International Conf. on Accelerator & Large Experimental Physics Control Systems; Nucl. Instrum. & Methods Proc. Series; Vancouver, Canada, 10/30-11/3/89.

BNL--43066

DE90 003645

PROGRAMS FOR CONTROL OF AN ANALOG-SIGNAL SWITCHING NETWORK*

T. D'Ottavio, R. Enriquez, R. Katz, and J. Skelly.
AGS Department, Brookhaven National Laboratory, Upton, NY 11973, USA

A suite of programs has been developed to control the network of analog-signal switching multiplexers in the AGS complex. The software is driven by a relational database which describes the architecture of the multiplexer tree and the set of available analog signals. Signals are routed through a three-layer multiplexer tree, to be made available at four consoles each with three 4-trace oscilloscopes. A menu-structured operator interface program is available at each console, to accept requests to route any available analog signal to any of that console's 12 oscilloscope traces. A common routing-server program provides automatic routing of requested signals through the layers of multiplexers, maintaining a reservation database to denote free and in-use trunks. Expansion of the analog signal system is easily accommodated in software by adding new signals, trunks, multiplexers, or consoles to the database. Programmatic control of the triggering signals for each of the oscilloscopes is also provided.

*Work performed under the auspices of the U.S. Department of Energy.

1. INTRODUCTION

The operators of the AGS have traditionally employed an extensive network of crossbar-multiplexers to route analog signals into the Main Control Room (MCR) for viewing on oscilloscopes. This network currently includes >50 multiplexers routing ~1500 analog signals. The recent improvement in MCR facilities has expanded the number of consoles and replaced the central analog multiplexers. The impending advent of the Booster at BNL promises to put increased pressure on this system. We describe here a suite of programs which controls this network of multiplexers. These programs provide intuitive operator interaction with the multiplexer system, concealing routing details much as a modern telephone switching system does. The programs rely on a relational database to describe the network, and use list-driven menus to guide the operators through the extensive inventory of available signals.

2. MULTIPLEXER NETWORK

The new 3-layer multiplexer network, fig 1, is expandable in both the number of consoles and the number of signals supported; the present implementation supports 4 consoles, each with an independent set of 12 signal spigots, and ~60 field multiplexers with a total of 208 trunks (ie multiplexer output channels) which direct signals from field multiplexers into the central facility. These field multiplexer trunks are partitioned into a low frequency set feeding the central digital-input multiplexer (160 trunks) which accommodates predigitized signals and distributes them digitally [1]; and a high frequency set feeding the central analog-input multiplexer (48 trunks) which uses coaxial relays to switch signals. The final distribution of signals for each console is made with a "shuffle" multiplexer (also using coaxial relays) which permits any of that console's 12 signals to be switched to any of its 12 oscilloscope inputs.

A variety of multiplexers are supported on the multiplexer network. Present-generation multiplexers are connected to "device controllers" which report to "stations" on the Relway control network [2]; these include the new central multiplexers, and ~10 field multiplexers. Booster era multiplexers will be connected to device controllers reporting to "stations" which are Apollo nodes on the token-ring network, avoiding the need for the Relway network. We also support an inventory of ~45 "traditional" (Datacon) field multiplexers which are directly controlled from the "traditional" control system on a DecSystem-10 computer. A process (DCMUX) running on the DecSystem-10 emulates the function of a device controller and communicates with the Apollo network (specifically an "Apollo station") via the Relway, to provide transparent control of these traditional multiplexers to Apollo software.

The present inventory of field multiplexers feeding the central digital-input multiplexer are of an older design, with coaxial cable trunks and centrally located signal digitizers; Booster-era (low-frequency) field multiplexers will digitize their trunk signals at the multiplexer and use fiber-optic cable between the field multiplexer and the central digital-input multiplexer.

3. COMPUTER ENVIRONMENT

The new AGS control system comprises a net of ~40 Apollo workstations, which communicate via Apollo's proprietary token-ring network. Two workstations are available at each MCR console for Operator programs. The Apollo workstations run the Aegis Operating System which provides for a multi-window multi-task user environment with excellent graphic facilities. One of these two workstations is dedicated to two standard programs, one of which is the operator interface for the analog multiplexer system.

4. SOFTWARE COMPONENTS

A number of software components cooperate to implement control of the analog signal multiplexer network. At each console an operator interface program is available, named XBAR (pronounced crossbar), through which an operator may enter signal connect and disconnect requests, and view the list of signals presently routed to that console. A common server program, called the ROUTING SERVER, is used to implement the connect and disconnect requests from all copies of XBAR at the multiple consoles. Tools are also available by which any application software code may forward signal connection requests to the ROUTING SERVER. A commercial relational database system maintains relations which describe the multiplexer network and available signals, as well as the dynamic state of trunk usage as the ROUTING

SERVER implements requests to connect and disconnect signals to the various consoles. Figure 2 indicates the relationships among these software components and with the hardware systems.

5. OPERATOR INTERFACE PROGRAM - XBAR

XBAR provides a highly graphical interface by which the operator may view the analog signals currently routed to the analog signal spigots at a console, and request connection/disconnection of signals to these spigots (fig 3). The inventory of signals available in the multiplexer network is described in a menu tree; XBAR presents this tree to the operator as a series of pop-up menus to guide the operator through the process of implementing a signal connection to a spigot at the console (fig 4). Additional options are available to the operator via XBAR: eg, disconnect a signal, copy a signal from one spigot to another, switch signals between spigots. Each of these requests is forwarded to the ROUTING SERVER, and, upon return acknowledgement, is reflected in the display of currently-connected signals which is maintained by XBAR.

It may occur that an operator's request to connect a signal cannot be fulfilled, because of limited trunk capacity between the field multiplexer and the central multiplexer facility, or between the central input multiplexer and that console's shuffle multiplexer. In this event, the ROUTING SERVER so notifies the requesting XBAR and provides details of the interference. If the interference is local (limited trunk capacity into the console shuffle multiplexer), the operator is so advised and permitted either to abort the request or to disconnect other signals to free an appropriate trunk. If the interference is global (limited trunk capacity from a field multiplexer) with possible impact on other consoles, the operator is so advised via a more extensive pop-up display/menu; the operator is permitted to clear the interference by selecting which signals and which consoles shall be disconnected, with these requests again being forwarded to the ROUTING SERVER. In this case, the ROUTING SERVER advises the XBAR program(s) at the affected console(s) of this change in their signal environment, so that they may update their display of connected signals appropriately.

XBAR also provides the interface by which the operator may program the trigger signals for each of the console's oscilloscopes. The bottom half of the XBAR display (fig 3) is devoted to this function. The trigger signal may be constructed from a library of up to 16 "start signals", with subsequent delay of up to 3 stages; each delay stage is programmed as an arbitrary count of pulses of a clock signal from the library of up to 16 clocks. Clocks available include various real-time clocks, the AGS magnetic field clock, and the AGS RF signal. These libraries are designed to accommodate the additional start signals and clocks which Booster operations will require.

6. ROUTING SERVER

One copy of the ROUTING SERVER program is required in the Apollo network, to respond to requests from the various XBAR programs and other possible application programs, and to implement their connection/disconnection requests. The ROUTING SERVER is responsible for determining all details of a signal routing path; the connection request forwarded by XBAR specifies only the desired signal and the target console and spigot. The ROUTING SERVER examines the relational database [3] to determine the location in the multiplexer network of the desired signal, and constructs a routing path to bring that signal to the requested spigot, and then issues the necessary commands to the multiplexer trunks to realize this routing. When the routing has been accomplished, a reply is issued to the requesting program acknowledging completion of the request. Should the requested signal connection not be realizable (eg, requested signal not found in the database, or no free trunk available), a failure reply is sent to the requesting program.

The ROUTING SERVER conserves trunk consumption by distributing multiply-viewed signals at the highest possible multiplexer, ie first at the console shuffle multiplexer, then at the central input multiplexer. This routing strategy has the consequence that lower level trunks may become simultaneously used by multiple spigots. The ROUTING SERVER ensures that no trunk is re-used or freed while still routing a signal being viewed elsewhere. This function is accomplished by maintaining a reservation record for each spigot, which applies to all the trunks required to route the signal to that spigot. This reservation record is maintained in the relational database; the commercial database product assures that this information is preserved across failure/restart of the software processes. A trunk may be liberated either by a specific disconnection request, or by a request to connect a new signal to a spigot which formerly viewed the signal routed by that trunk.

7. COMMUNICATIONS BETWEEN XBAR AND ROUTING SERVER

Two classes of communications are required between the ROUTING SERVER and the XBAR program at a console: (1) a synchronous request/reply sequence when the operator requests a connection or disconnection; (2) an asynchronous message from the ROUTING SERVER to XBAR advising that the signal environment (ie set of signals being viewed at the console) has been changed. The change in signal environment which prompts the asynchronous message may occur because (1) some applications program at that console (other than XBAR) has requested connection of some signal(s) (presumably as a result of an operator initiative), or (2) an operator at another console has solved a "global" interference problem with a requested signal connection by disconnecting a signal from this console. This asynchronous signal environment advisory message is needed if the XBAR is to be able to maintain a reliable display of signals currently routed to the console. The ability of one console to disconnect another console's signals to resolve its own routing impediments plainly opens the door to warfare between consoles, but relieves the need to move to another console to clear a global impediment. One presumes, as a design strategy, that a cooperative working relationship prevails among the operators, without requiring the control system to act as umpire.

The synchronous communication protocol between the ROUTING SERVER and XBAR also supports other types of messaging. Several diagnostic facilities are included to facilitate maintenance of the analog multiplexer system. One such facility permits an operator to request a private (additional) routing of a signal to a spigot; this technique may be useful to detect malfunctioning trunks. Another facility permits XBAR to request the ROUTING SERVER to verify that all trunks used in routing a particular signal are commanded as necessary, and to report that routing to XBAR. Moreover, XBAR may request the ROUTING SERVER to send it a signal environment advisory.

8. RELATIONAL DATABASE

Relations are defined in the relational database to describe: multiplexers (Table I), trunks (Table II), and signals (Table III); in addition, a spigots relation maintains the reservation record for the spigots. Tables I, II, and III illustrate portions of these relations. A Structured Query Language (SQL) is available to enter data into these relations, and to examine them. SQL examination of these relations is expected not to be normally necessary, but is available for diagnostic and maintenance purposes. Canned procedures have been constructed to facilitate entry into the database of new signals, trunks, and multiplexers by the technical staff of the Operations group.

TABLE I - MULTIPLEXERS RELATION

MUX NAME	NUM INPUTS	NUM OUTPUTS	MUX TYPE
LXB.RAD_MONS	32	2	5 = PDP_10_MUX
LXB.POS_MON	16	2	5 = PDP_10_MUX
AXB.PUE_MUX	32	2	5 = PDP_10_MUX
AXB.MISC1_EC	64	10	5 = PDP_10_MUX
AXB.RF.MISC1	32	4	3 = FIELD_MUX
CXB.ANAL_CENTRAL	48	16	1 = CENTRAL_ANA_MUX
CXB.DGTL_CENTRAL	160	32	0 = CENTRAL_DIG_MUX
CXB.SHFL_MCR_1	12	12	2 = SHUFFLE_MUX
CXB.SHFL_MCR_3	12	12	2 = SHUFFLE_MUX
CXB.SHFL_MCR_4	12	12	2 = SHUFFLE_MUX
CXB.SHFL_MCR_5	12	12	2 = SHUFFLE_MUX

Some comments may elucidate the function of the trunks and the signals relations. For the trunks relation: the field labeled "MUX_NAME" indicates the multiplexer from which the trunk originates; the field labeled signal name will be used dynamically by the ROUTING SERVER to record the signal switched onto the trunk when it is in use. For the signals relation: the fields labeled "MUX_NAME" and "CHANNEL_NUM" indicate the multiplexer and input channel into which the signal is cabled. Field multiplexer trunks are cabled as inputs into the CXB.ANAL_CENTRAL central analog-input multiplexer; thus these trunks appear in the signals relation as input signals to that multiplexer. Likewise, some trunks from CXB.ANAL_CENTRAL are cabled as inputs into the CXB.SHFL_MCR_1 shuffle multiplexer for console MCR_1, and thus appear in the signals relation as input signals to that multiplexer.

It is this trail of connections in the database that the ROUTING

SERVER is required to follow in establishing a routing path for a requested connection of signal to console spigot. This procedure is accomplished using qualified retrievals from the database, much as a programmer would do using an SQL. For example, the SQL request

PRINT SIGNALS WITH SIGNAL_NAME = "AXI.I3H.DAMPR.DIFF"

would return the information that this signal was available on the multiplexer named AXB.RF.MISC1; then one would use

PRINT TRUNKS WITH MUX_NAME = "AXB.RF.MISC1" AND

SIGNAL_NAME = "none"

to retrieve a list of free trunks from this multiplexer; one would then choose, perhaps, AXO.RF.MISC1.A, and request

PRINT SIGNALS WITH SIGNAL_NAME = "AXO.RF.MISC1.A"

in order to discover the central-input-multiplexer and channel to which AXO.RF.MISC1.A had been cabled, ie CXB.ANAL_CENTRAL channel 32. Then one needs a trunk from this multiplexer which goes to the shuffle multiplexer for the desired console, and the procedure continues.

TABLE II - TRUNKS RELATION

TRUNK NAME	MUX NAME	SIGNAL NAME	IN USE
LXO.RAD_MONS.A	LXB.RAD_MONS	none	0
LXO.RAD_MONS.B	LXB.RAD_MONS	none	0
LXO.POS_MON.A	LXB.POS_MON	none	0
LXO.POS_MON.B	LXB.POS_MON	none	0
AXO.RF.MISC1.A	AXB.RF.MISC1	none	0
AXO.RF.MISC1.B	AXB.RF.MISC1	none	0
AXO.RF.MISC1.C	AXB.RF.MISC1	none	0
AXO.RF.MISC1.D	AXB.RF.MISC1	none	0
CXO.ANAL_MCR_1.A	CXB.ANAL_CENTRAL	none	0
CXO.ANAL_MCR_1.B	CXB.ANAL_CENTRAL	none	0
CXO.ANAL_MCR_1.C	CXB.ANAL_CENTRAL	none	0
CXO.ANAL_MCR_1.D	CXB.ANAL_CENTRAL	none	0

One final comment about the names of the objects in these relations may be illuminating. The initial letter of each name designates the "accelerator component" to which the object belongs: L for Linac, A for AGS, C for Controls (the central facilities), B for Booster, etc.

TABLE III - SIGNALS RELATION

SIGNAL NAME	MUX NAME	CHANNEL NUM
AXI.I3H.DAMPR.DIFF	AXB.RF.MISC1	0
AXI.I3V.DAMPR.DIFF	AXB.RF.MISC1	1
AXI.I3H.DAMPR.V	AXB.RF.MISC1	2
AXI.I3V.DAMPR.V	AXB.RF.MISC1	3
AXO.RF.MISC1.A	CXB.ANAL_CENTRAL	32
AXO.RF.MISC1.B	CXB.ANAL_CENTRAL	33
AXO.RF.MISC1.C	CXB.ANAL_CENTRAL	34
AXO.RF.MISC1.D	CXB.ANAL_CENTRAL	35
CXO.ANAL_MCR_1.A	CXB.SHFL_MCR_1	0
CXO.ANAL_MCR_1.B	CXB.SHFL_MCR_1	1
CXO.ANAL_MCR_1.C	CXB.SHFL_MCR_1	2
CXO.ANAL_MCR_1.D	CXB.SHFL_MCR_1	3

9. References

[1] R. Mariotti, W. Buxton, R. Frankel, "Digital Capture and Optical Distribution of Analog Signals"; 1989 Particle Accelerator Conference, Chicago, Ill.

[2] A. Stevens, T. Clifford, R. Frankel, "Distribution of Computer Functionality at the Brookhaven AGS", IEEE Trans. Nucl. Sci., NS-32, 2023 (1985).

[3] T. Clifford, R. Katz, C. Griffiths, "A C Programmer's View of a Relational Database", this conference.

10. Figure Captions

fig 1 - Analog signal crossbar multiplexer system architecture.
fig 2 - Analog signal crossbar multiplexer software.
fig 3 - The XBAR normal operator display. The scope-1-channel-2 field is highlighted to indicate that it has been selected; it will be the target of a signal connection operation if the "Connect" option is picked. The start field under trigger-2 (for scope 2) is highlighted to indicate that it has been

selected; trigger-2 will be the target of a trigger definition operation if the "Triggers" option is picked.

fig 4 - The XBAR display during a "Connect" operation. The operator has stepped down the menu tree to the lowest level.

fig 1 - ANALOG SIGNAL CROSSBAR-MULTIPLEXERS

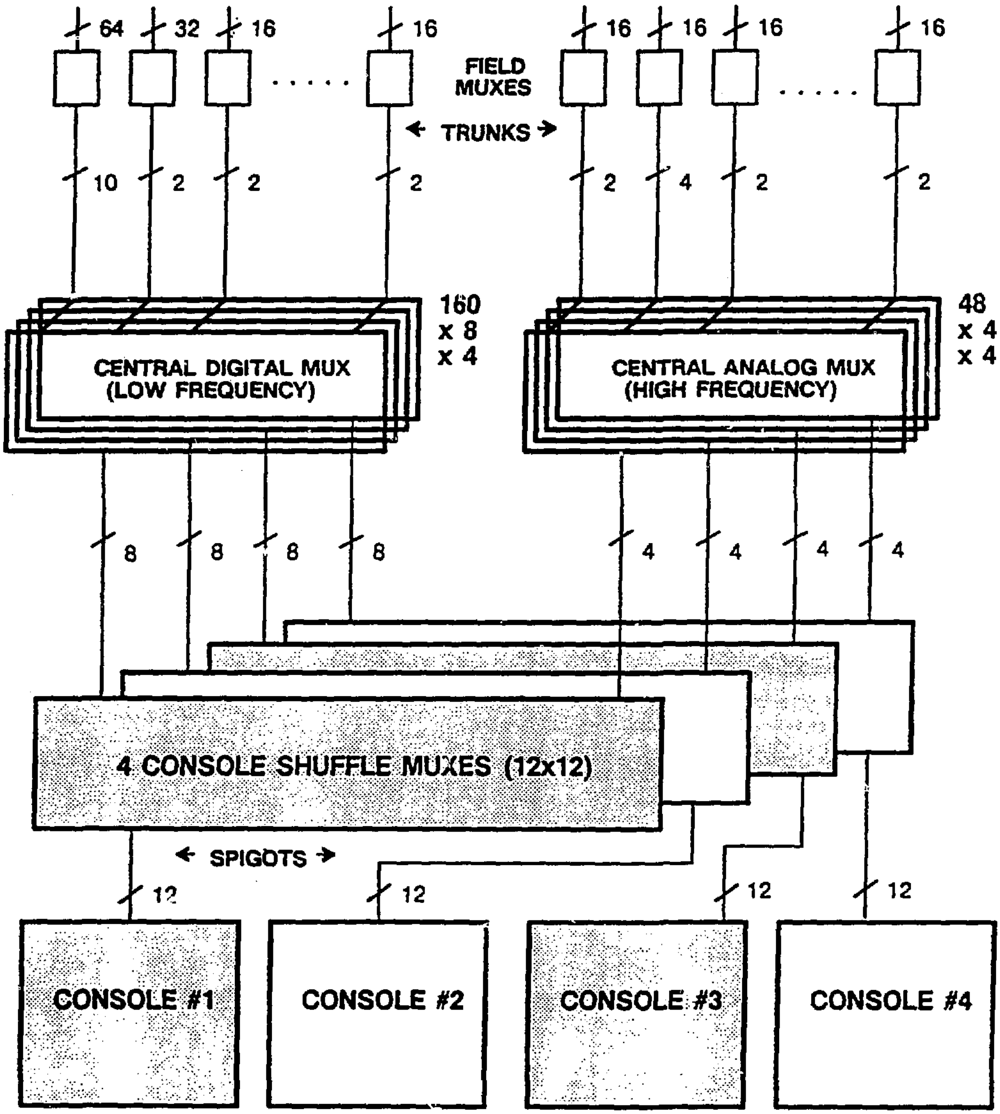
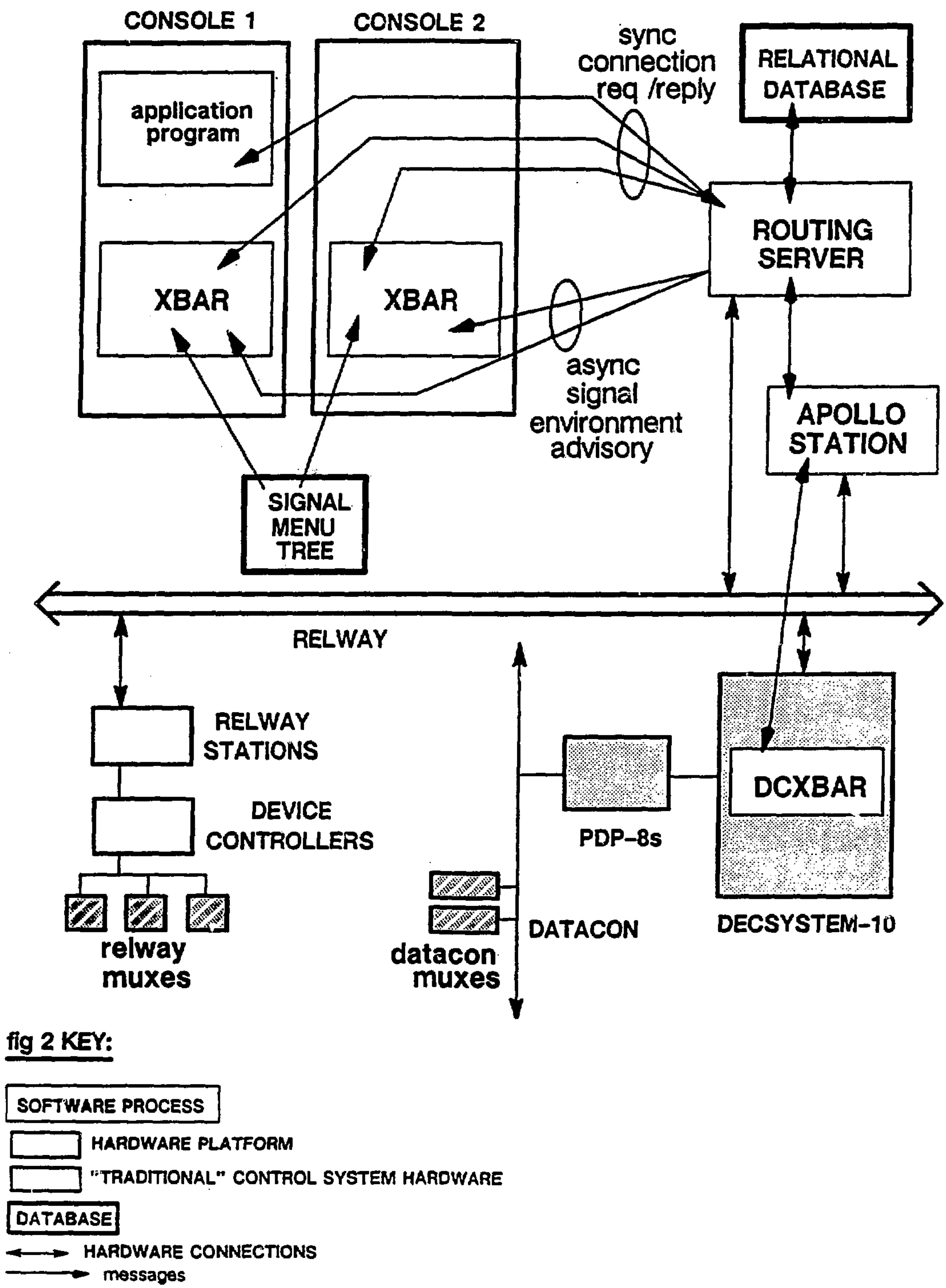


fig 2 – ANALOG SIGNAL CROSSBAR–MUX SOFTWARE



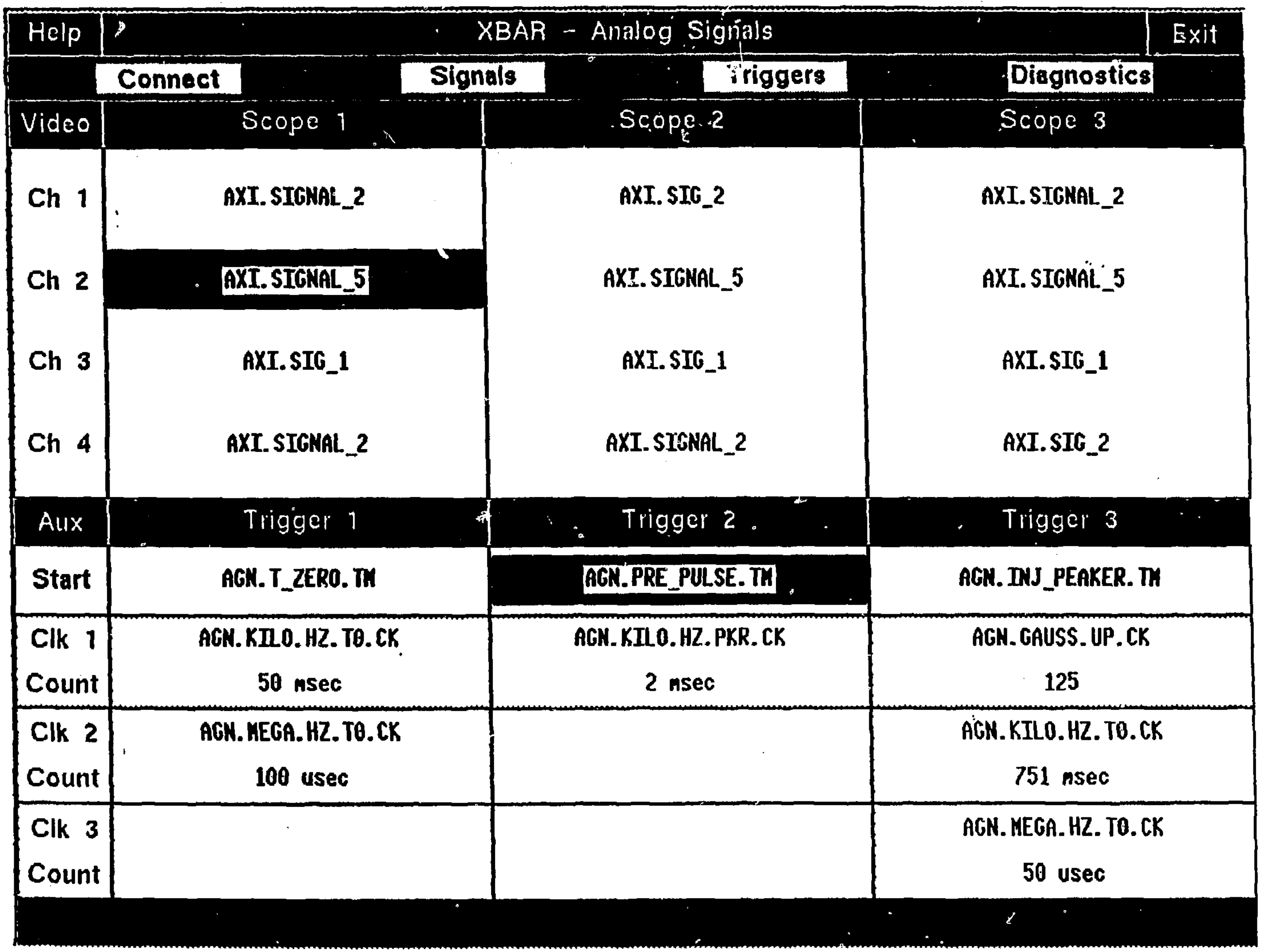

Help
XBAR - Analog Signals
Exit
Connect
Signals
Triggers
Diagnostics
Video
Scope 1
Scope 2
Scope 3
Ch 1
AXI.SIGNAL_2
AXI.SIG_2
AXI.SIGNAL_2
Ch 2
AXI.SIGNAL_5
AXI.SIGNAL_5
AXI.SIGNAL_5
Ch 3
AXI.SIG_1
AXI.SIG_1
AXI.SIG_1
Ch 4
AXI.SIGNAL_2
AXI.SIGNAL_2
AXI.SIG_2
Aux
Trigger 1
Trigger 2
Trigger 3
Start
AGN.T_ZERO.TM
AGN.PRE_PULSE.TM
AGN.INJ_PEAKER.TM
Clk 1
Count
AGN.KILO.HZ.T0.CK
50 msec
AGN.KILO.HZ.PKR.CK
2 msec
AGN.GAUSS.UP.CK
125
Clk 2
Count
AGN.MEGA.HZ.T0.CK
100 usec
AGN.KILO.HZ.T0.CK
751 msec
Clk 3
Count
AGN.MEGA.HZ.T0.CK
50 usec

fig 3

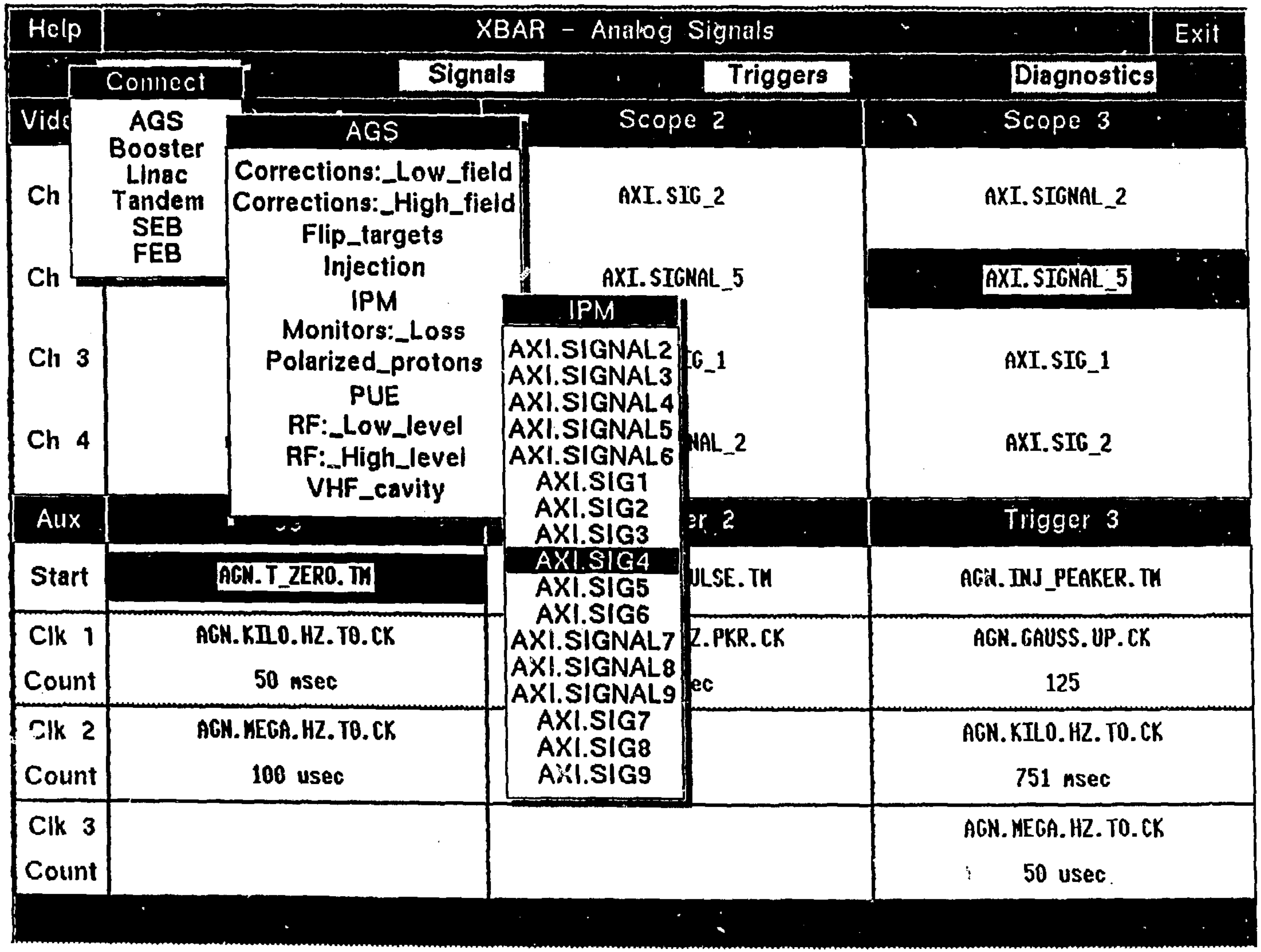

Help
XBAR - Analog Signals
Exit
Connect
Signals
Triggers
Diagnostics
AGS
Booster
Linac
Tandem
SEB
FEB
AGS
Corrections:_Low_field
Corrections:_High_field
Flip_targets
Injection
IPM
Monitors:_Loss
Polarized_protons
PUE
RF:_Low_level
RF:_High_level
VHF_cavity
IPM
AXI.SIGNAL2
AXI.SIGNAL3
AXI.SIGNAL4
AXI.SIGNAL5
AXI.SIGNAL6
AXI.SIG1
AXI.SIG2
AXI.SIG3
AXI.SIG4
AXI.SIG5
AXI.SIG6
AXI.SIGNAL7
AXI.SIGNAL8
AXI.SIGNAL9
AXI.SIG7
AXI.SIG8
AXI.SIG9
Scope 2
AXI.SIG_2
AXI.SIGNAL_5
Scope 3
AXI.SIGNAL_2
AXI.SIGNAL_5
AXI.SIG_1
AXI.SIG_2
Ch 3
Ch 4
Aux
Trigger 3
Start
AGN.T_ZERO.TM
AGN.INJ_PEAKER.TM
Clk 1
Count
AGN.KILO.HZ.TO.CK
50 msec
AGN.GAUSS.UP.CK
125
Clk 2
Count
AGN.MEGA.HZ.TO.CK
100 usec
AGN.KILO.HZ.TO.CK
751 msec
Clk 3
Count
AGN.MEGA.HZ.TO.CK
50 usec

fig 4

DISCLAIMER