INFORMATION TO USERS

This reproduction was made from a copy of a document sent to us for microfilming. While the most advanced technology has been used to photograph and reproduce this document, the quality of the reproduction is heavily dependent upon the quality of the material submitted.

The following explanation of techniques is provided to help clarify markings or notations which may appear on this reproduction.

1. The sign or "target" for pages apparently lacking from the document photographed is "Missing Page(s)". If it was possible to obtain the missing page(s) or section, they are spliced into the film along with adjacent pages. This may have necessitated cutting through an image and duplicating adjacent pages to assure complete continuity.

2. When an image on the film is obliterated with a round black mark, it is an indication of either blurred copy because of movement during exposure, duplicate copy, or copyrighted materials that should not have been filmed. For blurred pages, a good image of the page can be found in the adjacent frame. If copyrighted materials were deleted, a target note will appear listing the pages in the adjacent frame.

3. When a map, drawing or chart, etc., is part of the material being photographed, a definite method of "sectioning" the material has been followed. It is customary to begin filming at the upper left hand corner of a large sheet and to continue from left to right in equal sections with small overlaps. If necessary, sectioning is continued again—beginning below the first row and continuing on until complete.

4. For illustrations that cannot be satisfactorily reproduced by xerographic means, photographic prints can be purchased at additional cost and inserted into your xerographic copy. These prints are available upon request from the Dissertations Customer Services Department.

5. Some pages in any document may have indistinct print. In all cases the best available copy has been filmed.

University Microfilms International
300 N. Zeeb Road
Ann Arbor, MI 48106

Order Number 1332455

Design of a high speed fiber optic network interface for medical image transfer

Byers, Daniel James, M.S.

The University of Arizona, 1987

U·M·I

300 N. Zeeb Rd.
Ann Arbor, MI 48106

PLEASE NOTE:

In all cases this material has been filmed in the best possible way from the available copy. Problems encountered with this document have been identified here with a check mark __✓__.

1. Glossy photographs or pages _____
2. Colored illustrations, paper or print _____
3. Photographs with dark background _____
4. Illustrations are poor copy _____
5. Pages with black marks, not original copy _____
6. Print shows through as there is text on both sides of page _____
7. Indistinct, broken or small print on several pages ✓
8. Print exceeds margin requirements _____
9. Tightly bound copy with print lost in spine _____
10. Computer printout pages with indistinct print _____
11. Page(s) _____ lacking when material received, and not available from school or author.
12. Page(s) _____ seem to be missing in numbering only as text follows.
13. Two pages numbered _____. Text follows.
14. Curling and wrinkled pages _____
15. Dissertation contains pages with print at a slant, filmed as received _____
16. Other _____

_____

_____

DESIGN OF A
HIGH SPEED FIBER OPTIC NETWORK INTERFACE
FOR MEDICAL IMAGE TRANSFER

by

Daniel James Byers

A Thesis Submitted to the Faculty of the

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

In Partial Fulfillment of the Requirements
For the Degree of

MASTER OF SCIENCE
WITH A MAJOR IN ELECTRICAL ENGINEERING

In the Graduate College

THE UNIVERSITY OF ARIZONA

1 9 8 7

STATEMENT BY AUTHOR

This thesis has been submitted in partial fulfillment of requirements for an advanced degree at the University of Arizona and is deposited in the University Library to be made available to borrowers under rules of the Library.

Brief quotations from this thesis are allowable without special permission, provided that accurate acknowledgment of source is made. Requests for permission for extended quotation from or reproduction of this manuscript in whole or in part may be granted by the head of the major department or the Dean of the Graduate College when in his or her judgment the proposed use of the material is in the interests of scholarship. In all other instances, however, permission must be obtained from the author.

SIGNED: Daniel J. Byas

APPROVAL BY THESIS DIRECTOR

This thesis has been approved on the date shown below:

Ralph Martinez
Professor of Electrical
and Computer Engineering

11 December 1987
Date

ACKNOWLEDGMENTS

My sincere thanks go to all those who helped and encouraged me in completing this project. Specifically, I thank:

D. DOOL	my manager, for creating a productive work place and encouraging me to complete my thesis project.
R. Paulsen	for helping with formatting and typing of the thesis.
R. Martinez	my Thesis Advisor, for guiding me through the thesis generation and acceptance process.
Sharon Byers	my wife, for her patience and encouragement.

Thanks to IBM and my past managers for making it possible to take classes to complete my masters degree. To all of my colleagues in the Test Tool Development my thanks for sharing your experience and expertise.

TABLE OF CONTENTS

TABLE OF CONTENTS--Continued

TABLE OF CONTENTS--Continued

List of Illustrations

List of Illustrations--Continued

List of Illustrations--Continued

ABSTRACT

A high speed, 125 mega-bit per second data rate, data communication channel using fiber optic technology is described. Medical image data, generated by CT scanner or magnetic resonance imaging type imaging equipment, passes from standard American College of Radiology - National Electrical Manufactures Association (ACR-NEMA) interface equipment to the High Speed Fiber Optic Network Interface (HSFONI). The HSFONI implements the ACR-NEMA standard interface physical layer with fiber optics. The HSFONI accepts data from up to 8 devices and passes data to other devices or to a data base archive system for storage and future viewing and analysis. The fiber components, system level, and functional level considerations, and hardware circuit implementation are discussed.

INTRODUCTION

Today, most hospitals do distribution and management of radiological information manually. Hospitals convert CT scanner, ultrasound, magnetic resonance imaging scanners and other digital medical images into film images and store them in a film room. Hospitals manually distribute within and outside the hospital films for interpretation by placing film on a light panel for reading (Hedge, Gale, Giunta 1986, p 618). The radiologist and physician perform diagnosis based on these readings. The recorded diagnosis is later transcribed into text. This procedure may be performed several times on a single patient.

Available technology allows automating this process. Direct image acquisition from imaging equipment and digitizing existing films produces an electronic film library. An electronic data base archive stores the images. Library users access the library and view images from a display console. These so-called picture archiving and communication systems (PACS) are a subject of much research work today.

The typical PACS architecture contains some number of image equipments passing images through a common acquisition code to a data base archive. Multiple display consoles then access data base archive data. This is the architecture, shown in Figure 1, commonly found in PACS literature and specifically discussed by Hedge et al. (1986) and van der Voorde et al. (1986). This system is significantly different from traditional data systems because of the large image data file sizes. A typical 1024x1024 picture with 1024 grey resolution levels generates a 16 mega-bit file. Image data file storage and movement becomes a problem.

Until recently, each PACS vendor designed its own communications and archiving protocols. The American College of Radiology and the National Electrical Manufactures Association formed a joint committee (ACR-NEMA) writing a standard, Digital Imaging and Communications, defining inter-device communications (ACR-NEMA 1985, p. iii), helping standardize communications and speed PACS development. The standard's system level architecture is shown in Figure 2 (ACR-NEMA 1985, p. 5).

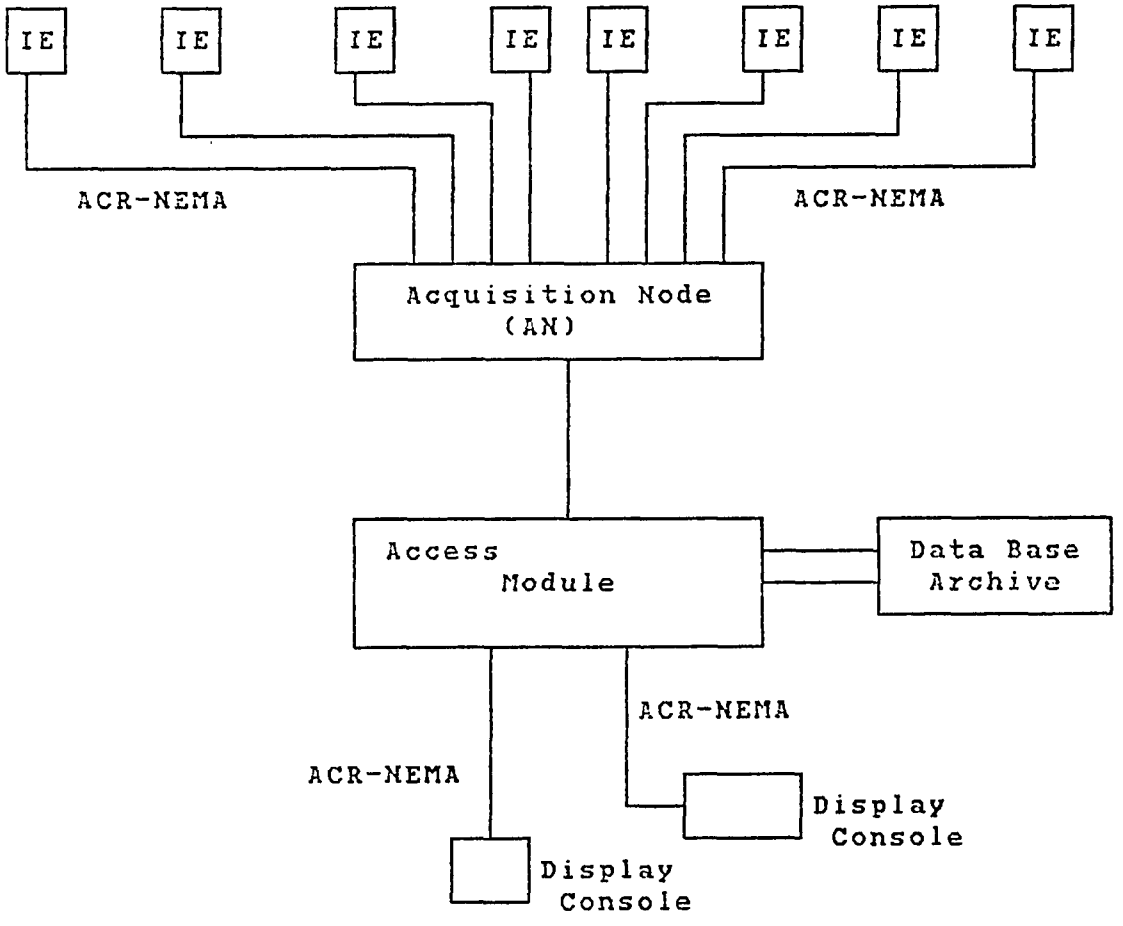

Figure 1. A Typical PACS Architecture

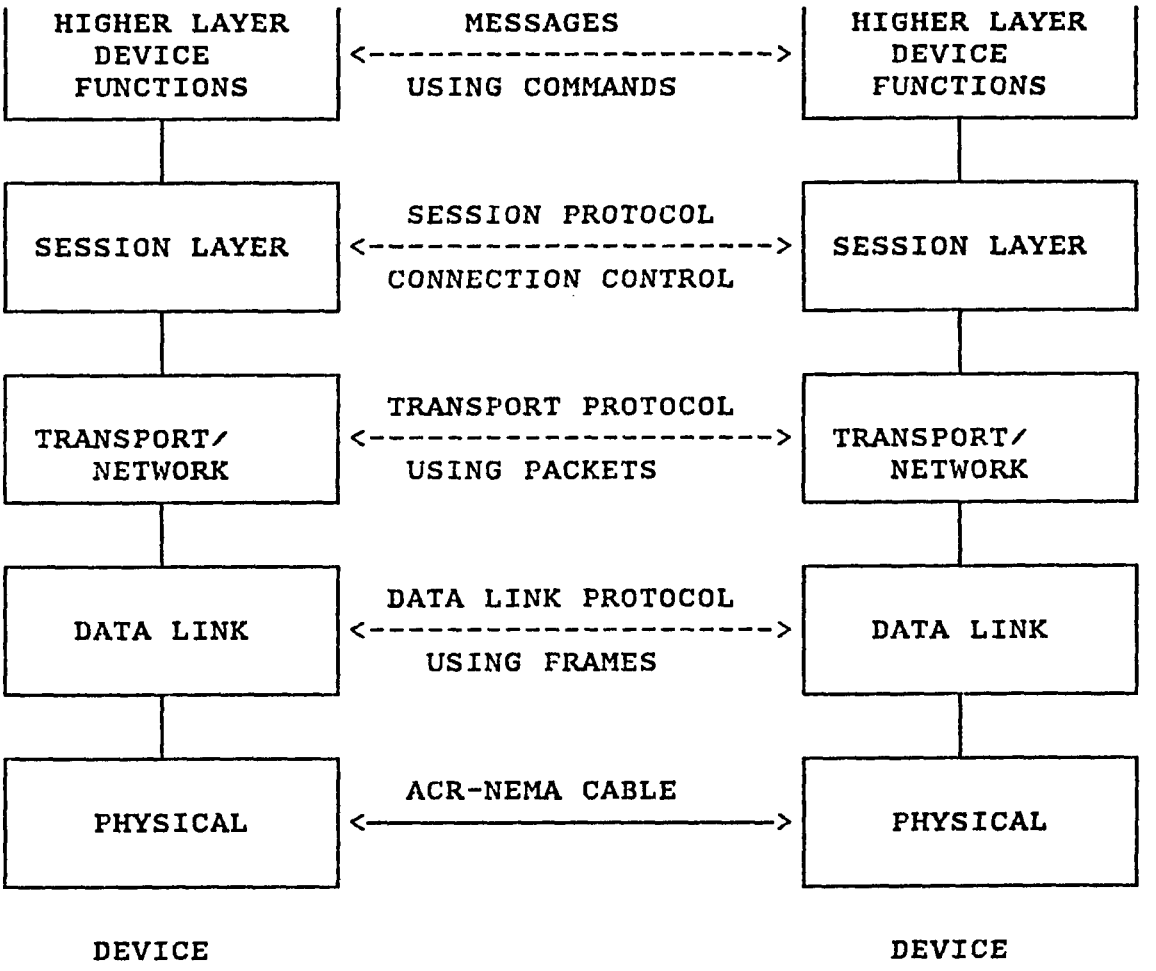

Figure 2. The ACR-NEMA Architecture

Problem Statement

Image data file transfer between the imaging equipment and the data base archive is non-trivial because of data file size. The file transfer speed, if too long, reduces system usability and PACS system advantages. The ACR-NEMA standard helps to standardize vendor interfaces. However, the ACR-NEMA standard does not specify a physical medium taking advantage of fiber communications. A communication link combining the ACR-NEMA standard with high performance fiber communication advantages offers a solution to many PACS environment problems. Today a link of this type is not commercially available.

Figure 3 shows how the High Speed Fiber Optic Network Interface (HSFONI) fits into the overall ACR-NEMA system architecture. A session layer required to send data to a peer session layer opens a virtual channel through which all communications take place. The transport/network layer receives session layer virtual channel data, fragments it, and passes the data through the appropriate service access point to the data link layer. With framing and check sequences added to ensure data validity the data link layer passes data to the physical layer for transmission across the communication medium. In a typical ACR-NEMA system this is a point to point wire cable.

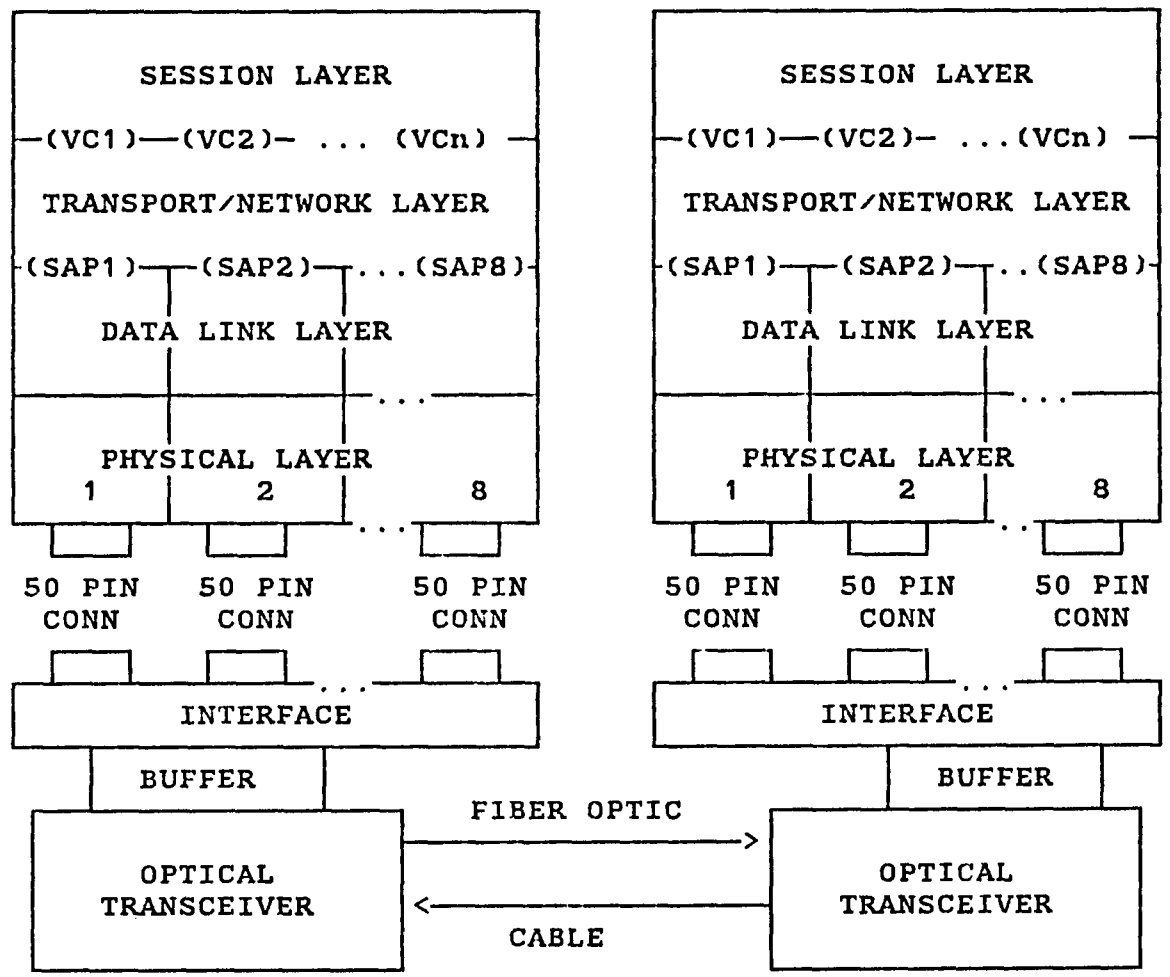

Figure 3. The ACR-NEMA to HSFONI Interface

With the HSFONI design, eight of these connection cables are multiplexed onto one fiber optic link. Because HSFONI only affects the physical layer, all other layers remain unmodified taking advantage of new technology without major system modification.

Objectives

The thesis objective is to extend the electrical ACR-NEMA standard to a fiber link. The link attaches to ACR-NEMA standard devices at each end, but the communication path is fiber instead of copper. This approach increases the distance between the data base archive and the imaging equipment, allowing large hospitals to use a centrally located data base archive instead of many smaller inter-connected data base archives. The high fiber bandwidth and small cross section make fiber especially attractive for this high performance application. Fiber addresses the problems of moving large data files quickly. The specific design objectives for HSFONI follow.

Performance Requirements

The link performance must be greater than the ACR-NEMA 64 mega bits per second (Mbps) device data rate. This allows multiple device communications on the fiber link. The fiber link must ensure data integrity, have high reliability, and low maintainability.

Use

The intended fiber link use is to connect the imaging equipment to the data base archive through an acquisition node in a PACS system. However, any ACR-NEMA standard hospital device requiring a communications channel can attach to the link. In the hospital environment link usage is 24 hours per day, 365 days a year, requiring the fiber link to be very reliable.

Technology

The fiber link design must use currently available vendor TTL and ECL technologies and not require any costly custom designed modules. The design must incorporate any available VLSI modules to reduce system design time. Fiber technology must also use some of the many available vendor components.

This paper proposes a High Speed Fiber Optic Network Interface (HSFONI) for image transfer meeting these objectives.

Approach

The HSFONI design takes three steps to meet the proposed objectives. First, to specify the link requirements requires research into fiber communication link operation and component selection. Then specified are overall system architecture, system functions, and their interfaces. The

last step is circuit design implementing the system.

The first design step is fiber optic research. Because fiber optic technology is so new and in constant change, making the best decision choosing optical components requires research. Products available require investigation, link operation requires understanding and performance affecting parameters require review. This information defines the best fiber optic link and surrounding system implementation method.

Defining the overall system architecture is the second step. This step defines major functional blocks, interconnection as a system, and specifies block interfaces in detail. This step considers data flow between system ends and through functional blocks and system bandwidth and speed. Because the next step requires this information, system definition in this step is important.

The last design step is implementing each system functional block into circuits. This may require further sub-division and interconnect definition. This step defines data flow within a functional block and checks bandwidth and speed of each block. Then implementing each system functional block at the gate level requires designing circuits meeting the functional block requirements. Then connecting the circuits implementing the blocks produces the system with functions as defined. An additional step would

include circuit simulation. However because of the ACR-NEMA interface and HSFONI complexity a system was not available to complete this level of simulation.

Presented in the following sections is the ACR-NEMA standard, the interface basis for HSFONI design. Then detailed are the three design steps. Next investigated is the HSFONI application to a PACS. Finally results and conclusions are covered.

THE ACR-NEMA SPECIFICATION

To fully understand the High Speed Fiber Optic Network Interface (HSFONI), the American College of Radiology and National Electrical Manufactures Association's (ACR-NEMA) Digital Imaging and Communications Specification needs discussion. The HSFONI attaches to medical equipment conforming to this specification. Because the HSFONI replaces the ACR-NEMA standard physical layer definition, system design revolves around this specification. The following sections present a technical overview and a layer by layer comparison of the ACR-NEMA specification to the ISO-OSI reference model. Following this are the hardware and protocols' functional description and performance specifications.

Technical Overview

The ACR-NEMA standard interface provides the medical imaging equipment manufacture a standard for data interchange between devices (ACR-NEMA 1985, p. 3). This allows interconnection of systems produced by different vendors. The standard includes both hardware and software protocols and formats definition. The interface design adheres to the International Standards Organization (ISO) Open Systems Interconnection (OSI) reference model. The

model defines self-contained layers and interfaces. This allows a very structured system.

The ACR-NEMA hardware definition, or physical layer, specifies connector and pin assignment, cabling and interface timings. The interface is parallel and data transfer asynchronous. The physical layer protocol controls data transfers and defines control signal handshakes and interrupts.

The ACR-NEMA standard also defines higher level layers. The data link layer supports data flow control using a frame protocol. A packet protocol used by the transport/network layer supports virtual channels and passes message segments across the interface. The session layer protocol, not specified by ACR-NEMA, handles end to end message delivery. The messages contain data and commands to direct in data handling.

Comparison to the ISO-OSI Reference Model

By comparing ACR-NEMA standard and the ISO-OSI model layers, the standard is better understood. The OSI model provides a framework for defining standards for linking heterogeneous computer systems (Stallings 1984, p37). In the OSI model, communication functions are partitioned into a hierarchical set of layers. Each layer performs a related subset of the functions required to communicate to another system. Each layer relies on the next

lower layer performing more primitive functions and providing services (Stallings 1984, p. 37). The OSI model to ACR-NEMA standard comparison is shown in Figure 4.

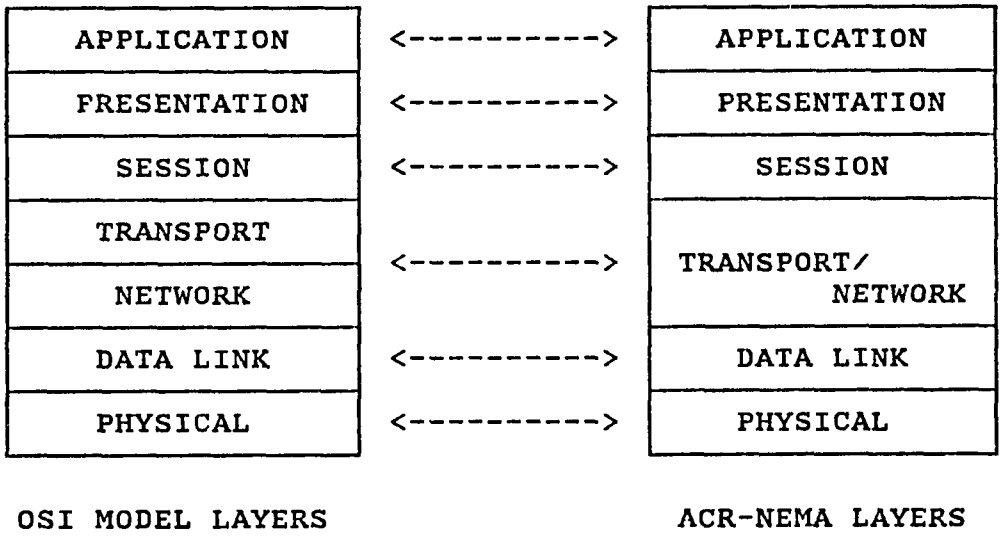

Figure 4. Comparison of OSI and ACR-NEMA Layers

Physical Layer

The physical layer defines the physical interface between devices and the bit passing protocol. The important characteristics defined are mechanical, electrical, functional and procedural (Stallings 1984, p. 43). The ACR-NEMA specification uses a 16 data and 1 parity bit bi-directional bus and six control signal physical layer.

This physical layer corresponds to the OSI model physical layer.

The HSFONI is the physical layer of the ISO model as shown in Figure 3. At the ends, the ACR-NEMA specification physical layer defines device attachment. In this work, a fiber optic link replaces the physical layer electrical connection. The ACR-NEMA data received at one interface end, is changed into a fiber optic link suitable form and protocol, transmitted across the link, and reconverted to the ACR-NEMA format at the other interface end. Since the interface replaces only the physical layer, other upper layers, data link to application, do not require changes.

Data Link Layer

The ACR-NEMA standard defines a data link layer corresponding to the OSI Model data link layer (ACR-NEMA 1985, p. 6). This layer increases the physical layer reliability and provides control to activate, deactivate and maintain link connections (Stallings 1984, p. 44). The ACR-NEMA standard protocol definition uses data framing and adds required control words during data transmission. This layer, upon data receipt removes framing, makes transmission error checks and retransmits data frames in error.

Transport/Network Layer

Above the data link layer the OSI model defines the network layer followed by the transport layer. The network layer handles the packet-switched network details of connecting and disconnecting from a network, and routing and delivery of network data. By fragmenting messages into packets with description headers and sequence numbers, the transport layer ensures end to end reliability. This means data units arrive error-free, in sequence and with no losses or duplications. The transport layer defines the type of service, the grade of service, and connection management (Stallings 1984, pp. 47-49).

The ACR-NEMA standard combines the functions corresponding to the OSI model network layer and transport layer into one layer, the transport/network layer (ACR-NEMA 1985,p. 6). The ACR-NEMA standard point to point interface does not require the network layer provided functions. Again, the ACR-NEMA standard defines the protocol to provide the transport/network layer functions.

Session Layer

The OSI model session layer provides a mechanism for controlling communication between presentation entities. This includes establishing and using end to end connections (Stallings 1984, p. 49). The ACR-NEMA standard does not specify a session layer as defined by the OSI model. The

standard leaves session layer implementation to the vendor.

Presentation and Application Layers

The presentation layer of the OSI model gives data transformation services. This layer does data formatting, translation, and syntax modification. The application layer, applications using the lower layer communication functions, generates and receives information. (Stallings 1984, pp. 49-50).

The ACR-NEMA defines these layers but only at the message format level. The presentation layer accepts data from the application layer and reorganizes the data into the ACR-NEMA standard defined groups and elements. Again the standard leaves these layers to the vendor to implement.

Functional Description

The detailed functional description of each layer is presented here. The description starts at the lowest layer, the physical layer, with hardware protocol discussion. Since the layers above the physical deal mainly with protocols, the upper layer protocols section presents each layer's protocol starting at data link layer and moving to higher layers.

Physical Layer Functions

This section presents the physical layer hardware and protocol. The hardware aspects include signal definition and timing, physical connectors and cabling. The protocol defines interface data transfers and provides an interface state model.

Interface Signal Definition. Shown in Figure 5 are the interface signals. Sixteen data lines and one parity line define a 16 bit parity checked data transfer bus. Three crossed control line pairs define 6 required bus control signals. The STBO/STBI pair constitute an outgoing data ready and an incoming data accepted handshake signal pair for validating bus data.

The pair REQO/REQI is for interface data transfer control. A device wishing to transfer data issues REQO. A device detecting REQI must listen for interface data. REQO and REQI active simultaneously indicate a collision and the devices wait a collision wait time. The collision wait time is an uniformly distributed time between 0 and 15 usec.

The signals INTO/INTI define a data transfer interrupting, output and input pair. A device indicates transfer halt by activating INTO for 1 usec. The other side detects INTI and both interface sides stop any data transfer related activity.

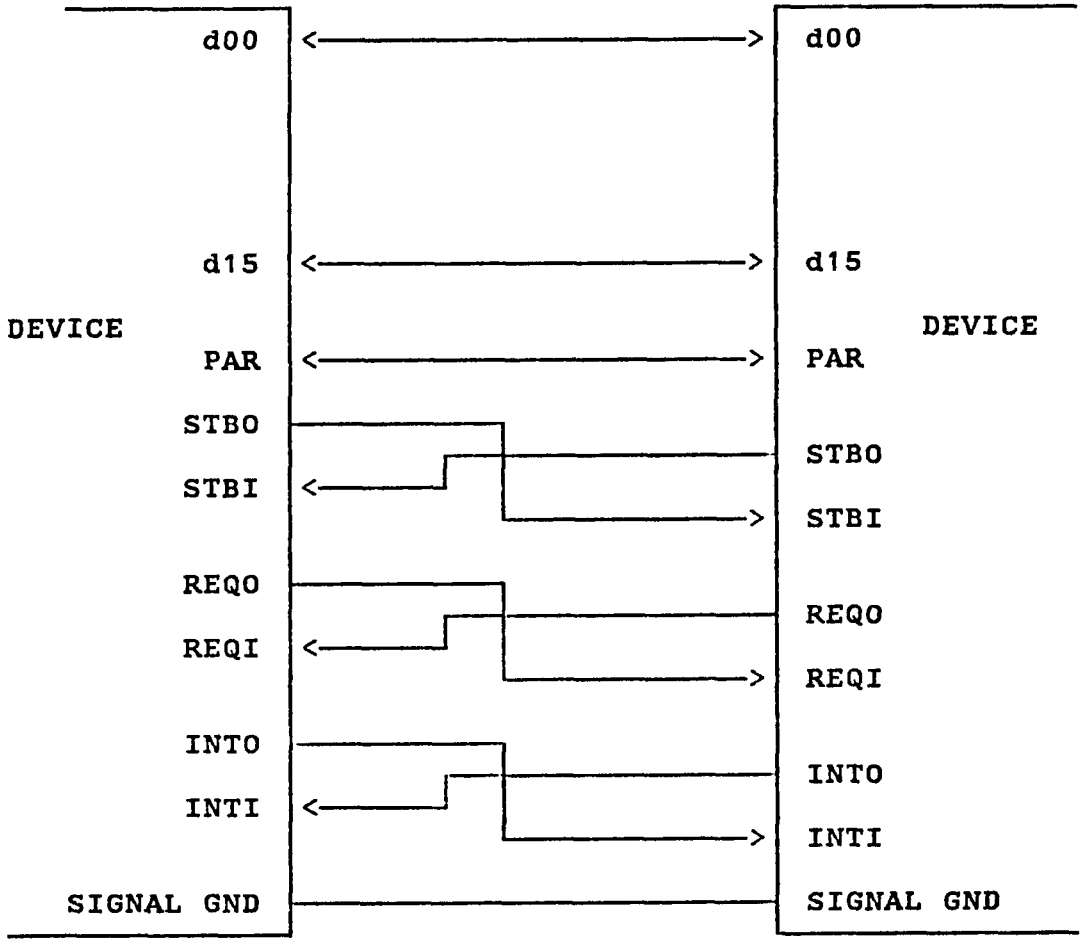

Figure 5. The ACR-NEMA Physical Interface

Interface Timing. Interface timing is uncomplicated, requiring minimum data set-up time of 70 nsec. The collision arbitration window definition is 1 usec plus or minus 250 nsec. Minimum times on the other signals are 0 nsec. The 1 second maximum times define an inactivity timeout, beyond which successful transmission is not assured (ACR-NEMA 1985, p. 28).

Cabling and Connectors. Electrical connection uses differential twisted pair for each data and control signal. The data bus and parity bit require differential tri-state drivers and receivers. Control circuits require a differential driver and receiver. Control circuit termination is with 120 ohms across the differential pair at the receiver side. Both data bus and parity circuit ends require 120 ohm termination resistors. The standard requires cables having a 120 ohm nominal impedance. Additional electrical characteristic details are not covered by the ACR-NEMA standard but are found in the EIA 485 specification (ACR-NEMA 1985, p. 28).

The 50 pin micro-type connector, is 2.530 inches long, with 25 pins per side located on .085 inch centers. Equipment connector is a female receptacle type. The cable, with corresponding male connector, has three control circuit three pairs crossed as shown in Figure 5.

Physical Layer Protocol. The state diagram in Figure 6. defines physical layer protocol states and valid state entry and exit (ACR-NEMA 1985, p.33). Binary values assigned to each state represent the control signal states in the order INTO, INTI, REQO, REQI, STBO, STBI.

Beginning in idle state, REQI detected enters rx_ready state and the interface is receive data ready. The STBI and STBO signals govern transition through rx_dav, rx_dak, and rx_compl states. Frame completion, indicated by REQI deactivating, returns the interface through release state to idle state. An INTI detected during transfer moves the interface to tx_int state, to release state, and back again to idle state.

The Data transmission sequence is similar, but REQO asserted moves the interface into tx_reqwt state. With REQI not detected, indicating a collision, the interface starts transfer moving through the tx_ready, tx_dav, tx_dak, and tx_compl states. The interface detecting a collision, enters collision state, and looks for REQI. If REQI is undetected, the interface re-asserts REQO, starting a transfer.

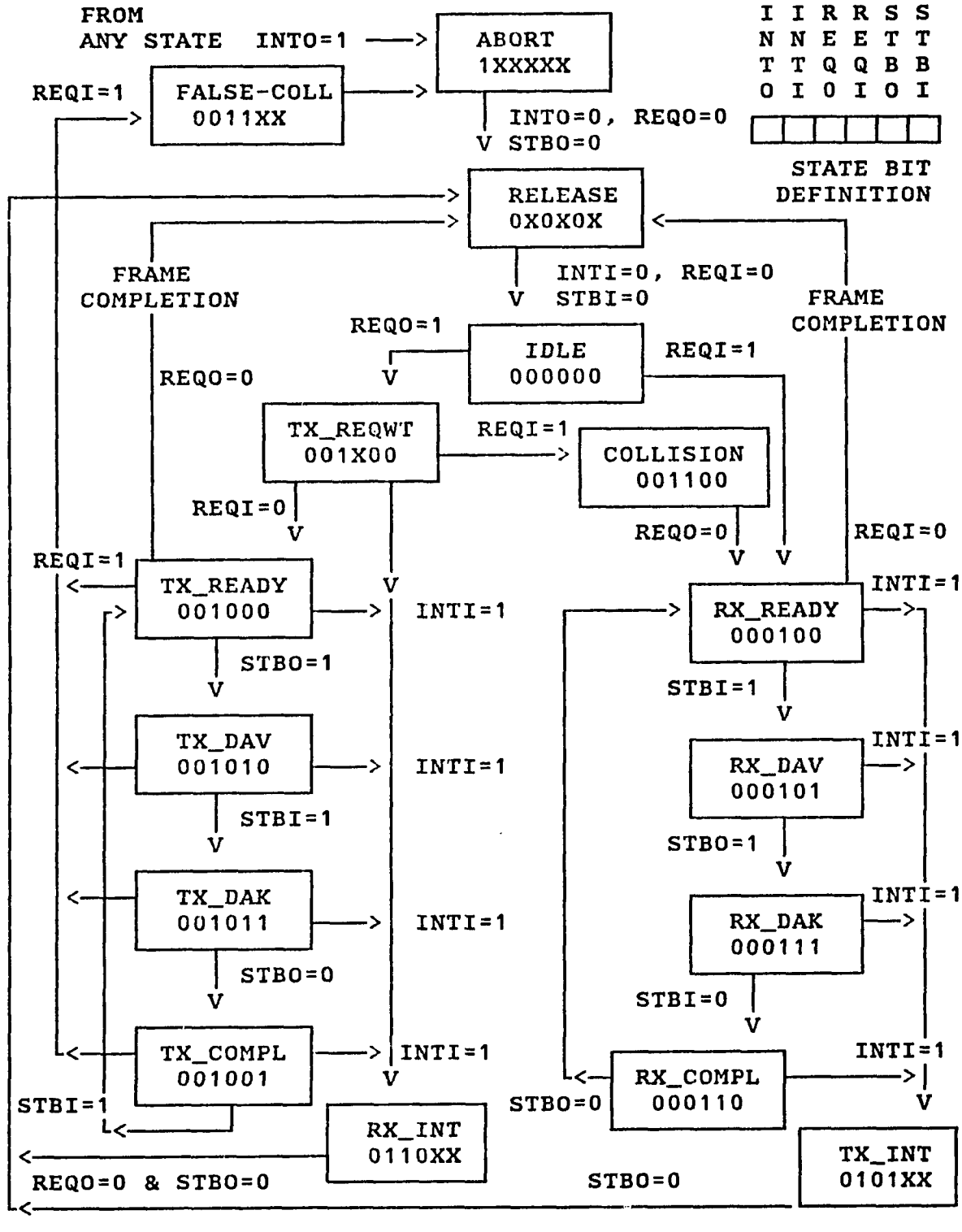

FIGURE 6. ACR-NEMA PHYSICAL LAYER STATE DIAGRAM

Upper Layer Protocols

Session layer connection control words and source/destination addresses control virtual channel end to end messages. Session layer information passes to the transport/network layer where fragmentation, packet descriptor word, and block number adding makes data packets. The data link layer frames the delivered data packets with a frame descriptor word and frame check sequence by the data link layer. The data link layer also handles interface status and error reporting. The frames sent to the physical layer pass across the physical interface. The protocols of these layers above the physical layer are now presented from the lowest level, data link layer, up to the session layer.

Data Link Layer Protocol. A beginning frame descriptor and an ending frame check sequence added to the transport/network layer's data packets by the data link layer make a frame for physical layer transmission. The frame descriptor word's frame type and parameter fields are for interface flow control, status, and error frame retransmission. The frame check sequence is for frame transmission error detection.

In the frame descriptor word, two main frame types, status and control, each with four frames, exist. The control frames are data, data/ack, echo request and echo indication. The status frames are acknowledge,

not-acknowledge, status request, and status indication. The frame descriptor word, parameter field, is the frame word count for control frame types or the frame status for status frame types.

Containing a data packet, the data frame is neither acknowledged or not-acknowledged. Also containing a data packet, the data/ack frame receives a status acknowledge frame if transmission is error free. The error response is the status not-acknowledge frame. A status request frame response is a status indication frame with the current interface status. The echo request frame contains a packet returned in an echo indication frame.

The ending frame check sequence's use is checking received data validity. The ones complement of the truncated binary sum of all words received before the frame check sequence is used. Although the standard only requires error detection, the frame check sequence with word parity values allow single-bit error correction.

The data link layer handles parity and frame check sequence detected data errors. A not-acknowledge frame is the reporting mechanism. Resolving problem conditions also uses status request and indication frames. The data link layer reports unresolved problems to higher layers.

The receiver interrupting the sender accomplishes flow control. The sender response is a status request, by

which the receiver indicates a buffer full condition. When the receiver indicates a normal condition transmission resumes.

Transport/Network Layer Protocol. Session layer messages and control sent to the transport/network layer are fragmented, and with packet descriptor word and block sequence number added, become packets. Fragmentation breaks the higher level message into a maximum 2048 word block. In the packet descriptor word are three fields, channel number, packet type, and service class. The receiving transport/network layer uses the block sequence number to reconstruct the original message in the correct order. It contains a 15 bit sequence number and a 16th bit, when set, indicates the last message segment block.

Transmitting transport layer communicates to receiving transport layer across virtual channels. These numbered channels are opened, closed and reset. Data and command messages move across the channels. The channel number field in the packet descriptor word define the virtual channel the packet belongs to.

The packet descriptor word, packet type field, specifies packets as data, command, or control. The data type packets contain up to 2048 words of upper layer message data blocks. Command packets contain upper layer message commands or command responses. Six control packets, reset

channel request and indication, close channel request and indication, and open channel request and indication, provide channel control. The sender makes a request and if the receiver can provide, makes an indication response. Two additional control packets, pause request and resume request, provide transport/network layer flow control. A receiver halts a transfer with pause request to the sender. Then the sender waits for the receiver resume request restarting the transfer.

The last field in the packet descriptor word, service class, dictates the data link layer service type. Without data link acknowledgment service indicated, the data link layer uses data frames for packet transfer. Acknowledgment service requests the data link layer use data/ack type frames.

Transport/network layer handle errors by sending reset channel request or close channel request packets. The layer reports problems and actions to the next layer.

Session Layer Protocol. The session layer handles end to end message connection. When the session layer instructs the transport/network layer to open a virtual channel it includes the source and destination process in the control packets. This information, generated by the session layer, is the connection control word and destination/source fields.

The session layer uses five connection control words, connect request and indication, disconnect request and indication, and continue. The transport/network layer control word response is opening, closing or resetting a virtual channel.

The destination/source field contains 8 logical address words, and 8 network address words for the destination and 8 logical address words and 8 network address words for the source. This allows for destination and source process definition flexibility.

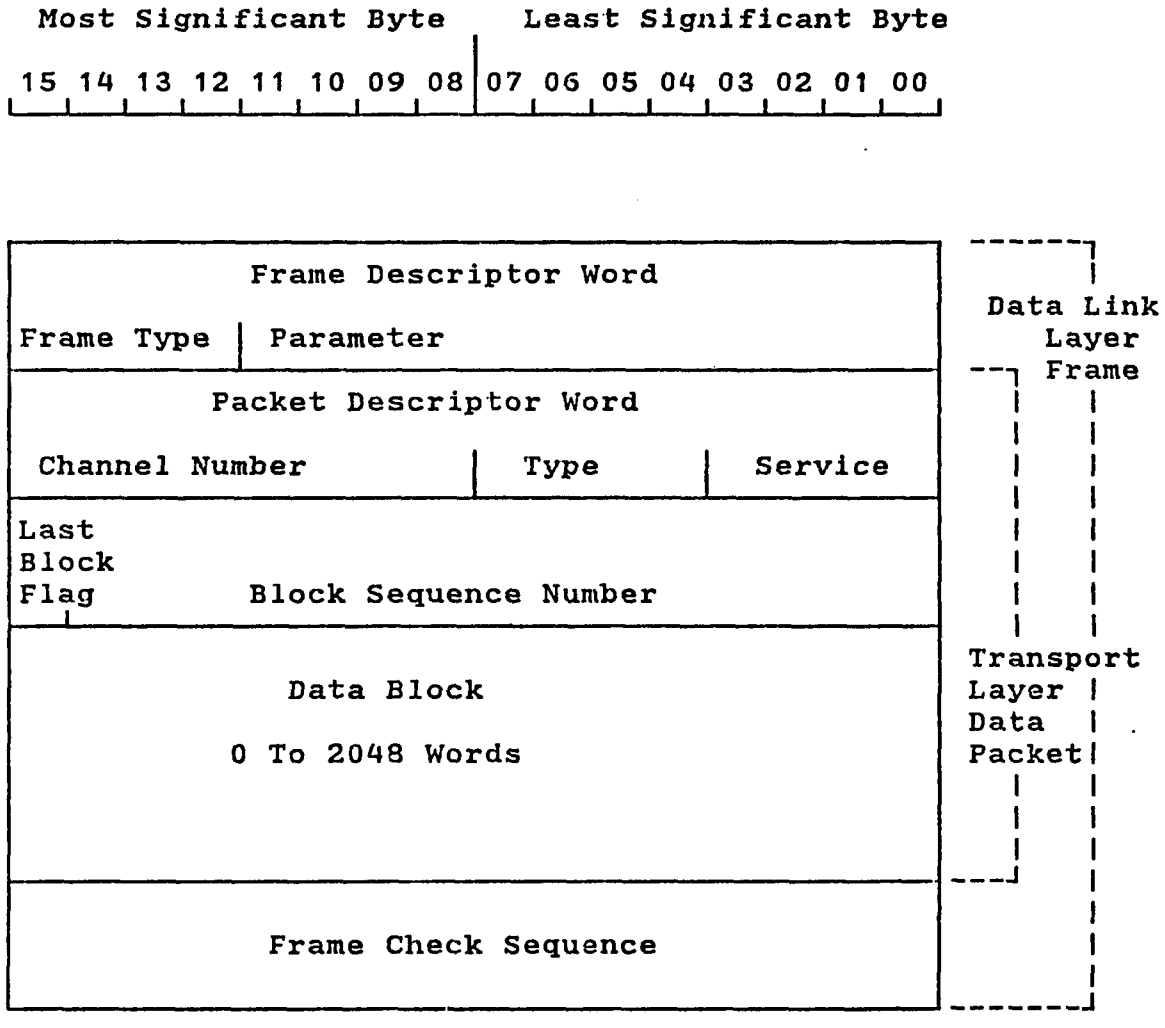

Figure 7. ACR-NEMA Data Frame and Packet

Performance Specifications

Performance specifications complete the ACR-NEMA standard discussion. The standard defines error rate, data transfer rate and environmental specifications. These specifications are summarized here.

Error Rates

The total error rate on the 16 data and 1 parity bus lines can not exceed 17 x 10E-9 before any correction. The standard specifies no error correction algorithms.

Data Transfer Rates

The maximum data rate is 8 mega-bytes per second. The asynchronous handshake protocols allow slower data rates.

Environment

The specification defines no environmental requirements. This is left to the individual equipment manufacturer.

HIGH SPEED FIBER OPTIC NETWORK INTERFACE DESIGN

This section describes the High Speed Fiber Optic Network Interface (HSFONI) design. First the basic design objectives lay out basic design ground rules, speed, technology and functions. Next, the design approach discusses steps taken in the design process. Then design step detail presentation documents the design.

Basic Design Objectives

Having a large design implementation impact, three major design objectives, function, speed, and technology, require early definition. First, addressing the system functional requirements defines a design direction. The functional requirements dictate the system and internal speeds and ultimately determine the required technologies. This section also discusses the HSFONI advantages.

Functions

The HSFONI replaces the ACR-NEMA point to point electrical connection physical layer between medical imaging equipment with a fiber optic link requiring no upper layer modification. Each interface end attaches up to 8 ACR-NEMA standard devices and multiplexes two of these into the fiber cable. Speed considerations allow 1 or 2 of the 8 devices

communicate to a paired receiver on the other interface end. Allowing fiber optic transmission requires serial data, contrasting the ACR-NEMA parallel interface. HSFONI uses full duplex serial communication with separate send and receive paths.

Speed

The ACR-NEMA standard defines a 8 mega-byte per second maximum data rate. Accommodating 2 channels operating at maximum rate requires at least 16 mega-byte per second bandwidth. The ACR-NEMA two byte word, 16 data bits and 1 parity making 17 bits, encodes into two 11 bit transmission channel bytes. Using 11 bit bytes the serial speed is 176 mega-bits per second (Mbps)

Evident with system speed defined are two internal speeds. The HSFONI requires two main sections, a parallel section and a serial section. The parallel section, receiving ACR-NEMA standard data of two channels, operates at 16 MHZ. The interface operates with 17 bit words, so using 2 byte data paths allow operation at half this speed or 8 MHZ for 2 channels. Further reducing speed by providing 2 byte data paths for the two incoming channels allows the speed reduction by half again to 4 MHZ. The serial section can not use dual data paths, requiring a speed of 176 MHZ.

Technology

With functional and speed requirements known, the next step chooses the technology for implementation. Supporting the large serial data rate, and reducing cabling requirements, the physical link uses fiber optic technology. The fast serial section speed requires ECL logic support. The slower parallel section speed can be implemented in bipolar TTL which performs at slower speeds. The design uses any required functions available in vendor VLSI chips.

Advantages

The HSFONI advantages are numerous. Optical fiber cost reductions makes fibers available today cheaper than their wire equivalents (Palais 1984, p.20). This is significant because of fiber advantages over metallic cables. In this design a small optical fiber pair replaces two of the large, 23 pair plus ground, ACR-NEMA metallic cables. When compared on a cross-sectional area basis, the bandwidth of fiber is significantly larger than metallic cables. The fiber is small and light weight allowing easier installation. Because fibers do not radiate the energy within them, they are more secure than metallic cables.

Design Approach

With the basic design objectives defined, system design can start. Meeting the objectives requires a structured, top down design method. Since the system design revolves around the fiber optic link parameters, the fiber optic research section defines them first. Next, the overall system design incorporates the fiber optic link definition. The system definition and operation section discusses this. Now system division into small functional blocks allows circuit design implementing the functions. This is the subject of the functional block definition and operation section.

Fiber Optic Research

This first design step investigates characteristic fiber optic communication properties. In this section operation, types, and performance parameters of light guides are considered. Discussion includes light source emitter and light detector types and characteristics. This information helps choose fiber, source, and detector, to meet the basic design objectives.

System Description

Now the system, including the fiber link and ACR-NEMA device attachment, require detailed operation specification. This step identifies major functional blocks

and specifies their interfaces. This lays out the system architecture requirements for subsequent circuit design.

Functional Block and Circuit Description

In this step HSFONI is implemented with standard logic gates. Detailed attention is given to signal levels and timing. Circuits are designed so functional block inputs and outputs meet interface requirements defined in the system description.

Fiber Optics

With basic design objectives outlined, fiber optic link specification is made first, with HSFONI system design around it. Any fiber optic link design must consider the three major components, source emitter, detector, and fiber. Once selected, system rise time budget and optical budget calculations complete the analysis.

Fiber Optic Wave Guides

The fiber used determines the main fiber optic system characteristics. Because of their importance, presented first are fiber considerations.

Optical fiber construction is with a central core, index of refraction N1, surrounded by a cladding with index N2. The index of refraction relates the speed of light in a material to the speed in a vacuum. When a light wave passes from one index to another, reflection and transmission take

place. An incident light wave at some critical angle into the core is completely internally reflected at the core to cladding interface. The light trapped inside the fiber propagates down the fiber. For the critical angle to exist, core index, N1, must be greater than cladding index, N2.

Multi-Mode Step Index Fiber. Today three different techniques, yielding three different fiber types generate required index of refraction change. The first type is multi-mode step index fiber. This fiber's distinct core index and a different distinct cladding index lead to a step difference at the core to cladding interface. The fiber's different light paths or modes, allow different light travel rates down the fiber causing pulse spreading. This fiber type is oldest and was the cheapest. However, graded index fiber is replacing it at about the same cost.

Multi-Mode Graded Index Fiber. The second fiber type is multi-mode graded index fiber. In this fiber, core and cladding indices at the core to cladding interface are equal with the core index graded larger towards the core center. Although there are multiple modes in this fiber also, the graded index of refraction reduces the number of modes from the number in step index. This reduces pulse spreading allowing greater bandwidth. Usage of these fibers is common today.

Single-Mode Step Index Fiber. The latest fiber

development is single-mode step index fiber. Using a step index at the core to cladding interface as before, but reducing core size yields a 5-10 um core compared to a 50-200 um core diameter. This allows a single propagation mode only. This development makes claims of no pulse spreading or attenuation loss. This fiber is becoming more available but is expensive.

Attenuation. Signal attenuation is a major factor in communication system design. All receivers require a minimum level input power, so transmission losses limit total path length. Likewise attenuation in fiber systems is a limiting factor. This section discusses glass fiber losses which are classified as absorption, scattering and geometric effects (Palais 1984, p. 96). Vendors make fibers from both plastics and glasses. This section does not discuss plastic fiber because their excessive losses cannot meet system performance requirements.

Even purest glass absorbs heavily within specific light wave lengths because of intrinsic absorption, a natural glass property. Intrinsic absorption is very strong in short wave length ultraviolet light. In the longer wave length infrared, intrinsic absorption also occurs. This leaves a low intrinsic absorption window between .8 - 1.7 um wave lengths (Senior 1985, p. 65). Glass impurities are another major loss source. Typically two impurity types are

transition metal ions and OH ions. The OH ions are less of a problem with newer fibers as manufactures learn how to keep water out of drawn fibers.

Scattering losses are due to Rayleigh Scattering and material inhomogeneties. Local variations in refractive index from objects smaller than the light wave length cause Rayleigh Scattering losses. Losses also occur from larger objects introduced during glass manufacture. Imperfect chemical mixing and dissolution cause material inhomogeneities.

Optical fiber bending cause geometric effect losses. Large radius bends do not affect light signal because they appear to light as a straight line. However as the radius becomes smaller than the critical angle, light is no longer internally reflected generating a large loss.

The absorption and scattering losses due lead to three windows where fibers have very low attenuation (Belden 1985, p. 6). The first is at 0.8 - 0.9 um, the second at 1.1 - 1.3 um, and the third at 1.5 um. This is significant because source emitters and detectors operate with these light wave lengths. Losses in the second two windows are less but source emitters and detectors are more expensive.

In fiber selection, attention to attenuation losses is important. Fiber specifications typically define these losses in decibels/kilometer (db/km).

Pulse Distortion. The second fiber optic link limiting factor is pulse distortion. Pulse distortion limits optical link bandwidth and path length. Light velocity variations in the fiber cause pulse distortion. Three factors, material and wave guide dispersion, and multi-mode pulse spreading contribute to signal degradation. Effects on link performance follow.

Dispersion is velocity variation with wave length (Palais 1984, p. 51). Material property caused velocity variation is material dispersion. Wave guide structure caused velocity variation is wave guide dispersion. Both cause pulse spreading and distortion.

A glass prism separates white light into colors with material dispersion. Each color or wave length travels a different speed in glass, because the glass structure affects differently each wave length, causing different colors to bend at different angles. This happens in fiber because light sources do not emit single wave length light. A light pulse entering one fiber end travels different speeds to the other end causing pulse spread.

Multiple modes or light paths down a fiber cause wave guide dispersion. Refractive index for one mode may be different from another mode even with equal path lengths. This leads to different travel times for equal path lengths, contributing to pulse spreading.

Using a more coherent light source reduces both material and wave guide dispersion. Lasers do this better than LEDs, but are more expensive.

Multiple modes in a fiber contribute to pulse spreading. Each incident ray takes a different path, some longer and some shorter. This leads to different net velocities and pulse spreading.

In multi-mode step index fibers all three pulse spreading mechanisms are present. However, multi-mode pulse spreading by far contributes most. Typically material and wave guide dispersion contribute only 2% of pulse spreading. With this type fiber, reducing dispersion distortion using a laser is not effective.

Like in multi-mode step index, the loss mechanism in multi-mode graded index is multi-mode pulse spreading. Graded index fiber reduces pulse spreading by a factor of 100 to 150. In the first window .8-.9 um a LED negates advantages gained with graded index because of LED wave length variation. This requires using a laser. In the second window, 1.1 - 1.3 um, dispersion caused pulse spread is small, allowing LED use.

In single-mode step index fiber, only a single mode propagates so signal distortion is only from dispersion. Using a 1.3 um laser further reduces distortion. A laser produces a narrower source wave length, and at 1.3 um

dispersion losses are negligible. This allows building long, high data rate systems in this configuration.

As with attenuation losses, fiber data sheets specify pulse spreading. The units MHZ-km specify this loss mechanism. The frequency, MHZ, is 1/ts where ts is the pulse spreading time in nanoseconds.

An interesting note here: like all things in nature, everything is a trade off. Pulse spreading is smallest at 1.3 um and attenuation loss smallest at 1.5 um.

Numerical Aperture. An important fiber characteristic is the ability to collect light over a wide incident angle range. Numerical aperture (NA) measures this and core and cladding refractive indices, N1 and N2, determine it. These define an acceptance cone and the fiber only accepts light at a smaller incident angle. A high NA fiber gathers more light than a low NA one.

Now another conflict requiring a trade off decision is clear. A large acceptance cone fiber, NA large, also has greater pulse distortion due to modal pulse spreading. This is because both NA and modal pulse spreading depend on fiber core and cladding refractive index difference. Larger index difference makes larger NA and more light into the fiber. Larger index difference also allows more fiber modes for more pulse distortion (Amp 1982, p. 24).

This problem produces more light source to fiber matching requirements. LED's emit light in all directions, a Lambertian emission pattern (AMP 1982, p. 95). Lasers emit in a narrow cone pattern. Because higher performance, low distortion, single-mode step index fiber also has a small NA, a laser is the required light source.

The third optical fiber property for consideration in selecting fiber is NA. This is because NA along with attenuation and pulse distortion determine light source type. Like attenuation and pulse distortion NA specifications are included in fiber data sheets.

Source Emitters

With optical fiber characteristics understood light source emitters studies follow. Fiber technology uses light emitting diodes (LEDS) and laser diodes because of fiber properties and easy electrical control.

Light Emitting Diodes (LED). The LED is a light emitting pn junction when forward biased. Fiber properties require a narrow light emission angle coupling the source light into the fiber. The LED's wide emission angle, Lambertian emission, require normal LED construction modification, reducing emission angle. A heterojunction LED's construction confines the emission angle. Two materials with different bandgap energies and refractive

indices create barriers so free charges meet and recombine only in a narrow well-defined active layer.

Two light emission types, due to different construction, surface and edge, occur in LEDs. To limit the emission angle on surface emitters an etched well or Burrus construction is used. In an edge emitter, close control over emission region allows a small rectangular region, a few microns by a few tens of microns, reducing emission angle.

Another emission angle reduction technique is a convex lens in the LED package. This focuses divergent LED light into the fiber.

Other LED characteristics include power, current, rise time, life length, spectral width, and output wave length. LED data sheets specify these properties. Optical power defines device output light and determines input current. Optical power, construction type, fiber NA, and LED to fiber interface determine coupled power into the fiber. Rise time, LED step input response current, contributes to system bandwidth. LEDs have a long life time relative to laser diodes but also suffer power degradation with time. LEDS have wide, compared to lasers, optical spectrum, contributing to pulse spreading. LEDs are available which output light from .8 to 1.7 um, the entire fiber operation wave length window.

Some typical LED characteristic values follow. LED power is in the microwatt to low milliwatt range (AMP 1982, p. 100) requiring 50 to 100 ma at 1.2 to 1.8 volts (Palais 1984, p. 125). Typical LED rise times range from a few nsec to 250 nsec (Palais 1984, p. 126). LED lifetimes are from 100,000 to 1,000,000 hours. LED spectral width is from 20 to 50 nm (Palais 1984, p. 126).

Laser Diodes. The second fiber optic light source type is the laser diode. The laser diode is also a pn junction but modified construction produces laser light. As with all lasers, emitted light amplifies in a laser cavity, producing an intense, narrow spectrum output light. Construction is similar to the heterojunction edge emitter LED, with a very defined cavity for light to travel in. However, laser diode ends are cleaved very precisely to allow constructive reflections off the ends. This causes the light amplification.

Laser diode characteristics, output power current, rise time, life length, spectral width, and output wave length follow LEDs. Except in life length, lasers out-perform LEDs. However, they are more expensive because of being more difficult to produce. Lasers are more temperature sensitive than LEDs. This requires temperature stabilization, adding cost.

Laser output power can exceed 10 mw but requires more input current, 50 to 300 ma at 1.2 to 2 volts, (Palais 1984, p. 134) than a LED. Laser rise times are less than 1 nsec (Palais 1984, p. 136) allowing a greater system bandwidth than LEDs. Early laser diodes life times were a few hours, but improvement increased this to 10,000 hours (AMP 1982, p. 101). Laser spectral width is 1 to 5 nm (Palais 1984, p. 136) much less than LED's, allowing longer links with less distortion due to dispersion in the fiber. Like LEDs, output wave length includes the entire window, .8 to 1.7 um, where fibers operate.

Light Detectors

The third fiber optic system component is the light detector. Source emitter light launches into the fiber and travels the fiber length, for conversion by the detector into an electrical signal. Although vacuum tube photodiodes and photomultiplier tubes exist, most commonly used is the semiconductor modified pn junction photodiode, PIN and avalanche photodiodes (APD).

In a pn junction photodiode the reversed biased junction increases the potential barrier between the p and n regions. Free electrons and holes cannot climb the barrier so no current flows. The incident photon absorbed energy raises the energy on an electron making it free to move. Incident photons from the fiber falling on the pn junction

create a net current flow.

The typical photodiode rise time is microseconds making them unsuitable for fiber systems (Palais 1984, p. 146). Construction modification makes them more suitable. A PIN (p-intrinsic-n) photo diode has the region between the p and n layers increased with an intrinsic material. This increases the probability of absorbing incoming photons.

In APDs construction modification increases response time and amplitude. Large depletion region electrical forces in the p-intrinsic-p-n layer construction amplify the incident light.

Important detector properties are responsivity, spectral response and rise time (Palais 1984, p. 142). Responsivity is the ratio of output current to input optical power. Amperes/watt (A/W) measure this. Spectral response is the curve of detector responsivity as a wave length function. Rise time is the output current change for an input step optical power variation. Diode data sheets specify these characteristics when evaluating detectors.

PIN Diodes. The PIN photodiode is most common in fiber systems. Different semiconductor materials have different band gap energies requiring more or less photons for current flow. This determines PIN diode spectral response. Different wave length windows require different material PIN diodes. Silicon responds to .3-1.1 um wave

length light in first window. Second and third windows use indium-galium-arsnide (InGaAs) with a 1.0-1.7 um wave length response. One material, germanium, responds to all three window wave lengths.

PIN diodes operate in two modes, photoconductive and photovoltaic. In reverse bias junction photoconductive mode, output current is proportional to optical power. Fiber systems work in this mode. Typical responsivitys are around 0.6 A/W (AMP 1982, p. 112). With no reverse bias provided, the diode operates in photovoltaic mode, producing output voltage from input optical radiation like in solar cells.

The third important PIN diode characteristic is rise time. The PIN diode limiting factor is transit time, the time for free charge to traverse the depletion layer, which in the PIN diode is intrinsic layer width (Palais 1984, p. 151). Capacitance, formed by p and n layers separated by an insulating intrinsic layer also slows rise time. High speed photodiodes typically have a few picofarads capacitance (Palais 1984, p. 151). Typical PIN photodiode rise time range from 0.5 to 10 nsec.

Avalanche Photo Diodes (APD). The APD construction is with p-intrinsic-p-n layers. Depletion region electrical forces, greater than in the PIN diode, cause incident photons to create high kinetic energy electrons. These collide with neutral atoms freeing more electrons generating

a multiplying effect. Avalanche multiplication gives the APD a gain of several hundred. APDs are more sensitive than PIN diodes, obtaining better signal-to noise ratios (Palais 1984, p. 153).

These advantages are not without a loss. The APD gain is temperature dependent. This may require temperature stabilization or compensation. The APD reverse bias voltage is several hundred volts (AMP 1982, p. 117). This makes operation with typical semiconductor systems difficult.

APD construction materials are same as PIN diodes. The same spectral response ranges apply to APDs.

Because of internal gain APD responsivity is greater than PIN diodes. Typical responsivities range from 20 to 80 A/W (Palais 1984, p. 154)

Because of similar construction, and similar charge carrier transit time and capacitance, APD rise times are similar to PIN diodes. Typically rise times are a few tenths of a nanosecond (Palais 1984, p. 154).

Component Selection

Now background in fiber optic link component operation allows component selection giving required performance at the best cost. Link fiber, light source emitter and detector selection is determined by system performance outlined in basic design objectives.

Fiber. First selected is fiber. With little cost difference and significant performance increase, choosing multi-mode graded index fiber over multi-mode step index is logical. Link length limits of 1.5 km, sufficient for building to building and intra-building connections, do not need single-mode step index fiber performance. The 200 Mbps operation over 1.5 km requires 600 MHZ-km cable. Dual fiber cable (one transmit and one receive) is available from vendors like Belden, AT&T, and AMFOX. It is a 125 um cladding and 62.5 um core fiber. At 1.3 um wave length attenuation is 2.5 db/km. The numerical aperture (NA) is .29. Fiber comes in 1.1, and 2.2 km lengths allowing a complete 1.5 km connection with splicing or connectors.

Source Emitter. Next chosen is a light source emitter. High cost, low reliability, and greater performance than required, make a laser diode light source a poor choice. A LED meets performance requirements, but at the 1.3 um, second wave length window, because the multi-mode graded index fiber operates more efficiently at this wave length. Available from Siecor is a 200 mbaud transmitter module in a 20 pin dip package meeting requirements. It uses a single power supply and input is ECL 100k compatible. Optical power into a 62.5/125 .29 NA fiber is 18 uw with 150 ma drive. The rise time is 2 nsec and the spectral width is 120 nm (Siecor Transmitter 1986, p.2-3).

Detectors. The light detector choice is the PIN diode. High reverse bias voltage makes the APD a poor choice when not requiring internal gain for light sensitivity. Matching the source emitter requires operation at 1.3 um light wave length. Transmitter matching, 200 mbaud receiver module from Siecor meets these requirements. Output is ECL 100k compatible and is in a 20 pin dip package. This removes receiver amplification circuit design requirements, a significant design effort at these frequences. Since this amplifier is integrated into the package responsivity in A/W is not specified. The receiver responds to light wave length between 1.28 and 1.37 um. The rise time is .7 nsec (Siecor Receiver 1986, p.2-3).

System Verification

Now with components selected, verification shows they meet the performance requirements. Two calculations, bandwidth budget and power budget, confirm performance. These calculations prove selected components will not allow pulse distortion or attenuation degrade system performance below requirement.

Bandwidth Budget. Bandwidth budget calculation determines if the design supports required bandwidth at required length. This determines if the limiting length factor is pulse distortion. Palais has shown (Palais 1984, p. 254).

$$tS^2 = tLS^2 + tF^2 + tPD^2 + tM^2$$

where tS is total system rise time, tLS is light source rise time, tF is fiber rise time, tPD is photodiode detector rise time, and tM is rise time margin.

A reasonable rise time estimate is 70% of the fastest system pulse duration (Palais 1984, p. 256). With transmitter and receiver operating at 200 MHZ the complete link design is around 200 MHZ even though the HSFONI only requires 178 MHZ. System rise time, tS, is 70% of 1/200 MHZ, or 3.5 nsec.

Data sheets specify the light source rise time and detector rise time. From the data sheets, tLS = 2 nsec and tPD = 0.7 nsec.

The last rise time calculation is fiber rise time. Palais has shown that the fiber electrical rise time is equal to its full duration half-maximum pulse spread (Palais 1984, p. 257). This calculation from the data sheet given 600 MHZ-km value and longest run allowed, 1.5 km, gives a value of tF = 1/600 MHZ-km x 1.5km = 2.5 nsec.

Now the bandwidth margin is calculated

$$3.5^2 = 2^2 + 2.5^2 + .7^2 + tM^2$$

or tM = 1.1 nsec. The components selected allow 200 MHZ operation at a cable length of 1.5 km.

As mentioned, total pulse spreading given by the

bandwidth-length product, 600 MHZ-km, is caused by material and wave guide dispersion and multi-mode pulse spreading. For longer than 1.5 km interconnection lengths, substituting a single mode fiber with no multi-mode pulse spreading distortion for multi-mode graded index gives the required performance.

Power Budget. The next calculation, power budget, ensures attenuation losses do not limit system length. The transmitted light power, less any connector , splice, and system losses must be greater than the detector light power required. This is summarized by

$$Ppd = Pls - Ploss$$

This calculation is in decibels (db).

Transmitter power, from the data sheet, is -18 dbm into the fiber. There is no connector or splice loss if the entire 1.5 km length is one fiber. If the system requires connectors and splices, each connector is a 1 db loss and each splice 0.1 db. Fiber attenuation loss, 2.5 db/km, for 1.5 km is 3.75 db. From the above formula, the detector power is -18 dbm -3.75 db or -21.75 dbm. This allows a -12.25 db power margin because the detector operates with as little as -34 dbm according to specification. This allows some connectors and splices if required. Attenuation losses do not limit system length to less than the specified 1.5 km connection length.

Fiber Optic Research Summary

The Siecor LED transmitter module coupled to a 600 MHZ-km multi-mode graded-index fiber and light detected with the Siecor PIN detector module allows link operation at 200 MHZ over 1.5 km. LED wave length is 1.3 um. In greater connection lengths, single-mode step-index fiber replaces multi-mode graded-index fiber.

Finally, modulation code requirements need mentioning. The NRZ pulse train spectrum contains a large and important DC component. The DC signal current partially determines receiver amplifier operation point. A changing DC level changes the amplifiers operation point resulting in the receiver's characteristics drifting and added errors (Palais 1984, p. 212).

Although receiver amplifier is integral to the Siecor receiver module, and specific circuit design not required, the internal amplifier is AC coupled. This places a requirement on the HSFONI transmitter to use a DC balanced code. In this code type "1"'s number equal "0"'s number. Raw data is not sent across the interface without first encoding. The code must also include clocks allowing receiver synchronization to the transmitter.

System Definition and Operation

With the fiber optic link operation defined, system integration starts. The system diagram is in Figure 8. A transmit fiber and receive fiber form the interconnecting fiber optic pair. Each HSFONI end is the same, with interface, buffer, transmitter, receiver, and optical functional blocks.

End "A" interface functional block receives data from ACR-NEMA equipment and puts it in the buffer functional block. The buffer is for speed matching between the ACR-NEMA interface and the fiber optic link. The transmitter functional block does data and link control framing, encoding and serializing. The optical functional block converts the serial bit stream to light pulses and transmits the data across the link for reception by end "B" optical functional block. The receiver functional block retrieves the original data and passes it through the end "B" buffer and interface blocks to the ACR-NEMA equipment.

This section defines required block functions and system operation. It presents functional block interconnection and communication also.

Interface Functional Block

The interface functional block (IFB) attaches to one to eight ACR-NEMA devices to receive data. The IFB arbitrates between devices allowing at most two data

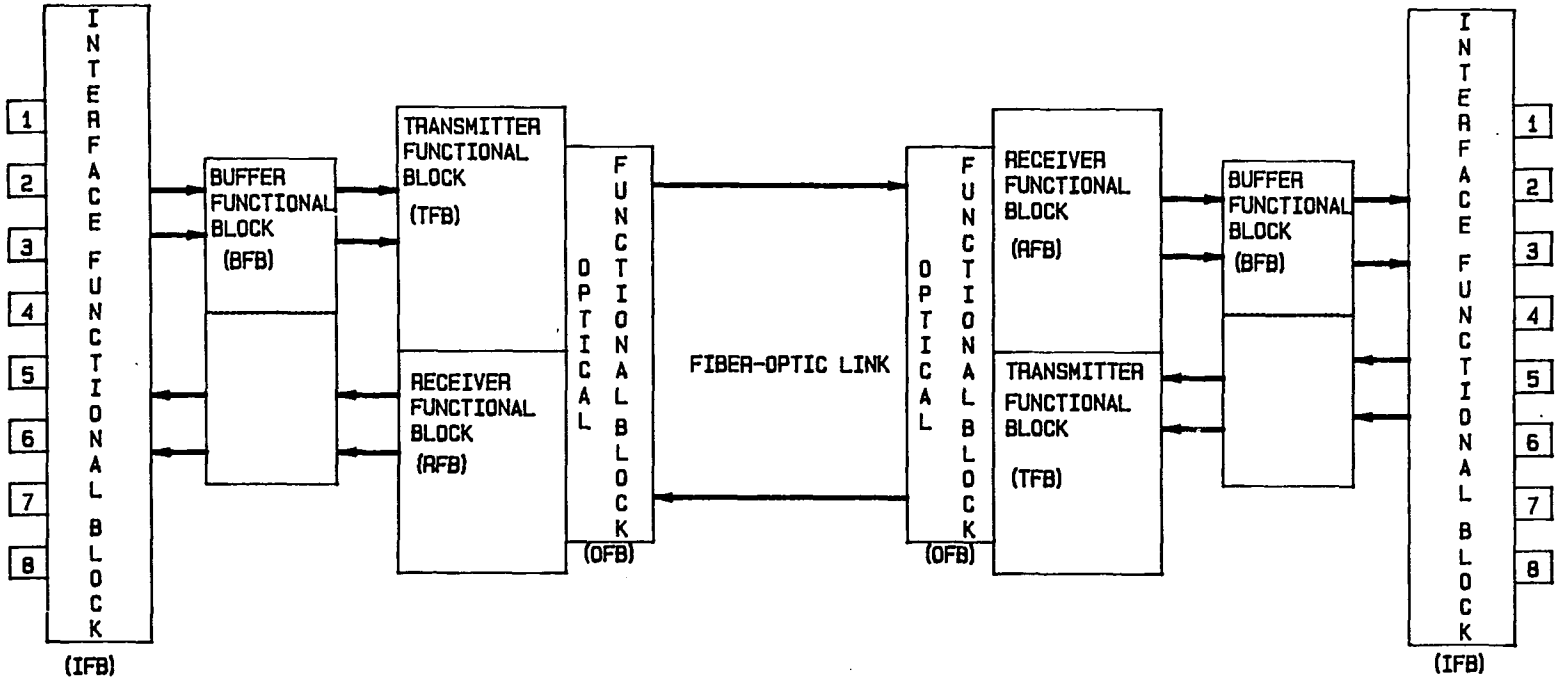

Figure 8. The HSFONI System Block Diagram

transmitters and two data receivers. This allows only two devices sending data in one direction even if less than two devices are sending in the other direction.

Two sixteen bit plus parity buses pass transmit data to the buffer functional block. Two identical data paths deliver receive data from the buffer functional block to the IFB. The IFB acts as a switch attaching and detaching requesting ACR-NEMA interfaces to these data paths. The data paths support a 8 mega-byte per second or 4 MHZ bandwidth. The ACR-NEMA REQI/REQO and INTI/INTO signals pass directly to the receiver or transmitter for control frame generation and detection.

Buffer Functional Block

The buffer functional block (BFB) receives IFB transmit data. It also sends receive data to the IFB. The BFB is really two buffers, a send buffer and receive buffer. Two sixteen bit plus parity data paths deliver buffer data to the transmitter for data frame generation. Two like data paths pass receiver data frame data to the BFB.

The buffer is for speed matching between the incoming ACR-NEMA interface and the outgoing fiber optic link. Although the total BFB input data rate, 16 mega-bytes per second, equals the output rate of 8 mega-byte per second for two interfaces, the buffer accommodates slight speed variations. Also, two transmitting ACR-NEMA interfaces are

unlikely sending words at precisely the same time, but the fiber-optic link requires them simultaneously. The buffer performs this formatting and holding function.

Transmitter Functional Block

The transmitter functional block's (TFB) function is formatting data for fiber optic link transmission. It encodes data into data frames and REQI/REQO and INTI/INTO commands into command frames. The encoded data ensures an adequate number of polarity changes for the receiver's phase lock loop (pll) to remain synchronized and equal amounts of on and off time to ensure the receiver DC balance. The BFB received data is encoded, serialized, and passed to the optical functional block.

The transmitter takes 16 bit data words from the BFB every 62.5 nsec or 16 MHZ. The data is encoded into two 11 bit words for a serial bit rate of 5.7 nsec or 176 MHZ. Realizing this speed requires emitter coupled logic (ECL).

Optical Functional Block

Grouped together into the optical functional block (OFB) are fiber optic transmission, reception, and related functions. The OFB converts the TFB ECL level serial bit stream into light pulses on the fiber optic link. The OFB also contains the hardware to receive light pulses and convert them into an ECL level serial bit stream for the

receiver functional block. The OFB transmit section on this HSFONI end connects to the other HSFONI end OFB receive section. This end's OFB receive section connects to the other OFB end transmit section.

Receiver Functional Block

At the other optical fiber end, receiving transmitted data, is the receiver functional block (RFB). The RFB decodes the OFB ECL serial bit stream into the original 16 bit data word, and passes it to the BFB. The RFB likewise decodes any command frames into the proper command.

Like the transmitter, the RFB accepts a 176 MHZ serial bit rate. and sends decoded data the BFB at 16 MHZ. This speed requires ECL technology at the RFB.

Functional Block Interconnection

A slightly more detailed block diagram of one HSFONI end is shown in Figure 9. This diagram further reduces the IFB further into two parts, interface cards and interface control. It divides the BFB into transmit and receive portions each with two parts, "A" and "B". The TFB, RFB and OFB remain as shown before.

Understanding the interconnection block diagram requires more IFB details. The IFB uses a bus type structure. Two 18 bit, 16 data and 2 parity, transmit buses and two 18 bit receive busses pass data to and from the BFB.

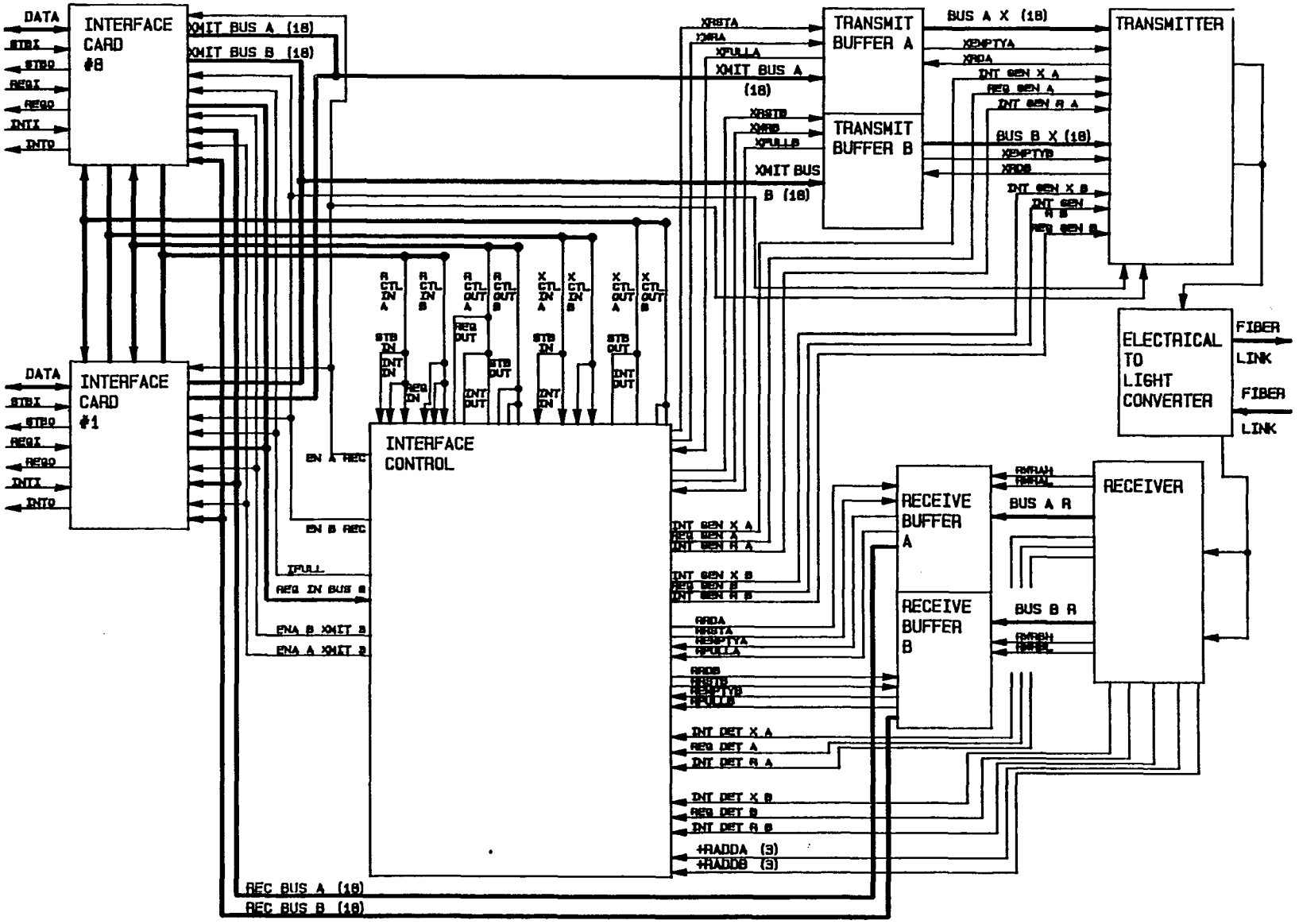

Figure 9. The HSFONI End Interconnect Diagram

Up to eight interface card control signals pass to the interface control section. The interface control section arbitrates, allowing an interface card to send or receive data on one of the available busses.

Each interface card attaches to an ACR-NEMA device. With no data path available, the interface card halts the device with the appropriate ACR-NEMA protocol. This cost effective solution allows interface card addition for expansion as more devices require link attachment.

The IFB interface control section receives the STBI and generates STBO and the BFB XWR to transfer data from interface card to BFB. The BFB RRD causes data transfer from BFB to interface cards. IFB monitors XFULL to not over run the buffer during write, and REMPTY for read over runs. The IFB controls two data paths "A" and "B". Two, three control line sets, go to the transmitte for control signal encoding. The matching two, three control line, sets come from the receiver indicating control signals detected.

The TFB requests BFB data with XRD and sends synchronizing characters if no data is available indicated by XEMPTY being active. RFB indicates valid data to BFB with RWRH and RWRL. Because the receiver can not stop incoming data, BFB over run by receiver, indicated by RFULL, indicates an error condition stopping the transfer and requiring data retransmission. Two differential ECL lines

connect transmitter and receiver to the fiber optic send and receive modules.

The adopted convention reduces signal name confusion. A preceding "X" indicates a transmit side signal. A "R" indicates receiver side. Two separate send and receive paths exist. Differentiating these paths are an "A" or "B" in the signal name.

Functional Block Definition and Operation

The following sections discuss functional block detail and circuit operation. Each functional block is further partitioned down until the partition requirements are relizable with standard TTL logic gates.

Interface Functional Block

As mentioned, the IFB is two main sections with an interconnecting bus. The first section, the interface card, repeated up to eight times, plugs into the interface control section controlled bus, and provides physical connection to ACR-NEMA devices. The interface card section followed by the interface control section are now discussed.

Interface Card. The interface card, with three logic sub-sections, is a programmable switch controlled by interface control section. Figure 10 shows the block diagram. Four, three bit control busses from the interface control

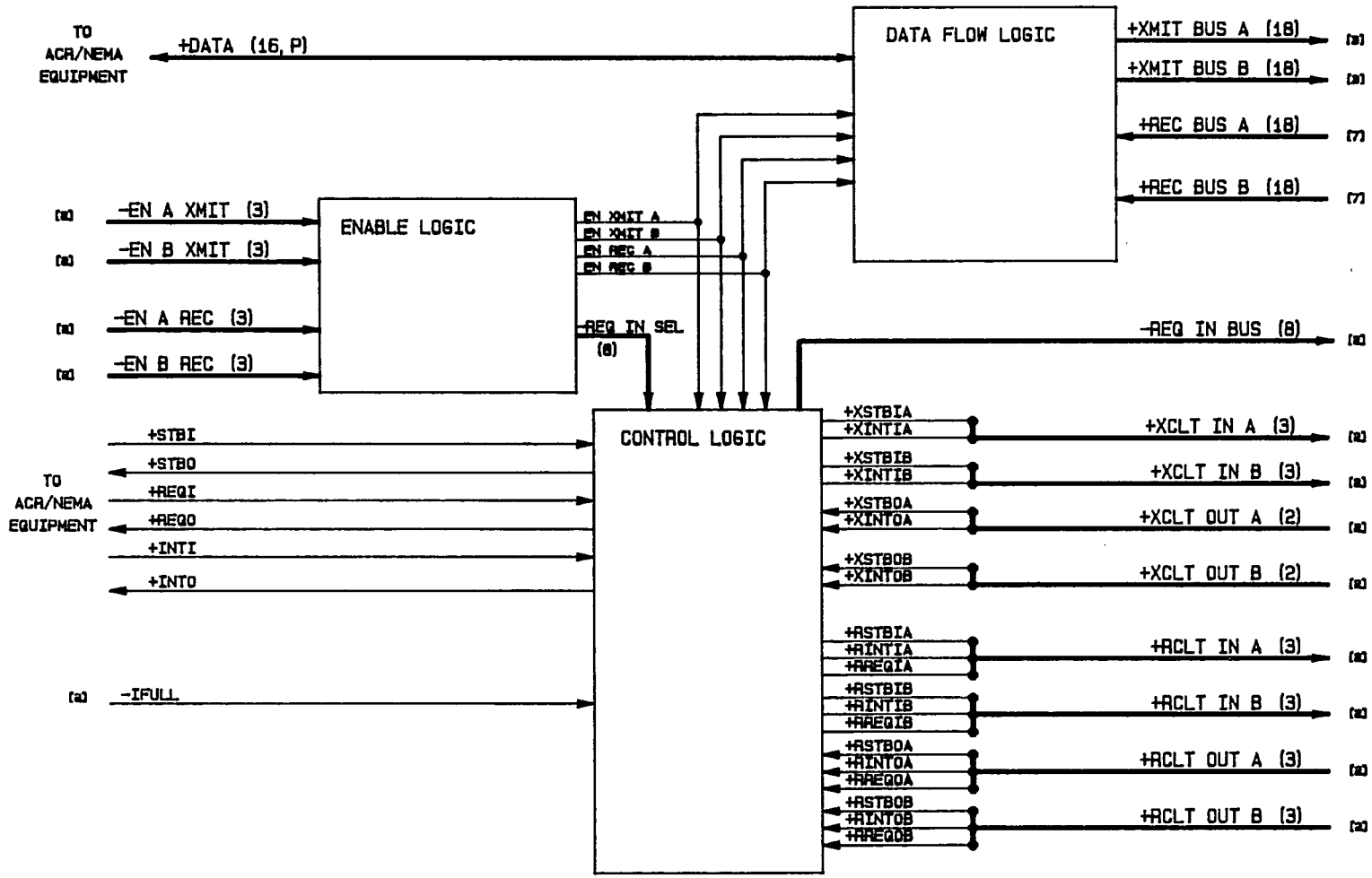

Figure 10. Block Diagram 1-IFB Interface Card

section, -EN A XMIT, -EN B XMIT, -EN A REC, and -EN B REC enter the enable logic. The busses contain a three bit enable address allowing a card to transmit or receive data. The two "XMIT" busses allow the ACR-NEMA connected device's data to transmit across the +XMIT BUS A or +XMIT BUS B to BFB, and inbound and outbound control signals pass across +XCLT IN A and +XCLT OUT A or +XCLT IN B and +XCLT OUT B as determined the address being on the "A" or "B" enable bus. Likewise the two "REC" busses allow passing BFB data on +REC BUS A or +REC BUS B and passing inbound and outbound control signals on +RCLT IN A and +RCLT OUT A or +RCLT IN B and +RCLT OUT B to connected ACR-NEMA equipment.

The enable logic compares a dip switch set, three bit, card address to the enable bus's address. This card address passes to the control logic for -REQ IN BUS generation. A match detected is indicated by the enable logic activating one output signal -EN XMIT A, -EN XMIT B, -EN REC A, or -EN REC B. These signals input to data flow logic and control logic.

Shown in Appendix A Figure 23 is an enable logic TTL realization. When installing a card, a unique 3 bit address is set in the dip switch. This allows using one card type in all eight card locations. Four, three 7486 exclusive or gate sets compare the dip switch address with the four enable bus addresses, and activate the corresponding enable

if an enable bus address matches the dip switch. A 74138 decodes the dip switch address and passes an eight line select bus to the control logic for placing the +REQI on the correct -REQ IN BUS bit.

The interface card data flow logic gates the bi-directional 16 bit plus parity ACR-NEMA data bus on or off one of four BFB data paths. The data paths are 18 bit busses instead of 17 if in future expansion parity checking is across 8 bits instead of 16. Gating control is from four enable logic input signals.

In Appendix A Figure 21 is a typical data flow logic implementation. The bi-directional ACR-NEMA data bus is gated to +XMIT BUS A or +XMIT BUS B through one of two sets of 75175 modules, or driven by +REC BUS A or +REC BUS B through the 75174 modules. The 75174 and 75175 modules convert the TTL signals into ACR-NEMA standard required differential EIA 485 level signals. The signal +EN XMIT A, +EN XMIT B, +EN REC A, and +EN REC B, control the module direction.

The interface card control logic handles inbound and outbound ACR-NEMA control signal gating onto or off the appropriate control path. A device wishing to transmit data activates +REQI and lines corresponding to the card address activate on the 8 bit -REQ IN BUS. This allows the interface control section know which interface card is

requesting a data path, and the interface control section returns the card address to the enable logic. With a data path selected, the interface card control logic grounds +REQO. The remaining ACR-NEMA input signals +STBI and +INTI, are gated to +XCLT IN A or +XCLT IN B, and output signals +STBO and +INTO gated to +XCLT OUT A or +XCLT OUT B selected by incoming signals from the enable logic. The interface full signal, -IFULL, indicates from the interface control all data paths in use. If the interface card receives +REQI from a connected device, it immediately generates a +REQO, stopping the device from starting data transfer.

The control logic handles attaching devices when receiving other side HSFONI data from a device. If incoming enable logic signals indicate, the control logic gates the three inbound control signals to +RCLT IN A or +RCLT IN B and outbound control signals to +RCLT OUT A or +RCLT OUT B.

Appendix A Figure 22 shows interface card control logic using TTL gates. A 75175 receives three EIA 485 level differential ACR-NEMA standard control signals. Logic gates these signals to +XCTL IN A, +XCTL IN B, +RCTL IN A, or +RCTL IN B according to enable lines active. HSFONI interface control generated signals are gated through a 75174 driver to attached ACR-NEMA devices.

Interface Control. The interface control section

contains 12 logic sub-sections. However four sub-sections are copies allowing for multiple data paths. No data passes through the interface control section, only control. The block diagram is in Figure 11.

A transmit request starts with an active -REQ IN BUS bit detected by the ARB logic. The ARB logic signals the interface card the data path available, and sends a REQ frame generation request to TFB. XMT STB logic A or XMT STB logic B handle STBI/STBO handshake and placing data in the BFB. XMT CTL logic A or XMT CTL logic B handle interface states and unusual conditions indicated by remaining control lines INTI/INTO.

In reference to Figure 6 on the state diagram receive side, the ARB logic detects active +REQI moving the ACR-NEMA interface from idle state to RX_READY state. XMT STB logic handles moving through RX_DAV, RX_DAK, and back to RX_READY. The XMT CTL logic monitors the interface and handles interrupts causing TX_INT and release state entry. The +REQI deactivation releases the interface card, returning the interface through release to idle.

When this HSFONI side receives data, the receiver signals frame recognition by activating +REQ DET A or REQ DET B. The REC CTL logic A or REC CTL logic B signals the REC BUS CTL logic to select the correct interface card

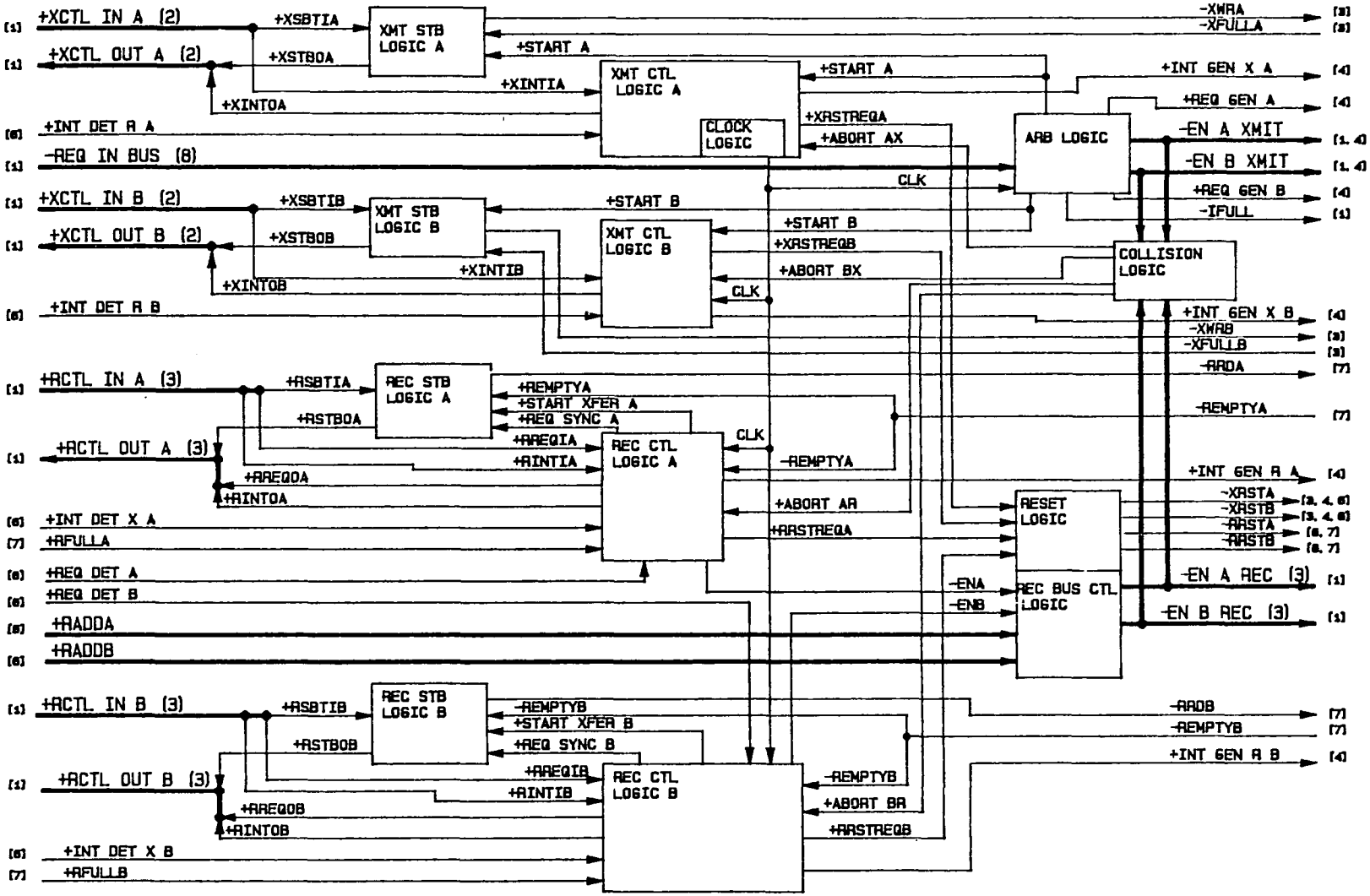

Figure11. Block Diagram 2-IFB Interface Control

and controls the interface states. STBI/STBO data transfer handshake of BFB data is under REC STB logic A or REC STB logic B control. The collision logic ensures the same interface card is not simultaneously selected by receive and transmit sections.

To transfer this HSFONI side received data to ACR-NEMA equipment uses Figure 6 ACR-NEMA interface state diagram transmit side. The interface control receive state machine starts when RFB actives +REQ DET A or +REQ DET B causing +REQO activation, and TX_REQWT state entry. If +REQI in not activated after a specified time the +Start XFER signals TX_READY state entered and the REC STB logic handles data transfer through TX_DAV, TX_DAK and TX_COMPL states. For any unusual conditions, REC CTL logic handles transition through RX_INT and Release states.

The XMT CTL and REC CTL logic state machines run off a master clock generated in XMT CTL logic A. The control signals entering these logic areas require synchronization to these clocks. However meeting the 16 mega-byte per second speed requirements, the XMT STB logic and REC STB logic areas run asynchronously.

The reset logic handles interface reset on power up. It also handles resetting the interface to a known state after detecting an interface error.

The -REQ IN BUS from attached interface cards enter the ARB logic. The ARB logic assigns a priority to interface card location number. Location 8 is highest decreasing in priority to location 1. With no interface cards previously selected, the ARB logic enables the highest priority card requesting a data path with its three bit address on -EN A XMIT. The ARB logic masks the corresponding -REQ IN BUS bit, and with another card request, the ARB logic places the second card's address on -EN B XMIT. The -IFULL signal becomes active allowing no other interface cards to request a data path.

The ARB logic signals control path XMT CTL logic A to handle the transfer by activating +Start A. Like wise +Start B starts control path B logic. The ARB logic signals TFB to send request control frames with the +REQ GEN A and +REQ GEN B signals.

The ARB logic schematics are in Appendix A Figure 24 and Figure 25 and the timing diagram in Appendix C Figure 46. A 74273 latches the -REQ IN Bus on +CLK XA rising edge. A second 74273 latches the bus on -CLK XA rising edge, preventing metastable conditions. The timing diagram shows RQ1 latched and feeding a 74148 priority encoder which encodes the 1 of 8 -REQ IN Bus into a three bit binary address. The encoder address feeds back into a 74138 decoder to enable the correct 74125 to drive the +START A signal to

the XMT STB logic A. When, as in the timing diagram, two signals RQ7 and RQ8 are active simultaneously, the priority encoder selects the highest numbered input. A 74126 mask register, enabled by the -ENA A XMIT 74138 removes the request fulfilled by the -ENA A XMIT logic from the +LCH REQ bus. The -ENA B XMIT logic then puts the address of interface card 8 on the bus allowing BFB access. Next, +START B and -IFULL activate, halting any devices from starting a data transfer.

The XMT CTL logic A function duplicates, except the clock generation, the XMT CTL logic B so discussion is of only one block. The XMT CTL logic is basically a state machine starting with unusual condition detection. An -INTI detected at this HSFONI side, a -INTI detected at the other side, or HSFONI transmitter and receiver selecting the same interface card indicated by +Abort, are conditions starting the state machine. The state machine handles generating correct interface timings, indicating the TFB to send an interrupt control frame, and resetting the HSFONI transmitter when recovery is complete.

The clock logic generated by XMT CTL logic A is a two phase clock 90 degrees apart. Other control logic areas use the clock to ensure correct interface timings. The ARB logic uses the clock to synchronize the incoming -REQI.

Appendix A Figure 28 and Figure 29 and Appendix C

Figure 47 show schematics and timing for XMT CTL logic A and XMT CTL logic B. The logic begins monitoring +XINTI, +INT DET R and +ABORT when ARB logic +START becomes active. The +XINTI generates slightly different timing than +INT DET R or +ABORT. Any three start clock divider circuity, three 74LS56's and three 7474's, timing 1 millisecond.

In the +XINTI started case, if +XINTI is still present after the clock divider waits 1 msec, +INT GEN X and +XRSTREQ activate. +INT GEN X requests the TFB to send a interrupt frame, stopping the other HSFONI end receiver. The +XRSTREQ causes reset logic to return a reset pulse, resetting associated transmit logic.

In +INT DET R or +ABORT started case, +XINTO generated for 1 msec causes +INTO to stop the ACR-NEMA device. At the end of 1 msec, +XRSTREQ generated resets the logic.

Like the XMT CTL logic, the XMT STB logic B is a duplicate of the XMT STB logic A. The +Start signal from the ARB logic tells the XMT STB logic it can proceed with a data transfer. The delayed +XSTBI input generates the +XSTBO output supporting ACR-NEMA data transfer handshake. The -XWR validates data to the BFB, and -XFULL is monitored and XMT STB logic halts the transfer if it becomes active until the buffer can receive data.

Appendix A Figure 27 is a XMT STB logic A and XMT

STB logic B TTL implementation and the corresponding timing diagram is in Appendix C Figure 48. This circuit driven by +XSTBI, runs asynchronous to any system clock. The rising +XSTBI edge, delayed by two gates, clocks a 7474 latch and the falling edge generated pulse clears the latch. The latch "Q" output is the handshake signal +XSTBO and the "Q" not output is BFB -XWR data valid pulse. The BFB -XFULL signal holds off generating a return +XSTBO until the buffer is no longer full.

The RFB indicates it received a request frame to the REC CTL logic and sends the other end HSFONI transmitting interface card address to REC BUS CTL logic. The logic under REC Bus CTL logic control gates the address onto the enable receive bus, enabling the same interface card on this HSFONI end. Appendix A Figure 33 shows this circuit. A 74240 controlled by -ENA and -ENB re-drives the +RADD bus from RFB.

The possibility exists for the interface control transmit section to try selecting an interface card when the receive section does. The collision logic monitors this by comparing enable buses. If the logic detects this condition it tells both colliding transmit and receive sections to abort.

In Appendix A Figure 26 is the TTL implementation of the collision logic. Four sets of three 7486 exclusive or

gates check all combinations of the receive and transmit enable buses. Matches detected generate +ABORT signals through 7432 to the appropriate transmitters and receivers.

Again, supporting the dual data paths, the REC CTL logic A and REC CTL logic B are the same so their discussion is together. On RFB request detection, the state machine activates +RREQO and monitors +RREQI. The +START XFER tells the REC STB logic to start transfer. An unusual condition, +RINTI at either HSFONI end, a transmitter-receiver collision, or a buffer over-run, causes REC CTL logic to halt the transfer with a +RINTO on this HSFONI end and an interrupt frame to the other HSFONI end. The state machine in these conditions control interface timings to specification.

Shown in Appendix A Figure 31 and Figure 32 is circuit schematics for REC CTL logic. Timings are displayed in Appendix C Figure 49 and Figure 50. The +RINTI, +ABORT, and +INT DET X monitoring functions are same as in XMT CTL logic with the +RFULL addition to stop buffer over-runs. However, the added +RREQO, transfer start notifying function is new. The logic synchronizes +REQ DET from RFB to the system clock and generates +RREQO, shown in the timing diagram. If during the next 1 usec +RREQI does not become active, the transfer proceeds until RFB is no longer sending data, indicated by +REQ DET falling and the BFB indicates it

is empty with -REMPTY. Detecting +RREQI during the 1 usec arbitration time, the logic notifies the other HSFONI end with +INT GEN R and the REC CTL logic is reset.

The REC STB logic A and REC STB logic B control STBO/STBI handshake of receiver data to the ACR-NEMA attached device. The -RRDA and -RRDB signals request BFB data. The logic monitors -REMPTYA and -REMPTYB to not transfer bad data to the device.

Appendix A Figure 30 shows REC STB logic A and REC STB logic B using TTL and Appendix C Figure 51 shows timing. +REQ SYNC generates the first 74163 counter load pulse. When REC CTL logic has successfully selected the device and activates +Start XFER the counter starts counting up from "B" hex to "F" hex. This count determines +RSTBO pulse separation which at the ACR-NEMA specification of 8 mega-bytes per second is 250 nsec. Since the clock rate is 20 MHZ or 50 nsec the five counts between +RSTBOs represent the required 250 nsec. The +RSTBO and +RSTBI signal overlap generate subsequent counter load pulses. The counter outputs Qa, Qb and Qc decoded generate a +RRD pulse, early enough to ensure valid TFB data the bus when STBO is active.

The reset logic handles resets after power up and after any unusual condition. During power up, the reset logic delays until power transients die out and then issues a reset to all logic on this HSFONI side. Logic detecting

unusual conditions signal reset logic with a +RSTREQ and reset logic responds with a reset to only the requesting logic. Shown in Appendix A Figure 33 with the REC BUS CTL logic is the reset logic implementation. The 74123 one shots generate reset pulses as requested. An RC network provides delay on power up and generates a reset request to reset all logic.

Buffer Functional Block

As the IFB is two main sections, two main sections compose the BFB, the transmit buffer and the receive buffer. These buffer data between ACR-NEMA device connected IFB and fiber optic link associated hardware. The detailed BFB operation follows.

Transmit Buffer. In Figure 12 is the block diagram of the transmit buffer of the BFB. Two 1k deep, 18 bit wide word fifos buffer data paths between IFB and TFB. Two 18 bit buses, +XMIT BUS A and +XMIT BUS B put IFB data into the fifo and two 18 bit buses, +BUS A X and +BUS B X, transfer fifo data to the TFB. Input control signals -XWRA and -XWRB write valid bus data into the fifo. The -XFULLA and -XFULLB signals tell IFB the fifo is full. When the IFB section is reset the appropriate fifo is reset with -XRSTA or -XRSTB. The TFB -XRDA and -XRDB signals remove fifo data while monitoring -XEMPTYA and -XEMPTYB for data end.

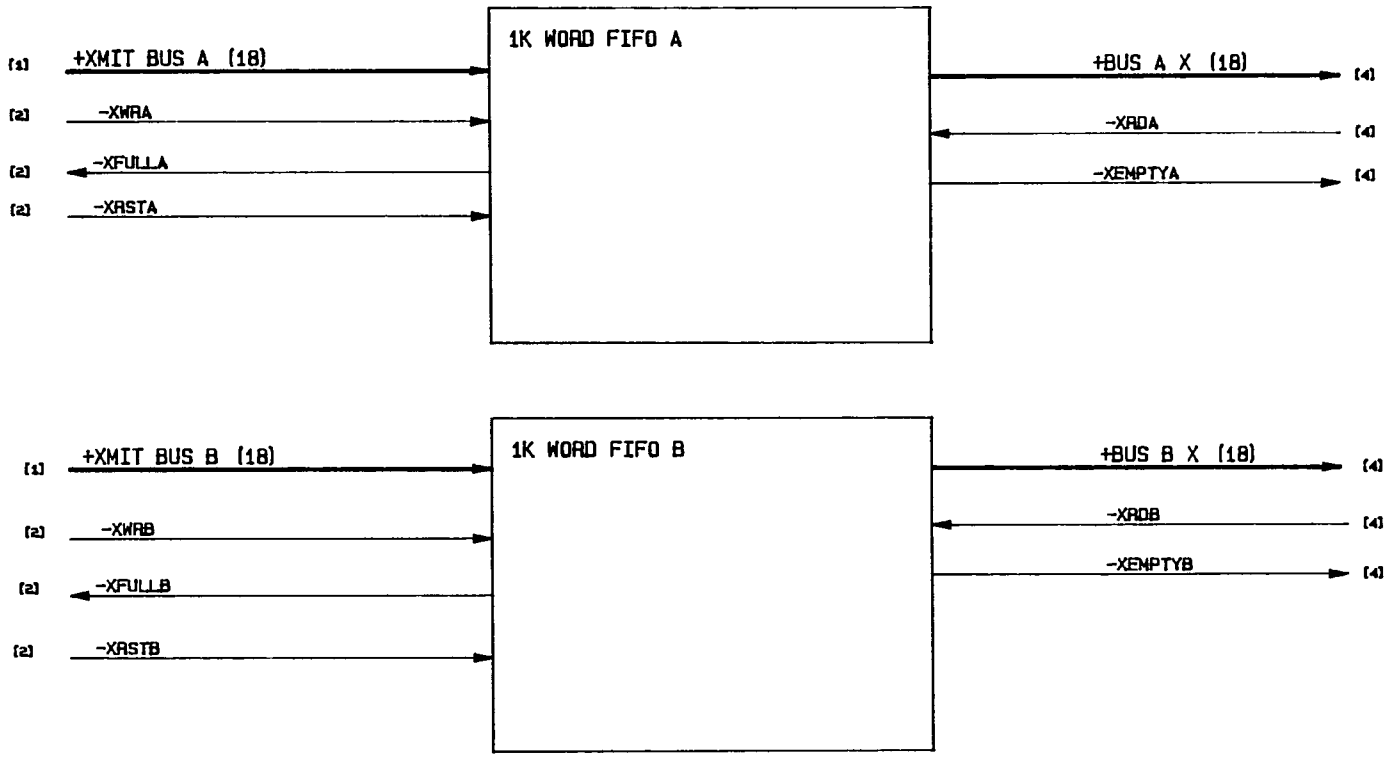

Figure 12. Block Diagram 3-BFB Transmit Buffer

Fifo circuitry tracks load and unload data address information.

The transmit buffer implementation schematic is in Appendix A Figure 34. It uses two 9 bit Integrated Device Technology (IDT) IDT7202 1k fifos. These are dual port memory devices using internally sequenced ring pointers tracking data input and removed. These devices width expanded, per the application note (IDT 1984, p.6), provide width required for the 18 bit bus. The IDT devices selected offer speed, 50 nsec access time, expandability, and a 9 bit data path. If future explanation requires more than 1k depth, these devices allow depth expansion also. The 9 bit data path allows parity bit handling with other data bits and without requiring an extra fifo for one bit.

Receive Buffer. The receive buffer block diagram is in Figure 13. The implementation is the same as the transmit buffer. However, the IFB removes data placed in the buffer by RFB. Control signals remain the same except they originate in different functional blocks. The design uses the IDT7202 fifos as shown in Appendix A Figure 43 like in the transmit buffer

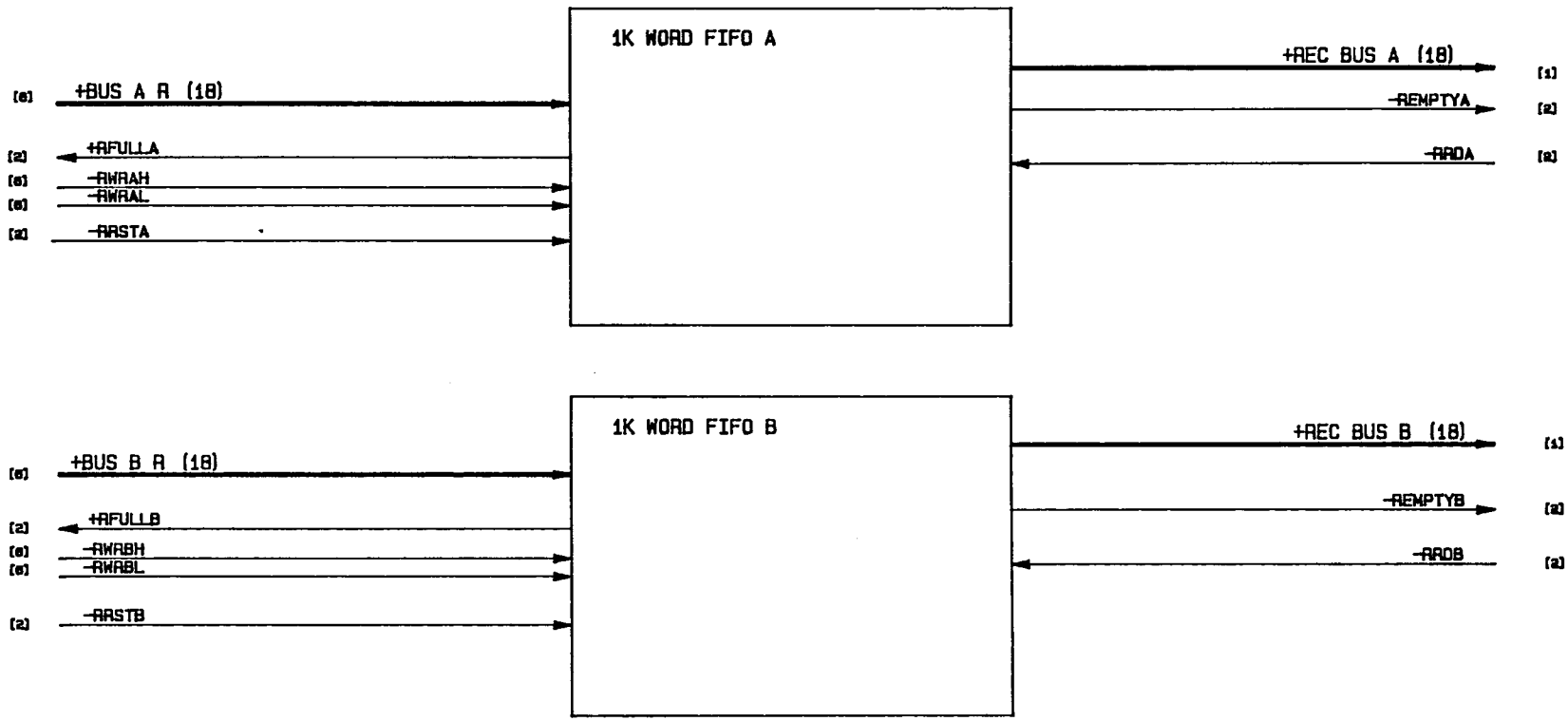

Figure 13. Block Diagram 7-BFB Receive Buffer

Transmitter Functional Block

With IFB attachment function and BFB interconnect function outlined, the HSFONI fiber optic side functions need discussion. The TFB encodes and serializes data and commands for fiber transmission in the optical functional block. This requires three logic sections, data control logic, command control logic, and transmit logic. Figure 14 shows the sections and their interconnection. Discussion turns now to this functional block's operational details.

Data Control Logic. The data control logic receives BFB data on +BUS A X and +BUS B X with BFB read logic signal -XRDA or -XRDB. The data control logic then passes these 18 bit words to the transmit logic in two 9 bit bytes over the +DATAA or +DATAB buses. The +RSB to the command control logic requests a +STRB write pulse to the transmit logic indicating valid data. The transmit logic indicates data taken with +ACK. The data control logic uses +1CLKX and -1CLKX clocks from the transmit logic generating the BFB read signals. The logic monitors -XEMPTYA and -XEMPTYB to know when BFB has data for transmission. The +STOPX is a stop indication from the command control logic with the +STOPPED handshake response indicating the data control logic has stopped.

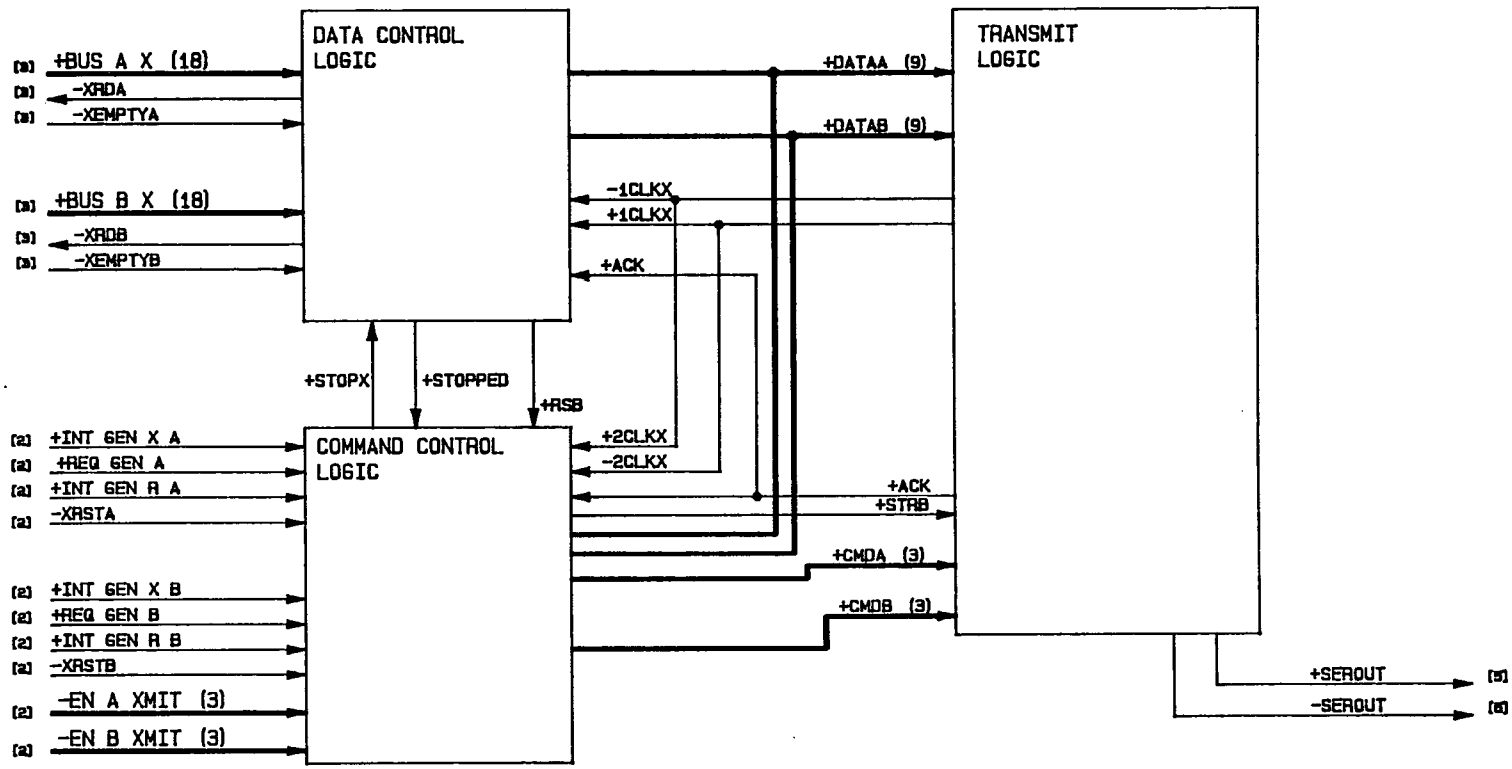

Figure 14. Block Diagram 4-Transmit Functional Block

Appendix A Figure 35 shows the data control logic circuit and the corresponding timing diagram is in Appendix C Figure 52. The transmit logic clock drives the data control logic. +1CLKX and -1CLKX, divided by two and four gives required timing. When either A or B BFB transmit fifo has data in it, +START activates. Two 7474 latches synchronize this to the system clock generating +EMP LCH. +EMP LCH activating causes a transmit fifo read, latching data in 74AS832 9 bit latches with -LCH WDR after data becomes valid in 50 nsec. Now data passes with +STRB and return handshake +ACK to the transmit logic. A 7474 latch generated +EN BYTE 1 and +EN BYTE 2 enables two data bytes onto the single byte wide buses, +DATAA or +DATAB to the transmit logic.

Command Control Logic. The command control logic handles encoding input commands into three bit binary understood by the transmit logic. Commands from each data path are +INT GEN X A and +INT GEN X B, +REQ GEN A and +REQ GEN B, and +INT GEN R A and +INT GEN R B. The IFB +REQ GEN signal indicates an ACR-NEMA device wishes transferring data and the signal remains active until the transfer completes. Upon signal activation, a binary "110" placed on the +CMD bus and interface card address placed on +DATA causes the transmit logic to send a request frame. When transfer ends, +REQ GEN deactivates and command control logic places a

"101" on +CMD bus causing transmit logic sending an end frame.

The command control logic also handles transfer termination commands +INT GEN X and +INT GEN R. A transmitting device detecting +INTI from the ACR-NEMA device causes +INT GEN X activation. The response "100" placed on the +CMD bus causes the transmit logic send an interrupt X frame. Likewise, a receiving device interrupted by +INTI causes +INT GEN R activation. The command control logic encodes a "011" on +CMD and the transmit logic sends a interrupt R frame.

Appendix A Figure 36 and Figure 37 shows the transmit command control logic circuity and the timing diagram is in Appendix C Figure 53. The data transfer sequence starts with +REQ GEN active and the first data byte is available indicated by -XEMPTY not active. This generates a strobe request, -RSA, from the 7421. The 74136 encodes +REQ GEN into a "001" on +CMD and the -RSA causes a +STRB on the next rising -2CLKX edge. When read by the transmit logic, indicated by an +ACK response, the sending interface card address passes through the 74240 with a +STRB and +ACK handshake. After data transfer completion, when +REQ GEN falls and -XEMPTY is active, the logic sends the frame end command, "010", in like manner.

The second half of Appendix C Figure 53 shows

transfer interrupt timing. The timing is the same for a transmitter interrupt, +INT GEN X, or a receiver interrupt, +INT GEN R. When the interrupt line comes active and strobe request, -RSA, activates, a command is placed on on +CMD. The +INT GEN X command is "011" and +INT GEN R "100". Again the transmit logic responds to +STRB generation with +ACK.

Transmit Logic. The HSFONI workhorse is the transmit logic. Data and commands are encoded, serialized, and passed as an ECL differential signal to OFB. First the transmitter logic encodes data. To ensure data recovery, receive end pll synchronization requires encoded data, not allowing raw user data to pass directly across the link. If no data is present, the transmit logic sends synchronization characters. Commands are encoded into unique code words not used in data encoding. The encoded data is serialized for link transmission. The transmit logic provides the mechanism to reliably transmit data across the link.

Taking advantage of available technology, two VLSI TAXI transmitter chips implement the transmit logic. They are Advanced Micro Devices (AMD) Am7968 parts. These chips, however, are slightly slower than the required 176 Mbps data rate required to support two 8 mega-byte per second ACR-NEMA devices. However the BFB will compensate for this. Shown in Figure 15 is the Am7968 TAXI block diagram.

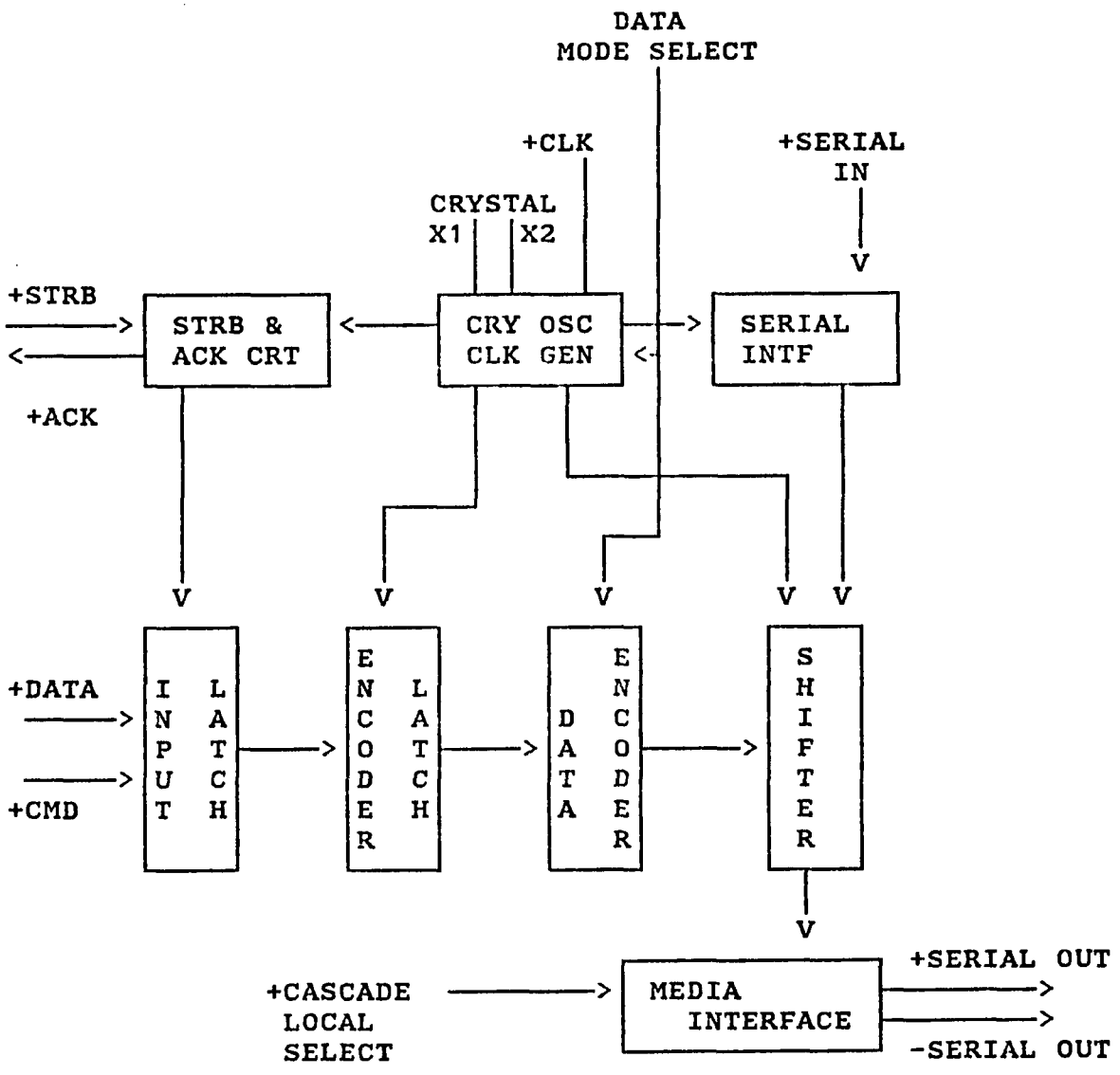

Figure 15. The Am7968 TAXI Transmitter Block Diagram

The Am7968 TAXI Transmitter runs at a 125 Mbps link speed. Device operation allows dividing the 12 input bits into data and command bits 3 ways. If using 8 data bits, then 4 command bits are available. The TAXI uses a 10 bit encode, taking data every 125/10 MHZ or 80 nsec. Using nine data bits leaves 3 command bits and data transfer is every 125/11 MHZ or 88 nsec using a 11 bit encode. Also available is 10 data bits with 2 command and a 124/12 MHZ or 96 nsec rate and 12 bit encode.

Since HSFONI uses 9 bit data, the 9 data and 3 command scheme is selected. The data input rate, 88 nsec, is slightly less than 12 mega-byte per second (11.36 mega-byte per second) and less than the required 16 mega-byte per second required for two ACR-NEMA devices. However, the advantages of using this device exceed the reduced data rate disadvantage. Data encoding circuits are a major circuit design task. Data serialization is tricky because of high speeds and circuit design rules required. Combining these functions into a single, easily interfaced chip is better than a unique design handling the slightly higher 16 mega-byte per second data rate.

The ACR-NEMA device does not see the slower link data rate impact because of the BFB. Logic removes data transferred into BFB at the 16 mega-byte per second out at 12 mega-byte per second. The BFB holds 1024 words, so

translating data rates into words yields 8 mega-words per second into, and 6 mega-words per second out of, the BFB. The ACR-NEMA frame is 2048 data words and 4 control words or 2052 words total. Transferring a frame into the BFB takes 2052 words x 1 sec/8 mega-words or 256.5 usec. Removing the data takes 2052 words x 1 sec/6 mega-words or 342 usec. The difference 85.5 usec x 1sec/8 mega-words, the input data rate, is 684 words. This means there enough room in the BFB compensating for the speed mis-match. With the size data frame used, the buffer will not fill before frame transmission, even though input rate is faster than output rate.

When transmitting multiple frames, the BFB will not be overrun because of the hardware design. However, when the buffer becomes full, the reduced input data rate is the 12 mega-byte per second output data rate. This is handled by the asynchronous ACR-NEMA STBO/STBI data transfer handshake. If this becomes a performance problem, the BFB expands in 1k depth increments as required.

The TAXI uses the ANSI X3T9.5 (FDDI) code requiring a maximum of three consecutive non-transition bit times. The FDDI code assumes NRZI data where a transition represents a "1" and a "0" no transition. The code symbol pattern has the same average amount of high and low times. This "DC" balance minimizes the data-induced noise effects

on a AC-coupled system. DC shifts in this system type cause jitter errors in the recovered wave form. (AMD 1986,p.6) This meets the Siecor receiver module AC coupling requirements.

The TAXI transmitter encodes data, command and synchronization bytes with unique symbols allowing receiver differentiation between the three types (Connor 1987, p. 81). Three command control logic bits, allow eight commands represented by unique code words. Only seven control words are available because of sending data when the command bus contains "000". The HSFONI uses four of the seven commands, request frame, end frame, interrupt X frame, and interrupt R frame.

Appendix A Figure 38 shows the circuit using the TAXI. The application note contains timing details. (AMD 1986, p21-25). The TAXI transmitters encode and serialize data control logic data and command control logic commands. The +SEROUT of the first stage feeds the second stage. Each data path uses a separate TAXI Transmitter. A single +STRB loads both TAXIs. A positive and negative phase clock is generated from the TAXI clock for data and command control logic section synchronization.

Optical Functional Block

The optical functional block contains Siecor transmitter and receiver modules. The fiber optic research section explains operational details. This section outlines HSFONI system interface. Each HSFONI end has a transmitter and receiver. A dual fiber cable connects the two ends. This end's transmitter connects to the other end's receiver. Current fiber technology does not allow transmit and receive over a single fiber. The block diagram is in Figure 16. The schematic is in Appendix A Figure 39.

Optical Transmitter. The Optical Transmitter receives an encoded, serial, ECL, bit stream from the TAXI transmitter in the TFB. The 1.3 um wave length LED in the Siecor transmitter module turns this bit stream into light pulses in the optical fiber.

Optical Receiver. At the receive end, the optical receiver gets the light pulses. The Siecor receive module using a PIN photodiode detects the light pulses, and an internal, AC coupled, differential amplifier amplifies the signal to exceptable ECL logic levels. The ECL bit stream is sent to the TAXI receiver module in the RFB.

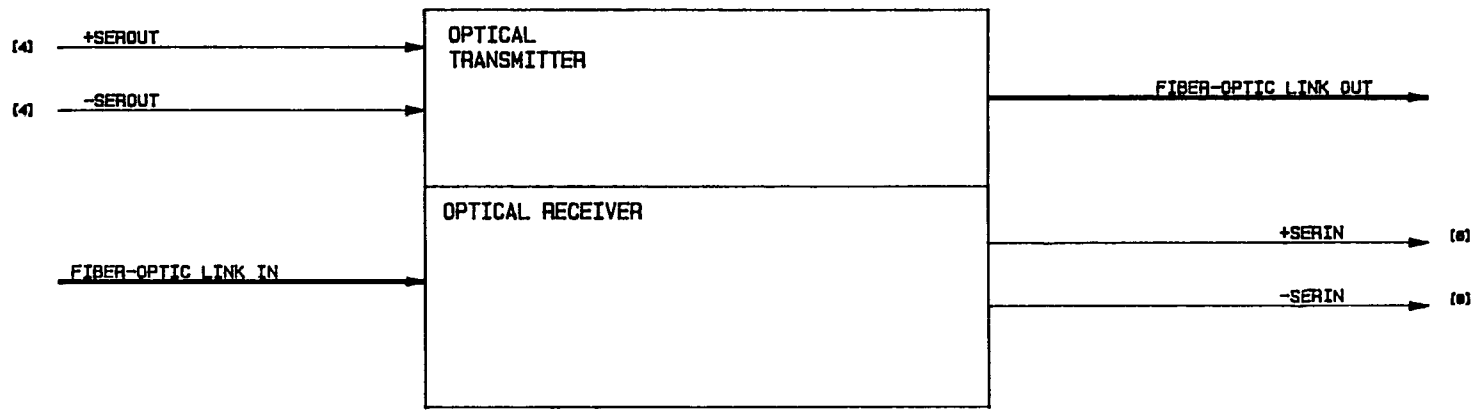

Figure 16. Block Diagram 5-Optical Functional Block

Receiver Functional Block

Receiving the OFB ECL bit stream, decoding and deserializing data is the RFB. Parallel RFB data then passes to the BFB. Three sections, receive logic, data control logic, and command control logic do this function. Showing the block diagram is Figure 17.

Receive Logic. Receiving the ECL bit stream, the receive logic, using the TAXI transmitter companion, the Am7969 TAXI receiver deserializes and decodes data. Data placed on one of the two nine bit data busses, +DATAA or +DATAB, moves to data control logic. The +DSTRB signal indicates valid data. Likewise a decoded command placed on one of the command busses, +CMDA or +CMDB is read by the command control logic. +CSTRB validates commands on the bus.

Two Am7969 TAXI receive modules implement the receive logic. Showing the TAXI receiver block diagram is Figure 18. Both TAXI receivers get serial data. The first receive TAXI gets the first byte after an idle character and placed on +DATAA. The first TAXI signals it got a byte, with +IGM to the second TAXI allowing the second TAXI to receive the next byte for placement on +DATAB. The +DSTRBA or +DSTRBB signals validate data on +DATAA or +DATAB. +CSTRBA or +CSTRBB indicate commands on +CMDA or +CMDB.

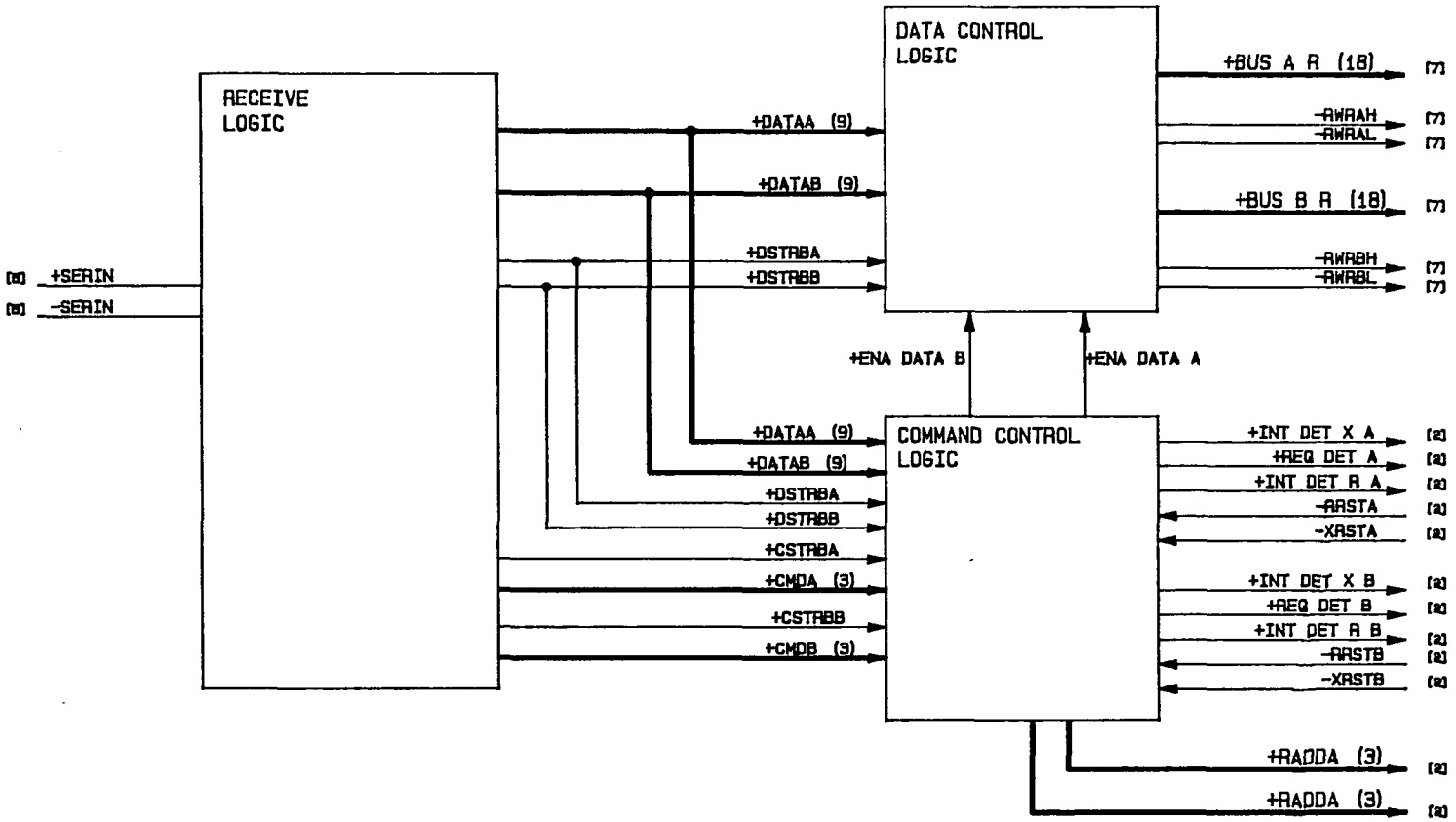

Figure 17. Block Diagram 6-Receiver Functional Block

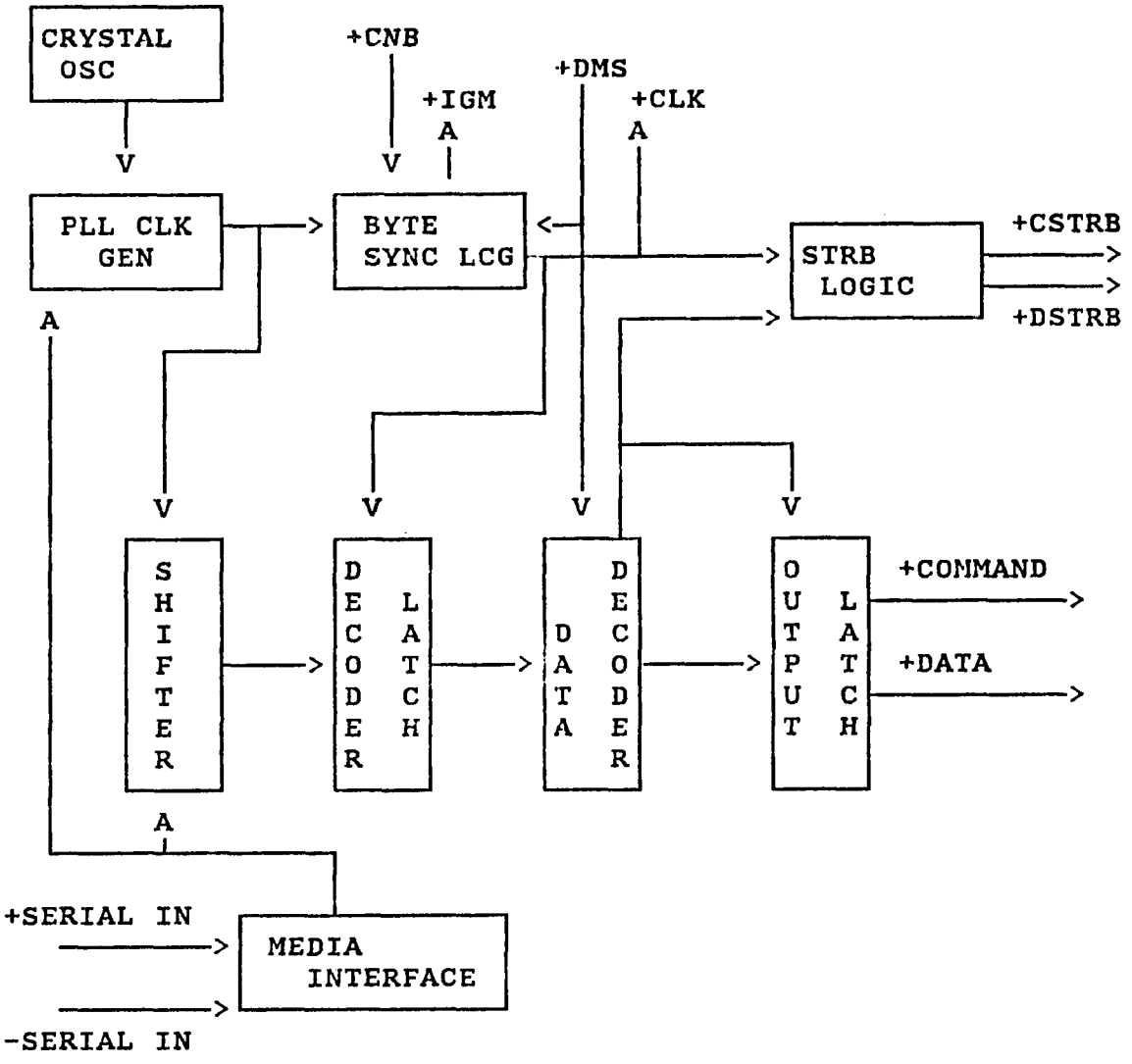

Figure 18. The TAXI Receiver Block Diagram

Appendix A Figure 40 shows the circuit using the TAXI and the found in the application note are timing details. (AMD 1986, p26-30).

Data Control Logic. The data control logic reformats received data from 9 bits to a 18 bits before passing to the BFB. Data input on +DATAA passes to +BUS A R and input data on +DATAB goes to +BUS B R. Two bus signals for each bus output confirm bus high and low bytes.

Appendix A Figure 41 shows the TTL circuitry implementing the data control logic and Appendix C Figure 54 shows the associated timing. The incoming +DSTRBA or +DSTRBB separates into a high byte strobe, +RWRAH or +RWRBH, and a low byte strobe, +RWRAL or +RWRBL, with a 7474 latch and 7400 gating. Byte wide input data on +DATAA or +DATAB is placed on the high and low bytes of BFB busses, +BUS A R or +BUS B R. The +REQ DET A or +REQ DET B ensures low byte passing first.

Command Control Logic. Command control logic receives receive logic decoded commands on the +CMDA or +CMDB bus. These are decoded into +XMIT INT DET A or +XMIT INT DET B, +REQ DET A or +REQ DET B,and +REC INT DET A or +REC INT DET B from 3 bit binary form. The interface card address, received with +REQ DET A or +REQ DET B placed on the +RADDA or +RADDB bus selects the companion interface card on this HSFONI side.

In Appendix A Figure 42 is TTL circuitry implementing the command control logic. The timing is in Appendix C Figure 55. A 74138 decodes the command bus, +CMDA or +CMDB. Valid data indicated by +CSTRBA or +CSTRBB latches the 74138 output. When a transfer starts with the request command, the first data byte following latches in the 74145. This byte contains the interface card address, and passes to the IFB enabling the correct interface card. Transmitter or receiver interrupt commands cause the +REQ DET A or +REQ DET B line to reset.

APPLICATIONS

Of the many viable applications for an interface like HSFONI, two specific applications were investigated. One application uses HSFONI for image transfer in a Picture Archive and Communications System (PACS). HSFONI provides high speeds required for image transfer. Another application attaches a number of HSFONIs together in a loop forming a local area network. This requires additional work designing a gateway for HSFONI end to end attachments.

A PACS System Using HSFONI

In Figure 19 is a PACS system incorporating HSFONI. Proposed by van der Voorde et al. (1986), Hedge, Gale, and Giunta (1986), as well as Martinez, Archwamety and Nemat (1985) is this general PACS architecture. Digital imaging equipments (IE) produce images passed over HSFONI to the acquisition node (AN). In IE and ANs supporting ACR-NEMA type connection, HSFONI replaces the connection cables with no additional system modification. HSFONI, allowing two of eight devices transmitting data fits this application well. The transmit data time is small compared to equipment set-up time. The equipment, likely not used 100% of the time, introduces more interface idle time. With one device being set-up or unused, another of the eight devices transmits

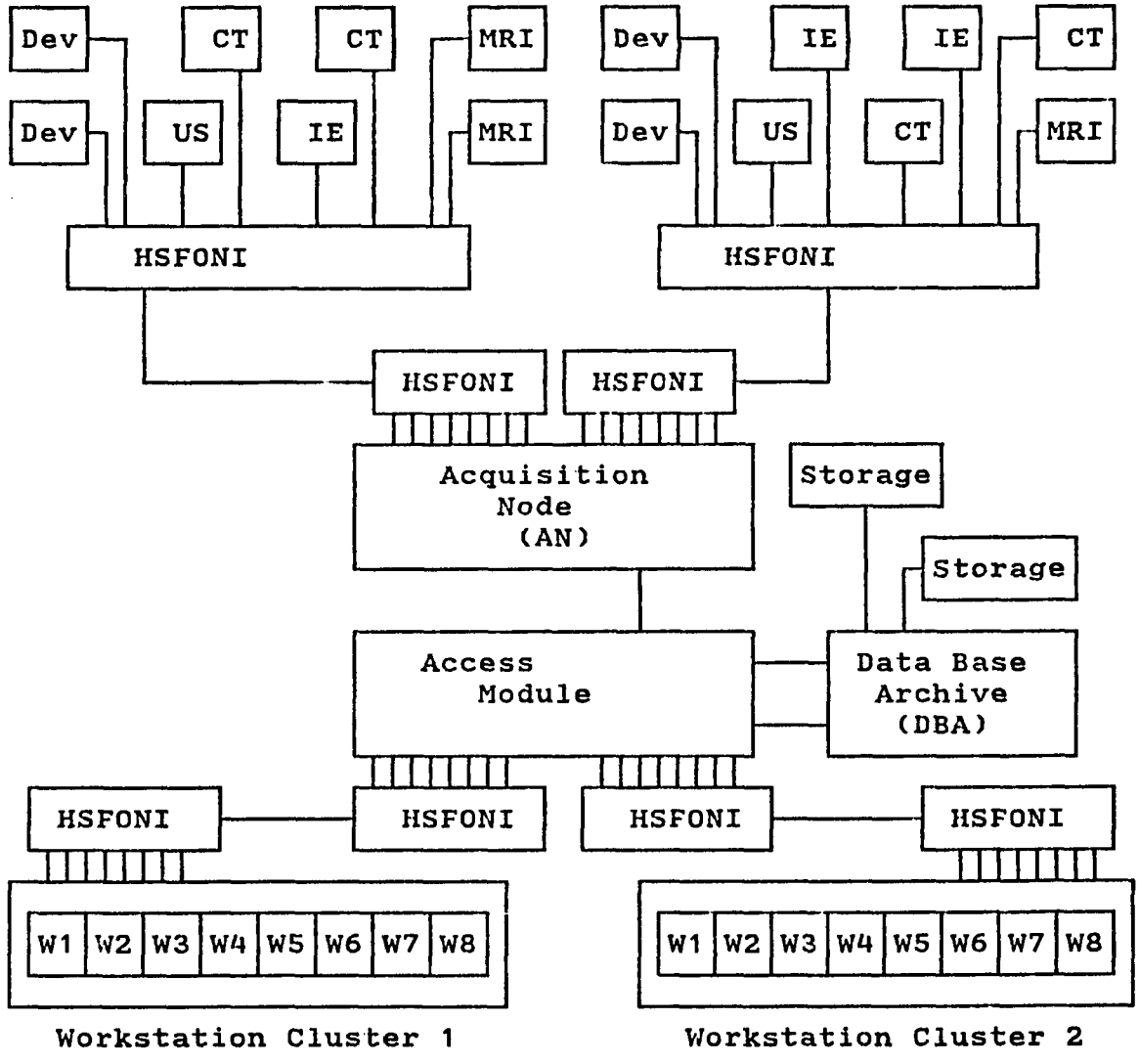

Figure 19. HSFONI Based PACS

data. Image data passes through the AN, a data steering device, to the data base archive (DBA) for storage.

The second HSFONI use in a PACS is connecting the DBA to the display console (DC) workstation. The system retrieves previously generated and stored images for viewing at the DC. HSFONI is useful here because commands to the DBA and images to the DC move quickly across the same interface. Once displayed, the image viewing for some time allows other work stations access the DBA from HSFONI.

A Local Area Network Using HSFONI

Adding a Tee connection gateway allows HSFONI end to end connection in a loop forming a local area network (LAN). This interconnection allows point to point communication as well as network connection. A typical system is shown in Figure 20. In this system configuration connects all devices on a hospital floor or area to the gateway. The gateway handles routing device data to the correct source.

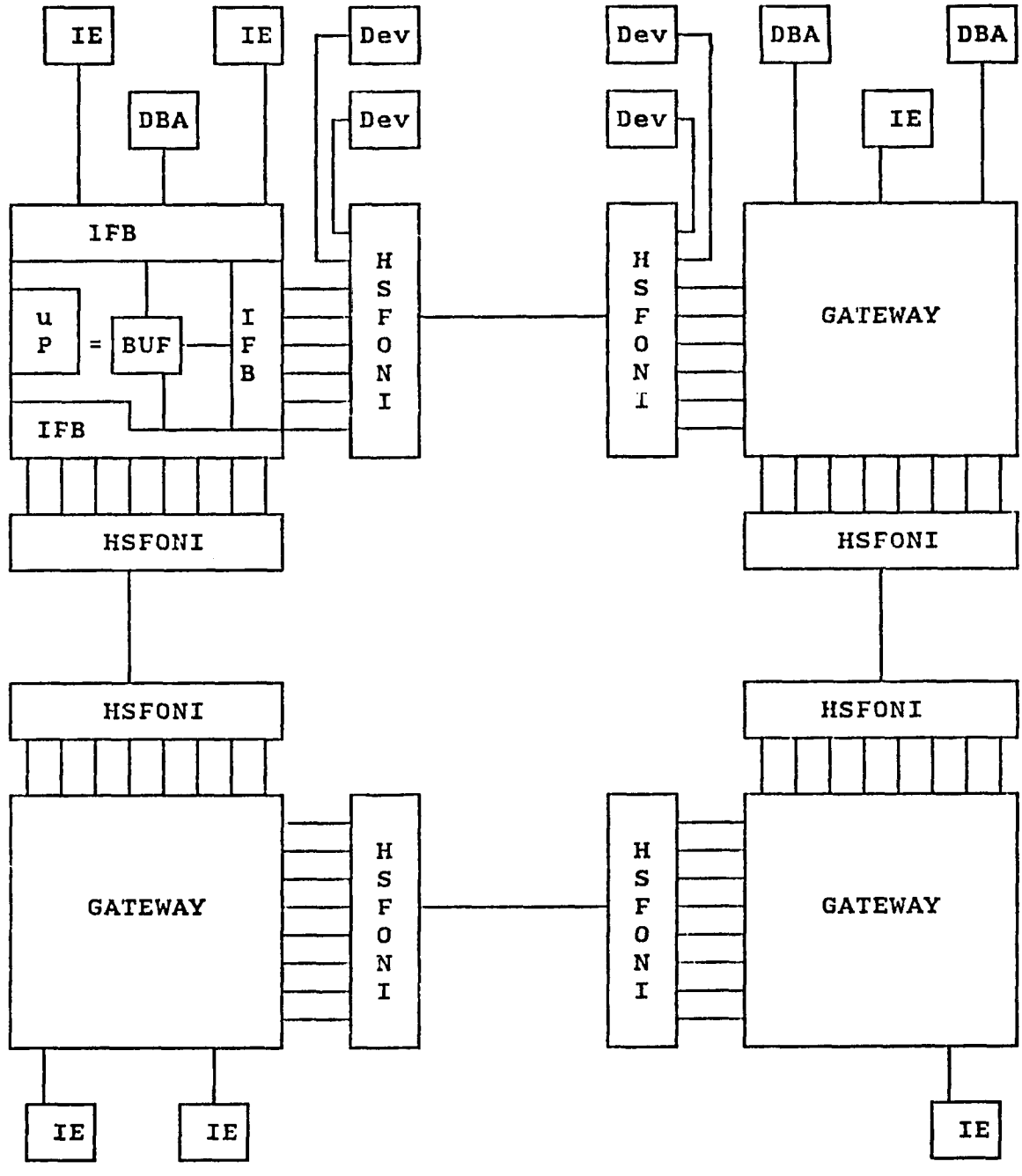

Figure 20. HSFONI Local Area Network

System Concepts

Multiple HSFONIs connected with gateways form a LAN. Each gateway forms a node taking data off or on the LAN. Gateway software handles data framing and data flow. Using the gateway to receive and re-transmit data is better than attaching optical splitters directly to the light in a fiber. Splitters tend to have high losses, limiting system length and performance. Since HSFONI has separate transmit and receive paths, the LAN can pass data in two directions. The ACR-NEMA collision arbitration scheme using +REQO/+REQI provides LAN access control.

Gateway Design

This system requires a gateway design with both hardware and software. Physical HSFONI attachment requires three hardware ACR-NEMA connection ports. Two ports attach to 8 devices and connect HSFONI end to end. The third port attaches to devices requiring link attachment. The ports could use the HSFONI IFB hardware.

Like in HSFONI, IFB passes data packets to a buffer. In the gateway the buffer requires multi-port access and micro-processor control. A port before receiving a data packet gets storage area assigned by the micro-processor software. After software analysis determining final destination, the software starts another port off loading the data packet.

The gateway software routes network data. Software routes data packets for a this gateway's attached devices to the correct device. Other data packets pass to the outgoing HSFONI. Software frames gateway attached device data for network transmission. Gateway software provides the OSI model layer functions above the Physical Layer, and fits the ACR-NEMA framework.

SUMMARY AND CONCLUSIONS

The HSFONI design shows that today's technology can realize a high speed fiber optic network interface for medical image transfer. In conclusion discussion turns to the design goals met, the design limitations, and future work topics.

Design Goals Met

The initial design was replacing the ACR-NEMA standard interface physical layer with a fiber optic link. The interface supports 2 devices communicating in one direction at a time. Each device operates at a maximum data rate of 8 mega-bytes per second with a link speed of 176 Mbps.

Data Rate

Although the optical interface operates at 200 Mbps, above the 176 Mbps design goal, taking advantage of VLSI technology compromised the electrical interface. Selecting devices available to encode and serialize and decode and deserialize, reduces circuit design. The devices, however, operate at 125 Mbps slightly less than desired link speed. The advantages of using VLSI devices out-weigh the small performance loss.

Technology Used

It is possible, as shown by HSFONI design, to obtain required performance using TTL components in parallel data handling areas. The serial portion, because of high speeds, requires ECL technology. The VLSI devices incorporate ECL internal to the devices requiring no special ECL circuit design. The optical technology includes a 1.3 um LED light source into a multi-mode graded index fiber detected with a PIN photo diode. This optical link is good for 200 MHZ operation up to 1.5 km.

Design Limitations

Early in this project, speed mis-match between optical technology and electrical technology in an interface like HSFONI became clear. The ECL technology upper limit is 300 MHZ. Available today, off the shelf, are fiber systems running at 500 MHZ. In the lab, work is being done at 1.2 and 1.7 GHZ. The fiber bandwidth is very large, but there are limits on the electrical technology getting data into the fiber.

This problem's answer is in the GaAs technology. Becoming available are GaAs multiplexers allowing time multiplexing multiple data streams onto one fiber (Tunick 1987 pp. 35,36,40). The technology is very new and at these frequencies a whole new set of circuit problems appear. Another type of multiplexing, wave length division

multiplexing, shows promise in the GTE labs (Lay 1987 pp. 27,47). This allows transmitting light waves on the same fiber with as little as 3 nm difference in wavelength. Future developments may open many possibilities in this area.

Future Work

Future topics are many, but a few deserve discussion. Component reduction using the many programmable devices available is one interest area. Electrically programmable logic devices (EPLD) of both the EPROM type and RAM type offer many advantages in reducing component count. Also available are programmable state machines. Using available GaAs technology and increasing the data into the fiber, taking advantage of all the fiber bandwidth is another topic of interest. This could extend the data rate to an upper limit of 2 GHZ. A gateway design, allowing an interconnected HSFONI circular LAN also is a rewarding topic for future work.

APPENDIX A: HSFONI CIRCUIT SCHEMATICS

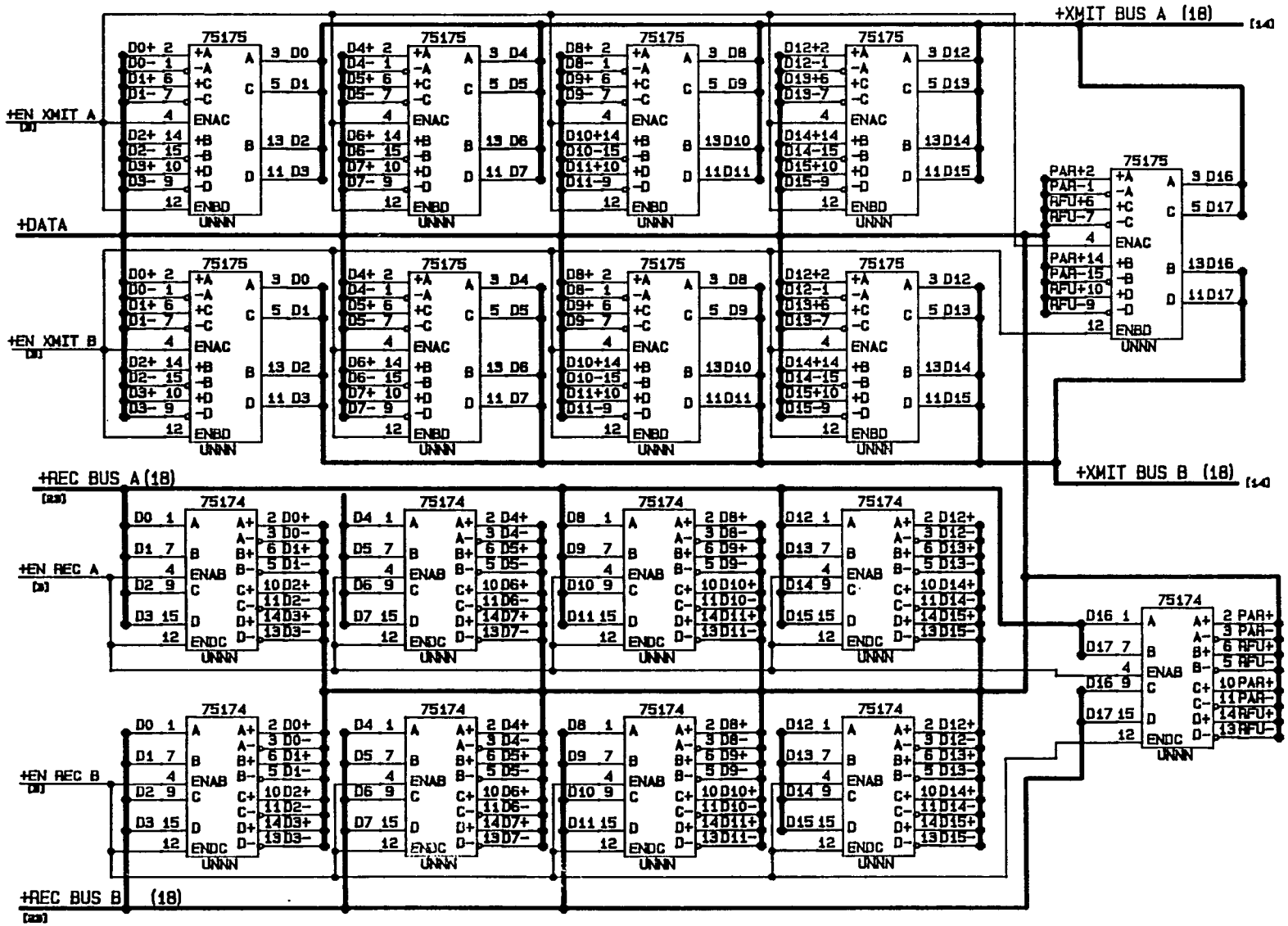

Figure 21. HSFONI Schematic 1 - Interface Card Data Flow Logic

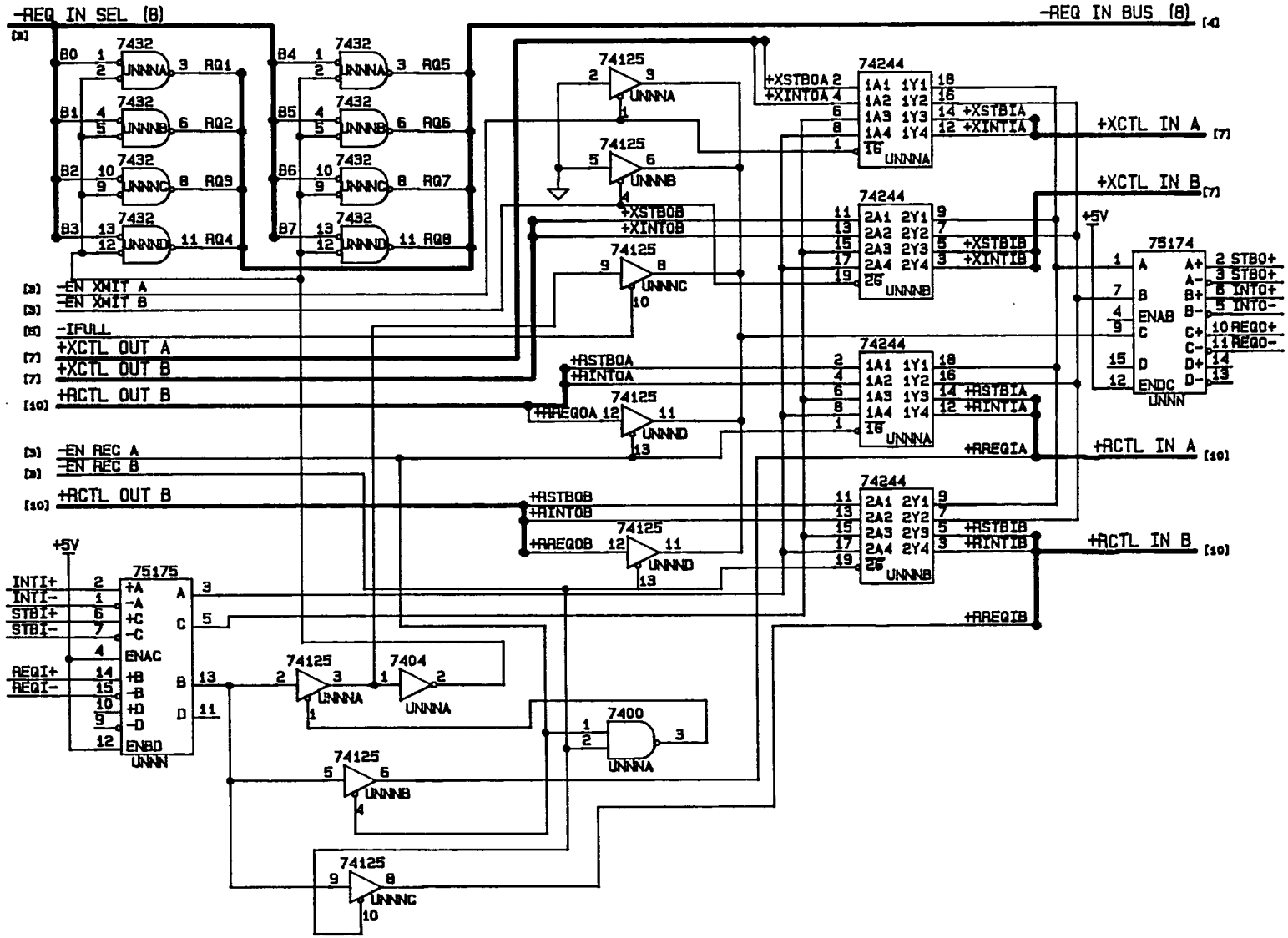

Figure 22. HSFONI Schematic 2 - Interface Card Control Logic

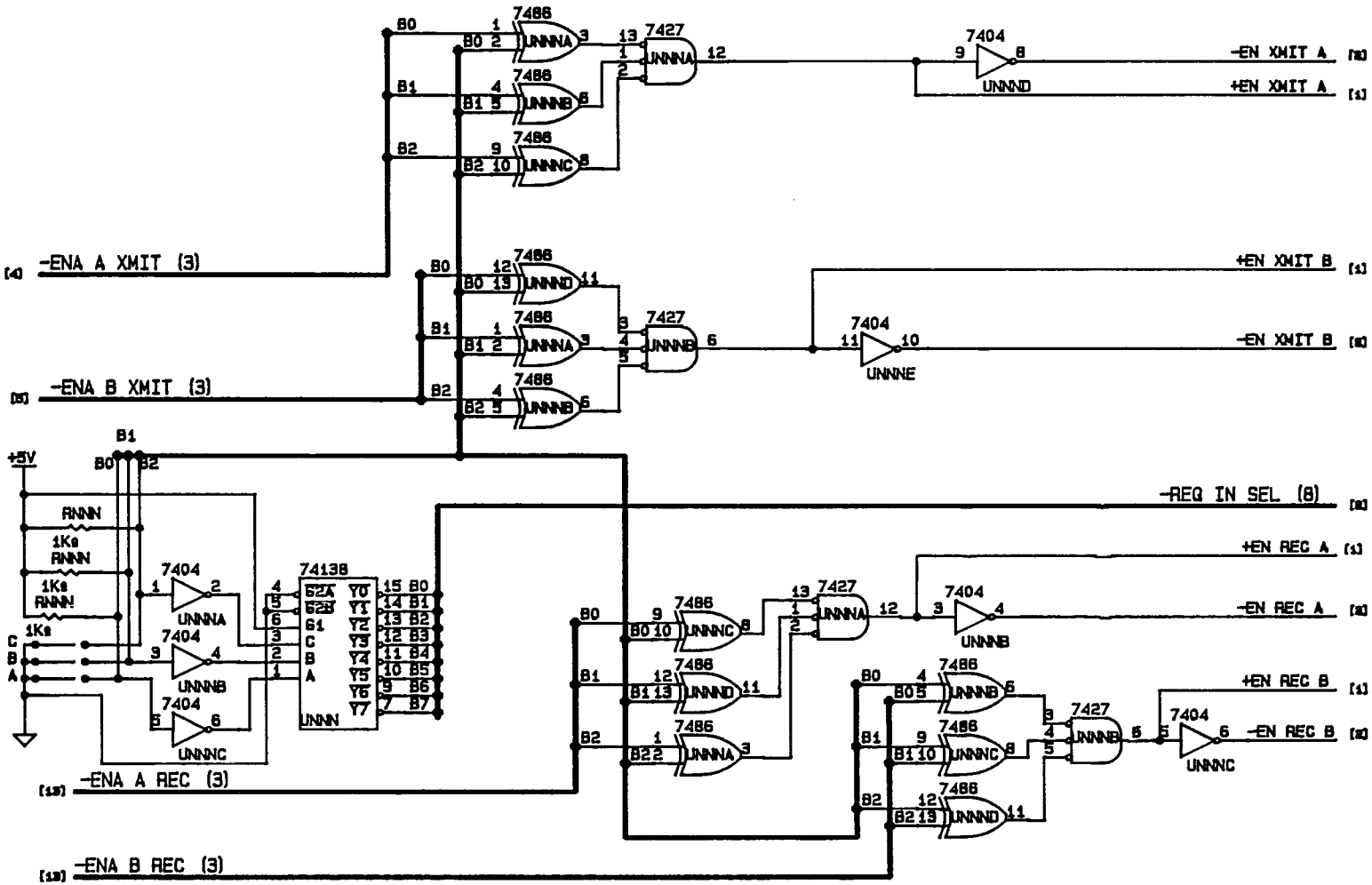

Figure 23. HSFONI Schematic 3 - Interface Card Enable Logic

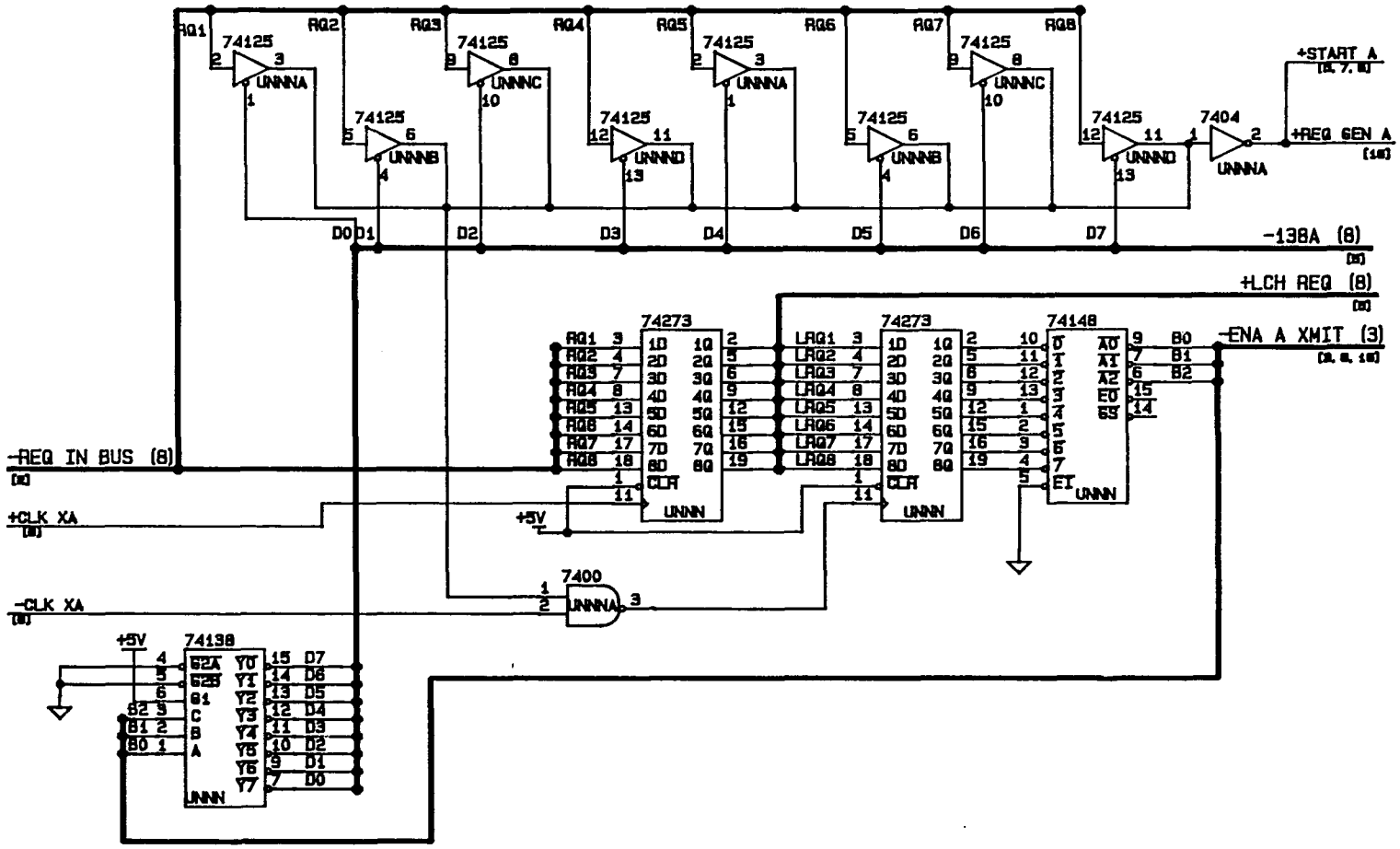

Figure 24. HSFONI Schematic 4 - Interface Control ARB Logic 1

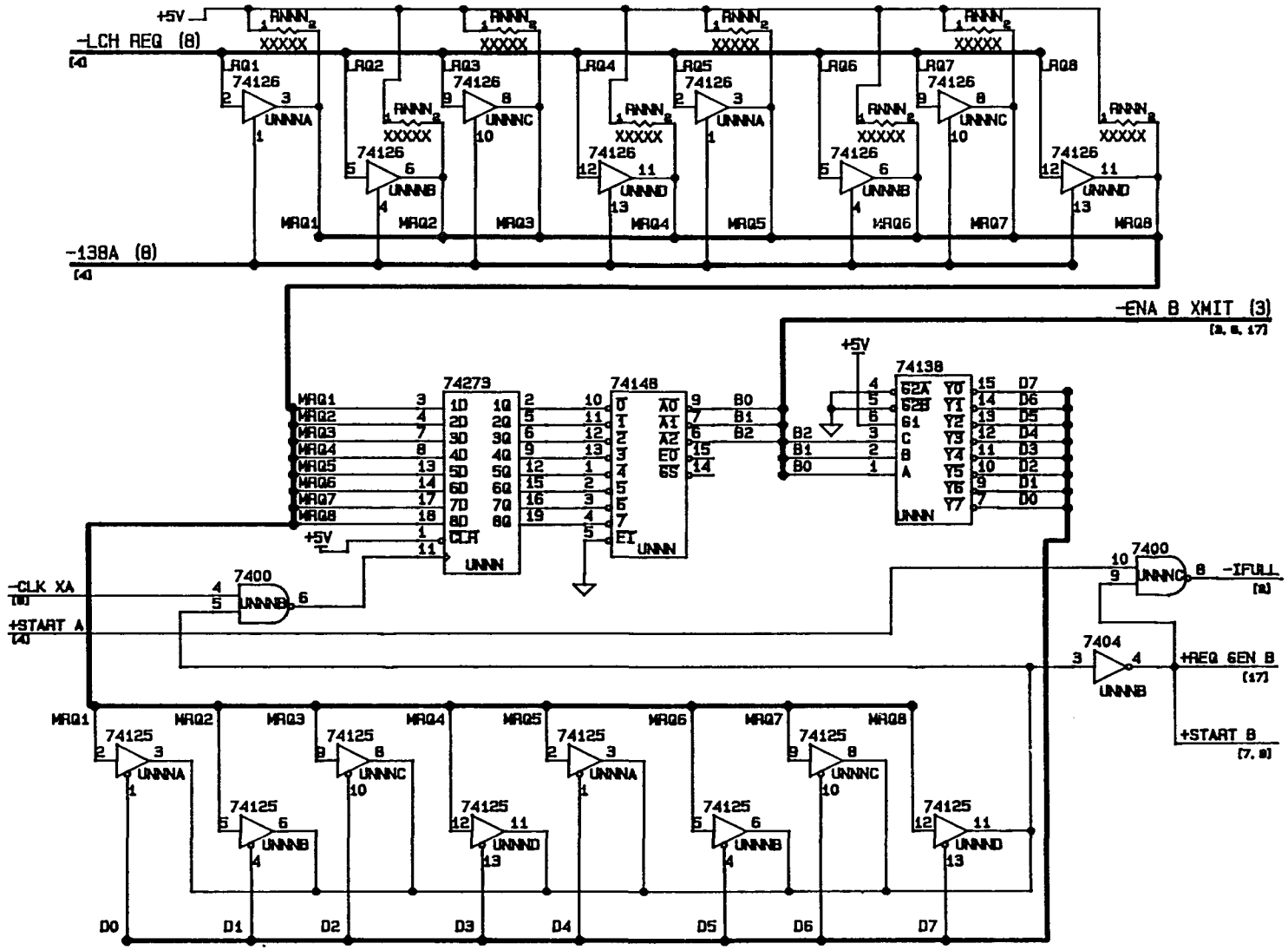

Figure 25. HSFONI Schematic 5 - Interface Control ARB Logic 2

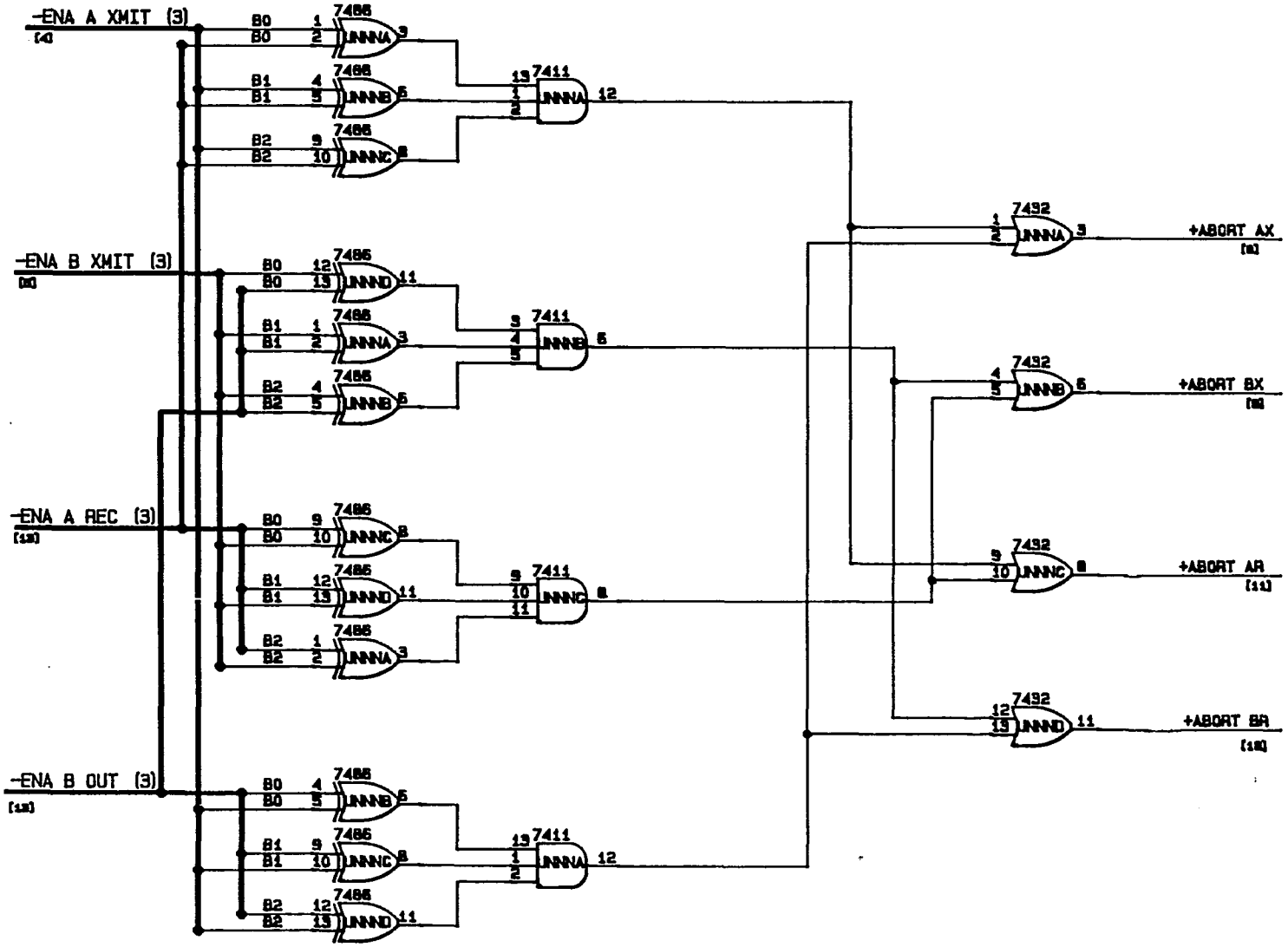

Figure 26. HSFONI Schematic 6 - Interface Control Collision Logic

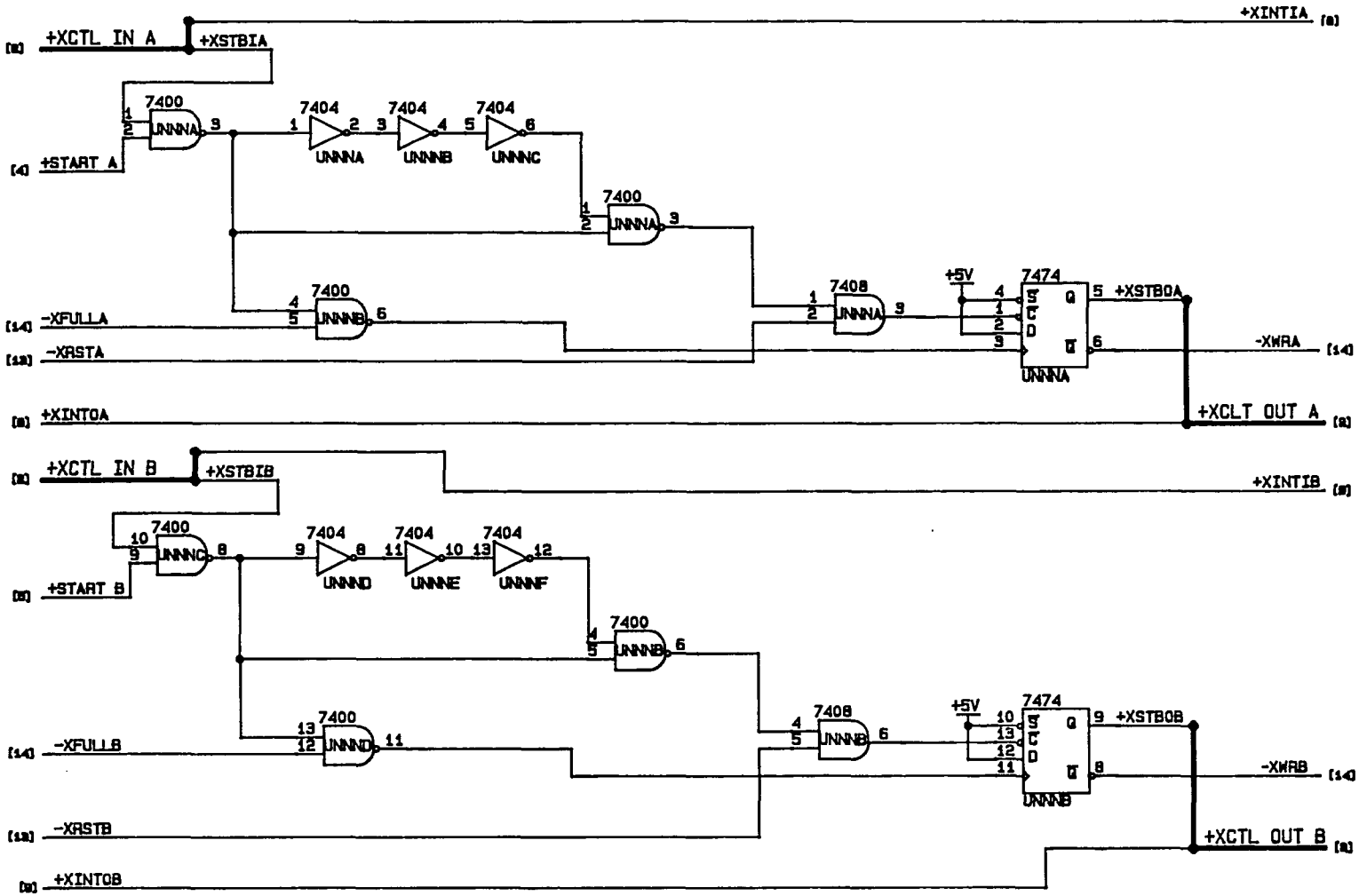

Figure 27. HSFONI Schematic 7 - Interface Control XMT STB Logic A/B

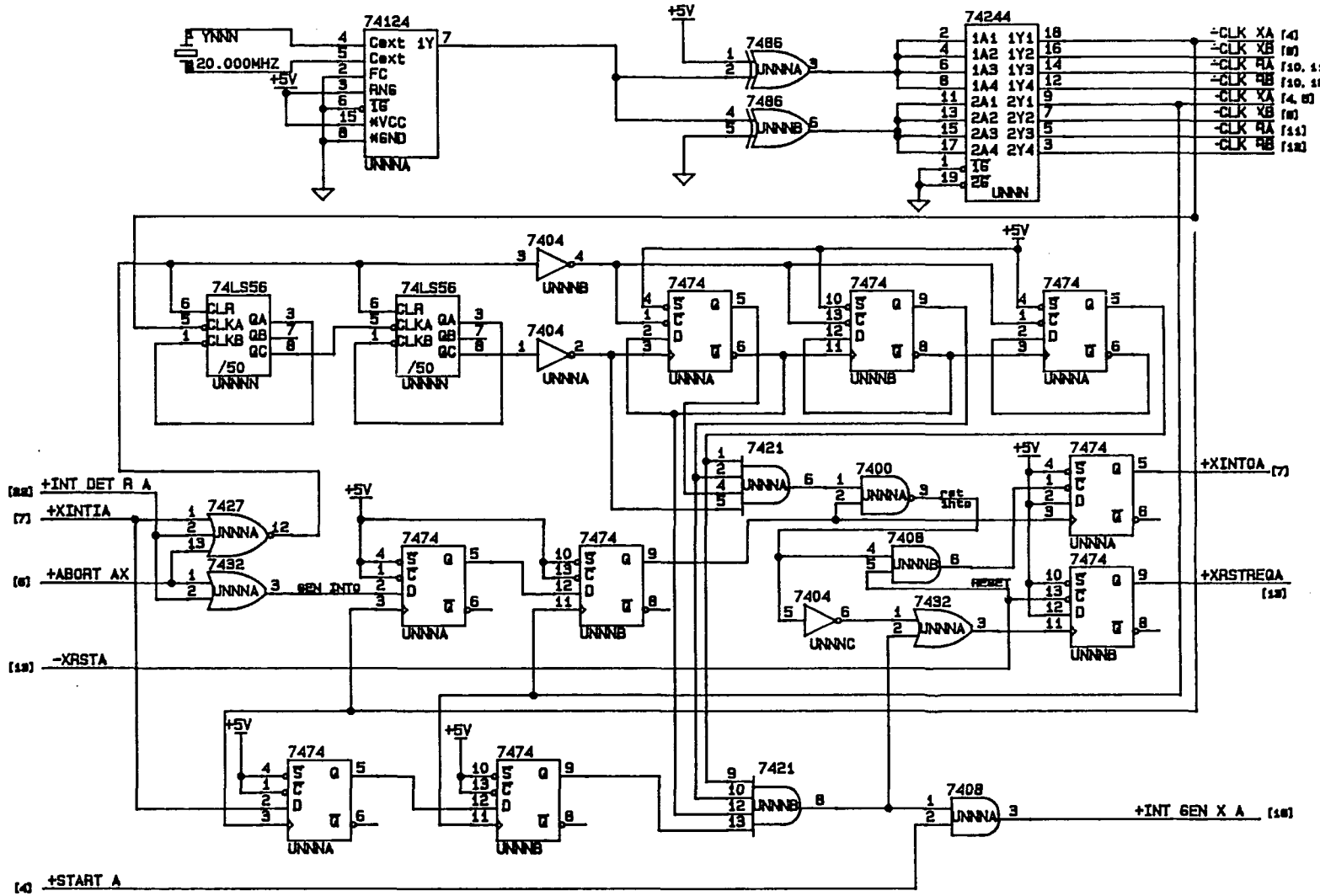

Figure 28. HSFONI Schematic 8 - Interface Control XMT CTL Logic A

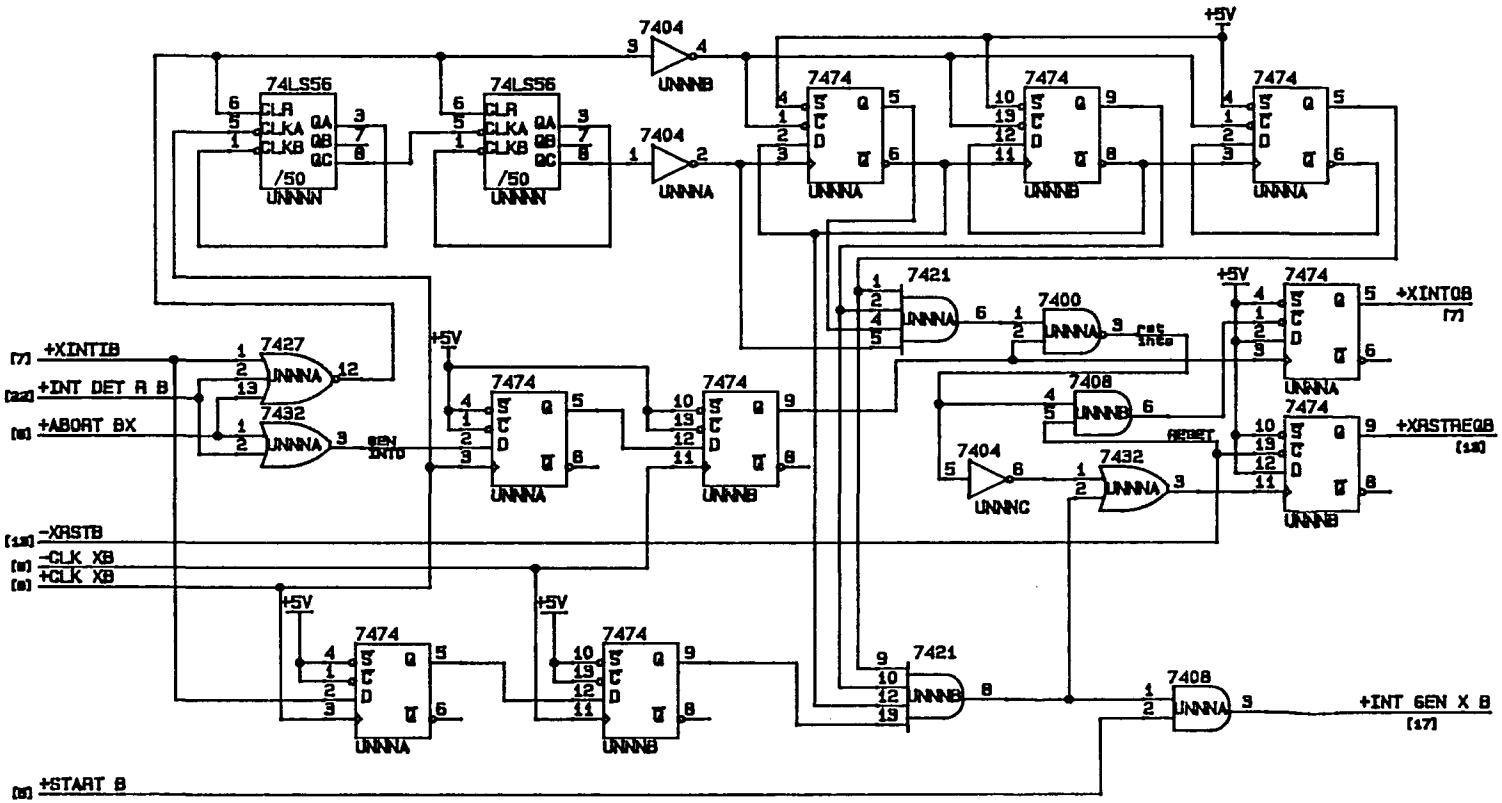

Figure 29. HSFONI Schematic 9 - Interface Control XMT CTL Logic B

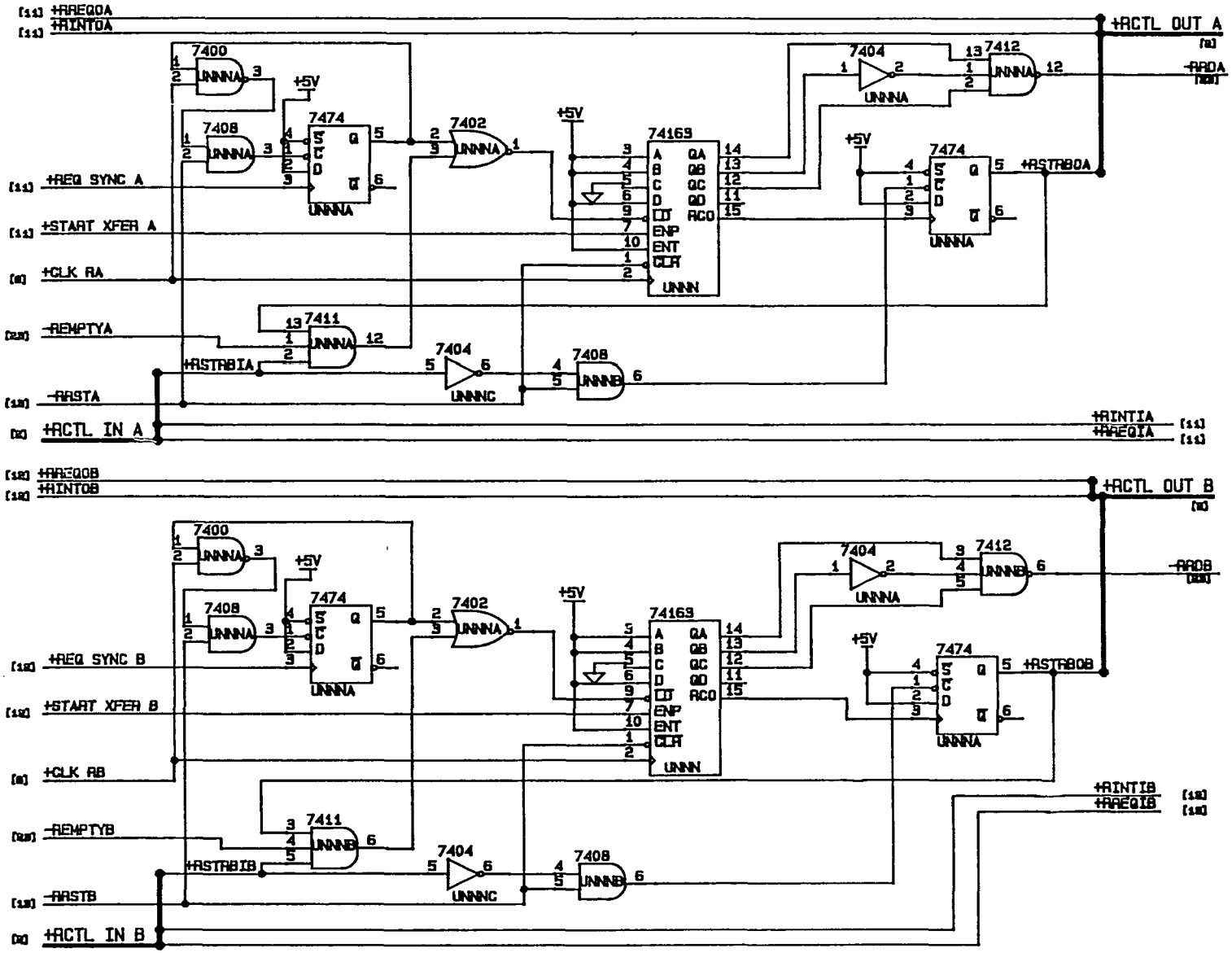

Figure 30. HSFONI Schematic 10 - Interface Control REC STB Logic A/B

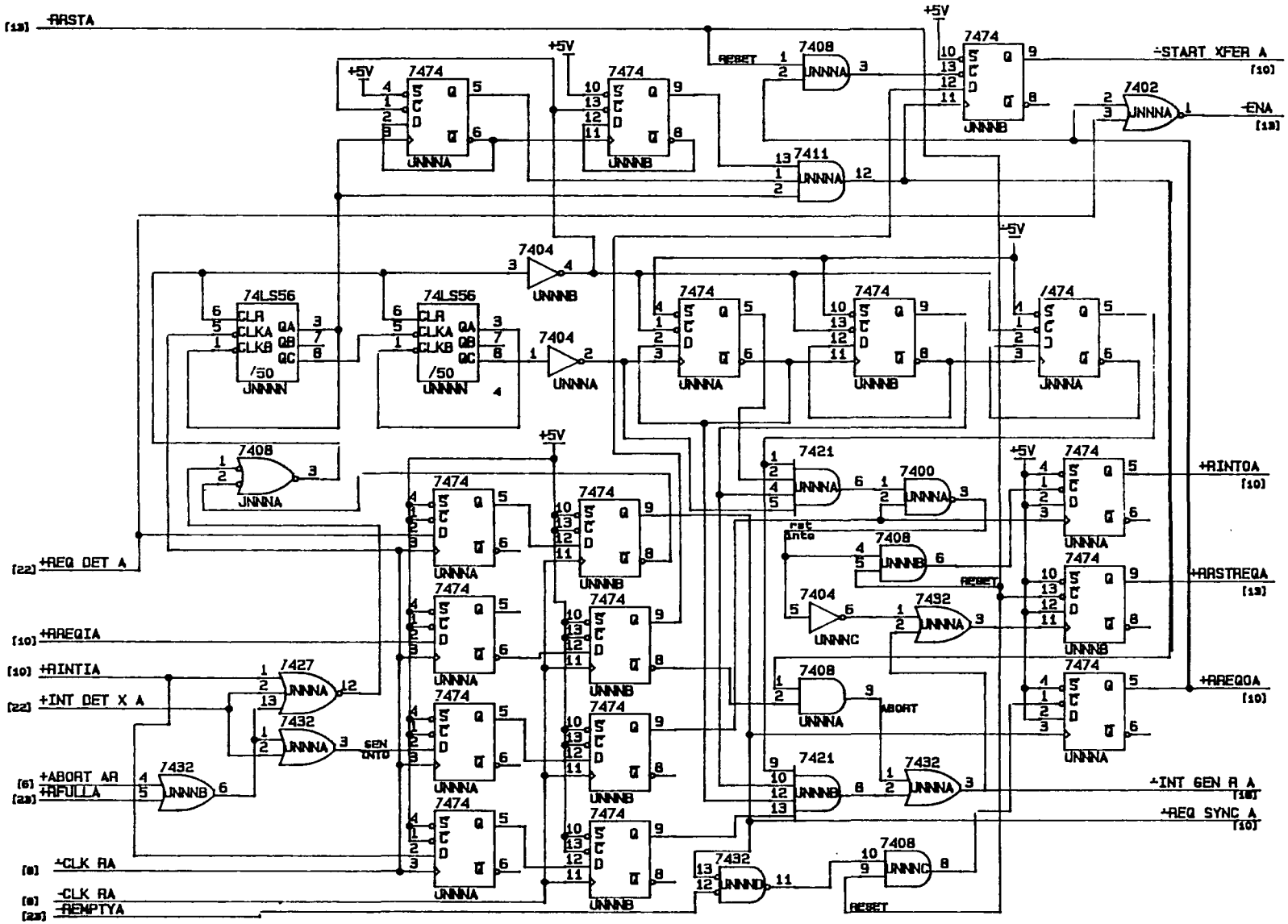

Figure 31. HSFONI Schematic 11 - Interface Control REC CTL Logic A

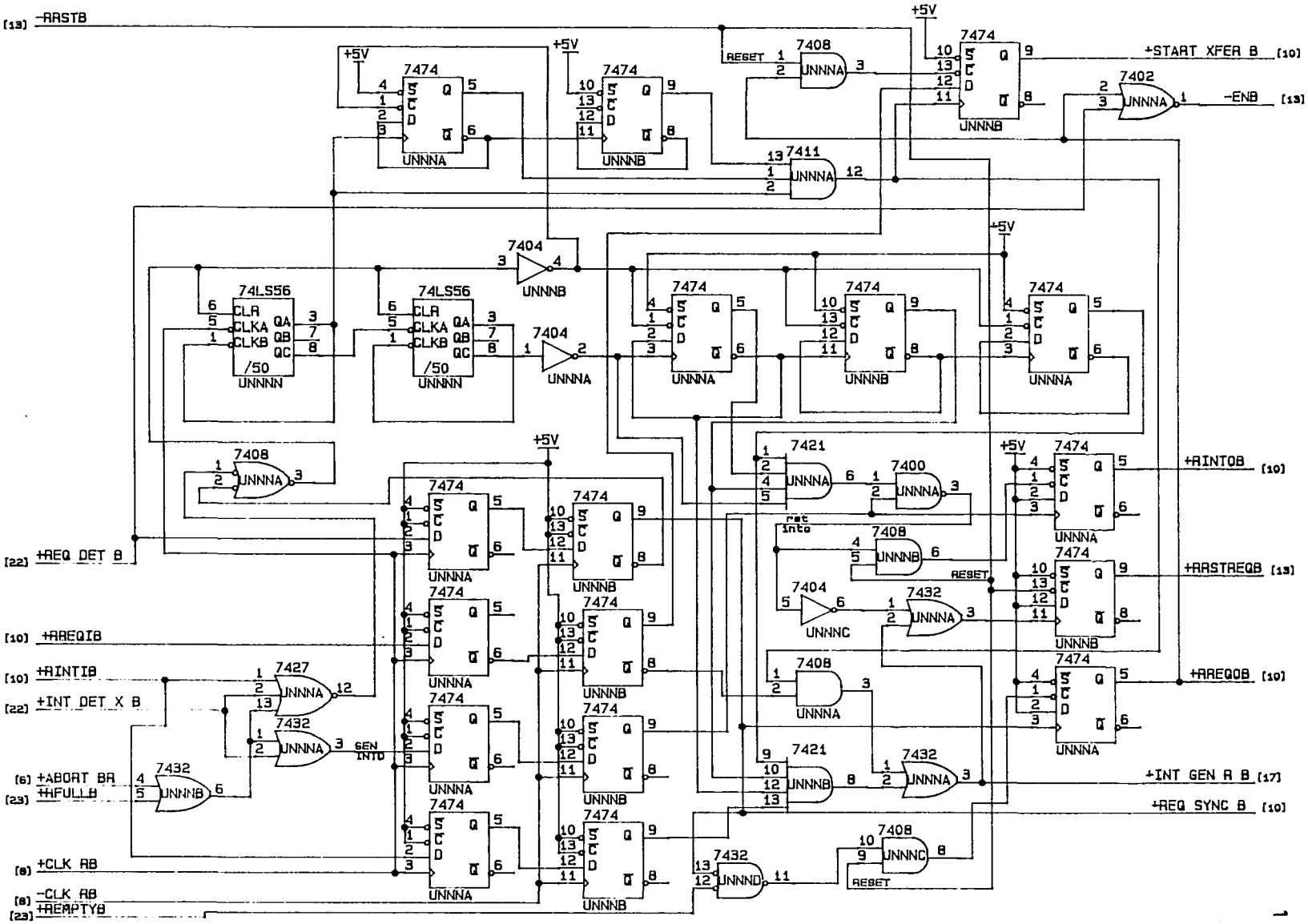

Figure 32. HSFONI Schematic 12 - Interface Control REC CTL Logic B

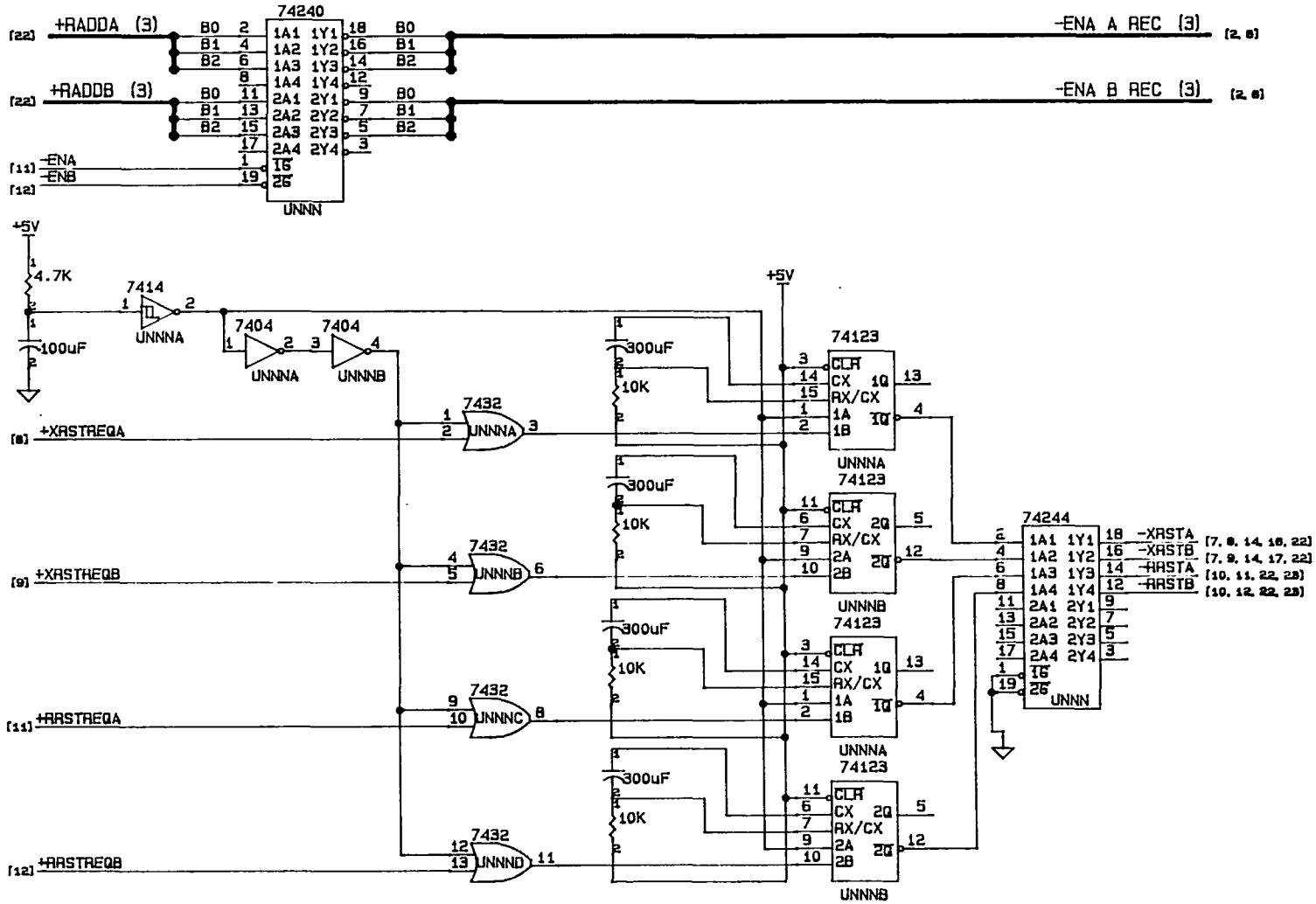

Figure 33. HSFONI Schematic 13 - Interface Control REC BUS CTL and Reset Logic

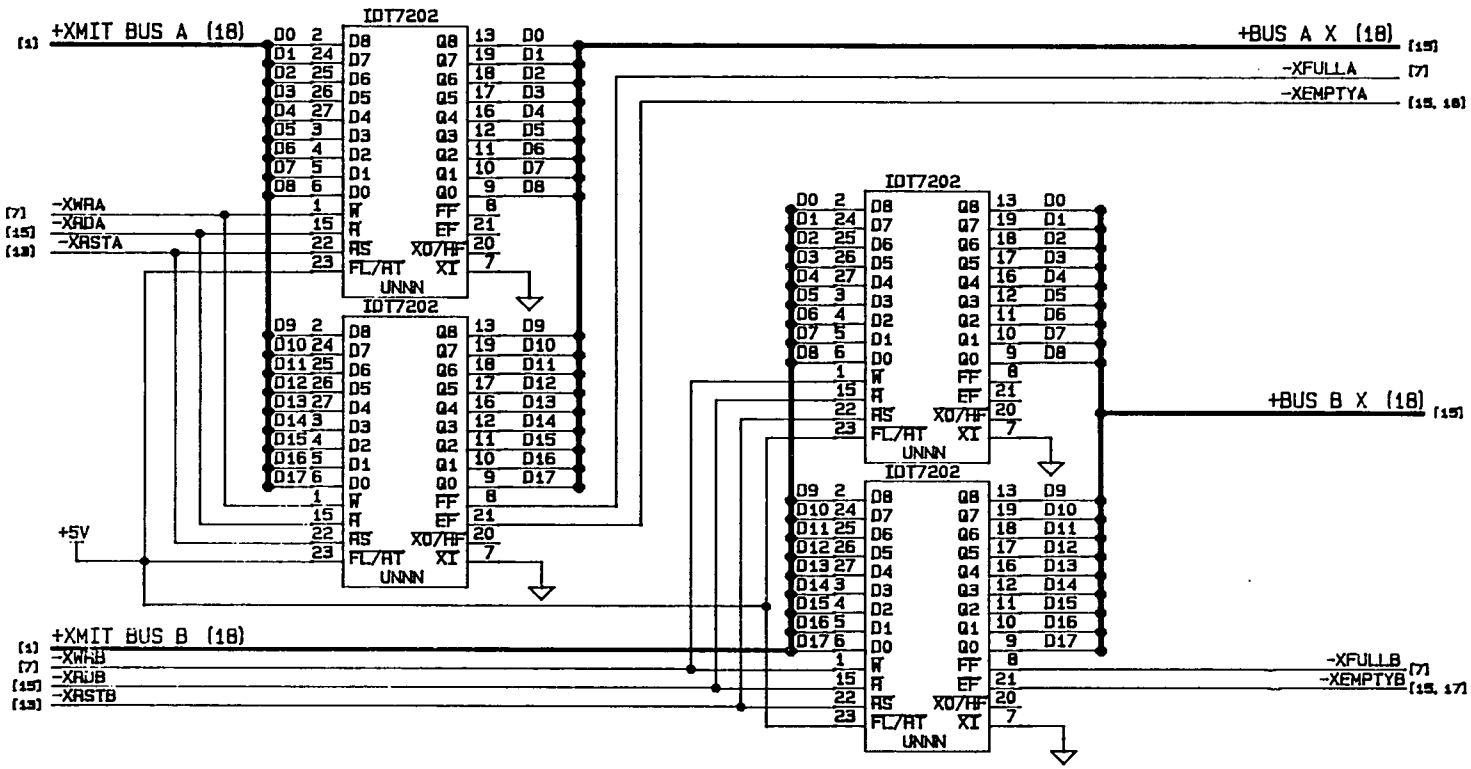

Figure 34. HSFONI Schematic 14 - Transmit Buffer

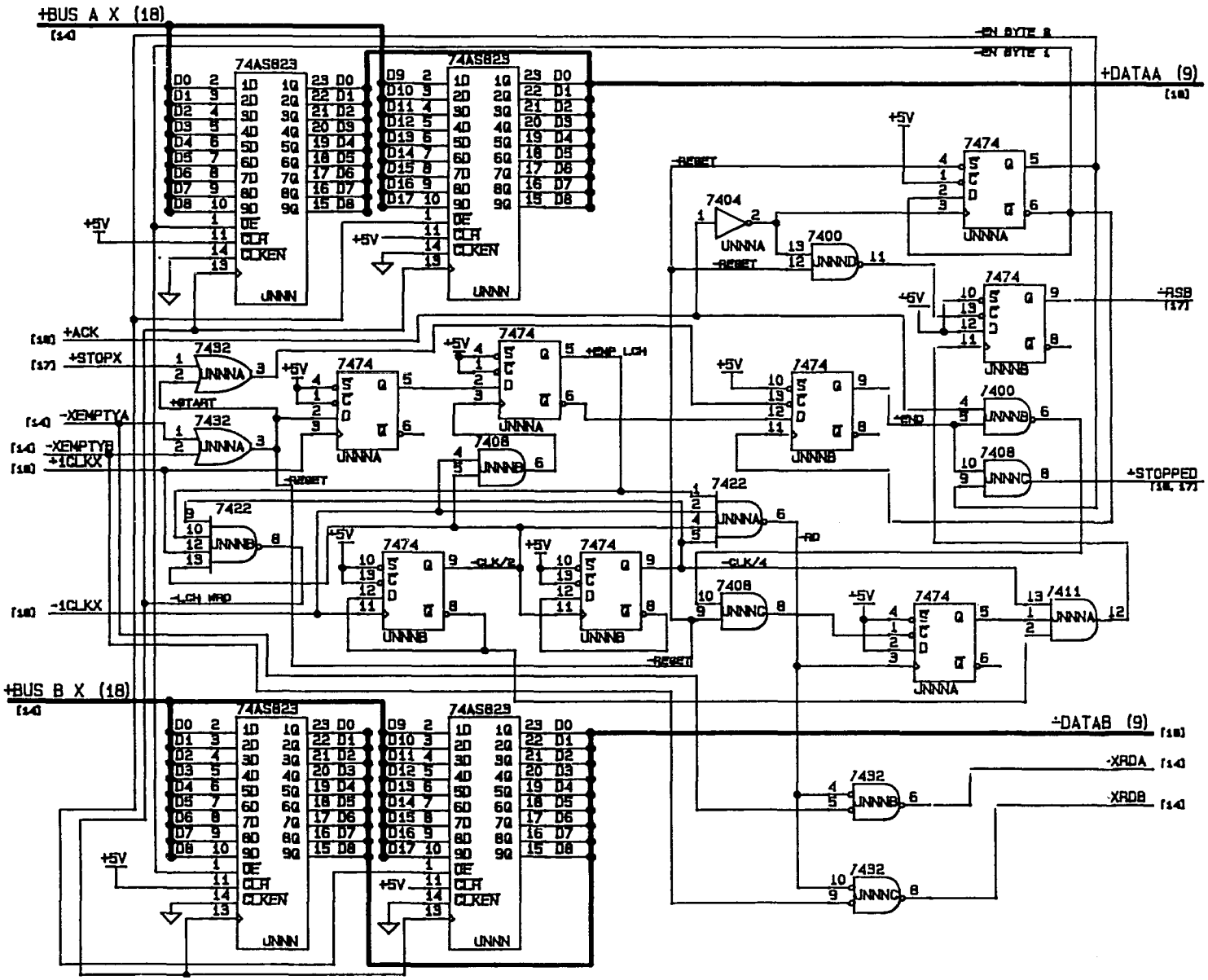

Figure 35. HSFONI Schematic 15 - Transmitter Data Control Logic

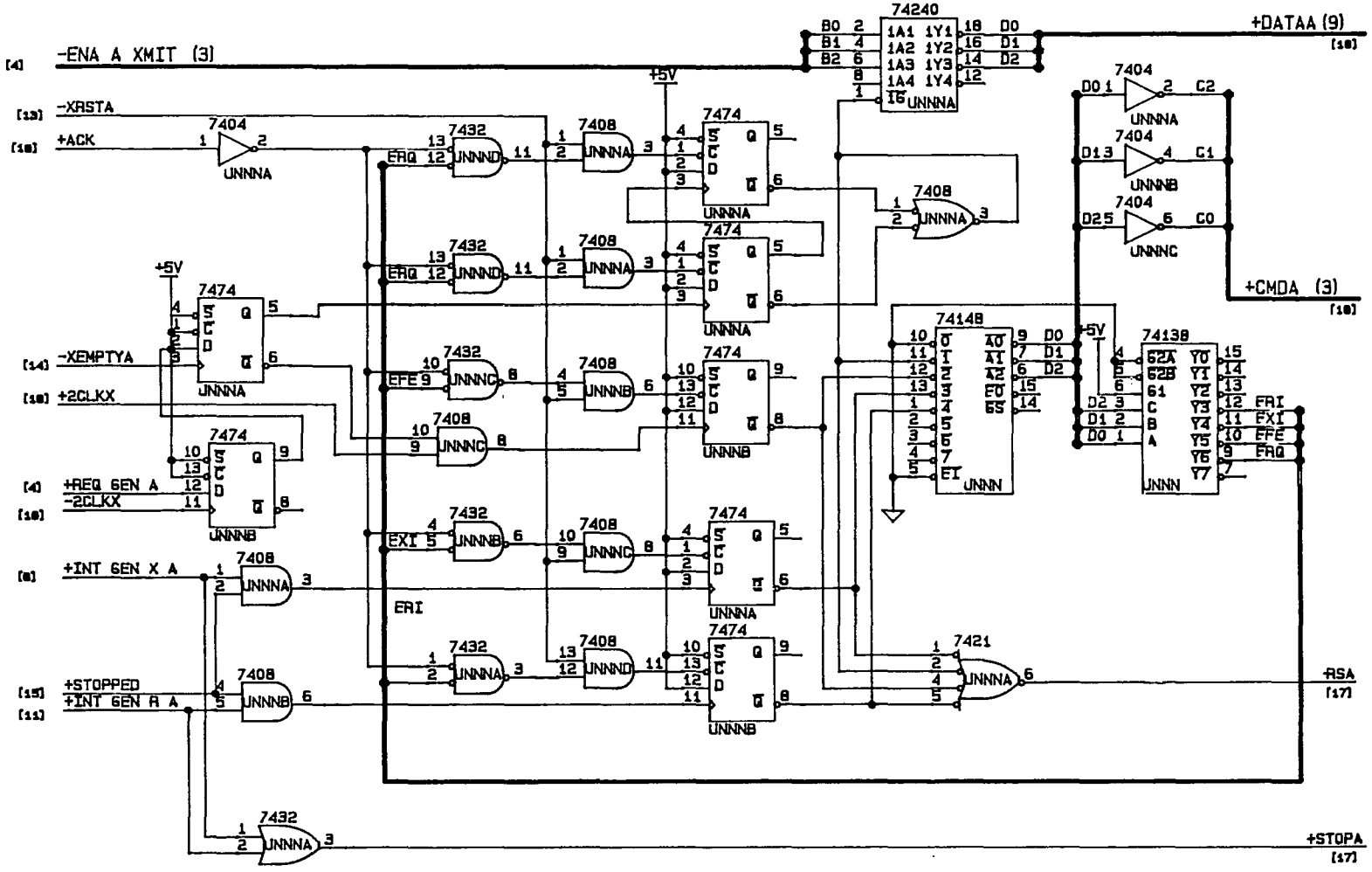

Figure 36. HSFONI Schematic 16 - Transmitter Command Control Logic 1

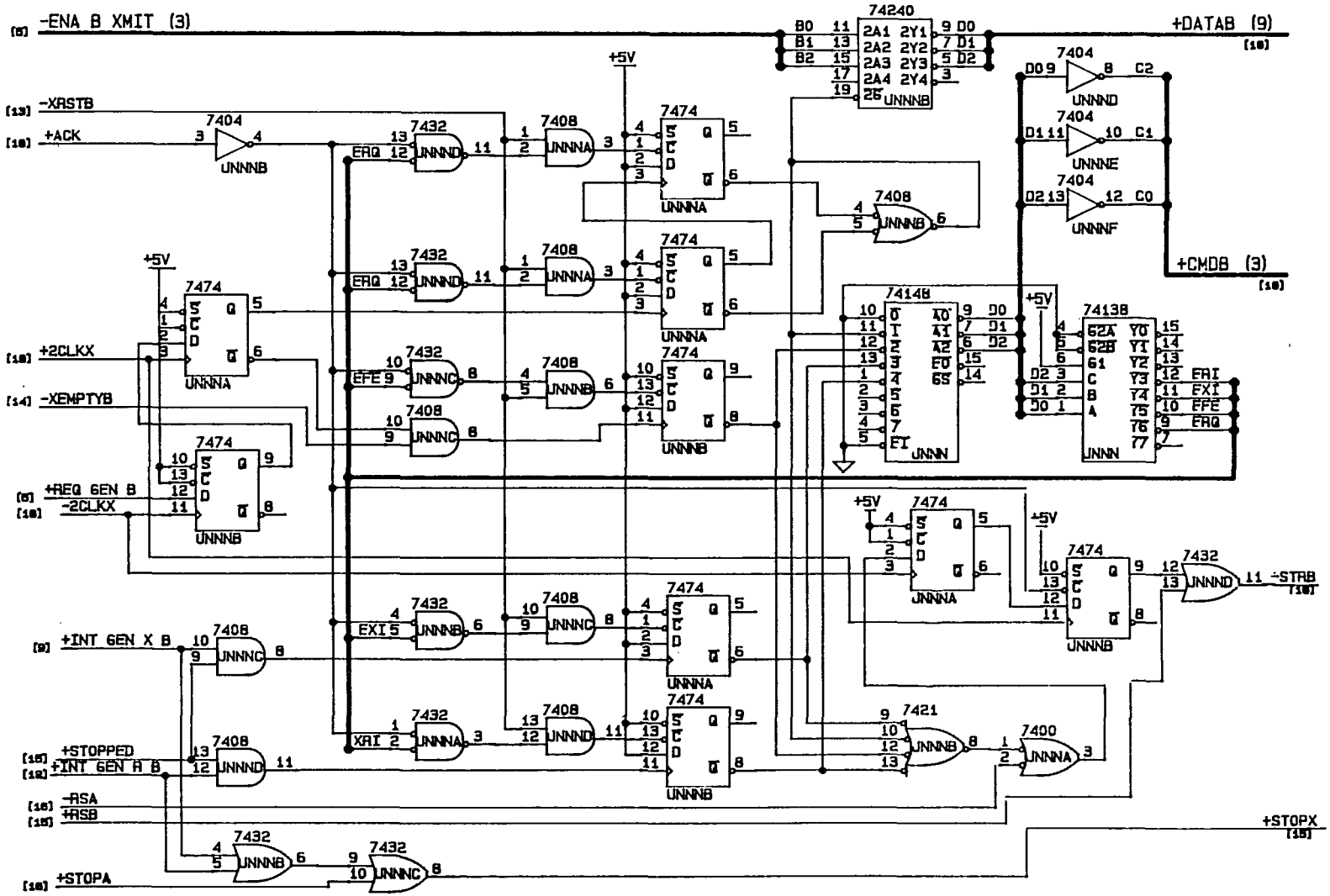

Figure 37. HSFONI Schematic 17 - Transmitter Command Control Logic 2

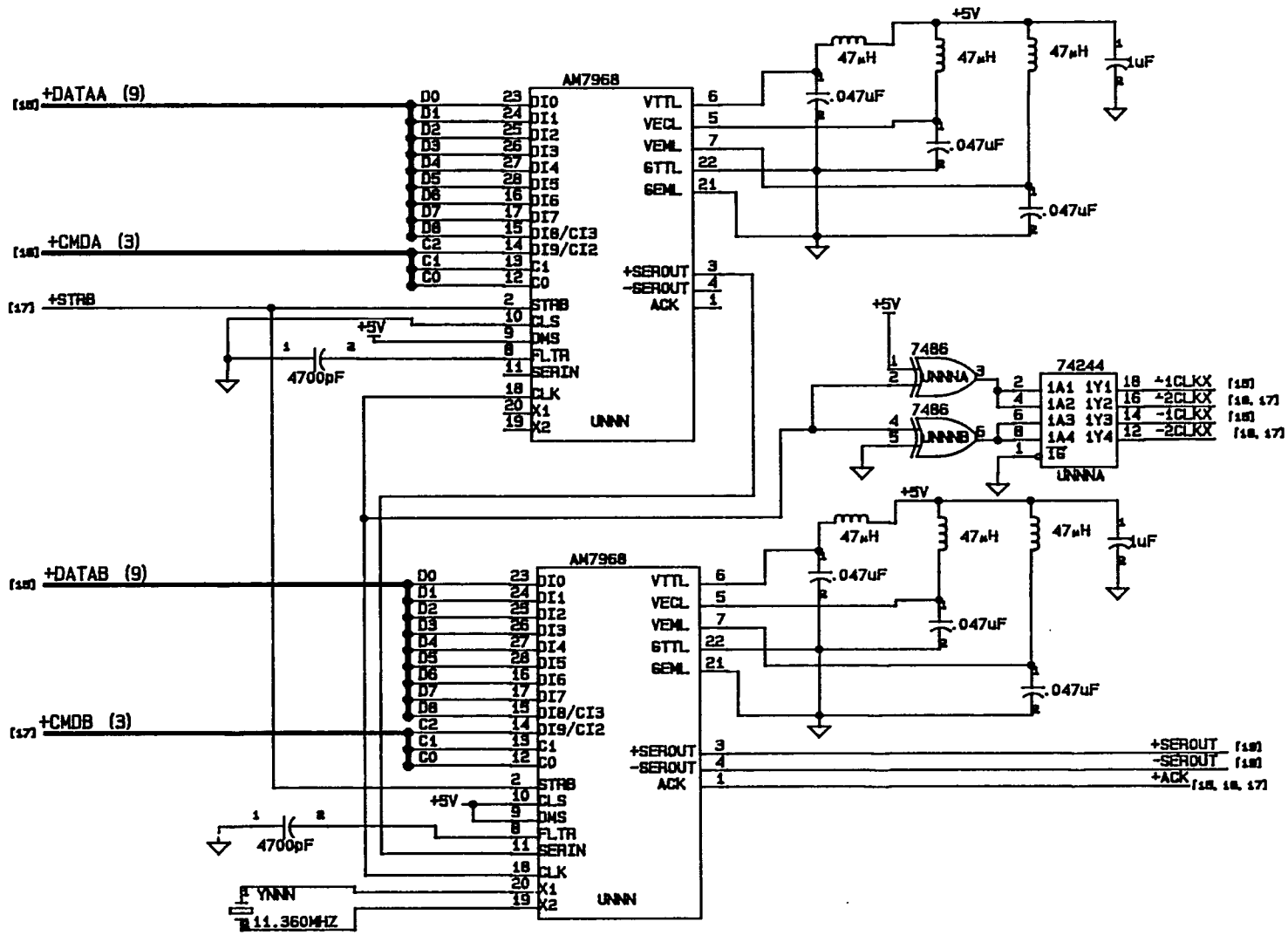

Figure 38. HSFONI Schematic 18 - Transmitter Transmit Logic

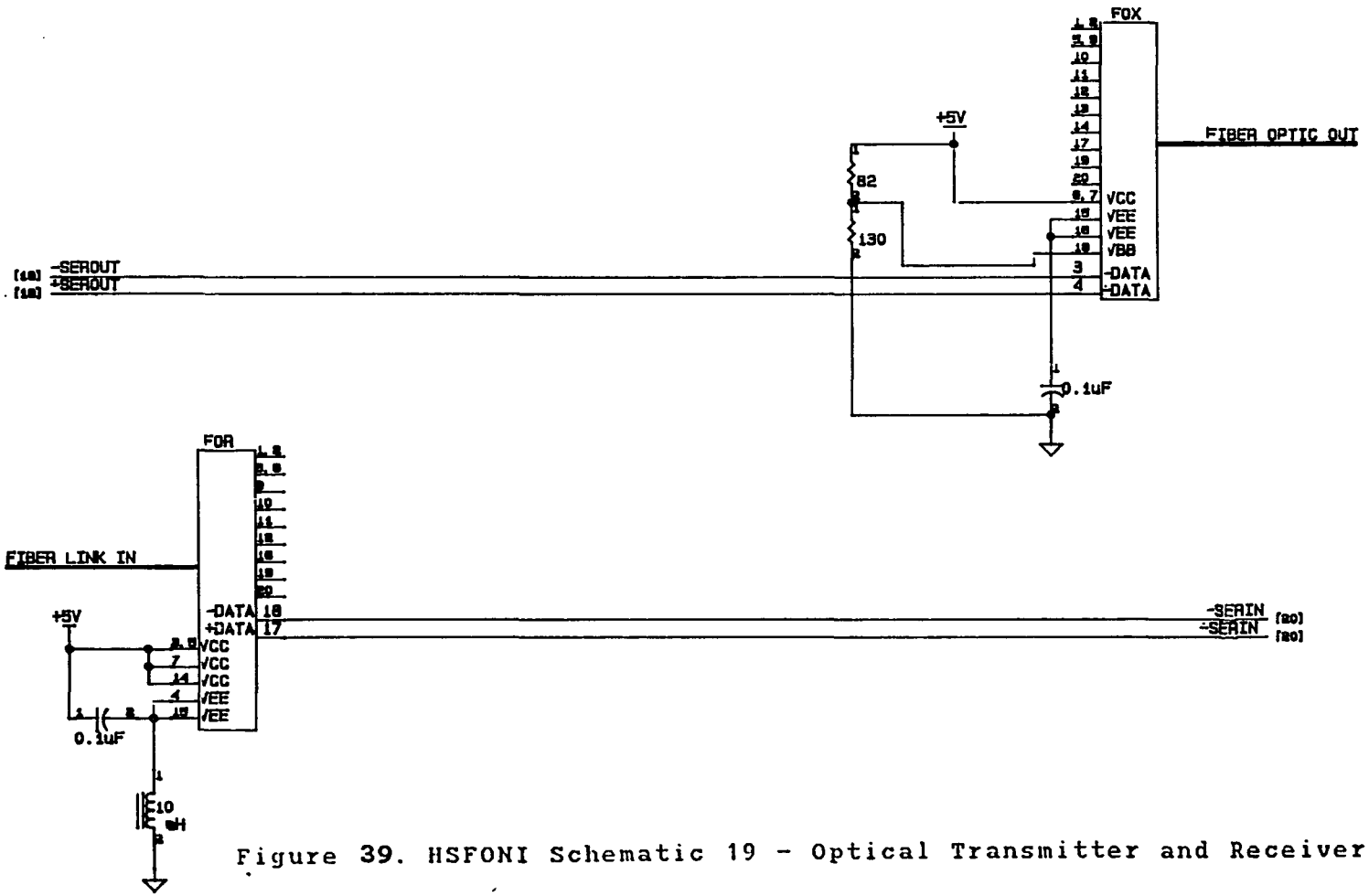

Figure 39. HSFONI Schematic 19 - Optical Transmitter and Receiver

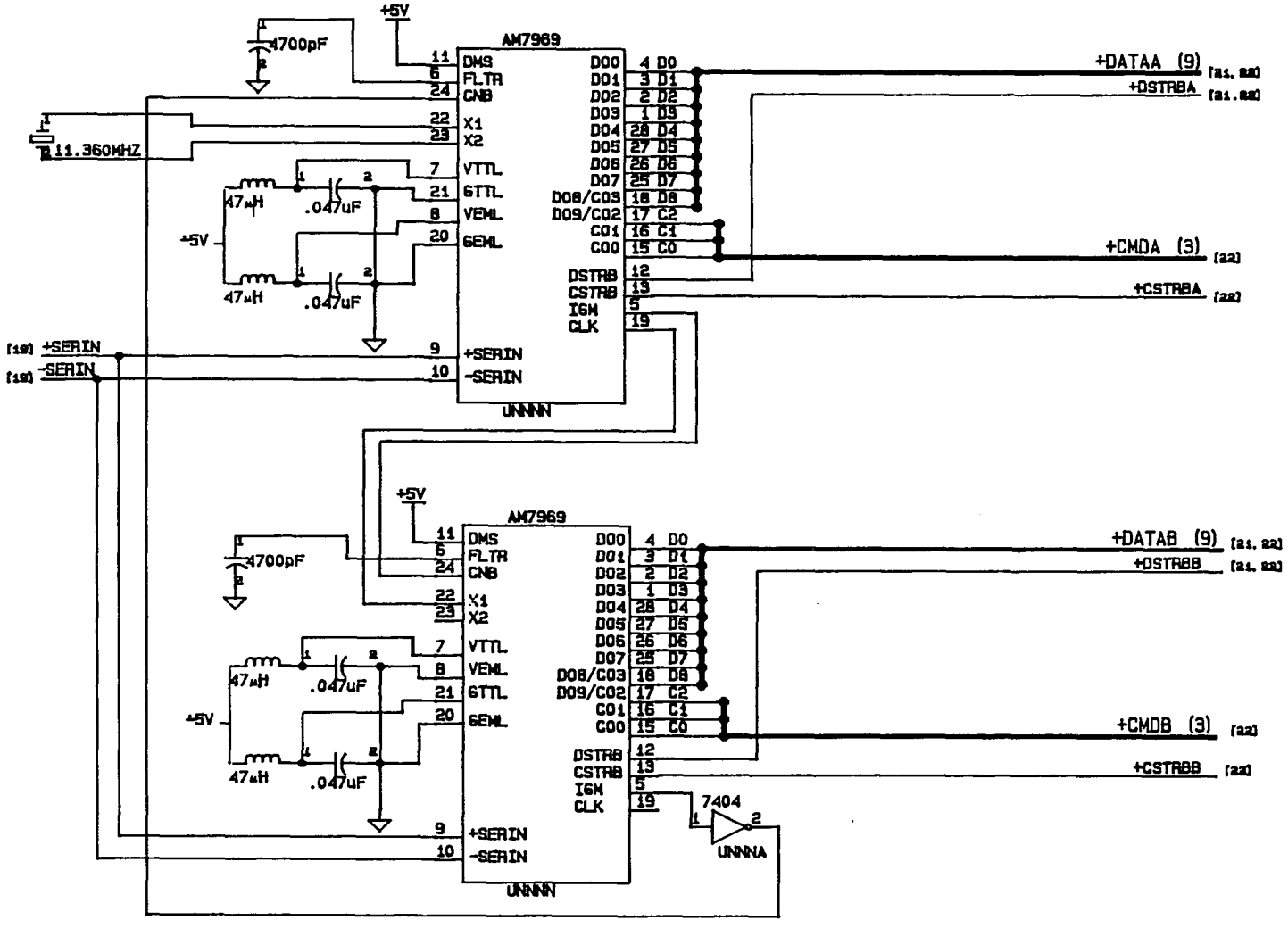

Figure 40. HSFONI Schematic 20 - Receiver Receive Logic

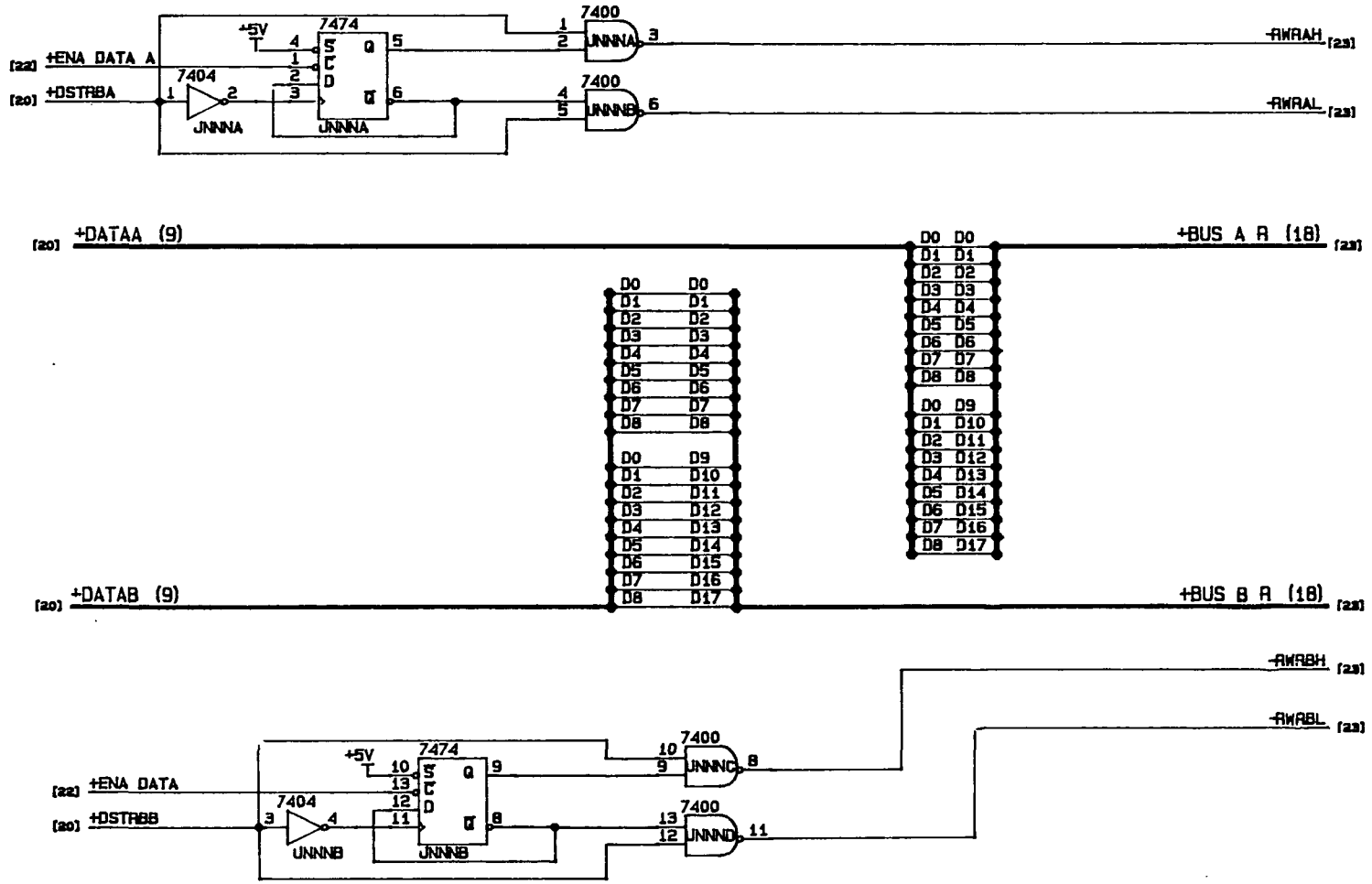

Figure 41. HSFONI Schematic 21 - Receiver Data Control Logic

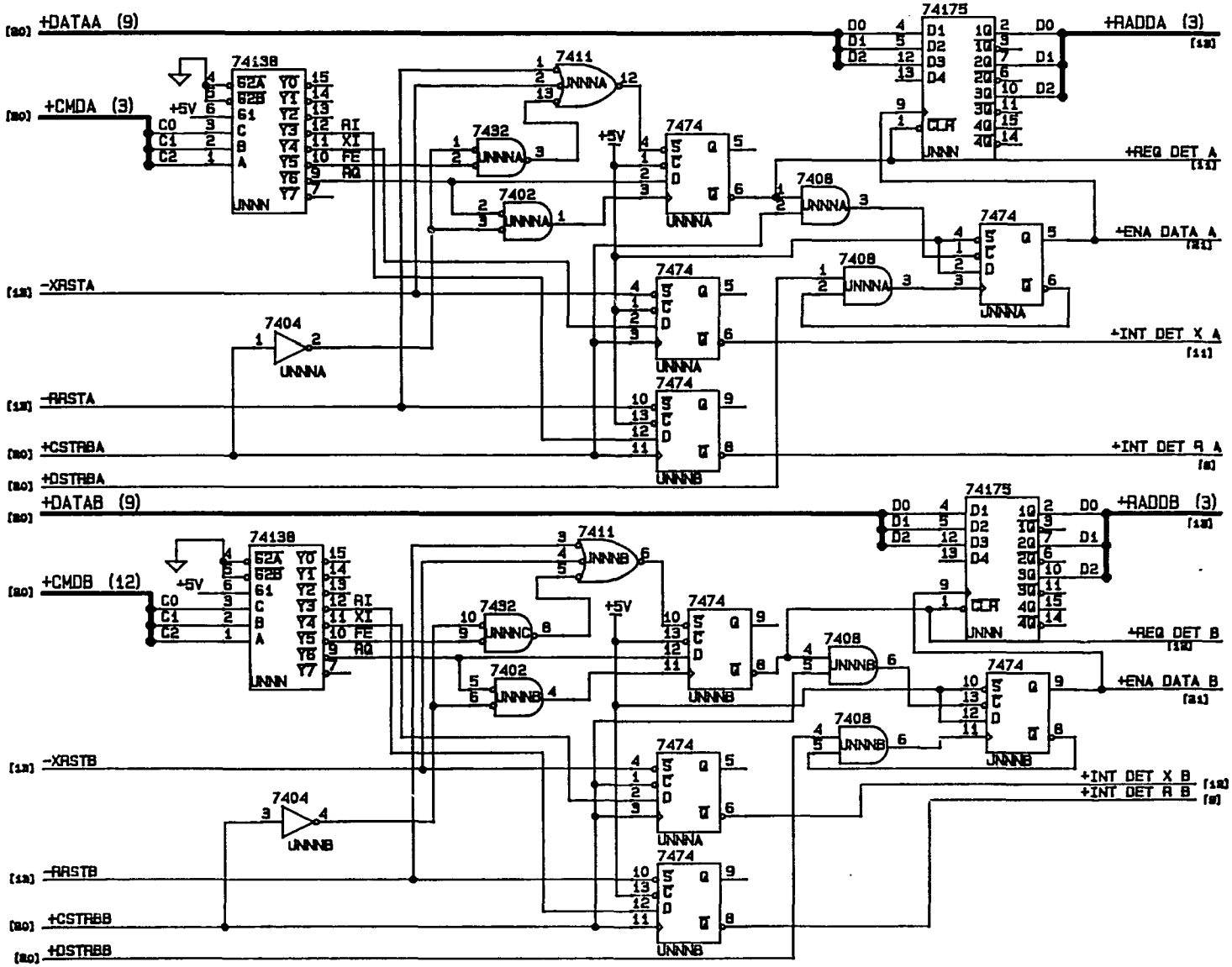

Figure 42. HSFONI Schematic 22 - Receiver Command Control Logic

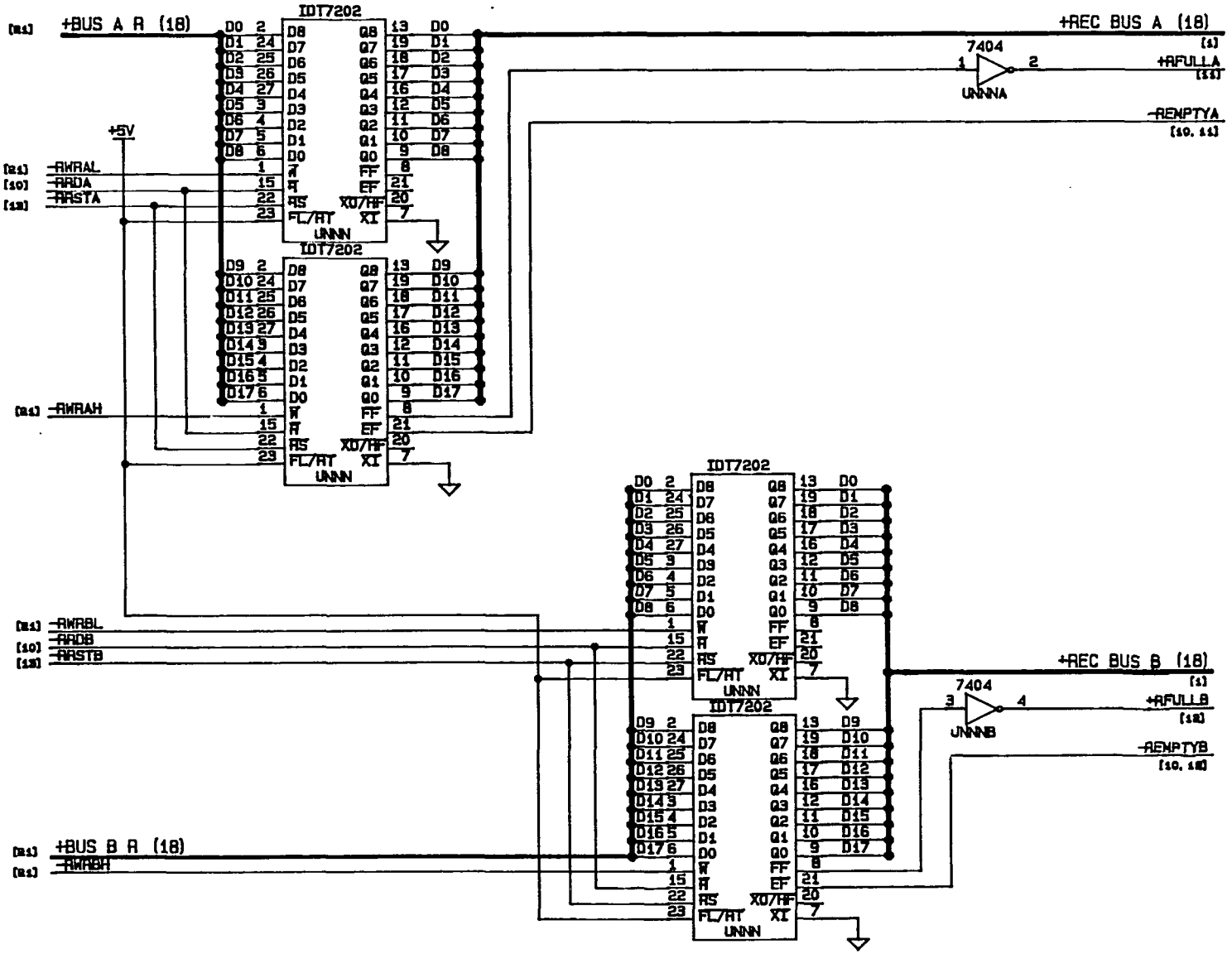

Figure 43. HSFONI Schematic 23 - Receiver Buffer

APPENDIX B: INTERFACE CONTROL STATE DIAGRAMS

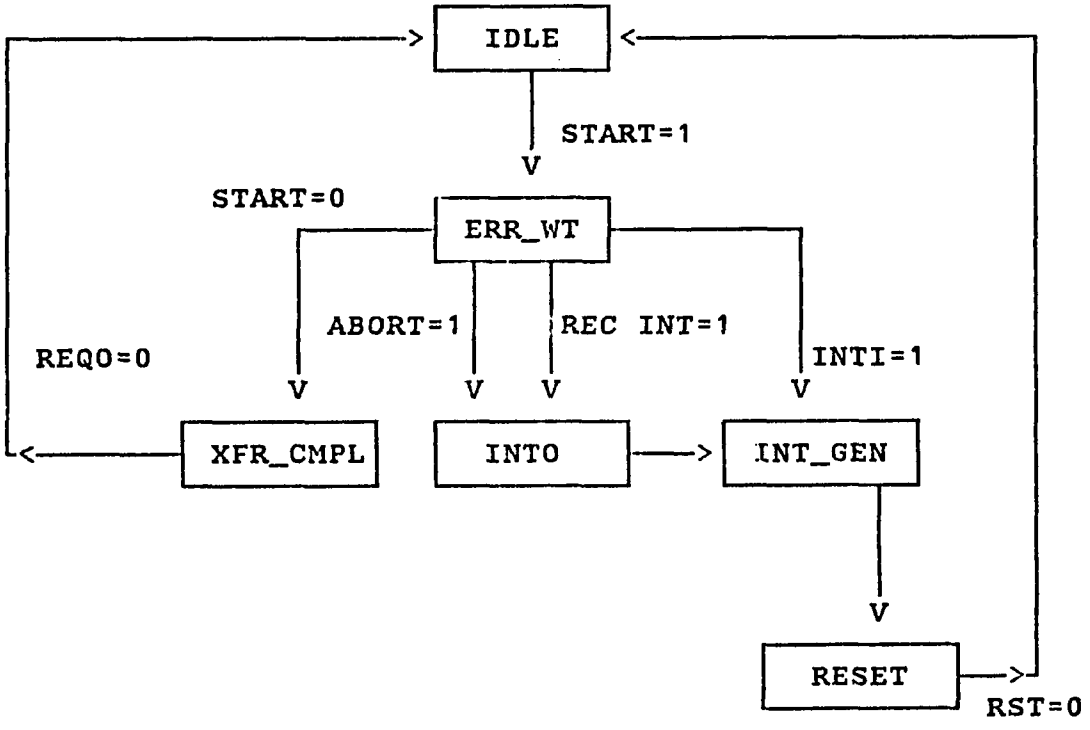

Figure 44. State Diagram
Interface Control XMT CTL Logic

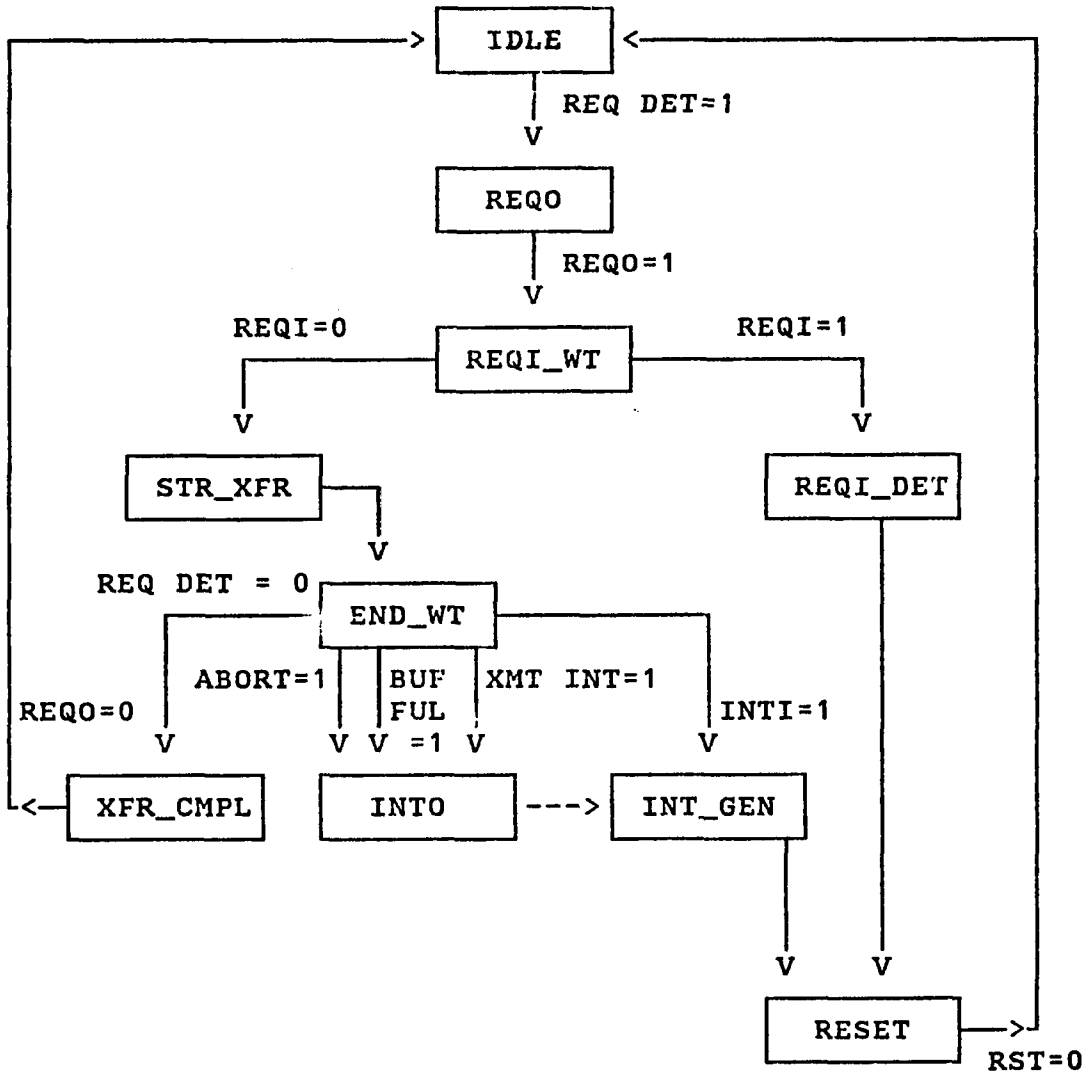

Figure 45. State Diagram
Interface Control REC CTL Logic

APPENDIX C: HSFONI CIRCUIT TIMINGS

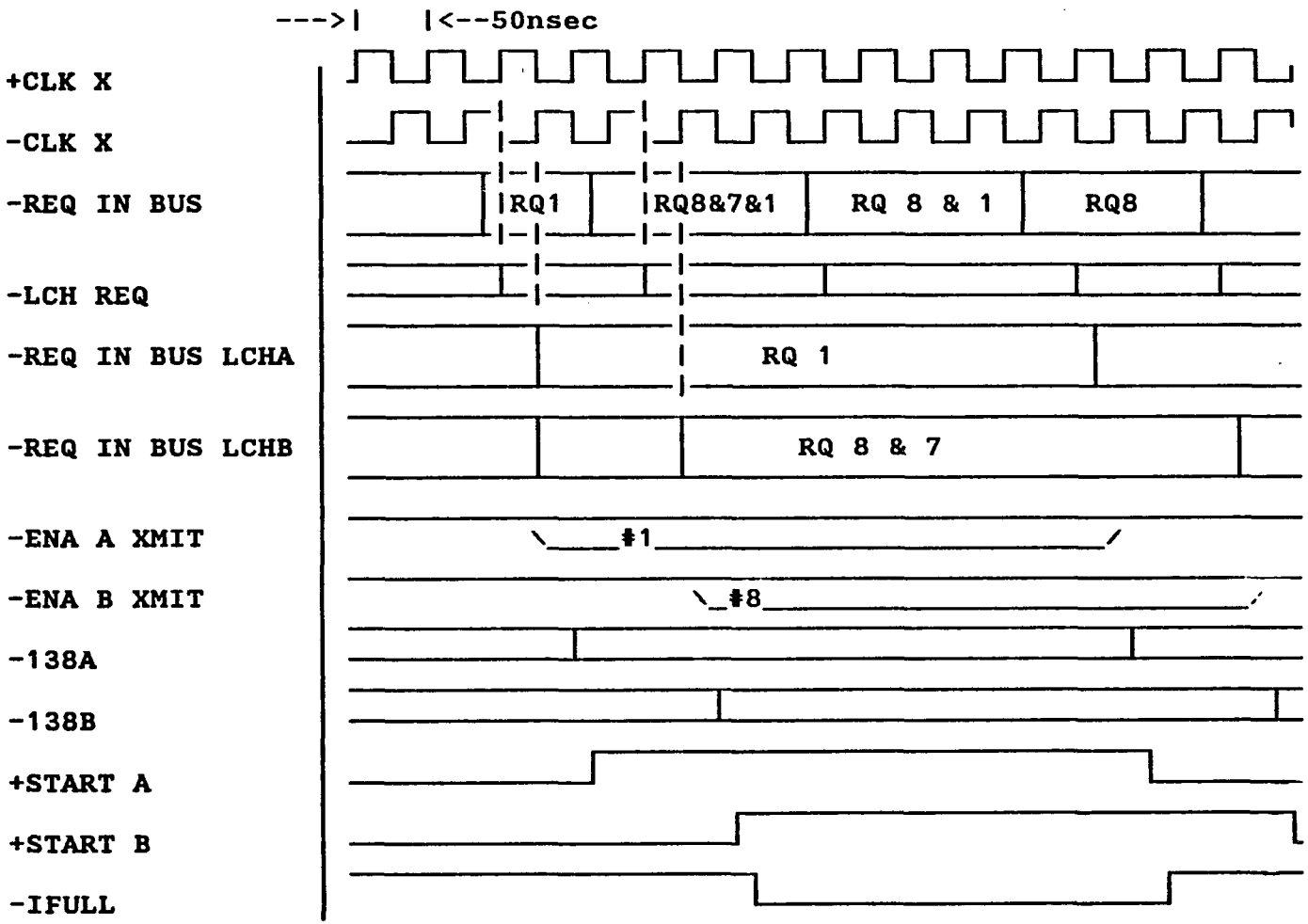

Figure 46. Interface Control Arbitration Timing

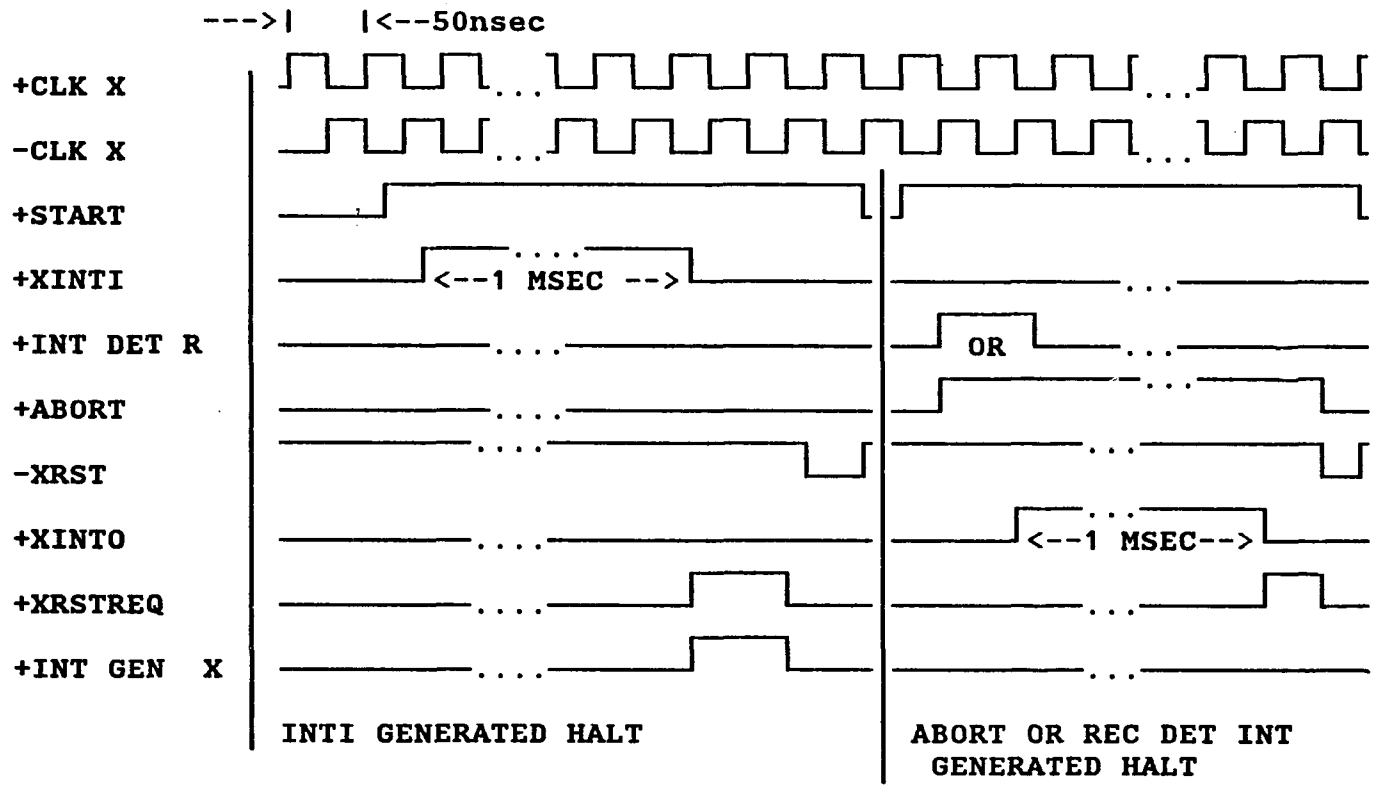

Figure 47. Interface Control XMT CTL A and B Timing

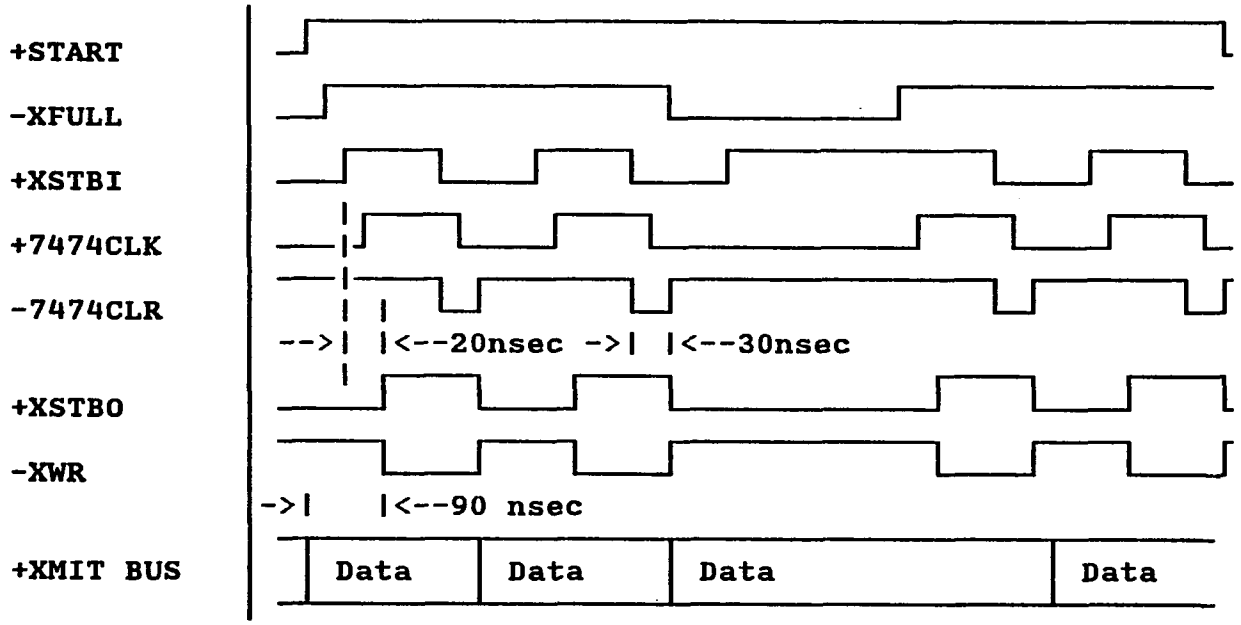

Figure 48. Interface Control XMT STB A and B Timing

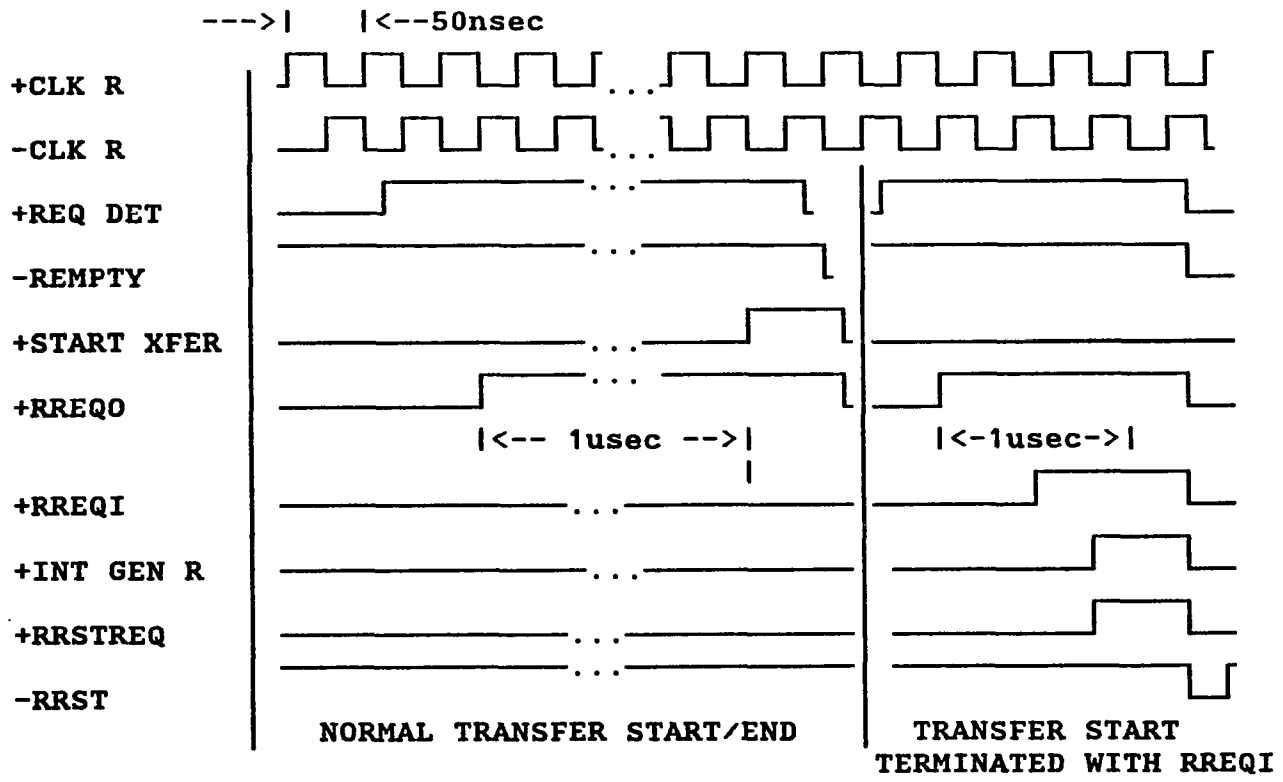

Figure 49. Interface Control REC CTL A and B Timing 1

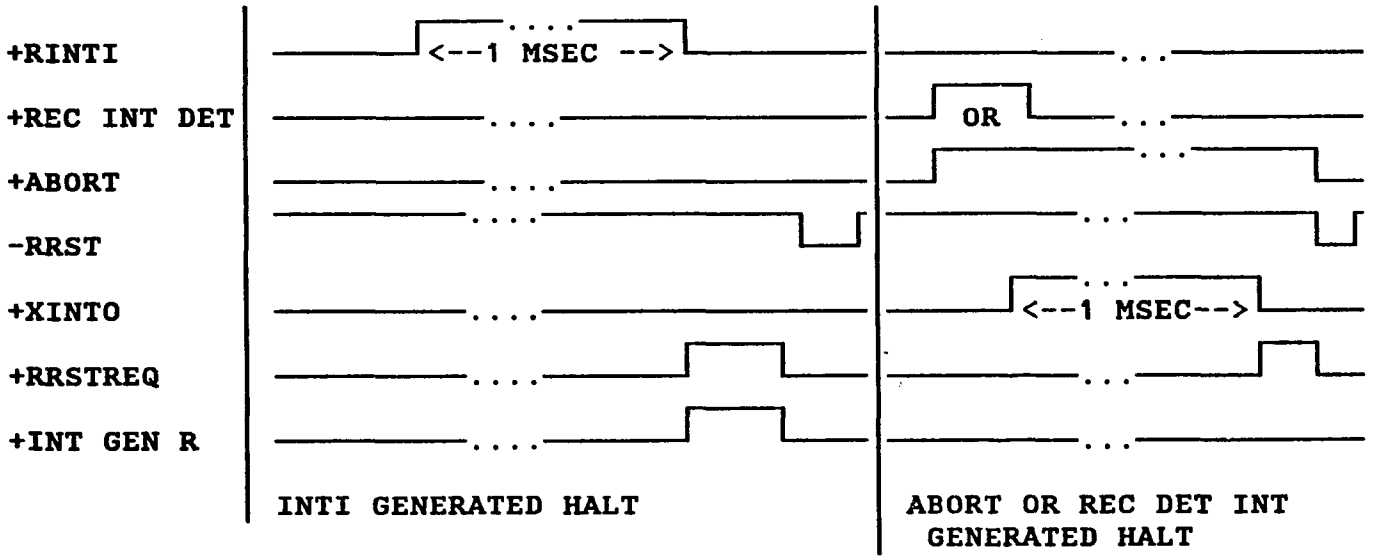

Figure 50. Interface Control REC CTL A and B Timing 2

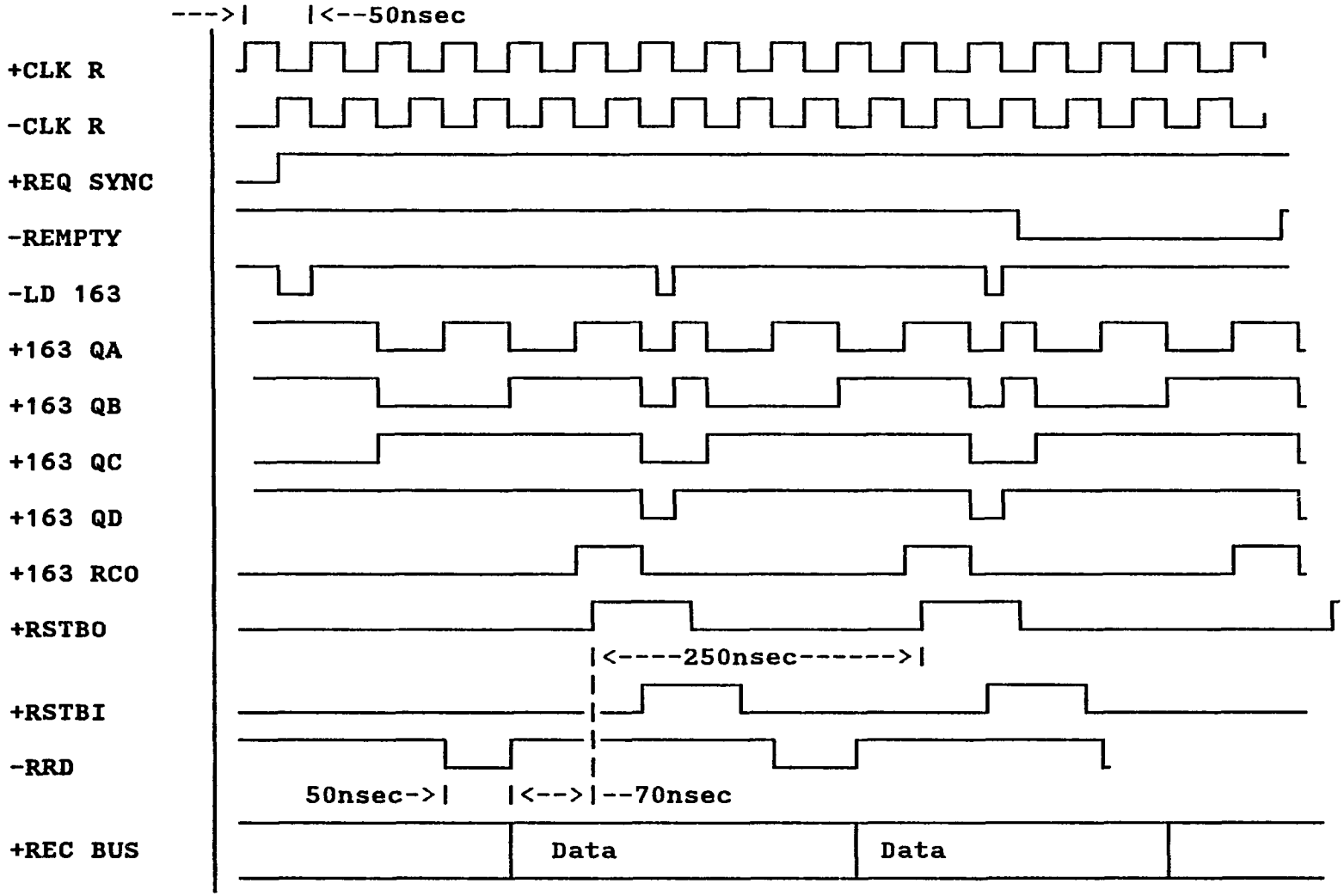

Figure 51. Interface Control REC STB A and B Timing

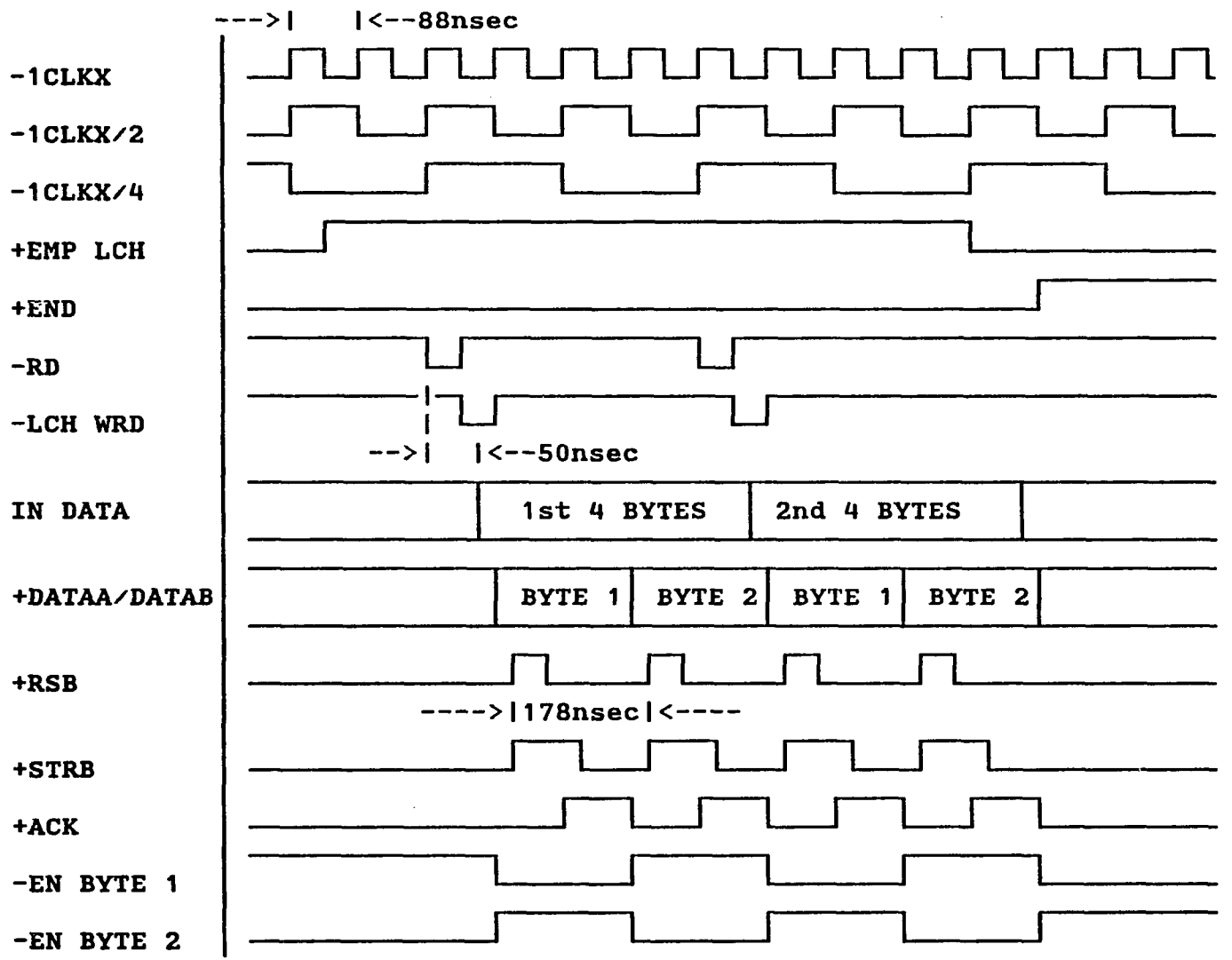

Figure 52. Transmitter Data Control Timing

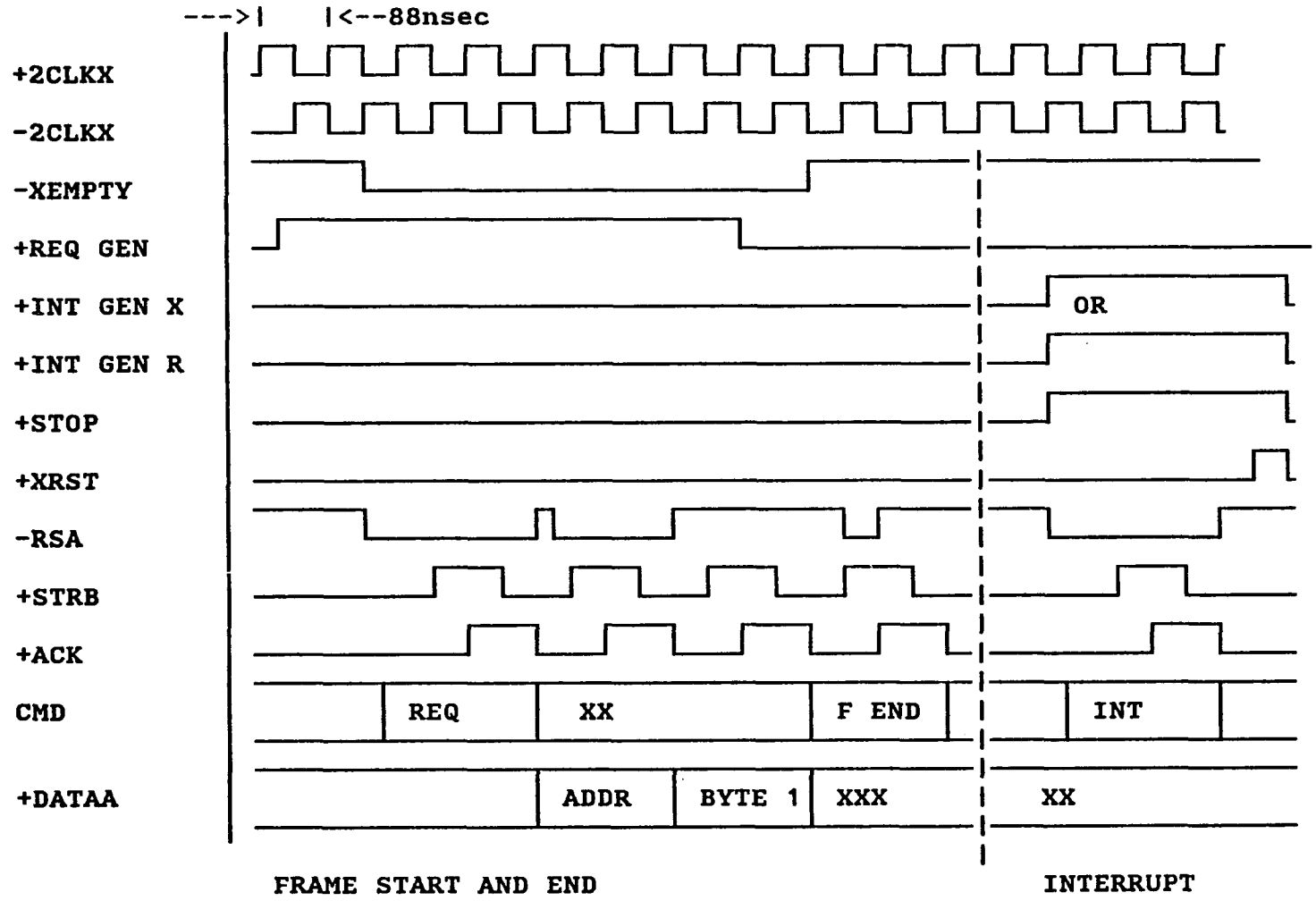

Figure 53. Transmitter Command Control Timing

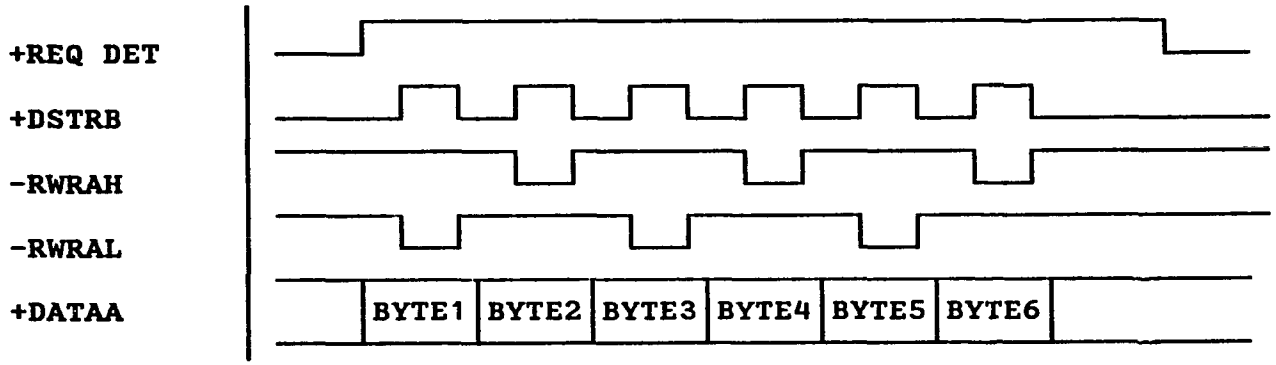

Figure 54. Receiver Data Control Timing

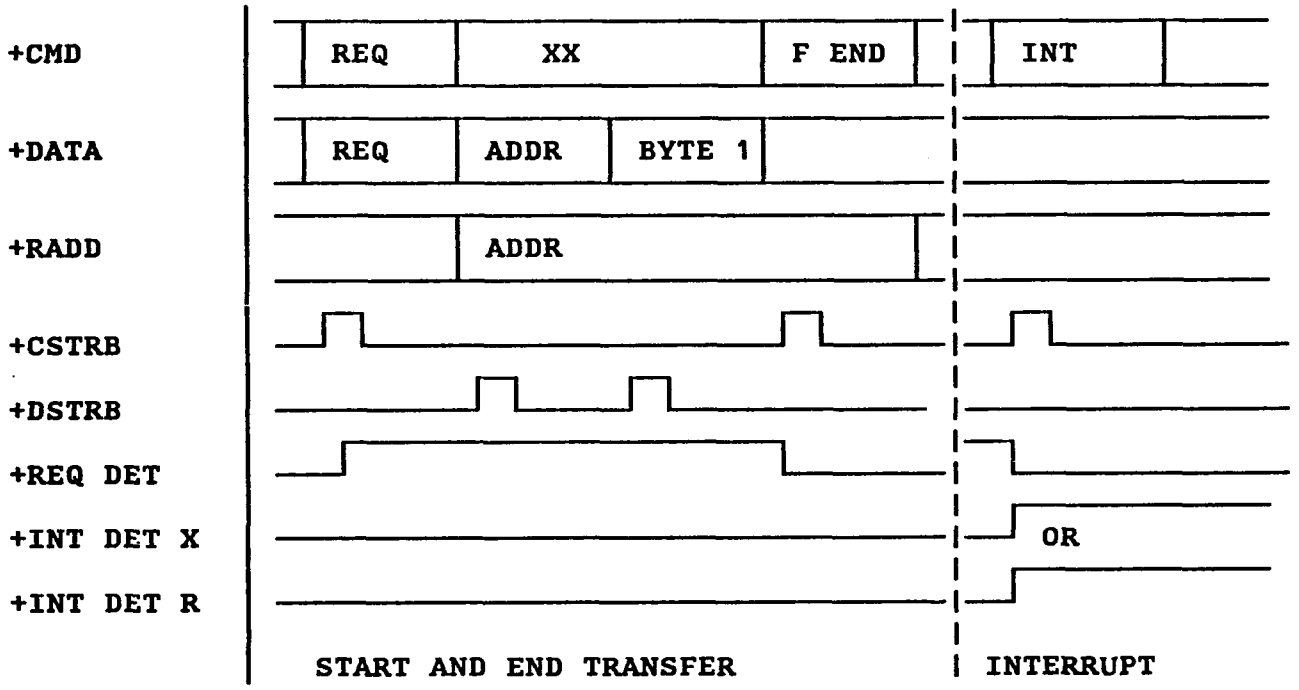

Figure 55. Receiver Command Control Timing

REFERENCES

Am7968/Am7968 Application Note. Sunnyvale: Advanced Micro Devices (AMD), Incorporated, 1986.

Gary Connor, Paul Scott, and Sarosh Vesuna. "Serial data races at parallel rates for the best of both worlds," Electronic Design, January 22, 1987 pp. 79-83.

Designers Guide To Fiber Optics. Harrisburg: Amp, Incorporated, 1982.

Digital Imaging and Communications. Washington, D.C.: American Colledge of Radiology-National Manufactures Association (ACR-NEMA), 1985.

A Guide to Fiber Optics System Design. Richmond: Belden/Cooper Cooper Industries, 1985.

S.S. Hedge, A,O, Gale, J.A. Giunta. "AT&T PACS Architecture.", Proceeding of SPIE, Volume 626 part two, 1986 p. 618-625.

IDT7201/IDT7202 Application Note. Santa Clara: Integrated Device Technology, Incorporated, 1984.

Lay, Richard W. "Mux Packs Many More Signals Into Optical Fiber," Electronic Engineering Times, March 2, 1987 p. 27 and p. 47.

Martinez, R. Archwamety, C. and Nemat, M. Working Papers and Draft Reports for TDIS Image Network and Data Base Archive Systems. Tucson: University of Arizona, 1986.

Palais, Joseph C. Fiber Optic Communications. Englewood Cliffs: Prentice-Hall, Incorporated, 1984.

Siecor 200 Mbaud Receiver Module. Research Triangle Park: Siecor Corporation, 1986.

Siecor 200 Mbaud Transmitter Module. Research Triangle Park: Siecor Corporation, 1986.

Senior, John M. Optical Fiber Communications Principles and Practice. Prentice-Hall, Incorporated, 1985.

Stallings, William Local Networks An Introduction. New York: Macmillam Publishing Co., 1984.

Tunick, Diane "Digital GaAs ICs make their move," Electronic Design, April 16, 1987 p. 35-36 and p. 40.

F.van der Voorde, R.Arenson, H.Kundel, W.Miller, D.Epstein, W.Gefter, S.Seshadri, I.Brikman, S.Khalsa. "Development of a Physician-Friendly Digital Image Display Console", Proceeding of SPIE, Volume 626 part two, 1986 p. 541-548.

SELECTED BIBLIOGRAPHY

Though these entries are not cited in the text, they are included in this section for the convenience of the reader in obtaining further information of a detailed or explanatory nature.

Belden Master Catalog 885. Richmond: Belden/Cooper Industries, 1985.

Fiber Data Sheet, Signal Hill: American Fiber Optics Corporation, 1985.

Lightguide Fiber Data Sheet, AT&T, 1985.

Optimate Data Links Data Sheet, Harrisburg: AMP Incorporated, 1986.

The TTL Data Book , Dallas: Texas Instruments, 1985.