

ISDN

ISDN Basic Rate Interface Software for the HPC16400E High Performance Data Communications Microcontroller



Literature Number: SNOS397A

ISDN Basic Rate Interface Software for the HPC16400E High Performance Data Communications Microcontroller

General Description

The ISDN Basic Rate Interface Software Package implemented on the National Semiconductor HPC™ Microcontroller Family contains the software elements that are necessary to implement CCITT standards Q.921 and Q.931 as approved by ANSI committees for North America.

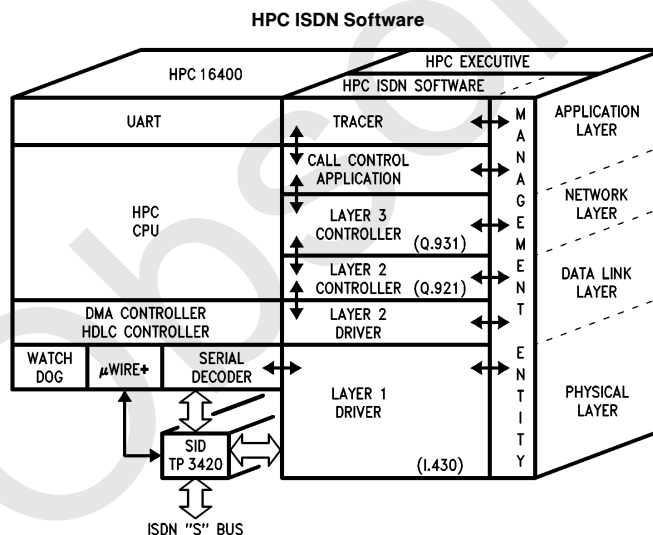
The software package is designed to be easily unbundled and used independently by a software developer. Each layer or function is written as a separate software task. This modular design and well defined task interface make it easy to interface application dependent software to the modules provided. The coding standards for software development have been designed to ensure development of consistent, structured code, which can be easily used and maintained over the life of the code.

This software is supplied as a disk set and is used in conjunction with HPC development tools and software.

Features

- Multi-tasking executive
- Preemptive scheduling
- Modular software design
- Multiple timer facility
- HPC physical layer I/O interface
- Layer 2 link access procedure for the D channel (LAPD)
- Layer 2 link access procedure for the B channel (LAPB)
- Layer 3 protocol control procedure for a terminal endpoint
- Layer management entity support
- Demonstration Call Control Task
- Task_View task exerciser and debugger
- Message trace capability
- Split frame message formatting
- Source code in C language

Block Diagram



TL/DD/10077-1

COMBO™, COMBO II™, HPC™ and MICROWIRE/PLUS™ are trademarks of National Semiconductor Corporation.

1.0 Architectural Description

1.1 INTRODUCTION

This description defines the software required to implement the ISDN Basic Rate Interface on the HPC family of micro-controllers, including the HPC16400 which has onboard hardware specifically designed for Data Communication and ISDN applications.

The software consists of the following main parts, shown in overview in Figure 1.1:

- HPC Executive, providing an operating environment and services for the ISDN software and for additional application software written by OEM users of the HPC.
- I/O Drivers, interfacing to the DMA/HDLC controllers on the HPC16400 and to the TP3420 "S" Interface Device.
- Data Link Layer Software, implementing the CCITT Q.921 and X.25 link access procedures (LAPD and LAPB).
- Network Layer Software, implementing the Protocol Control Procedures defined in the CCITT Q.931 standard.
- Demonstration Call Control Module, allowing a development engineer at a terminal to make and receive ISDN phone calls which exercise the above software.
- Tracer Module, allowing a development engineer at a terminal to monitor the operation of the above software.
- Management Entity Module

1.2 SOFTWARE ARCHITECTURAL PRINCIPLES

1.2.1 Modular Multitasking Environment

Each layer or function is written as a separate software task. Intertask communications and the interface between tasks

and I/O drivers is by means of mail messages and semaphores, which are managed by the multi-tasking HPC Executive.

This modular design and well defined intertask interface make it easy for users to interface application-dependent software to the modules provided. The services of the HPC Executive (mail, semaphores, timers, memory management) facilitate the writing of software tasks and I/O drivers. These services are available to all tasks and to interrupt-level drivers.

1.2.2 Event-Driven State-Machine Architecture

Telecommunication software typically involves many invocations of the same code (one per call, one per logical connection, etc.) and requires a particular software architecture: tasks must be structured as event-driven state machines. Each task has one or more mailboxes and operates by picking up mail, one message at a time, from the mail queue, and processing the message to completion before returning for the next mail message.

Each "entity" within a task (each call, each logical connection, etc.) has a state block, indicating its current state. After picking up a mail message, the task identifies the entity involved in the message, accesses the state block for that entity, processes the message based on the state of the entity, and finally sets the state block to the new state of the entity.

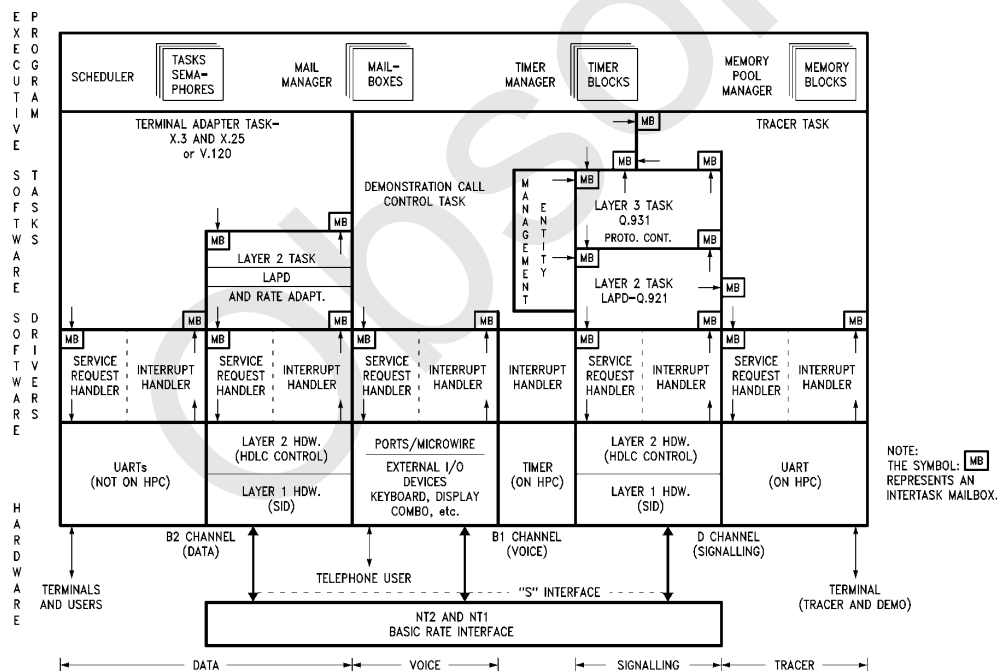


FIGURE 1.1 HPC16400 Software for ISDN

TL/DD/10077-2

1.0 Architectural Description (Continued)

1.2.3 Coding Standards

The coding standards for software development have been designed to ensure development of consistent, structured code, which can be easily used and easily maintained over the life of the code.

1.3 HPC EXECUTIVE

The HPC Executive provides an operating environment for the Layer 2 and Layer 3 tasks, the application tasks, the various support tasks, and the I/O drivers which interface to the hardware. It provides the following services to the tasks and I/O drivers:

- Scheduling of tasks that are ready to run, based on task priority. Preemptive scheduling and time slicing can be optionally enabled.
- Task-task and driver-task communication, by means of mail messages, which can be sent and picked up, and semaphores, which can be signaled and awaited.
- Timers, which are equivalent to mail messages with a specified delay and which allow tasks and drivers to time their activities and time out when an expected event does not occur.
- Memory management, to allocate and deallocate fixed-size buffers as needed by tasks and drivers.

Application tasks and I/O drivers developed by users of the HPC can easily be inserted in the HPC Executive environment and can take full advantage of its services.

1.4 ISDN TELECOMMUNICATIONS STANDARDS

1.4.1 CCITT Standards

The Layer 2 Task implements CCITT 1988 (Blue Book) specification Q.921 (LAPD) and Layer 2 (LA|B) of CCITT specification x.25.

The Layer 3 Task implements the Protocol Control Procedures of CCITT 1988 Specification Q.931.

The Layer 3 Task implements the circuit-switched procedures described in Section 5. The Layer 3 Task implements the Protocol Control procedures and some of the Resource Management. The Call Control Task implements a demonstration version of the Call Control Procedures and the balance of the Resource Management.

In terms of the specification and description language (SDL) diagrams in the Q.931 specification, the Layer 3 Task implements *Figure 38* (26 pages).

The establishment and release of logical links are fully covered in the Layer 2 specifications (Q.921), but the Layer 3 aspects of this are not handled in the version of Q.931 on which the Layer 3 Task is based. Therefore, additional Layer 3 states and SDL diagrams have been created and additional software has been written to handle this requirement.

1.5 ISDN TELECOMMUNICATIONS SOFTWARE

The software packages described below are designed to be easily "unbundled" and used independently by a software developer.

1.5.1 Layer 1 I/O Driver

The Layer 1 I/O Driver controls the HPC MICROWIRE/PLUSTM interface, and the onboard Serial Decoder. This driver is responsible for the hardware initialization, the control of the Serial Decoder, the activation and the deactivation of the Layer 1 I/O device. Use of the HPC Executive mail and semaphore services makes this driver simple to implement and easy to enhance by users that require additional Layer 1 hardware interfaces.

1.5.2 Layer 2 I/O Driver

The Layer 2 I/O Driver controls the HDLC/DMA controllers onboard the HPC16400, and interfaces this hardware to the Layer 2 Task. This driver is responsible for the hardware initialization, the reception of frames toward the HPC, the transmission of frames away from the HPC, and appropriate error handling. Use of the HPC Executive mail and semaphore services makes this driver simple to implement and easy to replace with alternative drivers that a user may wish to develop.

1.5.3 Layer 2 Task

The Layer 2 Task implements the full LAPD protocol defined in Q.921, providing error free in-sequence transmission, reception and multiplexing of messages received by an HDLC controller connected to the D signaling channel. The event-driven state machine architecture, described above, enables a single software module to support simultaneous activity on multiple logical connections. The Layer 2 Task also supports X.25 LAPB processing for messages received by a second HDLC controller connected to a bearer B channel.

1.5.4 Layer 3 Task

The Layer 3 Task implements the user side of the Protocol Control Procedures of Q.931, which are used to setup, answer, suspend, resume, and disconnect a call. Specifically, it implements all of *Figure 2/Q.931* of Q.931. The event-driven state machine architecture, described above, enables a single software module to support simultaneous activity relating to calls on both bearer B channels.

1.5.5 Demonstration Call Control Task

The latest versions of Q.931 separate the Layer 3 procedures into Protocol Control Procedures and Call Control Procedures. Call Control Procedures are application dependent. These procedures handle bearer channel selection and actual establishment of the voice channel. As Q.931 notes, these procedures can also be considered to be part of the Applications Layer. The Call Control Task implements a minimal subset of the Call Control Procedures, for demonstration purposes. In an actual application, this task will be replaced by an application-specific task, tailored to the capabilities of the actual terminal equipment (number of terminals, handsets, etc.).

1.5.6 Management Entity Task

The Management Entity Task, which is only generically defined in Q.921 and Q.931, handles housekeeping functions

1.0 Architectural Description (Continued)

for all layers. These functions include TEI negotiation with the network management entity, and the handling of unrecoverable errors. This task implements as much of the management entity as is currently defined and in addition whatever is necessary for the operation of the other tasks.

1.5.7 Tracer Task

The Tracer Task serves two purposes; to demonstrate the lower ISDN layers via a menu-driven telephone emulation mode, and to trace system mail message traffic.

1.5.8 Task_View Task Exerciser and Debugger

Task_View is a special-purpose task that can be inserted into the multi-tasking Executive environment in place of the Tracer Task. It reads and interprets a user supplied ASCII scenario file. Under control of this scenario file, Task_View sends mail messages to a specified mailbox (or mailboxes), where they are read by the task under test. Mail messages sent by the task under test in response to this input are then displayed by Task_View. In this way the task may be exercised and debugged.

2.0 Functional Description

2.1 INTRODUCTION

This description defines the functional requirements of the ISDN Basic Rate Interface Software Package implemented on the National Semiconductor HPC Microcontroller Family. Specifically, the HPC16400 Software Package implements or supports the following high-level functions:

- Multi-Tasking Executive
- HPC Physical Layer I/O Interface
- Layer 2 Link Access Procedure for the D Channel (LAPD)
- Layer 2 Link Access Procedure for the B Channel (LAPB)
- Layer 3 Protocol Control Procedure for a Terminal Endpoint

- Management Entity Support
- Call Control Demonstration Task
- Message Trace Capability

The HPC ISDN Software Package has been divided into several functional software elements, as illustrated in the HPC ISDN Functional Block Diagram, *Figure 2.1*. These functional elements correspond to software modules. The purpose of this section is to introduce the various software elements, to define their interactions, and to relate their functionality to the appropriate ISDN standards, where applicable.

The HPC ISDN Software Package will require additional software drivers and application-specific tasks prior to serving as a useful ISDN Terminal Endpoint (TE) entity. The HPC ISDN software has been coded and documented to allow easy integration of additional application code.

The HPC ISDN Software elements illustrated in *Figure 2.1* have been divided into the following categories.

- HPC Executive (2.2)
- I/O Device Drivers (2.3)
- ISDN Layer Protocol Tasks (2.4)
- Application Tasks (2.5)
- System Utilities (2.6)

The HPC Executive contains software elements that are necessary for HPC ISDN Applications. These elements include a Multi-Tasking Scheduler, a Memory Manager, a Timer Manager and a Mail Manager. The HPC Executive software elements are tightly coupled, and streamlined for the National Semiconductor HPC family of controllers.

The I/O Device Drivers interface the HPC hardware elements to the HPC ISDN Software. The Layer 1 Driver implements the ISDN PHYSICAL Layer 1 requirements for the HPC ISDN system. The Layer 2 Driver interfaces the HPC DMA/HDLC controller channels to the Layer 2 Link Access

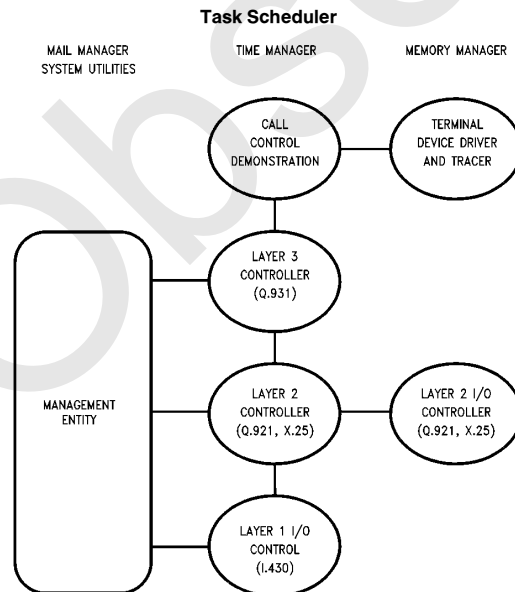


FIGURE 2.1 HPC ISDN Software Functional Block Diagram

TL/DD/10077-3

2.0 Functional Description (Continued)

Procedures. The Terminal Device Driver interfaces the HPC on-board UART to the ISDN Software. Device initialization sequences, service request tasks and accompanying interrupt service routines are all defined in the I/O Device Driver section of this document.

The Layer Protocol Tasks implement the ISDN DATA LINK Layer 2 and the NETWORK Layer 3 requirements for the HPC ISDN system. These tasks are designed to be hardware configuration and application independent. The Layer 2 Task provides both the "USER SIDE" and the "NETWORK SIDE" implementation of the CCITT Specification Q.921. The Layer 3 Task provides the "USER SIDE" implementation of CCITT Specification Q.931.

The Layer 2 Task has been designed to use many of the same routines to implement the link access procedures on either the signaling D channel or the bearer B channel (LAPD or LAPB). Design decisions have also been made to facilitate the implementation of V.120, the rate adaption protocol that processes LAPD frames on a bearer B channel.

The Management Entity Task and the Call Control Task are Application (Specific) Tasks that are closely coupled to the specific system hardware configuration and the Central Office Network Entity Software. These tasks are provided for demonstration purposes to drive the ISDN layer entities. Application users must either replace or extensively rewrite these tasks to match their particular ISDN Application environment.

The System Utilities include the power-up reset Main Task, the NMI handler, the Timer interrupt handler, and the Watchdog Task.

The Tracer utility provides the capability of on-line tracing of intertask mail messages and task states. Tracer is primarily a passive task; it displays messages that it receives from other tasks. Tracer also provides a user interface for Telephone Simulation.

The remainder of this document is devoted to defining each of the software elements at the functional level. Where applicable, specific ISDN standard documents such as CCITT Q.921, Q.931 and X.25 will be referenced, rather than duplicating the information here.

2.2 HPC EXECUTIVE

The HPC Executive provides a multitasking environment within which the ISDN and applications tasks can run and it provides various system services to those tasks. The services of the Executive are available to both tasks and interrupt service routines.

2.2.1 Tasks, Priorities, and the Ready Queue

A task is a subroutine which can be run (called) by the Executive. Tasks are managed by the Executive as Task Control Blocks (TCB's). A task's TCB contains all the parameters needed by the Executive to handle the task, in particular, the task's priority and its current starting address.

Tasks which are not blocked waiting for a semaphore or for mail are considered to be ready to run and their TCB's are queued on the Ready Queue, in the order of the tasks' priorities. The Task Scheduler runs the task at the head of the Ready Queue, i.e., the highest priority task that is ready to run. In this way the processor is always given to the highest priority task that is ready to run.

Once a task is started, it continues to run until it does a Semaphore Wait, ReadMail, or Return or, until a higher priority task is put on the Ready Queue, at which time the scheduler has the opportunity to once again choose the task at the head of prioritized Ready Queue and run that task.

A task may change the priority of any task, including itself. The priority change takes place immediately, to the extent that the target task's TCB is updated with the new priority and the queue in which the target task's TCB is waiting is resorted to reflect the new priority.

If the target task is in the Ready Queue and its new priority is higher than the priority of the running task, then the target task will run once all protected sections are exited. See Section 2.2.3, below.

2.2.2 Semaphores

A semaphore is a global variable, accessed through the Executive, which can be Signaled (incremented) by one task and Waited on by another task. A semaphore is typically used to manage the sharing between tasks of some resource, e.g., an I/O device, mail messages, etc. At any moment the value of a semaphore may be positive, negative, or zero. A positive value indicates the number of resources available, a negative value indicates the number of tasks waiting for resources and a zero value indicates that there are no resources available and no tasks waiting for them.

When a task Waits on a semaphore, if the semaphore has a nonzero positive value, the task will immediately go on the Ready Queue and the semaphore value will be decremented by one. On the other hand, if the semaphore has a zero or negative value, the task will be queued on the semaphore and the semaphore value will be decremented by one. When a task Signals a semaphore, the semaphore's value is incremented by one and the highest priority task waiting on the semaphore is put on the Ready Queue.

A common use for a semaphore is the management of a non-shareable resource, such as an I/O device. When the device is available, the associated semaphore has the value + 1. When a task wishes to obtain exclusive use of the device, it Waits on the semaphore, which is then decremented to 0, with the task going immediately back on the Ready Queue. If another task then attempts to use the device, its Wait call will cause it to be placed on the Semaphore Queue and the value of the semaphore will be decremented to - 1. Other tasks may also Wait on the semaphore, each decrementing its value by one. The negative value of the semaphore indicates the number of tasks Waiting for the device. The waiting tasks are ordered in the semaphore queue according to their priority. When the first task is done with the device, it Signals the semaphore, which moves the first waiting task to the Ready Queue and increments the semaphore or, if there are no waiting tasks, returns the semaphore to its original value of + 1.

2.2.3 Preemptive Scheduling

Preemptive scheduling enables the executive to respond quickly to high priority events. If a task that is waiting on a Semaphore Queue modes to the Ready Queue and if that task is of higher priority than the currently running task, then, as soon as the currently running task emerges from all critical sections and non-preempt sections, the currently running task will stop running. The task that was moved to the Ready Queue will run. The preempted task will be placed on the Ready Queue in the normal manner.

2.0 Functional Description (Continued)

Executive functions allow preemption to be selectively turned on or off by task or for an entire application.

2.2.4 Time Slicing

Time slicing modifies the task scheduling algorithm as follows: at each “tick” of the timer clock (the clock which also controls the time-out timers), if the currently running task has the same priority as the task at the head of the Ready Queue, then, if the currently running task is not in a non-preempt section, it will stop running and the task at the head of the Ready Queue will run. The task that stops running is placed on the Ready Queue in the normal manner, i.e., after all tasks of equal priority. Time slicing enables the Executive to share the processor equally between tasks of equal priority.

2.2.5 Mailboxes and Mail Messages

The main form of intertask communication is the sending and receiving of mail. Mailboxes exist independently of tasks; any task may send mail to any mailbox and any task may read mail from any mailbox. However, in a typical system, each task has one mailbox from which it reads all its mail and to which other tasks send mail destined for that task.

Mail is prioritized. When a task calls upon the Executive to perform a SendMail, it specifies the priority of the message, which is inserted in the specified mailbox queue sorted by priority.

2.2.6 Timers

The Executive includes a timing facility specifically designed to handle the time-outs typical of telecom protocols and other real-time applications.

Timers are essentially a form of delayed mail. When a task sets a timer, the task provides a mailbox identifier, a mail message, and a time delay value. When the specified time delay is up, i.e. when the timer “expires”, the mail message is mailed to the specified mailbox. When a task sets a timer, it receives a timer ID, which can be used to cancel the timer, if necessary, before it expires.

2.2.7 Memory Management

The memory manager is responsible for allocating and deallocating fixed-size memory blocks from fixed-size pools, which are completely defined at compile time. A memory pool may reside in extended memory.

2.2.8 System Module and Interface Module

The Multi-Tasking Executive Software is available either as source code or as object code. The interface module, which must be modified to insert application tasks, is always supplied as source code.

2.3 I/O DEVICE DRIVERS

I/O Device Drivers serve as interface routines between the HPC hardware machinery and the HPC Executive and Application Tasks. “Input” operations (data heading toward Application Tasks) are typically fielded by an Interrupt Service Routine (ISR). The ISR may SEND information to the appropriate task via the system mail facility, or it may signal the appropriate semaphore to schedule an I/O task. “Output” operations (data heading away from Application Tasks) are typically fielded by Service Request (SRQ) Tasks. SRQ Tasks communicate directly with the hardware control registers to initiate output operations. These tasks often work

closely with their accompanying ISR for output initiation and completion. Higher layer tasks send mail messages to the SRQ Tasks, using the system mail facility to queue messages pending output.

The HPC ISDN Software includes three I/O Device Drivers: the Layer 1 Driver, the Layer 2 Driver and the Terminal Driver. The functionality of these drivers is defined below. Details of particular Device Driver ISR and SRQ Task interactions are defined in the Software User's Manual.

2.3.1 Layer 1 I/O Device Driver

The Layer 1 I/O Device Driver provides implementation of the ISDN PHYSICAL Layer 1 for the HPC environment. This Device Driver controls the NSC MICROWIRE/PLUS Interface to the NSC TP3420 “S” Interface Device (SID), and the HPC16400 onboard, Serial Decoder. Control of a COMBO™ Codec, a display, and a keypad has been implemented later by either adding to this driver, or using it as a model for additional drivers.

The primary responsibility of this driver is to initialize and control the SID. The higher layer ISDN tasks mail activation and deactivation messages to the Layer 1 Service Request Task. This task sends the appropriate command to the SID via the MICROWIRE/PLUS Interface. The SID interrupts the HPC whenever it changes state. The Layer 1 Interrupt Service Routine fields responses when the SID changes state and mails the information to the Layer 2 Controller Task and to the Management Entity Task.

The Serial Decoder is initialized to MODE 4, with the ISDN D Channel terminated by DMA/HDLC Channel #1, and Bearer Channel B2 terminated by DMA/HDLC Channel #2. The SID can swap B1 and B2 internally to allow voice or data on either channel.

The Layer 1 I/O Device Driver can communicate with any other Task via the System Mail Utilities.

2.3.2 Layer 2 I/O Device Driver

The Layer 2 I/O Device Driver interfaces the two HPC16400 onboard DMA/HDLC channels; one to the 16 kbit per second “D” signaling channel, and one to the 64 kbit per second bearer (B2) channel. The Layer 2 Service Request Task receives Physical Layer (PH) Primitives from the Layer 2 Controller Task via the system mail utility. The Layer 2 Interrupt Service Routine handles block messages received from the DMA Controller and mails them as Physical Layer (PH) Data Primitives to the Layer 2 Controller Task. This generic mail message interface allows an Application User to easily introduce external DMA and HDLC Controllers, and accompanying device drivers, that either replace or complement the existing onboard controllers.

HDLC/DMA Channel #1 is attached to the ISDN signaling D channel, and will be referred to as the LAPD Channel. HDLC/DMA Channel #2 is attached to bearer channel B2, and will be referred to as the LAPB Channel. The two channels operate independently of each other as much as possible. Since they share the same interrupt hardware, the Layer 2 Interrupt Service Routine must poll the Message Pending Register and the Error Status Register to determine the source of each interrupt. Both HDLC/DMA channels use the HPC field separation feature for transmission and reception of data. This feature relieves some memory concerns, since it allows small memory buffers to be used for mes-

2.0 Functional Description (Continued)

sages that only have headers. In the transmit direction this feature allows large contiguous buffers to be broken up into smaller send buffers without having to copy them following a header. Issues specific to the HDLC/DMA Channels are defined below.

HDLC/DMA Channel #2, the LAPB Channel, requires frame sizes to be nominally 130 bytes, 2 bytes of header and 128 bytes of information. Provision can be made for messages with up to 1026 bytes, 2 bytes header and 1024 bytes of information.

The presentation of data between the Layer 2 Driver and Layer 2 Controller is identical regardless of which channel the frames are associated with.

2.3.3 Terminal Device Driver and Tracer

The Terminal Device Driver interfaces to the HPC onboard UART. The associated SRQ Driver Task, referred to as Tracer, serves primarily as a high-level demonstration vehicle. Tracer can field mail messages from any other task in the system, as well as keystroke mail messages from its accompanying ISR. Tracer's responsibilities include the following functions:

- Management of the Telephone Simulation User Interface,
- Display Management of the System Trace Mail Messages,
- Proper Display of Task-Related Information

The Telephone Simulation function of Tracer allows the user to enter "telephony-like" keystroke characters, that are passed to Tracer, then on to the ISDN layer tasks for processing. Menu responses are fielded by Tracer to select various levels of the Trace function, as well as to enter and exit the Telephone Simulation mode.

Depending on the level of trace that is selected, Tracer receives mail messages from the system tasks and properly formats them on the CRT display. Tracer offers various levels of trace capability. Trace can be turned off all together, in which case only the application layer Telephone Simulation inputs will be displayed. Trace can display all messages from every layer, or it can be set to "zoom" to display only the messages at a particular layer. Messages will generally have address fields and data fields.

The Terminal Driver's Interrupt Service Routine (ISR) handles keyboard characters from the UART and mails them to the Tracer SRQ Task for further processing. The ISR also handles transmission completion of a character that has been sent to the CRT.

The data structures and hardware interface requirements for the Terminal Device Driver, and capabilities of Tracer, are defined in the Software User's Manual.

2.4 ISDN LAYER PROTOCOL TASKS

The ISDN Layer Protocol Tasks provide implementation of the DATA LINK Layer 2 and the NETWORK Layer 3 in accordance with the protocol definitions of the CCITT Specifications. The two Layer Protocol Tasks (the Layer 2 Controller Task and the Layer 3 Controller Task) are designed to satisfy the ISDN Basic Rate Interface (BRI) Terminal Equipment requirements. They are independent of user applications and hardware environment. The PHYSICAL Layer 1 implementation is defined in the I/O Device Driver section of this document. Implementation of layers above the NETWORK Layer 3 are specific to user applications. Two such

layer tasks are provided, the Demonstration Call Control Task and the Management Entity Task. These tasks are defined in the Application Task section of this document.

The purpose of the Layer 2 Controller Task is to provide the NETWORK Layer 3 with an error free, sequenced data frame service. The Layer 2 Controller Task uses CCITT Specifications Q.921 and X.25 and the primary functional specifications. The Layer 2 Controller Task satisfies the Link Access Procedures for both the D Channel and the B Channel (LAPD and LAPB). Design considerations have also been included for the future implementation of V.120, the new CCITT rate adaption scheme.

The Layer 2 Controller Task's data frame delivery service allows the Layer 3 Controller Task to confidently setup and teardown user voice and data calls on the available facilities. The Layer 3 Controller Task uses CCITT Specification Q.931 as the primary functional specification. Note that the X.25 Layer 3 packet processor task is not included in the initial software package.

The Layer Protocol Tasks require a somewhat non-conventional task architecture in order to simultaneously manage a significant number of multiple logical connections. This event-driven state-machine architecture requires that a state memory block be created and maintained for each logical connection. When a Layer Protocol Task "wakes up" due to the arrival of mail, the message's address is interrogated to determine which logical connection is to receive the mail. The particular logical connection's state block is retrieved and the mail message is processed per the CCITT Specification requirements, depending on the state of the particular logical connection. Typically, processing the mail message results in sending a Primitive message to another task, and updating the logical connection's state block. The Layer Protocol Task then returns to its mail box to pick up any subsequent mail.

The interface between all of the ISDN Layer Tasks is deliberately achieved via the System Mail Utilities. This ensures a distinct, uniform layering mechanism in the event that application programmers wish to replace layers with their own implementations.

2.4.1 Layer 2 Controller Task

The primary job of the ISDN Data Link Layer 2 is to deliver error-free, sequenced data frames to the Network Layer 3. The Layer 2 Controller Task implements the following Layer 2 Link Access Procedures (LAP) for the HPC ISDN Software Package:

- LAPB per the X.25 CCITT Specification.
- LAPD per the Q.921 CCITT Specification.
- V.120 Terminal Adaption capability.

Since the Q.921 LAPD requirements were derived from the X.25 LAPB requirements, most of the same Layer 2 Controller Task routines can be used to implement both LAPB and LAPD. Design considerations have been made to allow future implementation of V.120.

The Layer 2 Controller Task communicates with the Layer 2 DMA/HDLC Controller Device Driver Task and the Management Entity Task, via the System Mail Utilities. These tasks interrogate the mail message headers to determine whether to process the frames using LAPB or LAPD procedures. The

2.0 Functional Description (Continued)

LPAD frames are mailed to the Q.931 Layer 3 Controller Task, while the LAPB frames are mailed to the X.25 Layer 3 Task.

The HPC16400 HDLC hardware handles the Layer 2 HDLC Procedures, which includes bit stuffing, address recognition, and Frame Check Sequence generation and detection. The Layer 2 Controller Task is responsible for the Layer 2 "Data Link Procedure", which includes the following functions:

- Data Transmission
- Protocol Exception Management
- LAPD-Specific Functions.

To accomplish these functions the Layer 2 Controller supports the full set of Layer 2 Peer-to-Peer messages defined in the CCITT Specification Q.921. These messages are listed below and defined further in the Software User's Manual.

UI	Unnumbered Information Frames
UA	Unnumbered Acknowledge
SABM(E)	Set Asynchronous Balanced Mode (Extended)
DISC	Disconnect Command
DM	Disconnect Mode
I	Acknowledged Information Frames
RR	Receiver Ready
RNR	Receiver Not Ready
REJ	Request Recrimination of Frames
FRMR	Unrecoverable Error, Frame Reject

The Layer 2 Controller Task also supports the primitives required to communicate with the other ISDN tasks.

2.4.1.1 Layer 2 Data Transmission

Layer 2 peer-to-peer Data Transmission is supported with two modes: Unacknowledged Data Mode and Multi-Frame Acknowledged Data Mode. The Unacknowledged Data Mode is used primarily for setting up logical connections and for peer-to-peer Management Entity communication. This mode uses the Unnumbered Information (UI) and the Unnumbered Acknowledge (UA) messages. The Multi-Framed Acknowledged Mode is established by the Set Asynchronous Balanced Mode (SABM) command. This mode provides the mechanism for acknowledgement of data frame transport in each direction. The Multi-Frame Acknowledged Mode is terminated with the Disconnect (DISC) command. The response to the DISC message can be either an Unnumbered Acknowledge (UA) message or a Disconnect Mode (DM) message. The actual Layer 2 data frames are transmitted in the Information (I) messages, while in the Multi-Framed Acknowledged Mode.

The Layer 2 Controller is responsible for avoiding message congestion and buffer overflow. A Layer 2 entity can issue the Receive Ready (RR) command to its peer to indicate that it is ready to continue data transmission. Likewise, the Layer 2 Controller can issue the Receiver Not Ready (RNR) command to its peer to indicate that it is not ready for data transmission.

2.4.1.2 Layer 2 Protocol Exception Management

The Layer 2 Controller Task is responsible for handling exceptions to the Data Link Protocol. These exceptions are of

two types: recoverable and unrecoverable. Recoverable exceptions in the receive direction are typically failed frames, which are handled by requesting the retransmission of the failed frame with the Reject (REJ) command. Recoverable exceptions in the transmit direction include the expiry of a Layer 2 Timer. Timer expiry requires the retransmission of the frame that was not acknowledged in time, and all subsequent frames. Timer expiry also prompts a message to the Management Entity. Unrecoverable exceptions result in the Frame Reject (FRMR) response. A message to the Management Entity Task is also sent in this case.

2.4.1.3 Layer 2 LAPD-Specific Functions

The following Layer 2 Controller Task functions are LAPD specific. These functions involve establishing and maintaining multiple logical data link connections. Note that a LAPB connection will be maintained as a special independent logical connection.

A two byte address is required for each logical data link. This address is referred to as the Data Link Connection Identifier (DLCI). The DLCI consists of a Service Access Point Identifier (SAPI) and a Terminal Endpoint Identifier (TEI). The Layer 2 Controller Task is responsible for supporting the TEI Assignment Procedure and the TEI Verification Procedure. These procedures are both initiated by the Management Entity. The Layer 2 Controller Task supports both the Automatic and Non-Automatic TEI Assignment Procedures.

Establishment of the LAPD multi-frame acknowledged data transmission mode requires an extended command (SABME) to prompt the peer entity that the frames are intended for a particular logical data connection identified by the accompanying DLCI. The Layer 2 Controller Task maintains each logical link's state and data frames independently, as explained earlier in this section.

The Layer 3 Controller Task addresses and maintains independent logical connections via an identifier called a Connection Endpoint Suffix (CES). Since the CES is different from the Layer 2 Terminal Endpoint Identifier (TEI), a mapping function is required. The Layer 2 Controller Task maintains a CES-TEI translation procedure to properly address Layer 3 logical entities.

2.4.2 Layer 3 Controller Task

The Layer 3 Controller Task implements the application independent portion of the ISDN NETWORK Layer 3 protocol, per the Q.931 CCITT Specification. The primary responsibility of the Layer 3 Controller Task is to establish a network access connection link between a terminal and its peer in the Central Office.

The Layer 3 Controller Task communicates with both the Layer 2 Controller Task and the Call Control Task by sending primitives via the System Mail Utilities. The Layer 3 Controller Task also communicates with the Management Entity Task. The HPC ISDN Layer 3 Controller Task is responsible for the following NETWORK functions:

- Call Establishment and Clearing
- Call Suspension and Resumption
- Call Status and Notification
- Protocol Exception Management.

2.0 Functional Description (Continued)

The Layer 3 Controller Task supports all the Network Layer Peer-to-Peer messages defined in the CCITT Specification Q.931, i.e.:

- Call Establishment and Clearing Messages:

ALERT	Alerting
CALL PROC	Call Proceeding
CONN	Connect
CONN ACK	Connect Acknowledge
INFO	Information
PROG	Progress
SETUP	Setup
SETUP ACK	Setup Acknowledge
DISC	Disconnect
REL	Release
REL COM	Release Complete

- Call Suspension and Resumption Messages

RESUME	Resume
RESUME ACK	Resume Acknowledge
RESUME REJ	Resume Reject
SUSPEND	Suspend
SUSPEND ACK	Suspend Acknowledge
SUSPEND REJ	Suspend Reject

- Miscellaneous Messages

NOTIFY	Notify
STATUS	Status
STATUS EN	Status Enquiry
USER INFO	User Information

2.4.2.1 Call Establishment And Clearing

The Layer 3 Controller Task's primary responsibility is to establish and clear user network connections on available bearer channel facilities. The Q.931 CCITT Specifications include Call Establishment and Clearing of both circuit-switched and packet-switched calls. Initially, the HPC ISDN Software Package only supports circuit-switched call procedures on Basic Rate Interface (BRI) Bearer Channels. The Layer 3 Controller Task is responsible for Call Reference assignment and maintenance. The Layer 3 Controller Task supports Call Establishment using both the Overlap and Non-Overlap (enbloc) addressing modes.

The procedure for establishing and clearing network connections is defined in CCITT Specification Q.931. It is important to note that the Layer 3 Controller Task maintains an associated state block for each network connection. Primitive mail messages arriving at the Layer 3 Controller Task will be interrogated to determine which network connection is to receive the mail. The mail message is processed depending on the state of the network connection. This processing typically includes the transmission of a Primitive to another Layer Task, and the appropriate update of the network connection state block.

2.4.2.2 Call Suspension And Resumption

Call Suspension (SUSPEND) requires that the Bearer Channel facility and the Call Reference for a call be temporarily relinquished. The network connection is left intact, but in the suspend state. The RESUME command reactivates the call by obtaining a Bearer Channel facility and establishing a new Call Reference. The Suspend function is somewhat analogous to the call HOLD feature.

2.4.2.3 Call Status And Notification

The Network can request the status of a network connection at any time via the USER INFO, NOTIFY and STATUS Commands. The information includes Service Validation and Channel Configuration.

2.4.2.4 Layer 3 Protocol Exception Management

The Layer 3 Controller is responsible for handling exceptions to the Network Control Protocol. The primary Layer 3 Controller Task protocol exception is the expiry of the Layer 3 timer. Such an exception requires the retransmission of the particular command and may prompt a message to the Management Entity Task.

2.4.2.5 Timer Support

The Layer 3 Controller supports the following system timers per CCITT Specification Q.931:

T303	SETUP ACK Timer
T305	DISCONNECT ACK Timer
T308	RELEASE ACK Timer
T313	CONNECT ACK Timer

2.4.2.6 SDL Updates

The Layer 3 Controller Task very closely follows the SDL procedures illustrated in CCITT Specification Q.931, with a few enhancements. These enhancements are listed here and fully defined in the Software User's Manual.

- Three new SDL States have been added to accommodate establishing the Data Link corresponding to a particular CES. The new states are:

- IDLESTATE
- RELEASEWAIT
- ESTABLISHWAIT

- The Q.931 NULLSTATE SDL now accepts a new command, CCBROADCASTRESP. This command is sent from the Call Control Task to allow transition from the NULLSTATE(0) to the CALLPRESENT State(6) during a Network Originated call via the Broadcast mechanism.

2.5 APPLICATION TASKS

The Application Tasks are very dependent on both the terminal equipment configuration and the far-end Network Entity software implementation. The HPC ISDN Software Package includes two sample Application Tasks: the Demonstration Call Control Task and the Management Entity Task. Both of these tasks can be replaced or updated when ported to a particular application. These tasks are included in the HPC ISDN Software Package primarily to verify the operation of the OSI Layer Protocol Tasks and the HPC Device Drivers.

2.5.1 Demonstration Call Control Task

The Demonstration Call Control Task is closely coupled to the specific facilities of an application. The interaction between the Layer 3 Controller Task and Call Control is defined in CCITT Specification Q.931. In the HPC ISDN Application, the Call Control Task communicates with the Layer 3 Controller Task and the Tracer Task. The availability of two circuit switched voice bearer channels is simulated in the Call Control Task. The Call Control Task sends standard Terminal Equipment prompts and messages to the Tracer Task where they are displayed on a UART driven CRT. The Call Control Task has the following responsibilities:

- B Channel Resource Management

2.0 Functional Description (Continued)

- Connection Endpoint Suffix (CES) Allocation
- Conversion between L3 Primitives and Terminal Action.

The Call Control Task and the Layer 3 Controller Task communicate via the NL_DATA_REQ and NL_DATA_IND Primitives. The messages that are supported between these tasks are listed below.

- Commands from Call Control to Layer 3

CC_SETUP_REQ	Setup Request
CC_SETUP_RESP	Setup Response
CC_SETUP_REJ_REQ	Setup Reject
CC_INFO_REQ	Information
CC_DISCONNECT_REQ	Disconnect
CC_RELEASE_REQ	Release
CC_ALERTING_REQ	Alerting
CC_BROADCAST_RESP	Broadcast Response
CC_CALLPROC_REQ	Call Proceeding
CC_PROGRESS_REQ	Progress
CC_NOTIFY_REQ	Notify
CC_RESUME_REQ	Resume
CC_RESUME_REJ	Resume Reject
CC_SUSPEND_REQ	Suspend Request
CC_SUSPEND_REJ	Suspend Reject
- Command from Layer 3 to Call Control

CC_SETUP_IND	Setup
CC_SETUP_CONF	Setup Confirm
CC_SETUP_COMP_IND	Setup Complete
CC_INFO_IND	Information Indication
CC_ALERTING_IND	Alerting
CC_PROGRESS_IND	Progress
CC_DISCONNECT_IND	Disconnect
CC_RELEASE_IND	Release
CC_CALLPROC_IND	Call Proceeding
CC_RELEASE_CONF	Release Confirm
CC_STATUS_IND	Status Indication
CC_ERROR_IND	Error Indication
CC_RESUME_CONF	Resume Confirm

The Call Control Task also communicates with the Tracer Task using single byte keystroke like commands. These commands are packaged mail messages containing two bytes: the first byte is the Sender Task's ID, the second byte is the keystroke command. The following messages are sent between Call Control and Tracer:

- Keystroke Commands from Tracer to Call Control Task

TR_ON_HOOK	ON Hook
TR_OFF_HOOK	OFF Hook
TR_DIGIT_1	Digit 1
TR_DIGIT_2	Digit 2
TR_DIGIT_3	Digit 3
TR_DIGIT_4	Digit 4
TR_DIGIT_5	Digit 5
TR_DIGIT_6	Digit 6
TR_DIGIT_7	Digit 7
TR_DIGIT_8	Digit 8
TR_DIGIT_9	Digit 9

TR_DIGIT_0	Digit 0
TR_DIGIT_STAR	Digit *
TR_DIGIT_POUND	Digit #

- Commands from Call Control Task to Tracer

TR_IDLE	Idle, ON HOOK
TR_DIALTONE	Dial Tone
TR_DIALING	Dialing
TR_RINGING	Ringing
TR_BUSY	Busy
TR_CONVERSATION	Conversation
TR_RINGBACK	Ringback
TR_ERROR	Error

2.5.2 Management Entity Task

The Management Entity Task is closely coupled to the accompanying Network Management Entity design and to the terminal hardware configuration. Implementation design decisions have been made that make the Management Entity Task unique to a particular application, while still following the general requirements of the CCITT Specifications. Modifications will be required in the Management Entity Task prior to its successful operation in a particular application environment. The Management Entity Task that is included in the HPC ISDN Software Package presumes a particular hardware configuration and Central Office Software implementation.

The Management Entity Task communicates with the Layer 3 Controller Task, the Layer 2 Controller Task, and the Layer 1 Device Driver Task via the System Mail Utilities.

The Management Entity Task has the following responsibilities:

- Initialization of the Terminal Equipment
- Configuration of the Terminal Equipment
- TEI Assignment and Verification
- Multiple Error Notification
- Unrecoverable Error Notification
- Activation/Deactivation of the Terminal Equipment.

2.6 SYSTEM UTILITIES

The system utilities initializes the HPC system upon power-up, and provide support for various machine specific features of the HPC.

2.6.1 Power-Up Reset Main Task

This task is the entry point upon system power-up. The Main Task is responsible for:

- Initializing the general HPC Hardware.
- Initializing the HPC Executive Data Structures.
- Queuing up the Tasks on the Ready Queue.

The Main Task starts with the highest priority, 255. After running, the Main Task has served its purpose and is removed from the system by waiting on a semaphore which is typically never signaled.

2.6.2 Nonmaskable Interrupt (NMI) Handler

Since terminal power is generally a concern, the HPC can go into an idle, low-power mode when the Terminal Equipment is idle. In this mode the HPC is awakened via an NMI, prompted by a local off hook indication, or by a far-end line

2.0 Functional Description (Continued)

signal detection signal from the SID. Conditions for determining when to go in and out of idle mode are application dependent.

2.6.3 Timer Interrupt Handler

The Timer Interrupt Handler fields interrupts from two of the HPC onboard timers. Timer T0, the Watchdog Timer, overflows every 65536 clock counts. When this occurs the Timer Interrupt Handler mails a message to start the Watchdog Task. Timer T1, the ISDN Software Timer, overflows every 10 ms. The ISDN Software Clock is incremented every tenth Timer T1 overflow, resulting in an ISDN Clock with 100 ms resolution, which is used by the Executive Timer facility.

2.6.4 Watchdog Task

A special task is performed by the HPC's watchdog feature to verify system sanity. The Watchdog Task waits for a mail message that is sent by the Timer Interrupt Handler when Timer T0 overflows. This operation requires that the Watchdog Task be regularly scheduled by the HPC Executive. The Watchdog Task is assigned the highest task priority, 255.

3.0 Ordering Information

3.1 LICENSE AGREEMENT

A license agreement is required for the use and sale of the National Semiconductor ISDN Software. Contact your local National Semiconductor field sales office for more information or contact the factory direct at:

National Semiconductor
ISDN Software Support
M/S 16-174
2900 Semiconductor Drive
Santa Clara, CA 95051
(408) 721-5719

3.2 SOFTWARE ORDER INFORMATION

ISDN software is available in either Object or Source Code format. A Demonstration package is also available. Manuals are included with the Demonstration package and with the Executive and Basic Rate Interface Software packages.

Basic Rate Interface (BRI) software is available for several different central office switches. The generic BRI includes a generalized CCITT Switch Interface.

Each BRI Package contains the following modules:

- Layer 1 Driver (controls S device)
- Layer 2 Driver (controls DMA/HDLC)
- Layer 2 Controller (Q.921)
- Layer 3 Controller (Q.931 Protocol Control)
- Management Entity (Q.921 and Q.931)
- Call Control (Demonstration Application)
- Tracer (Demonstration and Development Tool)

The Multi-Tasking Executive contains two modules:

- Executive Core Module
- Executive Interface Module

The Executive Interface Module is always supplied as source code to allow modification to insert application tasks. A Multi-Tasking Executive is required to run the Basic Rate Interface.

Order Part Number Description

Multi-Tasking Executive

HPC-ISDN-EXEC-O Multi-Tasking Executive Object Code

Basic Rate Interface

HPC-ISDN-BRI-S Basic Rate Interface (Generic) Source Code

HPC-ISDN-BRID-S Basic Rate Interface (DMS-100) Source Code

HPC-ISDN-BRI5-S Basic Rate Interface (5ESS) Source Code

Demonstration Package

HPC-ISDN-PCDEMO ISDN Basic Rate Interface Demonstration (includes Multi-Tasking Executive and Basic Rate Interface Software Manuals)

4.0 Other Related Information

4.1 DEVICE INFORMATION

Additional technical information on devices referenced in this datasheet is available from National:

HPC16400 High Performance microController

HPC16083 High Performance microController

TP3076 COMBO II™

TP3420 CCITT S/T Interface

4.2 DEVELOPMENT SUPPORT INFORMATION

Development tools are available for the HPC Family of Microcontrollers. These tools support the ISDN development environment. ISDN software must be ordered separately.

4.2.1 ISDN Demonstration Kit

A kit is available that demonstrates the software and hardware discussed in this datasheet. Included in this kit is a TP3500 development board featuring the HPC16400, TP3070 COMBO II, TP3420 "SID" and ISDN Basic Rate Interface software in ROM. A complete set of manuals are included. This demonstration kit may be ordered from National, part number.

ISDN-TP3500-Kit

4.2.2 Development Systems

Several different Microcontroller-On-Line-Emulator Development Systems are available for hardware and software development of the HPC Family of Microcontrollers. Complete information on Development Systems and Accessories may be found in the Microcontroller Development Support Datasheet.

LIFE SUPPORT POLICY

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.



National Semiconductor Corporation
1111 West Bardin Road
Arlington, TX 76017
Tel: 1(800) 272-9959
Fax: 1(800) 737-7018

National Semiconductor Europe
Fax: (+49) 0-180-530 85 86
Email: cnjwge@tevm2.nsc.com
Deutsch Tel: (+49) 0-180-530 85 85
English Tel: (+49) 0-180-532 78 32
Français Tel: (+49) 0-180-532 93 58
Italiano Tel: (+49) 0-180-534 16 80

National Semiconductor Hong Kong Ltd.
19th Floor, Straight Block,
Ocean Centre, 5 Canton Rd.
Tsimshatsui, Kowloon
Hong Kong
Tel: (852) 2737-1600
Fax: (852) 2736-9960

National Semiconductor Japan Ltd.
Tel: 81-043-299-2309
Fax: 81-043-299-2408

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

TI products are not authorized for use in safety-critical applications (such as life support) where a failure of the TI product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use. Buyers represent that they have all necessary expertise in the safety and regulatory ramifications of their applications, and acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning their products and any use of TI products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by TI. Further, Buyers must fully indemnify TI and its representatives against any damages arising out of the use of TI products in such safety-critical applications.

TI products are neither designed nor intended for use in military/aerospace applications or environments unless the TI products are specifically designated by TI as military-grade or "enhanced plastic." Only products designated by TI as military-grade meet military specifications. Buyers acknowledge and agree that any such use of TI products which TI has not designated as military-grade is solely at the Buyer's risk, and that they are solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI products are neither designed nor intended for use in automotive applications or environments unless the specific TI products are designated by TI as compliant with ISO/TS 16949 requirements. Buyers acknowledge and agree that, if they use any non-designated products in automotive applications, TI will not be responsible for any failure to meet such requirements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products

Audio	www.ti.com/audio
Amplifiers	amplifier.ti.com
Data Converters	dataconverter.ti.com
DLP® Products	www.dlp.com
DSP	dsp.ti.com
Clocks and Timers	www.ti.com/clocks
Interface	interface.ti.com
Logic	logic.ti.com
Power Mgmt	power.ti.com
Microcontrollers	microcontroller.ti.com
RFID	www.ti-rfid.com
OMAP Mobile Processors	www.ti.com/omap
Wireless Connectivity	www.ti.com/wirelessconnectivity

Applications

Communications and Telecom	www.ti.com/communications
Computers and Peripherals	www.ti.com/computers
Consumer Electronics	www.ti.com/consumer-apps
Energy and Lighting	www.ti.com/energy
Industrial	www.ti.com/industrial
Medical	www.ti.com/medical
Security	www.ti.com/security
Space, Avionics and Defense	www.ti.com/space-avionics-defense
Transportation and Automotive	www.ti.com/automotive
Video and Imaging	www.ti.com/video

TI E2E Community Home Page

e2e.ti.com

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2011, Texas Instruments Incorporated