Software Component Specification

SYSTEM:	MOD400
SUBSYSTEM:	LAN Facility
COMPONENT:	Configuration Reqmnts
PLANNED RELEASE:	Release 4.0
SPECIFICATION REVISION NUMBER:	1
DATE:	May. 1, 1986
AUTHOR:	I. L. Vivaldi

This specification describes the current definition of the subject software component, and will be revised as required to reflect design changes.

TABLE OF CONTENTS

REFERENCES

[1] CLM User Extensions, Richard Taufman, May 14,1979.

[2] Engineering Product Specification (H/W), Local Area Controller Subsystem (LACS), Rev F, A. C. Hirtle, Oct 4, 1984.

[3] Engineering Product Specification, LAN Software, R. Dhondy, Aug. 16, 1985.

[4] LAN S/W Component Specification, System Management, D. O'Shaughnessy Aug. 16, 1985.

[5] LAN S/W Component Specification, LACS Driver Interface Services, P. Stopera, Aug. 16, 1985.

[6] LAN S/W Component Specification, LACS Driver Megabus Services, P. Stopera, Aug. 16, 1985.

[7] LAN S/W Component Specification, LACS Driver Link Layer Services P. Stopera, Aug. 16, 1985.

[8] LAN S/W Component Specification, LACS Link Layer Protocol, H. King Aug. 16, 1985.

[9] L6 LAN Data Structures, P. Stopera - Aug. 16, 1985

DEFINITIONS

1.0 INTRODUCTION AND OVERVIEW

1.1 BACKGROUND

The development of the LAN Facility for Release 4.0 of MOD400 established certain requirements that have to be provided during MOD400 startup and at initialization time of the LAN Controller. This specification describes what has been provided to meet those requirements.

1.2 BASIC PURPOSE

The LAN Facility requires its own specialized processing at CLM time to scan user-supplied, LAN specific directives and establish the appropriate data structures and system linkages consistent with those directives, that are necessary for proper execution.

Some additional processing is required when the LACS (Local Area Controller Subsystem) environment has to be initialized. At that time, a special configuration file is processed to complete the LAN startup and initialization process.

1.3 BASIC STRUCTURE

MOD400 conventions for CLM extension modules call for a bound unit structure which has a root module containing a directory of the LAN CLM directive names, and 2 floating overlay modules which contain the processing logic required by the directives. The 1st overlay contains interpretive processing logic for the directives; the 2nd overlay finalizes the processing.

Post CLM processing of the Configuration File will be packaged as part of the System Management functionality resident in the L6.

1.4 BASIC OPERATION

This module will be executed in much the same fashion as is any extension to the Configuration Load Module function of MOD400. After the root module is loaded, it is linked with other CLM extension root modules. LAN specific CLM directives are then processed by the 1st phase with parameter information stored off in temporary structures. The 2nd phase is then loaded and final processing takes place with the construction of permanent control structures and the installation of BU's and tasks.

The processing of the Configuration File directives will be keyed off of the 1st Activate SAP RQIO issued by the 1st user application to establish an interface with the LAN Facility. At that time, the Configuration File will be OPEN'ed and directives will be read in and processed. This processing will result in the building and initializing of a small set of new data structures, and completing the initialization of CLM constructed data structures.

2.0 EXTERNAL SPECIFICATION

2.1 OWNED DATA STRUCTURES

Although this component is responsible for building a number of control data structures, they are not considered as owned by the component. Ownership of these structures will be assumed by the LACS Driver since the Driver is the primary user. Refer to Ref. [8] for a description of these structures.

2.2 EXTERNAL INTERFACES

This component will implement a collection of external interfaces that will call the standard CLM provided functions. These standard functions (currently there are 22 provided) are in place to supply common services to the various MOD400 CLM extension modules. These functions are very briefly described here. They are covered again in more detail in Section 3.0, the Internal Specification, of this document.

Function No.	Function Description
0	Get next parameter from directive
2	Validate directive parameters
3, 4 & 20	Get temporary or permanent memory
5	Report errors
11	Build Resource Control Table
17	Request free lrn
22	Locate controller table

Additionally, the LAN CLM extension module will execute the Spawn Task MCL on each occasion requiring the installation of an interrupt task for processing controller interrupts.

Initially, the LAN Facility will require no special functions in its interface with Release 4.0's load balancing algorithm. Basically, as each LACS controller is recognized, the LAN CLM extension module calculates its weight based on the number of adapters attached, and communicates this information to the central CLM module by way of Function call no. 22. After all controllers have been reported, load balancing takes place and controllers are allocated to processors.
Now, as Request tasks are created for servicing user requests issued to the LAN, they will be set up to execute on the same processor that the respective controller to which these requests will be issued, was assigned. A controller assigned to a specific processor will be instructed to direct its interrupts to that processor; consequently, an interrupt task for processing those interrupts will be set up to execute on that processor.

This arrangement essentially allocates an entire controller to a processor. Later refinements may allow, under certain configuration conditions, the allocation of a protocol layer instance to a processor.
Where multiple processors exist in a configuration, an interrupt task installed to execute on 1 processor will have its hardware level reserved on all the processors. This is a MOD400 convention for meeting resiliency requirements.

2.3 INITIALIZATION REQUIREMENTS

TBS.

2.4 TERMINATION REQUIREMENTS

This component has no special termination requirements.

2.5 ENVIRONMENT

Systems environment is MOD400/Release 4.0 and its successors.

2.6 TIMING AND SIZE REQUIREMENTS

Not applicable.

2.7 ASSEMBLY AND LINKING

This component will be implemented in L6 Assembly language and will use normal Assembly and Linker functions to prepare the CLM BU and the Configuration File processing module.

The CLM BU will be named ZQLCLM; it will be made up of 3 modules, ZQLCRT (the root module), ZQLCP1 (Phase 1 overlay) and ZQLCP2 (Phase 2 overlay.

The Configuration Processing module will be named SMCNFG and will be 1 of a number of modules linked together to make up the L6 LAN Manager BU, ZQLMGR.

2.8 TESTING CONSIDERATIONS

This component will be tested in a development environment to ensure correct operation. Ability to detect incorrect input and incorrect input sequences will be verified. Additionally, the component's ability to generate correct data structures and task instances at their specified priority levels and in the appropriate processors based on configuration directives will be verified.

2.9 DOCUMENTATION CONSIDERATIONS

Information relating to LAN configuration requirements will be contained in a number of documents:

- o component specification (this document) to convey development and maintenance information;
- o Software Maintenance Document (SMD) to provide a primary software maintenance reference for the LAN Facility;
- o MOD400 Manual on System Building and Administration (CZ02).

2.10 OPERATING PROCEDURES

LAN related directives, as defined in this section, are in 2 categories; CLM directives and Configuration File directives. CLM directives are to be included in the CLM_USER file and are processed by MOD400's Configuration Load Manager at system startup time. Configuration File directives are included in a standard sequential file and are processed after system startup by LAN System Management coincident with the 1st Activate SAP MCL issued by a user application.

2.10.1 LAN CLM Directives

TLIN, NLIN, LLIN, MLIN These directive names describe the presence of a layer instance, which means that a protocol layer entity of the type and instance specified in the directive will be configured in the LACS. This layer instance is being configured to provide its service to one or more user applications as defined by the following LCLUSR directive(s).

TLIN specifies a Transport protocol layer instance;
NLIN specifies a Network protocol layer instance;
LLIN specifies a Link protocol layer instance;
MLIN specifies the existence of the System Management layer instance.

LCLUSR Describes a user application interface to the protocol service described by the preceding layer instance directive. One occurrence of this directive is required for each such user interface desired.

Layer Instance Directive

This directive causes a number of control structures to be generated for managing and supporting the user interface(s) to this layer instance. An interrupt task is also created and installed at the priority level specified in the directive to process request completions being returned by the layer entity.

Format:

```
   TLIN    symb_name,protocol_type,chan_no,pl_addr,int_level
   NLIN
   LLIN
   MLIN
```

Argument Description:

symb_name
A name assignment through which this layer entity can be known and referenced. Since a layer instance defined in CLM also has to be defined in the Configuration File, this name must match its counterpart directive in the Configuration File.

Note:
In all cases where the 'symb_name' parameter is used, maximum length will be 16 characters, and allowable values will be 'A-Z', '0-9' and '_'.

protocol_type
Represents a particular protocol entity implemented for a layer. For example, the Network layer could have 2 protocol types implemented, a connectionless protocol and a sub-network protocol for DSA X.25 support. The particular type would be specified by this parameter, i.e., 'cnls' for the connectionless protocol and 'snap' for the sub-network protocol. Currently, the protocol types that have been defined are as follows:

Management layer	'mgmt'	(System Management)
Transport layer	'cls4'	(ISO Class4 protocol)
Network layer	'clns'	(connectionless network protocol)
Network layer	'snap'	(sub-network protocol for supporting DSA's X.25 network protocol
Link layer	'typ1'	(Type 1 connectionless protocol)

chan_no
The base channel number of the LACS controller that this layer instance resides in. A 4 hex digit field in the form "X'hhhh'" will be used this parameter. The 3 rightmost hex digits will always be "000"; the leftmost hex digit will be specified in the range 1 thru F.

pl_addr
This field is applicable only when the layer instance directive is describing a <u>link</u> layer instance. In this case, it specifies the address or position of the physical line on the controller. This parameter is specified as a decimal value in the range of 0 thru 3.

int_level
The hardware priority level at which will be installed an interrupt handler task for processing the interrupts generated by this layer instance entity in the LACS as it completes a request. This parameter will be specified as a decimal value, most probably in the range between 10 and 40. As a general guideline, the interrupt level for a LACS controller should be lower, i.e., a higher number, than other controller interrupt levels.

Local User Directive

This directive specifies that a local user application in the node being configured will be interfacing with the LAN Facility and specifically with the layer entity identified by the immediately preceding layer instance directive. A local user is regarded by the LAN Facility as an application entity availing itself of LAN services through a mechanism called a service access point (SAP). More than 1 local user may be configured as using the services of the layer instance, in which case, a SAP will be established for each of the users. The directive results in a Resource Control Table structure (RCT) being built on behalf of this local user. The RCT has a LAN specific extension area attached to it which is initialized by the LAN CLM process. An lrn is also allocated to this user.

Format:
LCLUSR symb_name,req_level

Argument Description
symb_name
A name assignment that is used for associating a particular MOD400 user entity with a service access point into the LAN Facility. This name is used during the initial association dialogue, as an argument in the Associate Local User MCL.

req_level
This value represents the hardware priority level at which request processing initiated by this LAN user will be executed.

2.10.2 LAN Configuration File Directives

These directives are described in 2 Component Specifications; this one and the specification for System Management. They are provided in both documents for ease of reading and comprehension. The set of Configuration File (C-F) directives consist of the following:

Controller Directive

This directive is optional and may be used to override defaulted parameters for flow control of read and write operations and for specifying no. of retries when an IOLD instruction is nak'd.

Format:
```
CTLR symb_name,addr,max_lcb,max_ios
```

Argument Description

symb_name
The symbolic name assigned to this controller; name is based on controller's megabus address, e.g, a controller with an address of "A000" will be named "lancta" where "lanct" is the constant part of the controller name.

addr
The base channel number of the controller; this parameter is specified as a single hex digit. In the example above, the base channel number of the controller would be "A".

max_lcb
Maximum number of lcb's allowed to be outstanding on this controller; specified as a 16 bit integer. Default value set to 99.

max_ios
Maximum number of retry attempts for executing IO or IOLD instructions when the controller is nak'ing the instruction. Default value set to 4.

Layer Instance Directive

This directive specifies an occurrence of a layer instance within the controller. Every layer instance described in CLM must also be described in the C-F. Additionally, any lower layer instance(s) which supports the CLM defined layer instance must also be described in the C-F. For example, a Link layer instance described in CLM is required to be described in the C-F along with the physical or MAC layer instance that is subordinate to and interfaces with the Link layer instance.
Where a layer instance directive is described, it will be followed by the local and remote SAP directives that are being configured for that layer instance.

Format:

```
TLIN
NLIN   symb_name,protocol_type,chan_no,pl_addr
LLIN
PLIN
```

Layer instance directives contained in the C-F are essentially the same as their CLM counterparts with the following exceptions:

Interrupt level is not required;
MLIN is not required in the C-F;
pl_addr is applicable only when the layer instance directive is describing a link (LLIN) or physical (PLIN) layer instance.
PLIN is required since 1 or more MAC layer instances must be configured; PLIN is not a CLM directive because user interface to MAC layer is not provided.

Argument description is the same as described for CLM layer instance directive.

Local SAP Directive

This directive specifies a local SAP (LCLSAP) through which a service requestor at a higher layer will make requests of the service provider that is the layer instance to which this LCLSAP belongs. LCLSAP's at higher layers are linked with LCLSAP's at immediately subordinate layers by way of mapping information contained in the LCLSAP directive. This linkage or association establishes a path for incoming data up through the multiple layers at a node and thereby enables delivery of that data to the user.

Format:

```
LCLSAP symb_name,mapping,phys_addr,flow_ctrl,attributes
```

Argument Description

symb_name

A name assignment that will be used in referring to, or associating with this LCLSAP. Where the layer instance that this LCLSAP belongs to is also described in CLM (and is therefore a layer instance with an interface to a user application, i.e., an exposed layer instance), this LCLSAP directive will have a counterpart LCLUSR directive, and the symb_name parameter must match the symb_name parameter of the LCLUSR directive. Where the layer instance that this LCLSAP belongs to is not described in CLM (unexposed), this symb_name will most likely be named in a mapping context by a higher layer LCLSAP directive, and result in a downward associative link when that higher layer LCLSAP is activated.

mapping

This parameter will be presented as 1 or more symbolic name elements where each element names a local SAP belonging to a layer instance at the next lower layer. Where more than 1 element is specified, the elements will be separated by a slash (/) character. A one-to-one or one-to-many downward mapping capability is therefore possible. For example, an LCLSAP directive being defined as belonging to a network layer instance would specify the name of an LCLSAP belonging to a configured link layer instance. That named link layer LCLSAP would be required to be configured.

phys_addr

Addressing information is specific to the layer in which the LCLSAP directive is being defined. Addressing conventions for LCLSAP's in each of three layers will be described here.

Link Layer

A LCLSAP address for link layer protocol is defined in IEEE 802.2 as consisting of a LLC address field (8 bits) plus the MAC address (which may be either 16 or 48 bits). Because the LACS architecture establishes a one to one correspondence between a link layer instance and a physical layer instance (or MAC adapter), the MAC address implicitly becomes part of each LCLSAP address and is therefore included in the link layer LCLSAP address.

Network Layer

For the Null Network Protocol, which is our first implementation, no addressing data is required. This parameter for a network layer LCLSAP would not be specified.

Note:

For the SNDCP sub-network protocol, a physical address would likewise not be required since there would be a one to one mapping between a sub-network layer LCLSAP and a link layer LCLSAP.

Discussion of network addresses in an Internetworking Protocol environment is deferred to a later time.

Transport Layer

Physical address for a transport layer LCLSAP is defined in ISO as consisting of a "transport selector" and a network LCLSAP address.

Our current implementation supports only 1 transport layer instance per controller and similarly, only 1 null network layer instance per controller. Only 1 network LCLSAP is required to be configured to support the transport entity, and as a result, a similar addressing situation is established between transport and network layer instances as exists between link and MAC layer instances. That is, the network LCLSAP address implicitly becomes part of each transport LCLSAP address and is therefore included with the transport selector value.

The transport selector portion of the address simply identifies the transport service user (DRMO, ISO Session, Fasttrack) and will be specified as a decimal integer. The network LCLSAP address will be specified according to our own internal conventions (since no standards exist for a null network protocol), and could also be specified as a decimal integer.

flow_ctrl

To be supplied

attributes

To be supplied

Remote SAP Directive

This directive describes a remote peer entity with which a local user of the layer instance wishes to communicate. REMSAP's at higher layers are linked with REMSAP's at immediately subordinate layers by way of mapping information contained in the REMSAP directive, similar to the manner in which LCLSAP's are linked. This linked sequence of REMSAP's is effectively established when a user application issues an Activate REMSAP RQIO. The linkage establishes a path for outgoing data and allows the identification of remote peers at each layer so that correct remote addresses can be formatted into the message header.

Format:
REMSAP symb_name,mapping,phys_addr,flow_ctrl,attributes

Argument Description

symb_name
A name assignment that will be used in referring to, or associating with this REMSAP. If the layer instance to which this REMSAP belongs has a direct interface with a user application, this symb_name will be used in an Activate REMSAP RQIO issued by the application. If the layer instance to which this REMSAP belongs is a subordinate layer, this parameter will be used in an internal association context, as described earlier.

mapping
This parameter will be presented as 1 or more symbolic name elements where each element names a remote SAP belonging to a layer instance at the next lower layer. Where more than 1 element is specified, the elements will be separated by a slash (/) character. A one-to-one or one-to-many downward mapping capability is therefore possible. In the normal case, a single name element is named in this parameter to specify the next downward step in the outward path for a message.
The network layer represents a special case for specifying this information in a network REMSAP directive. When Internetworking Protocol is implemented and routing options and decisions are possible, this parameter could specify multiple downward mappings naming different subordinate REMSAP's. This one-to-many mapping feature of the IP entity allows the configuring of more than 1 path to the same end point, and implies that the IP entity will make the decision as to which path to choose.

phys_addr
As described earlier for LCLSAP address specification, the format and content of this parameter is unique to the layer to which the REMSAP belongs.
For a Link layer REMSAP, the address consists of both the remote LLC address and the remote MAC address, specified in the same format as described for a Link layer LCLSAP.
For the Null Network protocol, no addressing information is required. For the SNDCP sub-network protocol, no addressing information is required as well. Because of the one-to-one relationship between SNDCP layer instance and Link layer instance in our architecture, the subordinate Link layer REMSAP address would insure delivery to the correct remote SNDCP peer entity.
The Transport layer REMSAP address will be specified as a transport selector field and a network REMSAP as described for a transport layer LCLSAP.

flow_ctrl
To be supplied.

attributes
To be supplied.

2.10.3 Example of a LAN Configuration

The following example illustrates the use of CLM and C-F directives in describing a LAN configuration.

The example describes a single user application that is to interface with the Type 1 connectionless link layer protocol and desires to communicate with 2 remote users. A single LACS controller with a single CSMA/CD adapter is to be configured.

CLM Directives

```
LLIN clsll,typ1,X'C',2,14
LUSR rfal,16
```

These two directives specify that a Type 1 link layer instance supporting a single user is to be installed in the LACS . The base channel number of the controller is "C", i.e., the megabus address is "C000", and the adapter, or physical line, that this link layer is to interface with is assigned position number 2 on the controller. An interrupt task at H/W priority level 14 is to be installed on the L6 processor for processing interrupts from this layer instance on this controller.

The single user interfacing with the connectionless Type 1 service will be assigned a system name of "rfal". The user will also be assigned an lrn and a request processing task will be installed at H/W priority level 16 for processing requests to the LACS Driver.

Configuration File Directives

```
LLIN clsll,typ1,X'C',2
LCLSAP rfal,abcd,X'01',X'0004'
REMSAP rlu1,,X'01',X'0003'
REMSAP rlu2,,X'01',X'0002'

PLIN ethrnt,csma,X'C',2
LCLSAP abcd,,X'0004'
```

The LLIN directive should have an identical counterpart in the CLM directive set; this is so because the layer instance being described is being configured to support a user interface and has to be described in both places.

1 local SAP and 2 remote SAP's are specified as belonging to the Link layer instance. Since the link layer instance in this configuration supports a direct user interface, the local SAP directive is the representation of that interface. The symbolic name specified in this directive is therefore required to be the same as that specified for the LUSR directive in CLM. The mapping parameter establishes a link with the single local SAP, "abcd", belonging to the MAC layer instance. The 2 remaining parameters in this LCLSAP directive specify its full address consistent with IEEE 802 standards. In this example, a 16 bit station address (the 2nd of the 2 parameters) is assumed.
The 2 remote SAP's, "rlu1" and "rlu2" contain no mapping parameters because mapping is not required for remote SAP's at the link layer. The remote SAP address is declared in the same format as it is for the local SAP.

The PLIN directive specifies that a MAC layer instance supporting CSMA/CD protocol is to be installed in the LACS. Controller and adapter address parameters are required to be the same as specified in the LLIN directive.

The single local SAP directive for the MAC layer instance (there is only 1 allowed) simply specifies its name and, if desired, attributes of the MAC layer.

Figure 2.1 illustrates the relationship of the entities described in this configuration.

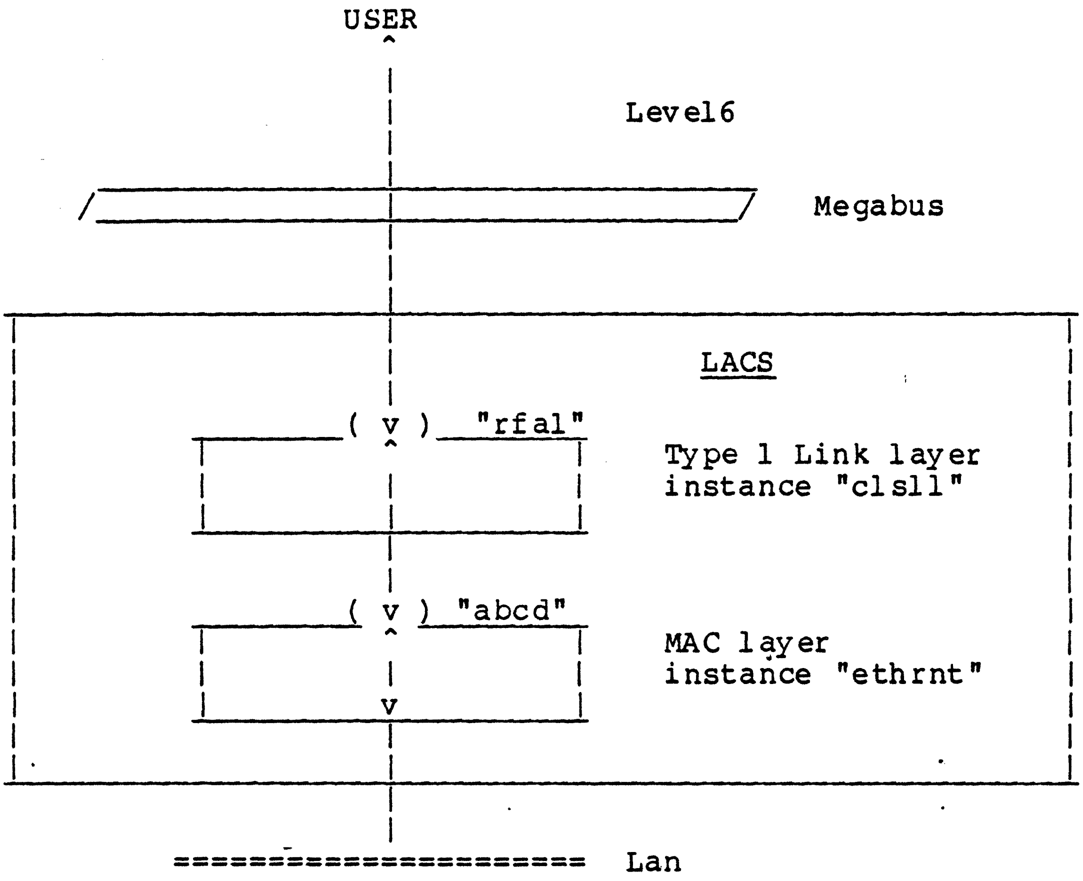
USER
Level6
Megabus
LACS
"rfal"
Type 1 Link layer
instance "clsll"
"abcd"
MAC layer
instance "ethrnt"
Lan

Figure 2.1

2.11 ERROR MESSAGES

To be supplied.

3.0 INTERNAL SPECIFICATION

3.1 OVERVIEW

3.1.1 System Initialization (CLM Processing)

The bound unit ZQLCLM is comprised of 3 modules; a root module and 2 floating overlay modules. The root module, as dictated by CLM extension module conventions, is a prescribed sequence of constants, displacements and pointers that is used by the central CLM function to process directives. When the root of this BU is loaded, it is linked together with other root modules of CLM extensions. Included in the root is a directory of the LAN CLM directive names that this BU is responsible for processing. While in the 1st phase of CLM processing, the 1st overlay will be loaded to process all LAN CLM directives. Each directive type will have its own processing routine which will be invoked each time its respective directive is encountered in the CLM file. During the 2nd phase, the 2nd overlay is loaded. Each directive type will have its own processing routine in this overlay as well. However, during the 2nd phase, each of the processing routines are executed only once.

3.1.2 Configuration File Processing

Processing of the Configuration File (C-F), which as described earlier, is a post-CLM function, will be performed by a module which will be incorporated into the LAN Manager BU (ZQLMGR). Its invocation will be triggered by the 1st Activate SAP MCL call issued to the LAN Facility.
A sequential file named "LANC-F" will be OPENed and read record by record. Each record will represent a directive. Directive names will be determined and the respective routine for processing that directive will be invoked. As a result of this processing, additional LAN data structures will be constructed and existing ones updated. The C-F will be required to be established by a system network administrator. It can be built using either MOD400's line editor or screen editor facilities, and it will be required to be named "LANC-F". It must reside under the >>SID directory.

3.2 SUBCOMPONENT DESCRIPTION

3.2.1 System Initialization Subcomponent

The following 3 sections provide a description of the structure and processing logic in each of the 3 modules that comprise the LAN CLM bound unit. A 4th section describes the data structures generated and initialized by the LAN CLM function.

3.2.1.1 Root Module

The root module contains no executable code. It is simply, as previously mentioned in Section 3.1, a sequence of constants, displacements and pointers structured according to prescribed conventions. The central CLM function works within the structure to locate a match between a directive name read from the CLM_USER file and one of the directive names contained in this root directory. When a match is found, floating overlay module #1 will be called to provide the processing for the respective directive. The structure and its elements comprising this root module are described in Ref. 1. This description is also provided here for purposes of convenience.

ELEMENT	DESCRIPTION
1 (3 words)	Name given to the LAN directive group (in ASCII)
2 (2words)	Represents 1st and 2nd characters of decade year (in ASCII)
3 (5 words)	Month, day, time (in ASCII)
4 (1 word)	Revision number
5 (1 word)	Displacement (bytes) to Device Descriptor Table
6 (2 words)	Pointer to work area for 1st and 2nd overlays
7 (1 word)	Displacement (bytes) within the 1st overlay to an initializing routine which will scan the System Device Table to detect presence of Lan Controllers and provide required processing
8 (2 words)	Size (in bytes) of interpretive overlay
9 (2 words)	Size (in bytes) of cleanup overlay
10 (1 word)	Control word

11 (2 words) Link field; this field is present in each root module for each CLM command group; points to the next CLM root directory that has been loaded.

12 (variable) Made up of occurrences of command blocks, one for each LAN directive type processed by this CLM BU

13 (1 word) End of structure sentinel

14 (variable) Work area

3.2.1.2 1st Phase Module

This module will be linked in the ZQLCLM BU as the 1st of 2 floating overlays. It will consist of the following functions:

Initializing Routine

This routine provides an initializing function for the LAN command group. It will be executed once when the BU root is initially loaded. It will access the System Device Table of active devices configured on the megabus which was previously constructed by the central CLM function. A scan of the directory is made to locate entries with LACS device ID's. The input device ID returned by an active LAN Controller will be '2Ehh' where the 8 least significant bits of the ID word will specify adapter type and sub-channel number (see Section 3.4 of Ref. 2). Because of the position of controller address and adapter number in the 10 bit channel number, and the method in which combinations of channel numbers are generated, a total of 32 "2E" entries will exist in the Device Table for each active LAN Controller configured on the megabus. Possible error conditions that may be detected as a result of scanning the Device Table are:

- No LAN Controller is configured;
- No adapter is configured on the controller;
- Adapter type is not CSMA/CD;
- Adapter QLT's have not executed properly.

Any of the above error conditions will result in setting a "reject" flag in the work area. Processing of this directive will be terminated and subsequent directives will be rejected. No LAN structures will be generated and LAN functionality will not be invoked.

For each controller configured, the following information is recorded:
- megabus address of controller,
- count of active adapters,
- adapter position(s) on the controller.

Layer Instance Directive Routine

A scan of the parameters supplied in the directive will be performed. If parameter format is correct, the following processing sequence is executed:

verify correct directive sequence, i.e., insure that this is the 1st LAN directive, or that previous directive was an LUSR directive;

verify that controller specified in ct_mask parameter matches a controller configured on the megabus;

calculate weight represented by this controller and its adapter(s) and call CLM function #22;

obtain block of temporary memory to store off name of layer instance and interrupt level.

LUSR Directive Routine

A scan of the parameter values supplied with this directive will be performed. If the parameter values are appropriate, the following sequence is executed:

verify correct directive sequence, i.e., insure that the previous directive was either a layer instance directive or an LUSR directive;

obtain a temporary memory block; link this block to the layer instance structure it belongs to, or to the previous LUSR structure; add 1 to LUSR count;

store LUSR name and request task level;

call CLM function 17 to reserve free lrn.

3.2.1.3 2nd Phase Module

This module is linked in the ZQLCLM BU as the 2nd of the 2 floating overlays. It is loaded and invoked following the completion of Phase 1 and CLM's load balancing processing.

Layer Instance Directive Routine

Working with the temporary structures built by Phase 1 processing, this routine proceeds to execute the following steps:

Get permanent memory block for 1 Controller Directory (CD) structure and 'n' Controller Table (CT) structures where 'n' equals the number of LACS Controllers configured. Initialize these structures and link them together.

Get a permanent memory block for 'n' Physical Line structures to be incorporated in a Physical Line Directory (PLD), where 'n' equals the number of adapters configured. Initialize these structures and link them backward to their respective CT structure.

For each controller configured, get a permanent memory block for 'n' Layer Table (LT) structures where 'n' equals the number of different protocol layer types configured to reside in this controller. Initialize these structures and link them with their superior CT structure.

> For each LT configured, get a permanent memory block for 'n' Layer Instance Table (LIT) structures where 'n' equals the number of layer instances configured within this protocol layer type. Initialize these structures and link them with their superior LT structure.
>
> > For each LIT configured, perform the following:
> >
> > > Get CPU # that this controller is assigned to by invoking CLM Function #22 with weight parameter set to zero;
> > >
> > > Determine if interrupt task level specified for this LIT is a unique value for this processor; if so, spawn an interrupt handler task at the level specified for the processor no. specified.
> > > If interrupt task level has already been installed, take no action.

LUSR Directive Routine

Again, working with the temporary structures built by Phase 1 processing, this routine proceeds to execute the following:

Get a permanent memory block for 'n' local user structures where 'n' equals the total number of local users specified in this configuration. This collection of structures will be referred to as the User Directory (UD). Link the UD to the CT.

For each LUSR directive processed during Phase 1, CLM Function #11 is called to build an RCT structure, and if the request level is a unique value for the processor, a TCB will also be generated by CLM. The RCT and the TCB structures will be initialized on return. For 1st release, the request task TCB will be set up to execute on the processor to which the corresponding LACS controller has been assigned. In other words, the request task that is created is for a particular user who is interfacing with a layer instance. That layer instance is configured to execute in a particular controller, and that controller has been assigned to a processor. It is this processor that the request task is set up to execute on. An LUSR structure within the UD is also initialized at this time.

3.2.2 Configuration File Processing Subcomponent

The function of processing C-F directives will be provided by the module described below. This module is linked as part ot the ZQLMGR BU. ZQLMGR is the L6 resident portion of the LAN System Management function.

3.2.2.1 C-F Processing Module

The processing logic of this module imposes certain requirements on the C-F relating to its characteristics. It must be a sequential file named "LANC-F". It must have been generated by MOD400's Line Editor or Screen Editor facilities; and it must reside in a specific directory - assume >>SID for now.

This file is processed using File Management and Data Management services. File Management calls are limited to $GTFIL (get-file), $OPFIL (open-file) and $CLFIL (close-file). The only Data Management call used is $RDREC (read-record).

The C-F will initially be read in its entirety in order to make a 1st pass over all the directives in the file. As a result of this 1st pass, a series of temporary data structures will be constructed. Following the necessary File Management calls to get and open the file, successive directives will be accessed by issuing $RDREC MCL's. A large memory block will be obtained in which to construct the temporary data structures.

For each layer instance directive encountered, a temporary controller block structure is allocated. Each time a new controller mask parameter is recognized in an li directive, a new controller block structure is allocated.
For each new type of li directive, a layer instance array block is allocated (layer type pointer in controller block points to li array block.
For each unique layer instance number, an li structure is allocated. The address to this li structure is stored in the li array block.
For each local SAP directive belonging to a layer instance, a local SAP structure is allocated and chained off of its li structure previously constructed. The parameters specified in the local SAP directive are stored in the local SAP structure.
Processing for remote SAP directives is practically the same as that described above for local SAP's.

In this way, when end of file is reached, a heirarchy of linked structures will have been constructed in a temporary storage area. An important result of this 1st pass is that counts of relevant entities will have been accumulated, e.g., local SAP's, remote SAP's and layer instances.