# NETWORK COMMAND PROCESSING SYSTEM OVERVIEW

Yonwoo Nam
Senior Engineer
Bendix Field Engineering Corporation
Columbia, Maryland 21045

Lisa D. Murphy
Senior Scientific Systems Programmer
Bendix Field Engineering Corporation
Columbia, Maryland 21045

## ABSTRACT

The Network Command Processing System (NCPS) developed for the National Aeronautics and Space Administration (NASA) Ground Network (GN) stations by the Bendix Field Engineering Corporation (BFEC) is a spacecraft command system utilizing a Multibus/68030 microprocessor. This system was developed and implemented at ground stations worldwide to provide a Project Operations Control Center (POCC) with command capability for support of spacecraft operations such as the LANDSAT, Tracking and Data Relay Satellite and Nimbus-7 spacecraft. The NCPS interacts with the POCC and a local operator to process configuration requests, generate modulated uplink sequences, and inform users of the ground command link status. This paper, which provides an NCPS description, consists of two sections. The first section presents the system functional description and the hardware description. The second section presents software design considerations including the implementation of a flexible and expandable operator interface, maximization of reusable software modules, and validation of BFEC designed development tools.

## BACKGROUND

In 1987, BFEC was directed by the NASA to design a replacement spacecraft command system for the early 1970's vintage Spacecraft Command Encoder (SCE). The project goal was to devise a system that would utilize state-of-the-art hardware and software, provide on-site fault isolation and eliminate cumbersome analog alignment requirements. The initial design study determined that there would be a significant cost benefit in modeling the system after the NASA Telemetry and Communications Data System (TCDS). The TCDS is a telemetry data processing system developed in the mid 1980s which utilized distributed processing techniques and was the replacement for outdated decommutators and data system computers on the NASA GN. Due to the differences in functionality between the TCDS and NCPS, only portions of the hardware components could be re-used. However, the basic hardware system architecture and the device driver software for these components could be implemented with minimal modification. The greatest benefit was in the software design since much of the software developed during the TCDS project could be incorporated into the NCPS thus cutting the software design time and the overall project costs.

## SYSTEM FUNCTIONAL DESCRIPTION

The NCPS provides the POCC and/or a local operator with the capability to command a spacecraft and to monitor the command system status. Figure 1 is a simplified functional block diagram of the system. There are three command modes: throughput, local, and hardline. Each mode has a different source of data. In throughput mode, data is received for uplink via the NASA Communication (NASCOM) Data Link in 4800 bit blocks. In local mode, data prestored in command pools is used. Each command pool may contain several spacecraft commands that are between 16 to 2000 bits in length. In hardline mode, serial command data is received via a connector on the rear of the NCPS and sent directly to the uplinking hardware with little intervention from the software.

The NCPS operator uses a menu interface to select one of the three commanding modes. Once a commanding mode has been selected, the operator enters a spacecraft identification code (SID) and a file designator character code. These two codes are used to retrieve the selected spacecraft attribute file from the hard disk which contains default commanding parameters. The NCPS loads this file into its database memory and enters a prepass state. The current configuration attributes, status information and menu selection items for modifying spacecraft attributes, is shown on the video terminal. Each commanding mode has two operating states: prepass and pass. Prepass is the state of operation prior to uplink. In this state, the operator can modify and override certain spacecraft attributes. Pass is the state of operation where data is uplinked to the spacecraft and is entered via an operator menu selection when the system is in the prepass state. In the pass state, some attributes that do not affect command uplink can be modified, including selecting and uplinking an idle data sequence between commands.

In the throughput commanding mode, three options are available: uplink immediate, buffered uplink, and rate adjust. The uplink immediate option permits the command data to be transmitted as soon as it is received. The buffered uplink option accumulates one command data block before beginning command transmission to the spacecraft. Buffering allows for compensation of irregularities in the data arrival time and a precisely metered continuous data flow to be generated. When the received average data rate is different than the precise uplink command data rate, a circumstance may arise in which the buffered data may be consumed or an excessive amount may accumulate. To diminish this possibility, a rate adjustment feature has been incorporated in which the uplink command rate is varied to match the average command rate received from the commanding source. If the number of blocks begins to decline, the software retards the uplink rate. If the number of blocks drops to zero, the software notifies the operator that an underflow has occurred. The software will also advance the uplink rate when the number of buffered blocks grows towards a predetermined maximum. The software notifies the operator that an overflow has occurred if the number of buffered blocks exceeds the predetermined maximum.

In the local commanding mode, the operator selects commands to be uplinked from a prestored command file by specifying a command “mark” number. This command will be uplinked according to the parameters in the selected spacecraft attribute file upon release by the NCPS local operator.

In the hardline commanding mode, the operator configures the NCPS for the selected spacecraft and enters the pass state. Any data received through the hardline connection is uplinked according to the parameters in the selected spacecraft attribute file. This mode is used at locations where serial command data can be received directly from a POCC command generator. This mode is primarily used for testing a spacecraft shortly before launch.

Regardless of the command mode selected, the output of the NCPS is a Phase Shift Keyed (PSK) modulated subcarrier which is routed to the transmitting system exciter. The subcarrier frequency and the data rate used to modulate the subcarrier are determined by the selected spacecraft attribute file.

The NCPS system status is reported using Site Status Messages (SSM) and command echo NASCOM blocks. Options to enable or disable both the SSM and echo block functions are available from the operator menu.

SSM blocks are generated and transmitted to the project specified by the status destination code at the rate of one block per second in both the prepass and pass state. SSMs contain information about the status of the NCPS including information about the site, identification of the spacecraft being commanded and the status of the uplink process.

Echo blocks contain the data that was uplinked and are transmitted to the project specified by the echo destination code. There are two types of echo blocks: asynchronous and synchronous. Asynchronous blocks contain the image of the uplink data as received from the verification receiver which samples the uplink RF output. Synchronous blocks are the NASCOM blocks that were received from a project using the source and destination codes interchanged and transmitted back to the originating project.

## HARDWARE DESCRIPTION

The NCPS is housed in a single 19" standard NASA equipment rack. The units mounted in the rack include a 140 MByte hard disk drive, a 5.25" floppy disk drive, a streaming tape unit, a 14" color graphics terminal, a standard 101 keyboard and the NCPS chassis. The functional and ergonomic layout provides the operators with easy access to the system. Figure 2 illustrates the NCPS rack elevation layout.

## CHASSIS

The NCPS chassis uses a multibus 1 architecture with a 20 slot card cage to accommodate the five PC cards and to provide for expansion or modification. Of the five board assemblies, one is a commercially available computer board and four are special purpose custom designed boards i.e. Serial Time Code Receiver board, Transmit/Receive board, External 5 MHZ board and a PSK Modulation Board (PMB).

## COMPUTER BOARD

The computer board utilizes a 68030 32-bit microprocessor with 4 megabytes of RAM. This board performs the central control function of the NCPS. The system initialization instructions are stored in EPROM. The computer board also provides interfaces the disks, streamer tape, graphic video terminal, and the external printer.

## SERIAL TIME CODE RECEIVER BOARD

The Serial Time Code Receiver board was developed by the NASA. The board decodes the received serial binary 1 (SB-1) time code which is a Manchester encoded RS-422 signal into parallel time with millisecond accuracy which the NCPS software inserts into the transmitted NASCOM 4800 bit block. The source of the SB-1 time code is the station master timing system.

## TRANSMIT/RECEIVE BOARD

The Transmit/Receive (T/R) board was developed by BFEC for the TCDS and is used to interface with the NASCOM communication system. It provides the channel for processing a digital serial data stream of NASCOM blocks entering or leaving the NCPS.

## EXTERNAL 5 MHZ BOARD

The External 5 MHZ board provides frequency synthesis by using the phase locked loop (PLL) technique. The purpose of the board is to generate a phase coherent 4.096 MHZ signal from the station's 5 MHZ signal. The monolithic PLL was used for fractional frequency synthesizer in External 5MHz board and consists of a Voltage Control Oscillator (VCO), phase comparator, and low pass filter. The monolithic PLLs were used in this application because of their low cost versus high performance at frequencies below 50 MHZ.

A block diagram representation of the fractional frequency synthesizer is shown in Figure 3. The phase locked loop operates by producing an oscillator frequency to match the frequency of an input signal. In this locked condition any slight change in input frequency, $f_a$, first appears as a change in phase between $f_a$ and the oscillator frequency, $f_c$. The phase shift then acts as an error signal to change the oscillator's frequency to match the $f_a$.

Using a crystal oscillator as the frequency source for the PLL and having it phase locked to the station's precise main timing system results in a long term stable clock. This procedure was incorporated in the hardware design to increase the stability of the modulated subcarrier.

## PSK MODULATION BOARD

The PMB is designed to provide command support for all subcarrier modulated compatible spacecraft. Control, setup, and ground command verification sequences are received via the multibus. The command data to be uplinked is received from the CPU via the multibus in all modes with the exception that in hardline mode it is received via a direct serial interface.

The PMB is divided into three functional areas: Command Data Control (CDC), Subcarrier Modulation/Demodulation (SMD), and the Serial Data Interface (SDI). Figure 4 illustrates the PMB, Functional Block Diagram.

The CDC interface permits control and setup of the PMB by the CPU via the multibus interface. The PMB setup/control words, provided from the hard disk's spacecraft attribute files, select the subcarrier frequency, data (modulation) rate, data type encoding, command idle, modulating source, and command mode.

The SMD process generates a composite modulated PSK waveform and utilizes a stable frequency source, rate multipliers, a sinusoidal look-up table, a digital-to-analog converter, and a single pole lowpass filter. The subcarrier rate multiplier along with the frequency reference, which can be from an onboard crystal or the External 5 MHZ board, generates subcarrier samples at 256 times the selected subcarrier frequency. The subcarrier samples are the result of a phase counter addressing a sinusoidal look-up table, that is PROM. The PROM contents are specified by the equation

$$D_i * \sin(2 * pi * K/256)$$

where K represents the address of the subcarrier phase counter and $D_i$ represents the sign of the data sequence. The active single pole low-pass filter eliminates the out of band harmonic power.

The SDI interface is provided to permit the processing of a serial command sequence. When the serial data mode is selected on the PMB, the transition tracking loop is selected versus the reference 4.096 MHZ. The transition tracking loop drives the subcarrier phase to provide proper alignment of the subcarrier transitions and the data symbols being transmitted.

## PERIPHERAL

An external serial line printer provides a hard copy of all NCPS activities and status information and is used primarily for historical data and as a troubleshooting aid.

## SOFTWARE DESCRIPTION

The NCPS software development task also began in 1987. The original software project included software to support a wide range of ground stations and spacecraft. Several ground stations were closing and some spacecraft were becoming old and obsolete. However, the NCPS was required to support both the aging spacecraft, as well as, future spacecraft. A design approach to maintain an everchanging system was needed. The Systems Utilization Enhancement (SUE) data system project, now known as TCDS, was also faced with application software that required modification on a continuing basis. In order to optimize the generation and maintenance of those applications, various tools were developed as part of the TCDS project. A distributed operating system and a multi-tasking executive (MX) were developed to support TCDS. A report “Distributed Operating System for NASA Ground Station” written by John Doyle and presented at the 1987 International Telemetering Conference (ITC) describes the TCDS software and provides background for this paper.

## NCPS SOFTWARE DESIGN GOAL AND CONSIDERATIONS

The NCPS software design goal was to utilize as much of TCDS software and tools as possible without inhibiting the development and uniqueness of the NCPS application. TCDS Software design objectives described in the following paragraphs were considered during the design phase.

"Selection of an appropriate language and development environment". The C language, the Unix development environment and MX were used in TCDS were found suitable for the NCPS. This development environment was incorporated into the NCPS design objectives.

"A plan for future growth in program size". TCDS uses a modular approach to software development where functions can be added and deleted without disrupting the entire system. This plan was included in the NCPS.

"Allow additional new subsystems and/or devices". The TCDS comprises eight varieties of subsystems which can be duplicated to meet the support requirements of a specific ground station. The subsystems are linked to four Ethernet 10 Mbit per second busses. The largest system has a total of thirty subsystems. This objective was incorporated into NCPS in order to add hardware and device driver code as needed. It was also foreseen that the NCPS could become a subsystem of the TCDS adding the issue of compatibility between the two systems to the design task.

As NCPS hardware components are improved and upgraded, it is necessary to upgrade software. An objective to "minimize dependence on hardware configuration" was included.

"Automation of system documentation through the use of graphical representation". This objective was included because of the continuous software updates and documentation changes for the NCPS.

"Automation of configuration control". This objective was included in the NCPS to have consistency in project software for future upgrades and releases.

"Utilize an organized system database". A memory resident database containing global information allowed for instant access to various system variables including data needed for displays and the operator interface.

"Base design on reusable modules". This objective was incorporated into the NCPS in order to reduce the duration and costs of software development.

"Allow incremental source code revision". This objective was included in the NCPS to allow for software upgrades and enhancements.

It was determined by the design team that its goals could be met effectively by using the TCDS as a foundation for building the NCPS software. Many of the TCDS objectives were applicable to the NCPS. This proved to be advantageous in that operator interface software, BFEC developed tools and some application software could be reused.

## OPERATOR INTERFACE SOFTWARE

Operator interface software known as opware was used for the NCPS Graphical User Interface. The term opware evolved during the development of the TCDS. It is "any implementation of a design, or solution to a problem, that is complete immediately upon its specification. It differs from software in that there is no compilation or linkage required"[1].

Opware for the NCPS includes batch files of operator menus, prompts and sequenced commands, tables of data in the system's memory resident database, ASCII files for specifying display formats and an interpreted block processing language. This facilitates an operator interface that can be custom designed on the fly. No recompilation of code is necessary. The opware uses the system database in order to instantly access system data. This allows for the screen to be refreshed once a second with up-to-date, real-time, information. For example, during a spacecraft pass, an operator can quickly choose whether or not to echo uplink information back to the project by turning echo off or on. A variable specifically defined to monitor the state of echo blocks resides in the system database. It is set to one when echo is turned on and a zero when it is off. The opware has direct access to the variable in the database. When the operator selects the option from the operator menu to turn echo on, the opware command file driving that menu sets the variable to one. A "C" function to build echo blocks samples the echo state variable. When the value of the state variable becomes a one, the code begins building the block. The display on the video terminal will also be update, instantly showing that echo has been turned on. A counter variable of the number of echo blocks built and transmitted is kept in the data base. It is updated and displayed on the video terminal once a second.

## BFEC DEVELOPED TOOLS

An engineering tool developed for the TCDS was incorporated into the NCPS. This tool is the Network Adaptive Schema for Modeling Asynchronous Computation (NASMAC) which allows the specification of software systems using directed graphs, and the automatic transformation of such graphs into operational software. NASMAC was developed for the purpose of minimizing software life cycle costs. It helped to achieve all of the objectives stated in the previous paragraphs.

NASMAC has a "computer-based, graphic network diagram to specify task and subsystem interfaces" that is used to build system directed graphs[1]. Each node on a the graph depicts either memory resident data objects or functions. Data objects are expressed as rectangles and functions are expressed as any other shape. Nodes are connected by data and control directives. Data directives are depicted as dotted lines and control directives are depicted as solid lines. The finished graph is the design of the application and the sequence of events that occur in the application. Figure 5 is an NCPS graphs used to build echo blocks. The graphs of the NCPS are compiled providing base line software that manages task scheduling and data flow control. The entire NCPS system is depicted in these NASMAC graphs providing current documentation and configuration control for the system. The modularity resulting from the use of NASMAC graphs allows for application software to be continuously modified and upgraded.

## REUSABLE MODULES

TCDS provided NCPS with reusable software modules, opware and engineering tools which facilitated approximately 80% of the NCPS software with only approximately 15% of that software requiring modifications. Table 1 is a table of source lines of code (SLOC) for the NCPS. It depicts the total lines of code taken from the TCDS including opware and tools plus the total lines of code developed for the NCPS. It also depicts the number of lines of code taken from TCDS which needed some modifications. Four to six months of startup development time was saved by using this design approach. The following is a list of reusable software modules from TCDS:

- Transmit/Receiver board - driver software was used to communicate to the NDL.
- Alarming functions which provide the operator with information about the state of the system.
- Operator Interface Software for an interpreted block processing language.
- Printer and Terminal output software.
- Hard disk file system utilities separate from the UNIX file system.
- Real Time Executive (MX).
- GMT timing board driver software.

## NCPS APPLICATION SOFTWARE

By using the TCDS as a foundation for the NCPS, the NCPS software needed to only add an application layer. Focus could then be turned to the NCPS functionality and the software that would be written and integrated into the NCPS design. Approximately 20% percent of software was newly written for the NCPS specific application.

Software to support commanding was written. It included software to build SSMs and echo blocks, software to verify incoming blocks, and software to load command pools.

Utility software was developed to maintain, display, and print the following files: attribute, command pool and test command.

Software was written to support PMB hardware drivers including mechanisms for command buffering and rate adjust.

Software to configure and monitor the system's state was constructed.

Floppy Disk and Tape Device utilities that can be executed from the real-time system were created. These utilities will become part of the TCDS reusable software pool for future projects.

## SUMMARY

After detailed research and analysis, the NCPS was developed and implemented to provide the GN sites with command capability for support of spacecraft operations. The modularization and commonality of parts have helped produce a system that can be expanded as needed. By using TCDS software as a foundation for that of the NCPS, additional software modules can be added to the NCPS as requirements are added or changed. The software tools and opware developed for TCDS also allows for a lesser experienced programmer to maintain and upgrade the system.

## ACKNOWLEDGEMENT

The hardware and software, with the exception of NASMAC described herein were developed on NASA contract NAS5-31000 for the Goddard Space Flight Center's Networks Division.

## REFERENCES

[1] "Distributed Operating System for NASA Ground Stations", Doyle, John F., International Telemetering Conference 1987, San Diego, CA Volume XXIII, p425.

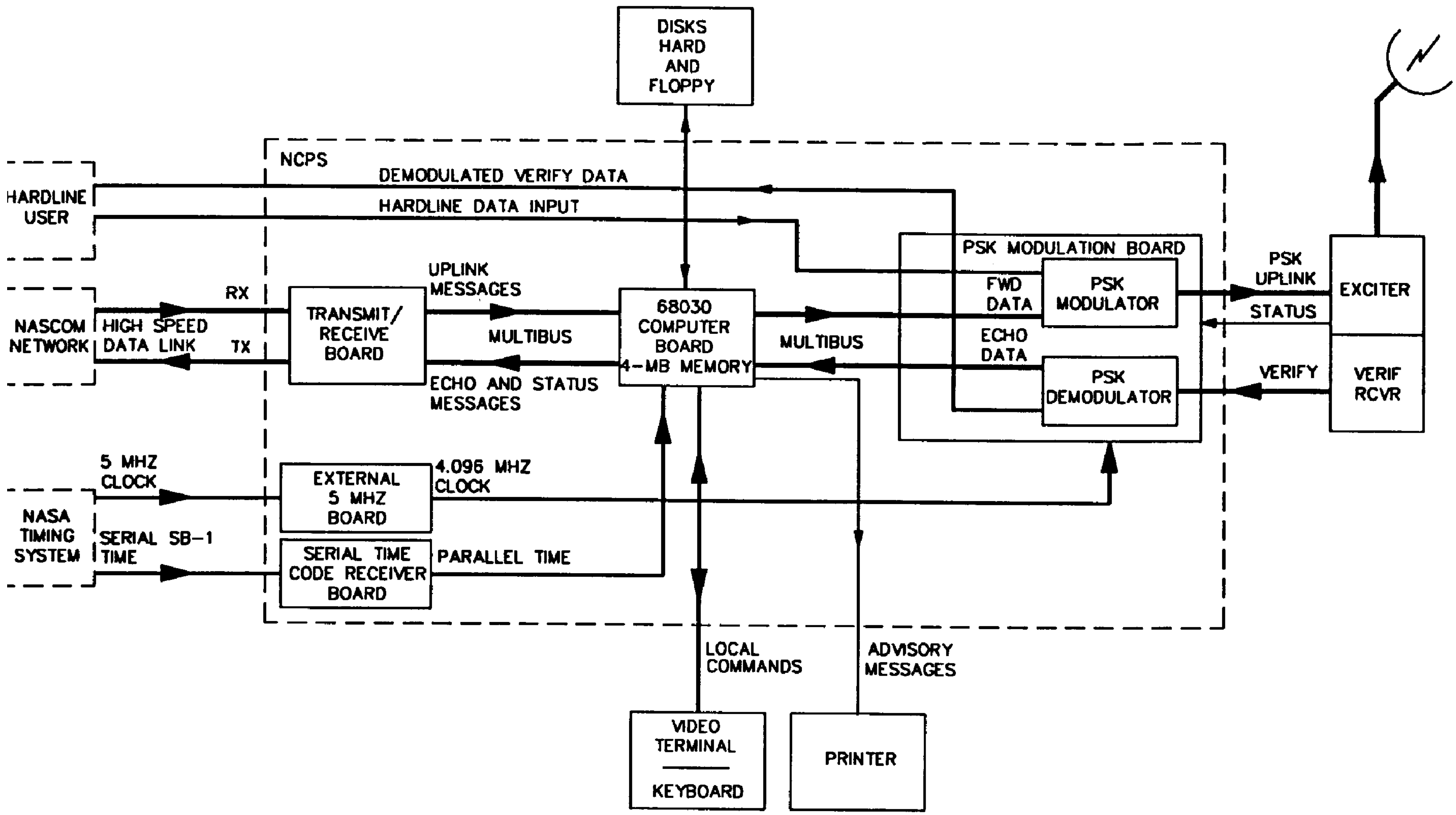


FIGURE 1. NCPS FUNCTIONAL BLOCK DIAGRAM

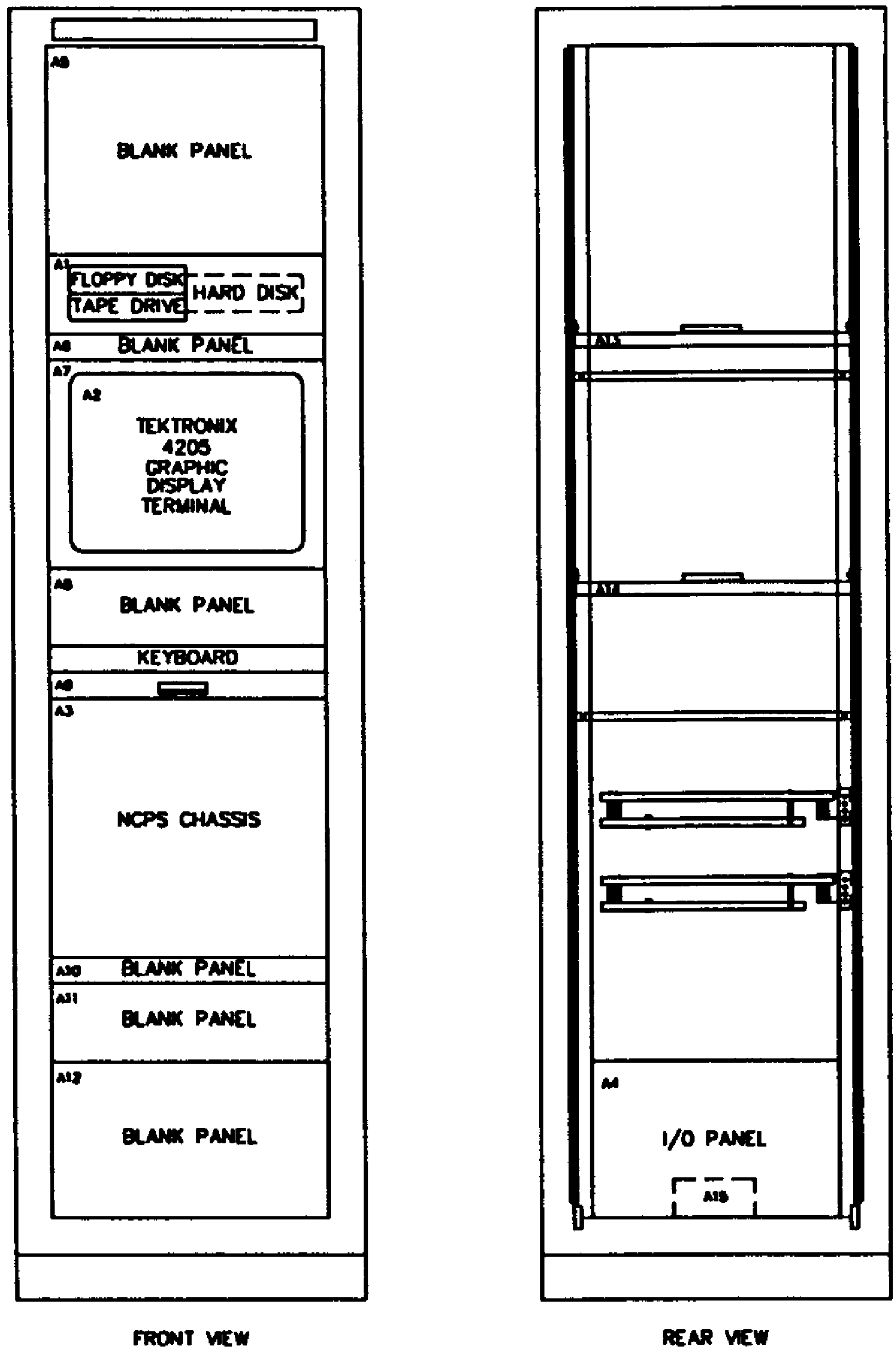


FIGURE 2. NCPS RACK ELEVATION

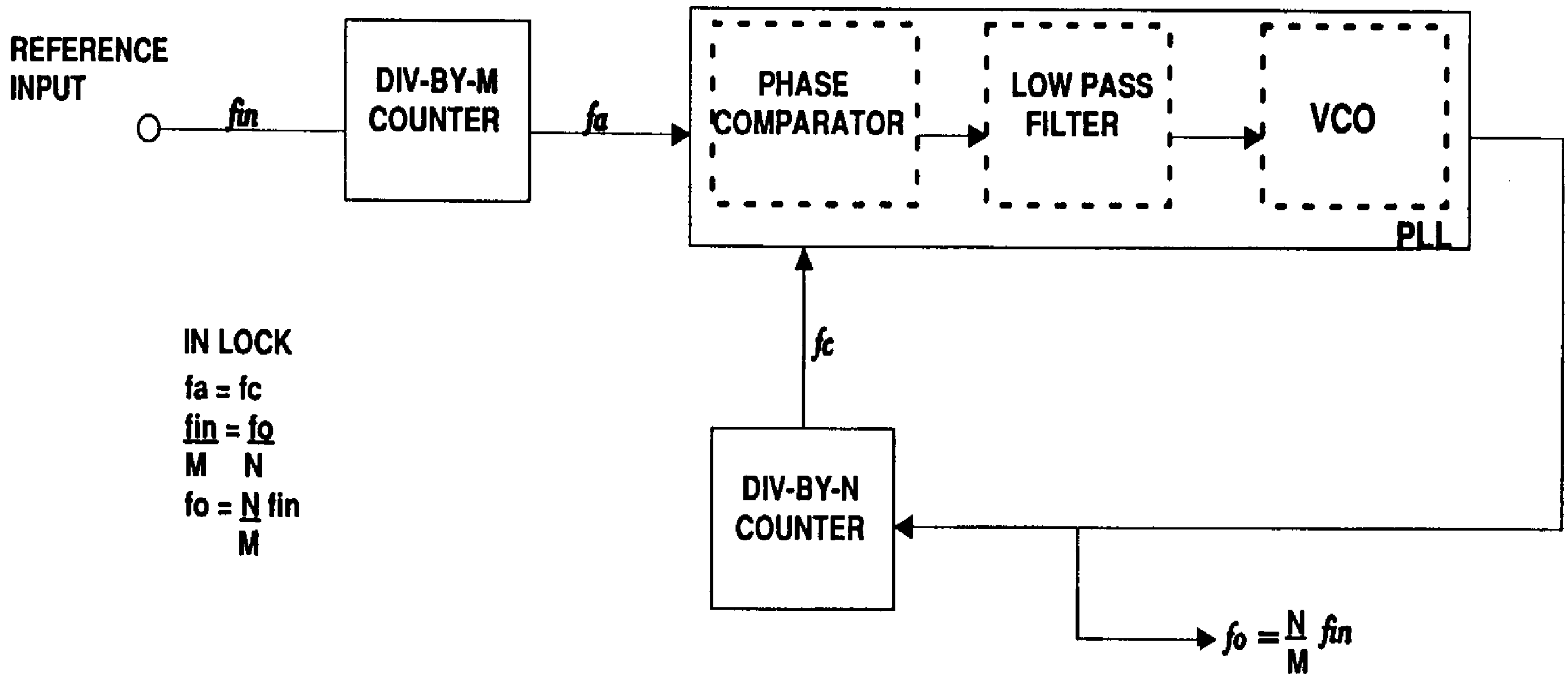


FIGURE 3. FRACTIONAL FREQUENCY SYNTHESIZER

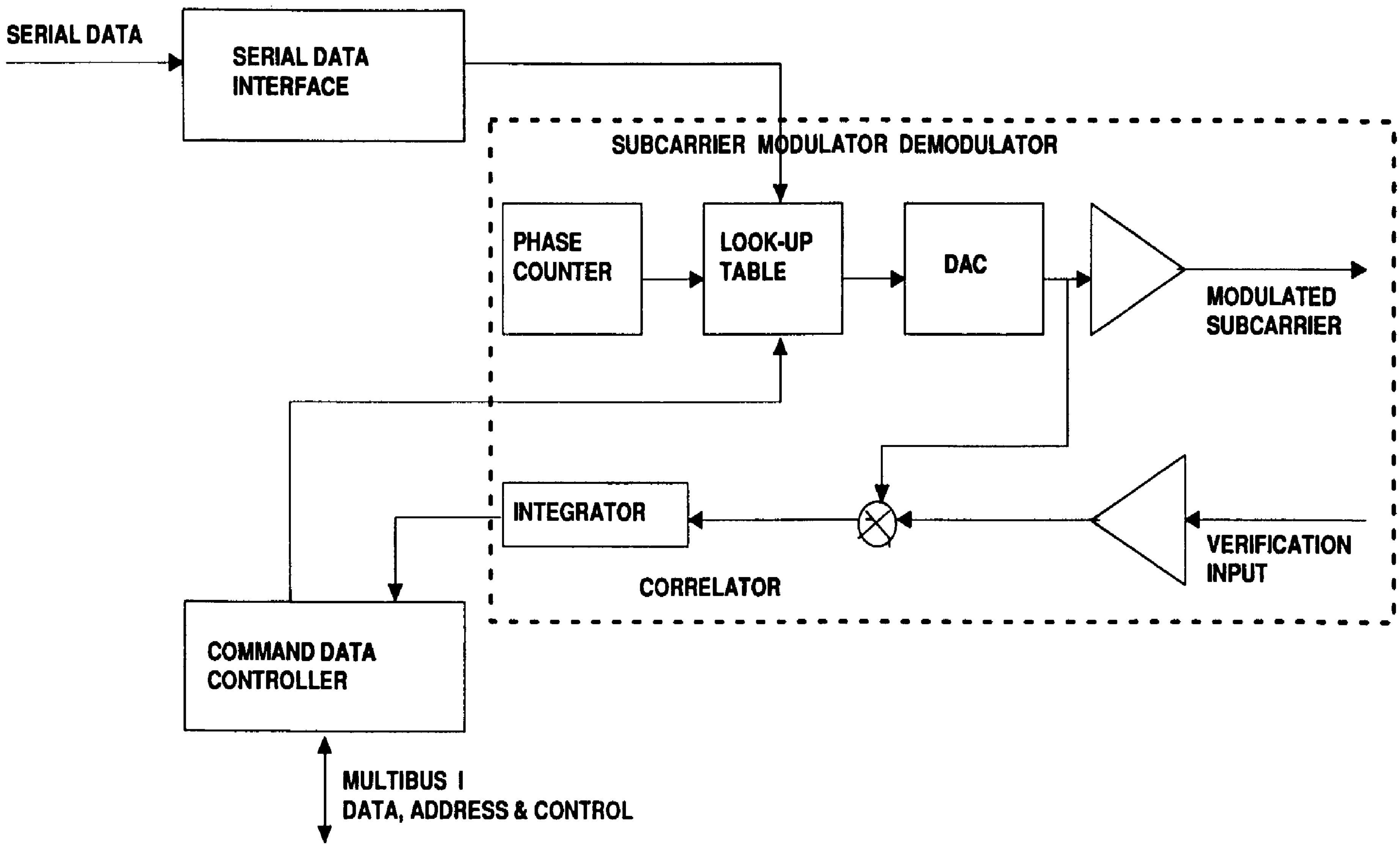


FIGURE 4. PMB FUNCTIONAL BLOCK DIAGRAM

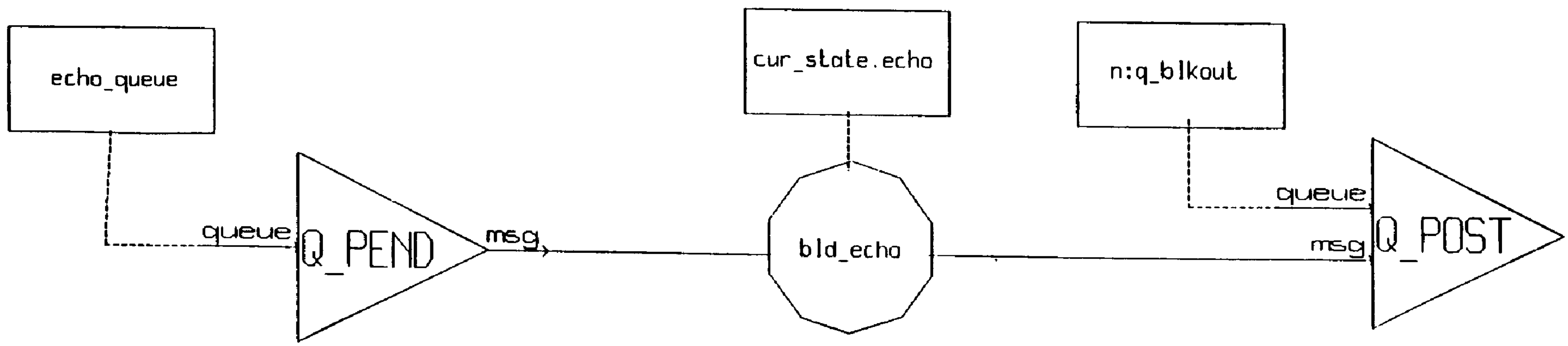


**FIGURE 5. NASMAC GRAPH FOR NCPS ECHO BLOCKS**

| | Total SLOC | Modified SLOC | % Effort | NCPS SLOC | % Effort |
|---|---|---|---|---|---|
| C and Assembly Code | 69193 | 10242 | 14.80 | 14232 | 20.57 |
| Include Files (.h) | 8204 | 398 | 4.85 | 5301 | 64.61 |
| MX Real Time Executive | 12777 | 3452 | 27.02 | 534 | 4.18 |
| Operator Interface Cmds | 5372 | 432 | 8.04 | 4865 | 90.56 |
| NASMAC + Opware Tools | 30626 | 367 | 1.20 | 547 | 1.79 |
| Total | 126172 | 14891 | | 25479 | |

| | |
|---|---|
| % of Reusable Software and Tools: | 79.81 |
| % of SLOC Specifically for NCPS: | 20.19 |
| % of Reusable Software Modified for NCPS: | 14.79 |

**TABLE 1. NCPS REUSABLE SOFTWARE MODULES**