# Everything you always wanted to know about SNA*

by

Sanders Data Systems

* BUT WERE AFRAID TO ASK

# The Sanders' Data Systems Primer

# on

# Systems Network Architecture (SNA)

# and

# Synchronous Data Link Control (SDLC)

First Edition (June 1975)



This document has been prepared as an educational service using the latest available literature on System Network Architecture. It does not constitute a specification for any Sanders product.

CHAPTER 1

# SNA Explained

## What is S.N.A.?

SNA stands for Systems Network Architecture, a new uniform structure for the implementation of teleprocessing networks. SNA was developed by IBM in order to achieve simplicity and increased efficiency in computer communications.

SNA establishes conventions which encompass:

- A line control discipline (SDLC),
- A communications access method (VTAM), and
- A communications control program (NCP/VS).

Each of these areas is integral to SNA. One of the major benefits of SNA is to provide users of telecommunications systems with a method of utilizing data processing capability without having to be concerned with the idiosyncracies of different communications procedures.

## What does Sanders know about SNA?

At Sanders' Headquarters in New Hampshire, communications engineers have been studying both SNA and SDLC for a great deal of time. This brochure constitutes Sanders' analysis of SNA. It was prepared to give you a simple introduction to the subject so that you can begin to evaluate its benefits for your organization. As you read these questions and answers, bear in mind that if you decide to employ SNA in your communications network, Sanders will be ready to support you.

## How does SNA differ from BSC?

In the first place, SNA and BSC are not directly comparable. SNA, which is a communication system architecture, is much broader in scope than BSC, which is a line control discipline. A more appropriate comparison would be between SNA and the entire BSC teleprocessing environment.

Unlike the BSC environment, where separate terminals and different communications lines are often required for each type of application, SNA establishes a single discipline for all types of applications and eliminates the need for the duplication of facilities. Because of the uniformity of this architecture, your host computer applications programmers will be able to ignore "handshake conventions" and communications procedures, because these will be handled by the system software.

There are other, more detailed, differences between SNA and the BSC environment, some of which will be illustrated by the following text, but the end result is as stated here.

## What about SDLC?

Synchronous Data Link Control (SDLC) is one part of SNA; specifically the line control discipline. The purpose of this discipline is to insure the integrity of data transmitted over one data link (i.e. a telephone line, a satellite circuit, etc.). Information on the type of data being transmitted, its ultimate destination (it may have to travel over multiple data links), the type of response expected from the addressee, etc., are handled by other parts of the SNA architecture.

## What are the advantages of SNA?

There are several immediate benefits which can be derived by using an SNA communications network. The most important of these are:

1) Use of full-duplex transmission capability: more efficient, less expensive communications.
2) Hardware independence of applications: many different types of applications can be run on the same terminals.

3) Decreased main processor workload: more functions are handled by communications terminal controllers and front end processors.

4) Reduced requirement for communications lines: multiple applications can use common procedures over shared lines.

In addition, SNA has an important capability for future improvements known as Distributed Function (or Distributed Data Processing). Distributed Function means spreading the processing load throughout a network rather than having all processing performed by a central mainframe. Distributed Function allows the user to combine the advantages of interactive and remote batch applications on the same hardware. This has the effect of: 1) improving operator response times when processing is done at the terminal rather than at the host processor, 2) allowing for off-line backup of on-line data entry applications, and 3) improving overall system efficiency by performing some or all of the processing for an application independent of the central processor.

## Are there any disadvantages?

Yes. Depending on your present level of teleprocessing activity, implementation of SNA may require additional expenditures in some parts of your EDP budget. These costs occur in three areas:

1) Additional memory which may be required for VTAM, NCP/VS, IMS/VS, CICS/VS, etc.,

2) Terminal costs which may be higher for SNA terminals than for their pre-SNA counterparts, and

3) Conversion costs which may be necessary in order to get the full benefit of SNA's features.

## What will an SNA network look like?

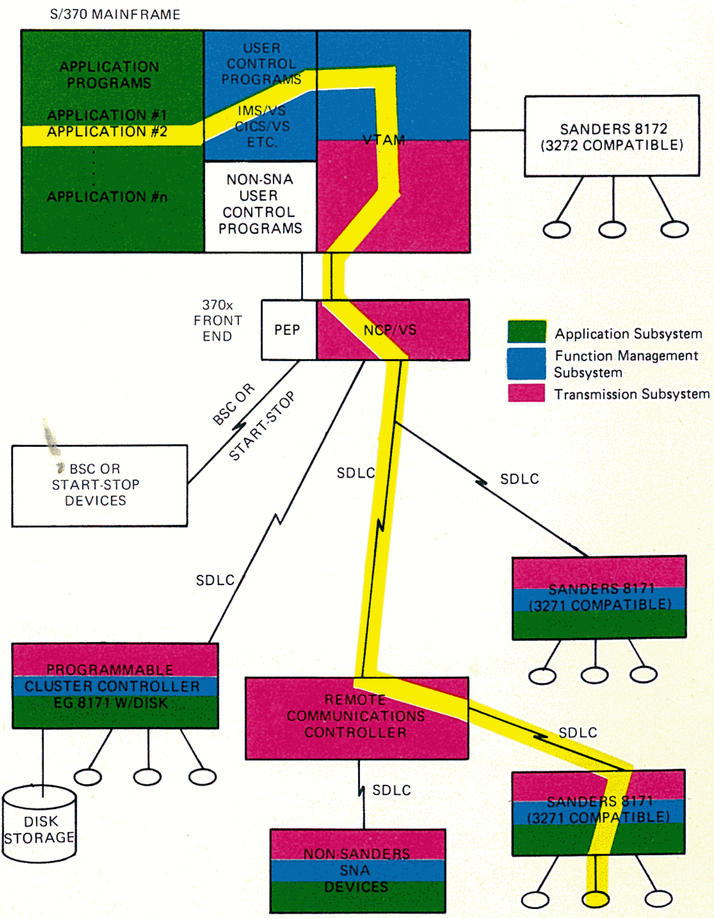
Figure 1 illustrates an SNA network. SNA configurations will have a 370 virtual system as host with a 3704 or 3705 front end and a network of terminals. It should be noted that different types of terminals can be multi-dropped on the same SDLC lines and that SNA also allows remote communications controllers to be used as concentrators.

The figure illustrates an interaction (known as a session) between an application program at the host and a corresponding end user at a remote 8170 Series terminal.

## Will this change the way I look at my computer center?

As shown in figure 1, the host processor in an SNA environment can be thought of as being divided into three parts; applications, user control programs, and an access method. Application programs (such as order entry, payroll, etc.) are designed to perform specific

# Figure 1 An SNA Configuration



user-oriented tasks and can be written with minimal consideration for communications system functions. Each application runs as a subtask under a User Control Program, such as IMS or CICS. The User Control Program uses an access method program which in SNA will always be the Virtual Telecommunications Access Method (VTAM).

The terminal user, in this system, establishes a logical "session" between himself and the application program running in the host computer. During this session the software and hardware between the user and his application program are invisible. VTAM and NCP/VS perform supervisory, control, and communication error handling functions without user or application program intervention.

The 370x front end processor, which is a programmable device, will contain the Network Control Program (NCP/VS). The 370x may also contain a Partitioned Emulator Program (PEP) which emulates a 270x and drives any BSC or Start-Stop devices which are included in the network.

## What kind of terminals will operate under SNA?

There will be a variety of terminals. Local interactive terminals such as the Sanders 8172 (3272 compatible) will be connected directly to VTAM via a channel interface. Remote interactive terminals, such as the Sanders 8171 (3271 compatible) will be connected to the 370x via SDLC communications lines. In addition, there will be programmable cluster terminal controllers with mass storage, such as the Sanders 8171 with disk, which will have their own application programs. These programmable clusters process data using either their own data base, the host's data base, or both. They will be able to act as interactive terminals, remote batch terminals, or hybrid combinations of these. Because it may not be desirable to change over to SNA for all applications at once, many networks will also include BSC and/or Start-Stop devices which will be connected to the 370x through the PEP as described above.

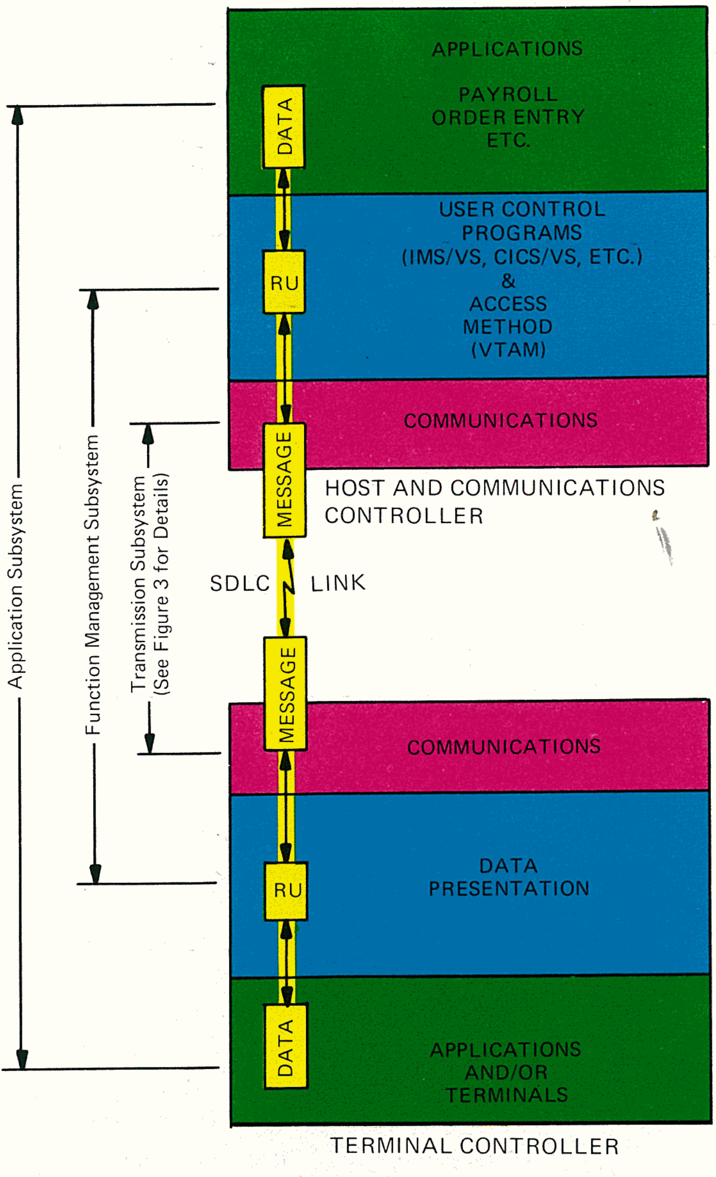
## Can you fill me in on the details?

The SNA protocols were designed to be flexible and efficient. SNA is a three-level architecture in which each level has a specific purpose. Figure 2 shows the three levels: the Application Subsystem, the Function Management Subsystem, and the Transmission Subsystem.

## What does the Application Subsystem do?

Under SNA, each layer is functionally independent of the other layers and each inner layer is transparent to all the surrounding layers. Therefore, the user interacts only with the Application Subsystem (the actual programs he

# Figure 2 SNA Subsystems



has written) to carry out his data processing task. In an on-line system for example, the application at the host computer is programmed as if it were interacting directly with the terminal operator. All of the communications procedures are handled by the two inner layers.

## What does Function Management do?

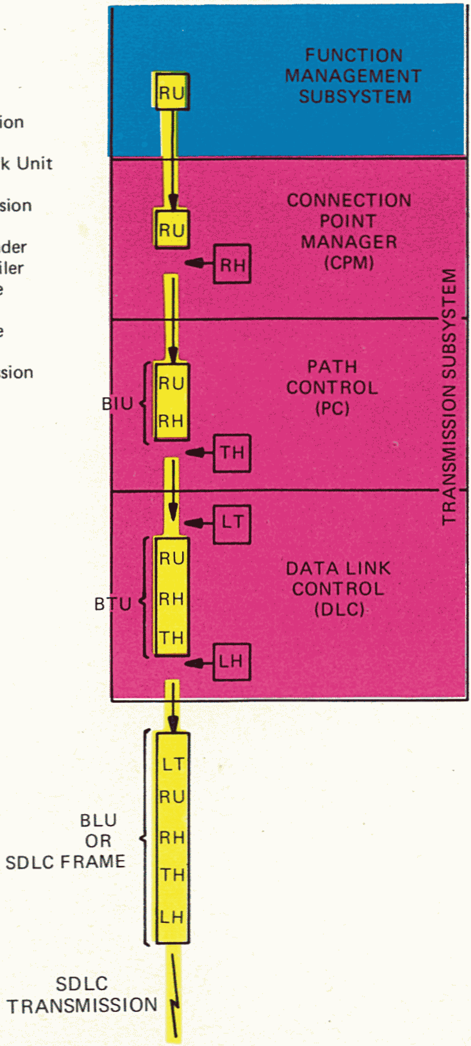
The Function Management Layer is concerned with making sure that data gets to and from the users in the proper format for the actual hardware they are using. As each transmitted block of data is passed to Function Management from the Application Subsystem, it is this layer that formats it into a device independent form, known as a Request Unit (RU), and presents it to the Transmission Subsystem.

## And then what happens?

The next layer, known as the Transmission Subsystem, is the one most intimately involved in the communications process. The Transmission Subsystem on the sending side, illustrated in figure 3, receives RU's from the Function Management Subsystem. A program called the Connection Point Manager prefixes each RU with a Response Header (RH), which indicates whether the RU is a command or data, what type of acknowledgement is expected, and whether the RU is part of a larger block of data which has been broken up into a chain. This RH plus RU combination is called a Basic Information Unit.

## Figure 3 SNA Transmission Subsystem Elements

BIU = Basic Information Unit
BLU = Basic Link Unit
BTU = Basic Transmission Unit
LH = Link Header
LT = Link Trailer
RH = Response Header
RU = Response Unit
TH = Transmission Header



A program called Path Control then adds on a Transmission Header (TH) which includes a sequence number, gives the origin and destination addresses, tells whether this is a high or normal priority RU, and provides a data count. A string of one or more RU's, each with an RH and TH appended, is called a Basic Transmission Unit (BTU).

Before the BTU can be transmitted, the Transmission Subsystem appends the SDLC Link Header (LH) and Link Trailer (LT) to form a Basic Link Unit (BLU), also known as an SDLC Frame, which is the form the message takes during transmission.

## So far only message sending has been covered. What about receiving SNA Messages?

When the message reaches the receiving end of the Communications Link it undergoes a process which is the "mirror image" of the process at the sending end. The Transmission Subsystem strips off the headers and trailers, takes any appropriate actions, and passes the remaining RU to the Function Management Subsystem. Function Management then formats the RU data for the specific device or application program for which it is intended and then passes it along to the Application Subsystem for use or processing.

## What role does SDLC play?

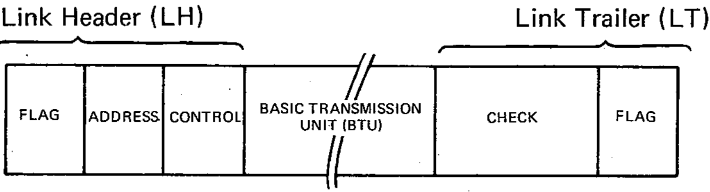
SDLC provides the setting within which data can be reliably transmitted across communications links. This is accomplished by establishing rules for the format and content of all messages in an SNA network. These rules are general enough to accommodate many different types of messages, including those composed of either data or commands or both. SDLC also accommodates full duplex operation so that the effective capacity of a given communications link can be increased.

## How are these SDLC capabilities implemented?

Whenever a message is communicated, it must travel over one or more data links. The format taken by the message as it traverses a data link is known as a Basic Link Unit (BLU) or an SDLC frame. This unit, as shown in figure 4, is composed of a BTU, as described above, enclosed in a Link Header (LH) and a Link Trailer (LT). The first field in the LH and the last field in the LT are "flag" characters which have a unique bit pattern, not allowed anywhere else in the message. These flags signal the beginning and end of each BLU.

Under the SDLC conventions, every data link has a primary node (usually the host) at one end and a secondary node at the other. Within the Link Header, the address field always refers to the secondary node,

# Figure 4 SDLC Message Format




Basic Link Unit (BLU)
or SDLC Frame

| Field | Contents |
|---|---|
| Flag | Unique bit pattern used to indicate the beginning and end of a message |
| Address | Address of the secondary station on the link. |
| Control | Link Control Information |
| BTU | Message and SNA headers |
| Check | Computed value used to determine transmission accuracy |

whether that node is the origin or the destination of the message. The Control field in the LH is used to convey control information to the receiving node and to confirm receipt of previously transmitted frames.

The value of the Check field within the Link Trailer is dependent on the contents of the message. It is first computed before transmission and is recomputed after receipt. If the recomputation does not agree with the check field contents, an error has occurred and the message must be retransmitted. This procedure insures data integrity across the communications link.

If a message must be transmitted across several links to reach its destination, new LH's and LT's are used for each link. In this case the contents of the message are held at every node along the way until that node has received confirmation of a successful transmission to the next node.

## This is very confusing. How can I get all these terms straightened out in my mind?

Because SNA is designed to be "transparent" to the user, you probably do not have to have an in-depth understanding of all the terms. However, if you should have a question about any particular term, refer to the SNA Dictionary in the back of this book.

CHAPTER 2

# Putting It All Together

## How do I use SNA?

Once you have the basic SNA building blocks, using them is easy! In most cases interactive applications will immediately begin taking advantage of the benefits cited above without application program modifications. Although all types of applications are expected to be upward compatible, some may require modification in order to fully utilize all of SNA's features. As new needs arise, SNA will allow for their development (taking full advantage of Distributed Function) in a manner which will be much simpler than was previously possible under the BSC conventions.

## Where do I use SNA?

You can use SNA in any teleprocessing network. The SNA conventions are applicable to all types of applications and terminals. This includes interactive or remote batch applications, leased or switched lines, and OS or DOS systems. It should be noted that not all of these capabilities will be supported under SNA initially. Some BSC compatible terminals, such as the Sanders 8170 Series, will be convertible for SNA usage, but many others will not.

## Who should I ask when I have questions about terminals in an SNA environment?

Ask Sanders! We have been a leader in communications technology for over 10 years. We pioneered what is now being called Distributed Function years ago with our 800 Series Programmable Display Systems. Now we can help you make the right decision for the efficient management of your communications network. For many companies, SNA is the right answer and we are committed to filling their needs. When you make a decision on SNA, yes or no, we will be ready to help you carry it out. So when you have a question, pick up the telephone and call:

Sanders . . . the intelligent answer.

# AN SNA DICTIONARY

**Application Subsystem** – The outer layer of the SNA architecture which contains application programs (and/or terminal logic) at both the host and terminal ends of a transmission.

**Asynchronous Data Flow** – High priority data flow, applied to messages which are to be transmitted on a high-priority basis, ahead of normal priority messages. See Synchronous Data Flow.

**BIU** – Basic Information Unit. Consists of a block of data known as a Response Unit (RU), preceded by a Response Header (RH).

**BLU** – Basic Link Unit. Consists of a Basic Transmission Unit (BTU) preceded and followed by the SDLC Link Header (LH) and Link Trailer (LT). This is the final format of the data which is transmitted over the SDLC communications lines.

**BTU** – Basic Transmission Unit. Consists of one or more Path Information Units, (PIU's) blocked together for transmission.

**Bracket Protocol** – The normal procedure for the sequencing of messages transmitted between two points in a communications network.

**BSC** – Binary synchronous communications. A communications discipline under which information is transmitted a block at a time. Used by most Sanders and IBM terminals.

**CCN** – Cluster Control Node. The controller for a cluster of terminals.

**CPM** – Connection Point Manager. The portion of the Transmission Subsystem which creates and interprets the Response Header (RH) which is appended to a Response Unit (RU) to form a Basic Information Unit (BIU).

**DLC** – Data Link Control. The part of the Transmission Subsystem which appends the SDLC Headers and Trailers to the Basic Transmission Unit (BTU) to create a Basic Link Unit (BLU).

**First Speaker** – In bracket protocol, the Logical Unit (LU) which can begin a series of messages (a bracket) between LU's without the other LU's permission.

**Full–Duplex** – A type of communications where information may be transmitted in both directions simultaneously across a single link. Requires special (Full-Duplex) transmission lines. Used in the SNA environment to attain substantial savings in communication costs.

**Function Management (FM) Subsystem** – The middle SNA layer which handles device dependent tasks.

**LH** – Link Header. The SDLC header which is prefixed to a Basic Transmission Unit (BTU) by Data Link Control. Used along with LT (see below) by SDLC communications equipment to perform the actual transmission.

**Half–Duplex** – A type of communications where information may travel across a communications link in only one direction at a time. Used in BSC and Start-Stop communications. (Note: Full-Duplex lines may be used for Half-Duplex communications in order to attain partial economies by eliminating "line-turnaround" time.)

**LT** – Link Trailer. The SDLC trailer appended to a Basic Transmission Unit (BTU) by Data Link Control. The LH, BTU, and LT together make up a Basic Link Unit (BLU) which is the actual unit of data transmitted along the SDLC communications lines.

**LU** – Logical Unit. An application program in a host processor or terminal. In an on-line terminal this is the standard terminal logic. The LU is the single entity in the Application Subsystem which is communicating with another entity at another point in the network.

**NAU** – Network Addressable Unit. An origin or destination point in an SNA network. NAU's always contain all three SNA layers.

**NCP** – Network Control Program. The program in a 370x which contains the SNA logic.

**Node** – An addressable point in an SNA network; it contains at least a Transmission Subsystem and may contain Function Management and Applications Subsystems.

**PC** – Path Control. The portion of the Transmission subsystem which creates and interprets the Transmission Header (TH) which is appended to a Basic Information Unit (BIU) to form a Path Information Unit (PIU). If the incoming BIU is too long, Path Control breaks it into segments, prefixing each with a Transmission Header (TH).

**PEP** – Partitioned Emulator Program. A 270x emulator program which runs in a 370x simultaneously with the Network Control Program (NCP) to allow some lines to run under BSC or Start-Stop protocols.

**PIU** – Path Information Unit. Consists of a Basic Information Unit (BIU) prefixed by a Transmission Header (TH).

**PU** – Physical Unit. The actual equipment which comprises an SNA node, e.g., a 370x or a Cluster Control Node (CCN).

**RH** – Response Header (or Request Header). A header prefixed to a Response Unit (RU) by the Connection Point Manager (CPM) which contains indications of whether the RU is data or commands, what type of acknowledgement is expected, and whether the RU is part of a chain.

**RU** – Response Unit (or Request Unit). The block of data or command which is being transmitted between application layers.

**Session** – The connection between a user and an application across communication lines. Begins when the user starts running the application and ends when he finishes using that application.

**Start-Stop Communications** – An asynchronous communications discipline under which information is transmitted one character at a time. Used by teletype and other similar systems. Not to be confused with asynchronous data flow.

**Synchronous Communications** – A general term which includes BSC communications. Not to be confused with synchronous data flow.

**Synchronous Data Flow** – Normal priority data flow, applied to all normal communications traffic during a session.

**SDLC** – Synchronous Data Link Control. The conventions used to set procedures for transmission in an SNA network and to determine the contents of the Link Header and Trailer in a Basic Link Unit (BLU).

**SSCP** – Systems Services Control Point. The portion of VTAM which keeps a directory of all the nodes, application programs, and end users in an SNA network.

**TH** – Transmission Header. A header which is prefixed to a Basic Information Unit (BIU) or BIU-segment by Path Control (PC) to form a Path Information Unit (PIU). The TH contains a sequencing number, an indication of the origin and destination nodes for the PIU, an indication of whether the PIU is to travel by Synchronous or Asynchronous Data Flow (see above), and a data count for the PIU.

**Transmission Subsystem** – The inner layer of the SNA architecture (containing the Connection Point Manager, Path Control, and Data Link Control) which formats and guides data through an SNA network.

# REFERENCES:


**Advanced Function for Communications: System Summary**, GA27-3309-0, IBM Corp., Aug., 74

"IBM's Protocols Part 2: SDLC", John E. Buckley, Computer Design, Jan., 75

"IBM's Systems Network Architecture", Gray and Blair, Datamation, Apr., 75

**IBM Synchronous Data Link Control: General Information**, GA27-3093-0, IBM Corp., Mar., 74

"IBM System Network Architecture & SDLC", Auerbach Computer Technology Reports, Auerbach Publishers, Inc., May, 75

**Introduction to Data Communications Systems**, SR20-4461-0, IBM Corp., Sept., 74

**Introduction to the IBM 3704 and 3705 Communications Controllers**, GA27-3051-2 IBM Corp., Apr., 74

**An Introduction to the IBM 3790 Communication System**, GA27-2767-1, IBM Corp., Sept., 74

**Introduction to VTAM**, GC27-6987-3, IBM Corp., Dec., 74

"Synchronous Data Link Control: A perspective", Donnan and Kersey, IBM Systems Journal, Vol. 13, #2, 1974

"The Synchronous Data Link Control (SDLC): How Will it Affect Your Life, and How Soon?", Auerbach Reporter, Computer Technology Division, Auerbach Publishers, Oct., 1974

"Synchronous Data Link Control: Part 1, 2, and 3", Communications Clinic, Modern Data, Feb–Apr., 1975

"Synchronous Data Link Control: Tutorial", J. Ray Kersey, Data Communications, May/Jun., 1974

**Systems Network Architecture: Data Flow and Concepts**, SR-20-4479-0, IBM Corp., Sept., 1974

**Systems Network Architecture, General Information**, GA27-3102-0, IBM Corp., Jan., 75

"Systems Network Architecture: IBM's Data Communications Game Plan", Auerbach Reporter, Computer Technology Edition, Auerbach Publishers, June, 1975

SANDERS
DATA SYSTEMS GROUP
Sanders Associates, Inc.

Daniel Webster Highway, South
Nashua, New Hampshire 03060

GA-75-288