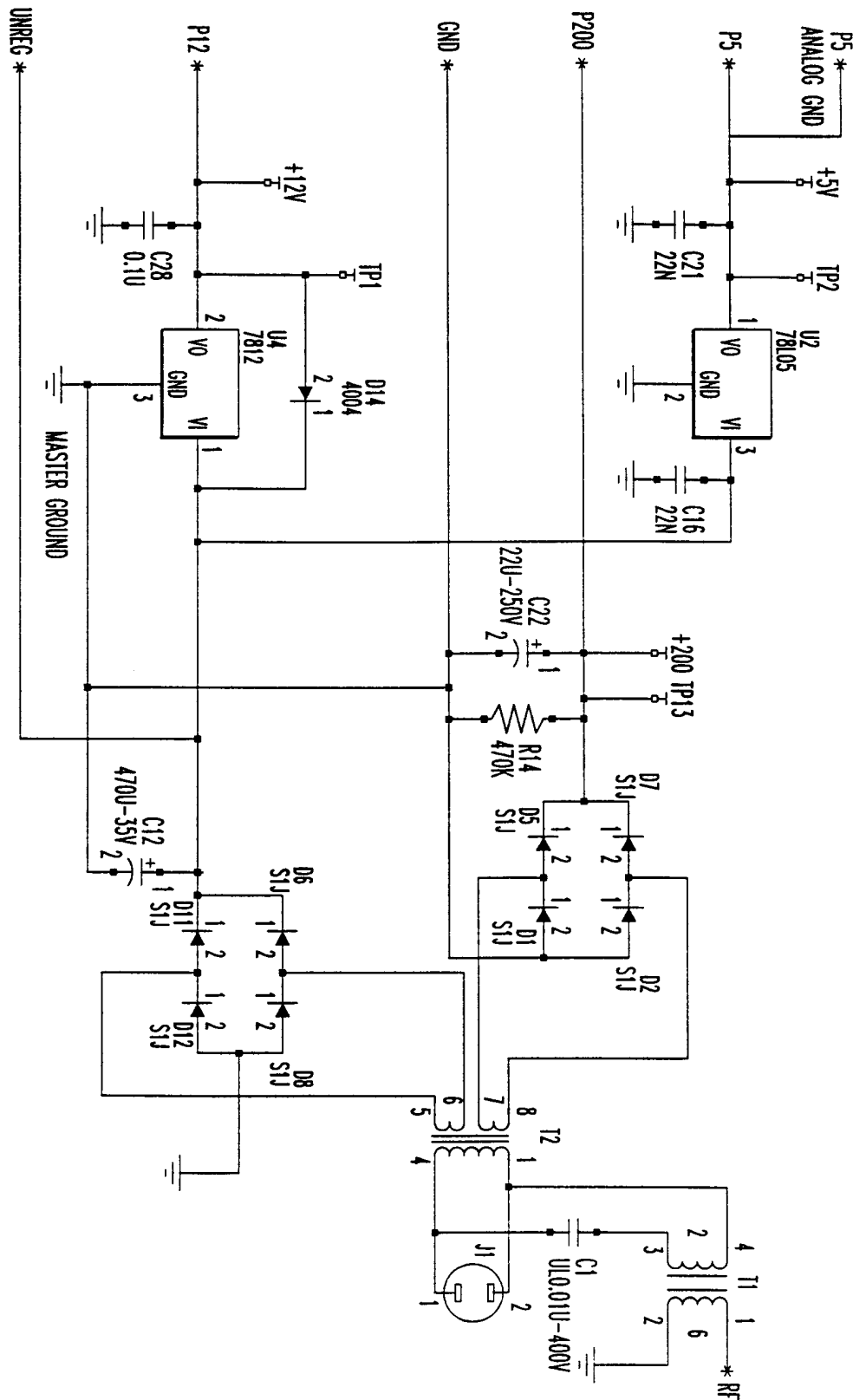
(請先閱讀背面之注意事項再填寫本頁)

裝

訂

急

五、發明說明 (54)



第 29 圖

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

為

四、中文發明摘要(發明之名稱：安全接通電話分機系統與方法)

一種方法與系統被提供來以全雙工在建築物之電力線路上進行安全電力線路載波通訊。電力線路載波電流電話分機系統利用：(1)每一條中央局線路一組單一基座單元(供介接線路載波信號與中央局線路)、與(2)一或多組分機單元，每一部分機電話機配合一組分機單元(供介接該等線路載波信號與該等分機電話機)。藉由在各基座與分機單元中使用一獨特及匹配安全鑰匙碼(導碼)的一組數位發訊系統，此方法運用系統控制邏輯來提供：A)多分機仲裁邏輯、B)防偷聽之私密性、與C)防第三者取用中央局線路之安全性；其中每一訊息指揮系統改變操作狀態，諸如：A)在通話期間，一分機單元可被置於保留狀態而使得另一部匹配的分機單元可獲取線路通話權，B)一基座單元發出振鈴信號給所有匹配的分機單元以通知有電話進來，或

(下頁續)

英文發明摘要(發明之名稱：Secure Access Telephone Extension System and Method)

A method and system is provided for conducting secure power line carrier communications in full duplex over the power lines of a building. Power line carrier current telephone extension systems utilize 1) a single BASE unit per central office line (for interfacing line carrier signals with the central office line) and 2) one or more EXTENSION units, one for each extension phone set (for interfacing said line carrier signals with said extension phone sets). The method employs system control logic for providing A) multiple extension arbitration logic, B) privacy from eavesdropping and C) security from third party central office line capture by means of a digital messaging system in each base and extension unit utilizing a unique and matching security key (preamble) with each message instructing the system to change

--To be continued on next page--

四、中文發明摘要(發明之名稱：)

(接上頁)

C)由一位使用者拿起某一部匹配之分機電話機之話筒成接聽狀態以獲取中央局線路而進行撥出電話動作。諸隨機安全碼的匹配動作可由使用者發起。此即意味著，同一棟建築物內之其他使用者以類似的線路載波電話分機產品未經許可地取用中央局線路（特別是用於打長途電話）、以及偷聽通話內容之行為，均被防止。

(請先閱讀背面之注意事項再填寫本頁各欄)

裝

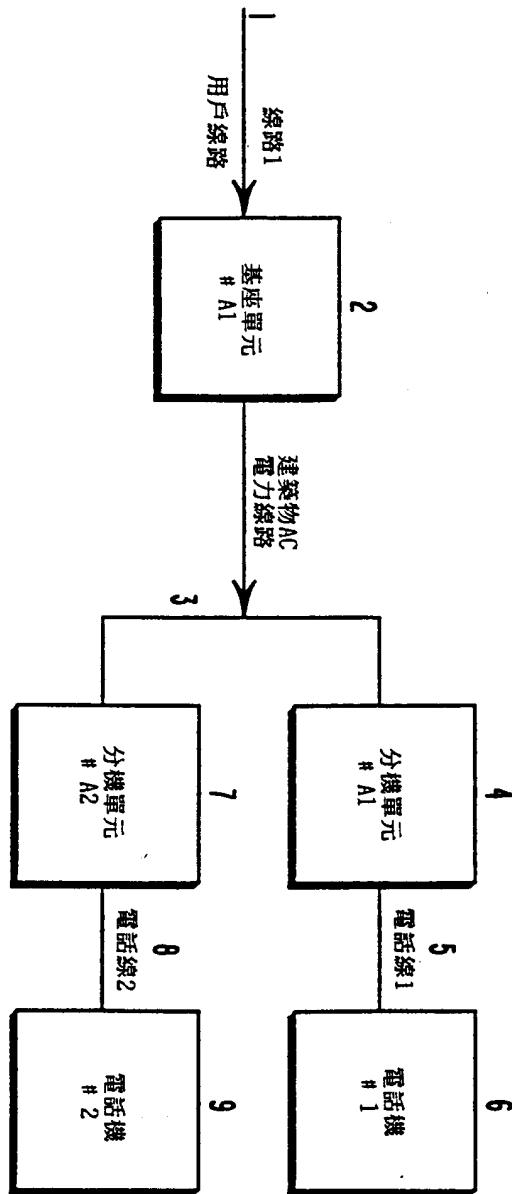
訂

線

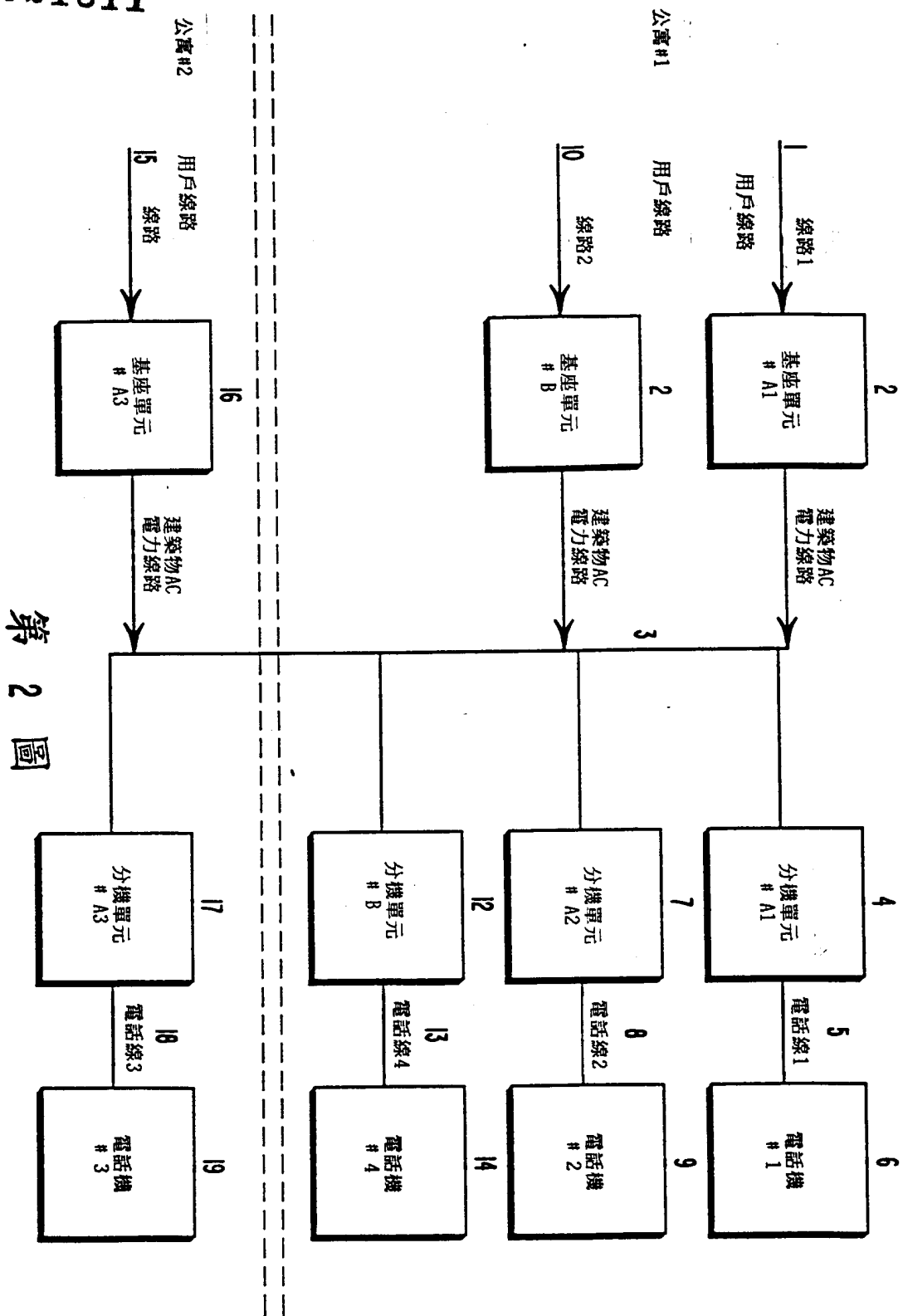
英文發明摘要(發明之名稱：)

operating states, such as, A) during a conversation an extension unit may be put on hold so that a second matching extension can pick up the line, B) a base unit rings matching extension units to signal an incoming call or C) an outgoing call is initiated by a user taking a matching extension phone off hook to access the central office line. Matching of the random security codes may be initiated by the user. By this means unauthorized access to the central office line (especially for long distance calls) and eavesdropping by other users in the same building with similar line carrier telephone extension products is prevented.

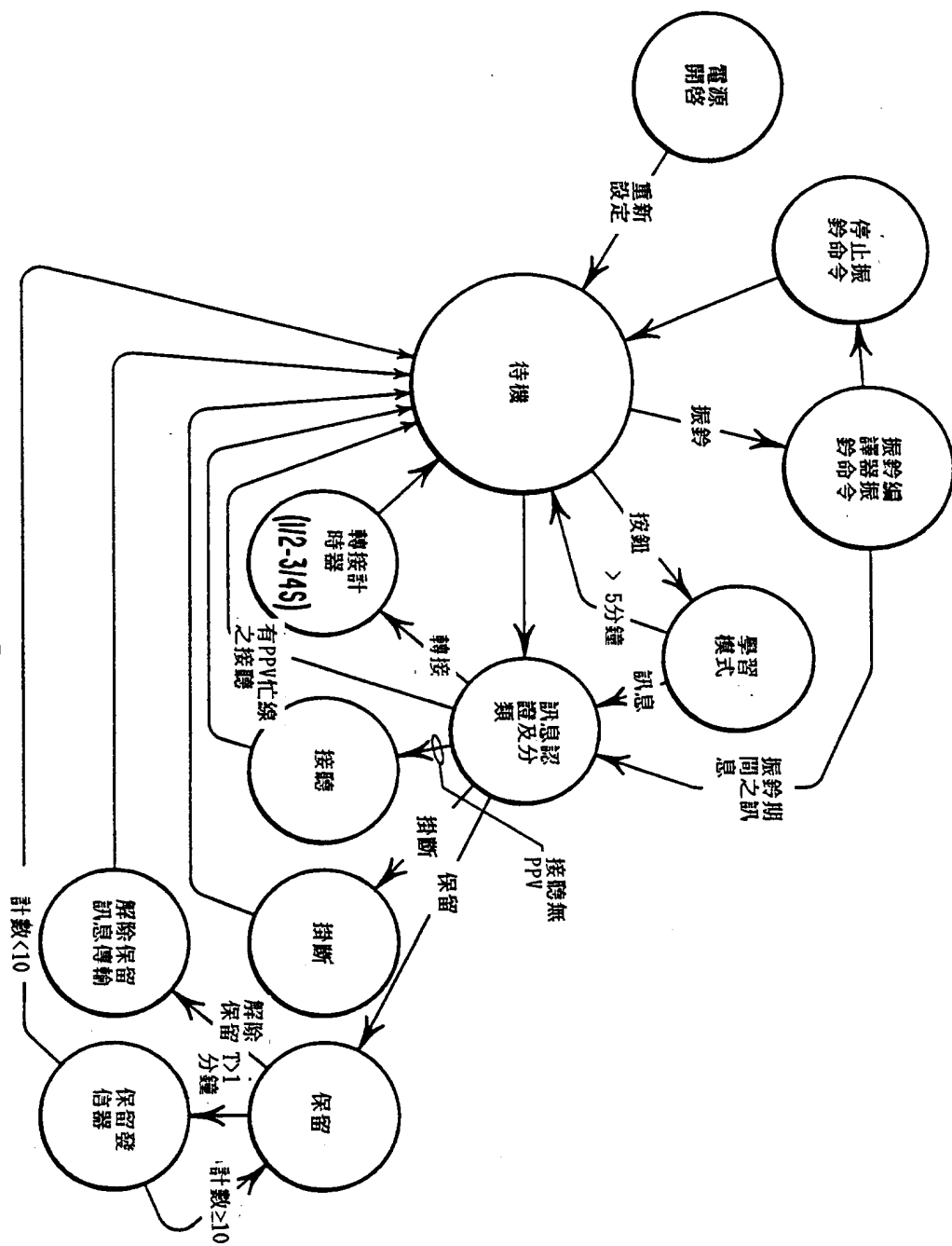
中譯圖



第一圖

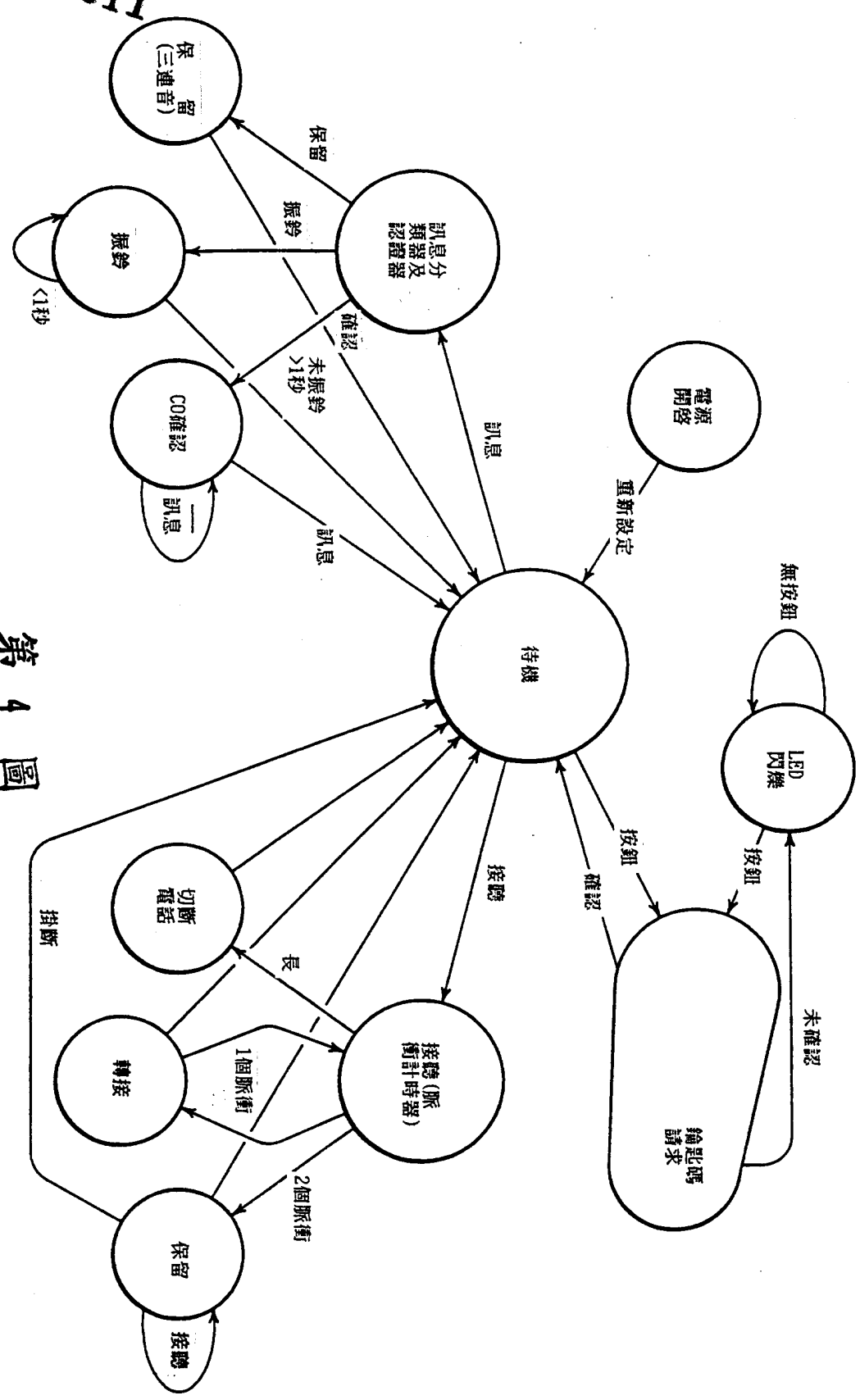


第 2 圖



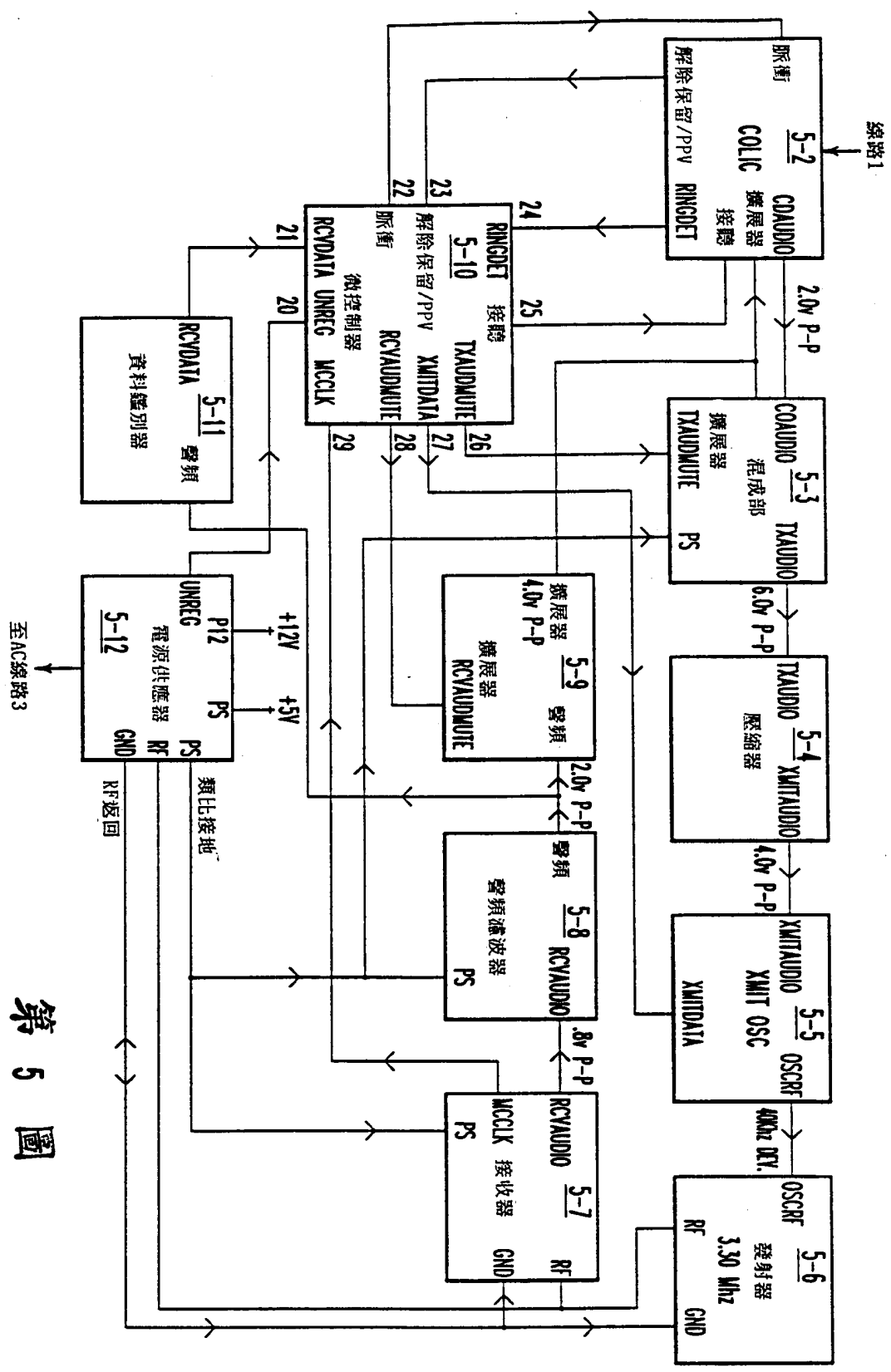
第 3 圖

321811

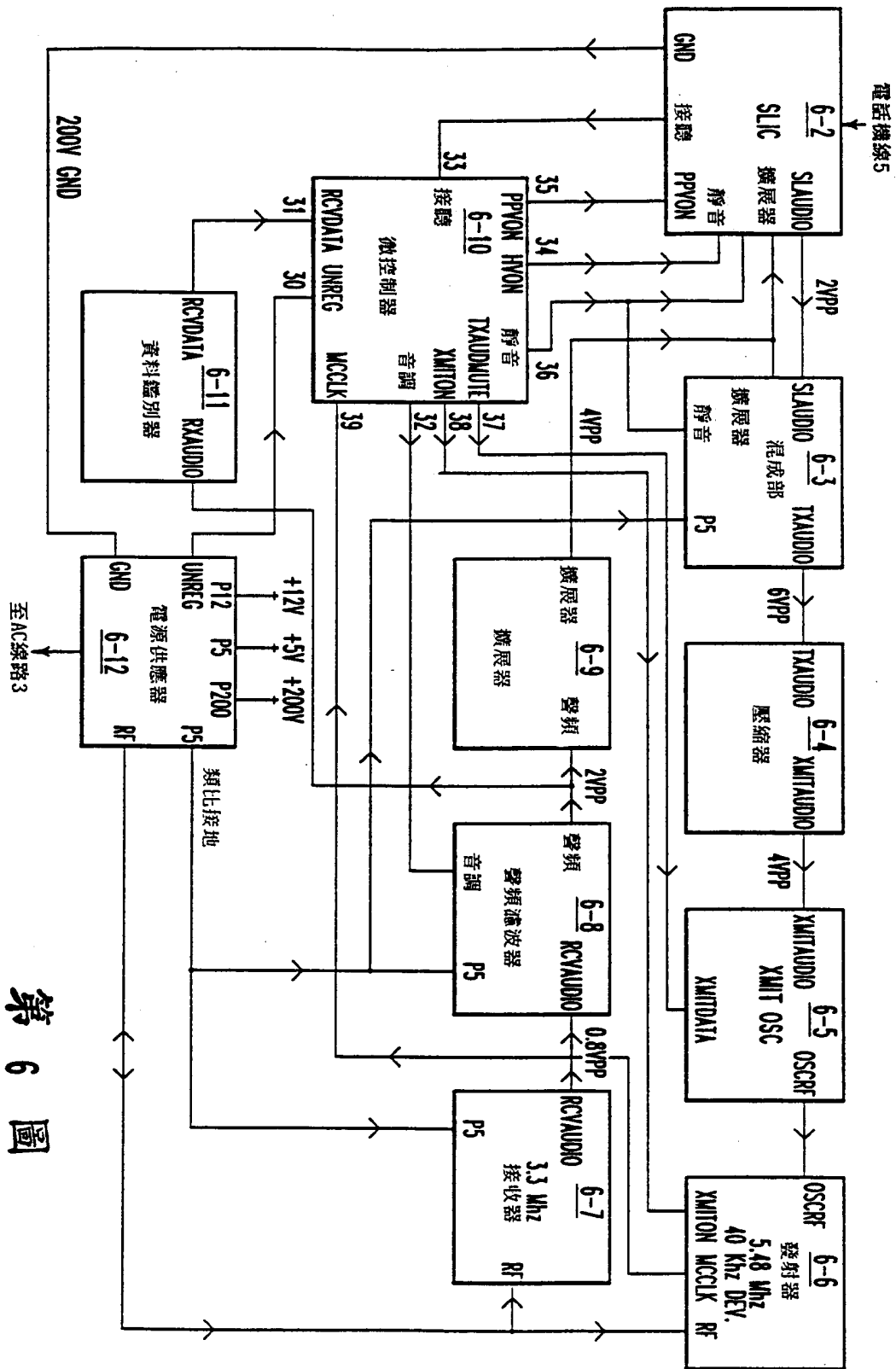


第 4 圖

321811



第 5 圖



第 6 圖

六、申請專利範圍

第84112430號申請案申請專利範圍修正本 85.09.20.

1. 一種電話分機系統之安全接達的方法，該電話分機系統具有：

一基座單元，其包含有：

用以耦合至一用戶電話線路之裝置；

用以與一或多個分機單元通訊之裝置；及

用以產生一隨機安全鑰匙碼之裝置；以及

至少一分機單元，其包含有：

用以與該基座單元通訊之裝置；及

用以儲存一安全鑰匙碼之裝置；

該方法包含之步驟有：

該基座單元產生一隨機安全鑰匙碼；

該分機單元自該基座單元要求該隨機安全鑰匙碼之拷貝；

該基座單元僅在該產生該隨機安全鑰匙碼的一預定時間內由該分機單元作出要求的情況下才將該隨機安全鑰匙碼的拷貝送回至該分機單元；

及

該分機單元將該隨機安全鑰匙碼儲存於該儲存裝置內。

2. 如申請專利範圍第1項所述之方法，其更包含有：

該分機單元在與該基座單元通訊之前傳輸儲存於該儲存裝置內之該隨機安全鑰匙碼的拷貝；以及

該基座單元只在該安全鑰匙碼之該拷貝和該隨機

(請先閱讀背面之注意事項再填寫本頁)

訂

六、申請專利範圍

安全鑰匙碼匹配時才容許與該分機單元通訊。

3. 如申請專利範圍第 1 項所述之方法，該基座單元更包含有：

用以起始該產生一隨機安全鑰匙碼動作的裝置，其可被該電話分機系統之使用者操作；以及

當該用以起始該產生一隨機安全鑰匙碼動作之裝置被使用者操作時，該產生一隨機安全鑰匙碼的步驟即發生。

4. 如申請專利範圍第 3 項所述之方法，其中該用以起始該產生一隨機安全鑰匙碼動作之裝置係一在該基座單元上的一按鈕。

5. 如申請專利範圍第 1 項所述之方法，該分機單元更包含有：

用以起始該隨機安全鑰匙碼之拷貝之要求的裝置，其可被該電話分機系統之使用者所操作；以及

當該用以起始該隨機安全鑰匙碼拷貝之要求的裝置被使用者操作時，即發生要求該隨機安全鑰匙碼拷貝的步驟。

6. 如申請專利範圍第 5 項之方法，其中該用以起始該隨機安全鑰匙碼拷貝之要求的裝置係為在該分機單元上的一個按鈕。

7. 一種電話分機系統之基座單元，該電話分機系統具有該基座單元及至少一分機單元，該基座單元包含有：

用以耦合至一用戶電話線路之裝置；

(請先閱讀背面之注意事項再填寫本頁)

訂

六、申請專利範圍

用以與一或多個分機單元通訊之裝置；
用以產生一隨機安全鑰匙碼之裝置；
用以接受自一分機單元要求該隨機安全鑰匙碼拷
貝的裝置；及

用以僅在該隨機安全鑰匙碼產生的一預定時間內
由該分機單元作出要求的情況下才將該隨機安全鑰匙
碼的拷貝送回至該分機單元的裝置。

8.如申請專利範圍第7項所述之基座單元，其更包含有
：

用以僅在該分機單元已傳輸該隨機安全鑰匙碼之
拷貝之情況下才允許與該分機單元通訊之裝置。

9.如申請專利範圍第7項所述之基座單元，其更包含有
：

用以起始該產生一隨機安全鑰匙碼之動作的裝置
，其可被該電話分機系統之使用者操作。

10.如申請專利範圍第9項所述之基座單元，其中該用以
起始該隨機安全鑰匙碼產生動作之裝置係為在該基座
單元上的一按鈕。

11.一種電話分機系統之分機單元，該電話分機系統具有
一基座單元及該分機單元，該分機單元包含有：

用以與該基座單元通訊之裝置；及

用以儲存一安全鑰匙碼之裝置；

用以自該基座單元要求隨機安全鑰匙拷貝之裝置
，該基座單元僅在其產生該隨機安全鑰匙碼後一預定

六、申請專利範圍

時間內由該分機單元作出要求的情況下才將該隨機安全鑰匙碼的拷貝送回至該分機單元；及

用以儲存該隨機安全鑰匙碼之裝置。

12. 如申請專利範圍第11項所述之分機單元，其更包含有：

用以在與該基座單元通訊之前傳輸儲存的隨機安全鑰匙碼拷貝的裝置，且該基座單元僅在該安全鑰匙碼拷貝與該隨機安全鑰匙碼匹配的情況下才允許與該分機單元通訊。

13. 如申請專利範圍第11項所述之分機單元，其更包含有：

用以起始該隨機安全鑰匙碼之拷貝之要求的裝置，其可被該電話分機系統之使用者所操作。

14. 如申請專利範圍第13項所述之分機單元，其中用以起始該隨機安全鑰匙碼拷貝之要求的裝置係為在該分機單元上的一個按鈕。

(請先閱讀背面之注意事項再填寫本頁)

訂

竣