THE DESIGN OF PABX WITH LAN ARCHITECTURE

Ko Wing Hoi

Information Engineering, CUHK

Abstract : This report first discusses the present status of PABX and LAN technology for voice and data applications. It then proposes an integrated PABX/LAN system for the combined functions of telephony and data communications. The software and hardware architectures of the integrated system are described.

June, 1992.

THE DESIGN OF PABX WITH LAN ARCHITECTURE

C O N T E N T S

1. INTRODUCTION

In this information age, the telephone has become a powerful instrument for the communication of information, and with the advent of display phones, it has become a recipient and processor of information. Similarly, over the past twenty years, the computer has evolved from an expensive, highly specialized device to a very inexpensive tool that has pervaded the workplace. Within the next ten years, it will become as commonplace as the telephone.

This creates two new demands on communication technology. First, more advanced and specialized functions will be demanded from the telephone to meet the need for increased integration of voice and data, and to provide more automatic responses to the additional information provided to the telephone user. Secondly, as every desk becomes equipped with a telephone and a computer, there arises a need for an integrated voice and data desk where the computer and telephone function as a single communication instrument.

For the voice communications, public telephone networks have penetrated into almost anywhere on the earth. For the last twenty years, there has seen a tremendous growth of private networks of telephones, known as Private Automatic Branch Exchange (PABX), to provide a more flexible and efficient use of telephones in small organizations.

For data communications between computers, Local Area Networks (LAN) are becoming more and more popular.

Since these two networks are used for the communications between different devices, it should save a lot of troubles in installing and maintaining just one rather two separate networks. Presently, the major direction of development is the integration of the networks already in operation into one network which can satisfy all the different requirements with respect to application profiles and their required qualities of services. This is motivated not only by the maintenance costs of different existing networks operating in parallel, but also by the variety of applications one may imagine in a multifunctional environment. However, these two networks are employing very different switching and networking technologies that combining the two together is not an easy task. Thus a network is proposed to incorporate both telephones and computers so that voice and data communications can be performed in an integrated environment.

In the next few chapters, such an integrated PABX/LAN system incorporating both computers and telephones will be discussed.

2. COMPARISONS OF LAN AND PABX

Before we try to attempt to design an integrated system involving computers, workstations and telephones, the most widely used network systems for computers and telephones are discussed. Their architectures, merits and demerits for voice and data applications will be compared.

2.1 Typical LAN system

With the widespread use of computers and workstations these days in almost every office and factory, LAN's are used to connect a multitude of computers to work efficiently and cost effectively.

2.1.1 Characteristics of a LAN [1]

1. Area covered

The network is restricted to an area of moderate size, such as an office block, a factory, or a campus. The limiting factors are usually the overall length of the cable used and any interdevice restrictions imposed. In practice the distances involved range from a few meters to a few kilometers. A very few can cover tens of kilometers, but are still described as local networks.

2. Speed

The raw data transmission rate on the network is high (of the order of 1 - 100 Mbps) when compared with ordinary telecommunications. 10 to 20 Mbps is the most common data transmission rate in LAN's today.

3. Connectivity

On some LAN's, particularly those associated with open systems, every device on the network has the potential to communicate with every other device on the same network.

4. Cost

Relatively inexpensive methods are used to connect to the network, compared to the cost of the device being connected. The overall utilization of the system is improved as resource sharing and time-sharing between different user devices is possible. Thus the cost of the system is much reduced.

5. Resource sharing

One of the fundamental features of most LAN's is that the network is shared between the devices. Whereas on a traditional network such as PABX, each machine will be directly wired into a switching device, on a LAN the one physical medium is usually shared, along with access to the medium. Resource sharing is one of the many reasons to connect computers, workstations and storage devices and output devices into a single network.

6. Low error rates

The errors introduced by the network are very few, when compared with traditional wide area networks.

2.1.2 Transmission medium of LAN

1. Twisted pair

For general networking the most common transmission medium is twisted pair cable. It is the most common because it is used for the telephone network to link each handset to its local exchange. Twisted pair, which has the additional advantage of being easy to install, requiring little specialized skill, is being used very successfully in local area networks running at up to 10 Mbps over shielded twisted pair. With further delicate shielding techniques, twisted pair cable has the capacity to carry data up to 16 Mbps.

2. Coaxial cable

Coaxial cable is probably the most versatile for LAN's, offering high speeds over moderate distances at moderate prices. Coaxial cable consists of a single inner conductor and a hollow outer conductor, with insulating material between, enables it to have higher bandwidth and work at higher data rates than twisted pair. There are two types of coaxial cable in common use, 50-ohm and 75-ohm. The 50-ohm will support speeds of up to 10Mbps, whereas the 75-ohm will go up to 50 Mbps.

3. Fiber optic cable

A fiber optic cable consists of a very thin, flexible, glass or plastic strand, down which light can be transmitted by reflection off the tube's internal walls. Fiber optic cable can operate with much greater speeds and distances than coaxial cable. However, fiber optic transmission requires the conversion of the electrical signals from a station into light, and this creates difficulties in implementation and maintenance.

2.1.3 LAN access control methods

1. Polling

Polling relies on a master-slave relationship between a central server and the other stations on the LAN. The central server simply maintains a list of the stations assigned to share its disk space, and asks each on in turn if it has any transmission to make. It it has, that station is permitted to send one or more packets to the central station. The station must then wait until it is polled again before it can send any more.

2. Carrier sense multiple access

CSMA techniques are by far the most common used in LAN's. This method is restricted to bus and tree LAN's, and all the stations share the same physical medium, or the same channel. As all the stations have equal priority, there being no polling from a central station, a station must listen to the medium to detect if it is idle or not before starting its own transmission. The problem with this method is that two stations could decide to transmit simultaneously, both having detected an idle medium. Furthermore it takes some time for a signal from one station to reach all the other stations in the LAN, and so there is a time after one station starts to transmit before all the other stations have detected the transmission, and there will not themselves try to transmit.

3. CSMA/CD

(Carrier sense multiple access with collision detection)

When the LAN is idle the station transmits its own packet but at the same time keeps monitoring --- 'listening' to the network. It is possible that two or more devices decide to transmit at about the same time, and when this happens collision is said to occur. When a transmitting station detects collision, it stops transmitting the packet and sends out a very high jamming signal, to ensure that the collision will be detected by other transmitting stations. The transmitting stations involved in the collision will each stop trying to transmit for random time intervals, before trying again.

2.1.4 Interfacing to the LAN

There are two basic approaches to interfacing to a LAN. The first is for the LAN manufacturer to produce a Network Interface Unit (NIU), which presents a number of standard computer interfaces to the outside world. Typically these will be RS-232 (V.24) or IEEE 488 ports, and they enable almost any PC, terminal or peripheral to be attached. A simple program, usually a network management operating system, which can be totally independent of how the LAN itself works, can then be used by a PC attached to an NIU to gain access.

The second approach is to provide a card which connects the PC or larger host directly to the LAN, by plugging into the host's internal bus. The card may have all of the LAN software on board, or may require extensive software on the host, but this method is likely to be more efficient and is potentially cheaper for a large host.

2.1.5 LAN topology

1. Star networks

With the simple star topology, each station is connected to a central switch by a dedicated physical link. The switch provides a path between any two devices wishing to communicate, either physically in a circuit switch or logically in a packet switch. Star networks tend to imply simple attachment hardware at each station, although the switches themselves can be complex and therefore expensive.

2. Ring networks

A ring network is one where the stations are connected by a loop of cable, and each connection point, called a repeater, is responsible for passing on each fragment of the data. The data is sent in packets, and within each station there is a controller board responsible for recognizing packets sent to that station and for controlling access to the ring. Unlike star networks, in a ring network access is not under central control.

3. Bus networks

Bus networks are the most common LAN topology. They have no switches and, in their simplest form, no repeaters, but simply share a common, linear communications medium. Each station requires a tap (hardware for attachment to the medium), which must be capable of delivering the signal to all the stations on the bus. The data are sent in packets, and each station 'hears' all the transmissions, picking up those addressed to it.

2.1.6 Switching techniques

1. Circuit Switching

Circuit switching is the technique of dedicating to the communication a physical path between the transmitter and the receiver, the path being termed the circuit. Communication requires three phases; setting up the circuit, transferring the data and closing down the circuit.

For station A to transfer data to station B it must ask its nearest node, say node 1, to set up a circuit to station B. Station B is identified by an address, normally a unique number, just as every telephone has a unique number. Node 1 will choose the next leg of the route, say to node 3, and so on until station B is reached. The essential point about circuit switched networks is that once the circuit has been established, neither station should be able to detect any difference between the networked connection and a direct physical cable between the two stations. These networks are commonly used to provide connections between terminals and individual computer ports, although the most common example is the telephone network.

2. Message switching

In message switching there is no dedicated path, but the transmitter appends the address of the destination station to a complete unit of information, called a message, and transmits it on the network. These units can be very large, and each is completely independent of any other. Each switching node the network will look at the address and decide how to get the message to its destination. This is much more efficient than circuit switching, which can waste line by keeping them open when no data are being transferred, but introduces considerable delays in switching, and requires nodes with sufficient capacity to store complete message.

3. Packet switching

Packet switching is an attempt to compromise between message switching and circuit switching, and it comes in two flavors : datagram and virtual circuit. In both kinds the whole message is split up into smaller messages, called packets, which will typically be under 1K bytes long. In virtual circuit switching, a logical circuit, called a virtual circuit, between the transmitter and the receiver is set up.

Details of these switching techniques are discussed in Chapter 3 of Ref. [12].

2.2 Applications of LAN

Two types of traffic patterns in LAN :

1. File transfer

A user receives a file from a central storage facility known as a file server. The user requires a very high transmission capacity for a short time and then none at all.

2. Message transfer

Sending of isolated, synchronous messages between human users. Use of the network is brief, at constant access to the network for transmission to or reception from multiple partners is necessary.

2.2.1 Small filestore LAN's

The essence of a small filestore LAN is that the central PC or filestore has total control over the LAN, i.e. it operates as a master with the other stations being slaves. The slave stations are configured such that a part of the central filestore, normally a large capacity winchester disk drive, appears as an extra disk drive to the save, possible with a shared area by means of which files can be transferred between the slave PC's. Small filestore LAN's are aimed at single offices or very small organizations and usually offer only limited speeds and distances from the shared PC.

2.2.2 Wiring replacement LAN's

Wiring replacement LAN's typically offer a 9600 bps synchronous interface for up to 255 stations, with the LAN itself running at 500 Kbps or less. The main advantage of these LAN's, apart from offering potentially greater speeds, is the saving which results from having a single wire running round a building, rather than radial wiring from each terminal to the switch.

2.2.3 Personal computer networks

The personal computer LAN is essentially an extended version of the small filestore LAN's. In this case, however, there is a physical LAN medium with one or more shared file or disk servers, and usually with each PC's disk drives and printer being shared with others on the network. the networks vary in size from a few dozen to many hundreds of PC's. The major difference between these LAN's and the wiring replacement LAN's is that a board within the PC is used to provide the attachment to the LAN. the board will commonly handle all of the networking protocol required, thus relieving the CPU of the PC from this load.

2.2.4 General purpose LAN's

With general purpose LAN's, it is possible to allow heterogeneous equipment from different manufacturers to communicate with each other. These equipment can range from large mainframes and minis, through PC's to dumb terminals. It is these LAN's, of which the best known are the IEEE 802.3 CSMA/CD and the IBM Token Ring, which have received most attention from the standardization committees.

Among the common LAN solutions to PC's and workstations, Ethernet and IBM Token ring are the most popular for the general purpose LAN's.

2.3 Typical PABX system

A PABX is an assembly of equipment, usually telephones and modems, which allows an individual within a community of users to originate and answer calls to and from :-

1. other users of the system (extensions);
2. the public switched telephone network (PSTN);
3. a private automatic exchange (PAX);
4. other PABX's via tie lines, such as E&M; and
5. leased circuits, such as T1 (1.544Mbps) or CEPT (2.488 Mbps).

Fig. 2.1 shows a typical PABX system. The PABX is connected to PSTN through CO trunks and to other PABX via tie lines or leased circuits. Telephones are connected through the PABX to form complete speech paths. PC's communicate with other PC's either through modems or data modules. With modems, the digital data of the PC will be modulated into audio frequencies and be treated similar to normal voice calls. Dialing is made through the modem to the destination and once the circuit path is established, the two PC's exchange data as voice call does. With data module, the digital form of the PC's data will be preserved. Since most PABX's today are digital in the internal architecture, data modules are very effective in transmitting data, and the speed can be up to 64 kbps (i.e. the maximum allowable speed for a voice channel). However, with data module, the PC can only communicate with public packet switch network such as X.25, and it cannot go through the conventional PSTN because of the incompatibility of data formats.

2.3.1 PABX topology

Strictly speaking, a PABX can be regarded as the network manager of a LAN consisting of telephones. All the devices in a telephone system are connected, through twisted pairs of copper wires, to a central switch known as PABX, in a simple star topology. Since the PABX is vital to the functioning of the whole telephone communication network, redundancy is built in the PABX to avoid the possibility of network failure.

2.3.2 Circuit switching

In the telephone networks, circuit switching technique has been adopted. A call set-up procedure to find a circuit path between two telephones is necessary before they can talk to each other. Although packet switching is now available for most telephone companies for data communications, circuit switching is still used for the voice communications. Circuit switching is also used for modems and facsimiles since the data are transformed into audible signals and transmitted like a voice communication.

2.3.3 Telephony signalling

Telephone signaling is the language used by telephone systems to talk to each other to control the telephone services such as dialing, ringing and answer supervision. The signaling method is the way in which on-hook/off-hook signals are sent over a circuit. On-hook refers to the state that the handset is sitting on the hook-switch of the telephone, while off-hook refers to the stat that the handset is lifted up. The signaling between the central office (CO) of the telephone company and the attached terminal equipment, such as telephones and PABX's, is called subscriber loop signalling. The subscriber loop signals are sent by both DC current and by tones.

A telephone circuit is always in one of three states :

1. When a circuit is not in use and is available for use, it is said to be idle.

2. The time during which the call connections are being made is the call setup state.

3. When both ends of the communication path are connected together and communication is possible, the circuit is in the connected state.

2.3.3.1 Pulsing

Pulsing is the transmission of digits which uniquely identify the terminating end of a call. There are two common types of pulsing; dial pulse (DP) and dual tone multifrequency (DTMF).

Dial pulse operation

Dial pulsing (DP) is a method of transmitting digits by a series of momentary on-hooks. Dial pulsing is used on subscriber loops when a subscriber has a rotary dial phone.

The numerical value of each digit is represented by a series of momentary openings (pulses) of the loop. Each pulse cycle is made up of a break interval (B) and a make interval (M). Dial pulses are best described by their pulses per second (PPS) and % break. The digits themselves are separated by a relatively long make interval called the interdigit time.

DTMF operation

Dual tone multi-frequency (DTMF) pulsing sends a unique pair of frequencies within the voice band to represent digits. Each digit is represented by one frequency from each of two mutually exclusive frequency groups.

2.3.3.2 Subscriber loop signaling [2]

1. Idle

The CO supplied the following voltages :

Tip - ground potential

Ring - -48V

2. Incoming seizure

Ringing voltage of 75VAC, 20Hz is applied to the Ring lead.

3. Seizure response

A termination is placed across Tip and Ring leads when the extension answers the incoming call.

The CO removes ringing voltage as soon as the termination is sensed, and the calling party is connected to the termination.

4. Outgoing seizure

PABX places a termination across Tip and Ring. The CO senses the current flow, and response by supplying dial tone. The CO may signal that the call is answered by reversing the potentials on Tip and Ring. This is referred to as Answer Supervision.

5. Talking state

The termination remains in place, current continuous to flow.

6. CO release

The CO normally has no way to signal a release. The voltage connections for an idle circuit are identical to the connections to an active circuit.

7. PABX release

The termination placed across Tip and Ring leads is removed, causing an open circuit. The circuit returns to its idle condition.

8. CO release acknowledge

The CO normally cannot acknowledge release by PABX.

2.3.4 ISDN (Integrated Services Digital Network)

At present, virtually every subscriber local loop of the PSTN (public switched telephone network) is analog. This is acceptable for voice communications, but it is far too slow and unreliable for data communication. Data transmission now accounts for more than 10% of total network traffic and its use is growing rapidly. Moreover, since the digital PABX transforms the analog voice signals from the handset to digital PCM (pulse-code modulation) form, it should save the troubles and circuitries to transform PCM back to analog to send to the CO, if the CO could accept digital signal. Thus, ISDN (integrated services digital network) is intended to extend digital technology over the subscriber loop to the end-user terminal by using common telephone wiring and a standard interface plug. Ideally, the numerous diverse interfaces will be reduced (or eliminated) with a limited set of common conventions. In essence, ISDN is digital connectivity for the end user.

Digital technology is extending the voice communication service to other forms of information. Ultimately, the ISDN will provide this service for voice, data, text, graphics, and video.

ISDN has five major goals :

1. to provide a worldwide uniform digital network which supports a wide range of services and uses the same standards across different countries;

2. to provide a uniform set of standards for digital transmission across and between networks;

3. to provide a standard ISDN user interface, such that internal changes to a network are transparent to the end user;

4. in conjunction with the third objective, to provide for end-user application independence -- no consideration is made as to their characteristics in relation to the ISDN itself.

5. as an adjunct to goals three and four, to provide portability of user DTE's and applications.

A lot of efforts have been devoted to develop and implement the ISDN. Refs. [13] and [14] discuss the ISDN in greater depth. However, since the present analog PSTN has been well established and running successfully, it will take a long time to replace the present PSTN to a complete digital ISDN.

The digitization of the traditional applications based on analog transmission signals (voice, TV, etc.) is the necessary premise for their integration into one universal network. This is necessary for the transmission of data through the WAN as well as for the development of multi-media end-systems coping with digital information.

2.4 Applications of PABX

In business, many of the telephone conversations are to people within their own facilities. Without PABX, these calls will go a full circle, from their premises to the telephone company's central office (CO) and back again. Moreover, phones are normally used on average only 1/12th of the busiest hour (i.e. a scant five minutes out of each peak hour), pp. 20 of Ref. [3]. If the phones are only used 1/12th of the time during their peak calling period, then 12 people could conceivably share an outside line, dramatically reducing the business's telephone bill.

PABX is mainly used by medium to large organizations such as business and government offices, factories, hospitals, and hotels, to provide enhanced telephony services to the users such as call hold, call transfer, call forward, conference, out-going call restriction, centralized attendant for incoming calls, efficient utilization of out-going trunks. These functions are normally not provided by the telephone company but very useful for a large organization to help run the business more efficiently and cost-effectively.

PABX is used not only for the voice communications but also data communications between computers. However, since only circuit switching technique is employed in a PABX, only certain data communication functions are possible and the efficiency for such use is not optimized. At present, PABX's are only used about 10% of the time for data transmissions.

PABX's can be tied together to form a greater telephone network. In a multi-national company, PABX's in offices in different geographical locations can form a private network to provide more flexible telephone and data services to the users within the organization. With dedicated leased lines such as E&M, T1 or CEPT, the long distance phone bill could also be reduced, depending on the network configuration and the traffic.

2.5 Comparisons of LAN and PABX

There are two categories in office communications networks, a digital PBX with a star topology and a LAN with a bus or ring topology.

Like a LAN, a PABX is a unifying medium for a diverse array of machines. Unlike a LAN, however, the PABX serves as the central point of control for a network; i.e., all devices in the network are connected to and controlled by it. If there is a major failure in the system, the entire network can be affected.

Modern PABX's are very sophisticated computer systems that support a wide range of functions. Once used primarily to switch telephone lines, PABX's now support both voice and data traffic, allowing data terminal and telephone users to share a common network. Data communications is a relatively new feature on the PABX. Most newer systems use internal digital transmission and switching schemes that eliminate the need to translate data to analog signals and reconvert it to digital form. Digital PABX's with data communications capabilities do not require modems to communicate directly with attached equipment, although modems are required to link into external facilities since the PSTN is still mostly analog. How PABX's handle data communications varies from product to product. In general, a data module provides the interface between the switch and data equipment. This module either attaches to a telephone set, is standalone device similar in appearance to a modem, or is an integrated part of a telephone set or data terminal.

A PABX adopts circuit switching technique to obtain maximum efficiency for voice communications. It provides terminals with transparent paths suitable for continuous communications at up to 64 kbps. Due to its circuit switching characteristics, the transmission in a PABX is not broadcast in nature and the information transferred from the source to the destination will not be received by other users in the same network. Dedicated bandwidth of circuit path provides high throughput and low delays.

A LAN has been developed and introduced in an office mainly to accommodate computers and workstations. It adopts packet switching technique and is especially suitable for high speed and bursty data communications. Due to the statistical multiplexing of bandwidth, the transmission is shared and used efficiently, thus lowering the operating costs.

3. INTEGRATION OF PABX WITH LAN

For several years, industry analysts have debated the pros and cons of using a LAN or a PABX to effect networking in the automated office. It is now generally accepted that LAN's and PABX's each offer significant advantages depending upon their applications. The two technologies are presently being combined into an integrated PABX/LAN system, which is often referred to as a fourth-generation switch.

Many organizations will probably install both a PABX and a LAN; with the PABX used primarily for traffic with distant machines and the LAN used primarily for in-house traffic Gateways between the two will make life easier for users requiring to access both.

This section describes how the PABX's and LAN's are connected together and the problems encountered. And a new PABX system having a LAN architecture is then proposed and described.

3.1 Advantages of integration of PABX with LAN

In the companies, universities, research centers and hospitals, the communication requirements increase both internally and externally. At the same time, diverse means of communication offered to companies are growing rapidly, e.g. PABX, data switching, LAN, dedicated lines via cables and satellites, etc. In fact, each of these is dedicated to a traffic type, which makes it compulsory for the user to establish several networks with the corresponding connections the telephone on the one hand, and data on the other hand.

The integrated service LAN will reduce the costs when several dedicated networks are required today for transmitting voice, images and data; especially the :

- maintenance,
- establishment of dedicated networks,
- necessary bridges and gateways between those networks,
- communication servers, that need be gone through between the user and those networks,
- dedicated and incompatible supervisory stations.

Telecommunication services can be grouped into five classes as, according to Ref. [4] :

Class 1 (Kilo-stream) : services requiring circuit setting up for real-time applications, the bit rate of which is not over 64 Kbps, such as voice, facsimile and still-picture video.

Class 2 (Mega-stream) : accounts for larger bandwidth services but limited to a few Mbps, such as videoconferencing, videotelephony and high-quality image processing.

Class 3 (Kilo-bursty) : characterized by the exchange of short data messages with a low bit rate (1200 to 9600 bps), e.g., distributed processing, messaging services, text processing, videotex, exchanges between consoles and computers.

Class 4 (Mega-bursty) : encompasses long bulk messages requiring neither continuity nor immediate delivery. they can be transferred at the rate of a few Mbps, e.g. file transfers, access to data bases and high-speed laser printers.

Class 5 (Giga-stream) : covers the requirements in high-quality full motion video services calling for some 100 Mbps rates and real time.

Applications of PABX and LAN to different communications needs

	Bursty	Stream
Kilo	LAN	PABX
Mega	LAN	ATM/BISDN ?
Giga	ATM/BISDN ?	ATM/BISDN ?

With the present state-of-the-art technologies, PABX is more tailored for the kilo-stream of data applications, whereas LAN can be applied to the kilo-bursty and mega-bursty needs. However, ATM switch or Broadband ISDN is required for giga bit communications. With an integrated PABX/LAN system, it is intended to combine the functions of kilo-bursty, kilo-stream, and mega-bursty into the same data communication network.

The packet type service is unsatisfactory for real-time applications such as voice and high-quality motion picture video. With the circuit type service, the files cannot be transferred; the transmission is not good enough; besides, in the cases where the data rate varies, the transmission is not used to the best.

Although all ISDN services have not been defined, they are expected to include digitized voice, data transfer between computers, facsimile, graphics, video, and other services such telemetry (in energy management and security monitoring), videotex, electronic mail, and database access for word processor --- in all, a broad scope of information services in a information society.

3.1.1. LAN-PABX Gateway

A common method of connecting a PABX with a LAN is by a device called gateway. The gateway attaches physically to both the LAN and PABX in the normal manner of other LAN and PABX stations. Messages sent from one station to another are transmitted onto the LAN or through the PABX to the gateway, which receives them and retransmits them on its other side.

There are a number of ways to construct a gateway interfacing LAN and PABX. They can roughly be classified as :

1. MAC layer gateways
2. Link layer gateways
3. Network layer gateways
4. Session layer gateways

5. Application layer gateways

In the Ref. [5], these gateways are discussed in details and the merits and demerits of each type of gateways are compared. However, since the gateway is the only interface between the LAN and PABX, the traffic across the gateway may be tremendous and that would be the bottle-neck for the combined systems. Moreover, if the traffics in the two sub-systems (LAN and PABX) are un-balanced, the idle bandwidth of one sub-system cannot be used by the other heavy traffic sub-system. Thus the total system efficiency will not be optimized. Therefore, an integrated system capable of providing PABX and LAN functions is preferred.

3.1.2. Problems in interconnecting PABX and LAN [6]

1. Service category

In a PABX, switch path connection/disconnection corresponds to network layer connection.

In a LAN, such as CSMA/CD LAN, terminals generally support the connectionless data transfer service for the lower three layers.

It is important to decide how to match them.

2. Addressing method

PABX and LAN generally use different numbering scheme/addressing methods, which should remain independent from each other even at interconnection.

It is important to determine how a calling PBX terminal indicated a destination address in LAN to a gateway, and vice versa.

3. Flow control method

PBX terminals usually transmit and receive data at various speed up to 64 kbps, while LAN terminals transmit and receive data at much higher speed, e.g. 10 Mbps in the case of CSMA/CD LAN.

The transmission speed difference between PBX and LAN terminals causes a difference in effective throughput to them. Thus a flow control function is necessary to prevent a gateway from buffer overflow.

3.1.3. ISDN-PABX [7]

ISDN-equipped PABX's are available in support of ISDN PC, from various vendors such as Alcatel, Hitachi, NEC, Northern Telecom, and Siemens. The reason for such offering is the potential of the PRI's (primary rate access, 1.544 Mbps) which enable the PBX to access a wide variety of public and private network services and networks, such as call selection, automatic number identification, However, before ISDN is widely adopted and implemented world-wide ISDN-PABX is not ready to solve the immediate problem of integrating the functions of LAN and PABX.

3.2 Architecture of Integrated LAN and PABX

In order to combine the advantages of both LAN and PABX systems for voice and data communications, a distributed architecture is adopted. Physical stations are used as interface between the LAN transmission medium and various end-user devices such as telephone sets, workstations, PC's or dumb terminals.

From the PABX design point of view, it is not critical what kind of LAN architecture is employed so long as certain criteria are met. These criteria are necessary so that real time voice conversation and conference can be realized. However, in this project, an OR-channel LAN is assumed with the following properties : [8], [9]

1. the LAN is broadcasting in nature
2. the LAN is a double-ring architecture,
3. the medium access protocol is implemented as OR-type channel.

The PABX in this design is just a physical station in the LAN, and its functions are primarily to handle and manage the call set-up and trunk allocation. This PABX is not a LAN manager which oversees all network managements. The network management functions are not critical in this design and they can be performed by one or more stations.

3.3 Typical applications

The first practical application of an integrated PABX/LAN is to link the telephone and computer terminal present on a desk into a single communication instrument. Typical uses for this arrangement include sales order desks, technical support centers, travel agents and telemarketing applications.

A typical sales order desk could utilize an integrated PABX-LAN. Upon receipt of the phone call, the PABX would advise the database to recall the correct order form to the computer terminal screen of the agent answering the call. Different order forms could be recalled based on the trunk number the call came in on.

Similarly for a technical service center, an integrated desk could use the incoming call to correctly select the agent by product and recall the correct product database. In some countries, the telephone companies now provide Automatic Number Identification on incoming calls, so if this additional information is available, not only could the system recall the correct product on the computer, but since it would also know the identity of the caller, it could also recall that particular customer's file before the call is answered. The integrated system could maintain it by allowing a call transfer or conference of the voice and data from one desk to another if additional assistance is required.

In the travel industry, an integrated environment could also add a new level of personalized service to the industry. Since the system could tell the agent who is calling and provide this customer's file on the screen, the agent can then greet the caller by name, ask about their last trip as shown in the file and immediately assist the caller with arranging their next trip. This highly personalized service will guarantee repeat business.

A number of physical stations are distributed along the LAN transmission medium. End-user devices such as PC, workstations and telephones are interfaced to these physical stations, through which end-user devices communicate with other end-user devices in the network. One of these physical stations is dedicated to function as a PABX (private automatic branch exchange). This PABX station includes trunk interfaces to the central office (CO) of the telephone company.

Details of the physical stations other than the PABX are discussed in another report by W. Y. Siu [10].

There are three main sub-systems in the PABX station, i.e. the control section, the trunk interface section and the network interface section. The design of the network interface section of the PABX station is the same as that interface section of other physical station. At this design stage, this section is assumed to be that of Refs. [9] and [11]. This network interface may be changed to another design during the course of the actual prototype implementation.

The trunk interface section includes electronic interface to the CO and the associated circuitries for digitizing the analog voice signal into PCM format and vice versa.

The control section mainly consists of a microcomputer system controlling the proper functioning of the trunk interface and network interface.

4. CALL PROCESSING

Call processing is the core of PABX activities and deals basically with call set-up and call release. In a circuit switching environment, the call set-up is the process to determine and allocate a free circuit path for the call. In a virtual-circuit packet-switching situation, a virtual circuit path has to be assigned for the call. In the design of the present PABX with an OR-channel LAN architecture, a logical channel number has to be assigned. In the OR-channel architecture, the allocation of a logical channel does not guarantee the successful transmission of the voice packet because of the possible collision with other packets. However, since real-time voice packet will be given the highest priority and no two real-time voice calls will be assigned the same logical channel by the call set-up procedures of the PABX, successful transmission of real-time conversation can be assured.

After the call is completed, call release procedure is performed to return the channel resources to idle so that they can be assigned to new calls.

A telephone call is originated as follows :

1. An individual at the source lifts the handset off the cradle to start the call. This is called going off-hook and is detected at the local telephone office.

2. The local telephone office returns a dial tone signal heard by the caller on the source phone.

3. Rotary dialling or push-button depressed by the caller provides instructions to the switch phone.

4. A connection is established through the switch to the receiver and ringing initiated. A ring-back signal is heard at the source telephone.

5. The receiver handset is lifted off the cradle and conversation may commence.

For the termination of a call, the following procedures are taken :

1. When the conversation is finished, either end will hang-up.

2. The on-hook signal will be detected by the CO, and the circuit path is removed. At the same time, a busy tone is sent to the other party which is still off-hook.

3. When the other party hears the busy tone, he hangs up the handset. The CO then removes the circuit and the call is completed.

4.1 Finite State Diagrams for voice calls

In a telecommunication system, each device will go through a number of states to carry out a call service. The states are usually represented as a finite state diagram.

In the state diagram, each node, i.e. state, has a name,. the state name, and each arc may have the name/names of the signal/s whose reception causes the activation of the transition. The sate diagrams represent the possible steps that a device will go through during a call.

Figs. 4.1 and 4.2 show the state diagrams for the telephone sets and the trunks for simple telephone conversations and conference.

State diagram for telephones

In Fig. 4.1, the Phone_idle is the state that the telephone is idle, i.e. on-hook state. This is the state of the phone for more than 90% of the time.

When one lifts up the handset, an off-hook signal is received by the phone and it goes to the state Off_hook_in_service. In this state, the phone will send out a dial tone through the speaker of the handset. Once the user dials a digit, it will go into the state Await_digit. When all the digits have been collected, they will be sent to the PABX or the CO. If the destination is ready and idle, ringing signal will be sent to the destination telephone and at the same time a ringback tone will be heard by the caller. If the other side goes off-hook, the circuit path is completed and ringing tone and ringback tone will be removed, and conversation can take place.

State diagram for trunks

The initial state of the trunk is Trunk_idle. When it receives signals from the PABX asking for outgoing trunk, it seizes the CO circuitry by forming a terminating loop in the trunk circuit. When the CO responses, it sends the digits to the CO. The CO then checks the validity of the dilled digits and availability of the destination. If the destination is idle, a ringback tone will be heard. When the other party picks up the phone, both will go into the Talking state.

For an incoming call, a ringing voltage will be sensed by the trunk circuit. The PABX will then search for an extension in the system to answer the call. Usually, the attendant (or the operator) phone will be the answering point. When the attendant picks up the phone the circuit path is completed and both enter the Talking state.

Usually, the attendant will transfer this call to other extension in the system. The state diagram for voice conference details the flow of a conference and thus transfer of a call.

4.2 SDL representations of voice calls

Based on the state diagrams for the call processing, much detailed hand-shaking protocols are developed. These hand-shaking protocols are represented in the Specification Description Language (SDL) forms. SDL is based on a set of constructs or graphic symbols to describe the behavior of a process that can be modelled as a finite state machine. Details of the SDL are described in Appendix A.

SDL of phones

The SDL representation of calls is shown in Fig. 4.7. The starting of the phones are in the Phone_idle state. While waiting for inputs to transit into another state, the telephone interface will send an Ext_idle message to the PABX so that the PABX will update the status of the phone to be idle, thus allowing other phones to make call to it. If an Off_hook message is received from the phone, the interface will put it into the Off_hook_in_service state; sending an Ext_off_hook message to the PABX and outputting Dial_tone to the handset. When a Dial_digit is received from the telephone, the physical station will cancel the dial tone and store the digit in the memory, and then wait for the next digit, if any. When no more digit is received before Time_out, it will outpulse these digits to the PABX. When B_idle and Call_ID messages are received from the PABX, a Ringback_tone is sent to the handset. When the other party lifts up the handset, a B_off_hook message is sent to the originating phone. Upon receiving this message, the ringback tone to the handset will be canceled and the Call_ID for this call will be sent to the called party. Both telephones then communicate with reference to this call identification until the conversation is over. A simplified call process is shown in Fig. 4.4 for internal calls.

SDL for trunks

Fig. 4.9 shows the detailed handshake protocol for the trunks with the PABX. Fig. 4.5 shows a much simplified call process for an outgoing call.

SDL for conference

Fig. 4.10 shows the SDL representation of the protocol for conference with two internal telephones

4.3 Software implementations of SDL diagrams

With the SDL diagrams, it is possible to implement the protocol in softwares.

4.3.1 PABX operating system

In the PABX, there are two important tables to contain the information of system resources so that the PABX can assign free available resources to the calls when required. These tables are : Call_ID_Table and the Device Activity Map (DAM). The Call_ID_Table lists all the call identifications in the system, and which parties are assigned to the call_ID. The structures of the DAM and Call_ID_Table are shown in Figs. 4.11 and 4.12.

```
+-------------+-----------+-----------+
| Device ID   |  Status   |  Call ID  |
+-------------+-----------+-----------+
|      1      |   IDLE    |           |
+-------------+-----------+-----------+
|      2      |   BUSY    | call ID 1 |
+-------------+-----------+-----------+
|      3      |   RING    |           |
+-------------+-----------+-----------+
|                                     |
|                                     |
/                                     /
/                                     /
|                                     |
+-------------+-----------+-----------+
| MAX_DEV - 1 |   IDLE    |           |
+-------------+-----------+-----------+
|   MAX_DEV   |   BUSY    | call ID N |
+-------------+-----------+-----------+
```

Fig. 4.11 Structure of Device Activity Map (DAM)

The Device Activity Map lists the status and the call ID associated with the call of each device, including telephones and trunks. This table will be updated when messages are received from the physical station. In order to reduce the communications between the physical stations and the PABX, only three states, IDLE, BUSY and RING, are recorded in the DAM.

```
+-----------+-----------+-----------+-----------+
|  Call ID  |  Party 1  |  Party 2  |  Party 3  |
+-----------+-----------+-----------+-----------+
|     1     |           |           |           |
+-----------+-----------+-----------+-----------+
|     2     |           |           |           |
+-----------+-----------+-----------+-----------+
|     3     |           |           |           |
+-----------+-----------+-----------+-----------+
|                                               |
|                                               |
/                                               /
/                                               /
|                                               |
+-----------+-----------+-----------+-----------+
| MAX_CID -1|           |           |           |
+-----------+-----------+-----------+-----------+
|  MAX_CID  |           |           |           |
+-----------+-----------+-----------+-----------+
```

Fig. 4.12 Call_ID_Table structure

If a call ID is assigned to one or more devices, telephones or trunks, the device ID will be entered into the Party fields. If all the call ID's are used up by different devices, a congestion is said to occur and busy tone or congestion tone will be returned to the caller. If a call is completed, the Party fields will be cleared and thus the call ID can be available for other new call. MAX_CID is the maximum number of call ID's in the system. That is, only MAX_CID of voice calls can take place simultaneously. This may not be the maximum channels available in the system, since some channels may be reserved for data and control messages.

The pseudo-codes for the PABX operating system are listed in section 4.4.

4.3.2 Trunk operating system

In the physical station interfacing the trunk circuits, a Trunk State Map (TSM) is maintained. TSM is similar to a DAM, except that only the status of the trunks are recorded.

The pseudo-codes for the trunk operating system are listed in section 4.5.

4.3.3 Message format

In the PABX/LAN system, control messages and data messages have to be sent from one physical station around the transmission medium and received by the intended receivers. The message format consists of the following fields :

```
+--------------+
|    Action    |
+--------------+
|    Source    |       Sending physical station
+--------------+
|  Destination |       Receiving physical station
+--------------+
| Calling_party|  Calling device id (trunk or telephone)
+--------------+
| Called_party |       Called device id
+--------------+
```

Fig. 4.13 Message format for control packet

```
+---------------+
|   Call_ID     |
+---------------+
| Source_party  |
+---------------+
| Voice packet  |
+---------------+
|               |
/               /
/               /
|               |
+---------------+
| Voice packet  |
+---------------+
```

Fig. 4.14 Message format voice packet

These message formats are used for the network layer. In the medium access layer, these formats may be altered or appended for the actual transmission in the physical layer.

4.4 Pseudo codes for PABX

```
typedef int DAM_state

enum DAM_state {
    BUSY = 1,
    IDLE = 2,
    RING = 3,
}

typedef int action;

enum action {
    B_idle              = 1,
    B_off_hook          = 2,
    B_on_hook           = 3,
    Call_connected      = 4,
    Ext_busy            = 5,
    Ext_idle            = 6,
    Ext_off_hook        = 7,
    Incoming_call       = 8,
    Invalid_number      = 9,
    Line_busy           = 10,
    Outgoing_number     = 11,
    Outpulse_number     = 12,
    Release_call_ID     = 13,
    Ringing             = 14,
    Trunk_idle          = 15,
    Trunk_ring          = 16,
    Trunk_seized        = 17,
    Voice_packet        = 18,
};

typedef int device_status;

enum device_status {
    B_idle              = 1,
    B_off_hook          = 2,
    B_on_hook           = 3,
    Call_answered       = 4,
    Call_connected      = 5,
    CO_on_hook          = 6,
    Ext_busy            = 7,
    Ext_idle            = 8,
    Idle                = 9,
    Invalid_number      = 10,
    Line_busy           = 11,
    Outgoing_call       = 12,
    Trunk_busy          = 13,
    Trunk_seized        = 14,
}


#define PABX ps_id_pabx
        /* the ps_id of the PABX station                         */
```

```
#define ATTENDANT extension_attendant
       /* the extension number of attendant                */

#define TRUNK_PS ps_id_trunk
       /* the ps_id of the trunk interface                 */


struct message {
/*   message format to transmit through the LAN medium */
/*   to the physical stations, including station PABX  */

     action     act;
     ps_id      source;/* ps_id is the identification */
     ps_id      dest;  /* of physical station         */
     extension  calling_p;
     extension  called_p;
}

PABX_OS()
{
     struct     message    msg;
     device_status         status, trunk_resp;
     call_identification   call_id;

     /* System initializations                    */
     DAM_Initialization();
     In_Buf_Initialization();
     Out_Buf_Initialization();
     Call_ID_Table_Initialization();

     do {
        msg = Get_Msg_From_In_Buf();
        switch (msg.act) {
        case 'Ext_idle' :
           Set_DAM(msg.calling_p,IDLE);
           break;

        case 'Ext_off_hook' :
           Set_DAM(msg.calling_p,BUSY);
           break;

        case 'Trunk_idle' :
           Set_DAM(msg.calling_p,IDLE);
           break;

        case 'Trunk_ring' :
           Set_DAM(msg.calling_p,RING);
           break;

        case 'Trunk_seized' :
           Set_DAM(msg.calling_p,BUSY);
           break;

        case 'Release_call_ID' :
           Release_Call_ID(msg.calling_p);
           break;
```

```
case 'Outpulse_number' :
   status = Check_Status(msg.called_p);
   switch (status) {
   case 'Invalid_number' :
      Output(Number_invalid,PABX,msg.source,
             msg.called_p,msg.calling_p);
      break;

   case 'Line_busy' :
      Output(Line_busy,PABX,msg.source,
             msg.called_p,msg.calling_p);
      break;

   case 'B_idle' :
      Set_DAM(msg.called_p,BUSY);
      call_id = Assign_Call_ID(msg.calling_p);
      if (call_id == 0) {
         Output(Line_busy,PABX,msg.source,
                msg.called_p,msg.calling_p);
         break;}
      else {
         Output(call_id,PABX,msg.source,
                msg.called.p,msg.calling_p);
         Output(Ringing,PABX,msg.source,
                msg.calling_p,msg.called_p);
         break;
      }

   case 'Outgoing_call' :
      trunk_no = Search_Idle_Trunk();
      if (trunk_no == 0) status = Trunk_busy;
      else {
         Output(Outgoing_call,PABX,TRUNK_PS,
                msg.calling_p,trunk_no);
         status = Get_Trunk_Call_Status(trunk_no);
      }
      while (status == Idle)
         status = Get_Trunk_Call_Status(trunk_no);

      switch (status) {
      case 'Trunk_busy' :
         Output(Line_busy,PABX,msg.source,
                msg.called_p,msg.calling_p);
         break;

      case 'Trunk_seized' :
         Send_Out_No(outgoing_number,trunk_no);
         trunk_resp =
            Set_Trunk_Call_Status(trunk_no);

         switch (trunk_resp) {
         case 'Trunk_busy' :
            Output(Line_busy,PABX,msg.source,
                   msg.called_p,msg.calling_p);
            break;

         case 'Call_connected' :
            call_id =Assign_Call_ID(msg.calling_p);
```

```
            if (call_id == 0) {
               Output(Line_busy,msg.called_p,
                      msg.calling_p);
               break;
            }
            else {
               Output(B_idle,PABX,msg.source,
                      msg.called_p,msg.calling_p);
               Output(call_id,PABX,msg.source,
                      msg.called_p,msg.calling_p);
               break;
            }

         default :
            Error_Handle();
            break;
         }

      default :
         Error_Handle();
         break;
      }

   case 'Incoming_call' :
      status = Check_Status(ATTENDANT);

      switch (status) {
      case 'Ext_busy' :
         Output(Ext_busy,PABX,msg.source,
                msg.called_p,msg.calling_p);
         break;

      case 'Idle' :
         call_id = Assign_Call_ID(msg.calling_p);
         if (call_id == 0) {
            Output(Ext_busy,PABX,TRUNK_PS,
                   msg.called_p,msg.calling_p);
            break;
         }
         else {
            Output(Ext_idle,PABX,TRUNK_PS,
                   msg.called_p,msg.calling_p);
            Output(call_id,PABX,TRUNK_PS,
                   msg.called_p,msg.calling_p);
            Output(Ringing,PABX,msg.source,
                   msg.calling_p,ATTENDANT);
            break;
         }

      default :
         Error_Handle();
         break;
      }

   default :
      Error_Handle();
      break;
   }
```

```
        default :
           Error_Handle();
           break;
        }
     } while (true);
}

int Assign_Call_ID(PARTY)
{
/*Assign a free call ID for the PARTY to proceed a call*/
/*  if no free call ID is available, zero is returned  */

     for (i=0; i<MAX_CID; i++) {
        if ((CID[i].party_1 == NULL) && (CID[i].party_2
              == NULL) && (CID[i].party_3 == NULL)) {
           CID[i].party_1 = PARTY;
           return(CID[i].call_id);
        }
     return(0);
}

void Call_ID_Table_Initialization()
{
/* Initialize the call identification table,        */
/* CID[].  MAX_CID is the maximum number of call */
/* ID in the system call_id[] is the array of       */
/* call ID's available CID[] is the array of        */
/* Call ID Table,  Each entry of Call ID Table      */
/* consists of four fields;   call_id is used to */
/* identify each call, party_1, party_2 and         */
/* party_3 contain the different parties involved*/
/* in the call.  A maximum of three parties are     */
/* allowed for each call                            */

     for (i=0; i<MAX_CID; i++) {
        CID[i].call_id = call_id[i];
        CID[i].party_1 = NULL;
        CID[i].party_2 = NULL;
        CID[i].party_3 = NULL;
     }
}

device_status Check_Status(NUMBER)
{
/* Check whether NUMBER corresponds to valid     */
/* extension number, extension idle, extension */
/* busy, or an outgoing number; and return the */
/* result;                                       */

     if (Out_No(NUMBER) == TRUE) return(Outgoing_call);
     else for (i=0; i<MAX_DEV; i++)
        if (DAM[i].dev_id == NUMBER)
           return(DAM[i].status);
     return(Invalid_number);
}
```

```
void DAM_Initialization()
{
/* Initialize the Device Activity Map, DAM      */
/* MAX_DEV is the maximum number of devices of */
/* the system DAM[] is the array to record     */
/* the activity of each device                 */
/* Each entry of DAM[] consists of three fields*/
/* DAM[].dev_id contains the extension number  */
/* or trunk number for which the activity is   */
/* recorded.  DAM[].status records the status  */
/* of the device   DAM[].call_id contains the  */
/* call_id, if any, associated with the device.*/

     for (i=0; i<MAX_DEV; i++) {
        DAM[i].dev_id = device[i];
        DAM[i].status = IDLE;
        DAM[i].call_id = NULL;
     }
}

device_status Get_Trunk_Call_Status(TRUNK)
{
/* Get from the trunk interface hardware the status */
/* of the trunk TRUNK, and return the status         */
}

boolean Out_No(NUMBER)
{
/* Check whether NUMBER is a valid telephone number */
/* to the CO, return TRUE or FALSE                   */
}

void Output(ACT,SOURCE,DEST,PARTY_1,PARTY_2)
{
/* Send message via communication medium from end-user */
/* PARTY_1 of physical station SOURCE to end-user      */
/* PARTY_2 of physical station DEST to take ACT;        */
}

int Release_Call_ID(PARTY)
{
/* Release call identification associated with PARTY,  */
/* so that it can be allocated to other calls;          */

     for (i=0; i<MAX_DEV; i++)
        if (DAM[i].dev_id == PARTY) {
           if (Reset_CID(DAM[i].call_id) == 1) exit(1);
           else {
              DAM[i].call_id = NULL;
              exit(0)
           }
        }
     exit(1);
}
```

```
int Reset_CID(CALL_ID)
{
/* Reset all parties associated with CALL_ID to NULL  */

     for (i=0; i<MAX_CID; i++)
        if (CID[i].call_id == CALL_ID) {
           CID[i].party_1 = NULL;
           CID[i].party_2 = NULL;
           CID[i].party_3 = NULL;
           exit(0);
        }
     exit(1);
}

int Search_Idle_Trunk()
{
/* Look through the DAM to find an idle trunk, and  */
/* return the trunk number;                          */
/* return zero if no free trunk is available         */

     for (i=0; i<MAX_TRUNK; i++)
        for (j=0; j<MAX_DEV; j++)
           if ((DAM[j].dev_id == trunk[i]) &&
                (DAM[j].status == IDLE))
              return(trunk[i]);
     return(0);
}

void Send_Out_No(NUMBER,TRUNK)
{
/* Send out the outgoing telephone number NUMBER    */
/* to the trunk interface for trunk TRUNK           */

     Output(Outgoing_number,NUMBER,PABX,TRUNK);
}

int Set_DAM(PARTY,STATE)
{
/* Set Device Activity Map for PARTY to STATE;      */

     for (i=0; i<MAX_DEV; i++)
        if (DAM[i].dev_id == PARTY) {
           DAM[i].status = STATE;
           if (STATE == IDLE) DAM[i].call_id = NULL;
           exit(0);
        }
     exit(1);
}
```

4.4 Pseudo codes for trunks

```
Trunk_OS()
{
     struct    message   msg;

     /* System initializations  */
     In_Buf_Initialization();
     Out_Buf_Initialization();
     Init_Trunk_Status();

     do {
        msg = Get_Msg_From_In_Buf();
        switch (msg.act) {
        case 'Outgoing_call' :
           trunk_no = msg.called_p;
           status = Seize_Trunk(trunk_no);

           switch (status) {
           case 'Line_busy' :
              Output(Trunk_busy,msg.dest,msg.source,
                     trunk_no,msg.calling_p);
              Trunk_Idle(trunk_no);
              break;

           case 'Trunk_seized' :
              Output(Trunk_seized,msg.dest,msg.source,
                     trunk_no,msg.calling_p);
              outgo_no = Get_Out_No(msg.calling_p);
              response = Outpulse(outgo_no,trunk_no);

              switch (response) {
              case 'Call_connected' :
                 Output(Call_connected,msg.dest,msg.source,
                        trunk_no,msg.calling_p);
                 call_status = Await_answer;
                 while (call_status == Await_answer)
                    call_status =Get_Call_Status(trunk_no);

                 switch (call_status) {
                 case 'Call_answered' :
                    Output(B_off_hook,msg.dest,msg.source,
                           trunk_no,msg.calling_p);
                    call_id = Get_Call_ID(trunk_no);
                    Talking(call_id,trunk_no,msg.calling_p);
                    break;

                 case 'B_on_hook' :
                    Trunk_Idle(trunk_no);
                    break;

                 default :
                    Error_Handle();
                    break;
                 }
```

```
        case 'Line_busy' :
          Output(Line_busy,msg.dest,msg.source,
                trunk_no,msg.calling_p);
          Release_Trunk(trunk_no);
          Trunk_Idle(trunk_no);
          break;

        default :
          Error_Handle();
          break;
        }

      default :
        Error_Handle();
        break;
      }

    default :
      Error_Handle();
      break;
    }

    if (trunk_status == Ringing) {
      trunk_no = Get_Ring_Trunk();
      Output(Trunk_ring,msg.dest,msg.source,
            trunk_no,PABX);
      Output(Incoming_call,msg.dest,msg.source,
            trunk_no,PABX);
      status = Get_Call_Status(trunk_no);

      switch (status) {
      case 'Ext_busy' :
        Call_Hang_Up(trunk_no);
        break;

      case 'Ext_idle' :
        call_id = Get_Call_ID(trunk_no);
        TSM[trunk_no].call_id = call_id;
        call_status = Seize_Trunk(trunk_no);
        while (call_status == Await_answer)
             call_status =Get_Call_Status(trunk_no);

        switch (call_status) {
        case 'B_off_hook' :
          Output(call_id,msg.dest,msg.source,
                trunk_no,msg.calling_p);
          Talking(call_id,trunk_no,msg.calling_p);
          break;

        case 'CO_on_hook' :
          Output(B_on_hook,msg.dest,msg.source,
                trunk_no,msg.calling_p);
          Release_Trunk(trunk_no);
          Trunk_Idle(trunk_no);
          break;

        default :
          Error_Handle();
```

```
          }
       default :
          Error_Handle();
          break;
       }
    } while (true);
}

ATD(TRUNK)
{
     Convert analog audio signal from the audio interface
     of TRUNK and convert it to PCM voice packets, and
     return the packets;
}

void Call_Hang_Up(TRUNK)
{
/* This is the state that the call has been hanged-up */
/* by the extension, while the CO is off-hook  The    */
/* call will be over completely when TRUNK is released*/

     Release_Trunk(trunk_no);
     Trunk_Idle(trunk_no);
}

DTA(PACKET,TRUNK)
{
     Convert PCM voice packets PACKET to analog audio
     signals and send it to the audio interface of TRUNK;
}

int Get_Call_ID(TRUNK)
{
/* Get and return the call ID assigned to TRUNK       */
/* for the call                                       */

     msg = Get_Msg_From_In_Buf();
     if ((msg.act == call_id) && (msg.dest == TRUNK))
        return(call_id);
}

device_status Get_Call_Status(TRUNK)
{
/* Get the call status for the TRUNK                  */
/* The calling party is also monitored continuously */

     call_status = Await_answer;
     if (TSM[TRUNK].status == CO_off_hook)
        call_status = Call_answered;
     if (TSM[TRUNK].status == CO_on_hook)
        call_status = CO_on_hook;
     msg = Get_Msg_From_In_Buf();
     if (msg.called == TRUNK) {
        if (msg.act == B_off_hook)call_status=B_off_hook;
        if (msg.act == B_on_hook)call_status = B_on_hook;
     }
     return(call_status);
}
```

```
int Get_Out_No(PARTY)
{
/* Get the outgoing telephone number from          */
/* extension PARTY                                 */

     msg = Get_Msg_From_In_Buf();
     if ((msg.source == PARTY) &&
         (msg.act == Outgoing_number))
     return(msg.number);
}

int Get_Ring_Trunk()
{
/*Search for the ringing trunk and return trunk number;*/

     Search from hardware the ringing trunk;
     TSM[TRUNK].status = RING;
     return(TRUNK);
}

device_status Get_Trunk_Status(TRUNK)
{
/* Return the status of TRUNK from hardware;       */

     Search trunk status from hardware;
     TSM[TRUNK] = state;
}

void Init_Trunk_Status()
{
/* Initialize the Trunk Status Map, TSM            */

     for (i=0; i<MAX_TRUNK; i++) {
        TSM[i].dev_id = trunk[i];
        TSM[i].status = IDLE;
        TSM[i].call_id = NULL;
     }
}

device_status Outpulse(NUMBER,TRUNK)
{
/* Outpulse the telephone number NUMBER to the CO    */
/* through trunk TRUNK, and return the result status */
}

void Release_Trunk(TRUNK)
{
/* Release the trunk TRUNK, i.e., equivalent to    */
/* on-hook of a telephone; and reset the TSM       */

     Release the trunk;
     TSM[TRUNK].status = IDLE;
     TSM[TRUNK].call_id = NULL;
}
```

```
device_status Seize_Trunk(TRUNK)
{
/* Seize the TRUNK and return the result;                   */

     Seize the TRUNK;
     status = Get_Trunk_Status(TRUNK);
     return(status);
}

Talking(call_id,calling,called)
{
     call_status = Talking;

     do {
        msg = Get_Msg_From_In_Buf();

        switch (msg.act) {
        case 'Voice_packet' :
           DTA(voice_p,calling);
           break;

        case 'B_on_hook' :

     Output(Release_call_id,TRUNK_PS,PABX,calling,PABX);
           call_status = On_hook;
           Call_Hang_Up(calling);
           break;
        }

        voice_p = ATD(calling);
        Voice_Pack(Voice_packet,call_id,voice_p,calling);
        if (Get_Trunk_Status(calling) == On_hook) {
           call_status = On_hook;

     Output(B_on_hook,TRUNK_PS,called,calling,called);
           Release_Trunk(calling);
           Trunk_Idle(calling);
        }

     } while (call_status == Talking);
}

void Trunk_Idle(TRUNK)
{
/* Send idle signal to PABX to reset the DAM of TRUNK */

     Output(Trunk_idle,TRUNK_PS,PABX,TRUNK,PABX);
}
```

5. HARDWARE IMPLEMENTATION

Fig. 5.1 shows the overall architecture of the integrated PABX/LAN. The LAN transmission medium is assumed to be a general structure with certain basic minimum requirements for the real-time voice communication. In this report, only the PABX station is discussed. The main functions of the PABX in this network are :

1. Call setup for the voice communications, within the system and with outside PSTN.

2. Allocation of call identification for all the voice calls.

Inside the PABX station, two functions are performed :

1. Call setup and call ID allocation for the whole system.

2. Interfacing with the CO for the trunk circuits.

Although these two functions can be performed in two different physical stations, they are combined into one PABX station in this discussion because there will be a substantial amount of communication between this two sections. If these functions are performed in two different physical stations, the transmission medium will be flooded with messages between this two physical stations and thus the system efficiency will be much reduced.

The network interface is not done in this design, and the design by W.D. Livingston, Ref. [11], is assumed for the present work.

5.1 TRUNK INTERFACE

The central office (CO) of the telephone company supplies the TIP and RING terminals of the trunk with 0V and -48V, respectively in series with a 450 ohm coil winding. The two windings are both wound on the same armature which is magnetically coupled to a normally open switch. The switch is adjusted to close when the magnetic field produced by 17mA of current flowing through each winding is present. The magnetic fields of both windings are additive, so it is also possible to close the switch by allowing 34mA of current to flow through only one winding.

5.1.1 PABX to CO call

The PABX puts a loading across the Tip and Ring leads, thus allowing a current of sufficient magnitude to flow through each of the windings. This notifies the CO that the trunk wants to make a call and the CO responds by delivering the dial-tone signal. The PABX can now send DTMF digits or rotary digits to the CO.

The CO deciphers the digits and make the connection to the other party.

5.1.2 CO to PABX call

The CO initiates a call by connecting the 75V ringing generator (75V rms @ 20 Hz referenced to -48V dc) to the Ring lead. A detector on the trunk circuit recognizes the ringing signal and notifies the PABX. When the PABX is notified of the presence of ringing it will ring an attendant's phone. When the attendant answers, the PABX will connect the load across the Tip and Ring leads. The CO responds this with the removal of ringing signal.

A more detailed specifications of the signaling formats used by the HK PSTN are listed in Appendix B.

5.2 Subscriber Interface Circuit

Fig. 5.2 shows the block diagrams for the subscriber interface circuit. This circuit is used to interface the CO to the microcomputer bus of the PABX system. The analog signal of speeches are converted to and from the PCM format processed by the PABX computer system.

The interface to CO trunks is done by the PSTN Trunk Interface, which senses the signalling of the CO and output to the control circuitry of the PABX. This signalling information is then multiplexed and sent to the CPU through the microcomputer bus. The codec is used to convert the analog voice signal into PCM format, and vice versa. With serial/parallel converter, the serial

PCM signal is converted to an 8-bit parallel format so that one voice sample can transmitted to the microcomputer bus at a time. Similarly, parallel/serial converter is used to convert the 8-bit data into a serial PCM format.

5.4 PSTN Trunk Interface

The PSTN trunk interface circuit is used to interface between the CO and the control section of the PABX station.

The Tip and Ring leads are connected directly to the two terminals provided by the telephone company. VR and VX are the audio signal paths for the outgoing and incoming directions. A seize (SZ) signal is used to signal the CO for the trunk seizure; this is equivalent to off-hook of telephone set. Other signals are used to notify the PABX about the trunk status such as ringing, trunk seized and trunk polarity.

Ring detector is used to detect ringing voltage from the CO, i.e., an incoming call.

Polarity detector is used to detect whether the voltage across the Tip and Ring leads has been reversed. This is important especially if the CO provide line reversal as a supervisory signal.

Details of the ring detector and polarity detector circuits are shown in Fig. 5.4.

Since the other parts of the trunk interface require a proper polarity of the tip and ring leads, a bridge rectifier is used to convert the tip and ring voltage to the right polarity.

Active termination, as shown in Fig. 5.5, is used to provide proper loop resistance for the CO so that the current flowing can be sensed by the CO for outgoing call. A seize signal (SZ) is used to connect the active termination to the CO for trunk seizure. The active termination circuitry is also used to simulate the on/off hook of the hook-switch, thus providing pulse for outpulsing dialed digits.

The transformer is used to isolate the audio circuit from the CO for safety purposes.

The audio circuit is the hybrid used to couple/decouple the transmit and receive speech signals to the CO. An analog echo cancellation circuit is used to reduce the signal level feeding into the receive path from the transmit path. Fig. 5.6 shows the audio circuit for the PSTN trunk interface.

6. CONCLUSIONS

In this report, only the basics of building an integrated system was discussed. In a commercial PABX, there are a lot of functions and features provided for the users. A few of these features are listed below :

- automatic route selection
- call forward
- call back
- call hold
- call pickup
- camp on
- speed call

These features can be implemented based on the basic call set-up processes described in Chapter 4.

Moreover, the medium access (MAC) protocol for this proposed integrated PABX/LAN is still being studied and developed by C. C. Yuan [18].

The design work reported here is based on a few assumptions of the LAN and it does not rely too much on the detail MAC protocol. The call setup and release protocols will remain the same with different MAC protocols. Nevertheless, the lower layer interface to the MAC and the network interface may have to be modified to be compatible with the MAC protocol.

The project to complelely build an integrated system of PABX/LAN surely requires a lot of effort and just a few man-years can only furnish a preliminary portion of the whole project. More efforts have to be devoted to the MAC protocol, network interface hardware, software implementation of the pseudo-codes listed in Chapter 4.

ACKNOWLEDGEMENTS

I would like to take this opportunity to thank Prof. Robert S. Y. Li for his many valuable advices to this project. Many thanks are also given to W.Y. Siu, C. C. Yuan and Alex Cheung, without the many open discussions with them, this project of integrated PABX/LAN will never reach this stage.

APPENDIX A

CCITT SPECIFICATION AND DESCRIPTION LANGUAGE [15]

The procedures by which a pair of communicating telecommunications processes interact can be described in graphic form by a language that has been developed over a number of years by the CCITT. This language, which is known as the Specification and Description Language (SDL), is used extensively in the specification of the ISDN protocols.

SDL is based on a set of constructs or graphic symbols to describe the behavior of a process that can be modelled as a finite state machine. The most important of these symbols for representing telecommunications process are shown in Fig. A.1.

The process exists at any one time in one of a finite number of states. When it receives an input signal it undergoes a transition to either the same state or to another state. during this transition it produces an output signal. Input signals may be received from more and output signals may be delivered to more than one user of the process. State transitions are sometimes governed by variables such as counters or timers and are subject to the outcome of a test on the value of one or more such variables. While in a particular state, the process can undergo a procedure such as the updating of a variable. It may also invoke a call to another entity for carrying out a procedure. The latter may itself be a finite state machine represented by an SDL diagram.

SDL is a standard for specifying telecommunication systems and data communication protocols. The virtues of SDL are its ability to describe and modularize the interactions and interfaces between any processes working concurrently with each other. SDL processes are concurrent extended finite state machines that model the dynamic behavior of an SDL system. Processes communicate with each other only by means of discrete, asynchronous messages, called signals. a process will wait in a state until it receives a valid signal from its environment. Then it will perform a transition to another state. During the transition it may perform actions such as manipulation of local data or sending signals to other processes or to the external environment of the system.

The state of the process

Input received from user .

Output or response to user

Input received from PABX

Output or response to PABX

Procedure which may, or may not be described in details in another SDL diagram

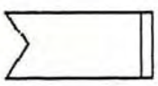
A decision made during the operation of the process

Fig. A.1 SDL graphic symbols

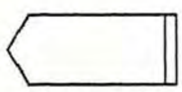
The state of the process

Input received from user

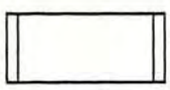
Output or response to user



Input received from PABX



Output or response to PABX



Procedure which may, or may not be described in details in another SDL diagram

A decision made during the operation of the process

Fig. A.1 SDL graphic symbols

APPENDIX B

SIGNALLING FOR SWITCHING SYSTEMS IN HK [16],[17]

B.1 Tone plan

Tone	Frequency (Hz)	Cadence
Dial tone	350 + 440	Continuous
Ringing tone	400 + 450	0.4/0.2 off/0.4/2.0 off
Busy tone	400	0.4/0.5 off
Congestion tone	400	0.25/0.25 off
Number unobtainable	400	Continuous

B.2 Tone levels

The nominal power level of tones when measured at the 2 wire termination point of the local exchange should be -8 dbm to -14 dbm for dial tone and 0 dbm to -12 dbm for other tones.

During the off period of a tone, the power of any noise from the tone source should not exceed -55 dbm.

B.3 Ringing frequency and voltage

Frequency = 16 to 27 Hz

Voltage = 75V rms nominal

(voltage may be as low as 50V rms and as high as 120V rms)

B.4 Dial pulse

Equipment should be capable of operating over exchange lines whose loop resistance is at the maximum of 1000 ohms. The equipment should be insensitive to exchange line polarity including any change in polarity which may occur during a call.

The pulse source should meet the following limits :-

1. Pulse repetition rate : 9 to 11 pulses per second
2. Pulse break-make ratio : within the range 63% to 72%

During the pulse-break period, there should be a disconnection of at least 100 kohms.

B.5 DTMF (Dual-tone multi-frequency)

Each signal should consist of two and only two of the signalling frequencies, in accordance with the following table. Both frequencies should be applied simultaneously to the line.

		High group frequency (Hz)			
		1209	1336	1447	1633
Low group	679	1	2	3	A
frequency	770	4	5	6	B
(Hz)	852	7	8	9	C
	941	*	0	#	D

All push-button and automatic DTMF dialers should meet the following signal timing requirements :

1. Minimum signal duration of 50ms.
2. Minimum interdigital pause of 50ms.

B.6 PCM coding

Quantizing noise over the dynamic range of the analog signal is made constant by compressing the scale. When the compressed PCM signal is reconstructed into analog at the receiving end, it is expanded to restore the dynamic range. The whole process is known as companding.

Where the coding of voice-frequency signals for digital switching is by means of PCM, Mu-law companding should be adopted.

$$v = V * \frac{\log (1 + u |x| / V)}{\log (1 + u)}$$

v = output of speech compressor

V = overload level of speech compressor

u = dimensionless constant, 255

x = input to speech compressor

REFERENCES

[1] W. Scott Currie
LANs Explained, A Guide to Local Area Networks
Ellis Horwood Limited, 1988.

[2] Technical Bulletin, 9104-970-001-NA
Issue 1, January 1989.
Mitel Corporation.

[3] Fred T. Schulz
ISDN In-depth, pp 20
Computer Technology Research Corp., 1989.

[4] A. Luvison, F. Toft
A High-Capacity Mutliservice Local Area Network
High Speed Local Area Networks,
O.Spaniol and A. Danthine (Editors),
Elsevier Science Publishers B.V., 1987.

[5] James P. Gray, Daniel Avery Pitt, Diane P. Pozefsky
LAN-PBX Gateway Alternatives
Local Communications Systems : LAN and PBX,
pp 143 - 156
Elsevier Science Publishers B.V., 1987.

[6] Kaazuyuki Maebara, Takzo Takeuchi
Network Interconnection between Digital PBX and LAN
Local Communications Systems : LAN and PBX,
pp 157 - 169
Elsevier Science Publishers B.V., 1987.

[7] William W. Wu an Adam Livne
ISDN : A Snapshot
Proceedings of the IEEE, Vol 79, No. 2, Feb., 1991.
pp 103 - 111

[8] Shuo-Yen Robert Li
A 100% Efficient Media-Access Protocol for
Multi-Channel LAN
to be published

[9] Bhaskarpillai Gopinath, Shuo-Yen Robert Li
High Capacity Communication Utilizing Static,
OR-Type Channels
United States Patent 4,768,189, Aug. 30, 1988.

[10] W. Y. Siu
Design of a Multimedia Local Area Network
using OR-type Channel
Thesis for the B.S. in Information Engineering,
Chinese University of Hong Kong, June, 1992.

[11] William D. Livingston
Prioritized Data Packet Communication
United States Paten 4,623,886, Nov. 18, 1986.

[12] Gerd E. Keiser
Local Area Networks
McGraw-Hill Book Co., 1989.

[13] Thomas C. Bartee
ISDN, DECnet and SNA Communications
Howard W. Sams & Company, 1989.

[14] Hermann J. Helgert
Integrated Services Digital Network
Architectures, Protocols, Standards
Addison-Wesley Publishing Company, 1991.

[15] Belina, F., and Hogrefe, D.
The CCITT-Specification and Description Language SDL
Computer Networks and ISDN Systems,
Vol. 16, pp. 311-341, 1988/89

[16] Customer Technical Guide Number 2
Requirements for The Connection of Customer-provided Apparatus to the Hong Kong Public Switched Telephone Network
Hong Kong Telephone Company, 1980.

[17] Customer Technical Guide Number 6
Requirements for 'Permission For Network Connection' of Supplier/Agent Installed and Maintained PABX Systems to the Hong KOng Public Switched Telephone Network
Hong Kong Telephone Company, 1983, (Amended 1987)

[18] C. C. Yuan
Thesis for the B.S. in Information Engineering,
Chinese University of Hong Kong, June, 1992.

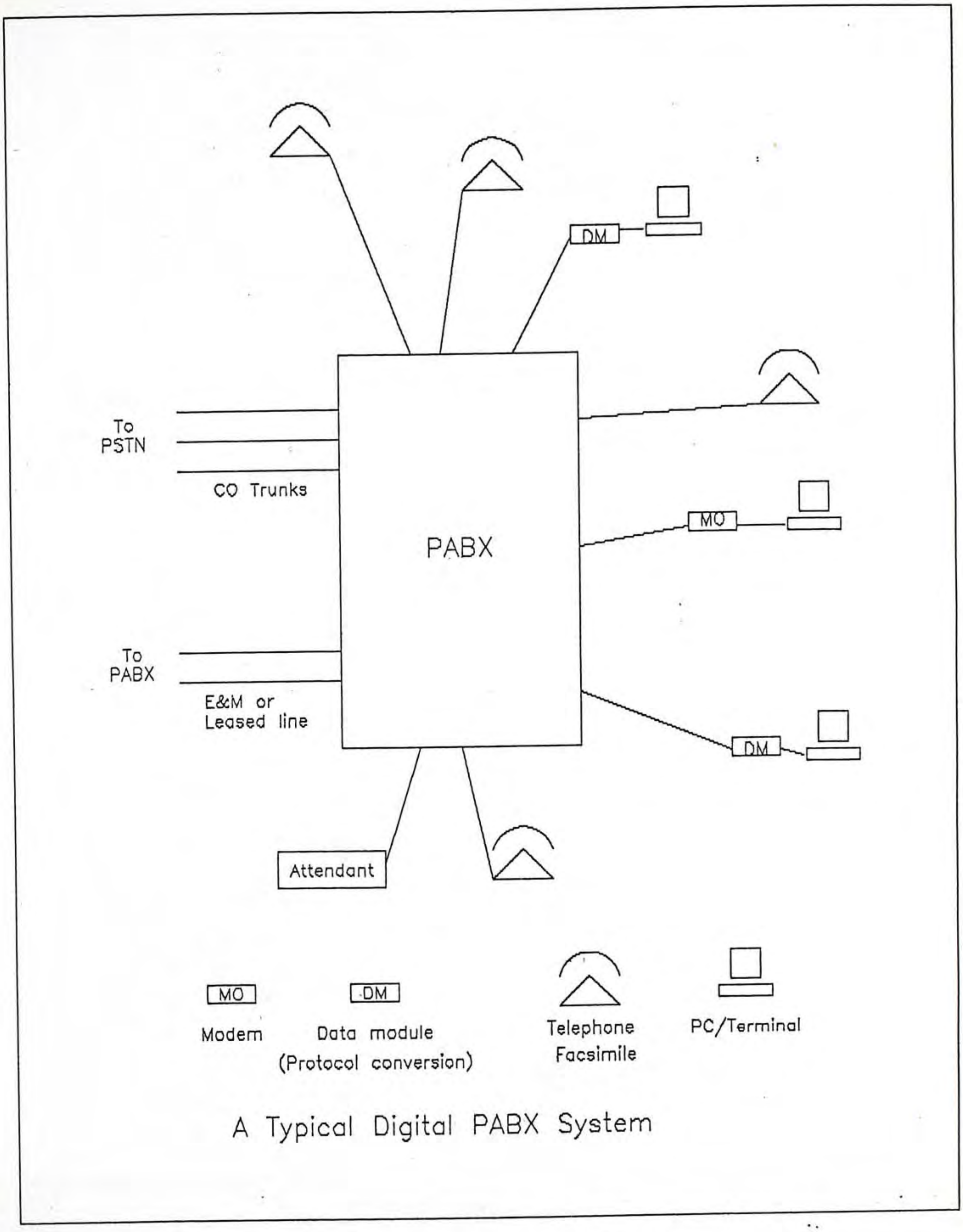


Fig. 2.1 A Typical Digital PABX System

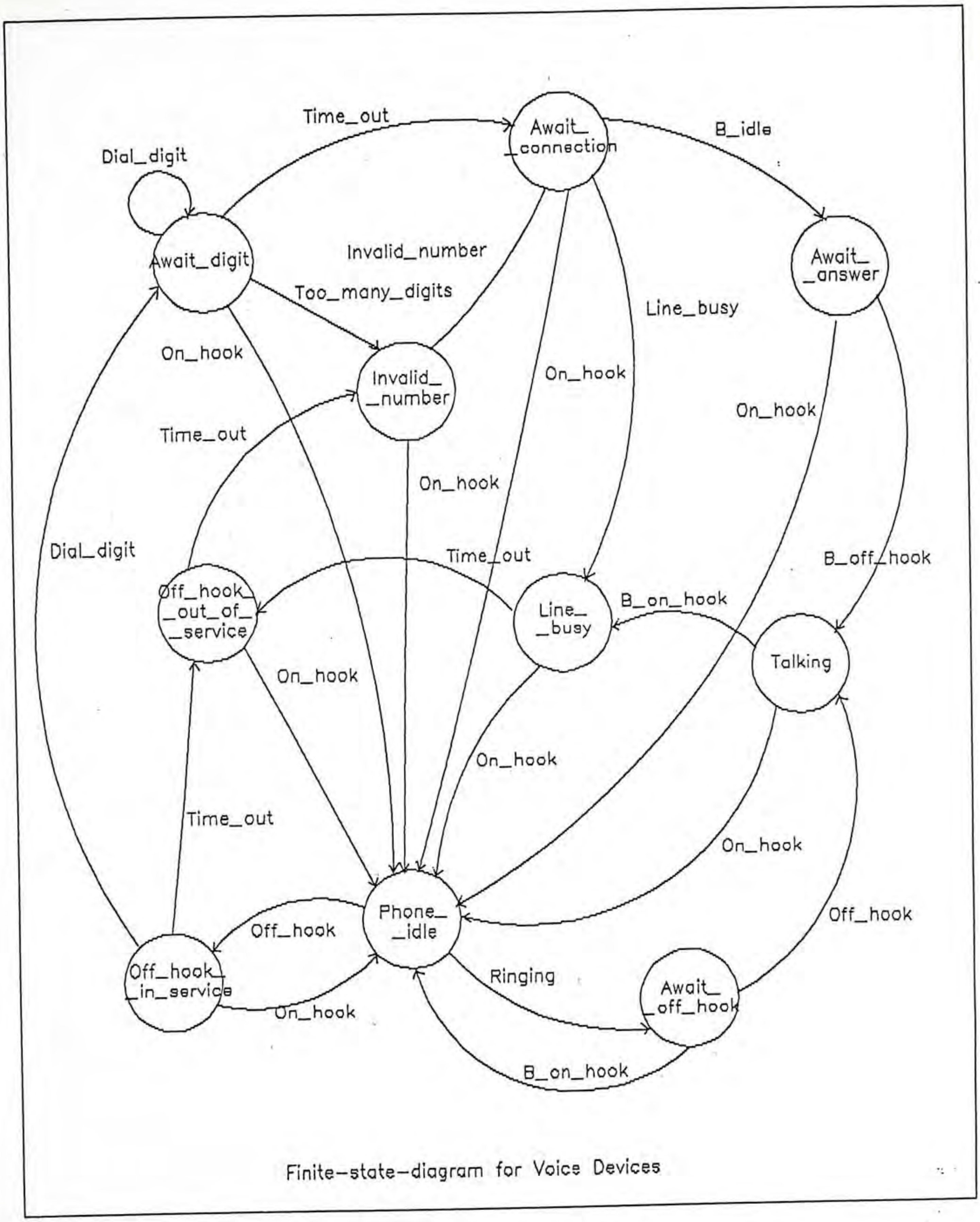


Fig. 4.1 Finite State Diagram for Telephones

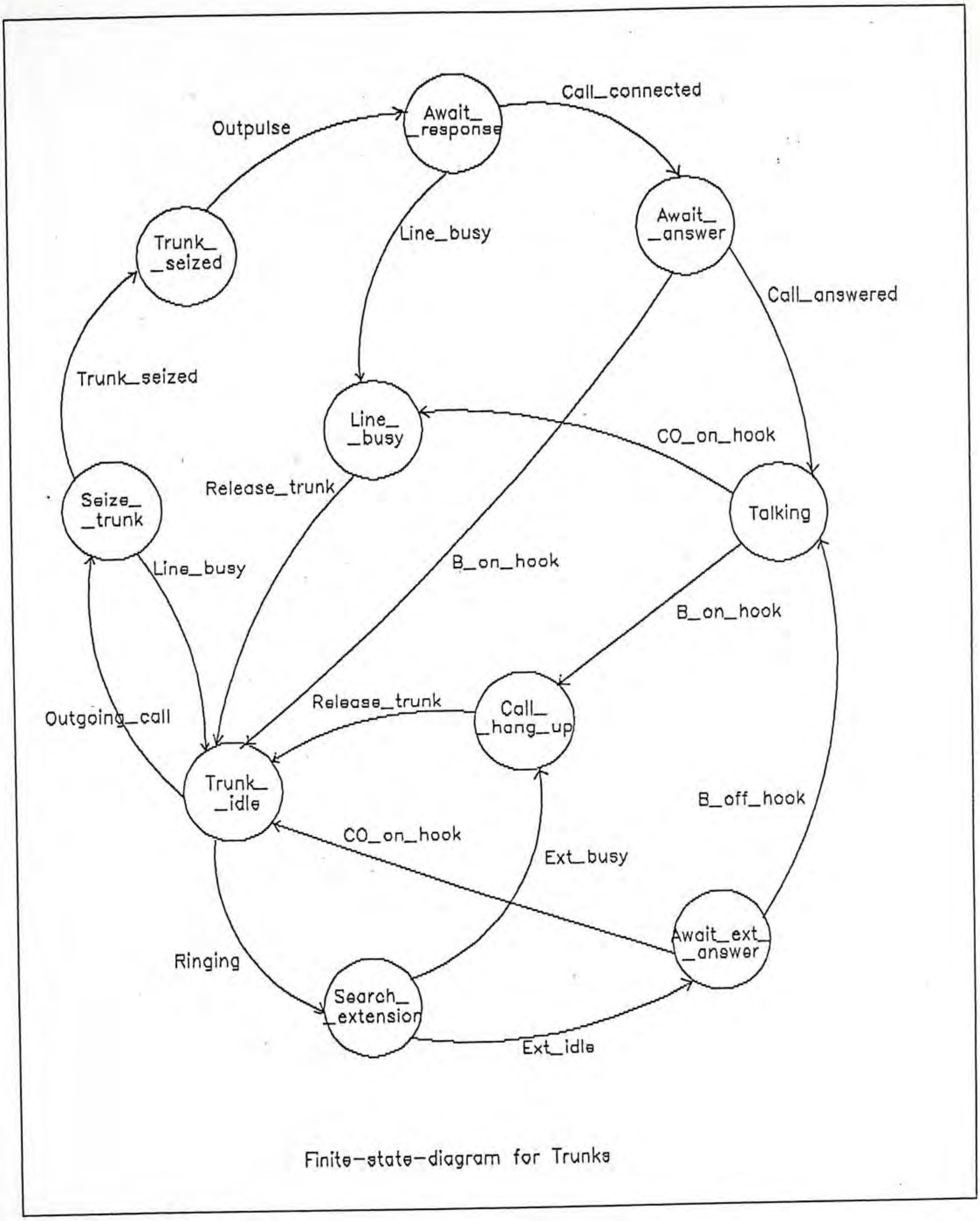
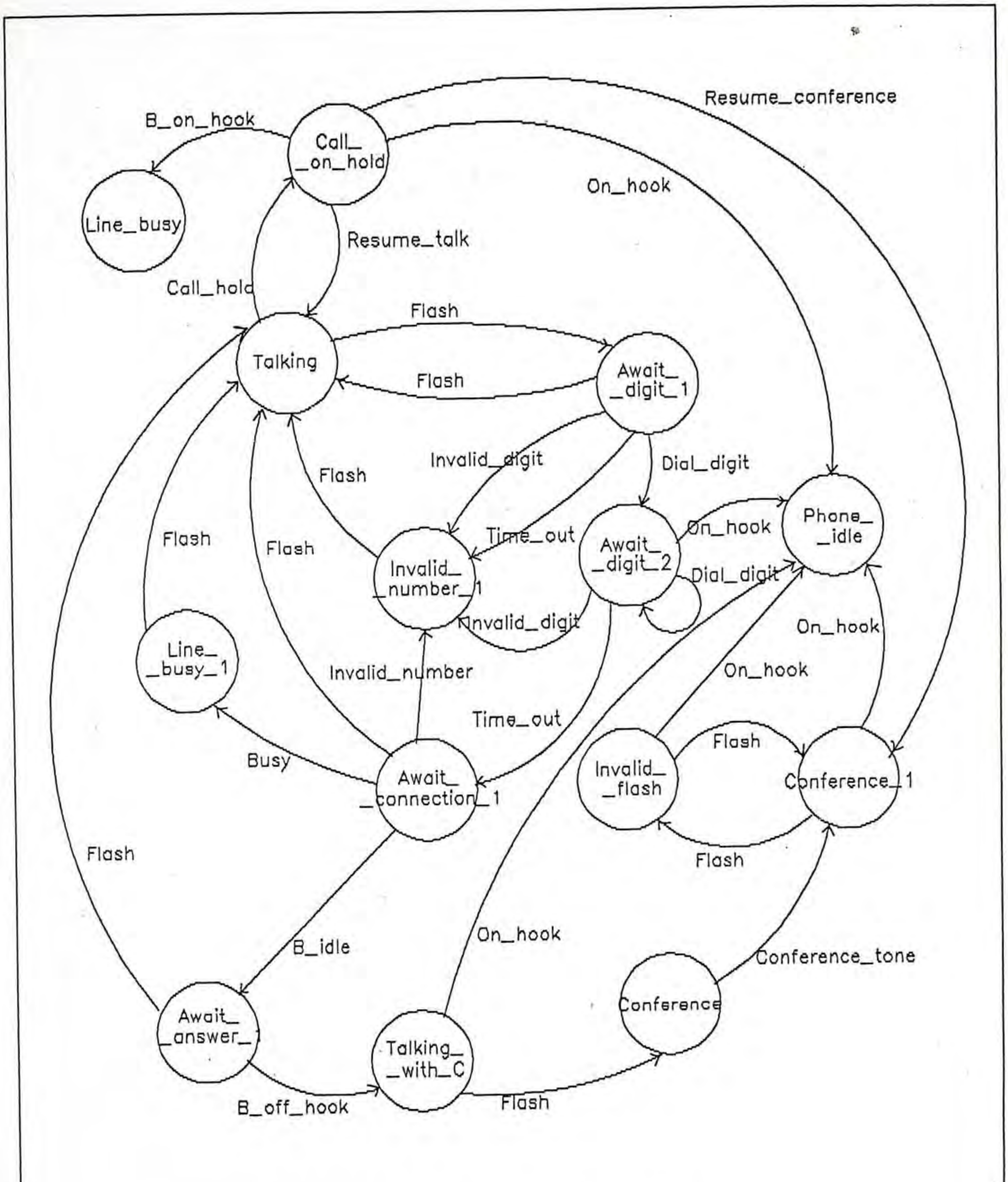


Fig. 4.2 Finite State Diagram for Trunks



Finite-state-diagram for Voice Conference

Fig. 4.3 Finite State Diagram for Telephone Conference

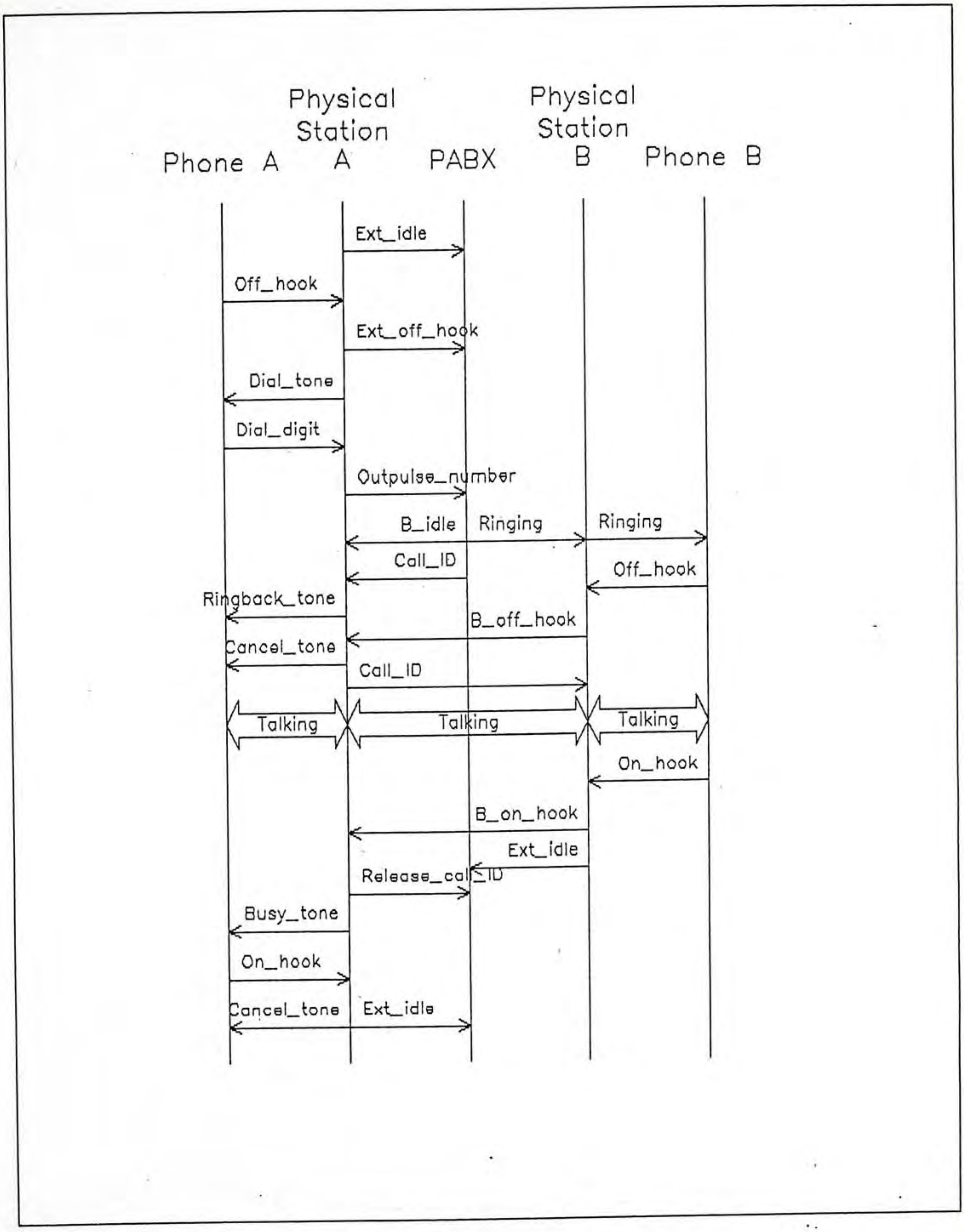


Fig. 4.4 Call Processing for Telephones

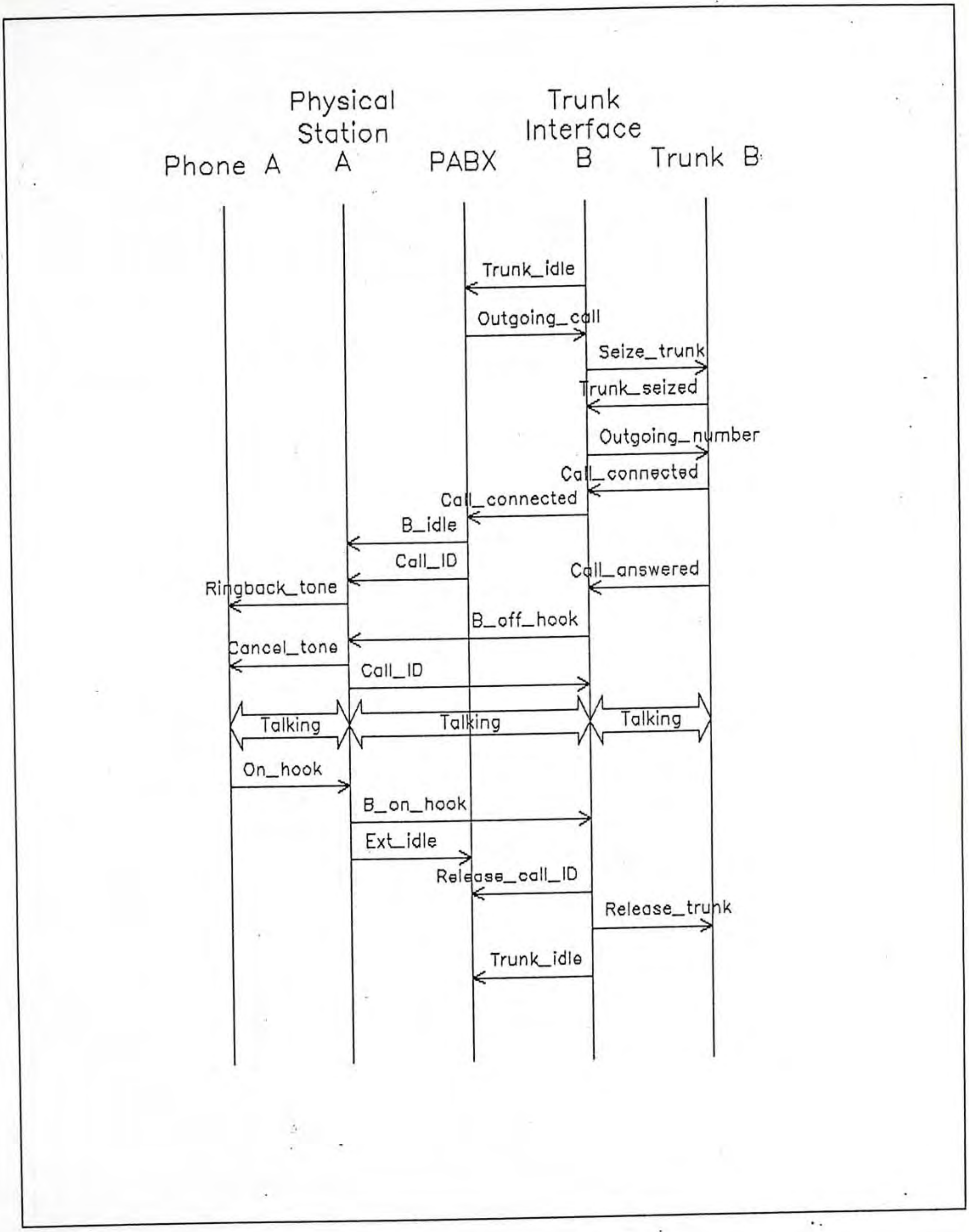


Fig. 4.5 Call Processing for Trunks

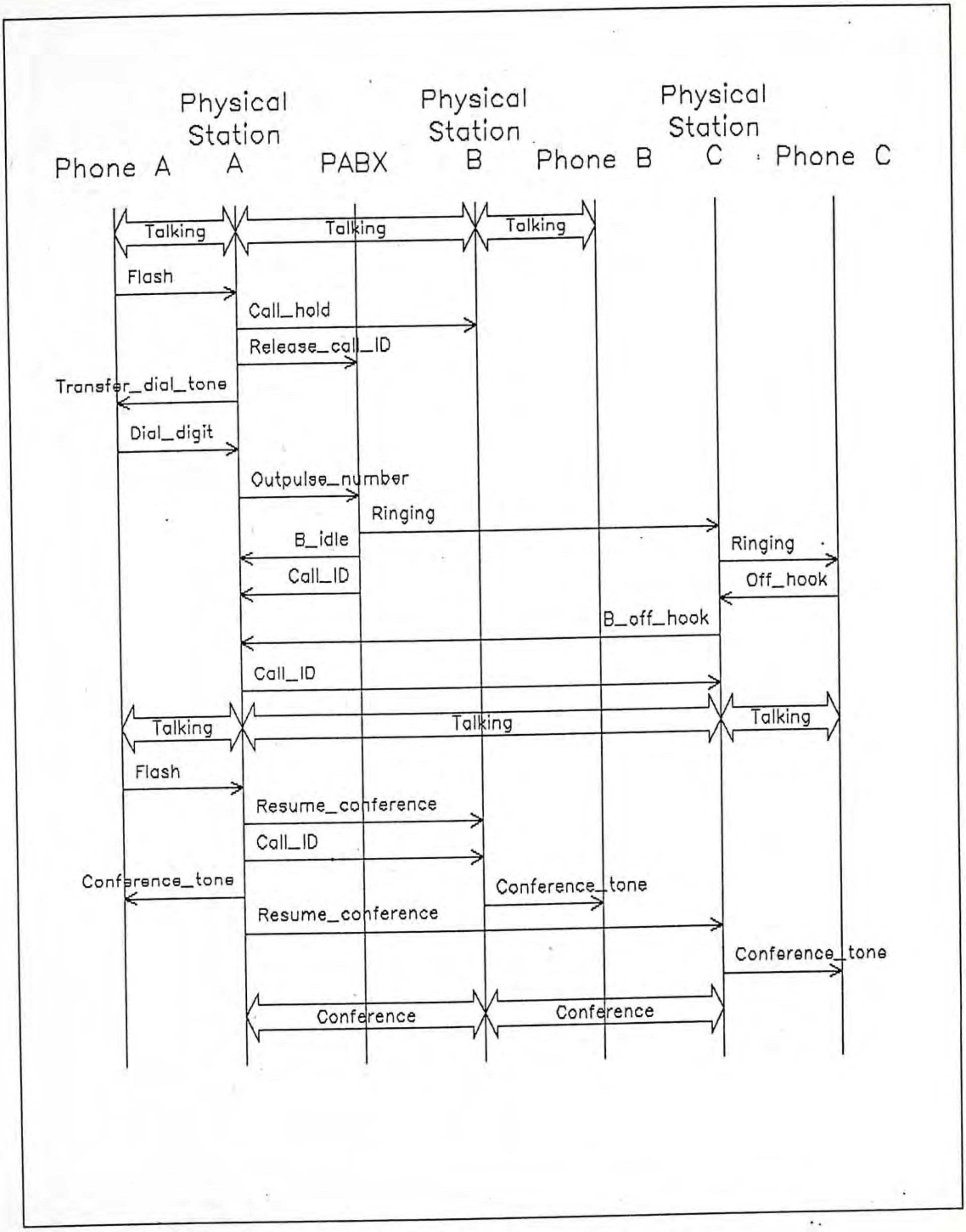


Fig. 4.6 Call Processing for Telephone Conferences

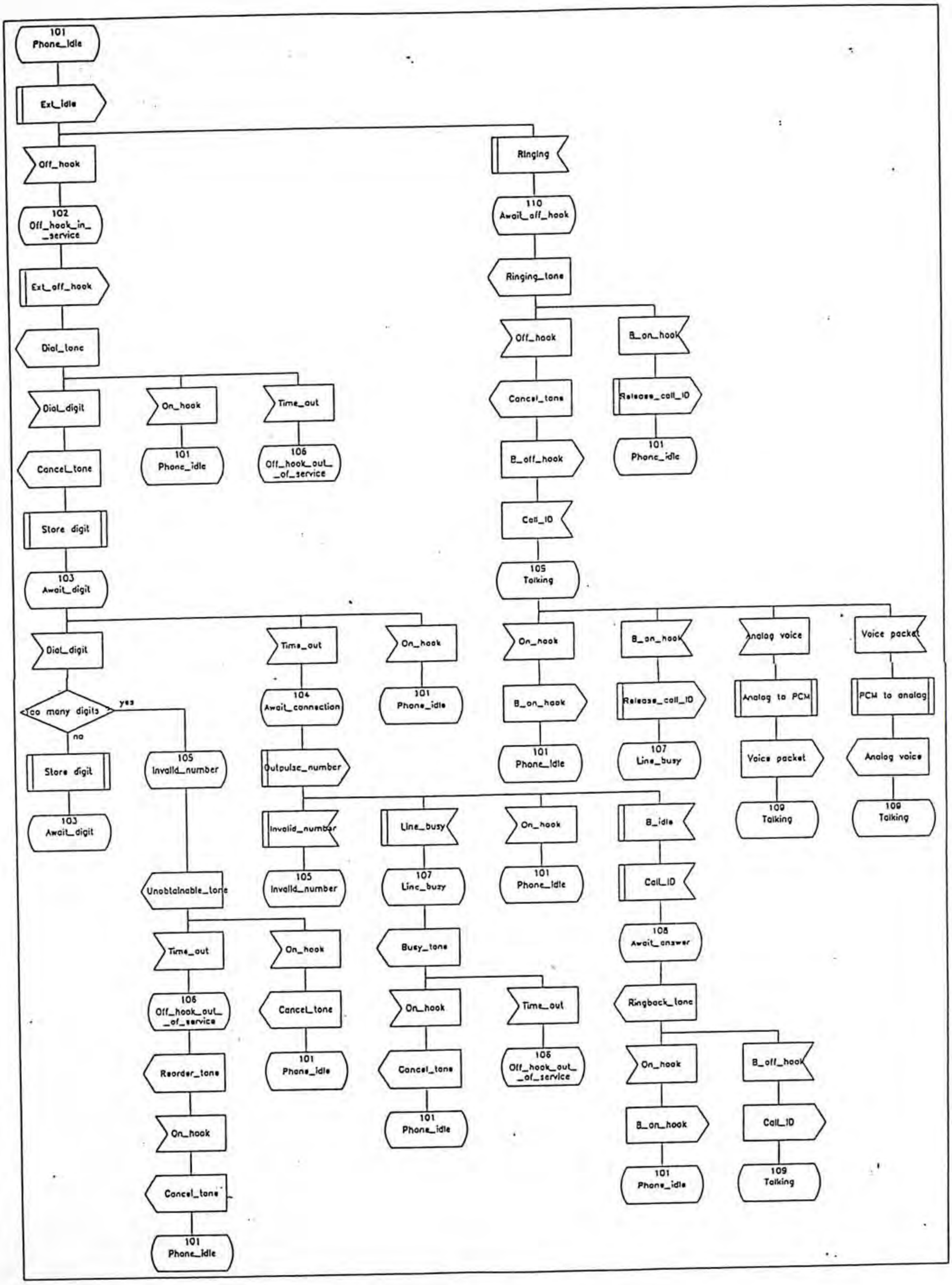


Fig. 4.7 SDL for Telephone Calls

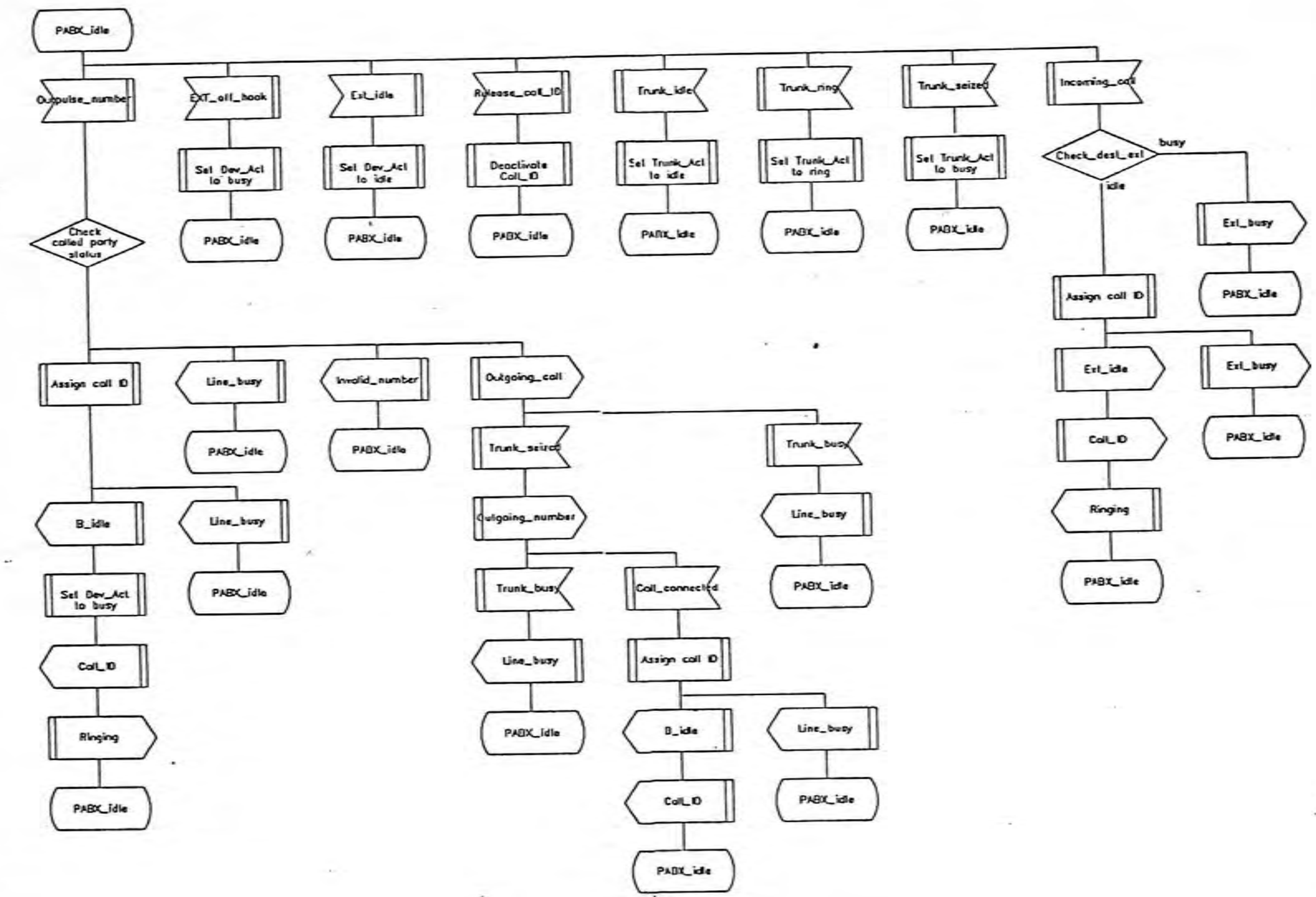


Fig. 4.8 SDL for PABX Processes

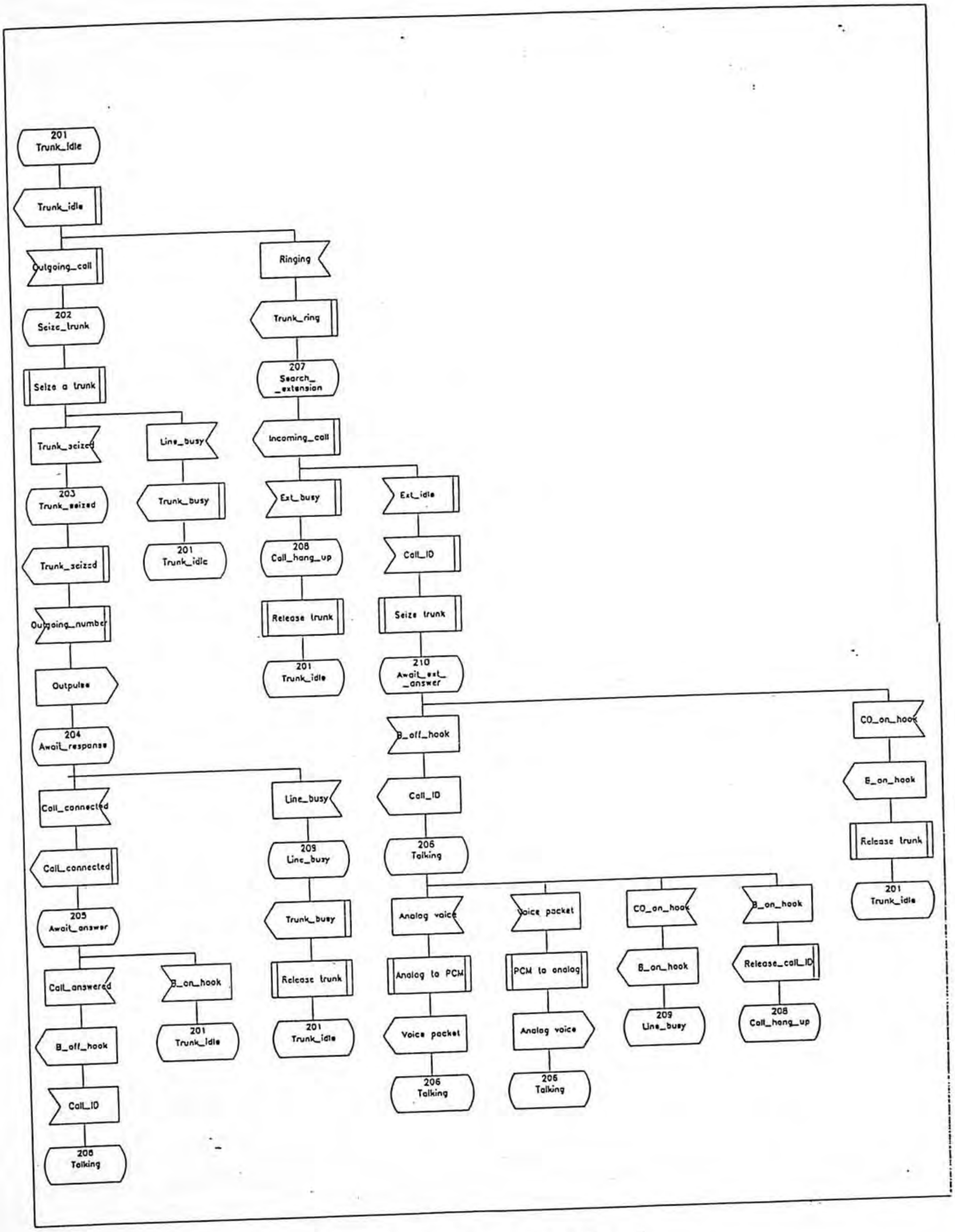


Fig. 4.9 SDL for Trunk Calls

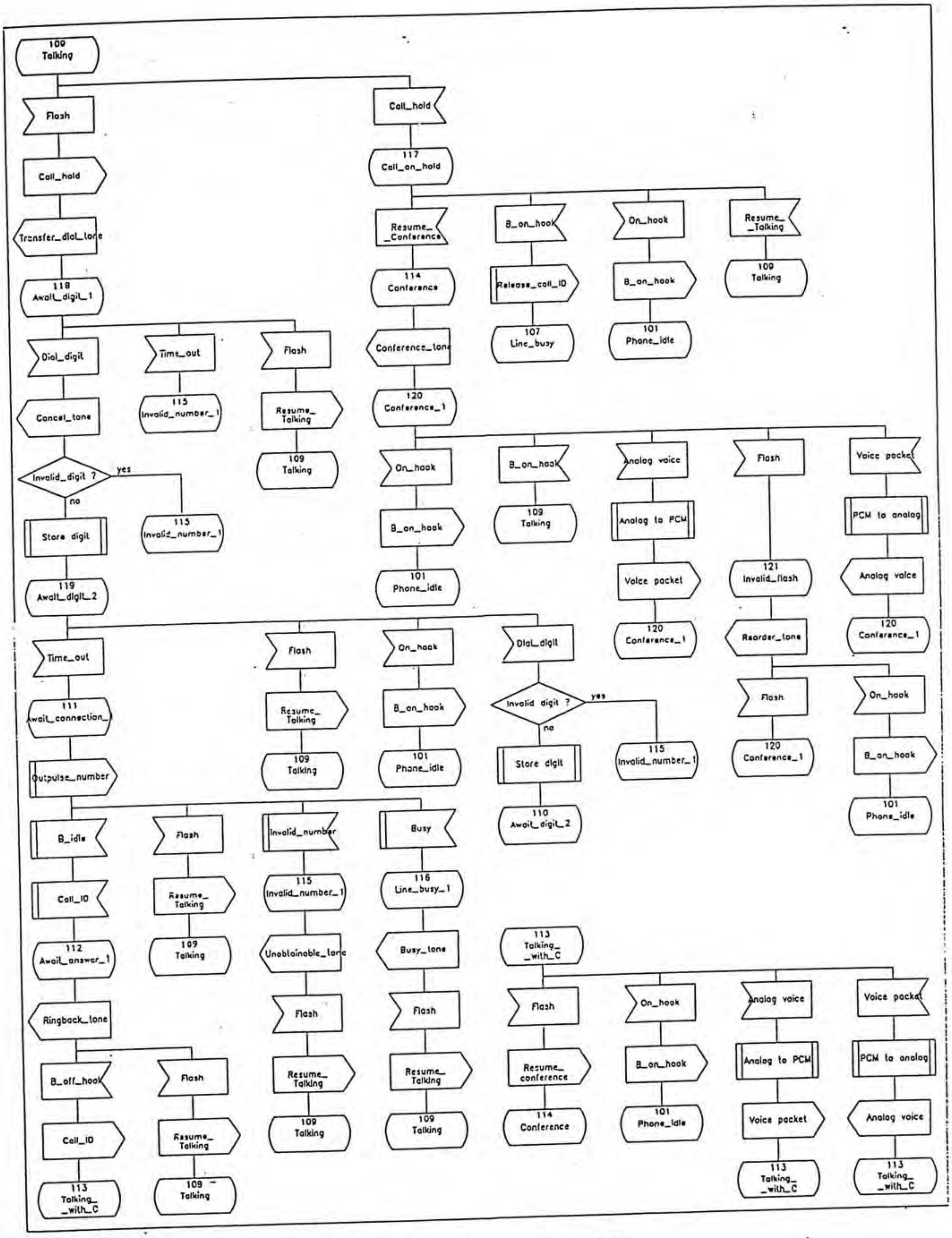


Fig. 4.10 SDL for Conference Calls

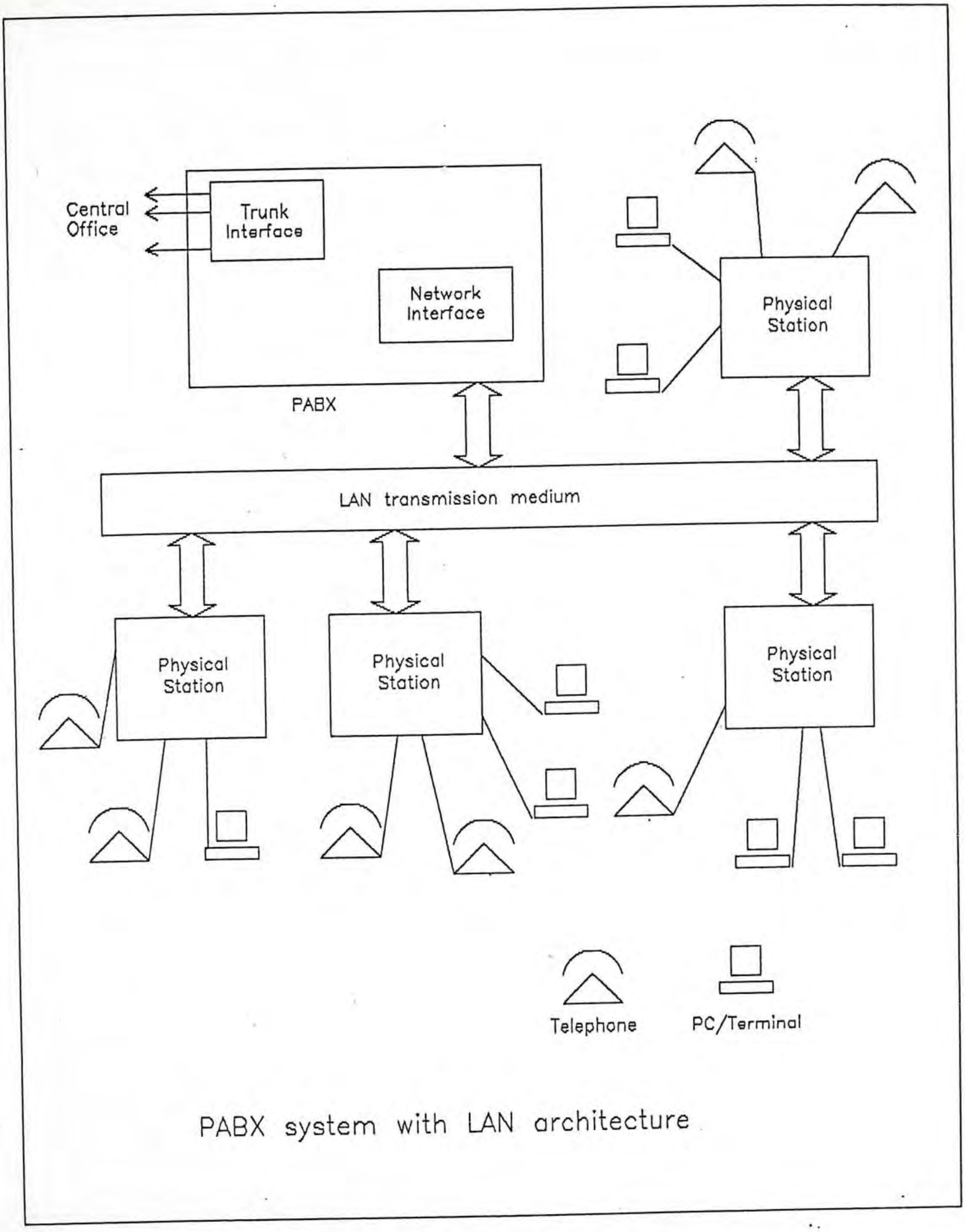


Fig. 5.1 Integrated PABX/LAN System

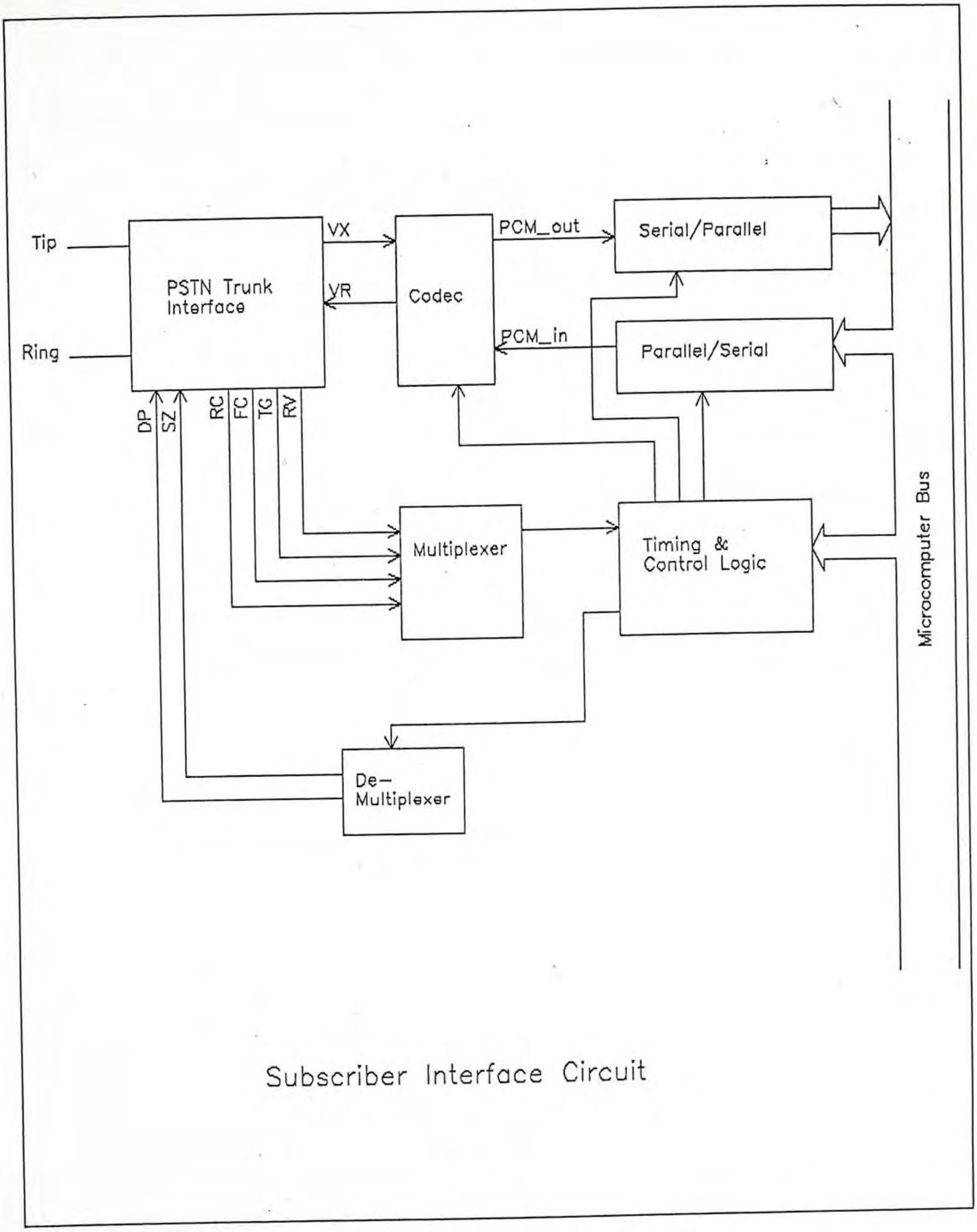


Fig. 5.2 Subscriber Interface Circuit

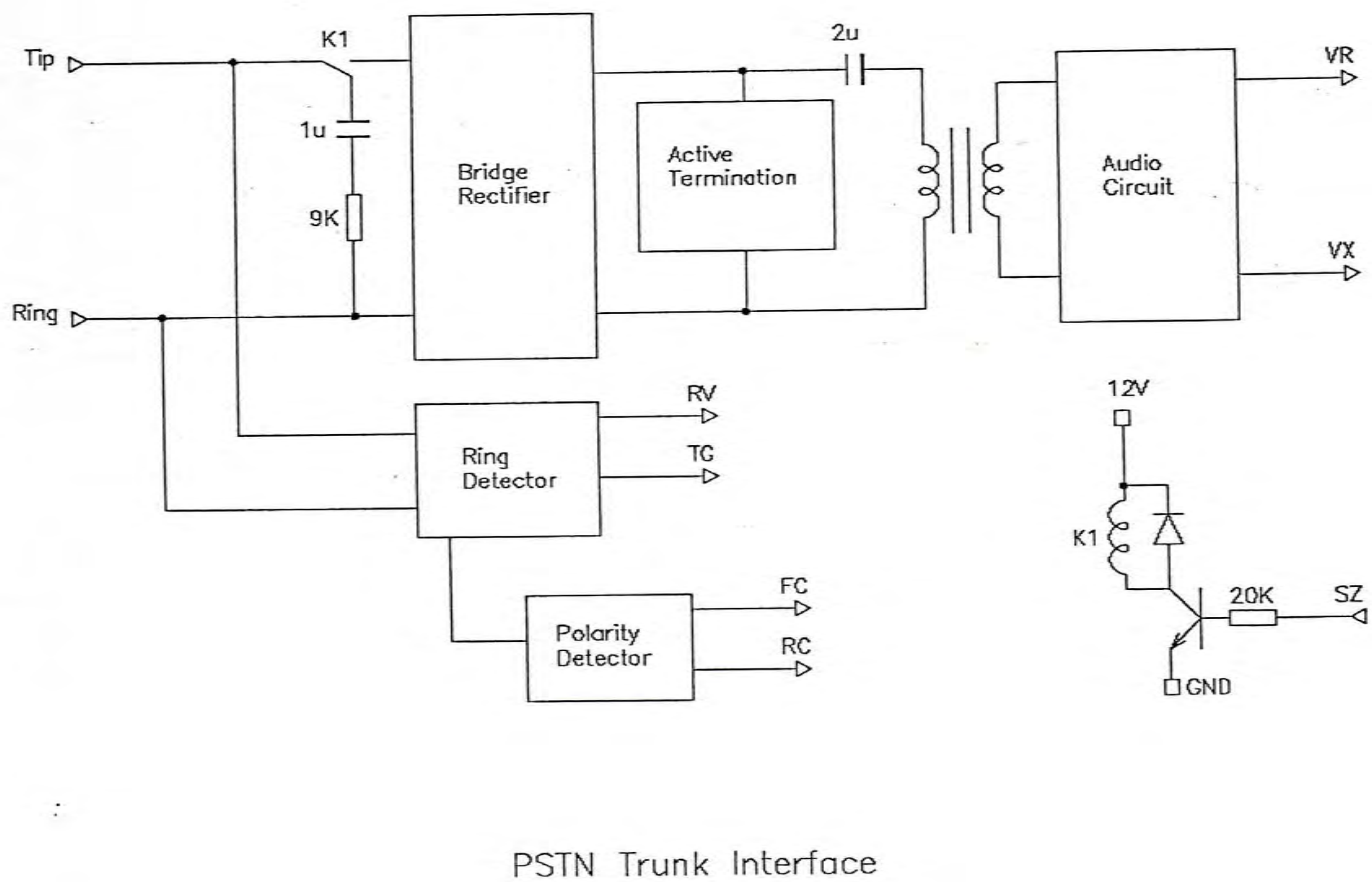


Fig. 5.3 PSTN Trunk Interface Circuit

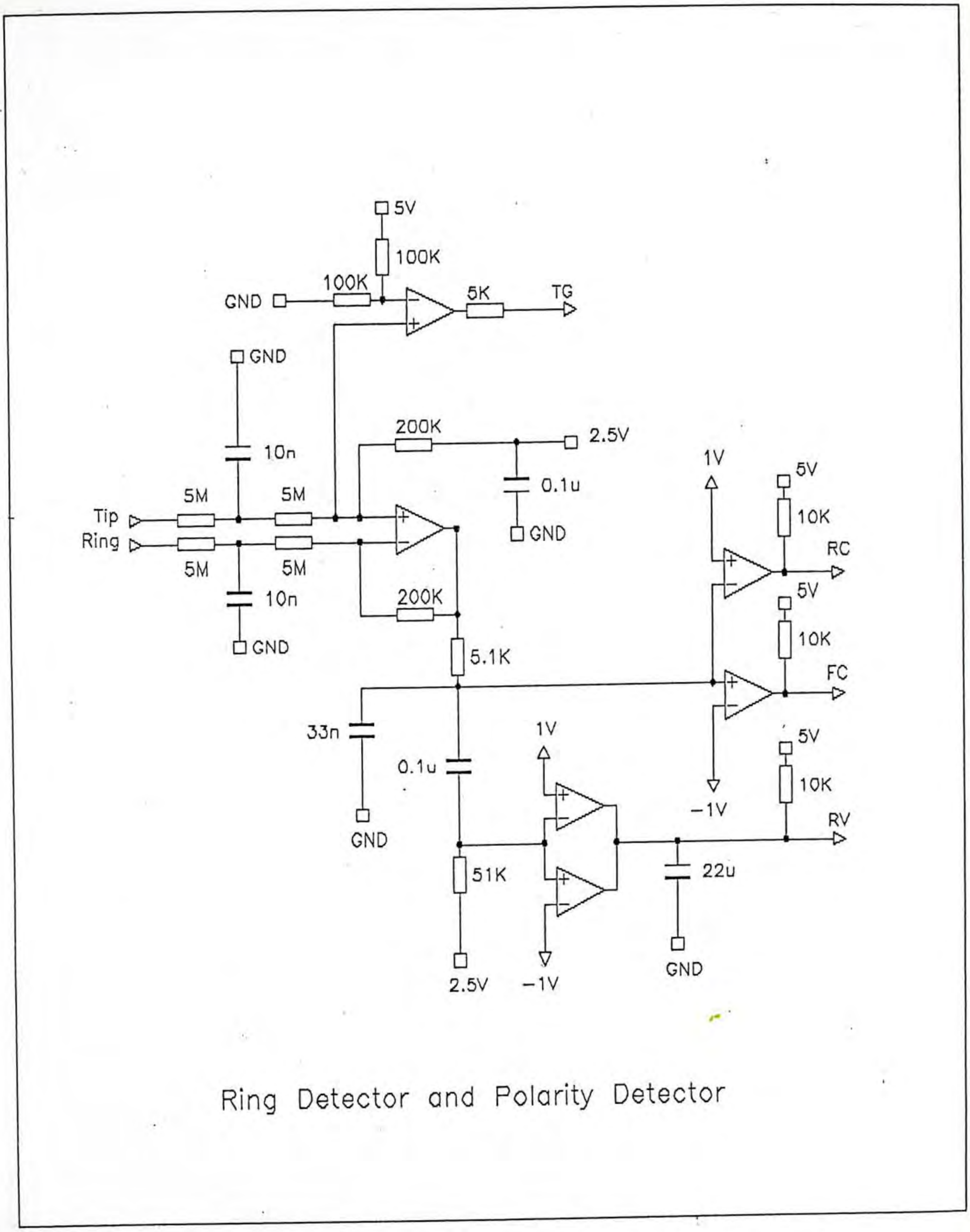


Fig. 5.4 Ring Detector and Polarity Detector Circuits

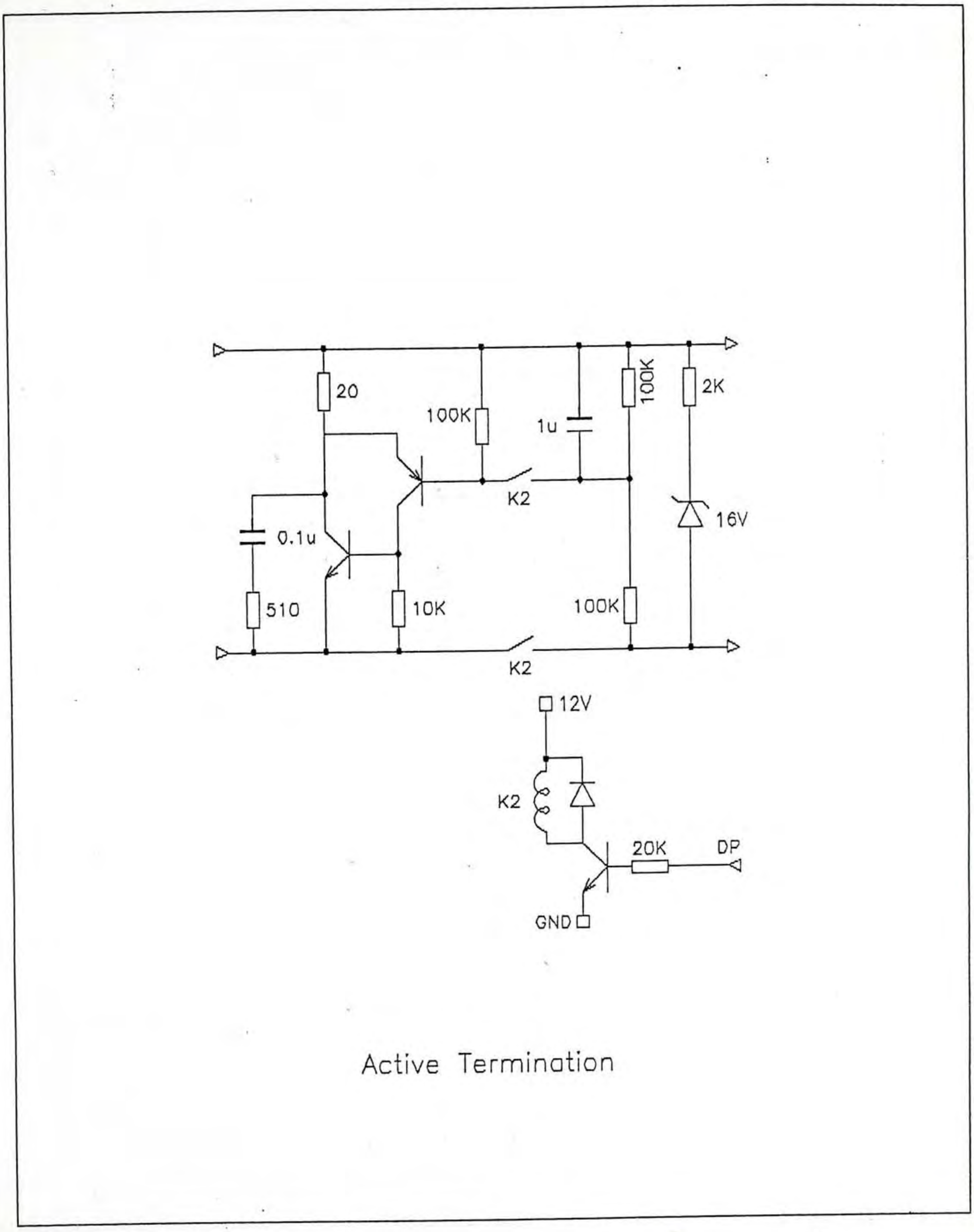


Fig. 5.5 Trunk Active Termination

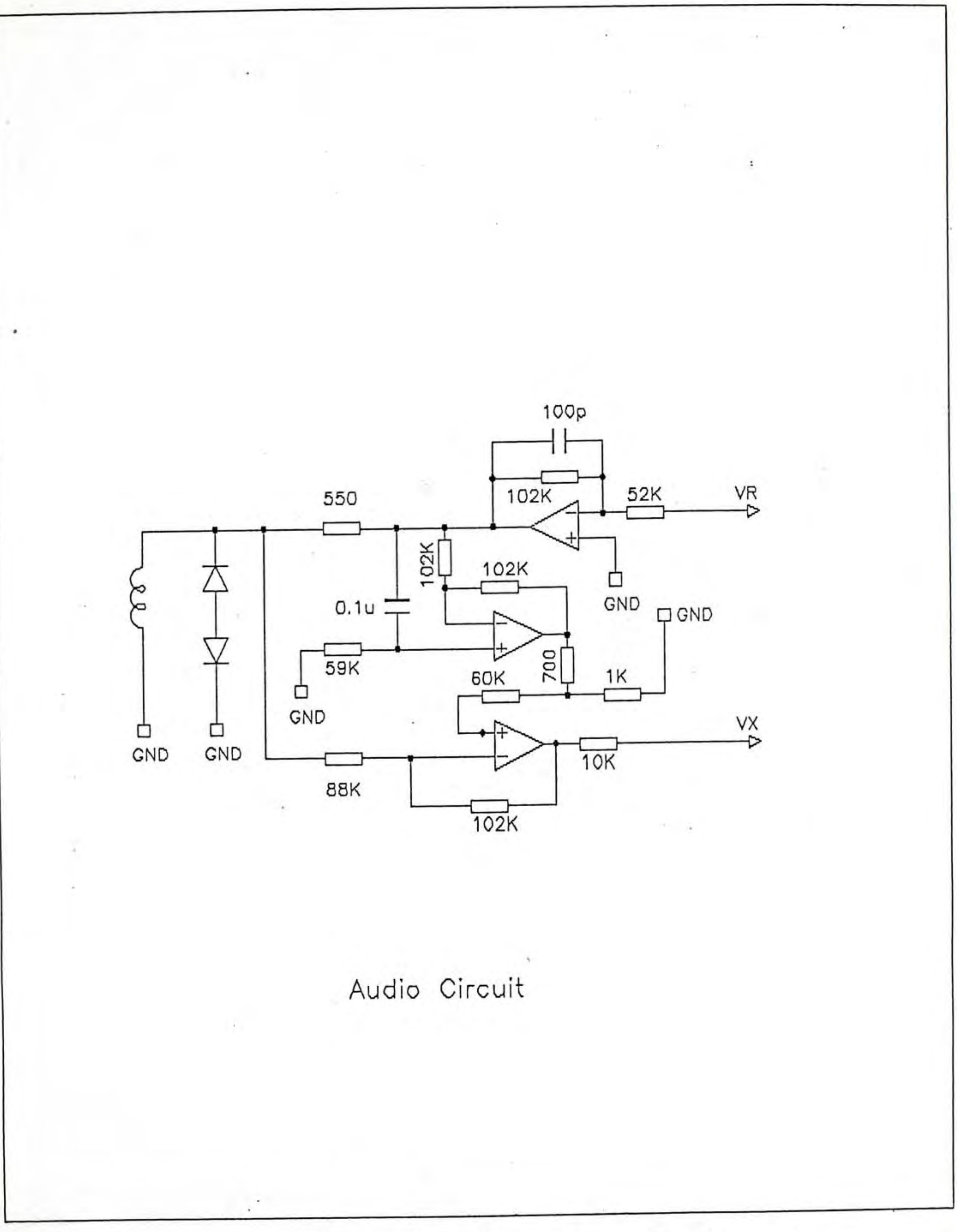


Fig. 5.6 Audio Circuit for Trunk Interface