SLAC-PUB-2990
October 1982
(I/A)

NEW DEVELOPMENTS IN SEGMENT ANCILLARY LOGIC FOR FASTBUS*

Helmut V. Walz and Boris Bertolucci
Stanford Linear Accelerator Center
Stanford University, Stanford, California 94305

Abstract

Segment Ancillary Logic hardware for FASTBUS systems provides logical functions required in common by all devices attached to a segment. It controls the execution of arbitration cycles, and geographical address cycles, and generates the system handshake responses for broadcast operations. The mandatory requirements for Segment Ancillary Logic in the FASTBUS specifications are reviewed. A detailed implementation based on ECL logic is described, and the hardware to be used on an ECL cable segment for an experimental FASTBUS system at SLAC is shown.

I. Introduction

FASTBUS is a proposed standard data bus for modular, high-speed data acquisition and control systems. It has been developed by the Fast System Design Group of the U.S. NIM Committee with participation from the European ESONE Committee. FASTBUS systems can be configured from multiple bus segments. These segments are able to operate independently or link together selectively for exchange of data. Two or more crate segments may be connected by means of a cable segment. The implementation of segments and execution of some operations require circuitry which is common to all devices residing on such segments. This is referred to as Ancillary Logic and contains hardware for arbitration timing control, geographical addressing, system handshake generation for broadcast cycles, run-halt control, and signal line terminations for the segment bus. The requirements for Ancillary Logic contained in the FASTBUS Specifications will be reviewed and discussed in detail.

An ECL design for a standard crate segment Ancillary Logic unit has been developed. Design details will be shown to illustrate implementation of the FASTBUS specifications.

A prototype Ancillary Logic Module for a differential ECL cable segment has been built.

This hardware will be used as part of an experimental FASTBUS data acquisition and control system for one of the particle detectors at SLAC. The presentation in this paper assumes basic familiarity with the FB Specification.

II. Basic Ancillary Logic Requirements

This discussion of Ancillary Logic (ANC) requirements is based on Section 7 of the FASTBUS (FB) Tentative Specification.[1] Two block diagrams and two tables of definitions and logic equations are presented to illustrate these requirements. The diagrams are not based on any particular hardware realization. Hence inputs to flip-flops, registers, latches, and timers assume only logic level sensitivity; for flip-flops S = R = 1 inputs cause no change, CLR input overrides S and R; an enabled timer asserts its output at the end of its time interval; the timer is reset and initialized by the CLR input.

A. Arbitration Timing Control Section

The ATC section is shown in Fig. 1 and Table 1. This contains logic functions for ATC, Arbitration Inhibit (AI), System Handshake (SHL), and Run-Halt (RH) Control.

Run-Halt Logic monitors the manual RH switch of the segment. A halt request from the switch inhibits new arbitration cycles. Upon completion of an arbitration cycle in progress and release of the bus by the current master, the Bus Halted (BH) and AK lines are asserted.

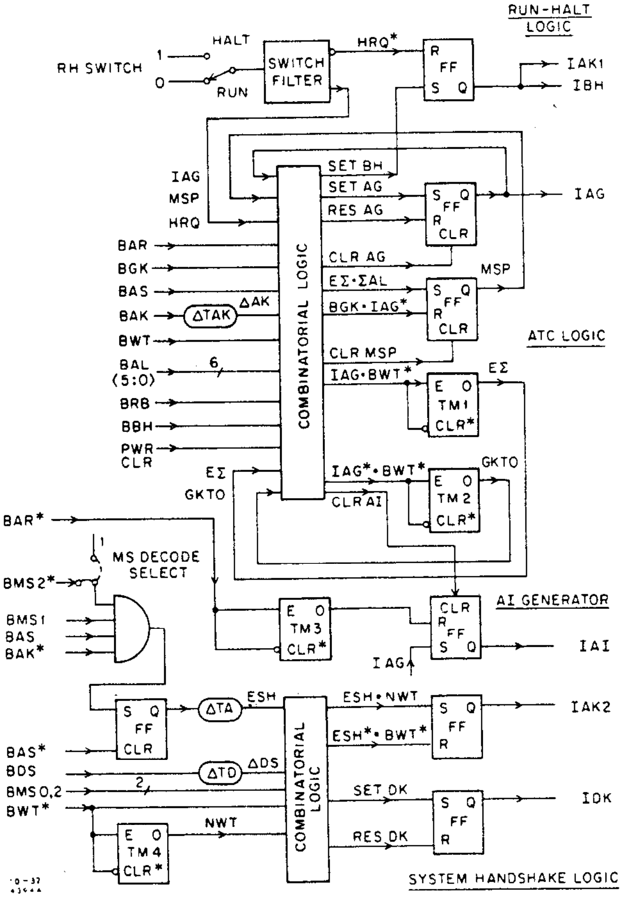

Fig. 1. ATC section basic block diagram.

Table 1. ATC section definitions and logic equations.

Definitions

ΔTAK	-	Bus Cleanup Time after AK(d)
TM1	-	Arbitration Timer for AG = 1 (EΣ)
TM2	-	GK(u) Timeout (GKTO)
TM3,4	-	Bus Delay Timers
ΔTA	-	Broadcast Address Response Delay
ΔTD	-	Broadcast Data Response Delay
HRQ	-	Halt Request
MSP	-	Mastership Pending
EΣ	-	Enable AL Sum
PWR CLR	-	Power-on Clear
NWT	-	No Wait Asserted
ESH	-	Enable System Handshake

Logic

SET BH	=	$\text{HRQ}\cdot(\text{MSP} + \text{IAG} + \text{BGK} + \text{BAS} + \text{BWT} + \Delta\text{AK})^{*}$
SET AG	=	$\text{BAR}\cdot\text{HRQ}^{*}\cdot\text{BGK}^{*}\cdot\text{BWT}^{*}\cdot\text{MSP}^{*}$
RES AG	=	$\text{BAS}^{*}\cdot\Delta\text{AK}^{*}\cdot\text{BGK}^{*}\cdot\text{BWT}^{*}\cdot\text{MSP}$
CLR AG	=	$\text{E}\Sigma\cdot\Sigma\text{AL}^{*} + \text{PWR CLR} + \int\text{BRB}\cdot\text{BBH}^{*}$
CLR MSP	=	$\text{GKTO} + \text{PWR CLR} + \int\text{BRB}\cdot\text{BBH}^{*}$
CLR AI	=	$\text{PWR CLR} + \int\text{BRB}\cdot\text{BBH}^{*}$
SET DK	=	$\text{ESH}\cdot\text{BMS2}^{*}\cdot\Delta\text{DS}\cdot\text{NWT}$
RES DK	=	$\Delta\text{DS}^{*}\cdot\text{NWT}\cdot\underbrace{\text{BMS}\emptyset + \text{BMS}\emptyset^{*}\cdot\Delta\text{DS}^{*}\cdot\text{BWT}^{*}}_{\text{Optional Fast Reset}}$

* Work supported by the Department of Energy, contract DE-AC03-76SF00515.

(Presented at the Nuclear Science Symposium, Washington, D.C., October 20-22, 1982.)

ATC Logic controls the timing of arbitration cycles. Two flip-flops and two timers regulate the execution of an arbitration cycle. Pending arbitration requests set the AG flip-flop when GK(d) is received from the current master. This starts the arbitration cycle and enables timer TM1. Participating masters assert their arbitration level on the bus for priority resolution. When TM1 signals completion of the arbitration time (EΣ), the AL lines are tested. If arbitration level zero (ΣAL = 0 an error condition) is found, AG is cleared. Otherwise the arbitration vector corresponds to the winning (highest priority) master and the Mastership Pending (MSP) flag is set. Now the arbitration cycle will wait for the current master to release the bus by terminating its AS-AK lock. After AK(d) is received and the bus has settled (ΔTAK delay), the ATC resets AG. This signals the prending master to assert GK and take bus mastership. The ATC enables timer TM2. Receipt of GK(u) from the pending master resets the MSP flag and thereby re-enables the SET AG gate. Alternately if no GK response is received in time, timer TM2 will complete the arbitration cycle by clearing the MSP flag.

The Arbitration Inhibit flip-flop asserts AI with AG(u) at the start of an arbitration cycle. This inhibits issuance of new AR requests by masters obeying the assured access protocol. Now AI remains set until all pending AR requests have been serviced, thus assuring bus mastership to low priority masters independent of the rate of requests from high priority masters. Timer TM3 requires AR = 0 for at least two bus delays before resetting AI and enabling a new set of arbitration requests.

System Handshake Logic for broadcast operations is shown at the bottom of Fig. 1. For broadcast cycles, handshake responses are generated by the SHL on behalf of all slave modules residing on a segment. Hence all transitions of AK and DK are delayed by ΔTA and ΔTD respectively to allow the slowest slaves to participate successfully in the broadcast. Propagation of address cycles [AS(u) and AK(u)] and data cycles [DS(u) and DK(u), and for block transfers DS(d) and DK(d)] from the broadcast master through all addressed segments to the end of each broadcast branch is controlled by WAIT generated by segment interconnect (SI) modules and the NWT timer TM4 in each SHL. With this mechanism AS(u) and DS(t) will propagate to the end of a branch with WT = 1 asserted by each SI along the path. On the last segment no WT is seen and the SHL generates AK(u) or DK(t) as appropriate. Now SIs on this last segment clear WT and the SHL on the next segment back toward master asserts acknowledge responses after its NWT timer TM4 has elapsed. In this fashion AK(u) and DK(t) are returned back to master. The use of WT* in the logic for AK and DK generation implements general WAIT rules. The optional fast reset shown in the RES DK equation is not mandatory. This resets DK immediately without NWT for non-block transfer cycles.

B. Geographical Address Control Section

The GAC requirements are shown in Fig. 2 and Table 2. GAC consists of an EG generator and a simple FB slave with two CSR registers. The EG generator detects geographical addresses. The address decoding logic (EGA) recognizes two formats for geographical addresses and rejects address 255 (FF hex) reserved for the GAC slave device. For a valid geographical address operation EG is set after an address decoding delay ΔTAS, selected by the designer to satisfy the decoding time requirements of a particular implementation and to meet the specification for maximum EG delay after AS(u).

The GAC slave device recognizes reserved geographical address 255 (ALA) in CSR space. The slave contains two CSR registers which are addressed by a 2-bit NTA register. The NTA register is accessable (RDNTA, LDNTA) via secondary address cycles. The CSR#0 register is

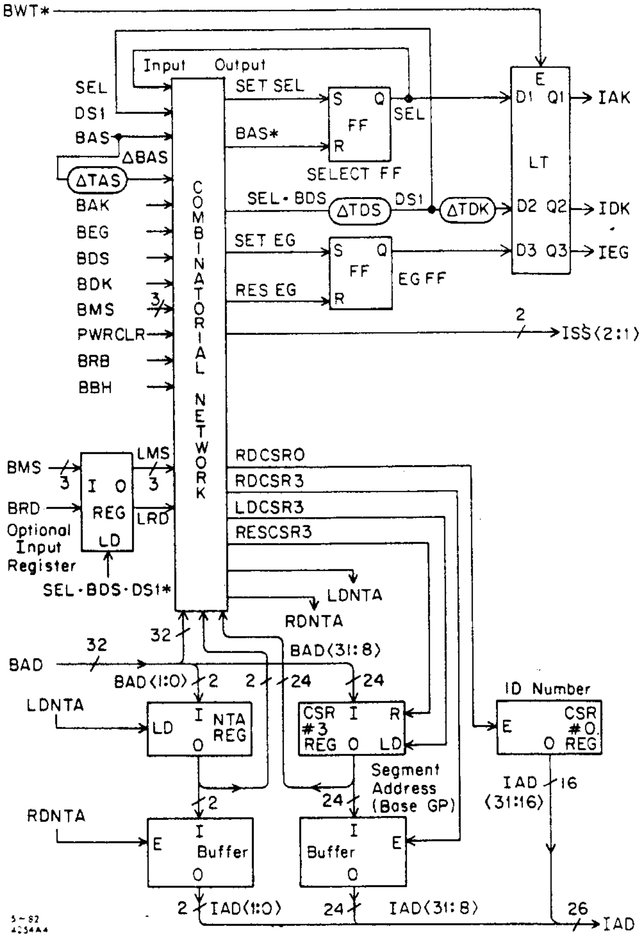

Fig. 2. GAC section basic block diagram.

Table 2. GAC section definitions and logic equations.

Definition

ΔTAS	-	Address Decoding Delay
ΔTDS	-	Data Decoding Delay
ΔTDK	-	Data Response Delay
ALA	-	Ancillary Logic Address
EGA	-	Enable Geographical Address
PWR CLR	-	Power-On Clear

Logic

SET SEL	=	ALA • (BMS=1) • BAK* • ΔBAS
SET EG	=	EGA•ΔBAS•(BMS=Ø+1)• [BEG + BAK + PWR CLR + ∫BRB•BBH*]*
RES EG	=	BAS* + BAK + PWR CLR + ∫BRB•BBH*
ALA	=	[BAD<31:Ø>=HEX(GPØ...ØFF + ØØØØ ØØFF)]
EGA	=	[BAD<7:Ø >=HEX(FF)]*• [BAD<31:8>=HEX(GPØ...Ø + ØØØØ ØØ)]
ISS<2:1>	=	(ΣBAD<31:2>=1 + BADØ ⊕ 1)•(LMS=2)•LRD*•DS1 + (LMS=0+2)*•DS1 + (LMS=0)•[(NTA=1+2) + (NTA=0)•LRD*]•DS1
RDCSRØ	=	(LMS=Ø) • (NTA=Ø) • LRD • DS1
RDCSR3	=	(LMS=Ø) • (NTA=3) • LRD • DS1
LDCSR3	=	(LMS=Ø) • (NTA=3) • LRD* • DS1 • BDK*
RESCSR3	=	PWR CLR + ∫BRB•BBH*
LDNTA	=	(LMS=2) • LRD* • DS1 • BDK*
RDNTA	=	(LMS=2) • LRD • DS1

Table 3. Ancillary logic timing specifications.

Item	Description	Time [ns]	19" Crate Segment	15 m Cable Segment
	ATC Section			
ΔTAK	Bus Clean-up Time after AK(d) -- Minimum Pulse Down Time	MIN	40	130
TM1	Arbitration Timer for AG = 1	MIN	120	440
TM2	GK(u) Timeout -- Master Address Timeout	MIN	650	830
TM3	Bus Delay Timer for AR = 0 -- 2 × Bus Delay	MIN	30	180
TM4	Bus Delay Timer for WT = 0 -- 2 × Bus Delay	MIN	30	180
ΔTA	Broadcast Address Response Delay for AK(t)	MIN	500	680
(TM5,6)		MAX	550	730
ΔTD	Broadcast Data Response Delay for DK(t)	MIN	2000	2180
(TM7,8)		MAX	2200	2380
	GAC Section			
ΔTAS	Slave Address Decoding Delay -- EG Delay	≲* MAX	60	60
ΔTDS	Slave Data Decoding Delay ΔTDS + ΔTDK	≲* MAX	1000	1000
ΔTDK	Slave Data Response Delay for DK(t)			
	Both Sections			
∫RB	Reset Bus Integration for RB(u)	MIN	100	200
		MAX	150	300
PWR CLR	Power-On Clear -- Design Choice	-	-	-

* Allow for Circuit Propagation Delays.

read-only and contains the mandatory 16-bit ID number. The CSR#3 register is 24-bit read-write and stores the segment base address (GP) used for geographical addressing. Slave status response SS = 6 is generated for error conditions. For secondary address write cycles all invalid NTA values are rejected. Only codes MS=0+2 for random data and secondary address cycles are accepted. Finally write cycles to CSR#0 are rejected. An optional input register is shown to sample the MS and RD lines at DS(u) time.

C. Ancillary Logic Timing Specifications

The timing specifications for ANC Logic are summarized in Table 3. This table is based on Table A.1.2 of the FB Specifications. Values are given for ANC Logic utilizing ECL for a 19" crate segment bus and a 15 m cable segment bus.

III. Crate Segment Implementation

A design for a standard crate segment ANC Logic has been developed. The implementation is based on 10K ECL integrated circuits and is packaged on two rear-mounted printed circuit boards. The ATC Board contains the logic described in Section II.A, Fig. 1 and Table 1 above. Implementation details are similar to the design block diagram shown for the cable segment in Fig. 4. The board also contains bus drivers, receivers and terminating resistors for the crate segment backplane. The board design utilizes approximately sixteen I.C. packages.

The GAC Board is shown in Fig. 3. The design satisfies all logic requirements discussed in Section II.B, Fig. 2 and Table 2. The hardware accommodates a 12-bit GP segment base address. Approximately thirty 10K ECL I.C. packages are utilized.

IV. Cable Segment Implementation

A design for a Cable Segment Ancillary Logic has been developed and built at SLAC to be used in a Liquid Argon Control System on the Mark II detector. The proposed system includes a VAX-FASTBUS interface, a microcontroller module, analog multiplexer modules, tiac output modules, one segment interconnect unit, a cable segment (15 meters), a cable segment ancillary logic module and terminators.

The VAX-FASTBUS interface and the microcontroller module are described in two papers at this Symposium.[4,5]

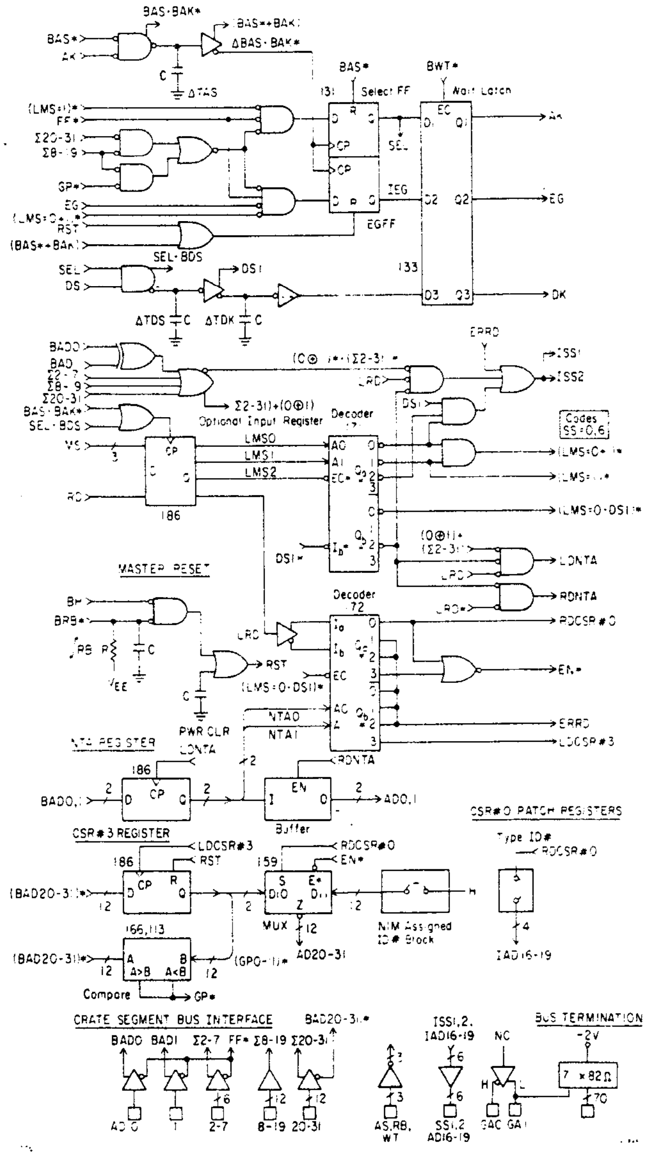

Fig. 3. Crate GAC board design block diagram.

The implementation of the Cable Segment Ancillary Logic uses 10K ECL logic, differential line drivers and receivers and active terminators for the Cable Segment connected through the Auxiliary Connector. The hardware is packaged in a single width FB module.

This module implements all the standard features described before with the exception of the line MS2 (not used). All timing values are for a 15 m cable.

The ID of this module is [140000 hex] and the CSR Register 3 is 8-bit wide. In Figs. 4 and 5 all summations (Σ) and products (Π) are done at the receiver outputs (wired OR/AND).

Figure 6 shows a typical active terminator circuit with 100 Ω impedance for the differential signal lines of the Cable Segment. Each differential line requires one termination circuit at each end of the cable. Only one module is expected to assert a line at any given time (except for AR, WT, and SR lines).

Hence a unit current value of 8 mA will be switched at the termination resistors (56 Ω) at each end of the cable (16 mA total for the driver circuit). This yields a differential voltage signal of 0.8 V at the receiver inputs. In order to allow a variable number of modules to be connected to the Cable Segment, the stacking of unit current values has to be compensated. This is achieved by means of an operational amplifier controlling a current source. Hence the current through the termination resistor is maintained at 8 mA.

Finally the termination voltage V_{TERM} is common to all lines and is adjusted to set the quiescent voltage levels for the differential lines in reference to V_{REF} selected as V_{BB}.

V. Summary and Acknowledgements

Details in this paper are based on the June 1982 FASTBUS Tentative Specifications. All applicable changes made in this document by the FB Working Groups through October 1982 have been included. However the final U.S. Department of Energy document of the FASTBUS Specifications may include additional minor changes affecting FB ANC Logic.

The Crate ANC Logic design shown in Section III is presently in PC layout.[2] Prototypes of the ATC and GAC boards are expected by the end of December 1982.

The Cable ANC Logic module shown in Section IV has been fabricated as a wirewrap prototype and is presently being tested.[3]

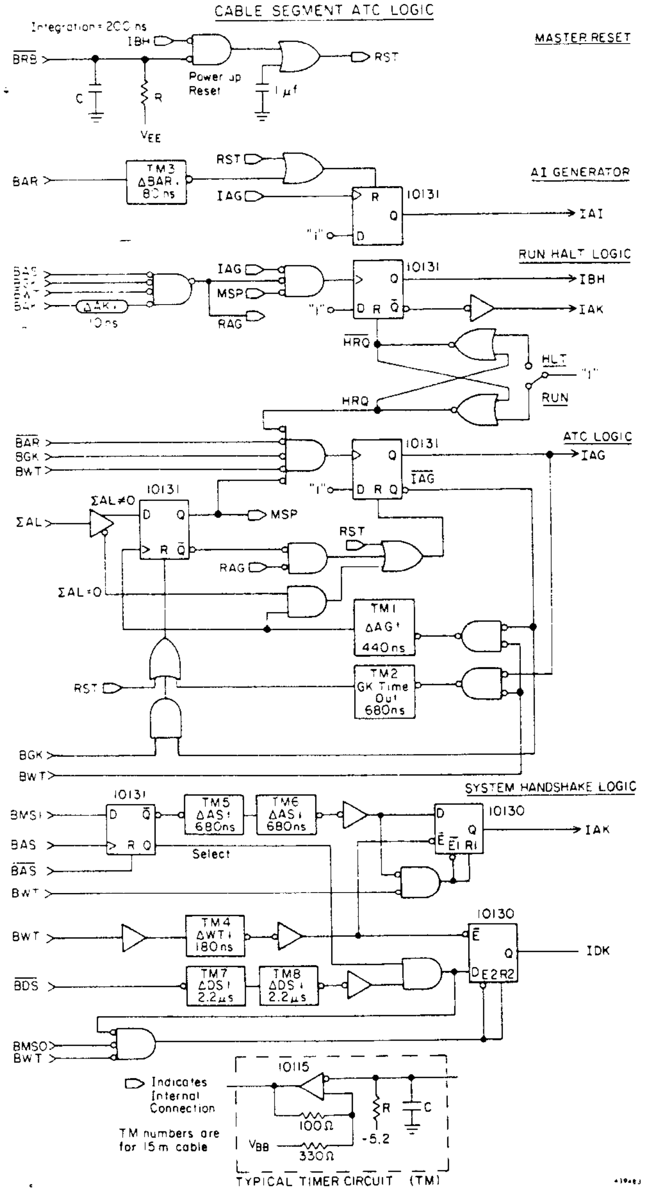

Fig. 4. Cable ATC section design block diagram.

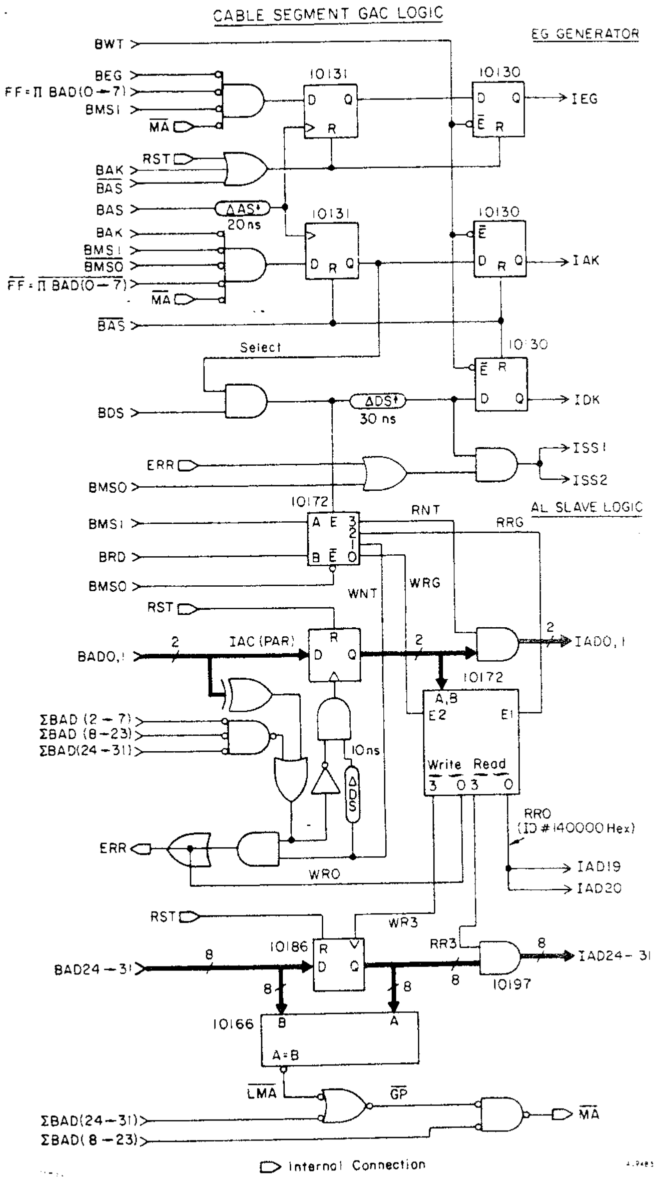

Fig. 5. Cable GAC section design block diagram.

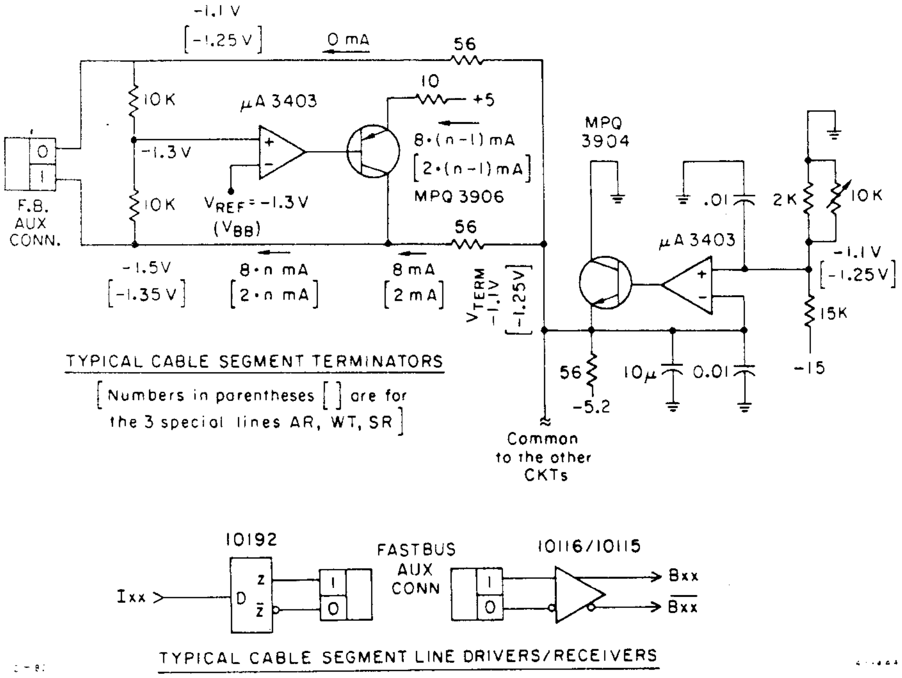

Fig. 6. Typical ECL differential terminators and line interface.

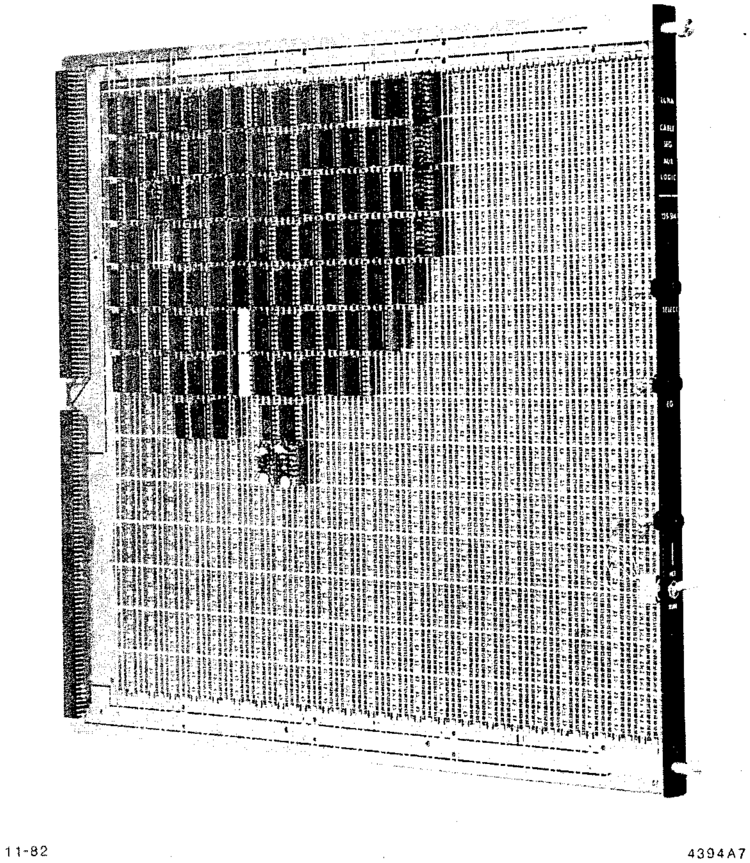

Fig. 7. Prototype cable ancillary logic module.

The Ancillary Logic requirements and Differential Cable Segment details presented in this paper are based on earlier contributions by a number of people on the FASTBUS committee, including R. Downing (University of Illinois), L. Paffrath (SLAC), L. Pregernig (CERN), E. J. Barsotti (FNAL), and W. K. Dawson (TRIUMF). Continued support of this work by R. S. Larsen is gratefully appreciated.

References

1. "FASTBUS Tentative Specification," U.S. NIM Committee, available from L. Costrell, NBS, Washington, D.C.
2. FASTBUS Crate Ancillary Logic Documentation, ATC Board SLAC 135-853, and GAC Board SLAC 135-854.
3. FASTBUS Cable Ancillary Logic Documentation, SLAC 135-514.
4. FASTBUS Host Interface for VAX/VMS, E. J. Siskind, Brookhaven National Laboratory; paper contributed to the IEEE Nuclear Science Symposium, Washington, D.C., October 1982.
5. A FASTBUS Controller Module Using a Multibus MPU, S. R. Deiss, Stanford Linear Accelerator Center; paper contributed to the IEEE Nuclear Science Symposium, Washington, D.C., October 1982.